UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
IECS International Doctoral School

A PORTABLE V-SLAM BASED SOLUTION FOR ADVANCED VISUAL 3D MOBILE MAPPING

Alessandro Torresani

Advisor

Dr. Fabio Remondino

Fondazione Bruno Kessler

Co-advisor

Dr. Fabio Menna

Fondazione Bruno Kessler

December 2022

Abstract

The need for accurate 3D reconstructions of complex and large environments or structures has risen dramatically in recent years. In this context, devices known as portable mobile mapping systems have lately emerged as fast and accurate reconstruction solutions. While most of the research and commercial works have relied so far on laser scanners, solutions solely based on cameras and photogrammetry are attracting an increasing interest for the minor costs, size and power consumption of cameras. This thesis presents a novel handheld mobile mapping system based on stereo vision and image-based 3D reconstruction techniques. The main novelty of the system is that it leverages Visual Simultaneous Localization And Mapping (V-SLAM) technology to support and control the acquisition of the images. The real-time estimates of the system trajectory and 3D structure of the scene are used not only to enable a live feedback of the mapped area, but also to optimize the saving of the images, provide geometric and radiometric quality measures of the imagery, and robustly control the acquisition parameters of the cameras. To the best of authors' knowledge, the proposed system is the first handheld mobile mapping system to offer these features during the acquisition of the images, and the results support its advantages in enabling accurate and controlled visual mapping experiences even in complex and challenging scenarios.

Keywords

[Portable mobile mapping, photogrammetry, V-SLAM, stereo vision, real-time, quality control]

Acknowledgements

Foremost, I would like to thank my supervisors Fabio Remondino and Fabio Menna, who always kept me motivated and wisely guided throughout my doctoral studies. Besides my supervisors, I would like to express my sincere gratitude to Franscesco Fassi and Pierre Drap, for their valuable and constructive feedback as reviewers of this thesis. My warmest thanks go also to all the wonderful people I met and worked with during my doctoral studies. I would especially like to thank my colleagues Roberto Battisti, Simone Rigon, Elisa Mariarosaria Farella, Michela Lecca and Eleonora Grilli, with whom I had the pleasure to share many collaborations and fruitful discussions, and from whom I learned a lot. Last but not least, my deepest gratitude goes to my parents Anna Maria and Arrigo, and my partner Giulia. Without their help and support, this work would not have been possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Three-dimensional (3D) mapping / reconstruction is defined as the process of deriving shape and appearance of *real* objects from multiple *sensor* measurements. When the sensor data is collected from a moving system and typically in an automatic fashion, we talk about *mobile* 3D mapping. This technique is becoming more and more crucial today for the 3D reconstruction of complex and extended structures or environments, where static acquisition techniques, e.g. terrestrial laser scanning or close range photogrammetry, would be highly inefficient in terms of time. This thesis is positioned in the context of *portable* mobile 3D mapping, which comprises all the systems intended to be carried and used by a walking person.

1.1 Context and Motivations

Portable mobile 3D mapping systems, or simply portable mobile mapping systems (PMMSs), were mainly born for the rapid and agile 3D mapping of those environments or structures that are not easily accessible by vehicles or drones. Common examples include rugged outdoor scenarios, indoor or underground structures, with frequent applications comprising inspection and monitoring, digitization of heritage assets, forestry mapping, soil analysis, or the generation of 3D models for virtual and augmented reality applications.

The PMMS context has drawn a lot of research and commercial attention since the first experimental systems were introduced roughly ten years ago [1]. So far, for 3D mapping, the vast majority of the works have relied on Light Detection And Ranging (LIDAR) sensors and laser scanner systems [2]. Although the latter have undoubted advantages in terms of accuracy, immediacy of the 3D result, acquisition frequency and capability to work effectively on different surfaces and environmental conditions, their costs are very high. Moreover, LIDAR technology alone does not provide color and texture information, necessitating the use of integrative photographic acquisition when the latter are required. This lately has motivated a strong interest towards the research and development of alternative and low-cost solutions.

Purely visual systems based on standard cameras and image-based 3D reconstruction techniques [3] are among the most interesting alternatives. Cameras are substantially simpler than laser scanners, resulting in lower prices, weight and power consumption as well as in less working range limitations due to the absence of emitters. Furthermore, thanks to remarkable advances in algorithms and computational capabilities of commodity hardware, image-based 3D reconstruction methods have improved dramatically in the previous decade [4], and can provide both geometric and color/texture information using a single sensor. Still, the visual and portable mobile mapping has several important and open challenges to address:

- **Sensor challenges**. Cameras are primarily light acquisition devices, so they require a sufficiently and possibly evenly lit scene to produce acceptable results. Unfortunately, this is not always the case, and the real world often presents strong illumination variations. In these cases, the dynamic range of the cameras might not be sufficiently wide to correctly expose all the scene elements, and the images are likely to contain burned/overexposed and/or extremely dark/underexposed regions (Figure 1.1a). Another important problem of moving visual systems is motion-blur. This blur effect occurs when the camera moves significantly during the exposure phase of the sensor (Figure 1.1b). Exposure and blur issues cause considerable information and quality losses in the images [5, 6], which can have a significant impact on the correctness and completeness of the reconstruction.

- **Algorithm challenges**. While sensor measurements of LIDAR systems, e.g. laser scanners or ToF cameras, are already three-dimensional, visual systems require extra algorithmic steps and mathematical models to convert multiple overlapped images to three-dimensional estimates. This makes the visual mapping significantly more complicated and computationally demanding than the LIDAR counterpart. Furthermore, there are still situations where the current algorithms fail to deliver accurate and complete results, like for example when working with poorly-textured (Figure 1.1c) and/or highly reflective (Figure 1.1d) surfaces [7, 8].

- **Practical challenges**. Due to the limited computational capabilities of existing portable hardware, the mapping process of visual systems is practically carried out in post-acquisition today [9, 10], typically leveraging the computational resources of powerful workstations. This impractical separation between acquisition and estimation phases currently makes a predictable use of visual systems an open problem [11, 12]. Without a real-time feedback on the mapping results, it is almost impossible for an operator to determine the correctness and completeness of the image acquisition, and possible areas of the environment could be missed or not sufficiently covered.

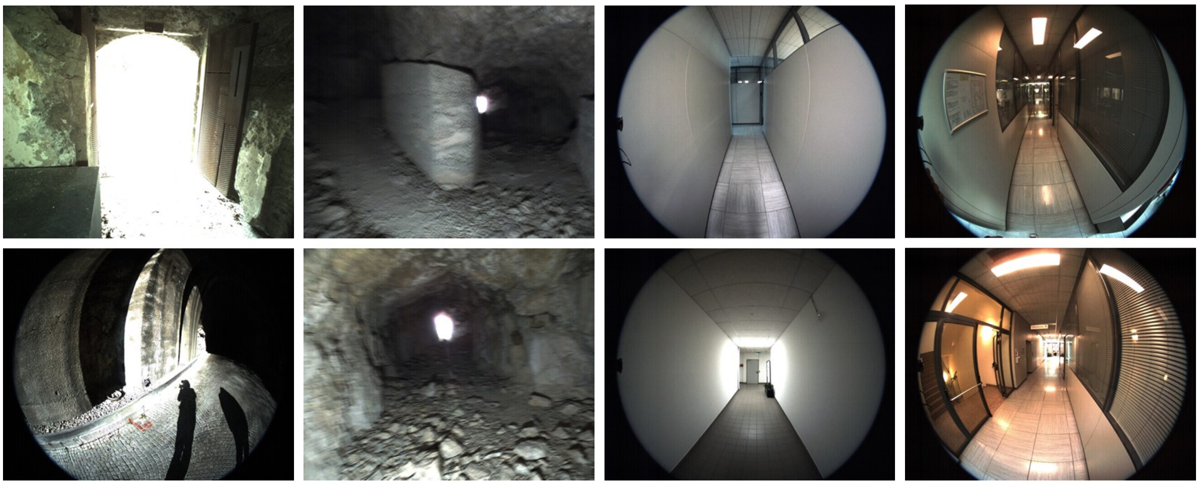


Figure 1.1: Some of the open challenges of portable visual mobile 3D mapping. (a) Strong illumination variations. (b) Motion-blur. (c) Poorly-textured environments. (d) Reflective surfaces.

1.2 Contributions and Outline

Motivated by the above observations, this thesis presents a novel visual portable (handheld) mobile mapping system. The key innovation of the system is to support and control the image acquisition with a real-time and sparse three-dimensional reconstruction of the acquired environment. The proposed system leverages a Visual Simultaneous Localization And Mapping (V-SLAM) algorithm, based on the open-source OpenVSLAM [13], to have continuous and real-time estimates of its local position and attitude in the environment as well as of the three-dimensional structure of the surrounding scene. This provides several advantages. As a starting point, the user can leverage the real-time 3D reconstruction to assess the overall progress of the acquisition and check that all the target areas are well covered by the images. Besides, the estimates of the system position and scene structure can be leveraged to optimally control the acquisition of the images, provide quality indicators of the acquisition distance and speed, and control robustly the camera exposure also in presence of challenging illumination conditions. To the best of the authors' knowledge, this is the first example of a visual handheld mobile mapping system that provides real-time feedback and assistance during the acquisition of the images, which cover both geometric and radiometric aspects of the imagery. Summarizing, the main contributions of this thesis work are:

- A novel visual handheld mobile 3D mapping system, named GuPho (**Gu**ided **Pho**togrammetric (system)), that leverages a real-time and sparse reconstruction of the scene to assist and guide the image acquisition process.
- Different methods for controlling the acquisition of the images that consider the pose of the system and the three-dimensional structure of the scene, which can ensure proper image overlap and an optimized amount of images.
- Methods for monitoring the acquisition distance and speed, and help the user to achieve a better positioning while avoiding, at the same time, important motion-blur problems.
- A novel camera exposure control and metering method that exploits the known structure and depth of the scene to perform an object-based exposure control, which can significantly help in situations of strong illumination variations.
- A Low-cost, lightweight and modular system design based on stereovision that can be easily adapted to different working scenarios.

The material presented in this thesis is the result of several own scientific publications. While [14] laid some of the system foundations, in particular regarding the acquisition control methods, the first version of the system

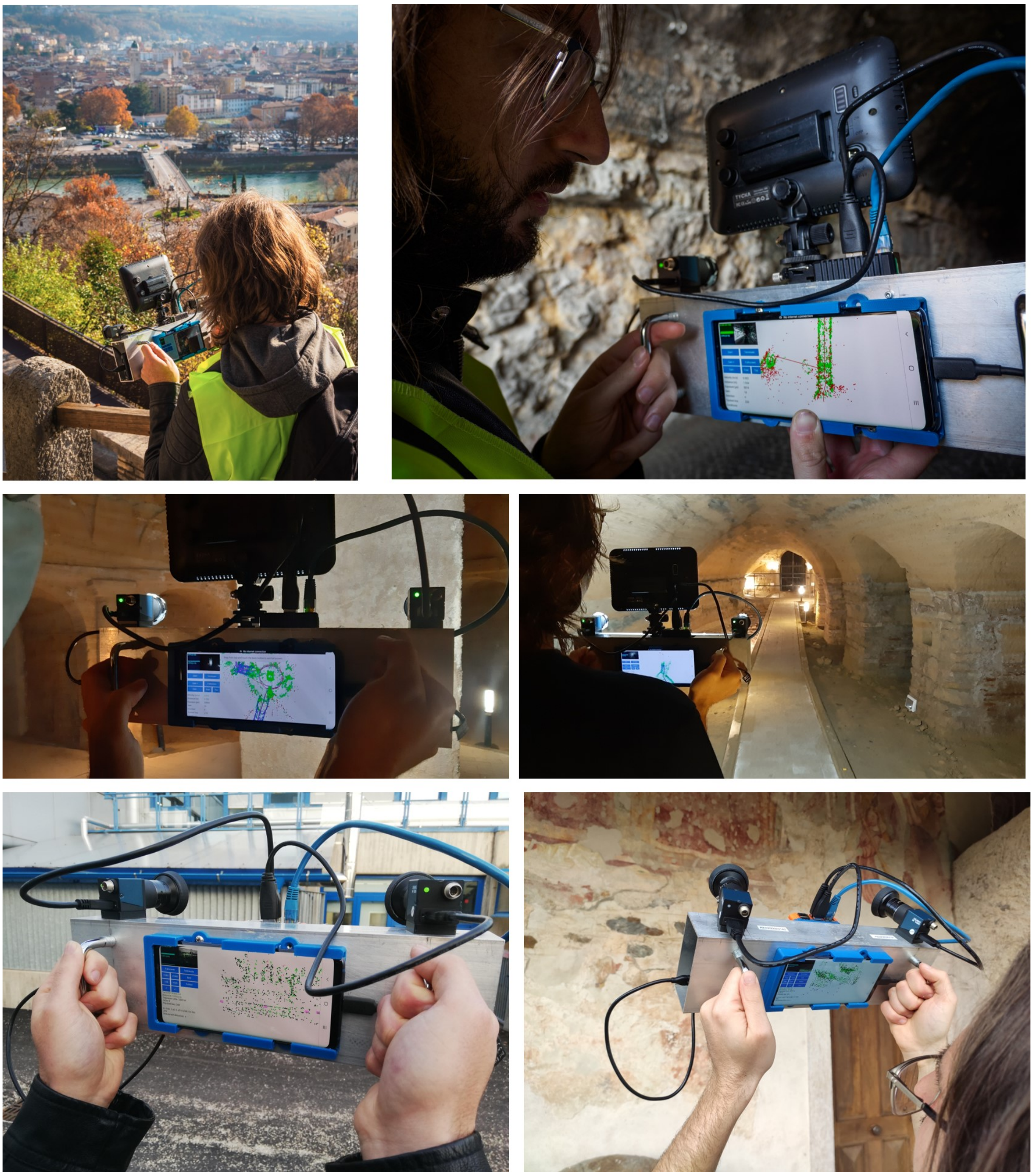


Figure 1.2: Some views of the proposed system during field operations.

was presented and evaluated in [15]. A second and improved iteration of the system was considered in [16], where different handheld mobile mapping solutions were tested and compared in an underground built heritage context. In Figure 1.2 it is possible to see some views of the system during various test operations in the field.

The remainder of the thesis is structured as follows. Chapter 2 presents the state of the art in the PMMS context. Both LIDAR and visual systems are covered and discussed. Chapter 3 summarizes the main concepts and principles related to camera geometry and Visual Simultaneous Localization And Mapping (V-SLAM) algorithms that the reader might need to better follow and understand the next parts. The chapter also reports the main motivations behind the choice of OpenVSLAM and briefly describes its architecture and stereo pipeline. Chapter 4 presents in details the proposed system and methods. Chapter 5 proposes an extensive evaluation of the proposed solutions, consider also multiple scenarios with available laser scanner ground truth. Finally, Chapter 6 draws the conclusions and hints at possible future research directions.

Chapter 2

State of the Art

In this chapter the current state of the art in the portable mobile mapping context is presented and discussed. First, a brief overview of portable mobile mapping systems is given in Section 2.1. Then, Section 2.2 covers the solutions based on LIDAR technology, using either laser scanners (Section 2.2.1) or scanner-less sensors (Section 2.2.2). Camera-based devices are finally presented in Section 2.3, where they are conveniently divided between systems based on existing platforms like smartphones or tablets (Section 2.3.1) and systems based on dedicated hardware (Section 2.3.2).

2.1 Overview

A portable mobile mapping system can be defined as one that possesses the following characteristics: (i) during the use, it is carried by a person; (ii) it is equipped with sensors that can enable three-dimensional measurements (iii) the sensor data is acquired automatically without requiring stop-and-acquire actions. Existing solutions can be divided in two main categories:

- **LIDAR-based systems**: they leverage LIDAR sensors to obtain the 3D measurements of the environment. LIDAR is a ranging technique that use collimated laser beams and photodetectors to measure the distance between the instrument and the target hit by the laser pulse. Depending on how the laser beams are emitted, existing sensors can be classified as scanner or scanner-less [17]. The former, generally known as laser scanners, employ rotating mirrors to diverge the laser beams in multiple directions, and their working range can extend up to several hundred of meters. The latter, also known as ToF (Time of Flight) cameras, can illuminate a large portion of the scene without the need of rotating mechanisms, but the working range is usually limited to a few meters. Finally, the acquired laser measurements are typically aligned and merged into a consistent 3D reconstruction using Simultaneous Localization And Mapping (SLAM) [18, 19, 20] algorithms and/or volumetric approaches [21].

- **Camera-based systems**: they use cameras and image-based 3D reconstruction algorithms to reconstruct in 3D the scene. In this category fall systems based on existing platforms, like smartphones or tablets, as well as dedicated solutions. Generally, the former can offer greater portability, but usually lack camera quality, e.g. rolling shutter or reduced dynamic range, and the possibility to use different lenses or multiple-camera configurations. The latter are typically bulkier but can generally rely on better cameras and optics, as well as on the ability to use multiple camera configurations to increase the covered area and reduce the acquisition times. Rarely, in current portable visual systems, the 3D reconstruction is performed during the acquisition of the images. The computational capability of portable hardware is still largely insufficient to allow for highly accurate and dense reconstructions of large environments. Nevertheless, there has recently been a surge in interest towards hybrid approaches, where the final 3D reconstruction is still done in post-acquisition, leveraging powerful workstation and robust algorithms known as Structure from Motion (SfM) [9] and Multi View

Stereo (MVS) [10], but the field operations are supported with real-time and low-resolution reconstructions. The system proposed in this thesis belongs to this novel category of solutions.

2.2 LIDAR-based systems

2.2.1 Scanner

The Akhka-Backpack from the Finnish Geodetic Institute [22] and the handheld Zebedee from the CSIRO ICT centre of Brisbane [23] are among the first portable systems based on laser scanners. The Akhka-Backpack is essentially a portable adaptation of a vehicle-based system [24], from which it inherited the same GNSS/INS positioning approach for direct geo-referencing of the laser scans. Hence, its use was practically limited to outdoor scenarios. The Zebedee was one of the first portable systems to use an online SLAM-based alignment [25] of the laser scans, and it proposes a peculiar combination of a 2D laser scanner with spring mechanism, that, by swinging, extends the field of view of the scanner. The Zebedee was later commercialized by the company GeoSLAM [26] under the name of ZEB1. In the successive years, new versions have been proposed with improvements in the SLAM implementation, rotating mechanism (ZEB Go [27]) and laser technology (ZEB Horizon [28]). In the commercial domain, another relevant system is the Kaarta Stencil 2 [29]. The employed SLAM algorithm combines LIDAR, inertial and visual data and it is based on the following research work [30]. Recently, the same company released a new system called Countour [31], adding an integrated display and an additional camera to colorize the point cloud. Other important commercial players in the portable field are Leica and Gexcel, with products like the Leica BLK2GO [32], the Leica Pegasus backpack [33] or the GEXCEL Heron systems [34]. Moving back to the academic works, it is worth mentioning also the systems of Nüchter et al. [35]

and Blaser et al. (BIMAGE backpack [36]). The former combines a 2D laser scanner for planar (3 Dof) motion estimation with a high-quality 3D laser scanner for the actual 3D mapping and full (6 Dof) motion estimation. The aligned laser scans and the system trajectory can be visualized in real-time on a connected laptop. The BIMAGE backpack is composed of two Velodyne VLP-16 laser scanners, an industrial-grade IMU and a Ladybug panorama camera. The system exploits the Robot Operating System (ROS) [37] to manage the data acquisition, and the Cartographer SLAM [38] to compute online the system trajectory. To achieve more accurate trajectory estimates, the images acquired by the Ladybug camera are finally used to refine the SLAM poses with SfM techniques. The interested reader may refer to these excellent reviews [1, 2] for a deeper reading on the topic.

2.2.2 Scanner-less

In the context of portable mobile mapping, scanner-less LIDAR systems are relatively new entrants. Although popular examples of scanner-less sensors like the second version of the Microsoft Kinect are around since some time, the use of TOF cameras in portable systems has just lately increased. In the latest years, companies like Microsoft and Apple started to equip some of their devices with scanner-less LIDAR sensors. These includes the latest pro versions of the iPhone or iPad as well as the second version of the HoloLens mixed reality headset. Since then, different real-time 3D reconstruction applications have been developed, both in the research [39] and commercial domains [40, 41, 42]. These works typically combines visual-inertial pose estimation algorithms with techniques to fuse interactively multiple depth images into volumetric representations, often relying also on augmented reality to display in real-time the reconstructed model to the user. Although large-scale applicability remains prohibitive, mainly due to the high computational cost of volumetric approaches and the reduced working range of

Year	**Device**	**Format**	**Sensor type**	**Domain**
2012	Akhka-Backpack [22]	Backpack	Scanner	Research
2012	Zebedee [23]	Handheld	Scanner	Research
2013	GeoSLAM Zeb1	Handheld	Scanner	Commercial
2015	Nüchter et al. [35]	Backpack	Scanner	Reserach
2017	GEXCEL HERON Lite [34]	Handheld	Scanner	Commercial
2018	BIMAGE backpack [36]	Backpack	Scanner	Research
2018	GeoSLAM Zeb Horizon [28]	Handheld	Scanner	Commercial
2018	Kaarta Stencil 2 [29]	Handheld	Scanner	Commercial
2019	Karam et al. [46]	Backpack	Scanner	Research
2019	Kaarta Contour [31]	Handheld	Scanner	Commercial
2019	Leica BLK2GO [32]	Handheld	Scanner	Commercial
2019	Microsoft HoloLens 2 [47]	Headset	Scanner-less	Commercial
2020	Apple iPhone/iPad pro	Handheld	Scanner-less	Commercial

Table 2.1: List of relevant academic and commercial portable LIDAR-based mobile mapping systems.

ToF sensors, these solutions are already able to deliver quite impressive and promising results [43, 44, 45], also considering the compact size of the devices and the real-time reconstruction process.

2.3 Camera-based systems

2.3.1 Existing platforms

Schöps et al. [48] proposed a 3D reconstruction system based a Project Tango Development Kit tablet. Leveraging GPU acceleration (Nvidia Tegra K1 chipset), a visual-inertial odometry algorithm, and a TSDF volumetric approach, they were able to compute in real-time dense 3D models of large-scale outdoor scenes from the tablet monochrome fisheye camera. Nocerino et al. [49] proposed a collaborative cloud-based solution where different users

Year	Device	Type	Camera conf.	Domain
2017	Schöps et al. [48]	Tablet	Monocular	Research
2017	Holdener et al. [52]	Dedicated	Five cameras	Research
2017	Nocerino et al. [49]	Smartphone	Monocular	Research
2018	Nawaf et al. [53]	Dedicated	Stereo	Research
2019	Hasler et al. [50]	Smartphone	Monocular	Research
2020	Ortiz-Coder and Sánchez-Ríos [54]	Dedicated	Two cameras	Research
2020	Pix4DCatch [51]	Smartphone	Monocular	Commercial
2021	Mokroš et al. [45]	Dedicated	Four cameras	Research
2022	Perfetti and Fassi [55]	Dedicated	Five cameras	Research

Table 2.2: List of relevant academic and commercial portable camera mobile mapping 3D systems.

can use their smartphone to scan the environment/object and automatically upload the images to a server. Here, incremental sparse 3D reconstructions are estimated and returned, close to real-time, to the users' smartphones. Hasler et al. [50] investigated the use of smartphone-based indoor mobile mapping. In particular, they exploited augmented reality frameworks to locally track the user trajectory and allow him/her to take in real-time measurements of the environment. Following the same idea, in 2020 the company Pix4D released Pix4DCatch [51] which takes advantage of mobile augmented reality frameworks to assist the user during the image acquisition. The app allows the user to see in real-time the locations of the acquired images and the sparse point cloud of the acquired scene. This is the most similar work to the system proposed in this thesis. Nevertheless, the app does not cover aspects related to ground sample distance or motion-blur issues and, being based on smartphones, the optical quality and flexibility is limited.

2.3.2 Dedicated platforms

Holdener et al. [52] presented a low-cost device for indoor mapping with a circular five-camera arrangement. The cameras mount fisheye lenses to provide a 360° coverage of the scene, and are triggered by a Raspberry Pi 3. The system proposed by Ortiz-Coder and Sánchez-Ríos [54] combines a laptop and two cameras with different resolutions. The low-resolution one is used to estimate the acquisition trajectory and select the most important frames from those acquired by the high-resolution camera. Nawaf et al. [53] proposed a handheld underwater stereo system that leverages an own developed visual odometry algorithm, running on a remote laptop, to provide approximate estimates of the covered area. Mokroš et al. [45] proposed a device for forestry mapping that combines four cameras, two looking towards the walking direction and two at the sides. The camera are triggered once every second with a TriggerBox. Perfetti and Fassi [55] proposed a handheld system composed of five fisheye cameras for the 3D mapping of environments with narrow passages.

Chapter 3

Background

This chapter lays the groundwork for a number of concepts and principles that are necessary for a better understanding of the proposed solution. First, some geometry basics regarding the modeling of the imaging process and of the camera motion are given. Then, the V-SLAM problem is introduced and the main approaches outlined. Finally, the chapter is concluded with a section dedicated to OpenVSLAM, including the motivation behind its choice and a brief description of its stereo pipeline.

Notation. In this and following sections, scalar number are denoted by italic lowercase letters (for example s), vectors by bold lowercase letters (for example $\mathbf{v}$) and matrices by bold uppercase letters (for example $\mathbf{M}$).

3.1 Geometry basics

In order to use the images acquired by a camera, we need to know how to relate them to the real world and the measurement process. This involves the definition of different mathematical models.

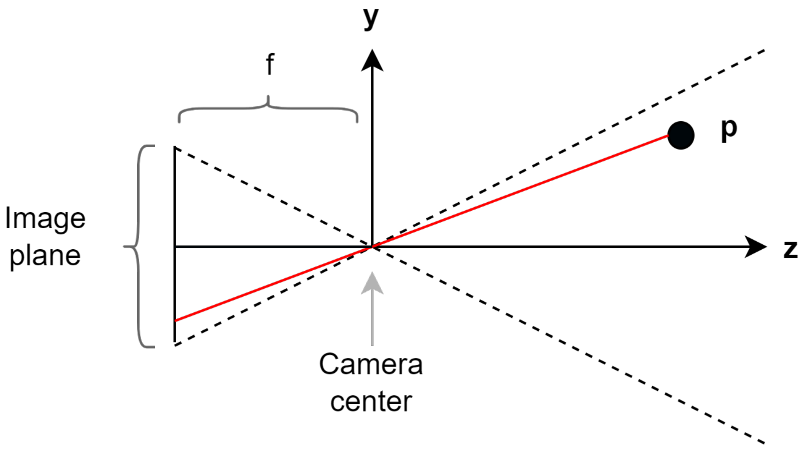


Figure 3.1: Two-dimensional illustration of the pinhole camera model. For convenience, often the image plane is placed in front of the camera center to avoid the mirroring effect.

3.1.1 Camera model

The camera model describes the relationship between a 3D point seen by the camera and its 2D projection on the image, and can be modelled as function $\pi : \mathbb{R}^3 \rightarrow \Omega$. A common and practical camera model is the Pinhole camera model, that assumes that the camera has no lenses and all the light rays pass through a single point, called the *camera center*, *optical center* or the *pinhole*. The image plane is assumed to face perpendicularly the camera center, whose relative distance is known as focal length f. An illustration of the Pinhole model is shown in Figure 3.1.

Given a 3D point in the camera coordinate system $\mathbf{p} := (x, y, z)^T \in \mathbb{R}^3$, and a unit focal length, the projection of $\mathbf{p}$ on the image plane can be computed as

$$\pi(\mathbf{p}) = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} := \frac{1}{z} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.1}$$

The obtained coordinates $(u, v)^T$ are relative to the image sensor. To compute their corresponding pixel coordinates we should apply the so called

Camera Matrix $\mathbf{K}$ which is defined as:

$$\mathbf{K} := \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

where f_x, f_y are the focal lengths expressed in pixels and c_x, c_y the coordinates of the principal point again expressed in pixel units. These parameters are also known as camera *intrinsic* parameters. The pixel coordinates of the projected point $\pi(\mathbf{p})$ can then be computed as

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\pi(\mathbf{p}) \tag{3.3}$$

The Pinhole projection does not take in account possible distortions caused by the fact the real-cameras use lenses. To account for this problem, it is common to apply a non-linear function $\tau : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ to the projected coordinates to correct possible lens distortion effects

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\tau(\pi(\mathbf{p})) \tag{3.4}$$

The *Radio-Tangential Model* [56] is a widely used polynomial model for approximating the distortion parameters of camera lenses. The OpenCV implementation, used in this thesis, has 8 parameters (6 for radial distortion $k_1, \dots, k_6$ and 2 for tangential distortion p_1 and p_2). The distortion function τ can then defined as

$$\begin{bmatrix} u_u \\ v_u \end{bmatrix} := \begin{bmatrix} u_d \, \frac{1+k_1r^2+k_2r^4+k_3r_6}{1+k_4r^2K_4r^2+k_5r^4+k_6r_6} + 2p_1u_dv_d + p_2(r^2+2u_d^2) \\ v_d \, \frac{1+k_1r^2+k_2r^4+k_3r_6}{1+k_4r^2K_4r^2+k_5r^4+k_6r_6} + p_1(r^2+2v_d^2) + 2p_2u_dv_d \end{bmatrix} \tag{3.5}$$

where (u_d, v_d) are the measured distorted point coordinates, and (u_u, v_u) are the undistorted point coordinates after applying τ.

3.1.2 Camera poses

The *camera pose*, or camera *extrinsic* parameters, describes the position and attitude of the camera when an image was taken. Assuming the existence of a world coordinate system W, the pose of a camera can be represented in two ways: either as a transformation from W to the camera coordinate system C, or its inverse. In both cases, the transformation is a rigid 3D transformation involving a translation (3 degree of freedom) and a rotation (3 degree of freedom) component, and it is an element of the special Euclidean group $SE(3)$. This kind of transformations are usually expressed as 4 x 4 matrices. The left 3 x 3 part of the matrix is the rotation matrix ($\mathbf{R}$) and the right 3 x 1 column vector is the translation vector ($\mathbf{t}$). The rotation matrices are orthogonal, so the inverse and the transpose operations return the same result ($\mathbf{R}^{-1} = \mathbf{R}^T$). Given a point $\mathbf{p}$ and a transformation $\mathbf{T}$, the new point coordinates $\mathbf{p}^n$ are computed as follows

$$\mathbf{p}^n = \mathbf{T} \cdot \mathbf{p} := \mathbf{R}\mathbf{p} + \mathbf{t} \tag{3.6}$$

The inverse transformation $\mathbf{T}^{-1} := (\mathbf{R}^T, -\mathbf{R}^T\mathbf{t})$ brings back $\mathbf{p}^n$ to $\mathbf{p}$

$$\mathbf{p} = \mathbf{T}^{-1} \cdot \mathbf{p}^n := \mathbf{R}^T\mathbf{p}^n - \mathbf{R}^T\mathbf{t} \tag{3.7}$$

A graphical representation of the transformation between a world and camera coordinate systems is shown in Figure 3.2.

3.1.3 Stereo rectification

The estimation of the three-dimensional position of the same scene point observed in two different images is known as *triangulation*. In the case of a stereo system, this process involves first the identification of corresponding pixels in the left and right images, and then, for each associated pair of pixels, the computation of the respective three-dimensional point. This problem

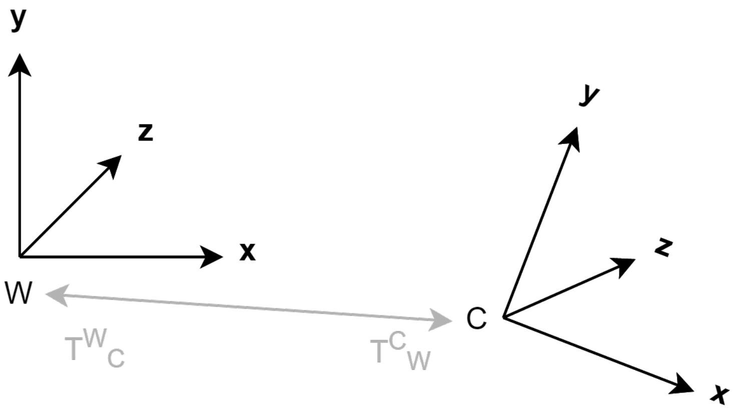


Figure 3.2: Illustration of $SE(3)$ transformations between a world coordinate system (W) and a camera (C) coordinate system. $\mathbf{T}_C^W$ transform points expressed in C to points expressed in W. $\mathbf{T}_W^C$ transform points expressed in W to points expressed in C. Note that these transformations do no change the locations of the points, just how they are expressed.

becomes significantly easier when the cameras are coplanar, that is when the right camera has only an horizontal offset with respect to the left camera or, equivalently, when the cameras' optical axes are parallel to each other and orthogonal to the baseline of the stereo camera. In practice, this property rarely holds, so a procedure known as *stereo rectification* [57] is commonly used to remap, through projective transformations, the left and right images as if they were acquired by two perfectly aligned cameras with the same intrinsic parameters (Figure 3.3). During this process, known the distortion parameters, the lens distortion can be removed from the images as well. Once the images are correctly rectified, corresponding stereo pixels will have a displacement only along the x axis, as all the epipolar lines are parallel to the stereo baseline. Given two corresponding stereo pixels $\mathbf{p}_l := (x_l, y_l)$ and $\mathbf{p}_r := (x_r, y_r)$, with $y_l = y_r$, their associated 3D point $\mathbf{p} = (x, y, z)$ can then be simply computed as

$$z = \frac{f\,b}{x_l - x_r} \qquad y = y_l\,\frac{z}{f} \qquad x = x_l\,\frac{z}{f} \tag{3.8}$$

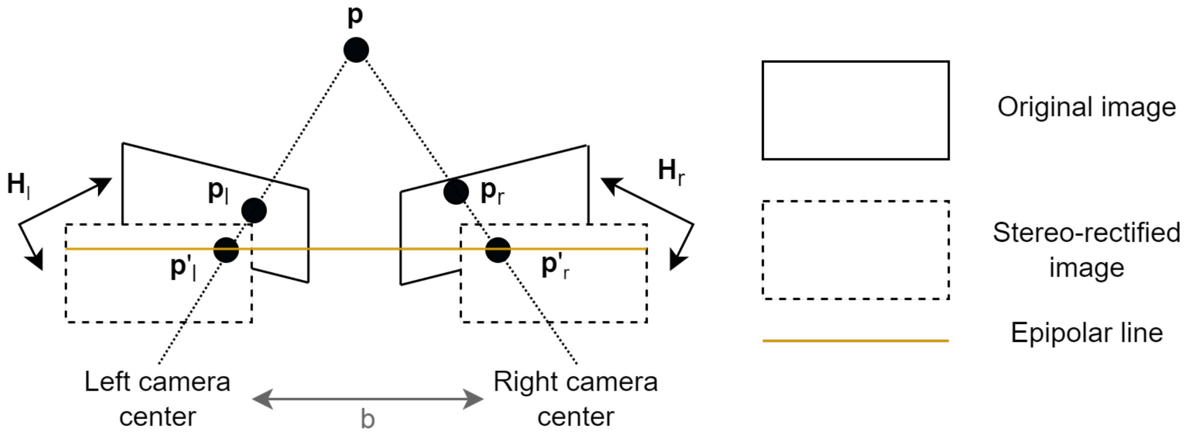


Figure 3.3: Epipolar geometry and stereo rectification.

where b is the baseline between the left and right cameras and f the focal length of the cameras.

3.2 V-SLAM

The problem of estimating in real-time the 3D structure of an unknown environment (*Mapping*) and the trajectory of one or more moving cameras (*Localization* or *Tracking*) is known today as Visual Simultaneous And Mapping (V-SLAM). Born initially in the robotic community, it rapidly spread also to other contexts, like real-time 3D reconstructions, augmented reality applications and autonomous driving. What makes V-SLAM so attractive is its very basic hardware requirements, as a simple camera can be the only required sensor. In this section we briefly describe the main existing approaches, motivate the choice of using OpenVSLAM in the proposed system, and outline the main steps of the OpenVSLAM pipeline. We refer the reader to these reviews [20, 58] for a deeper and more complete description of the topic.

3.2.1 Main approaches

Existing V-SLAM algorithm (Table 3.1) can be categorized in the following three groups:

- **Indirect**: indirect methods convert the images into a sparse set of distinctive image locations known as image features, and use them to perform the tracking and the mapping operations. They share a lot of concepts with the well-known Structure from Motion problem [9], such as feature extraction and matching, and the minimization of the geometric error (also called re-projection error) with bundle adjustment (BA) techniques. For real-time purposes, however, they usually use faster and less accurate binary features like ORB [59] or BRISK [60], and perform the geometric optimizations in local windows of selected images (*keyframes*). Relevant indirect works include PTAM [61], the three published versions of ORB-SLAM [62, 63, 64], and OPEN-V-SLAM [13].

- **Direct**: direct methods do not require image features to perform tracking and mapping operations. Instead of working on geometric errors, direct methods use formulations based on photometric errors which are computed directly on the pixels intensities. In this category fall important implementations like DTAM [65], LSD-SLAM [66] and DSO [67]. The former considers all the available pixels in the image, all the available pixels of the image while the latter restrict the tracking and mapping operations on selected pixels having high intensity gradient.

- **Semi-direct**: semi-direct methods combine together indirect and direct elements. SVO [68], for example, perform tracking using a fast direct approach based on image intensities, while features are extracted only for keyframes to initialize and expand the sparse 3D map.

In the above-argumentation, we slightly abused, for simplicity, the termi-

Year	Algorithm	Class	Loop closure	Re-localization
2007	MonoSLAM [69]	Indirect	No	No
2007	PTAM [61]	Indirect	No	No
2011	DTAM [65]	Direct	No	No
2014	LSD-SLAM [66]	Direct	Yes	Yes
2014	SVO [68]	Semi-direct	No	No
2015	ORB-SLAM [62]	Indirect	Yes	Yes
2017	DSO [67]	Direct	No	No
2017	ORB-SLAM2 [63]	Indirect	Yes	Yes
2018	LDSO [70]	Direct	Yes	No
2019	OPEN-V-SLAM [13]	Indirect	Yes	Yes
2020	KIMERA [71]	Indirect	Yes	No
2021	ORB-SLAM3 [64]	Indirect	Yes	Yes

Table 3.1: List of relevant academic V-SLAM and VO algorithms.

nology. We did not make a distinction between V-SLAM and visual odometry (VO) algorithms. The primary distinction is that the former perform additional operations known as *loop closure* and *re-localization*. Loop closure is a technique for detecting previously-visited locations, while re-localization is a recovery procedure than happens when the tracking fails. While these operations lead to increased computational loads, and hence are sometimes emitted, they bring important advantages. Loop closure can significantly reduce tracking errors, especially in long trajectories that start and end from the same position. After some track failures, for example due to abrupt motions or fast illumination changes, re-localization can restore the tracking, which would otherwise be unrecoverable. As a summary, we collected in Table 3.1 some of the most important V-SLAM and VO works.

3.2.2 OpenVSLAM

3.2.2.1 Motivation

At the time when the development the system began, the choice of using OpenVSLAM was made for a variety of reasons. First and foremost, OpenVSLAM is based on ORB-SLAM2, a complete V-SLAM algorithm with loop closure and re-localization capabilities that is still one of the best performing algorithms available [64]. Besides, OpenVSLAM includes a number of additional features that were judged important for the specific use case of the proposed system. In addition to the standard pinhole camera model for conventional lenses, OpenVSLAM supports also a wide-angle camera model [72] for fisheye lenses. This gives the system the flexibility to work both with rectilinear and fisheye lenses. Secondly, OpenVSLAM includes a web-based viewer and a data exchange system based on the WebSocket [73] protocol. The visualization of the V-SLAM output can be easily done in this way on a separate device and, as better explained in the next chapter, this feature is leveraged to better distribute the computation load inside the system.

3.2.2.2 Stereo pipeline

In this section we briefly describe the stereo pipeline of OpenVSLAM. A V-SLAM method is a complex combination of different algorithms and coding architectures. Here, we aim to outline the essential elements.

- **System calibration**. The geometric properties of the cameras (Section 3.1.1), i.e. the intrinsic and distortion parameters, and the relative pose between the left and right cameras (Section 3.1.2) must be known in advance. These parameters can be accurately estimated using calibration fields with known sizes and self-calibration bundle adjustment techniques [74, 75]. A more detailed description of the calibration procedure is given in Section A.3.

- **Stereo rectification**. Before being processed, the stereo images must be stereo-rectified (Section 3.1.3). In this way, the lens distortion is removed, and both the matching of the image features, which are ORB features in the case of OpenVSLAM, and their stereo triangulation are significantly simplified and faster to compute. The stereo-rectification requires an accurate system calibration to produce satisfactory results.

- **Initialization**. Initialization sets the world coordinate system and creates the first set of triangulated features of the map. The coordinate system is set equal to the pose of the left camera. The right camera is assumed rigid and simply translated with respect to the left one according to the given calibration and stereo-rectification information. If enough stereo matches can be successfully triangulated, the initialization phase is concluded and the algorithm is ready to start performing tracking and mapping operations and the subsequent images.

- **Tracking**. The process of estimating the current pose of the camera against the current map is called tracking. First, a set of matches between the triangulated features, also known as landmarks in the SLAM terminology, and the image features are obtained. Then, starting from an initial approximation, the pose is computed using a simplified BA formulation where the landmarks are kept fixed. Mur-Artal et al. [62] named this technique motion-only BA. If the optimization converges and a valid pose if found, the algorithm moves to the keyframe selection phase, otherwise the re-localization module is called into play. As optimization framework, OpenVSLAM uses the open-source G2O library [76].

- **Keyframe selection**. The keyframe selection decides which images (or frames) should be passed to the mapping module to update and expand the map (or sparse point cloud). The mapping operations are computationally expensive so they are limited on selected frames, the so-called keyframes. In OpenVSLAM, the selection essentially considers the number of matches between the current stereo frame and the visible map. If this number falls below a certain threshold, a new keyframe is selected and passed to the mapping module.

- **Mapping**. The mapping module runs on separate thread and its role is to keep the map updated and optimized, so that the tracking can accurately localize the camera. When a new keyframe is selected, new features are triangulated with the neighbouring keyframes, which are obtained using a graph structure called co-visibility graph, and the existing ones are jointly optimized together with the camera poses using local BA iterations. When a keyframe was processed, it is then passed to the loop closure module to check eventual loop closure events.

- **Loop closure**. The loop closure module runs on a third separate thread and basically performs three steps: (i) candidate retrieval; (ii) geometric verification; (iii) global correction. The first step retrieves the most similar keyframes to the considered one. The similarity is computed using visual bag of words [77] representations of the keyframes, and OpenVSLAM leverages the open-source DBow2 [78] library for this task. A similarity transform, which incorporates the scale and so has seven degrees of freedom, is then computed for each candidate keyframe using a RANSAC scheme and the Horn method [79]. If enough inliers are detected, the loop candidate is accepted and the third step begins. The global correction phase finally update the connections in the co-visibility graph and performs a global optimization on the whole trajectory using similarities transforms [80].

- **Re-localization**. Similarly to the loop closure case, the re-localization procedure is divided into two steps: (i) candidate retrieval and (ii) geometric verification. The candidate retrieval leverages again visual bag of words vectors to return the most similar keyframes, which are then used, in the geometric verification step, to try to compute a valid camera pose with RANSAC and the EPnP [81] algorithm. If a pose with enough matches is found, the re-localization terminates and the tracking can be normally resumed.

We refer the reader to the OpenVSLAM [13] and ORB-SLAM2 [63] papers for a more detailed description of the pipeline.

Chapter 4

Proposed Solution

This chapter presents the proposed solution. The chapter begins with a brief and high-level overview of the system (Section 4.1), which is followed by a detailed description of its hardware (Section 4.2) and software (Section 4.3) components.

4.1 Overview

A high-level overview of the proposed system is shown in Figure 4.1. The device is composed of an imaging part, a computing unit and a visualization unit. The imaging part (Section 4.2.1) is controlled by the computing unit (Section 4.2.2), to which it provides synchronized stereo pairs at every received triggering command. The stereo images are first pre-processed (Section 4.3.1) and then used by the computing unit to continuously estimate in real-time the pose of the system and a three-dimensional reconstruction of the environment in the form of a sparse point cloud (Section 4.3.2). These estimates are then exploited by four different modules to control the saving of the images (Section 4.3.3), provide feedback on the acquisition distance (Section 4.3.4) and speed (Section 4.3.5), and update the camera acquisition parameters (Section 4.3.6). The sparse three-dimensional reconstruction, the poses of the saved images and the quality control feedback are instantly and incre-

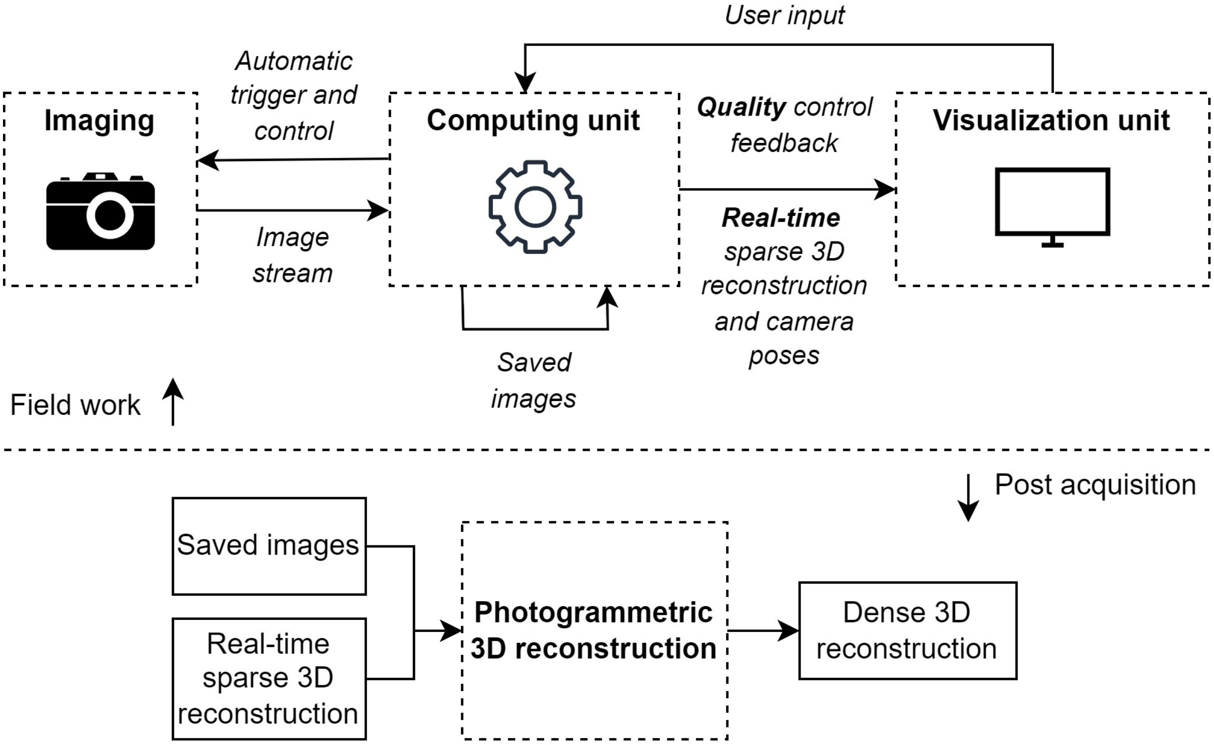


Figure 4.1: High-level overview of the proposed system.

mentally made available to the user through the visualization unit (Section 4.2.3) and a custom viewer (Section 4.3.7). After the acquisition, the saved images are finally processed with SfM and MVS pipelines commonly available today in many photogrammetric software applications. In this phase, the already-computed real-time sparse reconstruction is leveraged to simplify and speed up the computations and assign a prior to the initial poses of the images.

4.2 Hardware

The hardware was chosen looking for the right balance between quality and flexibility of the cameras, and overall weight and cost of the device. The main components of the system are rigidly attached to an empty aluminium bar and consist of two cameras, a microcomputer, a smartphone and powerbank.

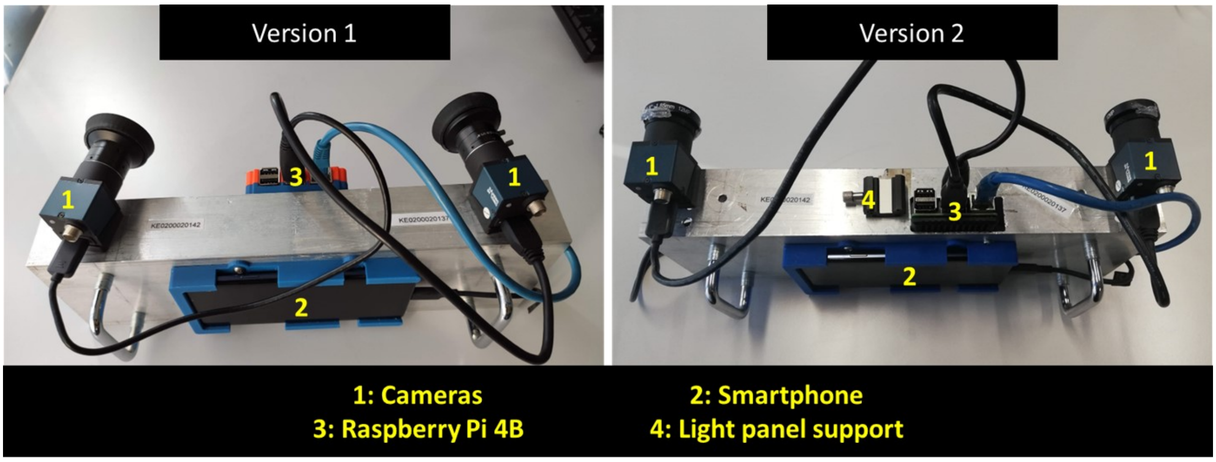


Figure 4.2: Pictures of GuPho showing two different versions and camera configurations. On the left the first version of the system, with rectilinear lenses and convergent cameras; on the right the newer version of the system, with fisheye lenses and a parallel-axes camera configuration. In the newer version we moved the Raspberry inside the case to avoid being imaged when using fisheye lenses. We also added a physical support for a LED light panel to illuminate the scene.

The microcomputer is connected to the cameras with USB3 cables, and to the smartphone with an Ethernet cable. The 5V/3A 10400 mAh powerbank is placed inside the bar and guarantees approximately two hours of working activity. Optionally, a LED light panel can be added in case of poorly illuminated environments. The following sections discuss and motivate the chosen components, while Figure 4.2 shows some pictures of the system. The overall hardware cost of the system is around 1000 Euros and the weight, without light panel, is 1.4 kilograms.

4.2.1 Imaging

The imaging system is composed of two global shutter color cameras placed in stereo configuration, whose acquisition is synchronized by software triggers (Section 4.3.1). This choice is motivated by several reasons. First, since the system is meant to be used in motion, rolling shutter cameras are in-convenient because the sensor array is exposed at different times and this may introduce significant geometric distortions in the images [82]. While there exist methods to correct the rolling-shutter distortion [83, 84], they add additional complexity to the problem that it is preferable to avoid. Secondly, the synchronization and stereo configuration, combined with a system calibration, allow OpenVSLAM to be executed in stereo mode which, in addition of being more robust that monocular pipelines [64], permits to perform tracking and mapping operations with a metric scale. Moreover, the relative transformation between the cameras can be leveraged during the offline 3D reconstruction to impose a scale without the need of ground control points. Finally, as industrial cameras, they come with rich software development kit (SDK) that allow for a fine and detailed management of the camera acquisition parameters. The cameras currently used by the system are produced by Daheng Imaging (model MER-131-210U3C), have a resolution of 1280x1024 pixels, a pixel size of 4.8 μm, and can be configured with different lenses. The latter are usually chosen based on the working scenarios. Rectilinear lenses are more common outdoor while in indoor contexts it may favourable to use fisheye ones to maximise the view coverage [85]. In the experimental section, both rectilinear (focal length 4.0mm) and fisheye (focal length 1.85mm) lenses will be employed.

4.2.2 Computing unit

Weight and size are crucial properties for a handheld device, so the system main computing unit is based on a microcomputer. In the latest years, the computational capabilities of microcomputers like the Raspberry Pi or the Nvidia Jetson increased remarkably, offering today incredible performances for their cost, size and power consumption. Moreover, unlike smartphones or tables, they offer both an open and flexible development environment and a vast selection of ports and general input/output interfaces. Both the Raspberry Pi 4 and the Nvidia Jeston Nano seemed like good options. Both are new, and their size and weight are suitable for a handheld device. Regardless, the system currently uses a Raspberry Pi 4. Compared to the Jetson Nano, it supports more RAM (8 GB) and has a faster and newer CPU. The Jetson Nano's actual benefit is its CUDA-compatible GPU, which is also substantially more powerful than the Raspberry's. However, since the system software pipeline does not use GPU-accelerated tasks, the Raspberry appeared like a better choice overall.

4.2.3 Visualization unit

The purpose of the visualisation unit is to allow the user to send control commands to the system and display and interact in real-time with the sparse 3D reconstruction and quality feedback. A tiny touch display might have been connected to the Raspberry Pi as an option. However, the Raspberry Pi can only run a limited number of threads concurrently, and OpenVSLAM already consumes the majority of the available resources. Consequently, it is preferable to manage the visualization on a separate device. Smartphones and tablets are ideal for this purpose because they are compact, contain a reasonable amount of computational power, and can easily register user inputs using touch screens. The system currently utilizes a Samsung S9 plus and a web browser-based system interface (Section 4.3.7). The advance of this choice is that any device with a modern web browser can serve as the visualization unit, being it a smartphone, a tablet or a remotely-connected computer.

IMAGING				
Model	**Resolution**	**Shutter type**	**Pixel size**	
2 x Daheng Imaging MER-131-210U3C	1280x1024	Global	4.8uM	
COMPUTING DEVICE				
Model	**CPU**	**Memory**	**Disk**	
Raspberry Pi 4 model B	Cortex-A72 1.5GHz	8 GB	128 GB (SD)	
VISUALIZATION DEVICE				
Model	**CPU**	**GPU**	**Memory**	**Screen**
Samsung s9	Qualcomm 845	Adreno 630	6 GB	6.2 inches
MISCELLANEOUS				
Power bank	**Light panel**	**Total weight**		
5V/3A 10400 mAh	Variable	1.4 Kg without light panel		

Table 4.1: Hardware summary of the proposed system.

4.3 Software

From a software perspective, the system can be logically divided into three main parts (Figure 4.3). One part deals with the camera triggering, image retrieval and pre-processing. The second, and core part, performs the mapping and tracking operations as well as the proposed acquisition, distance, motion-blur and camera controls. The third and final part manages the visualization and the handling of the user commands. Hereafter, the three parts and the relative sub-modules are accurately described.

4.3.1 Triggering and pre-processing

In the current implementation of the system, the acquisition of the stereo image pair is synchronized with software triggers, although hardware triggers are possible. The maximum synchronization error between the left and

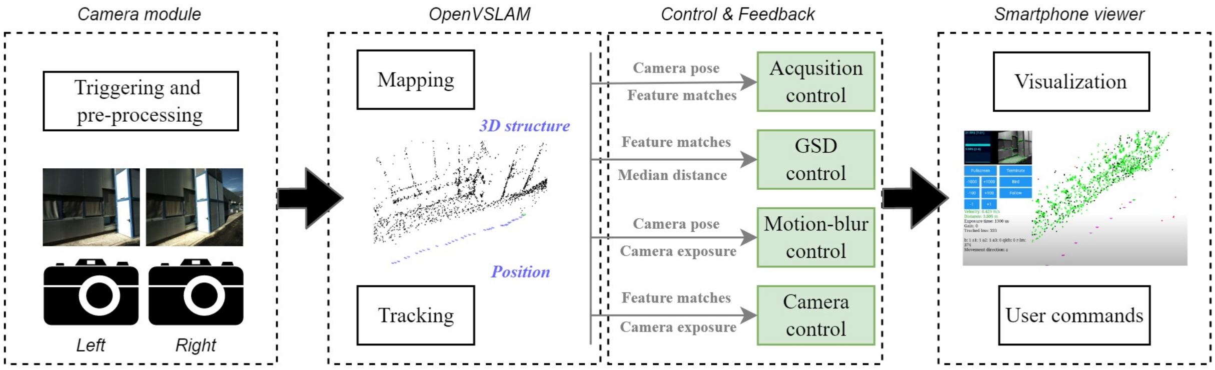
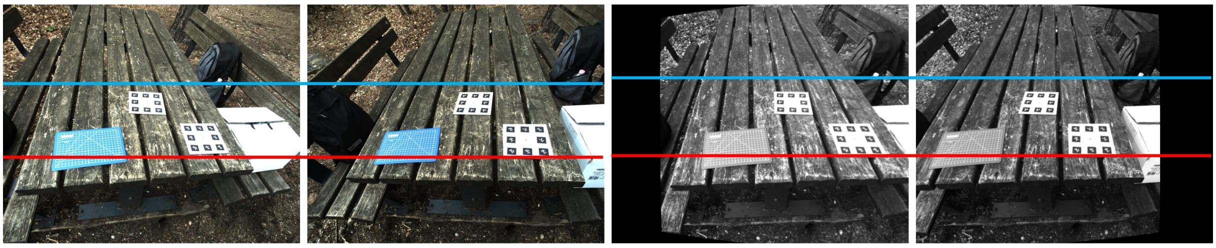


Figure 4.3: Software pipeline. See text for a detailed description.

the right image is about 1 millisecond (Appendix A.1), which is acceptable for a walking speed. At every new acquisition, the trigger order is flipped (left-right, right-left) to better distribute and thus compensate the synchronization errors. The triggering happens within the tracking thread of OpenVSLAM (Section 3.2.2.2), so the image stream frequency depends on the current tracking time. After sending the triggers, the availability of the images in a common shared buffer is checked at regular intervals, shortly sleeping the thread in between to avoid occupying unnecessary system resources. Once available, the images are then converted to single channel, down-scaled, and stereo-rectified (Figure 4.4) as part of the pre-processing procedure. Single channel images are required for the extraction of ORB features [59] used by OpenVSLAM for tracking and mapping operations. The down-scaling reduces the resolution of the images from 1280x1024 to 640x512, a value that allows the system to perform tracking at roughly 5 Hz. We believe that this a good compromise between frame rate and tracking/mapping accuracy, considering also that the device will be moved carefully by a walking person and fast and abrupt motions are unlikely to happen. Finally the images are stereo-rectified as required by OpenVSLAM to reduce the complexity of the feature matching and triangulation operations. The described pre-processing operations are done on a separate copy of the images. The



Original 1280 x 1024 stereo images

Rectified 640 x 512 stereo images

Figure 4.4: An example of input stereo images and effects of the pre-processing step. Note how in the rectified images corresponding scene elements lie on the same horizontal line.

high-resolution and colored images remain available to be eventually saved by the acquisition control (Section 4.3.3) and successively used for the final offline photogrammetric 3D reconstruction.

4.3.2 Tracking and mapping

Once that OpenVSLAM has successfully initialized the first set of sparse features (the initial map), the system starts to track the stereo camera and map the surrounding environment. For every new stereo pair, OpenVSLAM returns the poses of the left and right cameras as well as the set of features of the map that were matched during the pose estimation process. If the user returns to a previously visited location, such as when closing the acquisition loop or when re-visiting common areas, the loop closure module of OpenVSLAM might eventually detect it and perform a global optimization of the acquisition trajectory. In case the tracking is lost due to an expected event like a sudden change of light, an abrupt movement or a difficult area, the user can return to a previously visited location to perform a re-localization. We refer the reader to Section 3.2.2.2 for a more detailed description of the OpenVSLAM pipeline.

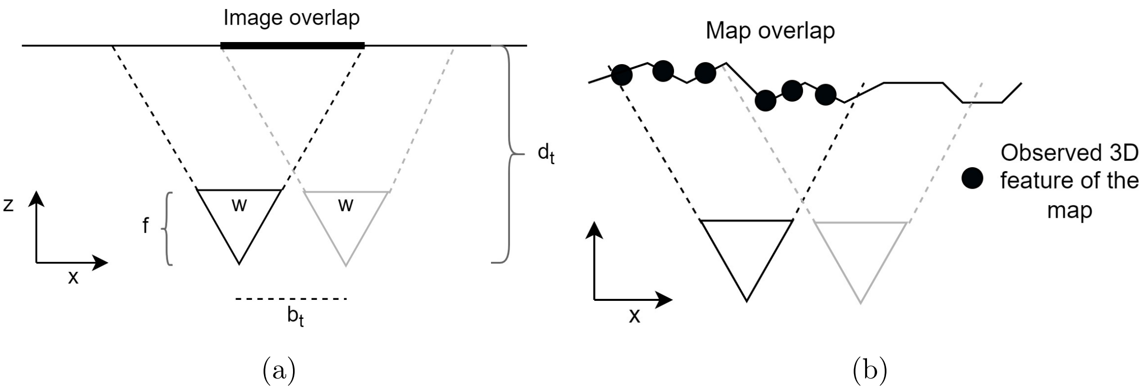


Figure 4.5: Two-dimensional and monocular representation of the two acquisition control methods based on image (a) and map (b) overlap.

4.3.3 Acquisition control

This module controls the saving of the high-resolution and colored images on the SD card. In the literature, it is common to do this operation using fixed time intervals, e.g. saving the images with regular frequencies like 1 Hz. The problem is that such approaches do not consider the scene, nor the relative position of the camera. The overlap among subsequent images may be excessive and poorly optimized or, worse, not sufficient to ensure a strong feature matching. While the latter case is clearly problematic, also an excessive acquisition of images should be avoided because the processing time of photogrammetric software massively increases with the number of images. Finally, another aspect to consider is that the process of saving the images is a costly operation in term of execution time, so it should be minimized in a real-time system already put under significant stress. Consequently, it is preferable to use methods that can optimize the saving of the images without however risking to compromise the image overlap. The proposed system aims to do this by leveraging the available tracking and mapping information, and supports two acquisition modes hereafter described.

4.3.3.1 Image overlap

This acquisition mode assumes that the environment is composed mainly of planar surfaces and the user is performing standard photogrammetric acquisitions keeping the cameras mostly parallel to the object surface. Under these assumptions, the following algorithm maintains constant the overlap among subsequent saved images taking advantage of the real-time estimates of the camera pose and the median depth of the scene. The control of the overlap is done considering a single moving camera, and the left camera is arbitrary chosen for this purpose. Let $\mathbf{I}_s$ be the last selected and saved image pair and $\mathbf{T}_s$ the corresponding estimated (left) pose. A new image pair $\mathbf{I}_t$ and associated (left) pose $\mathbf{T}_t$ is selected and saved if the baseline between the camera center of $\mathbf{T}_t$ and the camera center of $\mathbf{T}_s$ is bigger than a target baseline b_t, function of the image overlap along the trajectory. The value of b_t is updated in real-time according to the median depth of the environment, the internal properties of the camera, and a target image overlap value. More precisely, b_t is computed as

$$b_t = \begin{cases} w\frac{d_t}{f}(1 - O_x), & \text{if movement along camera } X \text{ axis} \\ h\frac{d_t}{f}(1 - O_y), & \text{if movement along camera } Y \text{ axis} \\ k, & \text{if movement along camera } Z \text{ axis} \end{cases} \tag{4.1}$$

where w and h are, respectively, the width and height of the camera sensor, d_t the median depth of the scene (computed taking the median depth of the matched features), f the focal length of the camera, O_x and O_y the target image overlaps in decimals along the X and Y axis, and k and constant. The different cases ensure that the same image overlap is enforced when the movement occurs along the shortest or the longest dimension of the image sensor (Figure 4.5a). The movement direction is detected in real-time from the largest direction cosine between the camera axes and the displacement vector between the camera centers of $\mathbf{T}_t$ and $\mathbf{T}_{t-1}$. The movement along the Z axis (forward, backward) is less common in these types of acquisitions, so it is managed with a simple constant baseline that can be adapted to the specific scenario. When a new image is saved, its pose is visualized in the viewer, and the actual saving operation happens on a separate thread to avoid slowing down the tracking operations.

4.3.3.2 Map overlap

The following acquisition mode does not make assumptions on the properties of the scene or on the acquisition modality. The saving of the images is controlled in this case by the amount of feature overlap between the current image pair and the estimated map of the environment (Figure 4.5b). This is essentially the same principle used by the keyframe selection procedure of OpenVSLAM (Section 3.2.2.2), so the two operations was conveniently merged. In other words, at every keyframe selection corresponds an image saving. However, a modified keyframe selection criteria is here proposed which is driven by an absolute feature overlap threshold. Given a new image pair $\mathbf{I}$ and a number n of feature matches against the sparse point cloud (or map), the original keyframe selection of OpenVSLAM considered $\mathbf{I}$ as a keyframe if $n < p$, where p is the number of feature matches against the sparse point cloud of the last selected keyframe. The proposal is substitute p with a constant value m. This has several advantages. First, it is possible to control the desired target overlap, as increasing (resp. decreasing) m will result in an increased (resp. decreased) number of saved images. Secondly, it enables a denser saving of the images when the scene presents challenging situations like homogenous or reflective surfaces as well as strong illumination variations, so that the successive dense reconstruction stage can have more data to cope with problematic textures. At the used image resolution of 640x512 and a limit of 1000 ORB features per image, a suitable value of m

was empirically determined to be between 100 and 200 point cloud matches.

4.3.4 Ground sample distance control

The ground sample distance (GSD) is the leading acquisition parameter of a photogrammetric survey [86]. It theoretically determines the size of the pixel in the object space and, consequently, its value is related to accuracy and level of resolution of the three-dimensional reconstruction [87]. Given a generic pixel $\mathbf{p}$ and a corresponding depth value d, the GSD of $\mathbf{p}$ can be computed as

$$GSD(\mathbf{p}) = \frac{s\ d}{f\ w} \tag{4.2}$$

where s is the sensor size, f the focal length of the camera, and w the number of image columns (or equivalently the width of the image in pixels). In the proposed system it is possible to configure a target GSD range $[\delta_m, \delta_M]$ and take advantage of a real-time GSD control during the acquisition of the images. Usually, the GSD is not uniform in an image because the imaged scene may present elements at different depths. Areas lying closer to the camera will have a smaller GSD than areas lying farther away. To properly manage these situations, the GSD is computed in multiple image locations leveraging the pixel coordinates of the image features having a known depth value. The latter are visualized over the live image in the system interface (Section 4.3.7), where colors are used to indicate their GSD. Red pixels have a GSD bigger than δ_M, blue pixels have GSD smaller than δ_m, while green pixels have a GSD bounded in the target range. This simple feedback can be easily visualized even on a small display, and can be exploited to check whether the current acquisition distance satisfies the target ground sample distance in the different areas of the image. Moreover, when a new image pair $\mathbf{I}$ is saved by the acquisition control (Section 4.3.3), the average GSD of map

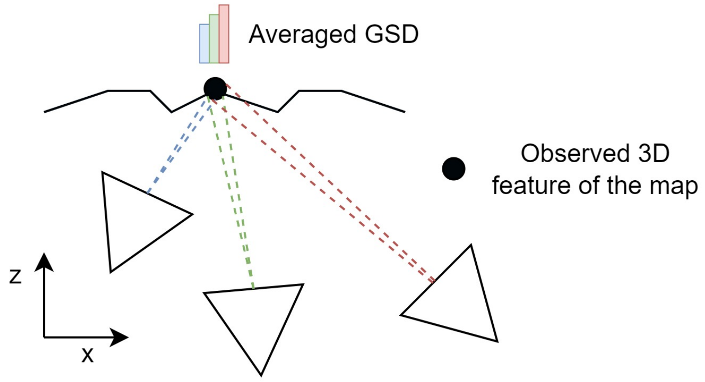


Figure 4.6: Exemplification of the GSD feedback.

features (or landmarks) observed by **I** (Figure 4.6) is updated. In this way, using the same color scheme as above, it is possible to visualize the average GSD of the overall image acquisition directly on the sparse reconstruction, and eventually detect the areas of the environment where the target GSD was not respected.

4.3.5 Motion blur control

Motion blur can significantly worsen the quality of image acquisitions performed in motion. It occurs when the camera, or the scene objects, move significantly during the exposure phase. The exposure time, i.e., the time interval during which the camera sensor is exposed to the light, is usually adjusted, either manually or automatically, during the acquisition. This is done to compensate for different lighting conditions and avoid under/over-exposed images. Consequently, also the acquisition speed should be adapted, especially when the scene is not well illuminated, and the exposure time is relatively long. Rather than detecting motion-blur with image analysis techniques [88], the proposed approach tries to prevent it by monitoring the acquisition speed and raise slow-down warnings when dangerous situations are detected. At a generic time t, the acquisition speed v_t of the cameras is computed as:

$$v_t = \frac{\|\mathbf{c}_t - \mathbf{c}_{t-1}\|^2}{t_t - t_{t-1}} \tag{4.3}$$

where $\mathbf{c}_t$ and $\mathbf{c}_{t-1}$, and t_t and t_{t-1} are respectively the left camera centers and the timestamps of the current and last image pair, and $\|\|^2$ the Euclidean norm. A speed warning is raised when

$$\Delta_x \geq max(\delta_{min}, \delta_s) \tag{4.4}$$

where Δ_x is the space travelled by the cameras at speed v_t during the known exposure time of the cameras, and δ_s is the ground sample distance of image feature with the smallest depth. To avoid considering outliers, δ_s is chosen as the 5-th percentile of all the features with a valid depth. The logic behind this control is that, to ensure that the details remain sharp, the camera should move, during the exposure time, less than the target GSD. The max function is used to avoid false warnings when the closest element in the scene is farther than the minimum GSD. A main assumption and simplification is here made that the imaging sensor is mostly parallel to the imaged surface.

4.3.6 Camera control

The literature of automatic exposure (AE) algorithms is quite rich [6, 89], and each solution has been usually conceived to fit a particular scenario. In the case of the proposed system, the image acquisition is presumably done at a known distance from the environment, according to the acquisition requirements and the specified ground sample distance settings (Section 4.3.4). Hence, it would be convenient to exploit this prior knowledge to perform an exposure control that gives higher priority to the imaged areas of the environment that are in the target acquisition range. A common strategy to automatically adjust the exposure time is to use the average gray value of

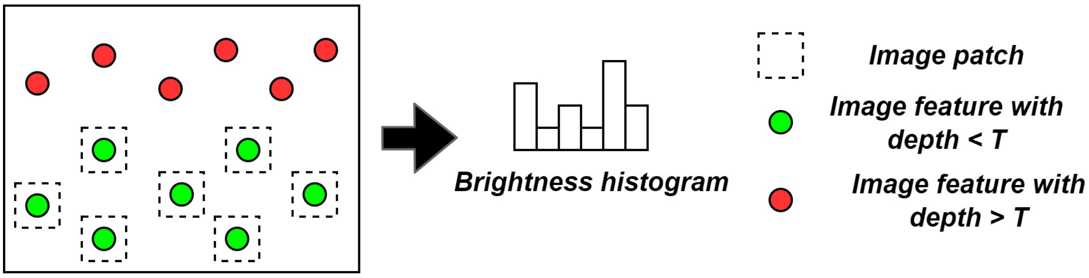


Figure 4.7: Representation of the proposed camera exposure control.

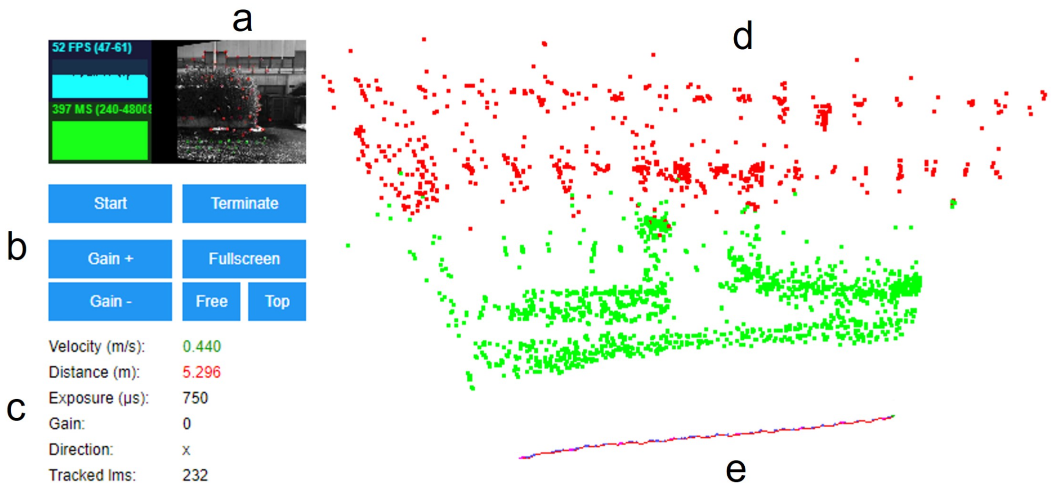
the image brightness histogram [90], lowering (resp. increasing) the exposure when the measured gray value is higher (resp. lower) than a target gray value. Instead of using the whole image, the proposed algorithm computes the brightness histogram considering some specific image regions, selected in the proximity of the location of the image features having depth values within the target acquisition range (Figure 4.7). For each valid image feature, an area of 5x5 pixels is considered around it. If not enough valid features are found, the whole image is instead considered. The camera exposure e is then adjusted according to obtained average gray value g and a target gray value h using a proportional controller

$$e = e + (h - g)\ k_p \tag{4.5}$$

where k_p is the proportional gain. The sensitivity to light of the camera sensor, also known as gain or ISO, is not taken in consideration by the presented camera control. The adjustment of the ISO is left to the user preferences, and, at that purpose, specific control buttons are present in the system interface (Section 4.3.7).

4.3.7 System interface

The interface of the system builds upon the web-viewer of OpenVSLAM, and therefore runs inside a web browser. The communication between the



Velocity (m/s): 0.440
Distance (m): 5.296
Exposure (μs): 750
Gain: 0
Direction: x
Tracked Ims: 232

Figure 4.8: Graphical interface of the system during a live execution. In this example, the target GSD range was set between three and ten millimeters.

interface, which runs on the smartphone (Section 4.2.3), and the V-SLAM algorithm, which runs on the Raspberry (Section 4.2.2), is managed by a Node.js [91] server running on the Raspberry. The data is exchanged using the WebSockets [73] and Google Protocol Buffers [92]. The developed interface is composed of several parts (Figure 4.8):

(a) The live stream of the left camera image with overlaid image features matched with the map. The latter are color-coded based on their estimated ground sample distance (Section 4.3.4). Some statistics on the tracking and rendering frame rates are given as well.

(b) The command section where the user can start and stop the tracking and mapping operations, adjust the gain of the camera sensors, and change some properties of the rendering camera of the three-dimensional viewer (see point (d)).

(c) The statistics section where are displayed live information of the acquisition speed and distance, exposure and gain settings of the cameras, movement direction of the camera (Section 4.3.3), and number of image featured matched with the map (or sparse point cloud). The acquisition distance and speed are colored in red (not ok) or green (ok) according to the ground sample distance (Section 4.3.4) and motion-blur (Section 4.3.5) controls.

(d) The live sparse reconstruction (Section 4.3.2) of the environment. The Tree.js library [93] is used for the 3D rendering of the point cloud. Each 3D point is colored with its average ground sample distance (Section 4.3.4).

(e) The current pose of the system, the overall acquisition trajectory and the camera poses of the saved images (Section 4.3.3). The camera poses are rendered with Tree.js as oriented image pyramids.

Chapter 5

Experiments

This chapter proposes several experimental evaluations of the system in different indoor and outdoor environments, considering both controlled and real-case scenarios. The experiments are organized as follows. First, the main system modules are tested individually, i.e. the acquisition control (Section 5.2.1), the ground sample distance control (Section 5.2.2), the motion-blur control (Section 5.2.2), and the camera control (Section 5.2.3). Finally, the system as a whole is evaluated in multiple real-case and complex scenarios.

5.1 Testing environments

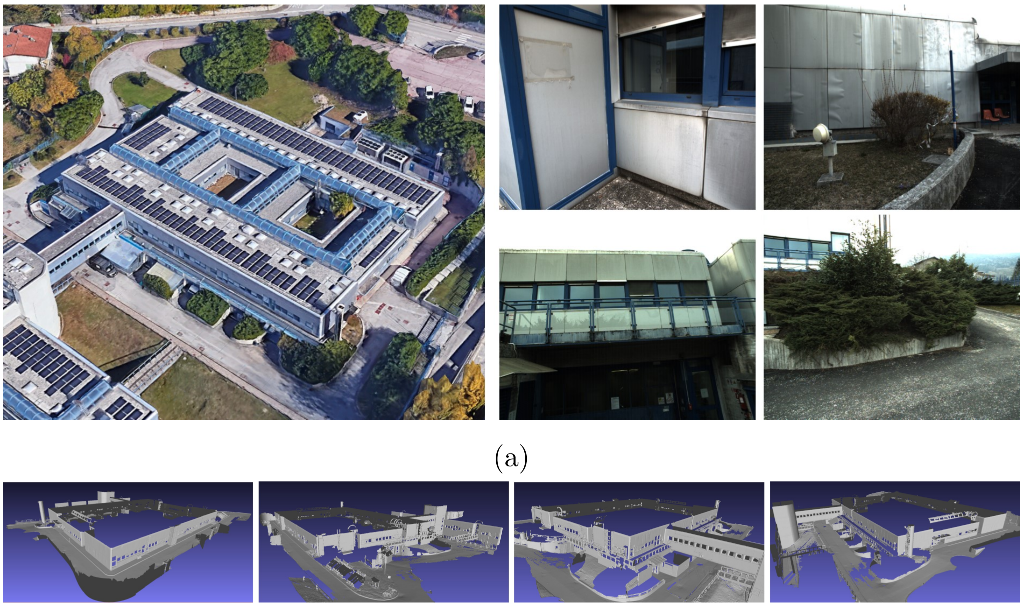
The experiments were carried out in multiple testing environments. For convenience and a better organization of the document, this section introduces the environments that were used in more than one experiment, leaving the description of the more specific cases in the respective sections.

5.1.1 FBK building

This environment represents the typical outdoor scenario where the object to reconstruct is a building. Unnamed aerial vehicles (UAV) are commonly used in these cases to acquire imagery also from above, but in our case the tests were limited to the areas reachable by a walking person. The considered building is inside the FBK headquarters of Povo (Trento - Italy) and spans approximately 40 × 60 meters (Figure 5.1a - left). The structure presents challenging elements like poorly-textured and metallic surfaces, glasses, and some vegetation (Figure 5.1a - right). A ground truth 3D reconstruction of the building is available by means of terrestrial laser scanning. A Leica HSD7000 [94] (angular accuracy 125 μrad, range noise 0.4 mm RMS at 10 m) was used to scan the building from 21 stations along the perimeters at an approximate distance of 10 meters. All the scans were manually cleaned from the vegetation, converted to a 3D model and co-registered with the global iterative closest point (ICP) algorithm offered by MeshLab [95]. The final median RMS of the residuals from the alignment transformation was about 4 mm. Some views of the ground truth model are shown in Figure 5.1b.

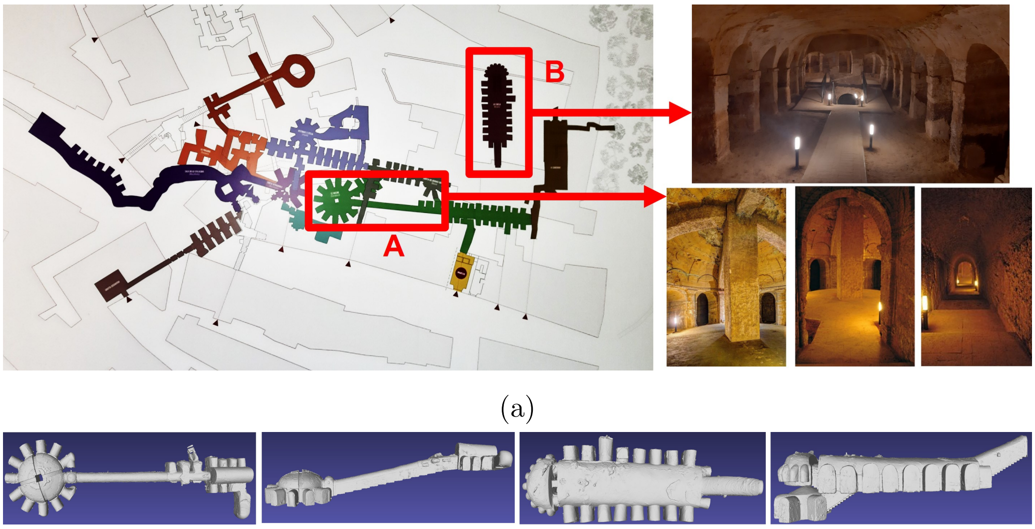
5.1.2 Camerano caves

This environment is completely different than the previous one, and represents an underground built heritage site. These are common working scenarios for portable 3D mapping systems because of their geometrical complexity and extension. Specifically, some portions of the Camerano caves in Ancona (Italy) are taken into consideration here. The caves consist of a tangled combination of halls and tunnels located below the city. For visual systems, this environment poses several challenges for the poor and uneven illumination, and the variable size of the spaces. More precisely, the proposed experiments are carried out in two areas of the site (Figure 5.2a) comprising a narrow tunnel and domed room (box "A" - span 41 × 11 meters), and a large hall with two floors characterized by many niches (box "B" - span 25 × 8 meters). Some images of the areas of interest are shown in the figure as well. All the areas were previously mapped with terrestrial laser scanning, so ground truth models (Figure 5.2b) are available for accuracy and comparison tests.


(a)

(b)

Figure 5.1: The FBK building environment. (a) Aerial (Google Earth) and close range views of the building. (b) Laser scanning ground truth mesh model.



(a)

(b)

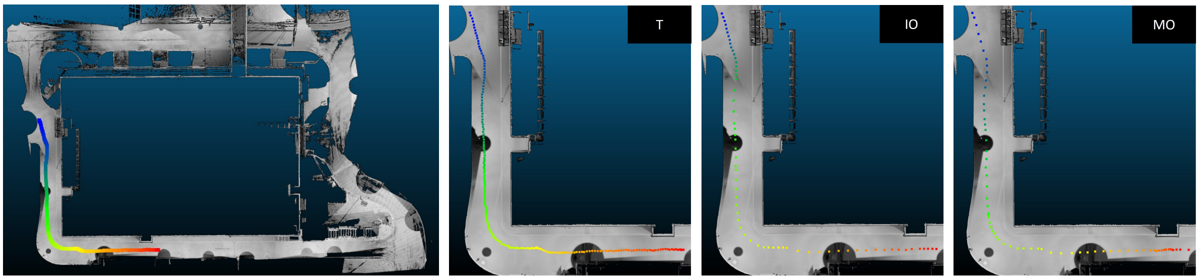
Figure 5.2: The Camerano caves environment. (a) Areas considered in the experiments and some area views. (b) Laser scanning ground truth mesh models.

5.2 Module tests

5.2.1 Acquisition control

The aim of this section is to evaluate the acquisition control module (Section 4.3.3), and, more precisely, it compares the proposed methods based on the image and map overlap against timed selections, currently very common in the literature. The system was used to record several dataset at 5Hz, on which the different image selection methods are then applied. The selected images are then processed withing a photogrammetric pipeline to obtain the dense point clouds that are evaluated against the ground truth models. The evaluation reports the number of selected images n, the 3D reconstruction time t, the average observation multiplicity γ, the ICP alignment error against the ground truth e, and the mean μ and standard deviation σ of the signed Euclidean distances between the dense point clouds and the ground truth mesh model.

Experiment A. This experiment takes place in the south-east corner of the FBK building environment (Section 5.1.1). The dataset has an approximate length of 60 meters, in which the cameras were kept mostly parallel to the building facade (Figure 5.3a). The acquisition lasted around 200 seconds so, at 5Hz, the system recorded 1000 stereo pairs. Three image selection procedures were then applied. A trivial time-based approach (T) with a selection frequency of 1Hz, which selected 200 image pairs (Figure 5.3b). The proposed image overlap selection (IO) with a target image overlap of 80%, which selected 64 image pairs (Figure 5.3c). The proposed map overlap selection (MO) with a target map overlap of 200 matches, which selected 70 image pairs (Figure 5.3d). Each selected set of image pairs was then processed with Agisoft Metashape [96] to retrieve the corresponding dense point clouds. The cameras intrinsic, distortion parameters and stereo baseline ob-



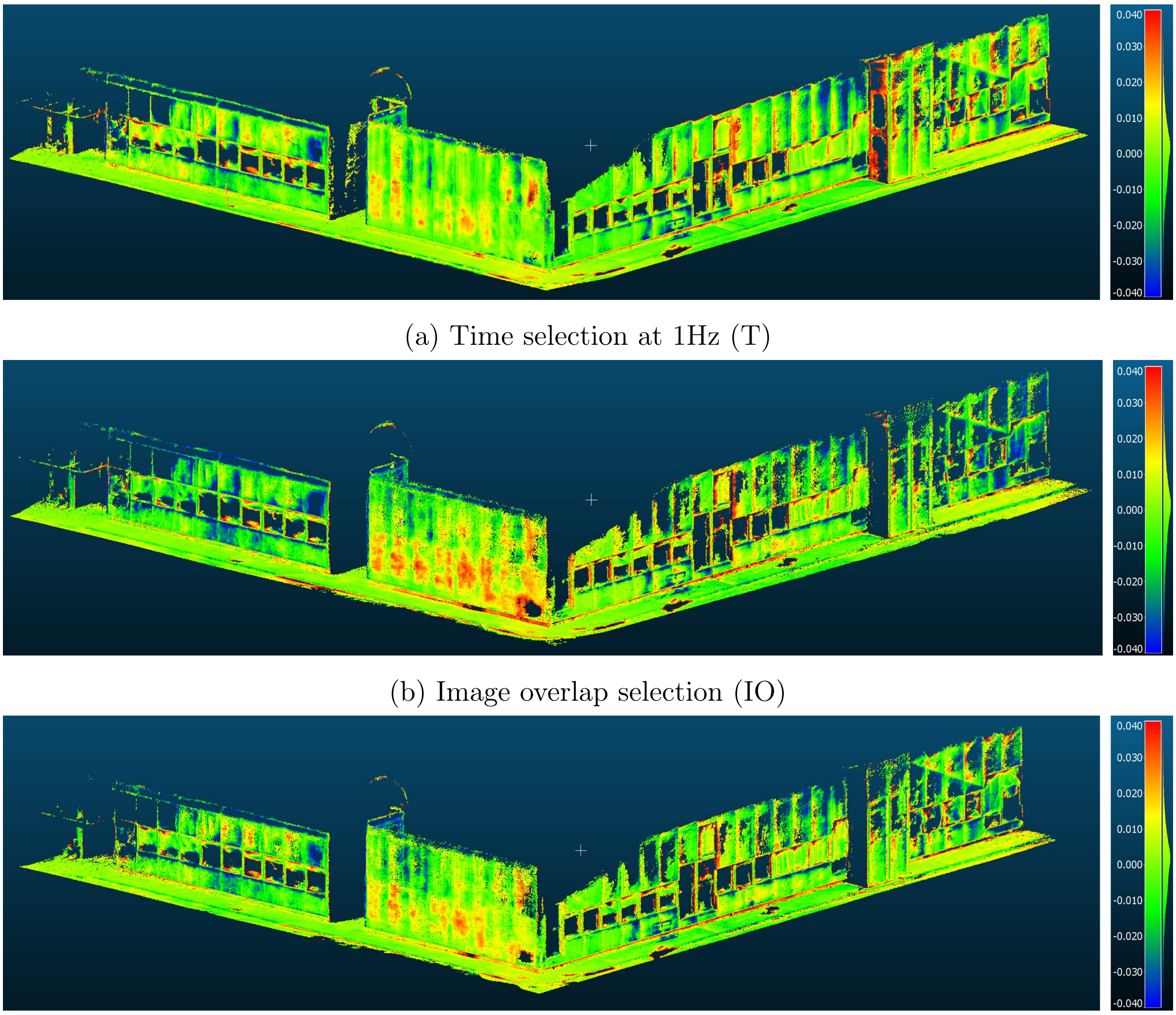
(a) (b) (c) (d)

Figure 5.3: Experiment A. (a) Top view of the dataset trajectory. Top view of the positions of the selected image pairs with the time (b), image overlap (c) and map overlap (d) methods. Colors encode image index where blue and red correspond respectively to the start and end of the trajectory.

tained from the system calibration (Table A.1) were kept fixed during the bundle adjustment. Regarding the feature matching strategy, the generic pre-selection was used in all the three cases. This strategy avoids to compute the feature matching across all the input images, which are instead pre-selected according the results of a brute force matching done at low image resolutions [97]. In the cases of the images selected by the image (IO) and map (MO) overlap methods, the generic pre-selection could have been replaced by the reference pre-selection, which instead considers a prior information of the image poses to select the matching pairs. The latter are indeed known from the trajectory estimates of OpenVSLAM. However, for the sake of comparability of the results, the generic pre-selection was used also in the IO and MO cases. The obtained dense point clouds were then cleaned from noisy estimations using the confidence scalar value provided by Metashape, and aligned with the ground truth data using the iterative closest point (ICP) implementation of CloudCompare [98]. The scale option was disabled and the algorithm was configured to remove the furthest point at every iteration. Finally, signed Euclidean distances were computed between the estimated dense point clouds and the ground truth model. Table 5.1 and Figure 5.4 summarize the obtained results.

Method	n	t (s)	γ	e (m)	μ (m)	σ (m)
T (1 Hz)	200	933	4.6052	0.0351	-0.0002	0.0142
IO	64	185	3.4899	0.0358	-0.0005	0.0151
MO	70	179	3.4727	0.0350	-0.0001	0.0141

Table 5.1: Experiment A. Summary of the results.



(a) Time selection at 1Hz (T)

(b) Image overlap selection (IO)

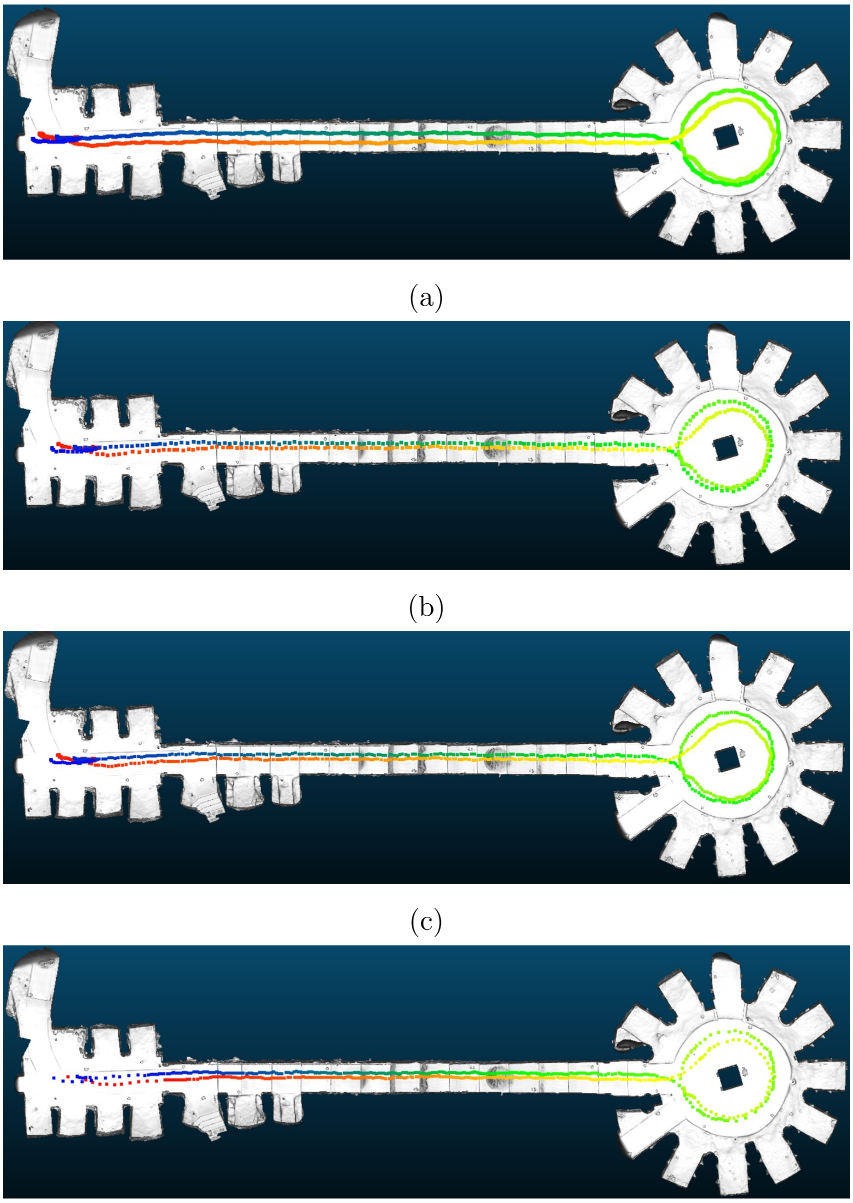
(c) Map overlap selection (MO)

Figure 5.4: Experiment A. Dense point clouds obtained with the different selection methods with color-coded signed Euclidean distances (meters) against the ground truth model.

Method	n	t (s)	γ	e (m)	μ (m)	σ (m)
T (1 Hz)	400	2487	3.7022	0.0513	0.0001	0.0457
T (1.6 Hz)	666	4946	4.2544	0.0521	0.0003	0.0510
MO	632	6029	4.4400	0.0412	-0.0059	0.0306

Table 5.2: Experiment B - Summary of the results.

Experiment B. This experiment was carried out in the part A of the Camerano caves environment (Section 5.1.2). The dataset trajectory is around 104 meters long, lasted approximately 400 seconds, and, at 5 Hz, 2000 image pairs were acquired in total. In this case, the acquisition started and ended from the same positions and traversed twice a narrow tunnel and a domed room (Figure 5.5a). A light panel was used to compensate the otherwise too dark environment. Taking advantage of the fisheye setup, the cameras pointed in the walking direction for most of the time. Consequently, the image overlap selection is not applicable here, and the evaluation will consider only the time and map overlap selections. The time selection was performed at two different frequencies: one at 1Hz, which selected 400 image pairs, and one at 1.6 Hz which selected 666 image pairs. Finally, the map overlap approach, with 200 matches overlap, selected 632 image pairs. As in the previous experiment, the selected pairs were processed with Metashape leveraging the generic pre-selection feature matching strategy, and the intrinsic, distortion, and stereo baseline parameters obtained with a previous system calibration (Table A.2). The latter were kept fixed during the processing. The obtained dense points clouds were then filtered, aligned with the ground truth, and used to compute the signed Euclidean distances against the TLS mesh model. The results are shown in Table 5.2 and Figure 5.6.



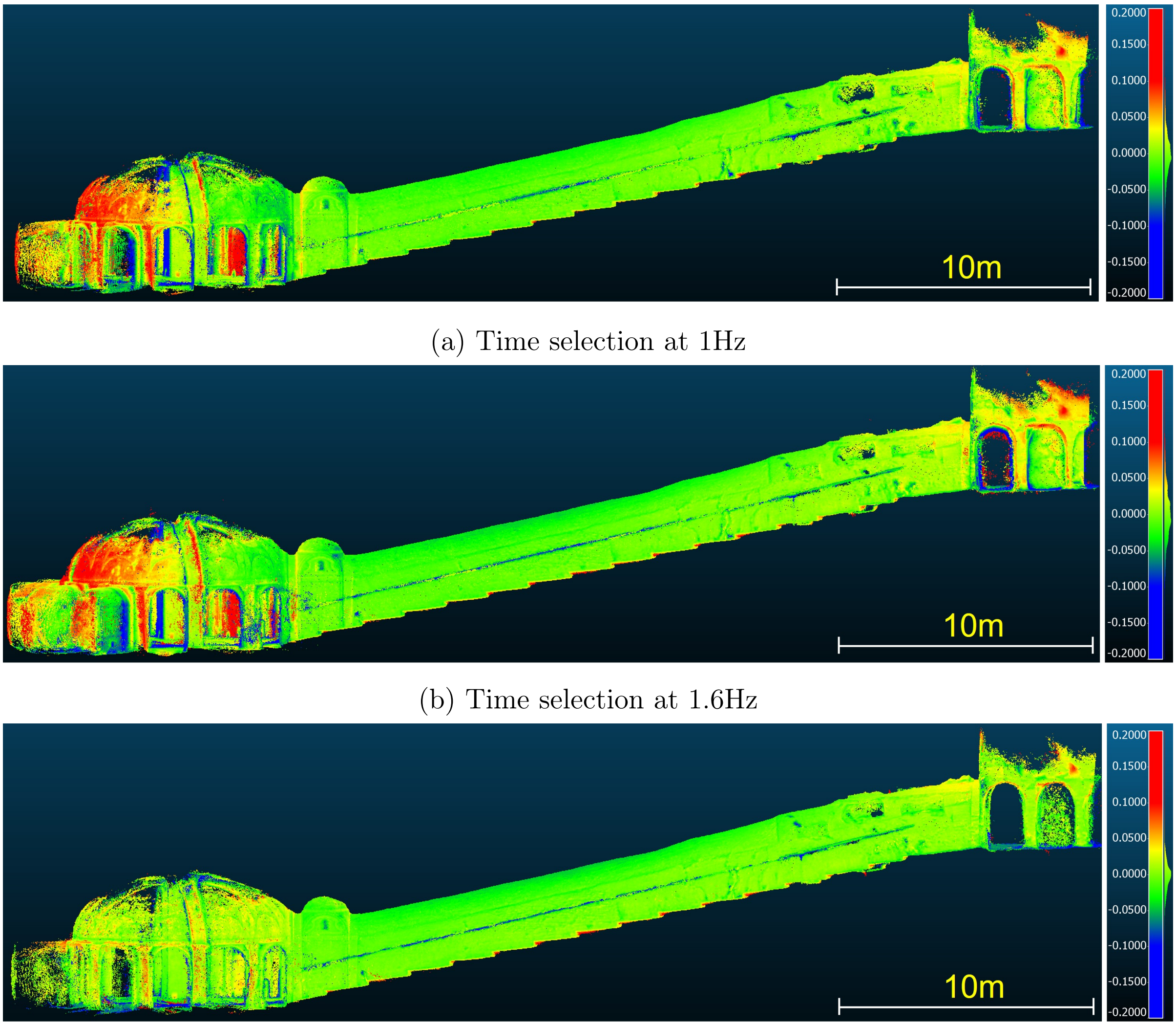
(a)

(b)

(c)

(d)

Figure 5.5: Experiment B. (a) Top view of the system trajectory. Top view of the positions of the images selected by the time (1 Hz (b), 1.6 Hz (c)) and map overlap (d) methods. Colors encode image index where blue and red correspond respectively to the start and end of the trajectory.



(a) Time selection at 1Hz

(b) Time selection at 1.6Hz

(c) Map overlap selection

Figure 5.6: Experiment B - Dense point clouds obtained with the different selection methods with color-coded signed Euclidean distances (meters) against the ground truth model.

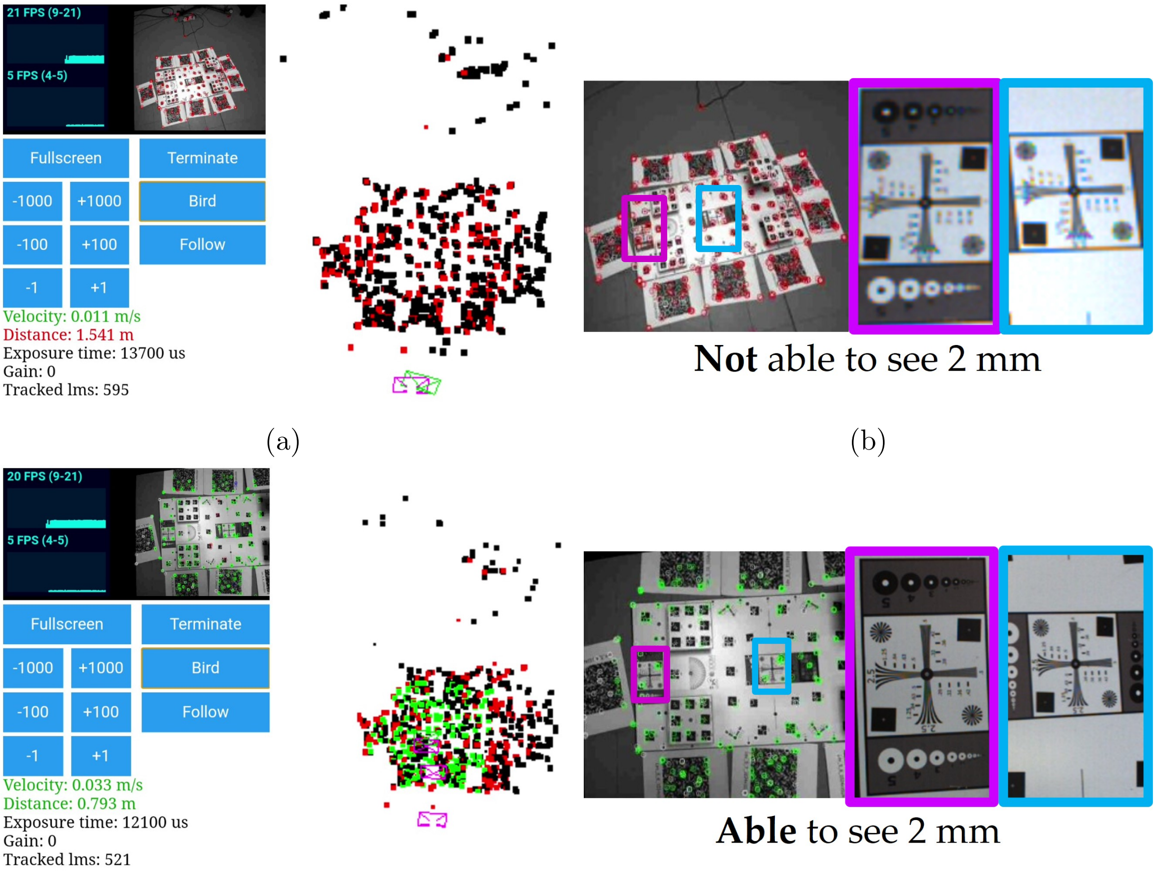
5.2.1.1 Discussion

The experiments show that the proposed acquisition control methods perform generally better than common time-based approaches. In experiment A, the methods were able to significantly reduce the number of saved images while practically retaining the same accuracy performances (Table 5.1 and Figure 5.4). Differently than the time selection (Figure 5.3b), the image overlap and the map overlap methods nicely adapted the acquisition frequency to actual distance to the building. This is clearly visible in Figures 5.3c and 5.3d, especially in the proximity of the corner and towards the end of the trajectory. Thanks to the reduced number of images, the speedup in the reconstruction times is massive (∼421%) and, although in this experiment the difference is measured in terms of minutes, larger dataset may experience differences of hours or days. The experiment B depicts a completely different scenario. The time selection (1 Hz) selected here less images then the map overlap approach but resulted in significantly higher 3D reconstruction errors (Table 5.2 and Figures 5.6a and 5.6c). To investigate if this was related to a problem of insufficient selection frequency, the second time-based experiment used a higher selection frequency (1.6 Hz) yielding a number of images more in line with the map overlap approach. Nevertheless, the reconstruction results shows a deterioration of the performances (Figure 5.6b) and a higher standard deviation error (Table 5.2). This suggests that, for the same number of images, the map overlap approach provides a significant better image selection and, consequently, more accurate 3D reconstructions. Although it is very complex to rigorously demonstrate it, the better results of the map overlap selection are likely due to the fact that the acquisition frequency was adapted to the actual scene geometry and camera distance. As shown in Figure 5.5d, the latter used different acquisition frequencies in the narrow tunnel and in the wider rooms at the tunnel ends, while time methods did not (Figures 5.5b and 5.5c). Of course, one may manually select different frequencies for different parts of the scene, but this would require to invest time to find the most appropriate frequency for each portion of the dataset. The proposed acquisition controls methods do no require any parameter tuning, and the results show that the map selection approach performed very well in two very different situations.

5.2.2 Ground sample distance and motion-blur controls

This section presents the tests on the ground sample distance (GSD) (Section 4.3.4) and motion-blur (Section 4.3.5) controls. The reported experiments took place in the 3DOM laboratory and leveraged a small calibration field (Figure A.3a) placed on the ground and composed of several markers with known resolutions. The system was previously calibrated (Table A.2), and, to enable a more precise GSD control, the actual power resolution of the lenses was measured using a Modulation Transfer Function (MTF) analysis (Section A.2). The target GSD range of the system was set between one and two millimeters. The experiment was then structured as follows: (i) start the acquisition significantly outside of the target range; (ii) move closer to the calibration field until the system notifies that we reached the target acquisition distance; (iii) start doing different strips over the calibration field, keeping the distance constant and alternating between strips predicted in the speed range with strips predicted out of the speed range.

Ground sample distance part. Figure 5.7a shows the system interface at the beginning of the test. All the GSD indicators, i.e. the sparse point cloud, the features of the live image stream and the median distance box are all red, indicating that we are out of the target GSD range. Figure 5.7b confirms the system prediction. The thickest lines of the displayed resolution chart are two and half millimeters thick but they are not clearly visible. Figure



Velocity: 0.011 m/s
Distance: 1.541 m
Exposure time: 13700 us
Gain: 0
Tracked lms: 595

(a) (b)

Velocity: 0.033 m/s
Distance: 0.793 m
Exposure time: 12100 us
Gain: 0
Tracked lms: 521

(c) (d)

Figure 5.7: Ground sample distance control test. See text for a detailed explanation.

5.7c displays the system interface upon reaching, according to the system prediction, the target acquisition range. All the GSD indicators are indeed green, and Figure 5.7d confirms again the correct prediction of the system: details of two millimeters can be clearly distinguished.

Motion-blur part. Keeping the reached distance, the system was then moved over the calibration field following several horizontal strips, alternating between strips predicted in the speed range with strips predicted outside the maximum allowed speed range. Figure 5.8 shows on top the estimated movement speed (blue) and the maximum allowed speed (orange) considering the current exposure time of the cameras and the desired two milliliter resolution. It is possible to notice four different strips, two in the speed range (frame indexes 0-50 and 80-175) and two outside the speed range (frame indexes 50-80, 175-200). Below the graph are shown some of the images acquired by the system during the strips, where border colors indicate the strip class (green means in the speed range, red means outside the speed range). A closer inspection of the resolution charts, visible at the bottom of the figure, reveals that only the images taken considering the speed feedback maintained a clear two milliliter resolution. The others, despite being taken at the same distance, present significant motion blur along the movement direction which made details at even two millimeter and half hardly visible.

5.2.2.1 Discussion

The presented experiment validated the correctness of the proposed approach to control the ground sample distance and motion-blur. Simply following the feedback of the system interface, we were able to achieve the desired image resolution of two millimeters (Figures 5.7c and 5.7d) and avoid motion-blur issues (Figure 5.8 - green strips). The results are also confirmed in the opposite direction. The target two millimeter resolution was not respected when the ground sampling distance (Figures 5.7a and 5.7b) and motion-blur controls (Figure 5.8 - red strips) returned negative results.

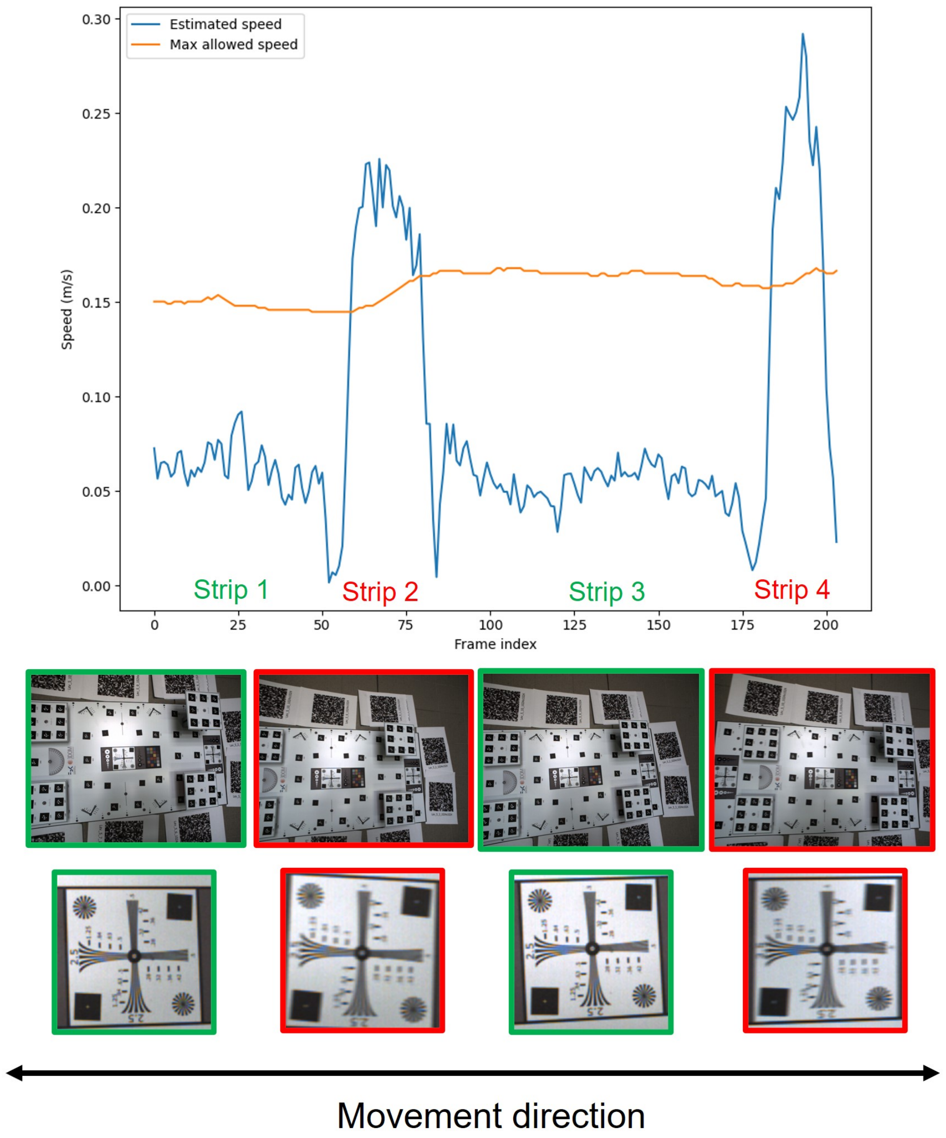


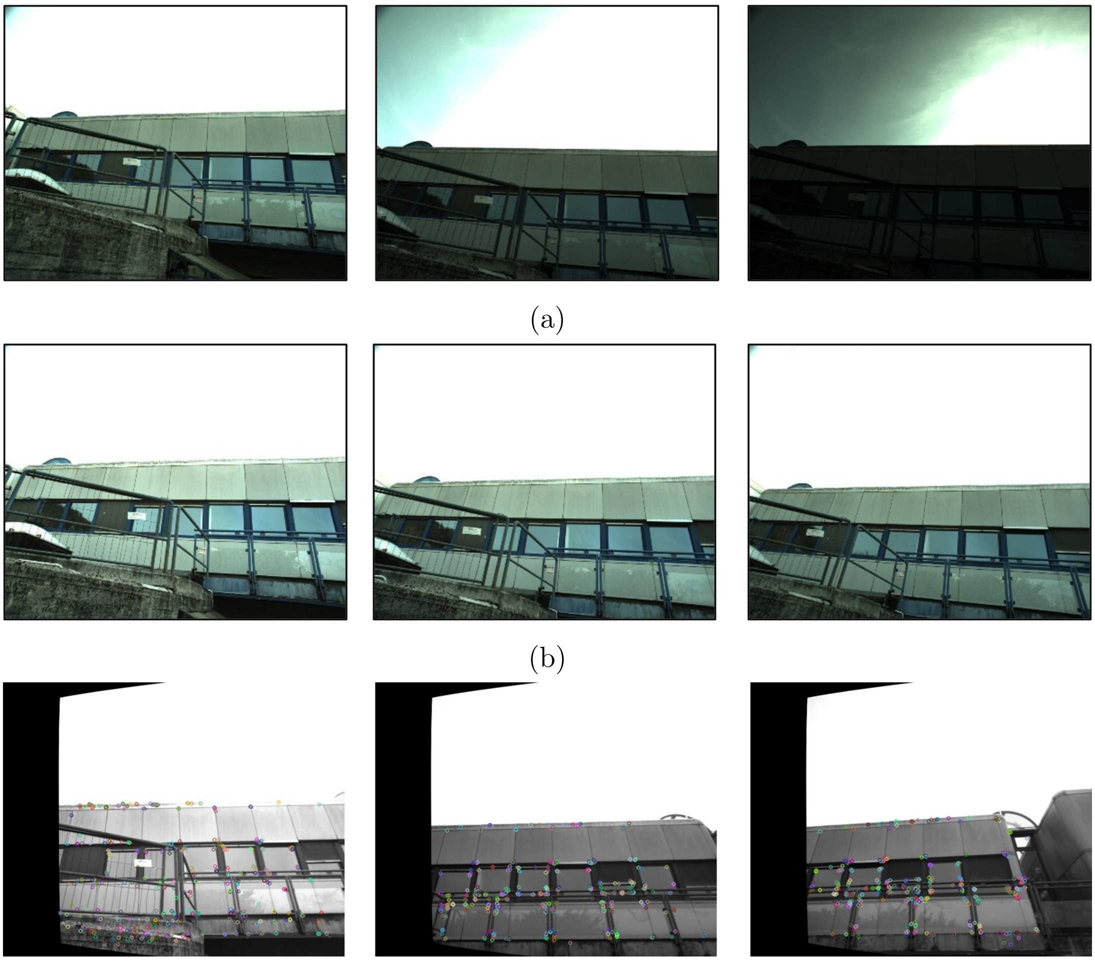
Figure 5.8: Motion-blur control test. See text for a detailed explanation.

5.2.3 Camera control

The proposed exposure control algorithm (Section 4.3.6) is here evaluated in scenarios characterized by challenging lighting conditions. These happen typically when the scene presents strong illumination variations that cannot be managed, due to the limited dynamic range of the cameras, with a single exposure time. We evaluate here two specific situations. One with the target object imaged against a brighter background, and one with the target object imaged against a darker background. For each experiment, the same scene is acquired twice following the same trajectory and using two different exposure control algorithms: the integrated automatic exposure control of the cameras and the proposed exposure control method.

Experiment A. This experiment took place in the north part of the FBK building environment (Section 5.1.1). The building was recorded with the system in two consecutive runs, keeping roughly the same trajectory. Since the sky was brighter than the building, this case represents a typical situation of brighter background. The two runs differed from each other by a few minutes, so the scene illumination practically remained the same. In the first run, the cameras used their integrated exposure control, while in the second run the camera exposure was controlled by the proposed algorithm. The system target ground sample distance was configured in such a way that the building resulted in range for the computation of the optimal exposure time. The obtained results are shown Figure 5.9.

Experiment B. This experiment considers instead a situation of darker background and was carried out in the 3DOM lab. The subject of the acquisition is in this case a white statue made of plastic material, which was illuminated by different LED light panels positioned around it. This creates a situation of darker background because the statue is more illuminated



(a)

(b)

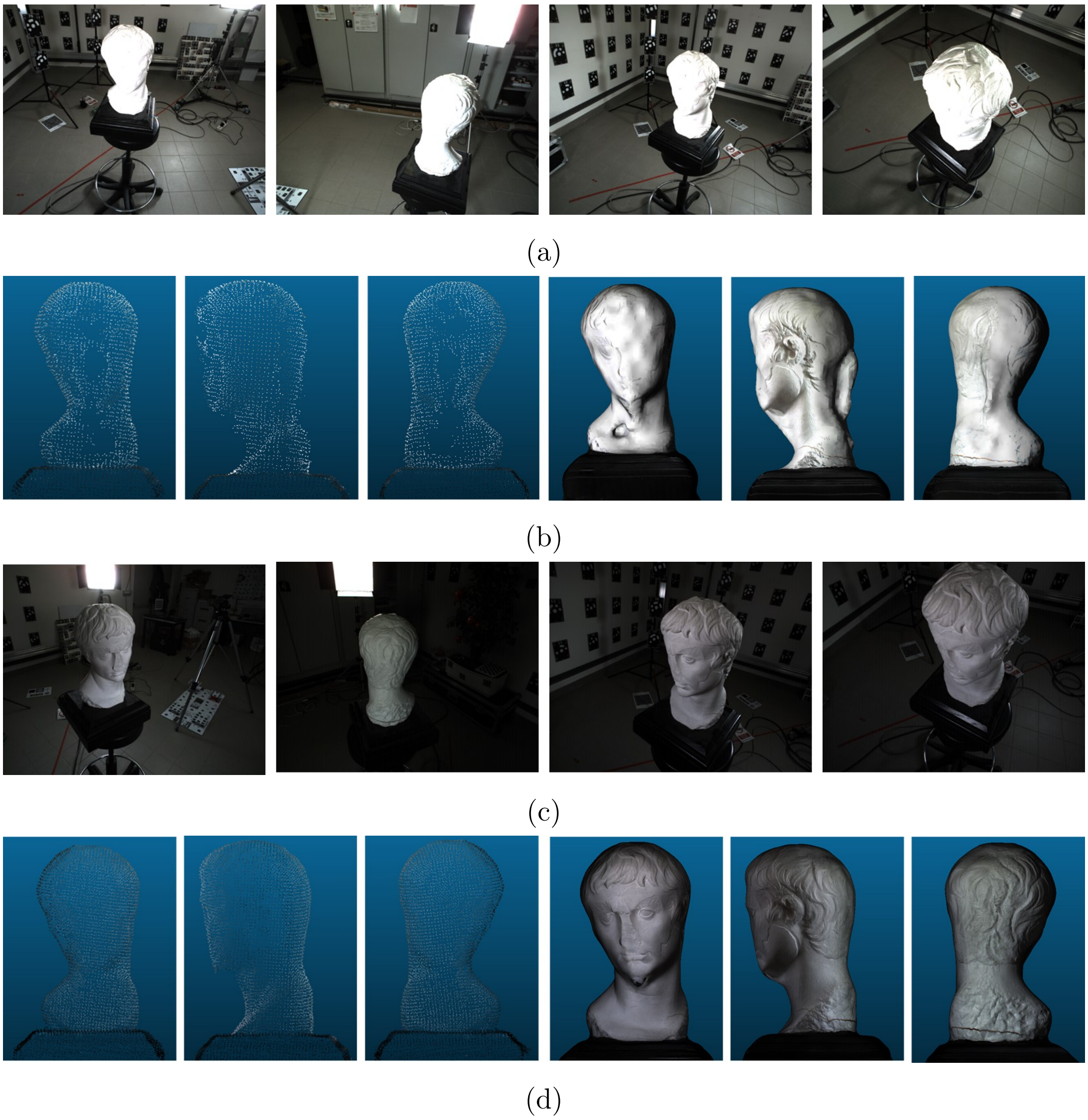
(c)

Figure 5.9: Experiment A - Some of the images acquired with integrated camera exposure control (a) and the proposed exposure control (b). (c) Examples of key-point locations used to mask the image .

than the surrounding areas. Two roughly identical acquisitions were then performed around the object, where the first acquisition used the integrated exposure control of the cameras and the second one the proposed algorithm. As in the previous experiment, the target ground sampling distance of the system was configured in such a way that the statue appears within the acquisition range while the background regions do not. The results of the experiment are shown in Figure 5.10.

5.2.3.1 Discussion

The experiments highlight the advantages of the proposed camera exposure control in conditions of challenging illumination. In experiment A, the system was able to correctly expose the building despite the presence of the strong back light of the sky (Figure 5.9b). As shown in Figure 5.9c, the optimal exposure time was computed on the key-points distributed over the building neglecting the sky contribution in the computations. On the other hand, the integrated exposure control of the cameras was negatively influenced by the sky and significantly underexposed the building (Figure 5.9a). Also the experiment B confirms the better results of the proposed method. Due to the darker background, the statue often resulted overexposed when using the camera exposure control (Figure 5.10a). Conversely, the statue does not present burned or missing information when the system used the proposed exposure control (Figure 5.10c). Two three-dimensional models of the statue were then computed from the two sets of images. The 3D model estimated from the overexposed images (integrated camera exposure control) presents holes in the dense 3D reconstruction (Figure 5.10b - left) as well as deformations and burned textures in the final textured 3D model (Figure 5.10b - right). On the other hand, the 3D model estimated from the images acquired with the proposed exposure control presents a much more complete dense reconstruction (Figure 5.10d - left) and a significantly more accurate



(a)

(b)

(c)

(d)

Figure 5.10: Experiment B - (a) Examples of images acquired with the integrated camera exposure control and (b) corresponding dense point cloud (left) and textured 3D model. (c) Examples of images acquired with the proposed exposure control and (d) corresponding dense point cloud (left) and textured 3D model.

and well-textured 3D model (Figure 5.10d - right). These results underline once more the importance of a correct object exposure in the context of three-dimensional reconstructions.

5.3 System tests

The aim of this section is to present several tests where the system and all its modules were used in extended and challenging scenarios, hence simulating real use case applications of 3D mobile mapping.

5.3.1 FBK building

The scenario of this test is the entire perimeter of the FBK building (Section 5.1.1). The system was configured with rectilinear lenses and it was calibrated before doing the test (Table A.3). The acquisition control was managed by the image overlap (IO) method (target overlap of 80%), while the target ground sample distance range was set between three and thirty millimeters. The exposure time of the cameras was automatically managed by the proposed exposure control.

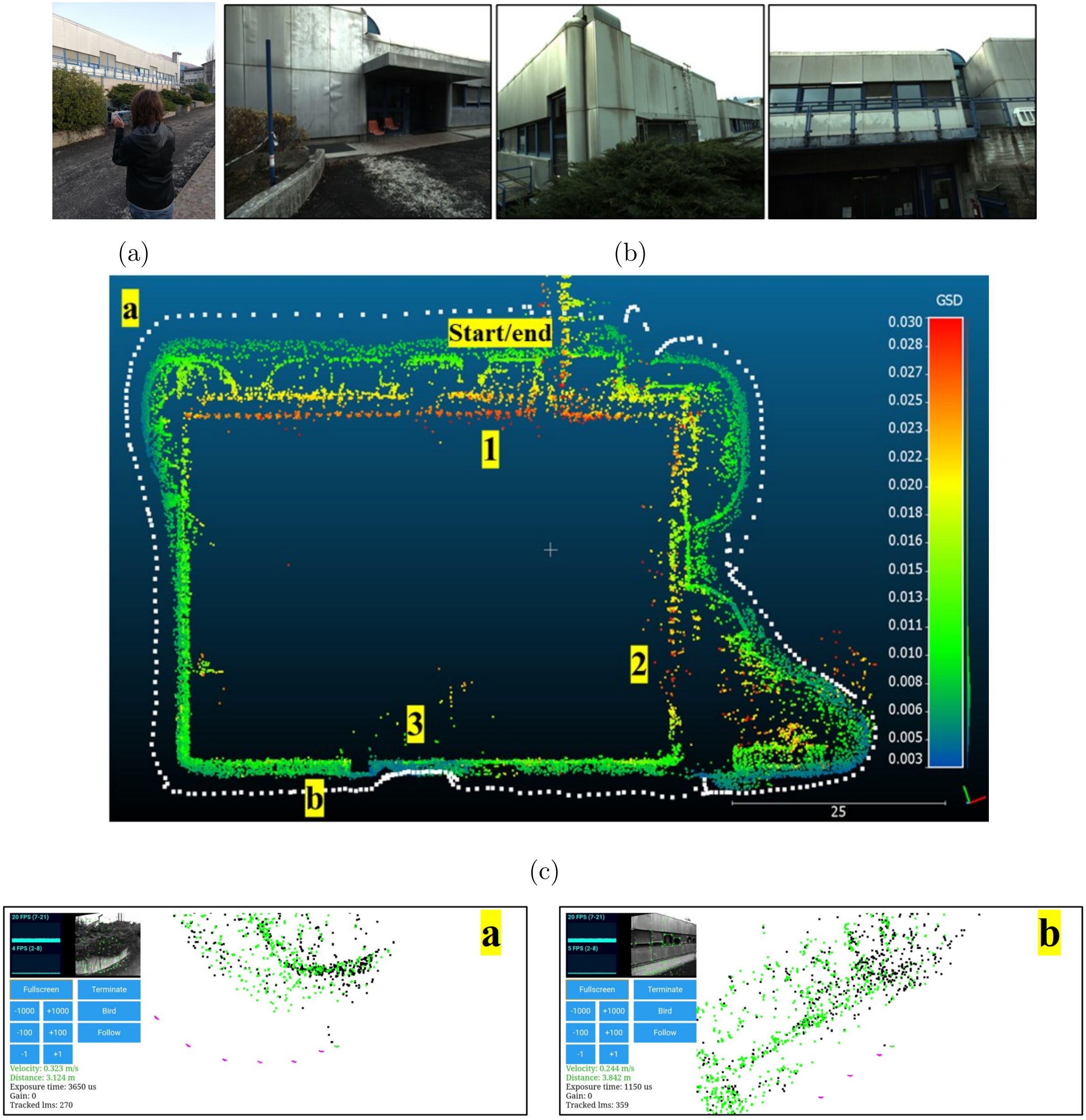
5.3.1.1 Data acquisition

The acquisition started and ended from the same position along the west side of the building, and it was performed keeping the camera sensors mostly parallel to the building facades. The real-time sparse reconstruction was used to continuously check that the whole building was correctly captured, following also the GSD and speed feedback to avoid problems related to distance and motion-blur. Along a trajectory of about 315 meters, the acquisition control selected and saved 271 image pairs. At the end of the acquisition, upon revisiting the starting area, OpenVSLAM correctly detected the loop and applied a global trajectory optimization. The whole acquisition lasted approximately 18 minutes. Figure 5.11 summarizes the acquisition phase. In Figure 5.11c are shown the positions of the saved images and the real-time sparse reconstruction, color-coded with the acquisition GSD. Some areas of the building, such as 1 and 2 in Figure 5.11c, were not acquired at the target distance due to physical limitations to reach closer positions while walking. In area 3 of the same figure, a van parked at the middle of the street that forced a slight trajectory variation. This unexpected event was automatically managed by the acquisition control that adapted the baseline considering the shorted distance from the building. Finally, figure 5.11d displays some live screenshots of the viewer showing the real-time sparse reconstruction and the poses of the saved images.

5.3.1.2 Data processing

The acquired stereo pairs were then processed with Agisoft Metashape to obtain the dense reconstruction of the building. This process was carried out following two different approaches:

- Approach 1: the dense reconstruction (MVS) is computed directly from the camera poses estimated in real-time by the system during the acquisition of the images. The intrinsic and distortion parameters of the cameras are kept fixed and set equal to those obtained during the calibration (Table A.3 and Section A.3).
- Approach 2: the camera poses are re-estimated from scratch using an offline SfM pipeline before computing the dense reconstruction (MVS). The real-time pose estimates are imported as initial approximations to enable a faster location-based feature matching (reference pre-selection in Metashape [97]), and later reset in the bundle adjustment phase. Leveraging the known baseline between the cameras, scale constraints, also known as scale bars in the software, are inserted between each stereo



(a) (b)

(c)

(d)

Velocity: 0.323 m/s
Distance: 3.124 m
Exposure time: 3650 us
Gain: 0
Tracked lms: 270

Velocity: 0.244 m/s
Distance: 3.842 m
Exposure time: 1150 us
Gain: 0
Tracked lms: 359

Figure 5.11: Data acquisition summary of the FBK building system test. (a) A view of the acquisition phase. (b) Examples of acquired images. (c) Positions of the acquired image pairs (white dots) and real-time sparse reconstruction with colors indicating the average acquisition GSD (meters). (c) Some live captures of the smartphone viewer.

pair. As in the first approach, the intrinsic and distortion parameters of the cameras are imported from the calibration results and kept fixed during the all process.

The first approach can provide an indirect evaluation of the tracking accuracy of the system, given that the accuracy of the dense point cloud is deeply related to that of the camera poses. Furthermore, because the camera poses are already known, this approach results quicker to complete. This can be a valuable choice in time-constrained scenarios where long offline computations cannot be afforded. On the other hand, in the second approach, the camera poses are recomputed using full image resolution and offline and robust computations. As a result, the latter approach should represent theoretically the most accurate reconstruction pipeline. In both cases, we fixed the values of the intrinsic, distortion, and baseline parameters with those obtained with the system calibration (Table A.1). The two reconstructions took respectively 21 and 30 minutes to correctly terminate on a powerful workstation (Intel i7-6800k, 24 GB RAM, NVIDIA 1070).

5.3.1.3 Evaluation

The two estimated dense point clouds were then filtered from noise using the confidence scalar and aligned with the laser ground truth using five selected and distributed areas of the building. The alignment was done using the iterative closest point algorithm of CloudCompare [98], disabling the scale adjustment and enabling the removal of the furthest point at every iteration. The final alignment errors were 0.02 meters (Approach 1) and 0.019 meters (Approach 2). Signed Euclidean distances were computed between the dense point clouds and the ground truth model. Figure 5.12 shows the obtained results. In both cases, the errors range from a few centimeters in the south and east part of the building, to some 20 centimeters in the north and west ones. The distribution of the signed distances of the first approach (Figures

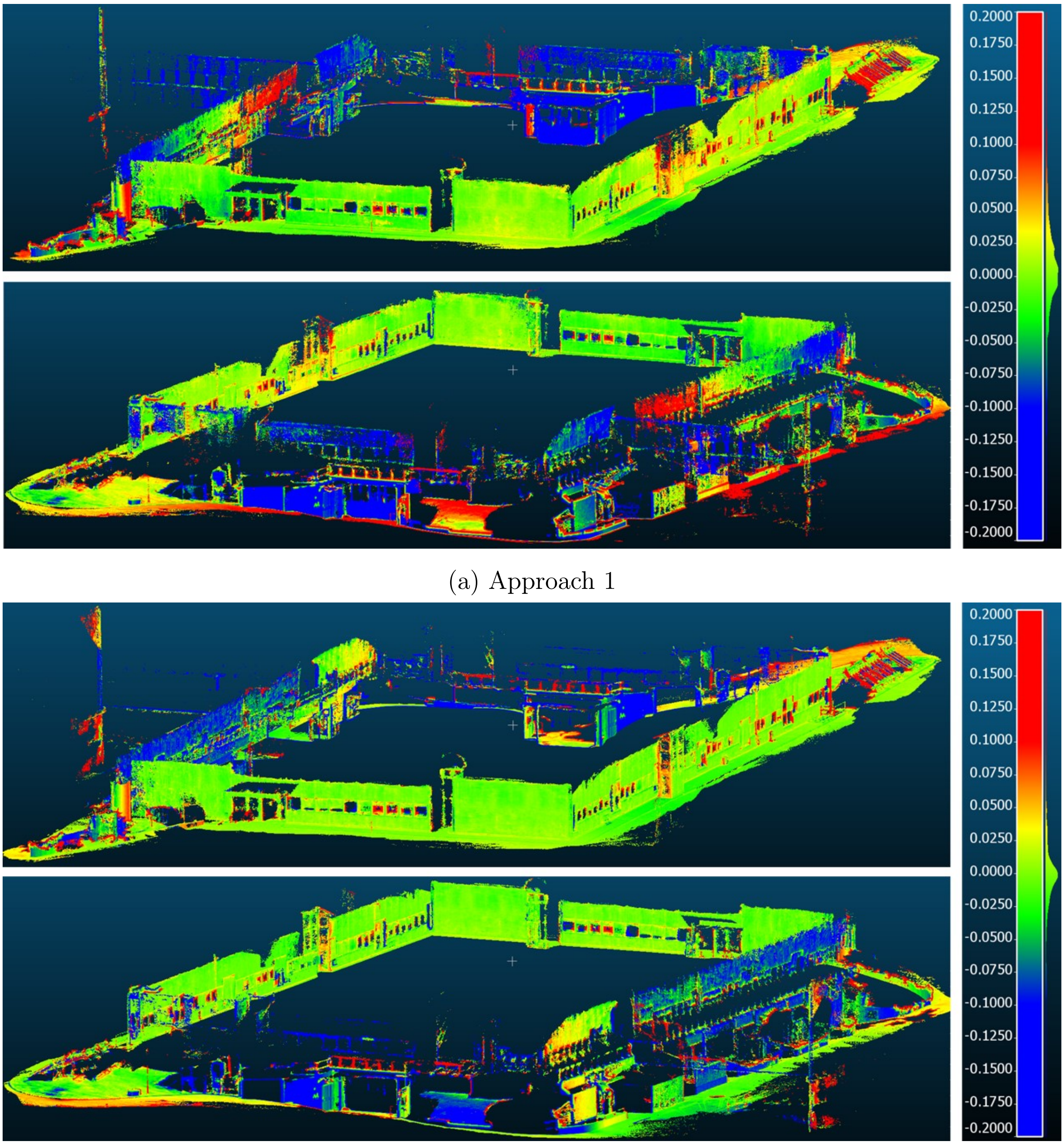
Pair	Laser [m]	Dense [m]	LME [m]	RLME [%]
1	4.7766	4.7508	0.0258	-0.5401
2	2.3843	2.3701	0.0142	-0.5956
3	3.5795	3.5854	0.0059	0.1648
4	0.4948	0.4901	0.0047	-0.9498
5	2.3888	2.3433	0.0455	-1.9047
6	2.3730	2.3840	0.011	0.4635
7	4.7647	4.7838	0.0191	0.4009
Median			0.0142	-0.5401

Table 5.3: LME and RLME results on the selected segments (Figure 5.14).

5.12a and 5.13a) has a mean of 0.003 meters and a standard deviation of 0.07 meters, with 95% of the differences bounded in bounded in the interval [-0.159, 0,166] meters. The second approach returned a slightly better error distribution (Figures 5.12b and 5.13b), with a mean value of -0.009 meters, a standard deviation of 0.054 meters and 95% of the differences falling in the interval [-0.144, 0.114] meters. Additionally, the local accuracy of the reconstruction was analyzed as well. Seven segments were selected from the the dense (Approach 1 - Figure 5.14) and ground truth point clouds, and their length measured in the three-dimension space. The length measurements were then compared using common metrics used in the literature [99] like the LME (length measurement error) and RLME (relative length measurement error) [99]. The results are reported in Table 5.3.

5.3.2 Camerano

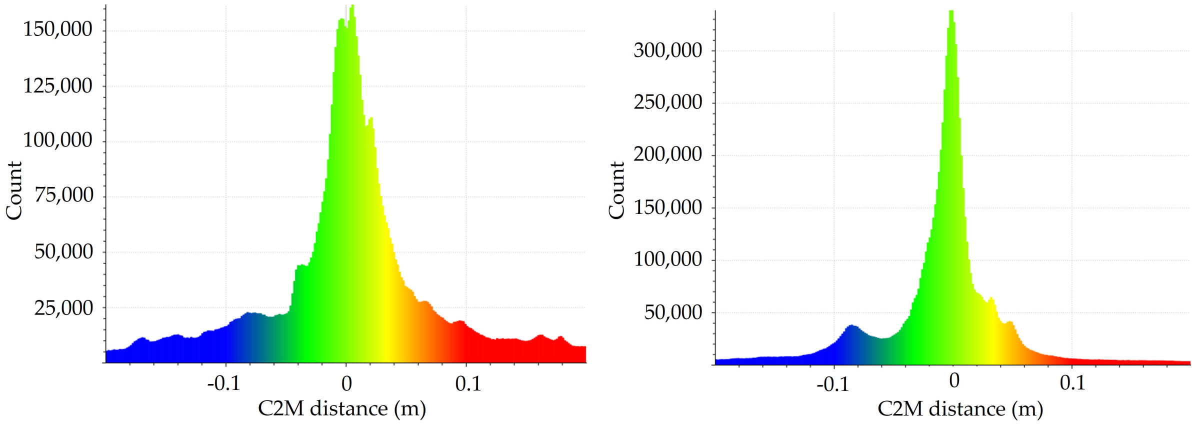
The system is here tested in the part B of the Camerano environment (Section 5.1.2). The environment is characterized by narrow passages and poorly and unevenly illuminated sections, so fisheye lenses and a LED light panel were employed in this experiment. The system was calibrated the day before the test (Table A.4). The acquisition control was here based on the map over-



(a) Approach 1

(b) Approach 2

Figure 5.12: FBK system test. Signed Euclidean distances (meters) between the evaluated dense point clouds and the ground truth model.



(a) Approach 1

(b) Approach 2

Figure 5.13: Distribution of the signed Euclidean errors shown in Figure 5.12

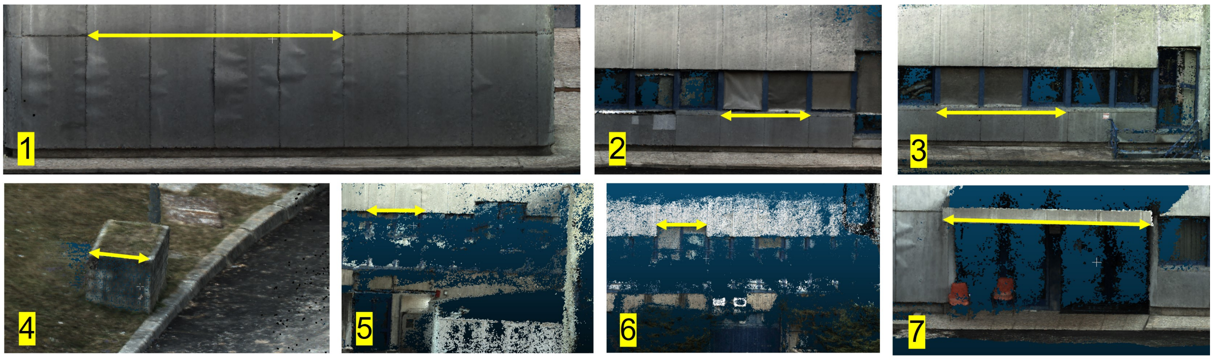
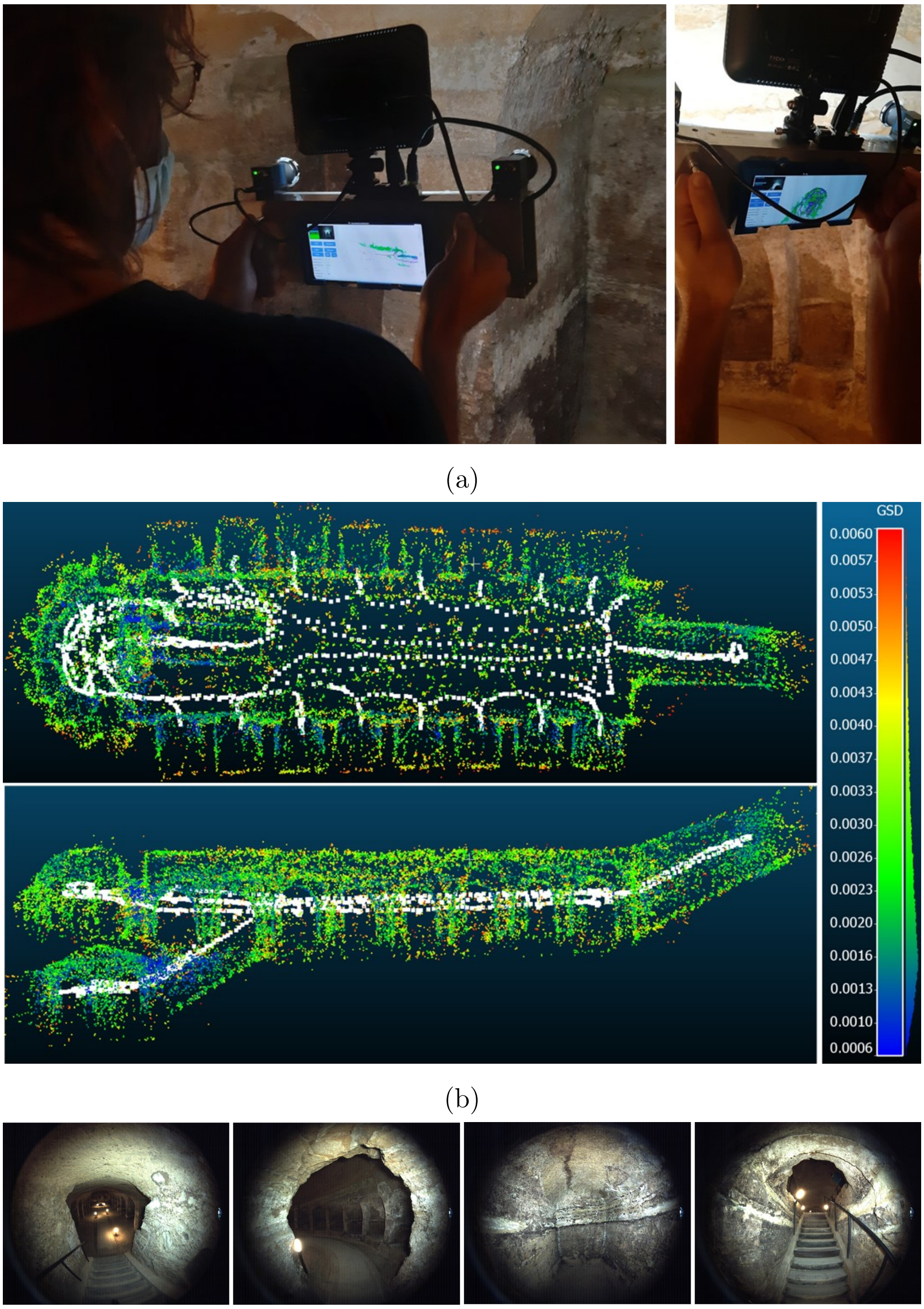


Figure 5.14: Segments considered in the LME and RLME analyses (Table 5.3).

lap method, and the target ground sample distance range was set between one and six millimeters. As in the previous test, the control of the camera exposure time was handled by the proposed method. In addition to the terrestrial laser scanner ground truth, the area was also previously mapped with a GeoSLAM ZEB Horizon (Section 2.2.1). This handheld device represents today the state-of-the-art in the commercial domain of handheld mobile laser-scanner systems. Therefore, this section presents also a comparison between the two systems, considering acquisition and processing times as well as quantitative and qualitative evaluations against the available TLS ground truth.

5.3.2.1 Data acquisition

The data acquisition was carried out in a closed-loop trajectory, so the starting and ending points coincide and correspond to the beginning of the narrow stairs leading to the church. During the acquisition, the real-time 3D reconstruction was leveraged to incrementally check that all the areas of the hall, including the multiple and complex niches, were covered by the images and imaged at the target resolution. Despite the LED light panel, a high camera exposure time was necessary in many parts of the hall, so a very conservative average movement speed of 0.15 meters/second was kept to avoid warnings from the motion-blur control. The acquisition lasted in total around 24 minutes, the trajectory was estimated, after the loop closure, approximately 223 meters long, and the acquisition control saved in total 1208 image pairs. The acquisition process is summarized in Figure 5.15. The same place was then immediately mapped with the GeoSLAM ZEB Horizon. An operator with previous working experience with the device carried out the acquisition process, which took approximately 7 minutes to be completed.



(a)

(b)

(c)

Figure 5.15: Data acquisition summary of the Camerano system test. (a) Some views of the acquisition phase. (b) Positions of the saved image pairs (white dots) and real-time sparse reconstruction with colors indicating the average acquisition GSD (meters). (c) Examples of acquired images.

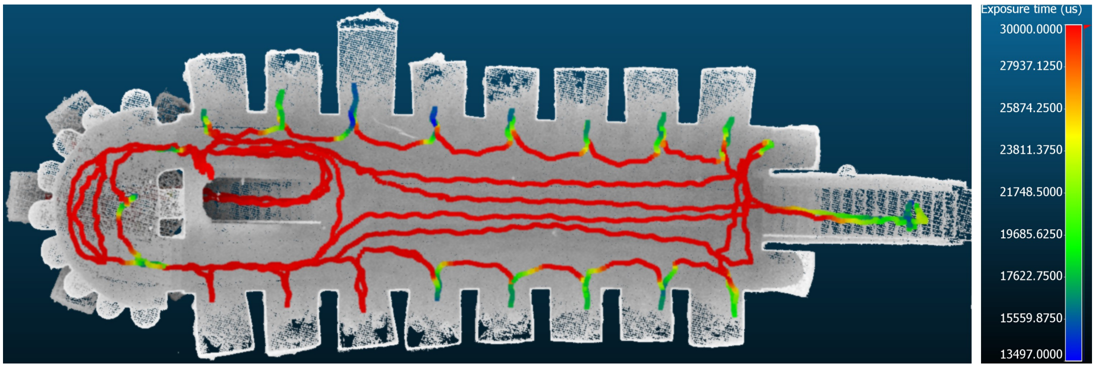


Figure 5.16: Acquisition trajectory color-coded with the camera exposure time (us).

5.3.2.2 Data processing

As in the previous experiment (Section 5.3.1), two different reconstruction approaches were followed to estimate, in Metashape and from the acquired images, the dense point clouds. In the first approach, the dense reconstruction was obtained directly from the poses estimated in real-time by the system, while, in the second approach, the camera poses were re-estimated with an offline SfM pipeline before running the dense reconstruction. The intrinsic, distortion and baseline parameters, obtained with the system calibration (Table A.4), were kept fixed during the estimation. Using the same workstation of the previous experiment, the two reconstructions took respectively 246 and 276 minutes to be successfully completed. The data acquired with the GeoSLAM ZEB Horizon was processed with the proprietary GeoSLAM Hub software. The software performs an offline optimization of the acquisition trajectory and returns the optimized point cloud. This operation was performed on the proprietary cloud service and lasted around 15 minutes.

5.3.2.3 Evaluation

The alignment of the evaluated point clouds, namely the two photogrammetric dense point clouds (P1 - first approach and P2 - second approach) and the GeoSLAM laser point cloud (L), with the ground truth (GT) was done

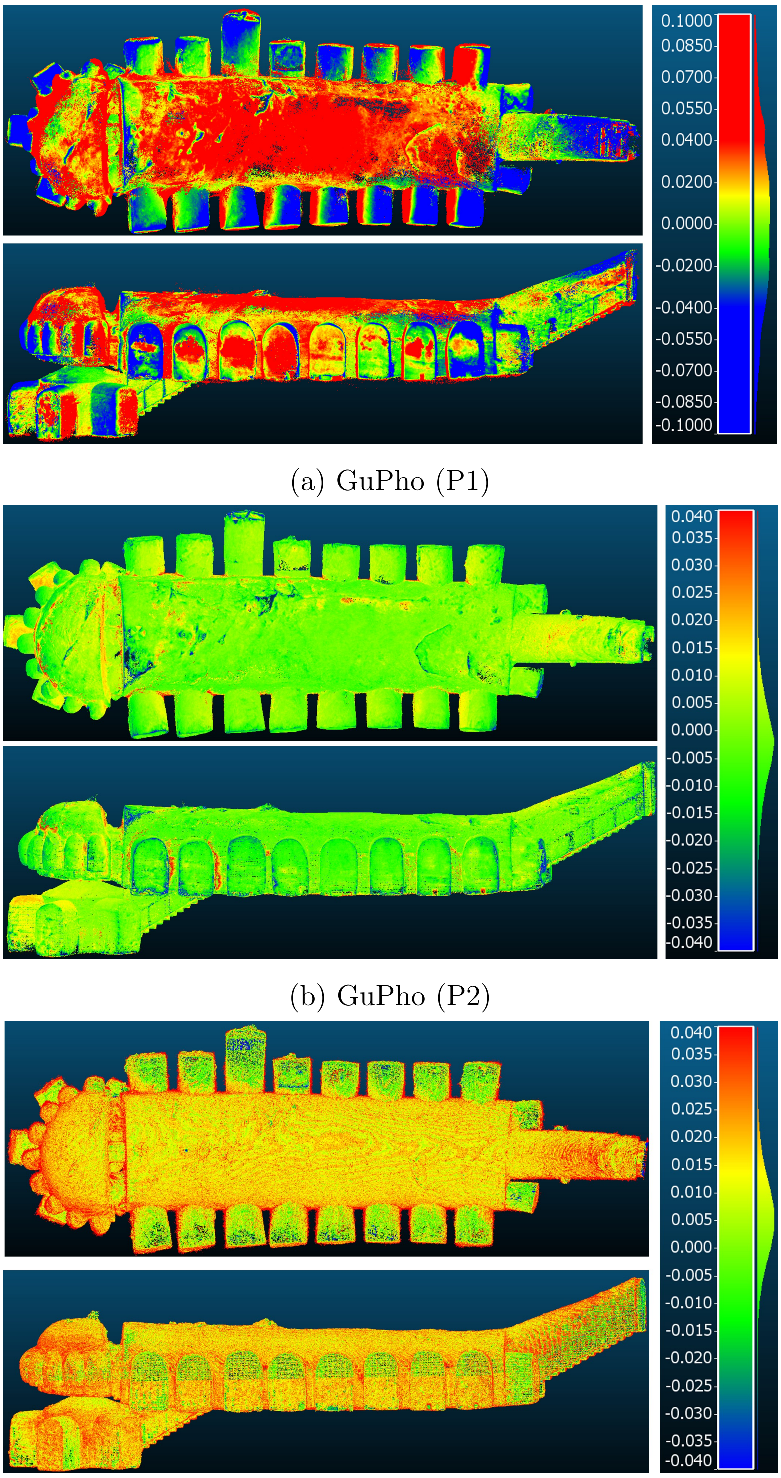
System	**Acquisition time**	**Processing time**	**Signed distances [m]**
GuPho (P1)	24 minutes	236 minutes	-0.007 ± 0.0522
GuPho (P2)		276 minutes	**-0.006 ± 0.0299**
GeoSLAM ZEB Horizon (L)	**7 minutes**	**15 minutes**	0.0039 ± 0.0331

Table 5.4: Summary of the Camerano system test.

with ICP and CloudCompare. The scale option was unchecked and at every iteration the furthest point was removed from the computations. The final alignment error, in meters, was 0.0045 for P1, 0.0043 for P2 and 0.0366 for L. Signed Euclidean distances were then computed between the evaluated P1, P2 and L point clouds and the ground truth mesh model. The results are shown in Figure 5.17. The photogrammetric reconstructions P1 and P2 returned respectively signed distances with an average of -0.007 and -0.006 meters, a standard deviation of 0.0522 and 0.0299 meters, and 90% of the values bounded in the intervals [-0.0788, 0.0793] and [-0.0278, 0.0125] meters. The L point cloud distances has an average value of 0.0039 meters, a standard deviation of 0.0331 meters, and 90% of the values fall in the interval [-0.008, 0.0203] meters. In addition to this quantitative evaluation, Figure 5.18 proposes also a qualitative comparison of the optimized P2 and L point clouds, showing several profiles selected in geometrically complex areas of the hall.

5.3.3 Other tests

This section collects some of the tests done with the system during the various development phases. In the following scenarios the TLS ground truth was not available, so metric evaluations are not reported here. Still, the presented results can be useful to have further proofs of the flexibility of the proposed system. The results are shown in Tables 5.5 and 5.6. Tests with a rectilinear



(a) GuPho (P1)

(b) GuPho (P2)

(c) GeoSLAM ZEB Horizon (L)

Figure 5.17: Camerano system test. Signed Euclidean distances (meters) between the evaluated point clouds and the ground truth model.

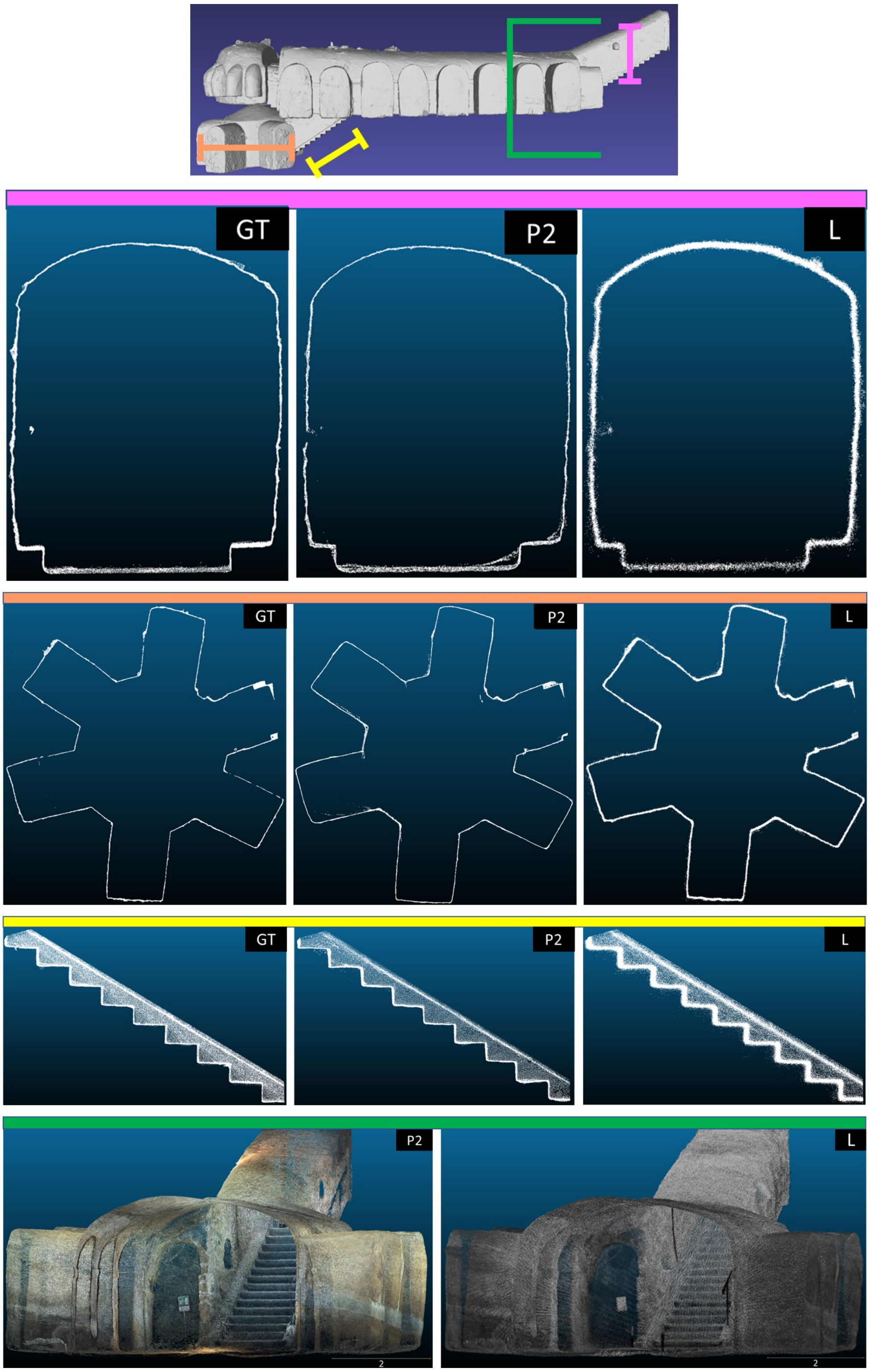


Figure 5.18: Qualitative evaluation of the Camerano system test.

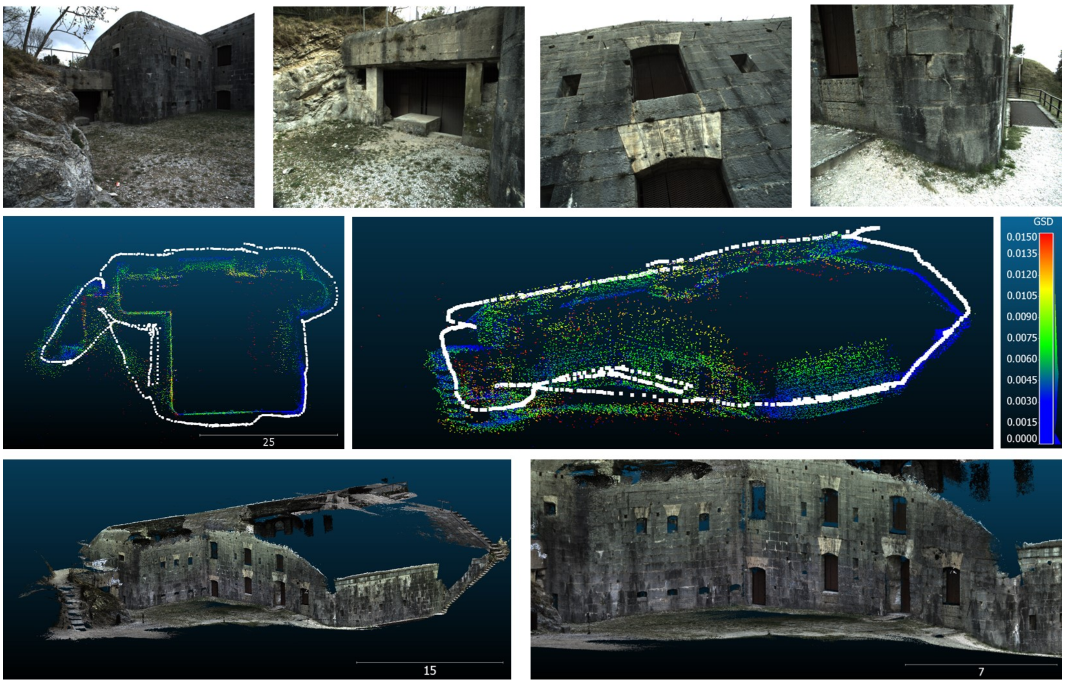
system configuration are reported in the first table, whereas tests with a fisheye configuration are shown in the second table. For each test, the rows display in order some of the acquired images, the real-time and GSD color-coded sparse 3D reconstruction and positions of the saved images, and the corresponding dense offline 3D reconstruction.

5.3.4 Discussion

The experiments presented in this section demonstrated how the proposed system and modules can enable good reconstruction results in complicated and extended environments, with results that are sometimes comparable to far more expensive systems. In the FBK building system test, the areas that were properly reachable on foot presents a reconstruction error of few centimeters (Figures 5.12 and 5.13) and RLME values of around 0.5% against the laser ground truth (Table 5.3 and Figure 5.14). These are, in my opinion, very good results considering the low cost of the device, the short acquisition time, and the challenging properties of the building in terms of size and surface type. Although it is difficult to quantify the contribution of each module to the final result, it is possible to notice their positive effects. The real-time feedback about the image coverage, object resolution and motion-blur (Figure 5.11d) greatly helped to ensure the building was acquired, where possible, entirely, at the target distance, and without significant motion blur. The acquisition control properly optimized the acquisition frequency considering the target overlap and the distance from the building (Figure 5.11c), handling also effectively unexpected environmental situations. The camera control module handled successfully the situations of high contrast, e.g. in the images shown in Figure 5.11b, and correctly exposed the building despite the sky back-light. Furthermore, we showed how it is possible to directly compute the dense reconstruction on the real-time camera poses returned by our system, achieving results that do not differ much from those obtained with

Forte Batteria di Mezzo (Riva del Garda, Trento, Italy)

Trajectory length: 286m, Acquisition time: 15min, Number of images: 380



Sentiero dei Cento Scalini (Celva, Trento, Italy)

Trajectory length: 147m, Acquisition time: 19min, Number of images: 1102

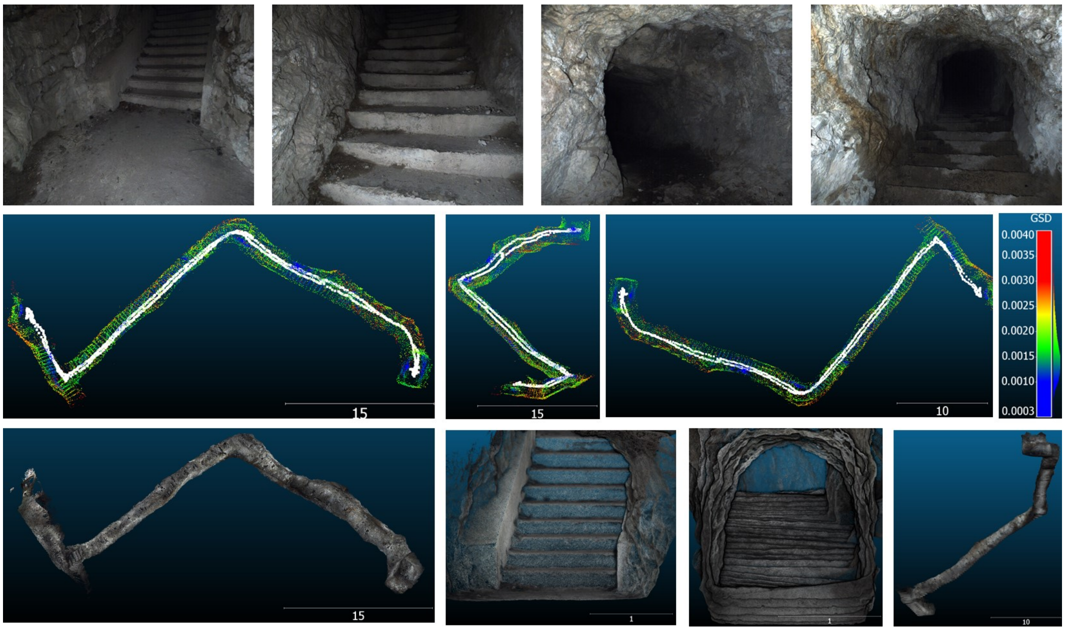
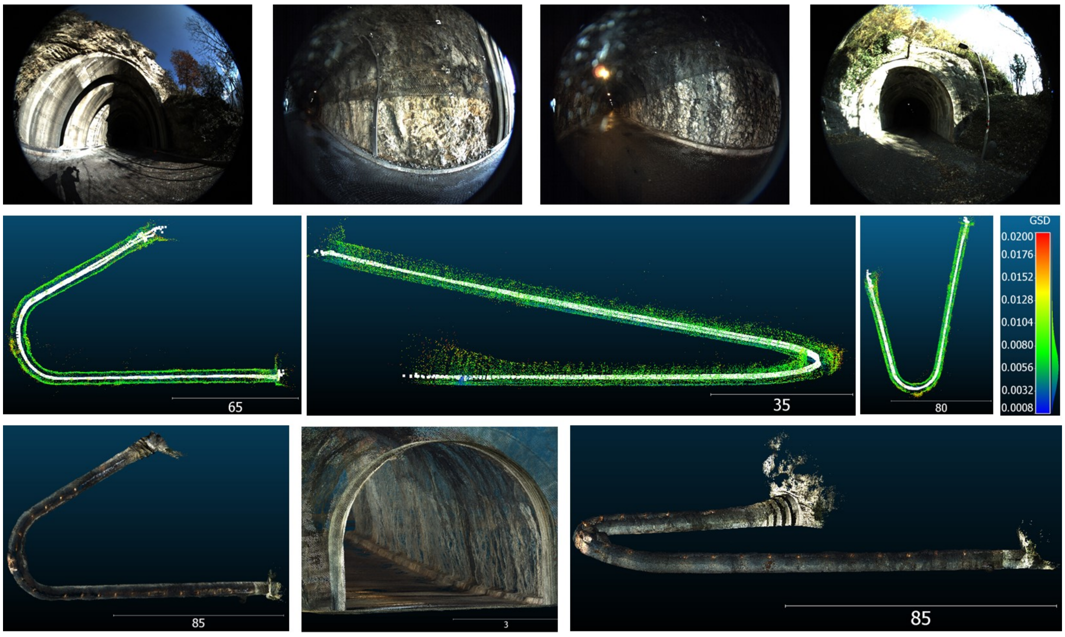


Table 5.5: Other system tests, rectilinear configuration.

Tunnel (Doss Trent, Trento, Italy)

Trajectory length: 584m, Acquisition time: 27min, Number of images: 1382



Forest (Celva, Trento, Italy)

Trajectory length: 154m, Acquisition time: 9min, Number of images: 678

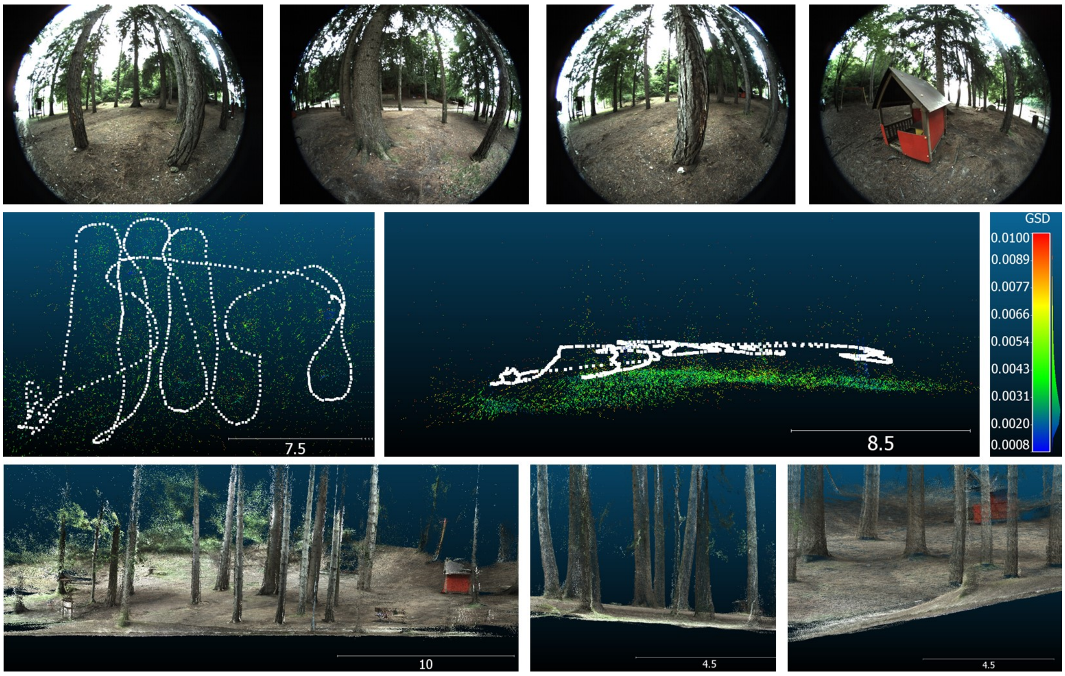


Table 5.6: Other system tests, fisheye configuration.

more time-consuming SfM pipelines (Figures 5.12a and 5.13a). The Camerano system test confirms the good results of the previous test and highlights how the system can successfully handle very different environments. Here, the real-time sparse reconstruction and GSD feedback were even more important than in the previous experiment because of the significant complexity of the scene. Thanks to the live feedback (Figures 5.15a and 5.15b), it was possible to effectively check that all the areas of the hall, including the multiple narrow niches, were correctly acquired. Also in this case, it is possible to see how the acquisition control adapted the acquisition frequency to the observed scene (Figure 5.15b), yielding higher values in the narrow parts and lower values in the larger sections. This experiment was also very challenging from an illumination point of view. Despite the LED light panel, the large areas of the hall required often a high exposure time, which was anyway limited to 30 milliseconds to avoid unpractical movement speeds. However, when the system approached narrow passages, like the niches or other small apertures, the scene in the proximity was strongly illuminated and the exposure time had to be rapidly dropped to avoid overexposure problems. These strong variations of the exposure time can be seen distinctly in Figure 5.16, where it is also possible to spot the locations where they occurred. Anyway, the proposed camera control module handled these situations very well, as it can been seen in Figure 5.15c. Considering the short focal length (1.8 millimeters) of the lenses and the target GSD set to 6 millimeters, the optimal exposure was computed on the image locations having a depth of approximately 1 meter. This helped particularly in situations like those shown in Figure 5.15c, second image from left, where the darker room in the background could have caused problems but was ignored by the our exposure control. The final reconstruction confirms the good performances of the single modules and presents errors bounded in the order of few centimeters (Figures 5.17a and 5.17b). Similarly to the FBK system test (Section 5.3.1), the dense reconstruction computed from the real-time camera poses yielded satisfactory results (Figure 5.17a and Table 5.4), despite the fact that, unlike the other two reconstructions (P2 and L), no offline optimization of the trajectory was performed in this case. Very interesting is also the comparison between the optimized reconstructions obtained with the proposed system (P2) and the significantly more expensive GeoSLAM device (L). The latter presents evident noise (Figure 5.18 - L labels) and the Euclidean distances (Figure 5.17c) are characterized by a higher standard deviation (0.0331 meters) than that obtained with our system (0.0299 meters). This are in my opinion very promising results that underline the capabilities of the proposed system. Nevertheless, as shown in Table 5.4, the acquisition and processing times are significantly in favour of the GeoSLAM device. Laser scanner systems offer incredibly high acquisition rates in multiple directions, and can maintain sustained movement speeds even in poorly illuminated scenarios. Moreover, lasers can directly measure the scene geometry without requiring complex and computationally-intensive algorithms, so the processing times are generally much shorter. Still, I believe that the presented results are very promising and highlight the advantages in terms of costs, flexibility, acquisition assistance and accuracy of the system.

Chapter 6

Conclusions

This thesis presented a novel visual portable 3D mapping system that combines low-cost and flexible hardware with an assisted and optimized acquisition of the images. Unlike most of existing handheld visual solutions, the proposed system supports the image acquisition with a real-time and low-resolution three-dimensional reconstruction of the acquired scene and a dedicated quality control feedback. The advantages of this approach were extensively and clearly demonstrated in chapter 5. On the one hand, the acquisition of complex environments is massively simplified, because the user can leverage the real-time reconstruction to monitor the progress, and assess that all the target areas are covered by the images. The experiments done in large-scale and geometrically complex environment demonstrated how crucial is this feedback to obtain a complete and accurate coverage of the scene. On the other hand, it was shown how the knowledge of the system pose and scene structure can be also exploited to actively control the acquisition parameters and provide quality measures of the acquired images. The experiments showed the effectiveness of adapting the acquisition frequency to the observed scene, providing better results than common time-based methods, both in terms of number of saved images and final reconstruction accuracy. This aspect is often scarcely considered in current systems but the experiments show how a smarter image selections can improve dramatically the processing times providing, at the same time, and improvement of the reconstruction accuracy. It was also demonstrated how to use the scene structure and a prior information of the acquisition settings to produce more accurate camera exposure results, even in situations of difficult illumination conditions. Although the exposure control cannot overcome the sensor limitations of the cameras, significant improvements were achieved by better guiding the algorithm attention and binding the exposure metering to the feature points distributed over the object of interest. It was also demonstrated how the estimates of the scene geometry and system poses can be used to prevent situations of motion-blur, and enable a precise and easy to understand feedback of the acquisition ground sample distance. This feedback can play a fundamental role in those applications that have strict object resolution requirements, like for example monitoring or inspection tasks. Furthermore, the real-time estimates of the camera poses provided by our system can be directly used, with satisfactory results, in dense reconstruction pipelines without needing time-consuming offline optimizations. Time-constrained applications, such as disaster management activities, can benefit greatly from this capability. Finally, the various system tests conducted in complex and heterogeneous environments proved the flexibility of the hardware choices and the efficacy of the proposed assistance methods. The combination of a small and power-efficient embedded architecture with industrial cameras with configurable optics and position allow the system to be easily employed in a variety of contexts, potentially including, for the reduced weight and size, even its application onboard of small robotic platforms in case of dangerous and/or very narrow environments. In the proposed system tests, the 3D reconstructions obtained from the images acquired by our system proved to be satisfactory in terms of accuracy and completeness, with results sometimes comparable with those obtained with significantly more expensive systems.

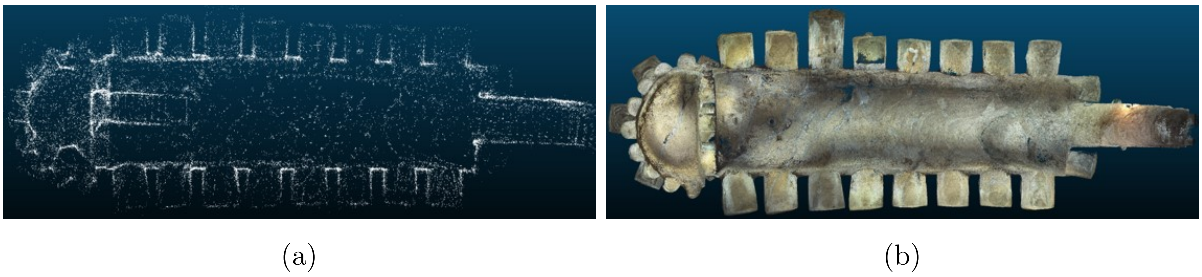
These are remarkable results that would have been difficult to achieve without the assistance and the features provided by the proposed system.

6.1 Outlook

While I hope that the contribution presented in this thesis may represent a step towards a highly assisted, rapid and accurate visual and portable 3D mapping, much work is still needed to achieve this goal. Hereafter, I collected some possible future research directions.

Real-time dense reconstruction. Although the sparse reconstruction provided in real-time by the proposed system can outline the main structure of the acquired scene and provide an important feedback of the acquisition progress, a dense reconstruction would be more effective and detailed (Figure 6.1). Furthermore, if accurate enough, it could represent the final result of the acquisition without the need for further time-consuming offline computations. The proposed experiments already demonstrated that the real-time camera poses computed by the system are sufficiently good to enable satisfactory offline dense reconstructions. Nevertheless, estimating dense and accurate three-dimensional reconstruction in real-time and on portable and power-constrained devices remain a major challenge today. Despite in the latest years some works were able to achieve it [48, 100], important compromises on the image resolutions, maximum scene size and reconstruction accuracy are still necessary.

Semantics. The combination of geometric and semantic understanding of the scene can potentially lead to important improvements. For example, leveraging the latest advances of convolutional neural networks (CNNs) [101] and pixel-wise image semantic segmentation [102], the system may understand what is the subject of the acquisition and prioritize it when computing



(a) (b)

Figure 6.1: Visual comparison of the level of detail between (a) the sparse reconstruction obtained in real-time with our system and (b) the dense reconstruction estimated after hours of processing on powerful hardware (b).

the geometrical structure of the scene or when adjusting the acquisition parameters. Another interesting application could be the detection of poorly textured areas, e.g. white walls, where geometric primitives could be leveraged to compensate for the notoriously bad performance, in this cases, of dense reconstruction algorithms [103]. Semantics could be exploited to generate on-the-fly labelled point clouds as well, which can be interesting for many applications in the context of monitoring/inspection or cultural heritage documentation.

Image enhancement. Another interesting research direction is in my opinion the employment of image enhancement techniques, as they can mitigate some important limitations of the sensors [104]. For example, in poorly illuminated environments it may be preferable to leverage some enhancement algorithms rather than using high exposure times. The latter would require slow movement speeds to avoid destructive motion blur issues, which might be undesired when the scene to acquire is large and the acquisition times should be optimized. Another situation where image enhancement could provide important improvements is when the scene presents large illumination variations that cannot be correctly captured by the cameras. In these cases, the enhancement could to be useful to extract more information from the wrongly-exposed areas, potentially enabling, in turn, more accurate and complete reconstructions. First steps in this direction have been already done by the authors [105], nevertheless the integration of the algorithms in the system as well as an in-depth analysis of the enhancement effect in the three-dimensional space still needs to be done.

Hardware improvements. One further significant research direction of the system is the investigation of different hardware configurations. An important add-on could be an Inertial Measurement Unit (IMU). In addition to providing the direction of gravity, which can be leveraged to express more conveniently the reconstruction orientation, it could be used to improve the visual pose estimation, especially when the scene is difficult and contains few visual features [106]. Another important question is how to increase the computational capability of the system without sacrificing too much the portability. The potential inclusion of dense reconstruction, semantics or image enhancement algorithms would require more computational power. An interesting option could be to use multiple computational units for different tasks, taking in consideration also Nvidia Jetson devices for GPU-related computations. Finally, the cameras can be upgraded to have higher resolution. This could significantly improve the acquisition times and the number of images as the same object resolution (GSD) can be achieved from a greater distance.

Bibliography

[1] E. Nocerino, P. Rodríguez-Gonzálvez, and F. Menna, "Introduction to mobile mapping with portable systems," in *Laser Scanning*, pp. 37–52, CRC Press, 2019.

[2] R. Otero, S. Lagüela, I. Garrido, and P. Arias, "Mobile indoor mapping technologies: A review," *Automation in Construction*, vol. 120, p. 103399, 2020.

[3] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," *Foundations and Trends® in Computer Graphics and Vision*, vol. 9, no. 1-2, pp. 1–148, 2015.

[4] E. K. Stathopoulou and F. Remondino, "Open-source image-based 3d reconstruction pipelines: Review, comparison and evaluation," in *6th International Workshop LowCost 3D–Sensors, Algorithms, Applications*, pp. 331–338, 2019.

[5] T. Sieberth, R. Wackrow, and J. Chandler, "Motion blur disturbs–the influence of motion-blurred images in photogrammetry," *The Photogrammetric Record*, vol. 29, no. 148, pp. 434–453, 2014.

[6] I. Shim, J.-Y. Lee, and I. S. Kweon, "Auto-adjusting camera exposure for outdoor robotics using gradient information," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1011–1017, IEEE, 2014.

[7] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[8] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3260–3269, 2017.

[9] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113, 2016.

[10] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision – ECCV 2016*, pp. 501–518, Springer International Publishing, 2016.

[11] C. Hoppe, M. Klopschitz, M. Rumpler, A. Wendel, S. Kluckner, H. Bischof, and G. Reitmayr, "Online feedback for structure-from-motion image acquisition," in *Proceedings of the British Machine Vision Conference*, pp. 70.1–70.12, BMVA Press, 2012.

[12] P. Ondruska, P. Kohli, and S. Izadi, "Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, p. 1251–1258, 2015.

[13] S. Sumikura, M. Shibuya, and K. Sakurada, "Openvslam: A versatile visual slam framework," in *Proceedings of the 27th ACM International Conference on Multimedia*, p. 2292–2295, Association for Computing Machinery, 2019.

[14] A. Torresani and F. Remondino, “Videogrammetry vs photogrammetry for heritage 3d reconstruction,” in *27th CIPA International Symposium “Documenting the past for a better future”*, vol. 42, pp. 1157–1162, 2019.

[15] A. Torresani, F. Menna, R. Battisti, and F. Remondino, “A v-slam guided and portable system for photogrammetric applications,” *Remote Sensing*, vol. 13, no. 12, 2021.

[16] F. Di Stefano, A. Torresani, E. M. Farella, R. Pierdicca, F. Menna, and F. Remondino, “3d surveying of underground built heritage: Opportunities and challenges of mobile technologies,” *Sustainability*, vol. 13, no. 23, p. 13289, 2021.

[17] R. Horaud, M. Hansard, G. Evangelidis, and C. Ménier, “An overview of depth cameras and range scanners based on time-of-flight technologies,” *Machine Vision and Applications*, vol. 27, no. 7, pp. 1005–1020, 2016.

[18] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (slam): Part ii,” *IEEE robotics & automation magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[19] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[20] 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,” *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[21] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proceedings of the 23rd Annual Con-*

ference on Computer Graphics and Interactive Techniques, p. 303–312, Association for Computing Machinery, 1996.

[22] A. Kukko, H. Kaartinen, J. Hyyppä, and Y. Chen, "Multiplatform mobile laser scanning: Usability and performance," *Sensors*, vol. 12, no. 9, pp. 11712–11733, 2012.

[23] M. Bosse, R. Zlot, and P. Flick, "Zebedee: Design of a spring-mounted 3-d range sensor with application to mobile mapping," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1104–1119, 2012.

[24] A. Kukko, C.-O. Andrei, V.-M. Salminen, H. Kaartinen, Y. Chen, P. Rönnholm, H. Hyyppä, J. Hyyppä, R. Chen, H. Haggrén, *et al.*, "Road environment mapping system of the finnish geodetic institute–fgi roamer," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 36, pp. 241–247, 2007.

[25] M. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based slam," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 667–691, 2008.

[26] "GeoSLAM." `https://geoslam.com/`. Accessed: 2022-02.

[27] "GeoSLAM ZEB Go." `https://geoslam.com/solutions/zeb-go/`. Accessed: 2022-02.

[28] "GeoSLAM ZEB Horizon." `https://geoslam.com/solutions/zeb-horizon/`. Accessed: 2022-02.

[29] "KAARTA Stencil 2." `https://www.kaarta.com/products/stencil-2-for-rapid-long-range-mobile-mapping/`. Accessed: 2022-02.

[30] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[31] "KAARTA Contour." https://www.kaarta.com/products/contour/. Accessed: 2022-02.

[32] "Leica BLK2GO." https://leica-geosystems.com/en-gb/products/laser-scanners/autonomous-reality-capture/leica-blk2go-handheld-imaging-laser-scanner. Accessed: 2022-02.

[33] "Leica Pegasus:Backpack." https://leica-geosystems.com/en-gb/products/mobile-mapping-systems/capture-platforms/leica-pegasus-backpack. Accessed: 2022-02.

[34] "GEXCEL Heron." https://gexcel.it/en/solutions/heron-portable-3d-mapping-system. Accessed: 2022-02.

[35] A. Nüchter, D. Borrmann, P. Koch, M. Kühn, and S. May, "A man-portable, imu-free mobile mapping system.," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3/W5, pp. 17–23, 2015.

[36] S. Blaser, S. Cavegn, and S. Nebiker, "Development of a portable high performance mobile mapping system using the robot operating system.," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1, pp. 13–20, 2018.

[37] "Robot operating system (ros)." https://www.ros.org/. Accessed: 2022-02.

[38] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1271–1278, IEEE, 2016.

[39] X. Xiang, H. Jiang, G. Zhang, Y. Yu, C. Li, X. Yang, D. Chen, and H. Bao, "Mobile3dscanner: An online 3d scanner for high-quality object

reconstruction with a mobile device," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 11, pp. 4245–4255, 2021.

[40] "Occipital Canvas." `https://canvas.io/`. Accessed: 2022-02.

[41] "Polycam." `https://poly.cam/`. Accessed: 2022-02.

[42] "3D Scanner App." `https://3dscannerapp.com/`. Accessed: 2022-02.

[43] M. Vogt, A. Rips, and C. Emmelmann, "Comparison of ipad pro®'s lidar and truedepth capabilities with an industrial 3d scanning solution," *Technologies*, vol. 9, no. 2, p. 25, 2021.

[44] M. Weinmann, S. Wursthorn, M. Weinmann, and P. Hübner, "Efficient 3d mapping and modelling of indoor scenes with the microsoft hololens: A survey," *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 89, no. 4, pp. 319–333, 2021.

[45] M. Mokroš, T. Mikita, A. Singh, J. Tomaštík, J. Chudá, P. Wężyk, K. Kuželka, P. Surový, M. Klimánek, K. Zięba-Kulawik, *et al.*, "Novel low-cost mobile mapping systems for forest inventories as terrestrial laser scanning alternatives," *International Journal of Applied Earth Observation and Geoinformation*, vol. 104, p. 102512, 2021.

[46] S. Karam, G. Vosselman, M. Peter, S. Hosseinyalamdary, and V. Lehtola, "Design, calibration, and evaluation of a backpack indoor mobile mapping system," *Remote sensing*, vol. 11, no. 8, p. 905, 2019.

[47] "Microsoft HoloLens 2." `https://www.microsoft.com/en-us/hololens/`. Accessed: 2022-02.

[48] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "Large-scale outdoor 3d reconstruction on a mobile device," *Computer Vision and Image Understanding*, vol. 157, pp. 151–166, 2017.

[49] E. Nocerino, F. Poiesi, A. Locher, Y. T. Tefera, F. Remondino, P. Chippendale, and L. Van Gool, “3d reconstruction with a collaborative approach based on smartphones and a cloud-based server,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, no. W8, pp. 187–194, 2017.

[50] O. Hasler, S. Blaser, and S. Nebiker, “Implementation and first evaluation of an indoor mapping application using smartphones and ar frameworks,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W17, pp. 135–141, 2019.

[51] “PIX4Dcatch.” `https://www.pix4d.com/product/pix4dcatch`. Accessed: 2022-02.

[52] D. Holdener, S. Nebiker, and S. Blaser, “Design and implementation of a novel portable 360 stereo camera system with low-cost action cameras,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W8, pp. 105–110, 2017.

[53] M. M. Nawaf, D. Merad, J.-P. Royer, J.-M. Boï, M. Saccone, M. Ben Ellefi, and P. Drap, “Fast visual odometry for a low-cost underwater embedded stereo system,” *Sensors*, vol. 18, no. 7, p. 2313, 2018.

[54] P. Ortiz-Coder and A. Sánchez-Ríos, “An integrated solution for 3d heritage modeling based on videogrammetry and v-slam technology,” *Remote Sensing*, vol. 12, no. 9, p. 1529, 2020.

[55] L. Perfetti and F. Fassi, “Handheld fisheye multicamera system: Surveying meandering architectonic spaces in open-loop mode – accuracy assessment,” *The International Archives of the Photogrammetry, Re-*

mote Sensing and Spatial Information Sciences, vol. XLVI-2/W1-2022, pp. 435–442, 2022.

[56] D. C. Brown, "Close-range camera calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.

[57] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications*, vol. 12, no. 1, pp. 16–22, 2000.

[58] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, pp. 1–11, 2017.

[59] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, pp. 2564–2571, IEEE, 2011.

[60] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, pp. 2548–2555, IEEE, 2011.

[61] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, IEEE, 2007.

[62] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[63] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[64] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[65] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 International Conference on Computer Vision*, pp. 2320–2327, IEEE, 2011.

[66] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision – ECCV 2014*, pp. 834–849, Springer International Publishing, 2014.

[67] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[68] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22, IEEE, 2014.

[69] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[70] X. Gao, R. Wang, N. Demmel, and D. Cremers, "Ldso: Direct sparse odometry with loop closure," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2198–2204, IEEE, 2018.

[71] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping,"

in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1689–1696, IEEE, 2020.

[72] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[73] "WebSocket." `https://datatracker.ietf.org/doc/html/rfc6455`. Accessed: 2022-02.

[74] F. Remondino and C. Fraser, "Digital camera calibration methods: considerations and comparisons," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 266–272, 2006.

[75] E. Nocerino and F. Menna, "Photogrammetry: linking the world across the water surface," *Journal of Marine Science and Engineering*, vol. 8, no. 2, p. 128, 2020.

[76] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3607–3613, IEEE, 2011.

[77] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[78] "DBoW2." `https://github.com/dorian3d/DBoW2`. Accessed: 2022-02.

[79] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Josa a*, vol. 4, no. 4, pp. 629–642, 1987.

[80] H. Strasdat, J. Montiel, and A. J. Davison, “Scale drift-aware large scale monocular slam,” *Robotics: Science and Systems VI*, vol. 2, no. 3, p. 7, 2010.

[81] V. Lepetit, F. Moreno-Noguer, and P. Fua, “Epnp: An accurate o (n) solution to the pnp problem,” *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, 2009.

[82] C.-K. Liang, L.-W. Chang, and H. H. Chen, “Analysis and compensation of rolling shutter effect,” *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1323–1330, 2008.

[83] D. Schubert, N. Demmel, L. von Stumberg, V. Usenko, and D. Cremers, “Rolling-shutter modelling for direct visual-inertial odometry,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2462–2469, IEEE, 2019.

[84] O. Saurer, M. Pollefeys, and G. H. Lee, “Sparse to dense 3d reconstruction from rolling shutter images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3337–3345, 2016.

[85] L. Perfetti, C. Polari, F. Fassi, S. Troisi, V. Baiocchi, S. Del Pizzo, F. Giannone, L. Barazzetti, M. Previtali, and F. Roncoroni, “Fisheye photogrammetry to survey narrow spaces in architecture and a hypogea environment,” *Latest Developments in Reality-Based 3D Surveying and Modelling; MDPI: Basel, Switzerland*, pp. 3–28, 2018.

[86] E. Nocerino, F. Menna, and F. Remondino, “Accuracy of typical photogrammetric networks in cultural heritage 3d modeling projects,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-5, pp. 465–472, 2014.

[87] F. Remondino, E. Nocerino, I. Toschi, and F. Menna, "A critical review of automated photogrammetric processing of large datasets.," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W5, pp. 591–599, 2017.

[88] H. Tong, M. Li, H. Zhang, and C. Zhang, "Blur detection for digital images using wavelet transform," in *2004 IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, pp. 17–20, IEEE, 2004.

[89] Z. Zhang, C. Forster, and D. Scaramuzza, "Active exposure control for robust visual odometry in hdr environments," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3894–3901, IEEE, 2017.

[90] N. Nourani-Vatani and J. Roberts, "Automatic camera exposure control," in *Proceedings of the Australasian Conference on Robotics and Automation 2007*, pp. 1–6, Australian Robotics and Automation Association Inc., 2007.

[91] "Node.js." `https://nodejs.org/en/`. Accessed: 2022-02.

[92] "Google Protocol Buffers." `https://developers.google.com/protocol-buffers`. Accessed: 2022-02.

[93] "Tree.js." `https://threejs.org/`. Accessed: 2022-02.

[94] "Leica HDS7000." `https://www.sccssurvey.co.uk/downloads/hds7000/Leica_HDS7000_brochure.pdf`. Accessed: 2022-02.

[95] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, G. Ranzuglia, *et al.*, "Meshlab: an open-source mesh processing tool.," in *Eurographics Italian Chapter Conference*, vol. 2008, pp. 129–136, 2008.

[96] "Agisoft metashape." `https://www.agisoft.com`. Accessed: 2022-02.

[97] "Agisoft metashape manual." `https://www.agisoft.com/pdf/metashape-pro_1_7_en.pdf`. Accessed: 2022-02.

[98] "Cloudcompare." `https://www.cloudcompare.org`. Accessed: 2022-02.

[99] E. Nocerino, F. Menna, F. Remondino, I. Toschi, and P. Rodríguez-Gonzálvez, "Investigation of indoor and outdoor performance of two portable mobile mapping systems," in *Videometrics, Range Imaging, and Applications XIV*, vol. 10332, p. 103320I, International Society for Optics and Photonics, 2017.

[100] X. Yang, L. Zhou, H. Jiang, Z. Tang, Y. Wang, H. Bao, and G. Zhang, "Mobile3drecon: real-time monocular 3d reconstruction on a mobile phone," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 12, pp. 3446–3456, 2020.

[101] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[102] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[103] E. K. Stathopoulou, R. Battisti, D. Cernea, F. Remondino, and A. Georgopoulos, "Semantically derived geometric constraints for mvs reconstruction of textureless areas," *Remote Sensing*, vol. 13, no. 6, p. 1053, 2021.

[104] G. Singh and A. Mittal, "Various image enhancement techniques-a critical review," *International Journal of Innovation and Scientific Research*, vol. 10, no. 2, pp. 267–274, 2014.

[105] M. Lecca, A. Torresani, and F. Remondino, "Comprehensive evaluation of image enhancement for unsupervised image description and matching," *IET Image Processing*, vol. 14, no. 16, pp. 4329–4339, 2020.

[106] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[107] F. Menna, E. Nocerino, and F. Remondino, "Optical aberrations in underwater photogrammetry with flat and hemispherical dome ports," in *Videometrics, Range Imaging, and Applications XIV*, vol. 10332, p. 1033205, International Society for Optics and Photonics, 2017.

[108] "Agisoft metashape python manual." `https://www.agisoft.com/pdf/metashape_python_api_1_8_0.pdf`. Accessed: 2022-02.

Appendix A

Camera synchronization and calibration

A.1 Triggering

In the current implementation of the system, the acquisition of the stereo image pair is synchronized with software triggers, although hardware triggers are possible. We have measured a maximum synchronization error between the left and the right image of 1 ms. This has been tested recording for several minutes the display of a simple self-built chronometer based on Arduino that can measure up to 1/10 of a millisecond. Figure A.1 shows some of the stereo images of the chronometer acquired over different runs. The chronometer shows the current millisecond digit on the display, which corresponds to a difference of maximum one millisecond between the left and right images. The images were acquired with a shutter speed of half millisecond.

A.2 Modulation transfer function

To enable a more accurate ground sampling distance (GSD) control (Section 4.3.4), we measured the actual modulation transfer function (MTF) of the lens using an ad-hoc test chart [107]. The chart includes photogrammetric targets that allow the relative pose of the camera to be determined with respect to the chart plane and, consequently, a better estimation of the actual

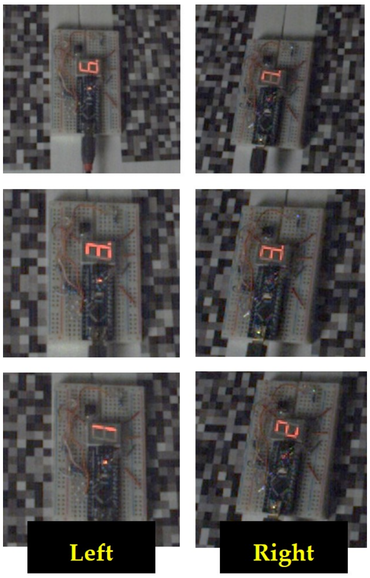
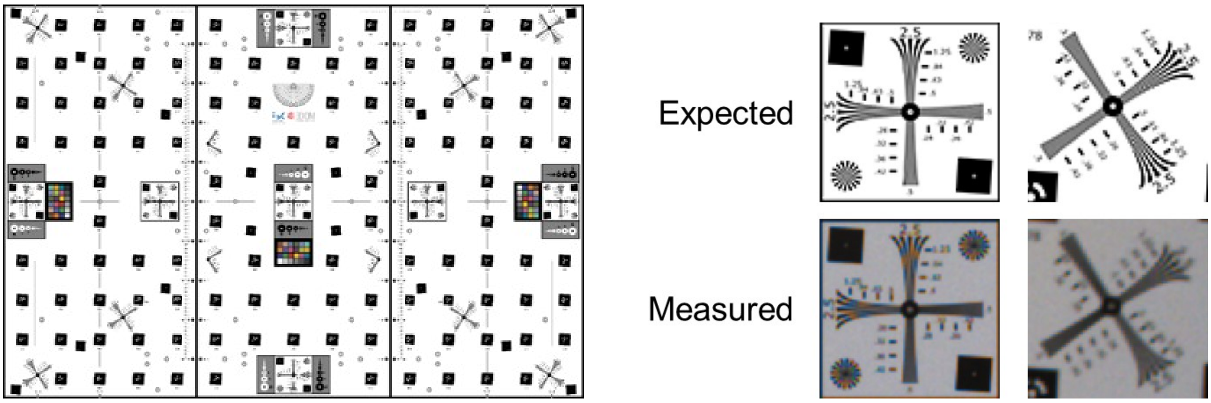


Figure A.1: Synchronisation test and self-built chronometer.

GSD as well as other optical characteristics such as the depth of field. The test chart uses slant-edges according to the ISO 12233 standard. Moreover, it includes resolution wedges along the diagonals with metric scale that allow a direct visual estimation of the limiting resolution of the lens. In our experiment we compared the expected nominal GSD at the measured distance from the chart against the worst of radially and tangentially resolved patterns along the diagonals of the chart as shown in Figure A.2. We estimated a ratio of about 2 considering both left and right cameras.

A.3 Calibration

The calibration of the system involves the estimation of the intrinsic parameters of the cameras and distortion coefficients of the lenses (Section 3.1.1), together with the relative $SE(3)$ transformation (Section 3.1.2) between the stereo cameras. The calibration procedure employed in this thesis relies on the use of coded targets with known coordinates and accuracy, and it is largely based on that described by Nocerino and Menna [75].



(a)

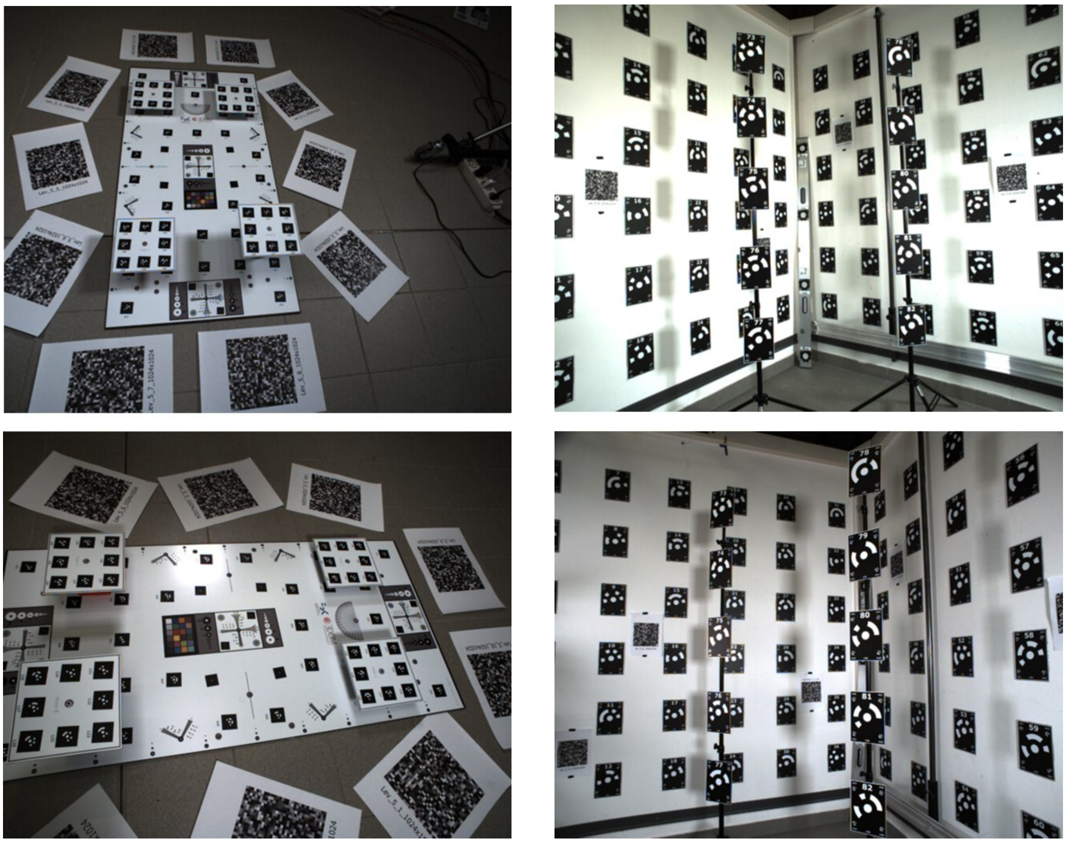
(b)

Figure A.2: (a) Resolution chart used to experimentally estimate the modulation transfer function of the used lenses. (b) Examples of the expected resolution patches (top), at the center (left) and 2/3 of the diagonal (right) against the imaged ones from the left camera (bottom).

The procedure can be summarised as follows:

1. Acquire a sequence of N stereo images of the calibration test field.
2. Orient each image separately using bundle adjustment optimisation with self calibration. The estimation of the intrinsic and distortion parameters is shared among all the images acquired by a single camera. This produces $2N$ camera poses and 2 sets of intrinsic and distortion parameters, one related to the left camera and one related to the right camera.
3. Detect the coded targets in the images and leverage their known distance to impose a metric scale to the estimated camera poses.
4. Express all the right camera poses in terms of the corresponding left ones, convert them to quaternion and translation vectors, and compute their average values among all the corresponding stereo pairs.

The two sets of intrinsic and distortion parameters estimated at step 2, and the average transformation between the stereo cameras estimated at step



(a) (b)

Figure A.3: Some images of the small (a) and big (b) test fields used to calibrate the system. Random patterns were placed alongside the targets to provide a more discriminative texture and improve the number and reliability of automatically-extracted tie points for the system calibration.

4 will represent the final outcome of the system calibration. Steps 2 and 3 are performed in Agisoft Metashape [96], while step 4 is done with an own-developed Python script. Two different calibration fields were used to calibrate the system. The small test field (Figure A.3a) was mostly used when the cameras were focused at close values, while the big one (Figure A.3b) was used in all the other cases. The targets were measured with an expected accuracy of 0.05 millimetres in the small test field, and 0.1 millimetres in the big test field. At every calibration procedure, the test fields are imaged from many positions (more than one hundred images) and

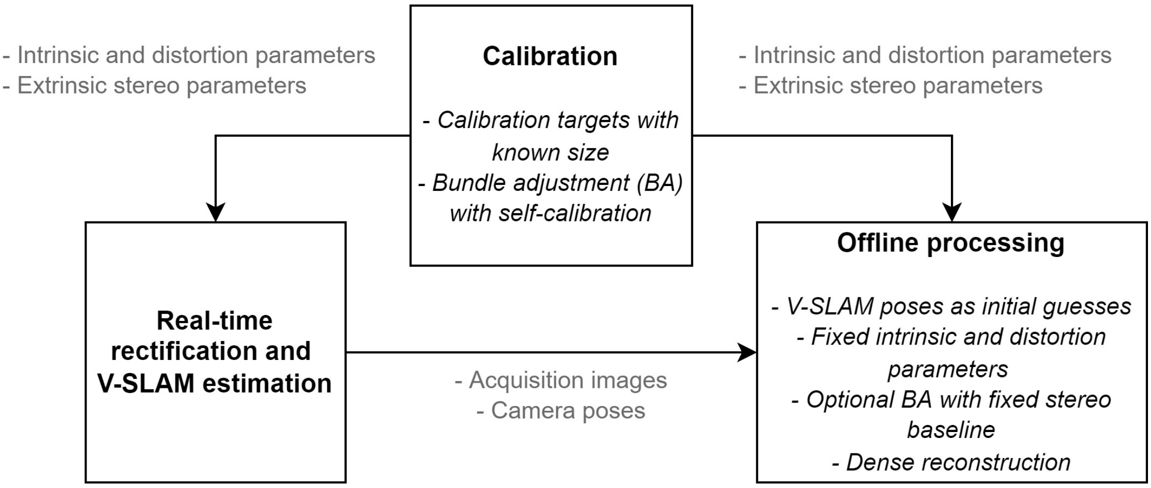

- Intrinsic and distortion parameters
- Extrinsic stereo parameters
Calibration
- Calibration targets with known size
- Bundle adjustment (BA) with self-calibration
- Intrinsic and distortion parameters
- Extrinsic stereo parameters
Real-time rectification and V-SLAM estimation
- Acquisition images
- Camera poses
Offline processing
- V-SLAM poses as initial guesses
- Fixed intrinsic and distortion parameters
- Optional BA with fixed stereo baseline
- Dense reconstruction

Figure A.4: Usage of the system calibration in the various working stages.

orientations (portrait, landscape) to ensure an optimal intersection geometry and reduce the risk of parameter correlations. Random patterns were also placed alongside the targets to ensure an increased and distributed number of stable features during the bundle adjustment iterations.

A.3.1 Usage

The results of the calibration are not only necessary for the real-time operations of the system, but are also exploited later during the offline processing steps. Figure A.4 helps to visualise how the calibration is leveraged in the different working stages. During the field work, the intrinsic and distortion parameters of the cameras, together with their relative $SE(3)$ transformation, are used in real-time to stereo-rectify the images and perform trajectory and map estimations with a metric scale (Sections 4.3.1 and 4.3.2). Successively, the offline processing also makes use of the calibration. At this stage different processing approaches are possible. One possibility is to perform the dense reconstruction directly from the V-SLAM poses; in this case only the calibrated intrinsic and distortion parameters are required. The second option is to use the V-SLAM poses only as initial approximations and carry out an offline bundle adjustment optimisation before computing the dense point cloud. Also in this case the processing uses the calibrated intrinsic and distortion parameters, but additionally the calibrated stereo baseline is required to constraint, during the optimisation, the distance between the input stereo pairs. The offline processing is managed with Agisoft Metashape [97] in this thesis. The import of the V-SLAM poses in Metashape is done through scripting and the Python Application Programming Interface (API) provided by the software [108].

Rectilinear - Big calibration field							
b(mm)	σ_b(mm)	ω(deg)	σ_ω(deg)	ϕ(deg)	σ_ϕ(deg)	κ(deg)	ω_κ(deg)
245.2239	0.3749	-0.0902	0.0113	-22.1633	0.0089	-0.0107	0.0142

Table A.1: Stereo system calibration 1.

Rectilinear - Small calibration field							
b(mm)	σ_b(mm)	ω(deg)	σ_ω(deg)	ϕ(deg)	σ_ϕ(deg)	κ(deg)	ω_κ(deg)
245.3167	0.1099	-0.0935	0.0394	-22.1123	0.0121	-0.0239	0.0341

Table A.2: Stereo system calibration 2.

Rectilinear - Big calibration field							
b(mm)	σ_b(mm)	ω(deg)	σ_ω(deg)	ϕ(deg)	σ_ϕ(deg)	κ(deg)	ω_κ(deg)
244.9946	0.2387	-0.0941	0.0136	-22.4721	0.0079	-0.0120	0.0055

Table A.3: Stereo system calibration 3.

Fisheye - Big calibration field							
b(mm)	σ_b(mm)	ω(deg)	σ_ω(deg)	ϕ(deg)	σ_ϕ(deg)	κ(deg)	ω_κ(deg)
305.928	0.1686	0.40515	0.0126	1.9697	0.0161	0.0424	0.007

Table A.4: Stereo system calibration 4.