SLAM with Laser Profilers for High Definition Mapping in Confined Spaces

Daqian Cheng

CMU-RI-TR-21-37

July 2021

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Howie Choset, *chair*
Michael Kaess
Wei Dong

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my girlfriend Miao for her unconditional love and support.

Abstract

Three-dimensional reconstruction in confined spaces is important for the manufacturing of aircraft wings, the inspection of narrow pipes, the examination of turbine blades, etc. It is also challenging because confined spaces tend to lack a positioning infrastructure. Therefore, a sensor that is capable of performing Simultaneous Localization and Mapping (SLAM) is required. Although there exist a variety of SLAM-capable sensors such as LiDARs and RGB-D sensors, there have been few, if any, sensors for confined spaces reconstruction, because such tasks require sensors that are compact, operate in short-range, and can self-localize.

In this thesis, we propose a sensor framework based on monocular laser profiling for confined spaces. This framework consists of a hardware structure, a software pipeline, and a SLAM method. Sensor prototypes designed using this framework are able to achieve photo-realistic 3D reconstruction in real-time. To generate photo-realistic reconstruction, conventional RGB-D sensors typically rely on multiple-camera suites to separately capture 3D geometry and visual color; e.g., the RealSense D435 uses one stereo camera with a pattern projector for 3D measurement and one monocular RGB camera for color. To minimize sensor size, the proposed framework employs a single camera to achieve photo-realistic reconstruction. This is achieved using our alternating-frame imaging technique which alternately captures color and geometry information in adjacent imaging frames by altering sensor states. A SLAM method tailored to laser profilers is proposed to accurately localize the sensor by tightly fusing laser, camera, and inertial measurements. Additional sensors can also be integrated into the SLAM thanks to its modular factor graph design.

This sensor framework's ability to generalize to different sensor configurations enables it to tackle various confined spaces. In this thesis, we propose two sensor prototypes named Blaser and PipeBlaser, both designed under the framework. For the general confined space setting, the Blaser prototype features a laser-stripe profiler and was designed to be compact and short-range-capable. It boasts a 1-inch minimum sensing range and is more than ten times smaller than Intel RealSense D435, one of the smallest, if not the smallest, commercial SLAM-capable sensor. For confined in-pipe environments, a more specialized prototype named PipeBlaser is designed. It has a laser-ring profiler configuration and can function in 12-inch diameter pipes. These two sensor prototypes exhibit

vastly different configurations but are designed under the same sensor framework with some modifications for corresponding applications.

A comprehensive qualitative and quantitative evaluation was performed on both sensor prototypes in a variety of environments, demonstrating their localization and mapping capability in a real-time fashion. We also compare the sensor system to other state-of-the-art SLAM methods as well as to a popular and capable RGB-D camera.

Acknowledgments

To begin with, I would like to express my gratitude to my supervisor Howie Choset for his insight, advice and support through out this work. I could not have asked for a better advisor who showed unwavering and enthusiastic encouragement and support. In addition to the critical thinking and the rigorous attitude towards research, I also learned from his clear way of expressing ideas and conducting arguments. If only I could also learn his hilarious and appropriate sense of humour, for which I will forever be envious. Finally, I appreciate the few times that Howie "went hard" on me, which have spared me from the same stupidities ever since.

During my master's study, I was fortunate enough to have learned from Lu Li, a scholar of vast knowledge and diverse skill sets. His valuable input not only shaped the work in this thesis but also taught me the first-principles thinking in conducting research and presenting ideas. After more than two years, I have finally discovered his secret or at least I would like to think so: a child-like curiosity about everything and an insatiable appetite for new knowledge. Lu is also the most supportive and caring friend anyone could ever have, and I will always cherish the good times we had.

I would also like to thank my thesis committee, Prof. Michael Kaess and Wei Dong, for their insightful advice and helpful comments.

I am forever thankful for being privileged to work with the bright minds at the Biorobotics Lab and the Robotics Institute at CMU. I would like to express special gratitude to Haowen Shi and Yihe Hua, without whom this thesis would not have been possible. In addition, I would like to thank Michelle Crivella from the Boeing side for her shrewd input into the Blaser project. Finally, I would like to thank every hardworking student and staff member of the Boeing Blaser and the ARPA-E Mapping team.

Funding

This work was supported by the Boeing Strategic University Program and the ARPA-E REPAIR Program. These funding are gratefully acknowledged.

Contents

1 Introduction **1**

2 Related Work **9**
2.1 Laser profilers . . . 10
2.2 RGB-D Cameras and Related SLAM Methods . . . 11
2.3 Monocular Visual-Inertial SLAM . . . 13
2.4 In-Pipe SLAM . . . 14

3 Blaser: Confined Space Scanner **17**
3.1 Hardware Design . . . 17
3.2 Sensor Model . . . 18
3.3 Sensor Calibration . . . 19
3.4 Sensitivity Analysis . . . 20
3.5 Software Framework . . . 22
3.5.1 Alternating-frame imaging . . . 23
3.5.2 Laser stripe detection . . . 25

4 Visual-laser-inertial SLAM **27**
4.1 Front-end . . . 28
4.1.1 Visual front-end . . . 28
4.1.2 Profiling front-end . . . 29
4.1.3 Inertial front-end . . . 29
4.2 Initialization . . . 29
4.3 Sliding-Window-based Factor Graph Formulation . . . 30
4.3.1 Features-on-Laser Depth Residual . . . 32
4.3.2 Feature Reprojection Residual . . . 33
4.3.3 Inertial Measurement Residual . . . 33
4.3.4 Marginalization . . . 33
4.4 Map Appearance Generation . . . 34
4.5 Mapping . . . 34
4.6 Window-to-map Tracking . . . 35
4.7 Solving Non-linear Least Squares . . . 36

5 PipeBlaser: In-pipe Mapping Sensor **37**

5.1 Hardware Design . . . 37
5.2 Sensor Model . . . 40
5.3 Calibration . . . 41
5.4 Sensitivity Analysis . . . 42
5.5 Software Framework . . . 44
5.6 SLAM . . . 46

6 Experimental Results **49**
6.1 Blaser . . . 49
6.1.1 Odometry Accuracy Evaluation . . . 50
6.1.2 3D Reconstruction Evaluation . . . 52
6.1.3 Mapping with External Positioning Aid . . . 54
6.2 PipeBlaser . . . 55
6.2.1 LiDAR 3D Measurement Characterization . . . 56
6.2.2 Profiling Evaluation . . . 57
6.2.3 SLAM Evaluation . . . 59

7 Conclusions **61**

Bibliography **63**

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

1.1 Software of the proposed sensor framework. 3

1.2 An overview of the Blaser sensor. (a), (b) The proposed sensor hardware prototype; (c) hand-held scanning with ground truthing experimental set up; (d) the reconstructed colored point cloud of a keyboard, scanned without external infrastructures. 5

1.3 An overview of the PipeBlaser sensor. (a) A conceptual drawing of the PipeBlaser in a 12-inch diameter pipe; (b) the PipeBlaser sensor prototype; (c) the PipeBlaser prototype in a 16-inch diameter pipe; (d) the reconstructed colored point cloud of a 16-inch diameter pipe. . 6

2.1 (a) Keyence commercial laser profiler capable of high accuracy 3D measurement but without SLAM capability; (b)setup illustration of DAVID Laserscanner [59] where a reference background is used to estimate the 3D laser plane position relative to the camera; (c) 3D reconstruction result achieved in [38] using a motorized gantry to provide pose of the laser profiler. 11

2.2 3D reconstruction results obtained with RGB-D cameras using a volumetric approach [10] (left) and a surfel-based approach [58] (right). . 12

3.1 Hardware design of the Blaser sensor prototype. (a) The exploded-view of the design showing the hardware components; (b) the assembled mechanical design; (c) corresponding pictures of the sensor hardware. 18

3.2 Theory of operation: Laser depth is triangulated by projecting a camera ray out from the camera origin and finding its intersection with the laser plane. 18

3.3 Calibration process and result visualization. (a) shows a camera image with detected checkerboard and fitted laser line (blue). (c) shows 3D laser points (red) and the fitted laser plane, for the user to examine the calibration result. 20

3.4 Illustration of the sensor's sensitivity analysis and sample result plots generated by the sensitivity analysis. (a) Illustration of the definition of depth and elevation angle; (b) Sensitivity analysis with depth, revealing severe sensitivity decrease as depth increases; (c) sensitivity with elevation angle, showing sensitivity in the center direction is slightly higher that off-center; (d) sensitivity with laser leaning angle, showing sensitivity first increase as the leaning angle increases to 45 degrees but then decreases rapidly. 22

3.5 Software framework and data flow visualization. 23

3.6 Timing sequence of the alternating-frame imaging method. The camera shutter alternates between long and short exposure "on" times, keeping frame duration constant for all frames. The laser's state toggles synchronously with the alternating exposure. This approach enables the monocular sensor to produce two sequence of images in an interleaving fashion that provide RGB and depth information. 25

3.7 Alternating-frame camera driver software diagram. Due to the lack of camera status output, a receiving buffer outputs a signal each time it receives raw data from the camera. This signal triggers the sensor configuration alternation. 25

3.8 Visualization of the laser detection process. The top left image shows the input image captured by the camera. An thresholding operation in the HSV domain segments out the region of interest, shown in the top right, that contains the laser stripe. For each pixel column, the procedure first find a segment of pixels with high intensities and then compute their center-of-mass in terms of intensity. In the bottom left image, the yellow points show the bounds of the per-column pixel segment, and the green points are the center-of-mass. 26

4.1 Illustration of the sliding window-based visual-laser-inertial SLAM. The sliding window is consisted of several keyframe poses, features observed by the keyframes, laser point cloud observed in the time span of the sliding window, adjacent laser point cloud in previously built map (if revisited), and inertial measurements. 31

4.2 The factor graph formulation. The SLAM problem consists of five types of factors: feature reprojection factor, inertial factor, marginalization factor, feature-laser association factor, and window-to-map tracking factor. 32

4.3 Illustration of the color estimation of a laser pixel using projective association. 34

5.1 A rendered image of the PipeBlaser prototype mechanical design. The prototype consists of multiple sensors, including a camera, an IMU, a laser-ring projector, and a LiDAR. An on-board computer enables online SLAM and data recording for post-analysis. A four-wheel-drive vehicle platform drives the sensor along pipes. 38

5.2 Hardware component diagram of the PipeBlaser prototype. 39

5.3 Theory of operation: for each laser pixel, a 3D laser point is triangulated by projecting a line-of-sight ray of the pixel and finding its intersection with the laser plane. 40

5.4 An illustration of the camera-laser extrinsics calibration process and the visualized result of fitting a 3D laser plane to the collected sample laser points. (a) The calibration uses images of a checkerboard which intersects with the laser disk, forming a straight laser stripe on the checkerboard. (b) A sample raw image captured by the fisheye camera. (c) The laser plane was fitted to more than six thousand 3D sample points, resulting in 0.9 mm average point-to-plane error. 41

5.5 Theoretical sensitivity analysis of the laser-ring profiler. The top figure shows a 3D plot of sensitivity with pipe diameter and baseline length. The result on the bottom left shows the sensitivity with pipe diameter given various baseline lengths. This can be used to study a sensor's performance limitations when using it in different pipe sizes. The result on the bottom right shows the sensitivity with baseline length given different pipe diameters. This function can help determine the optimal sensor design for a certain application. 43

5.6 PipeBlaser software framework for localization and photo-realistic mapping. 44

5.7 45

5.8 Process of generating the region-of-interest image mask. The software first rejects the image border area exceeding a certain field-of-view angle and then asks the user to manually label the laser mounting pole. 46

5.9 The laser ring detection process. (a) shows the original image captured by the camera. In (b), the cyan region is the HSV mask, and the blue region is the region-of-interest (ROI) mask. (c) visualizes the radial laser point detection. In (d), the green pixels are the extracted laser points. 47

5.10 Factor graph formulation of in-pipe SLAM using PipeBlaser. 47

6.1 Trajectories in top-down view of the proposed SLAM methods, VINS-Mono and ground truth and the associated translational and rotational errors. The background color of error plots indicates different passes in the zigzag trajectory. The top portion in the translational error plot is rescaled to accomodate the large error of VINS-Mono. 51
6.2 Intermediate mapping result after each scan pass. There were in total six passes to gradually complete the scan. The proposed VLI-SLAM was able to maintain mapping consistency under the repeated scanning. 53
6.3 Comparison of point cloud reconstruction. (a) is a photograph of the scanned scene. (d) and (g) are the photo-realistically and spatially colored reconstruction results by RealSense. Reconstructed using the proposed sensor, (b) and (c) show results using VLI-Odom, and (e), (f) and (h) are with VLI-SLAM. (i) and (j) are the sectional views of (g) and (h) respectively with the red dashed lines as the cutting planes. 54
6.4 Reconstruction using the proposed system of (a) a face mask, (b) a multimeter, (c) a industrial aerospace part, and (d) a toy car. 55
6.5 Using Blaser with a robot manipulator to scan a prostate model. (a) Experiment setup. The Blaser was mounted onto the end-effector of the manipulator, controlled by a 3D mouse. (b) The photo-realistic 3D reconstruction. When the Blaser uses external pose estimation to replace SLAM, it can achieve drift-free 3D reconstruction. 56
6.6 Accumulated Livox point cloud in a 12-inch diameter pipe while the Lidar is stationary. 57
6.7 Segmented pipe cross-section of the raw point cloud obtained from Livox LiDAR with a ground truth diameter reference. 58
6.8 A sample profiling image captured in a 12-inch diameter PVC pipe (ground truth diameter is 11.83 inches) and a cross-section profile generated using the image. 58
6.9 The reconstructed 3D map of a 12-inch diameter PVC pipe. 59
6.10 The reconstructed 3D map of a 16-inch diameter mock-up steel pipe. 60

List of Tables

6.1 Absolute localization errors and Drift rates 52
6.2 Mapping RMSE statistics 52
6.3 In-pipe localization error evaluation 60

Chapter 1

Introduction

Three-dimensional reconstruction is a fundamental problem in robotics and computer vision. Various sensor systems with wide-ranging capabilities (e.g., range and resolution), such as laser-stripe triangulators, RGB-D cameras, and LiDARs, and corresponding algorithms [7, 35, 45, 61] have made accurate 3D scanning possible in many types of spaces (e.g., indoor [8, 34, 58], outdoor [9, 39, 61], underwater [42, 43, 47], etc), revolutionizing many civil and industrial fields. These sensor hardware and software systems, in the authors' view, operate in wide-open spaces and are not well-suited, by design, for confined space operation. In fact, few, if any, SLAM sensor systems for 3D reconstruction have been developed for confined space operation. Such systems would be of great use for inspection applications, even more so than in open spaces where abundant choices of external positioning infrastructure, such as motion capture cameras and total stations, can be employed to eliminate the need for SLAM capability; confined spaces, on the other hand, tend to lack a positioning device. The challenge in building a sensor for confined space 3D reconstruction comes from the following constraints: the sensor must be 1) compact to fit into tight spaces; 2) able to operate at short-range; 3) able to perform SLAM due to the lack of positioning infrastructure.

Current commercial off-the-shelf (COTS) sensors for 3D reconstruction are either too large or dependent on external positioning infrastructure (e.g., robotic manipulators, motion-capture cameras, etc). Kinect (Microsoft, Redmond, WA, USA) and RealSense (Intel, Santa Clara, CA, USA) are two popular RGB-D camera families with

self-localization capability, but even the smallest model, RealSense D435, measures $90 \times 25 \times 25$ mm in size and has a minimum sensing range of 105 mm. Popular for mobile robots and autonomous vehicles, LiDARs boasts long range 3D measurement but are typically even larger than RGB-D cameras, rendering them unsuitable for confined spaces. On the other hand, highly accurate and compact laser profilers such as optoNCDT (Micro-Epsilon, Raleigh, NC, USA) can achieve small sensor foot prints as well as short-range measurement capabilities, but they only generate per-frame 3D measurements and do not have localization capability; to perform 3D mapping, external positioning devices are required. [30] introduced an ultra-compact 3D measurement sensor but also lacked self-localization capability.

In this thesis, we propose a sensor framework for high accuracy photo-realistic mapping in confined space. This sensor framework consists of a hardware structure design, a software pipeline, and a Simultaneous Localization and Mapping (SLAM) method. We adopt laser profiling as the 3D geometric measurement approach. Laser profilers are laser displacement sensors that collect depth data across a projected laser line using camera-laser triangulation. Compared to other 3D measuring method such as time-of-flight (ToF) or 2D pattern triangulation, laser profiling only measures depth on a 1-dimensional line but with significantly higher accuracy which is usually from sub-millimeter to micrometer level.

Based on the laser profiling approach, the sensor hardware is comprised of a monocular color camera, a laser projector, and an additional Inertial Measurement Unit (IMU) which helps with localization at almost no cost in sensor size. Additional sensors can be added to further aid SLAM by formulating additional factors in the factor graph. One challenge on the monocular sensor setup is to perform photo-realistic 3D reconstruction, which requires the sensor to capture two types of data: 3D geometry and visual color for adding appearance to the 3D reconstruction. The acquisition of these two data is typically achieved by using multiple cameras to separately capture each data, e.g., Intel RealSense D435 uses one stereo camera with a pattern projector for 3D measurement and one monocular RGB camera for color. In the proposed sensor framework, we use the single camera to capture both information using the alternating-frame imaging technique, where the camera alternately captures images for laser profiling and images for visual coloring by altering camera exposure and laser on/off state. This technique reduces the number of cameras needed for

photo-realistic reconstruction, thus minimizing the sensor's size and cost.

The software of the sensor framework consists of two parts as shown in Figure 1.1: a pipeline of photo-realistic 3D reconstruction and supporting software. In the reconstruction pipeline, software components including the alternating-frame controller, sensor data pre-processing, and SLAM progressively processes sensor measurements and outputs 3D reconstruction as well as sensor pose estimation. Supporting software mainly contains a calibration tool and a software that analyzes sensor sensitivity given the camera and the laser-camera placement configuration; this software can also generate the configuration with the optimal sensitivity.

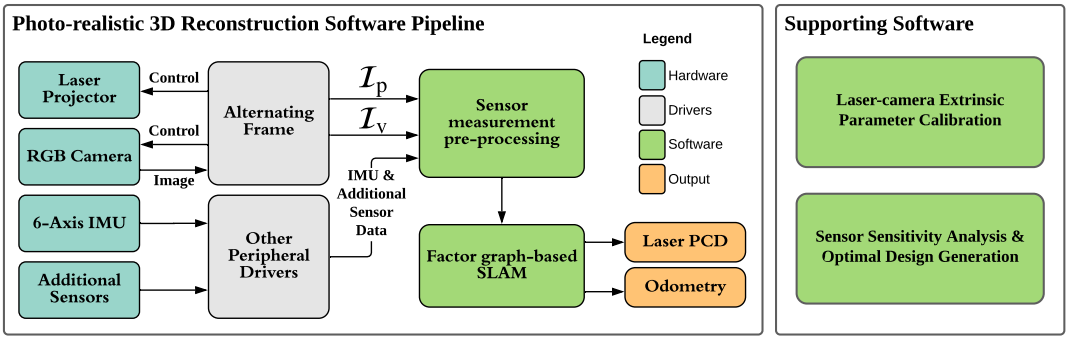

Figure 1.1: Software of the proposed sensor framework.

At the center of the proposed software solution is the SLAM method tailored for monocular laser profilers. Localization accuracy often determines reconstruction quality since individual laser scans are registered to a global reference frame according to the localization estimation. Since monocular SLAM suffers from scale ambiguity, monocular visual-inertial (VI) sensor setup is the smallest sensor-suite that the community uses to perform SLAM with metric scale. VI-SLAM methods have achieved promising results and are nowadays widely used in mobile robots, smartphone applications, and VR & AR. However, sensor motion in confined spaces is often much slower and IMU measurements are much less excited, which undermines metric scale estimation and localization accuracy. Therefore, we proposed a SLAM method designed for laser profilers. In the SLAM method, the laser scans not only are stitched together to generate a point cloud map but also help estimate the metric scale of the SLAM. The accurate but dimensionally degenerated laser-line scans are associated with visual features to effectively recover the metric scale, resulting in

low-drift localization. Further more, a window-to-map tracking component aligns the recent laser scans in the sliding-window to the historic map. In this way, mapping consistency can be maintained under back-and-forth re-scanning motion. Mapping consistency is important to the map's visual quality and desirable for real-world scanning applications.

The proposed sensor framework is able to generalize to various sensor configurations in order to adapt to different real-world confined spaces. When adapting the framework to a new sensor configuration, many of the software components can be reused with little modification. In this thesis, we put forward two sensor prototypes that each addresses a confined-space mapping challenge.

The first prototype named Blaser [5, 6] targets general confined spaces. The prototype is designed to be as compact as possible while managing short-range sensing. The Blaser prototype is equipped with a laser-stripe projector, a miniature camera, and an Micro-Electr-Mechanical System (MEMS) IMU. It achieves a size of $27 \times 15 \times 10$ mm, 14 times smaller than RealSense D435 in volume, and a sensing range of 20-150 mm. Figure 1.2 shows the proposed sensor hardware as well as a hand-held reconstruction result of a keyboard. The sensor size is mainly constrained by the size of the camera and the laser diode as well as the baseline length between the camera and the laser. Adequate baseline length is critical for the sensor's sensitivity.

The second prototype named PipeBlaser is designed for confined pipe environments, specifically for 12-inch to 16-inch diameter pipes. The geometric integrity of pipes are vital for the safety of operations. For natural gas pipes, the failure of pipes can cause explosion and result in severe casualties [24, 54]. In-pipe 3D reconstruction is a powerful tool for the analysis of pipe geometric integrity, such as pipe diameter, geometric deformation, etc. In addition, photo-realistic appearance of 3D reconstruction is also valuable since it can be used to detect defects including finer cracks and corrosion. Although in-pipe 3D mapping technology exists, they suffer from limitations. Many methods rely on visual Structure-from-motion (SfM) which generates low-definition map. Although some methods use laser profilers with high accuracy, they have limitations regarding localization due to the use of external pose estimation aid, wheel encoders (which limit motion estimation to 1-dimensional), and visual SLAM with strong assumptions on pipe diameter. We have not seen a technology that generates high-definition map and performs 6 Degree-of-freedom

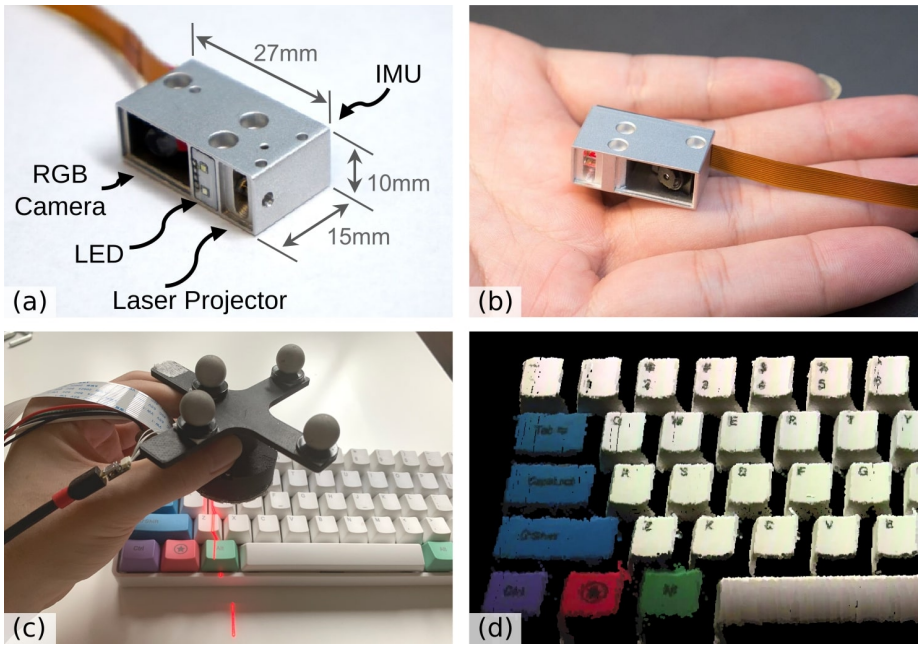

Figure 1.2: An overview of the Blaser sensor. (a), (b) The proposed sensor hardware prototype; (c) hand-held scanning with ground truthing experimental set up; (d) the reconstructed colored point cloud of a keyboard, scanned without external infrastructures.

(DoF) self-localization without making assumption on the pipe diameter.

In order to scan the cylindrical inner surface of pipes, the PipeBlaser employed a laser-ring projector in tandem with a fisheye camera. The camera faces the axial direction of the pipe and observes the entire laser ring to perform triangulation. The projector mounted in front of the camera projects a laser-ring parallel to the radial plane of the pipe, which allows the sensor to scan the pipe's cross sections. An LED array is added for active illumination inside dark pipes. The sensor uses the proposed SLAM method to accurately estimate 6 DoF pose and stitches the laser scans into a map accordingly. Figure 1.3 presents the prototype as well as the 3D mapping result of a 16-inch diameter pipe.

Extensive experiments were performed on both sensor prototypes. The Blaser sensor showed higher localization accuracy with the proposed SLAM method compared to a state-of-the-art VI-SLAM method. It also demonstrated the SLAM framework's ability to maintain mapping consistency under repeated re-scanning, and displayed its

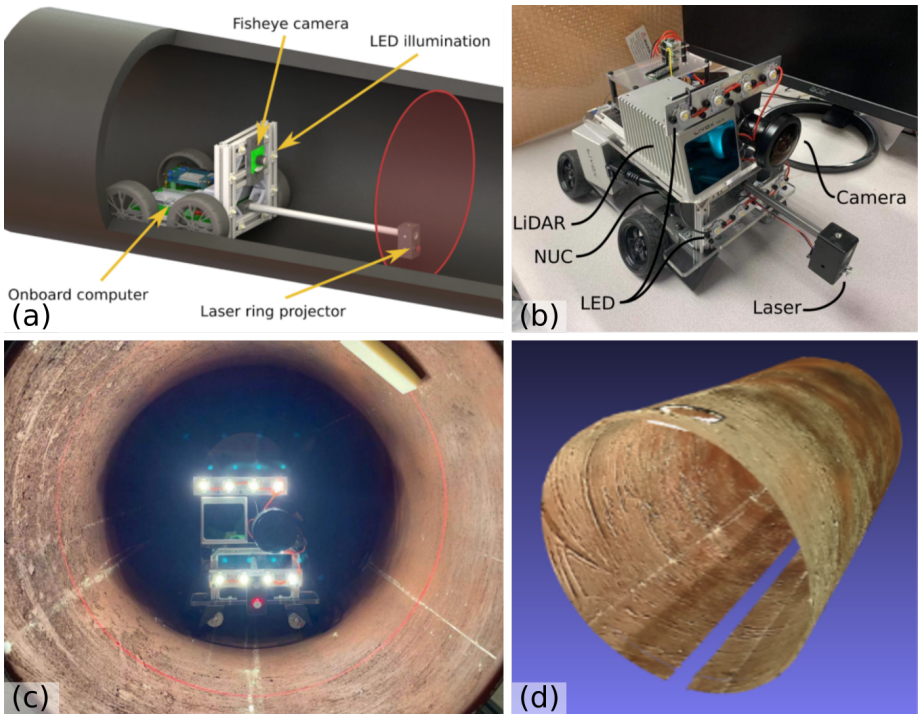

Figure 1.3: An overview of the PipeBlaser sensor. (a) A conceptual drawing of the PipeBlaser in a 12-inch diameter pipe; (b) the PipeBlaser sensor prototype; (c) the PipeBlaser prototype in a 16-inch diameter pipe; (d) the reconstructed colored point cloud of a 16-inch diameter pipe.

superior reconstruction quality compared to a COTS RGB-D camera. The PipeBlaser prototype was evaluated in a 12-inch and a 16-inch diameter pipe, exhibiting photo-realistic mapping and satisfactory localization accuracy.

The remainder of this thesis is structured as follows. Chapter 2 provides an overview on the previous works in related fields including laser profilers, SLAM with RGB-D cameras, monocular visual-inertial SLAM methods, and in-pipe SLAM methods. Chapter 3 describes the Blaser prototype in terms of hardware design, sensor model, and software pipeline. The proposed SLAM method for laser profilers is described in detail in Chapter 4. Chapter 5 discusses the PipeBlaser prototype in a structure similar to Chapter 3. Because the two prototype are designed under the same framework, some similarities are omitted for brevity while differences are highlighted. Modification to the SLAM method for in-pipe application are also described. Chapter 6 offers experimental results on the Blaser scanning various

household and industrial objects and the PipeBlaser performing mapping in two pipe environments. Finally, Chapter 7 reviews the contributions of this thesis.

Chapter 2

Related Work

In this chapter, we review a few types of 3D measurement sensors and corresponding SLAM methods if exist.

Since the proposed sensor framework is based on laser profiling, we first introduce the literature of laser profilers and related mapping work in Section 2.1. To the best of our knowledge, there has not been a SLAM method developed for laser profilers, and almost all related mapping work rely on external positioning aids to integrate the laser scans.

Prior to this work, RGB-D sensors are considered by the community the smallest SLAM-capable and low-cost sensor for accurate dense mapping. For this reason, we also compare the proposed sensor with one RGB-D camera in the experiments. Section 2.2 provides an overview of the 3D measurement technologies used in RGB-D cameras and a number of related SLAM methods. Since these cameras are able to capture depth data of the entire camera field-of-view, SLAM can be achieved by aligning point cloud frames to the previously built map. These SLAM methods, however, cannot directly apply to the proposed sensor framework, which only measures depth on a line instead of over the entire FoV.

The proposed sensor theoretically can achieve localization with metric scale by performing visual-inertial SLAM (VI-SLAM) using the onboard camera and IMU. However, the performance is usually poor since the IMU cannot be sufficiently excited from the slow sensor motion in confined space. In spite of this, the proposed SLAM method is inspired by many visual and inertial data processing methods introduced

in the VI-SLAM literature. For this reason, we provide an overview of VI-SLAM methods in Section 2.3.

2.1 Laser profilers

A laser profiler typically consist of a camera and a laser-stripe projector. It projects a laser-stripe onto the scanned surface and uses the camera to see the visible laser stripe and triangulates it into 3D space. Therefore, this type of sensor is also known as laser triangulators.

Active laser-stripe triangulation has been one of the mainstream 3D scanning approaches for decades [51]. Thanks to the simple hardware design and inexpensive components, laser-stripe triangulation is a popular choice for low-cost 3D scanning systems such as the DAVID Laserscanner [59]. Many high accuracy profilers such as Keyence Laser Profiler (Keyence Corporation, Osaka, Japan) and metallic surface scanners [14] also adopt laser-stripe triangulation due to its high accuracy and relative insensitivity to illumination compared to structured light.

There has been extensive work dedicated to reconstructing 3D models using laser profilers or laser-stripe triangulation [7]. However, positioning devices or localization aids are often needed to register individual scans. [38] uses a motorized gantry to move the scanner in 3D space, associating each scan with a pose given by the gantry. [59, 60] utilize a different triangulation method, where the camera is fixed while the user moves the laser-stripe projector to scan the object like with a paint brush; a 3D reference marker board is placed behind the object to help perform triangulation. [43] performs under water 3D mapping using a laser profiler, but a Doppler Velocity Log (DVL) together with other sensors are used to perform dead reckoning. There has been little work that focused on localization using laser-stripe scanners alone to enable infrastructure-free capability. This task is challenging because of the 3D measurement characteristic of a laser profiler: it only generates 3D information on the intersection of a 3D sheet of plane with the real-world surface. Therefore, there is generally insufficient correlation between adjacent measurement frames under motion, making it difficult to estimate frame-to-frame 6 Degree-of-freedom (DoF) motion.

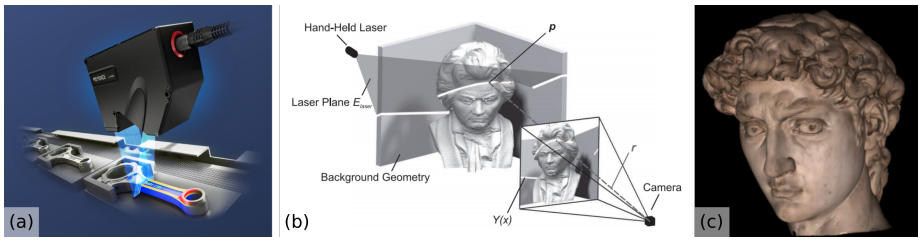

Figure 2.1: (a) Keyence commercial laser profiler capable of high accuracy 3D measurement but without SLAM capability; (b)setup illustration of DAVID Laserscanner [59] where a reference background is used to estimate the 3D laser plane position relative to the camera; (c) 3D reconstruction result achieved in [38] using a motorized gantry to provide pose of the laser profiler.

2.2 RGB-D Cameras and Related SLAM Methods

RGB-D cameras nowadays are extremely popular choices for 3D reconstruction thanks to their low costs and relative compactness. They are called RGB-D cameras since each pixel of the camera image contains a RGB color and a depth measurement. Structured light and time-of-flight are two core technologies behind today's RGB-D cameras. Structured light scanners project 2D invisible patterns of infrared (IR) light onto the scanned surface and use another onboard IR camera to see and triangulate the light pattern [19, 62, 63]. Time-of-flight sensors obtain depth of each pixel by measuring the travel time of emitted light signals. Intel RealSense and Microsoft Kinect are two popular and relatively low-cost RGB-D camera families and are widely used in the development of RGB-D SLAM methods in the literature.

A number of RGB-D SLAM algorithms with promising results have emerged in the past decade, including surfel-based [31, 58] methods and volumetric [8, 45, 57]. A surfel is a surface element and can be viewed as a 3D point with attributes including normal direction, size, color, etc. Surfel-based methods directly represent the world with surfels generated from RGB-D images. Point cloud alignment are used to align new frames to the surfel-map and adjacent surfels are fused together to limit the grow of number of surfels. On the other hand, volumetric methods employ discretized voxel-grid representations of the world, where each voxel stores its Signed Distance Function (SDF) value and other attributes such as color. The SDF value is defined as

the signed distance from the voxel to the nearest object surface being reconstructed. The interior and exterior voxels of the object will store negative and positive SDF values respectively. Therefore, the surface itself can be reconstructed by finding the zero-crossing of the SDF. Localization can be performed via point cloud alignment. [45] proposes a frame-to-map tracking approach where new measurement frames are aligned to virtual frames generated by ray casting the map into the estimated camera pose. Compared to the frame-to-frame tracking approach where each new image frame is tracked against the previous frame, the frame-to-map tracking approach can maintain mapping consistency. Since the predefined volume of the voxel-grid bounds the size of the map, voxel hierarchies [17, 50] and voxel hashing [10, 34] are proposed to allow flexible and fast memory allocation and access, improving the capability of volumetric approach. Both approaches have achieved promising reconstruction results as shown in Figure 2.2.

Figure 2.2: 3D reconstruction results obtained with RGB-D cameras using a volumetric approach [10] (left) and a surfel-based approach [58] (right).

Different from laser profilers, the RGB-D point cloud measurement frames generally ensures enough overlap between adjacent frames, allowing 6 DoF motion estimation via point cloud alignment. Since the laser profilers only measures depth along a single line, these RGB-D SLAM methods cannot be directly applied to laser-stripe triangulators. Nevertheless, SLAM methods developed for RGB-D cameras provide inspiration to this thesis. Inspired by the frame-to-map tracking approach in [45], this thesis proposes a window-to-map tracking approach, described in Section 4.6, where a sliding window of recent laser scans are aligned to the historic map to maintain mapping consistency.

2.3 Monocular Visual-Inertial SLAM

Heavily relying on the camera and the IMU on-board the laser profiler, the proposed SLAM method shares similarities with monocular visual-inertial SLAM. Visual SLAM has two main approaches. The indirect approach [4, 48] , also known as the feature-based approach, first track visual feature points across input images and then estimate camera motion based on the motion of feature points. On the other hand, the direct approach [12, 13] directly utilizes pixel intensities instead of converting the image into an intermediate feature representation. It estimate the motion that minimizes difference in pixel intensities across different views of the same portion of the scene [3]. Since the direct approach does not need to extract features but rely on image gradient instead, it functions well in feature-sparse or feature-less environments. However, one drawback is its assumption that the environment's brightness remains constant when observed across different views, which makes it difficult to handle changing illumination. On the other hand, feature-based methods can typically handle larger motion by performing feature matching. In this thesis, feature-based approach is preferred since the active illumination on the sensors violates the brightness constancy assumption mandated by direct methods.

A key problem with monocular SLAM is that it is not able to estimate the scale and is only able to estimate the visual structure up to an unknown scale factor. This problem is recognized as the scale ambiguity problem. To overcome this issue, an IMU is often incorporated to recover the metric scale [37, 48]. Based on the fusion method of the inertial and the visual information, visual-inertial SLAM methods can be classified as loosely- and tightly-coupled methods. The loosely-coupled approach uses two estimators to separately process visual and inertial measurements and then fuses the two estimations using filtering [18]. The tightly-coupled method jointly optimize both measurements using one estimator. The tightly-coupled approach is often more accurate and robust, while the loose-coupled method is generally computationally efficient, making it favorable for robots with limited computation power such as small unmanned aerial vehicles.

Although VI-SLAM is theoretically able to estimate the scale, we find that with the proposed sensors the scale estimation is often significantly incorrect. The reason behind this issue is two-fold: a) to achieve compact sensor sizes, we employed small,

low-cost MEMS IMUs, which suffer from high measurement noises; b) the slow sensor motion in confined spaces do not fully excite the IMU. Therefore, in the proposed SLAM method, we mainly rely on the laser information instead of inertial data to recover scale. In spite of this, we find inertial data useful in other aspects of localization. The IMU by itself is able to accurately estimate the roll and the pitch angle, making these quantities directly observable for the SLAM problem. Furthermore, the IMU can briefly handle visually degenerated situations where the camera encounters feature-less environments. For these benefits, the proposed SLAM method still incorporates an IMU and fuses visual feature, inertial data, and laser measurements. Since the computation load of tightly-coupled approach is usually acceptable to desktop computers used in this thesis, we adopts the tightly-coupled visual-inertial fusion approach for its superior accuracy.

2.4 In-Pipe SLAM

In-pipe 3D mapping is a valuable tool for the assessment of pipeline's integrity. Many common pipeline problems including corrosion, cracks, and distortion can be determined using the map. There are two main approaches for in-pipe dense 3D reconstruction: visual Structure-from-motion (SfM) [20, 22, 27, 29] and laser-ring profiler [2, 21, 53, 55]. It should be acknowledged that other approaches also exist for specific pipelines, such as pipe-profiling sonar is often employed for sewer pipeline [11, 46], but they are beyond the scope of this thesis.

Visual SfM approach is able to generate dense 3D colored map. Since cameras are now small enough to fit in narrow pipes, the visual SfM is the only mapping option for pipes where a laser profiler cannot fit. One extreme example [20] used a Scanning Fiber Endoscope (SFE), a extremely small imaging sensor, to perform SFM. Since the SFE was merely 1.2 mm in diameter, the system could function in 3-30 mm diameter pipes, and the authors demonstrated the sensor's ability by scanning a 7 mm diameter threaded hole. One challenge for the SfM approach is the scale ambiguity issue of monocular SfM described in Section 2.3. There exist various method to overcome this issue. In some works the pipe diameter is assumed to be known and cylindrical shapes are fitted to the 3D map [29]. [27] extended this idea by fitting conic shape to the map to account for scale drift. Similar to this thesis,

[22] used two laser-dot triangulators to estimate the pipe diameter and constrain the visual structure scale. Due to the sparsity of the 3D feature map, 3D cylindrical models were fitted to the map and texture appearance was mapped onto the model. The major disadvantage of visual SfM compared to laser profiling is with mapping quality. Visual maps are usually sparser and less accurate geometrically than maps obtained with laser profiling.

Laser-ring profilers are widely used when the geometric shape is of concern. Pipe wall thickness and geometric shape distortion can both be identified from cross-section scans. This type of profilers typically consists of a omnidirectional laser-ring project and a fisheye or a catadioptric camera. By optimizing the sensor's configuration, this type of sensor can often reach sub-millimeter 3D measurement error. To register individual scans into a 3D map, various methods have been proposed to measure the sensor's pose. [2] used an IMU to maintain the robot's orientation and wheel odometers to measure displacement along the pipe. This odometry method was able to reach 0.03% drift rate but was limited to straight pipes and was vulnerable to wheel slippage. An external stationary ranging station was employed in [53] to measure the robot's 6 DoF pose using three laser-dot range finders and a camera. In addition to wheel odometry, [21] added another RGB camera for point cloud map coloring. Similar to our proposed SLAM method, [55] adopted visual odometry but assumes a known pipe diameter for the scale ambiguity issue. By tightly-coupling laser depth information with visual data, our SLAM method makes no assumption regarding the environment and can function in unknown-diameter or non-cylindrical pipes.

Chapter 3

Blaser: Confined Space Scanner

The Blaser sensor prototype is designed to be compact to fulfill its application in general confined spaces. This chapter discusses the hardware design, sensor model, supporting software including calibration and sensitivity analysis, and the sensor software framework for the mapping task.

3.1 Hardware Design

The proposed scanner hardware consists of an RGB CMOS camera, a MEMS-based 6-axis accelerometer and gyroscope, and a laser-stripe projector. The camera is equipped with a wide-angle lens with an field of view angle of 160 degrees. A single laser stripe pattern is created by refracting a thin laser beam through a cylindrical lens. This laser stripe is projected to the region within the camera's field of view. There is also a white LED used to reduce motion blur and for color image illumination. Figure 3.1 presents the detailed mechanical design. Since the laser stripe in the image interferes with SLAM, the camera is required to capture images observing the laser stripe as well as images without seeing the laser stripe. For this reason, we propose the alternating-frame imaging technique where the camera alternately capture two types of images with two laser and exposure settings. To enable this technique, the red laser stripe can be toggled on/off in synchronization with our image shutter trigger. Section 3.5.1 describes the detailed motivation and method of the alternating frame imaging approach.

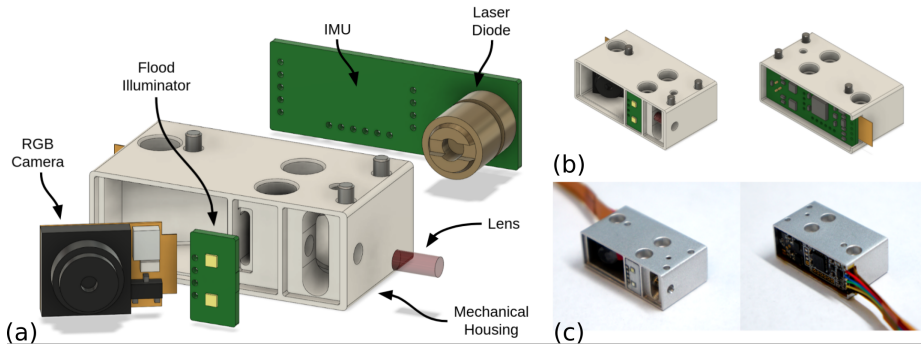

Figure 3.1: Hardware design of the Blaser sensor prototype. (a) The exploded-view of the design showing the hardware components; (b) the assembled mechanical design; (c) corresponding pictures of the sensor hardware.

3.2 Sensor Model

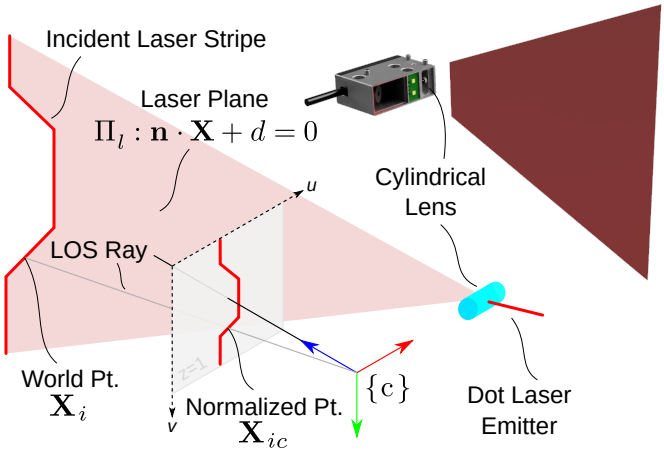

Figure 3.2: Theory of operation: Laser depth is triangulated by projecting a camera ray out from the camera origin and finding its intersection with the laser plane.

3D points on the laser stripe are recovered from 2D images using triangulation. We model the projected sheet of laser light as a plane $\Pi_l : \mathbf{n} \cdot \mathbf{X} + d = 0$ in 3D space, where $\mathbf{n}$ is the normal direction ($\|\mathbf{n}\| = 1$), $\mathbf{X}$ is any point on the plane, and d is a scalar parameter. The laser plane intersects with the physical world and forms a visible laser stripe. The observed 2D laser stripe on the image is discretized into laser pixels, and each laser pixel observation $\mathbf{x}_i \in \mathcal{R}^2$ corresponds to a 3D point $\mathbf{X}_i \in \mathcal{R}^3$

on the scanned surface in the 3D space, whose position can be estimated using triangulation. This triangulation operation involves solving a ray-plane intersection problem illustrated in Figure 3.2. In the figure, the normalized image plane is defined as the plane $z = 1$ in the camera frame. The triangulation is described in (3.1), where $\mathbf{X}_i$ denotes the triangulated 3D point and $\pi_c^{-1}(\cdot)$ denotes the back projection function that projects a pixel position onto the normalized image plane. The 2D laser pixel $\mathbf{x}_i$ is first projected to the normalized image plane, and then a *Line-of-Sight* (LOS) ray is cast from the camera optical origin C through the normalized image point. Finally the corresponding 3D point position $\mathbf{X}_i$ on the incident laser stripe is inferred by computing the intersection between the LOS ray and laser plane Π_l:

$$\mathbf{X}_i = \frac{-d}{\mathbf{n} \cdot \pi_c^{-1}(\mathbf{x}_i)} \pi_c^{-1}(\mathbf{x}_i) \tag{3.1}$$

3.3 Sensor Calibration

Accurate calibration of all sensor components is a crucial prerequisite to accurate SLAM. There are three groups of intrinsic and extrinsic parameters that need calibration: camera intrinsic parameters, camera-IMU extrinsic relative pose described by a Euclidean transformation, and the 3D position of the laser plane relative to the camera.

To accommodate the relative large image distortion from the wide-angle lens, the Kannala-Brandt camera model [28] is employed to model the image projection. The Mei model [44] and the pinhole model, two other popular camera models, were also tested, where the Mei model exhibited similar performance to the Kannala-Brandt model and the pinhole model resulted in significantly larger reprojection error. The camera model parameters were calibrated using the CamOdoCal tool [25]. The extrinsic transformation between the camera and the IMU was calibrated using the Kalibr tool [49].

A tool for calibrating the laser plane's 3D position Π_l was custom-developed. The calibration process, visualized in Figure 3.3, uses images of a known-sized checkerboard where the checkerboard is fully in the camera's view and the laser stripe is projected onto the checkerboard.

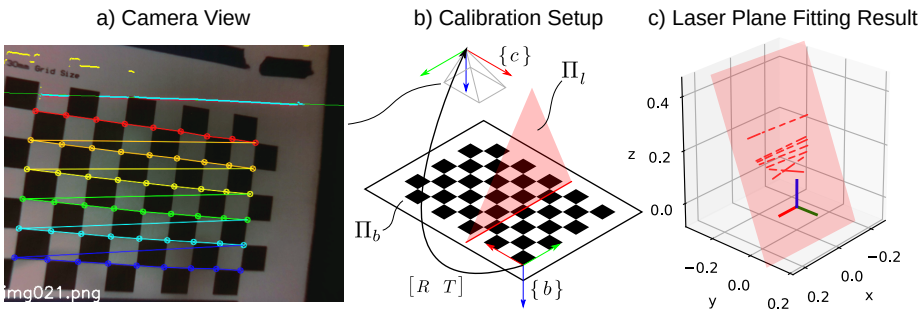

Figure 3.3: Calibration process and result visualization. (a) shows a camera image with detected checkerboard and fitted laser line (blue). (c) shows 3D laser points (red) and the fitted laser plane, for the user to examine the calibration result.

The process of this calibration is described in Algorithm 1. For each input image, undistortion operation is first applied using the pre-calibrated camera intrinsic parameter. Candidate laser pixels are then detected in the undistorted image. Since these laser pixels should form a straight line on the image, a 2D line is fitted to the candidate pixels using Random Sample Consensus (RANSAC) and outlier pixels are discarded. To ensure the process has correctly identified the laser pixels, a visualization is presented to the user. A visualization sample is shown in Figure 3.3, which shows inlier laser pixels (blue points), outlier laser pixels (yellow points), the fitted laser line (green line), and the detected checkerboard corners. If the user accepts this image, the checkerboard plane Π_b is computed using Perspective-n-point (PnP) method [36], and the incident laser points $\mathbf{X}$ in 3D space are triangulated by first normalizing each inlier laser pixel $\mathbf{x}_i$ and then casting a *Line-of-Sight* (LOS) ray from the camera optical center through the normalized laser points to intersect with Π_b. Therefore, incident laser points can be interpreted as point samples of the laser plane. The software finally solves the laser plane position Π_b by first using RANSAC to remove outliers and then using Singular Value Decomposition (SVD) to solve the plane parameters.

3.4 Sensitivity Analysis

The sensitivity of the sensor is defined as the sensor's response to a 1 mm depth change. The response is the shift in pixels of the observed laser stripe on the image.

Algorithm 1 Laser plane calibration

procedure SOLVE LASER PLANE(images)
  $\mathbf{X} \leftarrow [\cdot]$ ▷ Empty set for incident 3D laser points
  for each $\mathcal{I} \in$ images **do**
    Undistort($\mathcal{I}$) ▷ Using pre-calibrated camera model
    $\mathbf{x}_\mathcal{I} \leftarrow$ DetectLaserPixels($\mathcal{I}$)
    $\mathbf{x}'_\mathcal{I} \leftarrow$ RANSACFindLineInliers($\mathbf{x}$)
    if UserInspectResults($\mathcal{I}$, $\mathbf{x}'_\mathcal{I}$, $\mathbf{x}_\mathcal{I}$) = pass **then**
      $\Pi_b \leftarrow$ SolveCheckerboardPlane($\mathcal{I}$)
      $\mathbf{X}_\mathcal{I} \leftarrow$ TriangulateLaserPixels($\mathbf{x}'_\mathcal{I}, \Pi_b$)
      $\mathbf{X}$.insert($\mathbf{X}_\mathcal{I}$)
    end if
  end for
  $\Pi_l \leftarrow$ RANSACFit3DPlane($\mathbf{X}$)
  return Π_l
end procedure

Given an error bound of the laser detection algorithm in pixels, the sensor's 3D measurement error can be computed as the error bound divided by the sensitivity. As an example, if the laser detection error is 0.5 pixel and the sensitivity is 5 pixel/mm, the sensor's 3D measurement error is 0.1 mm.

The sensitivity is a function of the camera model, the laser leaning angle defined as the angle between the laser light and the camera's optic axis (0 leaning angle indicates the optical axis is parallel to the laser plane), the elevation angle along the fan-shaped laser light, and depth. Although the baseline distance between the laser projector and the camera also changes the sensitivity, it is often designed to be as small as possible for compactness. An illustration of the elevation angle and the depth definition is shown in Figure 3.4.

A custom software is developed to analyze the sensitivity. This software serves three purposes: a) to characterize the sensor's 3D measurement error; b) to study the distribution of sensitivity over depth and elevation angle; c) to find the best laser leaning angle in terms of sensitivity to achieve the optimal sensor design. Some sample results generated by the software is shown in Figure 3.4. From the results, it can be observed that the sensitivity does not change drastically with elevation angle or laser leaning angle. Specifically, in terms of elevation angle, the sensitivity is

slightly higher in the center than on the sides; in terms of laser leaning angle, the sensitivity reaches the global maximum value near 40 degrees. The major factor affecting the sensitivity is the depth: with a 45-degree leaning angle, the sensitivity drops from 6 to 0.3 pixel / mm as depth increases from 1 inch to 4 inches.

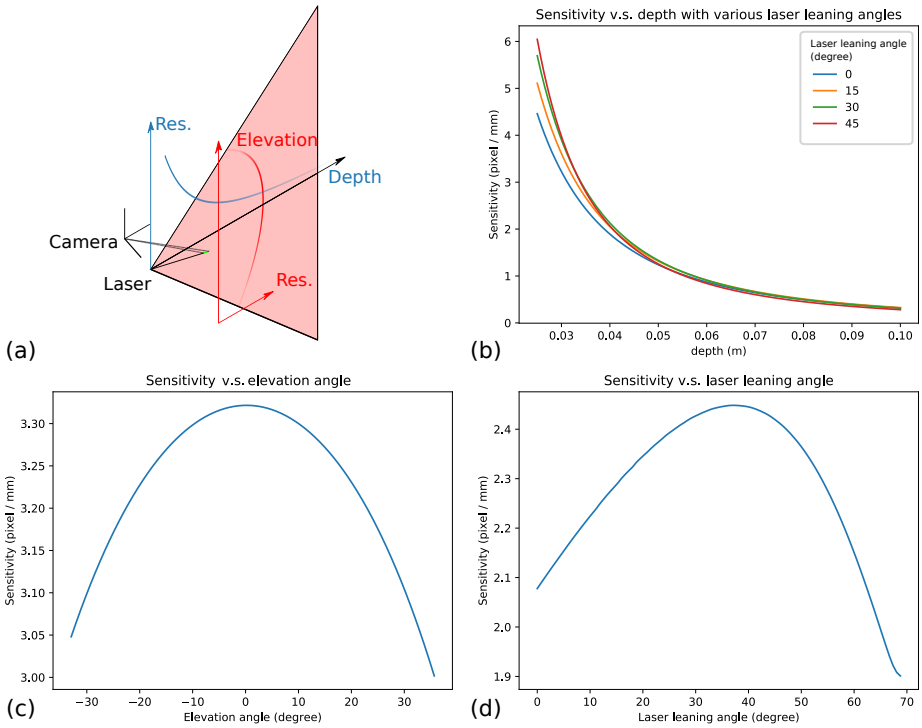

Figure 3.4: Illustration of the sensor's sensitivity analysis and sample result plots generated by the sensitivity analysis. (a) Illustration of the definition of depth and elevation angle; (b) Sensitivity analysis with depth, revealing severe sensitivity decrease as depth increases; (c) sensitivity with elevation angle, showing sensitivity in the center direction is slightly higher that off-center; (d) sensitivity with laser leaning angle, showing sensitivity first increase as the leaning angle increases to 45 degrees but then decreases rapidly.

3.5 Software Framework

The sensor software framework and the data flow are shown in Figure 3.5. This framework contains sensor control, sensor data driver, data pre-processing, SLAM, and dense colored mapping. The data flow starts with the sensor driver, where the

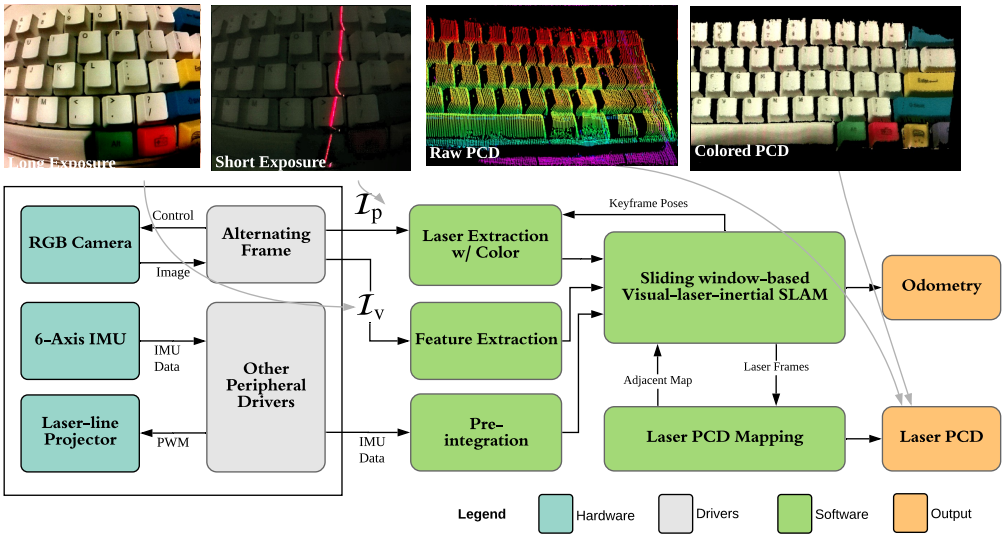

Figure 3.5: Software framework and data flow visualization.

alternating-frame imaging driver controls the camera exposure and laser on/off in synchronization and outputs two separate image streams in an alternating fashion. These two image streams are named visual frames and profiling frames, denoted as $\mathcal{I}_v$ and $\mathcal{I}_p$ respectively. Profiling frames serve the sole purpose of generating 3D laser point cloud via laser stripe detection and triangulation. Visual frames are used for feature-based SLAM, where visual features are detected and tracked. Other peripheral drivers handles other on-board sensors, such as the IMU. IMU data is preintegrated before fed to SLAM to reduce computation cost. The proposed SLAM method takes in laser points, feature tracking data, and preintegrated inertial data and performs a joint optimization in a tightly-coupled fashion. In the SLAM process, the laser point cloud is also colored by associating laser points with visual frames. Finally, the framework provides two outputs to the user: a 3D map represented as colored point cloud and sensor pose estimation in a local reference frame.

3.5.1 Alternating-frame imaging

A highlight of our software is a custom designed sensor driver, which enables measuring two unique types of information using a single camera sensor by alternating the sensor between two configurations. In order to perform SLAM using the monocular profiler,

the single on-board camera is required to fulfill two tasks: one is to track visual content to estimate camera motion, and the other is to capture and triangulate the laser stripe to generate dense 3D geometric data. However, these two tasks require contradictory sensor configurations. Tracking visual content mandates that the image to be neutrally exposes so that features can be stably detected and tracked. The presence of the laser stripe will also interfere with feature-tracking, for it not only result in false-positive features but also interrupts the tracking of any feature that passes the laser stripe on the image. On the other hand, the laser triangulation task requires that the image to be under-exposed such that the laser stripe exhibits high contrast against the background; otherwise, the laser detection is often prone to errors which burdens the SLAM.

The proposed alternating-frame imaging method addresses issue by alternating the camera and the laser between two configurations and generates two interleaving types of frames: visual frames $\mathcal{I}_{\text{le}}$ and profiling frames $\mathcal{I}_{\text{se}}$. Thus, both the images for camera motion estimation and the images for laser depth triangulation can be captured at adjacent sample frames. Figure 3.6 illustrates the interleaving timing sequence. The visual frame configuration, used for tracking visual features, has longer exposure time with the laser turned off, generating neutrally exposed images undisturbed by the laser stripe; The profiling frame for triangulating the laser stripe are under-exposed in order to exhibit a high laser-to-background contrast. In addition, the visual frames are also used to color the laser point cloud, enabling photo-realistic 3D reconstruction. Thus, this approach allows the monocular camera to capture both color and geometric information with minimal time gap, in order to eliminate the need for two separate cameras and thus reduce the sensor's physical size that is critical for confined space requirements.

A camera driver for the alternating-frame method is custom developed. The driver software extensively use the Multimedia Abstraction Layer (MMAL) library to communicate with and control the camera. Due to the sensor's size constraint, an low-cost camera without any triggering or synchronization support is chosen for its small footprint. Shown in Figure 3.7, the driver software has three components: receiving buffer, image encoder, and alternating-frame controller. To overcome the lack of camera status output, the receiving buffer outputs a signal each time it receives raw data from the camera. This timing signal triggers the alternating-frame controller

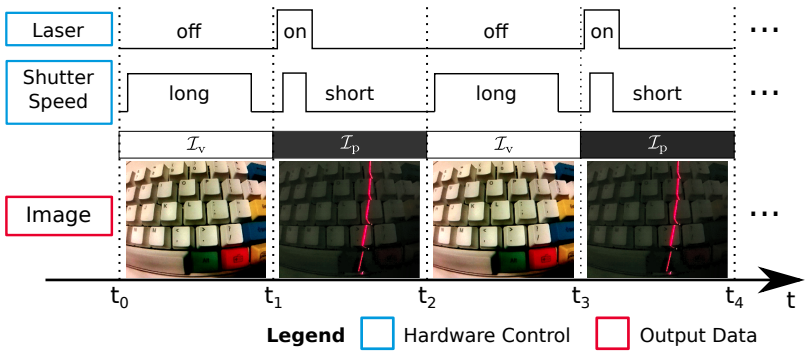

Figure 3.6: Timing sequence of the alternating-frame imaging method. The camera shutter alternates between long and short exposure "on" times, keeping frame duration constant for all frames. The laser's state toggles synchronously with the alternating exposure. This approach enables the monocular sensor to produce two sequence of images in an interleaving fashion that provide RGB and depth information.

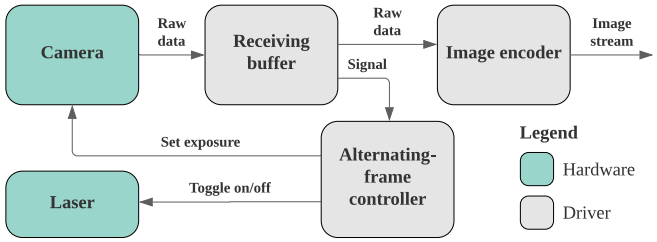

Figure 3.7: Alternating-frame camera driver software diagram. Due to the lack of camera status output, a receiving buffer outputs a signal each time it receives raw data from the camera. This signal triggers the sensor configuration alternation.

to toggle the camera exposure time and laser switch. This toggle has a delay to account for the time lag between the camera frame finish and the receiving buffer signal.

3.5.2 Laser stripe detection

For each $\mathcal{I}_{se}$, we detect the laser stripe pixels using the center-of-mass method [15] and then triangulate these pixels into 3D points as described in Section 3.2. To robustly detect the laser-ring from the images, we propose a comprehensive computer vision method which includes four steps. 1) A preprocessing step mitigates the noise of the raw image using a median filter. 2) The filtered image is converted from RGB space

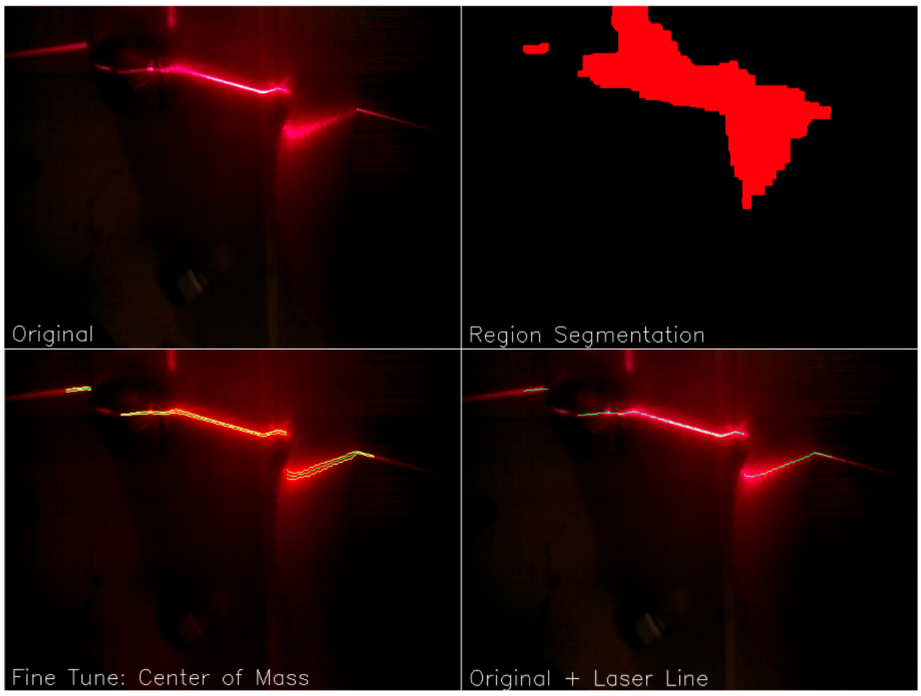

Figure 3.8: Visualization of the laser detection process. The top left image shows the input image captured by the camera. An thresholding operation in the HSV domain segments out the region of interest, shown in the top right, that contains the laser stripe. For each pixel column, the procedure first find a segment of pixels with high intensities and then compute their center-of-mass in terms of intensity. In the bottom left image, the yellow points show the bounds of the per-column pixel segment, and the green points are the center-of-mass.

into HSV (hue-saturation-value) space, and image regions with red hue and relatively large value (lightness) are segmented out. This HSV mask is then morphologically dilated for better robustness. 3) In the resulting masked-out image, the laser points are detected based solely on intensity. Since the sheet of laser light is parallel to the x axis of the camera frame, we make the assumption that there is only one laser pixel in each pixel column. Based on this assumption, for each pixel column we find the pixel segment with the highest intensities and compute the center-of-mass location of this segment. 4) Finally outliers of the laser points are detected and rejected by grouping the points into geometrically consecutive segments and then discarding the segments with short lengths. The steps described above are visualized in Figure 3.8.

Chapter 4

Visual-laser-inertial SLAM

The proposed SLAM method fuses visual feature measurements, depth measurements from laser scan, and inertial measurements to achieve high localization accuracy. Each sensor plays a different role. Visual features serve as the main source of camera motion estimation. The IMU helps handle abrupt motion and estimate orientation thanks to its observability of roll and pitch angles. Finally, the laser points provide the metric scale for the visual odometry and help maintain mapping consistency via point cloud alignment. The proposed SLAM framework can be broken down to the following components:

- A front-end that pre-process raw sensor data into visual features, 3D laser points, and pre-integrated inertial data.
- An initialization process that bootstraps the optimization problem structure.
- An odometry component for camera motion estimation.
- A map appearance generator by associate each map point with a color estimation.
- A mapping module which registers laser points into a point-based map representation.
- A window-to-map tracking component that aligns current measurements to the map to correct odometry drift.

4.1 Front-end

The front-end pre-processes the three raw sensor data (visual images $\mathcal{I}_v$, profiling images $\mathcal{I}_p$, and inertial data) into corresponding intermediate representations needed by the SLAM. Specifically, from visual images we detect and track visual features. From profiling images, the laser points are detected and triangulated into depth information. The inertial data is preintegrated to save computation time in the SLAM.

4.1.1 Visual front-end

The visual front end performs three tasks: feature detection and tracking, identifying *features-on-laser* from all the features, and select keyframes.

Visual features $\mathcal{F}$ are extracted and tracked in each $\mathcal{I}_\mathrm{v}$ image using KLT optical flow [40]: existing features in the previous frame are tracked and new feature points are extracted to maintain a minimum number of features. In order to maintain an even distribution of features on the image, a minimum distance in pixels between any two features is enforced. Each feature $f_i \in \mathcal{F}$ is typically observed over multiple consecutive frames where each observation is a 2D point on the image; the observation of the feature f_i in the jth frame is denoted as $\mathbf{x}_i^j$.

We define *features-on-laser* $\mathcal{F}_l$ as a subset of feature points $\mathcal{F}$ that are close to the laser scan; for these features, the laser point cloud can help accurately estimate feature depths. A feature f is defined to be a feature-on-laser if any of its observations is close to the laser stripe pixels. Since visual frames do not see the laser stripe, we assume that the motion is negligible between frames and we directly use laser stripe pixels in adjacent $\mathcal{I}_\mathrm{p}$'s to check for this criteria. Each feature is associated with a primary observation frame c_f^*. For a *features-on-laser* $f_i \in \mathcal{F}_l$, its primary observation frame is the frame whose observation of the feature is the closest to the laser stripe in the adjacent profiling frame. For a feature $f_i \notin \mathcal{F}_l$, its $c_{f_i}^*$ is defined as its first observation frame.

Keyframes are a subset of $\mathcal{I}_\mathrm{v}$ frames, an $\mathcal{I}_\mathrm{v}$ frame becomes a keyframe if the average feature parallax from the previous keyframe is sufficiently large or the number of tracked features from the previous keyframe is too low. The usage of keyframes

significantly benefits the computation efficiency and has been a popular approach since its early introduction [33].

4.1.2 Profiling front-end

For each $\mathcal{I}_{\mathrm{p}}$, we detect the laser stripe pixels using the center-of-mass method [15] described in Section 3.5.2 and triangulate these pixels to obtain corresponding 3D laser points using the method in Section 3.2.

4.1.3 Inertial front-end

Preintegration is a commonly used technique to handle inertial integration efficiently by avoiding repeated computation. We perform preintegration following the works in [16, 48].

4.2 Initialization

Since the factor graph formulation in Section 4.3 requires an initial estimation of keyframe poses and feature depths, an initialization is required to bootstrap the estimator. The initialization is a two-step structure-from-motion (SfM) process that first attempts to establish a transformation between two keyframes using two-view geometry and then estimate other keyframe poses in the sliding window. The laser information is also incorporated into the initialization to ensure a correct scale of the visual structure.

We initialize the sliding window-based SLAM framework for the initial estimation of keyframe poses and feature depths using the following procedures. 1) First find two keyframes in the sliding window with enough parallax, such that the first frame is the primary observation frame of several features-on-laser. The large parallax benefits the accuracy in transformation estimation. 2) The up-to-scale transformation between the two frames is estimated using the eight-point algorithm [23] with an arbitrary scale s_0. 3) Depth $\hat{d}$ of all the common feature points in the two frames are estimated using triangulation. 4) The correct scale $\hat{s}$ of the visual structure is then estimated using each feature-on-laser's closest laser pixel's depth $\bar{d}$: $\hat{s} = s_0 \cdot (\sum_i^K \bar{d}_i/\hat{d}_i)/K$. The

two keyframes' poses and feature depths are then corrected using $\hat{s}$. This step uses the laser information to estimate the visual structure's scale. It should be pointed out that for a feature $f_i \in \mathcal{F}$, its closest laser pixel's depth is usually not the same value as the feature's true depth; however, they are usually close enough and the estimated scale is often only slightly incorrect. Once initialized, the sliding window optimization is typically able to converge to the correct scale quickly thanks to a more accurate projective association method. 5) Given the initialized structure of the two keyframes, poses of other keyframes in the sliding window are estimated using the perspective-n-point algorithm [36], and depths of the remaining feature points in the sliding window are estimated using triangulation. 6) Finally, a bundle adjustment (BA) optimizes all camera poses and feature depths in the sliding window.

If the above initialization procedure finishes successfully, the global reference frame is set to be the camera reference frame of the first keyframe in the sliding window. Poses of profiling frames $\mathcal{I}_\mathrm{p}$'s are estimated by interpolating between poses of adjacent keyframes using inertial integration. Using these pose estimations, the individual laser scans associated with each $\mathcal{I}_\mathrm{p}$ can be registered into the global reference frame and form a 3D point cloud. If any of the steps fails, such as the BA optimization fails to converge, the initialization process is abandoned and will reattempt after the arrival of a new keyframe.

Given an initialized camera motion trajectory and pre-calibrated extrinsic transformation between camera and IMU, we initialize the inertial-related variables including biases, velocity and gravity using methods described in [48].

4.3 Sliding-Window-based Factor Graph Formulation

We propose a tightly-coupled *visual-laser-inertial SLAM* (VLI-SLAM) formulation in a sliding-window of keyframes. Ceres-Solver [1] is employed to solve the non-linear least squares optimization problem. In a sliding window consisted of n keyframes and m features with more than one observations, the full state vector $\mathcal{X}$ is defined in Equation 4.1.

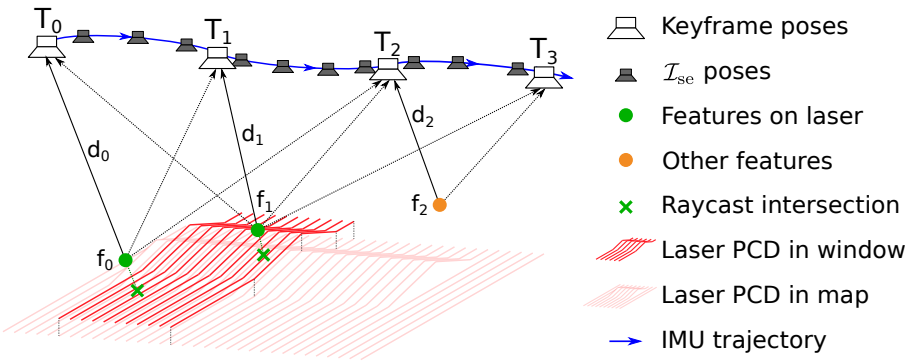

Figure 4.1: Illustration of the sliding window-based visual-laser-inertial SLAM. The sliding window is consisted of several keyframe poses, features observed by the keyframes, laser point cloud observed in the time span of the sliding window, adjacent laser point cloud in previously built map (if revisited), and inertial measurements.

$$\begin{aligned} \mathcal{X} &= [\chi_0, \chi_1, \ldots, \chi_n, \lambda_0, \lambda_1, \ldots, \lambda_m] \\ \chi_k &= \left[\mathbf{T}_{c_k}^w, \mathbf{v}_{c_k}^w, \mathbf{b}_a, \mathbf{b}_g\right], k \in [0, n] \end{aligned} \tag{4.1}$$

In Equation 4.1, χ denotes the states associated with each keyframe and λ_i denotes the inverse feature depth of the ith feature in its primary observation frame $c_{f_i}^*$. Each keyframe state χ contains a keyframe camera pose in the global reference frame $\mathbf{T}_c^w$, the linear velocity of the camera relative to the global frame $\mathbf{v}_c^w$, the accelerometer biases $\mathbf{b}_a \in \mathbb{R}^3$, and the gyroscope biases $\mathbf{b}_g \in \mathbb{R}^3$.

A combination of four types of residuals are minimized in the optimization problem: the visual feature depth residual given laser point cloud, the visual feature reprojection residual, the inertial measurement residual, and the marginalization factor. An additional factor named the window-to-map tracking factor can be added to correct local drift when re-visiting previous scanned areas. This factor is described in Section 4.6. An illustration of the proposed SLAM formulation is shown in Figure 4.1, and Figure 4.2 shows the corresponding factor graph formulation.

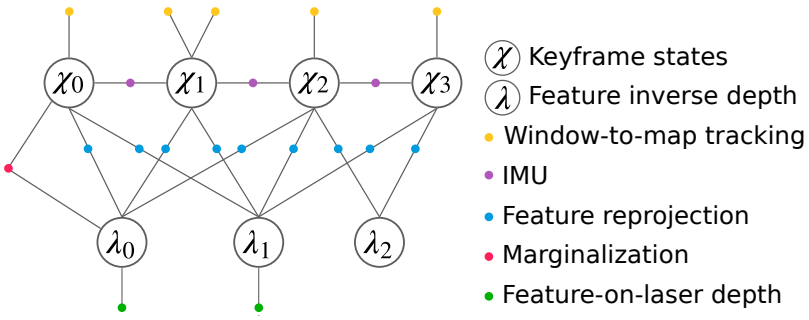

Figure 4.2: The factor graph formulation. The SLAM problem consists of five types of factors: feature reprojection factor, inertial factor, marginalization factor, feature-laser association factor, and window-to-map tracking factor.

4.3.1 Features-on-Laser Depth Residual

Depths of $\mathcal{F}_l$ can be accurately estimated using the depth prior from the registered laser point cloud. The depth prior $\bar{d}_i$ of a feature-on-laser $f_i \in \mathcal{F}_l$ is computed using projective data association involving four steps. 1) A feature LOS ray is cast in the primary observation frame $c^*_{f_i}$, from the camera optic center through the normalized feature point. 2) We find the patch of the 3D laser points that is near the feature LOS ray. 3) A 3D plane is then fitted to these laser points and the intersection between the plane and the feature LOS ray is computed to find $\bar{d}_i$. 4) Finally to ensure the correctness of the projective association, the result is rigorously checked with three criteria: the singular values of the plane fitting problem indicates that the laser point patch is planar, the normal vector of the planar patch is not perpendicular to the feature LOS ray, and the feature LOS ray passes through the center of the patch. Intuitively, a feature that sits on a planar surface that faces the camera will typically result in most accurate and robust projective feature-laser data association. Therefore, these three criteria help rule out error-prone cases such as the vertices of objects. Using these depth priors $\bar{d}$, we introduce a residual for $\mathcal{F}_l$ described in Equation 4.2.

$$r_l(\mathcal{X}) = \sum_{f_i \in \mathcal{F}_l} \left\| \frac{1}{\lambda_i} - \bar{d}_i \right\|^2 \tag{4.2}$$

4.3.2 Feature Reprojection Residual

For each feature $f_i \in \mathcal{F}$, reprojection residuals defined in Equation 4.3 are evaluated between the primary frame $c^*_{f_i}$ and every other observation frame in the sliding window $\mathcal{C}$. In Equation 4.3, $\mathbf{x}_i^j$ denotes the pixel observation of the ith feature in the jth keyframe, and $\mathbf{x}_i^*$ is the observation in $c^*_{f_i}$; $\pi_c(\cdot)$ denotes camera projection function which maps a 3D point in the camera frame onto the image; $\pi_c^{-1}(\cdot)$ denotes back projection function which maps a 2D image point onto the normalized image plane; $\mathbf{T} \in \mathrm{SE}(3)$ denotes a Euclidean transformation matrix.

$$r_c(\mathcal{X}) = \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{C}} \left\| \pi_c \left(\mathbf{T}_w^{c_j} \mathbf{T}_{c^*_{f_i}}^w \frac{1}{\lambda_i} \pi_c^{-1} \left(\mathbf{x}_i^* \right) \right) - \mathbf{x}_i^j \right\|^2 \tag{4.3}$$

4.3.3 Inertial Measurement Residual

We follow the IMU measurement residual definition in [16, 48] to help estimate linear velocity, IMU biases, and camera poses; details are not elaborated for brevity. Since the laser point cloud provides metric scale information, IMU is not necessary for the scanner to function but is still desirable for directly observing roll and pitch angles and being able to handle abrupt motion.

4.3.4 Marginalization

To maintain a fixed problem size, old keyframes exiting the sliding window are marginalized into a prior factor. We use Schur complement [26, 52] to carry out the marginalization. We acknowledge that the marginalization factor uses different linearization points than other factors that involves the same states, and some works [18, 56] use the First Estimate Jacobian technique to fix the linearization point. However, we observe that the drift of linearization point is usually negligible for the states that are connected to the marginalization factor.

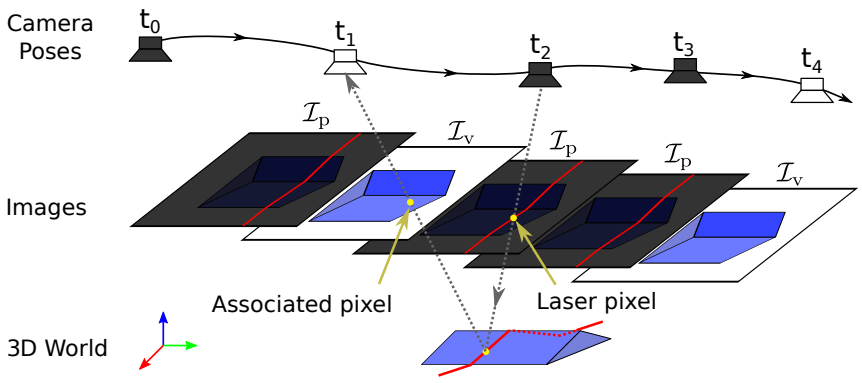

Figure 4.3: Illustration of the color estimation of a laser pixel using projective association.

4.4 Map Appearance Generation

Map appearance generation assigns an RGB color to each 3D laser point. This process enables photo-realistic mapping, which is valuable to many applications. Color information for each laser point is retrieved via projective data association using several temporally adjacent keyframes. The association process is illustrated in Figure 4.3 and described in Equation 4.4. Given a laser pixel $\mathbf{x}_i^{c_k}$ in a profiling image frame c_k, we first transform the corresponding 3D laser point in the camera frame $\mathbf{X}_i^{c_k}$ into the global reference frame $\mathbf{v}_i$. $\mathbf{v}_i$ is then projected onto an adjacent keyframe c_j to find the associated visual pixel $\mathbf{x}_i^{c_j}$, whose RGB value is the estimation of the laser pixel's color. To reduce noise, this process is performed with several adjacent keyframes and the averaged color is used as the final estimation.

$$\mathbf{x}_i^{c_j} = \pi_c \left(\mathbf{T}_w^{c_j} \mathbf{T}_{c_k}^w \mathbf{X}_i^{c_k} \right) \tag{4.4}$$

4.5 Mapping

We adopt a point-based map representation similar to [31, 58], where each map point is generated from a laser point and contains the following attributes: position in the global reference frame $\mathbf{v} \in \mathbb{R}^3$, normal vector $\mathbf{n} \in \mathbb{R}^3$, RGB color $\mathbf{c} \in \mathbb{R}^3$, and weight

$w \in \mathbb{R}$. Laser points from a profiling frame are added to the map after that frame exits the sliding window.

In order to prevent the map size from growing to infinity when the scale of the scanned scene is finite, a merging operation of map points reduces map size. For each laser point to add, if there exist a nearby map point $\mathbf{p}$ with compatible color and normal, then the new point is merged into $\mathbf{p}$; if not, the new point is added to the map and its normal is estimated using nearest neighbors algorithm [32]. The merging operation linearly combines the position and color attributes with a weighted average. The normal vectors are re-computed for all map points near merged points. The weight attribute is the number of times that a map point is merged with a new point.

4.6 Window-to-map Tracking

Back-and-forth scanning motion or similar motion patterns involving repeated scanning is common with laser profilers. Users perform these motion patterns in order to obtain a higher point density or to fill "holes" in the map. However, the accumulation of odometry drift, even when it is millimeter-level, violates the consistency of the map when the profiler revisits a previously scanned region. The drift often manifest in a mapping "ghosting" effect where the map has multiple layers of point cloud.

To account for this issue and to maintain mapping consistency, many RGB-D SLAM methods have adopted a frame-to-map tracking approach [31, 45, 58], where new RGB-D measurement frames are aligned to the previous built map instead of to the previous measurement frame. This approach can be recognized as map-centric instead of localization-centric. However, with a laser profiler, laser points in a single profiling frame are co-planar and geometrically insufficient to account for 6 DoF motion. Therefore, we propose a window-to-map tracking approach, where the registered laser point cloud in the sliding window is aligned to the map. Since odometry drift exists within the sliding window, a nonrigid Iterative Closest Point problem is formulated where the laser point cloud of each $\mathcal{I}_\mathrm{p}$ frame are treated as rigid, but transformation between $\mathcal{I}_\mathrm{p}$'s in the sliding window are treated as nonrigid. This is achieved by incorporating per-point point-to-plane residual defined in Equation 4.5 into the SLAM formulation. In Equation 4.5, $\mathbf{v}_i$ is a laser point from an $\mathcal{I}_\mathrm{p}$ in the sliding window, and c_k and c_{k+1} are the two temporally adjacent keyframes; $f(\cdot)$

denotes a pose interpolation function to estimate the $\mathcal{I}_{\mathrm{p}}$ pose using its timestamp; $\mathbf{v}_i^g$, $\mathbf{n}_i^g$, and w_i are attributes of the closest map point to $\mathbf{v}_i$, which is searched for using KD-Tree.

$$r_{icp} = \sum_i w_i \left\| \left(\mathbf{v}_i^g - f\left(\mathbf{T}_{c_k}^w, \mathbf{T}_{c_{k+1}}^w, t_i \right) \mathbf{v}_i \right) \cdot \mathbf{n}_i^g \right\|^2 \tag{4.5}$$

4.7 Solving Non-linear Least Squares

The non-linear least squares problem is solved using Dogleg, one of the popular trust region optimization methods [41]. To improve the system robustness to outliers, we use Cauchy loss function to reduce the influence of large residual values. To enable the SLAM algorithm to run in real-time, a fixed number of optimization iterations are carried out for each frame. As a result, the system may not be able to converge in the beginning but can typically converge within seconds.

Chapter 5

PipeBlaser: In-pipe Mapping Sensor

The PipeBlaser sensor prototype is designed for 3D dense mapping in pipes as narrow as 12-inch in diameter. The mapping system is consisted of multiple sensors, among which the main sensing component is a laser-ring profiler. The profiler scans the inner pipe surface using a laser-ring projector and uses an on-board camera to perform laser triangulation. In addition to the sensor system, a vehicle platform is also designed to drive the sensor along the pipe, although the sensor payload can be integrated to any robot platforms.

This chapter discusses the hardware design, sensor model, calibration, sensitivity analysis, and software framework of the PipeBlaser prototype. Although the PipeBlaser and the Blaser sensor has extremely different configurations and applications, they share many software components since they are designed using the same mapping profiler framework. This demonstrates the proposed framework's ability to generalize to vastly different configurations for various real-world mapping tasks.

5.1 Hardware Design

The prototype hardware consists of three major components: the sensing component, the on-board computer, and the vehicle platform. The vehicle platform serves the purpose of driving the prototype along pipes and carry out experiments, and the

sensor component can be detached from the vehicle and integrated onto other existing robots for various applications. The PipeBlaser prototype is purposefully designed to be compact to allow it to function in pipes as narrow as 12-inch in diameter. Figure 5.1 shows a rendered image of the PipeBlaser prototype mechanical design. The sensing component measures 6 inches in width, 6 inches in height, and 10 inches in length. The whole prototype including the vehicle platform measures 8 inches in width, 8 inches in height, and 12 inches in length.

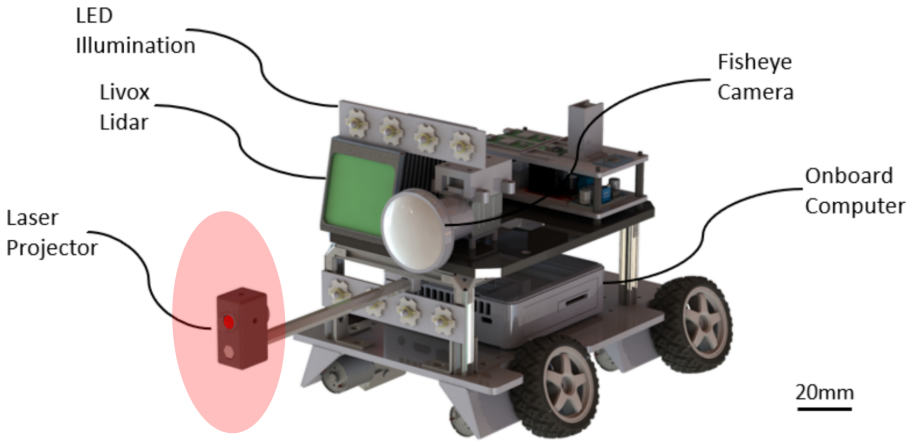

Figure 5.1: A rendered image of the PipeBlaser prototype mechanical design. The prototype consists of multiple sensors, including a camera, an IMU, a laser-ring projector, and a LiDAR. An on-board computer enables online SLAM and data recording for post-analysis. A four-wheel-drive vehicle platform drives the sensor along pipes.

Figure 5.2 shows the hardware component diagram. To achieve a comprehensive sensing capability and to increase robustness, a variety of sensors are incoporated to the PipeBlaser. The main mapping sensor is a laser-ring profiler comprised of a laser-ring projector and a camera. The laser-ring is generated by projecting a laser beam onto a conic mirror. An omnidirectional fisheye lens with 185-degree field-of-view is used to observe the laser-ring as well as the pipe inner surface. We chose Ximea MC050xG-SY camera in the consideration of its low-noise imaging quality, global shutter, high frames-per-second (FPS), and hardware trigger support which is valuable to the alternating-frame imaging and synchronization with other sensors. In the balance of compactness and quality, we selected the Microstrain 3DM-CX5-10 IMU.

An LED array is also integrated to actively illuminate the pipe for the camera. An rotary encoder is attached to a passive caster to measure 1-dimensional displacement and further decrease the localization drift. For computation, we installed an Intel NUC computer to perform SLAM online and to record sensor data for post-processing and analysis.

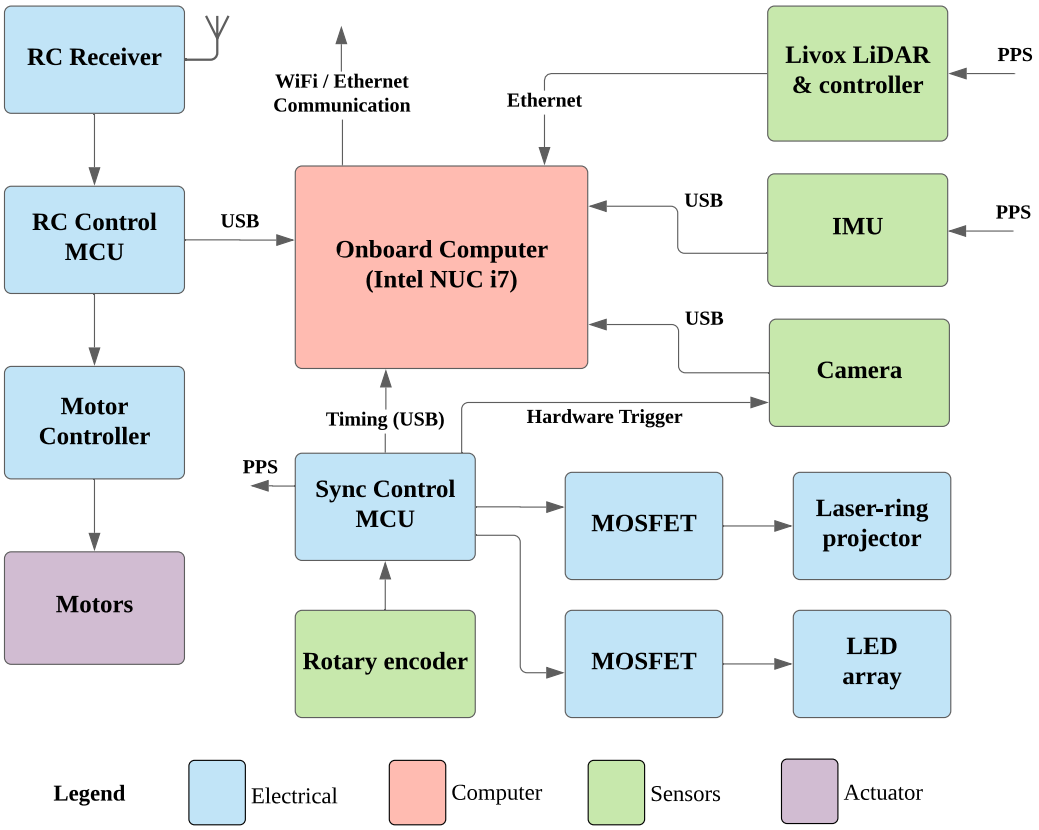

Figure 5.2: Hardware component diagram of the PipeBlaser prototype.

Although the above sensors are sufficient to carry out SLAM, a LiDAR is added to the prototype. The LiDAR chosen is a Livox MID-70. Compared to conventional rotary LiDARs like the Velodyne PUCK, the Livox has a scan pattern more suitable for in-pipe environments and a short minimum measurement range of 5 cm which is desirable for narrow pipes. However, the LiDAR exhibited large measurement error in our characterization, especially at short ranges, and is therefore not used for SLAM. It is added to the sensor suite because unlike the laser profiler which can only scan the side of the sensor, the LiDAR have 3D measurement capability in front of the

sensor which will be useful for path planning and navigation in the future.

A custom vehicle was designed to carry the sensor payload along pipes. This vehicle has four powered wheels that are angled so that they make contact with a 12-inch diameter pipe at right angles. To control the vehicle in a long segment of pipe, we use radio communication capable of transmitting commands over a range of 800 meters. An on-board Micro Control Unit (MCU) decodes the radio signal, controls the motors accordingly, and send control status information to the on-board computer.

5.2 Sensor Model

3D points on the laser ring are recovered from 2D images using triangulation. The sensor model of the laser-ring triangulator, shown in Figure 5.3, is very similar to that of the laser-stripe profiler. Assuming the projected disk-shaped sheet of laser light is perfectly co-planar, we model the laser light as a plane $\Pi_l : \mathbf{n} \cdot \mathbf{X} + d = 0$ in 3D space, which intersects with the inner pipe surface and forms a visible laser ring. The triangulation process is then the same as with the Blaser prototype which is elaborated in Section 3.2.

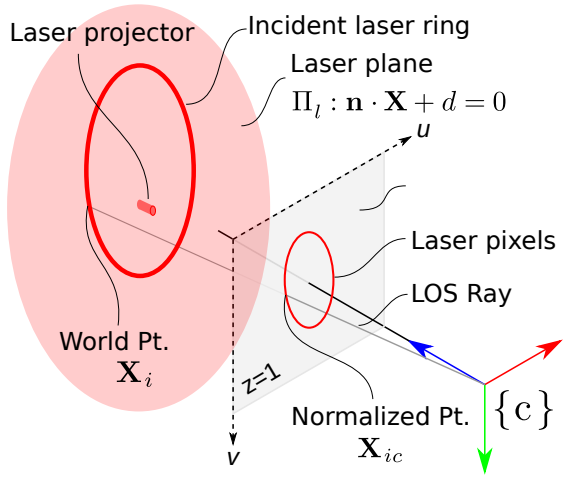

Figure 5.3: Theory of operation: for each laser pixel, a 3D laser point is triangulated by projecting a line-of-sight ray of the pixel and finding its intersection with the laser plane.

5.3 Calibration

The calibration of the PipeBlaser sensor involves four groups of parameters: camera model intrinsic parameters, camera-IMU extrinsic transformation, camera-LiDAR extrinsic transformation, and the 3D position of the laser plane relative to the camera. The camera model and the camera-IMU extrinsics are calibrated using the same methods for Blaser, described in Section 3.3.

The camera-laser extrinsic parameters are calibrated using a custom-developed calibration software. Most ring-laser sensors in the literature rely on mechanical alignment to align the laser with the camera, which would require accurately fabricated tools and is time-consuming. In contrary, the proposed sensor relies on a easier and faster data-driven calibration process. This calibration software takes in images of a checkerboard that intersects with the laser plane captured with the camera. It then extracts the laser points from all these images and computes their 3D positions. Finally, the 3D laser plane is fitted onto all these points. Figure 5.4 illustrates the above process. The software process is the same as the Blaser calibration software, whose algorithm and a detailed process description can be found in Section 3.3.

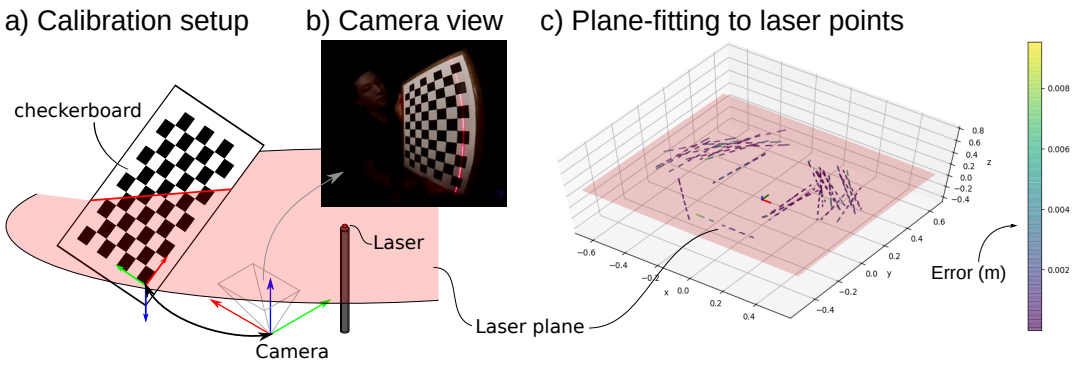

Figure 5.4: An illustration of the camera-laser extrinsics calibration process and the visualized result of fitting a 3D laser plane to the collected sample laser points. (a) The calibration uses images of a checkerboard which intersects with the laser disk, forming a straight laser stripe on the checkerboard. (b) A sample raw image captured by the fisheye camera. (c) The laser plane was fitted to more than six thousand 3D sample points, resulting in 0.9 mm average point-to-plane error.

In the calibration of the PipeBlaser, more than six thousand sample points were used to estimate the laser plane. The points were distributed over a disk shape around

the laser projector and spanned over 800 millimeters in diameter. The plane fitting returned accurate result: the average point-to-plane distance error reached 0.9 mm.

Although the LiDAR is not used for SLAM, we calibrate its transformation relative to the camera to register its data with the rest of the sensors. The official Livox-SDK is utilized for this task. The software takes in camera images and LiDAR point clouds both observing a rectangular board. The camera-LiDAR data association is performed by manually labelling the board corners on the image and in the point cloud.

5.4 Sensitivity Analysis

In order to theoretically analyze the sensor's sensitivity and facilitate the sensor prototype design, we developed a realistic sensitivity analysis software. The sensitivity of the sensor is defined as the image response (in pixels) of a 1 mm change in pipe diameter, which is the higher the better. This software takes in camera model parameters to perform analysis on the chosen camera. The core of this software analyzes the sensitivity given a pipe diameter and a baseline length, which is the distance between the camera optical center and the laser plane. From a user perspective, this software serves two purposes: one is to determine the best baseline length given a pipe diameter to facilitate sensor hardware design for a given application, and the other is to examine a sensor's sensitivity in various pipe diameters given a baseline in order to analyze a sensor prototype's limitations.

Figure 5.5 shows sensor sensitivity with different pipe diameters and baseline lengths. From the plots, the sensitivity decreases monotonically as the pipe diameter increases. We can also conclude that the optimal baseline exists given a certain pipe diameter. For 12-inch diameter pipe, the optimal baseline length is 14.9 cm, although the high-sensitivity band is rather wide compared to that with smaller diameter value. Furthermore, small baselines tend to perform well only under small diameters while longer baselines offer more balanced sensitivities in a wide range of pipe diameters. For the PipeBlaser prototype, we chose a baseline of 10 cm as a trade off between sensitivity and compactness.

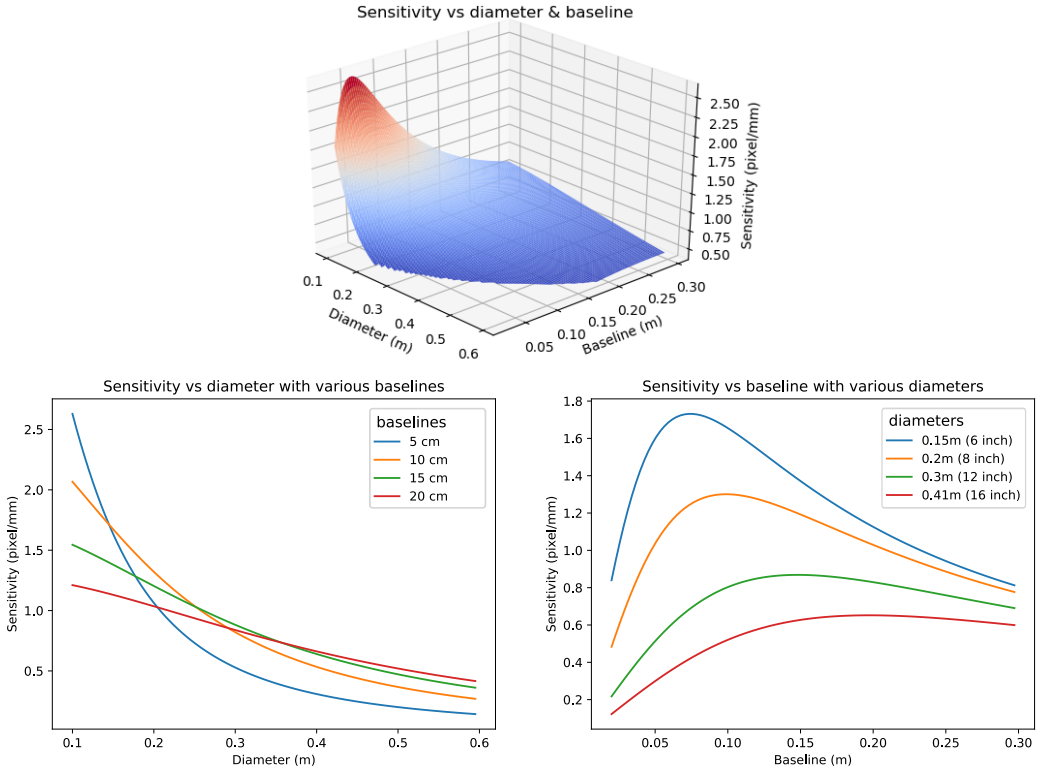

Figure 5.5: Theoretical sensitivity analysis of the laser-ring profiler. The top figure shows a 3D plot of sensitivity with pipe diameter and baseline length. The result on the bottom left shows the sensitivity with pipe diameter given various baseline lengths. This can be used to study a sensor's performance limitations when using it in different pipe sizes. The result on the bottom right shows the sensitivity with baseline length given different pipe diameters. This function can help determine the optimal sensor design for a certain application.

5.5 Software Framework

The software framework of PipeBlaser is shown in Figure 5.6. This framework comprises three components: sensor driver and controller, data pre-processing, and SLAM. The sensor driver takes in measurement from three sensors: a camera, an IMU, and a rotary encoder attached to a caster wheel. It also controls a laser projector and a LED illumination array to enable the camera to capture both visual information for camera motion estimation and the laser ring to generate geometric pipe cross-section profiles. These two image types are captured in an interleaving fashion using the alternating-frame imaging method, which is similar to the Blaser sensor. The raw sensor data pass through the pre-processing pipeline to generate the intermediate data representations required by the SLAM method. The pre-processing methods for each sensor are described in Section 4.1 and 5.6.

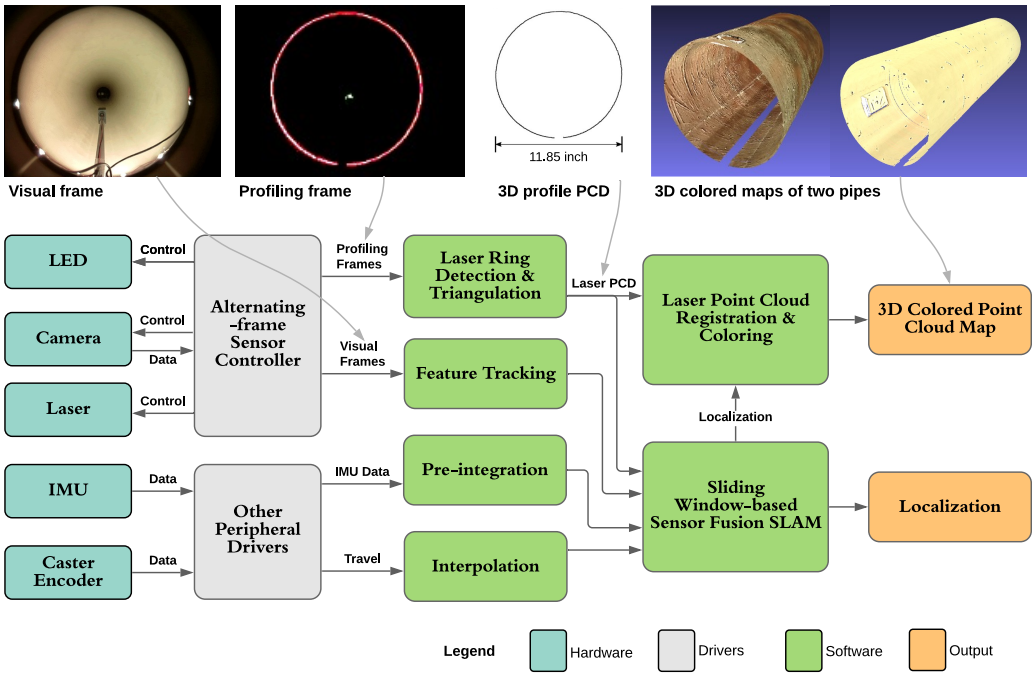

Figure 5.6: PipeBlaser software framework for localization and photo-realistic mapping.

The alternating-frame imaging is similar to the Blaser prototype with differences on the use of LED light and the camera control software implementation. Considering

the darkness of in-pipe environments, an array of eight LED lights are installed around the sensor. These lights are switched on for the visual frames and off for the profiling frames. Since the compactness requirement is not as stringent on the PipeBlaser as with the Blaser, we are able to opt for a larger but high-quality camera which supports hardware triggering. We made use of the "exposure-by-pulse-width" function to trigger camera, which uses a voltage signal to make the camera exposure active. This method enables accurate timing control, reducing the need for timing headroom between frames and thus the camera FPS can be significantly increased.

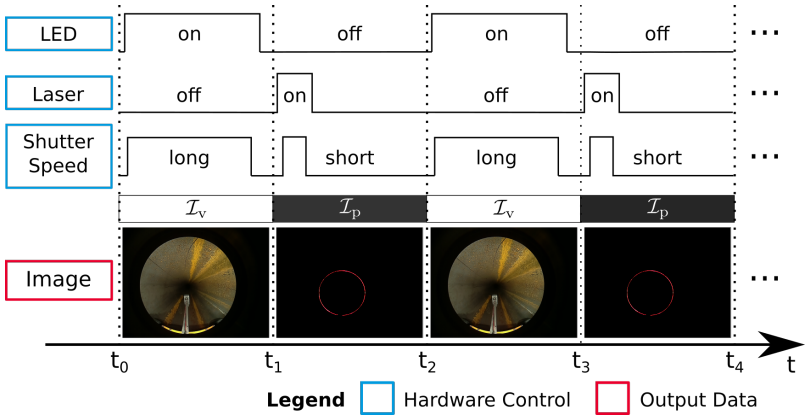

Figure 5.7

Due to the presence of the laser projector mounting pole and the distortion from the fisheye lens, some regions of the camera image do not see the pipe. An image region-of-interest (ROI) mask generation software is developed to reject two types of invalid regions: the area of the laser mounting pole, and the image border area. Since fisheye lenses tend to have inferior imaging quality on the border, the software rejects all the image pixels exceeding a user-defined field-of-view angle using image inverse projection function. Figure 5.8 shows the two-step process of generating the ROI mask.

The laser detection process is visualized in Figure 5.9 which is based on the center-of-mass method similar to the laser stripe detection in Section 3.5.2. The process is consisted of four steps. 1) Apply median filter to reduce image noise. 2) We perform a binary thresholding mask operation on the image in HSV (hue-saturation-value) space, keeping image regions with red hue and relatively large value (lightness). This HSV mask is then dilated to for better robustness. An additional ROI mask excludes

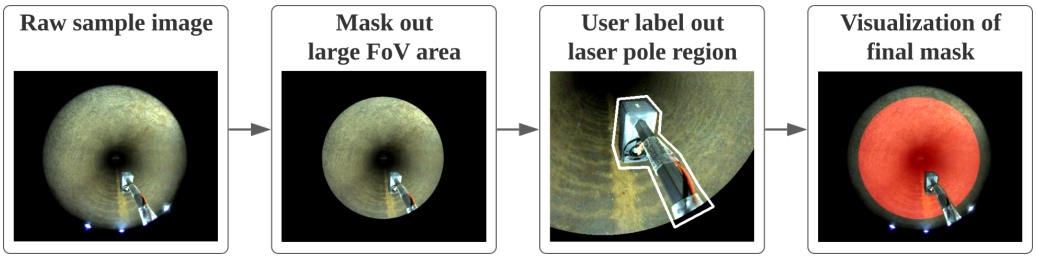

Figure 5.8: Process of generating the region-of-interest image mask. The software first rejects the image border area exceeding a certain field-of-view angle and then asks the user to manually label the laser mounting pole.

the image regions where the pipe is not visible. 3) The laser points are detected based only on intensity. For each ray that emits from the image optic center to an outward direction, we find the pixel segment with the highest intensities and compute the center-of-mass location of this segment. 4) Outliers are rejected using the method in Section 3.5.2.

5.6 SLAM

The VLI-SLAM proposed in Chapter 4 can be directly used for the in-pipe SLAM. Two modifications are made to further tailor the SLAM method to the in-pipe mapping problem. Firstly, the window-to-map tracking component is disabled since in-pipe mapping tends to always travel forward and rarely scans back-and-forth. Disabling this component reduces computation cost by relaxing the computation of the normal of each map point and the merging of map points. Secondly, the encoder information is added to the SLAM factor graph. The encoder is able to measure displacement in the axial direction of the pipe more accurately than visual odometry, and experiments show that it can reduce localization drift. We make the assumption that the robot travels in a straight segment of pipe so that the encoder measures motion in the axial direction. If other sensors indicate this assumption is violated, the encoder information is not used. The factor graph formulation for the in-pipe SLAM is shown in Figure 5.10

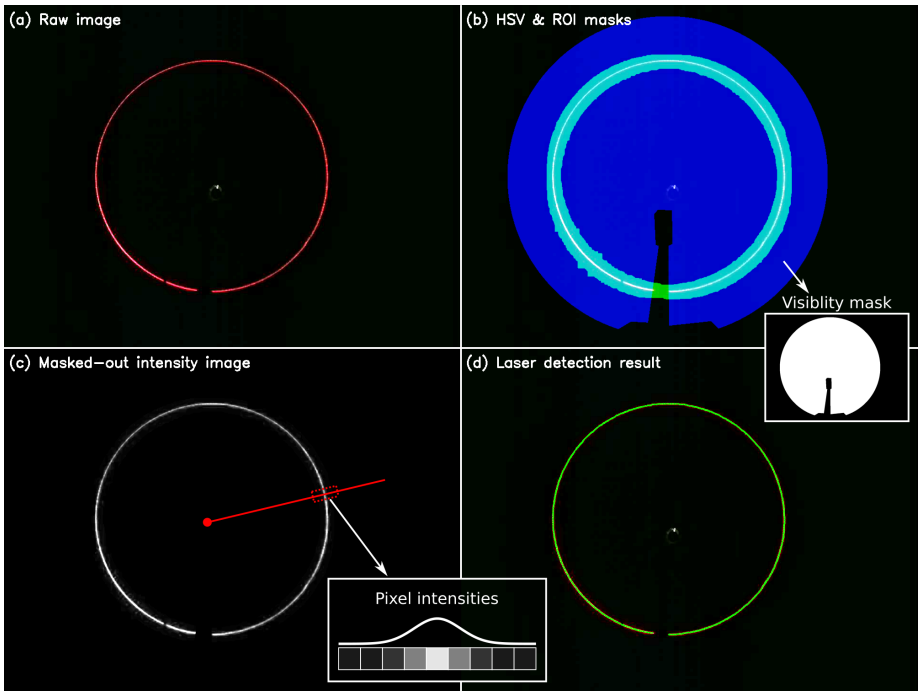

Figure 5.9: The laser ring detection process. (a) shows the original image captured by the camera. In (b), the cyan region is the HSV mask, and the blue region is the region-of-interest (ROI) mask. (c) visualizes the radial laser point detection. In (d), the green pixels are the extracted laser points.

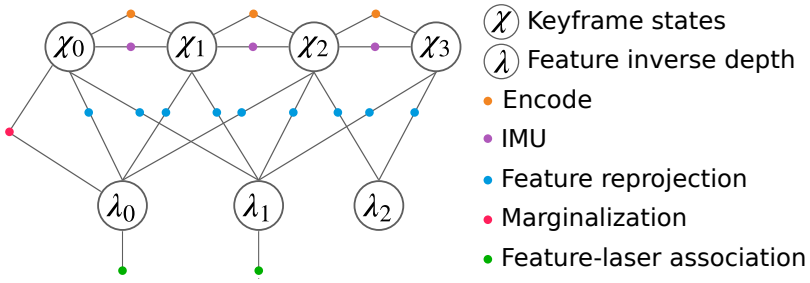

Figure 5.10: Factor graph formulation of in-pipe SLAM using PipeBlaser.

Chapter 6

Experimental Results

6.1 Blaser

The Blaser sensor's performance in hand-held 3D scanning is evaluated with real-world scanning experiments. Targeting localization and mapping benchmarking as main objectives, we first evaluated the localization accuracy of the proposed VLI-SLAM in Chapter 4 against VINS-Mono, a state-of-the-art visual-inertial SLAM method, using the same sensor hardware [48], followed by a comparison of colored point cloud reconstruction against a popular COTS RGB-D camera, Intel RealSense D435. We also showcase the scanning of several industrial and household objects in Figure 6.4. For applications were a positioning aid is present, it can replace the SLAM module and the Blaser sensor can achieve drift-free 3D mapping. This capability is demonstrated with a robot manipulator.

The experiments were conducted by hand-holding the sensor to scan a keyboard. To mimic 3D scanning in confined spaces, the sensor was held at approximately 3 cm above the keyboard facing downward to achieve high triangulation accuracy according to the sensitivity analysis in Section 3.4. The camera motion was kept slow for map density and to test the SLAM's ability to cope with low-excitation IMU data. Because the laser stripe only covered three rows of keys at a time, we scanned the keyboard using a back-and-forth zigzag motion pattern consisting of six passes to incrementally cover the scene, visualized in Figure 6.1. The total trajectory length was 185.4 cm and the average speed was 1.40 cm/s. Figure 1.2c shows the experiment

setup, where the sensor was mounted on a 3D-printed handle attached with motion capture markers for localization ground truth.

The sensor outputs $\mathcal{I}_{\mathrm{p}}$ and $\mathcal{I}_{\mathrm{v}}$ images of VGA resolution at 60 frames per second (FPS) combined and inertial measurements (linear acceleration and angular velocity) at 200 FPS. To achieve real-time SLAM, we used a sliding window size of 8 keyframes and extracted 100 visual features from each $\mathcal{I}_{\mathrm{v}}$. On the testing PC with AMD Ryzen 3700x CPU, the average computation time was 29.8 milliseconds per frame.

6.1.1 Odometry Accuracy Evaluation

We evaluated the proposed VLI-SLAM method against VINS-Mono. A visual-inertial SLAM method is chosen as benchmark because it the method that uses the most sensing capability on the Blaser sensor suite. To evaluate the window-to-map tracking component described in Section 4.6, we experimented with two versions of the proposed SLAM method: one denoted as VLI-Odom with the window-to-map tracking component turned off, and the other as VLI-SLAM with it turned on. The ground truth trajectory was obtained using the Vicon motion capture system (Vicon Industries, Hauppauge, NY, USA).

Figure 6.1 shows the estimated trajectories in the top-down view of VINS-Mono, VLI-Odom, VLI-SLAM against ground truth, with absolute translational and rotational errors analysis. In the error plots, the background colors divide the time period into six segments corresponding to the six passes in the zigzag trajectory, starting from the bottom-left. The performance statistics comparison are listed in Table 6.1, where drift is defined as maximum error over trajectory length.

Based on this experiment, VINS-Mono showed significantly larger translational drift compared to both VLI-Odom and VLI-SLAM, mainly due to inaccurate scale estimation, which was caused by high measurement noise of the low-cost MEMS IMU and low signal-to-noise ratio from the slow sensor motion. VLI-SLAM demonstrated slightly better translational accuracy than VLI-Odom thanks to the window-to-map tracking component, which reduced drift by registering current measurements with historic information. Since this drift-correction is map-centric rather than localization-centric, the mapping benefited more as described in Section 6.1.2. All three methods showed similar rotation estimation performance.

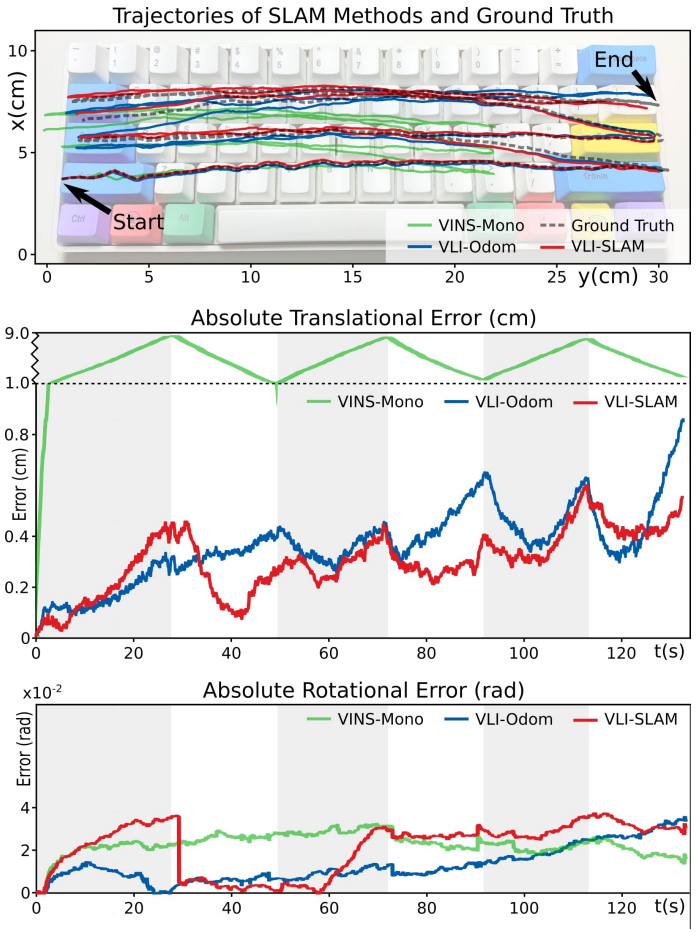

Figure 6.1: Trajectories in top-down view of the proposed SLAM methods, VINS-Mono and ground truth and the associated translational and rotational errors. The background color of error plots indicates different passes in the zigzag trajectory. The top portion in the translational error plot is rescaled to accomodate the large error of VINS-Mono.

Table 6.1: Absolute localization errors and Drift rates

Error metric	VINS-Mono	VLI-Odom	VLI-SLAM
t RMSE (cm)	5.1	0.39	**0.32**
t Max (cm)	8.6	0.86	**0.60**
t Drift (%)	4.6	0.46	**0.32**
r RMSE (rad)	0.022	**0.014**	0.023
r Max (rad)	**0.030**	0.033	0.035
r Drift (10^{-4} rad/cm)	**1.6**	1.8	1.9
t Drift$^{\text{Abs}}$ (%)	23.7	**0.65**	N/A
r Drift$^{\text{Abs}}$ (10^{-4} rad/cm)	4.3	**3.1**	N/A

One key performance metric for real-world 3D scanning is the absolute drift. However, in this experiment, the drift growth often alternated between positive and negative as the sensor motion changed direction, thus the average drift is smaller than the absolute drift. Therefore, we segmented the trajectory into six passes. Within each pass, drift is zeroed at the beginning to evaluate the absolute drift, which is then averaged across the six passes. These translational and rotational drifts are denoted as Drift$^{\text{Abs}}$ in Table 6.1. Since the window-to-map tracking would register the later passes to the first one, VLI-SLAM is not evaluated for absolute drift.

6.1.2 3D Reconstruction Evaluation

The proposed sensor was compared against Intel RealSense D435 since it is one of the smallest low-cost and infrastructure-free 3D scanner although still significantly larger than the proposed sensor. RTAB-Map [35] was employed for SLAM using D435.

Table 6.2: Mapping RMSE statistics

Error metric	VLI-Odom	VLI-SLAM	RealSense
Point-to-point (mm)	1.2	**0.97**	2.3
Point-to-plane (mm)	0.93	**0.76**	2.0

The reconstructed point clouds were geometrically evaluated using a ground truth point cloud, which we obtained using a UR5e robot manipulator (Universal Robots, Odense, Danmark) to scan the keyboard with the proposed scanner. The point-to-

point and point-to-plane RMSEs are shown in Table 6.2, where the proposed sensor with VLI-SLAM showed the smallest error. Figure 6.2 shows the mapping result after each pass. The VLI-SLAM method was able to gradually complete the scan of the entire keyboard and maintain mapping consistency. The repeated scanning also filled "holes" which emerged when a new area was scanned for the first time.

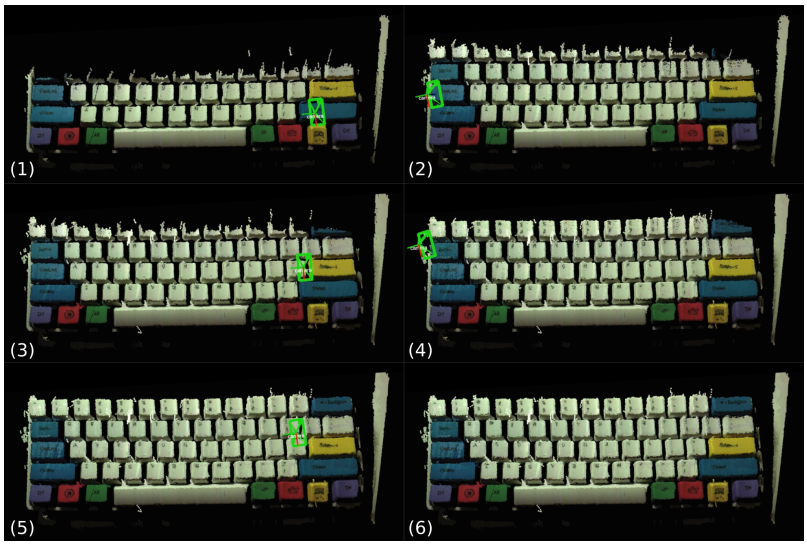

Figure 6.2: Intermediate mapping result after each scan pass. There were in total six passes to gradually complete the scan. The proposed VLI-SLAM was able to maintain mapping consistency under the repeated scanning.

To qualitatively compare both the reconstructed color texture and geometrical shape, we present the photo-realistic colored point clouds as well as spatially color-coded point clouds in Figure 6.3.

Based on the results, the proposed sensor system was able to achieve 3D reconstruction results with finer texture details as well as sharper geometries: comparing Figure 6.3 (d) and (e), our sensor delivered superior reconstruction details on letters and patterns of the keycaps; geometric structures were also sharper in (h) compared to (g) which is more evidently shown in the sectional views (i) and (j). This confirmed the claim that laser-stripe profilers are often able to achieve higher reconstruction accuracy than structured-light based RGB-D cameras. The window-to-map tracking component in VLI-SLAM significantly improved mapping consistency under back-and-forth scanning motions. Figure 6.3 (c) and (f) show the partial point cloud of

Figure 6.3: Comparison of point cloud reconstruction. (a) is a photograph of the scanned scene. (d) and (g) are the photo-realistically and spatially colored reconstruction results by RealSense. Reconstructed using the proposed sensor, (b) and (c) show results using VLI-Odom, and (e), (f) and (h) are with VLI-SLAM. (i) and (j) are the sectional views of (g) and (h) respectively with the red dashed lines as the cutting planes.

(b) and (e) respectively in spatial color-coding. In (c) the point clouds from different passes were clearly separated from each other due to SLAM drift, and in (f) the point clouds were tightly aligned. We observe that although the window-to-map tracking only slightly reduces localization error in Section 6.1.1, the mapping quality was drastically improved. This demonstrated the proposed VLI-SLAM's ability to maintain mapping consistency under back-and-forth coverage scanning, which is a common motion pattern for both laser-stripe profilers and other 3D scanners.

To comprehensively demonstrate the system's scanning capability, we also include the hand-held reconstructions of several other objects in Figure 6.4.

6.1.3 Mapping with External Positioning Aid

For applications that can provide external positioning aid and do not require localization capability, the Blaser can use external pose data to replace the SLAM component and achieve drift-free mapping. Laser scans are registered to the global reference frame to form a 3D map, and photo-realistic mapping is achieved using the

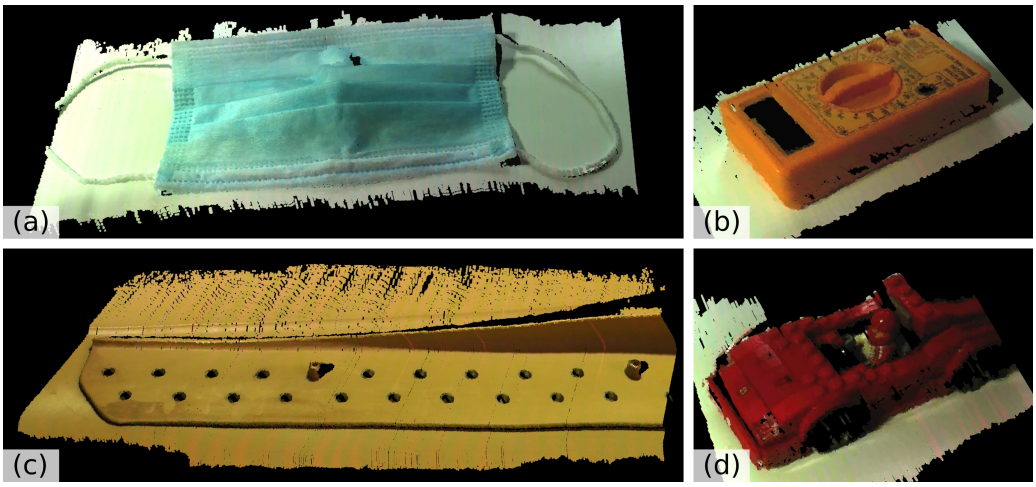

Figure 6.4: Reconstruction using the proposed system of (a) a face mask, (b) a multimeter, (c) a industrial aerospace part, and (d) a toy car.

same projective data association described in 4.4. To demonstrate this capability, we mounted the Blaser onto the end-effector of a UR5e robot manipulator (Universal Robots, Odense, Denmark) and scanned a medical prostate model. The experiment setup and the 3D reconstruction result are shown in Figure 6.5.

6.2 PipeBlaser

We first evaluated the measurement performance of an existing sensor, which is the Livox Mid-70 LiDAR. The PipeBlaser prototype's performance is then evaluated in two aspects: profiling accuracy and mapping accuracy. The profiling accuracy is tested by using the sensor to generate cross-section profiles of a 12-inch diameter pipe. The mapping experiment is performed using the custom vehicle platform to drive the sensor inside a 12-inch pipe and a 16-inch pipe. In both environments the robot travelled at 1.3 cm/s. The camera outputs $\mathcal{I}_{\text{p}}$ and $\mathcal{I}_{\text{v}}$ images of 1232×1028 resolution at 90 FPS combined and inertial measurements at 200 FPS. The rotary encoder generates 1000 pulses per revolution. Paired with a caster wheel of 300 mm in circumference, the encoder has a displacemnt resolution of 0.3 mm.

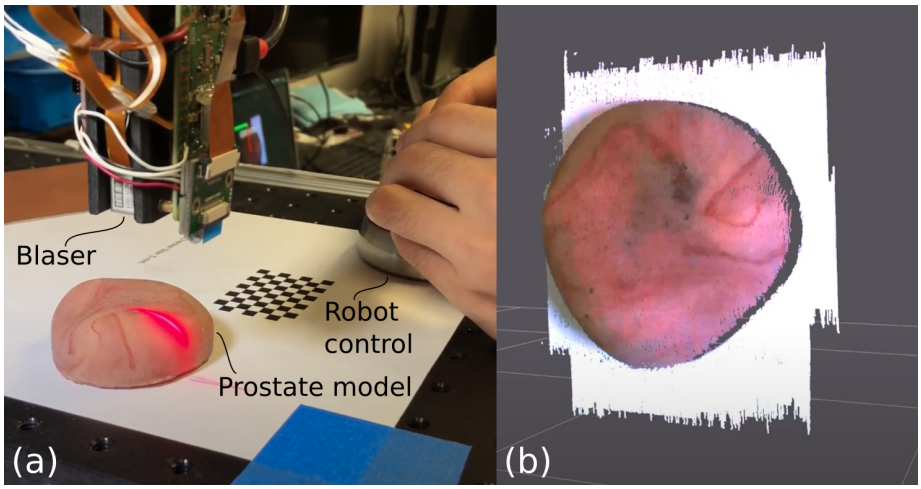

Figure 6.5: Using Blaser with a robot manipulator to scan a prostate model. (a) Experiment setup. The Blaser was mounted onto the end-effector of the manipulator, controlled by a 3D mouse. (b) The photo-realistic 3D reconstruction. When the Blaser uses external pose estimation to replace SLAM, it can achieve drift-free 3D reconstruction.

6.2.1 LiDAR 3D Measurement Characterization

Before the evaluation of the proposed sensor prototype, we first characterized the performance of an existing LiDAR technology. Among multiple LiDAR candidates including Livox Mid-70, Intel RealSense L515, and Velodyne Ultra Puck, we selected the forward-facing Livox MID-70 Lidar as the best fit for this task. Its wide FoV and the circular, nonrepetitive scanning pattern make it particularly suitable for in-pipe depth sensing. Compared to the Velodyne Ultra Puck, the Livox is able to obtain a full scan coverage of the pipe even when stationary. Figure 6.6 shows the accumulated Livox point cloud over 0.1 to 0.5 seconds which contains 1 to 5 data frames respectively. During the acquisition of this data, the Lidar was placed stationary inside a 12-inch diameter pipe. It can be observed that the Livox is able to obtain a full and dense coverage of the pipe.

Our preliminary experiments have shown that the Lidar measurement is subject to non-negligible error, including non-zero bias and large noise. This error is presented in Figure 6.7, where a cross-section segmentation of the point cloud measurement is compared to the ground truth parameter. This comparison reveals diameter mea-

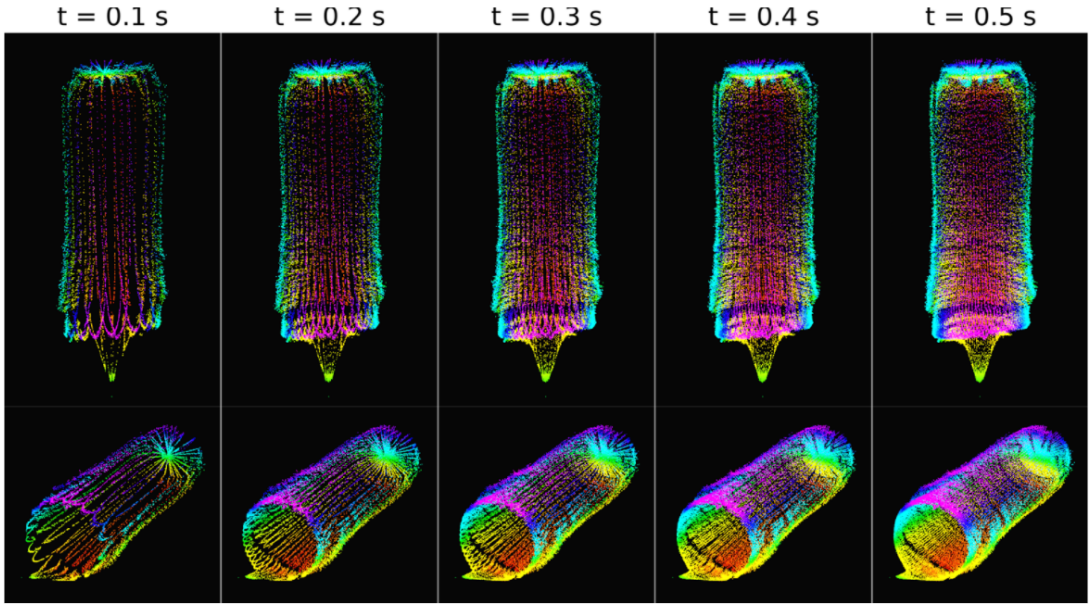

Figure 6.6: Accumulated Livox point cloud in a 12-inch diameter pipe while the Lidar is stationary.

surement bias as large as approximately 4 inches (33% relative error). In comparison, the proposed PipeBlaser exhibited significantly superior profiling accuracy in Section 6.2.2.

6.2.2 Profiling Evaluation

One major aspect of pipe inspection is the pipe diameter. Since the measured diameter can indicate the thickness of the pipe wall, it is one of the most important aspects in pipe geometric integrity. Using the PipeBlaser sensor, the diameter can be measured from the cross-section profiles generated with each profiling image frame.

In a 12-inch diameter pipe, the ground truth diameter, measured using a digital caliper, was 300.4 mm. The diameter measured by the PipeBlaser was 300.9 mm, resulting in an absolute error of 0.5 mm and a relative error of 0.17%. The cross-section profile and a sample profiling image in the 12-inch pipe are shown in Figure 6.8. This result is consistent with the sensitivity analysis in Section 5.4. The sensitivity at 12-inch diameter is approximately 1.0 pixel/mm. Given a laser stripe detection error bound of 0.5 pixel, the theoretical diameter measurement error is 0.5 mm.

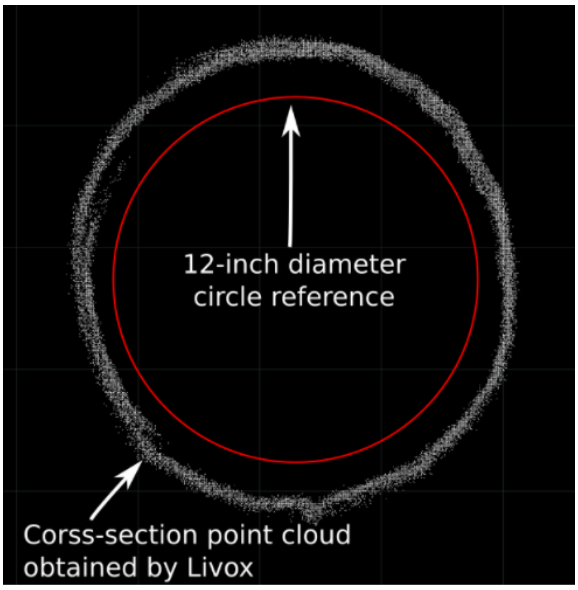

Figure 6.7: Segmented pipe cross-section of the raw point cloud obtained from Livox LiDAR with a ground truth diameter reference.

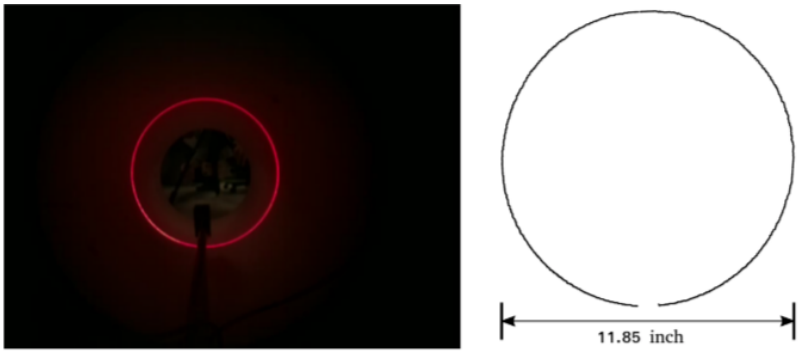

Figure 6.8: A sample profiling image captured in a 12-inch diameter PVC pipe (ground truth diameter is 11.83 inches) and a cross-section profile generated using the image.

6.2.3 SLAM Evaluation

The mapping performance was evaluated in two pipe environments: a 12-inch diameter white PVC pipe with taints and scratches on the inside and a 16-inch steel pipe which is painted to resemble rust texture.In both pipes the experiments were performed in the same way. We evaluated the localization drift using 3D-printed marker blocks by attaching these unique markers to each end of the pipe. The localization drift is calculated by comparing the distance between the two markers in the reconstructed 3D map and the manually measured distance.

In the 12-inch PVC pipe, the ground truth distance between the markers was 83 cm, and the 3D map distance was 82.5 cm, resulting in an absolute error of 0.5 cm and a relative error of 0.6%. Figure 6.9 shows the reconstructed 3D map with marker blocks. In this test the caster encoder data was not incorporated.

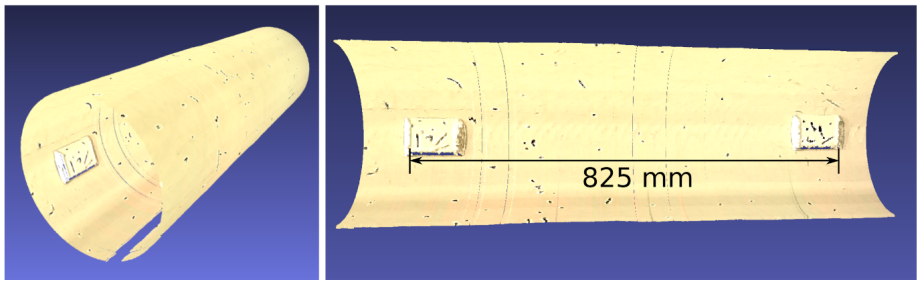

Figure 6.9: The reconstructed 3D map of a 12-inch diameter PVC pipe.

In the 16-inch PVC pipe, we evaluated the localization drift in the same way. However, the caster encoder was incorporated, and we were able to evaluate the localization performance with and without the caster encoder. In this test the ground truth distance between the markers was 60.9 cm. Figure 6.10 shows the 3D mapping result, and Table 6.3 shows the quantitative result. From the results, the localization drift in the 16-inch pipe is higher than in the 12-inch pipe. This may be caused by the inferior sensitivity in larger pipes as analyzed in Section 5.4. However, incorporating the encoder data drastically improves the localization accuracy.

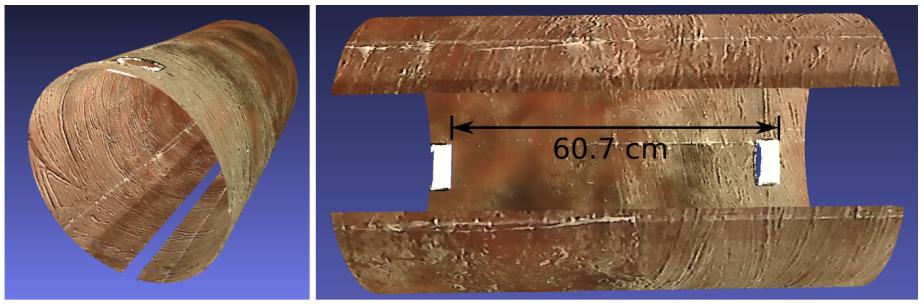

Figure 6.10: The reconstructed 3D map of a 16-inch diameter mock-up steel pipe.

Table 6.3: In-pipe localization error evaluation

Sensors	Meas. distance	Abs. error (cm)	Rel. error (%)
Camera, laser, IMU	59.9	1.0	1.7
Encoder, camera, laser, IMU	60.5	**0.40**	**0.65**

Chapter 7

Conclusions

In this thesis, a sensor framework for confined space mapping is proposed. Based on laser profiling technique, this framework can achieve compact size and infrastructure-free 3D reconstruction with high definition. A hardware structure, a software pipeline, and a SLAM method consists this comprehensive framework. Unlike conventional sensors that requires multiple cameras to perform visual content tracking and 3D geometry generation such as Microsoft Kinect and Intel RealSense, a novel imaging technique is proposed which allows the monocular sensor framework to capture both visual and geometric information "simultaneously". This technique relaxes the need for multiple cameras to perform photo-realistic reconstruction, and significantly reduces sensor footprint and cost.

The major contribution of this thesis is the Visual-laser-inertial SLAM (VLI-SLAM), which is tailored to laser profiling scanners. The laser scans are not only used to generate the 3D map but also to constrain the structure scale of visual odometry. A feature-laser association method is proposed, where a 2D association method on the image robustly bootstrap the SLAM system and a 3D projective association method accurately estimates the depths of some visual features using the laser point cloud. The mapping consistency issue for overlapping map regions under back-and-forth scanning is also addressed with a window-to-map tracking method. This method performs semi-rigid point cloud alignment between the sliding window and the map to account for the minor drift within the sliding window.

Under the proposed sensor framework, two different sensor prototypes are devel-

oped, each addressing a confined-space mapping challenge. A miniature sensor named Blaser targets general confined space without prior knowledge and is designed with compactness as the primary interest. Although more than ten times smaller than one of the smallest, if not the smallest, commercial offering, it is fully capable of performing infrastructure-free photo-realistic 3D reconstruction. To the best of the authors knowledge, the Blaser prototype is the most compact RGB-D photo-realistic reconstruction system for hand-held infrastructure-free 3D reconstruction, which provides a disruptive solution for a wide range of 3D scanning applications where sensor form factor and ultra short sensing range are critical. The other more specialized sensor prototype named PipeBlaser is designed for mapping inside 12-inch diameter pipe environment. Utilizing a different laser-ring profiler but the same SLAM method, the PipeBlaser can profile pipe cross sections and generate a dense map as it travels inside the pipe. Unlike the previous works in pipe mapping, the proposed sensor do not make any assumptions regarding the pipe shape and size and do not rely on external positioning devices such as tether encoder or laser range finder.

Experimental evaluation on localization demonstrated our SLAM method's performance compared to a state-of-the-art visual-inertial SLAM method. Since laser profilers has superior 3D measurement accuracy than RGB-D cameras, the reconstruction is of higher definition: the mapping comparison suggest that the Blaser sensor is able to capture finer details and sharper geometric shapes against a popular but larger COTS RGB-D camera.

Bibliography

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org. 4.3

[2] Pedro Buschinelli, Tiago Pinto, F Silva, J Santos, and A Albertazzi. Laser triangulation profilometer for inner surface inspection of 100 millimeters (4") nominal diameter. In *Journal of Physics: Conference Series*, volume 648, page 012010. IOP Publishing, 2015. 2.4

[3] 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: Towards the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. 2.3

[4] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam. *arXiv preprint arXiv:2007.11898*, 2020. 2.3

[5] Daqian Cheng, Haowen Shi, Michael Schwerin, Michelle Crivella, Lu Li, and Howie Choset. A compact and infrastructure-free confined space sensor for 3d scanning and slam. In *2020 IEEE SENSORS*, pages 1–4, 2020. doi: 10.1109/SENSORS47125.2020.9278586. 1

[6] Daqian Cheng, Haowen Shi, Albert Xu, Michael Schwerin, Michelle Crivella, Lu Li, and Howie Choset. visual-laser-inertial slam using a compact 3d scanner for confined space. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021. 1

[7] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 1, 2.1

[8] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 1, 2.2

[9] Fei Dai, Abbas Rashidi, Ioannis Brilakis, and Patricio Vela. Comparison of image-

based and time-of-flight-based technologies for three-dimensional reconstruction of infrastructure. *Journal of construction engineering and management*, 139(1): 69–79, 2013. 1

[10] Ivan Dryanovski, Matthew Klingensmith, Siddhartha S Srinivasa, and Jizhong Xiao. Large-scale, real-time 3d scene reconstruction on a mobile device. *Autonomous Robots*, 41(6):1423–1445, 2017. (document), 2.2, 2.2

[11] Csaba Ékes. New developments in multi-sensor condition assessment technologies for large diameter pipe infrastructure. In *Pipelines 2018: Utility Engineering, Surveying, and Multidisciplinary Topics*, pages 142–148. American Society of Civil Engineers Reston, VA, 2018. 2.4

[12] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 2.3

[13] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 2.3

[14] Peter Fasogbon, Luc Duvieubourg, and Ludovic Macaire. Fast laser stripe extraction for 3d metallic object measurement. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 923–927. IEEE, 2016. 2.1

[15] RB Fisher and DK Naidu. A comparison of algorithms for subpixel peak detection. In *Image technology*, pages 385–404. Springer, 1996. 3.5.2, 4.1.2

[16] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2016. 4.1.3, 4.3.3

[17] Simon Fuhrmann and Michael Goesele. Fusion of depth maps with multiple scales. *ACM Transactions on Graphics (TOG)*, 30(6):1–8, 2011. 2.2

[18] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. Openvins: A research platform for visual-inertial estimation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4666–4672. IEEE, 2020. 2.3, 4.3.4

[19] Jason Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011. 2.2

[20] Yuanzheng Gong, Richard S Johnston, C David Melville, and Eric J Seibel. Axial-stereo 3-d optical metrology for inner profile of pipes using a scanning laser endoscope. *International journal of optomechatronics*, 9(3):238–247, 2015. 2.4

[21] Amal Gunatilake, Lasitha Piyathilaka, Sarath Kodagoda, Stephen Barclay, and

Dammika Vitanage. Real-time 3d profiling with rgb-d mapping in pipelines using stereo camera vision and structured ir laser ring. In *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 916–921. IEEE, 2019. 2.4

[22] Peter Hansen, Hatem Alismail, Peter Rander, and Brett Browning. Visual mapping for natural gas pipe inspection. *The International Journal of Robotics Research*, 34(4-5):532–558, 2015. 2.4

[23] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997. 4.2

[24] Phil Helsel. 2 dead, 2 injured in texas gas pipeline explosion. *NBC News*, Jun 2021. URL https://www.nbcnews.com/news/us-news/two-dead-others-hurt-texas-gas-explosion-n1272569. 1

[25] Lionel Heng, Paul Furgale, and Marc Pollefeys. Leveraging image-based localization for infrastructure-based calibration of a multi-camera rig. *Journal of Field Robotics*, 32(5):775–802, 2015. 3.3

[26] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012. 4.3.4

[27] Sho Kagami, Hajime Taira, Naoyuki Miyashita, Akihiko Torii, and Masatoshi Okutomi. 3d pipe network reconstruction based on structure from motion with incremental conic shape detection and cylindrical constraint. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, pages 1345–1352. IEEE, 2020. 2.4

[28] Juho Kannala and Sami 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*, 28(8):1335–1340, 2006. 3.3

[29] Juho Kannala, Sami S Brandt, and Janne Heikkilä. Measuring and modelling sewer pipes from video. *Machine Vision and Applications*, 19(2):73–83, 2008. 2.4

[30] S. Katayose, Y. Kurata, K. Watanabe, R. Kasahara, M. Itoh, D. Watanabe, K. Matsuo, and K. Hanano. Ultra-compact 3d measurement module using silica-based plc. In *2019 24th Microoptics Conference (MOC)*, pages 84–85, 2019. doi: 10.23919/MOC46630.2019.8982901. 1

[31] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 1–8. IEEE, 2013. 2.2, 4.5, 4.6

[32] Klaas Klasing, Daniel Althoff, Dirk Wollherr, and Martin Buss. Comparison of

surface normal estimation methods for range sensing applications. In *2009 IEEE international conference on robotics and automation*, pages 3206–3211. IEEE, 2009. 4.5

[33] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. 4.1.1

[34] Matthew Klingensmith, Ivan Dryanovski, Siddhartha S Srinivasa, and Jizhong Xiao. Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields. In *Robotics: science and systems*, volume 4, page 1. Citeseer, 2015. 1, 2.2

[35] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019. 1, 6.1.2

[36] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81 (2):155, 2009. 3.3, 4.2

[37] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015. 2.3

[38] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, et al. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144, 2000. (document), 2.1, 2.1

[39] Yonggen Ling and Shaojie Shen. Real-time dense mapping for online processing and navigation. *Journal of Field Robotics*, 36(5):1004–1036, 2019. 1

[40] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981. 4.1.1

[41] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. Methods for non-linear least squares problems. 2004. 4.7

[42] M. Massot-Campos, G. Oliver-Codina, and B. Thornton. Laser stripe bathymetry using particle filter slam. In *OCEANS 2019 - Marseille*, pages 1–7, 2019. doi: 10.1109/OCEANSE.2019.8867106. 1

[43] Miquel Massot-Campos, Gabriel Oliver, Adrian Bodenmann, and Blair Thornton. Submap bathymetric slam using structured light in underwater environments. In *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 181–188. IEEE, 2016. 1, 2.1

[44] Christopher Mei and Patrick Rives. Single view point omnidirectional camera calibration from planar grids. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3945–3950. IEEE, 2007. 3.3

[45] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 2011. 1, 2.2, 2.2, 4.6

[46] Harutoshi Ogai and Bishakh Bhattacharya. Pipe inspection robots for gas and oil pipelines. In *Pipe Inspection Robots for Structural Health and Condition Monitoring*, pages 13–43. Springer, 2018. 2.4

[47] Albert Palomer, Pere Ridao, and David Ribas. Inspection of an underwater structure using point-cloud slam with an auv and a laser scanner. *Journal of Field Robotics*, 36(8):1333–1344, 2019. 1

[48] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4): 1004–1020, 2018. 2.3, 4.1.3, 4.2, 4.3.3, 6.1

[49] Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311. IEEE, 2016. 3.3

[50] Florian Reichl, J Weiss, and Rüdiger Westermann. Memory-efficient interactive online reconstruction from depth image streams. In *Computer Graphics Forum*, volume 35, pages 108–119. Wiley Online Library, 2016. 2.2

[51] Giovanna Sansoni, Marco Trebeschi, and Franco Docchio. State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9(1):568–601, 2009. 2.1

[52] Gabe Sibley, Larry Matthies, and Gaurav Sukhatme. Sliding window filter with application to planetary landing. *Journal of Field Robotics*, 27(5):587–608, 2010. 4.3.4

[53] Nikola Stanić, Mathieu Lepot, Mélanie Catieau, Jeroen Langeveld, and François HLR Clemens. A technology for sewer pipe inspection (part 1): Design, calibration, corrections and potential application of a laser profiler. *Automation in Construction*, 75:91–107, 2017. 2.4

[54] Tim Stelloh and Tom Winter. Gas explosion in massachusetts leaves one dead. *NBC News*, Sep 2018. URL `https://www.nbcnews.com/news/us-news/nearly-40-fires-explosions-erupt-massachusetts-n909446`. 1

[55] Rahul Summan, William Jackson, Gordon Dobie, Charles MacLeod, Carmelo Mineo, Graeme West, Douglas Offin, Gary Bolton, Stephen Marshall, and Alexandre Lille. A novel visual pipework inspection system. In *AIP Conference Proceedings*, volume 1949, page 220001. AIP Publishing LLC, 2018. 2.4

[56] Lukas Von Stumberg, Vladyslav Usenko, and Daniel Cremers. Direct sparse visual-inertial odometry using dynamic marginalization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2510–2517. IEEE, 2018. 4.3.4

[57] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous: Spatially extended kinectfusion. 2012. 2.2

[58] Thomas Whelan, Stefan Leutenegger, R Salas-Moreno, Ben Glocker, and Andrew Davison. Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems, 2015. (document), 1, 2.2, 2.2, 4.5, 4.6

[59] Simon Winkelbach, Sven Molkenstruck, and Friedrich M Wahl. Low-cost laser range scanner and fast surface registration approach. In *Joint Pattern Recognition Symposium*, pages 718–728. Springer, 2006. (document), 2.1, 2.1

[60] Lyubomir Zagorchev and Ardeshir Goshtasby. A paintbrush laser range scanner. *Computer Vision and Image Understanding*, 101(2):65–86, 2006. 2.1

[61] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2):401–416, 2017. 1

[62] Song Zhang. High-speed 3d shape measurement with structured light methods: A review. *Optics and Lasers in Engineering*, 106:119–131, 2018. 2.2

[63] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3d reconstruction with rgb-d cameras. In *Computer graphics forum*, volume 37, pages 625–652. Wiley Online Library, 2018. 2.2