Master's Thesis

On-the-fly Targetless Extrinsics Calibration For Multi-Stereo Systems Without Field-of-View Overlap

Chenfeng Tu

CMU-RI-TR-20-33

August 2020

Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Michael Kaess, Chair
George Kantor
Alexander Spitzer

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

Keywords: extrinsics calibration; visual odometry; on-the-fly calibration

Abstract

We propose an on-the-fly extrinsics calibration method for stereo pairs lacking overlapping field of view (FoV) that is robust to visual odometry errors. Multi-stereo systems are becoming increasingly popular because of their large field of view that benefits both state estimation and mapping. A critical challenge in multi-stereo systems is to calibrate the extrinsics among the stereo pairs, which becomes even more difficult when the FoV of different stereo pairs does not overlap. Moreover, due to external forces (e.g., impact or vibration) or changes of the environment (e.g., temperature or pressure), the extrinsics can change over time. As a result, on-the-fly calibration of the extrinsics is necessary. We propose an on-the-fly targetless extrinsic calibration method for a multi-stereo system without FoV overlap. Experimental results with both simulation and real-world data show that the proposed method is successful in estimating and updating the extrinsics on-the-fly and is fairly robust to outliers and degeneracy in the measurements.

Acknowledgments

I want to first thank my advisor, Michael Kaess, for his utmost patience, thoughtful guidance and unwavering support. Without his insights and helpful discussions, this thesis will have not existed. I also want to thank my committee Dr. George Kantor and Alexander Spitzer, and the numerous people who have read and helped with my thesis, all your feedback and comments have helped me improve my work. Thank you!

I would also like to thank all the members in the Robot Perception Lab: Joshua Mangelson, Joshua Jaekel, Eric Dexheimer, Eric Westman, Paloma, Ming, Monty, Jack, Suddhu, Zimo, Akshay, Wei, Allie, Akash, Tian, Allison, Anand, Prakhar, Zilin. Thanks for your help and understanding that helped me gradually overcome so many difficulties when I stepped on this unacquainted continent for the first time and started experiencing new culture and new life. I learned a lot from you, from as small as a new word to complex academic concepts, as well as your tremendous passion for research and being rigorous in work and meticulous in details. It were you that made my Master life fruitful, colorful and cheerful.

Finally, I'd like to acknowledge my gratefulness to my family. You have always been so considerate and always support every decision I make. Thank you for all your support and help.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Multi-stereo systems have emerged as a popular sensing modality for robotics applications. Compared with single stereo, multi-stereo systems usually have a larger combined field of view (FoV), which facilitates feature tracking over longer duration and increases the accuracy of state estimation algorithms. Larger FoV also contributes to the robustness and dexterity of the state estimation system because of the ability to observe more information in general [1]. Moreover, a richer map can be generated from the larger amount of data obtained by multi-stereo systems. High-accuracy system parameters, including intrinsics of each individual camera, extrinsics between the two monocular cameras in a stereo pair as well as extrinsics between stereo pairs (usually the extrinsics between the master monoculars of each stereo pair), are of vital importance to obtaining high-quality measurements and algorithms based on these parameters, such as state estimation.

Intrinsics calibration for stereo pairs is relatively easy to do, and has already been covered in many prior works [4][5]. Besides, intrinsic parameters are less prone to change, even in the event of crashing or long-term mechanical vibration [6]. So in this work, we assume the intrinsics are calibrated. For calibrating extrinsics between the two monocular cameras within a stereo pair or between stereo pairs having field-of-view overlap, it is relatively easy because we can utilize commonly seen features [6][7][8][9][10]. It is a more difficult task to calibrate the extrinsics between stereos without field of view overlap. The different system setups are shown in Fig. 1.1. In this work, we focus on calibrating the extrinsics between stereo pairs that do not have field-of-view overlap.

On-the-fly extrinsic calibration of multi-camera systems is highly desirable for a range of different products and applications, including virtual reality devices and autonomous aerial vehicles (Fig. 1.2). The calibration of a multi-camera system can be affected by impact during transportation and use, as well as by thermal expansion during normal operation, with the potential to significantly reduce performance of state estimation and other algorithms. On-the-fly calibration will be critical to maintain performance. On the other hand, although special targets are common for lab/factory calibration, they are not applicable when doing on-the-fly extrinsics calibration as

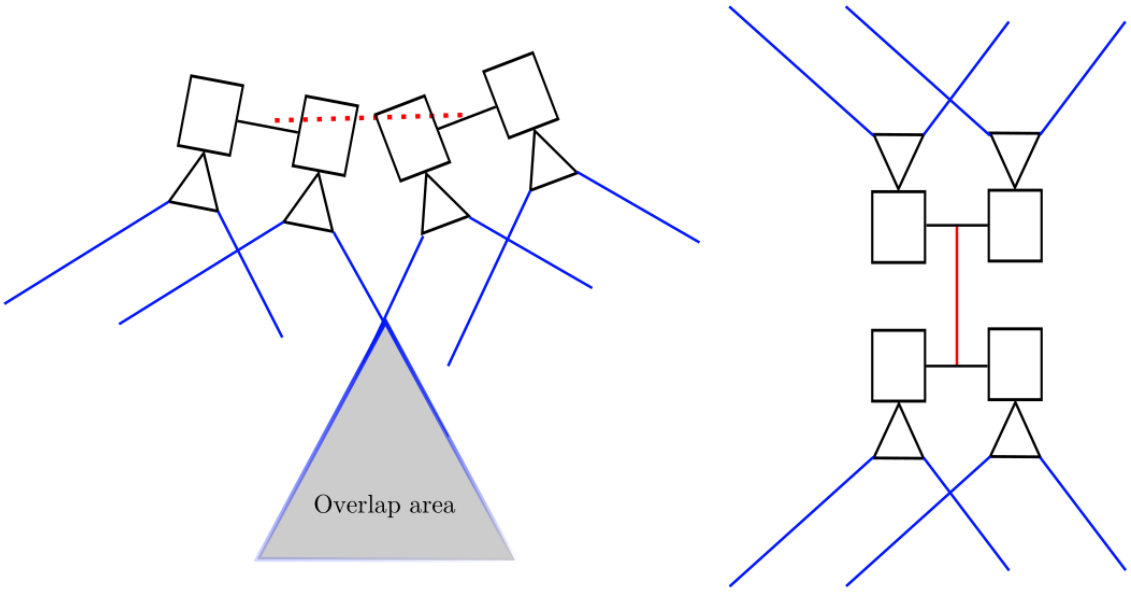

Figure 1.1: Examples of multi-stereo systems with different relative poses. *(left)* Two stereo pairs with field-of-view overlap. *(right)* Two stereo pairs without field-of-view overlap, the main focus of this work.

the system keeps moving and there is no guarantee that the special targets are always visible.

1.2 Current Challenges

Adopting multiple stereo cameras for state estimation and mapping has become a popular approach in recent years [1]. However, calibrating the extrinsics among all the stereo cameras in a multi-stereo system is a critical problem. Accurate extrinsics calibration can enhance the performance of data fusion across sensors.

The relative poses of stereo cameras in a multi-stereo system can be set up in various ways, which can be generally divided into two categories in terms of the calibration method: with FoV overlap or without FoV overlap. For the systems with FoV overlap, the extrinsics between stereo cameras can be calibrated utilizing the overlapping area in the images, which is similar to calibrating a single stereo pair. On the other hand, it is more difficult to calibrate systems without FoV overlap because there are no common features for use. Although it is possible to rotate the rig in order to see the same landmarks at different time, the rotation itself needs to be estimated, which will introduce additional error in the extrinsics estimation. Besides, the requirement of calibrating on the fly further increases the difficulty of the problem as the views keep changing and no specific calibration target can be assumed to be visible all the time. The third challenge is to ensure robustness. State estimation is very sensitive to extrinsics, small offsets could lead to large error or even make the system fail. So it is important to evaluate the confidence we have on

Figure 1.2: Examples of multi camera systems that require extrinsics calibration. *(left)* Facebook Oculus headset with camera sensor facing different directions. *(right)* Drone with two stereo camera sensors [1].

the extrinsics result before we can safely apply it to use. This includes dealing with the degeneracy in the calibration process and modeling the uncertainty of the calibrated extrinsics.

1.3 Contribution and Organization

To solve all the challenges mentioned above, we propose an on-the-fly targetless extrinsics calibration method for stereo pairs lacking overlapping field of view.

The contributions of this work are:

- We propose a targetless extrinsic calibration method for a rig consisting of multiple stereo pairs which can operate on the fly.
- We propose methods for dealing with inaccurate measurements and degeneracy when updating the extrinsics currently in use.
- We present comparison between the choice of including landmarks into factor graph as variables or not.
- Experiments are presented with both simulation and real-world data to show that the calibration performance is as desired and the method can work well regardless of the rig's trajectory.

The thesis is organized as follows: Section 1 gives a basic introduction to the motivation and contributions of the work. Section 2 and Section 3 overview the current existing methods for on-the-fly extrinsics calibration and introduce preliminary theory on visual odometry, factor graphs and the pairwise consistent measurement set maximization algorithm utilized in ensuring robustness. Section 4 details the whole system and Section 5 presents experiment results on showing the effectiveness and robustness of our method. Finally, Section 6 and Section 7 recap the contributions of the thesis and discuss future research directions, respectively.

Chapter 2

Related Work

In this section, we introduce related work on general camera calibration, extrinsics calibration and the different methods proposed when dealing with scenarios with/without camera field of view overlap.

2.1 Camera Calibration

The goal of camera calibration is to estimate the intrinsic and extrinsic parameters of a camera from one or more images. It is a necessary and important step in 3D computer vision and fundamental to applications based on these parameters, such as state estimation.

Techniques for single camera calibration can be divided into two categories [4]:
1. Target-based calibration. Calibration target with known pattern and 3D geometry parameters is needed to perform camera calibration [5][11].
2. Self-calibration. Techniques belonging to this category do not utilize any external special-designed targets, but utilize the rigidity of the scene to give constraints needed [12][13][14][15].

2.2 Extrinsics Calibration With Field-of-View Overlap

For calibrating the internal extrinsics between the sensors within a sensor rig, such as a stereo camera, the problem belongs to multiple view geometry and has been discussed much in the literature [16][17].

For calibrating sensors with field-of-view overlap, one set of works typically use a checkerboard or other patterned calibration targets [18].

Another line of research utilizes refine-based methods to refine the extrinsics on the fly, usually under some prior assumption on the extrinsics. Levinson et al. proposed a method to calibrate cameras and LiDAR online [19], by extracting features from the point clouds and images and align the features from different sensor modalities. They assumed they had a fairly precise calibration

at the beginning and then they can calibrate online by detecting the change of the alignment cost. Ling et al. [6] proposed a similar optimization-based method to calibrate a stereo rig. They also derived a mathematical expression for computing the covariance of stereo extrinsic estimates to identify when the calibration result is sufficiently accurate [20] [7].

Ishikawa et al. [21] introduced a motion-based 2D-3D calibration method to calibrate camera and LiDAR using motions estimated by sensor fusion odometry. They also talked about suitable motions to improve their calibration method [9] [8].

The difference of extrinsics calibration between with and without field of view is obvious. In the case of without field of view, we cannot utilize the common features seen by both of the sensors anymore, which makes the calibration problem more difficult.

2.3 Extrinsics Calibration Without Field-of-View Overlap

Even though there is an extensive body of work on calibrating multi-sensor systems, few approaches have been focusing on systems that do not have FoV overlap.

To perform extrinsic calibration between a stereo camera and LiDARs that do not have common FoV, Scott et al. [22] and Jeong et al. [23] exploit road markings as static and robust features among the various objects that are present in urban environments. The road markings are used to align sensor measurements to then estimate the extrinsics when some common landmarks appear. However, their method requires high accuracy of the relative vehicle pose estimation given by high precision odometry sensor.

Some works use mirrors to create an overlapping FoV [24] [25][26]. After creating the FoV overlap, traditional calibration methods using geometric calibration targets are used. The drawback of this approach is that it cannot be used for general online estimation because of the need for a mirror.

Another line of study estimates the motion of each sensor and performs extrinsic calibration based on the estimated motion relationship, which is theoretically similar to hand-eye calibration [1][27].

Lébraly et al. [28] proposed a method resembling hand-eye calibration, based on specific bundle adjustment. However, their method is designed for offline calibration and has the limitation of depending on specific static circular landmarks. Besides, their method does not provide a solution to deal with outliers in measurements and a solution to update the extrinsics, which is required for online application.

There are also studies in which extrinsics are calibrated utilizing other sensors, such as Inertial Measurement Unit (IMU) [29][30]. The introduction of IMU reduces the difficulty of the problem

due to non-overlapping FoV but makes the solution less general, and not applicable to scenario without an IMU.

Chapter 3

Preliminaries

This section acts as a primer on visual odometry and introduces the reader to factor graph representation of an inference optimization problem. This section also introduces the pairwise consistent measurement set maximization algorithm used in ensuring the robustness of our extrinsics calibration method.

3.1 Visual Odometry and Factor Graph

Visual Odometry

Visual odometry (VO) is the process of estimating the egomotion of an agent (e.g., vehicle, human, and robot) using only the input of a single or multiple cameras attached to it [31][32]. Similar to wheel odometry, visual odometry estimates the motion of the system by estimating the relative transformation between two consecutive time points. Compared to wheel odometry, which uses the difference of wheel encoder readings to track distance traveled, visual odometry uses visual differences between two consecutive images to track the transformation. In order for visual odometry to work, essential patterns for feature extraction [33] or light intensity difference [34] are needed.

Generally, visual odometry can be classified into feature-based methods and direct methods, depending on whether features are extracted from images. The pipeline of a visual odometry system is summarized in Fig. 3.1.

For feature-based methods, such as ORB feature [33], the motion of the camera, expressed as $(\mathbf{R}_k, \mathbf{t}_k)$, can be calculated in many ways, depending on the representation of correspondence types, such as 2D-2D, 3D-2D, 3D-3D. Here $\mathbf{R}_k \in \mathrm{SO}(3)$ and $\mathbf{t}_k$ is the rotation matrix and the translation vector respectively.

In the monocular case, especially at the initialization step, 2D-2D epipolar geometry is usually utilized as there are no 3D correspondences available. Let p be an image feature, the relationship

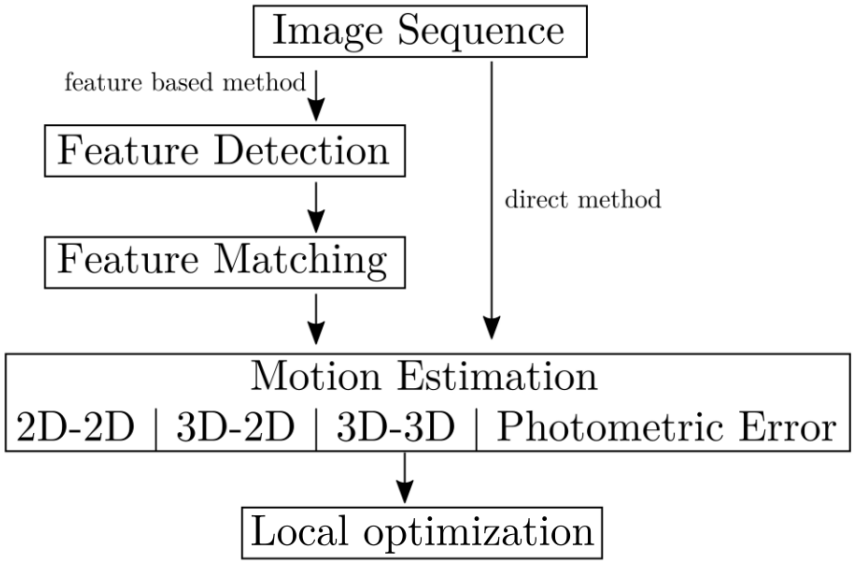

Figure 3.1: Pipeline of a visual odometry system

is expressed in a concise equation:

$$p_k^T K^{-T} [\mathbf{t}]_\times \mathbf{R} K^{-1} p_{k-1} = 0 \tag{3.1}$$

Here K is the intrinsic matrix of the camera. With robust methods, such as 7-point or 8-point algorithm [17], camera motion can be recovered algebraically, up to four candidate solutions with scale ambiguity (i.e. the absolute length of the translation vector is unknown). The four candidate solutions can be further verified by triangulating the features and examining the relative position of the 3D features to the cameras. The solution is based on minimizing the least squares epipolar residual:

$$e(\mathbf{R}, \mathbf{t}) = \sum_i \left\| p_{i,k}^\top \cdot [\mathbf{t}]_\times \mathbf{R} \cdot p_{i,k-1} \right\|^2 \tag{3.2}$$

For stereo camera setup, a 3D-2D correspondence approach is usually adopted to estimate camera motion, which is also referred to as $perspective - n - point$ (PnP) problem. Firstly, the 3D locations of the features are triangulated, and then they are re-projected to the image at the new pose. $P_k(p) \in \mathbb{R}^3$ is the corresponding 3D coordinates in frame k, the 3D-2D method minimizes the re-projection error function:

$$e(\mathbf{R}, \mathbf{t}) = \sum_i \left\| p_{k,i} - \pi \left(\mathbf{R} P_{k-1} \left(p_i \right) + \mathbf{t} \right) \right\|_{\Sigma_i}^2 \tag{3.3}$$

where $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ projects 3D landmarks to image pixel coordinates and Σ_i is the error covariance matrix of feature p. In order to get the optimal solution, optimization based methods, such as the Gauss-Newton method or the Levenberg-Marquardt algorithm are used.

Compared to visual SLAM (simultaneous localization and mapping), the difference between visual odometry and visual SLAM is that visual odometry mainly focuses on local consistency,

trying to estimate the trajectory of the camera incrementally, while SLAM also pays attention to achieving global consistency and maintaining a global map. Global optimization is usually applied in SLAM.

Factor Graph

A factor graph is a commonly used representation to represent an inference problem. A factor graph is a bipartite graph comprised of *variables* to be optimized and *factors* that constrain the system. The variable nodes represent the state we wish to estimate and the factors are the measurements obtained from sensors. An example factor graph taken from [2], is shown in Fig. 3.2. This factor graph represents a SLAM problem where the state $\mathcal{X}$ include poses x_i and landmarks l_j. Factors, corresponding to measurements, can be binary factors or unary factors, depending on the type of the measurements. For example, odometry, camera measurements, and loop closures are binary factors connecting two variable nodes; GPS measurements, pose priors are unary factors only attached to one variable.

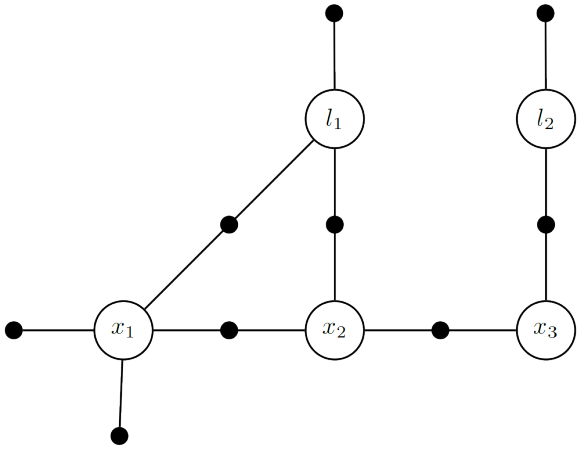

Figure 3.2: A example factor graph including poses and landmarks, figure taken from [2]. Variable nodes are the hollow circles in this figure, representing poses (x_i) or landmarks (l_i). Factors are represented by smaller black circles.

Solving the factor graph is equal to computing the *maximum a posteriori* (MAP) estimation $\mathcal{X}^*$, which is the state that maximally agrees with the given measurements:

$$\begin{aligned} \mathcal{X}^* &= \underset{\mathcal{X}}{\operatorname{argmax}} \; p(\mathcal{X}|\mathcal{Z}) \\ &= \underset{\mathcal{X}}{\operatorname{argmax}} \; p(\mathcal{X})\, p(\mathcal{Z}|\mathcal{X}) \\ &= \underset{\mathcal{X}}{\operatorname{argmax}} \; p(\mathcal{X})\, l(\mathcal{X};\mathcal{Z}) \\ &= \underset{\mathcal{X}}{\operatorname{argmax}} \; p(\mathcal{X}) \prod_{i=1}^{N} l(\mathcal{X};\mathbf{z_i}) \end{aligned} \tag{3.4}$$

here $l(\mathcal{X};\mathcal{Z})$ is proportional to $p(\mathcal{Z}|\mathcal{X})$ and denotes the likelihood of state $\mathcal{X}$ given measurements $\mathcal{Z}$. In the last step, we factor it into a product over individual measurements z_i. This is based on the assumption of conditional independence of measurements, which is encoded in the factor graph (Fig. 3.2). In Section 4.2, we will utilize factor graph as a tool to model the extrinsics calibration problem.

3.2 Pairwise Consistent Measurement Set Maximization

Outliers are common in measurements. A difficult problem in conducting inference is dealing with outliers as the result of least squares can be susceptible to outliers. A variety of methods have been proposed to get rid of the outliers and enhance the robustness of inference [35][36][37]. Mangelson et al. proposed a method to eliminate outlier measurements based on pairwise consistent measurements [3]. They formulate this problem as a combinatorial optimization problem, and then present a method that finds the optimal solution by extracting a maximized clique in this graph that are pairwise internally consistent sets. An example process for extracting the clique from loop closure measurements is shown in Fig. 3.3.

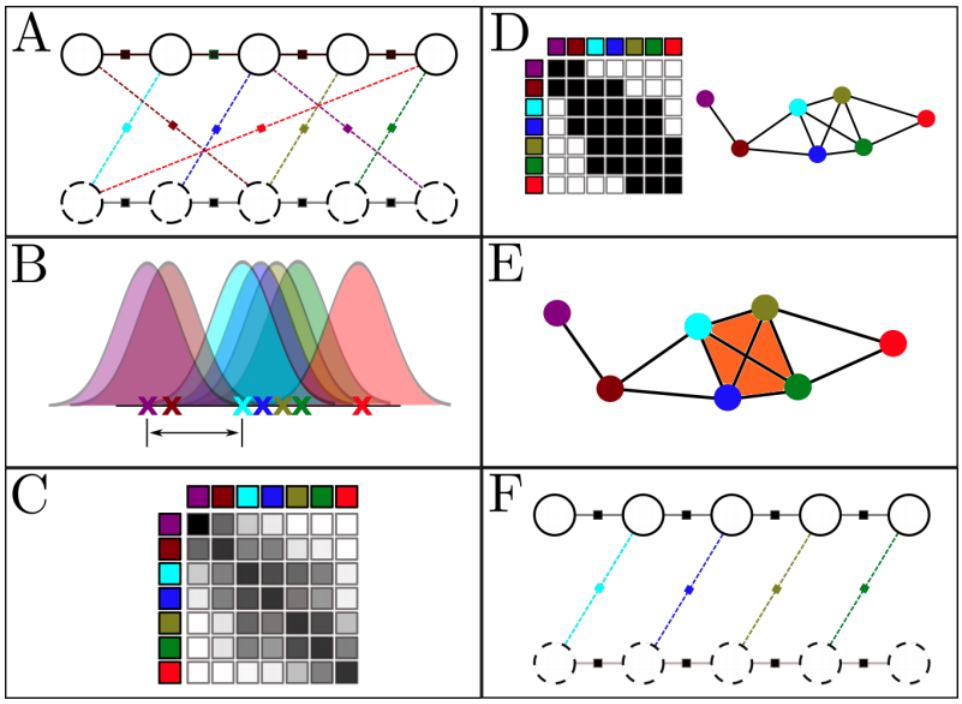

Figure 3.3: An example of the Pairwise Consistency Maximization (PCM) algorithm for selecting consistent inter-map loop closures measurements, image taken from [3].

Firstly, when given two independently derived pose graphs (shown in white and black in step A in Fig. 3.3) and a set of potential loop closures between them (shown by colored, dotted lines), they try to determine loop closures that are reliable and consistent with each other. Secondly, as

shown in step B, they use a consistency metric such as Mahalanobis distance, and they calculate the consistency value of each pairwise combination of measurements. Then in C they organize these pairwise consistency values in a matrix (could be asymmetrical, depending on the metric chosen), where each element corresponds to the consistency of a pair of measurements. In order to get a set of nodes in which nodes have mutual similarity above some thereshold, they threshold and truncate the consistency values, transforming the matrix into the adjacency matrix for a consistency graph. The matrix is also made symmetric by using the maximum consistency value when associated elements have different consistency values. In the graph in steps D and E, each node represents a measurement and edges represent consistency between measurements. Cliques in this graph are pairwise internally consistent sets. In step E, finding the maximum clique equals to finding the largest pairwise internally consistent set of measurements. The problem of finding the maximum clique for a given graph is an NP-hard problem [38], a pruning-based algorithm [39] is adopted to find a solution relatively quickly.

Using the consistent measurements from the clique, inference can be made more robust and reliable. We will utilize pairwise consistent measurement set maximization in Section 4.3.2 to extract consistent and reliable measurements, making our method more robust to outlier measurements.

Chapter 4

On-the-fly Targetless Extrinsics Calibration For Multi-Stereo System Without FOV Overlap

4.1 Problem Formulation and System Overview

For clarity, throughout this paper, we refer to two monocular cameras facing the same direction as a stereo pair. We refer to multiple stereo pairs that are rigidly mounted and face different directions as a multi-stereo system. Fig. 4.1 shows a three-stereo-pair system.

In our work, we mainly focus on systems that consist of two stereos pairs, as for systems with more than two stereo pairs, we can always choose a pivotal stereo pair and calibrate the others to the pivotal stereo pair. The pivotal stereo pairs act as the calibration reference for all other stereo pairs.

As shown in Fig. 4.1, $stereo1$ and $stereo2$ are rigidly connected to each other but facing opposite directions, without any FoV overlap. We assume all cameras are time synchronized. At timestamp i, the pose $\mathbf{T}_{[s],i} \in \mathrm{SE}(3)$ of each stereo pair s is defined as the pose of the left camera (also referred to as master camera) in the robot's body frame. The body frame of the robot is defined by the pose of the first stereo pair $\mathbf{T}_{[1],i}$. The transformation between two poses $\mathbf{T}_{[s],i}$ and $\mathbf{T}_{[s],j}$ of camera s is denoted as $\mathbf{T}_{[s],ij} = \mathbf{T}_{[s],i}^{-1}\mathbf{T}_{[s],j}$. The extrinsics between two stereo pairs s_1, s_2 at timestamp i is denoted as $\mathbf{E}_{[s_1,s_2],i} = \mathbf{T}_{[s_1],i}^{-1}\mathbf{T}_{[s_2],i} \in \mathrm{SE}(3)$ so that any point p in the coordinate frame of stereo pair s_2 is transformed into the coordinate frame of pair s_1 by $p' = \mathbf{E}_{[s_1,s_2],i}p$.

We want to calibrate the extrinsics between the two stereo pairs, while assuming the intrinsic parameters of each camera and extrinsics within each stereo pair are known and fixed. This is reasonable as it is comparatively much easier to calibrate the intrinsics and extrinsics of a single stereo pair than it is to calibrate cameras without any FoV overlap, and there are already many works that address this problem well [6]. More importantly, they are less likely to change because of temperature or impact.

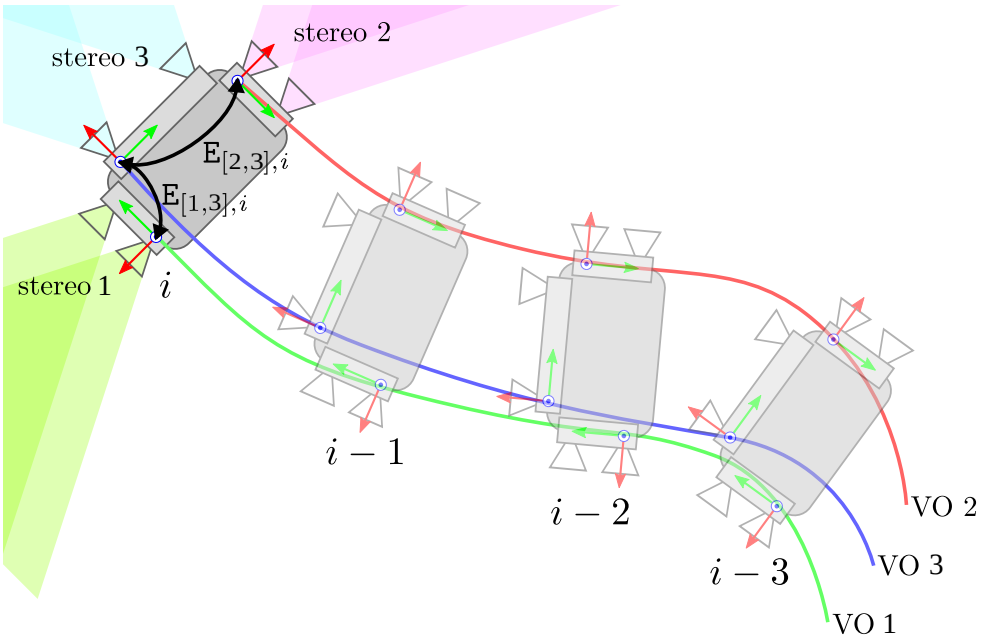

Figure 4.1: Overview of the proposed on-the-fly extrinsic calibration system. The rig consists of multiple stereo pairs without any FoV overlap, each generating their own visual odometry estimate. The goal of our method is to calibrate $\mathbf{E}$.

State estimation algorithms for multi-stereo systems depend heavily on the accuracy of extrinsics between stereos [40]. However, it is dangerous to do joint optimization and estimate the extrinsics and the system state at the same time, since small perturbation of extrinsics might have great impact on the pose estimation, and causing state parameters stuck in local minimum. Thus, we establish a separate extrinsics estimation system apart from the pose state estimation algorithm. We only update the extrinsics when we are confident about the newly estimated extrinsics.

4.2 Factor Graph Representation

We propose an on-the-fly extrinsics calibration method based on factor graph representation. The same stereo visual odometry algorithm is run on the two stereo pairs separately. Our extrinsics calibration works separately from the two visual odometry systems. While utilizing the output from them, our algorithm does not have direct influence on the visual odometry systems run on the stereo pairs. Based on the difference of incorporating landmarks into the factor graph optimization or not, we divide our method into a visual-odometry-based method and a bundle-adjustment-based method. We then compare the difference between these two representations.

4.2.1 Visual-Odometry-Based Factor Graph

If we only incorporate the camera poses from the visual odometry systems, along with their corresponding pose covariances, we can represent the on-the-fly extrinsic estimation problem as a

factor graph as shown in Fig. 4.2.

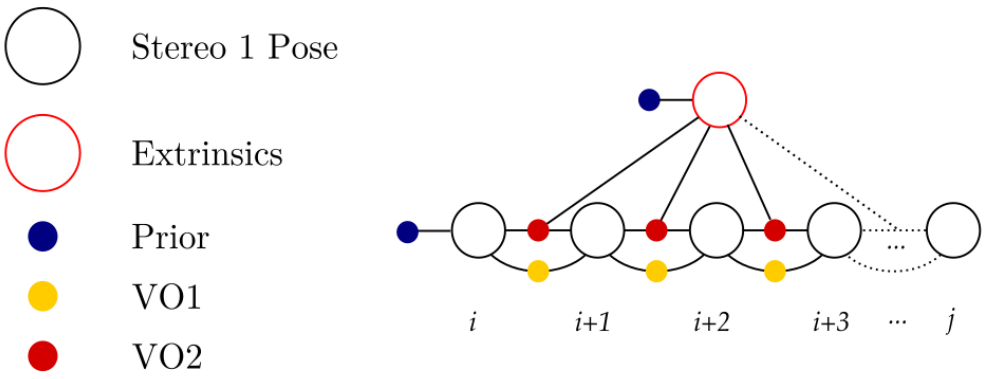

Figure 4.2: Factor graph for on-the-fly extrinsics calibration with only visual odometry constraints.

In this factor graph, there are two types of variable nodes: camera pose variable node and extrinsics variable node. A camera pose variable node represents the pose of $stereo1$ and the extrinsics variable node represents the extrinsics $\mathbf{E}_{[s_1,s_2],i}$ between $stereo1$ and $stereo2$. The three types of factor nodes in this factor graph are prior factor, factor of visual odometry constraint from $stereo1$ and $stereo2$ respectively. The very first pose variable is constrained by a prior factor representing the preliminary estimation to the state of $stereo1$. The extrinsics variable node is also constrained by a prior factor, representing our initial guess for this parameter. For the visual odometry (VO) factors from $stereo1$, each factor connects two consecutive pose variable nodes, representing the relative transformation $\mathbf{T}_{[s1],i(i+1)}$ between the two poses of $stereo1$ at time i and $i+1$ respectively. The error term for VO1 factor between time i and $i+1$ is:

$$e_{[s1],i(i+1)} = ||z_{[s1],i(i+1)} - \mathbf{T}^{-1}_{[s1],i}\mathbf{T}_{[s1],i+1}||^2_{\Sigma_{[s1],i(i+1)}} \tag{4.1}$$

Here $z_{[s1],i(i+1)}$ is the transformation measurement from VO1 and $\Sigma_{[s1],i(i+1)}$ is the corresponding measurement covariance matrix.

Note that the pose of $stereo2$ is not present in this factor graph. This formulation of only including the poses of $stereo1$ allows us to model the problem with a minimal number of variables to simplify the graph and achieve more efficient optimization. So, the error term for VO2 factor should be comparing measurement from $stereo2$ with transformation of $stereo2$, which is represented by $stereo1$ transformed via extrinsics variable. The error term for VO2 factor between time i and $i+1$ is:

$$e_{[s2],i(i+1)} = ||z_{[s2],i(i+1)} - (\mathbf{E}\mathbf{T}_{[s1],i})^{-1}(\mathbf{E}\mathbf{T}_{[s1],i+1})||^2_{\Sigma_{[s2],i(i+1)}} \tag{4.2}$$

Here $z_{[s2],i(i+1)}$ is the transformation measurement from VO2 and $\Sigma_{[s2],i(i+1)}$ is the corresponding measurement covariance matrix.

We assume the extrinsics remain fixed within the time period from i to j in the graph shown in Fig. 4.2. This is reasonable because the extrinsics will not change drastically and usually remain fixed within a short period. Even if it does change in this period, our window based selection method will classify it as outlier and this window will not be used to estimate the extrinsics, which will be covered in Section 4.3.

4.2.2 Bundle-Adjustment-Based Factor Graph

Another option of formulating the extrinsics calibration problem is forming a factor graph with the observed landmarks explicitly modeled in the graph, as shown in Fig. 4.3. This is similar to traditional bundle adjustment (BA) [41].

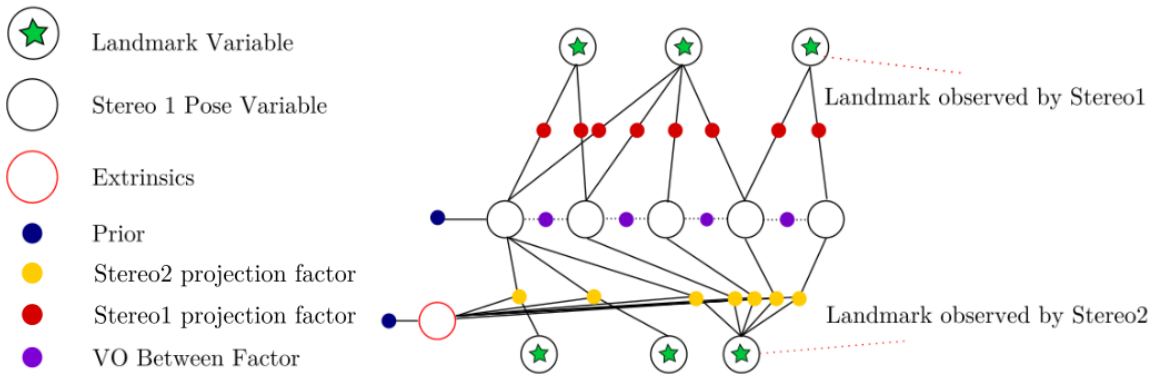

Figure 4.3: Factor graph for on-the-fly extrinsics calibration in the form of bundle adjustment, with landmarks constrained.

In this factor graph, apart from camera pose variable node and extrinsics variable node, we introduce another type of variable node, landmark variable node, which represents the locations of the landmarks. Different from our previous visual odometry factor connecting two consecutive pose variable nodes, in this factor graph we have projection factors connecting landmark variable node and pose variable node. The connection of these factors is determined by the observation of the landmarks at specific camera pose. The error term for $stereo1$ projection factor between pose variable node i and landmark variable node j is:

$$e_{[s_1],ij} = ||z_{[s_1],ij} - h(\mathbf{L}_j, \mathbf{T}_{[s_1],i})||^2_{\Sigma_{[s_1],ij}} \tag{4.3}$$

And the error term for $stereo2$ projection factors between pose variable node i and landmark variable node j is:

$$e_{[s_2],ij} = ||z_{[s_2],ij} - h(\mathbf{L}_j, \mathbf{ET}_{[s_1],i})||^2_{\Sigma_{[s_2],ij}} \tag{4.4}$$

Here $z_{[s_2],ij}$ is a four-element vector denoting the pixel measurements of landmark j on both left and right camera of $stereo2$. $h(\mathbf{L}_j, \mathbf{T}_{[s],i})$ is the projection function projecting landmark j to camera s at i.

Note that simply constructing the factor graph only with $stereo1$ and $stereo2$ projection factors is not enough to fully constrain the factor graph. It is still possible that in some cases, certain poses would not be fully constrained due to lack of landmark observations. For example, as shown in Fig. 4.4, camera pose nodes starting from $i+1$ are not fully constrained because pose node $i+1$ only observes one common landmark with its previous pose node at i, making it lack 2 DoF and leads to an unsolvable system. In order to fully constrain the graph, making the factor graph solvable, we also add factors between consecutive pose variable nodes. These factors are initialized with readings from VO1 and have very large covariances, which means we trust little on these measurements but they can help avoid numerical issues.

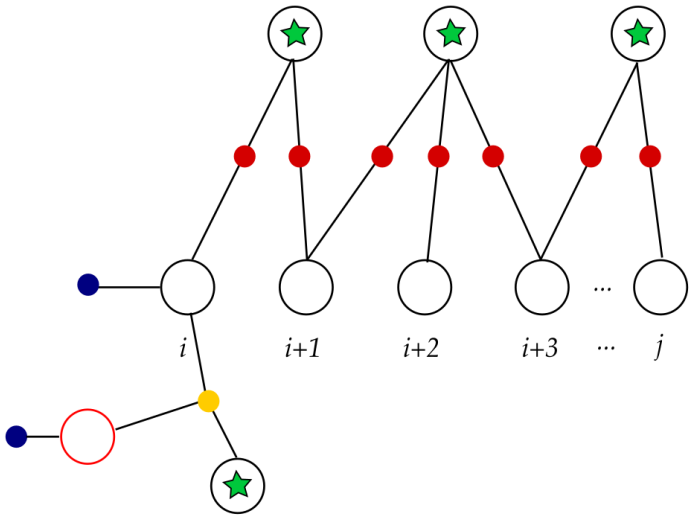

Figure 4.4: Factor graph in which some poses are not fully constrained. Camera pose nodes starting from $i+1$ are not fully constrained because pose node $i+1$ only observes one common landmark with its previous pose node at i, making it lack 2 DoF and leads to an unsolvable system.

Although having a simpler graph structure, visual odometry based factor graph has some problems in practice. Firstly, estimating the covariance of the VO factor is relatively hard. Secondly, comparing with bundle adjustment based factor graph, VO based factor graph needs longer time duration to get similarly good calibration. So, in the next sections, we will mainly talk about factor graph based on bundle adjustment. We will also provide some further discussion on the differences between these two graphs in the experiment Section 5.

4.3 Window Grouping and Window Selection

Calibration of extrinsics is very sensitive to outliers, but visual odometry suffers from outlier measurements due to a lot of reasons, such as feature mismatching. One outlier measurement could lead to drastic change in the calibration result. In order to get rid of the influence from outlier measurements as much as possible and increase the robustness of our method, we further propose an extrinsics calibration pipeline based on window grouping and window selection.

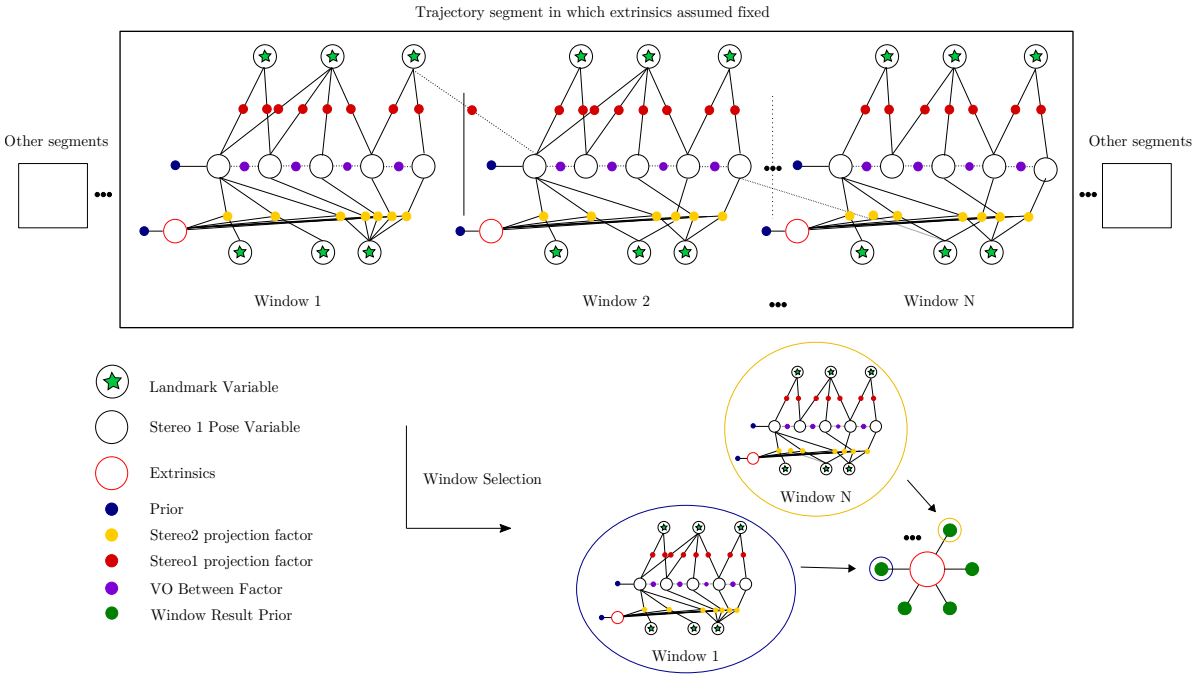

Figure 4.5: Window grouping and window selection pipeline in calibrating a two-stereo system using a bundle adjustment based factor graph. In the upper part of the figure, nodes are grouped into windows and each window estimates the extrinsics separately. We assume that throughout the segment, which consists of N windows, the extrinsics are consistent. But each window may have a different estimation due to outlier measurements or insufficient features. In the lower part of the figure, after filtering inaccurate estimations through window selection processes described in Section 4.3.1 and Section 4.3.2, consistent estimations from multiple windows are combined and optimized together in a final factor graph to decrease possible degeneracy in each window. The final estimation result is then integrated with current extrinsics using the extrinsics update strategy described in 4.4.1.

The window grouping and window selection pipeline in calibrating a two-stereo system using a bundle adjustment based factor graph is shown in Fig. 4.5. We split the trajectory of the system into discrete segments. Between segments, the extrinsic parameters could vary from each other. Extrinsics estimated from previous trajectory segment is utilized as a prior factor for the extrinsics parameter in the next trajectory segment.

As shown in the upper part of the Fig. 4.5, nodes are grouped into windows and each window estimates the extrinsics separately. We assume that throughout the segment, which consists of N windows, the extrinsics are consistent. But each window may have a different extrinsics estimation due to outlier measurements or insufficient features. In each window, there are fixed number of camera pose variables and the landmark connection relationship is determined by a connection graph extracted from the visual odometry system. All the landmarks observed by the camera poses within the window are included as landmark constraints.

In the lower part of Fig. 4.5, after filtering inaccurate estimations through window selection processes described in Section 4.3.1 and Section 4.3.2, consistent estimations from multiple windows are combined and optimized together in a final factor graph. Note that the problem

set up with the measurements in each window can be degenerate, the estimation result can also be uncertain in some dimensions [42], and different windows can be degenerate in different directions. So apart from optimizing using the final factor graph, which fuses the results from each window and decreases degeneracy in all directions, the result still needs to go through a threshold checking using the extrinsics update strategy described in 4.4.1.

This pipeline can be extended to more than two stereo pairs by adding extrinsics variable node and landmark variable node constraints that correspond to the additional stereo pairs.

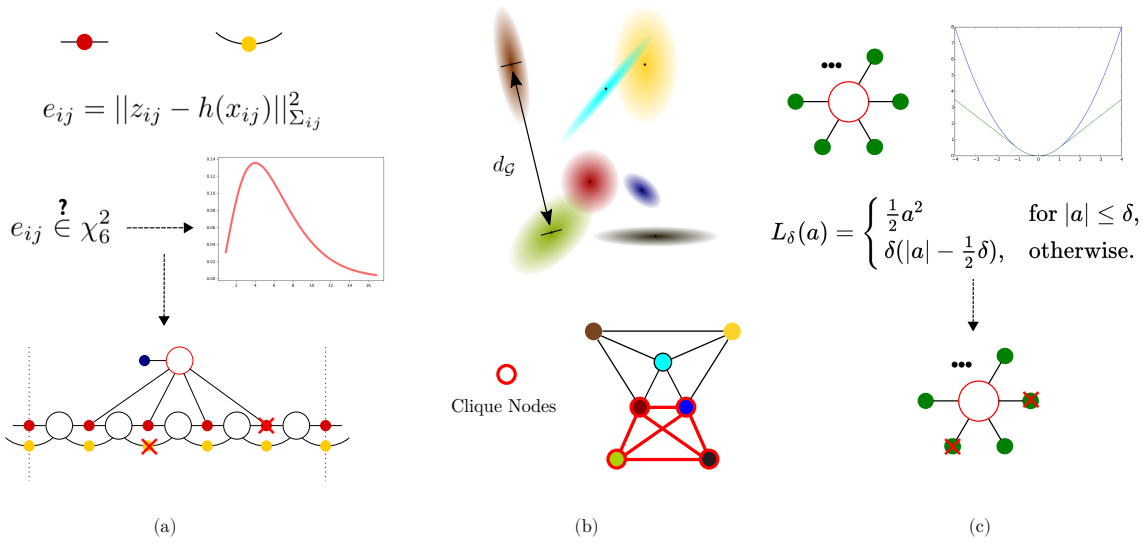

Figure 4.6: Window selection methods. (a) χ^2 based window selection: for measurements between two pose nodes and between camera pose and landmark nodes, the error follows a 6-DoF and 4-DoF χ^2 distribution respectively. When the error value exceeds the inverse of the p-value threshold, we mark the specific measurement as an outlier. If the number of outlier measurements exceeds some threshold, we discard the window. (b) Set Maximization Based Window Selection: The extrinsics estimated from each window might be different from all others. We construct a similarity graph and extract a clique that contains estimations which are reliable and consistent with each other. (c) Fuse results and apply robust cost function: when constructing the final graph to fuse the results from different windows, we apply robust function to reject possible outliers. Huber loss with threshold 0.5 is used. After optimization, we obtain extrinsics result along with its covariance matrix, which denotes the uncertainty of the estimation in each direction.

4.3.1 χ^2 Based Window Selection

This section discusses the first step in rejecting outlier measurements based on chi-square testing.

The chi-square distribution (also chi-squared or χ^2 distribution) with k degrees of freedom is the distribution of a sum of the squares of k independent standard normal random variables. For each specific $stereo1$ measurement in our visual odometry based factor graph, the error is

calculated as:

$$e_{[s_1],i(i+1)} = ||z_{[s_1],i(i+1)} - \mathbf{T}^{-1}_{[s1],i}\mathbf{T}_{[s1],i+1}||^2_{\Sigma_{[s_1],i(i+1)}} \tag{4.5}$$

$$= (\Sigma^{-\frac{1}{2}}_{[s_1],i(i+1)}(z_{[s_1],i(i+1)} - \mathbf{T}^{-1}_{[s1],i}\mathbf{T}_{[s1],i+1}))^T(\Sigma^{-\frac{1}{2}}_{[s_1],i(i+1)}(z_{[s_1],i(i+1)} - \mathbf{T}^{-1}_{[s1],i}\mathbf{T}_{[s1],i+1})) \tag{4.6}$$

in which $\Sigma_{[s_1],i(i+1)}$ is the covariance of the transformation measurement. When represented in Lie algebra, $e_{[s_1],i(i+1)}$ follows a 6-DoF χ^2 distribution,

$$e_{[s_1],i(i+1)} = ||\Sigma^{-\frac{1}{2}}_{[s_1],i(i+1)}(z_{[s_1],i(i+1)} - \mathbf{T}^{-1}_{[s1],i}\mathbf{T}_{[s1],i+1})||^2, \tag{4.7}$$

Setting p-value being 0.05 for a 6-DoF chi-square distribution, we compute the χ^2 value $X \approx 12.59$ as the threshold.

Similarly, in our bundle adjustment based factor graph, the error is calculated as:

$$e_{[s_1],ij} = ||z_{[s_1],ij} - h(\mathbf{L}_j, \mathbf{T}_{[s_1],i})||^2_{\Sigma_{[s_1],ij}} \tag{4.8}$$

$$= (\Sigma^{-\frac{1}{2}}_{[s_1],ij}(z_{[s_1],ij} - h(\mathbf{L}_j, \mathbf{T}_{[s_1],i})))^T(\Sigma^{-\frac{1}{2}}_{[s_1],ij}(z_{[s_1],ij} - h(\mathbf{L}_j, \mathbf{T}_{[s_1],i}))) \tag{4.9}$$

The error term $e_{[s_1],ij}$ follows a 4-DoF χ^2 distribution as each landmark is projected to both the left and right camera of the stereo camera, with x and y pixel coordinates on each image plane. We also compute the corresponding χ^2 value $X \approx 9.49$ as the threshold when p-value being 0.05 for a 4-DoF chi-square distribution.

So all the measurements with chi-squared value larger than X is regarded as outlier measurements. Then we count the number of outliers in each window. If the number of outliers selected by χ^2 test in a specific window exceeds a certain threshold, we drop that window for better robustness of the system. All the remaining windows are used in the next step of window selection.

4.3.2 Set Maximization Based Window Selection

Due to various reasons such as abrupt motion, there still could be windows in which the visual odometry system is not well functioning, which will lead to inaccurate extrinsics estimation. To further exclude abnormal windows, we utilize pairwise consistent measurement set maximization on a graph consisting of calibration results from each window.

After solving the factor graph in each window, the optimized extrinsics and their corresponding covariance are extracted. As shown in Fig. 4.6 (b), we analyze the similarity pattern among the results of the extrinsics estimation from each window and construct a graph.

We use a distance metric [43] similar to the Bhattacharyya distance and the symmetric Kullback-Leibler divergence to characterize the similarity between two distributions. The distance

metric is defined as follows:

$$d_{\mathcal{G}}\left(\mathcal{N}_1, \mathcal{N}_2\right) = \left(\mathbf{u}^\top \mathbf{S}^{-1} \mathbf{u}\right)^{\frac{1}{2}} + \left(\sum_{k=1}^{d} \ln^2 \lambda_k \left(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2\right)\right)^{\frac{1}{2}} \tag{4.10}$$

Here $\mathbf{u}$ represents the difference in mean between the two distributions and $\mathbf{S}$ is the averaged covariance matrix. λ_k is the solution of a generalized eigenvalue problem (GEP) [44] of $\boldsymbol{\Sigma_1}$ and $\boldsymbol{\Sigma_2}$. The advantage of this metric is that it is a full metric and satisfies the triangle inequality, which is essential when constructing the connection graph.

In the graph, each node denotes extrinsics from one window, and each edge represents similarity between the connecting nodes, with the similarity value beyond some threshold.

Then we further use the Pairwise Consistent Measurement (PCM) algorithm in [3] to find a clique containing most reliable and consistent estimations from all the windows. In the final optimization factor graph, which will be covered in the next section, we will only use windows that are in the clique.

4.3.3 Fusing Results

Extrinsics optimization results from each window, after going through the filtering processes described in Section 4.3.1 and Section 4.3.2, need to be fused to get a final estimation in the trajectory segment.

Utilizing the results corresponding to the clique nodes in Section 4.3.2, we construct a final fusion factor graph as shown in Fig. 4.6 (c). All clique nodes act as priors on the extrinsics variable. We also apply a robust loss function to reject possible outliers. We use Huber loss with threshold $\delta = 0.5$ as the robust loss function [45].

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta \\ \delta\left(|a| - \frac{1}{2}\delta\right), & \text{otherwise} \end{cases} \tag{4.11}$$

After optimization, we obtain extrinsics result along with its covariance matrix, which denotes the uncertainty of the estimation in each direction. This formulation has the benefit of compensating the degeneracy in the result from one window alone. One direction might be degenerate in the extrinsics from one window due to limited rotation or translation in that window, but the same direction might be well-constrained in another window. After being fused in this graph, information from all the windows is utilized to decrease uncertainty in all of the directions.

4.4 Extrinsics Update Strategy

4.4.1 Determining Degeneracy and Update Current Extrinsics

Due to degeneracy in the trajectory, it is still possible that the final result we get from optimization is degenerate in some dimensions. We check the covariance of the extrinsics estimation to verify the reliability of a specific dimension. If the marginal covariance of the extrinsics estimation is smaller than an empirical threshold in all directions, then we accept the estimation and update the original extrinsics.

Chapter 5

Experiments and Results

We conduct experiments in both simulation and real world experiments. In simulation experiments, we compare the extrinsics from our pipeline to the ground truth. In real-world experiments, since the ground truth extrinsics between two stereo pairs without FoV overlap is difficult to obtain, we examine the convergence and consistency of the results we get from our extrinsics estimation pipeline.

5.1 Simulation Experiments

5.1.1 Experiment Settings

A drone with a forward and backward stereo pair is used to gather image data from a photo-realistic environment simulated in Gazebo simulator, as shown in Fig. 5.1. We firstly evaluate our extrinsics calibration method in a single window as a baseline. Then we do experiments on trajectory segments with multiple windows.

Two identical visual odometry systems are run simultaneously on two stereo pairs. Our visual odometry system utilized to form the optimization factor graph is adapted from OpenVSLAM [46]. In order to make the two systems create keyframes at the same time, a third thread is created to monitor number of landmarks observed in each frame and percentage of landmarks that are different from landmarks in their corresponding previous keyframes. Keyframes are generated as many as possible. As long as minimum number of landmarks are observed and the current frame is different from the previous keyframe by a minimal visual change, we create keyframes simultaneously in the two visual odometry systems.

In implementing the factor graph optimization framework, we use the Georgia Tech smoothing and mapping (GTSAM) library [47]. The experiments are run on an Intel Core i7-7700 CPU @ 3.60GHz and 32GB RAM without GPU parallelization.

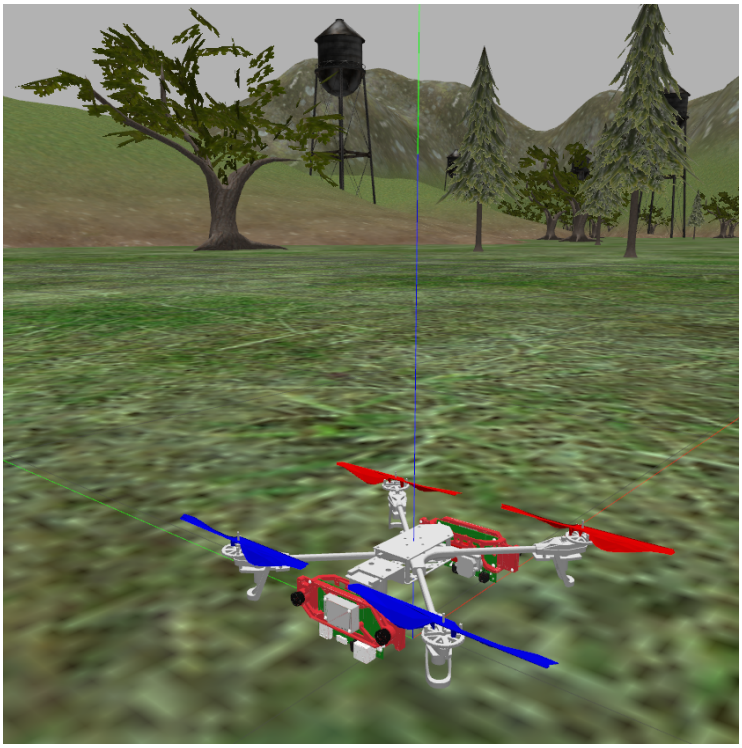

Figure 5.1: The simulation environment and configuration of the two-stereo system. One stereo pair is facing forward while the other is facing backward with no FoV overlap.

5.1.2 Factor Graph Based on Visual Odometry

To evaluate our visual-odometry-based method, we set the prior factor for the extrinsics to be within 3° for rotation and 3cm for translation, which is a larger error than would actually happen in most real-world scenarios. In order to form a complete factor graph, we also need to assign a noise model for all the factors in the factor graph. For the prior factor attached to the extrinsics variable, we use a diagonal covariance matrix with large values to have minor influence on the result while making sure that the system is well constrained even for situations where some variables are not observable. We also used noise model in the form of diagonal covariance matrix for the VO factors in the visual-odometry-based factor graph, with the values in the matrix generated by subtracting the ground truth transformation from the transformation estimated by the visual odometry algorithm. Note this cannot be achieved when used in real situation as we are utilizing the ground truth, which is not available in real situation. Extracting reliable covariances of the visual odometry transformation is difficult and here we are just using the representation mentioned above to demonstrate the best result it can achieve.

Trajectory with 2000 image frames and spanning around 100m was tested. The window selection result is shown in Fig. 5.2. Representative images are shown for selected and rejected windows. Around the scene in the left image, the window was rejected because the drone was flying close to a tree, which caused the window to have higher rates of erroneous measurements.

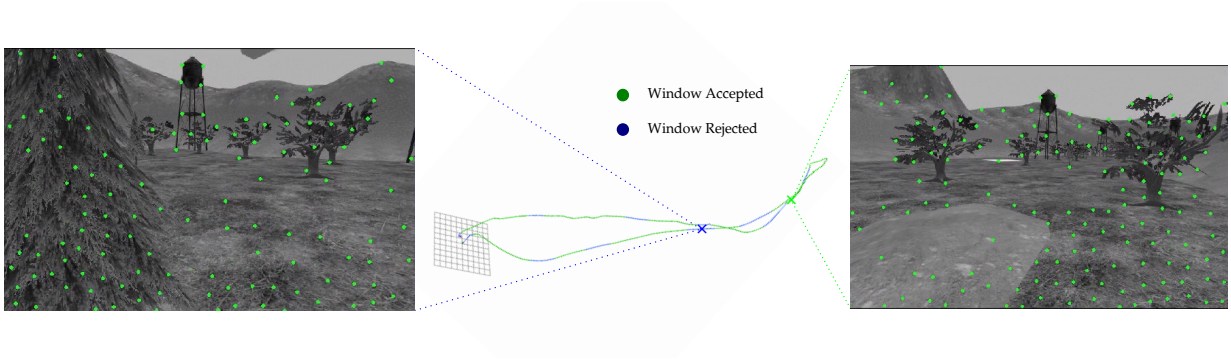

Figure 5.2: Window selection results from visual-odometry-based factor graph are shown on the trajectory in the simulated Gazebo environment. Representative images are shown for selected and rejected windows.

In Table 5.1, one sample result of the extrinsic estimation is shown comparing with ground truth. The result shows that with the rotation error within 3° and translation error within 3cm, our algorithm can converge with an error below 0.14° for rotation and 0.5cm for translation in well constrained dimensions. Notice that the accuracy of our extrinsic estimation algorithm depends on the shape of the trajectory. For example, from the result we can see that in the z axis we have larger error comparing with axis x and y. This is because the dimension of the z axis is not well constrained compared to x and y axis.

Table 5.1: Example result showing convergence and accuracy of our method of visual-odometry-based factor graph.

	Roll(°)	Pitch(°)	Yaw(°)	x(m)	y(m)	z(m)
Initial	0.00	0.00	-177.00	-0.210	0.000	0.000
Result	-0.00	0.03	-179.86	-0.197	-0.004	-0.005
GT	0.00	0.00	-180.00	-0.200	0.000	0.000

5.1.3 Factor Graph Based on Bundle Adjustment

When evaluating bundle-adjustment-based method, we set the prior factors the same as we did in visual-odometry-based method. For the visual odometry factors, we assign a noise model of 1 pixel standard deviation for the pixel coordinate measurements both from $stereo1$ and $stereo2$. For the VO between factors, we used noise model in the form of diagonal covariance matrix with large value, similar to the prior factor attached to the extrinsics variable.

We take 1000 random samples of the initial values of the extrinsics parameter we want to estimate, with the perturbation in each direction, $roll$, $pitch$, yaw, x, y, z, following a uniform distribution centered around the ground truth. The perturbation is within the range of 3° for rotation and 3cm for translation. We then plot the distribution of the results from our method in Fig. 5.3.

From the Fig. 5.3 we can see that the estimated extrinsics scatter around the ground truth, which means our method does converge to the ground truth, even though for one specific trial, it

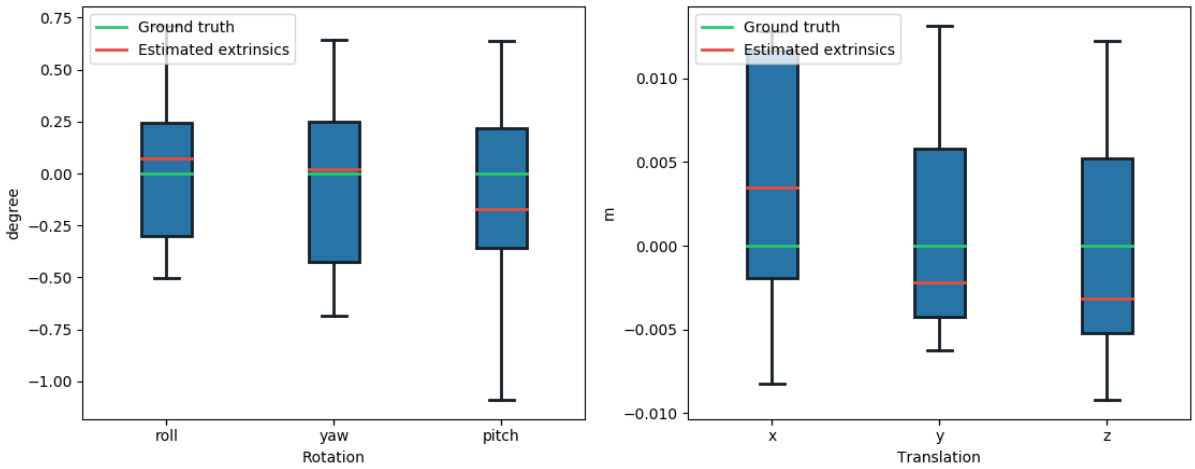

Figure 5.3: Distribution of estimated extrinsics vs. ground truth in simulated Gazebo environment.

may have a relatively large error. This could be caused by local optimum reached in the factor graph optimization when the perturbation is far away from the ground truth. From the figure we can see our method has an accuracy around 0.3° for rotation and 0.5cm for translation.

Comparing bundle-adjustment-based method to visual-odometry-based method, we can see that they have similar accuracy for both rotation and translation. Considering we are using ground truth in visual-odometry-based method, in practice we prefer to use bundle-adjustment-based method while maintaining similar level of accuracy.

5.2 Real World Experiments

5.2.1 Experiment Settings

Our real-world data was collected using a two stereo camera rig with time synchronized images, as shown in Fig. 5.4. The two stereo pairs do not have FoV overlap. The cameras operated at approximately 25 Hz and time synchronization was achieved using an FPGA. Each individual camera also had an on-board IMU which operated at 200 Hz. In the experiment, we did manual calibration with respect to one of the IMUs for both of the two stereo pairs and then got the ground truth * by canceling out the IMU coordinate frame.

The multi-stereo rig was moved around the highbay in the Field Robotics Center and moved back precisely to its original position. To include landmark observability, the data was intentionally made to several points of sudden occlusion occurring during the run. Sample images from the two stereo pairs are shown in Fig. 5.5.

*Not actually ground truth because there could still be some error, but at least can provide us with some reference.

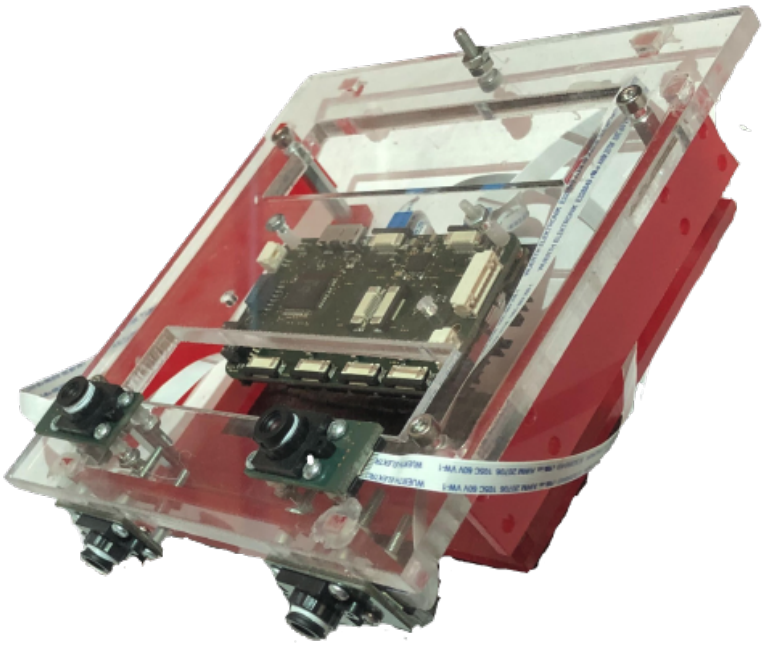

Figure 5.4: Two-stereo camera rig with time-synchronized cameras used to collect real-world data.

5.2.2 Results

To evaluate the convergence and consistency of our extrinsics calibration method, we initialize extrinsics with different values. Method effectiveness is validated as long as the result converges every time and the results remain consistent across all the different initial values.

Similar to the experiments done in Section 5.1. We use the bundle-adjustment-based factor graph and take 1000 random samples of the initial values of the extrinsics parameter we want to estimate, with the perturbation in each direction, $roll$, $pitch$, yaw, x, y, z, following a uniform distribution centered around the ground truth. The perturbation is within the range of 3° for rotation and 3cm for translation. The distribution of estimated extrinsics vs. ground truth * with real-world dataset is shown in Fig. 5.6.

From the Fig. 5.6 we can see that in real-world situation our method can still converge consistently to an estimation within a specific range. Even though the estimation varies for each trial, the range is much smaller than our perturbation. From the boxplot we can see that our method can achieve an accuracy of 0.5° for rotation and 0.6cm for translation. This could be the result of the insufficient features in one of the stereo pairs, which caused some degeneracy in the factor graph. In the future we plan to evaluate on more datasets with more diverse environment and camera

*Not actually ground truth because there could still be some error, but can provide us with some reference.

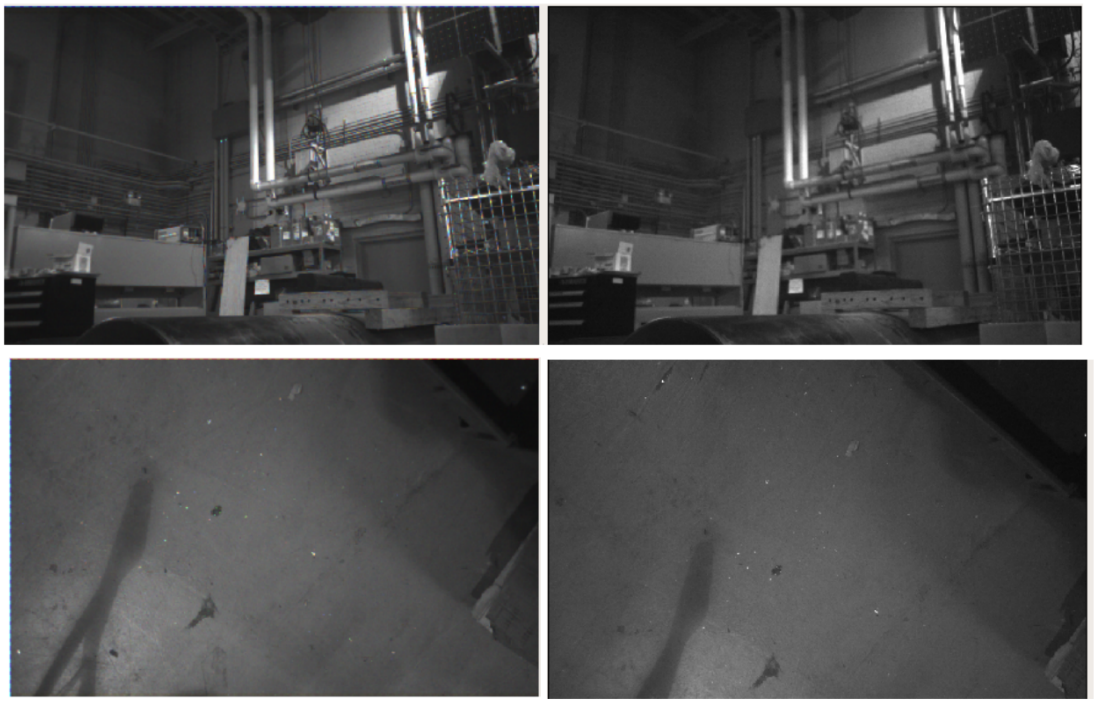

Figure 5.5: Sample images from the two stereo pairs taken in Field Robotics Center highbay. (up) images are from forward-facing stereo and $(down)$ images are from downward-facing stereo.

orientations.

In the experiment, the average time used to optimize the factor graph for a window containing 20 poses and 50 selected landmarks is 1.2 seconds. When multiple threads are used, for a trajectory with 100 keyframes, we can get an extrinsics estimation using our method within 4.5 seconds. This makes our method an on-the-fly extrinsics calibration method that can help to revise the extrinsic parameters when the system is running.

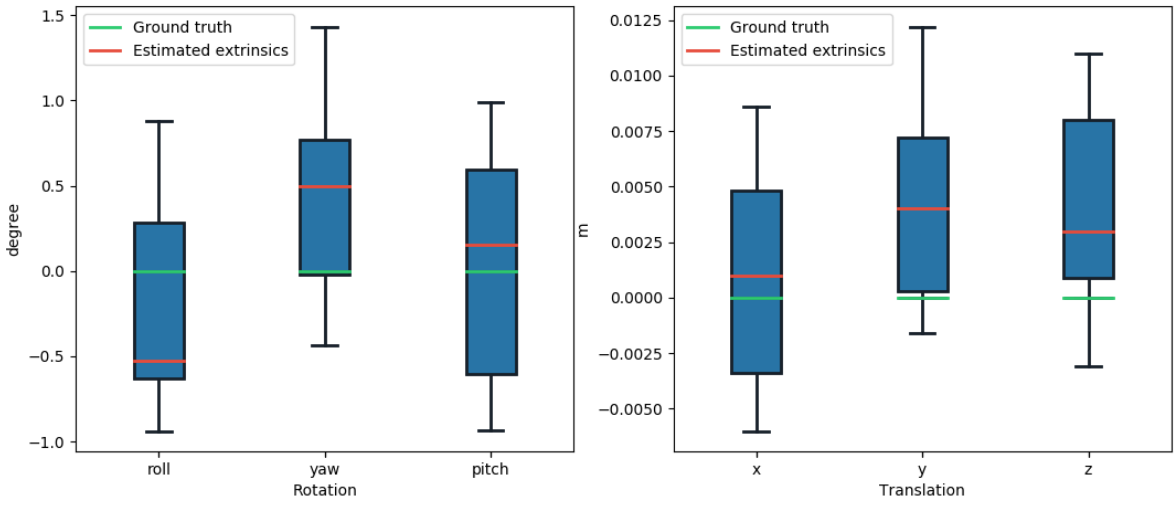

Figure 5.6: Distribution of estimated extrinsics vs. ground truth* with real-world dataset.

Chapter 6

Discussion

In this work, an online factor-graph-based method to calibrate the extrinsics of multi-stereo systems with no FoV overlap is proposed. Our approach does not require a specific calibration target and can operate in degenerate scenarios due to certain types of sensor motion or feature/landmark distributions.

We demonstrate successful online calibration of a two-stereo system with simulated Gazebo environment data as well as real-world data. In simulation experiments, results from our extrinsics calibration is compared against ground truth. In real-world experiments, we examine the convergence and consistency of our results. Both simulation and real-world experiments prove the feasibility of our method.

When choosing the components constrained in the factor graph, our method is divided into visual-odometry-based method and bundle-adjustment-based method. Even though more concise, visual-odometry-based factor graph requires incorporating correct covariance matrix representing the uncertainty of the transformation between two poses, which can be hard to get in practice. So for real-world data, we use bundle-adjustment-based factor graph.

Many parameters can have effect on the result and accuracy of our method, for example, the number of landmarks constrained, the different size of each window, and also the shape of the trajectory. Theoretically, the more landmarks constrained, the more accurate the result will be, but more landmarks could also increase the percentage of unstable or incorrect features that would pollute our result. For the size of the window, the larger the window size is, the more accurate the result will be. However, since the total number of keyframes in the entire trajectory is fixed, which means that the number of windows would decrease, this would further affect the result in the set maximization based window selection step. So there is a trade-off on the choice of the window size. For the shape of the trajectory, there could be degeneracy in the entire trajectory, which means it is impossible to recover full extrinsics from this trajectory. In most cases, the degeneracy should correspond to large values in the covariance matrix due to large uncertainty. But whether we can update the extrinsics for directions with small values in the covariance matrix in this degenerated case, or in what case can we safely update the extrinsics value when given the covariance matrix is still an open problem.

Chapter 7

Conclusion

We proposed an on-the-fly extrinsics calibration method for stereo pairs lacking overlapping field of view that is robust to visual odometry errors. Experimental results with both simulation and real-world data show that the proposed method is successful in estimating and updating the extrinsics on the fly.

Possible future work includes:

1. For visual odometry based factor graph, propose method to obtain transformation covariance accurately.

2. Include more sensor modalities, such as monocular, fisheye camera, IMU. There are many other sensor rigs that may contain other sensor modalities and it is possible to apply a similar calibration method to those sensor rigs. Nowadays sensors such as IMU are quite common on consumer electronics, incorporating measurements from these sensors to our optimization factor graph could be helpful to increase the accuracy.

3. Understand more about the covariance related to extrinsics and propose a more safe extrinsics update strategy. Errors resulting from outlier measurements that are omitted in our window grouping and window selection process could lead to local optimality in the factor graph optimization. Even though the values in the extrinsics covariance matrix might be small, it does not necessarily mean the estimation result is acceptable.

4. Combine trajectory planning and extrinsics calibration pipeline to better constrain the factor graph and reduce the effect of degenerate sensor motion. When we have control over the trajectory shape, we can design a trajectory that is more possible to constrain our extrinsics variable well if needed.

Bibliography

[1] Lionel Heng, Gim Hee Lee, and Marc Pollefeys. Self-calibration and visual SLAM with a multi-camera system on a micro aerial vehicle. *Autonomous Robots*, pages 259–277, 2015.

[2] Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1-2):1–139, 2017.

[3] Joshua G Mangelson, Derrick Dominic, Ryan M Eustice, and Ram Vasudevan. Pairwise consistent measurement set maximization for robust multi-robot map merging. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2916–2923. IEEE, 2018.

[4] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[5] Roger Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. 3(4):323–344, 1987.

[6] Yonggen Ling and Shaojie Shen. High-precision online markerless stereo extrinsic calibration. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1771–1778. IEEE, 2016.

[7] Michael Warren, David McKinnon, and Ben Upcroft. Online calibration of stereo rigs for long-term autonomy. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3692–3698. IEEE, 2013.

[8] Peter Hansen, Hatem Alismail, Peter Rander, and Brett Browning. Online continuous stereo extrinsic parameter estimation. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1059–1066. IEEE, 2012.

[9] Thao Dang, Christian Hoffmann, and Christoph Stiller. Continuous stereo self-calibration by camera parameter tracking. *IEEE Trans. on Image Processing*, 18(7):1536–1550, 2009.

[10] Michael Warren and Ben Upcroft. High altitude stereo visual odometry. *Proceedings of Robotics: Science and Systems IX*, 2013.

[11] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. The MIT Press, Cambridge, MA, 1993.

[12] Stephen J Maybank and Olivier D Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[13] Q-T Luong and Olivier D Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, 22(3):261–289, 1997.

[14] Richard I Hartley. An algorithm for self calibration from several views. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, volume 94, pages 908–912. Citeseer, 1994.

[15] Olivier D Faugeras, Q-T Luong, and Stephen J Maybank. Camera self-calibration: Theory and experiments. In *Eur. Conf. on Computer Vision (ECCV)*, pages 321–334. Springer, 1992.

[16] Simon JD Prince. *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

[17] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[18] Ziran Xing, Jingyi Yu, and Yi Ma. A new calibration technique for multi-camera systems of limited overlapping field-of-views. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5892–5899. IEEE, 2017.

[19] Jesse Levinson and Sebastian Thrun. Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems*, volume 2, page 7, 2013.

[20] Hsiang-Jen Chien, R. Klette, N. Schneider, and U. Franke. Visual odometry driven online calibration for monocular lidar-camera systems. In *International Conference on Pattern Recognition (ICPR)*, pages 2848–2853, 2016.

[21] Ryoichi Ishikawa, Takeshi Oishi, and Katsushi Ikeuchi. Lidar and camera calibration using motions estimated by sensor fusion odometry. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 7342–7349. IEEE, 2018.

[22] Terry Scott, Akshay A Morye, Pedro Piniés, Lina M Paz, Ingmar Posner, and Paul Newman. Choosing a time and place for calibration of lidar-camera systems. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4349–4356. IEEE, 2016.

[23] Jinyong Jeong, Younghun Cho, and Ayoung Kim. The road is enough! extrinsic calibration of non-overlapping stereo camera and lidar using road information. *IEEE Robotics and Automation Letters (RA-L)*, 4(3):2831–2838, 2019.

[24] Shuai Dong, Xinxing Shao, Xin Kang, Fujun Yang, and Xiaoyuan He. Extrinsic calibration of a non-overlapping camera network based on close-range photogrammetry. *Applied optics*, 55(23):6363–6370, 2016.

[25] Ram Krishan Kumar, Adrian Ilie, Jan-Michael Frahm, and Marc Pollefeys. Simple calibration of non-overlapping cameras with a mirror. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2008.

[26] Pierre Lébraly, Clément Deymier, Omar Ait-Aider, Eric Royer, and Michel Dhome. Flexible extrinsic calibration of non-overlapping cameras using a planar mirror: Application to vision-based robotics. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5640–5647. IEEE, 2010.

[27] Zachary Taylor and Juan Nieto. Motion-based calibration of multimodal sensor extrinsics and timing offset estimation. *IEEE Transactions on Robotics*, 32(5):1215–1229, 2016.

[28] Pierre Lébraly, Eric Royer, Omar Ait-Aider, Clément Deymier, and Michel Dhome. Fast calibration of embedded non-overlapping cameras. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 221–227. IEEE, 2011.

[29] Kevin Eckenhoff, Patrick Geneva, Jesse Bloecker, and Guoquan Huang. Multi-camera visual-inertial navigation with online intrinsic and extrinsic calibration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3158–3164. IEEE, 2019.

[30] Thomas Schneider, Mingyang Li, Cesar Cadena, Juan Nieto, and Roland Siegwart. Observability-aware self-calibration of visual and inertial sensors for ego-motion estimation. *IEEE Sensors Journal*, 19(10):3846–3860, 2019.

[31] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry. *IEEE Robotics & Automation Magazine*, 18(4):80–92, 2011.

[32] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, volume 1, pages I–I. Ieee, 2004.

[33] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[34] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Intl. Conf. on Computer Vision (ICCV)*, December 2013.

[35] Niko Sünderhauf and Peter Protzel. Towards a robust back-end for pose graph SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1254–1261. IEEE, 2012.

[36] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *Intl. J. of Robotics Research*, 32(7):826–840, 2013.

[37] Pratik Agarwal, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. Robust map optimization using dynamic covariance scaling. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 62–69. Ieee, 2013.

[38] Qinghua Wu and Jin-Kao Hao. A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242(3):693–709, 2015.

[39] Bharath Pattabiraman, Md Mostofa Ali Patwary, Assefaw H Gebremedhin, Wei-keng Liao, and Alok Choudhary. Fast algorithms for the maximum clique problem on massive graphs with applications to overlapping community detection. *Internet Mathematics*, 11(4-5):421–448, 2015.

[40] Joshua Jaekel, Michael Kaess, and Paloma Sodhi. Robust multi-stereo visual inertial odometry. 2019.

[41] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–375. Springer, 1999.

[42] Michael J Tribou, David WL Wang, and Steven L Waslander. Degenerate motions in multi-camera cluster SLAM with non-overlapping fields of view. *Image and Vision Computing*, 50:27–41, 2016.

[43] Karim T Abou-Moustafa, Fernando De La Torre, and Frank P Ferrie. Designing a metric for the difference between Gaussian densities. In *Brain, Body and Machine*, pages 57–70. Springer, 2010.

[44] Benyamin Ghojogh, Fakhri Karray, and Mark Crowley. Eigenvalue and generalized eigenvalue problems: Tutorial. *CoRR*, 2019.

[45] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[46] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. OpenVSLAM: a versatile visual SLAM framework. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2292–2295, 2019.

[47] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.