POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Multi-Sensor Extrinsic Parameters Calibration based on Sensors' Ego-Motion

Tesi di Laurea Magistrale in
Automation and Control Engineering - Ingegneria dell'Automazione

Author: **Giacomo Giorgi**

Student ID: 922186
Advisor: Prof. Matteo Matteucci
Co-advisors: Dott. Simone Mentasti
Academic Year: 2020-2021

Abstract

The fusion of sensors must guarantee accuracy and precision, fundamental characteristics for a system equipped with multiple and heterogeneous measuring devices. To obtain them, the sensors must be calibrated, i.e. their poses must be expressed through relative 6D transformations in a common reference system. This manuscript describes a calibration method, suitable for mobile systems, that estimates the extrinsic 6 DoFs of multiple sensors with respect to a reference one. This procedure is divided into two steps: the first one exploits the incremental motions of the sensors to estimate their parameters x, y and *yaw*, while the second one requires them to observe the ground for a limited period of time, in order for it to compute the remaining z, *roll* and *pitch*. This proposed approach is characterized by the estimation of calibration parameters that are initialized in closed form. Moreover, the scaling ambiguity, which results from the estimation of a motion by a monocular camera, is handled explicitly in order to allow the combination of these sensors with different others, such as Lidar and stereo cameras. Finally, an experimental calibration is performed through the exploitation of a collection of outdoor measured datasets and the results confirm the accuracy of this method, especially in the estimation of planar parameters.

Keywords: sensors calibration, extrinsic parameters, mobile system, incremental motions, ground observations.

Abstract in lingua italiana

La fusione di sensori deve garantire accuratezza e precisione, caratteristiche fondamentali per un sistema dotato molteplici dispositivi di misurazione eterogenei. Per ottenerle i sensori devono essere calibrati, cioè le loro pose devono essere espresse attraverso trasformazioni relative 6D in un sistema di riferimento comune . Questo manoscritto descrive un metodo di calibrazione adatto a sistemi mobili, che stima i parametri estrinsechi 6 DoFs di più sensori rispetto a uno di riferimento. Questa procedura è divisa in due parti: la prima sfrutta i moti incrementali dei sensori per stimarne i parametri x, y e yaw, mentre la seconda richiede le loro osservazioni del suolo per un periodo limitato di tempo, affinchè possa calcolare i rimanenti z, *roll* e *pitch*. Questo approccio proposto è caratterizzato dalla stima dei parametri di calibrazione che vengono inizializzati in forma chiusa. Inoltre, l'ambiguità di scala, che deriva dalla stima di un movimento da parte di una telecamera monoculare, è gestita esplicitamente al fine di consentire la combinazione di questi sensori con gli altri, come Lidar e telecamere stereo. Infine, viene eseguita una calibrazione sperimentale sfruttando una collezione di set di dati misurati all'aperto e i risultati confermano l'accuratezza di questo metodo, specialmente nella stima dei parametri planari.

Parole chiave: calibrazione dei sensori, parametri estrinseci, sistema mobile, movimenti incrementali, osservazioni del suolo.

Contents

1 | Introduction

1.1. Overview

Environment perception is a fundamental task required by any automatic mobile system for mapping and localization of its surrounding. For a vehicle or mobile robot equipped with multiple heterogeneous sensors, the perception capability is directly dependent on the technology for their fusion that allows combining different kinds of measurements. For this reason, sensor calibration is a fundamental step since it enables a variety of different measures, taken from devices of different nature, to be combined for an excellent perception of the static and dynamic environment.

As well defined by Peršić [20], the calibration of a sensor is a process that aims to compute its intrinsic and/or extrinsic parameters.
The firsts are the internal ones that are related to the specific working principle for each sensor, like the focal length for the camera or the bias for Lidar range measurements. On the other side, the extrinsic parameters represent the 6 DoF transformations between the pose of the sensors inside the system. My work will focus on the automatic calibration of these latter parameters, i.e. the computation of the position and rotation of a set of multiple sensors inside a vehicle. The extrinsic parameters of each sensor can be modified during the life of the vehicle by possible vibrations, bumps, or other external events, thus their calibration is fundamental to avoid drifts and mistakes in the measured data.

There are different ways to compute the intrinsic or extrinsic parameters calibrations and they are classified into three main categories :

- Target-based calibration;
- Targetless calibration;
- Motion-based calibration.

The first one includes the calibration techniques that exploit human-designed targets to estimate the parameters. Each target is designed for a specific sensor or group of them to be the most effective, so each calibration is particular and different from the others

depending on the analyzed device. For this reason, the target-based calibrations are usually more accurate and precise than the targetless ones and able to find better results. However, there are limitations, such as the necessity of human intervention to design the specific target or the offline execution of these processes, that restricts their usability.

On the other hand, the targetless calibration techniques work on the environment features to match correspondences in the data of the sensor. They were developed to compensate for the drawback of the previous category since the exploitation of the surrounding characteristics improves the usability of the procedure and makes it more automatic.
The sensors must guarantee enough information to extract the structure of the environment in order to find and recognize the specific feature on which the calibration is based. Therefore, these calibrations are mainly developed for cameras and lidars, which is the main limitation of this category.

The last class includes the motion-based calibration systems, which, as the name suggests, use the sensors' ego motions to estimate the parameters. These approaches could be considered targetless since they do not need any targets, but, on the other side, they are not limited by specific features in the environment as the previous category. Moreover, these techniques can calibrate a wider range of sensors since their only prerequisite is the capability of the single sensor to estimate its own motion. Therefore, motion-based calibration is the only viable option for sensors like IMU and odometry encoders since they are proprioceptive and can not measure the external state of the environment.
Generally, these calibrations are usually less accurate and precise than the target-based ones since they can only exploit the sensors' measurements of their incremental motions and not specifically designed targets.

1.2. Approach

In this manuscript, we describe a motion-based calibration approach that estimates the 6 DoFs extrinsic parameters for a mobile system equipped with multiple sensors. Their incremental motions are exploited to calibrate the planar extrinsic parameters and, additionally, this process uses the sensors' observations of the ground plane to compute the remaining three ones. All of this work is mainly influenced and inspired by Zuñiga-Noël et al. though [29].

The manuscript of this thesis is organized into three chapters, where one is purely theoretical, two refer to an experimental test and the last one concludes the work.
More specifically, Chapter 3 revolves around the mathematical theory to formulate this calibration problem. Here, we introduce the steps of the approach, their least-squares

formulations and the close form solutions. Moreover, we add a description of the observability analysis of the planar extrinsic parameters and its demonstration.
Chapter 4 describes our performed experimental test where we calibrate four sensors exploiting a collection of datasets. It displays the pipeline that starts from the sensors' measurements of the dataset and leads to the estimation of the parameters, reporting the exploited programs and files.
Chapter 5 shows the results of the experimental calibration through tables and plots. Here, we comment and explain the numerical values that we computed, comparing them with each other.
Last but not least, the conclusion exposes a final summary of the work presented in this manuscript and of the obtained results. Moreover, it introduces possible future developments on this subject.

2 | State of Art

Due to its increasing importance in the automation of mobile systems, several papers can be found in the literature concerning sensor calibration.
Most of them are focused on target-based calibration processes, where several targets were identified and developed for different groups of sensors. For example, An et al. [2] used a planar board with a chessboard pattern to calibrate a monocular camera and a Lidar, exploiting point correspondence between 3D-2D and 3D-3D.
Other than that, Velas et al. [27] designed a planar marker with four circular holes for an RGB camera and a Velodyne Lidar.
For the same sensors (a camera and a 3D Lidar), Mishra et al. [17] exploited Planar Surface Point to Plane and Planar Edge Point, using a markerless planar board as a target to estimate the extrinsic parameters.
Differently, Peršić et al. [21] calibrated the extrinsic parameters for a set of Velodyne Lidar, radars and cameras, constructing a triangular trihedral corner retroreflector.

In a different category, there are targetless calibration techniques since they are based on environment features. For example, Jeong et al. [15] calibrated a stereo camera and a Lidar using features like road marks that are constantly present in an urban environment. The same environment is also full of building walls, which can compose, with the floor, an orthogonal trihedron. Therefore this geometrical structure became the basis on which Gong, Lin and Liu [8] developed their calibration process.
Similarly, Gomez-Ojeda et al. [7] used the same urban environment structure to find the extrinsic parameters of a 2D laser rangefinder and a camera. In this case, however, they used the line-to-plane and point-to-plane constraints.
Other usable features are linear segments, extracted in the environment by intersections between planes and boundary lines. They were exploited by Moghadam et al. [18] to develop a calibration technique for a camera-Lidar couple.

All the previous techniques required an a-priori knowledge of the surrounding, both for the designed target or the specific environment feature. To avoid this, the sensors' ego motions can be exploited to calibrate the extrinsic parameters.

In the last decade, many motion-based calibration techniques were developed starting with Brookshire and Teller [4] who are the first ones to use this kind of sensors' measurements. They estimated the 2D extrinsic parameters by solving the calibration with iterative minimization of a least-squares cost function and, additionally, they developed an observability analysis of this problem.
Subsequently, they extend their research [5] improving the calibration since they estimated the 6 DoF parameters. For this, they exploited the Dual Quaternions to represent the transformation between the sensors. This extension works only for a couple of sensors that must travel together along a non-degenerate path, but it does not require synchronized sensors, overlapping fields of view or common features observation.
Censi et al. [6] presented a calibration for an odometry sensor plus an exteroceptive second one like a Lidar. The idea was to estimate the intrinsic parameters of the odometer (radii and distance between the wheels) plus the extrinsic ones of the Lidar. They solved, with a close form solution, the simultaneous calibration formulated as a maximum likelihood problem.
Differently, the Unscented Kalman Filter was firstly used by Schneider et al. [23] to develop an online extrinsic calibration, which is robust against the degeneracy of the vehicle motion. This process can compute the 3 DoFs parameters or the 6 DoFs ones depending on the DoFs of the sensors' motions measurements.
An important step in sensors calibration was given by Guo et al.[9] and by Zienkiewicz et al.[28], since they dealt with the ambiguity problem of the scale in monocular cameras. The first ones developed a least-squares solution in closed form for automatic motion-based calibration of an odometer and a camera.
On the other hand, Zienkiewicz et al. proposed a solution based on the minimization of the photometric errors. These letters are induced by the homography which results from the planar motions by observing the ground plane.
Differently from the previous works, Heng et al.[11] calibrated a set of multiple sensors, i.e. cameras in their case, plus an odometer. Their strategy works only for imaging sensors but still, they can calibrate both the parameters types of the cameras with respect to the odometer, whose extrinsic ones are computed a-priori.
On the complete opposite side, there are Huang and Stachniss [13] since their calibration is for multiple sensors except for only the cameras. It is so because all the motion measurements must be expressed on the same scale since the formulation is based on the Gauss-Helmert model.
Finally an automatic multi sensor calibration of extrinsic parameters was developed by David Zuñiga-Noël, Jose-Raul Ruiz-Sarmiento, Ruben Gomez-Ojeda and Javier Gonzalez-Jimenez [29]. Their approach estimates the 6 DoFs of a set of multiple and heterogeneous

sensors' poses with respect to a reference one. They exploit the ego-motions for the planar parameters (x, y, yaw) and short observation of the ground plane for the remaining three ones (z, pitch and roll). Moreover, during the motion-based calibration, this technique can also estimate the scale ambiguity of monocular cameras, allowing to estimate their extrinsic parameters too.

3 | 6DoF Multi Sensor Calibration

This chapter revolves around the mathematical formulations at the basis of a calibration that estimates the 6 DoF extrinsic parameters for a set of multiple heterogeneous sensors in a mobile system. This estimation process is mainly divided into two steps: the motion-based one and the ground observations-based one, where each of them focuses on a triplex of parameters.

This theory is inspired by the amazing work of David Zuñiga-Noël, Jose-Raul Ruiz-Sarmiento, Ruben Gomez-Ojeda and Javier Gonzalez-Jimenez described in *"Automatic Multi-Sensor Extrinsic Calibration For Mobile Robots"* [29] and by Andrea Censi, Antonio Franchi, Luca Marchionni and Giuseppe Oriolo in *"Simultaneous Calibration of Odometry and Sensor Parameters for Mobile Robots"* [6].

3.1. Motion-based Calibration

In this section, we focus on the calibration between two coplanar sensors, i.e. the estimation of the 2D transformation between the pose of a first one compared to the reference one. This calibration step is based on the sensors' ego-motions, thus it can be used only on devices that can measure their trajectory. These letters must be composed at least by 2D incremental poses that need to be synchronized.
Normally, measurements from heterogeneous sensors can be asynchronous, so they can have different sampling rates. To solve it, we can set the time of the reference sensor as the baseline and exploit linear interpolation to resample the planar incremental poses of the second sensor.
Moreover, the different natures of the devices can lead to different scales in which the translation components are expressed, i.e. x and y. This problem is faced in this calibration step since not only the planar extrinsic parameters are estimated, but also the scaling one.

3.1.1. Problem Formulation

The goal of this calibration step is the estimation of the parameters of a 2D similarity transformation between an i-th sensor and a j-th one, which are respectively the reference sensor and the one whose pose need to be computed. The similarity transformation represents a rigid transformation followed by a scaling one.
Let's express the parameters of this transformation as ${}^{i}\boldsymbol{t}_{j} = \begin{pmatrix} t_x & t_y & t_\theta & t_s \end{pmatrix} \in \mathrm{Sim}(2)$ where:

- $t_x, t_y \in \mathbb{R}$ indicate the translation components;
- $t_\theta \in \mathbb{R}$ the yaw rotational component;
- $t_s \in \mathbb{R}^+$ is the scaling factor;
- Sim(2) represents the group of orientation that preserves the similarity transformations.

The transformation parameters can be used to express the measurements taken from the j-th sensors ${}^{j}\boldsymbol{m}$ into the i-th reference frame applying this formula:

$$ {}^{i}\boldsymbol{m} = t_s \ (R(t_\theta) \ {}^{j}\boldsymbol{m} \ + \ \begin{bmatrix} t_x \\ t_y \end{bmatrix}) \tag{3.1} $$

where $R(t_\theta)$ is the 2D rotation matrix:

$$ R(t_\theta) \triangleq \begin{bmatrix} \cos(t_\theta) & -\sin(t_\theta) \\ \sin(t_\theta) & \cos(t_\theta) \end{bmatrix} \tag{3.2} $$

Before going deeper in the motion-based estimation of ${}^{i}\boldsymbol{t}_{j}$, let's define the group operator of Sim(2) and its inverse (respectively $\oplus$ *and* $\ominus$), that are exploited in this chapter.

$\oplus : \mathrm{Sim}(2) \times \mathrm{Sim}(2) \to \mathrm{Sim}(2)$

$$ \boldsymbol{a} \ \oplus \ \boldsymbol{b} = \begin{bmatrix} b_s^{-1} a_x + b_x \cos(a_\theta) - b_y \sin(a_\theta) \\ b_s^{-1} a_y + b_x \sin(a_\theta) + b_y \cos(a_\theta) \\ a_\theta + b_\theta \\ a_s b_s \end{bmatrix} $$

$\ominus : \mathrm{Sim}(2) \to \mathrm{Sim}(2)$

$$\ominus \boldsymbol{a} = \begin{bmatrix} -a_s a_x \cos(a_\theta) - a_s a_y \sin(a_\theta) \\ a_s a_x \sin(a_\theta) - a_s a_y \cos(a_\theta) \\ -a_\theta \\ a_s^{-1} \end{bmatrix}$$

Moreover, let's call $\boldsymbol{q} \in \mathrm{SE}(2)$ the pose of a sensor and $\boldsymbol{p} \in \mathrm{SE}(2)$ the incremental motion between two of them. For example, $\boldsymbol{p}_i^k$ represents the incremental rigid body motion of the reference sensor between the poses at time k and k+1, as we can see in Figure 3.1. It should be noted that SE(2) transformation is a particular case of Sim(2), i.e. the case where the scaling factor is set to the identity value. Therefore the operators previously defined for Sim(2) can be applied also to the SE(2) transformations.

For the motion-based step of the calibration process, the inputs are the sensors' incremental motions $\boldsymbol{p}_i^k$ and $\boldsymbol{p}_j^k$. Each of them can be derived from the poses of the specific sensor at time k and k+1 (respectively $\boldsymbol{q}^k, \boldsymbol{q}^{k+1} \in \mathrm{SE}(2)$), hence, for example, we can compute the i-th sensor incremental motions as:

$$\boldsymbol{p}_i^k \triangleq \ominus \boldsymbol{q}_i^k \oplus \boldsymbol{q}_i^{k+1} \tag{3.3}$$

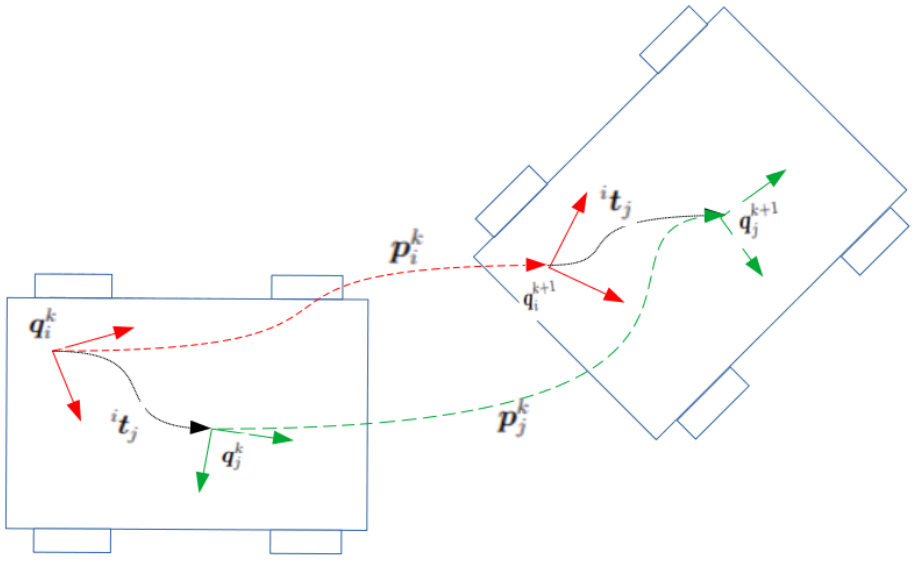


Figure 3.1: Representation of the i-th and j-th coplanar sensors on a vehicle. The incremental motions $\boldsymbol{p}_i^k$ and $\boldsymbol{p}_j^k$, between the poses at time step k and k+1 for each sensor, are related by the fixed similarity transformation expressed by ${}^i\boldsymbol{t}_j$.

The poses of the j-th sensor $\boldsymbol{q}_j^k$ can be expressed using the one of the i-th sensor plus the parameters of the transformation between them, though this equation:

$$\boldsymbol{q}_j^k = \boldsymbol{q}_i^k \oplus {}^i\boldsymbol{t}_j \tag{3.4}$$

Applying the (3.3) for the j-th sensor instead of the i-th one and inserting (3.4) in it, the incremental motions for the j-th sensors can be computed as:

$$\boldsymbol{p}_j^k = \ominus (\boldsymbol{q}_i^k \oplus {}^i\boldsymbol{t}_j) \oplus (\boldsymbol{q}_i^{k+1} \oplus {}^i\boldsymbol{t}_j) \tag{3.5}$$

Starting from the (3.5), it is possible to rearrange its terms in order to apply (3.3) and find:

$$\boldsymbol{p}_j^k = \ominus {}^i\boldsymbol{t}_j \oplus \boldsymbol{p}_i^k \oplus {}^i\boldsymbol{t}_j \tag{3.6}$$

This formula put in relation the relative incremental motions $\boldsymbol{p}_i^k, \boldsymbol{p}_j^k \in \mathrm{SE}(2)$ of i-th and j-th sensors at time step k with the extrinsic calibration parameters ${}^i\boldsymbol{t}_j \in \mathrm{Sim}(2)$.

The Equation (3.6) is exploited to derive an error function that represents the difference between the j-th incremental motions and the i-th incremental motions, where the letter is transformed with the calibration parameters :

$$\boldsymbol{\varepsilon}_{ij}^k(\boldsymbol{t}) \triangleq \boldsymbol{p}_j^k - \ominus {}^i\boldsymbol{t}_j \oplus \boldsymbol{p}_i^k \oplus {}^i\boldsymbol{t}_j \tag{3.7}$$

This error function can be finally used to define the cost function $\mathcal{C}_{ij}$, that it is subsequently exploited to express a least-squares formulation of the two-sensor extrinsic calibration problem:

$$\boldsymbol{\zeta}_{ij}(\boldsymbol{t}) \triangleq \frac{1}{2} \sum_k \left\| \boldsymbol{\varepsilon}_{ij}^k(\boldsymbol{t}) \right\|_2^2 \tag{3.8}$$

$${}^i\boldsymbol{t}_j^* = \underset{\boldsymbol{t}}{\operatorname{argmin}} \ \boldsymbol{\zeta}_{ij}(\boldsymbol{t}) \tag{3.9}$$

In conclusion, the idea is to solve the least-squares problem in (3.9) by finding the extrinsic parameters ${}^i\boldsymbol{t}_j^*$ that reduces to the minimum the cost function $\mathcal{C}_{ij}(\boldsymbol{t})$ defined by (3.8)

3.1.2. Close Form Solution

The problem in (3.9) can be solved following an approach similar to the one used in [6], i.e. reducing the least-squares formulation to a quadratic system with a quadratic constraint. This derived constrained optimization problem is then uniquely solved in close form with the method of Lagrange multipliers.
The first step to reduce the calibration problem to a quadratic system is to rearrange the terms of the error function in (3.7) in this way:

$$\boldsymbol{\varepsilon}_{ij}^{k}(\boldsymbol{t}) = {}^{i}\boldsymbol{t}_j \oplus \boldsymbol{p}_j^k - \boldsymbol{p}_i^k \oplus {}^{i}\boldsymbol{t}_j \tag{3.10}$$

Then the calibration parameter $t_\theta \in {}^{i}\boldsymbol{t}_j$ must be parameterized into $\cos(t_\theta)$ and $\sin(t_\theta)$, that are considered as two independent variables. At this point all unknown parameters, that compose ${}^{i}\boldsymbol{t}_j$, can be grouped into the vector $\boldsymbol{\varphi} \in \mathbb{R}^5$:

$$\boldsymbol{\varphi} = \begin{bmatrix} t_s^{-1} & t_x & t_y & \cos(t_\theta) & \sin(t_\theta) \end{bmatrix}^{\mathrm{T}} \tag{3.11}$$

Now, with (3.11), the error term in (3.10) can be rewritten in matrix form as:

$$\boldsymbol{\varepsilon}_{ij}^{k} = \boldsymbol{\mathcal{Q}}_k \, \boldsymbol{\varphi} \tag{3.12}$$

where:

$$\boldsymbol{\mathcal{Q}}_k \triangleq \begin{bmatrix} -p_{i\ x}^k & 1-\cos(p_{i\ \theta}^k) & \sin(p_{i\ \theta}^k) & p_{j\ x}^k & -p_{j\ y}^k \\ -p_{i\ y}^k & -\sin(p_{i\ \theta}^k) & 1-\cos(p_{i\ \theta}^k) & p_{j\ y}^k & p_{j\ x}^k \end{bmatrix} \tag{3.13}$$

is a matrix composed of the know coefficients, i.e. the terms that compose incremental motions $\boldsymbol{p}_i^k$ and $\boldsymbol{p}_j^k$ for, respectively, the reference sensor and the one to calibrate.

Subsequently, we express the cost function (3.8) in a more compact form:

$$\boldsymbol{\zeta}_{ij}(\boldsymbol{t}) = \frac{1}{2}\, \boldsymbol{\varphi}^{\mathrm{T}}\, \boldsymbol{M}\, \boldsymbol{\varphi} \; + \; C \tag{3.14}$$

where:

- C $\in \mathbb{R}$ is a constant term;

- $\boldsymbol{M} \triangleq \sum_k \boldsymbol{\mathcal{Q}}_k^{\mathrm{T}} \boldsymbol{\mathcal{Q}}_k$ is a symmetric matrix.

With the compact definition of the cost function (3.14), we can finally reduce the least squares formulation (3.9) into a quadratic system with a quadratic constraint:

$$\boldsymbol{\varphi}^* = \underset{\boldsymbol{\varphi}}{\operatorname{argmin}} \ \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{\varphi} \tag{3.15}$$

$$\text{subjected to } \ \varphi_4^2 + \varphi_5^2 = 1 \tag{3.16}$$

The constraint in (3.16) represents the fundamental identity of Trigonometry, i.e. $\cos(t_\theta)^2 + \sin(t_\theta)^2 = 1$ and can be expressed in matrix form as:

$$\boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{\varphi} = 1, \ \text{ where } \boldsymbol{W} \triangleq \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{2\times3} \\ \mathbf{0}_{3\times2} & \boldsymbol{I}_{2\times2} \end{bmatrix} \tag{3.17}$$

The reduction of the calibration problem (3.9) into a quadratic system with a quadratic constraint (3.15), (3.16) is finally completed.

Now we find the solution in close form for this system using the Lagrangian method:

$$\mathcal{L}(\boldsymbol{\varphi}, \lambda) = \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{\varphi} + \lambda \left(\boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{\varphi} - 1 \right) \tag{3.18}$$

and the necessary condition for optimality is:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\varphi}} = 2\boldsymbol{\varphi}^{\mathrm{T}} \left(\boldsymbol{M} + \lambda \boldsymbol{W} \right) = \mathbf{0}^{\mathrm{T}} \tag{3.19}$$

Since the scalar factor $\varphi_1 \in \mathbb{R}^+$ has to be a real positive number, the $\boldsymbol{M} + \lambda \boldsymbol{W}$, that is a 5 x 5 matrix must be singular to satisfy the Equation (3.19). Therefore let's compute $\lambda \in \mathbb{R}$ that assures:

$$\det(\boldsymbol{M} + \lambda \boldsymbol{W}) = 0 \tag{3.20}$$

and the solution $\boldsymbol{\varphi}^*$ can be found in the kernel of $\boldsymbol{M} + \lambda \boldsymbol{W}$.

The expression (3.20) represents a second-order polynomial in λ, due to the composition of the matrix $\boldsymbol{W}$, so it is possible to solve it in close form. The two real roots of this polynomial can be considered as candidate solutions λ^*, from which it is possible to derive

the final candidate solutions $\overline{\boldsymbol{\varphi}^*}$ of the calibration problem.
The rank of the matrix $\boldsymbol{M} + \lambda^* \boldsymbol{W}$ (5 x 5) is at most 4 by construction, but, additionally, we can say that this rank is exactly 4 if the two sensors perform non-degenerated trajectories while at least two independent incremental motions are observed. The observability analysis, to demonstrate the relation between the rank of $\boldsymbol{M} + \lambda^* \boldsymbol{W}$ and the observed incremental motions, is described in the Subsection 3.3.1.

If this rank is 4, then the kernel is one-dimensional and the solution $\boldsymbol{\varphi}^*$ is unique, given the constraints (3.16) and the one regarding the strict positivity of φ_1^*. To find it, let's define $\boldsymbol{\gamma}^*$ as any non-zero vector inside the kernel of $\boldsymbol{M} + \lambda^* \boldsymbol{W}$. The two candidate solutions $\overline{\boldsymbol{\varphi}^*}$, each one associated with one of the two λ^*, are computed with:

$$\overline{\boldsymbol{\varphi}^*} = \frac{\text{sign}(\gamma_1^*)}{\left|\left|\begin{bmatrix}\gamma_4^* & \gamma_5^*\end{bmatrix}^{\mathrm{T}}\right|\right|_2} \boldsymbol{\gamma}^* \tag{3.21}$$

where we have imposed:

- the constraint (3.16) in $\left|\left|\begin{bmatrix}\gamma_4^* & \gamma_5^*\end{bmatrix}^{\mathrm{T}}\right|\right|_2$;
- the positivity of $\overline{\varphi_1^*}$ in $\text{sign}(\gamma_1^*)$, such that $\overline{\boldsymbol{\varphi}^*}$ is uniquely identified from the null space.

Finally, the final solution $\boldsymbol{\varphi}^*$ is chosen between the two candidate solution $\overline{\boldsymbol{\varphi}^*}$ computed with (3.21), as the one that solves the quadratic system (3.15) generating the minimum cost according to (3.14).

At last, the optimal planar extrinsic calibration parameters ${}^i\boldsymbol{t}_j^* \in \text{Sim}(2)$ between the two sensors can be recovered from $\boldsymbol{\varphi}^*$ as:

$${}^i\boldsymbol{t}_j^* = \begin{bmatrix} \varphi_2^* & \varphi_3^* & \text{atan2}\left(\varphi_4^*, \varphi_5^*\right) & {\varphi_1^*}^{-1} \end{bmatrix}^{\mathrm{T}} \tag{3.22}$$

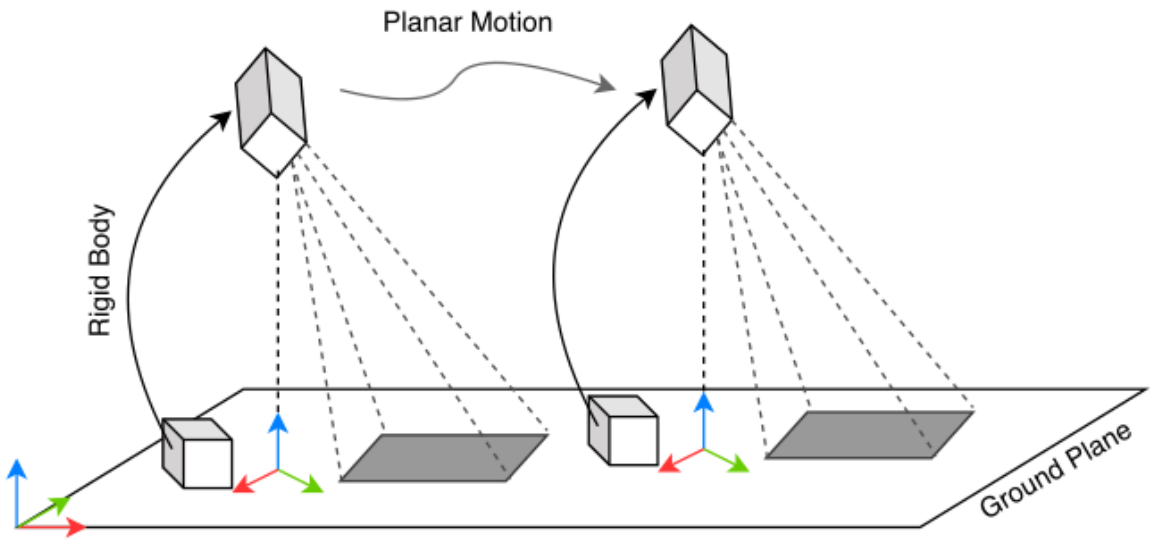


Figure 3.2: The image comes from [29]. It is a representation of two generic sensors, related by a rigid body transformation, moving on a planar surface. The top sensor (e.g a camera) can observe the ground plane during the motion.

3.2. Ground Observation-based Calibration

The second main step in this calibration process is the estimation of the z, *pitch* and *roll* extrinsic parameters of a sensor. The necessary condition for the sensor, to estimate its 3D parameters, is to have a view of the plane in which it moves. In fact, this step is based on the observations of the ground.
The idea is to set a common reference system, such that we can find the extrinsic parameters of the relative pose of the sensor with respect to the plane. With these estimated parameters we can project into the plane the incremental motions enforcing the coplanarity constraint.

3.2.1. Problem Formulation

Formally, the goal is to estimate the parameters of the relative rigid body motion $(\boldsymbol{R}_i, \boldsymbol{T}_i) \in$ SE(3) between an i-th sensor and the ground plane.

Let's start setting the plane's local reference system at the projection of the origin of the sensor on the plane, with the z-axis pointing upwards, as it is shown in Figure3.2. The rotation in-plane, i.e. the one around the z-axis expressed by the yaw parameter, can be arbitrarily set since that parameter is calibrated as t_θ through the motion-based step.

Because three extrinsic parameters $\begin{pmatrix} t_x & t_y & t_\theta \end{pmatrix}$ of the rigid transformation are estimated with the planar calibration described in the previous section, the objective of this step are the remaining three ${}^{i}\boldsymbol{t} = \begin{pmatrix} t_z & t_\psi & t_\phi \end{pmatrix} \in \mathbb{R}^3$ for each sensor. Specifically:

- $t_z \in \mathbb{R}^+$ represents the perpendicular distance of the sensor with respect to the plane;
- $t_\psi \in \mathbb{R}$ represents angular roll rotation, i.e. the one around the x-axis;
- $t_\phi \in \mathbb{R}$ represents angular pitch rotation, i.e. the one around the y-axis.

Firstly, the relative transformation SE(3) can be defined as:

$$\boldsymbol{R}_i \triangleq \boldsymbol{R}_y(t_\phi)\, \boldsymbol{R}_x(t_\psi) \tag{3.23}$$

$$\boldsymbol{T}_i \triangleq \begin{bmatrix} 0 & 0 & t_z \end{bmatrix}^{\mathrm{T}} \tag{3.24}$$

where $R_y(\cdot)$ and $R_x(\cdot)$ represents the parameterized rotation matrix along the y and x axis respectively.

The input of this calibration step is the pose of the j-th point on the ground plane observed by the i-th sensor, that we denominate ${}^{i}\boldsymbol{m}_j \in \mathbb{R}^3$. The vector ${}^{i}\boldsymbol{m}_j$ refers to the coordinates of the j-th ground point defined with respect to the sensor local reference frame.
This point's pose can be expressed, then, with respect to the local reference frame of the ground by applying:

$${}^{gp}\boldsymbol{m}_j = \boldsymbol{R}_i\, {}^{i}\boldsymbol{m}_j \; + \; \boldsymbol{T}_i \tag{3.25}$$

where ${}^{gp}\boldsymbol{m}$ indicates that the vector is expressed through the ground frame whereas ${}^{i}\boldsymbol{m}$ with respect to the sensor frame.

From (3.25) it is possible to define error function that represents the perpendicular distance $\boldsymbol{\eta}_j^i$ of the j-th point to the ground as :

$$\boldsymbol{\eta}_j^i(\boldsymbol{t}) \triangleq \; \boldsymbol{n} \; \cdot \; {}^{gp}\boldsymbol{m}_j \; - \; D \tag{3.26}$$

where the ground plane is defined by the parameters:

- $\boldsymbol{n} \in \mathbb{R}^3$, that is the unit normal vector and it is set, for convenience, to $\boldsymbol{n} \triangleq (0, 0, 1)$;

- D, that is the distance to the origin and it is set to $D \triangleq 0$.

Subsequently, the (3.26) is used to finally define the cost function for the weighted least-squares formulation of the coplanarity relaxation problem:

$$\boldsymbol{\xi}_i(\boldsymbol{t}) \triangleq \frac{1}{2} \sum_j w_j \left\| \boldsymbol{\eta}_j^i(\boldsymbol{t}) \right\|_2^2 \tag{3.27}$$

$${}^i\boldsymbol{t}^* = \underset{\boldsymbol{t}}{\operatorname{argmin}} \ \boldsymbol{\xi}_i(\boldsymbol{t}) \tag{3.28}$$

where $w_j \in \mathbb{R}^+$, present in (3.27), represents the weight.
This coplanarity relaxation problem is solved in close form in the next Subsection 3.2.2.

Now, the incremental motions of the i-th sensor are usually measured in 3D and the one at time step k can be expressed as $(\boldsymbol{R}_i^k \ , \ \boldsymbol{T}_i^k) \in \mathrm{SE}(3)$.
In order to become the input for the motion-based calibration step, these ego-motions need to respect the coplanarity relaxation constraint in the previous section. Therefore it is possible to project these motions to the ground plane using the rotational matrix $\boldsymbol{R}_i$ defined in (3.23) and composed by the extrinsic parameters t_ψ^* and t_ϕ^*, that are estimated solving the (3.28).
The projection can be performed applying:

$$\overline{\boldsymbol{R}_i^k} \triangleq \boldsymbol{R}_i \boldsymbol{R}_i^k \boldsymbol{R}_i^{\mathrm{T}} \tag{3.29}$$

$$\overline{\boldsymbol{T}_i^k} \triangleq \boldsymbol{R}_i \boldsymbol{T}_i^k \tag{3.30}$$

Finally, the planar incremental motions $\boldsymbol{p}_i^k = \begin{pmatrix} p_{i\,x}^k & p_{i\,y}^k & p_{i\,\theta}^k \end{pmatrix} \in \mathrm{SE}(2)$, input of the motion-based process, can be derived in this way:

- $p_{i\,x}^k$ and $p_{i\,y}^k$ are the x-y translation component of the vector $\overline{\boldsymbol{T}_i^k}$;
- $p_{i\,\theta}^k$ is extracted from $\overline{\boldsymbol{R}_i^k}$ as the rotation angle around the z-axis.

3.2.2. Close Form Solution

In order to solve the (3.28), we can adopt the same strategy exploited in Subsection 3.1.2, hence we reduce the least-squares formulation to a quadratic system with a quadratic

constraint. Then, the system is solved in close form with the method of Lagrangian multipliers.

Firstly, we reduce the error function, previously defined in (3.26), that represents the perpendicular distance $\boldsymbol{\eta}_j^i$ to the ground plane of the j-th point, expressed in ${}^{gp}\boldsymbol{m}_j$.
Since the the hessian form of the ground plane parameters are:

$$\boldsymbol{n} \triangleq \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}} \text{ and } D \triangleq 0 \tag{3.31}$$

the perpendicular distance $\boldsymbol{\eta}_j^i$ can be reduced to its third component:

$$\boldsymbol{\eta}_j^i = {}^{gp}m_{j\,z} \tag{3.32}$$

Now, we parameterized the unknown angles t_ψ and t_ϕ with three independent variables:

- $-\sin(t_\phi)$;
- $\cos(t_\phi)\,\sin(t_\psi)$;
- $\cos(t_\phi)\,\cos(t_\psi)$

that can be grouped into the vector:

$$\boldsymbol{r}_z = \begin{bmatrix} -\sin(t_\phi) & \cos(t_\phi)\,\sin(t_\psi) & \cos(t_\phi)\,\cos(t_\psi) \end{bmatrix} \tag{3.33}$$

This vector (3.33) represents the third row of the rotation matrix $\boldsymbol{R}_i$ of the i-th sensor as expressed in (3.23) and it can be exploited to continue the reduction begun with (3.32) into:

$$\boldsymbol{\eta}_j^i = {}^{gp}m_{j\,z} = \boldsymbol{r}_z\,{}^{i}\boldsymbol{m}_j + t_z \tag{3.34}$$

All the unknown parameters can now be reunited into the vector:

$$\boldsymbol{\varphi} = \begin{bmatrix} t_z & -\sin(t_\phi) & \cos(t_\phi)\,\sin(t_\psi) & \cos(t_\phi)\,\cos(t_\psi) \end{bmatrix}^{\mathrm{T}} \tag{3.35}$$

that allows to rewrite the error term from (3.34) in matrix form:

$$\boldsymbol{\eta}_j^i = \boldsymbol{\mathcal{Q}}_j\,\boldsymbol{\varphi} \tag{3.36}$$

where:

$$\boldsymbol{\mathcal{Q}}_j \triangleq \begin{bmatrix} 1 & {}^i m_{j\ x} & {}^i m_{j\ y} & {}^i m_{j\ z} \end{bmatrix} \tag{3.37}$$

The cost function, previously expressed in (3.27), becomes:

$$\boldsymbol{\xi}_i(\boldsymbol{t}) = \frac{1}{2}\, \boldsymbol{\varphi}^{\mathrm{T}}\, \boldsymbol{M}\, \boldsymbol{\varphi} \tag{3.38}$$

where $\boldsymbol{M} \triangleq \sum_j w_j\, \boldsymbol{\mathcal{Q}}_j^{\mathrm{T}}\, \boldsymbol{\mathcal{Q}}_j$ is a symmetric matrix.

At this point, (3.38) is exploited to transform the least squares formulation in (3.28) into a quadratic system with a quadratic constraint:

$$\boldsymbol{\varphi}^* = \underset{\boldsymbol{\varphi}}{\operatorname{argmin}}\ \boldsymbol{\varphi}^{\mathrm{T}}\, \boldsymbol{M}\, \boldsymbol{\varphi} \tag{3.39}$$

$$\text{subjected to } \varphi_2^2 + \varphi_3^2 + \varphi_4^2 = 1 \tag{3.40}$$

This constraint (3.40) corresponds to the orthogonality property of the matrices belonging to SO(3) and can be matrixed in this way:

$$\boldsymbol{\varphi}^{\mathrm{T}}\, \boldsymbol{W}\, \boldsymbol{\varphi} = 1, \text{ where } \boldsymbol{W} \triangleq \begin{bmatrix} 0 & \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{3\times 1} & \boldsymbol{I}_{3\times 3} \end{bmatrix} \tag{3.41}$$

After this, the quadratic system (3.39) with the quadratic constraint (3.40) are solved introducing the Lagrangian exactly as we did in Subsection 3.1.2.
The only difference is that for the matrix $\boldsymbol{W}$ in (3.41) the optimality necessary condition ((3.19) in the previous section) is characterized by a third-order polynomial in λ, therefore there are at most three candidate solutions λ^* that solved it in close form.

At this point we define $\boldsymbol{\gamma}^*$ as any non-zero vector inside the kernel of $\boldsymbol{M} + \lambda^* \boldsymbol{W}$ and from them we can uniquely recover the three candidate solution $\overline{\boldsymbol{\varphi}^*}$, each of one associated with a λ^*. To derive this computation of the candidates solutions we can impose the orthogonality constraint, expressed by (3.40), and the mandatory positivity of the perpendicular distance $\overline{\varphi_1^*}$ as:

$$\overline{\boldsymbol{\varphi}^*} = \frac{\operatorname{sign}(\gamma_1^*)}{\left\| \begin{bmatrix} \gamma_2^* & \gamma_3^* & \gamma_4^* \end{bmatrix}^{\mathrm{T}} \right\|_2} \boldsymbol{\gamma}^* \tag{3.42}$$

Among the three candidates $\overline{\boldsymbol{\varphi}^*}$, let's choose the optimal one $\boldsymbol{\varphi}^*$ as the solution who generate the minimum cost according to the function (3.27).

Finally, the sensor's extrinsic parameters ${}^i\boldsymbol{t}^*$ can be recovered from the vector $\boldsymbol{\varphi}^*$ that we have just computed.

3.3. Additional Notes

3.3.1. Observability Analysis

Let's make an introduction to the observability analysis we have recalled in 3.1.2 and exploited to solve the parameters estimation problems.

In general, the autonomous mobile system behaves according to a continuous dynamic whereas their sensors generate discretized observation. Therefore there are at least two kinds of observability analysis, that differ according to the considered aspect of the model. Indeed they can require a continuous-time or a discrete-time model of the same system and each formalization has different properties.
These two classes are:

1. The first kind of analysis demonstrates a weakly local observability from the point of view of the control theory [12]. For this, the system must be in continuous-time form, i.e. :

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}) \end{cases} \tag{3.43}$$

   where $\boldsymbol{x}$ includes the time-varying state and the parameters whereas $\boldsymbol{y}$ represents the continuous-time observations. Usually, successive Lie derivatives are computed to prove the observability, but this is often not constructive. These techniques are generally nonlinear.

2. The second kind is a "static" analysis where the system must be in the form:

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) \tag{3.44}$$

where, this time, $\boldsymbol{y}$ is a vector of discretized observation. To fulfill the analysis, the idea is to show constructively that the constraints are enough to determine a unique $\boldsymbol{x}$, for specific commands $\boldsymbol{u}$.

In our case, the observations of the exteroceptive sensors are naturally discretized so the second formalization is a better choice. The following analysis expresses static observability for our system.
It is possible to demonstrate that sensors' calibration parameters are observable if we can exploit at least the measurements of two incremental motions, providing that they are independent. Let's consider Log the logarithmic map on SE(2) where Log: SE(2) $\rightarrow$ use(2).

Proposition 3.1. *The parameters of the sensor are observable if and only if there are at least two incremental motions that respect these conditions:*

- *the motions of the reference sensor* $\boldsymbol{p}_i^k$ *and* $\boldsymbol{p}_i^{k+1}$ *are independent, which means that there is no* $\kappa \in \mathbb{R}$ *such that:*

$$log\,(\boldsymbol{p}_i^k) = \kappa \;\; log\,(\boldsymbol{p}_i^{k+1}) \tag{3.45}$$

- *the two motions* $\boldsymbol{p}_i^k$, $\boldsymbol{p}_i^{k+1}$ *are not pure translations.*

The first condition is generic since, for a fixed first incremental motion $\boldsymbol{p}_i^k$, almost every $\boldsymbol{p}_i^{k+1}$ guarantee the observability of the parameters. Moreover, it is possible to notice that an exact trajectory for the mobile system is not required.

At this point, we give a proof to the preposition 3.1, demonstrating the conditions under which the extrinsic parameters ${}^i\boldsymbol{t}_j$ of the j-th sensor with respect to the i-th one are observable.
With the measurements of two motions intervals by two sensors, the following lemma declares the necessary and sufficient conditions for the observability of ${}^i\boldsymbol{t}_j$.

Lemma 3.1. *Given two intervals (time steps k=1 and k=2), where the incremental motions of two sensors (i-th and j-th) are:*

$$\begin{aligned} \boldsymbol{p}_i^1 &= \begin{pmatrix} p_{i\,x}^1 & p_{i\,y}^1 & p_{i\,\theta}^1 \end{pmatrix} \quad \boldsymbol{p}_j^1 = \begin{pmatrix} p_{j\,x}^1 & p_{j\,y}^1 & p_{j\,\theta}^1 \end{pmatrix} \\ \boldsymbol{p}_i^2 &= \begin{pmatrix} p_{i\,x}^2 & p_{i\,y}^2 & p_{i\,\theta}^2 \end{pmatrix} \quad \boldsymbol{p}_j^2 = \begin{pmatrix} p_{j\,x}^2 & p_{j\,y}^2 & p_{j\,\theta}^2 \end{pmatrix} \end{aligned} \tag{3.46}$$

the extrinsic planar parameters ${}^i\boldsymbol{t}_j = \begin{pmatrix} t_x & t_y & t_\theta & t_s \end{pmatrix}$ *are observable if and only if the*

matrix $\boldsymbol{M}$ *(4x5) has rank equal to 4:*

$$\boldsymbol{M} = \begin{bmatrix} -p_{i\ x}^{1} & 1-\cos(p_{j\ \theta}^{1}) & \sin(p_{j\ \theta}^{1}) & p_{j\ x}^{1} & -p_{j\ y}^{1} \\ -p_{i\ y}^{1} & -\sin(p_{j\ \theta}^{1}) & 1-\cos(p_{j\ \theta}^{1}) & p_{j\ y}^{1} & p_{j\ x}^{1} \\ -p_{i\ x}^{2} & 1-\cos(p_{j\ \theta}^{2}) & \sin(p_{j\ \theta}^{2}) & p_{j\ x}^{2} & -p_{j\ y}^{2} \\ -p_{i\ y}^{2} & -\sin(p_{j\ \theta}^{2}) & 1-\cos(p_{j\ \theta}^{2}) & p_{j\ y}^{2} & p_{j\ x}^{2} \end{bmatrix} \tag{3.47}$$

Proof. If we start from the (3.10), that can be rewritten as:

$$^{i}\boldsymbol{t}_{j} \oplus \boldsymbol{p}_{j}^{k} = \boldsymbol{p}_{i}^{k} \oplus {}^{i}\boldsymbol{t}_{j} \tag{3.48}$$

we can derive:

$$t_s \left(R(t_\theta) \begin{bmatrix} p_{j\ x}^{k} \\ p_{j\ y}^{k} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right) = \left(R(p_{i\ \theta}^{k}) \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} p_{i\ x}^{k} \\ p_{i\ y}^{k} \end{bmatrix} \right) \tag{3.49}$$

If we let $p_{i\ \theta}^{k} = p_{j\ \theta}^{k}$ and we exploit the vector $\boldsymbol{\varphi}$. defined in Equation (3.11), we can express this linear constraint as:

$$\begin{bmatrix} -p_{i\ x}^{k} & 1-\cos(p_{j\ \theta}^{k}) & \sin(p_{j\ \theta}^{k}) & p_{j\ x}^{k} & -p_{j\ y}^{k} \\ -p_{i\ y}^{k} & -\sin(p_{j\ \theta}^{k}) & 1-\cos(p_{j\ \theta}^{k}) & p_{j\ y}^{k} & p_{j\ x}^{k} \end{bmatrix} \boldsymbol{\varphi} = \mathbf{0} \tag{3.50}$$

Remember that inside $\boldsymbol{\varphi}$, $\cos(t_\theta)$ and $\sin(t_\theta)$ are considered as independent variables but still constrained to $\cos^2(t_\theta) + \sin^2(t_\theta) = 1$.
Moreover the constraint derived by the two separate incremental motions, at time step k=1 and k=2, can be considered by overlapping twice the (3.50) in this way:

$$\begin{bmatrix} -p_{i\ x}^{1} & 1-\cos(p_{j\ \theta}^{1}) & \sin(p_{j\ \theta}^{1}) & p_{j\ x}^{1} & -p_{j\ y}^{1} \\ -p_{i\ y}^{1} & -\sin(p_{j\ \theta}^{1}) & 1-\cos(p_{j\ \theta}^{1}) & p_{j\ y}^{1} & p_{j\ x}^{1} \\ -p_{i\ x}^{2} & 1-\cos(p_{j\ \theta}^{2}) & \sin(p_{j\ \theta}^{2}) & p_{j\ x}^{2} & -p_{j\ y}^{2} \\ -p_{i\ y}^{2} & -\sin(p_{j\ \theta}^{2}) & 1-\cos(p_{j\ \theta}^{2}) & p_{j\ y}^{2} & p_{j\ x}^{2} \end{bmatrix} \boldsymbol{\varphi} = \mathbf{0} \tag{3.51}$$

This is an homogeneous linear constraint, like $\boldsymbol{M}\ \boldsymbol{\varphi} = 0$ where M is the one in (3.47).
Now, inside $\boldsymbol{\varphi} \in \mathbb{R}^5$ there are the five unknowns of the system and inside $\boldsymbol{M} \in \mathbb{R}^{4\times5}$ the four constraints. Therefore, considering only $\boldsymbol{M}\ \boldsymbol{\varphi} = 0$, the unknown vector $\boldsymbol{\varphi}$ can be observed only up to a 1-dimensional subspace, which is the kernel of $\boldsymbol{M}$, i.e. if $\boldsymbol{\varphi}^*$ is a

solution then, $\forall \alpha \in \mathbb{R}$ also $\alpha \boldsymbol{\varphi}^*$ is a solution.
In order to get a unique solution, we can exploit the constraint (3.16), that binds the absolute value of $\boldsymbol{\varphi}$ and impose $\varphi_1 > 0$ to guarantee the right sign to the unknown vector. In conclusion, $\boldsymbol{\varphi}$ is observable if and only if the rank of $\boldsymbol{M}$ is 4. □

The next step is the check on which motions makes $\boldsymbol{M}$ a 4-rank matrix.
Indeed it is possible to notice that this matrix depends on two incremental motions for each sensor, expressed in (3.46), but these vectors indicates the relative pose of the sensors at the end of each of the two intervals (interval with time step k=1 and k=2). Therefore the observability of the planar extrinsic parameters does not depend on how the sensors achieve their final poses, but only on those specific poses. Particularly, it does not matter if the sensors velocities are constant or time-varying. Moreover, if the parameters ${}^i\boldsymbol{t}_j$ are fixed, the displacements $\boldsymbol{p}_j^1$, $\boldsymbol{p}_j^2$ are directly function of $\boldsymbol{p}_i^1$, $\boldsymbol{p}_i^2$.

Proposition 3.2. *If Log: SE(2) → se(2) is the logarithm map on SE(2), the matrix $\boldsymbol{M}$ has rank less than 4 if and only if one of these conditions occurs:*

- *both $\boldsymbol{p}_j^1$ and $\boldsymbol{p}_j^2$ are pure translations;*
- *there exists a $\kappa \in \mathbb{R}$ such that:*

$$Log\left(\boldsymbol{p}_j^1\right) = \kappa \; Log\left(\boldsymbol{p}_j^2\right) \tag{3.52}$$

Proof. Let's start considering one of the two incremental motions for a sensor as pure translation, so we can choose the j-th sensor first displacement and impose $p_{j\,\theta}^1 = 0$ without loss of generality. It is important that $p_{j\,x}^1$ or $p_{j\,y}^1$ are non zero, otherwise $\boldsymbol{p}_j^1$ became a zero motions and the parameters are unobservable, simply using $\kappa = 0$ in (3.52). Anyway, with $p_{j\,\theta}^1 = 0$, for the transnational components:

$$\begin{cases} p_{i\,x}^1 &= p_{j\,x}^1 \\ p_{i\,y}^1 &= p_{j\,y}^1 \end{cases} \tag{3.53}$$

since the two sensors see the same exact motion and $\boldsymbol{M}$ becomes:

$$\boldsymbol{M} = \begin{bmatrix} -p_{j\;x}^1 & 0 & 0 & p_{j\;x}^1 & -p_{j\;y}^1 \\ -p_{j\;y}^1 & 0 & 0 & p_{j\;y}^1 & p_{j\;x}^1 \\ -p_{i\;x}^2 & 1-\cos(p_{j\;\theta}^2) & \sin(p_{j\;\theta}^2) & p_{j\;x}^2 & -p_{j\;y}^2 \\ -p_{i\;y}^2 & -\sin(p_{j\;\theta}^2) & 1-\cos(p_{j\;\theta}^2) & p_{j\;y}^2 & p_{j\;x}^2 \end{bmatrix} \tag{3.54}$$

If also the second incremental motion for the j-th sensor is a translation ($p^2_{j\,\theta} = 0$) then the rank of $\boldsymbol{M}$ becomes at most 2, since the second and third column of the matrix are zero and the first and fourth are linearly dependent.
Hence, for two pure translation motions of a sensor, the parameters are unobservable.

On the other side, if the second motion has $p^2_{j\,\theta} \neq 0$, the minor of $\boldsymbol{M}$ is:

$$\tilde{\boldsymbol{M}} = \begin{bmatrix} 0 & 0 & p^1_{j\ x} & -p^1_{j\ y} \\ 0 & 0 & p^1_{j\ y} & p^1_{j\ x} \\ 1-\cos(p^2_{j\ \theta}) & \sin(p^2_{j\ \theta}) & p^2_{j\ x} & -p^2_{j\ y} \\ -\sin(p^2_{j\ \theta}) & 1-\cos(p^2_{j\ \theta}) & p^2_{j\ y} & p^2_{j\ x} \end{bmatrix} \tag{3.55}$$

with a determinant equal to:

$$2\ \ \cos(p^2_{j\ \theta} - 1)\ \left\| \begin{pmatrix} p^1_{j\ x} & p^1_{j\ y} \end{pmatrix}^{\mathrm{T}} \right\|^2 \neq 0 \tag{3.56}$$

and $\boldsymbol{M}$ has the rank exactly equal to 4.

In case of no transnational motions, so where $p^1_{j\,\theta}$, $p^2_{j\,\theta} \neq 0$, the motions $\boldsymbol{p}^1_j$ and $\boldsymbol{p}^2_j$ can be written, using the exponential coordinates $(a_1\ \ b_1\ \ \omega_1)$ and $(a_2\ \ b_2\ \ \omega_2)$, in this way:

$$Log\,(\boldsymbol{p}^k_j) = T \begin{bmatrix} 0 & \omega_k & a_k \\ -\omega_k & 0 & b_k \\ 0 & 0 & 0 \end{bmatrix} \in \mathrm{se}(2) \tag{3.57}$$

for $(a_k,\ b_k,\ \omega_k)$ such that $|\omega_i| < \pi/T$, and $a_k,\ b_k \in \mathbb{R}$. The condition $|\omega_i T| < \pi$ assures that this data reparametrization is one-to-one.
These exponential coordinates vectors can be interpreted as constant velocities that would make the sensor achieve the two final poses in time T. However, these vectors are only parametrizations of the motions $\boldsymbol{p}^1_j$, $\boldsymbol{p}^2_j$, where there are no assumptions on constant velocities during the time step.
Now, we demonstrate that $\boldsymbol{M}$ has rank less than 4 if and only if there exists a $\kappa \in \mathbb{R}$ such that:

$$(a_1\ \ b_1\ \ \omega_1)^{\mathrm{T}} = \kappa\ \ (a_2\ \ b_2\ \ \omega_2)^{\mathrm{T}} \tag{3.58}$$

The next Lemma is useful for the rest of the demonstration since it defines an exponential map useful to rewrite the incremental motions and consequently the minor of $\boldsymbol{M}$.

Lemma 3.2. *The exponential map Exp: se(2)→SE(2) can be written as:*

$$\mathrm{Exp}\left(\mathrm{t}\begin{bmatrix}0 & \omega & a\\ -\omega & 0 & b\\ 0 & 0 & 0\end{bmatrix}\right)=\begin{bmatrix}R(\omega t) & \Upsilon(\omega t)\begin{bmatrix}a\,t\\ b\,t\end{bmatrix}\\ 0 \qquad\qquad & 1\end{bmatrix} \tag{3.59}$$

where :

$$\Upsilon(\omega t)=\begin{bmatrix}\dfrac{\sin(\omega t)}{\omega t} & \dfrac{\cos(\omega t)-1}{\omega t}\\ \dfrac{1-\cos(\omega t)}{\omega t} & \dfrac{\sin(\omega t)}{\omega t}\end{bmatrix} \tag{3.60}$$

With the Lemma 3.2 we can write the incremental motion $(p^k_{j\ x}\ \ p^k_{j\ y}\ \ p^k_{j\ \theta})$ in a close form expression dependent on the exponential coordinates $(a_k\ \ b_k\ \ \omega_k)$:

$$p^k_{j\ \theta}=\omega_k T \tag{3.61}$$

$$\begin{aligned}\begin{bmatrix}p^k_{j\ x}\\ p^k_{j\ y}\end{bmatrix}&=\begin{bmatrix}\dfrac{\sin(\omega_k T)}{\omega_k T} & \dfrac{\cos(\omega_k T)-1}{\omega_k T}\\ \dfrac{1-\cos(\omega_k T)}{\omega_k t} & \dfrac{\sin(\omega_k t)}{\omega_k T}\end{bmatrix}\begin{bmatrix}a_k T\\ b_k T\end{bmatrix}\\ &=\frac{1}{\omega_k}\begin{bmatrix}\sin(\omega_k T) & \cos(\omega_k T)-1\\ 1-\cos(\omega_k T) & \sin(\omega_k t)\end{bmatrix}\begin{bmatrix}a_k\\ b_k\end{bmatrix}\end{aligned} \tag{3.62}$$

Let's consider again the 4x4 minor of $\boldsymbol{M}$, previously expressed in (3.55), but adapted to the actual features (no transnational motions) :

$$\tilde{\boldsymbol{M}}=\begin{bmatrix}1-\cos(p^1_{j\ \theta}) & \sin(p^1_{j\ \theta}) & p^1_{j\ x} & -p^1_{j\ y}\\ -\sin(p^1_{j\ \theta}) & 1-\cos(p^1_{j\ \theta}) & p^1_{j\ y} & p^1_{j\ x}\\ 1-\cos(p^2_{j\ \theta}) & \sin(p^2_{j\ \theta}) & p^2_{j\ x} & -p^2_{j\ y}\\ -\sin(p^2_{j\ \theta}) & 1-\cos(p^2_{j\ \theta}) & p^2_{j\ y} & p^2_{j\ x}\end{bmatrix} \tag{3.63}$$

If we substitute (3.61) and (3.62) inside (3.63), it is possible to derive a minor $\tilde{\boldsymbol{M}}$ that is function of only $(a_k\ \ b_k\ \ \omega_k)$ and its determinant is:

$$\det \tilde{\boldsymbol{M}} =$$
$$= T^3 \operatorname{sinc}^2\left(\frac{\omega_1 T}{2}\right) \operatorname{sinc}^2\left(\frac{\omega_2 T}{2}\right) \left((a_2^2 + b_2^2)\omega_1^2 - 2(a_1 a_2 + b_1 b_2)\omega_1\omega_2 + (a_1^2 + b_1^2)\omega_2^2\right) \tag{3.64}$$

Remember that, the zeros of sinc(x) are the same of sin(x), except for x=0 for which sinc(0)=1 and sin(0)=0, thus, regarding the sinc(x), the determinant can be zero only for $\omega_i T/2 = \mathrm{k}\pi$ for |k|>0. However, these zeros are ignored since they corresponds to $\omega_i T = 2k\,\pi$ that are the singularities of the representation, so we have assumed $|\omega_i T| < \pi$ in the constraint of (3.57).
In conclusion the only zeros for the determinant (3.64) are generated by the second factor:

$$\mathrm{d} = (a_2^2 + b_2^2)\,\omega_1^2 \;-\; 2(a_1 a_2 + b_1 b_2)\,\omega_1\,\omega_2 \;+\; (a_1^2 + b_1^2)\,\omega_2^2 \tag{3.65}$$

This expression is a fourth-order polynomial with variables $(a_1 \;\; b_1 \;\; \omega_1)$ and $(a_2 \;\; b_2 \;\; \omega_2)$. Let's consider four cases:

1. The first motion is a rotation, so $a_1 = 0$ and $b_1 = 0$. In this case $d = (a_2^2 + b_2^2)\,\omega_1^2$ and it is a non-zero unless the second motion too becomes a pure rotation.

2. Both the coordinates are non-zero $a_1 \neq 0$ and $b_1 \neq 0$, so we can reparametrize the second motion triplex $(a_2 \;\; b_2 \;\; \omega_2)$ with $(\alpha \;\; \beta \;\; \gamma) \in \mathbb{R}^3$ in this way:

$$\begin{aligned} a_2 &= (\alpha\gamma)a_1 \\ b_2 &= (\beta\gamma)b_1 \\ \omega_2 &= (\gamma)\omega_1 \end{aligned} \tag{3.66}$$

   Since $\omega_1, \omega_2 \neq 0$, then γ must be different from 0.
   If we substitute the parametrization (3.66) in Equation (3.65):

$$\begin{aligned} \mathrm{d} &= (\alpha^2\gamma^2 a_1^2 + \beta^2\gamma^2 b_1^2)\,\omega_1^2 \;-\; 2(\alpha\gamma a_1^2 + \beta\gamma b_1^2)\,\gamma\omega_1^2 \;+\; (a_1^2 + b_1^2)\,\gamma^2\omega_1^2 = \\ &= (\gamma^2\omega_1^2)(\alpha^2 a_1^2 + \beta^2 b_1^2 - 2(\alpha a_1^2 + \beta b_1^2) + a_1^2 + b_1^2) = \\ &= (\gamma^2\omega_1^2)(\alpha^2 - 2\alpha + 1)a_1^2 + (\beta^2 - 2\beta + 1)b_1^2) = \\ &= (\gamma^2\omega_1^2)((\alpha - 1)^2 a_1^2 + (\beta - 1)^2 b_1^2) \end{aligned} \tag{3.67}$$

that become 0 if $\alpha = 1$ and $\beta = 1$, but these value for α and β means that $(a_2 \ b_2 \ \omega_2)$ is proportional to $(a_1 \ b_1 \ \omega_1)$ by a constant γ and this implies a proportionality between the two incremental motions $\boldsymbol{p}_j^1$, $\boldsymbol{p}_j^2$ too.

3. if $a_1 = 0$ and $b_1 \neq 0$ then the case is a simpler variant of the previous one. This time the parametrization is done through $(x \ \beta \ \gamma)$, such that :

$$\begin{aligned} a_2 &= \gamma x \\ b_2 &= (\beta\gamma) b_1 \\ \omega_2 &= (\gamma)\omega_1 \end{aligned} \tag{3.68}$$

and the determinant is proportional to the factor:

$$\begin{aligned} \mathrm{d} &= (x^2 + b_2^2)\omega_1^2 - 2(b_1 b_2)\omega_1\omega_2 + (b_1^2)\omega_2^2 = \\ &= (\gamma^2 x^2 + \beta^2\gamma^2 b_1^2)\omega_1^2 - 2(\beta\gamma b_1^2)\gamma\omega_1^2 + (b_1^2)\gamma^2\omega_1^2 = \\ &= (\gamma^2\omega_1^2)(x^2 + \beta^2 b_1^2 - 2(\beta b_1^2) + b_1^2) = \\ &= (\gamma^2\omega_1^2)(x^2 + (\beta - 1)^2 b_1^2) \end{aligned} \tag{3.69}$$

Therefore, necessarily, x = 0 and $\beta = 1$ and, moreover, in this case a linearly dependence like (3.58) must be present.

4. Finally, for $a_1 \neq 0$ and $b_1 = 0$, the situation is equivalent to the previous one but with a_1 and b_1 inverted, so the conclusion is the same one.

□

At last, we need to demonstrate the equivalence between (3.45) and (3.52), since it is the first one that is present in the thesis of Preposition 3.1.
Recalling that:

- $\boldsymbol{p}_j^k$ can be obtained by $\boldsymbol{p}_i^k$ with (3.6);
- the matrix logarithm satisfies the property: $Log(\boldsymbol{A}\boldsymbol{X}\boldsymbol{A}^{-1}) = \boldsymbol{A} \ Log(\boldsymbol{X}) \ \boldsymbol{A}^{-1}$;

it naturally follows that $Log(\boldsymbol{p}_j^k) = Log(\boldsymbol{p}_i^k)$, so these two (3.45) and (3.52) are demonstrated to be equivalent.

3.3.2. Practical Considerations

Firstly, we analyze the input of the motion-based calibration step, i.e. the sensors' ego-motions, that are usually measured with external strategies like, for example, the one developed by Jaimez et al. [14] for Lidars or ORB-SLAM for cameras [19]. The algorithms of these strategies can be subjected to drifts or tracking failures that generate influent errors in the observed motions and, therefore, in the calibration parameters too. In order to avoid this, it can be useful to couple the close form solution with a *Random Sample Consensus* framework (RANSAC [22]), that discards the outliers among the observed ego-motions. In order to make this selection, the framework requires a threshold defined a-priori applicable to an error function. Therefore, regarding the error, we can choose a function similar to (3.7), since the original has some problems like:

- it mixes the translation and rotation errors even if they have different magnitudes;
- it defines the translation error in the space of the second sensor (j-th);
- it has an arbitrarily scale in case a monocular camera is present.

The selected one for a RANSAC framework is:

$$\boldsymbol{\tau}_{ij}^{k}(\boldsymbol{t}) \triangleq \text{trans}\left(\boldsymbol{p}_i^k - {}^i\boldsymbol{t}_j \oplus \boldsymbol{p}_j^k \ominus {}^i\boldsymbol{t}_j\right) \tag{3.70}$$

that represents the translation error expressed in the space of the i-th sensor and its terms depend on both the translation and rotation parameters.
Anyway, the (3.7) is still exploited to solve the close form solution of the motion-based calibration whereas the new one (3.70) is only used to detect the motions outliers in a RANSAC framework.

A second consideration regards the calibration of multiple sensors. Indeed the motion-based calibration step described in Section 3.1 estimates only the transformation parameters between a couple of sensors, the reference one and the one to calibrate, since it considers the constraints only between them. The idea is to extend this calibration step from two to multiple sensors where we consider additional constraints between them in a joint optimization framework.
Let's give index 0 to the reference sensor and try to calibrate n sensor, thus the goal is to estimate the parameters ${}^0\boldsymbol{t}_1$, ${}^0\boldsymbol{t}_2$, ..., ${}^0\boldsymbol{t}_n \in \text{Sim}(2)$.
The joint calibration problem can be defined as:

$$
\begin{aligned}
{}^0\boldsymbol{t}_1^*,\ {}^0\boldsymbol{t}_2^*,\ \dots,{}^0\boldsymbol{t}_n^* = \underset{{}^0\boldsymbol{t}_1,\ {}^0\boldsymbol{t}_2,\ \dots,{}^0\boldsymbol{t}_n}{\operatorname{argmin}} & \sum_{i=1}^{n} \sum_{k} \left\| \rho \left(\boldsymbol{\tau}_{0i}^{k} \left({}^0\boldsymbol{t}_i \right) \right) \right\|_2^2 \\
& + \sum_{(i,j)\in S} \sum_{k} \left\| \rho \left(\boldsymbol{\tau}_{ij}^{k} \left(\ominus\, {}^0\boldsymbol{t}_i \oplus {}^0\boldsymbol{t}_j \right) \right) \right\|_2^2
\end{aligned}
\tag{3.71}
$$

where S is a set that contains all the pairs of sensors whose additional constraints are considered. These pairs need the first sensor to measures the motions in a metrically accurate way.
In the upwards formula the error function is the modified one $\boldsymbol{\tau}$ defined in (3.70) instead of $\boldsymbol{\varepsilon}$ defined in (3.7) since we want all sensors' terms of error expressed in the same metric space. The term $\ominus\, {}^0\boldsymbol{t}_i \oplus {}^0\boldsymbol{t}_j$ is equivalent to the extrinsic parameters of the relative transformation ${}^i\boldsymbol{t}_j$ between i-th and j-th sensors, where we highlighted the single calibration parameters ${}^0\boldsymbol{t}_i$, ${}^0\boldsymbol{t}_j$ between each sensor and the reference one. The symbol ρ represents the Cauchy loss function used to face the unmodeled mistakes not detected during the step of the RANSAC framework.
The joint calibration problem expressed in (3.71) can be solved iteratively from the close form solution described in Subsection 3.1.2, using the g^{2}o framework depicted by Kümmerle et al. in [16].

Finally, at last, the whole proposed calibration process for the 6 DoF extrinsic parameters can be summarized in two main separate steps. Firstly, there is the 3D parameters (z, *pitch* and *roll*) estimation, based on the ground plane observations and, secondly, the planar calibration of x, y and *yaw* parameters based on sensors' ego-motions. More specifically, the precise calibration pipeline is:

1. acquisition of the data from the sensors;
2. estimation of the motions for each sensor with methods like the ones described in [14] and [19];
3. estimation of the z, *pitch* and *roll* parameters through plane observations as described in Section 3.2 for each sensor that needs to be calibrated;
4. projection of the estimated trajectories into the ground plane through the (3.29) and (3.30), plus their re-sample into synchronous incremental motions;
5. calibration of the remaining parameters x, y, *yaw* through the motion-based step described in Section 3.1;
6. refinement of these parameters in a joint optimization framework through (3.71).

4 | Validation Setup and Dataset

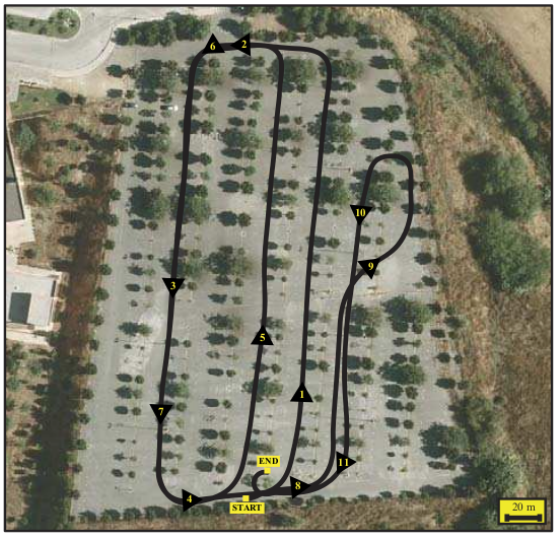
After the explanation of the theory in the previous chapter, we can now focus on an experimental estimation of the sensors' parameters. The idea is to use the measurements from a collection of datasets to estimate the extrinsic parameters of the sensors, following the pipeline described in the previous chapter. The choice of the exploited datasets and programs is inspired by the section *"Outdoor Evaluation"* (Section V-B) of Zuñiga-Noël et al. [29].

4.1. Datasets

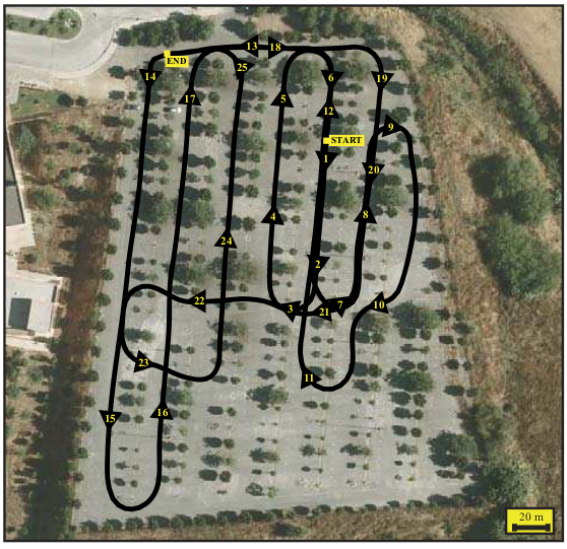
First of all, the collection of datasets exploited in this experimental evaluation is the one created by Jose-Luis Blanco, Francisco-Angel Moreno and Javier Gonzalez and described in [3]. The contained measurements are generated with a mobile platform equipped with multiple sensors, like RTK-GPS, lidars and cameras. The vehicle was driven along six different paths in Málaga, three of which are placed in the parking of the Computer Science School building while the others are at the Campus boulevard of the University of Málaga. For each of them, a dataset of measurements is generated and stored inside the collection.
However, a great feature of the Blanco et al. collection is the presence of *ground truth* for the vehicle paths, which is essential for the motion-based calibration techniques and SLAM processes.

Now, among the six paths followed by the vehicle, we are interested only in the three related to the parking environment, since they are more useful for our calibration goals. Indeed, these are mainly planar, with more loops than the campus ones. Additionally, their environment is full of trees and parked cars, whereas it lacks planar surfaces like buildings. These paths and their related datasets are called 0L, 2L, 6L depending on the number of loops they included.

(a) Top view of 0L path.



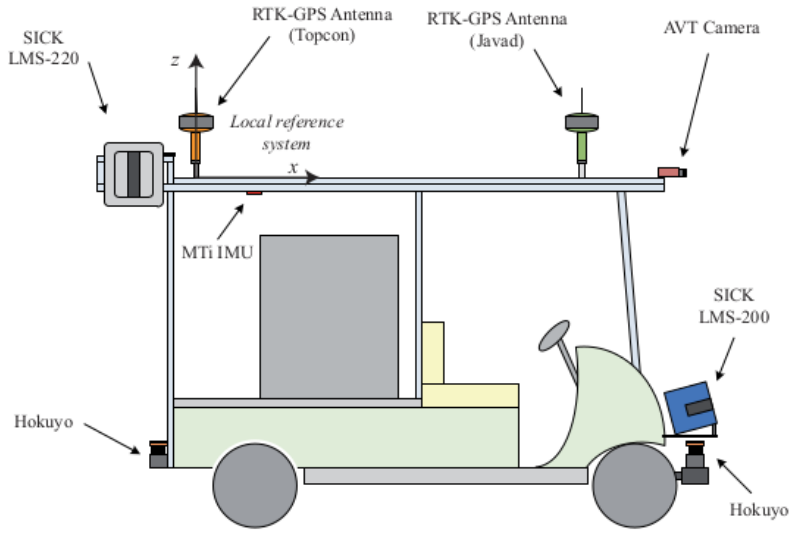
(b) Top view of 2L path.



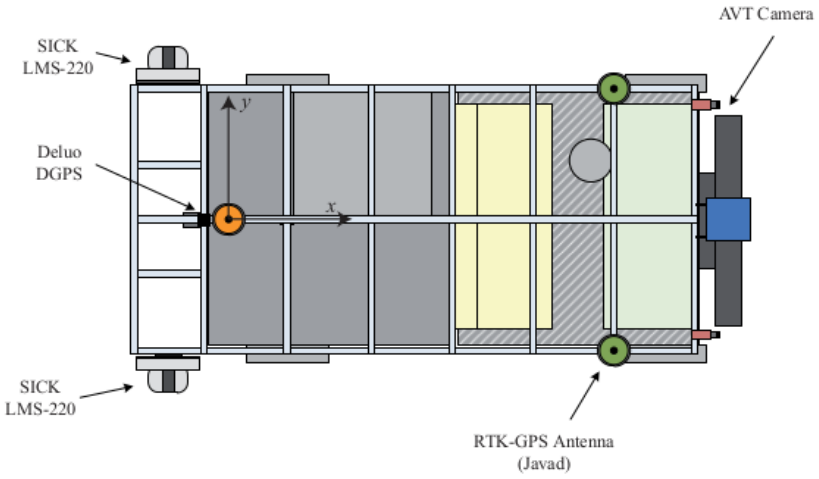
(c) Top view of 6L path.

Figure 4.1: These images come from [3] and represent the paths 0L(a), 2L(b) and 6L(c) where the datasets' measurements are collected.

In Figure 4.1 it is possible to see the three different routes followed by the mobile platform and a top view of their common environment. The three datasets are exploited, during the calibration, to estimate a set of extrinsic parameters for each trajectory and check how the choice of the path can influence the results.



(a) Lateral view.



(b) Top view.

Figure 4.2: The image comes from [3] and it represents the position of the sensors in the mobile platform. This image shows an error since the model of two lateral rear lidars is, in reality, the SICK-LMS 221.

The platform, exploited to measure the parking environment, is equipped with twelve heterogeneous sensors, including five lidars, two cameras, four GPS and one IMU. In Figure 4.2 there is a representation where the sensors are highlighted. Each type of sensor has been calibrated with a specific technique in order to achieve both accurate measurements and ground truth for their extrinsic parameters. For detailed information about the calibration of each specific type of sensor executed by Blanco et al., we refer the reader to the reference document [3].
Table 4.1 displays the ground truth 6 DoFs extrinsic parameters of the sensors in the

platform, that are calibrated with respect to the local reference frame represented in Figure 4.2.
Four sensors are highlighted since they are the only ones whose poses are completely optimized or completely calibrated. The selected ones are the two AVT cameras and the two SICK-LMS 221 Lidars, therefore these are the chosen sensors we are going to calibrate with the strategy described in Chapter 3.
This table is very important since we can confront our results with the ground truth of the calibration parameters contained in it.

Additionally, a very useful aspect of the Blanco et al. collection is the presence of the ground truth of the sensors' trajectories which can be exploited for our motion-based calibration step. In fact, in each dataset's folder there exist some text files filled with the trajectories of the four chosen sensors sampled at 1 Hz (*GT_path_<sensor>_name.txt*) and the vehicle trajectory, sampled at 1 Hz and 100 Hz (respectively *GT_path_vehicle.txt* and *GT_path_vehicle_interp.txt*). The vehicle trajectory is measured by combining the data from the RKT GPS devices whereas the sensors' ego-motions are generated by applying the appropriate and specific 6D pose to the vehicle trajectory.

	x (m)	**y** (m)	**z** (m)	$\boldsymbol{\theta}$ (deg.)	$\boldsymbol{\psi}$ (deg.)	$\boldsymbol{\phi}$ (deg.)
Rear GPS-RTK	0.000	0.000	0.132	×	×	×
Front Left GPS-RTK	1.729	0.5725	0.115	×	×	×
Front Right GPS-RTK	1.733	-0.5725	0.1280	×	×	×
DGPS*	-0.250	0.000	0.100	×	×	×
Left AVT Camera	2.216	0.430	0.022	-88.43	-87.23	-2.99
Right AVT Camera	2.200	-0.427	0.025	-90.31	-86.19	-3.53
Front Hokuyo*	2.420	0.000	-1.740	0.00	0.00	0.00
Rear Hokuyo*	-0.299	0.084	-1.725	178.81	0.00	0.00
Front SICK LMS-220*	2.278	0.000	-1.565	0.00	0.00	-6.84
Left SICK LMS-221	-0.3642	0.7899	0.0441	90.58	-89.66	6.82
Right SICK LMS-221	-0.3225	-0.8045	-0.0201	-90.33	89.85	-2.87
IMU*	×	×	×	0.000	0.000	0.000

*Sensor pose is not optimized. × (irrelevant or not applicable).

Table 4.1: This table comes from [3] and represents the ground truth of the extrinsic parameters for the sensors in the platform.

Moreover, the folder is full of the sensor measurements, obviously saved in different formats. For example, in the folder "Images_rect" there are the frames of the right and left cameras, where the images are rectified to compensate for the camera distortion. They are exploited in the step based on the ground observations to estimate the two cameras' parameters. Another useful file is *pointcloud.txt*, which is composed of the cloud of the environment points measured by the Lidars. However, all the data grabbed by the sensors are included in the binary log file *parking_ <name of dataset>L.rawlog*, which is based on the open-source MRPT.

4.2. Ground Observation-based Calibration

At this point, we can focus on the ground observation-based calibration where we exploit the observations of the ground plane performed by the sensors to estimate their (z, ψ, ϕ) extrinsic parameters. As mentioned in Section 4.1 we calibrate the left and right SICK LMS-221 Lidars and AVT cameras. The results can be compared with their ground truth contained in Table 4.2.

The pipeline to follow for this experimental calibration step is:

1. elaborate the data from the dataset to reconstruct the environment;
2. exploit the reconstruction to estimate the ground plane and hence, the ground points;
3. estimate the extrinsic parameters from the list of ground points with the method described in Section 3.2.

	z (m)	$\boldsymbol{\psi}$ (deg.)	$\boldsymbol{\phi}$ (deg.)
Left AVT Camera	2.250	-87.23	-2.99
Right AVT Camera	2.253	-86.19	-3.53
Left SICK LMS-221	2.2721	-89.66	6.82
Right SICK LMS-221	2.2079	89.85	-2.87

Table 4.2: This table represents the ground truth 3D parameters of the sensors we calibrate. It is obtained from the Table 4.1, even if the **z** parameter is increased by an offset of 2.228 m. The reason for that is that our ground observations-based calibration step estimates **z** with respect to the ground and not to the reference frame, represented in Figure 4.2, as in the previous table.

4.2.1. Environment Reconstruction

The reconstruction of the environment can be performed using different programs depending on the sensor type.
In order to calibrate the cameras, we can follow the suggestions of Zuñiga-Noël et al.[29] and exploit the Structure-from-Motion approach [25] through its implementation called COLMAP [24], [26]. We select 50 consecutive frames with high texture on the ground from the folder "Images_rect" and we use them as input to the program in addition to the intrinsic parameters of the camera we want to calibrate. These intrinsic parameters are findable inside the dataset file *malaga_datasets_grabber.ini* and their imposition to COLMAP during the reconstruction allows for an increase in the precision of the reconstructed environment points.
At the end of the reconstruction process, we find in the outcome folder the file *points3D.txt* with the list of 3D environmental features. This list has a row for each reconstructed point, where its (x, y, z) coordinates are expressed.
Additionally, Colmap generates another file, called *images.txt*, where there are 50 sets of parameters that can be used to compute the reconstructed poses of the camera. Each one of them is related to one of the input frames. This set of parameters is composed of three transnational ones and another four that represent a quaternion. In order to obtain the coordinates of reconstructed camera centres, it is necessary to apply the formula $-\boldsymbol{R}^{\mathrm{T}} * \boldsymbol{t}$ where $\boldsymbol{R}$ is the rotational matrix composed of the quaternion and $\boldsymbol{t}$ is the vector composed by the transnational parameters.
It is essential to notice that both the reconstructed environmental features and camera poses are expressed in a reference frame centred on a random point, which doesn't correspond to any camera poses. Therefore, after the Colmap reconstruction, we choose one of the camera poses and we apply a roto-translation off all the points to match the centre of the reconstruction reference frame with the chosen point.
For example, in Figure 4.3, we can see a reconstruction performed by Colmap on 50 consecutive left camera frames during the 0L path. The reference frame is placed in the 25-th reconstructed pose.
It is possible to notice that Colmap generates a piece of camera trajectory since it reconstructs 50 of its poses, but this one is not equal to the ground truth one placed in the Blanco et al. dataset. Indeed between these two sampled paths, there is a similarity transformation that defines their differences.
Therefore, when we estimate the z parameter of the camera exploiting the Colmap's reconstruction, it will be scaled by a factor with respect to the ground truth one.

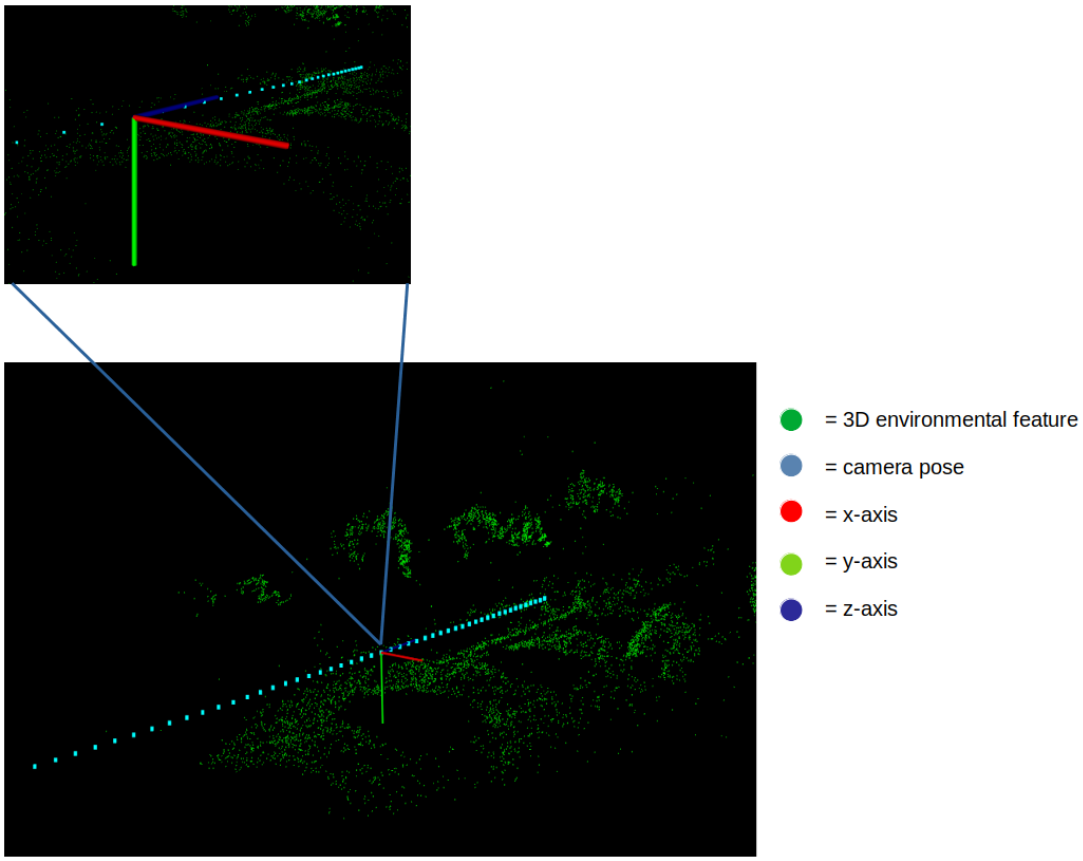


Figure 4.3: Dataset 0L. A plot of the reconstructed environment after matching the local reference frame centre with the 25-the camera pose.

In order to correct this error, we can use a program based on least-squares minimization that compute the parameters set ($\boldsymbol{R}_s$, $\boldsymbol{t}_s$, c_s) of the similarity transformation. Before that, we extrapolate from the total ground truth trajectory only the 50 poses that correspond to the reconstructed ones and we use them in the minimization program. We can see in Figure 4.4 the difference between the reconstructed trajectory (green) and the ground truth one (red) from the 0L dataset, both belonging to the left camera. The blue points represent the reconstructed ones after the application of the estimated transformation and in order to check its correctness. Actually, the only parameter that we are interested in is the scaling factor c_s, which must be multiplied by the estimated z extrinsic parameter of the camera to find its correct value.

Regarding the reconstruction of the environment through the lidars measurements, we can use another program. Indeed, we can read and manipulate the file *parking_<name of dataset>L.rawlog*, which includes all the measurements of the selected dataset, with the program RawLogViewer inside the open-source MRPT.

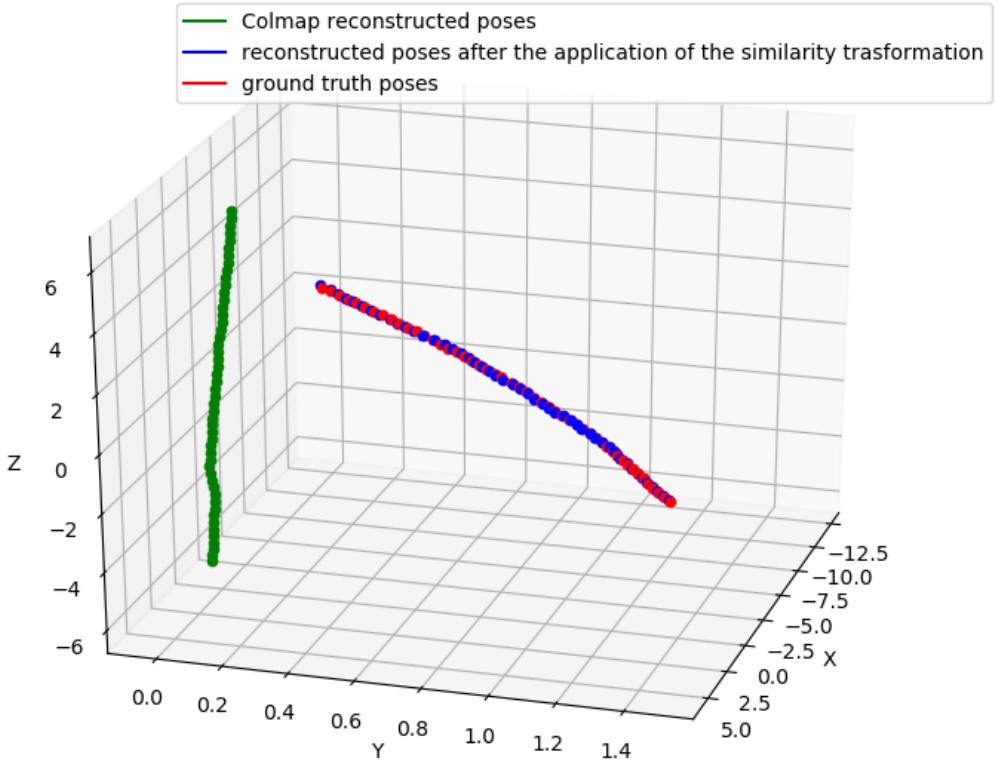


Figure 4.4: Representation of the ground truth camera trajectory and reconstructed ones before and after the application of the transformation parameters. The trajectories belong to the left camera and are generated by the 0L dataset.

Let's select only the measurements from the lidar we want to focus on (left or right), plus the data from the RKT-GPS devices and generate the so-called "map from RKT GPS". As we can see in Figure 4.5, this map is composed of all the 3D points of the environments reconstructed with the scans from the selected lidar. Moreover, there are plotted also ground-truth poses of the sensor.
With the RawLogViewer program, it is possible to save all the 3D points of the environment in a text file, generating a list of 3D coordinates. Also for this kind of sensor is essential to centre the reference frame into a chosen pose of the sensor as we did in the camera's environment reconstruction.

Finally, for both lidars and cameras, the reconstructed environment can be generated like a list of points in a text format. In the next passage, the reconstruction is exploited to estimate the ground plane and the ground points to calibrate the sensor.

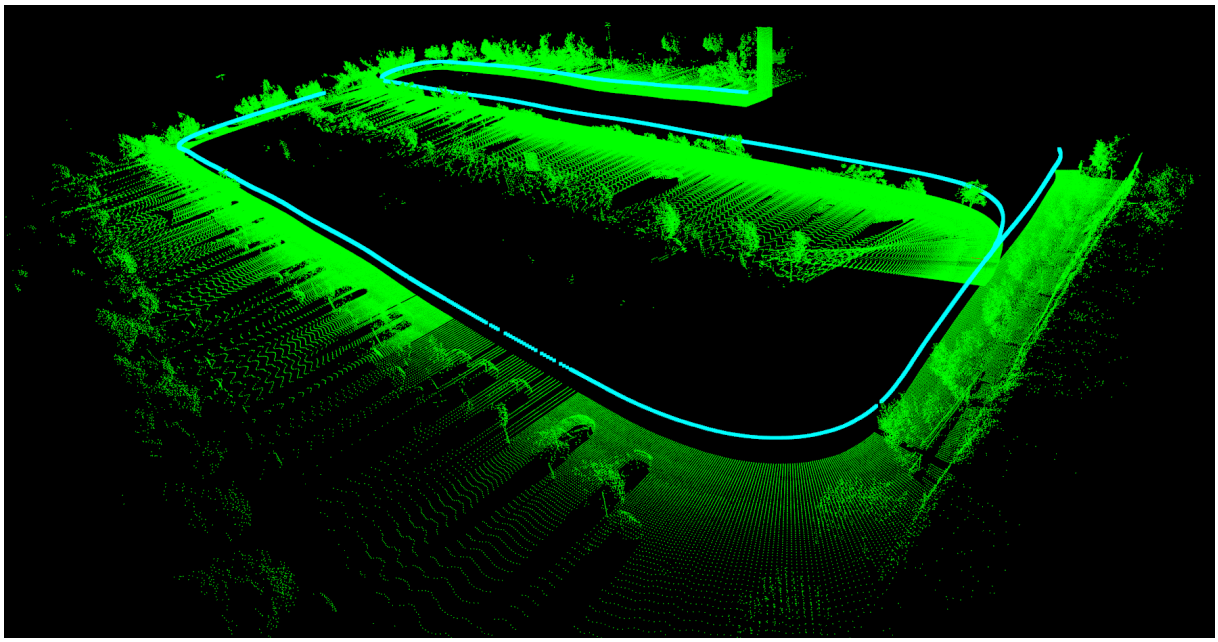


Figure 4.5: Reconstruction of the environment through the scans of the left lidar, which are collected inside the 0L dataset. In this example, the local reference frame is centred in the first lidar pose.

4.2.2. Ground Points Estimation

At this point, with the environment reconstruction, we can estimate the ground points useful to complete the sensor's calibration. These points are the ones that belong to the ground plane so its coefficients are the first we need to compute. The idea is to estimate the coefficients of the plane that includes the set with the highest number of 3D points in the environment reconstruction. This estimated plane corresponds to the ground one and the inlier points are the ground ones. Obviously, this strategy works better for reconstruction with a high number of reconstructed 3D points on the ground.
Now, to implement this strategy, it is possible to exploit the "plane model segmentation" tutorial of the Point Cloud Library (PCL). The whole points in the reconstruction are grouped in a principal point cloud. Then, the program generates and fills a second one with all the points whose distance from a plane is lower than a pre-set near-zero threshold. In this way, the coefficients of the plane with the highest number of inliers, are finally estimated.
In Figure 4.6 it is possible to see an example of the ground points estimation for the left camera moving along a piece of the 0L trajectory. Anyway, this strategy works for both the lidars and the cameras since it only needs the list of environment points expressed with respect to a selected sensor's pose.

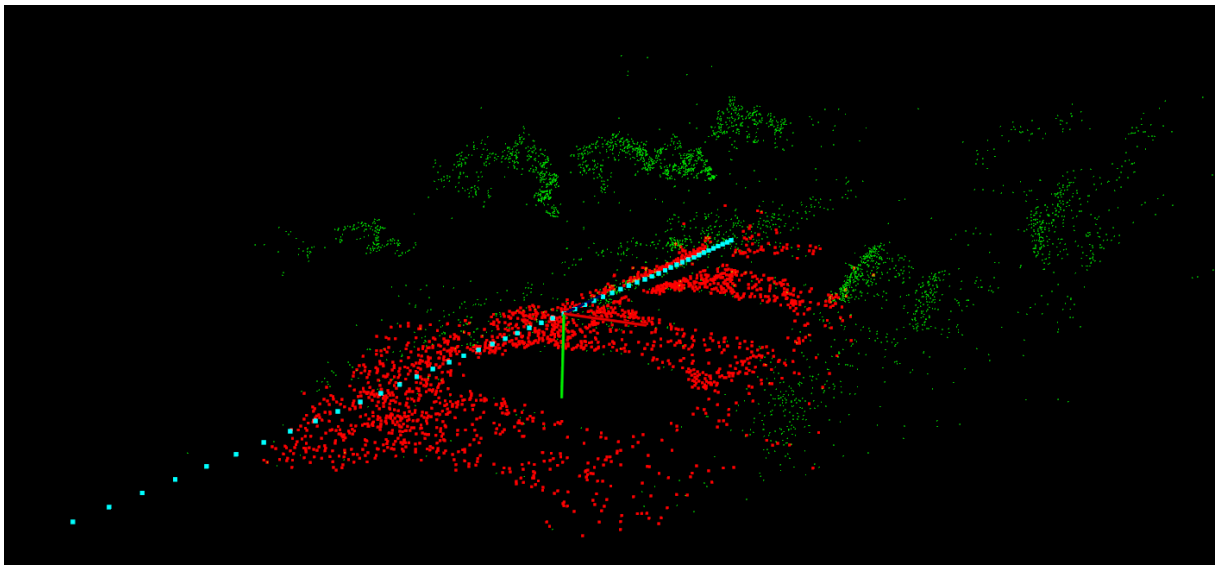


Figure 4.6: This image represents the same reconstruction of Figure 4.3 but with the ground points highlighted in red.

At the end of the estimation, a .csv file must be generated with the list of the ground points, since this is the input for the effective and final step of the ground observations-based calibration.

4.2.3. 3D Parameters Calibration

The conclusive passage of this calibration is the effective estimation of the (z, ψ, ϕ) extrinsic calibration parameters. As mentioned before, we need the list of ground points derived in the previous subsection, since they are the input for the *ground_calib.cpp* file. This letter is part of the C++ implementation of Zuñiga-Noël et al.[29] and it solves the least-squares formulation of the parameters' estimation in close form, following the procedure described in Subsection 3.2.2.
Additionally, the close form solution is refined in an iterative way using the Ceres solver [1] in order to improve the results.

4.3. Motion-based Calibration

In this section, let's perform the estimation of the remaining extrinsic parameters, i.e. the planar ones (x, y, θ) for the four selected sensors of the Blanco et al. datasets. Table 4.3shows the ground truth extrinsic parameters that we can use to confront the ones we estimated in this calibration step.

	x (m)	**y** (m)	$\boldsymbol{\theta}$ (deg.)
Left AVT Camera	2.216	0.430	-88.43
Right AVT Camera	2.200	-0.427	-90.31
Left SICK LMS-221	-0.3642	0.7899	90.58
Right SICK LMS-221	-0.3225	-0.8045	-90.33

Table 4.3: Table with the ground truth of the planar extrinsic parameters, obtained from Table 4.1.

We use the sensors' ground truth trajectories placed in the Blanco et al. datasets as inputs of this process. Since the motion-based calibration estimates the planar parameters of the sensors with respect to a reference one, we can use the so-called vehicle trajectory (*GT_path_vehicle.txt*), measured with the RKT-GPS devices, as the reference one.
However, a necessary condition for this step is the coplanarity between the input trajectories, thus we can use the previously estimated (ψ, ϕ) extrinsic parameters of each sensor to project its trajectory into the ground plane. This procedure can be performed by *planar.cpp* file in the C++ implementation of [29], where (3.29), (3.30) are recalled.
Additionally, if we want to calibrate simultaneously multiple sensors, all their ego motions must be synchronous. Therefore, we can use the *sync.cpp* file, from the same repository linked to [29], to complete this job. Indeed, the program can manipulate the input trajectories and synchronize them with the strategy described at the beginning of Section 3.1. It works with at least two trajectories, where the first one is the reference one, but also with five of them like it is in our case.
Finally, with the coplanar and synchronized sensors ego-motions, the last step is the effective calibration of the planar parameters and the scaling factor, in the case of monocular cameras. The solutions for this calibration are computed firstly in close form, as explained in Subsection 3.1.2 and then refined iteratively with Ceres Solver[1]. To implement this procedure, we use the *calibrate.cpp* file, that can be found in the previously cited repository.

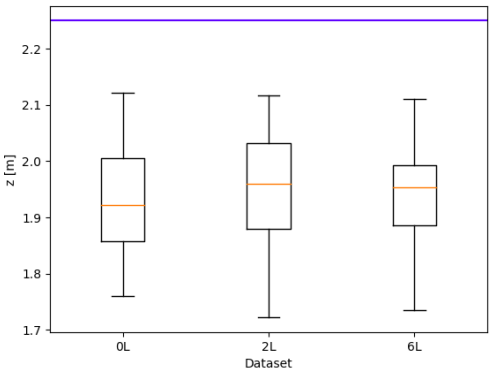
5 | Experimental Results

In the previous chapters, we described the procedure to calibrate two lidars and two cameras, using the Blanco et al. datasets collection.
Therefore, in this chapter, we expose the results of this experimental calibration and we compare the estimated extrinsic parameters with their ground truth values from Tables 4.2 and 4.3.

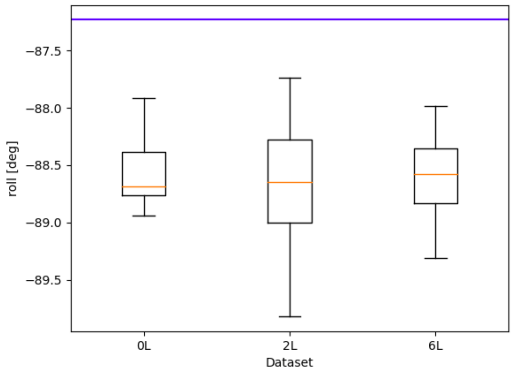
5.1. 3D Parameters Results

This section shows the results computed during the ground observation-based calibration step, described in Section 4.2. As explained in Subsection 4.2.1, during the reconstruction of the environment, we apply a transformation in order to match the centre of the local reference frame with the pose of the sensor. Therefore, in this way, the extrinsic parameters (z, ψ, ϕ) are estimated with respect to that specific pose.
For example, if we calibrate one of the cameras, we reconstruct 50 sensor poses for each path, so we can estimate 50 times the extrinsic parameters, each one with respect to one of the different poses. On the other side, the Lidars do not require the reconstruction of the poses, hence we select 50 of them, equally spaced, from the ground truth trajectories and we use them to match the environment's local reference frame.

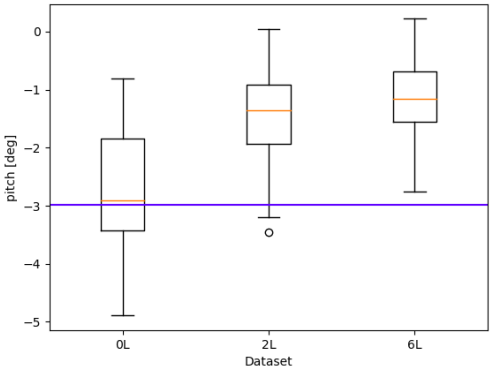
Figure 5.1 shows three box plots for each parameter (z, ψ, ϕ) of the Left AVT Camera with a blue horizontal line that indicates the ground-truth value for that parameter. Every single plot represents the values of one parameter expressed with respect to each of the 50 different reconstructed poses. Moreover, Table 5.1 reports the mean values between the 50 estimations for each parameter and for each dataset. Additionally, it also shows the average values between the three datasets, the standard deviations and the numerical values of the ground truth.
Figures 5.2, 5.3, 5.4 and Tables 5.2, 5.3, 5.4 show the results of the calibration step based on ground-observation for the other three sensors of the Blanco et al. datasets collection.



(a) z parameter calibration.



(b) ψ parameter calibration.



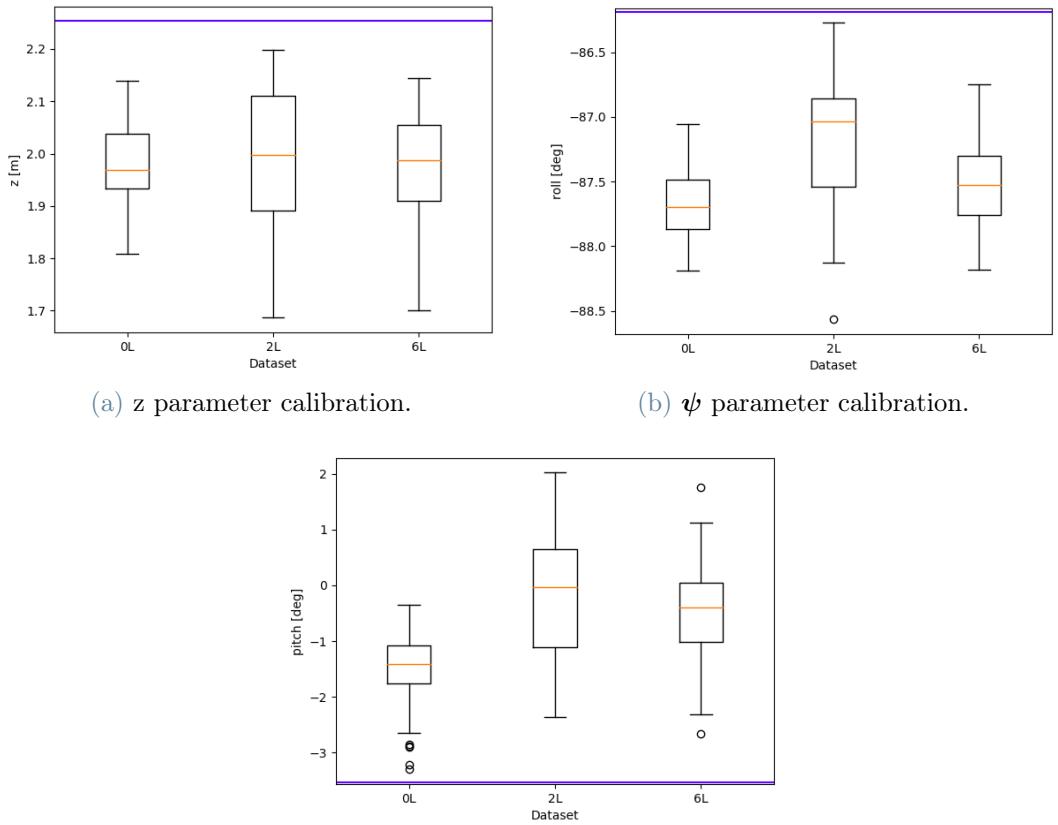
(c) ϕ parameter calibration.

Figure 5.1: Left AVT Camera. The box plots show the 3D calibration parameters estimated with respect to each different reconstructed sensor's pose and with respect to each different path.

Left AVT Camera

	z (m)	ψ (deg.)	ϕ (deg.)
Ground-Truth	2.250	-87.23	-2.99
0L mean value	1.926	-88.56	-2.67
2L mean value	1.950	-88.67	-1.46
6L mean value	1.943	-88.59	-1.19
Average value	1.93959	-88.60467	-1.77255
stdv.	0.010082	0.0485550	0.6425903

Table 5.1: Left AVT Camera. This table shows the mean values computed for each 3D extrinsic parameter and each dataset, plus the average values between the datasets and the standard deviations.



(a) z parameter calibration.

(b) ψ parameter calibration.

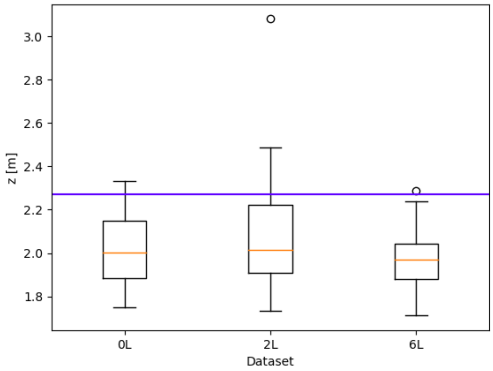
(c) ϕ parameter calibration.

Figure 5.2: Right AVT Camera. The box plots show the 3D calibration parameters estimated with respect to each different reconstructed sensor's pose and with respect to each different path.

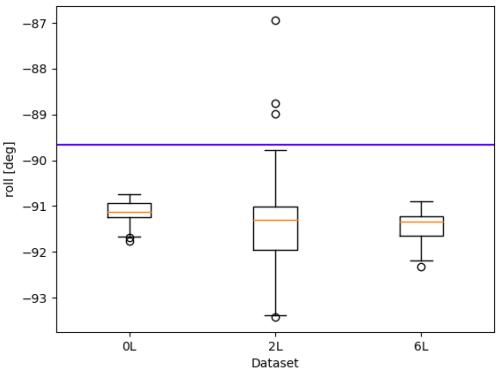
Right AVT Camera

	z (m)	ψ (deg.)	ϕ (deg.)
Ground-Truth	2.253	-86.19	-3.53
0L mean value	1.974	-87.67	-1.51
2L mean value	1.990	-87.19	-0.19
6L mean value	1.973	-87.48	-0.46
Average Value	1.97921	-87.45	-0.72007
stdv.	0.0074682	0.1950362	0.5717682

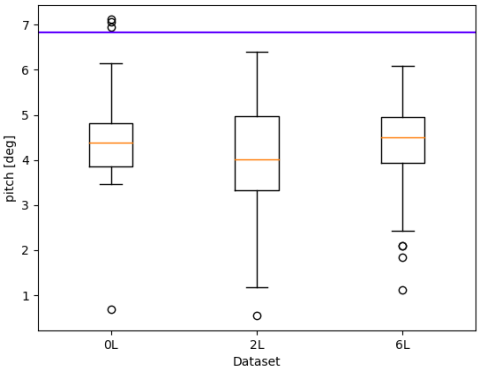
Table 5.2: Right AVT Camera. This table shows the mean values computed for each 3D extrinsic parameter and each dataset, plus the average values between the datasets and the standard deviations.



(a) z parameter calibration.



(b) ψ parameter calibration.



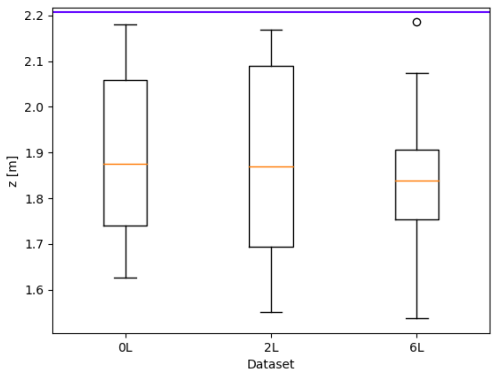
(c) ϕ parameter calibration.

Figure 5.3: Left SICK LMS-221 Lidar. The box plots show the 3D calibration parameters estimated with respect to each different sensor's pose and with respect to each different path.

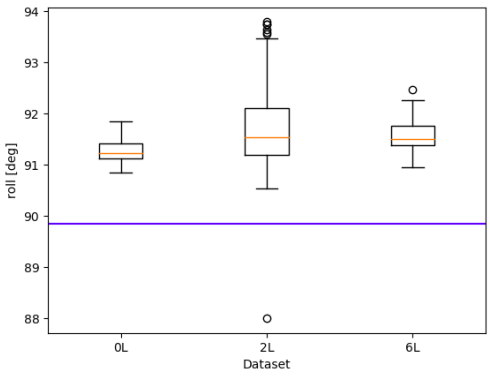
Left SICK LMS-221 Lidar

	z (m)	ψ (deg.)	ϕ (deg.)
Ground-Truth	2.2721	-89.66	6.82
0L mean value	2.0199	-91.13	4.48
2L mean value	2.0572	-91.37	4.06
6L mean value	1.9675	-91.45	4.38
Average Value	2.01488	-91.31467	4.30410
stdv.	0.0368001	0.1360299	0.1787142

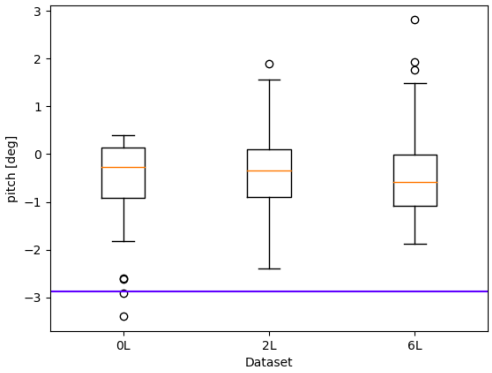
Table 5.3: Left SICK LMS-221 Lidar. This table shows the mean values computed for each 3D extrinsic parameter and each dataset, plus the average values between the datasets and the standard deviations.



(a) z parameter calibration.



(b) ψ parameter calibration.



(c) ϕ parameter calibration.

Figure 5.4: Right SICK LMS-221 Lidar. The box plots show the 3D calibration parameters estimated with respect to each different reconstructed sensor's pose and with respect to each different path.

Right SICK LMS-221 Lidar

	z (m)	ψ (deg.)	ϕ (deg.)
Ground-Truth	2.2079	89.85	-2.87
0L mean value	1.8981	91.27	-0.55
2L mean value	1.8881	91.80	-0.43
6L mean value	1.8290	91.58	-0.43
Average Value	1.87174	91.55151	-0.47118
stdv.	0.0304782	0.2198135	0.0572101

Table 5.4: Right SICK LMS-221 Lidar. This table shows the mean values computed for each 3D extrinsic parameter and each dataset, plus the average values between the datasets and the standard deviations.

It is easy to notice from the tables that the ϕ parameters derived from the 0L dataset are closer to the ground truth value. This, most probably, depends on the greater amount of texture on the ground that is present in the 0L path measurements with respect to the other ones.
Additionally, the estimation of this parameter for the cameras has higher variability than the z and ψ, since it has a greater standard deviation. This is probably caused by the intervention of the Colmap program that reconstructs the environment only during the cameras calibration. On the other side, the z parameter is the one with lower standard deviations for all the sensors, even if the differences with respect to its ground truth value lie in a range of 20-35 cm.

5.2. Planar Parameters Results

At this point, we show the results of the motion-based calibration step described in Section 4.3.
Tables 5.5, 5.6, 5.7 and 5.8 express the numerical solution of the (x, y, θ) parameters calibration for the four selected sensors in each of the three datasets. In order to compare in an easier way these values with the ground truth ones, the letters are reported here from Table 4.1. Additionally, as in the previous subsection, we compute the average values for each parameter and the standard deviations.
As stated in Section 4.3, the input ego-motions of the sensors must be coplanar, therefore we need to exploit their pitch and roll extrinsic parameters to project the trajectories in a common plane. However, in this experimental calibration, we decide to use the ground truth values for these angular parameters since we want to check the correctness of this estimation step without the influence of the one that estimates the ψ and ϕ parameters.

Left AVT Camera

	x (m)	**y** (m)	$\boldsymbol{\theta}$ (deg.)
Ground-Truth	2.216	0.430	-88.43
0L	2.218	0.429	-88.43
2L	2.217	0.433	-88.43
6L	2.215	0.425	-88.43
Average Value	2.21665	0.42889	-88.42992
stdv.	0.0012497	0.00333	0.0031149

Table 5.5: Our estimations of the (x, y, θ) parameters for the Left AVT Camera.

Right AVT Camera

	x (m)	**y** (m)	***θ*** (deg.)
Ground-Truth	2.200	-0.427	-90.31
0L	2.183	-0.436	-90.37
2L	2.330	-0.441	-90.34
6L	2.234	-0.439	-90.30
Average Value	2.24896	-0.43857	-90.33690
stdv.	0.0607178	0.0022070	0.0296625

Table 5.6: Our estimations of the (x, y, θ) parameters for the Right AVT Camera.

Left SICK LMS-221 Lidar

	x (m)	**y** (m)	***θ*** (deg.)
Ground-Truth	-0.3642	0.7899	90.58
0L	-0.3843	0.7777	90.49
2L	-0.3841	0.8061	90.59
6L	-0.3425	0.7890	90.79
Average Value	-0.37032	0.79091	90.62244
stdv.	0.0196684	0.0116646	0.1258744

Table 5.7: Our estimations of the (x, y, θ) parameters for the Left SICK LMS-221 Lidar.

Right SICK LMS-221 Lidar

	x (m)	**y** (m)	***θ*** (deg.)
Ground-Truth	-0.3225	-0.8045	-90.33
0L	-0.3299	-0.8080	-90.37
2L	-0.3045	-0.8034	-90.33
6L	-0.2900	-0.8011	-90.18
Average Value	-0.30813	-0.80413	-90.29337
stdv.	0.0164806	0.0028606	0.0800261

Table 5.8: Our estimations of the (x, y, θ) parameters for the Right SICK LMS-221 Lidar.

Comparing the Tables 5.5, 5.6, 5.7, 5.8 with the 5.1, 5.2, 5.3, 5.4 in the previous subsection, we can notice that the estimations of the x, y and θ parameters have much lower standard deviations and the average values much closer to the ground truths. This behaviour is caused by the strategy we adopted, indeed the planar parameters are estimated using the motion-based approach who exploits the ego-motions that in our experiment are ground truth sensors trajectories.
On the other side, the z, ψ and ϕ parameters are computed with the ground observation-based calibration who requires ground points that are estimated for each sensor. Therefore, the inputs of the first procedure are more accurate and precise than the ones of the second, which implies better results in the estimation of the planar parameters.

6 | Conclusions

This thesis has presented a calibration process to estimate the 6 DoF extrinsic parameters for multiple sensors placed in a mobile system.
Specifically, we first described the calibration step to estimate the planar parameters for two coplanar sensors, exploiting their ego motions. A peculiarity of this approach is that inside this step, we estimate also the scale ambiguity, characteristic of the monocular cameras, so that these sensors can be handled with the others in the same optimization framework. Then, we explained a procedure to compute the remaining three parameters, with the ground points estimations from each sensor as input, in order to find its 3D transformation relative to that plane. This transformation is required to project the sensors' motions into a common plane and make the sensors coplanar. Both the calibrations steps are formulated as a least-squares problem and solved in close form, using the Lagrangian multipliers. This kind of solution allows us to avoid initial guesses to find the parameters. Subsequently, we considered all pairs of sensors in a joint least-squares framework, extending the case of only two sensors. At last, we have tested the proposed approach to calibrate two lidars and two cameras, whose measurements come from a collection of datasets generated in an outdoor environment.
The generated results confirm the validity of the proposed approach, especially in the calibration step that estimates the planar extrinsic parameters. In fact, the numerical solutions lie very close to their ground truth values with a variability, between datasets, very narrow. On the other side, the 3D parameters calibration step, based on ground observations, exhibits some inaccuracies and imperfections. The estimation of the environment points could be the main source of this aspect.
Indeed, future works on this subject could focus exactly on the improvement of the environment reconstruction, since this is a fundamental prerequisite for the estimation of the ground plane. An increase in the precision for this reconstruction can surely lead to better results for the 3D extrinsic parameters.
Additionally, since this calibration approach is limited by the requirement of the sensors' incremental motions, some devices such as the radars are not comprehended. Another possible development could be the addition of a new step to this process where the li-

dars' scans are exploited to calibrate the radars' extrinsic parameters too, as described by Heng[10].

Bibliography

[1] S. Agarwal, K. Mierle, and T. C. S. Team. Ceres Solver, 3 2022. URL `https://github.com/ceres-solver/ceres-solver`.

[2] P. An, T. Ma, K. Yu, B. Fang, J. Zhang, W. Fu, and J. Ma. Geometric calibration for lidar-camera system fusing 3d-2d and 3d-3d point correspondences. *Optics express*, 28(2):2122–2141, 2020.

[3] J.-L. Blanco, F.-A. Moreno, and J. González. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27(4):327–351, November 2009. ISSN 0929-5593. doi: 10.1007/s10514-009-9138-7. URL `http://www.mrpt.org/Paper:Malaga_Dataset_2009`.

[4] J. Brookshire and S. Teller. Automatic calibration of multiple coplanar sensors. *Robotics: Science and Systems VII*, 33, 2012.

[5] J. Brookshire and S. Teller. Extrinsic calibration from per-sensor egomotion. *Robotics: Science and Systems VIII*, pages 504–512, 2013.

[6] A. Censi, A. Franchi, L. Marchionni, and G. Oriolo. Simultaneous calibration of odometry and sensor parameters for mobile robots. *IEEE Transactions on Robotics*, 29(2):475–492, 2013.

[7] R. Gomez-Ojeda, J. Briales, E. Fernandez-Moral, and J. Gonzalez-Jimenez. Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3611–3616. IEEE, 2015.

[8] X. Gong, Y. Lin, and J. Liu. 3d lidar-camera extrinsic calibration using an arbitrary trihedron. *Sensors*, 13(2):1902–1918, 2013.

[9] C. X. Guo, F. M. Mirzaei, and S. I. Roumeliotis. An analytical least-squares solution to the odometer-camera extrinsic calibration problem. In *2012 IEEE International Conference on Robotics and Automation*, pages 3962–3968. IEEE, 2012.

[10] L. Heng. Automatic targetless extrinsic calibration of multiple 3d lidars and radars. In

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 10669–10675. IEEE, 2020.

[11] L. Heng, B. Li, and M. Pollefeys. Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1793–1800. IEEE, 2013.

[12] R. Hermann and A. Krener. Nonlinear controllability and observability. *IEEE Transactions on automatic control*, 22(5):728–740, 1977.

[13] K. Huang and C. Stachniss. Extrinsic multi-sensor calibration for mobile robots using the gauss-helmert model. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1490–1496. IEEE, 2017.

[14] M. Jaimez, J. Monroy, M. Lopez-Antequera, and J. Gonzalez-Jimenez. Robust planar odometry based on symmetric range flow and multiscan alignment. *IEEE Transactions on Robotics*, 34(6):1623–1635, 2018.

[15] J. Jeong, Y. Cho, and A. Kim. The road is enough! extrinsic calibration of non-overlapping stereo camera and lidar using road information. *IEEE Robotics and Automation Letters*, 4(3):2831–2838, 2019.

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

[17] S. Mishra, G. Pandey, and S. Saripalli. Extrinsic calibration of a 3d-lidar and a camera. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1765–1770. IEEE, 2020.

[18] P. Moghadam, M. Bosse, and R. Zlot. Line-based extrinsic calibration of range and image sensors. In *2013 IEEE International Conference on Robotics and Automation*, pages 3685–3691. IEEE, 2013.

[19] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[20] J. Peršić. Calibration of heterogeneous sensor systems. *arXiv preprint arXiv:1812.11445*, 2018.

[21] J. Peršić, I. Marković, and I. Petrović. Extrinsic 6dof calibration of a radar–lidar–

camera system enhanced by radar cross section estimates evaluation. *Robotics and Autonomous Systems*, 114:217–230, 2019.

[22] R. Raguram, J.-M. Frahm, and M. Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *European conference on computer vision*, pages 500–513. Springer, 2008.

[23] S. Schneider, T. Luettel, and H.-J. Wuensche. Odometry-based online extrinsic sensor calibration. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1287–1292. IEEE, 2013.

[24] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[26] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[27] M. Velas, M. Španěl, Z. Materna, and A. Herout. Calibration of rgb camera with velodyne lidar. 2014.

[28] J. Zienkiewicz and A. Davison. Extrinsics autocalibration for dense planar visual odometry. *Journal of Field Robotics*, 32(5):803–825, 2015.

[29] D. Zuñiga-Noël, J.-R. Ruiz-Sarmiento, R. Gomez-Ojeda, and J. Gonzalez-Jimenez. Automatic multi-sensor extrinsic calibration for mobile robots. *IEEE Robotics and Automation Letters*, 4(3):2862–2869, 2019.

List of Figures

List of Tables

Acknowledgements

A conclusione di questo lungo percorso, desidero menzionare tutte quelle persone che mi hanno aiutato e supportato per tutto questo tempo e che, grazie alle quali, sono riuscito ad arrivare fin qui.

In primis, un sentito rigraziamento va al mio relatore, Prof. Matteo Matteucci, per la pazienza, per la disponibilità e per la sua fondamentale guida, con la quale sono riuscito a portare a termine lo sviluppo di questo progetto. Al Dott. Simone Mentasti, correlatore della tesi, per il supporto costante e per gli essenziali consigli dispensatemi nelle fasi più delicate del lavoro.

Un ringraziamento speciale a mia madre, mio padre e mia sorella: è grazie a loro inamovibile sostegno e al loro costante incoraggiamento se oggi sono riuscito a raggiungere questo traguardo.
A Nonna Luciana, Nonno Franco, lo Zio Marco, la Zia Chiara, lo Zio Angelo, Lorenzo ed Elenuccia, per esserci sempre stati in tutti questi anni, ma anche a Nonna Nara e Osvaldo, per tutto ciò che mi hanno lasciato.

Alla mia ragazza, Camilla, con la quale ho condiviso questi ultimi anni. Grazie per non avermi mai fatto mancare sostegno e conforto, soprattutto nei momenti più difficili e per essermi sempre stato accanto con straordinario amore e pazienza.

Al Bena, Mutti Abix, Bindo, per essere stati sempre presenti anche durante questa ultima fase del mio percorso di studi, per aver condiviso momenti ed esperienze in tutto questo periodo.
A Bianca, per aver sempre e comunque mantenuto salda la nostra amicizia, così importante, anche a distanza.
Infine, ad Amedeo, con il quale ho iniziato questo viaggio oramai quasi sette anni fa.