



Automatic 2-D LiDAR geometric calibration of installation bias



Bruno J. Guerreiro^{a,*}, Carlos Silvestre^{a,b}, Paulo Oliveira^a

^a Institute for Systems and Robotics, Instituto Superior Técnico, University of Lisbon, Portugal

^b Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Taipa, Macau

HIGHLIGHTS

- Automatic calibration algorithms for 2-D LiDARs on board aerial vehicles.
- No a priori knowledge of the trajectories or the terrain are necessary.
- Use of geometric optimization techniques on $SO(3)$.
- Extensive characterization of the proposed methods using simulated data.
- Validation of the proposed methods using experimental data.

ARTICLE INFO

Article history:

Received 14 June 2013

Received in revised form

21 April 2014

Accepted 5 May 2014

Available online 13 May 2014

Keywords:

Calibration

Range sensing

Special orthogonal group

Mapping

Aerial robotics

ABSTRACT

This paper proposes two estimation algorithms for the determination of the attitude installation matrix for 2-D light detection and ranging (LiDAR) systems on board unmanned aerial vehicles (UAVs). While a comparative calibration algorithm assumes the existence of a known calibration surface, an automatic calibration algorithm does not require any prior knowledge of the trajectories of the vehicle or the terrain where the calibration mission is performed. The proposed calibration algorithms rely on the minimization of the errors between the measured point cloud and a representation of the known calibration surface or, alternatively, the errors between several acquired point clouds, by comparing each measured point cloud with a surface representation of the others. The resulting optimization problems are addressed using two techniques: (i) nonlinear optimization, where the attitude installation rotation matrix is parameterized by the ZYX Euler angles, and (ii) optimization on Riemannian manifolds, enabling the estimation of the attitude installation matrix on the group of special orthogonal matrices $\mathcal{SO}(3)$. The proposed calibration techniques are extensively validated and compared using both simulated and experimental LiDAR data sets, demonstrating their accuracy and performance.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Light detection and ranging (LiDAR) technology is nowadays widely used in the industry as well as by the robotics and the remote sensing research communities. The development of airborne laser ranging sensors started in the 1970s in North America, mainly for topographic applications. Later, with the development of affordable inertial navigation systems (INS) and global positioning system (GPS) units, other applications captured the attention of the research community, such as monitoring ice sheets [1] or measuring canopy heights [2]. The robotics research community is

nowadays employing autonomous vehicles equipped with LiDARs to perform automatic acquisition and 3-dimensional (3-D) reconstruction of terrain, buildings, large infrastructures, and to obtain semantic descriptions of complex environments [3,4], using this information to safely and accurately navigate through unknown environments [5,6]. Data accuracy is essential for all these applications, as there are several sources of inaccuracy that can lead to considerable nonlinear reconstruction errors.

The calibration of 2-D LiDARs on board an autonomous vehicle capable of complex 3-D motion, is one of the most challenging problems in the extrinsic calibration of LiDAR sensors. There is comprehensive work in the literature that is dedicated to the analysis of the intrinsic and extrinsic LiDAR error sources, such as in [7,8] and references therein. Namely, the identified error sources include the attitude and position installation biases, range detection errors, scanning angle errors, vehicle attitude and position errors, time synchronization errors, etc. While most intrinsic

* Corresponding author. Tel.: +351 218418090.

E-mail addresses: bguerreiro@isr.ist.utl.pt (B.J. Guerreiro), cjs@isr.ist.utl.pt (C. Silvestre), pjcro@isr.ist.utl.pt (P. Oliveira).

errors can be identified and accounted for with laboratory experiments using the sensor, the installation biases are highly dependent on the vehicle and mounting apparatus. It is argued by these authors that the attitude installation bias, also simply referred to as the mounting bias, is a particularly important source of error in LiDAR systems. For instance, an airborne LiDAR acquiring terrain elevation 300 m above the ground, with 1° of roll mounting bias will generate points with more than 5 m of error. Moreover, when working with rough terrain with high slopes the distortions can become highly nonlinear, which further motivates the necessity of the calibration of these errors, in particular, the attitude installation bias, to meet the desired accuracy requirements.

1.1. Relevant work

Most applications requiring a three-dimensional (3-D) reconstruction of the surrounding environment use one or several LiDARs installed on board a vehicle equipped with an INS/GPS unit, which provide measurements of the relative distances to the terrain and the trajectory of the vehicle, respectively. To obtain consistent and accurate 3-D maps it is possible to formulate a problem that makes use of some characteristics of the environment to adjust potential errors in the obtained point cloud and vehicle trajectory, as in [9,10], or address the calibration of LiDAR sensors as a separate problem prior to the intended 3-D environment data acquisition.

The most common calibration procedures require particular terrain features and specific vehicle trajectories in order to calibrate a subset of the parameters, as found in [11]. For instance, a standard procedure in the literature for airborne LiDAR calibration would be to fly over a known flat surface while performing pitch or roll maneuvers, separately, which would enable the calibration of only these two parameters. Nonetheless, several difficulties render the problem of 2-D LiDAR attitude installation calibration special, including the absence of any matching information between the measured point clouds and a known calibration surface, and also the fact that, with each calibration correction step, the reconstructed clouds of points will change their shape in a nonlinear fashion, accordingly to the vehicle trajectory and the terrain.

There are two fundamental approaches in the literature to compare two clouds of points when there is no matching information between them. The first approach is to use a point-to-point metric, as in the by-now classic iterative closest point (ICP) algorithm [12], and assign to each point of one cloud a matching point of the other cloud. An alternative approach is to use a point-to-plane metric and, thus, to measure the closest distance between each point of one cloud to a surface approximation of the other cloud [13]. As the surface information is not taken into account in point-to-point based techniques, they suffer from the inability to slide overlapping clouds to find a better fit between them, demonstrating slower convergence rates than the point-to-plane alternatives [14].

To reduce the conservativeness of the calibration procedures of 2-D LiDAR sensors, new algorithms were proposed in [15,16], where the measured point cloud is compared with a plane-wise representation of the known calibration surface to obtain the calibration parameters. These authors consider the linearization of the error model to obtain a Gauss–Helmert model and then obtain the least-squares estimate of the installation bias. This approach was refined in [17] by considering the full nonlinear model of the reconstruction error and, consequently, resorting to nonlinear optimization techniques to obtain the optimal calibration parameters. More recently, specially with the use of expensive 3-D LiDARs installed on ground vehicles, several strategies have been proposed, such as those presented in [18,19], for LiDAR calibration that rely on the detection and association of artificial features in the point clouds. Nonetheless, the major drawback of these techniques is that they

make strong assumptions about the environment where the calibration procedure takes place, either assuming complete knowledge of the calibration surface or adding artificial marks to enable feature based association.

An alternative for the pose calibration of sensors is the joint formulation of the calibration, localization, and mapping problems into a single problem. In this approach, the typical state of a simultaneous localization and mapping problem (SLAM), such as [6,20,21], is augmented to include the intrinsic or extrinsic variables to be calibrated. For instance, in [22] the authors proposed a method rooted on graph-based SLAM to obtain the robot position in the mapped environment, the LiDAR position on the robot, as well as the kinematic variables of the robot. For the case of camera calibration, a method based on the unscented Kalman filter (UKF) SLAM is proposed in [23] for obtaining the map of the environment, the pose of the vehicle, and the pose of the camera relative to the vehicle.

Within the field of mobile robotics, the calibration of 3-D LiDARs on board ground vehicles has been addressed in [24,25]. The former uses a point-to-plane metric to define a cost function, based on Euler angles for the attitude installation bias, and a heuristic search in several directions to avoid local minima. The latter resorts to an entropy-based point cloud quality metric, which allows for the calibration of a 3-D LiDAR sensor using several overlapping 3-D scans, and based on this calibrated point cloud, any 2-D LiDAR on board the vehicle can also be calibrated. However, noting that both these methods rely on the use of 3-D LiDAR data, the proposed calibration methods would not be appropriate if only one statically mounted 2-D LiDAR was used for data acquisition. Furthermore, using an entropy-based point cloud quality metric might not be the best approach to use all the available information (see Appendix A) provided by the type of point clouds obtained with airborne 2-D LiDARs, as two uncalibrated point clouds might only be close to each other within a small boundary close to their intersection (as can be seen in Fig. 12(a)).

1.2. Proposed approach

The algorithms proposed in this work can cope with arbitrary vehicle 3-D motion and terrain topology, yielding accurate calibration of airborne 2-D LiDAR sensors. It is assumed that there is enough information on the calibration terrain and vehicle trajectories to enable the full attitude installation bias calibration. One obvious counter-example would be a flat terrain, thus, with no relevant topological features, for which there would always be one uncalibrated degree of freedom. The proposed methods use a point-to-plane metric to compare each acquired point cloud with either a known calibration surface or a surface approximation of all the remaining acquired point clouds, by measuring the minimum distance of each point to the approximated surface. A fully nonlinear model of the calibration errors is also defined, either using Euler angles or rotation matrices, thus avoiding the linearization problems of many works in the existing literature. In addition, by using rotation matrix representation of the attitude installation bias, optimization tools on the special orthogonal group can be used.

Two different approaches are considered for the calibration procedures addressed in this paper: the (i) *Comparative Calibration*, which considers arbitrary vehicle trajectories and assumes the existence of a known calibration surface that will be compared with the measured data during the calibration algorithm; and the (ii) *Automatic Calibration*, for which no a priori information on the terrain or trajectory is required and, at least, two measured data sets of the same terrain obtained from different vehicle trajectories are necessary. The former approach was introduced by the authors in [17] and is included in this paper for comparison purposes, as it represents the classical approach to this calibration problem. Regarding the automatic calibration, where a known calibration surface

is not required, several acquired overlapping point clouds of the same portion of terrain are compared between each other, while the vehicle performs arbitrary maneuvers. The fundamental idea supporting the automatic calibration algorithm is that, in an ideal noise-free world, only one attitude installation matrix should exist that fully justifies all the acquired point clouds, in which case the cost function would achieve its minimum value.

The estimation problem is formulated within the scope of maximum likelihood (ML) theory [26], allowing the formulation of the calibration problem as an optimization problem. The cost function to be minimized is defined by the summation of the errors between each measured data set and either the known calibration surface or a surface approximation of the other sets yielding a weighted scalar cost function. When using the ZYX Euler angles parameterization of the attitude installation bias rotation matrix, which is the standard approach in the literature, basic concepts of nonlinear optimization can be used. For completeness, this paper includes the use of the Gradient and Newton methods to find a search direction, as well as a line search algorithm, in particular the Wolfe rule, to compute the step size [27,28]. To take advantage of the use of the rotation matrix parameterization, optimization tools on Riemannian manifolds are devised, enabling the use of the Gradient and Newton methods to directly estimate the attitude installation matrix on the group of special orthogonal matrices $\mathcal{SO}(3)$, [29,30]. This latter optimization approach can also use a modified version of the Wolfe rule to compute the step size along a geodesic of the manifold. Nonetheless, there exists an exact computation of the step size within $\mathcal{SO}(3)$, as detailed in [31], which is shown to improve the performance of the algorithm. The proposed calibration techniques are compared using simulated and experimental data sets to assess their performance and limitations in realistic scenarios.

The main contributions of this work are: (i) the formulation of the attitude installation bias comparative 2-D LiDAR calibration problem as a nonlinear optimization problem which is not bound to any specific trajectory or terrain shape; (ii) the formulation of a novel 2-D LiDAR automatic calibration problem for vehicles capable of 3-D motion, for which no a priori knowledge on the terrain or trajectories used in the calibration procedure is assumed; (iii) the solution of the optimization problem resorting to a Riemannian optimization framework to solve the problem within the group of proper rotation matrices; (iv) the application of an exact line search algorithm to find the optimal step size for the Riemannian optimization calibration algorithm; (v) the analysis and comparison of the proposed algorithms for several optimization techniques using realistic simulations; and (vi) the validation of the algorithms using experimental field data using an unmanned aerial vehicle (UAV). Building on the comparative calibration results presented in [17], this paper extends and consolidates the results provided in a preliminary version presented in [32], which introduced the automatic calibration algorithm. Besides the significant extension of the simulation results, this paper also presents the validation of the automatic calibration algorithm using experimental data acquired on board an unmanned helicopter.

The paper is organized as follows. Section 2 introduces the 2-D LiDAR calibration problems. In Section 3 the optimization problem is addressed using the ZYX Euler angles parameterization, whereas Section 4 introduces the Riemannian optimization framework used to address the same calibration problems. The performance analysis of the proposed calibration algorithms as well as a comparison of the several optimization techniques are detailed in the simulation results presented in Section 5. In addition, the experimental results presented in Section 6 provide further validation of the automatic calibration approach. Finally, in Section 7 the main conclusions and directions for further research are offered.

2. Problem definition

This paper addresses the problem of finding the best possible estimate of the attitude installation bias of a 2-D LiDAR sensor mounted on board a UAV capable of complex 3-D motion. In this section the comparative and the automatic calibration problems are defined. Firstly, the 3-D point reconstruction model is introduced followed by the definition of a point-to-plane metric, which is then used to introduce the cost functions for the comparative and automatic calibration problems. Finally, the comparative and the automatic calibration algorithms are unified by rewriting them as particular cases of a common optimization problem, so that the same optimization tools can be easily applied.

2.1. 3-D point reconstruction

The point reconstruction model describes the computations required to obtain 3-D points by combining the vehicle trajectories and the LiDAR raw data. Assume that an earth-fixed reference frame $\{E\}$ and a body-fixed frame $\{B\}$ are appropriately defined. In addition, the following coordinate frames are required:

- the LiDAR frame $\{L\}$, with origin at the laser's firing point, z -axis indicating the opposite direction of the zero scanning angle, and the y -axis perpendicular to the scanning plane; and
- the laser beam frame $\{LB\}$, with origin at the laser firing point, y -axis collinear with that of frame $\{L\}$ and z -axis oriented opposite to the direction of the laser beam.

These frames and the relationships between them are depicted in Fig. 1. A measurement l is defined by the pair (ρ_l, α_l) , denoting the distance between the laser firing point and the laser hit point as ρ_l and the angular position of the laser beam as α_l , which stands for the angle from $\{L\}$ to $\{LB\}$. Let the rotation from $\{LB\}$ to $\{L\}$ be defined as ${}^L_{LB}\mathbf{R}(\alpha_l) = \mathbf{R}_y(\alpha_l)$, with $\mathbf{R}_y(\cdot)$ standing for the rotation matrix about the y axis, and the laser range measurement as $\mathbf{r}(\rho_l) = [0 \ 0 \ -\rho_l]^T$. Then, the position measurement described in $\{L\}$, denoted by \mathbf{p}_l , can be defined as $\mathbf{p}_l = {}^L_{LB}\mathbf{R}(\alpha_l) \mathbf{r}(\rho_l)$. Therefore, it can be seen that the expression that transforms the LiDAR measurement into the reconstructed 3-D point ${}^E\mathbf{p}_l$, expressed in the earth-fixed frame, is given by

$${}^E\mathbf{p}_l = {}^E_B\mathbf{R}_l(\mathbf{R}\mathbf{p}_l + \mathbf{b}) + {}^E\mathbf{p}_{B_l}, \quad (1)$$

where ${}^E_B\mathbf{R}_l$ is the platform attitude defined by the rotation from $\{B\}$ to $\{E\}$, at the time that measurement l is acquired, and ${}^E\mathbf{p}_{B_l}$ is the INS/GPS unit position expressed in $\{E\}$. The position installation bias is denoted by $\mathbf{b} \in \mathbb{R}^3$, describing the laser firing point expressed in $\{B\}$, whereas \mathbf{R} denotes the rotation matrix that defines the attitude installation bias, which is an element of the group of special orthogonal matrices $\mathcal{SO}(3)$. Let $\mathcal{M}(n, \mathbb{R}) = \{\mathbf{A} : n \times n \text{ matrix with real entries}\}$ and the group of orthogonal matrices be defined as $\mathcal{O}(n) = \{\mathbf{U} \in \mathcal{M}(n, \mathbb{R}) : \mathbf{U}\mathbf{U}^T = \mathbf{I}_n\}$, then, the group of special orthogonal matrices is $\mathcal{SO}(n) = \{\mathbf{R} \in \mathcal{O}(n) : |\mathbf{R}| = 1\}$, where the operator $|\cdot|$ denotes the determinant of the argument matrix.

In this paper, only the attitude installation bias is considered for calibration, noting that the position installation bias, \mathbf{b} , can be more accurately measured and its influence in the reconstruction error is not as significant as that of the attitude installation bias. Nonetheless, the optimization algorithms proposed in the next sections can be used to calibrate simultaneously the attitude and the position installation biases with simple changes to the search direction and step size computations.

2.2. Cloud-to-surface comparison

Due to the nature of the available measurements, it should be emphasized that the correspondence between the points of the

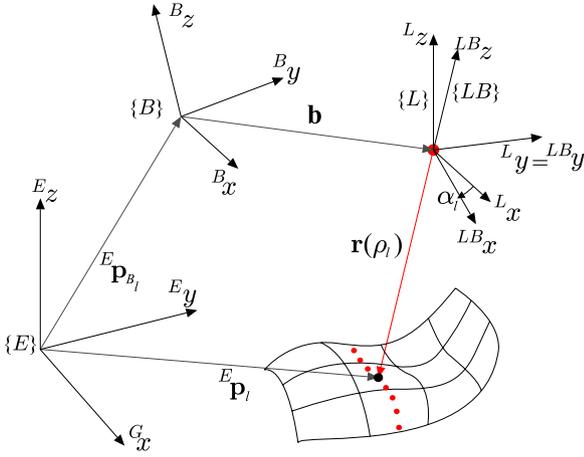


Fig. 1. LiDAR acquisition coordinate frames.

different clouds of data is not directly available for the calibration problems tackled in this paper. Thus, the use of techniques that are based on the a priori knowledge of the matching between points of different data sets is not an option. Assume there are n_c clouds of measured points, \mathcal{C}_i for $i = 1, \dots, n_c$, resulting from different trajectories of the vehicle, such that the intersection of the laser data boundaries of each cloud projected onto the other clouds is not empty. Each cloud is a set of n_{p_i} laser reconstructed points, $\mathcal{C}_i = \{^E \mathbf{p}_{i_l}\}$ for $l = 1, \dots, n_{p_i}$, which consists of a nonlinearly distorted sampling with noise of the actual surface \mathcal{S} . Each cloud \mathcal{C}_i can also be approximated by a piecewise surface \mathcal{S}_i , consisting of a set of n_{s_i} planes. Moreover, the known calibration surface approximation used in the comparative calibration algorithm is denoted as \mathcal{S}_0 and is also a set of n_{s_0} planes.

The comparative calibration, which assumes that a known calibration surface is available (see Appendix A), compares each cloud \mathcal{C}_i , for all $i = 1, \dots, n_c$, with the calibration surface \mathcal{S}_0 . Conversely, the proposed solution for the automatic calibration compares each cloud \mathcal{C}_i , for all $i = 1, \dots, n_c$, with a surface approximation of all the remaining clouds, \mathcal{S}_j , for all $j = 1, \dots, n_c$ and $j \neq i$. To compare the points of a cloud \mathcal{C}_i with an approximated surface \mathcal{S}_j , each point is linked to the closest plane of \mathcal{S}_j along its normal vector, thus, defining a point-to-plane metric. Considering the set of planes that define \mathcal{S}_j , let the plane associated with the measurement l of \mathcal{C}_i be defined by the set of points $\mathbf{p} \in \mathbb{R}^3$ that verify the equation of the plane $\mathbf{p}^T \mathbf{n}_{ij_l} + d_{ij_l} = 0$, where $\mathbf{n}_{ij_l} = [a_{ij_l} \ b_{ij_l} \ c_{ij_l}]^T \in \mathbb{R}^3$ is the plane normal unit vector and $d_{ij_l} \in \mathbb{R}$ is the distance to the origin. Thus, the point-to-plane distance metric, depicted in Fig. 2, is represented by the error between each point $^E \mathbf{p}_{i_l}$ and the associated plane of the surface. This distance is defined as $\mathbf{e}_{ij_l} = ^E \bar{\mathbf{p}}_{ij_l} - ^E \mathbf{p}_{i_l}$, where $^E \bar{\mathbf{p}}_{ij_l} = ^E \mathbf{p}_{i_l} - \mathbf{n}_{ij_l} D_{ij_l}$ denotes the point in the associated plane closest to $^E \mathbf{p}_{i_l}$ (naturally along the normal to the plane, \mathbf{n}_{ij_l}), and the distance along the normal vector is defined as $D_{ij_l} = ^E \mathbf{p}_{i_l}^T \mathbf{n}_{ij_l} + d_{ij_l}$.

2.3. Calibration problem

Using the reconstruction model (1) and after some algebraic manipulation, the measurement error introduced above can be written as

$$\mathbf{e}_{ij_l}(\mathbf{R}) = \mathbf{H}_{ij_l} \mathbf{R} \mathbf{p}_i + \mathbf{c}_{ij_l}, \quad (2)$$

where

$$\mathbf{H}_{ij_l} = -\mathbf{n}_{ij_l} \mathbf{n}_{ij_l}^T {}^E \mathbf{R}_i,$$

$$\mathbf{c}_{ij_l} = \mathbf{H}_{ij_l} \mathbf{b} - \mathbf{n}_{ij_l} \mathbf{n}_{ij_l}^T {}^E \mathbf{p}_{B_{i_l}} - d_{ij_l} \mathbf{n}_{ij_l},$$

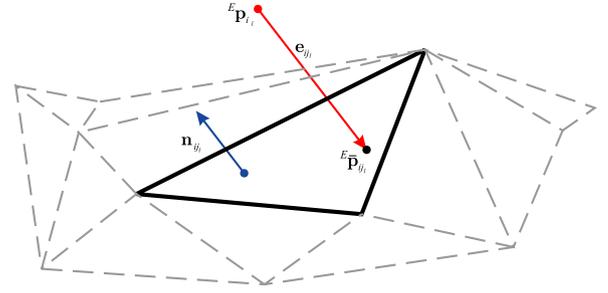


Fig. 2. Point-to-plane distance metric.

and $\mathbf{R} \in \mathcal{SO}(3)$ is the calibration parameter matrix to be estimated, whereas $\mathbf{H}_{ij_l} \in \mathcal{M}(3, \mathbb{R})$, $\mathbf{p}_i \in \mathbb{R}^3$, and $\mathbf{c}_{ij_l} \in \mathbb{R}^3$ are known parameters. Note that this error, which will be used to define the optimization problem, is fundamentally different from that of the well known Procrustes problem, see [33,34] and references therein, as the matrix \mathbf{H}_{ij_l} depends on the specific measurement l and the association between clouds i and j .

Assuming that the probability distribution of the measurement error is a Gaussian and defined by $p(\mathbf{e}_{ij_l}) = \mathcal{N}(\mathbf{0}, \sigma_{ij_l}^2 \mathbf{I}_3)$, the calibration problems can be formulated within the scope of the maximum likelihood estimation theory as that of maximizing the reconstruction error probability function, or equivalently, minimizing a log-likelihood function $f(\mathbf{R})$, i.e.,

$$\mathbf{R}^* = \arg \min_{\mathbf{R} \in \mathcal{SO}(3)} f(\mathbf{R}), \quad (3)$$

noting that the log-likelihood function $f(\mathbf{R})$ will be defined below. The comparative calibration problem can be formulated considering the errors between each measured point clouds, \mathcal{C}_i , and the known calibration surface approximation, \mathcal{S}_0 , yielding the log-likelihood function

$$f_{comp}(\mathbf{R}) = \sum_{i=1}^{n_c} \sum_{l=1}^{n_{p_i}} \frac{1}{2 n_c n_{p_i} \sigma_{i0_l}^2} \|\mathbf{e}_{i0_l}(\mathbf{R})\|^2. \quad (4)$$

On the other hand, the automatic calibration problem, where no known calibration surface is available, considers all the errors between each measured set of reconstructed points, \mathcal{C}_i , and the surface approximation of the remaining clouds, \mathcal{S}_j , for all $j = 1, \dots, n_c$ and $j \neq i$, yielding the log-likelihood function

$$f_{auto}(\mathbf{R}) = \sum_{i=1}^{n_c} \sum_{\substack{j=1 \\ j \neq i}}^{n_c} \sum_{l=1}^{n_{p_i}} \frac{1}{2 n_c (n_c - 1) n_{p_i} \sigma_{ij_l}^2} \|\mathbf{e}_{ij_l}(\mathbf{R})\|^2. \quad (5)$$

Both these cost functions can be further simplified, by mapping the successive summations into only one, and a new unified cost function can be defined by considering the errors for each combination of cloud-to-surface comparisons, yielding

$$f(\mathbf{R}) = \sum_{i=1}^n \frac{1}{2 n \sigma_i^2} \|\mathbf{e}_i(\mathbf{R})\|^2, \quad (6)$$

where $\mathbf{e}_i(\mathbf{R}) := \mathbf{H}_i \mathbf{R} \mathbf{p}_i + \mathbf{c}_i$ and σ_i are obtained from the previously defined variables $\mathbf{e}_{kjl}(\mathbf{R})$ and σ_{kjl} , for all $k = 1, \dots, n_c$, $j = 0, \dots, n_c$, $j \neq k$, and $l = 1, \dots, n_{p_i}$, considering the total number of point-to-plane comparisons, n , and using a simple index map to obtain the global index, $i = 1, \dots, n$. The uncertainty parameters σ_i can be used to model different error behaviors, for instance, increasing its value according to the range distance of each point or accounting for the differences in accuracy of several laser sensors.

It is fairly simple to obtain the complexity of computing both cost functions, (4) and (5), regarding the number of clouds, n_c , as

well as the maximum number of points per cloud and the maximum number of planes per approximation surface, respectively defined as

$$n_p = \max(n_{p_1}, \dots, n_{p_{n_c}}), \quad (7)$$

$$n_s = \max(n_{s_0}, n_{s_1}, \dots, n_{s_{n_c}}), \quad (8)$$

where n_{p_i} and n_{s_j} are defined above. To compute the error between a measured point of cloud i and the surface approximation of cloud j , $\mathbf{e}_{ij}(\mathbf{R})$, it is necessary to find the corresponding plane of the approximation surface \mathcal{S}_j , which has at most n_s planes. Thus, the complexity of computing the comparative calibration cost function can be shown to be $O(n_c n_p n_s)$, whereas the complexity of the automatic calibration cost function is given by $O(n_c^2 n_p n_s)$.

3. Optimization using Euler angles parameterization

In this section the optimization problem is reformulated to estimate an Euler angles vector, λ , instead of the rotation matrix \mathbf{R} . Let $\lambda \in \mathbb{R}^3$ be the ZYX Euler angles parameterization of the rotation matrix, i.e. $\mathbf{R} = \mathbf{R}(\lambda)$, then, the optimization problem (3) can be written as

$$\lambda^* = \arg \min_{\lambda} f(\lambda)$$

with $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ given by

$$f(\lambda) = \sum_{i=1}^n \frac{1}{2 n \sigma_i^2} \|\mathbf{H}_i \mathbf{R}(\lambda) \mathbf{p}_i + \mathbf{c}_i\|^2.$$

As described in Algorithm 1, the optimal calibration parameters, denoted by λ^* , are computed using the Gradient and Newton methods, that at each iteration k provide a descent direction, $\mathbf{d}_k \in \mathbb{R}^3$. This descent direction is used to update the next estimate λ_{k+1} by solving a minimization subproblem, usually called line search, to find the optimal step size along the direction \mathbf{d}_k .

Algorithm 1 Minimization using Euler angles

Initialize $k = 0$ and λ_k

repeat

 Compute descent direction \mathbf{d}_k

 Compute the step size by solving the minimization subproblem

$$t_k^* = \arg \min_{t \geq 0} f(\lambda_k + t \mathbf{d}_k);$$

 Compute next parameter estimate:

$$\lambda_{k+1} = \lambda_k + t_k^* \mathbf{d}_k;$$

until $\|\nabla f|_{\lambda_{k+1}}\| < \epsilon$

return $\lambda^* \leftarrow \lambda_{k+1}$ as the optimal parameter

3.1. Descent direction

To compute the descent direction \mathbf{d}_k , the most widely used method is the Gradient method, for which \mathbf{d}_k is computed using the gradient of the cost function, $\nabla f(\lambda)|_{\lambda_k}$. It can be seen that the descent direction for the gradient method is given by

$$\mathbf{d}_k = -\nabla f(\lambda)|_{\lambda_k}$$

where the gradient is computed using

$$\nabla f(\lambda)|_{\lambda} = \left[\frac{\partial f(\lambda)}{\partial \lambda_1} \quad \frac{\partial f(\lambda)}{\partial \lambda_2} \quad \frac{\partial f(\lambda)}{\partial \lambda_3} \right]^T$$

and

$$\frac{\partial f(\lambda)}{\partial \lambda_j} = \sum_{i=1}^n \frac{1}{n \sigma_i^2} \mathbf{e}_i^T \mathbf{H}_i \frac{\partial \mathbf{R}(\lambda)}{\partial \lambda_j} \mathbf{p}_i.$$

The Newton method uses the Hessian matrix, denoted by $\nabla^2 f(\lambda)|_{\lambda_k}$, yielding faster convergence near the optimum value. The descent direction for the Newton method is defined by

$$\mathbf{d}_k = -(\nabla^2 f(\lambda)|_{\lambda_k})^{-1} \nabla f(\lambda)|_{\lambda_k}$$

where the Hessian matrix is given by

$$\nabla^2 f(\lambda)|_{\lambda} = \frac{\partial^2 f(\lambda)}{\partial \lambda \partial \lambda^T} = \left\{ \frac{\partial^2 f}{\partial \lambda_j \partial \lambda_l} \right\}$$

and

$$\begin{aligned} \frac{\partial^2 f}{\partial \lambda_j \partial \lambda_l} &= \sum_{i=1}^n \frac{1}{n \sigma_i^2} \left[\mathbf{e}_i^T \mathbf{H}_i \frac{\partial^2 \mathbf{R}(\lambda)}{\partial \lambda_j \partial \lambda_l} \mathbf{p}_i \right. \\ &\quad \left. + \mathbf{p}_i^T \frac{\partial \mathbf{R}^T(\lambda)}{\partial \lambda_j} \mathbf{H}_i^T \mathbf{H}_i \frac{\partial \mathbf{R}(\lambda)}{\partial \lambda_l} \mathbf{p}_i \right]. \end{aligned}$$

3.2. Line search

The step size optimization subproblem in Algorithm 1 is numerically solved using the Wolfe conditions [28]. Consider the function $\phi: \mathbb{R} \rightarrow \mathbb{R}$ defined by $\phi(t) = f(\lambda_k + t \mathbf{d}_k)$ and derivative given by $\dot{\phi}(t) = \nabla^T f|_{\lambda_k + t \mathbf{d}_k} \mathbf{d}_k$ and let also $\mu(t_k) = \phi(0) + \sigma \dot{\phi}(0) t_k$ and $\mu_0 = \lambda \dot{\phi}(0)$, where σ and λ are parameters of the algorithm. The Wolfe rule classifies the step size according to the sets

$$\begin{aligned} \mathcal{A} &= \{t_k > 0 : \phi(t_k) \leq \mu(t_k) \wedge \dot{\phi}(t_k) \geq \mu_0\} \\ \mathcal{R} &= \{t_k > 0 : \phi(t_k) > \mu(t_k)\} \\ \mathcal{L} &= \{t_k > 0 : \phi(t_k) \leq \mu(t_k) \wedge \dot{\phi}(t_k) < \mu_0\} \end{aligned} \quad (9)$$

that define respectively the acceptable, the right unacceptable and the left unacceptable step sizes as depicted in Fig. 3. As detailed in Algorithm 2, the line search algorithm consists of finding an acceptable step size, i.e., an estimate of the optimal step size.

Algorithm 2 Line search using the Wolfe conditions

Require: Search direction \mathbf{d}_k

Set $i \leftarrow 0$

Initialize $t_i > 0$, $e = 0$, and $d = +\infty$

Compute $\phi(t_i)$ and $\dot{\phi}(t_i)$

while $t_i \notin \mathcal{A}$ **do**

if $t_i \in \mathcal{D}$ **then**

 Set $d = t_i$

else $\{t_i \in \mathcal{E}\}$

 Set $e = t_i$

end if

 Choose $t_{i+1} \in (e, d)$

 Set $i \leftarrow i + 1$

 Compute $\phi(t_i)$ and $\dot{\phi}(t_i)$

end while

return $t^* \leftarrow t_i$ as the optimal step size

4. Optimization on Riemannian manifolds

This section introduces the optimization methodology on Riemannian manifolds. Since the rotation matrix \mathbf{R} is an element of the group of special orthogonal matrices $\mathcal{SO}(3)$, which is an embedded submanifold of $\mathcal{M}(3, \mathbb{R})$, the optimization tools adopted in this section are based on the simple exercises of the Riemannian geometry theory and allow for the minimization of the log-likelihood function directly on the manifold of special orthogonal matrices $\mathcal{SO}(3)$. The concepts of intrinsic gradient and the Hessian

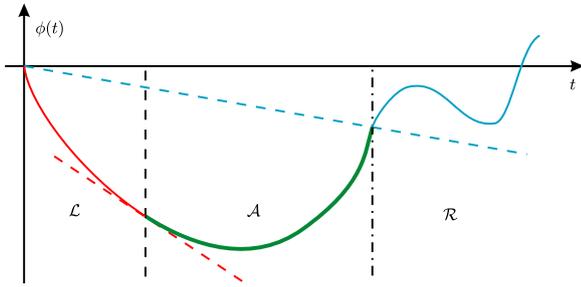


Fig. 3. Wolfe rule.

derived for $\mathcal{SO}(3)$ produce descent search directions in the manifold and the cost function is minimized along geodesics in $\mathcal{SO}(3)$ to obtain an estimate of the optimal step size. For a comprehensive introduction to the subject and for applications with orthogonality constraints the reader is referred to [29,30].

Considering the log-likelihood function $f : \mathcal{SO}(3) \rightarrow \mathbb{R}$ given in (6), the optimization problem reduces to the one defined in (3) and the optimal value \mathbf{R}^* can be computed using the Gradient or the Newton methods generalized to manifolds. As described in Algorithm 3, given the current parameter estimate \mathbf{R}_k , at iteration k , these methods compute a descent direction in the intrinsic tangent space, $\mathbf{d}_k \in T_{\mathbf{R}_k} \mathcal{SO}(3)$, allowing to obtain the new estimate \mathbf{R}_{k+1} by solving a minimization subproblem along the geodesic of the manifold, also denoted as the line search. A geodesic along the manifold is defined as $\boldsymbol{\gamma}_{\mathbf{d}_k}(t) \in \mathcal{SO}(3)$, with initial conditions $\boldsymbol{\gamma}_{\mathbf{d}_k}(0) = \mathbf{R}_k$ and $\dot{\boldsymbol{\gamma}}_{\mathbf{d}_k}(0) = \mathbf{d}_k$. The accuracy of the solution is determined by the constant ϵ and the norm is determined using a metric in the parameter space $\mathcal{SO}(3)$.

Algorithm 3 Minimization on $\mathcal{SO}(3)$

Initialize $k = 0$ and \mathbf{R}_k
repeat
 Compute descent direction \mathbf{d}_k
 Compute the step size by solving the minimization subproblem

$$t_k^* = \arg \min_{t \geq 0} f(\boldsymbol{\gamma}_{\mathbf{d}_k}(t)) ;$$

Compute next parameter estimate:

$$\mathbf{R}_{k+1} = \boldsymbol{\gamma}_{\mathbf{d}_k}(t_k^*) ;$$

until $\|\nabla f|_{\mathbf{R}_{k+1}}\| < \epsilon$
return $\mathbf{R}^* \leftarrow \mathbf{R}_{k+1}$ as the optimal parameter

The definition of the canonical metric for a given rotation matrix $\mathbf{R} \in \mathcal{SO}(3)$ is inherited from the canonical metric in the Euclidean space $\mathcal{M}(3, \mathbb{R})$. While the tangent space of $\mathcal{M}(3, \mathbb{R})$ is identified with $T_{\mathbf{R}} \mathcal{M}(3, \mathbb{R}) \simeq \mathcal{M}(3, \mathbb{R})$ and represented by the usual gradient, the tangent space of $\mathcal{SO}(3)$ at point \mathbf{R} is identified by $T_{\mathbf{R}} \mathcal{SO}(3) \simeq \mathbf{R} \mathcal{K}(3) = \{\mathbf{R} \mathbf{K} : \mathbf{K} \in \mathcal{K}(3)\}$, where $\mathcal{K}(n)$ is the set of $n \times n$ skew-symmetric matrices with real entries. To define the canonical metric in $\mathcal{SO}(3)$, let two tangent vectors $\{\delta_1, \delta_2\} \in T_{\mathbf{R}} \mathcal{SO}(3)$, which are identified with $\delta_1 \simeq \mathbf{R} \mathbf{K}_1$ and $\delta_2 \simeq \mathbf{R} \mathbf{K}_2$, with $\{\mathbf{K}_1, \mathbf{K}_2\} \in \mathcal{K}(3)$, then $\langle \delta_1, \delta_2 \rangle = \text{tr}(\delta_1^T \delta_2)$, where $\text{tr}(\cdot)$ stands for the trace of a matrix.

4.1. Descent direction

To improve the accuracy of the descent direction estimate, this section adopts generalizations for manifolds of the gradient and the Newton methods. The derivation of the gradient and the Hessian of the log-likelihood function are described below specifically for the $\mathcal{SO}(3)$ manifold.

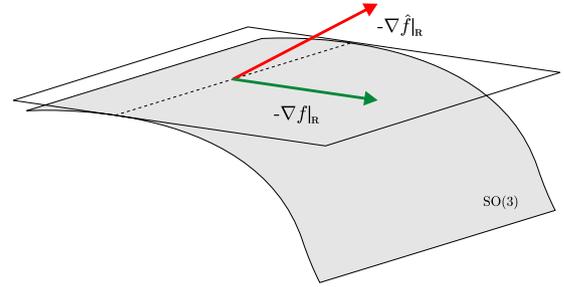


Fig. 4. Projection of the extrinsic gradient.

4.1.1. Gradient method

The log-likelihood function (6) can be generalized to $\mathcal{M}(3, \mathbb{R})$ by defining the smooth function $\hat{f} : \mathcal{M}(3, \mathbb{R}) \rightarrow \mathbb{R}$ such that $\hat{f}|_{\mathcal{SO}(3)} = f$. The tangent space on $\mathcal{M}(3, \mathbb{R})$ is characterized as the direct sum of two tangent spaces complementary to $\mathcal{SO}(3)$, that is

$$T_{\mathbf{R}} \mathcal{M}(3, \mathbb{R}) = T_{\mathbf{R}} \mathcal{SO}(3) \oplus (T_{\mathbf{R}} \mathcal{SO}(3))^{\perp},$$

where the operator \oplus stands for the direct sum of two sets and $(\cdot)^{\perp}$ is the orthogonal complement of its argument. The smooth vector field defined by the extrinsic gradient $\text{grad} \hat{f}|_{\mathbf{R}} \in T_{\mathbf{R}} \mathcal{M}(3, \mathbb{R})$ is decomposed as the sum of its tangent and orthogonal components

$$\text{grad} \hat{f}|_{\mathbf{R}} = (\text{grad} \hat{f}|_{\mathbf{R}})^{\top} + (\text{grad} \hat{f}|_{\mathbf{R}})^{\perp}$$

and is identified with the usual gradient in $\mathcal{M}(3, \mathbb{R})$, that is

$$\text{grad} \hat{f}|_{\mathbf{R}} \simeq \nabla \hat{f}|_{\mathbf{R}} := \frac{\partial f}{\partial \mathbf{R}} := \left[\frac{\partial f}{\partial r_{ij}} \right]_{i,j \in \{1,2,3\}},$$

which, for the proposed calibration cost function, is given by

$$\nabla \hat{f}|_{\mathbf{R}} = \sum_{i=1}^n \frac{1}{n \sigma_i^2} \mathbf{H}_i^T (\mathbf{H}_i \mathbf{R} \mathbf{p}_i + \mathbf{c}_i) \mathbf{p}_i^T.$$

Hence, the intrinsic gradient $\nabla f|_{\mathbf{R}} \in \mathbf{R} \mathcal{K}(3)$ is obtained by the projection of the extrinsic gradient on the tangent space $T_{\mathbf{R}} \mathcal{SO}(3)$, i.e. $\nabla f|_{\mathbf{R}} = (\nabla \hat{f}|_{\mathbf{R}})^{\top}$, as illustrated in Fig. 4. The projection of the extrinsic gradient results from an optimization problem with closed-form solution given by

$$\begin{aligned} \nabla f|_{\mathbf{R}} &= \mathbf{R} \arg \min_{\mathbf{K} \in \mathcal{K}} \|\nabla \hat{f}|_{\mathbf{R}} - \mathbf{R} \mathbf{K}\|^2 \\ &= \mathbf{R} \arg \min_{\mathbf{K} \in \mathcal{K}(3)} \|\text{skew}(\mathbf{R}^T \nabla \hat{f}|_{\mathbf{R}}) - \mathbf{K}\|^2 \\ &= \mathbf{R} \text{skew}(\mathbf{R}^T \nabla \hat{f}|_{\mathbf{R}}), \end{aligned}$$

where $\text{skew}(\mathbf{A}) := 1/2(\mathbf{A} - \mathbf{A}^T)$ is the skew-symmetric component of \mathbf{A} . Thus, at each iteration k the intrinsic gradient direction used in the optimization algorithm is

$$\mathbf{d}_k = -\nabla f|_{\mathbf{R}_k} = -\mathbf{R}_k \text{skew}(\mathbf{R}_k^T \nabla \hat{f}|_{\mathbf{R}_k}).$$

4.1.2. Newton method

The Newton method uses the second order properties of the log-likelihood function to compute the descent direction, resulting in an increased convergence rate near the optimal value when compared to that of the Gradient method. Nonetheless, it should be noted that this algorithm is harder to implement, requires more memory, and may yield non-descent directions, in which case the implementation of the algorithm must use the gradient direction instead.

Given two tangent vectors $\{\mathbf{X}, \mathbf{Y}\} \in T_{\mathbf{R}}\mathcal{S}\mathcal{O}(3)$ and the correspondent extension $\{\hat{\mathbf{X}}, \hat{\mathbf{Y}}\} \in T_{\mathbf{R}}\mathcal{M}(3, \mathbb{R})$, the intrinsic Hessian is given by compensating the external Hessian using

$$\text{Hess}f(\mathbf{X}, \mathbf{Y}) = \text{Hess}\hat{f}(\hat{\mathbf{X}}, \hat{\mathbf{Y}}) + \Pi_{\mathbf{R}}(\mathbf{X}, \mathbf{Y})\hat{f},$$

where the second fundamental form

$$\Pi_{\mathbf{R}} : T_{\mathbf{R}}\mathcal{S}\mathcal{O}(3) \times T_{\mathbf{R}}\mathcal{S}\mathcal{O}(3) \rightarrow (T_{\mathbf{R}}\mathcal{S}\mathcal{O}(3))^{\perp}$$

is a differentiable local vector field on $\mathcal{M}(3, \mathbb{R})$ normal to $\mathcal{S}\mathcal{O}(3)$. The external Hessian is identified by the usual second order derivative in Euclidean spaces

$$\text{Hess}\hat{f}(\hat{\mathbf{X}}, \hat{\mathbf{Y}}) = \text{vec}(\mathbf{X})^T \nabla^2 \hat{f}|_{\mathbf{R}} \text{vec}(\mathbf{Y}),$$

$$\nabla^2 \hat{f}|_{\mathbf{R}} = \frac{\partial^2 \hat{f}}{\partial \text{vec}(\mathbf{R}) \partial \text{vec}(\mathbf{R})^T},$$

where $\hat{\mathbf{X}} \simeq \mathbf{X} \in \mathcal{M}(3, \mathbb{R})$, $\hat{\mathbf{Y}} \simeq \mathbf{Y} \in \mathcal{M}(3, \mathbb{R})$, the $\text{vec}(\cdot)$ operator is the vectorization of a matrix, and it is a matter of algebraic manipulation to see that the extrinsic gradient for the calibration cost function is given by

$$\nabla^2 \hat{f}|_{\mathbf{R}} = \sum_{i=1}^n \frac{1}{n \sigma_i^2} \mathbf{p}_i \mathbf{p}_i^T \otimes \mathbf{H}_i^T \mathbf{H}_i,$$

where \otimes denotes the Kronecker product operator. The second fundamental form applied to \hat{f} yields

$$\Pi_{\mathbf{R}}(\mathbf{X}, \mathbf{Y})\hat{f} = -\langle \mathbf{R} \text{symm}(\mathbf{X}^T \mathbf{Y}), \nabla \hat{f}|_{\mathbf{R}} \rangle,$$

where $\text{symm}(\mathbf{A}) := 1/2(\mathbf{A} + \mathbf{A}^T)$ is the symmetric component of \mathbf{A} . The Newton method search direction at iteration k is the unique tangent vector $\mathbf{d}_k \in T_{\mathbf{R}}\mathcal{S}\mathcal{O}(3)$ that satisfies $\text{Hess}f(\mathbf{X}, \mathbf{d}_k) = -\langle \mathbf{X}, \mathbf{d}_k \rangle$ for all $\mathbf{X} \in T_{\mathbf{R}}\mathcal{S}\mathcal{O}(3)$. Let $\mathcal{E} = \{\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_m\}$ be an orthonormal basis for $T_{\mathbf{R}}\mathcal{S}\mathcal{O}(3)$, then the Newton direction coordinates z_{ik} in the basis \mathcal{E} , such that $\mathbf{d}_k = \sum_{i=1}^m z_{ik} \mathbf{E}_i$, are computed by solving the linear system

$$\mathbf{A}_{\text{hess}} \mathbf{z}_k = \mathbf{b}_{\text{hess}},$$

where $\mathbf{z}_k = [z_{1k} \ \dots \ z_{mk}]^T$ is the unknown parameter vector, $\mathbf{b}_{\text{hess}} = -[\langle \mathbf{E}_1, \nabla f|_{\mathbf{R}_k} \rangle \ \dots \ \langle \mathbf{E}_m, \nabla f|_{\mathbf{R}_k} \rangle]^T$, and

$$\mathbf{A}_{\text{hess}} = \begin{bmatrix} \text{Hess}f(\mathbf{E}_1, \mathbf{E}_1) & \dots & \text{Hess}f(\mathbf{E}_1, \mathbf{E}_m) \\ \vdots & \ddots & \vdots \\ \text{Hess}f(\mathbf{E}_m, \mathbf{E}_1) & \dots & \text{Hess}f(\mathbf{E}_m, \mathbf{E}_m) \end{bmatrix}.$$

The orthonormal basis used in this paper is given by $\mathbf{E}_i = \frac{1}{\|\omega_i\|} \mathbf{R}\mathbf{S}(\omega_i)$, for $i = 1, 2, 3$, where $\omega_1 = [1 \ 0 \ 0]^T$, $\omega_2 = [0 \ 1 \ 0]^T$, and $\omega_3 = [0 \ 0 \ 1]^T$. The skew-symmetric matrix $\mathbf{S}(\mathbf{a}_1) \in \mathcal{K}(3)$ stands for the cross product operator, such that $\mathbf{a}_1 \times \mathbf{a}_2 = \mathbf{S}(\mathbf{a}_1) \mathbf{a}_2$, for some $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^3$.

4.2. Line search

A geodesic is defined as the curve in the manifold with zero acceleration, and is fully characterized by its initial position and velocity conditions, respectively, $\mathbf{y}_{\mathbf{d}_k}(0)$ and $\dot{\mathbf{y}}_{\mathbf{d}_k}(0)$. In the particular case of $\mathcal{S}\mathcal{O}(3)$, the geodesic $\mathbf{y}_{\mathbf{d}_k} : \mathcal{I} \rightarrow \mathcal{S}\mathcal{O}(3)$ is defined as

$$\mathbf{y}_{\mathbf{d}_k}(t) = \mathbf{R}_k e^{\mathbf{R}_k^T \mathbf{d}_k t},$$

where \mathcal{I} is an interval in \mathbb{R} , $\mathbf{d}_k \in \mathbf{R}_k \mathcal{K}(3)$ identifies the tangent vector, and the initial conditions are given by $\mathbf{y}_{\mathbf{d}_k}(0) = \mathbf{R}_k$ and

$\dot{\mathbf{y}}_{\mathbf{d}_k}(0) = \mathbf{d}_k$. Differentiation of the geodesic function, relative to the step variable, t , yields

$$\dot{\mathbf{y}}_{\mathbf{d}_k}(t) := \frac{d \mathbf{y}_{\mathbf{d}_k}(t)}{dt} = \mathbf{d}_k e^{\mathbf{R}_k^T \mathbf{d}_k t}.$$

The step size optimization subproblem of Algorithm 3 is numerically solved using the Wolfe conditions [28], as presented in the previous section, generalized to line search on geodesics. To this end, consider the line search cost function along geodesics $\phi : \mathbb{R} \rightarrow \mathbb{R}$ defined as $\phi(t) = f(\mathbf{y}_{\mathbf{d}_k}(t))$ with derivative given by

$$\dot{\phi}(t) = \nabla^T f|_{\mathbf{y}_{\mathbf{d}_k}(t)} \dot{\mathbf{y}}_{\mathbf{d}_k}(t).$$

The remaining of the algorithm is in every aspect identical to that of the line search using the Wolfe rules described in Section 3.2.

4.3. Closed-form line search

Taking advantage of the periodicity of the objective function, the exact optimal solution for line search problem can be found by simply determining the roots of a fourth order polynomial, as detailed in [31] for a similar cost function. It is noted, however, that if the position installation bias is considered for calibration the direct application of this closed-form line search would not be possible, as the objective function would have a non-periodic component. Given a search direction \mathbf{d} , computed using either the Gradient or the Newton methods for $\mathcal{S}\mathcal{O}(3)$, the line search optimization subproblem is solved by minimizing $\phi(t) = f(\mathbf{y}_{\mathbf{d}}(t))$. It is a matter of algebraic manipulation to show that the line search cost function can be written as

$$\phi(t) = \sum_{i=1}^n \frac{1}{2n\sigma_i^2} \text{tr}(\mathbf{M}_i^T e^{-\Omega t} \mathbf{R}^T \mathbf{N}_i \mathbf{R} e^{\Omega t} \mathbf{M}_i + 2\mathbf{W}_i^T \mathbf{R} e^{\Omega t} \mathbf{M}_i + \mathbf{C}_i) \quad (10)$$

where $\mathbf{M}_i = \mathbf{p}_i$, $\mathbf{N}_i = \mathbf{H}_i^T \mathbf{H}_i$, $\mathbf{W}_i = \mathbf{c}_i^T \mathbf{H}_i$, $\mathbf{C}_i = \mathbf{c}_i^T \mathbf{c}_i$ and $\Omega = \frac{1}{\|\omega\|} \mathbf{S}(\omega)$ with $\mathbf{S}(\omega) = \mathbf{R}^T \mathbf{d}$. As Ω has unit length, Rodrigues' formula, $e^{\Omega t} = I + \Omega \sin t + \Omega^2 (1 - \cos t)$, can be readily applied. Substituting this formula into (10) and simplifying, it can be seen that

$$\phi(t) = k_1 + k_2 \sin t + k_3 \cos t + k_4 \sin 2t + k_5 \cos 2t,$$

where $k_j, j = 1, \dots, 5$ are constant scalars. To find the optimal value for the step size the first order condition of optimality, given by $\frac{d\phi(t)}{dt} = 0$, yields

$$k_2 \cos t - k_3 \sin t + 2k_4 \cos 2t - 2k_5 \sin 2t = 0$$

and the optimization subproblem is now to find the values of $t \in [0, 2\pi)$ for which the previous condition is satisfied. The first step is to make a trigonometric half-angle substitution, $x = \tan \frac{t}{2}$, reducing the first order condition of optimality to a fourth order polynomial in x ,

$$x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0 = 0,$$

where $b_i, i = 0, \dots, 3$ are constant scalars. After finding the roots of this quartic polynomial, which is a standard procedure described in [35], the optimal value of t is the real root for which the cost function is minimal.

As this closed-form step size computation algorithm does not guarantee, by itself, a decrease in the cost function, the Wolfe rule described in previous section is used whenever a higher cost is detected. This behavior may occur near the optimum, as a consequence of having noise in the acquired data.

5. Simulation results

To assess the performance of the proposed algorithms, a digital elevation map (DEM) is synthetically generated in Matlab, based

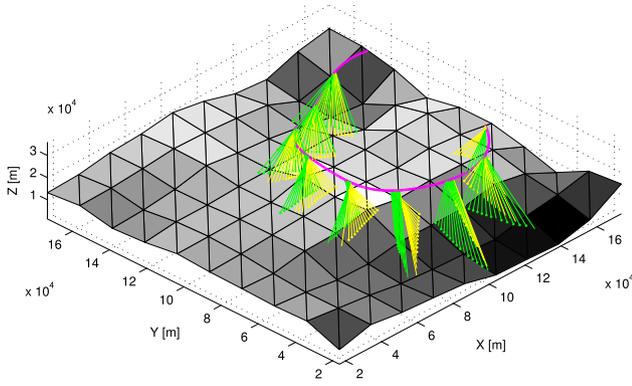


Fig. 5. Comparative calibration example: calibration surface \mathcal{S}_0 , trajectory (magenta), calibrated (green) and uncalibrated (yellow) laser beams. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

on the shuttle radar topography mission (SRTM) data for the area of Grand Canyon, AZ, USA (see [36–39] for further details). The data is used to define the terrain surface that has enough features disposed without symmetries, \mathcal{S} , which is then approximated by a set of planes to define the known calibration surface, \mathcal{S}_0 . This can be seen in the example of the comparative calibration shown in Fig. 5, where the platform trajectory, the calibrated points, and the uncalibrated points using the initial condition, are overlaid on the known calibration surface. An automatic calibration example is also shown in Fig. 6, featuring the two different trajectories and the approximated surfaces, \mathcal{S}_1 and \mathcal{S}_2 , which were obtained from two clouds of simulated laser data, \mathcal{C}_1 and \mathcal{C}_2 . Naturally, it can be seen that the uncalibrated surfaces are very distant from each other, Fig. 6(a), whereas the calibrated surfaces are virtually coincident, Fig. 6(b).

This section compares the accuracy and performance of the different optimization approaches and calibration algorithms introduced before, resorting to extensive Monte Carlo runs. To this end, the distance to the optimal value is defined in $\mathcal{S}\mathcal{O}(3)$ and the solution of the Euler angle optimization approach λ^* is evaluated as $\mathbf{R}^* = \mathbf{R}(\lambda^*) \in \mathcal{S}\mathcal{O}(3)$. Let \mathbf{R}_{true} , $\mathbf{R}^* \in \mathcal{S}\mathcal{O}(3)$ be, respectively, the true rotation matrix and the optimal rotation matrix obtained using the proposed algorithms, and let also $\tilde{\mathbf{R}} = \mathbf{R}_{true}^T \mathbf{R}^* \in \mathcal{S}\mathcal{O}(3)$ be the error rotation matrix. The performance of each method can be evaluated by defining the distance from $\tilde{\mathbf{R}}$ to \mathbf{I} as

$$\|\tilde{\mathbf{R}}\|_{\mathcal{S}\mathcal{O}(3)} = \arccos\left(\frac{1}{2}(\text{tr}(\mathbf{R}_{true}^T \mathbf{R}^*) - 1)\right).$$

The initial conditions for all the Monte Carlo runs presented in this section were computed using a uniform distribution between $\pm 30^\circ$ for each component of the ZYX Euler angle vector λ_0 yielding the rotation matrix $\mathbf{R}_0 = \mathbf{R}(\lambda_0)$. The true value for all the trials is defined by the rotation matrix, $\mathbf{R}_{true} = \mathbf{R}(\lambda_{true})$, obtained from the ZYX Euler angles vector $\lambda_{true} = [5.73 \ 2.86 \ -2.29]^T$ degree.

The stopping criterion used was $\frac{\|\nabla f|_{\mathbf{R}_k}\|}{\|\nabla f|_{\mathbf{R}_0}\|} < 10^{-6}$ and all the simulations were performed using the Matlab[®] simulation environment on an Intel[®] Xeon[™] E5645 processor at 2.4 GHz with Ubuntu operative system. Although both the Gradient and the Newton search directions were introduced for each optimization approach, the best convergence rates were obtained using the Newton direction, and for that reason, the results presented in this paper always use the Newton search direction.

Table 1
Summary of simulation results for the comparative calibration.

	Method			Noise
	NLW	SOW	SOC	
Average CPU time (s)	49.6	53.2	40.4	No
	95.8	99.5	46.4	Yes
Average number of iterations	6.8	6.9	5.9	No
	7.4	7.8	6.8	Yes
Failure (%)	5.8	2.0	0.0	No
	5.5	1.3	0.0	Yes
RMS error ($^\circ$)	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	No
	1.82	1.11	0.93	Yes
Maximum error ($^\circ$)	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	No
	4.48	1.56	1.33	Yes

5.1. Comparative calibration

Considering the comparative calibration algorithm, three different configurations of optimization methods and line search algorithms are tested:

- NLW:** nonlinear optimization using Euler angles parameterization and Wolfe rule;
- SOW:** optimization on $\mathcal{S}\mathcal{O}(3)$ using Wolfe rule; and
- SOC:** optimization on $\mathcal{S}\mathcal{O}(3)$ using closed-form step size.

These configurations will be compared in this section in order to highlight the advantages and disadvantages of each of them. Firstly, these three methods are tested using a computer generated noise-free data set, for which the objective function is smooth near the optimal value, as the reconstructed 3-D points can be made to be coincident to the calibration surface \mathcal{S}_0 . In addition, the methods are tested with another data set, for which the simulated LiDAR range data is corrupted with additive white Gaussian noise with zero mean and standard deviation of approximately 10% of the maximum height variation of the simulation terrain. The objective of these simulation scenarios is to see the influence of the imperfections when the actual surface \mathcal{S} is approximated by the control surface \mathcal{S}_0 and when the reconstructed 3-D points cannot be made coincident with the calibration surface. The experiment consisted of 1000 Monte Carlo runs with random initial conditions, and for each initial condition all methods were tested with and without noise addition. Considering the objective function complexity analysis detailed in Section 2.3, there are three main parameters involved: the number of point clouds, n_c ; the maximum number of points per cloud, n_p ; and the maximum number of planes used to approximate each cloud by a planewise surface, n_s , as defined in Eqs. (7) and (8). For the comparative calibration under analysis, there is only one point cloud, $n_c = 1$, with $n_p = 800$ points, and the surface approximation procedure considers a maximum of $n_s = 240$ planes.

Table 1 presents several statistical indicators of the simulation results for the comparative calibration algorithm: the average computation time, the number of iterations, the percentage of failure, and, most importantly, the root mean squared (RMS) and maximum errors, which are computed from the distance errors in $\mathcal{S}\mathcal{O}(3)$ between the obtained solutions and the true rotation matrix. The computational efficiency of the methods can be observed through the average CPU time and number of iterations. While NLW and SOW have similar values for these indicators, it is evident that the Riemannian method with closed-form step size (SOC) shows a significant decrease in both indicators, particularly in the CPU time, either with or without noisy data. When the data is corrupted by noise, there is also an increase of nearly 30% on the average CPU time of all the methods relative to the case where no

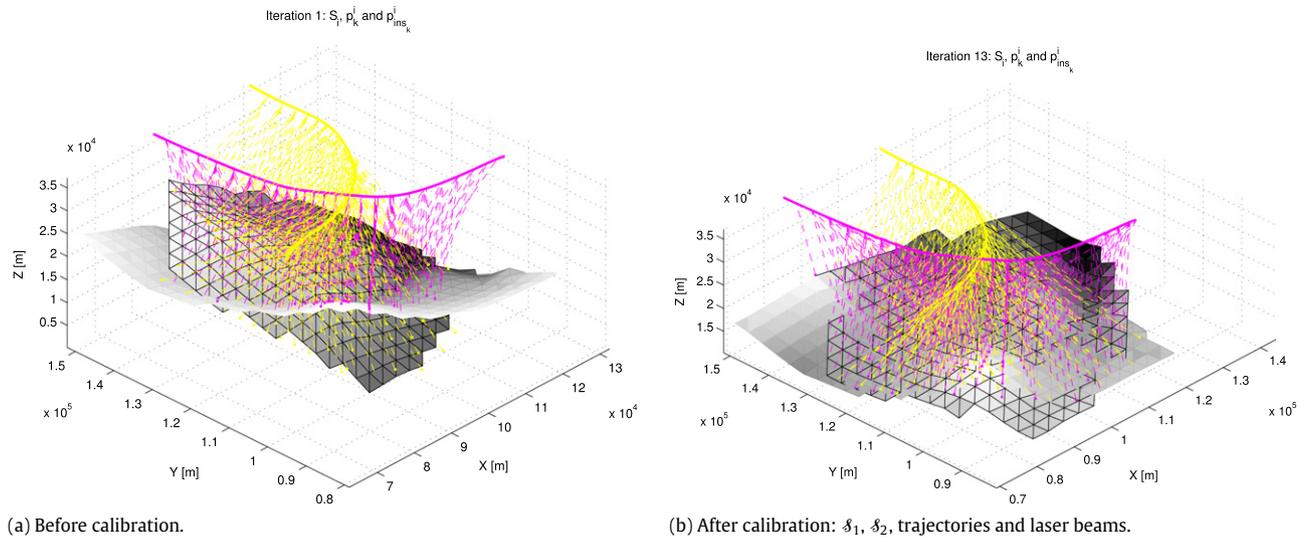


Fig. 6. Automatic calibration example: approximated surfaces δ_1 and δ_2 , trajectories, and laser beams.

noise is considered, which can be attributed, for instance, to the increased difficulty of the line search procedure.

The use of the percentage of failure is justified by the fact that the problem at hand is highly nonlinear and nonconvex, as the control surface has discontinuities and even the actual surface may introduce local minima. It can be noticed that, while the first two methods (NLW and SOW) have failure rates of less than 6%, the method that resorts to the closed-form step size (SOC) seems to be less sensitive to these hazards for the presented Monte Carlo simulations.

The performance of the comparative calibration algorithm and the three optimization methods can be evaluated using the RMS and maximum errors for the case where the data is corrupted by noise. Regarding the simulations without noise, the maximum error is always less than 3.0×10^{-6} degree. The main trend to consider in these indicators is that the optimization in $\delta\mathcal{O}(3)$ yields smaller errors than the optimization using Euler angles (NLW), showing a decrease of, at least, 39% in the RMS error and 65% in the maximum error. Besides having smaller average CPU time, the SOC method also proved to be the most accurate method, with a RMS error of 0.93° and a maximum error of 1.33° . Therefore, it is fair to say that the Riemannian optimization using a closed-form step size is more efficient and has the best performance, presenting the best qualities in the analyzed aspects: computation time, number of iterations, less prone to failures and most accurate estimates in the presence of noise.

5.2. Automatic calibration

The simulation results presented in this section aim at validating and analyzing the properties of the automatic calibration algorithm as well as to demonstrate that there is a relation between the number of planes considered for the surface approximation and the accuracy of the solution that can be achieved by the optimization method, for the same number of laser measurements. Noting that the previous results indicate that the optimization on $\delta\mathcal{O}(3)$ has a higher accuracy than the optimization using Euler angles, the presented results only consider the following two configurations for the automatic calibration:

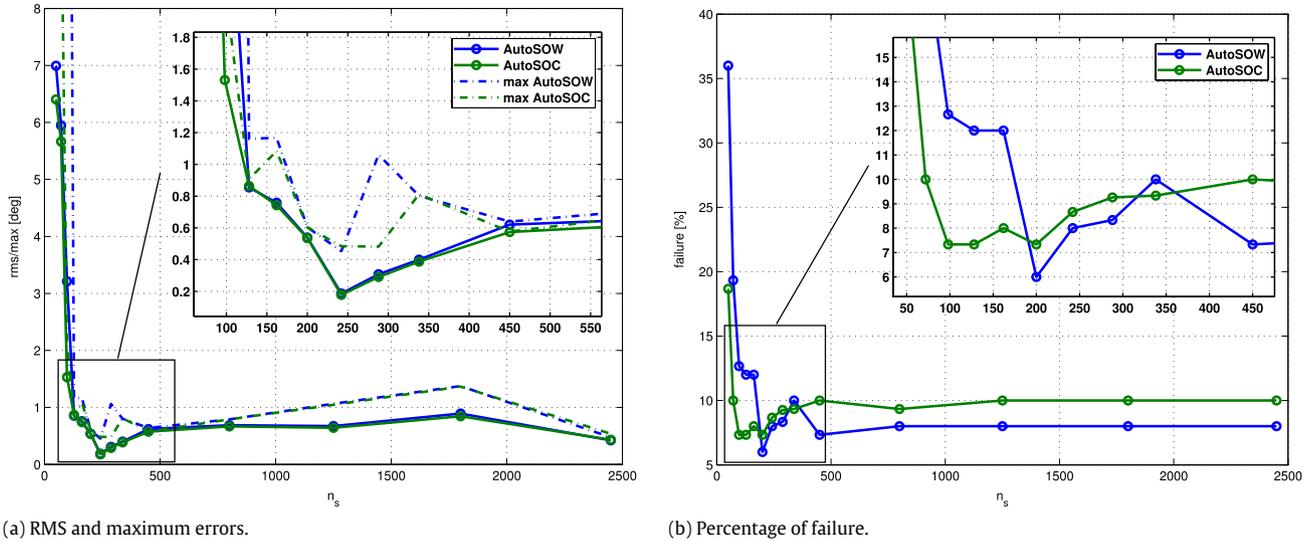
AutoSOW: automatic calibration using optimization on $\delta\mathcal{O}(3)$ and Wolfe rule; and

AutoSOC: automatic calibration using optimization on $\delta\mathcal{O}(3)$ and closed-form step size.

The results consist of Monte Carlo runs with random and uniformly distributed initial conditions, as detailed for the comparative calibration results. Two point clouds are considered, $n_c = 2$, each of them with 2334 points (thus, $n_p = 2334$), and the surface approximation is based on a xy -plane uniform sampling grid for each cloud of laser points, finding the best z -coordinate value based on the surrounding laser points. A set of planes is then generated to be used by the calibration method in each iteration as a surface approximation of that point cloud.

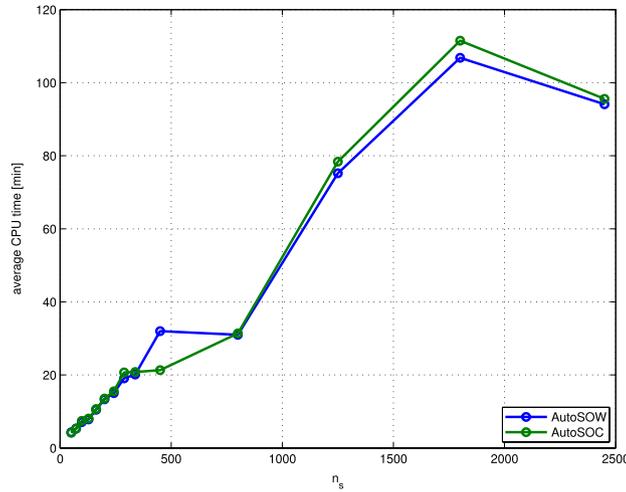
To establish the relation between the maximum number of planes and the solution accuracy, several surface approximation scenarios are considered, ranging from $n_s = 50$ to $n_s = 2450$ planes. Fig. 7 presents the main statistical indicators of the simulation described above: the average computation time, the percentage of failure, and also the RMS and maximum errors. Each point in these plots represents the results of 150 or 100 Monte Carlo runs, respectively, for $n_s \leq 800$ or for $n_s > 800$. The first striking evidence provided by these results is that AutoSOW and AutoSOC have similar performance if the maximum number of planes n_s is greater than 500. It is also visible that the AutoSOC method consistently yields better accuracy in terms of RMS and maximum errors than the AutoSOW method, which seems to perform marginally better only when $n_s = 2450$. Regarding the percentage of failure, the AutoSOC method shows a much greater immunity to local minima than the AutoSOW method when n_s is smaller than 200 planes, for instance, achieving a value lower than 10% even for n_s as low as 70 planes. Conversely, for n_s greater than 200 planes the AutoSOW method is consistently less prone to local minima.

One major conclusion emerging from these simulation results is that the most appropriate choice of the parameter n_s might not be the highest possible in terms of computational power, as the most accurate solutions are obtained for $n_s \in [200, 300]$, where both methods were able to obtain maximum errors below 0.5° and RMS errors below 0.2° . A possible explanation for this behavior is that, when the number of planes used for the surface approximation approaches the number of points of each cloud, the accuracy of the approximation might not increase, or may even be overfitted to the measured point cloud. This can result in degradation of the achievable accuracy when comparing one point cloud to the overfitted surface approximation for another point cloud. Nevertheless, for any value of n_s above 125, both methods show RMS errors below 1° while having automatic calibration times as low as 8 min. These results indicate that the proposed automatic calibration methods can provide accurate estimates of the attitude installation matrix of 2-D LiDARs.



(a) RMS and maximum errors.

(b) Percentage of failure.



(c) Average CPU time.

Fig. 7. Automatic calibration analysis.

The evaluation of the performance differences between the comparative and the automatic calibration approaches is somehow unfair and difficult, as the former considers a known calibration surface to compare with the LiDAR data. It is also impossible for the latter approach to have a completely error free simulation, as each measured cloud of points is approximated by a set of planes and, thus, will always have approximation errors. Nonetheless, the accuracy of the comparative calibration, when noise is considered, can be confronted with the accuracy of the automatic calibration methods. Thus, it is noted that the RMS errors of the automatic methods, when n_s is above 125, are always below those of the former methods, which for the SOW and SOC methods are 1.114° and 0.935°, respectively. Regarding the much higher computational times, there are two factors that explain this difference: (i) the increased number of point-to-plane comparisons and (ii) the higher surface approximation accuracy used in the Monte Carlo simulation runs.

6. Experimental results

This section presents the validation of the automatic calibration algorithms using experimental data acquired using a custom unmanned helicopter shown in Fig. 8, which is a property of the Institute for Systems and Robotics. The vehicle is equipped with a

Memsense nanoIMU, an Ashtech GPS receiver, and a Hokuyo UTM-30LX laser scanner. In addition to the available continuously operating reference stations (CORS), an additional reference station was setup in the field where the LiDAR data was acquired. Although these reference stations can be used to provide real-time GPS corrections to the vehicle positioning system for higher accuracy, this information was stored and used for post-processing of the GPS data. The post-processed GPS and IMU measurements were used to obtain the pose of the vehicle at each time instant, resorting to well known attitude and positioning filters based on the results presented in [40,41], which, together with the laser measurements, can be used to obtain two clouds of points to test the calibration algorithms. While the performance of the considered attitude filter, in similar conditions to those used in this experimental results, is reported to be 0.125° in terms of mean angle error, the post-processed GPS data has been reported to have a standard deviation below 2 cm when the base station is not more than 5 km away from the vehicle, which was ensured during the data acquisition. As the position filter used in this experimental setup implements the sensor fusion of the GPS and acceleration measurements, the standard deviation of the position errors should be smaller than 2 cm. The available specifications for the Hokuyo laser sensor used in these experiments indicate that the range of measurement of standard deviation is 1 cm from 0.1 to 10 m and 3 cm from 10 to 30 m, it has



Fig. 8. Unmanned helicopter used for airborne LiDAR data acquisition for the calibration experiment.



Fig. 9. Airborne image of terrain acquired for calibration, from which two overlapping patches with approximate dimensions of 90×30 m were chosen for the calibration experiment.

0.25° of angular resolution, and a laser footprint of approximately 20 cm at a distance of 20 m.

The worst case scenario for the proposed calibration algorithm is a flat terrain, as it would be impossible to solve the calibration problem independently of the vehicle trajectories. Conversely, the calibration will achieve better performance if the environment has smooth but steep slopes and contains non-symmetrical 3-D features, as long as an appropriate surface approximation algorithm is considered. The terrain acquired for calibration is shown in Fig. 9, where the presence of vegetation (which generates a sparser cloud of points) and the small slope variations of the terrain can be noticed, thus, increasing the difficulty of the calibration procedure. In the data sets used for calibration, the vehicle describes two different and overlapping trajectories, as shown in Figs. 11(c) and 12(b).

The experimental results presented in this section are focused on the automatic calibration algorithms, AutoSOW and AutoSOC, as it is extremely difficult to have a known calibration surface, necessary for the comparative calibration algorithms (NLW, SOW, and SOC). As it is equally demanding to obtain accurate enough ground truth attitude installation matrix, these results do not intend to show that the optimization solutions are close to some true angular installation bias. Instead, a statistical consistent measure of the performance can be obtained through extensive Monte Carlo runs for random initial conditions, showing that the solutions provided by the proposed methods have a low standard deviation. Exploiting this experimental analysis direction, the presented experimental results consider $n_c = 2$ point clouds and several scenarios for the remaining parameters of the computational complexity, n_p and n_s . In particular, a subsampling procedure is applied to the two acquired point clouds, which have a total of 42 834 and 53 899 points,

Table 2

Summary of experimental results for the automatic calibration with $n_p = 4000$.

n_s	200	800	2450	Method
Avg. CPU time (h)	1.7	4.1	11.8	AutoSOW
	2.0	4.4	12.6	AutoSOC
Avg. number of iterations	11.0	11.2	11.7	AutoSOW
	15.0	12.6	13.2	AutoSOC
Percentage of failure (%)	5.3	6.0	8.0	AutoSOW
	4.7	3.0	6.0	AutoSOC
Solution standard deviation ($^\circ$)	1.79	1.35	0.75	AutoSOW
	1.68	0.94	0.75	AutoSOC
Solution maximum deviation ($^\circ$)	16.33	2.30	2.63	AutoSOW
	16.31	1.92	1.22	AutoSOC
Number of runs	150	100	50	

respectively, so that the maximum number of points per cloud is limited to a value in the interval $n_p \in [1000, 4000]$ points. This results resort to the same computational capabilities used in the simulation results, presented in the previous section, and the initial conditions were computed using a uniform distribution between $\pm 30^\circ$ for each component of the ZYX Euler angles. For instance, the average solution is found to be $\lambda_{avg} = [1.65 \quad -2.64 \quad -2.66]^T$ degree, for the AutoSOC algorithm with 4000 points per cloud and $n_s = 800$, which was computed from the average rotation matrix in $\mathcal{S}\mathcal{O}(3)$ obtained with the method described in [42].

The statistical indicators of the performance of the two automatic calibration algorithms, considering several scenarios of accuracy, are presented in Fig. 10 and, particularly for the case when $n_p = 4000$, in Table 2. One key conclusion drawn from the analysis of these experimental results is the fact that the standard deviation of the calibration solutions drops as the values of n_p and n_s are increased, either using SOW or SOC algorithms, achieving 0.75° for $n_p = 4000$ and $n_s = 2450$. An exception to this rule is when values of n_s are near or above those of n_p , which yields a stagnation or degradation of performance, while also increasing the CPU requirements. In addition, the AutoSOC algorithm, which uses the closed-form step size, achieves higher accuracy in terms of standard deviation than the one using only the Wolfe rule (AutoSOW), echoing the simulation results presented before. The downside of AutoSOC in these experimental results seems to be a consistently (but marginally) larger average computational time.

It is also interesting to note that, when $n_s = 800$, an increase of $n_p = 3000$ to $n_p = 4000$ results in a mild decrease on the computational time of both algorithms, which might indicate that the lower values are not appropriate to fully describe the terrain, yielding poor convergence rates of the optimization algorithm. Also, the fact that for $n_p = 4000$ the maximum deviation of the AutoSOW algorithm increases when changing from $n_s = 800$ to $n_s = 2450$ is also interesting. The reasoning behind this may be related to having n_s and n_p too close to each other, or to the previously supported evidence that the AutoSOW algorithm may be more prone to local minima than the AutoSOC algorithm for certain choices of n_s .

As mentioned in the automatic calibration simulation results, when the number of planes used for the surface approximation approaches the number of points of each cloud, the accuracy of the approximation might not increase, or the surface approximation may even be overfitted to the measured point cloud, yielding a degradation of the achievable solution accuracy. It is also noted that the choice of the total number of points in each cloud (and ultimately n_p) defines how well the point cloud used for calibration represents the actual surface, thus influencing the limitations in the choice of n_s for the surface approximation. Provided that each point cloud is a good representation of the actual terrain, it is argued that the choice of n_s is limited by, but not directly dependent

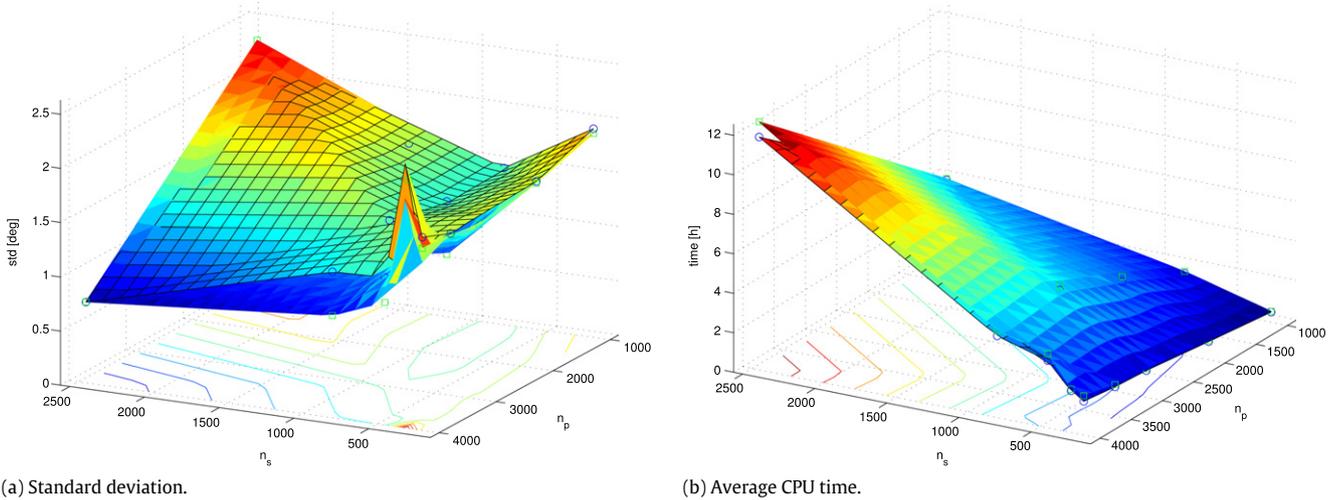


Fig. 10. Automatic calibration experimental data analysis. The AutoSOW algorithm is represented using dark surface edges and the AutoSOC is represented with interpolated colors for the surface edges. The data points used to generate the surfaces are represented with blue circles and green squares, respectively, for the AutoSOW and AutoSOC algorithms. Note that the axes are rotated for the sake of clarity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

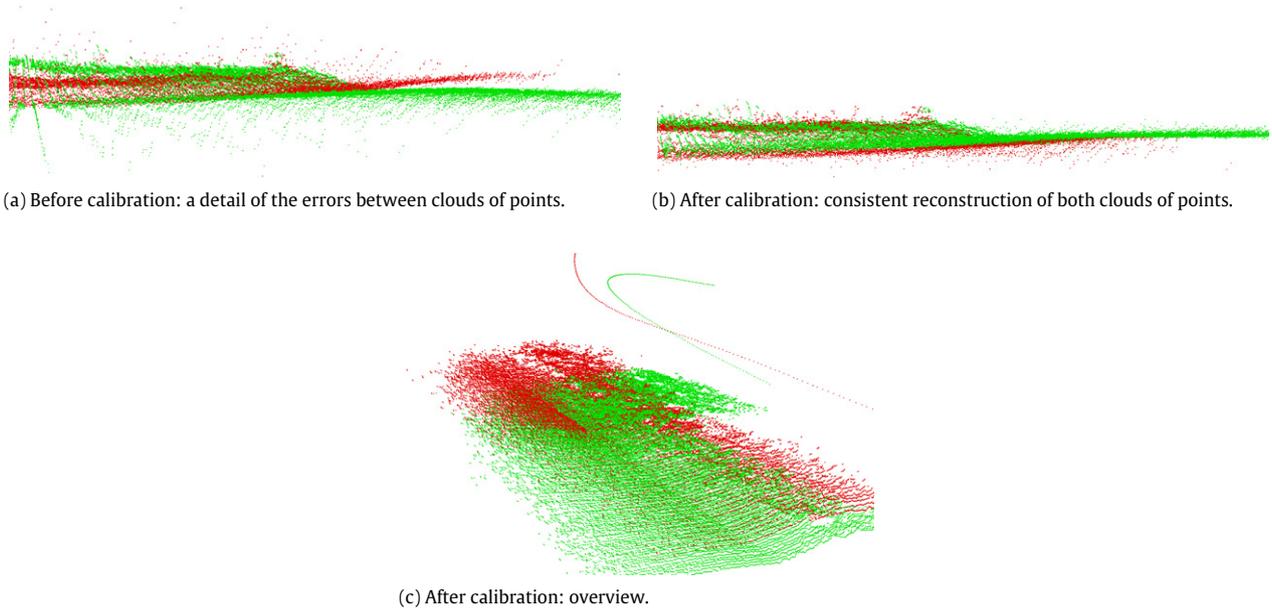


Fig. 11. Automatic calibration experimental data: \mathcal{C}_1 in red, \mathcal{C}_2 in green, and respective vehicle trajectories. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

on the choice of n_p , but rather dependent on the size, geometry, and complexity of the most relevant terrain features for the calibration.

The reconstructed point clouds and their approximation surfaces used in the algorithms are presented, respectively, in Figs. 11 and 12, before and after the calibration. It can be seen that without calibration there is a considerable error between the two clouds of points, Fig. 11(a), which is mitigated using the calibrated attitude installation bias found by the proposed algorithms, Fig. 11(b).

7. Concluding remarks

This paper addressed the problem of 2-D LiDAR calibration on aerial vehicles by building on a comparative calibration algorithm and proposing a novel automatic calibration algorithm, resorting to numerical optimization methods to estimate the attitude mounting bias rotation matrix $\mathbf{R} \in \mathcal{SO}(3)$. While for the comparative calibration algorithm this is achieved by minimizing the distance between a cloud of acquired LiDAR points and a known calibra-

tion surface, the automatic calibration algorithm minimizes the distances between several clouds of acquired LiDAR points, assuming no a priori knowledge of the terrain. The latter algorithm is one of the major contributions of this paper, as it suppresses the use of a known calibration surface, representing the true terrain to which all measured points are compared. To solve these estimation problems, two numerical optimization approaches were used: (i) non-linear optimization using the ZYX Euler angles parameterization of the attitude bias installation matrix, $\mathbf{R} = \mathbf{R}(\lambda)$, and (ii) a Riemannian optimization framework for the group of special orthogonal matrices, $\mathcal{SO}(3)$. Both approaches were tested using the Newton direction, the Wolfe rule for the line search subproblem and for the case of the optimization in $\mathcal{SO}(3)$, a closed-form solution for the computation of the step size was also used.

The performance and limitations of the comparative and the automatic calibration algorithms, as well as the different numerical optimization approaches, were tested in Matlab simulation environment and with experimental data, from where it is possible to

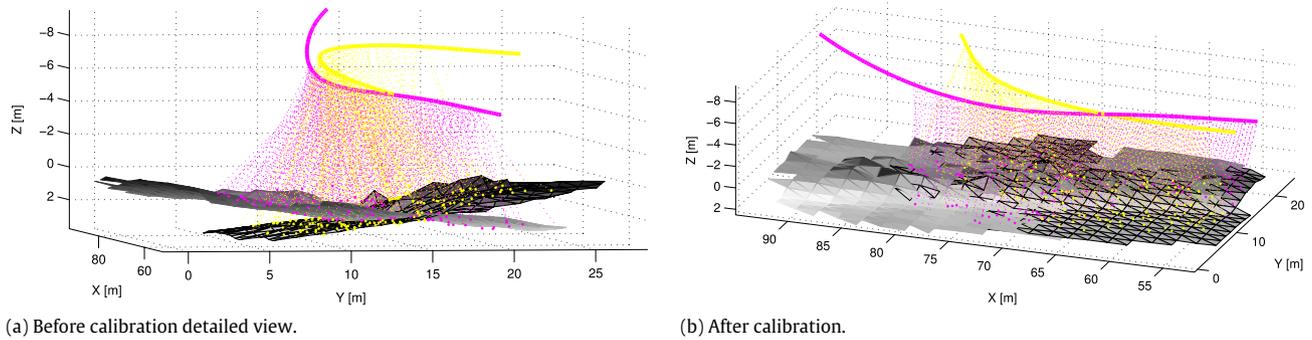


Fig. 12. Automatic calibration experimental data: plane-wise surface representation of the two point clouds (using a gray scale depending on the height of each surface plane), vehicle trajectories (in magenta) and a subset of the laser beams used in the presented results (in dashed yellow). To better distinguish the two different surfaces, one is represented using dark edges for each plane and the other is represented with interpolated colors for the edges, depending on the face color of each plane. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

conclude that all methods are able to find good estimates for the 2-D LiDAR calibration problem, even with real life and noisy airborne data. The optimization on $\mathcal{J}\mathcal{O}(3)$ with the closed-form step size computation displays some immunity to local minima and higher accuracy estimates. Additionally, the accuracy of the estimated parameters obtained using the automatic calibration algorithm tends to increase with the number of planes used to describe the surface approximation of each cloud of laser measured points, as well as with the number of points per cloud, despite possible limitations imposed by the sensors, terrain, and trajectories of the vehicle. Nonetheless, the presented results also indicate that values of n_s close to or above n_p can result in a degradation of accuracy in the solution of the calibration problem, as the surface approximation will be an overfitted surface for that particular point cloud.

Future directions of research include the convergence analysis of the algorithm and considering nonuniform surface approximations in order to decrease the computational time, as the same approximation error might be achieved with less planes. Further work is also needed in order to address the computational complexity of the algorithm, by using efficient data structures to speed up the search procedures and dedicated C/C++ functions to tackle some Matlab limitations, and also to include other sources of reconstruction errors, such as the time delay between laser acquisition and the INS/GPS data or the range measurement error.

Acknowledgments

The authors would like to express their gratitude to J.M. Vasconcelos, coauthor in the first conference article regarding this subject, and to the DSOR Lab team, particularly to R. Cunha, B. Carreira, A. Oliveira, B. Gomes, and J. Tojeira, for providing the ideal platform for data acquisition, which has enabled the experimental validation of the calibration algorithms presented in this paper. The SRTM US1 data used as a base for the simulated terrain was obtained online through the Reverb tool at the NASA Land Processes Distributed Active Archive Center (LP DAAC), USGS/Earth Resources Observation and Science (EROS) Center, Sioux Falls, South Dakota (at https://lpdaac.usgs.gov/data_access).

This work was partially supported by project FCT, LARSys [PEst-OE/EEI/LA0009/2013], by project FCT AMMAIA (PTDC/HIS-ARQ/103227/2008), and by project AIRTICI from AdI through the POS Conhecimento Program that includes FEDER funds. The work of Bruno Guerreiro was partially supported by the Ph.D. Student Grant SFRH/BD/21781/2005 from the Portuguese FCT POCTI programme.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <http://dx.doi.org/10.1016/j.robot.2014.05.001>.

References

- [1] W.B. Krabill, R.H. Thomas, C.F. Martin, R.N. Swift, E.B. Frederick, Accuracy of airborne laser altimetry over the greenland ice sheet, *Int. J. Remote Sens.* 16 (1995) 1211–1222.
- [2] E. Naesset, T. Okland, Estimating tree height and tree crown properties using airborne scanning laser in a boreal nature reserve, *Remote Sens. Environ.* 79 (2002) 105–115.
- [3] I. Posner, D. Schroeter, P. Newman, Online generation of scene descriptions in urban environments, *Robot. Auton. Syst.* 56 (2008) 901–914.
- [4] K.M. Wurm, H. Kretschmar, R. Kümmerle, C. Stachniss, W. Burgard, Identifying vegetation from laser data in structured outdoor environments, *Robot. Auton. Syst.* 62 (2014) 675–684.
- [5] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: part I, *IEEE Robot. Autom. Mag.* 13 (2006) 99–110.
- [6] S. Thrun, M. Diel, D. Hähnel, Scan alignment and 3-D surface modeling with a helicopter platform, in: *Proceedings of the 4th International Conference on Field and Service Robotics*, 2003.
- [7] T. Schenk, Modeling and analysing systematic errors of airborne laser scanners, *Technical Notes in Photogrammetry 19*, Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University, Columbus, OH, 2001.
- [8] S. Filin, Calibration of airborne and spaceborne laser altimeters using natural surfaces, (Ph.D. thesis) The Ohio State University, 2001.
- [9] M. Bosse, R. Zlot, P. Flick, Zebedee: design of a Spring-mounted 3-D range sensor with application to mobile mapping, *IEEE Trans. Robot.* 28 (2012) 1104–1119.
- [10] K. Georgiev, R.T. Creed, R. Lakaemper, Fast plane extraction in 3D range data based on line segments, in: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2011, pp. 3808–3815. <http://dx.doi.org/10.1109/IROS.2011.6094916>.
- [11] J.R. Ridgway, J. Minster, N. Williams, J.L. Bufton, W.B. Krabill, Airborne laser altimeter survey of Long Valley, California, *Geophys. J. Int.* 131 (1997) 267–280.
- [12] P.J. Besl, H.D. McKay, A method for registration of 3-D shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (1992) 239–256.
- [13] Y. Chen, G.G. Medioni, Object modelling by registration of multiple range images, *Image Vis. Comput.* 10 (1992) 145–155.
- [14] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, 2001, pp. 145–152. <http://dx.doi.org/10.1109/IM.2001.924423>.
- [15] A. Habib, T. Schenk, A new approach for matching surfaces from laser scanners and optical sensors, *Int. Arch. Photogramm. Remote Sens.* XXXII (3/W14) (1999).
- [16] S. Filin, Recovery of systematic biases in laser altimeters using natural surfaces, *Int. Arch. Photogramm. Remote Sens.* 34 (2001).
- [17] B. Guerreiro, C. Silvestre, P. Oliveira, J. Vasconcelos, Nonlinear and geometric optimization methods for LADAR calibration, in: *IEEE International Conference on Robotics and Automation*, 2008, 2008, pp. 1406–1411. <http://dx.doi.org/10.1109/ROBOT.2008.4543399>.
- [18] J. Underwood, A. Hill, T. Peynot, S. Scheding, Error modeling and calibration of exteroceptive sensors for accurate mapping applications, *J. Field Robot.* 27 (2010) 2–20.
- [19] C. Gao, J. Spletzer, On-line calibration of multiple LIDARs on a mobile vehicle platform, in: *IEEE International Conference on Robotics and Automation*, 2010, pp. 279–284. <http://dx.doi.org/10.1109/ROBOT.2010.5509880>.
- [20] J.A. Castellanos, R. Martínez-Cantin, J.D. Tardós, J. Neira, Robocentric map joining: improving the consistency of EKF-SLAM, *Robot. Auton. Syst.* 55 (2007) 21–29.
- [21] B.J. Guerreiro, P. Batista, C. Silvestre, P. Oliveira, Globally asymptotically stable sensor-based simultaneous localization and mapping, *Trans. Robot.* 29 (2013).
- [22] R. Kümmerle, G. Grisetti, W. Burgard, Simultaneous parameter calibration, localization, and mapping, *Adv. Robot.* 26 (2012) 2021–2041.
- [23] J. Kelly, G.S. Sukhatme, Visual-inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration, *Int. J. Robot. Res.* 30 (2011) 56–79.

- [24] J. Levinson, S. Thrun, Unsupervised calibration for multi-beam lasers, in: Proceedings of the International Symposium on Experimental Robotics, 2010.
- [25] W. Maddern, A. Harrison, P. Newman, Lost in translation (and rotation): rapid extrinsic calibration for 2D and 3D LIDARs, in: IEEE International Conference on Robotics and Automation, 2012, pp. 3096–3102. <http://dx.doi.org/10.1109/ICRA.2012.6224607>.
- [26] S.M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1993.
- [27] D.P. Bertsekas, Nonlinear Programming, second ed., Athena Scientific, 1999.
- [28] J. Nocedal, S. Wright, Numerical Optimization, in: Springer Series in Operation Research, Springer, 1999.
- [29] A. Edelman, T.A. Arias, S.T. Smith, Geometry of algorithms with orthogonality constraints, *SIAM J. Matrix Anal. Appl.* 20 (1998) 303–353.
- [30] P.A. Absil, R. Mahony, R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, 2007.
- [31] F.C. Park, J. Kim, C. Kee, Geometric descent algorithms for attitude determination using the global positioning system, *AIAA J.* 23 (2000) 26–33.
- [32] B. Guerreiro, C. Silvestre, P. Oliveira, Automatic LADAR calibration methods using geometric optimization, in: IEEE International Conference on Robotics and Automation, 2011, pp. 969–974. <http://dx.doi.org/10.1109/ICRA.2011.5979729>.
- [33] S. Umeyama, Least-squares estimation of transformation parameters between two point patterns, *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (1991) 376–380.
- [34] J.C. Gower, G.B. Dijkstra, Procrustes Problems, Oxford University Press, 2004.
- [35] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Dover Publications, New York, 1972.
- [36] T.G. Farr, M. Kobrick, Shuttle radar topography mission produces a wealth of data, *EOS Trans. Am. Geophys. Union* 81 (2000) 583–585.
- [37] T.G. Farr, P.A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, D. Seal, S. Shaffer, J. Shimada, J. Umland, M. Werner, M. Oskin, D. Burbank, D. Alsdorf, The shuttle radar topography mission, *Rev. Geophys.* 45 (2007).
- [38] M. Kobrick, On the toes of giants—how srtm was born, *Photogramm. Eng. Remote Sens.* 72 (2006) 206–210.
- [39] P. Rosen, S. Hensley, I. Joughin, F.K. Li, S. Madsen, E. Rodriguez, R.M. Goldstein, Synthetic aperture radar interferometry, *Proc. IEEE* 88 (2000) 333–382.
- [40] P. Batista, C. Silvestre, P. Oliveira, On the observability of linear motion quantities in navigation systems, *Systems Control Lett.* 60 (2011) 101–110.
- [41] P. Batista, C. Silvestre, P. Oliveira, Sensor-based globally asymptotically stable filters for attitude estimation: analysis, design, and performance evaluation, *IEEE Trans. Autom. Control* 57 (2012) 2095–2100.
- [42] I. Sharf, A. Wolf, M. Rubin, Arithmetic and geometric solutions for average rigid-body rotation, *Mech. Mach. Theory* 45 (2010) 1239–1251.



of autonomous vehicles, laser calibration methods and 3-D reconstruction, as well as nonlinear model predictive control.



Computer Engineering of the Faculty of Science and Technology of the University of Macau. Over the past years, he has conducted research on the subjects of navigation guidance and control of air and underwater robots.

His research interests include linear and nonlinear control theory, coordinated control of multiple vehicles, gain scheduled control, integrated design of guidance and control systems, inertial navigation systems, and mission control and real time architectures for complex autonomous systems with applications to unmanned air and underwater vehicles.



Research projects, over the last 20 years.

Bruno J. Guerreiro received the Licenciatura and Ph.D. degrees in Electrical and Computer Engineering in 2004 and 2013, respectively, both from Instituto Superior Técnico (IST), Lisbon, Portugal.

He is currently a post-doctoral researcher at the Dynamic Systems and Ocean Robotics Lab of the Institute for Systems and Robotics, at IST, and has been involved in several research and development projects regarding the use of autonomous vehicles for automatic inspection of infrastructures and archeological sites. His research interests include the sensor-based navigation and control

Carlos Silvestre received the Licenciatura degree in Electrical Engineering from the Instituto Superior Técnico (IST) of Lisbon, Portugal, in 1987 and the M.Sc. degree in Electrical Engineering and the Ph.D. degree in Control Science from the same school in 1991 and 2000, respectively. In 2011 he received the Habilitation in Electrical Engineering and Computers also from IST. Since 2000, he is with the Department of Electrical Engineering of the Instituto Superior Técnico, where he is currently an Associate Professor of Control and Robotics on leave. Since 2012 he is an Associate Professor of the Department of Electrical and

Computer Engineering of the Faculty of Science and Technology of the University of Macau. Over the past years, he has conducted research on the subjects of navigation guidance and control of air and underwater robots.

His research interests include linear and nonlinear control theory, coordinated control of multiple vehicles, gain scheduled control, integrated design of guidance and control systems, inertial navigation systems, and mission control and real time architectures for complex autonomous systems with applications to unmanned air and underwater vehicles.

Paulo Oliveira completed the Ph.D. in 2002 from the Instituto Superior Técnico, Lisbon, Portugal. He is an Associate Professor of the Department of Mechanical Engineering of the Instituto Superior Técnico, Lisbon, Portugal and a researcher in the Institute for Systems and Robotics of the Associated Laboratory for Robotics and Systems in Engineering and Science, Lisbon, Portugal.

His areas of scientific activity are Robotics and Autonomous Vehicles with special focus on the fields of Sensor Fusion, Navigation, Positioning, and Estimation. He participated in more than 15 Portuguese and European