2D Lidar to Kinematic Chain Calibration Using Planar Features of Indoor Scenes

Bernardo Lourenço[3], Tiago Madeira[2]
Paulo Dias[1,2], Vitor Santos[1,3,4], Miguel Oliveira[1,3,4]

[1] Institute of Electronics and Informatics Engineering of Aveiro, University of Aveiro
[2] Department of Electronics, Telecommunications and Informatics, University of Aveiro
[3] Department of Mechanical Engineering, University of Aveiro
[4] Institute for Systems and Computer Engineering, Technology and Science

Abstract
2D Laser Range Finders (2D-LRFs) are sensors commonly used in the field of robotics as they provide accurate range measurements with high angular resolution. These sensors can be coupled with mechanical units, which, by granting an additional degree of freedom to the movement of the LRF, enable the 3D perception of a scene. The reconstruction procedure consists of the concatenation of several consecutive scans in a single point cloud representation. To be successful, the extrinsic transformation between the LRF and the kinematic chain of the actuated mechanism (a Pan-tilt unit for example) must be estimated with high accuracy, or the quality of the 3D reconstructed point clouds is compromised. In this work, a calibration procedure to estimate this extrinsic transformation is proposed. The method does not require predefined markers or calibration patterns since it uses planar features in the acquired point clouds. Qualitative inspections show that the proposed method significantly reduces the artifacts that typically appear in point clouds obtained from moving 2D-LRFs due to inaccurate calibrations. Furthermore, quantitative results and comparisons with a high-resolution laser scanner demonstrate that the calibrated point cloud represents correctly the existing geometries and generates point clouds which are very similar to the ones collected by the reference scanner.

Keywords: LRF, calibration, reconstruction, point cloud, registration, optimization

1. Introduction

Laser Range Finder, or LRFs, are among the most used sensors in the field of robotics. Due to their precision, they are often used in applications such as Simultaneous Localisation and Mapping (Cadena et al. (2016)) and 3D reconstruction (Saito et al. (2010)). Their use is even increasing due to several factors, such as decreasing costs, new research and development, and a fundamental role in some relevant applications such as autonomous driving.

In this paper, a 2D-LRF is used as part of a 3D reconstruction system to produce accurate models of real scenes. The use of a 2D-LRF, when compared with a 3D Laser Scanner, significantly reduces the costs and might help to generalise this kind of application (Dias et al. (2006)). To transform a 2D-LRF into a 3D sensor, the common solution involves the integration of the unit on a turntable (Maurelli et al. (2009)) or a Pan-tilt unit (PTU) (Klimentjew et al. (2009)), as this enables a full 3D perception. One of the challenges of this approach is the calibration of the position of the LRF relative to the coordinate frame of a dynamic kinematic chain, such as a PTU. This process is known as the LRF-PTU extrinsic calibration.

This calibration is crucial since any error introduces a shift in the position of each point, which can be seen as deformations on the point cloud resulting from the acquisition. Such deformation can be observed in Figure 1, which presents multiple misaligned planes. These deformations have a significant impact on subsequent point cloud processing methods, that rely on the accurate representation of the 3D environments, i.e., registration (for example, the Iterative Closest Point (ICP) algorithm (Pomerleau et al. (2015))), segmentation and feature extraction algorithms.

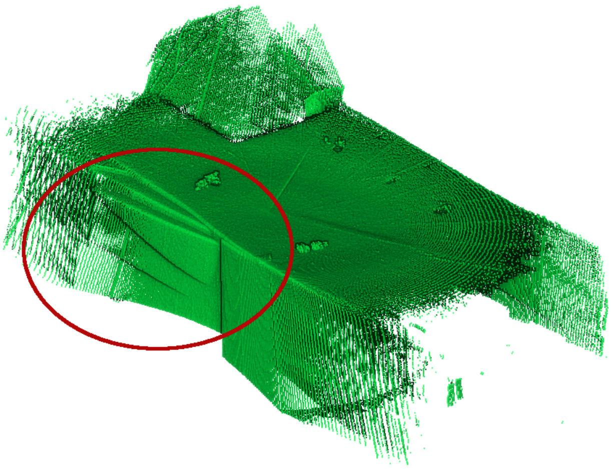

FIGURE 1: Example of a deformed point cloud, product of an inaccurate calibration of the LRF. The points where this deformation is most visible is marked in red.

A possible solution to perform the LRF-PTU calibration relies on the study of the geometry of the system, using a precise measuring instrument such as a tactile Coordinate Measuring Machine. However, this approach has several disadvantages: it is time-consuming, the instruments required are expensive and generally not easily available.

Therefore, reliable and automatic methods for the LRF extrinsic calibration are necessary. In this field, there are two main approaches, depending on which sensor the LRF is calibrated to. The first and the most studied method is the calibration of the LRF relative to a camera (Chen et al. (2016); Vasconcelos et al. (2012); Fan et al. (2019)), or cameras (Häselich et al. (2012); Oliveira et al. (2020)). In this method, the static transformation between the two sensors relies on finding correspondences between visual features captured by the camera and geometric features captured by the LRF. These features might be pre-defined markers viewed by both sensors, such as a planar chessboard (Kassir and Peynot (2010)) or a 3d trihedron chessboard (Tian et al. (2020)). However, these methods present some limitations in the case of LRF mounted on robotic arms: first, they require the inclusion of an additional camera in the system; secondly, the calibration of the camera has to be performed; and lastly, the error of both calibrations is combined during the reconstruction, which may increase the errors of the final reconstruction. In Nagy et al. (2019), the calibration of the LRF 3D sensor is performed online by reducing the distance between the point cloud obtained from the LRF sensor and one obtained by a Structure From Motion using a camera.

Another approach is to calibrate the 2D-LRF directly against a moving platform (Zeng et al. (2018)). Most of these methods rely on the comparison between the structure of the reconstructed points and an object with a known 3D geometry, such as a plane, a ball (Pereira et al. (2016)), or a cone (Almeida et al. (2012)). In Kim et al. (2013), a marker formed by two perpendicular planes is used, and their points are segmented from the reconstructed point cloud. Afterwards, the relationship between the normals of these planes is used to evaluate the calibration. Also, in Lembono et al. (2019), the LRF is positioned at the end-effector of a robotic arm, along with a sharp tool that is used to measure constraints in a flat plane. The planarity and distance constraints measured by the sharp tool and the 2D-LRF are enough to formulate a non-linear optimization where the position of the 2D-LRF can be determined. These methods, while interesting, have a few downsides: it may prove difficult to ensure that the two planes are perfectly perpendicular since it requires a very well-built marker; increasing the number of planes is not straightforward; and only a small portion of the sensors' field of view is used, which may lead to calibration errors.

This paper is an extension of Lourenço et al. (2019), where the approach was originally proposed. This extension focuses on the comparison of the point-clouds generated, in the same room, by our moving 2D-LRF and a commercial sensor from FARO (used as a reference sensor given its high precision and resolution). This is a significant improvement of the evaluation methodology, providing quantitative results by comparing the registration outcome with a High-Resolution Laser Scanner, and showing that the method created produced results on par with the point cloud taken with the reference sensor.

2. Problem Formulation

The 2D-LRF is a sensor that outputs several laser scans. Each laser scan is a set of range measurements taken sequentially along several directions, all coincident with a plane: the scanning plane. This is in fact why this equipment has a 2D nature. Thus, for each scan, a 2D slice of the scene is measured. Since the LRF is mounted on top of an actuated kinematic chain (the PTU), the movement of this chain positions the sensor in different poses in the 3D space. Through the accumulation of several laser scans acquired at different angles, a dense point cloud of the scene is produced. Note that, theoretically, the movement of the PTU may influence the measurements of the single scan, as described in

Pascoal and Santos (2010). However, because the frequency of scanning acquisition is so high when compared with the velocity of the joints of the PTU, the influence of this effect is reduced and was not considered in this work.

The process of accumulation is, in essence, based on the concatenation of 3D points from multiple scans. However, this grouping requires that the points of each laser scan are transformed into a common, static, reference frame. To correctly perform these transformations, one must apply a geometric transformation to each 3D point $\mathbf{q}^{(i)}$ measured in scan i, represented in the LRF's local coordinate system l. The transformed point $\mathbf{p}^{(i)}$, defined in the map reference frame, is determined by:

$$\mathbf{p}^{(i)} = {}^{m}\mathbf{T}_{e}^{(i)} \cdot {}^{e}\mathbf{T}_{l} \cdot \mathbf{q}^{(i)}, \tag{1}$$

where ${}^{m}\mathbf{T}_{e}{}^{(i)}$ is the transformation between the map and the PTU coordinate frames, which is dynamic and given by the process that receives a description of the kinematic chain, the joint values at the scan i, and computes the direct kinematics. In turn, ${}^{e}\mathbf{T}_{l}$ denotes the position of the LRF w.r.t, the PTU. In the end, the set of all the points $\mathbf{p}$ is denoted as the reconstructed point cloud $\mathcal{P}$.

Note that, although the transformation ${}^{e}\mathbf{T}_{l}$ is static, it influences the coordinates of the points $\mathbf{p}^{(i)}$, therefore being crucial to the accuracy with which three-dimensional objects are represented in the accumulated point cloud $\mathcal{P}$. Consequently, an accurate estimation of transformation ${}^{e}\mathbf{T}_{l}$ is paramount to the process of 3D reconstruction. This procedure that estimates the LRF to PTU transformation is referred to as the LRF-PTU extrinsic calibration.

In this work, a novel approach to perform LRF-PTU extrinsic calibration is proposed. The approach is based on an optimization procedure, which computes the transformation that generates planar point cloud sections in areas marked as planes. To this end, several existing scene structures may be used, such as walls, ceilings, or pavements. This is an additional advantage of our proposed method: it does not require dedicated calibration targets.

3. Proposed Approach

The working principle of this method relies on the following assumption: in a good calibration, the deviation of a measured point set w.r.t. to the real scene geometry should be minimal. This method focuses on planar geometries because of two reasons: they are easily segmented from the full point cloud, and it is straightforward to compute a similarity metric between the point cloud and the expected planar geometry. In other words, the calibration is correct if the deviation from a point set to the corresponding planar surface is minimal.

This method can be formulated as an optimization problem as illustrated in Figure 2. For each extrinsic transformation $\mathbf{T}$, a point cloud $\mathcal{P}$ is obtained merging the 2D LRF scans. Several point clouds are registered using the transformations $\mathbf{T}^{(i)}$ (obtained by the combination of the optimized parameters $\mathbf{x}$ and the initial transformations guess ${}^{m}\mathbf{T}_{e}^{(i)}$), to produce the point cloud $\mathbf{p}^{(i)} \in \mathcal{P}$. Several planar clusters segmented from this point cloud C_i are then compared using a cost function resulting in a the cost $\mathbf{f}$. By reducing the cost value $\mathbf{f}$, the optimizer will update the transformation parameter $\mathbf{x}$ resulting in a better calibration. $\mathbf{x}$ is defined in the optimization as:

$$\mathbf{x} = \underset{\mathbf{x}}{\operatorname{argmin}} \{ \mathrm{f}(\mathcal{P}) \}, \tag{2}$$

The optimization process involves several components: the parameterization, the cost function, and the optimizer. Each component is independent and can be understood separately. In the following section, each of the components will be further explained.

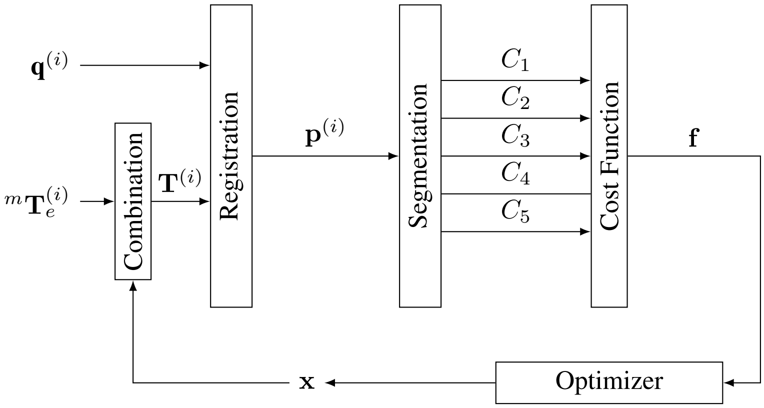

FIGURE 2: Overview of the calibration process. Points $\mathbf{q}^{(i)}$ from a scan are registered with the transformations $\mathbf{T}^{(i)}$ (a combination of the parameters $\mathbf{x}$ and the transformations ${}^{m}\mathbf{T}_{e}^{(i)}$), to produce the point cloud $\mathbf{p}^{(i)} \in \mathcal{P}$. Several planar clusters C_i from the point cloud are then optimized to reduce the cost f while updating the parameters $\mathbf{x}$.

3.1. Parametrization

The LRF-PTU extrinsic calibration requires the evaluation of 6 parameters representing the geometric transformation in space, determined by the transformation matrix $\mathbf{T}$. This transformation is decomposed into two components: a translation and a rotation. The translation can be represented as the translation vector $\mathbf{t} = [t_x, t_y, t_z]$, and the rotation is represented as a 3×3 rotation matrix $\mathbf{R}$. The rotation matrix has $3 \times 3 = 9$ elements, but only 3 degrees of freedom need to be used to represent the rotation. Popular parameterizations for rotations are Euler angles, quaternions, and axis-angle. However, not all representations are suitable for the optimization procedure. Parameterizations should not introduce more numerical sensibility than the one inherent to the problem itself. These parameterizations are called fair (Hornegger and Tomasi (1999)). Common parameterizations, such as the Euler angles and quaternions, are not fair parameterizations because of the *gimbal-lock* singularities, introduced by the former, and the unitary length constraint, in the case of the latter (Schmidt and Niemann (2001)).

The axis-angle representation is the most widely used to represent a rotation in optimization procedures, as it is a fair parameterization and has only three components. Any rotation can be represented as a rotation around an axis a, by an angle θ. Since a only represents the direction of the rotation (hence only having 2 degrees of freedom), it can be combined with the angle θ into a single rotation vector $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]$, as follows:

$$\begin{aligned} \theta &= |\boldsymbol{\omega}| \\ a &= \frac{\boldsymbol{\omega}}{|\boldsymbol{\omega}|} \end{aligned} \tag{3}$$

Computing the rotation matrix from ω is done using the Rodrigues' formula (Schmidt and Niemann (2001)):

$$\mathbf{R} = \mathbf{I} + \frac{\sin\theta}{\theta}\boldsymbol{\Omega} + \frac{1-\cos\theta}{\theta^2}\boldsymbol{\Omega}, \tag{4}$$

where $\mathbf{I}$ is the 3×3 identity matrix, and $\boldsymbol{\Omega}$ is given by:

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{5}$$

In conclusion, the parameter vector to be optimized will have six values: three representing the translation $\mathbf{t}$ and three representing the rotation vector ω (using the axis/angle representation). The parameter vector $\boldsymbol{x}$ to optimize is defined as:

$$\mathbf{x} = [t_1, t_2, t_3, \omega_1, \omega_2, \omega_3]\,. \tag{6}$$

3.2. Cost Function

The cost function is used in optimization procedures to evaluate the error between the real measurements and the expected results from a model. It yields a value that quantitatively describes the dissimilarity between the two. In this case, the cost function evaluates the distance of each point $\mathbf{p}$ in the point cloud $\mathcal{P}$ to their respective plane, assuming that all surfaces are planar. The greater this value, the larger the deformation of the point cloud, and conversely, the worse the calibration.

As an initial step, the segmentation of the point cloud is performed. The segmentation is a procedure that divides the point cloud $\mathcal{P}$ into multiple clusters $\mathcal{C}_i$, each one corresponding to different planes of the scene. This segmentation is done before the calibration procedure, using the initial point cloud reconstructed as a first estimate. This estimate is based on rough values for the calibration obtained from an estimate of the translation and rotation of the position of the sensor, by visual inspection.

The segmentation of the plane was performed manually, using the software *CloudCompare*[1]. In this case, manual segmentation was the preferred method, as no automatic method performed adequately, and it did not involve a significant overhead to the user, as this kind of calibration is performed sparingly. In the future, an automatic method can be studied to further streamline this process.

At the beginning of the optimization process, the segmentation is loaded into memory as a lookup table (LUT). At each iteration, the reconstructed point cloud is segmented the same way, via this lookup table. An example of a segmentation can be seen in Figure 3.

[1] CloudCompare (https://www.danielgm.net/cc/) is a 3D point cloud and mesh processing software.

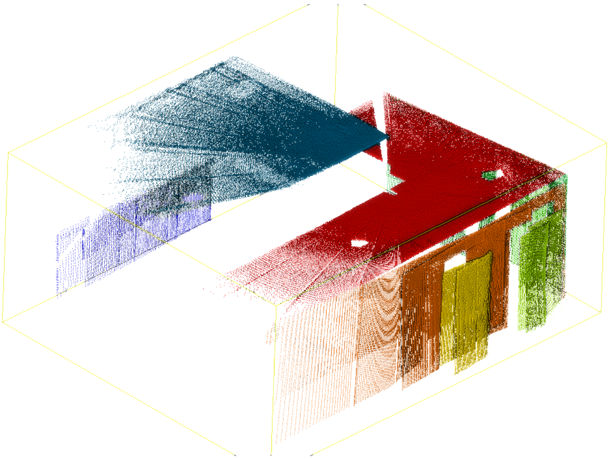

FIGURE 3: Example of a plane segmentation, where each colour represents a cluster.

After the segmentation, the calibration procedure can be performed. The plane equation for each point cluster is computed using the Principal Component Analysis method, or PCA. First, the centroid $\bar{p}$ of each plane is found, which is the same as the mean value of all the points $\mathbf{p} : (x, y, z) \in \mathbb{R}^3$:

$$\bar{p} = \sum_{i}^{N} p_i. \tag{7}$$

The co-variance 3×3 matrix $\mathbf{C}$ is then calculated:

$$\mathbf{C} = \sum_{i}^{N} (p_i - \bar{p}) \otimes (p_i - \bar{p}), \tag{8}$$

where $\otimes$ is the outer tensor product.

Then, the normal axis of the plane is found by an Eigen decomposition of the covariance matrix. The smallest eigenvalue λ_3 will be the variance σ^2 of the cluster. In other words, σ^2 is the mean square of the orthogonal distance of all points in the cluster to the plane. So, σ^2 can be a quantitative factor to measure the cost or each cluster. Formally, let us admit that the σ^2 has two components: the statistical error of the laser sensor σ^2_{sensor}, which is not affected by the calibration, and a second component σ^2_{calib}, that depends on the calibration error, resulting in $\sigma^2 = \sigma^2_{calib} + \sigma^2_{sensor}$. Thus, the idea is that, by minimizing σ^2, we get closer to an exact calibration. Therefore, the loss of each cluster can be expressed by the value of σ. This is commonly known in the literature as the Root Mean Square Deviation or RMS.

The principal axes of the plane are found by the eigenvalue decomposition of the covariance matrix $\mathbf{C}$. According to the PCA method, each eigenvalue represents the statistical deviation of the points in each principal axis, more specifically, the mean square error. Therefore, the deviation, or statistical error σ_i, of each point cluster $\mathbf{C}_i$ to the corresponding plane is determined by the smallest eigenvalue λ_3. The cost function can the sum of all σ_i, of the N clusters, according to:

$$\mathrm{f} = \sum_{i}^{N} \sigma_i. \tag{9}$$

3.3. Optimiser

The optimisation algorithm chosen was Powell's method, described in Powell (1964). This method finds a local minimum of a multi-dimensional unconstrained function that fits this particular optimisation, while not requiring the function's gradient, which is not known is this problem.

This algorithm performs the optimization using a standard gradient descent method. However, because the gradients can not be directly computed (the gradient of the objective function is unknown), the algorithm determines the jacobian by evaluation of the objective function in every principal direction, at every step. The best direction is then chosen with the jacobian. This has the advantage of not requiring the value of the gradient of the function, but at the same time, it demands far more computational power resulting in a longer optimization time when compared to other methods. For high-dimensional problems, this algorithm is not appliable, because the number of function evaluations per step grows quadratically with the number of dimensions to be optimized. However, the number of dimensions, in this case, is small (6 dimensions) so it does not perform poorly. The condition for stoping of the optimization can be specified as a maximum number of iterations, or when the error is below a certain threshold.

The implementation of this optimizer can be found in the Python scientific library SciPy [2], which was easily integrated into the calibration algorithm.

4. Results

This section presents some results of the calibration method proposed. Firstly, we present some clouds of points acquired from a real scene after the calibration. Secondly, we compare quantitatively the point cloud produced by our LRF-PTU calibrated system with a high-resolution FARO 3D Laser scanner.

4.1. Experimental Set-up

The hardware infrastructure used in this work was the 3D scanner *lemonbot*, shown in Figure 4. The principal components of this 3D scanner are the PTU *FLIR PTU-D46* and the 2D laser scanner *Hokuyo UTM30LX*[3]. The LRF was mounted on the PTU to have the scan plane in the vertical direction. The laser scanner can cover 270° with an angular resolution of 0.25°. The PTU has an angular resolution of 0.0032°, so the error in the transformation ${}^{m}\mathbf{T}_{e}$ (see Equation 1) is negligible. This 3D scanner was programmed using the *ROS* framework, which handles the hardware drivers, the transformation graph, and the recording of the scanning data.

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fmin_powell.html

[3] The 3D scanner, as seen in the Figure 4, has two laser scanners: the *Hokuyo UTM30LX* and the *SICK LMS100*. In this work, only the first was used.

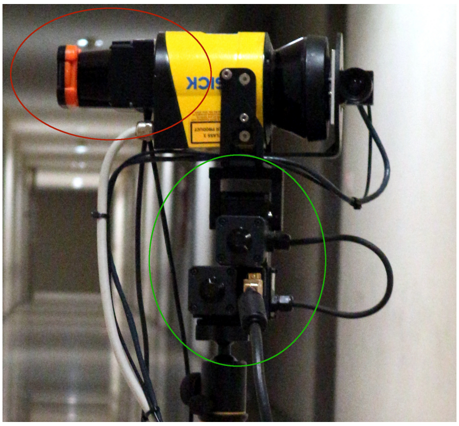

FIGURE 4: The *lemonbot* 3D scanner, used in this work. The LRF *Hokuyo UTM30* is marked in red and the *FLIR PTU-D46* is marked in green.

4.2. Data acquisition from real scene

For the verification, a dataset was acquired from a single room, which consists of 3 acquisitions[4]. The acquisitions were performed in different positions in the room. The room selected to evaluate the calibration process was a laboratory, not very cluttered, with the ceiling, walls, and floors all perpendicular and known to be planar surfaces. This was done to facilitate the plane segmentation and minimize calibration errors.

The optimization procedure was run minimizing the objective function in every dataset, as seen in Figure 5. In the final point clouds, the qualitative improvement is clear, when compared to the uncalibrated point cloud, as shown in Figure 6. These examples show the most common defects in poorly calibrated reconstructions. In the first example, the ceiling surface appears as multiple misaligned surfaces. This effect can be also seen in a wall, in the third example, which appears duplicated. The second example shows a deformation that results in a deformed corner structure in the point cloud. These deformations, found in the uncalibrated point clouds, are almost imperceptible in the calibrated point cloud.

Empirical studies showed that this calibration is resilient to the initial parameters. This study was made by testing different initial parameters, and the results were consistent for all trials. Also, in our extensive testing, we have not observed local minima. The algorithm, in a laptop with a *Intel Core i7-8550U* CPU, running in a single core takes about 1 hour to converge.

Despite the good qualitative result, a further quantitative analysis was performed to evaluate the method. The results were prepared in the following steps:

- A planar surface was extracted from both the calibrated and uncalibrated point cloud.

[4] One acquisition corresponds to a set of laser scans taken from a particular position in the scene. Each acquisition is processed and results in a point cloud.

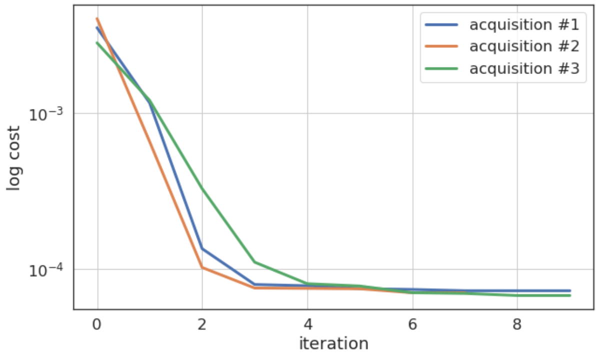

FIGURE 5: Graph of the evolution of the cost function during the optimization process. As seen, the cost function decreased roughly 50 times.

- A fitting plane was calculated for each surface, and the distance between the points to the surface was computed.
- The statistic deviation of the distances was calculated.

This method was performed for 3 manually segmented planes, such as the ones seen in Figure 3, and the segments can be seen in Table 1. As can be observed, the statistical deviation is, in fact, smaller in the calibrated point cloud. As an alternative view, in Figure 7, distribution of the signed deviation are shown for both the point clouds. The deviations of the calibrated point cloud are less scattered than the deviations of the uncalibrated point cloud. These results indicate that this calibration method does improve the results.

TABLE 1
Comparison between the mean μ and standard deviation σ of the distances of the points to the plane for both the calibrated and uncalibrated point clouds.

	μ(m)	σ(m)
Calibrated	0.0495	0.2589
Uncalibrated	0.0025	1.1211

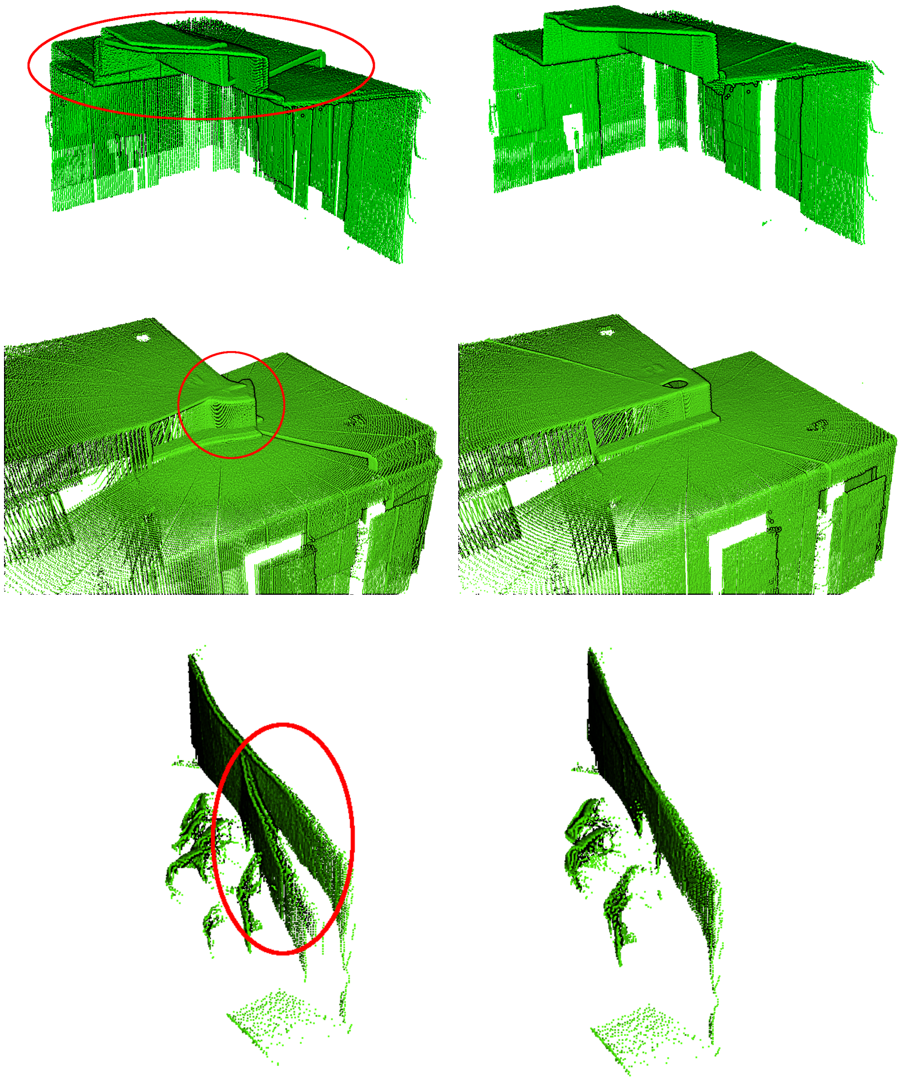

FIGURE 6: Side-by-side comparison of the un-calibrated (left) and calibrated (right) point clouds. The most noticeable distortions are marked in the un-calibrated point clouds and do not appear in the respective calibrated point clouds. This indicates that the method successfully reduces the distortions in the point clouds.

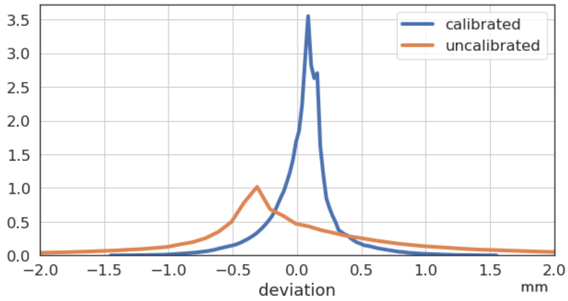

FIGURE 7: Distribution of the deviations of the points to the plane. The uncalibrated points' deviations are wider than the calibrated points, which indicates that the calibration improved the reconstructed point cloud.

4.3. FARO scan comparison

To validate our approach, a comparison was performed between the point cloud generated by the lemonbot after planar calibration, with the point cloud acquired using a FARO Focus Laser Scanner. Unfortunately, it was not possible to acquire the scene at the same time with both pieces of equipment. For this reason, some areas of the point cloud cannot be directly compared (as an example, sofas in the room are not located exactly in the same position between the two acquisitions). Nonetheless, these results are presented, since structural information (walls, ground, and ceilings) are common in both acquisitions.

The metric used for this evaluation procedure was the Hausdorff distance. It measures how far two subsets of a metric space are from each other, and is often used to evaluate the distance between two point clouds. It is defined as the greatest of all the distances from a point in one set to the closest point in the other set. This metric evaluates, not only an absolute distance but also the distance at each point of the cloud. These local values can be showcased visually through the application of a colormap. This way, large or small variations can be distinguished, as well as areas in the scene that were physically different between the acquisitions of either sensor.

Before computing the Hausdorff distance between the point clouds, it is necessary to align them in the same coordinate frame. ICP was used to perform this step. Afterward, the Hausdorff distance was calculated between the uncalibrated point cloud and the FARO point cloud, and also between the calibrated point cloud and the FARO point cloud. A qualitative colormap was applied, based on the values obtained.

Several views with the computed Hausdorff distance to the FARO laser scan, obtained from the calibrated and uncalibrated point clouds, are presented in Figure 8,9.

The mean and Root Mean Square (RMS) values (see Table 2) indicate an improvement of roughly $8\,\mathrm{mm}$, which corresponds to approximately 30%. The maximum distance to consider a point for Hausdorff distance calculation was empirically set to $50\,\mathrm{mm}$.

TABLE 2
Hausdorff distance from uncalibrated and calibrated point clouds to the FARO laser scan point cloud.

	μ (m)	σ (m)	min (m)	max (m)
Calibrated	0.0168	0.0200	0.0001	0.0500
Uncalibrated	0.0246	0.0283	0.0001	0.0500

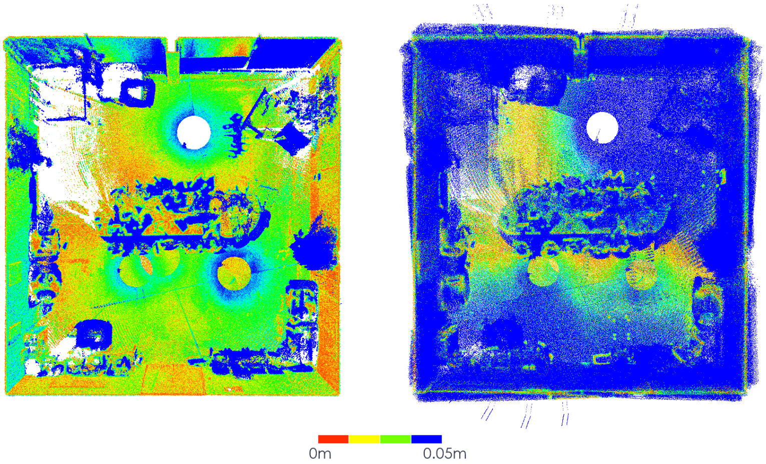

FIGURE 8: Hausdorff distance from captured point clouds to the FARO point cloud (top view). Calibrated point cloud on the left, uncalibrated point cloud on the right.

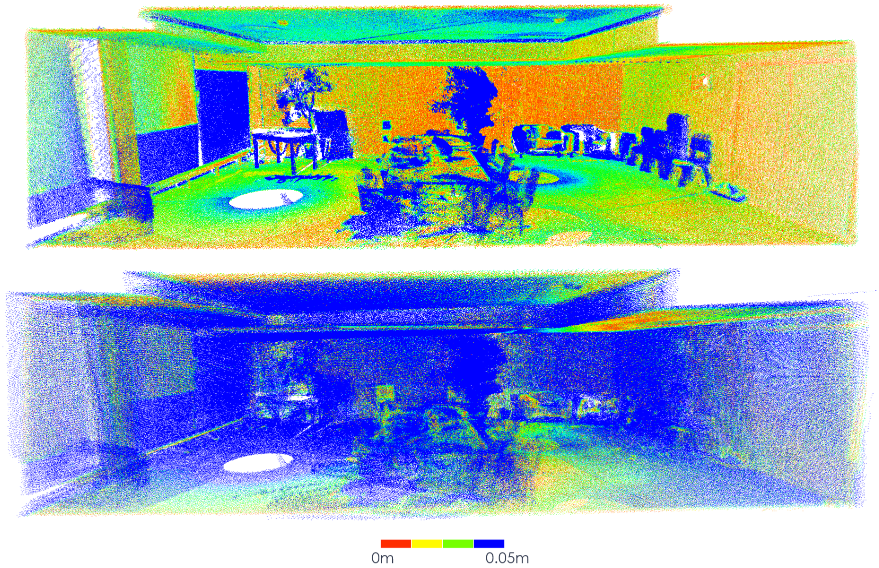

FIGURE 9: Hausdorff distance from captured point clouds to the FARO point cloud (side view). Calibrated point cloud on the top, uncalibrated point cloud on the bottom.

5. Conclusions

This paper presents a calibration procedure to determine the extrinsic transformation of 2D-LRF and PTU. The accuracy of this transformation is critical for multiple applications, of which, the most noticeable are 3D reconstruction systems. The incorrect calibration of these systems results in artifacts and deformations of the final reconstructions, which invalidates any successive work in these point clouds. The proposed method is based on the measurement of the deformations of reconstructed planar surfaces, and their minimization, through an optimization procedure. The effectiveness of the method was confirmed, through visual inspection of the resulting point cloud and statistical evaluation of the reconstructed geometries. A comparison with a high-resolution 3D Laser scanner also showed improvements in the reconstructed point cloud after the planar calibration procedure.

Future work can be done to improve or validate this algorithm further, such as a study of an automatic segmentation procedure; changing the optimization method to gradient descent, which would reduce significantly optimization time; the validation of this method with other dynamic kinematic chains, such as a robotic arm; and studying the impact of the calibration environment (for example, the number and pose of the planes and size of the room) in the calibration accuracy.

Acknowledgements

This Research Unit is founded by National Funds through multiple projects through the FCT, the Foundation for Science and Technology. This work was produced in the context of the project SeaAI-FA_02_2017_011 - Sistema de Enrolamento Filamentar Assistido por Braço Robótico e Visão com Inteligência Artificial, UID/CEC/00127/2019, and CYTED/TICs4CI - Aplicaciones TICS para Ciudades Inteligentes.

References

Almeida, M., Dias, P., Oliveira, M., and Santos, V. (2012). 3d-2d laser range finder calibration using a conic based geometry shape. In *Image Analysis and Recognition*, pages 312–319.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332.

Chen, Z., Yang, X., Zhang, C., and Jiang, S. (2016). Extrinsic calibration of a laser range finder and a camera based on the automatic detection of line feature. In *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 448–453.

Dias, P., Matos, M., and Santos, V. (2006). 3d reconstruction of real world scenes using a low-cost 3d range scanner. *Comp.-Aided Civil and Infrastruct. Engineering*, 21:486–497.

Fan, J., Huang, Y., Shan, J., Zhang, S., and Zhu, F. (2019). Extrinsic calibration between a camera and a 2d laser rangefinder using a photogrammetric control field. *Sensors*, 19:2030.

Hornegger, J. and Tomasi, C. (1999). Representation issues in the ml estimation of camera motion. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 1, pages 640–647.

Häselich, M., Bing, R., and Paulus, D. (2012). Calibration of multiple cameras to a 3d laser range finder. In *2012 IEEE International Conference on Emerging Signal Processing Applications*, pages 25–28.

Kassir, A. and Peynot, T. (2010). Reliable automatic camera-laser calibration. *Proc. Australasian Conference on Robotics & Automation*.

Kim, D., Kim, H., and Ha, J. (2013). Extrinsic calibration of rotating laser range finder using scene structure. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pages 701–703.

Klimentjew, D., Arli, M., and Zhang, J. (2009). 3d scene reconstruction based on a moving 2d laser range finder for service-robots. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1129–1134.

Lembono, T. S., Suarez-Ruiz, F., and Pham, Q. (2019). Scalar: Simultaneous calibration of 2-d laser and robot kinematic parameters using planarity and distance constraints. *IEEE Transactions on Automation Science and Engineering*, 16(4):1971–1979.

Lourenço, B., Oliveira, P., and Oliveira, M. (2019). Extrinsic calibration of 2d laser range finders using planar features. In *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 1–6.

Maurelli, F., Droeschel, D., Wisspeintner, T., May, S., and Surmann, H. (2009). A 3d laser scanner system for autonomous vehicle navigation. In *2009 International Conference on Advanced Robotics*, pages 1–6.

Nagy, B., Kovács, L., and Benedek, C. (2019). Sfm and semantic information based online targetless camera-lidar self-calibration. pages 1317–1321.

Oliveira, M., Castro, A., Madeira, T., Dias, P., and Santos, V. (2020). *A General Approach to the Extrinsic Calibration of Intelligent Vehicles Using ROS*, pages 203–215.

Pascoal, R. and Santos, V. (2010). Compensation of azimuthal distortions on a free spinning 2d laser range finder for 3d data set generation. In *Int. Conf. on Mobile Robots and Competitions*, pages 41–46.

Pereira, M., Silva, D., Santos, V., and Dias, P. (2016). Self calibration of multiple lidars and cameras on autonomous vehicles. *Robotics and Autonomous Systems*, 83:326 – 337.

Pomerleau, F., Colas, F., and Siegwart, R. (2015). *A Review of Point Cloud Registration Algorithms for Mobile Robotics*, volume 1.

Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162.

Saito, K., Yata, N., and Nagao, T. (2010). Three-dimensional scene reconstruction using stereo camera and laser range finder. In *Proceedings of SICE Annual Conference 2010*, pages 1515–1520.

Schmidt, J. and Niemann, H. (2001). Using quaternions for parametrizing 3-d rotations in unconstrained nonlinear optimization. In *Vision, Modeling, and Visualization*.

Tian, Z., Huang, Y., Zhu, F., and Ma, Y. (2020). The extrinsic calibration of area-scan camera and 2d laser rangefinder (lrf) using checkerboard trihedron. *IEEE Access*, 8:36166–36179.

Vasconcelos, F., Barreto, J. P., and Nunes, U. (2012). A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2097–2107.

Zeng, Y., Yu, H., Dai, H., Song, S., Lin, M., Sun, B., Jiang, W., and Q-H Meng, M. (2018). An improved calibration method for a rotating 2d lidar system. *Sensors*, 18:497.