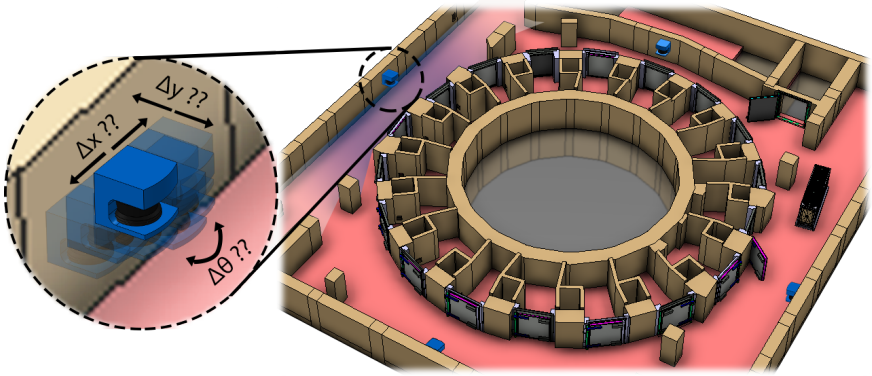
TÉCNICO LISBOA



Calibration of multiple laser range finders for mobile robot localization in ITER

Tiago André Reis de Sousa

Thesis to obtain the Master of Science Degree in

Engenharia Eletrotécnica e de Computadores

Supervisors: Doutor Alberto Manuel Martinho Vale

Professor Rodrigo Martins de Matos Ventura

Examination Committee

Chairperson: Professor João Fernando Cardoso Silva Sequeira

Supervisor: Professor Rodrigo Martins de Matos Ventura

Member of the Committee: Professor Vítor Manuel Ferreira dos Santos

June 2015

Aos meus pais e avós

À Margarida

Acknowledgments

My first word of appreciation goes to my supervisors, Doutor Alberto Vale, who introduced me to the laser distance measurement subject, and Professor Rodrigo Ventura, for their unwavering support, dedication to assist me, sharing mathematical expertise and insightful suggestions and advices.

I wish to express my sincere thanks to Director of ISR/IST, Professor João Sequeira, and President of IPFN/IST Doutor Bruno Gonçalves for providing me with all the necessary facilities for the research.

I place on record, my sincere thank you to ISR Department, for giving permission to use all required equipment and devices needed to perform the experimental tests.

I owe a special gratitude to Doutor Joaquim Marques, not only for the help at mathematical expertise, but mostly for the sincere and valuable guidance and encouragement extended to me specially at the beginning of this journey.

My deepest gratitude goes to my family, specially my parents and grandparents who sacrificed themselves on my behalf, for their support at all levels, unconditional encouragement and attention, providing me with the stability needed to succeed and accomplishment my objectives.

I want to express my warm thanks to my beloved girlfriend, Margarida, for being there when I most needed, emotional support throughout this venture, for all the caring, love and affection.

Last but not least, I also want to mention, and express my appreciation to my friends and all those who, directly or indirectly, have contributed to shortening the stressful times.

Resumo

As operações de manutenção no reator de fusão nuclear do ITER são efetuadas com recurso a veículos controlados remotamente. Por este motivo, é utilizado um algoritmo de localização para o qual um dos métodos propostos recorre a uma rede de sensores Laser Range Finder (LRF). Este método de localização necessita de um conhecimento preciso das poses (posição e orientação) de cada um dos sensores que compõem a rede. Um desvio na pose dos sensores compromete a precisão da localização dos veículos. Assim, para garantir o correto funcionamento da localização, tanto após a instalação dos sensores como ao longo da sua operação, é necessário proceder à sua (re)calibração com regularidade. É dada especial ênfase ao ambiente do ITER, contudo é possível a sua aplicação em ambiente industrial. Neste trabalho é abordado o problema da calibração de uma rede de sensores LRF. Foi desenvolvido um algoritmo para identificar a pose de cada um dos sensores no referencial de um mapa que se assume sempre disponível e preciso. Numa primeira fase, os dados obtidos através das leituras dos sensores sofrem um pré-processamento de modo a eliminar medidas díspares. O algoritmo foi preparado para lidar com diferentes cenários e situações quanto ao conhecimento de uma estimativa inicial da pose. Diferentes algoritmos de calibração, tendo como base o algoritmo ICP, foram abordados e estudados. Versões de força bruta do ICP ou alternativas como um algoritmo de votação foram implementadas para o caso em que a estimativa inicial da pose é parcialmente disponibilizada. Para situações em que a estimativa inicial não está disponível, foi necessário proceder à extração de elementos geométricos básicos como linhas e vértices a partir das leituras dos sensores, recorrendo a métodos de *clustering* e *feature extraction*. Na fase de associação, foi desenvolvido um algoritmo de força bruta que utiliza os elementos extraídos para efeitos de localização. Foram realizados vários testes tanto em ambiente de simulação como em cenários reais, utilizando sensores *laser* comerciais. Para efeitos de teste, em todas as situações identificadas, foram utilizadas poses distintas para cada um dos seis mapas elaborados. A conclusão propõe os algoritmos que melhor se adequam a cada uma das situações.

Palavras-chave: Laser Range Finder, Calibração, Localização, ICP.

Abstract

Maintenance operations on the ITER nuclear fusion reactor are carried out using remote-controlled vehicles. This is the reason why a localization algorithm is used. One of the methods proposed to implement it requires a network of Laser Range Finder (LRF) sensors. This method relies on the precise knowledge of the pose (position and orientation) of each sensor that composes the network. A deviation in the pose of the sensors compromises the localization accuracy of the vehicles. Thus, in order to ensure they keep on localizing properly, both after installing the sensors and while they are in use, it is necessary to (re)calibrate them regularly. They excel in the ITER environment, but it is also possible to use them in an industrial environment. This thesis addresses the calibration of a network of LRF sensors. An algorithm was developed to identify the pose of each of the sensors in the reference frame of a map that should always be available and accurate. In the first phase, the data obtained from the readings of the sensors are pre-processed to eliminate differing measurements. The algorithm was designed to deal with different scenarios and situations regarding the initial estimate of the pose. Nevertheless, different calibration algorithms based on the ICP algorithm are also discussed and studied. Brute force approaches of the ICP were implemented as well as other alternative versions such as a voting algorithm in situations where the initial estimate of the pose was only partially available. In case the initial estimate is not available, it is necessary to extract basic geometric elements, such as lines and vertices, from the readings of the sensors, by using clustering and feature extraction methods. In the phase of association, a brute force algorithm was developed. It uses the elements extracted for purposes of locating. Several tests were carried out both in simulations and in real-life scenarios using commercial laser sensors. For testing purposes, in all predefined settings, different poses were used for each one of the six maps that were conceived. This thesis finishes by describing the algorithms that best suit each one of these scenarios.

Keywords: Laser Range Finder, Calibration, Localization, ICP.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Producing enough energy to cover our civilizational primary energy needs and living standards, has always been one of the main goals and focuses of our modern civilization. The increasing demand of energy supply projected for the future, has raised the concern on the environmental impact of current ways of energy production and lead to the debate and discussion of reliable and new alternatives. Fusion power promises itself as a clean, and sustainable source of usable heat-energy meaning almost no green-house gases emissions, recyclable radioactive waste, and abundant, safe and efficient fuel supplies. The biggest challenge now is to prove that a large scale functionality and production is possible.

The International Thermonuclear Experimental Reactor (ITER) is a multinational in-progress experimental project of a fusion reactor involving investigators of many countries and represents the next step of demonstration. Facilities are located in the south of France in Cadarache and the first plasma creation is expected in 2020. Nuclear fusion energy tries to answer the problem of world growing energy demand without the environment impact that the actual sources of energy deliver. One of its greatest advantages is that it uses hydrogen isotopes (Deuterium and Tritium) which exists in relatively great quantities and can be easy obtained. During ITER operation, the Tokamak component will be responsible for holding a high temperature plasma (about 150M°C) resultant of the reaction of the isotopes. The source of energy is the avail of this heat produced. The plasma is then conducted with the aid of very powerful magnetic fields inside the vacuum vessel never touching its walls which acts as a first safety containment barrier. Inside is the Blanket layer covering its walls providing this way protection from heat and radiation. Among the great scientific challenges this project implies, such as controlling the plasma temperature, nominal and maintenance operations [1] are necessary in this hazardous and confined environment. Workers are not allowed to enter the ITER facilities during its operation and transportation of activated components due to high levels of radiation and temperature. Instead all operations will be handled completely remotely through cluttered environments. Some of the critical operations include the transportation of shielded casks, that enclosures the load and spare parts by the Cask and Plug Remote Handling System (CPRHS) vehicles, formed by the Cask Transport System (CTS) and Cask Envelop

System (CES) [2], and the operation of rescue vehicles such as the proposed Multi-Purpose Rescue Vehicle (MPRV) [3] for inspection, repair and component replacement. The vehicles could only be seen by a video surveillance system so a robust and precise localization system is necessary. To perform the previous and other tasks with minimal error margin, an accurate navigation method is needed due to the tight safety margins inside the building. The navigation method proposed in [4] uses the vehicles localization estimation to autonomously maneuver the vehicle through its path or to help a human operating it remotely.

Nowadays, the presence of active ranging devices such as LIght Detection And Ranging sensor (LIDAR) or, also called Laser Range Finder (LRF) systems, have becoming of widespread use in many domains specially in vision and remote handled vehicles [5]. It is no surprise to see these platforms including many sensors like cameras (IR and RGB), odometry, sonar and LIDAR and merging their information in contribution to such operations as navigation, map construction or object recognition, often simultaneously. A network of LRF sensors, illustrated in Figure 1.1, is proposed, and is very likely that this technology will be used, due to their immunity to magnetic fields that proliferate near the Tokamak and highly precise and accuracy distance measurements in long and short ranges, which the localization estimation method is based and heavily reliant on. An array of these sensors strategically placed is required to cover the maximum possible area and mitigate the occlusion problem. It was chosen to install the sensor systems on the building walls rather than opting for installing them in the vehicle because the radiation emitted from the vehicle load would rapidly damage the electronic hardware. Near the walls, the device mirror exposition to radiation is minimized since levels of radiation are relatively low. In order for a network of LIDAR sensors to be useful they should return measurements in a common relative or absolute frame coordinates so that the measurements from different sensors could have a valid correspondence between them.

The calibration procedure can be found in several distinct areas such as music, electronics, image, sensors. In general terms, the act of calibration refers to adjust some device(s) parameters in order to obtain a correct and desirable output or behavior. A comparison between a measurement of the variable subject to calibration and a standard measurement, which is consider correct, is necessarily implied in every process and the objective is to determine the necessary variation of these parameters to align the object of calibration with the standard. The measurement method must be the same in the two phases to mitigate the uncertainties associated with measurement nature. The standards used often agree to some international reference standards. Taking for instance the calibration of a simple analog scale: by measuring a mass of a predefined and accurate weight one only need to adjust the pointer to match the known weight value on the scale.

Calibration is a problem that relies on most of the sensors, in particular, the ones used in high demanding accuracy applications. Even if it is very precise, what use is there for a sensor whose readings cannot be trusted? This brings another pertinent subject on the table and that is the difference between precise and accurate and whom are affected by calibration. Precision is the capability of a sensor to reproduce the same results under the same measurement circumstances whether those are correct or not. It is a measure of the repeatability and is limited by the random errors. On the other hand,

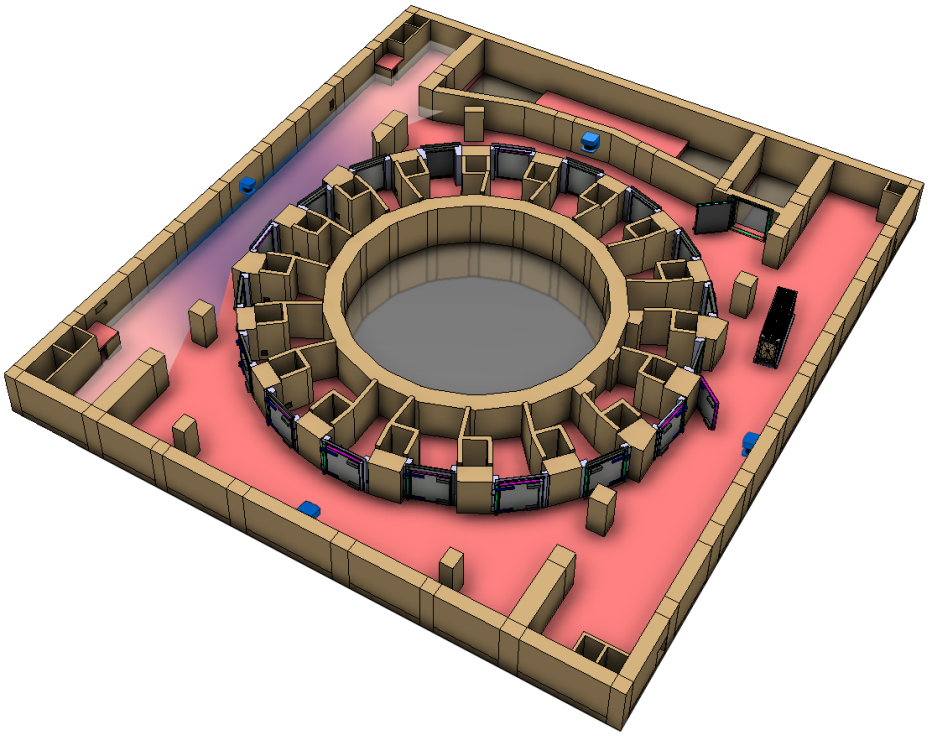


Figure 1.1: ITER B1 building with LRF sensors application.

accuracy is the capability the device has to delivery results close to the true value of the quantity being measured and it is limited by systematic errors. An illustration of the two situations is shown in Figure 1.2 from [6]. The calibration process operates only on the field of accuracy by correcting for bias (the difference between the mean and the true value).

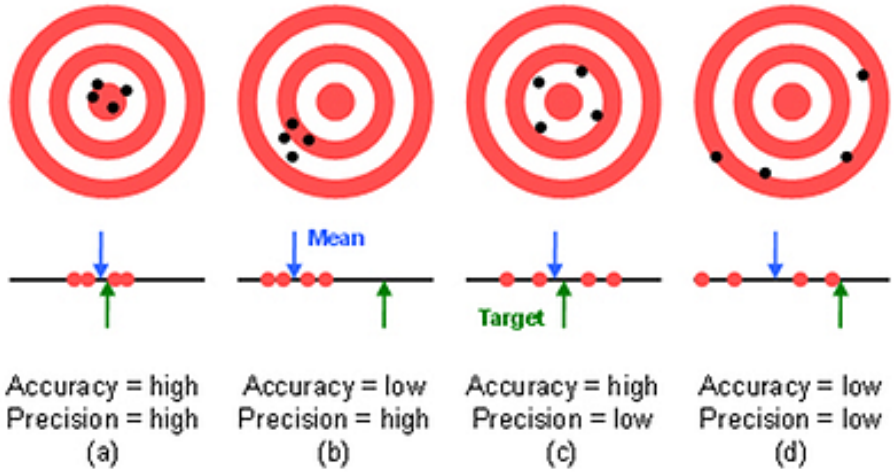


Figure 1.2: Accuracy versus precision.

From time to time sensors require a new calibration due to some changing in parameters that affect the measurements. One cannot expect that a calibration is a one time in device's life thing even if apparently nothing has changed. After the first calibration, every sensor might need a recalibration depending on some working condition factors such as the level of tear, stability and wear the device will

suffer during its operation period and the importance of the accuracy that the application demands. The type of environment the device is exposed can play an important role in the frequency of calibrations. Despite the general problems in calibration regarding sensor own measurements being of most importance, in the case of this thesis, it does not rely directly on that type of calibration. For the purpose of this thesis, the LRF devices are assumed calibrated on their readings. Even if it is possible to do a calibration of the readings using the methods described in this thesis it is not intended for it. Here the focus is on calibrating the measurements between the LRF sensor devices of the network in such a way the measurements from different sensors could have a valid correspondence between them. Given the physical access constraints of ITER, a map description of the environment shall provide the valuable opportunity to check and compare where the LRF measurement data best fits on the map. Therefore the map layout description should be as accurate as possible to mitigate the impact of erroneous walls dimensions or unmapped areas/objects on results.

Some other applications could benefit from the outcome of this work such as house schematic areas measurement confirmation, in industrial facilities warehouses where the need to move containers in a very confined space demands high precision moving operations, keeping track of vehicles or personnel.

1.2 Thesis objectives

The objective of this work is to develop and implement a calibration algorithm to estimate the exact position and orientation of each LRF installed in the environment (see Figure 1.3). The algorithm should return the exact pose, in 2D (3D case is not covered), of each sensor in the network on an absolute frame. The resultant poses are to be used in a localization algorithm in ITER scenario to guide a vehicle and the localization algorithm assumes that the pose of all sensors are known. The type of available LIDAR systems are the planar (or line scanning) ones that return 2D measurements. The devices are detached from the vehicle to be guided. The calibration algorithm assumes a map of the environment is known. Some scenarios may occur where the pose estimate is given and a comparison can be made between the given data and the calibration result. When the position and orientation is completely unknown the algorithm should be fully autonomous estimating the pose. Since it is impossible to test it in a real ITER scenario, a simulator should be developed and results evaluated. Every result must have an uncertainty associated with the LIDAR device specifications.

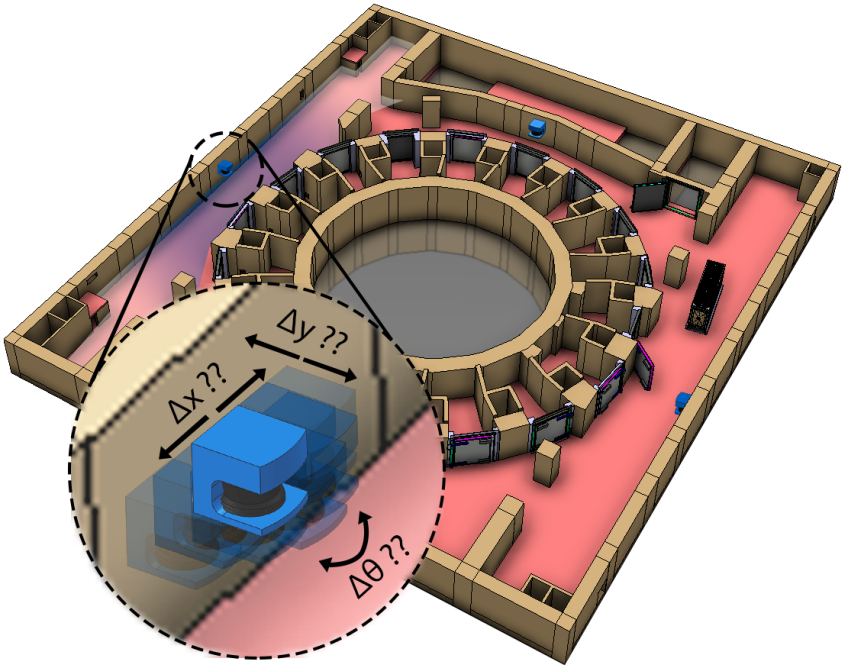


Figure 1.3: Problem illustration in ITER B1 building.

1.3 Problem statement

Besides some of the general problems in sensors calibration, like the ones identified in Section 1.1 above, in this thesis, specially in ITER context, there are other far more important issues. In ITER, the optimal locations where the LRF sensors should be installed are determined beforehand by an algorithm described in [4]. A problem occurs when the sensor real location is different from the supposedly predefined optimal location. As a consequence of sensor misplacement, measurements from distinct LRF lose correspondence between them. Address and solving this problem is the main focus and objective of the present work. This issue, where the exact position and orientation of the sensors are spoiled, can be caused by human error on installation procedure, erroneous map measurements or any other factor derived from sensor operation such as the ones mentioned in Section 1.1.

In worst case scenario, the expected position and orientation where the sensors must be installed could even be unknown. For these cases it is possible that some algorithms (like the vehicle localization algorithm in [4] that assumes all sensor poses are known and heavily rely on the sensor readings accuracy) fail or mislead their outputs because measurements are biased. A simulation was carried in [4] to test and demonstrate this situation where sensors suffer from a small deviation from their correct position. The results in Table 1.1 and in Figure 1.4 both from [4], state that the impact of misplacing the sensor is huge even for low errors and the localization algorithm, where the measurements are used, is not robust to these errors. A deviation of 100 mm in sensor position and 5° in orientation could cause as much as a deviation of about 120 mm (as stated in Table 1.1) and 873 mm respectively in vehicle position. These deviations compromise maneuvers of hazard material transportation vehicles on the cluttered ITER environment.

Table 1.1: Error for different sensor deviations [mm, °].

		$dev = 0$		$dev = 10$		$dev = 30$		$dev = 50$		$dev = 100$	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
EKF	$\|l_{err}\|$	91.2	63.8	100.2	84.0	93.6	78.6	109.2	114.7	119.3	89.8
	θ_{err}	0.11	1.67	0.16	1.56	0.09	1.30	-0.09	1.68	0.30	1.64
PF	$\|l_{err}\|$	46.2	26.6	47.9	27.9	50.7	28.8	54.9	31.5	78.5	39.4
	θ_{err}	0.04	0.66	0.02	0.65	-0.01	0.68	0.01	0.69	0.01	0.76

For the reasons identified in Section 1.1, in ITER, it is not possible to perform maintenance operations with personnel direct intervention on site. Likely, calibration is no exception and as other remote controlled related ITER operations, it can only be made remotely. All available information about the scenario should not be discarded and, in this case, maps represent the most important environment description and could be extremely useful. So a possible solution would be to do a localization procedure of the devices using the available map before operation. The procedure would have to be done on all the sensors across the network in order to mitigate the error derived from the devices misplacement. As a consequence of the proposed solution other problems arise and an important one is the presence of error in the map layout description: errors in walls dimensions and unmapped areas or features that

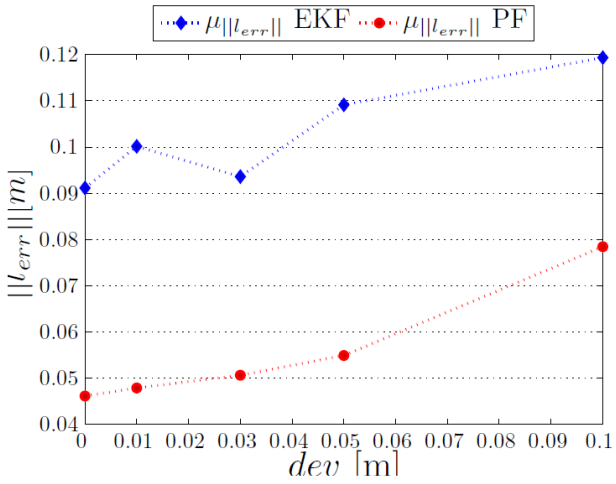


Figure 1.4: Error mean for different sensor placement deviation.

do not have any correspondence in the physical place the map describes. How robust it needs to be to overcome such issues?

The pose P, for a given sensor in the network, includes its position coordinates (x_p, y_p) in meters and its orientation θ_p in angle degrees. Both are given in the map coordinate frame (x_w, y_w) as illustrated in Figure 1.5. The LRF sensors return data in two dimensional polar coordinate system where the distance

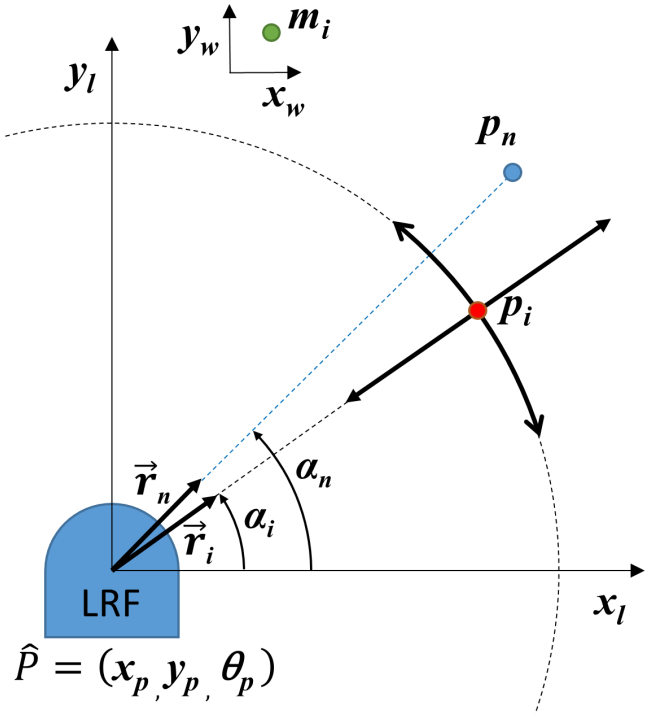


Figure 1.5: Variables involved in the problem description.

measurements $\|r_n\|$ and the associated angle α_i are given in device frame (x_l, y_l). Since the distance measurements are affected by noise, LRF returns $\|r_n\|$ measurements instead of the true $\|r_i\|$ yielding the p_n points. The objective is to estimate the absolute final pose, $\hat{P}_f$, of each sensor in the network that minimizes the mean quadratic error, e_{pp}, function in (1.1), also designated by point to point error,

or p-p error, throughout the text. The residual is composed by the point to point Euclidean distance between the transformed LRF measurements $T(\hat{P}_f, p_n)$ and the respective map closest points m_n. This is a difficult optimization problem due to several local minima existence.

$$e_{pp} = \underset{\hat{P}_f}{\operatorname{argmin}} \frac{1}{N} \sum_{n}^{N} \|T(\hat{P}_f, p_n) - m_n\|^2 \tag{1.1}$$

1.4 State-of-the-art

Calibration of a LIDAR, in this thesis context, can be summarized in determining its pose in a given map absolute frame. Pose determination can be described in other words as a localization problem or extrinsic parameters extraction. Many of the available techniques to extract LIDAR extrinsic parameters could be classified in three different categories: the ones that make use of additional sensors, the ones that use a fixed or moving target and the ones that use their own motion information.

For the first case, on [7] calibration is achieved by freely moving a checkerboard pattern and register a set of planes and lines extracted from LIDAR. A standard extrinsic calibration algorithm (for instance, by intersection of three laser planes) returns the camera pose with respect to LIDAR frame. Some variations of this method include visible LRF rays [8] which consists in making a correspondence between visible dots projected on the camera without using a calibration target. Some other techniques for calibration between a camera and LRF have been proposed in [9].

For LRF calibration techniques using targets, [5] proposes an autonomous calibration process using a carefully designed calibration target with fixed LIDAR systems. Using a pyramid shaped target, the measurements data are used in a RANSAC [10] algorithm to extract line segments and vertices. Pose and orientation estimates are given solving the homography between vertices correspondences. On the other hand, using moving targets, [11] is able to match a persons or robot tracks using ICP point registration method [12] to estimate the sensors pose. Both of the methods have one thing in common: they may require physical access to the environment where the calibration takes place. Since neither hand calibration nor any other method that requires human presence on site is possible, the calibration procedure, like many other operations in ITER context, must be made entirely remotely. The calibration techniques stated before, use static or moving objects (or humans) that must be asymmetrical to uniquely identify the target. That is not the case in ITER as the CTS vehicle features a symmetrical rectangular shape and uses a rhombic-like kinematics configuration for high maneuvering and flexibility making heading extraction not trivial.

For the third case, with moving sensory systems, the Simultaneous Localization And Mapping (SLAM) [13] technique is the most used. Some of the localization techniques that exist today are present in SLAM algorithms due to its importance in the autonomous robotic field regarding path following and task execution. Methods such as particle filtering, despite not being new, have become popular recently since with more powerful computation resources a solution can be achieved in feasible time for this type of application where time is a constrain. That is not the case in this thesis scenario as the map is assumed available, and that is one of the major differences. Also in SLAM, the sensors are calibrated with respect to the robot body frame. LIDAR returns measurement data in its coordinates system which, to be useful, should be transformed in robot coordinate system beforehand. Calibration operation will proceed when no other operation where the LIDAR network is involved is required. Other main difference from the scenarios where localization on robots is applied, is that the readings don't change with time. In ITER, sensors are fixed to walls. When calibration procedure takes place, it is assumed that all measurements are derived from the map so, no fix or moving outliers presence were considered. The only changes in

LRF consecutive measurements should be associated with the sensor precision limits. Kalman Filtering, which is mainly used in SLAM technique, should not suit the best option for this purpose because the relative positions between different scans are known from odometry information.

For the purpose of the present work, the output of LIDAR (which is nothing more than a cloud of points) are the main data resource. Point oriented data process and geometric techniques were the first subject to explore. Clustering (which is a classification technique), feature extraction and geometric alignment are some techniques, described in [14], that belong to this subject. Also, a comparison between some of the most popular feature extraction algorithms in 2D data is presented in [14] concluding Split and Merge could be the best choice for a localization algorithm with an a priori map.

1.5 Thesis structure

This thesis is composed by three main Chapters: the approach and algorithms used in the proposed solution, the simulator developed to test these algorithms and the results where results from different algorithms are described.

On Chapter 2, each subsection identifies a different approach based on the a priori available information about the sensor. Each one of the scenarios identified are taken into account for the general proposed solution. A detailed algorithm description is provided including a discussion about its effectiveness and limits in certain scenarios.

Chapter 3 describes the simulation related procedures implemented. First, and before the development of a simulated LRF model, comes the need to study such sensors in their operation principle to understand the relevance of its properties for the concerned application. LRF scan procedure is the core of the simulation process and its implementation in the simulation environment, as well as maps and sensor properties definitions are explained.

Results in Chapter 4 are divided into two categories: the simulation results using ITER and other created test maps, and the field tests results. The later using three recreated real maps from distinct environments. Two different commercial LRF devices are tested. The noise effect, model parameters and a priori information are subject of comparison in the battery test performed. A robustness section is also presented for testing the solution behavior in the presence of map misspecification and readings outliers.

Finally in Chapter 5 a discussion of the obtained results is presented in conclusions alongside the future work for some unexpected problems that have been identified and other non-approached problems.

In the appendix some figures are shown to complement the results.

Chapter 2

Proposed Solution

The proposed solution is a main algorithm that receives as input a map description, LRF measurement data, and an optional initial pose estimate $\hat{P}_i$ and returns a final pose estimate $\hat{P}_f$ as depicted in Figure 2.1. Note that a worst case scenario where the initial pose is completely unknown is addressed. The uncertainty degree associated with this parameter leads to four different scenarios (denoted A, B, C, and D described below) that specify the way the algorithm behaves. In case an initial pose estimate is given (scenario A), the algorithm behaves locally trying to extract the best pose for the given measurements. Instead, if an initial pose estimate is not given at all (scenario D), the algorithm behaves globally on the map, assuming any pose is plausible. Other two intermediate scenarios (B and C) may occur whenever one parameter of the initial pose is missing. In scenario B the position estimate is known and orientation is unknown. In scenario C the opposite happens, orientation estimate is known and the position is unknown. Since position is composed by two parameters (x and y coordinates), the case where one of these parameters is unknown is treated as a D scenario.

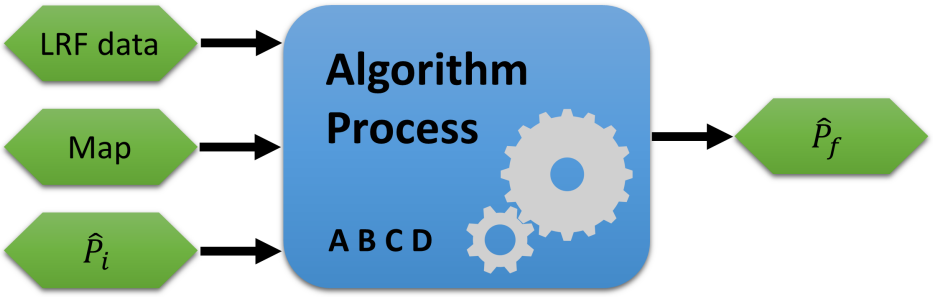


Figure 2.1: Main algorithm overview.

2.1 Data pre-processing

This first phase in the algorithm execution is common for every scenario. Since it is possible to mitigate the errors in readings derived from the random errors in LRF devices, this step is responsible to process raw data coming from the LRF device in order to improve the distance measurements results. By av-

eraging data from multiple scans, precision improves proportionally to the square root of the number of complete scans as stated in [15]. For instance, if one hundred scans are taken, the precision might be ten times higher than using directly raw data. But before averaging, a normality test takes place to verify if data fits well a modeled normal distributed population so an averaging metric could be meaningful in this context.

2.1.1 Normality test

Normality tests are used to test and evaluate how well the sample data in question is modeled and fits a normal distribution population. If that is the case the mean is an acceptable measure of central tendency. This property is useful in localization algorithms [16]. In that respect some tests like Kolmogorov-Smirnov (K-S), Lilliefors [16] or Geary's tests used in [17] could be used to verify this property. The difference between these tests is, in a simple way, the distribution parameters (mean and variance) specification. K-S tests a complete specified normal distribution, this means the mean and variance should be given, where Lilliefors tests for normality with unspecified parameters. The Geary's test is based on a ratio between two estimators of the sample standard deviation. Lilliefors was the test chosen because parameters can have any value within the LRF device valid maximum range interval. These tests results do not limit the algorithm behavior, they are used just for confirmation purposes. Also, tests in [18] and [19] show that measurements from real LRF devices follow a Gaussian pattern.

2.1.2 Data averaging

The averaging method implemented uses a significant number of scans (at least one hundred [15]) to determine the average value μ, and the standard deviation σ for each respective measurement. The raw averaging metric alone might consist in a naive approach to assess the most likely measurement value because average metric is strongly influenced by extreme values. To reduce this negative influence, a window interval, w, given by (2.1) was used to reject absurd values.

$$w = [\mu - \sigma, \mu + \sigma] \tag{2.1}$$

After some experimentation, only the values within σ (68%) range away from the average μ were chosen to mitigate the extremes value influence. A new calculation of average value μ_w is handled, this time, for the values selected by the window, seeking this way the achievement of better results. The process is repeated for every angle data array. The resultant μ_w, given by (2.2) is the value taken for figuring in the final readings output.

$$\mu_w = \frac{1}{N_w} \sum_{i=1}^{N_w} d_i \tag{2.2}$$

2.2 Scenario A: initial pose known

In this situation, the initial pose estimate (position and orientation) is completely known, and, therefore, a local based search is performed using LRF readings against the map. The algorithm will try to guess the best match using the measurements the device returns against the map near the given position coordinates and orientation which should be given in the map frame. The origin of the LRF points are translated to the given position coordinates and rotated according to the given orientation. Then, a matching procedure takes place to adjust and align the LRF points with the map points to determine the best fitting pose. To match the two point clouds and evaluate the displacement error, the Iterative Closest Point [12] (ICP) algorithm was chosen to perform the task.

2.2.1 ICP algorithm description

ICP is an algorithm for calculating the best match between two clouds of points. Since the map is composed by a set of line segments, the first procedure consists in transforming the map in a set of points. The number of map points should be chosen accordingly being, at least, bigger than the number of measurements set and should be, also, uniformly distributed along the map line segments. To preserve map data integrity as much as possible, the map line segments need to be transformed into a significant number of points. The map points, m_i, are chosen as the reference and LRF points, p_i, are subject to a rigid body transformations R, t that aligns the two by minimizing the quadratic error of the point to point distance metric at each k iteration for the all the N LRF points. The algorithm stops when the point associations between the k and $k-1$ iterations are the same. Other stop condition was tested and the results are described in the Appendix Section A.2. The resultant output, R_k and t_k, is the rigid body transformation that best fits the two point clouds and is determined from SVD decomposition [20]. By applying the result to the initial pose guess, the final pose estimate is revealed. One major drawback in this algorithm is that the ICP can be trapped in local minima and this is the main reason why it has been chosen for local search scenario. Variations of this algorithm that matches points to lines such as [21], [22] and [23], have been proposed, but the error metrics require nonlinear minimization metrics which are solved using approximations and are not robust against large initial displacement error. Point to point metric presents a closed form solution and a faster convergence despite time not being a priority in the context of this application. If the solution presents an associated p-p error with a higher order of magnitude than the devices standard deviation (STD) value, other case scenarios may be applied. In these other scenarios where the initial pose estimate is incomplete or non-existent, the resultant pose estimate always suffer a final ICP alignment.

The algorithm works as follows:

1. Find the closest points pairs using Nearest Neighbor Search (NNS) algorithm between the two given point sets;

$$M = \{m_1, m_2, ..., m_n\}, \tag{2.3a}$$

$$P = \{p_1, p_2, ..., p_N\} \tag{2.3b}$$

Where:

m_i: map/model point

p_i: scan/scene point

n, N: the number of points in the respective set. $n = N$ is not mandatory

After finding the pairs, two sets of corresponding points of length $l = N$ are obtained:

$$M_l = \{m_1, m_2, ..., m_l\} \tag{2.3c}$$

$$P_l = \{p_1, p_2, ..., p_l\} \tag{2.3d}$$

2. Calculate the alignment;

   - The objective is to find the translation vector t and the rotation matrix R that minimizes the point to point error function:

   $$R, t = \underset{R,t}{\operatorname{argmin}} E(R, t) \tag{2.4a}$$

   - Point to point error function is given by the sum of the squared error:

   $$E(R, t) = \frac{1}{N} \sum_{i}^{N} \|Rp_i + t - m_i\|^2 \tag{2.4b}$$

   Where:

   p_i and m_i are the corresponding closest points given by NNS.

   - Closed form solution

   Rotation and translation matrices are given by:

   $$R = UV^T \tag{2.4c}$$

   $$t = \bar{m} - R\bar{p} \tag{2.4d}$$

   Where:

   $$\bar{m} = \frac{1}{l} \sum_{i}^{l} m_i \tag{2.4e}$$

   $$\bar{p} = \frac{1}{l} \sum_{i}^{l} p_i \tag{2.4f}$$

   Are the centers of mass of the two point sets

U, V are given by SVD decomposition of the following matrix W:

$$W = \sum_i (m_i - \bar{m})(p_i - \bar{p})^T = USV^T \tag{2.4g}$$

Note: rank $W = 3$ is mandatory

3. Update scan points and repeat until stop condition is reached;

   - Apply rigid body transformation to scan points:

$$p_i' = Rp_i + t \tag{2.5}$$

   Where:
   p_i' is the new set of aligned scan points

   - while $M_l^{k+1} \neq M_l^k$ apply a iterative process:

$$R_k, t_k = \underset{R,t}{\operatorname{argmin}} \frac{1}{N} \sum_i^N \|Rp_i^k + t - m_i^k\|^2 \tag{2.6}$$

   where:

   $p_i^k = R_{k-1} p_i^{k-1} + t_{k-1}$

   k is the iteration number.

2.2.2 ICP complexity

ICP complexity is bounded by NNS algorithm. If no space partition method is applied, like k-d tree, the algorithm tries every point of the matching cluster with every point of the base cluster (linear search) so time complexity is given by:

$$\mathcal{O}(M \times N) \tag{2.7}$$

If k-d tree is applied [24], the algorithm moves down the search tree recursively dividing search space into different regions. In this case time complexity is given by:

$$\mathcal{O}(\log N) \tag{2.8}$$

2.3 Initial pose partially known

When the information of one of the variables is not available, whether it is the position or the orientation, the information available should be taken into account when searching for a solution. The available information is determinant for the methods developed for the two different following scenarios.

2.3.1 Scenario B: only position known

In this case only a position initial estimate is known, making the orientation missing. Taking advantage of the available information provided from both the position and map together, the proposed solution consists in an ICP brute force approach. For brute force it means the ICP is initialized many times, each time using a different initial pose, from a predefined set of poses where only the estimate orientation angles vary, and the position is maintained the same. So it is better called ICP orientation brute force. Defining the initial set of orientation values as low as possible apart is important as well because ICP algorithm showed a higher chance to converge to the optimum solution without being stuck in a local minima. After some experimentation, ICP has a higher chance to converge to optimum solution, if, at best, angles are defined from a 30 degree maximum deviation apart. Based on this concept, a value below 30 degree is advisable. A possible heuristic consists of beginning with an orientation angle approximately equal to nearest wall normal. The final pose estimate is given by the pose with the lowest e_{pp} associated.

2.3.2 Scenario C: only orientation known

In this case, an orientation initial estimate is given, making the position estimate missing. Position possibilities in a map, could lead to a very high number of combinations to use in a brute force ICP. Despite being possible, the successive ICP could become very computationally heavy with no guarantee of a true pose convergence. As an alternative, a novel method was developed consisting in using the readings as a projection of the possible location of the sensor. Every single LRF measurement is projected backwards from each map point revealing a LRF possible location. A vote is then accumulated for the respective coordinates. The coordinates which have a higher vote count determines the most likely position estimate. A heuristic for this situation consists of select only walls whose normal angles are approximately equal to orientation angle given.

Voting method description

The voting algorithm is a democratic algorithm uses the readings as a projection of the possible location of the LIDAR. The map is first discretized, this time, into a grid of bins. Using the information returned by a LRF readings, each measurement can be described by a vector $\vec{r}$ from the LRF position (origin in LRF frame) to a point given by the distance and respective angle. The map is once again discretized in points and for a given map point all of the measurements are taken backwards from this map point to a point of the possible location of the LRF In other words, the inverse vector $\vec{r}$ is added to the map point resulting in a prediction point of LIDAR position. This process is repeated for every map point and for every LRF measurement. The resultant points vote each one for a discrete bin and the bin with more votes dictates the final pose estimate.

The algorithm works as follows:

1. Divide map into a grid of bins;

2. Given the laser output distances array and corresponding angles array O, and map points M, the prediction points for a fixed value of θ is:

$$p_{i,j} = m_i + (-\vec{r_j}) \tag{2.9a}$$

3. Increment respective bin vote count for the correspondent results p;

4. Repeat $\forall i \in M \wedge \forall j \in O$;

5. Final pose estimate $\hat{P}_f$ is given by the mode of the bins coordinates (bin with the highest vote count);

Voting complexity

For every readings points P_i, a vote is done for every map point M_i by backward projection. Time complexity is then given by:

$$\mathcal{O}(P_i \times M_i) \tag{2.10}$$

2.4 Scenario D: initial pose unknown

This is the worst case scenario as the LRF location can be anywhere in the map. Neither position or orientation initial estimates are known, leading to an almost infinite number of possible pose combinations. What is known for sure is that the position must belong to an area bounded by map walls. The devices are installed on the walls (in ITER case) and they should be installed approximately according to the wall normal. Assuming the previous constraints, the search space could be reduced, but, in any case, even taking into account the installation in walls constrain, depending on the size of the map and in the density discretization of it, a brute force ICP could become impractical in terms of time given the huge number of hypotheses. Nevertheless, the method was applied for experimental purpose and is described below. In order to reduce the search space, when compared to the previous solutions, a different approach was taken. The geometric features method presented does not take into account the wall installation constrain and is divided in two phases: feature extraction and matching features. A flowchart is provided on Figure 2.2, which depicts an overview of the following different phases and main steps involved in this case scenario.

The objective in the feature extraction phase is to extract line segments and vertex points from the processed LRF output data. These features may contain useful and rich information about the environment. A comparison between some of the most common line extraction methods is present in [14]. Based on the conclusions, the most correct and best suited method for the localization problem with an a priori map was the Split and Merge technique preceded by a clustering algorithm. So the first step is to perform a primary classification of points using a clustering technique.

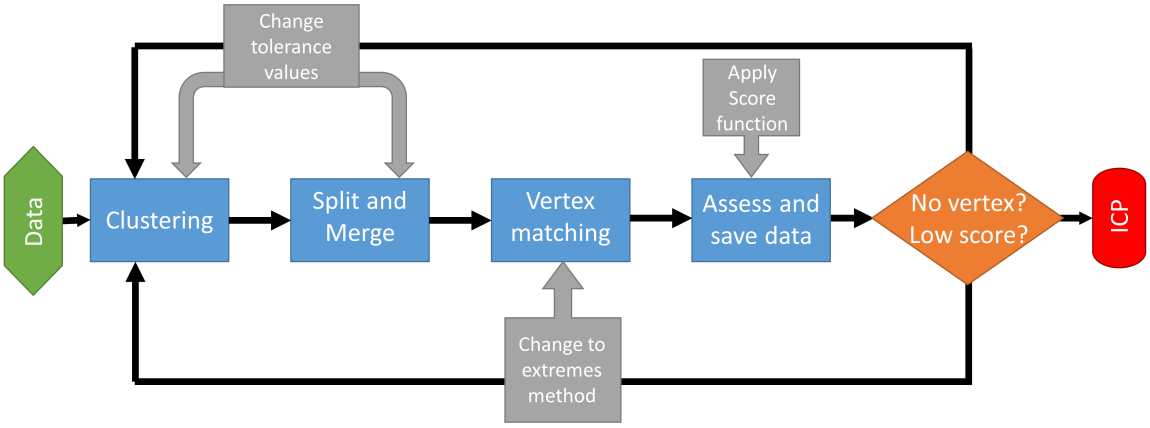


Figure 2.2: Flowchart of scenario D.

2.4.1 Clustering

Clustering is a classification technique used to identify groups of elements that share similar characteristics and patterns. It leads to data reduction in a sense that elements are represented by its cluster and enables the classification prediction of new samples to be made based on existing groups. In the particular case of this work, point coordinates formulate the data elements. The application of this technique allows for a trial of the data in the first place, testing whether a discontinuity exists between two consecutive points, leading, in a later stage, to a distinction between different walls. It takes advantage of the sequentiality of the data points and groups of points can be formed based on the distance between them.

DCC algorithm

The simple solution, of classifying clusters based on a threshold distance value between consecutive scan points alone is dangerous because it can lead to erroneous cluster formation. This represents a problem for very oblique walls as the points may belong to the same wall but many clusters, composed by only one or two points, could be formed for the same wall. To avoid this problem, the Distance based Convolution Clustering (DCC) method is proposed by [25]. It consists on identifying break points based on a sudden change of the distance between consecutive scan points, and once one is found, a new cluster is created. To do that, a high pass filter is applied to the set of Euclidean distances between consecutive points. The break points are identified where the convolution is greater than a cluster threshold value which is proportional to the LRF readings standard deviation.

The algorithm works as follows:

1. Given the laser dataset P:

$$P = \{p_1, ..., p_n\} \tag{2.11a}$$

Compute the vector of the Euclidean distances between each pair of consecutive points p of the laser dataset.

$$D = \{d_1, ..., d_{n-1}\} \tag{2.11b}$$

2. Compute the discrete convolution:

$$(d * k) = \sum_{j=-\frac{m-1}{2}}^{\frac{m-1}{2}} d(i+j) \cdot k(j) \tag{2.11c}$$

Where:

- σ is the laser standard deviation;
- k is a high pass convolution kernel of size m (m is necessarily an odd number)

$$k[-\frac{m-1}{2}, ..., \frac{m-1}{2}] \tag{2.11d}$$

Suggested kernels by the authors:

$$k_1 = [-3, -3, 5, -3, -3] \tag{2.11e}$$

$$k_2 = [-1, -2, -3, 5, -3, -2, -1] \tag{2.11f}$$

3. Identify the breakpoints and create a new cluster when:

$$(d * k) \geq b\sigma \tag{2.11g}$$

- b is a tuned parameter based on σ and distance to obstacles. Suggested b value, by the authors, when $\sigma = 1cm$:

$$b = 5 \tag{2.11h}$$

Robust line regression

Line extraction is a method for transforming the raw scan points into geometric features, in this case line segments, which, among geometric primitives, are the simplest ones. It can be used to extract corners as well. The extracted features are then used for matching purposes in the next stage. Geometric features require much less memory because their representation is much more compact. Take for instance a wall that is represented by a line segment consisting of only two points, instead of a cluster of points taken from the scan. Therefore, they could be more efficient compared to point based algorithms.

There are three main problems in line extraction in an unknown scenario [26]:

- How many lines are there?
- Which points belong to which line?

- Given the points that belong to a line, how to estimate the line model parameters?

LIDAR measurements are given in polar coordinates which, recalling to basic algebra, can be converted to Cartesian coordinates. The geometric shape of the scanned surroundings is revealed when the converted points are plotted in the Cartesian plane. Depending on the map, but in most cases, these points seem to have a relation between them. For instance, some points together can form a line shape which can be associated with a map wall. Regression is a technique used to test the relation between a dependent and independent variables. In this case, since walls are assumed to be represented by line segments in map description, linear regression and modeling techniques seem the most appropriate methods to estimate and extract the relation between a given set of points. Ordinary least squares (OLS) method is the most popular linear regression model to estimate the line that best fits a set of points. The drawback in this kind of model is the lack of robustness against outliers. Robust line regression methods come in answer to the outliers problems. These methods attempt to reduce the influence of outliers when the line regression takes place. In Figure 2.3 a comparison of robust line regression against OLS is performed, the example was taken from [27].

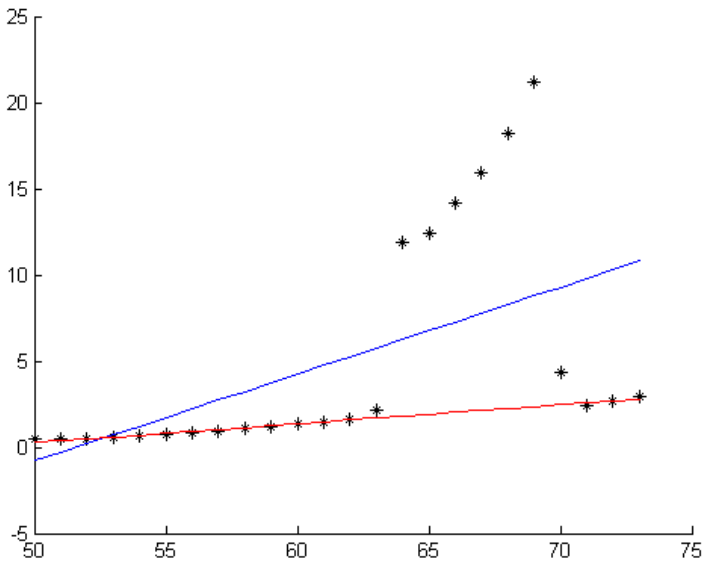


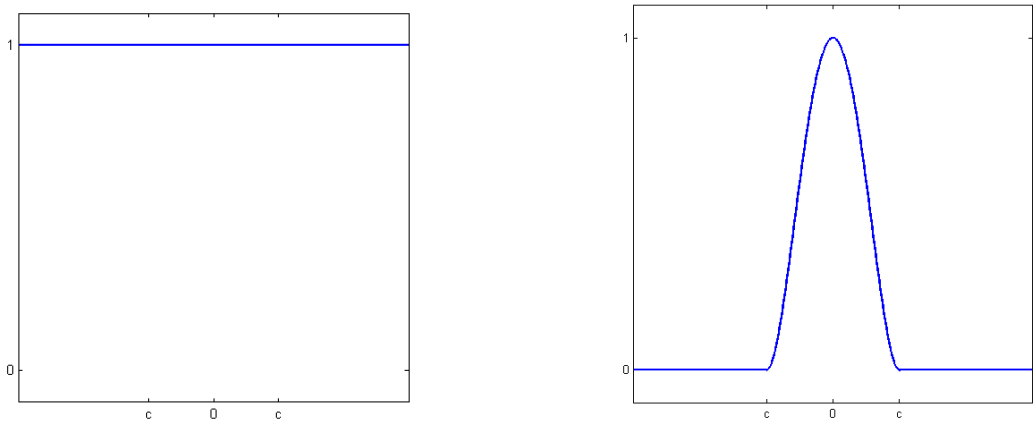
Figure 2.3: Robust line regression in red, OLS line regression in blue.

The class of robust estimators used is the maximum likelihood type estimators, or M estimators [28]. The main difference between them is where OLS tries to minimize the squared residual (2.12), the class of M estimators tries to minimize other function of the residual (2.13) that present particular properties.

$$min \sum_{i=1}^{n} r_i^2 \tag{2.12}$$

$$min \sum_{i=1}^{n} \rho(r_i) \tag{2.13}$$

For $\rho(r)$ function to be considered a robust M-estimator, it must accomplish the following properties:



(a) Ordinary least squares weight function.

(b) Bi-square weight function.

Figure 2.4: Weight functions used.

- Definite positive with unique minimum at zero: $\rho(r) \geq 0 \; \forall \; r$;
- Symmetrical: $\rho(r) = \rho(-r) \; \forall \; r$;
- Must increase slower than r^2: $\frac{d\rho}{dx} \leq \frac{dr^2}{dx}$

The derivative of ρ is called the influence function and is defined as:

$$\psi(x) = \frac{d\rho(x)}{dx} \tag{2.14}$$

And the respective weight function is defined as:

$$w(x) = \frac{\psi(x)}{x} \tag{2.15}$$

The weight function represents the influence a sample is given regarding its distance to the M-estimator (distance to middle of the data set). There are many candidates for a ρ function. The one used was the Tukey bi-square function (see Figure 2.4b) together with the concept of an iterative reweighed measure of central tendency. The Figure 2.4 depicts the two weight functions, with OLS weight function, given by 2.16, on the left and Tukey bi-square function, given by 2.17, on the right.

$$w_{OLS}(x) = 1 \tag{2.16}$$

$$w_{bi-squared}(x, c) = \begin{cases} (1 - (\frac{x}{c})^2)^2 & \text{if } |x| < c \\ 0 & \text{otherwise} \end{cases} \tag{2.17}$$

Looking at Figure 2.4, for OLS, the same weight is given for every sample regardless the distance from the samples to the model, where for bi-square, as far as the sample is, the lower is its influence to the model. For bi-square, past the cut-off user defined c value, the weigh value is zero. This property comes in handy when extracting lines from samples with a few but huge discrepant values. The value for c can be chosen according to the LRF system measurement uncertainty.

2.4.2 Split and Merge segmentation algorithm

Wrong or missing breakpoint identification can happen since clustering technique is a simple and primary approach. The second step in the geometric features method uses the Split and Merge (SM) [29] algorithm to confirm and correct for those erroneous cases in order to finally extract the definitive lines and vertex points. This algorithm is composed by the split phase and the merge phase (see figure 2.5). At each phase, line fitting methods are used. The SM technique begins by the splinting phase where the objective is to create sub-clusters of a cluster based on the evidence of a line pattern points. A line given by the two extreme points of the cluster is extracted and a breakpoint can be identified if the most distant point to the line is greater than a given split threshold. If not, the line is extracted using the robust line regression identified above. Since the previous procedure can wrongly create more than one line segment for the same correspondent line in the map, a merging phase is conducted, afterwards, to unify similar clusters. In the merge phase, if two neighbor line segments have its respective, angle difference inferior to a given slope threshold, and the distance between two extreme points inferior to a proximity threshold, the two respective clusters are unified. The segment line parameters, for this unified cluster, are extracted, using a M estimator robust line regression technique with a bi-square weight function. Vertex points are extracted extending and intercepting consecutive and close enough line segment extremes.

The algorithm works as follows:

1. Split
   (a) Get the line that passes by the two extreme points of a given cluster composed by at least three points;
   (b) Search for the most distant point to the line (break point);
   (c) If the distance is superior to a predefined threshold, split the cluster in two at the break point and repeat for the two new clusters.
   (d) If the distance is inferior to a predefined threshold, obtain the line equation that best fits the points using robust line regression technique, and move on to the next cluster;
2. Merge
   (a) Get the neighbor segments
   (b) If two neighbor segments have its respective line slope parameter difference inferior to a given threshold and extreme points are close enough (inferior to a given proximity threshold), merge the two respective clusters and extract the line that best fits the new set of points using a M estimator robust line regression technique;

2.4.3 Vertex method description

The matching phase takes advantage of the geometric features extracted previously, specially the vertices. An extraction of map vertices is first performed, and then, a matching hypotheses is formed for every pair combinations of map vertices and extracted vertices. Assuming a vertex is the common extreme

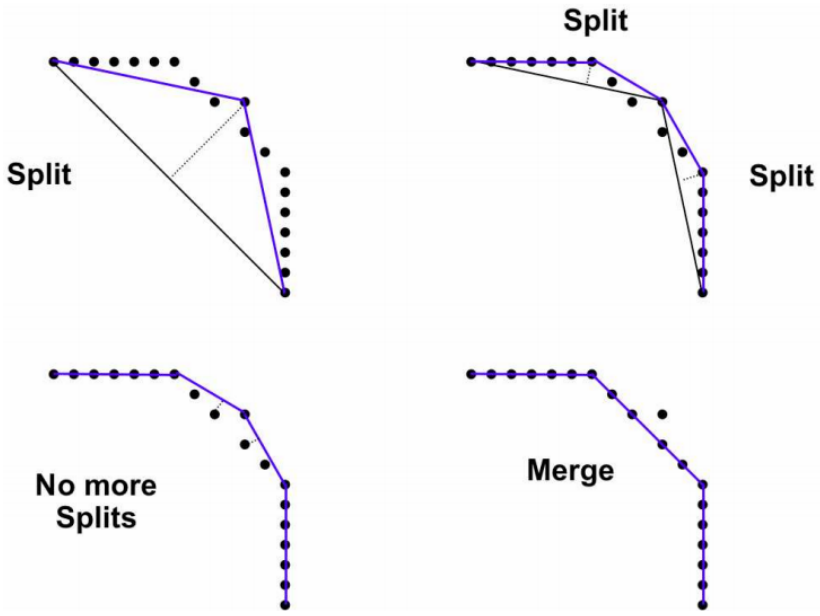


Figure 2.5: Split and Merge example.
Picture by J.Tardós.

point of two different line segments, after overlapping the two vertex points in the pair by a translation transformation, there exists a four possible pair combinations of alignments for the vertices respective line segments. This alignments are done by means of rotations. The resultant rigid body transformation is applied to all LRF points and after this transformation the p-p error function in (1.1) is used to assess and evaluate the displacement between the two point clouds. This time, the displacement error between the vertex points, e_v, can be introduced on the score of the vertex pairing, vp, and line pair association, lp, in (2.18). The resultant estimated pose, $\hat{P}_f$, is determined from the transformation that yields the highest score.

$$score_{vp,lp} = \frac{1}{e_{pp} + e_v} \tag{2.18}$$

In case no vertices are extracted, threshold values that affect the feature extraction phase are changed and the procedure is repeated. If the combinations of tested threshold values are not enough to produce a valid result, i.e., a result which the respective point to point error value is relatively close to the device standard deviation, an extremes matching method is applied. This method procedure is similar to the vertex method, except it uses end points of extracted line segments instead of vertices. For every pair of points (one map vertex and one extreme point from an extracted line) a two combination matching (instead of four) is done by aligning the line segment with the two map lines that originate the map vertex in question. At last, whichever the method (extremes or vertex) that was applied, like in the previous scenarios, the estimated pose, $\hat{P}_f$, that resulted from the best score, is used as a $\hat{P}_i$ in the final ICP alignment.

The method proceeds as follows:

1. Extract vertex points from map and from LIDAR readings;

2. Extract adjacent line vectors for every vertex;

3. Match every map vertex with extracted vertex by:

   (a) Align vertices by means of translation;

   (b) For every possible combinations of vectors pairs align each vector pair by means of rotation;

   (c) Evaluate the resultant rigid body transformation;

4. Choose the rigid body transformation which yielded the highest score.

Vertex complexity

For every extracted vertex V_e, a matching with every map vertex V_m is tried by aligning the corresponding line segments. Complexity is then given by:

$$\mathcal{O}(V_e \times V_m \times 4) \tag{2.19}$$

2.4.4 Brute force ICP search algorithm

The brute force ICP test consists in applying the ICP algorithm for a set of predefined poses. Both initial position and orientation values vary between consecutive ICP algorithm executions. To define the set of poses, the map is divided into a grid, using a given resolution, and for each point of the grid, a set of orientations are tried. In overall, this method is computationally heavier than the previous one. Also, it presents no guarantee of converging to the true solution, and has the potential to not be feasible in terms of time, both depending mainly on the resolution value chosen.

2.5 Solution brief review

A block diagram of the main algorithm is presented in Figure 2.6. With the help of this diagram, the algorithm is summarized as follows: the algorithm begins by receiving the user input initial pose estimate, the scan data from one LRF device, and the map description of the environment. The following stage consists in determining the scenario and is composed by one decision block that identifies if the given pose is completely defined (scenario A), partial defined (scenario B or C) or totally undefined (scenario D). In case the pose is partial defined, if only the position is given, scenario B is applied, otherwise if only the orientation is given, scenario C is applied. In scenario A, since the pose estimate is completely defined, the ICP algorithm is applied right after. On the remaining scenarios the pose needs to be completed before the ICP final alignment. To complete the pose initial estimate in Scenario B, a brute force ICP is applied. In scenario C the voting method is responsible for determining the position estimate and in Scenario D the vertex (or extremes) method yields the complete pose initial estimate. The final step applies the ICP algorithm revealing the final pose estimate.

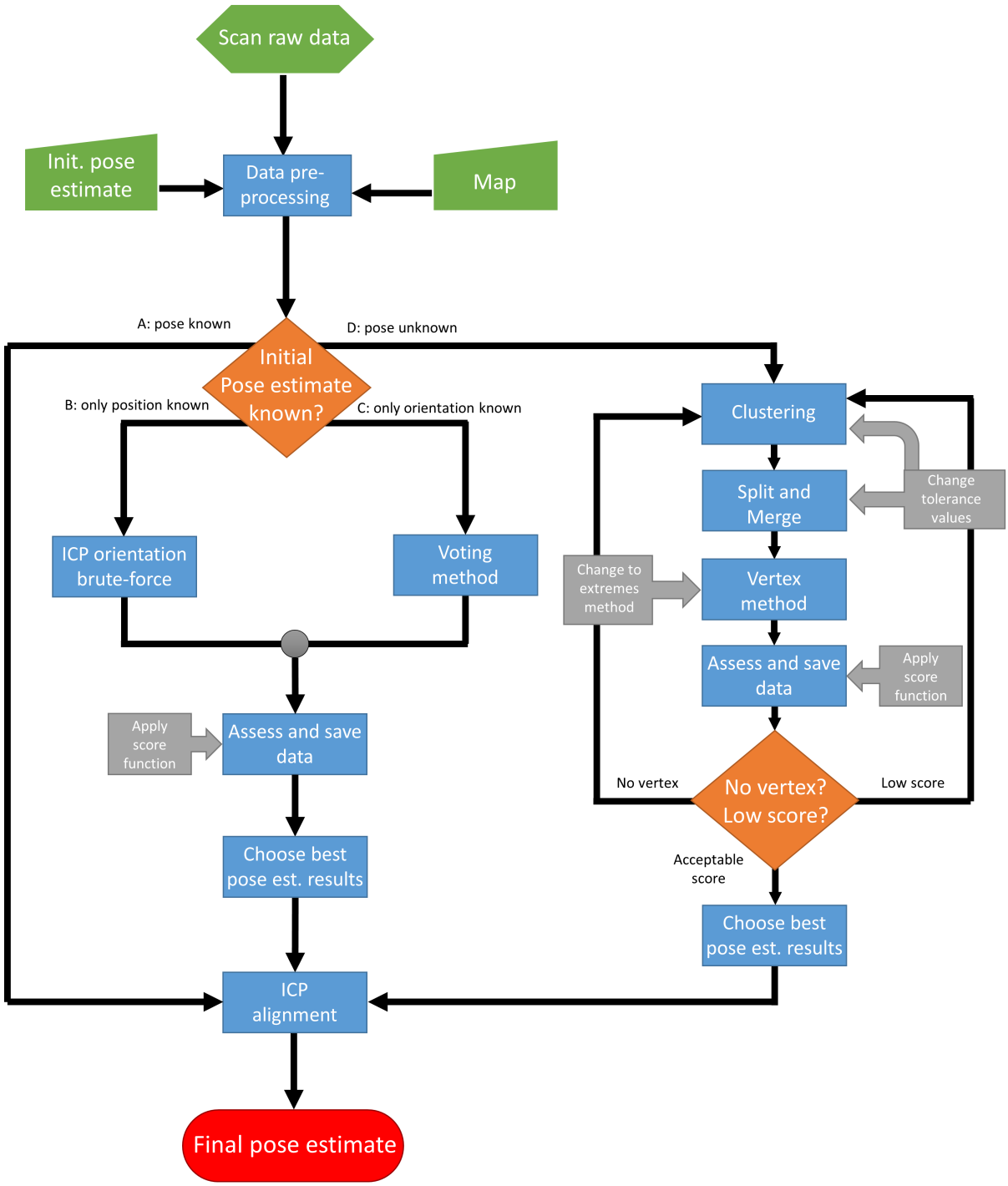


Figure 2.6: Major algorithm block diagram.

Chapter 3

Simulation

3.1 LIDAR operation principle

Laser Range Finder devices are active non-contact absolute distance measurements sensors. Usage of these sensors cover wide range of areas but in most cases, the main interest is their application in industrial environments [15], especially in the field of productivity and quality of products. Some of the applications include atmosphere measurements of cloud heights and meteorological statistics [30], ocean and lakes on water surface roughness, earth surface and vegetation using measured reflectivity data and structure topology.

The main characteristics that distinguish different LRF types and models and their best suited application comprise their power, maximum range, scan and acquisition rate, and measurements directions (two or three dimensions). For long distances, up to several kilometers, a Nd:YAG laser [15] can be used, giving peak power to the megawatt level. These are usually military range lasers. Below comes the pulsed laser diodes, with a single or double-heterostructure which are capable of reading up to hundreds of meters but, unlike the previous, the measurements rate is much higher, reaching the megahertz level. The most common and cheaper, and the ones addressed in the present work, are electro-optical laser measurement systems that electro-sensitively scan the perimeter of its surroundings in its field of view (FOV). It is done in a circular manner, on a plane (2D) with the aid of laser beams (see Figure 3.1 taken from [31]). Laser beams are sent at a regular angular steps using an angular encoder.

The basic idea behind a LRF is the time of flight distance measuring technique. It consists in measuring the time an optical signal (pulse of light energy) takes to travel from the sensor emitter to the target and the reflected signal to travel back from target to the sensor receiver. After receiving the reflected signal, half of the measured time between the transmission and the reception (t_t) is used to calculate the distance, d, based on the velocity of light c (roughly $c = 30cm/ns$), using (3.1).

$$d = \frac{ct_t}{2} \tag{3.1}$$

Other distance sensors, like sonar, use the same idea, the difference is in the signal used whereas

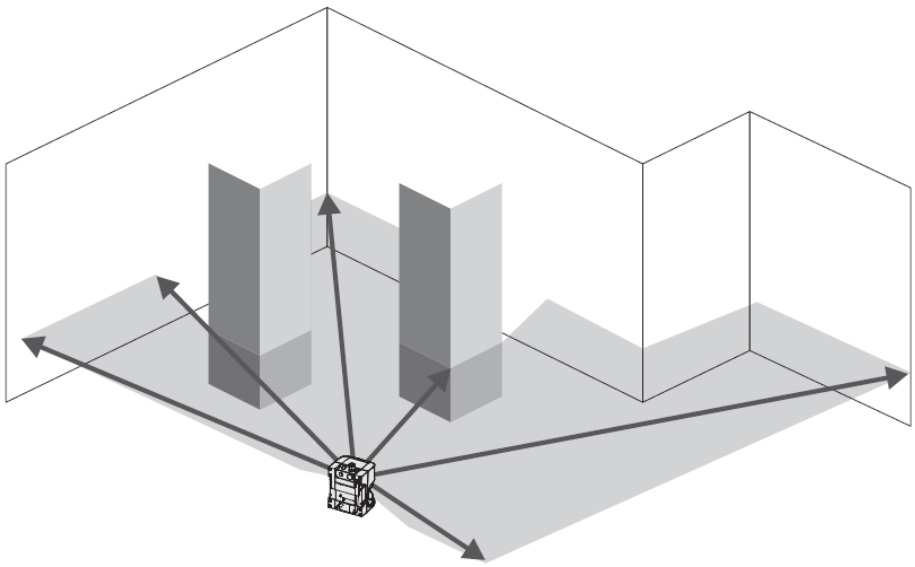


Figure 3.1: Scan measurement principle.

instead of using an optical signal, sonar uses a ultrasonic signal. When high resolution measurements are a constraint, LIDAR takes the advantage because ultrasonic wave beam isn't as narrow as optical wave leading to a lower accuracy. The capability of the beam to maintain its focus and profile changes as it propagates in free space. Since in the output cavity of a laser device is an aperture, the beam width behavior is governed by the law of diffraction. For a Gaussian laser beam profile (see Figure 3.2a from [32]), the electric field and radiance are approximated as Gaussian functions, which the majority of lasers (like the one in Figure 3.2b from [33]) typically emit. In this case, the law of diffraction is given by (3.2).

$$w(z) = w_0 \sqrt{1 + \left(\frac{z\lambda}{\pi w_0^2}\right)^2} \tag{3.2}$$

For a given wavelength, typical near the IR range, the radius of the beam spot at distance z is given by $w(z)$. The minimum value is w_0 at $z = 0$ which corresponds to the radial size of the beam in its narrowest point known as beam waist (see Figure 3.3 taken from [34]).

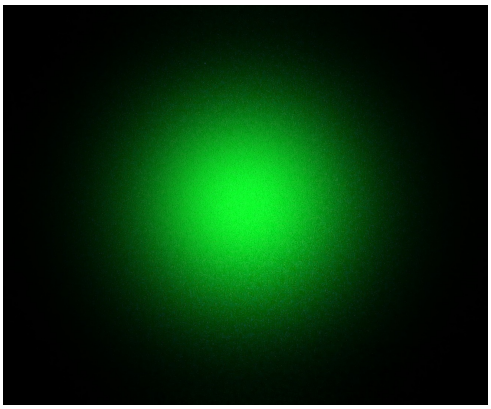
The Rayleigh length is the distance from the waist to the place where the area of the cross section is the doubled. For a Gaussian beam the Rayleigh length is given by:

$$z_R = \frac{\pi w_0^2}{\lambda} \tag{3.3}$$

The confocal parameter, or depth of field, using the Rayleigh criterion gives:

$$D_f = \frac{2\pi w_o^2}{\lambda} \tag{3.4}$$

When the wavelength is small and beam wait is large, the depth of field increases. For high values of z



(a) Gaussian laser profile.

(b) Diode laser commercial available.

Figure 3.2: Diode laser characteristics.

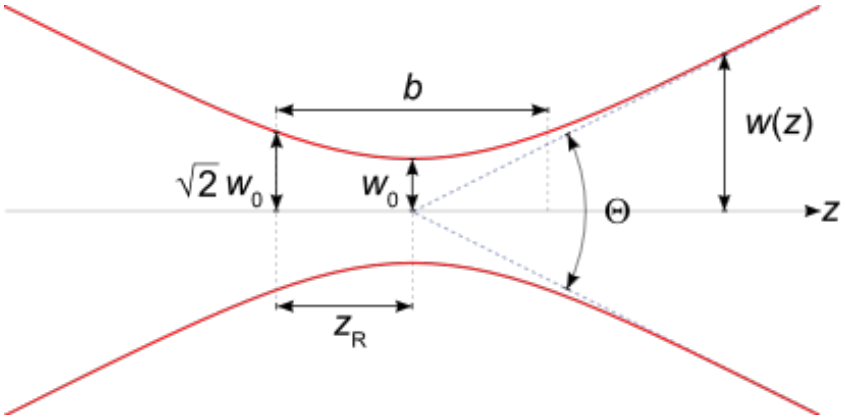


Figure 3.3: Beam spot size geometry.

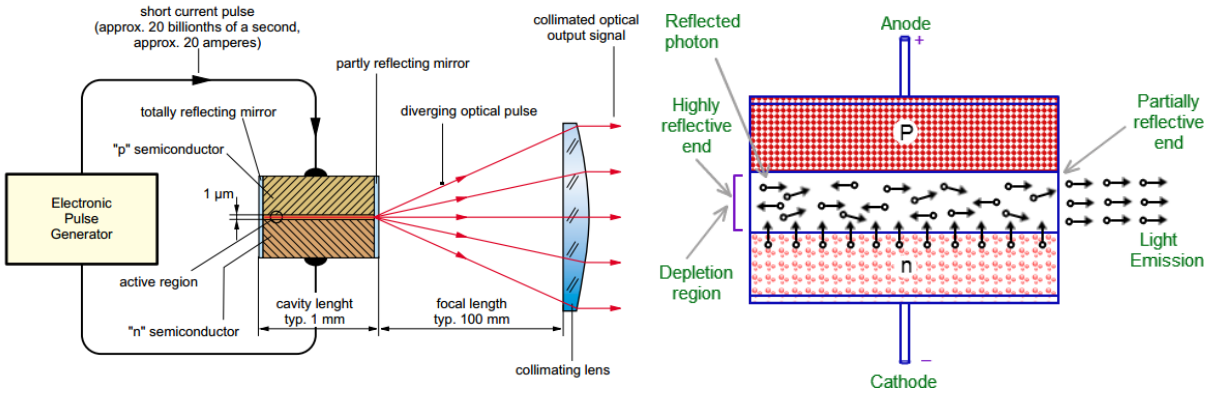
and small values of beam divergence, θ, its value is given, in radians, approximately by:

$$\theta \approx \frac{\lambda}{\pi w_0} = \frac{\Theta}{2} \tag{3.5}$$

In the heart of a LRF system is the semiconductor laser diode (see Figure 3.2b). An electronic pulse generator drives the laser by sending, short in time but powerful, current pulses across the semiconductor. The semiconductor has two of the faces cleaved so they are flat and parallel thus forming the two mirrors of the laser cavity to reflect photons back on track along their path (for more detail see Figures 3.4a, 3.4b both taken from [35] and Figure 3.5, taken from [36]). The electrical energy is then converted to optical energy in the p-n junction, a very narrow active region, that, when forward-biased, electrons recombine with holes and the photons are released with wavelength given by (3.6). An avalanche effect takes place when the movement of photons strike atoms releasing, this way, even more photons. This conversion from electrical energy to light energy is called electroluminescence. The light pulses produced have a duration of 5 to 50 ns and are collimated by the transmitter lens which is responsible for creating a very narrow beam, ideally suited for optical range finders. Using an encoder and a mo-

tor, a rotating mirror sends the beams in a regular angular spaced directions. Simultaneously, a time measurement unit (chronometer) is triggered by that start pulse counting time using a quartz stabilized high clock frequency. Part of the echo signal reflected by the target (the stop pulse) is received trough lens to a photodiode responsible to generate an electrical signal which is then amplified to be able to stop the time counter. The measured time is fed to an internal process unit which processes the data for the output range data values. To obtain $1mm$ in distance measurements accuracy, the accuracy of the time measurement should be $6.7ps$. Data processing methods described in [36] and in [31] include averaging, filtering, clutter suppression (due to particles in between sensor and target, common in foggy conditions), elimination of background echoes specially in short distance measurement scenarios and varying the acquisition and measuring times.

$$\lambda = \frac{hc}{E_g} \tag{3.6}$$



(a) LRF Laser Diode unit.

(b) Laser diode in detail.

Figure 3.4: Diode laser functioning details.

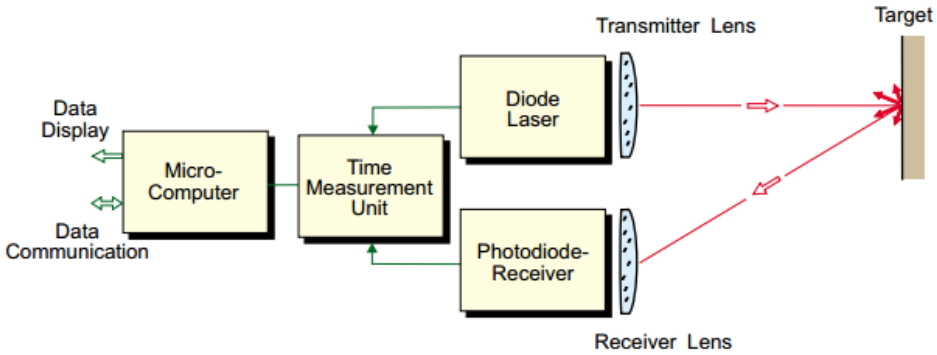


Figure 3.5: LRF Block diagram.

Target influences

Target shape, size, thickness, color and material, as well as meteorological conditions are some of the most important factors that play a critical role in the light detection phase and laser efficiency. A reflectivity coefficient is introduced to characterize the amount of incident electromagnetic power a target surface returns when is struck by a light beam. Not to confuse with reflection which is the ratio between reflected and incident electric field. The reflection of a laser beam vary as a function of the target structure and color. Black objects are the worst to project a laser on because black color has the capability to absorb all light wavelengths converting them into heat. With that said, light surfaces reflect light better than dark surfaces which can improve the measurements range. Also worth mentioning is the object density, when objects are ticker, impervious and/or opaque, a higher percentage of the incident energy is reflected back. Regarding the shape, a target can be classified in three types: diffuse, mirror-like reflection and retro-reflection (see Figure 3.7 taken from [31]). The reflectivity, r, of a perfectly diffuse reflecting white surface corresponds to the definition of a 100% and this is the maximum value for this type of surface. Consequentially a reflectivity value superior to 100% is possible for mirror shape surfaces or retro-reflective targets (see the Table 3.1 taken from [36] for reflectivity values for different materials). The most common surface is the diffuse type (see Figure 3.7a). If the surface is very rough, part of the energy is lost due to shading effect. An approximation of an ideal diffusely reflecting surface is often used to be able to use the Lambert's cosine law. This law says that the signal is reflected in an omnidirecional pattern and its intensity is directly proportional to the cosine of the angle between surface normal and incident light (see Figure 3.6 from [37]). So if the incidence angle is perpendicular

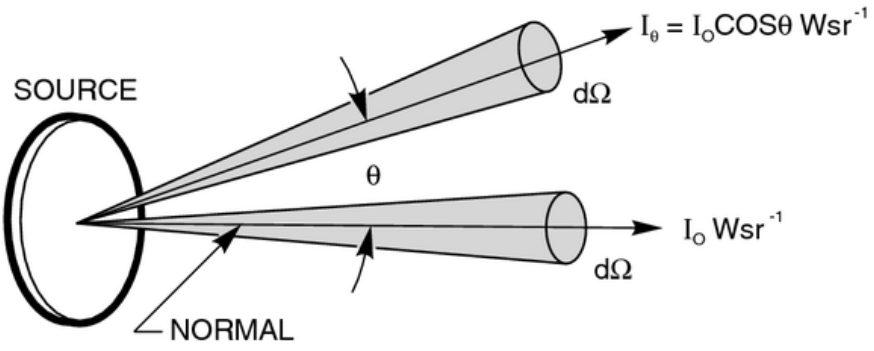
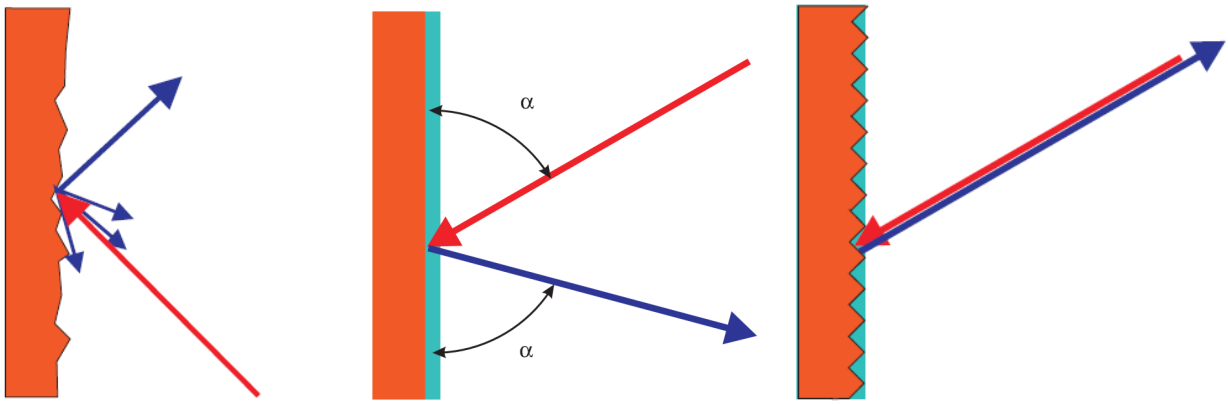


Figure 3.6: Lambert's cosine law.

to the surface, the energy is optimally reflected, where in case of an incidence angle different from 90° a corresponding energy and scanning range loss is incurred. The mirror like surface acts, like the name suggests, as a mirror, sending the incident beam almost entirely in another direction with an angle, with respect to the target, equal to the angle of incidence (see Figure 3.7b). Except the 90° case, this could lead to incongruous readings as the reflected ray acts as incident ray when colliding with other targets, making these secondary targets the detected ones. Contrasting to the previous surface is the retro-reflective type (see Figure 3.7c) where, regardless of the incident angle, the reflected angle is the same as the incident. This is the best case scenario as the reflected energy can reach over 100% making



(a) Remissions on a diffuse surface.

(b) Remissions on a mirror-like surface.

(c) Remissions on a retro-reflective surface.

Figure 3.7: Different types of surfaces.

the target easily detectable even at long ranges. This property can be found in plastic reflectors ("cat's eyes"), reflective tape and triple prisms. Maximum range itself, as implied before, strongly depends on target reflectivity. This relation is expressed by a function of reflectivity where the lower the target reflectivity, the lower the maximum range that can be achieved for the same device (see Figure 3.14 taken from [36]). A range correction factor characteristic, like the one seen in Figure 3.8a, is given by the device manufacturers to correct the maximum range for a given target reflectivity. Also meteorological visibility have a negative impact on maximum range being the fog the harmful scenario. Even the sunlight or strong artificial light has the potential to overwhelm and interfere with the sensor reducing, this way, the maximum range. Again, a factor is introduced to describe this relation where maximum range is reduced due to the atmospheric conditions according to the range reduction factor like the one depicted in Figure 3.8b.

Wavelength also contributes to the reflection coefficient. Regarding the targets size, there is a problem when the light beam hemisphere diameter becomes larger than the target dimensions. In this case, the target reflects only a portion of the transmitter power, and the energy that is not reflected is lost. If the reflectivity is reported less than it really is, the maximum range is affected. As stated in (3.2) the spot size increases proportionally with distance (z). Another problem arises when the angular resolution is lower, at distance, the beam spots might overlap because the distance between spots decreases. Using (3.5) the area covered by the light cone at a distance (z) is given by:

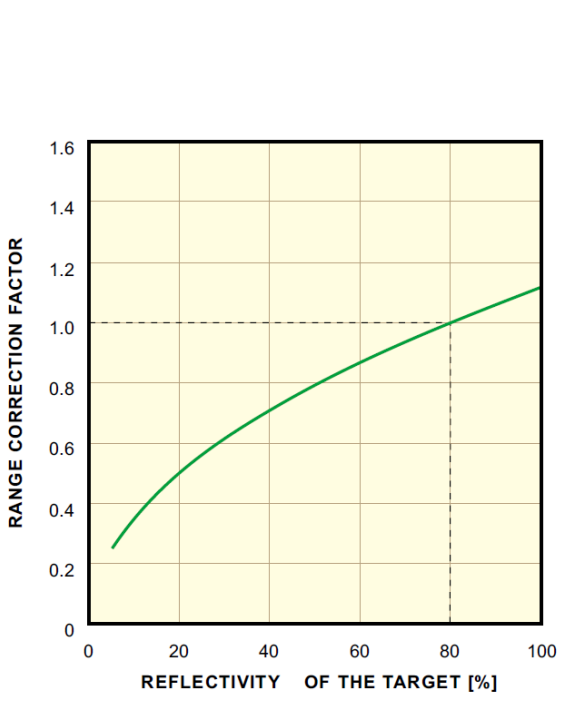
$$A_{Beam} = \pi\theta^2 z^2 \tag{3.7}$$

According to [38] the power reflected by the target can be written as:

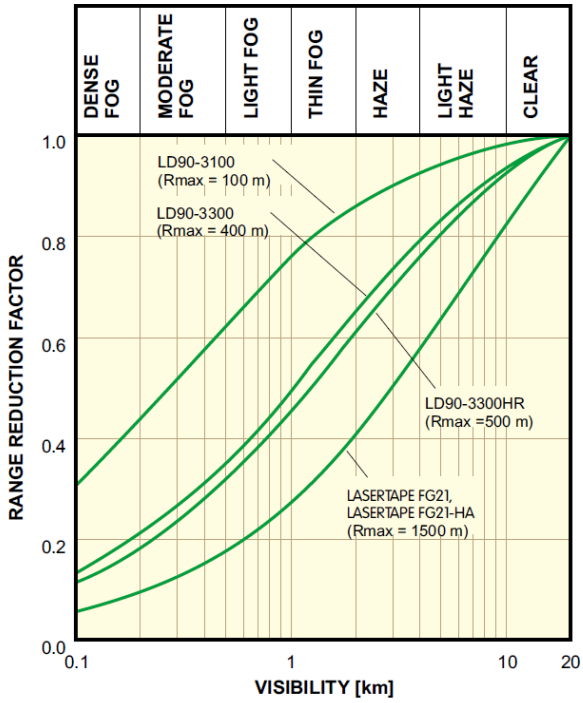
$$P_{Target} = \frac{A_{Target}}{A_{Beam}} P_{Transmitter} \tag{3.8}$$

Where A_{Target} is the target area exposed to the beam, $P_{Transmitter}$ is the power of the transmitted pulse.

Given the detector lens diameter, D, and electromagnetic propagation attenuation proportional to the



(a) RIEGL range correction factor characteristic due to target reflectivity.



(b) RIEGL range correction factor characteristic due to atmospheric conditions.

Figure 3.8: Different types of characteristics.

square of the distance, the power detected by the photodetector is given by:

$$P_{Detector} = \frac{D^2 A_{Target}}{8\pi\theta^2 z^2} P_{Transmitter} \tag{3.9}$$

Here can be seen a quadratic dependence of the detector dimensions and a linear dependence for the laser output power. In fact, contrary to what would be expected, what drives a laser maximum range, and consequentially the price tag, is less the output power and more the quality of the electronic receiver and its lens.

Taking into account the Lambert's cosine law, (3.9) must be multiplied by $\cos\phi$. According to [38], the overall resultant value of $\cos\phi$ may be approximated by 0.5 because the tested target did not have a flat surface and presented a topographical structure. So (3.9) becomes:

$$P_{Detector} = \frac{D^2 A_{Target}}{16\pi\theta^2 z^2} P_{Transmitter} \tag{3.10}$$

For the final (3.12), again according to [38], additional parameters and coefficients were added. Respective physical range coefficients, were taken into account, the target reflectance ρ, optical efficiency of the transmitter $\eta_{Transmitter}$, optical efficiency of the receiver $\eta_{Receiver}$ and atmosphere transmission factor given by:

$$T = e^{-\alpha z} \tag{3.11}$$

Where α is the atmosphere extension coefficient due to absorption and scattering. As stated above, it

Table 3.1: Target material reflectivity for wavelength of 0.9 micrometers.

	Material	**Reflectivity**
Diffuse	white paper	up to 100%
	Dimension lumber (pine, clean, dry)	94%
	Snow	80 - 90%
	Beer foam	88%
	White masonry	85%
	Limestone clay	up to >75%
	Newspaper print	69%
	Tissue paper, two ply	60%
	Deciduous trees	typ.60%
	Coniferous trees	typ.60%
	Carbonate sand (dry)	57%
	Carbonate sand (wet)	41%
	Beach sands, bare areas in desert	typ. 50%
	Rough wood pallet (clean)	25%
	Concrete smooth	24%
	Asphalt with pebbles	17%
	Lava	8%
	Black neoprene	5%
	Black rubber tire wall	2%
Mirror-like	Reflecting foil 3M2000X	1250%
	Opaque white plastic	110%
	Opaque black plastic	17%
	Clear plastic	50%

is assumed the beam profile is Gaussian, characterized by its beam divergence in (3.5) and a circular target object, the Gaussian beam range final equation is given by:

$$P_{Detector} = \frac{T^2 \rho \eta_{Transmitter} \eta_{Receiver} D^2}{16 z^2} \left\{ 1 - \exp\left\{ -\frac{2 r_{Target}}{\theta^2 z^2} \right\} \right\} P_{Transmitter} \tag{3.12}$$

Noise

Noise sources in LRF devices include noise generated by electronics, shot noise caused by the background radiation inducted current and shot noise created by the noise of the signal current. The main sources of inaccuracy in LRF devices are noise-generated timing jitter, walk, nonlinearity and drift [15]. Timing jitter refers to a deviation in the moment of pulse light detection. The higher the slope of rising edge of the corresponding electronic pulse signal of the detector preamplifier, the lower the jitter. On the other hand the higher the root mean square value of the noise the higher the jitter. The walk error affects precision and is caused by pulse amplitude and shape variations which occur often in long distances readings as energy decreases with the square of the distance, leading to earlier or late in time pulse

detection too (see Figure 3.9 taken from [15]). Nonlinear and drifting errors in readings are caused by system time interval discretization errors. Due to the nature of these errors, the precision can be greatly improved by averaging the readings. The improvement is proportional to the square root of the number of results averaged [15].

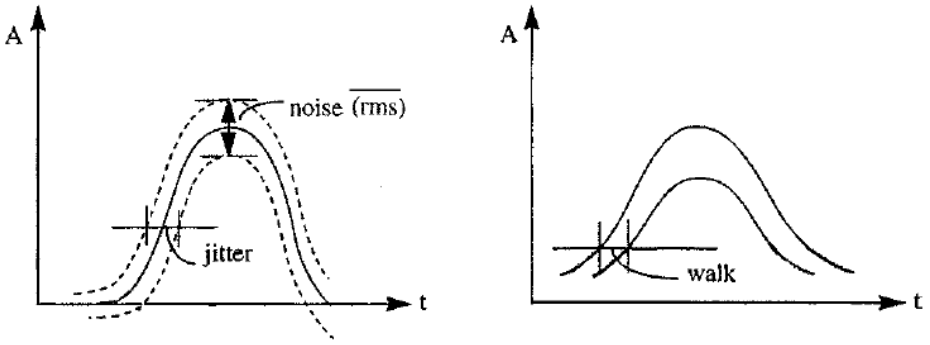


Figure 3.9: Timing jitter and walk noise derived errors.

3.2 LIDAR model

The sensor model developed includes the main features and common properties across the range of this type of sensors. These features can affect, directly or and indirectly, the measurements which, at most, could lead to different final results. Some of the properties have constant values while others could change depending on the scenario. The following list describes the identified and implemented properties and features of the LIDAR model developed. A notation table is presented in Table B.1 in the appendix.

Noise

Noise is a random variable introduced to simulate the readings imperfections due to internal settings or external influences on the device. It is modeled by additive and independent Gaussian noise with a probability density function (PDF) with zero mean, as shown in [18] and [19]. Some examples of Gaussian PDF variants are presented in Figure 3.10 (Image by [39]). The description parameters (μ and σ) of simulated models are constant for each device and are described as follows:

$$\zeta = N(\mu, \sigma) \tag{3.13a}$$

$$\mu = 0 \tag{3.13b}$$

$$\sigma^2 \in \Re \tag{3.13c}$$

In simulation context, noise is used to model the deviation from the correct readings, p_i. This point is obtained in the simulator by intercepting every sensor ray with every wall and selecting the appropriate

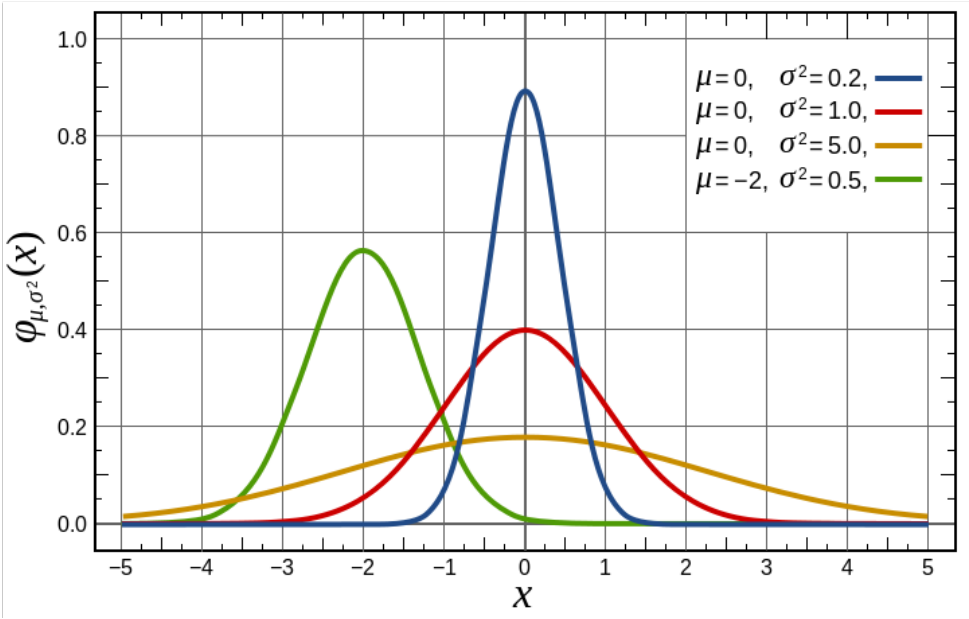


Figure 3.10: Some examples of Gaussian distributions.

point. The coordinates of a point affected by noise, p_n, are given by (3.13k). LIDAR noise is composed by two components: angular noise and linear noise (see Figure 3.11).

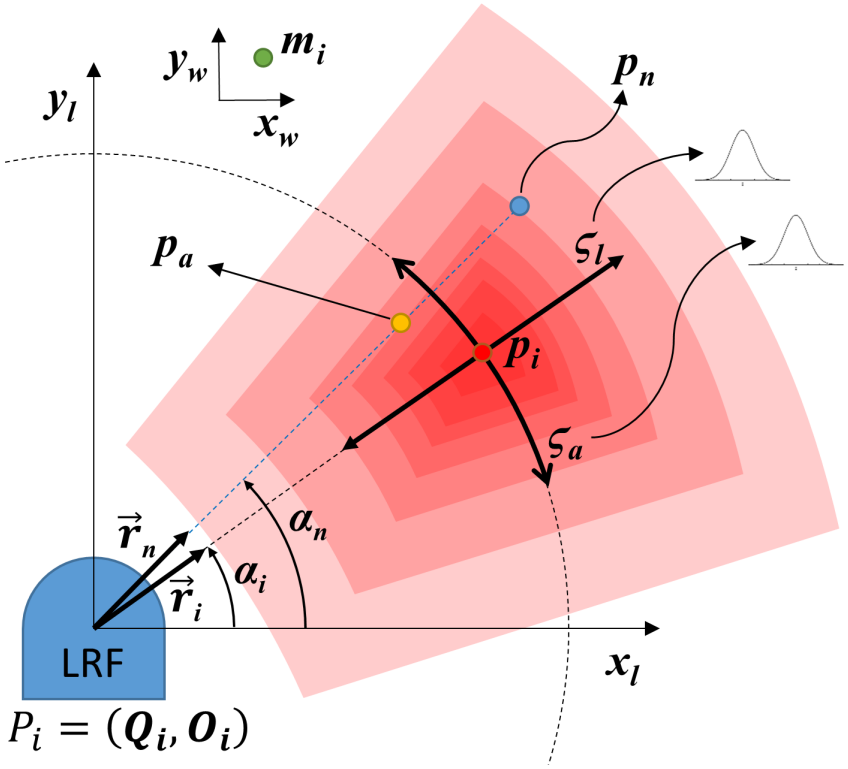


Figure 3.11: Noise effect on LIDAR readings.

It should be noted that linear and angular noise are not commutative operations due to the nonlinearity of map and so, they must be calculated in sequence: first the angular noise and then the linear noise. That being said, first, the point p_a is obtained from provoking an angular deviation ζ_a to $\vec{r}_i$ yielding

$\vec{r_n}$. This angular noise, that affects the direction of the laser beam, is calculated as follows:

- Angular noise:

$$\zeta_a = N(\mu_a, \sigma_a) \tag{3.13d}$$

$$\vec{r_n} = R(\zeta_a) \times \vec{r_i} \tag{3.13e}$$

where:

$$R(\zeta_a) = \begin{bmatrix} cos(\zeta_a) & -sin(\zeta_a) \\ sin(\zeta_a) & cos(\zeta_a) \end{bmatrix} \tag{3.13f}$$

r_i: ray correct direction p_a: resultant intersection point

Then $T_v(\zeta_l)$ applies a linear deviation of ζ_l in the direction of $\vec{r_n}$ (see Figure 3.11). This linear noise affects the value of the readings in the direction of the noisy beam and is calculated as follows:

- Linear noise:

$$\zeta_l = N(\mu_l, \sigma_l) \tag{3.13g}$$

$$p_n = T_v(\zeta_l) \times p_a \tag{3.13h}$$

Where:

$$\vec{v}_n = \begin{bmatrix} \hat{r}_x \\ \hat{r}_y \end{bmatrix} \times \zeta_l \tag{3.13i}$$

$$T_v = \begin{bmatrix} I_{2\times 2} & \vec{v}_n \\ 0 & 1 \end{bmatrix} \tag{3.13j}$$

In summary each reading is affected by noise the following way:

$$p_n = T_v(\zeta_l) \times (p_a(\zeta_a, \vec{r_i})) \tag{3.13k}$$

Figure 3.12 shows, in detail, the resultant points for a map corner, before and after applying 3.13k. The longer lines indicate the supposedly true ray lines, the points over the map lines are the p_a points and the remaining points, connected to them by a shorter line, are the p_n points.

Pose

The pose P_i of a particular LIDAR device i is composed by the set of its position in space and its orientation both given on the map frame (x_w, y_w).

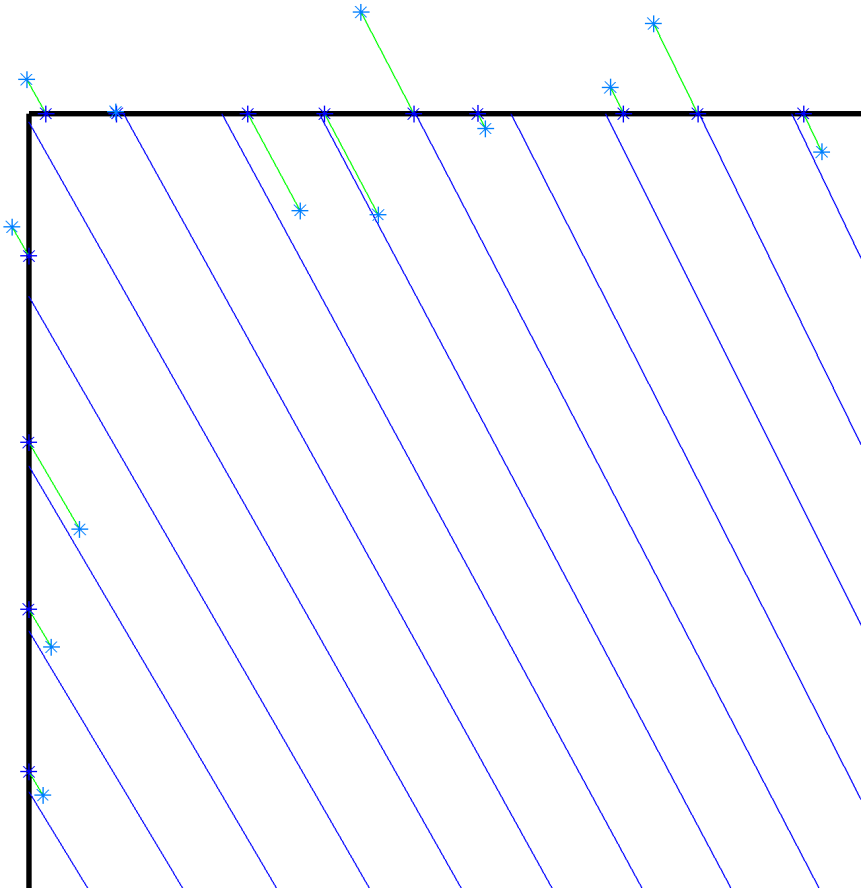


Figure 3.12: Noise Simulation results.

Position

It is assumed that the position of a particular LIDAR from the D devices in the network is restricted to the map limits l. The coordinates are given in the map frame and are defined as:

$$Q_i = (x_w, y_w, z_w), \forall x, y, z \in l, \forall i \in D \tag{3.14a}$$

Orientation

The map frame is used as-well to describe LRF orientation. In a three dimension scenario it takes three Euler angles to completely describe an orientation: roll, pitch and yaw. All angles are given in degree units.

roll

Roll is defined as the rotation angle of xx axis:

$$\psi \in [0°, 360°] \tag{3.14b}$$

pitch

Pitch is defined as the rotation angle of yy axis:

$$\theta \in [-90, 90] \tag{3.14c}$$

yaw

Yaw is defined as the rotation angle of zz axis:

$$\alpha \in [0°, 360°] \tag{3.14d}$$

Finally, LRF orientation is defined as:

$$O_i = \{\psi_i, \theta_i, \alpha_i\} \tag{3.14e}$$

Two dimensions case

In 2D case, the z coordinate is nonexistent so position is fully defined by:

$$Q_i = (x_w, y_w), \forall x, y \in l, i \in D \tag{3.14f}$$

The orientation, in 2D case, is given by one angle only, the α angle (see Figure 3.11), as the other angles (ψ and θ) are null. Alpha is defined as the rotation angle defined positive from xx axis to yy axis:

$$\alpha \in [0°, 360°] \tag{3.14g}$$

And then:

$$O_i = \{\alpha_i\}, \forall i \in D \tag{3.14h}$$

Final Pose definition

Only points returned by the LIDAR are given in respect to its frame p_i represented in Figure 3.11. The relation between the map frame and the LIDAR frame is given by:

$$p_w = {}^{w}_{i}\mathbf{T} \times p_i \tag{3.14i}$$

Where ${}^{w}_{i}\mathbf{T}$ is the homogeneous transformation matrix that combines rotation and translation, applied in the stated order, given by:

$${}^{w}_{l}\mathbf{T} = \begin{bmatrix} R & \vec{t} \\ 0 & 1 \end{bmatrix} \tag{3.14j}$$

The rotation angles and translation vector values are given by the algorithm resultant alignment output and are defined as follows:

$$
\begin{aligned}
R &= R_x(\psi) \times R_y(\theta) \times R_z(\alpha) \\
&= \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \\
&= \begin{bmatrix} c(\psi)c(\theta) & c(\psi)c(\theta)s(\alpha) - s(\psi)c(\alpha) & c(\psi)s(\theta)c(\alpha) + s(\psi)s(\alpha) \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\alpha) + c(\psi)c(\alpha) & s(\psi)s(\theta)c(\alpha) - c(\psi)s(\alpha) \\ -s(\theta) & c(\theta)s(\alpha) & c(\theta)c(\alpha) \end{bmatrix}
\end{aligned}
\tag{3.14k}
$$

Note: c means $\cos$, and s means $\sin$

$$
\vec{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{3.14l}
$$

It is now possible to fully define the pose of each sensor:

$$
P_i = \{Q_i, O_i\}, \forall i \in D \tag{3.14m}
$$

Range

Range is the maximum valid distance reading value the device can output in ideal conditions. This value may vary with the features stated in Section 3.1. A value for the LRF range parameter is often given by the manufacturers under certain ideal conditions, and a range correction factor is often provided to compensate for changes of these conditions.

$$
R_{max} \in [0; +\infty] \tag{3.15}
$$

FOV

Field of view is the angle aperture from which the readings are taken from (see Figure 3.13). It is defined using an initial and final angle, in degree unit, and the scanning direction, with respect to sensor frame. Also, a maximum FOV angle needs to be defined.

$$
\theta_{ini} \in [0^\circ, 360^\circ] \tag{3.16}
$$

$$
\theta_{fin} \in [0^\circ, 360^\circ] \tag{3.17}
$$

The maximum FOV angle given by:

$$FOV = \theta_{ini} - \theta_{fin}, \forall\theta \in [0°, 360°] \tag{3.18}$$

The relation between maximum FOV and FOV is the following:

$$FOV \leq mod(|\theta_{ini} - \theta_{fin}|)_{360°} \tag{3.19}$$

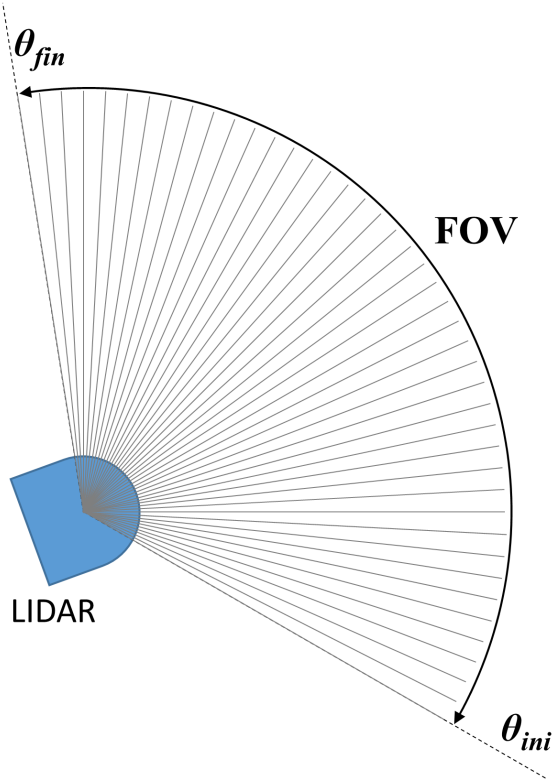


Figure 3.13: FOV.

Scan direction

The scan direction defines the rotation direction of the LRF mirror which derives the readings sequence.

$$Dir \in \{CW, CCW\} \tag{3.20}$$

Where:

- CW: clockwise;
- CCW: counter clockwise.

Scan velocity

Scan velocity is the angular speed at which the LRF mirror rotates. Is given in degrees per second.

$$s_w \in \Re \tag{3.21}$$

Scan rate

The output rate of the device. This parameter depends on the processing power and can change with FOV and angular resolution. It is given in Hz.

$$s_r \in \Re \tag{3.22}$$

Number of scans

The number of complete scans necessary before data processing application and LIDAR output

$$s_n \in \mathbb{N}_1 \tag{3.23}$$

Angular resolution

Minimum necessary input that leads to a minimum angular step in output. It is given in degrees.

$$res_a \in \Re^+ \tag{3.24}$$

Given the FOV value and the angular resolution it is possible to define the number of sensor steps (default rounding):

$$t = \frac{FOV}{res_a} \tag{3.25}$$

Linear resolution

Minimum necessary real distance variation that leads to a minimum reading step in output. It is given in meters.

$$res_l \in \Re^+ \tag{3.26}$$

Minimum energy

Energy/reflectivity threshold below which the device rejects the reading. This value depends on the lens sensibility of the sensor.

$$rl_{min} \in [0; 1] \tag{3.27}$$

Required reflectivity

Function that validates a reading. A reading is considered valid when, for the respective distance, the obstacle reflectivity is above a certain threshold. The shape of the function varies for different models and is also affected by the environment. So, for every distance in the sensor valid range, there exists a minimum energy value below which the reading is considered invalid. To approximate the model function to a real sensor curve, in Figure 3.14 (image taken from [31]), a polynomial curve was generated. The function inputs and outputs are the following:

- Input: reflectivity, incidence angle, LIDAR readings, environment.
- Output: binary value (True or False)

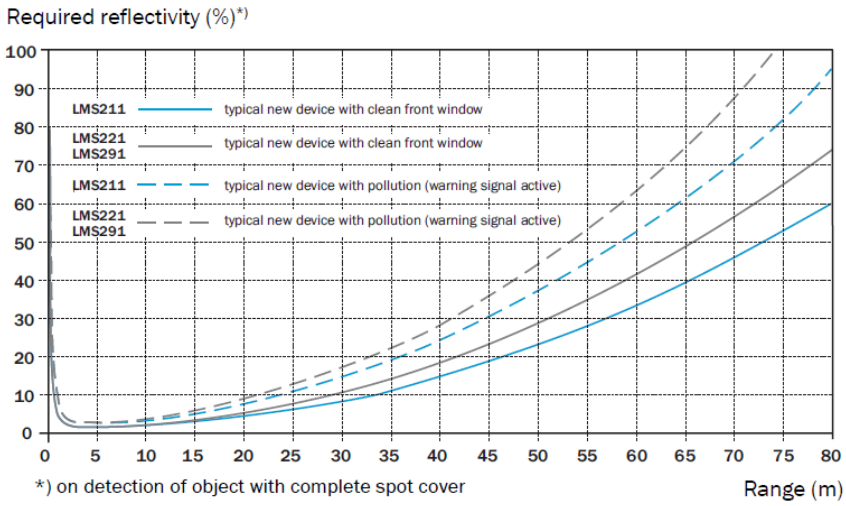


Figure 3.14: Required reflectivity characteristic on SICK LIDARS.

Style

Style includes all cosmetic features (shapes and colors) associated with the LIDAR graphical representation on the simulation environment.

Results

Sensor output for simulation purpose after a complete scan:

$$Out_i^s = [d_n \ \alpha_n \ p_n \ rl_n \ w_n \ \theta_n]^\top, n \in [0; t], i \in D \tag{3.28}$$

Where:

- d is the Euclidean distance beginning at the lens an ending at the obstacle;
- α is the lens angle on the device frame;

- p are the Cartesian coordinates of the intersection point on the map frame;
- rl is the obstacle reflectivity;
- w is the wall index;
- θ is the incident angle.

The d distance is calculated using basic geometry calculations as shown in Figure 3.15. Only distance readings, angle and eventually reflectivity values are available in simulation of a real scenario. Every other information is used for simulation validation purpose and is not available for the matching methods to use. Having said that, the available output, for the testing phase is the following:

$$O_i^r = [d_n \ \alpha_n \ rl_n]^\top, n \in [0; t], i \in D \tag{3.29}$$

Where:

- d is the distance readings including noise given in the LRF frame;
- α is the lens angle (without noise) given in the device frame;
- rl is the obstacle reflectivity.

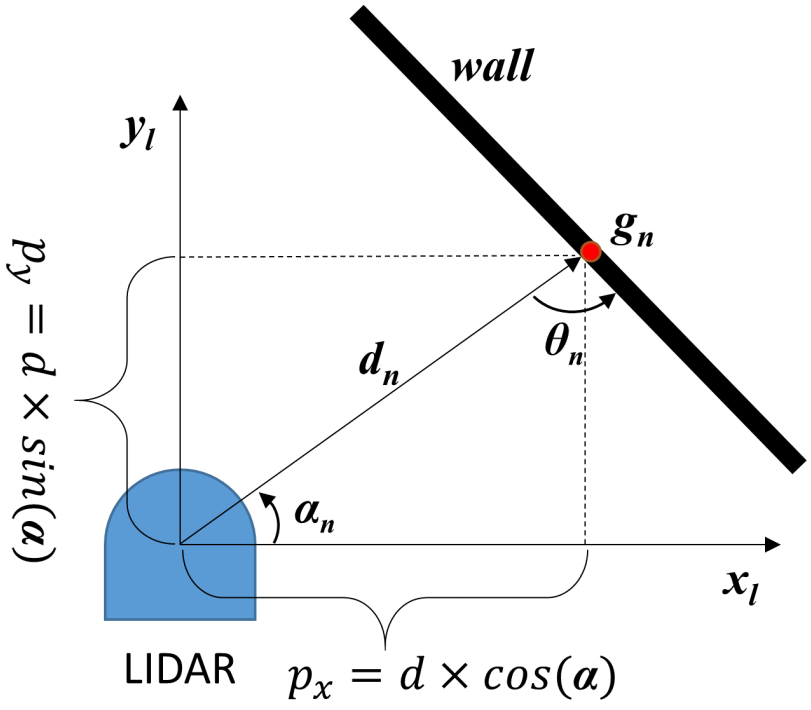


Figure 3.15: Geometry associated with LRF simulation variables.

3.3 Map

Map description

The map is composed by a set of points (usually the vertices) and line segments. Walls are represented by line segments which are described by their two extreme points: the initial point p_k and the final point p_j with $k \neq j$ (see Figure 3.16 for a 2D representation and Figure 3.17 for a 3D representation of the same map example). To maintain the representation complexity relatively low and computationally efficient, every wall, and specially the curved ones, are represented by a set of line segments with arbitrary length. A value of the wall reflectivity is mandatory and should be provided. The total number of walls is given by w_t.

$$M = \{Pts, LS, RL\} \tag{3.30}$$

Where:

- Pts is an array of points;
- LS is a matrix of line segments (initial and final points);
- RL is an array of reflectivity values;

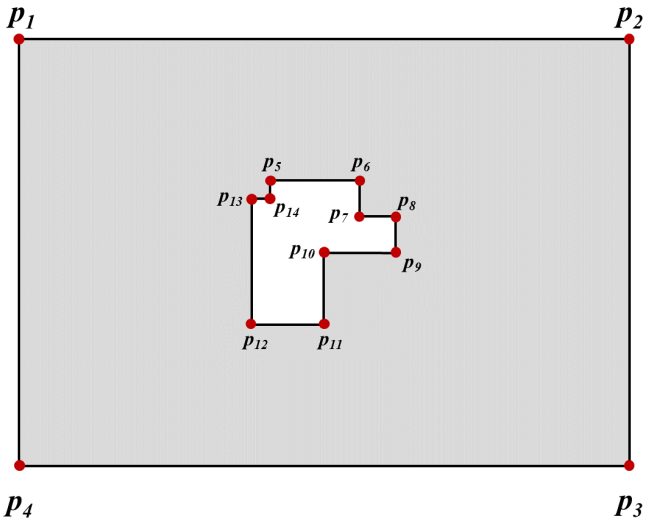


Figure 3.16: Two dimension map presentation.

Coordinate system adopted

A Cartesian coordinates system was adopted for map point description and for LRF data as well. In general, the origin is chosen to be located in the lower left side of map, so every point has positive coordinate values. Figure 3.18 shows a two dimension frame representation example for the map and for LRF. In Figure 3.19 the same map representation is shown, this time in three dimensions where the map (or world) frame is in blue color and the LRF sensor frame is represented in red color.

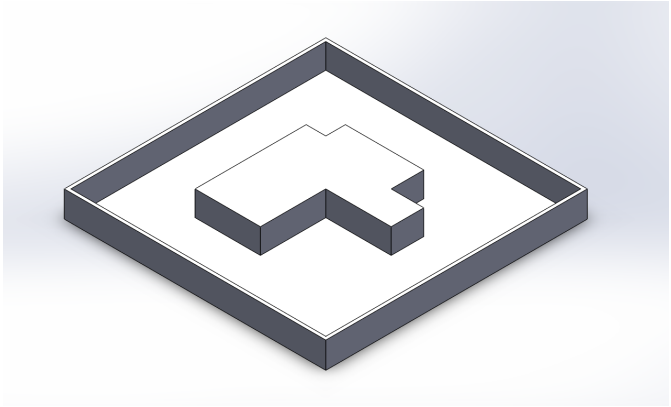


Figure 3.17: Three dimension map presentation.

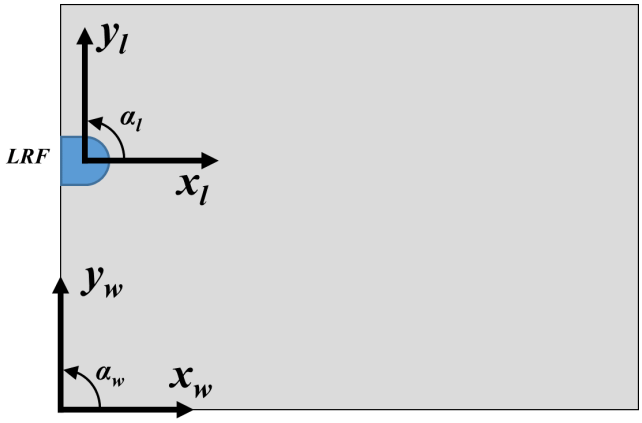


Figure 3.18: Two dimension frame presentation.

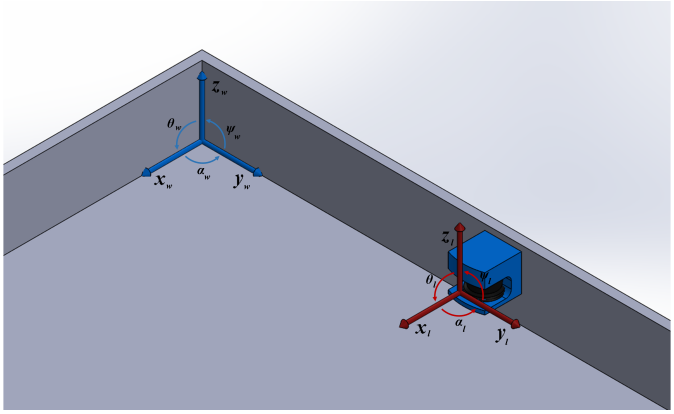


Figure 3.19: Three dimension frame presentation.

Points

Points define the extremes of a line segment, often describing a map vertex. They are defined as follows:

$$Pts = \{p_1, ..., p_m, ..., p_n\} \tag{3.31}$$

$$p_m = (x_m, y_m, z_m), \forall x, y, z \in \mathbb{R}^+, \forall m \in [1, 2, ..., w_t] \tag{3.32}$$

Line segments

Line segments are defined by an initial and a final point. They represent a wall or part of it and are described as follows:

$$ls_i = (a, b), \forall a, b \in Pts \wedge a \neq b, \forall i \in [1, 2, ..., w_t] \tag{3.33}$$

Reflectivity

Reflectivity is a wall property defined as follows:

$$rl_i \in [0, 1], \forall i \in [1, 2, ..., w_t] \tag{3.34}$$

Where:

- 0 when all the pulse energy is absorbed by the wall or sent in another direction (mirror wall);
- 1 when all the pulse energy is reflected back by the wall (retro-reflective wall).

For the record, in the intermediate case ($rl_i \in]0, 1[$), for simplification and efficiency proposes, it is considered that the walls have a diffuse characteristic. Other characteristics like the randomness of wall roughness are diluted in the reflectivity final value for the whole wall. It is also considered negligible the increase in the size of the focus and its influence on the readings. The value of reflectivity described above is defined for an incidence angle perpendicular to the obstacle surface. For values of θ (see Figure 3.15) different from 90 degree, similar to Lambert's law, the reflectivity varies as follows:

$$rl_i^c = rl_i \times sen(\theta_i), \forall i \in t \tag{3.35}$$

Environment conditions

The environment conditions describe the meteorological conditions of the environment, (in this case the map) where the measures are carry out by the LRF. A disturbance of the environment ideal conditions can have a detrimental impact on the measurements made by the LRF since they can hinder the passage of light. It is assumed that the entire map is affected by the environment, that is, there are no areas of the map with different environment conditions from that defined. Some examples of relevant environmental conditions [31] considered are clear weather, which have no negative effect on measurements but the same is not true for rainy, sunny or snow whether.

Environment conditions is defined by:

$$Env \in \{clear, foggy, rainy, sunny, snow\} \tag{3.36}$$

Number of LRF sensors

The number of sensors is defined as the amount of LRF devices present in a given map in a given moment, operational or not, featuring the same model or different models with equal or different charac-

teristics. The number of sensors is defined by:

$$D \in \mathbb{N}_1 \tag{3.37}$$

3.4 Scan simulation process

Scan simulation process is the simulation stage responsible for generating LRF data output. It integrates map data and LRF model profile(s) to deliver an accurate as possible simulation of LRF functioning in the given environment (see Figure 3.20). Before a scan process occurrence, some mandatory variables, described in Section 3.2 must be given in the LRF model profile file. Pose is one of them and should be given in the map coordinate frame. The position, q, indicates the origin of the scan rays and the orientation corresponds to the direction of the device center (or its x axis) where the initial and final FOV angles are defined from. Given those two angles and the angular resolution, the number of scan rays, t, and its respective angles are defined. An array R of normalized ray vectors $\vec{r}$, is generated, all of them having the same initial point q and final point p. Walls are converted into vectors too, by subtracting the initial points w_i and final points w_f, and an array W of $m = w_t$ wall vectors is generated as well. Now, for every combinations of scan ray vectors and wall vectors, intersections calculated using (3.43) except for cases where (3.42) is zero which corresponds to parallel or collinear vectors. The two line intersection variables, in a plane, are identified in Figure 3.21.

$$\vec{r} = (q_x, q_y) - (p_x, p_y) \tag{3.38}$$

$$\vec{w} = (w_{x1}, w_{y1}) - (w_{x2}, w_{y2}) \tag{3.39}$$

$$R = \begin{bmatrix} \vec{r_1} & \vec{r_2} & \dots & \vec{r_t} \end{bmatrix} \tag{3.40}$$

$$W = \begin{bmatrix} \vec{w_1} & \vec{w_2} & \dots & \vec{w_m} \end{bmatrix} \tag{3.41}$$

$$det = \begin{vmatrix} \vec{r_i} & \vec{w_j} \end{vmatrix} \tag{3.42}$$

$$(g_x, g_y) = \left(\frac{(q_x p_y - q_y p_x)(w_{x1} - w_{x2}) - (q_x - p_x)(w_{x1} w_{y2} - w_{y1} w_{x2})}{det}, \frac{(q_x p_y - q_y p_x)(w_{y1} - w_{y2}) - (q_y - p_y)(w_{x1} w_{y2} - w_{y1} w_{x2})}{det} \right) \tag{3.43}$$

After getting all intersection points, g, for a scan line, $\vec{r_i}$, a procedure to check if points belong to wall line segments is needed because wall segment lines were being treated as lines for intersection calculations. A certain point, g_i, belong to a wall with initial point w_1 and final point w_2 if (3.44) are

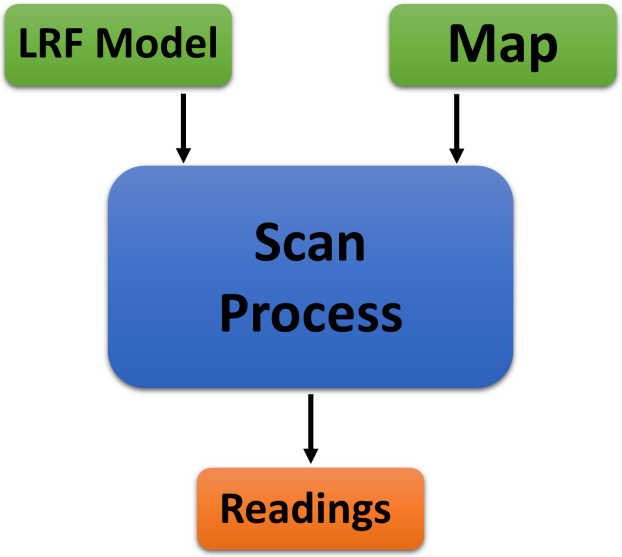


Figure 3.20: Scan block diagram.

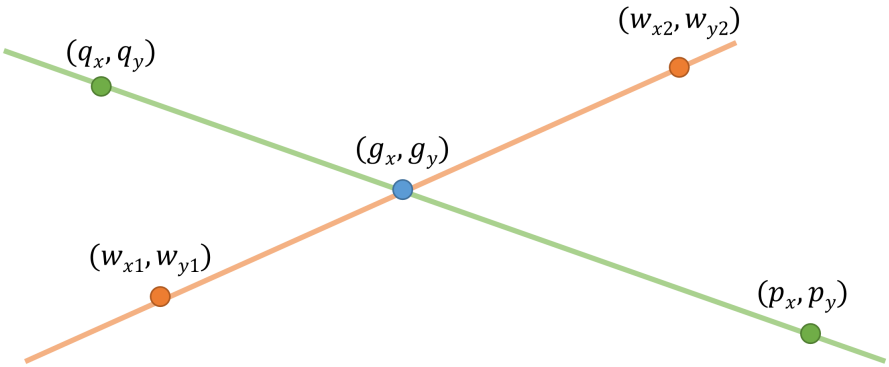


Figure 3.21: Two line intersection on plane.

verified.

$$g_{i,x,y} \geq w_{x,y,1} \wedge g_{i,x,y} \leq w_{x,y,2} \wedge w_{x,y,2} > w_{x,y,1}$$
$$g_{i,x,y} \leq w_{x,y,1} \wedge g_{i,x,y} \geq w_{x,y,2} \wedge w_{x,y,2} < w_{x,y,1} \tag{3.44}$$

Another verification needed is to check if points of intersection are ahead of the LRF and not behind. To do that, a set of vectors R' are calculated from the LRF origin q to all of the g points. To test if all the r' vectors are in the same direction of r, the dot product condition (3.47) was used. Those whom the resultant signal was not positive were discarded.

$$\vec{r'} = (q_x, q_y) - (g_x, g_y) \tag{3.45}$$

$$R' = \begin{bmatrix} \vec{r'_1} & \vec{r'_2} & \dots & \vec{r'_t} \end{bmatrix} \tag{3.46}$$

$$\vec{r} \cdot \vec{r'} > 0 \tag{3.47}$$

Finally, after conditions in (3.42), (3.44), (3.47) are all applied, the final result point may not be yet determined as more than one point may fit these conditions. But the other points are just out of laser reach, laying behind the first wall intersection. If more than one point is returned, the point to choose is the one which presents the lower respective norm from the vectors in R'. It is given by (3.48).

$$g_f = \min_{\|R'\|} g \tag{3.48}$$

The obtained point, g_f, is the perfect point meaning no noise, or any other factor identified in Section 3.1, causes interference in returned measurements. This process repeats for every scan line generated, so in the end an array, G_f, of perfect (or correct) intersection points is returned. To simulate in an accurate way the output of a LIDAR system, the matrix of the distances affect by noise and model limits needs to be returned. The process of noise application, described in 3.2 is then applied.

$$G_f = \begin{bmatrix} g_{f1} & g_{f2} & \cdots & g_{fn} \end{bmatrix} \tag{3.49}$$

After noise simulation, is time to apply the influence of other factors. First, the maximum range limit is applied, distances greater than the given range limit are discarded. Regarding resolution limit, distance values are rounded to the nearest uncertainty value. Next, as stated in Section 3.1, reflectivity has a strong influence on measurement validity. Since the scan lines and the respective wall lines of intersection are known it is possible to extract the angle of incidence θ. For every wall, a reflectivity value must be given as seen in (3.30). Walls shape were considered diffuse so the target reflectivity value "seen" by the LRF is corrected using Lambert's cosine law and it is given by (3.50).

$$rl^c = rl \times \cos(\theta) \tag{3.50}$$

The LIDAR reflectivity characteristic, described in Section 3.2, was approximated using a polynomial function. The Figure 3.22 shows an example of the obtained polynomial function in red, and the cataloged in blue. In case the corrected reflectivity does not surpasses the maximum reflectivity for the corresponding distance measured, the measurement is considered invalid.

The remaining factors identified in Section 3.1 such as beam diameter, meteorological conditions or target color and size influences were not considered due to the inherent lack of information of objects geometry given on 2 dimension space map and its representation. Meteorological conditions are considered optimal since they are to be used in indoor scenario. In this case excessive light could cause measurement interference but optimal light conditions are assumed as well.

At last, the output of the simulation is the array in (3.51) that includes the distances measured and affected by the above described influences, the angles are returned without noise because, typically, the angles are inferred from the number of distances measured per complete scan, or angular resolution, and FOV. Some LRF systems are able to return the reflectivity value for the respective readings, which

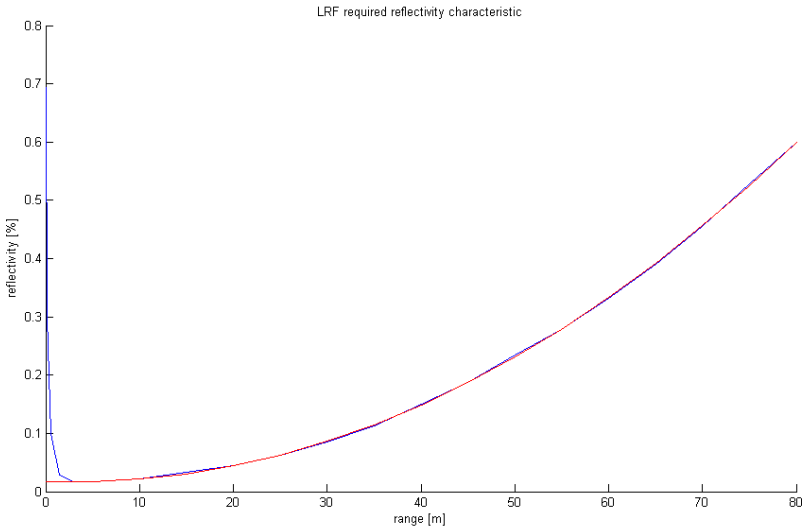


Figure 3.22: Required reflectivity characteristic polynomial approximation.

is an optional parameter in the LRF model description. When a measurement value is discarded, null data is returned in place for the respective measurement. The final available output for the testing phase is given by (3.51).

$$Out_i^r = \begin{bmatrix} d_{f1} & d_{f2} & \dots & d_{ft} \\ \alpha_1 & \alpha_2 & \dots & \alpha_t \\ rl_1^c & rl_2^c & \dots & rl_t^c \end{bmatrix}, i \in D \tag{3.51}$$

Chapter 4

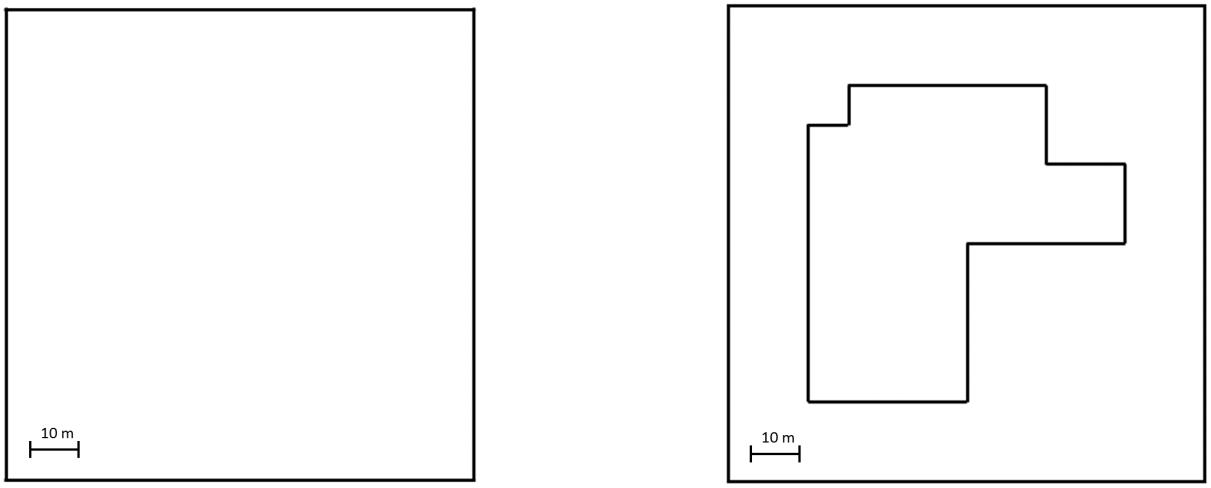
Results

4.1 Simulation Results

The tests to evaluate the algorithms performance were carried out in simulation environment. Due to the nature of simulation, in this case, the true pose is available to compare against the results.

4.1.1 Maps used

Three maps were created to test and evaluate the performance of the different algorithms in simulation. The first one, represented in Figure 4.1a, is the simplest possible symmetrical map with four walls of the same length. The second one, depicted in Figure 4.1b, was constructed to be as asymmetrical and simple as possible. The third, in Figure 4.2, is a complex map from ITER basement, where the actual CPRHS will navigate and the LRF devices will operate. In this case, the calibration environment configuration requires the doors for the Cask compartment to be closed and the absence of the CPRHS vehicle. The accuracy in this map is expected to be as high as possible, at least of the order of magnitude of the LRF system measurements uncertainty.



(a) Symmetrical map.

(b) Asymmetrical map.

Figure 4.1: Maps developed.

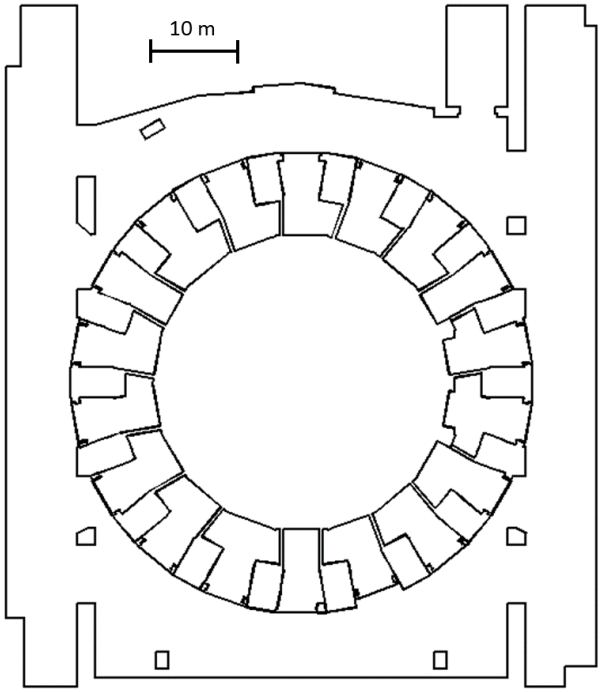


Figure 4.2: Map from ITER Tokamak building level B1.

4.1.2 Simulation Experiments Description

An array of four LRF devices were tested at each map. Figure 4.3 shows the respective location and number of each device for the three different maps used. Each device was tested for the different scenarios described in Chapter 2. Both maps are presented without outliers, and were converted to points with a resolution of 50 points per meter except for the voting tests, where the density was changed to 1 point per meter, so that the algorithm execution could end in feasible time. The initial pose estimate values, $\hat{P}_i$, used across the tests were chosen on purpose to be least 1 meter away from the true position and, at most, rotated 30° from the true orientation. These unreasonable values were intentionally chosen to test the algorithms performance. In the ITER map, there are map lines that were impossible to be reached by the laser rays, and since its absence makes no difference on the results, they were suppressed to increase performance. The resultant map is shown at the lower left part of Figure 4.3.

For each map, a model of a LRF was created featuring the following main properties:

- FOV: 180°;
- STD: ±50 mm;
- Max range: 100 m;
- Linear resolution: 1 mm;
- Angular resolution: 0.5°.

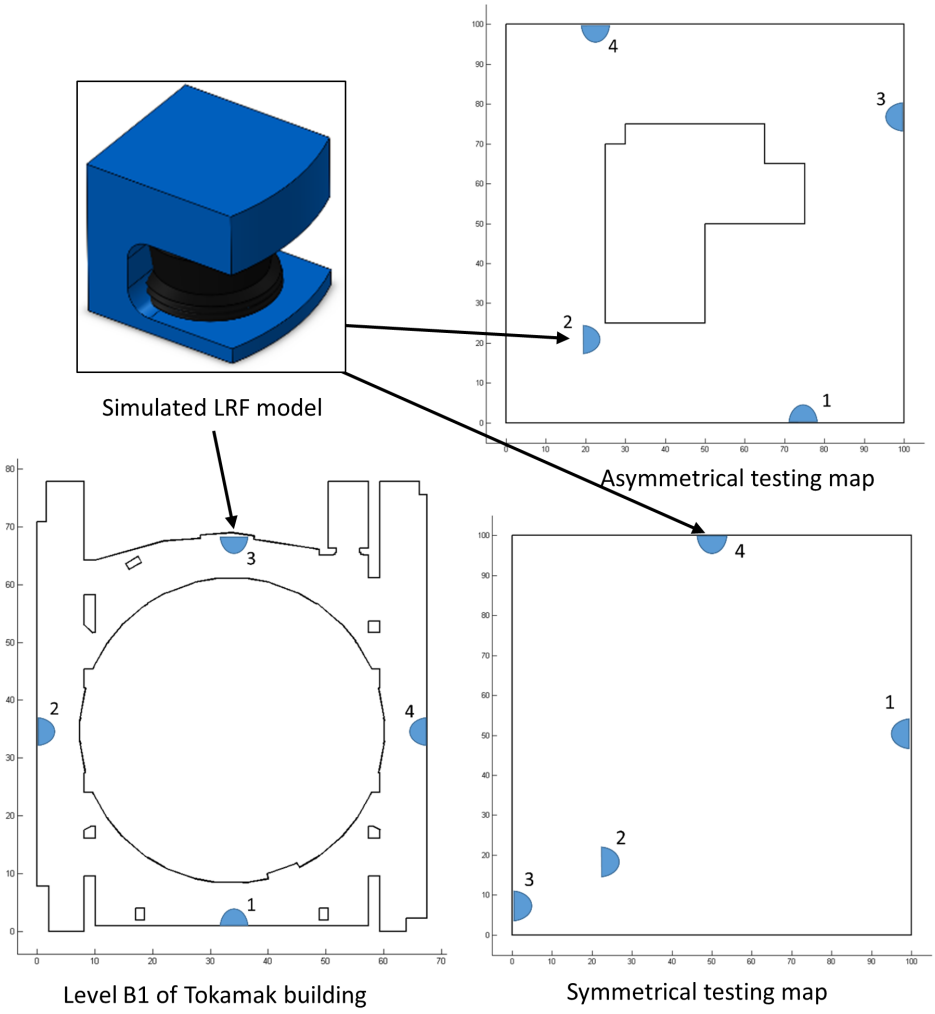


Figure 4.3: Test locations for each of the simulation experiments maps.

Four LRF devices were set in the predefined poses represented in Figure 4.3, from where each device took a scan. In case no initial pose estimate is given, the tolerance values (described in Chapter 2.4.2) that gave the best results, for the clustering and feature extraction phases were the following:

- proximity: 0.5 m;
- slope: 5°;
- cluster: 1 m;
- split: 0.5 m.

4.1.3 Results

The results obtained for the simple symmetrical, asymmetrical and for ITER maps are presented in Table 4.1. The “LRF” column indicates the pose numbers for the respective map, the Scen. indicates the scenario. $\hat{P}_t$ indicates the exact true pose where the measurements were taken from, $\hat{P}_i$ is a guess for the initial pose estimate, $\hat{P}_f$ is the final pose resulting from the algorithm output, “e_{pp}” is the point to

point error, given by (1.1), $e_{abs.}$ is the absolute error and measures the difference between the true pose and final pose estimate. It is composed by two components: the first measures the position difference and is given by the Euclidean distance between true position and the position estimate from the final pose, the second measures the angle difference and is given by the absolute difference between the true angle and the angle estimate from the final pose. "RT" (run time) is the time elapsed in the algorithm execution[1]. The "CC" column means correct convergence and indicates whether the respective final pose estimate, when used as $\hat{P}_i$ in ICP final alignment, converges to the same result from the respective scenario A. The Figures in 4.4, 4.5 and 4.6 illustrate, respectively, the results obtained for the simple symmetrical, asymmetrical and ITER maps, for the poses identified as 1 in each map. In each Figure three images are displayed: at the left side the ICP resultant alignment of LRF points (in red) and the respective zoomed iterative evolution from $\hat{P}_i$ to $\hat{P}_f$; the heat map resulting from the voting method is shown on the upper right section; and on the lower section the geometrical features extracted (vertices represented by the small circles) in the vertex/extremes method are pictured. For all the results displayed for the vertex or extremes method, the score function used was not (2.18) but instead p-p error to maintain a congruent evaluation across all methods. For scenario B an array of 13 different and equally spaced angle values were used and the result is not displayed because it is similar to the ICP result. The brute force ICP experiment described in Section 2.4.4 was also conducted. Its results for the asymmetrical map are shown in Figure 4.7. Regarding voting results, in the Appendix Section A.1, Figures of the votes accumulation results are shown for each of the maps.

On the first map results (see Figure 4.4), the simple symmetrical map, for pose 1, the side walls were just partially "seen" by the LRF despite the distance being lower than the maximum range. This

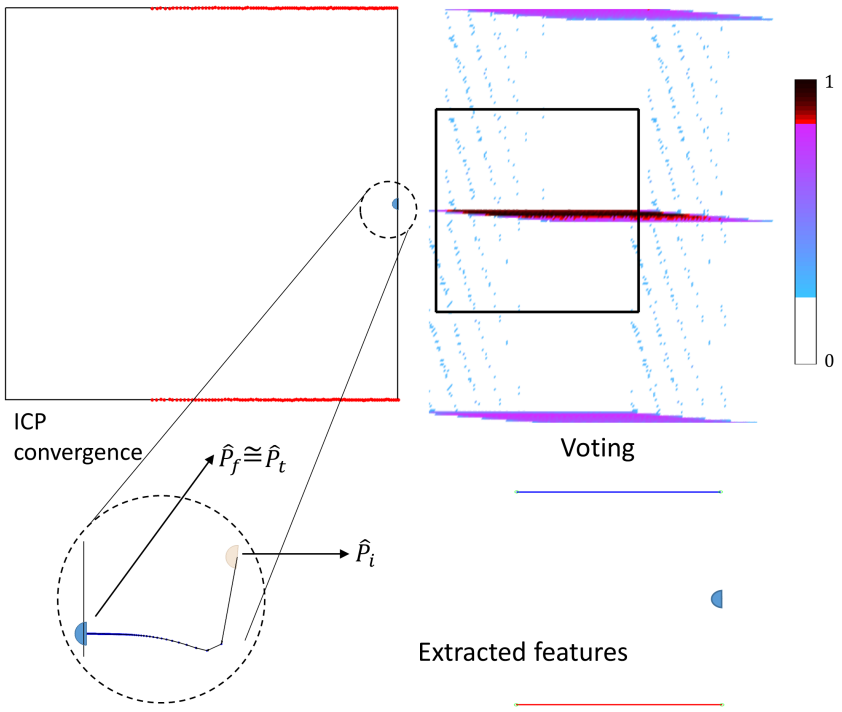


Figure 4.4: Simulation results for simple asymmetrical map.

[1]tests were always done on the same machine, an i5 processor.

is caused by two factors together: every wall has a reflectivity coefficient of value 1, but the incidence angle decreases the effective reflectivity as stated in Section 3.1. On the other end, the greater the distance, the more reflectivity value is required for an accurate and valid measurement. In this case, the minimum reflectivity value for the correspondent distance constraint was not met, thus, no measurement was delivered. In scenario A, for pose 1, the ICP alignment took more time and more iterations to converge when compared to other poses for the same map. It also presents a p-p error much lower than the absolute error. This happens due to the presence of two parallel line shaped point clusters, and the given $\hat{P}_i$ being outside of the box. After the brief rotation alignment, the points that remained outside of the "box" became less influent at each iteration. The same situation is presented for pose 3 but, this time, with a larger initial angle deviation, leading to an almost correct position alignment (about 1m away from the true position) after the rotation alignment finished. It couldn't totally recover because of local minima problem. Pose 4 composes a similar case where the $\hat{P}_i$ is given, this time, inside the four walls. In this situation, it took much less time and iterations for the ICP to converge because right after the rotation alignment, the algorithm became trapped in a local minima. The local minima pattern, in this case, is best presented by the voting result that shows a line in the middle representing the higher chance for the possible LRF pose. The upper and the lower map lines both vote symmetrically for the middle of the map and for the outside. The superposition of the middle votes reveal the line pattern of higher chance for possible LRF location. Pose 2 doesn't suffer the same problem as it has perpendicular line shaped points that help improve the algorithm convergence by dismantling and reducing local minima points. In scenario D, where $\hat{P}_i$ is completely unknown, despite a specific result being shown on the respective Table entries, for every pose, four different results delivered the same error value. These four different results correspond to equal probability associated with the four symmetrical and identical sides of the map. This way it's impossible, for the algorithm to recover the correct pose estimate. Also, since no vertex could be extracted, except in pose 2 test, the extremes method variant, described in Section 2.4.3, took place to extract line segment end points.

The asymmetrical map was created to be simple with distinct and different environment perspectives from every place to reduce localization ambiguity. This problem is more prominent in scenario D where the global search takes place due to the lack of initial pose estimate information. Effectively, the result show an improvement that, in some cases, rivals the ICP method when in comparison to the previous map. The correct extraction of lines and vertices (see Figure 4.5) have given a positive contribution to the accuracy as well as for the algorithm run time. Wrong vertex identification and extraction means worthless computational effort not only spent on point transformations but in the p-p error computation which has high complexity (see Section 2.2.1). Wrong extracted lines associated with a correct extracted vertex, can induce a correct position into a higher p-p error because of the orientation angle derived from the erroneous lines. The opposite can happen too where the orientation is correct but a miss extracted vertex spoils the results. This case is more beneficial than the previous for the ICP application in final alignment because of its sensibility to initial angular deviations. In scenario A, since this map presents perpendicular and parallel walls, the multiple continuous local minima problem from the previous map is attenuated. Although not being the same situation, this time, the voting results heat-map image shows a

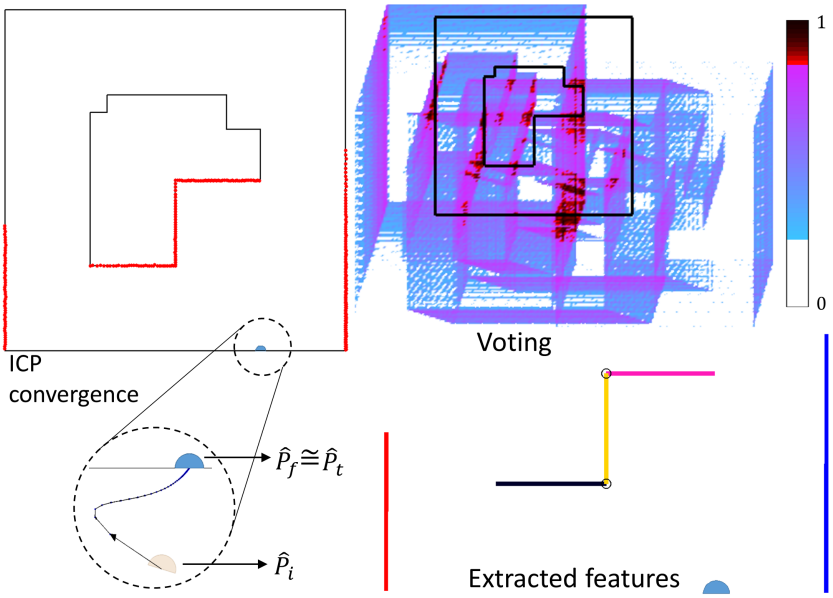


Figure 4.5: Simulation results for asymmetrical map.

focal points pattern instead of a contiguous line pattern as happened in the previous map. In the same image, near the correct location (pose 1) a relatively large red dot spot zone can be seen indicating a higher probability of the scan been taken from there. Other red but small spots on the reaming map locations indicate votes for the similar target patterns across the map as it is composed only from vertical and horizontal lines with different lengths. Scenario B, again, reached the same p-p error as scenario A, but when coming from different initial angles, the results will vary slightly (in the order of LRF accuracy) for the better or for the worse.

ITER environment is the main focus and the reason of the presented work. Given a reasonably good initial pose estimate, in scenario A, the algorithm was able to recover the LRF pose with an error below 3 mm presenting the same order of magnitude of the device accuracy. The result obtained for the p-p error is around 50 mm which supports the value of 50 mm chosen for noise STD. An experiment was carried out where the correct pose was given as initial estimate. Result shown an even better precision of 1.2 mm of absolute error and 50.8 mm of p-p error. Scenario B achieved an even better result showing how influent is, for the ICP algorithm, the initial orientation deviation. The voting algorithm have struggled to achieve good results. In the results Figure 4.6 many red spots can be observed. Only the pose 2 has achieve acceptable results, curiously, the test with lower angle displacement to true pose when compared to the other three poses respective scenarios. For pose 1, the upper part of the map has lead the algorithm in error, a plausible and acceptable one as the similarity with the lower part of the map, where the scan was taken from, can be observed. Nevertheless a big red spot can be seen in the lower part indicating a better hypotheses might not be that far. In fact, its associated vote count came in second place in the most voted ranking. As for Scenario D, lines and vertices, for the chosen tolerance values, have been almost correctly extracted. Extracting lines from a cluster with a lower number of points presents a challenging procedure due to the hard task of classification of outliers resulting from the statistical lower significance of the dataset. That is what happened with the lines extracted from the

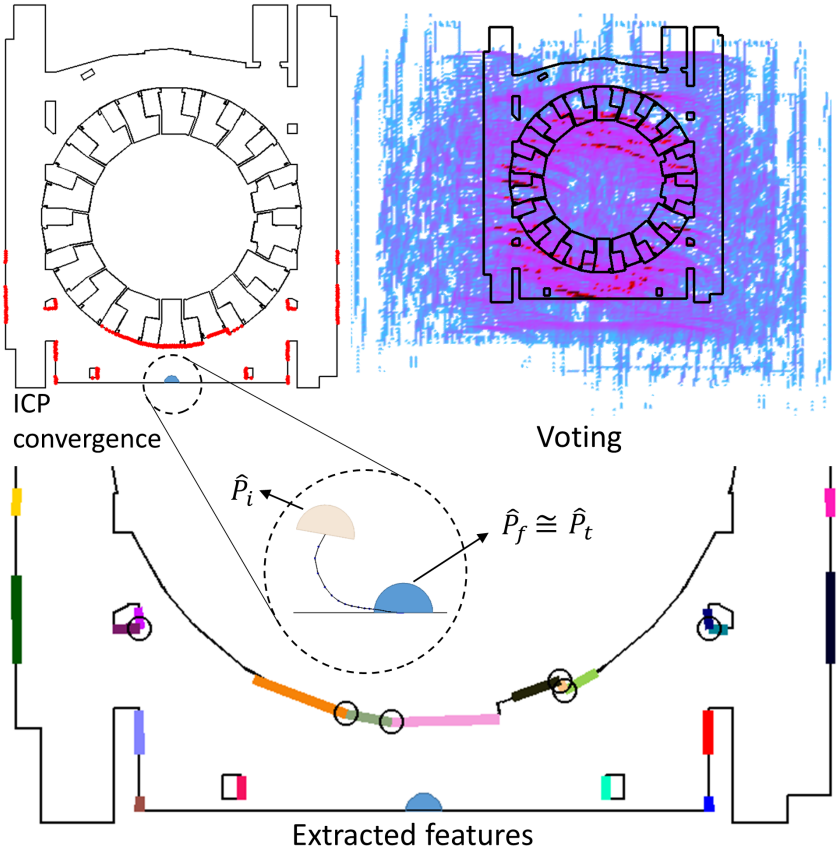


Figure 4.6: Simulation results for ITER map.

left and right columns of the map. The lines presented a deviation of about 5 degrees, which, although not very high, can cause a misleading p-p error evaluation as all points suffer the same deviation. The most difficult lines extraction were the ones from the circular shape walls in the center of the map, in front of the devices. They are composed by small line segments and have a lower value of inclination difference between consecutive segments. In this context lines are harder to extract. Decreasing the slope tolerance value could increase the number of lines by segmentation means all across the map creating false vertices and lines. And as said before, smaller extracted lines are harmful for the matching process. Vertex extraction also depends on correctly extracted lines and their intersection. In this case, the extracted vertices presented a lower error index with only one missing. The occlusion problem caused by the map columns, inhibits the algorithm from extracting the entire lines behind them. In case no vertex could be extracted, the extremes method could take advantage of some of the lines affected by the mentioned problem in the matching process, mitigating this way the occlusion problem. Both the p-p error and absolute error values in scenario D are higher than the scenario A respective results for all four tested poses. This means the result could be improved and justifies the application of a final ICP alignment before returning the final pose estimate.

As an additional experiment, the ICP brute force method described in Section 2.4.4 was applied to the asymmetrical map for pose 1. Both horizontal and vertical resolution of the grid was set to 1 point for every 10 m beginning at 0 m. Since the asymmetrical map is a 100 by 100 m map, it led to a total of 121 initial poses for each of the predefined initial orientations. Those were set to a resolution of 20°

from 0° to 360°, leading to a set of 19 different initial orientations. In total 2299 ICP algorithm executions were carried which took more than 3 hours to complete. The results are shown in Figure 4.7 which is divided into four parts. At the lower left part, the overall results are shown by means of spherical representation where the larger and reddish color of the spheres represent a lower error associated to the ICP initialization pose. The initialization poses used, for each ICP execution, includes the respective positions associated to each sphere location which is given by the x and y axis. The initial orientation is given by the z axis. Results from three layers, each one corresponding to different initial orientation initializations, were extracted from the overall results image, and are depicted in the remaining three parts of the Figure. The lower right image corresponds to an initial orientation estimate of 80°, the upper right to 200° and the upper left image to a 320°. The color map used is the same as the previous Figures where dark and red colors represent a lower p-p error and blue and white color represent a higher p-p error value. Additional images of layers corresponding to 90° and 100° are represented in Figures 4.8a and 4.8b respectably from left to right.

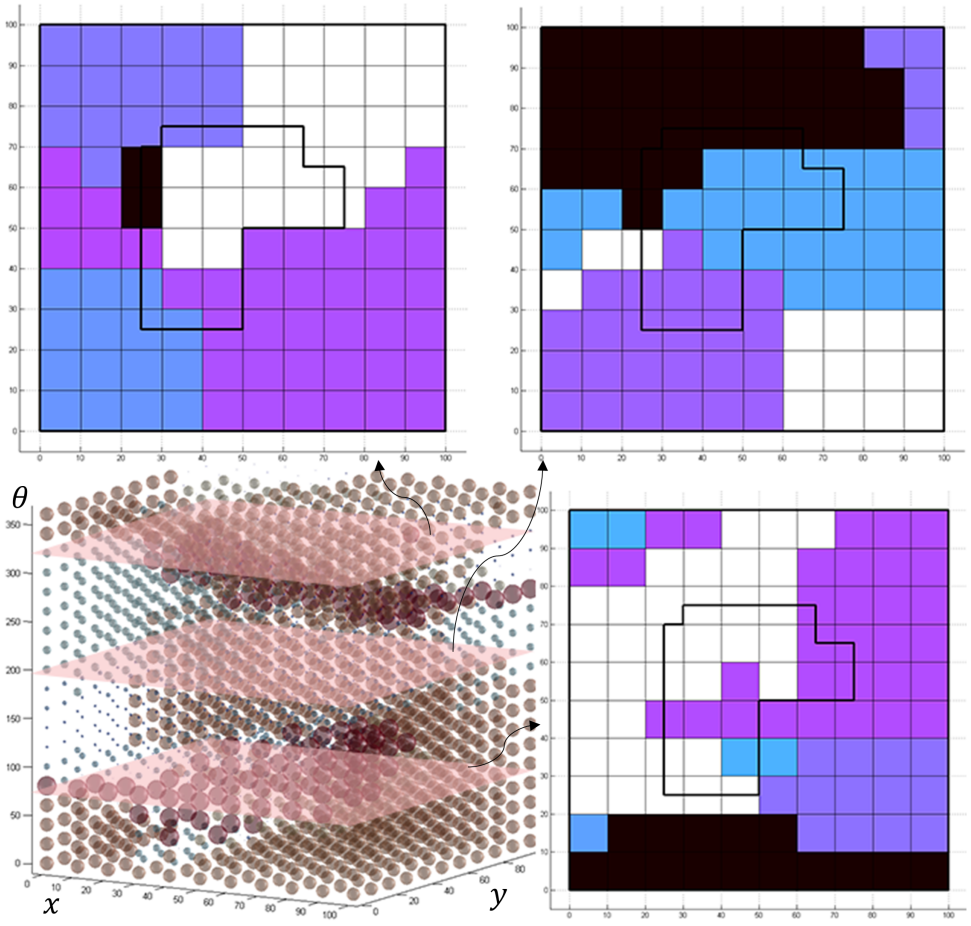
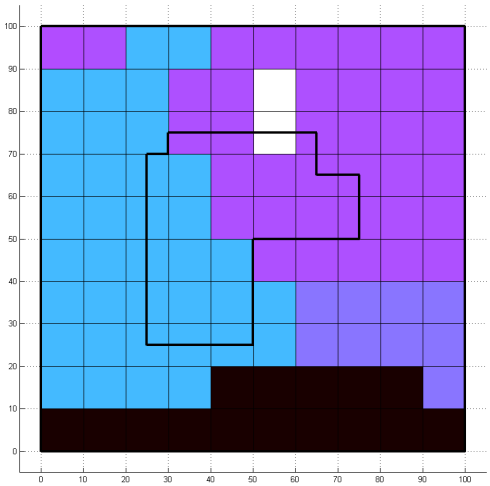
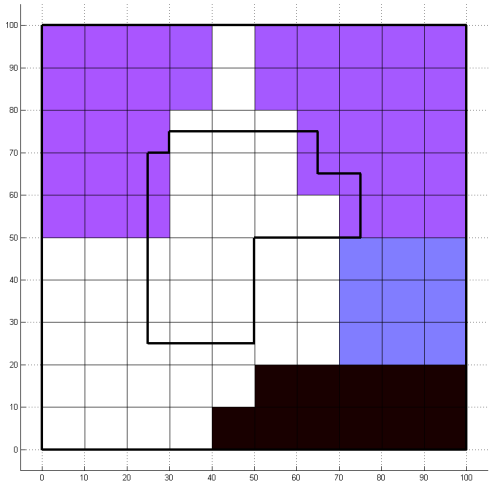


Figure 4.7: ICP brute force result.

Looking at the layers results, each color island formed by the different p-p errors suggests an attraction bay, that means if $\hat{P}_i$ is given inside its limits, the ICP algorithm will always converge for the same local minima. From consulting the respective entry of Table 4.1, it can be observed that the true pose is given by $(75.0, 0.0, 90.0)$. At 80° initialization, once it is close to the true orientation value (90.0°), it



(a) ICP brute force result for 90° layer.



(b) ICP brute force result for 100° layer.

Figure 4.8: Maps developed.

seems reasonable for the algorithm to converge to the correct solution when the initial position is given near the true position. From observation of layers images corresponding to 80°, 90° and 100°, it can be confirmed that ICP present the expected behavior and converges to the correct local minima when near the true position. It confirms, also, the suspicion of the algorithm nonlinearity convergence behavior and high sensibility for different, yet close, initial orientations. With 80° and 90° the algorithm is able to converge to the correct solution from 75 meters away in horizontal, but in vertical, due to the presence of another local minima, which is above the true pose, 15 m deviation in vertical are enough to spoil the results. As it goes from 80° to 100° the convergence bay area above the true pose becomes larger in height and smaller in width. For the 200° case, surprisingly, it can be observed that for the upper part it converges for the correct solution. As a matter of fact, that doesn't happen, because, despite the associated error value being low in absolute terms, it is higher than the error value associated with correct convergences (53.1mm). In this case the color is misleading. If the ICP begins with such a large deviation in orientation value, even near the true pose, there is no way to make it correctly converge from anywhere in the map. For the 320° case, a small black area can be seen. This time the color correctly corresponds to a p-p error value close to 53.1mm. This is due to a lack in the map design as the measures fit correctly the map walls from pose (25.0, 75.0, -90.0). In any case, the walls it matches are impossible to be reached by laser beams because they are behind other walls. Again, with this large deviation in initial orientation, the algorithm could never converge to the correct solution from anywhere.

Table 4.1: Simulation results table.

LRF	Scen.	$\hat{P}_t$ [m, m, °]	$\hat{P}_i$ [m, m, °]	$\hat{P}_f$ [m, m, °]	e_{pp} [mm]	$e_{abs.}$ [mm, °]	RT [s]	CC
Symmetrical map								
1	A	(100.000, 50.000, 180.0)	(110.000, 55.000, 175.0)	(100.185, 49.999, 180.0)	54.5	(185.3, 0.0)	10.3	
	B		(110.000, 55.000, -)	(50.001, 35.047, 90.0)	53.0	(52187.0, 0.0)	33	
	C		(-, -, 175.0)	(98.171, 47.840, 175.0)	1524.3	(2830.1, 5.0)	110	✓/✗
	D		(-, -, -)	(62.819, 50.010, 180.0)	53.0	(37181.2, 0.0)	0.7	✗
2	A	(25.000, 25.000, 0.0)	(40.000, 35.000, 30.0)	(24.997, 24.999, 0.0)	50.4	(3.4, 0.0)	3.8	
	B		(40.000, 35.000, -)	(24.997, 24.999, 0.0)	50.4	(3.3, 0.0)	46	
	C		(-, -, 30.0)	(36.109, 18.714, 30.0)	7534.9	(12764.5, 30.0)	112	✓
	D		(-, -, -)	(75.648, 24.998, 90.0)	525.3	(50648.1, 90.0)	1.4	✗
3	A	(0.000, 10.000, 0.0)	(-12.000, 12.000, -30.0)	(1.011, 10.004, 0.0)	42.8	(1011.2, 0.0)	2.9	
	B		(-12.000, 12.000, -)	(0.431, 10.004, 0.0)	42.8	(430.9, 0.0)	32	
	C		(-, -, -30.0)	(102.728, 109.774, -30.0)	50027.9	(143205.4, 30.0)	111	✗
	D		(-, -, -)	(89.996, 0.002, 90.0)	42.8	(90549.6, 90.0)	0.6	✗
4	A	(50.000, 100.000, -90.0)	(40.000, 90.000, -60.0)	(49.997, 91.971, -90.0)	60.2	(8029.2, 0.0)	2.7	
	B		(40.000, 90.000, -)	(49.999, 90.206, -90.0)	60.2	(9794.4.2, 0.0)	29	
	C		(-, -, -60.0)	(37.040, 76.845, -60.0)	6990.0	(26535.7, 30.0)	115	✓/✗
	D		(-, -, -)	(62.788, 49.999, 180)	60.4	(51608.2, 90.0)	0.7	✗
Asymmetrical map								
1	A	(75.000, 0.000, 90.0)	(73.550,-5.280,73.0)	(75.004, 0.002, 90.0)	53.1	(4.1, 0.0)	4.2	
	B		(73.550, -5.280, -)	(75.008, 0.007, 90.0)	53.1	(10.4, 0.0)	59	
	C		(-, -, 73.0)	(64.689, 0.852, 73.0)	4197.5	(10346.6, 17.0)	216	✓
	D		(-, -, -)	(75.013, 0.011, 90.0)	53.4	(17.4, 0.0)	5	✓
2	A	(20.000, 20.000, 0.0)	(18.000, 19.000, -5.0)	(20.003, 19.998, 0.0)	42.5	(3.6, 0.0)	4.1	
	B		(18.000, 19.000, -)	(20.005, 19.995, 0.0)	42.5	(7.0, 0.0)	59	
	C		(-, -, -5.0)	(19.106, 20.936, -5.0)	1231.4	(1294.5, 5.0)	214	✓
	D		(-, -, -)	(20.008, 19.994, 0.0)	42.7	(6.0, 0.0)	2.8	✓
3	A	(100.000, 75.000, 180.0)	(95.000, 75.000, 194.0)	(100.003, 75.001, 180.0)	51.0	(2.6, 0.0)	4.3	
	B		(95.000, 75.000, -)	(100.001, 74.995, 180.0)	51.0	(7.6, 0.0)	45	
	C		(-, -, 194.0)	(57.951, 51.931, 194.0)	12899.2	(47961.3, 14.0)	215	✗
	D		(-, -, -)	(99.982, 75.030, 180.1)	55.8	(34.5, 0.1)	4.7	✓
4	A	(25.000, 100.000, -90.0)	(26.000, 99.000, -106.0)	(25.010, 100.001, -90.0)	52.4	(9.8, 0.0)	5.1	
	B		(26.000, 99.000, -)	(25.010, 100.001, -90.0)	52.4	(9.8, 0.0)	44	
	C		(-, -, -106.0)	(58.136, 79.047, -106.0)	11618.6	(39205.4, 16.0)	209	✗
	D		(-, -, -)	(24.946, 99.984, -90.0)	65.9	(56.6, 0.0)	4.9	✓
ITER map								
1	A	(33.750, 1.000, 90.0)	(32.750, 2.000, 80.0)	(33.747, 1.000, 90.0)	50.8	(2.6, 0.0)	3.2	
	B		(32.750, 2.000, -)	(33.751, 1.000, 90.0)	50.8	(1.3, 0.0)	48	
	C		(-, -, 80.0)	(44.366, 59.367, 80.0)	5106.6	(59324.7, 10.0)	276	✗
	D		(-, -, -)	(33.713, 0.989, 90.0)	82.8	(38.5, 0.0)	33	✓
2	A	(0.000, 34.750, 0.0)	(1.000, 40.000, 10.0)	(-0.007, 34.747, 0.0)	46.5	(7.8, 0.0)	2.7	
	B		(1.000, 40.000, -)	(-0.007, 34.747, 0.0)	46.5	(7.8, 0.0)	49	
	C		(-, -, 10.0)	(0.641, 28.020, 10.0)	2219.2	(6760.0, 10.0)	274	✓
	D		(-, -, -)	(0.021, 34.742, 0.0)	79.0	(22.5, 0.0)	49	✓
3	A	(33.750, 68.950, -90.0)	(36.000, 67.000, -105.0)	(33.754, 68.957, -90.0)	43.2	(7.9, 0.0)	4.5	
	B		(36.000, 67.000, -)	(33.751, 68.956, -90.0)	43.2	(6.3, 0.0)	55	
	C		(-, -, -105.0)	(50.113, 73.813, -105.0)	3868.3	(17070.0, 15.0)	278	✗
	D		(-, -, -)	(33.738, 68.947, -89.9)	75.8	(12.6, 0.1)	50	✓
4	A	(67.500, 34.750, 180.0)	(55.000, 25.000, 185.0)	(67.501, 34.746, 180.0)	48.6	(4.7, 0.0)	3.2	
	B		(55.000, 25.000, -)	(67.501, 34.746, 180.0)	48.6	(4.7, 0.0)	46	
	C		(-, -, 185.0)	(67.551, 37.385, 185.0)	1139.8	(2635.1, 5.0)	287	✓
	D		(-, -, -)	(67.455, 34.715, 180.0)	87.5	(57.1, 0.0)	44	✓

4.2 Field Results

4.2.1 Maps used

Three distinct maps were used to test the algorithm in real scenarios. A manual topographic survey was done for both maps so it is possible that the maps representation is not that accurate. An error value higher than the sensor accuracy is expected. The first map, represented in Figure 4.9, is the map from a house balcony which includes a rectangular shape object near a wall, a tube on a upper wall, an aluminum rail and glass surface on another wall and a carpet on the longest wall. Only the rectangular shape object is considered in the map description leaving the other things unmapped. These objects were intended to make it harder for the algorithm to find a solution at the same time testing robustness. The second one, represented in Figure 4.10, is the map from Instituto de Plasmas e Fusão Nuclear (IPFN) research laboratories at IST. It is a corridor composing a common office environment with closed doors by its sides and long walls. The third map in Figure 4.11 is from a garage featuring wide empty spaces and a bigger area than the two previous maps. This map features multiple outliers and a worse map description accuracy when compared to the other two maps. Both maps were converted to points with a density of 50 points per meter except for the voting tests where the density was changed to 5 in order to the algorithm execution terminate in a feasible time.

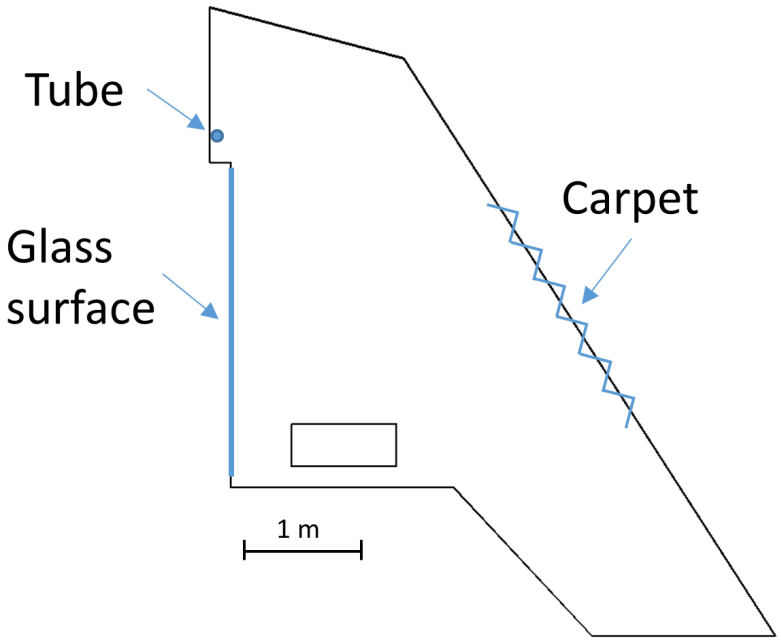


Figure 4.9: House balcony map.

Figure 4.10: IPFN corridor map.

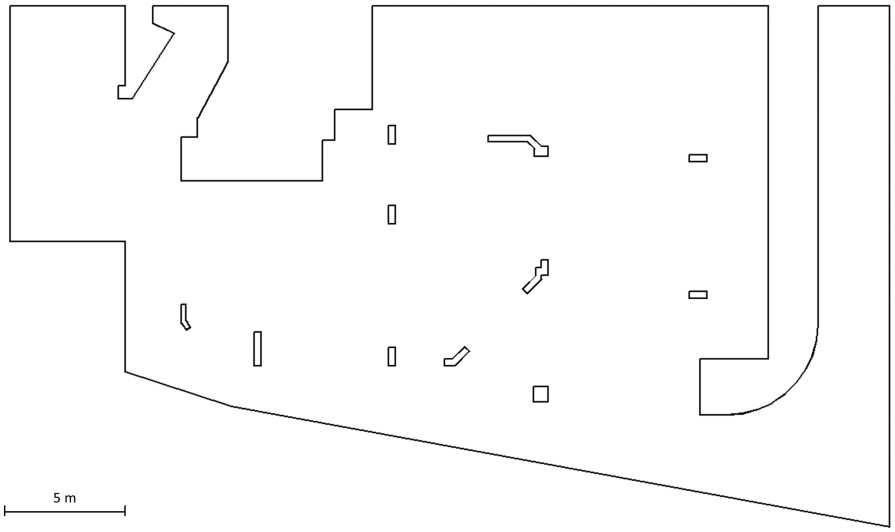


Figure 4.11: Garage map.

4.2.2 Field Experiments Description

Two different devices were used according to the appropriate requirements. For the IPFN and balcony maps the LRF device used was a Hokuyo URG-04LX-UG01 (see Figure 4.12) due to the lower range needs. The following main specifications were taken from the sensor manual [40]:

- FOV: 240°;
- Accuracy: 0.06-1 m: ±30 mm , 1-4 m: 3% of the detected distance;
- Max range: 4 m;
- Linear resolution: 1 mm;
- Angular resolution: 0.36°.

For the garage map, which presents a wider area, the LRF device used was a SICK LMS 200 which features a higher maximum range. The following main specifications were taken from the respective device manual [41]:

- FOV: 180°;
- Accuracy: ±15 mm;
- Max range: 80 m;
- Linear resolution: 10 mm;
- Angular resolution: 1°;
- Systematic error: typical ±4 cm at range 1 to 20 m.

For the repetition of the balcony map tests, which has a smaller area than the garage map, the SICK LRF device was configured to mm mode which limits maximum range to 8 meters and decreases systematic errors. The following main specifications were taken from the respective device manual [41]:

- FOV: 180°;
- Accuracy: ±15 mm;
- Max range: 8m;
- Linear resolution: 10mm;
- Angular resolution: 1°;
- Systematic error: typical ±15 mm at range 1 to 8 m.

Both LRF sensor devices, one after another and never at the same time, were set in the predefined and measured poses for each map. The chosen predefined poses and the respective number identifications, are shown in Figures 4.12, 4.13, and 4.14 for the balcony, IPFN and garage map respectively. Since only one device of each of the models was available, the scans were taken from different time intervals and not simultaneously. Nevertheless it was guaranteed that the environment didn't change between the scans. At least, one hundred consecutive scans were taken for each pose in order to have a large enough dataset for posterior process, in particular for rejecting discrepant value measurements. Again, the four possible scenarios (A, B, C, and D) were tested for each pose. Using Hokuyo LRF, in scenario D, the tolerance values that gave the best results, for the clustering and feature extraction were the following:

- proximity: 0.10m;
- slope: 15°;
- cluster: 0.10m;
- split: 0.050m.

Using Sick LRF, in scenario D, the tolerance values that gave the best results, for the clustering and feature extraction at the garage map were the following:

- proximity: 0.40m;
- slope: 15°;
- cluster: 0.10m;
- split: 0.20m.

Instead, for the balcony map, using Sick LRF, in scenario D, the tolerance values used were the same as the Hokuyo ones stated before.

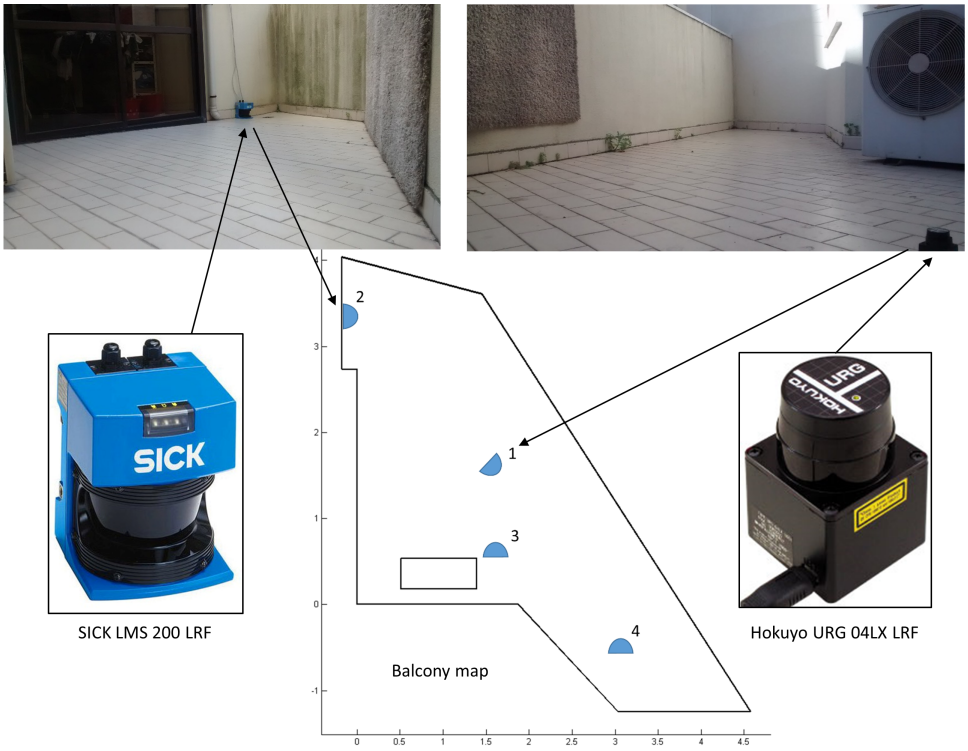


Figure 4.12: Balcony map test locations.

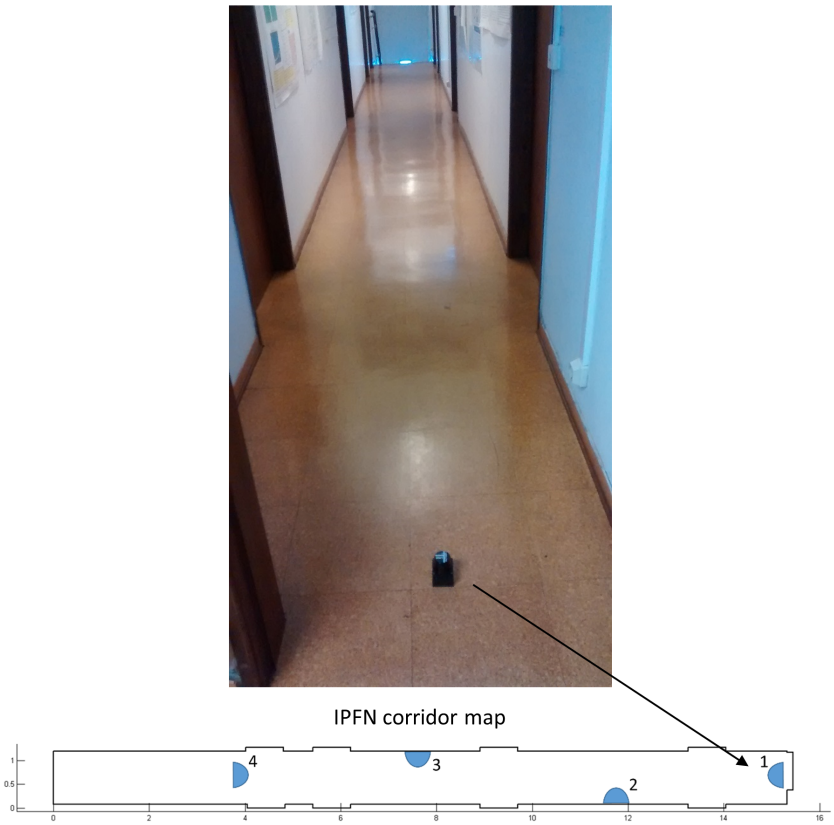


Figure 4.13: IPFN map test locations.

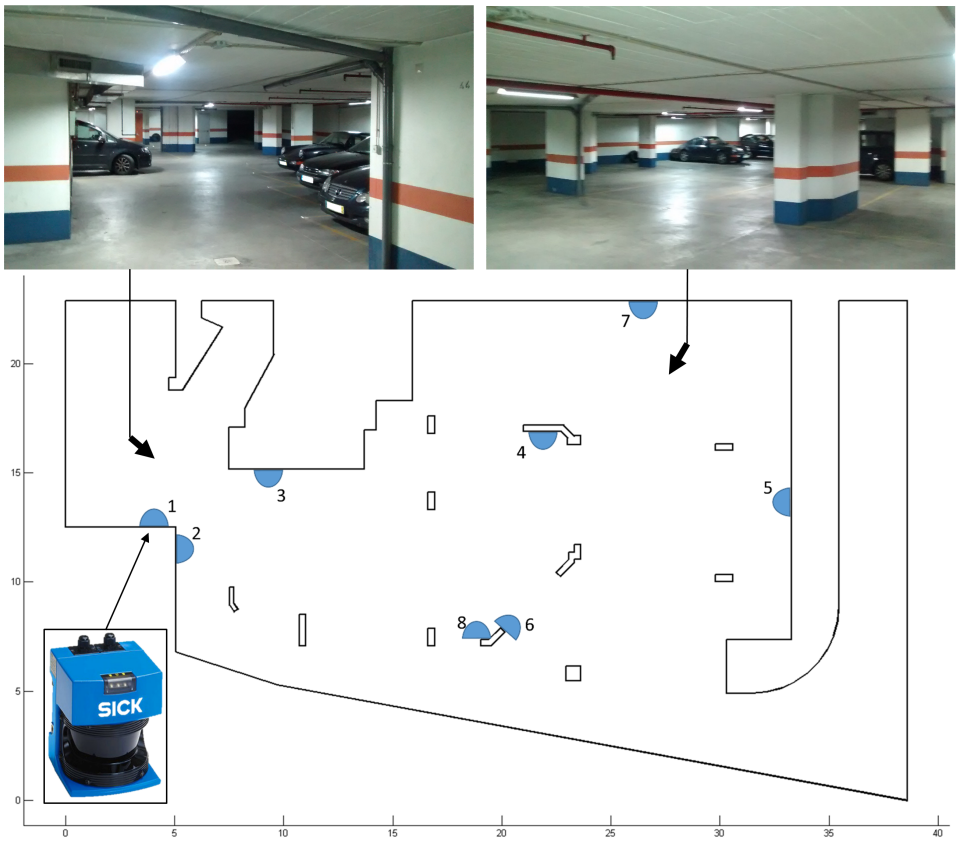


Figure 4.14: Garage map test locations.

4.2.3 Field tests results

The results obtained for the field tests are presented in Tables 4.2 and 4.3 for the Hokuyo and SICK devices respectively. The columns of these Tables are identical to the columns of the simulated results Table explained before except for $\hat{P}_m$. $\hat{P}_m$ means pose measurement, consisting the only way to first guess the initial pose (by hand measuring) so a relative error column ("$e_{rel.}$") was introduced. Hereupon $\hat{P}_m$ is always equal to $\hat{P}_i$ for all the field tests. Images of the Hokuyo results are displayed in Figures 4.15 and 4.17 for the balcony and IPFN maps respectively. As for the SICK device, its results are displayed in Figures 4.18 and 4.16 for the garage and balcony maps respectively. All the images illustrate the pose 1 results and follow the same layout as the simulation Figures stated above.

Hokuyo LRF device accuracy ranges from 30 to 120 mm for a sheet of white paper target as stated in the respective manual. Measurements were taken outside on daylight which could affect sensor performance as stated in Section 3.1, and a manual topographic survey was done to extract the map description measurements. Because of the used tool uncertainty, in best case scenario, the map accuracy is ±0.5 mm. In this case, it is more likely for the map accuracy to be around the ±5 mm of accuracy. Also the objects referred in Section 4.2.1 for this map were not represented in the map description on purpose, so they could be classified as outliers. Other minor outliers and undesirable factors, like target roughness or LRF inclination, may exist inherent from the real-world environment that were not possible to mitigate making impractical to perfectly represent the scenario with such level of detail.

In Field tests case, the relative error results are meaningless to evaluate the algorithms performance

since $\hat{P}_m$ is itself a measure subject to error. Having said that, p-p error is the measure considered for the effect. The exception goes for the relative error results that exhibit absurd values i.e. more than one meter away from $\hat{P}_m$.

Looking at balcony map tests, for scenario A, results show a p-p error from 20 to 44 mm which is congruent with map dimensions and device accuracy. They are, indeed, good results considering the presence of outliers. For pose 1, from the ICP result image, outliers can be identified by the lower left points which seem to have low influence on the results mostly explained by the large wall attraction. At the most distant set of readings (far bottom right section), it can be observed that the readings gradually become erroneous as they approach the 4 m maximum range limit of the device. On overview, there are some places on the map where the measurements don't fit as well as other places do, implying errors in one of the parts (map or laser) or both parts.

For scenario B, nearly the same results, as scenario A, were obtained for every four poses due to lower difference between initial orientation and final resultant orientation estimate. Voting algorithm results, for scenario C, shown an overall acceptable p-p error as well as a relative error values lower than 1 meter for every tested pose. This method is, by far, the most time consuming of the four, taking about 42 second to terminate the execution.

In scenario D, the vertex method was able to correctly extract correct vertices and among eventual erroneous ones except for pose 4. In this case the algorithm wasn't able to extract any vertex so the extremes method took place delivering a low error result. For pose 1, in the middle of the large wall, an outlier point in the readings caused by the carpet can be spotted in the respective results images of Figure 4.15. Turns out that this outlier zone didn't prevent the algorithm from correctly extract the

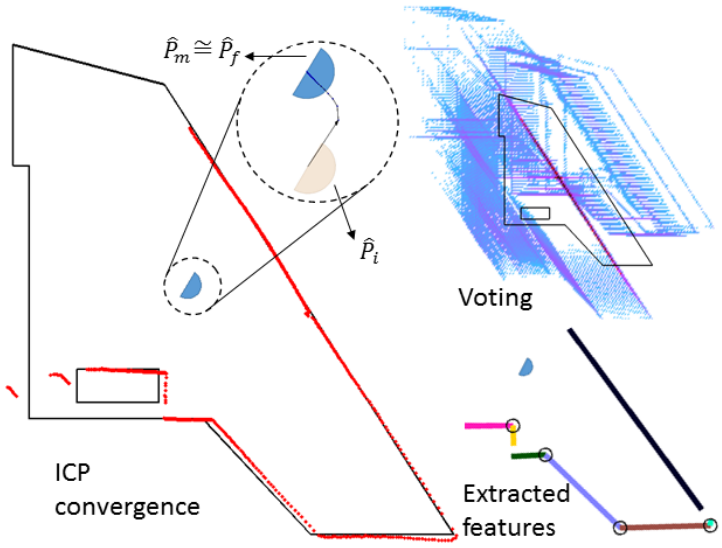


Figure 4.15: Balcony map experimental results for Hokuyo LRF.

respective line segment. Only the lower end extreme point of the line segment was wrongly extracted due to the increasing distance between points. This is a common problem of scans of long oblique walls. The bottom wall was partially correctly extracted as in the right end a small segment line was wrongly extracted creating by chance, an almost correct vertex. Beside those cases, the remaining lines

and vertices were correctly extracted. The vertex and line that gave the best results, presented in the mentioned Table, were the ones from the rectangle box of the map.

The SICK results for this same map are presented in Table 4.3 and illustrated in images of Figure 4.16 that follows the same layout as the previous one. The four test locations, where the scans were taken from, were approximately the same of the Hokuyo device case. The SICK device has lower resolution, lower number of measurements and lower FOV but an overall better accuracy when compared to Hokuyo device. Comparing pose 1 results, SICK was able to achieve a lower p-p error, in scenario A, mainly because with a lower FOV the outliers in lower left part of the map were not caught up in the scan. In scenario A, for the remaining 3 poses, a higher p-p error was obtained. This could be expected as, in this case, fewer correct points exists among the outliers meaning less correct points to contribute to a lower p-p error. Pose 2 and 4 stand out for the worst results of the four tests. The readings from pose 2 were the less accurate ones and pose 4 readings show that the measurements went beyond the glass surface. On the other hand, pose 3 scanned the entire glass surface. This phenomenon confirms the target reflectivity level required for different distances, characteristic of this device as stated in Sections 3.1 and 3.2. For scenario B the previous situation holds up. For the C scenario, a relatively

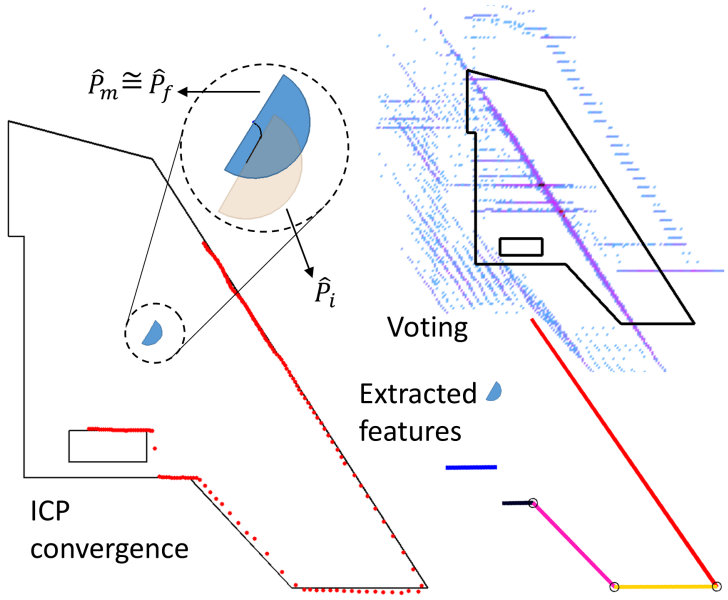


Figure 4.16: Balcony map experimental results for SICK LRF.

lower p-p error was achieved for every test together with acceptable and lower relative error. The same middle line pattern can be observed from both respective resultant voting images. For the D scenario, it is worth mentioning the line extraction for pose 2. In pose 2 case, as said before, with less accurate measurements, it should be expect to compromise the line extraction phase. Contrary to what would be expected, not only the lines were correctly extracted, also the p-p error was quite close (difference of 0.1 mm) to the respective ICP result. Figure A.9 in the Appendix Section A.2, illustrates and proves what was said. For the remaining tests, lines and vertex extracted produced reasonable results indicating a robust choice for tolerance values.

The IPFN map is an indoor long corridor composed only by walls and doors identified by the occa-

sional recess spaces. There were no intentional outliers present on the environment this time. The map is highly symmetric in the vertical direction except for the right end and a slight misalign bottom leftmost door. On the horizontal direction the map is less symmetric due to the different space between doors. Featuring roughly 15 m of length, it was impossible, on most of the tests, for the Hokuyo to reach the side walls. A stairway is present in the left side of the map, which begins at 4 m making impossible to take scans from that place.

Results of Table 4.2 show, for scenario A, an overall lower p-p error in every test when compared to the balcony map. This can be explained by the average lower distance measurements and consequentially lower associated error uncertainty (which is proportionally dependent to measurement values as have been seen above on the device properties). Pose 1 and 2 scanned the unique reachable perpendicular wall (right wall). As for the others, they didn't share the same luck. Despite presenting low p-p error values, it seems reasonable to think that if it was not for the close to truth $\hat{P}_i$, the chance was high for the algorithm to be stuck in a local minima. For scenario B pretty much the same scenario A results were obtained. Voting algorithm results (for scenario C) show an overall acceptable p-p error but a relative error larger than one meter for every pose which makes the alarm bell sound, despite, in pose 1 case, not being far from the ICP result. And rightly so, a horizontal line pattern can be observed from the respective pose 1 results image of Figure 4.17. Indeed, a similar situation has been observed for the symmetrical case in Figure 4.4 confirming the suspicion that symmetrical issues might interfere with the results. Neither tests gave an acceptable results because of the respective high relative error

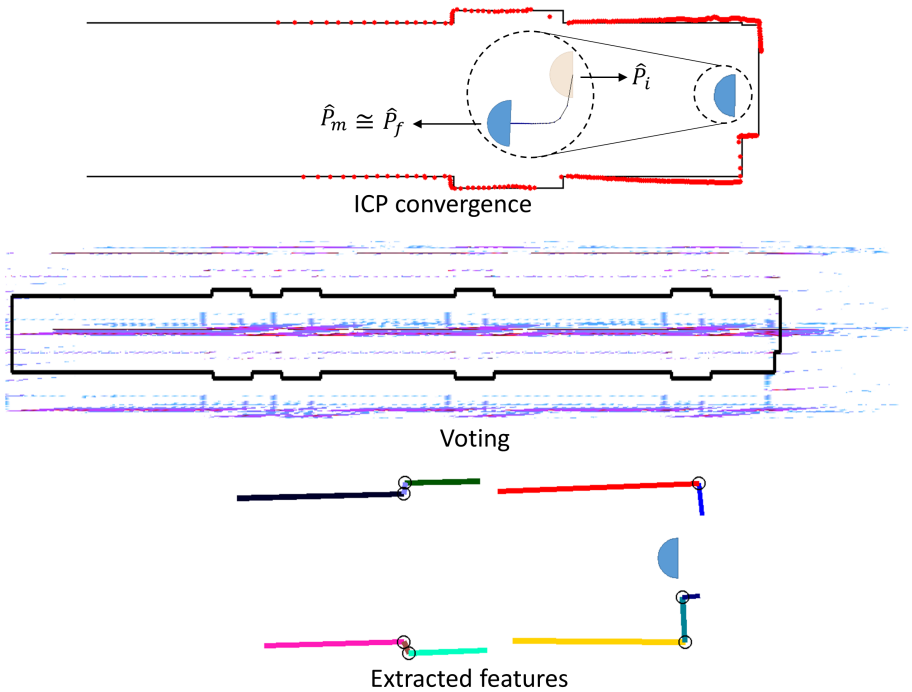


Figure 4.17: IPFN map experimental results.

value. Resolution is the suspect in this case because with 5 points per meter yield about 5 points to the vertical line which is, in this case, a key factor to disambiguate the symmetry problem. In door jambs the situation is worse due to its low lengths. In scenario D, lines and vertices were mostly correctly ex-

tracted. A mixture of long walls and small ones (from the doors and from their jambs), together with the occlusion problem, makes the task of finding the best tolerance values harder. Unlike the small walls, as stated before, the long walls present a valuable chance to extract accurate line segments. In effect, the p-p error for pose 1 only varies 0.1 mm from the scenario A result. The other tests results maintain low errors tendency, specially the pose 4 with a 0.3 mm p-p error difference from scenario A.

The garage map is the largest of the three maps from the experimental results. Given its dimension, Hokuyo LRF wouldn't fit well unless the object of interest were small areas of it, which is not the case. The intent is to have a large area to scan. The map was built with the aid of the garage blueprint schematic. If the description could be trusted, there is, at most, an error of 5 mm. As it can be seen in the photos that describe the place, in Figure 4.14, there are some cars parked in the garage. To mitigate the outliers cause by the cars, scans were taken from high enough for the laser rays to be able to pass above the cars. Despite this attempt, there was one tall car (a jeep) that couldn't be avoided. Other outliers such as air ducts, cabinets, water pipes, electric cables, door recesses or fire extinguishers, couldn't be avoided as well. Adding to this, there exists a possibility of a minor device inclination during the scan, which could have tampered the dataset. Right from pose 1 results observation from Figure 4.18, it can be noticed a mismatching zone at the top right center area. There were also some points with

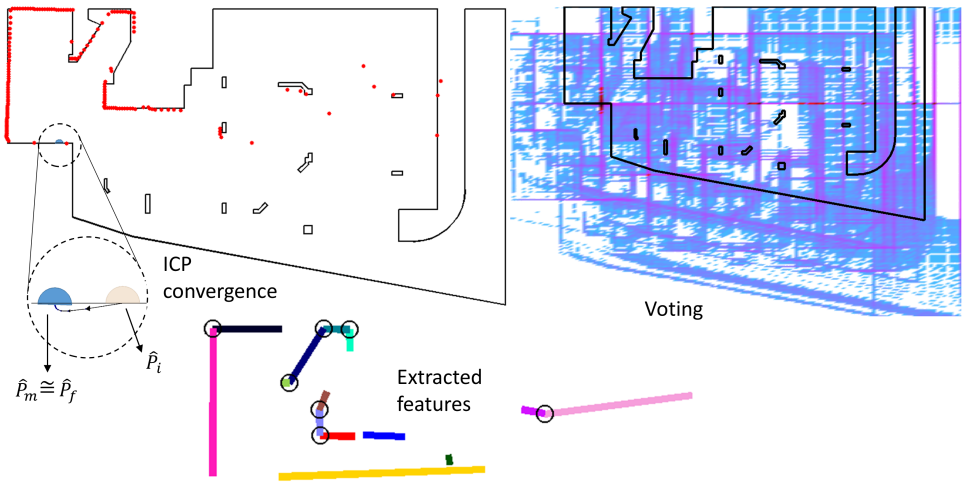


Figure 4.18: Garage map experimental results.

no correspondence on the right. For the other results, rarely the columns at the center of the environment have aligned correctly. An average of about 35 cm of p-p error in scenario A can be observed, a value higher than any other test. But this result comes with no surprise because long measurements are more susceptible to higher error with the aggravating circumstance of LRF configuration to cm mode which has a higher systematic error. The lowest p-p error cases are associated with the lower range measurements of long walls, like poses 1, 2 and 3. Scenario B gave the expected results as usual. Scenario C results show a p-p error always bigger but relatively closer to Scenario A p-p errors. The same is not true for the relative error column which presents acceptable results, mostly much lower than 1 meter. As for geometric feature extraction, in scenario D, error results are on par with the ones from other scenarios. The p-p error is never lower than scenario A, but the relative error indicates the results

are acceptable. Pose 1 respective results image show two lines wrongly extracted which didn't have any influence on the final result. Pose 5 scan is the most affect by the jeep outliers but that didn't stop the vertex method from achieving relative low error results.

Table 4.2: Field results table for Hokuyo LRF.

LRF	Scen.	$\hat{P}_m$ [m, m, °]	$\hat{P}_i$ [m, m, °]	$\hat{P}_f$ [m, m, °]	e_{pp} [mm]	$e_{rel.}$ [mm, °]	RT [s]	CC
Balcony map								
1	A	(1.700, 1.400, 330.0)	(1.700, 1.400, 330.0)	(1.699, 1.453, 329.2)	44.6	(52.5, 0.8)	0.39	
	B		(1.700, 1.400, -)	(1.699, 1.452, 329.2)	44.6	(52.5, 0.8)	4.3	
	C		(-, -, 330.0)	(1.906, 1.117, 330.0)	139.2	(349.7, 0.0)	42	✓
	D		(-, -, -)	(1.599, 1.490, 330.7)	70.8	(134.8, 0.7)	1.4	✓
2	A	(0.000, 3.400, 0.0)	(0.000, 3.400, 0.0)	(-0.115, 3.392, 0.6)	38.1	(115.5, 0.6)	0.16	
	B		(0.000, 3.400, -)	(-0.115, 3.392, 0.6)	38.1	(115.5, 0.6)	3.7	
	C		(-, -, 0.0)	(-0.119, 3.398, 0.0)	73.9	(118.5, 0.0)	42	✓
	D		(-, -, -)	(-0.085, 3.365, 1.6)	49.4	(91.9, 1.6)	1.1	✓
3	A	(1.600, 0.600, 90.0)	(1.600, 0.600, 90.0)	(1.555, 0.539, 85.6)	35.8	(75.8, 4.4)	0.18	
	B		(1.600, 0.600, -)	(1.556, 0.537, 85.6)	35.8	(76.4, 4.4)	2.9	
	C		(-, -, 90.0)	(2.008, 0.021, 90.0)	200.3	(707.9, 0.0)	42	✗
	D		(-, -, -)	(1.561, 0.569, 84.8)	46.8	(50.1, 5.2)	1.6	✓
4	A	(3.000, -0.300, 90.0)	(3.000, -0.300, 90.0)	(2.981, -0.307, 87.9)	20.5	(20.3, 2.1)	0.24	
	B		(3.000, -0.300, -)	(2.981, -0.307, 87.9)	20.5	(20.3, 2.1)	6.1	
	C		(-, -, 90.0)	(2.885, -0.180, 90.0)	81.3	(166.3, 0.0)	42	✓
	D		(-, -, -)	(2.916, -0.194, 87.5)	35.7	(134.9, 2.5)	0.91	✓
IPFN map								
1	A	(15.300, 0.700, 180.0)	(15.300, 0.700, 180.0)	(15.271, 0.665, 178.7)	26.3	(45.6, 1.3)	0.29	
	B		(15.300, 0.700, -)	(15.269, 0.665, 178.7)	26.3	(47.3, 1.3)	6.2	
	C		(-, -, 180.0)	(13.385, 0.707, 180.0)	154.8	(1914.8, 0.0)	93	✗
	D		(-, -, -)	(15.278, 0.665, 178.7)	26.4	(41.9, 1.3)	9.4	✓
2	A	(11.750, 0.100, 90.0)	(11.750, 0.100, 90.0)	(11.777, 0.103, 87.2)	12.1	(27.2, 2.8)	0.38	
	B		(11.750, 0.100, 90.0)	(11.780, 0.103, 87.2)	12.0	(30.2, 2.8)	7.9	
	C		(-, -, 90.0)	(15.366, 1.193, 90.0)	1552.9	(3777.3, 0.0)	92	✓
	D		(-, -, -)	(11.742, 0.120, 86.8)	23.1	(21.6, 3.2)	4.0	✓
3	A	(7.550, 1.150, -90.0)	(7.550, 1.150, -90.0)	(7.582, 1.162, -91.8)	12.0	(34.3, 1.8)	0.16	
	B		(7.550, 1.150, -90.0)	(7.581, 1.164, -91.8)	12.0	(33.7, 1.8)	8.3	
	C		(-, -, -90.0)	(0.104, 0.057, -90.0)	1184.7	(7525.6, 0.0)	93	✓/✗
	D		(-, -, -)	(7.622, 1.145, -91.8)	23.5	(71.8, 1.8)	2.6	✓
4	A	(3.900, 0.700, 0.0)	(3.900, 0.700, 0.0)	(3.895, 0.731, -2.5)	18.3	(31.8, 2.5)	0.27	
	B		(3.900, 0.700, 0.0)	(3.895, 0.731, -2.5)	18.3	(31.8, 2.5)	8.7	
	C		(-, -, 0.0)	(14.059, -0.482, 0.0)	916.4	(10227.2, 0.0)	92	✓/✗
	D		(-, -, -)	(3.880, 0.731, -2.6)	18.6	(37.5, 2.6)	13.5	✓

Table 4.3: Field results table for SICK LRF.

LRF	Scen.	$\hat{P}_m$ [m, m, °]	$\hat{P}_i$ [m, m, °]	$\hat{P}_f$ [m, m, °]	e_{pp} [mm]	$e_{rel.}$ [mm, °]	RT [s]	CC
Balcony map								
1	A	(1.400, 1.600, 330.0)	(1.700, 1.400, 330.0)	(1.411, 1.655, 328.1)	24.8	(55.6, 1.9)	0.06	
	B		(1.700, 1.400, -)	(1.411, 1.655, 328.1)	24.8	(55.6, 1.9)	1.0	
	C		(-, -, 330.0)	(1.419, 1.648, 330.0)	80.5	(51.6, 0.0)	24.2	✓
	D		(-, -, -)	(1.382, 1.718, 328.1)	46.3	(118.9, 1.9)	0.8	✓
2	A	(0.000, 3.400, 0.0)	(0.000, 3.400, 0.0)	(0.015, 3.305, 3.2)	91.1	(96.0, 3.2)	0.07	
	B		(0.000, 3.400, -)	(0.015, 3.305, 3.2)	91.1	(96.0, 3.2)	1.6	
	C		(-, -, 0.0)	(-0.246, 3.397, 0.0)	208.8	(246.5, 0.0)	23.8	✓
	D		(-, -, -)	(0.054, 3.292, 2.9)	92.2	(120.8, 2.9)	1.1	✓
3	A	(1.600, 0.600, 90.0)	(1.600, 0.600, 90.0)	(1.708, 0.679, 90.1)	39.5	(134.2, 0.1)	0.07	
	B		(1.600, 0.600, -)	(1.708, 0.679, 90.1)	39.5	(134.2, 0.1)	1.3	
	C		(-, -, 90.0)	(1.696, 0.673, 90.0)	72.5	(120.8, 0.0)	23.8	✓
	D		(-, -, -)	(1.744, 0.662, 91.3)	46.6	(156.8, 1.3)	1.2	✓
4	A	(3.000, -0.300, 90.0)	(3.000, -0.300, 90.0)	(3.352, -0.622, 88.4)	78.0	(477.4, 1.6)	0.14	
	B		(3.000, -0.300, -)	(3.352, -0.622, 88.4)	78.0	(477.4, 1.6)	1.9	
	C		(-, -, 90.0)	(3.444, -0.731, 90.0)	110.4	(618.8, 0.0)	23.8	✓/✗
	D		(-, -, -)	(3.328, -0.579, 90.0)	92.3	(430.6, 0.0)	1.0	✓
Garage map								
1	A	(4.300, 12.500, 90.0)	(4.300, 12.500, 90.0)	(4.000, 12.494, 88.3)	352.9	(300.5, 1.7)	0.4	
	B		(4.300, 12.500, -)	(4.000, 12.494, 88.3)	352.9	(300.5, 1.7)	8.1	
	C		(-, -, 90.0)	(3.969, 12.490, 90.0)	401.7	(331.4, 0.0)	299	✓/✗
	D		(-, -, -)	(4.075, 12.431, 89.5)	370.9	(235.4, 0.5)	10.5	✓
2	A	(5.050, 11.500, 0.0)	(5.050, 11.500, 0.0)	(5.107, 11.778, -0.1)	208.2	(285.0, 0.1)	0.18	
	B		(5.050, 11.500, -)	(5.107, 11.778, -0.1)	208.2	(284.0, 0.1)	6.5	
	C		(-, -, 0.0)	(5.161, 11.691, 0.0)	226.0	(221.2, 0.0)	268	✓
	D		(-, -, -)	(5.149, 11.732, 0.3)	221.5	(252.0, 0.3)	3.2	✓
3	A	(8.500, 15.170, -90.0)	(8.500, 15.170, -90.0)	(8.631, 15.129, -90.9)	213.0	(137.4, 0.9)	0.38	
	B		(8.500, 15.170, -)	(8.631, 15.130, -90.9)	213.0	(137.4, 0.9)	7.1	
	C		(-, -, -90.0)	(8.656, 14.893, -90.0)	272.8	(317.9, 0.0)	265	✓
	D		(-, -, -)	(8.618, 15.033, -89.6)	273.3	(180.5, 0.4)	7.7	✓
4	A	(22.000, 16.900, -90.0)	(22.000, 16.900, -90.0)	(22.313, 16.777, -91.0)	402.6	(335.8, 1.0)	0.39	
	B		(22.000, 16.900, -)	(22.313, 16.777, -90.1)	402.6	(335.8, 0.1)	6.3	
	C		(-, -, -90.0)	(22.171, 16.885, -90.0)	424.9	(171.4, 0.0)	259	✓
	D		(-, -, -)	(22.518, 16.963, -92.1)	436.5	(521.4, 2.1)	6.3	✓/✗
5	A	(33.300, 14.300, 180.0)	(33.300, 14.300, 180.0)	(33.174, 13.984, 179.2)	379.0	(339.9, 0.8)	0.35	
	B		(33.300, 14.300, -)	(33.174, 13.984, 179.2)	379.0	(339.8, 0.8)	6.8	
	C		(-, -, 180.0)	(33.191, 14.251, 180.0)	419.4	(119.2, 0.0)	261	✓
	D		(-, -, -)	(33.017, 14.264, 177.8)	546.2	(125.6, 2.2)	3.3	✓
6	A	(20.000, 7.800, 45.0)	(20.000, 7.800, 45.0)	(20.222, 7.342, 45.4)	545.7	(508.9, 0.4)	0.36	
	B		(20.000, 7.800, -)	(20.222, 7.342, 45.4)	545.7	(508.9, 0.4)	7.2	
	C		(-, -, 45.0)	(20.055, 6.993, 45.0)	609.8	(808.5, 0.0)	265	✓
	D		(-, -, -)	(20.340, 7.177, 46.0)	561.4	(709.4, 1.0)	14.7	✓
7	A	(26.300, 22.700, -90.0)	(26.300, 22.700, -90.0)	(26.319, 22.375, -90.1)	352.4	(325.7, 0.1)	0.35	
	B		(26.300, 22.700, -)	(26.319, 22.375, -90.1)	352.4	(325.7, 0.1)	7.2	
	C		(-, -, -90.0)	(26.242, 22.234, -90.0)	372.5	(470.0, 0.0)	259	✓
	D		(-, -, -)	(26.400, 22.433, -89.5)	381.6	(285.4, 0.5)	4.7	✓
8	A	(19.250, 7.370, 90.0)	(19.250, 7.370, 90.0)	(19.423, 7.063, 92.1)	406.0	(352.6, 2.1)	0.44	
	B		(19.250, 7.370, -)	(19.423, 7.063, 92.1)	406.0	(352.6, 2.1)	5.6	
	C		(-, -, 90.0)	(19.461, 7.051, 90.0)	462.5	(382.8, 0.0)	263	✓
	D		(-, -, -)	(19.557, 6.959, 91.4)	440.5	(513.0, 1.4)	8.9	✓

4.3 Robustness tests

Some robustness tests were carried to evaluate the algorithms permeability to error and variation in input data. Scan data and the map are the two inputs, besides the previously tested initial pose estimate, that are subject to uncertainty. To simulate variations in each of the two inputs at stake (scan data and the map), two different approaches were taken while maintaining the same environment. For the first, to produce variations in scan data, foreign objects were introduced in the map. Then, scans were taken on this tampered environment followed by the matching phase that was performed using the original unmodified map. For the second, variations in the map description where introduced for each of the tests while maintaining the correct scan data across all tests. No new objects were introduced or taken out of the different maps, as this would lead to the first case. Instead, some structures were slightly deviated from their original place in some random directions in order to simulate the changes of map description. The performance results of the ICP, voting, vertex and extremes algorithms are shown in Tables 4.4 and 4.5 respectively for scan outliers (first approach described) and map changes (second approach described) . Every ICP algorithm execution began at the correct pose to mitigate error possibility. Also, the voting method was initialized with the correct orientation value, in addition to a higher (3 points per meter) resolution to improve and better assess results. Both Tables columns are similar. The variations column indicates three values that measure the variations between the affected environment and the original one. The first value, var_n, is the percentage of the number of affected points pts_a of the total number of points, pts_t, and is given by (4.1). The affected points are defined and selected by direct observation and comparison between the correct scan and the affected one.

$$var_n = \frac{pts_a}{pts_t} \times 100 \tag{4.1}$$

The second value measures the mean global variation, i.e. the relative variation between what the scan should have been (pts_c is the correct scan distance measures) and the actual tampered scan. It is given by (4.2).

$$var_g = \frac{|pts_c - pts_a|}{pts_c} \times 100 \tag{4.2}$$

The third one is similar to the previous, but is focused on the local variation, i.e. only the measurements affected (pts_x^l) are included in its calculation which is given by (4.3).

$$var_l = \frac{|pts_c^l - pts_a^l|}{pts_c^l} \times 100 \tag{4.3}$$

In both Tables, the ITER environment was the map chosen to perform the tests and, the first two tests serve as template for comparison. For the test number 1, the scan and map used were the same as used in the respective tests in Section 4.1.3. The second is a new scan but a repetition of the previous scenario, just to have a second correct scan reference. From these two scans results it can be seen that, on average, a global variation of about 0.7% exists between different scans for the same scenario. The third and sixth tests simulate the presence of the CPRHS in the environment. The third affected fewer

points than the sixth because it appears further away from the LRF. The remaining two test consisted in simulating opened doors. The fourth test presents a door more open producing more outliers than the fifth test.

Looking at the first approach results in Table 4.4, none of the algorithms were able to pass the tests, with a 10% map variation in some cases, completely destroying the results. Nevertheless, some performed better than others for variation values below 10% change. Despite extremes method being deprecated over vertex method, it have slightly outperformed vertex method showing the lowest average absolute error of the two. Perhaps would not be a bad idea to run the two in parallel. It is noteworthy that, the values of the absolute error were selected on the basis of the respective result lowest p-p error value. In some cases, better (or lower) absolute error values were achieved but with higher associated p-p error, and thus not selected. This indicates the p-p error should not be taken as a faithful evaluation measurement specially in the presence of outliers. This also contributes to the nonlinear variation of results with the map changing rate. On the other hand, excluding the last test, and despite showing the worst position results, voting method is the most consistent featuring a lower standard deviation value. The results prove the ICP algorithm is not robust to outliers in the measurements, and a change in about 1% of the map is enough to provoke a 10 cm deviation from the correct position. In tests 4 and 5, despite the error in scans being superior to test 3, the absolute error gave an inferior value. This happen because of the non-linearity of the measurements changes: in case the affected points result in a place closer to the map points, when the set is aligned, the p-p error will have a lower value, since it begins at the correct pose. This was the effect provoked by the tests where the doors have changed, and is most evident in the fourth test.

For the second approach, the same test methodology was applied except the last two variation measures are given in absolute values. Again the first two tests serves as reference for the following ones. Since neither the map nor the scan has changed, no variations were registered. From third to fifth tests, the columns objects within LRF sight were progressive changed from their original place in about 1 meter in a random direction. In the sixth test, the rightmost wall, of the already changed map, suffered a deviation. The map resolution and perimeter was kept constant throughout its change in order to maintain the consistency of point comparison.

Looking at the second approach results in Table 4.5, the results for the vertex and extremes method are exactly the same and provided the best overall results. Again voting method is the most consistent despite vertex and extremes method not being far behind. The ICP reveals itself less robust to map changes than for readings changes. It performed even worse for a minor change (1.1%) of the map description displaying the worst overall results of the map variation tests.

Table 4.4: Scanned outliers variation impact on pose 1 results for ITER map.

Nº	Variations	Error	ICP	Voting	Vertex	Extremes
1	0%, 0%, 0%	e_{pp} [mm]	50.8	265.4	59.2	59.2
		$e_{abs.}$ [mm, °]	1.2, 0.0	354.8, 0.0	38.5, 0.0	38.5, 0.0
2	0%, 0.7%, 0%	e_{pp} [mm]	48.6	271.0	61.7	61.7
		$e_{abs.}$ [mm, °]	4.3, 0.0	372.0, 0.0	84.3, 0.2	84.3, 0.2
3	1.7%, 1.1%, 24.5%	e_{pp} [mm]	318.4	391.6	343.3	329.8
		$e_{abs.}$ [mm, °]	94.6, 0.8	347.7, 0.0	153.6, 0.4	76.1, 0.6
4	2.5%, 2.7%, 79.3%	e_{pp} [mm]	166.8	294.9	172.8	172.8
		$e_{abs.}$ [mm, °]	48.4, 0.4	367.2, 0.0	53.5, 0.1	53.5, 0.1
5	5.5%, 1.1%, 9.6%	e_{pp} [mm]	109.3	275.1	199.2	146.4
		$e_{abs.}$ [mm, °]	48.8, 0.1	348.4, 0.0	197.0, 0.6	158.6, 0.5
6	10.5%, 18.8%, 173.5%	e_{pp} [mm]	1196.2	1350.5	1108.7	1108.7
		$e_{abs.}$ [mm, °]	1945.4, 4.5	324.7, 0.0	67545.2, 178.1	67545.2, 178.1
		$\mu(abs)$	534.3, 1.5	347.0, 0.0	16987.3, 44.8	13583.5, 44.8
		$\sigma(abs)$	941.0, 2.1	17.4, 0.0	33705.3, 88.9	30165.5, 88.9

Table 4.5: Map variations impact on pose 1 results for ITER map.

Nº	Variations	Error	ICP	Voting	Vertex	Extremes
1	0%, 0 mm, 0 m	e_{pp} [mm]	50.8	265.4	59.2	59.2
		$e_{abs.}$ [mm, °]	1.2, 0.0	354.8, 0.0	38.5, 0.0	38.5, 0.0
2	0%, 0 mm, 0 m	e_{pp} [mm]	50.8	265.4	59.2	59.2
		$e_{abs.}$ [mm, °]	1.2, 0.0	354.8, 0.0	38.5, 0.0	38.5, 0.0
3	1.1%, 15.9 mm, 1.4 m	e_{pp} [mm]	211.7	375.6	217.9	217.9
		$e_{abs.}$ [mm, °]	213.3, 0.6	354.8, 0.0	296.8, 0.8	296.8, 0.8
4	2.2%, 38.3 mm, 1.7 m	e_{pp} [mm]	410.1	501.5	0.430	429.6
		$e_{abs.}$ [mm, °]	451.2, 1.5	354.8, 0.0	193.3, 0.9	193.3, 0.9
5	3.5%, 50.4 mm, 1.5 m	e_{pp} [mm]	433.3	508.3	445.6	445.6
		$e_{abs.}$ [mm, °]	297.9, 1.3	354.8, 0.0	193.3, 0.9	193.3 0.9
6	15.1%, 166.4 mm, 1.1 m	e_{pp} [mm]	514.7	564.4	526	526
		$e_{abs.}$ [mm, °]	524.4, 1.2	261.1, 0.0	186.1, 0.1	186.1, 0.1
		$\mu(abs)$	371.7, 1.2	331.4, 0.0	217.4, 0.7	217.4, 0.7
		$\sigma(abs)$	141.6, 0.4	46.9, 0.0	53.1, 0.4	53.1, 0.4

Chapter 5

Conclusions

A network of LRF devices installed on site can be used in ITER as well as in industrial environments for localization and remote operation of vehicles. A robust localization method is necessary for not compromising vehicle maneuvers. In order to achieve accurate and meaningful results, the position and orientation of every sensor in the network needs to be exactly known in a given common reference frame. This thesis proposes an algorithm to calibrate a LRF sensor network. The outcome is an estimated pose for each device in the network. Simulated and experimental test results show that the pose estimate error depends on the accuracy of the sensor and map description.

The algorithm developed receives three input parameters: a map description of the environment, LRF data, and an initial pose estimate for each device of the network. The first two are mandatory and the uncertainty associated with the last one defines a scenario which determines the algorithm behavior. Four possible scenarios were considered: $\hat{P}_i$ completely known, only position known, only orientation known and $\hat{P}_i$ completely unknown. Regardless of the scenario, the map is always converted to points in order to assess the results using e_{pp} in (1.1). The map point conversion density should be chosen accordingly as the previous equation presents an $\mathcal{O}(n^2)$ operation.

Whenever an initial pose estimate is completely known, the ICP algorithm is executed. The closer the initial pose is to the correct pose, the more accurate the results are. The final pose estimate is calculated by applying the ICP resultant rigid body transformation to the initial pose estimate. This is the fastest and most accurate method, but it is not robust against the presence of outliers in the LRF data or map description.

As for the intermediate situation, only the position or orientation initial estimate value is known. In the first case, a brute force ICP is carried out in the given position for a set of predefined angles. In the second case, a voting based algorithm is applied to assign an estimate probability of the LRF position to each of the map coordinates. It can be time consuming and the least accurate method, but nevertheless it proposes a consistent valid alternative and a case of study for a global search method.

An extreme situation may occur when the $\hat{P}_i$ of the sensors is completely unknown. In this case, ICP alone cannot be used because of the problem of local minima. Therefore, a developed vertex method is applied. This method comprises two phases: first, geometric features, as segment lines and vertices,

are extracted and subsequently, a matching technique based on the geometric combination of these features is applied. The main problem here is to figure out the best combination of threshold values that lead to a correct features extraction. The time complexity is proportional to the number of extracted features and, when using an inaccurate map description, it can outperform ICP. As an alternative to this method, the ICP brute force experiment was conducted where the map was divided into a grid and for each point of the grid, the ICP was executed. Although much more time consuming than the vertex method, the results obtained were slightly more accurate.

5.1 Issues/weaknesses of the methods developed

The classical version of the ICP algorithm is the chief method for the final alignment. Despite its local minima problem, excellent results were achieved. In the presence of outliers, the algorithm tends to lose its accuracy due to the outbreak of multiple local minima near the correct solution. This could prevent the algorithm from achieving accurate results no matter how accurate the $\hat{P}_i$ is. ICP is an algorithm with many parameters that can be highly explored and customizable to incorporate additional available information for a better alignment process. To improve the accuracy results, an outliers rejection method, like the Smirnov-Grubbs test [42], should be implemented. A simple one would be to reject pairs of matched points that are more distant than a given threshold and try to make the threshold dimensionless by relating it to other pair results. It is also possible to associate a weight value to a corresponding pair of points and that could be based on the uncertainty associated to device accuracy, or between vertex points. It is also possible to use a two dimension quadratic loss function as an error metric that takes into account, apart from the geometric distance, information about the measurements reflectivity. If applicable, and in case the map is accurate enough, a wall attraction constraint could also benefit the accuracy of the results.

The voting algorithm is the method that showed the overall worst results. Observing the tables with the results of the experiments, one may draw the conclusion that this method is very sensible to initial angular deviations, since it takes the angle estimate as the truth orientation and it does not have any mechanism to correct it during its execution. As a consequence, another problem arises when ranking the results. It is sensible to choose the most voted result as suggested by the algorithm design and objective. When the angular deviation from the initial pose to the truth pose is large, mainly superior to 5°, the deterioration of the results is bigger. The correct bin loses votes for other bins, nevertheless, the supposedly correct result can be found near the top in the ranking, as it has been stated in Section 4.1.3. Choosing the highest voting, in this case, would most likely lead to an erroneous result, as one can see on Figure 5.1. In order for the method to withstand large orientation deviations, the ranking method should be changed. Another factor that could have affected the results is the resolution value of the map conversion to points. Ranking results for pose 1 of the simulated ITER map featuring a value of 1 point per meter resolution revealed a supposedly correct 10^{th} place, which corresponds to the red spot at the bottom of the respective image in Figure 4.6. This is the expected result because of readings deviation caused by a large initial orientation value. A repetition of these tests was conducted

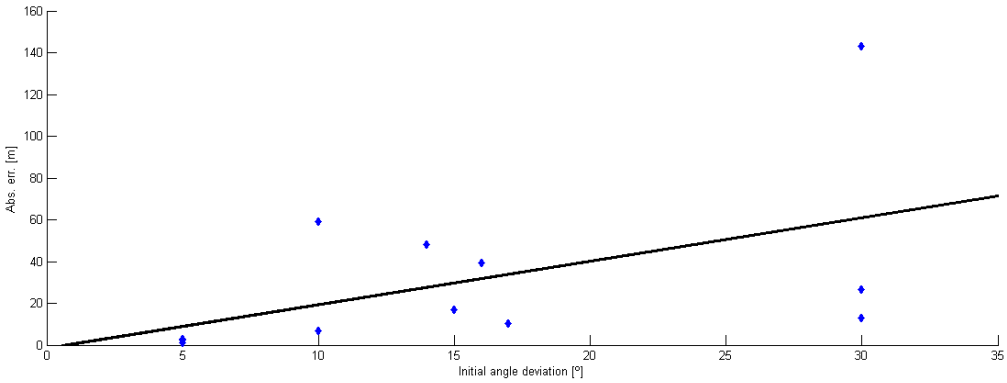


Figure 5.1: Absolute error results for voting method varying initial angle deviations.

Table 5.1: Map resolution impact on voting method results.

Res.	1st place		2nd place		3rd place		RT [h]
	votes	$e_{abs.}$ [m]	votes	$e_{abs.}$ [m]	votes	$e_{abs.}$ [m]	
1	173	59.32	172	59.10	168	59.21	0.1
2	277	59.10	276	58.99	274	6.49	0.2
3	387	6.49	363	59.10	356	58.99	0.4
5	627	6.49	549	58.99	545	59.10	1.1
10	1244	6.49	1033	58.90	1027	59.10	4.1

for the same scenario maintaining the same variables but with a different map resolution. The results are presented in Table 5.1 which shows the expected result (6.49 m) moving closer to the first place, starting from a resolution value of 3 points per meter. Even after 2 points per meter, it ascends to the top three. The same behavior was observed in the balcony map, pose 2, of SICK readings, where a test of 20 points per meter of resolution revealed a p-p error of 113.4 mm which is closer to the ICP result. A test of high resolution value (200 points per meter) was also tried for this map, but the slight improvement in the results did not compensate the execution time. Also, for the IPFN pose 1 test, a change from 5 to 20 point per meter led to a variation of the p-p error from 154.8 to 43.6 mm. Taking this behavior into account, choosing a value closer to the scan points density seems to lead to better results without compromising execution time. There are several steps that can improve the accuracy of this method, namely: using the p-p error evaluation together with vote counting, different bin resolution and/or stages to decrease the granularity level of the possible location to vote for, varying the map point resolution or eliminating the improbable poses hypotheses (out of reach map sections).

The vertex/extremes method developed is used to overcome the main ICP constraints: the availability of an initial pose estimate and global matching. The geometric feature extraction process is its main weakness. The Split and Merge method is based on thresholds and tolerances whose values should be chosen taking into account many factors such as the size and shape of the map, number of points of the scan dataset, as well as LRF maximum range, resolution and accuracy. Disperse first points from a scan sequence can also have a negative impact on the clustering algorithm, since it uses the points sequentiality, it can result in wrong extracted lines and/or vertices as shown in Figure 4.18. Small

lines are a problem to extract with the correct slope, as they are often composed by fewer points. As a consequence, it affects the correct vertex extraction. In the presence of outliers, the main problem is the error evaluation, as lines are often correctly extracted. In this case, the p-p error may not be the best score indicator. Consequentially, results with lower absolute error, were ignored due to higher p-p error association. And this also explains why ICP loses its accuracy.

Two other algorithms were devised as a direct alternative to the vertex method to tackle its main weaknesses. One was a line tracking algorithm that takes into account the LRF accuracy on cluster formation. The other one, using Hough transform, consisted in a global matching in the Hough domain that would be more efficient. The Hough line representation could reduce the search complexity.

5.2 Achievements

It was possible to verify, throughout the simulation, that, in the best case scenario, the error associated to the pose of a LRF has a level of uncertainty of the same order of magnitude of the device accuracy. The simulation results of a LRF featuring a 1 mm accuracy in the ITER environment, shown an error as low as 3 mm, for its position, which is enough not to compromise the correct functioning of the localization algorithm. The developed algorithm is able to reach such accurate results without even knowing an initial pose estimate. Although the constraint of LRF being installed in ITER building walls, the results are not restricted to walls. The ICP algorithm global matching problem was tackled using a developed vertex method that only needs a vertex and one associated line to achieve reasonable results.

5.3 Future Work

In future developments, the ICP algorithm should be improved to a more refined and appropriate state in order to be used in the described scenarios. Furthermore, new methods should be created or the existing ones should be improved, so that they become more robust and not so much reliant on thresholds. Also, the research and development for the 3D should be expanded, so that new variables are introduced, increasing the problem complexity (some of them are identified in Section 3.2 such as the three Euler angles and the z coordinate). Finally, the situation where no map is available that was not addressed, even though it was in the initial objectives. Regarding ITER operation, it is possible to develop a monitoring system to periodically check for malfunctioning and issues with the devices, or environment changes, by observation of variations in the resultant pose estimations over time.

In a commercial perspective, it would be interesting to develop a real time calibration kit (hardware and software) where the installment and calibration adjustments of devices, in the predefined poses, could be made on the fly by an operator with the aid of voice commands. The hardware would be an adapter to connect to a wide range of different sensor devices and able to communicate via Wi-Fi to a mobile computer device using dedicated software which would include the required sensor model libraries.

Bibliography

[1] A. Tesini, J. Palmer. The ITER remote maintenance system. *Fusion Eng. Des.*, 83(7-9):810–816, 2008.

[2] Darren Locke, Carmen Gutiérrez. Progress in the conceptual design of the ITER cask and plug remote handling system. *Fusion Eng. Des.*, 89(9-10):2419–2424, 2014.

[3] J. Soares, R. Ventura, A. Vale. Human-robot interface architecture for a multi-purpose rescue vehicle for remote assistance. In *Symposium on Fusion Technology*, 2014.

[4] J. Ferreira, A. Vale, R. Ventura. Vehicle localization system using off-board range sensor network. *Proc. of the 8th IFAC Intelligent Autonomous Vehicles Symposium*, 8(1):102–107, June 2013.

[5] Matthew Antone, Yuli Friedman. Fully automated laser range calibration. In *Proc. BMVC*, pages 66.1–66.10, 2007. ISBN 1-901725-34-0. doi:10.5244/C.21.66.

[6] EVISA. Error sources in speciation analysis - overview, June 2013. URL `http://www.speciation.net/Public/Document/2013/06/13/6724.html`.

[7] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[8] V. Caglioti, A. Giusti, and D. Migliore. Mutual calibration of a camera and a laser rangefinder. *International Conference on Computer Vision Theory and Applications*, 2008.

[9] Vincenzo Caglioti, Alessandro Giusti, Davide Migliore. Mutual calibration of a camera and a laser rangefinder. *Proc. Int'l Conf. on Computer Vision Theory and Applications*, pages 33–42, 2008.

[10] Fischler M., Bolles R. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 1981.

[11] Konrad Schenk, Alexander Kolarow, Markus Eisenbach, Klaus Debes, and Horst-Michael Gross. Automatic calibration of a stationary network of laser range finders by matching movement trajectories. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 431–437, 2012.

[12] P. Besl and N. McKay. A method for registration two 3-d shapes. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 14(2):239–256, 1992.

[13] Sebastian Thrun, John J. Leonard. *Handbook of Robotics*. Springer, 1^{st} edition, 2008. ISBN:978-3-540-30301-5.

[14] Viet Nguyen, Stefan Gächter, Agostino Martinelli, Nicola Tomatis, Roland Siegwart. A comparison of line extraction algorithms using 2d range data for indoor mobile robotics. *Autonomous Robots*, 23(2):97–111, 2007.

[15] Markus-Christian Amann, Thierry Bosch, Marc Lescure, Risto Myllyla, Marc Rioux. Laser ranging: a critical review of usual techniques for distance measurement. *Opt. Eng.*, 40(1):10–19, 2000.

[16] Nornadiah Mohd Razali, Yap Bee Wah. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of Statistical Modeling and Analytics*, 2:21–33, 2011.

[17] Jing Wu, Hong Zhang. Camera sensor model for visual slam. *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*, pages 149–156, 2007.

[18] Cang Ye and Johann Borenstein. Characterization of a 2-d laser scanner for mobile robot obstacle negotiation. *IEEE International Conference on Robotics and Automations*, 2002.

[19] Martin Dekan, František Duchoň, Ladislav Jurišica, Anton Vitko. Probabilistic model of laser rangefinder. *Ad Alta : Journal of Interdisciplinary Research*, 01(02):151, 2011.

[20] K. S. Arun, T. S. Huang, S. D. Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 9(5):698–700, 1987.

[21] M. Alshawa. Icl: Iterative closest line: A novel point cloud registration algorithm based on linear features. *IEEE Trans. Pattern Anal. and Mach. Intell.*, (1):53–59, 2007.

[22] Kok-Lim Low. Linear least-squares optimization for point-to-plane ICP surface registration. 2004. Tech. report.

[23] A. Censi. An ICP variant using a point-to-line metric. *Robotics and Automation*, pages 19–25, 2008.

[24] Michael Greenspan, Mike Yurick. Approximate k-d tree search for efficient icp. *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, pages 442–448, 2003.

[25] Carlos Fernández, Vidal Moreno, Belen Curto. Clustering and line detection in laser range measurements. *Robotics and Autonomous Sys.*, 58(5):720–726, 2010.

[26] David Forsyth, Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 1^{st} edition, 2003. ISBN:0131911937, 9780131911932.

[27] P. Rousseeuw, V. Yohai. Robust regression by means of s-estimators. *Robust and Nonlinear Time Series Analysis*, 26:256–272, 1984.

[28] Muthukrishnan R, Radha Myilsamy. M-estimators in regression models. *Journal of Mathematics Research*, 2(4):23, 2010.

[29] Geovany Araujo Borges, Marie-José Aldon. Line extraction in 2d range images for mobile robotics. *Journal of Intelligent and Robotic Systems*, 40(3):267–297, 2004.

[30] Pierre H. Flamant. *Atmospheric and meteorological Lidar: from pioneers to space applications*, volume 6. 2005.

[31] *SICK LMS500 Operating Instructions*.

[32] Zaereth. A 5 mw green laser pointer beam profile, showing the tem00 profile, 2012. URL `http://commons.wikimedia.org/wiki/File:Green_laser_pointer_TEM00_profile.JPG`.

[33] Global Laser Ltd. To-can n-type laser diode, 2015. URL `http://www.global-lasertech.co.uk/`.

[34] DrBob. Gaussianbeamwaist, 2009. URL `http://commons.wikimedia.org/wiki/File:GaussianBeamWaist.svg`.

[35] NCS Pearson. Laser diode, 2015. URL `http://physics.tutorvista.com/electricity-and-magnetism/laser-diode.html`.

[36] *RIEGEL Application note*, an-gi002 edition.

[37] Newport Corporation. Laws of radiation, 2015. URL `http://www.newport.com/Laws-of-Radiation/381843/1033/content.aspx`.

[38] Sahapong Kruapech, Joewono Widjaja. Laser range finder using gaussian beam range equation. *Optics and Laser Technology*, 2009.

[39] Inductiveload. Normal distribution pdf, 2008. URL `http://commons.wikimedia.org/wiki/File:Normal_Distribution_PDF.svg`.

[40] *Hokuyo Scanning Laser Range Finder URG-04LX-UG01, Specifications manual*, 2009.

[41] *SICK LMS200 Technical Description manual*, 2004.

[42] Yoshiaki Takeda, Norifumi Aoyama, Takahiro Tanaami, Syoto Mizumi, Hiroyuki Kamata. Study on the indoor slam using kinect. *Proceedings in Information and Communications Technology*, 4:217–225, 2012.

Appendix A

Results Figures

A.1 Voting accumulation

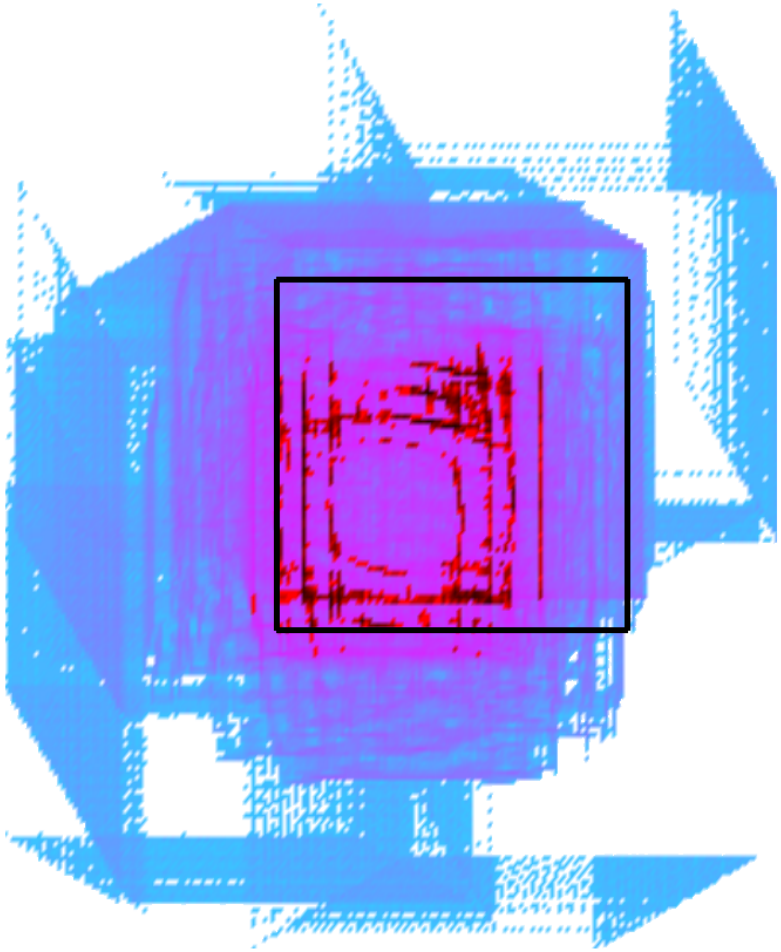


Figure A.1: Voting accumulation for all the four poses for the symmetrical map.

Six voting accumulations image results are shown in this section, each one for a different map used in the test phase. Instead of showing four images of voting results, one for each pose, each accumulation image aggregates the results by superposition the vote count from the four poses tested, for a given

map. Then, the vote count is normalized to apply for the same scale and color map of the respective voting images presented in Chapter 4.

Figure A.1 shows the voting accumulation for the symmetrical map. Only one of the darkest spots corresponds to a correct pose, and that is the pose 2 which was the least ambiguous of the four poses. Results are not symmetrical distributed, they tend to the bottom left of the map because two of the four scans (pose 2 and 3) were taken from there.

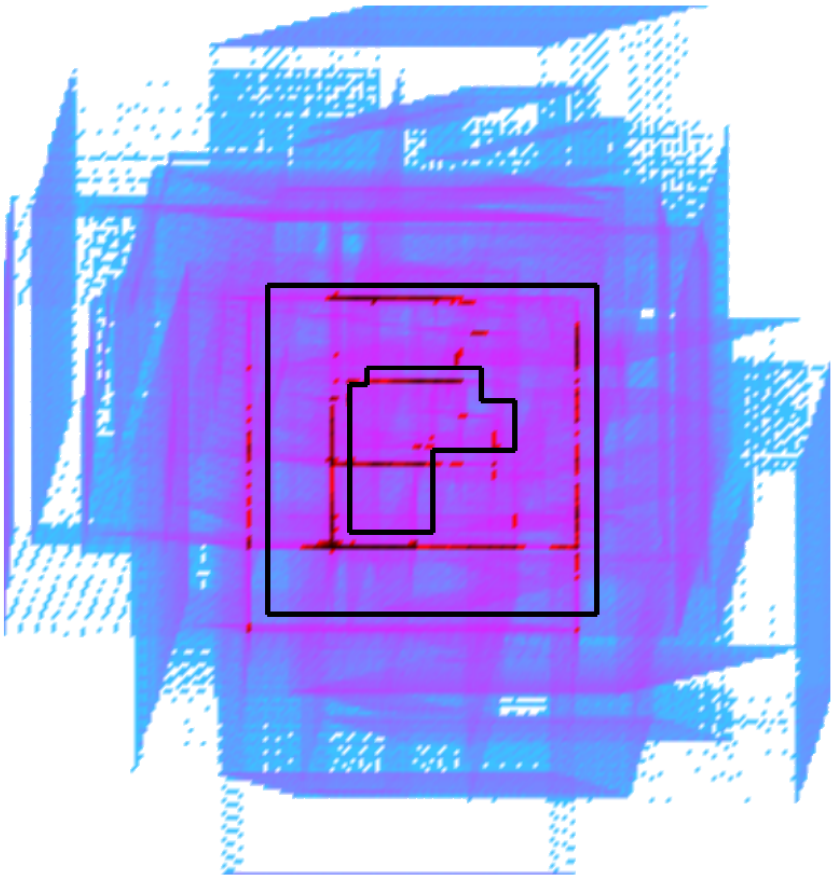


Figure A.2: Voting accumulation for all the four poses for the asymmetrical map.

Figure A.2 shows the voting accumulation for the asymmetrical map. All the four poses, except pose 1, show a dark spot in their corresponding location. The local minima, discovered by ICP results, that exists right above pose 1, seems to accumulate more votes than the location of pose 1 because of the bottom wall contribution of pose 4 projections. In addition, target walls from different poses are at the same distance when projected from different walls.

Figure A.3 shows the voting accumulation for the ITER map. In this case, all four poses correspondent locations show a high vote count, although the number of votes of some other locations surpass the correct ones. One example is the big dark spot in the superior right section of the image which is discussed in detail in Section 4.1.3.

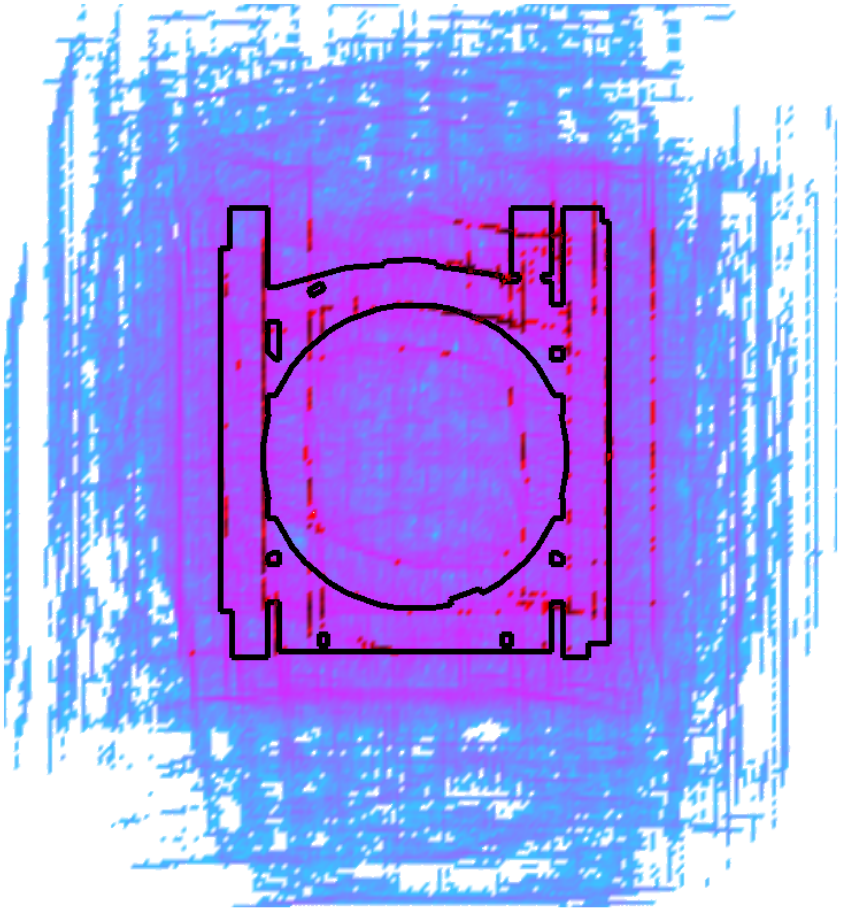


Figure A.3: Voting accumulation for all the four poses for the ITER map.

Figure A.4 shows the voting accumulation for the balcony map using the SICK results. Image seems a bit discretized, this could be caused by the lower value of map resolution point conversion and lower FOV of the SICK device. Nevertheless, the correct locations of the poses are marked with a dark spot, among other locations as well. For instance, the right long wall is projected in the center of the map not only because of pose 1 scan influence, but from the others as well since it is always picked up in the other scans.

Figure A.5 shows the voting accumulation for the IPFN map. The image shows a darker horizontal line across the map followed by the two lines on the side walls. The pose locations do not show prominent results because of symmetric problem, and long horizontal walls, covered in Section 4.2.

Figure A.6 shows the voting accumulation for the garage map. This image shows, as happened with previous ones, that the large walls, which are picked up on most of the different scans, have a high impact on the vote counting. The darker spots show patterns of these walls, specially the bottom, horizontal and vertical ones that compose the map limits. In all the poses locations an accentuated dark spot can be observed among few incorrect other ones.

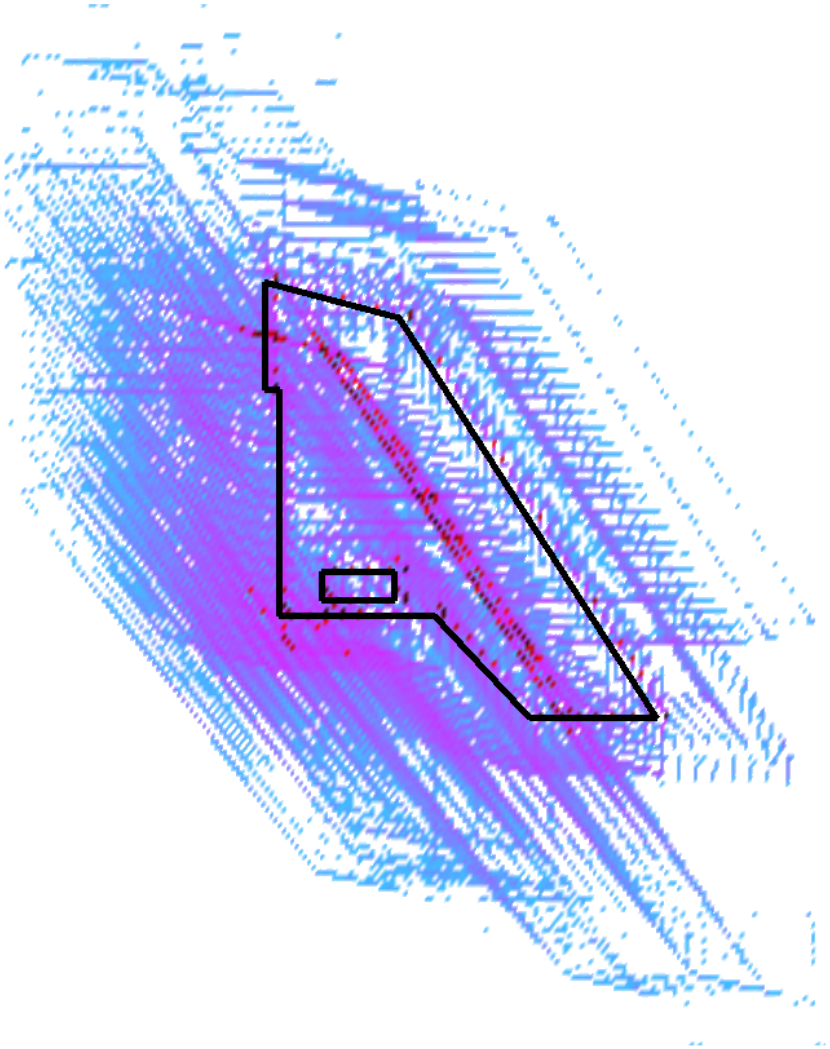


Figure A.4: Voting accumulation for all the four poses for the balcony map.

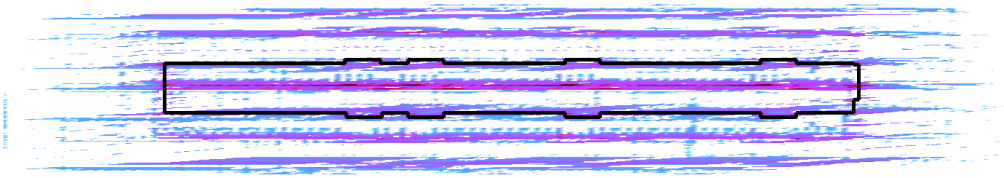


Figure A.5: Voting accumulation for all the four poses for the IPFN map.

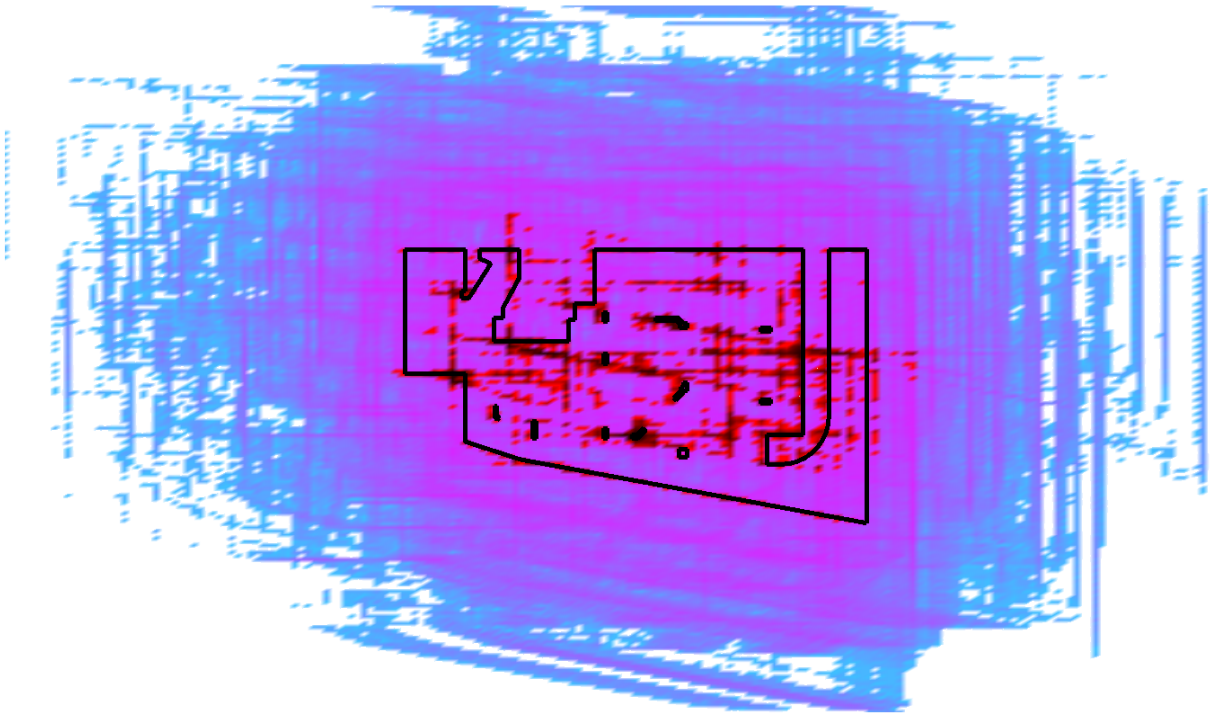


Figure A.6: Voting accumulation for all the eight poses for the garage map.

A.2 ICP error evolution and others

The stop condition criteria of the ICP algorithm can affect the results. In Figures A.7 and A.8 the error evolution along iterations of the ICP algorithm is shown for the same situation (same initial pose, correct pose, map, scan data, etc.). They only differ in the stop criteria. In both, the upper left image shows the absolute position error, in the upper right the absolute orientation error and in the bottom the root mean square (RMS) p-p error is displayed. For the first, the stop criteria was a threshold value: when the RMS error difference between consecutive iterations were below that value, the algorithm stopped. In the second, the stop condition was the one described in Section 2.2.1 and actually implemented. It consists in stopping when points pairing is the same between consecutive iterations. From the figures observation, it can be seen that, the different stop criteria led to different absolute error values, the first being the incorrect one. Although the low error difference between consecutive early iterations (around 30), the algorithm still had not completed its execution. That is an example of how dangerous it can be to choose the first stop criteria, and that was the main reason why the second criteria was implemented in the used ICP algorithm.

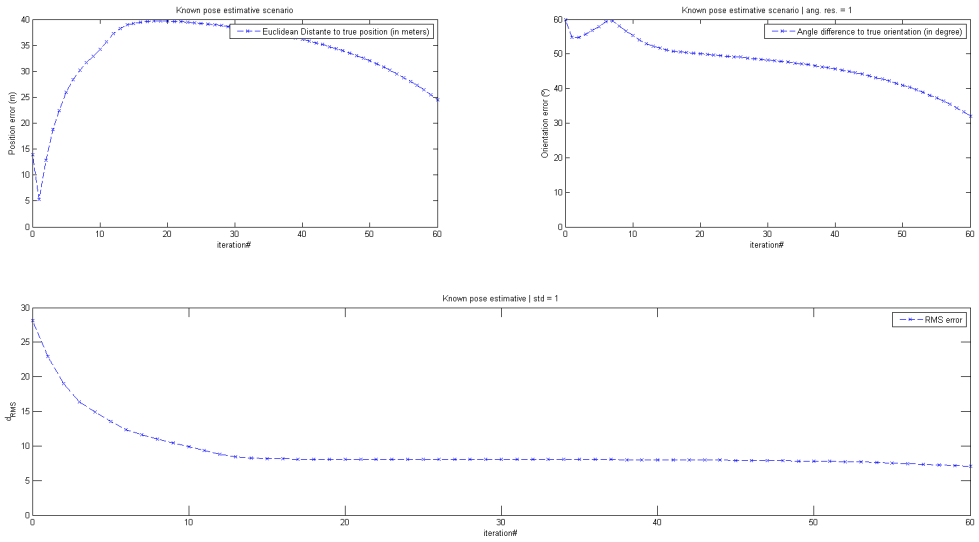


Figure A.7: ICP error evolution for a relative low number of iterations.

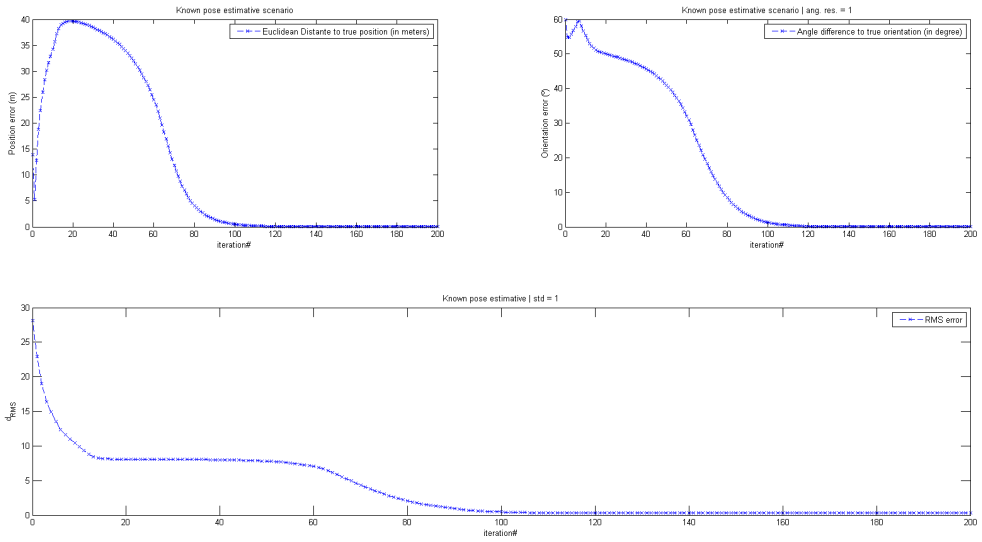


Figure A.8: ICP error evolution for a relative high number of iterations.

Figure A.9 is an example of a correct line extraction in the presence of outliers that happened during the experimental tests. The scan was taken from the upper section of the map and, somehow, the most distant scanned points began to show high deviations from the target wall, composing the outliers. Nevertheless, the longest wall was correctly extracted, due to the robust fit technique that was used in the features extraction phase. The vertex method was then applied resulting in the alignment that can be seen in the respective Figure.

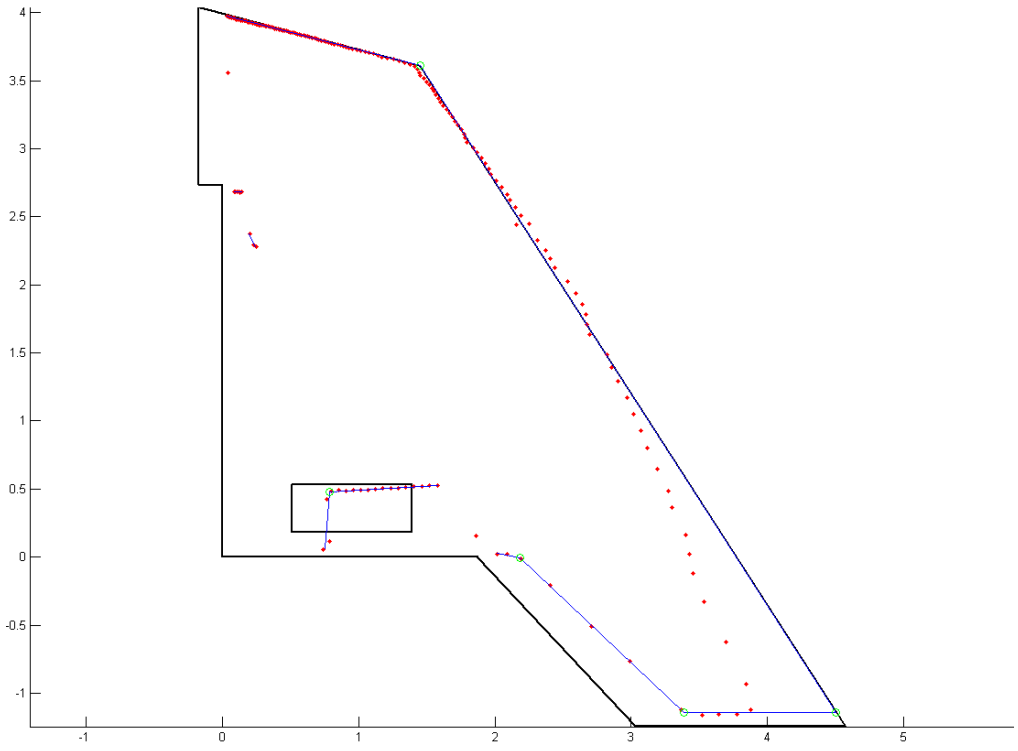


Figure A.9: Pose 2 line extraction example for SICK data.

Appendix B

Notation

B.1 Notation table

The following Table presents the notation and respective short description of the variables used in Chapter 3.

Table B.1: Notation of used variables.

Notation	Description
ζ_a	Angular noise deviation
ζ_l	Linear noise deviation
r	LRF ray line
r_i	LRF ray line without noise
r_n	LRF ray line with noise
p_n	Point (x, y) of intersection of LRF ray with obstacle with linear and angular noise applied
p_a	Point (x, y) of intersection of LRF ray with obstacle with only angular noise applied
α_i	Angle without noise
α_n	Angle with noise
$\vec{\hat{v}}_n$	Normalized vector of linear noise
P	Pose (position and orientation)
Q	LRF position
$p_w = (x_w, y_w)$	2D point coordinates in world frame
$p_i = (x_i, y_i)$	2D point coordinates in LRF frame
p_m	Map point
O	Orientation angle in world frame
Out^s	Output of Simulator
Out^r	Output of Simulator for testing phase
D	Total number of sensor devices in the network
l	Map limits
R_{max}	Maximum range
θ_{ini}	Inital angle of FOV
θ_{fin}	Inital angle of FOV
θ_{max}	Maximum FOV angle
θ	Incidence angle
Dir	Scan direction
s_w	Scan velocity
s_r	Scan rate
res_a	Angular resolution
res_l	Linear resolution
t	Number of LRF rays per scan or angular steps
rl	Reflectivity value
rl^c	Reflectivity value corrected
M	Map description
ls	Line segment
w	Wall line segment
w_t	Number of walls
g	Point of intersection between LRF ray lines and walls
d	Distance to obstacle measured by LRF