UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXIX Ciclo

Robust Feature-based LIDAR Localization and Mapping in Unstructured Environments

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Stefano Caselli

Dottorando: *Fabjan Kallasi*

Dicembre 2016

Om Namah Shivaya

Contents

Introduction

Robot localization is a fundamental task and one of the most studied topics in mobile robotics. This capability is fundamental to fully provide autonomous functions to mobile robots. In order to localize itself, a robot uses information provided by one or several on-board sensors. Localization data are then used by higher level algorithms to understand situations and make decisions. Several approaches have been presented in the literature, each covering a specific combination of working environment, installed sensors and requested performance. The state of the art of robot self-localization, place recognition and environmental mapping includes techniques based on data fusion of heterogeneous as well as single sensors. Even though the data acquired by sensors are often affected by uncertainty or errors, localization algorithms must provide accuracy and robustness. These capabilities can be obtained in several ways:

- *Improved perception*: adopting sensors with better technical specification and improving extraction of relevant information from raw data,
- *Data fusion*: merging several perception data given by different sensors to overcome individual shortcomings or flaws,
- *Modeled uncertainty*: representing the problems with a characterization of each source of uncertainty, so as to compute the solution in an optimum way,
- *Data tracking*: matching data over time to avoid spurious detection and unstable results,

- *Place recognition*: using the past and the known data to refine the estimation,
- *Structured environment*: arranging the environment with physical structures or markers to facilitate sensor data extraction and robot localization.

Since the beginning of mobile robot localization studies, several kind of sensors have been tested and reviewed. Sensors can be roughly divided in two main classes: proprioceptive and exteroceptive, where the first class represents sensors which measure the internal robot status, and the latter one includes sensors which measure external features of the working environment. In robot localization, both sensor classes are used simultaneously. Proprioceptive sensors are usually used to give an initial guess of the robot position by incremental integration of derivative measures (wheel speed, acceleration, displacement, etc.). Due to the integrative nature of the data, this type of perception is often affected by drift. The drift can be corrected using environmental measurements given by exteroceptive sensors (sonars, lasers, gps, optical flows) and the robot localization refined. Sensors can be also selected by accuracy, range and type of measurements. The mixed-class sensor configuration is commonly used in autonomous ground vehicle (AGV) operating in environment like warehouses, museums and roads.

Although their cost is often higher than other sensors, laser scanners provide a reliable measure of relative distance with obstacles. A laser scanner can be used in almost any environment (industrial, aerial, underwater) and it is affected only to limited extent by external or natural conditions. While in literature the most studied and applied methods of localization with planar laser scanner are grid maps[1, 2, 3, 4], in the last few years the new feature-based approach is gaining interest by the scientific community. In computer vision and point cloud processing, the feature approach [5, 6, 7] has become an established paradigm to address mobile robot localization and mapping problems.

This thesis deals with the problem of mobile robot localization and mapping in human-made environments. The contribution of this work is an innovative method for simultaneous localization and mapping (SLAM) based on robot odometry and a two dimensional laser scanner. The presented method does not requires an initial

configuration of the environment and therefore can be adopted wherever the robot can operate.

Keypoint Features for LIDAR

The goal of the feature approach is the detection of interest points, also known as *keypoints*, from the sensor data and the computation of a distinctive signature for each of them. The term keypoint feature or, briefly, feature refers to an interest point with the corresponding descriptor. These descriptors have the form of a vector of values which are usually dependent from the interest point neighborhood. The main requirements of keypoint detection are invariance from viewpoints, stability and repeatability. Descriptors facilitate keypoint identification and matching between two different scans. Several approaches for keypoint matching have been proposed. Most of them are based on geometric matching, although descriptors enable more robust correspondence matching among points acquired from different viewpoints. Keypoint features have been originally proposed for images and point clouds, whereas only few keypoint features exist for sensor data acquired from planar range finders [8, 9, 10]. While images and point clouds provide richer information, laser scans consist of just relatively few two-dimensional points and can express only geometrical representations of the working environment. The peculiar nature of laser measurements makes the extraction of stable and distinguishable interest points more difficult.

In this thesis, I propose a novel keypoint detector, named FALKO (*Fast Adaptive Laser Keypoint Orientation-invariant*) [11], and two novel descriptors, *Binary Shape Context* (BSC) and *Cumulative Gaussian Histogram* (CGH). The proposed detector extracts high curvature points in laser scans. However, FALKO has been designed to detect more stable points like corners rather than general elements like gaps and isolated points that are dependent on viewpoint. FALKO selects interest points through a careful neighborhood selection and an effective cornerness scoring. *Binary Shape Context* (BSC) is a variant of the well known shape context [8] that is less sensitive to point concentration (Figure 1). The *Cumulative Gaussian Histogram* (CGH) counts the polar distribution of neighbor points w.r.t. the main orientation of the interest points. FALKO has been compared with the state-of-the-art approaches [8, 11] on

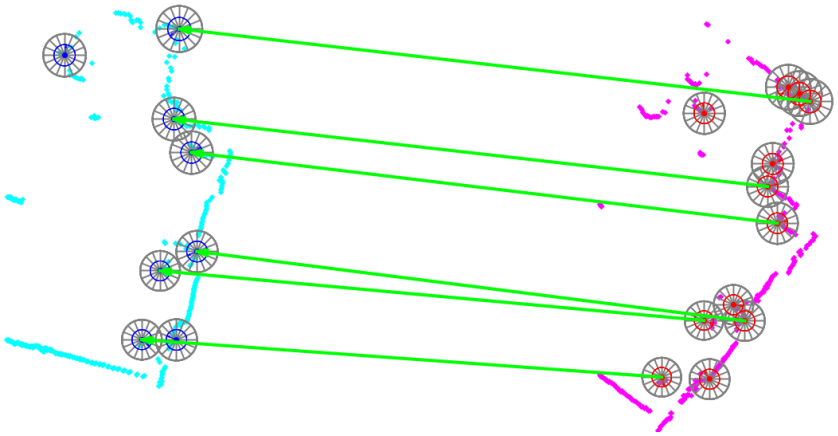


Figure 1: Example of feature association with FALKO detector and BSC descriptors.

standard benchmark datasets used in the robotic community to assess feature stability and localization performance.

Place Recognition with LIDAR Keypoint Features

Feature-based approaches allow a lightweight representation of the environment, which is indeed more manageable than the well known grid map. The recognition of distinctive regions and places from sensor measurements is a main capability in robot localization and mapping problems. When the robot comes back to an already visited region after traveling for a long path, the system needs to be able to recognize similarities in the sensor measurements. This operation is commonly named *loop closure* and is crucial to globally localize the robot and to achieve consistent maps. After loop closure, the computation of the relative transformation between the current robot frame and the matching one is an important requirement. Although this topic has been extensively investigated, place recognition with range finder is comparatively less developed than in the case of computer vision or other exteroceptive sensor data. Planar laser scans provide limited information beyond metric and occupancy data and, therefore, loop closure has been mainly addressed by matching occupancy grid maps [1, 2, 3, 4]. If scans are represented as collections of points, then the problem of

detecting loops in robot trajectory is reformulated as a comparison between a query point set and all the stored point sets of the map. Since point-to-point association is a computationally complex procedure, the exhaustive matching of point sets has been substituted by the efficient comparison of scan signatures. A signature is a vector obtained from the keypoints and detectors extracted from a laser scan that can be used to select candidate loops. A typical loop closure procedure can be divided in two steps: the selection of candidate loops using signatures and the point-to-point association to estimate the transformation relating the scans.

In this thesis, I present novel loop closure methods based on FALKO keypoints (Fig. 2) and a fully rotation-invariant geometric-based signature (GLAROT) [13], and compare their performance in online localization and mapping problems. The methods based on geometric constraint signatures and point-to-point correspondences better perform with stable and viewpoint-invariant keypoints like FALKO. The loop closure procedures considered in my experimental assessment are obtained by combining a keypoint detector, a signature, and a point-to-point association algorithm. The scan signatures include the state-of-the-art Geometrical FLIRT Phrase (GFP) [14], Geometrical Surface Relations (GSR) [15], and GLAROT, a novel version of GLARE (Geometric LAndmark RElations) [16] fully invariant to rigid transformation. The tested point-to-point association algorithms are RANSAC, Correspondence Graph approach and Affine Hough Transform. Experiments illustrate the performance of discussed methods in the context of both offline and online loop closure, i.e. both disregarding and considering causality and time discontinuity constraints in scan association. Moreover, the loop closure methods are applied to online map estimation.

Applications of Landmark-based LIDAR Localization

Although the presented methods address the problem of robot localization and mapping, localization is performed w.r.t. the sensor frame instead of the robot logical point. Indeed, robot pose computation is possible knowing the exact position of each sensor frames w.r.t. the logical point. The parameters relating the sensor measurements to the robot reference frame are called *extrinsic parameters*. Further, knowledge of the exact parameters describing the relationship between the actuator state

and the robot kinematics and used by proprioceptive sensors to measure the internal state of the robot, enables a more stable and precise estimation of position initial guess. These parameters are called *intrinsic parameters*. Formally, the problem of estimating the robot sensor configuration, including both extrinsic and intrinsic parameters, is called calibration. A standard approach for mobile robot calibration is to compare the expected motion of the robot from some input commands with the observed trajectory. Usually, robots are calibrated through a manual procedure requiring the intervention of an operator to measure the effective trajectory or simply using physical measurement and technical specification. The calibration accuracy largely depends on the skills and experience of the operator performing the calibration or on the accuracy of technical design and information. Moreover, consistency of manual calibration with multiple, possibly different robots operating in the same environment is often questionable and can lead to robots exhibiting slightly different behavior for the same nominal position. Research has addressed several formulations of calibration problems for different robotic systems, including the calibration of multi-sensor systems and robot odometry [17, 18, 19, 20, 21]. Many works focus on the estimation of either intrinsic [22, 23, 24, 25, 26, 21] or extrinsic parameters [27, 28, 19, 29, 30, 31]. The kinematic model considered in works addressing mobile robot calibration is often the popular differential drive model, although in real world scenarios other kinematic configuration are preferred by constructors. E.g., in industrial environments, due to the simplicity of a self-standing configuration as well as its suitability for fetching, carrying, and depositing heavy loads, a *tricycle structure* is often preferred. Tricycles consist of a steering actuated front wheel and two passive back wheels. In this thesis, I present a novel autonomous calibration method for tricycle wheeled robots along with the formal analysis of its correctness, and its application to the calibration of industrial AGVs [32]. Indeed, no approaches have addressed so far automatic calibration of both intrinsic and extrinsic parameters of tricycle robots equipped with a sensor. The proposed algorithm allows simultaneous estimation of both intrinsic and extrinsic parameters of a tricycle robot using the data collected while moving on predefined paths.

In this thesis, I also present an application in a real industrial scenario. The pro-

posed fully automated calibration system has been implemented and tested with industrial LGVs (Laser Guided Vehicles) thanks to Elettric80 S.p.a.[1] support. Furthermore, the feature-based localization method has been tested and compared to an existing localization system based on structured environment.

This thesis is organized as follows. Chapter 1 describes further the problem addressed by this work together with the state of art, with a particular focus on laser based localization, feature extraction and sensor robot calibration. A description of the fully autonomous LGV calibration algorithm is given in chapter 2. Chapter 3 presents the development and results of the proposed keypoint and descriptors extractors. The evaluation of the developed feature based SLAM method is discussed in Chapter 4. Chapter 5 reports the implementation, comparison and results of the method proposed in this thesis in a real industrial scenario. Finally, some conclusions and recommendations for future works are proposed in the final chapter.

[1] http://www.elettric80.com

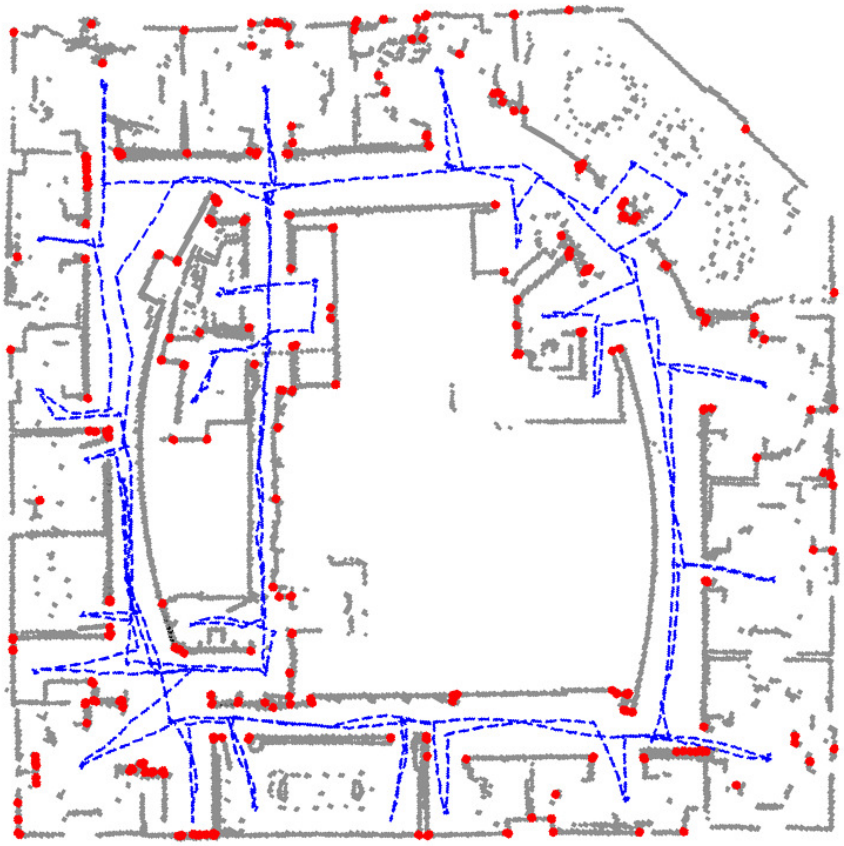


Figure 2: Example of FALKO keypoints (red points) extracted in *intel-lab* dataset [12].

Chapter 1

Relevant Prior Art

Simultaneous localization of a robot and map building (SLAM) is an important capability for robot autonomy in several applications. In literature, several solutions to this problem have been proposed to enable robot navigation and operation in unknown environments. Proposed methods exploit a large variety of ideas starting from the simplest *scan-to-scan* registration to the more complete graph SLAM approach. Range finders are widely used in the robotic community due to their reliability and accuracy of measurements. Since the feature-based approach is a well established paradigm in computer vision robot localization, recent works exploit the potential of LIDAR feature based localization methods. In applications where the robot must simultaneously localize and estimate the map, it often moves on a path without visiting previously explored regions. Accurate odometry measurements and sensor calibration compensate the localization errors introduced by drift and noise. In this chapter, I will overview the most used SLAM algorithms and LIDAR feature extraction methods. I will also describe the state of the art in laser scan signatures. Finally, I will summarize the best known techniques for AGV Calibration with differential drive kinematics.

1.1 Laser based scan matching, localization and mapping

Even if multiple solutions to localization problems have been proposed, several issues are still open. In particular, localization in dynamic environments and with incomplete maps is still an active research area. Range finder scans provide a good approximation of the environment shape in the form of two dimensional points. Given an accurate localization, an iterative accumulation of those points can result in high quality maps representing the obstacles in the environment. These maps can be used by the robot to navigate, avoid obstacles and re-localize itself. In this section, I will present some of the more representative localization methods that have been discussed in literature.

1.1.1 Registration

Given two consecutive scans representing the same static environment, the relative motion between them can be estimated using scan matching algorithms. Incremental execution of scan matching exploits a first method for online localization and mapping, calling this method *registration*. The general idea of scan matching approaches can be summarized as follows. At any point in time, the robot is given a measurement from the laser scanner. After the robot moved further on and after taking a new measurement of the environment, the robot determines the most likely relative pose w.r.t. the previous scan. It does so by trading off the consistency of the former scan with the latter one. Several approaches to measure scan consistency have been proposed. Common techniques in the context of laser range scans are the iterative closest point (ICP) [33], iterative matching range point (IMPR) [34], and the popular iterative dual correspondence (IDC) [35]. Pfister et al. proposed an interesting point-wise scan matching approach [36]. In this algorithm, each points is given a weight, based on the contribution to the overall matching error and according to its uncertainty. Although the approach obtained good evaluation results, the main drawback remains the high computation cost which prevents the use in a real time scenario. The SLIP method [37] establishes point-wise correspondences using a probabilistic distance metric for robust rejection of outliers. Minguez et al. present a metric-based ICP (MbICP) [38]

matching algorithm by point-wise matching dense range scans. It uses a geometric distance that takes into account the translation and rotation of a robot simultaneously. Given the relative motion between two acquired scans, the map is then extended by the new measurement using the pose at which this measurement was taken.

The key limitation of these approaches lies in the drifting error iteratively accumulated. Once the location at a fixed time has been computed it is not revised afterwards so that the robot cannot recover from registration errors. Although they have been proven to be able to correct enormous errors in odometry, the resulting maps often are globally inconsistent.

1.1.2 Bayesian Filters

Since the 1990s, the probabilistic paradigm has been predominant in SLAM problems [39]. Modeling and characterizing uncertainty and sensor noise in robot mapping are the main reasons for the popularity of these probabilistic techniques [40]. The probabilistic estimation methods primarily used for localization and mapping have been Bayesian filters like Kalman Filters (KF) and Particle Filters (PF). These techniques decouple estimation of the posterior distribution of state (i.e. robot state and map) into prediction and correction steps according to the Bayes' theorem. Prediction computes the prior distribution using robot controls. Correction estimates the posterior through comparison with sensors measurements.

Kalman filters are Bayesian derived filters that represent priors and posteriors using Gaussians distributions. Thus, KF SLAM relies on the hypothesis that the state prediction and the measurement functions are linear with additive Gaussian noise. In literature, there are two main extensions of KF used for nonlinear systems: the Extended Kalman Filter (EKF) and Information Filtering (IF) or Extended IF (EIF). Since the motion and observation models are often nonlinear in real world problems, the EKF accommodates the nonlinearities by approximating them using linearization. EKF is the most popular variant of Gaussian filters [41, 42, 43, 44]. Instead, the IF propagates the inverse of the state error covariance matrix. The main advantages of the IF filter over the KF are a higher accuracy in state estimation, provided by simply summing the information matrices and vector data [45], and a higher stability

than KF [46]. Furthermore, EKF is relatively slow when estimating high dimensional maps since each measurement affects all parameters of the posterior distribution [47]. Moreover, IF have some important limitations when applied to nonlinear systems. IF requires several inversions of matrices both in prediction and update steps. For high dimensional state spaces the IF become computationally heavier than the Kalman filter due to all the matrix inversions needed. Hence, these operations motivate the popularity of EKF compared to EIF [48]. When the information matrix possesses a particularly sparse structure, these limitations are overcome. In this case, information filters can be thought of as graphs, where states are connected whenever the corresponding off-diagonal element in the information matrix is non-zero. When the information matrix is (approximately) sparse, some algorithms allow development of an extended information filter that is significantly more efficient than both Kalman filters and non-sparse Information Filter [49, 50].

When nonlinearity in process and observation models are impossible to linearize or linearization is computationally expensive, the use of Unscented Kalman Filter (UKF) [51] overcomes the problem. KF performs optimally with linear system dynamics. For nonlinear models, EKF approximates the optimal terms through linearization of the dynamic equations. Therefore, EKF can be viewed as a first-order approximation of the optimal solution. In this case, divergence in state estimation is possible due to large errors in the true posterior mean and covariance provided by the approximation. In UKF the state distribution is represented by a minimal set of carefully chosen sample points (sigma points). These sample points provide a better approximation of mean and covariance of the prior distribution. When these points are propagated through the nonlinear system, the resulting points capture the posterior mean and covariance accurately to the 3rd order of the nonlinearity. In long duration missions, KF and its variations exhibit one of the main drawbacks: the number of landmarks will increase and, eventually, computer resources will not be sufficient to update the map in real-time. The advantage is that it provides an optimal Minimum mean-square Error (MMSE) estimate of the state (robot and landmark positions), and a strong convergence in covariance matrix estimation. However, the Gaussian noise assumption restricts the adaptability of the KF for data association and number of

landmarks.

A Particle Filter (PF) [52] is also a recursive Bayesian filter but it is based on a sequential Monte-Carlo (SMC) approach. SMC represents distribution through a set of random samples, called *particles*. Since the posterior distribution is not represented by a function with parameters but by samples, PF is also called *non-parametric*. This representation enables effective treatment of highly nonlinear models and sensors that cannot be modeled by Gaussian distributions. The computational complexity of this techniques is significantly higher and unsuitable for estimation problems with large state dimension, like mapping problems [53]. Therefore, there are few works that deal with the SLAM problem using a combination of PF with other techniques (e.g. Rao-Blackwellisation) as the FastSLAM [53] and the FastSLAM2.0 [54]. Since in SLAM problem landmarks estimates are conditionally independent given the robot path, FastSLAM takes advantage of this important characteristic and decomposes the SLAM problem into a robot localization problem, and a collection of landmark estimation problems that are conditioned on the robot pose estimate.

1.1.3 Graph SLAM

In the graph-based SLAM approach [48], the robot trajectory is described as a graph structure where the nodes represent the robot poses and the edges between the nodes represent the spatial relation between these poses. The edges contain for instance loop closures or odometry information. In general, the nodes can also represent features in the world, such as virtual landmarks extracted from images and laser point clouds; nodes may also represent physical landmarks such as trees or structures provided by a map. Typically, graph SLAMs problem are divided in two different steps. The first step is the creation of a graph (factor-graph), by identifying nodes and edges between the nodes based on sensors data. This step is often solved by a front-end system. The second step entails finding the configuration of the nodes that minimize the error induced by edges from the front-end. This last step exploits the computation of the maximum-likelihood map; a system solving it is typically referred to as a back-end. Although SLAM back-ends are not the focus of this thesis, in the last decade several approaches have been proposed and discussed. Nonlinear least squares optimization

was used in approaches like iSAM [55]. Relaxation techniques, such as Gauss-Seidel have been used for a while [56, 57]. Olson et al. introduced stochastic gradient descent, a method which increases efficiency and solves pose graphs even with large initial errors [58]. Later, Grisetti et al. extended the approach towards non-flat environments with TORO [59], a non-linear constraint network optimization framework. HOG-MAN [60] is an optimization approach that proved to outperform the simpler methods operating in Euclidean space; the approach stems from the insight tha,t due to the involved rotations, SLAM cannot be entirely and correctly modeled using flat, Euclidean spaces. Combining the insights and learned lessons from HOG-MAN and TORO, *g2o* [61] can be seen as the state of the art approach to solve large-scale SLAM problems containing several variables such as both poses and landmarks, and leading even to simultaneous calibration, localization, and mapping.

Graph-based SLAM front-end systems aim to identify the geometrical relations between multiple nodes in the graph, by interpreting extracted data from proprioceptive and exteroceptive sensors. Since for relatively close robot poses relations by nodes are typically computed by odometry or integrating inertial measurements, in long runs we can expect the robot position estimation to be significantly different from the correct position due to drift error. Front-end systems process sensor data to identify when the robot is back to a previously visited place, i.e. identifying loop closures. The earliest loop-closure systems were based on shape registration of laser scans (section 1.1.1). Weiss et al. [62] proposed a cross correlation based laser scan matching method where each laser scan is represented with a histogram and the matching is computed by maximizing the cross correlation between them. Since then, significant improvements have been made in front-ends based on laser scan matching. Biber et al. [63] represents each scan with a normal distribution. In this method, called normal distribution transform (NDT), the laser scans are divided in small buckets which are represented with Gaussians distribution. The correct transformation between scans is found by maximizing the overlap of points with the computed Gaussians. In the correlative scan matching method proposed by Olson [64], instead of approximating local points with a Gaussian as in the NTD algorithm, a rasterized cost table based on the log probability of observing a new point is built. The

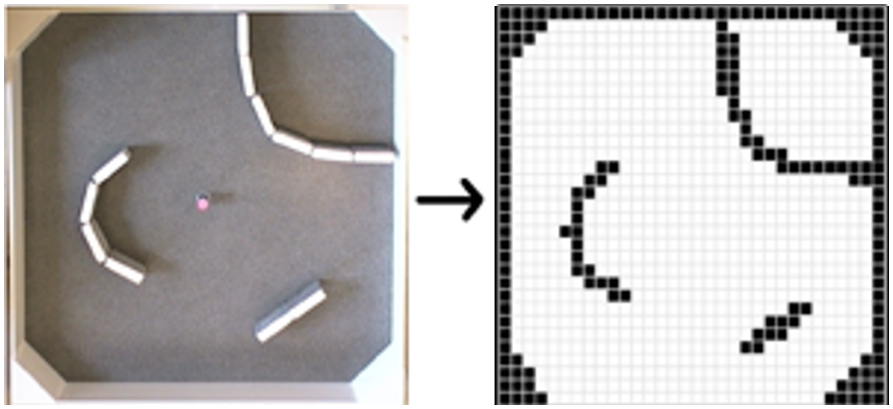


Figure 1.1: Example of grid map extracted from a toy sample.

cost table is used to evaluate different alignments providing a solution initialization independent.

Other authors, following the features paradigm evolved by computer vision, extract local descriptors from laser scans and match these descriptors to perform alignment [65, 8]. In the recent years there have been significant improvements in scan alignment based on ICP and its variants. Many authors have extended scan matching to 3D [66, 67] as well as modeled with high accuracy the registration errors for better mapping results [68, 69].

1.2 Feature extraction from LIDAR

In the early days of robot localization, a synthetic, rich and easy to compute description of the environment was necessary to enable robot to operate with safety and accuracy. So called occupancy grid maps [1, 2, 3, 4] were therefore a natural representation of the environment at that time. Grid maps, can be seen as a quantization of the environment into discrete regions or cells that cover a given area of the environment. To each of these cells is associated a value that expresses the probability to be occupied by an obstacle. Figure 1.1 illustrates this idea.

Apart from standard geometric features like segments, there are few works re-

lated to 2D range keypoint features; these can be classified into two categories. The first category includes methods that convert the laser scan into an image and compute keypoints using an existing computer vision method [9, 10]. This approach enables the re-use of many algorithms proposed in computer vision, but it is computationally expensive, requires rasterization of laser measurements, and suffers from inconsistencies due to obstacle projection in an image. The second category includes keypoint features explicitly designed for the laser scan peculiarities. Indeed, the scan radial order is taken into account in all the operations involving neighbor points, e.g. smoothing, downsampling, curvature computation. This second category includes FLIRT [8], which detects interest points according to properties such as range-differences or curvature and associates a descriptor to them.

1.2.1 Laser Scan Keypoints and Descriptors

During the last two decades, several keypoint features have been proposed for both images and point clouds. SIFT [70] is often considered the first example and the prototype of keypoint features with descriptor. SIFT achieves scale invariance through point extraction at different scales and distinguishes points using descriptors. In later years, several other keypoint methods in computer vision have been proposed [5, 6]. The keypoint feature paradigm has also been successfully applied in the context of 3D perception and shape recognition [7]. Extraction of significant local features from planar range finder data is more difficult due to the limited number of points in a scan compared to an image, and the peculiar distribution of neighbor points. Hence, the occupancy grid approach is prevailing in works on laser scan processing [4, 2], and elementary geometric features like segments [71], curvature points [72], and B-splines [73] have been used in association with laser scanners.

One of the first attempts to define laser keypoints with descriptor is illustrated in [74], where an orientation histogram obtained from the local submap is associated to each selected high-curvature point. This approach is improved in a subsequent work [67], but the global descriptors still represent a complete submap. The only true examples of keypoint features with descriptors for laser scans are proposed in [9, 10, 8]. Li and Olson [9] propose to rasterize laser scans and to extract interest points

from the obtained image using Kanade-Tomasi corner detector. The SIFT descriptor is then computed on the neighborhood of each point. The method is improved in the successive work [10] through scan smoothing pretreatment, more efficient corner extraction and candidate suppression. The main disadvantage of this approach lies in the inaccuracies resulting from rasterization.

Tipaldi and Arras [8] proposed FLIRT, the first keypoint feature explicity designed for planar laser scans. The input laser scan is rescaled several times and keypoints are detected at different scales according to parameters like curvature, normal and range. Next, the local shape is encoded by two descriptors, one representing the polar points distribution and the other representing occupancy. While FLIRT is a practical and conceptual improvement over the previous approaches, the detection is based on a single parameter, which may result into the identification of multiple and potentially unstable interest points. For example, it can find keypoints where discontinuities due to specific viewpoint occlusions occur, and also in unstable isolated points. Furthermore, the multiscale approach, which reduces the dependency from neighbor point distribution due to angular resolution and distance, is computationally expensive and not always effective, since it is performed only through local smoothing without downsampling.

Human-made indoor environments often consist of straight linear walls and architectural elements arranged along orthogonal directions. Laser scans obtained in such environments are composed of several points aligned along two orthogonal dominant directions. This condition can be exploited to find stable keypoints, named Orthogonal Corners (OC) [11] at the intersection of orthogonal line pairs. To find the dominant direction of the laser scan points, the algorithm computes the *Hough Spectrum* [75] of such points.

1.2.2 Feature Based Loop Closure

The approaches to data association and loop closure are strongly dependent on the map model. The recent proposal of keypoint features for planar range finders has enabled adaptation of pairing methods designed for features and landmarks. Pairing algorithms produce explicit pairs of corresponding features and include joint compat-

ibility test [76], Correspondence Graph [77] and affine Hough-based assocation [78]. However, point-to-point association is fast, robust and feasible only between small point sets thus the keypoints are usually collected into smaller local maps. Several state-of-the-art methods address global localization and loop closure problems by matching signatures, which encode the relations between the keypoints of a local map. Tipaldi et al. [14] propose geometrical FLIRT phrases (GFPs) to represent the keypoints detected in a laser scan. GFPs extend the bag-of-words techniques, popular in computer vision, to planar range finders using FLIRT descriptors. Instead of relying on the weak descriptors, Geometric LAndmark RElations (GLARE) [16] are based on geometric invariants in a keypoint set like pairwise distances and angles. Although robust, GLARE signature is not invariant to rotation, and geometrical surface relations (GSR) have been proposed to overcome this limitation [15]. The loop closure performance of these signatures depends on the stability of the keypoints detected from the range scans.

1.3 Techniques for AGV Calibration

In several localization and mapping applications, algorithms like EKF and graph SLAM are widely used due to their robust and reliable solutions. The data processed by those algorithms are often raw data acquired by on board sensors. In mobile robot localization, sensor data are the main source of information to estimate poses and build maps. The problem of identifying configuration parameters like sensor position w.r.t. the robot logical point or working thresholds is called calibration. Calibration of mobile robots has been addressed from several points of view by academic research. Calibration may refer only to the estimation of the robot kinematic parameters used in odometry computation or may include the geometrical relationship between the robot and its sensors. During the last two decades, research has addressed the assessment of mobile robot intrinsic parameters, of the extrinsic parameters of a sensor mounted on it, or of both sets of parameters.

1.3.1 Manual Odometry Calibration

Most of odometry calibration literature is devoted to differential drive robots. Several wheeled mobile robots are designed with a differential drive actuation system due to the simplicity of such configuration. In particular, the parameters to be estimated are the two wheel diameters and the wheelbase distance. Borenstein et al. [22] likely proposed the first specific calibration method for differential drive robots, the *UMBmark*. This technique requires the robot to move along a square trajectory in both directions, clockwise and counterclockwise, and measures the displacement between the final and the starting points after the execution of a loop to compute the correction factors of wheel-base and wheel diameters. This approach has been applied also to other kinematic models such as the car-like [26] or the tricycle [79], but mainstream research has been focused on differential drive. Odometry calibration is coupled with the calibration of internal sensors like gyroscopes and IMUs used to correct odometry [79]. Currently, similar procedures are often used to estimate odometric parameters in industrial practice: the robot follows specific paths (typically straight lines or loops) and the distance between the expected path with initial parameters and the measured one is used to correct parameters. Such distance is measured either manually or through an external absolute positioning system. Moreover, each calibration parameter is estimated sequentially after performing a specific step, instead of performing a simultaneous optimization. The separate assessment of each parameter is usually less accurate.

1.3.2 Calibration based on Filtering Methods

In several works the calibration problem has been addressed using the same Bayesian filtering algorithms adopted for robot localization. The system state consists of both the robot pose and its kinematic parameters, although the latter do not change or slowly change over the time. Larsen et al. [80] and Martinelli et al. [23] present augmented Extended Kalman Filter (AEKF) algorithms that simultaneously localize and calibrate the mobile robot. The method illustrated in [81] jointly uses a gyroscope, the wheel encoders, and a GPS unit in a Kalman filter to correct systematic errors.

In [18] a simultaneous SLAM and calibration algorithm specific for feature maps is presented. These works are designed for differential drive and, thus, cannot be applied to other drive models. Furthemore, they do not simultaneously estimate the intrinsic and extrinsic parameters.

Calibration of on-board sensors through EKF is a rather straightforward step. Extrinsic calibration parameters usually describe the pose of a sensor w.r.t. a common reference frame fixed on the robot. The EKF has been applied to the calibration of different kinds of sensors or of heterogeneous sensors. Early examples of camera calibration techniques based on Kalman filter can be found in [82], the latter specific for eye-in-hand cameras. The extrinsic parameters of laser scanners are estimated by tracking moving targets [28] or by comparing the robot pose evolution and the landmark measurements [19]. Foxlin [27] proposed a general EKF framework that allows localization and calibration of multiple sensors. Martinelli et al. [20] describe an EKF for assessing the parameters of a camera mounted on a robot during the robot motion. This work presents one of the first observability analysis for a calibration problem based on discrete-time system state evolution, which has been further developed in [83]. The authors also provide an observability analysis proving the formal correctness of the calibration process. Mirzaei and Roumeliotis [84] illustrate a method for calibrating a camera and an inertial sensor using a Kalman filter. The works discussed above are designed only to estimate the pose of one or more sensors w.r.t. the robot or to another sensor.

1.3.3 Calibration based on Least-square Optimization

Least square optimization has been used to estimate both intrinsic and extrinsic parameters. Historically, optimization is the earliest approach to specific calibration problems in robotics and computer vision like pinhole single and stereo cameras [85, 86] and eye-in-hand cameras [87, 88]. These techniques usually compute a closed-form initial estimation based on a simplified model and next refine this estimation by numerically optimizing the error associated to more complex sensor models taking into account optical distortions, offsets, etc. Calibration of heterogeneous sensor systems requires a target that is observable from different sensor domains and geome-

tries. Zhang and Pless [89] illustrate a method for a planar range finder and a camera exploiting a planar checkerboard.

In mobile robotics, the extrinsic or intrinsic calibration based on least-square optimization is more recent. Sometimes the function to be optimized is obtained from a stochastic formulation of the problem. Roy and Thrun [90] proposed the computation of the intrinsic parameters according to maximum likelihood criterion: the robot builds a map while moving, and estimates the likelihood function to be maximized. Antonelli et al. [91, 24] focus on the differential drive model and on the linear relationship between the observed robot motion and the quantities depending on intrinsic parameters. The calibration is achieved by solving a linear optimization problem and a bound on the calibration error is provided. An analysis of odometry error propagation is described in [92]. The work in [21] presents a calibration framework based on a graphical model formulation. The framework is rather general, but it does not explicitly consider the robot model and is potentially prone to convergence problems and inaccuracies. Underwood et al. [29] and Brookshire and Teller [30] address the calibration of multiple sensors mounted on a mobile robot using least-square optimization. The latter work has been extended from coplanar sensors to 3D range sensors like depth cameras [31].

The only complete odometry and sensor calibration method is reported in [17]. The estimation of intrinsic and extrinsic parameters is decoupled into two steps. The intrinsic calibration is the straightforward application of the differential drive solution [24]. The extrinsic parameters are obtained by comparing the robot motion and the sensor egomotion. The closed-form solution for extrinsic parameters proposed in [17] is dependent from the intrinsic calibration parameters. Such method cannot be used for the calibration of industrial AGVs, which are designed according to the tricycle kinematic model and not to the differential drive one.

Chapter 2

Automatic AGV Calibration

A common requirement of robotic systems is the capability to repeat operations with adequate precision. Precision is a fundamental requirement of environments where objects and products are moved by robot manipulators or *automated guided vehicles* (AGVs). AGVs are mobile robots equipped with forks or other grasping devices to transport objects and materials from a environment location to another according to assigned logistic policies. Since the exact location of each object is registered into a database, AGVs estimate their pose using an exteroceptive sensor, often a navigation range finder and artificial reflectors, to accurately pick and drop items. Figure 2.1 shows a typical AGV in an industrial setting.

The accuracy of robotic systems depends on the estimation of parameters that describe their motion and configuration. These parameters either represent the robot internal state or relate other sensors or devices to the robot. Respectively, the former are called *intrinsic parameters* and the latter *extrinsic parameters*. Intrinsic parameters usually describe the relationship between the actuators state and the robot kinematics. Extrinsic parameters relate the sensor measurements to the robot reference frame. The correct assessment of the AGV parameters affects its navigation accuracy, namely its odometry and localization. Hence, the aim of calibration is the accurate estimation of these parameters.

In this chapter I will go through the study and the development of a novel calibra-

Figure 2.1: An industrial AGV (Elettric80 CB25) equipped with a navigation laser scanner on the top of a telescopic pole.

tion method for tricycle wheeled robots [32]. Tricycle kinematics is a very common configuration for AGVs but, to my knowledge, no approaches have addressed the problem of automatic calibration of both intrinsic and extrinsic parameters of tricycle robots.

2.1 Problem Formulation

2.1.1 Standard Tricycle Model

The typical configuration of industrial AGVs is the tricyle wheeled robot model, which comprises three wheels and is also self-standing. The front wheel, also called *driving wheel*, is actuated and controlled by setting its steering angle and velocity.

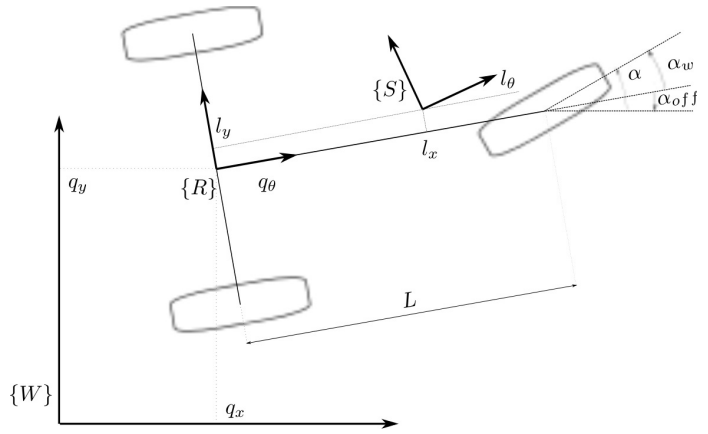


Figure 2.2: The tricycle model: $\{W\}$ world reference frame; $\{R\}$ robot frame in the robot logical point; $\{S\}$ sensor frame.

The two back wheels are coaxial, i.e. their rotation axis is the same, and passive. The robot reference frame $\{R\}$ is placed in the so called *logical point*, which is the projection of the driving wheel center on the back axis. Figure 2.2 illustrates the tricycle robot model. The robot is equipped with a sensor, which allows localization and egomotion estimation, and models the laser scanner usually mounted on industrial AGVs. The world reference frame $\{W\}$ is fixed in the environment. The sensor reference frame $\{S\}$ is rigidly mounted on the robot so that their relative pose is constant over time. Let $q = [q_x, q_y, q_\theta]^\top \in \mathrm{se}(2)$ be the state vector representing the position and orientation of reference frame $\{R\}$ w.r.t. the world frame $\{W\}$ and $l = [l_x, l_y, l_\theta]^\top$ be the constant relative pose of frame $\{S\}$ w.r.t. frame $\{R\}$. The kinematic model that describes the evolution of state vector q over time is

$$\dot{q} = \begin{bmatrix} v_{lp} \cos(q_\theta) \\ v_{lp} \sin(q_\theta) \\ \omega_{lp} \end{bmatrix} \tag{2.1}$$

The state vector and all the variables that depend on time will be sometimes explicitly written as a function of time, i.e. $q(t)$. Although the linear and angular velocities v_{lp}

and ω_{lp} are convenient to describe the motion of the logical point, the real controls for the tricycle robot are set on the front *steering wheel*, whose dynamic configuration is described by its linear velocity v_w and its steering angle α_w. The steering wheel state is controlled by two motors, one for rotating the wheel and the other for steering, and monitored by the corresponding encoders.

The distance traveled by the wheel is proportional to the *number of ticks* n_w counted by the driving encoder. The *driving scale* s_w is a constant that binds the distance travelled by the robot to the rotation of the wheel. The driving scale depends on the wheel radius (which changes in time as the wheel wears out), on the transmission gears, and on the angular resolution of the encoder. Hence, the wheel linear velocity v_w can be written as the product of the driving scale s_w and the derivative of tick number in time $\dot{n}_w$.

The robot rotation velocity depends on the *wheelbase* L, which is the distance between the logical point and the steering wheel axis, and on the steering angle α_w (see Figure 2.2). In industrial practice, the wheelbase is assumed to be accurately known from the mechanical design. The steering wheel direction for $\alpha_w = 0$ is orthogonal to the back wheel axis and the robot moves on a straight line. The encoder of the motor controlling the steer measures the *steering angle* α w.r.t. the encoder reference angle. Unfortunately, it is very difficult to mount the encoder such that its reference angle is perfectly aligned with the straight direction. Thus, there is a *steer offset* α_{off} between the measured steering angle α w.r.t. the reference angle and α_w, so that $\alpha_w = \alpha + \alpha_{off}$ as shown in Figure 2.2. Thus, the linear and angular velocities of the logical point in equation (2.1) can be written as

$$v_{lp} = v_w \cos\alpha_w = s_w \dot{n}_w \cos(\alpha + \alpha_{off}) \tag{2.2}$$

$$\omega_{lp} = \frac{v_w \sin\alpha_w}{L} = \frac{s_w \dot{n}_w \sin(\alpha + \alpha_{off})}{L} \tag{2.3}$$

The range finder mounted on the robot is such that the scanning plane is parallel to the ground plane. Hence, the sensor pose $s(t)$ w.r.t. the world reference frame $\{W\}$ is conveniently described by the robot pose $q(t)$ and the relative pose of the sensor $l = [l_x, l_y, l_\theta]^\top \in \mathrm{se}(2)$. The relationships between l, $q(t)$ and the other poses can be expressed using standard compounding and inversion operators on special Euclidean

Lie algebra se(2), respectively $\oplus$ and $\ominus$ [93]. Furthermore, the symbol $\mathrm{R}(\cdot)$ refers, henceafter, both to the map from an angle $\beta \in S^1$ to the corresponding rotation matrix of $\mathrm{R}(\beta) \in \mathrm{SO}(2)$ and to the map from a vector $b \in \mathbb{R}^2$ to a skew matrix $\mathrm{R}(b) \in \mathbb{R}^{2\times 2}$ defined respectively as

$$\mathrm{R}(\beta) \triangleq \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix}, \mathrm{R}(b) \triangleq \begin{bmatrix} b_x & -b_y \\ b_y & b_x \end{bmatrix} \tag{2.4}$$

This abuse of notation allows us to write compact formulas and to straightforwardly compute the expressions. The position of the sensor is then equal to $s(t) = q(t) \oplus l$. The measurement values of actuators and sensors are acquired at given sampling times $t_1, \ldots, t_k$.

The problem addressed in this chapter is the estimation of the following parameters:

- *intrinsic parameters*: the driving scale of the wheel s_w, which relates the encoder ticks and the linear velocity, and the steering offset angle α_{off};
- *extrinsic parameters*: the laser scanner pose parameters $l = [l_x, l_y, l_\theta]^\top$.

The available data are the relative motions of the robot and the laser scanner frames at the sample times t_k with $k = 0, \ldots, n$. The choice of the number of samples is formally discussed in section 2.4 and experimentally assessed in section 5.2. In particular, the relative robot motion r^k corresponds to the relative motion of robot $\ominus q(t_{k-1}) \oplus q(t_k)$.

The value of r^k depends on the controls and the intrinsic parameters of the robot; sometimes the notation $r^k(s_w, \alpha_{off})$ will emphasize the dependence from intrinsic parameters. The sensor relative pose between the two time instants t_{k-1} and t_k is called $s^k = [s^k_x, s^k_y, s^k_\theta]^\top$. We assume that s^k are measured or computed using the sensor measurements. The measurements r^k and s^k are constrained by the following relationship

$$l \oplus s^k = r^k \oplus l \tag{2.5}$$

Given a value of l, the difference between the first and second members of equation (2.5) represents the error on extrinsic parameters l related to the k-th measurements of s^k and r^k. The best estimation of l can be found by minimizing a goal error function that could be defined as the sum of such square errors with $k = 1, \ldots, n$.

Definition 2.1.1 (Standard Tricycle Calibration, STC)**.** *Given the measured steering angle $\alpha(t)$ and the number of travelled encoder ticks n_w^k on interval $[t_{k-1}, t_k]$ with $k = 1, \ldots, n$, and the sensor egomotion $s^k = [s_x^k, s_y^k, s_\theta^k]^\top$ on each interval, a calibration algorithm computes the values of the intrinsic parameters α_{off} and s_w, and of extrinsic parameters $l = [l_x, l_y, l_\theta]^\top$.*

2.1.2 Asymmetric Tricycle Model

The parameter model given by equations (2.2) and (2.3) enables accurate estimation of robot motion for most applications. However, it does not fully model some peculiarities related to the mechanical structure or to the space geometry of the system. The model accuracy is compromised by mechanical clearance and friction, which are high in industrial AGVs like the one shown in Figure 2.1. In particular, the wheel steer axis is subject to torques, which are significantly different whether the robot moves forward or backward. Such difference is more clearly seen when the AGV carries a pallet using the bottom fork-lift. The different motion direction affects the value of the steering offset. For this reason, an *asymmetric model* with two different steer offsets α_F and α_B, respectively forward and backward steer offsets, has been developed. The asymmetry can be modeled by a piecewise function

$$\alpha_{off}(\dot{n}_w) = \begin{cases} \alpha_F & \text{if } \dot{n}_w > 0 \\ \alpha_B & \text{otherwise} \end{cases} \tag{2.6}$$

which can be substituted in equations (2.2) and (2.3). The new formulation has the disadvantage of a discontinuous value of α_{off}, but for our purposes the robot motion can be divided into segments of forward or backward motion with a single α_{off} value. Thus, a second formulation of the calibration problem can be given.

Definition 2.1.2 (Asymmetric Tricycle Calibration, ATC)**.** *Given the measured steering angle $\alpha(t)$ and the number of travelled encoder ticks n_w^k on interval $[t_{k-1}, t_k]$ with $k = 1, \ldots, n$, and the sensor egomotion $s^k = [s_x^k, s_y^k, s_\theta^k]^\top$ on each interval, a calibration algorithm computes the values of the intrinsic parameters α_F, α_B and s_w, and of extrinsic parameters $l = [l_x, l_y, l_\theta]^\top$.*

The solution of ATC problem is obtained from the same equations of STC problem, but using different constraints. Since the difference between the two formulations lies in intrinsic parameters, the estimation of the sensor pose l is reasonably obtained using the same method.

2.2 Calibration Method

The calibration method proposed in this chapter estimates both the intrinsic and extrinsic parameters by moving the robot with constant input controls. The hypothesis of constant input controls is not restrictive and yields simpler model equations while allowing accurate estimation of the robot motion. If its controls are constant, then a tricycle robot moves along circular path segments. The relative pose displacement of the sensor frame at the beginning and end of each path can be measured by the sensor egomotion. Such information within the control setpoints can be used to compute both the intrinsic and extrinsic parameters in two consecutive steps. First, the values of intrinsic parameters are estimated from odometry according to STC or ATC formulation. Second, the navigation sensor pose is computed by minimizing the square mismatch between robot motion and sensor egomotion.

2.2.1 Standard

The relative robot motion r^k on time interval $[t_{k-1}, t_k]$ is obtained by integrating the differential equation (2.1). It is convenient to relate the value of r^k with the kinematic variables α and $\dot{n}_w$ that represent the motion of the driving wheel. The linear and angular velocities of the logical point, respectively v_{lp} and ω_{lp}, depend from α and $\dot{n}_w$ according to the equations (2.2) and (2.3). The orientation and position can be straightforwardly obtained by a separated integration of the terms of equation (2.1) when the input control $\alpha(t)$ is constant on the time interval. Henceafter, the path travelled by the robot on time interval $t \in [t_{k-1}, t_k]$ with $\alpha(t) = \alpha^k$ and with constant sign $\dot{n}_w(t)$ is called k-th *path segment*. In particular, if $\dot{n}_w(t) > 0$, it is a *forward path segment*, otherwise if $\dot{n}_w(t) < 0$ it is a *backward path segment*. The expression of orientation increment is computed as

$$\begin{aligned} r_\theta^k =& q_\theta(t_k) - q_\theta(t_{k-1}) = \int_{t_{k-1}}^{t_k} \dot{q}_\theta(\tau)\, d\tau \\ =& \frac{s_w \cos\alpha_{off}}{L} \int_{t_{k-1}}^{t_k} \dot{n}_w(\tau) \sin\alpha(\tau)\, d\tau \\ &+ \frac{s_w \sin\alpha_{off}}{L} \int_{t_{k-1}}^{t_k} \dot{n}_w(\tau) \cos\alpha(\tau)\, d\tau \end{aligned} \tag{2.7}$$

If the steering angle $\alpha(t) = \alpha^k$ is constant in time interval $[t_{k-1}, t_k]$, then $\sin\alpha^k$ and $\cos\alpha^k$ can be exported from the above integrals. The quantities $n_w^k = \int_{t_{k-1}}^{t_k} \dot{n}_w(\tau) d\tau$ and $\alpha = \alpha^k$ can be measured by the encoders of the robot. Thus, the analytical expression of angular increment is

$$r_\theta^k = (n_w^k \sin\alpha^k)\, \frac{s_w \cos\alpha_{off}}{L} + (n_w^k \cos\alpha^k)\, \frac{s_w \sin\alpha_{off}}{L} \tag{2.8}$$

The control variables n_w^k and α^k have been deliberately separated from the unknown calibration parameters s_w and α_{off} since they can be measured by the robot encoders on actuated wheels and on the steer axis. The position component of relative robot motion, i.e. $r_{pos}^k = [r_x, r_y]^\top$, is equal to

$$\begin{aligned} r_{pos}^k &= \mathrm{R}\left(-q_\theta(t_{k-1})\right) \int_{t_{k-1}}^{t_k} \begin{bmatrix} v_{lp}\, \cos q_\theta(\tau) \\ v_{lp}\, \sin q_\theta(\tau) \end{bmatrix} d\tau \\ &= \frac{L}{\tan(\alpha^k + \alpha_{off})} \begin{bmatrix} \sin r_\theta^k \\ 1 - \cos r_\theta^k \end{bmatrix} \end{aligned} \tag{2.9}$$

Equations (2.8) and (2.9) give the value of relative robot motion on time interval $[t_{k-1}, t_k]$ under the assumption that the input controls are constant over the interval. Intrinsic calibration can be achieved by keeping constant input controls for specific time intervals.

Although the value of r_{pos}^k cannot be easily measured, the relative orientation r_θ^k is equal to the relative sensor orientation s_θ^k on $[t_{k-1}, t_k]$. Thus, all the terms of

equation (2.8) can be measured by internal or external sensors. All the unknown variables in such equation can be collected into the vector

$$\psi = \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \frac{1}{L} \begin{bmatrix} s_w \cos \alpha_{off} \\ s_w \sin \alpha_{off} \end{bmatrix} \tag{2.10}$$

Given the measured values of n_w^k, α^k and s^k for several path segments $[t_{k-1}, t_k]$ with $k = 1, \ldots, n$, a linear system $A_\psi \psi = b_\psi$ is defined by n instances of equation (2.8) with the matrix and known term vector

$$A_\psi = \begin{bmatrix} n_w^1 \sin \alpha^1 & n_w^1 \cos \alpha^1 \\ \vdots & \vdots \\ n_w^n \sin \alpha^n & n_w^n \cos \alpha^n \end{bmatrix}, \quad b_\psi = \begin{bmatrix} s_\theta^1 \\ \vdots \\ s_\theta^n \end{bmatrix} \tag{2.11}$$

If there are more than two independent equations, the linear system is overdetermined. Hence, the value of ψ that better meets the given conditions is the one that minimizes the error, i.e.

$$\psi^* = \underset{\psi}{\operatorname{argmin}} \|A_\psi \psi - b_\psi\|^2 \tag{2.12}$$

Such problem can be solved by computing the Moore–Penrose pseudoinverse of matrix A_ψ

$$\psi^* = (A_\psi^\top A_\psi)^{-1} A_\psi^\top b_\psi \tag{2.13}$$

The existence of the pseudoinverse of A_ψ is discussed in detail in section 2.4.1. Given the value of ψ, the corresponding intrinsic parameters are estimated from equation (2.10) as

$$\alpha_{off} = \operatorname{atan2}(\psi_2^*, \psi_1^*) \tag{2.14}$$

$$s_w = L\sqrt{\psi_1^{*2} + \psi_2^{*2}} \tag{2.15}$$

Since the wheelbase $L > 0$ is known by hypothesis, equations (2.14) and (2.15) provide the desired values of the two intrinsic parameters.

2.2.2 Asymmetric

The ATC problem in Definition 2.1.2 is an extension of the STC problem illustrated in the previous section. After the substitution of α_{off} with α_F and α_B depending on the motion direction, the equations (2.8) and (2.9) still hold. Let the path segments be sorted so that the first $\bar{n} < n$ segments are acquired during forward robot motion and the remaining ones during backward motion: $n_w^k \geqslant 0$ for all $k = 1, \ldots, \bar{n}$ and $n_w^k < 0$ for $k = \bar{n}+1, \ldots n$. The orientation equations (2.8) for each k-th segment are then split into two groups according to the direction. Following the same procedure, a different unknown vector ψ of 4 variables is defined as

$$\psi = \left[\begin{array}{c} \psi_F \\ \hline \psi_B \end{array}\right] = \left[\begin{array}{c} \psi_1 \\ \psi_2 \\ \hline \psi_3 \\ \psi_4 \end{array}\right] = \frac{1}{L} \left[\begin{array}{c} s_w \cos\alpha_F \\ s_w \sin\alpha_F \\ \hline s_w \cos\alpha_B \\ s_w \sin\alpha_B \end{array}\right] \tag{2.16}$$

The orientation equations are arranged into a linear system with matrix and known term vector

$$\begin{aligned} A_\psi &= \left[\begin{array}{c|c} A_{\psi,F} & 0 \\ \hline 0 & A_{\psi,B} \end{array}\right] \\ &= \left[\begin{array}{cc|cc} n_w^1 \,\mathrm{s}\,\alpha^1 & n_w^1 \,\mathrm{c}\,\alpha^1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ n_w^{\bar{n}} \,\mathrm{s}\,\alpha^{\bar{n}} & n_w^{\bar{n}} \,\mathrm{c}\,\alpha^{\bar{n}} & 0 & 0 \\ \hline 0 & 0 & n_w^{\bar{n}+1} \,\mathrm{s}\,\alpha^{\bar{n}+1} & n_w^{\bar{n}+1} \,\mathrm{c}\,\alpha^{\bar{n}+1} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & n_w^n \,\mathrm{s}\,\alpha^n & n_w^n \,\mathrm{c}\,\alpha^n \end{array}\right] \end{aligned} \tag{2.17}$$

$$b_\psi = \left[\begin{array}{ccc|ccc} s_\theta^1 & \ldots & s_\theta^{\bar{n}} & s_\theta^{\bar{n}+1} & \ldots & s_\theta^n \end{array}\right]^\top \tag{2.18}$$

where the symbols $\mathrm{c}\,\alpha^k = \cos\alpha^k$ and $\mathrm{s}\,\alpha^k = \sin\alpha^k$ have been introduced for brevity. Thus, the intrinsic parameters for ATC problem can be solved by minimizing $\|A_\psi \psi - b_\psi\|$ subject to the condition $\psi_F^\top \psi_F = \psi_B^\top \psi_B$. This consistency constraint is quadratic and comes from the observation that the driving scale s_w is the same in both forward

and backward motions. The goal function and constraints can be written as explicit quadratic functions

$$\begin{aligned} \min \ & \tfrac{1}{2}\psi^\top M_\psi \psi - P_\psi^\top \psi + \tfrac{1}{2} b_\psi^\top b_\psi \\ \text{s.t.} \ & \psi^\top W_\psi \psi = 0 \end{aligned} \tag{2.19}$$

where the matrices M_ψ, P_ψ and W_ψ have the following explicit expressions

$$M_\psi = A_\psi^\top A_\psi = \left[\begin{array}{cc|cc} m_1 & m_2 & 0 & 0 \\ m_2 & m_3 & 0 & 0 \\ \hline 0 & 0 & m_4 & m_5 \\ 0 & 0 & m_5 & m_6 \end{array}\right] \tag{2.20}$$

$$P_\psi = A_\psi^\top b_\psi = \left[\begin{array}{cccc} p_1 & p_2 & p_3 & p_4 \end{array}\right]^\top \tag{2.21}$$

$$W_\psi = \left[\begin{array}{c|c} I_2 & 0 \\ \hline 0 & -I_2 \end{array}\right] = \left[\begin{array}{cc|cc} w_1 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 \\ \hline 0 & 0 & w_3 & 0 \\ 0 & 0 & 0 & w_4 \end{array}\right] \tag{2.22}$$

where $w_1 = w_2 = 1$ and $w_3 = w_4 = -1$. Problem (2.19) is a quadratic programming problem subject to a quadratic equality constraint (QPQEC) [94]. The feasibility of such problem and the uniqueness of its solution depend on the properties of matrices W_ψ and M_ψ. Matrix W_ψ is always full rank. M_ψ is positive semi-definite by construction whereas the conditions for its being positive definite will be given in section 2.4. Its solution requires a change of variables $\xi = S_c V^\top \psi$ to diagonalize matrix M_ψ, where

$$V = \left[\begin{array}{cc} \mathrm{R}\left(\frac{\mathrm{atan2}(-2m_2, m_3 - m_1)}{2}\right) & 0 \\ 0 & \mathrm{R}\left(\frac{\mathrm{atan2}(-2m_5, m_6 - m_4)}{2}\right) \end{array}\right]$$

and S_c is a diagonal matrix whose diagonal elements are $s_{c,i} \in \{-1, 1\}$ with $i = 1, \ldots, 4$ (hence, $S_c^\top = S_c^{-1} = S_c$). Each $s_{c,i}$ is chosen such that all the elements c_i

of vector $c = S_c V P_\psi$ have a non-positive value, i.e. $c_i \leqslant 0$. Hence, it is sufficient to compute VP_ψ and to set $s_{c,i} = -1$ for the positive elements of VP_ψ, otherwise $s_{c,i} = 1$. After the change of variables, we obtain a problem equivalent to (2.19) with form

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{i=1}^{4} d_i \xi_i^2 + \sum_{i=1}^{4} c_i \xi_i \\ \text{s.t.} & \sum_{i=1}^{4} w_i \xi_i^2 = 0 \end{array} \tag{2.23}$$

where d_i are the non-negative elements of diagonal matrix $V^\top S_c^\top M_\psi S_c V$ (M_ψ is positive semi-definite), and c_i are the elements of vector $c = S_c V P_\psi$. The matrix W_ψ is invariant to the change of variables since $S_c V^\top W_\psi V S_c = W_\psi$.

Proposition 2.2.1. *If $c_i \leqslant 0$ for each i, then there exists an optimal solution $\xi^* = [\xi_1^*, \ldots, \xi_4^*]^\top$ of (2.23) with $\xi_i^* \geqslant 0$ for each i.*

Proof. Suppose that the optimal solution of (2.23) is s.t. $\xi_i^* < 0$ for some i. Then, the value of objective function, after changing ξ_i^* with $-\xi_i^*$, is less or equal than such supposed optimal solution, since $c_i \leqslant 0$ (strictly less if $c_i < 0$). □

Being a QPQEC, the Karush-Kuhn-Tucker (KKT) conditions hold and the Lagrangian associated to (2.23) can be defined with the multiplier λ for the equality constraint. The critical points of the Lagrangian function satisfy for $i = 1, \ldots, 4$ the conditions

$$\begin{aligned} d_i \xi_i + c_i + \lambda w_i \xi_i &= 0 \\ \xi_i = -\frac{c_i}{d_i + \lambda w_i} &\geqslant 0 \end{aligned} \tag{2.24}$$

where the inequality in (2.24) follows from optimality condition of Proposition 2.2.1. The inequality is satisfied when the Lagrange multiplier λ lies in the interval (A, B) defined as

$$A = \max_{i\,:\,w_i>0} \left\{ -\frac{d_i}{w_i} \right\} < \lambda < \min_{i\,:\,w_i<0} \left\{ -\frac{d_i}{w_i} \right\} = B \tag{2.25}$$

Observe that if both $A = B = 0$ (when M_ψ is not positive definite) the interval is empty. The expression of ξ_i in equation (2.24) can be substituted into the equality constraint of (2.23) yielding

$$g(\lambda) = \frac{1}{2} \sum_{i=1}^{4} w_i \left(\frac{c_i}{d_i + \lambda w_i} \right)^2 = 0$$

The derivative $g'(\lambda)$ is negative for $\lambda \in (A,B)$. Besides,

$$\lim_{\lambda \to A^+} g(\lambda) = +\infty, \quad \lim_{\lambda \to B^-} g(\lambda) = -\infty$$

Thus, there exists a unique λ^* such that $g(\lambda^*) = 0$ on interval (A,B) and can be straightforwardly found using any numerical technique for algebraic equations like bisection or Newton–Raphson methods. The back-substitution of λ^* into equation (2.24) allows us to find ξ^* and, recursively, ψ^*. Finally, the asymmetric intrinsic parameters are found as

$$\alpha_F = \operatorname{atan2}(\psi_2^*, \psi_1^*), \quad \alpha_B = \operatorname{atan2}(\psi_3^*, \psi_4^*) \tag{2.26}$$

$$s_w = L\sqrt{\psi^{*2}_1 + \psi^{*2}_2} = L\sqrt{\psi^{*2}_3 + \psi^{*2}_4} \tag{2.27}$$

Since there are two equations for s_w, in practice it is computed as the average of the two estimations to balance the potential (and very slight) floating-point arithmetic errors.

2.3 Laser position calibration

The aim of extrinsic calibration is to compute the pose of the sensor mounted on the robot represented by pose $l \in \mathrm{se}(2)$. The value of l is obtained from the comparison between the sensor egomotion measurements s^k over different path segments defined in section 2.1 and the robot motion r^k measured from odometry. Once the intrinsic calibration is solved, r^k can be computed using the intrinsic parameters and equation (2.8). Equation (2.5) describes the spatial relationship between these poses and can be expanded into position and angular parts as

$$\begin{bmatrix} l_{pos} + \mathrm{R}(l_\theta)\, s^k_{pos} \\ l_\theta + s^k_\theta \end{bmatrix} = \begin{bmatrix} r^k_{pos} + \mathrm{R}(r^k_\theta)\, l_{pos} \\ r^k_\theta + l_\theta \end{bmatrix} \tag{2.28}$$

where the subscript $\cdot_{pos}$ refers to the position coordinate vector of a pose.

The angular part of equation (2.28) has been already used to substitute $r^k_\theta = s^k_\theta$ in equation (2.7) to solve the STC and ATC problems. The position part of equation (2.28) enables us to estimate the value of extrinsic parameters l. In particular, the

position error on k-th measurement can be defined as

$$\begin{aligned} e^k_{pos} &= (l \oplus s^k)_{pos} - (r^k \oplus l)_{pos} \\ &= (l_{pos} + \mathrm{R}(l_\theta)\, s^k_{pos}) - (r^k_{pos} + \mathrm{R}\left(r^k_\theta\right) l_{pos}) \\ &= \underbrace{\begin{bmatrix} I_2 - \mathrm{R}\left(r^k_\theta\right) & \mathrm{R}\left(s^k_{pos}\right) \end{bmatrix}}_{Q_k} \underbrace{\begin{bmatrix} \varphi_{pos} \\ \varphi_{ang} \end{bmatrix}}_{\varphi} - r^k_{pos} \end{aligned} \tag{2.29}$$

where $\varphi_{pos} = [\varphi_1, \varphi_2]^\top = [l_x, l_y]^\top$ and $\varphi_{ang} = [\varphi_3, \varphi_4]^\top = [\cos l_\theta, \sin l_\theta]^\top$. The vector φ_{ang} is subject to constraint $\varphi^\top_{ang}\varphi_{ang} = 1$ to satisfy trigonometric consistency that can be written as

$$\begin{aligned} h(\varphi) = \varphi_3^2 + \varphi_4^2 - 1 &= \varphi^\top \begin{bmatrix} 0 & 0 \\ 0 & I_2 \end{bmatrix} \varphi - 1 \\ &= \varphi^\top\, W\, \varphi - 1 = 0 \end{aligned} \tag{2.30}$$

The error function can be chosen in order both to properly represent a distance from the consistent estimation and to allow the computation of its minimum. Such function must depend on all the measurements collected by the robot, while moving along the n path segments. Although more complex functions could weigh the different components of e^k_{pos}, it is convenient to use the square sum function defined as

$$\begin{aligned} E(\varphi) &= \sum_{k=1}^{n} {e^k_{pos}}^\top e^k_{pos} \\ &= \sum_{k=1}^{n} (\varphi^\top Q_k - {r^k}_{pos})^\top (Q_k\, \varphi - r^k_{pos}) \\ &= \varphi^\top M_\varphi \varphi - 2\varphi^\top P_\varphi + \left(\sum_{k=1}^{n} {r^k_{pos}}^\top r^k_{pos} \right) \end{aligned} \tag{2.31}$$

where

$$M_\varphi = \sum_{k=1}^{n} Q_k^\top \, Q_k = \begin{bmatrix} m_1 & 0 & m_2 & -m_3 \\ 0 & m_1 & m_3 & m_2 \\ m_2 & m_3 & m_4 & 0 \\ -m_3 & m_2 & 0 & m_4 \end{bmatrix} \tag{2.32}$$

$$m_1 = \sum_{k=1}^{n} 2(1-\cos r_\theta^k) \tag{2.33}$$

$$m_2 = \sum_{k=1}^{n} \left(s_x^k \, (1-\cos r_\theta^k) - s_y^k \, \sin r_\theta^k \right) \tag{2.34}$$

$$m_3 = \sum_{k=1}^{n} \left(s_x^k \, \sin r_\theta^k + s_y^k \, (1-\cos r_\theta^k) \right) \tag{2.35}$$

$$m_4 = \sum_{k=1}^{n} \left((s_x^k)^2 + (s_y^k)^2 \right) \tag{2.36}$$

$$\begin{aligned} P_\varphi &= \sum_{k=1}^{n} Q_k^\top \, r_{pos}^k = \sum_{k=1}^{n} \begin{bmatrix} (I_2 - R^\top(r_\theta^k)) \, r_{pos}^k \\ R^\top(s_{pos}^k) \, r_{pos}^k \end{bmatrix} \\ &= \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix}^\top \end{aligned} \tag{2.37}$$

Thus, the extrinsic calibration problem is equivalent to the constrained optimization problem with target function in equation (2.31) and constraint from equation (2.30), i.e. to

$$\begin{aligned} \min \quad & E(\varphi) = \varphi^\top M_\varphi \varphi - 2\varphi^\top P_\varphi + \text{const} \\ \text{s.t.} \quad & h(\varphi) = \varphi^\top W_\varphi \varphi - 1 = 0 \end{aligned} \tag{2.38}$$

There are two values of φ satisfying the above problem as discussed in section 2.4.1. Therefore, an additional constraint is added to select only one of two valid solutions as suggested in [17], for example $\varphi_1 \geqslant 0$ due to the frontal placement of the sensor. Since $\varphi_1 \geqslant 0$ is an inequality constraint, the KKT conditions must hold thanks to Slater's conditions. The Lagrangian point can be found by solving the linear system $(M_\varphi - \lambda W)\varphi = P$ under the constraint $h(\varphi)$. The system matrix can be decomposed as $(M_\varphi - \lambda W) = L_\varphi D_\varphi L_\varphi^\top$ according to the modified Cholesky decomposition with lower triangular and diagonal matrices

$$L_\varphi = \begin{bmatrix} I_2 & 0 \\ \frac{1}{m_1} \mathrm{R}([m_2, -m_3]) & I_2 \end{bmatrix}, \; D_\varphi = \begin{bmatrix} \mu_1 I_2 & 0 \\ 0 & (\mu_2 - \lambda) I_2 \end{bmatrix}$$

where $\mu_1 = m_1$ and $\mu_2 = (m_1 m_4 - m_2^2 - m_3^2)/m_1$ are the eigenvalues of matrix M_φ (both with multiplicity 2). The expression of φ_3 and φ_4 can be obtained by solving the linear system with the unknown Lagrange multiplier λ and substituted into the constraint.

The result of the substitution is the following second-degree polynomial whose solution gives the admissible values of Lagrange multiplier λ

$$\lambda^2 + b_\varphi \lambda + c_\varphi = 0 \tag{2.39}$$

where its coefficients are

$$b_\varphi = 2\mu_2$$
$$c_\varphi = \mu_2^2 - \frac{(m_1 p_3 - m_2 p_1 - m_3 p_2)^2 + (m_1 p_4 + m_3 p_1 - m_2 p_2)^2}{m_1^2}$$

Each $\lambda_{1,2}$ satisfying equation (2.39) can be back-substituted into the linear system and the two respective solutions $\varphi^{(1,2)}$ can be obtained. The existence of two solutions is due to the symmetries of the tricycle model equations. The previously discussed condition $\varphi_1 \geqslant 0$ allows the choice between the two outputs. Section 2.4 will provide additional insights into the nature of the solution.

2.4 Formal Discussion of Results

In this section, the formal issues of the calibration algorithm presented in section 2.2 are discussed.

2.4.1 Observability

Real AGVs have a unique set of calibration parameters for both STC and ATC problems. The algorithms proposed in the previous section estimate these parameters by finding the values that better match the given input controls and sensor observations. Thus, the calibration procedure is reduced to a set of constrained optimization problems. A formal analysis is required to assess the conditions guaranteeing the existence

and uniqueness of the solution both to check the correctness of the proposed methods and, from a practical point of view, to correctly choose the robot motion.

The internal state of a system is observable if its estimation from the input controls and sensor observations is feasible. There are several ways to ascertain the observability of a system as defined from system theory. In the context of tricycle robot calibration, the internal state consists of the intrinsic and extrinsic parameters which do not evolve in time. The method shown in section 2.2 consists of several equations that return the calibration parameters from the sensor measurements. Here, observability is constructively proved by setting the conditions on the input controls (the trajectories) such that those equations have a unique solution.

The solution of intrinsic calibration, in equations (2.12) and (2.19) respectively for STC and ATC formulations, and extrinsic calibration in equation (2.38) have the form of least-square optimization on quadratic functions. These quadratic functions are equal to the square of overdetermined linear equations representing the relationship between the measurements and the calibration variables. In the case of asymmetric intrinsic and extrinsic calibration, the variables of these linear equations are subject to additional consistency constraints. In the following, the observability of each set of calibration parameters is proven by showing that, under proper conditions, there is a unique solution to the optimization problem.

The estimation of intrinsic parameters for the standard tricycle model has the form of an overdetermined linear system, which is the argument of equation (2.12). The solution is defined by the pseudoinverse. A condition for the estimation of α_{off} and s_w, i.e. the observability of the two parameters, is given by the following proposition.

Proposition 2.4.1. *The solution of equation (2.12) exists and is unique iff the input dataset contains at least two trajectories k_1 and k_2, $k_1 \neq k_2$, s.t. the corresponding steering angles $\alpha^{k_1} \neq \alpha^{k_2} + i\pi$ for some i and $n_w^{k_1}, n_w^{k_2} \neq 0$.*

Proof. If. The solution of (2.12) exists if the left pseudoinverse of A_ψ in equation (2.13) exists. The pseudoinverse of a $n \times 2$ matrix exists if A_ψ is full rank, i.e. its rank is 2

in this case. The submatrix obtained from rows k_1 and k_2 has determinant

$$\det \begin{bmatrix} n_w^{k_1} \sin \alpha^{k_1} & n_w^{k_1} \cos \alpha^{k_1} \\ n_w^{k_2} \sin \alpha^{k_2} & n_w^{k_2} \cos \alpha^{k_2} \end{bmatrix} = n_w^{k_1} n_w^{k_2} \sin\left(\alpha^{k_1} - \alpha^{k_2}\right)$$

which cannot be zero due to the hypotheses.
Only if. Let the quadratic function of equation (2.12) have a unique minimum ψ^*, i.e. ψ^* is a critical point, $A_\psi^\top (A_\psi \psi^* - b_\psi) = 0$, and the Hessian matrix $A_\psi^\top A_\psi$ is positive definite. If $\alpha^{k_1} - \alpha^{k_2} = i\pi$ or $n_w^k = 0$ for all k, k_1, k_2, then $\det(A_\psi^\top A_\psi) = 0$ contradicting $A_\psi^\top A_\psi$ positive definiteness. □

The observability proof for intrinsic ATC and extrinsic parameters follows a similar pattern with the difference that there are other constraints on the variables. In section 2.2.2, the method for estimating the asymmetric intrinsic parameters has been illustrated and the discussion has shown that there is a unique global solution. The following proposition explicitly states the conditions for which this result holds.

Proposition 2.4.2. *Let $\alpha^{k_1} \neq \alpha^{k_2} + i\pi$ and $n_w^{k_1}, n_w^{k_2} \neq 0$ for at least two forward segments $0 < k_1 < k_2 \leqslant \bar{n}$, $\alpha^{k_3} \neq \alpha^{k_4} + j\pi$ and $n_w^{k_3}, n_w^{k_4} \neq 0$ for at least two backward path segments $\bar{n} < k_3 < k_4 \leqslant n$. Then, the problem (2.19) is feasible and its solution is unique.*

Proof. The discussion of calibration method in section 2.2.2 shows the existence and uniqueness of the solution according to [94], assuming that M_ψ is positive definite. The hypotheses on forward and backward path segments are sufficient to prove the positive definiteness of M_ψ, similarly to the proof of Proposition 2.4.1. □

Propositions 2.4.1 and 2.4.2 also provide criteria to correctly choose the minimum number of path segments and the steering angles.

It remains to be discussed the feasibility of extrinsic calibration, which is common between STC and ATC formulation. The first issue concerns the number of solutions of the extrinsic calibration problem. Under the conditions discussed in the following, there are two symmetric solutions $l = [l_x, l_y, l_\theta]^\top$ and $l' = [-l_x, -l_y, l_\theta + \pi]^\top$. However, the additional constraint due to the physical placement of the sensor allows their disambiguation.

The second critical issue is related to the existence of only two solutions. Although the extrinsic calibration is a QPQEC, it cannot be solved using the same procedure of intrinsic ATC due to the rank of quadratic constraint matrix W. The procedure in section 2.3 leads to a closed-form solution by solving a parametric linear system. The solution of such system depends on the positive definiteness of matrix M_φ, which is always at least semi-positive definite. The following proposition gives the additional conditions on the robot paths granting the feasibility of the problem.

Proposition 2.4.3. *Let $n > 1$, $r_\theta^{k_1} \neq 2\pi i$ and $\|s_{pos}^{k_2}\| > 0$ for some $0 < k_1, k_2 \leq n$ and $i \in \mathbb{Z}$. Then, the problem (2.38) is feasible and has two solutions.*

Proof. Observe that all the terms of the sum in equations (2.33) and (2.36) are non-negative and $m_1, m_4 \geq 0$. Hence, if there is at least one $r_\theta^{k_1} \neq 2\pi i$, then $\cos r_\theta^{k_1} < 1$ and $m_1 > 0$. Similarly, if there is $\|s_{pos}^{k_2}\| > 0$, then also $m_4 > 0$. The eigenvalues of M_φ are m_1 and $(m_1 m_4 - m_2^2 - m_3^2)/m_1$, both with multiplicity 2. Using triangular and Cauchy-Schwartz inequalities, it is straightforward to show that

$$m_1 m_4 \geqslant 2 \sum_{k=1}^{n} (1 - \cos s_\theta^k) \|s_{pos}^k\| \geqslant m_2^2 + m_3^2 \tag{2.40}$$

When the hypotheses are all satified, the strict inequality holds, the eigenvalues are all positive and M_φ is positive definite. □

2.4.2 Error Propagation

Numerical robustness is an important property to be considered in order to successfully apply the proposed technique to real world AGV calibration. However careful the sensor measurements are, the input data of the calibration are invariably noisy and uncertain. There are several ways to assess the numerical robustness of an algorithm. The statistical evaluation of the calibration parameters computed in repeated trials and different conditions is the more direct and effective approach and will be presented in section 5.2. On the other hand, the numerical analysis of the algorithms allows a consideration of all the possible occurrences that cannot be experimentally reproduced and also the identification of all the most convenient conditions. The

propagation of error in a procedure can be measured by the variance or by the deterministic error of the computed parameters. Since the calibration parameter equations, such as equations (2.12), (2.19) and (2.38), are linear or nearly linear, their numerical stability depends on the *condition number* of the problem matrix.

The solution of intrinsic STC requires the solution of an overdetermined linear system where the matrix A_ψ depends on the parameters of the path segment, i.e. the steering angles α^k and the travelled path length n_w^k, and the known term vector b_ψ is a function of sensor angular displacements s_θ^k. Most of uncertainty lies in the egomotion estimation of s_θ^k, but the propagation of uncertainty is affected by condition number $\kappa(A_\psi)$ and, hence, by the choice of path segments. In the general case, the expression of $\kappa(A_\psi)$ cannot be straightforwardly obtained, but the path segments are chosen according to regularity criteria. For example, the AGVs perform paths with the same steering angle both on left and right, i.e. $\alpha^{n-k} = -\alpha^k$ (the order of α^k is not restrictive), and the angular length of all the paths is approximately equal to β, i.e. $n_w^k = \frac{\beta L}{\sin \alpha^k}$. The straight trajectory is not used in the proposed calibration procedure for a practical reason: it is easier to sequentially execute uniform circular paths in a bounded space by progressively increasing the value of steering $|\alpha|$. Although possible, the inclusion of a straight path segment would require a specific and time-consuming positioning of the robot. Hence, in the following the steering angles $|\alpha^k| > 0$ are never null. The minimum $|\alpha|$ corresponds to the maximum radius of path segments. The maximum value of $|\alpha|$ depends on the mechanical limits of the steering wheel.

Proposition 2.4.4. *Let matrix A_ψ in equation (2.11) be s.t. $\alpha^{n-k} = -\alpha^k$, $|\alpha^k| > 0$ and $n_w^k = \beta L / \sin \alpha^k$ where $k = 1, \ldots, n$ and β is the angular length of path segments. Then, the condition number of problem in equation (2.12) is equal to*

$$\kappa(A_\psi) = \frac{\max\left\{n, \sum_{k=1}^n \tan^{-2} \alpha^k\right\}}{\min\left\{n, \sum_{k=1}^n \tan^{-2} \alpha^k\right\}} \tag{2.41}$$

Proof. The condition number is obtained from the minimum and maximum singular values of A_ψ that can be found by computing $A_\psi^\top A_\psi$. Due to the symmetry hypothesis $\alpha^{n-k} = -\alpha^k$, the non-diagonal terms of $A_\psi^\top A_\psi$ are zero, $(\beta L)^2 \sum_{k=1}^n (1/\tan \alpha^k) = 0$. Thus, $A_\psi^\top A_\psi$ is diagonal with elements $(\beta L)^2 n$ and $(\beta L)^2 \sum_{k=1}^n \tan^{-2} \alpha^k$. The thesis

follows from the definition of condition number. □

Given the discussed hypotheses, the problem is well-conditioned if the average value of $|\alpha^k|$ is close to $\pi/4$. A steering angle above 50° stresses the steer mechanics and increases friction and slip between the rubber wheels and the ground. Thus, a trade-off between well-conditioned data given by relatively large steering angle and motion accuracy must be found.

The same considerations on error propagation extend to the intrinsic ATC problem. Although the asymmetric calibration involves the solution of a constrained linear system, the structure of matrix A_ψ in equation (2.17) is similar to the standard case. The main difference lies in the separate management of the path segments according to the direction of the AGV motion.

2.5 Discussion

In this chapter, a calibration method simultaneously computing the intrinsic and extrinsic parameters of an industrial AGV compliant to the tricycle wheeled robot model has been described. The calibration is performed by computing the parameters better fitting the input commands and the sensor egomotion estimation obtained from the sensor measurements. A closed form solution is provided for Standard Tricycle Calibration (STC) and Laser position calibration problems, while the Asymmetric Tricycle Calibration (ATC) is solved through a one-dimension numerical search. The methods are easily implementable in low computation systems as PLCs or industrial computers. Moreover, the observability property of the method has been formally proved and feasibility conditions on the input trajectory for the estimation have been provided.

Chapter 3

Fast and Adaptive LIDAR Features

The feature-based approach is used by a wide range of topics and applications in computer vision and depth imaging. Although there are several combinations of detectors-descriptors for image data and 3D point clouds, only few methods are available for 2D laser scan data. Localization and mapping using feature-based maps extracted from LIDAR data is a recently developed research topic in robotics.

In this section, I describe FALKO [11], a keypoints extractor for laser scans which exploits the concept of edge intersection in 2D range data. Next, two novel descriptors are illustrated: the first is derived from shape context descriptor [95], while the second represents the relative orientation between each keypoint and its neighborhood. The performance of FALKO and the proposed descriptors is then evaluated and comparedc with the state of the art of 2D laser scan features.

3.1 Keypoints

Laser scan data consist in a set of points which approximate the perceived environment shape. In LIDAR with low angular resolution, the approximation is less accurate. Accurate detection of interest points from noisy data can be performed only

after higher level inference than detection of simple geometric shapes like segments or circles. FALKO exploits the concept of edge intersection in 2D range data usually representing *corners* in the environment. This detector has been designed to be orientation invariant and point density independent.

3.1.1 FALKO

Let $\mathscr{S}$ be the point set from an input laser scan and $p_i \in \mathscr{S}$ (i index in the scan) a candidate interest point of the scan. The neighborhood of each candidate scan point p_i is defined as $C(p_i) = \{p_j \in \mathscr{S} : \|p_j - p_i\| \leq r_i\}$, where r_i is a radius which increases with $\|p_i\|$, the distance from the sensor origin of the considered point. Thus, an approximately uniform number of neighbors for p_i is guaranteed. In my implementation, the radius is computed as $r_i = a\exp(b\|p_i\|)$ where the parameters $a = 0.2$ and $b = 0.07$ are chosen s.t. $0.20\ m \lesssim r_i \lesssim 0.40\ m$ for ranges in $1 - 10\ m$.
A set of geometric considerations avoids evaluation of points which cannot be considered as candidate corners due to the lack of neighbors, geometric inconsistency or artifacts originated by range discontinuities. For each point p_i, the point set $C(p_i)$ is divided in two subsets

$$C_L(p_i) = \{p_j \in C(p_i) : j < i\} \tag{3.1}$$

$$C_R(p_i) = \{p_j \in C(p_i) : j > i\} \tag{3.2}$$

Figure 3.1 shows an example of this step on real extracted data. The first consideration is based on the cardinality of the two subsets (fig. 3.2). If $|C_L| < 2$ or $|C_R| < 2$, the point is discarded from the candidate corner set. For each remaining candidate points, two neighborhood endpoints x_L and x_R are defined:

$$x_L = p_{j_{min}} : \qquad j_{min} = \arg\min_j \{p_j \in C_L(p_i)\} \tag{3.3}$$

$$x_R = p_{j_{max}} : \qquad j_{max} = \arg\max_j \{p_j \in C_R(p_i)\} \tag{3.4}$$

The triangle $\triangle p_i x_L x_R$ is then evaluated as a rough approximation of the corner. Let $\overline{x_L x_R}$ be the base of the triangle. If the base length $\|\overline{x_L x_R}\|$ or the triangle height is less than $\frac{r_i}{\beta}$ the point is discarded. Parameter β is chosen taking into account specific

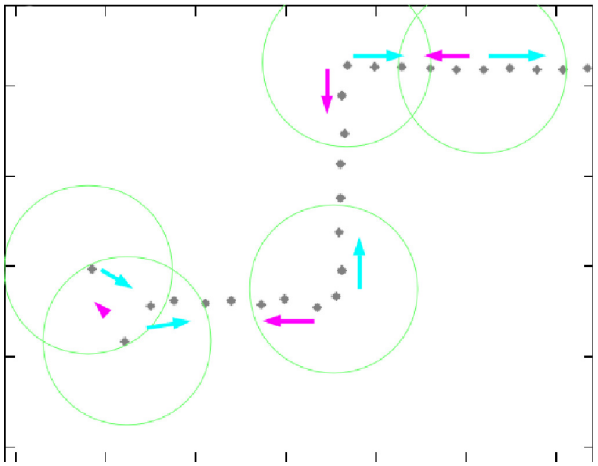


Figure 3.1: Example of FALKO keypoint extraction: green circles represents the neighborhood search range for each point p_i, while magenta and cyan arrows indicate respectively the $C_L(p_i)$ and $C_R(p_i)$ subsets.

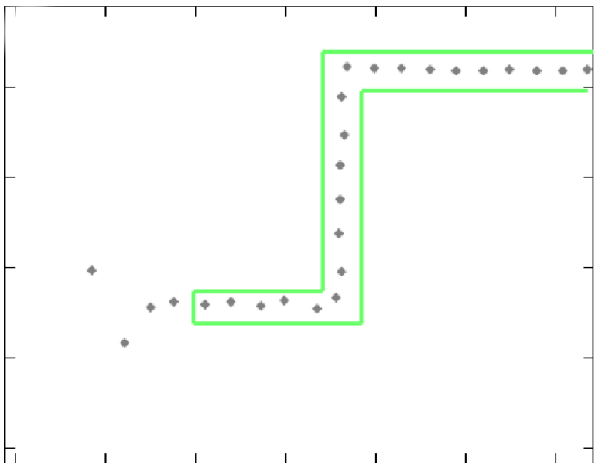


Figure 3.2: Example of FALKO keypoint extraction: set of points which have enough neighbors in both $C_L(p_i)$ and $C_R(p_i)$ subsets.

limitations on the aperture and subtended angle of the corner. In particular, greater values of β allow both wider and sharper corners to be selected as candidates. Figure

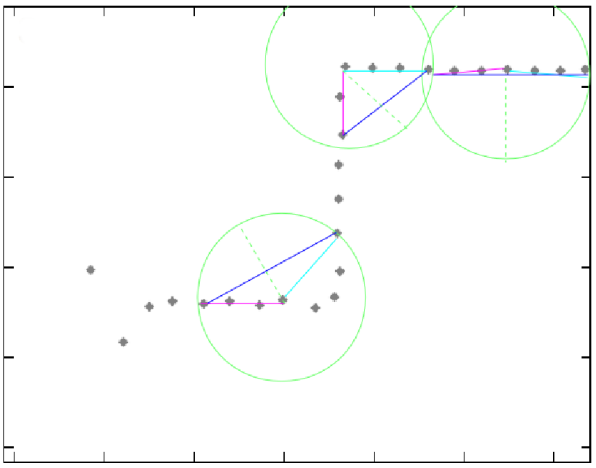


Figure 3.3: Example of FALKO keypoint extraction: triangle approximation of each candidate points computed by $\triangle p_i x_L x_R$.

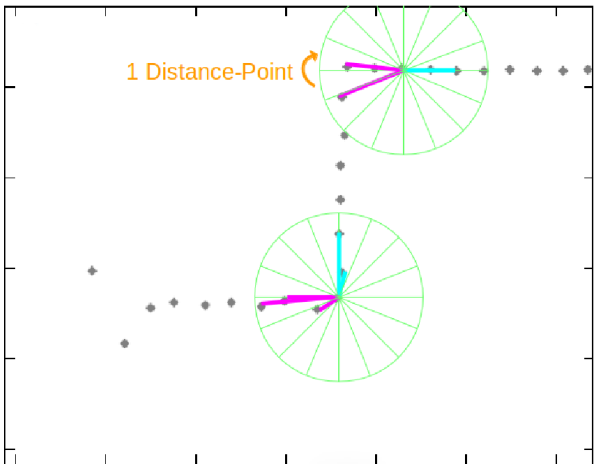


Figure 3.4: Example of FALKO keypoint extraction: polar grids used for score computation.

3.3 shows the triangle approximations for each candidate points. These conditions efficiently filter unsuitable corner candidates based on simple geometric properties. Then, for each candidate point which has passed the geometric verifications, a *cornerness* score is computed. A polar grid (fig. 3.4), which quantizes the space in cir-

cular sectors, is centered on the candidate point p_i. For each point $p_{j,L} \in C_L(p_i)$ and $p_{j,R} \in C_R(p_i)$, a quantized orientation w.r.t. the candidate point is computed as

$$\phi_{j,L} = \left\lfloor \frac{s_n}{2\pi} \tan^{-1}\left(\frac{p_{j,y} - p_{i,y}}{p_{j,x} - p_{i,x}}\right) \right\rfloor, \quad \forall p_j \in C_L(p_i) \tag{3.5}$$

$$\phi_{j,R} = \left\lfloor \frac{s_n}{2\pi} \tan^{-1}\left(\frac{p_{j,y} - p_{i,y}}{p_{j,x} - p_{i,x}}\right) \right\rfloor, \quad \forall p_j \in C_R(p_i) \tag{3.6}$$

where s_n is the number of circular sectors in the polar grid.
Let

$$d_\theta(\phi_1, \phi_2) = \left((\phi_1 - \phi_2) + \frac{s_n}{2}\right) (\text{mod } s_n) - \frac{s_n}{2} \tag{3.7}$$

be a distance function between the quantized orientations in circular sector units. The score for a candidate point is defined as

$$score_L(p_i) = \sum_{h=i-1}^{j_{min}} \sum_{k=h-1}^{j_{min}} |d_\theta(\phi_h, \phi_k)| \tag{3.8}$$

$$score_R(p_i) = \sum_{h=i+1}^{j_{max}} \sum_{k=h+1}^{j_{max}} |d_\theta(\phi_h, \phi_k)| \tag{3.9}$$

$$score(p_i) = score_L(p_i) + score_R(p_i) \tag{3.10}$$

The score function (3.10) measures the alignment of the two point sets C_L and C_R and it is orientation invariant. For each set, the more points are aligned in the same *direction* the smaller is the score value. Keypoints are then chosen as local minima of the score function in (3.10) with a non-maxima suppression procedure with range 0.20 m. Figure 3.6 shows an example of whole computation on synthetic data.

3.1.2 Subbeam Accuracy

FALKO extractor select a sets of keypoints $kp_i \in \mathscr{S}$ from the laser scan $\mathscr{S}$. Unfortunately, those keypoints are not guaranteed to represent the exact edge generating the corner. Indeed, due to the discrete quantization of the laser scan, two consecutive laser beams can miss the corner edge and hit respectively the right and left sides of the edge. To increase the detection accuracy and the stability of the keypoint, a *subbeam* evaluation is performed. The subbeam accuracy is reached approximating the

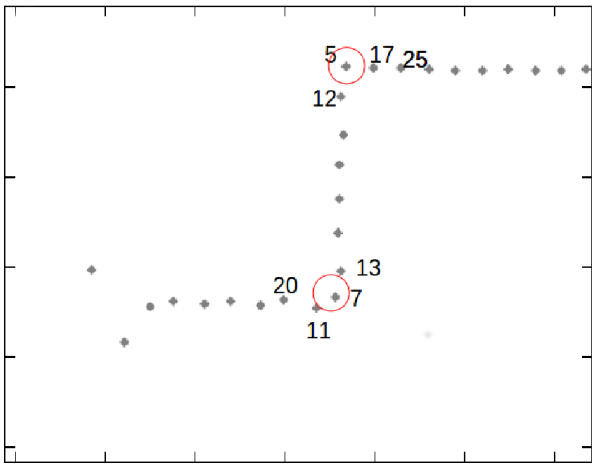


Figure 3.5: Example of FALKO keypoint extraction: score values computed with equation 3.10. Red circles indicates keypoints chosen as local minima of the score function.

two subsets $\{C_L(kp_i);kp_i\}$ and $\{C_R(kp_i);kp_i\}$ with respectively two lines, l_L and l_R. The approximation is performed with a least-square regression over the point subset. The intersection of the two lines gives the new keypoint $kp_i^\star$. To avoid mismatch of the subbeam detection due to outliers in the least-square regression, the new keypoint coordinates $kp_i^\star$ are accepted only if not too far from the former evaluation kp_i ($\|kp_i^\star - kp_i\| < 0.2m$).

3.2 Descriptors

While the main requirements for a good keypoint extractor are the stability and the repeatability of the point, the descriptor must be significantly discriminative to provide good matching performance. In this work, I propose two different descriptors. The first is a binary variant of the well known shape context descriptor proposed by Belongie et al. [95] and modified in [8]. The second is a histogram based descriptor which accumulates the relative orientation between the keypoint and its neighborhood.

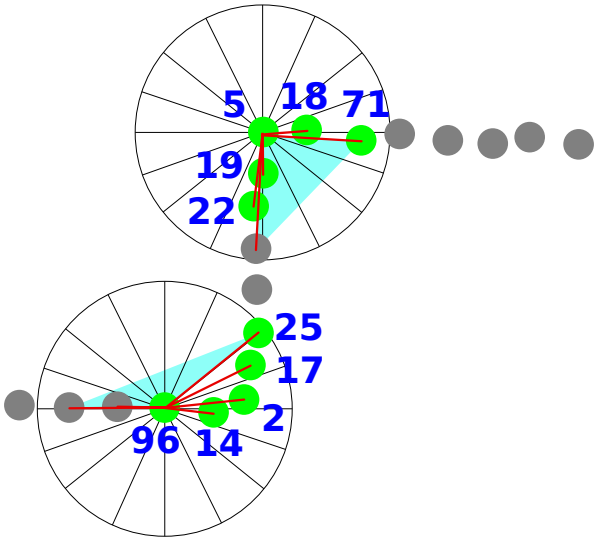


Figure 3.6: A complete example of FALKO keypoint extraction with scores from eq. (3.10) (blue) of each candidate point (green), corresponding triangle approximation (light blue), and orientation vectors (red).

3.2.1 BSC

Shape context is a local descriptor which represents the points distribution in a linear-polar histogram. The bins of the histogram centered on the keypoint count the neighbor points lying inside the region corresponding to the bin. The main drawback of this approach is its dependency from point density. Indeed, due to the angular quantization of the laser scanner, the same shape detected from two different viewpoints may results in two different histogram distributions. Figure 3.7(a) shows an example of shape context descriptor for the same shape viewed from two near poses. The two descriptors exhibit different peaks in the linear-polar histogram which can cause mismatch in the recognition of the keypoint. I propose a binary version of the local shape descriptor which is less sensitive to neighbor points density than the original one. Formally, given r_n and α_n, respectively the number of radial and polar quantization in the descriptor grid, for each keypoint kp the *BSC* descriptor is defined as the

grid

$$BSC_{mj} = \begin{cases} 1 & \text{if } \exists\ p_i : (p_i - kp) \in \overline{BSC_{mj}} \\ 0 & otherwise \end{cases} \tag{3.11}$$

where p_i is a point of kp neighborhood, $m = 1, \ldots, r_n$, $j = 1, \ldots, \alpha_n$ and $\overline{BSC_{mj}}$ is a cell in the linear-polar grid. A distance function for the BSC descriptor can be defined as the Hamming-like distance function

$$d_{BSC}(BSC_1, BSC_2) = \sum_{m=1}^{r_n} \sum_{j=1}^{\alpha_n} BSC_{1,mj} \oplus BSC_{2,mj} \tag{3.12}$$

where $\oplus$ is the XOR operator. Figure 3.7(b) shows the resulting descriptors computed for the same keypoint viewed from different poses.

3.2.2 CGH

Laser scan data provide only geometric information of the perceived environment. Thus, geometric shape or neighborhood point distribution are commonly used as interest point *description*. However, these approaches are viewpoint and density dependent. Therefore, I propose a relative orientation-based descriptor that exploits the viewpoint and density invariant property of orientation between points.
We compute a histogram based on the same radial grid presented in section 3.1.1. Given α_n, the number of histogram bins, for each point p_i in the keypoint kp neighborhood, a relative quantized orientation ϕ_i is computed as

$$\phi_i = \left\lfloor \frac{\alpha_n}{2\pi} \tan^{-1}\left(\frac{p_{i,y} - kp_y}{p_{i,x} - kp_x}\right) \right\rfloor \tag{3.13}$$

Then, for each point a discrete Gaussian distribution $\mathcal{N}(\phi_i, \sigma)$ is constructed. The *CGH* is defined as

$$CGH_j = \sum_{\phi_i} \mathcal{N}(j - \phi_i, \sigma) \tag{3.14}$$

where $j = 1, \ldots, \alpha_n$ and CGH_j is a cell in the radial histogram. Figure 3.7(c) shows two examples of CGH descriptor. Histograms have the same radial distribution although the two scans are perceived from two different robot poses. A normalization

of the histogram makes the descriptor less sensitive to neighbor points density, and a symmetric χ^2 test can be used to evaluate the distance between two CGH descriptors.

3.2.3 Computing Corner Orientation

Descriptor matching can be computed through the evaluation of a distance function. Since the same corner can be perceived from different view-angles, the resulting descriptors are not aligned w.r.t. the same start angle. Thus, for matching, descriptors are rotated according to the *corner orientation* estimated by the keypoint detector. For OC keypoints, the corner orientation is given by the bisector angle between two points aligned to the dominant direction $\bar{\theta}$ and its orthogonal direction $\bar{\theta} + \pi/2$. For FALKO keypoints, the corner orientation is computed with a variant of the *Intensity Centroid* [96]. First, for both sets C_L and C_R given by equations (3.1) and (3.2), the centroid is computed w.r.t. the interest point. The average of the two centroids is used to compute the orientation vector $\mathbf{o}$. The corner orientation is estimated as $atan2(\mathbf{o}_y, \mathbf{o}_x)$. This method results in good orientation estimation without point density dependency. Figure 3.7 shows some examples of corner orientation vectors which can be used to align the descriptors w.r.t. the same starting angle.

3.3 Feature evaluation

The aim of the experiments presented in this section is the evaluation and comparison of the proposed keypoint detector and descriptors w.r.t. the state of the art. The most important property of keypoints is their stability, i.e., the detection of the same point in the environment after changing viewpoint and regardless of laser scanner properties like noise level and resolution. On the other hand, the performance of descriptors depends on their capability to distinguish among different places of the environment. In order to make the comparison reproducible, I adopted the criteria employed in the experimental assessment of [8], which in turn follows the approach of Mikolajczyk et al. [97, 5]. Thus, I compared features by changing the viewpoint, the noise level and the number of measurements of a scan.

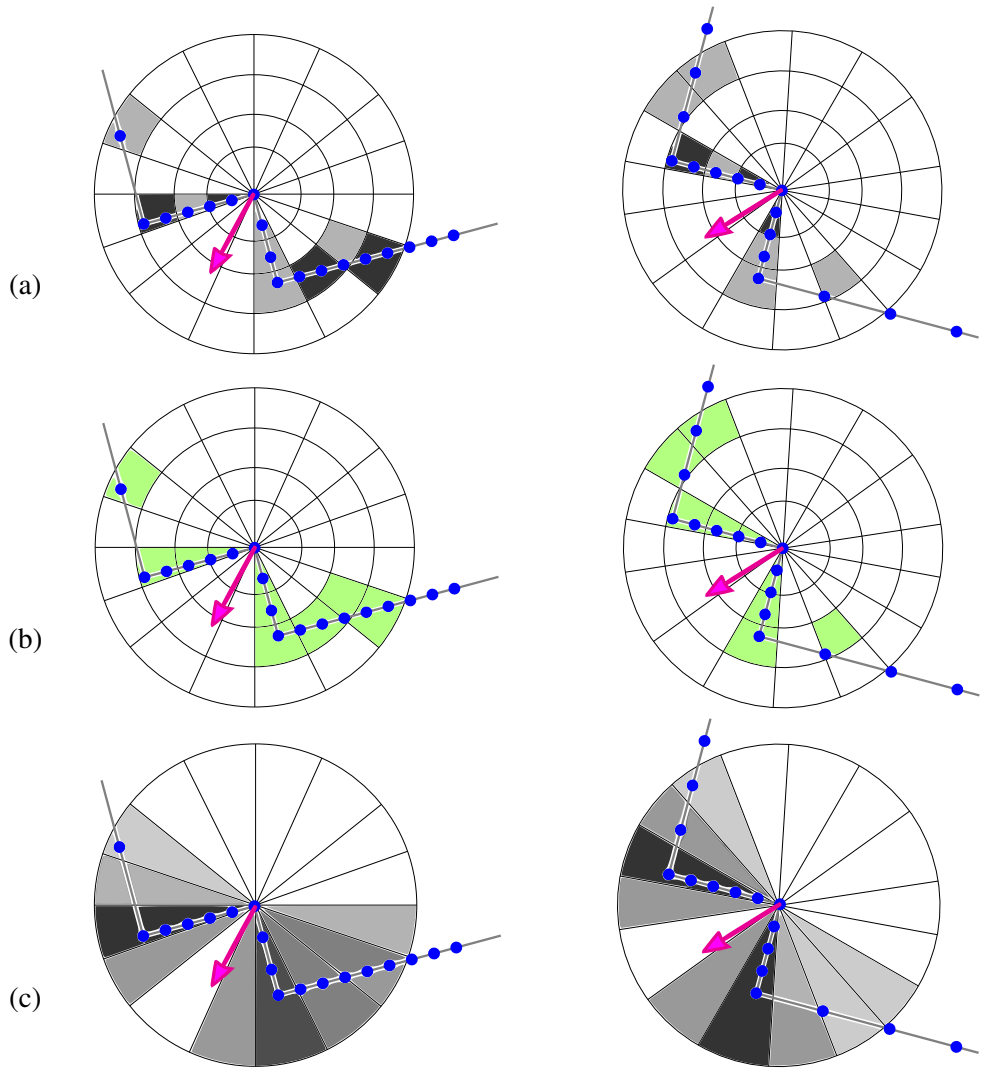


Figure 3.7: Descriptor grids for example data with two different viewpoints and relative *corner orientation vector* (purple arrow): (a) Local Shape Context proposed in [8]; (b) *Binary Shape Context*; (c) *Cumulative Gaussian Histogram*. Darker grey level represents higher value in the histogram bin.

The proposed keypoint detector and descriptors have been compared with the state of the art of 2d laser scan feature: Orthogonal Corner (OC) [11] and FLIRT [8]. The feature performance are assessed on five datasets: *fr-079*, *fr-clinic*, *intel*, *mit-csail* and *victoria-park*. All log files are freely available from Radish and other online repositories[1]. However, the assessment about features requires a groundtruth where all scan viewpoints are referred to a single global reference frame. I used the datasets with the registered scans provided with [8], which consists of both the original scans and the corresponding adjusted pose graph. For each test, a total of 100 random scans, 20 from each dataset, have been evaluated. Results are then averaged over the 100 samples.

The scans in each dataset are used to assess the performance of both the proposed keypoint detector and the descriptors. The experimental assessment is based on the comparison between two similar features sets. The stability of keypoints has been measured in two ways. In the first one, the keypoints have been extracted from all the scans of the datasets and associated together according to the procedure described in [10]. The stable interest points are detected several times. The covariance matrices of the associated keypoints provide a measure of their stability.

The second metric for keypoint stability is the *repeatability* of the points detected from two similar scans. In particular, the sets of keypoint $\mathscr{P}_R$ and $\mathscr{P}_S$ are detected respectively from a given reference scan and from another similar scan. The keypoints appearing in both sets can be found by associating the points, e.g., according to nearest neighbor criterion. Like in [97], the repeatability is defined as the percentage of common keypoints over the smaller of the two keypoint sets, i.e. $|\mathscr{P}_R \cap \mathscr{P}_S| / \min\{|\mathscr{P}_R|, |\mathscr{P}_S|\}$. In my experiments I performed the four different tests proposed in [8] that assess the feature robustness to changing conditions. A given scan is compared with another scan obtained either by changing viewpoint or by transforming the original scan.

1. *Viewpoint*. Each reference scan $\mathscr{S}_{Ri}$ of the dataset is compared with all the scans $\mathscr{S}_{Sj}$ with a viewpose in 1 m range. Then, the set of close scan pairs

[1] For example, `http://kaspar.informatik.uni-freiburg.de/~slamEvaluation/datasets.php`

is partitioned according to their level of similarity. Since the points of a scan geometrically delimit a free region around the laser, the similarity of two scans $\mathscr{S}_{Ri}$ and $\mathscr{S}_{Sj}$ is measured by their *overlap*, which is the ratio between the area of intersection $\mathscr{S}_{Ri} \cap \mathscr{S}_{Si}$ and the area of their union $\mathscr{S}_{Ri} \cup \mathscr{S}_{Sj}$.

2. *Noise*. Each scan $\mathscr{S}_{Sj}$ is obtained by adding increasing level of gaussian noise to the range measurement of the original scan $\mathscr{S}_{Ri}$. The standard deviation of the noise (measured in meters) represents the level of noise.

3. *Oversampling*. The scan pairs used in this test are obtained by adding new points to the original scan through linear interpolation. The oversampling level corresponds to the number of points obtained by interpolating two adjacent points.

4. *Subsampling*. The scan pairs used in this test are obtained by removing points from each scan of the dataset. The subsampling level corresponds to the number of points removed between two remaining points.

Descriptor performance is assessed using the same scan pairs used in the four keypoint detector tests illustrated above. Each pair of scans has been processed as follows. The keypoints sets $\mathscr{P}_R$ and $\mathscr{P}_S$ are detected respectively from the reference scan $\mathscr{S}_R$ and the similar scan $\mathscr{S}_S$ as in the tests discussed above and the corresponding sets of descriptors, respectively $\mathscr{D}_R$ and $\mathscr{D}_S$, are computed. Next, the sets $\mathscr{D}_R$ and $\mathscr{D}_S$ have been associated measuring the distances between descriptors. Two descriptors match if their distance is below a given threshold. Several values of threshold have been used in order to compute an evaluation through precision-recall curves. The ground truth for these tests is obtained with a geometric association of the keypoint used for descriptors computation.

3.3.1 Keypoints repeatability

Evaluation of keypoints repeatability has been performed with default parameters for each dataset. In particular, the parameters used for FLIRT and OC are those indicated

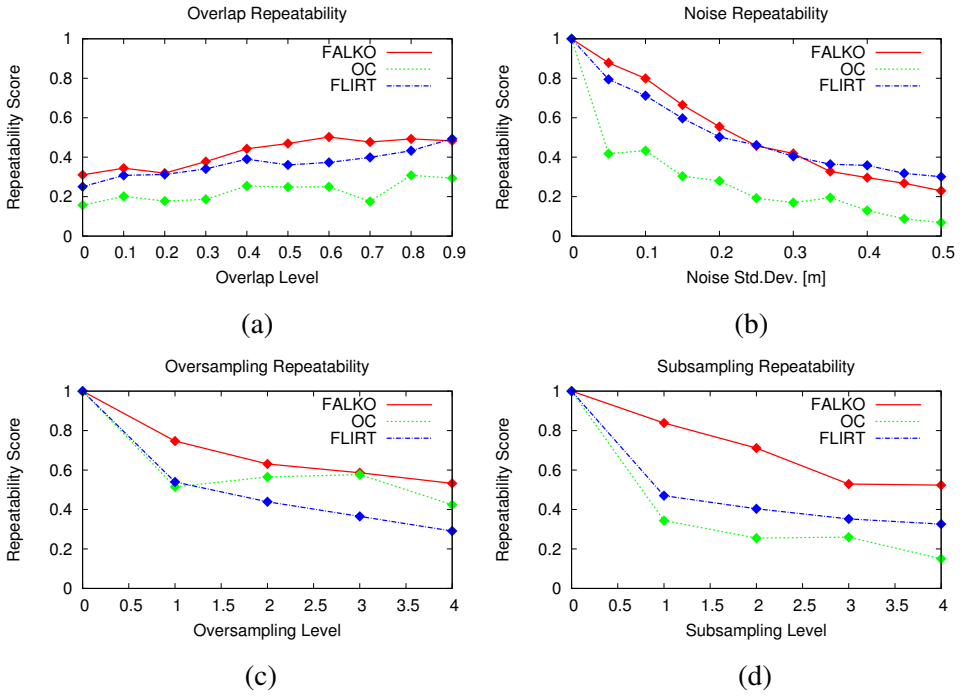


Figure 3.8: Repeatability values of keypoint detectors FALKO, OC and FLIRT in four tests: (a) viewpoint with different scan overlap value; (b) noise level with different noise variance; (c) oversampling of scan measurement; (d) subsampling of scan measurement.

in the respective papers [8]. For FALKO, we set $\beta = 4.0$ and $s_n = 16$. These parameter values have been used with both indoor and outdoor environments for a fair comparison among methods. However, to achieve better performance a fine tuning of the parameters is recommended. Figure 3.8 shows the average results of repeatability tests. In each test, FALKO outperforms both FLIRT and OC, except for high noise deviation levels where, anyway, the scan data are no longer reliable (above 0.3 m). In outdoor datasets, where the scan is more scattered and there are stronger violations of the orthogonal hypothesis, OC results in poor performance, which drops its average repeatability value over the five datasets.

Table 3.1 summarizes the results of the global stability tests of keypoint detectors

Dataset	Keypoint	Num. Points	$\bar{\lambda}_{mean}$	$\bar{\lambda}_{max}$	Single Points
fr079	FALKO	**1127**	0.027	0.039	**21.9**%
	OC	698	**0.022**	**0.032**	30.2%
	FLIRT	1094	0.026	0.037	24.8%
fr-clinic	FALKO	**39077**	0.028	0.040	**49.5**%
	OC	1804	**0.021**	**0.033**	59.1%
	FLIRT	19314	0.024	0.035	52.6%
intel	FALKO	1153	0.028	0.039	**24.8**%
	OC	405	**0.019**	**0.028**	33.1%
	FLIRT	**1155**	0.025	0.035	26.2%
mit-csail	FALKO	**1463**	0.025	0.037	**33.5**%
	OC	716	**0.021**	**0.032**	48.5%
	FLIRT	1446	0.024	0.035	34.4%
victoria-park	FALKO	**9764**	0.027	0.039	**61.3**%
	OC	760	**0.021**	**0.036**	72.5%
	FLIRT	7120	0.026	0.038	70.5%

Table 3.1: Average of the geometric mean and maximum of the keypoint covariance matrix square-rooted eigenvalues. Best values are in bold font.

for all the datasets. As in [10], a global map of keypoints is built for each dataset by adding a new landmark, when a new point is distant at least $d_{min} = 0.2\ m$ from any landmark in current map, and by associating a point to the closest landmark l in the range $d_{max} < 0.05\ m$. The covariance matrix Σ_l is computed using the k_l keypoints associated to the map landmark l, when $k_l > 1$. If $k_l = 1$, l is computed from a single point and is, therefore, an ephemeral landmark. The maximum $\lambda_{max,l}$ and the geometric mean $\lambda_{mean,l}$ of the eigenvalues of Σ_l are used to measure the uncertainty of l. The average of maximum eigenvalues $\bar{\lambda}_{max}$ and of eigenvalue geometric means $\bar{\lambda}_{mean}$ are reported in Table 3.1 with the percentage of landmarks obtained from single points. As can be seen, OC has the lowest values of $\bar{\lambda}_{max}$ and $\bar{\lambda}_{mean}$ in all the datasets, though with the higher percentage of ephemeral points and lower number of detected

keypoints.

FALKO obtains slightly higher values of $\bar{\lambda}_{max}$ and $\bar{\lambda}_{mean}$ than FLIRT, but it tends to detect more stable points as shown by the lowest percentage of single points, i.e. ephemeral landmarks.

3.3.2 Descriptors recognition

Like in section 3.3.1, descriptors have been evaluated with the same parameters for all datasets. FLIRT's descriptor, β-*grid* was set with the default parameters from [8]. For BSC and CGH descriptors we set $r_n = 8$ (only for BSC), $\alpha_n = 16$ and a neighbors radius equal to 0.5 m. For CGH we set standard deviation $\sigma = 0.6$. Figure 3.9 shows the 1-Precision Recall curves computed for the four scan transformations. The results of a pure descriptor-association are poor for all the evaluated descriptors due to the intrinsic lack of information in laser scan data, as shown also in [5, 8]. A pure descriptor-association without position or geometry information is not recommended for pose estimation purpose. Descriptors can be used as a gating rule in geometric associations like Nearest-Neighbors, Hungarian algorithm or RANSAC. BSC and CGH perform almost the same or even slightly better as β-grid in each test, while requiring much less computation. Indeed, table 3.2 shows the registered computation time. Performance is computed over 1000 samples of the same dataset (fr079) on an IntelÂő CoreâĎć i7-4770 CPU @ 3.40GHz, 8GB RAM. Results show that FALKO, BSC and CGH significantly outperform the other methods in computation performance. OC performs better than FLIRT, but is 10 times slower than FALKO due to rotation alignment. Computational times of FALKO and OC are compatible with real-time execution.

3.4 Discussion

In this chapter, the performance of the proposed algorithms has been tested on widely used benchmark datasets and compared with OC and FLIRT, the state-of-the-art features for LIDARs. The results about detection show that FALKO achieves higher

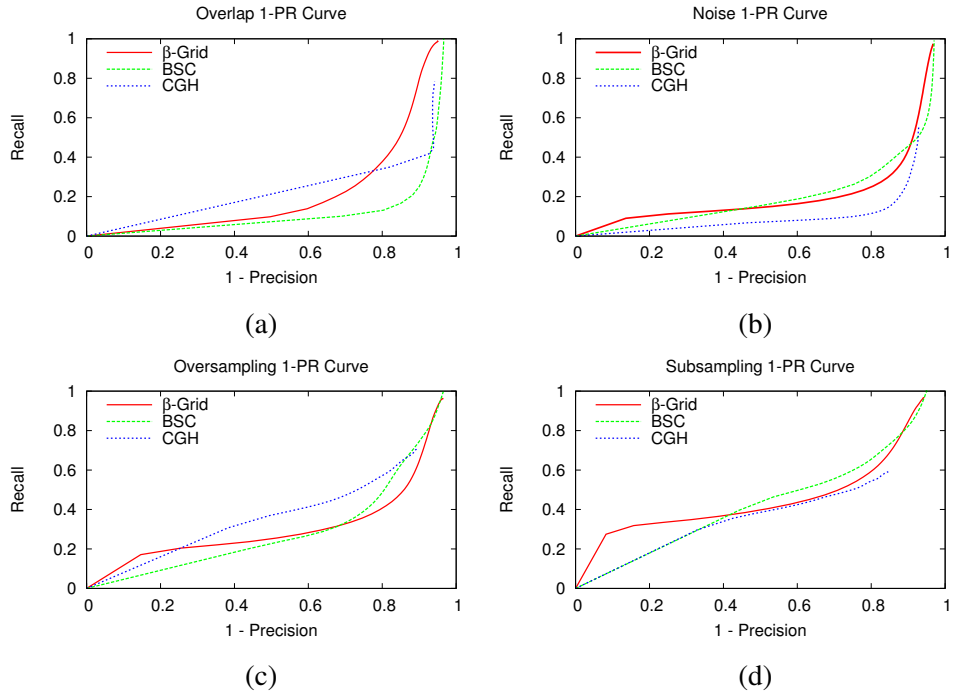


Figure 3.9: 1-Precision Recall curves for the descriptors β-grid, BSC and CGH in four tests: (a) viewpoint (scan overlap value in interval $50\% - 90\%$); (b) noise level; (c) oversampling of scan measurements; (d) subsampling of scan measurements.

repeatably score and extracts less ephemeral points than the other keypoint detectors. Moreover, the precision-recall curves of the proposed detectors are consistent with the achievable results obtained from computer vision and laser scan data descriptors. A complete evaluation on a localization and mapping application of the proposed features is reported in the next chapter.

Keypoint	Time [μs]	Descriptor	Time [μs]
FLIRT	23469	β-grid	1106
OC	5362	BSC	34
FALKO	567	CGH	27

Table 3.2: Average computation time of keypoints and descriptors

Chapter 4

Feature based SLAM

The recognition of distinctive regions and places from sensor measurements is a fundamental capability in robot localization and mapping problems. When the robot comes back to an already visited region after traveling a long path, this operation is commonly named loop closure and is crucial to globally localize the robot and to achieve consistent maps. Beyond place recognition, an additional requirement of loop closure is the computation of the relative transformation between the current robot frame and the matching one.

In this chapter, I present novel loop closure methods based on FALKO keypoints and compare their performance in online localization and mapping applications. The loop closure procedures considered in my experimental assessment are obtained by combining a keypoint detector, a signature-based matcher and a point-to-point association algorithm. FALKO keypoints have been compared with FLIRT as keypoint detector candidate for the proposed loop closure method. The scan signatures include the state-of-the-art Geometrical FLIRT Phrase (GFP) [14], Geometrical Surface Relations (GSR) [15], and GLAROT, a novel version of GLARE (Geometric LAndmark RElations) [16] fully invariant to rigid transformation. The tested point-to-point association algorithms are RANSAC, Correspondence Graph [77] approach and Affine Hough Transform [78]. Experiments illustrate the performance of discussed methods in the context of both offline and online loop closure, and finally even in online map

estimation.

4.1 Loop closure methods

The detection of stable features from laser scans can be exploited to build lightweight maps alternative to traditional occupancy grid maps. While in principle each feature could be used as an independent landmark in a global map, point association becomes more robust when points extracted from a laser scan are collected into local maps. The point-to-point association among the keypoints of consecutive scans can rely on the relative pose estimation between the two frames given by odometry. However, when the robot comes to an already visited region after travelling a long path and closes a loop, the estimation of point correspondences becomes more difficult due to the larger uncertainty of robot pose. Thus, loop closure should exploit invariants like the mutual geometric relations of the keypoints in a local map. In the proposed method, a graph of local maps is used, where each local map is built from a laser scan and the local map frame is the robot frame from which the laser measurement has been acquired. Each local map consists of the stable FALKO keypoints extracted from the corresponding laser scan. To make loop closure efficient, a vector of real values called signature is computed for each keypoint map. I propose GLAROT signature, a variant of GLARE signature [16] that is provided with a novel rotation invariant metric. The loop closure procedure is performed in two steps: selection of candidate loops and keypoint-to-keypoint association. First, candidate loops are selected by comparing the signatures of current local map with the other signatures in the database according to the metric. Second, point-to-point association is performed to validate candidate loops and to compute the transformation between the current and candidate map frames. In the remaining of this section, I illustrate the GLAROT signature and two point-to-point association methods: Correspondence Graph (CG) and Affine Hough Transform (AH).

4.1.1 GLAROT

The Geometric landmark relations (GLARE) [16] algorithm computes a signature that encodes the pairwise distances and angles of a point set. This method only requires the coordinates of the input points and no other information like descriptor values often provided by keypoint features. Given a set of 2D points $\mathscr{P}$, the pairwise angles and distances of two different points $p_i, p_j \in \mathscr{P}$ (assume that $p_{i,y} > p_{j,y}$) are defined respectively as

$$\theta_{ij}^{+} = \text{atan2}(p_{i,y} - p_{j,y}, p_{i,x} - p_{j,x}) \quad \rho_{ij} = \|p_i - p_j\| \tag{4.1}$$

GLARE signature encodes all the above pairwise relations into an accumulator array G similar to the Hough transform one [98]. G consists of $n_\theta \times n_\rho$ cells with an assigned size $\Delta\theta \times \Delta\rho$ and the accumulator $S_g(t,r)$ is associated to each cell, where t and r are the indices of the cell. If the pairwise angle θ_{ij}^{+} and distance ρ_{ij} belong to a cell (t,r), then the corresponding $G_{t,r}$ is incremented. To take into account the point uncertainty, the accumulators of adjacent cells are also incremented by the values obtained by sampling a Gaussian function centered in $(\theta_{ij}^{+}, \rho_{ij})$.

The similarity between a source point set $\mathscr{P}^S$ and a target one $\mathscr{P}^T$ is measured by the distance between the respective signatures G^S and G^T according to L_1 norm [16]. Unfortunately, GLARE signature is invariant only to translation, but not to rotation. To overcome this problem, the GSR signature [15] estimates the normal direction of the implicit surface from which points are sampled by the laser scan. The normal angle is computed by collecting the points of the original scan into an Euclidean grid and by finding the normal distribution of the points lying inside a cell. The novel pairwise angles θ_{ij}^{*} are defined as the differences between normal angles. While rotational invariant, the computation of GSR is expensive due to the adoption of a grid and is less general than GLARE. To make GLARE rotation invariant I propose GLAROT (GLAre ROTation-invariant), a novel procedure for comparing GLARE signatures. A rotation of the point set about an angle β, shifts the values of pairwise angles from θ_{ij}^{+} to $\langle \theta_{ij}^{+} + \beta \rangle_\pi$, where $\langle x \rangle_m$ is the m modulo operator. The angular shifts discretized with cell resolution $\Delta\theta$ are executed as circular shifts of an array. Even though the rotation angle β between two point sets $\mathscr{P}^S$ and $\mathscr{P}^T$ is unknown, it can be recovered by searching for the angular shift that minimizes the distance between the signatures $G^{S,k}$ and G^T, where $G^{S,k}$ is obtained through a circular shift

on the columns of G^S. In particular, the *shifted* L_1 *norm* (SL_1) is defined as

$$SL_1(G^S, G^T) = \min_k \sum_{t=0}^{n_\theta - 1} \sum_{r=0}^{n_\rho - 1} \left| G^T_{t,r} - G^S_{\langle t+k \rangle_{n_\theta}, r} \right| \tag{4.2}$$

The computation of SL_1 requires n_θ evaluations of L_1 norm for $k = 0, \ldots, n_\theta - 1$. In practice, since the standard value of n_θ is 8, such operation is faster than the computation of normals needed by GSR.

4.1.2 Correspondence graph

The signature comparison makes an initial fast selection of candidate local maps for loop closure. However, the validation of such candidates and the estimation of the rigid transformation between local maps require point-to-point associations between point sets. The most effective methods for point correspondences do not depend on the initial estimation of relative pose between the local maps, but compare the internal mutual relations among the points of each set. The distance between point pairs is an example of geometric relation internal to a point set. Correspondences between the points of two sets are considered valid only if their internal relations are compatible.

The *Correspondence Graph* method [77] provides an elegant way to represent both the point set internal relations and the associations using graphs. Let $\mathscr{P}^S$ and $\mathscr{P}^T$ be respectively the source and target point sets. Let $\rho^S_{i_s,j_s}$ be the Euclidean distance between $p^S_{i_s}, p^S_{j_s} \in \mathscr{P}^S$ for $i_s \neq j_s$. All the pairs of $p^S_{i_s}$ and $p^S_{j_s}$ and their distances are collected into set $\mathscr{E}^S$. The feature graph of the source point sets is the complete undirected graph with vertices $\mathscr{P}^S$ and with edges $\mathscr{E}^S$. The feature graph of target point set with vertices $\mathscr{P}^T$ and edges $\mathscr{E}^T$ (within mutual distances $\rho^T_{i_t,j_t}$) is similarly defined.

The correspondence graph is defined as follows. A vertex v_{i_s,i_t} of the correspondence graph represents the association between points $p^S_{i_s} \in \mathscr{P}^S$ and $p^T_{i_t} \in \mathscr{P}^T$. Two vertices v_{i_s,i_t} and v_{j_s,j_t} are connected by an edge *iff* the internal relations of $p^S_{i_s}$ with $p^S_{j_s}$ and of $p^T_{i_t}$ with $p^T_{j_t}$ are compatible. In particular, there is compatibility when their distances are equal up to a tolerance ε, i.e. if $|\rho^S_{i_s,j_s} - \rho^T_{i_s,j_s}| < \varepsilon$. The reliability of point-to-point data association increases with the number of mutually compatible

associations. Thus, the optimal solution is given by the maximum set of mutually compatible associations, i.e. the maximum clique of the correspondence graph.

4.1.3 Affine Hough Transform

Point-to-point association can be exploited using a voting method based on the Generalized Hough Transform [78],[99]. The proposed algorithm uses a generalization of the Hough transform for affine matrices representing 2D rigid transformations. In particular, each element in the Hough space is indexed with a parameter vector $\mathbf{v} = [t_x, t_y, \theta]^\top$ which characterizes the corresponding affine transform. For computational reason, the Hough space is bounded in $[\mathtt{min}_x, \mathtt{max}_x] \times [\mathtt{min}_y, \mathtt{max}_y] \times [\mathtt{min}_\theta, \mathtt{max}_\theta]$ and each parameter is quantized with respectively three steps Δ_x, Δ_y and Δ_θ. Thus, the Hough space is represented by a three dimensional matrix where each element stores the matching pairs subset which is compatible with the corresponding quantized affine transform. As in section 4.1.2, let $\mathscr{P}^S$ and $\mathscr{P}^T$ be respectively the source and target point sets. Let $p_i^S \in \mathscr{P}^\mathscr{S}$ and $p_j^T \in \mathscr{P}^\mathscr{T}$ be two points of the respective sets. For each matching pair (p_i^S, p_j^T), an infinite set of compatible affine transforms can be computed as follow:

$$\left[\begin{array}{c} p_j^T \\ \hline 1 \end{array}\right] = \left[\begin{array}{c|c} \mathscr{R}(\theta_{ij}) & \begin{array}{c} t_{x_{ij}} \\ t_{y_{ij}} \end{array} \\ \hline 0 & 1 \end{array}\right] \left[\begin{array}{c} p_i^S \\ \hline 1 \end{array}\right] \tag{4.3}$$

$$\left[\begin{array}{c} t_{x_{ij}} \\ t_{y_{ij}} \end{array}\right] = p_j^T - \mathscr{R}(\theta_{ij}) p_i^S \tag{4.4}$$

where $\mathscr{R}(\cdot)$ is a 2d rotation matrix. The Hough voting is performed varying θ_{ij} in the range $[\mathtt{min}_\theta, \mathtt{max}_\theta]$ and computing $t_{x_{ij}}$ and $t_{y_{ij}}$. Then, the matching pair (p_i^S, p_j^T) is added in the Hough space element list with indexes $[t_{x_{ij}}, t_{y_{ij}}, \theta_{ij}]$. In particular, each subset of compatible affine transform given by the voting step can be seen as a helix in the three dimensional Hough space (Fig. 4.1). The optimal matching is given by the subset of matching pairs in the affine Hough space element with maximum cardinality.

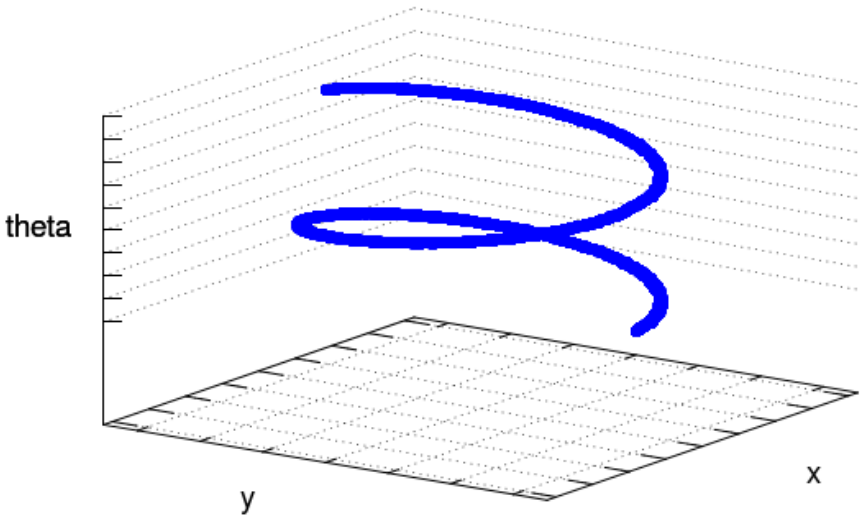


Figure 4.1: An example of helix shape resulting with Hough voting algorithm in Affine Hough Transform.

4.2 Evaluation Results

In this section I present the experimental setup and results based on loop-closure and map estimation. The experiments, are assessed in five public datasets: three indoor datasets (*fr079*, *intel*, *mit-csail*) and two outdoor (*fr-clinic*, *victoria-park*). The datasets, provided by [8], contain both original scans and corresponding corrected ground truth. Results are presented with precision-recall curves for loop-closure tests and with the resulting pose map for online map estimation.

4.2.1 Experimental Setup

The loop-closure experiments are evaluated both comparing scans of the whole dataset (offline mode) and simulating an online map estimation:

- *Offline*. Each scan $\mathscr{S}_R$ of a dataset has been compared to the other candidate scans $\mathscr{S}_i$ with $i \neq R$.

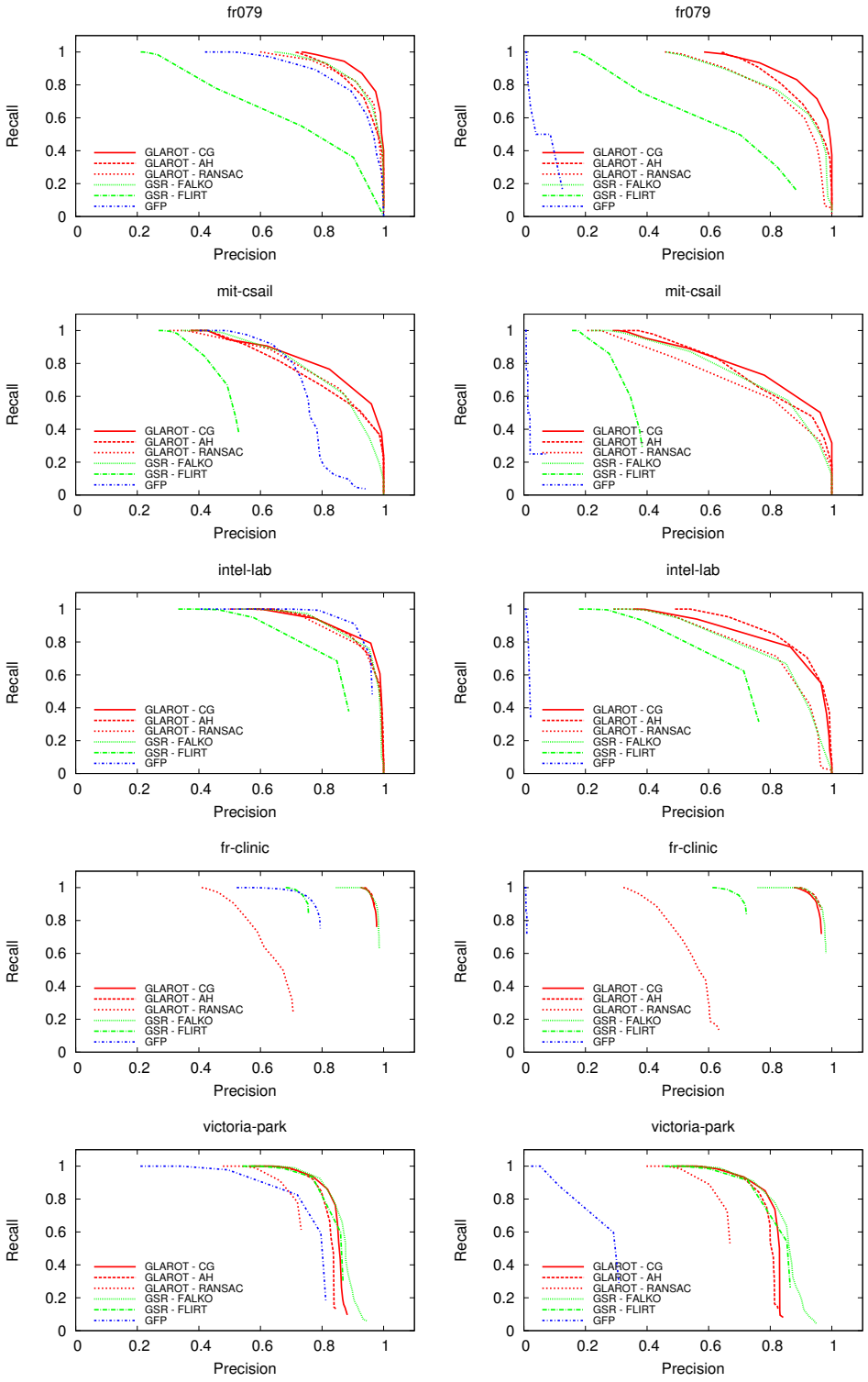


Figure 4.2: Precision-Recall curves for offline (a) and online (b) loop-closure tests.

- *Online*. Each scan $\mathscr{S}_R$ of a dataset has been compared to the other candidate scans $\mathscr{S}_i$ with $i < R$. To avoid trivial matching of consecutive scans I limit the loop-closure evaluation to those scans that have been registered in sufficiently far robot poses. Formally, let $[x_R, y_R, \theta_R]_R^\top$ and $[x_i, y_i, \theta_i]_i^\top$ be respectively the pose where reference scan $\mathscr{S}_R$ and candidate scan $\mathscr{S}_i$ have been registered; the loop-closure is evaluated only if at least one of the following conditions holds: $|x_R - x_i| > x_{th}$, $|y_R - y_i| > y_{th}$ or $|\theta_R - \theta_i| > \theta_{th}$.

The proposed loop-closure algorithms have been tested in each dataset and compared with two state-of-the-art algorithms, GFP and GSR. GSR is rather similar to the original GLARE and its performance has been already proven [15]. Hence, GLARE has not been evaluated in my experiments. First, I evaluate the original GFP and GSR algorithms using FLIRT keypoints and descriptors. Then, FALKO keypoints have been used as primitives in GSR signatures to evaluate how the higher stability of FALKO w.r.t. FLIRT keypoints influences place recognition performance. Finally, FALKO features with the proposed GLARE variant, GLAROT, has been evaluated with three different point-to-point matching methods: RANSAC, *Correspondence Graph* (CG) and *Affine Hough Transform* (AH). GFP and GSR variants use RANSAC algorithm as point-to-point association method.
For each algorithm, both offline and online loop-closure performance as the online global mapping has been evaluated. The loop-closure evaluation has been performed as follows. For each scan $\mathscr{S}$ of a dataset, the scan signatures have been computed and stored. Then, let $\mathscr{S}_R$ be the reference scan, a set $\mathscr{C}_S$ of 10 candidate scans (50 for GFP), whose signatures are closer to the signatures of the reference scan $\mathscr{S}_R$ according to the respective loop-closure method, are extracted. The set of closest signatures $\mathscr{C}_S$ is evaluated according to the corresponding distance function and thresholds of each loop-closure method. The keypoints of each scan in $\mathscr{C}_S$ are associated with $\mathscr{S}_R$ using the selected point-to-point association algorithm and are used to compute robot pose in least-squares sense. Finally, the scan selected for loop-closure $\bar{\mathscr{S}}_i$ is the scan that, after alignment, has the greatest number of keypoints with a neighbor point in $\mathscr{S}_R$ within the 0.10 m range. The results of the loop-closure tests are shown with precision-recall curves. The robot is considered localized if the associated points are

at least N_{min}. The precision-recall curves are computed w.r.t. the threshold N_{min}. A localization is considered correct when the position error of the aligned scan is less than 0.50 m and the angular error less than 10°.
Online map estimation is performed similarly to the loop-closure tests. First, a pose graph is constructed with the odometry poses provided by the datasets. Then, loop-closures have been evaluated as previously presented. In particular, if the selected scan $\bar{\mathscr{S}}_i$ has a number of associated points greater than N_{th} then the loop is considered closed. The final pose graph is then optimized with the state-of-the-art SLAM back-end *g2o*[61].

4.2.2 Loop closure tests evaluation

Loop-closure evaluation has been performed with default parameters for each algorithm. In particular, GFP, GSR and FALKO have been tested with corresponding recommended parameters [11, 14, 15]. RANSAC inlier probability for point-to-point association is set to 0.7. The tolerance ε of Correspondence Graph is set to $0.10m$. For Affine Hough Transform the searching range is set to $[-5m, 5m] \times [-5m, 5m] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$; Δ_x, Δ_y and Δ_θ are set respectively to $0.1m$, $0.1m$ and $0.04rad$. Finally, for online loop-closure evaluation, x_{th}, y_{th} and θ_{th} are set respectively to $0.20m$, $0.20m$ and $0.35rad$.
Figures 4.2 shows the offline and online results for loop-closure experiments with N_{min} varying within the range $[0, 20]$. The top three rows are the results with indoor datasets whereas the last two rows are with outdoor ones. As can be seen, in offline mode (fig. 4.2.(a)) each algorithm performs slightly better than online. This is also possibly due to the avoidance of trivial associations between consecutive scans in online mode. GFP performs drastically different between offline and online. GFP computes the bag-of-words (BoW) using the whole dataset, and the availability of a limited number of phrases, inherent in online operation, causes this performance drop. Due to the higher stability and repeatability of FALKO keypoints w.r.t. FLIRT, GSR-FALKO outperforms GSR-FLIRT in all the datasets, except for *victoria-park* where the absence of corners in the environment leads to the same results for both algorithms. Finally, GLAROT (which is also based on FALKO keypoints) obtains almost

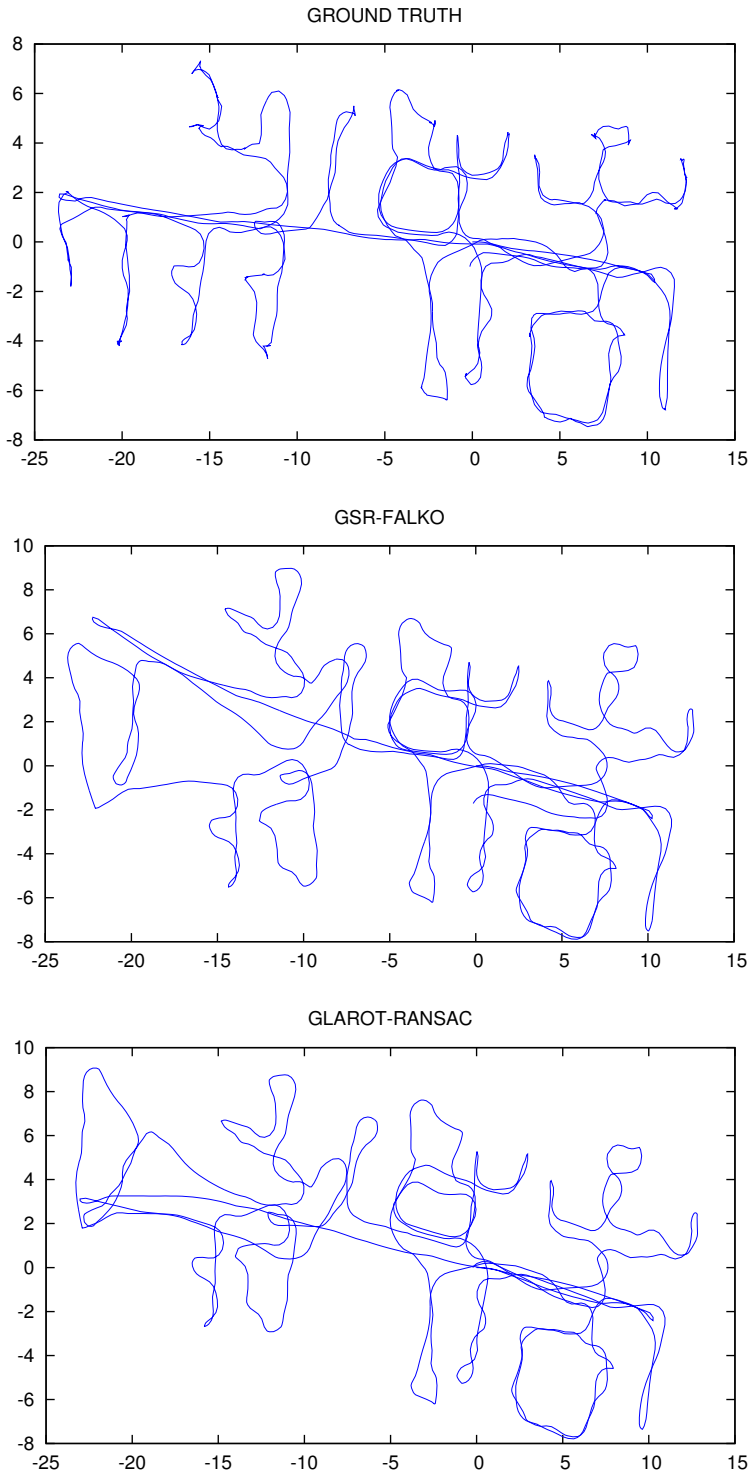
always the best results in all the datasets. In particular, GLAROT-CG and GLAROT-AH slightly outperform GSR-FALKO and GLAROT-RANSAC, with GLAROT-CG obtaining higher precision and recall values, especially in indoor datasets.

4.2.3 Online SLAM tests evaluation

This section illustrates the qualitative results of online map estimation. Except for *fr079*, figure 4.2.(b) shows the persistence of false positives in the loop closure results due to the limited number of revisited places. Even a single false loop detection drastically affects the final map reconstruction. Hence, the online SLAM results are shown only for *fr079* which provides reliable loop closure detection. Figure 4.3.(a) shows the correct ground truth [8], while the other images in figures 4.3.(b)-(c) and 4.4.(d)-(e) represent the map computed with the proposed loop-closure methods: GSR-FALKO, GLAROT-RANSAC, GLAROT-AH, and GLAROT-CG. Each algorithm has been evaluated choosing N_{th} as the threshold generating the break even point in the corresponding precision-recall curve. Due to the persistent presence of false positives in the loop-closure results, maps generated by GFP and GSR-FLIRT are not shown. As can be seen in figures 4.3.(b)-(c) and 4.4.(d)-(e), each method provides a valid optimization of the right side of the map, whereas in the left side the differences between algorithms are more pronounced. In particular, GLAROT-CG achieves the best global map estimation, also in the leftmost part of the map where loop closures are less frequent.

4.2.4 Loop Closure Efficiency

The efficiency of the algorithms has been empirically assessed by performing loop closure on on dataset *fr079*. The average execution times obtained with Intel i7-3630QM CPU @ 2.40GHz, 8 GB RAM are illustrated in Table 4.1. Such values are affected by several parameters, including the total number of scans, the cardinality of the candidate set $\mathscr{C}_S$, the number of keypoints extracted from each scan. I report the average execution time per scan. The initialization of GFP takes significantly more time than the initialization of geometric signatures GLAROT and GSR. GFP is an

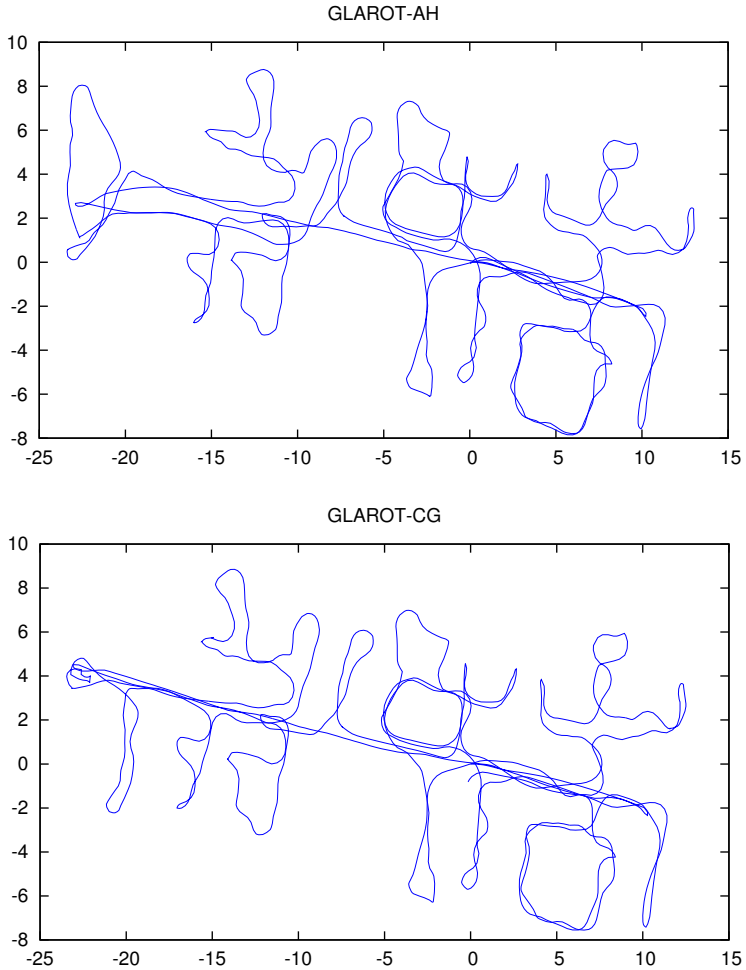


(a)

(b)

(c)

Figure 4.3: Online SLAM pose map results for five algorithms: (a) ground truth provided by [8]; (b) GSR signatures using FALKO keypoints; (c) GLAROT with original RANSAC for point-to-point matching.



(d)

(e)

Figure 4.4: Online SLAM pose map results for five algorithms: (d) GLAROT with Affine Hough Transform; (e) GLAROT with Correspondence Graph.

operation		# candidate	avg time [ms]
GFP	creation	-	2.702
	association	50	2.376
GSR	creation	-	0.123
	association	10	36.204
GLAROT	creation	-	0.123
	association	10	12.154
CG		10	16.314
AH		10	356.927
RANSAC		10	49.430

Table 4.1: Average times per scan required for creation, signature and point-to-point association using different methods in offline experiments with dataset fr079.

intrinsically offline method that transfers most of the loop closure complexity to the computation of BoW histograms from the whole scan dataset. On the other hand, geometric signatures can be efficiently computed, since they only depend on the keypoints detected in a given scan. The complexity of closest signatures search depends on the number of scans in the set and is less efficient, but comparable with GFP association. Point-to-point association is the most computationally demanding step of loop closure. CG has proven more efficient than AH and RANSAC.

4.3 Discussion

In this chapter, novel loop closure methods based on FALKO keypoints have been illustrated and their performance have been compared in both offline and online localization and mapping problems. The FALKO keypoints detected from each laser scan become part of a corresponding local map. To find loops, the current local map is matched with the other local maps in two steps. The candidate loops are found by comparing signatures and, then, point-to-point association is applied to match

individual keypoints and to compute the accurate rigid transformation between local maps. The novel signature GLAROT has also been proposed and compared with state-of-the-art signature algorithms. Moreover, several point-to-point data association techniques have been evaluated. Results show that the stability of FALKO keypoints has proven crucial for loop closure. FALKO detector combined with GLAROT signature and point-to-point association outperforms the other approaches.

Chapter 5

Industrial Application

Localization and mapping enable autonomous robots to safely and efficiently navigate in environments. In general, this capability is explicitly acknowledged in service robotic applications with strong human robot interaction whereas it is neglected for industrial context. A typical industrial scenario consists of assembly lines where products are transported through known paths and positions by robot manipulators and other mechanical devices. In last decades, autonomous ground vehicles (AGVs) for warehouse logistics and advanced sensing technology are common occurrence and have boosted demand for increased flexibility and robot autonomy. Accurate estimation of position and orientation is the most important requirement of robot localization in industrial applications. It is usually satisfied by structuring warehouses with artificial landmarks, e.g. example reflective markers that are easily detectable by laser scanners. Landmark-based navigation with laser sensors has become a popular and effective solution for industrial AGVs. Unfortunately, it is not always possible or convenient to completely structure industrial environments due to space limitations, mutable configurations and unpredictable conditions in specific regions of operative space. The methods described in the previous chapters of this thesis can effectively address these problems. In this chapter, I will present how different approaches like improved odometry and feature detection in raw sensor measurement can be combined to achieve accurate robot navigation in unstructured industrial environments.

Figure 5.1: AGV used in calibration tests provided by Elettric80 s.p.a.

In particular, the automatic AGV calibration method results will be discussed using several comparison criteria. Later, the localization method based on LIDAR features will be described and compared to existing localization systems.

5.1 Application Scenario: Industrial Warehouse Logistics

AGVs are commonly used to automatically and efficiently manage warehouse logistics. These vehicles are used to transport products packed into standard format pallets from assembly lines to storage and delivery areas of plant according to the requirements of production. Figures 2.1 and 5.1 show two examples of AGVs manufactured by Elettric80 S.p.A. On their rear they are equipped with a mechanical forklift, which can be raised or lowered according to pick or drop the pallets. Storage areas are arranged differently according to the specific plant. Common examples include simple stacks of pallets called block storage (see Figure 5.2) or specific metallic racks like in Figure 5.3. Pallets may be raised to storage position at significant height, which requires accurate positions of AGVs and their forks.

AGVs are usually wheeled mobile robots with tricycle kinematic configuration

Figure 5.2: Block storage example.

described in chapter 2. The AGV front with the actuated wheel is at the opposite side of the fork-lift. The robot is equipped with four laser scanners for safety (Sick S3000 and S300) with a scanning plane close to the ground plane, and one laser scanner for navigation (Sick NAV350). The navigation laser scanner allows the detection of artificial reflective markers in the environments that are used as landmarks for localization and egomotion estimation.

Mission planner assigns transportation tasks to each AGV according to plant requirements provided by Enterprise Resource Planning (ERP). Path planner provides the specific path to each AGV though composition of pre-defined path segments. Segments paths are designed a priori by human operators during plant setup and arranged into a path layout. An example of simple layout is shown in Figure 5.5. Adoption of segment layout has several advantages. It simplifies path planning and resource allocation. It guarantee control on all possible AGVs configurations. It enables specific assignments for each path segment (laser safety areas, etc.). However, it requires accurate localization to monitor navigation since AGVs cannot perform arbitrary task

Figure 5.3: Drive-in example.

according to sensor perception in reactive-like fashion.

5.2 AGV Calibration results

The proposed calibration algorithm has been implemented and tested on industrial AGVs in a warehouse building. Two vehicle models have been used in the experiments for data collection and performance analysis: the CB16 and the CB25 (manufactured by Elettric80 S.p.A.), both compliant with the tricycle model illustrated in section 2.1. The choice of CB16 or CB25 for the experiments discussed in this sec-

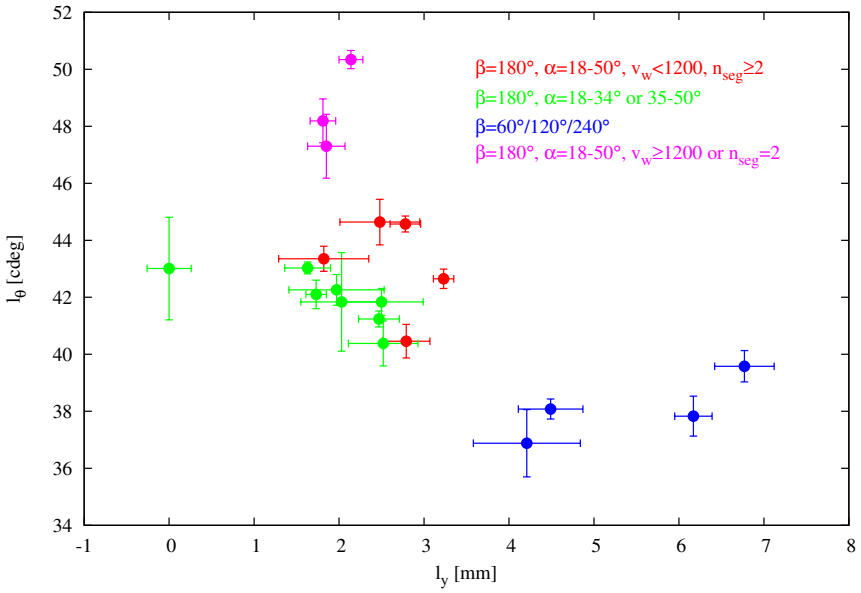


Figure 5.4: STC calibration parameters l_y and l_θ with standard deviation error bars obtained at different conditions. Color labels highlight values obtained at specific conditions: standard $\beta = 180^\circ$ segment paths, large range of steering angle α and moderate speed (red); $\beta = 180^\circ$ with limited α range (green); $\beta \neq 180^\circ$ segments (blue); standard conditions, but with $n_{seg} = 2$ or higher speed (purple).

tion is related to the AGV availability in the specific setting, where the experiment has been performed. The CB25 is shown in Figure 2.1. The reflective markers, which are commonly placed in industrial warehouses, correspond to stable and easily distinguishable points in a laser scan. Thus, the association of corresponding landmarks between two scans is reliable and is substantially free of outliers.

The robot egomotion is estimated exploiting the landmarks detected by the navigation range finder. The landmarks observed from different scans are associated and aligned using a registration algorithm similar to [34, 100]. Theoretically, it should be sufficient to compare the landmark sets at the beginning and the end of the trajectory

in order to compute the AGV motion along a segment. In practice, this operation is repeated for each pair of consecutive scans to track landmark associations and to avoid data association errors. Markers are matched according to the nearest neighbor criterion. When the landmark map is available, it can be used to further attenuate the uncertainty associated with sensor measurements. Moreover, the AGV is not moving during the acquisition of the initial and final landmark sets to avoid position errors on the detected landmarks due to the robot motion. Registration with landmarks is much more robust and accurate than generic scan matching methods such as [17] and substantially free of outliers.

The proposed calibration methods for STC and ATC (see chapter 2) require forward and backward robot motions along circular path segments as illustrated in section 2.2. In the specific case of ATC, the forward and backward motions are treated separately to compute the forward and backward steer offsets. Moreover, the AGV follows trajectories with different steering values and directions, i.e. left and right steering angles (respectively positive and negative). The setup required for AGV calibration is a free area in the warehouse large enough to accommodate circular trajectories with maximum radius of 5 m.

Two experiments have been designed to assess the correctness and the precision of the proposed calibration method. The goal of the first experiment is the assessment of the six calibration parameters at different calibration conditions. The second experiment estimates the positioning precision of different AGVs at operation points.

5.2.1 Calibration Condition Experiment

Since the groundtruth parameters are not available, the measurement of the accuracy of the calibration methods can be indirectly performed by assessing the repeatability of the estimation. Therefore, the calibration procedure has been repeated several times using different sets of circular path segments with the AGV moving at different speeds. A CB16 AGV has been calibrated in a warehouse environment with enough space for the required motion. The calibration is organized in four phases each distinguished by the motion direction (forward or backward) and by the steering side (left or right). During each phase, the AGV executes circular path segments and gradually

increases the (absolute) value of steering angle α. The curvature radius is equal to $r_{lp} = L/\tan\bar{\alpha}$ at the logical point and $r_w = L/\sin\bar{\alpha}$ at the front wheel. Hence, the radius of circular path segments r_{lp} decreases during a single phase. In order to keep a regular pattern, path segments are circular arcs of fixed angular length β. Hence, the length of a path segment is equal to βr_{lp}, where β is in radians and r_{lp} depends on steering angle α as above. For example, if $\beta = \pi$, the robot covers half-circles (henceafter, β is expressed in degrees for reader convenience). In the different trials, we changed the number of path segments n_{seg} used in each phase, the minimum and maximum steering angles α, the angular length of each path segment β, and the speed of the actuated wheel v_w. The complete calibration procedure takes about $8 \div 15$ minutes depending on these parameters and, in particular, on the number of path segments n_{seg}.

The calibration procedure has been performed three or four times for each configuration (n_{seg}, α-min/max, β and v_w) in order to estimate the average value and standard deviation of calibration parameters. Tables 5.1 and 5.2 illustrate the results achieved in 63 calibrations for a CB16 AGV according respectively to the STC and ATC models. Given the same calibration setup, the calibration parameters computed according to STC and ATC problems are very close. The value of the single steer offset α_{off} for STC has an intermediate value between α_F and α_B for ATC. Even with different calibration conditions and with different calibration problems, the computed calibration parameters do not change significantly across the different trials.

The sensor orientation l_θ is the angular parameter that is most sensitive to experimental conditions with differences slight above 10 *cdeg* (1 $cdeg = 0.01^\circ$), while l_y is the most sensitive position parameter with differences of about 6 *mm* (varying from 0.0 to 6.77). The laser scanner cross-sectional coordinate l_y is less steady than the longitudinal one l_x as shown by their respective average value columns. Figure 5.4 graphically displays the values of l_y and l_θ at different conditions of STC calibration. These two sensitive parameters are more affected by the choice of β, which determines both the length of the path segments and the visible landmarks, by a too low number of segments ($n_{seg} < 4$) or by a high speed during the AGV motion. If a low number of path segments is used (row $n_{seg} = 2$ in Table 5.2), the standard deviation of

α_F is slightly higher and the estimated l_θ is different from the other assessments. The results (in particular, see the values of l_θ in Tables 5.1 and 5.2) show that $\beta = 180^\circ$ is the best trade-off that allows sufficiently long trajectories and observation of the landmarks from a similar angle at the beginning and the end of a path segment. In practice, $n_{seg} = 6 \div 8$ has been considered adequate for application purposes and has been adopted in the proposed procedure. With $n_{seg} = 8$ and $\beta = 180^\circ$, complete AGV calibration requires about 12 minutes.

5.2.2 Position Precision Experiment

Several tasks performed by AGVs require to stop at given points of the warehouse, e.g. to load or unload pallets. The position precision at such operation points depends on different factors including the robot control and navigation system, the pose estimation of the AGV and the calibration parameters. The localization system provides feedback to the control and navigation system in the form of AGV pose w.r.t. an inertial reference frame in the environment. The system computes the motion commands of the AGV in order to reach the operation point. Both the localization and the control-navigation systems exploit the AGV model with the calibration parameters. The intrinsic parameters are part of the robot model and influence both the control system and the prediction of the localizer. The extrinsic parameters define the pose of the navigation laser scanner. An important aim of calibration is to make all the AGVs working in a warehouse stop at the same operation points with adequate precision.

A set of experiments has been executed in two real warehouses, hereafter termed *Setting* 1 and *Setting* 2, to assess AGV positioning precision obtained by automatic calibration. The CB25 AGVs operating in these two warehouses were available for a limited time for our calibration tests. Hence, we focused on assessing position precision with ATC calibration. In each environment, a set of operation points has been selected. In particular, we have chosen 5 points for *Setting* 1 (labeled with letters A-E) and 10 points for *Setting* 2 (labeled as $P_1 - P_{10}$). Figure 5.5 shows the path layout of *Setting* 1 with the selected operation points for the purpose of experimentation. Observe that points A and B (purposely defined for this experiment) are reached by uneven and high curvature paths that stress the control system, whereas the other

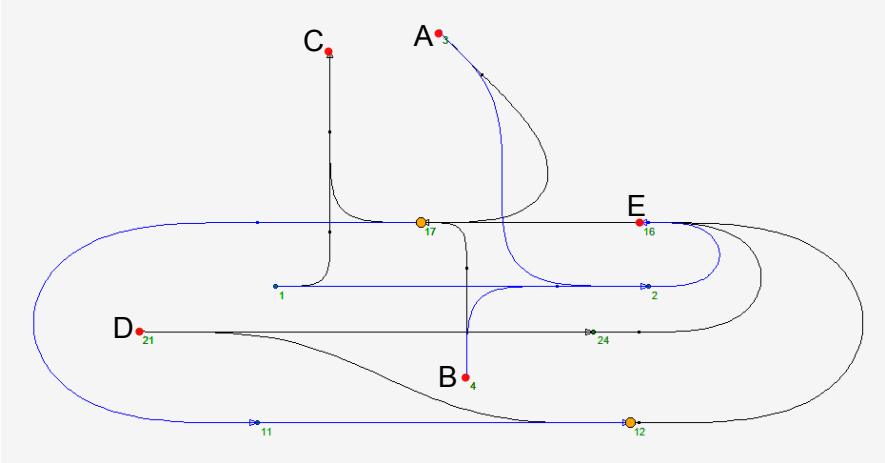


Figure 5.5: *Setting* 1: trajectory layout and operation points for position precision experiments. Points are labelled with letters A, B, C, D, E. The layout area is approximately 9.3 $m \times 4.2$ m.

points (C, D and E) are approachable through smoother trajectories. *Setting* 2 refers to an industrial plant with larger size. In this case, the paths to the operation points are compliant with the requirements of a production warehouse and therefore do not include high curvature segments. Paths consist of both straight and curved segments.

The experiments in *Setting* 1 have been performed using three CB25-type AGVs, labelled LGV71, LGV74 and LGV01. In *Setting* 2 four CB25-AGVs, labelled R32, R30, R31 and R36, have been used. All the AGVs are calibrated using the proposed method with ATC model, with the only exception of R32 which has been calibrated according to a manual procedure by an expert operator. The manual procedure has been executed by separately tuning each calibration parameter. The parameter value is iteratively adjusted by observing the corresponding AGV motion (e.g. α_F/α_B is changed until straight line is obtained, s_w until the travelled distance corresponds to the expected one, etc.). The estimated calibration parameters of all the AGVs are reported in Table 5.3. *Setting* 1 tests enable assessment of the precision at the operation point achieved with the proposed calibration method. In *Setting* 2 tests, the manu-

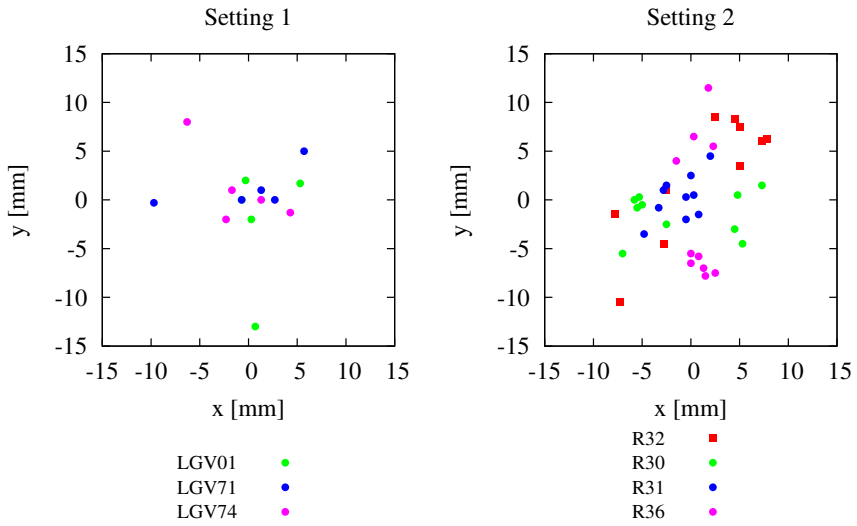


Figure 5.6: Distribution of halting points around the local reference frames of *Setting* 1 (left) and *Setting* 2 (right). The figure is obtained by overlapping points referred to different local reference frames.

ally calibrated robot can be used as a reference for the experiments and enables a comparison with the automatic calibration method.

The operating point coordinates are referred to an inertial reference frame in the environment, but it is difficult to compare each halting point of the AGVs with the nominal operating point. Thus, a marking board has been fixed on the floor near each operation point, oriented approximately according to the expected robot orientation. The halting point of each AGV has been measured according to the local reference frame of the marking board. The origin of the local reference frame is placed in the mean point of the halting points reached by all the AGVs. Tables 5.4 and 5.5 illustrate the coordinates of the halting points and their distance to the local frame origin respectively for *Setting* 1 and *Setting* 2. Figure 5.6 is obtained by overlapping

all the AGV halting points expressed w.r.t. their local frame for both *Setting* 1 and *Setting* 2.

In the experiments performed in *Setting* 1, points A and B are reached by uneven and high curvature paths that stress the control system, whereas the other points (C, D and E) are approachable through smoother trajectories. The distance of AGV halting positions from the local reference origin is larger for operating points A and B (up to 10 *mm*). This result was expected since the paths reaching these two points tend to stress the control system. Indeed, halting points immediately after such high curvature path segments are never adopted in actual industrial plants. On the other hand, the difference among the halting points of AGVs in C, D and E is about 5 *mm* in the worst case and 2 *mm* on the average (Table 5.4). The global average distance to the mean point for all five operation points is less than 5 mm.

The results obtained in *Setting* 2 enable comparison of the halting positions of a manually calibrated AGV (R32) and three AGVs calibrated with the proposed approach (R30, R31, R36). The average distance between the halting points and the local reference frame origin is larger for R32 than the other vehicles, as shown in Table 5.5. In particular, the average distance to the origin of R32 is about 8 *mm*, whereas the other AGVs obtain values close to 6 *mm*. The slightly different distribution of halting points of R32 w.r.t. R30, R31 and R36 is more apparent from Figure 5.6, where it can be observed that halting points of R32 are distributed along all directions. Thus, the proposed automated calibration method obtains a worst-case AGV localization accuracy comparable or better than the accuracy obtained by the best manual calibration, which is assumed to guarantee a position error of about 10 *mm* in industrial practice.

5.3 AGV Localization

AGV localization in industrial environment requires special capabilities and faces constraints which usually are not present in approaches exploited in the state of the art. Indeed, state of the art approaches usually requires high computational power and memory consumption. AGVs are often equipped with industrial PCs or PLCs

which cannot guaranty enough computational power to execute algorithms like Graph SLAM or particle filters. Therefore, algorithm like EKF are typically adopted in industrial AGV localization where the most important factors are robustness and repeatability, rather than large scale mapping capability or loop closure detection.

AGVs provided by Elettric80 S.p.A. localize using an EKF algorithm with a fixed map of landmarks. In particular, the state of the robot is given by

$$q_k = [q_x(k),\ q_y(k),\ q_\theta(k)]^\top \tag{5.1}$$

Where $q_x(k)$ and $q_y(k)$ are the position of the logical point of the robot on the xy plane and $q_\theta(k)$ is the robot orientation at step k. Given the AGV tricycle kinematic model

$$\dot{q} = \begin{bmatrix} v_w\ \cos(\alpha_w)\ \cos(q_\theta) \\ v_w\ \cos(\alpha_w)\ \sin(q_\theta) \\ \dfrac{v_w\ \sin(\alpha_w)}{L} \end{bmatrix} \tag{5.2}$$

where v_w is the linear wheel speed, α_w the wheel steer angle and L the AGV wheelbase, and given the discrete EKF equations

$$\overline{q_k} = g(q_{k-1}, u_k) + \varepsilon_k \tag{5.3}$$

$$\overline{\Sigma_k} = G_k\ \Sigma_{k-1}\ G_k^\top + R_k \tag{5.4}$$

$$K_k = \overline{\Sigma_k} H_k^\top (H_k \overline{\Sigma_k} H_k^\top + Q_k)^{-1} \tag{5.5}$$

$$q_k = \overline{q_k} + K_k(z_k - h(\overline{q_k})) \tag{5.6}$$

$$\Sigma_k = (I - K_k H_k)\overline{\Sigma_k} \tag{5.7}$$

where $g(\cdot)$ and $h(\cdot)$ are non linear function describing respectively system evolution and sensor model, G_k and H_k the associated jacobians, Σ_k the state covariance ad step k and z_k the observations vector. Let define system evolution function $g(\cdot)$ for AGV tricycle localization as the second order Runge-Kutta integration of tricycle kinematics

$$g(q_{k-1}, u_k) = \begin{bmatrix} q_x(k-1) + LS_L \cos\left[q_\theta(k-1) + \dfrac{S_L}{2}\sin(\alpha_{34})\right]\cos(\alpha_2) \\ q_y(k-1) + LS_L \sin\left[q_\theta(k-1) + \dfrac{S_L}{2}\sin(\alpha_{34})\right]\cos(\alpha_2) \\ q_\theta(k-1) + S_L \sin(\alpha_2) \end{bmatrix} \tag{5.8}$$

where

$$S_L = \frac{\Delta t \; v_k}{L} \tag{5.9}$$

$$\alpha_2 = \frac{\alpha_k + \alpha_{k-1}}{2} \tag{5.10}$$

$$\alpha_{34} = \frac{\alpha_k + 3 \; \alpha_{k-1}}{4}. \tag{5.11}$$

$$\tag{5.12}$$

The subscript w has been omitted for simplicity. The corresponding Jacobian is

$$G_k = \frac{\delta g}{\delta q_{k-1}} = \begin{bmatrix} 1 & 0 & -LS_L \sin\left[q_\theta(k-1) + \frac{S_L}{2} \sin\left(\alpha_{34}\right)\right] \cos\left(\alpha_2\right) \\ 0 & 1 & LS_L \cos\left[q_\theta(k-1) + \frac{S_L}{2} \sin\left(\alpha_{34}\right)\right] \cos\left(\alpha_2\right) \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.13}$$

The prediction step of EKF filter is based on odometry measurements and therefore sensor calibration widely affects the results of this process. Since the odometry is based on numerical integration of encoders data, state estimation is affected by drift error. The EKF correction step with a *range-and-bearing* measurement model is used to bound the error and correct the robot position. Navigation laser scanner like the one used by Elettric80 AGVs (Sick NAV350) can detect cylindrical artificial markers equipped with a high reflective surface. The laser scanner processes geometrical points corresponding to the marker and refines the center position using the reflectance profile of the points. The extracted points provide high accuracy, repeatability and robustness. Figure 5.7 shows and example of markers detection with a still AGV positioned $1m$ far from the reflector. As can be seen, the standard deviation in position detection of the artificial marker is on the order of one centimeter. Therefore, the cylindrical reflective markers can be used as landmarks for EKF correction step with centimeter accuracy. To obtain the maximum accuracy in robot localization, the reflectors set is manually mapped by an operator and then stored in the AGVs memory. The observation model used in EKF localization is then

$$h(q_k)^i = \begin{bmatrix} \rho_k^i \\ \phi_k^i \end{bmatrix} = \begin{bmatrix} [(l_{x_k} - q_x(k))^2 + (l_{y_k} - q_y(k))^2]^{\frac{1}{2}} \\ atan2(l_{y_k} - q_y(k), l_{x_k} - q_x(k)) - q_\theta(k) \end{bmatrix} \tag{5.14}$$

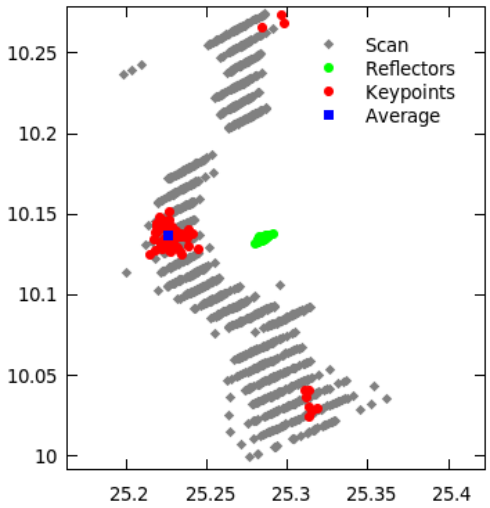


Figure 5.7: Example of reflectors and keypoints extracted from 60 scans with a still AGV. Grey points are original scan points, green points are reflector detection while red points are FALKO keypoint extraction results. The blue point represents the average of the keypoint cluster.

and the corresponding Jacobian

$$H_k = \frac{\delta h}{\delta q_k} = \begin{bmatrix} H_k^0 \\ H_k^1 \\ \vdots \\ H_k^n \end{bmatrix} \tag{5.15}$$

$$H_k^i = \begin{bmatrix} \frac{q_x(k) - l_{x_k}^i}{\rho_k^i} & \frac{q_y(k) - l_{y_k}^i}{\rho_k^i} & 0 \\ \frac{l_{y_k}^i - q_y(k)}{(\rho_k^i)^2} & \frac{q_x(k) - l_{x_k}^i}{(\rho_k^i)^2} & -1 \end{bmatrix} \tag{5.16}$$

where $l_k^i = (l_{x_k}^i, l_{y_k}^i)$ is the i-th landmarks in the EKF map at k-th step. The presented range-and-bearing observation model can be substituted by a two dimensional cartesian observation model without loss of performance or generalization.

When enough reflectors are simultaneously detected, the presented model provides high repeatability in robot positioning allowing a precise and safe navigation of the AGVs in the warehouse. In particular situations like narrow drive-in or block storage, where the number of detected landmarks is low, other localization systems are used (e.g. lateral sonar sensor for *stay-in-the-middle* navigation).

5.4 FALKO-EKF Localization

In this section, I will describe the integration of the proposed FALKO features in an AGV localization system. The EKF localization algorithm presented in section 5.3 uses two dimensional points extracted from reflectors detection as landmark observation in the filter correction step. Since FALKO keypoints are described as two dimensional cartesian points, the integration in the existing localization system can be easily achieved just replacing reflectors observation with extracted features and updating an environmental map of the pre-extracted features. Simultaneous localization and mapping with EKF, requires state augmentation [48]. State augmentation is the correct and standard procedure to add to the map new elements that are discovered during exploration of environment. Augmented-state EKF (AEKF) requires that each observed landmark is added to the filter state. Thus, the state dimension is directly proportional to the number of landmarks observed during operation, resulting in high memory requirements and expensive computation for high-dimensional matrix inversions. Moreover, association between observations and map landmarks will become one of the bottle-neck due to the high number of matching required. Unfortunately, the implementation and onboard execution of AEKF is not compatible or, at last, difficult for AGVs equipped with embedded system with limited computational capabilities and strong software architecture constraints. Indeed, standard industrial AGVs embedded CPU are often adopted to guarantee real-time execution and commitment to safety regulations. Therefore, a dynamic map not managed as EKF state is a convenient trade-off between dynamic map increment and computational constraints. This solutions can be adopted to effectively address shot-term localization an mapping, but not for large-scale mapping. In this context, data association algo-

rithm can also be simplified. For example, the FALKO features extracted from laser measurements can be matched with landmarks using only keypoint geometric information without the descriptors presented in chapter 3. Moreover, no explicit loop closure procedure is necessary.

5.4.1 External dynamic map

Landmark maps can be either fixed or dynamic. The first solution requires a feature mapping of environment, extraction and map insertion of each detected feature with high accuracy localization. Usually, these maps are used in immutable scenarios where landmark configuration do not change. Instead, dynamic maps are constructed and updated during operation, saving each detected feature in the corresponding estimated position at given time. Industrial scenarios like those described in section 5.1 often cannot be managed as fixed maps for long time intervals. Pallets are usually moved and, thus, dynamic maps are the only reliable solutions. Moreover, the limited computational and memory capabilities of AGV embedded CPU also constrain the size of the map. Thus, I propose to manage the map as a dynamic structure that can store that can store a maximum number of landmarks. the map is adapted at each EKF iteration. Each landmark in the map is given a *hit* count describing how many times the feature has been updated. The hit count is then decremented each time the feature is not updated leading to a deletion of the feature when the hit count reachess zero. Each new feature not present in the map is added while an updated feature is averaged with existing landmark through an update rate threshold. Algorithm 1 illustrates the map update procedure.

The proposed external map management provides a lightweight solution to AEKF. However, this kind of approach is decoupled from the state estimation and can lead in drifting of feature position. Moreover, the map do not contains information about landmark covariance.

Algorithm 1: External Dynamic Map Update.

Require:

F: Extracted features vector
L: Landmarks vector
M: Matches between F and L vector
h: Hit count vector
ur: update rate threshold

for $F_i \in F$ **do**
  if $isAssociated(F_i, L, M)$ **then**
    $j \leftarrow exctractLIndex(F_i, L, M)$
    $updateLandmark(F_i, L_j, ur)$
    $h_j \leftarrow h_j + 1$
  else
    $insertNewFeature(F_i, L)$
  end if
end for
for $L_j \in L$ **do**
  if $h_j == 0$ **then**
    $deleteLandmark(L_J)$
  else
    if $hasNotBeenUpdated(L_j)$ **then**
      $h_j \leftarrow h_j - 1$
    end if
  end if
end for

5.5 Dataset protocols and acquisition

For the localization results several datasets in industrial environment have been acquired. A communication protocol between AGVs and acquiring PC has been firstly defined. The protocol specifies the data acquired by sensors and used in the existing localization system as well as additional information like raw scan, timestamps, AGV parameters and others.

Figure 5.8 shows the main environment used in the experiments. The scenario is

Figure 5.8: Block storage simulation.

a simulation of a block storage warehouse where pallets have been placed to form a corridor. Several additional reflectors have been used to guarantee a certain quality in localization. The existing localization system is used to allow the AGV to navigate through the layout paths and to provide comparison data. Since the localization with features is mainly needed when the AGV is moving in places with occluded reflectors, the acquired datasets represent a typical scenario of an AGV entering and operating in a block storage with high lateral walls of pallets. Acquired paths describe different

entering path with different lengths. Each path is acquired at least six times. Thus, multiple runs of the same test can be performed.

5.6 Localization results

Two main aspects are taken in account in the following experiments:

- Repeatability: the AGV needs to repeat the path with high precision. If a positioning error is present during the test, this error has to be constant and systematic. Thus, the AGV behavior is always predictable.
- Accuracy: to allow the AGV to correctly fork the pallets even at high height, the position error on the operational point needs to be as small as possible. The requirements are that the pose error is bounded in position with $2cm$ and orientation with $10cdeg$. These values guarantee a successful operation of pallet picking or placing.

Before testing the whole feature-based localization system, the performance of feature navigation has been tested with an off-line fixed map of keypoints previously extracted.

Following the mapping approach in [65], first all scans are acquired and positioned with laser poses provided by reflector-based localization. For each scan, FALKO keypoints are extracted and geometrically clustered. For each cluster with enough points, the keypoint centroid is computed and inserted in the map at fixed position. Figure 5.7 shows an example of keypoint cluster and corresponding centroid.

The performance of AGV localization with FALKO feature off-line mapping has been evaluated making the AGV travel a linear $12m$ long path inside the block storage. The EKF localization filter described in section 5.3 has been used and results compared to existing localization system data. The path has been traveled six times to evaluate repeatability of localization. Position and orientation errors have been extracted (Fig. 5.10). As can be seen in figure 5.10, the localization results for each run of the test are fully compliant with the requirements. Positioning error is bounded by half centimeter, while orientation error is near zero for almost each run.

Experiments with external dynamic maps have been performed similar to off-line mapping experiments. Several paths which lead the AGV to the operational point inside the block storage have been traveled. In particular, four paths of different lengths have been evaluated with multiple runs. Table 5.6 shows the path length.

The three longest paths differ in starting and ending points position while the latter one is kept always inside the block storage. The dynamic map update rate threshold has been set to 0.1 resulting in a very low update of keypoints position. Thus, when a keypoint is inserted in the dynamic map, its position is almost immutable during its life. The choice of a very low update rate of keypoints position reduces the drift error. Figures 5.11, 5.12 and 5.13 show the results for the three longer paths, while figure 5.14 shows the results for the $7m$ long path. As can be seen, for all tests the orientation error is quite limited, while position errors are different for each path. The positioning error is mainly due to the drift error in robot orientation leading to high lateral error position in long paths. Path 4 test shows performance compliant with the requirement, allowing a safe and correct AGV navigation through a path at most $7m$ long.

5.7 Discussion

In this chapter, an application in industrial environment of the methods described in the previous chapters has been illustrated. The automatic calibration method presented in chapter 2 has been evaluated in a real-case scenario resulting in a valid alternative to manual calibration. Indeed, the results show a compatibility with manual calibration results but with a significant reduction of calibration time. Feature-based localization and mapping with EKF has also been tested in unstructured industrial environment. The results show that FALKO-based localization accuracy is compatible with typical industrial requirements. Localization accuracy primarily depends on feature stability and sensor calibration. A dynamic map, not included in the filter state, provides a good solution for short-term localization. Furthermore, the proposed method demonstrates the possibility of life-long mixed localization (features and artificial reflectors) thanks to the compatibility of landmark types.

n_{seg}	α-min/max	β	v_w	α_{off} [cdeg]		s_w [mm/tick]		l_x [mm]		l_y [mm]		l_θ [cdeg]	
	[deg]	[deg]	[mm/s]	avg	std	avg	std	avg	std	avg	std	avg	std
2	18 ÷ 50	180	1000	-75.93	0.39	0.249685	0.000036	390.28	0.13	2.14	0.14	50.34	0.32
4	18 ÷ 34	180	1000	-75.89	0.30	0.249806	0.000023	389.09	0.16	1.63	0.27	43.03	0.21
4	18 ÷ 50	60	1000	-77.05	0.72	0.249576	0.000043	395.79	0.34	4.21	0.63	36.88	1.18
4	18 ÷ 50	120	1000	-75.36	0.32	0.249880	0.000059	386.68	0.51	6.17	0.22	37.83	0.70
4	18 ÷ 50	180	800	-76.73	0.59	0.249652	0.000111	387.81	0.60	2.78	0.18	44.57	0.28
4	18 ÷ 50	180	1000	-76.27	0.62	0.249668	0.000040	387.10	0.25	2.48	0.47	44.64	0.80
4	18 ÷ 50	180	1200	-75.82	0.35	0.249690	0.000007	387.89	0.66	1.85	0.22	47.30	1.12
4	18 ÷ 50	180	1400	-75.57	0.63	0.249686	0.000023	388.04	0.43	1.81	0.15	48.19	0.77
4	35 ÷ 50	180	1000	-73.69	0.44	0.250081	0.000004	386.03	0.22	0.00	0.26	43.01	1.80
6	18 ÷ 50	180	1000	-76.01	0.31	0.249816	0.000085	389.03	0.32	1.82	0.53	43.35	0.44
8	18 ÷ 50	180	1000	-75.64	0.11	0.249750	0.000015	389.23	0.09	2.79	0.28	40.46	0.59
10	18 ÷ 50	180	1000	-75.03	0.09	0.249772	0.000023	388.94	0.17	3.23	0.12	42.65	0.34
4	18 ÷ 34	180	800	-77.47	0.48	0.249716	0.000122	389.00	0.03	2.03	0.48	41.84	1.73
4	18 ÷ 34	180	1200	-77.74	0.30	0.249789	0.000006	388.91	0.39	1.73	0.12	42.10	0.50
4	18 ÷ 34	180	1400	-77.40	0.65	0.249736	0.000039	389.45	0.37	1.97	0.56	42.26	0.54
6	18 ÷ 34	180	800	-75.48	0.33	0.249731	0.000061	389.48	0.15	2.47	0.24	41.24	0.28
6	18 ÷ 34	180	1000	-75.75	0.29	0.249753	0.000027	389.29	0.45	2.52	0.41	40.38	0.79
6	18 ÷ 34	180	1200	-74.74	0.16	0.249706	0.000027	389.76	0.45	2.50	0.49	41.84	0.47
6	18 ÷ 34	240	1200	-75.03	0.11	0.249807	0.000007	390.13	0.15	4.49	0.38	38.08	0.35
6	18 ÷ 50	240	800	-75.49	0.16	0.249744	0.000052	390.14	0.21	6.77	0.35	39.58	0.55

Table 5.1: Standard Tricycle Calibration parameters at different conditions for a CB16 AGV

n_{seg}	$\alpha_{min/max}$	β	v_w	α_F [cdeg]		α_B [cdeg]		δ_w [mm/tick]		l_x [mm]		l_y [mm]		l_θ [cdeg]	
	[deg]	[deg]	[mm/s]	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
2	18 ÷ 50	180	1000	-74.84	0.81	-77.13	0.33	0.249746	0.000038	390.27	0.16	2.21	0.10	50.32	0.40
4	18 ÷ 34	180	1000	-72.30	0.43	-79.22	0.47	0.249888	0.000008	389.09	0.20	1.43	0.36	42.98	0.26
4	18 ÷ 50	60	1000	-71.59	0.76	-82.81	0.52	0.249664	0.000029	395.79	0.42	4.34	0.50	36.82	1.46
4	18 ÷ 50	120	1000	-73.24	0.07	-76.00	0.62	0.249978	0.000040	386.68	0.63	5.33	0.19	37.81	0.85
4	18 ÷ 50	180	800	-75.71	0.61	-76.76	0.82	0.249717	0.000129	387.80	0.73	2.23	0.49	44.56	0.34
4	18 ÷ 50	180	1000	-75.07	1.04	-76.69	0.36	0.249762	0.000047	387.10	0.30	2.04	0.48	44.63	0.98
4	18 ÷ 50	180	1200	-75.09	0.41	-76.28	0.55	0.249790	0.000030	387.89	0.81	1.71	0.68	47.29	1.37
4	18 ÷ 50	180	1400	-74.97	0.99	-75.87	0.39	0.249810	0.000040	388.04	0.52	1.64	0.40	48.18	0.94
4	35 ÷ 50	180	1000	-71.15	0.63	-75.93	0.26	0.250092	0.000005	386.03	0.27	-0.09	0.30	42.96	2.20
6	18 ÷ 50	180	1000	-72.02	0.37	-78.76	0.37	0.249882	0.000105	389.03	0.39	1.16	0.66	43.29	0.53
8	18 ÷ 50	180	1000	-72.70	0.34	-77.47	0.46	0.249823	0.000027	389.23	0.11	2.22	0.39	40.43	0.72
10	18 ÷ 50	180	1000	-71.25	0.44	-77.87	0.39	0.249842	0.000025	388.94	0.21	2.76	0.23	42.60	0.40
4	18 ÷ 34	180	800	-74.79	0.72	-79.68	0.34	0.249776	0.000163	389.00	0.04	1.69	0.66	41.81	2.12
4	18 ÷ 34	180	1200	-74.41	0.39	-80.38	0.34	0.249869	0.000028	388.91	0.48	1.23	0.30	42.06	0.62
4	18 ÷ 34	180	1400	-74.65	0.99	-79.94	0.64	0.249831	0.000034	389.45	0.46	1.82	0.83	42.22	0.66
6	18 ÷ 34	180	800	-72.07	0.48	-78.67	0.34	0.249800	0.000069	389.47	0.18	2.31	0.42	41.19	0.33
6	18 ÷ 34	180	1000	-72.21	0.32	-79.13	0.27	0.249821	0.000028	389.29	0.56	2.40	0.55	40.32	0.96
6	18 ÷ 34	180	1200	-71.42	0.07	-78.05	0.19	0.249788	0.000030	389.77	0.55	2.49	0.56	41.79	0.57
6	18 ÷ 34	240	1200	-72.41	0.23	-78.05	0.34	0.249854	0.000013	390.14	0.18	4.78	0.39	38.04	0.42
6	18 ÷ 50	240	800	-71.74	0.33	-78.40	0.16	0.249791	0.000059	390.14	0.26	6.33	0.51	39.54	0.67

Table 5.2: Asymmetric Tricycle Calibration parameters at different conditions for a CB16 AGV

AGV		α_F [cdeg]	α_B [cdeg]	s_w [mm/tick]	l_x [mm]	l_y [mm]	l_θ [cdeg]
Setting 1	$LGV71^{(a)}$	-88.77	-76.70	0.250911	1485.86	-9.80	236.49
	$LGV74^{(a)}$	-61.68	-56.34	0.249778	1508.43	-9.00	-22.32
	$LGV01^{(a)}$	-87.79	-86.39	0.249122	1500.61	-2.90	-82.36
Setting 2	$R32^{(m)}$	-120.65	-120.65	0.251885	1310	17	140
	$R30^{(a)}$	-107.32	-107.81	0.251801	1293	17	-75
	$R31^{(a)}$	-111.12	124.94	0.252952	1317	7	-131
	$R36^{(a)}$	-107.53	-112.13	0.253143	1294	14	-62

Table 5.3: Calibration parameters of the AGVs used in *Setting* 1 and *Setting* 2 according to ATC model. Label $^{(a)}$ refers to AGVs calibrated using the proposed method and label $^{(m)}$ to the manually calibrated ones.

Operation	$LGV74^{(a)}$			$LGV74^{(a)}$			$LGV01^{(a)}$		
Points	x [mm]	y [mm]	dist [mm]	x [mm]	y [mm]	dist [mm]	x [mm]	y [mm]	dist [mm]
A	0.7	-13.0	13.0	5.7	5.0	7.6	-6.3	8.0	10.2
B	5.3	1.7	5.6	-9.7	-0.3	9.7	4.3	-1.3	4.5
C	-0.7	0.0	0.7	-0.7	0.0	0.7	1.3	0.0	1.3
D	0.3	-2.0	2.0	1.3	1.0	1.7	-1.7	1.0	1.9
E	-0.3	2.0	2.0	2.7	0.0	2.7	-2.3	-2.0	3.1
Avg			4.7			4.4			4.2

Table 5.4: Halting point coordinates and distance to the origin of the local reference frame for the three AGVs of *Setting* 1.

Operation	$R32^{(m)}$			$R30^{(a)}$			$R31^{(a)}$			$R36^{(a)}$		
Points	x [mm]	y [mm]	dist [mm]	x [mm]	y [mm]	dist [mm]	x [mm]	y [mm]	dist [mm]	x [mm]	y [mm]	dist [mm]
P_1	-2.8	-4.5	5.3	7.3	1.5	7.4	-4.8	-3.5	5.9	0.3	6.5	6.5
P_2	-7.8	-1.5	7.9	5.3	-4.5	6.9	0.3	0.5	0.6	2.3	5.5	5.9
P_3	7.8	6.3	10.0	-5.3	0.3	5.3	-3.3	-0.8	3.3	0.8	-5.8	5.8
P_4	-2.5	1.0	2.7	4.5	-3.0	5.4	-0.5	-2.0	2.1	-1.5	4.0	4.3
P_5	7.3	6.0	9.4	-5.8	0.0	5.8	-2.8	1.0	2.9	1.3	-7.0	7.1
P_6	-7.3	-10.5	12.8	4.8	0.5	4.8	0.8	-1.5	1.7	1.8	11.5	11.6
P_7	2.5	8.5	8.9	-2.5	-2.5	3.5	-2.5	1.5	2.9	2.5	-7.5	7.9
P_8	4.5	8.3	9.4	-5.5	-0.8	5.6	-0.5	0.3	0.6	1.5	-7.8	7.9
P_9	5.0	7.5	9.0	-7.0	-5.5	8.9	2.0	4.5	4.9	0.0	-6.5	6.5
P_{10}	5.0	3.5	6.1	-5.0	-0.5	5.0	0.0	2.5	2.5	0.0	-5.5	5.5
Avg			8.1			5.9			2.7			6.9

Table 5.5: Halting point coordinates and distance to the origin of the local reference frame for the four AGVs of *Setting* 2.

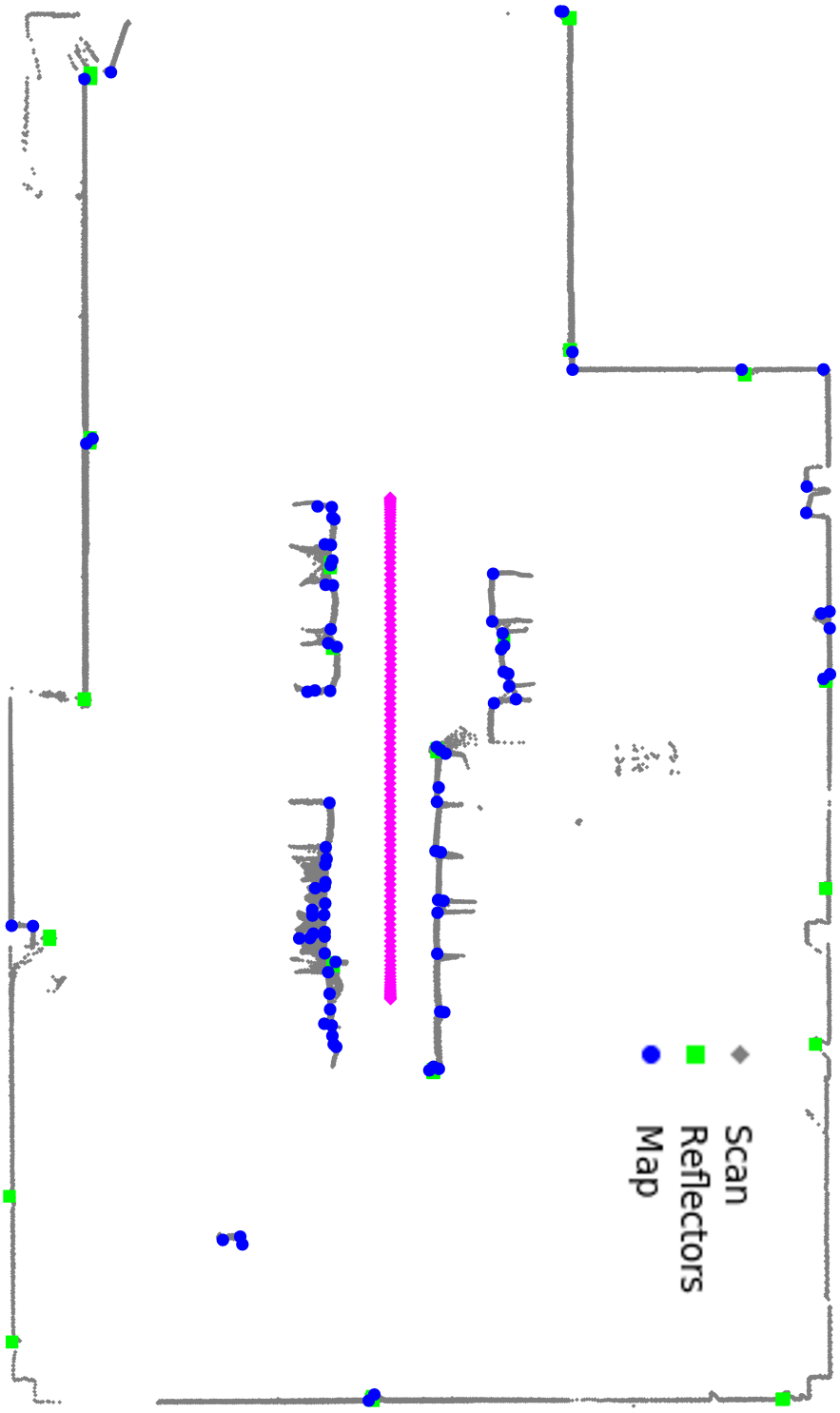


Figure 5.9: FALKO keypoints mapping. Grey points are original scan points representing the warehouse profile, green points are detected reflectors. Blue points represents FALKO keypoints in the map. The pink line represents the robots poses during mapping.

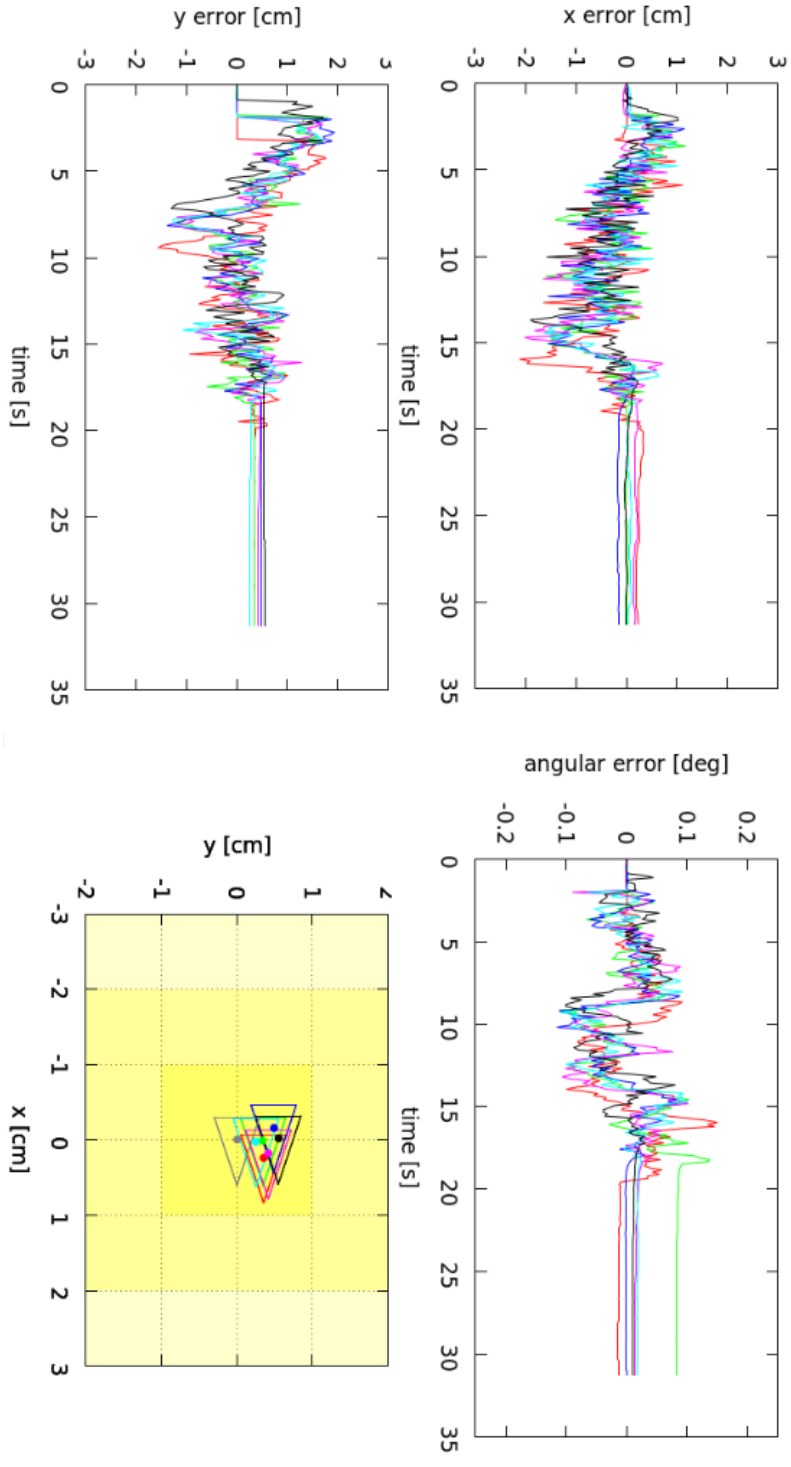


Figure 5.10: FALKO keypoints mapping results. 6 runs with a path 12m long

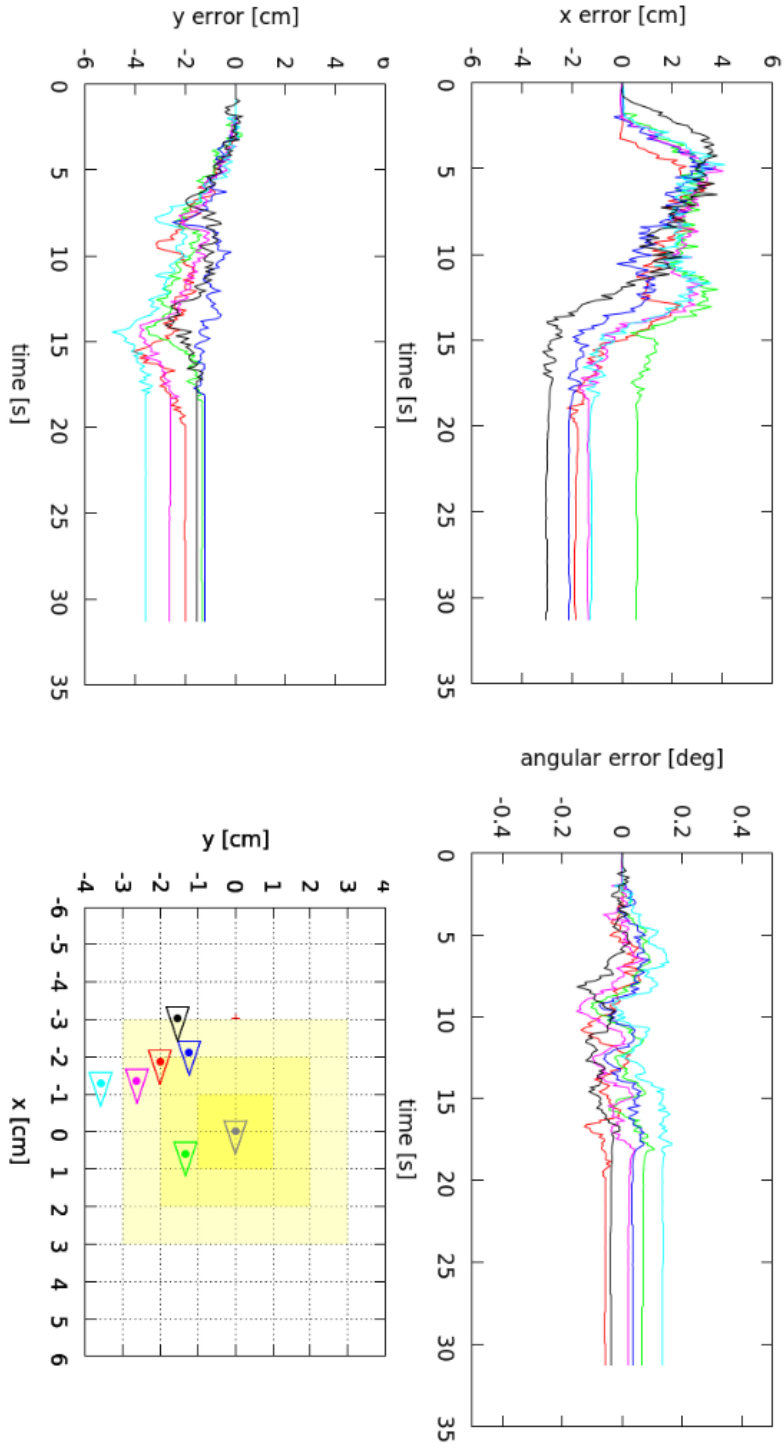


Figure 5.11: FALKO keypoints dynamic map results. 6 runs with path 1 (12m long)

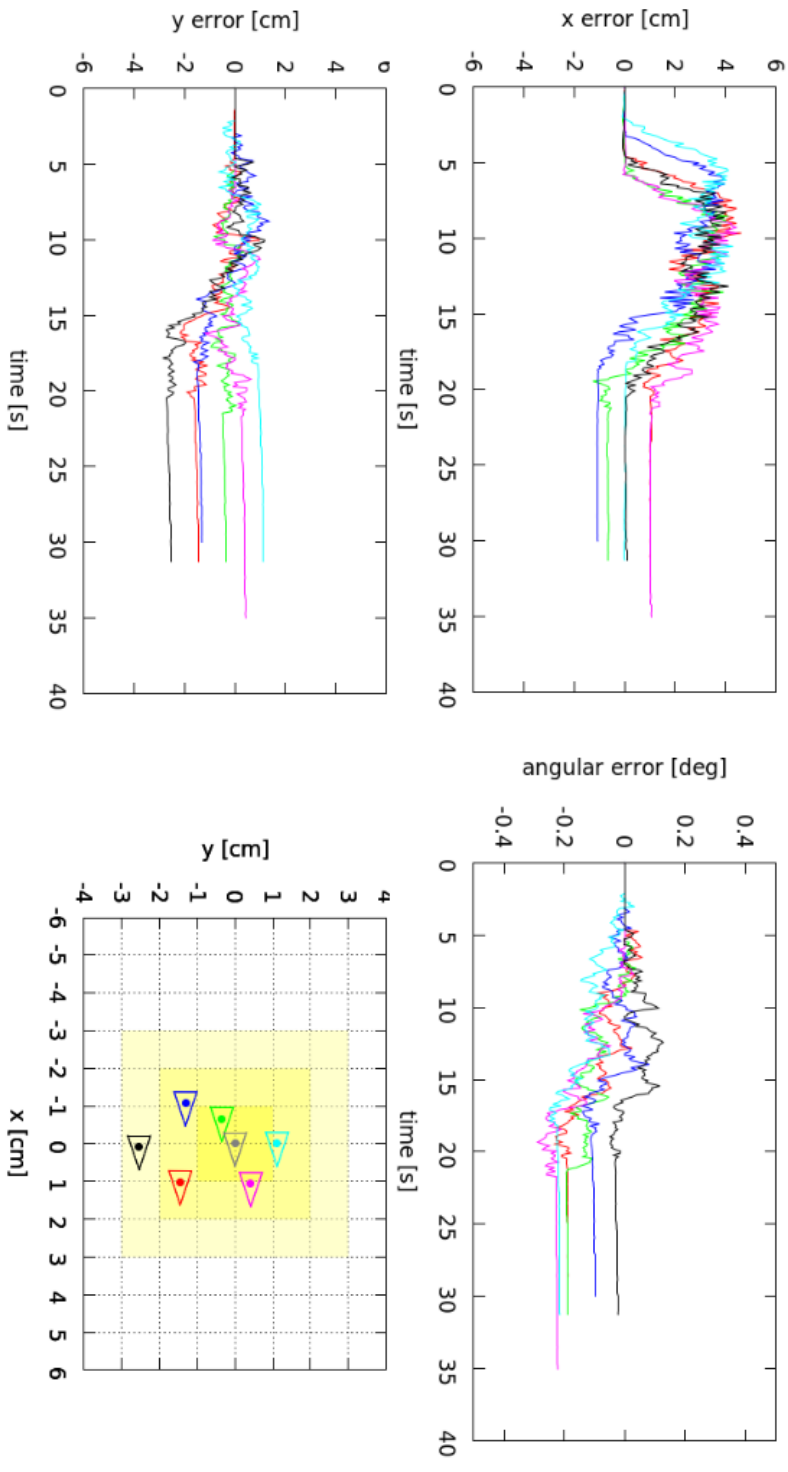


Figure 5.12: FALKO keypoints dynamic map results. 6 runs with path 2 (12m long)

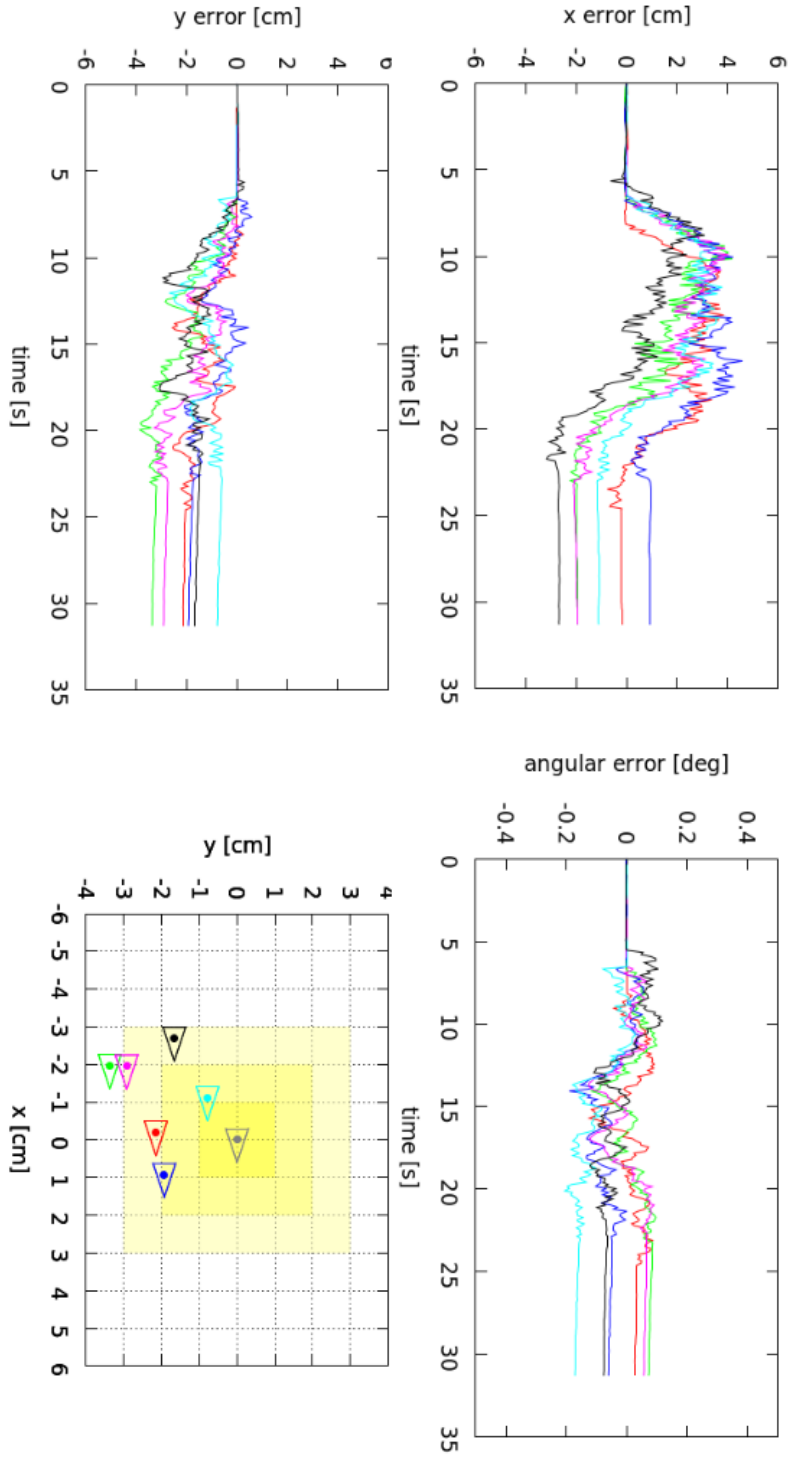


Figure 5.13: FALKO keypoints dynamic map results. 6 runs with path 3 (12m long)

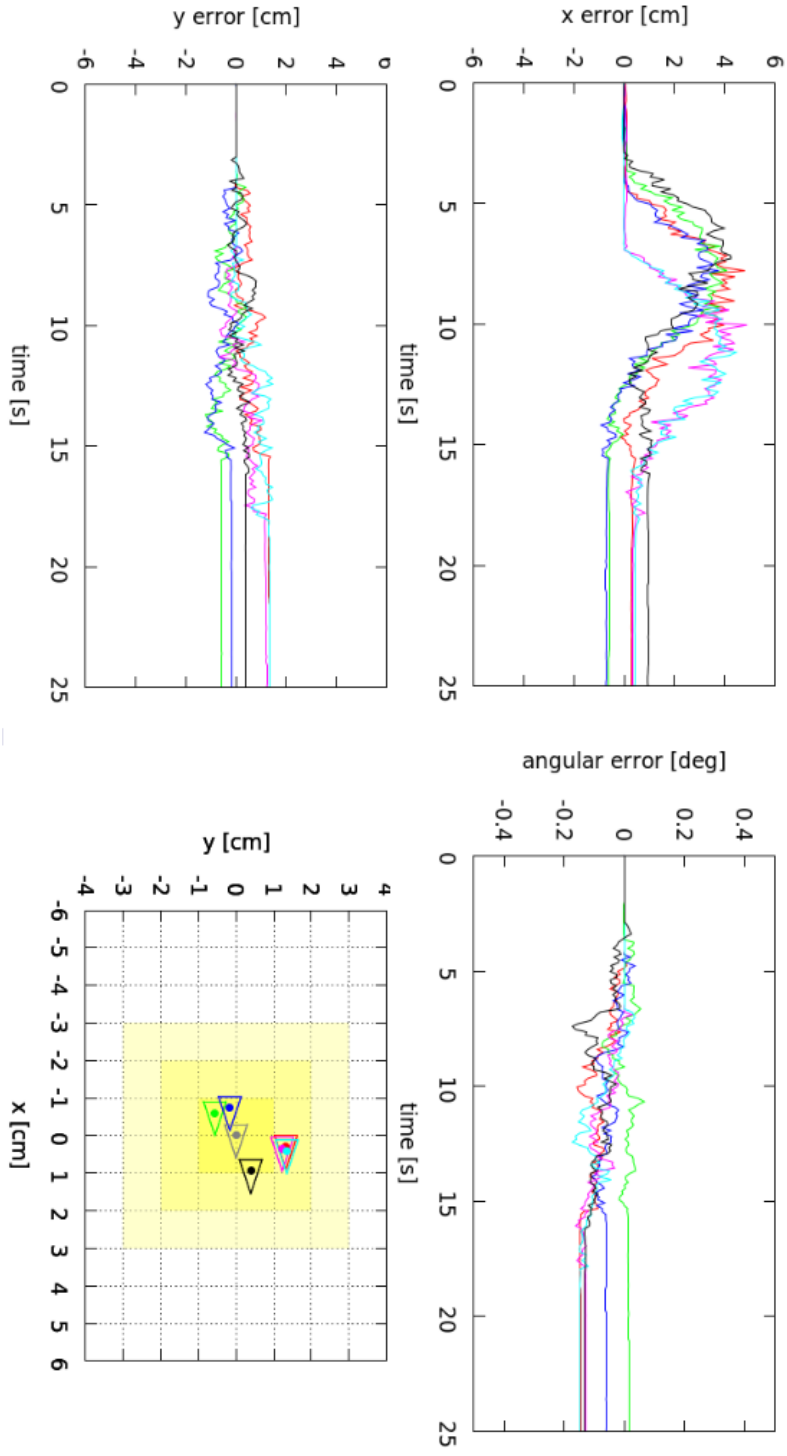


Figure 5.14: FALKO keypoints dynamic map results. 6 runs with path 4 (7*m* long)

Path index	length [m]
1	12
2	12
3	12
4	7

Table 5.6: Index and length of the paths used in EKF-FALKO localization.

Conclusions

This thesis has dealt with the problem of robot autonomous localization and mapping with odometry and laser scanner sensors in unstructured environments. While several type of sensors are today available on the market, the laser scanner is still preferred in applications where accuracy and reliability are fundamental requirements. Feature-based approaches are a class of methods well studied in computer vision and 3D point clouds processing, but relatively new in 2d range sensing.

In this thesis, I have proposed a keypoint detector, named FALKO, with two novel descriptors, BSC and CGH, designed to be distinguishable features in range finder measurements (chapter 3). Like other state-of-the-art detectors, FALKO is conceived to find stable high curvature points in a laser scan, and to be invariant to sensor viewpoint and point density. FALKO satisfies these properties through efficient evaluation of the neighbor point distribution instead of relying on a computationally expensive multi-scale approach. The time required to process a laser scan with FALKO is one or two orders of magnitude less than with other state-of-the-art keypoint detectors. The proposed descriptors BSC and CGH provide an equally efficient signature for the keypoints. The performance of the proposed algorithms has been tested on widely used benchmark datasets and compared with OC and FLIRT, the state-of-the-art features for LIDARs. The results about detection show that FALKO achieves higher repeatably score and extracts less ephemeral points than the other keypoint detectors. Moreover, the precision-recall curves of the proposed descriptors are consistent with the achievable results obtained from computer vision and laser scan data descriptors.

In chapter 4, I have illustrated novel loop closure methods based on FALKO keypoints and compared their performance in both offline and online localization and mapping problems. The FALKO keypoints detected from each laser scan become part of a corresponding local map. To find loops, i.e. regions already visited by the robot, the current local map is matched with the other local maps in two steps. The candidate loops are found by comparing signatures and, then, point-to-point association is applied to match individual keypoints and to compute the accurate rigid transformation between local maps. The novel signature GLAROT has also been proposed and compared with state-of-the-art signature algorithms. Moreover, several point-to-point data association techniques have been evaluated. Experimental results obtained from publicly available datasets have assessed performance of loop closure methods applied to both offline and online localization and mapping problems. Online place recognition imposes stronger constraints over results and has not been addressed in previous works based on laser keypoints. Results show that the stability of FALKO keypoints has proven crucial for loop closure. FALKO detector combined with GLAROT signature and point-to-point association outperforms the previously proposed approaches.

The proposed keypoint feature for LIDARs has been applied to perform accurate robot localization in industrial warehouse applications. In such specific scenario, accuracy is achieved by removing all potential error sources including systematic errors. In chapter 2, I have proposed a calibration method that simultaneously computes the intrinsic and extrinsic parameters of a mobile robot compliant to the tricycle wheeled robot model, which is a common kinematic configuration of industrial AGVs. The calibration is performed by computing the parameters better fitting the input commands and the sensor egomotion estimation obtained from the sensor measurements. Two formulations of the calibration problem have been developed. The Standard Tricycle Calibration (STC) problem refers to a five parameter model (the steering offset and driving scale, and three sensor pose coordinates). The Asymmetric Tricycle Calibration (ATC) problem considers a six parameter model that distinguishes the value

of steering offset in forward and backward motion. A closed form solution is provided for STC problem, while the ATC is solved through a one-dimension numerical search. Moreover, the observability property of the method has been formally proved and feasibility conditions on the input trajectory for the estimation have been provided. Experiments (chapter 5) have been carried out using real industrial AGVs in a warehouse to assess the effectiveness of the proposed calibration. The variance of estimated parameters over repeated calibration trials is low if qualitatively compared with results of the manual iterative procedure. Furthermore, AGVs calibrated with the proposed method have shown the ability to stop at the same operation points with a typical accuracy of 10 mm. With the suggested number of trajectory segments, the proposed calibration method takes about 12 minutes instead of one hour or more required by the manual iterative procedure. The availability of an automatic, fast, and accurate calibration method brings the potential for more frequent recalibration and hence better AGV navigation in real warehouses.

Chapter 5 illustrates an application of LIDAR featured-based localization to industrial AGVs used to manage warehouse logistics. In current industrial systems, AGV navigation usually exploits artificial landmarks that are easily detected by laser scanner sensors. This solution enables accurate and robust pose estimation using lightweight maps at the cost of placing and manually mapping landmarks. The proposed FALKO features allow accurate localization based on dynamic landmark maps as well as reliability and robustness as required in industrial applications, while reducing the need for artificial reflectors. Moreover, keypoint features can be easily integrated in existing localization systems. Results show that the feature-based approach achieves performance comparable to artificial landmark localization and compliant with requirements of reliable navigation. Both static and dynamic feature mapping have been tested showing that a static map of FALKO features performs like the reflectors counterparts. Dynamic maps without loop closure maintain good performance in paths about $10m$ long.

Bibliography

[1] J.-S. Gutmann and K. Konolige. Incremental Mapping of Large Cyclic Environments. In *Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 318–325, 1999.

[2] D. Lodi Rizzini and S. Caselli. Metric-topological maps from laser scans adjusted with incremental tree network optimizer. *Robotics & Autonomous Systems*, 57(10):1036 – 1041, 2009. doi:http://dx.doi.org/10.1016/j.robot.2009.07.022.

[3] D. Hähnel, S. Thrun, B. Wegbreit, and W. Burgard. Towards lazy data association in SLAM. In *Proc. of the International Symposium on Robotics Research (ISRR)*, 2003.

[4] G. Grisetti, C. Stachniss, and W. Burgard. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE Trans. on Robotics*, 23(1):34–46, 2007.

[5] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. 27(10):1615–1630, 2005.

[6] S. Krig. Interest point detector and feature descriptor survey. In *Computer Vision Metrics*, pages 217–282. Springer, 2014.

[7] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan. 3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey. *IEEE*

Trans. on Pattern Analysis and Machine Intelligence, 36(11):2270–2287, Nov 2014. doi:10.1109/TPAMI.2014.2316828.

[8] G. D. Tipaldi and K. O. Arras. Flirt-interest regions for 2d range data. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3616–3622, 2010.

[9] Y. Li and E. Olson. Extracting general-purpose features from LIDAR data. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.

[10] Y. Li and E. Olson. Structure tensors for general purpose lidar feature extraction. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1869–1874, 2011.

[11] F. Kallasi, D. Lodi Rizzini, and S. Caselli. Fast keypoint features from laser scanner for robot localization and mapping. *RAL*, 1(1):176–183, jan 2016. DOI 10.1109/LRA.2016.2517210. doi:10.1109/LRA.2016.2517210.

[12] A. Howard and N. Roy. Radish: The Robotics Data Set Repository, Standard data sets for the robotics community. URL: http://radish.sourceforge.net/.

[13] F. Kallasi and D. Lodi Rizzini. Efficient Loop Closure based on FALKO LIDAR Features for Online Robot Localization and Mapping. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1–8, 2016.

[14] G. D. Tipaldi, L. Spinello, and K. O. Arras. Geometrical flirt phrases for large scale place recognition in 2d range data. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2693–2698, 2013.

[15] M. Himstedt and E. Maehle. Geometry matters: Place recognition in 2D range scans using Geometrical Surface Relations. In *Proc. of the European Conference on Mobile Robots (ECMR)*, pages 1–6, 2015. doi:10.1109/ECMR.2015.7324185.

[16] M. Himstedt, J. Frost, S. Hellbach, H.-J. Boehme, and E. Maehle. Large scale place recognition in 2D lidar scans using geometrical landmark relations. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 5030–5035, 2014.

[17] A. Censi, A. Franchi, L. Marchionni, and G. Oriolo. Simultaneous calibration of odometry and sensor parameters for mobile robots. *IEEE Trans. on Robotics*, 29(2), April 2013. doi:10.1109/TRO.2012.2226380.

[18] J.G. Kang, W.-S. Choi, S.-Y. An, and S.-Y. Oh. Augmented EKF based SLAM method for improving the accuracy of the feature map. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3725–3731, Oct 2010. doi:10.1109/IROS.2010.5652938.

[19] J.H. Song and G.-I. Jee. Kalman filter based on-line calibration of laser scanner for vehicle navigation. In *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*, pages 1437–1441, Oct 2011.

[20] A. Martinelli, D. Scaramuzza, and R. Siegwart. Automatic self-calibration of a vision system during robot motion. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 43–48, May 2006. doi:10.1109/ROBOT.2006.1641159.

[21] D. Cucci and M. Matteucci. A Flexible Framework for Mobile Robot Pose Estimation and Multi-Sensor Self-Calibration. In *Proc. of the Intl. Conf. on Informatics in Control, Automation and Robotics (ICINCO)*, 2013.

[22] J. Borenstein and L. Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, 12(6):869–880, Dec 1996. doi:10.1109/70.544770.

[23] A. Martinelli, N. Tomatis, and R. Siegwart. Simultaneous localization and odometry self calibration for mobile robot. *Autonomous Robots*, 22(1):75–85, 2007.

[24] G. Antonelli and S. Chiaverini. Linear estimation of the physical odometric parameters for differential-drive mobile robots. *Journal of Autonomous Robots*, 23(1):59–68, 2007. doi:10.1007/s10514-007-9030-2.

[25] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics : modelling, planning and control*. Advanced Textbooks in Control and Signal Processing. Springer, London, 2009.

[26] K. Lee and W. Chung. Calibration of kinematic parameters of a Car-Like Mobile Robot to improve odometry accuracy. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2546–2551, May 2008. doi:10.1109/ROBOT.2008.4543596.

[27] E.M. Foxlin. Generalized architecture for simultaneous localization, auto-calibration, and map-building. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 1, pages 527–533, 2002. doi:10.1109/IRDS.2002.1041444.

[28] T. Sasaki and H. Hashimoto. Calibration of laser range finders based on moving object tracking in Intelligent Space. In *Int. Conf. on Networking, Sensing and Control (ICNSC)*, pages 620–625, March 2009. doi:10.1109/ICNSC.2009.4919349.

[29] J.P. Underwood, A. Hill, T. Peynot, and S.J. Scheding. Error modeling and calibration of exteroceptive sensors for accurate mapping applications. *Journal of Field Robotics*, 27(1):2–20, 2010. doi:10.1002/rob.20315.

[30] J. Brookshire and S. Teller. Automatic calibration of multiple coplanar sensors. In *Robotics: Science and Systems*. MIT Press, 2011.

[31] J. Brookshire and S. Teller. Extrinsic Calibration from Per-Sensor Egomotion. In *Robotics: Science and Systems*. MIT Press, 2012.

[32] F. Kallasi, D. Lodi Rizzini, F. Oleari, M. Magnani, and S. Caselli. A Novel Calibration Method for Industrial AGVs. *Submitted for Publication*, 2016.

[33] P.J. Besl and H.D. Mckay. A method for registration of 3-d shapes. *IEEE Trans. Pat. Anal. Mach. Intel*, 14(2):239–256, 1992.

[34] F. Lu and E. Milios. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 935–938, 1994.

[35] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Journal of Autonomous Robots*, 4:333–349, 1997.

[36] S.T. Pfister, K.L. Kriechbaum, S.I. Roumeliotis, and J.W. Burdick. Weighted Range Sensor Matching Algorithms for Mobile Robot Displacement Estimation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1667–1674, 2002.

[37] B. Jensen and R. Siegwart. Scan alignment with probabilistic distance metric. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 3, pages 2191–2196 vol.3, Sept 2004. doi:10.1109/IROS.2004.1389734.

[38] J. Minguez, F. Lamiraux, and L. Montesano. Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3557–3563, 2005.

[39] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó. The slam problem: A survey. In *Proceedings of the 2008 Conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*, pages 363–371, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press. URL: http://dl.acm.org/citation.cfm?id=1566899.1566949.

[40] S. Thrun. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[41] A. J. Davison and D. W. Murray. Simultaneous localization and map-building using active vision. 24(7):865–880, Jul 2002. doi:10.1109/TPAMI.2002.1017615.

[42] M. Bosse, P.M. Newman, J.J. Leonard, and S. Teller. An ALTAS framework for scalable mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2003.

[43] P. Jensfelt, D. Kragic, J. Folkesson, and M. Bjorkman. A framework for vision based bearing only 3d slam. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1944–1950, May 2006. doi:10.1109/ROBOT.2006.1641990.

[44] S. Se, D. Lowe, and J. Little. Mobile Robot Localization and Mapping with uncertainty using Scale-Invariant Visual Landmark. *Int. Journal of Robotics Research*, 21(8):735–758, August 2002.

[45] S. Thrun and Y. Liu. *Multi-robot SLAM with Sparse Extended Information Filers*, pages 254–266. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. URL: http://dx.doi.org/10.1007/11008941_27, doi:10.1007/11008941_27.

[46] S. Thrun, C. Martin, Y. Liu, D. Hahnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard. A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics and Automation*, 20(3):433–443, June 2004. doi:10.1109/TRA.2004.825520.

[47] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Y. Ng. *Simultaneous Mapping and Localization with Sparse Extended Information Filters: Theory and Initial Results*, pages 363–380. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. URL: http://dx.doi.org/10.1007/978-3-540-45058-0_22, doi:10.1007/978-3-540-45058-0_22.

[48] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.

[49] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually Navigating the RMS Titanic with SLAM Information Filters. In *Proceedings of Robotics: Science and Systems*, 2005.

[50] R. Eustice, H. Singh, and J.J. Leonard. Exactly Sparse Delayed-State filters. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2428–2435, 2005.

[51] E.A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. pages 153–158, 2000. doi:10.1109/ASSPCC.2000.882463.

[52] P. Del Moral. Nonlinear filtering: Interacting particle resolution, 1996.

[53] M. Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Carnegie Mellon University, 2003.

[54] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1151–1156, Acapulco, Mexico, 2003.

[55] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental Smoothing and Mapping. *IEEE Trans. on Robotics*, 24(2):1365–1378, 2008.

[56] T. Duckett, S. Marsland, and J. Shapiro. Fast, On-line Learning of Globally Consistent Maps. *Journal of Autonomous Robots*, 12(3):287 – 300, 2002.

[57] U. Frese. Treemap: An $O(logn)$ Algorithm for Indoor Simultaneous Localization and Mapping. *Journal of Autonomous Robots*, 21(2):103–122, 2006. URL: http://www.informatik.uni-bremen.de/~ufrese/published/fresear06b.pdf.

[58] E. Olson, J. Leonard, and S. Teller. Fast Iterative Optimization of Pose Graphs with Poor Initial Estimates. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2262–2269, 2006.

[59] G. Grisetti, D. Lodi Rizzini, C. Stachniss, E. Olson, and W. Burgard. Online Constraint Network Optimization for Efficient Maximum Likelihood Map Learning. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1880–1885, 2008.

[60] G. Grisetti, R. K ummerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical Optimization on Manifolds for Online 2D and 3D Mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.

[61] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A General Framework for Graph Optimization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[62] G. Weiss and E. Puttkamer. A Map based on laser scans without geometric interpretation. *Intelligent Autonomous Systems*, 4:403–407, 1995.

[63] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2743–2748, 2003.

[64] E. Olson. Real-time correlative scan matching. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 4387–4393, 2009.

[65] Y. Li and E. Olson. A general purpose feature extractor for light detection and ranging data. *Sensors*, 10(11):10356–10375, Nov. 2010. doi:10.3390/s101110356.

[66] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally consistent 3d mapping with scan matching. *Robot. Auton. Syst.*, 56(2):130–142, February 2008. URL: http://dx.doi.org/10.1016/j.robot.2007.07.002, doi:10.1016/j.robot.2007.07.002.

[67] M. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2D LIDAR maps. *Robotics & Autonomous Systems*, 57(12):1211–1224, 2009.

[68] A. Censi. An ICP variant using a point-to-line metric. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2008.

[69] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009. doi:10.15607/RSS.2009.V.021.

[70] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[71] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart. A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1929–1934, 2005.

[72] P. Nunez, R. Vazquez-Martin, J.C. Del Toro, A. Bandera, and F. Sandoval. Feature extraction from laser scan data based on curvature estimation for mobile robotics. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1167–1172, 2006.

[73] L. Pedraza, D. Rodriguez-Losada, F. Matía, G. Dissanayake, and J.V. Miró. Extending the limits of feature-based SLAM with B-splines. *IEEE Trans. on Robotics*, 25(2):353–366, 2009.

[74] M. Bosse and R. Zlot. Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM. *Int. Journal of Robotics Research*, 27(6):667–691, Jun 2008. doi:10.1177/0278364908091366.

[75] A. Censi, L. Iocchi, and G. Grisetti. Scan Matching in the Hough Domain. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.

[76] J. Neira and J.D. Tardós. Data Association in Stochastic Mapping Using the Joint Compatibility Test. *IEEE Trans. on Robotics*, 17(6):890–897, 2001.

[77] T. Bailey, E.M. Nebot, J.K. Rosenblatt, and H.F. Durrant-Whyte. Data association for mobile robot navigation: a graph theoretic approach. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2512–2517, 2000.

[78] L.M. Paz, P. Pinies, J. Neira, and J.D. Tardos. Global localization in slam in bilinear time. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2820–2826, Aug 2005. doi:10.1109/IROS.2005.1545055.

[79] M. De Cecco. Self-calibration of AGV inertial-odometric navigation using absolute-reference measurements. In *Proc. IEEE Instrumentation and Measurement Technology Conference (IMTC)*, volume 2, pages 1513–1518 vol.2, 2002. doi:10.1109/IMTC.2002.1007183.

[80] T.D. Larsen, M. Bak, N.A. Andersen, and O. Ravn. Location estimation for an autonomously guided vehicle using an augmented Kalman filter to autocalibrate the odometry. In *Proc. Int. Conf. Multisource-Multisens. Inf. Fusion*, 1998.

[81] P. Goel, S. Roumeliotis, and G.S. Sukhatme. Robust localization using relative and absolute position estimates. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1134–1140, 1999.

[82] P. Liang, Y.L. Chang, and S. Hackwood. Adaptive self-calibration of vision-based robot systems. *Systems, Man and Cybernetics, IEEE Trans. on*, 19(4):811–824, 1989.

[83] A. Martinelli and R. Siegwart. Observability Properties and Optimal Trajectories for On-line Odometry Self-Calibration. In *Proc. of the International Conference on Decision and Control*, pages 3065–3070, 2006.

[84] F.M. Mirzaei and S.I. Roumeliotis. A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation. *IEEE Trans. on Robotics*, 24(5):1143–1156, Oct 2008. doi:10.1109/TRO.2008.2004486.

[85] O.D. Faugeras and G. Toscani. Camera calibration for 3-D computer vision. In *Proc. Int. Work. Industrial Applications of Machine Vision and Machine Intelligence*, pages 240–247, 1987.

[86] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000. doi:10.1109/34.888718.

[87] R.Y. Tsai and R.K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. on Robotics and Automation*, 5(3):345–358, Jun 1989. doi:10.1109/70.34770.

[88] R. Horaud and F. Dornaika. Hand-eye Calibration. *Int. Journal of Robotics Research*, 14(3):195–210, 1995.

[89] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2301–2306, 2004.

[90] N. Roy and S. Thrun. Online self-calibration for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, volume 3, pages 2292–2297, 1999. doi:10.1109/ROBOT.1999.770447.

[91] G. Antonelli, S. Chiaverini, and G. Fusco. A calibration method for odometry of mobile robots based on the least-squares technique: theory and experimental validation. *IEEE Trans. on Robotics*, 21(5):994–1004, 2005.

[92] A. Kelly. Linearized Error Propagation in Odometry. *Int. Journal of Robotics Research*, 23(2):179–218, Feb. 2004.

[93] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Int. Journal of Robotics Research*, 5(4):56–68, 1986.

[94] H. Hmam. Quadratic optimisation with one quadratic equality constraint. Technical Report DSTO-TR-2416, Electronic Warfare and Radar Division,

Defence Science and Technology Organisation (Australia), 2010. URL: http://nla.gov.au/nla.arc-24592.

[95] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24, 2002.

[96] P. L. Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, February 1999. doi:10.1006/cviu.1998.0719.

[97] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005. doi:10.1007/s11263-005-3848-x.

[98] R.O. Duda and P.E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, Jan 1972. doi:10.1145/361237.361242.

[99] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111 – 122, 1981. doi:http://dx.doi.org/10.1016/0031-3203(81)90009-1.

[100] D. Ronzoni, R. Olmi, C. Secchi, and C. Fantuzzi. AGV global localization using indistinguishable artificial landmarks. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 287–292, 2011. doi:10.1109/ICRA.2011.5979759.