Wi-Closure: wireless sensing for multi-robot map matching

Enabling fast and reliable search of inter-robot loop closures in repetitive environments

A.C. Kemmeren

TU Delft

Wi-Closure: wireless sensing for multi-robot map matching

Enabling fast and reliable search of inter-robot loop closures in repetitive environments

by

A.C. Kemmeren

to obtain the degree of Master of Science

at Delft University of Technology,

to be defended publicly on Tuesday December 13, 2022 at 3:00 PM.

Student number:	4441370	
Project duration:	December 2021 – December 2022	
Thesis committee:	Dr. J. Alonso-Mora,	TU Delft, supervisor
	Dr. M. Jafarian,	TU Delft, supervisor
	W. Wang,	Harvard, daily supervisor
	Dr. M. Kok,	TU Delft
	Dr. J.F.P. Kooij	TU Delft

Contents

Abstract

This thesis proposes a novel algorithm, Wi-Closure, to improve computational efficiency and robustness of map matching in multi-robot SLAM. Current state-of-the-art techniques connect maps with inter-robot loop closures, that are usually found through place recognition. Wi-Closure decreases the computational overhead of these approaches by pruning the search space of potential loop closures, prior to evaluation by a typical place recognition algorithm. Wi-Closure achieves this by identifying where trajectories are close to each other through sensing spatial information directly from the wireless communication signal. Then, place recognition is only performed on scans taken at locations close to each other. Wireless sensing provides information even when operating in non-line-of-sight or without existing communication infrastructure. The validity of Wi-Closure is demonstrated in simulation and hardware experiments. Results show that using Wi-closure greatly reduces computation time, by 54% in simulation and by 77% in hardware, compared with a multi-robot SLAM baseline. Importantly, this is achieved without sacrificing accuracy. Using Wi-closure reduces absolute trajectory estimation error by 99% in simulation and 89% in hardware experiments. This improvement is due in part to Wi-Closure's ability to avoid catastrophic optimization failure that typically occurs with classical approaches in challenging repetitive environments.

1

Introduction

At present, robots are deployed in society as waitresses in a restaurant, to clean the house autonomously, or to help the police to explore drugs labs. Yet, in practice these robots rarely have elaborate collaborative capabilities. A major challenge to enable many collaborative tasks is that robots must first agree on what the environment looks like and where each robot is located. Pre-loading a map of the environment onto the robots or using global GPS for localization can largely solve these issues, but these information sources are not always available. Then, the problem becomes significantly more challenging: how do robots agree on a shared situational awareness, if no groundtruth map or location is available? This question has been a core motivation for the development of Simultaneous Localization and Mapping for multiple robots (multi-robot SLAM), but a number of open problems remain before multi-robot SLAM can be widely used in practice.

1.1. Multi-robot SLAM

SLAM enables robots to simultaneously estimate a (local) map or model of the environment and the location of the robot within this map, using information from on-board sensors such as Inertial Measurement Units (IMU's) and cameras [3]. In some cases, other robot states such as velocity are also estimated. The extension to multiple robots requires that the local maps are somehow related to each other. Current state-of-the-art methods use inter-robot loop closures [14, 25], which is in general a transformation (rotation and translation) between poses of two robots.

There are two distinct approaches to finding inter-robot loop closures. First, robots may observe, recognize and localize each other [32]. Unfortunately, this method does not generalize well to different environments. Observation requires the robots to move to a place where they are in direct line-of-sight of each other, which is difficult if the robots do not know their relative positions in an environment with obstructions. Once the robots are in sight, they need to recognize each other. Researchers often use visually recognizable structures for this such as April-tags [26], but this method may fail if the camera has low quality or in low-light

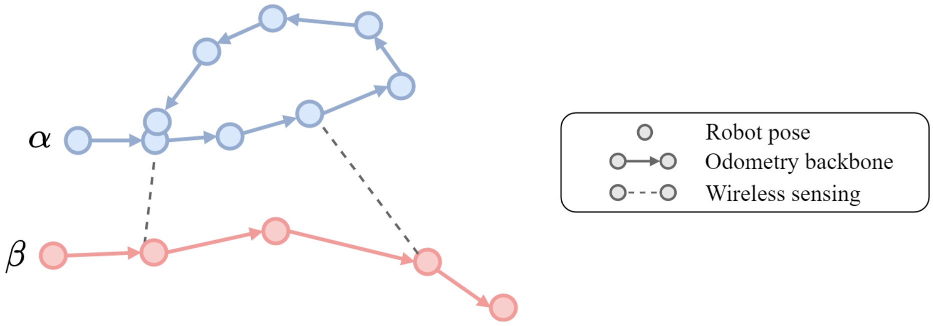


Figure 1.1: Trajectories of two robots α and β, with two wireless measurements depicted by the dotted lines.

conditions. When no cameras are used (e.g. only LiDAR), no standard approach for recognition is available at all. Due to these limitations, this first approach fails to have good performance in many practical applications.

Fortunately, a second less restrictive approach exists to finding inter-robot loop closures. Robots can exchange data on the observed environment and match similar locations through place recognition [9]. Then, the robots are only required to have traversed the same location, not even necessarily at the same time. This method is already mature due to its use for single-robot SLAM, where place recognition is used to correct for drift in the position estimate [9, 24]. Place recognition has two downsides too however, which have been identified as some of the most significant open problems in the SLAM community [3]. First, exchanging models of the environment and performing place recognition introduce a large computation and communication overhead [10]. Secondly, scenes may be accidentally mismatched if they look similar, leading to a failed map estimate - a problem referred to as "perceptual aliasing" [15].

In single-robot SLAM the robot can alleviate these two problems of computation and mismatching using odometry data [5]. Figure 1.1 shows how odometry measurements form the backbone of the trajectories of two robots α and β. The (noisy) odometry backbone shows that the second and last pose of robot α's trajectory are in approximately the same area and are thus promising candidates to do place recognition. However, the figure also shows that this odometry backbone does not connect trajectories of different robots. It is thus impossible to extend this approach directly to multi-robot SLAM. This gap can potentially be bridged by using wireless sensing through the communication signal, which provides information to connect the different trajectories (Figure 1.1). Then, place recognition may be done more efficiently and robustly for multiple robots too, by first identifying where the trajectories of the robots overlap.

1.2. Wireless sensing

Wireless sensing refers to measuring spatial information from the wireless communication signal. The communication signal is an electro-magnetic wave. Measuring time-of-flight (ToF) of the wave gives an estimate of the distance between the two communicating robots, and measuring the angle of arrival (AoA) at the receiving robot gives an estimate of what direction the communication signal is coming from. Wireless sensing is a surprisingly accessible source of information, since it only requires communicating very lightweight "ping packets". This is much easier to obtain than establishing a reliable connection to transfer multiple megabytes of measurement data for elaborate place recognition.

The information obtained from wireless sensing is the relative pose of the robots at that time instant. It is thus conceptually similar to finding inter-robot loop closures by observing, recognizing and localizing each other. However, wireless sensing does not have the same practical restrictions. First, it does not require line-of-sight observations since the communication signal can often pass through obstacles. Second, there is no need for recognition modules since the communication signal already carries identifying information.

There are certain challenges that need to be overcome when using wireless sensing. The AoA measurement has a relatively large standard deviation of 11° in line-of-sight situations and up to 25° when the signal is obstructed [16]. Especially at long distances, this results in a large positional error. Additionally, wireless sensing is subjected to multipath propagation of the wireless signal. Multipath propagation refers to the phenomenon where the signal bounces off of various objects to arrive at the receiver from different angles. Con-

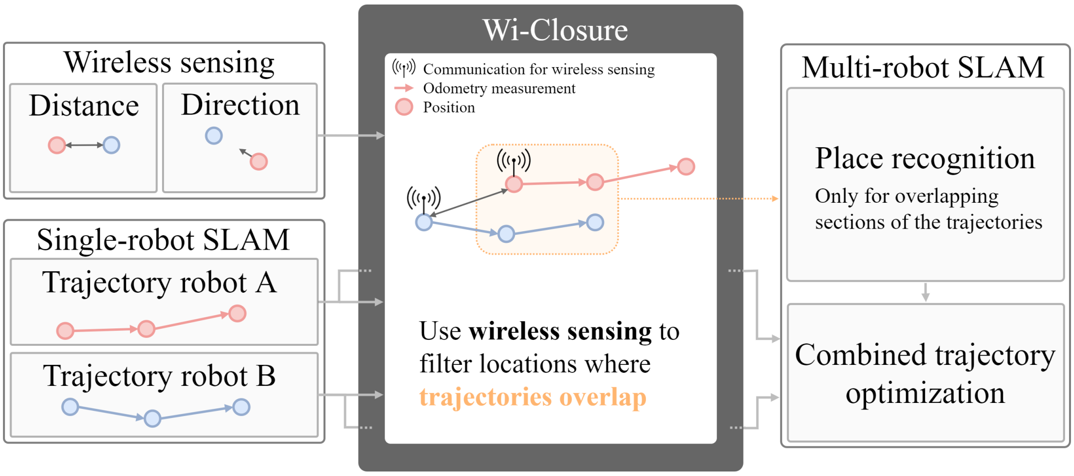

Wireless sensing
Distance
Direction
Single-robot SLAM
Trajectory robot A
Trajectory robot B
Wi-Closure
Communication for wireless sensing
Odometry measurement
Position
Use wireless sensing to filter locations where trajectories overlap
Multi-robot SLAM
Place recognition
Only for overlapping sections of the trajectories
Combined trajectory optimization

Figure 1.2: This figure shows the architecture as proposed in this thesis for an efficient and resilient multi-robot SLAM pipeline. Adding wireless sensing allows us to find trajectory overlap, and only do place recognition and activate the multi-robot SLAM algorithm for these locations.

sequently, the AOA measurement may include multiple directions, of which at most one is directly pointing towards the other robot. Properly addressing these error sources is vital to use wireless sensing in practice.

1.3. Contribution of this thesis

This thesis contributes to more reliable and faster multi-robot SLAM by using wireless sensing to select the locations on which place recognition should be performed, as shown in Figure 1.2. Doing so, wireless sensing may remedy the flaws in place recognition, namely real-time computation and resilience against perceptual aliasing. This approach is fundamentally different from previous works that either used place recognition in multi-robot SLAM, or used wireless sensing in multi-robot SLAM. This thesis thus fills the gap by combining these approaches into a hybrid method, that may be able to combine the strengths of both methods to address important open problems in multi-robot SLAM.

The thesis is structured as follows. The second chapter covers advances in the multi-robot SLAM and wireless sensing literature and identifies gaps in more detail. The third chapter contains the conference paper presenting the approach *Wi-Closure*, designed in this thesis. Then, the fourth chapter provides more background on the approach and results presented in the paper. Lastly, the fifth chapter discusses the implications of this thesis, and reflects on limitations of the approach and avenues for future research.

2
Related literature

The architecture of the multi-robot SLAM pipeline developed in this thesis (Figure 1.2) builds upon work from three different topics: SLAM, multi-robot SLAM and wireless sensing. This chapter highlights relevant previous work for each topic.

2.1. The SLAM framework

SLAM is a collective term for multiple modular algorithms that can be combined to estimate location and map from sensory (camera or LiDAR) measurements. The algorithms are usually divided into the SLAM front-end and back-end. The front-end overlaps with computer vision research to abstract dense information from the high-dimensional sensory input. On the other hand, the SLAM back-end is in charge of using the abstracted information to solve for the robot trajectory and landmark locations.

Regarding the back-end, maximum likelihood estimation (MLE) of the map and location through graph-SLAM has become the de facto SLAM method, replacing its filtering-based predecessors EKF-SLAM and particle filter SLAM [28]. Graph-SLAM instead is an optimization-based approach which depicts the SLAM problem in a factor-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (Figure 2.1). Nodes $\mathcal{V}$ contain the states x, consisting of the robot pose history p and locations of landmarks L. Each edge (i.e. factor) $e \in \mathcal{E}$ is a probability density function $p_e(y_e|x)$, denoting the probability of observing measurement y_e given state vector x [28]. Graph-SLAM solves for states x by maximizing the following MLE problem

$$x_{MLE} = \arg\max_{x \in \mathcal{X}} \prod_{e \in \mathcal{E}} p_e(y_e|x) \tag{2.1}$$

The recent popularity of graph-SLAM stems from several attractive properties. The graph representation is intuitive, enabling researchers to gain a better fundamental understanding of the SLAM problem and develop solutions even in complex (multi-robot) scenarios [28]. For example, researchers have previously shown that graph-SLAM is asymptotically consistent: an increasing amount of measurements will guarantee that the estimate is eventually unbiased and the estimated mean squared error is equal to the true mean squared error. This property is not shared by EKF and particle filter SLAM, which have produced inconsistent estimates in some scenarios as a result of accumulating linearization errors [13]. Other researchers used the

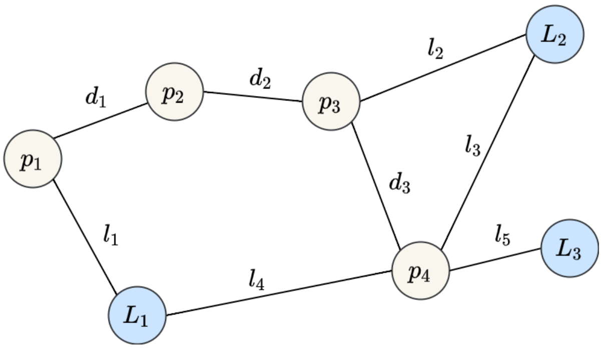


Figure 2.1: Factor graph for robot with pose history p_1 to p_4 and observed landmarks L_1, L_2 and L_3, which form state-vector $x = [p, L]$. Each edge represents an observation y_i (either odometry data d_i or pose transform to a detected landmark l_i), connecting two states with the local conditional likelihood $p_e(y_e, x)$.

graph structure of graph-SLAM to find duality properties that can certify when the MLE solution to the non-linear problem is optimal under mild assumptions [27]. Also, the sparse structure of the factor graph provides options for efficient sparse optimization techniques, resulting in real-time execution of current graph-SLAM algorithms [28]. Lastly, graph-SLAM is especially amenable to the multi-robot setting: its flexible framework readily extends to multiple robots and distributed computation [14], and the complexity scales linearly with the explored area and number of robots rather than quadratically as filtering-based SLAM approaches do [6, 32].

2.2. Problems in multi-robot SLAM

This thesis aims to improve the process of how trajectories of different robots are related to each other, which is central to multi-robot SLAM. The following sections revisit a number of important issues in this field. The first section discusses the importance of taking into account the noise in the local robot trajectory estimates. The second section characterizes to what extend and in which environments computation and scene mismatching remain a bottleneck. Lastly, the final section briefly describes the role of parameter tuning in multi-robot SLAM.

2.2.1. Rigid trajectories

A popular approach in the multi-robot SLAM literature is to assume trajectories (and thus the maps) to be rigid, greatly simplifying the problem to finding a single transformation between each pair of robots. These works directly match maps, e.g. finding sections of gridmaps with similar pixel patterns or similarly shaped Voronoi cells [18, 22, 23, 31].

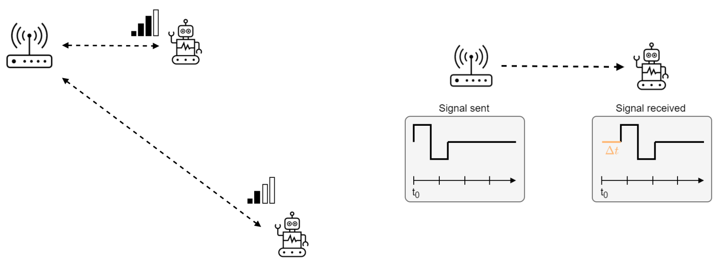
However, due to noise in the odometry measurements, the trajectories drift over time and thus this rigidity assumption becomes problematic with increasing noise or trajectory length. The resulting maps can be significantly distorted such that the representation is far from reality. In the worst-case, robots may be unable to execute tasks that would have been straightforward with a correct map. For example, a robot could infer that all paths to the goal position are blocked due to a distorted map, while in reality an accessible route exists. Inter-robot loop closures are better suited to handle these situations, since they only locally relate different trajectories and thus do not have this rigidity requirement [25]. This opens the opportunity to flexibly correct maps: when a false inter-robot loop closure is included, the mistake can be corrected locally by removing the offending loop closure.

2.2.2. Computation and scene mismatching

Inter-robot loop closures are extracted from sensory data, a task appointed to the front-end of SLAM. These loop closures are found either by directly observing and recognizing the other robot [11, 17], or by exchanging camera or LiDAR measurements and finding places that both robots have visited, i.e. place recognition [12, 37]. Place recognition is fundamentally less restrictive than directly observing each other, but in practice it is subjected to two problems: it is computationally expensive and prone to mismatching of similar-looking scenes, also known as perceptual aliasing.

To reduce computation, a multitude of multi-robot SLAM implementations, such as DiSCO-SLAM [14], Kimera-Multi [37] and others [1, 7, 21, 38], first compress images or pointclouds to lower dimensional descriptors [30]. Matching a Bag-of-Words or ScanContext descriptor instead of the raw measurement data substantially lowers communication and computation overhead [9]. Additionally, efficient data structures such as look-up trees further decrease the time needed to find matching measurements, with the current state-of-the-art having a $\mathcal{O}(log(n))$ time complexity to find a match in a database with n measurements [9]. A downside of compressing measurements is that information is lost, which increases the risk of mismatching scenes.

Scene mismatching results in false inter-robot loop closures, which can lead to catastrophic optimization failure of the SLAM result [15]. Therefore, various techniques exist in the SLAM back-end to reject false loop closures. Mangelson et al. [25] introduce pairwise consistency maximization (PCM). PCM identifies the largest set of inter-robot loop closures that are all consistent with each other, assuming that false inter-robot loop closures lead to inconsistent measurements. Alternatively, Yang et al. [42] use a robust cost-function in the SLAM back-end, such that all outliers receive the same maximum penalty. This approach efficiently integrates outlier-rejection into the optimization problem. Although both methods reach impressive performance against random outliers, their performance deteriorates in repetitive environments.



(a) Determining distance d by measuring radio signal strength (RSS), typically using $RSS = P_0 - c\ \log(d)$ with P_0 transmitted power and c a tuning parameter.

(b) Measuring distance using time delay of a high-bandwidth block signal. Time delay Δt is related to distance d through speed of light c using $d = c\,\Delta t$.

Figure 2.2: Measuring distance from the communication signal.

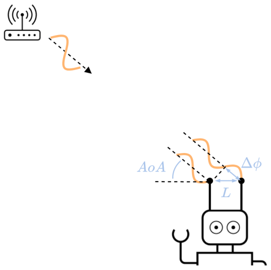


Figure 2.3: Measuring AoA from the communication signal. Phase difference $\Delta\phi$ between two antenna's is measured. By knowing wavelength λ of the signal and distance L between the two antenna's, AoA is computed as $AoA = \cos^{-1}(\lambda\Delta\phi L^{-1})$

In repetitive environments, false inter-robot loop closures found by place recognition are not random. Large numbers of false loop closures can then be consistent, bypassing detection by PCM or robust cost functions. Perceptual aliasing refers to this problem of falsely selecting false loop closures because two distinct locations look too similar. To solve this problem, authors in [39] identify when perceptual aliasing occurs, and keep track of multiple hypotheses that describe the different ways in which the measurements of the robots can be reasonably matched. Unfortunately, as a consequence robots have to reason about the best action to take when multiple hypotheses on the environment could be true, which becomes exponentially more difficult to compute with each added hypothesis [34]. To prevent intractable computation or catastrophic failure in repetitive environments, an important and open question is how to reduce the impact of perceptual aliasing and reliably and efficiently identify the correct hypothesis.

2.2.3. Parameter tuning

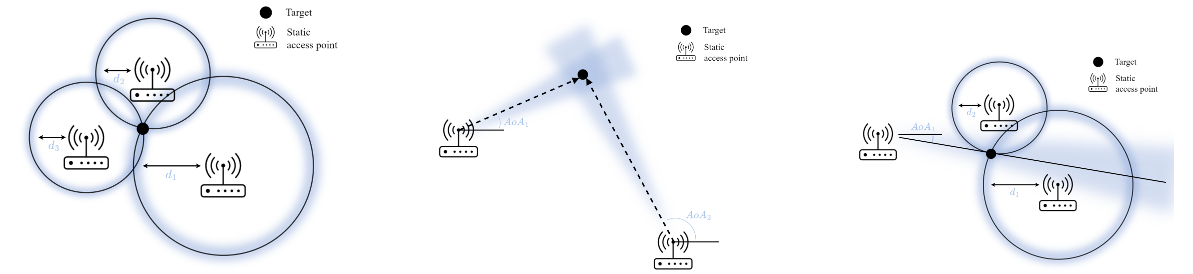
Previous research has shown that various multi-robot SLAM pipelines can perform well in a number of tested environments [14, 21, 38]. However, SLAM algorithms are notorious for the curse of parameter tuning [3]. Parameters can be tuned endlessly, until the multi-robot SLAM algorithm reaches sufficient performance in the tested situations. Unfortunately, the tuned parameters for place recognition do not necessarily generalize well to other untested environments, such that performance is very sensitive to the setting of a few parameters. In fact, some works even have differently tuned parameter sets for tests on different datasets [14].

2.3. Leveraging wireless sensing

Only recently, a limited number of works have leveraged wireless sensing in multi-robot SLAM. The idea of wireless sensing is to exploit the spatial information that the communication signal between robots can provide, as shown in Figures 2.2 and 2.3. The distance between two communicating robots can be measured by time-of-flight (ToF) or radio signal strength (RSS) [29, 35], while the receiving robot can estimate what direction the communication signal is coming from by measuring the angle of arrival (AoA) [16].

Wireless sensing has captured the attention of researchers to improve multi-robot SLAM, by using distance measurements from ultrawideband (UWB) sensors directly as distance constraints in the graph-SLAM optimization problem [2, 8]. This method does not rely on finding scenes with similar appearances like scene recognition, and thus completely solves the problem of perceptual aliasing in repetitive environments. Unfortunately, only using range information results in a difficult to solve nonlinear optimization problem that may be slow to solve. This can be countered by using a multi UWB-tag set-up for each robot [2]. However, to be effective this requires the robots to be large enough to accommodate sufficient distance between the multiple tags ($0.7m$ in the set-up in [2]), limiting the application of this approach to environments with sufficiently wide corridors.

Many works on wireless sensing in multi-robot SLAM directly extend the more extensive literature of wireless sensing in single-robot SLAM, which requires environments with a pre-existing communication infrastructure with so-called access points. When a robot then communicates with these access points, it measures its distance or direction to them. Since the access points are static, multiple wireless measurements over time can be used to counter the drift in the trajectory estimate [4, 43]. Various techniques exist for localization based on wireless sensing, based on AoA, ToF, or RSS measurements or a mix thereof, as illustrated in Figure



(a) Distance localization scheme. At least three static anchors measure distance to target.

(b) AoA localization scheme. Noise becomes more prominent at large distances.

(c) Hybrid localization scheme combining AoA and distance measurements.

Figure 2.4: Localization schemes based on wireless sensing.

2.4. Interestingly, AoA has not been used yet in the context of multi-robot SLAM. It could provide information to speed up the nonlinear optimization problem in [8], possibly enabling a real-time multi-robot SLAM algorithm, while retaining the resilience against perceptual aliasing that wireless sensing in general provides.

3
Conference paper

Wi-Closure: Reliable and Efficient Search of Inter-robot Loop Closures Using Wireless Sensing

Weiying Wang[1], Anne Kemmeren[2], Daniel Son[1], Javier Alonso-Mora[2], Stephanie Gil[1]

***Abstract*—In this paper we propose a novel algorithm, *Wi-Closure*, to improve computational efficiency and robustness of loop closure detection in multi-robot SLAM. Our approach decreases the computational overhead of classical approaches by pruning the search space of potential loop closures, prior to evaluation by a typical multi-robot SLAM pipeline. *Wi-Closure* achieves this by identifying candidates that are spatially close to each other by using sensing over the wireless communication signal between robots, even when they are operating in non-line-of-sight or in remote areas of the environment from one another. We demonstrate the validity of our approach in simulation and hardware experiments. Our results show that using *Wi-closure* greatly reduces computation time, by 54% in simulation and by 77% in hardware compared, with a multi-robot SLAM baseline. Importantly, this is achieved without sacrificing accuracy. Using *Wi-closure* reduces absolute trajectory estimation error by 99% in simulation and 89.2% in hardware experiments. This improvement is due in part to *Wi-Closure*'s ability to avoid catastrophic optimization failure that typically occurs with classical approaches in challenging repetitive environments.**

I. Introduction

Loop closure detection has been widely studied as a fundamental aspect of Simultaneous Localization and Mapping (SLAM) [1], [2]. The location estimate of the robot drifts over time due to the noise in the on-board odometer and loop closure detection is essential to correct for this drift by recognizing previously visited places. Without such corrections, the world as perceived by the robot may diverge substantially from reality. If multiple robots intend to collaborate, they require a shared situational awareness being consistent with reality, as obtained by multi-robot SLAM. The key to obtaining this shared understanding are inter-robot loop closures. Where regular loop closures constrain the positions of one robot itself, the inter-robot loop closure defines spatial relations between pairs of robots. These inter-robot loop closures enable robots to merge local sensor data into a shared model of the world and obtain relative locations.

A common method that robots use to find inter-robot loop closures is place recognition. However, place recognition remains challenging in practice, especially when the environment has repetitive elements [3] and when communication between robots is intermittent. We introduce *Wi-Closure* to address two persistent problems in this setting. First, since robots do not know each other's location, they may mismatch similar-looking scenes that they have encountered in different locations – a problem also referred to as perceptual aliasing [1]. Second, during the short intervals that communication between robots is established, feeding a large set of inter-robot loop closures into the multi-robot SLAM pipeline puts a large strain on computational resources [4]. Repetitive elements further increase computation by falsely recognizing more inter-robot loop closures. Previous work introduced pairwise consistency maximization (PCM) to prevent scene mismatching by identifying false inter-robot loop closures [5]. However, recent research demonstrates that if repetitive elements are present, catastrophic failure of the SLAM algorithm can occur even if using PCM [6]. A more robust solution to perceptual aliasing is tracking all possible (mis)matches, resulting in various hypotheses of what the world looks like [7]. Unfortunately, working with multiple hypotheses is costly since multiple-hypothesis tracking and planning are computationally complex [8]. This makes these methods less viable for real-time execution on commonly available robot hardware.

Our approach *Wi-Closure* is a computationally lightweight method that robustly finds inter-robot loop closures in perceptually aliased environments. We use spatial information from WiFi and ultra-wideband (UWB) communication signals to identify where robots' trajectories are close. WiFi is an electromagnetic wave, and thus the receiving robot can lo-

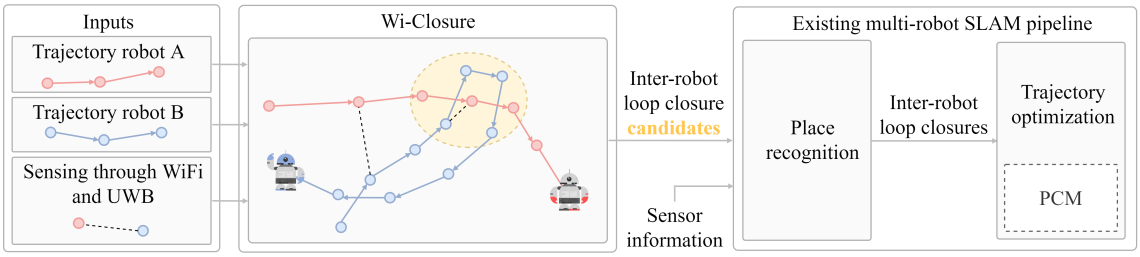

Inputs
Trajectory robot A
Trajectory robot B
Sensing through WiFi and UWB
Wi-Closure
Inter-robot loop closure candidates
Sensor information
Existing multi-robot SLAM pipeline
Place recognition
Inter-robot loop closures
Trajectory optimization
PCM

Fig. 1: *Wi-Closure* efficiently finds locations where robots' trajectories overlap, as indicated by the yellow area. Only inter-robot loop closures at these locations need to be processed by the multi-robot SLAM pipeline. This increases robustness against perceptual aliasing and decreases overall computation of the pipeline.

[1] John A. Paulson School of Engineering and Applied Sciences, Harvard University, Allston, MA 02134, USA

[2] Faculty of Mechanical, Maritime and Materials Engineering, Technical University of Delft, 2628 CD Delft, The Netherlands

cally derive the direction or Angle of Arrival (AOA) to the transmitting robot from the phase information [9]. Similarly, commercial UWB devices measure time-of-flight to estimate distance. Importantly, sensing through the communication signal has wide applicability in this setting since it passes through obstacles and thus works in non line-of-sight situations [10], and it doesn't require the robots to identify each other through vision-based methods, e.g. using Apriltags [11]. Our method solely uses spatial information and thus it works together seamlessly with existing place recognition methods based on appearance. As depicted in Fig. 1, the *Wi-Closure* algorithm is used at the start of the multi-robot SLAM pipeline.

In order to achieve good performance, *Wi-Closure* must also address a major challenge to sensing over the communication signal; namely, it must address multipath propagation of the wireless signal. Multipath refers to the phenomenon where the signal bounces off of various objects to arrive at the receiver from different angles. Consequently, the AOA measurement may include multiple directions, of which at most one is the direct-line path to the other robot. We address this issue with PCM, since only the true direct paths will give consistent pairs of AOA measurements over time. In our hardware experiments, after collecting 4 AOA measurements with in total 3 direct paths and 17 multipaths, we are able to accurately distinguish all direct paths from the multipaths.

Our numerical and hardware experiment results demonstrate that our method efficiently prunes the search space of loop closure candidates by **99%** in simulation and **78.7%** in hardware experiments, and increases robustness against perceptual aliasing by rejecting up front inter-robot loop closures between distinct places and reducing absolute trajectory estimation error by **99%** in simulation and **89.2%** in hardware results. We summarize the contributions of this paper as follows:

1) We introduce a resource efficient approach, *Wi-Closure*, to detect inter-robot loop closures in perceptually aliased environments, based on spatial information from the communication signal. It can work in tandem with existing place recognition methods.
2) We address the challenging situation of multipath propagation of the communication signal with PCM.
3) We demonstrate the merits of our approach in terms of robustness against false inter-robot loop closures and improved computation time in simulation with the KITTI dataset and in hardware experiments.

II. Related Work

For decades, the majority of research on loop closure detection has focused on a single robot [12], [13]. Recently however, loop closure detection algorithms are being adapted to fleets of robots, to ensure reliable and efficient retrieval of shared map and location estimates [5], [14]. We leverage previous work on sensing over the communication signal to simultaneously address two open problems: 1) computation to match large trajectories is high, and 2) place recognition easily mismatches trajectories in repetitive environments.

Wireless sensing Extensive research has shown that we can obtain spatial information from wireless signals [9], [15]. Many works use UWB sensors to obtain ranging information between two robots by measuring the time-of-flight of the ultra-wideband signal. [16], [17] use the ranging information amongst robots to improve the joint position estimate even without being in line of sight of each other. Recently, [10] also introduced sensing direction from the WiFi communication signal to the robotics community, requiring only a single WiFi antenna and movement of the robot. These innovations avoid the need of bulky equipment and anchors as used in classical works to estimate position, which come with additional infrastructure requirements [18].

Range-only SLAM Previously, [19] used UWB sensors in a multi-robot SLAM setting coined range-only SLAM, where distance measurements are directly used as inter-robot loop closures. This avoids the problem of perceptual aliasing, but it only introduces connections between the maps of the robots where the robots are communicating. In realistic scenarios the communication is intermittent, and trajectories can overlap in places where communication is unavailable and where the position estimate is uncertain due to odometer drift. Additional place recognition increases the accuracy of the map by matching these overlapping locations. To our knowledge, we are the first to speed up place recognition using ranging and direction information from the communication signal.

Computation in loop closure Researchers sought to reduce computation of loop closure detection, e.g. with easily obtainable ORB features for vision-based approaches [20], and efficient look-up trees to match scenes [12]. Unfortunately, these methods may result in mismatched maps in perceptually aliased environments [6]. In [21] the authors consider sampling a subset of most informative inter-robot loop closures to reduce overall time consumption. However, the authors also note that the performance guarantee of their sampling method decreases if a scene can be potentially matched to many others - i.e. when there is substantial perceptual aliasing.

Perceptual aliasing Although repetitive scenes are pervasive in many environments, classical place recognition approaches find it notoriously difficult to deal with them. Researchers have focused on simultaneously representing all possible matches as multiple hypotheses in one framework [22]. However, to properly use these multiple hypotheses to determine the best course of action for the robot, we need computationally expensive methods such as data-association belief space planning (DA-BSP) [8], [23]. In DA-BSP the computation time scales exponentially with the hypotheses.
We observe that many methods have a trade-off between robustness against perceptual aliasing and computation: increased robustness requires large computation, while computationally efficient methods decrease robustness or perform worse in repetitive environments. Our approach instead aims to improve both computation and robustness against perceptually aliasing. By sensing lightweight information over the communication signal, we efficiently pinpoint where inter-robot loop closures connect scenes that are likely in the same location.

III. Problem Formulation

Consider a team of robots operating in an unknown environment, unaware of their relative positions to each other. All robots obtain odometry measurements to estimate their trajectories locally. These trajectories are spatially connected through measurements on relative position and orientation of the robots, retrieved by sensing over an intermittent communication signal. Based on the information collected so far, we aim to determine where their trajectories overlap with each other, such that these inter-robot loop closure candidates can be further refined by existing place recognition systems.

We consider a classical graph-SLAM setup of a team of robots denoted by the set Ω. Let two robots be represented by $\alpha, \beta \in \Omega$. Each robot estimates its own trajectory $\mathcal{T}^\alpha$ with respect to its local frame α. A trajectory is defined by a set of homogeneous transformations from time $t = 0$ to τ, as denoted by $\mathcal{T}^\alpha = \{T_t^\alpha : t = 0, \dots, \tau\}$. Here, T_t^α is an element in the Special Euclidean Lie group $T_t^\alpha \in SE(d)$, consisting of a rotation matrix in the Special Orthogonal Lie group $R_t^\alpha \in SO(d)$ and a translation vector $\mathbf{x}_t^\alpha \in \mathbb{R}^d$. Throughout this paper, we adopt the convention of denoting the reference frame as superscript, and the target frame as subscript for T, R and $\mathbf{x}$, i.e. $\mathbf{x}_s^r$ denotes the translation of position s with respect to reference r. We estimate the trajectory $\mathcal{T}$ using the graph-SLAM framework, by maximum-likelihood estimation (MLE) of likelihood function $\mathcal{L}$ given measurement set $\mathcal{Z}$ [24].

$$\hat{\mathcal{T}} = \arg\max_{\mathcal{T}} \mathcal{L} = \arg\max_{\mathcal{T}} \prod_k f_k(z_k|\mathcal{T}) \tag{1}$$

Here, factors $f_k(z_k|\mathcal{T})$ are conditional probability density functions that encode the probability of observing an odometry measurement $z_k \in \mathcal{Z}$, given the pose information in $\mathcal{T}$.

A. Multi-robot SLAM with wireless measurements

We propose to use information from the communication signal between robots to relate their trajectories. UWB and WiFi signals provide measurements on the distance d between robots, and the direction ϕ of the signal-transmitting robot with respect to the signal-receiving robot, respectively. Previously, [25] formulated these AOA and distance measurements as factors to solve a localization problem. We adopt this formulation, with factors $f_{uwb}(d|\mathcal{T}^{\alpha,\beta})$ and $f_{aoa}(\phi|\mathcal{T}^{\alpha,\beta})$.

$$f_{uwb}(d|\mathcal{T}^{\alpha,\beta}) = c_1 \exp\left(\frac{-1}{\sigma^2_{\alpha,\beta}}(d - \|\mathbf{x}_k^p\|_2)^2\right) \tag{2}$$

$$f_{aoa}(\phi|\mathcal{T}^{\alpha,\beta}) = c_2 \exp\left(-\kappa_{\alpha,\beta}\mathbf{u}^T(\phi)\frac{\mathbf{x}_k^p}{\|\mathbf{x}_k^p\|_2}\right) \tag{3}$$

where $c_1 = \frac{1}{\sqrt{2\pi\sigma^2_{\alpha,\beta}}}$, $c_2 = \frac{1}{2\pi I_0(\kappa_{\alpha,\beta})}$ and $\mathbf{u} = [\cos\phi, \sin\phi]^T$. Here, $I_0(.)$ is the modified Bessel function of the first kind of order zero, $\sigma^2_{\alpha,\beta}$ the variance of the distance measurement, and $\kappa_{\alpha,\beta}$ a concentration parameter computed as the inverse of the AOA variance, i.e. $\kappa_{\alpha,\beta} > 10$ rad^2. Then, these factors are combined into one factor.

$$f_{comm}(d, \phi|\mathcal{T}^{\alpha,\beta}) = f_{aoa}(\phi|\mathcal{T}^{\alpha,\beta}) f_{uwb}(d|\mathcal{T}^{\alpha,\beta}) \tag{4}$$

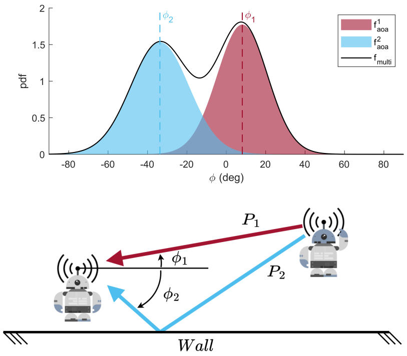


Fig. 2: The communication signal can reach the robot through different paths $\mathcal{P} = \{P_1, P_2\}$, resulting in multiple Gaussian modes f^i_{aoa} in the AOA measurement f_{multi}.

Importantly, [25] models the AOA measurement as a single Gaussian. However, this may not realistically represent the AOA measurement in practice due to multipath propagation of the signal. Objects in the environment can reflect the communication signal, causing it to arrive at the robot via different paths as shown in Fig. 2. These paths cause the AOA measurement to have multiple (approximately Gaussian) modes. For the j^{th} AOA measurement, we parameterize these paths by the set of binary variables $\mathcal{P}^j = \{P_1^j, P_2^j, \dots, P_n^j\}$, where $P_i^j = 1$ indicates that path i in the j^{th} AOA measurement is the direct path. Then, the multimodal AOA measurement f_{multi} can be modeled as a marginalization over multiple f_{aoa}.

$$f_{multi,j}(\phi|\mathcal{T}) = \sum_i^{|\mathcal{P}|} f^i_{aoa,j}(\phi|\mathcal{T}, P_i^j = 1)p(P_i^j = 1) \tag{5}$$

Each AOA measurement has at most one path as the true direct path. Therefore, an important problem that *Wi-Closure* addresses is how to determine the set of direct paths for multiple AOA measurements, which we denote by realization $\mathcal{R} = \{P_i^j \mid P_i^j = 1,\ \forall j\ \sum_{i=1}^n P_i^j \leq 1\}$. Then we can obtain an estimate of what we will refer to as ***the shared robot trajectory*** $\mathcal{T}^{\alpha,\beta}$, by adding communication factors $f^i_{comm,j}$ corresponding to $P_i^j \in \mathcal{R}$ to the MLE in Equation 1.

B. Inter-robot loop closures as a set of nearby poses

With the spatial information contained in $\mathcal{T}^{\alpha,\beta}$, we are interested in retrieving the positions where the trajectories of robot α and robot β are nearby each other. To assess whether some position $\mathbf{x}_p^\alpha \in \mathcal{T}^\alpha$ of robot α is nearby some position $\mathbf{x}_k^\beta \in \mathcal{T}^\beta$ of robot β, we use the Mahalanobis distance.

$$d_{MH}(\mathbf{x}_p^\alpha, \mathbf{x}_k^\beta) = \sqrt{(\mathbf{x}_k^p)^\top\ \Sigma_{p,k}^{-1}\ \mathbf{x}_k^p} \tag{6}$$

The main objective of *Wi-Closure* is then to efficiently find all position-pairs $(\mathbf{x}_p^\alpha, \mathbf{x}_k^\beta)$ that have a Mahalanobis distance smaller than some threshold D, as collected in set G and are thus good loop closure candidates.

$$G = \{(\mathbf{x}_p^\alpha, \mathbf{x}_k^\beta) \mid d_{MH}(\mathbf{x}_p^\alpha, \mathbf{x}_k^\beta) < D,\ \mathcal{R}\} \tag{7}$$

Note that a different set G will be found for different guesses of the direct paths $P_i^j = 1$, i.e. for different realizations $\mathcal{R}$.

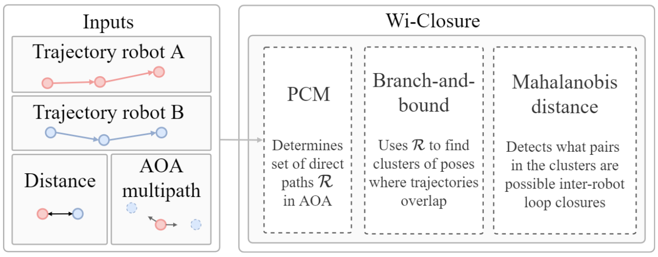

Inputs
Trajectory robot A
Trajectory robot B
Distance
AOA multipath
Wi-Closure
PCM
Determines set of direct paths $\mathcal{R}$ in AOA
Branch-and-bound
Uses $\mathcal{R}$ to find clusters of poses where trajectories overlap
Mahalanobis distance
Detects what pairs in the clusters are possible inter-robot loop closures

Fig. 3: Overview of *Wi-Closure*.

IV. Approach

This section explains the approach taken by *Wi-Closure* on three core aspects, of which an overview is shown in Fig. 3. First, we show how trajectory information and communication measurements combined can give a reliable estimate of the shared trajectory. Here, we reject spurious multipaths in the communication measurements using PCM. Secondly, an algorithm akin the branch-and-bound algorithm quickly finds areas where the trajectories overlap. Lastly, for each position pair in the overlapping areas, we determine whether it is a candidate inter-robot loop closure using the Mahalanobis distance.

A. A shared trajectory estimate while rejecting AOA multipath

Wi-Closure uses as input the robot trajectories $\mathcal{T}^\alpha$ and $\mathcal{T}^\beta$ and communication factors f_{uwb} and f_{multi}. Due to the multipath propagation problem, the AOA measurement determining factor f_{multi} can be multimodal, while only one mode possibly gives useful information on the direct path. In this section we therefore show how PCM finds a set of direct paths, denoted by realization $\mathcal{R}$. Communication factors f_{comm} are then constructed using the paths in $\mathcal{R}$ to connect the robot trajectories while avoiding AOA multipaths.

The PCM method first determines for each pair of measurements whether they are consistent with each other [5]. As shown in Fig. 4, two communication measurements are consistent with each other if we can traverse them and the odometry backbone of the robot trajectories in a loop. Let T_j^i and T_l^k be two transformations defined by some communication factor $f_{comm,1}$ and $f_{comm,2}$ respectively, and define the trajectory sections $T_k^j = (T_j^\alpha)^{-1}(T_k^\alpha)$ and $T_i^l = (T_l^\beta)^{-1}(T_i^\beta)$. Then, if the loop is closed the following equality should hold.

$$T_{loop} = T_j^i\ T_k^j\ T_l^k\ T_i^l = I \tag{8}$$

To account for noise in the transformation estimates, we identify consistent loops using the Mahalanobis distance d_{PCM}. For this we use Lie algebra to express the transformation as a 6D vector $\xi_{loop} \in \mathfrak{se}(3)$ with $\xi_{loop} = \log(T_{loop})$.

$$d_{PCM} = \sqrt{\xi_{loop}^\top \Sigma_{loop}^{-1} \xi_{loop}} \tag{9}$$

where Σ_{loop} is the covariance matrix corresponding to ξ_{loop}. Then, the largest set of communication measurements that are all consistent with each other, gives us a set with likely only measurements of direct paths. Hence we have found realization $\mathcal{R}$ with which we can estimate how the trajectories $\mathcal{T}^\alpha$ and $\mathcal{T}^\beta$ are positioned with respect to each other.

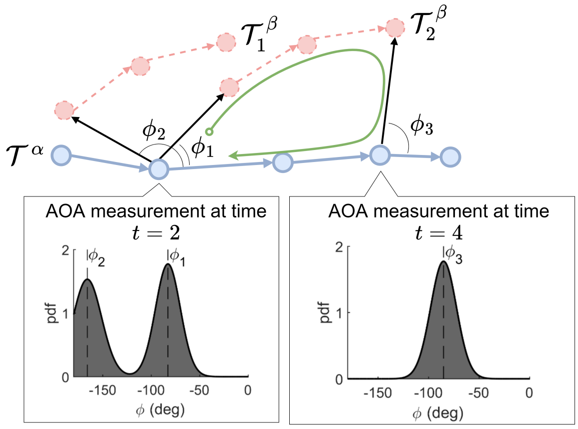


Fig. 4: From robot α's perspective, at time $t = 2$ the trajectory of robot β could be either at $\mathcal{T}_1^\beta$ or $\mathcal{T}_2^\beta$ due to the multipath in the AOA measurement (black arrows). By additionally using the AOA measurement at time $t = 4$, PCM determines that the paths corresponding to ϕ_1 and ϕ_3 are direct paths, since they can form a loop (green arrow). Therefore, $\mathcal{T}_2^\beta$ is robot β's real trajectory.

B. Efficiently finding trajectory overlap

We quickly find clusters where trajectories overlap using a method similar to the classical branch-and-bound algorithm. As shown in Fig.5, our approach first bounds the area's traversed by robots α and β, and selects the poses within this overlap. These poses are divided into smaller clusters, and the process is repeated for each cluster. The initial bounds on the area are found by selecting the minimum and maximum position coordinate along each dimension. However, we need to account for possible distance between true loop closures and uncertainty in the poses. We add d_{buffer} to the bounds, which is computed such that we retain all position pairs that are later included as inter-robot loop closures when computing the Mahalanobis distance.

$$d_{buffer} = D\sigma_{UB} + R_{sensor} \tag{10}$$

where D is the threshold of the Mahalanobis distance used in Equation 7, σ_{UB} is an upper bound to the worst-case uncertainty that we can expect in any direction for any position pair, and R_{sensor} is the range of the sensor that will determine at what distance we can expect to find loop closures.

First, consider the maximum uncertainty σ_{kp}^{max} for the translation $\mathbf{x}_p^k$ between a single position pair, computed as the square root of the largest eigenvalue of Σ_{kp}. We then aim to distributively find σ_{UB} that is an upper bound to σ_{kp}^{max} for any two poses $\mathbf{x}_p^\alpha \in \mathcal{T}^\alpha$ and $\mathbf{x}_k^\beta \in \mathcal{T}^\beta$.

$$\sigma_{UB}^2 \geq \max_{k,p}(\sigma_{kp}^{max})^2 = \max_{k,p} \lambda_{max}(\Sigma_{kp}), \quad p, k \in t \tag{11}$$

where λ_{max} is the largest eigenvalue of Σ_{kp}. Secondly, $\mathbf{x}_k^p$ is rewritten as a pose composition of poses in the local frames.

$$\mathbf{x}_k^p = \ominus \mathbf{x}_p^\alpha \oplus \mathbf{x}_\beta^\alpha \oplus \mathbf{x}_k^\beta \tag{12}$$

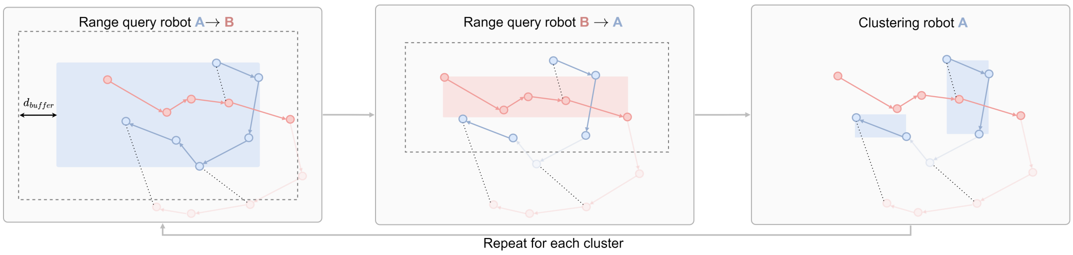


Fig. 5: Finding area's where trajectories overlap by iterative refinement of overlapping bounding boxes.

This allows us to determine an upper bound on $\lambda_{max}(\Sigma_{kp})$.

$$\begin{aligned}\lambda_{max}(\Sigma_{kp}) &\approx \lambda_{max}(\Sigma_{\alpha p} + J_{\alpha\beta}^T\Sigma_{\alpha\beta}J_{\alpha\beta} + J_{\beta k}^T\Sigma_{\beta k}J_{\beta k}) \\ &\leq \lambda_{max}(\Sigma_{\alpha p}) + \lambda_{max}(J_{\alpha\beta}^T\Sigma_{\alpha\beta}J_{\alpha\beta}) \\ &\quad + \lambda_{max}(J_{\beta k}^T\Sigma_{\beta k}J_{\beta k}) = \lambda_{kp}^{UB}\end{aligned}$$

where J is the Jacobian of $\mathbf{x}$. Then, worst-case uncertainty σ_{UB} is distributively computed as the maximum of all variances λ_{kp}^{UB} between any position pair $\mathbf{x}_p^\alpha \in \mathcal{T}^\alpha$ and $\mathbf{x}_k^\beta \in \mathcal{T}^\beta$.

$$\sigma_{UB}^2 = \max_{k,p}(\lambda_{kp}^{UB}) \geq \max_{k,p}(\lambda_{max}(\Sigma_{kp})) = \sigma_{max}^2$$

Note that $\Sigma_{\alpha\beta}$ is computed using the communication factor f_{comm} and can be taken out of the maximization. We need to take the maximum of largest eigenvalues only of covariance matrices $\Sigma_{\alpha p}$ and $\Sigma_{\beta k}$, which are both computed distributively from the MLE trajectory estimates $\mathcal{T}^\alpha$ and $\mathcal{T}^\beta$. The graph-SLAM formulation using factors $f(z|x)$ enables us to retrieve these covariance matrices with a Gaussian approximation.

$$\Sigma = \left(-E_z\left[\frac{\partial^2 \log f(\mathbf{z}|\mathbf{x})}{\partial \mathbf{x}^2}\middle|\mathbf{x}\right]\right)^{-1} \tag{13}$$

C. Identifying inter-robot loop closures

A position pair $(\mathbf{x}_p^\alpha, \mathbf{x}_\beta^k)$ identified by the clustering in the previous section is included into set G as a candidate inter-robot loop closure if the Mahalanobis distance is smaller than D (Equation 7). This requires an estimate of the relative distance and corresponding uncertainty between these two poses, which we extract from our MLE to the shared trajectory estimate. When solving for this MLE, we could include all communication factors corresponding to $P_i^j \in \mathcal{R}$ simultaneously into our optimization problem. However, when connecting trajectories $\mathcal{T}^\alpha$ and $\mathcal{T}^\beta$ through multiple communication factors this is a nonlinear optimization, which also alters the solution to the local trajectory estimates $\mathcal{T}^\alpha$ and $\mathcal{T}^\beta$. Meanwhile, a single communication measurement has a straightforward solution, since this constraint only repositions the trajectories with respect to each other and does not alter the local solutions to $\mathcal{T}^\alpha$ and $\mathcal{T}^\beta$. For each position pair $(\mathbf{x}_p^\alpha, \mathbf{x}_k^\beta)$ we choose one communication measurement connecting the trajectories at positions $\mathbf{x}_{c1}^\alpha$ and $\mathbf{x}_{c2}^\beta$. Then, pose and uncertainty information is propagated from $\mathbf{x}_p^\alpha$ to $\mathbf{x}_k^\beta$.

$$T_k^p = T_{c1}^p T_{c2}^{c1} T_p^{c2} \tag{14}$$

$$\Sigma_{pk} = \Sigma_{pc1} + J_{c1c2}^\top \Sigma_{c1c2} J_{c1c2} + J_{c2k}^\top \Sigma_{c2k} J_{c2k} \tag{15}$$

where J_{ij} is the Jacobian of T_j^i.

For each position pair, our algorithm uses the communication link that results in minimum route length from $\mathbf{x}_p^\alpha$ to $\mathbf{x}_k^\beta$ over the odometry backbone and communication link. The subsequently retrieved values for T_k^p and Σ_{pk} (using equation 13 and 15) are used to compute the Mahalanobis distance $d_{MH}(\mathbf{x}_p^\alpha, \mathbf{x}_k^\beta)$, which determines whether the position pair should be included in set G.

V. Experiments

In this section, We evaluate *Wi-Closure* through simulation and hardware experiments. Our results show that *Wi-Closure* can efficiently and robustly detect loop closures, while processing large trajectories in batches and in repetitive environments. Our approach also successfully handles the multipath phenomenon of the wireless signal in practice.

A. Simulation experiments

Simulations are performed on the KITTI 08 dataset modified by [26], where a trajectory is split into sections to emulate the multiple robot case with trajectory overlap. Since this dataset does not contain measurements from the wireless signal, we simulate these based on the groundtruth (GPS) trajectory. We use a standard deviation of $0.5\ m^2$ for distance and $10\,\text{deg}$ for AOA, based on previous work characterizing these measurements [10]. All comparisons are performed on a desktop computer running an Intel i9 5.2GHz processor in Ubuntu Linux 18.04. We assess the efficacy of *Wi-Closure* by comparing the performance of the multi-robot DiSCo-SLAM pipeline with and without using *Wi-Closure*. The performance is assessed based on average trajectory error (ATE) and the number of correctly and falsely included inter-robot loop closures. To determine which loop closures are true and false, we use a GPS-based groundtruth trajectory to find positions that are at a maximum distance of $35\ m$, such that the LiDAR scans with a range of $30\ m$ overlap for $> 20\%$.

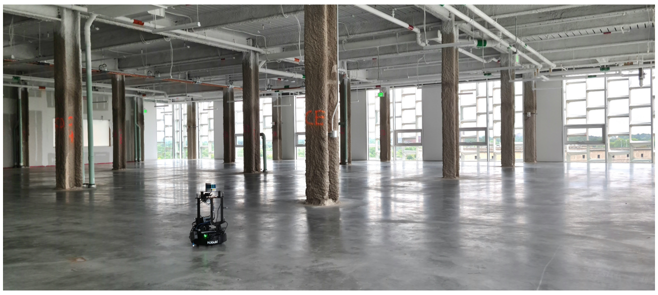


Fig. 6: The $25m \times 23m$ testing field for hardware experiments with highly repetitive features including identical pillars.

Originally, [26] tuned the parameters of the DiSCO-SLAM algorithm such that it has good performance against mismatching on the modified KITTI 08 dataset. However, as we show in our hardware experiments, parameters do not always generalize to other environments. We therefore consider a worse set of parameters in this comparison. Then, we show that while the original DiSCO-SLAM pipeline fails with this parameter set, using the same set of parameters and adding Wi-Closure can still recover good performance.

Table I shows that including *Wi-Closure* in the multi-robot SLAM pipeline results in a lower ATE. Fig. 7 shows that the baseline approach includes too many false loop closures resulting in catastrophic failure. Also, without *Wi-Closure*, DiSCO-SLAM processes all 1,099,101 position pairs as possible loop closures, of which 5544 are true loop closures. Meanwhile, *Wi-Closure* substantially reduces this search space to 7,049 inter-robot loop closures, of which 3,631 are true positive loop closures. This comes at a cost of missing 1,913 potential loop closures. As a result, the whole pipeline takes 896 seconds for the baseline algorithm, whereas adding *Wi-Closure* reduces it to 464 seconds. of which 53 seconds caused by added computation of the *Wi-Closure* module.

B. Hardware experiments

We evaluate our approach on a dataset collected in an unfinished shell space as shown in Fig. 6 with repetitive features. We deploy two customized Locobot PX100, which are installed with a Velodyne VLP-16 LiDAR, a MicroStrain 3DM-GX5-AHRS IMU, DWM1001 UWB, 5dBi Antenna and Intel NUC 10. We process the AOA measurements using the WiFi sensing Toolbox from our earlier work [10]. To accompany the scale of the test field, we limit the range of the LiDAR to 10 meters. For the purpose of computing the ground truth error, we set up 5 UWB nodes in the space to localize the robot in real-time.

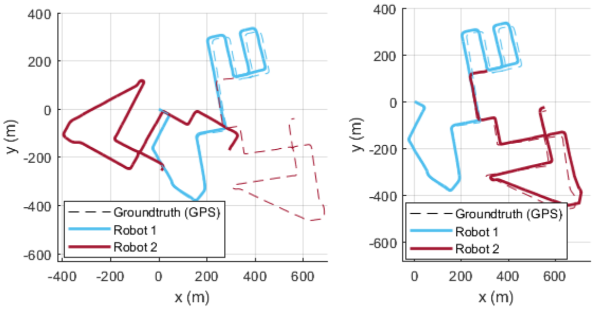


Fig. 7: Simulation results in KITTI 08 dataset. Left: optimized trajectory from Disco-SLAM without *Wi-Closure*. Right: optimized trajectory from Disco-SLAM using *Wi-Closure*.

	Baseline	***Wi-Closure***
ATE (m)	66.1	1.3
Correctly rejected false LC (%)	N/A	99
Missed true LC (%)	0	1
Total Computation Time (s)	896	411
Total Wi-Closure time (s)	N/A	53

TABLE I: Loop Closure (LC) performance comparison between *Wi-Closure* and DiSCO-SLAM in the KITTI Dataset.

Two robots are set up at different locations without knowing each other's frames. They traverse the space collecting LiDAR scans and IMU data. Every 10 meters one robot collects AOA and ranging measurements to the other robot. Trajectories have two rendezvous points for loop closure opportunities. Again, we compare computation time and ATE with and without *Wi-Closure*, and we assess if loop closures are filtered correctly.

We directly apply original DiSCO-SLAM parameters from [26], and show that the original method fails in our environment while adding *Wi-Closure* recovers performance.

As shown in Table II, our approach successfully reduces computation time of the whole SLAM pipeline by 4.3 times and reduces the trajectory error by 89.2%. Fig. 8 shows the optimized trajectories. Because of the repetitiveness of the pillars, the original algorithm fails in the challenging environment. Similar to the simulation results, applying our approach substantially reduces the search space from 1,848 loop closures to only 119 of which 115 are true inter-robot loop closures. Consequently, our method increases speed and prevents failure of the algorithm.

Also, our approach successfully handles the multipath phenomenon in our hardware experiment. Each of four AOA measurements contains five multipath. *Wi-Closure* is able to distinguish all three direct path from the 17 multipath, leading to consistent optimization results as shown in Fig. 8.

VI. Conclusion

In this paper we propose an efficient and robust loop closure finding method *Wi-Closure*, utilizing light-weight information from the wireless signal between robots. We properly handle the multipath phenomenon, and are able to exclude the majority of false loop closures. This drastically reduces processing time of the muli-robot SLAM pipeline and increases the robustness of the results.

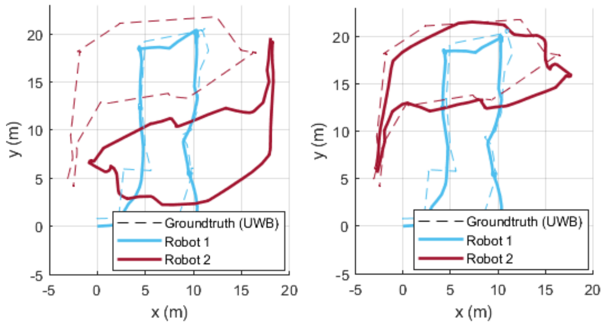


Fig. 8: Hardware experiment results. Left: optimized trajectory from Disco-SLAM without using *Wi-Closure*. Right: optimized trajectory from Disco-SLAM using *Wi-Closure*.

	Baseline	***Wi-Closure***
ATE (m)	17.6	1.9
Correctly rejected false LC (%)	N/A	78.7
Missed true LC (%)	0	15
Total Computation Time (s)	155	36
Total Wi-Closure time (s)	N/A	0.5 seconds

TABLE II: Loop Closure (LC) performance comparison between *Wi-Closure* and DiSCO-SLAM in hardware experiments.

References

[1] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE transactions on robotics*, vol. 24, no. 5, pp. 1027–1037, 2008.

[2] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[3] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 1722–1729.

[4] R. Dubé, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot slam system for 3d lidars," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1004–1011.

[5] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2916–2923.

[6] M. H. Ikram, S. Khaliq, M. L. Anjum, and W. Hussain, "Perceptual aliasing++: Adversarial attack for visual slam front-end and back-end," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4670–4677, 2022.

[7] M. Hsiao and M. Kaess, "Mh-isam2: Multi-hypothesis isam using bayes tree and hypo-tree," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1274–1280.

[8] M. Shienman and V. Indelman, "D2a-bsp: Distilled data association belief space planning with performance guarantees under budget constraints," 05 2022, pp. 11 058–11 065.

[9] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *MobiCom '14*, 2014.

[10] N. Jadhav, W. Wang, D. Zhang, S. Kumar, and S. Gil, "Toolbox release: A wifi-based relative bearing sensor for robotics," 2021. [Online]. Available: https://arxiv.org/abs/2109.12205

[11] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[12] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[13] K. L. Ho and P. Newman, "Loop closure detection in slam by combining visual and spatial appearance," *Robotics and Autonomous Systems*, vol. 54, no. 9, pp. 740–749, 2006, selected papers from the 2nd European Conference on Mobile Robots (ECMR '05). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889006000844

[14] M. Giamou, K. Khosoussi, and J. P. How, "Talk resource-efficiently to me: Optimal communication planning for distributed loop closure detection," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3841–3848.

[15] Y. Song, M. Guan, W. P. Tay, C. L. Law, and C. Wen, "Uwb/lidar fusion for cooperative range-only slam," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6568–6574.

[16] A. Fishberg and J. P. How, "Multi-Agent relative pose estimation with UWB and constrained communications," Mar. 2022.

[17] E. R. Boroson, R. Hewitt, N. Ayanian, and J.-P. de la Croix, "Inter-Robot range measurements in pose graph optimization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 4806–4813.

[18] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, ser. nsdi'13. USA: USENIX Association, 2013, p. 71–84.

[19] E. R. Boroson, R. Hewitt, N. Ayanian, and J.-P. de la Croix, "Inter-robot range measurements in pose graph optimization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4806–4813.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[21] Y. Tian, K. Khosoussi, and J. P. How, "A resource-aware approach to collaborative loop-closure detection with provable performance guarantees," *The International Journal of Robotics Research*, vol. 40, no. 10-11, pp. 1212–1233, 2021. [Online]. Available: https://doi.org/10.1177/0278364920948594

[22] M. Hsiao and M. Kaess, "Mh-isam2: Multi-hypothesis isam using bayes tree and hypo-tree," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1274–1280.

[23] S. Pathak, A. Thomas, A. Feniger, and V. Indelman, "Da-bsp: Towards data association aware belief space planning for robust active perception," in *ECAI*, 2016.

[24] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[25] H. Naseri and V. Koivunen, "A bayesian algorithm for distributed network localization using distance and direction data," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 2, pp. 290–304, 2019.

[26] Y. Huang, T. Shan, F. Chen, and B. Englot, "Disco-slam: Distributed scan context-enabled multi-robot lidar slam with two-stage global-local graph optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1150–1157, 2022.

4

Supporting information

This chapter provides supporting information on the submitted conference paper *"Wi-Closure: Reliable and Efficient Search of Inter-robot Loop Closures Using Wireless Sensing"*. It first elaborates on the used methods and the code implementation, and then delves into practical details about how the results were obtained.

4.1. Methods

4.1.1. Mahalanobis distance with sensor range

Wi-Closure uses the Mahalanobis distance to determine whether two robot positions are an inter-robot loop closure. Consider the situation in Figure 4.1, where the blue robot has estimated the location of the yellow robot with a 2D Gaussian approximation $\mathcal{N}(\Delta\mathbf{p}, \Sigma)$, with $\Sigma = \mathrm{diag}(\sigma_x^2, \sigma_y^2)$. The yellow ellipse denotes the uncertainty ellipse, with principal axes having a length of σ_x and σ_y. Then, the Mahalanobis distance expresses the distance between the two robots in the number of standard deviations.

$$d_{MH} = \sqrt{\Delta\mathbf{p}^T \Sigma^{-1} \Delta\mathbf{p}} = \frac{||\Delta\mathbf{p}||}{\sigma'} \tag{4.1}$$

This metric effectively rejects two positions as inter-robot loop closure when the positions are with certainty a large distance apart (i.e. d_{MH} is larger than some threshold).

However, the metric is less effective if considering that robot sensors (e.g. camera or LiDAR) have certain ranges within which they observe the environment, denoted by r_1 and r_2 respectively. Two positions could lead to inter-robot loop closure whenever the sensor ranges overlap, thus $||\Delta\mathbf{p}|| < r_1 + r_2$. Unfortunately, the current formulation of the Mahalanobis distance could reject these cases if $\sigma' << ||\Delta\mathbf{p}||$, since then the Mahalanobis distance is large. This problem arises whenever the estimated distance $||\Delta\mathbf{p}||$ is small enough for the sensor ranges to overlap, but the uncertainty σ' along the same direction is even smaller.

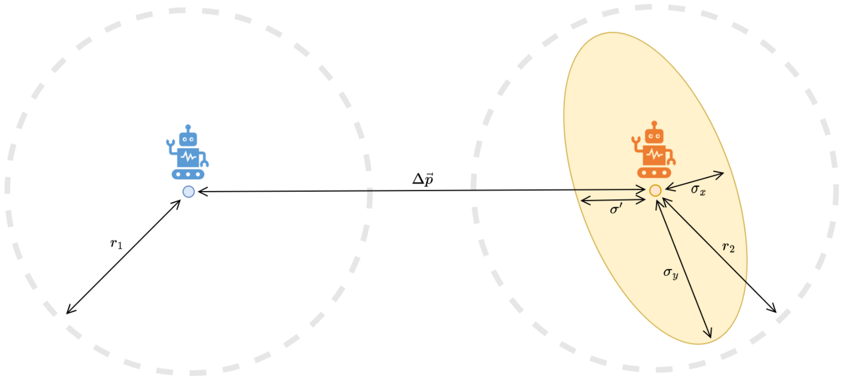


Figure 4.1: Mahalanobis distance $d_{MH} = \sqrt{\Delta\mathbf{p}^T \Sigma^{-1} \Delta\mathbf{p}}$ should be adjusted when considering sensor ranges r_1 and r_2.

The proposed solution used in *Wi-Closure* is to adjust Mahalanobis distance such that $d^*_{MH} = 0$ if $||\Delta\mathbf{p}|| < r_1 + r_2$, which is when the sensor ranges overlap. In the other cases, the Mahalanobis distance is computed over the residual distance after subtraction of the sensor ranges.

$$d^*_{MH} = \begin{cases} \sqrt{\mathbf{q}^T\Sigma^{-1}\mathbf{q}} & \text{if } \mathbf{q}^T\mathbf{u} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

$$\mathbf{q} = \big(||\Delta\mathbf{p}|| - (r_1 + r_2)\big)\,\mathbf{u}, \quad \mathbf{u} = \frac{\Delta\mathbf{p}}{||\Delta\mathbf{p}||} \tag{4.3}$$

4.1.2. Buffer distance in branch-and-bound algorithm

Due to space limitations in section IV-B in the conference paper, a more elaborate derivation of the distance d_{buffer} is provided here. The developed branch-and-bound algorithm rejects two positions indexed at i and j, if their spacing is larger than d_{buffer}. The aim is that in these cases, they also exceed the threshold for the adjusted Mahalanobis distance from section 4.1.1, i.e.

$$||\Delta\mathbf{p}_{ij}|| > d_{buffer} \implies d^*_{MH}(\Delta\mathbf{p}_{ij}) > D \tag{4.4}$$

with D the user parameter determining at what distance the two positions are considered an inter-robot loop closure. Hence, if position pairs are rejected based on the left condition within the branch-and-bound algorithm, they would also be discarded based on the right condition in the later step when computing the Mahalanobis distance. The benefit of using the branch-and-bound approach is a more favorable complexity. For n positions in the trajectories, the branch-and-bound algorithm has a complexity of $\mathcal{O}(r\sqrt{n})$, with $r << n$ the number of iterations before the algorithm converges. Meanwhile, computing all pairwise Mahalanobis distances would result in a complexity of $\mathcal{O}(n^2)$. The Mahalanobis distance will only be computed for positions included by the branch-and-bound algorithm, since this step will discard additional pairs that were missed by the branch-and-bound algorithm. The rest of this section focuses on how d_{buffer} is chosen such that is has the outlined property.

The buffer distance should be larger than the sensor ranges r_1 and r_2. It should also be larger than the worst-case expected Mahalanobis distance. Therefore, d_{buffer} has the following shape, with variable σ_{UB} an upper bound on the standard deviation.

$$d_{buffer} = (r_1 + r_2) + D \cdot \sigma_{UB} \tag{4.5}$$

Upper bound σ_{UB} is subject to certain conditions. Plugging in $||\Delta\mathbf{p}_{ij}|| > d_{buffer}$ into Equation 4.2 gives the condition on σ_{UB} in terms of the following inequality.

$$\mathbf{q} = \big(||\Delta\mathbf{p}|| - (r_1 + r_2)\big)\,\mathbf{u} \;>\; D\,\sigma_{UB}\,\mathbf{u} \tag{4.6}$$

$$d^*_{MH}(\Delta\mathbf{p}_{ij}) = \sqrt{\mathbf{q}^T\Sigma_{ij}^{-1}\mathbf{q}} \tag{4.7}$$

$$> D\,\sigma_{UB}\sqrt{\mathbf{u}^T\Sigma_{ij}^{-1}\mathbf{u}} \tag{4.8}$$

$$= \frac{D\sigma_{UB}}{\sqrt{\rho(\Sigma_{ij})}} \tag{4.9}$$

with $\rho(\cdot)$ the spectral radius. The last equality is a result of the min-max theorem applied to Hermitian matrices, which include covariance matrices [36]. Then, the condition in Equation 4.4 is fulfilled when choosing σ_{UB} as follows.

$$\sigma_{UB} > \sqrt{\rho(\Sigma_{ij})} \quad \forall i,j \implies d^*_{MH} > D \tag{4.10}$$

To allow robots α and β to distributively compute $\rho(\Sigma_{ij})$, the covariance matrix is divided into parts using noise propagation [5].

$$\Sigma_{ij} \approx \Sigma_{i\alpha} + J^T_{\alpha\beta}\Sigma_{\alpha\beta}J_{\alpha\beta} + J^T_{\beta j}\Sigma_{\beta j}J_{\beta j} \tag{4.11}$$

Lastly, using Weyl's inequality an upper bound to the spectral radius can be computed distributively. This finally gives us an expression for σ_{UB} that fulfills all requirements.

$$\max_{i,j}\rho(\Sigma_{ij}) \approx \max_{i,j}\rho(\Sigma_{i\alpha} + J^T_{\alpha\beta}\Sigma_{\alpha\beta}J_{\alpha\beta} + J^T_{\beta j}\Sigma_{\beta j}J_{\beta j}) \tag{4.12}$$

$$\leq \max_{i}\rho(\Sigma_{i\alpha}) + \rho(J^T_{\alpha\beta}\Sigma_{\alpha\beta}J_{\alpha\beta}) + \max_{j}\rho(J^T_{\beta j}\Sigma_{\beta j}J_{\beta j}) \tag{4.13}$$

$$= \sigma^2_{UB} \tag{4.14}$$

4.2. Code implementation

This section describes the code implementation of *Wi-Closure* into the existing *DiSCo-SLAM* code package [14]. The code can be applied in distributed robot networks. It thus does not require a "lead" robot, and each robot executes the same code packages. The code is written in C++ and uses ROS to streamline various inter- and intra-robot communication processes. An overview of the various modules and their most notable interactions is given in Figure 4.2.

4.2.1. High level overview

Each robot uses *LIO-SAM* as local SLAM engine to find a trajectory and map from IMU, LiDAR and (if available) GPS measurements [33]. This package also handles loop closure detection to correct for drift in the local trajectory. Then, if communication is established with another robot, the robots exchange robot trajectories. Simultaneously, the communication signal is used to measure AoA and distance. *Wi-Closure* then processes the trajectory data and AoA and distance measurements to find trajectory overlap, as described by the conference paper in this thesis [40]. Finally, DiSCo-SLAM only processes LiDAR scans filtered by *Wi-Closure* [14]. The LiDAR scans are compressed into lightweight ScanContext descriptors and exchanged with the other robot [20]. A place recognition algorithm then finds inter-robot loop closures. Lastly, a graph-SLAM algorithm merges local robot trajectories and inter-robot loop closures into a shared situational awareness.

4.2.2. LIO-SAM

LIO-SAM is short for Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping [33]. Its goal was creating a low-drift and real-time solution to LiDAR based SLAM that is suitable for multi-sensor fusion. It applies graph-SLAM as explained in Chapter 2.1. Thus, it uses factors to represent the problem of estimating robot trajectory in a factor graph. By assuming Gaussian noise models in the factor graph, the problem reduces to solving a nonlinear least-squares problem. Upon insertion of a new measurement, LIO-SAM efficiently solves this problem through incremental smoothing and mapping with the Bayes tree (iSAM2) [19].

LIO-SAM considers four types of factors: IMU pre-integration factors, LiDAR odometry factors, GPS factors and loop closure factors.

The IMU factor is affected by a drift error due to a slowly varying bias in the IMU measurements. LIO-SAM estimates the bias in the IMU by assuming that angular velocity and acceleration remain constant inbetween two consecutive IMU measurements. Hence, the IMU sensor needs to have a sufficiently high time resolution, taking measurements at least at a rate of 200 Hz.

LiDAR scans require preprocessing prior to computation of the LiDAR odometry factor. LiDAR pointclouds are measured with a rotating laser mechanism, and combined with motion of the LiDAR this distorts the scans. To deskew the scans, IMU gives an initial estimate of the motion. Afterwards, the LiDAR odometry factor is obtained by locally matching the new LiDAR scan with the n most recent scans. Note that not all recent LiDAR scans are used for matching. A LiDAR scan is used only if the robot has moved more than $1m$

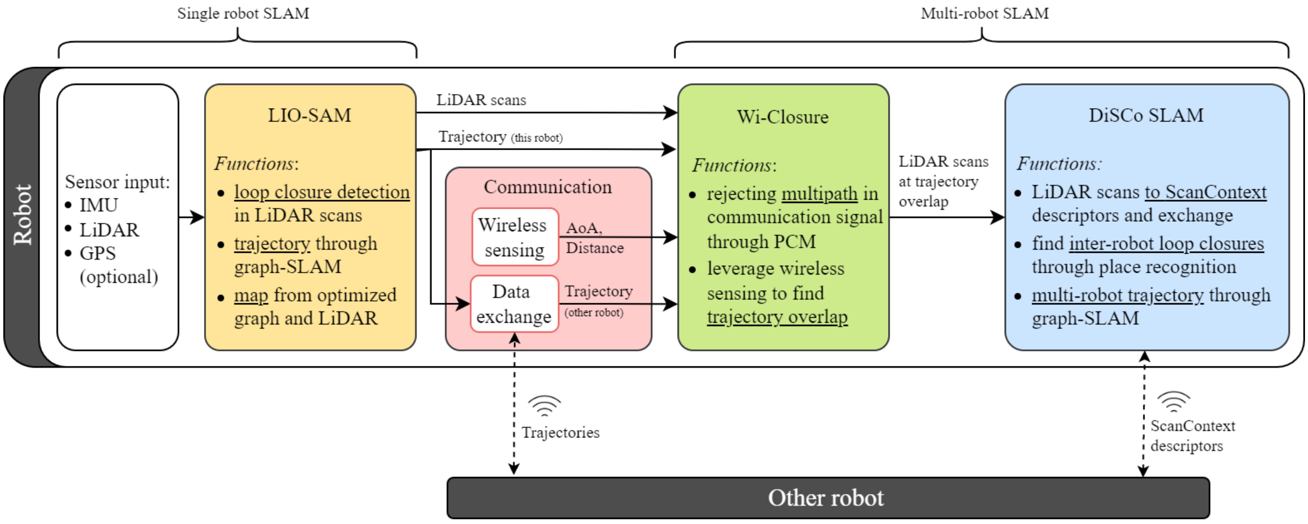

Single robot SLAM
Multi-robot SLAM
Robot
Sensor input:
• IMU
• LiDAR
• GPS (optional)
LIO-SAM
Functions:
• loop closure detection in LiDAR scans
• trajectory through graph-SLAM
• map from optimized graph and LiDAR
LiDAR scans
Trajectory (this robot)
Communication
Wireless sensing
AoA, Distance
Data exchange
Trajectory (other robot)
Wi-Closure
Functions:
• rejecting multipath in communication signal through PCM
• leverage wireless sensing to find trajectory overlap
LiDAR scans at trajectory overlap
DiSCo SLAM
Functions:
• LiDAR scans to ScanContext descriptors and exchange
• find inter-robot loop closures through place recognition
• multi-robot trajectory through graph-SLAM
Trajectories
ScanContext descriptors
Other robot

Figure 4.2: Overview of code modules and the most notable interactions. Each robot has the same set-up, thus supporting a distributed network.

or 10° from its previous position, creating a set of "keyframes". This boosts the computation efficiency and enables real-time execution of the algorithm.

LIO-SAM detects loop closures efficiently by looking for matching scans only within some Euclidean distance of the new LiDAR scan. Because of little drift in the odometry backbone in LIO-SAM, Euclidean distance is a naive but effective approach to find loop closures. For larger drift in the odometry, loop closure detection in LIO-SAM is less effective and may miss matching opportunities since uncertainty should be taken into account, e.g. using Mahalanobis distance instead of Euclidean distance.

4.2.3. DiSCo SLAM

DiSCo-SLAM is a distributed multi-robot SLAM approach based on 3D LiDAR. This approach contributes to improved estimation accuracy and communication efficiency compared to existing approaches [14]. It uses LIO-SAM as SLAM engine for each individual robot, and has an original code implementation to extend this to multi-robot SLAM. Specifically, the authors introduce two major innovations for multi-robot SLAM regarding place recognition and distributed SLAM optimization respectively.

DiSCo SLAM improves the efficiency of place recognition by compressing 3D LiDAR scans into ScanContext descriptors. These ScanContext descriptors are then exchanged and matched. In the new approach presented in this thesis, DiSCo SLAM only processes LiDAR scans if accepted first by Wi-Closure, i.e. if the LiDAR scans were measured in a location close to another robot's trajectory (Figure 4.2).

Every two matching ScanContext descriptors may result in an inter-robot loop closure. After some filtering steps, these are included into a factor graph for optimization of the robot trajectory. In multi-robot SLAM, the factor graph can be divided into sets of intra-robot edges $\mathcal{E}^k_{intra}$ constraining only the trajectory of robot k, and sets of inter-robot loop closures $\mathcal{E}^{mp}_{inter}$ that connect trajectories of two different robots m and p. Consider robots $\alpha, \beta \in \Omega$, then the graph-SLAM optimization objective to obtain the robot trajectories $x_\alpha, x_\beta \in \mathcal{X}$ is split into two parts as follows.

$$\mathcal{X}^{MLE} = \arg\max_{\mathcal{X}} \prod_{k\in\Omega, i\in\mathcal{E}^k_{intra}} p_i(y_i|x_k) \prod_{j\in\mathcal{E}^{\alpha,\beta}_{inter}} p_j(y_j|x_\alpha, x_\beta) \tag{4.15}$$

where $p(y|x)$ are probability density functions denoting the probability of observing measurement y given trajectory x. Generally, these probability density functions assume a Gaussian measurement noise model with covariance matrix Σ and $f(x)$ a nonlinear function.

$$p(x|y) = e^{-\frac{1}{2}||y-f(x)||^2_\Sigma} \tag{4.16}$$

with the norm $||\cdot||_\Sigma$ denoting the Mahalanobis distance. Taking the log on both sides in Equation 4.15 does not change the optimization result, and simplifies the objective to a nonlinear least-squares problem. DiSCo SLAM leverages this formulation to solve the multi-robot SLAM problem distributively in a two-step approach, by alternatingly solving for a local solution of a robot's own trajectory, and collaboratively solving for a global inter-robot solution for the transformation between robot frames.

4.3. Simulation and experiment set-up

4.3.1. User parameter settings

To obtain the simulation and hardware results, some user parameters were changed. It is important to have a sound argument for changing these parameters. After all, many algorithms will work after sufficient parameter tuning, but it is questionable how well performance will be in different environments or even in the same environment but from a different angle. In this thesis, parameters from existing software packages were therefore only changed if there was reason to.

Table 4.1 summarizes the parameters used for place recognition, and gives the settings in the original DiSCo-SLAM code and in the simulation and hardware experiments done for this thesis. The original parameter set from DiSCo-SLAM was tuned to give good matching performance on the KITTI 08 dataset [14]. The hardware experiments purposefully used the original parameters of the baseline DiSCo-SLAM algorithm. Only the `max_range` parameter was changed to reflect the smaller LiDAR range of 10 m rather than 30 m. Notably, in the hardware experiments the baseline DiSCo-SLAM algorithm was not able to find the correct solution despite using the tuned parameter set from the original DiSCo-SLAM paper. This reflects the problem that tuned parameters do not necessarily generalize well to other environments.

DiSCo-SLAM parameters	Original [14]	Simulation	Hardware	Description
`max_range`	30	30	10	Range of LiDAR scan.
`knn_feature_dim`	64	64	64	Number of radial dimensions to keep when compressing LiDAR scans to ScanContext descriptors.
`num_sector`	60	60	60	Number of angular dimensions to keep when compressing LiDAR scans to ScanContext descriptors.
`num_nearest_matches`	50	50	50	Search for this number of ScanContexts closest to input ScanContext descriptor.
`loop_threshold`	0.2	1	0.2	Reject matched ScanContext if distance is larger than this value.
`num_match_candidates`	1	10	1	Choose this number of ScanContext descriptors with smallest distance.
`icp_threshold`	5	5	5	Reject matched pointclouds if residual error after ICP is larger than this value.
`pcm_start_threshold`	5	5	5	Start performing PCM if at least this number of matches were found.
`pcm_threshold`	100	20	100	Two matches are consistent if Mahalanobis distance in PCM algorithm is smaller than this value.

Table 4.1: Parameter settings related to place recognition in the original DiSCo-SLAM code and to the simulation and hardware experiments in this thesis.

The simulations in this thesis used the same KITTI 08 dataset as the DiSCo-SLAM paper, and thus the baseline algorithm works well in simulation when using the original parameter set. However, the purpose of the simulations in this thesis was to assess whether adding *Wi-Closure* could recover good performance, and hence the baseline DiSCo-SLAM algorithm should fail on the KITTI 08 dataset. Therefore, parameters were changed to purposefully make the baseline DiSCo-SLAM algorithm fail, but changes were not random. The original DiSCo-SLAM algorithm matched only 9 inter-robot loop closures, while there are potentially 5544 true inter-robot loop closures within the KITTI 08 dataset. Thus, the rationale for changing parameters was to relax `loop_threshold` and `num_match_candidates` to include more matches, since the current settings led to many missed matching opportunities. The threshold to `pcm_threshold` was lowered to see whether a more strict selection of consistent sets of inter-robot loop closures would recover performance, but this was to no avail. These minor changes in parameters thus rendered the original DiSCo-SLAM algorithm ineffective, but adding *Wi-Closure* could restore performance to similar levels as achieved with the original (well-tuned) parameter set.

4.3.2. Ground-truth positioning with UWB sensor network

Hardware experiments on *Wi-Closure* were conducted in a room with a layout as depicted in Figure 4.3. Since no GPS measurements were available indoors, a network of UWB sensors was set-up to retrieve groundtruth positions of the robot. Five UWB anchors were distributed throughout the room. A laser measurement device provided all pairwise distances between the anchors which were collected in adjacency matrix A. The groundtruth coordinates $\mathbf{p}_i$ of all anchors could then be precisely determined by minimizing error $\epsilon^2 = (||\mathbf{p}_j - \mathbf{p}_k|| - A_{jk})^2$ with a gradient descent algorithm from a good initial guess.

During robot operation, robot location was determined from its range measurements r_i to at least three anchors, using the following linear least-squares solution from Wang et al. [41]. Denote anchor coordinates as $\mathbf{p}_i = (x_i,\ y_i)^T$, robot coordinates as $\mathbf{p} = (x,\ y)$ and the distance measured between robot and anchor as d_i. Then, the equation relating robot and anchor coordinates and distance measurements is

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2 \tag{4.17}$$

One of the anchors is assigned to be reference anchor $\mathbf{p}_r$. Then, after subtracting the nonlinear expression of the reference anchor from all other anchor equations, the resulting expression is linear in robot coordinates (x, y).

$$2(x_i - x_r)x + 2(y_i - y_r)y = d_r^2 - d_i^2 - k_r + k_i \tag{4.18}$$

with $k_i = x_i^2 + y_i^2$, $i = 1, ..., N$ and $N \geq 2$. This is rewritten in matrix form $A\mathbf{p} = \mathbf{b}$ for the distance measurements to all anchors except for the reference anchor, where

$$A = \begin{bmatrix} x_1 - x_r & y_1 - y_r \\ \vdots & \vdots \\ x_N - x_r & y_N - y_r \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} d_r^2 - d_1^2 - k_r + k_1 \\ \vdots \\ d_r^2 - d_N^2 - k_r + k_N \end{bmatrix} \tag{4.19}$$

The least square solution to this linear equation is then given by

$$\mathbf{p}_{LSQ}^{\star} = (A^T A)^{-1} A^T \mathbf{b} \tag{4.20}$$

At each timestep, robot location $\mathbf{p}_{LSQ}^{\star}(t)$ is computed as the groundtruth position of the robot. To remove outliers from these position estimates, a median filter with a moving window over time was used. The window had a width of 10 measurements, corresponding to 0.05 seconds.

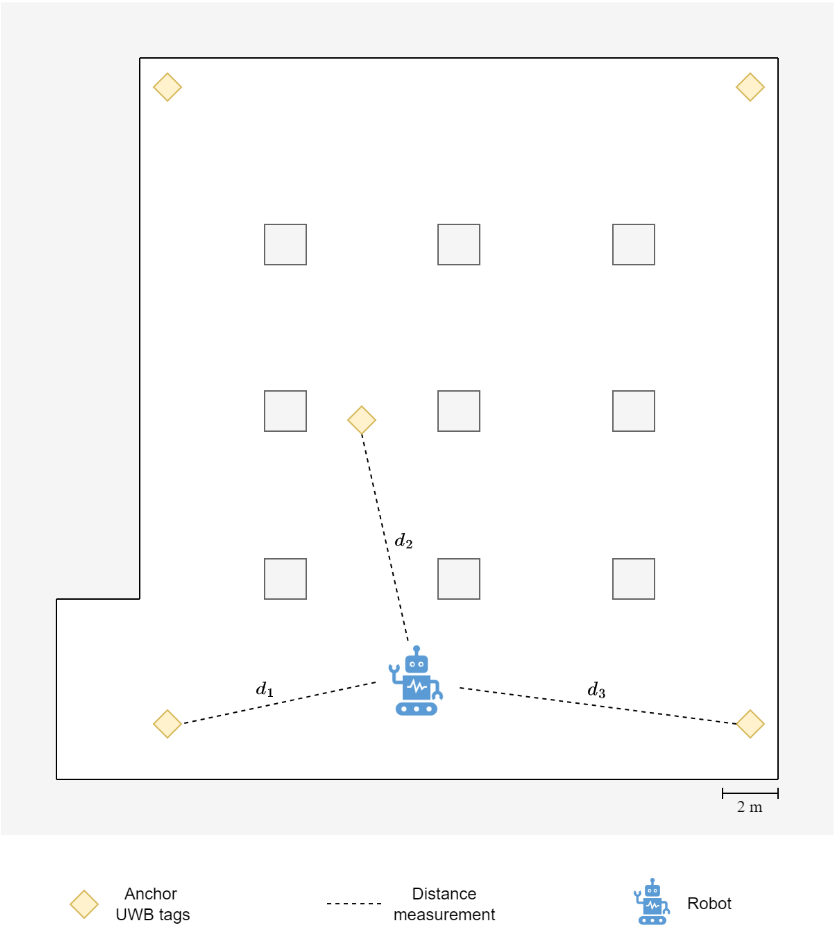


Figure 4.3: Overview of UWB set-up in the space where the hardware experiments were conducted.

5 Conclusion

The outcomes of this thesis have provided insight into the applicability of wireless sensing to support multi-robot SLAM. Simulation studies and hardware experiments on *Wi-Closure* provided promising initial results. This chapter reflects on the interpretation and limitations of the results, and ends with possible avenues for future research.

This thesis demonstrated the benefit of wireless sensing in multi-robot SLAM through the algorithm termed *Wi-Closure*. The three modules of *Wi-Closure* use existing techniques to address the problems in multi-robot SLAM and wireless sensing in a novel way: (1) the multipath problem in the communication signal is addressed with PCM [25]; (2) trajectory overlap is efficiently found with a branch-and-bound type of method; and (3) two positions are only considered to be potential inter-robot loop closures if it cannot be established with certainty that they are a large distance apart using the Mahalanobis distance. Integration of *Wi-Closure* with a state-of-the-art multi-robot SLAM approach, *DiSCo-SLAM*, improved results compared to the baseline of using *DiSCo-SLAM* only. Using *Wi-Closure* decreased total computation time by 54% in simulation and 77% in hardware experiments, without sacrificing accuracy. Also, *Wi-Closure* recovered the correct trajectories with a reduction in absolute trajectory error of 89% and 99% in hardware and simulation, respectively. These results can be explained by *Wi-Closure's* ability to use the communication signal to largely reduce the search space for inter-robot loop closures to the overlapping trajectory only, and reject false, potentially detrimental, inter-robot loop closures up-front. The process is faster because the multi-robot SLAM algorithm processes fewer inter-robot loop closures, and catastrophic optimization failure can be avoided because harmful outliers have already been excluded by *Wi-Closure*.

The implications of these findings are that wireless sensing could play a role to partially solve some significant open problems in multi-robot SLAM. SLAM algorithms generally suffer the curse of parameter tuning, where good performance requires endless tuning of user parameters which often do not generalize well to other environments [3]. As illustrated by the simulation study in this thesis, a poor choice of parameters could lead to inclusion of false inter-robot loop closures and hence catastrophic failure. However, despite using the same poor set of user parameters as for the baseline method, *Wi-Closure* was able to recover performance. Hence, combining wireless sensing and place recognition may make the algorithm perform well on a wider range of parameter settings. This could make the multi-robot SLAM pipeline less sensitive to manual tuning of the parameters. Additionally, as the hardware experiments in this thesis show, repetitive environments are difficult for place recognition even when parameters are optimized [3, 15]. It is thus promising that *Wi-Closure* was able to recover performance in those environments, since it brings the field closer to a reliable multi-robot SLAM pipeline in repetitive environments. Lastly, real-time execution of the multi-robot SLAM algorithms on robots without specialized hardware is a real concern [3, 9, 10]. In this area *Wi-Closure* showed promising results, by at least halving computation time in simulation and hardware experiments.

The generalizability of the results is limited by the few test cases in simulation and hardware so far. One should be especially cautious when interpreting the results on identifying the direct path from the multipath in the communication signal, since multipath propagation is a complex phenomenon that may behave unexpectedly in different environments. Consequently, further research is needed to establish whether results are consistent when hardware experiments take place in other environments. In general, thorough characterisation of the dependency of multipath propagation on the environment and movements of the robot would increase confidence that wireless sensing can be useful in robotics. Some phenomena, such as reflection, are known to induce multipath, but these do not fully explain the behaviour of multipath measurements in practice. Therefore, more research is warranted to determine which environmental factors influence the quality of wireless sensing, such as the number of multipaths and the strength of the direct path signal.

Also, the approach in this thesis relies on multiple measurements from the communication signal over time to reject multipaths in the AoA measurement. However, in some situations robots communicate with each other at only a single instance, and thus not enough wireless measurements are available to use PCM to detect the direct path and reject the multipaths. Future research should investigate whether wireless sensing could still be beneficial in the setting of only a single wireless measurement and multipath propagation. This will require a different approach than *Wi-Closure*.

Currently, *Wi-Closure* needs synchronous distance and AoA measurements from UWB sensors and WiFi signals, respectively. However, the rate of these two measurements can be vastly different. UWB sensors can easily work at a frequency of 200 Hz, but AoA is measured at only up to 2 Hz when a single or dual antenna set-up is used. AoA sensing is much slower because the robots need to collect multiple phase measurements over a small trajectory to determine AoA. A different reason that UWB and WiFi measurements are not necessarily received at the same time is that they can have different ranges. For example, a robot could receive a WiFi signal to determine AoA, while the UWB signal has been attenuated too much to obtain a usable distance measurement. Avenues for future research therefore include whether *Wi-Closure* can be improved to handle asynchronous distance and orientation measurements.

Wi-Closure assumes that trajectories overlap as soon as two positions are nearby, independent of the orientation. This is a good assumption when using 360° LiDAR measurements, since orientation does not influence at what distance the LiDAR scans overlap. However, this is different for directional cameras. Views then only capture the environment in front of the robot. It could thus be interesting to make the approach more amenable to directional camera views as well, by including the effect of orientation on when camera views may overlap.

Other researchers could use the approach in this thesis to improve their multi-robot SLAM pipelines if *Wi-Closure* were written to a publishable and portable package. This would be a step closer to realizing reliable and efficient multi-robot SLAM in practice.

Lastly, a main focus in the SLAM field is to make single- and multi-robot SLAM algorithms "real-time". However, it is inherent to the SLAM problem that there is a breaking point at which all SLAM algorithms are not real-time anymore. One can quickly infer from the graph-SLAM objective that when covering more ground with a robot, more factors are included into the problem, thus making it more difficult to solve. Little effort has been made to characterize breaking points of the algorithms; rather, current research is focused on proving that their algorithm performs real-time and better than the state-of-the-art on some tested datasets. But there is little discussion whether these datasets collected with specific (often high-end) hardware reflect practical problems. Therefore, an open question remains: when are improvements enough such that SLAM algorithms are ready for use in practice? There has been an exceptional effort to standardize comparisons for SLAM algorithms for example on the KITTI benchmark datasets and identify which algorithm works best. As an avenue of future research, perhaps this effort could be extended to include stress-test datasets that are more challenging than most situations that SLAM should be able to solve in practice. For example, situations in very large environments with many features, or measurements taken with cheap noisy hardware. This would help to identify breaking points of each SLAM algorithm. Then, the focus could move from simply "which SLAM algorithm is best on this dataset" to characterising in which practical applications a SLAM algorithm can perform well, and in which situations it will break.

Bibliography

[1] F. Bonin-Font and A. Burguera. Towards multi-robot visual graph-slam for autonomous marine vehicles. *Journal of Marine Science and Engineering*, 8(6), 2020. ISSN 2077-1312. doi: 10.3390/jmse8060437.

[2] E. R. Boroson, R. Hewitt, N. Ayanian, and J.-P. de la Croix. Inter-robot range measurements in pose graph optimization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4806–4813, 2020. doi: 10.1109/IROS45743.2020.9341227.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. doi: 10.1109/TRO.2016.2624754.

[4] Y. Cao, C. Yang, R. Li, A. Knoll, and G. Beltrame. Accurate position tracking with a single uwb anchor. pages 2344–2350. Institute of Electrical and Electronics Engineers Inc., 2020. doi: 10.1109/ICRA40945.2020.9197345.

[5] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss. OverlapNet: Loop closing for LiDAR-based SLAM. In *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, jul 2020. doi: 10.15607/rss.2020.xvi.009.

[6] E. Eade, P. Fong, and M. E. Munich. Monocular graph slam with complexity reduction. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3017–3024, 2010. doi: 10.1109/IROS.2010.5649205.

[7] R. Egodagamage and M. Tuceryan. Distributed monocular visual slam as a basis for a collaborative augmented reality framework. 71:113–123, 2018. doi: 10.1016/j.cag.2018.01.002.

[8] A. Fishberg and J. P. How. Multi-agent relative pose estimation with uwb and constrained communications. *ArXiv*, abs/2203.11004, 2022. doi: 10.48550/ARXIV.2203.11004.

[9] D. Galvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. doi: 10.1109/TRO.2012.2197158.

[10] M. Giamou, K. Khosoussi, and J. P. How. Talk resource-efficiently to me: Optimal communication planning for distributed loop closure detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3841–3848, 2018. doi: 10.1109/ICRA.2018.8460783.

[11] K. Hattori, E. Homma, T. Kagawa, M. Otani, N. Tatebe, Y. Owada, L. Shan, K. Temma, and K. Hamaguchi. Generalized measuring-worm algorithm: high-accuracy mapping and movement via cooperating swarm robots. *Artificial Life and Robotics*, 21(4):451 – 459, 2016. doi: 10.1007/s10015-016-0301-x.

[12] L. Hou, R. Li, S. Cao, and Z. Deng. Combination of multiple robots slam. pages 7243–7248. Institute of Electrical and Electronics Engineers Inc., 2020. doi: 10.1109/CAC51589.2020.9327888.

[13] S. Huang and G. Dissanayake. A critique of current developments in simultaneous localization and mapping. *International Journal of Advanced Robotic Systems*, 13, 10 2016. doi: 10.1177/1729881416669482.

[14] Y. Huang, T. Shan, F. Chen, and B. Englot. Disco-slam: Distributed scan context-enabled multi-robot lidar slam with two-stage global-local graph optimization. *IEEE Robotics and Automation Letters*, 7(2): 1150–1157, 2022. doi: 10.1109/LRA.2021.3138156.

[15] M. H. Ikram, S. Khaliq, M. L. Anjum, and W. Hussain. Perceptual aliasing++: Adversarial attack for visual slam front-end and back-end. *IEEE Robotics and Automation Letters*, 7(2):4670–4677, 2022. doi: 10.1109/LRA.2022.3150031.

[16] N. Jadhav, W. Wang, D. Zhang, S. Kumar, and S. Gil. Toolbox release: A wifi-based relative bearing sensor for robotics. 2021. doi: 10.48550/ARXIV.2109.12205.

[17] Y. Jang, C. Oh, Y. Lee, and H. Kim. Multirobot collaborative monocular slam utilizing rendezvous. *IEEE Transactions on Robotics*, 37(5):1469–1486, 2021. doi: 10.1109/TRO.2021.3058502.

[18] J. Jessup, S. Givigi, and A. Beaulieu. Merging of octree based 3d occupancy grid maps. pages 371–377. IEEE Computer Society, 2014. doi: 10.1109/SysCon.2014.6819283.

[19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012. doi: 10.1177/0278364911430419.

[20] G. Kim, B. Park, and A. Kim. 1-day learning, 1-year localization: Long-term lidar localization using scan context image. *IEEE Robotics and Automation Letters*, 4(2):1948–1955, April 2019. doi: 10.1109/LRA.2019.2897340.

[21] P. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame. Door-slam: Distributed, online, and outlier resilient slam for robotic teams. *IEEE Robotics and Automation Letters*, 5(2):1656–1663, 2020. doi: 10.1109/LRA.2020.2967681.

[22] H. Li and F. Nashashibi. A new method for occupancy grid maps merging: Application to multi-vehicle cooperative local mapping and moving object detection in outdoor environment. pages 632–637, 2012. doi: 10.1109/ICARCV.2012.6485231.

[23] J. Liang, Z. Chen, Q. Lv, H. Wei, and M. Manzhen. Vision feature extraction algorithm for occupancy grid maps merging. pages 290–293. Association for Computing Machinery, 2017. doi: 10.1145/3158233.3159373.

[24] P. C. Lusk and J. P. How. Global data association for slam with 3d grassmannian manifold objects. *arXiv*, 2022. doi: 10.48550/ARXIV.2205.08556.

[25] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan. Pairwise consistent measurement set maximization for robust multi-robot map merging. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2916–2923, 2018. doi: 10.1109/ICRA.2018.8460217.

[26] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE, May 2011.

[27] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, 38 (2-3):95–125, 2019. doi: 10.1177/0278364918784361.

[28] D. M. Rosen, K. J. Doherty, A. T. Espinoza, and J. J. Leonard. Advances in inference and representation for simultaneous localization and mapping. *CoRR*, abs/2103.05041, 2021.

[29] T. J. Rouphael. Chapter 4 - high-level requirements and link budget analysis. In T. J. Rouphael, editor, *RF and Digital Signal Processing for Software-Defined Radio*, pages 87–122. Newnes, Burlington, 2009. ISBN 978-0-7506-8210-7. doi: https://doi.org/10.1016/B978-0-7506-8210-7.00004-7.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. doi: 10.1109/ICCV.2011.6126544.

[31] S. Saeedi, L. Paull, M. Trentini, M. Seto, and H. Li. Efficient map merging using a probabilistic generalized voronoi diagram. pages 4419–4424, 2012. doi: 10.1109/IROS.2012.6386001.

[32] A. Schmidt. Multi-robot, ekf-based visual slam system. pages 562–569, 09 2014. ISBN 9783319113302. doi: 10.1007/978-3-319-11331-9_67.

[33] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE, 2020.

[34] M. Shienman and V. Indelman. D2a-bsp: Distilled data association belief space planning with performance guarantees under budget constraints. pages 11058–11065, 05 2022. doi: 10.1109/ICRA46639.2022.9811984.

[35] P. Tarrio, A. M. Bernardos, J. A. Besada, and J. R. Casar. A new positioning technique for rss-based localization based on a weighted least squares estimator. In *2008 IEEE International Symposium on Wireless Communication Systems*, pages 633–637, 2008. doi: 10.1109/ISWCS.2008.4726133.

[36] G. Teschl. Mathematical methods in quantum mechanics: With applications to schrödinger operators. American Mathematical Society, 01 2009. ISBN 978-1-4704-1888-5.

[37] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone. Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. 2021. doi: 10.48550/ARXIV.2106.14386.

[38] D. Van Opdenbosch and E. Steinbach. Collaborative visual slam using compressed feature exchange. *IEEE Robotics and Automation Letters*, 4(1):57–64, 2019. doi: 10.1109/LRA.2018.2878920.

[39] J. Wang and B. J. Englot. Robust exploration with multiple hypothesis data association. pages 3537–3544, 2018. doi: 10.1109/IROS.2018.8593753.

[40] W. Wang, A. Kemmeren, D. Son, J. Alonso-Mora, and S. Gil. Wi-closure: Reliable and efficient search of inter-robot loop closures using wireless sensing. *arXiv*, 2022. doi: 10.48550/ARXIV.2210.01320.

[41] Y. Wang. Linear least squares localization in sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2015, 12 2015. doi: 10.1186/s13638-015-0298-1.

[42] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134, 2020. doi: 10.1109/LRA.2020.2965893.

[43] H. Zhou, Z. Yao, Z. Zhang, P. Liu, and M. Lu. An online multi-robot slam system based on lidar/uwb fusion. *IEEE Sensors Journal*, 22(3):2530–2542, 2022. doi: 10.1109/JSEN.2021.3136929.