Titre: Title:	Resource-Efficient Decentralized Collaborative Simultaneous Localization And Mapping
Auteur: Author:	Pierre-Yves Lajoie
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Lajoie, P.-Y. (2024). Resource-Efficient Decentralized Collaborative Simultaneous Localization And Mapping [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/61248/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/61248/
Directeurs de recherche: Advisors:	Giovanni Beltrame
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Resource-Efficient Decentralized Collaborative Simultaneous Localization And Mapping

PIERRE-YVES LAJOIE
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Décembre 2024

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

**Resource-Efficient Decentralized
Collaborative Simultaneous Localization And Mapping**

présentée par **Pierre-Yves LAJOIE**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Tarek OULD-BACHIR, président
Giovanni BELTRAME, membre et directeur de recherche
Guillaume-Alexandre BILODEAU, membre
Sajad SAEEDI, membre externe

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Giovanni Beltrame, and the MISTLab team for their unwavering support, guidance, and encouragement throughout my PhD journey. Your collaboration, enthusiasm, and willingness to share knowledge have made this research experience truly enriching.

Thank you to everyone who helped build robots, conduct experiments, and contribute to insightful discussions—your dedication has made all the difference. Special thanks to Vivek Shankar Varadharajan, Karthik Soma, Haechan Mark Bong, Alice Lemieux-Bourque, Rongge Zhang, and Seyed Ehsan Marjani Bajestani, along with all the members who offered feedback, co-authored papers, or helped troubleshoot even the smallest issues. Your contributions were invaluable to the success of this work. A heartfelt thanks as well to my friend and collaborator, Christophe Bédard, for your input, support, and humor.

I would also like to express my sincere gratitude to the Samsung AI Center team, particularly Bobak Hamed Baghi, Sachini Herath, Saria Al Laham, Xue Liu, and Gregory Dudek, for their trust, expertise, and advice. Your mentorship and support gave me the confidence to tackle challenges I once thought were beyond my reach.

To Benjamin Ramtoula, thank you for your friendship and the warm welcome during my stay in Oxford. Your sharp mind and humor were a great help, and I am grateful for the meaningful conversations we shared throughout this journey. I also want to extend my thanks to Daniele De Martini for your insights, advice, and countless ideas during our research meetings.

Lastly, I would like to thank my close family and friends, whose love and support have been a constant source of strength. You stood by me through the ups and downs, and your encouragement carried me forward during the most challenging moments. To my wonderful partner, thank you for your patience and for being by my side during this exciting yet demanding time—I couldn’t have done this without you.

RÉSUMÉ

La robotique collaborative possède un potentiel considérable pour transformer la société et de multiples secteurs en permettant aux systèmes autonomes d'exécuter des tâches difficiles, désagréables, ou tout simplement impossibles pour l'humain. À l'origine de ces capacités se trouve la Localisation et Cartographie Simultanées Collaborative (mieux connue sous le nom Collaborative Simultaneous Localization And Mapping ou son acronym C-SLAM), qui permet à plusieurs robots de cartographier leur environnement et de se localiser, que l'environnement soit préalablement connu ou non, à l'intérieur ou l'extérieur, sous terre ou sous-marin, sur Terre ou sur Mars. Les cartes ainsi générées servent à explorer et inspecter des espaces inaccessibles aux humains, à analyser la traversabilité des terrains, ou encore à planifier et exécuter des tâches complexes avec plusieurs agents.

Cette thèse apporte des contributions pour faire progresser le domaine du C-SLAM, en mettant l'accent sur l'amélioration de la précision, de la robustesse, de l'efficacité et de l'adaptabilité. Elle débute par une revue exhaustive de l'état de l'art, mettant en lumière les nombreux défis. À partir de ces observations, plusieurs cadriciels innovants sont proposés: Swarm-SLAM, un système C-SLAM décentralisé, optimisé pour les réseaux robotiques ad hoc; MOLD-SLAM, exploitant des larges modèles d'apprentissage pour fusionner des cartes avec peu de chevauchement entre elles; et PEOPLEx, qui applique les principes du SLAM pour améliorer la localisation des piétons. En complément, une technique de calibration de domaine auto-supervisée est proposée, visant à améliorer les performances du SLAM dans divers environnements sans nécessiter de réglages manuels.

Les approches développées ont été évaluées lors d'expérimentations sur le terrain, incluant des essais sur un terrain analogue planétaire. Les résultats démontrent des améliorations en termes de précision de localisation, avec des réductions notables des besoins en mémoire, en calcul et en communication. Ces avancées rapprochent de la réalité le déploiement d'essaims robotiques à grande échelle, ouvrant la voie à de nombreuses nouvelles applications.

La thèse se conclut par une réflexion sur les implications plus larges de ces contributions et des perspectives sur l'avenir du C-SLAM. Elle souligne l'importance de développer des approches résilientes et économes en ressources et insiste sur la nécessité d'encourager la reproductibilité scientifique afin de stimuler le progrès du domaine. À mesure que les systèmes autonomes évoluent, le C-SLAM jouera un rôle central en permettant aux robots de collaborer efficacement et de fonctionner de manière indépendante, favorisant ainsi des innovations à travers de nombreux secteurs industriels et sociétaux.

ABSTRACT

Collaborative robotics has immense potential to transform society and multiple sectors by enabling autonomous systems to perform tasks that are difficult, unpleasant, or simply impossible for humans. At the core of this capability lies Collaborative Simultaneous Localization and Mapping (C-SLAM), which empowers multiple robots to collectively map and localize themselves within any space, whether known or unknown, indoor or outdoor, underground or underwater, on Earth or on Mars. The generated maps can be used to explore and inspect spaces unaccessible to humans, to analyse and infer terrain traversability, or to plan and execute complex multi-agent tasks.

This thesis introduces several contributions to advance the field of collaborative SLAM (C-SLAM), with a focus on accuracy, robustness, resource efficiency, adaptability, and calibration. It begins with a comprehensive survey of the state of the art, identifying critical challenges and open research questions in C-SLAM. Building on these insights, a self-supervised domain calibration technique is introduced to enhance SLAM performance across diverse environments without requiring manual tuning. Swarm-SLAM is presented as a decentralized C-SLAM system designed for ad-hoc robotic networks. MOLD-SLAM leverages 3D foundation models to address the challenge of limited map overlap between collaborating robots in C-SLAM. Finally, PEOPLEx is introduced as a framework that applies SLAM principles to improve pedestrian positioning using consumer-grade hardware.

The proposed techniques were thoroughly evaluated through both dataset and real-world field experiments, including trials on a planetary analogue terrain. The results demonstrate significant improvements in localization accuracy and resource efficiency, with notable reductions in memory, computational, and communication overhead. These advancements bring large-scale collaborative robot swarms closer to practical deployment, unlocking new opportunities for real-world applications.

The thesis concludes by discussing the broader implications of these contributions and offering an outlook on the future of C-SLAM research. It highlights the importance of resilient and resource-efficient approaches, as well as the need for open-source implementation and reproducibility to foster scientific progress. As autonomous systems continue to evolve, C-SLAM will be pivotal in enabling robots to operate collaboratively and independently, driving innovations across many industrial and societal domains.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

SLAM	Simultaneous Localization And Mapping
C-SLAM	Collaborative SLAM
GPS	Global Positioning System
ATE	Average Translation Error
LiDAR	Light Detection and Ranging
UWB	Ultra-Wide Band
BLE	Bluetooth Low-Energy
AR	Augmented Reality
VR	Virutal Reality
WP	Work Package

CHAPTER 1 INTRODUCTION

This thesis is written as part of the requirements for a Philosophiæ Doctor degree in Computer Engineering at Polytechnique Montréal. It presents research work that was conducted within the MIST Laboratory of Polytechnique Montréal (Montréal, Québec, Canada), the Samsung AI Center (Montréal, Québec, Canada), and the Mobile Robotics Group of Oxford University (Oxford, United Kingdom) between September 2020 and November 2024. The structure of the document is that of a thesis by articles comprising six published or submitted papers, including a peer-reviewed literature review (Chapter 2), four original scientific contributions (Chapters 5, 6, 8 and 9) and one field study (Chapter 7).

1.1 Context and Motivation

Large-scale adoption of robotics could revolutionize industries and improve daily life, but realizing this promise hinges on robots performing tasks autonomously and reliably for extended periods of time. Achieving true autonomy demands not only sophisticated hardware but also precise and robust environmental perception. In particular, accurate localization—knowing the robot's position—and mapping—building a usable representation of the surroundings—are fundamental for ensuring that robots can navigate safely, make informed decisions, and accomplish their objectives. Just as humans depend on tools like maps, compasses, or GPS to find their way through unfamiliar territories, robots require similar technologies to operate effectively and efficiently.

At the heart of robotic autonomy lies Simultaneous Localization and Mapping (SLAM), a key algorithm that allows a robot to build and update a map of its environment while simultaneously tracking its own position within it. SLAM empowers robots to operate independently in dynamic and unpredictable settings by incrementally refining the map as the robot explores, ensuring it can avoid obstacles, detect changes, and adjust its path. While positioning systems such as GPS or external tracking systems (e.g. using cameras or Ultra-Wideband (UWB) anchors) provide useful positioning in specific environments, they are often unsuitable solutions. Indeed, tracking cameras or UWB anchors require extensive infrastructure that must be installed and maintained, while GPS signals degrade or become unavailable in environments like underground tunnels, dense urban centers, underwater environments, indoor spaces with poor satellite reception, or on the Moon and Mars. In these scenarios, SLAM becomes essential, providing the localization accuracy needed for safe and effective autonomous operation.

In multi-robot systems, the challenges of localization become even more complex. Robots operating together must not only navigate their environments independently but also align their individual perception with that of their counterparts. At the start of a mission, each robot typically relies solely on its own local map, generated from the limited perspective of its onboard sensors. Without a shared frame of reference or initial knowledge of each other's positions, robots cannot easily exchange meaningful information about obstacles, goals, or resources. This lack of shared situational awareness limits the effectiveness of collaborative efforts, reducing the system's overall potential.

This is where Collaborative SLAM (C-SLAM) emerges as a crucial enabler of multi-robot autonomy. C-SLAM provides a framework for map merging and shared localization, allowing each robot to contribute its local map to a collective, global representation. As robots align their positions within a unified map, they unlock new opportunities for cooperation—such as multi-agent task planning, resource sharing, distributed exploration, or collaborative transport of heavy objects. With a shared situational awareness, robots can coordinate their efforts seamlessly, minimizing redundant actions and maximizing efficiency, even in challenging environments.

The societal implications of SLAM and C-SLAM go far beyond robotic research. These technologies pave the way for future applications in areas like autonomous planetary exploration missions, precision agriculture, smart warehouses, and infrastructure inspection. Whether enabling fleets of drones to scan underground caves or allowing robots to autonomously explore underwater ecosystems, SLAM and C-SLAM are cornerstones of the next generation of robotics. They not only enhance the individual capabilities of robots but also unlock the full potential of collaborative systems, driving the realization of robotics' broader promise: a safer, smarter, and more connected world.

1.2 Problem Statement

Despite extensive research efforts, several persistent challenges remain in the fields of SLAM and C-SLAM. While single-robot SLAM has made significant strides, it still struggles to achieve the robustness and resilience required for deployment beyond controlled laboratory settings [1]. Many SLAM systems are vulnerable to sensor noise, and unpredictable error sources encountered in real-world applications. Achieving real-time performance on resource-constrained devices, such as small drones, augmented reality (AR) glasses, or mobile robots, presents another challenge [2]. Moreover, SLAM systems often require extensive tuning and calibration [3], creating significant barriers to entry for non-expert users looking to deploy these systems efficiently.

The challenges become even more pronounced in multi-robot collaborative SLAM (C-SLAM), where additional constraints emerge from the need for inter-robot communication and map merging. Exchanging large 3D environmental representations demands high communication bandwidth, while substantial computational resources are required to detect overlapping regions between individual robot maps and merge them into a coherent, consistent global map. Furthermore, the synchronization of maps across multiple agents introduces complexities that are not present in single-robot SLAM, compounding the difficulties of ensuring seamless collaboration across diverse operating environments.

In this thesis, we adopt the pose graph formulation for both SLAM and C-SLAM [4]. Pose graph SLAM provides a structured, cohesive, and computationally efficient solution by framing localization and mapping as a maximum likelihood estimation problem over the robot's poses and inter-robot measurements. This approach offers several key advantages: it reduces computational complexity by marginalizing map features into inter-pose measurements, minimizing the overhead of full map optimization. In contrast to filtering methods, pose graph SLAM continuously refines past estimates, leading to more accurate maps over time and enhanced long-term reliability [1]. Additionally, pose graph SLAM allows for outlier rejection by re-evaluating and removing erroneous measurements—a critical capability for mitigating errors caused by data association failures and perceptual aliasing, both of which are common challenges in SLAM [5].

Successfully implementing C-SLAM requires an intricate balance of three core resources: compute, memory, and communication bandwidth. These trade-offs are heavily influenced by the type of sensors used (e.g., cameras, LiDAR, IMUs), the scale of the mission (e.g., number of robots, trajectory length, and duration), and the deployment environment. Critical factors include the extractability of stable and distinctive map features, the degree of overlap between individual maps, and the alignment between the data collected in the field and the learned models used for SLAM processing. The interplay between these factors can significantly impact the system's accuracy and efficiency.

This thesis explores innovative solutions to advance pose graph C-SLAM along three principal research axes: accuracy and robustness; adaptability; and resource efficiency. Particular attention is given to minimizing computational, memory, and communication overhead to enable real-time performance on resource-constrained devices. Additionally, the work addresses practical deployment challenges, such as parameter tuning and system calibration, to facilitate smoother adoption in real-world applications.

1.3 Research Objectives

In response to the identified challenges, this thesis addresses three primary research objectives, each divided into targeted subobjectives:

- **Accuracy and robustness:** Minimize the average pose error of SLAM and C-SLAM solutions:
    - **Acc1** Develop a decentralized C-SLAM framework that eliminates single points of failure and maintains functionality during network disruptions;
    - **Acc2** Reduce spurious measurements and data association errors throughout the mapping process;
    - **Acc3** Perform uncertainty estimation to optimize map measurements effectively and enhance solution reliability.
- **Resource efficiency:** Minimize the computational, memory, and communication resources required to operate SLAM and C-SLAM onboard robots:
    - **Res1** Lower the communication bandwidth needed for map convergence without compromising accuracy;
    - **Res2** Define explicit communication and computational budgets to optimize localization accuracy within the resource constraints;
    - **Res3** Expand the scalability of C-SLAM to support larger multi-robot teams.
    - **Res4** Evaluate and document the resources needed, and the respective trade-offs, of current state-of-the-art techniques;
    - **Res5** Reduce the map size in memory without sacrificing localization accuracy;
    - **Res6** Design specialized solvers that provide equivalent accuracy to general-purpose alternatives but with reduced computational complexity.
- **Adaptability:** Ensure C-SLAM systems perform reliably across a wide range of environments and operational contexts:
    - **Adapt1** Build C-SLAM systems capable of functioning across diverse environments and comprehensively document their performance in different settings;
    - **Adapt2** Achieve successful registration between maps with minimal overlap, enabling map merging in a broad range of scenarios;

  - **Adapt3** Perform domain adaptation of deep learning modules used in SLAM when there is a domain discrepancy between the training data and the deployment environment;
  - **Adapt4** Develop SLAM techniques incorporating radio signals, such as UWB, WiFi or BLE, when available in the environment.

1.4 Novelty and Impact

This thesis contributes six articles, published or under review in high-impact, peer-reviewed journals and conferences. Each article addresses specific subobjectives outlined above:

- **Peer-Reviewed Literature Review:** A structured review of current C-SLAM research, identifying key methodologies, challenges, and future research directions. This work synthesizes existing research while highlighting new areas for innovation, guiding future developments in the field.
- **Original Scientific Contributions:** These include three published articles and one currently under review. Each contribution tackles important challenges and objectives for SLAM and/or C-SLAM:
  - **Self-Supervised Domain Calibration:** Focused on adapting SLAM systems to new environments, this method, presented in Chapter 5, offers uncertainty estimation and domain adaptation, making SLAM systems more robust and practical for real-world deployments. We provide our open-source implementation: `https://github.com/MISTLab/vpr-calibration-and-uncertainty`. This research aims to fulfill objectives Acc2, Acc3, and Adapt3;
  - **Swarm-SLAM:** Presented in Chapter 6, this work introduces a decentralized C-SLAM framework with a sparsification strategy to efficiently map environments under limited communication bandwidth. Our implementation is made available publicly to support reproducibility: `https://github.com/MISTLab/Swarm-SLAM`. This project addresses objectives Acc1, Res1, Res2, Res3, and Adapt1;
  - **MOLD-SLAM:** In Chapter 8, we detail a novel approach leveraging 3D foundation models to merge multi-robot maps with varying viewpoints and low-overlap trajectories. The objectives addressed in this work include Acc2, Acc3, Res5, Res6, and Adapt2;
  - **PEOPLEx:** A pedestrian positioning framework, presented in Chapter 9, leveraging diverse sensor inputs, such as IMU, UWB, BLE, and Wi-Fi, to create reliable

SLAM solutions for commercial smartphones. This project targets the completion of objectives Res6, and Adapt4.

- **Field Experiment Study:** This study, presented in Chapter 7 and currently under review, documents the deployment of Swarm-SLAM in a planetary analogue environment. It evaluates its performance under realistic conditions, identifies practical limitations, and outlines future research challenges to guide further development. It also introduces a novel dataset, gathered during our experiments, including inter-robot latency and throughput estimation along with sensor data for mapping. This dataset will allow researchers to develop techniques better tailored to the actual capabilities of current robotic networks. This field study tackles objectives Acc1, Res2, Res4, and Adapt1. The dataset is available publicly: `https://github.com/MISTLab/Mars_Analogue_CSLAM_Dataset`.

Overall, the contributions of this thesis span new calibration methods, optimization formulations, and complete system implementations. Our research outputs include open-source code and datasets, facilitating further development in the robotics community. Swarm-SLAM, in particular, has garnered significant attention, receiving over 400 stars on its public repository and serving as the foundation for further research [6].

Through these contributions, I aim to advance the state of SLAM and C-SLAM research, offering practical solutions that address real-world deployment challenges. I hope that this work will not only bridge gaps in current technology but also inspire future developments that push the boundaries of autonomous robotics.

The following sections present the peer-reviewed literature review, followed by an addendum covering recent advancements that have emerged since its publication. Subsequently, Chapter 4 provides further details on the research approach and methodologies used in this thesis.

CHAPTER 2 ARTICLE 1 : TOWARDS COLLABORATIVE SIMULTANEOUS LOCALIZATION AND MAPPING: A SURVEY OF THE CURRENT RESEARCH LANDSCAPE

Preface: This literature review provides an in-depth analysis of the field of Collaborative Simultaneous Localization and Mapping (Collaborative SLAM), examining its evolution, key challenges, and emerging trends. By synthesizing existing research, the review highlights the technological advancements that have shaped Collaborative SLAM, addressing topics such as resource efficiency, loop closure detection, and pose graph optimization.
The review has undergone a rigorous peer-review process and has been published in the journal Field Robotics. Previously a Green Open Access independent journal, Field Robotics has been integrated into IEEE under the name Transactions on Field Robotics.
Contributions: For this paper, my contributions include conducting a comprehensive literature search to identify a large set of potentially relevant papers, identifying subthemes, and broadly categorizing the papers according to these themes. I then assigned some themes and their corresponding sets of papers to my coauthors, while I personally took responsibility for most of the themes. After reviewing the papers and extracting key takeaways from the relevant literature, we compiled our notes and discussed how to organize the paper. I wrote the majority of the sections and reviewed and edited those I delegated to my coauthors. I was also responsible for writing the discussion and conclusion sections, summarizing both my insights and those of my coauthors gathered throughout the process.

Full Citation: Pierre-Yves Lajoie, Benjamin Ramtoula, Fang Wu, Giovanni Beltrame, “Towards Collaborative Simultaneous Localization and Mapping: a Survey of the Current Research Landscape,” *Field Robotics*, 2022.
Submission date: August 7th 2021
Publication date: May 31st 2022
DOI: 10.55417/fr.2022032

2.1 Abstract

Motivated by the tremendous progress we witnessed in recent years, this paper presents a survey of the scientific literature on the topic of Collaborative Simultaneous Localization and Mapping (C-SLAM), also known as multi-robot SLAM. With fleets of self-driving cars on the horizon and the rise of multi-robot systems in industrial applications, we believe that

Collaborative SLAM will soon become a cornerstone of future robotic applications. In this survey, we introduce the basic concepts of C-SLAM and present a thorough literature review. We also outline the major challenges and limitations of C-SLAM in terms of robustness, communication, and resource management. We conclude by exploring the area's current trends and promising research avenues.

2.2 Introduction

Collaborative Simultaneous Localization and Mapping (C-SLAM), also known as multi-robot SLAM, has been studied extensively with early techniques dating back as far as the early 2000s (e.g. [7–11]). These techniques were introduced only a short time after the inception of single-robot SLAM by researchers who were already envisioning collaborative perception of the environment. Although they were small-scale proofs of concept, they laid the foundations that still shape the field nowadays.

After years of confinement to laboratory settings, C-SLAM technologies are finally coming to fruition into industry applications, ranging from warehouse management to fleets of self-driving cars. Those long awaited success stories are a strong indicator that C-SLAM technologies are poised to permeate other fields such as marine exploration [12, 13], cooperative object transportation [14], or search and rescue operations [15, 16].

SLAM is the current method of choice to enable autonomous navigation, especially in unknown and GPS-denied environments. SLAM provides an accurate representation of the robot surroundings which can in turn enable autonomous control and decision making. Similarly, in multi-robot systems, C-SLAM enables collaborative behaviors by building a collective representation of the environment and a shared situational awareness.

Moreover, many ambitious applications remain for multi-robot systems, such as the exploration of other planets [17, 18]. To reach those moonshot goals, ongoing trends in the research community aim to push the boundaries of multi-robot systems towards increasingly larger teams, or swarms of robots [19, 20], which potentially allow parallel operations that are more efficient and versatile. However, this is still largely uncharted territory since current multi-robot applications either involve very few robots or rely upon large amounts of centralized computation in server clusters. Current C-SLAM techniques are no exception. They are prone to deteriorated performance when the team size increases above a few robots, and could be infeasible when minimal or no prior information is available about the operating environment.

Even though C-SLAM-enabled swarms of robots are still far from reality, C-SLAM remains a useful tool when operating as few as two autonomous robots. In exploration and mapping applications, even small teams can yield a significant boost in performance compared to a single robot system [21]. Notably, autonomous mapping using C-SLAM has recently received a lot of attention due to the latest DARPA Subterranean Challenge [22] and its potential applications in space technologies [23].

Thus, this paper presents a survey of the relevant literature on the topic of C-SLAM, aiming to give a complete overview of the main concepts, current developments, open challenges, and new trends in the field. We hope it will help new as well as established researchers to evaluate the state-of-the-art and offer valuable insights to guide future design choices and research directions. Compared to previous reviews [24, 25], this paper provides an update on the tremendous progress in the past five years. In particular, we delve into the major advances towards the deployment of complete C-SLAM systems outside closely monitored laboratory environments, and we address the specific challenges of the different submodules (i.e., front-end, back-end, etc.). We also focus on the emergent trends and new opportunities coming from adjacent fields of research (e.g. deep learning, edge computing, etc.). This paper aims for a broader overview of the field than surveys covering specific C-SLAM subproblems such as map merging [26], practical implementations [27], particle filter techniques [28], vision-based techniques [29], and search and rescue applications [30].

2.2.1 Outline

The rest of this paper consists of seven sections covering the main C-SLAM subfields of research presented in Table 2.1: Section 2.3 presents an overview of the single robot SLAM problem; Section 2.4 explains the core differences with C-SLAM; Section 6.5 explores the different modules of the C-SLAM front-end and their challenges; Section 2.6 introduces the C-SLAM back-end and discusses the different inference techniques; Section 2.7 looks into important system-level challenges. Section 2.8 discusses the available benchmarking datasets; Section 2.9 presents open problems and ongoing trends in the fields; and Section 8.6 concludes the survey and discusses future research avenues.

2.3 What is SLAM?

At its core, SLAM is a joint estimation of a robot's state and a model of its surrounding environment, with the key assumption that a moving robot performs the data collection sequentially. On one hand, the robot's state comprises its pose (position and orientation) and possibly other quantities such as sensors' calibration parameters. On the other hand,

SLAM	Odometry Intra-Robot Loop Closures Pose Estimation
C-SLAM Front-End	Direct Inter-Robot Loop Closures Indirect Inter-Robot Loop Closures Heterogeneous Sensing
C-SLAM Back-End	Extended Kalman Filters Particle Filters Pose Graph Optimization Perceptual Aliasing Mitigation
System-Level Challenges	Map Representation Communication Constraints
Open Problems	Resilient Inter-Robot Communication Managing Limited Computation Resources Adapting to Dynamic Environments Active C-SLAM Semantic C-SLAM Augmented Reality

Table 2.1 Collaborative Simultaneous Localization and Mapping Subfields of Research

the environment model (i.e., the *map*) consists of representations of landmarks, built with processed data from the robot's exteroceptive sensors such as cameras or lidars. This makes SLAM an essential part of many applications that require building an accurate map of the surrounding environment, whether it be for collision-free navigation, scene understanding, or visual inspection by a remote human operator. Since dead-reckoning approaches (e.g. IMU, wheel or visual odometry) drift over time due to noise accumulation, the environment map in SLAM is also used internally to correct the robot trajectory when known areas are re-visited. The recovered links between previously visited locations are called loop closures. SLAM is useful when neither an a priori map nor localization information are available, when a map needs to be built, or long-term accurate localization estimates are required. Common scenarios include robotics applications without external positioning systems, such as the exploration of unknown indoor environments, caves, mines, or other planets.

2.3.1 Single-robot SLAM problem

Formally, the overall goal of SLAM is to maximize the posterior of the map and robot state. We can formulate this with the state variables X of both the landmarks (map) and the robot, and the set of measurements Z acquired by the moving robot [31]:

$$p(X|Z) \tag{2.1}$$

This estimation problem is solved by either updating the current state at each time step given the new observations (i.e., filtering) or optimizing over the whole trajectory and past observations (i.e., smoothing).

Although filtering in SLAM is still an active research topic, current state-of-the-art techniques are mostly based on smoothing [1, 32]. The common formulation for smoothing techniques is a *Maximum A Posteriori* (*MAP*) estimation problem that leverages the moving robot assumption by introducing a prior distribution (e.g. obtained by odometry) over the robot trajectory.

Thus, the SLAM problem for a single robot, designated with the lower case letter α, can be expressed as finding X^*_α, the solution of the *MAP* problem:

$$X^*_\alpha \doteq \underset{X_\alpha}{\operatorname{argmax}}\, p(X_\alpha|Z_\alpha) = \underset{X_\alpha}{\operatorname{argmax}}\, p(Z_\alpha|X_\alpha)p(X_\alpha) \tag{2.2}$$

The decomposition of the posterior distribution is obtained with Bayes' theorem: $p(Z_\alpha|X_\alpha)$ is the likelihood of the measurements Z_α given a certain X_α, and $p(X_\alpha)$ is the prior distribution of the robot motion state. Intuitively, the SLAM problem finds the set of state variables (environment landmarks and robot poses) X^*_α that is most likely to produce the measurements Z_α given a prior estimation $p(X_\alpha)$.

It is important to also note that SLAM is closely related to the well-studied problem of bundle adjustment in *Structure from Motion* for which we refer the reader to [33].

2.3.2 SLAM Systems Architecture

SLAM systems are commonly divided into a front-end and a back-end, each involving different fields of research. The front-end is in charge of perception-related tasks, such as feature extraction and data association which are both related to fields such as computer vision and signal processing. The back-end produces the final state estimates using the front-end's outputs. The back-end uses tools from the fields of optimization, probability theory and graph theory. In practice, the front-end processes the sensor data to generate ego-motion, loop closure, and landmark measurements, while the back-end performs the joint estimation of the map and the robot state. Figure 2.1 provides an overview of a common SLAM structure in which the robot trajectory is represented as a graph of poses at consecutive discrete times (i.e., a pose graph) and the map as a set of observed landmarks [1]. In a 3D pose graph, the nodes are the robot poses $[\mathbf{R}, \mathbf{t}] \in \mathrm{SE}(3)$ comprised of a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and a translation $\mathbf{t} \in \mathbb{R}^3$, and the edges represent the relative measurements between the poses [34].

Single-robot SLAM still faces many challenges that consequently apply to C-SLAM such as its long-term use, its robustness to perception failures and incorrect estimates, or its need for performance guarantees [1]. To circumvent those limitations in their specific settings, SLAM and C-SLAM developers often have to adapt the architecture and consider some trade-offs between the sensors capabilities, the onboard computing power, and available memory.

2.4 What is Collaborative SLAM?

Many tasks can be performed faster and more efficiently by using multiple robots instead of a single one. Whether SLAM is used to provide state estimation to support an application (e.g. estimate each robot's position to plan for actions), or whether it is at the core of the task (e.g. mapping an environment), it is beneficial and sometimes necessary to extend SLAM solutions into coordinated C-SLAM algorithms rather than performing single-robot SLAM on each robot.

C-SLAM algorithms aim to combine data collected on each individual robot into globally consistent estimates of a common map and of each robot's state. This coordination allows each robot to benefit from experience of the full team, leading to more accurate localization and mapping than multiple instances of single-robot SLAM. However, this coordination introduces many new features and challenges inherent to multi-robot systems.

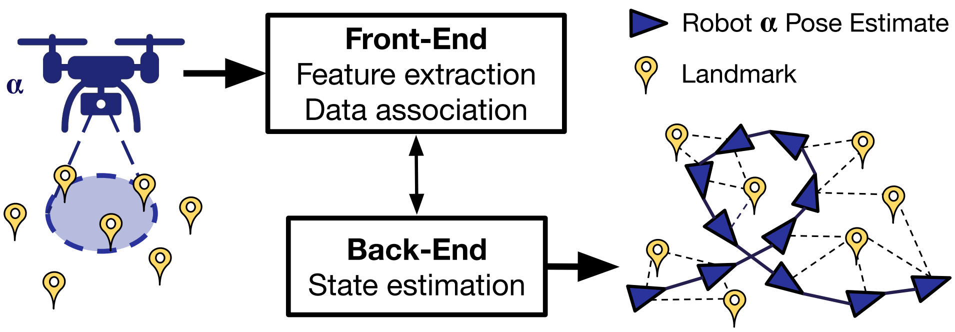


Figure 2.1 Single-robot SLAM Overview

2.4.1 Multi-robot systems

In multi-robot systems, data collection and state estimation are no longer entirely located on a single entity, so there is an inevitable need for communication between the agents (i.e., robots, base stations, etc.) which is the crux of the problem.

Moreover, multi-robot systems have additional properties to consider when designing C-SLAM systems, and taxonomies can be defined to classify approaches and highlight their benefits and tradeoffs. The taxonomy proposed in [35] presents considerations that are well suited to the C-SLAM problem. It distinguishes approaches according to the following aspects:

Team size The number of robots in the system. Larger teams usually perform tasks more efficiently but may be harder to coordinate.

Communication range Direct communication between robots is limited by their spatial distribution and the communication medium. In some cases, robots might be unable to communicate for long periods of time, while in others they might always be in range of another robot.

Communication topology The communication network topology affects how robots communicate with one another. For example, they might be limited to either broadcast or one-to-one messages.

Communication bandwidth The bandwidth of the communication channel affects what information robots can afford to share.

System reconfigurability The robots will move and are likely to change spatial configuration over time. This can affect the communication topology and bandwidth.

Team unit processing ability Individual robot's computational capability can affect the computation cost of C-SLAM approaches and the distribution of computation tasks.

Team composition Robots can be homogeneous or heterogeneous over several aspects such as locomotion methods and available sensors.

The main differences between most C-SLAM techniques in the literature lie in the properties of the multi-robot system considered, especially their resource management strategy. One subclass of multi-robot systems particularly relevant to the future of C-SLAM are swarm robotics systems [36], which are inspired by social animals. Two main characteristics are required for swarm-compatibility in C-SLAM: robots' sensing and communication capabilities must be local, and robots can not have access to centralized control and/or to global knowledge. Such systems would present considerable benefits: they would have robustness to the loss of individual units, and they could scale well to large numbers of robots.

2.4.2 C-SLAM Problem definition

When all robots' initial states are known or can be estimated, the C-SLAM problem is a simple extension of the single-robot SLAM *MAP* problem that includes all the robots' states, measurements, and additional inter-robot measurements linking different robots' maps. In a setup with two robots (α, β), where X_α and X_β are the state variables from robot α and β to be estimated, Z_α and Z_β are the set of measurements gathered by robot α and β independently, $Z_{\alpha\beta}$ is the set of inter-robot measurements linking both robot maps containing relative pose estimates between one pose of robot α and one of robot β in their respective trajectories, and X_α^*, X_β^* are the solutions, the problem can be formulated as:

$$\begin{aligned}(X_\alpha^*, X_\beta^*) &\doteq \operatorname*{argmax}_{X_\alpha, X_\beta} p(X_\alpha, X_\beta | Z_\alpha, Z_\beta, Z_{\alpha\beta}) \\ &= \operatorname*{argmax}_{X_\alpha, X_\beta} p(Z_\alpha, Z_\beta, Z_{\alpha\beta} | X_\alpha, X_\beta) p(X_\alpha, X_\beta)\end{aligned} \tag{2.3}$$

However, when the relative starting locations and orientations of the robots cannot be determined, the initial guess of the robots states $p(X_\alpha, X_\beta)$ is not available. In that case, there are infinite possible initial alignments between the multiple robot trajectories. Therefore, in absence of a prior distribution, C-SLAM is reduced to the following *Maximum Likelihood Estimation* (MLE) problem.

$$(X_\alpha^*, X_\beta^*) \doteq \operatorname*{argmax}_{X_\alpha, X_\beta} p(Z_\alpha, Z_\beta, Z_{\alpha\beta} | X_\alpha, X_\beta) \tag{2.4}$$

The C-SLAM problem formulation is still evolving to this day and progress still needs to be made to achieve an efficient decentralized, distributed and robust implementation. To give some perspective, Figure 2.2 presents some major milestones in the evolution of the C-SLAM problem over time. More details on these milestone works are provided in the following sections.

2.4.3 Centralized, Decentralized and Distributed Systems

An important distinction in C-SLAM, and in multi-robot systems in general, is the difference between the *global* and *local* perspectives. The local perspective is the default point of view in single-robot SLAM. Accordingly, the pose and map estimates are expressed in an internal

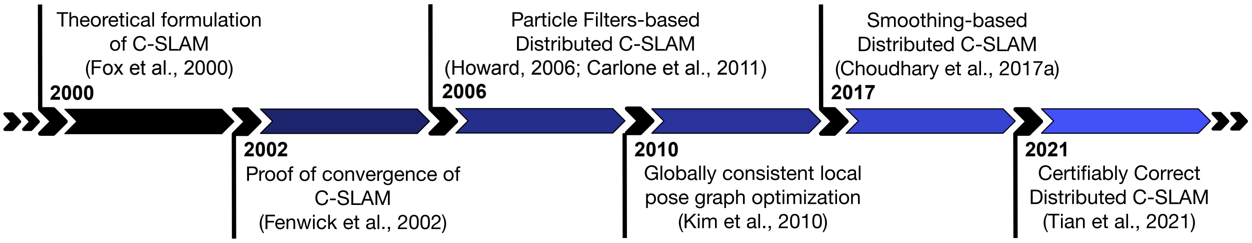
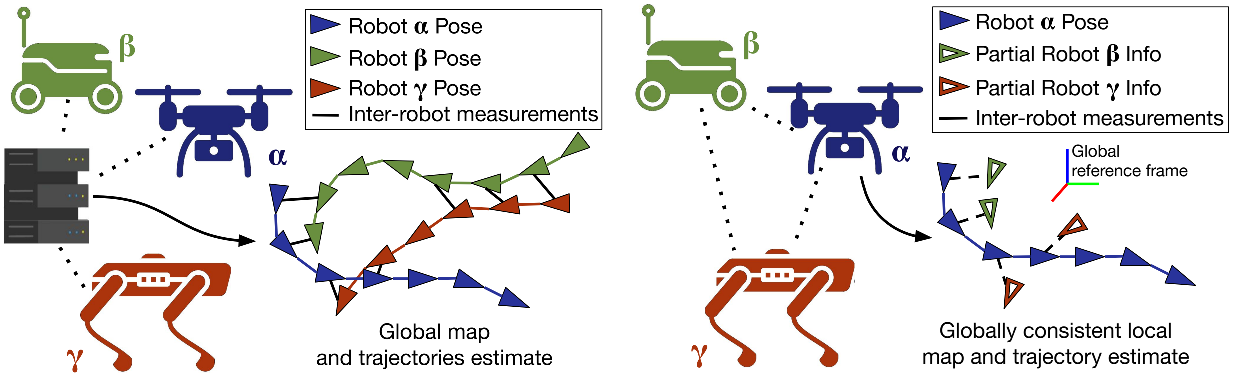

Theoretical formulation of C-SLAM (Fox et al., 2000)
2000
2002
Proof of convergence of C-SLAM (Fenwick et al., 2002)
Particle Filters-based Distributed C-SLAM (Howard, 2006; Carlone et al., 2011)
2006
2010
Globally consistent local pose graph optimization (Kim et al., 2010)
Smoothing-based Distributed C-SLAM (Choudhary et al., 2017a)
2017
2021
Certifiably Correct Distributed C-SLAM (Tian et al., 2021)

Figure 2.2 C-SLAM Problem Major Milestones

reference frame which is usually the starting location of the robot's mission. However, in C-SLAM, one has to consider the global perspective of the system since the pose and map of each robot need to be expressed in a shared global reference frame. This means that every landmark can be expressed within the same coordinates system by every robot in the team. Otherwise, shared information (e.g. position of observed landmarks) would have no significance to the receiving robot due to the representation being in another unknown local reference frame. Establishing this global reference frame using C-SLAM allows the robots to collectively perceive the environment and benefit from each other's observations.

To achieve this global understanding, one could either solve C-SLAM in a centralized or decentralized manner. In a centralized solution, the estimator has a global view of the entire team of robots: it performs the estimation given perfect knowledge of the measurements of each robot. These measurements can be raw or preprocessed, and shared on demand depending on the communication limits.

Unfortunately, due to communication constraints, solving centralized C-SLAM quickly becomes intractable as the number of robots increases [24]. Thus, a better solution for scalability is to solve C-SLAM in a decentralized manner [37]. This means that each robot only has access to a local view comprised of its own data and partial information from its neighbors. Therefore, decentralized systems cannot solve the C-SLAM problem for all the robots at once, but aim instead for local solutions on each robot that are consistent with their neighbors. Then, iteratively and over time, with the robots gradually improving their estimates given their neighbors' latest data, decentralized techniques converge to local solutions that are mutually consistent across the team of robots. So, upon convergence, the individual robots reach a common understanding and their local maps are aligned with the common (global) reference frame. Figure 2.3 provides examples of the C-SLAM problem and output in both perspectives.



(a) Centralized C-SLAM

(b) Decentralized C-SLAM

Figure 2.3 Illustration of centralized and decentralized approaches to solve the C-SLAM estimation problem. The decentralized illustration is from the local perspective of robot α.

Aside from the centralized/decentralized classification, a system is distributed if the computation load is divided among the robots. The two notions are independent. Therefore, a system could be centralized and distributed at the same time, if, for example, each robot performs parts of the computation, but a central node is required to merge the individual results from all the robots [38].

Centralized C-SLAM Many seminal C-SLAM works are centralized and solve the estimation problem in eq. 2.4 from the global perspective. In those approaches, the robots are essentially reduced to mobile sensors whose data is collected and processed on a single computer. Examples of centralized C-SLAM techniques include [39,40] that gather all the robots' measurements at a central station to compute the global map. [41] improves this solution by marginalizing unnecessary nodes in the local pose graphs so only a few condensed measurements need to be shared to the central computer. Other centralized approaches [42–44] perform C-SLAM with monocular cameras, successfully solving the associated 3D estimation challenges, while [45] focuses on micro-aerial vehicles constraints. [46] proposes a framework to reuse existing single robot SLAM solutions for C-SLAM. The same idea is explored in [47], in which a popular single-robot SLAM technique [48] is converted into C-SLAM. [49,50] integrate inertial measurements from IMUs in their centralized C-SLAM systems. [51] proposes that the central node spreads the resulting map across the robots to limit the memory usage.

Improving upon the pure centralized methods, some techniques do not rely on a single computer, but can use different robots or base stations for the computation. This way, the system can adapt itself to the failure of one node or communication link and complete the mission.

A typical solution is to use replicated central servers among the robots [52].

Decentralized C-SLAM Solving the C-SLAM problem in a decentralized manner is radically different, but offers major benefits in terms of computation, communication and privacy [37, 53]. Such systems are usually distributed and solve the estimation problem from eq. 2.4 partially on each robot. As shown in Figure 2.3b, each robot computes its own local map and uses partial information from other robots as well as inter-robot measurements to achieve a local solution. Over several iterations with its neighbors, each robot's resulting local solution converges to a solution consistent with the global reference frame. These techniques mitigate communication and computation bottlenecks since the loads are spread across the robot team [54]. Alternatively, the full mapping data can be sent to every robot for redundancy and a subset of robots can be designated for computation [55–57].

As one would expect, decentralized and distributed techniques come with many additional challenges that need to be tackled such as complex bookkeeping, information double counting or synchronization issues.

2.4.4 Complete C-SLAM Systems

In C-SLAM, as well as in single-robot SLAM, the front-end handles perception-related tasks and the back-end generates state estimates using all measurements gathered. However, in C-SLAM, the front-end and back-end computations do not necessarily occur fully on a single robot anymore depending on the sensing, communication, and estimation strategies. For example, in a centralized system, all robots could send their sensor data directly to a single unit which would then perform the front-end and back-end steps for the whole team. While in a decentralized and distributed setup, a robot could perform feature extraction on its own and communicate with other robots for data association and state estimation. Every part of a C-SLAM system can be subject to distribution or decentralization.

In addition, the front-end needs to find links and relative measurements between the individual maps. Therefore, the front-end must also perform data association to detect and compute inter-robot loop closures, which will be detailed in Section 6.5. It follows that the back-end must generate estimates combining the individual and shared measurements as described in Section 2.6.

In the recent years, several complete C-SLAM systems have been developed and deployed in realistic scenarios. For example, some solutions deployed in large-scale environments during the DARPA Subterranean Challenge [58,59] led to the developments of new C-SLAM systems, such as the robust lidar-based approach of [60]. Alternatively, [61] proposes a vision-based centralized C-SLAM system incorporating inertial measurements, which has been tested with up to 12 robots in simulation. In another line of work, [62] presents a distributed and decentralized system robust to spurious measurements, along with online experiments on real robots, and a publicly available implementation. A subsequent approach, detailed in [63], puts together a completed decentralized and distributed C-SLAM system including a novel robust distributed pose graph optimization back-end, and a front-end producing globally consistent metric-semantic 3D mesh models of the explored environment. Those works are some of the best starting points for researchers and engineers looking to implement, improve and deploy practical C-SLAM systems in challenging conditions.

2.5 C-SLAM Front-End

Although the division between the front-end and the back-end is sometimes blurry due to the presence of feedback loops between the two processes, a typical C-SLAM front-end is in charge of producing landmark estimates, odometry measurements, and both intra-robot and inter-robot loop closures.

Odometry measurements aim to capture the translation and rotation of a robot from one time step to the next. Common techniques measure wheel movements, integrate from an IMU, and/or perform geometric matching between consecutive images or laser-scans. Intra-robot loop closures are the measurements used by a SLAM system to relocalize itself and reduce its estimate error caused by odometry drift. Using place recognition, the system can detect previously visited locations and compute relative measurements between them. In other words, intra-robot loop closures are estimates relating non-consecutive poses in the robot trajectory that observed the same places. Since the computing of odometry and intra-robot loop closure measurements can be fully done locally on each robot, the approaches used are the same in both SLAM and C-SLAM. Thus, we refer the reader to [1, 64, 65] for surveys of the current techniques.

Conversely, inter-robot loop closures relate poses of different robots trajectories. They are the seams that stitch together the estimates from multiple robots: they draw connections between the individual robots' local maps to build the global understanding of the environment. Generating inter-robot loop closures is the main focus of contributions to the front-end of C-SLAM systems, and key to ensure consistency of the estimates.

2.5.1 Direct vs Indirect Inter-Robot Loop Closures Measurements

Inter-robot loop closures can be classified as direct or indirect [40]. Direct inter-robot loop closures occur when two robots meet, and they are able to estimate their current relative location with respect to each other through direct sensing. Indirect inter-robot loop closures are produced by looking back into maps to find partial overlaps for places that have been visited by both robots. Given these measurements, the robots can estimate the relative transformation between their maps. In general, indirect inter-robot loop closures detection produce more measurements and usually achieve a higher accuracy, but require more communication and processing. Indeed, the detection process is often the communication bottleneck of C-SLAM given the large amount of data required to compare landmarks between the individual local maps [66].

Direct Inter-Robot Loop Closures

The idea of direct inter-robot loop closures is to compute the relative pose between two or more robots when they physically meet in the same location. This is usually done through direct sensing of one another. For example, [40] operated a quadcopter and a ground robot and the latter was equipped with a checkerboard pattern that could be detected by the quadcopter's camera. [67] used a combination of direct and indirect detection approaches, where colored cylinders were installed to be detected by omnidirectional cameras. In addition, [68–70] propose to replace visual loop closures by Ultra-Wide Band (UWB) measurements from beacons onboard the robots. Given a few distance measurements provided by the UWB sensors, the robots can estimate their current relative pose with respect to each other and establish a common reference frame.

Indirect Inter-Robot Loop Closures

Indirect inter-robot loop closure detection is the extension of single-robot loop closure detection to multiple maps. In fact, approaches to find indirect inter-robot loop closures often rely on the same core algorithms as intra-robot loop closures. The first challenge is the loop closures detection, which consists of detecting overlaps between the individual maps. This task is usually handled by a place recognition module which can efficiently compare new observations against previous sections of the robots' maps. Following place recognition matches, geometric estimation is performed to compute the relative pose between the two places.

In the case of visual sensors, the place recognition problem has been studied extensively [65].

The seminal work of visual bags of binary words [71] is still popular, but newer approaches based on deep learning, such as NetVLAD [72], are more accurate and data-efficient. Loop closure relative pose measurements can be computed using visual features matching and multi-view geometry [73].

Finding inter-robot overlaps is a harder task with 3D point clouds given the dense data that need to be shared and the lack of expressive features to perform place recognition. To that end, compact and robust global point cloud descriptors [74] can be relied upon to compare point clouds for place recognition. Other approaches extract features from the point cloud that can serve for place recognition while providing initial guesses for later geometric alignments [18], or even directly compute loop closure measurements [75]. While the classical *Iterative Closest Point* method [76] is still commonly used in single robot SLAM to compute relative pose measurements between two matching point clouds, it is not well suited for multi-robot operation due to its reliance on a good initial guess, which is usually not available between the robots' local maps. To handle the initialization problem, early work from [77] presents a correlation-based algorithm that can be efficiently solved on a GPU for real-time scan matching. Another common solution is to use submaps matching for both stereo cameras [78–80] and lidars [18, 81]. During this process, multiple laser scans or 3D point clouds are clustered into submaps which can in turn be registered more efficiently.

2.5.2 Heterogeneous Sensing

In many applications, the teams of robots are composed of different platforms equipped with different onboard sensors. Heterogeneous sensing comes with the additional challenge of matching map data in different representation to perform relocalization and/or map fusion. To this end, a recent study evaluated the repeatability of existing keypoint detectors between data from stereo cameras and lidars For example, when matching data from both stereo cameras and lidars, one needs to chose repeatable 3D feature representations that are consistent despite the differences in density and field-of-view [82]. Another approach is to use an intermediate map representation that can be produced by different kinds of sensors [83]. For example, [84] proposes to compare elevation maps that are invariant to sensor choice: lidars or cameras.

2.5.3 Non-Conventional Sensing

While most C-SLAM techniques use the typical SLAM sensors (i.e., lidars and monocular, RGB-D, or stereo cameras), many recent research works have explored the use of non-conventional sensors: [85] uses omnidirectional (i.e., fish-eye) cameras, [86] performs C-SLAM with event-based vision sensors, and [87] integrates ambient radio signals (i.e., signals of opportunity) into their system. In a similar vein, [88] leverages existing WiFi access points in most indoor environments to perform loop closures based on their radio signal fingerprint. Alternatively, some approaches use only a few higher-level landmarks, such as objects, for tracking and place recognition [89, 90]. This type of approach have regained popularity with the increasing performance of deep learning-based methods in semantic segmentation as discussed in Section 2.9.5.

2.6 C-SLAM Back-End

As mentioned before, the role of the C-SLAM back-end is to estimate the state of the robot and the map given the front-end measurements. The difference with single-robot SLAM is the presence of inter-robot measurements, the need to reach consensus, and the potential lack of an initial guess since the global reference frame and the starting positions of the robots are usually initially unknown. Nevertheless, similar to single-robot solvers, C-SLAM back-ends are roughly divided in two main categories of inference techniques: filtering-based and smoothing-based. Although filtering-based approaches were the most common among the early techniques (e.g. EKF [91] and particle filters [92]), smoothing-based approaches quickly gained in popularity and are currently considered as superior in most applications [93]. This section provides an overview of the different categories of estimation workhorses for C-SLAM and presents examples from the literature.

2.6.1 Filtering-Based Estimation

Filtering approaches are often characterized as online in the sense that only the current robot pose is estimated and all previous poses are marginalized out [31] at each time step. Consequently, the estimation of the posterior in eq. 2.1 at time t only depends on the posterior at time $t-1$ and the new measurements.

The classical filtering technique for nonlinear problems (i.e., all problems in robotics except trivial ones) is the Extended Kalman Filter (EKF). It has been applied to C-SLAM in various ways among which the information filter method presented in [94]. In a nutshell, EKF are Gaussian filters that circumvent the linear assumptions of Kalman filters through linearization (i.e., local linear approximation); however, the linearization process potentially leads to inconsistencies when the noise is too large. A major advantage of EKF techniques [94–97] over smoothing techniques is that the covariance matrix is available without requiring additional computation, which can be useful for feature tracking or active exploration. For example, one could prioritize the exploration in the most uncertain directions. Yet, an explicit covariance matrix is rarely required, so alternative filtering techniques seek to avoid its computation, such as the smooth variable structure filters approach presented in [98].

Building on the EKF, Rao-Blackwellized Particle Filters (RBPF) [99] are another popular filtering approach for the C-SLAM problem. Techniques, such as [100], use samples (particles) to represent the posterior distribution in eq. 2.1 and perform variable marginalization using an EKF which drastically reduce the size of the sampling space. [101] extends on [100] and improves its consistency while making it fully distributed. [102] adapts RBPF to visual C-SLAM and [103] showcases the potential of RBPF C-SLAM for industrial applications.

It is important to note that, according to theoretical analysis results [104], reducing the number of relative position measurements between the robots to a minimum, to limit communication and computation, only inflicts a small penalty on the localization performance. It was also shown that the presence of even only one robot equipped with an absolute positioning sensor is enough to bound the positioning uncertainty of the whole team. Additionally, analytical upper bounds can be computed to predict the positioning uncertainty as function of the size of the explored area, the number of landmarks, the number of robots, and the accuracy of the onboard sensors [105]. Those theoretical results can be of great use in the design of a C-SLAM system.

2.6.2 Smoothing-Based Estimation

Besides the linearization error, another drawback of filtering techniques is that the marginalization of past pose variables leads to many new links among the remaining variables. Indeed, the elimination of each pose variable leads to interdependence between every landmark variables to which it was connected. As a result, the variables become increasingly coupled and this leads to more computation. This problem also affects smoothing approaches, but a clever ordering during variable elimination can significantly reduce its impact on performance [106]. Moreover, in smoothing, there is less marginalization required which means that the variables will stay sparsely connected. This sparsity is exploited by modern solvers to yield significant speed-ups [93]. In addition, unlike filtering-based approaches, smoothing techniques improve their accuracy by revisiting past measurements instead of only working from the latest estimate. Hence, filtering techniques fell out of favor due to the better performance of smoothing both in terms of accuracy and efficiency. Moreover, in the context of C-SLAM, the sparsity reduces the amount of data to be exchanged during the estimation process [107].

In order to formalize the estimation problem solved by C-SLAM back-ends, we present a general smoothing formulation for pose-graph C-SLAM with two robots (α, β) in which the map landmarks are marginalized into odometry and loop closure measurements. The robots poses and measurements are elements of the special Euclidean manifold SE(d) where d is the dimension of the problem (i.e., 2 or 3) [108].

First, assuming that the measurements noises are uncorrelated, we can factorize eq. 2.4 as follows:

$$\begin{aligned}(X_\alpha^*, X_\beta^*) &\doteq \underset{X_\alpha, X_\beta}{\operatorname{argmax}}\, p(Z_\alpha, Z_\beta, Z_{\alpha\beta} | X_\alpha, X_\beta) \\ &\doteq \underset{X_\alpha, X_\beta}{\operatorname{argmax}} \Bigg(\prod_{i=1}^{l} p(z_\alpha^i | X_\alpha^i) \prod_{j=1}^{m} p(z_\beta^j | X_\beta^j) \\ &\qquad\qquad \prod_{k=1}^{n} p(z_{\alpha\beta}^k | X_\alpha^k, X_\beta^k) \Bigg)\end{aligned} \tag{2.5}$$

where $p(z_\alpha^i | X_\alpha^i)$ is the likelihood of the i^{th} measurement of robot α (i.e., z_α^i) given the subset of variables X_α^i on which it is dependent, $p(z_\beta^j | X_\beta^j)$ is the likelihood of the j^{th} measurement of robot β (i.e., z_β^j) given the subset of variables X_β^j on which it is dependent, and $p(z_{\alpha\beta}^k | X_\alpha^k, X_\beta^k)$ is the likelihood of the k^{th} inter-robot measurement (i.e., $z_{\alpha\beta}^k$) given the subset of variables X_α^k and X_β^k. There are l measurements related only to state variables from robot α, m measurements related only to state variables from robot β, and n measurements related to state variables from both robots.

Second, assuming that the measurements are disturbed by zero-mean Gaussian noise with information matrix Ω (i.e., inverse of the covariance), we can express the individual measurement likelihood as

$$p(z_\alpha^i | X_\alpha^i) \propto \exp\left(-\frac{1}{2} \left\| h_\alpha^i(X_\alpha^i) - z_\alpha^i \right\|_{\Omega_\alpha^i}^2 \right) \tag{2.6}$$

where h_α^i is a function that maps the state variables to the measurements.

Third, since maximizing the likelihood is equivalent to minimizing the negative log-likelihood, we obtain the following nonlinear least squares formulation of problem 2.4:

$$\begin{aligned}(X_\alpha^*, X_\beta^*) &\doteq \underset{X_\alpha, X_\beta}{\operatorname{argmin}} -\log\Bigg(\prod_{i=1}^{l} p(z_\alpha^i | X_\alpha^i) \prod_{j=1}^{m} p(z_\beta^j | X_\beta^j) \\ &\qquad \prod_{k=1}^{n} p(z_{\alpha\beta}^k | X_\alpha^k, X_\beta^k) \Bigg) \\ &\doteq \underset{X_\alpha, X_\beta}{\operatorname{argmin}} \Bigg(\sum_{i=1}^{l} \left\| h_\alpha^i(X_\alpha^i) - z_\alpha^i \right\|_{\Omega_\alpha^i}^2 + \sum_{j=1}^{m} \left\| h_\beta^j(X_\beta^j) - z_\beta^j \right\|_{\Omega_\beta^j}^2 \\ &\qquad + \sum_{k=1}^{n} \left\| h_{\alpha\beta}^k(X_\alpha^k, X_\beta^k) - z_{\alpha\beta}^k \right\|_{\Omega_{\alpha\beta}^k}^2 \Bigg)\end{aligned} \tag{2.7}$$

This nonlinear least squares problem can be solved either on a single computer or in a distributed fashion. In the centralized case, one can simply use single-robot pose graph optimization solvers [4, 109–111]. Incremental single-robot solvers [112] can also be adapted for the centralized C-SLAM problem to continuously update the global pose graph with the latest measurements from the robots [113]. Recently, a client-server architecture has been proposed in which resource-limited clients (e.g. robots or mobile phones) only optimize small parts of the map while the server processes the rest [114]. This centralized and distributed approach allows for accurate real-time state estimation even with limited computation and memory capacity onboard the clients.

Among the distributed solvers, many early techniques used Gaussian elimination [89, 115, 116]. Although popular, those approaches require the exchange of dense marginals which means that the communication cost is quadratic on the number of inter-robot measurements. Furthermore, those approaches rely on linearization, so they require complex bookkeeping to ensure the consistency at the linearization point within the team of robots. To reduce the complexity, [117] introduces a distributed marginalization scheme to limit the size of the optimization problem.

More recently, the approach in [53] leverages the Distributed Gauss-Seidel algorithm introduced in [118] to solve eq. 2.7. This technique avoids complex bookkeeping and information double-counting in addition of satisfying privacy constraints by exchanging minimal information on the robot individual trajectories. In another line of work, [119] extends a single-robot incremental solver [112] towards distributed multi-robot setups. This is useful in online missions as it can update the current estimate based on the latest observations without recomputing the whole problem.

Optimization on Riemannian manifolds [120] has also been considered extensively to solve the C-SLAM problem [121, 122]. Approaches in [123–125] introduce a multi-stage distributed Riemannian consensus protocol with convergence guarantees to globally optimal solutions in noiseless scenarios. Expanding on those ideas, a recent technique [126], based upon a sparse semidefinite relaxation, provides exactness guarantees even in the presence of moderate measurement noise. Moreover, this latter technique has been extended to consider asynchronous scenarios and parallel computation [127], which are often critical to deal with communication delays inherent to multi-robot systems.

2.6.3 Other Estimation Techniques

Other estimation techniques have been proposed for C-SLAM. Among them, the distributed Jacobi approach has been shown to work for 2D poses [128]. [129, 130] look into consensus-based algorithms and prove their convergence across teams of robots. Also, apart from the solver itself, researchers have studied which measurement and noise models are the best suited for C-SLAM [131].

We observe that more exciting new directions are still being discovered, considering that recent approaches such as [126] have been shown to outperform, both in accuracy and convergence rate, the well established Distributed Gauss-Seidel pose graph optimization method [53] reused in many state-of-the-art C-SLAM systems such as [37, 62, 132]. Those promising approaches also include the majorization-minimization technique from [133], the consensus-based 3D pose estimation technique inspired by distributed formation control from [134,135], and [136] distributed estimator based on covariance intersection.

2.6.4 Perceptual Aliasing Mitigation

As it is the case in single robot SLAM, loop closure detection is vulnerable to spurious measurements, i.e., outliers, due to perceptual aliasing. This phenomenon occurs when two different places are conflated as the same during the place recognition process. This motivates the need for robust techniques that can detect and remove those outliers to avoid dramatic distortions in the C-SLAM estimates. A common approach is to adopt a robust objective function less sensitive to outliers [5, 137–140]. Outliers mitigation might also help against adversarial attacks by rejecting spurious measurements injected by a nefarious agent.

The classic approach to remove outliers is to use the RANSAC algorithm [141] to find a set of mutually consistent measurements [113]. While RANSAC works well in centralized settings, it is not adapted to distributed systems. Therefore, researchers recently explored other ways of detecting outliers such as leveraging extra information from the wireless communication channels during a rendezvous between two robots [132]. Since such approaches work only for direct inter-robot loop closures, there is a need for general robust data association in the back-end. To that end, [142] uses expectation maximization to infer which inter-robot measurements are inliers and which ones are outliers. One currently popular approach in C-SLAM is the use of pairwise consistency maximization to search for the maximal clique of pairwise consistent measurements among the inter-robot loop closures [143]. [62] introduces a distributed implementation of this technique which does not require any additional communication when paired with distributed pose graph optimization, while [144] proposes an incremental version, and [145] extends the pairwise consistency evaluation with a data similarity metric. Another recent work [63] extends to distributed setups the Graduated Non-Convexity approach for outlier rejection previously applied to single-robot SLAM [140]. It is important to note that those latest approaches only apply to smoothing-based C-SLAM since, unlike filtering, it allows the removal of past measurements from the estimation.

2.7 System-Level Challenges

Along with the front-end and back-end specific challenges, some issues and design choices affect the whole C-SLAM system. As in single-robot SLAM, the map representation has strong repercussions on motion tracking, place recognition and state estimation. On top of this, multi-robot systems (described in Section 2.4.1) present unique challenges to C-SLAM in terms of communication and coordination.

2.7.1 Map Representation

When designing large multi-robot systems, the choices of map representation could affect computation load, memory usage, and communication bandwidth. First, it is important to note that an interpretable map is not always required. For example, when the sole objective is collaborative localization, a feature map can be sufficient. In those cases, each robot locally tracks landmarks, or features, and searches for correspondences in other robots' feature maps to obtain indirect inter-robot loop closure measurements. The local feature maps can be merged frequently so that the robots can navigate and track features in a global map, or they can be shared on demand upon place recognition events. This way, the robots can operate in the same reference frame without the computation and communication burden of building an interpretable map model.

When required, the chosen map representation depends on the mission objective and environment. For example, in the case of ground robots in flat indoor environments, a 2D map might be sufficient [146]. In those scenarios, occupancy grid maps have been shown to be a compact and more accurate solution [55, 147] than feature-based maps [148]. However, 3D representations are sometimes required (e.g. for rough terrain navigation) at the cost of more computation, storage, and communication, which can be difficult to handle when resources are limited on the robots. Given the communication constraints in C-SLAM systems, compact or sparse representations, such as topological maps [149,150], are often preferred. In the same vein, some works aim for semantic-based representations in the form of sparse maps of labelled regions [90]. Map representations can also affect long-term operations due to the increasing size of the map in memory [151], which is also a challenge in single-robot SLAM.

2.7.2 Efficient and Robust Communication

One of the core implementation differences between SLAM and C-SLAM is the need for communication and coordination within the robotic team. For efficiency, the required bandwidth needs to be minimal, and the communication network needs to be robust to robot failures and varying topologies.

The communication bottleneck of a C-SLAM system is generally caused by the exchanges of sensor data or representations used to compute inter-robot loop closures [66]. Robots need to share enough data to detect whether other robots have visited the same area, and then estimate a map alignment using any overlapping sections of their maps. Hence, contributions to the front-end of C-SLAM systems often consist of mechanisms to perform an efficient search for loop closure candidates over a team, considering communication constraints. Conversely, the back-end generally relies on a pose-graph which can be shared without the need for large bandwidth.

Efficient Data Sharing

While some early techniques simply share all the data from one robot to another, new sensors produce increasingly rich and dense data. The days of raw sensor data transmission are over and most current techniques in literature opt for some sort of compression or reduction. Even among the early techniques [152], the idea of a communication budget has been explored. More recently, the topic has gathered more attention with new techniques carefully coordinating the exchange of data when two robots meet, accounting for the available communication and computation resources [153–156]. One idea is to compress the generated maps using self-organizing maps obtained through unsupervised learning [157,158]. The use of compact representations has also been explored with high-level semantic features: [90] relies on objects as landmarks, needing to communicate only object labels and poses to other robots, and [159] presents a compact object-based descriptor relying on the configuration of objects in a scene to perform place recognition. In addition to making representations compact, it is useful to ensure that only helpful information is shared. Hence, [160] introduces a novelty metric so that only sufficiently novel measurements compared to the existing map are transmitted.

The problem has been extensively studied specifically for visual C-SLAM: [66] proposes to share visual vocabulary indexes instead of feature descriptors to reduce the required bandwidth. Other approaches focus on scalable team-wide place recognition by assigning each robot with a predetermined range of words from a pretrained visual bag of words [161], or regions of full-image descriptors [162]. [163, 164] remove landmarks that are not necessary for localization, [165] introduces a new coding to efficiently compress features, and [166] proposes data sharing algorithms specialized for visual inertial C-SLAM.

In some extreme cases, communication is severely limited due to the properties of the transmission medium or the large distance between the robots: [12, 107] explore the special case of underwater operations with low bandwidth acoustic communication, and [79] considers long distance radio modules with very limited bandwidth to build the collaborative map through small incremental updates.

Network Topology

Another important aspect to consider is the network topology. Current techniques either assume full connectivity, multi-hop connectivity or are rendezvous-based. Full connectivity means that each robot can directly communicate with all other robots at any time such as in [161, 162]. Multi-hop connectivity implies that robots can only share information with their neighbors and it might require multiple neighbor-to-neighbor transmissions to reach all robots [167, 168]. Rendezvous-based communication means that the robots share data only when they meet and, therefore, do not require any connectivity maintenance. Rendezvous-based C-SLAM also offers the opportunity to perform direct inter-robot loop closure detection during the encounters [67].

The impact of the network topology is especially important during the inference step because disconnections or multi-hop paths can lead to inconsistencies or synchronization issues. Thus, [169, 170] examine the conditions that allow distributed inference to reach the same result as a centralized equivalent approach. Another approach [171] leverages the progress in the field of distributed computing to improve the robustness to connectivity losses, while [172] evaluates the use of *Wireless Sensor Network*-based communication which is less reliable and predictable, but offers a flexible architecture with self-organization capabilities.

2.8 Benchmarking C-SLAM

Despite the tremendous progress in the field during the last decade, C-SLAM techniques face tough challenges in terms of reproducibility and benchmarking. C-SLAM systems involve multiple software modules and lots of different hardware components, making it hard to replicate perfectly. While standardized benchmarking approaches have been emerging for single-robot SLAM [173], such systematic evaluation techniques are still lacking for C-SLAM.

Moreover, only a few datasets dedicated to C-SLAM exist. [174] consists of 9 monocular camera subdatasets and [175] is dedicated to stereo-inertial C-SLAM. Therefore, the common approach to evaluate C-SLAM solutions is to split single robot SLAM datasets into multiple parts and to associate each one to a robot. When splitting the dataset, careful attention has to be given to ensure the presence of overlaps between the parts for loop closing. In addition, one should avoid overlaps near the cutting points, where the viewpoint and lighting conditions are exactly the same since they depict the same place viewed by the robot at the same point in time: this kind of overlaps is highly unrealistic in multi-robot operations. One of the most used dataset in the literature is the KITTI self-driving car dataset comprised of lidar and stereo camera data [176]. New datasets of interest include KITTI360 [177] which adds fish-eye cameras, the very large Pit30M lidar and monocular camera dataset that contains over 30 million frames [178], and the DARPA SubT datasets which come with standardized evaluation tools for SLAM [179, 180].

2.9 Open Problems and Ongoing Trends

This section presents open problems and trending ideas in the research community to improve C-SLAM. These new trends push the boundaries of what C-SLAM can do and offer an exciting view of the field's future.

2.9.1 Resilient Inter-Robot Communication

Although the limitations of inter-robot communication have been a major concern since the inception of C-SLAM, it is still one of the main open problems in the field. In particular, there is a need for resilient communication strategies, aiming beyond robustness to endure unexpected disruptions and ensure swift recovery [181]. Delays and dropouts are inevitable in realistic systems, and their effects are amplified when multiple robots operating simultaneously are flooding the network. Delays and out-of-sequence messages can have dramatic effects on real-time robot control which heavily relies on accurate and up-to-date state estimates from C-SLAM [182], and yet they still have not been thoroughly addressed by the research community. Instead, current approaches focus primarily on minimizing communication, which can be achieved, for example, by posing distributed loop closure detection as an optimization problem subject to a budget constraint on total data transmission [154].

Another open problem inherent to C-SLAM and inter-robot communication is the risk of adversarial attacks. In a future in which robots, such as autonomous cars, collaborate on a large scale, security and data integrity will be one of the major concerns of consumers. In addition to the usual risks of infection and hijacking, byzantine data manipulation could lead to map merging poisoning and intentionally erroneous C-SLAM estimates [183]. Thus, further investigation and more efforts have to be deployed on system security.

An interesting, yet still underdeveloped, trend is to leverage the communication medium for inter-robot measurements. This has been successfully done with UWB [68–70] or WiFi [88], and could be a promising avenue using multipath analysis with channel estimators in 5G networks [184]. Future techniques might even sidestep inter-robot data transfer completely by communicating via sensor observations of each other and predetermined cues such as visual tags or behavioral patterns [40].

2.9.2 Managing Limited Computing Resources

Aside from communication, computational constraints are an essential consideration in robotics since robots are usually equipped with limited onboard processing devices. It is particularly important in C-SLAM where multiple sub-processes from sensory analysis to inter-robot communication need to be run simultaneously. Thus, to support the current expansion of C-SLAM capabilities, there is a constant need for efficiency gains. In fact, computation improvements are often at the forefront of new trends in C-SLAM such as the rise of semantic methods, discussed in Section 2.9.5, which were enabled by GPU-based deep learning [185]. Moreover, as discussed in Section 2.9.6, many new applications of C-SLAM are designed for even smaller platforms such as mobile phones.

Centralized techniques are a natural solution to limited onboard computation capabilities, and, in that regard, recent research suggest that C-SLAM could efficiently leverage the progress in cloud computing. The connection between the two fields is somewhat intuitive: why perform all the processing on robots with limited resources when we could use powerful remote clusters of servers instead? For example, [186] offloads the expensive map optimization and storage to a server in the cloud. [187] proposes a cloud robotics framework for C-SLAM based on available commercial platforms. Using a similar approach, [188] manage to perform C-SLAM with up to 256 robots. This is orders of magnitude more than the current techniques based on onboard computation can achieve.

However, while cloud techniques solve the problem of limited computing power onboard the robots, they still face the issue of limited communication bandwidth which is exacerbated when many robots transmit their data through a single communication link. Hence, instead of using remote servers, other strategies need to be explored. For example, a subset of a team of robots could act as a computing cluster to free other robots from the heavy computation burden [189]. Such moving clusters performing computing closer to the sources of data are in accordance with the edge computing paradigm [190] to save bandwidth and reduce response time [191].

2.9.3 Adapting to Dynamic Environments

Another inherent problem in multi-robot system is the presence of moving objects in the environment (e.g. people or vehicles). In this regard, the other moving robots in the team are especially problematic. This is a substantial issue since SLAM techniques rely on the tracking of static landmarks. Attempting to solve this problem, [192] proposes the simple idea of pointing the cameras towards the ceiling when operating indoors with ground robots so that they cannot see each other. [193] proposes instead to classify dynamic points using the reprojection error and to keep only the static points for estimation. In a different vein, [194–196] and more recently [197] extend upon the Rao-Blackwellized particle filters framework to track moving features, potentially neighboring robots, and remove them from the estimation process. Those works use *Random-Finite-Sets* which were originally developed for multi-target tracking. This way, they manage to incorporate data association, landmark appearance and disappearance, missed detections, and false alarms in the filtering process. Nevertheless, handling dynamic landmarks remains an open topic given that most current works still rely on static environment assumptions.

2.9.4 Active C-SLAM

The concept of active C-SLAM comes from the powerful idea that while C-SLAM naturally improves path planning and control, path planning and control can also improve C-SLAM. Interestingly, some of the earliest works in collaborative localization were already leveraging coordination between robots to improve accuracy. Instead of mapping the environment, they relied on subsets of robots, in alternance, to serve as landmarks for the others [198, 199]. In an interesting turn of events, the recent progress in C-SLAM has brought back this active sensing trend to the forefront of research.

In C-SLAM, gains can be made by leveraging the coordination between the mapping robots. Having feedback loops to the C-SLAM algorithm allows path planning optimization for faster coverage and mapping of the environment [200, 201]. To achieve those goals, [202] aims to minimize the global exploration time and the average travelled distance among the robots. Other examples of the coupling between path planning and SLAM include [203], which shows the advantages of UAVs flying in formation for monocular C-SLAM, and [204] which uses deep Q-learning to decide whether a robot should localize the others or continue exploring on its own.

Active C-SLAM can also increase the estimation accuracy. To that end, [205] uses reinforcement learning to determine the best moment to merge the local maps, and [206] leverages instead the covariance matrix computed by the EKF-based inference engine to select trajectories that reduce the map uncertainty. Similarly, [207] develop a theoretical approach to design a sensor control policy which minimizes the entropy of the estimation task, while [208] proposes to broadcast the weakest nodes in the C-SLAM pose graph topology to actively increase the estimation accuracy.

Those works are most likely the mere beginning of active C-SLAM research given that C-SLAM systems are now being integrated on actual industrial, scientific or consumer robots, opening many possibilities of interaction between C-SLAM and other robotics subsystems.

2.9.5 Semantic C-SLAM

With the rise of deep learning and its impressive semantic inference capabilities, a lot of interest have been directed towards semantic mapping in which the environment is interpreted using class labels (i.e., person, car, chair, etc.). Representing maps as a collection of objects or semantic classes usually leads to much more compressed representations of the environment [209], and this can be especially beneficial for C-SLAM. Indeed, fewer landmarks and smaller maps are better suited to tight communication constraints since they reduce the amount of data sharing between the robots.

Semantic segmentation was first applied to C-SLAM in [210] which detects blobs of colors as salient landmarks in the robots maps. [90] later leverages deep learning-based object detection to perform object-based C-SLAM. However, such object-based techniques rely heavily upon the presence of many objects of the known classes in the environment (i.e., classes in the training data). Thus, they do not generalize well to arbitrary settings.

The other current preferred approach for semantic C-SLAM is to annotate maps of the environment with class labels. For example, [159, 211] use constellations of landmarks each comprised of a 3D point cloud, a class label and an appearance descriptor. The relatively small number of semantic landmarks reduces the required inter-robot communication significantly. [212] considers the joint estimation of object labels and poses in addition to the robots poses in order to improve both estimates. [144] build instead globally consistent local metric maps that are enhanced with local semantic labelling, hence preserving the accuracy of pure geometric C-SLAM approaches while incorporating useful high-level information in the robots individual maps.

The tremendous progress still occurring in the field of deep learning strongly suggests that there is more to come in terms of integration with C-SLAM and enhanced collaborative understanding of the environment.

2.9.6 Augmented Reality

Apart from the well known UAV or self-driving cars applications, Augmented Reality (AR) is probably one of the biggest field of application of SLAM. Indeed, SLAM makes markerless AR applications possible by building a map of the surrounding environment which is essential to overlay digital interactive augmentations. In other words, SLAM is required to make AR work in environments without motion capture, localization beacons or predetermined markers. In the foreseeable future, AR applications and games will push for multi-agent collaboration and this is where C-SLAM comes into play [213, 214]. To that end, [215] proposes a centralized approach in which virtual elements are shared by all agents, and [216] introduces a decentralized AR technique with smartphones, making use of the visual and inertial sensors already present in those devices. In a similar vein, [217] presents a resource-aware technique capable of trading off accuracy to adjust the computational cost to the available resources on mobile devices.

Some other techniques also look at the tremendous potential of collaborative AR for intuitive human-robot interfaces which is especially complex when the number of agents (i.e., humans or robots) and viewpoints increases. For example, to improve supervised mapping tasks, [218] equips a human operator with an AR system to edit and correct the map produced by a robot during a mission. Interestingly, [219] goes in the opposite direction: humans equipped with smartphones map an environment and get feedback from a central server to indicate which unscanned areas still need to be explored.

Augmented Reality might soon become the main application of C-SLAM in our daily lives, but there is still a lot of research work ahead to efficiently satisfy its inherent constraints and achieve robust large-scale deployments.

2.10 Conclusions

In this paper, we presented the core ideas behind Collaborative Simultaneous Localization and Mapping and provided a survey of existing techniques. First, we introduced the basic concepts of a C-SLAM system. We provided explanations and bits of historical context to better understand the astonishing progress recently made in the field. Then, we presented the building blocks of a typical C-SLAM system and the associated techniques in the literature. We also touched upon the difficulties of reproducibility and benchmarking. Afterwards, we explored new trends and challenges in the field that will certainly receive a lot more interests in the future. In summary, we focused on providing a complete overview of the C-SLAM research landscape.

We have shown, through numerous examples, how C-SLAM systems are varied and need to match closely the application requirements: sparse or dense maps, precise or topological localization, the number of robots involved, the networking limitations, etc. We wish for this survey to be a useful tool for C-SLAM practitioners looking for adequate solutions to their specific problems.

Nevertheless, despite the current growing interest for C-SLAM applications, it is still a young topic of research and many fundamental problems have to be resolved before the advance of C-SLAM-based commercial products. In particular, we believe that current systems scale poorly and are often limited to very few robots. So, a lot of work is still required to achieve large teams of robots building maps and localizing themselves collaboratively. We also note the growing interest for semantic C-SLAM to make robotic maps more interpretable and more actionable. Scene understanding techniques in the computer vision field could bring more compact and expressive environment representations into the SLAM system, which potentially increase the map readability while reducing the inter-robot communication burden. Furthermore, the rise of AR, in conjunction with C-SLAM and semantics, will offer incredible opportunities of innovation in the fields of collaborative robotics, mobile sensing, and entertainment.

CHAPTER 3 ADDITIONAL RECENT RESEARCH

This chapter extends the literature review presented in the previous chapter, focusing on new approaches and emerging trends in collaborative SLAM (C-SLAM) that have arisen in the two years since the survey's publication. These advancements reflect the rapid evolution of the field, particularly in areas such as place recognition, object-based map representations, state estimation, dataset availability, and the integration of foundation models.

Shortly after the publication of the initial survey, Cramariuc et al. [220] introduced a novel modular, multi-modal centralized C-SLAM framework. This adaptable framework has already been extended to support collaboration between augmented reality (AR) devices and robots, as shown by Chen et al. [221]. Such integration opens new possibilities for human-robot interaction, enabling AR devices to actively contribute to map-building and localization alongside robotic agents.

In the domain of place recognition, Dutto et al. [222] proposed a promising technique to train visual place recognition neural networks in a distributed manner using federated learning. This method enables teams of robots to collaboratively train a shared place recognition model while exploring their environment, enhancing adaptability and accuracy without the need for centralized data collection. In another line of work, Keetha et al. [223] and Ramtoula et al. [224] propose techniques that achieve state-of-the-art visual place recognition performance with minimal or no task-specific training. These methods leverage recent large self-supervised pretrained models [225], which have demonstrated strong generalization across diverse tasks.

A notable trend in recent years is the resurgence of object-based map representations, often structured as scene graphs that connect objects and locations through relational edges. When a sufficient number of recognizable objects are present in the environment, these scene graphs can be collaboratively constructed and serve as robust map representations for C-SLAM [226–228]. This approach provides a semantically rich map structure, enhancing both navigation and interaction capabilities in complex environments.

In terms of state estimation, several innovative methods have emerged to enhance the efficiency and scalability of C-SLAM. McGann et al. [229] introduced an incremental distributed pose graph optimization technique, enabling faster state estimation by avoiding the need to recompute the entire problem with each new measurement. Another notable advancement is from Murai et al. [230], who utilized Gaussian Belief Propagation [231] for fully distributed and scalable localization. This approach holds particular promise for swarm robotics, where large numbers of robots must achieve efficient and precise co-localization. While distributed methods still face challenges with slower convergence speeds compared to centralized approaches, the field is advancing rapidly. This progress suggests that these more scalable methods will play an increasingly important role in the future.

In support of SLAM and C-SLAM research across varied conditions and sensor configurations, several new datasets have recently been introduced. Feng et al. [232] released a dataset featuring three robots equipped with stereo and lidar sensors, operating in both indoor and outdoor scenes. This range of environments is essential for a comprehensive evaluation of C-SLAM systems performance. Zhu et al. [233] introduced a large-scale, multimodal dataset featuring fourteen robots, including both ground and aerial platforms, covering kilometer-long trajectories and providing high-quality ground truth data. This dataset incorporates precisely fused GNSS signals with high-fidelity IMUs, providing the accuracy necessary to track advancements as C-SLAM techniques approach sub-meter precision. Additionally, this dataset will be instrumental for the progress of C-SLAM with heterogeneous platforms, as merging maps from the differing perspectives of ground and aerial robots remains a significant challenge. Zhao et al. [234] contributed sequences recorded in degraded underground environments and under various weather conditions. These challenging environments particularly affect front-end processes by hindering the generation of accurate map measurements. Thus, this dataset is invaluable for improving the robustness of SLAM and C-SLAM systems in adverse conditions.

Finally, the substantial increase in computational power and the availability of internet-scale training datasets have spurred enthusiasm for foundation models—large-scale models pretrained on vast datasets that generalize effectively across a range of domains and environments. For SLAM and C-SLAM, foundation models have already demonstrated impressive capabilities in image matching and relative pose estimation [235–238]. The emergence of these models has been facilitated by the introduction of a novel, challenging dataset and benchmark for map-free relocalization [239]. Despite its name, map-free relocalization—defined as metric pose relocalization relative to a single image—holds considerable relevance for C-SLAM. The ability to infer a camera's metric localization from a single image, or a small set of images, could significantly improve the detection and computation of inter-robot loop closures, vital for map merging. Thus, these models could dramatically extend C-SLAM's applicability, enabling robots to operate across a wide variety of scenarios with minimal additional training or calibration, and marking a significant frontier in SLAM research.

CHAPTER 4 RESEARCH APPROACH

The research presented in this thesis is divided into five work packages (WPs), with each WP corresponding to a published or submitted article. These WPs align along three primary research axes, derived from the objectives outlined in Section 1.3:

- **Accuracy and Resilience**
- **Resource Efficiency**
- **Adaptability**

The relationships between these research axes and the work packages are illustrated in Fig. 4.1, emphasizing how each WP contributes to these overarching themes.

Below is a summary of the five work packages (WPs):

- **WP1: Self-Supervised Domain Calibration**
  This package (Chapter 5) presents methods for calibrating SLAM systems to adapt to new environments through self-supervised learning. It proposes domain adaptation and uncertainty estimation strategies, making SLAM solutions more robust when deployed outside controlled laboratory settings.
- **WP2: Swarm-SLAM - Sparse Decentralized Collaborative SLAM**
  This WP (Chapter 6) proposes a decentralized C-SLAM framework designed for ad-hoc networks of robots collaborating in unknown environments. It introduces a new inter-robot loop closure prioritization strategy based on recent advancements in graph theory. The goal is to reduce communication overhead and accelerate map estimation convergence within a multi-robot team.
- **WP3: Field Experiments in Planetary Analogue Environments**
  This WP (Chapter 7) documents real-world deployment of Swarm-SLAM in a planetary analogue setting. The experiments evaluate the framework under challenging conditions, identifying both practical limitations and potential future research avenues. It also introduces a novel dataset which include both sensor data for SLAM and inter-robot throughput and latency estimates.
- **WP4: MOLD-SLAM - Minimal Overlap Loop Detection SLAM**
  This work (Chapter 8) focuses on improving multi-robot SLAM by using foundation

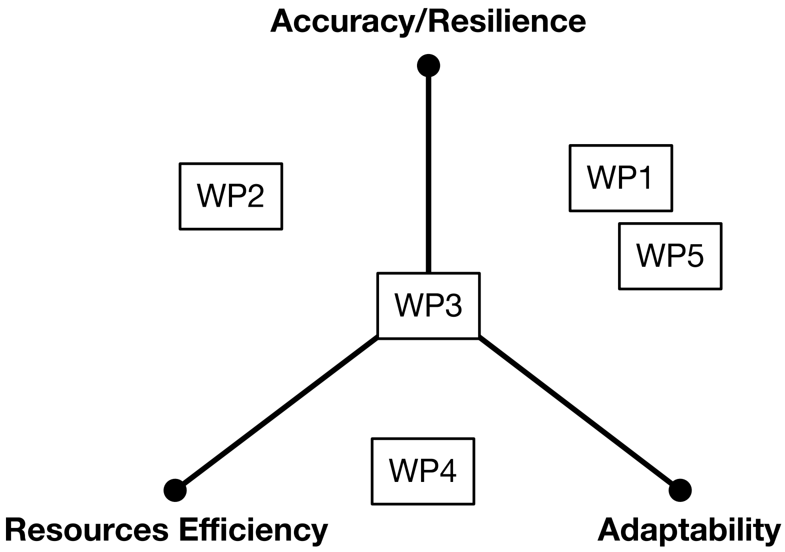


Figure 4.1 Work packages mapped along the three research axes.

models to detect loop closures even with minimal overlap between trajectories. The WP addresses the challenge of associating data from largely different viewpoints and trajectories, ensuring accurate map merging.

- **WP5: PEOPLEx - Pedestrian Opportunistic Positioning Framework**
  This WP (Chapter 9) introduces PEOPLEx, a framework leveraging various sensors (IMU, UWB, BLE, and Wi-Fi) to enable pedestrian positioning on commercial smartphones. It contributes to advancing SLAM applications beyond robotics.

4.1 Methodology

While each contribution addresses a different aspect of the SLAM problem, they follow a unified research methodology. The methodology is summarized in the following steps:

- **Problem Identification**:
  Identify real-world challenges in localization and mapping by engaging in practical deployments with robotic systems. This step involves understanding the constraints and operational difficulties encountered in realistic scenarios.

- **Formulate Hypotheses and Potential Solutions**:
  Develop hypotheses on the root causes of identified problems and propose potential solutions. This stage involves brainstorming and refining initial ideas through discussions

with collaborators.

- **Create a Minimal Viable Solution (MVS)**:
  Implement a basic version of the proposed solution to validate key hypotheses. Rapidly prototyping ideas helps to identify flaws early, ensuring efficient progress toward robust innovations.

- **Incremental Solution Development**:
  Build upon the initial MVS by incorporating insights from preliminary testing. Iterate through multiple cycles of design, validation, and testing to create a comprehensive solution suitable for real-world deployment.

- **Evaluate on Real-World Datasets**:
  Test the developed solutions using both public datasets and custom-collected data to ensure broad applicability. Perform evaluations against benchmark from the literature.

- **Deployment and Engineering Integration**:
  Deploy the solution on physical robots in real-world environments. This step ensures that the developed system performs reliably outside laboratory conditions.

- **Benchmarking and Comparative Analysis**:
  Compare the performance of the developed solution with existing baselines. In this thesis, benchmarking focuses on metrics such as localization accuracy, computation time, and communication bandwidth.

We closely adhered to the research methodology described above to ensure meaningful results and insights that can guide future research.

4.2 Document Structure

The remaining Chapters 5 to 9 present each contribution individually, with dedicated introduction, related work, methods, experiments, discussion and conclusion sections. Chapter 10 provides a general discussion tying all the work together and extracting insights from this body of research. It highlights the interconnections between different work packages and offers reflections on the broader implications of the findings. The discussion also outlines potential improvements and open challenges that could further advance the field. Finally, Chapter 11 concludes this thesis by summarizing the work presented and suggesting future research directions.

CHAPTER 5 ARTICLE 2 : SELF-SUPERVISED DOMAIN CALIBRATION AND UNCERTAINTY ESTIMATION FOR PLACE RECOGNITION

Preface: This article presents a self-supervised domain calibration method to improve visual place recognition in unfamiliar environments, using SLAM-based pose graph optimization without GPS or manual labeling. Our approach enhances both performance and uncertainty estimation, offering a practical solution for robust, safety-critical applications.

This work focuses on the single-robot SLAM problem but is readily applicable to multi-robot collaborative SLAM (C-SLAM). The following articles in this thesis address the C-SLAM problem more directly. This work has been peer-reviewed and was published in IEEE Robotics and Automation Letters (RA-L). The source code is available to the public at: `https://github.com/MISTLab/vpr-calibration-and-uncertainty`.

Contributions: My contributions to this article include conceptualizing the project after discussions with my supervisor, conducting a focused literature review on visual place recognition, developing the approach with my supervisor's feedback, carrying out the evaluation, and writing the majority of the paper. I was also responsible for implementing, releasing, and maintaining the accompanying open-source code.

Full Citation: Pierre-Yves Lajoie, Giovanni Beltrame, "Self-Supervised Domain Calibration and Uncertainty Estimation for Place Recognition," *IEEE Robotics and Automation Letters*, Vol.8, Issue 2, 2022.

Submission date: September 14th 2022

Publication date: December 26th 2022

DOI: 10.1109/LRA.2022.3232033

5.1 Abstract

Visual place recognition techniques based on deep learning, which have imposed themselves as the state-of-the-art in recent years, do not generalize well to environments visually different from the training set. Thus, to achieve top performance, it is sometimes necessary to fine-tune the networks to the target environment. To this end, we propose a self-supervised domain calibration procedure based on robust pose graph optimization from Simultaneous Localization and Mapping (SLAM) as the supervision signal without requiring GPS or manual labeling. Moreover, we leverage the procedure to improve uncertainty estimation for place recognition matches which is important in safety critical ap-

plications. We show that our approach can improve the performance of a state-of-the-art technique on a target environment dissimilar from its training set and that we can obtain uncertainty estimates. We believe that this approach will help practitioners to deploy robust place recognition solutions in real-world applications. Our code is available publicly: `https://github.com/MISTLab/vpr-calibration-and-uncertainty`

5.2 Introduction

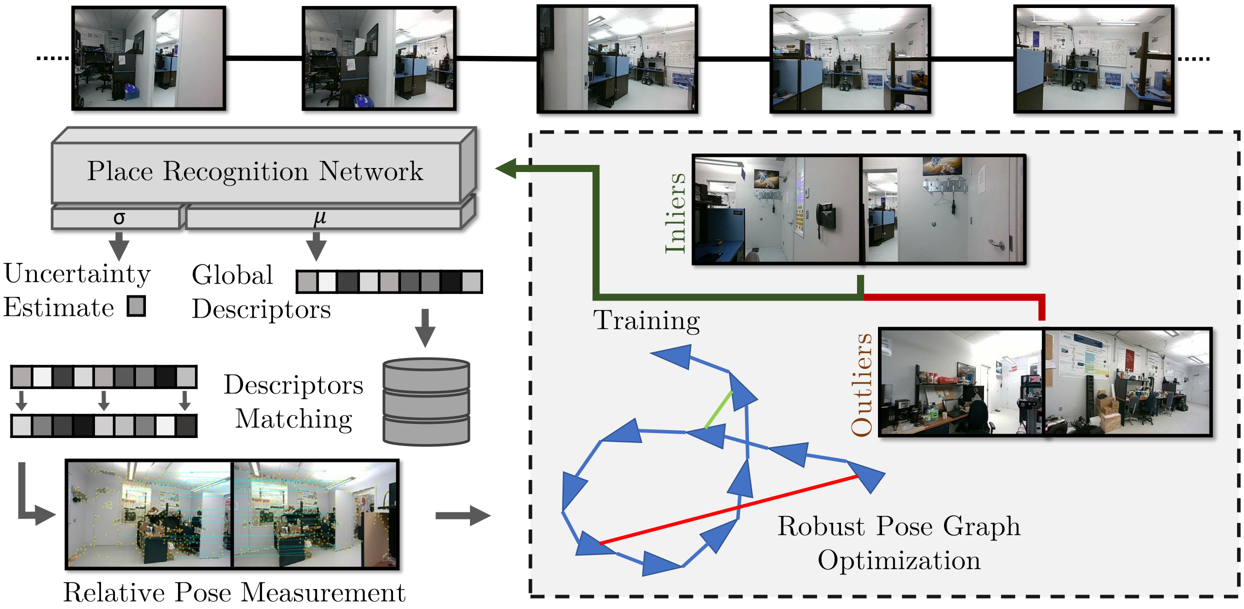


Figure 5.1 Self-Supervised Domain Calibration and Uncertainty Estimation via Robust SLAM: Using a single calibration sequence through a new environment, our proposed self-supervised technique for visual place recognition verifies putative loop closures using recent progress in robust pose graph optimization, and uses both the resulting inliers and outliers to fine-tune the place recognition network. The place recognition network tuned to the new domain achieves better performance on subsequent sequences in visually similar environments, and provides uncertainty estimates tailored to those environments. Our calibration approach does not rely on GPS or any ground truth information, and can thus improve place recognition systems in any environment.

Visual Place Recognition (VPR) remains one of the core problems of autonomous driving and long-term robot localization. Recognizing previously visited places is essential for decision-making, to reduce localization drift in Simultaneous Localization and Mapping (SLAM), and to improve robots' situational awareness in general [65]. While VPR techniques based on deep learning can achieve very high levels of accuracy on standard datasets [240], domain generalization is still a major concern when the deployment environment is visually and/or structurally different from the training data. The problem of domain discrepancies is especially important for indoors or subterranean deployments since most popular approaches are trained using GPS data on city streets images [72, 241, 242].

Generalization and feature transferability from one domain to another, are common issues in deep learning [243]. The most common and effective approach is still to calibrate or fine-tune, the representation to the testing domain. Given that most robots are deployed in known domains (e.g. roads, warehouses, etc.), one can refine the network using additional labeled samples directly from the known testing environment to tailor the representation to the target domain. While an effective approach, the data labelling necessary to obtain new training samples can be prohibitively expensive in practice.

To solve this problem, we believe that robust SLAM can be used as a self-supervised tool for data mining in any environment without the need for external sensors or ground truth information. In standard SLAM, place recognition errors are known to cause catastrophic localization failures. However, recent progress in robust state estimation has shown that such erroneous VPR matches can be detected and removed during pose graph optimization (PGO) [5, 140]. In other words, robust PGO leverages the 3D structure of the environment and robot trajectory to classify VPR matches as correct and incorrect. Both correct and incorrect matches can in turn be used to fine-tune VPR networks to improve their performance or obtain uncertainty estimates.

Therefore, in this paper, we propose a self-supervised domain calibration approach to extract new training samples from any target domains and improve VPR networks accuracy. In addition, we propose a technique to train an uncertainty estimator for place recognition using the new extracted samples.

First, we show that our self-supervised approach to gather training samples can be used to train a VPR network from a pretrained classification model and achieve reasonable performance, thus demonstrating the strength of our self-supervised training signal. We then show that our approach can improve the performance of existing VPR solutions when applied to environments that are visually different from their training domain, as well as providing uncertainty estimates.

Previous self-supervised approaches [72, 244] relied on GPS localization to extract training samples from datasets by selecting images with minimal distance. However, this is not suitable for GPS-denied environments such as indoor, underwater or underground. Also, contrary to techniques using structure-from-motion (SfM) for data mining [245], our approach leverages additional outlier samples identified with robust SLAM to further improve the VPR network. Moreover, our approach is able to extract samples in any environment in which odometry estimates can be obtained, leveraging sensors such as IMUs and wheel encoders that are not used in SfM.

Our approach offers practical benefits for the deployment of VPR systems in real applications: it could be used to collect correct and incorrect training samples from a single calibration run through an environment similar to the target domain, or could be employed online for lifelong learning/tuning directly on the target environment. After calibration, the VPR network is able to detect more correct matches and identify uncertain images. Moreover, by producing fewer incorrect matches, it reduces the expensive computational burden of processing and rejecting them [246]. Our contributions can be summarized as follows:

- A self-supervised training samples extraction method that does not require any external sensor (e.g. GPS), ground truth or manual labelling;
- A VPR sample classification method in correct and incorrect matches based on robust SLAM estimates;
- A domain calibration procedure for existing VPR techniques to improve their performance on any target environment using both correct and incorrect samples.
- An uncertainty estimator leveraging the new correct and incorrect samples during training;
- Open-source packages for sample extraction, network refinement and uncertainty estimation.

In the rest of this paper, Section 5.3 presents some background knowledge and related work, Section 5.4 details the proposed approach, Section 5.5 demonstrate the effectiveness of the technique, and Section 8.6 offers conclusions and discusses future work.

5.3 Background and Related Work

5.3.1 Visual Place Recognition

The ability to recognize places is crucial for localization, navigation, and augmented reality, among other applications [247]. The most popular approach is to compute and store global descriptors for each image to match, followed by an image retrieval scheme using a database of descriptors. Global descriptors are usually represented as high-dimensionality vectors which can be compared with simple distance functions (e.g. Euclidean or cosine distance) to obtain a similarity metric between two images. The seminal work of NetVLAD [72] extracts descriptors using a CNN and leverages Vectors of Locally Aggregated Descriptors [248] to get a representation well-suited for image retrieval. The descriptor network is typically trained using tuples of images mined from large datasets. An anchor image is first chosen, then positive and negative samples are selected based on close and far GPS localization, respectively. A triplet margin loss pushes the network to output similar representations for positive and anchor samples and dissimilar representations for negative ones. Recent work has extended the concept of global descriptors by extracting local-global descriptors from patches in the feature space of each image [241]. In another line of work, [242] proposed a Generalized Contrastive loss (GCL) function that relies on image similarity as a continuous measure instead of binary labels (i.e., positive and negative samples).

Recent works in place recognition have aimed at computing uncertainty estimates for individual samples (i.e., images) using an uncertainty-aware loss during training [249–251]. This loss function allows the system to reduce its confidence and the priority of samples with high uncertainty. Similar to standard place recognition, the uncertainty estimates are dependent on the training domain, meaning it is beneficial to train those estimates on the target domain. In this work, we use the Bayesian Triplet Loss from [250] for correct samples and a Kullback-Leibler divergence loss for incorrect samples as described in Section 5.4.4.

5.3.2 Robust SLAM

In SLAM, place recognition is used to produce loop closure measurements between the current pose (i.e., rotation and translation) of a robot and the pose corresponding to the last time it has visited the place. Loop closure measurements are combined with odometry (i.e., egomotion) measurements in a graph representing the robot/camera trajectory. In other words, the SLAM algorithm builds a pose graph with odometry links between subsequent poses and loop closure links between recognized places. Pose graph optimization is then performed to reduce the localization drift of the robot [1]. When using a global descriptor method, such as NetVLAD, VPR serves as a first filter through potential matches, which is followed by the more expensive task of feature matching and registration to obtain the relative pose measurement corresponding to the loop closure. Due to the occurrence of perceptual aliasing (i.e., when two distinct similar-looking places are confused as the same), some loop closure measurements are incorrect and, if left undetected, they can lead to dramatic localization failures [5]. This phenomenon is particularly important when computing loop closures between multiple robots maps for collaborative localization [252].

To mitigate the negative effect of incorrect loop closure measurements during pose graph optimization, several approaches have been proposed. They vary from adding decision variables to the optimization problem [5], to leveraging clusters in the pose graph structure [253]. For the purpose of this paper, we chose a recent approach based on Graduated Non-Convexity which as been shown to efficiently achieve superior results [140].

It is important to note that these approaches allow us indirectly to classify loop closure measurements, and by extension also VPR matches, as correct (inliers) or incorrect (outliers).

5.3.3 Domain Calibration

The goal of domain calibration is to improve the performance of a system on a target domain that is different from the training domain. This can be done through fine-tuning the model using samples from the target domain or through more complex domain adaptation approaches to enhance the generalization ability of the model. Domain calibration is also a major concern for long-term visual localization in changing environments. As presented in [254], one approach is to store multiple maps of the same environment to account for scene variation. To ensure the scalability, [255] proposes to summarize the maps, and [256, 257] suggest the use of compressed or coarse represention based on Hidden Markov Models.

Our approach could be use to adapt the VPR network to appearance changes. In fact, approaches in that line of research have been proposed for sequence adaption to cope with changing weather conditions during long-term missions [258, 259].

Interestingly, the exploitation of local feature patterns has been identified as a key to domain adaptation since they are more generic and transferable than global approaches [260]. Alternatively, recent work have proposed to include geometric and semantic information into the VPR latent embedding representation for visual place recognition [261] to better adapt to the target domain. In another line of work, [262] uses temporal and feature neighborhoods in panoramic sensor data to mine training samples for VPR fine-tuning: they classify samples as correct using geometric verification, as opposed to our work where we leverage recent progress in robust SLAM.

Unlike related techniques based on GPS data [244], our approach is suited to any environment in which an odometry system (i.e., visual inertial odometry, lidar-based odometry paired with VPR, etc.) can be deployed, such as indoors, subterranean, or underwater. Our approach also requires significantly less data than SfM-based approaches [245]. Moreover, we include incorrect matches corresponding to loop closing outliers in the learning process to avoid such occurrences in the improved network. In addition, by adding an uncertainty head to the VPR network and using a Bayesian Triplet Loss [250], we are able to train an estimator for the heteroscedastic aleatoric uncertainty [263] (i.e, uncertainty corresponding to a particular data input) using only the extracted samples.

5.4 Self-Supervised Domain Calibration and Uncertainty Estimation

The main challenge addressed by our method is to extract pseudo-ground truth labels for training images. To tune the model to the target domain, we need to gather positive and negative place recognition matches from a single preliminary run through the environment. The classic approach is to use external positioning systems (e.g. GPS) to identify images that where captured in the same location as positive samples and images captured in distant locations as negative samples [72, 244]. We aim to extend this data mining scheme to any environment, regardless of the availability of ground truth localization. Our process is split in three sequential steps. First, we perform SLAM on an initial run through the target environment with a camera, or robot. In particular, we compute visual odometry, and we gather putative VPR matches from an initial network that was not tuned to the specific environment. Second, we sort the putative matches as correct or incorrect samples using robust pose graph optimization. Third, we use all the resulting samples to fine-tune the VPR network to the target domain and train an uncertainty estimator. A summary of the method is illustrated in Fig. 5.1.

5.4.1 Finding VPR Matches

Global images descriptors can be complex structures such as Vector of Locally Aggregated Descriptors [248], used in NetVLAD [72], or simply the features extracted from the penultimate layer of a standard classification network [264]. The descriptors are represented as a vector $f(I_i)$, where f is the image representation extraction function and I_i the i_{th} keyframe. As keyframes are processed, we store the computed global descriptors in a database. Then, for each keyframe we query the best matches using nearest neighbors search, by sorting the global descriptors based on the Euclidean distance $d(q, I_i)$ between the query descriptor $f(q)$ and the other images descriptors $f(I_i)$. This results in a sorted list of the best putative VPR matches for each keyframe for the run through the environment. To avoid trivial matches in the same location, we do not consider matches with keyframes in the vicinity of the query.

5.4.2 Classifying Matches

To filter the VPR matches, we first compute the relative pose between the pairs of images and integrate this information, as loop closures, in the SLAM pose graph. For each keyframe, we compute the relative pose between itself and the first image in its associated list (the best match). If we are able to successfully compute a relative pose measurement (i.e., loop closure), we store the two images as a (anchor, positive) training sample. Otherwise, we repeat the process with the next best match in the list. To obtain the negative samples, we go through the remaining best VPR matches in the sorted list and select up to N images for which a loop closure cannot be computed due to a lack of keypoint correspondences. N is set according to the available GPU memory for training. This way, we ensure that we extract the negative samples that appear the most similar to the anchor, yet that are not sufficiently similar to compute a loop closure. In other words, we select the most valuable negative samples for training, since they represent invalid VPR matches made by the uncalibrated network. This results in training tuples (1 anchor, 1 positive, N negatives) for each keyframe in the sequence.

Given the possible occurrence of perceptual aliasing, the computability of a relative pose measurement between the current anchor and positive frames, is not enough to guarantee that it is a correct place recognition match (see Fig. 5.2). Thus, we add the computed relative pose measurements to the SLAM pose graph as a loop closure and perform robust estimation using the Graduated Non-Convexity method [140].

From the resulting optimized pose graph, we can compute the error associated with every measurement and classify the VPR matches as correct or incorrect. A large error means that the match is in contradiction with the geometric structure of the pose graph and therefore incorrect. We then sort the measurements into the subsets of training samples $S_{correct}$ and $S_{incorrect}$. The two subsets will be used with different loss functions during training.

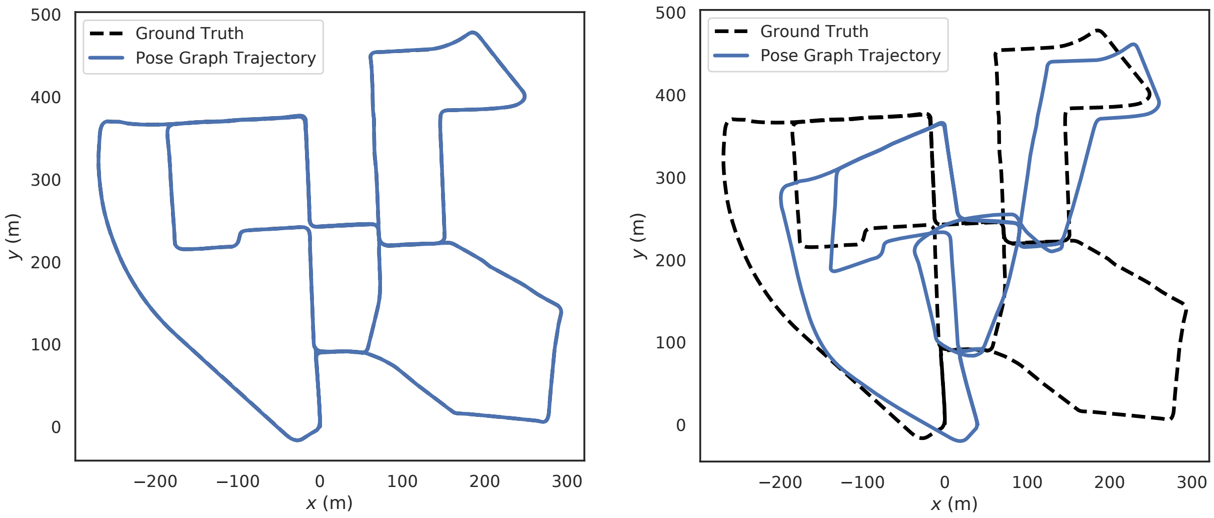
5.4.3 Domain Calibration

The domain calibration of our VPR network is done through fine-tuning using the filtered training tuples in the tuning sets. In other words, starting from the pretrained network, we performed additional training iterations using the extracted data.

For the subset of correct samples, we applied the triplet margin loss L for each training tuple $(q, p^q, \{n_i^q\}) \in S_{correct}$,

$$L = \sum_i \max(d(q, p^q) + m - d(q, n_i^q), 0) \tag{5.1}$$

where m is the margin, q is the global descriptor of the query image, p^q is the global descriptor of the positive image associated with the query, and n_i^q are the corresponding negative samples descriptors. The global descriptors are $1 \times K$ vectors resulting from a forward pass through the VPR network and the distance function d is the Euclidean distance between the vectors. This strategy is analog to the training method used in NetVLAD [72].



(a) With correct loop closures

(b) With 1 incorrect loop closure

Figure 5.2 Illustration of the resulting KITTI00 pose graphs with and without an incorrect loop closure. We can see the large negative effect of even a single incorrect measurement on the localization accuracy. This motivates the need to detect such incorrect VPR matches using robust pose graph optimization. Those incorrect matches correspond to some of the most confusing parts of the environment and can thus be used in training to futher improve VPR networks.

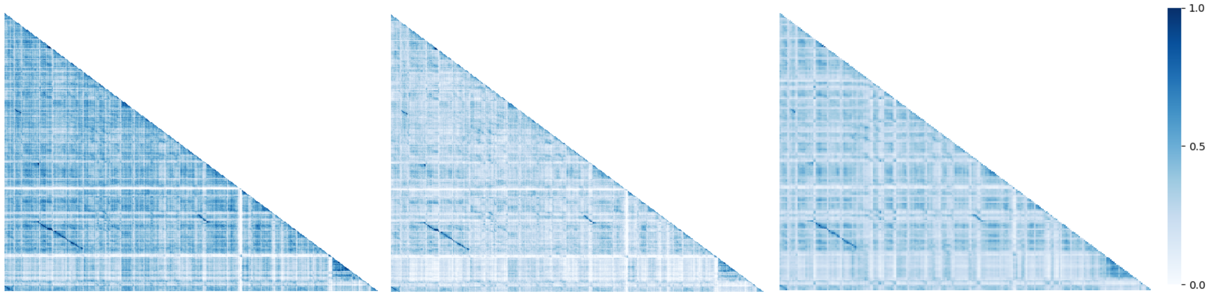
On the other hand, the incorrect samples $S_{incorrect}$ are composed of only one query and one negative images (q, n^q) and do not contain a positive image p such that we cannot use the triplet margin loss. Therefore, for each incorrect sample, we use a negative *Mean Squared Error* loss to increase the distance between the corresponding descriptors,

$$L = -\frac{1}{K}\sum_{k}^{K}(q_k - n_k^q)^2 \tag{5.2}$$

At each epoch we train on both correct and incorrect samples.

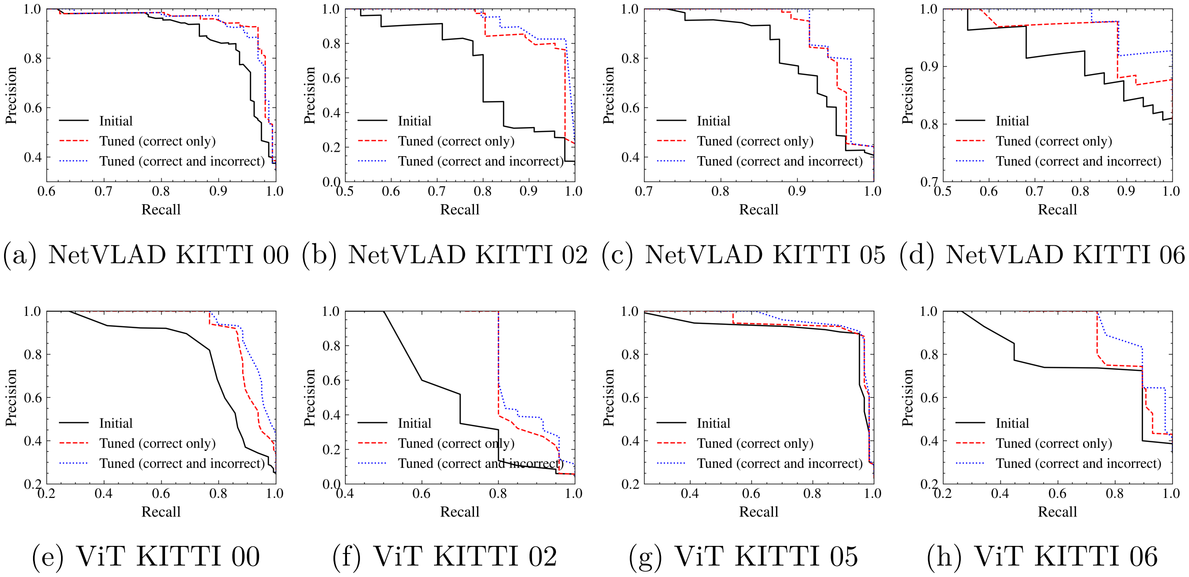
5.4.4 Uncertainty Estimation

To learn an estimator tuned to the desired target environment, we follow [250] and add an uncertainty head to the baseline CNN network composed of a Generalized Mean (GeM) layer [245] followed by two fully connected layers with a softplus activation function. The resulting network has two outputs, a mean μ and a variance σ, such that it encodes the descriptors as isotropic Normal distributions $\mathcal{N}(\mu, \sigma)$ instead of point estimates.



(a) Initial (b) Tuned (c) NetVLAD

Figure 5.3 Self-Supervised learning of a visual-similarity metric. An illustration of the similarity matrix before (Initial) and after (Tuned) training compared with the similarity obtained from NetVLAD on the KITTI-00 sequence. As expected, the similarity between positive pairs has increased (blue), and it has decreased between negative pairs (white).



(a) NetVLAD KITTI 00 (b) NetVLAD KITTI 02 (c) NetVLAD KITTI 05 (d) NetVLAD KITTI 06

(e) ViT KITTI 00 (f) ViT KITTI 02 (g) ViT KITTI 05 (h) ViT KITTI 06

Figure 5.4 Precision-Recall performance in loop-closure detection on various KITTI sequences with two different network architecture: NetVLAD and ViT. As expected, we can see that the networks tuned on the KITTI-360-09 sequence achieves better precision and recall on all sequences.

For each correct sample, the uncertainty-aware loss computes the probability that the query q is closer to a positive p than a negative n given a margin m, and a prior $1/K$ for normalization,

$$P\left(\|q-p\|^2 < \|q-n\|^2 - m\right), \tag{5.3}$$

For incorrect samples, we use instead the Kullback-Leibler divergence $D_{KL}(V \| T)$ loss between the descriptors distribution estimates ($V \in \mathcal{N}(\mu, \sigma)$) and a target high variance distribution with the same mean ($T \in \mathcal{N}(\mu, \sigma_{high})$) for which we have set the variance σ_{high} to H times the prior. A higher H increases the incorrect matches loss and thus their importance during training. For isotropic Normal distributions, $D_{KL}(V \| T)$ is defined as follows [265],

$$D_{KL}(V \| T) = \frac{1}{2} \left(\log \frac{\sigma_{high}^2}{\sigma^2} + \frac{\sigma^2}{\sigma_{high}^2} - 1 \right) \tag{5.4}$$

We sum the Kullback-Leibler divergences of the query q ($D_{KL}(V_q \| T)$) and the negative n^q ($D_{KL}(V_{n^q} \| T)$) to obtain the combined loss L of the incorrect sample $s \in S_{incorrect}$,

$$L = \frac{1}{2} \left(D_{KL}(V_q \| T) + D_{KL}(V_{n^q} \| T) \right) \tag{5.5}$$

The intuition behind the use of this loss function is to increase the variance estimates of both the query and negative images towards a higher variance without changing their means, since those images are confusing for the VPR system and led to loop closing outliers.

It is important to note that training using uncertainty-aware losses can have detrimental effects on the resulting precision of the place recognition network [249]. However, as we show in the following section, a variance estimator with reasonable performance can be trained on a separate smaller network that can be run cheaply on a CPU. This could allow practitioners to keep the precision of a conventionally trained VPR network and run a smaller uncertainty estimation network in parallel.

5.5 Experiments

Our experiments are divided into three parts. We first demonstrate the quality of the extracted training tuples by using them to train a network for the task of place recognition from a classification baseline. Second, we demonstrate that we can calibrate a state-of-the-art VPR approach for a target domain using our technique. Third, we show that we can achieve uncertainty estimates tailored to the target environment. All the hyperparameters values used in our experiments can be found in our open-source implementation and correspond to the ones used in [72] and [250].

5.5.1 Training a new VPR System

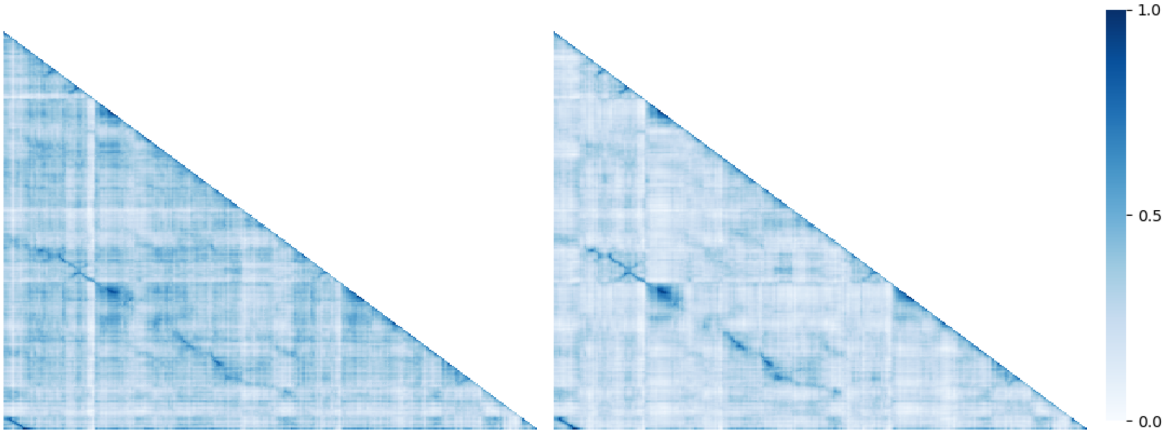
To show the effectiveness of our approach to produce valuable tuning samples from a calibration run through an environment, we trained a new VPR network based exclusively on the samples extracted from a single KITTI-360 [266] sequence and we tested the resulting VPR network on KITTI [176] sequences. All the sequences were collected in the streets of the same mid-size city. We used the KITTI-360-09 sequence for tuning, and the KITTI-{00, 02, 05, 06} sequences for testing. The testing sequences were collected years apart from the tuning sequence and were selected based on the significant overlaps within their trajectory, which are essential to recognize places.

Our initial model consist of a VGG16 network pretrained on ImageNet [267] for which we replaced the classification head with a randomly initialized NetVLAD pooling layer [72]. To show the generalization of the approach on different network architectures, we also tuned a Vision Transformer (ViT) [268] using the penultimate layer features as descriptors.

The new networks were tuned for place recognition using a triplet margin loss for 10 epochs, which was enough to achieve convergence. The relative poses, loop closures, are estimated with stereo pairs and the SLAM visual odometry is computed and managed using RTAB-Map [269]. Our technique successfully extracted 291 training tuples in $S_{correct}$, and 49 in $S_{incorrect}$, from the tuning sequence.

To validate the training procedure, we computed the similarity score, based on the L_2 distance between global descriptors, of all pairs of images in KITTI-00 sequence. In Fig. 5.3, we compared the resulting similarity matrix with the one before tuning and the one obtained with NetVLAD. NetVLAD, which is pretrained on city streets images, is known to achieve high accuracy on the KITTI sequences [37]. We can see that our approach converges to a similar result as NetVLAD, especially in the zones where multiples places are revisited and recognized (i.e., high similarity) near the bottom left and right corners (darker blue). The contrast with negative matches is also accentuated.

Using the ground truth poses of the KITTI sequences [176], we computed the precision and recall of our VPR system resulting matches before and after tuning, with or without the incorrect matches. The curves in Fig. 5.4 represent the performance for varying detection threshold values for loop closure detection. A loop closure is considered as detected if the distance between the two images global descriptors is inferior to the threshold, there exists sufficient keypoints matches to compute a relative pose measurement, and it passes the test of robust pose graph optimization. We can see a clear improvement in precision and recall after tuning both NetVLAD and ViT. We also see some small improvements when using the



(a) Original NetVLAD

(b) Tuned NetVLAD

Figure 5.5 Self-Supervised domain calibration of a visual-similarity metric. An illustration of the similarity matrix before (Original NetVLAD) and after training (Tuned NetVLAD) compared. As expected, the similarity between positive pairs has increased (blue), and it has decreased between negative pairs (white).

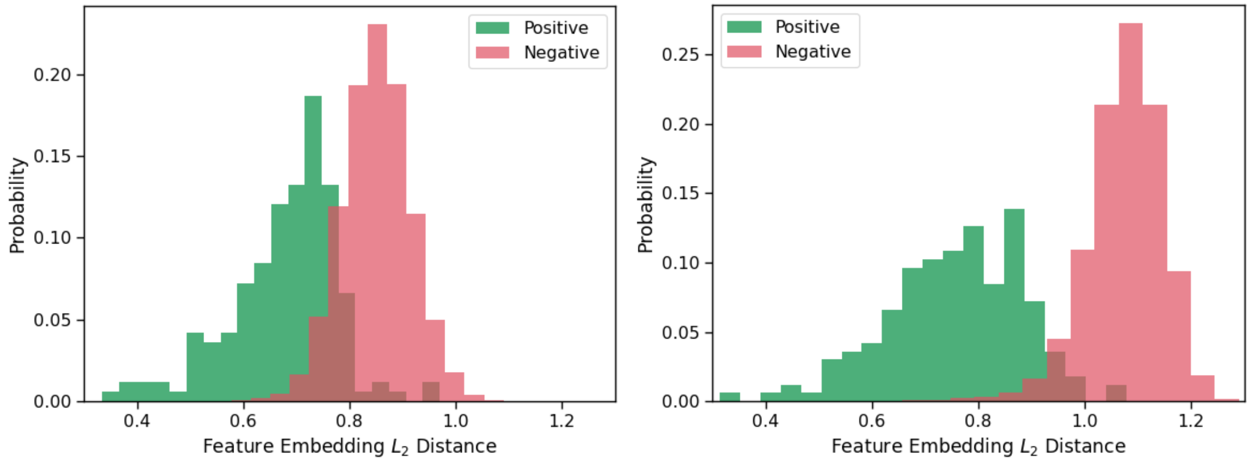
Table 5.1 Average percentage and standard deviation of correct matches obtained by NetVLAD before and after tuning. We can see that the domain calibration increased the percentage of correct matches and thus the number of loop closures.

	Indoor 1	Indoor 2	Indoor 3
Original NetVLAD	$65.4 \pm 8.9\%$	61.7 ± 12.4 %	75.8 ± 5.9 %
Tuned NetVLAD	$72.0 \pm 8.7\%$	71.1 ± 10.6 %	82.6 ± 4.8 %

incorrect matches, especially on the KITTI 06 sequence, however we expect the effects of incorrect matches to be greater for initial networks with low precision since it should reduce the number of false positives. While state-of-the-art results were not expected on the well-studied KITTI dataset, we are able to demonstrate the quality and efficiency of the gathered training samples from a single run through the environment without manual labelling or GPS bootstrapping.

5.5.2 Calibration of an Existing VPR system

In this set of experiments, we demonstrate that we can improve the performance of a pre-trained state-of-the-art VPR network (NetVLAD [72]) by tuning it to a different target domain. We performed four runs through an indoor office environment (see images in Fig. 5.1) using an Intel Realsense D455 camera, with multiple overlaps to ensure place recognition.



(a) Original NetVLAD

(b) Tuned NetVLAD

Figure 5.6 Separation distance calibration. Histograms of the L_2 distance between the positive pairs (green) and negative pairs (red). As expected, the separation increased between the positive pairs and negative pairs after tuning, making it easier to set a VPR threshold.

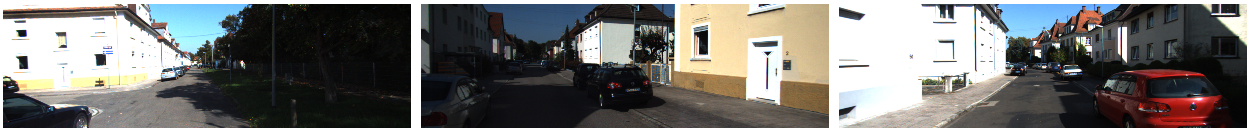


Figure 5.7 Examples of images with high estimated uncertainty. We can see the presence of under/overexposure and occlusions.

The first run served to extract 166 training tuples, and the three others have been used for testing.

As shown in Fig. 5.5, we computed the similarity score for each image pairs in the training sequence before (i.e., original pretrained version of NetVLAD on the Pittsburg dataset [72]) and after tuning. We can observe a significant improvement in contrast between positive and negative matches hinting to a better distinction between them during testing.

We corroborate this result in Fig. 5.6 which shows histograms of the L_2 distance between all pairs of images. The positive pairs are noted in green and the negative ones in red. Confirming the previous result, the separation between the positive and negative pairs is greater after tuning. Fig. 5.6 shows that our calibration technique is able to fine-tune the VPR network and distort the feature embedding to increase the distance between similar and dissimilar places. This has practical implications for the deployment of VPR systems since the threshold to determine if the distance represents a match becomes easier to set. Moreover, our approach leads to fewer false positives which can be detrimental to the system

Table 5.2 Uncertainty Estimation results on KITTI sequences. We report the F_1 score (computed from precision and recall), the Expected Calibration Error using the cosine similarity between the descriptors as uncertainty estimates (ECE_{sim}), and the ECE using our trained uncertainty estimates (ECE_{ours}) for each sequence with 2 different CNN backbone networks. We also indicate the networks size as well as their respective inference time on GPU (NVIDIA RTX3070) and CPU (AMD Ryzen 7).

	KITTI 00			KITTI 02			KITTI 05			KITTI 06			Resources		
	F_1 score	ECE_{sim}	ECE_{ours}	F_1 score	ECE_{sim}	ECE_{ours}	F_1 score	ECE_{sim}	ECE_{ours}	F_1 score	ECE_{sim}	ECE_{ours}	Size (MB)	GPU (s)	CPU (s)
VGG16	0.848	0.774	**0.186**	0.815	0.801	**0.226**	0.821	0.769	**0.125**	0.732	0.802	**0.305**	82.0	0.004	0.052
MobileNetv3	0.874	0.949	**0.242**	0.704	0.946	**0.150**	0.779	0.948	**0.197**	0.720	0.945	**0.367**	23.8	0.007	0.015

accuracy and computational performance [246]. In Table 5.1 we confirm on the three test sequences that NetVLAD tuned with our technique obtains on average a significantly higher number of correct VPR matches over possible threshold values than its original version (t-test, Bonferroni-corrected, $p <$ 1e-5). Therefore, our approach allows practitioners to improve the performance of their VPR system by calibrating its domain through a single run of the environment.

5.5.3 Uncertainty Estimation for VPR

As expected, results in Table 5.2 show that training a network explicitly for uncertainty estimation performs better than directly using the cosine similarity between descriptors as a confidence measure. Also, we noticed that training with an uncertainty loss does not provide as much improvement in precision and recall as the triplet margin loss. However, having uncertainty estimates is preferable in safety critical applications. One could even combine a network trained for state-of-the-art precision and another trained for uncertainty in the same system if the computing resources are sufficient. To that end, we show that we can achieve reasonable results in uncertainty estimation with a smaller backbone network (MobileNetv3 [270]) which can be run in real-time on a CPU. This decoupling offers flexibility for practical deployments of uncertainty-aware visual place recognition.

In Fig. 5.7, we present some examples of images with high estimated uncertainties by our technique. The uncertain images are under/overexposed and have occlusions, such that very few useful visual features and keypoints could be extracted from them in order to successfully compute 3D registration.

To measure the accuracy of the uncertainty estimates, we use the *Expected Calibration Error* (ECE), commonly used for classification tasks [271], which computes how well the uncertainty estimates correspond to the model's precision,

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left| \mathrm{mAP@1}\left(B_m\right) - \mathrm{conf}\left(B_m\right) \right| . \tag{5.6}$$

As in [251], we compute this metric by dividing the uncertainty estimates into M equally spaced bins B_m with corresponding uncertainty level $U(B_m)$. For each bin, we compute the precision of the queries it contains, $\mathrm{mAP@1}\left(B_m\right)$, and compare it with the bin confidence $\mathrm{conf}\left(B_m\right) = 1 - U(B_m)$. The ECE is low when the high confidence images lead to high precision matches. As expected, the resulting ECEs in Table 5.2 are in a similar range as the results presented in the Bayesian Triplet Loss paper [250]. Interestingly for resource-constrained deployments, the smaller MobileNetv3 achieves comparable results to the larger VGG16 while being able to evaluate images in real-time at more than 60Hz on a CPU.

5.6 Conclusions

In this paper, we present a self-supervised method for training and tuning a place recognition neural network leveraging robust SLAM which does not require GPS or ground truth labels for bootstrapping. We demonstrate the efficiency of the method by training a visual place recognition network from a pretrained classification model, using only the training samples extracted by our method. We also show that our technique can improve the accuracy of an existing deep learning-based VPR system by calibrating it to the target environment. In addition, we show that we can train an uncertainty estimation network for place recognition using the extracted samples.

We consider that our approach has practical benefits for the real-world deployment of place recognition systems. It could be used in an online fashion to perform lifelong learning/tuning on the target environment. Our approach has also the potential for data mining of labeled place recognition training samples on any sequential dataset, which could help increased the overall accuracy of VPR networks. Moreover, while we applied our technique to visual sensors, the same approach could be used for other type of sensors used for place recognition (e.g. lidars). Finally, we believe that leveraging the recent progress in robust SLAM to improve the performance of deep learning based techniques is a promising avenue that could lead to a tighter integration between the two fields of research.

CHAPTER 6 ARTICLE 3 : SWARM-SLAM: SPARSE DECENTRALIZED COLLABORATIVE SIMULTANEOUS LOCALIZATION AND MAPPING FRAMEWORK FOR MULTI-ROBOT SYSTEMS

Preface: This paper introduces a novel approach to Collaborative Simultaneous Localization and Mapping (C-SLAM), a critical technology for enabling effective multi-robot operations. This work has been peer-reviewed and was published in IEEE Robotics and Automation Letters (RA-L). The source code is available to the public at: `https://github.com/MISTLab/Swarm-SLAM`.

Contributions: This project began with conceptualization sparked by insightful discussions with my supervisor and fellow researchers at the ICRA 2022 conference. My contributions include conducting a focused literature review on inter-robot loop closure detection, developing the approach with my supervisor's feedback, performing dataset evaluations, deploying the approach on three robots, and writing the majority of the paper. I was also responsible for implementing, releasing, and maintaining the accompanying open-source code.

Full Citation: Pierre-Yves Lajoie and Giovanni Beltrame, "Swarm-SLAM: Sparse Decentralized Collaborative Simultaneous Localization and Mapping Framework for Multi-Robot Systems," *IEEE Robotics and Automation Letters*, Vol.9, Issue 1, 2023.

Submission date: July 15th 2023

Publication date: November 17th 2023

DOI: 10.1109/LRA.2023.3333742

6.1 Abstract

Collaborative Simultaneous Localization And Mapping (C-SLAM) is a vital component for successful multi-robot operations in environments without an external positioning system, such as indoors, underground or underwater. In this paper, we introduce Swarm-SLAM, an open-source C-SLAM system that is designed to be scalable, flexible, decentralized, and sparse, which are all key properties in swarm robotics. Our system supports lidar, stereo, and RGB-D sensing, and it includes a novel inter-robot loop closure prioritization technique that reduces communication and accelerates convergence. We evaluated our ROS 2 implementation on five different datasets, and in a real-world experiment with three robots communicating through an ad-hoc network. Our code is publicly available: `https://github.com/MISTLab/Swarm-SLAM`

6.2 Introduction

Collaborative perception is an important problem for the future of robotics. The shared understanding of the environment it provides is a prerequisite to many applications from autonomous warehouse management to subterranean exploration. One of the most powerful tools for robotic perception is Simultaneous Localization And Mapping (SLAM) which tightly couples the geometric perception of the environment with state estimation [1]. In addition to producing high-quality maps of the robot surroundings, it provides localization estimates that are essential for planning and control. However, single-robot SLAM estimates are local in the individual robot reference frame. Therefore, when multiple robots operate in GPS-denied environments, they do not share situational awareness unless they manage to connect, or merge, their local maps. To solve this problem, Collaborative SLAM (C-SLAM) searches for inter-robot map links and uses them to combine the local maps into a shared global understanding of the environment. One of the main practical challenges in C-SLAM is resource management [252], in particular considering the severe communication and computation limitations of mobile robots. Those limitations need to be addressed to achieve real-time performance, especially when a large number of robots work together. While effective in some scenarios, centralized C-SLAM solutions, which rely on a single server for data association and optimization, suffer from a communication bottleneck between the robots and the server, which limits their scalability. Besides, due to networking coverage challenges in large indoor or subterranean environments, robots cannot realistically maintain a stable connection to a central server. Thus, decentralized solution relying only on occasional communication between the robots are better suited for large-scale deployment. While collaborative perception within small teams of autonomous robots is currently challenging, we believe it useful to look forward to very large teams, or swarms of robots and start tackling the problems specific to this scale of deployment. Prior works on swarm robotics have identified a few key properties required for swarm compatibility [36] such as: communication and sensing must be local to the robot neighborhood, and robots should not rely on a centralized authority or global knowledge. In the specific case of C-SLAM [20], we consider the following four properties described in Sections 6.4 and 6.5: scalability, flexibility, decentralization, and sparsity.

In this paper, we propose novel techniques assembled in a complete resource-efficient C-SLAM framework compliant with these key swarm compatibility properties. Our approach is fully decentralized, supports different types of sensors (stereo cameras, RGB-D cameras, and lidars), and requires significantly less communication than previous techniques. To reduce data exchanges, we introduce a novel budgeted approach to select candidate inter-robot loop closures based on algebraic connectivity maximization, inspired from recent work

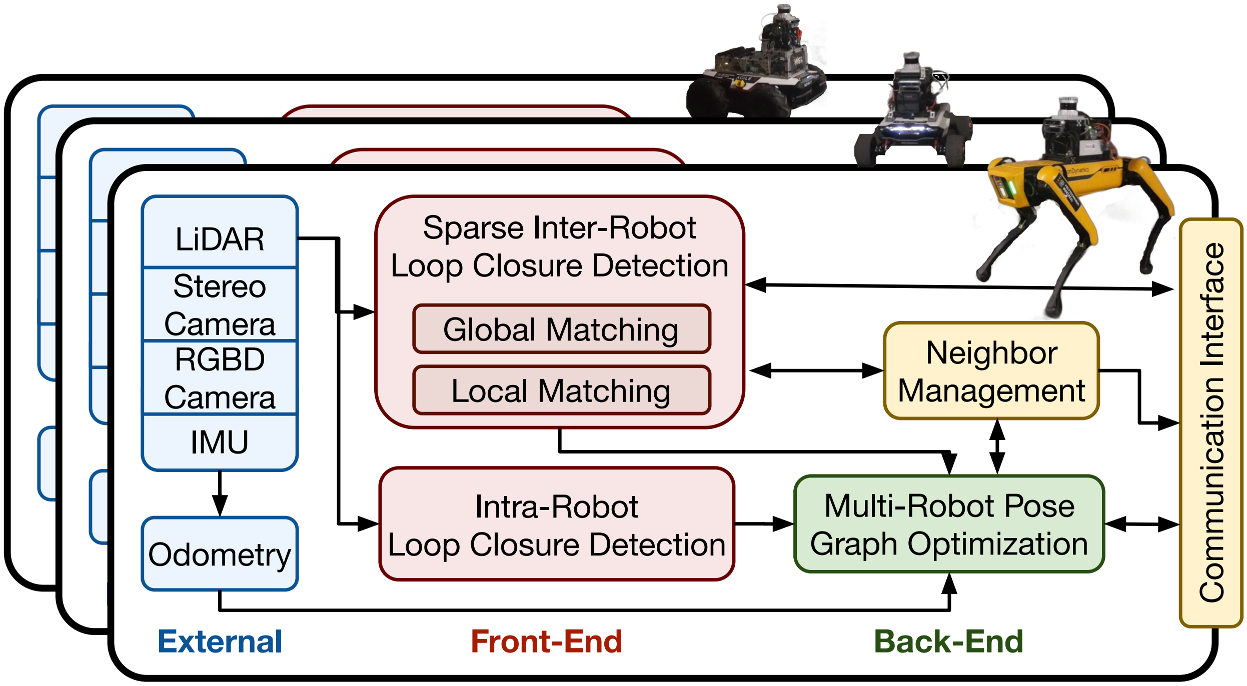


Figure 6.1 Swarm-SLAM Overview

on pose graph sparsification [272]. This preprocessing of place recognition matches allows us to achieve accurate C-SLAM estimates faster and using fewer communication resources. Moreover, we leverage advances in robotic software engineering, to make our framework compatible with ad-hoc networks. In summary, we offer the following **contributions**:

- A sparse budgeted inter-robot loop closure detection algorithm under communication constraints based on algebraic connectivity maximization;
- A decentralized approach to neighbor management and pose graph optimization suited for sporadic inter-robot communication;
- A swarm-compatible open-source framework which supports lidars, as well as stereo or RGB-D cameras;

We extensively evaluate of the overall system performance on datasets and in a real-world experiment.

6.3 Background and Related Work

6.3.1 Collaborative SLAM

C-SLAM systems can usually be divided into two categories: centralized and decentralized. Centralized systems rely on a remote base station to aggregate map data and compute the global SLAM estimates for all the robots. However, in those systems, the robots need a reliable permanent connection with the base station, and the scalability is severely limited by the communication bottleneck to the central server. Such stringent networking constraints are often unrealistic, especially in large environments. Decentralized approaches, relying only on occasional communication links between robots and without any need for a central authority, are preferred in those scenarios. However, decentralized systems are limited by the onboard computation and communication capabilities of the robots, and they require more sophisticated data management and bookkeeping strategies to obtain accurate SLAM estimates [252]. Similar to single robot SLAM systems, C-SLAM contains two parts commonly named front-end and back-end, see Fig. 6.1. The front-end is in charge of feature extraction and data association, while the back-end performs state estimation [1].

Front-End

The most challenging step in the C-SLAM front-end is the detection and computation of inter-robot loop closures in a resource-efficient manner. Inter-robot loop closures correspond to common features or places previously visited by two or more robots. Those shared features between the robots maps act as stitching points to merge the local maps together and obtain a shared (global) reference frame.

Since the communication cost of sharing entire maps is usually prohibitive, inter-robot loop closure detection can be performed in two stages [37, 62]. In the first stage, compact global descriptors of images [273] or lidar scans [274], are shared between the robots for place recognition. Similarity scores are computed between the global descriptors from both robots to recognize places, or overlaps, between their respective maps. The recognized places then correspond to loop closure candidates for the second stage. In the second stage, for each candidate with high global descriptors similarity, the corresponding costly local descriptors such as 3D keypoints or scans are transmitted to compute the geometric registration between the two robots images or scans.

Back-End

The role of the C-SLAM back-end is to estimate the most likely poses and map from the noisy measurements gathered by all robots. To this end, Choudary et al. [53] propose the distributed Gauss-Seidel (DGS) technique which allows robots to converge to a globally consistent local pose graph by communicating only the pose estimates involved in inter-robot loop closures, and therefore preserving the privacy of their whole trajectories. Tian et al. [126] significantly improve on that approach and provide a certifiably correct distributed solver for pose graph optimization. This technique performs multiple exchanges between the robots until they converge to globally consistent local solutions. In a different vein, recent work by Murai et al. [230] laid the foundation for larger-scale multi-robot collaborative localization based on Gaussian Belief Propagation. One of the main challenges in both single-robot and collaborative SLAM is the frequent occurrence of erroneous measurements among inter-robot loop closures due to perceptual aliasing [5]. While many techniques exist for the single-robot problem, Lajoie et al. [62] first combined DGS with Pairwise Consistency Maximization (PCM) [143], which computes the maximal clique of pairwise consistent inter-robot measurements, to perform robust and distributed optimization. More recently, Yang et al. [140] introduced the Graduated Non-Convexity (GNC) algorithm, a general approach for robust estimation on various problems including pose graph optimization. GNC was integrated with [126] in a robust distributed solver (D-GNC) [63].

Open-Source C-SLAM Systems

Many open-source C-SLAM systems have been proposed in the recent years. Cieslewski et al. [37] introduce DSLAM, which uses CNN-based global descriptors for distributed place recognition, and DGS for estimation. DOOR-SLAM [62] robustified the approach by integrating PCM for outlier rejection and adapted it for sporadic inter-robot communication. DiSCo-SLAM [275] extends those ideas to lidar-based C-SLAM using ScanContext global descriptors [274]. Kimera-Multi [63] integrates D-GNC and incorporates semantic data in the resulting maps. In an other line of work, centralized C-SLAM system have also evolved considerably. The lidar-based system LAMP 2.0 [276] introduces a centralized Graph Neural Network-based prioritization mechanism to predict the outcome of pose graph optimization for each inter-robot loop closure candidates. The multi-modal maplab 2.0 [220] supports heterogeneous groups of robots with different sensor setups. In contrast, Swarm-SLAM combines the latest advances from previous frameworks and introduce a new sparse inter-robot loop closure prioritization to further reduce communication. Additionally, unlike previous techniques, Swarm-SLAM leverages ROS 2 [277] and introduces a neighbor management sys-

Table 6.1 Open-Source C-SLAM Frameworks supporting stereo cameras (s), lidars (l) and/or RGB-D cameras (d).

	Sensor	Decentralized	Robust	Sporadic	Sparse
DSLAM [37]	s	✓			
DOOR-SLAM [62]	s,l	✓	✓	✓	
Kimera-Multi [63]	s	✓	✓	✓	
Disco-SLAM [275]	l	✓	✓	✓	
LAMP 2.0 [276]	l		✓		✓
maplab 2.0 [220]	s,d,l		✓		
Swarm-SLAM	s,d,l	✓	✓	✓	✓

tem to seamlessly integrate C-SLAM with ad-hoc networking. Table 6.1 offers a comparison of the various systems based on key desirable properties. We refer the reader to [252] for a thorough survey on C-SLAM.

6.3.2 Graph Sparsification for C-SLAM

The ever-growing map and pose graph during long-term operations is an important memory and computation efficiency challenge in both single-robot and multi-robot SLAM. One favored solution is graph sparsification, which aims to approximate the complete graph with as few edges as possible, mainly by removing redundant edges that are not providing new information during the estimation process. To this end, Doherty et al. [272] formulate the graph sparsification of single robot pose graphs as a *maximum algebraic connectivity augmentation* problem, and solve it efficiently using a more tractable convex relaxation. In this paper, instead of sparsifying the pose graph after all the measurements have already been computed, we aim to preemptively sparsify the inter-robot loop closure candidates generated by the place recognition module. This way, we can prioritize the geometric verification of inter-robot loop closures that will approximate the full pose graph, thus avoiding wasting resources on redundant measurements. Importantly, unlike other work maximizing the determinant of the information matrix [156], we leverage the results from [272] and focus on the algebraic connectivity of the pose graph which has been shown to be a key measure of estimation accuracy [278] (i.e., higher algebraic connectivity is associated with lower estimation error). Solving a similar problem, Denniston et al. [279] prioritize loop closure candidates based on point cloud characteristics, the proximity of known beacons, and the information gain predicted with a graph neural network. Interestingly, Tian et al. [280] explore spectral sparsification in the C-SLAM back-end to reduce the required communication during distributed pose graph optimization.

6.4 System Overview

As described in Fig. 6.1, Swarm-SLAM is composed of three modules. First, to enable decentralization, the neighbor management module continuously tracks which robots are in communication range (i.e., neighbors that can be reached reliably) and what data has been exchanged. Robots publish heartbeat messages at a fixed rate such that network connectivity can be evaluated periodically. To make the system scalable (see Property 1), the other modules query the neighbor management process to determine which robots, if any, are available, and orchestrate the operations.

Property 1. ***Scalable****. The number of robots using the framework is not predetermined and it does not require connectivity maintenance during the whole mission. Communication and computation budgets are set to fit the available bandwidth and computation power onboard the individual robots.*

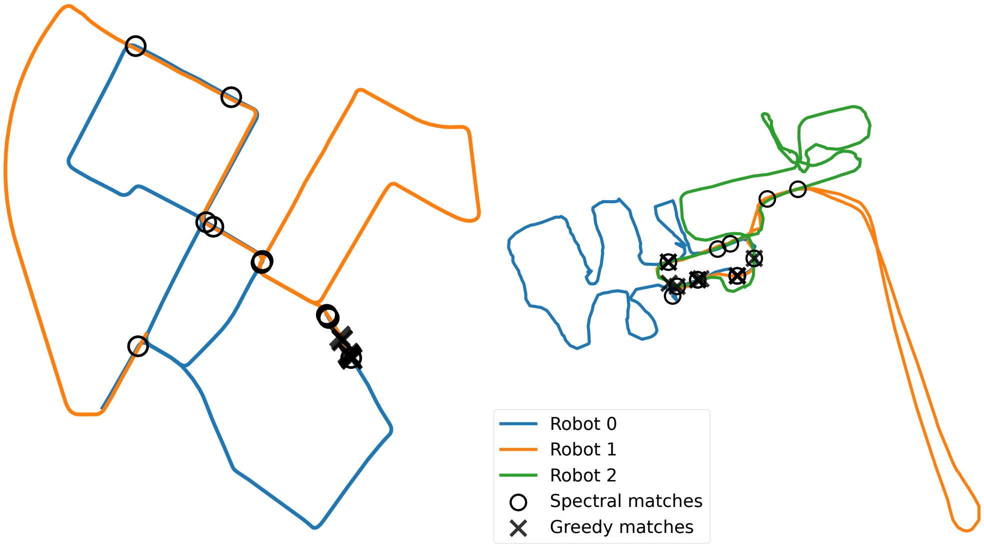
The front-end takes as input odometry estimates (obtained using an arbitrary technique) along with synchronized sensor images or pointclouds (see Property 2). Upon reception, the front-end extracts global (e.g. compact learned representation) and local descriptors (e.g. 3D keypoints). Global descriptors allow us to identify candidate place recognition matches (i.e., loop closures) between the robots, then local descriptors are used for 3D registration.

Property 2. ***Flexible****. The framework supports multiple sensors (i.e., stereo cameras, RGB-D cameras, lidars) and is decoupled from the odometry source.*

In our decentralized (see Property 3) back-end, the resulting intra-robot and inter-robot loop closure measurements are combined with the odometry measurements into a pose graph. Local pose graphs are transmitted to the robot selected, through neighbor management negotiation, to perform the optimization and the resulting estimates are sent back to the respective robots.

Property 3. ***Decentralized****. All computation is performed onboard the robots without any central authority and they rely only on peer-to-peer communication.*

Current pose estimates, resulting from the whole process, are made available periodically in the form of ROS 2 messages for a minimally invasive integration into existing robotic systems. To avoid needless bandwidth use, the neighbor manager keeps track of which measurements have been exchanged. Mapping data for planning or visualization can also be queried at



(a) 2 robots KITTI-00

(b) 3 robots S3E Laboratory

Figure 6.2 Visualization of 10 first loop closure candidates selected by the Spectral and Greedy Approaches. The Spectral approach selects candidates in different regions of the pose graph to increase the accuracy, while the Greedy approach selects redundant candidates in high similarity regions.

the cost of additional computation and communication. For debugging purposes, we provide a minimal visualization tool which opportunistically collects mapping data from robots in communication range. Overall, we divided place recognition, geometric verification and pose graph optimization into modular and decoupled processes with clear data interfaces to enable researchers to leverage Swarm-SLAM to easily test new ideas in each subsystems.

6.5 Front-End

Similar to many comparable inter-robot loop closure detection techniques (e.g. [37, 62, 63]), we adopt a two stage approach in which global matching generate candidate place recognition matches that are verified using local features in the latter stage, i.e. local matching.

6.5.1 Global Matching

For each keyframe, compact descriptors, that can be compared with a similarity score, are extracted from sensor data and broadcast to neighboring robots. When two robots meet, we perform bookkeeping to determine which global descriptors are already known by the other robot and which ones need to be transmitted. We use ScanContext [274] as global descriptors of lidar scans and the recent CNN-based CosPlace [273] for images. We use nearest neighbors based on cosine similarity for descriptor matching. Once matches are computed, Swarm-SLAM offers two candidate prioritization mechanisms: a **greedy** prioritization algorithm, used in prior work [37, 62, 63, 275], and a novel **spectral** approach. To perfom the candidate prioritization, we define the multi-robot pose graph as:

$$\mathcal{G} = (V, \mathcal{E}^{\text{local}}, \mathcal{E}^{\text{global}}) \tag{6.1}$$

$$V = (V_1, \ldots, V_n) \tag{6.2}$$

$$\mathcal{E}^{\text{local}} = (\mathcal{E}_1^{\text{local}}, \ldots, \mathcal{E}_n^{\text{local}}) \tag{6.3}$$

$$\mathcal{E}^{\text{global}} = (\mathcal{E}_{\text{fixed}}^{\text{global}}, \mathcal{E}_{\text{candidate}}^{\text{global}}) \tag{6.4}$$

where V are the vertices from every n robots pose graphs, each vertex corresponding to a keyframe; $\mathcal{E}^{\text{local}}$ are the local pose graphs edges such as odometry measurements and intra-robot loop closures; and $\mathcal{E}^{\text{global}}$ are the global pose graph edges corresponding to inter-robot loop closures. $\mathcal{E}^{\text{global}}$ is further divided between $\mathcal{E}_{\text{fixed}}^{\text{global}}$ which contains the fixed measurements that have already been computed, and the candidate inter-robot loop closures $\mathcal{E}_{\text{candidate}}^{\text{global}}$ on which the prioritization is performed. Detailed measurements (i.e., pose estimates) are not required for our proposed candidate prioritization mechanism. Therefore, fixed measurements, both local and global, are undirected unweighted edges between two vertices, and candidates edges contain an additional weight value corresponding to their respective similarity score. This reduced multi-robot pose graph can be built directly from the global matching information and does not require any additional inter-robot communication.

The number of edges B to select at each time step is set by the user. This budget should reflect the communication and computation capacities of the robots. The common candidate prioritization approach widely used prior works is a basic *greedy prioritization* in which the top B candidates with the highest similarity scores are selected.

In our proposed *spectral prioritization* process, we frame pose graph sparsification as a candidate prioritization problem, and leverage recent work on spectral sparsification. We observe that the two problems are mathematically equivalent, one being solved before loop closure computation and the other after. Specifically, we perform sparsification on the candidate inter-robot matches before computing the corresponding 3D measurements, reducing resource usage for the costly inter-robot geometric verification of redundant candidates, and achieving better accuracy (see Property 4).

Property 4. ***Sparse***. *The framework prioritizes communication using algebraic connectivity maximization sparsification to acheive better localization accuracy with fewer data exchanges.*

At every stage, it requires less communication than comparable techniques.

As shown in [278], the algebraic connectivity of the pose graph controls the worst-case error of the solutions of the SLAM *Maximum Likelihood Estimation* problem. The pose graph algebraic connectivity corresponds to the second-smallest eigenvalue λ_2 of the *rotation weighted Laplacian* with entries for pairs of vertices (i, j) defined as:

$$L_{ij} = \begin{cases} \sum_{(i,j')\in\delta(i)} \kappa_{ij'}, & i = j, \\ -\kappa_{ij}, & \{i,j\} \in \mathcal{E}, \\ 0, & \{i,j\} \notin \mathcal{E}. \end{cases} \tag{6.5}$$

where κ_{ij} denotes the edge weight and $\delta(i)$ is the set of edges incident to vertex i. Instead of using a noise model for the edge weights as in [272], we use the similarity score $s_e \in [0, 1]$ from global matching as confidence metric. Thus, we define $\kappa_{ij} = 1 \; \forall \; e \in (\mathcal{E}^{\text{local}}, \mathcal{E}^{\text{global}}_{\text{fixed}})$, and $\kappa_{ij} = s_e \; \forall \; e \in \mathcal{E}^{\text{global}}_{\text{candidate}}$. This approach forgoes the need to communicate additional information regarding the edges' estimated noise level. However, it is important to note that it loses the theoretical guarantees from [272], yet we show in Section 8.5 that this heuristic approach works well in realistic cases.

For our purposes, we leverage the property that the Laplacian L can be expressed as the sum of subgraph Laplacians corresponding to each of its edges to define the augmented pose graph Laplacian as follows:

$$L(\omega) \triangleq L^{\mathcal{E}^{\text{local}}} + L^{\mathcal{E}^{\text{global}}_{\text{fixed}}} + \sum_{e\in\mathcal{E}^{\text{global}}_{\text{candidate}}} \omega_e L_e \tag{6.6}$$

where $\omega_e \in \{0, 1\}$ is the binary variable which determines the prioritization of candidate edge e. Therefore, according to our previously stated goal, we aim to select the subset $\mathcal{E}^\star \subseteq \mathcal{E}^{\text{global}}_{\text{candidate}}$ of fixed budgeted size $|\mathcal{E}^\star| = B$ which maximizes the algebraic connectivity $\lambda_2(L(\omega))$:

Problem 1. *Candidate prioritization via Algebraic Connectivity Maximization*

$$\begin{gathered} \max_{\omega_e\in\{0,1\}} \lambda_2(L(\omega)) \\ |\omega| = B. \end{gathered} \tag{6.7}$$

Problem 1 is NP-Hard [281] due to the integrality constraint on ω_e. Therefore, we relax the integrality constraints and, when necessary, we round the optimization result to the nearest solution in the feasible set of Problem 1. We solve the relaxed problem using the simple and computationally inexpensive approach developed in [272]. It is important to note that this approach requires the pose graph to be connected, so we first perform greedy prioritization up until at least one inter-robot loop closure exists between the local pose graphs. We also use the greedy solution as initial guess for the algebraic connectivity maximization.

In Fig. 6.2, we present a visualization of our spectral approach results in comparison to the ones obtained with the standard greedy approach. We can see that the candidates selected using our spectral technique are more evenly distributed along the pose graph while the greedy candidates are mostly concentrated in high-similarity areas. Our selected candidates are therefore less redundant for the estimation process.

6.5.2 Local Matching

Once the inter-robot loop closure candidates are selected, the next step is to perform local matching (i.e., geometric verification). This step leverages larger collections of local features, keypoints or point clouds depending on the sensor, to compute the 3D relative pose measurement between the candidate's two vertices. To avoid computing the same loop closure twice and to reduce the communication burden of geometric verification, we follow [153] and formulate the vertices local features sharing problem as a vertex cover problem. When two or more inter-robot loop closure candidates share a vertex in common, only the common vertex needs to be transmitted to effectively compute all the associated relative pose measurements. Thus, by computing the minimal vertex cover, optimally for bipartite graphs and approximately with 3 robots or more, we obtain an exchange policy which avoids redundant communication.

6.5.3 Inter-Robot Communication

It is worth noting, that both for the spectral matching and the vertex exchange policy, a temporary *broker* needs to be dynamically elected among the robots in communication range. The broker then computes the matches and sends requests for the vertices to be transferred. In our current implementation, the broker is simply the robot in range with the lowest ID according to our neighbor management system, but it could be elected with a different decentralized mechanism (e.g. based on the available computation resources onboard each robot).

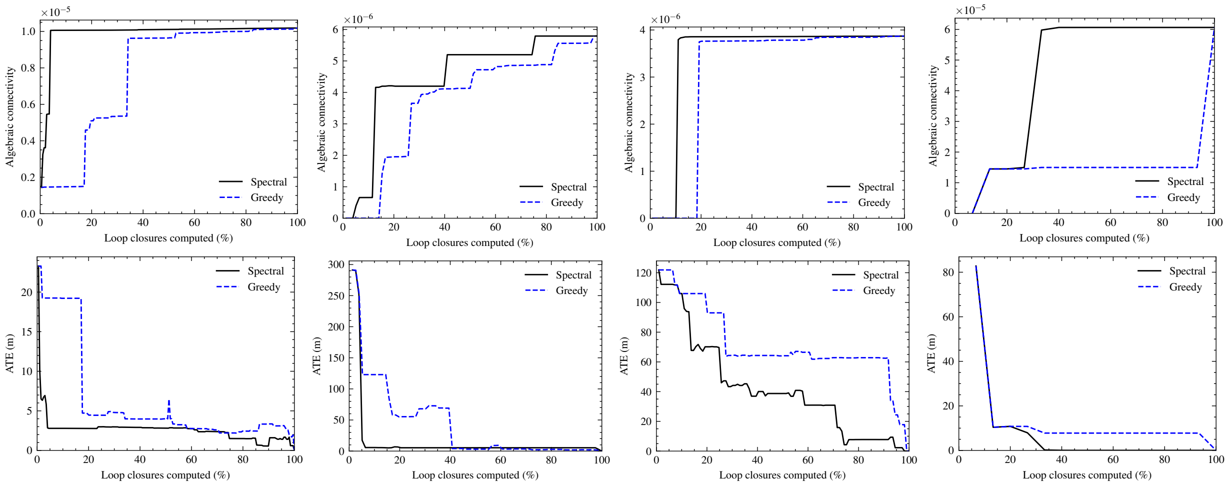
6.6 Back-End

The role of the back-end is to gather the odometry, intra- and inter-robot loop closure measurements from the front-end in a pose graph, and then estimate the most likely map and poses based on those noisy measurements. As mentioned above, unlike other recent systems [62,156] based on distributed pose graph optimization, we opt for a simpler decentralized approach. Similar to the front-end, a robot is dynamically elected to perform the computation among the robots in communication range. The other robots share their current pose graph estimates with the elected robot and receive the updated estimates once the computation is completed. Importantly, any robot can be temporarily elected through negotiation to perform the pose graph optimization during a rendezvous between robots. Swarm-SLAM performs the pose graph optimization using the Graduated Non-Convexity [140] solver, with the robust Truncated Least Square loss.

To ensure convergence to a single global localization estimate after multiple sporadic rendezvous without enforcing a central authority, we introduce an anchor selection process to keep track of the current global reference frame. During pose graph optimization, the anchor usually corresponds to a prior which assigns a fixed value to the first pose of the graph. This anchor then becomes the reference frame of the resulting estimate. In the beginning, all robots are within their own local reference frames where the origin corresponds to their first pose (i.e., initial position and orientation). Then, when some robots meet for the first time (e.g. robots 0, 4 and 5), we choose the first pose of the robot with the lowest ID (e.g. robot 0) as the anchor. Therefore, as a result of the estimation process, the involved robots estimates share the same reference frame (e.g. robot 0's first pose). In subsequent rendezvous (e.g. robots 2, 3 and 4), the anchor is selected based on the reference frame with the lowest ID (e.g. robot 4's first pose is selected as the anchor since its reference frame is robot 0's). After a few rendezvous, the robots converge to a single global reference frame without requiring rendezvous including all robots (e.g. after the second rendezvous, robots 2 and 3 are also within robot 0's reference frame). This means that Swarm-SLAM can scale to large groups of robots, through iterative estimation among smaller groups of robots.

6.7 Experimental Results

To evaluate the effectiveness of our proposed solutions for the ongoing challenges in Collaborative Simultaneous Localization and Mapping, we conducted extensive experiments on several public datasets, as well as in a real-world deployment. Our experiments involved three robots exploring and mapping an indoor environment and communicating via ad-hoc



(a) KITTI 00 (b) KITTI-360 09 (c) GrAco Ground (d) S3E Square

Figure 6.3 Comparison between *Greedy* and *Spectral* prioritization of candidate inter-robot loop closures. For each datasets, we show that the *spectral* approach outperforms the *greedy* one in terms of algebraic connectivity (higher is better), and the *Absolute Translation Error* (ATE) (lower is better). We can see that the *spectral* prioritization converges to the best estimate with less inter-robot loop closures.

networking. We specifically evaluated our key contributions to inter-robot loop closure detection and decentralized C-SLAM estimation. Additionally, we present detailed statistics of the communication and computation load during our real-world experiment, providing insight into the system's performance and resource requirements.

6.7.1 Dataset Experiments

We tested Swarm-SLAM on seven sequences from five different datasets. To demonstrate the flexibility of our framework, we used IMUs, stereo cameras, lidars, or a combination as inputs. First, we tested on the widely known autonomous driving KITTI 00 stereo sequence [176] which we split into two parts to simulate a two-robots exploration. Second, we split the very large (∼10km) KITTI360 09 lidar sequence [266] into 5 parts that contain a large number of loop closures, making it particularly well suited for inter-robot loop closure detection analysis. Third, we experimented on the first three overlapping lidar sequences of the very recent GrAco dataset [233] acquired with custom ground robots on a college campus. Fourth, we evaluate our system on the three lidar Gate sequences of the M2DGR dataset [282]. Fifth, we tested on three sequences of the recent C-SLAM-focused S3E dataset [232]. To avoid tracking failures and obtain more robust results on S3E sequences, we combined lidar-IMU odometry

Table 6.2 C-SLAM Estimates Evaluation on Public Datasets with Different Back-Ends.

		Communication (kB)			Time (s)			ATE (m)		
	Robots	GNC	DGS+PCM	D-GNC	GNC	DGS+PCM	D-GNC	GNC	DGS+PCM	D-GNC
KITTI 00	2	**280.00**	30045.50	13499.49	**20.11**	230.85	70.903	**2.17**	9.08	3.77
KITTI-360 09	5	**484.91**	3241.21	4485.13	196.93	102.51	**70.82**	**4.02**	6.67	7.15
GrAco Ground	3	**105.82**	44686.69	78162.42	**8.06**	120.25	143.79	**6.19**	33.73	8.47
M2DGR Gate	3	**51.29**	242.02	721.39	**1.42**	5.02	7.85	**0.70**	1.35	3.07
S3E Square	3	**80.97**	1261.12	620.19	**6.05**	50.31	26.98	**4.20**	9.74	20.21
S3E College	3	**150.37**	3420.23	3012.28	**11.96**	125.42	35.62	**3.57**	25.32	4.13

and stereo camera-based inter-robot loop closure detection, highlighting the versatility of Swarm-SLAM. Overall, we chose the sequences with the most trajectory overlaps to obtain more loop closures, and with available GPS ground truth (except for S3E Laboratory). For simplicity and robustness, we used off-the-shelf software [269] to compute and provide the required odometry input to Swarm-SLAM. To better evaluate the inter-robot loop closure detection, we consider the worst-case scenario in which the robots are within communication range only at the end of their trajectories, such that they have to find loop closures between their whole maps at once. This scenario, analog to multiple robots exploring different parts of an environment and meeting back at the end, is among the most challenging in terms of communication and computation load, and therefore benefits the most from our novel spectral candidate prioritization mechanism. We refer the reader to our open-source implementation for all the parameters and configuration details of the experiments.

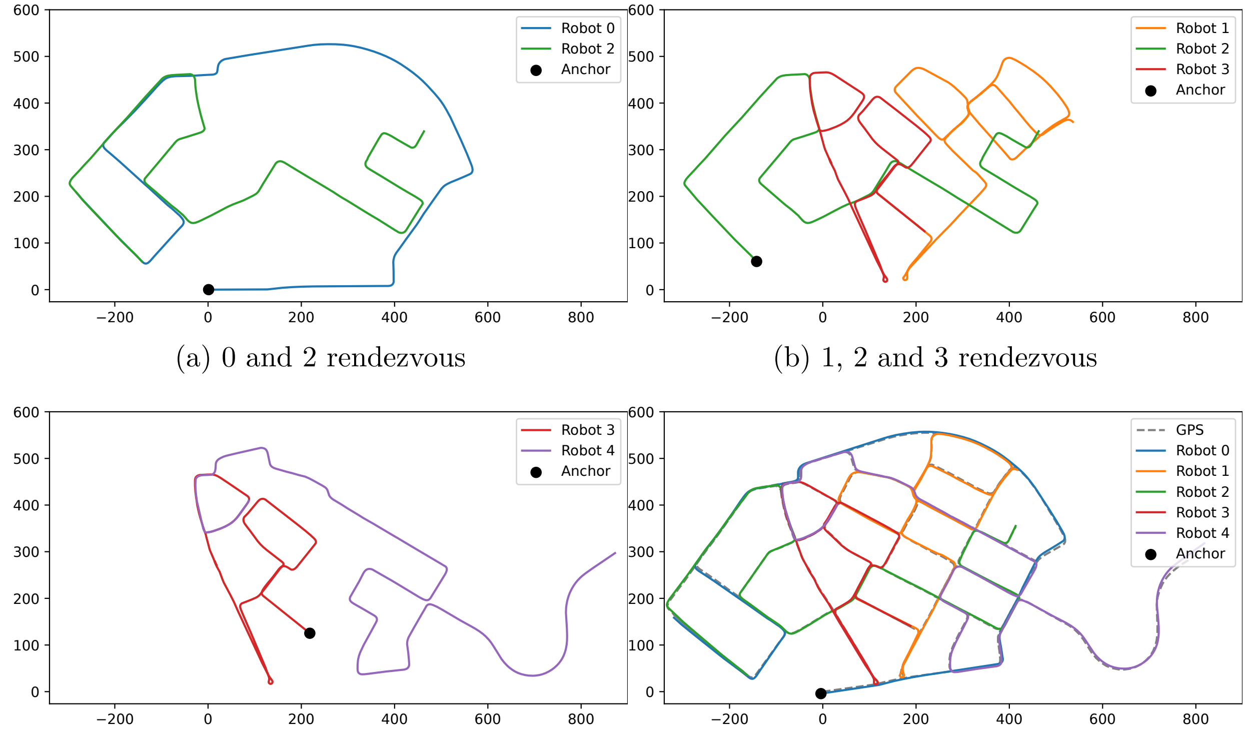
Inter-Robot Loop Closure Detection Evaluation

In Fig. 6.3, we compare the greedy and spectral inter-robot loop closure candidate prioritization techniques with respect to algebraic connectivity, and Absolute Translation Error (ATE). Each approach is used to prioritize the computation of loop closures from the same set of candidates with a budget B of 1, i.e. selecting one loop closure at a time. We plot each metric against the percentage of loop closures computed within the set of candidate (x-axis). We perform prioritization successively up until all the possible matches are selected (i.e., 100% of loop closures computed). We expect that a better prioritization will acheive reasonable accuracy early on, with only a fraction of the matches selected. The ATE is computed against the final pose graph estimate containing all possible inter-robot loop closures, and thus constitutes the best estimate we can achieve. On the first row, we can see that, as intended, our spectral prioritization is correctly maximizing the algebraic connectivity of the pose graph. On the second row, as expected, we can see that our spectral prioritization decreases the error faster than the greedy prioritization. Overall, our experiments show that carefully selecting candidates early on requires the computation of fewer inter-robot loop closures to significantly reduce the estimation error (ATE).

Decentralized C-SLAM Evaluation

In Table 6.2, we present the estimates computed in the back-end on all the sequences for which GPS latitude and longitude data is available as ground truth. We report the computation time on a AMD Ryzen 7 CPU and the total communication required in kB. Using our same front-end, we compared our GNC-based decentralized back-end against two state-of-the-art distributed approaches: the Distributed Gauss-Seidel (DGS) pose graph optimization [53] combined with Pairwise Consistency Maximization (PCM) [143] for outlier rejection as used in [62]; and a distributed implementation of Graduated Non-Convexity (D-GNC) [63] based on the RCBD solver [126]. Our chosen back-end consistently achieved the highest level of accuracy (ATE) whereas alternative methods occasionally fell short of generating reasonable estimates. We also consistently outperforms the other approaches in terms of required communication and computation time. Interestingly, when tested on KITTI-360 09, our dataset with the highest number of robots, both DGS+PCM and D-GNC take less computation time compared to GNC, yet they don't achieve equivalent accuracy and necessitate more than five times the amount of data transmission. While distributed approaches benefit from the division of labour on large problems, more research is required to obtain the same levels of accuracy, robustness, and communication bandwidth. This justifies our practical choice of a simpler approach computing the back-end on a single decentrally-elected robot, which is more robust to communication failures and easier to implement.

In Fig. 6.4, we show the Swarm-SLAM resulting estimates on the KITTI360 09 sequence from four different rendezvous, defined as an encounter in which a subset of robots are within communication range of each other. Our anchor selection scheme ensures that by choosing the current first pose estimate from the robot with the lowest reference frame ID (i.e., first poses of (a) robot 0, (b) robot 2, (c) robot 3), we can propagate the global reference frame among the team of robots. In other words, we are able to converge to a single global reference frame through successive estimations between subsets of robots, without enforcing connectivity maintenance or a central authority. This decentralized approach improves the scalability of the system by relying only on local interactions among neighboring robots. We present the Swarm-SLAM solutions on the remaining dataset sequences in Fig. 6.5.



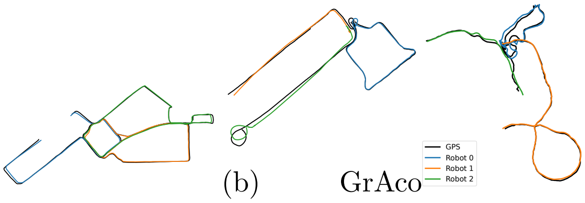
(a) 0 and 2 rendezvous

(b) 1, 2 and 3 rendezvous

(c) 3 and 4 rendezvous

(d) All robots rendezvous

Figure 6.4 Reference frame convergence via occasional rendezvous. From (a) to (c), we show trajectory estimates from successive rendezvous between different groups of robots: {0,2}, then {1,2,3}, and finally {3,4}. After the three rendezvous, all estimates are within the same global reference frame. For comparison, we also include the result of a rendezvous with all robots along side the GPS estimate (d).



(a) S3E College

(b) GrAco Ground

(c) M2DGR

Figure 6.5 Swarm-SLAM trajectory estimates on various dataset sequences compared with GPS ground truth.

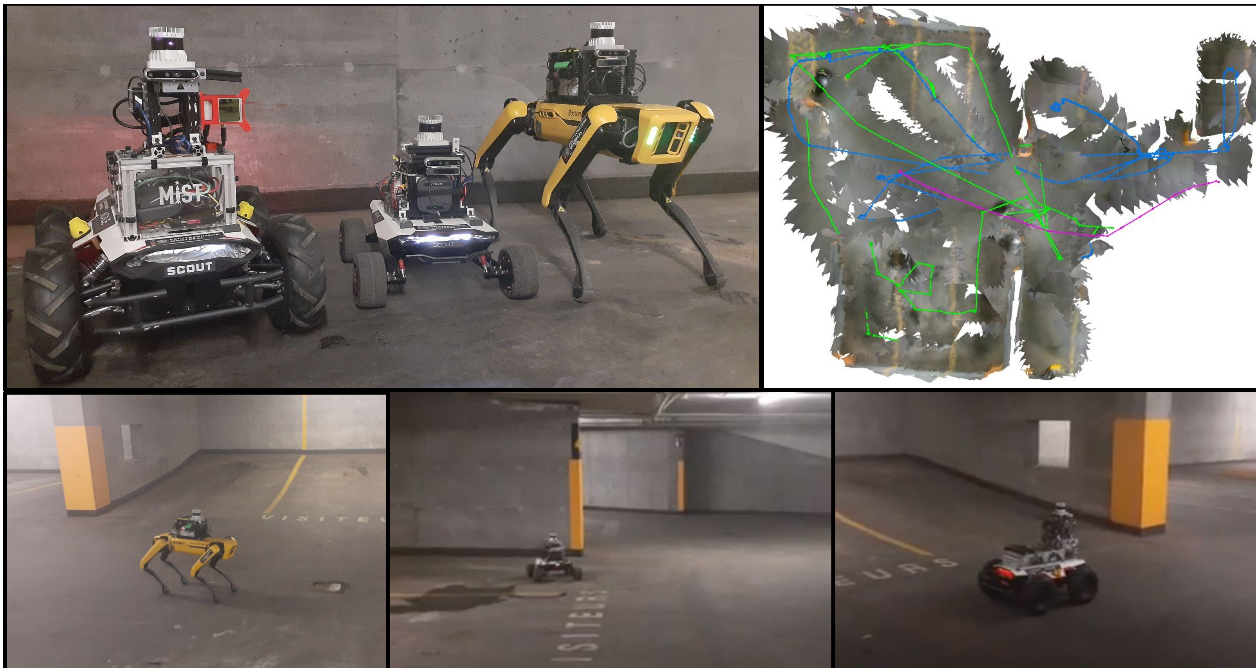


Figure 6.6 Swarm-SLAM experiment with 3 robots, equipped with lidars and RGB-D cameras, simultaneously exploring an indoor parking lot, and achieving shared situational awareness via collaborative perception. A visualization of the resulting map and pose graphs is showed in the top right corner.

6.7.2 Real-World Experiments

To assess the viability of Swarm-SLAM on resource-constrained platforms, we deployed the system in an indoor parking lot and gathered statistics regarding the computation time and communication load. As shown in Fig. 6.6, we performed an online real-world demonstration with 3 different robots (Boston Dynamics Spot, Agilex Scout, and Agilex Scout Mini), all equipped with an NVIDIA Jetson AGX Xavier onboard computer, an Intel Realsense D455 camera, an Ouster lidar OS0-64, a VectorNav VN100 IMU, and a GL-iNet GL-S1300 OpenWrt gateway for ad hoc networking. We used lidars and IMUs for odometry and the RGB-D cameras for inter-robot loop closure detection.

Table 6.3 Real-World Experiment Statistics

# Robots	3	Length (m)	475.42
# Keyframes	3103	Total comm. (MB)	94.95
# Inter-robot loop cl.	67	# Outliers	10
Optimization time (s)	5.52 ± 7.11	Sparsification time (s)	2.71 ± 2.39

As stated in Table 6.3, our robots travelled a total of 475 meters during the experiment and produced a total of 3103 keyframes that needed to be matches and verified in the search for inter-robot loop closures. The process resulted in 67 loop closures, including 10 that were rejected by the GNC optimizer. This large number of outliers is attributable to the many similar-looking sections of the parking lot. Swarm-SLAM acheived accurate localization with the transmission of only 94.95 MB of data between the robots, excluding the visualization. The communication load is mostly attributable to the front-end and thus dependent on the number of keyframes. In Table 6.3, we also report the average sparsification and pose graph optimization times. We can observe that the sparsification time, while being non-negligible, is lower than the pose graph optimization. To mitigate this, we implemented sparsification and optimization within separate threads.

6.8 Conclusions And Future Work

In this paper, we presented Swarm-SLAM, a comprehensive resource-efficient C-SLAM framework that is designed to comply with essential properties of swarm robotics. In future work, we aim to investigate collaborative domain calibration and/or uncertainty estimation in place recognition [283] to reduce the prevalence of measurement outliers among inter-robot loop closures, and therefore increase the overall accuracy and resilience of C-SLAM. Overall, we hope that our open-source framework will be useful as a testbed for the research and development of new methods and techniques in place recognition, inter-robot loop closure detection, multi-robot pose graph optimization, and other open-problems.

CHAPTER 7 ARTICLE 4 : MULTI-ROBOT DECENTRALIZED COLLABORATIVE SLAM IN PLANETARY ANALOGUE ENVIRONMENTS: DATASET, CHALLENGES, AND LESSONS LEARNED

Preface: This paper presents insights from C-SLAM experiments with three robots on Mars analogue terrain, addressing challenges like intermittent communication and perceptual aliasing. It also introduces a novel dataset with real-time inter-robot communication metrics to support future research on decentralized, communication-constrained multi-robot missions. This work was submitted for review to IEEE Transactions on Field Robotics.

Contributions: This project is the result of a significant collaborative effort by various members of the MIST laboratory. The hardware and networking setup were originally developed for other field experiments and demonstrations also conducted in the summer of 2024. These particular hardware and networking efforts were led by Vivek Shankar Varadharajan and Giovanni Beltrame, with support from various lab members, including myself. Building on this collective work, I conceptualized this article along with the data collection and analysis it presents. Karthik Soma and Haechan Mark Bong were instrumental in the GPS setup and sensor monitoring, Rongge Zhang assisted with software setup and calibration, and Alice Lemieux-Bourque supported logistics before and during the experiments. I led the experiments, managed the data release, and wrote the majority of the paper.

Full Citation: Pierre-Yves Lajoie, Karthik Soma, Haechan Mark Bong, Alice Lemieux-Bourque, Rongge Zhang, Vivek Shankar Varadharajan, Giovanni Beltrame, "Multi-Robot Decentralized Collaborative SLAM in Planetary Analogue Environments: Dataset, Challenges, and Lessons Learned," *Submitted to IEEE Transactions on Field Robotics*, October 2024.

Submission date: October 29th 2024

7.1 Abstract

Decentralized Collaborative Simultaneous Localization and Mapping (C-SLAM) is essential to enable multi-robot missions in unknown environments without relying on pre-existing localization and communication infrastructure. This technology is anticipated to play a key role in the exploration of the Moon, Mars, and other planets. In this paper, we share insights and lessons learned from C-SLAM experiments involving three robots operating on a Mars analogue terrain and communicating over an ad-hoc network. We examine the impact of limited and intermittent communication on C-SLAM performance, as well as the unique

localization challenges posed by planetary-like environments. Additionally, we introduce a novel dataset collected during our experiments, which includes real-time peer-to-peer inter-robot throughput and latency measurements. This dataset aims to support future research on communication-constrained, decentralized multi-robot operations.

7.2 Introduction

Multi-robot systems hold the potential to revolutionize space and planetary exploration. Teams of robots can parallelize work, be more resilient to individual failures, and, most importantly, collaborate to accomplish collective tasks that are out of reach of single-robot systems. However, operating robots on another planet presents some unique challenges, such as large communication delays with base stations on Earth, difficult and unknown terrain, or the absence of any pre-existing infrastructure. In these conditions, robots need high levels of autonomy to operate safely.

One of the key enablers of robot autonomy is the Simultaneous Localization and Mapping (SLAM) algorithm [1], which provides localization estimates of the robot and a map of the surrounding environment that can be used for terrain analysis, path planning, and decision making. In the case of multi-robot systems, the robots need to collaborate in the localization and mapping process in order to converge to a consistent perception of the environment across the team of robots. Without shared situational awareness, individual robots are constrained by their limited view of the environment and are not able to collaborate efficiently. Thus, Collaborative SLAM (C-SLAM) [24, 252] is likely to be a vital component of future multi-robot missions on the Moon, Mars, or other planets.

That being said, there are additional key requirements for the efficient deployment of C-SLAM algorithms in space. First and foremost, due to the high cost of space missions, the systems need to be as robust and resilient to failures as much as possible. Also, due to the lack of pre-existing localization and networking infrastructure, individual robots need enough autonomy from any base station (on Earth or on the explored planet itself) to survive frequent and lengthy disconnections over the course of their missions. Given those constraints, typical centralized approaches to C-SLAM [39, 44], in which robots send measurements to a central computing node that computes and shares back the merged global map and localization estimates, are highly vulnerable to network disconnections or the outright failure of the central computer.

To mitigate these risks, decentralized techniques that are purposely built to work with ad-hoc networking and withstand prolonged disconnections between robots are preferable. In

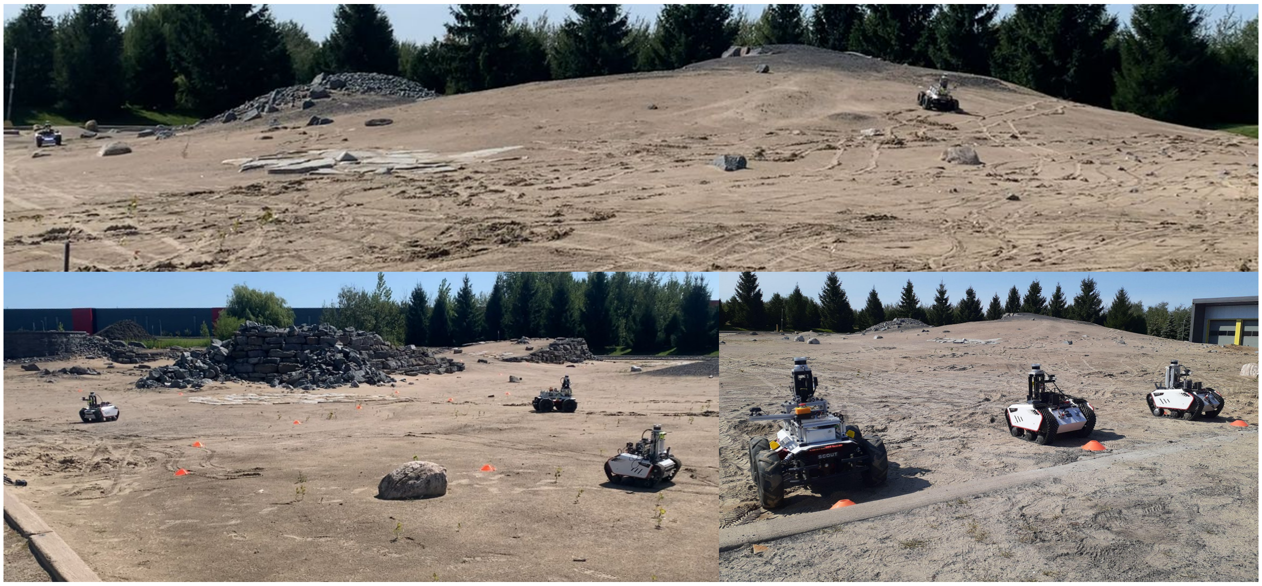


Figure 7.1 Multi-robot experiments at the Canadian Space Agency Mars Yard in Saint-Hubert, Québec. In our experiments, three robots simultaneously explored the simulated Martian terrain. The robots collaborated through peer-to-peer communication to map the environment and localize themselves within the landscape, demonstrating the effectiveness of our C-SLAM system in a challenging, planetary analogue setting.

prior work, we introduced Swarm-SLAM [284], a decentralized C-SLAM framework satisfying those requirements. Building on this framework, we conducted a series of field experiments, and collected a novel dataset, at the Canadian Space Agency Mars Yard [285], a planetary analogue terrain designed to simulate realistic planetary conditions. We use Swarm-SLAM as a case study to evaluate the current performance of state-of-the-art decentralized C-SLAM. We look in particular at the challenges related to limited and intermittent inter-robot communication, as well as the difficulties posed by the terrain in terms of vibrations, lack of distinctive features, and perceptual aliasing.

7.2.1 Contributions

This paper is the culmination of extensive multi-robot experiments on the planetary analogue environments shown in Fig. 7.1. As a result, we present the following contributions:

- The design of a decentralized three-robot system connected through ad-hoc networking, and its deployment on a planetary analogue environment;
- A novel dataset collected during our experiments that includes LiDAR, IMU, and, unlike prior works, peer-to-peer inter-robot communication throughput and latency

estimates that are valuable for evaluating the communication consumption of C-SLAM or other multi-robot algorithms. Our dataset is available here: `https://github.com/MISTLab/Mars_Analogue_CSLAM_Dataset`;

- A thorough analysis of the accuracy and efficiency of decentralized C-SLAM, exposing limitations of current approaches and open challenges.

We believe that the challenges and lessons learned from our experiments will be valuable for both the space robotics and C-SLAM research communities.

The remainder of this paper is divided as follows: Section 8.3 presents background knowledge and related work on decentralized C-SLAM, inter-robot networking, and localization challenges in space analogue environments; Section 7.4 describes our experimental setup comprising the robots and sensors used and key characteristics of the terrain on which they were deployed; Section 7.5 presents the accuracy results of our Swarm-SLAM decentralized C-SLAM algorithm and localization challenges inherent to planetary analogue terrains; Section 7.6 discusses the calibration trade-offs between accuracy and resource efficiency in terms of communication and computing; finally, Section 8.6 offer some insights gained during our experiments and open challenges that we identify for the future of C-SLAM for space robotics.

7.3 Background and Related Work

7.3.1 Decentralized Collaborative Simultaneous Localization and Mapping

Centralized vs Decentralized

C-SLAM systems are typically categorized into centralized or decentralized solutions. Centralized C-SLAM relies on a main server or base station that gathers mapping data from all participating robots and computes a common global localization and mapping estimate for the whole team. This setup requires robots to maintain a stable, continuous connection to the base station, leading to significant challenges in scalability due to potential communication bottlenecks. It is also vulnerable to failures in the central computing node [181]. Such stringent connectivity and reliability requirements can be impractical, particularly in planetary environments without pre-existing networking architectures.

In contrast, decentralized C-SLAM systems operate without a centralized server, instead utilizing occasional, peer-to-peer communication among robots. This approach is advantageous in environments where constant connectivity is unreliable or impossible. However, decentralized systems face their own set of challenges, as they are limited by the robots' individual computing and communication resources. They also demand more complex strategies for data handling and coordination to ensure accurate and consistent SLAM results across the team of robots [252].

Both centralized and decentralized C-SLAM systems share a similar architecture with single-robot SLAM systems, consisting of two key components: the front-end and the back-end. The front-end is responsible for tasks such as feature extraction and data association, which involve identifying and matching environmental features to aid in robot relocalization. The back-end, on the other hand, is dedicated to state estimation, determining the robots' poses (i.e., their positions and orientations) relative to the constructed map of the environment [1]. In collaborative state estimation, the maps and poses of the robots are integrated into a common frame of reference, allowing for consistent and unified positioning across all robots [24].

Front-End

A major challenge in the C-SLAM front-end is efficiently detecting and computing inter-robot loop closures. These loop closures occur when two or more robots identify common landmarks or locations they have previously explored. Such shared features act as connection points that allow the integration of local maps from individual robots into a unified global reference frame. Because transmitting entire maps can be prohibitively expensive in terms of communication costs, inter-robot loop closure detection is typically handled in a two-step process [37, 62].

In the first step, robots exchange compact descriptors, which are simplified representations of their data, such as image descriptors (e.g., CosPlace [273]) or LiDAR scan descriptors (e.g., ScanContext [274]). These descriptors enable place recognition by calculating similarity scores to identify overlaps in the environments mapped by different robots. High similarity scores suggest potential loop closure candidates, indicating that the robots might have traversed the same place.

The second step focuses on these identified candidates. For each candidates with high similarity, more detailed and computationally expensive descriptors, like 3D keypoints or full scans, are exchanged. These are used to perform precise geometric registration between the corresponding data from different robots, establishing accurate positional and rotational links between them. The resulting pose measurements, called loop closures, are then integrated into the robots' pose graphs, to merge the maps and enhance the state estimation accuracy.

In the case of LiDAR scans, since point clouds often include noise and outliers, robust methods like TEASER++ [286] are employed to ensure accurate registration without needing an initial pose guess. This capability is especially valuable in C-SLAM, as the robots generally do not know their relative positions or trajectories prior to the first map merging.

Back-End

The C-SLAM back-end is tasked with estimating the most likely robot poses and maps from the noisy data collected by the robots. To reduce the computational cost of large-scale SLAM problems, which is especially challenging for multi-robot mapping, most approaches utilize some form of pose graph optimization. This method marginalizes environmental features into inter-pose measurements, solving the optimization problem by focusing only on the poses. While single-robot solvers can be directly applied to multi-robot scenarios, as demonstrated in our approach [284], several distributed multi-robot solvers have been developed [53, 126, 230]. These distributed approaches offer the advantage of better computational scalability as they distribute the workload among connected robots. However, they require substantial bookkeeping and data exchange during optimization, and the associated network delays can negate the computational benefits. To address these issues, our method opts to perform optimization on a single dynamically elected robot during each robot rendezvous.

A common challenge in SLAM systems, including C-SLAM, is the prevalence of erroneous measurements due to perceptual aliasing [5]. Perceptual aliasing occurs when different locations in the environment appear too similar and lack distinctive features, leading to incorrect data associations during place recognition. To mitigate this issue, Mangelson et al. [143] introduced Pairwise Consistency Maximization (PCM), which enhances robustness by identifying the largest set of consistent inter-robot measurements. In another line of work, Yang et al. [140] proposed the Graduated Non-Convexity (GNC) algorithm, a flexible and robust tool for various optimization tasks. GNC is employed by the elected robots in Swarm-SLAM for pose graph optimization. Tian et al. [63] have extended GNC for use in distributed implementations.

Recently, numerous C-SLAM frameworks have emerged, each contributing to advancements in distributed mapping. DSLAM [37] was a pioneering system that used CNN-based image descriptors for place recognition and distributed pose graph optimization. DOOR-SLAM [62] further developed these ideas by integrating PCM for outlier rejection and adapting the system to handle intermittent inter-robot communication, eliminating the need for full connectivity between robots. DiSCo-SLAM [275] expanded these concepts to LiDAR-based mapping, using ScanContext descriptors [274] for place recognition. Kimera-Multi [63] combines classical approaches to place recognition and registration with a distributed version of GNC.

Centralized C-SLAM systems have also advanced considerably. COVINS [61] optimizes visual-inertial SLAM by reducing computational load through the elimination of redundant keyframes. LAMP 2.0 [276] employs a Graph Neural Network-based prioritization mechanism to evaluate inter-robot loop closure candidates, predicting the optimization outcomes and selecting the most promising candidates for further processing. Maplab 2.0 [220] is designed to support varying sensor modalities and configurations, enabling flexible multi-robot mapping.

Swarm-SLAM [284], our recently proposed system described in Fig. 7.2, builds on these advancements by introducing a novel sparse inter-robot loop closure prioritization technique to reduce communication overhead. It uses ROS 2 and includes a neighbor management system that integrates smoothly with ad-hoc networking, enhancing C-SLAM's adaptability to intermittent communication scenarios. For a comprehensive review of C-SLAM technologies, we refer the readers to [252].

7.3.2 Ad-hoc Inter-Robot Communication

Ad hoc networks play a crucial role in enabling multi-robot mapping, allowing robots to communicate directly with one another without relying on a pre-existing infrastructure. In the early exploration of ad-hoc inter-robot communication for collaborative mapping, Sheng et al. [287] proposed a 2D grid-based approach that minimizes data exchange by leveraging known relative poses between robots. More recently, Varadharajan et al. [288] addressed the broader challenge of efficiently sharing large volumes of data, such as maps, within distributed robot networks by introducing a peer-to-peer data sharing system specifically designed for high data loads. To ensure reliable inter-robot communication, it is crucial to consider the robot topology, as it directly influences the available communication paths between connected agents [289, 290]. For example, Varadharajan et al. [291] proposed a fully decentralized connectivity algorithm robust against individual robot failures. This approach allows robots to autonomously adjust their positions to maintain network connectivity with a ground station, ensuring stable communication despite dynamic conditions and potential disconnections.

While some C-SLAM systems, like those described in [37,61] require fully connected networks, recent approaches have been designed to be resilient to disconnections. For instance, our prior work [62] and Tian et al. [292] propose C-SLAM frameworks that can withstand intermittent communication losses. More recently [284], we successfully deployed C-SLAM with ad-hoc networking in real-world settings, demonstrating the practical feasibility of decentralized multi-robot mapping in challenging environments without the need for complex continuous connectivity maintenance.

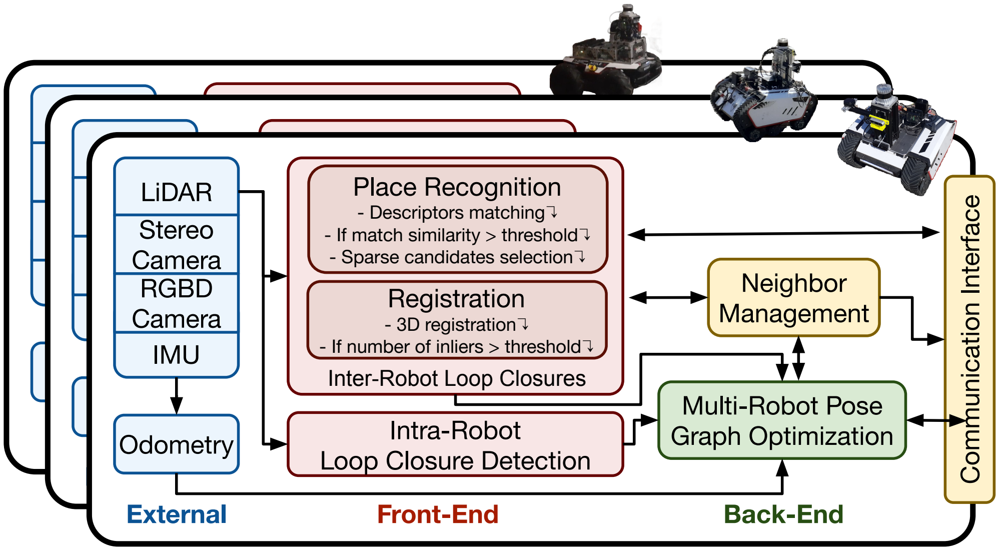

LiDAR
Stereo Camera
RGBD Camera
IMU
Odometry
Place Recognition
- Descriptors matching
- If match similarity > threshold
- Sparse candidates selection
Registration
- 3D registration
- If number of inliers > threshold
Inter-Robot Loop Closures
Intra-Robot Loop Closure Detection
Neighbor Management
Multi-Robot Pose Graph Optimization
Communication Interface
External
Front-End
Back-End

Figure 7.2 Swarm-SLAM Overview. Our collaborative SLAM system is fully decentralized and supports intermittent inter-robot communication. Swarm-SLAM takes odometry and raw sensor data as input, performs place recognition and registration to produce loop closures, and dynamically elects a robot among the network neighbors to optimize the multi-robot pose graph.

7.3.3 Multi-Robot Mapping Datasets

Existing datasets in C-SLAM typically fall short of capturing realistic multi-robot scenarios, as they often involve only one robot at a time, resulting in the absence of dynamic objects, and systematically lack inter-robot network conditions estimates. This represents a significant gap in the literature, particularly regarding experimental data from planetary analogue environments where inter-robot communication is intermittent due to large distances and obstacles that cause non-line-of-sight conditions between robots.

One of the early efforts in multi-robot datasets was the UTIAS dataset by Leung et al. [174], which involved five robots operating within a single indoor room. This dataset set the groundwork for collaborative SLAM but was limited to a static, confined environment. More recently, Dubois et al. [175] proposed incorporating both ground and aerial robots in indoor settings, using stereo cameras. Collected during larger scale outdoor environment, Zhu et al. [233] introduce GrAco a multimodal dataset featuring ground and aerial LiDAR and stereo sequences captured on a college campus. This dataset offered more diverse environmental settings but remained limited to structured outdoor spaces. A notable step forward in scaling and realism came from Tian et al. [292], who developed an online dataset featuring up to eight robots equipped with cameras and LiDAR, operating in large-scale, indoor

Table 7.1 Content per robot of our novel dataset, comprising SLAM and inter-robot communication sensing data.

Sensor	Data Types	Frequency (Hz)
Ouster LiDAR OS0	Point Clouds	9.81
	IMU	98.54
Vector VN-100	IMU	196.32
GL iNet AX1800	Pairwise latency with each other robot (ms)	0.92
	Pairwise throughput with each other robot (Kbps)	0.93
U-Blox ZED F9 GPS	GPS fix	9.00

and outdoor environments with human-made structures. Feng et al. [232] introduced the S3E dataset, which specifically targets C-SLAM scenarios with three synchronized robots in multiple indoor and outdoor environments.

Most relevant to our work, Zhao et al. [234] addressed some of these limitations with the SubT-MRS dataset, which includes diverse robots operating in various environments, including challenging, degraded conditions similar to the planetary analogue field used in our experiments. This dataset significantly contributes to the field by simulating more realistic conditions for multi-robot systems. Our dataset aims to further advance robustness and resilience in degraded environments, while also providing data on real inter-robot communication capacity in the field. To that end, our dataset, described in Table 7.1, includes periodic pairwise latency and throughput estimates.

7.4 Experimental Setup

Our experiments were conducted in a Mars analogue field designed to simulate the challenging conditions of planetary surfaces. The terrain included a mix of sand, various types of rocks, slopes, and uneven ground, closely mimicking the environment that robots would encounter on actual space missions (see Fig. 7.3). These features pose significant challenges for robot mobility, perception, and communication, making this setup ideal for testing decentralized C-SLAM algorithms in realistic conditions.

The experiments involved three robots exploring the field simultaneously, each remotely controlled by human operators. This setup allowed for testing the robots' ability to maintain ad hoc communication and correctly localize themselves in a dynamic and degraded environment. As illustrated in Fig. 7.7, the robots followed roughly similar trajectories, but each

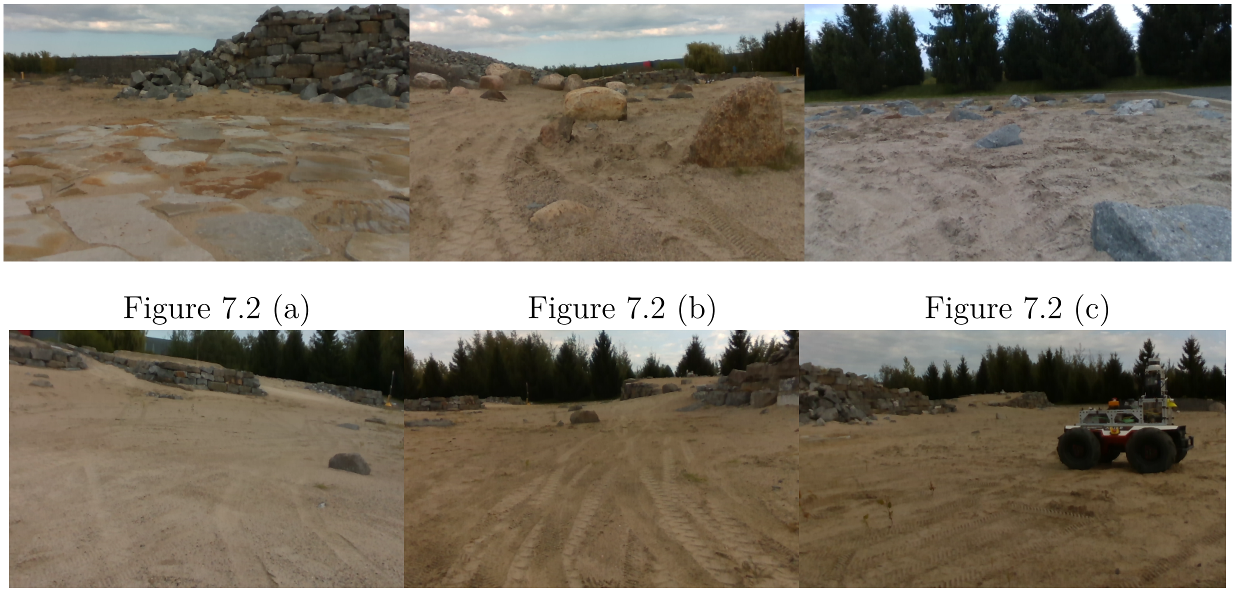


Figure 7.2 (a) Figure 7.2 (b) Figure 7.2 (c)

Figure 7.2 (d) Figure 7.2 (e) Figure 7.2 (f)

Figure 7.3 Images of the Canadian Space Agency Mars Yard terrain captured by cameras onboard the robots. Images (a) to (e) illustrate the diverse and challenging features of the terrain. Panel (f) demonstrates instances where the robots are occasionally within line-of-sight of each other, acting as challenging dynamic objects.

in different order and directions. This approach ensured that each robot covered nearly the entire field, maximizing overlap between the maps, and thus producing as many loop closing matches as possible for our system analysis in Section 7.6.

7.4.1 Robot Design

Our experiments were conducted using three robots: one AgileX Scout 2.0 Rover, shown in Fig. 7.4, and two AgileX Bunkers, shown in Fig. 7.5. All robots were mounted with NVIDIA Jetson Xavier (32GB) for data processing, GL iNet AX1800 router for wireless network, Ouster LiDAR OS0, Vector IMU VN-100 and U-Blox ZED F9 modules for GPS positioning (F9P for bunkers and F9R for the rover). Our Swarm-SLAM system was configured with LiDAR as the primary sensor for environment mapping and localization. For odometry, we used the LIO-SAM algorithm [293], which integrates LiDAR and inertial data to provide accurate and robust local pose estimation in real-time. For place recognition, we employed ScanContext [274], a method that generates compact descriptors of LiDAR scans for recognizing previously visited locations. To determine similarities between locations, we used cosine similarity to compare the ScanContext descriptors. To ensure robust 3D registration of the point clouds matched with ScanContext, we integrated TEASER++ [286], a state-of-the-art algorithm designed for fast and certifiable registration of point clouds. We applied a threshold of 80 inlier points required for a registration to be considered successful, ensuring that only high-confidence matches contributed to the collective map. To evaluate Swarm-SLAM's odometry estimates, the robots were equipped with U-Blox ZED GPS modules. The modules on the robots were programmed to correct their GPS estimates by receiving RTCM correction data from our GPS station. Our GPS RTK station comprises of a laptop connected to our ad-hoc network, and a U-Blox ZED GPS module. First the module is configured to perform the *survey-in* procedure, where the module calibrates its position with an accuracy of up to 20cm. After reaching the target accuracy, our GPS station starts a NTRIP server to broadcast the RTCM correction data on the network.

7.4.2 Ad-hoc Networking

We developed a Mobile Ad-Hoc Network (MANET) using a custom network stack implemented on GL iNet AX1800 routers, which run OpenWRT [294]. Each robot, along with the ground station, was equipped with a router responsible for managing both internal communications among the robot's hardware components and external communications with other robots via the MANET. The inter-robot links were based on IEEE 802.11s, combined with batman-adv, a dynamic link-state routing protocol operating at the data link layer (Layer 2). Batman-adv continuously broadcasts network updates, maintaining a routing table that ensures seamless communication between nodes (i.e., robots) throughout the deployment. The overall communication architecture is illustrated in Fig. 7.6.

The MANET backbone relies on IEEE 802.11s, configured without frame replication to optimize bandwidth usage and allow batman-adv to provide adaptive routing. Key parameters were adjusted to support fast adaptation to network changes (root mode, active path timeouts, etc.), allowing for rapid disconnection and reconnection under low signal conditions. The 2.4GHz and 5GHz radios were bonded into a shared interface managed by batman-adv, enabling dynamic frequency switching. This setup leverages the range of 2.4GHz, which is more susceptible to interference but offers wider coverage, and the higher bandwidth of 5GHz, which is better suited for shorter-range, high-data-rate communications. This dual-frequency capability allows the network to automatically choose the optimal frequency and routing path, supporting both direct and multi-hop communication. We also consider an optional 900MHz channel for specific use cases requiring long-range communication (several kilometers), which was not needed given the size of the CSA Mars Yard. In fact, we used the standard antennas present on the AX1800 routers to have a more challenging communication environment with ranges up to 40 meters, whereas high-gain antennas or more powerful

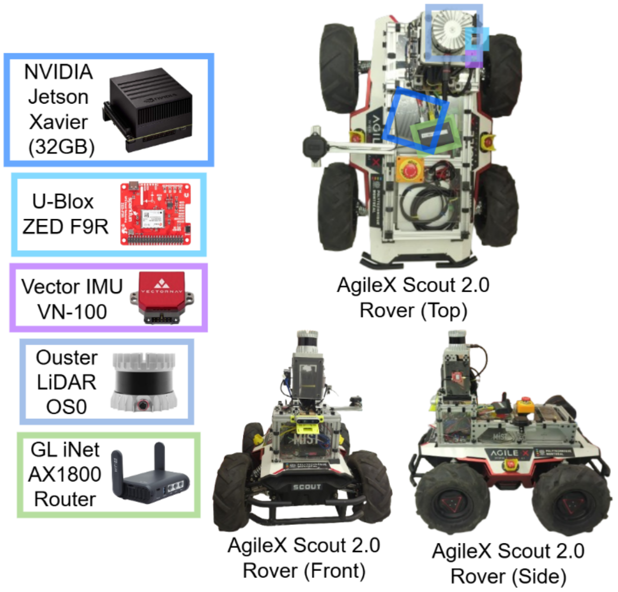

NVIDIA Jetson Xavier (32GB)
U-Blox ZED F9R
Vector IMU VN-100
Ouster LiDAR OS0
GL iNet AX1800 Router
AgileX Scout 2.0 Rover (Top)
AgileX Scout 2.0 Rover (Front)
AgileX Scout 2.0 Rover (Side)

Figure 7.4 Robot 1 design: The base of the robot is an AgileX Scout 2.0 Rover, equipped with a LiDAR, IMU sensor suite, and an ad-hoc networking router enabling peer-to-peer communication.

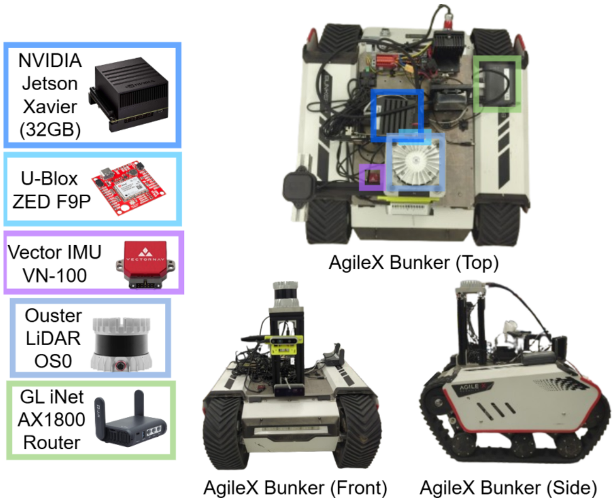


Figure 7.5 Robot 2 and 3 design: The robots are based on AgileX Bunker platforms, featuring the same sensor suite as Robot 1. These robots are equipped with tracks, enhancing mobility over rough terrain.

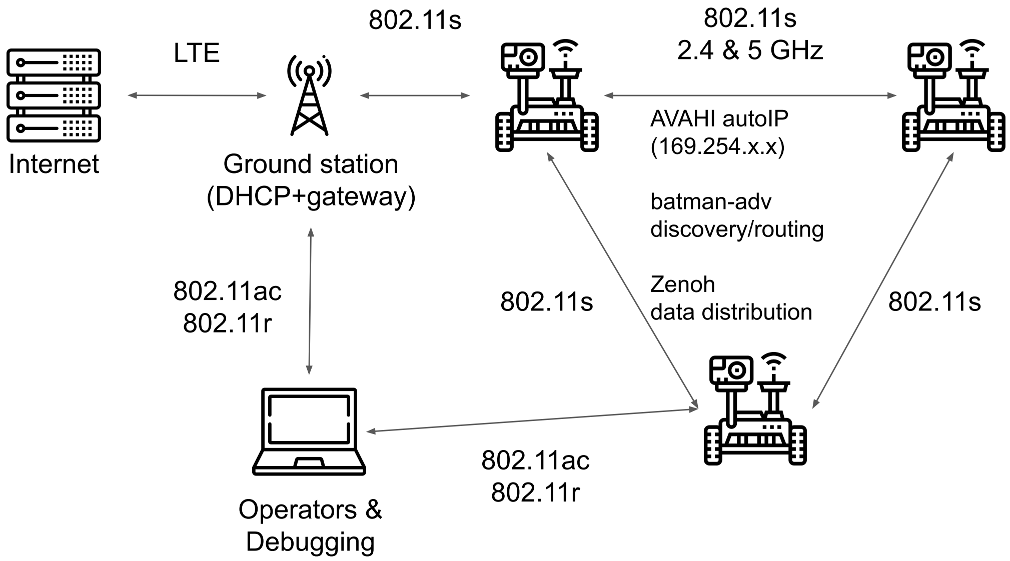


Figure 7.6 Network architecture: Robots are interconnected via 802.11s on bonded 2.4GHz and 5GHz interfaces, with routing provided by batman-adv and data distribution managed by Zenoh.

routers could have provided communication range up to hundreds of meters and covered the entire terrain.

To ensure connectivity with the operator computers, the bonded interface was bridged with a 802.11ac link in station mode and the robots' local Ethernet network. In other words, each robot acts as a wireless access point, and all the on-board computers share the same subnet. The network incorporates 802.11r for fast transitions, enabling operator laptops to seamlessly connect and switch between access points, whether on the ground station or directly to robots. All robots were configured to operate on a unified subnet, while VLANs were used to segregate local components outside the bridge, reducing interference and preventing flooding (e.g., isolating LiDAR data streams from other local network traffic). All computers were set up to use AVAHI AutoIP for service and name discovery under the .local domain. All data distribution between robots is provided by Zenoh [295], which offers a minimal overhead publish-subscribe communication API.

The ground station was configured similarly to the robots but had an additional gateway for Internet access. This gateway facilitated software updates for the robots and synchronized their system clocks, ensuring consistent timing across the network. Furthermore, the ground station acted as a control hub, enabling operators to monitor robot status, trigger behaviors, and manage overall mission coordination within the network's communication range.

7.4.3 Peer-to-Peer Bandwidth Estimation

To assess the communication performance between the robots, we conducted pairwise peer-to-peer communication estimations every second throughout our data collection experiments. We measured throughput using iperf [296], a tool designed for active measurements of the maximum achievable bandwidth. For latency measurements, we employed fping [297], a program that sends ICMP echo probes to calculate round-trip times between the robots. The robots clocks were all synchronized using NTP before data collection. This approach enabled us to monitor the latency of communications dynamically as the robots moved through the Mars analogue field, reflecting the impact of varying distances, obstacles, and network topology changes on communication delays. To ensure that our estimates were as accurate and reflective of real-world conditions as possible, we minimized additional network traffic by not running any other software that required data transmission.

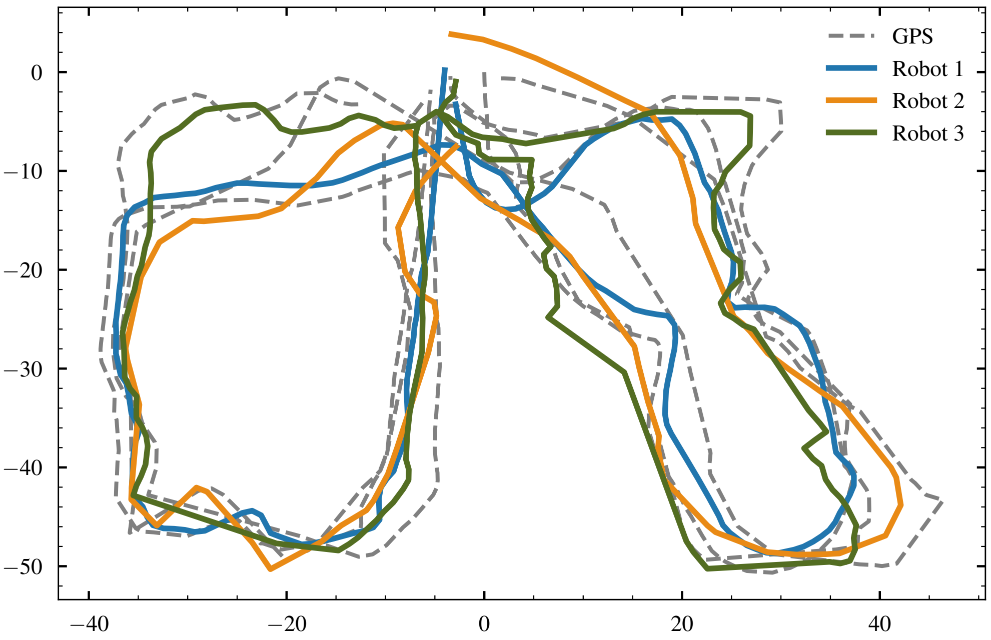


Figure 7.7 Swarm-SLAM Pose Graph Estimates and GPS ground truth. Absolute Translation Error: 3.74±1.63 meters.

7.5 Decentralized Collaborative SLAM

We used Swarm-SLAM to process the entire data sequences from the robots, generating the multi-robot pose graph solution illustrated in Fig. 7.7. The GPS ground truth data is also displayed for comparison. To quantify the accuracy of the SLAM estimates, we employed the evo software [298] to compute error metrics. The results showed an Average Translation Error (ATE) of 3.74 ±1.63 meters between the estimated poses and the ground truth. Interestingly, while reasonable, our results are far from perfect, suggesting that future research could leverage our dataset to develop new techniques that improve localization and mapping accuracy in C-SLAM within this type of difficult environment.

A detailed breakdown of the error distribution per robot is presented in Fig. 7.8. Notably, the error for Robot 1 is significantly lower than that of Robots 2 and 3. We hypothesize that this variation is due to differences in robot design. While Robot 1 features large wheels and suspension, which is well suited for navigation on sandy terrain with numerous small rocks, Robots 2 and 3 are equipped with tracks, which are less effective in these conditions. To verify this hypothesis, we analyzed the linear acceleration data from the IMUs embedded in the 3D LiDAR sensors of each robot, shown in Fig. 7.9. This revealed that Robots 2 and 3 experienced significantly higher vibration levels compared to Robot 1, indicating that the wheeled configuration of Robot 1 is more compatible with the challenging terrain.

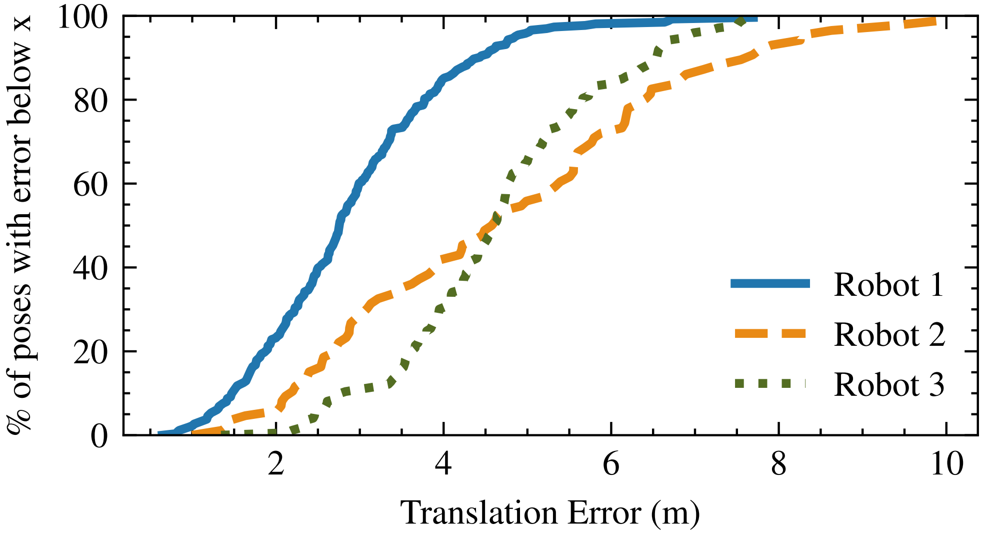


Figure 7.8 Absolute Transation Error Distribution. On the y-axis, we report the percentage of poses with absolute translation error below the value on the x-axis.

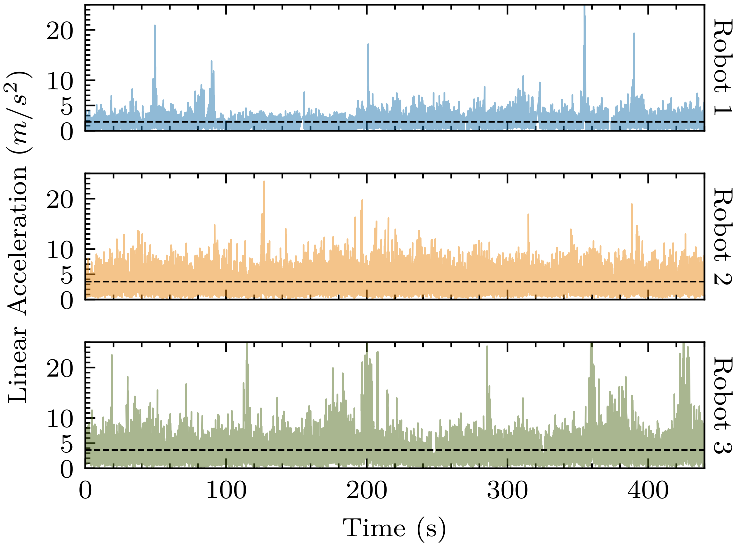


Figure 7.9 Linear acceleration magnitude from the onboard LiDAR IMUs. We removed the IMU bias and show the average as a dotted line for better comparison. We can see that the Robot 1 (on wheels) was less affected by vibrations than Robot 2 and 3 (on tracks).

These increased vibrations in Robots 2 and 3 likely led to reduced LiDAR odometry accuracy as it may cause the loss of points and hinder data association [299]. To confirm this, we measured the ATE for the odometry of each robot individually: Robot 1 exhibited an odometry ATE of 2.45±1.14 meters, while Robot 2 and Robot 3 had higher ATEs of 4.29 ±1.76 meters and 3.61 ±1.72 meters, respectively.

Despite the wheeled robot's better performance in terms of reduced error, it faced its own challenges. Robot 1, with its large wheels, was more susceptible to becoming stuck, especially in wet sand, whereas the track-equipped Robots 2 and 3 demonstrated superior traction and reliability.

7.6 Resource Efficiency

This section delves into the resource efficiency of our decentralized C-SLAM solution when deployed in planetary analogue environments. Decentralized C-SLAM systems must operate within the constraints of available computing power, memory, and inter-robot communication resources, requiring strategic trade-offs to ensure effective performance under these limited conditions.

In Section 7.6.1, we analyze the inter-robot communication metrics gathered during our experiments and compare them to the default requirements of our C-SLAM approach. This analysis provides a critical evaluation of the realism and feasibility of our solution in practical scenarios, assessing whether the system's communication demands are compatible with the actual network conditions experienced in the field.

Section 7.6.2 addresses the calibration of key parameters to tune the C-SLAM system for real-world deployments. We explore the trade-offs between map accuracy and available communication bandwidth, illustrating how adjusting these parameters can influence the system's overall performance. Clear understanding of these trade-offs is essential, as it enables more informed decisions regarding the suitability of the approach for different mission scenarios.

Moreover, well-defined trade-offs significantly enhance the tunability of the system. A solution that is easier to calibrate not only streamlines deployment but also increases the likelihood of adoption, particularly by users who are not SLAM experts. By simplifying the tuning process, the technology becomes more accessible and adaptable, making it a practical choice for a wider range of applications and user groups.

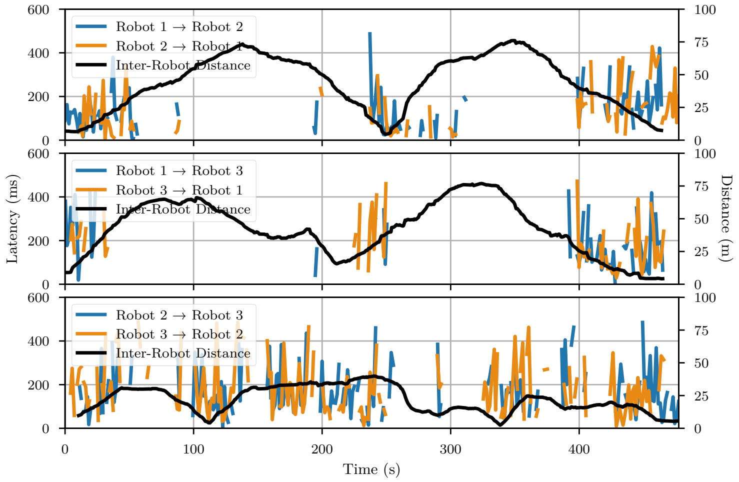


Figure 7.10 Inter-robot latency measurements between each pair of robots during the field mission. The plot shows latency values in both directions for all robot pairs, along side the inter-robot distance, illustrating the variability in communication delays and the impact of inter-robot distance on latency. Higher latency values highlight the challenges of maintaining real-time communication in decentralized C-SLAM deployments.

7.6.1 Ad-hoc Inter-Robot Communication

We analyzed the available pairwise peer-to-peer inter-robot communication bandwidth during our field mission, as peer-to-peer communication is the backbone of scalable and resilient multi-robot operations. A system that relies primarily on local inter-agent data transmission can better scale to large groups of robots because it avoids the need for a central communication node, which can become a bottleneck or critical point of failure [20]. To demonstrate the capabilities of our system, we present latency measurements in both directions for each of the three robot pairs in Fig. 7.10. The observed latencies range from 100ms to 400ms, which, even without considering computation time, imposes significant constraints on real-time C-SLAM deployment.

To address this challenge, our approach maintains a real-time, local single-robot SLAM estimates, which are periodically updated and corrected using the multi-robot estimates that incorporate the multi-robot pose graph and inter-robot loop closures. Additionally, Fig. 7.10 shows the distance between robots at each timestep, calculated using GPS data. The comparison between latency and inter-robot distance indicates that our networking setup tends to lose connectivity when robots are approximately 40 meters or more apart. This finding highlights the need for approaches that can handle disconnections and effectively recompute a consistent map when robots reconnect.

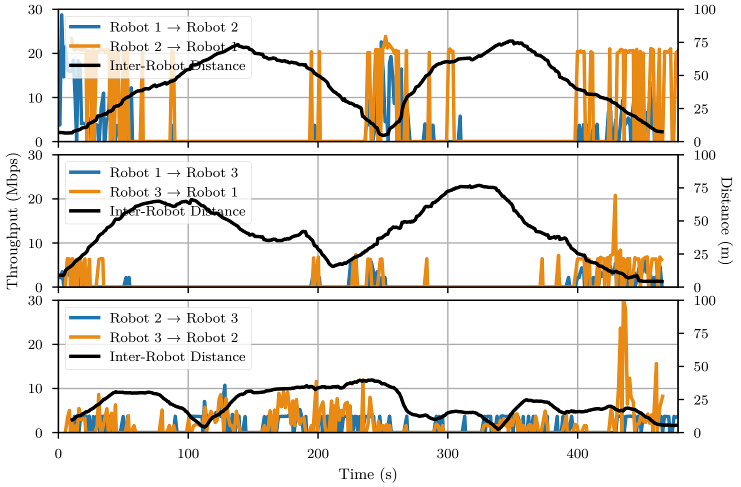


Figure 7.11 Inter-robot throughput between each pair of robots during the field mission, measured in both directions. The plot shows data transmission rates ranging from approximately 5 to 20 Mbps when the robots are within communication range, underscoring the tight communication constraints in decentralized C-SLAM systems.

In Fig. 7.11, we also present the throughput between robots in both directions for each pair, with values ranging from approximately 5 to 20 Mbps when connected. Comparing throughput with inter-robot distances confirms the consistency of our latency estimates with the throughput data.

Using these throughput estimates, Fig. 7.12 illustrates the accumulated communication throughput over time across all robot pairs, plotting the cumulative available communication bandwidth in megabytes at each timestep. We compared this field-measured bandwidth with the unconstrained bandwidth usage of Swarm-SLAM. To accurately measure the unconstrained bandwidth usage, or maximum bandwidth consumption, we ran three agents in parallel on a single machine, each processing sensor data from one robot in our dataset. We mesured all data transmission through the ROS 2 nodes of different agents, distinguishing between back-end and front-end processes to better identify their relative bandwidth requirements. Our findings indicate that front-end processes dominate the overall communication load. A key insight from Fig. 7.12 is that while available bandwidth is initially sufficient, communication demands rapidly increase, eventually exceeding the available capacity. This is attributed to the initially small individual robot maps with limited overlap, resulting in minimal need for resource-intensive 3D registration. However, as each robot explores and expands its map, the level of overlap with other robots increase. Subsequently, the number of successful place recognition matches grows, leading to a rise in communication requirements. Interestingly, this shows that although increased map overlap will ultimately enhance map

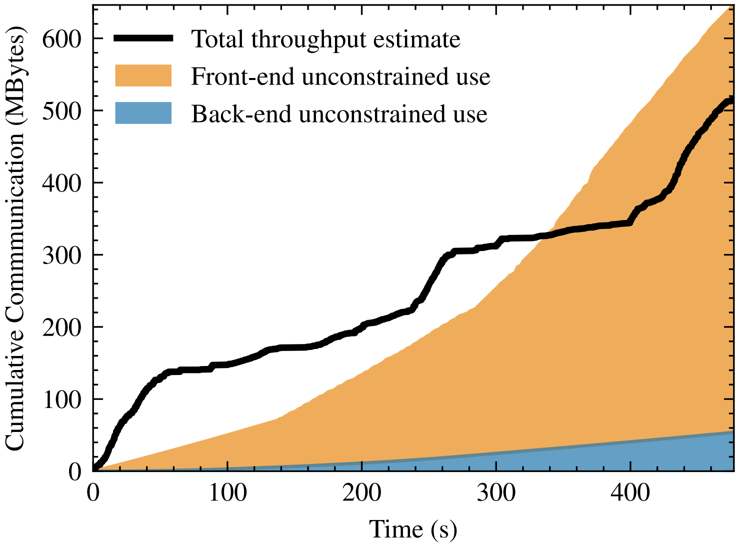


Figure 7.12 Comparison of communication usage versus total available throughput over time during the field mission. The plot shows the cumulative communication bandwidth utilized by the C-SLAM system against the measured total throughput, highlighting how communication demands increase and eventually exceed available capacity.

merging accuracy, it also demands more communication and computation to be effectively processed. Therefore, our experiments represent a challenging scenario in terms of inter-robot bandwidth due to the substantial overlap between robot trajectories and maps, making this a valuable case study for understanding the communication limits of C-SLAM systems.

It is important to note that the results in Fig. 7.12 were obtained using default parameters: a ScanContext cosine similarity threshold of 0.7 and a minimum of 80 inlier points for registration. As will be discussed in Section 7.6.2, these settings generate numerous loop closure candidates, including incorrect ones, and may not be the most communication-efficient. We will explore how strategic tuning of these parameters can enhance communication performance without compromising the accuracy of the C-SLAM solution.

7.6.2 Impacts of C-SLAM Calibration

Calibrating key parameters of decentralized C-SLAM systems is crucial for meeting communication constraints, which is especially important in resource-limited environments like planetary analogue settings.

Communication Budget

In Swarm-SLAM [284], we introduced a communication budget, defined as the number of inter-robot loop closure matches selected from all candidate matches identified through place recognition. This budget uses a spectral sparsification approach to prioritize candidate matches before they are send to the, more communication and computation-intensive, 3D registration step, as shown in Fig. 7.2.

The prioritization process focuses on selecting matches that are most likely to improve the accuracy of the multi-robot pose graph, making it particularly valuable when high map overlap generates an excess of place recognition matches beyond available resources. It is also useful when robots reconnect after extended periods of disconnection, during which they accumulate a backlog of place recognition matches that could take significant time and communication to process entirely. By carefully prioritizing which matches to process, we can improve the trade-off between communication and accuracy, allowing a small number of well-chosen loop closures to closely approximate the optimal C-SLAM solution.

Therefore, the communication budget directly controls how much data Swarm-SLAM transmits at each timestep. In Fig. 7.13, we show the optimal match selection budget at each timestep versus the cumulative throughput. In practice, this budget is often set to a fixed value, but our results suggest that adapting the budget dynamically could better utilize available bandwidth. However, evaluating throughput online during experiments poses a challenge as most estimation techniques require sending large volumes of data to test the limits of the network, which could interfere with ongoing communication.

Place Recognition

In Fig. 7.14, we explore the relationship between the number of loop closures (y-axis) and the place recognition similarity threshold (x-axis). We categorize loop closures into three groups: correct loop closures (in green) with translation errors below the average error of the optimized multi-robot pose graph, less accurate or incorrect loop closures (in yellow) with errors above the average, and failed matches (in red) where high descriptor similarity did not result in successful registration due to insufficient inlier points. The plot demonstrates that lower, less conservative, similarity thresholds result in more loop closures, but many of these are incorrect or failed registrations. Conversely, increasing the threshold reduces incorrect matches but also significantly decreases the number of correct ones within a certain range (0.7 to 0.85), revealing a trade-off between conservativeness and loop closure quantity. This gap indicates that current similarity measures, such as those used in ScanContext, cannot

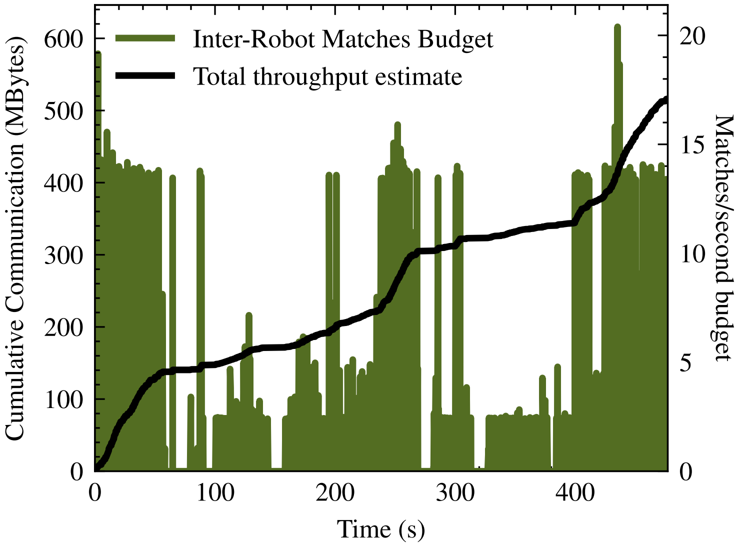


Figure 7.13 Communication budget versus total available throughput during the field mission. The plot illustrates the relationship between the communication budget—defined as the number of prioritized inter-robot loop closures—and the cumulative available throughput. This emphasizes the need for adaptive communication strategies to optimize bandwidth usage in decentralized multi-robot systems.

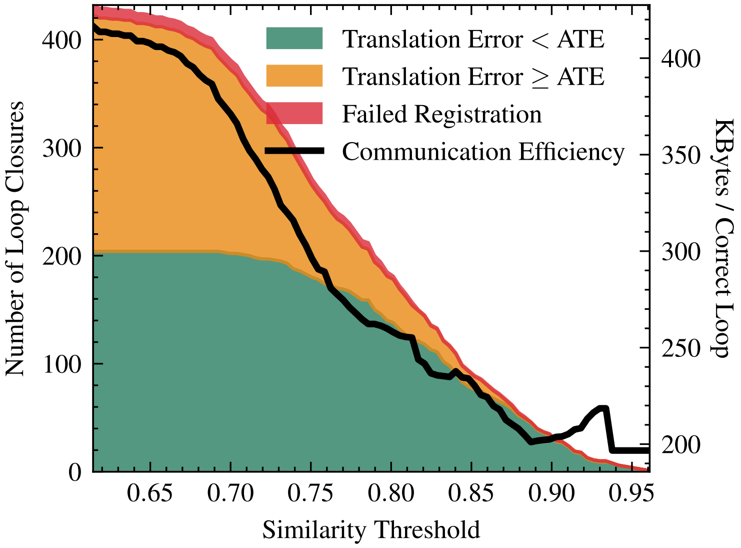


Figure 7.14 Number of loop closures versus place recognition similarity threshold. The plot categorizes loop closures into correct, incorrect, and failed registrations, showing how varying the similarity threshold impacts the quantity and quality of loop closures. This highlights the trade-offs involved in setting similarity thresholds for optimal performance.

perfectly predict the quality of loop closures post-registration.

To isolate the effect of the similarity threshold, we used a very loose threshold of only 10 inlier points for subsequent 3D registration. Although robust pose graph optimizers like GNC [140] can tolerate some incorrect or outlier loop closures, they incur significant computational costs, often requiring several seconds compared to milliseconds for standard optimizers. Reducing the number of incorrect loop closures through better front-end calibration could thus lead to notable efficiency gains in back-end optimization.

In the same figure, the right y-axis shows communication efficiency in terms of KBytes per correct loop closure. Generally, more conservative similarity thresholds lead to better communication efficiency due to fewer incorrect loop closures. However, while conservative thresholds may work well in high-overlap scenarios like our experiments, they risk missing inter-robot loop closures in environments with less map overlap, where loop closures are more rare. Thus, in low-overlap scenarios, it may be advisable to use less conservative thresholds in order to perform map merging and achieve a C-SLAM solution.

Registration

In Fig. 7.15, we examine the number of loop closures (correct in green, incorrect in yellow) relative to the number of inlier points during registration, using a very low similarity threshold of 0.1 to focus on the number of inliers effect. The results show that, again, there is a trade-off between setting more conservative thresholds and the total number of loop closures. Unfortunately, this parameter alone is not a reliable predictor of loop closure accuracy, as some loop closures with over 300 inliers were still incorrect. Consequently, more conservative thresholds can worsen communication efficiency, as they reduce correct loops without effectively filtering out incorrect ones.

As shown in Figs. 7.14 and 7.15, our experiments indicate that planetary analogue environments are prone to place recognition outliers and inaccurate 3D registrations. The flat terrain and lack of distinctive features often cause different places to appear similar. In Fig. 7.16, we illustrate this phenomenon with two point clouds with a high ScanContext similarity of 0.757 and a substantial number of inliers (381), which, despite looking similar, represent distinct locations 17.28 meters apart. Importantly, as we have shown, these outliers not only affect overall accuracy but also negatively impact communication efficiency.

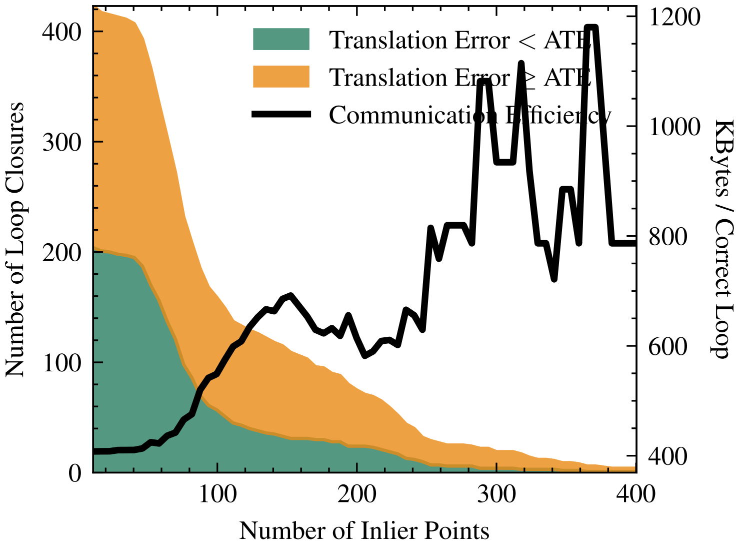


Figure 7.15 Number of loop closures versus the number of inlier points during 3D registration. The plot distinguishes between correct and incorrect loop closures, demonstrating how varying the inlier threshold affects the accuracy and quantity of loop closures.

7.7 Conclusions and Open Challenges

In this paper, we presented a comprehensive evaluation of decentralized C-SLAM in planetary analogue environments, addressing the unique challenges posed by difficult terrain, limited resources, and the need for efficient communication strategies. Our experiments highlighted several critical insights into deploying decentralized C-SLAM in such challenging settings, where the terrain affects robot mobility and sensor data quality due to vibrations and uneven surfaces. These conditions underscore the need for robust, adaptable SLAM algorithms capable of maintaining accuracy in uncontrolled environments.

One of the primary challenges identified is the constraint imposed by limited resources—namely, communication bandwidth, computational power, and memory. Effective operation in such environments demands careful and adaptive tuning of system parameters, with a significant emphasis on optimizing communication, which frequently emerges as the most limiting factor. Our findings show that the C-SLAM front-end consumes the bulk of the communication bandwidth, underscoring the need for future research to reduce its demands. Potential strategies include more efficient data representation, compression, and selective data sharing, which can alleviate the communication burden without compromising performance.

Moreover, our study revealed a persistent trade-off between communication and accuracy in current C-SLAM approaches. Enhancing this trade-off remains an open challenge, with fu-

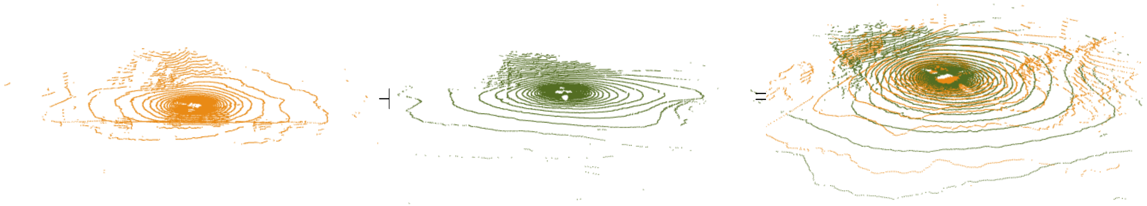


Figure 7.16 Ambiguity in LiDAR scan matching, showing two scans with a ScanContext similarity of 0.757 and 381 inlier points during 3D registration. Despite the high similarity and significant number of inliers, the ground truth distance between the two scans is 17.28 meters, illustrating the challenge of accurately distinguishing distinct locations in feature-sparse, flat terrain environments. This example highlights the limitations of current place recognition metrics in differentiating between similar but distinct places.

ture research potentially focusing on improving accuracy without proportionately increasing communication and computational demands. This could be achieved by refining existing algorithms, exploring novel sensing and mapping paradigms, or developing more efficient data fusion techniques.

Our Swarm-SLAM approach is intentionally designed to be general and applicable across a wide range of scenarios, beyond just planetary analogue environments. It relies on inexpensive onboard sensors and simple peer-to-peer communication links, making it particularly suitable for early space missions where permanent networking or localization infrastructure—such as satellites or base stations—has yet to be established. However, the integration of infrastructure like orbital satellites or base stations with long-range, high-power networking and localization capabilities could significantly enhance SLAM performance by providing external or global sensing for the entire group of robots.

We believe that, in the future, the most effective approaches for space exploration will involve a hybrid strategy that fuses local sensing and estimation with global sensing capabilities. This fusion would ensure safe and reliable autonomy through local sensing and allow for decentralized inter-robot mapping during periods of communication loss or base station outages, while benefiting from external sensing and larger computing resources when available. This balanced integration of local autonomy and global coordination is key to overcome the unique challenges of operating in extraterrestrial environments.

Ultimately, advancing C-SLAM technology for space exploration will require continuous refinement of these adaptive strategies to meet the evolving demands of complex and resource-constrained environments. By enhancing our understanding of the trade-offs between communication, computation, and accuracy, we can better equip multi-robot systems to navigate and map new frontiers—whether on Earth, the Moon, Mars, or beyond.

CHAPTER 8 ARTICLE 5 : MOLD-SLAM: MINIMAL OVERLAP LOOP DETECTION FOR MULTI-ROBOT COLLABORATIVE VISUAL SLAM

Preface: This paper introduces MOLD-SLAM, a decentralized collaborative SLAM (CSLAM) approach that uses 3D foundation models to generate reliable inter-robot loop closures even with large viewpoint differences. MOLD improves localization accuracy, resolves 3D scale ambiguities, and outperforms existing methods in accuracy, efficiency, and scalability for multi-robot applications.
This work has been submitted to the IEEE Transactions on Robotics.
Contributions: This project was conducted jointly at the MIST Laboratory at Polytechnique Montréal and the Mobile Robotics Group at Oxford University. The project was conceptualized in close collaboration with Benjamin Ramtoula, under the guidance of Daniele De Martini and my supervisor, Giovanni Beltrame. I implemented the proposed approach, conducted the experiments, and wrote the majority of the paper, regularly incorporating feedback and discussing preliminary results with my coauthors.
Full Citation: Pierre-Yves Lajoie, Benjamin Ramtoula, Daniele De Martini, Giovanni Beltrame, "MOLD-SLAM: Minimal-Overlap Loop Detection for Multi-Robot Collaborative Visual SLAM," *Submitted to IEEE Transactions on Robotics*, 2024.
Submission date: November 3rd 2024

8.1 Abstract

Decentralized C-SLAM often faces challenges in identifying map overlaps due to differences in viewpoints among robots, resulting from varying trajectories and sensor placements. Motivated by the capability of recent 3D foundation models to successfully register images with large viewpoint differences, we present MOLD-SLAM. Our approach consists in a novel decentralized C-SLAM approach that leverages a 3D foundation model to generate inter-robot loop closures. Our contributions include: 1) integrating 3D foundation models for pose estimation from pairs of monocular images into C-SLAM; 2) developing robust techniques to mitigate outliers; and 3) proposing pose graph formulations that efficiently merge individual robot maps and resolve 3D scale ambiguities. Experimental results demonstrate the effectiveness of MOLD-SLAM in enhancing C-SLAM performance compared to state-of-the-art approaches. We evaluate our system's performance in terms of accuracy, as well as computational, memory, and communication efficiency, highlighting its potential for large-scale multi-robot applications.

8.2 Introduction

Decentralized Collaborative Simultaneous Localization and Mapping (C-SLAM) is a critical capability for multi-robot systems operating in unknown environments. In these scenarios, as illustrated in Fig. 8.1, multiple robots must explore the environment independently, while exchanging information to build a shared global map. This task becomes particularly challenging when the robots' viewpoints differ significantly due to varying trajectories and diverse sensor placements, making it difficult to establish overlapping map sections and generate reliable inter-robot loop closures [284]. As a result, many existing C-SLAM systems struggle to achieve robust map fusion and accurate localization in such situations, limiting their performance and scalability.

To address these challenges, we propose Minimal Overlap Loop Detection SLAM (MOLD-SLAM), a novel decentralized C-SLAM framework that leverages the latest advancements in 3D foundation models [237, 238] to improve inter-robot loop-closure detection. The core idea behind MOLD-SLAM is to exploit the impressive ability of 3D foundation models, shown in Fig. 8.2, to perform relative-pose estimation from pairs of monocular images with minimal overlap between the robots' observed areas, or even from opposite viewpoints.

MOLD-SLAM introduces three main contributions:

- the integration of a 3D foundation model (MASt3R [238]) into the C-SLAM pipeline to estimate relative poses between robots based on monocular image pairs, providing a means to detect inter-robot loop closures in situations with limited viewpoint overlap;
- outlier detection and uncertainty modelling specifically designed for 3D relative pose estimation with MASt3R, to ensure that only reliable loop closures are used for map fusion;

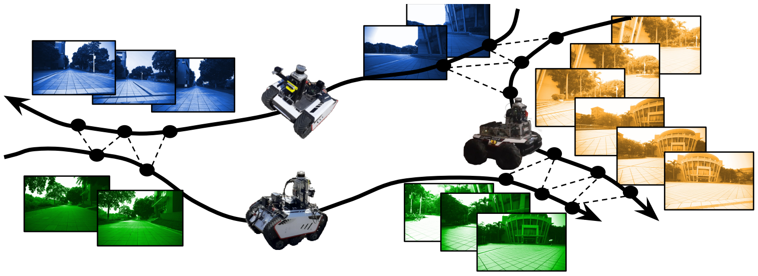


Figure 8.1 Illustration of the decentralized C-SLAM problem. Each robot independently explores an unknown environment, generating its own map. Upon meeting, the robots exchange data and compute inter-robot loop closures to register and eventually fuse their maps, thereby enhancing localization accuracy and enabling collaborative behaviors.

- a set of specialized pose-graph optimization formulations to merge individual robot maps, resolving 3D scale ambiguities of the generated loop closures and refining the overall localization accuracy.

Overall, our contributions enable us to leverage recent advances in 3D foundation models to enhance the robustness of C-SLAM and its performance across a wider range of environments and multi-robot missions. We evaluate the performance of MOLD-SLAM against state-of-the-art decentralized C-SLAM algorithms in multiple multi-robot dataset sequences. Our results demonstrate that the powerful representations generated by 3D foundation models enable substantial improvements in localization accuracy while reducing computational, memory, and communication overhead through specialized optimization and keyframe sparsification. This makes MOLD-SLAM a promising solution for large-scale multi-robot deployments in unknown environments, where inter-robot collaboration is crucial for efficient exploration and mapping.

The remainder of the paper is organized as follows: Section 8.3 reviews related work in image-based relative pose estimation and decentralized C-SLAM; Section 8.4 details the proposed MOLD-SLAM approach; and Section 8.5 presents the experimental results, comparing MOLD-SLAM with existing approaches, and a detailed ablation study to validate the effectiveness of initialization, outlier-rejection, and optimization variations.

8.3 Background and Related Work

In C-SLAM, multiple robots work together to build a shared map and localize within it. By sharing sensor data and detecting overlap between their maps, the robots can improve their individual localization, and create a globally consistent view of the environment across the robots. In this section, we present the background and related work on C-SLAM, as well as on image-based relative-pose estimation used to create inter-robot loop closures.

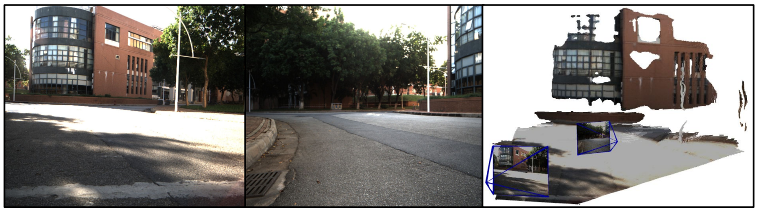


Figure 8.2 Successful registration using MASt3R with minimal image overlap, illustrating its robustness in handling challenging loop closures.

Collaborative SLAM

In C-SLAM, robots typically perform SLAM individually and then share information about their maps to fuse them into a globally consistent estimate of the traversed environment. Similar to single-robot SLAM, C-SLAM is typically divided in two parts: the front-end, which is responsible for feature extraction and data association, and the back-end, which manages state estimation [1].

One of the most challenging task of the front-end is efficiently detecting and computing inter-robot loop closures. These loop closures correspond to connections between independent robots' estimates that can be discovered when the same places are visited by different robots. They serve as stitching points to merge local maps into a global representation of the environment. To efficiently merge large maps, loop closure detection is typically performed in two stages: place recognition, in which compact descriptors are shared to identify possible map overlaps, followed by registration to compute the 3D relative pose between the individual overlaps.

The back-end then estimates the most likely poses and map based on measurements collected from all robots. Our work focuses on pose-graph formulations of SLAM, where features are marginalized into inter-pose measurements, as this approach is generally more efficient for handling large maps [252].

However, the perceptual aliasing phenomenon, where distinct places are mistaken for the same location, can cause front-end techniques to fail, leading to spurious measurements. Several methods have been proposed to address this issue, such as Pairwise Consistency Maximization [143], which was adapted for multi-robot SLAM in [62], or Graduated Non-Convexity (GNC) [140], for which a distributed version [63], based-on Riemannian Block Coordinate Descent (RBCD) [126], was developed.

Several complete C-SLAM systems have been proposed. Notably, among decentralized approaches, DSLAM [37] was one of the first to employ compact learned descriptors for efficient distributed place recognition in the front-end. Systems like Kimera-Multi [63] further enhance decentralized SLAM by incorporating robust distributed back-end solvers. On the other hand, centralized approaches such as LAMP 2.0 [276] and maplab 2.0 [220] were designed to efficiently manage heterogeneous groups of robots. More recently, Swarm-SLAM [284] built on these advancements and introduces a sparsification technique to prioritize inter-robot loop closures and improve the front-end efficiency.

8.3.1 Relative Pose Estimation

Producing globally consistent maps in C-SLAM heavily relies on the ability to accurately relate different robots' poses based on their observations. This step is performed by relative pose estimation, which involves performing 3D registration between two keyframes. In the collaborative case, these come from different robots.

Classical techniques such as Iterative Closest Point (ICP) [286] are widely used for LiDAR-based registration, while stereo systems typically rely on bundle adjustment for accurate pose estimation [269]. In monocular setups, geometric methods are usually employed [73], though they can only estimate relative poses up-to-scale, meaning the absolute metric scale of the transformation remains unknown. While scale information for consecutive images can be recovered by combining image-based estimates with motion sensors such as IMUs [300], these sensors cannot be used to recover the scale of relative poses between non-consecutive images. Moreover, compared to consecutive images, non-consecutive ones typically exhibit larger viewpoint differences, where traditional keypoint-based matching methods—such as those relying on hand-crafted features [301]—often struggle. Recent advancements in monocular depth estimation – such as DPT [302] – offer promising alternatives for scale estimation. However, despite their potential, these methods often require domain-specific fine-tuning, which can limit their generalizability when mapping unknown environments, a common use case for SLAM.

Recent advancements in learning-based approaches seek to overcome viewpoint limitations by enhancing keypoint detection and description. Methods like SuperPoint [303] and SuperGlue [304] use deep-learning techniques to enhance keypoint-matching robustness, incorporating reasoning across the entire image to improve performance. The field has also introduced new datasets and benchmarks that push the limits of pose estimation under extreme conditions. For example, the Map-Free challenge [239] focuses on scenarios with drastic viewpoint and illumination changes, requiring the emergence of new techniques to succeed, such as MicKey [236] and DUSt3R [237]. The first predicts metric correspondences directly in 3D camera space instead of the usual 2D pixel space, while DUSt3R reformulates the image matching problem as a pairwise 3D reconstruction task, predicting and aligning 3D pointmaps to estimate relative poses. In further work, MonST3R [305] extends DUSt3R to infer time-varying 3D reconstructions and view poses in dynamic image sequences. Also building on DUSt3R, MASt3R [238] regresses local features and explicitly trains for pairwise matching, further advancing matching performance under challenging conditions.

In summary, while classical methods continue to play an important role in 3D relative pose estimation, recent advancements in learning-based matching techniques and the introduction of new benchmarks are rapidly advancing the field. An emerging trend in this space is the use of foundation models, which demonstrate robustness across diverse domains due to their large-scale training. These models, such as DUSt3R [237] and MASt3R [238], show great potential for enhancing the accuracy and scalability of C-SLAM by improving cross-domain generalization and enabling more reliable matching under challenging conditions. We believe these innovations are opening new opportunities for advancing C-SLAM.

8.4 MOLD-SLAM

Our decentralized pose-graph visual C-SLAM approach aims to estimate the trajectories of multiple collaborating robots based on shared map measurements. This allows the robots to correct drift in their localization estimates and establish a shared situational awareness within the environment.

Our approach, illustrated in Fig. 8.3, is independent of the odometry source used by individual robots, treating odometry as an external input to the system. This design choice, as also adopted in [220, 284], offers several advantages: it enhances the generalizability of our approach, making it adaptable to various robots, as the optimal odometry technique is often closely linked to the specific configuration of each robot (e.g., sensor type, sensor placement, and calibration). Furthermore, it allows us to marginalize and discard dense map features tracked by the odometry software, thereby reducing memory usage. Finally, it enables the use of different sensors for odometry and loop closure, as different sensors may be better suited for different tasks – in this case, motion estimation through tracking consecutive frames versus recognizing previously visited places.

Importantly, in MOLD-SLAM, we assume that the input odometry has a correct metric scale, e.g. via visual-inertial [300], stereo [269], or LiDAR-based [306] systems. We define the input odometry pose estimates of a robot α as:

$$T_{\alpha,0}, T_{\alpha,1}, \ldots, T_{\alpha,i}, \ldots, T_{\alpha,n} \in \mathrm{SE}(3), \tag{8.1}$$

where $T_{\alpha,i}$ represents the pose estimate of robot α at each of the $i \in n$ keyframes along its trajectory. The number and sparsity of these keyframes, which impact memory and compute time, depend on the keyframing strategy. In this paper, we adopt a simple and widely used approach based on the distance traveled: a new keyframe is added to the trajectory whenever the robot has moved more than an estimated distance d_{kf}. Alongside each keyframe, our approach also takes as input the corresponding image frame I_i^{α}, which will be used for inter-robot loop-closure detection.

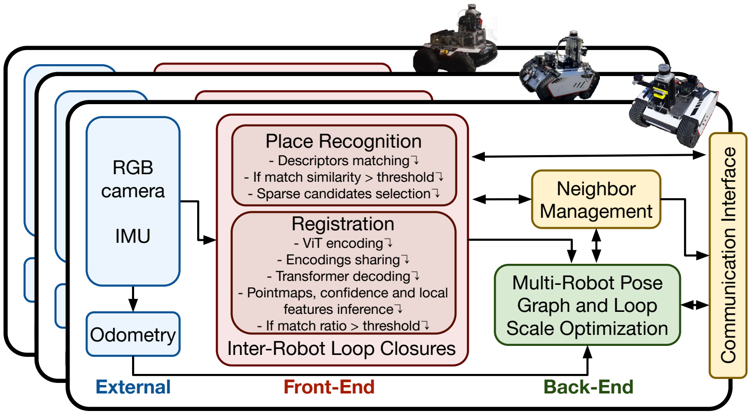

RGB camera
IMU
Odometry
Place Recognition
- Descriptors matching
- If match similarity > threshold
- Sparse candidates selection
Registration
- ViT encoding
- Encodings sharing
- Transformer decoding
- Pointmaps, confidence and local features inference
- If match ratio > threshold
Inter-Robot Loop Closures
Neighbor Management
Multi-Robot Pose Graph and Loop Scale Optimization
Communication Interface
External
Front-End
Back-End

Figure 8.3 System overview of the proposed MOLD-SLAM framework for decentralized C-SLAM. The figure illustrates the key components, highlighting how inter-robot loop closures are generated using a 3D foundation model, the multi-robot pose graph and loop scale optimization. Robots communicate with each other during place recognition, registration, and optimization, distributing tasks among neighboring robots to prevent duplicate processing.

8.4.1 Place Recognition

The first step of our pipeline is to perform place recognition between the maps of different robots. By identifying locations visited by two or more robots, we can create inter-robot loop closures that link the individual robot pose graphs into a consistent, shared localization estimate. For place recognition, we utilize CosPlace [273], a full-image feature extractor specifically trained for this task. Locally, each robot extracts a feature descriptor f_i^α for each of its keyframes as:

$$f_i^\alpha = \mathrm{CosPlace}(I_i^\alpha). \tag{8.2}$$

When two or more robots are within communication range – i.e., they become “neighbors” – the robots exchange their compact CosPlace descriptors with one another. The neighbor management, data sharing strategy, and bookkeeping follow the methods outlined in Swarm-SLAM [284].

Once a robot α has received descriptors from a neighboring robot β, it compares them with its own descriptors using cosine similarity. For each pair of keyframes (I_i^α, I_j^β), we compute the similarity score $s_{\beta,j}^{\alpha,i}$ as:

$$s_{\beta,j}^{\alpha,i} = \frac{f_i^\alpha \cdot f_j^\beta}{||f_i^\alpha||\;||f_j^\beta||}. \tag{8.3}$$

The best match for each keyframe is determined through a nearest neighbor search and, if the best match similarity exceeds a predetermined threshold, the corresponding pair of keyframes (I_i^α, I_j^β) is considered an inter-robot loop-closure candidate. These candidates are then passed to the subsequent registration step, where the relative pose $T_{\beta,j}^{\alpha,i}$ between the two keyframes is computed.

8.4.2 Registration

While previous techniques [62, 63, 284] relied on 3D registration using hand-crafted image features that are sensitive to viewpoint changes, a key novelty in our work is that we leverage recent foundation models for 3D image matching [237, 238]. Such models can robustly infer accurate relative poses between images captured from significantly different viewpoints (see Fig. 8.2), even in new unseen environments. This capability allows us to generate more inter-robot loop closures.

Whereas traditional C-SLAM methods use stereo, RGB-D, or LiDAR data for relative pose estimation with accurate scale, we rely solely on monocular images and resolve the scale ambiguity at a later stage (see Section 8.4.3). For registration, each image of a loop closure candidate (I_i^α, I_j^β) is encoded locally on its corresponding robot with the ViT encoder of the MASt3R model [238], as illustrated in Fig. 8.4. The encoding of I_j^β is then shared with robot α for decoding, feature matching, and inter-robot relative pose inference. This process yields a relative pose measurement $T_{\beta,j}^{\alpha,i}$ along with the number of feature correspondences between the two frames.

For confidence estimation, robot α also performs MASt3R inference on the pair of images $(I_{i-1}^\alpha, I_i^\alpha)$, for which the odometry estimates T_{i-1}^α and T_i^α are known. We then compute the ratio $r_{\beta,j}^{\alpha,i}$, which is defined as the number of correspondences for the loop closure pair (I_i^α, I_j^β) divided by the number of correspondences for the odometry pair $(I_{i-1}^\alpha, I_i^\alpha)$:

$$r_{\beta,j}^{\alpha,i} = \frac{|(I_i^\alpha, I_j^\beta) \text{ feature correspondences}|}{|(I_{i-i}^\alpha, I_i^\alpha) \text{ feature correspondences}|} \tag{8.4}$$

As illustrated in Fig. 8.5, the odometry pair $(I_{i-i}^\alpha, I_i^\alpha)$ typically exhibits a higher number of matching points due to their similar viewpoints, whereas the loop closure pair has fewer correspondences due to larger viewpoint differences. Assuming that the odometry pair is a reliable match for MASt3R, the ratio $r_{\beta,j}^{\alpha,i}$ serves as a good surrogate for evaluating the confidence of the loop closure match relative to the high-confidence odometry match.

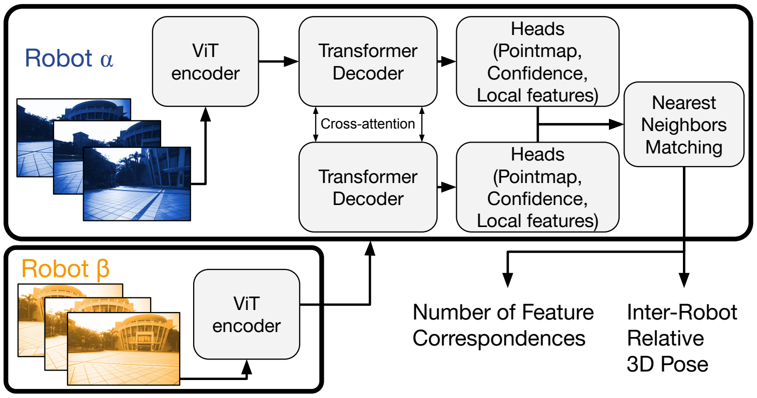

Robot α
ViT encoder
Transformer Decoder
Cross-attention
Transformer Decoder
Heads (Pointmap, Confidence, Local features)
Heads (Pointmap, Confidence, Local features)
Nearest Neighbors Matching
Robot β
ViT encoder
Number of Feature Correspondences
Inter-Robot Relative 3D Pose

Figure 8.4 Illustration of the MASt3R inference pipeline, highlighting how the processing is divided between two robots to compute an inter-robot loop closure.

We use this ratio in two ways: first, we use it to filter out failed registrations when below a minimum threshold R_{thr}. Despite the impressive results of MASt3R, some minimal overlap is still required to successfully infer the relative pose.

Second, we use it to inform the pose graph optimisation. For loop closures that pass the filtering threshold, we map the ratio to an estimated probability, $p^{\alpha,i}_{\beta,j}$, using a sigmoid function:

$$p^{\alpha,i}_{\beta,j} = \left[1 + \exp(-k \cdot (r^{\alpha,i}_{\beta,j} - 1))\right]^{-1}. \tag{8.5}$$

The probability and its parameter k can be manually tuned, or learned if training data from the deployment or a similar environment is available a priori [156, 283].

The confidence probability is then multiplied to the information matrix associated with each measurement. The information matrix acts as a weight during pose graph optimization, ensuring that high-confidence loop closures have a greater influence on the final solution than low-confidence ones.

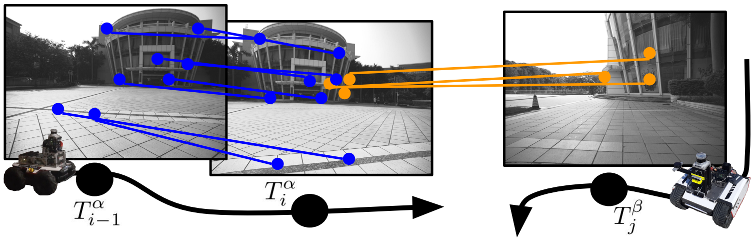


Figure 8.5 Example triplet of images used for inference, illustrating the ratio of correspondences between loop frames (in orange) and odometry frame pairs (in blue).

For each inter-robot loop closure, we can define a cost function $\phi_{\beta,j}^{\alpha,i}$ as follows:

$$\phi_{\beta,j}^{\alpha,i} = \left\| T_{\alpha,i}^{-1} \cdot T_{\beta,j} - \bar{T}_{\beta,j}^{\alpha,i} \right\|^2_{\Omega_{\beta,j}^{\alpha,i}(p_{\beta,j}^{\alpha,i})} \tag{8.6}$$

where $\bar{T}_{\beta,j}^{\alpha,i}$ is the relative pose measurement, and $\Omega_{\beta,j}^{\alpha,i}$ is the measurement information matrix which is proportional to the confidence $p_{\beta,j}^{\alpha,i}$. This cost function can then be incorporated into a global nonlinear least-squares problem alongside odometry measurements to perform multi-robot pose graph optimization.

8.4.3 Multi-Robot Pose Graph And Loop Scale Optimization

In our decentralized C-SLAM framework, multi-robot pose graph optimization ensures that the maps and estimated trajectories of all robots are aligned within a common reference frame, providing a unified representation of the environment. For clarity and ease of visualization, the optimization problems can be formulated as a factor graph, where the variables represent the unknown quantities (e.g., the robot poses), and the factors define functions over subsets of these variables (see Fig. 8.6). The factor graph is optimized by a dynamically elected robot within the group of neighboring robots as in [284].

Factor graph formulation

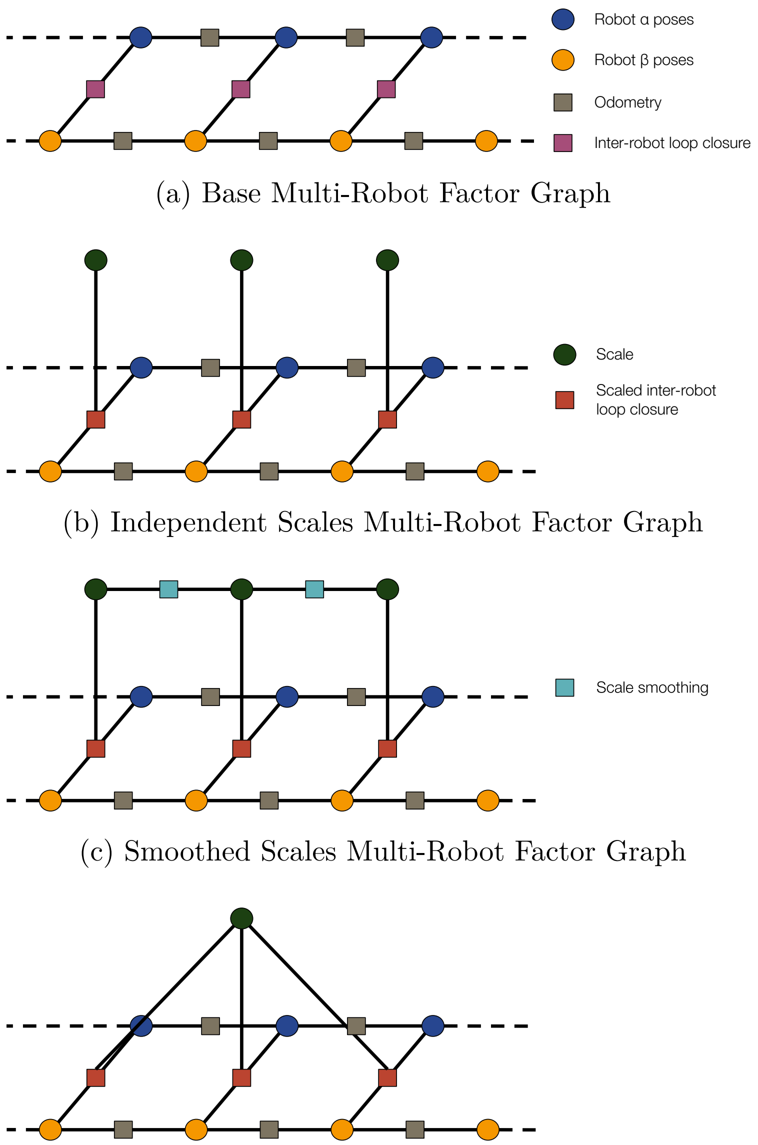
We present the problem structure as a sequence of factor graphs, each incrementally refining the model. We begin with the base multi-robot optimization problem, followed by the introduction of scale estimation, and proceed with approaches to model the relationships between scale variables.

Base Multi-Robot Factor Graph The base optimization problem, illustrated in Fig. 8.6a consists primarily of two types of factors: odometry factors, which link consecutive keyframe poses, and loop closure factors, which link non-consecutive keyframes.

Odometry cost functions for each robot α are defined as follows:

$$\phi_{\alpha,i}^{\alpha,i-1} = \left\| T_{\alpha,i-1}^{-1} \cdot T_{\alpha,i} - \bar{T}_{\alpha,i}^{\alpha,i-1} \right\|^2_{\Omega_{\text{odom}}} \tag{8.7}$$

where $\bar{T}_{\alpha,i}^{\alpha,i-1}$ is the relative pose measurement, and Ω_{odom} is the corresponding information matrix.



(a) Base Multi-Robot Factor Graph

(b) Independent Scales Multi-Robot Factor Graph

(c) Smoothed Scales Multi-Robot Factor Graph

(d) Single Scale Multi-Robot Factor Graph

Figure 8.6 Illustrations of the proposed multi-robot factor graph formulations for pose graph optimization with or without scale estimation. In the figures, circles represent variable nodes (e.g., poses or scales), while squares represent factor nodes, which define relationships between the variables. Formulation (a) is basic pose graph optimization. Formulation (b) introduces separate variables for scale optimization. Formulation (c) add smoothing factors to model the correlation between scale variables. Formulation (d) assumes that all loop closures share the same scale. In our experiments, we compare the proposed formulation and conclude that (b) and (c) offer the most accurate solutions.

Inter-robot loop closure factors are defined as detailed in Section 8.4.2.

Thus, for the multi-robot optimization problem, we minimize the sum of all odometry and inter-robot loop closures cost functions:

$$\boldsymbol{T}^* = \underset{\boldsymbol{T}}{\operatorname{argmin}} \sum_{\alpha \in \mathcal{R}} \sum_{i \in (1:n_\alpha)} \phi_{\alpha,i}^{\alpha,i-1} + \sum_{(\alpha,i),(\beta,j) \in L_{\mathcal{R},\mathcal{R}'}} \phi_{\beta,j}^{\alpha,i} \tag{8.8}$$

where $\mathcal{R}$ denotes the set of all robots, n_α represents the number of keyframes for robot α, $L_{\mathcal{R},\mathcal{R}'}$ is the set of loop closures linking different robots, and $\boldsymbol{T}$ is the set of all poses forming the robots trajectories. The set of robots $\mathcal{R}$ in Eq. (8.8) can involve more than two robots; in fact, we perform the optimization with all neighboring robots, detecting and incorporating inter-robot loop closures into the optimization problem for each pair. Also, we do not consider intra-robot loop closures $\phi_{\alpha,k}^{\alpha,i}$, as we aim to isolate the effects of inter-robot loop closures, but they could straightforwardly be integrated into the formulation as an additional cost function.

Independent Scales Multi-Robot Factor Graph The challenge with the relative poses measured with MASt3R is that their scaling is often incorrect or very imprecise. Thus, to properly integrate these measurements into the optimization problem, we must either determine the scale in advance using other sensors or, as we propose here, treat the loop closure scale as a variable to optimize during pose graph optimization.

Naïvely, we could leverage the correctly scaled odometry measurements (e.g., from VIO) between the two odometry poses to estimate the loop closure scale. This approach would not require additional communication or processing since we already used the three required frames to compute the feature correspondences ratio (see Fig. 8.5). It assumes that the same scaling factor applies to both the odometry and the loop closure:

$$\bar{t}_{\beta,j}^{\alpha,i} = \frac{\left\| \bar{t}_{\alpha,i}^{\alpha,i-1} \right\|}{\left\| t_{\alpha,i}^{\alpha,i-1} \right\|} \cdot t_{\beta,j}^{\alpha,i} \tag{8.9}$$

where $t_{\alpha,i}^{\alpha,i-1}$ and $t_{\beta,j}^{\alpha,i}$ are the relative translations output by MASt3R, and $\bar{t}_{\alpha,i}^{\alpha,i-1}$ is the known translation from odometry. While we cannot guarantee that the loop closure and odometry share the same scaling factors, this often provides a reasonable initial guess, as discussed in Section 8.5.2.

To achieve more accurate solutions, we propose to explicitly optimize the loop closures' scale.

For this, we draw inspiration from [307], which introduced scaling factors for pedestrian trajectories estimated via IMU dead-reckoning and refined using sporadic UWB distance measurements. While this prior work focused on rescaling odometry estimates, we instead apply scaling to the non-consecutive relative pose measurements.

As a first step, we decompose the measured relative pose $\bar{T}^{\alpha,i}_{\beta,j} \in \mathrm{SE}(3)$, from MASt3R, associated with a loop closure as follows:

$$\bar{R}^{\alpha,i}_{\beta,j} \in \mathrm{SO}(3), \;\; \bar{t}^{\alpha,i}_{\beta,j} \in \mathbb{R}^3, \;\; s^{\alpha,i}_{\beta,j} \in \mathbb{R} \tag{8.10}$$

where $\bar{R}^{\alpha,i}_{\beta,j}$ is the measured relative rotation matrix, $\bar{t}^{\alpha,i}_{\beta,j}$ is the measured translation vector between the two poses, and $s^{\alpha,i}_{\beta,j}$ adjusts the magnitude of the translation vector to the correct scale. Together, the scaled relative pose $\hat{T}^{\alpha,i}_{\beta,j}$ is defined as:

$$\hat{T}^{\alpha,i}_{\beta,j} = \begin{bmatrix} \bar{R}^{\alpha,i}_{\beta,j} & s^{\alpha,i}_{\beta,j} \cdot \bar{t}^{\alpha,i}_{\beta,j} \\ 0 & 1 \end{bmatrix} \tag{8.11}$$

We then define a new loop closure cost function that incorporates the scale value:

$$\hat{\phi}^{\alpha,i}_{\beta,j} = \left\| T^{-1}_{\alpha,i} \cdot T_{\beta,j} - \hat{T}^{\alpha,i}_{\beta,j} \right\|^2_{\Omega^{\alpha,i}_{\beta,j}(p^{\alpha,i}_{\beta,j})} \tag{8.12}$$

To optimize this novel factor, we need to provide the corresponding derivatives. Specifically, since we use the efficient GTSAM [4] solver as our factor graph optimization framework, we provide the analytical measurement Jacobian matrices for efficient computation, which are evaluated in the tangent space at the current estimate during optimization:

$$\boldsymbol{H}_{T_{\alpha,i}} = -\mathrm{Adj}(\mathrm{inv}(T^{-1}_{\alpha,i} \cdot T_{\beta,j})) \tag{8.13}$$

$$\boldsymbol{H}_{T_{\beta,j}} = \boldsymbol{I} \tag{8.14}$$

$$\boldsymbol{H}_s = \begin{bmatrix} 0 & 0 & 0 & -\bar{t}^{\alpha,i}_{\beta,j} \end{bmatrix}^\top \tag{8.15}$$

Similar to the base *BetweenFactor* in GTSAM [4], the Jacobian $\boldsymbol{H}_{T_{\alpha,i}}$ is the adjoint matrix of the inverse relative pose transformation, while $\boldsymbol{H}_{T_{\beta,j}}$ is simply the identity matrix $\boldsymbol{I}$. Specific to our new cost function, the partial derivative of the residual in tangent space with respect to the scale factor $s^{\alpha,i}_{\beta,j}$, i.e., $\boldsymbol{H}_s$, is the negative of the measured translation $\bar{t}^{\alpha,i}_{\beta,j}$.

Using $\hat{\phi}^{\alpha,i}_{\beta,j}$ instead of $\phi^{\alpha,i}_{\beta,j}$ in Eq. (8.8) we derive the optimization problem illustrated in Fig. 8.6b, where the scale of each loop closure is treated independently.

Smoothed Scales Multi-Robot Factor Graph We also introduce a third formulation, which includes an additional factor that links scale factors from related loop closures to ensure they remain similar. We cluster the loop closures that link relative poses fewer than ten keyframes apart. These clusters typically occur when the robot revisits a specific area for an extended period, resulting in multiple loop closures being detected within a single large overlap between the maps. Previous work have explored identifying such clusters to detect outliers among loop closure measurements [139, 253]. In our case, the intuition behind linking them is that, since MASt3R is data-driven, relative poses inferred from a similar image domain should exhibit similar scaling factors. We illustrate these scale smoothing factors in Fig. 8.6c.

The cost function to link the scale values, along with its corresponding Jacobians, is defined as follows:

$$\phi_{s_{i,j}} = \|s_j - s_i\|^2_{\Omega_s}, \quad \boldsymbol{H}_{s_i} = -1, \quad \boldsymbol{H}_{s_j} = 1. \tag{8.16}$$

where the scale residual $\phi_{s_{i,j}}$ is defined as the difference between scale estimates s_i and s_j from two different loop closures, ensuring consistent scale across the loop closure cluster. The corresponding Jacobians, $\boldsymbol{H}s_i$ and $\boldsymbol{H}s_j$, are the partial derivatives of $\phi_{s_{i,j}}$ with respect to s_i and s_j, respectively.

For completeness, in our experimental analysis we also consider the scenario where all loop closures share the same scale value, as illustrated in Fig. 8.6d.

Factor Graph Optimization

Following the approach of [284], we propose performing factor graph optimization in a decentralized manner onboard a dynamically elected robot when the robots meet. The resulting estimates are then shared back with neighboring robots. We compare this approach with distributed pose graph optimization in Section 8.5.

8.5 Experiments

We conducted our experiments on multiple dataset sequences using a robot onboard computer NVIDIA Jetson AGX Xavier (32GB). Our approach, MOLD-SLAM, was benchmarked against the Swarm-SLAM framework [284] in two configurations: stereo and LiDAR. In the stereo configuration, place recognition was performed using CosPlace [273], and pose registration was carried out with PnP and RANSAC [269]. In the LiDAR configuration, place recognition utilized ScanContext [274], and pose registration was done using TEASER++ [286]. Swarm-SLAM serves as a strong baseline since it utilizes 3D sensing, whereas our method relies solely on monocular images for loop closure.

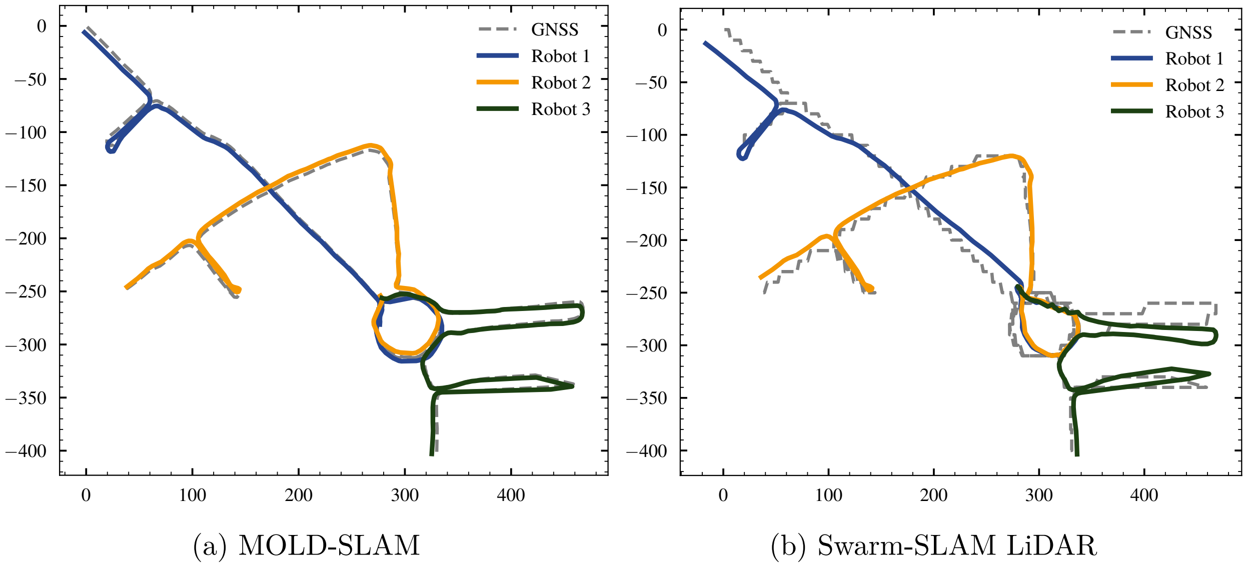
For a thorough evaluation, we also configured Swarm-SLAM with several back-end optimizers: the fast Levenberg-Marquardt solver (LM) [4], the robust Graduated Non-Convexity optimizer (GNC) [140], which can detect and reject outliers among the loop closures, and Riemannian Block Coordinate Descent (RBCD) [63], a distributed solver that incorporates GNC for outlier rejection. Notably, Swarm-SLAM with RBCD in stereo mode closely resembles Kimera-Multi [63], except for place recognition—Kimera-Multi uses DBoW2 [71], while Swarm-SLAM employs the more recent CosPlace [273].

For MOLD-SLAM, we tested three different factor graph formulations: base multi-robot pose graph (Fig. 8.6a), with independent scale values (**IS**) (Fig. 8.6b), and with smoothed scale values (**SS**) (Fig. 8.6c). Unless otherwise specified, the base formulation was used. Since the base formulation does not optimize scale, it relies only on the scale initialization technique presented in Eq. (8.9).

We first evaluated the techniques on the S3E dataset [232], which features three robots navigating large, dynamic indoor and outdoor environments. We selected the most challenging sequences with minimal overlap and differing viewpoints. Additionally, we tested on the GrAco dataset [233], which involves six robots in a large outdoor environment. An ablation study was performed using the GrAco dataset. For evaluation, we compared all our results against GNSS data (for outdoor environments) and motion capture data (for indoor environments), using the *evo* package [298].

8.5.1 Full System Performance

In Fig. 8.7, we illustrate MOLD-SLAM's performance on the challenging S3E Dormitory sequence, demonstrating a close alignment with the ground truth compared to the Swarm-SLAM baseline. The visualizations are supported by the detailed results presented in Table 8.1, where we report the *Average Translation Error* (ATE), the number of loop closures



(a) MOLD-SLAM

(b) Swarm-SLAM LiDAR

Figure 8.7 S3E Dormitory trajectory estimates with MOLD-SLAM versus Swarm-SLAM in LiDAR configuration. Our novel approach leads to significant improvements in accuracy.

N, and the computation time required by the factor graph solver for each technique on four sequences.

The best performance was achieved by MOLD-SLAM using the smoothed scales formulation (**SS**) optimized with the **LM** solver. This approach generated orders of magnitude more loop closures, significantly enhancing the accuracy of the resulting solution. To identify successful matches, we applied a correspondences ratio threshold R_{thr} of 0.3. While some outliers might have passed through, their effects were mitigated by the ratio-based confidence mechanism (see Eq. (8.5)), without needing a more computationally expensive robust solver like GNC. The ability of our approach to generate more loop closures and directly address scale ambiguity—without relying on approximate scale estimates or costly outlier rejection mechanisms—significantly enhances performance.

While MOLD-SLAM with smoothed scales required more computation time in some cases compared to the independent scales version (e.g., for the Dormitory sequence), the improved accuracy indicates that loop closure scales within clusters may be correlated, making the trade-off worthwhile. Notably, on the S3E Dormitory sequence, the stereo Swarm-SLAM configuration was unable to merge all three trajectories due to the challenges of detecting and computing loop closures from opposite viewpoints using traditional stereo matching techniques.

In Fig. 8.8, we demonstrate again MOLD-SLAM's superior performance, this time on the

Table 8.1 Average translation error (ATE), number of loop closures (N) and optimization time on S3E sequences.

		Campus			Teaching			Square			Dormitory		
		ATE (m)	N	Time (s)	ATE (m)	N	Time (s)	ATE (m)	N	Time (s)	ATE (m)	N	Time (s)
MOLD-SLAM	GNC	19.25 ± 6.81	1122	27.45	2.40 ± 0.76	2274	41.58	9.53 ± 3.87	226	9.25	46.00 ± 30.36	175	34.79
	IS-LM	5.50 ± 2.16	1122	2.46	1.87 ± 0.65	2273	3.46	5.64 ± 2.46	226	1.41	3.84 ± 1.35	175	1.50
	SS-LM	**5.41 ± 2.13**	1122	1.82	**1.84 ± 0.67**	2273	4.74	**5.07 ± 2.56**	226	1.63	**2.41 ± 0.99**	175	11.05
Swarm-SLAM	RBCD	9.41 ± 7.59	29	286.63	3.59 ± 1.56	110	143.31	113.58 ± 80.99	6	120.33	7.41 ± 5.32	226	147.80
	GNC	10.50 ± 9.47	29	21.38	3.69 ± 1.52	110	13.97	164.83 ± 99.39	6	36.45	10.18 ± 5.47	226	26.14
LiDAR	LM	10.21 ± 8.97	29	1.23	3.85 ± 1.63	110	0.99	165.94 ± 98.98	6	1.73	8.34 ± 4.82	226	4.02
Swarm-SLAM	RBCD	6.79 ± 7.08	16	744.65	2.23 ± 0.94	27	840.53	42.97 ± 25.27	7	2.46		✗	
	GNC	6.56 ± 7.07	16	48.64	2.25 ± 1.15	27	23.85	9.73 ± 6.67	7	15.34		✗	
Stereo	LM	6.56 ± 7.07	16	2.90	2.20 ± 1.05	27	3.91	33.37 ± 22.98	7	1.27		✗	

GrAco ground sequence with six robots. Our method surpassed the LiDAR-based Swarm-SLAM baseline, using GNC [140], which struggled with noisy loop closures. Similar to the S3E Dormitory sequence, the stereo-based Swarm-SLAM failed to compute sufficient loop closures to fuse all trajectories on GrAco.

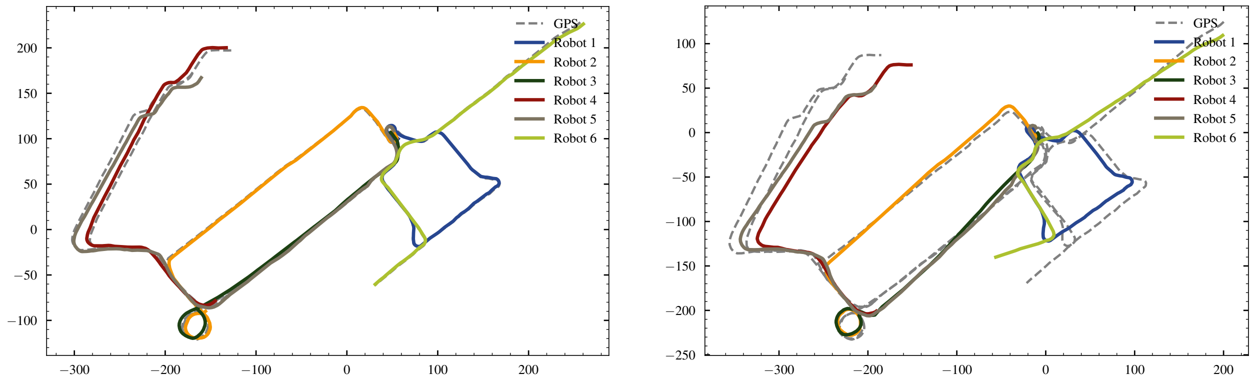
To further analyze our proposed approach, the following subsections present an ablation study we conducted to evaluate the effectiveness and impact of the various novel components.

8.5.2 Relative Pose Estimation Accuracy and Robustness

For our first set of ablation experiments, we used the GrAco dataset [233], which involves six robots, and extracted 88,096 image pairs that are less than 10 meters apart according to GNSS data. We then applied our registration pipeline to each pair to obtain the relative poses.

The first issue we investigated was determining the validity of the matches. Due to perceptual aliasing, place recognition sometimes fails, incorrectly identifying different locations as the same. This challenge is exacerbated by DUSt3R [237], and subsequently MASt3R [238], which are now capable of matching images from almost any viewpoint, making it harder to verify which image matches correspond to valid relative poses and should be included as inter-robot loop closures in the pose graph. In Fig. 8.9, we evaluate the precision and recall of five different metrics. A match is considered an outlier if its translation error compared to the ground truth exceeds 2 meters, with the measurements scaled using ground truth data.

Initially, we confirmed that relying solely on CosPlace similarity, used for place recognition, is insufficient to distinguish registration inliers from outliers. To address this, we examined the average confidence produced by MASt3R and computed the confidence ratio between the loop



(a) MOLD-SLAM

(b) Swarm-SLAM (LiDAR)

Figure 8.8 Trajectory estimates of MOLD-SLAM on the GrAco 6-robot dataset compared to the LiDAR-based Swarm-SLAM solution. MOLD-SLAM demonstrates a significant improvement in localization accuracy, highlighting the effectiveness of our approach in multi-robot C-SLAM scenarios.

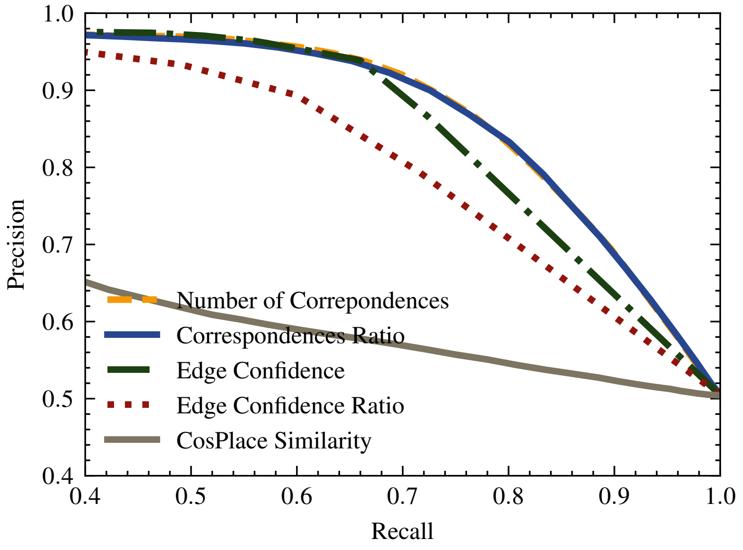


Figure 8.9 Precision-Recall curves for various inlier/outlier detection techniques. The best-performing approaches leverage the correspondences generated by the MASt3R model, using the ratio of the number of correspondences between two loop frames versus two associated odometry frames. The ratio-based approach stands out for its performance, being both unitless and easy to tune, making it particularly effective for identifying inliers in inter-robot loop closures.

match and the odometry match. We then compared these metrics with the number of feature correspondences and the correspondences ratio, as illustrated in Fig. 8.5. The results show that both the number of correspondences and the correspondences ratio provided strong performance, with the correspondences ratio being preferable due to its unitless nature, making it easier to tune compared to the number of correspondences, which is affected by factors such as image size and field of view.

Using the loop closure inliers from our set of image pairs, we compared different scale initialization techniques, as shown in Fig. 8.10. We evaluated ground truth scaling, the direct approach (which uses the raw output of MASt3R), and scaling based on odometry estimates (see Eq. (8.9)). Additionally, we compared these results with relative poses obtained using SuperPoint+SuperGlue and OpenCV (which implements Nister's method [308] with ORB [301] features and descriptors) to assess whether our scale initialization method would also benefit those approaches.

The results in Fig. 8.10 demonstrate that our approach significantly reduces translation error for data-driven MASt3R, whereas the same improvement was not observed with the other methods. This may be explained by MASt3R's ability to output 3D point maps that are scale-consistent within a given image domain or environment. However, the mean translation error remains above 2 meters, which is high for accurate pose graph optimization. Therefore, further refinement of the scale during pose graph optimization is necessary.

8.5.3 Multi-Robot Scale and Pose Graph Optimization

In the following experiments, we evaluate the full pose graph solution involving all six robots from the GrAco sequence. Based on the previously mentioned feature correspondences ratio, we mapped it to a confidence metric (see Eq. (8.5)), which we used to weight the inter-robot loop closures during pose graph optimization.

Fig. 8.11 compares translation errors based on different confidence estimation methods for inter-robot loop closures and optimization techniques during map optimization. Our proposed confidence estimation, derived from the feature correspondences ratio, significantly outperforms the commonly used uniform confidence estimates, which are often adopted due to the difficulty of accurately estimating confidence [62,283]. The figure illustrates that, even with the non-robust Levenberg-Marquardt optimization [4], our method produces accurate results, whereas uniform confidence estimates lead to failure. Moreover, our approach achieves results comparable to the computationally expensive Graduated Non-Convexity solver [140].

Next, in Fig. 8.12, we compare different correspondences ratios threshold, used for outlier

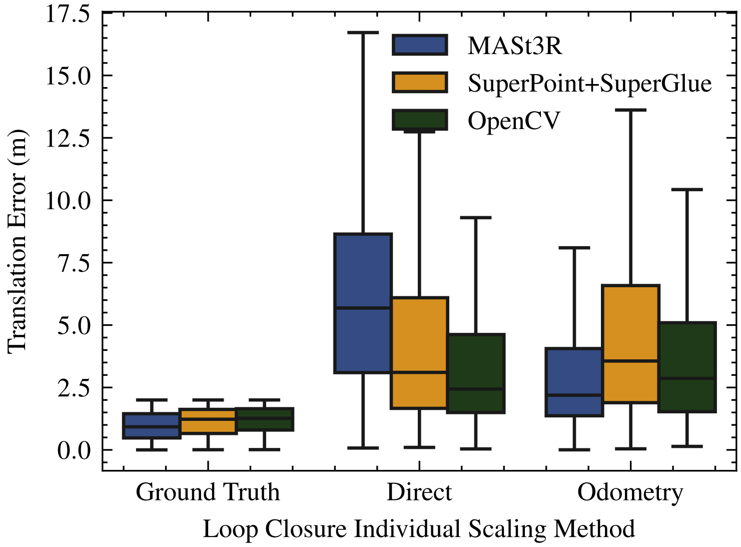


Figure 8.10 Comparison of translation errors for different image matching methods (MASt3R, SuperPoint+SuperGlue, OpenCV) combined with various individual loop closure scaling techniques: direct output scale, ground truth scale, or from one odometry estimate. The results indicate that while it does not achieve perfect accuracy, the odometry-based scaling works best with MASt3R, possibly due to MASt3R's higher scale consistency when visually similar images are used as inputs. In contrast, the odometry-based scaling method has little to no effects for other image matching techniques.

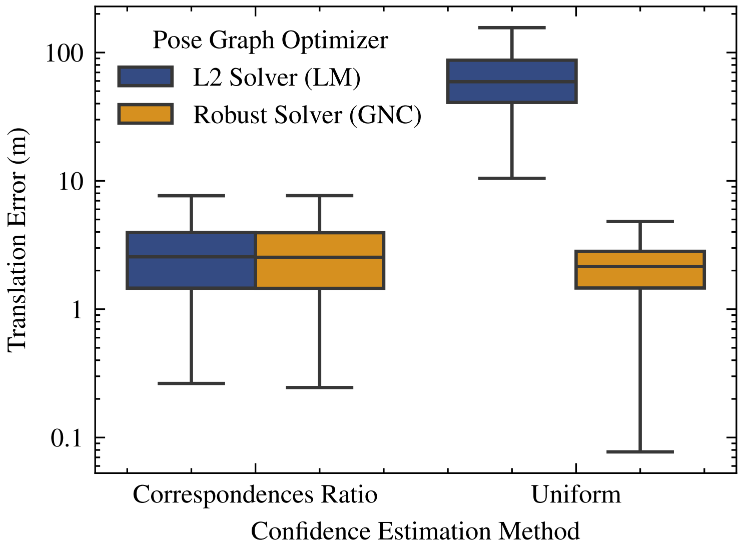


Figure 8.11 Comparison of translation errors with respect to different confidence estimation methods used for inter-robot loop closures and optimization techniques for map fusion on the GrAco 6-robot dataset. Our proposed confidence estimation based on the feature correspondences ratio significantly outperforms uniform confidence estimates, which are commonly used in other approaches due to the challenges in estimating confidence. The figure shows that, with our method, even the non-robust Levenberg-Marquardt optimization yields accurate solutions, whereas it completely fails with uniform confidence estimates. Our results are comparable to those obtained using the computationally expensive robust solver, Graduated Non-Convexity.

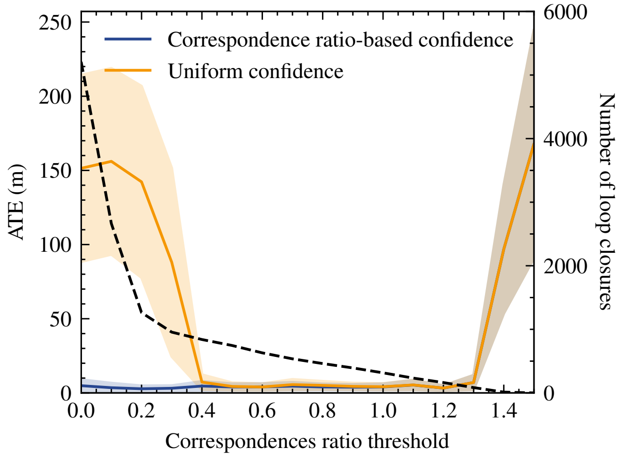


Figure 8.12 Comparison of the correspondences ratio threshold against the Average Translation Error (ATE) of the pose graph optimization solution on the GrAco 6-robot dataset, using either uniform confidence or correspondences ratio-based confidence. The dashed line indicates the number of loop closures obtained at different threshold values. A higher ratio indicates a more conservative solution, resulting in fewer loop closures. Notably, with our ratio-based confidence estimation, the ATE remains low even with less conservative thresholds, demonstrating that our confidence estimation method makes the ratio parameter easy to tune, as the effects of outliers are effectively minimized.

rejection, with the Average Translation Error (ATE), using the LM solver with either uniform confidence or correspondences ratio-based confidence. The dashed line represents the number of loop closures at various threshold levels. A higher ratio results in a more conservative solution, yielding fewer loop closures. Notably, with our ratio-based confidence estimation, the ATE remains low even with less conservative thresholds, showing that this method simplifies parameter tuning by effectively minimizing the influence of outliers without relying on a robust solver. However, if the ratio threshold is set too conservatively, insufficient inter-robot loop closures are included, leading to decreased accuracy or failure to merge the maps.

In the following experiments, presented in Tables 8.2 and 8.3, we analyze the effects of different factor graph formulations, illustrated in Fig. 8.6, and the impact of the odometry backbone. We tested two odometry backbones: VINS-Mono [300], making the system fully monocular, and LiDAR-based LIO [293], to demonstrate our approach's effectiveness when the odometry has near-perfect scale. While monocular odometry yielded reasonable results, inaccuracies in scale at various parts of the trajectory—particularly at the start, where the online scale estimation had not yet converged—led to reduced performance. Since the GrAco dataset lacks VIO calibration sequences, all robot trajectories were affected by these initial scale inaccuracies. In real-world scenarios, proper calibration remains critical for effective

Table 8.2 Comparison of different pose graph optimization formulations with a VINS-Mono odometry backbone.

	GT Scale			Direct Scale			Odometry Scale		
	ATE (m)	Onb.(s)	Srv.(s)	ATE (m)	Onb.(s)	Srv.(s)	ATE (m)	Onb.(s)	Srv.(s)
LM	4.76 ± 4.50	1.60	0.33	39.93 ± 44.43	3.63	0.91	39.59 ± 45.68	3.27	0.79
GNC	4.89 ± 4.51	31.09	10.28	11.98 ± 5.21	32.93	10.49	5.56 ± 4.07	35.86	11.78
OneS-LM	12.35 ± 13.50	1.55	0.34	38.55 ± 44.71	3.99	1.10	40.35 ± 44.51	3.77	0.92
OneS-GNC	12.42 ± 13.53	32.07	10.64	4.91 ± 3.31	34.54	11.86	10.14 ± 5.13	37.17	12.12
IS-LM	6.99 ± 6.47	3.27	0.79	7.11 ± 6.81	3.24	0.78	7.89 ± 7.01	3.49	0.87
IS-GNC	6.34 ± 6.51	104.21	29.65	6.40 ± 6.82	96.54	27.38	7.86 ± 7.01	112.94	31.98
SS-LM	7.43 ± 6.93	3.89	0.94	8.23 ± 7.34	3.50	0.82	7.42 ± 7.06	3.95	0.94
SS-GNC	7.45 ± 7.06	166.24	45.99	91.50 ± 59.82	296.27	85.31	58.94 ± 62.61	305.20	85.38

Table 8.3 Comparison of different pose graph optimization formulations with a LIO odometry backbone.

	GT Scale			Direct Scale			Odometry Scale		
	ATE (m)	Onb.(s)	Srv.(s)	ATE (m)	Onb.(s)	Srv.(s)	ATE (m)	Onb.(s)	Srv.(s)
LM	2.75 ± 1.36	1.44	0.29	10.36 ± 3.65	1.44	0.30	6.90 ± 3.95	1.51	0.30
GNC	2.73 ± 1.36	29.61	9.58	9.25 ± 2.98	29.12	9.56	2.95 ± 1.32	30.71	9.89
OneS-LM	3.06 ± 1.32	1.46	0.30	4.47 ± 2.57	1.44	0.30	10.15 ± 3.63	1.43	0.31
OneS-GNC	2.97 ± 1.28	29.81	9.74	3.63 ± 1.75	28.95	9.54	6.53 ± 2.21	31.78	10.47
IS-LM	3.45 ± 2.03	3.16	0.74	3.14 ± 1.80	2.83	0.66	3.41 ± 1.96	3.01	0.70
IS-GNC	3.45 ± 2.00	93.70	26.45	3.12 ± 1.77	87.59	24.63	3.40 ± 1.94	94.17	26.31
SS-LM	2.92 ± 1.93	4.06	0.95	3.29 ± 1.94	3.42	0.80	3.11 ± 1.81	3.85	0.84
SS-GNC	2.93 ± 1.90	156.52	41.42	3.23 ± 1.92	139.69	37.32	13.89 ± 11.23	342.36	89.20

VIO deployments. Results from the LIO-based solution demonstrate impressive accuracy, with errors below 4 meters over 3.5 kilometers of trajectories.

For each backbone, we compared several formulations previously introduced, including the single-scale formulation (OneS), as shown in Fig. 8.6d. As expected, this formulation performed poorly. While clustering loop closures with similar scales provides some benefit, it does not apply to all loop closures. For completeness, we also report results for both the non-robust Levenberg-Marquardt (LM) solver and the robust Graduated Non-Convexity (GNC) solver for each of our factor graph formulations.

Additionally, we report the computation time on both our robot onboard computer and a desktop server equipped with an AMD Ryzen 7 3700X CPU and an NVIDIA RTX 3070 GPU. This comparison is interesting given that C-SLAM approaches often offload computational tasks to more powerful servers [61].

We provide each result with different scale initialization techniques, whether using ground truth, directly the raw output from MASt3R, or our odometry-based scaling (see Eq. (8.9)). The results in Tables 8.2 and 8.3 confirm that our formulations with independent or smoothed scales perform the best and do not require the computationally expensive robust optimization of GNC. These results are consistent with those in Fig. 8.11, showing that the use of a robust solver offers limited benefits and, in fact, decreases performance when paired with the smoothed scale formulation. Our optimization takes less than a second on the server and only a few seconds on the onboard computer, compared to minutes when using GNC.

We also observed that while our odometry-based scaling initialization sometimes improved results, it could also lead to worse performance. This suggests that our assumption—that MASt3R yields similar relative pose scales for odometry images and loop closure images—may not always hold, and altering initial estimates in local solvers can have a significant impact. Therefore, using the raw output directly and relying solely on scale optimization may offer equally good results.

Overall, these findings demonstrate the robustness and flexibility of our approach. Even with initial scale inaccuracies, the system can adapt and optimize effectively without requiring computationally expensive methods. Our independent and smoothed scale formulations consistently provide accurate results while maintaining efficient performance.

8.5.4 Resource Efficiency

In addition to the computational efficiency provided by our scale-aware optimization compared to robust methods, our approach offers further efficiency advantages. Specifically, we split the registration into two stages where keyframes are encoded locally on each of the two robots involved in the loop closure, and decoding is done on only one of the robots, as outlined in Fig. 8.4. This way, a keyframe part of multiple loop closures only needs to be encoded once—particularly useful when managing a large number of loop closure candidates. In Fig. 8.13, we show the computation gain on the GrAco dataset against a naïve baseline where encoding of both images is done for all candidates. We can see that MASt3R inference is quite costly on a robot onboard computer (in blue) compared to a server with a larger GPU (in orange). Thus, further gains could be achieved through more advanced load-sharing strategies between robots and servers, or across robots with varying computational capacities.

Another resource efficiency gain comes from reducing the number of keyframes to process and store in memory. However, reducing keyframes typically leads to decreased accuracy as the mapping becomes more coarse and fewer loop closures can be detected. In Fig. 8.14, we assess how increasing the distance between keyframes—and thus proportionally reducing their number—affects the ATE of the solution. We also compared this with loop closures computed using SuperPoint+SuperGlue and OpenCV. While other methods show a rapid performance decline as the keyframe distance increases, our approach maintains low ATE values even with distances of up to 4 meters between keyframes. This demonstrates its

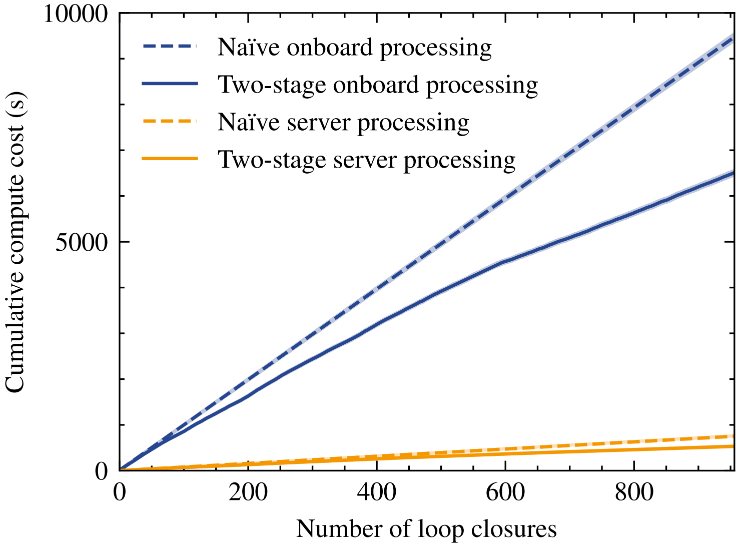


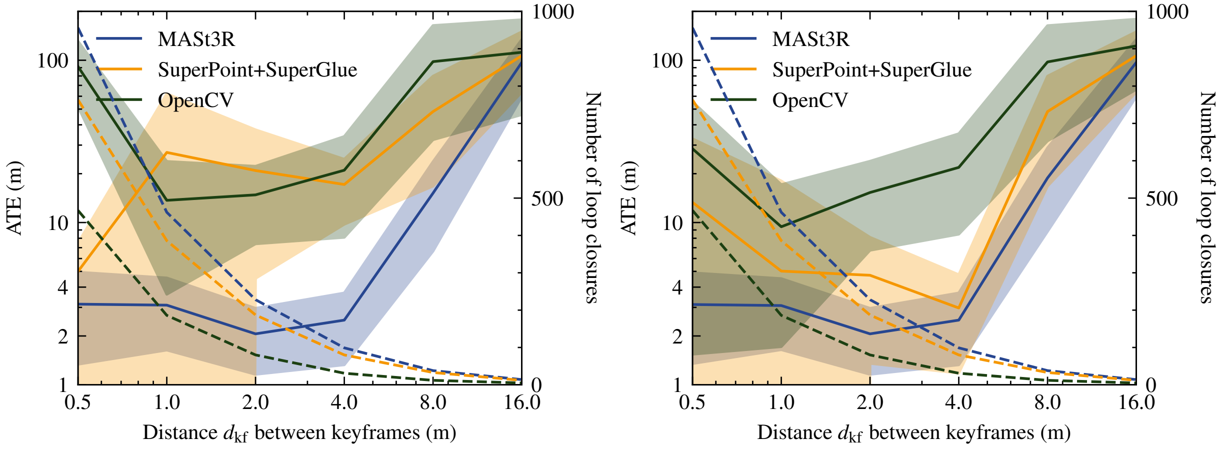
Figure 8.13 Computational savings of the two-stage (split) processing of the MASt3R encoder and decoder in C-SLAM. Over time, the two-stage approach reduces computational load by avoiding re-encoding images for which loop closures have already been computed. This saving is particularly significant for onboard computers, where MASt3R processing is much more computationally demanding compared to processing on a desktop server.

ability to support sparser and more memory-efficient C-SLAM solutions without sacrificing accuracy. Although SuperPoint+SuperGlue exhibits some resistance to keyframe sparsity when paired with a computationally intensive robust solver, our method achieves comparable, or better, results with both solvers. The efficiency gains of using sparser keyframes extend beyond memory savings, as sparse maps can also be compared—via place recognition and registration—using less communication bandwidth and in less time. Therefore, achieving high-accuracy sparse maps could represent a significant advantage for deploying C-SLAM on resource-constrained platforms, such as small robots or consumer electronics.

8.6 Conclusion

In this paper, we introduced MOLD-SLAM, a novel decentralized collaborative SLAM approach that leverages 3D foundation models to address the challenge of limited overlap in multi-robot mapping. By incorporating monocular image-based pose estimation and scale optimization, MOLD-SLAM effectively improves inter-robot loop closures in scenarios where traditional methods struggle due to differing robot viewpoints and trajectories.

Our experimental results, and ablation study, demonstrate that MOLD-SLAM outperforms state-of-the-art approaches, particularly in terms of accuracy. Furthermore, we showed that



(a) With L2 Solver (Levenberg-Marquardt)

(b) With Robust Solver (GNC)

Figure 8.14 Comparison of the Average Translation Error (ATE) with respect to the distance between keyframes for three image matching techniques: MASt3R, SuperPoint+SuperGlue, and OpenCV. The respective numbers of loop closures are represented by dashed lines. A longer distance between keyframes reduces the number of poses and corresponding images that need to be stored in memory. While other methods show a rapid decline in performance as the keyframe distance increases, our approach maintains low ATE values up to 4 meters between keyframes, demonstrating its ability to support sparser and more memory-efficient C-SLAM solutions without sacrificing accuracy. Results are shown both with and without a robust solver; while SuperPoint+SuperGlue exhibits greater resistance to sparsity when paired with a computationally intensive robust solver, our method achieves similar results with both solvers.

when accurate odometry with the correct scale is available, it is possible to easily integrate up-to-scale loop closures in multi-robot pose graph optimization, making our approach both simple and resource-efficient.

Looking ahead, there is considerable potential for tighter integration between 3D foundation models and C-SLAM systems, not only for improving measurement accuracy but also for advancing the overall representation of the explored environments.

CHAPTER 9 ARTICLE 6 : PEOPLEX: PEDESTRIAN OPPORTUNISTIC POSITIONING LEVERAGING IMU, UWB, BLE AND WIFI

Preface: This paper introduces PEOPLEx, a real-time pedestrian localization framework using IMU-based navigation as its core, opportunistically integrating UWB, BLE, and WiFi signals without prior environmental knowledge. It improves indoor positioning accuracy through adaptive scaling and loop closure techniques, validated on commercial smartphones. This work was presented during the IEEE International Conference on Communications.

Contributions: This project was conducted during an internship at the Samsung AI Center in Montreal, where I led the application of SLAM techniques to positioning using consumer-grade hardware. While the initial objective was set by the team, I devised most of the solution. Bobak Hamed Baghi made a significant contribution by setting up the UWB hardware and the data collection application, while Sachini Herath assisted with the deployment of the RoNIN odometry. All of this was accomplished under the guidance and frequent feedback of the senior coauthors.

Full Citation: Pierre-Yves Lajoie; Bobak Hamed Baghi; Sachini Herath; Francois Hogan; Xue Liu; Gregory Dudek, "PEOPLEx: PEdestrian Opportunistic Positioning LEveraging IMU, UWB, BLE and WiFi," *ICC 2024 - IEEE International Conference on Communications*, 2024.

Submission date: November 3rd 2023

Publication date: August 20th 2024

DOI: 10.1109/ICC51166.2024.10622566

9.1 Abstract

This paper advances the field of pedestrian localization by introducing a unifying framework for opportunistic positioning based on nonlinear factor graph optimization. While many existing approaches assume constant availability of one or multiple sensing signals, our methodology employs IMU-based pedestrian inertial navigation as the backbone for sensor fusion, opportunistically integrating Ultra-Wideband (UWB), Bluetooth Low Energy (BLE), and WiFi signals when they are available in the environment. The proposed PEOPLEx framework is designed to incorporate sensing data as it becomes available, operating without any prior knowledge about the environment (e.g. anchor locations, radio frequency maps, etc.). Our contributions are twofold: 1) we introduce an opportunistic multi-sensor and real-time

pedestrian positioning framework fusing the available sensor measurements; 2) we develop novel factors for adaptive scaling and coarse loop closures, significantly improving the precision of indoor positioning. Experimental validation confirms that our approach achieves accurate localization estimates in real indoor scenarios using commercial smartphones.

9.2 Introduction

With an ever-growing number of interconnected devices and systems, effective and accurate indoor positioning has become increasingly critical. The real-time location of the user is a valuable data point that heavily impacts the *state* of the environment in both household and industrial settings, and can influence the interpretation of other data streams as well as any potential action to be undertaken by the IoT system. Indoor Positioning Systems (IPS) hold the potential to transform the operations and experiences of pedestrians, underpinning applications such as navigation assistance, location-based services, safety enhancement, and augmented reality experiences.

Historically, building reliable indoor positioning solutions often necessitates prior knowledge of the environment, such as the precise locations of Radio Frequency (RF) anchors [309], floor plans [310], and more [311]. However, acquiring and utilizing such comprehensive information is not always feasible, nor desirable, due to issues related to availability, operational cost, and privacy concerns. Recognizing these challenges, our research focuses on smartphone-based localization strategies that use exclusively the sensing modalities available in the environment, are allowed by the users, and do not require prior knowledge about the environment. Smartphones, which are ubiquitous, offer a rich set of built-in sensors and radios (e.g., IMUs, Wi-Fi, Bluetooth Low Energy (BLE), Ultra-Wideband (UWB)) that can be utilized for localization. Unlike cameras, also available in smartphones, those sensors do not require active supervision and raise less privacy concerns.

The method proposed in this paper, PEOPLEx, leverages pedestrian inertial navigation to fuse multiple sensor modalities, capitalizing on whatever sensor data is readily available at any given time. To achieve this, we introduce a methodology that uses nonlinear factor graph optimization as a unifying framework to integrate information from the diverse sensor modalities available on smartphones. At the center of this approach is the inertial motion estimation, which produces noisy and up-to-scale pedestrian trajectories, effectively acting as the ‘glue’ that ties together the input from other sensors like Wi-Fi, BLE, and UWB. Utilizing these trajectories, our nonlinear factor graph approach enables simultaneous optimization of both UWB anchor positions and user locations, eliminating the necessity for predefined initial knowledge of the environment. PEOPLEx integrates two distinct forms of IMU-based pedestrian inertial navigation, both agnostic to sensor-placement on the body, thereby offering a comprehensive analysis of the approach's effectiveness and adaptability. In essence, our novel formulation allows us to simultaneously estimate key motion parameters, such as step length or scaling, localize the user, and construct a coarse map of RF sources.

In summary, we present the following contributions:

- An opportunistic framework that leverages pedestrian inertial navigation to fuse sensor measurements when available, without requiring any initial knowledge about the environment such as anchor locations or radio frequency maps;
- Novel factor formulations for adaptive scaling and coarse loop closing to robustly integrate data from RF sensors, enhancing the precision of indoor positioning;

We validate our contributions in extensive experiments with smartphone IMUs and two exterosceptive sensing mechanism: WiFi and BLE fingerprinting, and UWB ranging.

9.3 Background and Related Work

9.3.1 Pedestrian Inertial Navigation

Pedestrian inertial navigation is a central component of indoor positioning research due to its independence from external infrastructures and its relatively low computational requirements. It leverages the inertial measurement units (IMU), available in many consumer devices to track pedestrian motion, estimating position based on parameters such as step length, velocity and heading direction [312]. Recent studies have used learning-based approaches towards refining the accuracy and reliability, exploiting the regularity of pedestrian motion to infer the overall user motion from the high frequency and high noise accelerometer and gyroscope data [313].

Despite these advancements, those approaches still face notable challenges, most significantly the issue of accumulated error over time. This drift error arises because the estimation process incrementally incorporates motion estimates from one step to another without correcting the noise from previous steps. Therefore, small inaccuracies compound over time leading to significant positioning errors [312].

9.3.2 RF-based Indoor Positioning Systems

Radio Frequency (RF)-based systems have become a popular solution for indoor positioning due to their capability to provide relatively accurate positioning based on existing infrastructure in indoor environments [311]. There exists a variety of RF technologies utilized for indoor positioning, including Wi-Fi, Bluetooth Low Energy (BLE), and Ultra-Wideband (UWB), each with their unique strengths and limitations.

Wi-Fi based positioning systems are among the most common and cost effective due to the widespread availability of Wi-Fi infrastructure (e.g. routers, consumer devices, etc.). They typically employ techniques such as Received Signal Strength Indicator (RSSI) fingerprinting to estimate devices positions. However, unless extensive and costly acquisition of fingerprint maps is undertaken, Wi-Fi-based systems tend to suffer from signal interference, often leading to suboptimal localization accuracy [314]. With the increasing number of connected devices, BLE fingerprint-based systems have emerged as a suitable alternative to Wi-Fi. BLE beacons can provide positioning data with a small energy footprint, making them particularly suitable for battery-powered devices. However, they can be affected by signal instability and require a dense beacon deployment for optimal performance [315].

Ultra-Wideband (UWB) positioning systems stand out due to their high precision and resilience to multipath effects. UWB systems often utilize Two-Way Ranging (TWR), since it does not necessitate clock synchronization between the devices involved, thus reducing system complexity while making it easier to deploy [316]. In conventional indoor positioning using (single antenna) UWB, at least three to four anchors with known locations are typically needed for 2D or 3D estimation. The system's precision is also sensitive to the placement of these sensors, deteriorating when the anchors are sub-optimally located [317]. Unlike most UWB-based techniques, our approach does not assume any prior knowledge of UWB anchors placement, and can improve positioning estimates using a single UWB transceiver.

9.3.3 Sensor Fusion Approaches

The importance of sensor fusion, the process of combining data from multiple sensory sources, has gained substantial recognition in the domain of indoor positioning. By coalescing information from various sensors, these methods strive to harness the advantages and counterbalance the limitations of individual sensors, thereby enhancing the accuracy and robustness of positioning systems [318].

Numerous methodologies have been employed to implement sensor fusion. One prevalent approach is the integration of data from inertial measurement units (IMUs) with RF signals such as UWB. Conventionally, filtering techniques like Kalman filters and Particle filters have been the standard methods for merging the data from these diverse sources. While these techniques have demonstrated improvement in positioning accuracy compared to single sensor-based systems [319], they exhibit certain limitations. These methods often necessitate intricate sensor and motion modeling and typically lack the capability to reassess and refine past estimates.

Smoothing-based formulations based on factor graph address those limitations. Factor graphs provide a flexible framework for modeling the complex dependencies between various sensor measurements and position estimates, and are able to refine past estimates as new measurements are acquired [320]. Representing variables and constraints as nodes and edges, respectively, this approach leverages the problem's sparse structure for efficient computation, which is crucial for large-scale, long-term tasks like indoor positioning [321].

9.3.4 Opportunistic Approaches

Opportunistic approaches [88, 322, 323] have recently gained attention, providing potential solutions for scenarios where traditional localization systems may not be suitable. Unlike conventional methods, opportunistic approaches do not depend on the constant availability of specific signals or information sources but instead utilize whatever data is readily available. In other words, opportunistic methods will use localization infrastructures such as UWB or GPS if available, but will still provide reasonable localization estimates without.

While various works in the field strive to enhance inertial pedestrian navigation by fusing it with data from supplementary sensors, they often fall short of providing a comprehensive solution. For example, Tian et al. [322] utilize a particle filter to combine inertial navigation and UWB range measurements from a single anchor. However, this approach demands an initial anchor position estimation and lacks the capability for continuous refinement, contrasting with our factor graph methodology. Liu et al. [88], although employing a factor graph approach similar to ours, make a simplifying assumption of a constant step length, thereby introducing drift and scale inaccuracies, and does not integrate range measurements.

Jao et al. [323] implement an Extended Kalman Filter to merge data from foot-mounted IMUs and UWB sensors. While foot-mounted IMUs offer enhanced tracking performance, they lack the practicality of widely available smartphone sensors. In a similar vein, Chen et al. [324] and Lu et al. [325] offer methodologies that necessitate prior knowledge or surveying of the environment. Chen et al. [324] fuses pedestrian inertial navigation with BLE fingerprinting and trilateration, while Lu et al. [325] proposes a data-driven IMU and WiFi indoor localization system. Our approach, in contrast, does not assume any prior environmental information and provides a real-time, multi-modal sensor fusion framework capable of online parameter estimation, such as step length and anchor positions.

9.4 PEOPLEx Framework

In this work, we leverage nonlinear factor graph optimization in conjunction with the sensing of environmental radio-frequency signals (i.e. WiFi, BLE, and UWB) to correct the scale and drift of IMU-based localization techniques. In absence of sufficient radio-frequency signals, our technique performs inertial navigation alone which can be relied upon for short periods of time. Thus, our approach is opportunistic in nature, enabling the use of an available RF sensing modality when possible, yet consistently delivering a solution —albeit potentially less accurate— even when it's not available.

The real power of our methodology comes from the inclusion of custom nonlinear factors tailored specifically for pedestrian indoor positioning. These factors encode domain-specific knowledge such as unique sensor characteristics and key pedestrian parameters directly in the optimization problem. To efficiently optimize on the Special Euclidean group SE(3) (i.e. rotation and translation of the user), nonlinear factor graph optimization linearizes the problem at the current estimate, approximating it within a tangent space. The linearized problem is then solved, and the solution is mapped back to the original SE(3) space for variable updates [321]. The iterative process of linearization and updates is repeated until it eventually converges to a solution. Thus, to implement our custom factors, we must specify appropriate error functions and their associated Jacobians in the SE(3) tangent space, which are described in the following subsections.

9.4.1 Adaptive Scale for Pedestrian Motion

In our study, we employ two types of inertial navigation techniques: step counting-based Pedestrian Dead Reckoning (PDR) and Robust Neural Inertial Navigation (RoNIN [313]). Both techniques use accelerometers and gyroscopes to estimate relative motion. However, they are inherently prone to accumulating errors over time, i.e. drift.

On one hand, PDR counts the step of the user by detecting spikes in acceleration [326], and estimates the heading direction of each step (i.e. yaw) by combining accelerometer and gyroscope data. In closely related prior work [88], PDR operates under the assumption of a constant step length provided as an input parameter. This approach is not realistic, as step length can vary significantly between individuals and can also fluctuate over time even within a single trajectory. Solutions for step length estimation, such as foot-mounted IMUs [323], are often not practical as they require specialized hardware. Therefore, in our framework, we extend traditional Pedestrian Dead Reckoning (PDR) by incorporating a custom nonlinear factor to jointly estimate the step length and the agent's pose during the localization process.

On the other hand, RoNIN [313] integrates IMU data overtime using a neural network architecture and computes a stable and accurate relative velocity. Yet, the scale of the velocity estimates is intrinsically tied to the data used for training the network. Thus if the user is smaller, taller or has a different gait from the original data collectors, the resulting RoNIN trajectory estimates need to be scaled by a constant parameter analog to the step length in PDR.

Step Counting-based Pedestrian Dead Reckoning

We define the agent's pose before and after a step as $\boldsymbol{T}_0$ and $\boldsymbol{T}_1$ respectively. Both poses belong to SE(3). The measured relative rotation matrix $\bar{\boldsymbol{R}}$ from the IMU belongs to SO(3) and, finally, s and $\boldsymbol{u}$ are the scaling variable corresponding to the step length and the unit vector corresponding to the walking direction, respectively.

$$\boldsymbol{T}_0, \boldsymbol{T}_1 \in \mathrm{SE}(3), \quad \bar{\boldsymbol{R}}_{0,1} \in \mathrm{SO}(3), \quad s \in \mathbb{R} \tag{9.1}$$

$$\bar{\boldsymbol{t}}_{0,1} = s \cdot \boldsymbol{u} \tag{9.2}$$

To build the measured relative pose $\bar{\boldsymbol{T}}_{\text{step}}$ for a step, we utilize the measured rotation $\bar{\boldsymbol{R}}_{0,1}$ and the translational component $\bar{\boldsymbol{t}}_{0,1}$ which is dependent on the scale variable s. We also define the relative pose $\boldsymbol{T}_{0,1}$ between our estimates and the pedestrian motion error term ρ_{motion}:

$$\bar{\boldsymbol{T}}_{\text{step}} = \begin{bmatrix} \bar{\boldsymbol{R}}_{0,1} & \bar{\boldsymbol{t}}_{0,1} \\ 0 & 1 \end{bmatrix}. \tag{9.3}$$

$$\boldsymbol{T}_{0,1} = \boldsymbol{T}_0^{-1} \cdot \boldsymbol{T}_1. \tag{9.4}$$

$$\rho_{\text{motion}} = \bar{\boldsymbol{T}}_{\text{step}} - \boldsymbol{T}_{0,1}. \tag{9.5}$$

The Jacobian matrices are critical for the optimization process, as they tell the optimizer how a small change in each variable would affect the error. In our case, we need the Jacobians $\boldsymbol{H}_{\boldsymbol{T}_0}^{\rho_{\text{motion}}}$, $\boldsymbol{H}_{\boldsymbol{T}_1}^{\rho_{\text{motion}}}$, and $\boldsymbol{H}_{s}^{\rho_{\text{motion}}}$ of the error function ρ_{motion} with respect to the pose before the step, the pose after the step, and the scaling variable s respectively.

$$\boldsymbol{H}_{\boldsymbol{T}_0}^{\rho_{\text{motion}}} = -\mathrm{Adj}(\boldsymbol{T}_{0,1}^{-1}), \quad \boldsymbol{H}_{\boldsymbol{T}_1}^{\rho_{\text{motion}}} = \boldsymbol{I}, \tag{9.6}$$

$$\boldsymbol{H}_{s}^{\rho_{\text{motion}}} = \Big[0, 0, 0, -\boldsymbol{u}\Big]^\top. \tag{9.7}$$

The Jacobian $\boldsymbol{H}_{\boldsymbol{T}_0}^{\rho_{\text{motion}}}$ is the negative adjoint representation of the relative pose $\boldsymbol{T}_{0,1}^{-1}$. It essentially captures how a small change in the initial SE(3) pose influences the error term in tangent space $\mathfrak{se}(3)$. Since $\boldsymbol{T}_1$ directly contributes to $\boldsymbol{T}_{0,1}$, the Jacobian $\boldsymbol{H}_{\boldsymbol{T}_1}^{\rho_{\text{motion}}}$ is simply the identity matrix. Any change in $\boldsymbol{T}_1$ would directly translate to the same magnitude of change in the error term. The Jacobian $\boldsymbol{H}_{s}^{\rho_{\text{motion}}}$ reflects how a change in the scale s would negatively impact the translational component of the error term along the walking direction $\boldsymbol{u}$.

We also add a scale smoothing factor ρ_{scale} to the factor graph. This factor is used to penalize large changes between the scales s_i and s_j of consecutive steps. The scale smoothing factor is defined as follows:

$$\rho_{\text{scale}} = s_j - s_i, \quad \boldsymbol{H}_{s_i}^{\rho_{\text{scale}}} = -1, \quad \boldsymbol{H}_{s_j}^{\rho_{\text{scale}}} = 1. \tag{9.8}$$

Robust Neural Inertial Navigation (RoNIN)

In contrast to step counting-based PDR, RoNIN employs a neural network to integrate IMU data and produce robust relative velocity estimates $\bar{\boldsymbol{v}}$ over a time interval Δ_{time}. RoNIN's initial velocity scale is influenced by the ground truth data used during the network training. To accommodate variations in user size, gait, and other parameters, we introduce a scaling variable s analog to the one we use for the step counting-based approach. Note that scale smoothing is also used for RoNIN.

$$\bar{\boldsymbol{t}}_{0,1} = s \cdot \bar{\boldsymbol{v}} \cdot \Delta_{\text{time}} \tag{9.9}$$

The relative pose $\bar{\boldsymbol{T}}_{\text{step}}$ in the RoNIN-based method is assembled using the scaled translation $\bar{\boldsymbol{t}}_{0,1}$ in a slightly different manner compared to the PDR. In this case, the relative pose matrix represents only translation in x and y directions, $\bar{x}_{0,1}$ and $\bar{y}_{0,1}$, without accounting for rotation. The relative pose error function stays the same.

$$\bar{\boldsymbol{T}}_{\text{step}} = \begin{bmatrix} & \boldsymbol{I} & & \bar{x}_{0,1} \\ & & & \bar{y}_{0,1} \\ & & & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \rho_{\text{motion}} = \bar{\boldsymbol{T}}_{\text{step}} - \boldsymbol{T}_{0,1}. \tag{9.10}$$

$$\boldsymbol{H}_{s}^{\rho_{\text{motion}}} = \Big[0, 0, 0, -\bar{v}_x \cdot \Delta_{\text{time}}, -\bar{v}_y \cdot \Delta_{\text{time}}, 0\Big]^{\top}. \tag{9.11}$$

The Jacobians $\boldsymbol{H}_{\boldsymbol{T}_0}^{\rho_{\text{motion}}}$ and $\boldsymbol{H}_{\boldsymbol{T}_1}^{\rho_{\text{motion}}}$ are the same as in the step counting method. However, the Jacobian $\boldsymbol{H}_{s}^{\rho_{\text{motion}}}$ reflects a different sensitivity of the error term to changes in the scaling variable s. It takes into account the effect on both x and y components of the velocity $\bar{\boldsymbol{v}}$ over the time interval Δ_{time}.

9.4.2 Drift Correction via Coarse Relocalization

In our solution, we also incorporate other measurement types such as BLE (Bluetooth Low Energy) and WiFi fingerprinting, which are widely used in indoor localization [311]. We perform fingerprinting online, which means that we collect fingerprints as we move through the environment and use them to correct the drift in our inertial navigation estimates by producing loop closures between poses (i.e., steps) sharing two highly similar fingerprints. Frequently used to reduce localization drift SLAM, loop closures are constraints added between non-consecutive steps that are recognized to be in the same location [283]. Specifically, using consumer smartphones, we periodically perform BLE and WiFi scans to collect Received Signal Strength Indicators (RSSI), in dBm (decibel-milliwatts), from devices present in the environment. The RSSI values from a scan i are stored into a fingerprint vector f_i for which each entry correspond to a unique device. We compare fingerprint vectors from different scans (e.g. f_i and f_j) using a cosine similarity. If the similarity is sufficiently high, we introduce a loop closure constraint. To avoid false positive, we only consider scans with at least ten RSSI values. Due to limitations in fingerprinting accuracy, the discretization of steps, and the limited BLE/WiFi scan frequencies, the association of step-to-scan data is inherently imperfect. Thus, with smartphones usually constrained by low scan rates, BLE and Wi-Fi fingerprinting solutions lack reliability for precise localization. While the standard approach in the literature typically employs simple proximity constraints, which can be enhanced by learning a mapping from similarity to distance as seen in [88], this approach has the drawback of being highly environment and device-specific.

To address these challenges in a way that is agnostic to the environment, we introduce a coarse loop closure method. This method defines a circular region within which loop closures are costless, while outside of this region the cost increases in function of the distance. Essentially, it encourages two estimated locations to remain close up to a certain distance. The distance threshold is conservatively set to account for the inherent inaccuracies of the fingerprinting system.

$$\boldsymbol{T}_i, \boldsymbol{T}_j \in \mathrm{SE}(3), \quad \boldsymbol{q} = \boldsymbol{R}_i^{\top} \cdot (\boldsymbol{t}_j - \boldsymbol{t}_i). \tag{9.12}$$

Here, $\boldsymbol{T}_i$ and $\boldsymbol{T}_j$ are transformation matrices representing two non-consecutive poses linked by a fingerprint loop closure, and $\boldsymbol{q}$ is the relative translation vector between them in pose $\boldsymbol{T}_i$ reference frame.

$$\rho_{\text{loop}} = \begin{cases} \|\boldsymbol{q}\| - r & \text{if } \|\boldsymbol{q}\| > r, \\ 0 & \text{otherwise.} \end{cases} \tag{9.13}$$

The loop closure cost, ρ_{loop}, is defined as follows: if $\|\boldsymbol{q}\|$ (the Euclidean norm of $\boldsymbol{q}$) is greater than a predefined trust radius r, the cost is the difference between $\|\boldsymbol{q}\|$ and r. This encourages loop closures when poses are outside the radius, otherwise, if $\|\boldsymbol{q}\|$ is less than or equal to r, the cost is set to 0, indicating that the poses are within the loop closure radius. We define the Jacobians as follows:

$$\boldsymbol{q}_{\text{norm}} = \left[\frac{\boldsymbol{q}_x}{\|\boldsymbol{q}\|}, \frac{\boldsymbol{q}_y}{\|\boldsymbol{q}\|}, \frac{\boldsymbol{q}_z}{\|\boldsymbol{q}\|}\right], \tag{9.14}$$

$$\boldsymbol{H}_{\boldsymbol{T}_i}^{\rho_{\text{loop}}} = \boldsymbol{q}_{\text{norm}} \cdot \begin{bmatrix} 0 & -\boldsymbol{q}_z & \boldsymbol{q}_y & -1 & 0 & 0 \\ \boldsymbol{q}_z & 0 & -\boldsymbol{q}_x & 0 & -1 & 0 \\ -\boldsymbol{q}_y & \boldsymbol{q}_x & 0 & 0 & 0 & -1 \end{bmatrix}, \tag{9.15}$$

$$\boldsymbol{H}_{\boldsymbol{T}_j}^{\rho_{\text{loop}}} = \left[0, 0, 0, \boldsymbol{q}_{\text{norm}} \cdot \boldsymbol{R}_i^{\top} \cdot \boldsymbol{R}_j\right]. \tag{9.16}$$

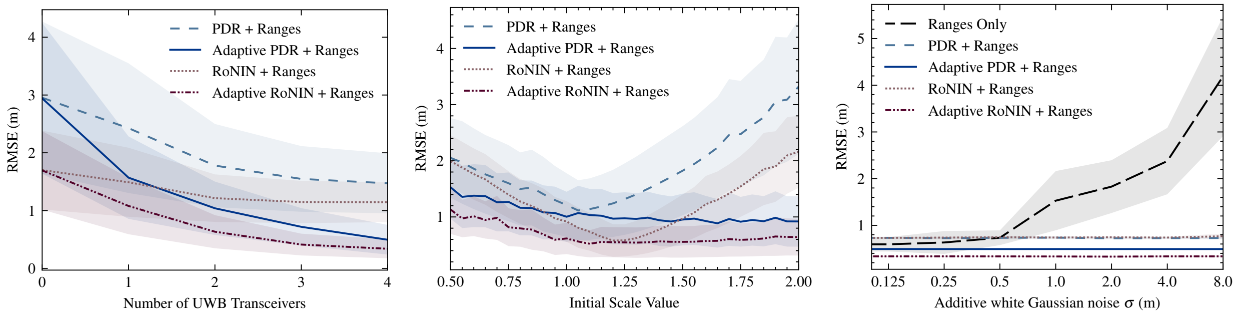
where $\boldsymbol{q}_{\text{norm}}$ is the normalized relative translation vector, while $\boldsymbol{H}_{\boldsymbol{T}_i}^{\rho_{\text{loop}}}$ and $\boldsymbol{H}_{\boldsymbol{T}_j}^{\rho_{\text{loop}}}$ represent the sensitivity of the loop closure cost to changes in the poses i and j. In the cases where ρ_{loop} is equal to 0 (i.e., the poses are within the loop closure radius), we set the Jacobians to zero.

9.4.3 Opportunistic Positioning Solution

Our method results in a solution that is both robust and uniquely suited for the complex challenges of indoor pedestrian localization. We seek to recover the optimal sequence of 3D transformations $\boldsymbol{T}_0^*, \boldsymbol{T}_1^*, \ldots, \boldsymbol{T}_n^*$ by minimizing the weighted sum of squared residuals corresponding to the measurements. These transformations represent the poses of the pedestrian at different time steps. Specifically, we solve:

$$
\begin{aligned}
&\boldsymbol{T}_0^* \dots \boldsymbol{T}_n^* && (9.17)\\
=&\operatorname*{argmin}_{\boldsymbol{T}_0\dots\boldsymbol{T}_n} \sum_{i\in\text{steps}} \|\rho_{\text{motion}}(\boldsymbol{T}_i, \boldsymbol{T}_{i+1})\|^2_{\boldsymbol{\Omega}_{\text{motion}}} + \|\rho_{\text{scale}}(s_i, s_{i+1})\|^2_{\boldsymbol{\Omega}_{\text{scale}}} && (9.18)\\
&\quad + \sum_{(bi,bj)\in\text{BLE loops}} \|\rho_{\text{loop}}(\boldsymbol{T}_{bi}, \boldsymbol{T}_{bj})\|^2_{\boldsymbol{\Omega}_{\text{BLE}}} && (9.19)\\
&\quad + \sum_{(wi,wj)\in\text{WiFi loops}} \|\rho_{\text{loop}}(\boldsymbol{T}_{wi}, \boldsymbol{T}_{wj})\|^2_{\boldsymbol{\Omega}_{\text{WiFi}}} && (9.20)\\
&\quad + \sum_{(u,a,d)\in\text{UWB ranges}} \|\rho_{\text{range}}(\boldsymbol{T}_u, \boldsymbol{t}_a; d)\|^2_{\boldsymbol{W}_{\text{UWB}}}. && (9.21)
\end{aligned}
$$

The weight matrices $\boldsymbol{\Omega}$, represented as $\boldsymbol{\Omega}_{\text{motion}}$, $\boldsymbol{\Omega}_{\text{scale}}$, $\boldsymbol{\Omega}_{\text{BLE}}$, and $\boldsymbol{\Omega}_{\text{WiFi}}$, are the information matrices (i.e., inverse of the covariance) of the corresponding measurements. The error term ρ_{range} is the standard range measurement error term between a pose from the trajectory $\boldsymbol{T}_u$ and the 3D position of an anchor $\boldsymbol{t}_a$, with d as the measured range [321]. The weight matrix $\boldsymbol{W}_{\text{UWB}}$ incorporates a Cauchy robust loss function based on the residual value [327] to account for outliers in the collected ranges, particularly those arising from non-line-of-sight measurements. To this formulation is added a pose prior fixing the first pose to the origin of the world frame. Data from smartphones and UWB sensors are continuously collected and transmitted to a central server for real-time processing. The optimization process is performed using the Levenberg-Marquardt-based iSAM2 iterative algorithm [320]



(a) Effect of the number of UWB anchors.

(b) Effect of the initial scale value.

(c) Effect of initial UWB anchors estimates

Figure 9.1 Comparison between fixed and adaptive scaling approaches for inertial navigation and UWB ranging solutions.

9.5 Experiments

For the experiments, we used Samsung S22 phones carried by the pedestrian subjects. Through the Android API, we collect orientation data, accelerometer data for step counting, and both accelerometer and gyroscope data for RoNIN-based inertial navigation. Decawave DW1001 Ultra Wideband (UWB) transceivers operating at 10 Hz were employed for ranging, with one module on each phone and four additional modules serving as anchors randomly placed within the environment. Bluetooth Low Energy (BLE) scans and WiFi scans were carried out at frequencies of 1 Hz and 3 Hz, respectively. We use Google's ARCore for ground truth which is an accurate Visual-Inertial SLAM software available on Android.

9.5.1 Adaptive Scaling using Range Measurements

Our evaluation explores various parameters, including the number of UWB anchors, the initial scale value, and the initial position estimates of the UWB anchors. We perform evaluation with the four variants of our method: PDR, RoNIN, Adaptive PDR, and Adaptive RoNIN. The first two variants use a fixed scale value, while the latter two estimate the scale online.

First, we assessed the system's accuracy with respect to the number of UWB anchors employed. We tested the accuracy with all combinations of 0 to 4 anchors, 0 corresponding to the pedestrian motion estimates alone. Fig. 9.1a shows the mean and standard deviation results for the four variants of our method. The results clearly demonstrate that accuracy increases with the number of anchors. Importantly, even a single anchor significantly enhances accuracy, showcasing the effectiveness of our system in environments with limited anchor availability. This is especially true for the adaptive versions of our pedestrian motion (i.e. Adaptive PDR and Adaptive RoNIN), as they can correct the scale of the trajectory even with a single anchor. Notably, RoNIN consistently outperforms PDR due to its reduced susceptibility to drift. Next, in Fig. 9.1b, we examined the influence of the initial scale value s (referenced in Eqs. 9.2 and 9.9) on accuracy. For fixed versions of pedestrian motion (PDR and RoNIN), an erroneous initial scale value can lead to a large error accumulation. Conversely, adaptive versions (Adaptive PDR and Adaptive RoNIN) estimate scale online and correct for poor initializations. Our evaluation also examined the impact of initial position estimates of UWB anchors. Beginning with accurate anchor positioning and introducing increasing levels of white Gaussian noise, Figure 9.1c demonstrates that our system's accuracy remains robust, as it estimates anchor positions as part of the optimization process. We compared our results with a standard range-only approach [311] which necessitates accurate initial anchor positioning to acheive reasonable estimates. This experiments showcases the versatility of our system, which can be used in environments where anchor positions are unknown.

9.5.2 Drift Correction via Coarse Relocalization

A further contribution is the integration of BLE and WiFi-based coarse loop closing into our framework. We conducted experiments to assess the impact of this loop closing technique on the accuracy and robustness of our system.

As detailed in Section 9.4.2, our loop closing method defines a circular region where loop closures incur no cost. Beyond this region, the cost increases as a function of distance. To account for the low scan rates of BLE and WiFi, we set the radius of this region to 2 meters. The experimental results depicted in Fig. 9.2 showcase the efficacy of this approach. To evaluate the practical utility of BLE and WiFi loop closing, we conducted ten trajectories within an indoor environment, excluding UWB ranging. The resulting average accuracy, expressed as RMSE (m) and standard deviations, are summarized in Table 9.1. We compared against the standard proximity-based loop closing, which assumes that two steps poses linked by a

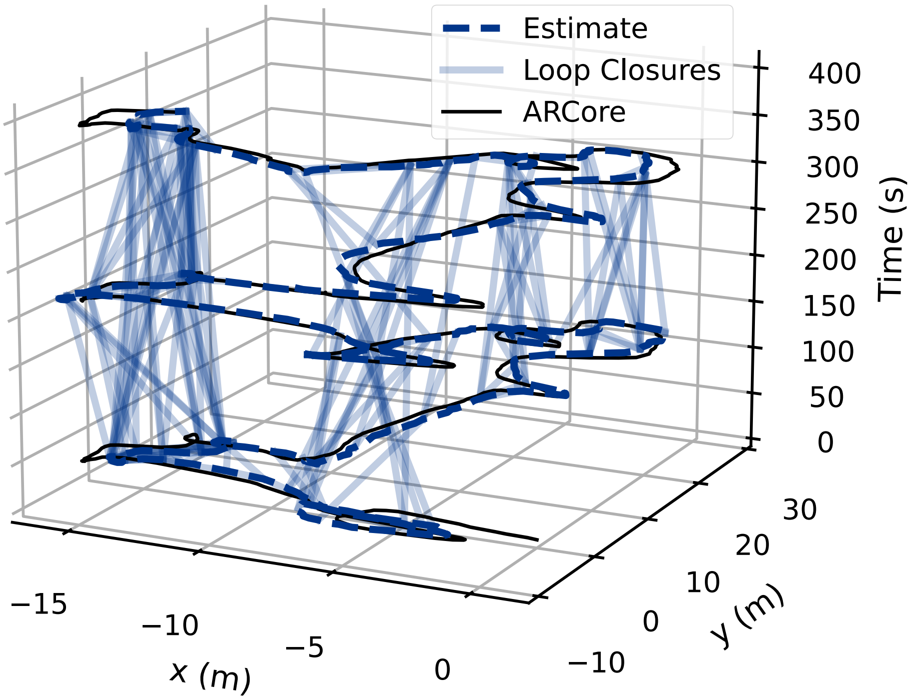


Figure 9.2 Illustration of the effect of BLE and WiFi-based coarse loop closing on localization accuracy. Our approach results in a RMSE of 1.05 ± 0.52, which is a significant improvement over RoNIN alone acheiving a RMSE = 2.88 ± 1.55.

Table 9.1 RMSE (m) achieved with BLE and WiFi loop closing over 10 runs. Comparison between standard proximity loop closing and our approach. Smaller values are better.

		BLE	WiFi	BLE and WiFi
PDR	Proximity	2.39±0.96	3.31±1.61	3.22±1.33
	PEOPLEx	1.46±0.60	1.63±0.66	1.46±0.58
RoNIN	Proximity	2.38±1.06	2.68±1.22	2.85±1.22
	PEOPLEx	1.14±0.49	1.23±0.53	1.10±0.48

loop closure are at the exact same position. Conversely, our coarse loop closing method that is better adapted to the low scan rate allowed on commercial phones, the step discretization, and the inherent inaccuracies of fingerprinting such as multi-path effects.

The results demonstrate a substantial improvement in accuracy when employing our coarse loop closing method in conjunction with imperfect BLE and WiFi data. More importantly, as can be expected, naively performing proximity loop closing degrades the accuracy of the estimates. Therefore, while direct proximity-based approaches, such as the one used in [88], can achieve good performance when the measurement rate is at least as frequent as the user steps, they are vulnerable to large errors when the scanning rate is too low. This motivates the need for our more permissive coarse loop closing mechanism.

9.6 Conclusion

In conclusion, this paper introduces a novel framework for pedestrian localization that capitalizes on an opportunistic multi-sensor approach, leveraging IMU-based inertial navigation as the backbone and integrating UWB, BLE, and WiFi when available to enhance accuracy. The framework requires no prior environmental knowledge, and incorporates novel factors for adaptive scaling and coarse loop closures. Experimental validation using commercial smartphones in real indoor environments demonstrates the effectiveness of our approach. Future work will explore activity-aware localization, on-device estimation, and the integration of additional sensing signals, such as WiFi RTT or CSI, in our framework.

CHAPTER 10 GENERAL DISCUSSION

The preceding chapters have presented contributions for the improvement of accuracy, adaptability, and resource efficiency in collaborative SLAM systems. In this section, we synthesize the findings, discuss the insights gained, and reflect on the challenges and open questions that remain. This discussion is organized according to the three main research axes: Accuracy and Resilience, Resource Efficiency, and Adaptability.

10.1 Accuracy and Resilience

This thesis introduced several techniques to enhance the accuracy and robustness of SLAM and C-SLAM systems, addressing key challenges across diverse scenarios. We tackled incorrect place recognition matches through improved self-supervised calibration, as discussed in Chapter 5, and addressed low overlap between robot trajectories—a common issue in collaborative mapping—through the use of foundation models in MOLD-SLAM (Chapter 8). Although our latest results demonstrate that high-precision localization is achievable even with limited map overlap, several challenges remain. As shown in Chapter 7, the overall system's robustness continues to rely heavily on the quality of loop closure detection and data association, both of which remain vulnerable to noisy environments and spurious measurements.

An important conceptual distinction I wish to highlight is the difference between robustness and resilience in C-SLAM systems. While robustness focuses on minimizing the likelihood of failures, resilience emphasizes the ability to recover from failures with minimal disruption to the system [181]. In this thesis, we have aimed to improve both aspects. While in Chapters 5 and 8 we enhanced robustness by reducing errors in place recognition and registration, in Chapter 6 we introduced improvements in resilience through neighbor management and ad-hoc networking compatibility, ensuring that the system can continue to operate even if one or more robots fail. This capacity to adapt and recover dynamically is essential for real-world deployments where unpredictable failures are inevitable. This need for localization resilience is exemplified by the recent localization failure of NASA's Ingenuity Mars Helicopter [328], which permanently put an end to its mission.

Therefore, I believe that more attention should be given to resilience in future research. Resilience has been relatively overlooked in the localization literature compared to robustness, but it may prove even more crucial for the widespread deployment of multi-robot systems. As robotic platforms become more interconnected and complex, the ability to maintain system functionality in the face of partial failures or environmental challenges will be key to ensuring reliable operation across industrial, planetary, and urban applications [329].

10.2 Resource Efficiency

Managing the balance between communication, memory, and compute resources is critical, especially as the size of robotic teams and the scale of operations grow. In Swarm-SLAM (Chapter 6), we demonstrated that well-informed communication prioritization can significantly reduce the volume of data shared and the associated complexity required to build accurate, globally consistent maps. By selectively transmitting only the most relevant information, Swarm-SLAM facilitates rapid reconstruction of environments, ensuring efficient operation even in ad-hoc robotic networks with constrained or intermittent communication, such as those deployed in search-and-rescue missions or planetary exploration.

In MOLD-SLAM (Chapter 8), we explored the trade-offs between computational complexity and system performance by integrating advanced registration techniques. These techniques, while more computationally demanding, allow for high-precision alignment of maps generated by robots with divergent trajectories and minimal overlap. This high-accuracy registration unlocks significant computational gains during optimization by reducing the need for costly outlier detection and error corrections. Moreover, keyframe sparsification was evaluated to compress map data, leading to substantial savings in computation, communication, and memory resources. These results are promising for deployment on resource-constrained robotic platforms enabling them to participate effectively in collaborative SLAM and multi-agent tasks.

In PEOPLEx (Chapter 9), we extended SLAM principles beyond robotics, demonstrating their applicability to low-power consumer devices like smartphones. By leveraging data from embedded sensors (e.g., accelerometers, gyroscopes, UWB, WiFi, and BLE), PEOPLEx enables accurate pedestrian positioning and tracking in both urban and indoor environments. This innovation unlocks new possibilities for location-based services on mobile devices, such as augmented reality, smart navigation, and personal safety monitoring, all without requiring specialized hardware. As illustrated by recent approaches [216, 220, 221], the C-SLAM community has shown growing interest in leveraging this technology to enable multi-agent augmented reality experiences and improve human-robot interactions.

From the field experiments presented in Chapter 7, we investigated the scalability of our proposed methods, revealing that scalability in SLAM is influenced not only by the number of collaborating robots but also by the size of the map. Although this issue also exists in single-robot SLAM, it becomes more pronounced in C-SLAM due to the communication overhead involved in synchronizing and scaling maps across multiple agents, especially when network latency and bandwidth limitations are taken into account. Our results show that trade-offs between communication, memory, and computation can be adjusted to achieve a balance between accuracy and robustness through key parameter tuning. While this approach may not offer a perfect solution, it represents the best available strategy until major breakthroughs in computational methods or hardware are achieved.

Throughout this thesis, I tried to be as transparent as possible regarding the challenges of expert tuning, a topic underexplored in the literature. While a few studies, such as [292], acknowledge this problem, many omit this discussion or lack the open-source implementations needed to facilitate reproducibility, thereby hindering scientific progress. Given that reproducibility lies at the heart of science, we have published and publicly released our code wherever possible, ensuring that our results—whether in terms of accuracy or robustness—can be independently validated and extended by other researchers and practitioners.

Collectively, our contributions on resource-efficiency demonstrate the current scalability of SLAM-based approaches across diverse scenarios—from ad-hoc robotic networks operating in Mars-analogue environments to consumer smartphones in urban settings. We highlight the importance of communication-aware design, computational trade-offs, and resource-efficient mapping techniques in achieving state-of-the-art performance. However, despite these advancements, further research is required to develop real-time SLAM solutions for large-scale operations, beyond a few robots, ensuring seamless collaboration across heterogeneous platforms outside controlled laboratory settings.

10.3 Adaptability

Adaptability is essential to ensure that SLAM systems can perform reliably across diverse and unpredictable environments. The self-supervised domain calibration method introduced in Chapter 5 addresses this challenge by providing a solution to adapt place recognition models without manual intervention. This approach helps the system recalibrate itself to new environments, reducing the dependence on domain-specific models.

The root of the problem lies in the fact that most place recognition models are still trained almost exclusively on a single domain—typically datasets captured by test vehicles on city roads [72, 242]. This is due to the availability of large-scale datasets focused on autonomous driving [176, 330, 331]. Consequently, these models struggle to generalize to indoor, underground, or aerial environments where drones or other autonomous systems operate [332]. This domain rigidity limits the utility of SLAM systems in real-world applications.

While our self-supervised calibration approach successfully tunes place recognition models by adjusting based on a preliminary run with loop closures, this process requires an initial, although partial, exploration of the environment. This constraint presents a practical challenge for dynamic or previously unmapped environments, where a priori access to in-domain data is not always feasible.

Looking ahead, the emergence of foundation models—pretrained on vast amounts of multimodal data—offers a promising solution in addition of, or in replacement of, Chapter 5. Large-scale pretraining could enable place recognition models to generalize across multiple domains [223, 224], reducing the reliance on targeted calibration and enhancing cross-domain adaptability. However, while foundation models hold promise, challenges remain in highly degraded environments, such as low-light conditions, occluded spaces, or environments with sensor degradation [332]. The lack of high-quality datasets with labeled data from such environments continues to hinder progress, as degraded sensing scenarios are rarely captured comprehensively in publicly available datasets [234].

To fully unlock the potential of adaptable SLAM systems, the research community must invest in the creation of diverse, high-quality datasets that reflect the complexities and corner cases of real-world sensing conditions. Additionally, there is a need for models trained to adapt more effectively across various domains, from outdoor settings to low-visibility indoor environments. Building more generalizable models, capable of handling environmental variability and sensor limitations, will be essential for the widespread deployment of SLAM technologies across diverse robotic platforms.

10.4 Open Questions

In the light of the contributions presented in this thesis, some open questions remain for future research:

- **Scaling to Larger Teams and Missions:** As we move towards larger-scale robot teams, resource bottlenecks—whether computational, memory-related, or communication based—will become more pronounced [20]. How can SLAM systems balance these constraints while ensuring real-time performance?

- **Integration with Broader Robotic Capabilities:** Future research must explore the integration with navigation, coordination, privacy, and security more comprehensively.

As robots operate in sensitive or collaborative settings, new frameworks will be needed to safeguard data and ensure robust multi-agent interactions with both cooperative and perhaps byzantine robots [6];

- **Map Representations:** The maps used in this thesis are built with explicit representations based on 3D features and geometric measurements. However, recent trends in implicit representations, such as NeRF [333] or Gaussian Splatting [334], offer a new way to represent environments with impressive realistic rendering capabilities. How can these methods be scaled for real-time operations and efficiently integrated into C-SLAM systems?

- **Handling Computational Costs of Learned Models:** As learned models grow more complex, the computational demands of inference often increase. Lightweight model architectures [270] or distributed inference [335] will be necessary to handle these challenges. Additionally, how can future systems leverage cloud infrastructure to offload heavy computations without compromising latency or accuracy?

In summary, this thesis demonstrated the feasibility of decentralized collaborative SLAM in real-world deployments. The proposed methods—ranging from Swarm-SLAM to MOLD-SLAM—provide a strong foundation for future research. However, open challenges remain in the areas of scalability, long-term operation, and adaptability. Addressing these issues will require new algorithmic developments, better hardware integration, and continuous refinement through extensive real-world testing.

CHAPTER 11 CONCLUSION

The research presented in this thesis has made contributions toward improving accuracy, adaptability, and resource efficiency in collaborative SLAM (C-SLAM) systems. Each work package addressed specific challenges along these axes, and together they form a cohesive effort to advance the state of the art in C-SLAM.

Through this journey, I have learned valuable lessons about conducting research, collaborating with other teams, identifying critical research problems, and finding solutions using the scientific method. I have also gained insights into how to evaluate and present research effectively. An essential aspect I came to appreciate was the importance of sharing code and providing detailed documentation to ensure reproducibility. This practice not only facilitates scientific progress and helps others build upon existing solutions, but also provides a foundation for extending these efforts into future projects.

The progress made in this thesis contributes to a modest but significant progress in C-SLAM research. Beyond the immediate outcomes, the tools and frameworks developed here can also benefit other fields where collaborative behaviors among autonomous agents are essential, including multi-robot systems, multi-agent augmented reality, and beyond.

11.1 Insights and Reflections

Throughout this research, I identified critical resource trade-offs required for practical C-SLAM deployments. Although significant improvements were achieved, one of the key takeaways is the need for easier system tuning and generalization. Ideal C-SLAM systems would work out of the box with default parameters across a wide range of scenarios. Without such flexibility, these systems will likely remain confined to controlled laboratory environments, limiting their broader applicability.

My interactions with users and feedback, particularly on Swarm-SLAM, have also highlighted the importance of actionable maps. Maps must not only provide accurate localization and shared situational awareness among robots but also support decision-making and planning tasks. In practice, maps serve various purposes, such as path planning, terrain analysis, and environmental reasoning. However, current approaches—whether volumetric, mesh-based, or relying on implicit representations—may not always suit all applications or match the unique constraints faced by multi-robot teams.

Collaborative SLAM holds tremendous potential for transforming various sectors of society, from autonomous transportation and smart infrastructure to emergency response, environmental monitoring, and space exploration. Solving the remaining challenges in scalability, adaptability, and efficiency is therefore not just a technical goal—it is a critical enabler for the future of robotics and autonomous systems. Ensuring that these technologies are robust, resilient, and adaptable across different environments and devices will help to unlock their full potential and drive meaningful societal impact.

11.2 Future Research Directions

A natural extension of this thesis involves enhancing C-SLAM to better suit downstream tasks like path planning and collaborative decision-making, enabling fully autonomous collaboration in large-scale robot teams or swarms. In particular, the next steps involve scaling C-SLAM to support swarms of robots that can operate autonomously with minimal human intervention. By focusing on real-time adaptability and robustness, these systems can unlock new possibilities for industrial applications, space exploration, and public services. The ultimate vision is to move from aspirational research to practical deployments, ensuring that C-SLAM solutions evolve from promising prototypes to solutions that have tangible impacts on industries and societies.

I hope the body of work presented in this thesis, along with the conclusions drawn, will support future researchers in their endeavors. I aim for it to provide a comprehensive overview of the field, highlight the aspirations for improvement, and inspire new developments. My hope is that it helps future researchers grasp the field's most pressing limitations, avoid common pitfalls, and ultimately push the boundaries of Collaborative Simultaneous Localization and Mapping science toward new frontiers.

REFERENCES

[1] 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, Dec. 2016.

[2] H.-S. Lee, S. Kim, Y.-T. Kim, M. Jeon, W. Seo, D. Yang, C.-K. Lee, S. Moon, N. Kwon, J. Seo, J.-S. Chung, B. Shin, J. Pi, Y. Kim, V. Druzhin, G. Sung, and S. Hong, “AR Glasses: Fatigue-free Optical Engines and Energy-efficient SLAM Sensors,” in *2021 IEEE International Electron Devices Meeting (IEDM)*, Dec. 2021, pp. 35.5.1–35.5.4.

[3] I. A. Putra and P. Prajitno, “Parameter Tuning of G-mapping SLAM (Simultaneous Localization and Mapping) on Mobile Robot with Laser-Range Finder 360° Sensor,” in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2019, pp. 148–153.

[4] F. Dellaert et al., “Georgia Tech Smoothing And Mapping (GTSAM),” http://gtsam.org/.

[5] P.-Y. Lajoie, S. Hu, G. Beltrame, and L. Carlone, “Modeling Perceptual Aliasing in SLAM via Discrete–Continuous Graphical Models,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1232–1239, Apr. 2019.

[6] A. Moroncelli, A. Pacheco, V. Strobel, P.-Y. Lajoie, M. Dorigo, and A. Reina, “Byzantine Fault Detection in Swarm-SLAM Using Blockchain and Geometric Constraints,” in *Swarm Intelligence - ANTS 2024*, H. Hamann, M. Dorigo, L. Pérez Cáceres, A. Reina, J. Kuckling, T. K. Kaiser, M. Soorati, K. Hasselmann, and E. Buss, Eds. Cham: Springer Nature Switzerland, 2024, pp. 42–56.

[7] C. Jennings, D. Murray, and J. J. Little, “Cooperative robot localization with vision-based mapping,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 4, May 1999, pp. 2659–2665 vol.4.

[8] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, “A Probabilistic Approach to Collaborative Multi-Robot Localization,” *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, Jun. 2000.

[9] S. Thrun, “A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots,” *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, May 2001.

[10] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, “Towards multi-vehicle simultaneous localisation and mapping,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, May 2002, pp. 2743–2748 vol.3.

[11] J. W. Fenwick, P. M. Newman, and J. J. Leonard, “Cooperative concurrent mapping and localization,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, May 2002, pp. 1810–1817 vol.2.

[12] L. Paull, M. Seto, and J. J. Leonard, “Decentralized cooperative trajectory estimation for autonomous underwater vehicles,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 184–191.

[13] F. Bonin-Font and A. Burguera, “Towards Multi-Robot Visual Graph-SLAM for Autonomous Marine Vehicles,” *Journal of Marine Science and Engineering*, vol. 8, no. 6, p. 437, Jun. 2020.

[14] A. Rioux, C. Esteves, J. Hayet, and W. Suleiman, “Cooperative SLAM-based object transportation by two humanoid robots in a cluttered environment,” in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov. 2015, pp. 331–337.

[15] Y. Tian, K. Liu, K. Ok, L. Tran, D. Allen, N. Roy, and J. P. How, “Search and rescue under the forest canopy using multiple UAVs,” *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1201–1221, Sep. 2020.

[16] S. Lee, H. Kim, and B. Lee, “An Efficient Rescue System with Online Multi-Agent SLAM Framework,” *Sensors*, vol. 20, no. 1, p. 235, Jan. 2020.

[17] E. Vitug, “Cooperative Autonomous Distributed Robotic Exploration (CADRE),” http://www.nasa.gov/directorates/spacetech/game_changing_development/projects/CADRE, Feb. 2021.

[18] K. Ebadi, M. Palieri, S. Wood, C. Padgett, and A.-a. Agha-mohammadi, “DARE-SLAM: Degeneracy-Aware and Resilient Loop Closing in Perceptually-Degraded Environments,” *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, p. 2, Apr. 2021.

[19] G. Beni, "From swarm intelligence to swarm robotics," in *Proceedings of the 2004 International Conference on Swarm Robotics*, ser. SAB'04. Berlin, Heidelberg: Springer-Verlag, Jul. 2004, pp. 1–9.

[20] M. Kegeleirs, G. Grisetti, and M. Birattari, "Swarm SLAM: Challenges and Perspectives," *Frontiers in Robotics and AI*, vol. 8, 2021.

[21] R. G. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. L. S. Younes, "Coordination for Multi-Robot Exploration and Mapping," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence.* AAAI Press, Jul. 2000, pp. 852–858.

[22] DARPA, "DARPA Subterranean Challenge," https://www.subtchallenge.com/, 2020.

[23] W. Bezouska and D. Barnhart, "Decentralized Cooperative Localization with Relative Pose Estimation for a Spacecraft Swarm," in *2019 IEEE Aerospace Conference*, Mar. 2019, pp. 1–13.

[24] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-Robot Simultaneous Localization and Mapping: A Review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[25] W. Rone and P. Ben-Tzvi, "Mapping, localization and motion planning in mobile multi-robotic systems," *Robotica*, vol. 31, no. 1, pp. 1–23, Jan. 2013.

[26] H. Lee, Seung-Hwan Lee, Tae-Seok Lee, Doo-Jin Kim, and B. Lee, "A survey of map merging techniques for cooperative-SLAM," in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Nov. 2012, pp. 285–287.

[27] J. Kshirsagar, S. Shue, and J. M. Conrad, "A Survey of Implementation of Multi-Robot Simultaneous Localization and Mapping," in *SoutheastCon 2018*, Apr. 2018, pp. 1–7.

[28] R. U. Gupta and J. M. Conrad, "A Survey on Multi-robot Particle Filter SLAM," in *2019 SoutheastCon*, Apr. 2019, pp. 1–5.

[29] D. Zou, P. Tan, and W. Yu, "Collaborative visual SLAM for multiple agents:A brief survey," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 5, pp. 461–482, Oct. 2019.

[30] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision," *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.

[31] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotic | The MIT Press.* The MIT Press, 2005.

[32] D. M. Rosen, K. J. Doherty, A. Terán Espinoza, and J. J. Leonard, “Advances in Inference and Representation for Simultaneous Localization and Mapping,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 215–242, 2021.

[33] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, “A survey of structure from motion,” *Acta Numerica*, vol. 26, pp. 305–364, May 2017.

[34] T. D. Barfoot, *State Estimation for Robotics.* Cambridge: Cambridge University Press, 2017.

[35] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, “A taxonomy for swarm robots,” in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’93)*, vol. 1, Jul. 1993, pp. 441–447 vol.1.

[36] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, “Swarm robotics: A review from the swarm engineering perspective,” *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, Mar. 2013.

[37] T. Cieslewski, S. Choudhary, and D. Scaramuzza, “Data-Efficient Decentralized Visual SLAM,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2466–2473.

[38] K. Y. K. Leung, “Cooperative Localization and Mapping in Sparsely-Communicating Robot Networks,” Ph.D. dissertation, University of Toronto, Toronto, Ontario, Canada, 2012.

[39] L. A. A. Andersson and J. Nygards, “C-SAM: Multi-Robot SLAM using square root information smoothing,” in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 2798–2805.

[40] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, “Multiple relative pose graphs for robust cooperative mapping,” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 3185–3192.

[41] M. T. Lázaro, L. M. Paz, P. Piniés, J. A. Castellanos, and G. Grisetti, “Multi-robot SLAM using condensed measurements,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 1069–1076.

[42] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, “Collaborative monocular SLAM with multiple Micro Aerial Vehicles,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 3962–3970.

[43] P. Schmuck and M. Chli, “Multi-UAV collaborative monocular SLAM,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3863–3870.

[44] ——, “CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams,” *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.

[45] G. Loianno, J. Thomas, and V. Kumar, “Cooperative localization and mapping of MAVs using RGB-D sensors,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4021–4028.

[46] I. Deutsch, M. Liu, and R. Siegwart, “A framework for multi-robot pose graph SLAM,” in *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, Jun. 2016, pp. 567–572.

[47] F. Li, S. Yang, X. Yi, and X. Yang, “CORB-SLAM: A Collaborative Visual SLAM System for Multiple Robots,” in *Collaborative Computing: Networking, Applications and Worksharing*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, I. Romdhani, L. Shu, H. Takahiro, Z. Zhou, T. Gordon, and D. Zeng, Eds. Cham: Springer International Publishing, 2018, pp. 480–490.

[48] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[49] M. Karrer, P. Schmuck, and M. Chli, “CVI-SLAM—Collaborative Visual-Inertial SLAM,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2762–2769, Oct. 2018.

[50] M. Karrer and M. Chli, “Towards Globally Consistent Visual-Inertial Collaborative SLAM,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 3685–3692.

[51] A. C. Jiménez, V. García-Díaz, R. González-Crespo, and S. Bolaños, "Decentralized Online Simultaneous Localization and Mapping for Multi-Agent Systems," *Sensors*, vol. 18, no. 8, p. 2612, Aug. 2018.

[52] T. Bailey, M. Bryson, H. Mu, J. Vial, L. McCalman, and H. Durrant-Whyte, "Decentralised cooperative localisation for heterogeneous teams of mobile robots," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 2859–2865.

[53] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *The International Journal of Robotics Research*, vol. 36, no. 12, pp. 1286–1311, Oct. 2017.

[54] M. Pfingsthorn, B. Slamet, and A. Visser, "A Scalable Hybrid Multi-robot SLAM Method for Highly Detailed Maps," in *RoboCup 2007: Robot Soccer World Cup XI*, ser. Lecture Notes in Computer Science, U. Visser, F. Ribeiro, T. Ohashi, and F. Dellaert, Eds. Berlin, Heidelberg: Springer, 2008, pp. 457–464.

[55] S. Saeedi, L. Paull, M. Trentini, and H. Li, "Multiple robot simultaneous localization and mapping," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 853–858.

[56] G. Bresson, R. Aufrère, and R. Chapuis, "Consistent multi-robot decentralized SLAM with unknown initial positions," in *Proceedings of the 16th International Conference on Information Fusion*, Jul. 2013, pp. 372–379.

[57] S. Saeedi, L. Paull, M. Trentini, and H. Li, "Occupancy grid map merging for multiple robot simultaneous localization and mapping," *International Journal of Robotics and Automation 2015*, vol. 30, no. 6, Jan. 2015.

[58] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. Lo Surdo, T. Molnar, R. Steindl, M. Wildie, I. Sa, N. Kottege, K. Stepanas, E. Hernandez, G. Catt, W. Docherty, B. Tidd, B. Tam, S. Murrell, M. Bessell, L. Hanson, L. Tychsen-Smith, H. Suzuki, L. Overs, F. Kendoul, G. Wagner, D. Palmer, P. Milani, M. O'Brien, S. Jiang, S. Chen, and R. Arkin, "Heterogeneous Ground and Air Platforms, Homogeneous Sensing: Team CSIRO Data61's Approach to the DARPA Subterranean Challenge," *Field Robotics*, vol. 2, no. 1, pp. 595–636, Mar. 2022.

[59] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund, M. F. Ginting, K. Ebadi, M. Anderson, T. Pailevanian, E. Terry, M. Wolf, A. Tagliabue, T. S. Vaquero, M. Palieri, S. Tepsuporn,

Y. Chang, A. Kalantari, F. Chavez, B. Lopez, N. Funabiki, G. Miles, T. Touma, A. Buscicchio, J. Tordesillas, N. Alatur, J. Nash, W. Walsh, S. Jung, H. Lee, C. Kanellakis, J. Mayo, S. Harper, M. Kaufmann, A. Dixit, G. Correa, C. Lee, J. Gao, G. Merewether, J. Maldonado-Contreras, G. Salhotra, M. S. Da Silva, B. Ramtoula, Y. Kubo, S. Fakoorian, A. Hatteland, T. Kim, T. Bartlett, A. Stephens, L. Kim, C. Bergh, E. Heiden, T. Lew, A. Cauligi, T. Heywood, A. Kramer, H. A. Leopold, C. Choi, S. Daftry, O. Toupet, I. Wee, A. Thakur, M. Feras, G. Beltrame, G. Nikolakopoulos, D. Shim, L. Carlone, and J. Burdick, "NeBula: Quest for Robotic Autonomy in Challenging Environments; TEAM CoSTAR at the DARPA Subterranean Challenge," *Accepted for publication in the Journal of Field Robotics, 2021*, Oct. 2021.

[60] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood, L. Carlone, and A.-a. Agha-mohammadi, "LAMP: Large-Scale Autonomous Mapping and Positioning for Exploration of Perceptually-Degraded Subterranean Environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 80–86.

[61] P. Schmuck, T. Ziegler, M. Karrer, J. Perraudin, and M. Chli, "COVINS: Visual-Inertial SLAM for Centralized Collaboration," in *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2021, pp. 171–176.

[62] P.-Y. 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*, vol. 5, no. 2, pp. 1656–1663, Apr. 2020.

[63] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2022–2038, Aug. 2022.

[64] S. A. S. Mohamed, M. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A Survey on Odometry for Autonomous Navigation Systems," *IEEE Access*, vol. 7, pp. 97 466–97 486, 2019.

[65] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[66] D. Tardioli, E. Montijano, and A. R. Mosteo, "Visual data association in narrow-bandwidth networks," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 2572–2577.

[67] X. S. Zhou and S. I. Roumeliotis, “Multi-robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 1785–1792.

[68] C. Gentner, M. Ulmschneider, and T. Jost, “Cooperative simultaneous localization and mapping for pedestrians using low-cost ultra-wideband system and gyroscope,” in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Apr. 2018, pp. 1197–1205.

[69] E. Boroson, R. Hewitt, and N. Ayanian, “Inter-Robot Range Measurements in Pose Graph Optimization,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, p. 8.

[70] Y. Cao and G. Beltrame, “VIR-SLAM: Visual, inertial, and ranging SLAM for single and multi-robot systems,” *Autonomous Robots*, vol. 45, no. 6, pp. 905–917, Sep. 2021.

[71] 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, Oct. 2012.

[72] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN Architecture for Weakly Supervised Place Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, Jun. 2018.

[73] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. USA: Cambridge University Press, 2003.

[74] M. A. Uy and G. H. Lee, “PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 4470–4479.

[75] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, “SegMatch: Segment based place recognition in 3D point clouds,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5266–5272.

[76] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[77] E. B. Olson, “Real-time correlative scan matching,” in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 4387–4393.

[78] M. J. Schuster, C. Brand, H. Hirschmüller, M. Suppa, and M. Beetz, "Multi-robot 6D graph SLAM connecting decoupled local reference filters," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 5093–5100.

[79] C. Schulz, R. Hanten, M. Reisenauer, and A. Zell, "Simultaneous Collaborative Mapping Based on Low-Bandwidth Communication," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, Feb. 2019, pp. 413–414.

[80] R. Dubois, A. Eudes, J. Moras, and V. Fremont, "Dense Decentralized Multi-Robot SLAM Based on Locally Consistent TSDF Submaps," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, p. 8.

[81] 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)*, Sep. 2017, pp. 1004–1011.

[82] E. R. Boroson and N. Ayanian, "3D Keypoint Repeatability for Heterogeneous Multi-Robot SLAM," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 6337–6343.

[83] P. Koch and S. Lacroix, "Managing environment models in multi-robot teams," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 5722–5728.

[84] R. Käslin, P. Fankhauser, E. Stumm, Z. Taylor, E. Mueggler, J. Delmerico, D. Scaramuzza, R. Siegwart, and M. Hutter, "Collaborative localization of aerial and ground robots through elevation maps," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct. 2016, pp. 284–290.

[85] Y.-W. Choi, K.-K. Kwon, S.-I. Lee, J.-W. Choi, and S.-G. Lee, "Multi-robot Mapping Using Omnidirectional-Vision SLAM Based on Fisheye Images," *ETRI Journal*, vol. 36, no. 6, pp. 913–923, 2014.

[86] N. Waniek, J. Biedermann, and J. Conradt, "Cooperative SLAM on small mobile robots," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2015, pp. 1810–1815.

[87] J. Morales and Z. M. Kassas, "Information fusion strategies for collaborative radio SLAM," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Apr. 2018, pp. 1445–1454.

[88] R. Liu, S. H. Marakkalage, M. Padmal, T. Shaganan, C. Yuen, Y. L. Guan, and U. Tan, "Collaborative SLAM Based on WiFi Fingerprint Similarity and Motion Information," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1826–1840, Mar. 2020.

[89] A. Cunningham, M. Paluri, and F. Dellaert, "DDF-SAM: Fully distributed SLAM using Constrained Factor Graphs," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 3025–3030.

[90] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, Z. Liu, H. I. Christensen, and F. Dellaert, "Multi Robot Object-Based SLAM," in *2016 International Symposium on Experimental Robotics*, ser. Springer Proceedings in Advanced Robotics, D. Kulić, Y. Nakamura, O. Khatib, and G. Venture, Eds. Cham: Springer International Publishing, 2017, pp. 729–741.

[91] I. Rekleitis, G. Dudek, and E. Milios, "Probabilistic cooperative localization and mapping in practice," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, Sep. 2003, pp. 1907–1912 vol.2.

[92] R. Madhavan, K. Fregene, and L. E. Parker, "Distributed Cooperative Outdoor Multirobot Localization and Mapping," *Autonomous Robots*, vol. 17, no. 1, pp. 23–39, Jul. 2004.

[93] H. Strasdat, J. M. M. Montiel, and A. Davison, "Visual SLAM: Why filter?" *Image Vis. Comput.*, 2012.

[94] S. Thrun and Y. Liu, "Multi-robot SLAM with Sparse Extended Information Filers," in *Robotics Research. The Eleventh International Symposium*, ser. Springer Tracts in Advanced Robotics, P. Dario and R. Chatila, Eds. Berlin, Heidelberg: Springer, 2005, pp. 254–266.

[95] T. Sasaoka, I. Kimoto, Y. Kishimoto, K. Takaba, and H. Nakashima, "Multi-robot SLAM via Information Fusion Extended Kalman Filters," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 303–308, Jan. 2016.

[96] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive Decentralized Collaborative Localization for Sparsely Communicating Robots," in *Robotics: Science and Systems XII*. Robotics: Science and Systems Foundation, 2016.

[97] M. J. Schuster, K. Schmid, C. Brand, and M. Beetz, "Distributed stereo vision-based 6D localization and mapping for multi-robot teams," *Journal of Field Robotics*, vol. 36, no. 2, pp. 305–332, 2019.

[98] F. Demim, A. Nemra, K. Louadj, M. Hamerlain, and A. Bazoula, “Cooperative SLAM for multiple UGVs navigation using SVSF filter,” *Automatika*, vol. 58, no. 1, pp. 119–129, Jan. 2017.

[99] A. Doucet, N. de Freitas, K. P. Murphy, and S. J. Russell, “Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks,” in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, ser. UAI ’00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Jun. 2000, pp. 176–183.

[100] A. Howard, “Multi-robot Simultaneous Localization and Mapping using Particle Filters,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, Dec. 2006.

[101] L. Carlone, M. Kaouk Ng, J. Du, B. Bona, and M. Indri, “Simultaneous Localization and Mapping Using Rao-Blackwellized Particle Filters in Multi Robot Systems,” *Journal of Intelligent & Robotic Systems*, vol. 63, no. 2, pp. 283–307, Aug. 2011.

[102] A. Gil, Ó. Reinoso, M. Ballesta, and M. Juliá, “Multi-robot visual SLAM using a Rao-Blackwellized particle filter,” *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 68–80, Jan. 2010.

[103] S. Dörr, P. Barsch, M. Gruhler, and F. G. Lopez, “Cooperative longterm SLAM for navigating mobile robots in industrial applications,” in *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2016, pp. 297–303.

[104] A. Mourikis and S. Roumeliotis, “Analysis of positioning uncertainty in reconfigurable networks of heterogeneous mobile robots,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*, vol. 1, Apr. 2004, pp. 572–579 Vol.1.

[105] A. I. Mourikis and S. I. Roumeliotis, “Predicting the Performance of Cooperative Simultaneous Localization and Mapping (C-SLAM),” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1273–1286, Dec. 2006.

[106] F. Dellaert and M. Kaess, “Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.

[107] L. Paull, G. Huang, M. Seto, and J. J. Leonard, “Communication-constrained multi-AUV cooperative SLAM,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 509–516.

[108] F. Dellaert, “Factor Graphs: Exploiting Structure in Robotics,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 141–166, 2021.

[109] S. Agarwal, K. Mierle, and Others, “Ceres Solver — A Large Scale Non-linear Optimization Library,” http://ceres-solver.org/.

[110] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3607–3613.

[111] 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*, vol. 38, no. 2-3, pp. 95–125, Mar. 2019.

[112] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3281–3288.

[113] J. Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert, “Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5807–5814.

[114] Y. Zhang, M. Hsiao, Y. Zhao, J. Dong, and J. J. Engel, “Distributed Client-Server Optimization for SLAM with Limited On-Device Resources,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 5336–5342.

[115] A. Cunningham, V. Indelman, and F. Dellaert, “DDF-SAM 2.0: Consistent distributed smoothing and mapping,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5220–5227.

[116] A. Cunningham, K. M. Wurm, W. Burgard, and F. Dellaert, “Fully distributed scalable smoothing and mapping with robust multi-robot data association,” in *2012 IEEE International Conference on Robotics and Automation*. St Paul, MN, USA: IEEE, May 2012, pp. 1093–1100.

[117] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, “Distributed maximum a posteriori estimation for multi-robot cooperative localization,” in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 1402–1409.

[118] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[119] Y. Zhang, M. Hsiao, J. Dong, J. Engel, and F. Dellaert, “MR-iSAM2: Incremental Smoothing and Mapping with Multi-Root Bayes Tree for Multi-Robot SLAM,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 8671–8678.

[120] N. Boumal, “An introduction to optimization on smooth manifolds,” 2020.

[121] J. Knuth and P. Barooah, “Collaborative 3D localization of robots from relative pose measurements using gradient descent on manifolds,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 1101–1106.

[122] ——, “Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1534–1539.

[123] R. Tron and R. Vidal, “Distributed image-based 3-D localization of camera sensor networks,” in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*, Dec. 2009, pp. 901–908.

[124] ——, “Distributed 3-D Localization of Camera Sensor Networks From 2-D Image Measurements,” *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3325–3340, Dec. 2014.

[125] R. Tron, J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, “A Distributed Optimization Framework for Localization and Formation Control: Applications to Vision-Based Measurements,” *IEEE Control Systems Magazine*, vol. 36, no. 4, pp. 22–44, Aug. 2016.

[126] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, “Distributed Certifiably Correct Pose-Graph Optimization,” *IEEE Transactions on Robotics*, pp. 1–20, 2021.

[127] Y. Tian, A. Koppel, A. S. Bedi, and J. P. How, “Asynchronous and Parallel Distributed Pose Graph Optimization,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5819–5826, Oct. 2020.

[128] R. Aragues, L. Carlone, G. Calafiore, and C. Sagues, “Multi-agent localization from noisy relative pose measurements,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 364–369.

[129] M. Franceschelli and A. Gasparri, “On agreement problems with gossip algorithms in absence of common reference frames,” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 4481–4486.

[130] R. Aragues, J. Cortes, and C. Sagues, “Distributed Consensus on Robot Networks for Dynamically Merging Feature-Based Maps,” *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 840–854, Aug. 2012.

[131] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein, “Graph-based distributed cooperative navigation for a general multi-robot measurement model,” *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1057–1080, Aug. 2012.

[132] W. Wang, N. Jadhav, P. Vohs, N. Hughes, M. Mazumder, and S. Gil, “Active Rendezvous for Multi-Robot Pose Graph Optimization using Sensing over Wi-Fi,” in *International Symposium on Robotics Research (ISRR)*, Hanoi, 2019.

[133] T. Fan and T. Murphey, “Majorization Minimization Methods for Distributed Pose Graph Optimization with Convergence Guarantees,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 5058–5065.

[134] E. Cristofalo, E. Montijano, and M. Schwager, “Consensus-based Distributed 3D Pose Estimation with Noisy Relative Measurements,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*. Nice, France: IEEE, Dec. 2019, pp. 2646–2653.

[135] ——, “GeoD: Consensus-based Geodesic Distributed Pose Graph Optimization,” *arXiv:2010.00156 [cs, eess]*, Sep. 2020.

[136] P. Zhu, P. Geneva, W. Ren, and G. Huang, “Distributed Visual-Inertial Cooperative Localization,” *arXiv:2103.12770 [cs]*, Aug. 2021.

[137] N. Sünderhauf and P. Protzel, “Switchable constraints for robust pose graph SLAM,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 1879–1884.

[138] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, “Robust map optimization using dynamic covariance scaling,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 62–69.

[139] Y. Latif, C. Cadena, and J. Neira, “Robust loop closing over time for pose graph SLAM,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1611–1626, Dec. 2013.

[140] 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*, vol. 5, no. 2, pp. 1127–1134, Apr. 2020.

[141] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[142] V. Indelman, E. Nelson, N. Michael, and F. Dellaert, “Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 593–600.

[143] 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)*, May 2018, pp. 2916–2923.

[144] Y. Chang, Y. Tian, J. P. How, and L. Carlone, “Kimera-Multi: A System for Distributed Multi-Robot Metric-Semantic Simultaneous Localization and Mapping,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 11 210–11 218.

[145] H. Do, S. Hong, and J. Kim, “Robust Loop Closure Method for Multi-Robot Map Fusion by Integration of Consistency and Data Similarity,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5701–5708, Oct. 2020.

[146] A. Caccavale and M. Schwager, “Wireframe Mapping for Resource-Constrained Robots,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 1–9.

[147] A. Martin and M. R. Emami, “Just-in-time cooperative simultaneous localization and mapping,” in *2010 11th International Conference on Control Automation Robotics Vision*, Dec. 2010, pp. 479–484.

[148] D. Benedettelli, A. Garulli, and A. Giannitrapani, “Multi-robot SLAM using M-Space feature representation,” in *49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 3826–3831.

[149] H. Jacky Chang, C. S. George Lee, Y. Charlie Hu, and Yung-Hsiang Lu, “Multi-robot SLAM with topological/metric maps,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2007, pp. 1467–1472.

[150] S. Saeedi, L. Paull, M. Trentini, M. Seto, and H. Li, “Group Mapping: A Topological Approach to Map Merging for Multiple Robots,” *IEEE Robotics Automation Magazine*, vol. 21, no. 2, pp. 60–72, Jun. 2014.

[151] H. Zhang, X. Chen, H. Lu, and J. Xiao, “Distributed and collaborative monocular simultaneous localization and mapping for multi-robot systems in large-scale environments,” *International Journal of Advanced Robotic Systems*, vol. 15, no. 3, p. 1729881418780178, May 2018.

[152] E. Nettleton, S. Thrun, H. Durrant-Whyte, and S. Sukkarieh, “Decentralised SLAM with Low-Bandwidth Communication for Teams of Vehicles,” in *Field and Service Robotics: Recent Advances in Reserch and Applications*, ser. Springer Tracts in Advanced Robotics, S. Yuta, H. Asama, E. Prassler, T. Tsubouchi, and S. Thrun, Eds. Berlin, Heidelberg: Springer, 2006, pp. 179–188.

[153] 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)*, May 2018, pp. 3841–3848.

[154] Y. Tian, K. Khosoussi, M. Giamou, J. How, and J. Kelly, “Near-Optimal Budgeted Data Exchange for Distributed Loop Closure Detection,” in *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, Jun. 2018.

[155] Y. Tian, K. Khosoussi, and J. P. How, “Resource-Aware Algorithms for Distributed Loop Closure Detection with Provable Performance Guarantees,” in *Algorithmic Foundations of Robotics XIII*. Springer, Cham, Dec. 2018, pp. 422–438.

[156] ——, “A resource-aware approach to collaborative loop-closure detection with provable performance guarantees,” *The International Journal of Robotics Research*, p. 0278364920948594, Sep. 2020.

[157] S. Saeedi, L. Paull, M. Trentini, and H. Li, “Neural Network-Based Multiple Robot Simultaneous Localization and Mapping,” *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2376–2387, Dec. 2011.

[158] G. Best and G. Hollinger, “Decentralised Self-Organising Maps for Multi-Robot Information Gathering,” *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 8, 2020.

[159] B. Ramtoula, R. de Azambuja, and G. Beltrame, “CAPRICORN: Communication Aware Place Recognition using Interpretable Constellations of Objects in Robot Networks,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 8761–8768.

[160] M. Kepler and D. Stilwell, “An Approach to Reduce Communication for Multi-Agent Mapping Applications,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[161] T. Cieslewski and D. Scaramuzza, “Efficient decentralized visual place recognition from full-image descriptors,” in *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, Dec. 2017, pp. 78–82.

[162] ——, “Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 640–647, Apr. 2017.

[163] M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, and P. Furgale, “The gist of maps - summarizing experience for lifelong localization,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2767–2773.

[164] L. Contreras and W. Mayol-Cuevas, “O-POCO: Online point cloud compression mapping for visual odometry and SLAM,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4509–4514.

[165] D. V. Opdenbosch and E. Steinbach, “Collaborative Visual SLAM Using Compressed Feature Exchange,” *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 57–64, Jan. 2019.

[166] R. Dubois, A. Eudes, and V. Frémont, “On Data Sharing Strategy for Decentralized Collaborative Visual-Inertial Simultaneous Localization And Mapping,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 2123–2130.

[167] R. Aragüés, E. Montijano, and C. Sagüés, "Consistent data association in multi-robot systems with limited communications," in *In Robotics: Science and Systems*, 2010, pp. 97–104.

[168] E. Montijano, R. Aragues, and C. Sagüés, "Distributed Data Association in Robotic Networks With Cameras and Limited Communications," *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1408–1423, Dec. 2013.

[169] K. Y. K. Leung, T. D. Barfoot, and H. H. T. Liu, "Distributed and decentralized cooperative simultaneous localization and mapping for dynamic and sparse robot networks," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3841–3847.

[170] ——, "Decentralized Cooperative SLAM for Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach," *Journal of Intelligent & Robotic Systems*, vol. 66, no. 3, pp. 321–342, May 2012.

[171] A. Quraishi, T. Cieslewski, S. Lynen, and R. Siegwart, "Robustness to connectivity loss for collaborative mapping," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 4580–4585.

[172] G. Tuna, V. Ç. Güngör, and S. M. Potirakis, "Wireless sensor network-based communication for cooperative simultaneous localization and mapping," *Computers & Electrical Engineering*, vol. 41, pp. 407–425, Jan. 2015.

[173] M. Bujanca, P. Gafton, S. Saeedi, A. Nisbet, B. Bodin, M. F. P. O'Boyle, A. J. Davison, P. H. J. Kelly, G. Riley, B. Lennox, M. Luján, and S. Furber, "SLAMBench 3.0: Systematic Automated Reproducible Evaluation of SLAM Systems for Robot Vision Challenges and Scene Understanding," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 6351–6358.

[174] K. Y. Leung, Y. Halpern, T. D. Barfoot, and H. H. Liu, "The UTIAS multi-robot cooperative localization and mapping dataset," *The International Journal of Robotics Research*, vol. 30, no. 8, pp. 969–974, Jul. 2011.

[175] R. Dubois, A. Eudes, and V. Frémont, "AirMuseum: A heterogeneous multi-robot dataset for stereo-visual and inertial Simultaneous Localization And Mapping," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2020, pp. 166–172.

[176] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI: IEEE, Jun. 2012, pp. 3354–3361.

[177] J. Xie, M. Kiefel, M. Sun, and A. Geiger, “Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 3688–3697.

[178] J. Martinez, S. Doubov, J. Fan, and I. A. Bã, “Pit30M: A Benchmark for Global Localization in the Age of Self-Driving Cars,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, p. 8.

[179] J. G. Rogers, J. M. Gregory, J. Fink, and E. Stump, “Test Your SLAM! The SubT-Tunnel dataset and metric for mapping,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 955–961.

[180] J. G. Rogers, A. Schang, C. Nieto-Granda, J. Ware, J. Carter, J. Fink, and E. Stump, “The DARPA SubT Urban Circuit Mapping Dataset and Evaluation Metric,” in *Experimental Robotics*, B. Siciliano, C. Laschi, and O. Khatib, Eds. Cham: Springer International Publishing, 2021, pp. 391–401.

[181] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, “Beyond Robustness: A Taxonomy of Approaches towards Resilient Multi-Robot Systems,” *arXiv:2109.12343 [cs, eess]*, Sep. 2021.

[182] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[183] G. Deng, Y. Zhou, Y. Xu, T. Zhang, and Y. Liu, “An Investigation of Byzantine Threats in Multi-Robot Systems,” in *24th International Symposium on Research in Attacks, Intrusions and Defenses*. San Sebastian Spain: ACM, Oct. 2021, pp. 17–32.

[184] Y. Ge, F. Jiang, M. Zhu, F. Wen, L. Svensson, and H. Wymeersch, “5G SLAM with Low-complexity Channel Estimation,” in *2021 15th European Conference on Antennas and Propagation (EuCAP)*, Mar. 2021, pp. 1–5.

[185] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012.

[186] L. Riazuelo, J. Civera, and J. M. M. Montiel, “C2TAM: A Cloud framework for cooperative tracking and mapping,” *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 401–413, Apr. 2014.

[187] P. Yun, J. Jiao, and M. Liu, “Towards a Cloud Robotics Platform for Distributed Visual SLAM,” in *Computer Vision Systems*, ser. Lecture Notes in Computer Science, M. Liu, H. Chen, and M. Vincze, Eds. Cham: Springer International Publishing, 2017, pp. 3–15.

[188] P. Zhang, H. Wang, B. Ding, and S. Shang, “Cloud-Based Framework for Scalable and Real-Time Multi-Robot SLAM,” in *2018 IEEE International Conference on Web Services (ICWS)*, Jul. 2018, pp. 147–154.

[189] B. D. Gouveia, D. Portugal, D. C. Silva, and L. Marques, “Computation Sharing in Distributed Robotic Systems: A Case Study on SLAM,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 410–422, Apr. 2015.

[190] M. Satyanarayanan, “The Emergence of Edge Computing,” *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[191] P. Huang, L. Zeng, X. Chen, K. Luo, Z. Zhou, and S. Yu, “Edge Robotics: Edge-Computing-Accelerated Multi-Robot Simultaneous Localization and Mapping,” *arXiv:2112.13222 [cs]*, Dec. 2021.

[192] H. S. Lee and K. M. Lee, “Multi-robot SLAM using ceiling vision,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 912–917.

[193] D. Zou and P. Tan, “CoSLAM: Collaborative Visual SLAM in Dynamic Environments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 354–366, Feb. 2013.

[194] D. Moratuwage, B. Vo, and D. Wang, “Collaborative Multi-vehicle SLAM with moving object tracking,” in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5702–5708.

[195] D. Moratuwage, D. Wang, A. Rao, N. Senarathne, and H. Wang, “RFS Collaborative Multivehicle SLAM: SLAM in Dynamic High-Clutter Environments,” *IEEE Robotics Automation Magazine*, vol. 21, no. 2, pp. 53–59, Jun. 2014.

[196] G. Battistelli, L. Chisci, and A. Laurenzi, “Random Set Approach to Distributed Multivehicle SLAM,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2457–2464, Jul. 2017.

[197] L. Gao, G. Battistelli, and L. Chisci, "Random-Finite-Set-Based Distributed Multi-robot SLAM," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1758–1777, Dec. 2020.

[198] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 1250–1257 vol.2.

[199] N. Trawny and T. Barfoot, "Optimized motion strategies for cooperative localization of mobile robots," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 1, Apr. 2004, pp. 1027–1032 Vol.1.

[200] M. Bryson and S. Sukkarieh, "Co-operative Localisation and Mapping for Multiple UAVs in Unknown Environments," in *2007 IEEE Aerospace Conference*, Mar. 2007, pp. 1–12.

[201] ——, "Architectures for Cooperative Airborne Simultaneous Localisation and Mapping," *Journal of Intelligent and Robotic Systems*, vol. 55, no. 4, pp. 267–297, Aug. 2009.

[202] N. Mahdoui, V. Frémont, and E. Natalizio, "Communicating Multi-UAV System for Co-operative SLAM-based Exploration," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, pp. 325–343, May 2020.

[203] J.-C. Trujillo, R. Munguia, E. Guerra, and A. Grau, "Cooperative Monocular-Based SLAM for Multi-UAV Systems in GPS-Denied Environments," *Sensors (Basel, Switzerland)*, vol. 18, no. 5, Apr. 2018.

[204] Z. Pei, S. Piao, M. Quan, M. Z. Qadir, and G. Li, "Active collaboration in relative observation for multi-agent visual simultaneous localization and mapping based on Deep Q Network," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, p. 1729881420920216, Mar. 2020.

[205] P. Dinnissen, S. N. Givigi, and H. M. Schwartz, "Map merging of Multi-Robot SLAM using Reinforcement Learning," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2012, pp. 53–60.

[206] M. Kontitsis, E. A. Theodorou, and E. Todorov, "Multi-robot active SLAM with relative entropy optimization," in *2013 American Control Conference*, Jun. 2013, pp. 2757–2764.

[207] N. Atanasov, J. L. Ny, K. Daniilidis, and G. J. Pappas, “Decentralized active information acquisition: Theory and application to multi-robot SLAM,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4775–4782.

[208] Y. Chen, L. Zhao, K. M. B. Lee, C. Yoo, S. Huang, and R. Fitch, “Broadcast Your Weaknesses: Cooperative Active Pose-Graph SLAM for Multiple Robots,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2200–2207, Apr. 2020.

[209] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, “SLAM++: Simultaneous Localisation and Mapping at the Level of Objects,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition.* Portland, OR, USA: IEEE, Jun. 2013, pp. 1352–1359.

[210] M. Wu, F. Huang, L. Wang, and J. Sun, “Cooperative Multi-Robot Monocular-SLAM Using Salient Landmarks,” in *2009 International Asia Conference on Informatics in Control, Automation and Robotics*, Feb. 2009, pp. 151–155.

[211] K. M. Frey, T. J. Steiner, and J. P. How, “Efficient Constellation-Based Map-Merging for Semantic SLAM,” in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 1302–1308.

[212] V. Tchuiev and V. Indelman, “Distributed Consistent Multi-Robot Semantic Localization and Mapping,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4649–4656, Jul. 2020.

[213] R. Egodagamage and M. Tuceryan, “A Collaborative Augmented Reality Framework Based on Distributed Visual Slam,” in *2017 International Conference on Cyberworlds (CW)*, Sep. 2017, pp. 25–32.

[214] ——, “Distributed monocular visual SLAM as a basis for a collaborative augmented reality framework,” *Computers & Graphics*, vol. 71, pp. 113–123, Apr. 2018.

[215] J. G. Morrison, D. Gálvez-López, and G. Sibley, “MOARSLAM: Multiple Operator Augmented RSLAM,” *Distributed Autonomous Robotic Systems*, pp. 119–132, 2016.

[216] K. Sartipi, R. C. DuToit, C. B. Cobar, and S. I. Roumeliotis, “Decentralized Visual-Inertial Localization and Mapping on Mobile Devices for Augmented Reality,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 2145–2152.

[217] C. X. Guo, K. Sartipi, R. C. DuToit, G. A. Georgiou, R. Li, J. O'Leary, E. D. Nerurkar, J. A. Hesch, and S. I. Roumeliotis, "Resource-Aware Large-Scale Cooperative Three-Dimensional Mapping Using Multiple Mobile Devices," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1349–1369, Oct. 2018.

[218] A. Sidaoui, I. H. Elhajj, and D. Asmar, "Collaborative Human Augmented SLAM," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 2131–2138.

[219] K. Yu, J. Ahn, J. Lee, M. Kim, and J. Han, "Collaborative SLAM and AR-guided navigation for floor layout inspection," *The Visual Computer*, vol. 36, no. 10, pp. 2051–2063, Oct. 2020.

[220] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena, "Maplab 2.0 – A Modular and Multi-Modal Mapping Framework," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 520–527, Feb. 2023.

[221] J. Chen, B. Sun, M. Pollefeys, and H. Blum, "A 3D Mixed Reality Interface for Human-Robot Teaming," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 11 327–11 333.

[222] M. Dutto, G. Berton, D. Caldarola, E. Fanì, G. Trivigno, and C. Masone, "Collaborative Visual Place Recognition through Federated Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4215–4225.

[223] N. Keetha, A. Mishra, J. Karhade, K. M. Jatavallabhula, S. Scherer, M. Krishna, and S. Garg, "AnyLoc: Towards Universal Visual Place Recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1286–1293, Feb. 2024.

[224] B. Ramtoula, D. D. Martini, M. Gadd, and P. Newman, "VDNA-PR: Using General Dataset Representations for Robust Sequential Visual Place Recognition," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 15 883–15 889.

[225] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning Robust Visual Features without Supervision," 2023.

[226] E. Greve, M. Büchner, N. Vödisch, W. Burgard, and A. Valada, "Collaborative Dynamic 3D Scene Graphs for Automated Driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 11 118–11 124.

[227] M. B. Peterson, Y. X. Jia, Y. Tian, A. Thomas, and J. P. How, "ROMAN: Open-Set Object Map Alignment for Robust View-Invariant Global Localization," Oct. 2024.

[228] D. Honerkamp, M. Büchner, F. Despinoy, T. Welschehold, and A. Valada, "Language-Grounded Dynamic Scene Graphs for Interactive Object Search with Mobile Manipulation," Jul. 2024.

[229] D. McGann and M. Kaess, "iMESA: Incremental Distributed Optimization for Collaborative Simultaneous Localization and Mapping," Jun. 2024.

[230] R. Murai, J. Ortiz, S. Saeedi, P. H. J. Kelly, and A. J. Davison, "A Robot Web for Distributed Many-Device Localization," *IEEE Transactions on Robotics*, vol. 40, pp. 121–138, 2024.

[231] J. Ortiz, T. Evans, and A. J. Davison, "A visual introduction to Gaussian Belief Propagation," Jul. 2021.

[232] D. Feng, Y. Qi, S. Zhong, Z. Chen, H. Chen, J. Wu, and J. Ma, "S3E: A Multi-Robot Multimodal Dataset for Collaborative SLAM," Oct. 2022.

[233] Y. Zhu, Y. Kong, Y. Jie, S. Xu, and H. Cheng, "GRACO: A Multimodal Dataset for Ground and Aerial Cooperative Localization and Mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 966–973, Feb. 2023.

[234] S. Zhao, Y. Gao, T. Wu, D. Singh, R. Jiang, H. Sun, M. Sarawata, Y. Qiu, W. Whittaker, I. Higgins, Y. Du, S. Su, C. Xu, J. Keller, J. Karhade, L. Nogueira, S. Saha, J. Zhang, W. Wang, C. Wang, and S. Scherer, "SubT-MRS Dataset: Pushing SLAM Towards All-weather Environments," May 2024.

[235] S. Chen, T. Cavallari, V. A. Prisacariu, and E. Brachmann, "Map-Relative Pose Regression for Visual Re-Localization," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Jun. 2024, pp. 20 665–20 674.

[236] A. Barroso-Laguna, S. Munukutla, V. A. Prisacariu, and E. Brachmann, "Matching 2D Images in 3D: Metric Relative Pose from Metric Correspondences," in *2024 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, Jun. 2024, pp. 4852–4863.

[237] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, “DUSt3R: Geometric 3D Vision Made Easy,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.

[238] V. Leroy, Y. Cabon, and J. Revaud, “Grounding Image Matching in 3D with MASt3R,” Jun. 2024.

[239] E. Arnold, J. Wynn, S. Vicente, G. Garcia-Hernando, Á. Monszpart, V. Prisacariu, D. Turmukhambetov, and E. Brachmann, “Map-Free Visual Relocalization: Metric Pose Relative to a Single Image,” in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 690–708.

[240] T. Barros, R. Pereira, L. Garrote, C. Premebida, and U. J. Nunes, “Place recognition survey: An update on deep learning approaches,” *arXiv:2106.10458 [cs]*, Jun. 2021.

[241] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, “Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 14 136–14 147.

[242] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, “Generalized Contrastive Optimization of Siamese Networks for Place Recognition,” Mar. 2021.

[243] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014.

[244] S. Pillai and J. Leonard, “Self-Supervised Visual Place Recognition Learning in Mobile Robots,” *Learning for Localization and Mapping Workshop IROS 2017*, Nov. 2017.

[245] F. Radenović, G. Tolias, and O. Chum, “Fine-Tuning CNN Image Retrieval with No Human Annotation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, Jul. 2019.

[246] H. Carson, J. J. Ford, and M. Milford, “Predicting to Improve: Integrity Measures for Assessing Visual Localization Performance,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9627–9634, Oct. 2022.

[247] S. Garg, T. Fischer, and M. Milford, “Where is your place, Visual Place Recognition?” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Aug. 2021, pp. 4416–4425.

[248] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2010, pp. 3304–3311.

[249] A. Taha, Y.-T. Chen, T. Misu, A. Shrivastava, and L. Davis, “Unsupervised Data Uncertainty Learning in Visual Retrieval Systems,” Feb. 2019.

[250] F. Warburg, M. Jørgensen, J. Civera, and S. Hauberg, “Bayesian Triplet Loss: Uncertainty Quantification in Image Retrieval,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV).* IEEE Computer Society, Oct. 2021, pp. 12 138–12 148.

[251] K. Cai, C. X. Lu, and X. Huang, “STUN: Self-Teaching Uncertainty Estimation for Place Recognition,” Mar. 2022.

[252] P.-Y. Lajoie, B. Ramtoula, F. Wu, and G. Beltrame, “Towards Collaborative Simultaneous Localization and Mapping: A Survey of the Current Research Landscape,” *Field Robotics*, vol. 2, no. 1, pp. 971–1000, Mar. 2022.

[253] F. Wu and G. Beltrame, “Cluster-based Penalty Scaling for Robust Pose Graph Optimization,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6193–6200, Oct. 2020.

[254] W. Churchill and P. Newman, “Experience-based navigation for long-term localisation,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, Dec. 2013.

[255] P. Mühlfellner, M. Bürki, M. Bosse, W. Derendarz, R. Philippsen, and P. Furgale, “Summary Maps for Lifelong Visual Localization,” *Journal of Field Robotics*, vol. 33, no. 5, pp. 561–590, 2016.

[256] D. Doan, Y. Latif, T.-J. Chin, Y. Liu, T.-T. Do, and I. Reid, “Scalable Place Recognition Under Appearance Change for Autonomous Driving,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV).* IEEE Computer Society, Oct. 2019, pp. 9318–9327.

[257] A.-D. Doan, Y. Latif, T.-J. Chin, and I. Reid, “HM^4: Hidden Markov Model With Memory Management for Visual Place Recognition,” *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 167–174, Jan. 2021.

[258] H. Porav, T. Bruls, and P. Newman, "Don't Worry About the Weather: Unsupervised Condition-Dependent Domain Adaptation," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct. 2019, pp. 33–40.

[259] S. Schubert, P. Neubert, and P. Protzel, "Graph-based non-linear least squares optimization for visual place recognition in changing environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 811–818, Apr. 2021.

[260] J. Wen, R. Liu, N. Zheng, Q. Zheng, Z. Gong, and J. Yuan, "Exploiting local feature patterns for unsupervised domain adaptation," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'19/IAAI'19/EAAI'19. Honolulu, Hawaii, USA: AAAI Press, Jan. 2019, pp. 5401–5408.

[261] H. Hu, Z. Qiao, M. Cheng, Z. Liu, and H. Wang, "DASGIL: Domain Adaptation for Semantic and Geometric-Aware Image-Based Localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 1342–1353, 2021.

[262] C. Chen, X. Liu, X. Xu, Y. Li, L. Ding, R. Wang, and C. Feng, "Self-Supervised Visual Place Recognition by Mining Temporal and Feature Neighborhoods," Aug. 2022.

[263] A. D. Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?" *Structural Safety*, vol. 31, no. 2, pp. 105–112, Mar. 2009.

[264] B. Zhou, A. Lapedriza, A. Torralba, and A. Oliva, "Places: An Image Database for Deep Scene Understanding," *Journal of Vision*, vol. 17, no. 10, p. 296, Aug. 2017.

[265] C. P. Robert, "Intrinsic losses," *Theory and Decision*, vol. 40, no. 2, pp. 191–214, Mar. 1996.

[266] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[267] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255.

[268] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An

Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, Mar. 2022.

[269] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[270] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 1314–1324.

[271] F. K. Gustafsson, M. Danelljan, and T. B. Schon, "Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, pp. 1289–1298.

[272] K. J. Doherty, D. M. Rosen, and J. J. Leonard, "Spectral Measurement Sparsification for Pose-Graph SLAM," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 01–08.

[273] G. Berton, C. Masone, and B. Caputo, "Rethinking Visual Geo-Localization for Large-Scale Applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4878–4888.

[274] G. Kim and A. Kim, "Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4802–4809.

[275] 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, Apr. 2022.

[276] Y. Chang, K. Ebadi, C. E. Denniston, M. F. Ginting, A. Rosinol, A. Reinke, M. Palieri, J. Shi, A. Chatterjee, B. Morrell, A.-a. Agha-mohammadi, and L. Carlone, "LAMP 2.0: A Robust Multi-Robot SLAM System for Operation in Challenging Large-Scale Underground Environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9175–9182, Oct. 2022.

[277] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, p. eabm6074, May 2022.

[278] K. J. Doherty, D. M. Rosen, and J. J. Leonard, “Performance Guarantees for Spectral Initialization in Rotation Averaging and Pose-Graph SLAM,” in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 5608–5614.

[279] C. E. Denniston, Y. Chang, A. Reinke, K. Ebadi, G. S. Sukhatme, L. Carlone, B. Morrell, and A.-a. Agha-mohammadi, “Loop Closure Prioritization for Efficient and Scalable Multi-Robot SLAM,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9651–9658, Oct. 2022.

[280] Y. Tian and J. P. How, “Spectral Sparsification for Communication-Efficient Collaborative Rotation and Translation Estimation,” *IEEE Transactions on Robotics*, vol. 40, pp. 257–276, 2024.

[281] D. Mosk-Aoyama, “Maximum algebraic connectivity augmentation is NP-hard,” *Operations Research Letters*, vol. 36, no. 6, pp. 677–679, Nov. 2008.

[282] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, “M2DGR: A Multi-Sensor and Multi-Scenario SLAM Dataset for Ground Robots,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2266–2273, Apr. 2022.

[283] P.-Y. Lajoie and G. Beltrame, “Self-Supervised Domain Calibration and Uncertainty Estimation for Place Recognition,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 792–799, Feb. 2023.

[284] ——, “Swarm-SLAM: Sparse Decentralized Collaborative Simultaneous Localization and Mapping Framework for Multi-Robot Systems,” *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 475–482, Jan. 2024.

[285] “Canadian Space Agency Analogue Terrain,” https://www.asc-csa.gc.ca/eng/laboratories-and-warehouse/analogue-terrain.asp, Aug. 2021.

[286] H. Yang, J. Shi, and L. Carlone, “TEASER: Fast and Certifiable Point Cloud Registration,” *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, Apr. 2021.

[287] W. Sheng, Q. Wang, Q. Yang, and S. Zhu, “Minimizing data exchange in ad hoc multi-robot networks,” in *ICAR ’05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, Jul. 2005, pp. 811–816.

[288] V. S. Varadharajan, D. St-Onge, B. Adams, and G. Beltrame, “SOUL: Data sharing for robot swarms,” *Autonomous Robots*, vol. 44, no. 3, pp. 377–394, Mar. 2020.

[289] C. Ghedini, C. H. C. Ribeiro, and L. Sabattini, “Toward efficient adaptive ad-hoc multi-robot network topologies,” *Ad Hoc Networks*, vol. 74, pp. 57–70, May 2018.

[290] L. Siligardi, J. Panerati, M. Kaufmann, M. Minelli, C. Ghedini, G. Beltrame, and L. Sabattini, “Robust Area Coverage with Connectivity Maintenance,” in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 2202–2208.

[291] V. S. Varadharajan, D. St-Onge, B. Adams, and G. Beltrame, “Swarm Relays: Distributed Self-Healing Ground-and-Air Connectivity Chains,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5347–5354, Oct. 2020.

[292] Y. Tian, Y. Chang, L. Quang, A. Schang, C. Nieto-Granda, J. P. How, and L. Carlone, “Resilient and Distributed Multi-Robot Visual SLAM: Datasets, Experiments, and Lessons Learned,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023, pp. 11 027–11 034.

[293] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 5135–5142.

[294] F. Fainelli, “The OpenWrt embedded development framework,” *Proceedings of the Free and Open Source Software Developers European Meeting*, p. 106, 2008.

[295] A. Corsaro, L. Cominardi, O. Hecart, G. Baldoni, J. E. P. Avital, J. Loudet, C. Guimares, M. Ilyin, and D. Bannov, “Zenoh: Unifying Communication, Storage and Computation from the Cloud to the Microcontroller,” in *2023 26th Euromicro Conference on Digital System Design (DSD)*, Sep. 2023, pp. 422–428.

[296] J. Dungan, S. Elliot, B. A. Mah, J. Poskanzer, and P. Kaustubh, “iPerf - The TCP, UDP and SCTP network bandwidth measurement tool,” https://iperf.fr/.

[297] R. Schemers, “fPing,” https://fping.org/.

[298] M. Grupp, “Evo: Python package for the evaluation of odometry and SLAM.” 2017.

[299] B. Schlager, T. Goelles, M. Behmer, S. Muckenhuber, J. Payer, and D. Watzenig, “Automotive Lidar and Vibration: Resonance, Inertial Measurement Unit, and Effects

on the Point Cloud," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 426–434, 2022.

[300] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[301] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2564–2571.

[302] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision Transformers for Dense Prediction," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 12 159–12 168.

[303] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2018, pp. 337–33 712.

[304] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching With Graph Neural Networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 4937–4946.

[305] J. Zhang, C. Herrmann, J. Hur, V. Jampani, T. Darrell, F. Cole, D. Sun, and M.-H. Yang, "MonST3R: A Simple Approach for Estimating Geometry in the Presence of Motion," Oct. 2024.

[306] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast Direct LiDAR-Inertial Odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.

[307] P.-Y. Lajoie, B. H. Baghi, S. Herath, F. Hogan, X. Liu, and G. Dudek, "PEOPLEx: PEdestrian Opportunistic Positioning LEveraging IMU, UWB, BLE and WiFi," in *ICC 2024 - IEEE International Conference on Communications*, Jun. 2024, pp. 3518–3523.

[308] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, Jun. 2004.

[309] H. Pan, X. Qi, M. Liu, and L. Liu, "Indoor scenario-based uwb anchor placement optimization method for indoor localization," *Expert Systems with Applications*, vol. 205, p. 117723, 2022.

[310] S. Herath, S. Irandoust, B. Chen, Y. Qian, P. Kim, and Y. Furukawa, "Fusion-dhl: Wifi, imu, and floorplan fusion for dense history of locations in indoor environments," in *IEEE ICRA*, 2021, pp. 5677–5683.

[311] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[312] R. Harle, "A Survey of Indoor Inertial Positioning Systems for Pedestrians," *IEEE Comm. Surveys*, vol. 15, no. 3, pp. 1281–1293, 2013.

[313] S. Herath, H. Yan, and Y. Furukawa, "RoNIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, & New Methods," in *IEEE ICRA*, May 2020, pp. 3146–3152.

[314] S. Shang and L. Wang, "Overview of WiFi fingerprinting-based indoor positioning," *IET Communications*, vol. 16, no. 7, pp. 725–733, 2022.

[315] R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, Nov. 2015.

[316] B. Barua, N. Kandil, and N. Hakem, "On performance study of TWR UWB ranging in underground mine," in *DINWC*, Apr. 2018, pp. 28–31.

[317] W. Zhao, A. Goudar, and A. P. Schoellig, "Finding the Right Place: Sensor Placement for UWB Time Difference of Arrival Localization in Cluttered Indoor Environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6075–6082, Jul. 2022.

[318] A. Poulose, J. Kim, and D. S. Han, "A sensor fusion framework for indoor localization using smartphone sensors and wi-fi rssi measurements," *Applied Sciences*, vol. 9, no. 20, 2019.

[319] Y. Zhong, T. Liu, B. Li, L. Yang, and L. Lou, "Integration of UWB and IMU for precise and continuous indoor positioning," in *UPINLBS*, Mar. 2018, pp. 1–5.

[320] 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*, vol. 31, no. 2, pp. 216–235, Feb. 2012.

[321] F. Dellaert, "Factor graphs: Exploiting structure in robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 141–166, 2021.

[322] Q. Tian, K. I.-K. Wang, and Z. Salcic, “A Low-Cost INS and UWB Fusion Pedestrian Tracking System,” *IEEE Sensors Journal*, vol. 19, no. 10, pp. 3733–3740, May 2019.

[323] C.-S. Jao, D. Wang, J. Grasso, and A. M. Shkel, “UWB-Foot-SLAM: Bounding Position Error of Foot-Mounted Pedestrian INS with Simultaneously Localized UWB Beacons,” in *2023 IEEE/ION PLANS*, Apr. 2023, pp. 150–159.

[324] J. Chen, B. Zhou, S. Bao, X. Liu, Z. Gu, L. Li, Y. Zhao, J. Zhu, and Q. Li, “A data-driven inertial navigation/bluetooth fusion algorithm for indoor localization,” *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5288–5301, 2022.

[325] J. Lu, C. Shan, K. Jin, X. Deng, S. Wang, Y. Wu, J. Li, and Y. Guo, “Onavi: Data-driven based multi-sensor fusion positioning system in indoor environments,” in *IPIN*, 2022, pp. 1–8.

[326] A. Wang, X. Ou, and B. Wang, “Improved step detection and step length estimation based on pedestrian dead reckoning,” in *IEEE ISEMC*, 2019, pp. 1–4.

[327] T. Mlotshwa, H. van Deventer, and A. S. Bosman, “Cauchy loss function: Robustness under gaussian and cauchy noise,” in *Artificial Intelligence Research*. Springer Nature Switzerland, 2022, pp. 123–138.

[328] E. Ackerman, “Blade Strike on Landing Ends Mars Helicopter’s Epic Journey - IEEE Spectrum,” *IEEE Spectrum*, Jan. 2024.

[329] M. Dorigo, G. Theraulaz, and V. Trianni, “Swarm Robotics: Past, Present, and Future [Point of View],” *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1152–1165, Jul. 2021.

[330] M. J. Milford and G. F. Wyeth, “Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1038–1053, Oct. 2008.

[331] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, and J. Civera, “Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 2623–2632.

[332] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak, L. Nogueira, M. Palieri, P. Petráček, M. Petrlík, A. Reinke, V. Krátký, S. Zhao, A.-a. Agha-mohammadi, K. Alexis, C. Heckman, K. Khosoussi, N. Kottege, B. Morrell, M. Hutter, F. Pauling, F. Pomerleau,

M. Saska, S. Scherer, R. Siegwart, J. L. Williams, and L. Carlone, “Present and Future of SLAM in Extreme Environments: The DARPA SubT Challenge,” *IEEE Transactions on Robotics*, vol. 40, pp. 936–959, 2024.

[333] A. Rosinol, J. J. Leonard, and L. Carlone, “NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023, pp. 3437–3444.

[334] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, “Gaussian Splatting SLAM,” in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2024, pp. 18 039–18 048.

[335] C. Hu and B. Li, “Distributed Inference with Deep Learning Models across Heterogeneous Edge Devices,” in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. London, United Kingdom: IEEE, May 2022, pp. 330–339.