POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Simultaneous Localization and Mapping Systems Robust to Perceptual Aliasing

PIERRE-YVES LAJOIE
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Décembre 2019

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Simultaneous Localization and Mapping Systems
Robust to Perceptual Aliasing**

présenté par **Pierre-Yves LAJOIE**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Benoît OZELL, président
Giovanni BELTRAME, membre et directeur de recherche
Jérôme LE NY, membre

DEDICATION

« I used to be indecisive, but now I'm not so sure. »
- Umberto Eco,
The Name of the Rose (1980)

ACKNOWLEDGEMENTS

This work was made possible by the continued support of many people. First, I want to thank Giovanni Beltrame, Benjamin Ramtoula and all the members of Polytechnique Montreal's MIST lab for their precious expertise, mentorship, and collaboration. I am also grateful to Luca Carlone, Siyi Hu, Yun Chang and the rest of the team from the Massachusetts Institute of Technology's SPARK Lab for welcoming me to Boston for seven months and our fruitful collaboration within the last two years. This research would not have existed without the Société Technique Élikos's team at Polytechnique with whom I have discovered the field of robotics during my undergraduate studies. Finally, I wish to thank my family and friends for their unconditional support.

RÉSUMÉ

De nos jours, la robotique gagne rapidement en popularité et promet un large éventail de nouvelles applications. Bien que le marché actuel soit dominé par les robots téléguidés, plusieurs compagnies cherchent à révolutionner notre quotidien avec des robots pleinement autonomes comme les voitures sans conducteur. En effet, les géants des technologies de partout dans le monde nous promettent régulièrement de nouvelles percées extraordinaires au niveau de l'autonomie des robots et multiplient des démonstrations plus impressionnantes les unes que les autres. Toutefois, ces systèmes autonomes devront se prouver extrêmement fiables et sécuritaires afin d'obtenir l'acceptabilité sociale nécessaire à leur succès. Malheureusement, les techniques présentement offertes par la littérature scientifique n'ont pas un niveau de robustesse à la hauteur des attentes de la population. C'est pourquoi les chercheurs universitaires et industriels doivent redoubler d'efforts afin de trouver de meilleures solutions qui sauront inspirer la confiance du public envers les systèmes robotiques autonomes. En particulier, une des composantes cruciales de tels systèmes est la localisation du robot dans son environnement. Cette composante est essentielle pour le déploiement de robots dans des environnements sans GPS (ex. à l'intérieur, sous terre, sous l'eau, etc.), puisque dans ces situations un robot doit estimer précisément sa position sur la seule base des mesures extraites à partir de ses propres senseurs. Pour y parvenir, une des techniques les plus populaires est la cartographie et localisation simultanée (SLAM) lors de laquelle un robot construit une carte de son environnement afin de suivre et estimer son propre mouvement et sa position. Cette technique est efficace, mais elle est tout de même vulnérable aux erreurs d'association et à la présence de mesures aberrantes. Les ingénieurs contournent généralement ce problème en performant une calibration très précise. Une telle calibration spécifique à l'environnement d'opération est appropriée pour des environnements très contrôlés comme ceux qu'on retrouve dans les laboratoires de recherche. Par contre, cette solution n'est pas viable pour des systèmes robotiques vendus au grand public et opérés par des utilisateurs sans formation. Une des principales causes d'erreurs en cartographie et localisation simultanée est l'aliasing perceptuel. Ce phénomène engendre des mesures aberrantes lorsqu'un robot confond deux endroits différents comme étant le même. L'addition de mesures aberrantes dans l'estimateur mène généralement à l'échec complet du système et donc possiblement à des conséquences dramatiques en termes de sécurité. Afin d'offrir des solutions à ces enjeux de robustesse, ce mémoire propose deux contributions à la littérature scientifique. La première introduit une nouvelle formulation pour le problème d'optimisation au coeur de la cartographie et localisation simultanée. Cette nouvelle formulation inclut un modèle explicite du phénomène d’aliasing perceptuel de façon à rejeter efficacement les mesures aberrantes. La seconde présente une nouvelle méthode de cartographie et localisation simultanée pour systèmes multi-robot qui est distribuée et robuste aux mesures aberrantes. Cette contribution est particulièrement importante puisque les systèmes multi-robots sont davantage vulnérables à l’aliasing perceptuel que les systèmes avec un seul robot. Plusieurs résultats expérimentaux obtenus lors de simulations, avec des jeux de données réelles et sur le terrain montrent que les techniques proposées produisent des estimés précis de localisation en présence de mesures aberrantes.

ABSTRACT

Autonomous robotics is growing fast in popularity and has a large range of potential new applications. While the current market is dominated by human-controlled robots, many companies aim to revolutionize our daily lives by focusing on autonomous robotic platforms such as self-driving cars. Indeed, companies around the world regularly promise ground-breaking innovations and show very impressive demontrations of autonomous robots. However, to get the public acceptance they need to prosper, those autonomous systems have to be as safe and as reliable as possible. Unfortunately, the current implementations are not yet sufficiently robust, so academic and industrial researchers need to investigate better and more trustworthy solutions to the many challenges of autonomous navigation and behaviors. In particular, one of the most crucial components of most autonomous systems is the self-localization mechanism. This component is essential for the deployment of robots in GPS-denied environments (e.g. indoors, underground, submarine, etc.) since a robot would need to estimate is own position in its environment based on the measurements acquired by its own onboard sensors. In that regard, one of the most popular techniques is the simultaneous localization and mapping (SLAM) approach in which the robot builds a map of its surrounding environment to track and estimate its own movements and position. This technique has been proven to be very efficient, but it is also known as quite vulnerable to data association errors and the presence of spurious measurements. Engineers often circumvent those problems by doing a very precise, yet cumbersome, parameter tuning. Such environment-specific parameter tuning is appropriate for the controlled environment found in research laboratories, but it is by no means a sufficient solution for consumer robots deployed in the wild and sold to untrained customers. One of the main causes of errors in SLAM is the perceptual aliasing phenomenon in which two different places are confused as the same by the robot. This phenomenon leads to the addition of spurious measurements in the estimation mechanism which in turn leads to the failure of the whole system. In regard to the robustness challenges in SLAM systems, this thesis proposes two contributions to the scientific literature. The first introduces a new robust formulation of the core optimization problem in SLAM that models explicitly the perceptual aliasing phenomenon to efficiently reject spurious measurements. The second presents a distributed, online and robust solution for multi-robot SLAM in robotic teams. This contribution is particularly important since multi-robot systems are more vulnerable to perceptual aliasing than single-robot systems. Extensive experimental results in simulation, on datasets and on the field show that the proposed techniques can produce accurate localization estimates in the presence of spurious measurements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

SLAM	Simultaneous Localization and Mapping
PGO	Pose Graph Optimization
SO	Special Orthogonal group
MRF	Markov Random Field
MAP	Maximum a posteriori
SfM	Structure from Motion
DC-GM	Discrete-Continuous Graphical Model
GPS	Global Positioning System
SDP	Semidefinite Programming
LS	Least Squares
RANSAC	Random Sample Consensus
NP	Non-deterministic Polynomial-time
CNN	Convolutional Neural Networks
ROS	Robot Operating System
ATE	Average Translation Error
RAM	Random-Access Memory
GB	Gigabyte
ICP	Iterative Closest Point
UGV	Unmanned Ground Vehicle

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

This Master's thesis presents the research work accomplished within the MIST Laboratory of Polytechnique Montréal (Montréal, Québec, Canada) and the SPARK Laboratory of the Massachusetts Institute of Technology (Cambridge, Massachusetts, United States of America) between January 2018 and September 2019. This document is in the format of a thesis by articles - one published and one submitted contributions are presented in Chapters 4-5.

- Pierre-Yves Lajoie, Siyi Hu, Giovanni Beltrame, and Luca Carlone, "Modeling perceptual aliasing in SLAM via discrete-continuous graphical models," *IEEE Robotics and Automation Letters (RA-L)*, 2019.

Giovanni Beltrame and Luca Carlone were cosupervisors of this work and I was helped by Siyi Hu for the experiments.

- Pierre-Yves Lajoie, Benjamin Ramtoula, Yun Chang, Luca Carlone, and Giovanni Beltrame, "DOOR-SLAM: distributed, online, and outlier resilient slam for robotic teams," *IEEE Robotics and Automation Letters (RA-L)*, 2020. [2nd revision].

In this work, Luca Carlone was a supervisor for the distributed SLAM back-end while Giovanni Beltrame was the overall supervisor. Benjamin Ramtoula was in charge of the distributed SLAM front-end and had a key role in the experiments. Yun Chang helped with the experiments on the DARPA datasets.

1.1 Context and Basic Concepts

Robotics research has been thriving for decades now, yet there are still no widely adopted autonomous consumer robotic applications. While industrial robots have had more success, they are usually kept inside cages and they are only executing repetitive and precisely calibrated tasks. This speaks to the fact that most current autonomous robots are deemed unreliable and/or unsafe even for highly trained workers. Therefore, it is imperative to drastically improve the robustness and the resilience of robotic systems in order to finally fulfill the dreams of researchers and other sci-fi enthusiasts. Moreover, we need not only to convince those who are already excited by robotics, but we need to reassure the general public that those systems are safe. Otherwise, even promising technologies in terms of convenience and productivity, such as autonomous cars, will not get the public approval they absolutely need to prosper.

Although every part of a robotic system is vulnerable to failures, the perception engine, used to fuse the data from the onboard sensors to understand the environment, is particularly prone to mistakes. Perception is one of the key aspects of autonomy in robotics since this is how a robot can localize itself. While some simple autonomous applications (e.g. going from point A to point B) using GPS can circumvent to need for onboard perception, it is absolutely essential for applications navigating without external positioning systems or involving interactions with the environment (e.g. grasping, transport of objects, etc.). In those applications, the perception system needs to provide an accurate localization of the robot in its environment in order to get reliable navigation and stable control. Indeed, if a robot is asked to move X meters forward, it needs some sort of feedback from the environment through sensing (i.e. perception) to estimate its own movement and know when to stop.

1.1.1 Simultaneous Localization and Mapping

A practical solution to the localization problem for autonomous robots is called Simultaneous Localization and Mapping (SLAM). To estimate the precise motion and localization of a robot, a SLAM system collects distinctive features (usually visual or spatial) in the surrounding environment and uses them to produce a map and to compute the position of the robot within it. As shown in Fig.1.1, SLAM systems can usually be split into two parts called the front-end and the back-end. In the front-end, information is collected from the onboard sensors and used to produce measurements of the surrounding environment such as motion estimates or the distance to some distinctive features. Given those noisy measurements, the back-end needs to estimate the map with the maximal likelihood or, in other words, the map that best explains all the collected measurements. For the sake of simplicity, this thesis focuses on visual SLAM systems.

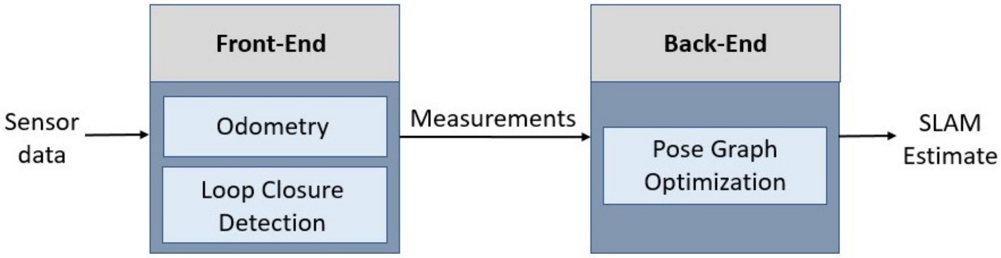


Figure 1.1: SLAM Overview

Front-End

The front-end itself can be split into two parts: the odometry module and the loop closure detection module. The odometry module estimates the motion of the robot and the loop closure detection module is used to reduce the localization error accumulated over time.

Odometry To estimate the motion of the robot, the odometry module periodically collects features from an onboard sensor and tracks them over time. Then, the module can infer the motion of the robot from the inverse motion of the collected features around it. Therefore, the odometry module can produce noisy motion measurements at every time step and feed them to the SLAM back-end. While odometry can be computed in various ways, a visual version based on stereo camera images is used in Chapter 5. In visual stereo odometry, the environment features are represented by keypoints extracted from the left and right images (e.g. SIFT [1], SURF [2], ORB [3], etc.) and they are tracked from frame to frame.

Loop Closure Detection When performing odometry, each motion measurements produced is noisy and therefore, when we concatenate those measurements to estimate the localization of the robot, there is an accumulation of the error over time (i.e. drift). To correct the accumulated error, SLAM front-end systems use a second component called the loop closure detection module. When humans get lost in a city, their first instinct will be to look around to find something they can recognize, because once they find it they can localize themselves with respect to this known feature of the environment. Loop closures detection implements this idea in the SLAM system. The component works as a place recognition system and produces measurements called loop closures when a place, seen previously, is recognized. The addition of loop closures in the computed map limits the accumulation of noise and thus reduces the localization error of the robot.

Back-End

Many techniques are available to implement the SLAM back-end, however, this thesis focuses on pose graph optimization since it is the technique used in Chapter 4 and 5. Pose graph optimization is often preferred over the competing filtering techniques because it is generally faster to solve. To understand the pose graph optimization, we first need to introduce the notion of pose which is the state of the robot (i.e. rotation and translation) at a given time frame. The robot pose at time i is defined as $\boldsymbol{T}_i \doteq [\boldsymbol{R}_i \ \boldsymbol{t}_i]$ comprised of a translation vector $\boldsymbol{t}_i \in \mathbb{R}^d$ and a rotation matrix i.e. $\boldsymbol{R}_i \in \mathrm{SO}(d)$, where $\mathrm{SO}(d)$ is the special orthogonal group and $d = 2$ in planar problems or $d = 3$ in three-dimensional problems. The robot poses are initially unknown and the aim of pose graph optimization is to compute the most likely estimate of the robot poses given the noisy measurements computed by the front-end.

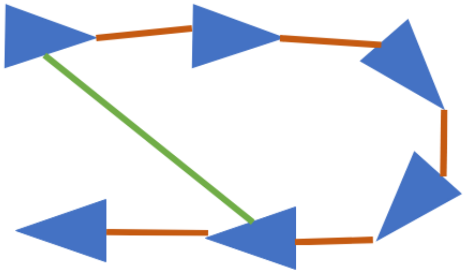


Figure 1.2: Simple pose graph

The poses are the nodes in the pose graph and the relative measurements between the poses are the edges. As mentioned before, the front-end feeds the back-end with two types of noisy measurements: odometry and loop closures. The odometry measurements link consecutive poses while the loop closures link non-consecutive poses in the graph. Fig.1.2 shows a very simple pose graph in which the poses are in blue, the odometry measurements are in orange and a single loop closure is shown in green.

Each noisy measurement $\boldsymbol{T}_{ij} \doteq [\bar{\boldsymbol{R}}_{ij}\ \bar{\boldsymbol{t}}_{ij}]$ between two poses is assumed to be sampled from the following generative model:

$$\bar{\boldsymbol{t}}_{ij} = \boldsymbol{R}_i^{\mathsf{T}}(\boldsymbol{t}_j - \boldsymbol{t}_i) + \boldsymbol{t}_{ij}^{\epsilon}, \qquad \bar{\boldsymbol{R}}_{ij} = \boldsymbol{R}_i^{\mathsf{T}}\boldsymbol{R}_j\boldsymbol{R}_{ij}^{\epsilon} \tag{1.1}$$

where $\boldsymbol{t}_{ij}^{\epsilon} \in \mathbb{R}^d$ and $\boldsymbol{R}_{ij}^{\epsilon} \in \mathrm{SO}(d)$ represent translation and rotation measurement noise, respectively.

Pose Graph Optimization (PGO) Assuming the translation noise is Normally distributed with zero mean and information matrix $\omega_t\mathbf{I}_d$ (i.e. inverse of the convariance matrix) and the rotation noise follows a Langevin distribution [4,5] with concentration parameter ω_r, the back-end can solve the following minimization problem to find the pose estimates with the maximal likelihood from the relative pose measurements in set $\mathcal{E}$:

$$\min_{\substack{\boldsymbol{t}_i \in \mathbb{R}^d \\ \boldsymbol{R}_i \in \mathrm{SO}(d)}} \sum_{(i,j)\in\mathcal{E}} \omega_t \|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i\bar{\boldsymbol{t}}_{ij}\|_2^2 + \frac{\omega_r}{2}\|\boldsymbol{R}_j - \boldsymbol{R}_i\bar{\boldsymbol{R}}_{ij}\|_F^2 \tag{1.2}$$

In this equation, $\|\cdot\|_F$ denotes the Frobenius norm. The resulting poses give an accurate estimation of the position of the robot along its trajectory and can, therefore, be used to evaluate its motion over time and accomplish its tasks.

1.2 Problem Statement

1.2.1 Perceptual Aliasing

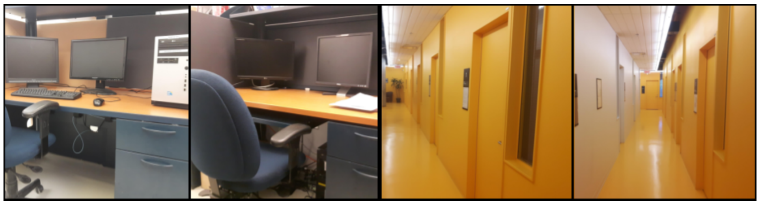


Figure 1.3: Examples of perceptual aliasing

Unfortunately, some phenomenons can cause the failure of the SLAM estimator. In particular, the presence of spurious measurements among the loop closures usually causes a considerable distortion in the resulting estimates since Eq.1.2 is using an L2 cost function which is known to be vulnerable to outliers. The phenomenon of perceptual aliasing, in which two different places are considered as the same by the SLAM front-end, is the main cause of such spurious measurements. Indeed, when two different places look very similar (see Fig.1.3 for examples), it is sometimes hard to differentiate them during the loop closure detection phase. This leads to the addition of incorrect loop closures in the pose graph, which in turn leads to very problematic distortions in the resulting estimates. Those distortions are usually very significant even in the presence of one single spurious measurement and, of course, a robot using a distorted pose graph will fail at localizing itself accurately. This can lead to the failure of the robot's mission and to dangerous or unpredictable behaviors.

Spurious Measurements

For the sake of clarity, a spurious measurement is a measurement that is wrong in the sense that the information it contains does not match with the physical reality. Spurious measurements due to perceptual aliasing are usually arbitrarily wrong and are not accounted for in the noise model. In this thesis, the term outlier is also used to refer to a spurious measurement.

1.2.2 Initialization

Another issue with the classical PGO techniques implementing formulations similar to Eq.1.2 is the need for an accurate initial guess. Indeed, since this is a local optimization problem, the initial estimates have to be close to the optimal solution in order to avoid falling into a local minimum.

1.2.3 Robust Estimation

The research work in this thesis focuses on tackling the problem of perceptual aliasing in SLAM systems. Although it could be solved by major improvements in the SLAM front-end, it is safer and realistic to expect failures from the front-end and to have a back-end able to handle spurious measurements. Therefore, the SLAM back-end needs to be based on robust estimation techniques. In the context of pose graph optimization, robustness means that the estimator can detect and ignore the wrong measurements or at least it can mitigate their effects on the resulting pose graph.

1.2.4 Multi-Robot Systems

Also, this thesis explores the robust estimation in the case of multi-robot SLAM. Multi-robot systems offer a lot of benefits over single robots in terms of resilience and productivity. Thus, it is crucial to develop techniques handling the spurious measurements in multi-robot systems. Also, while the initialization issue is not a big deal for single robot systems since the odometry usually provides a reasonably accurate initial guess, it is not possible to determine an initial alignment between multiple trajectories when we assume that there might be some outliers among the inter-robot loop closures. Hence, to solve this particular class of problem, we need SLAM estimators that do not rely on initial guesses.

1.3 Research Objectives

The research work presented in this thesis and the articles presented in Chapters 4 and 5 pursues the following objectives:

- Build a mathematical model of the perceptual aliasing phenomenon.
- Develop a pose graph optimization formulation robust to spurious measurements due to perceptual aliasing.
- Develop a distributed outlier rejection technique for multi-robot systems.
- Develop a multi-robot SLAM system robust to spurious measurements using the distributed outlier rejection technique.
- Perform field experiments of the multi-robot system robust to perceptual aliasing to prove that it works in real conditions.

1.4 Thesis Outline

The remainder of this thesis is divided into five sections. Chapter 2 presents a literature review. Chapter 3 gives an overview of the research approach and the organization of the thesis explaining the consistency between the articles and the research objectives. Chapter 4 presents an article proposing a new formulation of the robust pose graph optimization problem that explicitly models the perceptual aliasing phenomenon. Chapter 5 presents an article on a novel multi-robot SLAM system robust to spurious measurements. Chapter 6 offers a general discussion of the results shown in the previous chapters. Chapter 7 sums up the findings, the limitations and suggests future avenues of research. Finally, the references are presented in the last section.

CHAPTER 2 LITERATURE REVIEW

This chapter is an overview of the existing contributions in the two main fields of research covered in this thesis: pose graph optimization and loop closure detection. A greater emphasis is put on the former since it is at the core of the contributions presented in both Chapters 4 and 5. The existing contributions specifically related to multi-robot SLAM scenarios are also presented in each section. Multi-robot setups usually add new constraints to the SLAM problem such as the available communication bandwidth or connectivity maintenance.

2.1 Pose Graph Optimization

As stated in Chapter 1, Pose Graph Optimization (PGO) consists in the estimation of a set of poses from relative pose measurements. While this formulation is specific to robotics, a very similar problem exists in computer vision and many othe fields, usually involving the rotation estimation only. This formulation is used in solving the problem of Structure from Motion (SfM) [6] which consists to build a 3D representation of an object using a set of 2D images from a moving camera.

The classical derivation of the PGO problem (1.2), presented in Chapter 1, is taken from [5, Proposition 1].

$$\min_{\substack{\boldsymbol{t}_i \in \mathbb{R}^d \\ \boldsymbol{R}_i \in \mathrm{SO}(d)}} \sum_{(i,j)\in\mathcal{E}} \omega_t \|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij}\|_2^2 + \frac{\omega_r}{2} \|\boldsymbol{R}_j - \boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij}\|_F^2 \tag{2.1}$$

First, it is important to note that the rotation matrices are in the Special Orthogonal group $\mathrm{SO}(d)$ and, since this set is not convex, solving this problem involves performing a nonconvex optimization. This means that the problem has multiple local minima and is therefore vulnerable to an incorrect initial guess. Indeed, if the initial guess is closer to a local minimum than to the global minimum, a nonconvex optimization will not converge to the optimal solution. However, [5,7] show that it is possible to compute the globally-optimal solution using convex relaxations when the measurement noise is reasonable. In other words, it has been shown that we can approximate the original nonconvex problem to a convex formulation by relaxing the problematic constraints such as the ones on the rotation set $\mathrm{SO}(d)$. This kind of approximation is only valid if the noise in the measurements is not too high. This result is at the foundation of the work presented in Chapter 4. We will leverage the existent convex relaxation techniques to build a globally-optimal formulation robust to outlying measurements.

2.1.1 Multi-Robot PGO

The problem of multi-robot PGO is at its core the same as in single robot scenarios with the addition of inter-robot loop closures that link individual robot graphs together. Section 2.3.1 explores the ways to gather such inter-robot measurements. Most current multi-robot PGO techniques can be classified as either centralized or distributed. In the centralized approaches, the system collects all measurements from each robot at a central computation node which then jointly computes the pose estimates for all the robots involved [8–12]. The major drawback of such centralized techniques is that the computation workload and the communication bandwidth grow at least linearly with the number of robots in the team. Therefore, many research groups look into ways to compute the multi-robot PGO in a distributed fashion. Those techniques aim to rely only on local computation and communication between the robots. They can be thought of as peer-to-peer systems as opposed to master-slave systems in the centralized case. Among the early techniques to perform this distributed estimation, there are the distributed Jacobi approach to estimate 2D poses [13] and the use of Gaussian elimination [14, 15]. In more recent works, Choudhary *et al.* [16] developed a distributed Gauss-Seidel approach. This method requires only the communication of the latest pose estimates involved in the inter-robot measurements and has the benefits to avoid information double counting and complex bookkeeping. Hence, the method is used as the optimization workhorse in Chapter 5. This technique also has the benefit of preserving the privacy of the robot trajectories. Indeed, since the robots do not send their whole trajectories to each other, this limits the possibilities of attacks by an adversarial agent in the system.

2.2 Robust Pose Graph Optimization

Since the minimization problem (1.2) presented in Chapter 1 relies on the assumption that the measurement noise is light-tailed (e.g. Normal distribution), this formulation produces totally distorted pose estimates in presence of spurious measurements. In other words, the L2 cost function used is known to be brittle and not robust to outliers. Thus, it is important to mitigate the presence of such outlying measurements. RANSAC [17] and branch and bound [18] are certainly among the first techniques used to solve the problem. The former iteratively select random samples of measurements until it finds a mutually consistent set while the latter performs a graph-based search using some heuristics to select the consistent measurements. Unfortunately, they either rely heavily on precise and tedious parameter tuning or they require an exponential computation time. Those shortcomings motivated the research on robust estimators specific to pose graph optimization. Early work include the use of various robust M-estimators [6, 19] (e.g. Huber loss), or max-mixture techniques [20] which consider multiple hypotheses on the noise distribution of a measurement. The approach presented in Chapter 4 is based on the implementation of an ideal robust M-estimator known as the Truncated-Least-Squares.

Instead of simply mitigating the effect of spurious measurements, more recent approaches generally try to explicitly disable the outlying measurements during the optimization. To do so, the Vertigo technique [21, 22] introduces latent binary variables, relaxed to continuous variables, in the PGO problem to activate or deactivate the loop closure measurements. The DCS technique [23] adapts the Vertigo technique to dynamically adjust the measurement covariances to significantly reduce the weight of the outliers in the cost function. Other techniques reformulate the problem of rejecting outlying measurements as, instead, the selection of large sets of mutually consistent measurements [24–27]. While an outlier is hard to define, those techniques have the advantage of providing a simple definition of an inlier which is a measurement that is consistent with the greatest number of other measurements. Therefore, it is easier to select those well-defined measurements than to reject the outliers following an unknown distribution.

2.2.1 Multi-Robot Robust PGO

A few works have extended the research on robust PGO to multi-robot setups. In particular, Mangelson *et al.* [28] adapts the methods based on the selection of sets of mutually consistent measurements among the inter-robot loop closures, which are the most likely to contain outliers. This approach leverages some recent results in graph theory to formulate the selection of inliers as a maximal clique problem and solves it efficiently. Other works adopt an expectation-maximization approach [12] or use some extra data, such as the wireless channel information [29], to verify if a measurement is valid. Indeed, a robot can roughly estimate the distance and angle-of-arrival of a WiFi signal by collecting and analyzing multiple subsequent measurements. Then this information can be used to verify the likelihood of the vision-based measurements.

2.3 Loop Closure Detection

Considering that the spurious measurements are typically among the loop closure measurements, the detection of loop closure and the generation of such measurements is very relevant. This section focuses on the visual techniques since this is the type used in Chapter 5. First, as stated before, the detection of loop closures is essentially a place recognition problem. Therefore, most techniques compute some sort of image descriptors to compare them efficiently and to match the ones representing the same places. Those descriptors need to be relatively invariant to view-point and illumination changes because the robots are "seeing" the places at different moments. To build such descriptors multiple works have used global visual features [30, 31] or local visual features [2, 32] (e.g. SIFT [1], SURF [2], ORB [3], etc.) quantized in a bag-of-word model [33]. However, the current techniques that yield the best results are based on convolutional neural networks (CNN) [34, 35] which produce trained descriptors instead of "hand-crafted" ones. Yet, those trained descriptors can be vulnerable to environments significantly different from the ones used to train them. Indeed, even though they aim to learn a general understanding of the place recognition problem, it is not clear that they can still work on data sampled from a different distribution than the one they were trained on.

2.3.1 Multi-Robot Loop Closure Detection

In multi-robot scenarios, the challenge of finding loop closures is even greater since the robots may have different sensors or calibrations. On top of this, it is unlikely that two robots will "see" a place from the exact same angle and illumination level. Furthermore, since the information needs to be transmitted, it has to be compressed as much as possible to avoid overloading the communication bandwidth. To solve parts of the bandwidth problem, Tardioli *et al.* [36] propose the use of visual vocabulary indexes which are smaller than the full traditional keypoint descriptors (e.g. SIFT [1], SURF [2], ORB [3], etc.). Other techniques [37–39] leverage the assumption of full connectivity maintenance among the team of robots to efficiently split the communication and computation workload. By enforcing this connectivity constraint, they can safely assign a portion of the visual descriptors "spectrum" to each robot. Then, each robot only has to send its visual data to the robot in charge of the corresponding range of descriptors. Some related works also consider the resources available and adapt their communication and computation strategies accordingly [40–42]. Those adaptative strategies are a good starting point for solutions on resource-constrained platforms such as drones.

CHAPTER 3 RESEARCH APPROACH AND THESIS ORGANIZATION

This chapter presents the research approach used, its consistency with the objectives stated in Section 1.3, the links between the articles presented in Chapters 4 and 5, and the detailed organization of the thesis. The research approach follows two distinct axes to tackle the problem of perceptual aliasing in SLAM. The first axis is from the construction of theoretical models to the practical applications in robotics systems. The second axis is from single robot systems to multi-robot systems.

3.1 From Theory to Practice

Theoretical models are really useful to describe the phenomenons we are facing and get a better understanding of them. Therefore, it is useful to build them and to base our techniques on the insight they provide to produce solutions that target specifically the effects of those phenomena. In Chapter 4, we aim to explicitly model the perceptual aliasing phenomenon within the minimization problem of pose graph optimization. The new formulation resulting from this inclusion is then able to efficiently reject the spurious measurements within the loop closure measurements. However, while experiments on datasets and in simulations are a good indicator of the correctness of one solution, they are not enough to prove that the technique can overcome the reality gap and work in real-world scenarios. Proofs of concept, such as the one presented in Chapter 4, are often too computationally expensive to be directly applied to robotic platforms with limited resources. Yet, the development of fast solvers is generally a non-trivial problem on its own. Therefore, in order to build a real-world solution to the problem of perceptual aliasing, Chapter 5 presents a technique that drops some of the theoretical guarantees presented in Chapter 4, but that can solve the pose graph optimization problem with fewer computational resources. Both techniques contribute to the progress of science. One opens new avenues that may lead to new fast and robust PGO solvers, while the other provides an actual solution adapted to the current robotic platform constraints.

3.2 From a Single Robot to a Team of Robots

While most current SLAM techniques are focused on single robot systems, many autonomous robotic applications would benefit from a multi-robot SLAM solution. According to many researchers, the future of robotics will be defined by the collaboration of multiple simple robots to accomplish complex tasks, instead of very complex and task-specific single robot systems. In fact, teams of robots are often more effective since they can work in parallel or accomplish their tasks even if some members fail. On top of this, collaborative SLAM systems have the benefit of producing a shared reference frame or map for the robotic team. Such collaborative maps, also called global maps in opposition to the individual local maps, enable a large range of new applications. Indeed, many multi-robot systems need to share pieces of spatial information (e.g. position of objects, distance to an obstacle, etc.) between the robots, but spatial data is often meaningless if the robots sharing it do not have the same reference frame. So, multi-robot SLAM systems are essential to many applications in GPS-denied environments. Therefore, it is interesting to go beyond the current works and try to bring robust SLAM solutions to multi-robot systems. This entails solving many additional challenges such as distributing efficiently the computational workload and minimizing the required bandwidth between the robots. To that end, Chapter 5 provides a practical solution to the multi-robot SLAM problem in presence of outliers and its related challenges.

3.3 Document Structure

This document follows the recommended structure for a thesis by articles in which the published or submitted contributions are included in the body of the work as separate chapters.

- Chapter 1 introduces the research context and the basic concepts.
- Chapter 2 provides an overview of the relevant contributions in the literature.
- Chapter 3, this chapter, presents the research approach and the links between the different chapters.
- Chapter 4 provides a unified framework to model perceptual aliasing in SLAM and provides practical algorithms that can cope with outliers without relying on any initial guess. This work was published in the IEEE Robotics and Automation Letters in January 2019.
- Chapter 5 presents a fully distributed SLAM system with an outlier rejection mechanism that can work with less conservative parameters. This work has been submitted to the IEEE Robotics and Automation Letters in September 2019.
- Chapter 6 provides a general discussion of the results presented in chapters 4 and 5. It also presents a summary of the work, their limitations, and future research directions.
- Chapter 7 presents a summary of the work, their limitations, and future research directions.

CHAPTER 4 ARTICLE 1: MODELING PERCEPTUAL ALIASING IN SLAM VIA DISCRETE-CONTINUOUS GRAPHICAL MODELS

Preface:

Full Citation: Pierre-Yves Lajoie[1], Siyi Hu[2], Giovanni Beltrame[1], and Luca Carlone[2], "Modeling perceptual aliasing in SLAM via discrete-continuous graphical models," *IEEE Robotics and Automation Letters (RA-L)*, 2019.

DOI: 10.1109/LRA.2019.2894852

Abstract - Perceptual aliasing is one of the main causes of failure for Simultaneous Localization and Mapping (SLAM) systems operating in the wild. Perceptual aliasing is the phenomenon where different places generate a similar visual (or, in general, perceptual) footprint. This causes spurious measurements to be fed to the SLAM estimator, which typically results in incorrect localization and mapping results. The problem is exacerbated by the fact that those outliers are *highly correlated*, in the sense that perceptual aliasing creates a large number of mutually-consistent outliers. Another issue stems from the fact that most state-of-the-art techniques rely on a *given trajectory guess* (e.g., from odometry) to discern between inliers and outliers and this makes the resulting pipeline brittle, since the accumulation of error may result in incorrect choices and recovery from failures is far from trivial. This work provides a unified framework to *model* perceptual aliasing in SLAM and provides practical algorithms that can cope with outliers without relying on any initial guess. We present two main contributions. The first is a *Discrete-Continuous Graphical Model* (DC-GM) for SLAM: the continuous portion of the DC-GM captures the standard SLAM problem, while the discrete portion describes the selection of the outliers and models their correlation. The second contribution is a semidefinite relaxation to perform inference in the DC-GM that returns estimates with provable sub-optimality guarantees. Experimental results on standard benchmarking datasets show that the proposed technique compares favorably with state-of-the-art methods while not relying on an initial guess for optimization.

This work was carried out during P. Lajoie's research stay in LIDS, and was partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), the J.A. DeSève Foundation, ARL DCIST CRA W911NF-17-2-0181, ONR RAIDER N00014-18-1-2828, and MIT Lincoln Laboratory.

[1] P. Lajoie and G. Beltrame are with the Department of Computer and Software Engineering, École Polytechnique de Montréal, Montreal, Canada {pierre-yves.lajoie,giovanni.beltrame}@polymtl.ca

[2] S. Hu and L. Carlone are with the Laboratory for Information & Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, USA, {siyi,lcarlone}@mit.edu

4.1 Introduction

Simultaneous Localization and Mapping (SLAM) is the backbone of several robotics applications. SLAM is already widely adopted in consumer applications (e.g., robot vacuum cleaning, warehouse maintenance, virtual/augmented reality), and is a key enabler for truly autonomous systems operating in the wild, ranging from unmanned aerial vehicles operating in GPS-denied scenarios, to self-driving cars.

Despite the remarkable advances in SLAM, both researchers and practitioners are well aware of the brittleness of current SLAM systems. While SLAM failures are a tolerable price to pay in some consumer applications, they may put human life at risk in several safety-critical applications. For this reason, SLAM is often avoided in those applications (e.g., self-driving cars) in favor of alternative solutions where the map is built beforehand in an offline (and typically human-supervised) manner, even though this implies extra setup costs.

Arguably, the main cause of SLAM failure is the presence of incorrect data association and outliers [43]. Incorrect data association is caused by *perceptual aliasing*, the phenomenon where different places generate a similar visual (or, in general, perceptual) footprint. Perceptual aliasing leads to incorrectly associating the measurements taken by the robot to the wrong portion of the map, which may lead to map deformations and potentially to catastrophic failure of the mapping process. The problem is exacerbated by the fact that those outliers are *highly correlated*: due to the temporal nature of the data collection, perceptual aliasing creates a large number of mutually-consistent outliers. This correlation makes it even harder to judge if a measurement is an outlier, contributing to the brittleness of the resulting pipeline. Surprisingly, while the SLAM literature extensively focused on mitigating the effects of perceptual aliasing, none of the existing approaches attempt to explicitly model positive correlation between outliers.

Contribution. This work provides a unified framework to model perceptual aliasing and outlier correlation in SLAM. We propose a novel approach to obtain provably-robust SLAM algorithms: rather than developing techniques to *mitigate* the impact of perceptual aliasing, we *explicitly model* perceptual aliasing using a *discrete-continuous graphical model* (DC-GM). A simple illustration is given in Fig. 4.1. The figure shows a DC-GM where the continuous variables, shown in blue, describe a standard SLAM formulation, i.e., a pose graph [43], where the triangles represent the trajectory of a moving robot while the edges represent measurements. The figure shows that we associate a discrete variable (large red circles) to each edge/measurement in the pose graph. The discrete variables decide between accepting or rejecting a measurement. The red edges in the top portion of the figure model the cor-

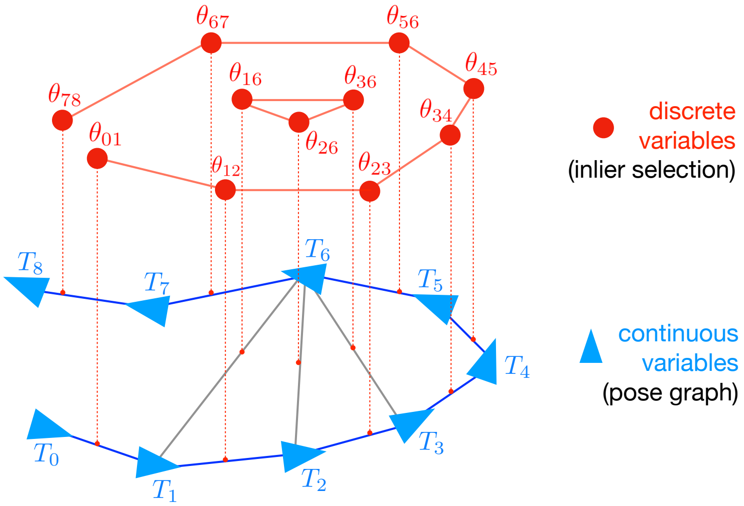


Figure 4.1: We introduce a *Discrete-Continuous Graphical Model* (DC-GM) to model perceptual aliasing in SLAM. The model describes the interactions between continuous variables (e.g., robot poses) and discrete variables (e.g., the binary selection of inliers and outliers), and captures the correlation between the discrete variables (e.g., due to perceptual aliasing).

relation between discrete variables. The expert reader can recognize the top of the figure (graph in red), to be a discrete *Markov Random Field* (MRF) [44]. The proposed model can naturally capture positive correlation between outliers: for instance, we can model the correlation between three nearby edges, (T_1, T_6), (T_2, T_6), (T_3, T_6) in Fig. 4.1, as a clique involving the corresponding discrete variables (θ_{16}, θ_{26}, θ_{36}) in the MRF (red triangle in the figure). Similarly, we can capture the temporal correlation of wheel slippage episodes by connecting variables corresponding to consecutive edges (e.g., θ_{12}, θ_{23}).

Our second contribution is the design of a semidefinite (SDP) relaxation that computes a near-optimal estimate of the variables in the DC-GM. Inference in DC-GM is intractable in general, due to the nonconvexity of the corresponding estimation problem and to the presence of discrete variables. We show how to obtain an SDP relaxation with per-instance sub-optimality guarantees, generalizing previous work on provably-correct SLAM without outliers [4, 5, 7, 45, 46]. The SDP relaxation can be solved in polynomial time by off-the-shelf convex solvers without relying on an initial guess.

Our last contribution is an experimental evaluation on standard SLAM benchmarking datasets. The experimental results show that the proposed DC-GM model compares favorably with state-of-the-art methods, including Vertigo [21], RRR [24] and DCS [23]. Moreover, they confirm that modeling outlier correlation further increases the resilience of the proposed model, which is able to compute correct SLAM estimates even when 50% of the loop closures are highly-correlated outliers. Our current (Matlab) implementation is slow, compared to state-of-the-art methods, but the proposed approach can be sped-up by designing a specialized solver along the lines of [7]. We leave these numerical aspects (which are both interesting and non-trivial on their own) for future work.

Paper structure. Section 4.2 provides preliminary notions on MRFs and pose graph optimization. Section 4.3 presents our new hybrid discrete-continuous graphical model. Section 4.4 presents a semidefinite programming relaxation for inference in the DC-GM. Section 4.5 presents experimental results, while Section 4.6 concludes the paper.

4.2 Preliminaries and Related Work

This section reviews basic concepts about Markov Random Fields and Pose Graph Optimization.

4.2.1 Markov Random Fields (MRFs)

Markov Random Fields (MRFs) are a popular graphical model for reconstruction and recognition problems in computer vision and robotics [44, 47, 48]. A *pairwise MRF* is defined by a set of ℓ *nodes* we want to label, and a set of *edges* or *potentials*, representing probabilistic constraints involving the labels of a single or a pair of nodes. Here we consider *binary* MRFs, where we associate a binary label $\theta_i \in \{-1, +1\}$ to each node $i = 1, \ldots, \ell$.

The *maximum a posteriori* (MAP) estimate of the variables in the MRF is the assignment of the node labels that attains the maximum of the posterior distribution of an MRF, or, equivalently, the minimum of the negative log-posterior [47]:

$$\min_{\substack{\theta_i \in \{-1,+1\} \\ i=1,\ldots,\ell}} \quad -\sum_{i \in \mathcal{U}} \bar{c}_i \theta_i - \sum_{(i,j) \in \mathcal{B}} \bar{c}_{ij} \theta_i \theta_j \tag{4.1}$$

where $\mathcal{U} \subseteq \{1, \ldots, \ell\}$ is the set of *unary potentials* (terms involving a single node), $\mathcal{B} \subseteq \{1, \ldots, \ell\} \times \{1, \ldots, \ell\}$ is the set of *binary potentials* (involving a pair of nodes). Intuitively, if $\bar{c}_i > 0$ (resp. $\bar{c}_i < 0$), then the unary terms encourage $+1$ (resp. -1) labels for node i. Similarly, if $\bar{c}_{ij} > 0$, then the binary term (i, j) encourages nodes i and j to have the same label (positive correlation) since that decreases the cost (4.1) by $\bar{c}_{ij}$. While several choices of unary and binary potentials are possible, the expression in eq. (4.1) is a very popular model, and is referred to as the *Ising model* [44, Section 1.4.1].

Related works consider extensions of (4.1) to continuous [49], discrete-continuous [50], or discretized [51] labels, while to the best of our knowledge, our paper is the first to propose a semidefinite solver for discrete-continuous models and use these models to capture perceptual aliasing in SLAM.

4.2.2 Pose Graph Optimization (PGO)

Pose Graph Optimization (PGO) is one of the most popular models for SLAM. PGO consists in the estimation of a set of poses (i.e., rotations and translations) from pairwise relative pose measurements. In computer vision a similar problem (typically involving only rotation) is used as a preprocessing step for bundle adjustment in Structure from Motion (SfM) [6].

PGO estimates n poses from m relative pose measurements. Each to-be-estimated pose $\boldsymbol{T}_i \doteq [\boldsymbol{R}_i \;\; \boldsymbol{t}_i]$, $i = 1, \ldots, n$, comprises a *translation* vector $\boldsymbol{t}_i \in \mathbb{R}^d$ and a rotation matrix $\boldsymbol{R}_i \in \mathrm{SO}(d)$, where $d = 2$ in planar problems or $d = 3$ in three-dimensional problems. For a pair of poses (i, j), a relative pose measurement $[\bar{\boldsymbol{R}}_{ij} \;\; \bar{\boldsymbol{t}}_{ij}]$, with $\bar{\boldsymbol{t}}_{ij} \in \mathbb{R}^d$ and $\bar{\boldsymbol{R}}_{ij} \in \mathrm{SO}(d)$, describes a noisy measurement of the relative pose between $\boldsymbol{T}_i$ and $\boldsymbol{T}_j$. Each measurement is assumed to be sampled from the following generative model:

$$\bar{\boldsymbol{t}}_{ij} = \boldsymbol{R}_i^{\mathsf{T}}(\boldsymbol{t}_j - \boldsymbol{t}_i) + \boldsymbol{t}_{ij}^{\epsilon}, \qquad \bar{\boldsymbol{R}}_{ij} = \boldsymbol{R}_i^{\mathsf{T}}\boldsymbol{R}_j\boldsymbol{R}_{ij}^{\epsilon} \tag{4.2}$$

where $\boldsymbol{t}_{ij}^{\epsilon} \in \mathbb{R}^d$ and $\boldsymbol{R}_{ij}^{\epsilon} \in \mathrm{SO}(d)$ represent translation and rotation measurement noise, respectively. PGO can be thought as an MRF with variables living on manifold: we need to assign a pose to each node in a graph, given relative measurements associated to edges $\mathcal{E}$ of the graph. The resulting graph is usually referred to as a *pose graph*.

Assuming the translation noise is Normally distributed with zero mean and information matrix $\omega_t \mathbf{I}_d$ and the rotation noise follows a Langevin distribution [4, 5] with concentration parameter ω_r, the MAP estimate for the unknown poses can be computed by solving the following optimization problem:

$$\min_{\substack{\boldsymbol{t}_i \in \mathbb{R}^d \\ \boldsymbol{R}_i \in \mathrm{SO}(d)}} \sum_{(i,j)\in\mathcal{E}} \omega_t \|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij}\|_2^2 \;+\; \frac{\omega_r}{2}\|\boldsymbol{R}_j - \boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij}\|_F^2 \tag{4.3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The derivation of (4.3) is given in [5, Proposition 1]. The estimator (4.3) involves solving a nonconvex optimization, due to the nonconvexity of the set $\mathrm{SO}(d)$. Recent results [5, 7] show that one can still compute a globally-optimal solution to (4.3), when the measurement noise is reasonable, using convex relaxations.

Unfortunately, the minimization (4.3) follows from the assumption that the measurement noise is light-tailed (e.g., Normally distributed translation noise) and it is known to produce completely wrong pose estimates when this assumption is violated, i.e., in presence of outlying measurements.

An interpretation of landmark-based SLAM as a pairwise MRF with continuous variables is given by Dellaert in [52].

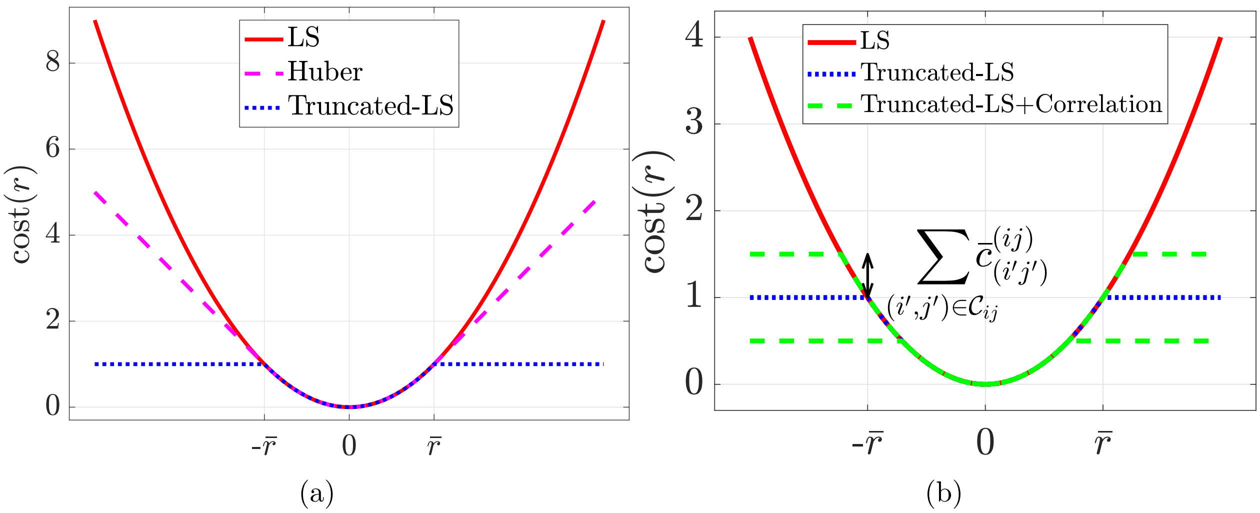


Figure 4.2: (a) Cost associated to each residual error in the least squares (LS), Huber, and truncated LS estimators. (b) The correlation terms $\bar{c}^{(ij)}_{(i'j')}$ in eq. (4.9) have the effect of altering the error threshold in the truncated LS estimator.

4.2.3 Robust PGO

The sensitivity to outliers of the formulation (4.3) is due to the fact that we minimize the squares of the *residual errors* (quantities appearing in the squared terms): this implies that large residuals corresponding to spurious measurements dominate the cost. Robust estimators reduce the impact of outliers by adopting cost functions that grow slowly (i.e., less than quadratically) when the residual exceeds a given upper bound $\bar{r}$. This is the idea behind robust M-estimators, see [53]. For instance, the Huber loss in Fig. 4.2 grows linearly outside the quadratic region $[-\bar{r}, +\bar{r}]$. Ideally, one would like to adopt a truncated least squares (LS) formulation (Fig. 4.2) where the impact of arbitrarily large outliers remains bounded. Such a formulation, however, is non-convex and non-differentiable, typically making the resulting optimization hard.

Traditionally, outlier mitigation in SLAM and SfM relied on the use of robust M-estimators, see [6,19]. Agarwal *et al.* [23] propose *Dynamic Covariance Scaling* (DCS), which dynamically adjusts the measurement covariances to reduce the influence of outliers. Olson and Agarwal [20] use a max-mixture distribution to accommodate multiple hypotheses on the noise distribution of a measurement. Casafranca *et al.* [54] minimize the ℓ_1-norm of the residual errors. Lee *et al.* [55] use expectation maximization. An alternative set of approaches attempts to explicitly *identify* and reject outliers. Early techniques include RANSAC [17] and branch & bound [18]. Sünderhauf and Protzel [21, 22] propose Vertigo, which augments the PGO problem with latent binary variables (then relaxed to continuous variables) that are responsible for deactivating outliers. Latif *et al.* [24], Carlone *et al.* [26], Graham *et al.* [27], Mangelson *et al.* [28] look for large sets of "mutually consistent" measurements. Pfingsthorn and Birk [56,57] model ambiguous measurements using hyperedges and mixture of Gaussians, and provide a measurement selection approach that also constructs an initial guess for PGO. Both [20] and [57] implicitly model negative correlation (or, more precisely, mutual exclusivity) between multiple edge hypotheses. The introduction of discrete variables has also been used to reconcile data association and semantic SLAM [58], and to deal with unknown data association in SfM [59].

4.3 Discrete-continuous Graphical Models for Robust Pose Graph Optimization

We propose a novel approach for robust PGO that addresses the three main limitations of the state of the art. First, rather than mitigating outlier correlation, we explicitly model it. Second, our PGO method (Section 4.4) does not rely on any initial guess. Third, we go beyond recently proposed convex relaxations for robust rotation and pose estimation [46,60,61], and use a *nonconvex* loss, namely, the *truncated LS* cost in Fig. 4.2. This circumvents issues with convex robust loss functions which are known to have low breakdown point (e.g., the Huber loss [46] or ℓ_1 norm [46,60,61] can be compromised by the presence of a single "bad" outlier).

4.3.1 A unified view of robust PGO

Let us partition the edges of the pose graph into odometric edges $\mathcal{E}_{od}$ and loop-closure edges $\mathcal{E}_{lc}$. Perceptual aliasing affects exteroceptive sensors, hence —while we can typically trust odometric edges— loop closures may include outliers.

According to the discussion in Section 4.2.3, an ideal formulation for robust PGO would use a truncated LS cost for the loop-closure edges in $\mathcal{E}_{lc}$:

$$\begin{aligned}\min_{\substack{\boldsymbol{t}_i\in\mathbb{R}^d\\ \boldsymbol{R}_i\in\mathrm{SO}(d)}} \; & \sum_{(i,j)\in\mathcal{E}_{od}} \omega_t\|\boldsymbol{t}_j-\boldsymbol{t}_i-\boldsymbol{R}_i\bar{\boldsymbol{t}}_{ij}\|_2^2 \;+\; \frac{\omega_r}{2}\|\boldsymbol{R}_j-\boldsymbol{R}_i\bar{\boldsymbol{R}}_{ij}\|_F^2 \\ & + \sum_{(i,j)\in\mathcal{E}_{lc}} \omega_t f_{\bar{c}_t}(\|\boldsymbol{t}_j-\boldsymbol{t}_i-\boldsymbol{R}_i\bar{\boldsymbol{t}}_{ij}\|_2) + \frac{\omega_r}{2} f_{\bar{c}_R}(\|\boldsymbol{R}_j-\boldsymbol{R}_i\bar{\boldsymbol{R}}_{ij}\|_F)\end{aligned} \tag{4.4}$$

where, for a positive scalar c, the function $f_c(\cdot)$ is:

$$f_c(x) = \begin{cases} x^2 & \text{if } |x|\leq c \\ c^2 & \text{otherwise}\end{cases} \tag{4.5}$$

While the formulation (4.4) would be able to tolerate arbitrarily "bad" outliers, it has two main drawbacks. First, $f_c(\cdot)$ is non-convex, adding to the non-convexity already induced by the rotations ($\mathrm{SO}(d)$ is a non-convex set). Second, the cost is non-differentiable, as shown in Fig. 4.2, hence also preventing the use of fast (but local) smooth optimization techniques.

The first insight behind the proposed approach is simple but powerful: we can rewrite the truncated LS cost (4.5) as a minimization over a binary variable:

$$f_c(x) = \min_{\theta \in \{-1;+1\}} \frac{(1+\theta)}{2} x^2 + \frac{(1-\theta)}{2} c^2 \tag{4.6}$$

To show the equivalence between (4.6) and (4.5), we observe that for any $\bar{x}$ such that $\bar{x}^2 < c^2$ (or $|\bar{x}| < c$), the minimum in (4.6) is attained for $\theta = +1$ and $f_c(\bar{x}) = \bar{x}^2$; on the other hand, for any $\hat{x}$ such that $\hat{x}^2 > c^2$ (or $|\hat{x}| > c$), the minimum in (4.6) is attained for $\theta = -1$ and $f_c(\hat{x}) = c^2$.

We can now use the expression (4.6) to rewrite the cost function (4.4) by introducing a binary variable for each rotation and translation measurement:

$$\begin{aligned}
\min_{\substack{\boldsymbol{t}_i \in \mathbb{R}^d \\ \boldsymbol{R}_i \in \mathrm{SO}(d) \\ \theta^t_{ij} \in \{-1;+1\} \\ \theta^R_{ij} \in \{-1;+1\}}} & \sum_{(i,j)\in\mathcal{E}_{od}} \omega_t \|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij}\|_2^2 \; + \; \frac{\omega_r}{2} \|\boldsymbol{R}_j - \boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij}\|_F^2 + \\
& + \sum_{(i,j)\in\mathcal{E}_{lc}} \omega_t \frac{(1+\theta^t_{ij})}{2} \|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij}\|_2^2 + \omega_t \frac{(1-\theta^t_{ij})}{2} \bar{c}_t^2 \\
& + \frac{\omega_r}{2} \frac{(1+\theta^R_{ij})}{2} \|\boldsymbol{R}_j - \boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij}\|_F^2 + \frac{\omega_r}{2} \frac{(1-\theta^R_{ij})}{2} \bar{c}_R^2
\end{aligned} \tag{4.7}$$

where $\bar{c}_t$ and $\bar{c}_R$ are simply the largest admissible residual errors for a translation and rotation measurement to be considered an inlier. Intuitively, θ^t_{ij} decides whether a translation measurement is an inlier ($\theta^t_{ij} = +1$) or an outlier ($\theta^t_{ij} = -1$); θ^R_{ij} has the same role for rotation measurements. While eq. (4.7) resembles formulations in the literature, e.g., Sünderhauf's switchable constraints [21], establishing connections with the truncated LS cost provides a physically meaningful interpretation of the parameters $\bar{c}_t$ and $\bar{c}_R$ (maximum admissible residuals). Moreover, we will push the boundary of the state of the art by modeling the outlier correlation (next sub-section) and proposing global semidefinite solvers (Section 4.4).

4.3.2 Modeling outlier correlation and perceptual aliasing

The goal of this section is to introduce extra terms in the cost (4.7) to model the correlation between subsets of binary variables, hence capturing outlier correlation. For the sake of simplicity, we assume that a unique binary variable is used to decide if both the translation and the rotation components of measurement (i, j) are accepted, i.e., we set $\theta^t_{ij} = \theta^R_{ij} \doteq \theta_{ij}$.

This assumption is not necessary for the following derivation, but it allows using a more compact notation. In particular, we rewrite (4.7) more succinctly as:

$$\min_{\substack{\boldsymbol{T}_i \in \mathrm{SO}(d)\times\mathbb{R}^d \\ \theta_{ij}\in\{-1;+1\}}} \sum_{(i,j)\in\mathcal{E}_{od}} \|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} + \sum_{(i,j)\in\mathcal{E}_{lc}} \frac{(1+\theta_{ij})}{2}\|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} + \frac{(1-\theta_{ij})}{2}\bar{c} \tag{4.8}$$

where for two matrices $\boldsymbol{M}$ and $\boldsymbol{\Omega}$ of compatible dimensions $\|\boldsymbol{M}\|^2_{\boldsymbol{\Omega}} \doteq \mathrm{tr}\left(\boldsymbol{M}\boldsymbol{\Omega}\boldsymbol{M}^\mathsf{T}\right)$, and –following [62]– we defined:

$$\boldsymbol{T}_i \doteq [\boldsymbol{R}_i\ \boldsymbol{t}_i], \quad \bar{\boldsymbol{T}}_{ij} \doteq \begin{bmatrix} \bar{\boldsymbol{R}}_{ij} & \bar{\boldsymbol{t}}_{ij} \\ \mathbf{0}_d^\mathsf{T} & 1 \end{bmatrix}, \quad \boldsymbol{\Omega} \doteq \begin{bmatrix} \frac{\omega_r}{2}\mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d^\mathsf{T} & \omega_t \end{bmatrix}$$

and for simplicity we called $\bar{c} \doteq \omega_r \bar{c}_t^2 + \frac{\omega_t}{2}\bar{c}_R^2$.

We already observed in Section 4.2.1 that to model the correlation between two discrete variables θ_{ij} and $\theta_{i'j'}$ we can add terms $-\bar{c}^{(ij)}_{(i'j')}\theta_{ij}\theta_{i'j'}$ to the cost function, which penalize a mismatch between θ_{ij} and $\theta_{i'j'}$ whenever the scalar $\bar{c}^{(ij)}_{(i'j')}$ is positive. This leads to generalizing problem (4.8) as follows:

$$\min_{\substack{\boldsymbol{T}_i \in \mathrm{SO}(d)\times\mathbb{R}^d \\ \theta_{ij}\in\{-1;+1\}}} \sum_{(i,j)\in\mathcal{E}_{od}} \|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} + \sum_{(i,j)\in\mathcal{E}_{lc}} \frac{(1+\theta_{ij})}{2}\|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} + \frac{(1-\theta_{ij})}{2}\bar{c} - \sum_{(i,j),(i',j')\in\mathcal{C}} \bar{c}^{(ij)}_{(i'j')}\theta_{ij}\theta_{i'j'} \tag{4.9}$$

where the set $\mathcal{C}$ contains pairs of edges that are correlated, i.e., pairs of edges $(i,j),(i',j')$ for which if (i,j) is found to be an outlier, it is likely for (i',j') to be an outlier as well.

In the supplemental material (appendix A), we show that the correlation terms have the effect of altering the threshold $\bar{c}$. For instance, if all neighbors (i',j') of an edge (i,j) are inliers ($\theta_{i'j'} = 1$), the correlation terms become $\bar{c}^{(ij)}_{(i'j')}\theta_{ij}$ and they have the effect of increasing $\bar{c}$. We also show that perturbations of $\bar{c}$ are bounded in the interval $[\bar{c} - 2\sum_{(i',j')\in\mathcal{C}_{ij}}\bar{c}^{(ij)}_{(i'j')}, \bar{c} + 2\sum_{(i',j')\in\mathcal{C}_{ij}}\bar{c}^{(ij)}_{(i'j')}]$, where $\mathcal{C}_{ij}$ is the set of edges correlated to the edge (i,j), see Fig.4.2(b) for an illustration.

Problem (4.9) describes a *discrete-continuous graphical model* (DC-GM) as the one pictured in Fig. 4.1: the optimization problems returns the most likely assignment of variables in the graphical model, which contains both continuous variables ($\boldsymbol{T}_i$) and discrete variables (θ_{ij}). The reader can notice that if the assignment of discrete variables is given, (4.9) reduces to PGO, while if the continuous variables are given, then (4.9) becomes an MRF, where the second sum in (4.9) defines the unary potentials for each discrete variable in the MRF.

4.4 Inference in DC-GM via Convex Relaxation

The DC-GM presented in Section 4.3 captures two very desirable aspects: (i) it uses a robust truncated LS loss function and (ii) it can easily model outlier correlation. On the downside, the optimization (4.9) is intractable in general, due to the presence of discrete variables and the non-convex nature of the rotation set SO(d).

Here we derive a convex relaxation that is able to compute near-optimal solutions for (4.9) in polynomial time. While we do not expect to compute exact solutions for (4.9) in all cases in polynomial time (the problem is NP-hard in general), our goal is to obtain a relaxation that works well when the noise on the inliers is reasonable (i.e., similar to the one found in practical applications) and whose quality is largely insensitive to the presence of a large number of (arbitrarily "bad") outliers.

In order to derive our convex relaxation, it is convenient to reformulate (4.9) using a more compact matrix notation. Let us first "move" the binary variables inside the norm and drop constant terms from the objective in (4.9):

$$\begin{aligned}\min_{\substack{\boldsymbol{T}_i \in \mathrm{SO}(d)\times\mathbb{R}^d \\ \theta_{ij}\in\{-1;+1\}}} \sum_{(i,j)\in\mathcal{E}_{od}} \|\boldsymbol{T}_j - \boldsymbol{T}_i \bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} + \sum_{(i,j)\in\mathcal{E}_{lc}} \| \frac{(1+\theta_{ij})}{2} (\boldsymbol{T}_j - \boldsymbol{T}_i \bar{\boldsymbol{T}}_{ij}) \|^2_{\boldsymbol{\Omega}} \\ - \sum_{(i,j)\in\mathcal{E}_{lc}} \frac{\theta_{ij}}{2}\bar{c} - \sum_{(i,j),(i',j')\in\mathcal{C}} \bar{c}^{(ij)}_{(i'j')} \theta_{ij}\theta_{i'j'}\end{aligned} \tag{4.10}$$

where we noted that $\frac{(1+\theta_{ij})}{2}$ is either zero or one, hence it can be safely moved inside the norm, and we dropped $\frac{1}{2}\bar{c}$.

We can now stack pose variables into a single $d \times (d+1)n$ matrix $\boldsymbol{T} \doteq [\boldsymbol{T}_1 \, \dots \, \boldsymbol{T}_n]$. We also use a *matrix representation* for the binary variables $\Theta \doteq [\Theta_1 \, \dots \, \Theta_\ell] \in \{-\boldsymbol{I}_d; +\boldsymbol{I}_d\}^\ell$ where $\ell = |\mathcal{E}_{lc}|$ denotes the number of loop closures and $\boldsymbol{I}_d$ denotes the identity matrix of size d. Finally, we define:

$$\begin{aligned}\boldsymbol{X} = [\boldsymbol{T} \quad \Theta \quad \boldsymbol{I}_d] \in (\mathrm{SO}(d)\times\mathbb{R}^d)^n \times \{-\boldsymbol{I}_d; +\boldsymbol{I}_d\}^\ell \times \boldsymbol{I}_d \\ \left(\text{note: } \boldsymbol{X}^\mathsf{T}\boldsymbol{X} = \begin{bmatrix} \boldsymbol{T}^\mathsf{T}\boldsymbol{T} & \boldsymbol{T}^\mathsf{T}\Theta & \boldsymbol{T}^\mathsf{T} \\ \Theta^\mathsf{T}\boldsymbol{T} & \Theta^\mathsf{T}\Theta & \Theta^\mathsf{T} \\ \boldsymbol{T} & \Theta & \boldsymbol{I}_d \end{bmatrix} \right)\end{aligned} \tag{4.11}$$

The following proposition provides a compact reformulation of problem (4.10) using the matrix $\boldsymbol{X}$ in (4.11):

Proposition 1 (Inference in DC-GM). *Problem* (4.10) *can be equivalently written in compact* *form using the matrix variable* $\boldsymbol{X}$ *in* (4.11) *as follows:*

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \operatorname{tr}\left(\boldsymbol{Q}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\right) + \sum_{e=(i,j)\in\mathcal{E}_{lc}} \operatorname{tr}\left(\boldsymbol{U}_e\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{W}_e\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\right) \\ \textit{subject to} \quad & \boldsymbol{X} \in (SO(d) \times \mathbb{R}^d)^n \times \{-\boldsymbol{I}_d; +\boldsymbol{I}_d\}^{\ell} \times \boldsymbol{I}_d \end{aligned} \tag{4.12}$$

where $\boldsymbol{Q}, \boldsymbol{U}_e, \boldsymbol{W}_e \in \mathbb{R}^{(n(d+1)+d\ell+d)\times(n(d+1)+d\ell+d)}$ *are sparse matrices (for all loop closures* $e \in \mathcal{E}_{lc}$*). The expressions for these (known) matrices are given in Appendix.*

Intuitively, $\boldsymbol{Q}$ in (4.12) captures the terms in the first, third, and fourth sum in (4.10), while the sum including $\boldsymbol{U}_e, \boldsymbol{W}_e$ (one term for each loop closure e) captures the terms in the second sum in (4.10) which couples discrete and continuous variables.

The final step before obtaining a convex relaxation is to write the "geometric" constraints in (4.12) in terms of linear algebra. Towards this goal, we relax the set SO(d) (rotation matrices) to O(d) (orthogonal matrices), i.e., we drop the constraint that rotation matrices need to have determinant +1. In related work, we found the determinant constraint to be redundant [63]. Moreover, this is done for the sake of simplicity, while the determinant constraints can be still modeled as shown in [63]. Then, we obtain an SDP relaxation of Problem (4.12) by (i) introducing a matrix variable $\boldsymbol{Z} = \boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}$ and rewriting (4.12) as a function of $\boldsymbol{Z}$, (ii) noting that any matrix $\boldsymbol{Z} = \boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}$ is a positive-semidefinite ($\boldsymbol{Z} \succeq 0$) rank-d matrix, and (iii) relaxing the non-convex rank-d constraint.

Proposition 2 (Semidefinite Relaxation of DC-GM)**.** *The following SDP is a convex relaxation of Problem* (4.12)*:*

$$\begin{aligned} \min_{\boldsymbol{Z}} \quad & \operatorname{tr}(\boldsymbol{Q}\boldsymbol{Z}) + \sum_{e=(i,j)\in\mathcal{E}_{lc}} \operatorname{tr}(\boldsymbol{U}_e\boldsymbol{Z}\boldsymbol{W}_e\boldsymbol{Z}) \\ \textit{subject to} \quad & [\boldsymbol{Z}]_{ii} = \begin{bmatrix} \boldsymbol{I}_d & * \\ * & * \end{bmatrix} \qquad {\scriptstyle i=1,\ldots,n} \\ & [\boldsymbol{Z}]_{ii} = \boldsymbol{I}_d \qquad {\scriptstyle i=n+1,\ldots,n+\ell+1} \\ & [\boldsymbol{Z}]_{ij} = \operatorname{idiag}([\boldsymbol{Z}]_{ij}) \qquad {\scriptstyle i,j=n+1,\ldots,n+\ell+1} \\ & \boldsymbol{Z} \succeq 0 \end{aligned} \tag{4.13}$$

where $[\boldsymbol{Z}]_{ij}$ *denotes the block of* $[\boldsymbol{Z}]$ *in block row* i *and block column* j*, the symbol "*" denotes entries that are unconstrained (we follow the notation of [62]), and where* $[\boldsymbol{Z}]_{ij} = \operatorname{idiag}([\boldsymbol{Z}]_{ij})$ *enforces the block* $[\boldsymbol{Z}]_{ij}$ *to be an isotropic diagonal matrix, i.e., a scalar multiple of* $\mathbf{I}_d$*.*

Let us explain the constraints in (4.13), by using the block structure of $\boldsymbol{Z}$ described in (4.11). For $i = 1, \ldots, n$, the diagonal blocks $[\boldsymbol{Z}]_{ii}$ are in the form of $\boldsymbol{T}_i^{\mathsf{T}}\boldsymbol{T}_i$, hence the first constraint in (4.13) captures the orthogonality of the rotation matrix included in each pose $\boldsymbol{T}_i$. For $i = n+1,\ldots,n+\ell$, the diagonal blocks $[\boldsymbol{Z}]_{ii}$ are in the form of $\Theta_i^{\mathsf{T}}\Theta_i$ and since $\Theta_i \in \{-\mathbf{I}_d, +\mathbf{I}_d\}$, $\Theta_i^{\mathsf{T}}\Theta_i = \mathbf{I}_d$, which is captured in the second constraint in (4.13); similar considerations hold for $i = \ell + 1$. Finally, the products $\Theta_i^{\mathsf{T}}\Theta_j$ (captured by the blocks $[\boldsymbol{Z}]_{ij}$ when $i, j = n+1,\ldots,n+\ell+1$) must be diagonal matrices, producing the third constraint in (4.13).

The SDP relaxation can be solved using off-the-shelf convex solvers. In particular, we note that the constraint $[\boldsymbol{Z}]_{ij} = \operatorname{idiag}([\boldsymbol{Z}]_{ij})$ can be implemented as a set of linear equality constraints. Indeed, this constraint can be rewritten as $[\boldsymbol{Z}]_{ij} = [\boldsymbol{Z}]_{ij,11} \cdot \mathbf{I}_d$ where $[\boldsymbol{Z}]_{ij,11}$ is the top left entry of $[\boldsymbol{Z}]_{ij}$. Therefore, the constraint enforces that the matrix has offdiagonal elements equal to zero and diagonal elements equal to a single scalar $[\boldsymbol{Z}]_{ij,11}$. The SDP relaxation (4.13) enjoys the typical per-instance optimality guarantees described in related work [4, 5, 7, 46]. In particular, if the solution $\boldsymbol{Z}^{\star}$ of (4.13) has rank d, then the relaxation solves (4.12) *exactly*. Moreover, the optimal objective of (4.13) is a lower bound for the optimal objective (4.12), a property that can be used to evaluate how sub-optimal a given estimate is, see [4, 5].

4.5 Experiments

This section presents two sets of experiments. Section 4.5.1 reports the results of Monte Carlo runs on a synthetic dataset and shows that the proposed technique compares favorably with the state of the art, and that modeling outlier correlation leads to performance improvements. Section 4.5.2 evaluates the proposed techniques in three real benchmarking datasets and shows that our approach outperforms related techniques while not requiring any initial guess.

4.5.1 Experiments On Synthetic Dataset

Methodology. For this set of experiments, we built a synthetic dataset composed of a simple trajectory on a grid of 20 by 10 nodes. Then we added random groups of loop closures between the rows as described in [21]. Typically, in presence of perceptual aliasing, the outliers are in mutually-consistent groups, e.g., the SLAM front-end generates multiple false loop closures in sequence. To simulate this phenomenon, we set the loop closures in each group to be either all inliers or all outliers. We set the standard deviation of the translation and rotation noise for the inlier measurements (odometry and correct loop closures) to 0.1m and 0.01rad. The maximum admissible errors for the truncated LS (4.5) is set to 1σ of the measurement noise. We tested the performance of our techniques for increasing levels of outliers, up to the case where 50% of the loop closure are outliers. Fig. 4.4 shows the overlay of multiple trajectories

(5 runs) estimated by our techniques versus the ground truth trajectory (green), when 50% of the loop closures are outliers.

Compared Techniques. We evaluate the performance of the proposed technique, DC-GM, which solves the minimization problem (4.9). In order to show that capturing outlier correlation leads to performance improvements, we also test a variation of the proposed approach, called DC-GMd, which implements the minimization problem (4.8), where outliers are assumed uncorrelated (the "d" stands for decoupled). In both DC-GM and DC-GMd, we solve the SDP using cvx [64] in Matlab. If the resulting matrix does not have rank $d = 2$ (in which case we are not guaranteed to get an exact solution to the non-relaxed problem), we round the result to detect the set of outliers, and re-run the optimization without the outliers.

We benchmarked our approach against three other robust PGO techniques, i.e., Vertigo [21], RRR [24] and DCS [23]. For Vertigo we use the default parameters, while for RRR and DCS we report results for multiple choices of parameters, since these parameters have a significant impact on performance. In particular, for RRR we consider three cluster sizes ($t_g = \{1, 5, 10\}$) and for DCS we considered three values of the parameter $\Phi = \{1, 10, 100\}$ [23]. For all these techniques, we used the odometric estimate as initial guess.

Results and Interpretation. Fig. 4.3 reports the average translation error for all the compared approaches and for increasing percentage of outliers. Vertigo's error grows quickly beyond 30% of outliers. For DCS, the performance heavily relies on correct parameter tuning: for some choice of parameters ($\Phi = \{10, 100\}$) it has excellent performance while the approach fails for $\Phi = 1$. Unfortunately, these parameters are difficult to tune in general (we will observe in Section 4.5.2 that the choice of parameters mentioned above may not produce the best results in the real tests). The proposed techniques, DC-GMd and DC-GM, compare favorably against the state of the art while they are slightly less accurate than RRR, which produced the best results in simulation.

In order to shed light on the performance of DC-GM and DC-GMd, Fig. 4.5 reports the average percentage of outliers rejected by these two techniques. While from the scale of the y-axis we note that both techniques are able to reject most outliers, DC-GM is able to reject *all* outliers in all tests even when up to 50% of the loop closures are spurious. As expected, modeling outlier correlation as in DC-GM improves outlier rejection performance. We also recorded the number of *incorrectly rejected inliers*: both approaches do not reject any inlier and for this reason we omit the corresponding figure.

In our tests, the SDP relaxation (4.13) typically produces low-rank solutions with 2 relatively large eigenvalues, followed by 2 smaller ones (the remaining eigenvalues are numerically zero).

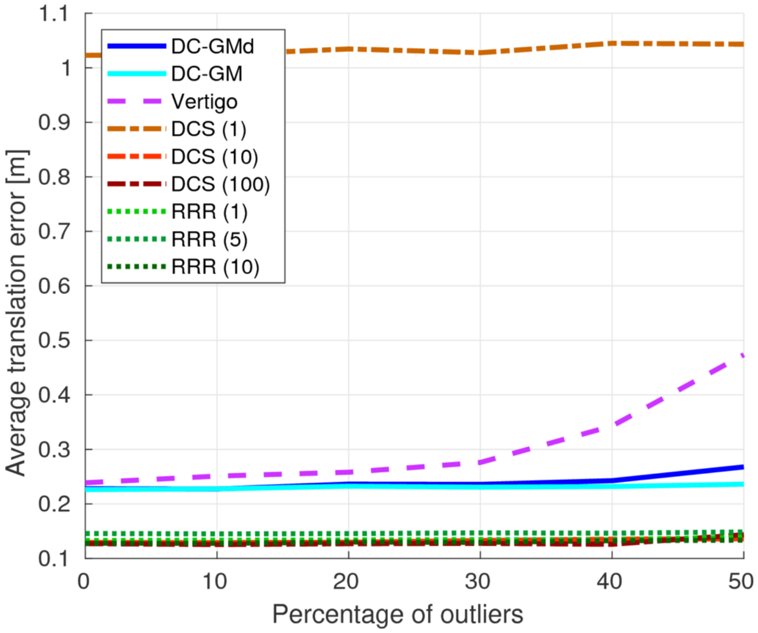
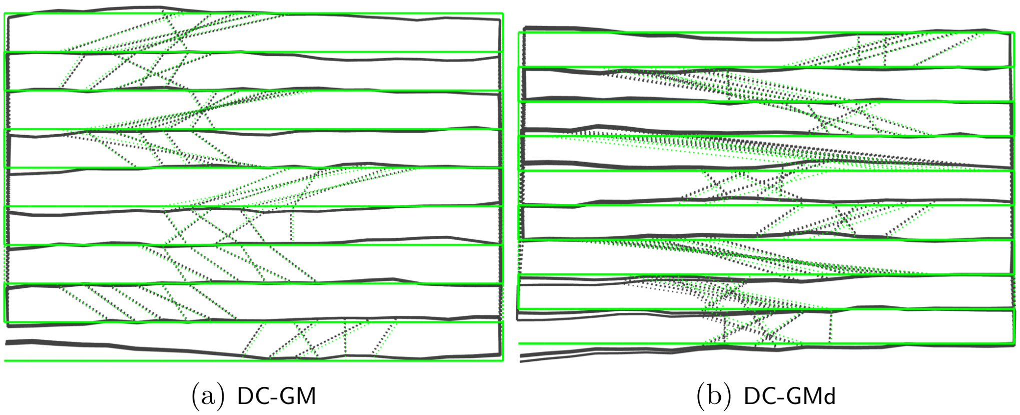


Figure 4.3: Average translation error of the 9 approaches tested in this paper with an increasing percentage of outliers.



(a) DC-GM

(b) DC-GMd

Figure 4.4: Trajectory estimates computed by the proposed techniques (black, overlay of 5 runs) versus ground truth (green) for the simulated grid dataset.

The interested reader can find statistics on the average rank, results for different choices of the thresholds $\bar{c}$ and $\bar{c}^{(ij)}_{(i'j')}$, and additional tests in a simulated Manhattan World in the supplemental material (appendix A).

4.5.2 Experiments On Real Datasets

Methodology. In this section, we consider three real-world standard benchmarking datasets, the CSAIL dataset (1045 poses and 1172 edges), the FR079 dataset (989 poses and 1217 edges), and the FRH dataset (1316 poses and 2820 edges). We spoiled those datasets with 20 randomly grouped outliers. We add correlation terms with $\bar{c}^{(ij)}_{(i'j')} = 1$ for each pair of edges connecting consecutive nodes, e.g., (i, j) and $(i \pm 1, j \pm 1)$. We benchmarked our approach against Vertigo, RRR, and DCS.

Results and Interpretation. Table 4.1 presents the average translation error (computed with respect to the optimized trajectory without outliers) for all datasets and techniques. We also report the average translation error of the odometric estimate. All compared techniques achieve very good results on the FRH dataset. This is probably due to the fact that this dataset provides a very good initial guess, hence the techniques that rely on iterative optimization are favored. This intuition is confirmed by the high accuracy of the odometry. The results on the FR079 dataset are more interesting. In this case, DC-GM and RRR achieve the best results with a slight advantage towards DC-GM. However, Vertigo performs poorly and DCS performance remains worse than the proposed technique even with its best parameter choice. DC-GM has also the best performance on the CSAIL dataset. Again, RRR achieves very good results while Vertigo and DCS have poor performance except for some parameter choice (e.g., DCS performs well for $\Phi = \{100\}$). We attribute this performance boost to the fact that the proposed approach provides a more direct control on the maximum admissible

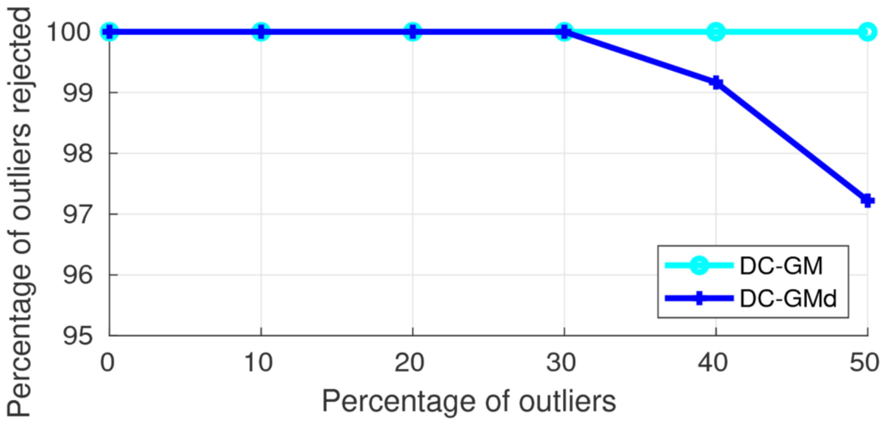
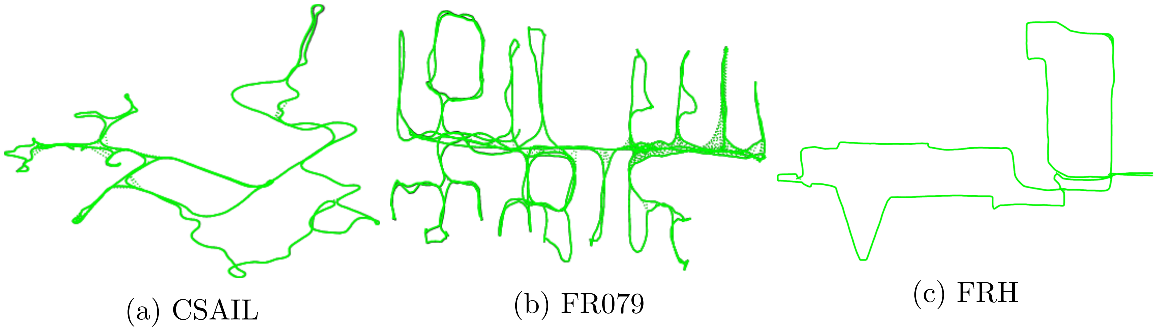


Figure 4.5: Percentage of rejected outliers for the proposed techniques.


(a) CSAIL

(b) FR079

(c) FRH

Figure 4.6: Trajectory estimates computed by DC-GM (black) versus ground truth (green) for the real datasets CSAIL, FR079, and FRH.

Table 4.1: Average translation error (meters) on real benchmarking datasets

	DC-GM	Vertigo	RRR (t_g=1)	RRR (t_g=5)	RRR (t_g=10)
FRH	0.0008	0.0005	0.0004	0.0003	0.0003
FR079	0.0438	0.2751	0.0546	0.0520	0.0521
CSAIL	0.0430	1.4625	0.0495	0.0613	0.0506

	DCS ($\Phi = 1$)	DCS ($\Phi = 10$)	DCS ($\Phi = 100$)	Odometry
FRH	0.0004	0.0004	0.0004	0.0021
FR079	0.2721	0.1804	0.1250	0.2836
CSAIL	1.4576	1.4240	0.0521	1.4480

error of each measurement, while the parameters in DCS and Vertigo have a less clear physical interpretation. This translates to the fact that it is more difficult for DCS and Vertigo to strike a balance between outlier rejection and inlier selection. Therefore, even when these approaches are able to discard most outliers, they may lose accuracy since they also tend to discard good measurements. The difficulty in performing parameter tuning for DCS is confirmed by the fact that the value $\Phi = 1$ (recommended by Agarwal *et al.* [23]) leads to good results on FRH, but fails on FR079 and CSAIL.

Fig. 4.6 shows the trajectory estimates produced by DC-GM for the three real datasets, CSAIL, FR079, and FRH.

4.6 Conclusion

We introduced a discrete-continuous graphical model (DC-GM) to capture perceptual aliasing and outlier correlation in SLAM. Then we developed a semidefinite (SDP) relaxation to perform near-optimal inference in the DC-GM and obtain robust SLAM estimates. Our experiments show that the proposed approach compares favorably with the state of the art while not relying on an initial guess for optimization. Our approach also enables a more intuitive tuning of the parameters (e.g., the maximum admissible residual $\bar{c}$). The supplemental material (appendix A) contains extra results to provide more insights on the performance and limitations of the proposed approach. This paper opens several avenues for future work. First, our Matlab implementation is currently slow: we plan to develop specialized solvers to optimize the SDP relaxations presented in this paper efficiently, leveraging previous work [7]. Second, we plan to extend our testing to 3D SLAM problems: the mathematical formulation in this paper is general, while for numerical reasons we had to limit our tests to relatively small 2D problems. Third, it would be useful to develop incremental solvers that can re-use computation when the measurements are presented to the robot in online (rather than batch) fashion. Finally, it would be interesting to provide a theoretical bound on the number of outliers the proposed technique can tolerate.

4.7 Appendix

This appendix proves Proposition 1 by showing how to reformulate problem (4.10) using the matrix $\boldsymbol{X}$ in (4.11). Let us start by rewriting problem (4.10) and replacing the (scalar) discrete variables $\theta_{ij} \in \{-1, +1\}$ with "binary" selection matrices $\Theta_{ij} \in \{-\boldsymbol{I}_d; +\boldsymbol{I}_d\}$:

$$\begin{aligned}
&\min_{\substack{\boldsymbol{T}_i \in \mathrm{SO}(d)\times\mathbb{R}^d \\ \Theta_{ij}\in\{-\boldsymbol{I}_d;+\boldsymbol{I}_d\}}} \sum_{(i,j)\in\mathcal{E}_{od}} \|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} \\
&-\sum_{(i,j)\in\mathcal{E}_{lc}} \frac{\bar{c}}{2d}\mathrm{tr}\left(\Theta_{ij}\right) - \sum_{(i,j),(i',j')\in\mathcal{P}} \frac{\bar{c}^{(ij)}_{(i'j')}}{d}\mathrm{tr}\left(\Theta_{ij}^{\mathsf{T}}\Theta_{i'j'}\right) \\
&+\sum_{(i,j)\in\mathcal{E}_{lc}} \frac{1}{4}\|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij} + \Theta_{ij}^{\mathsf{T}}(\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij})\|^2_{\boldsymbol{\Omega}}
\end{aligned} \tag{4.14}$$

where we also rearranged the summands. Note the division by d in the second and third sum in (4.14), needed to compensate for the fact that we are now working with $d \times d$ matrices Θ_{ij}.

The first summation in (4.14) can be written as

$$\begin{aligned}
&\sum_{(i,j)\in\mathcal{E}_{od}} \|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} \\
&= \sum_{(i,j)\in\mathcal{E}_{od}} \mathrm{tr}\left((\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij})\boldsymbol{\Omega}(\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij})^{\mathsf{T}}\right) \\
&= \mathrm{tr}\left(\boldsymbol{L}(\mathcal{G}_{od})\boldsymbol{T}^{\mathsf{T}}\boldsymbol{T}\right)
\end{aligned} \tag{4.15}$$

where $\boldsymbol{L}(\mathcal{G}_{od}) \in \mathbb{R}^{(d+1)n\times(d+1)n}$ is the *Connection Laplacian* [7] of the graph $\mathcal{G}_{od} = (\mathcal{V}, \mathcal{E}_{od})$, which has the same set of nodes $\mathcal{V}$ as the original pose graph, but only includes odometric edges $\mathcal{E}_{od}$. We can use a derivation similar to [62] to show that the *Connection Laplacian* of a generic graph $\mathcal{G} = (V, E)$ can be written as

$$\boldsymbol{L}(\mathcal{G}) = \boldsymbol{A}(\mathcal{G})\boldsymbol{\Omega}(\mathcal{G})\boldsymbol{A}(\mathcal{G})^{\mathsf{T}} \tag{4.16}$$

where, the matrices $\boldsymbol{A}(\mathcal{G}) \in \mathbb{R}^{(d+1)|V|\times(d+1)|E|}$ and $\boldsymbol{\Omega}(\mathcal{G}) \in \mathbb{R}^{(d+1)|E|\times(d+1)|E|}$ are given as follows:

$$[\boldsymbol{A}(\mathcal{G})]_{r,e} \doteq \begin{cases} -\bar{\boldsymbol{T}}_{i_e,j_e} & \text{if } r = i_e, \\ +\boldsymbol{I}_{d+1} & \text{if } r = j_e, \qquad \text{for } e = 1,\ldots,|E| \\ \boldsymbol{0}_{d+1} & \text{otherwise.} \end{cases} \tag{4.17}$$

$$\boldsymbol{\Omega}(\mathcal{G}) \doteq \mathrm{blkdiag}(\boldsymbol{\Omega}_1, ..., \boldsymbol{\Omega}_{|E|}) \tag{4.18}$$

The notation $[\boldsymbol{A}(\mathcal{G})]_{r,e}$ denotes the $(d+1)\times(d+1)$ block of $\boldsymbol{A}(\mathcal{G})$ at block row r and block column e, while the e-th edge in E is denoted as (i_e, j_e).

The second summation in (4.14) can be developed as follows:

$$\sum_{(i,j)\in\mathcal{E}_{lc}} -\frac{\bar{c}}{2d}\operatorname{tr}\left(\Theta_{ij}\right) = -\frac{\bar{c}}{2d}\operatorname{tr}\left(\boldsymbol{I}_{d,\ell}^{\mathsf{T}}\Theta\right) \tag{4.19}$$

where $\boldsymbol{I}_{d,\ell} \doteq [\boldsymbol{I}_d \; \ldots \; \boldsymbol{I}_d]$ is a row of ℓ identity matrices.

Similarly, the third summation in (4.14) can be written as:

$$-\sum_{(i,j),(i',j')\in\mathcal{P}} \frac{\bar{c}^{(ij)}_{(i'j')}}{d}\operatorname{tr}\left(\Theta_{ij}^{\mathsf{T}}\Theta_{i'j'}\right) = -\frac{1}{2d}\operatorname{tr}\left(\boldsymbol{N}(\mathcal{C})\Theta^{\mathsf{T}}\Theta\right) \tag{4.20}$$

where $\boldsymbol{N}(\mathcal{C}) \in \mathbb{R}^{d\ell\times d\ell}$ has $d\times d$ blocks in the form:

$$[\boldsymbol{N}(\mathcal{C})]_{e,e'} \doteq \begin{cases} \bar{c}^{(ij)}_{(i'j')}\boldsymbol{I}_d & \text{if } e=(i,j), e'=(i',j')\in\mathcal{C} \\ \bar{c}^{(i'j')}_{(ij)}\boldsymbol{I}_d & \text{if } e=(i',j'), e'=(i,j)\in\mathcal{C} \\ \mathbf{0}_d & \text{otherwise.} \end{cases} \tag{4.21}$$

The first three terms in (4.14) are linear with respect to parts of the matrix $\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}$ in (4.11), so we write the sum of (4.15), (4.19), (4.20) compactly as $\operatorname{tr}\left(\boldsymbol{Q}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\right)$ where

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{L}(\mathcal{G}_{od}) & \mathbf{0}_{(d+1)n,d\ell} & \mathbf{0}_{(d+1)n,d} \\ \mathbf{0}_{d\ell,(d+1)n} & -\frac{1}{2d}\boldsymbol{N}(\mathcal{C}) & -\frac{\bar{c}}{4d}\boldsymbol{I}_{d,\ell}^{\mathsf{T}} \\ \mathbf{0}_{d,(d+1)n} & -\frac{\bar{c}}{4d}\boldsymbol{I}_{d,\ell} & \mathbf{0}_{d,d} \end{bmatrix} \tag{4.22}$$

which is the first term in eq. (4.12). Here $\mathbf{0}_{p,q}$ denotes a zero matrix of size $p\times q$.

In order to complete the proof, we only need to show that the last sum in (4.14) can be written as $\sum_{e=(i,j)\in\mathcal{E}_{lc}}\operatorname{tr}\left(\boldsymbol{U}_e\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{W}_e\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}\right)$, *cf.* (4.12). Towards this goal, we develop each squared norm in the last sum using a derivation similar to (4.15) and get:

$$\begin{aligned} &\|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij} + \Theta_{ij}^{\mathsf{T}}(\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij})\|_{\boldsymbol{\Omega}}^2 \\ &= \operatorname{tr}\left(\boldsymbol{L}(\mathcal{G}_e)\boldsymbol{T}^{\mathsf{T}}\boldsymbol{T}\right) + \operatorname{tr}\left(\boldsymbol{L}(\mathcal{G}_e)(\Theta_{ij}^{\mathsf{T}}\boldsymbol{T})^{\mathsf{T}}(\Theta_{ij}^{\mathsf{T}}\boldsymbol{T})\right) \\ &+ \operatorname{tr}\left(\boldsymbol{L}(\mathcal{G}_e)(\Theta_{ij}^{\mathsf{T}}\boldsymbol{T})^{\mathsf{T}}\boldsymbol{T}\right) + \operatorname{tr}\left(\boldsymbol{L}(\mathcal{G}_e)\boldsymbol{T}^{\mathsf{T}}(\Theta_{ij}^{\mathsf{T}}\boldsymbol{T})\right) \end{aligned} \tag{4.23}$$

where $\mathcal{G}_e = (\mathcal{V},\mathcal{E}_{ij})$ denotes a graph with a single edge $e=(i,j)$. We can write $\boldsymbol{T}$ and $\Theta^{\mathsf{T}}\boldsymbol{T}$

as matrix blocks in $\boldsymbol{X}^\top \boldsymbol{X}$:

$$\begin{aligned} \boldsymbol{T} =& [\mathbf{0}_{d,(d+1)n+d\ell} \;\; \boldsymbol{I}_d] \boldsymbol{X}^\top \boldsymbol{X} \\ & [\boldsymbol{I}_{(d+1)n} \;\; \mathbf{0}_{((d+1)n,d(\ell+1))}]^\top \\ \Theta_{ij}^\top \boldsymbol{T} =& [\mathbf{0}_{d,(d+1)n+d(e-1)} \;\; \boldsymbol{I}_d \;\; \mathbf{0}_{d,d(\ell-e+1)}] \boldsymbol{X}^\top \boldsymbol{X} \\ & [\boldsymbol{I}_{(d+1)n} \;\; \mathbf{0}_{((d+1)n,d(\ell+1))}]^\top \end{aligned} \tag{4.24}$$

which enables to write each squared norm in terms of $\boldsymbol{X}^\top \boldsymbol{X}$ as follows:

$$\|\boldsymbol{T}_j - \boldsymbol{T}_i \bar{\boldsymbol{T}}_{ij} + \Theta_{ij}^\top (\boldsymbol{T}_j - \boldsymbol{T}_i \bar{\boldsymbol{T}}_{ij})\|_{\boldsymbol{\Omega}}^2 = \operatorname{tr}\left(\boldsymbol{U}_e \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{W}_e \boldsymbol{X}^\top \boldsymbol{X} \right) \tag{4.25}$$

where

$$\begin{aligned} \boldsymbol{U}_e =& \begin{bmatrix} \boldsymbol{L}(\mathcal{G}_e) & \mathbf{0}_{(d+1)n,d\ell} & \mathbf{0}_{(d+1)n,d} \\ \mathbf{0}_{d\ell,(d+1)n} & \mathbf{0}_{d\ell,d\ell} & \mathbf{0}_{d\ell,d} \\ \mathbf{0}_{d,(d+1)n} & \mathbf{0}_{d,d\ell} & \mathbf{0}_{d,d} \end{bmatrix} \\ \boldsymbol{W}_e =& [\mathbf{0}_{d,(d+1)n+d(e-1)} \;\; \boldsymbol{I}_d \;\; \mathbf{0}_{d,d(m-e)} \;\; \boldsymbol{I}_d]^\top \\ & [\mathbf{0}_{d,(d+1)n+d(e-1)} \;\; \boldsymbol{I}_d \;\; \mathbf{0}_{d,d(m-e)} \;\; \boldsymbol{I}_d] \end{aligned} \tag{4.26}$$

Summing over all loop-closure edges results in the second term in eq. (4.12), concluding the proof.

CHAPTER 5 ARTICLE 2: DOOR-SLAM: DISTRIBUTED, ONLINE, AND OUTLIER RESILIENT SLAM FOR ROBOTIC TEAMS

Preface:

Full Citation: Pierre-Yves Lajoie[1], Benjamin Ramtoula[1,3], Yun Chang[2], Luca Carlone[2] and Giovanni Beltrame[1], "DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams," *IEEE Robotics and Automation Letters (RA-L)*, [under review] 2019.

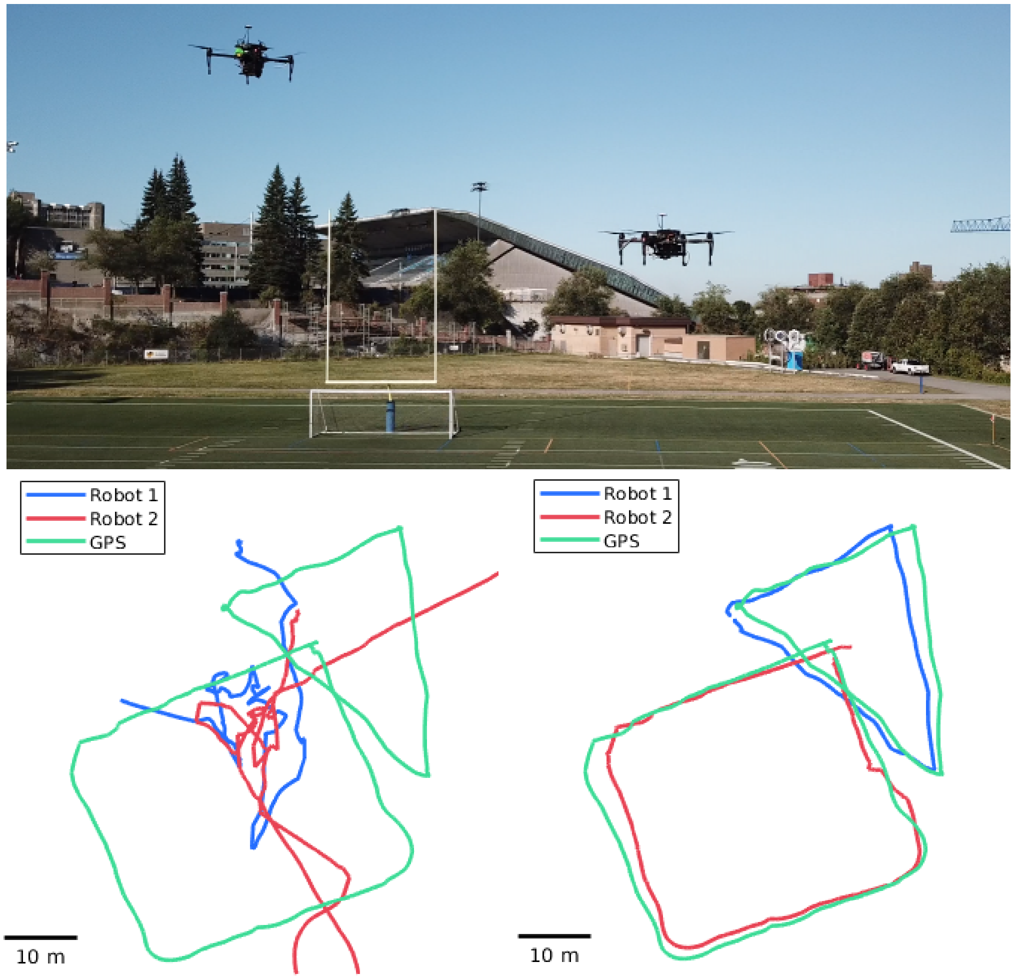
Abstract - To achieve collaborative tasks, robots in a team need to have a shared understanding of the environment and their location within it. Distributed Simultaneous Localization and Mapping (SLAM) offers a practical solution to localize the robots without relying on an external positioning system (e.g. GPS) and with minimal information exchange. Unfortunately, current distributed SLAM systems are vulnerable to perception outliers and therefore tend to use very conservative parameters for inter-robot place recognition. However, being too conservative comes at the cost of rejecting many valid loop closure candidates, which results in less accurate trajectory estimates. This paper introduces DOOR-SLAM, a fully distributed SLAM system with an outlier rejection mechanism that can work with less conservative parameters. DOOR-SLAM is based on peer-to-peer communication and does not require full connectivity among the robots. DOOR-SLAM includes two key modules: a pose graph optimizer combined with a distributed *pairwise consistent measurement set maximization* algorithm to reject spurious inter-robot loop closures; and a distributed SLAM front-end that detects inter-robot loop closures without exchanging raw sensor data. The system has been evaluated in simulations, benchmarking datasets, and field experiments, including tests in GPS-denied subterranean environments. DOOR-SLAM produces more inter-robot loop closures, successfully rejects outliers, and results in accurate trajectory estimates, while requiring low communication bandwidth. Full source code is available at `https://github.com/MISTLab/DOOR-SLAM.git`.

This work was carried out during P. Lajoie's research stay in LIDS, and was partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), the J.A. DeSève Foundation, ARL DCIST CRA W911NF-17-2-0181, and the DARPA "Specification-guided and Capability-aware Autonomy for Long-endurance Situational Awareness in Subterranean Environments" project.

[1] Department of Computer and Software Engineering, Polytechnique Montréal, Montreal, Canada, {pierre-yves.lajoie,benjamin.ramtoula, giovanni.beltrame}@polymtl.ca

[2] Laboratory for Information & Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, USA, {yunchang,lcarlone}@mit.edu

[3] School of Engineering, École Polytechnique Fédérale de Lausanne, Switzerland.



(a) **without** outlier rejection.

(b) **with** outlier rejection.

Figure 5.1: Trajectory estimates from DOOR-SLAM (red and blue) and GPS ground truth (green, only used for benchmarking).

5.1 Introduction

Multi-robot systems already constitute the backbone of many modern robotics applications, from warehouse maintenance to self-driving cars, and have the potential to impact other endeavors, including search & rescue and planetary exploration. These applications involve a team of robots completing a coordinated task in an unknown or partially known environment, and require the robots to have a shared understanding of the environment and their location within it. While a common practice is to circumvent this need by adding external localization infrastructure (e.g., GPS, motion capture, geo-referenced markers), such a solution is not always viable; for instance, when robots are deployed for cave exploration or building inspection, the deployment of an external infrastructure may be dangerous, expensive, or impractical. Therefore, multi-robot SLAM solutions that can work without external localization infrastructure and provide reliable situational awareness are highly desirable.

Obtaining such a shared situational awareness is challenging since the sensor data required for SLAM is distributed across the robots, and communicating raw data may be slow (due to bandwidth constraints) or infeasible (due to limited communication range). For these reasons, current systems either rely on a centralized and offline post-processing step [28], assume all robots are always within communication range [38], or assume centralized pre-processing of the sensor data (e.g., to remove outliers [16]). We believe more flexible solutions are necessary for a broader adoption of multi-robot technologies. For instance, bandwidth issues can be mitigated by relying on local exchange of processed data among the robots to collaboratively compute a SLAM solution.

In addition to the communication constraints, multi-robot SLAM is challenging and prone to failures due to incorrect data association and perceptual aliasing. The latter is particularly problematic since it generates incorrect loop closures between scenes that look similar but correspond to different places. While this topic has received considerable attention in the centralized case [20,21,23,25,28,65], the literature currently lacks *distributed* outlier rejection methods. We believe implementing distributed outlier rejection would improve the robustness of multi-robot systems, allow users to be less conservative during parameters tuning, and enable the detection of more loop closures, improving the accuracy of the SLAM solution.

Contribution. In this system paper, we present DOOR-SLAM, a fully distributed SLAM system for robotic teams. DOOR-SLAM has the following desirable features: (i) it does not require full connectivity maintenance between the robots, (ii) it is able to detect inter-robot loop closures without exchanging raw data, (iii) it performs distributed outlier rejection to remove incorrect inter-robot loop closures, and (iv) it executes a distributed pose graph optimization to retrieve the robots' trajectory estimates.

The proposed system includes two key modules. The first module is a pose graph optimizer that is robust to spurious measurements. We propose an implementation of distributed pose graph optimization along the lines of [16] combined with an outlier rejection mechanism based on [28], that we adapted for online and distributed operation. An example of the robustness afforded by the proposed module is showcased in Fig. 5.1, which reports the trajectory estimates with and without outlier rejection. Our implementation is robust to perceptual aliasing and allows practitioners to use a less conservative tuning of the SLAM front-end. The second module is a data-efficient distributed SLAM front-end. Similar to the recent approach [37], our system uses NetVLAD descriptors [35] for place recognition. However, our approach trades off some data-efficiency to obviate full connectivity maintenance and environment-specific pre-training requirements.

DOOR-SLAM has been evaluated in simulations, benchmarking datasets (KITTI [66]), and field experiments, including tests in GPS-denied subterranean environments. DOOR-SLAM runs online on an NVIDIA Jetson TX2 computer, successfully rejects outliers, and results in accurate trajectory estimates, while requiring a low bandwidth. We release the source code and Docker images for easy reuse of the system components by the community: `https://github.com/MISTLab/DOOR-SLAM.git`.

5.2 Related Work

5.2.1 Distributed Pose Graph Optimization (PGO)

Pose Graph Optimization (PGO) is a popular estimation engine for SLAM. Centralized approaches for multi-robot PGO collect all measurements at a central station, which computes the trajectory estimates for all the robots [8–12]. Since the computation workload and the communication bandwidth of a centralized approach grow with the number of robots, related work has also explored *distributed techniques*, in which robots only exploit local computation and communication. Aragues *et al.* [13] use a distributed Jacobi approach to estimate 2D poses. Cunningham *et al.* [14, 15] use Gaussian elimination. Recent work from Choudhary *et al.* [16] introduces the Distributed Gauss-Seidel approach, which supports 3D cases and avoids the complex bookkeeping and information double counting required by the previous techniques. It requires only to share the latest pose estimates involved in inter-robot measurements. Recent distributed SLAM solutions [37] and [29] have used the implementation of Choudhary *et al.* [16] as back-end for their experiments. While here we focus on PGO, we refer the reader to [16] for an extensive review on other distributed estimation techniques.

5.2.2 Robust PGO

The problem of mitigating the effects of outliers in pose graph optimization has received substantial attention in the literature, due to the dramatic distortion that even one incorrect measurement can cause. Early work in the field includes techniques such as RANSAC [17], branch & bound [18], and M-estimation (see [6, 19] for a review). Sünderhauf *et al.* [21] introduce the idea of outliers deactivation using binary variables that are then relaxed to continuous variables. Agarwal *et al.* [23] build on top of this idea to dynamically scale the measurement covariances. Other works on the single robot case include Olson and Agarwal [20] and Pfingsthorn and Birk [56, 57] which consider multi-modal distributions for the noise. Recent work from Lajoie *et al.* [65] and Carlone and Calafiore [46] focus on robust global solvers based on convex relaxations. Instead of classifying the measurements individually, Latif *et al.* [25], Carlone *et al.* [26], Graham *et al.* [27] look for sets of mutually consistent measurements. Mangelson *et al.* [28] extend the latter idea to the multi-robot case and propose an effective graph-theoretic technique to find pairwise-consistent measurements among the inter-robot loop closures. Alternatives for multi-robot cases include Dong *et al.* [12] which search for consistent inter-robot measurements using expectation maximization. Wang *et al.* [29] leverage extra information from wireless channels to detect outliers during a multi-robot rendezvous.

5.2.3 Distributed Loop Closure Detection

Inter-robot loop closures are critical to align the trajectories of the robots in a common reference frame and to improve the trajectory estimates. In a centralized setup, a common way to obtain loop closures is to use visual place recognition methods, which compare compact image descriptors to find potential loop closures. This is traditionally done with global visual features [30, 31], or local visual features [2, 32] which can be quantized in a bag-of-word model [33]. More recently, convolutional neural networks (CNN), either using features trained on auxiliary tasks [34] or directly trained end-to-end for place recognition, such as NetVLAD [35], have generated more robust descriptors. Geometric verification using local features is then used to validate putative loop closures and estimate transformations between the corresponding observation poses [67, 68].

Distributed loop closure detection has the additional challenge that the images are not collected at a single location and their exchange is problematic due to range and bandwidth constraints. Tardioli *et al.* [36] use visual vocabulary indexes instead of descriptors to reduce the required bandwidth. Cieslewski and Scaramuzza [37] propose distributed and scalable solutions for place recognition in a fully connected team of robots. A first approach [38] relies on bag-of-words of visual features [33] which are split and distributed among the team. Another one [39] pre-assigns a range of descriptors from NetVLAD to each robot, allowing place recognition search over the full team by communicating with a single other robot. These methods minimize the required bandwidth and scale well with the number of robots, but are designed for situations with full connectivity in the team. Tian *et al.* [40, 41] and Giamou *et al.* [42] propose complementary approaches to these methods. They consider robots having rendezvous and efficiently coordinate the data exchange during the geometric verification step, accounting for the available communication and computation resources.

5.3 The DOOR-SLAM System

Our distributed SLAM system relies on peer-to-peer communication: each robot performs single-robot SLAM when there is no teammate within communication range, and executes a distributed SLAM protocol during a rendezvous.

Our implementation leverages Buzz [69], a programming language specifically designed for multi-robot systems. Buzz offers useful primitives to build a fully decentralized software architecture, and seamlessly handles the transition between single-robot and multi-robot execution. Buzz is a scripting language that lets us abstract away the details concerning communication, neighbor detection and management, and provides a uniform framework to implement and compare multi-robot algorithms (such as SLAM, task allocation, exploration, etc.). It provides a uniform gossip-based interface, implemented on WiFi, Xbee, Bluetooth, or custom networking devices. Buzz is thought of as an *extension* language, i.e. it is designed to be laid on top of other frameworks, such as the Robot Operating System (ROS). This allows us to run DOOR-SLAM on virtually any type and *any number* of robots that support ROS without modification. Experiments [69] show that Buzz can scale up to thousands of robots.

A system overview of DOOR-SLAM is given in Fig. 5.2. Each robot collects images from an onboard stereo camera and uses a (single-robot) Stereo Visual Odometry module to produce an estimate of its trajectory. In our implementation, we use the stereo odometry from RTAB-Map [70]. The images are also fed to the Distributed Loop Closure Detection module (Section 5.3.1) which communicates information with other robots (when they are within communication range) and outputs inter-robot loop closure measurements. Then, the Distributed Outlier Rejection module (Section 5.3.2) collects the odometry and inter-robot measurements to compute the maximal set of pairwise consistent measurements and filters out the outliers. Finally, the Distributed Pose Graph Optimization module (Section 5.3.2) performs distributed SLAM. For simplicity, in the current implementation, we only consider inter-robot loop closures [16] (i.e., loop closures involving poses of different robots). The system can be easily extended to use intra-robot loop closures (i.e., the loop closures commonly encountered in single-robot SLAM) by replacing stereo odometry [70] with a visual SLAM solution.

In the following sections, we focus on the distributed place recognition module and on the distributed robust PGO module, while we refer the reader to [70] for a description of the stereo visual odometry module.

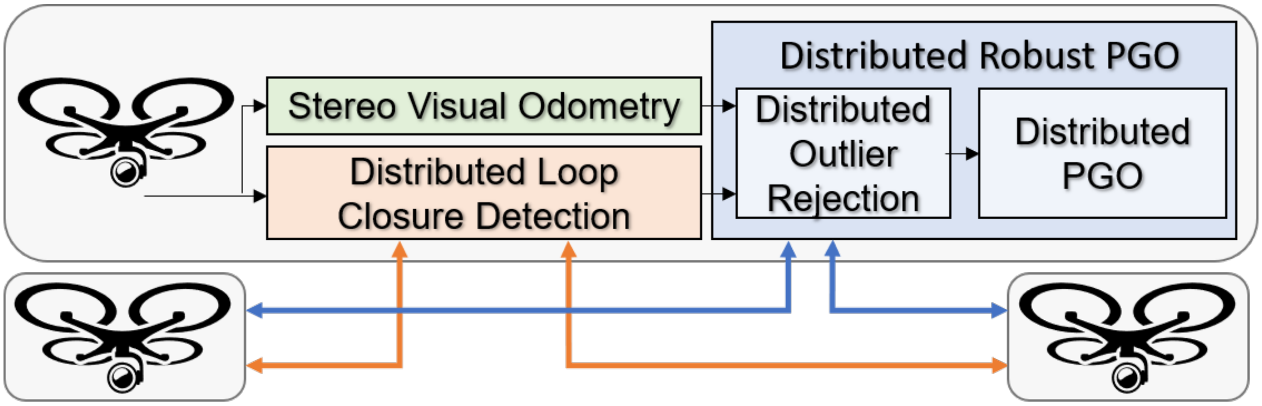

Stereo Visual Odometry
Distributed Loop Closure Detection
Distributed Robust PGO
Distributed Outlier Rejection
Distributed PGO

Figure 5.2: DOOR-SLAM system overview

5.3.1 Distributed Loop Closure Detection

The distributed loop closure detection includes two submodules. The first submodule, *place recognition*, allows to find loop closure candidates using compact image descriptors. The second submodule, *geometric verification*, computes the relative pose estimate between two robot poses observing the same scene. The process is illustrated in Fig. 5.3.

The **place recognition submodule** relies on NetVLAD descriptors [35] which are compact and robust to viewpoint and illumination changes. Each robot locally computes the NetVLAD descriptors for each keyframe provided by the stereo visual odometry module. Once two robots (α and β) are in communication range, one of them (α) sends NetVLAD descriptors to the other (β). Robot α only sends the descriptors which have been generated since both robots' last encounter or all of them if it is their first rendezvous. Robot β compares the received NetVLAD descriptors against the ones it has generated from its own keyframes. By doing so, robot β selects potential loop closures corresponding to pairs of keyframes having Euclidean distance below a given threshold. This process provides putative loop closures without requiring the exchange of raw data, full connectivity maintenance, or additional environment-specific pre-training.

Each robot also extracts visual features from the left image of the stereo pair, the associated feature descriptors, and their corresponding estimated 3D positions; these are used by the **geometric verification submodule**. After finding a set of putative loop closures, robot β sends the visual features, along with their descriptors and 3D positions, back to robot α. This is done for each keyframe involved in a putative loop closure. Using these features, robot α performs geometric verification using the solvePnpRansac function from OpenCV [71], which returns a set of inlier features and a relative pose transformation. If the set of inliers is sufficiently large (see Section 5.4), robot α considers the corresponding loop closure successful. Finally, robot α communicates back the relative poses corresponding to successful loop closures to robot β. Once the inter-robot loop closures are found and shared, both robots initiate the distributed robust pose graph optimization protocol described in the following section.

5.3.2 Distributed Robust PGO

This module is in charge of estimating the robots' trajectories given the odometry measurements from the stereo visual odometry module and the relative pose measurements from the distributed loop closure detection module. The module also includes a distributed outlier rejection approach that removes spurious loop closures that may accidentally pass the geometric verification step described in Section 5.3.1.

The (to-be-computed) trajectory of each robot is represented as a discrete set of poses, describing the position and the orientation of its camera at each keyframe. We denote the trajectory of robot α as $\boldsymbol{x}_\alpha \doteq [\boldsymbol{x}_{\alpha_0}, \boldsymbol{x}_{\alpha_1}, ...]$, where $x_{\alpha_i} = [\boldsymbol{R}_{\alpha_i}, \boldsymbol{t}_{\alpha_i}] \in \mathrm{SE}(3)$, and $\boldsymbol{R}_{\alpha_i} \in \mathrm{SO}(3)$ and $\boldsymbol{t}_{\alpha_i} \in \mathbb{R}^3$ represent the rotation and the translation of the pose associated to the i-th keyframe of robot α.

The stereo visual odometry module produces odometry measurements, describing the relative pose between consecutive keyframes: for instance, $\bar{\boldsymbol{z}}^{\alpha_{i-1}}_{\alpha_i} \doteq [\bar{\boldsymbol{R}}^{\alpha_{i-1}}_{\alpha_i}, \bar{\boldsymbol{t}}^{\alpha_{i-1}}_{\alpha_i}]$, denotes the (measured) motion of robot α between keyframe $i-1$ and keyframe i. On the other hand, the distributed loop closure detection module produces noisy relative pose measurements of the relative pose of two robots observing the same place: for instance, the inter-robot measurement $\bar{\boldsymbol{z}}^{\alpha_i}_{\beta_k} \doteq [\bar{\boldsymbol{R}}^{\alpha_i}_{\beta_k}, \bar{\boldsymbol{t}}^{\alpha_i}_{\beta_k}]$ describes a measurement of the relative pose between the i-th keyframe of robot α and the k-th keyframe of robot β.

Our system includes two submodules: distributed outlier rejection and distributed pose graph optimization.

The **distributed outlier rejection submodule** rejects spurious inter-robot loop closures $\bar{\boldsymbol{z}}^{\alpha_i}_{\beta_k}$ that may be caused by perceptual aliasing; if undetected, these outliers cause large distortions in the robot trajectory estimates (Fig. 5.1).

We adopt the *Pairwise Consistent Measurement Set Maximization* (PCM) technique proposed by Mangelson *et al.* [28] for outlier rejection and tailor it to a fully distributed setup. The key insight behind PCM is to check if pairs of inter-robot loop closures are consistent with each other and then search for a large set of mutually-consistent loop closures (as shown in [28], the largest set of pairwise consistent measurements can be found as a maximum clique). Although PCM does not check for the joint consistency of all the measurements, the approach typically ensures that gross outliers are rejected. The following metric is used to determine

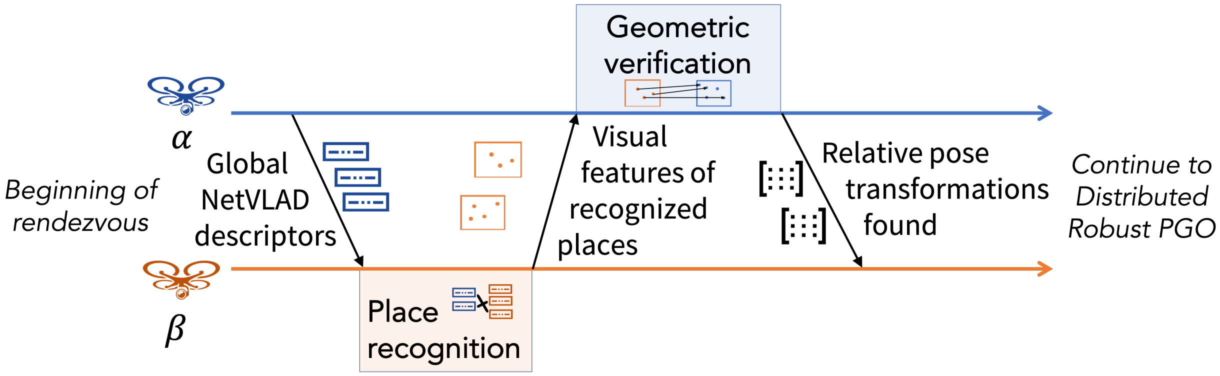

Geometric verification
α
β
Beginning of rendezvous
Global NetVLAD descriptors
Place recognition
Visual features of recognized places
Relative pose transformations found
Continue to Distributed Robust PGO

Figure 5.3: Distributed loop closures detection overview.

if two inter-robot loop closures $\bar{\boldsymbol{z}}^{\alpha_j}_{\beta_k}$ and $\bar{\boldsymbol{z}}^{\alpha_i}_{\beta_l}$ are pairwise consistent:

$$\|(\bar{\boldsymbol{z}}^{\alpha_i}_{\alpha_j} \oplus \bar{\boldsymbol{z}}^{\alpha_j}_{\beta_k} \oplus \bar{\boldsymbol{z}}^{\beta_k}_{\beta_l}) \ominus \bar{\boldsymbol{z}}^{\alpha_i}_{\beta_l}\|_\Sigma \leq \gamma \tag{5.1}$$

In this equation, $\|\cdot\|_\Sigma$ represents the Mahalanobis distance and we use the notation of [72] to denote the pose composition $\oplus$ and inversion $\ominus$. Intuitively, in the noiseless case, measurements along the cycle (shown in green in Fig. 5.4) formed by the loop closures $(\bar{\boldsymbol{z}}^{\alpha_i}_{\beta_l}, \bar{\boldsymbol{z}}^{\alpha_j}_{\beta_k})$ and the odometry $(\bar{\boldsymbol{z}}^{\alpha_i}_{\alpha_j}, \bar{\boldsymbol{z}}^{\beta_k}_{\beta_l})$ must compose to the identity, and the consistency metric (5.1) assesses that the noise accumulated along the cycle is consistent with the noise covariance Σ. The PCM likelihood threshold γ can be determined from the quantile of the chi-squared distribution for a given probability level [28].

The key insight of this section is that the consistency metric (5.1) can be computed from the loop closure measurements $(\bar{\boldsymbol{z}}^{\alpha_i}_{\beta_l}, \bar{\boldsymbol{z}}^{\alpha_j}_{\beta_k})$ and the odometric estimates of the poses involved $(x_{\alpha_i}, x_{\alpha_j}, x_{\beta_l}, x_{\beta_k})$. Since both quantities are already used in the distributed PGO algorithm (described below), the outlier rejection can be performed "for free", without requiring extra communication. After the pairwise consistency checks are performed, each robot computes the maximum clique of the measurements for each of its neighbors to find inlier loop closures. The inliers are passed to the distributed PGO.

The **distributed PGO submodule** uses the odometry measurements and the inlier inter-robot loop closures to compute the trajectory estimates of the robots. We use the approach proposed in [16]: the robots repeatedly exchange their estimate for the poses involved in inter-robot loop closures till they reach a consensus on the optimal trajectory estimate. More specifically, the approach of [16] solves pose graph optimization in a distributed fashion using a two-stage approach: first, it computes an estimate for the rotations of the robots along their trajectories; and then it recovers the full poses in a second stage. Each stage can be solved using a distributed Gauss-Seidel algorithm [16] which avoids complex bookkeeping

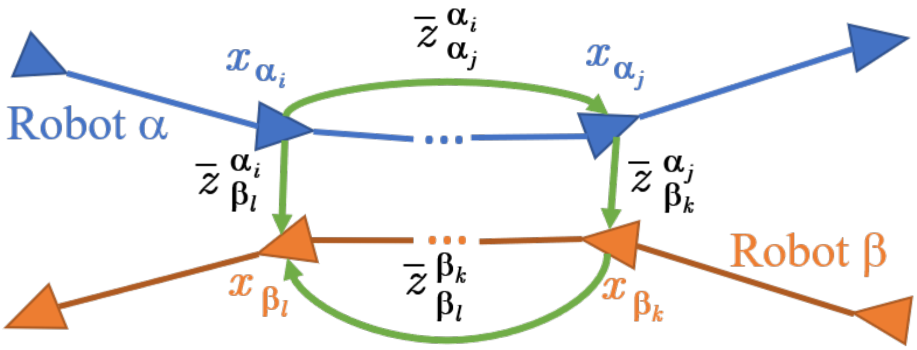

Robot α
x_{α_i}
$\bar{z}^{\alpha_i}_{\alpha_j}$
x_{α_j}
$\bar{z}^{\alpha_i}_{\beta_l}$
$\bar{z}^{\alpha_j}_{\beta_k}$
Robot β
x_{β_l}
$\bar{z}^{\beta_k}_{\beta_l}$
x_{β_k}

Figure 5.4: Measurements needed to check pairwise consistency.

and information double counting, and requires minimal information exchange.

5.4 Experimental Results

This section presents four sets of experiments. Section 5.4.2 tests the performance of the outlier rejection mechanism in a simulated multi-robot SLAM environment. Section 5.4.3 evaluates the results of DOOR-SLAM on the widely used KITTI00 sequence [66]. Section 5.4.4 reports the results of field experiments conducted with two flying drones on an outdoor football field. Finally, Section 5.4.5 reports the results of field tests conducted in underground environments in the context of the DARPA Subterranean Challenge [73].

5.4.1 Implementation Details

The DOOR-SLAM system is the result of the combination of many frameworks and libraries. First, we use the Robot Operating System to interface with the onboard camera and handle information exchange between the different core modules. We use the Buzz [69] programming language and runtime environment for communication and scheduling. In the front-end, we use the latest version of RTAB-Map [70] for stereo visual odometry and we use the tensorflow implementation of NetVLAD provided in [37], with the default neural network weights trained in the original paper [35]. We only keep the first 128 dimensions of the generated descriptors to limit the data to be exchanged, as done in [37]. The visual feature extraction and relative pose transformation estimation are done by adapting the implementation in RTAB-Map and keeping their default parameters. The features used are Good Features to Track [74] with ORB descriptors [3]. We implemented the distributed robust PGO module in C++ using the GTSAM library [75] and building on the implementation of Choudhary et al. [16]. We followed a simulation, software-in-the-loop, hardware-in-the-loop, robot deployment code base implementation paradigm, starting from ARGoS simulation and ending with full deployment using Docker containers on NVIDIA Jetson TX2 on-board computers.

5.4.2 Simulation Experiments

To verify that our online and distributed implementation of PCM is able to correctly reject outliers, we designed a simulation using ARGoS [76]. We refer the reader to the video attachment for a visualization. We use 5 drones with limited communication range following random trajectories. We simulate the SLAM front-end by building their respective pose graphs using noisy measurements. When two robots come within communication range, they exchange inter-robot measurements based on their current poses and then use our SLAM back-end (PCM + distributed PGO) to compute a shared pose graph solution in a fully distributed manner. Inlier inter-robot loop closures are added with realistic Gaussian noise ($\sigma_R = 0.01$rad and $\sigma_t = 0.1$m for rotation and translation measurements, respectively) while outliers are sampled from a uniform distribution.

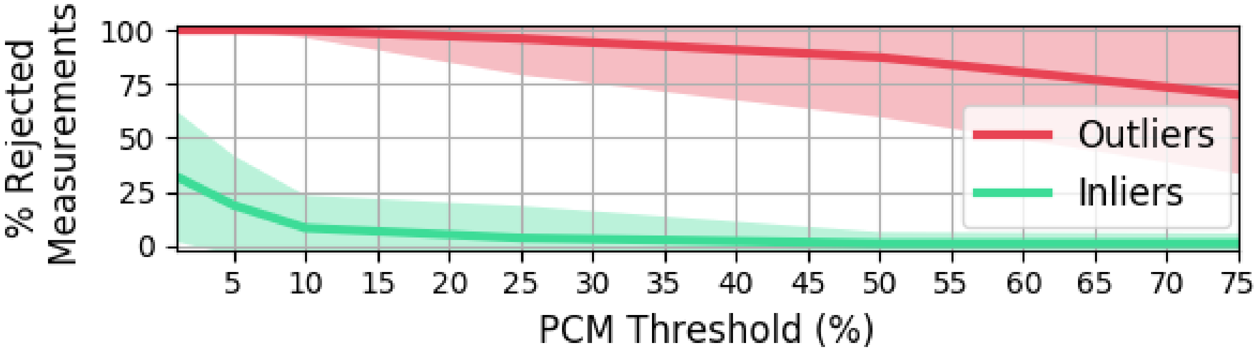


Figure 5.5: Percentage of inliers and outliers rejected w.r.t. PCM likelihood threshold (100 runs avg. ± std.) in ARGoS.

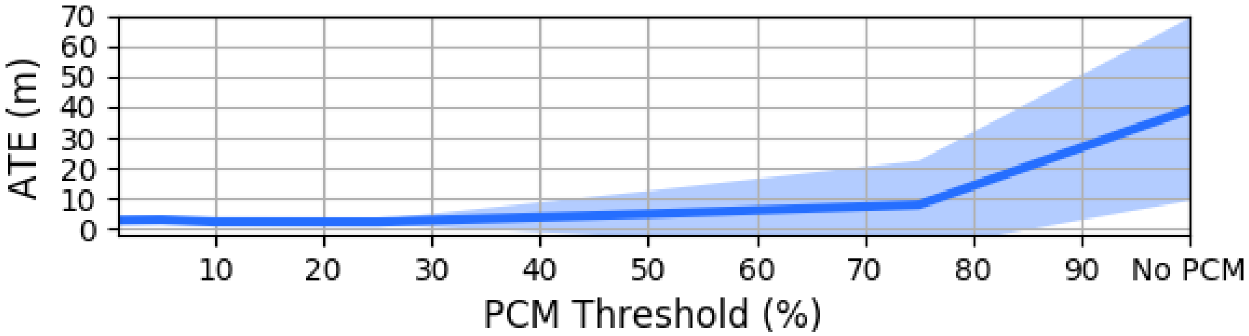


Figure 5.6: Average Translation Error (ATE) w.r.t. PCM likelihood threshold (10 runs avg. ± std.) in ARGoS.

Results. We look at three metrics in particular: the percentage of outliers rejected, the percentage of inliers rejected and the average translation error (ATE). The first evaluates if the spurious measurements are successfully rejected; the ideal value for this metric is 100%. The second indicates if the technique is needlessly rejecting valid measurements; the ideal value is 0%. The third evaluates the distortion of the estimates. Fig. 5.5 shows the percentage of outliers (in red) and inliers (in green) rejected with different PCM thresholds while Fig. 5.6 shows the ATE (in blue); the threshold represents the likelihood of accepting an outlier as inlier. As expected, using a lower threshold leads to the rejection of more measurements, including inliers, while using a higher threshold can lead to the occasional acceptance of outliers which in turn leads to a larger error. Therefore, in all our experiments, we used a threshold of 1% to showcase the performance of our system in its safest configuration.

5.4.3 Dataset Experiments

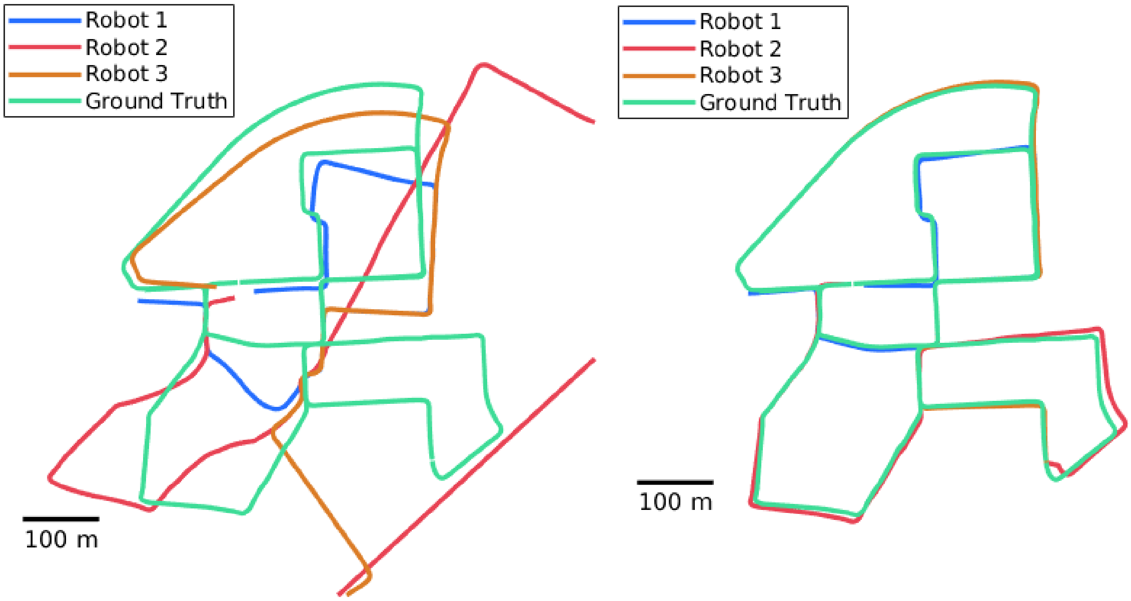
The KITTI00 [66] sequence is a popular benchmark for SLAM. In our evaluation, we split the sequence into three parts and execute DOOR-SLAM on three NVIDIA Jetson TX2s. We used a PCM threshold of 1%, a NetVLAD comparison threshold of 0.15, and a minimum of 5 feature correspondences in the geometric verification to get a high number of loop closure measurements. While related work uses more conservative thresholds for NetVLAD and the number of feature correspondences to avoid outliers [37], we can afford more aggressive thresholds thanks to PCM.

Results. Fig. 5.7 shows that outliers are present among the loop closure measurements and that their effect on the pose graph is significant. The average translation error (ATE) without outlier rejection is 86.85m, while the error is reduced to 8.00m when using PCM. It is important to note that the error is higher than recent SLAM solutions on this sequence since for simplicity's sake we do not make use of any *intra-robot* loop closures. Additionnal results on other KITTI sequences are available in the supplemental material [**?**].

5.4.4 Field Tests with Drones

To test that DOOR-SLAM can overcome the reality gap and map environments with severe perceptual aliasing using resource-constrained platforms, we also performed field experiments with two quadcopters featuring stereo cameras, flying over a football field. The cameras facing slightly downward are subject to perceptual aliasing, due to the repetitive appearance of the field (see video attachment). The hardware setup is described in Fig. 5.8.

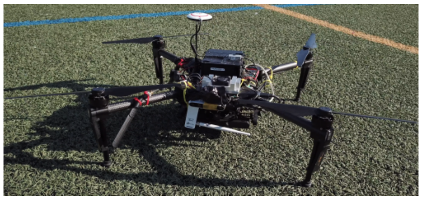
We performed manual flights with trajectories approximately following simple geometric shapes as seen in Fig. 5.1. For the first experiments we recorded images and GPS data on the field and we executed DOOR-SLAM in an offline fashion on two NVIDIA Jetson TX2 con-



(a) **without** outlier rejection.

(b) **with** outlier rejection.

Figure 5.7: Experiment on the KITTI00 dataset. Optimized trajectories (red, blue, and orange) and ground truth (green).



Platform	DJI Matrice 100
Camera	Intel Realsense D435
Computer	NVIDIA Jetson TX2

Figure 5.8: Hardware setup used in field experiments.

nected through WiFi. This allowed us to reuse the same recordings with various combinations of the three major parameters of DOOR-SLAM and study their influence (Section 5.4.4) as well as assess DOOR-SLAM's communication requirements (Section 5.4.4). Finally, we performed an online experiment where DOOR-SLAM is executed on the drones' onboard computers during flight (see Section 5.4.4 and video attachment).

Influence of Parameters

As practitioners know, SLAM systems often rely on precise parameter tuning, especially to avoid outlier measurements from the front-end. We show that DOOR-SLAM is less sensitive to the parameter tuning since our back-end can handle spurious measurements. Moreover, we can leverage the robustness to outliers to significantly increase the number of loop closure candidates and potentially the number of valid measurements.

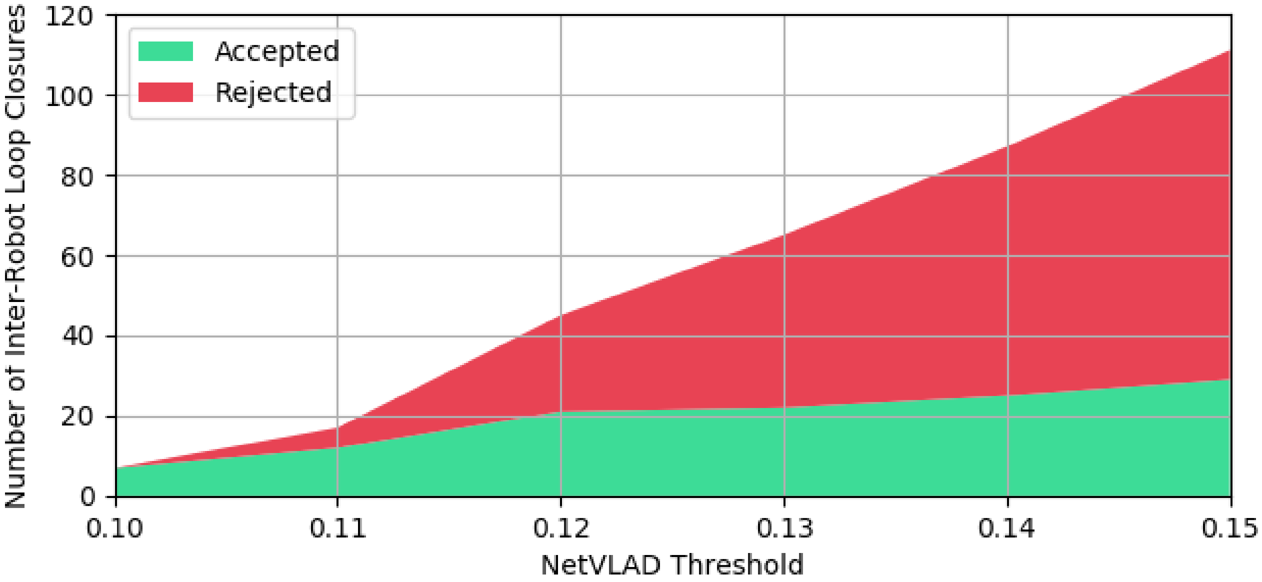


Figure 5.9: Number of inter-robot loop closures accepted and rejected by PCM w.r.t. the NetVLAD threshold. We fix the minimum number of feature correspondences to 5.

Results. In many scenarios, loop closures are hard to obtain due to external conditions such as illumination changes. Hence, it is important to consider as many loop closure candidates as possible. Instead of rejecting them prematurely in the front-end, DOOR-SLAM can consider more candidates and only reject the outliers before the optimization. To analyze the gain of being less conservative, we looked at the number of inter-robot loop closures detected with various NetVLAD thresholds (Fig. 5.9). As expected, when we increase this threshold, we obtain more candidates. Interestingly, even though most of the new loop closures are rejected by PCM (in red), we also get about three times more valid measurements (green) when using a looser threshold (0.15) as opposed to a more conservative one (0.10). Therefore, the use of less stringent thresholds allows adding valid measurements to the pose graph, enhancing the trajectory estimation accuracy.

Similarly, reducing the minimum number of feature correspondences that need to pass the geometric verification step for a loop closure to be considered successful leads to more loop closure candidates. RTAB-Map uses a default of 20 correspondences. As shown in Fig. 5.10, we can double the number of valid inter-robot loop closures when reducing the number of correspondences to 4 or 5.

Table 5.1: Effect of the PCM threshold on the accuracy.

Threshold (%)	1	10	25	75	No PCM
ATE (m)	2.1930	2.3185	3.1461	18.255	22.0159

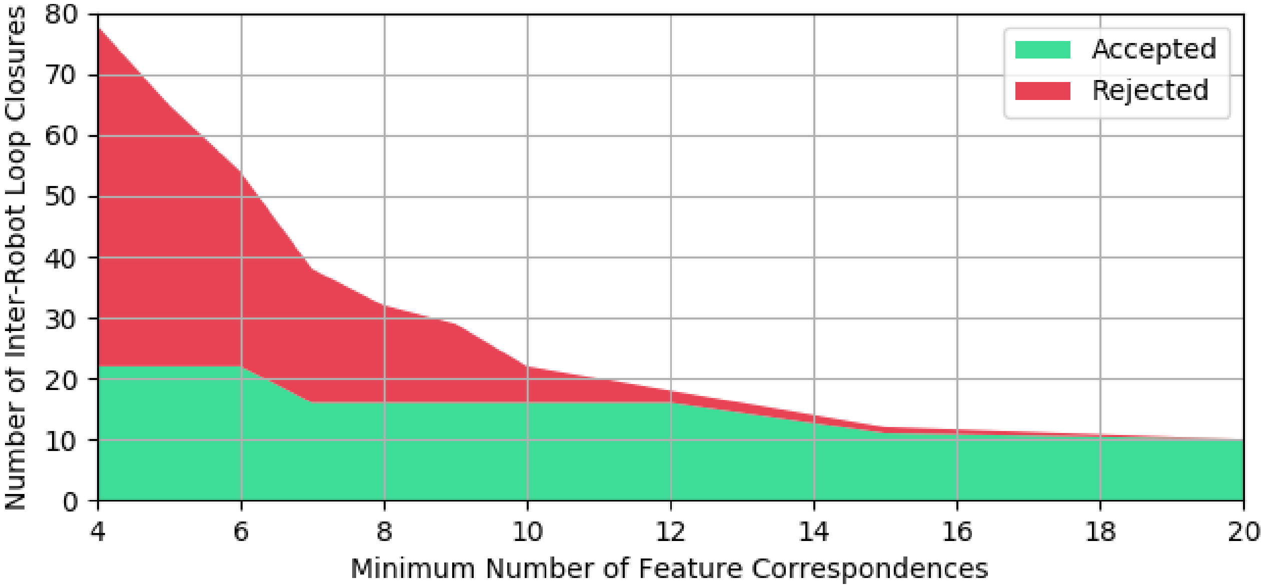


Figure 5.10: Number of inter-robot loop closures accepted and rejected by PCM w.r.t. the minimum number of feature correspondences to consider geometric verification successful. We fix the NetVLAD threshold to 0.13

The last parameter we analyzed is the PCM likelihood threshold to reject outliers. As seen in Section 5.4.2, a lower threshold leads to the rejection of more measurements, including inliers. However, since we are mapping a relatively small environment, we get many loop closures linking the same places. Therefore, as long as we do not disconnect the recognized places in the pose graph, a lower PCM threshold has the benefit of filtering out the noisiest loop closures and keeping the more precise ones. We can see in Table 5.1 that the resulting trajectories are affected by the noisier loop closures when we use a higher threshold, but that we still avoid the dramatic distortion caused by outliers seen in Fig. 5.1. Indeed, the average translation error (ATE) compared to the GPS ground truth is the lowest when we use the most conservative PCM threshold (i.e. 1%), for which we show the visual result in Fig. 5.1. On the other hand, we can see a large increase in the error when we use a threshold larger than 75% or no PCM, which indicates that outliers have not been rejected.

In light of those results, DOOR-SLAM can use less conservative parameters in the front-end to obtain more loop closure candidates and a more conservative PCM threshold to keep only the most accurate ones. This combination leads to a larger number valid loop closures and to more accurate trajectory estimates.

Table 5.2: Data sizes of messages sent.

Details of message sent for each		Avg. Size (kB) ± Std.
Keyframe	NetVLAD descriptor	1.00 ± 0.00
	RGB image	*900.04 ± 0.00*
NetVLAD **match**	Keypoints Information	34.51 ± 0.68
	Keypoints Descriptors	25.00 ± 0.49
	Grayscale images	*600.06 ± 0.0*
Inter-robot loop closure	Pose Estimate	0.34 ± 0.00
	Loop Closure Measurement	0.34 ± 0.00

Communication

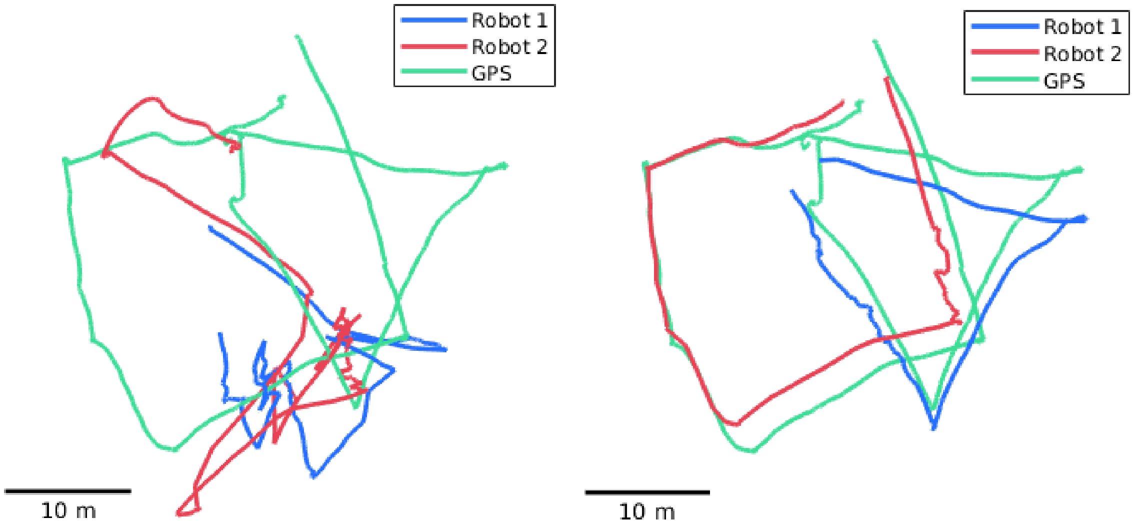
As described in Section 5.3.1, the distributed loop closure detection module needs to share information between the robots about each keyframe to detect loop closure candidates. When a NetVLAD match occurs, the module needs to send the keypoint information for each matching keyframe. If there are enough feature correspondences, the module can compute the relative pose transformation and send the resulting inter-robot measurement to the other robot. Here we evaluate the communication cost of the proposed distributed front-end.

Results. Table 5.2 reports the average data size sent at each keyframe. These averages were computed during our field experiments. For comparison, we also report (in gray) the size of the messages sent in case the robots were to directly transmit camera images. We see that the proposed front-end reduces the required bandwidth by roughly a factor of 10.

Online Experiments

We tested DOOR-SLAM online with two quadcopters. The main challenge of performing live experiments with DOOR-SLAM on the NVIDIA Jetson TX2 platforms is to run every module in real-time with the additional workload of the camera driver and the connection to the flight controller. To achieve this feat, we limited the frame rate of the onboard camera to 6Hz. Modules such as the stereo odometry or the Tensorflow implementation of NetVLAD were particularly demanding in terms of RAM which required us to add 4GB of swap space to the 8GB initially available. We also tuned some visual odometry parameters to gain computational performance at the cost of losing some accuracy.

Results. Fig. 5.11 reports the trajectory estimates of our online experiments, compared with the trajectories from GPS. We performed this experiment with a PCM threshold of 1%, a NetVLAD threshold of 0.13, and a minimum of 5 inliers for geometric verification. Although we note a degradation of the visual odometry accuracy, the results in Fig. 5.11 are consistent



(a) **without** outlier rejection.

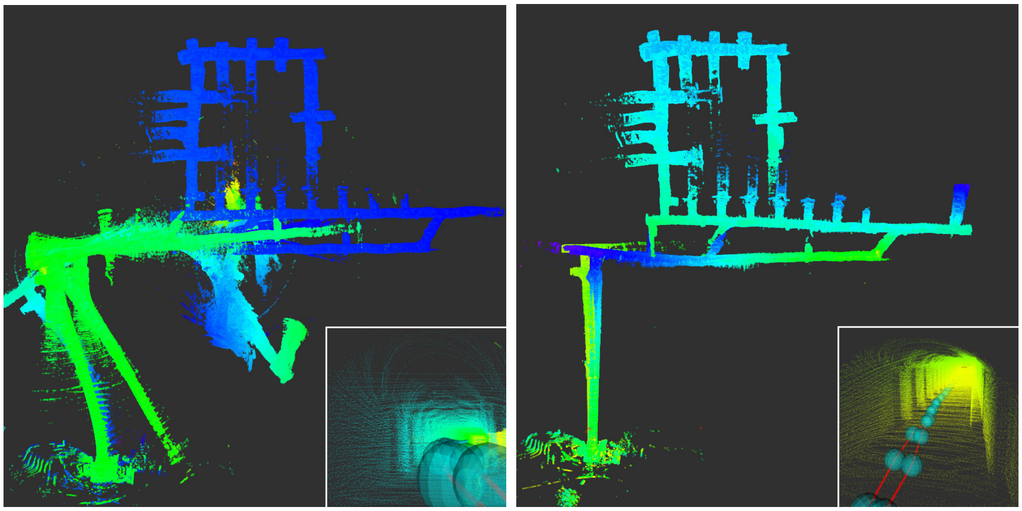
(b) **with** outlier rejection.

Figure 5.11: Online Trajectory estimates from DOOR-SLAM (red and blue) and GPS ground truth (green, only used for benchmarking).

with the ones observed in Fig. 5.1.

5.4.5 Field Tests in Subterranean Environments

To remark on the generality of the DOOR-SLAM back-end, this section considers a different sensor front-end and shows that DOOR-SLAM can be used in a lidar-based SLAM setup with minimal modifications. For this purpose we used lidar data collected by two Husky UGVs during the Tunnel Circuit competition of the DARPA Subterranean Challenge [73]. The data is collected with the VLP-16 Puck LITE 3D lidar and the loop closures are detected by scan matching using ICP. The environment, over 1 kilometer long, is a coal mine whose self-similar appearance is prone to causing perceptual aliasing and outliers. Fig. 5.12 shows the effect of using PCM: the left figure shows a top-view of the point cloud resulting from multi-robot SLAM without PCM, while the figure on the right is produced using PCM with a threshold of 1%. The reader may notice the deformation on the left figure, caused by an incorrect loop closure between two different segments of the tunnel. Although PCM largely improves the mapping performance, we notice that there is still an incorrect loop closure on the right figure. This kind of error is likely due to the fact that PCM requires a correct estimate of the measurement covariances which is not always available. To compute the trajectory estimates, our distributed back-end required the transmission of 92.27kB, while in a centralized setup the transmission of the initial pose graph data and the resulting estimates from one robot to the other would require 196.30kB. In summary, our distributed back-end implementation roughly halves the communication burden.



(a) **without** outlier rejection. (b) **with** outlier rejection.

Figure 5.12: Lidar-based multi-robot SLAM experiment during the DARPA Subterranean Challenge.

5.5 Conclusion

We present DOOR-SLAM, a system for distributed multi-robot SLAM consisting of a data-efficient peer-to-peer front-end and an outlier-resilient back-end. Our experiments in simulation, datasets, and field tests show that our approach rejects spurious measurements and computes accurate trajectory estimates. We also show that our approach can leverage its robust back-end to work with less conservative front-end parameters. In future work, we plan to explore not only the robustness to additional perception failures, such as large groups of correlated outliers, but also the robustness to communication issues (i.e., packet drop) to improve the safety and resilience of multi-robot SLAM systems.

CHAPTER 6 GENERAL DISCUSSION

This chapter presents a joint discussion of the different results presented in the previous sections. It aims to recapitulate the main findings and to discuss the impact of those contributions on the research community. First we discuss the results regarding the robustness to outliers, then we address our results for multi-robot SLAM.

6.1 Robustness to Outliers

Chapter 4 presents a PGO formulation (DC-GM) robust to outliers that includes the modeling of the perceptual aliasing phenomenon. The article also shows how to adapt the formulation to be solved by global solver using convex relaxations. This work is mostly theoretical in the sense that it provides a novel formulation, but that is essentially a proof of concept since the implementation using the general-purpose convex solver CVX is quite slow and could not realistically be used onboard a robot. Nonetheless, the results in Chapter 4 show that the DC-GM technique compares favorably to the other current state-of-the-art techniques. Moreover, the technique does not require an initial guess and its main parameter is easier to tune than the ones of the other techniques since it has a clear physical meaning which is based on the expected level of noise in the measurements. The experiments on real-world datasets also show how the technique is resilient to realistic levels of noise, which is sometimes a problem with convex relaxation techniques. Indeed, performing the convex relaxation comes at the cost of considerably changing the problem that is actually given to the solver. Therefore, if the noise level is too high, the convex relaxation is not necessarily tight which means that the result of the relaxed optimization is not necessarily an acceptable solution to the original problem. Another interesting conclusion from the results is that the version without the terms to model perceptual aliasing (DC-GMd) is performing as well as the one with those terms (DC-GM). This indicates that the robust formulation presented in this article can also be used to reject outliers in SLAM even in cases where the clustering of loop closures is not an easy task. In regard to the computation time issues (the current implementation could not be used on real-time systems), as it is hinted at in Chapter 4, the DC-GM technique formulation is quite similar to the fast convex solver SE-Sync [7], which leads us to believe that it is possible to build a similar fast solver for DC-GM.

While the technique presented in Chapter 4 is an interesting proof of concept, Chapter 5 introduces DOOR-SLAM which is a practical implementation of robust estimation in SLAM. The outlier rejection technique used in DOOR-SLAM (PCM) is quite different than the DC-GM approach. PCM is not part of the optimization problem. In fact, PCM is a preprocessing step to remove the outliers among the loop closure measurements before the pose graph optimization step. Removing measurements in preprocessing instead of during the optimization has the drawback of relying much more on the accuracy of the odometry measurements. Indeed, if the odometry has a high level of noise even the valid loop closure measurements will have a low consistency with each other. However, when the outlier rejection is part of the optimization, the valid loop closure measurements can partially correct the odometry drift (i.e. noise accumulation) and make it easier to detect the spurious measurements. This explains why the results show that PCM is less accurate in its classification of inliers and outliers than DC-GM. Nevertheless, this is one of the only ways to proceed to reject outlying measurements in multi-robot scenarios and the approach is significantly faster, so it can be applied to real systems like DOOR-SLAM. Moreover, the PCM implementation presented in Chapter 5 is working in a distributed fashion and is shown to lead to large improvements in the pose graph accuracy. Another benefit of DOOR-SLAM illustrated in Chapter 5 is its capacity to produce more valid loop closure measurements than non-robust solutions since it can leverage the outlier rejection mechanism to use less conservative parameters in the SLAM front-end. Since the code for DOOR-SLAM has been released publicly, it could be reused by other research groups in the future to build even more robust and reliable SLAM solutions for multi-robot systems.

6.2 Multi-Robot SLAM

Most robust PGO techniques are designed for single robot systems, yet multi-robot systems are even more vulnerable to perceptual aliasing since each robot is using a different sensor. While results in Chapter 4 are only considering single robot systems, the DC-GM formulation is also compliant with multi-robot systems. Indeed, the main reason why most classical PGO solver cannot be used for multi-robot systems is that they rely on the odometry measurements to provide a good initial guess. In the multi-robot case, such an initial guess cannot be obtained because the multiple trajectories are not initially in the same reference frame (i.e. the alignment of the trajectories with respect to each other is initially unknown). Since the classical methods are local solvers, the initial guess needs to be sufficiently close to the global minimum to avoid falling into a local minimum during the optimization. However, DC-GM offers a convex relaxation adapted for global solvers. Therefore, the resulting solution is the global minimum even if no initial guess is given. Nonetheless, DC-GM is a centralized technique and could not be solved in a distributed fashion on multiple robots. The robots would need to send all their relative pose measurements to a central node that would perform the joint optimization of all trajectories.

In light of those challenges, a multi-robot pose graph optimization technique ideally needs to be distributed and to use a convex formulation to avoid initialization problems. The back-end of DOOR-SLAM proposes a solution to both challenges. The optimization is distributed with minimal information sharing and it is based on the chordal relaxation [77] which is a convex problem.

Also, multi-robot SLAM has limitations in terms of available bandwidth, communication delays, and scalability. Those challenges concern mostly the front-end, which produces most of the data to be shared. DOOR-SLAM leverages the literature to minimize the size of the transmitted messages. One of the main findings shown in Chapter 5 is the effects of parameter tuning in the front-end to increase the number of valid measurements. Indeed, although some of them are outliers, the system obtains more putative loop closures by using less conservative parameters. Then, the system can filter out the outliers and retrieve more valid measurements than it would be possible with more conservative parameters. It is also interesting that DOOR-SLAM, unlike DC-GM and many techniques in the field, has been tested on actual robots during field experiments. Although this requires considerable investments in terms of time and other resources, this provides better evidence that the system is fit for real applications. It also a good argument to convince other research groups to build on DOOR-SLAM to develop new and better multi-robot SLAM techniques. The code is released publicly to facilitate the reproduction and improvement of the system.

Finally, DOOR-SLAM solves the multi-robot SLAM problem in a more flexible way than competing techniques such as the one presented by Cieslewski *et al.* [37]. Indeed, DOOR-SLAM strategy does not require full connectivity maintenance within the robotic team. In other words, robots in the teams do not need to be within the communication range of each other at every point in time. They can rather perform the inter-robot loop closures detection and distributed pose graph optimization when they meet each other. This enables a wide range of applications from collaborative automated warehouse management to the efficient exploration of underground environments.

CHAPTER 7 CONCLUSION

The research work presented in this thesis aimed at the improvement of the robustness of current Simultaneous Localization and Mapping techniques. This thesis introduced two new robust estimation techniques. The first one is designed for single robot systems and offers interesting theoretical guarantees while the other tackles the specific challenges of performing robust estimation in multi-robot systems. The results obtained have the potential to produce significant impacts on the research community given that an accurate localization is crucial to build a truly autonomous robotic system and that current state-of-the-art techniques are vulnerable to data association errors and the presence of outlying measurements.

7.1 Summary of Works

Overall, the two novel techniques presented in this thesis accomplished the research objectives initially stated. First, the DC-GM technique presented in Chapter 4 is partially based on a mathematical model of the perceptual aliasing phenomenon which helps to remove efficiently the outliers among the loop closure measurements. Indeed, the technique leverages the links between the binary variables in the discrete Markov random field to explicitly model the presence of groups of mutually consistent loop closures. The approach presented in Chapter 5 accomplished the other objectives. In fact, DOOR-SLAM is a practical, efficient and robust solution to the multi-robot SLAM problem. The results from simulation to field experiments show that the technique is resilient to difficult conditions, is well adapted for real applications of multi-robot systems and works with less conservative parameters than other techniques.

7.2 Limitations

Nevertheless, even if the research objectives were accomplished, this research work presents several limitations. First, the DC-GM technique relies on a convex relaxation and this kind of approach is known to work only with reasonable levels of noise, which means that the global solver will fail to find the solution to the PGO problem if the noise level is too high. Although the experiments on real-world datasets are conclusive, the noise threshold over which the method will fail can only be determined empirically. Also, the clustering technique used to define which loop closures are grouped together is quite simplistic and its failure could cause the failure of the whole system. Moreover, since the novel PGO formulation introduced in Chapter 4 is solved using a general-purpose SDP solver in Matlab, the current implementation is slow and is not adapted to robotic platforms with limited computation or real-time operations. The DOOR-SLAM technique presents also some significant limitations. In particular, the current implementation of the technique is not resilient to networking errors, large communication delays or packet drops. Unfortunately, those communication issues are very common in the deployment of ad hoc networks.

7.3 Future Research

The natural next step of this research work is to ameliorate the proposed techniques by mitigating their respective limitations. Therefore, it would be necessary to design and implement a specialized fast solver for DC-GM along the lines of [7]. It would also be interesting to develop a more effective way to group the loop closures together in order to avoid failures caused by the incorrect modeling of perceptual aliasing. Concerning DOOR-SLAM, the next steps involve major improvement in networking management to build a system resilient to communication failures and thus better adapted for actual multi-robot systems. Indeed, while ad-hoc networks are solutions of choice for multi-robot systems they are not as reliable as centralized networks.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: speeded up robust features," in *European Conf. on Computer Vision (ECCV)*, 2006.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Intl. Conf. on Computer Vision (ICCV)*. IEEE, 2011, pp. 2564–2571.

[4] L. Carlone, D. Rosen, G. Calafiore, J. Leonard, and F. Dellaert, "Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015, pp. 125–132, (pdf) (code) (datasets: (web)) (supplemental material: (pdf)).

[5] L. Carlone, G. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Trans. Robotics*, vol. 32, no. 3, pp. 545–565, 2016, (pdf) (code).

[6] R. Hartley, J. Trumpf, Y. Dai, and H. Li, "Rotation averaging," *IJCV*, vol. 103, no. 3, pp. 267–305, 2013.

[7] D. Rosen, L. Carlone, A. Bandeira, and J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the Special Euclidean group," in *Intl. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, San Francisco, CA, December 2016, extended arxiv preprint: 1611.00128, (pdf) (pdf) (code).

[8] L. Andersson and J. Nygards, "C-SAM : Multi-robot SLAM using square root information smoothing," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2008.

[9] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Anchorage, Alaska, May 2010, pp. 3185–3192.

[10] T. Bailey, M. Bryson, H. Mu, J. Vial, L. McCalman, and H. Durrant-Whyte, "Decentralised cooperative localisation for heterogeneous teams of mobile robots," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 2859–2865.

[11] M. Lazaro, L. Paz, P. Pinies, J. Castellanos, and G. Grisetti, “Multi-robot SLAM using condensed measurements,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 1069–1076.

[12] 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 *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Seattle, WA, May 2015, pp. 5807–5814.

[13] R. Aragues, L. Carlone, G. Calafiore, and C. Sagues, “Multi-agent localization from noisy relative pose measurements,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 364–369.

[14] A. Cunningham, M. Paluri, and F. Dellaert, “DDF-SAM: Fully distributed slam using constrained factor graphs,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[15] A. Cunningham, V. Indelman, and F. Dellaert, “DDF-SAM 2.0: Consistent distributed smoothing and mapping,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[16] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. Christensen, and F. Dellaert, “Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models, accepted,” *Intl. J. of Robotics Research*, 2017, arxiv preprint: 1702.03435.

[17] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography,” *Commun. ACM*, vol. 24, pp. 381–395, 1981.

[18] J. Neira and J. Tardós, “Data association in stochastic mapping using the joint compatibility test,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 6, pp. 890–897, December 2001.

[19] M. Bosse, G. Agamennoni, and I. Gilitschenski, “Robust estimation and applications in robotics,” *Foundations and Trends in Robotics*, vol. 4, no. 4, pp. 225–269, 2016.

[20] E. Olson and P. Agarwal, “Inference on networks of mixtures for robust robot mapping,” in *Robotics: Science and Systems (RSS)*, July 2012.

[21] N. Sünderhauf and P. Protzel, “Switchable constraints for robust pose graph SLAM,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[22] N. Sunderhauf and P. Protzel, “Towards a robust back-end for pose graph SLAM,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012, pp. 1254–1261.

[23] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, “Robust map optimization using dynamic covariance scaling,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013.

[24] Y. Latif, C. D. C. Lerma, and J. Neira, “Robust loop closing over time.” in *Robotics: Science and Systems (RSS)*, 2012.

[25] L. Yasir, G. Huang, J. Leonard, and J. Neira, “An Online Sparsity-cognizant Algorithm for Visual Navigation,” in *Robotics: Science and Systems (RSS)*, 2014.

[26] L. Carlone, A. Censi, and F. Dellaert, “Selecting good measurements via ℓ_1 relaxation: a convex approach for robust estimation over graphs,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014, (pdf).

[27] M. Graham, J. How, and D. Gustafson, “Robust incremental SLAM with consistency-checking,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 117–124.

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

[29] 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,” *CoRR*, vol. abs/1907.05538, 2019.

[30] A. Oliva and A. Torralba, “Modeling the shape of the scene: a holistic representation of the spatial envelope,” *Intl. J. of Computer Vision*, vol. 42, pp. 145–175, 2001.

[31] I. Ulrich and I. Nourbakhsh, “Appearance-based place recognition for topological localization,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 2, April 2000, pp. 1023 – 1029.

[32] D. Lowe, “Object recognition from local scale-invariant features,” in *Intl. Conf. on Computer Vision (ICCV)*, 1999, pp. 1150–1157.

[33] J. Sivic and A. Zisserman, “Video google: a text re- trieval approach to object matching in videos,” in *Intl. Conf. on Computer Vision (ICCV)*, 2003.

[34] N. Suenderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, “On the performance of ConvNet features for place recognition,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 4297–4304.

[35] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5297–5307.

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

[37] T. Cieslewski, S. Choudhary, and D. Scaramuzza, “Data-efficient decentralized visual SLAM,” *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2018.

[38] T. Cieslewski and D. Scaramuzza, “Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 640–647, Apr. 2017.

[39] 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.

[40] Y. Tian, K. Khosoussi, M. Giamou, J. P. How, and J. Kelly, “Near-Optimal Budgeted Data Exchange for Distributed Loop Closure Detection,” *arXiv:1806.00188 [cs]*, Jun. 2018, arXiv: 1806.00188.

[41] Y. Tian, K. Khosoussi, and J. P. How, “A Resource-Aware Approach to Collaborative Loop Closure Detection with Provable Performance Guarantees,” *arXiv:1907.04904 [cs]*, Jul. 2019, arXiv: 1907.04904.

[42] M. Giamou, K. Khosoussi, and J. P. How, “Talk Resource-Efficiently to Me: Optimal Communication Planning for Distributed Loop Closure Detection,” *arXiv:1709.06675 [cs]*, Sep. 2017, arXiv: 1709.06675.

[43] 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 Trans. Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016, arxiv preprint: 1606.05830, (pdf).

[44] A. Blake, P. Kohli, and C. Rother, *Markov Random Fields for Vision and Image Processing*. The MIT Press, 2011.

[45] L. Carlone and F. Dellaert, "Duality-based verification techniques for 2D SLAM," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 4589–4596, (pdf) (code).

[46] L. Carlone and G. Calafiore, "Convex relaxations for pose graph optimization with outliers," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 2, pp. 1160–1167, 2018, arxiv preprint: 1801.02112, (pdf).

[47] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068–1080, 2008.

[48] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother, "A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems," *Intl. J. of Computer Vision*, vol. 115, no. 2, pp. 155–184, 2015.

[49] A. Fix and S. Agarwal, "Duality and the continuous graphical model," in *European Conf. on Computer Vision (ECCV)*, 2014, pp. 266–281.

[50] C. Zach and P. Kohli, "A convex discrete-continuous approach for Markov random fields," in *European Conf. on Computer Vision (ECCV)*, 2012, pp. 386–399.

[51] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher, "SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion," *IEEE Trans. Pattern Anal. Machine Intell.*, 2012.

[52] F. Dellaert, "Square Root SAM: Simultaneous location and mapping via square root information smoothing," in *Robotics: Science and Systems (RSS)*, 2005.

[53] P. Huber, *Robust Statistics*. John Wiley & Sons, New York, NY, 1981.

[54] J. Casafranca, L. Paz, and P. Piniés, "A back-end ℓ_1 norm based solution for factor graph SLAM," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013, pp. 17–23.

[55] G. Lee, F. Fraundorfer, and M. Pollefeys, “Robust pose-graph loop-closures with expectation-maximization,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[56] M. Pfingsthorn and A. Birk, “Simultaneous localization and mapping with multimodal probability distributions,” *Intl. J. of Robotics Research*, vol. 32, no. 2, pp. 143–171, 2013.

[57] ——, “Generalized graph SLAM: Solving local and global ambiguities through multimodal and hyperedge constraints,” *Intl. J. of Robotics Research*, vol. 35, no. 6, pp. 601–630, 2016.

[58] S. Bowman, N. Atanasov, K. Daniilidis, and G. Pappas, “Probabilistic data association for semantic slam,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 1722–1729.

[59] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun, “Structure from motion without correspondence,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2000.

[60] L. Wang and A. Singer, “Exact and stable recovery of rotations for robust synchronization,” *Information and Inference: A Journal of the IMA*, vol. 30, 2013.

[61] F. Arrigoni, B. Rossi, P. Fragneto, and A. Fusiello, “Robust synchronization in SO(3) and SE(3) via low-rank and sparse matrix decomposition,” *Comput. Vis. Image Underst.*, 2018.

[62] J. Briales and J. Gonzalez-Jimenez, “Cartan-sync: Fast and global SE(d)-synchronization,” *IEEE Robot. Autom. Lett*, vol. 2, no. 4, pp. 2127–2134, 2017.

[63] R. Tron, D. Rosen, and L. Carlone, “On the inclusion of determinant constraints in lagrangian duality for 3D SLAM,” in *Robotics: Science and Systems (RSS), Workshop “The problem of mobile sensors: Setting future goals and indicators of progress for SLAM”*, 2015, (pdf).

[64] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming.” [Online]. Available: http://cvxr.com/cvx

[65] P. Lajoie, S. Hu, G. Beltrame, and L. Carlone, “Modeling perceptual aliasing in SLAM via discrete-continuous graphical models,” *IEEE Robotics and Automation Letters (RA-L)*, 2019, extended ArXiv version: (pdf), Supplemental Material: (pdf).

[66] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Providence, USA, June 2012, pp. 3354–3361.

[67] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8.

[68] D. Scaramuzza and F. Fraundorfer, “Visual odometry: Part I the first 30 years and fundamentals,” 2011.

[69] C. Pinciroli and G. Beltrame, “Buzz: An extensible programming language for heterogeneous swarm robotics,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 3794–3800.

[70] M. Labbe 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.

[71] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[72] R. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty,” *Intl. J. of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1987.

[73] DARPA, “DARPA Subterranean Challenge,” https://www.subtchallenge.com/, 2019, accessed: 2019-09-09.

[74] J. Shi and C. Tomasi, “Good features to track,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 593–600.

[75] F. Dellaert, “Factor graphs and GTSAM: A hands-on introduction,” Georgia Institute of Technology, Tech. Rep. GT-RIM-CP&R-2012-002, September 2012.

[76] C. Pinciroli, V. Trianni, R. O’Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, “ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems,” *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[77] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, “Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 4597–4604, (pdf) (code) (supplemental material: (pdf)).

APPENDIX A MODELING PERCEPTUAL ALIASING IN SLAM VIA DISCRETE-CONTINUOUS GRAPHICAL MODELS - SUPPLEMENTAL MATERIAL

Pierre-Yves Lajoie, Siyi Hu, Giovanni Beltrame, Luca Carlone

This supplemental material presents a set of additional experimental results to provide more insights on the performance and limitations of the proposed approach. The experiments are organized in four sections. The experiments in Section A evaluate the impact of the maximum admissible residuals parameter $\bar{c}$. Section A analyzes the effect of the correlation terms $\bar{c}^{(ij)}_{(i'j')}$ on the truncated least squares objective function and gives some intuition on how to choose these parameters. Following this analysis, the experiments in Section A show the impact of an incorrect modeling of outlier correlation in DC-GM. Finally, Section A shows extra simulation results in a more realistic Manhattan World graph that complement the results on the Grid graph shown in the main paper.

Effect of the Maximum Admissible Residual Threshold $\bar{c}$

This set of experiments evaluates the impact of the choice of the maximum admissible residuals threshold $\bar{c}$. In order to illustrate the role of $\bar{c}$, we used the same simulation setup of the paper, but we varied the number of standard deviation of the measurement noise (σ) that we wished to accept. Intuitively, we expect that a lower $\bar{c}$ will lead to more inliers being rejected and a higher $\bar{c}$ will lead to more outliers being accepted.

We evaluate the results for 3 choices of $\bar{c}$, in particular we consider $\bar{c} = \{0.01\sigma, 1\sigma, 2\sigma\}$, and for each set of tests we report (i) the average translation error, (ii) the percentage of rejected inliers, (iii) the percentage of rejected outliers, and (iv) the rank of the matrix $\boldsymbol{Z}^{\star}$ computed by the proposed SDP relaxation (in planar problems, the relaxation is tight when $\operatorname{rank}(\boldsymbol{Z}^{\star}) = 2$). The rank is computed using a numerical threshold of $10^{-3} \cdot \lambda_{max}$, where λ_{max} is the maximum eigenvalue of $\boldsymbol{Z}^{\star}$. All data points are averaged over 5 runs, and statistics are computed for increasing percentage of outliers.

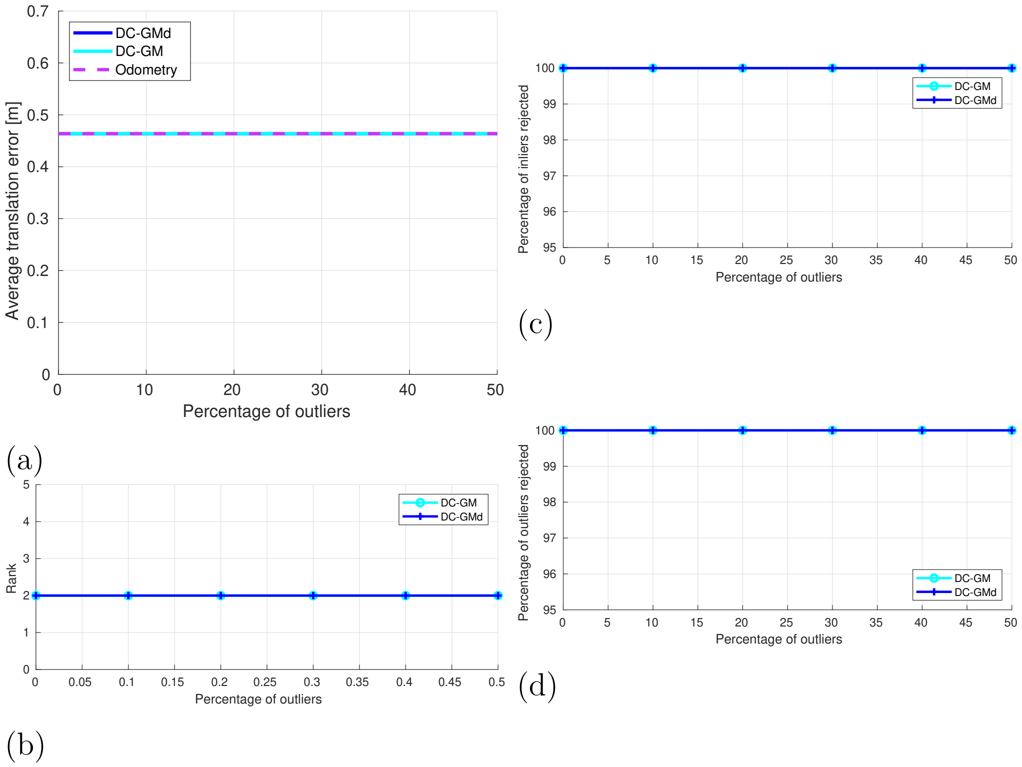
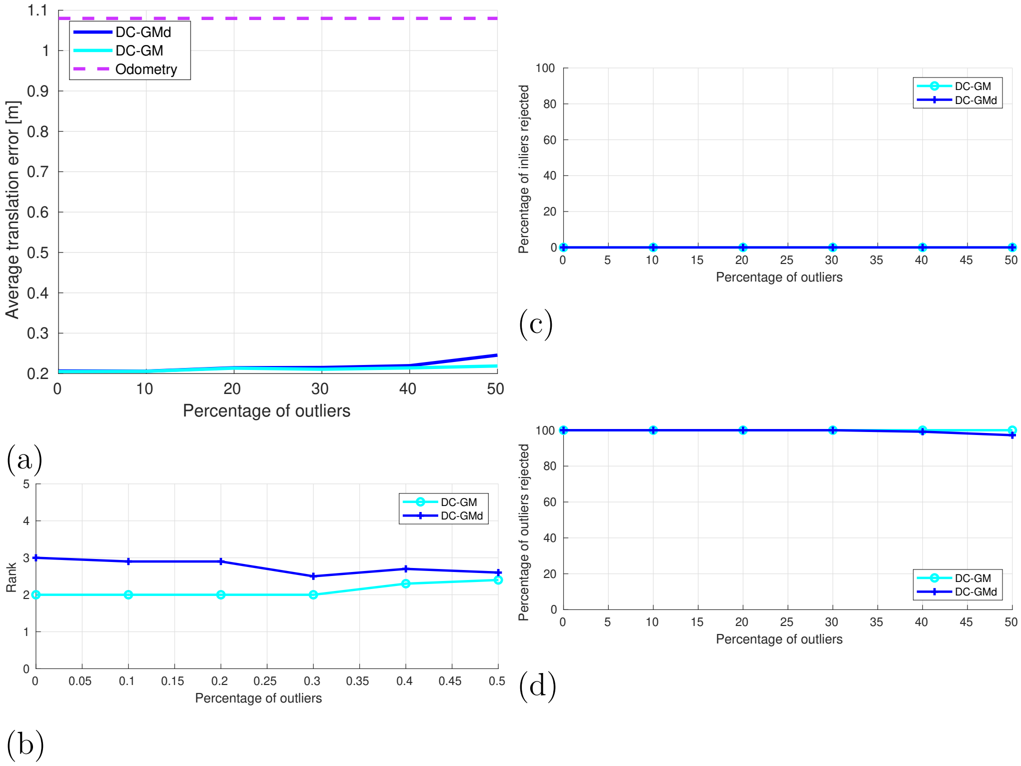


Figure A.1: Results on the simulated grid graph with maximum admissible residuals of 0.01σ. (a) average translation error of the DC-GM and DC-GMd solutions compared with the odometric estimate; (b) rank of $\boldsymbol{Z}^{\star}$, (c) percentage of rejected inliers, and (d) percentage of rejected outliers for DC-GM and DC-GMd.

As expected, in Figure A.1 we observe that with a very low tolerance on the residuals, our technique rejects all loop closures and therefore falls back to the odometric estimate. It is worth noting that in this case the rank of the returned solution $\boldsymbol{Z}^{\star}$ is exactly 2 which indicates that the relaxation is tight when all loop closures are rejected.



(a)

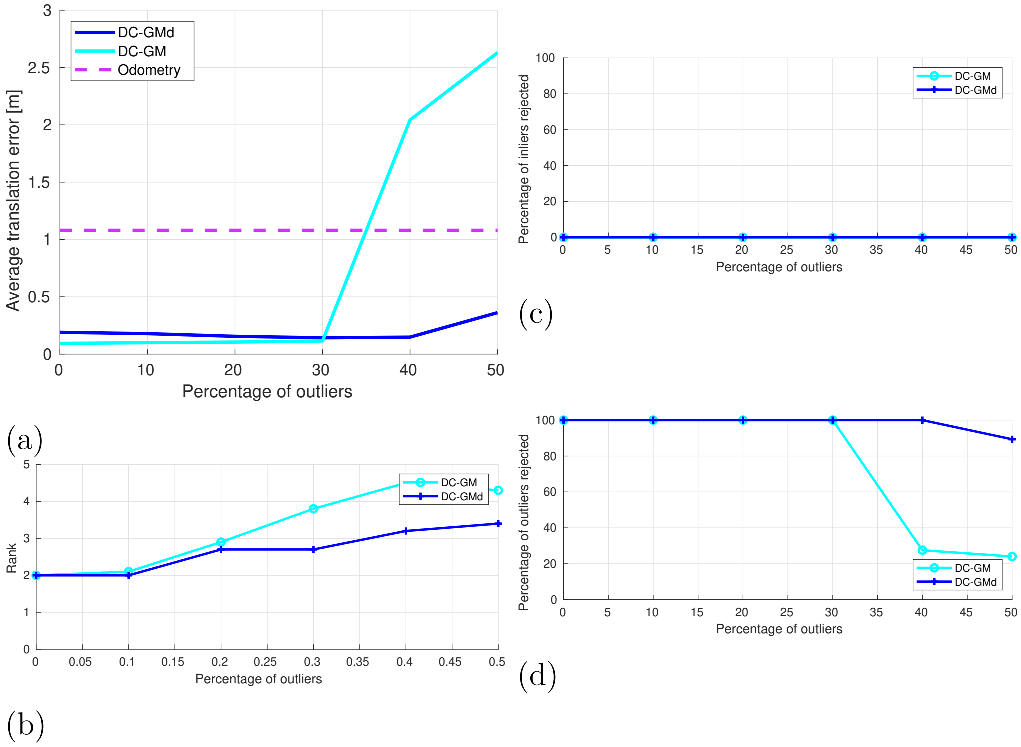
(c)

(b)

(d)

Figure A.2: Results on the simulated grid graph with maximum admissible residuals of 1σ.

Figure A.2 presents the results obtained with a threshold of 1 standard deviation on the residuals. Those results have already been discussed in the paper. We observe that for increasing percentage of outliers, DC-GM is able to reject all outliers, while DC-GMd tends to incorrectly accept a very small portion of outliers. The plot of the rank of the SDP solutions provides some extra insight on the performance of the relaxation and shows that the relaxation of the coupled approach (DC-GM) is tighter than the decoupled one (DC-GMd).



(a)

(b)

(c)

(d)

Figure A.3: Results on the simulated grid graph with maximum admissible residuals of 2σ.

Figure A.3 shows that with a looser threshold on the maximal residual threshold of 2σ, the proposed techniques tend to fail in presence of a large amount of outliers (40% and above). Surprisingly, the coupled formulation DC-GM has a lower breakdown point, and is dominated by DC-GMd for large percentages of outliers. Figure A.3(c) shows that both techniques accept all the inliers, but Figure A.3(d) shows that the loss in accuracy of DC-GM stems from accepting several outliers. This can be partially explained by the the rank in Figure A.3(b), which tends to be larger in this case for DC-GM, leading to a looser relaxation. A second explanation is provided in Section A which shows that the correlation terms have the effect of "inflating" the maximum admissible residual threshold, making DC-GM more prone to accept outliers when $\bar{c}$ is large.

Effect of the Correlation Terms $\bar{c}^{(ij)}_{(i'j')}$ on the Objective Function

In order to understand the impact of the correlation terms $\bar{c}^{(ij)}_{(i'j')}$ on the objective function, let us consider a single loop closure (i,j), and call $\mathcal{C}_{ij}$ the set of edges correlated to (i,j). From the coupled formulation, we can isolate all the terms involving the loop closure (i,j)

which we report below after omitting constant terms:

$$\tfrac{(1+\theta_{ij})}{2}\|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} \quad -\frac{\theta_{ij}}{2}\bar{c} \ \ - \sum_{(i',j')\in\mathcal{C}_{ij}} \bar{c}^{(ij)}_{(i'j')}\theta_{ij}\theta_{i'j'} \tag{A.1}$$

Now assume that all neighbors "decide to accept" the corresponding measurements, i.e., $\theta_{i'j'} = +1$ for all $(i', j') \in \mathcal{C}_{ij}$. Then, eq. (A.1) becomes:

$$\tfrac{(1+\theta_{ij})}{2}\|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} \quad -\frac{\theta_{ij}}{2}\left(\bar{c} + 2\sum_{(i',j')\in\mathcal{C}_{ij}} \bar{c}^{(ij)}_{(i'j')}\right) \tag{A.2}$$

Similarly, when all neighbors "decide to reject" the corresponding measurements, i.e., $\theta_{i'j'} = -1$ for all $(i', j') \in \mathcal{C}_{ij}$. Then, eq. (A.1) becomes:

$$\tfrac{(1+\theta_{ij})}{2}\|\boldsymbol{T}_j - \boldsymbol{T}_i\bar{\boldsymbol{T}}_{ij}\|^2_{\boldsymbol{\Omega}} \quad -\frac{\theta_{ij}}{2}\left(\bar{c} - 2\sum_{(i',j')\in\mathcal{C}_{ij}} \bar{c}^{(ij)}_{(i'j')}\right) \tag{A.3}$$

It is clear that in general, the presence of the correlation term alters the value of the threshold $\bar{c}$. In other words, the fact that neighboring edges accept a measurement, makes the other edges "more permissive" by increasing the corresponding threshold $\bar{c}$. The threshold however always remains in the interval:

$$[\bar{c} - 2\sum_{(i',j')\in\mathcal{C}_{ij}} \bar{c}^{(ij)}_{(i'j')} \quad , \quad \bar{c} + 2\sum_{(i',j')\in\mathcal{C}_{ij}} \bar{c}^{(ij)}_{(i'j')}] \tag{A.4}$$

A pictorial representation is given in Figure A.4. This understanding also informs us on how to set the coefficients $\bar{c}^{(ij)}_{(i'j')}$. According to (A.4), one should make sure that $\bar{c}^{(ij)}_{(i'j')}$ is relatively small compared to $\bar{c}$, such that the correlation does not dominate the outlier rejection decisions. Similarly, the size of the interval in (A.4) depends on the number of neighbors of edge (i, j); this suggests normalizing the coefficients by the number of neighbors such that the term $\sum_{(i',j')\in\mathcal{C}_{ij}} \bar{c}^{(ij)}_{(i'j')}$ does not dominate the outlier rejection threshold $\bar{c}$.

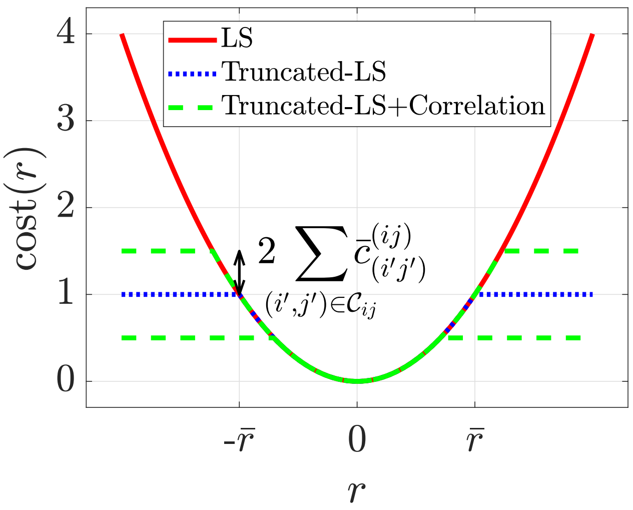


Figure A.4: Effect of the correlation terms on the robust cost function.

Finally, it is interesting to note that the presence of the correlation terms can degrade the performance of DC-GM if the correlation terms are chosen incorrectly. We analyze this aspect in the following section where we consider experiments with heterogeneous groups of measurements (i.e., a mix of inliers and outliers).

Effect of heterogeneous groups of loop closures

The experiments in this section evaluate the impact of an incorrect modeling of the outlier correlation. In particular, we consider a setup where heterogeneous loop closure groups (composed of both inliers and outliers) are added to the graph and we add correlation terms $\bar{c}^{(ij)}_{(i'j')}$ between each pair of edges in the groups. This incorrect modeling is expected to challenge the performance of DC-GM, since the model will attempt to encourage consistent inlier/outlier decisions within each group, despite the fact that each edge in the group is assigned to be an inlier/outlier at random. In particular, we expect DC-GM to perform poorly when the correlation term $\bar{c}^{(ij)}_{(i'j')}$ is large, while it is expected to fall back to the performance of the decoupled approach DC-GMd when $\bar{c}^{(ij)}_{(i'j')}$ is small. We present results for decreasing value of the correlation term $\bar{c}^{(ij)}_{(i'j')}$ equal to 10%, 1%, and 0.1% of the maximum admissible residuals parameter $\bar{c}$, respectively. The results, showing the percentage of rejected inliers and outliers for increasing percentage of outliers, confirm the expected behavior.

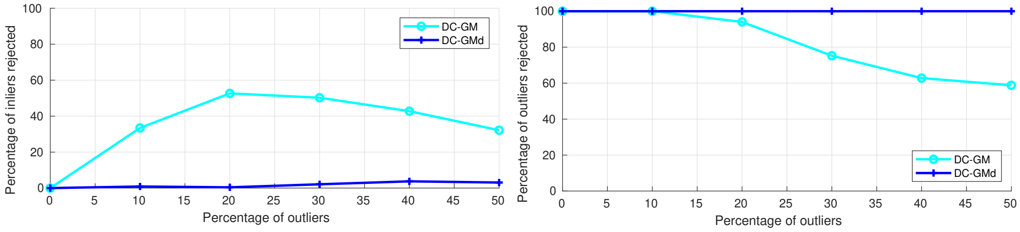


Figure A.5: Results on the simulated grid graph with heterogeneous groups of loop closures and correlation terms $\bar{c}_{(i'j')}^{(ij)}$ equal to $0.1\bar{c}$. (left) Percentage of rejected inliers; (right) Percentage of rejected outliers.

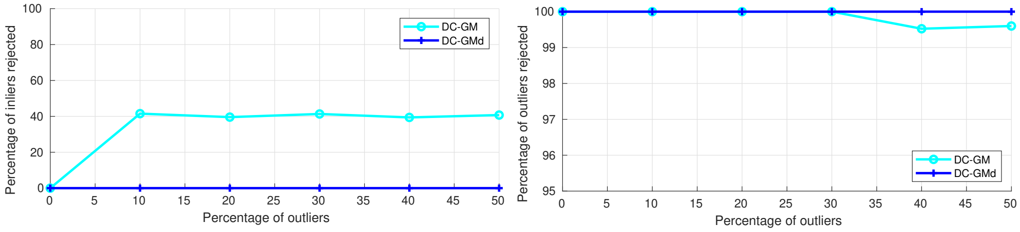


Figure A.6: Results on the simulated grid graph with heterogeneous groups of loop closures and correlation terms $\bar{c}_{(i'j')}^{(ij)}$ equal to $0.01\bar{c}$.

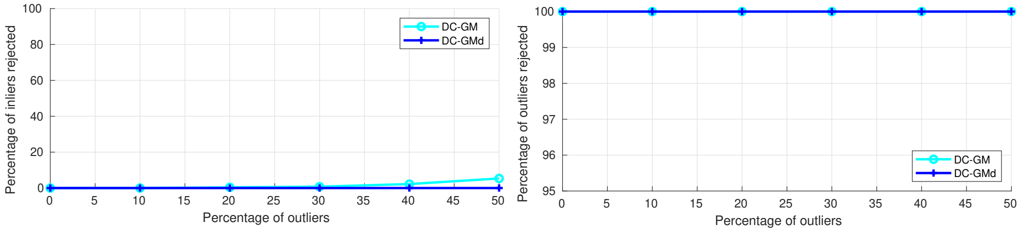


Figure A.7: Results on the simulated grid graph with heterogeneous groups of loop closures and correlation terms $\bar{c}_{(i'j')}^{(ij)}$ equal to $0.001\bar{c}$.

Figures A.5, A.6, and A.7 show that the performance of DC-GM is worse when the correlation terms are large (and incorrect) while it approaches DC-GMd when the correlation terms are small. This is also consistent with the interpretation of the cost function in Figure A.4 where higher values of the correlation terms lead to a larger range of values for the maximum admissible residuals. A larger range is more likely to lead to the acceptance of outliers and/or the rejection of inliers.

Additional simulation results

We performed additional simulation experiments to provide further insights to the reader on the performance of the proposed techniques. These experiments involve a more realistic Manhattan World graph. Below we report the average translation error, the percentage of rejected inliers and outliers, and a visualization of the estimated trajectory. Statistics are computed over 10 runs with increasing percentage of outliers. The proposed approaches (DC-GM and DC-GMd) are compared against other techniques (Vertigo, DCS, RRR) potentially reporting multiple choices of parameters for the competing techniques.

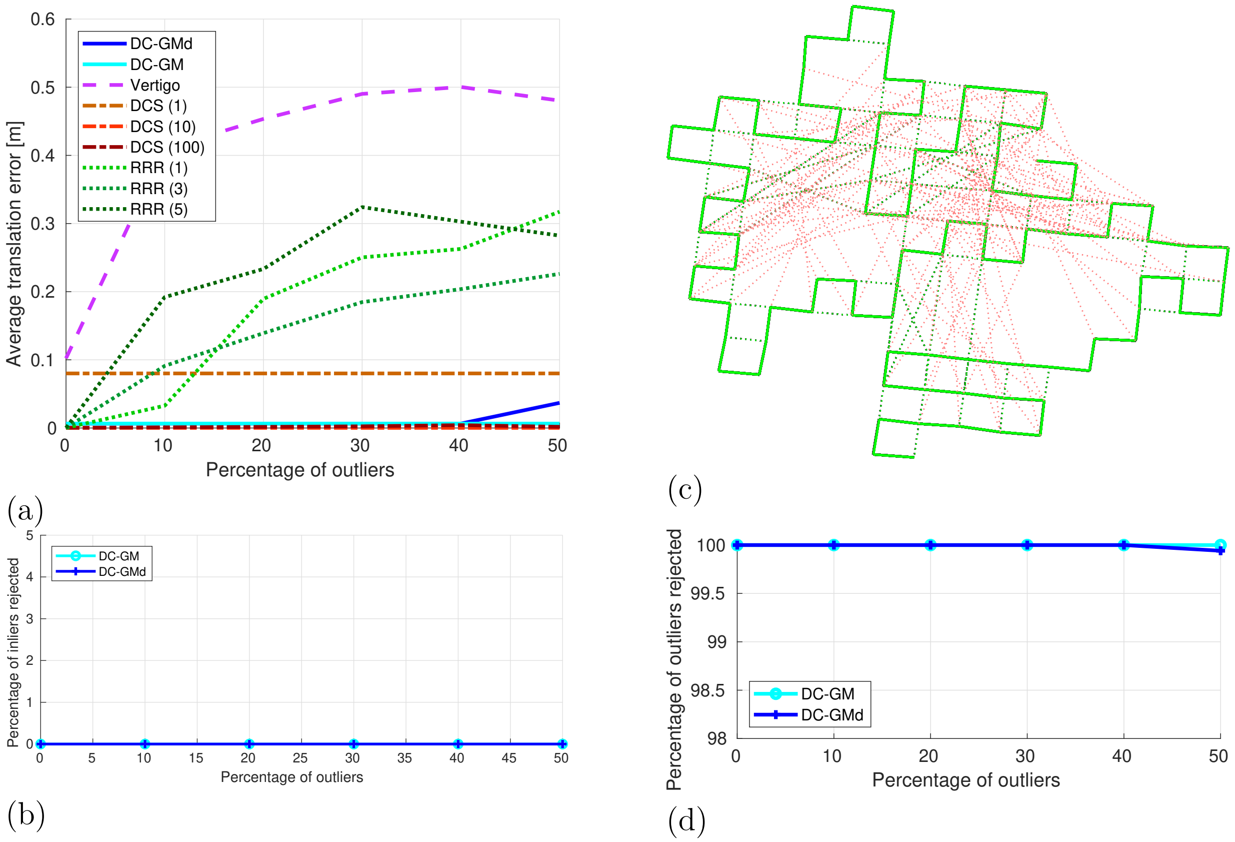


Figure A.8: Results on a Manhattan World graph. (a) Average translation error for the different techniques; (c) Ground truth (green) overlaid on the DC-GM solution (black, indistinguishable from the ground truth), and outlier loop closures (red). (b) Percentage of rejected inliers; (d) Percentage of rejected outliers.

Consistently with the results on the grid graph, Figure A.8 shows that the performance of DC-GM and DC-GMd is comparable, but DC-GM ensures slightly more accurate results when the percentage of outliers is large. On the other hand, both Vertigo and RRR performed remarkably worse than DC-GM and DC-GMd on the Manhattan World graph (for RRR the performance was poor for any choice of parameters). DCS performed well when the tuning parameter Φ was chosen to be 10 or 100, but performed worse than DC-GM when the default parameter $\Phi = 1$ was used.