UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

MULTI-OBJECT METRICS FOR ROBOTIC MAP EVALUATION AND DATA REGISTRATION

TESIS PARA OPTAR AL GRADO DE
DOCTOR EN INGENIERÍA ELÉCTRICA

PABLO ADOLFO BARRIOS SOTO

PROFESOR GUÍA:
MARTIN ADAMS

MIEMBROS DE LA COMISIÓN:
CLAUDIO PÉREZ FLORES
BA-NGU VO
REZA HOSEINNEZHAD

Este trabajo ha sido parcialmente financiado por Becas CONICYT - Doctorado Nacional, 2012, ANID/PIA AFB180004 y por CONICYT FONDECYT project 1231658.

SANTIAGO DE CHILE
2023

RESUMEN DE LA TESIS PARA OPTAR
AL GRADO DE DOCTOR EN INGENIERÍA ELÉCTRICA
POR: PABLO ADOLFO BARRIOS SOTO
FECHA: 2023
PROF. GUÍA: MARTIN ADAMS

MÉTRICAS MULTI-OBJETO PARA LA EVALUACIÓN DE MAPAS ROBOTICOS Y REGISTRO DE DATOS

En mapeo robótico y en Simultaneous Localization And Mapping (SLAM), la capacidad de evaluar la calidad de los mapas estimados es crucial. Aunque existen conceptos para cuantificar el error en la trayectoria estimada de un robot, o en un subconjunto de caracteristicas del mapa estimado, rara vez se considera conjuntamente la diferencia entre todas las estimaciones y el ground truth. En la literatura de tracking, la métrica Optimal Sub-Pattern Assignment (OSPA) proporciona una solución a este problema. A pesar de sus ventajas sobre otras métricas, la métrica OSPA tiene algunas limitaciones en mapeo robótico. Por lo tanto, la primera componente de esta tesis introduce la métrica Cardinalized Optimal Linear Assignment (COLA), para la evaluación de mapas robóticos. Además, la segunda componente de esta tesis amplía el uso de las metricas multi-objeto al problema del registro de nubes de puntos, el cual es una componente crucial en muchas aplicaciones, incluyendo la estimación del movimiento de los sensores sensor y la reconstrucción 3D. Los resultados de esta tesis demuestran que la métrica COLA es fiable para evaluar mapeo robótico y SLAM. Además, la métrica COLA mejora el registro de nubes de puntos abordando conjuntamente los errores espaciales y de cardinalidad.

RESUMEN DE LA TESIS PARA OPTAR
AL GRADO DE DOCTOR EN INGENIERÍA ELÉCTRICA
POR: PABLO ADOLFO BARRIOS SOTO
FECHA: 2023
PROF. GUÍA: MARTIN ADAMS

MULTI-OBJECT METRICS FOR ROBOTIC MAP EVALUATION AND DATA REGISTRATION

In robotic mapping and Simultaneous Localization And Mapping (SLAM) the ability to assess the quality of estimated maps is crucial. While concepts exist for quantifying the error in the estimated trajectory of a robot, or a subset of the estimated feature locations, the difference between all current estimated and ground-truth features is rarely jointly considered. The Optimal Sub-Pattern Assignment (OSPA) metric provided a solution to the problem of assessing target tracking algorithms. Despite its advantages over other metrics when gauging multi-target tracking errors, the OSPA metric has some limitations when applied to robotic mapping errors. Therefore, the first component of this thesis introduces the Cardinalized Optimal Linear Assignment (COLA) metric, as a complement to the OSPA metric, for feature map evaluation Further, the point cloud registration problem is addressed as an application of multi-object metric concepts. Therefore, the second component of this thesis extends the application of multi-object metrics to the fundamental research problem of point cloud data registration for sensor motion estimation and 3D reconstruction. The results of this thesis demonstrate that the COLA metric is reliable for evaluating robotic mapping and SLAM. Also, the COLA metric improves point cloud registration by jointly addressing spatial and cardinality errors.

A Gabriela, mi esposa. Por todo tu amor, apoyo y paciencia infinita.

A Cecilia, mi madre. Por estar siempre conmigo en los tiempos difíciles.

A mis abuelos, Inés y Sergio. Por toda su sabiduría y enseñanzas.

A Claudia, quien me cuida desde el cielo.

A mis hijos, León y Leticia. Son la luz al final de este camino.

Acknowledgments

First, I want to express my deepest gratitude to my family for their unconditional support throughout this process.

I also sincerely thank Dr. Martin Adams, my advisor, for generously sharing his experience and knowledge. His dedication has left a significant positive imprint on my professional development. Thank you for your patience and constant guidance.

In addition, I thank the Advanced Mining Technology Center (AMTC) for the valuable funding provided, which made the partial completion of this work possible.

My friends in the 3rd floor Computer Vision lab deserve special thanks. To Felipe, Leonardo, Keith, Vicente, Daniel, and Ghayur, I thank you from the bottom of my heart for all the shared moments and your invaluable help and friendship.

Finally, I would like to acknowledge CONICYT, now called ANID, for awarding me the "Becas Conicyt - Doctorado Nacional 2012", whose funding made this doctoral thesis possible, and also to ANID/PIA AFB180004 and CONICYT FONDECYT project 1231658.

TABLE OF CONTENT

List of Tables

List of Figures

Acronyms

4PCS 4-Points Congruent Sets.

BnB Branch and Bound.

COLA Cardinalized Optimal Linear Assignment.

DA Data Association.

DE = FA + MD Detection Errors.

FA False Alarms.

FastDesp Fast Descriptors and Correspondence Propagation.

FCFM Faculty of Physical and Mathematical Sciences.

FPFH Fast Point Feature Histograms.

GM Gaussian mixture.

GOGMA Globally Optimal Gaussian Mixture Alignment.

Go-ICP Global ICP.

GORE Guaranteed Outlier Removal.

GOSPA Generalized Optimal Sub-Pattern Assignment.

GT Ground Truth.

ICP Iterative Closest Point.

KC Known Correspondence.

MC Monte Carlo.

MD Missed Detections.

MH Multi-Hypothesis.

NDT Normal Distributed Transform.

NEES Normalized Estimation Error Squared.

NN Nearest Neighbor.

OMAT Optimal MAss Transfer.

OSPA Optimal Sub-Pattern Assignment.

OSPA(2) Optimal Sub-Pattern Assignment(2).

PCD Point Cloud Data.

PFH Point Feature Histograms.

PSO Particle Swarm Optimization.

PSO-COLA Particle Swarm Optimization-Cardinalized Optimal Linear Assignment.

RANSAC RANdom Sample Consensus.

RFS Random Finite Set.

RMS Root Mean Squared.

SHOT Signature of Histograms of OrienTations.

SLAM Simultaneous Localization And Mapping.

SSEE Sum of Squared Euclidean Errors.

Super4PCS Super 4-Points Congruent Sets.

TEASER Truncated least squares Estimation And SEmidefinite Relaxation.

UC Unknown Correspondence.

Chapter 1

Introduction

Fundamental to any state estimation problem is the concept of estimation error. Solutions to robotic mapping and SLAM, in which usually the location of an unknown number of features should be estimated, are numerous offering various degrees of performance. Examples include classical methods such as recursive EKF SLAM [1, 2], Multi-Hypothesis (MH) FastSLAM [3], batch estimation methods such as GraphSLAM [4] and iSAM [5, 6] and Random Finite Set (RFS) methods [7, 8]. Irrespective of the estimation methods used, while clear concepts exist for quantifying the error in the estimated pose or trajectory of a robotic vehicle [9] and/or a *subset* of the estimated feature locations within a single reference frame [2], the absolute difference between *all* currently estimated and GT features in the map is rarely jointly considered. In SLAM, this is of equal importance to the vehicle trajectory estimate.

Previous methods, which have been used to compare estimated and GT maps, include the Root Mean Squared (RMS) and Normalized Estimation Error Squared (NEES) metrics [10, 11, 12, 13]. These metrics are only defined if the number of estimated and GT features are the same. In realistic scenarios, these numbers will differ, meaning that a *sub-set* of at least one of the map's features must be used. Examining the spatial errors between sub-sets of the estimated and GT features may illustrate the consistency of the spatial state of the selected features, but gives no indication of the quality of the estimate of the *joint* multi-feature map state. The primary difficulty in mathematically defining map estimation error is caused by the differences between the estimated and true *number* of features, and the need to satisfy the four metric axioms [14][1]. Map estimators which miss detections and/or produce false alarms are potentially dangerous, since missed detections can result in robot to object collisions and false alarms in a (e.g., search and rescue) robot thinking it is blocked, when in fact it could maneuver to a victim in need of assistance. Therefore, the first component of this thesis suggests that a concise map estimation metric, which penalizes estimators for both feature detection as well as description (of which the location state is an example) errors, is important, but lacking in the literature. It therefore introduces the concept of multi-object metrics and their application to robotic map estimation error. The second component of this

[1]The four metric axioms can be defined as follows. Let $\mathcal{X}$ be an arbitrary, non-empty set. Then the function d is a metric iff: 1) $\mathrm{d}(\mathbf{x},\mathbf{y}) \geq 0$, for all $\mathbf{x},\mathbf{y} \in \mathcal{X}$; 2) $\mathrm{d}(\mathbf{x},\mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$, for all $\mathbf{x} \in \mathcal{X}$ (identity axiom); 3) $\mathrm{d}(\mathbf{x},\mathbf{y}) = \mathrm{d}(\mathbf{y},\mathbf{x})$, for all $\mathbf{x},\mathbf{y} \in \mathcal{X}$ (symmetry axiom); 4) $\mathrm{d}(\mathbf{x},\mathbf{y}) \leq \mathrm{d}(\mathbf{x},\mathbf{z}) + \mathrm{d}(\mathbf{z},\mathbf{y})$, for all $\mathbf{x},\mathbf{y},\mathbf{z} \in \mathcal{X}$ (triangle inequality axiom).

thesis extends the application of multi-object metrics to the fundamental research problem of point cloud data registration for sensor motion estimation and 3D reconstruction.

1.1 Motivation: The Dilemma of Map Error Quantification

To illustrate the dilemma in map quality estimation, recall that in state-of-the-art feature-based SLAM formulations, the map is constructed by stacking features into a vector, and consider the simplistic scenarios showed in Fig. 1.1. Fig. 1.1a depicts a scenario in which

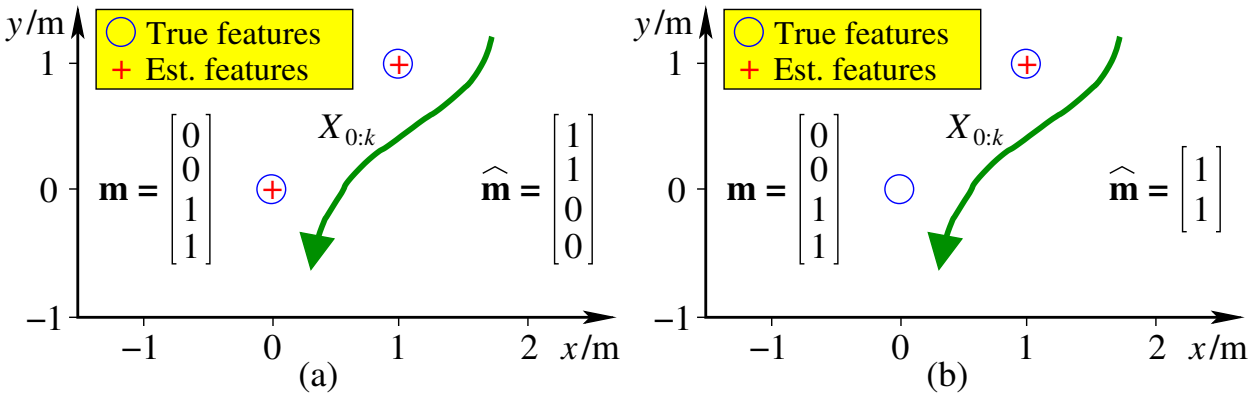


Figure 1.1: *A hypothetical scenario showing a fundamental inconsistency with vector representations of feature maps. If* $\mathbf{m}$ *is the GT map (blue circles), how should the error be assigned when the number of features in the map estimate,* $\widehat{\mathbf{m}}$*, (red crosses) is incorrect?*

there are two true features at coordinates $(0, 0)$ and $(1, 1)$ (blue circles). The true map, $\mathbf{m}$, is then represented by the vector $\mathbf{m} = [0\ 0\ 1\ 1]^\top$. If the estimated features (red crosses) are stacked into a vector, in the order in which they were detected given the green vehicle trajectory $X_{0:k}$ (as shown in the figure), and perfect measurements, the estimated map may be given by the vector $\widehat{\mathbf{m}} = [1\ 1\ 0\ 0]^\top$. Despite a seemingly perfect estimate of the map, the Euclidean error of the estimated map, $||\mathbf{m} - \widehat{\mathbf{m}}||$, is 2. This inconsistency arises because the *ordering* of the features in the representation of the map should not be relevant. A vector representation however, imposes a mathematically strict arrangement of features in the estimated map, usually based on the order in which they were detected [1, 2]. Intuitively, the elements of $\widehat{\mathbf{m}}$ could be permuted, via successful data association, to obtain zero error, however the representation of all possible permutations of the elements of a vector is, by definition, a set. Hence, such a permuting operation implies that the resulting error distance should be a distance for sets.

Another problem is depicted in Fig. 1.1b, in which there are again two features at (0,0) and (1,1), but due to a missed detection, the estimated map comprises only one feature at (1,1). In such a situation, it is difficult to define a mathematically consistent error metric (Euclidean error, RMS, NEES) between the vectors $\mathbf{m}$ and $\widehat{\mathbf{m}}$ since they contain a different number of features.

In the target tracking literature, a similar dilemma exists in which the performance of trackers must be assessed, which led to the development of multi-object metrics, based on sets instead vectors.

1.2 The Application of Set-based metrics to Point Cloud Registration

Point Set Registration is the process of estimating a rigid transformation, defined by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$, to align two Point Cloud Data (PCD) sets defined as a reference set of points $\mathcal{M}$ and a model set of points $\widehat{\mathcal{M}}$, in a consistent manner. In the literature numerous solutions exist with different degrees of performance. One of the first solutions to this problem is the Iterative Closest Point (ICP) [15] algorithm which provides a simple conceptual method based on vector-based L_2 norm. Applications are numerous in robotics and computer vision, such as robot localization [16] [17], object recognition [18] [19], and 3D reconstruction [20] [21].
Similarly to mapping error estimation, point set registration in particular has been developed traditionally by using vector-based metrics, such as the L_2 metric. However, in the most general case, $\mathcal{M}$ and $\widehat{\mathcal{M}}$ do not necessarily contain the same number of points as illustrated in Figure 1.2, due to the presence of miss-detections and false alarms. Also, in general the correspondence between points is not given a priori. Therefore, methods which utilize vector-based metrics require heuristics to cope with false alarms and miss-detection in the presence of spatial uncertainty within the point sets, in order to solve the data association between points from $\mathcal{M}$ and $\widehat{\mathcal{M}}$.

For solving the registration problem shown in Figure 1.2, in which the Chui & Rangarajan fish dataset was used [22], the ICP algorithm assigns the closest points within $\mathcal{M}$ and $\widehat{\mathcal{M}}$. Because of this, the ICP algorithm suffers limitations since it, often erroneously, assumes that the closest pairs define the correct correspondence. This limitation often causes ICP to fail for various reasons. One possibility is that there are many false alarms and miss-detections in the point clouds, which requires complementary methods to remove them, e.g., the RANdom Sample Consensus (RANSAC) algorithm [23], or the Guaranteed Outlier Removal (GORE) algorithm [24]. If the point clouds are not initially approximately aligned, ICP is susceptible to local minima solutions producing inaccurate estimates. Further, ICP uses an L_2 metric to quantify registration error, which cannot penalize cardinality errors.

The lack of any criteria to penalize cardinality errors requires other metrics to evaluate the errors between PCD sets and in this thesis, multi-object set-based metrics are suggested to cope with both spatial and cardinality errors in a joint manner.

1.3 Thesis Hypotheses

In this thesis, the following hypotheses are proposed

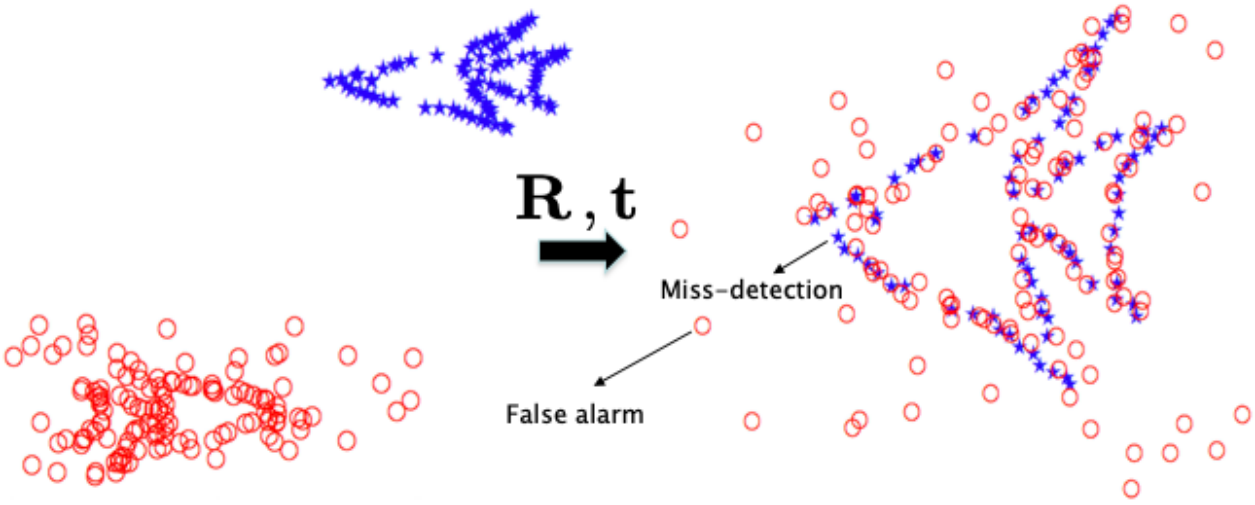


Figure 1.2: *A scenario of point cloud data registration is presented here, in the presence of miss-detections and false alarms, based on the Chui & Rangarajan fish synthetic dataset. Blue stars represent the reference set $\mathcal{M}$, and the model set $\widehat{\mathcal{M}}$ with red circles. Note that in realistic scenarios $|\mathcal{M}| \neq |\widehat{\mathcal{M}}|$. The excessive rate of false alarms presented here highlights one of the difficulties of point cloud registration, called the correspondence problem. More precisely, the correspondence problem is the estimation process that determines the correspondences between all points within $\mathcal{M}$ and $\widehat{\mathcal{M}}$. Therefore, when a large rate of false alarms or miss-detections exists, the probability of correct correspondence decreases. Thus, if the correspondence problem is not accurately solved, point cloud registration may fail in its execution, which suggests the requirements of new methods to cope with both spatial and cardinality errors in a joint manner.*

1. Error assessments in feature-based robotic mapping can be evaluated in a consistent manner based on multi-object metrics.
2. In SLAM, assessing a robot's trajectory error does not necessarily give an accurate assessment of the robot's mapping error.
3. Multi-object metrics have applications as principled measures for point cloud registration and general Data Association (DA), which take into account data differences (e.g., object location) and their cardinalities.

1.4 Thesis Objectives

The main objectives of this thesis are as follows:

1. As its name implies, SLAM jointly estimates a robot's trajectory and map. State-of-the-art SLAM evaluation methods often only consider the trajectory. This thesis will demonstrate that the meaningful quantification of mapping error is also important and necessary for true SLAM performance evaluation. Section 5.2 addresses this objective.
2. As a consequence of objective 1, a new multi-object metric, the COLA metric is derived, which takes into account detection as well as spatial errors. The objective is to

overcome some of the problems associated with state-of-the-art metrics, thus providing a meaningful mapping error. Chapter 3 presents COLA as a new metric that satisfies the four metric axioms. Additionally, Section 5.2 highlights its performance evaluation based on the "Parque O'Higgins" and the "CityTrees10k" data sets.

3. It will be demonstrated that multi-object metrics can also be applied to the problem of point data registration. A new method, referred to as the PSO-COLA algorithm, will be derived to improve point cloud data registration in the presence of the realistic phenomena of detection and spatial uncertainty. Chapter 4 presents PSO-COLA for point data registration based in multi-object metric. Additionally, Section 5.3 highlights its performance compare with state-of-the-art methods under different conditions.

The specific objectives of this thesis are as follows:

1. The COLA mapping metric will be implemented and its performance in terms of providing meaningful mapping error estimates, when compared with state-of-the-art methods, will be validated. Chapter 3 addresses this objective.
2. Since SLAM is a joint (robot trajectory and map) estimation problem, different SLAM algorithms should be compared based on *both* their trajectory and mapping estimation errors. Therefore, overall SLAM performance will be evaluated based on the mapping component performance evaluations of the COLA metric as well as the trajectory component evaluations performed by state-of-the-art single-object metrics. Section 5.2 addresses this objective.
3. Based on both controlled and real 3D datasets, improved point cloud data registration in the presence of *both* detection and spatial errors will be demonstrated with the PSO-COLA algorithm. Section 5.3 addresses this point.

1.5 Thesis Outline

The thesis proceeds as follows. Chapter 2 discusses the state of the art in robotic map evaluation methods and 3D point data registration. Chapter 3 introduces the COLA metric as a viable map evaluation alternative which takes into account both detection and spatial uncertainty. Chapter 4 introduces the PSO-COLA algorithm as an alternative for point data registration. Chapter 5 presents the results. Multi-object metrics are evaluated in real SLAM scenarios using the "Parque O'Higgins" dataset [25] as well as the "CityTrees10k" SLAM dataset [5] and mapping results based on iSAM. Also, the performance of the PSO-COLA algorithm as a point cloud registration solution is evaluated in controlled scenarios based on the "Stanford Bunny" and "Dragon" datasets [26], and in the real "ETH Apartment" scenario [27]. Chapter 6 concludes the thesis. Finally, Appendix A corresponds to the published papers based on this thesis.

Chapter 2

State of the Art

2.1 Introduction

In [28], it is noted that many robotic mapping performance measures are not true metrics, since in order to obtain an intuitive measure, at least one of the metric axioms is violated. To demonstrate the basic problem of precisely quantifying feature-based mapping error, Fig. 5.2 in Chapter 5, shows posterior map estimates from two different feature-based SLAM filters. The true feature maps in each figure are shown as blue stars and the estimated maps as red spatial uncertainty ellipses. A natural question is: "Which estimate is closer to GT" Visual intuition is difficult due to the combination of missed detections, false alarms and spatial errors. A metric to answer this fundamental question is lacking in the mobile robotics literature.

It will be shown in this thesis, that multi-object[1] metrics developed in the target tracking community, which consider both multi-target state estimation cardinality as well as state errors and which obey the metric axioms, provide a basis for gauging robotic maps in an intuitive manner. Such set-based metrics are analyzed and described for the evaluation of feature-based maps.

The first part of this thesis is an extension of the author's paper [29], in which the COLA metric was first presented. It extends the analyses by deriving the COLA metric from the Wasserstein construction, providing a theoretical comparison of the way in which false alarms and missed detections are quantified by the OSPA and COLA metrics, deriving the map conditions which can cause them to differ, and providing more in depth results and conclusions demonstrating the complementary nature of the OSPA and COLA metrics for useful robotic map evaluation as presented in [25].

In the introduction, Section 1.1 demonstrated the difficulty in comparing feature map estimates. In this chapter, Section 2.2 presents a discussion of state-of-the-art methods for map error evaluation while Section 2.3 provides the mathematical definitions necessary to

[1] In the tracking community, a vector-based estimator is often referred to as a *single-object* estimator, where as the RFS form is referred to as a *multi-object* estimator.

quantify feature maps, for metric-based evaluation. Section 2.4 discusses the applicability of mapping metrics when no ground truth is available and Section 2.5 overviews current metrics and defines the COLA metric. Finally Section 2.6 highlights the problems associated with state-of-the-art point cloud data registration methods and introduces a solution, which overcomes these issues, based on multi-object metrics.

2.2 Robotic Mapping Error Estimation

Contrary to feature-based mapping approaches, occupancy grid methods use a predefined quantity of occupancy values and do not need to encapsulate cardinality estimation error - i.e. an estimated grid map with N cells is typically compared with a GT grid map, also with N cells. In these cases, error metrics for fixed dimension problems, such as a sum of the squared error [30, 31], or more complex errors such as "brokenness", [32] can be applied. Other work provides methods to gauge grid-based SLAM or mapping performance in robotic competitions such as search and rescue and RoboCup [33, 34].

In other mapping approaches, [35] presented the concept of assessing map quality through a binary classification of point cloud data by automatically labelling sections as plausible or suspicious, through the use of conditional random fields, but without relating the data to GT.

Various data-base websites such as [36] compare SLAM algorithms based on the error in the corrected SLAM trajectory component, without analyzing the corresponding map. A metric-based on the energy required to deform the estimated trajectory to its GT value was defined in [9]. This metric is based on an intuitive concept for comparing estimated and GT trajectories, and complies with the metric axioms. However, it will be shown in this thesis that the trajectory energy metric can contradict principled multi-object metrics which evaluate the map accuracy of the SLAM algorithm, which produced both the trajectory and the map.

In [37], a method for comparing maps suggested the use of the Hausdorff metric. Although this metric has been successfully applied in numerous vision applications for gauging the similarity between pixelated images, it suffers various problems in the way it penalizes cardinality errors between estimated and GT maps, as will be demonstrated in this thesis.

A feature-based map quality assessment method, based on searching for nearest neighbor equivalent features between an estimated and GT map was given in [38]. Map quality was assigned according to the number of associated point to point feature matches, however no penalties for false alarms and missed detections were considered.

The need for feature map metrics has also been identified in publicly available data sets, such as the European Union's FP6 Rawseeds project [39], where corner-based performance metrics are suggested for comparing estimated and GT maps.

Recent work in SLAM has suggested that a collection of map features can be modelled as a finite set, rather than a vector [7, 8]. Indeed the mathematical definitions of the four

metric axioms apply to a *set* of vectors and it will be demonstrated in this thesis that if a GT set-based map $\mathcal{M}_k$ and its set-based estimate $\widehat{\mathcal{M}}_k$, which vary with discrete time k, are modelled as *finite sets* of feature location vectors, then a mathematically consistent notion of estimation error is possible, even when the number of estimated and GT features differ. This is because, the 'distance', or error between sets, is a well understood concept. Examples include the Hausdorff distance and more recent metrics defined in the multi-target tracking literature, such as the OMAT [40] and OSPA [14] distances.

In 2015, the COLA metric was initially introduced as an alternative to the OSPA metric [29]. Instead of converting cardinality errors into distance errors, as takes place with the OSPA metric, the COLA metric converts localization errors into fractional cardinality errors. As with the OSPA metric, the COLA metric was also derived from the Wasserstein construct and is a true metric, as it obeys the four metric axioms. An analysis of this metric, together with comparisons with both the COLA and other metrics, which extends the analyses given in [29], forms the subject of Chapter 3.

Subsequently, in 2017, a multi-object metric referred to as the Generalized Optimal Sub-Pattern Assignment (GOSPA) metric was published [41]. In a manner similar to the COLA metric, the GOSPA metric is also an absolute, rather than normalized metric, however it has not been shown that it can be derived from the Wasserstein construction. It is also not clear why this metric is referred to as "Generalized" OSPA, since no proof is given to demonstrate any kind of generalization.

Finally, the Optimal Sub-Pattern Assignment(2) (OSPA(2)) metric [42, 43] was presented as an extension of the OSPA metric, for comparing trajectories over time, rather than target/map states at a particular time instant.

The pioneering work by [14] introduced the OSPA metric, based on the concepts that it should:

1. be a metric on the space of finite sets,
2. have a natural (meaningful) physical interpretation,
3. capture cardinality errors and state errors meaningfully,
4. be easily computed.

This philosophy is continued in this thesis, by analyzing current as well as the proposed COLA set-based metrics.

2.3 Map Set Definitions

Modelling a robotic feature map as a set $\mathcal{M}_k$, provides a general model, since the vectors within each set can contain any (spatial and/or description) information of relevance to the type of feature to be estimated. Further, the order in which these features are estimated, in terms of gauging overall map quality, is irrelevant. Throughout this thesis, the GT map

set $\mathcal{M}_k$ is considered to contain $\mathfrak{m}_k$ vectors $\mathbf{m}_k^{\mathrm{i}}$, $1 \leq \mathrm{i} \leq \mathfrak{m}_k$. For ease of notation and explanation, and without loss of generality, $\mathbf{m}$ will be referred to as a *spatial* variable and the time index k is now dropped. Therefore the GT map $\mathcal{M} = \{\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^{\mathfrak{m}}\}$, where $\mathbf{m}^{\mathrm{i}} \in \mathbb{R}^N \; \forall \mathrm{i} : 1 \leq \mathrm{i} \leq \mathfrak{m}$. The estimated map $\widehat{\mathcal{M}}$ is considered to contain $\widehat{\mathfrak{m}}$ vectors modelling the location of map objects - i.e. $\widehat{\mathcal{M}} = \{\widehat{\mathbf{m}}^1, \widehat{\mathbf{m}}^2, \ldots, \widehat{\mathbf{m}}^{\widehat{\mathfrak{m}}}\}$, where $\widehat{\mathbf{m}}^{\mathrm{i}} \in \mathbb{R}^N \; \forall \mathrm{i} : 1 \leq \mathrm{i} \leq \widehat{\mathfrak{m}}$. N is the dimension of the general feature space. Note that $\widehat{\mathfrak{m}}$ is itself *an estimate* of $\mathfrak{m}$ and therefore, in general, $\widehat{\mathfrak{m}} \neq \mathfrak{m}$ ($|\widehat{\mathcal{M}}| \neq |\mathcal{M}|$).

2.4 When GT is Unavailable

There are cases in SLAM in which reliable GT maps are unavailable or incomplete, such as dense visual-based maps. This thesis therefore analyses metrics which compare *all* of the features estimated by a mapping or SLAM algorithm, either with those produced by another algorithm, or GT. In the absence of GT, a possibility for evaluation is to select the most robust known algorithm X, and then if the mapping performance of algorithms Y and Z are to be compared, determine which is closer to X. Further, when GT is unavailable, the triangular inequality can be exploited under certain conditions, to at least provide bounds for the mapping error between estimated and GT maps. Assume that two estimated maps $\widehat{\mathcal{M}}_1$ and $\widehat{\mathcal{M}}_2$ are available and that the GT map is $\mathcal{M}$. If it is known that estimated map $\widehat{\mathcal{M}}_2$ is closer to GT than estimated map $\widehat{\mathcal{M}}_1$ then

$$\mathrm{d}(\widehat{\mathcal{M}}_2, \mathcal{M}) = \alpha \mathrm{d}(\widehat{\mathcal{M}}_1, \mathcal{M}) \text{ where } 0 \leq \alpha < 1. \tag{2.1}$$

Any chosen true metric $\mathrm{d}(\widehat{\mathcal{M}}_1, \widehat{\mathcal{M}}_2)$, which compares sets $\widehat{\mathcal{M}}_1$ and $\widehat{\mathcal{M}}_2$ can then be used to determine upper bounds on the error between $\widehat{\mathcal{M}}_1$ and $\mathcal{M}$ and $\widehat{\mathcal{M}}_2$ and $\mathcal{M}$. From the triangular inequality

$$\mathrm{d}(\widehat{\mathcal{M}}_1, \mathcal{M}) \leq \mathrm{d}(\widehat{\mathcal{M}}_1, \widehat{\mathcal{M}}_2) + \mathrm{d}(\widehat{\mathcal{M}}_2, \mathcal{M}) \tag{2.2}$$

From (2.1) and (2.2)

$$\mathrm{d}(\widehat{\mathcal{M}}_1, \mathcal{M}) \text{ and } \mathrm{d}(\widehat{\mathcal{M}}_2, \mathcal{M}) \leq \left(\frac{1}{1-\alpha}\right) \mathrm{d}(\widehat{\mathcal{M}}_1, \widehat{\mathcal{M}}_2), \tag{2.3}$$

meaning that upper bounds on the mapping error between $\widehat{\mathcal{M}}_1$ and $\mathcal{M}$ and $\widehat{\mathcal{M}}_2$ and $\mathcal{M}$ can be determined in terms of the error between the map estimates of the two algorithms.

If a third map estimation algorithm is introduced which yields $\widehat{\mathcal{M}}_3$, the above analysis can be extended to give

$$\mathrm{d}(\widehat{\mathcal{M}}_3, \mathcal{M}) \leq \mathrm{d}(\widehat{\mathcal{M}}_3, \widehat{\mathcal{M}}_1) + \left(\frac{1}{1-\alpha}\right) \mathrm{d}(\widehat{\mathcal{M}}_1, \widehat{\mathcal{M}}_2). \tag{2.4}$$

Note that since a GT map $\mathcal{M}$ is not required in the RHS of (2.3) and (2.4), the map feature representations are not restricted to ones knowledge of GT, and can include dense (e.g., visual) feature representations. Note here the importance of the chosen map metric being a true metric, such that the 4 axioms are obeyed. It is also desirable that α be as small as possible.

From here on, comparisons will be referred to as being between estimated and GT maps, although the concepts equally apply to comparisons of multiple map estimates.

2.5 Metrics for Comparing Sets

2.5.1 The Hausdorff metric

Definition of the Hausdorff Metric

The Hausdorff metric calculates the degree of difference between the two sets of vectors $\mathcal{M}$ and $\widehat{\mathcal{M}}$ for the general case when $|\mathcal{M}|$ and $|\widehat{\mathcal{M}}|$ may differ. Formally, the Hausdorff distance from a finite nonempty set $\mathcal{M}$ to another finite nonempty set $\widehat{\mathcal{M}}$ is a *maximin* function, defined as:

$$\begin{aligned} \mathrm{d}_H(\mathcal{M}, \widehat{\mathcal{M}}) = \quad & \max\{\max_{\mathbf{m}\in\mathcal{M}} \min_{\widehat{\mathbf{m}}\in\widehat{\mathcal{M}}} \mathrm{d}(\mathbf{m}, \widehat{\mathbf{m}}), \\ & \max_{\widehat{\mathbf{m}}\in\widehat{\mathcal{M}}} \min_{\mathbf{m}\in\mathcal{M}} \mathrm{d}(\widehat{\mathbf{m}}, \mathbf{m})\} \end{aligned} \tag{2.5}$$

where $\mathrm{d}(\mathbf{m}, \widehat{\mathbf{m}})$ is the Euclidean distance between $\mathbf{m}$ and $\widehat{\mathbf{m}}$.

Intuitive Map Quality Issues with the Hausdorff Metric

The Hausdorff metric has been applied extensively in computer vision analyses as a measure of similarity of pixelated images. Examples include checking if a template image is present in a test image [44], object matching between images [45] and automatic target recognition [46]. The metric has also been applied to the evaluation of multi-target data fusion algorithms [47, 48].

In the work of Hoffman and Mahler, it was noted that although the Hausdorff distance provides a useful assessment of the overall localization performance of multi-target estimation algorithms, it lacks the ability to penalize false target estimates, and tends to over-penalize single estimate outliers [40][2]. These same issues affect the Hausdorff distance's ability to provide a meaningful and intuitive assessment of robotic map quality, as will be demonstrated in Section 3.7. The inability to provide a meaningful measure of multi-target estimate quality, motivated the introduction of the OMAT metric, explained next.

2.5.2 The OMAT metric

Definition of the OMAT metric

Hoffman and Mahler generalized the concept of a miss-distance in which they derived the OMAT metric with power p is defined as [40]

$$\mathrm{d}^p_{\mathrm{OMAT}}(\mathcal{M}, \widehat{\mathcal{M}}) = \min_{\mathbf{C}_{\mathfrak{m}\times\widehat{\mathfrak{m}}}} \left(\sum_{\mathrm{i}=1}^{\mathfrak{m}} \sum_{j=1}^{\widehat{\mathfrak{m}}} c_{\mathrm{i},j} \mathrm{d}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j})^p \right)^{1/p} \tag{2.6}$$

[2]It should be noted that there are generalizations of the Hausdorff metric which are able to specifically avoid the outlier problem [49].

where $\mathbf{C}_{\mathfrak{m}\times\widehat{\mathfrak{m}}} \in \mathbb{R}^{\mathfrak{m}\times\widehat{\mathfrak{m}}}$ is the *transportation matrix* with elements $c_{\mathrm{i},j}$. The OMAT metric determines the matrix $\mathbf{C}_{\mathfrak{m}\times\widehat{\mathfrak{m}}}$, which minimizes the objective function in (2.6), where the chosen metric $\mathrm{d}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j})$ represents the unit cost of transporting $c_{\mathrm{i},j}$ (quantity of) weights from $\mathbf{m}^{\mathrm{i}}$ to $\widehat{\mathbf{m}}^{j}$ for features $\mathrm{i} = 1, 2, ..., \mathfrak{m}$ from the ground truth map $\mathcal{M}$ to the features $j = 1, 2, ..., \widehat{\mathfrak{m}}$ in the estimated map $\widehat{\mathcal{M}}$, subject to the constraints given by

$$\sum_{j=1}^{\widehat{\mathfrak{m}}} c_{\mathrm{i},j} = \frac{1}{\mathfrak{m}} \;\; 1 \leq \mathrm{i} \leq \mathfrak{m} \quad \text{and} \quad \sum_{\mathrm{i}=1}^{\mathfrak{m}} c_{\mathrm{i},j} = \frac{1}{\widehat{\mathfrak{m}}} \;\; 1 \leq j \leq \widehat{\mathfrak{m}}, \tag{2.7}$$

which is a generalized distance between two probability distributions on some metric space $\mathcal{M}$. The OMAT metric was introduced in a tracking context when Mahler *et al* addressed the data association problem in the context of multi-object estimation [50].

Intuitive Explanation of the OMAT Metric

The OMAT metric calculates the *fractional* assignment of features from set $\mathcal{M}$ to $\widehat{\mathcal{M}}$ and can be defined as a balanced transportation problem, since the assignment can be described in terms of the total supply from set $\mathcal{M}$ being equal to the total demand required by set $\widehat{\mathcal{M}}$ [51]. When $p = 1$, the transportation problem is also known as the "earth mover's distance" [52], used in image processing.

Limitations of the OMAT Metric

While the OMAT metric has been shown to partially solve the outlier problem associated with the Hausdorff metric, various other problems exist, which will be demonstrated for theoretical and real estimated map cases in Sections 3.7 and 5.2.1. Further, the OMAT metric is mathematically undefined when either $\mathcal{M}$ or $\widehat{\mathcal{M}} = \emptyset$.

2.5.3 The OSPA metric

In 2008, [14] introduced the OSPA metric, which obeys the metric axioms and improves most of the problems of the OMAT metric, as will be shown in Section 3.7. It has recently gained popularity in gauging target tracking [53, 54, 55, 56] as well as robotic mapping performance [8, 7].

Definition of the OSPA Metric

The OSPA metric $\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})$ with power p and cut-off parameter c, for $\widehat{\mathfrak{m}} > \mathfrak{m}$, is defined as:

$$\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}}) = \left(\frac{1}{\widehat{\mathfrak{m}}}\min_{\sigma}\sum_{\mathrm{i}=1}^{\widehat{\mathfrak{m}}}\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})})^p\right)^{1/p} \tag{2.8}$$

$$= \left(\frac{1}{\widehat{\mathfrak{m}}}\left(\min_{\sigma}\sum_{\mathrm{i}=1}^{\mathfrak{m}}\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})})^p + c^p(\widehat{\mathfrak{m}}-\mathfrak{m})\right)\right)^{1/p} \tag{2.9}$$

where σ is a permutation of the set $\{1,\ldots,\mathfrak{m}\}$ which minimizes $\left(\sum_{\mathrm{i}=1}^{\mathfrak{m}}\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})})^p\right)$, $1 \leq p < \infty$ and the cut-off parameter $c > 0$. The assignment $\sigma(\mathrm{i})$ can be determined via an optimal assignment method, such as the Hungarian method, [57, 58]. If both sets are empty, $\mathfrak{m} = \widehat{\mathfrak{m}} = 0$, $\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}}) = 0$. For $\mathfrak{m} > \widehat{\mathfrak{m}}$, the metric is defined as $\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\widehat{\mathcal{M}},\mathcal{M})$. The distance $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})})$ is defined as

$$\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})}) = \min(c,\mathrm{d}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})})) \tag{2.10}$$

where $\mathrm{d}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})})$ is any metric distance (e.g., Euclidean, Mahalanobis, Hellinger) between $\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})}$.

Intuitive Explanation of the OSPA Metric

For $\widehat{\mathfrak{m}} > \mathfrak{m}$ and $p = 1$, the first term of the RHS of (2.9), determines individual assignments between all $\mathfrak{m}$ of the feature location vectors within $\mathcal{M}$ and a subset of dimension $\mathfrak{m}$ of the feature vectors within $\widehat{\mathcal{M}}$. Due to (2.10), each assignment is given a value equal to its (possibly statistical) distance up to a maximum value of c (statistical) distance units. The remaining $\widehat{\mathfrak{m}} - \mathfrak{m}$ features in $\widehat{\mathcal{M}}$ which were not assigned, constitute a cardinality error (possible false alarms). Each of these is penalized with the maximal distance error c, hence yielding the RHS residual error $c(\widehat{\mathfrak{m}} - \mathfrak{m})$ in (2.9). To comply with other metrics (L_2 norm etc.) the assigned distance values can be raised to the general power p. Intuitively, when the (statistical) distance between feature i and its assigned feature $\sigma(\mathrm{i})$ reaches a maximum value (c), feature i becomes a cardinality error, contributing one more fixed distance error c^p. Hence the OSPA metric yields a measure of the difference between $\widehat{\mathcal{M}}$ and $\mathcal{M}$ in units of distance. From (2.9) and (2.10), $\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}})$ has minimum value zero and saturates to a maximum value c for all $\mathcal{M}$ and $\widehat{\mathcal{M}}$.

Selecting the c and p parameters in the OSPA metric

In [14], the cut-off parameter c can be determined according to the question: What distance c should be used to establish whether or not an estimated feature is good? Based on this, the selection of the parameter c is not unique and depends on the designer. A possible recipe for this question is to find the cut-off c, which establishes a good balance to penalize the combination of detection and cardinality errors. Section 5.2 studies several intuitive examples related to this point. It utilizes the Mahalanobis (statistical) distance as an inner metric.

Therefore, the cut-off c is defined as the validation gate, corresponding to a probability interval within a chi-square test, within which an estimated feature is considered to correspond to a GT feature. Finally, selecting the power p determines how sensitive the OSPA metric is to penalize outlier estimates. In general, if $p = 1$, the OSPA metric corresponds to the sum of cardinality and detection errors. However, this thesis will use $p = 2$ since it yields smooth errors and is more in-line with other metrics, often L2-norms.

2.6 Point Cloud Data Registration

Previous work has demonstrated that if the correspondence between two point clouds is known, then the optimal transformation can be computed as a closed-form solution [59] [60]. However, in real scenarios, the correspondence between points is usually unknown. This variation is known in the literature as correspondence and pose estimation [61] since it estimates an optimal transformation and a robust correspondence between the point clouds. The ICP [15] algorithm and its variants [62] [63] historically provided a simple conceptual solution to this problem. However, ICP has limitations due to the assumption that the closest pairs define the correct correspondence. This limitation often causes ICP to fail for various reasons. One possibility is that there are many outliers in the point clouds, which require complementary methods to remove them, e.g., the RANSAC algorithm [23], or the GORE algorithm [24]. If the point clouds are not initially approximately aligned, ICP is susceptible to local minima solutions producing inaccurate estimates. Finally, ICP uses an L_2 metric to quantify registration error, which cannot penalize detection errors due to outliers.

An ICP variant, Global ICP (Go-ICP) [64], is considered one of the first methods to solve the point registration problem robustly. However, due to the Branch and Bound (BnB) strategy for finding optimal global registration solutions, it can make the problem computationally expensive. Also, by maintaining the L_2 metric as a cost function, this method inherits some of the disadvantages of ICP, when outliers are present.

In [14], [25], the multi-object OSPA and COLA metrics were proposed as solutions to the evaluation of errors between two sets, which are directly applicable to two different point cloud data sets such as $\mathcal{M}$ and $\widehat{\mathcal{M}}$. These multi-object metrics have applications in assessing multi-target tracking and robotic mapping algorithms in the presence of both state and state cardinality errors. As was shown in [25], the OSPA and COLA metrics can differ in their evaluations when comparing two estimated map/target sets with a third ground truth map/target set. In contrast, in the case of point cloud registration, a metric is required, which calculates the difference between a model set and its transformed reference set, so that this difference can be minimized to yield an optimal estimated transformation between them. Therefore, in the case of point cloud registration, both the OSPA and COLA metrics yield the same estimated transform, as will be proved in Section 4.2 and therefore without loss of generality, this thesis applies the COLA metric.

In related work [65], extended the use of multi-object metrics to the evaluation of estimated shape errors in image processing. For the direct comparison of point-based data sets however, the COLA (and OSPA) metrics remain of relevance.

Registration approaches are split into local and global methods. In each case, the following literature review corresponds to joint pose and correspondence estimation.

2.6.1 Local registration methods.

As mentioned above, ICP [15] and its variants [62] [63] are known as local registration methods. However, these have several problems in the presence of outliers. Probabilistic approaches such as the Normal Distributed Transform (NDT) algorithm [66] and Gaussian mixture (GM)-based registration methods [67] have been proposed to improve the robustness of ICP. Based on the Kullback-Leibler divergence between two GMs, the aforementioned methods demonstrate improvements in robustness to outliers. However, the optimization is based on local search, providing solutions susceptible to local minima.

2.6.2 Global registration methods.

Currently, there are a variety of solutions to the global registration problem using BnB techniques. Go-ICP [64] is one of the first methods that proposed this solution. It presents a global variant of the ICP algorithm using the BnB technique. However, this method is sensitive to occlusions due to its non-robust cost function, the L_2 norm. For outlier removal, Go-ICP uses trimming strategies. However, these trimming strategies increase the algorithm's computational time exponentially. Another algorithm by Li and Hartley, also based on BnB, proposes an algorithm with increased search speed by using simpler bounding functions on the L_2 metric for error minimization [68]. This method is however still unable to penalize cardinality errors.

In a manner similar to Go-ICP, the Globally Optimal Gaussian Mixture Alignment (GOGMA) algorithm [69] also uses a BnB strategy to determine the optimal transformation between point clouds. However, in contrast to directly applying ICP to the point cloud data sets for local minimization of the objective function, GOGMA first converts the point cloud data sets into Gaussian mixtures and then aligns the mixtures based on the authors' definition of an L_2 metric between the mixtures. BnB is then used to achieve global minimization. This method can yield significant improvements in performance over its Go-ICP predecessor, however it can also fail to terminate due to excessive memory requirements. This is because GOGMA generates a priority queue consisting of all current branches connecting the prior transformation estimate with all nearest possible transformation estimates. This process requires large memory requirements and it is possible that the priority queue grows faster than it can be pruned, resulting in intractability. Because of this problem, this algorithm has been discarded for comparison with the PSO-COLA algorithm.

The Truncated least squares Estimation And SEmidefinite Relaxation (TEASER) algorithm [70] proposes a method that decouples the rotation and translation estimates. Using a truncated least-squares cost function, the authors show improvements in the presence of data outliers. With even only a few given correct point correspondences, TEASER is shown to converge to a global minimum, even when a low number of correct correspondences is made.

In the sense of RANSAC [23], the 4-Points Congruent Sets (4PCS) [71] algorithm has been proposed. This method obtains a transformation hypothesis by matching four congruent coplanar points. Due to the quadratic time complexity of 4PCS, Super 4-Points Congruent Sets (Super4PCS) [72] is presented as a solution that improves the time complexity. However, the search for congruent points between the point clouds is not guaranteed to converge in the presence of noise.

The Fast Descriptors and Correspondence Propagation (FastDesp) algorithm [73] has been proposed as an efficient solution using descriptors computed from eigenvectors and normals computed at multiple scales. Results show promising applications in large point cloud datasets. However, when performing correspondence propagation, which is the process of adding new possible correspondences to an initial correspondence set, it uses the nearest neighbor algorithm, which ignores the joint compatibility of the data.

PHASER presents a point cloud registration solution in the context of the Fourier transform [74]. This solution exhibits accurate convergence to a global solution, however assumes only spatial point cloud data errors, while ignoring the possibility of detection errors (i.e. missing data and/or false alarms).

2.6.3 Preprocessing Point Cloud Data

This section describes point cloud data preprocessing, for the purpose of reducing the complexity of data registration. In general, point cloud data sets contain a huge number of points and their attributes. Therefore, a great deal of research addresses the problem of data reduction without eliminating important information. In the following, the most commonly used techniques for data reduction will be introduced, namely down-sampling and the use of data descriptor-based methods.

Data Reduction-Based on Down-Sampling

There has been extensive research in the field of data down-sampling and the intention of this section is to explain the classical techniques.

The Random Sample Filter randomly samples the point cloud according to a uniform probability distribution, while preserving the spatial densities within the data [75]. The computational complexity of this method is $\mathcal{O}(n)$, where n is the number of points in the point cloud.

The Octree method on the other hand, starts by constructing the cube of minimum volume which contains the entire point cloud data set [76]. Then the algorithm iterates, and the cube is sub-divided into 8 equally sized cubes, called the children. After every iteration, it is verified if the child cubes are filled. Empty cubes are pruned and the subdivision stops if a certain discretization level is reached. Finally, the subsampled point data set becomes the set of points corresponding to the centers of each filled child cube.

The Uniform Grid Sampling Filter uses a voxelized grid approach [77]. This technique

uses 3D spatial voxels, which can contain any number of points within the point cloud. Every point within a particular voxel will be replaced by a single point, which is closest to the center of that voxel. Therefore, the data reduction in this case is sensitive to the voxel grid size.

In this thesis, the PSO-COLA registration algorithm will initially be applied to the Stanford Bunny and Dragon datasets from the Stanford 3D Scanning Repository [26]. This is because these datasets are recorded from a single object so that the PSO-COLA registration algorithm can be tested in the absence of possible occlusions. These datasets have been obtained from a high precision Cyberware 3030 MS laser stripe optical triangulation scanner, to which controlled amounts of spatial and detection errors can be added. Due to the high number of points in the raw Bunny and Dragon data sets, subsampling is necessary. To subsample these data sets, the Uniform Grid Sampling Filter method will be used, due to its superior performance over Octree and the Random Sample Filter in preserving the shape of the datasets.

Data Reduction-Based on Descriptors

Descriptor-based methods for modelling point cloud data sets are also numerous in the literature. In general these techniques are useful for computing correspondences between points.

The Signature of Histograms of OrienTations (SHOT) algorithm is a local 3D descriptor for surface matching [78]. It is rotation invariant and robust to spatial measurement noise. The 3D SHOT descriptors are split into two categories, namely "Signatures" and "Histograms". The novelty of this method is the signature descriptor, which provides a description of a given point with respect to its 3D surface neighborhood.

The Fast Point Feature Histograms (FPFH) descriptor [79], has been published as an improvement over its Point Feature Histograms (PFH) descriptor predecessor [80]. FPFH descriptors contain information describing the local geometry around a given point in 3D point cloud datasets. The computational complexity of this algorithm is given by $\mathcal{O}(n \times k)$, where n is the number of points in the PCD, and k is the number of its neighbors, within a predefined spatial region. Due to this low computational complexity, this algorithm can be used for computing the point correspondences necessary for 3D registration.

To apply the PSO-COLA registration algorithm in the presence of possible occlusions, such as environments with walls, corners and corridors, in which a mobile robot may need to navigate, the "ETH-Apartment" dataset will be used [27]. In the case of the ETH-Apartment data set, data reduction is also necessary before the PSO-COLA, and most other, registration algorithms can be applied in a tractable manner. In contrast to the Stanford Bunny and Dragon datasets descriptor-based data reduction, as opposed to subsampling, will be used. This is because, when recorded from a moving vehicle, consecutive scans can vary significantly due to occlusions. For example, this occurs when passing from one room to another. In this case, descriptors have the potential to improve the speed and accuracy of the PSO-COLA algorithm. Due to the low computational complexity necessary to generate them, FPFH descriptors will be applied.

Chapter 3

The Cardinalized Optimal Linear Assignment metric

3.1 Introduction

In this chapter the COLA metric is presented, for evaluating multi-object error estimation. In particular, the COLA metric is a class of Wasserstein metric, which obeys the four metric axioms, and measures dissimilarity between two probability distributions. Additionally, the COLA metric is introduced for improving some of the limitations of the state of the art OSPA metric in robotic mapping. Specifically, OSPA saturates to c when all of the localization errors are larger than c, and is then insensitive to the size of the map cardinality error. Further, it will be shown that unlike the OSPA metric, the COLA metric penalizes missed detections and false alarms in an equal and symmetric manner.

3.2 Definition of the COLA Metric

The COLA metric can be derived from the Wasserstein construction as shown in Section 3.3. For $\widehat{\mathfrak{m}} \geq \mathfrak{m}$, it is defined as

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}) = \left(\min_{\sigma} \sum_{\mathrm{i}=1}^{\widehat{\mathfrak{m}}} \left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})}{c} \right)^{p} \right)^{1/p} \tag{3.1}$$

$$= \left(\min_{\sigma} \sum_{\mathrm{i}=1}^{\mathfrak{m}} \left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})}{c} \right)^{p} + (\widehat{\mathfrak{m}} - \mathfrak{m}) \right)^{1/p} \tag{3.2}$$

where, σ, p, c and $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})$ carry the same definitions as the OSPA metric. For $\mathfrak{m} > \widehat{\mathfrak{m}}$, the metric is defined as $\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\widehat{\mathcal{M}}, \mathcal{M})$. Section 3.4 proves that $\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\widehat{\mathcal{M}}, \mathcal{M})$ is a true metric.

3.3 Derivation of the COLA Metric

Consider two probability densities $f(\mathcal{M})$ and $g(\widehat{\mathcal{M}})$ in the n-Euclidean space. By definition, the Wasserstein distance calculates the similarity between f and g and is given by:

$$\mathrm{d}_w^p(f,g) := \min_h \left(\int \int \mathrm{d}(\mathbf{m},\widehat{\mathbf{m}})^p \, h(\mathbf{m},\widehat{\mathbf{m}}) \mathrm{d}\mathbf{m}\mathrm{d}\widehat{\mathbf{m}} \right)^{1/p}, \tag{3.3}$$

where $1 \leq p < \infty$, $\mathrm{d}(\mathbf{m},\widehat{\mathbf{m}})$ is any distance metric between $\mathbf{m}$ and $\widehat{\mathbf{m}}$, and $h(\mathbf{m},\widehat{\mathbf{m}})$ is any joint distribution whose marginals are $f(\mathbf{m})$ and $g(\widehat{\mathbf{m}})$ such that:

$$\int h(\mathbf{m},\widehat{\mathbf{m}})\mathrm{d}\widehat{\mathbf{m}} = f(\mathbf{m}), \quad \int h(\mathbf{m},\widehat{\mathbf{m}})\mathrm{d}\mathbf{m} = g(\widehat{\mathbf{m}}). \tag{3.4}$$

Consider the marginal distributions in (3.4) to be:

$$f(\mathbf{m}) = \frac{1}{\mathfrak{m}} \sum_{\mathrm{i}=1}^{\mathfrak{m}} \delta(\mathbf{m} - \mathbf{m}^{\mathrm{i}}), \quad g(\widehat{\mathbf{m}}) = \frac{1}{\widehat{\mathfrak{m}}} \sum_{j=1}^{\widehat{\mathfrak{m}}} \delta(\widehat{\mathbf{m}} - \widehat{\mathbf{m}}^{j}) \tag{3.5}$$

where δ is the Dirac delta function. This implies that the joint distribution $h(\mathbf{m},\widehat{\mathbf{m}})$ is given by

$$h(\mathbf{m},\widehat{\mathbf{m}}) = \sum_{\mathrm{i}=1}^{\mathfrak{m}} \sum_{j=1}^{\widehat{\mathfrak{m}}} c_{\mathrm{i},j} \delta(\mathbf{m} - \mathbf{m}^{\mathrm{i}}) \delta(\widehat{\mathbf{m}} - \widehat{\mathbf{m}}^{j}). \tag{3.6}$$

The derivation of the COLA metric follows a similar procedure as the OSPA metric [14]. A unique assignment coefficient

$$c_{\mathrm{i},j} = \delta_{j,\sigma(\mathrm{i})} / \max(\mathfrak{m},\ \widehat{\mathfrak{m}}), \tag{3.7}$$

which satisfies (3.6), is used where $\sigma(\mathrm{i})$ is a permutation of the larger set and $\delta_{j,\sigma(\mathrm{i})} = 1$ iff $j = \sigma(\mathrm{i})$ and 0 otherwise.

Consider a GT map $\mathcal{M}' = \{\mathbf{m}^1, \ldots, \mathbf{m}^{\mathfrak{m}}\}$, and its estimated map $\widehat{\mathcal{M}} = \{\widehat{\mathbf{m}}^1, \ldots, \widehat{\mathbf{m}}^{\widehat{\mathfrak{m}}}\}$. The COLA metric can then be derived from (3.3) by replacing

$$\mathrm{d}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{j}) = \frac{(\max(\mathfrak{m},\ \widehat{\mathfrak{m}}))^{1/p}}{c} \mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{j}). \tag{3.8}$$

Note that if $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{j})$ is a metric, then $\mathrm{d}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{j})$ is also guaranteed to be a metric as required in (3.3). For $\widehat{\mathfrak{m}} \geq \mathfrak{m}$, replacing $c_{\mathrm{i},j}$ in (3.6) with (3.7) and then substituting the result, together with (3.8) into (3.3) yields a new metric

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}}) = \left(\min_{\sigma} \sum_{\mathrm{i}=1}^{\widehat{\mathfrak{m}}} \left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})})}{c} \right)^p \right)^{1/p}, \tag{3.9}$$

where $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})})$ is defined in the OSPA sense, with cut-off parameter c. For $\mathfrak{m} \geq \widehat{\mathfrak{m}}$, the metric is defined as $\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\widehat{\mathcal{M}},\mathcal{M})$.

3.4 Proof that $\mathrm{d}_{\mathbf{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})$ is a true Metric

The proof that $\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})$ is a metric follows a similar procedure to the the proof that $\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})$ is a metric in [14]. $\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}) \geq 0$ for all $\mathcal{M}, \widehat{\mathcal{M}}$ because metric $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\mathrm{j}}) \geq 0$ for all $\mathbf{m}^{\mathrm{i}}$ and $\widehat{\mathbf{m}}^{\mathrm{j}}$. Similarly $\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}) = 0$ iff $\widehat{\mathcal{M}} = \mathcal{M}$ - proof: From (3.2), if $\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}) = 0$, then

$$\min_{\sigma} \sum_{\mathrm{i}=1}^{\mathfrak{m}} \frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})^p}{c^p} = -(\widehat{\mathfrak{m}} - \mathfrak{m}) \leq 0, \tag{3.10}$$

since (3.2) is defined for $\widehat{\mathfrak{m}} \geq \mathfrak{m}$. The RHS of (3.10) therefore implies that $\widehat{\mathfrak{m}} = \mathfrak{m}$ (since the LHS is positive), meaning that $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})}) = 0 \quad \forall \mathrm{i}$. $\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}) = \mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\widehat{\mathcal{M}}, \mathcal{M})$ because $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\mathrm{j}})$ satisfies the symmetry property. It remains to be verified that the triangle inequality is satisfied.

Consider the set $\widehat{\mathcal{N}} = \{\widehat{\mathbf{n}}^1, \ldots, \widehat{\mathbf{n}}^{\widehat{\mathfrak{n}}}\}$, with cardinality $\widehat{\mathfrak{n}} \in \mathbb{N}_0$. Consider the following sets of dummy points $\mathcal{U} = \{\mathbf{u}^{\mathrm{i}}\}_{\mathrm{i} \in \mathbb{N}_0}$ and $\mathcal{V} = \{\mathbf{v}^{\mathrm{j}}\}_{\mathrm{j} \in \mathbb{N}_0}$ in $\mathbb{R}^N$ where

$$\mathrm{d}(\mathbf{u}^{\mathrm{i}}, \mathbf{x}) \geq c, \ \mathrm{d}(\mathbf{v}^{\mathrm{i}}, \mathbf{x}) \geq c, \ \mathrm{d}(\mathbf{u}^{\mathrm{i}}, \mathbf{v}^{\mathrm{j}}) \geq c \quad \forall\, \mathbf{x}, \ \mathrm{i}, \mathrm{j}.$$

Case 1: ($\mathfrak{m} \leq \widehat{\mathfrak{m}} \leq \widehat{\mathfrak{n}}$): In order to raise the cardinality of sets $\mathcal{M}$ and $\widehat{\mathcal{M}}$ to $\widehat{\mathfrak{n}}$, consider the following dummy points:

$$\mathbf{m}^{\mathfrak{m}+\mathrm{i}} = \mathbf{u}^{\mathrm{i}}, \quad 1 \leq \mathrm{i} \leq \widehat{\mathfrak{n}} - \mathfrak{m} \tag{3.11}$$

$$\widehat{\mathbf{m}}^{\widehat{\mathfrak{m}}+j} = \mathbf{v}^{j}, \quad 1 \leq \mathrm{j} \leq \widehat{\mathfrak{n}} - \widehat{\mathfrak{m}} \tag{3.12}$$

Then choose $\sigma, \tau \in \Pi_{\widehat{\mathfrak{n}}}$ such that,

$$\min_{\pi \in \Pi_{\widehat{\mathfrak{n}}}} \sum_{\mathrm{i}=1}^{\widehat{\mathfrak{n}}} \left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{n}}^{\pi(\mathrm{i})})}{c} \right)^p = \sum_{\mathrm{i}=1}^{\widehat{\mathfrak{n}}} \left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{n}}^{\sigma(\mathrm{i})})}{c} \right)^p \tag{3.13}$$

$$\min_{\pi \in \Pi_{\widehat{\mathfrak{n}}}} \sum_{\mathrm{i}=1}^{\widehat{\mathfrak{n}}} \left(\frac{\mathrm{d}^{(c)}(\widehat{\mathbf{n}}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\pi(\mathrm{i})})}{c} \right)^p = \sum_{\mathrm{i}=1}^{\widehat{\mathfrak{n}}} \left(\frac{\mathrm{d}^{(c)}(\widehat{\mathbf{n}}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\tau(\mathrm{i})})}{c} \right)^p \tag{3.14}$$

$$\therefore \quad \mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}}) = \left(\min_{\pi\in\Pi_{\mathfrak{m}}}\sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{m}}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{m}}^{\pi(\mathfrak{i})})}{c}\right)^{p}\right)^{1/p} \tag{3.15}$$

$$\leq \left(\min_{\pi\in\Pi_{\mathfrak{m}}}\sum_{\mathfrak{i}=1}^{\mathfrak{m}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{m}}^{\pi(\mathfrak{i})})}{c}\right)^{p} + (\widehat{\mathfrak{n}}-\mathfrak{m})\right)^{1/p} \tag{3.16}$$

$$\leq \left(\min_{\pi\in\Pi_{\widehat{\mathfrak{n}}}}\sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{n}}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{m}}^{\pi(\mathfrak{i})})}{c}\right)^{p}\right)^{1/p} \tag{3.17}$$

$$\leq \left(\sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{n}}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{n}}^{\sigma(\mathfrak{i})}) + \mathrm{d}^{(c)}(\widehat{\mathbf{n}}^{\sigma(\mathfrak{i})},\widehat{\mathbf{m}}^{\tau(\sigma(\mathfrak{i}))})}{c}\right)^{p}\right)^{1/p} \tag{3.18}$$

$$\leq \left(\sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{n}}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{n}}^{\sigma(\mathfrak{i})})}{c}\right)^{p}\right)^{1/p} + \left(\sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{n}}}\left(\frac{\mathrm{d}^{(c)}(\widehat{\mathbf{n}}^{\sigma(\mathfrak{i})},\widehat{\mathbf{m}}^{\tau(\sigma(\mathfrak{i}))})}{c}\right)^{p}\right)^{1/p} \tag{3.19}$$

$$\leq \mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{N}}) + \mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\widehat{\mathcal{N}},\widehat{\mathcal{M}}). \tag{3.20}$$

In (3.16) $\widehat{\mathfrak{n}}-\mathfrak{m}$ dummy points were added to the set $\mathcal{M}$ yielding (3.17). In (3.17) the triangular inequality on the metric $\mathrm{d}^{(c)}$ and the application of (3.13) and (3.14) resulted in (3.18). Finally, Minkowski's inequality yielded (3.19).

Case 2: $(\mathfrak{m},\widehat{\mathfrak{n}}\leq\widehat{\mathfrak{m}})$: In order to raise the cardinality of sets $\mathcal{M}$ and $\widehat{\mathcal{N}}$ to $\widehat{\mathfrak{m}}$, consider the following dummy points:

$$\mathbf{m}^{\widehat{\mathfrak{m}}-\mathfrak{i}+1} = \mathbf{u}^{\mathfrak{i}}, \quad 1\leq\mathfrak{i}\leq\widehat{\mathfrak{m}}-\mathfrak{m} \tag{3.21}$$

$$\widehat{\mathbf{n}}^{\widehat{\mathfrak{m}}-\mathfrak{j}+1} = \mathbf{u}^{j}, \quad 1\leq\mathfrak{j}\leq\widehat{\mathfrak{m}}-\widehat{\mathfrak{n}} \tag{3.22}$$

where $\mathrm{d}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{n}}^{\mathfrak{i}})=0, \quad \max(\mathfrak{m},\widehat{\mathfrak{n}})\leq\mathfrak{i}\leq\widehat{\mathfrak{m}}$.

Then choose $\sigma,\tau\in\Pi_{\widehat{\mathfrak{m}}}$ such that,

$$\min_{\pi\in\Pi_{\mathfrak{m}\vee\widehat{\mathfrak{n}}}}\sum_{\mathfrak{i}=1}^{\mathfrak{m}\vee\widehat{\mathfrak{n}}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{n}}^{\pi(\mathfrak{i})})}{c}\right)^{p} =$$

$$\min_{\pi\in\Pi_{\widehat{\mathfrak{m}}}}\sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{m}}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{n}}^{\pi(\mathfrak{i})})}{c}\right)^{p} = \sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{m}}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{n}}^{\sigma(\mathfrak{i})})}{c}\right)^{p} \tag{3.23}$$

$$\min_{\pi\in\Pi_{\widehat{\mathfrak{m}}}}\sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{m}}}\left(\frac{\mathrm{d}^{(c)}(\widehat{\mathbf{n}}^{\mathfrak{i}},\widehat{\mathbf{m}}^{\pi(\mathfrak{i})})}{c}\right)^{p} = \sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{m}}}\left(\frac{\mathrm{d}^{(c)}(\widehat{\mathbf{n}}^{\mathfrak{i}},\widehat{\mathbf{m}}^{\tau(\mathfrak{i})})}{c}\right)^{p} \tag{3.24}$$

where $\mathfrak{m}\vee\widehat{\mathfrak{n}}=\max(\mathfrak{m},\widehat{\mathfrak{n}})$. Therefore, finally

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}}) = \left(\min_{\pi\in\Pi_{\widehat{\mathfrak{m}}}}\sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{m}}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{m}}^{\pi(\mathfrak{i})})}{c}\right)^{p}\right)^{1/p} \tag{3.25}$$

$$\leq \left(\sum_{\mathfrak{i}=1}^{\widehat{\mathfrak{m}}}\left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathfrak{i}},\widehat{\mathbf{n}}^{\sigma(\mathfrak{i})}) + \mathrm{d}^{(c)}(\widehat{\mathbf{n}}^{\sigma(\mathfrak{i})},\widehat{\mathbf{m}}^{\tau(\sigma(\mathfrak{i}))})}{c}\right)^{p}\right)^{1/p} \tag{3.26}$$

$$\leq \left(\sum_{\mathrm{i}=1}^{\widehat{\mathfrak{m}}} \left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{n}}^{\sigma(\mathrm{i})})}{c} \right)^p \right)^{1/p}$$

$$+ \left(\sum_{\mathrm{i}=1}^{\widehat{\mathfrak{m}}} \left(\frac{\mathrm{d}^{(c)}(\widehat{\mathbf{n}}^{\sigma(\mathrm{i})}, \widehat{\mathbf{m}}^{\tau(\sigma(\mathrm{i}))})}{c} \right)^p \right)^{1/p} \tag{3.27}$$

$$\leq \mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{N}}) + \mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\widehat{\mathcal{N}}, \widehat{\mathcal{M}}). \tag{3.28}$$

In (3.25) the triangular inequality on the metric $\mathrm{d}^{(c)}$ and the application of (3.23) and (3.24) resulted in (3.26). Again, Minkowski's inequality [1] yielded (3.27).

3.5 Intuitive Explanation of the COLA Metric

Whereas the OSPA metric has the units of localization error (i.e. distance), the COLA metric has the units of the cardinality error (i.e. no units). In contrast to the OSPA metric, when the distance between an assigned feature i and feature $\sigma(\mathrm{i})$ *decreases* to c it changes from a cardinality error to a fractional cardinality error $\left(\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})/c\right)^p$. Although the difference between the OSPA and COLA metrics may seem trivial, Sections 3.7, 3.8 will demonstrate significant differences in the intuitive behavior of the COLA metric over its OSPA counterpart, when evaluating feature maps.

For $\widehat{\mathfrak{m}} > \mathfrak{m}$, the first term of the RHS of (3.2) again determines individual assignments between all $\mathfrak{m}$ of the feature location vectors within $\mathcal{M}$ and a subset of dimension $\mathfrak{m}$ of the feature vectors within $\widehat{\mathcal{M}}$. Now however, due to (2.10), this term is fractional if $\mathrm{d}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})}) < c$ or unity otherwise, in which case it is effectively added as a single cardinality error on to the pure cardinality error $(\widehat{\mathfrak{m}} - \mathfrak{m})$ on the RHS of (3.2). In contrast to the OSPA metric, it can be seen from (3.2) and (2.10) that $\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})$ has minimum value zero and maximum value $(\widehat{\mathfrak{m}})^{1/p}$ if $\widehat{\mathfrak{m}} > \mathfrak{m}$ or $(\mathfrak{m})^{1/p}$ otherwise.

3.6 Interpreting the Components of the COLA Metric

In [14], it was shown that the OSPA error contains two components, which separately account for localization and cardinality errors. This also applies to the COLA metric with individual components $\mathrm{d}_{\mathrm{LOC}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})$ and $\mathrm{d}_{\mathrm{CARD}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})$:

$$\mathrm{d}_{\mathrm{LOC}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}) = \left(\min_{\sigma} \sum_{\mathrm{i}=1}^{\mathfrak{m}} \left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})}{c} \right)^p \right)^{1/p} \tag{3.29}$$

$$\mathrm{d}_{\mathrm{CARD}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}) = (\widehat{\mathfrak{m}} - \mathfrak{m})^{1/p}. \tag{3.30}$$

[1] The Minkowski inequality is

$$\|x + y\|_p \leq \|x\|_p + \|y\|_p,$$

where $\|\cdot\|_p$ denotes the L_p norm and $p \geq 1$. Equality holds if, and only if, $x = \alpha y$ for some scalar, α.

Similarly to the note in [14], the functions $\mathrm{d}_{\mathrm{LOC}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}})$ and $\mathrm{d}_{\mathrm{CARD}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}})$ themselves are not strict metrics on the space of finite subsets and the decomposition of the metrics should not usually be necessary for gauging mapping performance. (3.29) and (3.30) however can be evaluated to provide extra information regarding the contributions of description and cardinality mapping errors.

The selection of the COLA metric's parameters c and p, and their physical interpretation, must also be addressed.

3.6.1 The effect of power variable p

In a similar manner to the OSPA metric [14], as p increases, the emphasis on localization errors diminishes. Therefore the COLA metric also becomes more unforgiving to cardinality errors for higher values of p. Based on the COLA metric form, given in (3.1), it is possible to show that for the same value of c, the COLA metric is also ordered with respect to p, however contrary to the OSPA metric:

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p_1)}(\mathcal{M},\widehat{\mathcal{M}}) \geq \mathrm{d}_{\mathrm{COLA}}^{(c,p_2)}(\mathcal{M},\widehat{\mathcal{M}}) \text{ for } 1 \leq p_1 < p_2 < \infty. \tag{3.31}$$

Proof: For $|\widehat{\mathcal{M}}| \geq |\mathcal{M}|$, the COLA metric is defined as:

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}}) = \left(\min_{\sigma} \sum_{\mathrm{i}=1}^{|\mathcal{M}|} \left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})}{c} \right)^{p} \right)^{1/p}. \tag{3.32}$$

If π is the permutation set which achieves the minimization of the COLA metric, then

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M},\widehat{\mathcal{M}}) = \left(\sum_{\mathrm{i}=1}^{|\mathcal{M}|} \left(\frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\pi(\mathrm{i})})}{c} \right)^{p} \right)^{1/p}. \tag{3.33}$$

Consider $1 \leq p_1 \leq p_2$ and the following substitutions:

$$x_{\mathrm{i}} = \frac{\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\pi(\mathrm{i})})}{c}, \quad s = \left(\sum_{\mathrm{i}=1}^{|\mathcal{M}|} x_{\mathrm{i}}^{p_1} \right)^{1/p_1}, \quad y_{\mathrm{i}} = \frac{x_{\mathrm{i}}}{s} \leq 1 \tag{3.34}$$

$$\therefore \quad y_{\mathrm{i}}^{p_2} \leq y_{\mathrm{i}}^{p_1} \tag{3.35}$$

$$\frac{\sum_{\mathrm{i}=1}^{|\mathcal{M}|} x_{\mathrm{i}}^{p_2}}{s^{p_2}} \leq \frac{\sum_{\mathrm{i}=1}^{|\mathcal{M}|} x_{\mathrm{i}}^{p_1}}{s^{p_1}} \tag{3.36}$$

$$\frac{\sum_{\mathrm{i}=1}^{|\mathcal{M}|} x_{\mathrm{i}}^{p_2}}{\left(\sum_{\mathrm{i}=1}^{|\mathcal{M}|} x_{\mathrm{i}}^{p_1} \right)^{p_2/p_1}} \leq 1 \tag{3.37}$$

$$\left(\sum_{\mathrm{i}=1}^{|\mathcal{M}|} x_{\mathrm{i}}^{p_2} \right)^{1/p_2} \leq \left(\sum_{\mathrm{i}=1}^{|\mathcal{M}|} x_{\mathrm{i}}^{p_1} \right)^{1/p_1} \tag{3.38}$$

and substituting for x_{i} from (3.34) into (3.38) yields

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p_2)}(\mathcal{M},\widehat{\mathcal{M}}) \;\leq\; \mathrm{d}_{\mathrm{COLA}}^{(c,p_1)}(\mathcal{M},\widehat{\mathcal{M}}).\square \tag{3.39}$$

Choosing $p = 1$ also makes the COLA metric behave in a relatively simple manner, in that it is then composed of the sum of cardinality and localization errors, which can be interpreted as fractional cardinality errors. Selecting $p = 2$ also makes the COLA metric more in-line with other metrics, which are often L_2-norms, and this value will be used throughout this article.

3.6.2 The effect of cut-off variable c

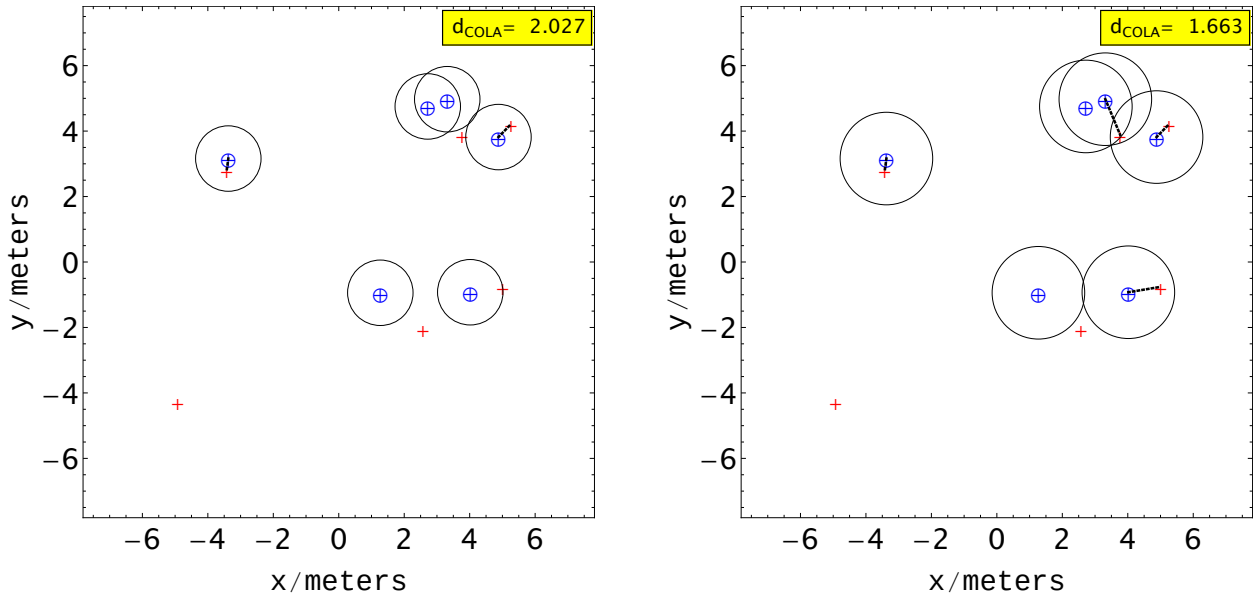
Analysis of the COLA metric form in (3.2) shows that as $c \to \infty$ the COLA metric becomes only sensitive to cardinality errors. Therefore, as in the OSPA metric, increasing c emphasizes the cardinality errors. However, contrary to the OSPA metric, from (3.2):

$$\mathrm{d}_{\mathrm{COLA}}^{(c_1,p)}(\mathcal{M},\widehat{\mathcal{M}}) \geq \mathrm{d}_{\mathrm{COLA}}^{(c_2,p)}(\mathcal{M},\widehat{\mathcal{M}}) \text{ for } 1 \leq c_1 < c_2 < \infty. \tag{3.40}$$

Figure 3.1 demonstrates the effect of c shown in Equation (3.40). Similarly to the OSPA metric [14], c should be chosen based on "What distance (e.g., how many meters) the designer wants to penalize a false or missing estimate", which in any application should significantly aid its practical choice. It should also be noted that the single object metric $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}},\widehat{\mathbf{m}}^{\sigma(\mathrm{i})})$ does not have to be a distance (e.g., Euclidean) metric - but could be the Mahalanobis (statistical) distance. In this case c could be chosen as a validation gate, corresponding to a probability interval within a chi-square test, within which an estimated feature is considered to correspond to a GT feature.

3.6.3 Setting a Threshold for the COLA Metric

A natural question which arises in the use of the COLA metric is: What threshold should be used to establish whether or not an estimated map is good? In contrast to the OSPA metric, the COLA metric does not saturate to a limiting value (c). In the case of $p = 1$, the COLA metric yields the total cardinality error between two set-based maps, which can be made up of fractional cardinality errors (assigned features which do not exactly coincide in terms of their attributes) and integer cardinality errors, due to a difference in the number of estimated and GT features. Therefore, a threshold to determine the difference between a good and bad map can be determined in terms of the number of effective cardinality errors, or outliers, one is willing to tolerate for a given application.



(a) $c = 1$ and $p = 1$.

(b) $c = 2$ and $p = 1$.

Figure 3.1: *A demonstration of the COLA metric via the assignment solution between GT features (blue circles), with their corresponding estimated map (red crosses). Figure 3.1a utilizes a cut-off parameter of $c = 1$, while Figure 3.1b employs a cut-off parameter of $c = 2$. Both of these figures represent c with black circles. Note that the number of correct assignments by the COLA metric increases with c. Finally, the COLA metric decreases while c increases, maintaining the same cardinality error. This reduction in their penalization is due to its localization component also decreasing while c increases.*

3.7 Comparison of the COLA and OSPA metrics

The physical interpretation of the OSPA and COLA metrics is now analyzed and compared for particular and general cases of the maps $\mathcal{M}$ and $\widehat{\mathcal{M}}$, to highlight the usefulness of each metric in assigning a meaningful score to map estimators. [14] demonstrated theoretical map scenarios which showed that the Hausdorff and OMAT metrics, although sensitive to spatial errors in the estimated map, are both insensitive to the cardinality error between $\mathcal{M}$ and $\widehat{\mathcal{M}}$. In those theoretical scenarios, it was shown that the OSPA metric provides intuitive results in terms of its sensitivity to both spatial and cardinality errors, and in [29] it was shown that the COLA metric provides intuitive results also. Therefore, in this section, in order to highlight differences between the OSPA and COLA metrics, and to study realistic mapping results, scenarios will be analyzed in which at least one of the maps is empty; one of the maps contains multiple features which are imbalanced with respect to the other map; and one of the maps contains outliers. An analysis of their penalizations of false alarms and missed detections will be provided as well as the theoretical conditions under which they can disagree in their comparisons of mapping performance.

To simplify the ensuing analyses here, $\mathrm{d}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})$ will be the Euclidean distance metric. To demonstrate their generality, actual SLAM performance evaluations in Section 5.2.1 will apply the COLA and OSPA metrics, in which $\mathrm{d}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})$ is the Mahalanobis distance,

allowing the incorporation of estimated feature covariance information, as often provided by SLAM algorithms.

3.7.1 Non-Empty Map Set vs Empty Map Set

Consider a GT map $\mathcal{M} = \emptyset$ and its estimate $\widehat{\mathcal{M}} = \{\widehat{\mathbf{m}}^1, \ldots, \widehat{\mathbf{m}}^{\widehat{\mathfrak{m}}}\}$ or vice versa. This corresponds to either a region containing no GT features or an empty estimated map, which could occur when no features are detected. The Euclidean, Hausdorff and OMAT metrics are all undefined in this case, since both sets must be non-empty. Meanwhile, the OSPA metric is given by

$$\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}) = \left(\frac{1}{\widehat{\mathfrak{m}}} c^p (\widehat{\mathfrak{m}} - 0)\right)^{1/p} = c, \tag{3.41}$$

and in this case, the COLA metric yields

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}) = (\widehat{\mathfrak{m}} - 0)^{1/p} = \widehat{\mathfrak{m}}^{1/p}. \tag{3.42}$$

Both metrics demonstrate a desirable asset since a metric should be defined when one of the sets is empty. However, the COLA metric can be considered to provide a more intuitive result. Irrespective of the difference in cardinality, the OSPA metric gives the same score (c) and is insensitive to this difference, whereas the COLA metric increases with $\widehat{\mathfrak{m}}$. For $p = 1$, the COLA metric increases linearly with $\widehat{\mathfrak{m}}$ which is the true value of the cardinality error in this case.

3.7.2 Multiple GT and Estimated Features

Consider the OSPA and COLA metrics' performances with GT map $\mathcal{M} = \{\mathbf{m}^1, \ldots, \mathbf{m}^{\mathfrak{m}}\}$ and estimated map

$$\begin{aligned} \widehat{\mathcal{M}} \;=\; & \{\widehat{\mathbf{m}}^{1,1}, \ldots, \widehat{\mathbf{m}}^{1,q}, \ldots, \widehat{\mathbf{m}}^{\mathfrak{m}-1,1}, \ldots, \widehat{\mathbf{m}}^{\mathfrak{m}-1,q}, \\ & \widehat{\mathbf{m}}^{\mathfrak{m},1}, \ldots, \widehat{\mathbf{m}}^{\mathfrak{m},q-s}\} \end{aligned} \tag{3.43}$$

(i.e. the GT landmarks $\mathbf{m}^1$ to $\mathbf{m}^{(\mathfrak{m}-1)}$ have in their neighborhoods q estimates whereas GT landmark $\mathbf{m}^{\mathfrak{m}}$ has in its neighborhood $q-s$ estimates, making the estimated map imbalanced [14]). The subset $\mathcal{M}' = \{\mathbf{m}^1, \ldots, \mathbf{m}^{\mathfrak{m}-1}\} \in \mathcal{M}$ is balanced and:

$$\begin{aligned} \mathrm{d}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\mathrm{i},l}) &= \mathrm{d} \leq c,\ 1 \leq \mathrm{i} \leq \mathfrak{m}-1,\ 1 \leq l \leq q \\ \mathrm{d}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j,l}) &> \mathrm{d} \quad \forall \quad \mathrm{i} \neq j, l \\ \mathrm{d}(\mathbf{m}^{\mathfrak{m}}, \widehat{\mathbf{m}}^{\mathfrak{m},l}) &= \mathrm{d} \leq c, \quad 1 \leq l \leq q-s \\ \mathrm{d}(\mathbf{m}^{\mathrm{i}}, \mathbf{m}^{j}) &> 2\mathrm{d} \quad \forall \quad \mathrm{i} \neq j. \end{aligned} \tag{3.44}$$

In this case

$$\mathrm{d}_{\mathrm{OSPA:imbal}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})^p = \left(\frac{1}{q\mathfrak{m} - s}(\mathfrak{m}\mathrm{d}^p + c^p(\mathfrak{m}(q-1) - s))\right)^{\frac{1}{p}}. \tag{3.45}$$

Note that if $s = 0$ in (3.43), the map is balanced such that $\mathrm{d}_{\mathrm{OSPA:bal}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})^p$ is given by (3.45) with $s = 0$.

Comparing the OSPA metric values for the cases of balanced and imbalanced maps gives

$$\begin{aligned} \mathrm{d}^{(c,p)}_{\mathrm{OSPA:bal}}(\mathcal{M},\widehat{\mathcal{M}})^p &\geq \mathrm{d}^{(c,p)}_{\mathrm{OSPA:imbal}}(\mathcal{M},\widehat{\mathcal{M}})^p, \quad s \geq 0, \\ \mathrm{d}^{(c,p)}_{\mathrm{OSPA:imbal}}(\mathcal{M},\widehat{\mathcal{M}})^p &\geq \mathrm{d}^{(c,p)}_{\mathrm{OSPA:bal}}(\mathcal{M},\widehat{\mathcal{M}})^p, \quad s < 0 \end{aligned} \tag{3.46}$$

meaning that, unlike the Hausdorff and OMAT metrics, the OSPA metric penalizes false alarms in an intuitive manner. This can be seen in Fig. 3.2 since the estimates in Fig. 3.2b and Fig. 3.2d are better than those of Fig. 3.2a and Fig. 3.2c because they have less cardinality errors but the same spatial errors. These facts are not reflected correctly in the

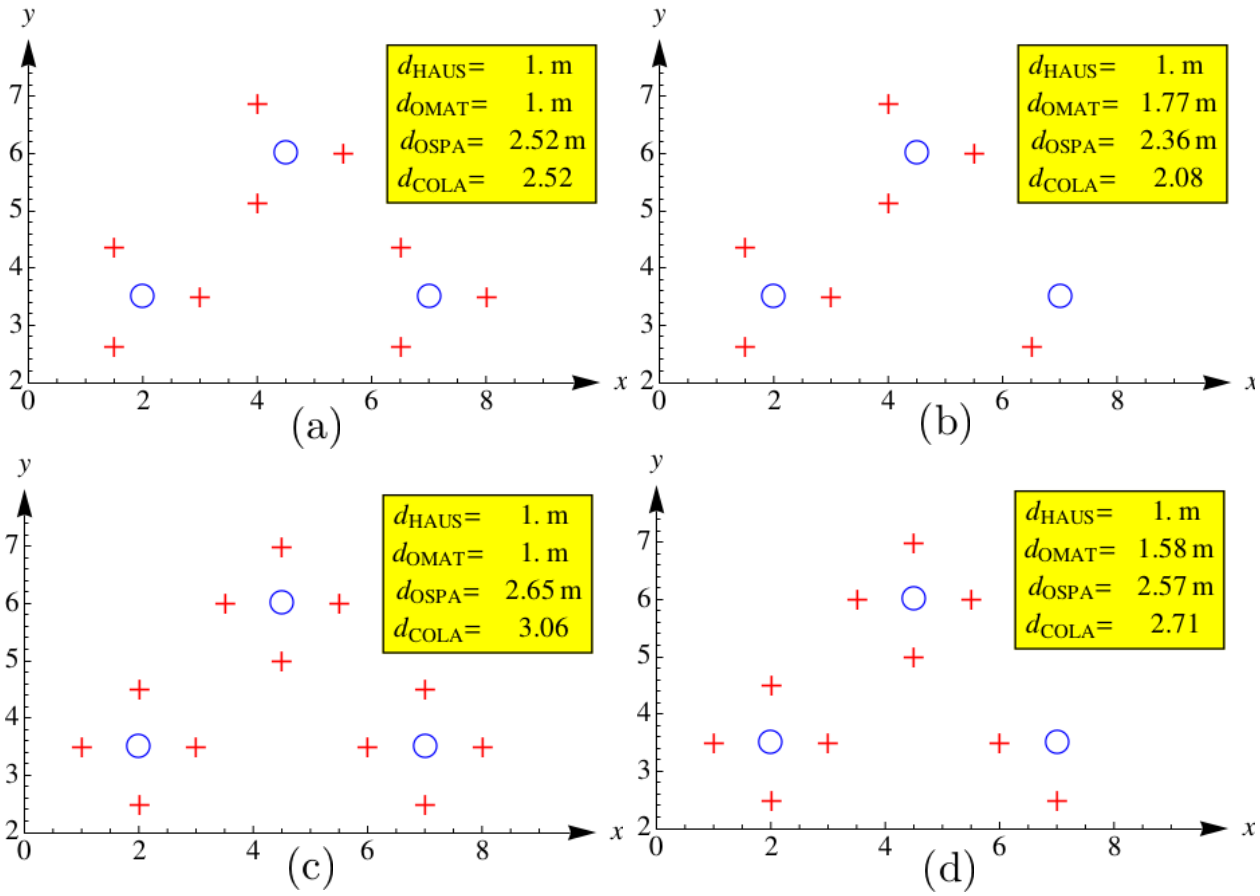


Figure 3.2: *Metric performances when cardinality errors exist. Figs. (a) and (c) are balanced maps while Figs. (b) and (d) are imbalanced. The distance between the center of each GT landmark (blue circles) and its neighboring estimates (red crosses) is* $1m$. *For calculating* d_{OSPA} *and* d_{COLA}, $c = 3.00m$ *and* $p = 2$.

Hausdorff and OMAT metric values.

The general imbalanced scenario described in (3.43) yields a COLA metric value

$$\mathrm{d}^{(c,p)}_{\mathrm{COLA:imbal}}(\mathcal{M},\widehat{\mathcal{M}})^p = \left(\mathfrak{m}\left(\frac{\mathrm{d}}{c}\right)^p + (\mathfrak{m}(q-1)-s)\right)^{1/p}, \tag{3.47}$$

Comparing the COLA metric values for the cases of balanced ($s = 0$) and imbalanced maps, it can also be seen that $\mathrm{d}^{(c,p)}_{\mathrm{COLA:bal}}(\mathcal{M},\widehat{\mathcal{M}})^p \geq \mathrm{d}^{(c,p)}_{\mathrm{COLA:imbal}}(\mathcal{M},\widehat{\mathcal{M}})^p$ for $s \geq 0$ and vice versa for

$s < 0$, which again complies with intuition. It should also be noted that, true to its nature of having units of cardinality error, for $p = 1$, $\mathrm{d}^{(c,p)}_{\mathrm{COLA:bal}}(\mathcal{M}, \widehat{\mathcal{M}})^p - \mathrm{d}^{(c,p)}_{\mathrm{COLA:imbal}}(\mathcal{M}, \widehat{\mathcal{M}})^p = s$, yielding the exact cardinality error between the maps.

3.7.3 Outliers

Consider an estimated map with one outlier - i.e. $\mathcal{M} = \{\mathbf{m}^1, \ldots, \mathbf{m}^{\mathfrak{m}}\}$ and $\widehat{\mathcal{M}} = \{\widehat{\mathbf{m}}^1, \ldots, \widehat{\mathbf{m}}^{\mathfrak{m}+1}\}$, and assume that every single GT landmark has a perfect estimate - i.e. $\mathrm{d}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\mathrm{i}}) = 0$ for $1 \leq \mathrm{i} \leq \mathfrak{m}$. From (2.9), the OSPA metric is

$$\mathrm{d}^{(c,p)}_{\mathrm{OSPA}}(\mathcal{M}, \widehat{\mathcal{M}})^p = c \left(\frac{1}{\mathfrak{m}+1} \right)^{1/p}. \tag{3.48}$$

If $\mathfrak{m} \to \infty$, the OSPA metric $\mathrm{d}^{(c,p)}_{\mathrm{OSPA}}(\mathcal{M}, \widehat{\mathcal{M}})^p \to 0$ distance units, giving an intuitive result.

In the case of the COLA metric,

$$\mathrm{d}^{(c,p)}_{\mathrm{COLA}}(\mathcal{M}, \widehat{\mathcal{M}})^p = 1, \tag{3.49}$$

which, since this metric yields cardinality, as opposed to average distance units, can also be considered to be intuitive. In this case there is a single outlier, and the COLA metric correctly reports it. It should be noted here that the COLA metric is somewhat unforgiving to cardinality errors. For example, if a mapping algorithm estimates a large number of perfectly located estimates, with just one false alarm, the COLA metric always penalizes the algorithm, even though as $\mathfrak{m} \to \infty$ the algorithm can be argued to be approaching perfection. In this sense the OSPA metric behaves more intuitively. It should be noted however, that this does not cause any problems in the use of the COLA metric, when estimated maps are compared to the *same GT, or other algorithm's map*, as carried out at the end of mapping/SLAM tasks, since the same strict penalization of cardinality errors is applied to all estimates. However, when assessing mapping performance *during* SLAM execution, as more of the GT map passes through the field(s) of view of the sensor(s), the COLA metric would never lower its value due to previous outliers, even if the map estimator approaches perfection later on. This would not allow a useful on-line evaluation of mapping performance. In this case, due to the per feature averaged distance nature of the OSPA metric, it provides the most intuitive evaluation of the time varying estimated map.

3.7.4 Penalization of False Alarms and Missed Detections

A theoretical analysis will now demonstrate that the OSPA and COLA metrics penalize over and underestimated cardinality errors (false alarms and missed detections) in a significantly different manner. Consider a GT map $\mathcal{M} = \{\mathbf{m}^1, \ldots, \mathbf{m}^{\mathfrak{m}}\}$, and two map estimates, $\widehat{\mathcal{M}}_1 = \{\widehat{\mathbf{m}}^1, \ldots, \widehat{\mathbf{m}}^{\mathfrak{m}-\mathfrak{a}}\}$ and $\widehat{\mathcal{M}}_2 = \{\widehat{\mathbf{n}}^1, \ldots, \widehat{\mathbf{n}}^{\mathfrak{m}+\mathfrak{a}}\}$, where $\mathfrak{m} \geq \mathfrak{a}$. Estimate $\widehat{\mathcal{M}}_1$ underestimates the map size by $\mathfrak{a}$ and estimate $\widehat{\mathcal{M}}_2$ over estimates it by the same amount, $\mathfrak{a}$. Suppose also that $\widehat{\mathcal{M}}_1$ and a sub-set of $\widehat{\mathcal{M}}_2$, have no localization errors - i.e. $\widehat{\mathbf{m}}^{\sigma(\mathrm{i})} = \mathbf{m}^{\mathrm{i}}$ for $1 \leq \mathrm{i} \leq \mathfrak{m} - \mathfrak{a}$ and $\widehat{\mathbf{n}}^{\pi(\mathrm{i})} = \mathbf{m}^{\mathrm{i}}$ for $1 \leq \mathrm{i} \leq \mathfrak{m}$, where $\sigma(\mathrm{i})$ and $\pi(\mathrm{i})$ are the assignments used in comparing maps

$\widehat{\mathcal{M}}_1$ with $\mathcal{M}$ and $\widehat{\mathcal{M}}_2$ with $\mathcal{M}$ respectively. The OSPA errors for both cases are

$$\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_1) = c\left(\frac{\mathfrak{a}}{\mathfrak{m}}\right)^{1/p} \tag{3.50}$$

$$\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_2) = c\left(\frac{\mathfrak{a}}{\mathfrak{m}+\mathfrak{a}}\right)^{1/p}, \tag{3.51}$$

implying that, independent of parameters c and p, the OSPA metric penalizes the estimate with the missed detections *more* than the estimate with the same number of false alarms.

On the other hand, the COLA metric for both cases yields

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_1) = \mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_2) = \mathfrak{a}^{1/p}, \tag{3.52}$$

implying that the COLA metric penalizes both estimates by the same amount. Arguably, the COLA metric can be considered to provide a more intuitive error estimate of both mapping errors, since neither the OSPA nor the COLA metrics are designed to judge them in an unequal manner. The consequences of missed detections and false alarms on robot navigation are application specific, and judging one type of error to be worse than the other is beyond the scope of these metrics alone.

3.8 Can the COLA and OSPA Metrics Differ?

This section demonstrates the conditions under which the COLA and OSPA metrics disagree. Consider a GT map $\mathcal{M}$ with cardinality $\mathfrak{m}$ and its estimates $\widehat{\mathcal{M}}_1$ and $\widehat{\mathcal{M}}_2$, with cardinalities $\widehat{\mathfrak{m}}_1$ and $\widehat{\mathfrak{m}}_2$ respectively, where:

$$\widehat{\mathfrak{m}}_1 \leq \mathfrak{m} < \widehat{\mathfrak{m}}_2 \tag{3.53}$$

$$\text{and} \quad \mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\widehat{\mathcal{M}}_1, \mathcal{M}) \geq \mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_2) \tag{3.54}$$

The COLA metric values for both cases are then:

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\widehat{\mathcal{M}}_1, \mathcal{M}) = \frac{\mathfrak{m}^{1/p}}{c}\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\widehat{\mathcal{M}}_1, \mathcal{M}) \tag{3.55}$$

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_2) = \frac{\widehat{\mathfrak{m}}_2^{1/p}}{c}\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_2). \tag{3.56}$$

For the metrics to disagree, it is necessary that

$$\mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\widehat{\mathcal{M}}_1, \mathcal{M}) < \mathrm{d}_{\mathrm{COLA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_2), \tag{3.57}$$

which requires that

$$\frac{\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\widehat{\mathcal{M}}_1, \mathcal{M})}{\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_2)} < \left(\frac{\widehat{\mathfrak{m}}_2}{\mathfrak{m}}\right)^{1/p} \tag{3.58}$$

A similar analysis shows that if

$$\mathfrak{m} \leq \widehat{\mathfrak{m}}_1 < \widehat{\mathfrak{m}}_2 \tag{3.59}$$

$$\text{and} \quad \mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_1) \geq \mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_2) \tag{3.60}$$

the metrics can also disagree if

$$\frac{\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_1)}{\mathrm{d}_{\mathrm{OSPA}}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}}_2)} < \left(\frac{\widehat{\mathfrak{m}}_2}{\widehat{\mathfrak{m}}_1}\right)^{1/p} \tag{3.61}$$

Hence disagreement between both metrics is only possible if (3.53), (3.54) and (3.58) or (3.59), (3.60) and (3.61) are simultaneously satisfied, as will be demonstrated in Section 5.2.1.

Note that if $\widehat{\mathfrak{m}}_1,\ \widehat{\mathfrak{m}}_2 \leq \mathfrak{m}$, both metrics always agree.

3.9 Computational Complexity of the COLA Metric

Since metrics are often determined off-line, after experiments have terminated, their computational complexity is usually not of major concern. However, if the metric is to continuously gauge mapping/SLAM performance during algorithm execution, its complexity is important. For the COLA and OSPA metrics, their computational complexities are equivalent and dependent on the assignment method used to determine $\sigma(\mathrm{i})$. The Jonker and Volgenant algorithm is an efficient assignment method, which is, in general, faster than the Hungarian method, and has worst case cubic complexity in the dimension of the distance matrix $\mathbf{D}_{\mathrm{i},j} = \mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j})^p$. In particular if $\mathfrak{m} = |\mathcal{M}| = |\widehat{\mathcal{M}}| = \widehat{\mathfrak{m}}$, implies that $\mathbf{D}_{\mathrm{i},j}$ is an squared matrix with dimensions $\mathfrak{m} \times \mathfrak{m}$. This yields to a computational complexity of $\mathcal{O}(\mathfrak{m}^3)$ [14, 58, 81].

Other computational improvements of the Hungarian method are available, which are reported to reduce the execution time in linear assignment problems by up to 90% [82]. Recently, further optimizations of the Jonker and Volgenant algorithm have also been reported [83].

Since, in general $\mathfrak{m} = |\mathcal{M}| \neq |\widehat{\mathcal{M}}| = \widehat{\mathfrak{m}}$, assuming $\mathfrak{m} \leq \widehat{\mathfrak{m}}$, in the implementation of the COLA metric, it is required to add dummy points to $\mathcal{M}$, generating the new set $\bar{\mathcal{M}} = \{\mathbf{u}^1, \ldots, \mathbf{u}^{(\widehat{\mathfrak{m}}-\mathfrak{m})}\} \cup \mathcal{M}$ such that $\mathrm{d}^{c}(\mathbf{u}^k, \widehat{\mathbf{m}}^{j})^p = c^p$, for all j, k. Finally the construction of the $\bar{\mathbf{D}}_{\mathrm{i},j} = \mathrm{d}^{(c)}(\bar{\mathbf{m}}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j})^p$ has dimensions of $\widehat{\mathfrak{m}} \times \widehat{\mathfrak{m}}$, yielding a computational complexity of $\mathcal{O}(\widehat{\mathfrak{m}}^3) = \mathcal{O}(\max(\mathfrak{m}, \widehat{\mathfrak{m}})^3)$. In conclusion, the computational complexity of OSPA and COLA increases, when $\widehat{\mathcal{M}}$ contains false alarms.

Chapter 4

PSO-COLA: A Robust Solution for Correspondence-Free, Point Set Registration

4.1 Introduction

In this chapter we introduce a new technique called PSO-COLA for 3D PCD registration. This method is based on the context of multi-object metrics, which quantify the error between point cloud datasets in the presence of cardinality and spatial errors. This concept allows for the robust registration of scans in terms of translation and rotation between them, also taking into account possible detection errors. Recent methods address this problem in a variety of ways, but rarely take into account detection errors, i.e., false alarms and missed detections. The resulting PSO-COLA registration algorithm is shown to outperform state of the art local and global point cloud registration algorithms in various scenarios in the presence of data outliers and spatial uncertainty.

4.2 Registration with Multi-Object Metrics

Based on the concept of multi-object metrics, we show an alternative solution to rigid global point cloud registration. To define the point set registration problem, it is necessary to consider two sets: The *reference* set $\mathcal{M} = \{\mathbf{m}^1, \ldots, \mathbf{m}^{\mathfrak{m}}\}$ and its *model* set $\widehat{\mathcal{M}} = \{\widehat{\mathbf{m}}^1, \ldots, \widehat{\mathbf{m}}^{\widehat{\mathfrak{m}}}\}$ recorded in different frames of reference defined by $\mathbf{S}$ and $\widehat{\mathbf{S}}$ respectively. Note that in the presence of detection errors, in general, $\mathfrak{m} \neq \widehat{\mathfrak{m}}$. Our goal is to find a solution of:

$$\text{argmin}_{\mathbf{R},\mathbf{t}}\, \mathrm{d}^{(c,p)}(\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t}), \widehat{\mathcal{M}}), \tag{4.1}$$

where $\mathrm{d}^{(c,p)}(\mathcal{M}, \widehat{\mathcal{M}})$ is given by the OSPA or COLA metric, $\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t})$ is the transformed set of $\mathcal{M}$ from $\mathbf{S}$ to $\widehat{\mathbf{S}}$ via rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, c is the COLA metric's cut-off distance and p the power [25]. Without loss of generality, this thesis considers 3D

space in which

$$\begin{aligned}
\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t}) &= \{\mathbf{R}\mathbf{m}^1+\mathbf{t},\ldots,\mathbf{R}\mathbf{m}^{\mathfrak{m}}+\mathbf{t}\} && (4.2)\\
\mathbf{R}(\alpha,\beta,\gamma) &= \mathbf{R}_z(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_x(\alpha) && (4.3)\\
\mathbf{R}_x(\alpha) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} && (4.4)\\
\mathbf{R}_y(\beta) &= \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} && (4.5)\\
\mathbf{R}_z(\gamma) &= \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} && (4.6)\\
\mathbf{t} &= \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} && (4.7)\\
\alpha &\in [-\pi,\pi] && (4.8)\\
\beta &\in [-\pi/2,\pi/2] && (4.9)\\
\gamma &\in [-\pi,\pi], && (4.10)
\end{aligned}$$

where α, β and γ are the roll, pitch and yaw angles respectively and t_x, t_y and t_z are the x, y and z components of the translation vector.
Note that in general, minimizing (4.1) with respect to $\mathbf{R}$ and $\mathbf{t}$ is problematic, because the function

$$f^{(c,p)}(\mathbf{R}(\alpha,\beta,\gamma),\mathbf{t}) = \mathrm{d}^{(c,p)}(\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t}),\widehat{\mathcal{M}}), \tag{4.11}$$

belongs to a family of non-convex functions with several local maxima and minima, which depend on both c and p. Therefore, a method which calculates the best solution in a global sense is required.

Note that if either the OSPA or COLA metrics are used within the point cloud registration algorithm of equation (4.11), the same estimated transform results. To prove this, assume that the rotation matrix $\mathbf{R}^\star$ and the translation vector $\mathbf{t}^\star$ satisfy

$$\min_{\mathbf{R},\mathbf{t}} \mathrm{d}^{(c,p)}_{\text{OSPA}}(\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t}),\widehat{\mathcal{M}}) = \mathrm{d}^{(c,p)}_{\text{OSPA}}(\mathcal{M}_{\text{trans}}(\mathbf{R}^\star,\mathbf{t}^\star),\widehat{\mathcal{M}}) \tag{4.12}$$

Using the relationship between the COLA and OSPA metrics from equation (3.55)

$$\begin{aligned}
\min_{\mathbf{R},\mathbf{t}} \mathrm{d}^{(c,p)}_{\text{COLA}}(\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t}),\widehat{\mathcal{M}}) &= \min_{\mathbf{R},\mathbf{t}} \left(\frac{\max(\mathfrak{m},\widehat{\mathfrak{m}})^{1/p}}{c}\right) \mathrm{d}^{(c,p)}_{\text{OSPA}}(\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t}),\widehat{\mathcal{M}}) \\
&= \left(\frac{\max(\mathfrak{m},\widehat{\mathfrak{m}})^{1/p}}{c}\right) \min_{\mathbf{R},\mathbf{t}} \mathrm{d}^{(c,p)}_{\text{OSPA}}(\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t}),\widehat{\mathcal{M}}). && (4.13)
\end{aligned}$$

Substituting (4.12) into equation (4.13) yields

$$\begin{aligned}
\min_{\mathbf{R},\mathbf{t}} \mathrm{d}^{(c,p)}_{\text{COLA}}(\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t}),\widehat{\mathcal{M}}) &= \left(\frac{\max(\mathfrak{m},\widehat{\mathfrak{m}})^{1/p}}{c}\right) \min_{\mathbf{R},\mathbf{t}} \mathrm{d}^{(c,p)}_{\text{OSPA}}(\mathcal{M}_{\text{trans}}(\mathbf{R}^\star,\mathbf{t}^\star),\widehat{\mathcal{M}}) \\
&= \mathrm{d}^{(c,p)}_{\text{COLA}}(\mathcal{M}_{\text{trans}}(\mathbf{R}^\star,\mathbf{t}^\star),\widehat{\mathcal{M}}). && (4.14)
\end{aligned}$$

Therefore, the rotation matrix $\mathbf{R}^\star$ and the translation vector $\mathbf{t}^\star$ optimize both the OSPA and COLA metrics simultaneously. Therefore without loss of generality, this thesis applies the COLA metric.

In the following sections, we will explain the method used in this work, in order to minimize equation (4.1). For this, section 4.3 considers a method to estimate a coarse alignment between $\mathcal{M}$ and $\widehat{\mathcal{M}}$ as an initialization. Subsequently, Section 4.4 defines a refinement solution based on this initialization, using Particle Swarm Optimization (PSO). The complete algorithm is referred to as the PSO-COLA algorithm.

4.3 Initialization: Coarse pose and correspondence estimation by distance pattern matching.

It is assumed that the correspondence between points within the model and the reference sets is unknown. Furthermore if the transformation that best aligns these sets is rigid, as assumed in this article, then the distance between corresponding points is equivalent if the data is not corrupted with spatial noise. The concept used here is to define correspondence hypotheses between point pairs and then use multi-object metrics to estimate which correspondence is most likely. Mathematically, suppose that $\widehat{\mathbf{m}}^{\mathrm{i}}$ and $\widehat{\mathbf{m}}^{j}$ correspond to $\mathbf{m}^k$ and $\mathbf{m}^l$ respectively, i.e.

$$\widehat{\mathbf{m}}^{\mathrm{i}} = \mathbf{R}\mathbf{m}^k + \mathbf{t}, \quad \widehat{\mathbf{m}}^{j} = \mathbf{R}\mathbf{m}^l + \mathbf{t}. \tag{4.15}$$

Due to the assumption of rigid transformation, the distance between each corresponding pair of points, within each set, satisfies

$$\|\widehat{\mathbf{m}}^{\mathrm{i}} - \widehat{\mathbf{m}}^{j}\| = \|\mathbf{R}\mathbf{m}^k - \mathbf{R}\mathbf{m}^l\| = \|\mathbf{R}(\mathbf{m}^k - \mathbf{m}^l)\| = \|\mathbf{m}^k - \mathbf{m}^l\|, \tag{4.16}$$

and the distances between $\widehat{\mathbf{m}}^{\mathrm{i}}$ and $\widehat{\mathbf{m}}^{j}$ and $\mathbf{m}^k$ and $\mathbf{m}^l$ are equivalent. Equation (4.16) suggests that we can define the distance:

$$\mathrm{d}_o(\widehat{\mathbf{m}}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j}, \mathbf{m}^k, \mathbf{m}^l) = |\|\widehat{\mathbf{m}}^{\mathrm{i}} - \widehat{\mathbf{m}}^{j}\| - \|\mathbf{m}^k - \mathbf{m}^l\|| \tag{4.17}$$

as the error associated with this correspondence hypothesis. Note that $\mathrm{d}_o(\widehat{\mathbf{m}}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j}, \mathbf{m}^k, \mathbf{m}^l) = 0$ iff $\widehat{\mathbf{m}}^{\mathrm{i}}$ and $\widehat{\mathbf{m}}^{j}$ correctly correspond to $\mathbf{m}^k$ and $\mathbf{m}^l$ respectively.

In this paper we generalize this concept, for computing correspondences based on multi-object metric concepts similar to the COLA metric. The key idea is to find the optimal correspondence, in the sense of the COLA metric, between the sets

$$\mathcal{D}^{\mathcal{M}}_{\mathbf{m}^{\mathrm{i}}} = \{\|\mathbf{m}^1 - \mathbf{m}^{\mathrm{i}}\| \dots \|\mathbf{m}^{\mathfrak{m}} - \mathbf{m}^{\mathrm{i}}\|\}, \quad \mathcal{D}^{\widehat{\mathcal{M}}}_{\widehat{\mathbf{m}}^{l}} = \{\|\widehat{\mathbf{m}}^1 - \widehat{\mathbf{m}}^{l}\| \dots \|\widehat{\mathbf{m}}^{\widehat{\mathfrak{m}}} - \widehat{\mathbf{m}}^{l}\|\}, \tag{4.18}$$

where $\mathcal{D}^{\mathcal{M}}_{\mathbf{m}^{\mathrm{i}}}$ is the set of distances within $\mathcal{M}$, with respect to a chosen point $\mathbf{m}^{\mathrm{i}}$. Similarly, $\mathcal{D}^{\widehat{\mathcal{M}}}_{\widehat{\mathbf{m}}^{l}}$ is the set of distances within $\widehat{\mathcal{M}}$, with respect to a chosen point $\widehat{\mathbf{m}}^{l}$ (see Fig. 4.1). Assuming that $\widehat{\mathcal{M}}$ is corrupted with spatial and detection uncertainty, we define a multi-object COLA metric $\mathrm{d}^{(c,p)}_{\mathrm{Match}}(\mathcal{D}^{\mathcal{M}}_{\mathbf{m}^{\mathrm{i}}}, \mathcal{D}^{\widehat{\mathcal{M}}}_{\widehat{\mathbf{m}}^{l}})$ such that if $\widehat{\mathfrak{m}} \geq \mathfrak{m}$.

$$\mathrm{d}^{(c,p)}_{\mathrm{Match}}(\mathcal{D}^{\mathcal{M}}_{\mathbf{m}^{\mathrm{i}}}, \mathcal{D}^{\widehat{\mathcal{M}}}_{\widehat{\mathbf{m}}^{l}}) = \left(\min_{\sigma} \sum_{s=1}^{\mathfrak{m}} \left(\frac{\mathrm{d}^{(c)}(\|\mathbf{m}^s - \mathbf{m}^{\mathrm{i}}\|, \|\widehat{\mathbf{m}}^{\sigma(s)} - \widehat{\mathbf{m}}^{l}\|)}{c} \right)^p + \widehat{\mathfrak{m}} - \mathfrak{m} \right)^{1/p}, \tag{4.19}$$

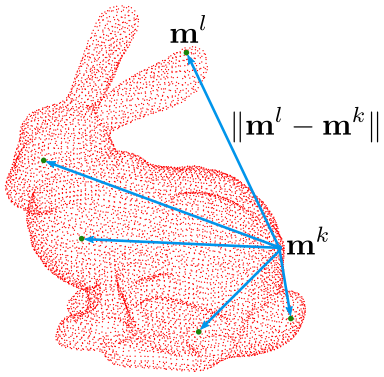


Figure 4.1: Highlighting the construction of the set of distances $\mathcal{D}^{\mathcal{M}}_{\mathbf{m}^{\mathrm{i}}}$ with a respect to a chosen point $\mathbf{m}^k$ within the Stanford Bunny dataset.

where σ is a set of permutations of $\{1, ..., \mathfrak{m}\}$, associated with correspondences between elements of $\mathcal{M}$ and $\widehat{\mathcal{M}}$, which minimizes $\sum_{s=1}^{\mathfrak{m}} \mathrm{d}^{(c)}(\|\mathbf{m}^s - \mathbf{m}^{\mathrm{i}}\|, \|\widehat{\mathbf{m}}^{\sigma(s)} - \widehat{\mathbf{m}}^l\|)^p$. $\mathrm{d}^{(c)}$ is defined as the inner metric:

$$\mathrm{d}^{(c)}(\|\mathbf{m}^s - \mathbf{m}^{\mathrm{i}}\|, \|\widehat{\mathbf{m}}^{\sigma(s)} - \widehat{\mathbf{m}}^l\|) = \min(c, |\|\mathbf{m}^s - \mathbf{m}^{\mathrm{i}}\| - \|\widehat{\mathbf{m}}^{\sigma(s)} - \widehat{\mathbf{m}}^l\||), \tag{4.20}$$

and $c > 0$ is the cut-off parameter. Note that for $\widehat{\mathfrak{m}} \leq \mathfrak{m}$, the metric is defined as $\mathrm{d}^{(c,p)}_{\mathrm{Match}}(\mathcal{D}^{\widehat{\mathcal{M}}}_{\widehat{\mathbf{m}}^l}, \mathcal{D}^{\mathcal{M}}_{\mathbf{m}^{\mathrm{i}}})$. The following points should also be noted:

- This generalizes the concept of comparing distances between sets based on RANSAC, since it allows the computation of correspondences even in the presence of cardinality differences between $\mathcal{M}$ and $\widehat{\mathcal{M}}$.
- Based on the Hungarian method, the metric $\mathrm{d}^{(c,p)}_{\mathrm{Match}}(\mathcal{D}^{\mathcal{M}}_{\mathbf{m}^{\mathrm{i}}}, \mathcal{D}^{\widehat{\mathcal{M}}}_{\widehat{\mathbf{m}}^l})$ is globally minimized.

The computation of the optimal correspondence to compute an initial coarse set of transformations $\mathbf{R}$ and $\mathbf{t}$ is as follows:

1. Compute $\mathrm{d}^{(c,p)}_{\mathrm{Match}}(\mathcal{D}^{\mathcal{M}}_{\mathbf{m}^{\mathrm{i}}}, \mathcal{D}^{\widehat{\mathcal{M}}}_{\widehat{\mathbf{m}}^l})$ for each $\mathbf{m}^{\mathrm{i}} \in \mathcal{M}$ and $\widehat{\mathbf{m}}^l \in \widehat{\mathcal{M}}$. For this purpose, set $n = 1$ and compute the following indices

$$\mathrm{i}(n) = \left\lceil \frac{n}{\widehat{\mathfrak{m}}} \right\rceil, \quad l(n) = n - \widehat{\mathfrak{m}} \left\lfloor \frac{n-1}{\widehat{\mathfrak{m}}} \right\rfloor, \tag{4.21}$$

   where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ are the ceiling and floor functions respectively. Note that the index $n = 1, \ldots, \mathfrak{m}\widehat{\mathfrak{m}}$. Then, compute the corresponding $\mathrm{d}_{\mathrm{i}(n),l(n)}$ value as follows

$$\mathrm{d}_{\mathrm{i}(n),l(n)} = \mathrm{d}^{(c,p)}_{\mathrm{Match}}(\mathcal{D}^{\mathcal{M}}_{\mathbf{m}^{\mathrm{i}(n)}}, \mathcal{D}^{\widehat{\mathcal{M}}}_{\widehat{\mathbf{m}}^{l(n)}}). \tag{4.22}$$

2. Use only the correspondences σ determined in equation (4.19) for which

$$\mathrm{d}^{(c)}(\|\mathbf{m}^s - \mathbf{m}^{\mathrm{i}(n)}\|, \|\widehat{\mathbf{m}}^{\sigma(s)} - \widehat{\mathbf{m}}^{l(n)}\|) < c. \tag{4.23}$$

   Refer to this subset of permutations as $\gamma(s)$. Then, compute the transformations $\mathbf{R}^\star$ and $\mathbf{t}^\star$ via Arun's method [60] as presented in Algorithm 1.

Algorithm 1: The Arun's Method

Input: Reference set $\mathcal{M} = \{\mathbf{m}^1, \ldots, \mathbf{m}^{\mathfrak{m}}\}$, Model set $\widehat{\mathcal{M}} = \{\widehat{\mathbf{m}}^1, \ldots, \widehat{\mathbf{m}}^{\mathfrak{m}}\}$.
Output: Rotation matrix $\mathbf{R}^\star$, translation vector $\mathbf{t}^\star$.
// Compute the centroids $\mathbf{m}_c$ and $\widehat{\mathbf{m}}_c$ of $\mathcal{M}$ and $\widehat{\mathcal{M}}$ respectively,
Initialize $\mathbf{m}_c = \frac{1}{\mathfrak{m}} \sum_{s=1}^{\mathfrak{m}} \mathbf{m}^{\gamma(s)}$
Initialize $\widehat{\mathbf{m}}_c = \frac{1}{\mathfrak{m}} \sum_{s=1}^{\mathfrak{m}} \widehat{\mathbf{m}}^{\sigma(\gamma(s))}$.
// Compute the translated sets $\mathcal{M}'$ and $\widehat{\mathcal{M}}'$ with respect to their respective centroids $\mathbf{m}_c$ and $\widehat{\mathbf{m}}_c$
Initialize $\mathcal{M}' = \{\mathbf{m}^1 - \mathbf{m}_c, \ldots, \mathbf{m}^{\mathfrak{m}} - \mathbf{m}_c\}$
Initialize $\widehat{\mathcal{M}}' = \{\widehat{\mathbf{m}}^1 - \widehat{\mathbf{m}}_c, \ldots, \widehat{\mathbf{m}}^{\mathfrak{m}} - \widehat{\mathbf{m}}_c\}$.
// Compute the covariance matrix $\mathbf{H}$.
Initialize $\mathbf{H} = \sum_{s=1}^{\mathfrak{m}} \mathbf{m}'^s (\widehat{\mathbf{m}}'^s)^\top$.
// Compute the orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$, via SVD
$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ // where $\mathbf{\Sigma}$ is a diagonal matrix, with singular value components.
// Compute the rotation matrix $\mathbf{R}^\star$ and translation vector $\mathbf{t}^\star$ using the orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$, as well as the centroids $\mathbf{m}_c$ and $\widehat{\mathbf{m}}_c$ via $\mathbf{R}^\star = \mathbf{V}\mathbf{U}^\top$ and $\mathbf{t}^\star = \widehat{\mathbf{m}}_c - \mathbf{R}\mathbf{m}_c$.
Return $\mathbf{R}^\star$, $\mathbf{t}^\star$. ;

This is a closed-form solution to the least-squares approach to register two point sets in 3D space. It uses the Singular Value Decomposition (SVD) method, applied to the centroids and covariance matrices from each point set.

3. Using the equation (4.11), compute the corresponding COLA metric between $\mathcal{M}$ and $\widehat{\mathcal{M}}$, using the transformation $\mathbf{R}^\star$ and $\mathbf{t}^\star$. In particular, if $n = 1$, $\mathbf{R}_o = \mathbf{R}^\star$ and $\mathbf{t}_o = \mathbf{t}^\star$.
4. When the minimum according to equation (4.11) is found, $\mathbf{R}_o = \mathbf{R}^\star$ and $\mathbf{t}_o = \mathbf{t}^\star$.
5. Return to step 1 and iterate using $n := n + 1$ in equations (4.21) until $n = \mathfrak{m}\widehat{\mathfrak{m}}$.
6. Finally, the transformation for registration initialization is $\mathbf{R}_o$, $\mathbf{t}_o$.

4.4 The PSO-COLA Algorithm

The PSO is currently applied in several disciplines. This is due to its robustness in finding optimal global solutions in few steps with a set of particles exploring an objective function. The PSO algorithm uses interactive particles $\mathbf{p}^{\mathrm{i}}(k)$ at iteration k, with positions $\mathbf{x}^{\mathrm{i}}(k)$ and velocity $\mathbf{v}^{\mathrm{i}}(k)$. Further, all members of the particle population communicate their cost function values $g(\mathbf{x}^{\mathrm{i}}(k))$ to the other particles $\mathbf{p}^j(k)$, $j \neq \mathrm{i}$. $g(\mathbf{x}^{\mathrm{i}}(k))$ is the cost function evaluated at position $\mathbf{x}^{\mathrm{i}}(k)$, where in this case the function $g()$ is the objective function defined by

$$g(\mathbf{x}^{\mathrm{i}}(k)) = f^{(c,p)}(\mathbf{R}(k), \mathbf{t}(k)). \tag{4.24}$$

Finally, the method uses an iterative motion model for the particles given by the following equations:

$$\begin{aligned} \mathbf{v}^{\mathrm{i}}(k+1) &= \chi \mathbf{v}^{\mathrm{i}}(k) + c_1 r_1 (\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k) - \mathbf{x}^{\mathrm{i}}(k)) \\ &+ c_2 r_2 (\mathbf{x}_{\mathrm{global}}(k) - \mathbf{x}^{\mathrm{i}}(k)) \end{aligned} \tag{4.25}$$

$$\mathbf{x}^{\mathrm{i}}(k+1) = \mathbf{x}^{\mathrm{i}}(k) + \mathbf{v}^{\mathrm{i}}(k+1), \tag{4.26}$$

where $\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k)$ corresponds to the best personal experienced position, i.e.

$$g(\mathbf{x}^{\mathrm{i}}_{local}(k)) < g(\mathbf{x}^{\mathrm{i}}(k)) \text{ for all } k, \tag{4.27}$$

and the particle's position $\mathbf{x}_{\mathrm{global}}(k)$ corresponds to the global best position corresponding to

$$g(\mathbf{x}_{global}(k)) < g(\mathbf{x}^{\mathrm{i}}(k)) \text{ for all } \mathbf{p}^{\mathbf{i}}, \text{ and for all } k. \tag{4.28}$$

The interpretation of each of the three terms in equation (4.25) is now explained. The first term defines the inertial weight, which determines the contribution of $\mathbf{v}^{\mathrm{i}}(k)$ to $\mathbf{v}^{\mathrm{i}}(k+1)$. This is referred to as the inertial factor $\chi \in (0, 1]$.

The second term $c_1 r_1(\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k) - \mathbf{x}^{\mathrm{i}}(k))$ is the local exploration factor of each particle $\mathbf{p}^{\mathrm{i}}(k)$, based on each particle's best personal experience $\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k)$. Note that exploration occurs along the direction of the vector $c_1 r_1(\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k) - \mathbf{x}^{\mathrm{i}}(k))$ as shown in Figure 4.2. The parameter c_1 is known in the literature as the local acceleration [84], and r_1 corresponds to a random variable in $[0, 1]$. Finally, the third term $c_2 r_2(\mathbf{x}_{\mathrm{global}}(k) - \mathbf{x}^{\mathrm{i}}(k))$ is the global exploitation factor of each particle $\mathbf{p}^{\mathrm{i}}(k)$, based on the best global position $\mathbf{x}_{\mathrm{global}}(k)$. Note that the exploitation occurs in the direction $c_2 r_2(\mathbf{x}_{\mathrm{global}}(k) - \mathbf{x}^{\mathrm{i}}(k))$ (Figure 4.2). The parameter c_2 is referred to as the social acceleration [84], and r_2 corresponds to a random variable in $[0, 1]$.

Figure 4.2 shows a diagram of the PSO algorithm's evolution. Assume a particle $\mathbf{p}^{\mathrm{i}}$ at position $\mathbf{x}^{\mathrm{i}}(k)$ and velocity $\mathbf{v}^{\mathrm{i}}(k)$, given $\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k)$, $\mathbf{x}_{\mathrm{global}}(k)$. The diagram shows the addition of each term from equation (4.25), i.e., the inertia weight, exploration, and exploitation terms. As a result, the updated position $\mathbf{x}^{\mathrm{i}}(k+1)$ of the particle $\mathbf{p}^{\mathrm{i}}$ evolves as a position nearer to both $\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k)$ and $\mathbf{x}_{\mathrm{global}}(k)$, as shown in Figure 4.2. Therefore, the PSO algorithm tries to determine the global optimum between the positions $\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k)$ and $\mathbf{x}_{\mathrm{global}}(k)$.

The application of the PSO method to point cloud data requires the definition of $\mathbf{x}^{\mathrm{i}}(k)$, $\mathbf{v}^{\mathrm{i}}(k)$, $\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k)$, $\mathbf{x}_{\mathrm{global}}(k)$, corresponding to particle i at iteration k. For the case of 3D PCD registration, these definitions are given as follows

$$\mathbf{x}^{\mathrm{i}}(k) = \begin{bmatrix} \alpha^{\mathrm{i}}(k) \\ \beta^{\mathrm{i}}(k) \\ \gamma^{\mathrm{i}}(k) \\ {t_x}^{\mathrm{i}}(k) \\ {t_y}^{\mathrm{i}}(k) \\ {t_z}^{\mathrm{i}}(k) \end{bmatrix} \quad , \quad \mathbf{v}^{\mathrm{i}}(k) = \begin{bmatrix} v^{\mathrm{i}}_{\alpha}(k) \\ v^{\mathrm{i}}_{\beta}(k) \\ v^{\mathrm{i}}_{\gamma}(k) \\ {v_{t_x}}^{\mathrm{i}}(k) \\ {v_{t_y}}^{\mathrm{i}}(k) \\ {v_{t_z}}^{\mathrm{i}}(k) \end{bmatrix} \tag{4.29}$$

$$\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k) = \begin{bmatrix} \alpha^{\mathrm{i}}_{\mathrm{local}}(k) \\ \beta^{\mathrm{i}}_{\mathrm{local}}(k) \\ \gamma^{\mathrm{i}}_{\mathrm{local}}(k) \\ t^{\mathrm{i}}_{x,\mathrm{local}}(k) \\ t^{\mathrm{i}}_{y,\mathrm{local}}(k) \\ t^{\mathrm{i}}_{z,\mathrm{local}}(k) \end{bmatrix} \quad , \quad \mathbf{x}_{\mathrm{global}}(k) = \begin{bmatrix} \alpha_{\mathrm{glocal}}(k) \\ \beta_{\mathrm{global}}(k) \\ \gamma_{\mathrm{global}}(k) \\ t_{x,\mathrm{global}}(k) \\ t_{y,\mathrm{global}}(k) \\ t_{z,\mathrm{global}}(k) \end{bmatrix}, \tag{4.30}$$

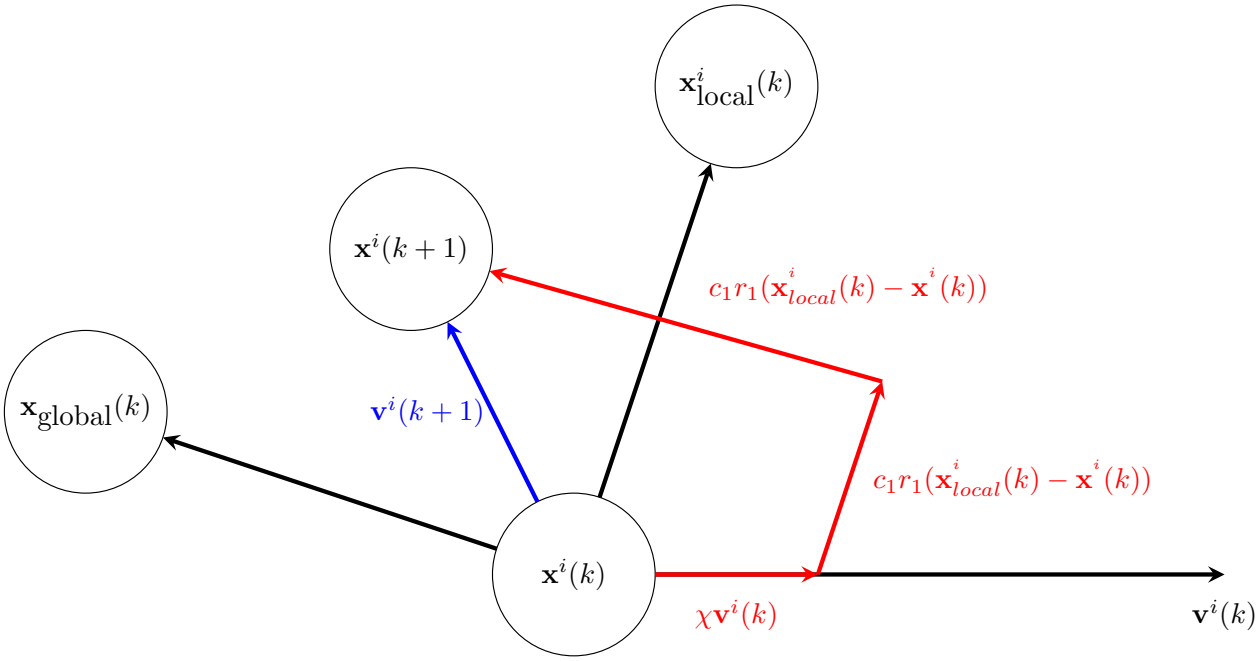


Figure 4.2: Intuitive explanation of the Particle Swarm Optimization algorithm, highlighting the pose recursion based on global position $\mathbf{x}_{\text{global}}(k)$ and local best experience $\mathbf{x}^{\text{i}}_{\text{local}}(k)$ of each member in the swarm.

where the first three components of each position vector $\mathbf{x}^{\text{i}}(k)$, $\mathbf{x}^{\text{i}}_{\text{local}}(k)$, $\mathbf{x}_{\text{global}}(k)$, correspond to the roll, α, pitch, β and yaw γ angles for the rotation matrix given by equation (4.3). Similarly, their last three components correspond to the translation vector components t_x, t_y and t_z. Therefore,

$$\begin{aligned} \mathbf{R}(k) &= \mathbf{R}(\alpha^{\text{i}}(k), \beta^{\text{i}}(k), \gamma^{\text{i}}(k)) && (4.31) \\ \mathbf{t}(k) &= [t_x{}^{\text{i}}(k), t_y{}^{\text{i}}(k), t_z{}^{\text{i}}(k)]. && (4.32) \end{aligned}$$

Conceptually, point set registration requires the correct computation of point/feature relative translation, rotation and inter scan detection errors. To initialize the PSO algorithm [84], the concept adopted is to spread $N-1$ particles randomly from rest, i.e. $\mathbf{v}^{\text{i}}(k=0) = 0$, with positions $\mathbf{x}^{\text{i}}(k=0)$ uniformly distributed in the domain of the objective function to be minimized. The domain used for the PSO-COLA method is given by:

$$\begin{aligned} t_x, t_y, t_z &\in [t_{min}, t_{max}] && (4.33) \\ \alpha, \gamma &\in [-\pi, \pi] && (4.34) \\ \beta &\in [-\pi/2, \pi/2]. && (4.35) \end{aligned}$$

Additionally, we include a single particle $\mathbf{p}^N$ at position $\mathbf{x}^N(k=0) = [\alpha_o, \beta_o, \gamma_o, t_{xo}, t_{yo}, t_{zo}]^\top$ from rest, where parameters α_o, β_o, γ_o t_{xo}, t_{yo} and t_{zo} satisfy

$$\begin{aligned} \mathbf{R}_o &= \mathbf{R}_z(\gamma_o)\mathbf{R}_y(\beta_o)\mathbf{R}_x(\alpha_o) && (4.36) \\ \mathbf{t}_o &= [t_{xo}, t_{yo}, t_{zo}]^\top, && (4.37) \end{aligned}$$

where $\mathbf{R}_o$ and $\mathbf{t}_o$ are computed utilizing the methodology shown in Section 4.3.

4.4.1 Robustness of the PSO-COLA Registration Algorithm

Stagnation is the phenomenon of premature convergence, which may generate erroneous results, such as local minima. This problem occurs in PSO methods. This phenomenon can occur due to several factors, such as poor initialization or due to the characteristics of the objective function f. To illustrate this phenomenon, consider the one-dimensional objective discontinuous function $f : \mathbb{R} \to \mathbb{R}$ such that

$$f(x) = \begin{cases} 1, & \text{if } x \neq 0, 1 \\ 0, & \text{if } x = 1 \\ -1, & \text{if } x = 0 \end{cases} \tag{4.38}$$

Note that f has a global minimum at $x = 0$ and a local minimum at $x = 1$.

Assume that N particles are available, initialized from rest, such that

$$f(x^{\mathrm{i}}(k=0)) = \begin{cases} 1, & \text{if i} = 1, \ldots, N-1 \\ 0, & \text{if i} = N. \end{cases} \tag{4.39}$$

Without loss of generality, assume that $|x^{\mathrm{i}}(k=0)| \gg 0$ for all $\mathrm{i} = 1, \ldots, N-1$. Then $x^{\mathrm{i}}_{local}(k=0)$ and $x_{global}(k=0)$ are determined as follows.

$$f(x^{\mathrm{i}}_{\mathrm{local}}(k=0)) = \begin{cases} 1, & \text{if i} = 1, \ldots, N-1 \\ 0, & \text{if i} = N \end{cases} \tag{4.40}$$

$$x_{\mathrm{global}}(k=0) = 1. \tag{4.41}$$

Then for $k < \infty$ iterations, it can be seen that the probability of obtaining the global position $x_{\mathrm{global}} = 0$ is zero due to the discontinuities of f. Therefore, the equation of motion of each particle $\mathbf{p}^{\mathrm{i}}$, with $\mathrm{i} = 1, \ldots, N-1$ will be:

$$\mathbf{v}^{\mathrm{i}}(k+1) = \chi \mathbf{v}^{\mathrm{i}}(k) \tag{4.42}$$

$$\mathbf{x}^{\mathrm{i}}(k+1) = \mathbf{x}^{\mathrm{i}}(k) + \mathbf{v}^{\mathrm{i}}(k+1). \tag{4.43}$$

Equation (4.42) shows that the evolution of velocities $\mathbf{v}^{\mathrm{i}}(k+1)$ is dominated by the inertial factor χ, reducing the velocity of the particles at each iteration. On the other hand, equation (4.43) shows that the particle with position $\mathbf{x}^{\mathrm{i}}(k+1)$ performs exploration in the neighborhood of $\mathbf{x}^{\mathrm{i}}(k)$ with velocity $\mathbf{v}^{\mathrm{i}}(k+1)$. Similarly, the equations of motion for the particle $\mathbf{p}^N$ are:

$$\mathbf{v}^N(k+1) = \chi \mathbf{v}^N(k) + (c_1 r_1 + c_2 r_2)(1 - \mathbf{x}^N(k)) \tag{4.44}$$

$$\mathbf{x}^N(k+1) = \mathbf{x}^N(k) + \mathbf{v}^N(k+1). \tag{4.45}$$

Equation (4.44) shows that the updated velocity $\mathbf{v}^N(k+1)$ is given by the addition of the inertial factor χ and the global exploitation factor $(c_1 r_1 + c_2 r_2)(1 - \mathbf{x}^N(k))$. Since in the example of the function defined in equation (4.38) $x_{\mathrm{global}}(k) \neq 0$, particle $\mathbf{p}^N$ performs a random search given by equation (4.45). Then, the solution will be stagnated at the local minimum $x = 1$ observing a premature convergence of the system to a local minimum.

In general, this problem highlights the importance of a good initialization, such that in the set of all particles, at least one of them (particle $\mathbf{p}^{\mathrm{i}}$) obeys the following

$$\mathbf{x}^{\mathrm{i}}_{local}(k=0) \approx \mathbf{x}_{global}(k=0). \tag{4.46}$$

4.4.2 Implementation of the PSO-COLA Registration Algorithm

Algorithm 2: The PSO-COLA algorithm

Input: Rotation Initialization $\mathbf{R}_o$, Translation Initialization $\mathbf{t}_o$, Number of Particle N, Maximum number of iterations K_{max}

Output: Rotation matrix $\mathbf{R}$, Translation matrix $\mathbf{t}$.

`// Initialize particles from rest with position`

$\mathbf{x}^{\mathrm{i}}(k) = [\alpha^{\mathrm{i}}(k), \beta^{\mathrm{i}}(k), \gamma^{\mathrm{i}}(k), {t_x}^{\mathrm{i}}(k), {t_y}^{\mathrm{i}}(k), {t_z}^{\mathrm{i}}(k)]$

for $i \leftarrow 1$ ***to*** N **do**
- Initialize $\mathbf{v}^{\mathrm{i}}(k) \leftarrow 0$
- Initialize $\mathbf{x}^{\mathrm{i}}(k)$ uniformly distributed with domains according equations (4.33), (4.34), (4.35)
- Initialize $\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k)$ using equation (4.27)
- Initialize $\mathbf{x}_{\mathrm{global}}(k)$ using equation (4.28)

end

`// Main loop`

for $k \leftarrow 1$ ***to*** K_{max} **do**
- **for** $i \leftarrow 1$ ***to*** N **do**
  - Update $\mathbf{v}^{\mathrm{i}}(k+1) \leftarrow \mathbf{v}^{\mathrm{i}}(k)$ using equation (4.25)
  - Update $\mathbf{x}^{\mathrm{i}}(k+1) \leftarrow \mathbf{x}^{\mathrm{i}}(k)$ using equation (4.26)
- **end**
- Update $\mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k+1) \leftarrow \mathbf{x}^{\mathrm{i}}_{\mathrm{local}}(k)$ using equation (4.27)
- Update $\mathbf{x}_{\mathrm{global}}(k+1) \leftarrow \mathbf{x}_{\mathrm{global}}(k)$ using equation (4.28)

end

Return $\mathbf{x}_{\mathrm{global}}(K_{max})$

Return $\mathbf{R}$, $\mathbf{t}$ using equations (4.3), (4.31) and (4.32).

4.4.3 Computational Complexity

The PSO-COLA algorithm, first requires the computation of the objective function $g(\mathbf{x}^{\mathrm{i}}(k))$, which, due to equations (4.11) and (4.24) is equivalent to the computation of the COLA metric-based on sets $\mathcal{M}_{\mathrm{trans}}(\mathbf{R}, \mathbf{t})$ and $\widehat{\mathcal{M}}$. The COLA metric's computational complexity is given by $\mathcal{O}(\max(|\mathcal{M}|, |\widehat{\mathcal{M}}|)^3)$. If a particle swarm with N particles is chosen, PSO-COLA must compute the COLA metric for each member in the swarm. Additionally, as can be seen in Algorithm 2, the number of iterations k of the PSO-COLA algorithm is restricted to a maximum value of $K_{\max}$. Therefore the computational complexity of the PSO-COLA registration algorithm is

$$\mathcal{O}(N \times k \times \max(|\mathcal{M}|, |\widehat{\mathcal{M}}|)^3), \tag{4.47}$$

with worst case complexity $\mathcal{O}(N \times K_{\max} \times \max(|\mathcal{M}|, |\widehat{\mathcal{M}}|)^3)$.

It can be seen from equation (4.47) that this complexity is dominated by the complexity of the COLA metric. This in turn is dependent upon the complexity of the Hungarian method, used to estimate the correspondences between points within the two data sets, making it potentially intractable for very large data sets.

As mentioned in Section 3.9, regarding the computational complexity of the COLA metric itself, computational improvements beyond the Hungarian method are available, which are reported to reduce the execution time by up to 90% [82]. Recently, further optimizations of the Jonker and Volgenant algorithm have also been reported, which could be applied to the PSO-COLA registration algorithm in future work [83].

Note that if the point correspondences between the datesets are given, no assignment is necessary, and the computational complexity of the COLA metric reduces to $\mathcal{O}(\min(\mathfrak{m}, \widehat{\mathfrak{m}}))$, resulting in a computational complexity for the PSO-COLA algorithm, with known correspondences, of

$$\mathcal{O}(N \times k \times \min(\mathfrak{m}, \widehat{\mathfrak{m}})), \tag{4.48}$$

which is linear with respect to the dimensions of the smaller point cloud set.

Chapter 5

Results

5.1 Introduction

This chapter presents the COLA metric mapping error performance and the PSO-COLA algorithm performance for point set registration. In the case of robotic mapping error and feature-based-SLAM, multi-object metrics are evaluated in real SLAM scenarios using the "Parque O'Higgins" dataset [25] as well as the "CityTrees10k" SLAM dataset [5] and mapping results based on iSAM. Finally, the performance evaluation of PSO-COLA as a point cloud registration solution is applied first in controlled scenarios based on the "Stanford Bunny" and "Dragon" datasets [26] . For real scenarios, PSO-COLA is evaluated using a real dataset called the "ETH Apartment" dataset [27].

5.2 Map Error Quantification Results

5.2.1 The "Parque O'Higgins" Data Set

In this section the performance of the Hausdorff, OMAT, OSPA and COLA metrics, as well as the SLAM trajectory energy metric of [9], will be analyzed by comparing their ability to evaluate real estimated SLAM results in a physically meaningful manner based on the Parque O'Higgins dataset shown in Figure 5.1.

Parque O'Higgins is a traditional urban park, founded in 1873 and located in Santiago, Chile. It is in the vicinity of the Faculty of Physical and Mathematical Sciences (FCFM) of the University of Chile. In addition, it has a perimeter fence, with an area of approximately 80 hectares. The "Parque O'Higgins" dataset was collected in a small region inside the entire park, highlighted with a blue square as shown in Figure 5.1, covering a surface with an approximate area of $16,800[m^2]$.

The results will demonstrate that the trajectory energy metric can yield map scores which contradict the OSPA, COLA and Hausdorff map metrics for a given SLAM result. Further,

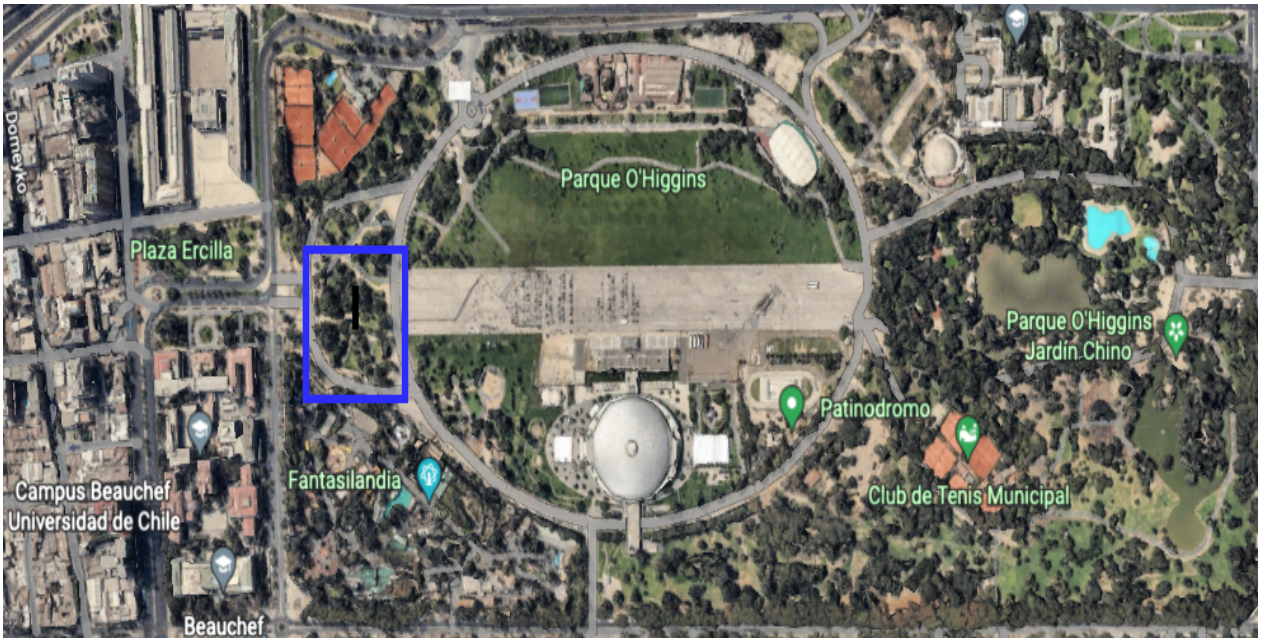


Figure 5.1: *An aerial view of Parque O'Higgins, Santiago, Chile. The "Parque O'Higgins" dataset was collected within the highlighted blue region, which contains a small forest, in the presence of people carrying out outdoor activities.*

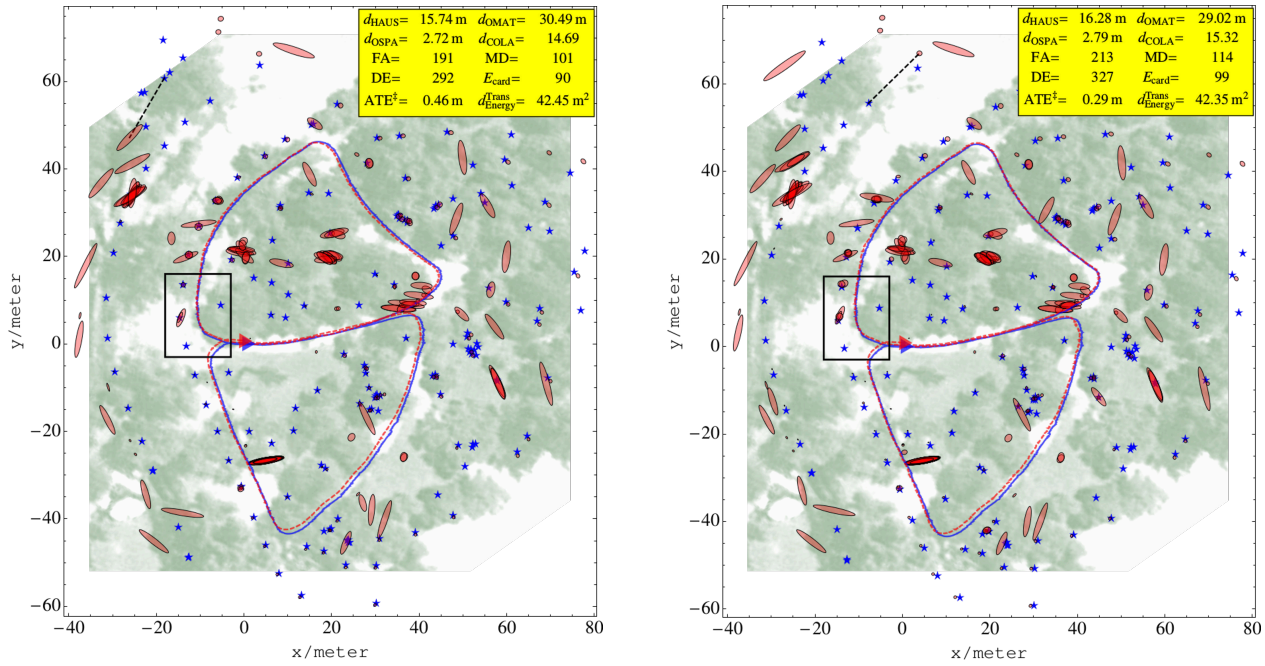
while the OSPA metric often saturates to its maximum cut-off value c, the COLA metric does not, allowing maps to be compared and, in contrast to the OSPA metric, penalizes missed detections and false alarms equally. The effect of varying c will also be further elucidated.

Fig. 5.2 shows the SLAM estimates from two different SLAM algorithms[1], which were designed to estimate 2D vehicle trajectories and maps corresponding to the x, y location of the centers of tree trunks. Each algorithm is referred to as "SLAM Alg. 1" and "SLAM Alg. 2" and each result is superimposed onto a satellite image of the area to show the tree coverage and spacing corresponding to the vehicle's "Figure 8" trajectory. The GT trajectories (blue lines) were obtained via manual scan-matching[2] due to the lack of reliable GPS in the environment, and the red dashed lines represent the estimated trajectories from each SLAM algorithm. The blue stars represent the GT features (tree trunk center locations), again obtained through independent, manual scan matching procedures[3]. The red ellipses are centered at the estimated feature locations and represent the covariances for each estimated feature. The ellipses shown correspond to "3-Sigma ellipses", which from 2 degree-of-freedom Chi-squared tables, correspond to a theoretical probability mass of a feature being within

[1]The estimated SLAM solutions are based on MH-FastSLAM [3] and Rao-Blackwellized (RB)-PHD-SLAM [7].

[2]This corresponds to manually identifying points in successive scans from identifiable tree trunks, and determining the corresponding vehicle displacement to align such points within the global coordinate system.

[3]The GT map is also of limited precision since the laser data, used for scan matching, is prone to range errors of up to 5cm, and more importantly, the determination of the centers of the trees was prone to errors of up to 20cm. Importantly, these distance errors are significantly less than the average mapping distance errors and the GT map is guaranteed to contain all circular objects detectable by the algorithm used here.



(a) SLAM Alg. 1 (b) SLAM Alg. 2

Figure 5.2: *GT and estimated trajectories from SLAM Algs. 1 (a) and 2 (b). The GT feature locations (blue stars) are superimposed onto a satellite image of the park. The estimated maps produced by SLAM Algs. 1 (a) and 2 (b) are shown as red "3-sigma" confidence interval ellipses. The dashed line in each figure represents the Hausdorff distance between each map set. "FA" = no. false alarms, "MD" = no. missed detections, "DE" = no. detection errors (= MD + FA) and* $E_{card} = ||\widehat{\mathcal{M}}| - |\mathcal{M}||$.

each ellipse of 0.99. Hence, $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})$ in (2.9) can be the Mahalanobis distance such that

$$\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{\sigma(\mathrm{i})}) = \min\left(c, \sqrt{(\mathbf{m}^{\mathrm{i}} - \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})^{\top} (\mathbf{P}^{\mathrm{i}})^{-1} (\mathbf{m}^{\mathrm{i}} - \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})}\right)$$

where $c = 3.00$ and $\mathbf{P}^{\mathrm{i}}$ is the covariance matrix corresponding to estimated feature $\widehat{\mathbf{m}}^{\sigma(\mathrm{i})}$ and GT feature $\mathbf{m}^{\mathrm{i}}$. In this analysis it is assumed that the error associated with all GT features is zero[4] and that the estimated feature covariance values are available from the SLAM estimator. In each figure, the Hausdorff ($\mathrm{d}_{\mathrm{HAUS}}$), OMAT ($\mathrm{d}_{\mathrm{OMAT}}$), OSPA ($\mathrm{d}_{\mathrm{OSPA}}$), COLA ($\mathrm{d}_{\mathrm{COLA}}$) errors, numbers of False Alarms (FA), Missed Detections (MD), Detection Errors (DE = FA + MD) and estimated cardinality ($E_{\mathrm{card}} = ||\widehat{\mathcal{M}}| - |\mathcal{M}||$) errors are provided. In all of the experiments, $p = 2$, which according to [14] yields smooth distance curves, and is commonly used in other metrics, such as the L_2 distance.

Performance of the Trajectory Energy Metric

The energy metric proposed in [9], which calculates the energy required to deform the estimated (red dashed) trajectory $\widehat{\mathbf{X}}_{0:k}$ to the GT (blue solid) trajectory $\mathbf{X}_{0:k}$, can be calculated

[4]In the target tracking literature, [56] applied the OSPA metric-based on a Hellinger distance metric, in which the GT target covariances were replaced with their Cramer-Rao lower bound values.

based on the trajectories shown in Fig. 5.2. In Fig. 5.3 the values of the translational component of the energy metric, $\mathrm{d}^{\mathrm{Trans}}_{\mathrm{Energy}}(\mathbf{X}_{0:k}, \widehat{\mathbf{X}}_{0:k})$ over time, given the GT trajectory, during each SLAM run are shown. The final values of $\mathrm{d}^{\mathrm{Trans}}_{\mathrm{Energy}}(\mathbf{X}_{0:k}, \widehat{\mathbf{X}}_{0:k})$ for SLAM Algs. 1 and 2 suggests

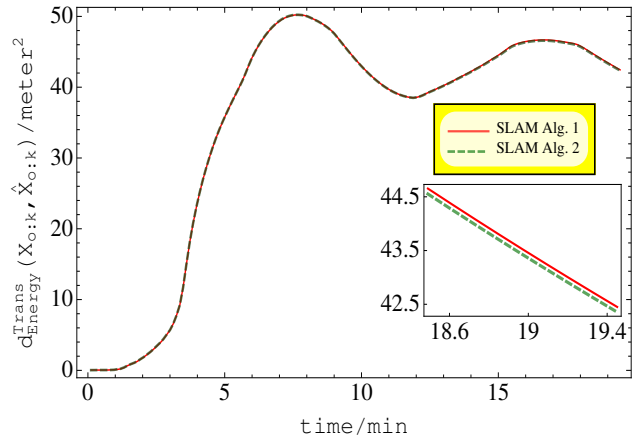


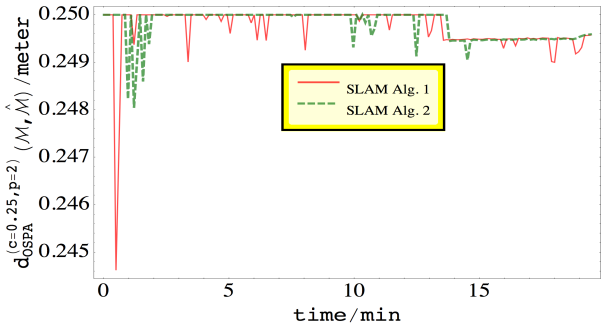
Figure 5.3: *The translational component of the energy metric applied to SLAM Alg. 1 (Red line) and SLAM Alg. 2 (Green dashed line).*

that SLAM Alg. 2 slightly out-performs SLAM Alg. 1. Interestingly, this contradicts the final OSPA, COLA and Hausdorff metric evaluations, shown in Fig. 5.2 and also contradicts most of the time varying OSPA and COLA metric values recorded during the SLAM runs, shown in Fig. 5.4 for two different values of c. This demonstrates the necessity of gauging mapping as well as trajectory performance.

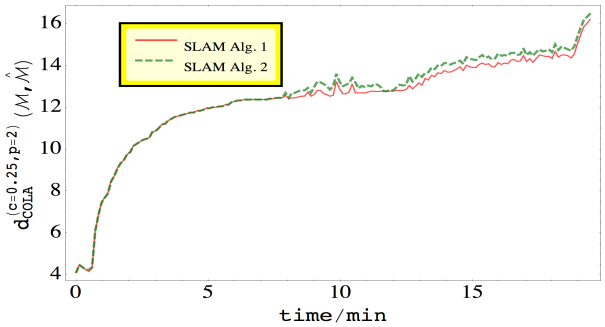
Performance of Multi-Object Metrics

Referring to Fig. 5.4, it is important to note that both the OSPA and COLA map error evaluations are based on the time varying number of features which have theoretically been covered by the vehicle's sensor's field of view (GT) and those which have been estimated at each time. Setting the Mahalanobis distance based cut-off parameter $c = 0.25$ in both metrics, Fig. 5.4a shows that during most of the SLAM trial, the OSPA metric for both algorithms saturates to its maximum value $c = 0.25$. This indicates that most of the estimated spatial errors are larger than 0.25m within both SLAM estimates. It should be noted however, that under these circumstances, the OSPA metric fails to give any indication as to which algorithm is superior, even in terms of the detection (map cardinality) errors. Conversely however, the corresponding COLA metric plots (Fig. 5.4b) show that although the first 8 minutes of each SLAM trial perform similarly, after this time, it is possible to define which algorithm performs better (usually SLAM Alg. 1) despite most estimates having Mahalanobis distances larger than $c = 0.25$. This is due to the COLA metric's ability to continue gauging cardinality errors, even when the OSPA metric saturates.

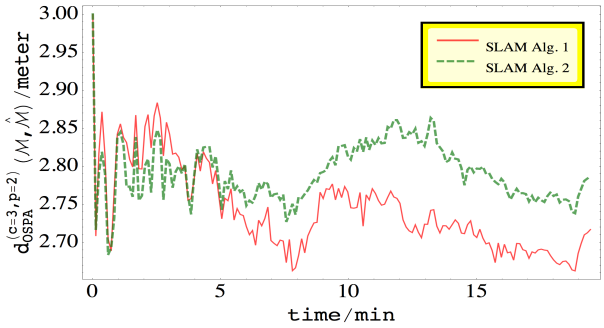
In Figs. 5.4c and 5.4d, the OSPA and COLA metrics are again plotted versus time, during the same SLAM experiments, but this time with cut-off parameter $c = 3.0$. Now, as shown in [14], the values of the OSPA metric increase for both SLAM algorithms. However, now



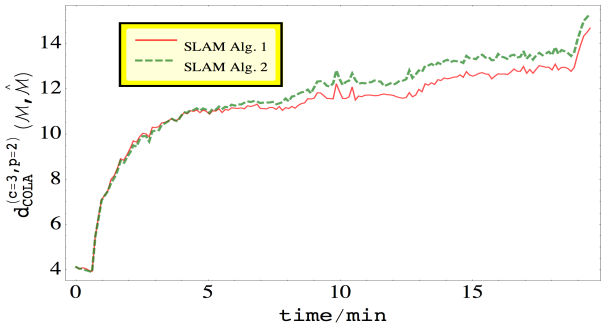
(a) OSPA metric vs time ($c = 0.25$ and $p = 2$).



(b) COLA metric vs time ($c = 0.25$ and $p = 2$).



(c) OSPA metric vs time ($c = 3.00$ and $p = 2$).



(d) COLA metric vs time ($c = 3.00$ and $p = 2$).

Figure 5.4: *The OSPA and COLA metrics versus time for SLAM Algs. 1 and 2, with differing values of c.*

SLAM Alg.	$\widehat{\mathfrak{m}}$	Gated	MD	FA	$\mathrm{d}_{\mathrm{COLA}}$	$\mathrm{d}_{\mathrm{OSPA}}$ [m]	$\mathrm{d}_{\mathrm{HAUS}}$ [m]	$\mathrm{d}_{\mathrm{OMAT}}$ [m]	$\mathrm{d}^{\mathrm{Trans}}_{\mathrm{Energy}}$ [m^2]	NEES*
1	263	72	101	191	14.69 (1)	2.72 (1)	15.74 (1)	30.49 (2)	42.45 (2)	231.33 (2)
2	272	59	114	213	15.32 (2)	2.79 (2)	16.28 (2)	29.02 (1)	42.35 (1)	159.34 (1)

Table 5.1: *Detection errors and map metric results for SLAM Algs. 1 and 2 ($c = 3.00$ and $p = 2$). $\widehat{\mathfrak{m}}$ = number of estimated features, "Gated" = number of gated features, "MD" = number of Missed Detections, "FA" = number of False Alarms. The bracketed numbers give the ranking by each metric. *The NEES metric was calculated only based on the number of gated features, ignoring detection errors.*

that more features are assigned to (gated with) GT values, the curves in Fig. 5.4c remain unsaturated, and provide a significant difference in their judgement of SLAM Algs. 1 and 2, at all times. Note that as time increases, the effective field of view of the sensed area has increased. The COLA metric, in the right hand graphs of Fig. 5.4, shows its tendency to increase as it is unforgiving to an increase in cardinality error. On the other hand, the OSPA metric, in Fig. 5.4c shows its averaging nature as the number of erroneous feature estimates increases with increasing sensor coverage.

Table 5.1 shows the number of gated features, MD and FA and $\mathrm{d}_{\mathrm{COLA}}$, $\mathrm{d}_{\mathrm{OSPA}}$ and $\mathrm{d}_{\mathrm{OMAT}}$ based on a validation gate $c = 3.00$. $\mathrm{d}_{\mathrm{HAUS}}$ and $\mathrm{d}^{\mathrm{Trans}}_{\mathrm{Energy}}$ are also given. The NEES metric, undefined for the full estimated and GT maps since their cardinalities are different, has been calculated based purely on the gated features (for $c = 3.00$). Note that both $\mathrm{d}^{\mathrm{Trans}}_{\mathrm{Energy}}$ and the NEES metric, often used in gauging SLAM performance, favor SLAM Alg. 2, disagreeing

with the multi-object metrics $\mathrm{d}_{\mathrm{COLA}}$, $\mathrm{d}_{\mathrm{OSPA}}$ and $\mathrm{d}_{\mathrm{HAUS}}$. The NEES metric can only be calculated for the 59 gated, out of the 272 estimated, features in SLAM Alg. 2, and ignores the remaining estimates, even though they constitute detection errors. For SLAM Alg. 1, the NEES metric gave a higher value, purely because more features were gated (72).

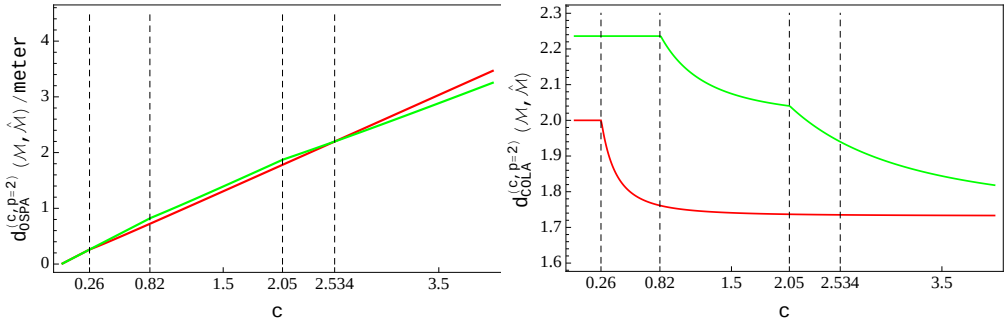
In light of Table 5.1, SLAM Alg. 1 has less detection errors and more gated features and $\mathrm{d}_{\mathrm{COLA}}$, $\mathrm{d}_{\mathrm{OSPA}}$ and even $\mathrm{d}_{\mathrm{HAUS}}$ all confirm its mapping superiority over SLAM Alg. 2.

Assessing Metric Performance Based on a Sub-Map

To highlight the differences between the OSPA and COLA metrics, the estimated and GT sub-maps in the boxes in Figs. 5.2a and 5.2b are analyzed for various values of c. Comparing the smaller sub-maps simplifies the assessment of the COLA and OSPA metrics' *intuitive performances*. In these boxes, the number of GT features is $|\mathcal{M}| = 4$, the number of estimates for SLAM Alg. 1 is $|\widehat{\mathcal{M}_1}| = 3$ and that for the SLAM Alg. 2 is $|\widehat{\mathcal{M}_2}| = 5$. Under these conditions, it will be shown that there are values of c where (3.53), (3.54) and (3.58) are satisfied, meaning that the OSPA and COLA metrics can disagree.

The Impact of c on the OSPA and COLA Metrics

Figs. 5.5a and 5.5b show the values of the OSPA and COLA metrics, for the sub-maps within Figs. 5.2a and 5.2b, for $p = 2$ and varying c. As shown in [14], $\mathrm{d}_{\mathrm{OSPA}}$ increases with



(a) *The OSPA metric versus c, for p = 2.*

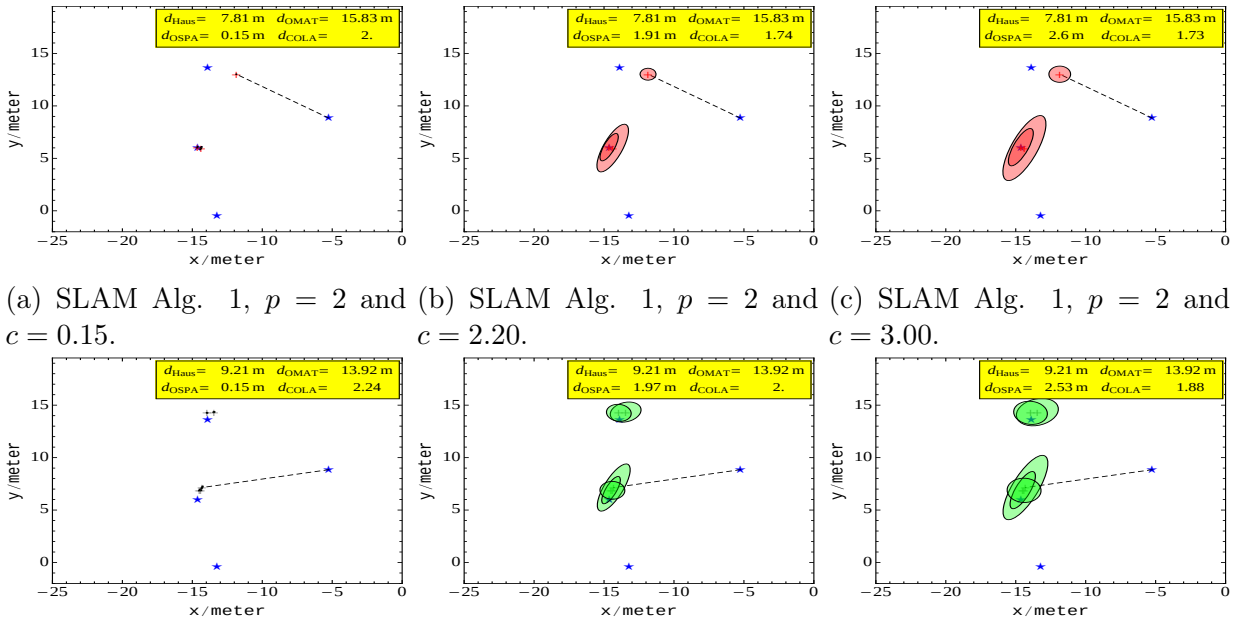
(b) *The COLA metric versus c, for p = 2.*

Figure 5.5: *The OSPA and COLA metrics versus c. The red and green curves shows the metric values for SLAM Algs. 1 and 2 respectively.*

c, whereas, as shown in (3.40), $\mathrm{d}_{\mathrm{COLA}}$ decreases. The gradient changes in Figs. 5.5a and 5.5b occur when estimated features become gated. The metric evaluations of SLAM Algs. 1 and 2, based on this sub-map, now follow for three values of c.

a) Figs. 5.6a and 5.6d show the performances of SLAM Algs. 1 and 2, and their metric values, when $c = 0.15$. Each estimated feature is surrounded by its validation gate ellipsoid, corresponding to the region in which the Mahalanobis distance

$$\sqrt{(\mathbf{m}^{\mathrm{i}} - \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})^{\top} (\mathbf{P}^{\mathrm{i}})^{-1} (\mathbf{m}^{\mathrm{i}} - \widehat{\mathbf{m}}^{\sigma(\mathrm{i})})} < c,$$



(a) SLAM Alg. 1, $p = 2$ and $c = 0.15$.

(b) SLAM Alg. 1, $p = 2$ and $c = 2.20$.

(c) SLAM Alg. 1, $p = 2$ and $c = 3.00$.

(d) SLAM Alg. 2, $p = 2$ and $c = 0.15$.

(e) SLAM Alg. 2, $p = 2$ and $c = 2.20$.

(f) SLAM Alg. 2, $p = 2$ and $c = 3.00$.

Figure 5.6: *Metric values for $p = 2$ and three different values of the cut-off parameter c.*

in equation (5.1). Note that for $c = 0.15$, these ellipses are barely visible, but will be for larger values of c. Since all of the GT features are ungated, the map estimation result of SLAM Alg. 1 constitutes 4 missed detections and 3 false alarms (a total of 7 detection errors). The result of SLAM Alg. 2 constitutes 4 missed detections and 5 false alarms (a total of 9 detection errors). For both SLAM Algs. 1 and 2, OSPA saturates to its cut-off value $c = 0.15$ and is unable to differentiate between the two. The COLA metric (intuitively) penalizes SLAM Alg. 2 more than SLAM Alg. 1, showing its dependence on the cardinality of the larger set.

b) Figs. 5.6b and 5.6e show the case when $c = 2.20$. Note the significantly larger validation gates, centered on each estimate. Two feature estimates are gated by SLAM Alg. 2, implying a total of 2 missed detections and 3 false alarms. Interestingly, this matches the total number of detection errors exhibited by SLAM Alg. 1 (3 missed detections and 2 false alarms). For SLAM Alg. 1, the COLA metric has almost settled to a steady state value wrt c, as indicated by the red curve in Fig. 5.5b.

c) Figs. 5.6c and 5.6f show the case when $c = 3.00$. In the case of SLAM Alg. 1, its estimate that was gated when $c = 2.20$ becomes statistically better localized (due to a more tolerant, larger validation gate), however 3 missed detections and 2 false alarms remain. For SLAM Alg. 2, two of its previously gated estimates become statistically better localized as c increases, still leaving 2 missed detections and 3 false alarms - i.e. still the same cardinality error as SLAM Alg. 1. Interestingly, as verified in Fig. 5.5a, the OSPA metric now reverses its decision, favoring SLAM Alg. 2 over SLAM Alg. 1, in contrast to the COLA metric. Note that only in this case are (3.53), (3.54) and (3.58) simultaneously satisfied, indicating disagreement between the OSPA and COLA metrics. It can be seen from Fig. 5.5a, that the OSPA metric decisions are very sensitive to small changes in gradient, when features are gated. Initially, for $c \leq 0.26$, $\mathrm{d}_{\mathrm{OSPA}}^{(c,p=2)}(\mathcal{M}, \widehat{\mathcal{M}}) = c$, and as c increases, a small number of

features start to gate, causing the OSPA metric to only lower its value very slightly below c. Since $\mathrm{d}_{\mathrm{OSPA}}^{(c,p=2)}(\mathcal{M}, \widehat{\mathcal{M}}_1)$ and $\mathrm{d}_{\mathrm{OSPA}}^{(c,p=2)}(\mathcal{M}, \widehat{\mathcal{M}}_2)$ are both very similar, and close to c for most values of c, the slight gradient changes which occur when features gate, result in non-intuitive changes in the mapping performance decisions of the OSPA metric. This problem is avoided in the COLA metric, as can be seen in Fig. 5.5b.

In the limit, as $c \to \infty$, the 3 estimates in SLAM Alg. 1 will gate with 3 of the 4 GT features. Similarly, for SLAM Alg. 2, 4 of its 5 estimates will all gate with the 4 GT features. Therefore, (2.9) shows that for SLAM Alg. 1, $\mathrm{d}_{\mathrm{OSPA}}^{(c,p=2)}(\mathcal{M}, \widehat{\mathcal{M}}_1) \to (1/\sqrt{4})c$ and for SLAM Alg. 2, $\mathrm{d}_{\mathrm{OSPA}}^{(c,p=2)}(\mathcal{M}, \widehat{\mathcal{M}}_2) \to (1/\sqrt{5})c$, demonstrating the averaging ability of the OSPA metric, since for SLAM Alg. 1, 1 out of 4 of the GT features remains unassigned and for SLAM Alg. 2, 1 out of 5 of the estimated features remains unassigned. This is also a particular case of the analysis presented in Section 3.7.4, with $\mathfrak{m} = 4$ and $\mathfrak{a} = 1$, in which (3.50) and (3.51) confirm the above values, showing the OSPA metric's lower penalization of false alarms compared with missed detections.

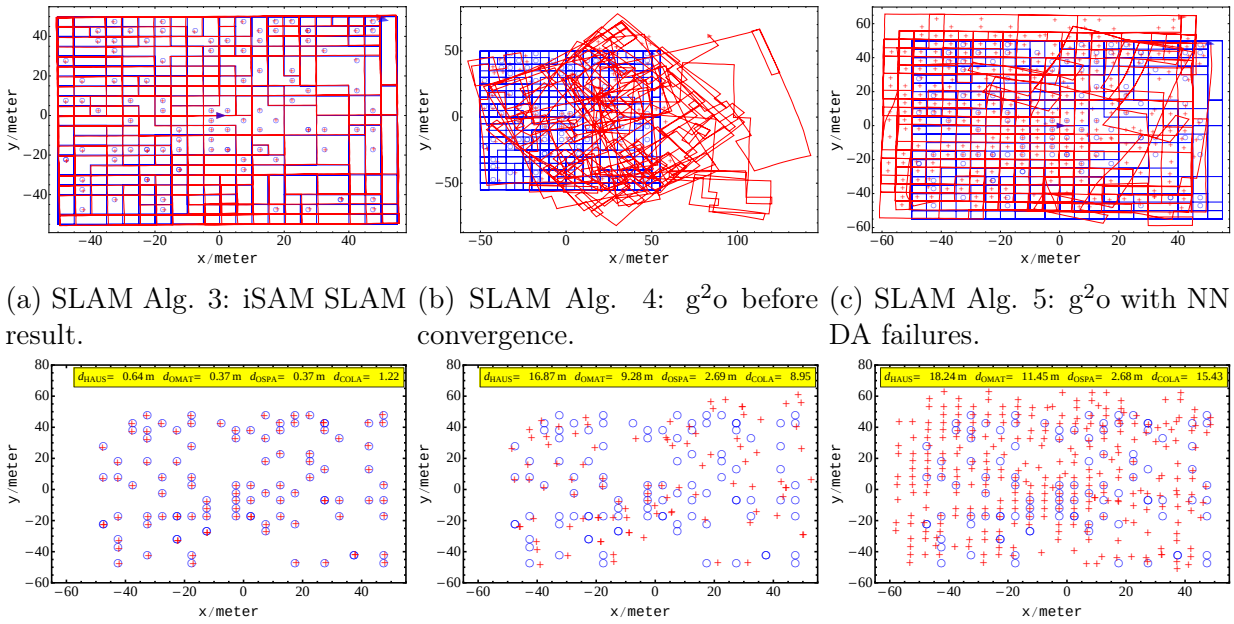
For the COLA metric, (3.2) shows that as $c \to \infty$, under a similar analysis, $\mathrm{d}_{\mathrm{COLA}}^{(c,p=2)}(\mathcal{M}, \widehat{\mathcal{M}}_1)$ and $\mathrm{d}_{\mathrm{COLA}}^{(c,p=2)}(\mathcal{M}, \widehat{\mathcal{M}}_2) \to \sqrt{1}$, again demonstrating the COLA metric's nature of gauging cardinality, rather than averaged distance error, and its ability to equally penalize missed detections and false alarms.

The dashed lines in Fig. 5.6 indicate the Hausdorff distances. The OMAT metric disagrees with all the other metrics. Note that it made assignments only within the considered region.

5.2.2 The "CityTrees10k" Data Set

To test the intuitive behavior of each metric with a publicly available data set, "CityTrees10k" was chosen, due to its available GT map [5]. This data set comprises 10,000 simulated robot movements and 100 GT features (simulated trees) which can be detected within a field of view of 10 meters around the robot. iSAM [5, 6] and the g^{2}o graph optimization SLAM solver [85], with perfect and estimated DA, were applied, yielding three solutions, SLAM Algs. 3, 4 and 5. Fig. 5.7a (SLAM Alg. 3) shows the SLAM result based on perfect DA (provided with the data set) using iSAM. Fig. 5.7d shows the corresponding GT and estimated maps only. Figs. 5.7b (SLAM Alg. 4) and 5.7e show the SLAM and mapping results respectively, with perfect DA, but before complete convergence of the g^{2}o solver, yielding a distorted trajectory. As shown in Fig. 5.7e, due to perfect DA, all of the feature locations are reasonably close to GT, since they were estimated within the less erroneous, initial part of the trajectory. Figs. 5.7c (SLAM Alg. 5) and 5.7f show the SLAM and mapping results respectively, of allowing the g^{2}o solver to converge, when replacing the given (perfect) DAs with an automated Nearest Neighbor (NN) DA algorithm, based on a Euclidean gating distance of 3m.

Table 5.2 summarizes the metric evaluations of Fig. 5.7. SLAM Alg. 3 appears to perform best, as reflected by all the metrics. Some interesting differences occur in each metric's ranking of SLAM Algs. 4 and 5. Intuitively, SLAM Alg. 5 may initially appear superior than SLAM Alg. 4, since the trajectory appears to bear a closer relationship to GT, as indicated in Fig. 5.7c, and the Trajectory Energy Metric $\mathrm{d}_{\mathrm{Energy}}^{\mathrm{Trans}}$ in Table 5.2. However,



(a) SLAM Alg. 3: iSAM SLAM result.

(b) SLAM Alg. 4: g^2o before convergence.

(c) SLAM Alg. 5: g^2o with NN DA failures.

(d) SLAM Alg. 3: Estimated and GT maps.

(e) SLAM Alg. 4: Estimated and GT maps.

(f) SLAM Alg. 5: Estimated and GT maps.

Figure 5.7: *"CityTrees10k" data set SLAM and mapping results with known data association and differing numbers of the g^2o solver iterations and with automatic Nearest Neighbor (NN) data association. Blue lines/circles: GT trajectory/map, red lines/circles: corresponding estimates.*

SLAM Alg.	$\widehat{\mathfrak{m}}$	Gated	MD	FA	d_{COLA}	d_{OSPA} [m]	d_{HAUS} [m]	d_{OMAT} [m]	d^{Trans}_{Energy} [m^2]	SSEE*
3	100	100	0	0	1.22 (1)	0.37 (1)	0.64 (1)	0.37 (1)	0.893 (1)	13.47 (1)
4	100	35	65	65	8.95 (2)	2.69 (3)	16.87 (2)	9.28 (2)	39.214 (3)	136.54 (3)
5	299	68	32	231	15.43 (3)	2.68 (2)	18.24 (3)	11.45 (3)	10.974 (2)	64.73 (2)

Table 5.2: *Detection errors and map metric results for SLAM Algs. 3 to 5 ($c = 3.00$ and $p = 2$). *The SSEE metric was calculated only based on the number of gated features, ignoring detection errors. The bracketed numbers give the ranking by each metric.*

SLAM Alg. 5 has committed multiple data association errors, causing it to vastly over estimate the number of map features ($\widehat{\mathfrak{m}} = 299$) (compare Figs. 5.7f and 5.7e). From a mapping perspective, intuition dictates that SLAM Alg. 5 is worst, with many more false alarms than the map produced by SLAM Alg. 4, and, contrary to the OSPA metric, the COLA metric correctly reflected this. The reason that OSPA, non-intuitively reports a lower mapping error can be seen from (2.9), where the averaging nature of the OSPA metric requires division by $\widehat{\mathfrak{m}}$. Since in SLAM Alg. 4 (Fig. 5.7e) $\widehat{\mathfrak{m}} = 100$ a higher OSPA value results than for SLAM Alg. 5 (Fig. 5.7e), with $\widehat{\mathfrak{m}} = 299$. Note that in the case of SLAM Algs. 4 and 5, (3.53), (3.54) and (3.58) are obeyed, indicating that disagreement between the COLA and OSPA metrics will result.

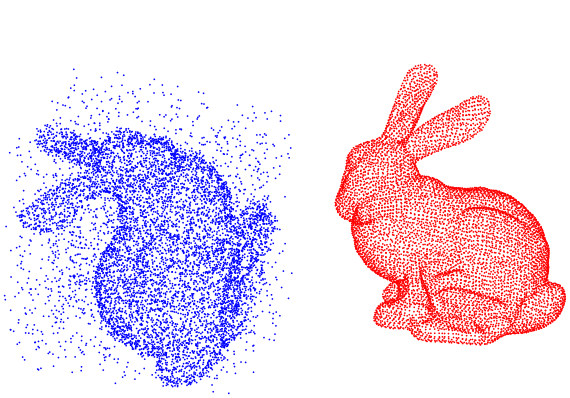
For comparison, and since no feature covariance information was used, the Sum of Squared Euclidean Errors (SSEE) values are given in Table 5.2 which, as in the case of the NEES metric, can only be calculated based on the number of gated features. Although SLAM Alg. 5 has gated more features (68) than SLAM Alg. 4 (35), the sum of their localization errors is lower, resulting in a lower SSEE value. Since the SSEE metric ignores detection errors, it

disagrees with the COLA metric.

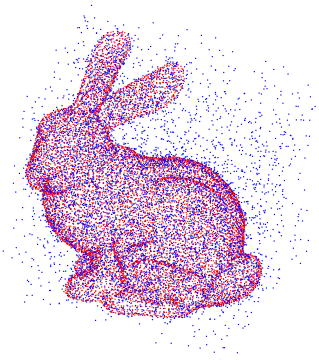
5.3 Point Cloud Data Registration Results

5.3.1 Synthetic Data: The Stanford Bunny and Dragon datasets

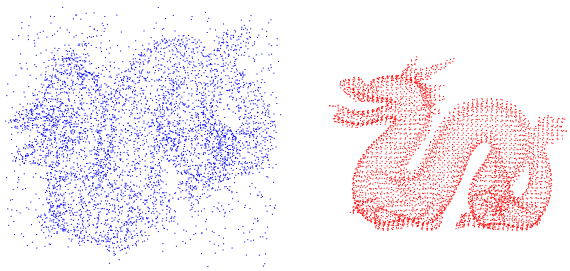
This section shows the 3D PCD data registration performance of PSO-COLA in comparison to state-of-the-art methods. The datasets used are the publicly available Stanford Bunny and Dragon datasets [26] as shown in Figure 5.8. For this purpose, we perform (i) experiments which evaluate registration performance in the presence of spatial noise, (ii) experiments that simulate cardinality differences between the model and reference sets (iii) experiments that simulate overlapping variations between the two sets while keeping them of equal cardinality. Additionally, the evaluation of performance is divided into two parts: (a) we evaluate the rotation error as $|\arccos((\mathrm{tr}(\hat{\mathbf{R}}^{\top}\mathbf{R})-1)/2)|$ which is the geodesic distance between the ground truth rotation $\mathbf{R}$ and its estimate $\hat{\mathbf{R}}$ [70] and (b), the translation error, defined as $||\mathbf{t}-\widehat{\mathbf{t}}||$, the L_2 norm of the difference between the ground truth translation $\mathbf{t}$ and its estimate $\widehat{\mathbf{t}}$.



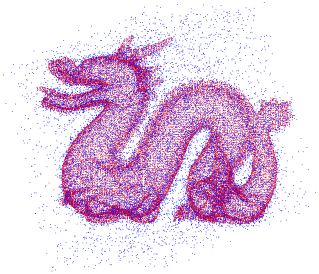
(a) *Unregistered scenario for the Bunny dataset.*



(b) *Registration solution based on PSO-COLA for the Bunny dataset.*



(c) *Unregistered scenario for the Dragon dataset.*



(d) *Registration solution based on PSO-COLA for the Dragon dataset.*

Figure 5.8: *Representation of unregistered and registered point clouds. The reference set $\mathcal{M}$ is shown as red points, while the model set $\widehat{\mathcal{M}}$ is presented as blue points. Note that the model set is corrupted with spatial and detection uncertainty.*

Description of the Experiments

For the construction of the simulated model dataset $\widehat{\mathcal{M}}$, in a manner similar to the procedure given in [70], the reference dataset $\mathcal{M}$ is first normalized to fit within a cube of dimension $[0,1]^3$. Subsequently, the reference dataset $\mathcal{M}$ is transformed into $\mathcal{M}_{\text{trans}}(\mathbf{R},\mathbf{t})$ (equation (4.2)) and corrupted with spatial noise via the following equation,

$$\widehat{\mathbf{m}}^{\mathrm{i}} = \mathbf{R}^k \mathbf{m}^{\mathrm{i}} + \mathbf{t}^k + \mathbf{w}^{\mathrm{i}}, \tag{5.1}$$

where $\mathbf{m}^{\mathrm{i}} \in \mathcal{M}$, $\widehat{\mathbf{m}}^{\mathrm{i}} \in \widehat{\mathcal{M}}$, $\mathbf{R}^k$ and $\mathbf{t}^k$ are the ground truth rotation and translation respectively. In all the experiments, the model dataset is corrupted with Gaussian noise samples $\mathbf{w}^{\mathrm{i}}$ drawn from a zero mean multivariate Normal distribution $\mathcal{N}(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \text{diag}(\sigma^2))$ where σ is the chosen standard deviation of the spatial noise. Note that the addition of random noise implies that the transformed reference data set is no longer necessarily restricted to a unit cube.

The possibility of detection errors is accounted for in the following manner.

1. *False alarms* are simulated by the addition of false points to the transformed reference data set. The number of added false alarms is $p_{FA} \times |\mathcal{M}|$ where p_{FA} is the probability of false alarm. Since the transformed, noisy data set is no longer necessarily restricted to a unit cube, the added false alarms are uniformly distributed within a sphere of radius $r = 2[m]$, centered at the center of the transformed unit cube [70].
2. *Miss-detections* are simulated by the random extraction of $p_{MD} \times |\mathcal{M}|$ points, where p_{MD} is the probability of miss-detection.

The resulting transformed, spatially corrupted and contaminated data set containing detection errors, defines the *model point cloud data set* $\widehat{\mathcal{M}}$.

Finally, for comparison purposes, an overlapping simulation is performed as in [70]. This corresponds to the case in which a percentage p_o of points in the model dataset are replaced by random points within a sphere of radius $r = 2[m]$. Therefore in the results, each registration algorithm's performance is also demonstrated as a function of the reference to model dataset percentage *overlap*, $100 - p_o$.

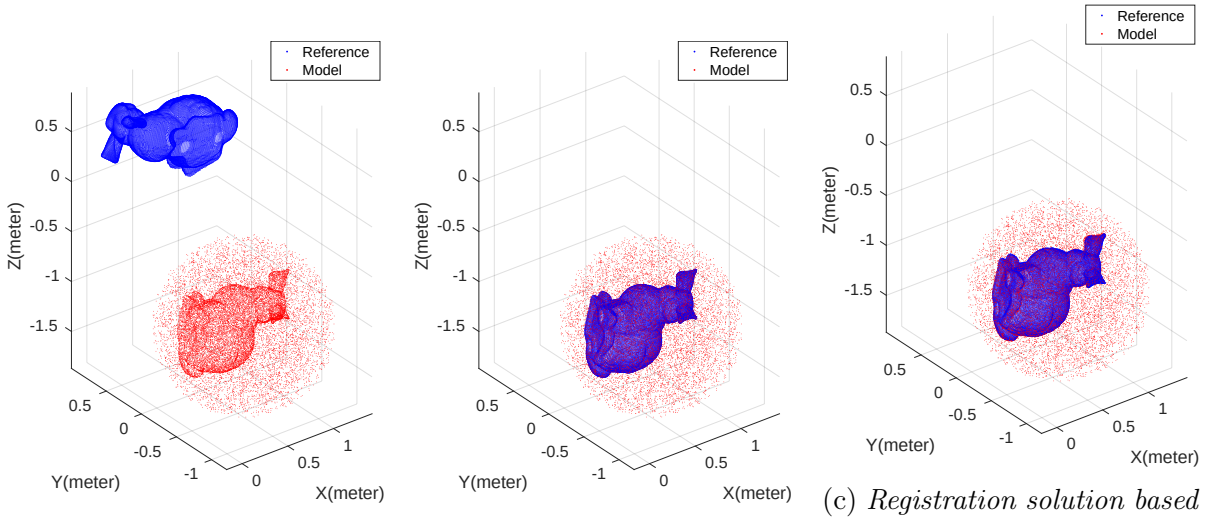
The number of simulations consists of 10 different rotations and translations computed with 40 Monte Carlo runs for each registration algorithm. Table 5.3 shows the roll, α, pitch, β, and yaw, γ, angles as well as the x, y and z translations, t_x, t_y and t_z, used in each of the 10 transformation experiments. Figures 5.9 and 5.10 show the registration result for the Bunny and Dragon dataset, using transformation $\mathbf{H}_1$ (see Table 5.3).

In the particular case of the Go-ICP registration algorithm, a trimming factor is available which allows the user to preselect what may be assumed to be the percentage of possible incorrect correspondences between $\mathcal{M}$ and $\widehat{\mathcal{M}}$, with the aim of improving robustness in the presence of outliers. Therefore, the Go-ICP registration algorithm was executed with different values of this trimming factor, namely 30, 60, and 90%, as well as the standard Go-ICP algorithm with 0% trimming factor.

Transformation	α	β	γ	t_x	t_y	t_z
$\mathbf{H}_1$	113.3005	73.0425	-134.2847	0.4134	0.1324	-0.4025
$\mathbf{H}_2$	-79.7406	8.4387	164.7025	0.4649	-0.3424	0.4706
$\mathbf{H}_3$	164.5801	-2.6324	108.101	-0.3581	-0.0782	0.4157
$\mathbf{H}_4$	105.1946	82.7086	56.0667	-0.4643	0.3491	0.434
$\mathbf{H}_5$	64.3447	46.3932	87.5277	-0.1078	0.1555	-0.3288
$\mathbf{H}_6$	74.1766	-84.2701	-80.3077	-0.4538	-0.4029	0.3235
$\mathbf{H}_7$	70.1383	-32.9221	162.0799	-0.4656	-0.0613	-0.1184
$\mathbf{H}_8$	95.586	53.136	-112.7259	-0.0102	-0.0544	0.1463
$\mathbf{H}_9$	75.3713	45.8436	-80.631	0.1797	0.1551	-0.3374
$\mathbf{H}_{10}$	-137.1608	-0.2945	165.5078	-0.1596	0.0853	-0.2762

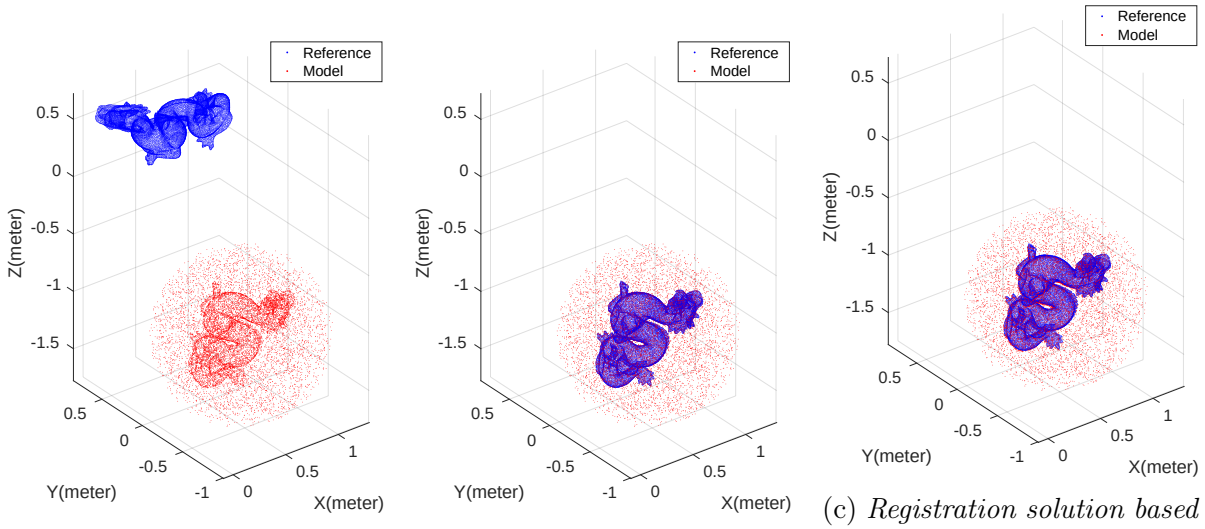
Table 5.3: Parameters used for the 10 rigid transformations from $\mathcal{M}$ to $\widehat{\mathcal{M}}$. The Table shows the Roll, α, pitch, β, and yaw, γ, angles as well as the x, y and z translations, t_x, t_y and t_z, used in each of the 10 transformation experiments.

Finally, the PSO-COLA algorithm utilizes two different approaches. In the first, the PSO-COLA with Unknown Correspondence (Unknown Correspondence (UC)) algorithm utilizes the Hungarian method to estimate the set of correspondences and the estimated transformations to register $\mathcal{M}$ and $\widehat{\mathcal{M}}$ jointly. In the second case, the PSO-COLA with Known Correspondence (KC) algorithm, utilizes the assumed correct set of correspondences between $\mathcal{M}$ and $\widehat{\mathcal{M}}$, avoiding the use of the Hungarian algorithm.



(a) *Unregistered scenario for the Bunny dataset.* (b) *Bunny registration using GT transformation* $\mathbf{H}_1$. (c) *Registration solution based on PSO-COLA for the Bunny dataset.*

Figure 5.9: *Registration-based on the Stanford Bunny dataset. The model set* $\widehat{\mathcal{M}}$ *is corrupted with Gaussian noise* $\mathbf{w}^{\mathrm{i}}$ *drawn from zero mean multi-variate Normal distribution* $\mathcal{N}(\mu = \mathbf{0}, \mathbf{\Sigma} = \mathrm{diag}(\sigma^{\mathbf{2}}))$ *with* $\sigma = 0.01[m]$. $p_{FA} = 0.25$ *and* $p_{MD} = 0.75$.



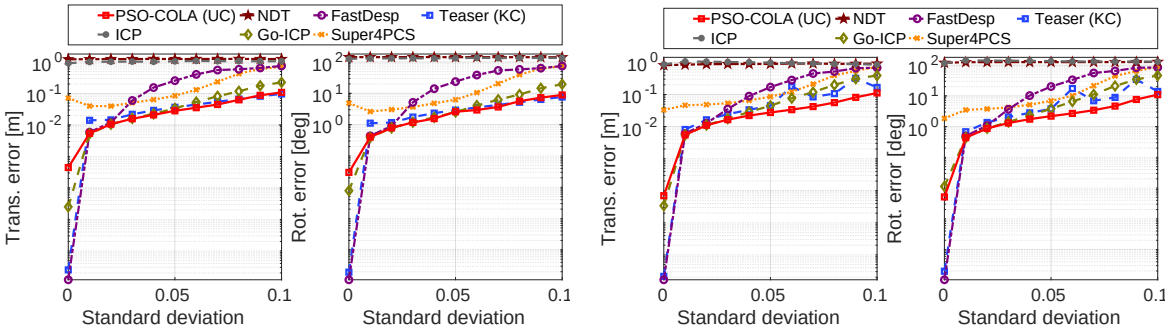
(a) *Unregistered scenario for the Dragon dataset.* (b) *Dragon registration using GT transformation* $\mathbf{H}_1$. (c) *Registration solution based on PSO-COLA for the Dragon dataset.*

Figure 5.10: *Registration-based on the Dragon dataset. The model set* $\widehat{\mathcal{M}}$ *is corrupted with Gaussian noise* $\mathbf{w}^{\mathrm{i}}$ *drawn from zero mean multi-variate Normal distribution* $\mathcal{N}(\mu = \mathbf{0}, \mathbf{\Sigma} = \mathrm{diag}(\sigma^{\mathbf{2}}))$ *with* $\sigma = 0.01[m]$. $p_{FA} = 0.25$ *and* $p_{MD} = 0.75$.

Registration Robustness in the Presence of Spatial Noise

To assess the robustness of each registration algorithm to only spatial noise within the model dataset, the spatial noise standard deviation is varied from $\sigma = 0[m]$ to $\sigma = 0.1[m]$, in the absence of miss-detections and false alarms.

This particular scenario is of interest, to determine the effects of the cut-off parameter c and the Hungarian assignment algorithm within the COLA metric component of the PSO-COLA algorithm. In contrast, most of the other registration algorithms simply apply the L_2 metric to determine transformed reference to model dataset correspondences. PSO-COLA with UC in Figure 5.11 (solid red curves) is able to obtain an accuracy comparable to TEASER KC (blue dashed curves). This demonstrates the PSO-COLA algorithm's robust-



(a) *Standard deviation experiment for the Bunny dataset.*

(b) *Standard deviation experiment for the Dragon dataset.*

Figure 5.11: *Monte Carlo Simulations for the Stanford Bunny and Dragon datasets showing the rotation and translation errors as a function of spatial standard deviation σ in the registration process, with respect to the ground truth transformations.*

ness to unknown correspondences between points in $\mathcal{M}$ and $\widehat{\mathcal{M}}$. Go-ICP (with 0% trimming factor) shows an increase in error when $\sigma = 0.06[m]$. This is due to the problem of Go-ICP eliminating correct matches, generating incorrect results. It is also interesting to note that the FastDesp registration algorithm demonstrates good performance when $\sigma < 0.02[m]$. This is because, with increased spatial noise, the probability of obtaining correct data correspondences reduces, as a consequence of FastDesp ignoring the joint compatibility of all the data. Super4PCS exhibits high errors for all values of σ. This shows the fragility of Super4PCS when spatial noise exists, since the number of correct congruent point assignments between $\mathcal{M}$ and $\widehat{\mathcal{M}}$ is reduced.

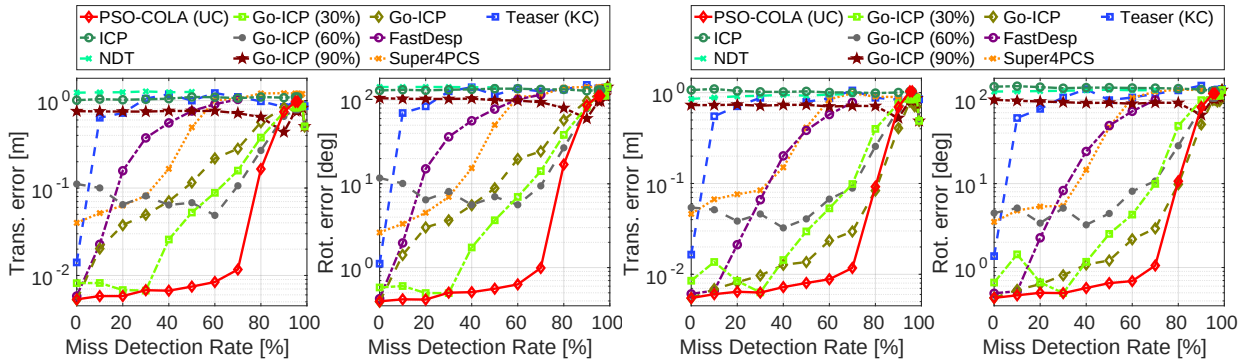
As expected, the local ICP and NDT registration algorithms exhibit the worst performances. This due do to the large rotational component within some of the transformations used in the Monte Carlo (MC) runs, as shown in Table 5.3.

Miss-detection experiments

This set of experiments evaluates the performance of each registration algorithm with increasing miss-detection rate. To simulate realistic scenarios, spatial Gaussian noise $\mathbf{w}^{\mathrm{i}}$ drawn

from a zero mean multivariate Normal distribution $\mathcal{N}(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \text{diag}(\sigma^2))$ with $\sigma = 0.01[m]$ is also added.

The results of the PSO-COLA algorithm (solid red curves) are shown in Figures 5.12a and 5.12b for the Stanford Bunny and Dragon datasets respectively for the case of miss-detections. It can be seen that the PSO-COLA algorithm outperforms all the state-of-the-art



(a) *Miss-detection experiment for the Bunny dataset.*

(b) *Miss-detection experiment for the Dragon dataset.*

Figure 5.12: *Monte Carlo runs for the Stanford Bunny and Dragon datasets showing the rotation and translation errors in the registration process as a function of the miss-detection rate, with respect to the ground truth transformations.*

algorithms considered, for values of probability of miss-detection, $p_{MD} \leq 0.8$ with respect to the reference set. Go-ICP shows different results depending on the trimming factor utilized and, as expected, every time the overlap between the datasets corresponds to the trimming factor, Go-ICP demonstrates its best performance, with Go-ICP (30%) showing the best results.

TEASER (blue dashed curves) requires some of the correct correspondences between points in $\mathcal{M}$ and $\widehat{\mathcal{M}}$. As the number of miss-detections is increased, the number of correct corresponding points which remain, is reduced. Figures 5.12a and 5.12b verify this since the translational and rotational errors corresponding to TEASER increase significantly for miss-detection rates beyond $p_{MD} = 0$.

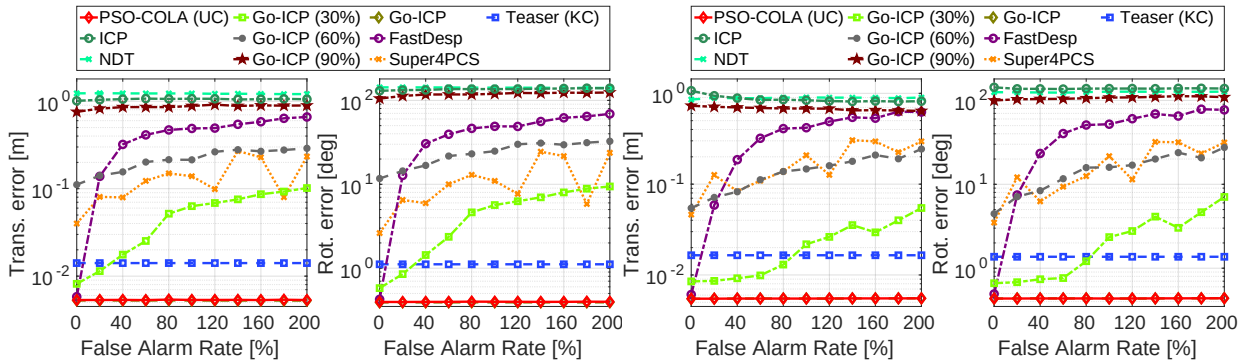
FastDesp (purple dashed curves) exhibits higher errors that both PSO-COLA and Go-ICP as expected. This is because as p_{MD} increases, there are less points in $\widehat{\mathcal{M}}$ to compute the necessary descriptors. Similarly, Super4PCS (yellow dotted curves) also shows lower performance because as p_{MD} increases, there are less points in $\widehat{\mathcal{M}}$ to compute the necessary congruent points.

The local registration algorithms ICP and NDT perform worst along with Go-ICP 90% (brown dashed curves). As expected, the local methods are sensitive to the large rotational component within some of the transformations used in the MC runs, as shown in Table 5.3. When $p_{MD} \geq 0.9$, all algorithms start to diverge. The slight reduction in error for Go-ICP 90% is expected, since the overlap between the datasets corresponds to the trimming factor.

False Alarm experiments

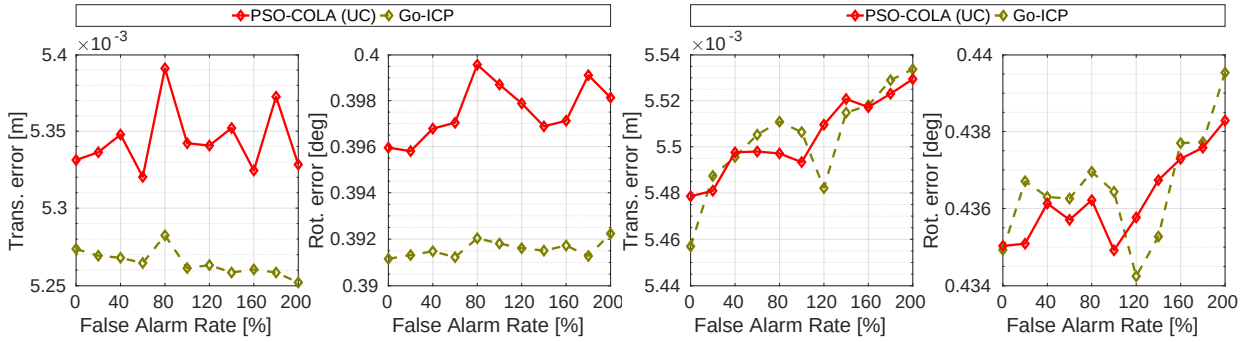
This set of experiments evaluates the performance of each registration algorithm with increasing false alarm rate. To simulate realistic scenarios, spatial Gaussian noise $\mathbf{w}^{\mathrm{i}}$ drawn from a zero mean multivariate Normal distribution $\mathcal{N}(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \mathrm{diag}(\sigma^2))$ with $\sigma = 0.01[m]$ is also added.

The PSO-COLA and Go-ICP registration algorithms provide the best, and similar, results when the probability of false alarms, p_{FA}, increases, as can be seen in Figures 5.13a and 5.13b. Zoomed views of the errors are shown in Figures 5.13c and 5.13d where it can be



(a) *False Alarms experiment for the Bunny dataset.*

(b) *False Alarms experiments for the Dragon dataset.*



(c) *Zoomed False Alarms experiment for the Bunny dataset.*

(d) *Zoomed False Alarms experiments for the Dragon dataset.*

Figure 5.13: *Monte Carlo runs for the Stanford Bunny and Dragon datasets showing the rotation and translation errors in the registration process, as a function of false alarm rate, with respect to the ground truth transformations.*

seen that the differences between them is of the order of fractions of millimeters. Notice that in Figure 5.13c, the Go-ICP algorithm narrowly outperforms PSO-COLA, whereas, in Figure 5.13d both Go-ICP and PSO-COLA show similar behavior. This could be because the BnB optimization method used in the Go-ICP registration algorithm is more robust to stagnation than PSO under certain point cloud geometries such as symmetries within the data. The Bunny dataset has large rotational symmetry within the body section of the data, which can cause the PSO-COLA registration algorithm to stagnate, a problem which can potentially be overcome with the Go-ICP algorithm, due to its BnB component.

It is expected that the PSO-COLA registration algorithm would perform better with the

Dragon dataset since it contains less symmetries than the Bunny dataset, making stagnation less likely with PSO-COLA, such that it performs equally as well as Go-ICP. This is evident in Figures 5.13b and 5.13d.

We note in Figures 5.13a and 5.13b that Go-ICP (with 0% trimming factor) provides better results than Go-ICP with 30, 60 and 90% trimming factor, because all points from the reference set are present in the model set (no miss-detections). When using trimming factors, Go-ICP could incorrectly eliminate correct matches, generating higher errors. Therefore, as expected, Go-ICP with trimming factor 90% gives the worst performance in this case.

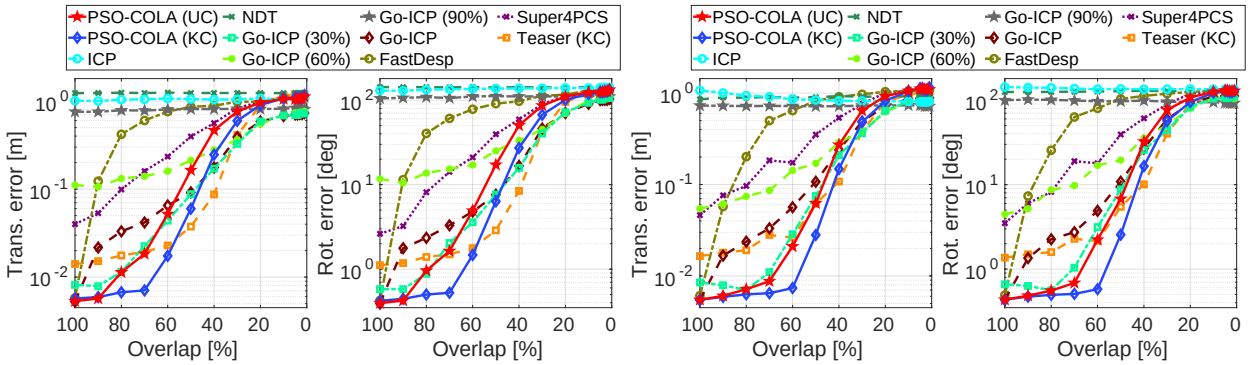
In this case, TEASER is provided with the correct correspondences between $\mathcal{M}$ and $\widehat{\mathcal{M}}$. As expected, it yields low translational and rotational errors, however, it fails to outperform PSO-COLA. This is because of the spatial noise added to $\widehat{\mathcal{M}}$, showing that TEASER is more sensitive to spatial noise than false alarms when given the correspondences.

It is also interesting to note that Super4PCS has a similar performance to Go-ICP with trimming factor 60%, while FastDesp shows large errors for $p_{FA} \geq 0.5$.

The local registration algorithms ICP and NDT exhibit the worst performance. Again, this is due do to the large rotational component within some of the transformations used in the MC runs, as shown in Table 5.3.

Simulated overlapping experiments

Overlapping experiments, which correspond to the case in which a percentage p_o of points in the model are replaced by random points within a sphere of radius $r = 2[m]$, are shown in Figures 5.14a and 5.14b. It can be seen that if p_o increases from 0% to 100%, PSO-COLA with



(a) *Overlapping experiment for the Bunny dataset.*

(b) *Overlapping experiment for the Dragon dataset.*

Figure 5.14: *Monte Carlo runs for the Stanford Bunny and Dragon datasets showing the rotation and translation errors in the registration process, as a function of overlap with respect to the ground truth transformations. The x axis is represented by the overlap defined as Overlap $= 100 - p_o$.*

known correspondence (PSO-COLA (KC) solid blue curves) has the best performance for the Bunny when the overlap $100 - p_o > 60\%$ and for the Dragon when $100 - p_o > 70\%$. At lower percentages however, PSO-COLA (KC) loses its accuracy due to particle stagnation at local

minima. The reason for this stagnation is related to the loss of accuracy in the initialization proposed in our method. Interestingly, if the correspondences between points in $\mathcal{M}$ and $\widehat{\mathcal{M}}$ are unknown, PSO-COLA (UC) (solid red curves) is able to obtain an accuracy comparable to TEASER with known correspondences (TEASER (KC) orange dashed curves). This demonstrates the PSO-COLA algorithm's robustness to unknown correspondences between points in $\mathcal{M}$ and $\widehat{\mathcal{M}}$.

Notice when the Overlap $< 70\%$ for the Stanford Bunny dataset and Overlap $< 60\%$ for the Dragon dataset, the Go-ICP algorithm with 0%, 30%, 60% and 90% trimming factor, and TEASER perform better than PSO-COLA. This is due to the PSO method being susceptible to local minima. However, due to robust initialization given in Section 4.3, the solution given by PSO-COLA is still close to the global minima, providing acceptable registration results.

Note again the sensitivity of the Go-ICP algorithm with respect to the trimming factors. The merit of the PSO-COLA algorithm shown in this experiment is that unlike Go-ICP, it does not require a correspondence related trimming factor. PSO-COLA automatically eliminates incorrect correspondences by using the cut-off parameter c used in its COLA metric.

Note that the FastDesp and Super4PCS algorithms yield an inferior performance compared with the other global registration methods.

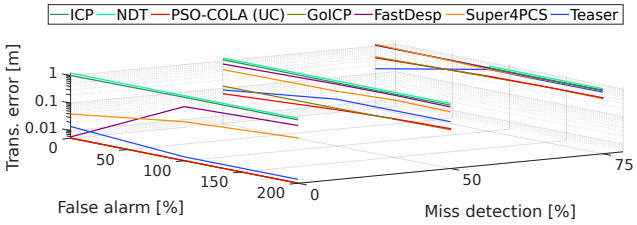
Again, the local registration algorithms ICP and NDT exhibit the worst performance. This due to the large rotational component within some of the transformations used in the MC runs, as shown in Table 5.3.

Combined Miss-detection and False Alarm experiments

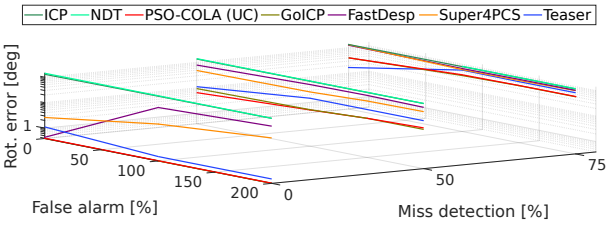
In this case, the construction of each dataset contains both false alarms and miss-detections, and again the model dataset is corrupted with Gaussian noise $\mathbf{w}^{\mathrm{i}}$ drawn from a zero mean multivariate Normal distribution $\mathcal{N}(\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \mathrm{diag}(\sigma^2))$ with $\sigma = 0.01[m]$. Interestingly, in the absence of miss-detections, Go-ICP and PSO-COLA, provide the best performances as can be seen in Figure 5.15. This scenario is drastically changed when miss-detections appear. In particular, when $p_{MD} = 0.5$, it is clear that PSO-COLA outperforms the Go-ICP algorithm. However, when $p_{MD} = 0.75$ we observe errors of the order of 0.5[m], with both PSO-COLA and Go-ICP. The reason for this behavior is the failures in the correct assignment between points within $\mathcal{M}$ and $\widehat{\mathcal{M}}$, when both miss-detections and false alarms increase.

Note that the FastDesp and Super4PCS algorithms still yield inferior performances compared with other global registration algorithms. Both methods are very sensitive to miss-detections. Additionally, because this experiment also contains false alarms, it is more difficult for these algorithms perform robust registration.

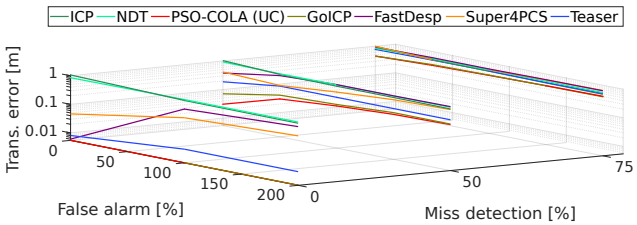
Again, the local registration algorithms ICP and NDT exhibit the worst performance. This due to the large rotational component within some of the transformations used in the MC runs, as shown in Table 5.3.



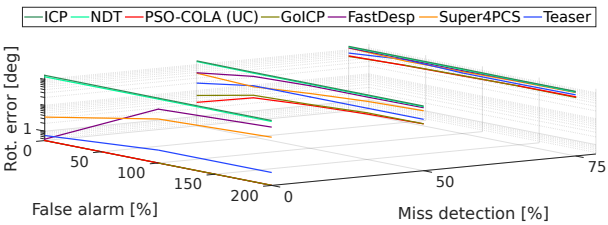
(a) *Translation error versus p_{MD} & p_{FA} experiment for the Bunny dataset.*



(b) *Rotation error versus p_{MD} & p_{FA} experiments for the Bunny dataset.*



(c) *Translation error versus p_{MD} & p_{FA} experiment for the Dragon dataset.*



(d) *Rotation error versus p_{MD} & p_{FA} for the Dragon dataset.*

Figure 5.15: *Monte Carlo runs for the Stanford Bunny and Dragon datasets showing the rotation and translation errors in the registration process, with respect to the ground truth transformations.*

5.3.2 The "ETH Apartment" Dataset

In this section, the large-scale point cloud datasets corresponding to the "ETH Apartment" from [27] are used to compare the scan matching algorithms. The entire dataset contains 44 3D scans recorded at different poses inside the apartment. Figure 5.16 shows all 44 scans superimposed, displayed relative to the ground truth locations at which they were recorded. These are shown as black dots. The yellow line shows the ground-truth trajectory of the laser range finder. Ground truth in this case was determined by manually matching all 44 scans using the ICP registration algorithm. The ground truth locations are useful, since they allow the transformation errors to be calculated when comparing different registration algorithms.

At consecutive time stamps, the laser range finder records two point clouds defined as $\mathcal{M}(\tau-1)$ and $\mathcal{M}(\tau)$, with $\tau = [1,\ldots,44]$. Then, FPFH feature descriptors [79] are computed for $\mathcal{M}(\tau-1)$ and $\mathcal{M}(\tau)$ to establish an estimated set of correspondences between them, as explained in Section 2.6.3. This estimated set of correspondences may contain incorrect correspondences and the aim of this experiment is to test the robustness of each registration algorithm accordingly. The advantage of the FPFH algorithm is that it reduces the computational complexity of PSO-COLA drastically, as explained in Section 4.4.3.

Figures 5.17a and 5.17b summarise the errors committed by each registration algorithm. The height of each stacked bar, represents the translation and rotation error of each registration algorithm at each time stamp τ. In particular, these results show high variability in the errors for the local registration algorithms ICP and NDT. On the other hand, the global registration algorithms such as PSO-COLA, TEASER, Go-ICP and SuperPCS exhibit signif-

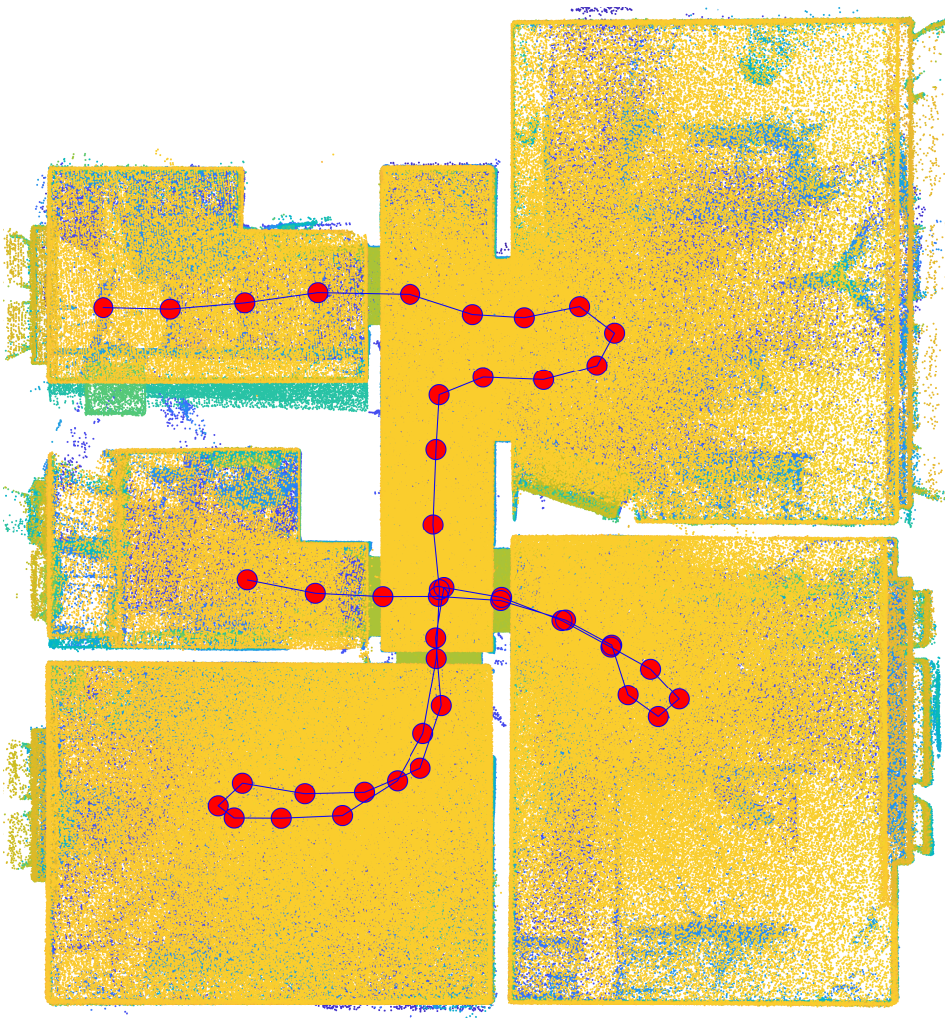
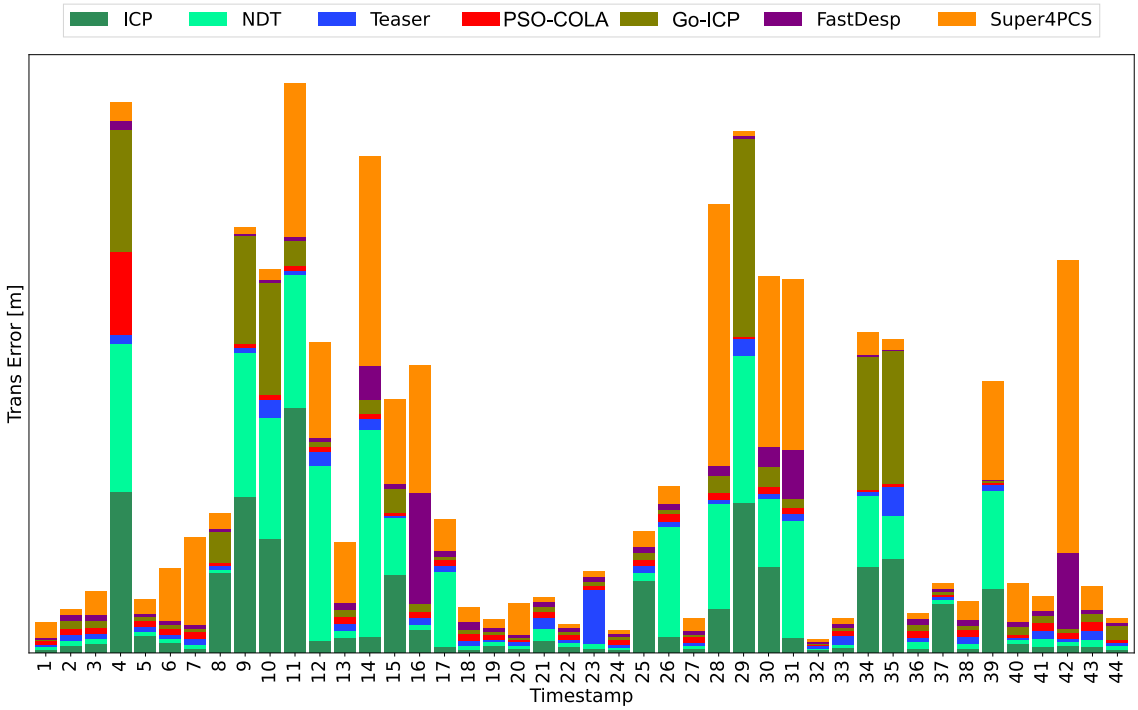
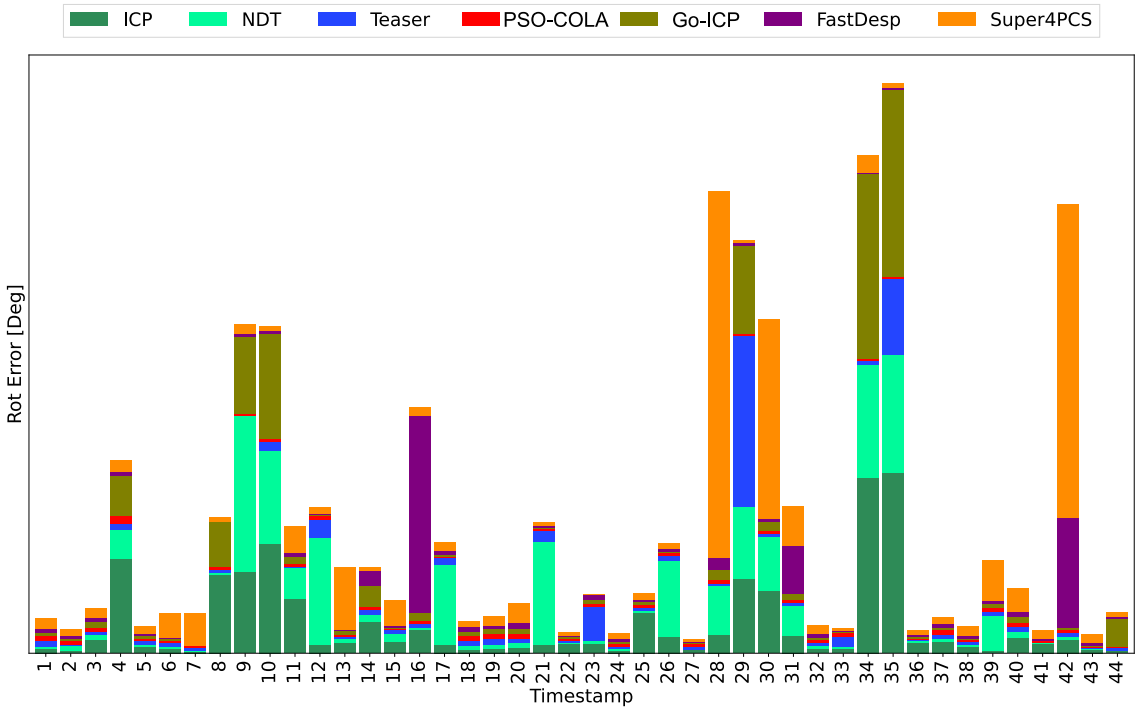


Figure 5.16: *The ETH Apartment dataset, containing 44 3D laser range scans.*



(a) *The translational error resulting from each registration algorithm in meters.*

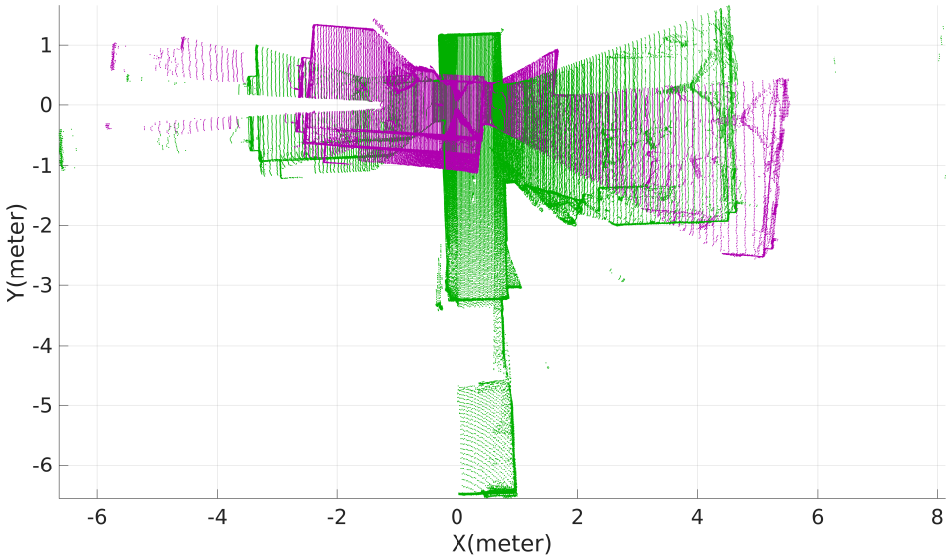


(b) *The rotational error resulting from each registration algorithm in degrees.*

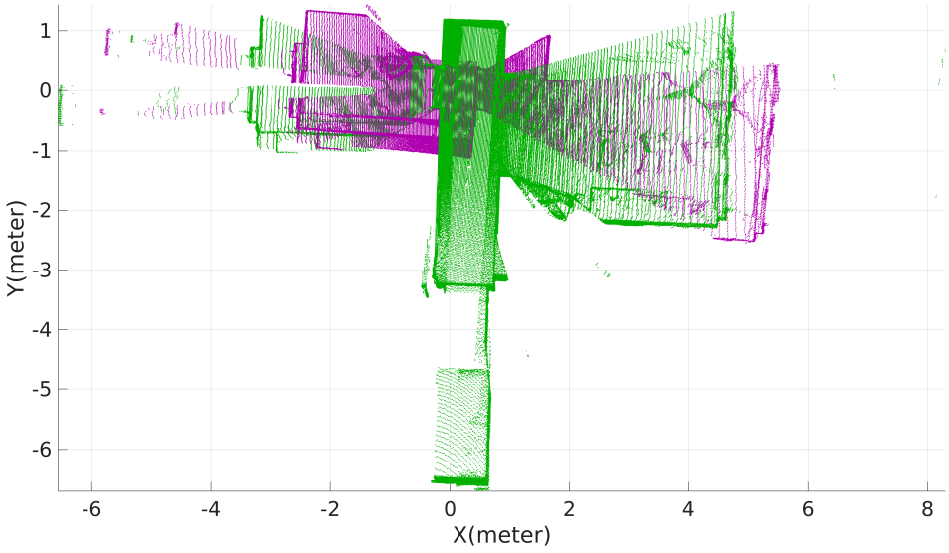
Figure 5.17: *Error bar plots resulting from each registration algorithm.*

cantly lower errors. The height, and therefore error, corresponding to PSO-COLA is lowest in most cases.

Note that in Figure 5.17a, PSO-COLA exhibits a large translation error at time stamp $\tau = 4$, corresponding to the registration of the scans at times $\mathcal{M}(\tau = 4)$ and $\mathcal{M}(\tau = 3)$. This is due to the PSO algorithm's susceptibility to stagnate to an incorrect minimum. Figure 5.18 shows the registration results with the purple scan representing the scan at time $\tau = 3$ and the green scan showing the scan at time $\tau = 4$. Figure 5.18a shows the unregistered scans and Figure 5.18b shows the registered version of the scans. Note that the scan at time $\tau = 4$



(a) *Unregistered datasets* $\mathcal{M}(\tau = 3)$ *and* $\mathcal{M}(\tau = 4)$.



(b) *PSO-COLA Registration at time stamp* $\tau = 4$.

Figure 5.18: *Plan view of the unregistered and registered scans at time stamps* $\tau = 4$ *(green) and* $\tau = 3$ *(purple).*

(green) is slightly rotated, yielding a good registration with respect to rotation, as verified at time $\tau = 4$ in Figure 5.17b. The translational registration shown in Figure 5.18b has a relatively large error however, as verified in Figure 5.17a. Close examination of Figure 5.18b

reveals that the estimated transform has produced a better overlap in parts of the environment, for example the left hand lower wall in the x direction, but has failed to yield the displacement necessary to align the end walls in the y direction.

Figure 5.19 presents the box plot statistics corresponding to each registration algorithm. The size of each box plot is defined as the distance between the quartile $Q_3(75\%)$ and

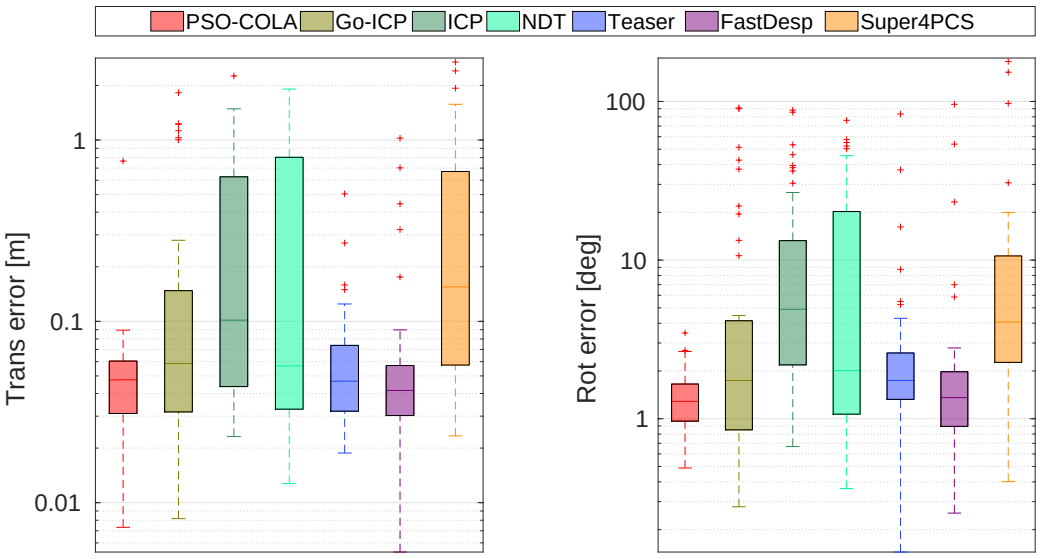


Figure 5.19: *Box-Plot of the translational and rotational error components.*

$Q_1(25\%)$, referred to as the interquartile range

$$IQR = Q_3(75\%) - Q_1(25\%).$$

The median, $Q_2(50\%)$ is represented by a line segment within each box plot. The upper and lower whiskers (U_w and L_w respectively) of each box plot are given by the equations:

$$U_w = Q_3(75\%) + 1.5IQR \tag{5.2}$$
$$L_w = Q_1(25\%) - 1.5IQR. \tag{5.3}$$

We consider a registration error ("trans." or "rot." respectively) to be "large" if either the translational or rotational errors are above the upper whisker U_w as shown in Figure 5.19.

Note that, as seen in Figure 5.19, although FastDesp and TEASER have similar error statistics to PSO-COLA, their registration solutions yield greater numbers of outliers. This is due to their poor performance in evaluating the correct registrations, given incorrect correspondence estimates by the FPFH algorithm. In contrast, the PSO-COLA registration algorithm is able to cope with such incorrect correspondences due to its ability to quantify them as detection errors, due to its cut-off parameter c.

Tables 5.4 and 5.5, summarize the statistics of each box-plot, where the highlighted blue and red numbers correspond to the best and worst performances for each statistical error

value (median and mean errors, standard deviation, minimum and maximum errors), for each registration algorithm respectively. From Table 5.4, it can be seen that the number

Statistics	**ICP**	**NDT**	**TEASER**	**PSO-COLA**	**GoICP**	**FastDesp**	**Super4PCS**
Median error	0.102	0.057	0.047	0.048	0.059	0.042	0.155
Mean error	0.362	0.409	0.070	0.062	0.230	0.096	0.477
Std. Dev. of error	0.508	0.548	0.081	0.110	0.423	0.189	0.667
Minimum error	0.023	0.013	0.019	0.007	0.008	0.005	0.023
Maximum error	2.257	1.910	0.505	0.766	1.823	1.023	2.691
Time stamp of large trans. errors	4-11	14	10-23-29-35	4	4-9-10-29-34-35	14-16-30-31-42	14-28-30-31-42
Number of large trans. errors	2	1	4	1	6	5	5

Table 5.4: Translational error statistics for the ETH Apartment in meters.

Statistics	**ICP**	**NDT**	**TEASER**	**PSO-COLA**	**GoICP**	**FastDesp**	**Super4PCS**
Median error	4.90	2.01	1.74	1.29	1.74	1.36	4.07
Mean error	14.95	15.63	5.64	1.33	11.04	6.12	16.93
Std. Dev. of error	21.73	21.46	14.57	0.57	22.75	17.80	39.70
Minimum error	0.89	0.36	0.14	0.49	0.35	0.27	0.40
Maximum error	88.12	76.05	83.28	2.70	90.96	95.96	178.75
Time stamp of large rot. errors	4-8-9-10-29-30-34-35	9-10-12-21-34-35	12-21-23-29-33-35	4-18	4-8-9-10-14-29-34-35-44	14-16-28-31-42	13-28-30-42
Number of large rot. errors	8	6	6	2	9	5	4

Table 5.5: Rotational error statistics for the ETH Apartment in degrees.

of large translational errors is low in comparison with those of the other global registration algorithms. Further, in Table 5.5 it can be seen that the number of large rotational errors is low in comparison with those of the other global registration algorithms and significantly lower than those committed by the local registration algorithms. In general, in both tables, it is evident that the PSO-COLA registration algorithm outperforms the other algorithms in most cases, yielding a small number of large registration errors and high accuracy.

Despite Super4PCS being a global registration algorithm it yielded the worst results (red numbers in the tables) for most of the statistical values. The reason for this is mentioned in [72], where the authors state that Super4PCS is a coarse refinement solution, and that it requires a refinement methodology such as trimming-ICP to cope with point cloud cardinality differences.

We also compared the PSO-COLA registration algorithm with the TEASER and Go-ICP registration algorithms, due to their high performances demonstrated in Section 5.3. These comparisons will be demonstrated with the scans recorded at time stamps $\tau = 28$ and $\tau = 29$, corresponding to Figure 5.20a, and $\tau = 34$ and $\tau = 35$, corresponding to Figure 5.21a.

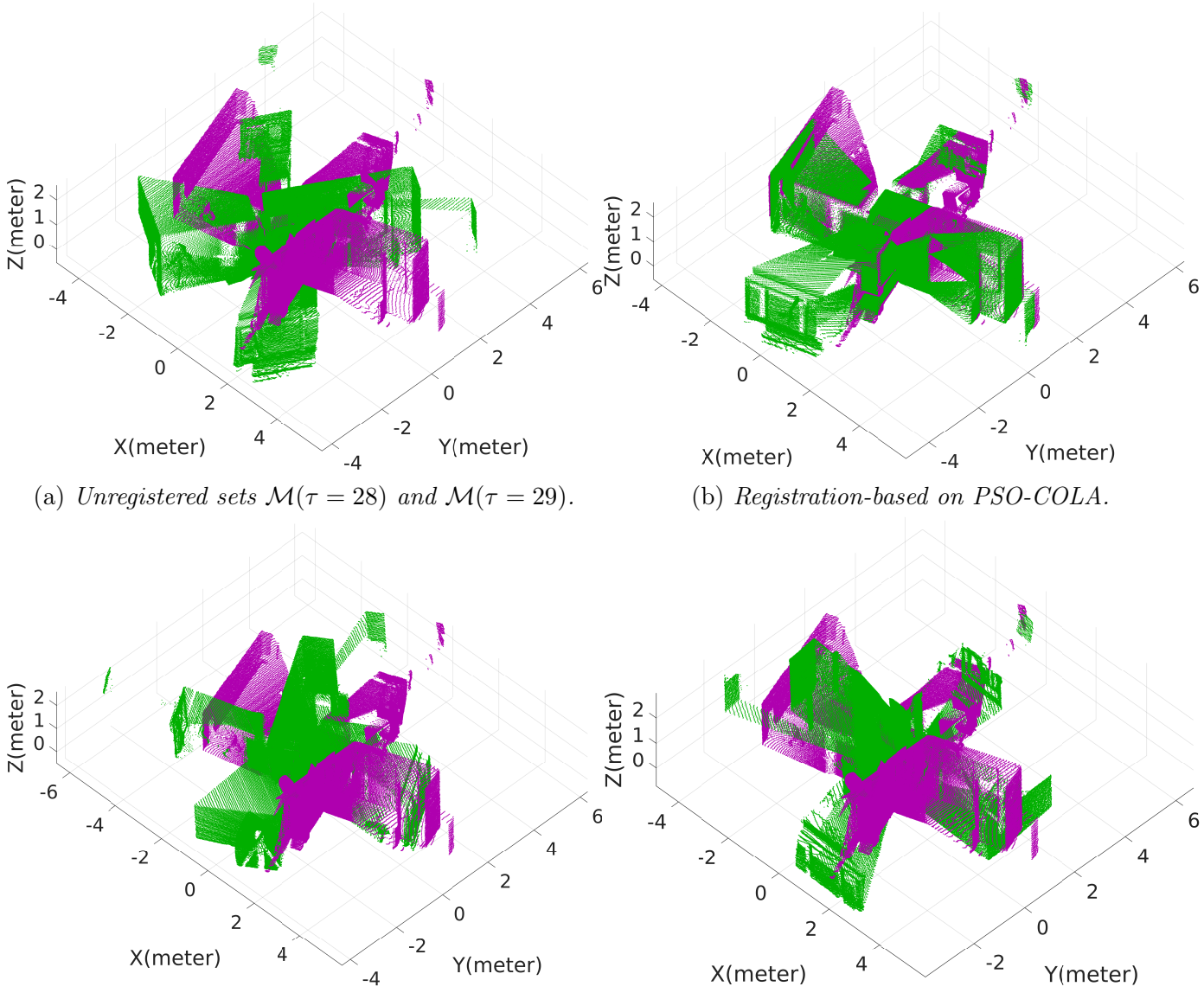
For the case of the scans with time stamps $\tau = 28$ and $\tau = 29$ (Figure 5.20a), it can be seen that $\mathcal{M}(\tau = 28)$ and $\mathcal{M}(\tau = 29)$, have approximate angular symmetries of 90 degrees in the $x - y$ plane, meaning that if either of them are rotated by 90 degrees, there are significant similarities. Similarly, for the case of the scans with time stamps $\tau = 34$ and $\tau = 35$ (Figure 5.21a), it can be seen that $\mathcal{M}(\tau = 34)$ and $\mathcal{M}(\tau = 35)$, have approximate angular symmetries of 180 degrees in the $x - y$ plane, meaning that if either of them are

rotated by 180 degrees, there are significant similarities. This is because the width and length of the room shown in Figure 5.21a are slightly different.

The registration results for $\mathcal{M}(\tau = 28)$ and $\mathcal{M}(\tau = 29)$ based on PSO-COLA, TEASER and Go-ICP are shown in Figures 5.20b, 5.20c and 5.20d respectively, with their plan views shown in Figures 5.22a, 5.22b, 5.22c respectively. Note that despite the approximate 90 degree symmetry between these data sets, the PSO-COLA registration algorithm yields a correct registration as shown in Figures 5.20b and 5.22a. TEASER and Go-ICP however yield transformation estimates which are incorrect in their rotational components by approximately 90 degrees, showing their fragility in the presence of data symmetries. This is evident in Figures 5.20c and 5.22b and Figures 5.20d and 5.22c

The registration results for $\mathcal{M}(\tau = 34)$ and $\mathcal{M}(\tau = 35)$ based on PSO-COLA, TEASER and Go-ICP are shown in Figures 5.21b, 5.21c and 5.21d respectively with their plan views shown in Figures 5.23a, 5.23b, 5.23c respectively. Similarly, note that despite the approximate 180 degree symmetry between these data sets, the PSO-COLA registration algorithm yields a correct registration as shown in Figures 5.21b and 5.23a. Again, TEASER and Go-ICP however yield transformation estimates which are incorrect in their rotational components by approximately 180 degrees, again showing their fragility in the presence of data symmetries. This is evident in Figures 5.21c and 5.23b and Figures 5.21d and 5.23c.

It should also be noted that the overlap between both $\mathcal{M}(\tau = 28)$ and $\mathcal{M}(\tau = 29)$ and $\mathcal{M}(\tau = 34)$ and $\mathcal{M}(\tau = 35)$ is unknown. Therefore, in the case of the Go-ICP registration algorithm, it is difficult to choose a correct trimming factor, which despite the data symmetries could help improve its performance.



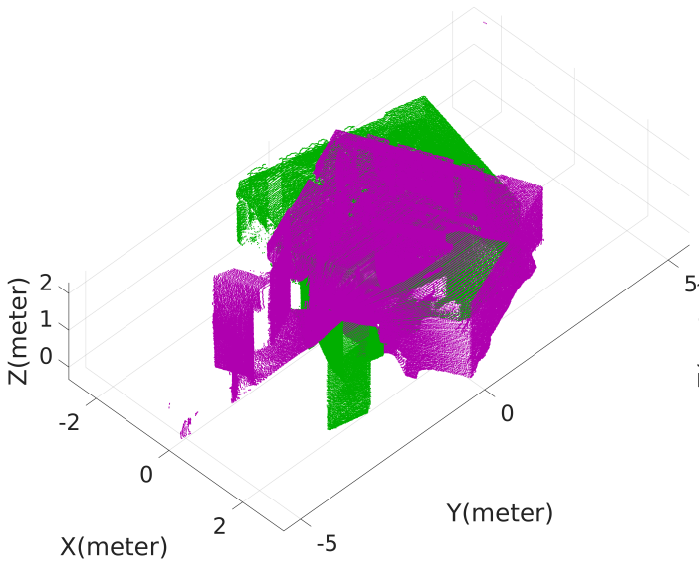
(a) *Unregistered sets* $\mathcal{M}(\tau = 28)$ *and* $\mathcal{M}(\tau = 29)$.

(b) *Registration-based on PSO-COLA.*

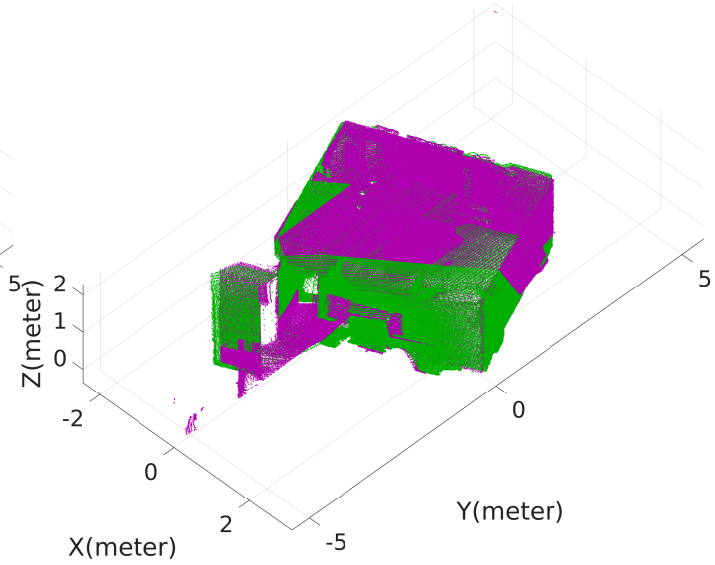
(c) *Registration-based on TEASER.*

(d) *Registration-based on Go-ICP.*

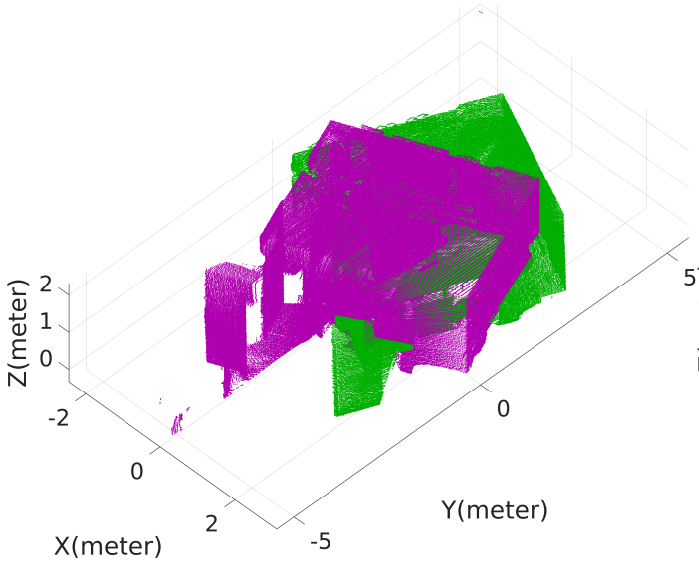
Figure 5.20: *Registration solutions at* $\tau = 29$. *Green (model set) and purple (reference) point clouds are represented by the set* $\mathcal{M}(\tau = 28)$ *and* $\mathcal{M}(\tau = 29)$ *respectively.*



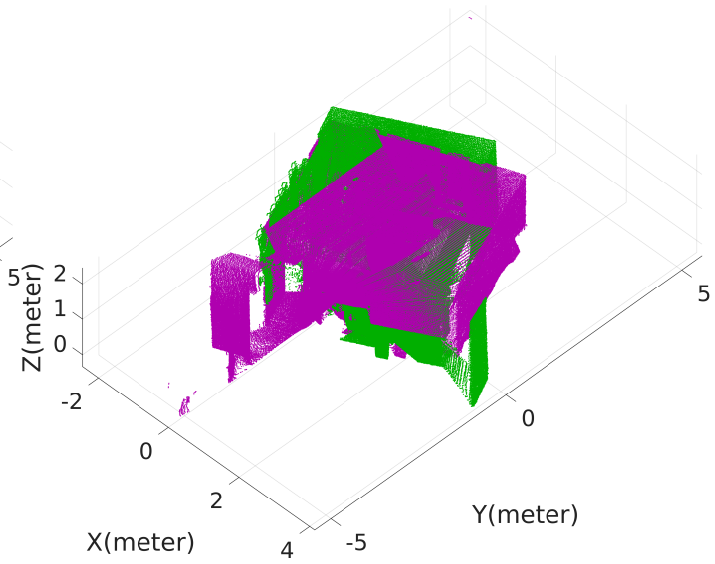
(a) *Unregistered sets* $\mathcal{M}(\tau = 34)$ *and* $\mathcal{M}(\tau = 35)$.



(b) *Registration-based on PSO-COLA.*

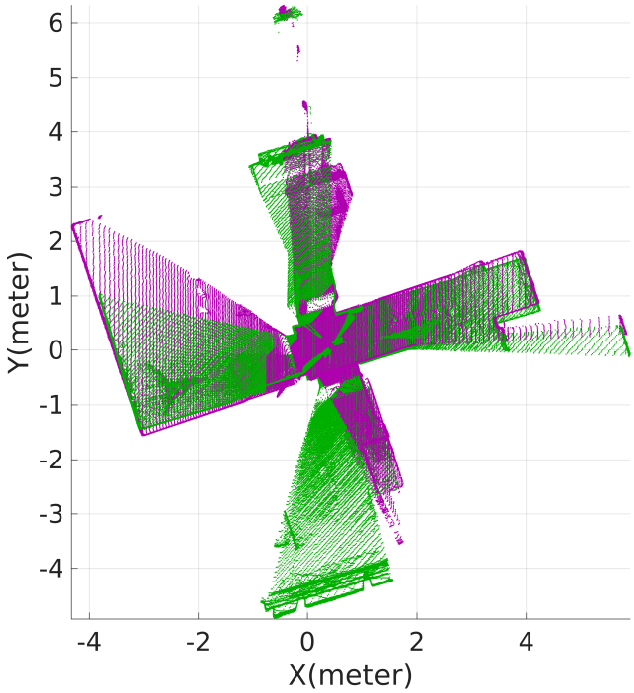


(c) *Registration-based on TEASER.*

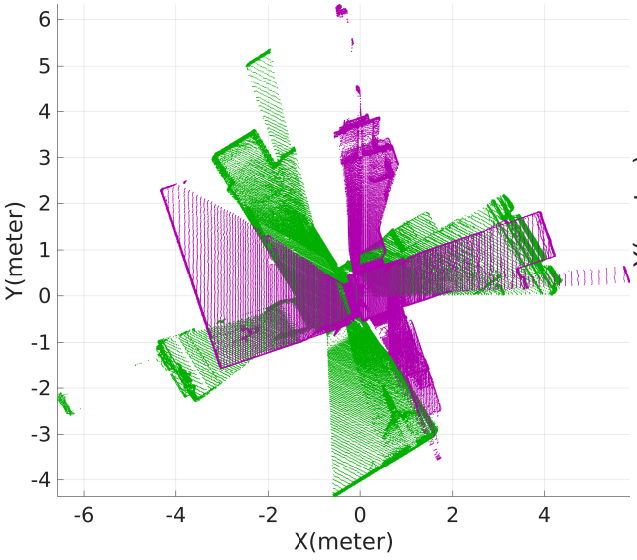


(d) *Registration-based on Go-ICP.*

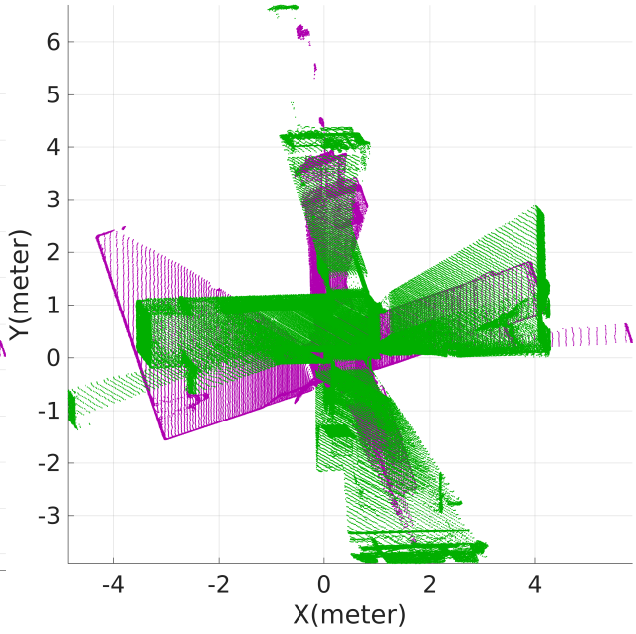
Figure 5.21: *Registration solutions at* $\tau = 35$. *Green (model set) and purple (reference) point clouds are represented by the set* $\mathcal{M}(\tau = 34)$ *and* $\mathcal{M}(\tau = 35)$ *respectively.*



(a) *Registration-based on PSO-COLA.*

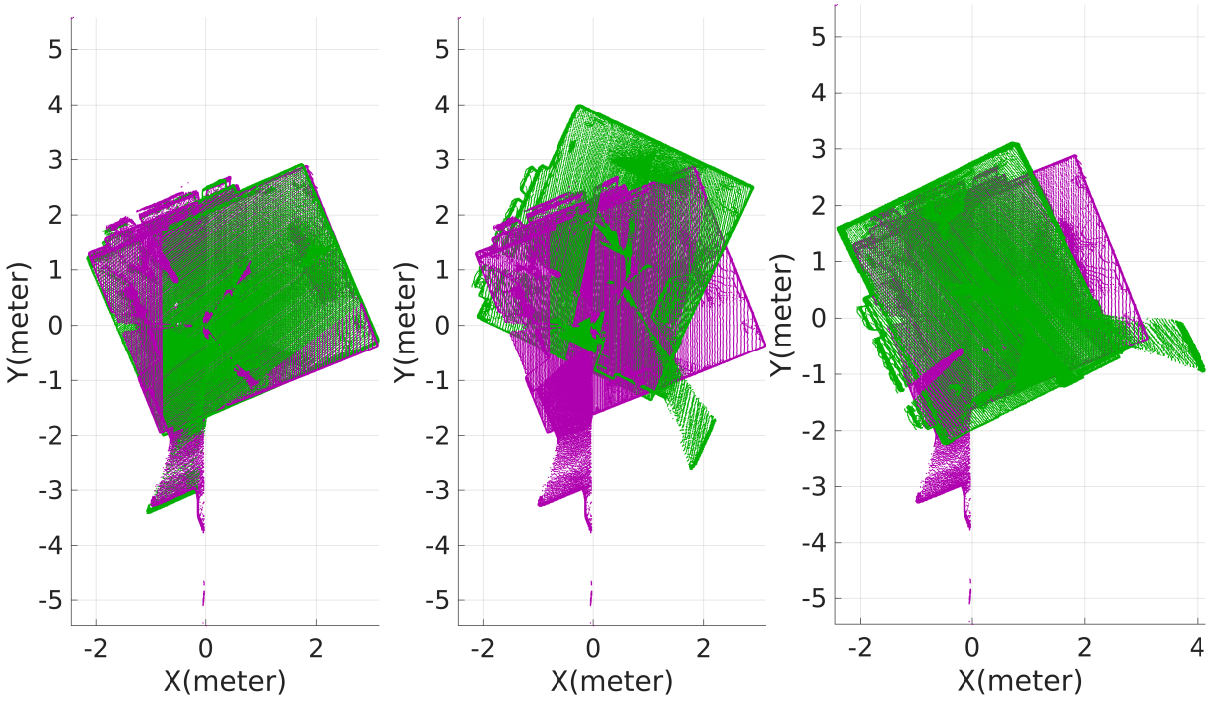


(b) *Registration-based on TEASER.*



(c) *Registration-based on Go-ICP.*

Figure 5.22: *Plan view of the registration solutions at* $\tau = 29$. *Green (model set) and purple (reference) point clouds are represented by the set* $\mathcal{M}(\tau = 28)$ *and* $\mathcal{M}(\tau = 29)$ *respectively.*



(a) *PSO-COLA Registration.* (b) *TEASER Registration.* (c) *Go-ICP Registration.*

Figure 5.23: *Plan view of the registration solutions at* $\tau = 35$. *Green (model set) and purple (reference) point clouds are represented by the set* $\mathcal{M}(\tau = 34)$ *and* $\mathcal{M}(\tau = 35)$ *respectively.*

Chapter 6

Conclusions and Future Work

6.1 Conclusions I: Multi-object metrics for evaluating feature-based robotic mapping errors

Metrics which gauge the error in the SLAM trajectory, do not necessarily provide intuitive assessments of the full SLAM error. It was demonstrated that the multi-object map metrics, which provide intuitive assessments of mapping errors (Table 5.2), can significantly disagree with the trajectory evaluation metrics. In the case of robotic maps, single object metrics, such as the Euclidean distance, RMS, or NEES, can only gauge the error between maps, or subsets of maps, with the same cardinality. When feature cardinalities differ, multi-object metrics offer a solution. The OSPA metric is an averaged distance metric. When assessing mapping performance during SLAM execution, as more of the GT map passes through the field(s) of view of the sensor(s), this metric provides the most intuitive evaluation of the time varying estimated map (see Section 3.7.3). When comparing multiple estimated maps against a fixed GT map, the OSPA metric has some disadvantages, such as its saturation to parameter c when no features are gated, irrespective of the cardinality errors, and its higher penalization of missed detections over false alarms. On the other hand, when comparing multiple estimated maps with a fixed GT map, the COLA metric solves some of the problems associated with the OSPA metric. In particular, it is able to distinguish between map qualities, even if no features are gated (it does not saturate), and it equally penalizes missed detections and false alarms. Finally, since the OSPA metric often yields values close to c, small changes in c can result in the metric changing its decision when comparing different maps to GT. This problem is reduced with the COLA metric.

6.2 Conclusions II: Point set feature-based on multi-object metrics

The PSO-COLA algorithm was presented as a robust solution to the point cloud registration problem. It performs an initialization based on distance pattern matching and an opti-

mization based on PSO using the multi-object COLA metric as its objective function. In contrast to most state-of-the-art metrics, the COLA metric penalizes datasets in terms of their cardinality and spatial differences and within the PSO-COLA framework provides the core component for point cloud registration, taking into account both detection and spatial errors.

For testing the performance of the PSO-COLA algorithm, the publicly available Stanford Bunny and Dragon datasets were used. State-of-the-art algorithms such as Go-ICP, TEASER among others were compared. The results show that the PSO-COLA algorithm outperforms state-of-the-art methods in cases were cardinality and spatial errors exist, demonstrating its robustness in such realistic situations. This is due to the ability of the COLA metric to penalize cardinality as well as spatial errors.

The PSO-COLA algorithm increases its computational complexity when the point cloud dataset becomes large. This problem is inherited, from the COLA metric, which in turn inherits the problem from the Hungarian method. One way of reducing this complexity is to apply the PSO-COLA algorithm to FPFH descriptors instead of the point cloud data directly. Therefore, when comparing its performance with other methods using the large "ETH Apartment" dataset, the registration of FPFH descriptors within the PSO-COLA algorithm yielded a computationally tractable and robust solution. Relatively low translation and rotation errors resulted, even in the presence of significant point cloud data symmetries.

6.3 Future Work

Firstly, due to the computational complexity of the OSPA and COLA metrics ($\mathcal{O}(|\mathcal{M}|, |\widehat{\mathcal{M}}|)^{\ni}$), future work could develop a new probabilistic multi-object metric which avoids the Hungarian method. This problem could be addressed by using metrics with power $q \geq 1$ such that:

$$\mathrm{d}(\mathcal{M}, \widehat{\mathcal{M}}) = \left(\sum_{\mathrm{i}=1}^{\mathfrak{m}} \sum_{j=1}^{\widehat{\mathfrak{m}}} p_{\mathrm{i},j} \mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j})^{q} \right)^{1/q}, \tag{6.1}$$

with cut-off value c and inner truncated metric $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j})$, equivalent to the inner metrics of the OSPA and COLA metrics. $p_{\mathrm{i},j}$ is the probability of assigning points from $\mathbf{m}^{\mathrm{i}} \in \mathcal{M}$ to $\widehat{\mathbf{m}}^{j} \in \widehat{\mathcal{M}}$ according to a Maxwell-Boltzmann distribution. In this case

$$p_{\mathrm{i},j} = \frac{\exp(-\beta \mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j}))}{\sum_{\mathrm{i},j} \exp(-\beta \mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j}))}, \tag{6.2}$$

where β is a scale factor. In physics, the value β is related to the inverse of the temperature of a system of particles.

Equation (6.2) suggests the following scenarios: Consider two points $\mathbf{m}^{\mathrm{i}} \in \mathcal{M}$ and $\widehat{\mathbf{m}}^{j} \in \widehat{\mathcal{M}}$ such that $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j})) = 0$. $p_{\mathrm{i},j}$ is then the highest probability of associating $\mathbf{m}^{\mathrm{i}} \in \mathcal{M}$ and $\widehat{\mathbf{m}}^{j} \in \widehat{\mathcal{M}}$. Otherwise, if $\mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j})) = c$, $p_{\mathrm{i},j}$ corresponds to the lowest probability of associating $\mathbf{m}^{\mathrm{i}} \in \mathcal{M}$ and $\widehat{\mathbf{m}}^{j} \in \widehat{\mathcal{M}}$.

Note that in equation (6.2), the factor in the denominator

$$\sum_{\mathrm{i},j} \exp(-\beta \mathrm{d}^{(c)}(\mathbf{m}^{\mathrm{i}}, \widehat{\mathbf{m}}^{j})),$$

is constant and corresponds to the normalization factor of the Maxwell-Boltzmann distribution, and can be computed once with a quadratic computational complexity. Therefore, probabilistic multi-object metrics could be applied, resulting in a significantly reduced computational complexity compared with the OSPA and COLA metrics.

A second proposal for future work is to improve the registration algorithm presented in this thesis. The PSO technique in general suffers stagnation at local minima. Hence, a new registration solution based in multi-object metrics could be implemented using a BnB technique in a manner similar to the concept of Go-ICP.

Finally, it should be noted that the initialization method proposed for PSO-COLA only takes into account distance measures in its inner metric in equation (4.20), ignoring rotational errors. This can cause incorrect assignments for σ in equation (4.19) and an avenue for future research would be to derive an initialization inner metric capable of considering rotational measures also.

Bibliography

[1] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Machine intelligence and pattern recognition*, volume 5, pages 435 – 461. Elsevier, 1988.

[2] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241, June 2001.

[3] J. Nieto, J. Guivant, E. Nebot, and S. Thrun. Real time data association for FastSLAM. In *International Conference on Robotics and Automation (ICRA)*, volume 1, pages 412–418, September 2003.

[4] J. Folkesson and H. Christensen. Graphical SLAM - A self-correcting map. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 383–390, New Orleans, USA, April 2004.

[5] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, December 2008.

[6] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2011.

[7] J. Mullane, B.N. Vo, M. Adams, and B.T. Vo. A random-finite-set approach to Bayesian SLAM. *IEEE Transactions on Robotics*, 27(2):268–282, April 2011.

[8] C. Lee, D. Clark, and J. Salvi. SLAM with dynamic target via single cluster PHD filtering. *IEEE Selected Topics on Signal Processing*, 7(3):543–552, June 2013.

[9] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J.D. Tardos. A comparison of SLAM algorithms based on a graph of relations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, USA, October 2009.

[10] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

[11] M. Walter, R. Eustice, and J. Leonard. Exactly sparse extended information filters for feature based SLAM. *International Journal of Robotics Research*, 26(4):335–359, April 2007.

[12] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardos, and J. Neira. Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems*, 55(1):21–29, 2007.

[13] G. Hu, S. Huang, and G. Dissanayake. Evaluation of pose only SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3732–3737, Oct 2010.

[14] D. Schuhmacher, B.T. Vo, and B.N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 86(8):3447–3457, 2008.

[15] P.J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[16] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM-3D mapping outdoor environments. *Journal of Field Robotics*, 24(8-9):699–722, 2007.

[17] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.

[18] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3D object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010.

[19] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[20] R. Y. Takimoto, M. de Sales Guerra Tsuzuki, R. Vogelaar, T. de Castro Martins, A. K. Sato, Y. Iwao, T. Gotoh, and S. Kagei. 3D reconstruction and multiple point cloud registration using a low precision RGB-D sensor. *Mechatronics*, 35:11–22, 2016.

[21] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[22] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, 2003.

[23] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[24] A. Parra and T.J. Chin. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2868–2882, 2017.

[25] P. Barrios, M. Adams, K. Leung, F. Inostroza, G. Naqvi, and M. E. Orchard. Metrics for evaluating feature-based mapping performance. *IEEE Transactions on Robotics*, 33(1):198–213, 2017.

[26] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[27] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, December 2012.

[28] S. Schwertfeger. *Robotic Mapping in the Real World: Performance Evaluation and System Integration.* PhD thesis, School of Engineering and Science, Jacobs University, Bremen, Germany, 2012.

[29] P. Barrios, G. Naqvi, K. Leung, and F. Inostroza. The cardinalized optimal linear assignment (COLA) metric for multi-object error evaluation. In *18th International Conference on Information Fusion (Fusion)*, pages 271–279, July 2015.

[30] H. Moravec and A.E. Elfes. High resolution maps from Wide Angle Sonar. In *IEEE International Conference on Robotics and Automation*, pages 116–121, New Jersey, March 1985.

[31] J. Mullane, M. Adams, and W.S. Wijesoma. Robotic mapping using measurement likelihood filtering. *International Journal of Robotics Research*, 2(28):172–190, 2009.

[32] A. Birk. A quantitative assessment of structural errors in grid maps. *Autonomous Robots*, 28(2):187–196, 2010.

[33] B. Balaguer, S. Balakirskyand S. Carpin, and A. Visser. Evaluating maps produced by urban search and rescue robots: Lessons learned from robocup. *Aut. Robots*, 27(4):449–464, 2009.

[34] J. Pellenz and D. Paulus. Mapping and map scoring at the robo-cup rescue competition. *Quantitative Performance Evaluation of Navigation Solutions for Mobile Robots (RSS 2008, Workshop CD)*, 46:47–66, 2008.

[35] M. Chandran-Ramesh and P. Newman. Assessing map quality using conditional random fields. *Field and Service Robotics - Springer Tracts in Advanced Robotics*, 2008.

[36] J. Sturm. RGB-D SLAM dataset and benchmark. [Online] Available: `http://vision.in.tum.de/data/datasets/rgbd-dataset`, Last accessed on 2017.

[37] F. Amigoni, S. Gasparini, and M. Gini. Good experimental methodologies for robotic mapping: A proposal. In *International Conference on Robotics and Automation (ICRA)*, Rome, Italy, April 2007.

[38] A.I. Wagan, A. Godil, and X. Li. Map quality assessment. In *PerMIS '08, Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, pages 278 – 282, New York, USA, 2008.

[39] European Union FP6. The rawseeds project. [Online] Available: `http://www.rawseeds.org/rs/methods/view//10`, Last accessed on 2017.

[40] J.R. Hoffman and R. Mahler. Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems, Man and Cybernetics*, 34(3):327–336, May 2004.

[41] A. Rahmathullah, A. García-Fernández, and L. Svensson. Generalized optimal sub-pattern assignment metric. In *20th International Conference on Information Fusion (Fusion)*, pages 1–8, 2017.

[42] M. Beard, Ba Tuong Vo, and Ba-Ngu Vo. OSPA(2): Using the OSPA metric to evaluate multi-target tracking performance. In *2017 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 86–91, 2017.

[43] M. Beard, Ba Tuong Vo, and Ba-Ngu Vo. A solution for large-scale multi-object tracking. *IEEE Transactions on Signal Processing*, 68:2754–2769, 2020.

[44] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[45] D.G. Sim, O.K. Kwon, and R.H Park. Object matching algorithms using robust Hausdorff distance measures. *IEEE Transactions on Image Processing*, 8(3):425–429, March 1999.

[46] C.F. Olson and D.P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, January 1997.

[47] T. Zajic, J. Hoffman, and R. Mahler. Scientific performance metrics for data fusion: New results. In *Proc. SPIE 4052, Signal Processing, Sensor Fusion, Target Recognition IX*, pages 172–182, I. Kadar Ed.,, 2000.

[48] J. Hoffman, R. Mahler, and T. Zajic. User-defined information and scientific performance evaluation. In *Signal Processing, Sensor Fusion, and Target Recognition X*, volume 4380, pages 300–311, I. Kadar Ed., 2001. International Society for Optics and Photonics, SPIE.

[49] A.J. Baddeley. Errors in binary images and an L^p version of the Hausdorff metric. *Nieuw Archief voor Wiskunde*, 10:157–183, 1997.

[50] J.R. Hoffman and R.P.S. Mahler. Multitarget miss distance and its applications. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 1, pages 149–155, July 2002.

[51] D. W. Pentico. Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176(2):774 – 793, 2007.

[52] E. Levina and P. Bickel. The earth mover's distance is the mallows distance: some insights from statistics. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume 2, pages 251–256, 2001.

[53] K. Ganstrom, C. Lundquist, and O. Orguner. Extended target tracking using a Gaussian-mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4), October 2012.

[54] B. Ristic, B.N. Vo, D. Clark, and B.T. Vo. A metric for performance evaluation of multi-target tracking algorithms. *IEEE Transactions on Signal Processing*, 59(7):3452–3457, July 2011.

[55] D.F. Crouse, P. Willett, M. Guerriero, and L. Svensson. An approximate minimum mospa estimator. In *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pages 3644–3647, May 2011.

[56] S. Nagappa, D.E. Clark, and R. Mahler. Incorporating track uncertainty into the OSPA metric. In *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, pages 1–8, July 2011.

[57] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[58] C.H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publications, 1998.

[59] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.

[60] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987.

[61] S. Gold, A. Rangarajan, C.P. Lu, S Pappu, and E Mjolsness. New algorithms for 2D and 3D point matching: pose estimation and correspondence. *Pattern recognition*, 31(8):1019–1031, 1998.

[62] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[63] A. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and vision computing*, 21(13-14):1145–1153, 2003.

[64] J. Yang, H. Li, D. Campbell, and Y. Jia. Go-ICP: a globally optimal solution to 3D ICP point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2016.

[65] T. T. D. Nguyen, H. Rezatofighi, B.N. Vo, B.T. Vo, S. Savarese, and I. Reid. How trustworthy are performance evaluations for basic vision tasks? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–15, 2022.

[66] M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007.

[67] W. Tabib, C. O'Meadhra, and N. Michael. On-manifold GMM registration. *IEEE Robotics and Automation Letters*, 3(4):3805–3812, 2018.

[68] H. Li and R. Hartley. The 3D-3D registration problem revisited. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007.

[69] D. Campbell and L. Petersson. GOGMA: Globally-optimal gaussian mixture alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5685–5694, 2016.

[70] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020.

[71] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008.

[72] N. Mellado, D. Aiger, and N. J. Mitra. Super4PCS fast global pointcloud registration via smart indexing. In *Computer graphics forum*, volume 33, pages 205–215. Wiley Online Library, 2014.

[73] H. Lei, G. Jiang, and L. Quan. Fast descriptors and correspondence propagation for robust global point cloud registration. *IEEE Transactions on Image Processing*, 26(8):3614–3623, 2017.

[74] L. Bernreiter, L. Ott, J. Nieto, R. Siegwart, and C. Cadena. PHASER: A robust and correspondence-free global pointcloud registration. *IEEE Robotics and Automation Letters*, 6(2):855–862, 2021.

[75] J.S. Vitter. Faster methods for random sampling. *Communications of the ACM*, 27(7):703–718, 1984.

[76] A. Nüchter. *3D robotic mapping: the simultaneous localization and mapping problem with six degrees of freedom*, volume 52. Springer, 1st edition, 2009.

[77] B. Xiong, W. Jiang, D. Li, and M. Qi. Voxel grid-based fast registration of terrestrial point cloud. *Remote Sensing*, 13(10):1905, 2021.

[78] S. Salti, F. Tombari, and L. Di Stefano. SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[79] R.B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3d registration. In *International Conference on Robotics and Automation (ICRA)*, pages 3212–3217. IEEE, 2009.

[80] R.B. Rusu, Z.C. Marton, N. Blodow, and M. Beetz. Persistent point feature histograms for 3D point clouds. In *Proceedings of the 10th International Conference on Intelligent Autonomous Systems (IAS-10), Germany*, pages 119–128, 2008.

[81] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[82] M.B. Wright. Speeding up the hungarian algorithm. *Computers & Operations Research*, 17(1):95–96, 1990.

[83] W. Jones, A. Chawdhary, and A. King. Optimising the Volgenant–Jonker algorithm for approximating graph edit distance. *Pattern Recognition Letters*, 87:47–54, 2017.

[84] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[85] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G^2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, Shanghai, China, May 2011.

ANNEX A

List of Publications

The list of all journal and conference publications by the author, related to this thesis work, is as follows:

Journal Papers

- P. Barrios, M. Adams, K. Leung, F. Inostroza, G. Naqvi, M. E. Orchard, Metrics for evaluating feature-based mapping performance. *IEEE Transactions on Robotics*, 33(1), pp 198-213, 2016.
- L. Cament, M. Adams, P. Barrios, Space debris tracking with the Poisson labeled multi-Bernoulli filter. *MDPI Sensors*, 21(11), pp 1-28, 2021.

Conference Papers

- P. Barrios, V. Guzman, M. Adams. PSO-COLA: A Robust Solution for Correspondence-Free Point Set Registration. *In IEEE International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 1-8. 2022.
- L. Cament, M. Adams, P. Barrios. Multi-Space-Object Tracking with the Poisson Labeled Multi-Bernoulli (PLMB) Filter Probabilistic Admissible Region Constraints. *In 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, pp. 1-15, 2021.
- P. Barrios, M. Adams. (2017). Point set registration based on multi-object metrics. *In IEEE International Conference on Control, Automation and Information Sciences (ICCAIS)*, pp. 245-250, 2017.
- P. Barrios, G. Naqvi, M. Adams, K. Leung, F. Inostroza. The Cardinalized Optimal Linear Assignment (COLA) metric for multi-object error evaluation. *In IEEE 18th International Conference on Information Fusion (Fusion)*, pp. 271-279, 2015.