TÉCNICO LISBOA

A Unified Approach for Pose Graph Optimization

Gabriel Antunes Moreira

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Prof. João Paulo Salgado Arriscado Costeira
Dr. Manuel Ricardo de Almeida Rodrigues Marques

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Prof. João Paulo Salgado Arriscado Costeira
Members of the Committee: Prof. Frank Dellaert
Dr. Ricardo da Silveira Cabral

September 2020

Acknowledgements

Firstly, I would like to express my appreciation for the academic journey I underwent at IST, for all the great Professors of the integrated M.Sc. in Aerospace Engineering whom I had the pleasure of meeting, for all the good friends I made along the way and for having had the opportunity to pursue a Double-Degree at ISAE-SUPAERO. From my time in Toulouse, a big thank you goes out to Roland Brochard who supervised my internship at Airbus Defence & Space and helped cement my interest in Computer Vision.

As far as this thesis goes, I would like to express my gratitude to my incredible supervisors Prof. João Paulo Costeira and Dr. Manuel Marques for their constant support, insight and feedback. The countless hours of invaluable discussions and brainstorming not only laid the groundwork for this thesis, but contributed towards my growth as a researcher. Their knowledge and guidance proved vital when it came to give shape to my nebulous ideas and find the best way to convey them. They truly are one of the best teams I have had the priviledge of working with.

Last but not least, I would like to thank my father for passing on to me his enthusiasm for mathematics and engineering, my sister for her uninterrupted support and my mother, who was always eager to see the end results of my work.

Resumo

A optimização de grafos de poses (PGO) é um importante problema em robótica e visão computacional, cujo objetivo consiste em determinar um conjunto de transformações rígidas que permita registar a informação visual proveninente de múltiplos observadores num referencial global comum. Apesar do sucesso das técnicas atuais de PGO, o teto de performance foi atingido aquando do aparecimento de novas aplicações de grande-escala, entre as quais, a navegação de veículos aéreos autónomos e *smart-cities*, onde é necessário fundir inúmeras imagens correspondentes a cenas de grandes dimensões. As formulações de PGO atuais mais comuns baseiam-se em métodos iterativos que refinam uma estimativa inicial até à convergência da mesma. Alternativamente, trabalhos de investigação recentes levaram à identificação de uma nova restrição global e ao aparecimento de novos algoritmos. Até ao momento, estas duas formulações existiam em paralelo, incapazes de tirar partido uma da outra. Nesta tese, apresentamos um novo modelo que permite a sua unificação, e que admite uma solução em forma-fechada próxima do óptimo global. A nossa solução permite optimizar grafos de poses de grande-escala com elevada precisão e baixo custo computacional.

Palavras-chave: Nuvens de pontos, Optimização, SLAM, Reconstrução 3D, Registo

Abstract

Pose Graph Optimization (PGO) is an important problem in computer vision, particularly in motion estimation, whose objective consists of finding the rigid transformations that achieve the best global alignment of visual data on a common reference frame. PGO's relative success in robotics has been recently overshadowed by large-scale applications such as unmanned aerial vehicles, self-driving cars and smart-cities, where it is necessary to fuse many images covering large areas but where each of them has few pairwise observations. The vast majority of state-of-the-art PGO approaches rely on iterative techniques which refine an initial estimate until convergence is achieved. Alternatively, recent works have identified a global constraint which has cast this problem into the matrix completion domain. Until now, both formulations coexisted unable to share the advantages of each other. We developed a methodology that unifies them, leading to a closed-form solution near the global optimum. Our formulation allows for high scalability, low computational cost and high precision, simultaneously.

Keywords: SLAM, Point clouds, Registration, Pose Graph Optimization, 3D reconstruction

Contents

List of Tables

List of Figures

Nomenclature

Matrices

R 3D rotation matrix

M Rigid transformation

$\widetilde{R}_{\mathcal{B}}$ Block matrix with measured relative rotations

$\widetilde{M}_{\mathcal{B}}$ Block matrix with measured relative rigid transformations

Graphs

$\mathcal{G}$ Graph

$\mathcal{L}$ Graph Laplacian

A Adjacency matrix

D Degree matrix

Linear Algebra

I_m $m \times m$ Identity matrix

$\mathbb{1}_{m \times n}$ $m \times n$ matrix with ones

$|| \, . \, ||_F$ Frobenius norm

$\mathrm{SO}(3)$ Special Orthogonal Group

$\mathrm{SE}(3)$ Special Euclidean Group

$A \circ B$ Hadamard product

$A \otimes B$ Kronecker product

$A^\top$ Transpose

$\mathrm{tr}\ A$ Trace operator

$\mathcal{K}_r(A, b)$ Krylov subspace of order r generated by A and b

Acronyms

ALM Augmented Lagrangian Multipliers.

DN Damped Newton.

EM Expectation Maximization.

ICP Iterative Closest Point.

IRLS Iterative Reweighted Least Squares.

LM Levenberg-Marquardt.

LR-MC Low Rank Matrix Completion.

LRS Low Rank and Sparse.

MA Motion Averaging.

MD Median.

MLE Maximum Likelihood Estimation.

MN Mean.

PDF Probability Density Function.

PGO Pose Graph Optimization.

RANSAC Random Sample Consensus.

RMSE Root-Mean-Square Error.

SLAM Simultaneous Localization and Mapping.

SNR Signal-to-Noise Ratio.

SVD Singular Value Decomposition.

Chapter 1

Introduction

The registration of 3D point sets obtained by LiDARs, RGB-D cameras or other 3D scanners is an important problem in Robotics and Computer Vision, with applications ranging from dense scene reconstruction to localization. This task can be defined as follows: given two point clouds such those represented in Fig. 1.1, find the rigid transformations to the same reference frame, M_1 and M_2, that achieve the best alignment of the visual data (bottom of the figure).

If the point correspondences are known, the desired transformations are those which minimize the pairwise distances between corresponding points and can be computed in closed-form via Generalized Procrustes Analysis [1].

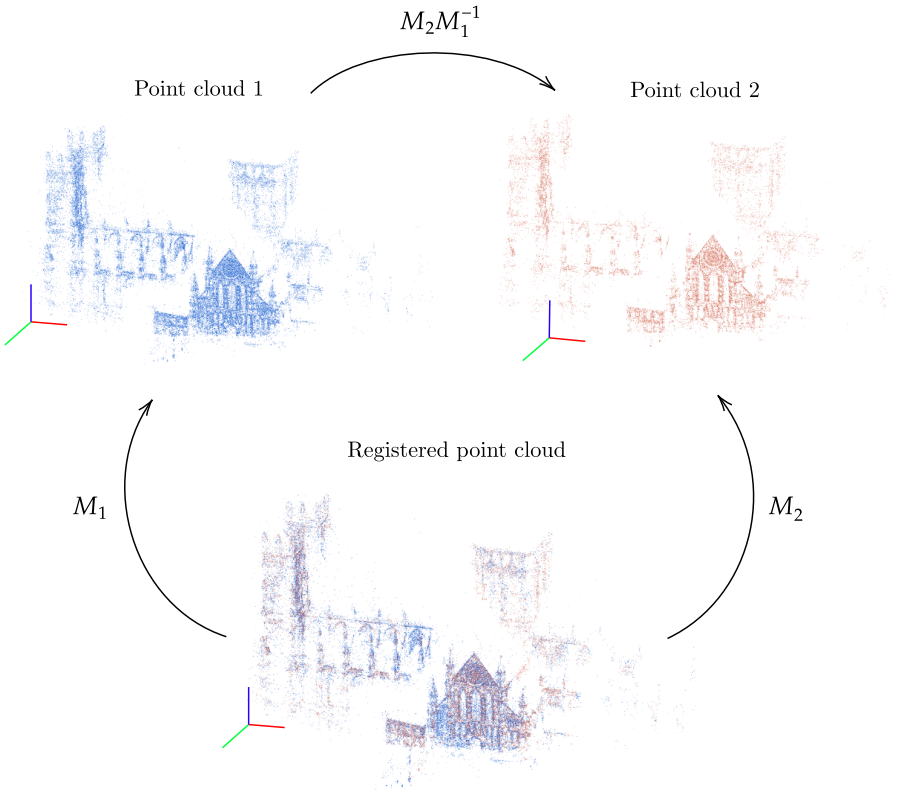


Figure 1.1: Example of the registration of two point clouds from the Yorkminster dataset [2].

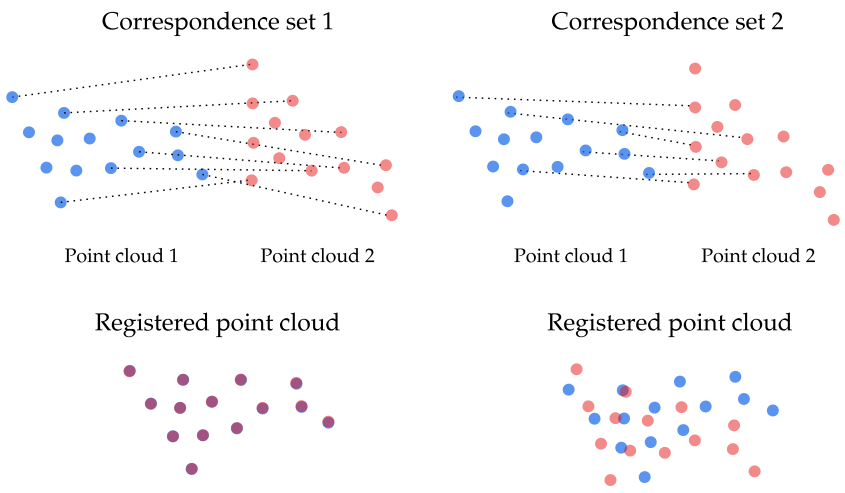


Figure 1.2: The correspondence problem.

In most applications of interest, however, the correspondences are unknown. Consider the example in Fig. 1.2. On the left side, using a subset of the true correspondences, we can accurately register the point sets. As depicted on the right side, by using an alternative set of correspondences the result is far from optimal. A brute-force attempt to compute the rigid transformation for each possible set of point combinations would be computationally infeasible even for moderately sized point clouds. We are thus faced with a problem which is not only geometrical but also combinatorial and complexity-wise NP-complete. That being so, it is common for registration procedures to obtain a coarse initial registration, e.g. by using image descriptors in the case of RGB-D data, and then iteratively alternate between estimating point correspondences and computing the best transformation between them. This strategy is part and parcel of a class of algorithms known as Iterative Closest Point (ICP) [3–6].

1.1 Registration of multiple point clouds

Consider now that we dispose of not two, but multiple point clouds corresponding to different views of the same scene, as is represented in the example from Fig. 1.3. These may be obtained via an array of 3D scanners, or alternatively, they may represent the visual data acquired by single observer as it moves through space. Both situations have seen a rise in popularity lately with applications such as smart-cities and autonomous transportation systems e.g. self-driving cars and drones, where the registered 3D data may be used to perform object detection, tracking, and mapping.

A fast and reliable ICP equivalent is yet to be designed for the registration of multiple point sets. Consequently, it comes as no surprise that recent research efforts have turned their attention to this problem. Initial attempts to solving it consisted of an incremental registration framework where two point clouds would be registered to obtain an initial 3D model, which would then be updated iteratively via the registration of new point clouds, one at a time, to the model from the previous iteration. If on the one hand, this method might work for small number of point sets, real-world applications like the ones mentioned typically involve a large number of frames. Therefore, error integration becomes a problem.

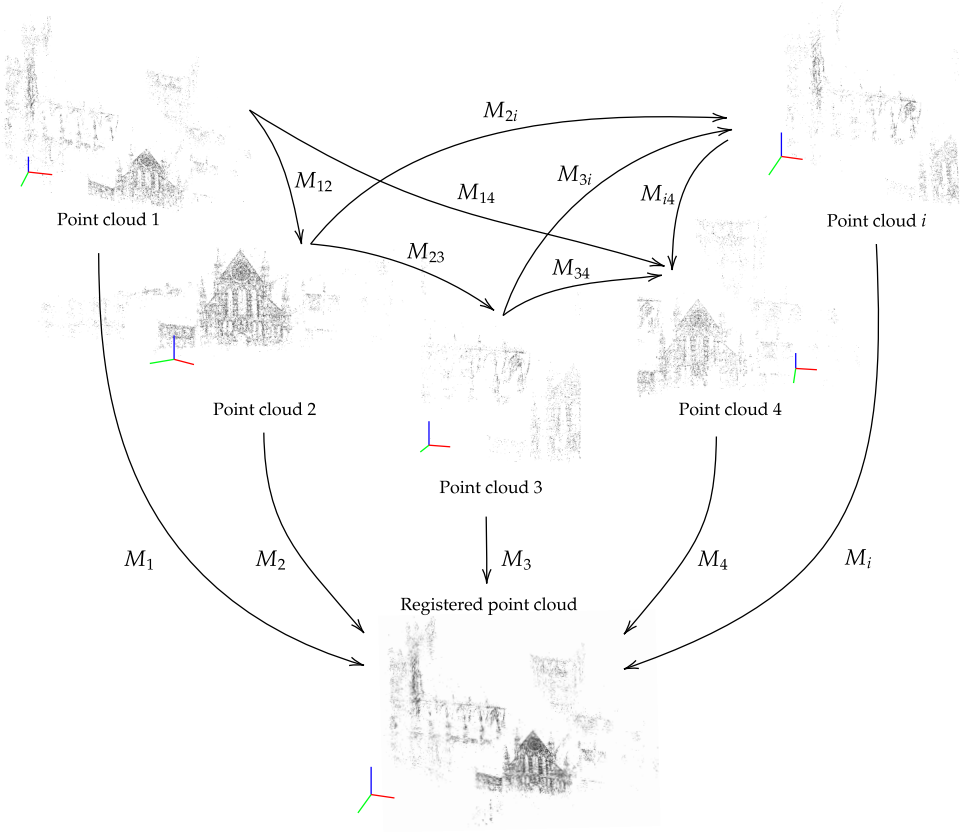
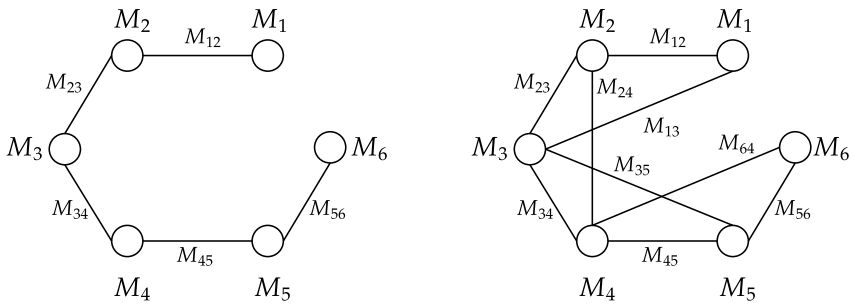


Figure 1.3: Example of the registration of multiple point clouds from the Yorkminster dataset [2].

Furthermore, it is not uncommon for different subsets of point clouds to correspond to non-overlapping regions. This may be a consequence of occlusion or the sheer dimensions of the whole scene. In these situations, it is impossible to establish point correspondences and ICP invariably fails to produce the desired rigid transformation. This incremental formulation has since been abandoned and in the modern literature we can identify two strategies for dealing with the problem of multiple point cloud registration:

- **Optimization in the point space**. The registration results from minimizing the distances between overlapping regions in all the point clouds. These techniques are capable of reaching a global minimum and thus the solution to the problem in question. When the correspondences are known in advance there is a closed-form least-squares solution obtained through Extended Procrustes Analysis [7]. For unknown correspondences, Toldo et al. [8] propose an iterative framework considering all the point clouds simultaneously and estimating correspondences by using mutual nearest neighbour groups. While, in theory, this solution deals with both the correspondence problem and the rigid transformations, without a point correspondence heuristic it implies searching for the nearest neighbors of all the points in all the point clouds.

- **Motion averaging (MA)**. This approach consists of finding the poses M_i relative to the same reference frame, given a set of noisy pairwise measurements of their ratios $M_i M_j^{-1}$ which can be interpreted as edges (i, j) of a simple and connected graph $\mathcal{G}$ known as the pose graph in the literature. If the only ratios (or edges) known are observations of $M_i M_{i+1}^{-1}$, for $i = 1, \ldots, n-1$,



Error redistribution not possible

Error redistribution possible

Figure 1.4: Examples of pose graphs. For MA to be possible, the associated graph cannot be acyclic.

this formulation degenerates to the incremental approach aforementioned. The associated graph is acyclic and cannot be optimized. However, when $\mathcal{G}$ has at least one cycle, MA can be used to redistribute the error among the point clouds. Additionally, as represented in the examples of Fig. 1.4, the graph need not be complete, only cyclic. The information available on the leftmost pose graph cannot be used to derive a better solution. For the pose graph on right, it is possible to look for the transformations M_i such that the ratios $M_i M_j^{-1}$ are as close as possible to the observations of M_{ij}. As we will show extensively throughout this thesis, this means that even without all the pairwise measurements, it may be possible to accurately estimate the global transformations.

In scenarios involving a large number of dense point clouds, optimization in the point space quickly becomes intractable due to the challenge of establishing point correspondences as the search space grows. Consequently, researchers have turned to MA in order to achieve a good trade-off between accuracy and computational cost. Two seemingly independent MA formulations exist in the literature:

- When associated with the problem of Simultaneous Localization and Mapping (SLAM), MA is usually referred to as Pose Graph Optimization (PGO) due to the underlying graph structure [9–12]. Common PGO frameworks consist of a probabilistic model for the pairwise transformation measurements and Maximum Likelihood Estimation (MLE) to find the global transformations that best fit those measurements. This is, however, a difficult task since the likelihood function is non-concave and hard to maximize.

- Recent works in the Computer Vision literature have identified a global optimization constraint which has cast MA into the Low Rank Matrix Completion (LR-MC) domain [13, 14]. This novel interpretation of the problem led researchers to the use of different iterative optimization techniques and to the discovery of closed-form suboptimal solutions [15]. Nonetheless, these formulations often fail to attain the same level of accuracy as MLE.

Since the advances in both fields seem to complement each other, we raise the question regarding the possibility of a unification.

1.2 Contribution

We address the problem of PGO in the context of point cloud registration. The desiderata for our solution are: scalability, efficiency and accuracy. While existing methods, namely those proposed in [9, 10, 13, 15] satisfy some of these requirements, they usually incur an efficiency-accuracy trade-off.

This thesis builds on MLE and LR-MC, improves upon both and closes the gap between them. More specifically, we make a three-fold contribution:

1. We demonstrate that current formulations of MLE and LR-MC are actually inconsistent under the generative noise model commonly assumed by the former. This explains the poorer results produced by the latter. Furthermore, we also show that the matrix completion approach is both unnecessary and inefficient.

2. We show that in applications where a high Signal-to-Noise Ratio (SNR) is a valid assumption, e.g. in Computer Vision, the problem of finding the optimal rigid transformations can be split into two manageable subproblems which deal separately with rotations and translations.

3. We propose two closed-form solutions to the problem of MA corresponding to the subproblems mentioned. These solutions arise from the unification of the objective function used in MLE and the block matrix model used in LR-MC put forward by Arrigoni et al. [13]. They are not initialization-dependent and are shown empirically to be in the basin of attraction of the global optimum, not suffering from the failure modes displayed by the closed-form solution proposed by Arrigoni et al. [15]. Additionally, a simple outlier detection method is presented which succeeds in improving the closed-form solution when the noise model assumption is not verified.

In summary, the pith of this thesis focuses on a very fast closed-form solution to PGO which, under a high SNR assumption, yields accurate results. The entirety of the code, comprising our proposed solution and additional modules is available at https://github.com/gabmoreira/pipe. We list below some of the functionalities implemented, from the ground up, in C++:

- **Visualization Module**: Displays camera trajectories and 3D scene reconstructions.

- **Krylov-Schur**: Computes a small subset of eigenvalues and eigenvectors of large sparse matrices efficiently. Based on the seminal paper by Stewart [16].

- **Riemannian Grad**: Gradient descent optimization on the Special Orthogonal group manifold.

- **3D RANSAC**: Computes rigid transformation estimates based on image features, for RGB-D data.

- **Global ICP**: Implementation of the algorithm proposed by Toldo et al. [8] to compute the global registration of multiple point clouds in the point space.

- **RGB-D 3D Reconstruction Pipeline**: Identifies loop closures, computes pairwise transformations to build the pose graph and optimizes it.

One of the main results of this thesis is illustrated in Fig. 1.5. Using 1124 RGB-D images from the Burghers dataset [17] we reconstructed the original 3D scene by estimating a subset of the relative poses and optimizing the pose graph using our proposed algorithm (which takes 0.03 seconds).

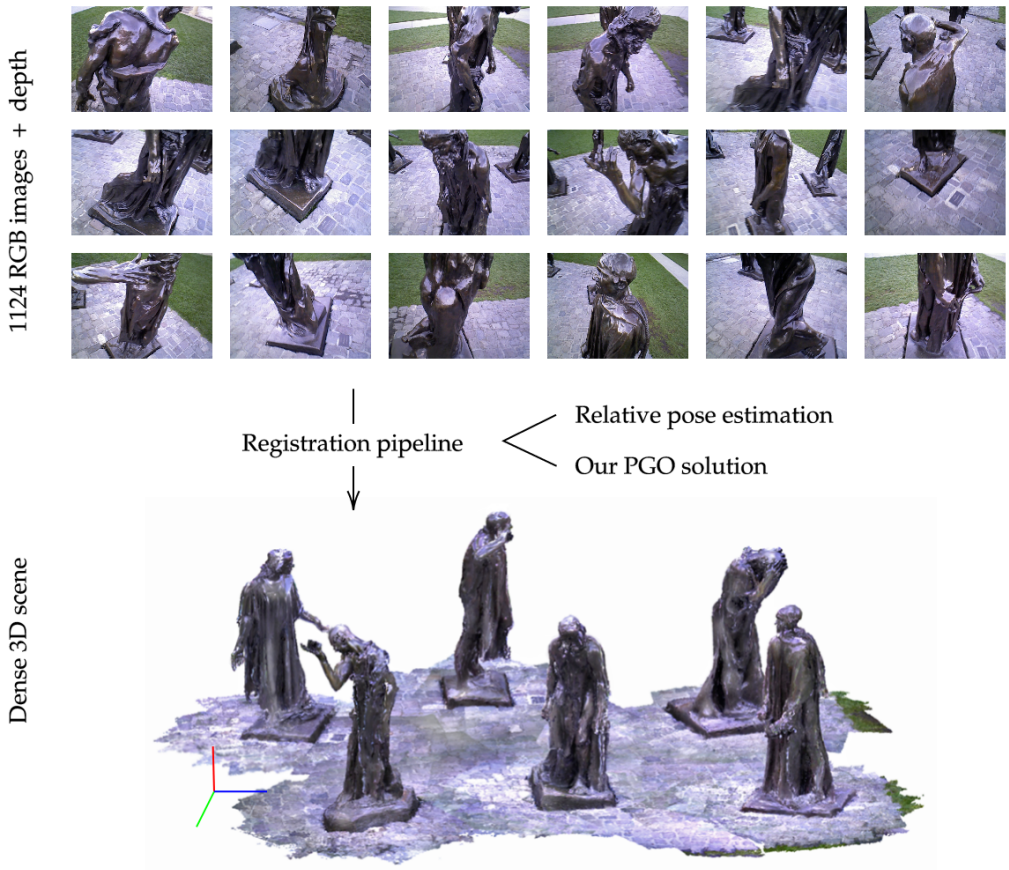


Figure 1.5: Dense scene reconstruction of the Burghers dataset [17] using our registration pipeline.

1.3 Thesis outline

The remainder of the thesis is organized as follows. In Chapter 2 we define the general problem of PGO or equivalently MA, as is currently presented in the literature. We then put forward two independent state-of-the-art formulations, MLE (Section 2.2) and LR-MC (Section 2.3). For each of them, we describe several optimization strategies.

The theoretical fundamentals underlying our formulation are presented in Chapter 3. Our model is subsequently described in Chapter 4, as well as the derivation of our closed-form solution. Finally, all the experiments conducted to validate our solution and compare it against the state-of-the-art are explained in detail in Chapter 5, including accuracy metrics, datasets used and benchmarks.

For the reader unfamilizaried with Graph theory or the Special Orthogonal Group, used througout this thesis to represent 3D rotations, some basics concepts, simulations and interesting results are available in Appendix A. Since many formulas are too long to include in the main text, the reader is referred to Appendix B for the complete derivations. Finally, images of popular SLAM PGO datasets and the complete 3D scene from Fig. 1.5 can be found in Appendices C and D, respectively.

Chapter 2

Background

In this chapter we provide an overview of prior work done on Motion Averaging (MA), or more generally, group synchronization with applications in point cloud registration and SLAM. We begin by defining the general problem of MA, in Section 2.1. We then present two independent approaches: MLE in Section 2.2 and Low Rank Matrix Completion (LR-MC) in Section 2.3. For each of them we put forward different optimization techniques.

2.1 Motion averaging or pose graph optimization?

MA is the problem of estimating a set of rigid transformations $M_1 \dots M_n$ from noisy measurements of their ratios $M_i M_j^{-1}$. Let $E \subseteq \{1 \dots n\}^2$ be the set of index tuples corresponding to the measured relative transformations and $\widetilde{M}_{ij}$ for $(i,j) \in E$, the relative transformation measurement from pose j to pose i. The poses here mentioned may encode only attitude, in which case they correspond to 3D rotations and can be represented as orthonormal matrices in the *special orthogonal group* $\mathrm{SO}(3)$ (Appendix A.1). Alternatively, they may refer to both attitude and position. In this case, each pose can be associated with a pair $\{R_i, t_i\}$ in the Euclidean group $\mathrm{E}(3)$, where $R_i \in \mathrm{SO}(3)$ and $t_i \in \mathbb{R}^3$. A matrix representation is also possible since $\mathrm{E}(3)$ is isomorphic to the *special Euclidean group* $\mathrm{SE}(3)$, defined as

$$\mathrm{SE}(3) = \left\{ M \in \mathbb{R}^{4\times 4}, \;\; M = \begin{bmatrix} R & t \\ 0_3^\top & 1 \end{bmatrix} : R \in \mathrm{SO}(3) \text{ and } t \in \mathbb{R}^3 \right\}$$

MA consists of seeking the global poses $\{M_i\}_{i=1,\dots,n}$ such that $M_i M_j^{-1}$ is as close as possible to the measurement $\widetilde{M}_{ij}$, for all $(i,j) \in E$. Mathematically, this problem can be stated as:

$$\underset{\substack{M_i \in \mathrm{SE}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmin}} \sum_{(i,j)\in E} \rho\big(d(\widetilde{M}_{ij} - M_i M_j^{-1})^2\big) \tag{2.1}$$

where d denotes a distance measure in $\mathrm{SE}(3)$ and ρ is a kernel function. While Eq. (2.1) already captures the essence of the problem, it is possible to introduce, or identify a graph structure underlying this formulation, which justifies the alternative designation of PGO. To illustrate this, consider the following

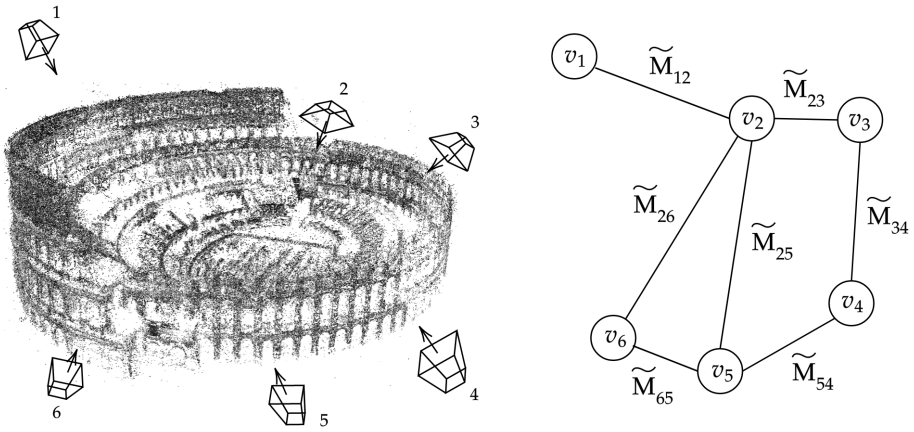


Figure 2.1: Pose graph example using the Colosseum dataset (left) from *Building Rome in a day* [18].

example, represented in Fig. 2.1. A system of 6 observers produces 6 point clouds of the same scene. In general, a complete overlap of all the point clouds is unattainable either due to occlusions or the dimensions of the scene. However, it can be assumed that it is possible to find a subset E of point cloud pairs which do overlap and such that for $(i, j) \in E$, where i and j denote point cloud indices, it is possible to obtain the pairwise transformation estimates $\widetilde{R}_{ij}$ and $\widetilde{t}_{ij}$. Equivalently, we can write the measured transformations as $\widetilde{M}_{ij} \in \mathrm{SE}(3)$.

By associating each transformation M_i to a vertex $v_i \in V$ and each pairwise measurement $\widetilde{M}_{ij}$ to an edge $(i, j) \in E$, we obtain a simple, undirected and connected graph $\mathcal{G} = (V, E)$. An important remark is that albeit defined as undirected, if the measurement $\widetilde{M}_{ij}$ is available, then the transformation in the opposition direction is obviously $\widetilde{M}_{ji} = \widetilde{M}_{ij}^{-1}$, i.e. we know how to go from i to j and from j to i. Using the equality $M_{ij} = M_i M_j^{-1}$, the distance in $\mathrm{SE}(3)$ corresponding to the Frobenius norm and the kernel $\rho(x) = x$, the problem boils down to

$$\boxed{\underset{\substack{M_i \in \mathrm{SE}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmin}} \sum_{(i,j)\in E} ||\widetilde{M}_{ij} - M_i M_j^{-1}||_F^2} \tag{2.2}$$

Equivalently, by using the definition of product of matrices in $\mathrm{SE}(3)$

$$M_i M_j^{-1} = \begin{bmatrix} R_i R_j^\top & t_i - R_i R_j^\top t_j \\ 0^\top & 1 \end{bmatrix}$$

we can write Eq. (2.2) more explicitly as

$$\underset{\substack{t_i \in \mathbb{R}^3 \\ R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmin}} \sum_{(i,j)\in E} ||\widetilde{t}_{ij} - t_i + R_i R_j^\top t_j||^2 + ||\widetilde{R}_{ij} - R_i R_j^\top||_F^2 \tag{2.3}$$

The l^2-norm (or Euclidean distance) in the left term is a common choice of distance in $\mathbb{R}^3$. As explained in Appendix A.1, the Frobenius norm of the difference of rotations, on the right, can easily be related to the geodesic distance on $\mathrm{SO}(3)$, or equivalently the angle, between $\widetilde{R}_{ij}$ and $R_i R_j^\top$.

2.2 Maximum likelihood estimation

The MA problem in Eq. (2.3) makes no assumptions regarding the observations $\widetilde{M}_{ij}$. Alternatively, we can formulate MA based on the assumption of a statistical noise model for the measurements. This allows us to derive a likelihood function that gauges how consistent a set of global rigid transformations is with the measured pairwise data. This is known as MLE and is the subject of this section. We begin by describing the generative noise models for the translation and rotation measurements commonly adopted in the literature: i.i.d. Gaussian and isotropic Langevin distributions, respectively [9, 19]. We then derive the corresponding likelihood function which we seek to maximize. We finish off by presenting different optimization strategies to accomplish this: two rotation initialization techniques, the chordal relaxation [20] and the Riemannian gradient descent [21], and finally a non-linear Gauss-Newton solver, g2o [11]. The former serve as an initialization to bootstrap the latter.

2.2.1 Generative noise model

While in reality, one can not claim that the different measurements $\widetilde{t}_{ij}$ and $\widetilde{R}_{ij}$ are independent, this simplification is often employed in the literature, producing good results. Therefore, for the remainder of this document the following hypothesis is assumed.

Hypothesis 2.2.1. *Let $\mathcal{G} = (V, E)$ denote a pose graph. All the measurements associated with different edges are deemed to be independent, as well as rotation and translation measurements corresponding to the same edge, i.e.*

$$\begin{cases} \forall_{(i,j)\in E} & \widetilde{t}_{ij} \perp\!\!\!\perp \widetilde{R}_{ij} \\ \forall_{(i,j)\in E}\ \forall_{(k,l)\in E\setminus(i,j)} & \widetilde{t}_{ij} \perp\!\!\!\perp \widetilde{t}_{kl} \\ \forall_{(i,j)\in E}\ \forall_{(k,l)\in E\setminus(i,j)} & \widetilde{R}_{ij} \perp\!\!\!\perp \widetilde{R}_{kl} \end{cases}$$

Translations An additive Gaussian noise model is assumed for the relative translation measurements. Let $\widetilde{t}_{ij}$ denote the measured translation from pose j to pose i, then

$$\widetilde{t}_{ij} = e_{ij} + t_{ij} = e_{ij} + t_i - R_i R_j^{\top} t_j$$

where $e_{ij} \in \mathbb{R}^3$ is a random variable following a Normal distribution whose covariance matrix is $\sigma_t^2 I_3$.

$$e_{ij} \sim \mathcal{N}(0_3, \sigma_t^2 I_3)$$

Consequently, and under Hypothesis 2.2.1 the random variables e_{ij} are i.i.d. with the following PDF

$$f_{\mathrm{N}}(e_{ij}) \propto \exp\Big(-\frac{1}{2\sigma_t^2}||e_{ij}||^2\Big) \tag{2.4}$$

Rotations The noisy rotation measurement of a relative pose i.e. the transformation from the coordinate system of pose j to that of pose i will be equal to the true rotation $R_i R_j^\top$, rotated by a noise matrix $E_{ij} \in \mathrm{SO}(3)$ as follows

$$\widetilde{R}_{ij} = E_{ij} R_i R_j^\top$$

The rotation matrices are often modeled in the literature using the isotropic Langevin distribution on $\mathrm{SO}(3)$ defined in Appendix A.1. It was also the one adopted here.

$$E_{ij} \sim \mathrm{Lang}(I_3, 1/\sigma_R^2)$$

The mean rotation was set to the Identity I_3, and the concentration parameter to $1/\sigma_R^2$, where σ_R^2 is an approximation of the variance (see in Appendix A.1 that for large $1/\sigma_R^2$, the Langevin distribution can be approximated by a Normal distribution with variance σ_R^2). Similarly to the model used for translations, the random variables E_{ij} are assumed to be i.i.d. with the following PDF

$$f_{\mathrm{L}}(E_{ij}) \propto \exp\big(1/\sigma_R^2 \ \mathrm{tr}\ E_{ij}\big) \tag{2.5}$$

2.2.2 Log-likelihood

Using Hypothesis 2.2.1 and the probability density functions presented, the log-likelihood function can now be derived. Let $\theta = \begin{bmatrix} R_1 & \dots & R_n & t_1 & \dots & t_n \end{bmatrix}^\top$ be the vector of parameters which need to be estimated and y the vector of observations of E_{ij} and e_{ij}. Under the assumed generative noise models, the likelihood function is given by

$$L(\theta, y) = p(y|\theta) = \prod_{(i,j)\in E} f_{\mathrm{N}}(e_{ij}) f_{\mathrm{L}}(E_{ij})$$

Since the natural logarithm is a smooth, differentiable and monotonous function, we use it to transform the product into a sum:

$$\log L(\theta, y) = \sum_{(i,j)\in E} \log f_{\mathrm{N}}(e_{ij}) + \log f_{\mathrm{L}}(E_{ij})$$

The observations of e_{ij} and E_{ij} can be expressed as $\widetilde{t}_{ij} - t_i + R_i R_j^\top t_j$ and $\widetilde{R}_{ij} R_j R_i^\top$, respectively. Thus,

$$\log L(\theta, y) = \sum_{(i,j)\in E} \log f_{\mathrm{N}}(\widetilde{t}_{ij} - t_i + R_i R_j^\top t_j) + \log f_{\mathrm{L}}(\widetilde{R}_{ij} R_j R_i^\top)$$

Using the PDFs in Eqs. (2.4) and (2.5) yields

$$\boxed{\log L(\theta, y) = c - \frac{1}{2\sigma_t^2} \sum_{(i,j)\in E} ||\widetilde{t}_{ij} - t_i + R_i R_j^\top t_j||^2 + \frac{1}{\sigma_R^2} \sum_{(i,j)\in E} \mathrm{tr}\ \widetilde{R}_{ij} R_j R_i^\top} \tag{2.6}$$

Maximum likelihood estimation consists of finding the vector of rotations and translations θ such that,

for a certain vector of observations y, the likelihood function $L(\theta, y)$, or equivalently the log-likelihood function is maximized. This approach can thus be reduced to the following problem

$$\underset{\theta \in \mathrm{SO}(3)^n \times (\mathbb{R}^3)^n}{\operatorname{argmax}} \quad \log L(\theta, y) \tag{2.7}$$

Equivalently, and as minimization problems are more common in the optimization literature, we could reframe Eq. (2.7) as seeking the minimizer of the negative log-likelihood.

2.2.3 Optimization strategies

The maximization problem in Eq. (2.7) is not concave. Therefore, it is in general hard to solve. This difficulty stems directly from the rotation matrices. As mentioned in Appendix A.1, $\mathrm{SO}(3)$ is a non-convex set and there does not seem to be any practical or exact way of performing optimization in this manifold. In fact, the work of Hartley et al. [22] indicates that the multiple rotation averaging term in the log-likelihood, $\sum_{(i,j)\in E} \operatorname{tr} \widetilde{R}_{ij} R_j R_i^\top$ has multiple peaks, far from the global optimum. This means that iterative first or second-order methods with a bad initialization may converge to local optima. We arrive thus at the conclusion that a good initialization for the rotations R_i is of the utmost importance.

Initialization

From the many rotation initialization techniques available in the literature, we will present two from [10]. These are the chordal relaxation by Martinec and Pajdla [20] and the Riemannian gradient descent by Tron and Vidal [21]. Both focus on solving

$$\boxed{\underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmax}} \sum_{(i,j)\in E} \operatorname{tr} \widetilde{R}_{ij} R_j R_i^\top = \underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmin}} \sum_{(i,j)\in E} \left|\left| R_i - \widetilde{R}_{ij} R_j^\top \right|\right|_F^2} \tag{2.8}$$

which constitutes the already mentioned multiple rotation averaging problem. Note that the optimal rotations for this problem are not necessarily the optimal rotations of Eq. (2.7) since the objective function there is not separable in translations and rotations.

Chordal relaxation Let us write the rotation matrix R_i in terms of its columns r_i^j, where j denotes the j-th column:

$$R_i = \begin{bmatrix} r_i^1 & r_i^2 & r_i^3 \end{bmatrix}$$

The problem in Eq. (2.8) can be stated as

$$\underset{\{r_i^k\}}{\operatorname{argmin}} \sum_{(i,j)\in E} \sum_{k=1}^{3} \left|\left| r_i^k - \widetilde{R}_{ij} r_j^k \right|\right|_F^2 \quad \text{s.t.} \quad \begin{bmatrix} r_i^1 & r_i^2 & r_i^3 \end{bmatrix} \in \mathrm{SO}(3) \text{ for } i = 1, \dots, n$$

Since the $\mathrm{SO}(3)$ constraint is hard to enforce, the authors propose instead solving the unconstrained least-squares problem

$$\underset{\{r_i^k\}}{\operatorname{argmin}} \sum_{(i,j)\in E} \sum_{k=1}^{3} ||r_i^k - \widetilde{R}_{ij} r_j^k||_F^2$$

Then, we let $M_i = \begin{bmatrix} r_i^1 & r_i^2 & r_i^3 \end{bmatrix}$ and since M_i will not necessarily be in $\mathrm{SO}(3)$ we solve the orthogonal Procrustes problem via SVD to retrieve the rotation estimate $\widehat{R}_i$,

$$\widehat{R}_i = \underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmin}} ||R_i - M_i||_F^2$$

This is a suboptimal solution since it needs to be projected to $\mathrm{SO}(3)$. Provided that the projection error is small, this solution yields satisfactory results.

Riemannian gradient descent Instead of solving a constrained problem, Tron and Vidal [21] propose the optimization to be performed directly on the $\mathrm{SO}(3)$ manifold. The authors consider both the geodesic distance on $\mathrm{SO}(3)$ and the chordal distance, mentioning that the latter was easier to optimize. As explained in Appendix A.1, their intrinsic metrics are equivalent. Since the chordal distance was the one used thus far, we will independently derive the results for this metric, which were omitted from [21]. To save space let us denote by ψ the scalar function we are optimizing:

$$\psi : \mathrm{SO}(3)^n \to \mathbb{R}$$
$$\psi = \sum_{(i,j)\in E} d^2_{\text{chordal}}(R_i, \widetilde{R}_{ij} R_j)$$

The authors conjectured that ψ does not have local minima and thus a gradient descent method using the Riemannian gradient could be used to find the global minimum. These claims were supported only by the experiments conducted. In order to compute the Riemannian gradient of ψ, we need the Euclidean derivative w.r.t to the k-th rotation (full derivation in Appendix B.5):

$$\frac{\partial \psi}{\partial R_k} = 2 \sum_{(k,j)\in E} (R_k - \widetilde{R}_{kj} R_j) + 2 \sum_{(i,k)\in E} (R_k - \widetilde{R}_{ik}^\top R_i)$$

Since we can assume that measurements are only taken in one direction, then $\widetilde{R}_{ik}^\top = \widetilde{R}_{ki}$. Moreover, both terms are summing over neighbors of v_k. Since k is fixed, we have:

$$\frac{\partial \psi}{\partial R_k} = 4 \sum_{v_j \in E(v_k)} (R_k - \widetilde{R}_{kj} R_j)$$

Using Definition A.1.1 (Appendix A.1) of the Riemannian gradient,

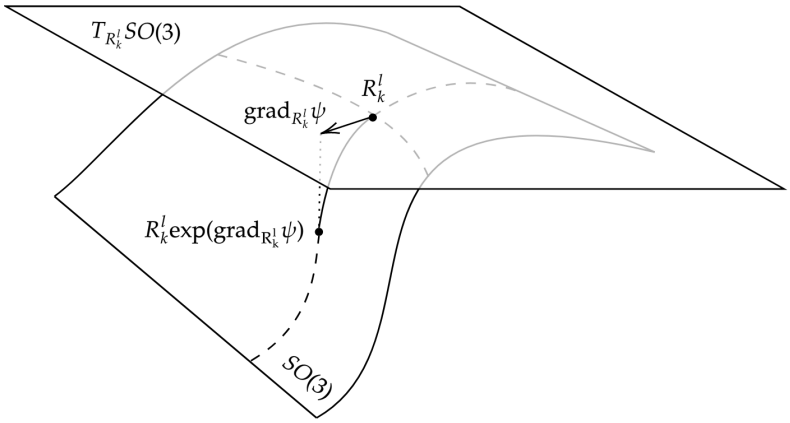


Figure 2.2: Riemannian gradient on $\mathrm{SO}(3)$

$$
\begin{aligned}
\text{grad}_{R_k}\psi &= \frac{\partial \psi}{\partial R_k} - R_k\Big(\frac{\partial \psi}{\partial R_k}\Big)^\top R_k \\
&= 4 \sum_{v_j \in E(v_k)} (R_k - \widetilde{R}_{kj} R_j) - 4R_k \sum_{(k,j)\in E} (R_k^\top - R_j^\top \widetilde{R}_{kj}^\top) R_k \\
&= 4 \sum_{v_j \in E(v_k)} \big(R_k R_j^\top \widetilde{R}_{jk} R_k - \widetilde{R}_{kj} R_j\big) \\
&= R_k \underbrace{\sum_{v_j \in E(v_k)} 4\big(R_j^\top \widetilde{R}_{jk} R_k - R_k^\top \widetilde{R}_{kj} R_j\big)}_{\text{skew}-\text{symmetric}}
\end{aligned}
\tag{2.9}
$$

As a sanity check note that $\text{grad}_{R_k}\psi = 0$ in the noiseless case, i.e. when $\widetilde{R}_{ij} = R_i R_j^\top$. As represented in Fig. 2.2, this gradient constitutes a vector tangent to $\mathrm{SO}(3)$ at R_k, i.e. $\text{grad}_{R_k}\psi \in \mathrm{T_{R_k}SO(3)}$. Therefore we can use the exponential map defined in Appendix A.1 to obtain a point of the manifold in the opposite direction. According to the definition of the exponential map, $\exp_{R_k}(R_k\Omega) = R_k \exp(\Omega)$, where Ω is skew-symmetric. Therefore,

$$
\begin{aligned}
\exp_{R_k}(\text{grad}_{R_k}\psi) &= \exp_{R_k}\left(R_k \sum_{v_j \in E(v_k)} 4\big(R_j^\top \widetilde{R}_{jk} R_k - R_k^\top \widetilde{R}_{kj} R_j\big)\right) \\
&= R_k \exp\left(4 \sum_{v_j \in E(v_k)} \big(R_j^\top \widetilde{R}_{jk} R_k - R_k^\top \widetilde{R}_{kj} R_j\big)\right)
\end{aligned}
$$

Now the definition of the gradient descent method on $\mathrm{SO}(3)$ is straightforward. Let R_k^l be the k-th rotation estimate at the l-th iteration. The gradient descent optimization on the manifold consists of the following updates

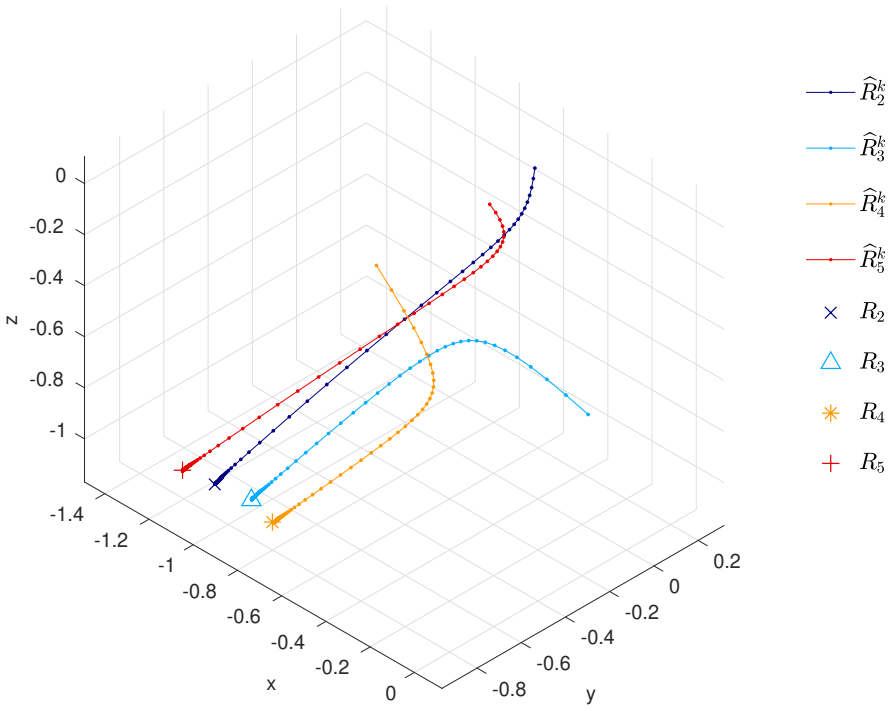


Figure 2.3: Convergence of four rotation estimates $\widehat{R}_i^k$ to the ground truth R_i, starting from a random initialization, per iteration k of the Riemannian gradient method, with $\epsilon = 10^{-4}$ (each rotation is represented as a vector in $\mathbb{R}^3$). Graph parameters: $|V| = 553$, $|E| = 103932$. The measurements were simulated with $\sigma_R = 3$ deg, and a probability of outliers of 0.02.

$$
\begin{aligned}
R_k^{l+1} &= R_k^l \exp_{R_k}(-\epsilon \operatorname{grad}_{R_k^l} \psi) \\
&= R_k^l \exp\left(-4\epsilon \sum_{v_j \in E(v_k)} \left(R_j^{l\top} \widetilde{R}_{jk} R_k^l - R_k^{l\top} \widetilde{R}_{kj} R_j^l \right) \right)
\end{aligned}
$$

where the positive scalar ϵ is the step size. This parameter needs to be chosen according to the characteristics of the pose graph to ensure convergence. For more on this topic, the reader is referred to [21]. An interesting remark is that the updates of the Riemannian gradient descent for a given rotation R_k will depend only on the immediate neighbors of v_k. This is similar to what happens in graph consensus algorithms where each vertex updates its state based on its immediate connections.

An example is provided in Fig. 2.3, illustrating the convergence of four rotation variables of a pose graph to the ground-truth. A vector representation in $\mathbb{R}^3$ for each $\widehat{R}_i^k$ was used. Interestingly, the gradient descent on the manifold does not seem to suffer from the typical noisy convergence which is common on the Euclidean space.

Iterative solvers

Assuming that a good initialization has been computed, an iterative solver can be used to seek the maximum of the likelihood function. In most applications, this optimization is carried out via Gauss-Newton or Levenberg-Marquardt (LM) algorithms. These methods can be summarized as follows. Let $f_{\mathrm{ML}}(\theta^k)$ denote our cost function with θ^k the vector of variables at the k-th iteration. Optimization comprises iteratively linearizing f_{ML} near the current estimate θ^k, computing an increment $\Delta\theta^k$ and performing the updates

$$\theta^{k+1} = \theta^k + \Delta\theta^k \tag{2.10}$$

until convergence is achieved. The difference between Gauss-Newton and LM happens in the computation of the increment $\Delta\theta^k$. The latter refines the convergence by introducing a variable damping factor to control the step size.

Among popular implementations of these methods to solve PGO problems are *g2o* [11] and *GTSAM* [12]. However, the updates performed by these solvers are not as straightforward as Eq. (2.10) since the space of variables is not Euclidean. In particular, by using the overparameterized representation of 3D rotations as matrices in $\mathrm{SO}(3)$ the update in Eq. (2.10) would lead to the violation of the group contraint ($\mathrm{SO}(3)$ is non-convex). Other rotation representations also incurr the same problem. In order to avoid this, *g2o* represents the current state θ_i^k as a translation vector and a normalized quaternion, $\theta_i^k = (t_i^k,\ q_i^k)$. The increment $\Delta\theta^k$ is represented as a translation vector and the axis of a quaternion $\Delta\theta_i^k = (\Delta t_i^k,\ \Delta q_i^k)$, with $\Delta q_i^k = (\Delta q_x,\ \Delta q_y, \Delta q_z)^\top$. The update is then accomplished via a nonlinear motion composition operator $\oplus$:

$$\theta_i^{k+1} = \theta_i^k \oplus \left(\Delta t_i^k,\ \frac{\Delta q_i^k}{\sqrt{1 - ||\Delta q_i^k||^2}}\right)$$

which is defined as

$$x_i \oplus x_j = \left(R_{q_i} t_j + t_i,\ q_i q_j\right)$$

These methods are typically fast, however only local optimality is guaranteed. This justifies the need for a good initialization in the basin of attraction of the global optimum, as was previously mentioned.

2.3 Motion averaging as a low-rank matrix completion problem

We now present a different formulation of MA based on the problem from Eq. (2.2) which aims to find the global poses M_i whose ratios $M_i M_j^{-1}$ are closest to the measurements $\widetilde{M}_{ij}$. The gist of this formulation lies in replacing the sum of squared errors $\sum_{(i,j)\in E} ||\widetilde{M}_{ij} - M_i M_j^{-1}||_F^2$ which we seek to minimize by an equivalent cost of the form $||W \circ (B - CD)||_F^2$, where W is square a binary mask and B is a square block matrix containing the measurements. The global transformation matrices we are seeking are vertically stacked in C which is a block column and their inverses in D which is a block line. This approach allows us to treat MA from a missing data perspective. Alternatively, we can search instead

for $X = CD$ which has low rank, hence the name low-rank matrix completion. We will begin by deriving this new cost function, which was put forward by Arrigoni et al. [13], and present two robust variants, one proposed by the same authors, making use of a sparse term to capture outlier measurements, another by Jin et al. [14], replacing $||\ ||_F^2$ by a l^1-norm . The problem of minimizing these cost functions is addressed at the end of the Section. This can be accomplished via zero-order methods such as the Expectation-Maximization algorithm by Guerreiro and Aguiar [23] or second-order algorithms, e.g. Damped-Newton by Buchanan and Fitzgibbon [24] and Augmented Lagrangian Multipliers (ALM) by Jin et al. [14]. A suboptimal closed-form solution based on eigendecomposition has also been proposed by Arrigoni et al. [15].

2.3.1 Problem statement

For $M_{ij} \in \mathrm{SE}(3)$, let us consider the following equality, which encodes the fact that the pairwise transformations should be consistent with the global transformation ratios,

$$M_{ij} = M_i M_j^{-1} \ \text{ for } (i,j) \in \{1,\dots,n\}^2 \tag{2.11}$$

Arrigoni et al. [13] derived a matrix version of Eq. (2.11):

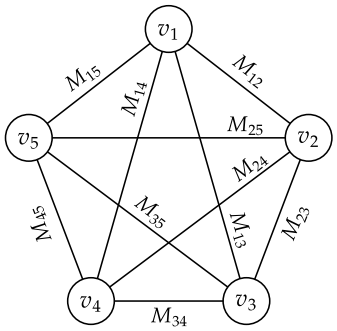
$$\underbrace{\begin{bmatrix} I & M_{12} & \dots & M_{1n} \\ M_{21} & I & \dots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \dots & I \end{bmatrix}}_{M_{\mathcal{B}} \in \mathbb{R}^{4n \times 4n}} = \underbrace{\begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix}}_{M_{\mathcal{C}} \in \mathbb{R}^{4n \times 4}} \underbrace{\begin{bmatrix} M_1^{-1} & M_2^{-1} & \dots & M_n^{-1} \end{bmatrix}}_{M_{\mathcal{C}}^{-b} \in \mathbb{R}^{4 \times 4n}} \tag{2.12}$$

which will be henceforth denoted as $M_{\mathcal{B}} = M_{\mathcal{C}} M_{\mathcal{C}}^{-b}$, where the subscripts $\mathcal{B}$ and $\mathcal{C}$ indicate a block matrix and a block column respectively. The superscript $-b$ denotes a block inverse. Note that apart from rigid transformations, Eq. (2.12) is valid for any $M_{ij} \in \mathrm{GL}_m(\mathbb{R})$. From the factorization $M_{\mathcal{C}} M_{\mathcal{C}}^{-b}$ we can say that $\mathrm{rank}\ M_{\mathcal{B}} = 4$. In fact, $\mathrm{rank}\ M_{\mathcal{B}} = \mathrm{rank}\ M_{\mathcal{C}} M_{\mathcal{C}}^{-b}$. Since $M_{\mathcal{C}}^{-b}$ is a $4 \times 4n$ matrix of rank 4 it follows that $M_{\mathcal{B}}$ has rank 4 as well.

A matrix equivalent to $M_{\mathcal{B}}$ can be built using the relative pose measurements $\widetilde{M}_{ij} \in \mathrm{SE}(3)$. The difference being that the block entries $(i,j) \in E$ are set to $\widetilde{M}_{ij}$ and those corresponding to missing measurements $(i,j) \notin E$ are set to a null block $0_4 \in \mathbb{R}^{4\times4}$. We will denote this new matrix by $\widetilde{M}_{\mathcal{B}}$:

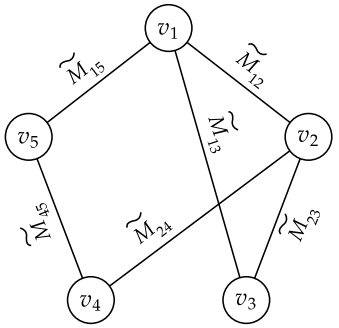
$$\widetilde{M}_{\mathcal{B}_{ij}} = \begin{cases} I_4 & \text{if } i = j \\ \widetilde{M}_{ij} & \text{if } (i,j) \in E \\ 0_4 & \text{otherwise} \end{cases}$$

An example is provided in Fig. 2.4, illustrating two pose graphs with $|V| = 5$ and the respective block matrices. The pose graph in Fig. 2.4a, denoted as $\mathcal{G}_{\mathrm{GT}} = (V, V^2)$ is complete and the edges correspond to the ground truth transformations. The respective relative pose matrix $M_{\mathcal{B}}$ does not have missing



$$M_{\mathcal{B}} = \begin{bmatrix} I_m & M_{12} & M_{13} & M_{14} & M_{15} \\ M_{21} & I_m & M_{23} & M_{24} & M_{25} \\ M_{31} & M_{32} & I_m & M_{34} & M_{35} \\ M_{41} & M_{42} & M_{43} & I_m & M_{45} \\ M_{51} & M_{52} & M_{53} & M_{54} & I_m \end{bmatrix}$$

(a) Ground truth pose graph $\mathcal{G}_{\mathrm{GT}}$ (left) and the respective relative pose block matrix (right).



$$\widetilde{M}_{\mathcal{B}} = \begin{bmatrix} I_m & \widetilde{M}_{12} & \widetilde{M}_{13} & 0_m & \widetilde{M}_{15} \\ \widetilde{M}_{21} & I_m & \widetilde{M}_{23} & \widetilde{M}_{24} & 0_m \\ \widetilde{M}_{31} & \widetilde{M}_{32} & I_m & 0_m & 0_m \\ 0_m & \widetilde{M}_{42} & 0_m & I_m & \widetilde{M}_{45} \\ \widetilde{M}_{51} & 0_m & 0_m & \widetilde{M}_{54} & I_m \end{bmatrix}$$

(b) Measured pose graph $\mathcal{G}$ (left) and the respective relative pose block matrix (right).

Figure 2.4: Pose graphs and the respective relative pose block matrices.

blocks and has rank 4. Fig. 2.4b shows $\mathcal{G} = (V, E)$, a spanning subgraph of $\mathcal{G}_{\mathrm{GT}}$ where each edge is associated with a relative pose measurement. Missing edges correspond to null blocks on the respective pose matrix $\widetilde{M}_{\mathcal{B}}$, which has unknown rank.

Using this block notation and letting A be the adjacency matrix of $\mathcal{G}$, we can formulate MA as the problem of seeking the rank-4 square matrix $M_{\mathcal{B}} = M_{\mathcal{C}} M_{\mathcal{C}}^{-b}$ whose non-zero blocks are closest to those of $\widetilde{M}_{\mathcal{B}}$, using the Frobenius norm of the difference as our distance function. Mathematically, this translates to the following optimization problem

$$\begin{aligned} \underset{M_{\mathcal{B}}}{\operatorname{argmin}} \ & \left|\left|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{B}})\right|\right|_F^2 \\ \text{s.t.} \ & M_{\mathcal{B}} = M_{\mathcal{C}} M_{\mathcal{C}}^{-b} \\ & M_{\mathcal{C}} \in \mathrm{SE}(3)^n \end{aligned} \tag{2.13}$$

By expanding the cost function, so as to write it in terms of rotations and translations, it becomes clear that it differs from the MLE objective. Enforcing the group constraint $M_{\mathcal{C}} \in \mathrm{SE}(3)^n$ is the most difficult part of the problem. Therefore, and as is common in the literature, Arrigoni et al. [13] propose solving MA via the following relaxation,

$$\begin{aligned} \underset{M_{\mathcal{B}}}{\operatorname{argmin}} \ & \left|\left|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{B}})\right|\right|_F^2 \\ \text{s.t.} \ & \operatorname{rank} M_{\mathcal{B}} = 4 \end{aligned} \tag{2.14}$$

which uses the rank constraint previously mentioned and corresponds to a Low Rank Matrix Completion (LR-MC) problem. Even though this relaxation may yield suboptimal solutions to the original problem, since a projection to $\mathrm{SE}(3)^n$ is necessary afterwards, the authors claim that it successfully averages transformations under Gaussian noise[1]. Using the rank constraint, the factorization $M_{\mathcal{B}} = M_{\mathcal{C}} M_{\mathcal{C}}^{-b}$ and replacing the group constraint for an orthogonality constraint on the columns of $M_{\mathcal{C}}$, the same authors derived an alternative formulation which admits a closed-form solution obtained via eigendecomposition:

$$\begin{aligned} \underset{M_{\mathcal{C}}}{\operatorname{argmin}} \ & \left|\left|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b})\right|\right|_F^2 \\ \text{s.t.} \ & M_{\mathcal{C}}^{-b} M_{\mathcal{C}} = n I_4 \end{aligned}$$

Due to the use of l^2-norms: $||\ ||_F$, these techniques may yield poor results in the presence of outliers. Additional robustness can be achieved by adding a sparse matrix to the cost function or by replacing the l^2-norm by a l^1-norm.

Sparse term Arrigoni et al. [13] suggest using the Frobenius norm but adding an additional sparse S matrix to take into account the outlier blocks. Their robust formulation is then

$$\begin{aligned} \underset{M_{\mathcal{B}}}{\operatorname{argmin}} \ & \left|\left|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{B}}) + S\right|\right|_F^2 \\ \text{s.t.} \ & S \text{ sparse in } \operatorname{supp} A \\ & \operatorname{rank} M_{\mathcal{B}} \leq 4 \end{aligned} \tag{2.15}$$

where $\operatorname{supp} A$ denotes the support of the adjacency matrix, i.e. the set of non-zero entries. Eq. (2.15) constitutes what is known as Low Rank and Sparse (LRS) decomposition problem.

l^1-norm The approach suggested by Jin et al. [14] draws from LRS, but applies the work of Zheng et al. [25] on low-rank matrix approximation which proposes changing the l^2-norm in Eq. (2.13) to a l^1-norm, known for its robustness. Instead of seeking $M_{\mathcal{B}}$, the authors make use of the $M_{\mathcal{C}} M_{\mathcal{C}}^{-b}$ factorization and relax the group constraint for the orthogonality one, which is easier to implement. The new problem, which can be tackled e.g. using the ALM method, becomes

[1] Even though the noise in the rotations is only approximately Gaussian for large values of the concentration parameter k in the Langevin distribution.

$$\underset{M_{\mathcal{C}}}{\operatorname{argmin}} \ \left|\left|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b})\right|\right|_1$$
$$\text{s.t.} \ \ M_{\mathcal{C}}^{-b} M_{\mathcal{C}} = nI_4$$

2.3.2 Optimization strategies

Hitherto in this section, several LR-MC formulations of MA were put forward based on the factorization identified by Arrigoni et al. [13]. These constitute different optimization problems on their own, with different constraints and cost functions. Consequently, one cannot use the same optimization technique in all of them. Table 2.1 provides a quick summary of the problems, the respective expressions, constraints and one possible optimization method to solve each one, based on the literature. Four optimization techniques are then explained in detail.

Problem	Cost function	Constraints	Optimization method
P_{EM}	$\left\|\left\|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{B}})\right\|\right|_F^2$	$\operatorname{rank} M_{\mathcal{B}} = 4$	Expectation-Maximization
$\mathrm{P}_{\mathrm{LRS}}$	$\left\|\left\|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{B}}) + S\right\|\right|_F^2$	$\substack{S \text{ sparse} \in \operatorname{supp}(A \otimes \mathbb{1}_{4\times 4}) \\ \operatorname{rank} M_{\mathcal{B}} \leq 4}$	Low Rank and Sparse
P_{DN}	$\left\|\left\|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b})\right\|\right|_F^2$	–	Damped Newton
$\mathrm{P}_{\mathrm{EIG}}$	$\left\|\left\|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b})\right\|\right|_F^2$	$M_{\mathcal{C}}^{-b} M_{\mathcal{C}} = nI_4$	Eigendecomposition
$\mathrm{P}_{\mathrm{ALM}}$	$\left\|\left\|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b})\right\|\right|_1$	$M_{\mathcal{C}}^{-b} M_{\mathcal{C}} = nI_4$	Augmented Lagrangian Multipliers

Table 2.1: Different problem formulations of MA using the block matrix approach and one possible optimization technique for each.

Expectation-Maximization

Problem P_{EM} can be regarded as a low-rank approximation problem with missing data, since it does not impose any group constraint. This allows for simpler optimization techniques like the Expectation Maximization (EM) algorithm put forward by Guerreiro and Aguiar [23]. Assuming the algorithm converges, the solution must be projected to $\mathrm{SE}(3)$ which renders it suboptimal. The logic behind the procedure is as follows. If all the block entries were available, then $X_{\mathcal{B}}$ would be the rank-4 matrix closest to $\widetilde{M}_{\mathcal{B}}$, in the least-squares sense. The solution would be obtained via SVD (or truncated SVD to achieve better performance). There is not however a SVD equivalent for matrices with missing entries, which is the case we are considering. EM tackles the problem in two alternating steps: estimate the missing entries of $\widetilde{M}_{\mathcal{B}}$ (E-step); find the rank-4 matrix closest to the complete matrix of the previous step (M-step).

Let $\mathcal{S}_4$ denote the space of rank-4 matrices and $\downarrow \mathcal{S}_4$ the operator that projects a matrix onto this space via SVD. For a given error tolerance ϵ, the EM algorithm is below.

Algorithm 1 Expectation-Maximization

procedure EM($A, \widetilde{M}_{\mathcal{B}}$, $M_{\mathcal{B}}^{\text{init}}$, ϵ)
  Initial estimate $\widehat{M}_{\mathcal{B}_0} \leftarrow \widehat{M}_{\mathcal{B}}^{\text{init}}$
  $k \leftarrow 1$
  E-STEP (estimate missing data): $\overline{M}_{\mathcal{B}_k} \leftarrow (A \otimes \mathbb{1}_{4\times4}) \circ \widetilde{M}_{\mathcal{B}} + \widehat{M}_{\mathcal{B}_{k-1}} \circ (1 - (A \otimes \mathbb{1}_{4\times4}))$
  M-STEP (estimate matrix): $\widehat{M}_{\mathcal{B}_k} \leftarrow \overline{M}_{\mathcal{B}_k} \downarrow \mathcal{S}_4$
  error $\leftarrow \left|\left|(A \otimes \mathbb{1}_{4\times4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{B}})\right|\right|_F^2$
  if error $< \epsilon$ **then**
    $\widehat{M}_{\mathcal{B}_k} \leftarrow \widehat{M}_{\mathcal{B}_k} \downarrow \mathrm{SE}(3)$
    return $\widehat{M}_{\mathcal{B}_k}$
  $k \leftarrow k + 1$
  goto E-STEP.

Although EM has a small computational cost per step for small matrices, convergence requires in general a large number of iterations, i.e. it is prone to *flatlining*. If we associate this with the time complexity of the SVD in $\mathcal{O}(N^3)$, where N is the matrix size, which must be carried out at each iteration, we arrive at the conclusion that P_{EM} is intractable for pose graphs of high order.

Damped Newton

Still in the realm low-rank matrix approximation with missing data (and no group constraints), Buchanan and Fitzgibbon [24] propose a second-order method immune to the *flatlining* problem in P_{EM}. Let ϵ denote the cost function,

$$\epsilon = \left|\left|(A \otimes \mathbb{1}_{4\times4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b})\right|\right|_F^2$$

And let x be the vector of variables, i.e. $x = \text{vectorize } (M_{\mathcal{C}})$, so that $\epsilon = \epsilon(x)$. The second-order Taylor series expansion of $\epsilon(x)$ can be written as

$$\epsilon(x + \delta x) \approx e(x) + \delta x^\top \nabla\epsilon + \frac{1}{2}\delta x^\top H \delta x \tag{2.16}$$

where ∇ is the gradient operator and H the Hessian matrix. If H is positive definite the quadratic approximation is convex and finding the minimum can be achieved by setting δx to the Newton's step:

$$\delta x \leftarrow -H^{-1}\nabla\epsilon \tag{2.17}$$

So far, this is just the application of Newton's method. The authors remarked however that H is prone to lose its positive-definiteness. To tackle this, they propose shifting its eigenvalues by adding a constant diagonal matrix λI. Therefore, the update in Eq. (2.17) is changed to

$$\delta x \leftarrow -(H + \lambda I)^{-1}\nabla\epsilon$$

This is known as the Damped Newton (DN) method since increasing λ restricts the step size. In partic-

ular, the suitability of the quadratic approximation in Eq. (2.16) is reflected by the spectrum of H. If the λ required to ensure positive-definitiness is large, Eq. (2.17) is a bad approximation and the step size should be smaller. The algorithm, as proposed by the Buchanan and Fitzgibbon [24] and adapted to our cost function is below:

Algorithm 2 Damped Newton

procedure DN($A, \widetilde{M_{\mathcal{B}}}, M_{\mathcal{B}}^{\text{init}}, \eta$)
  $\epsilon \leftarrow \left|\left|(A \otimes \mathbb{1}_{4\times4}) \circ (\widetilde{M_{\mathcal{B}}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b})\right|\right|_F^2 + \alpha \left|\left|M_{\mathcal{C}}\right|\right|_F^2$
  $x \leftarrow \text{vectorize}(M_{\mathcal{C}})$
  $\lambda \leftarrow 0.01$
  while $\epsilon(x) > \eta$ **do**
    $d \leftarrow \nabla\epsilon$
    $H \leftarrow \frac{D^2E}{Dx^2}$
    do
      $\lambda \leftarrow \lambda \times 10$
      $y \leftarrow x - (H + \lambda I)^{-1} d$
    while $\epsilon(y) > \epsilon(x)$
    $x \leftarrow y$
    $\lambda \leftarrow \lambda/10$
  return $\widehat{M_{\mathcal{C}}} \leftarrow \text{unvectorize}(x)$

Augmented Lagrangian multipliers

If on the one hand $\mathrm{P}_{\mathrm{ALM}}$ benefits from the robustness of the l^1-norm, on the other hand this norm is known for its poor convergence properties due to the nonconvexity it introduces. To counteract this effect, we can try to *convexify* the problem, e.g. by adding convex regularizer terms to the cost function, provided that its minimizer is not pulled away from the original solution.

This solution was put forward by Jin et al. [14], which in turn draws from the approach proposed by Zheng et al. [25] to solve LR-MC problems under the l^1-norm. In order to render the function:

$$\left|\left|(A \otimes \mathbb{1}_{4\times4}) \circ (\widetilde{M_{\mathcal{B}}} - M_{\mathcal{B}})\right|\right|_1$$

more convex, the authors propose adding the trace-norm regularizer $||M_{\mathcal{B}}||_*$ which is equal to the sum of the singular values of $M_{\mathcal{B}}$. Since $M_{\mathcal{B}} = M_{\mathcal{C}} M_{\mathcal{C}}^{-b}$ and $M_{\mathcal{C}}^{-b} M_{\mathcal{C}} = nI_4$ then $||M_{\mathcal{B}}||_* = ||M_{\mathcal{C}}||_*$. This term is convex. The regularized problem is then

$$\begin{aligned} \underset{M_{\mathcal{C}}}{\text{argmin}} \quad & \left|\left|(A \otimes \mathbb{1}_{4\times4}) \circ (\widetilde{M_{\mathcal{B}}} - M_{\mathcal{B}})\right|\right|_1 + \lambda \left|\left|M_{\mathcal{C}}\right|\right|_* \\ \text{s.t.} \quad & M_{\mathcal{C}}^{-b} M_{\mathcal{C}} = nI_4 \\ & M_{\mathcal{C}} M_{\mathcal{C}}^{-b} = M_{\mathcal{B}} \end{aligned} \tag{2.18}$$

and can be solved via ALM, for which the following augmented Lagrangian is used

$$f_{\text{ALM}}(M_{\mathcal{B}}, M_{\mathcal{C}}, L, \mu) = \underbrace{\left|\left|(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{B}})\right|\right|_1}_{\text{Cost function}} + \underbrace{\lambda \left|\left|M_{\mathcal{C}}\right|\right|_*}_{\text{Convex regularizer}} + \underbrace{\langle L, M_{\mathcal{B}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b} \rangle}_{\text{Lagrange multiplier term}} + \underbrace{\frac{\mu}{2} ||M_{\mathcal{B}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b}||_F^2}_{\text{Quadratic penalty term}}$$

where L is a matrix with Lagrange multipliers and μ the quadratic penalty parameter. The objective is then to minimize f_{ALM} while enforcing orthogonality of the columns of $M_{\mathcal{C}}$. The authors accomplish this using the Gauss-Seidel optimization strategy which alterates the optimization direction by iteratively solving different subproblems.

Eigendecomposition

When the cost function is formulated using the squared Frobenius norm, and under an orthogonality constraint on the columns of $M_{\mathcal{C}}$, the problem

$$\begin{aligned} &\underset{M_{\mathcal{C}}}{\operatorname{argmin}} \, ||(A \otimes \mathbb{1}_{4\times 4}) \circ (\widetilde{M}_{\mathcal{B}} - M_{\mathcal{C}} M_{\mathcal{C}}^{-b})||_F^2 \\ &\text{s.t. } M_{\mathcal{C}}^{-b} M_{\mathcal{C}} = n I_4 \end{aligned}$$

can be solved in closed-form. Let $\mathcal{L} \in \mathbb{R}^{n\times n}$ be the pose graph Laplacian. Letting $U\Sigma V^*$ denote the SVD of $(\mathcal{L} \otimes \mathbb{1}_{4\times 4}) \circ \widetilde{M}_{\mathcal{B}}$, the optimal solution $M_{\mathcal{C}}^*$ is given by

$$M_{\mathcal{C}}^* = U_{1,\dots,4} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix} \tag{2.19}$$

where u_i are the left singular vectors corresponding to the smallest singular values. The authors named this algorithm EIG-SE(3). When the measurements are exact, i.e. $\sigma_R = 0$ and $\sigma_t = 0$, it is possible to find $U_{1,\dots,4} \in \text{SE}(3)^n$, i.e. this solution is the ML estimate. However, when dealing with noisy measurements, assuming the noise model considered thus far, this solution drifts away from the global optimum. A proof sketch on why this happens is provided below.

Let $L = (\mathcal{L} \otimes \mathbb{1}_{4\times 4}) \circ \widetilde{M}_{\mathcal{B}}$. Additionally we will denote by $\widehat{M}_{\mathcal{C}}$ the solution obtained from EIG-SE(3). Then the solution verifies

$$L^\top L \widehat{M}_{\mathcal{C}} = \widehat{M}_{\mathcal{C}} \, \text{diag}(\sigma_1, \dots, \sigma_4)$$

and can be written more explicitly as

$$L^\top L \begin{bmatrix} \widehat{R}_1 & \widehat{t}_1 \\ 0_3^\top & 1 \\ \vdots & \vdots \\ \widehat{R}_n & \widehat{t}_n \\ 0_3^\top & 1 \end{bmatrix} = \begin{bmatrix} \widehat{R}_1 & \widehat{t}_1 \\ 0_3^\top & 1 \\ \vdots & \vdots \\ \widehat{R}_n & \widehat{t}_n \\ 0_3^\top & 1 \end{bmatrix} \operatorname{diag}(\sigma_1, \ldots, \sigma_4)$$

This equality can be separated into two independent equations:

$$L^\top L \begin{bmatrix} \widehat{R}_1^\top & 0_3 & \ldots & \widehat{R}_n^\top & 0_3 \end{bmatrix}^\top = \begin{bmatrix} \widehat{R}_1^\top & 0_3 & \ldots & \widehat{R}_n^\top & 0_3 \end{bmatrix}^\top \operatorname{diag}(\sigma_1, \sigma_2, \sigma_3) \tag{2.20a}$$

$$L^\top L \begin{bmatrix} \widehat{t}_1^\top & 1 & \ldots & \widehat{t}_n^\top & 1 \end{bmatrix}^\top = \sigma_4 \begin{bmatrix} \widehat{t}_1^\top & 1 & \ldots & \widehat{t}_n^\top & 1 \end{bmatrix}^\top \tag{2.20b}$$

For a set of rotation estimates $\{\widehat{R}_i\}_{i=1,\ldots,n}$, the ML translations (assuming the noise models presented in Section 2.2) are obtained by solving

$$\underset{\substack{t_i \in \mathbb{R}^3 \\ i \in \{1,\ldots,n\}}}{\operatorname{argmin}} \sum_{(i,j)\in E} ||\widetilde{t}_{ij} - t_i + \widehat{R}_i \widehat{R}_j^\top t_j||^2 \tag{2.21}$$

However, Eq. (2.20b) does not depend on $\widehat{R}_i$ and thus, in general, will not yield the ML solution. In other words, an improvement on EIG-SE(3) would be to use the rotation estimates obtained from the eigendecomposition in Eq. (2.20a) to solve the convex least-squares problem in Eq. (2.21).

In order suppress outliers, which may compromise the accuracy of this algorithm, the authors embedded EIG-SE(3) in an Iterative Reweighted Least Squares (IRLS) framework: IRLS EIG-SE(3). At each iteration, the global transformation estimates result from the SVD of $((W \circ \mathcal{L}) \otimes \mathbb{1}_{4\times 4}) \circ \widetilde{M}_{\mathcal{B}}$, where W is the weights matrix, updated iteratively so that relative transformations unexplained by the estimated model incur a larger penalty (smaller weight).

Chapter 3

Understanding the pose graph: Invariant subspaces

In this chapter, several new algebraic propositions pertaining to the block transformation matrices used in LR-MC are presented. Whilst these theoretical results pave the ground for our approach presented in the next chapter, the reader may skip ahead and refer back to them when the need arises.

3.1 Definition

In Section 2.3, it was demonstrated that the ground-truth relative pose matrix $M_{\mathcal{B}}$ has rank 4 for $M_{ij} \in \mathrm{SE}(3)$. This property has already been exploited in the past to tackle the global pose optimization problem via LR-MC. Notwithstanding, several other interesting properties about invariant subspaces of matrices with the same structure as that of $M_{\mathcal{B}}$ are worth mentioning. To render these properties more general, we will consider $M_{\mathcal{B}}$ to be any $(n \times m) \times (n \times m)$ real block matrix built as follows

$$M_{\mathcal{B}} = \begin{bmatrix} M_1 \\ \vdots \\ M_n \end{bmatrix} \begin{bmatrix} M_1^{-1} & \dots & M_n^{-1} \end{bmatrix}$$

for any $M_i \in \mathrm{GL}_m(\mathbb{R})$. In this case $M_{\mathcal{B}}$ has rank m. The results presented in the remainder of this section regard properties of invariant subpaces of $M_{\mathcal{B}}$. The definition of invariant subspace is now put forward.

Definition 3.1.1 (*Stewart et al.* [26])**.** *$\mathcal{X}$ is an invariant subspace of A if*

$$A\mathcal{X} \subseteq \mathcal{X}$$

Let X form a basis for $\mathcal{X}$, then there is a unique matrix L such that

$$AX = XL$$

3.2 Propositions

Proposition 3.2.1 (Krylov subspaces)**.** *Any Krylov subspace generated by $M_{\mathcal{B}}$ and any vector b has maximal dimension 2, i.e.* $\forall\, b \in \mathbb{R}^{m\times n} \;\; \forall\, r \in \mathbb{N}_0 \;\; \dim \mathcal{K}_r(M_{\mathcal{B}}, b) \leq 2$

Proof. The proof starts by showing that the column space of $M_{\mathcal{B}}$ is an invariant subspace of this matrix, which is can be demonstrated by induction. Consider the following equation

$$M_{\mathcal{B}}^{p+1} = n^p M_{\mathcal{B}} \tag{3.1}$$

For $p = 0$ it is obvious that the statement is true. Let us now assume that the statement is valid for a certain $p > 0$, i.e. $M_{\mathcal{B}}^{p+1} = n^p M_{\mathcal{B}}$ and prove that the formula is valid for $p + 1$:

$$M_{\mathcal{B}}^{p+2} = M_{\mathcal{B}} M_{\mathcal{B}}^{p+1} = n^p M_{\mathcal{B}}^2 \tag{3.2}$$

To calculate $M_{\mathcal{B}}^2$, we use the definition of the matrix block-product combined with the fact that $M_{\mathcal{B}_{ik}} M_{\mathcal{B}_{kj}} = M_i M_k^{-1} M_k M_j^{-1} = M_{\mathcal{B}_{ij}}$ for any k, to arrive at

$$M_{\mathcal{B}}^2 = \sum_{k=1}^{n} M_{\mathcal{B}_{ik}} M_{\mathcal{B}_{kj}} = \sum_{k=1}^{n} M_{\mathcal{B}_{ij}} = n M_{\mathcal{B}}$$

By plugging this result into Eq. (3.2) we get

$$M_{\mathcal{B}}^{p+2} = n^p M_{\mathcal{B}}^2 = n^p n M_{\mathcal{B}} = n^{p+1} M_{\mathcal{B}}$$

It is therefore proven, by induction that Eq. (3.1), is true for all $p \in \mathbb{N}_0$. The Krylov subspace of order r is defined as

$$\mathcal{K}_r(M_{\mathcal{B}}, b) = \operatorname{span}\{b, M_{\mathcal{B}} b, M_{\mathcal{B}}^2 b, \ldots, M_{\mathcal{B}}^{r-1} b\} = \operatorname{span}\{b, M_{\mathcal{B}} b, n M_{\mathcal{B}} b, \ldots n^{r-2} M_{\mathcal{B}} b\}$$

Every vector after the first one is linearly dependent on the previous vectors thus $\dim \mathcal{K}_r(M_{\mathcal{B}}, b) \leq 2$. □

Proposition 3.2.2 (Column space)**.** *The column space of $M_{\mathcal{B}}$ is an eigenspace and the corresponding eigenvalue is n, i.e.* $\forall\, v \in \operatorname{range} M_{\mathcal{B}} : M_{\mathcal{B}} v = nv$*. Therefore, the column space of $M_{\mathcal{B}}$ is an invariant subspace of the matrix.*

Proof. Direct consequence of Proposition 3.2.1. In fact, considering $p = 1$ we have $M_{\mathcal{B}} M_{\mathcal{B}} = n M_{\mathcal{B}}$ □

Proposition 3.2.3 (Spectrum)**.** *$M_{\mathcal{B}}$ has two distinct eigenvalues, 0 and n with algebraic multiplicities $\mu_{M_{\mathcal{B}}}(0) = n - m$ and $\mu_{M_{\mathcal{B}}}(n) = m$.*

Proof. Since $\operatorname{rank} M_{\mathcal{B}} = m$, then from the rank-nullity theorem

$$\operatorname{nullity} M_{\mathcal{B}} = n - m$$

which implies that

$$\mu_{M_{\mathcal{B}}}(0) \geq \gamma_{M_{\mathcal{B}}}(0) = n - m$$

where γ denotes the geometric multiplicity. Plus, we know that

$$\mu_{M_{\mathcal{B}}}(n) \geq \gamma_{M_{\mathcal{B}}}(n) = \operatorname{rank} M_{\mathcal{B}} = m$$

Since $\mu_{M_{\mathcal{B}}}(0) + \mu_{M_{\mathcal{B}}}(n) = n$ we have $\mu_{M_{\mathcal{B}}}(0) = n - m$ and $\mu_{M_{\mathcal{B}}}(n) = m$. □

To make the link with PGO where only a few blocks of $M_{\mathcal{B}}$ are available, consider now a simple, undirected and connected graph $\mathcal{G} = (V, E)$, with $|V| = n$, a degree matrix D and an adjacency matrix A. Let $\overline{M}_{\mathcal{B}}$ be any $(n \times m) \times (n \times m)$ real block matrix with the block entry i, j defined as

$$\overline{M}_{\mathcal{B}_{ij}} = \begin{cases} M_i M_j^{-1} & \text{if } (i,j) \in E \\ I_m & \text{if } i = j \\ 0_m & \text{otherwise} \end{cases}$$

for any $M_i \in \mathrm{GL}_m(\mathbb{R})$. $\overline{M}_{\mathcal{B}_{ij}} = M_i M_j^{-1}$ corresponds to a linear transformation associated with the edge $(i, j) \in E$. Its inverse, $\overline{M}_{\mathcal{B}_{ji}}$ is obviously the transformation associated with the same edge in the opposite direction.

Proposition 3.2.4 (Invariant subspaces of $(\mathcal{L} \otimes \mathbb{1}_{m\times m}) \circ \overline{M}_{\mathcal{B}}$)**.** *Let (λ, v) be an eigenpair of $\mathcal{L}$ with algebraic multiplicity μ and geometric multiplicity γ. Then λ is an eigenvalue of $(\mathcal{L} \otimes \mathbb{1}_{m\times m}) \circ \overline{M}_{\mathcal{B}}$ with algebraic and geometric multiplicites $m\mu$ and $m\gamma$, respectively. The respective eigenspace is determined by v.*

Proof. The eigenvalue equation for the Laplacian is as follows

$$\mathcal{L}v = \lambda v$$

where $v = [v_1 \dots v_n]^\top$. Let l_{ij} define the elemental entry i, j of $\mathcal{L}$. Then,

$$\sum_{j \in E(v_i) \cup \{i\}} l_{ij} v_j = \lambda v_i$$

Let us define a $(n \times m) \times m$ block column V as follows

$$V = \begin{bmatrix} v_1 M_1 \\ \vdots \\ v_n M_n \end{bmatrix}$$

Then,

$$\big((\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \overline{M}_{\mathcal{B}}\big) V = \sum_{j \in E(v_i) \cup \{i\}} l_{ij} M_i M_j^{-1} M_j v_j$$

which, by eliminating $M_j^{-1}M_j$ and placing M_i outside the sum, becomes

$$\big((\mathcal{L}\otimes \mathbb{1}_{m\times m})\circ \overline{M}_{\mathcal{B}}\big)V = M_i \sum_{j\in E(v_i)\cup\{i\}} l_{ij}v_j = \lambda v_i M_i$$

Therefore,

$$\big((\mathcal{L}\otimes \mathbb{1}_{3\times 3})\circ \overline{M}_{\mathcal{B}}\big)V = \lambda V$$

By construction $\operatorname{rank} V = m$, thus the geometric multiplicity of λ in the previous equation is $m\gamma$. The total geometric multiplicity of $(\mathcal{L}\otimes \mathbb{1}_{3\times 3})\circ \overline{M}_{\mathcal{B}}$ is equal to $m\times n$. The algebraic multiplicity in the previous equation is at least $m\gamma$. Since the total algebraic multiplicity must be $m\times n$ this implies that both multiplicities are the same for each eigenvalue. □

This proposition allows us to assert the following about $(\mathcal{L}\otimes \mathbb{1}_{m\times m})\circ \overline{M}_{\mathcal{B}}$:

- Its spectrum is determined by the spectrum of $\mathcal{L}$, i.e. it is not influenced by the transformation matrices.
- All the eigenvalues verify $\mu_\lambda \geq m$.

Proposition 3.2.5 (Consequence of Proposition 3.2.4)**.** *If $\mathcal{G}$ is connected then the rank of $(\mathcal{L}\otimes \mathbb{1}_{m\times m})\circ \overline{M}_{\mathcal{B}}$ is $n-m$ and the column space of this matrix is an invariant subspace.*

Proof. Since $\mathcal{G}$ is connected then $\operatorname{rank}\ \mathcal{L} = n-1$ (Appendix A.2). This implies that $\mathcal{L}$ has a null eigenvalue $\lambda_0 = 0$ with $\gamma_0(\mathcal{L}) = 1$. Since the Laplacian is symmetric then $\mu_0(\mathcal{L}) = \gamma_0(\mathcal{L}) = 1$. According to Proposition 3.2.4 we have that $(\mathcal{L}\otimes \mathbb{1}_{m\times m})\circ \overline{M}_{\mathcal{B}}$ has a null eigenvalue with

$$\mu_0((\mathcal{L}\otimes \mathbb{1}_{m\times m})\circ \overline{M}_{\mathcal{B}}) = \gamma_0((\mathcal{L}\otimes \mathbb{1}_{m\times m})\circ \overline{M}_{\mathcal{B}}) = m$$

The rank-nullity theorem then allows us to arrive at the conclusion that $\operatorname{rank}\ (\mathcal{L}\otimes \mathbb{1}_{m\times m})\circ \overline{M}_{\mathcal{B}} = n-m$. The fact that the column space of this matrix is an invariant subspace is a direct consequence of Proposition 3.2.4, by choosing the eigenpair $(\lambda = 0, v = \mathbb{1}_{n\times 1})$ of $\mathcal{L}$. □

The propositions presented thus far are applicable if the non-null blocks of $\overline{M}_{\mathcal{B}}$ belong to $\mathrm{SE}(3)$ or $\mathrm{SO}(3)$, provided that the pairwise transformations are exact, i.e. assuming a noise model with $\sigma_t = 0$ and $\sigma_R = 0$. However, in most applications this is not a valid assumption, i.e. measurements are noisy and as a consequence the theoretical results aforementioned no longer hold.

We expect the spectrum and invariant subspaces of $(\mathcal{L}\otimes \mathbb{1}_{4\times 4})\circ \overline{M}_{\mathcal{B}}$ to be different from those of its noise-riddled counterpart $(\mathcal{L}\otimes \mathbb{1}_{4\times 4})\circ \widetilde{M}_{\mathcal{B}}$. The former can be referred to as the ground-truth and the latter as the actual observation. To conclude this Chapter, we will show that, when considering a symmetric block matrix $\widetilde{R}_{\mathcal{B}}$ with the same structure as that of $\widetilde{M}_{\mathcal{B}}$ but consisting of 3×3 blocks in $\mathrm{SO}(3)$, matrix perturbation theory allows us to assess the quality of the measurements by means of spectra comparison.

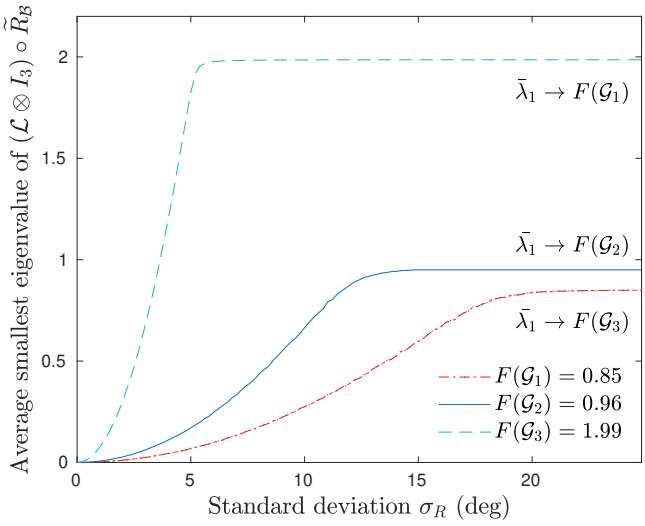


Figure 3.1: Average of the smallest eigenvalue of $(\mathcal{L} \otimes \mathbb{1}_{3\times3}) \circ \widetilde{R}_{\mathcal{B}}$ for three synthetic rotation graphs $\mathcal{G}_i$ with $|V| = 553$ and different algebraic connectivities (*Fiedler* values) $F(\mathcal{G}_i)$, as the standard deviation of the measurements increases (assuming a Langevin distribution and i.i.d. measurements). Each point of each curve corresponds to an average of 10 eigenvalue samples.

Conjecture 3.2.1 (Spectrum)**.** *The expected value of the smallest eigenvalue of* $(\mathcal{L} \otimes \mathbb{1}_{3\times3}) \circ \widetilde{R}_{\mathcal{B}}$*, hereafter denoted by* λ_1*, converges monotonically to the Fiedler value* $F(\mathcal{G})$ *of the graph, for increasing* σ_R*. Mathematically this translates to:*

$$\boxed{\lim_{\sigma_R \to \pi} \overline{\lambda_1}((\mathcal{L} \otimes \mathbb{1}_{3\times3}) \circ \widetilde{R}_{\mathcal{B}}) = F(\mathcal{G})}$$

There is no theoretical proof for this conjecture other than empirical evidence as represented in Fig. 3.1.

When dealing with rotation matrices, regardless of σ_R, the block matrix $(\mathcal{L} \otimes \mathbb{1}_{3\times3}) \circ \widetilde{R}_{\mathcal{B}}$ is symmetric. This matrix can be decomposed as

$$(\mathcal{L} \otimes \mathbb{1}_{3\times3}) \circ \widetilde{R}_{\mathcal{B}} = H + P$$

where $H = (\mathcal{L} \otimes \mathbb{1}_{3\times3}) \circ R_{\mathcal{B}}$ is a symmetric matrix with the Laplacian-weighted true rotations and P is a perturbation matrix, also symmetric. The eigenvalues of $(\mathcal{L} \otimes \mathbb{1}_{3\times3}) \circ \widetilde{R}_{\mathcal{B}}$ can be regarded as the perturbed eigenvalues of H, which according to Proposition 3.2.4 are in the spectrum of $\mathcal{L}$. Let $\lambda_i(A)$ denote the i-th eigenvalue of a matrix A. Using Weyl's inequalities [27], together with a special case of Holders' inequality and the fact that P is symmetric we have

$$|\lambda_i(\mathcal{L}) - \lambda_{3i}((\mathcal{L} \otimes \mathbb{1}_{3\times3}) \circ \widetilde{R}_{\mathcal{B}})| \leq ||P||_\infty, \quad \forall i \in \{1, \ldots, n\}$$

In summary, comparing the spectrum of $\mathcal{L}$ to that of $(\mathcal{L} \otimes \mathbb{1}_{3\times3}) \circ \widetilde{R}_{\mathcal{B}}$ allows for an assessment of the quality and consistency of the pairwise rotation measurements.

Chapter 4

Unified ML framework for PGO

In this Chapter, we present our approach to solving MA by combining the formulations presented in Chapter 2, i.e. the statistical MLE model and LR-MC, and derive a closed-form solution close to the global optimum. The crux of our approach is presented in Section 4.1. There, we put forward our objective function: the log-likelihood function borrowed from Section 2.2 and we show that, under a high Signal-to-Noise Ratio (SNR), its maximum can be attained by optimizing a rotations-only term and using that estimate to seek the optimal translations.

Both optimization problems are solved in closed-form via eigendecomposition in Sections 4.2 and 4.3, respectively. In order to do so, we use the the Propositions presented in Chapter 3. Finally, to handle cases where our noise model assumptions are not verified, we propose an iterative method for dealing with outlier measurements. Despite the similarities between our approach and the closed-form solution EIG-SE(3) presented in Section 2.3, our solution is faster, more accurate and was derived independently, without knowledge of the latter.

4.1 Observation model

Our formulation is based on the observation model presented in Section 2.2, where the MLE approach to MA was set forth. It comprises i.i.d. translation measurements following a Gaussian distribution and i.i.d. rotation measurements following an isotropic Langevin distribution. The log-likelihood function here considered is thus that of Eq. (2.6):

$$\boxed{\log L(\theta, y) = -\frac{1}{2\sigma_t^2} \sum_{(i,j)\in E} ||\widetilde{t}_{ij} - t_i + R_i R_j^\top t_j||^2 + \frac{1}{\sigma_R^2} \sum_{(i,j)\in E} \operatorname{tr} \widetilde{R}_{ij} R_j R_i^\top} \tag{4.1}$$

This objective function is, as mentioned in Section 2.2 not separable in the translation and rotation terms. Notwithstanding, provided there is a high Signal-to-Noise Ratio (SNR), an approximate optimum can be obtained by maximizing first the second term on the right side of Eq. (4.1), which is a function of the rotations only, and then using these rotation estimates to compute the approximate optimal translations. In order to show that this is a reasonable approximation we start by rewriting the log-likelihood as

a sum of two functions, g and h. The goal is to group together in g all the rotations to be estimated.

$$\log L(\theta, y) = \underbrace{\sum_{(i,j)\in E} \left(-\frac{1}{\sigma_t^2}(\widetilde{t}_{ij}^{\top} - t_i^{\top}) R_i R_j^{\top} t_j + \frac{1}{\sigma_R^2} \operatorname{tr} \widetilde{R}_{ij} R_j R_i^{\top} \right)}_{g(R_i, t_i)} + h(t_i) \tag{4.2}$$

The full derivation of Eq. (4.2) is available in Appendix B.3. As indicated by the curly brace, let $g(R_i, t_i)$ denote the left term. We have

$$\begin{aligned} g(R_i, t_i) &= \sum_{(i,j)\in E} \left(-\frac{1}{\sigma_t^2} (\widetilde{t}_{ij} - t_i)^{\top} \widetilde{R}_{ji} \widetilde{R}_{ij} R_i R_j^{\top} t_j + \frac{1}{\sigma_R^2} \operatorname{tr} \widetilde{R}_{ij} R_j R_i^{\top} \right) \\ &= \sum_{(i,j)\in E} \operatorname{tr} \left(\left(\frac{1}{\sigma_t^2} t_j (t_i - \widetilde{t}_{ij})^{\top} \widetilde{R}_{ji} + \frac{1}{\sigma_R^2} I_3 \right) \widetilde{R}_{ij} R_j R_i^{\top} \right) \end{aligned}$$

Given the set of optimal translations $\{t_i^*\}_{i=1,\dots,n}$, i.e. the translations that maximize $\log\ L(\theta, y)$, the problem of finding the optimal rotations can be stated as follows

$$\underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmax}}\ g(R_i, t_i^*) = \underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmax}} \sum_{(i,j)\in E} \operatorname{tr} \left(\underbrace{\left(\frac{1}{\sigma_t^2} t_j^* (t_i^* - \widetilde{t}_{ij})^{\top} \widetilde{R}_{ji} + \frac{1}{\sigma_R^2} I_3 \right)}_{U \Sigma V^{\top}} \widetilde{R}_{ij} R_j R_i^{\top} \right)$$

where $U\Sigma V^{\top}$ corresponds to the singular value decomposition of the term indicated by the curly brace. Using the cyclic property of the trace operator we have

$$\underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmax}} \sum_{(i,j)\in E} \operatorname{tr} \left(U \Sigma V^{\top} \widetilde{R}_{ij} R_j R_i^{\top} \right) = \underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmax}} \sum_{(i,j)\in E} \operatorname{tr} \left(\Sigma \underbrace{V^{\top} \widetilde{R}_{ij} R_j R_i^{\top} U}_{\text{orthonormal matrix}} \right)$$

An upper bound for the trace in the previous expression can be found by using the fact that Σ is a non-negative diagonal matrix multiplied by an orthonormal matrix.

$$\operatorname{tr} \Sigma V^{\top} \widetilde{R}_{ij} R_j R_i^{\top} U \leq \operatorname{tr} \Sigma \tag{4.3}$$

In order to attain, or approximate this upper bound an hypothesis is now presented.

Hypothesis 4.1.1 (High SNR)**.** *Let t_i^* be the optimal translation corresponding to pose i. We are going to assume*

$$\boxed{\forall\ (i,j) \in E\ :\ \widetilde{R}_{ji} t_j^* \overset{H4.1.1}{\approx} t_i^* - \widetilde{t}_{ij}}$$

It is easy to see that the approximation becomes an equality when the measurements are equal to the ground truth (noiseless case without outliers). With this assumption we have

$$\frac{1}{\sigma_t^2} t_j^* (t_i^* - \widetilde{t}_{ij})^{\top} \widetilde{R}_{ji} + \frac{1}{\sigma_R^2} I_3 \overset{H4.1.1}{\approx} \frac{1}{\sigma_t^2} t_j^* t_j^{*\top} + \frac{1}{\sigma_R^2} I_3$$

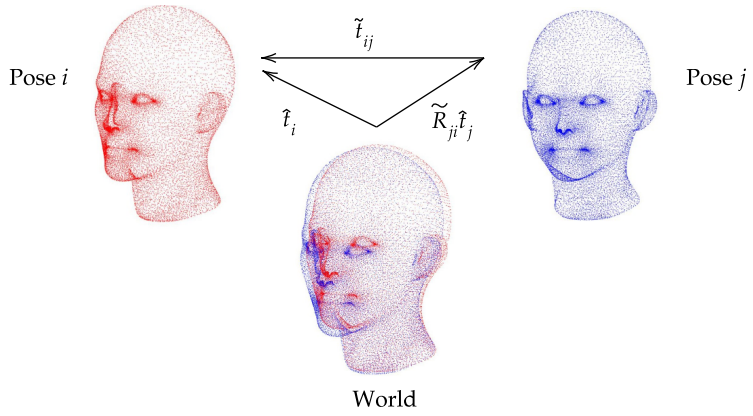


Figure 4.1: Representation of Hypothesis 4.1.1. The point clouds were purposefully registered with an angular error of 4 deg to show that, in Computer Vision, a high SNR can be assumed. The translations $\hat{t}_i$, $\widetilde{t}_{ij}$ and $\widetilde{R}_{ij}\hat{t}_j$ are in the coordinate system of pose i.

which corresponds to a symmetric matrix. Thus, in the SVD previously defined we have $UV^\top \overset{H4.1.1}{\approx} I_3$.

Under this hypothesis, the bound in Eq. (4.3) can be attained by setting $\widetilde{R}_{ij}R_jR_i^\top$ to $VU^\top = I$. Without any knowledge of the translations, the following problem should therefore yield a solution close to the optimal rotations, under a high SNR hypothesis:

$$\underset{\substack{R_i\in\mathrm{SO}(3)\\ i\in\{1,\dots,n\}}}{\operatorname{argmax}} \sum_{(i,j)\in E} \operatorname{tr}\left(\widetilde{R}_{ij}R_jR_i^\top\right) = \underset{\substack{R_i\in\mathrm{SO}(3)\\ i\in\{1,\dots,n\}}}{\operatorname{argmin}} \sum_{(i,j)\in E} ||\widetilde{R}_{ij} - R_iR_j^\top||_F^2 \tag{4.4}$$

which corresponds to a non-linear multiple rotation averaging problem under the chordal metric. Should these rotations be close to the optimal ones, we can use them to then solve for the translations, which is a convex least-squares problem. To summarize, assuming a certain generative noise model and under a high SNR hypothesis, the log-likelihood function we used to formulate MA can be separated in rotations and translations (Fig. 4.2), which can be optimized for in this order. Section 4.2 deals with the former while Section 4.3 tackles the latter.

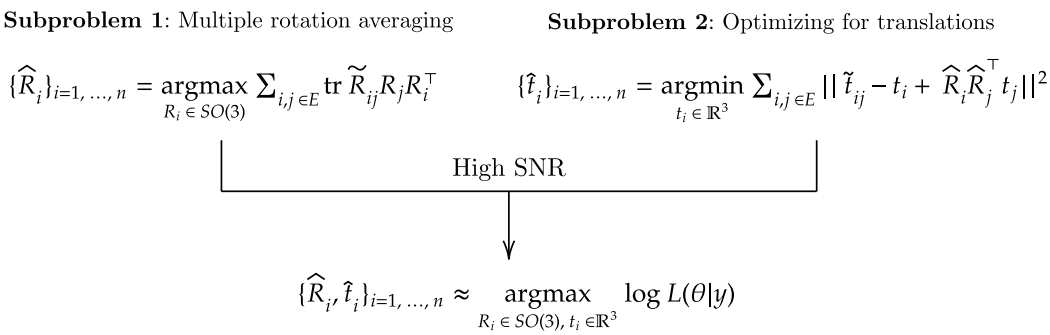


Figure 4.2: Proposed model combining two subproblems under a high SNR hypothesis.

4.2 Solving for rotations

The multiple rotation averaging problem is now considered. In order to solve it, we will rewrite the objective function as a block matrix product and use the propositions on invariant subspaces from Section 3.1 to arrive at a solution. As per the last section, multiple rotation averaging can be formulated as a constrained non-concave maximization problem:

$$\boxed{\underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmax}} \sum_{(i,j)\in E} \operatorname{tr} \widetilde{R}_{ij} R_j R_i^\top = \underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmax}} f_{\mathrm{ML}}} \tag{4.5}$$

where f_{ML} denotes the objective function. Let $\widetilde{R}_{\mathcal{B}} \in \mathbb{R}^{3n\times 3n}$ be a block matrix containing the measured rotations:

$$\widetilde{R}_{\mathcal{B}_{ij}} = \begin{cases} \widetilde{R}_{ij} & \text{if } (i,j) \in E \\ I_3 & \text{if } i = j \\ 0_3 & \text{otherwise} \end{cases}$$

Similarly, let $R_{\mathcal{C}} \in \mathrm{SO}(3)^n$ be a $3n \times 3$ block column defined as

$$R_{\mathcal{C}_i} = \begin{bmatrix} R_1^\top & \dots & R_n^\top \end{bmatrix}^\top, \quad R_i \in \mathrm{SO}(3)$$

Then, we have the following equality

$$\begin{aligned} \operatorname{tr} R_{\mathcal{C}}^\top \widetilde{R}_{\mathcal{B}} R_{\mathcal{C}} = \operatorname{tr} \sum_{i=1}^{n} R_i^\top \left(R_i + \sum_{j\in E(v_i)} \widetilde{R}_{ij} R_j + \sum_{j\,:\, i\in E(v_j)} \widetilde{R}_{ji}^\top R_j \right) &= \operatorname{tr} \left(2 \sum_{(i,j)\in E} R_i^\top \widetilde{R}_{ij} R_j + |V| I_3 \right) \\ &= 2 f_{\mathrm{ML}} + 3|V| \end{aligned}$$

Therefore, maximizing the trace of $R_{\mathcal{C}}^\top \widetilde{R}_{\mathcal{B}} R_{\mathcal{C}}$ yields the solution to the problem from Eq. (4.5), i.e.

$$\boxed{\underset{\substack{R_i \in \mathrm{SO}(3) \\ i \in \{1,\dots,n\}}}{\operatorname{argmax}} f_{\mathrm{ML}} = \underset{R_{\mathcal{C}} \in \mathrm{SO}(3)^n}{\operatorname{argmax}} \operatorname{tr} R_{\mathcal{C}}^\top \widetilde{R}_{\mathcal{B}} R_{\mathcal{C}}}$$

In other words, the block matrix formulation proposed by Arrigoni et al. [13], which was presented in Section 2.3 is equivalent to MLE when dealing with rotations only and assuming i.i.d. measurements. Note that due to the group constraint $R_{\mathcal{C}} \in \mathrm{SO}(3)^n$ we have $\operatorname{tr} R_{\mathcal{C}}^\top (D \otimes I_3 + I_{3n}) R_{\mathcal{C}} = 6|E| + 3|V|$. This constant term can be introduced in the objective function,

$$\begin{aligned} \underset{R_{\mathcal{C}} \in \mathrm{SO}(3)^n}{\operatorname{argmax}} \operatorname{tr} R_{\mathcal{C}}^\top \widetilde{R}_{\mathcal{B}} R_{\mathcal{C}} &= \underset{R_{\mathcal{C}} \in \mathrm{SO}(3)^n}{\operatorname{argmax}} \operatorname{tr} R_{\mathcal{C}}^\top \widetilde{R}_{\mathcal{B}} R_{\mathcal{C}} - \underbrace{\operatorname{tr} R_{\mathcal{C}}^\top (D \otimes I_3 + I_{3n}) R_{\mathcal{C}}}_{\text{constant}} \\ &= \underset{R_{\mathcal{C}} \in \mathrm{SO}(3)^n}{\operatorname{argmax}} \operatorname{tr} R_{\mathcal{C}}^\top \left(\widetilde{R}_{\mathcal{B}} - (D \otimes I_3 + I_{3n}) \right) R_{\mathcal{C}} \end{aligned}$$

Moreover, the term in parentheses can be written using the Laplacian since

$$\widetilde{R}_{\mathcal{B}} - (D \otimes I_3 + I_{3n}) = -(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}}$$

Consequently, the multiple rotation averaging problem can be stated as:

$$\boxed{\underset{R_{\mathcal{C}} \in \mathrm{SO}(3)^n}{\operatorname{argmin}} \ \operatorname{tr} R_{\mathcal{C}}^{\top} \left((\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}} \right) R_{\mathcal{C}}} \tag{4.6}$$

The purpose of introducing the Laplacian is that in the noiseless case, according to Proposition 3.2.4, the spectrum and invariant subspaces of $(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ R_B$ are determined by those of $\mathcal{L}$. Furthermore, when dealing with noisy measurements, according to Conjecture 3.2.1 the expected value of the smallest eigenvalue of this random matrix is a monotonic function of the standard deviation of the measurements.

Solving the problem from Eq. (4.6) The matrix $(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_B$ is symmetric. Let us denote its SVD by $U\Sigma U^{\top}$, where U is a $3n \times 3n$ orthonormal matrix and Σ a $3n \times 3n$ diagonal matrix. The following notation will be used for the remainder of this chapter:

$$U^{\top} = \underbrace{\begin{bmatrix} U_{11}^{\top} & \dots & U_{1n}^{\top} \\ \vdots & \ddots & \vdots \\ U_{n1}^{\top} & \dots & U_{nn}^{\top} \end{bmatrix}}_{U_{ij} \triangleq 3\times 3 \text{ block}} = \underbrace{\begin{bmatrix} U_1^{\top} \\ \vdots \\ U_n^{\top} \end{bmatrix}}_{U_i^{\top} \triangleq 3\times 3n \text{ block line}} \qquad \Sigma = \begin{bmatrix} |\lambda_1| & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & |\lambda_{3n}| \end{bmatrix} = \underbrace{\begin{bmatrix} \Sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \Sigma_n \end{bmatrix}}_{\Sigma_i \triangleq 3\times 3 \text{ block}}$$

with $0 \leq |\lambda_1| \leq \cdots \leq |\lambda_{3n}|$. The problem can thus be written as

$$\boxed{\underset{R_{\mathcal{C}} \in \mathrm{SO}(3)^n}{\operatorname{argmin}} \ \operatorname{tr} R_{\mathcal{C}}^{\top} U \Sigma U^{\top} R_{\mathcal{C}}}$$

Factorizing the diagonal matrix, it is possible to identify the following Frobenius norm

$$\boxed{\underset{R_{\mathcal{C}} \in \mathrm{SO}(3)^n}{\operatorname{argmin}} \ \operatorname{tr} \left(\Sigma^{\frac{1}{2}} U^{\top} R_{\mathcal{C}} \right)^{\top} \left(\Sigma^{\frac{1}{2}} U^{\top} R_{\mathcal{C}} \right) = \underset{R_{\mathcal{C}} \in \mathrm{SO}(3)^n}{\operatorname{argmin}} \ \left|\left| \Sigma^{\frac{1}{2}} U^{\top} R_{\mathcal{C}} \right|\right|_F^2} \tag{4.7}$$

which we can use to find a lower bound for the rotation averaging problem. Consider the following set of subspaces of $\mathbb{R}^{3\times n}$

$$W_i = \operatorname{span} \Sigma_i^{\frac{1}{2}} U_i$$

It is easy to see that $\Pi_{i=1}^{n} W_i = \mathbb{R}^{3n}$. All the subspaces W_i are orthogonal to eachother, consequently Eq. (4.7) is essentially projecting the column space of $R_{\mathcal{C}}$ onto n orthogonal spaces, i.e.

$$\left|\left|\Sigma^{\frac{1}{2}}U^\top R_{\mathcal{C}}\right|\right|_F^2 = \underbrace{\left|\left|\Sigma_1^{\frac{1}{2}}U_1^\top R_{\mathcal{C}}\right|\right|_F^2}_{\text{projection onto } W_1} + \cdots + \underbrace{\left|\left|\Sigma_n^{\frac{1}{2}}U_n^\top R_{\mathcal{C}}\right|\right|_F^2}_{\text{projection onto } W_n}$$

Due to the orthogonality constraint of the columns of $R_{\mathcal{C}}$ we can bound each term as follows

$$0 \leq \left|\left|\Sigma_i^{\frac{1}{2}}U_i^\top R_{\mathcal{C}}\right|\right|_F^2 \leq n\sum_{k=1}^{3}|\lambda_{3(i-1)+k}| \quad \text{for} \quad i = 1, \ldots, n$$

The cost function can be bounded as well,

$$\boxed{\underbrace{n\sum_{k=1}^{3}|\lambda_k|}_{\text{if } R_{\mathcal{C}} \in W_1} \leq \left|\left|\Sigma^{\frac{1}{2}}U^\top R_{\mathcal{C}}\right|\right|_F^2 \leq \underbrace{n\sum_{k=0}^{3}|\lambda_{3n-k}|}_{\text{if } R_{\mathcal{C}} \in W_n}}$$

Ideally, the estimate should be $\widehat{R}_{\mathcal{C}} \in W_1$ so as to attain the lower bound. It is easy to see that it is possible to obtain such estimate when $\sigma_R = 0$ (the solution lies in the eigenspace spanned by U_1 as per Proposition 3.2.4).

If we consider, however, the case where $\nexists\, R_{\mathcal{C}} \in \mathrm{SO(3)^n} : \left|\left|U_1 R_{\mathcal{C}}\right|\right|_F = \left|\left|R_{\mathcal{C}}\right|\right|_F$, according to matrix perturbation theory and under the hypothesis of small σ_R, the solution $R_{\mathcal{C}}^*$ should nonetheless be such that $\left|\left|U_1 R_{\mathcal{C}}\right|\right|_F \approx \left|\left|R_{\mathcal{C}}\right|\right|_F$. This is due to the fact that the invariant subspaces of a symmetric matrix are robust to small perturbations. Consequently, it is reasonable to say that we want to find $R_{\mathcal{C}}$ for which $\left|\left|U_1 R_{\mathcal{C}}\right|\right|_F$ is maximal. In order to do so, we approximate the solution of the problem

$$\underset{R_{\mathcal{C}} \in \mathrm{SO(3)}^n}{\operatorname{argmax}} \; ||U_1^\top R_{\mathcal{C}}||_F^2$$

by $\widehat{R}_{\mathcal{C}} = U_{1\,\downarrow \mathrm{SO(3)}^n}$. This approximation can be further refined. In fact, the sequence of rotation estimates $\{R_{\mathcal{C}}^k\}_k$, with $R_{\mathcal{C}}^0 = U_{1\downarrow \mathrm{SO(3)}^n}$, defined by the following fixed-point iteration

$$R_{\mathcal{C}}^{k+1} = \left(\widetilde{R}_{\mathcal{B}} R_{\mathcal{C}}^k\right)_{\downarrow \mathrm{SO(3)}^n}$$

converges to a stationary point of f_{ML} on $\mathrm{SO(3)}^n$. In other words, for $i \in \{1, \ldots, n\}$ and provided k is large enough, $\mathrm{grad}_{R_i^k}(\psi) \approx 0$. In practice, we found that, despite the small cost per iteration, this refinement is unnecessary assuming a high SNR, as our approximation yields satisfactory results at a fraction of the computational cost.

In summary, we propose solving rotation averaging by building the block matrix $\widetilde{R}_{\mathcal{B}}$, normalizing it with the graph Laplacian to obtain $(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}}$, computing three eigenvectors corresponding to the three smallest eigenvalues (in absolute value), and projecting them to $\mathrm{SO(3)}^n$. The validity of this approximation will depend on the magnitude of the eigenvalues.

4.3 Solving for translations

Now we will deal with the translations of the log-likelihood. Here it is assumed that rotation averaging has been dealt with and thus we have a set of rotation estimates $\{\widehat{R}_i\}_{i=1,\dots,n}$. The likelihood function used in our model is

$$\log L(\theta, y) = -\frac{1}{2\sigma_t^2} \sum_{(i,j)\in E} ||\widetilde{t}_{ij} - t_i + R_i R_j^\top t_j||^2 + \frac{1}{\sigma_R^2} \sum_{(i,j)\in E} \text{tr}\, \widetilde{R}_{ij} R_j R_i^\top$$

As shown in Appendix B.4, the derivative of this function w.r.t. to the k-th translation t_k is given by

$$\boxed{\frac{\partial}{\partial t_k} \log L(\theta, y) = \frac{2}{\sigma_t^2} \left(\sum_{j\in E(v_k)} \left(R_k R_j^\top t_i + \frac{1}{2}(\widetilde{t}_{kj} + R_k R_j^\top \widetilde{R}_{jk} \widetilde{t}_{kj}) \right) - \deg(v_k) t_k \right)^\top}$$

Since the log-likelihood is convex in the translation terms, the solution to Eq. (4.8) is the set of translations which, for a given set of rotations, maximizes the log-likelihood.

$$\frac{\partial}{\partial t_k} \log L(\theta, y)|_{R_i = \widehat{R}_i} = 0 \tag{4.8}$$

We show that this solution is the invariant subspace of a matrix. Let $\{\widehat{R}_i\}_{i=1\dots n}$ be a set of rotation estimates, e.g. the ones estimated using the multiple rotation averaging strategy previously presented. We will introduce a new relative pose matrix $M'_{\mathcal{B}}$ which we build as follows

$$M'_{\mathcal{B}_{ij}} = \begin{cases} \begin{bmatrix} \widehat{R}_i \widehat{R}_j^\top & \frac{1}{2}(\widetilde{t}_{ij} + \widehat{R}_i \widehat{R}_j^\top \widetilde{R}_{ji} \widetilde{t}_{ij}) \\ 0_3^\top & 1 \end{bmatrix} & \text{if } v_j \in E(v_i) \\ 0_{4\times 4} & \text{if } v_j \notin E(v_i) \end{cases}$$

This matrix has the same structure as the block matrix $\widetilde{M_{\mathcal{B}}}$ introduced in Section 2.3, but the pairwise relative rotation observations were replaced by their estimates $\widehat{R}_i \widehat{R}_j^\top$ and the observed translations by a rotation-averaged translation. The invariant subspace considerations previously presented still hold in this case. There is a block column matrix $M_{\mathcal{C}} \in \mathbb{R}^{4n\times 4}$,

$$M_{\mathcal{C}} = \begin{bmatrix} \widehat{R}_1 & \widehat{t}_1 \\ 0_3^\top & 1 \\ \vdots & \vdots \\ \widehat{R}_n & \widehat{t}_n \\ 0_3^\top & 1 \end{bmatrix}$$

such that

$$(\mathcal{L} \otimes \mathbb{1}_{4\times 4}) \circ M'_{\mathcal{B}} M_{\mathcal{C}} = 0$$

this implies that for $i \in \{1, \dots, n\}$ we have

$$\boxed{\sum_{j \in E(v_i) \cup \{i\}} \left(\widehat{R}_i \widehat{R}_j^\top \widehat{t}_j + \frac{1}{2} \left(\widetilde{t}_{ij} + \widehat{R}_i \widehat{R}_j^\top \widetilde{R}_{ji} \widetilde{t}_{ij} \right) \right) - (\deg(v_i) + 1) \widehat{t}_i = 0}$$

Therefore,

$$\frac{\partial}{\partial t_k} \log L(\theta | y)|_{R_i = \widehat{R}_i, t_i = \widehat{t}_i} = 0$$

Since the log-likelihood is convex in t_i, this is the global minimum (for fixed rotations). In conclusion, the optimal translations in homogeneous coordinates for a given set of rotations are in the nullspace of $(\mathcal{L} \otimes \mathbb{1}_{4\times 4}) \circ M'_{\mathcal{B}}$. Note that this result is exact, i.e. by construction $M'_{\mathcal{B}}$ always has a null eigenvalue with geometric multiplicity 4.

4.4 Quasi-optimality of our solution

Let $\widehat{R}_i$ and $\hat{t}_i$ denote the i-th pose rotation and translation estimates, respectively. Two conditions are sufficient for these solutions to be close to the global optimum.

1. Hypothesis 4.1.1 is verified, i.e. $\forall\ (i,j) \in E : \widetilde{R}_{ji} \hat{t}_j \approx \hat{t}_i - \widetilde{t}_{ij}$
2. The rotation averaging solution is quasi-optimal, i.e. $\operatorname{tr} \widehat{R}_{\mathcal{C}}^\top \left((\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}} \right) \widehat{R}_{\mathcal{C}} \approx n \sum_{i=1}^{3} |\lambda_i|$.

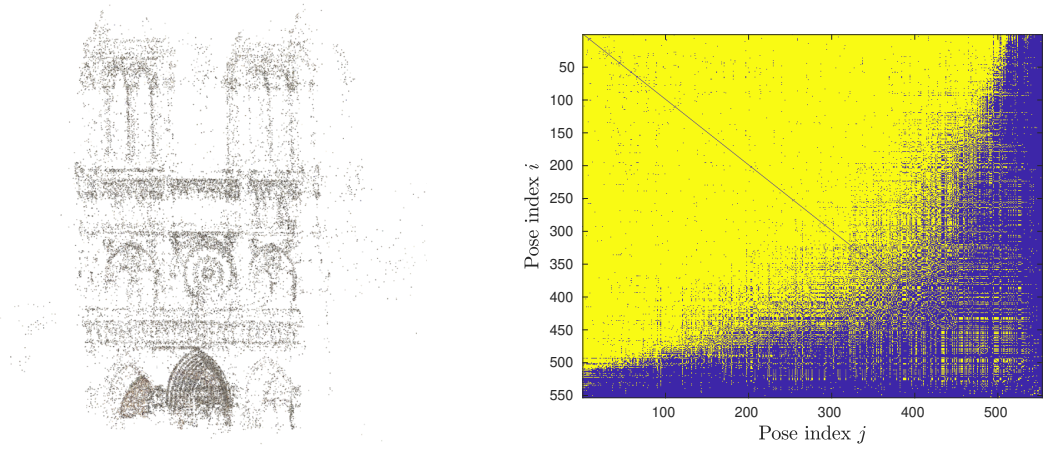
Let $S_{ij} := \hat{t}_j (\hat{t}_i - \widetilde{t}_{ij})^\top \widetilde{R}_{ji}$. We are going to define a symmetry metric $s \in [-1, 1]$ as:

$$\boxed{s(X) := \frac{||X_{\text{sym}}||_F - ||X_{\text{anti}}||_F}{||X_{\text{sym}}||_F + ||X_{\text{anti}}||_F}}$$

where $X_{\text{sym}} = \frac{1}{2}(X + X^\top)$ and $X_{\text{anti}} = \frac{1}{2}(X - X^\top)$. For X symmetrical we have $s(X) = 1$. Therefore, for a certain $\epsilon \geq 0$, we consider Hypothesis 4.1.1 to be valid if $\forall\ (i,j) \in E : |s(S_{ij}) - 1| < \epsilon$. The second condition is the application of the lower bound of the rotation averaging cost function, as previously presented. If the hypothesis is valid then the problem is separable in the rotations and translations. Provided the rotations are optimal, the translations are as well since they are the solution to the convex, translations-only problem.

4.5 Outlier detection

We have presented two closed-form solutions to the problems of rotation averaging and motion averaging in $\mathrm{SE}(3)$. Under the noise models described in Section 2.2, these solutions successfully approximate the ML estimate, as will be demonstrated empirically in Chapter 5. However, outliers were not taken into



(a) Point cloud (complete 3D scene).

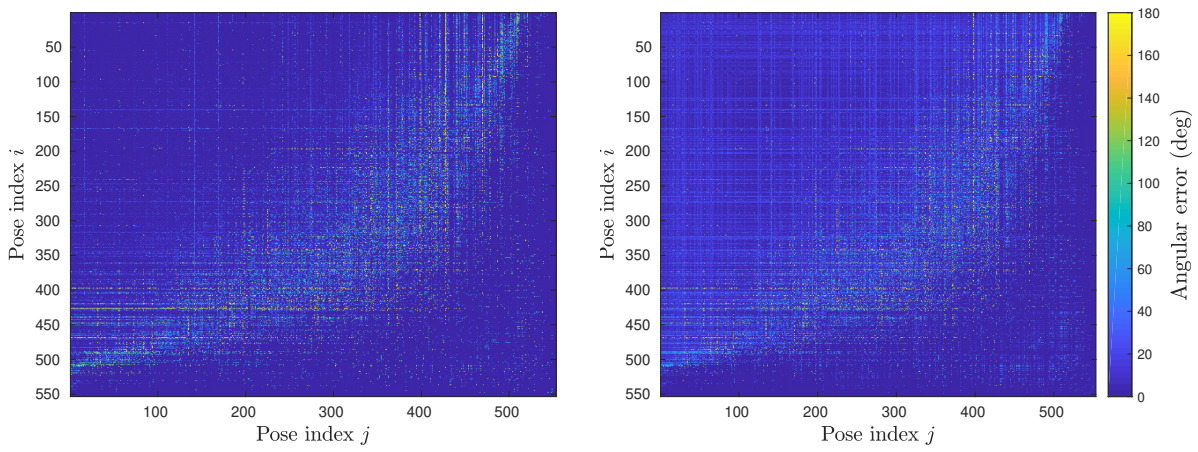
(b) Adjacency matrix (yellow: ones, blue: zeros).

Figure 4.3: Notre Dame dataset from [2].

account in our model. In particular, our derivation was motivated by the log-likelihood function put forward by Carlone et al. [10], which makes use of l^2-norms, known for their poor performance in the presence of outliers. In this Section, we present one technique to identify model-incoherent rotation measurements, inspired by the method of IRLS.

In the context of point cloud registration outliers are expected. This is due to the fact that the pairwise relative transformations are obtained via ICP or image matching (in the case of RGB-D cameras), which either produce very good results or very poor ones, usually with no middle ground. As a consequence, there will be a subset of relative transformations in the pose graph which are not consistent with the rest. One possible solution to increase robustness in the presence of outliers would be to reformulate the model using e.g. the l^1-norm. As we have seen in Section 2.3 this technique has already been featured in the literature. However, it would imply discarding the eigenvector solutions presented. To avoid this, we will instead consider approaches based on outlier detection and elimination, rather than those involving a complete model overhaul. One way of going about identifying incoherent measurements are Random Sample Consensus (RANSAC) algorithms. However, in MA the sheer amount of transformations and the CPU time necessary to compute the model from a random sample of relative pairwise measurements prevents RANSAC from being efficient. This has led us to implement a novel outlier detection method. Albeit built *ad hoc* to deal with the poor performance of our eigenvector solution in bundle adjustment rotation averaging datasets, this technique successfully increased the accuracy of our solution.

In order to motivate our outlier identification technique we are going to consider a famous rotation averaging problem usually found in the bundle adjustment literature, the Notre Dame dataset represented in Fig. 4.3. By analyzing its adjacency matrix we arrive at the conclusion that its graph has a very densely connected region and a sparsely connected one (yellow and blue regions, respectively, of the adjacency matrix). It turns out this is usually the case in bundle adjustment datasets. It is expected for these loosely connected regions of graph to be more difficult to optimize since the error cannot be redistributed as easily. In particular, for lone vertices with a single incident edge, there is no optimization possible, and their error will be dependent on this unique connection.



(a) Ground-truth fit error.

(b) EigenRA fit error.

Figure 4.4: Notre Dame dataset angular errors.

In order to witness this effect consider the heat map in Fig. 4.4b, which represents the angular error matrix e of the measurements w.r.t the ground-truth, i.e. $e_{ij} = ||\log \widetilde{R}_{ij} R_j R_i^\top||_F$ for $(i,j) \in E$ (converted to degrees). While the dense region has overall a small error, outliers can be identified in the frontier, as had been posited. Let us now analyze how consistent the same measurements are with our eigenvector solution. The latter shall be denoted as $\widehat{R}_i$. In Fig. 4.4a a heat map displays $||\log \widetilde{R}_{ij} \widehat{R}_j \widehat{R}_i^\top||_F$ for each edge $(i,j) \in E$. Similar considerations can be put forward in this case, i.e. our solution appears to be mostly inconsistent with the measurements in the frontier between the dense and sparse regions of the pose graph. In other words, for a small percentage of outliers, the eigenvector solution will fit the noisy measurements well enough to render the detection of outliers possible. An interesting remark is that in this second heat map, several lighter horizontal and vertical lines are visible in the upper-left corner. Empirical evidence suggests that these are due to the outliers in the frontier and sparse regions pushing the model away from the *good* solution of the dense subgraph. In fact, by considering only the rotations corresponding to the 300×300 block on the upper-left corner of the adjacency matrix, these lines were not visible.

Similar results were obtained for different rotation averaging datasets. In conclusion, we posit that assuming a small percentage of outliers, the eigenvector fit can be used to assess the quality of the measurements. The method devised to identify outliers can be described as follows. A $n \times n$ weight matrix W is initialized with ones. Iteratively: the block matrix $\widetilde{R}_{\mathcal{B}}$ with the measurements, weighted by W, i.e. $(W \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}}$ is used to compute the eigenvector solution $\widehat{R}_{\mathcal{C}}$, let λ_1 be the smallest eigenvalue; an error matrix e, similar to the ones displayed in the heat maps, is computed containing the errors of all the edges of the graph; rotation measurements with an error above a certain threshold η are replaced in $\widetilde{R}_{\mathcal{B}}$ by their estimate $\widehat{R}_i \widehat{R}_j^\top$ (instead of removing the respective edge from the graph which would reduce its connectivity); the weight matrix is updated as $W \leftarrow W \circ \rho(e)$ (where ρ is a loss function); the sum of the entries of the i-th line of W, except the diagonal entry, is set to $\deg(v_i)$ and the diagonal of W is replaced by ones; finally, W is replaced by its symmetric component. The steps are repeated until convergence of λ_1. We found, empirically, that the choice of the loss function ρ played an important role

in the rate of convergence and ground-truth errors. After running multiple simulations, we chose the loss function from Eq. (4.9), which can be fine-tuned via the parameters a and b.

$$\rho(e; a, b) = a + b \exp\left(\pi - \frac{e}{2}\right) \tag{4.9}$$

The reasoning behind this algorithm comes from the knowledge that in the case without outliers λ_1 should be close to 0. By iteratively updating the weights, the transformations which remain unexplained by the eigenvector (large error) over many iterations incur increasingly larger penalties (smaller weights). Consequently, outlier transformations will have an increasingly smaller impact on the spectrum and eigenspaces and thus λ_1 decreases in absolute value. Generally, from a certain weight matrix W onwards, λ_1 will cease to change and the algorithm stops.

4.6 Implementation details

This section concludes the chapter by describing the implementation details behind the algorithms proposed to solve rotation averaging under a Langevin noise assumption, which will be called EigenRA; rotation averaging under Langevin noise with outliers, named EigenRAO; and finally, motion averaging in $\mathrm{SE}(3)$ entitled EigenMA, assuming a high SNR and Langevin and Gaussian noise distributions for the rotations and translations, respectively.

The three algorithms described can be implemented in Python using SciPy or MATLAB, since the only requirement is a routine capable of performing efficient eigendecomposition. The most popular algorithm to compute a small subset of eigenvectors of a sparse matrix is arguably the Krylov-Schur method [16], implemented in MATLAB *eigs* function. We implemented this algorithm in C++, to solve the rotation averaging problem. For a given scalar shift operator σ near zero, and an initial vector v_0 our solver generates a Krylov subspace of dimension 20 by iteratively computing

$$v_{k+1} = \underbrace{\left((\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}} - \sigma I\right)^{-1}}_{\text{Shift and invert spectral transformation}} v_k \tag{4.10}$$

Since the matrix being inverted in Eq. (4.10) is symmetric, we use Cholesky $LDL^\top$ factorization to solve for v_{k+1}. The vectors $\{v_k\}_k$ are orthogonalized at each iteration, and their projections are used to build a Hessenberg matrix containing the Ritz values and vectors from which our solution is obtained. Should the algorithm not converge, additional Krylov vectors are computed.

To solve for translations, we proposed a second eigenvector problem using the auxiliary matrix $M'_{\mathcal{B}}$. However, since the eigenvalue is known to be zero this optimization can be expedited. The eigendecomposition proposed can be rewritten as

$$\left((\mathcal{L} \otimes \mathbb{1}_{4\times 4}) \circ M'_{\mathcal{B}}\right)\widehat{t} = 0 \tag{4.11}$$

Eq. (4.11) states that solving for translations boils down to finding the nullspace of a matrix. Nonetheless, this subspace has dimension 4. We can rearrange the equation by taking into account that every fourth

entry of the vector $\widehat{t}$ is one. Thus

$$(\mathcal{L} \otimes \mathbb{1}_{4\times 4}) \circ \widehat{R}_{\mathcal{C}} \widehat{R}_{\mathcal{C}}^{\top} \begin{bmatrix} \widehat{t}_1 \\ \vdots \\ \widehat{t}_n \end{bmatrix} + \underbrace{\begin{bmatrix} \sum_{j \in E(v_1)} \frac{1}{2}(\widetilde{t}_{1j} + \widehat{R}_1 \widehat{R}_j^{\top} \widetilde{R}_{j1} \widetilde{t}_{1j}) \\ \vdots \\ \sum_{j \in E(v_n)} \frac{1}{2}(\widetilde{t}_{nj} + \widehat{R}_n \widehat{R}_j^{\top} \widetilde{R}_{jn} \widetilde{t}_{nj}) \end{bmatrix}}_{b \in \mathbb{R}^{3n}} = 0$$

Similarly to the rotation averaging problem aforementioned, the symmetry of $(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widehat{R}_{\mathcal{C}} \widehat{R}_{\mathcal{C}}^{\top}$ allows us to use Cholesky factorization to solve the (sparse) linear system efficiently. Letting b denote the right-handmost vector, and $LDL^{\top}$ the Cholesky decomposition of $(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widehat{R}_{\mathcal{C}} \widehat{R}_{\mathcal{C}}^{\top}$, solving for translations in Eq. (4.12) is now trivial.

$$LDL^{\top} \begin{bmatrix} \widehat{t}_1 \\ \vdots \\ \widehat{t}_n \end{bmatrix} = b \tag{4.12}$$

Since the matrices factorized when solving for translations and rotations share the same structures of zero and non-zero entries, the Cholesky solver only has to analyze the matrix patterns once. Our implementation uses Intel® MKL PARDISO LDLT solver. For other CPUs our code can be compiled with Eigen SimplicialLDLT.

Both EigenRAO and EigenMA depend on the implementation of EigenRA. To aid comprehension, this algorithm is represented in Fig. 4.6. As it can be observed, the procedure is a direct application of the results presented earlier in this chapter. One point worth mentioning is the anchoring step. If there is a set of rotation estimates $\{\widehat{R}_i\}_{i=1,\dots,n}$ which are a solution to the problem, then the set $\{\widehat{R}_i S\}_{i=1,\dots,n}$, for any $S \in \mathrm{SO}(3)$ is also a solution. This is known as *gauge freedom* and can be dealt with by fixing one rotation. More details can be found in Appendix B.6.

The output of EigenRA is used as an input to EigenMA together with the measurements, the graph Laplacian and the preconditioned Cholesky solver, as represented in Fig. 4.6. The result is the set of Euclidean transformation estimates. To finish off, the pseudocode for EigenRAO is presented in Algorithm 3. This algorithm can be fine-tuned by changing the parameters a, and b of the loss function, and the error threshold η. Our implementation is parameterized by default with $a = -1.75$, $b = 0.85$ and $\eta = \pi/2$.

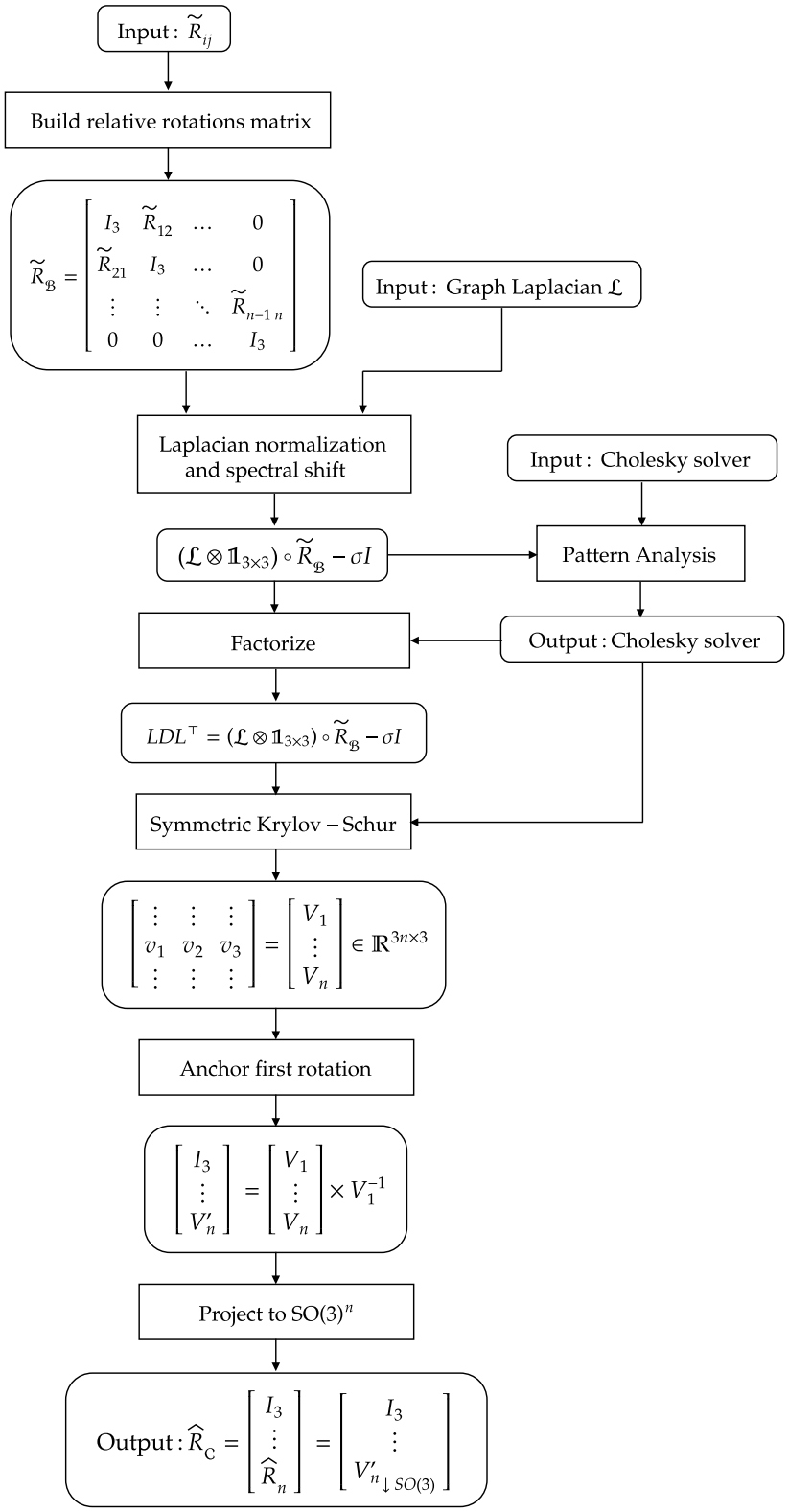


Figure 4.5: Diagram of the EigenRA algorithm

Algorithm 3 Outlier detection in rotation averaging

procedure EIGENRAO($\widetilde{R}_{\mathcal{B}}, \mathcal{L}, A, a, b, \eta$)
  $\overline{R}_{\mathcal{B}} \leftarrow \widetilde{R}_{\mathcal{B}}$
  $W \leftarrow \text{ones}(n, n)$
  do
    $E \leftarrow \text{zeros}(n, n)$
    $(\widehat{R}_{\mathcal{C}},\ \lambda_0) \leftarrow \text{eigenRA}((W \otimes \mathbb{1}_{3\times 3}) \circ \overline{R}_{\mathcal{B}}, \mathcal{L})$
    $E_{ij} \leftarrow ||\log \widetilde{R}_{\mathcal{B}_{ji}} \widehat{R}_{\mathcal{C}_i} \widehat{R}_{\mathcal{C}_j}^{\top}||_F \ \text{ if } \ (i,j) \in E$
    $E_{ji} = E_{ij}$
    $\overline{R}_{\mathcal{B}} \leftarrow \big((E > \eta) \otimes \mathbb{1}_{3\times 3}\big) \circ (\widehat{R}_{\mathcal{C}} \widehat{R}_{\mathcal{C}}^{\top}) + \big((E \leq \eta) \otimes \mathbb{1}_{3\times 3}\big) \circ \widetilde{R}_{\mathcal{B}}$
    $E \leftarrow \rho(E; a, b)$
    $W \leftarrow W \circ E$
    $W_{ij} \leftarrow \deg(v_i) \times E_{ij} \big(\sum_{i=1, i\neq j}^{n} W_{ij}\big)^{-1} \quad \text{for } \ i \neq j$
    $W_{ii} \leftarrow 1$
    $W \leftarrow \frac{1}{2}(W + W^{\top})$
  while λ_0 not converged
  return $E, \widehat{R}_{\mathcal{C}}$

Besides the three algorithms proposed, other programs and routines were created. The most relevant are listed in Table 4.1 and available at https://github.com/gabmoreira/pipe. A working version of our PGO solver, together with the complete RGB-D registration pipeline will be made available, so that users may upload either a pose graph (as a .g2o file) or an RGB-D image sequence, and download the reconstructed 3D scene and/or the camera trajectory.

Program	Language	Dependencies
Krylov-Schur	C++	Eigen, Intel® MKL
Damped-Newton	Python	NumPy
Riemannian Grad	C++ / MATLAB	Eigen
EigenRA	C++ / MATLAB	Eigen, Intel® MKL
EigenRAO	MATLAB	-
EigenMA	C++ / MATLAB	Eigen, Intel® MKL
RANSAC 3D	C++ / MATLAB	Eigen
RGB-D Registration Pipeline	C++ / MATLAB	SIFT
Global ICP	C++	Eigen, ANN
Visualization Module	C++	OpenGL, GLM, GLFW

Table 4.1: Programs created available online.

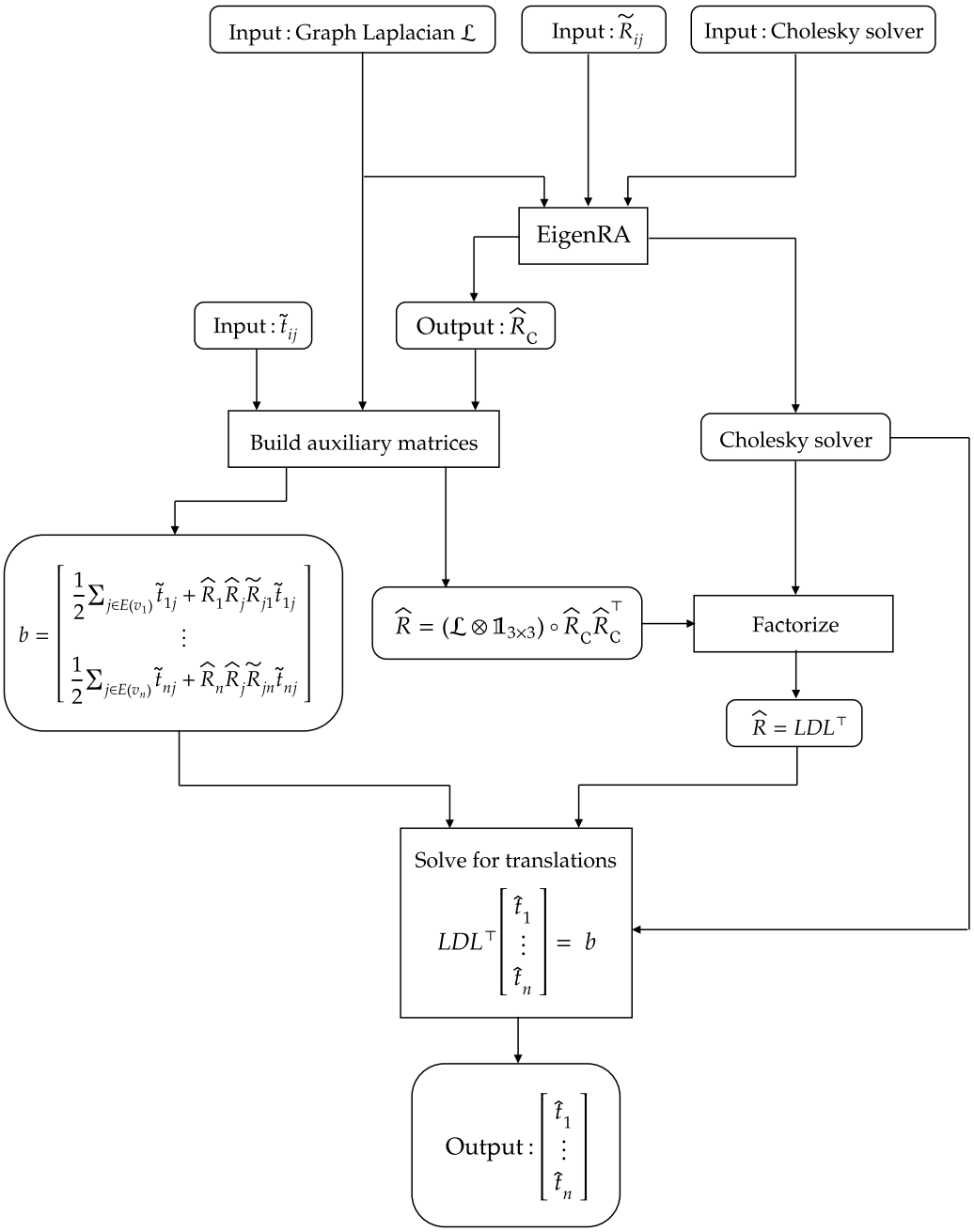


Figure 4.6: Diagram of the EigenMA algorithm.

Chapter 5

Evaluation and experiments

In this chapter we present the results of several motion averaging simulations and benchmarks in order to assess the performance of our algorithms and establish meaningful comparisons against the state-of-the-art. All the tests were conducted using an Intel Core i7-4700HQ CPU with a maximum clock frequency of 3.4GHz and 16GB of RAM.

5.1 Performance metrics

In order to evaluate the perfomance of our algorithms and to be able to benchmark them against state-of-the-art methods we first need to introduce the performance metrics used to make these assessments. We will denote by $\widehat{R}_i$ and $\widehat{t}_i$ our estimates of a true rotation R_i and a true translation t_i, respectively. Measurements are written with the usual tilde notation, as $\widetilde{t}_i$ and $\widetilde{R}_i$. Block matrices with measured rotations and transformations are denoted as, $\widetilde{R}_{\mathcal{B}}$ and $\widetilde{M}_{\mathcal{B}}$, respectively.

Motion averaging

In the case of motion averaging in $\mathrm{SE}(3)$, the log-likelihood is our objective function. It allows for an assessment of how consistent the estimates are with the measurements.

$$f_{\mathrm{ML}}(\widehat{t}_i, \widehat{R}_i) = -\frac{1}{2} \sum_{i,j \in E} ||\widetilde{t}_{ij} - \widehat{t}_i + \widehat{R}_i \widehat{R}_j^\top t_j||^2 + \sum_{i,j \in E} \mathrm{tr}\ \widetilde{R}_{ij} \widehat{R}_j \widehat{R}_i^\top$$

This objective is actually dependent on the quality of the measurements and may not reflect the the true accuracy of the estimations. Consequently, whenever ground truth data is available we will compute the ground truth error as

$$f_{\mathrm{GT}}(\widehat{M}_i) = \sum_{i=1}^{n} ||M_i - \widehat{M}_i||_F^2 \tag{5.1}$$

Rotation averaging

For performance comparisons when dealing with the problem of multiple rotation averaging, the following metrics are often used in the literature: root mean squared error (RMSE), mean angular error (MN), median angular error (MD) and the cost function from Eq. (4.6). The first three are based on the geodesic distance on $\mathrm{SO}(3)$ between the ground truth and the estimated rotations. The cost function evaluates the consistency between the estimates and the measurements. A summary of the different evaluation criteria is provided in Table 5.1.

Metric	Abbreviation	Formula
Median error (deg)	MD	$\operatorname{median}\left\{d_{\mathrm{geo}}(\widehat{R}_i, R_i)\right\}_{i=1,\dots,n}$
Mean error (deg)	MN	$\frac{1}{n}\sum_{i=1}^{n} d_{\mathrm{geo}}(\widehat{R}_i, R_i)$
Root mean square error (deg)	RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n} d_{\mathrm{geo}}(\widehat{R}_i, R_i)^2}$
Cost function	-	$\operatorname{tr} \widehat{R}_{\mathcal{C}}\big((\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}}\big)\widehat{R}_{\mathcal{C}}^{\top}$

Table 5.1: Performance metrics for the multiple rotation averaging problem

5.2 Generating synthetic data

In order to evaluate our algorithms in different scenarios, synthetic data was generated. This allowed for a careful study on the influence of parameters such as the noise standard deviation, the probability of outliers and the graph connectivity. Since every MA problem in $\mathrm{SE}(3)$ has a rotation averaging problem associated, we will describe the procedure used to generate random pose graphs in the Euclidean space. For a certain graph order n, and standard deviations σ_R and σ_t:

1. Generate n rotation matrices R_i and n vectors $t_i \in \mathbb{R}^3$. For every pair $\{R_i, t_i\}$, build the respective matrix $M_i \in \mathrm{SE}(3)$. Stack the matrices M_i vertically to create the block column $M_{\mathcal{C}} \in \mathrm{SE}(3)^n$, which will be our ground-truth.

2. Compute the ground-truth relative transformations block matrix $M_{\mathcal{B}} = M_{\mathcal{C}} M_{\mathcal{C}}^{-b}$.

3. Generate a random connected graph $\mathcal{G} = (V, E)$ with an adjacency matrix A. Add ones or zeros to A to change connectivity of the graph (and the *Fiedler* value).

4. For every $(i, j) \in E$ draw a random rotation E_{ij} from $\mathrm{Lang}(\mathrm{I}_3, 1/\sigma_{\mathrm{R}}^2)$, e.g. using rejection sampling, and a random translation e_{ij} from $N(0, \sigma_t^2)$. Create a new matrix $\widetilde{M}_{\mathcal{B}}$ with the blocks (i, j) and (j, i) defined as

$$\widetilde{M}_{\mathcal{B}_{ij}} = \begin{bmatrix} E_{ij}R_{ij} & t_{ij} + e_{ij} \\ 0_3^\top & 1 \end{bmatrix}$$

and $M_{\mathcal{B}_{ji}} \leftarrow \widetilde{M}_{\mathcal{B}_{ij}}^{-1}$.

5. Mask out transformation blocks which do not correspond to edges of the graph, i.e.

$$\widetilde{M}_{\mathcal{B}} \leftarrow (A \otimes \mathbb{1}_{4\times 4}) \circ \widetilde{M}_{\mathcal{B}} + I_{4n}$$

6. For rotation averaging problems, we can build $\widetilde{R}_{\mathcal{B}}$ by using the rotation blocks of $\widetilde{M}_{\mathcal{B}}$.

5.3 Rotation averaging

5.3.1 On the optimality of our solution

We begin our analysis of the proposed rotation averaging algorithm EigenRA by an empirical study on the optimality of our solution. Being based on eigenspaces and projections to $\mathrm{SO}(3)$, there is no *a priori* guarantee of optimality[1]. Furthermore, this solution may not even correspond exactly to a local cost function minimum on the manifold. In Chapter 4, a lower bound was put forward for this cost function. Attaining it is a sufficient condition to ensure optimality. Since this cannot always be verified, we will borrow the Riemannian gradient descent method from Section 2.2.3 to find the local minimum closest to our solution.

As an example consider a rotation graph $\mathcal{G} = (V, E)$ with $|V| = 553$ and $|E| = 103932$. Fig. 5.1 represents the minimum cost attained using EigenRA, the Riemannian gradient initialized from our solution and the lower bound, for different measurement standard deviations. As expected, for $\sigma_R = 0$ our solution is optimal since the eigenspace intersects $\mathrm{SO}(3)^n$. Remarkably, for $\sigma_R \leq 5.0$ deg our solution still corresponds to the global optimum. Note that this is already considered a high noise level for most Computer Vision applications. The fact that the lower bound flatlines for large values of σ_R is in accordance with Conjecture 3.2.1 which states that the eigenvalues of $(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}}$ converge for increasing σ_R. Even if this hinders optimality verification, the Riemannian gradient suggests that our solution is *at least* close to a local optimum, even for unrealistic noise levels.

In order analyse the ground truth error in terms of RMSE, MN and MD of these two optimization schemes, for the same graph, consider the plots in Figs. 5.2a, 5.2c and 5.2e. The first conclusion is that for small σ_R there is an approximate linear relationship between these metrics and the noise standard deviation. Moreover, and as expected, the two curves coincide for $\sigma \leq 5$ deg, and then the error from EigenRA seems to grow quicker than that of the Riemannian gradient (for all metrics). This can be explained as follows. A close inspection of the curves from Fig. 5.1 suggests that for larger standard deviations, our cost was already slightly larger than the local minimum obtained in the Riemannian

[1] Even if the work presented thus far suggests that under certain assumptions the solution is near the global optimum.

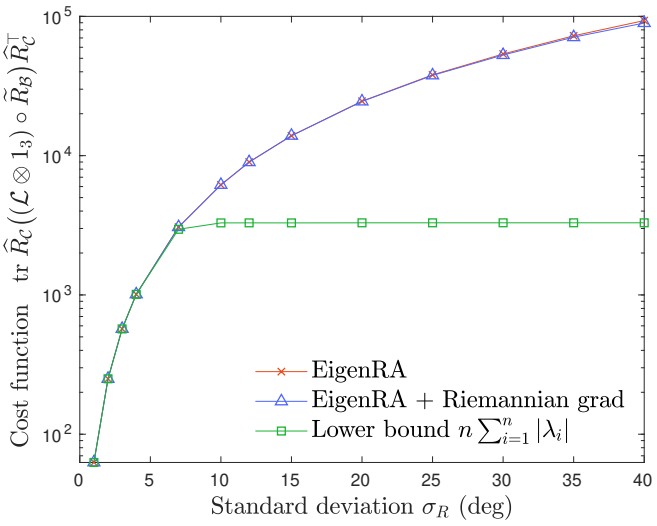


Figure 5.1: Minimum cost attained by our solution using EigenRA and Riemannian gradient descent (Riemannian grad) initialized from our solution, compared against the cost function's lower bound.

descent. This is reflected as a larger ground-truth error. Nonetheless, these considerations refer to noise levels which do not occur in real applications.

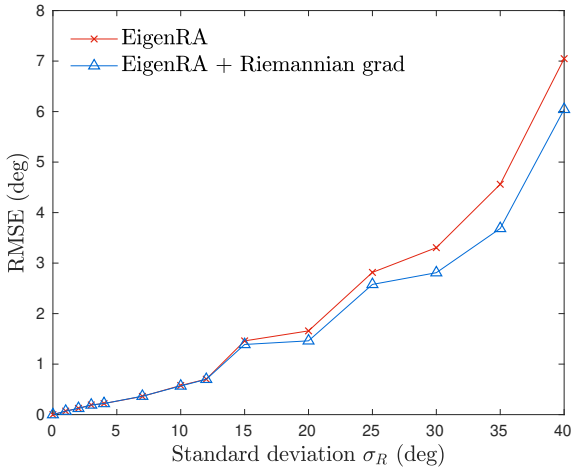
5.3.2 Impact of graph connectivity

It is reasonable to assume that well-connected graphs are easier to optimize. To understand the impact of the graph connectivity (Fielder value) on the quality of the solution produced by EigenRA we are going to consider four random rotation graphs $\mathcal{G}_i = (V, E_i)$, with $|V| = 550$ fixed and E_i variable. The results of the RMSE, MN and MD ground-truth errors (in degrees) for a varying rotation noise standard-deviation, are plotted in Figs. 5.2b, 5.2d and 5.2f, respectively.

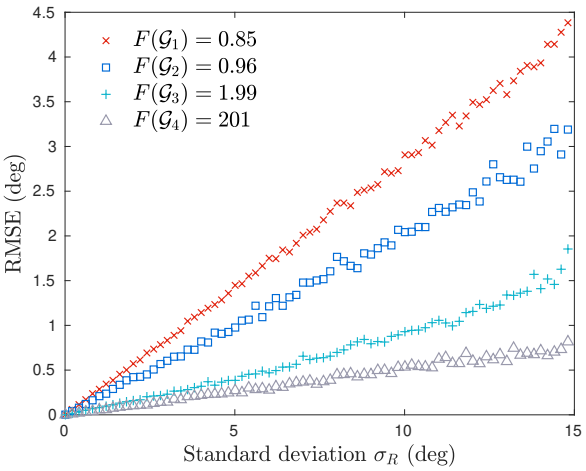
The linear relationship aforementioned between the ground-truth performance metrics and σ_R is evident in the three plots. More specifically, there is a proportionality relationship between each error metric and the standard deviation. As was expected, the proportionality constant increases as the connectivity, indicated by the Fiedler value $F(\mathcal{G}_i)$ decreases. In other words, optimization is more difficult in less connected graphs, since the redistribution of error inherent to multiple rotation averaging is not as easily accomplished.

5.3.3 Impact of outliers

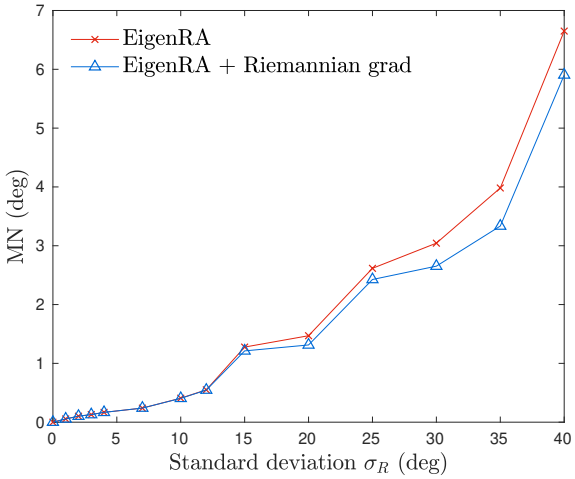
The closed-form solution in EigenRA was built on the assumption of Langevin distributed noise and absence of outliers. EigenRAO improved upon this formulation, and we claimed that this algorithm was capable of detecting model-incoherent rotation measurements. We present in Table 5.2, a comparison between EigenRA and EigenRAO, in terms of the RMSE for the same graph ($|V| = 553$, $|E| = 103932$), with different measurement standard deviations σ_R and probabilities of outliers p. The difference between the two solutions is remarkable. For noiseless measurements, EigenRAO retrieved the ground-truth solution for p up to 15%. Evidently, this cannot be generalized to other graphs, however empirical



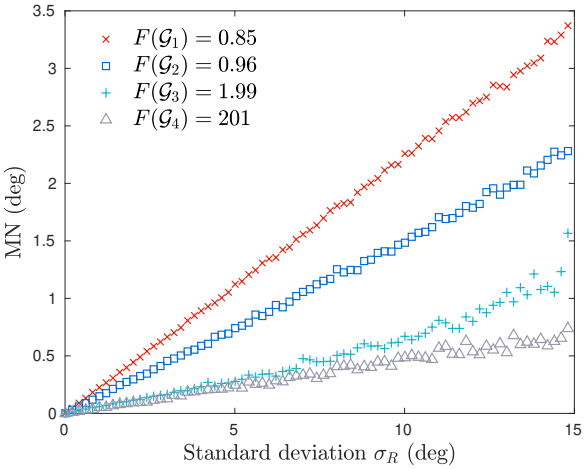
(a) RMSE



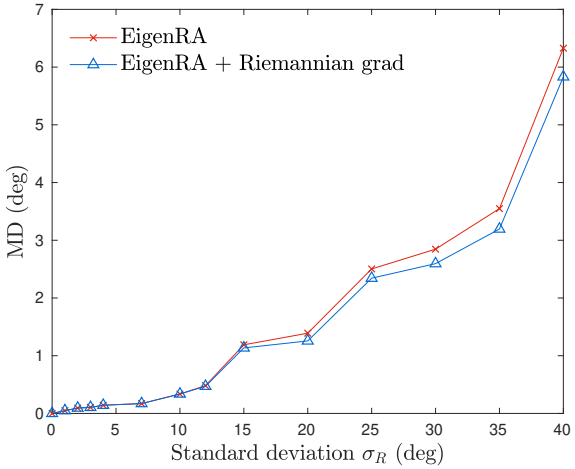
(b) EigenRA RMSE



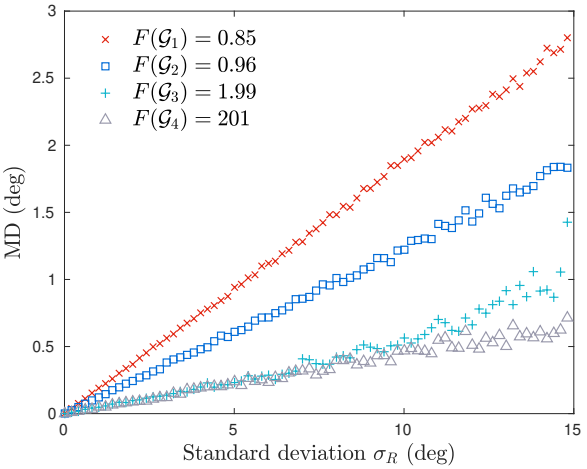
(c) Mean error



(d) EigenRA mean error



(e) Median error



(f) EigenRA median error

Figure 5.2: Left column: Ground-truth errors of EigenRA and EigenRA refined by Riemannian grad applied to synthetic pose graphs with 533 global rotations and different noise standard deviations σ_R. Right column: Ground-truth errors of EigenRA applied to 4 synthetic pose graphs with different connectivities $F(\mathcal{G}_i)$ and noise standard deviations σ_R.

evidence indicates that this algorithm can successfully detect most outliers and therefore improve the closed-form solution. As expected, for $p = 0$ the solutions yield the same result, since in this case all the measurements were simulated according to $\mathrm{Lang}(I_3, 1/\sigma_R^2)$. We can also observe that outliers have a greater impact on the quality of the solution for larger values of σ_R. Fig. 5.3 illustrates the convergence of the performance metrics per iteration of EigenRAO.

	Simulated rotation noise σ_R (deg)							
	0		0.5		1.5		2.5	
p	EigenRA	EigenRAO	EigenRA	EigenRAO	EigenRA	EigenRAO	EigenRA	EigenRAO
0.00	0.00	**0.00**	0 76	**0.08**	0.24	**0.24**	0.35	**0.33**
0.01	0.27	**0.00**	0.3	**0.08**	0.37	**0.25**	0.45	**0.30**
0.02	0.37	**0.00**	0.33	**0.08**	0.51	**0.24**	0.65	**0.35**
0.05	0.58	**0.00**	0.66	**0.08**	0.95	**0.55**	0.91	**0.26**
0.10	1.34	**0.00**	1.37	**0.08**	1.79	**0.45**	3.17	**0.33**
0.15	1.54	**0.01**	1.7	**0.09**	2.59	**0.89**	5.33	**2.16**

Table 5.2: RMSE (deg) comparison between EigenRA and EigenRAO for different probabilities of outliers p and noise levels σ_R. The same graph structure with $|V| = 553$ and $|E| = 103932$ was used at all times.

5.3.4 Benchmark datasets

To test our algorithm in real applications we will consider 12 real rotation averaging datasets compiled by Wilson and Snavely [28]. Six examples are represented in Fig. 5.5. These datasets contain not only noisy observations of the relative rotations but also the ground-truth information obtained through bundle adjustment using Bundler [2]. This allowed us to assess the accuracy of our solution in terms of MD, MN and RMSE. Since these metrics are common in the literature, they provide a means to establishing direct performance comparisons against other algorithms. Furthermore, this analysis also serves the purpose of empirically validating Conjecture 3.2.1, related to the the smallest eigenvalue (in absolute value) of $(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}}$. We start thus by analyzing the smallest eigenvalues of this matrix.

Spectral analysis

In Table 5.3 we present the smallest eigenvalues of the matrix of interest, graph characteristics such as its Fiedler value F, number of vertices $|V|$ and number of edges $|E|$, and finally the CPU time required by the eigendecomposition function in MATLAB (*eigs*).

The first result of interest is that Conjecture 3.2.1 holds in every case, i.e. the smallest eigenvalues are always smaller than or equal to F. However, note that most datasets verify $\lambda_i \approx F$ for $i = 1, 2, 3$. Recall from Fig. 3.1 that in synthetic datasets, this would occur for large σ_R. While the exact noise model for each of these datasets is unknown, a considerable fraction of them have large outlier percentages (this can be verified using ground-truth data). Consequently, we can expect EigenRA to perform poorly. Since most state-of-the-art approaches for solving multiple rotation averaging do consider outliers, it

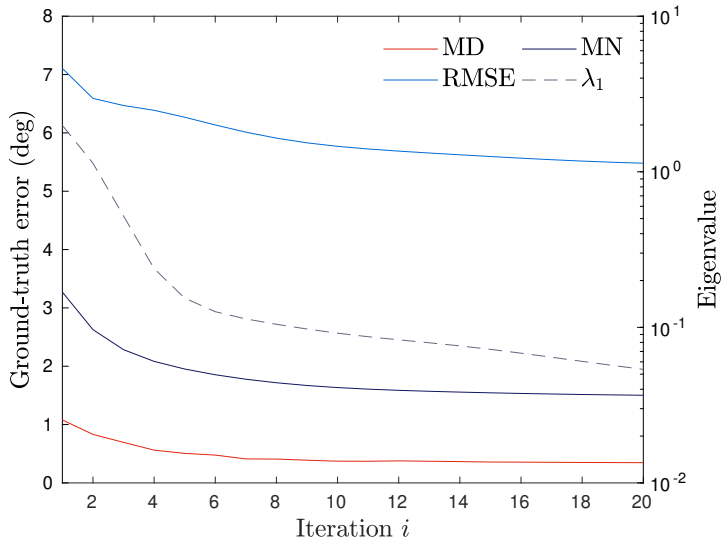


Figure 5.3: Convergence of the ground-truth errors as measured by MN, MD and RMSE, and of the smallest eigenvalue of the block matrix used in EigenRAO for the dataset *Ellis Island*.

would be unfair to present a direct comparison between them and our solution. For these datasets we will use instead EigenRAO.

Dataset	Smallest eigenvalues of $(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}}$			*Fiedler* value of $\mathcal{G}$	$\lvert V\rvert$	$\lvert E\rvert$	t_{CPU} (s)
Notre Dame (ND)	1.99	1.99	1.99	1.99	553	103932	0.18
Montreal ND (MND)	1.51	1.51	1.51	1.51	474	52424	0.12
Alamo (AL)	0.97	0.97	0.97	0.97	627	97206	0.21
Ellis Island (EI)	1.99	1.99	1.99	1.99	247	20297	0.03
Vienna Cathedral (VC)	0.80	0.80	0.80	0.80	918	103550	0.44
Roman Forum (RF)	0.95	0.95	0.95	0.95	1134	70187	1.14
Piazza del Popolo (PP)	0.83	0.91	0.91	0.91	354	24710	0.07
NYC Library (NY)	0.99	0.99	0.99	0.99	376	20680	0.07
Yorkminster (YM)	0.97	0.97	0.97	0.97	458	27729	0.15
Tower of London (TL)	0.45	0.97	0.97	0.97	508	23863	0.28
Madrid Metropolis (MM)	0.99	0.99	0.99	0.99	394	23784	0.09
Union Square (US)	0.30	0.44	0.44	0.44	930	25561	0.35

Table 5.3: Spectral analysis of popular datasets used in rotation averaging and bundle adjustment.

Ground truth error

The ground truth error was computed for the three metrics defined in the beginning of the chapter. EigenRAO was benchmarked against a novel deep learning approach proposed by Purkait et al. [29], named NeuRoRa, which combines two neural networks that suppress outliers and estimate the rotations, in this order. A second benchmark is the algorithm devised by Chatterjee and Govindu [30], based on a Quasi-Newton optimization scheme using a $l^{\frac{1}{2}}$-norm kernel function. For these algorithms, the results we present are those claimed by the authors in the respective papers. Finally, we also bench-

marked IRLS EIG-SE(3) by Arrigoni et al. [15], presented in Section 2.3, for which the author's MATLAB implementation can be found online. The results are presented in Table 5.4.

	EigenRAO			Chatterjee et al. [30]			Purkait et al. [29]			Arrigoni et al. [15]		
Dataset	MN	MD	RMSE	MN	MD	RMSE	MN	MD	RMSE	MN	MD	RMSE
AL	**2.26**	**0.63**	**6.97**	4.16	1.06	12.68	4.94	1.16	16.09	3.85	1.30	12.09
EI	**1.43**	**0.27**	**5.37**	2.87	0.51	10.36	2.59	0.64	12.82	3.07	0.81	10.50
YM	**2.04**	**0.98**	**5.11**	3.51	1.60	8.41	2.52	0.99	6.55	3.79	1.82	9.37
MND	**1.10**	**0.28**	6.29	1.54	0.51	7.45	1.23	0.64	**2.67**	1.86	0.62	11.22
VC	**3.4**	**0.85**	10.23	8.29	1.28	27.84	3.91	1.54	**9.93**	8.59	1.59	28.59
PP	3.14	**0.50**	**6.37**	4.06	0.89	8.41	**3.05**	0.79	9.01	3.89	0.97	9.45
US	**4.30**	3.31	**7.68**	9.33	3.93	22.44	5.98	**2.01**	17.61	6.92	5.41	13.07
NY	**1.89**	**0.79**	3.99	3.04	1.35	6.99	1.90	1.18	**2.89**	3.67	2.05	7.78
ND	2.05	**0.55**	8.06	3.53	0.65	14.61	**1.65**	0.68	**6.37**	3.94	1.20	14.85
RF	3.05	2.62	**5.39**	3.15	1.59	10.21	**2.39**	**1.31**	5.52	26.05	4.56	44.04
TL	2.74	1.73	**5.68**	3.94	2.43	9.06	**2.63**	**1.46**	5.78	4.47	2.58	10.56
MM	4.69	**1.09**	11.49	6.97	1.29	17.28	**2.55**	1.13	**6.59**	9.80	4.35	18.69

Table 5.4: Ground truth error comparison between EigenRAO and other rotation averaging approaches using bundle adjustment datasets.

Our solution outperforms that of Chatterjee and Govindu [30] for nearly all metrics and datasets. The same cannot be said about NeuRoRa which produces results comparable or better than ours in certain datasets. However, EigenRAO fares better overall, especially in terms of the MD error. By comparing EigenRAO's errors with those of the IRLS method by Arrigoni et al. [15], we conclude that the latter is surpassed in every dataset and metric. While the eigendecomposition step is the same in both algorithms, our iterative reweighting scheme guarantees the symmetry of the block matrix, uses a different loss function and replaces low-weight measurements by their estimates.

We did not conduct any analysis on CPU time since EigenRAO and IRLS EIG-SE(3) are implemented in MATLAB and cannot compete with a neural network without additional code optimization.

5.4 Motion averaging

5.4.1 Impact of noise and graph connectivity

In order to analyze the performance of MA algorithms in pose graphs with different connectivities and measurement noise levels, we simulated 15 synthetic pose graphs corresponding to the path travelled by a robot over a spherical surface of radius 100. These graphs shared a common number of poses $|V| = 1000$, but 6 different levels of connectivity, from 80% to 99% of missing pairwise transformations (or edges). In terms of noise models, 3 different configurations of tuples (σ_R, σ_t) were tested. Since our proposed MA algorithm, EigenMA, is similar to EIG-SE(3) we start off by presenting a comparison between the two, using the pose graphs described. The ground-truth errors are displayed in Table 5.5.

	$\sigma_R = 1.0$ deg $\sigma_t = 0.01$		$\sigma_R = 2.0$ deg $\sigma_t = 0.01$		$\sigma_R = 5.0$ deg $\sigma_t = 0.01$	
% missing	EigenMA	EIG-SE(3)	EigenMA	EIG-SE(3)	EigenMA	EIG-SE(3)
80	**7.88e+01**	1.02e+02	**3.51e+02**	3.80e+02	**2.53e+03**	2.86e+03
85	**1.53e+02**	1.92e+02	**4.15e+02**	4.89e+02	**2.67e+03**	3.31e+03
90	**1.74e+02**	2.43e+02	**4.18e+02**	6.13e+02	**3.28e+03**	3.41e+03
95	**6.63e+02**	8.65e+02	**1.98e+03**	2.99e+03	**1.01e+04**	1.21e+04
99	**2.18e+03**	1.19e+04	**9.25e+03**	1.31e+05	**9.12e+04**	2.56e+06

Table 5.5: Comparison between EigenMA and EIG-SE(3) [15] in terms of ground-truth error as measured by f_{GT} for a synthetic spherical pose graph comprising 1000 poses and different graph connectivities. The errors presented were averaged over 10 random simulations.

Our algorithm achieved an average ground-truth error smaller than our competitor EIG-SE(3) in every scenario. While in some cases, especially for low percentages of missing data, the difference between the two might be small, the last row of the table is particularly remarkable. At 99% of missing pairwise transformations, EIG-SE(3)'s solution degenerates entirely. While this may seem an extremely high percentage of missing data, it is not uncommon to find such values in real robotics applications. In fact, in order to have 1% of relative transformations in this graph one still needs to run a pairwise registration algorithm 4995 times. In Appendix C.4, six examples of pose graph trajectories illustrate how EigenMA outperforms EIG-SE(3) for noisy observations and poorly connected graphs.

5.4.2 Benchmark datasets

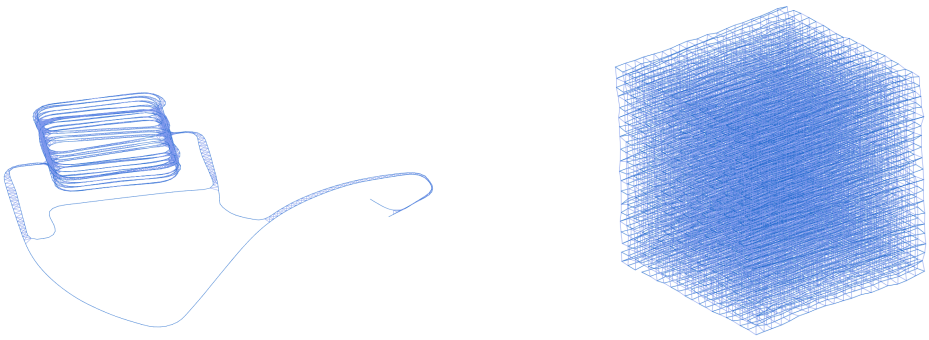
A direct comparison with state-of-the-art methods was conducted using six well-known pose graph datasets compiled by Carlone et al. [10]. Three of them are synthetic (Sphere, Torus3D, Grid3D) and the other three are real (Garage, Cubicle, Rim). Since ground-truth data is not available for either dataset, we will rely on the likelihood function to compare the different approaches.

Spectral analysis

A spectral analysis of the rotation component of the MA datasets considered allows us to assess the quality of the rotation measurements. The smallest eigenvalues of the normalized rotation block matrix, and the graph properties of each dataset are provided in Table 5.6.

Dataset	Smallest Eigenvalues of $(\mathcal{L} \otimes I_3) \circ \widetilde{R}_{\mathcal{B}}$			*Fiedler* value of $\mathcal{G}$	$\lvert V \rvert$	$\lvert E \rvert$
Garage	4.2e-07	5.4e-07	6.0e-07	3.7e-04	1661	6275
Grid	8.7e-03	8.7e-03	8.8e-03	2.3e-02	8000	22236
Cubicle	9.0e-06	1.9e-04	1.9e-04	3.2e-03	5750	16869
Sphere	2.2e-01	2.2e-01	2.2e-01	2.2e-01	2200	8647
Torus	3.9e-03	4.0e-03	4.1e-03	3.9e-03	5000	9048
Rim	1.7e-05	1.1e-03	1.1e-03	1.3e-03	10195	29743

Table 5.6: Spectral analysis of popular 3D datasets used in SLAM.



(a) Parking garage (n=1661) (b) Grid3D (n=8000)

Figure 5.4: Camera trajectory and loop closures of two datasets optimized by EigenMA.

The results are, once again, in accordance with Conjecture 3.2.1. However, unlike the rotation averaging datasets previously considered, the smallest eigenvalues of the normalized block matrix here are, in general, smaller than the Fiedler value. This is true for the Garage, Grid3D and Cubicle datasets.

Maximum likelihood

For each MA dataset we computed the global pose estimates in closed-form using EigenMA and EIG-SE(3). For the latter we used the author's original MATLAB implementation. Additionally, and due to the prominence of the chordal relaxation method in the literature as an initialization for iterative solvers, we benchmarked this technique as well, using a C++ implementation. Since neither of these methods are optimal (locally or globally), we resort to Gauss-Newton (g2o) initialized from our solution to obtain a local maximum of f_{ML}. A maximum of 10 iterations was set for all datasets, despite some of them converging in less than that. The log-likelihood maximum attained and the CPU time required by each method are presented in Table 5.7. Is this Table we also list the average symmetry $\overline{s}$, used to validate Hypothesis 4.1.1 (high SNR). As an example, in Fig. 5.4 we show the camera trajectory resulting from our optimization of Garage and Grid3D as well as the loop closures.

	EigenMA			Chordal [20]		EIG-SE(3) [15]		Gauss-Newton [11]	
Dataset	f_{ML}	$\overline{s}$	t_{CPU} (s)	f_{ML}	t_{CPU} (s)	f_{ML}	t_{CPU} (s)	f_{ML}	t_{CPU} (s)
Garage	18824.3	1.0	**0.03**	18824.3	0.28	14136.5	1.78	18824.4	0.22
Grid3D	66376.1	0.94	**0.83**	66376.8	13.46	65770.4	2.11	66377.0	492.17
Cubicle	50599.0	0.99	**0.23**	50596.4	1.53	37420.3	2.12	50599.5	3.65
Sphere	16644.6	0.74	**0.16**	13289.0	1.04	-516589.2	0.43	24667.3	3.04
Torus3D	27051.5	0.95	**0.20**	27051.3	1.19	26985.6	4.04	27051.9	7.01
Rim	89136.0	0.98	**0.46**	88873.9	2.71	66407.3	7.81	89152.2	10.69

Table 5.7: Comparison between EigenMA, Chordal relaxation, EIG-SE(3) and g2o (10 iterations).

Since Garage, Grid3D and Cubicle are the datasets with smallest eigenvalues (in absolute value) we posit that these pose graphs should be easy to optimize. The experiments confirm this, since 10 Gauss-Newton iterations did not increase the value of f_{ML} by a significant amount, i.e. our solution is quasi-optimal (locally). The estimated pose graphs produced by EigenMA can be visualized in Appendix C.1. Those same pose graph trajectories are superimposed on g2o's solutions in Appendix C.2. EIG-SE(3)'s solutions can be found in Appendix C.3.

The only datasets for which there is a considerable difference between the local maximum and our solution are Sphere and Rim. This difference is common to all three of the suboptimal methods, ours attaining the highest objective among them. The Sphere case is particularly remarkable. We hypothesize that the lacklustre performance of EigenMA is due to the configuration of the pose graph itself. In fact, since it simulates the successive poses of a robot travelling on a spherical surface, the relative rotations between equidistant positions are nearly constant and this may hinder optimization techniques relying on eigenspaces.

When comparing EigenMA to EIG-SE(3) one notices that, except for the Garage dataset, the latter produces poorer results. This is evident both in the maximum likelihood attained and in the depiction of the estimated trajectories, provided in the Appendix. This performance difference stems from the high percentage of missing data in all the six datasets considered. As we have previously shown, EIG-SE(3)'s solution degenerated in synthetic datasets when the percentage of missing transformations was 99%, a figure that is common among SLAM datasets.

The chordal initialization is the only suboptimal technique that produces results comparable to ours. However, we attain higher objectives in Cubicle, Sphere, Torus3D and Rim and a nearly equal objective in Grid3D. Furthermore, the computation of the eigenvalues in EigenRA, which is done *by default* within the Krylov-Schur algorithm, allows for an assessment of the consistency of the measurements. A similar procedure in the Chordal method would translate to higher CPU times.

The value of the average symmetry $\overline{s}$ allows us to validate or disprove the high SNR Hypothesis assumed by our model. We have, for each dataset, with the exception of Sphere, $\overline{s} \approx 1$[2]. This leads us to confirm that, in these cases, $S_{ij} = \hat{t}_j(\widetilde{t}_{ji} + \widetilde{R}_{ij}\hat{t}_i)^\top$ is approximately symmetrical for $(i,j) \in E$ and thus our separate optimization of rotations and translations is valid[3]. The low value obtained for the Sphere dataset explains the spread between the objective we attained and the nearest local maximum (g2o).

In terms of CPU time, our compiled C++ version of EigenMA outperforms every method currently in existence. EigenMA fares better than g2o, even if the CPU time for the latter is dependent upon the number of iterations. As an example, the optimization carried out on Grid3D using our algorithm yields approximately the same result as g2o, but it is nearly 600 times faster. Furthermore, the CPU time we have indicated for this algorithm does not take into account the initialization which dictates how well it can perform. EIG-SE(3) is also slower than EigenMA, which can be explained by the fact that it performs eigendecomposition of non-symmetric matrices. Finally, the Chordal relaxation (implemented in C++) also lags behind EigenMA by a considerable amount.

[2]Evidently, the average symmetry being close to one does not translate to $\forall_{(i,j)\in E} s_{ij} \approx 1$. However, this metric provides a good insight into the validity of the high SNR hypothesis.

[3]Depending on the degree of precision desired.

(a) Montreal Notre Dame

(b) Ellis Island

(c) Alamo

(d) NYC Library

(e) Piazza del Popolo

(f) Roman Forum

Figure 5.5: Examples of point clouds obtained through bundle adjustment corresponding to the rotation averaging datasets [28] we used to benchmark EigenRA.

Chapter 6

Conclusions

6.1 Summary

This thesis was written in the hope of developing an efficient and precise methodology for solving the problem of pose graph optimization in the context of multiple point cloud registration. We presented an extensive review of the state-of-the-art (Chapter 2) and shed light on the divide which exists therein.

On the one hand, MLE (Section 2.2) is capable of modeling the problem with a high degree of accuracy but the optimization strategies involved are often too cumbersome and require good initializations in order to attain the global optimum. On the other hand, even if LR-MC (Section 2.3) has proved to be an elegant alternative, current optimization techniques often fail to achieve the same degree of accuracy as MLE, due to the relaxations employed.

We set out to bridge the gap between these two seemingly independent formulations and developed a new framework that combines the observation model from MLE and the algebraic properties intrinsic to LR-MC. Our solution to the otherwise difficult task of PGO, when applicable, outperforms the state-of-the-art in terms of efficiency without compromising precision.

6.2 Achievements

In terms of theoretical achievements, we showed that the column space of the complete relative transformations block matrix, which has known rank (Arrigoni et al. [13]), is an invariant subspace of this matrix (Proposition 3.2.1). By establishing the link with the underlying graph, we generalized this proposition to relative transformation matrices with missing blocks by showing that, when normalized by the graph Laplacian, the global transformations we seek belong to an invariant subspace of this matrix (Proposition 3.2.4). Unbeknownst to us, some of these results had already been featured in the literature [15]. Nevertheless, the theory we set forth in Chapter 3 goes beyond what has been published thus far.

Finally, we showed that the invariant subspace Propositions we presented could be used to retrieve, in closed-form, the approximate ML transformation estimates, under a High-SNR hypothesis and a Langevin-Gaussian generative noise model.

The empirical results we presented in Chapter 5 validate the claims we made throughout this thesis and allow us to assert that:

- Under Langevin noise and assuming a standard deviation no greater than 5 degrees, the optimal solution of multiple rotation averaging can be computed in closed-form using our algorithm, EigenRA. Moreover, for larger noise standard deviations, our solution may still be quasi-optimal.
- Our iterative method to solve multiple rotation averaging with outliers, EigenRAO, proved to be more accurate than the state-of-the-art, when tested in bundle adjustment datasets (Table 5.4). However, our current implementation still lags behind other approaches in terms of CPU time.
- Our C++ implementation of EigenMA, based on the Krylov-Schur method for spectral decomposition and Cholesky factorization, outperforms EIG-SE(3) in terms of precision (Table 5.5). Moreover, it is considerably faster than the Chordal relaxation solution, which is often used to initialize iterative methods. This increased efficiency does not compromise the accuracy of the solution.

All in all, the three algorithms we put forward have a competitive edge over the state-of-the-art. Additionally, we designed an entire registration pipeline, which at the moment caters to RGB-D visual data. As an example, our experiment using the Burghers RGB-D dataset [17] attests the capabilities of our registration software when it comes to 3D reconstruction and motion estimation (Appendix D).

6.3 Future work

The research we have conducted opens up different possiblities of future work, both in terms of theoretical developments and more efficient implementations. Among others, we propose to:

- Attempt to create a hybrid version of EigenRAO combining our closed-form solution for Langevin distributed noise with a neural network to filter outlier observations. This would allow us to benefit from the accuracy of EigenRA without the computational cost overhead of EigenRAO.
- Analyze how the invariant subspace propositions we put forward can impact the selection of pairwise relative transformations to build the pose graph. As we showed in this thesis, the spectrum of the normalized relative transformations block matrix reflects the quality of the observations. This information could be used to select or discard new transformations, e.g. in real-time applications.
- Implement an efficient online version of our pipeline. Possible applications include e.g. real-time airborne 3D mapping by groups of drones with embedded RGB-D cameras and autonomous flight.
- Attempt to introduce our formulation in bundle adjustment. Despite being a problem inherently different from PGO since it deals with the minimization of the total reprojection error, there may be room for improvement using the invariant subspace considerations we presented.

Additionally, we intend to contribute towards open-source projects such as AI4EU and ROS by implementing our RGB-D registration pipeline in C++ with Python and MATLAB bindings. This is currently under development. See GitHub page for updates.

Bibliography

[1] P. H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[2] N. Snavely, S. M. Seitz, and R. Szeliski. Photo Tourism: Exploring image collections in 3D. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, 2006.

[3] 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.

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

[5] J. M. Phillips, R. Liu, and C. Tomasi. Outlier robust ICP for minimizing fractional RMSD. *3DIM 2007 - Proceedings 6th International Conference on 3-D Digital Imaging and Modeling*, (3dim):427–434, 2007.

[6] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, pages 145–152, 2001.

[7] P. H. Schönemann and R. M. Carroll. Fitting one matrix to another under choice of a central dilation and a rigid motion. *Psychometrika*, 35(2):245–255, 1970.

[8] R. Toldo, A. Beinat, and F. Crosilla. Global registration of multiple point clouds embedding the Generalized Procrustes Analysis into an ICP framework. *3Dpvt 2010*, (January):8, 2010.

[9] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert. Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 125–132. IEEE.

[10] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert. Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4597–4604, 2015.

[11] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.

[12] F. Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.

[13] F. Arrigoni, B. Rossi, and A. Fusiello. Global registration of 3D point sets via LRS decomposition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9908 LNCS:489–504, 2016.

[14] C. Jin, J. Zhu, Y. Li, S. Pang, L. Chen, and J. Wang. Multi-view Registration Based on Weighted Low Rank and Sparse Matrix Decomposition of Motions. pages 1–9, 2017. URL http://arxiv.org/abs/1709.08393.

[15] F. Arrigoni, B. Rossi, and A. Fusiello. Spectral Synchronization of Multiple Views in SE(3). *SIAM Journal on Imaging Sciences*, 9:1963–1990, 01 2016.

[16] G. W. Stewart. A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2002.

[17] Q.-Y. Zhou and V. Koltun. Dense scene reconstruction with points of interest. *ACM Trans. Graph.*, 32(4), 2013.

[18] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[19] N. Boumal, A. Singer, P. A. Absil, and V. D. Blondel. Cramér-Rao bounds for synchronization of rotations. *Information and Inference*, 3(1):1–39, 2014.

[20] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[21] R. Tron and R. Vidal. Distributed image-based 3-D localization of camera sensor networks. *Proceedings of the IEEE Conference on Decision and Control*, pages 901–908, 2009.

[22] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, 2013.

[23] R. F. Guerreiro and P. M. Aguiar. Factorization with missing data for 3D structure recovery. *Proceedings of 2002 IEEE Workshop on Multimedia Signal Processing, MMSP 2002*, pages 105–108, 2002.

[24] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, II(3):316–322, 2005.

[25] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi. Practical low-rank matrix approximation under robust L 1-norm. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1410–1417, 2012.

[26] G. Stewart and J. guang Sun. *Matrix Perturbation Theory*. Computer science and scientific computing. Academic Press, 1990.

[27] H. Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen 71*, (4):441–479, 1912.

[28] K. Wilson and N. Snavely. Robust Global Translations with 1DSfM. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[29] P. Purkait, T.-J. Chin, and I. Reid. NeuRoRA: Neural robust rotation averaging. *ArXiv*, abs/1912.04485, 2019.

[30] A. Chatterjee and V. M. Govindu. Robust Relative Rotation Averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):958–972, 2018.

[31] M. do Carmo. *Riemannian Geometry*. Birkhauser Basel, 1 edition, 1992.

[32] Y. Ma, J. Košecká, and S. Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 44(3):219–249, 2001.

[33] I. Dhillon and S. Sra. Modeling data using directional distributions. *Department of Computer Science. University of Texas*, pages 1–21, 2003.

[34] R. Diestel. *Graph Theory (Graduate Texts in Mathematics)*. 2000.

[35] J. Gross and J. Yellen. *Handbook of Graph Theory*. CRC Press, 2004.

[36] K. Petersen, M. Pedersen, et al. The matrix cookbook, vol. 7. *Technical University of Denmark*, 15, 2008.

Appendix A

Theory

A.1 The special orthogonal group

The *special orthogonal group* $\mathrm{SO}(3)$ is an algebraic subgroup of the group of invertible matrices $\mathrm{GL}_3(\mathbb{R})$ and is defined as

$$\mathrm{SO}(3) = \{R \in \mathbb{R}^{3\times 3} : R^\top R = I_3 \text{ and } \det R = 1\} \tag{A.1}$$

$\mathrm{SO}(3)$ is particularly useful in PGO since it can be used to represent rotations in $\mathbb{R}^3$. Other rotation representations, e.g. quaternions and vector-magnitude, are also featured in the literature. In order to fully grasp the properties of this group which render PGO more tractable we will define its geometry and present a widely used probability distribution. For in-depth results, the reader is referred to the works by Boumal et al. [19] and Hartley et al. [22].

Geometry

Besides being an algebraic group, $\mathrm{SO}(3)$ is a connected Lie group and thus a Riemannian manifold with a tangent space at each point [19]. We will denote the tangent space at a rotation R by $\mathrm{T_R SO}(3)$. The tangent space at the identity corresponds to the *Lie-algebra* of $\mathrm{SO}(3)$ and constitutes the set of skew-symmetric matrices:

$$\mathfrak{so}(3) \triangleq \mathrm{T_I SO}(3) = \{\Omega \in \mathbb{R}^{3\times 3} : \Omega = -\Omega^\top\}$$

For an arbitrary rotation R, the tangent space can be written as

$$\mathrm{T_R SO}(3) = R\,\mathfrak{so}(3) = \{R\Omega \in \mathbb{R}^{3\times 3} : \Omega = -\Omega^\top\} \tag{A.2}$$

To the tangent space at each point of $\mathrm{SO}(3)$ we can associate a Riemannian metric [31]. Let $\langle\ ,\ \rangle_R$ define an inner product in $\mathrm{T_R SO}(3)$. Following the steps in [19], we can choose the usual Frobenius inner product,

$$\langle R\Omega_1, R\Omega_2 \rangle_R = \text{tr } \Omega_1^\top R^\top R \Omega_2 = \text{tr } \Omega_1^\top \Omega_2$$

Consequently, the norm on the tangent space of R, denoted by $||\ .\ ||_R$ is simply

$$||R\Omega||_R = \langle R\Omega, R\Omega \rangle_R = ||\Omega||_F^2$$

The introduction of tangent spaces is not complete without presenting the exponential map. This local-defined mapping allows us to go from the tangent space back to the manifold. Let v be tangent to $\text{SO}(3)$ at R. Then there is a unique geodesic[1] in this manifold, $\gamma(t)$ such that $\gamma(0) = R$ and $\gamma'(0) = v$. The exponential map is then defined as $\exp_R(v) = \gamma(1)$. More formally

$$\begin{aligned} \text{Exp}_R \ &: \ \text{T}_\text{R}\text{SO}(3) \to \text{SO}(3) \\ \exp_R(\Omega R) &= R \exp(\Omega) \end{aligned} \tag{A.3}$$

The Logarithmic map is the inverse of the exponential map, and can be be defined as

$$\begin{aligned} \text{Log}_R \ &: \ \text{SO}(3) \to \text{T}_\text{R}\text{SO}(3) \\ \log_{R_1}(R_2) &= R_1 \log(R_1^\top R_2) \end{aligned} \tag{A.4}$$

The geodesic curve mentioned above can be written as $\gamma : t \to \exp_R(tR\Omega)$ since any tangent to $\text{SO}(3)$ at R is of the form $R\Omega$, with Ω skew-symmetric. Similarly, the geodesic from R_1 to R_2 is given by $\gamma : [0, 1] \to \text{SO}(3)$

$$\gamma(t) = \exp_{R_1}\big(t \log(R_1^\top R_2)\big)$$

It is easy to see that $\gamma(0) = R_1$ and $\gamma(1) = R_2$. We are interested in the length of this geodesic $L(\gamma)$, which can be calculated by integrating the norm of the velocity $\gamma'(t)$ over the entirety of the curve:

$$L(\gamma) = \int_0^1 \langle \gamma'(t), \gamma'(t) \rangle^{1/2} \, dt$$

By definition, being a geodesic implies that the length of the tangent vector is constant thus $\langle \gamma'(t), \gamma'(t) \rangle^{1/2}$ is obviously constant and in this case equal to $||\ \log\ (R_1^\top R_2)\ ||$. Therefore,

$$L(\gamma) = ||\log(R_1^\top R_2)|| \int_0^1 dt = ||\log(R_1^\top R_2)||$$

This length is known as the geodesic distance in $\text{SO}(3)$ between rotations R_1 and R_2,

$$\boxed{d_{\text{geo}} = ||\log(R_1^\top R_2)\ ||}$$

As it turns out, this distance is actually equal to the angle $\theta \in [0, \pi]$ of the rotation $R_1^\top R_2$. It can thus be

[1] Distance minimizing curve.

referred to as *angular distance*. As demonstrated by Hartley et al. [22], the distance $||R_1 - R_2||_F$, which is known as the *chordal distance*, can be easily related to the geodesic distance by the following formula

$$\boxed{d_{\text{chordal}} = ||R_1 - R_2||_F = 2\sqrt{2}\,\sin\big(d_{\text{geo}}/2\big)}$$

Alternatively, the distance between two rotations can also be defined in terms of their quaternionic representation. Let q_1 and q_2 be the unit quaternions representing rotations R_1 and R_2, respectively. Then

$$d_{\text{quat}}(R_1, R_2) = \min\big\{(q_1 - q_2), (q_1 + q_2)\big\} = 2\sin(\theta/4)$$

Another important concept is that of gradient on the Riemannian manifold, since it can be used in optimization problems.

Definition A.1.1 (Riemannian gradient on $\mathrm{SO}(3)$, Ma et al. [32])**.** *The Riemannian gradient* $\mathrm{grad}_R f$ *on the special orthogonal group manifold for a smooth function* $f : \mathrm{SO}(3) \to \mathbb{R}$ *is given by*

$$\begin{aligned}
&\mathrm{grad}_R f : \mathrm{SO}(3) \to \mathrm{T_R SO}(3) \\
&\mathrm{grad}_R f = \frac{\partial f}{\partial R} - R\Big(\frac{\partial f}{\partial R}\Big)^\top R
\end{aligned}$$

where $\frac{\partial f}{\partial R}$ denotes the Euclidean derivative of f w.r.t R. To see that the gradient lies in the tangent space, note that the formula can be rearranged as

$$R\left(R^\top \frac{\partial f}{\partial R} - \left(\frac{\partial f}{\partial R}\right)^\top R\right) = R\underbrace{\left(R^\top \frac{\partial f}{\partial R} - \left(R^\top \frac{\partial f}{\partial R}\right)^\top\right)}_{\text{skew}-\text{symmetric}}$$

which is in accordance with Eq. (A.2).

Probability distribution

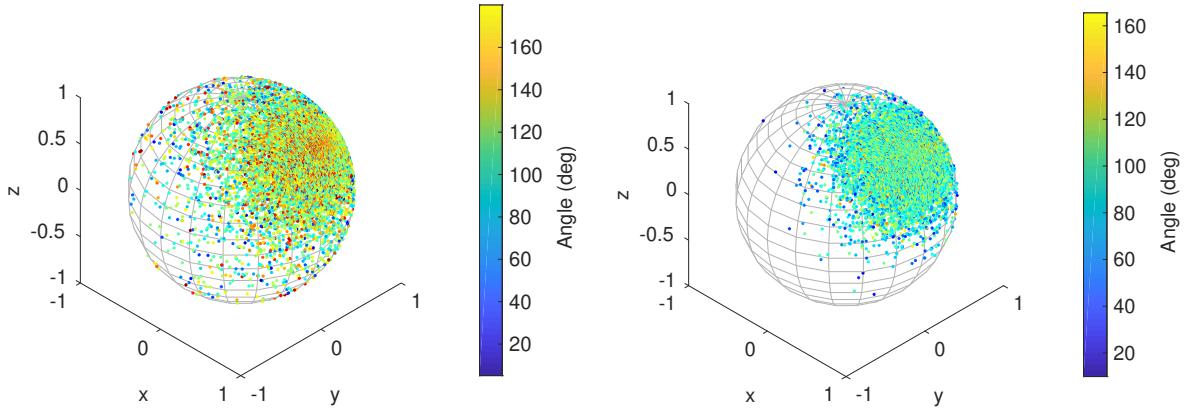
A probability distribution on $\mathrm{SO}(3)$ used frequently in the Robotics and Computer Vision communities is now presented. Other distributions exist in the literature, associated with different rotation representations, namely quaternions and vector-magnitude.

Definition A.1.2 (Isotropic Langevin distribution [19])**.** *The isotropic Langevin (or Von Mises-Fischer) distribution on* $\mathrm{SO}(3)$ *with mean* $S \in \mathrm{SO}(3)$ *and concentration parameter* $k \geq 0$ *has a Probability Density Function (PDF) given by*

$$\boxed{f_L(R) = \frac{1}{c_n(k)}\exp(k\; tr(S^\top R))}$$

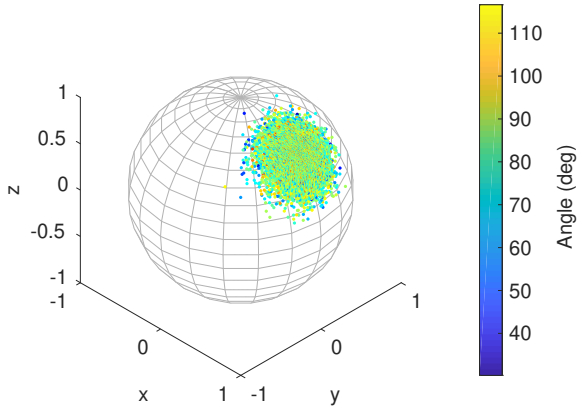
where $\frac{1}{c_n(k)}$ *is a normalization constant. This distribution will be denoted as* $\mathrm{Lang}(S, k)$.

In order to understand the effect of the parameters k and S in the PDF f_{L}, *rejection sampling* was used to generate random rotations following this PDF. This technique can be implemented by generating random rotations according to a uniform distribution on $\mathrm{SO}(3)$, and accepting them with a certain probability. Figs. A.1a to A.1d show a vector-magnitude representation of 10000 rotation samples obtained this way.

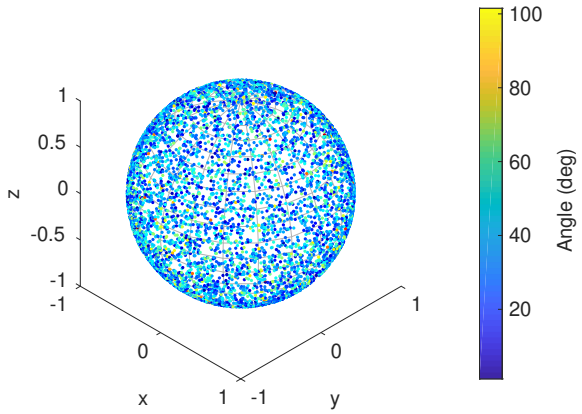


(a) $S = \mathrm{rot}([\frac{\pi}{2\sqrt{2}}\ 0\ \frac{\pi}{2\sqrt{2}}])\ \ k = 2$

(b) $S = \mathrm{rot}([\frac{\pi}{2\sqrt{2}}\ 0\ \frac{\pi}{2\sqrt{2}}])\ \ k = 5$



(c) $S = \mathrm{rot}([\frac{\pi}{2\sqrt{2}}\ 0\ \frac{\pi}{2\sqrt{2}}])\ \ k = 15$



(d) $S = I_3\ \ k = 5$

Figure A.1: Examples of rotations as vectors in $\mathbb{R}^3$ drawn from different isotropic Langevin probability distributions.

The first three images correspond to a mean rotation of $\pi/2$ rad around the axis $\left[\frac{1}{\sqrt{2}}\ 0\ \frac{1}{\sqrt{2}}\right]$. It can be observed that the probability distribution on the sphere becomes narrower as k increases (hence the name concentration parameter). For $k \to \infty$ we would have $R = S$ with probability 1. The magnitudes approach $\pi/2$ as k surges. The last image corresponds to a mean rotation equal to the identity and thus does not have a vector-magnitude representation. In this case the probability distribution becomes a circular normal distribution (or Von-Mises distribution), as described in Appendix B.1:

$$
\begin{aligned}
f_{\mathrm{L}}(I_3, k) &= \frac{1}{c_n(k)} \exp\left(k \operatorname{tr}\left(R\right)\right) \\
&= \frac{1}{c_n(k)} \exp\left(k(1 + 2\cos\left(\theta\right))\right) \\
&= \frac{e^k}{c_n(k)} \exp\left(2k\cos\left(\theta\right)\right) \propto f_{\mathrm{VM}}(\theta \mid 0, 2k)
\end{aligned} \tag{A.5}
$$

where $\theta \in [-\pi, \pi]$ is the rotation angle, obtained by using the equality $\operatorname{tr} R = 1 + 2\cos(\theta)$. The PDF in Eq. (A.5) behaves like an Normal distribution with variance $1/2k$ for large k and for the rotation angle θ, i.e. it is equivalent to sampling the rotation vector uniformly over the unit sphere and then independently sampling the rotation magnitude according to f_{VM}.

To end this section, we derive the ML estimate for the mean S of a *Langevin* distribution. In the case of a *Von Mises-Fischer* distribution on the hypersphere, this estimate can be obtained by differentiating the Lagrangian obtained from the log-likelihood with an added constraint. For more information the reader is referred to the work of Dhillon and Sra [33]. For the *Langevin* distribution on $\mathrm{SO}(3)$ a similar procedure would be hard to accomplish since enforcing the determinant constraint of $\mathrm{SO}(3)$ is difficult. Nonetheless, the log-likelihood can be maximized by solving the orthogonal Procrustes problem as is now demonstrated. Let $\{R_1, \ldots, R_n\}$ be a set of rotation samples drawn from f_L. These are deemed to be independent. Consequently, the likelihood can be written as

$$
L(S, k) = p(R_1, \ldots, R_n | S, k) = \prod_{i=1}^{n} \frac{1}{c_n(k)} \exp(\operatorname{tr} k\, S^T R_i)
$$

As is common when dealing with likelihood functions of independent measurements, the logarithm can be applied to eliminate the product,

$$
\log L(S, k) = c + \sum_{i=1}^{n} \operatorname{tr} k S^T R_i = c + k \operatorname{tr} S^\top \Big(\sum_{i=1}^{n} R_i \Big)
$$

where c denotes a constant. Since $k \geq 0$, maximizing $\log L(S, k)$ is equivalent to maximizing the trace in the expression, i.e.

$$
\underset{S \in \mathrm{SO}(3)}{\operatorname{argmax}} \log L(S, k) = \underset{S \in \mathrm{SO}(3)}{\operatorname{argmax}} \operatorname{tr} S^\top \Big(\sum_{i=1}^{n} R_i \Big)
$$

Let $\sum_{i=1}^{n} R_i = U \Sigma V^\top$ denote the SVD of the sum of rotation measurements, where U and V are orthonormal and Σ is a positive semidefinite diagonal matrix. Then,

$$
\underset{S \in SO(3)}{\operatorname{argmax}} \log L(S, k) = \underset{S \in SO(3)}{\operatorname{argmax}} \operatorname{tr} S^\top U \Sigma V^\top = \underset{S \in SO(3)}{\operatorname{argmax}} \operatorname{tr} \Sigma \underbrace{V^\top S^\top U}_{\text{orthonormal}} \tag{A.6}
$$

We can therefore conclude that this optimization corresponds to the orthogonal Procrustes problem [1]. The value of the trace is maximal when

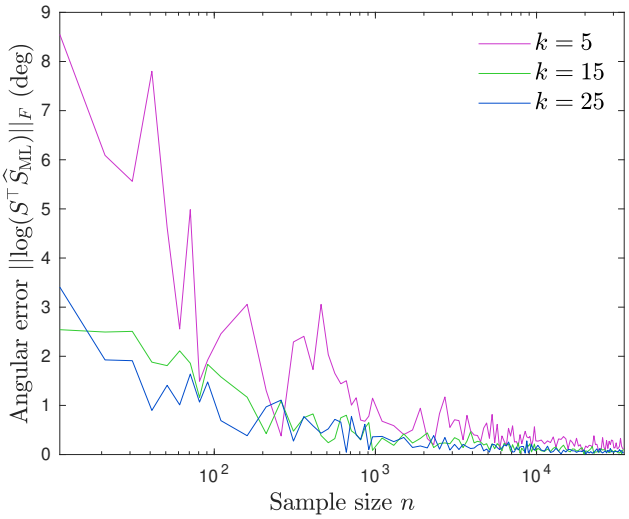


Figure A.2: Angular error of the isotropic Langevin maximum likelihood mean estimate for $S = I_3$, using samples of different size n and for three concentration parameters k.

$$V^\top S^\top U = I_3$$

And thus the solution is simply,

$$S = UV^\top$$

In summary, given n i.i.d. rotation measurements drawn from a *Langevin* distribution, the ML estimate of the mean rotation S is given by

$$\boxed{\widehat{S}_{\mathrm{ML}} = \left(\frac{1}{n}\sum_{i=1}^{n} R_i\right)_{\downarrow \mathrm{SO}(3)}}$$

where $\downarrow \mathrm{SO}(3)$ denotes the projection to $\mathrm{SO}(3)$. Fig. A.2 shows the convergence of the estimation error in degrees, using the geodesic distance in $\mathrm{SO}(3)$: $||\log\, S^\top \widehat{S}_{\mathrm{ML}}||_F$ as the sample size increases, for different concentration parameters and for $S = I_3$. Fig. A.3 illustrates the direction of the mean $S = \mathrm{rot}\left(\begin{bmatrix} \frac{\pi}{\sqrt{2}} & 0 & \frac{\pi}{\sqrt{2}} \end{bmatrix}\right)$ and the ML mean estimate $\widehat{S}_{\mathrm{ML}}$, using a vector representation in $\mathbb{R}^3$, for two values of k. As expected, for larger concentration parameters, the estimator is a better approximation of the true mean rotation.

A.2 Graph theory

Let $\mathcal{G} = (V, E)$ be a graph, where V denotes the set of vertices, with $|V|$ equal to its cardinality, i.e. the order of the graph. E corresponds to the set of edges and verifies $E \subseteq V \times V$. In the context of PGO we are interested in graphs which are both simple and undirected, i.e. without multiple edges incident on the same vertices and where there is no direction associated with each edge. It also makes sense

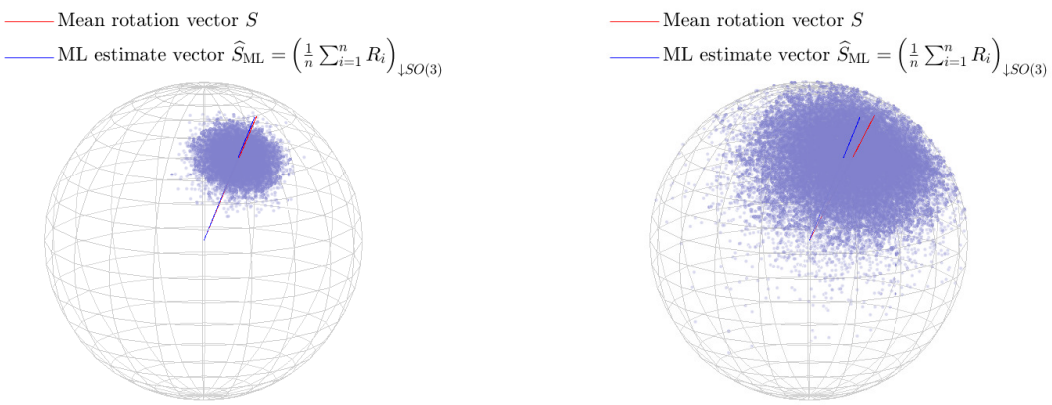


Figure A.3: Comparison of the ML mean estimate of $S = \mathrm{rot}\left([\pi/\sqrt{2}\ 0\ \pi/\sqrt{2}]\right)$ for $k = 20$ (left) and $k = 4$ (right) using 10000 rotation samples.

to restrict our analysis to connected graphs, i.e. graphs where it is possible to go from every vertex to any other vertex, since this is the basis of the error redistribution central to PGO. In this section some definitions and interesting results in graph theory used throughout the thesis are put forward.

Definition A.2.1 (Degree of a graph, *Diestel* [34])**.** *The degree of a vertex $v \in V$, here denoted by $\deg(v)$, is the number of edges incidents on v.*

Definition A.2.2 (Degree matrix)**.** *The degree matrix D of a simple graph $\mathcal{G} = (V, E)$ is a $|V| \times |V|$ diagonal matrix defined as*

$$D_{ii} := \deg(v_i)$$

The total degree of a graph $\deg(\mathcal{G})$ is equal to $\mathrm{tr}\ D$.

Definition A.2.3 (Adjacency matrix)**.** *The adjacency matrix of $\mathcal{G}$ is a $|V| \times |V|$ symmetrical matrix defined as follows*

$$A_{ij} := \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E \end{cases}$$

Definition A.2.4 (Laplacian)**.** *The Laplacian of $\mathcal{G}$ is a $|V| \times |V|$ symmetrical matrix defined as*

$$\mathcal{L} = D - A$$

where D is the degree matrix and A the adjacency matrix previously defined. The Laplacian is always positive-semidefinite and singular.

Theorem A.2.1 (Nullity of the Laplacian, Gross and Yellen. [35])**.** *Let $\mathcal{G}$ be a graph and $\mathcal{L}$ its associated Laplacian. Then $\mathrm{Nullity}(\mathcal{L})$ is equal to the number of connected components of $\mathcal{G}$.*

Definition A.2.5 (Fiedler value, Gross and Yellen. [35])**.** *The Fiedler value of $\mathcal{G}$, denoted as $F(\mathcal{G})$, also known as the algebraic connectivity of the graph is the second smallest eigenvalue of the Laplacian matrix. Its value is related to the overall connectedness of the graph.*

Appendix B

Auxiliary expressions and derivations

B.1 Von Mises distribution

The Von-Mises distribution with support $[-\pi, \pi]$, also known as the circular normal distribution has a PDF given by

$$\boxed{f_{\mathrm{VM}}(\theta|k,\mu) = \frac{1}{2\pi I_0(k)}\exp\big(k\,\cos(\theta-\mu)\big)} \tag{B.1}$$

where θ is the angle, μ denotes the mean, k the concentration parameter and $I_0(k)$ the modified Bessel function of order 0. For $k = 0$, this distribution is uniform. For large k, samples drawn according to this PDF become more clustered together and the distribution can be approximated by a normal distribution.

B.2 Matrix derivatives

Adapted from *Petersen et al.* [36].

$$\frac{d||Av||^2}{dv} = \frac{d\big(||Av||^2\big)}{d\big(Av\big)}\frac{d\big(Av\big)}{dv} = 2(Av)^\top A \tag{B.2}$$

$$\frac{\partial||A-BC||_F^2}{\partial A} = 2(A-BC) \tag{B.3}$$

$$\frac{\partial||A-BC||_F^2}{\partial B} = -2C^\top(A-BC) \tag{B.4}$$

B.3 Derivation of Equation 4.2

$$\begin{aligned}\log L(\theta, y) &= -\frac{1}{2\sigma_t^2}\sum_{i,j\in E}||\widetilde{t}_{ij} - t_i + R_i R_j^\top t_j||^2 + \frac{1}{\sigma_R^2}\sum_{i,j\in E}\text{tr}\,\widetilde{R}_{ij}R_j R_i^\top \\ &= \underbrace{-\frac{1}{2\sigma_t^2}\sum_{i,j\in E}(\widetilde{t}_{ij}^\top - t_i^\top + t_j^\top R_j R_i^\top)(\widetilde{t}_{ij} - t_i + R_i R_j^\top t_j)}_{A} + \frac{1}{\sigma_R^2}\sum_{i,j\in E}\text{tr}\,\widetilde{R}_{ij}R_j R_i^\top\end{aligned}$$

$$\begin{aligned}A = -\frac{1}{2\sigma_t^2}\sum_{i,j\in E}&||\widetilde{t}_{ij}||^2 - \widetilde{t}_{ij}^\top t_i + \widetilde{t}_{ij}^\top R_i R_j^\top t_j - t_i^\top \widetilde{t}_{ij} + ||t_i||^2 - t_i^\top R_i R_j^\top t_j \\ &+ t_j^\top R_j R_i^\top \widetilde{t}_{ij} - t_j^\top R_j R_i^\top t_i + ||t_j||\end{aligned}$$

$$A = f(t_i) - \frac{1}{\sigma_t^2}\sum_{i,j\in E}\widetilde{t}_{ij}^\top R_i R_j^\top t_j - t_i^\top R_i R_j^\top t_j$$

$$\log L(\theta, y) = -\frac{1}{\sigma_t^2}\sum_{i,j\in E}(\widetilde{t}_{ij}^\top - t_i^\top)R_i R_j^\top t_j + \frac{1}{\sigma_R^2}\sum_{i,j\in E}\text{tr}\,\widetilde{R}_{ij}R_j R_i^\top + f(t_i)$$

B.4 Log-likelihood derivative w.r.t. translations

The derivative of the log-likelihood w.r.t to the k-th translation t_k is given by

$$\frac{\partial}{\partial t_k}\log L(\theta|y) = \frac{1}{2\sigma_t^2}\sum_{i,j\in E}\frac{\partial}{\partial t_k}||R_i R_j^\top t_j + \tilde{t}_{ij} - t_i||^2$$

Using the norm derivatives provided in Appendix B.2 we have

$$\frac{\partial}{\partial t_k}||R_i R_j^\top t_j + \tilde{t}_{ij} - t_i||^2 = 2(R_i R_j^\top t_j + \tilde{t}_{ij} - t_i)^\top \frac{\partial}{\partial t_k}(R_i R_j^\top t_j + \tilde{t}_{ij} - t_i)$$

The vector derivative can be written using the Kronecker delta as follows

$$\frac{\partial}{\partial t_k}(R_i R_j^\top t_j + \tilde{t}_{ij} - t_i) = R_i R_j \delta_{jk} - \delta_{ik} I_3$$

Hence

$$\frac{\partial}{\partial t_k}\log L(\theta|y) = \frac{1}{\sigma_t^2}\sum_{i,j\in E}(R_i R_j^\top t_j + \tilde{t}_{ij} - t_i)^\top (R_i R_j^\top \delta_{jk} - \delta_{ik} I_3)$$

To simplify the expression, let us split it in two

$$\frac{\partial}{\partial t_k}\log L(\theta|y) = d_i + d_j$$

$$\begin{aligned} d_i &= \frac{1}{\sigma_t^2}\sum_{i,j\in E}\delta_{ik}(R_i R_j^\top t_j + \tilde{t}_{ij} - t_i)^\top \\ &= \frac{1}{\sigma_t^2}\sum_{j\in E(v_k)}(R_k R_j^\top t_j + \tilde{t}_{kj} - t_k)^\top \end{aligned}$$

and similarly

$$\begin{aligned} d_j =& \frac{1}{\sigma_t^2}\sum_{i,j\in E}\delta_{jk}(R_i R_j^\top t_j + \tilde{t}_{ij} - t_i)^\top (R_i R_j^\top) \\ =& \frac{1}{\sigma_t^2}\sum_{i\in E(v_k)}(R_i R_k^\top t_k + \tilde{t}_{ik} - t_i)^\top (R_i R_k^\top) \\ =& \frac{1}{\sigma_t^2}\sum_{i\in E(v_k)}\left(t_k + R_k R_i^\top \tilde{t}_{ik} - R_k R_i^\top t_i\right)^\top \end{aligned}$$

The dummy indices j and i in d_i and d_j can be set to the same index and both these terms of the derivative can be added together

$$\begin{aligned} \frac{\partial}{\partial t_k}\log L(\theta|y) &= \frac{1}{\sigma_t^2}\sum_{j\in E(v_k)} -(R_k R_j^\top t_j + \tilde{t}_{kj} - t_k)^\top + \left(t_k + R_k R_j^\top \tilde{t}_{jk} - R_k R_j^\top t_j\right)^\top \\ &= \frac{1}{\sigma_t^2}\Big(\sum_{j\in E(v_k)} -2R_k R_j^\top t_i + 2t_k - \widetilde{t}_{kj} + R_k R_j^\top \widetilde{t}_{jk}\Big)^\top \end{aligned} \tag{B.6}$$

Here we use following equality $\widetilde{t}_{jk} = -\widetilde{R}_{jk}\widetilde{t}_{kj}$ to arrive at

$$\begin{aligned} \frac{\partial}{\partial t_k}\log L(\theta|y) =& \frac{1}{\sigma_t^2}\Big(\sum_{j\in E(v_k)} -2R_k R_j^\top t_i + 2t_k - \widetilde{t}_{kj} - R_k R_j^\top \widetilde{R}_{jk}\widetilde{t}_{kj}\Big)^\top \\ =& \frac{2}{\sigma_t^2}\Big(\sum_{j\in E(v_k)} R_k R_j^\top t_i - t_k + \frac{1}{2}\big(\widetilde{t}_{kj} + R_k R_j^\top \widetilde{R}_{jk}\widetilde{t}_{kj}\big)\Big)^\top \end{aligned} \tag{B.7}$$

By placing t_k outside the sum we finally get

$$\frac{\partial}{\partial t_k}\log L(\theta|y) = \frac{2}{\sigma_t^2}\Big(\sum_{j\in E(v_k)}\Big(R_k R_j^\top t_i + \frac{1}{2}\big(\widetilde{t}_{kj} + R_k R_j^\top \widetilde{R}_{jk}\widetilde{t}_{kj}\big)\Big) - \deg(v_k)t_k\Big)^\top \tag{B.8}$$

B.5 Euclidean derivative of the rotation averaging cost function

$$\begin{aligned}
\frac{\partial \psi}{\partial R_k} &= \sum_{(i,j)\in E} \frac{\partial}{\partial R_k}\, d^2_{\text{chordal}}(R_i, \widetilde{R}_{ij} R_j^\top) \\
&= \sum_{(i,j)\in E} \frac{\partial}{\partial R_k}\, d^2_{\text{chordal}}(R_i, \widetilde{R}_{ij} R_j)\delta_{ik} + \frac{\partial}{\partial R_k}\, d^2_{\text{chordal}}(R_i, \widetilde{R}_{ij} R_j)\delta_{jk} \\
&= \sum_{(k,j)\in E} \frac{\partial}{\partial R_k} \left|\left|R_k - \widetilde{R}_{kj} R_j\right|\right|_F^2 + \sum_{(i,k)\in E} \frac{\partial}{\partial R_k} \left|\left|R_i - \widetilde{R}_{ik} R_k\right|\right|_F^2 \\
&= 2 \sum_{(k,j)\in E} (R_k - \widetilde{R}_{kj} R_j) + 2 \sum_{(i,k)\in E} (R_k - \widetilde{R}_{ik}^\top R_i)
\end{aligned}$$

B.6 Rotation anchoring

Let $R_{\mathcal{C}} = V_{\mathcal{C}_{\downarrow \text{SO}(3)}}$ denote the solution produced by EigenRA, where the columns of $V_{\mathcal{C}} \in \mathbb{R}^{3n\times 3}$ correspond to eigenvectors of $(\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}}$. One possible way of obtaining the rotation estimates w.r.t. a known reference frame, e.g. the first rotation, is to force the corresponding 3×3 block, hereafter denoted by V_1, to be the identity

$$R_{\mathcal{C}} = (V_{\mathcal{C}} V_1^{-1})_{\downarrow \text{SO}(3)}$$

In the case of noiseless measurements, such operation is valid, since the three eigenvectors have the same eigenvalues, and thus the columns of the resulting matrix $V_{\mathcal{C}} V_1^{-1}$ are still a basis for the corresponding eigenspace. On the contrary, if the eigenvalues differ by a considerable amount, the solution may be affected since the columns are no longer eigenvectors, as demonstrated below.

$$\big((\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}}\big) V_{\mathcal{C}} V_1^{-1} = V_{\mathcal{C}} \Lambda V_1^{-1}$$

A different strategy consists of forcing the first rotation to be the identity beforehand. Let

$$L = (\mathcal{L} \otimes \mathbb{1}_{3\times 3}) \circ \widetilde{R}_{\mathcal{B}} = \begin{bmatrix} I_3 & L_1^\top \\ L_1 & L_2 \end{bmatrix}$$

By setting $R_1 = I_3$, solving for rotations can then be stated as finding $R_{2:n}$ such that

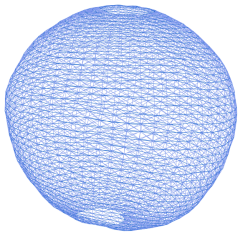
$$\begin{bmatrix} I_3 & L_1^\top \\ L_1 & L_2 \end{bmatrix} \begin{bmatrix} I_3 \\ R_{2:n} \end{bmatrix} = 0$$

This system of equations does not have a solution, unless $\sigma_R = 0$. However, since L_2 is symmetric and nonsingular (Conjecture 3.2.1), the equation $L_1 + L_2 R_{2:n} = 0$ can be solved for $R_{2:n}$ via Cholesky decomposition of L_2. In general, this solution will not verify the first equation $I_3 + L_1^\top R_{2:n}$, which can be discarded. The resulting rotation estimates are equal to those of the chordal relaxation initialization mentioned in Section 2.2.3.

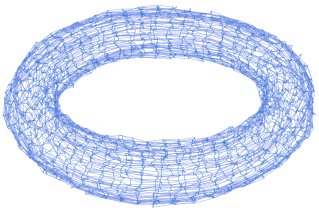
Appendix C

Pose graph images

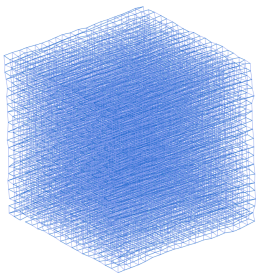
C.1 EigenMA applied to SLAM datasets



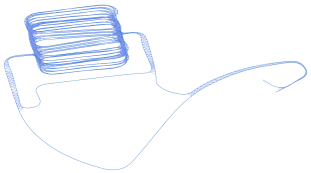
(a) Sphere



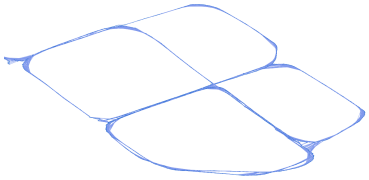
(b) Torus3D



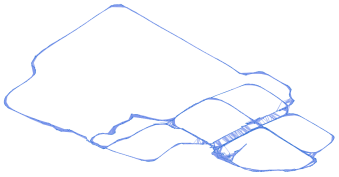
(c) Grid3D



(d) Garage



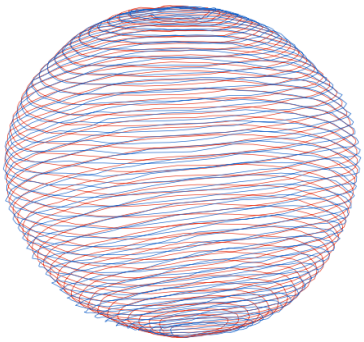
(e) Cubicle



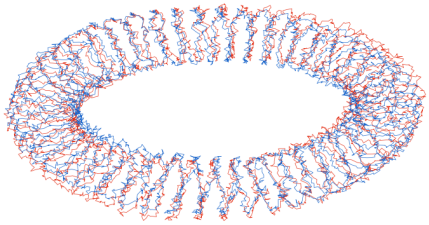
(f) Rim

Figure C.1: Optimization of SLAM datasets using EigenMA. Camera trajectory and loop closures.

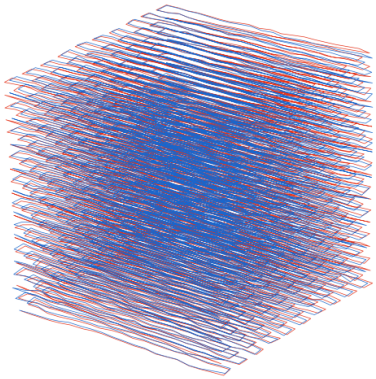
C.2 EigenMA vs. Gauss-Newton in SLAM datasets



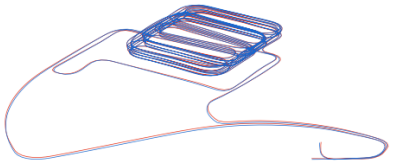
(a) Sphere



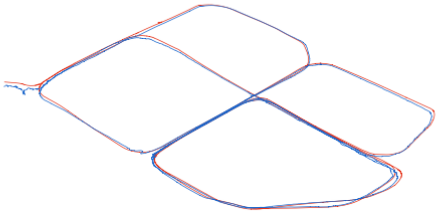
(b) Torus3D



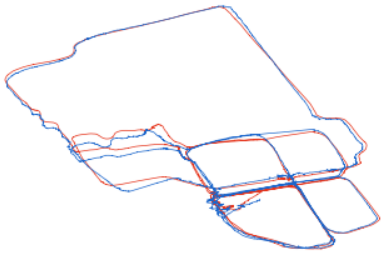
(c) Grid3D



(d) Garage



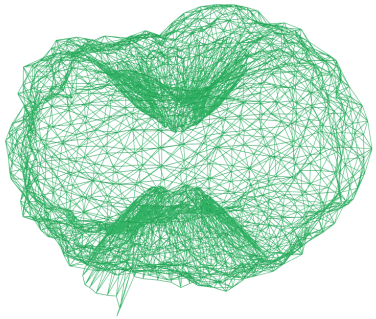
(e) Cubicle



(f) Rim

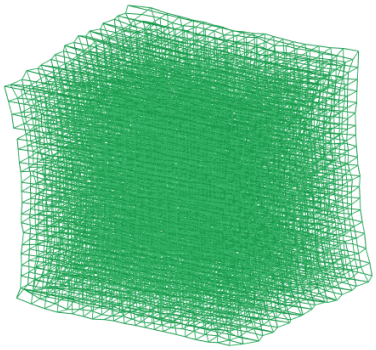
Figure C.2: Camera trajectory comparison between EigenMA (blue) and 10 Gauss-Newton iterations using g2o (red) initialized from EigenMA.

C.3 EIG-SE(3) applied to SLAM datasets



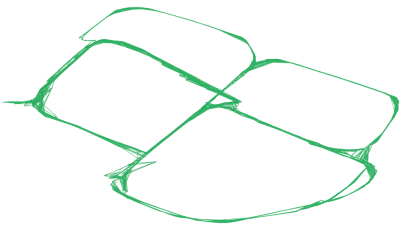
(a) Sphere

(b) Torus3D



(c) Grid3D

(d) Garage

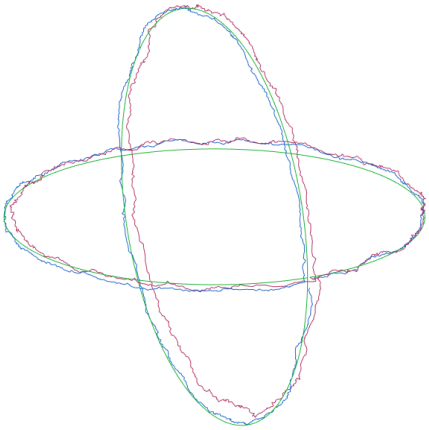


(e) Cubicle

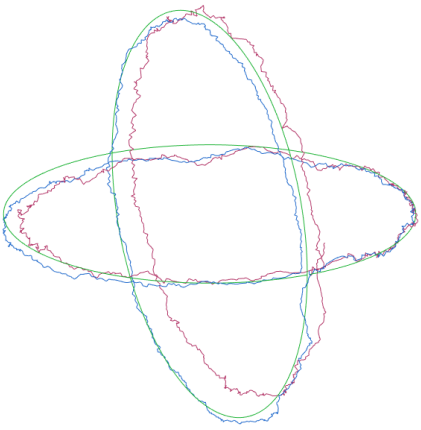
(f) Rim

Figure C.3: Camera trajectories and loop closures of EIG-SE(3) applied to popular SLAM datasets.

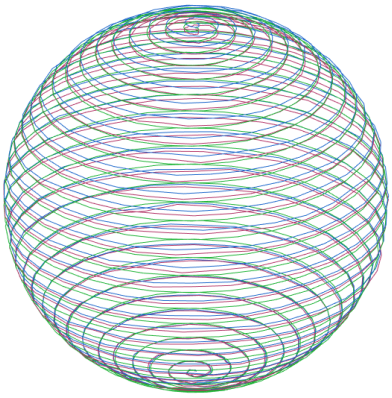
C.4 Failure modes of EIG-SE(3)



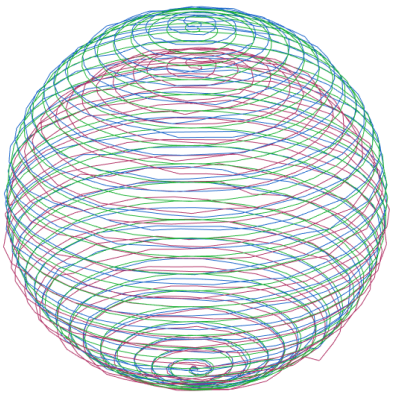
(a) $\sigma_R = 2.5$ deg $\quad \sigma_t = 0.005$



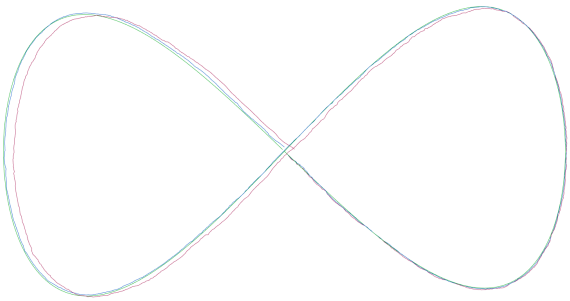
(b) $\sigma_R = 4.0$ deg $\quad \sigma_t = 0.01$



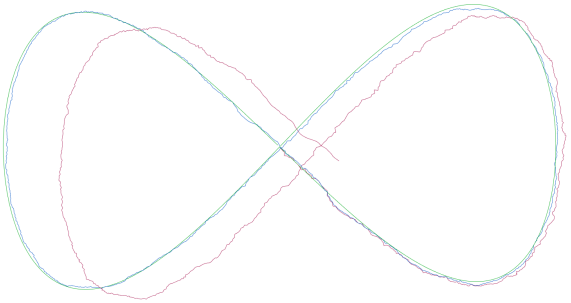
(c) $\sigma_R = 1.0$ deg $\quad \sigma_t = 0.005$



(d) $\sigma_R = 2.5$ deg $\quad \sigma_t = 0.005$



(e) $\sigma_R = 1.0$ deg $\quad \sigma_t = 0.005$



(f) $\sigma_R = 2.5$ deg $\quad \sigma_t = 0.005$

Figure C.4: Failure modes of EIG-SE(3) [15] (red) for synthetic pose graphs with 1000 poses and 99% of missing data. In blue we represent our EigenMA solution and in green the ground truth trajectory. EIG-SE(3)'s solution degrades quickly as the measurements become noisier.

Appendix D

Dense scene reconstruction

This appendix contains an example of our PGO solution, EigenMA, applied to dense 3D scene reconstruction, using 1124 images from the Burghers RGB-D dataset [17] which comprises 6 statues approximately 2 meters tall.

Since the images form a timeseries, our RGB-D registration pipeline starts by estimating relative transformations between adjacent frames (akin to dead reckoning) and then attempts to increase the connectivity of the pose graph by estimating additional edges, selected according to a specific heuristic. The relative transformation between any two frames is computed by combining two Computer Vision classics: SIFT and RANSAC. The latter is used to identify inlier keypoint matches amongst the pool of matches retrieved by the former. These inlier matches yield the relative transformation estimate via Procrustes analysis. Table D.1 provides an overview of the optimization parameters. Fig. D.1 represents the camera trajectory and the edges of the graph. Finally, Fig. D.2 shows the reconstructed 3D scene after downsampling and denoising.

Number of point clouds $\|V\|$	1124
Number of relative transformations $\|E\|$	4118
Total number of points	169163102
Depth cutoff (m)	2.0
SIFT specifications	Default
RANSAC maxiter	7.0e+03
RANSAC RMSE threshold	3.0e-06
RANSAC min no. matches	100
Average pairwise RMSE	2.816e-06
Average pairwise no. inliers	150
Average pairwise CPU time (s)	0.5s
EigenRA cost function lower bound	7.372e-02
EigenRA minimum cost	7.375e-02
Average symmetry $\bar{s}$ (Validation of Hyp. 4.1.1)	0.99
EigenMA CPU time (s)	0.03
Downsample grid size (MATLAB)	0.007
Denoise no. neighbors (MATLAB)	30

Table D.1: Specifications of our RGB-D registration pipeline, optimization parameters and results.

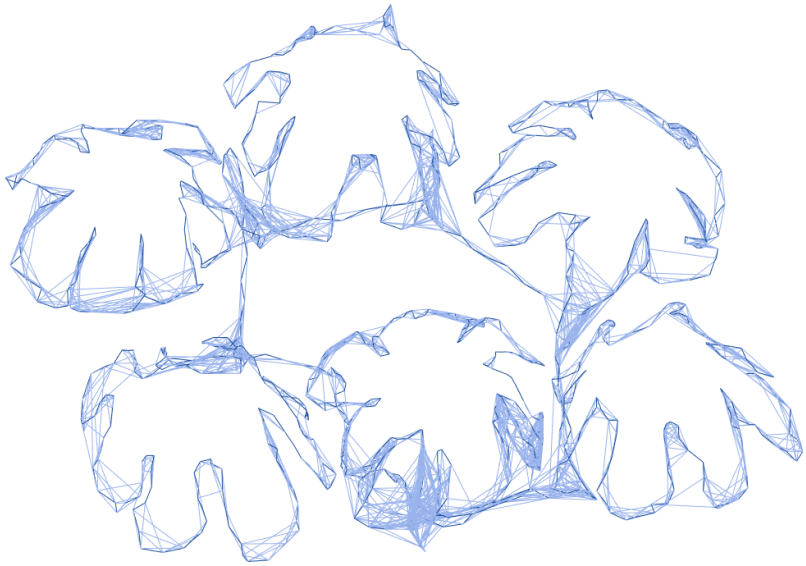


Figure D.1: Burghers dataset [17] camera trajectory computed by EigenMA and loop closures.

Figure D.2: Burghers dataset [17] dense 3D reconstruction using our RGB-D registration pipeline.