

# Fast Pose Graph Optimization via Krylov-Schur and Cholesky Factorization

Gabriel Moreira    Manuel Marques    João Paulo Costeira  
 Institute for Systems and Robotics, Instituto Superior Técnico  
 Av. Rovisco Pais, Lisboa, Portugal  
 {gmoreira, manuel, jpc}@isr.tecnico.ulisboa.pt

## 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. The vast majority of PGO approaches rely on iterative techniques which refine an initial estimate until convergence is achieved. On the other hand, recent works have identified a global constraint which has cast this problem into the matrix completion domain. The success which both these formulations have had in computing accurate solutions efficiently has been overshadowed by large-scale industrial applications such as autonomous flight, self-driving cars and smart-cities, where it is necessary to fuse numerous images covering large areas but where each one of them has few pairwise observations. We propose a highly efficient algorithm to solve PGO which leverages the sparsity of the data by combining the Krylov-Schur method for spectral decomposition with Cholesky LDL factorization. Our method allows for high scalability, low computational cost and high precision, simultaneously.*

## 1. Introduction

The registration of 3D point sets obtained by LiDARs, RGB-D and stereo cameras is one of the core problems in Robotics and Computer Vision, with applications ranging from dense scene reconstruction to localization. For two points sets, Iterative Closest Point (ICP) [3, 20] is a common procedure to solve the registration problem. This class of algorithms requires an initialization and iterates between estimating point correspondences and computing the optimal transformation between them.

Consider now that we have a set of point clouds corresponding to different views of the same scene. The different point sets may be obtained via an array of 3D scanners. Alternatively, they may represent the visual data acquired by a single observer as it moves through space. Both sit-



Figure 1: Camera trajectory, loop closures and 3D reconstruction of the Burghers dataset [25] using our algorithm.

uations have seen a rise in popularity recently, 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. For known point correspondences, the rigid transformations that allow for an optimal registration of the point sets can be solved for in closed-form via Generalized Procrustes Analysis [9]. For unknown correspondences, optimization strategies analogous to ICP have been proposed [22]. However, such methods are computationally inefficient even for moderately sized registration tasks. To circumvent this issue, Pose Graph Optimization (PGO) is the method of choice in Simultaneous Localization and Mapping (SLAM) and 3D reconstruction (Fig. 1).

PGO consists of estimating a set of rigid transformations, or poses, given a subset of pairwise measurements of their ratios. The latter can be computed *e.g.*, by ICP algorithms initialized with 2D image matches. By associating each rigid transformation measurement  $\tilde{M}_{ij}$ , from point cloud  $i$  to point cloud  $j$ , to an edge  $(i, j) \in E$ , we obtain a simple graph  $\mathcal{G} = (V, E)$  which, if connected, can be used to derive a cost function that is minimized by the global transformations  $M_i$ , for  $i \in V$ , that best fit the measured data.

An example of a possible PGO statement is

$$\arg \min_{\{\mathbf{M}_1, \dots, \mathbf{M}_n\} \in \text{SE}(3)^n} \sum_{(i,j) \in E} \|\widetilde{\mathbf{M}}_{ij} - \mathbf{M}_i \mathbf{M}_j^{-1}\|_F^2. \quad (1)$$

In real world applications, guaranteeing globally optimal solutions to PGO is paramount. However, this optimization task is a high-dimensional and non-convex problem. Second-order methods bootstrapped with robust initializations can converge to the sought-after optimum. Nevertheless, they do not scale well. On the other hand, even if certain relaxations of the original problem allow for more efficient implementations, their solutions may be far from those of the original problem.

We address PGO in the context of point cloud registration. The desiderata for our algorithm are: scalability, efficiency and accuracy. While existing methods, namely those proposed in [2, 7, 17, 19] satisfy a subset of these requirements, they usually incur an efficiency accuracy trade-off. This paper improves upon the spectral synchronization algorithm proposed by Arrigoni et al. [2] and employs the optimality verification techniques in SO(3) put forward by Eriksson et al. [12] in order to better approximate the global optimum of the Maximum Likelihood (ML) function derived by Carlone et al. [6].

We developed an efficient algorithm to solve PGO in SO(3), also known as rotation averaging, and SE(3) that accounts for the high degree of sparsity that is common to the applications we are considering. Our solution is computed in two stages. The first consists of the Krylov-Schur algorithm [21] for spectral decomposition of large sparse matrices. The second boils down to solving a sparse linear system via Cholesky LDL factorization. Not only is our method considerably faster than the state-of-the-art but also we show empirically that our solution is either optimal or lies in the basin of attraction of the global optimum.

A C++ implementation of our algorithm is available online: <https://github.com/gabmoreira/maks>

## 2. Related work

The literature on PGO can be segmented into three different clusters: Maximum Likelihood Estimation (MLE) via nonlinear iterative solvers; suboptimal relaxations which do not guarantee local optimality but can be used to bootstrap other methods; optimality verification and globally optimal methodologies.

MLE is arguably the most popular approach to PGO. The optimization problem it gives rise to, which is both non-convex and high-dimensional, is usually tackled via state-of-the-art Gauss-Newton and Levenberg-Marquardt methods. Noteworthy examples of such solvers are g2o by Kummerle et al. [15] and GTSAM by Dellaert et al. [10]. To ensure global convergence however, these techniques rely

on initializations in the basin of attraction of the global optimum. This problem has been addressed by Carlone et al. [6], who studied different rotation initialization techniques.

In SO(3), Tron et al. [23] established a link between graph consensus algorithms and the Riemannian gradient descent on the SO(3) manifold. In spite of its good convergence properties, this technique is arguably slower than most approaches. Martinec et al. [17] addressed the same problem by means of a chordal relaxation, whereby the solution to a least-squares problem is projected to SO(3). Despite its inherent suboptimality, it scales well and can be used to initialize iterative solvers. Other remarkable works in rotation averaging include the seminal paper by Hartley et al. [14], a Quasi-Newton method set forth by Chatterjee et al. [8] and more recently, a deep learning approach by Purkait et al. [18].

In contrast to the formulations mentioned hitherto, Arrigoni et al. [1] have cast PGO into the Low-Rank and Sparse (LRS) decomposition domain. The relaxation underlying this methodology allows it to work both in SE(3) and SO(3), but lends itself to invariably suboptimal solutions. Foregoing the sparse term, introduced to capture outliers, this approach degenerates to the well studied problem of low-rank matrix completion [5] [13]. In another paper, the same authors derived a different relaxation which admits a closed-form solution that can be computed efficiently via spectral decomposition [2].

In recent literature, globally optimal solutions have been the focus of extensive research. Carlone et al. [6] derived the Lagrangian dual problem for PGO in SE(3), which can be used to validate an estimate as the global optimum and, in certain cases, allows for the direct retrieval of the optimal solution. A similar strategy was adopted to solve rotation averaging in SO(3) by Eriksson et al. [12], who put forward a method to solve the semidefinite program corresponding to the dual problem. Alternatively, Dellaert et al. [11] perform this optimization task on manifolds of higher dimension than SO(3). Their algorithm yields optimal solutions under certain assumptions. A faster and certifiably correct approach for PGO in SE(3) via a semidefinite relaxation was proposed by Rosen et al. [19].

While PGO research seems to converge toward globally optimal algorithms, these are still too computationally expensive for real-time applications. As we will demonstrate, under a high Signal-to-Noise Ratio (SNR), our solution is a good approximation of the global optimum and can be computed in a fraction of the time of state-of-the-art methods.

## 3. Proposed approach

Let  $\mathcal{G} = (V, E)$  be a simple and connected graph with  $|V| = n$  the number of poses and  $\{\widetilde{\mathbf{R}}_{ij}, \hat{t}_{ij}\}$  for  $(i, j) \in E$  the rigid transformation measurement from pose  $i$  to pose  $j$ . To render algebraic manipulation more tractable we will

henceforth make use of the following block-matrix notation. Let  $t \in \mathbb{R}^{3n}$  and  $\mathbf{R} \in \text{SO}(3)^n \subset \mathbb{R}^{3n \times 3}$  be defined as

$$t = [t_1^\top \quad \dots \quad t_n^\top]^\top \quad (2)$$

$$\mathbf{R} = [\mathbf{R}_1^\top \quad \dots \quad \mathbf{R}_n^\top]^\top, \quad \mathbf{R}_i \in \text{SO}(3) \quad (3)$$

with  $\{\mathbf{R}_i, t_i\}_{i=1, \dots, n}$  the rigid transformation corresponding to the  $i$ -th pose. Let  $\theta = \{\mathbf{R}, t\}$  and  $y = \{\tilde{\mathbf{R}}_{ij}, \tilde{t}_{ij}\}_{(i,j) \in E}$  be the set of parameters and observations, respectively. Assuming an isotropic Gaussian generative noise model with variance  $\sigma_t^2$  for the translation measurements, an isotropic Langevin noise model with concentration parameter  $1/\sigma_R^2$  for the rotations [4] and interdependence amongst the random variables involved, the log-likelihood function is given by

$$\begin{aligned} \log L(\theta|y) = & -\frac{1}{2\sigma_t^2} \sum_{(i,j) \in E} \|\tilde{t}_{ij} - t_i + \mathbf{R}_i \mathbf{R}_j^\top t_j\|^2 \\ & + \frac{1}{\sigma_R^2} \sum_{(i,j) \in E} \text{tr}(\tilde{\mathbf{R}}_{ij} \mathbf{R}_j \mathbf{R}_i^\top). \end{aligned} \quad (4)$$

PGO can be formulated as seeking  $\theta^* = \{\mathbf{R}^*, t^*\}$  that minimizes the negative log-likelihood, *i.e.*,

$$\theta^* = \arg \min_{\theta \in \text{SO}(3)^n \times \mathbb{R}^{3n}} -\log L(\theta|y). \quad (5)$$

The optimization task from (5) is high-dimensional and non-convex. Notwithstanding, provided there is an estimate  $\hat{\mathbf{R}}$  close to  $\mathbf{R}^*$ , solving for translations is a least-squares problem. We can thus ponder the validity of separating the optimization into two subproblems:  $\mathcal{R}$ , which is known in the literature as rotation averaging and  $\mathcal{T}$ , which computes the set of optimal translations for a given  $\hat{\mathbf{R}}$ .

$$\mathcal{R} : \arg \max_{\mathbf{R} \in \text{SO}(3)^n} \text{tr}(\tilde{\mathbf{R}}_{ij} \mathbf{R}_j \mathbf{R}_i^\top) \quad (6)$$

$$\mathcal{T} : \arg \min_{t \in \mathbb{R}^{3n}} \sum_{(i,j) \in E} \|\tilde{t}_{ij} - t_i + \hat{\mathbf{R}}_i \hat{\mathbf{R}}_j^\top t_j\|^2 \quad (7)$$

Provided there is a high SNR, which we formalize as

$$\forall (i, j) \in E : \tilde{t}_{ij} \approx t_i^* - \tilde{\mathbf{R}}_{ij} t_j^*, \quad (8)$$

the aforementioned separation of (5) is valid (see supplementary material). In computer vision applications *e.g.*, RGB-D registration, (8) is often verified since the relative transformations can be accurately estimated by means of ICP algorithms initialized from 2D image matches. In the subsequent sections, we will present a solution to problem  $\mathcal{R}$ , which we then use to solve  $\mathcal{T}$ .

### 3.1. Rotation averaging via spectral decomposition

In order to solve  $\mathcal{R}$ , we will show that a good approximation of the global optimum  $\mathbf{R}^*$  can be found by means of spectral decomposition, under the high SNR hypothesis we put forward. Furthermore, the respective eigenvalues, which are implicitly computed in the process, allow us to assess the optimality of the solution.

We will stack the rotation measurements  $\tilde{\mathbf{R}}_{ij}$  in a symmetric block matrix  $\tilde{\mathbf{R}} \in \mathbb{R}^{3n \times 3n}$ , defined as

$$\tilde{\mathbf{R}} = \begin{cases} \tilde{\mathbf{R}}_{ij} & \text{if } (i, j) \in E \\ \mathbf{I}_3 & \text{if } i = j \\ \mathbf{0}_3 & \text{otherwise,} \end{cases} \quad (9)$$

where  $\mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{0}_3 \in \mathbb{R}^{3 \times 3}$  denote the identity matrix and null matrix, respectively. By doing so, we can restate rotation averaging  $\mathcal{R}$  as seeking  $\mathbf{R}^*$  such that

$$\mathbf{R}^* = \arg \min_{\mathbf{R} \in \text{SO}(3)^n} -\text{tr}(\mathbf{R}^\top \tilde{\mathbf{R}} \mathbf{R}). \quad (10)$$

Drawing from the work of Eriksson et al. [12], let  $\Lambda \in \mathbb{R}^{3n \times 3n}$  be a symmetric diagonal block matrix defined as

$$\Lambda = \begin{bmatrix} \Lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \Lambda_n \end{bmatrix}. \quad (11)$$

For all stationary points  $\mathbf{R}^*$ , there will be  $\Lambda^*$  such that

$$(\Lambda^* - \tilde{\mathbf{R}}) \mathbf{R}^* = 0. \quad (12)$$

This estimate will be the global optimum if

$$\Lambda^* - \tilde{\mathbf{R}} \succeq 0. \quad (13)$$

According to [2], in the trivial case with  $\sigma_R = 0$ , the optimal Lagrange multiplier is  $\Lambda^* = \mathbf{D} \otimes \mathbf{I}_3 + \mathbf{I}_{3n}$ , where  $\mathbf{D} \in \mathbb{R}^{n \times n}$  is the graph degree matrix. In fact, the symmetric matrix

$$\mathbf{S} := \Lambda^* - \tilde{\mathbf{R}} = (\mathcal{L} \otimes \mathbf{J}_3) \circ \tilde{\mathbf{R}}, \quad (14)$$

where  $\mathcal{L} \in \mathbb{R}^{n \times n}$  denotes the graph Laplacian and  $\mathbf{J}_3 \in \mathbb{R}^{3 \times 3}$  an all-ones matrix, is positive semidefinite, with rank  $3(n-1)$ . Moreover, its nullspace intersects  $\text{SO}(3)^n$ .

For noisy measurements, we found empirically that the expected value of the smallest eigenvalue of  $\mathbf{S}$ , converges monotonically to a positive scalar, for increasing  $\sigma_R$ . Therefore, while in general  $\mathbf{S}$  becomes positive definite for noisy measurements, within the validity of the high SNR hypothesis, its smallest eigenvalue should be close to zero. Under this assumption, there will be  $\hat{\mathbf{R}} \in \text{SO}(3)^n$  such that

$$\mathbf{S} \hat{\mathbf{R}} \approx 0. \quad (15)$$

Our solution to problem  $\mathcal{R}$  consists thus of computing the three eigenvectors of  $\mathbf{S}$  corresponding to the smallest eigenvalues and then projecting them to  $\text{SO}(3)^n$  by solving  $n$  orthogonal Procrustes problems. This estimate can be accepted as the global optimum, provided the absolute value of smallest eigenvalue lies below a positive threshold.

**For large sparse matrices** (a common occurrence in SLAM problems), a small subset of eigenvectors can be computed via the Krylov-Schur algorithm [21], or equivalently, the restarted Lanczos method since the matrix is symmetric. In order to compute the three eigenvectors corresponding to the smallest eigenvalues (in absolute value) of  $\mathbf{S}$ , this method starts by building an orthonormal basis  $\{u_1, \dots, u_p\}$  for the Krylov subspace of

$$(\mathbf{S} - \sigma \mathbf{I}_{3n})^{-1}, \quad (16)$$

where  $\sigma$  denotes the spectral shift, in our case close to zero. This is accomplished by iteratively computing  $v_{k+1}$  via

$$v_{k+1} = (\mathbf{S} - \sigma \mathbf{I}_{3n})^{-1} u_k \quad (17)$$

and then orthogonalizing it against the previous vectors to produce  $u_{k+1}$ . We can leverage the symmetry of the shifted matrix in (17) by computing its Cholesky LDL factorization beforehand and then using it to solve for  $v_{k+1}$ .

These iterations represent a dominant fraction of the total computational cost of the eigensolver. In our implementation, we set the dimension of the Krylov subspace to 20 and found that, for small  $\sigma_R$ , this value allowed for the convergence of the three eigenvectors containing the rotation estimates. It is thus reasonable to assume that, for highly sparse problems, this task can be performed efficiently.

### 3.2. Optimizing for translations via Cholesky LDL

For a set of rotation estimates  $\widehat{\mathbf{R}}$ , the optimal translations can be computed by solving a symmetric linear system with the same pattern of zeros as the one used in the Krylov-Schur method. Let  $b \in \mathbb{R}^{3n}$  be defined as

$$b = \begin{bmatrix} \sum_{(1,j) \in E} \frac{1}{2} (\tilde{t}_{1j} + \widehat{\mathbf{R}}_1 \widehat{\mathbf{R}}_j^\top \tilde{\mathbf{R}}_{j1} \tilde{t}_{1j}) \\ \vdots \\ \sum_{(n,j) \in E} \frac{1}{2} (\tilde{t}_{nj} + \widehat{\mathbf{R}}_n \widehat{\mathbf{R}}_j^\top \tilde{\mathbf{R}}_{jn} \tilde{t}_{nj}) \end{bmatrix}. \quad (18)$$

The derivative of the log-likelihood w.r.t.  $t$  is given by

$$\frac{\partial}{\partial t} \log L(\theta|y) = ((\mathcal{L} \otimes \mathbf{J}_3) \circ \mathbf{R}\mathbf{R}^\top) t + b. \quad (19)$$

It follows then that the solution to the convex problem  $\mathcal{T}$  can be obtained by solving

$$((\mathcal{L} \otimes \mathbf{J}_3) \circ \widehat{\mathbf{R}}\widehat{\mathbf{R}}^\top) t + b = 0. \quad (20)$$

Since the term in parentheses is symmetric, we can use Cholesky LDL factorization to solve for  $t$ . The advantage of doing so comes from the fact that the Krylov-Schur method makes use of the same solver. Since the factorized matrices share the same pattern of zero entries, the symbolic decomposition performed by the solver before starting the Krylov iterations, is valid afterwards, when optimizing for translations. This results in a considerable performance gain. Our algorithm for PGO in  $\text{SO}(3)$  and  $\text{SE}(3)$  will be hereafter referred to as MAKS (Motion Averaging via Krylov-Schur).

### 3.3. Iteratively reweighted spectral decomposition

Since the computation of pairwise transformations via ICP is prone to generate outliers, we embedded MAKS in an Iteratively Reweighted Least Squares (IRLS) framework, in order to identify model-incoherent rotation measurements. This method works as follows.

A weight matrix  $\mathbf{W} \in \mathbb{R}^{n \times n}$  is initialized with ones. Iteratively: the block matrix  $\widetilde{\mathbf{R}}$  with the measurements, weighted by  $\mathbf{W}$  *i.e.*,  $(\mathbf{W} \otimes \mathbf{J}_3) \circ \widetilde{\mathbf{R}}$ , is used to compute the eigenvector solution  $\widehat{\mathbf{R}}$ , let  $\lambda_1$  be the associated smallest eigenvalue; an error matrix  $\mathbf{P} \in \mathbb{R}^{n \times n}$  is computed containing the errors of all the edges of the graph; rotation measurements with an error above a certain threshold  $\eta$  are replaced in  $\widetilde{\mathbf{R}}$  by their estimate  $\widehat{\mathbf{R}}_i \widehat{\mathbf{R}}_j^\top$ ; the weight matrix is updated as  $\mathbf{W} \leftarrow \mathbf{W} \circ \rho(\mathbf{P})$  (where  $\rho$  is an elementwise loss function); the non-diagonal entries of  $\mathbf{W}$  are row-wise normalized so that they sum up to the number of non-zero entries and the main diagonal of  $\mathbf{W}$  is set to ones; finally,  $\mathbf{W}$  is replaced by its symmetric component. These steps are repeated until  $|\lambda_1|$  falls below a predefined threshold.

We used an elementwise exponential loss function, with parameters  $a$  and  $b$ , defined as follows

$$\rho(\mathbf{P}; a, b) = a + b \exp\left(\pi - \frac{\mathbf{P}}{2}\right). \quad (21)$$

The reasoning behind this algorithm comes from the knowledge that, in the case without outliers,  $\lambda_1$  should be close to zero. By iteratively updating the weights, the transformations which remain unexplained by the eigenvector over many iterations incur increasingly larger penalties. Consequently, outlier transformations will have an increasingly smaller impact on the spectrum and eigenspaces.

## 4. Evaluation and experiments

In this section we present the results of several PGO simulations and benchmarks used to assess the performance of our algorithm. All the tests were conducted on an Intel® Core i7-4700HQ @ 3.4GHz with 16GB of RAM. MAKS was implemented in C++ using the Eigen template library for linear algebra optimized with Intel® MKL.

## 4.1. Rotation averaging

In order to assess the quality of a set of rotation estimates and quantify its error, we will use three metrics often featured in the literature: the mean (MN), the median (MD) and the root-mean-square error (RMSE). These are computed for the set of geodesic distances in  $SO(3)$  [14], between an estimate and the global optimum of  $\mathcal{R}$  or between an estimate and the corresponding ground-truth.

### 4.1.1 On the optimality of our solution

We begin the analysis of the proposed rotation averaging algorithm by an empirical study on the optimality of our solution. Our goal is to show that, assuming a small noise standard deviation  $\sigma_R$ , our solution is in the basin of attraction of the global optimum and can be accepted as optimal with a negligible angular error.

Consider three random graphs  $\mathcal{G}_i = (V, E_i)$  with  $|V| = 5750$  and a variable set of edges. The graphs have the following algebraic connectivities (Fiedler values)  $F_1 = 0.005$ ,  $F_2 = 0.015$ ,  $F_3 = 0.1$ . To each edge of the graph we assigned a ground-truth rotation corrupted by isotropic Langevin noise, with  $\sigma_R$  varying from 0.5 to 3.5 deg. We used MAKS to compute a set of rotation estimates and the Riemannian gradient descent on  $SO(3)$  [23] to converge toward the nearest stationary point. The stopping criterion was defined as the Riemannian gradient having a Frobenius norm smaller than  $10^{-8}$ . This stationary point was then used to compute the Lagrange multiplier estimate  $\hat{\Lambda}$ , according to (12). Finally, we consider this stationary point to be the global optimum (13) if  $|\lambda_{\min}(\hat{\Lambda} - \hat{\mathbf{R}})| \leq 10^{-6}$ .

The suboptimality of our solution, as measured by the RMSE in degrees, with respect to the global optimum is plotted in Fig. 2. Regardless of the graph, and for all  $\sigma_R$  we were able to converge to the optimal solution from our estimate. More importantly however, is the fact that the angular RMSE between our estimate and the global optimum is considerably small. This suboptimality error appears to decrease as the connectivity of the graph increases.

### 4.1.2 Benchmarks

In order to test our solution with real data, we used 12 real rotation averaging datasets assembled by Wilson et al. [24]. These datasets contain not only noisy observations of the relative rotations but also ground-truth information obtained through Bundle Adjustment.

Due to the large percentage of outlier transformations in the datasets considered, MAKS is expected to perform poorly. Consequently, we resort to IRLS-MAKS, parameterized with  $a = -1.75$ ,  $b = 0.85$  and  $\eta = \pi/2$ . Our iterative method was benchmarked against a novel deep learning approach proposed by Purkait et al., NeuRoRa [18], which

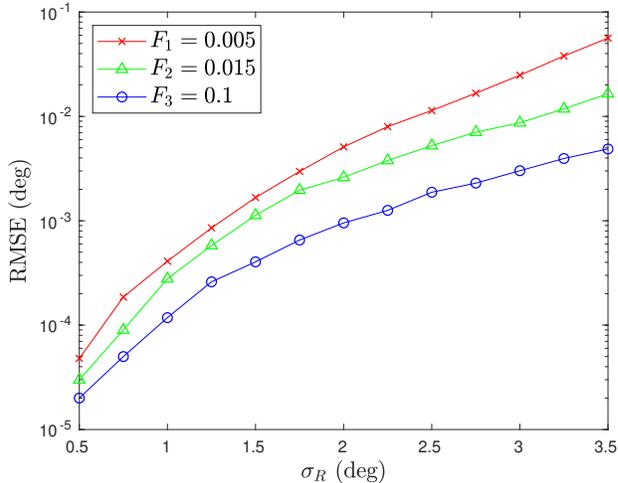


Figure 2: RMSE between our estimate and the global optimum for three graphs with  $n = 5750$ , different connectivities  $F$  and variable  $\sigma_R$ . Averaged over 10 simulations.

combines two neural networks that suppress outliers and estimate rotations. A second benchmark was the algorithm devised by Chatterjee et al. [8], 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 set forth in [18]. Finally, we also compare our results to IRLS EIG-SE(3) by Arrigoni et al. [2], for which the authors' MATLAB code can be found online.

The ground-truth error was computed for three metrics defined in Section 4.1. The results are presented in Table 1. Our solution outperforms that of Chatterjee et al. [8] 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, IRLS-MAKS fares better overall, especially in terms of the MD error. By comparing our errors with those of the IRLS method by Arrigoni et al. [2], 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 IRLS-MAKS and IRLS EIG-SE(3) are implemented in MATLAB.

## 4.2. PGO in SE(3)

To assess the performance of MAKS in 3D-SLAM we optimized six datasets by Carlone et al. [6]. Three of them were simulated (Sphere, Torus3D, Grid3D) and the rest were built from visual data (Garage, Cubicle, Rim). Since ground-truth information is unavailable, we will rely on the likelihood function to compare the different approaches.

Dataset	Graph		IRLS-MAKS (ours)			Chatterjee et al. [8]			Purkait et al. [18]			Arrigoni et al. [2]		
	$ V $	$ E $	MN	MD	RMSE	MN	MD	RMSE	MN	MD	RMSE	MN	MD	RMSE
Alamo	627	97296	<b>2.3</b>	<b>0.6</b>	<b>6.9</b>	4.2	1.1	12.7	4.9	1.2	16.1	3.9	1.3	12.1
Ellis Island	247	20297	<b>1.4</b>	<b>0.3</b>	<b>5.4</b>	2.8	0.5	10.4	2.6	0.6	12.8	3.1	0.8	10.5
Yorkminster	458	27729	<b>2.0</b>	<b>0.9</b>	<b>5.1</b>	3.5	1.6	8.4	2.5	<b>0.9</b>	6.6	3.8	1.8	9.4
Montreal Notre Dame	474	52424	<b>1.1</b>	<b>0.3</b>	6.3	1.5	0.5	7.5	1.2	0.6	<b>2.7</b>	1.9	0.6	11.2
Vienna Cathedral	918	103550	<b>3.4</b>	<b>0.9</b>	10.2	8.2	1.2	27.8	3.9	1.5	<b>9.9</b>	8.6	1.6	28.6
Piazza del Popolo	354	24710	3.1	<b>0.5</b>	<b>6.4</b>	4.0	0.8	8.4	<b>3.0</b>	0.7	9.0	3.9	1.0	9.5
Union Square	930	25561	<b>4.3</b>	3.3	<b>7.7</b>	9.3	3.9	22.4	5.9	<b>2.0</b>	17.6	6.9	5.4	13.1
NY Library	376	20680	<b>1.9</b>	<b>0.8</b>	3.9	3.0	1.3	6.9	<b>1.9</b>	1.1	<b>2.9</b>	3.7	2.1	7.8
Notre Dame	553	103932	2.1	<b>0.5</b>	8.0	3.5	0.6	14.6	<b>1.6</b>	0.6	<b>6.4</b>	3.9	1.2	14.9
Roman Forum	1134	70187	3.0	2.6	<b>5.4</b>	3.1	1.5	10.2	<b>2.3</b>	<b>1.3</b>	5.5	26.1	4.6	44.0
Tower of London	508	23863	2.7	1.7	<b>5.7</b>	3.9	2.4	9.1	<b>2.6</b>	<b>1.4</b>	5.8	4.5	2.6	10.6
Madrid Metropolis	394	23784	4.7	<b>1.0</b>	11.5	6.9	1.2	17.3	<b>2.5</b>	1.1	<b>6.6</b>	9.8	4.4	18.7

Table 1: Graph characteristics and comparison between IRLS-MAKS and other rotation averaging algorithms.

For each dataset we computed the global pose estimates using MAKS and EIG-SE(3) [2]. For the latter we used the authors' MATLAB implementation. Due to the prominence of the chordal relaxation method [17] in the literature as an initialization for iterative solvers, we benchmarked this technique as well, using a C++ implementation. Since neither of these methods is optimal, we resort to Gauss-Newton (g2o) [15] initialized from our solution to obtain the global maximum of the likelihood function (verified with SE-Sync [19]). 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 2. As an example, in Fig. 3 we show the camera trajectory resulting from our optimization of Garage and Cubicle.

Since Garage and Cubicle are the datasets with the smallest eigenvalues (in absolute value), we posit that these pose graphs were accurately optimized using MAKS. The experiments confirm this, since 10 Gauss-Newton iterations did not increase the value of the log-likelihood by a significant amount. This is also the case for Grid3D and Torus3D, in spite of their larger eigenvalues. For these four datasets, our solution is a good approximation of global optimum.

The only datasets for which there is a considerable difference between the global optimum and our estimate are Sphere and Rim. This disparity is common to all the suboptimal methods considered, ours attaining the highest objective among them. The Sphere case is particularly remarkable. We hypothesize that the lackluster performance of MAKS is due to the configuration of the pose graph itself. The fact that it simulates the poses of a robot traveling on a spherical surface, combined with an adjacency matrix made up of  $k$ -diagonals, results in nearly constant relative rotations. Since its smallest eigenvalue is considerably larger than those of the other datasets, we can place Sphere outside the applicability domain of our method.

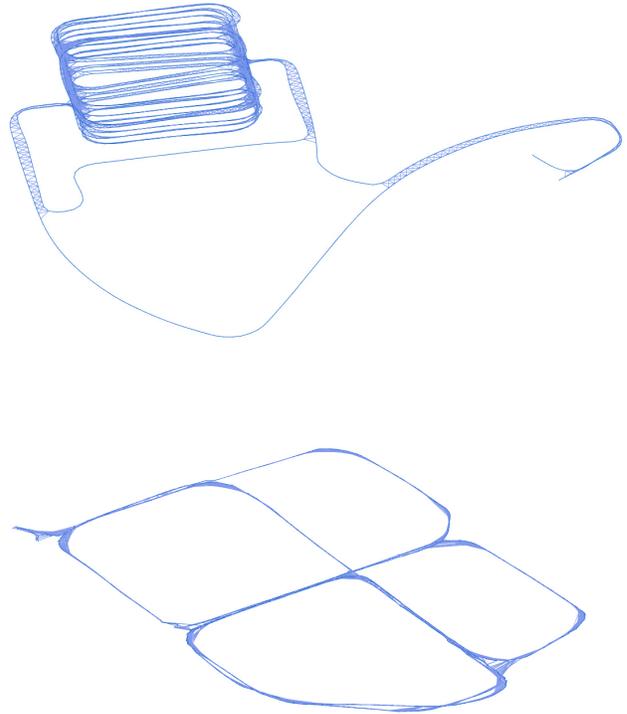


Figure 3: Trajectories estimated by MAKS and loop closures. Top: Garage (n=1661). Bottom: Cubicle (n=5750).

When comparing MAKS to EIG-SE(3) one notices that, except for the Garage dataset, the latter produces poorer results. The discrepancy between these otherwise similar methods stems from the fact that EIG-SE(3) performs a single eigendecomposition to obtain four eigenvectors which

Dataset	Graph		MAKS (ours)			Chordal [17]		EIG-SE3 [2]		Gauss-Newton [15]	
	$ V $	$ E $	$\lambda_1$	$\log L$	$t_{\text{CPU}}(s)$	$\log L$	$t_{\text{CPU}}(s)$	$\log L$	$t_{\text{CPU}}(s)$	$\log L^*$	$t_{\text{CPU}}(s)$
Parking garage	1661	6275	4.2e-7	<b>1.88e4</b>	<b>0.03</b>	<b>1.88e4</b>	0.28	1.41e4	1.78	<b>1.88e4</b>	0.22
Torus3D	5000	9048	3.9e-3	<b>2.71e4</b>	<b>0.20</b>	<b>2.71e4</b>	1.19	2.69e4	4.04	<b>2.71e4</b>	7.01
Grid3D	8000	22236	8.7e-3	<b>6.64e4</b>	<b>0.83</b>	<b>6.64e4</b>	5.35	6.58e4	2.11	<b>6.64e4</b>	492.17
Cubicle	5750	16869	9.0e-6	<b>3.77e4</b>	<b>0.23</b>	3.75e4	1.53	3.74e4	2.12	<b>3.77e4</b>	3.65
Rim	10195	29743	1.7e-5	6.67e4	<b>0.46</b>	6.67e4	2.71	6.64e4	7.81	<b>6.69e4</b>	10.69
Sphere	2200	8647	2.2e-1	1.66e4	<b>0.16</b>	1.33e4	0.78	-5.17e5	0.43	<b>2.06e4</b>	3.04

Table 2: Graph characteristics and comparison between MAKS, Chordal relaxation, EIG-SE(3) and Gauss-Newton (g2o).

are then projected to  $SE(3)^n$ . Consequently, even if the rotation estimates should, in theory, be the same, its translation estimates are, in general, not optimal when considering the rotation estimates computed. The chordal method is the only suboptimal relaxation that produces results comparable to ours. However, we attain higher objectives in Cubicle, Sphere and Rim. Furthermore, the computation of the eigenvalues in MAKS allows for an optimality assessment. A similar procedure in the Chordal method would translate to a higher CPU time.

In terms of CPU time, our compiled C++ implementation of MAKS outperforms the other methods. We fare better than Gauss-Newton (g2o) in all the datasets, 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 Gauss-Newton, but it is nearly 600 times faster. Furthermore, the CPU time we have indicated for this solver does not take into account the initialization, which dictates how well it can perform. The Chordal relaxation, implemented in C++, also lags behind MAKS by a considerable amount. A conclusive statement on the CPU time of EIG-SE(3) cannot be made at this stage since this method was originally implemented in MATLAB.

### 4.3. Dense 3D reconstruction

We now justify the applicability of our PGO solution to the problem of registering multiple point clouds obtained from RGB-D cameras. We implemented a RGB-D registration pipeline that estimates pairwise rigid transformations via ICP, initialized with RANSAC-filtered 2D image matches computed with SIFT [16]. This set of pairwise transformation estimates is used to build a pose graph which we optimize with MAKS, in order to obtain the camera trajectory and subsequently reconstruct the 3D scene.

We tested our pipeline using four RGB-D datasets: Burghers, Stonewall, Sports car and Lounge from a collection by Zhou et al. [25]. Due to the large overlap between adjacent frames in the first two datasets, we registered them with a stride of 10 images. The last two were registered in their entirety. The four 3D scenes were fully reconstructed and are shown in Figs. 4 5, 6 and 7.

In all the datasets reconstructed, the smallest rotation averaging eigenvalue, which we use to assess optimality according to (13), had a magnitude no greater than  $10^{-5}$ . This is in accordance with our claim that, when considering RGB-D registration, pairwise transformations can be estimated with a high SNR by means of ICP algorithms. We can therefore expect our PGO solution to be, in these cases, a good approximation of the global optimum.

Regarding CPU time, MAKS took 0.02 seconds for the smallest dataset (Stonewall with 271 poses) and 0.35 seconds for the largest (Sports car with 6523 poses). With such CPU times, our algorithm lends itself to real-time PGO applications, namely 3D-SLAM and online reconstruction.

## 5. Conclusions

On the one hand, MLE is capable of modeling PGO with a high degree of accuracy. Nonetheless, the optimization strategies involved are often too cumbersome and require good initializations in order to attain the global optimum. On the other hand, light-weight relaxations often fail to achieve the same degree of precision as MLE. Our solution to the otherwise difficult problem of PGO, when applicable, outperforms the state-of-the-art in efficiency without compromising precision.

The empirical results we presented validate the claims we made throughout this paper and allow us to assert the following. Under isotropic Langevin noise, MAKS produces a good approximation of the rotation averaging global optimum. Considering outliers, our IRLS framework proved to be more accurate than the state-of-the-art, when tested in Bundle Adjustment datasets. For PGO in  $SE(3)$ , our approach, combining the Krylov-Schur method for spectral decomposition with Cholesky LDL factorization, is faster than the state-of-the-art and optimal under high SNR. This is usually the norm in computer vision applications, namely RGB-D registration, as evidenced by the 3D scene reconstructions produced by our pipeline.

**Acknowledgments.** This work was funded by Fundação para a Ciência e Tecnologia, grant [UIDB/50009/2020]. J. Costeira was also supported by the European Union’s Horizon 2020 project (GA 825619, AI4EU).

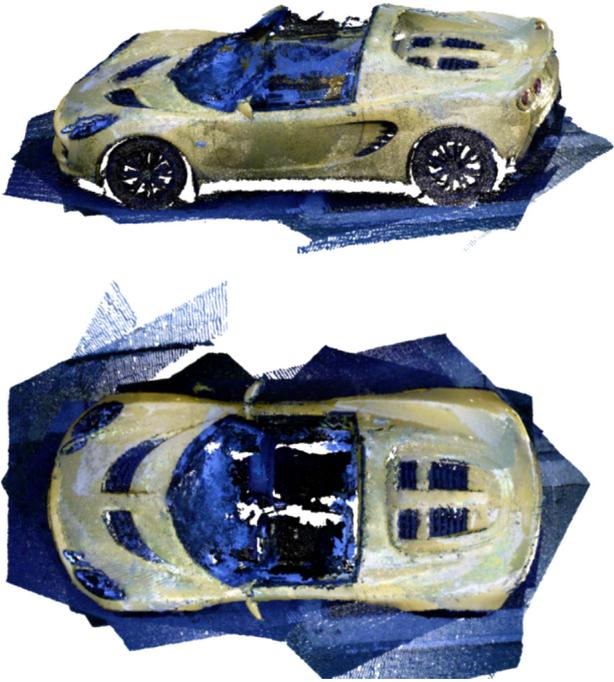


Figure 4: 3D reconstruction of the Sports car dataset [25] with 6523 poses and 22207 edges. CPU time: 350ms.



Figure 5: 3D reconstruction of the Stonewall dataset [25] with 271 poses and 888 edges. CPU time: 20ms.



Figure 6: 3D reconstruction of the Burghers dataset [25] with 1124 poses and 4118 edges. CPU time: 40ms.



Figure 7: 3D reconstruction of the Lounge dataset [25] with 3000 poses and 15102 edges. CPU time: 150ms.

## References

- [1] Federica Arrigoni, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. Robust synchronization in  $SO(3)$  and  $SE(3)$  via low-rank and sparse matrix decomposition. *Computer Vision and Image Understanding*, 174(3):95–113, 2018.
- [2] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Spectral Synchronization of Multiple Views in  $SE(3)$ . *SIAM Journal on Imaging Sciences*, 9(4):1963–1990, 2016.
- [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] Nicolas Boumal, Amit Singer, P. A. Absil, and Vincent D. Blondel. Cramér-Rao bounds for synchronization of rotations. *Information and Inference*, 3(1):1–39, 2014.
- [5] Aeron M Buchanan and Andrew W Fitzgibbon. Damped Newton algorithms for matrix factorization with missing data. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 316–322, 2005.
- [6] Luca Carlone, David M. Rosen, Giuseppe Calafiore, John J. Leonard, and Frank Dellaert. Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 125–132, 2015.
- [7] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert. Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4597–4604, 2015.
- [8] Avishek Chatterjee and Venu Madhav Govindu. Robust Relative Rotation Averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):958–972, 2018.
- [9] F. Crosilla and A. Beinat. Use of generalised Procrustes analysis for the photogrammetric block adjustment by independent models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 56(3):195–209, 2002.
- [10] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.
- [11] Frank Dellaert, David M Rosen, Jing Wu, Robert Mahony, and Luca Carlone. Shonan rotation averaging: Global optimality by surfing  $SO(p)^n$ . In *European Conference on Computer Vision (ECCV)*, 2020.
- [12] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat Jun Chin. Rotation Averaging and Strong Duality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–135, 2018.
- [13] Rui F.C. Guerreiro and Pedro M.Q. Aguiar. Factorization with missing data for 3D structure recovery. In *IEEE Workshop on Multimedia Signal Processing*, pages 105–108, 2002.
- [14] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, 2013.
- [15] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.
- [16] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [17] Daniel Martinec and Tomáš Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.
- [18] Pulak Purkait, Tat-Jun Chin, and Ian Reid. NeuRoRA: Neural robust rotation averaging. In *European Conference on Computer Vision (ECCV)*, 2020.
- [19] David M. Rosen, Luca Carlone, Afonso S. Bandeira, and John J. Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *International Journal of Robotics Research*, 38(2-3):95–125, 2019.
- [20] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, pages 145–152, 2001.
- [21] Gilbert W Stewart. A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2002.
- [22] Roberto Toldo, Alberto Beinat, and Fabio Crosilla. Global registration of multiple point clouds embedding the generalized procrustes analysis into an ICP framework. In *3DPVT 2010 Conference*, 2010.
- [23] Roberto Tron and René Vidal. Distributed image-based 3-D localization of camera sensor networks. In *Proceedings of the IEEE Conference on Decision and Control*, pages 901–908, 2009.
- [24] Kyle Wilson and Noah Snavely. Robust Global Translations with 1DSfM. In *European Conference on Computer Vision (ECCV)*, 2014.
- [25] Qian-Yi Zhou and Vladlen Koltun. Dense Scene Reconstruction with Points of Interest. *ACM Transactions on Graphics*, 32(4):1–8, 2013.