Shonan Rotation Averaging: Global Optimality by Surfing $SO(p)^n$

Frank Dellaert*[1][0000−0002−5532−3566], David M. Rosen*[2], Jing Wu[1], Robert Mahony[3][0000−0002−7803−2868], and Luca Carlone[2]

[1] Georgia Institute of Technology, Atlanta, GA {fd27,jingwu}@gatech.edu
[2] Massachusetts Inst. of Technology, Cambridge, MA {dmrosen@,lcarlone}@mit.edu
[3] Australian National University, Canberra, Australia Robert.Mahony@anu.edu.au

Abstract. Shonan Rotation Averaging is a fast, simple, and elegant rotation averaging algorithm that is guaranteed to recover globally optimal solutions under mild assumptions on the measurement noise. Our method employs semidefinite relaxation in order to recover provably globally optimal solutions of the rotation averaging problem. In contrast to prior work, we show how to solve large-scale instances of these relaxations using manifold minimization on (only slightly) higher-dimensional rotation manifolds, re-using existing high-performance (but *local*) structure-from-motion pipelines. Our method thus preserves the speed and scalability of current SFM methods, while recovering *globally* optimal solutions.

1 Introduction

Rotation averaging is the problem of estimating a set of n unknown orientations $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n \in \mathrm{SO}(d)$ from noisy measurements $\bar{\boldsymbol{R}}_{ij} \in \mathrm{SO}(d)$ of the *relative rotations* $\boldsymbol{R}_i^{-1}\boldsymbol{R}_j$ between them [18,17]. This problem frequently arises in geometric reconstruction; in particular, it occurs as a sub-problem in bundle adjustment [34,3], structure from motion [29], multi-camera rig calibration [25], and sensor network localization [35]. The development of *fast* and *reliable* algorithms for solving the rotation averaging problem is therefore of great practical interest.

While there are numerous (inequivalent) ways of formalizing the rotation averaging problem in common use [18], unfortunately all of them share the common features of (a) *high dimensionality*, due to the typically large number n of orientations $\boldsymbol{R}_i$ to be estimated, and (b) *non-convexity*, due to the non-convexity of the space of rotations itself. In consequence, *all* of these approaches lead to optimization problems that are computationally hard to solve in general.

In this work, we address rotation averaging using *maximum likelihood estimation*, as this provides strong statistical guarantees on the quality of the resulting estimate [11,19]. We consider the maximum likelihood estimation problem:

$$\max_{\boldsymbol{R} \in \mathrm{SO}(d)^n} \sum_{(i,j) \in \mathcal{E}} \kappa_{ij} \operatorname{tr}\left(\boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij} \boldsymbol{R}_j^{\mathsf{T}}\right), \tag{1}$$

* Equal contribution

where the $\kappa_{ij} \geq 0$ are concentration parameters for an assumed Langevin noise model [5,9,27]. Our goal in this paper is to develop a *fast* and *scalable* optimization method that is capable of computing *globally* optimal solutions of the rotation averaging problem (1) in practice, despite its non-convexity.

We propose a new, straightforward algorithm, *Shonan Rotation Averaging*, for finding *globally optimal* solutions of problem (1). At its core, our approach simply applies the standard Gauss-Newton or Levenberg-Marquardt methods to a *sequence* of successively *higher-dimensional* rotation averaging problems

$$\max_{\boldsymbol{Q} \in \mathrm{SO}(p)^n} \sum_{(i,j) \in \mathcal{E}} \kappa_{ij} \mathrm{tr}\left(\boldsymbol{Q}_i \boldsymbol{P} \bar{\boldsymbol{R}}_{ij} \boldsymbol{P}^\top \boldsymbol{Q}_j^\top\right), \tag{2}$$

for increasing $p \geq d$. Note that the only difference between (1) and (2) is the $p \times d$ projection matrix $\boldsymbol{P} \triangleq [\mathbf{I}_d; 0]$, which adapts the measurement matrix $\bar{\boldsymbol{R}}_{ij}$ to the higher-dimensional rotations $\boldsymbol{Q}_i$. We start by running local optimization on (2) with $p = d$, and if this fails to produce a globally-optimal solution, we increase the dimension p and try again. Under mild conditions on the noise, we prove that this simple approach succeeds in recovering a *globally* optimal solution of the rotation averaging problem (1).

A primary contribution of this work is to show how the fast optimization approach developed in [27,26] can be adapted to run directly on the manifold of rotations (rather than the Stiefel manifold), implemented using the venerable Gauss-Newton or Levenberg-Marquardt methods. This approach enables Shonan Averaging to be easily implemented in high-performance optimization libraries commonly used in robotics and computer vision [2,22,12].

2 Related Work

By far the most common approach to addressing smooth optimization problems in computer vision is to apply standard first- or second-order nonlinear optimization methods to compute a critical point of the objective function [24]; this holds in particular for the rotation averaging problem (see [18] generally). This approach is very attractive from the standpoint of computational efficiency, as the low per-iteration cost of these techniques (exploiting the sparsity present in real-world problems) enables these methods to scale gracefully to very large problem sizes; indeed, it is now possible to process reconstruction problems (of which rotation averaging is a crucial part) involving *millions* of images on a single machine [15,20]. However, this computational efficiency comes at the expense of *reliability*, as the use of *local* optimization methods renders this approach susceptible to convergence to bad (significantly suboptimal) local minima.

To address this potential pitfall, several recent lines of work have studied the convergence behavior of local search techniques applied to the rotation averaging problem. One thrust proposes various initialization techniques that attempt to start the local search in low-cost regions of the state space, thereby favoring convergence to the true (global) minimum [23,10,6,30]. Another direction investigates the size of the locally convex region around the global minimizer, in order to understand when local search is likely to succeed [18,38]. A third class of approaches attempts to evaluate the *absolute* quality of a candidate solution $\hat{\boldsymbol{R}}$ by employing Lagrangian duality to derive an *upper bound* on $\hat{\boldsymbol{R}}$'s (global) suboptimality [16,9]. Interestingly, while these last two works employ different representations for rotations ([16] uses quaternions, whereas [9] uses rotation matrices), *both* of the resulting dual problems are semidefinite programs [37], and *both* of these duals are observed to be *tight* unless the measurements $\bar{\boldsymbol{R}}_{ij}$ are contaminated by large amounts of noise; this fact enables the *certification* of optimality of a global minimizer $\boldsymbol{R}^*$.

Motivated by the striking results reported in [16,9], a recent line of work proposes to recover *globally* optimal solutions of the rotation averaging problem from a solution of the Lagrangian dual. Both [16] and [9] propose to compute such solutions using an off-the-shelf SDP solver; however, as general-purpose SDP methods do not scale well with the problem size, this approach is limited to problems involving only a few hundred states. More recently, [13,14] proposed a block-coordinate-descent method specifically tailored to the dual of (1), and showed that this method was 1-2 orders of magnitude faster than the standard SDP algorithm SeDuMi [32]; however, the reported results were still limited to problems involving at most ≈ 300 states. Finally, [27] presents a global optimization approach for pose-graph SLAM based upon the *dual* of the Lagrangian dual (also an SDP), together with a fast optimization scheme that is capable of solving problems involving tens of thousands of poses in a few seconds. However, this optimization approach uses a truncated-Newton Riemannian optimization method [1] employing an *exact* model Hessian, and so cannot be deployed using the Gauss-Newton framework [24] that forms the basis of standard optimization libraries commonly used in computer vision applications [2,22,12].

Our method builds on the approach of [27], but adapts the optimization scheme to run directly on the manifold of rotations using the Gauss-Newton method or a trust-region variant like Levenberg-Marqardt. In this way, it is able to leverage the availability of high-performance software libraries [2,22,12] for performing *local* search on problems of the form (2) while preserving *global optimality* guarantees. In addition, working with the rotation manifold $\mathrm{SO}(p)$, for $p \geq 3$ avoids introducing new, unfamiliar objects like Stiefel manifolds in the core algorithm. The result is a simple, intuitive method for globally optimal rotation averaging that improves upon the scalability of current global methods [13] by an order of magnitude.

3 Gauss-Newton for Rotation Averaging

This section reviews how the Gauss-Newton (GN) algorithm is applied to find a first-order critical point of the rotation averaging problem (1). We first rewrite (1) in terms of minimizing a sum of Frobenius norms:

$$\min_{\boldsymbol{R} \in \mathrm{SO}(d)^n} \sum_{(i,j) \in \mathcal{E}} \kappa_{ij} \left\| \boldsymbol{R}_j - \boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij} \right\|_{\mathrm{F}}^2 . \tag{3}$$

This can in turn be vectorized as:

$$\min_{\boldsymbol{R}\in \mathrm{SO}(d)^n} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij} \left\| \mathrm{vec}(\boldsymbol{R}_j) - \mathrm{vec}(\boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij}) \right\|_2^2 \tag{4}$$

where "vec" is the column-wise vectorization of a matrix, and we made use of the fact that $\|\boldsymbol{M}\|_{\mathrm{F}}^2 = \|\mathrm{vec}(\boldsymbol{M})\|_2^2$.

Problem (4) does not admit a simple closed-form solution. Therefore, given a feasible point $\boldsymbol{R} = (\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n) \in \mathrm{SO}(d)^n$, we will content ourselves with identifying a set of tangent vectors $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_n \in \mathfrak{so}(d)$ along which we can *locally perturb* each rotation $\boldsymbol{R}_i$:

$$\boldsymbol{R}_i \leftarrow \boldsymbol{R}_i e^{[\boldsymbol{\omega}_i]} \tag{5}$$

to *decrease* the value of the objective; here $[\boldsymbol{\omega}_i]$ is the $d \times d$ skew-symmetric matrix corresponding to the hat operator of the Lie algebra $\mathfrak{so}(d)$ associated with the rotation group $\mathrm{SO}(d)$. We can therefore reformulate problem (4) as:

$$\min_{\boldsymbol{\omega}\in \mathfrak{so}(d)^n} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij} \left\| \mathrm{vec}(\boldsymbol{R}_j e^{[\boldsymbol{\omega}_j]}) - \mathrm{vec}(\boldsymbol{R}_i e^{[\boldsymbol{\omega}_i]} \bar{\boldsymbol{R}}_{ij}) \right\|_2^2. \tag{6}$$

In effect, equation (6) replaces the optimization over the *rotations* $\boldsymbol{R}_i$ in (4) by an optimization over the *tangent vectors* $\boldsymbol{\omega}_i \in \mathfrak{so}(d)$. This is advantageous because $\mathfrak{so}(d)$ is a *linear* space, whereas $\mathrm{SO}(d)$ is not. However, we still cannot solve (6) directly, as the $\boldsymbol{\omega}_i$ enter the objective through the (nonlinear) exponential map.

However, we can *locally approximate* the exponential map to first order as $e^{[\boldsymbol{\nu}]} \approx \mathbf{I} + [\boldsymbol{\nu}]$. Therefore, for any matrix $\boldsymbol{A}$ we have

$$\begin{aligned}
\mathrm{vec}(\bar{\boldsymbol{R}} e^{[\boldsymbol{\nu}]} \boldsymbol{A}) &\approx \mathrm{vec}(\bar{\boldsymbol{R}}(\mathbf{I} + [\boldsymbol{\nu}])\boldsymbol{A}) && (7)\\
&= \mathrm{vec}(\bar{\boldsymbol{R}}\boldsymbol{A}) + \mathrm{vec}(\bar{\boldsymbol{R}}[\boldsymbol{\nu}]\boldsymbol{A}) && (8)\\
&= \mathrm{vec}(\bar{\boldsymbol{R}}\boldsymbol{A}) + (\boldsymbol{A}^\top \otimes \bar{\boldsymbol{R}})\mathrm{vec}([\boldsymbol{\nu}]), && (9)
\end{aligned}$$

where we made use of a well-known property of the Kronecker product $\otimes$. We can also decompose the skew-symmetric matrix $[\boldsymbol{\nu}]$ in terms of the coordinates ν^k of the vector $\boldsymbol{\nu}$ according to:

$$\mathrm{vec}([\boldsymbol{\nu}])) = \mathrm{vec}(\sum \boldsymbol{\nu}^k \boldsymbol{G}_k) = \sum \boldsymbol{\nu}^k \mathrm{vec}(\boldsymbol{G}_k) = \bar{\boldsymbol{G}}_d \boldsymbol{\nu} \tag{10}$$

where $\boldsymbol{G}_k$ is the kth generator of the Lie algebra $\mathfrak{so}(d)$, and $\bar{\boldsymbol{G}}_d$ is the matrix obtained by concatenating the vectorized generators $\mathrm{vec}(\boldsymbol{G}_k)$ column-wise.

The (local) Gauss-Newton model of the rotation averaging problem (1) is obtained by substituting the linearizations (7)–(9) and decomposition (10) into (6) to obtain a *linear least-squares* problem in the tangent vectors $\boldsymbol{\omega}_i$:

$$\min_{\boldsymbol{\omega}\in \mathfrak{so}(d)^n} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij} \left\| \boldsymbol{F}_j \boldsymbol{\omega}_j - \boldsymbol{H}_i \boldsymbol{\omega}_i - \boldsymbol{b}_{ij} \right\|_2^2, \tag{11}$$

where the Jacobians $\boldsymbol{F}_j$ and $\boldsymbol{H}_i$ and the right-hand side $\boldsymbol{b}_{ij}$ can be calculated as

$$\boldsymbol{F}_j \doteq (\mathbf{I} \otimes \bar{\boldsymbol{R}}_j)\bar{\boldsymbol{G}}_d \tag{12}$$

$$\boldsymbol{H}_i \doteq (\bar{\boldsymbol{R}}_{ij}^{\mathsf{T}} \otimes \bar{\boldsymbol{R}}_i)\bar{\boldsymbol{G}}_d \tag{13}$$

$$\boldsymbol{b}_{ij} \doteq \mathrm{vec}(\bar{\boldsymbol{R}}_i \bar{\boldsymbol{R}}_{ij} - \bar{\boldsymbol{R}}_j). \tag{14}$$

The *local model* problem (11) can be solved efficiently to produce an optimal *correction* $\boldsymbol{\omega}^\star \in \mathfrak{so}(d)^n$. (To do so one can use either direct methods, based on sparse matrix factorization, or preconditioned conjugate gradient (PCG). As an example, to produce the results in Section 5 we use PCG with a block-Jacobi preconditioner and a Levenberg-Marquardt trust-region method.) This correction is then applied to *update* the state $\boldsymbol{R}$ as in equation (5). Typically, this is done in conjunction with a trust-region control strategy to prevent taking a step that leaves the neighborhood of $\boldsymbol{R}$ in which the local linear models (11)–(14) well-approximate the objective. The above process is then repeated to generate a *sequence* $\{\boldsymbol{\omega}\}$ of such corrections, each of which improves the objective value, until some termination criterion is satisfied.

We emphasize that while the Gauss-Newton approach is sufficient to *locally improve* on an initial estimate, but because of non-convexity the final iterate returned by this method is ***not*** guaranteed to be a minimizer of (1).

4 Shonan Rotation Averaging

4.1 A convex relaxation for rotation averaging

The main idea behind Shonan Averaging is to develop a *convex relaxation* of (1) (which can be solved *globally*), and then exploit this relaxation to search for good solutions of the rotation averaging problem. Following [27, Sec. 3], in this section we derive a convex relaxation of (1) whose minimizers in fact provide *exact*, *globally optimal* solutions of the rotation averaging problem subject to mild conditions on the measurement noise.

To begin, we rewrite problem (1) in a more compact, matricized form as:

$$f^*_{\mathrm{MLE}} = \min_{\boldsymbol{R} \in \mathrm{SO}(d)^n} \mathrm{tr}\left(\bar{\boldsymbol{L}} \boldsymbol{R}^{\mathsf{T}} \boldsymbol{R}\right), \tag{15}$$

where $\boldsymbol{R} = (\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n)$ is the $d \times dn$ matrix of rotations $\boldsymbol{R}_i \in \mathrm{SO}(d)$, and $\bar{\boldsymbol{L}}$ is a symmetric $(d \times d)$-block-structured matrix constructed from the measurements $\bar{\boldsymbol{R}}_{ij}$. The matrix $\bar{\boldsymbol{L}}$, known as the *connection Laplacian* [31], is the generalization of the standard (scalar) graph Laplacian to a graph having matrix-valued data $\bar{\boldsymbol{R}}_{ij}$ assigned to its edges.

Note that $\boldsymbol{R}$ enters the objective in (15) only through the product $\boldsymbol{R}^{\mathsf{T}}\boldsymbol{R}$; this is a rank-$d$ positive-semidefinite matrix (since it is an outer product of the rank-d matrix $\boldsymbol{R}$), and has a $(d \times d)$-block-diagonal comprised entirely of identity matrices (since the blocks $\boldsymbol{R}_i$ of $\boldsymbol{R}$ are rotations). Our convex relaxation of (15) is derived simply by replacing the rank-d product $\boldsymbol{R}^\mathsf{T}\boldsymbol{R}$ with a *generic* positive-semidefinite matrix $\boldsymbol{Z}$ having identity matrices along its $(d \times d)$-block-diagonal:

$$\begin{aligned} f^*_{\mathrm{SDP}} &= \min_{\boldsymbol{Z}\succeq 0} \operatorname{tr}\left(\bar{\boldsymbol{L}}\boldsymbol{Z}\right) \\ \text{s.t. } & \operatorname{BlockDiag}_{d\times d}(\boldsymbol{Z}) = (\mathbf{I}_d, \ldots, \mathbf{I}_d). \end{aligned} \tag{16}$$

Problem (16) is a *semidefinite program* (SDP) [37]: it requires the minimization of a linear function of a positive-semidefinite matrix $\boldsymbol{Z}$, subject to a set of linear constraints. Crucially, since the set of positive-semidefinite matrices is a convex cone, SDPs are *always* convex, and can therefore be solved *globally* in practice. Moreover, since by construction (15) and (16) share the same objective, and the feasible set of (16) contains every matrix of the form $\boldsymbol{R}^\mathsf{T}\boldsymbol{R}$ with $\boldsymbol{R}$ feasible in (15), we can regard (16) as a convexification of (15) obtained by *expanding the latter's feasible set.* It follows immediately that $f^*_{\mathrm{SDP}} \leq f^*_{\mathrm{MLE}}$. Furthermore, if it so happens that a minimizer $\boldsymbol{Z}^\star$ of (16) admits a factorization of the form $\boldsymbol{Z}^\star = \boldsymbol{R}^{\star\mathsf{T}}\boldsymbol{R}^\star$ with $\boldsymbol{R}^\star \in \mathrm{SO}(d)^n$, then it is clear that $\boldsymbol{R}^\star$ is also a (global) minimizer of (15), since it attains the lower-bound f^*_{SDP} for (15)'s optimal value f^*_{MLE}. The key fact that motivates our interest in the relaxation (16) is that this favorable situation *actually occurs*, provided that the noise on the observations $\bar{\boldsymbol{R}}_{ij}$ is not too large. More precisely, the following result is a specialization of [27, Proposition 1] to the rotation averaging problem (15).

Theorem 1. *Let $\boldsymbol{L}$ denote the connection Laplacian for problem (15) constructed from the true (noiseless) relative rotations $\boldsymbol{R}_{ij} \triangleq \boldsymbol{R}_i^{-1}\boldsymbol{R}_j$. Then there exists a constant $\beta \triangleq \beta(\boldsymbol{L})$ (depending upon $\boldsymbol{L}$) such that, if $\|\bar{\boldsymbol{L}} - \boldsymbol{L}\|_2 \leq \beta$:*

(i) The semidefinite relaxation (16) has a unique solution $\boldsymbol{Z}^\star$, and
(ii) $\boldsymbol{Z}^\star = \boldsymbol{R}^{\star\mathsf{T}}\boldsymbol{R}^\star$, where $\boldsymbol{R}^\star \in \mathrm{SO}(d)^n$ is a globally optimal *solution of the rotation averaging problem (15).*

4.2 Solving the semidefinite relaxation: The Riemannian Staircase

In this section, we describe a specialized optimization procedure that enables the fast solution of large-scale instances of the semidefinite relaxation (16), following the approach of [27, Sec. 4.1].

Theorem 1 guarantees that in the (typical) case that (16) is exact, the *solution* $\boldsymbol{Z}^\star$ that we seek admits a concise description in the factored form $\boldsymbol{Z}^\star = \boldsymbol{R}^{\star\mathsf{T}}\boldsymbol{R}^\star$ with $\boldsymbol{R}^\star \in \mathrm{SO}(d)^n$. More generally, even in those cases where exactness fails, minimizers $\boldsymbol{Z}^\star$ of (16) generically have a rank p not much greater than d, and therefore admit a symmetric rank decomposition $\boldsymbol{Z}^\star = \boldsymbol{S}^{\star\mathsf{T}}\boldsymbol{S}^\star$ for $\boldsymbol{S}^\star \in \mathbb{R}^{p\times dn}$ with $p \ll dn$. We exploit the existence of such low-rank solutions by adopting the approach of [8], and replacing the decision variable $\boldsymbol{Z}$ in (16) with a symmetric rank-p product of the form $\boldsymbol{S}^\mathsf{T}\boldsymbol{S}$. This substitution has the effect of dramatically reducing the size of the search space, as well as rendering the positive-semidefiniteness constraint *redundant*, since $\boldsymbol{S}^\mathsf{T}\boldsymbol{S} \succeq 0$ for *any* $\boldsymbol{S}$. The resulting *rank-restricted* version of (16) is thus a standard *nonlinear program* with decision variable the low-rank factor $\boldsymbol{S}$:

$$\begin{aligned} f^*_{\mathrm{SDPLR}}(p) = & \min_{\boldsymbol{S} \in \mathbb{R}^{p \times dn}} \operatorname{tr}\left(\bar{\boldsymbol{L}} \boldsymbol{S}^\mathsf{T} \boldsymbol{S}\right) \\ \text{s.t. } & \operatorname{BlockDiag}_{d \times d}(\boldsymbol{S}^\mathsf{T} \boldsymbol{S}) = (\mathbf{I}_d, \ldots, \mathbf{I}_d). \end{aligned} \tag{17}$$

The identity block constraints in (17) are equivalent to the $p \times d$ block-columns $\boldsymbol{S}_i$ of $\boldsymbol{S}$ being orthonormal d-frames in $\mathbb{R}^p$. The set of all orthonormal d-frames in $\mathbb{R}^p$ is a matrix manifold, the **Stiefel manifold** $\mathrm{St}(d, p)$ [1]:

$$\mathrm{St}(d, p) \doteq \{\boldsymbol{M} \in \mathbb{R}^{p \times d} \mid \boldsymbol{M}^\mathsf{T} \boldsymbol{M} = \mathbf{I}_d\}. \tag{18}$$

We can therefore rewrite (17) as a low-dimensional *unconstrained* optimization over a product of n Stiefel manifolds:

$$f^*_{\mathrm{SDPLR}}(p) = \min_{\boldsymbol{S} \in \mathrm{St}(d,p)^n} \operatorname{tr}\left(\bar{\boldsymbol{L}} \boldsymbol{S}^\mathsf{T} \boldsymbol{S}\right). \tag{19}$$

Now, let us compare the rank-restricted relaxation (19) with the original rotation averaging problem (15) and its semidefinite relaxation (16). Since any matrix $\boldsymbol{R}_i$ in the (special) orthogonal group satisfies conditions (18) with $p = d$, we have the set of inclusions:

$$\mathrm{SO}(d) \subset \mathrm{O}(d) = \mathrm{St}(d, d) \subset \cdots \subset \mathrm{St}(d, p) \subset \cdots \tag{20}$$

Therefore, we can regard the rank-restricted problems (19) as a *hierarchy* of relaxations (indexed by the rank parameter p) of (15) that are intermediate between (15) and (16) for $d < p < r$, where r is the lowest rank of any solution of (16). Indeed, if (16) has a minimizer of rank r, then it is clear by construction that for $p \geq r$ any (global) minimizer $\boldsymbol{S}^\star$ of (19) corresponds to a minimizer $\boldsymbol{Z}^\star = {\boldsymbol{S}^\star}^\mathsf{T} \boldsymbol{S}^\star$ of (16). However, unlike (16), the rank-restricted problems (19) are no longer convex, since we have reintroduced the (nonconvex) orthogonality constraints in (18). It may therefore not be clear that anything has really been gained by relaxing problem (15) to problem (19), since it seems that we may have just replaced one difficult (nonconvex) optimization problem with another. The key fact that justifies our interest in the rank-restricted relaxations (19) is the following remarkable result [7, Corollary 8]:

Theorem 2. *If $\boldsymbol{S}^\star \in St(d, p)^n$ is a rank-deficient second-order critical point of* (19)*, then $\boldsymbol{S}^\star$ is a* global *minimizer of* (19)*, and $\boldsymbol{Z}^\star = {\boldsymbol{S}^\star}^\mathsf{T} \boldsymbol{S}^\star$ is a global minimizer of the semidefinite relaxation* (16)*.*

This result immediately suggests the following simple algorithm (the **Riemannian staircase** [7]) for recovering solutions $\boldsymbol{Z}^\star$ of (16) from (19): for some small relaxation rank $p \geq d$, find a second-order critical point $\boldsymbol{S}^\star$ of problem (19) using a local search technique. If $\boldsymbol{S}^\star$ is rank-deficient, then Theorem 2 proves that $\boldsymbol{S}^\star$ is a *global* minimizer of (19), and $\boldsymbol{Z}^\star = {\boldsymbol{S}^\star}^\mathsf{T} \boldsymbol{S}^\star$ is a solution of (16). Otherwise, increase the rank parameter p and try again. In the worst possible case, we might have to take p as large as $dn + 1$ before finding a rank-deficient solution. However, in practice typically only one or two "stairs" suffice – just a *bit* of extra room in (19) vs. (15) is all one needs!

Many popular optimization algorithms only guarantee convergence to *first*-order critical points because they use only limited second-order information [24]. This is the case for the Gauss-Newton and Levenberg-Marquardt methods in particular, where the model Hessian is positive-semidefinite by construction. Fortunately, there is a simple procedure that one can use to test the global optimality of a *first*-order critical point $\boldsymbol{S}^\star$ of (19), and (if necessary) to construct a *direction of descent* that we can use to "nudge" $\boldsymbol{S}^\star$ away from stationarity before restarting local optimization at the next level of the Staircase [4].

Theorem 3. *Let $\boldsymbol{S}^\star \in St(d,p)^n$ be a first-order critical point of (19), define*

$$\boldsymbol{C} \triangleq \bar{\boldsymbol{L}} - \frac{1}{2}\,\mathrm{BlockDiag}_{d\times d}\left(\bar{\boldsymbol{L}}\boldsymbol{S}^{\star\mathsf{T}}\boldsymbol{S}^\star + \boldsymbol{S}^{\star\mathsf{T}}\boldsymbol{S}^\star\bar{\boldsymbol{L}}\right), \tag{21}$$

and let $\lambda_{\min}$ be the minimum eigenvalue of $\boldsymbol{C}$, with corresponding eigenvector $v_{\min} \in \mathbb{R}^{dn}$. Then:

(i) If $\lambda_{\min} \geq 0$, then $\boldsymbol{S}^\star$ is a global minimizer of (19), and $\boldsymbol{Z}^\star = \boldsymbol{S}^{\star\mathsf{T}}\boldsymbol{S}^\star$ is a global minimizer of (16).

(ii) If $\lambda_{\min} < 0$, then the higher-dimensional lifting $\boldsymbol{S}^+ \triangleq \left[\boldsymbol{S}^\star; 0\right] \in St(d, p+1)^n$ of $\boldsymbol{S}^\star$ is a stationary point for the rank-restricted relaxation (19) at the next level $p+1$ of the Riemannian Staircase, and $\dot{\boldsymbol{S}}^+ \triangleq \left[0; v_{\min^\mathsf{T}}\right] \in T_{\boldsymbol{S}^+}\left(St(d,p+1)^n\right)$ is a second-order descent direction from $\boldsymbol{S}^+$.

Remark 1 (Geometric interpretation of Theorems 1–3). With $\boldsymbol{C}$ defined as in (21), part (i) of Theorem 3 is simply the standard necessary and sufficient conditions for $\boldsymbol{Z}^\star = \boldsymbol{S}^{\star\mathsf{T}}\boldsymbol{S}^\star$ to be a minimizer of (16) [37]. $\boldsymbol{C}$ *also* gives the action of the Riemannian Hessian of (19) on tangent vectors; therefore, if $\lambda_{\min}(\boldsymbol{C}) < 0$, the corresponding direction of negative curvature $\dot{\boldsymbol{S}}^+$ defined in part (ii) of Theorem 3 provides a second-order descent direction from $\boldsymbol{S}^+$. Theorem 2 is an analogue of Theorem 3(i). Finally, Theorem 1 follows from the fact that the certificate matrix $\boldsymbol{C}$ depends continuously upon both the data $\bar{\boldsymbol{L}}$ and $\boldsymbol{S}^\star$, and is *always* valid for $\boldsymbol{S}^\star = \boldsymbol{R}^\star$ in the noiseless case; one can then employ a continuity argument to show that $\boldsymbol{C}$ *remains* a valid certificate for $\boldsymbol{Z}^\star = \boldsymbol{R}^{\star\mathsf{T}}\boldsymbol{R}^\star$ as a minimizer of (16) for sufficiently small noise; see [28, Appendix C] for details.

4.3 Rounding the solution

Algorithm 1 provides a truncated SVD procedure to extract a feasible point $\hat{\boldsymbol{R}} \in \mathrm{SO}(d)^n$ for the original rotation averaging problem (1) from the optimal factor $\boldsymbol{S}^\star \in \mathrm{St}(d,p)^n$ obtained via the rank-restricted relaxation (19). We need to ensure that $\hat{\boldsymbol{R}}$ is a *global minimizer* of (1) whenever the semidefinite relaxation (16) is *exact*, and is at least an *approximate* minimizer otherwise. The factor $\boldsymbol{R}$ of a symmetric factorization $\boldsymbol{Z} = \boldsymbol{R}^\mathsf{T}\boldsymbol{R}$ is only unique up to left-multiplication

Algorithm 1 Rounding procedure for solutions of (19)

Input: A minimizer $\boldsymbol{S} \in \mathrm{St}(d,p)^n$ of (19).
Output: A feasible point $\hat{\boldsymbol{R}} \in \mathrm{SO}(d)^n$ for (1).

```
1: function RoundSolution(S)
2:     Compute a rank-d truncated SVD U_d Ξ_d V_d^T for S and assign R̂ ← Ξ_d V_d^T.
3:     Set N_+ ← |{R̂_i | det(R̂_i) > 0}|.
4:     if N_+ < ⌈n/2⌉ then
5:         R̂ ← Diag(1, ..., 1, −1) R̂.
6:     end if
7:     for i = 1, ..., n do
8:         Set R̂_i ← NearestRotation(R̂_i) (see [36])
9:     end for
10:    return R̂
11: end function
```

by some $\boldsymbol{A} \in \mathrm{O}(d)$. The purpose of lines 3–6 is to choose a representative $\hat{\boldsymbol{R}}$ with a majority of $d \times d$ blocks $\hat{\boldsymbol{R}}_i$ satisfying $\det(\hat{\boldsymbol{R}}_i) > 0$, since these should *all* be rotations in the event (16) is exact [27, Sec. 4.2].

4.4 From Stiefel manifolds to rotations

In this section we show how to reformulate the low-rank optimization (19) as an optimization over the product $\mathrm{SO}(p)^n$ of rotations of p-dimensional space. This is convenient from an implementation standpoint, as affordances for performing optimization over rotations are a standard feature of many high-performance optimization libraries commonly used in robotics and computer vision [2,22,12].

The main idea underlying our approach is the simple observation that since the columns of a rotation matrix $\boldsymbol{Q} \in \mathrm{SO}(p)$ are orthonormal, then in particular the first d columns $\boldsymbol{Q}_{[1:d]}$ form an orthonormal d-frame in p-space, i.e., the submatrix $\boldsymbol{Q}_{[1:d]}$ is itself an element of $\mathrm{St}(d,p)$. Let us define the following projection, which simply extracts these first d columns:

$$\begin{aligned} \pi \colon \mathrm{SO}(p) &\to \mathrm{St}(d,p) \\ \pi(\boldsymbol{Q}) &= \boldsymbol{Q}\boldsymbol{P} \end{aligned} \tag{22}$$

where $\boldsymbol{P} = [\mathbf{I}_d; 0]$ is the $p \times d$ projection matrix appearing in (2). It is easy to see that π is surjective for any $p > d$:[4] given any element $\boldsymbol{S} = [s_1, \ldots, s_d] \in \mathrm{St}(d,p)$, we can construct a rotation $\boldsymbol{Q} \in \pi^{-1}(\boldsymbol{S})$ simply by extending $\{s_1, \ldots s_d\}$ to an orthonormal basis $\{s_1, \ldots, s_d, v_1, \ldots, v_{p-d}\}$ for $\mathbb{R}^p$ using the Gram-Schmidt process, and (if necessary) multiplying the final element v_{p-d} by -1 to ensure that this basis has a positive orientation; the matrix $\boldsymbol{Q} = [s_1, \ldots, s_d, v_1, \ldots, v_{p-d}] \in \mathrm{SO}(p)$ whose columns are the elements of this basis is then a rotation satisfying $\pi(\boldsymbol{Q}) = \boldsymbol{S}$. Conversely, if $\boldsymbol{Q} \in \pi^{-1}(\boldsymbol{S})$, then by (22) $\boldsymbol{Q} = [\boldsymbol{S}, \boldsymbol{V}]$ for some

[4] In the case that $d = p$, $\mathrm{St}(p,p) \supset \mathrm{SO}(p)$, and it is impossible for π to be surjective.

$\boldsymbol{V} \in \mathbb{R}^{p\times(p-d)}$; writing the orthogonality constraint $\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q} = \mathbf{I}_p$ in terms of this block decomposition then produces:

$$\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{S}^\mathsf{T}\boldsymbol{S} & \boldsymbol{S}^\mathsf{T}\boldsymbol{V} \\ \boldsymbol{V}^\mathsf{T}\boldsymbol{S} & \boldsymbol{V}^\mathsf{T}\boldsymbol{V} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_d & 0 \\ 0 & \mathbf{I}_{p-d} \end{bmatrix}. \tag{23}$$

It follows from (23) that the preimage of any $\boldsymbol{S} \in \mathrm{St}(d,p)$ under the projection π in (22) is given explicitly by:

$$\pi^{-1}(\boldsymbol{S}) = \left\{ [\boldsymbol{S}, \boldsymbol{V}] \mid \ \boldsymbol{V} \in \mathrm{St}(p-d, p),\ \boldsymbol{S}^\mathsf{T}\boldsymbol{V} = 0,\ \det([\boldsymbol{S},\boldsymbol{V}]) = +1 \right\}. \tag{24}$$

Equations (22) and (24) provide a means of representing $\mathrm{St}(d,p)$ using $\mathrm{SO}(p)$: given any $\boldsymbol{S} \in \mathrm{St}(d,p)$, we can represent it using one of the rotations in (24) in which it appears as the first d columns, and conversely, given any $\boldsymbol{Q} \in \mathrm{SO}(p)$, we can extract its corresponding Stiefel manifold element using π. We can extend this relation to *products* of rotations and Stiefel manifolds in the natural way:

$$\begin{aligned} \Pi \colon \mathrm{SO}(p)^n &\to \mathrm{St}(d,p)^n \\ \Pi(\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_n) &= (\pi(\boldsymbol{Q}_1), \dots, \pi(\boldsymbol{Q}_n)) \end{aligned} \tag{25}$$

and Π is likewise surjective for $p > d$.

The projection Π enables us to "pull back" the rank-restricted optimization (19) on $\mathrm{St}(d,p)^n$ to an equivalent optimization problem on $\mathrm{SO}(p)^n$. Concretely, if $f \colon \mathrm{St}(d,p)^n \to \mathbb{R}$ is the objective of (19), then we simply define the objective $\tilde{f}$ of our "lifted" optimization over $\mathrm{SO}(p)^n$ to be the pullback of f via Π:

$$\begin{aligned} \tilde{f} \colon \mathrm{SO}(p)^n &\to \mathbb{R} \\ \tilde{f}(\boldsymbol{Q}) &= f \circ \Pi(\boldsymbol{Q}). \end{aligned} \tag{26}$$

Comparing (19), (25), and (26) reveals that the pullback of the low-rank optimization (19) to $\mathrm{SO}(p)^n$ is exactly the Shonan Averaging problem (2).

Theorem 4. *Let $p > d$. Then:*

- *(i) The rank-restricted optimization (19) and the Shonan Averaging problem (2) attain the same optimal value.*
- *(ii) $\boldsymbol{Q}^\star \in \mathrm{SO}(p)^n$ minimizes (2) if and only if $\boldsymbol{S}^\star = \Pi(\boldsymbol{Q}^\star)$ minimizes (19).*

Similarly, we also have the following analogue of Theorem 3 for problem (2):

Theorem 5. *Let $\boldsymbol{Q}^\star = (\boldsymbol{Q}_1^\star, \dots, \boldsymbol{Q}_n^\star) \in \mathrm{SO}(p)^n$ be a first-order critical point of problem (2), and $\boldsymbol{S}^\star = (\boldsymbol{S}_1^\star, \dots, \boldsymbol{S}_n^\star) = \Pi(\boldsymbol{Q}^\star) \in St(d,p)^n$. Then:*

- *(i) $\boldsymbol{S}^\star$ is a first-order critical point of the rank-restricted optimization (19).*
- *(ii) Let $\boldsymbol{C}$ be the certificate matrix defined in (21), $\lambda_{\min}$ its minimum eigenvalue, and $v = (v_1, \dots, v_n) \in \mathbb{R}^{dn}$ a corresponding eigenvector. If $\lambda_{\min} < 0$, then the point $\boldsymbol{Q}^+ \in \mathrm{SO}(p+1)^n$ defined elementwise by:*

$$\boldsymbol{Q}_i^+ = \begin{bmatrix} \boldsymbol{Q}_i^\star & 0 \\ 0 & 1 \end{bmatrix} \in \mathrm{SO}(p+1) \tag{27}$$

Algorithm 2 Shonan Rotation Averaging

Input: An initial point $\boldsymbol{Q} \in \mathrm{SO}(p)^n$ for (2), $p \geq d$.
Output: A feasible estimate $\hat{\boldsymbol{R}} \in \mathrm{SO}(d)^n$ for the rotation averaging problem (1), and the lower bound f^*_{SDP} for problem (1)'s optimal value.

1: **function** ShonanAveraging($\boldsymbol{Q}$)
2:     **repeat** ▷ Riemannian Staircase
3:         $\boldsymbol{Q} \leftarrow$ LocalOpt($\boldsymbol{Q}$) ▷ Find critical point of (2)
4:         Set $\boldsymbol{S} \leftarrow \Pi(\boldsymbol{Q})$ ▷ Project to $\mathrm{St}(d,p)^n$
5:         Construct the certificate matrix $\boldsymbol{C}$ in (21), and compute its minimum eigenpair $(\lambda_{\min}, v_{\min})$.
6:         **if** $\lambda_{\min} < 0$ **then** ▷ $\boldsymbol{Z} = \boldsymbol{S}^\mathsf{T}\boldsymbol{S}$ is *not* optimal
7:             Set $p \leftarrow p+1$ and $\boldsymbol{Q}_i \leftarrow \begin{pmatrix} \boldsymbol{Q}_i & 0 \\ 0 & 1 \end{pmatrix}$ $\forall i$. ▷ Ascend to next level
8:             Construct descent direction $\dot{\boldsymbol{Q}}$ as in (28).
9:             $\boldsymbol{Q} \leftarrow$ LineSearch($\boldsymbol{Q}, \dot{\boldsymbol{Q}}$) ▷ Nudge $\boldsymbol{Q}$
10:         **end if**
11:     **until** $\lambda_{\min} \geq 0$ ▷ $\boldsymbol{Z} = \boldsymbol{S}^\mathsf{T}\boldsymbol{S}$ solves (16)
12:     Set $f^*_{\mathrm{SDP}} \leftarrow \operatorname{tr}\left(\bar{\boldsymbol{L}}\boldsymbol{S}^\mathsf{T}\boldsymbol{S}\right)$.
13:     Set $\hat{\boldsymbol{R}} \leftarrow$ RoundSolution($\boldsymbol{S}$).
14:     **return** $\left\{\hat{\boldsymbol{R}}, f^*_{\mathrm{SDP}}\right\}$
15: **end function**

is a first-order critical point of problem (2) in dimension $p+1$, and the tangent vector $\dot{\boldsymbol{Q}}^+ \in T_{\boldsymbol{Q}^+}(\mathrm{SO}(p+1)^n)$ defined blockwise by:

$$\dot{\boldsymbol{Q}}^+_i = \boldsymbol{Q}^+_i \begin{bmatrix} 0 & -v_i \\ v_i^\mathsf{T} & 0 \end{bmatrix} \in T_{\boldsymbol{Q}^+_i}(\mathrm{SO}(p+1)) \tag{28}$$

is a second-order descent direction from $\boldsymbol{Q}^+$.

Theorems 4 and 5 are proved in Appendix **??** of the supplementary material.

4.5 The complete algorithm

Combining the results of the previous sections gives the complete *Shonan Averaging* algorithm (Algorithm 2). (We employ Levenberg-Marquardt to perform the fast local optimization required in line 3 – see Appendix **??** for details).

When applied to an instance of rotation averaging, Shonan Averaging returns a feasible point $\hat{\boldsymbol{R}} \in \mathrm{SO}(d)^n$ for the maximum likelihood estimation (1), and a lower bound $f^*_{\mathrm{SDP}} \leq f^*_{\mathrm{MLE}}$ for problem (1)'s optimal value. This lower bound provides an *upper* bound on the suboptimality of *any* feasible point $\boldsymbol{R} \in \mathrm{SO}(d)^n$ as a solution of problem (1) according to:

$$f(\bar{\boldsymbol{L}}\boldsymbol{R}^\mathsf{T}\boldsymbol{R}) - f^*_{\mathrm{SDP}} \geq f(\bar{\boldsymbol{L}}\boldsymbol{R}^\mathsf{T}\boldsymbol{R}) - f^*_{\mathrm{MLE}}. \tag{29}$$

Furthermore, if the relaxation (16) is exact, the estimate $\hat{\boldsymbol{R}}$ returned by Algorithm 2 *attains* this lower bound:

$$f(\bar{\boldsymbol{L}}\hat{\boldsymbol{R}}^\mathsf{T}\hat{\boldsymbol{R}}) = f^*_{\mathrm{SDP}}. \tag{30}$$

			Shonan Averaging				
	n	m	p	$\lambda_{\min}$	f^*_{SDP}	Opt. time [s]	Min. eig. time [s]
smallGrid	125	297	5	-9.048×10^{-6}	4.850×10^{2}	2.561×10^{-1}	2.613×10^{-1}
sphere	2500	4949	6	-1.679×10^{-4}	5.024×10^{0}	1.478×10^{1}	1.593×10^{1}
torus	5000	9048	5	-2.560×10^{-4}	1.219×10^{4}	7.819×10^{1}	8.198×10^{1}
garage	1661	6275	5	1.176×10^{-4}	2.043×10^{-1}	1.351×10^{1}	1.420×10^{1}
cubicle	5750	16869	5	-9.588×10^{-5}	3.129×10^{1}	1.148×10^{2}	1.208×10^{2}

Table 1. Shonan Averaging results for the SLAM benchmark datasets

Consequently, verifying *a posteriori* that (30) holds provides a *certificate* of $\hat{\boldsymbol{R}}$'s *global* optimality as a solution of the rotation averaging problem (1).

5 Experimental Results

In this section we evaluate Shonan Averaging's performance on (a) several large-scale problems derived from standard pose-graph SLAM benchmarks, (b) small randomly generated synthetic datasets, and (c) several structure from motion (SFM) datasets used in earlier work. The experiments were performed on a desktop computer with a 6-core Intel i5-9600K CPU @ 3.70GHz running Ubuntu 18.04. Our implementation of Shonan Averaging (SA) is written in C++, using the GTSAM library [12] to perform the local optimization. The fast minimum-eigenvalue computation described in [26, Sec. III-C] is implemented using the symmetric Lanczos algorithm from the Spectra library[5] to compute the minimum eigenvalue required in line 5. We initialize the algorithm with a randomly-sampled point $\boldsymbol{Q} \in \mathrm{SO}(p)^n$ at level $p = 5$, and require that the minimum eigenvalue satisfy $\lambda_{\min} \geq -10^{-4}$ at termination of the Riemannian Staircase.

The results for the pose-graph SLAM benchmarks demonstrate that Shonan Averaging is already capable of recovering *certifiably globally optimal* solutions of large-scale real-world rotation averaging problems in tractable time. For these experiments, we have extracted the rotation averaging problem (1) obtained by simply ignoring the translational parts of the measurements. Table 1 reports the size of each problem (the number of unknown rotations n and relative measurements m), together with the relaxation rank p at the terminal level of the Riemannian Staircase, the optimal value of the semidefinite relaxation (16) corresponding to the recovered low-rank factor $\boldsymbol{S}^\star$, and the total elapsed computation times for the local optimizations in line 3 and minimum-eigenvalue computations in line 5 of Algorithm 2, respectively. We remark that the datasets torus in Table 1 is *1-2 orders of magnitude larger* than the examples reported in previous work on direct globally-optimal rotation averaging algorithms [13,16].

We attribute Shonan Averaging's improved scalability to its use of superlinear *local* optimization in conjunction with the Riemanian Staircase; in effect, this strategy provides a means of "upgrading" a fast *local* method to a fast *global*

[5] https://Spectralib.org/

one. This enables Shonan Averaging to leverage existing heavily-optimized, scalable, and high-performance software libraries for superlinear *local* optimization [2,22,12] as the main computational engine of the algorithm (cf. line 3 of Algorithm 2), while preserving the guarantee of *global* optimality.

n,σ	method	error	time (s)	success
n=20 σ=0.2	SA	0.000%	0.030	100%
	BD	1.397%	0.374	60%
	LM	-0.001%	0.003	80%
n=20 σ=0.5	SA	0.000%	0.037	100%
	BD	0.014%	0.446	40%
	LM	-0.001%	0.004	20%
n=50 σ=0.2	SA	0.000%	0.141	100%
	BD	0.291%	6.261	60%
	LM	-0.039%	0.011	20%
n=50 σ=0.5	SA	0.000%	0.194	100%
	BD	-7.472%	6.823	60%
	LM	-0.008%	0.014	40%
n=100 σ=0.2	SA	0.000%	0.403	100%
	BD	nan%	nan	0%
	LM	-0.014%	0.016	20%
n=100 σ=0.5	SA	0.000%	0.373	100%
	BD	3.912%	32.982	20%
	LM	-0.142%	0.023	60%
n=200 σ=0.2	SA	0.000%	1.275	100%
	BD	nan%	nan	0%
	LM	-0.102%	0.049	60%
n=200 σ=0.5	SA	0.000%	1.845	100%
	BD	nan%	nan	0%
	LM	-0.293%	0.046	40%

Table 2. Synthetic results for varying problem sizes n and noise levels σ. SA=Shonan, BD=block coordinate descent from [13], LM=Levenberg-Marquardt.

The synthetic results confirm that Shonan Averaging always converges to the true (*global*) minimizer, and significantly outperforms the block coordinate descent method from [13,14] as soon as the number of unknown rotations becomes large. We followed the approach for generating the data from [13], generating 4 sets of problem instances of increasing size, for two different noise levels. We generated 3D poses on a circular trajectory, forming a cycle graph. Relative rotation measurements were corrupted by composing with a random axis-angle perturbation, where the axis was chosen randomly and the angle was generated from a normal distribution with a standard deviation $\sigma = 0.2$ or $\sigma = 0.5$. Initialization for all three algorithms was done randomly, with angles uniform random over the range $(-\pi, \pi)$. We compare our results with two baselines: (LM) Levenberg-Marquardt on the $\mathrm{SO}(3)^n$ manifold, also implemented in GT-

dataset	method	error	time	success
reichstag	SA	0.000%	0.197	100%
(n=71, m=2554)	BD	0.000%	0.247	100%
	LM	0.001%	0.085	80%
pantheon_interior	SA	0.000%	0.971	100%
(n=186, m=10000)	BD	0.000%	1.823	100%
	LM	0.001%	0.313	100%

Table 3. Results on SFM problems from the YFCC dataset.

SAM, which is fast but not expected to find the global minimizer; (BD): the block coordinate descent method from [13,14]. Because no implementation was available, we re-implemented it in Python, using a 3×3 SVD decomposition at the core. However, our implementation differs in that we establish optimality using the eigenvalue certificate from Shonan.

The synthetic results are shown in Table 2. In the table we show the percentage of the cases in which either method found a global minimizer, either as certified by the minimum eigenvalue $\lambda_{\min} \geq -10^{-4}$, or being within 5% of the optimal cost. Shonan Averaging ("SA" in the table) finds an optimal solution every time as certified by $\lambda_{\min}$. Levenberg-Marquardt is fast but finds global minima in only about 40% to 60% of the cases. Finally, the block coordinate descent method (BD) is substantially slower, and does not always converge within the allotted time. We found that in practice BD converges very slowly and it takes a long time for the algorithm to converge to the same value as found by SA or LM. In the table we have shown the relative error and running time (in seconds), both averaged over 5 runs. For $n = 100$ and $n = 200$ the BD method did often not converge to the global optimum within reasonable time.

Finally, Table 3 shows results on two datasets derived by Heinly et al. [21] from the large-scale YFCC-100M dataset [33,21]. The relative measurements for these were derived from the SFM solution provided with the data, and corrupted with noise as before, using $\sigma = 0.2$. All three methods agree on the globally optimal solution, and the timing data shows the same trend as in Table 2.

6 Conclusion

In this work we presented Shonan Rotation Averaging, a fast, simple, and elegant algorithm for rotation averaging that is *guaranteed* to recover globally optimal solutions under mild conditions on the measurement noise. Our approach applies a fast *local* search technique to a *sequence* of higher-dimensional lifts of the rotation averaging problem until a globally optimal solution is found. Shonan Averaging thus leverages the speed and scalability of existing high-performance *local* optimization methods already in common use, while enabling the recovery of *provably optimal* solutions.

References

1. P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2007. 3, 7
2. Sameer Agarwal and Keir Mierle. *Ceres Solver: Tutorial & Reference*. Google Inc., 2012. 2, 3, 9, 13
3. S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *European Conf. on Computer Vision (ECCV)*, pages 29–42, 2010. 1
4. N. Boumal. A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints. *arXiv preprint arXiv:1506.00575*, 2015. 8
5. N. Boumal, A. Singer, and P.A. Absil. Cramer-rao bounds for synchronization of rotations. *Information and Inference*, 3(1):1–39, 2014. 2
6. N. Boumal, A. Singer, and P.-A. Absil. Robust estimation of rotations from relative measurements by maximum likelihood. In *52nd IEEE Conference on Decision and Control*, pages 1156–1161. IEEE, 2013. 2
7. N. Boumal, V. Voroninski, and A. Bandeira. The non-convex Burer–Monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2757–2765. 2016. 7
8. S. Burer and R. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003. 6
9. L. Carlone, D.M. Rosen, G.C. Calafiore, J.J. Leonard, and F. Dellaert. Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 125–132, 2015. (pdf) (code) (datasets: (web)) (supplemental material: (pdf)). 2, 3
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 *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4597–4604, 2015. (pdf) (code) (supplemental material: (pdf)). 2
11. T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, NY, 1991. 1
12. Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology, September 2012. 2, 3, 9, 12, 13
13. A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin. Rotation averaging and strong duality. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 12, 13, 14
14. Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging with the chordal distance: Global minimizers and strong duality. *IEEE Trans. Pattern Anal. Machine Intell.*, 2019. 3, 13, 14
15. J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. Building Rome on a cloudless day. In *European Conference on Computer Vision*, pages 368–381. Springer, 2010. 2
16. J. Fredriksson and C. Olsson. Simultaneous multiple rotation averaging using lagrangian duality. In *Asian Conf. on Computer Vision (ACCV)*, 2012. 3, 12
17. V.M. Govindu. Combining two-view constraints for motion estimation. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2. IEEE, 2001. 1
18. R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *IJCV*, 103(3):267–305, 2013. 1, 2, 3

19. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1
20. J. Heinly, J.L. Schonberger, E. Dunn, and J.-M. Frahm. Reconstructing the world* in six days*(as captured by the Yahoo 100 million image dataset). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3287–3295, 2015. 2
21. Jared Heinly, Johannes L. Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days *(as captured by the Yahoo 100 million image dataset). In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 14
22. R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2011. 2, 3, 9, 13
23. D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 2
24. Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, 1999. 2, 3, 8
25. R. Pless. Using many cameras as one. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 587–593, 2003. 1
26. D.M. Rosen and L. Carlone. Computational enhancements for certifiably correct SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017. Workshop on "Introspective Methods for Reliable Autonomy", (pdf). 2, 12
27. D.M. Rosen, L. Carlone, A.S. Bandeira, and J.J. Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the Special Euclidean group. In *Intl. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, San Francisco, CA, December 2016. extended arxiv preprint: 1611.00128, (pdf) (pdf) (code). 2, 3, 5, 6, 9
28. D.M. Rosen, L. Carlone, A.S. Bandeira, and J.J. Leonard. SE-Sync: a certifiably correct algorithm for synchronization over the Special Euclidean group. *Intl. J. of Robotics Research*, 2018. accepted, arxiv preprint: 1611.00128, (pdf). 8
29. J.L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1
30. A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Appl. Comput. Harmon. Anal.*, 30:20–36, 2010. 2
31. A. Singer and H.-T. Wu. Vector diffusion maps and the connection Laplacian. *Comm. Pure Appl. Math.*, 65:1067–1144, 2012. 5
32. J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999. 3
33. Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: the new data in multimedia research. *Commun. ACM*, 59(2):64–73, 2016. 14
34. B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *LNCS*, pages 298–372. Springer Verlag, 2000. 1
35. R. Tron, R. Vidal, and A. Terzis. Distributed pose averaging in camera networks via consensus on SE(3). In *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–10. IEEE, 2008. 1
36. S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(4):376 –380, April 1991. 9

37. L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996. 3, 6, 8
38. K. Wilson, D. Bindel, and N. Snavely. When is rotations averaging hard? In *European Conference on Computer Vision*, pages 255–270. Springer, 2016. 3