When is Rotations Averaging Hard?

Kyle Wilson[1,3](✉), David Bindel[3], and Noah Snavely[2,3]

[1] Washington College, Chestertown, MD, USA
kwilson24@washcoll.edu
[2] Google Inc., Mountain View, CA, USA
[3] Cornell University, Ithaca, NY, USA
{bindel,snavely}@cs.cornell.edu

Abstract. Rotations averaging has become a key subproblem in global Structure from Motion methods. Several solvers exist, but they do not have guarantees of correctness. They can produce high-quality results, but also sometimes fail. Our understanding of what makes rotations averaging problems easy or hard is still very limited. To investigate the difficulty of rotations averaging, we perform a local convexity analysis under an L_2 cost function. Although a previous result has shown that in general, this problem is locally convex almost nowhere, we show how this negative conclusion can be reversed by considering the gauge ambiguity. Our theoretical analysis reveals the factors that determine local convexity—noise and graph structure—as well as how they interact, which we describe by a particular Laplacian matrix. Our results are useful for predicting the difficulty of problems, and we demonstrate this on practical datasets. Our work forms the basis of a deeper understanding of the key properties of rotations averaging problems, and we discuss how it can inform the design of future solvers for this important problem.

1 Introduction

Rotations averaging is the problem of assigning a rotation matrix to every vertex in a graph, in a way that best respects given relative rotations on each edge. This problem has become a staple of recent global Structure from Motion (SfM) methods, where the vertices represent cameras and the rotation matrices are their orientations [1–4]. In many global SfM approaches, camera orientations and positions of many photographs are recovered by (1) estimating relative poses among pairs or triplets of cameras, (2) computing camera orientations via rotations averaging, and (3) computing camera positions from translation direction constraints.

Despite the practical success of recent rotations averaging methods, they largely come without guarantees. Indeed, the cost functions in question are non-convex. Both L_1 and L_2 formulations of rotations averaging can have local minima. Beyond these facts, little is known about the practical properties of rotation averaging problems—in particular, what makes a problem easy or hard?

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46478-7_16) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VII, LNCS 9911, pp. 255–270, 2016.
DOI: 10.1007/978-3-319-46478-7_16

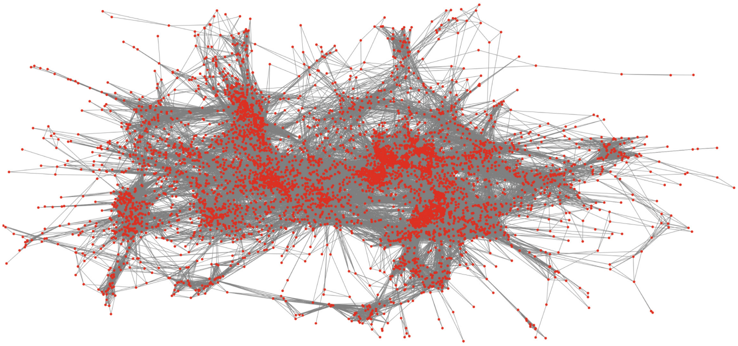

Fig. 1. In Structure from Motion, each vertex in a rotations averaging problem represents a camera's orientation. Edges are measurements of the relative rotation between two cameras. Some real rotations averaging problems have complicated graph structure, such as the Arts Quad problem pictured above [5].

An instance of the rotation averaging problem is a graph with a measurement on each edge. Figure 1 shows this measurement graph for the ArtsQuad dataset [5], a difficult real-world SfM problem. Intuitively, the performance of a solver should depend on the structure of the graph (dense vs. sparse, well connected, clustered, etc.), as well as the inherent noise level of the measurements.

The goal of this paper is to seek a principled answer to what makes a given problem easy or hard. We pursue that via a local convexity analysis. We show that the extent of local convexity depends on the smallest eigenvalue of a particular *normalized graph Laplacian.* Such eigenvalues have found broad application in describing random walks and diffusion on graphs, and are related to many combinatorial graph properties [6].

Our results provide insight into the sources of problem difficulty. We see that well-connected problems are easier, but that larger and noisier problems may be difficult. This could motivate a future multistage approach that solves larger, less connected problems by first considering small, simpler, well-connected subproblems.

2 Related Work

Rotations averaging was first proposed within the vision community in Govindu's pioneering work on global SfM [7]. Like most subsequent methods, that paper computes orientations for many cameras, subject to relative orientations between some pairs of cameras. The solver in [7] is based on a quaternion representation of rotations. If the constraint that a rotation is a *unit* quaternion is relaxed, minimizing the difference between quaternions is a linear problem. However, Hartley et al. [8] later showed that this method does not exactly minimize a reasonable cost function, due in part to quaternions only representing unique rotations up to a sign. Martinec and Pajdla [9] propose a similar solver: in the spirit of [7], they represent rotations as orthogonal matrices and relax the orthonormality constraints. This is again a linear problem, and is reported to work better than [7] in practice. Arrigoni et al. [10] augment the orthogonality relaxation with a low-rank/sparse decomposition to be more robust to outlier measurements. Wang and Singer [11] propose an unsquared version of the cost in [9] and show that the correct answer is exactly recovered under a certain noise model.

Crandall et al. [5] take an entirely different approach. They simplify the problem greatly by assuming rotations without roll, and solve under robust cost functions with a Markov Random Field. This has the advantage of robustness to outlier data, but uses a complicated general purpose solver with many parameters, rather than taking advantage of problem structure.

A third category of *intrinsic* solvers makes explicit use of the geometric structure of the rotations group. Govindu [12] iteratively projects the problem into tangent spaces, solving a Euclidean problem at each step. Tron et al. [13,14] give a distributed consensus algorithm for sensor arrays. Hartley et al. [15] also give a consensus algorithm, this time motivated by the Weiszfeld algorithm for Euclidean averaging. They minimize an L_1 cost function, which is considered to be more robust to noisy input. Chatterjee and Govindu [16] also give an iterative tangent space scheme, but this time minimize a Huber-like robust loss. We will analyze an intrinsic cost function in this paper.

Many of these methods can produce high quality results, but none of them come with guarantees of the correctness or optimality of the underlying problem. Fredriksson and Olsson [17] seek to verify the optimality of a solution. They frame a dual problem such that if the dual and the original problem have the same optimal cost then the solution is guaranteed to be optimal. In practice, this works on problems with small inlier noise. We do not offer this chance at a guarantee of optimality, but we instead provide broader insight into which problems are easy.

Closely related to our work, other papers have also discovered connections to the eigenvalues of graph Laplacians. Bandeira et al. [18] analyze the worst case performance of a spectral algorithm for the closely related problem of synchronization on the orthogonal group $O(n)$, finding that it depends on the smallest eigenvalues of a graph Laplacian. In [19,20], Boumal et al. take a statistical modeling approach to rotations averaging and compute Cramér-Rao lower bounds for maximum likelihood estimators. As in our results, these bounds are in terms of the eigenvalues of graph Laplacians with boundary conditions. These results are concerned with the quality of solutions, but not with distinguishing between local and global minima.

3 Representing Rotations

Rotations averaging attempts to assign a 3D rotation to every vertex in a graph, where often these vertices correspond to cameras. In this section we give preliminaries by describing two representations for 3D rotations: rotation matrices and angle-axis vectors.

Rotation Matrices. A rotation matrix is a 3×3 real orthogonal matrix with determinant 1. The set of all such rotations is the *special orthogonal group* $\mathrm{SO}(3)$. The group's operation is the usual matrix product. Understood this way, $\mathrm{SO}(3)$ is a three dimensional manifold inside $\mathbb{R}^{3\times 3}$.

Angle-Axis Representation. Euler's rotation theorem shows that any rotation $\mathtt{R}$ may be viewed geometrically as a rotation by some angle θ around some unit vector $\mathbf{v}$. The vector $\theta\mathbf{v} \in \mathbb{R}^3$ is the angle-axis representation of $\mathtt{R}$. The angle-axis representation is not unique, since $\theta\mathbf{v} \sim (2\pi - \theta)(-\mathbf{v})$. A common convention is to restrict $\theta \in [0, \pi]$, which is only ambiguous for $\theta = 0$ and $\theta = \pi$. See [8] for conversion formulas between rotations matrices and angle-axis vectors.

The Tangent Space. Rotation matrices and angle-axis vectors are connected in a deep way. Since $\mathrm{SO}(3)$ is a 3D manifold in $\mathbb{R}^{3\times 3}$, at any point $\mathtt{R}$ on $\mathrm{SO}(3)$ there is a 3D subspace of directions where an infinitesimal step remains on the manifold (this is the tangent space at $\mathtt{R}$), and an orthogonal 6D subspace of directions that step away from $\mathrm{SO}(3)$. In fact, $\mathrm{SO}(3)$ is a *Lie group*—a continuous symmetry group—and its tangent space at the identity (the *Lie algebra*) is the additive group of skew-symmetric 3×3 matrices. For any differentiable manifold, there are maps between the tangent space at a point and the manifold in the neighborhood of that point: $\exp_{\mathtt{R}}$ takes a step in the tangent space at $\mathtt{R}$ to a point on the manifold, and $\log_{\mathtt{R}}$ maps a point on the manifold into the tangent space at $\mathtt{R}$. Because $\mathrm{SO}(3)$ is a Lie group there is a simple connection between the tangent spaces at a rotation $\mathtt{S}$ and the Lie algebra:

$$\exp_{\mathtt{S}}(\varOmega) = \mathtt{S}\exp_{\mathtt{I}}(\varOmega) \tag{1}$$

where $\varOmega$ is any skew matrix. Moreover, at the identity $\mathtt{I} \in \mathrm{SO}(3)$, the exponential and log maps are exactly the conversions between rotation matrices and angle-axis vectors:

$$\log_{\mathtt{I}}(\mathtt{R}) = \theta[\mathbf{v}]_\times \quad \text{and} \quad \exp_{\mathtt{I}}\left(\theta[\mathbf{v}]_\times\right) = \mathtt{R} \tag{2}$$

where $[\cdot]_\times$ denotes the cross product matrix:

$$[\mathbf{v}]_\times = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}_\times = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \tag{3}$$

We will write exp and log for $\exp_{\mathtt{I}}$ and $\log_{\mathtt{I}}$. These are precisely the ordinary matrix exponent and log.

Distances on $\mathrm{SO}(3)$. There are several reasonable metrics on $\mathrm{SO}(3)$ [8]. In this paper we will be concerned with the *angular distance*, $d_\angle(\cdot,\cdot)$. This is the angle of the relative rotation between two rotation matrices (here $\mathtt{R}$ and $\mathtt{S}$):

$$d_\angle(\mathtt{R},\mathtt{S}) = \frac{1}{2}\|\log(\mathtt{R}\mathtt{S}^{-1})\|_2 \tag{4}$$

since for any rotation $\mathtt{Q} = \exp(\theta[\mathbf{v}]_\times)$, $\frac{1}{2}\|\log(\mathtt{Q})\|_2 = \frac{1}{2}\|\theta[\mathbf{v}]_\times\|_2 = \|\theta\mathbf{v}\|_2 = \theta$. This is the most natural metric on $\mathrm{SO}(3)$, also called the *geodesic distance*.

4 Rotations Averaging Problems

In this section we introduce rotations averaging problems and consider some of their properties. In SfM, each camera in a scene has a 3D orientation (i.e. the yaw, pitch, and roll of the camera). We represent these orientations as rotation matrices, which map from a world coordinate system to a camera-centered system.

Problems and Solutions. A *rotations averaging problem*$(G, \widetilde{\mathcal{R}})$ is a graph $G = (V, E)$ where vertices represent absolute rotations, and edges are annotated with measurements $\widetilde{\mathcal{R}} : E \rightarrow \mathrm{SO}(3)$ of relative rotation. We will write $V = \{1, 2, \ldots, n\}$ and assume that G is connected. A *solution* $\mathcal{R} = (\mathtt{R}_1, \ldots, \mathtt{R}_n)$ is an assignment of absolute rotations to vertices.[1]

Cost Function. We measure the quality of a solution by how well the *measured*[2] relative rotation $\widetilde{\mathtt{R}}_{ij}$ on each edge (i, j) matches the *modeled* relative rotation $\mathtt{R}_i\mathtt{R}_j^\top$. We quantify this as ϕ^2, the L_2 rotations averaging cost function:

$$\phi^2(\mathcal{R}) = \sum_{(i,j)\in E} \left(d_\angle(\widetilde{\mathtt{R}}_{ij}, \mathtt{R}_i\mathtt{R}_j^\top)\right)^2 \tag{5}$$

We will often refer to the *residuals* $\mathtt{R}_i^\top\widetilde{\mathtt{R}}_{ij}\mathtt{R}_j$ in their angle-axis form:

$$\log\left(\mathtt{R}_i^\top\widetilde{\mathtt{R}}_{ij}\mathtt{R}_j\right) = \theta_{ij}\mathbf{w}_{ij} \tag{6}$$

so that the objective function ϕ^2 becomes $\phi^2(\mathcal{R}) = \sum_{(i,j)\in E} \theta_{ij}^2$.

Gauge Ambiguity. If $\phi^2(\mathcal{R}) = c$, then $\phi^2(\mathcal{R}\mathtt{S}) = c$ as well, where $\mathcal{R}\mathtt{S} = (\mathtt{R}_1\mathtt{S}, \ldots, \mathtt{R}_n\mathtt{S})$. We see that solutions are invariant to a global rotation. This is the standard *gauge ambiguity*, and it is always understood that solutions are only unique up to such a global rotation. The gauge ambiguity can be "fixed" by arbitrarily setting the value of exactly one rotation; for example, requiring $\mathtt{R}_1 = \mathtt{I}$. We will see later that appreciating the gauge ambiguity is crucial in revealing convexity structure in rotations averaging problems.

Hardness. Because no closed-form way to find globally optimal solutions to rotations averaging is known, solvers proceed iteratively. That is, the user supplies a preliminary guessed solution, and then that solution is refined by taking a series of steps in directions which reduce ϕ^2. These initial guesses can be generated at random, or from spanning trees, but more commonly come from *relaxed problems* [7,9] which do have closed-form solutions, but may be only a loose proxy for the true problem. We would like to know how good of a guess is necessary. We will approach this question by asking where ϕ^2 is locally convex.

[1] We will usually wish to reason about G in an undirected manner, since a measurement $\widetilde{\mathtt{R}}_{ij}$ on (i, j) is equivalent to $\widetilde{\mathtt{R}}_{ji} = \widetilde{\mathtt{R}}_{ij}^\top$ on (j, i).

[2] Throughout this paper, we use tildes, such as $\widetilde{\mathcal{R}}$ and $\widetilde{\mathtt{R}}_{ij}$, to represent measured quantities—inputs to the problem. We use light block fonts, such as $\mathtt{R}_i$ and $\widetilde{\mathtt{R}}_{ij}$ for rotation matrices, and caligraphic fonts, such as $\mathcal{R}$ and $\widetilde{\mathcal{R}}$ for sets of things.

Local Convexity. Optimizing convex functions is easy: they have the property that all guesses are "good enough" to get to the right answer. That is, all local minima of a convex function are also global minima. Unfortunately, rotations averaging is not convex. However, we can consider the weaker property of *local convexity*. A problem $(G, \widetilde{\mathcal{R}})$ is locally convex at a solution $\mathcal{R}$ if there is some ball around $\mathcal{R}$ on which the problem is convex. A function that is locally convex on all of a convex domain is convex on that domain.

Functions are locally convex where the eigenvalues of their second derivative (Hessian) matrices are all non-negative—that is, when the Hessian is positive semi-definite. Local convexity can be a sufficient property for an optimization problem to be easy if the problem is locally convex in a large region around a global minimum. Even when local convexity fails, a function whose Hessian is more nearly positive-definite is less prone to having local minima.

Matrices Associated to Problems. As a result of the graph structure underlying rotations averaging problems, when we inspect their Hessian matrices, we will find components of some well-studied matrices in spectral graph theory. We will define those here and indicate their other uses.

Consider our graph G with each edge (i, j) weighted by θ_{ij}. These θ will later be the residuals that come from evaluating ϕ^2 at a particular solution. We write $i \sim j$ if i and j are neighbors. The degree $\delta(i; \theta)$ of a vertex $i \in V$ is $\sum_{i \sim j} \theta_{ij}$, and the maximum degree $\Delta(\theta) = \max_{v \in V} \delta(v; \theta)$. (We continue to emphasize the weights θ because we will need to distinguish between different sets of weights.)

The degree matrix $\mathbf{D}(\theta)$ is the diagonal matrix $\operatorname{diag}([\delta(1; \theta) \cdots \delta(n; \theta)])$ and the adjacency matrix $\mathbf{A}(\theta)$ has entries $\mathbf{A}_{ij} = \theta_{ij} \mathbb{1}(i \sim j)$, where $\mathbb{1}$ is the boolean indicator function. The graph Laplacian is $\mathbf{L}(\theta) = \mathbf{D}(\theta) - \mathbf{A}(\theta)$. Because the rows and columns of $\mathbf{L}$ sum to zero, the smallest eigenvalue λ_1 of $\mathbf{L}(\theta)$ is always 0 with corresponding eigenvector $[1, 1, \ldots, 1]$. If G is connected, then $\lambda_2 > 0$. This second-smallest eigenpair has special significance and is used for spectral clustering. In the unweighted case when all $\theta_{ij} = 1$ we write simply $\mathbf{D}$, $\mathbf{A}$, and $\mathbf{L}$. Then λ_2 is called the *algebraic connectivity*.

Further varieties of graph Laplacians arise in practice. The normalized graph Laplacian has the form $\mathbf{D}(\theta)^{-1/2}\mathbf{L}(\theta)\mathbf{D}(\theta)^{-1/2}$. Normalized graph Laplacians have been used for image segmentation [21] and are also known to be closely connected to many combinatorial graph measures [6]. In the following sections we will also encounter a normalized graph Laplacian *with boundary conditions*, similar to Laplacians which arise in the numerical solutions to Poisson's equation [22].

5 Local Convexity Theorems for Rotations Averaging

In this section we develop a sufficient condition for rotations averaging to be locally convex. The proof will work by finding a condition that implies that the Hessian of ϕ^2 positive definite. Since ϕ^2 is a sum of terms, one for each edge (i, j), we begin by computing the Hessian of a single term.

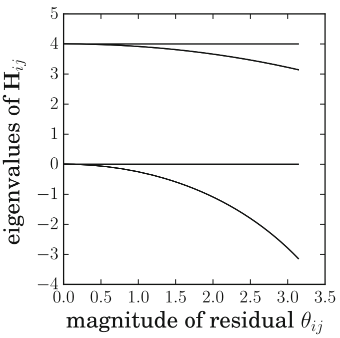

Fig. 2. The eigenvalues of the Hessian $\mathbf{H}_{ij}$. Two are constant: $\lambda_6 = 4$ and $\lambda_3 = 0$, and the other four appear in pairs.

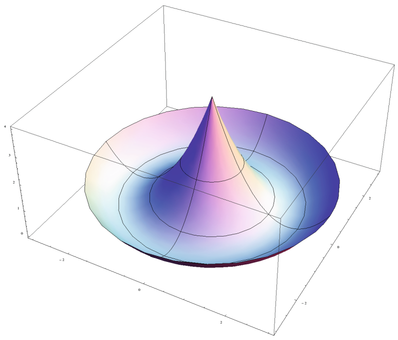

Fig. 3. A function which is not convex because of a rotational gauge ambiguity.

Theorem 1. *The Hessian matrix of $d_\angle(\widetilde{R}_{ij}, R_i R_j^\top)^2$, evaluated at the point (R_i, R_j), is given by*

$$\mathbf{H}_{ij} = \begin{bmatrix} \mu\mathbf{I} + (2-\mu)\mathbf{w}\mathbf{w}^\top & -\mu\mathbf{I} - (2-\mu)\mathbf{w}\mathbf{w}^\top - \theta[\mathbf{w}]_\times \\ -\mu\mathbf{I} - (2-\mu)\mathbf{w}\mathbf{w}^\top + \theta[\mathbf{w}]_\times & \mu\mathbf{I} + (2-\mu)\mathbf{w}\mathbf{w}^\top \end{bmatrix} \tag{7}$$

where the residual $R_i^\top \widetilde{R}_{ij} R_j$ is a rotation by angle $\theta \in [0, \pi)$ around axis $\mathbf{w}$, and where $\mu = \theta \cot(\theta/2)$.

We give a proof of Theorem 1 in Appendix A. Note that $\mathbf{H}_{ij}$ is a 6-by-6 real-valued symmetric matrix, and that ϕ^2 is not differentiable for $\theta = \pi$.

As has been observed in [14], $\mathbf{H}_{ij}$ is positive semidefinite when $d(\widetilde{\mathtt{R}}_{ij}, \mathtt{R}_i\mathtt{R}_j^\top) = 0$, and indefinite everywhere else. This can be seen in Fig. 2, because some of the eigenvalues of $\mathbf{H}_{ij}$ immediately become negative when moving away from the global minimum. We could well conclude at this point that ϕ^2 may be locally convex almost nowhere. However, this is not the case.

Some Gauge Intuition. In light of a previous result, the indefiniteness of $\mathbf{H}_{ij}$ is surprising. Hartley [8] has reported that $d_\angle(\mathtt{S}, \cdot)^2$ is locally convex almost everywhere. We also know that the gauge ambiguity can be freely fixed (for instance, by setting $\mathtt{R}_k = \mathtt{I}$ for some k) without altering the problem in any meaningful way. So our two-rotations problem has reduced to Hartley's:

$$\min_{\mathtt{R}_1, \mathtt{R}_2} d_\angle(\mathtt{S}, \mathtt{R}_1\mathtt{R}_2^\top)^2 \quad \text{s.t. } \mathtt{R}_2 = \mathtt{I} \qquad \equiv \qquad \min_{\mathtt{R}_1} d_\angle(\mathtt{S}, \mathtt{R}_1)^2 \tag{8}$$

Similarly, Fig. 3 shows a toy example of a simple polar optimization, chosen to have a rotational gauge ambiguity. This problem is locally convex on $\{(r, \theta) | 1 \le r < 2\}$, but it is not locally convex on $\{(r, \theta) | 0 < r < 1\}$. However, this distinction is only an artifact of the gauge. We see that the root problem, min $(r - 1)^2, 0 < r < 2$, is actually convex and very easy. A rotational gauge ambiguity can introduce spurious nonconvexity into a problem.

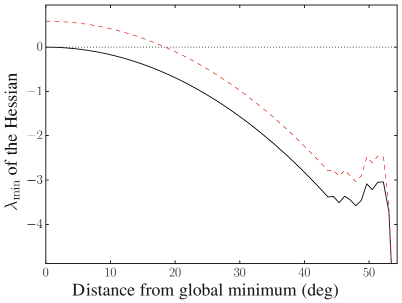

Fig. 4. Plots of $\lambda_{\min}$ for the original (solid) and gauge-fixed (dashed) problems along a ray of solutions.

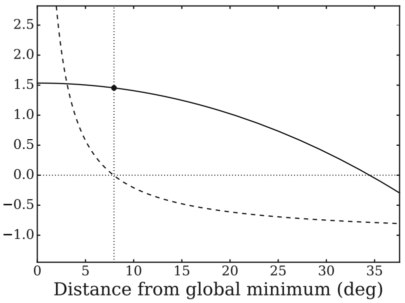

Fig. 5. A demonstration of Theorem 2. $\lambda_{\min}(\mathbf{H}^{\hat{k}})$ (solid line) is positive where $\lambda_{\min}(\mathbf{L}^{\hat{k}}_{G,\text{norm}})-1$ (dashed line) is positive.

Could it be that fixing the gauge ambiguity will reveal local convexity in general rotations averaging problems? Figure 4 shows the difference that fixing the gauge makes on a real problem. Both lines plot the smallest eigenvalue of the Hessian matrix along a 1D family of solutions, starting at a global minimum and moving away in a random direction. Notice that the fixed problem is now locally convex from the minimum to about 18° away. However, even with the gauge fixed, the problem is not locally convex everywhere, because the nonconvexity arises both from the gauge ambiguity and from the curvature of SO(3) (i.e., the cross product term in Eq. 20). The graph in Fig. 4 is an instance of the standard random graph $G_{n,p}$ with $n = 40$ and edge probability $p = 0.4$.

We are now ready to state our main result. (We give a proof in Appendix B.) By bounding the smallest eigenvalue of $\mathbf{H}$, while also restricting the gauge ambiguity by requiring $\mathtt{R}_k = \mathtt{I}$ (for some $k \in V$), we derive a sufficient condition for local convexity. We lose some precision by approximating away the directions $\mathbf{w}_{ij}$ of residuals in order to produce an interpretable result.

Theorem 2. *A rotations problem $(G, \widetilde{\mathcal{R}})$ is locally convex at solution $\mathcal{R}$ if for any $k \in V$ the smallest eigenvalue of a weighted, normalized graph Laplacian is large enough:*

$$\lambda_{min}\left(\mathbf{L}^{\hat{k}}_{\text{norm}}\right) > 1 \tag{9}$$

$$where \;\; \mathbf{L}_{\text{norm}} = \mathbf{D}(\theta_{ij})^{-1/2}\mathbf{L}(\mu_{ij})\mathbf{D}(\theta_{ij})^{-1/2} \tag{10}$$

and where θ_{ij} are the magnitudes of the residuals of this solution, where $\mu_{ij} = \theta_{ij}\cot(\theta_{ij}/2)$ is a convenience function of the residuals, and where $\mathbf{M}^{\hat{k}}$ is the matrix produced by removing row and column k from matrix $\mathbf{M}$.

We demonstrate Theorem 2 in Fig. 5. Note that the effect of fixing the gauge at different vertices varies: fixing a high degree node can reveal more local convexity than a vertex on the periphery of G. In our experiments, we always fix the gauge at a maximum degree vertex.

6 Consequences and Applications

In Theorem 2 we presented our main technical contribution: a sufficient condition for the local convexity of a rotations averaging problem $(G, \widetilde{\mathcal{R}})$ at a solution $\mathcal{R}$. In this section we explore consequences, both theoretical and practical, of Theorem 2. First we will look at what this theorem can tell us about what makes a rotations averaging problem difficult, and then we will look about how we can derive a way to quantify the difficulty of problems.

Interpretation of Theorem 2. Theorem 2 directly connects local convexity to the smallest eigenvalue of a weighted, normalized graph Laplacian with boundary conditions. It indicates that locally convex behavior depends on both the structure of the graph and the size of the residuals. Lower noise and a more connected graph structure interact with each other to produce easier problems.

To build some intuition about how this tradeoff works, consider complete graphs, which are maximally connected. In Fig. 6 we plot the quantity $\lambda_{\min}(\mathbf{L}^{\hat{k}}_{\text{norm}}) - 1$ over many sizes of complete graphs K_n, supposing a solution with identical residuals θ on every edge. Our theorem says that where this quantity is positive, problems are locally convex. Notice that more nodes yield less useful bounds, when all else is equal.

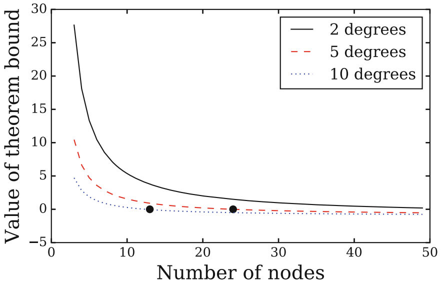

Fig. 6. Plots of the bound in Theorem 2 on the complete graph K_n, with identical residual error $\theta = \{2°, 5°, 10°\}$ on each edge. The zero crossings are indicated with black dots. With all else equal, more nodes yield lower (less useful) bounds.

Figure 7 shows a problem on which Theorem 2 can directly demonstrate local convexity. The problem is built on an instance of the random graph $G_{n,p}$ with $n = 10$, $p = 0.6$, and noise of 5° standard deviation added to each edge. Each line gives the value of $\lambda_{\min}(\mathbf{H}^{\hat{k}})$ along a path moving away from the global minimum. The circles mark where Theorem 2 transitions from guaranteeing local convexity to not applying. It appears that problem actually becomes locally non-convex around 32° from the minimum. An initial guess inside the locally convex region is good enough to find a global minimum.

Algebraic Connectivity as a Measurement of Hardness. In the previous section, we demonstrated how to use Theorem 2 to take a problem and a solution

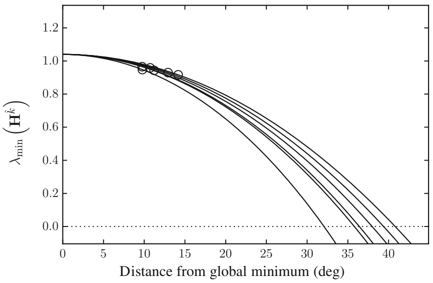

Fig. 7. A plot of $\lambda_{\min}(\mathbf{H}^{\hat{k}})$ along six random directions away from a global minimum. To the left of the circles on each line, Theorem 2 guarantees local convexity. To the right of them, the theorem fails to apply.

and certify if the problem is locally convex there. However, this is unlikely to be a useful operation for solving problems in practice. The greater utility of Theorem 2 is the insight it provides: It describes the way graph structure and noise in the problem interact to make a problem difficult.

Now we will take this a step further. When considering an unsolved problem, it is unclear quite how noisy it is. Similarly, when collecting data to form a problem instance, the noisiness of the measurements may not be easy to control. Can we understand the contribution of graph structure alone to problem difficulty?

In the following Theorem, we relax Theorem 2 to get an easily interpretable (although less precise) bound which separates the dependencies on noise and graph structure:

Theorem 3. *A rotations averaging problem $(G, \widetilde{\mathcal{R}})$ is locally convex if*

$$\frac{\lambda_2(\mathbf{L})}{n} > \frac{\Delta(\theta_{ij})}{\mu(\theta_{max})} \tag{11}$$

where $\lambda_2(\mathbf{L})$ is the algebraic connectivity of the graph G, $\mu(\theta_{max})$ is the μ convenience function applied the the largest residual, and $\Delta(\theta_{ij})$ is the maximum degree in the graph G with weights θ_{ij}.

Proof. From the proof of Theorem 2, we have a constraint that is sufficient for local convexity:

$$\lambda_{\min}\left(\mathbf{L}_G(\mu_{ij}) - \mathbf{D}_G(\theta_{ij}).\right) > 0 \tag{12}$$

Now recalling that $\mathbf{D}_G(\theta_{ij})$ is a non-negative diagonal matrix,

$$\Longleftarrow \quad \lambda_{\min}\left(\mu(\theta_{\max})\mathbf{L} - \mathbf{D}(\theta_{ij})\right) > 0 \tag{13}$$

$$\Longleftarrow \lambda_{\min}\left(\mu(\theta_{\max})\mathbf{L}\right) > \lambda_{\max}\left(\mathbf{D}(\theta_{ij})\right) \tag{14}$$

$$\Longleftarrow \frac{1}{n}\lambda_2(\mathbf{L}) > \frac{\Delta(\theta_{ij})}{\mu(\theta_{\max})} \tag{15}$$

where the last implication follows by considering the eigensystem of $\mathbf{L}$. The maximum projection of any vector whose kth element is 0 onto $[1, 1, \ldots, 1]$ is $\sqrt{(n-1)/n}$, so the projection onto other eigenvectors of $\mathbf{L}$ must be at least $1/\sqrt{n}$. We conclude that $\mathbf{x}^\top \mathbf{L}^{\hat{k}} \mathbf{x} \geq \lambda_2(\mathbf{L})/n$. This harmonic bound is the least precise approximation in this paper. □

To arrive at the separation of graph structure and noise in Theorem 3 we necessarily made many approximations that reduce the precision of the result. In fact, Theorem 3 will only be directly applicable on unrealistic problems with very low noise. However, its value is in the insight that it brings.

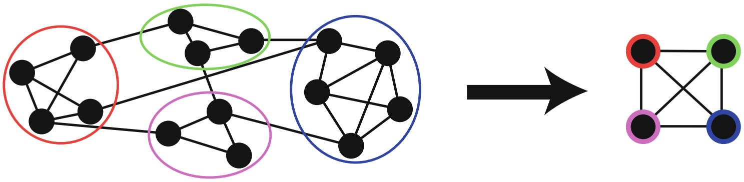

Fig. 8. A cartoon depiction of first solving easy, well-connected subproblems to simplify a harder problem.

We propose using $\lambda_2(G)/n$, the graph structure term that appears in Theorem 3, as an indicator to distinguish easy problems from harder ones. To demonstrate that this is effective, consider this indicator computed on each of the 1DSfM datasets [1] in Fig. 9. These are challenging, real-world Structure from Motion datasets compiled from Internet community photo collections. We plot $\lambda_2(G)/n$ on the x-axis. To estimate the true difficulty of a problem, we initialize with a linear method [9] and then minimize ϕ^2 with the Ceres non-linear least squares solver [23]. The average (gauge-aligned) distance between our solution and the reference solution provided with the datasets is our error, plotted on the y-axis. As claimed, we see that in general problems with higher indicator can be solved with lower error.

Applications to Solver Design. Our results indicate that smaller, well-connected graphs are generally easier than larger and less noisy ones. How can our results inform the design of the next generation of solvers? Figure 8 is a cartoon where the original problem is very poorly-connected and probably rather hard. However, if we first solve a set of easy, well-connected subproblems, then we can reduce the hard problem to a series of easy ones. We demonstrate this in Fig. 9, where by splitting a dataset Union Square into two pieces using normalized spectral clustering [6], both pieces can be solved more accurately. Notice that both pieces have a better (larger) hardness score than the original problem.

Approaches based on solving a sequence of increasingly large subproblems have been used elsewhere in Structure from Motion [24,25], although not for rotations averaging. Rather than resorting to empirical heuristics, Theorem 3

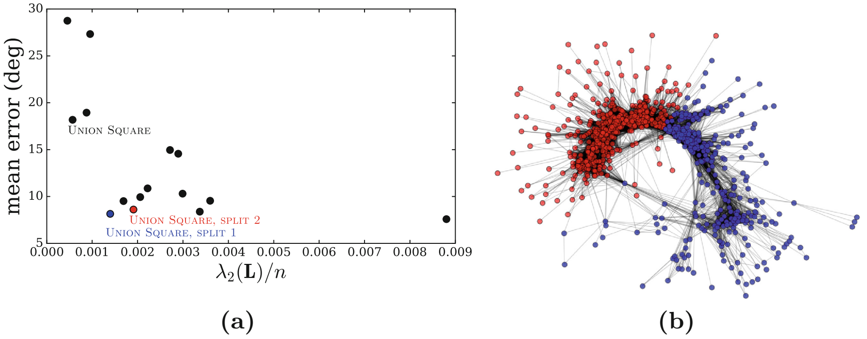

Fig. 9. (a) The structure term from Theorem 3 (x-axis) can serve as an observable estimate of structural difficulty (higher is less difficult). Average error after running a solver (y-axis) plotted against this indicator. Notice that problems with a larger calculated indicator tend to yield better solutions. (b) Normalized spectral clustering splits the Union Square problem into two smaller, easier problems (with corresponding dots shown in (a)).

gives a principled score of the difficulty of a subproblem, which could be used to guide an algorithm.

7 Summary and Conclusions

Future Work. Two weaknesses of these theorems—the choice of vertex in fixing the gauge, and the lossy harmonic approximation—may be closely related. While fixing a vertex is quite simple, the existence of the choice of node suggests that in some sense this approach is not the most naturally suited to the problem. A more sophisticated understanding of the class of gauge-fixing constraints, including distributed constraints, may be able to greatly improve upon Theorems 2 and 3.

Conclusion. Rotations averaging has become a key subroutine in many global Structure from Motion methods. It is known to be nonconvex, yet many solvers perform reasonably well in practice. Global convergence results for nonlinear optimization problems like this one are rare and hard to achieve. We do something more tractable, using local convexity to show global convergence on a (hopefully large) subdomain of reasonable solutions. We give sufficient but not necessary conditions. Our analysis locates the root sources of non-convexity: both the gauge ambiguity, and the curvature of the rotations group. The extent of local convexity depends on the interaction of structure and noise, as captured in a particular graph Laplacian. We also approximate the contribution of graph structure alone to problem difficulty, which can be used to estimate the difficulty of a problem instance and can lead to new algorithms that subdivide problems based on this measure. This deeper understanding of the structure and challenges of rotations averaging can inform the construction of an ever-more reliable next generation of solvers.

Acknowledgements. This work was funded in part by the National Science Foundation (IIS-1149393) and by a grant from Google. We also thank the anonymous reviewers and Daniel Miller for their valuable help in catching errors.

Appendix A: Proof of Theorem 1

We will calculate derivatives of $d(\widetilde{\mathtt{R}}_{ij}, \mathtt{R}_i\mathtt{R}_j^\top)^2$ at $(\mathtt{R}_i, \mathtt{R}_j)$ by introducing perturbations $\mathbf{x}_i$ and $\mathbf{x}_j$ in the respective tangent spaces at $\mathtt{R}_i$ and $\mathtt{R}_j$.

$$\begin{aligned}
\mathbf{H}_{ij} &= D^2 d(\widetilde{\mathtt{R}}_{ij}, \exp_{\mathtt{R}_i}[\mathbf{x}_i]_\times \exp_{\mathtt{R}_j}[\mathbf{x}_j]_\times^\top)^2 &(16)\\
&= D^2 d(\widetilde{\mathtt{R}}_{ij}, \mathtt{R}_i \exp[\mathbf{x}_i]_\times \exp[\mathbf{x}_j]_\times^\top \mathtt{R}_j^\top)^2 &(17)\\
&= D^2 d(\mathtt{R}_i^\top \widetilde{\mathtt{R}}_{ij}\mathtt{R}_j, \exp[\mathbf{x}_i]_\times \exp[-\mathbf{x}_j]_\times)^2 &(18)\\
&= D^2 d(\mathtt{R}_i^\top \widetilde{\mathtt{R}}_{ij}\mathtt{R}_j, \exp BCH([\mathbf{x}_i]_\times, [-\mathbf{x}_j]_\times))^2 &(19)
\end{aligned}$$

We refer the reader to [26] for an explanation of the Baker-Campbell-Hausdorff formula, which relates the product $\exp A \exp B$ to $\exp BCH(A, B)$. Although the BCH itself is quite messy, we can approximate it. Since we are computing second derivatives, we go to second order:

$$\mathbf{H}_{ij} \approx D^2 d(\mathtt{R}_i^\top \widetilde{\mathtt{R}}_{ij}\mathtt{R}_j, \exp([\mathbf{x}_i]_\times - [\mathbf{x}_j]_\times - [\mathbf{x}_i \times \mathbf{x}_j]_\times))^2 \qquad (20)$$

Note the cross product term $\mathbf{x}_i \times \mathbf{x}_j$. This is the main source of nonconvexity in rotations averaging, and arises from the twistedness of the space SO(3).

In [8], Hartley et al. compute the simpler case of $d(\mathtt{S}, \exp[\cdot]_\times)^2$ (where $\mathtt{S}$ is some constant) and conclude that

$$D\, d(\mathtt{S}, \exp[\mathbf{x}]_\times)^2 = -\mathbf{w} \quad \text{and} \quad D^2 d(\mathtt{S}, \exp[\mathbf{x}]_\times)^2 = \mu\mathbf{I} + (2-\mu)\mathbf{w}\mathbf{w}^\top \qquad (21)$$

where $\mathtt{S} = \exp[\theta\mathbf{w}]_\times$ and and $\mu = \theta \cot \frac{\theta}{2}$.

Our problem has now been reduced to computing the derivatives of a composition of functions, both of which themselves have known derivatives. The conclusion follows after repeated application of the chain rule. Refer the supplementary materials for a supporting Mathematica notebook. □

Appendix B: Proof of Theorem 2

We wish to find a constraint that is sufficient for $\mathbf{H}$ positive definite. In order to do this, we will have to fix the gauge of the problem by requiring $\mathtt{R}_k = \mathtt{I}$. Our approach will be to uncover what structure we can find within $\mathbf{H}$, and then return to the matter of the gauge. We use $A \succeq B$ for symmetric matrices A and B to mean that $A - B$ is positive semi-definite, and $\mathbf{A} \otimes \mathbf{B}$ for the Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$. We begin by analyzing the Hessian for a single residual block $\mathbf{H}_{ij}$:

$$\mathbf{H}_{ij} = (2-\mu_{ij}) \begin{bmatrix} \mathbf{w}_{ij}\mathbf{w}_{ij}^\top & -\mathbf{w}_{ij}\mathbf{w}_{ij}^\top \\ -\mathbf{w}_{ij}\mathbf{w}_{ij}^\top & \mathbf{w}_{ij}\mathbf{w}_{ij}^\top \end{bmatrix} + \mu_{ij} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} + \theta_{ij} \begin{bmatrix} \mathbf{0} & -[\mathbf{w}_{ij}]_\times \\ [\mathbf{w}_{ij}]_\times & \mathbf{0} \end{bmatrix} \tag{22}$$

$$\succeq \mu_{ij} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} + \theta_{ij} \begin{bmatrix} \mathbf{0} & -[\mathbf{w}_{ij}]_\times \\ [\mathbf{w}_{ij}]_\times & \mathbf{0} \end{bmatrix} \tag{23}$$

$$\succeq \mu_{ij} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} - \theta_{ij} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{24}$$

$$= \left(\begin{bmatrix} \mu_{ij} & -\mu_{ij} \\ -\mu_{ij} & \mu_{ij} \end{bmatrix} - \begin{bmatrix} \theta_{ij} & 0 \\ 0 & \theta_{ij} \end{bmatrix} \right) \otimes \mathbf{I}_3 \tag{25}$$

where we began with Theorem 1 and approximated away all of the terms which depended on the directions $\mathbf{w}_{ij}$ of the residuals, by first dropping a positive semi-definite term, and then bounding a skew matrix from below. We now analyze the full Hessian:

$$\mathbf{H} = \sum_{(i,j)\in E} \mathbf{H}_{ij} \tag{26}$$

$$\succeq \sum_{(i,j)\in E} \left(\begin{bmatrix} \mu_{ij} & -\mu_{ij} \\ -\mu_{ij} & \mu_{ij} \end{bmatrix} - \begin{bmatrix} \theta_{ij} & 0 \\ 0 & \theta_{ij} \end{bmatrix} \right) \otimes \mathbf{I}_3 \tag{27}$$

$$= \left(\sum_{(i,j)\in E} \begin{bmatrix} \mu_{ij} & -\mu_{ij} \\ -\mu_{ij} & \mu_{ij} \end{bmatrix} - \begin{bmatrix} \theta_{ij} & 0 \\ 0 & \theta_{ij} \end{bmatrix} \right) \otimes \mathbf{I}_3 \tag{28}$$

$$= (\mathbf{L}(\mu_{ij}) - \mathbf{D}(\theta_{ij})) \otimes \mathbf{I}_3 \tag{29}$$

To fix the gauge, we need to remove the rows and columns of $\mathbf{H}$ pertaining to vertex k. Since we can view this as projecting onto a subspace, by the Cauchy interlacing theorem this operation preserves the matrix inequality:

$$\mathbf{H}^{\hat{k}} \succeq (\mathbf{L}(\mu_{ij}) - \mathbf{D}(\theta_{ij}))^{\hat{k}} \otimes \mathbf{I}_3 \tag{30}$$

Since a Kronecker product with the identity only alters the multiplicity of the eigenvalues, a sufficient constraint for $\mathbf{H}^{\hat{k}} \succ \mathbf{0}$ is

$$\mathbf{L}(\mu_{ij})^{\hat{k}} - \mathbf{D}(\theta_{ij})^{\hat{k}} \succ \mathbf{0} \tag{31}$$

$$\iff \left(\mathbf{D}(\theta_{ij})^{-1/2}\mathbf{L}(\mu_{ij})\mathbf{D}(\theta_{ij})^{-1/2}\right)^{\hat{k}} - \mathbf{I}_{n-1} \succ 0 \tag{32}$$

$$\iff \mathbf{L}_{\text{norm}}^{\hat{k}} - \mathbf{I}_{n-1} \succ 0 \tag{33}$$

$$\iff \lambda_{\min}(\mathbf{L}_{\text{norm}}^{\hat{k}}) > 1 \tag{34}$$

We call $\mathbf{L}_{\text{norm}}^{\hat{k}}$ a weighted, normalized graph Laplacian with a boundary condition. □

References

1. Wilson, K., Snavely, N.: Robust global translations with 1DSfM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 61–75. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10578-9_5
2. Ozyesil, O., Singer, A.: Robust camera location estimation by convex programming. In: CVPR (2015)
3. Sweeney, C., Sattler, T., Hollerer, T., Turk, M., Pollefeys, M.: Optimizing the viewing graph for structure-from-motion. In: ICCV, December 2015
4. Cui, Z., Tan, P.: Global structure-from-motion by similarity averaging. In: ICCV (2015)
5. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.: Discrete-continuous optimization for large-scale structure from motion. In: CVPR (2011)
6. Von Luxburg, U.: A tutorial on spectral clustering. Stat. comput. **17**(4), 395–416 (2007)
7. Govindu, V.M.: Combining two-view constraints for motion estimation. In: CVPR (2001)
8. Hartley, R., Trumpf, J., Dai, Y., Li, H.: Rotation averaging. In: IJCV (2013)
9. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: CVPR (2007)
10. Arrigoni, F., Magri, L., Rossi, B., Fragneto, P., Fusiello, A.: Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In: 3DV (2014)
11. Wang, L., Singer, A.: Exact and stable recovery of rotations for robust synchronization. Inf. Infer. **2**(2), 145–193 (2013)
12. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: CVPR (2004)
13. Tron, R., Vidal, R., Terzis, A.: Distributed pose averaging in camera networks via consensus in SE(3). In: ICDSC (2008)
14. Tron, R., Afsari, B., Vidal, R.: Intrinsic consensus on $SO(3)$ with almost-global convergence. In: CDC (2012)
15. Hartley, R., Aftab, K., Trumpf, J.: L_1 rotation averaging using the Weiszfeld algorithm. In: CVPR (2011)
16. Chatterjee, A., Govindu, V.M.: Efficient and robust large-scale rotation averaging. In: ICCV (2013)
17. Fredriksson, J., Olsson, C.: Simultaneous multiple rotation averaging using lagrangian duality. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012. LNCS, vol. 7726, pp. 245–258. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37431-9_19
18. Bandeira, A.S., Singer, A., Spielman, D.A.: A Cheeger inequality for the graph connection Laplacian. SIAM J. Matrix Anal. Appl. **34**(4), 1611–1630 (2013)
19. Boumal, N., Singer, A., Absil, P.A.: Robust estimation of rotations from relative measurements by maximum likelihood. In: Decision and Control. IEEE (2013)
20. Boumal, N., Singer, A., Absil, P.A., Blondel, V.D.: Cramér-Rao bounds for synchronization of rotations. Inf. Infer. **3**(1), 1–39 (2014)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI **22**(8), 888–905 (2000)
22. Golub, G.H., Loan, C.F.V.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2012)
23. Agarwal, S., Mierle, K., et al.: Ceres solver. http://ceres-solver.org

24. Steedly, D., Essa, I., Delleart, F.: Spectral partitioning for structure from motion. In: ICCV (2003)
25. Toldo, R., Gherardi, R., Farenzena, M., Fusiello, A.: Hierarchical structure-and-motion recovery from uncalibrated images. Comput. Vis. Image Underst. **140**, 127–143 (2015)
26. Kanatani, K.: Group-Theoretical Methods in Image Understanding. Springer, Heidelberg (1990)