It Is All In The Weights: Robust Rotation Averaging Revisited

Chitturi Sidhartha
Indian Institute of Science
Bengaluru, India-560012
chitturis@iisc.ac.in

Venu Madhav Govindu
Indian Institute of Science
Bengaluru, India-560012
venug@iisc.ac.in

Abstract

Rotation averaging is the problem of recovering 3D camera rotations from a number of pairwise relative rotation estimates. The state-of-the-art method of [5] involves robust averaging in the Lie-algebra of 3D rotations using an $\ell_{\frac{1}{2}}$ loss function which is carried out using an iteratively reweighted least squares (IRLS) minimization. In this paper we argue that the performance of IRLS-based rotation averaging is intimately connected with two factors: a) the nature of the robust loss function used; and b) the initialization. We make two contributions. Firstly, we analyse the pitfalls associated with the unbounded weights in IRLS minimization of $\ell_p (0 < p < 2)$ loss functions in the context of rotation averaging. We elucidate the design choices and modifications implicit to the state-of-the-art method of [5] that overcomes these problems. Secondly, we argue that the $\ell_{\frac{1}{2}}$-based IRLS method is inflexible in adapting to the specific noise characteristics of individual datasets, leading to poorer performance. We remedy this limitation by means of a Geman-McClure loss function embedded in a graduated optimization framework. We present results on a number of large-scale real-world datasets to demonstrate that our proposed method outperforms state-of-the-art methods in terms of both efficiency and accuracy.

1. Introduction

Global methods are desirable for solving large-scale structure-from-motion (SfM) problems owing to their greater speed and completeness in reconstruction compared to incremental methods [27, 32, 25]. Rotation Averaging (RA) plays an important role in such global methods by providing a good initialization to Bundle-adjustment [28], a highly non-linear, non-convex optimization problem, which is the backbone of any SfM solution.

Consider a viewgraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $N = |\mathcal{V}|$ vertices and $M = |\mathcal{E}|$ edges, where vertices represent camera rotations and edges represent relative rotations between their corresponding vertices. Given relative rotation estimates $\{\boldsymbol{R}_{ij}\}_{(i,j)\in\mathcal{E}}$ for each edge $(i, j) \in \mathcal{E}$, the problem of rotation averaging is to recover the global camera rotations, *i.e*. the rotations $\boldsymbol{R}_{\mathcal{V}} = \{\boldsymbol{R}_1, \cdots, \boldsymbol{R}_N\}$ where $\boldsymbol{R}_i$ is the estimated 3D rotation of camera $i \in \mathcal{V}$ (w.r.t a global frame of reference). Rotation averaging uses the observation that relates relative rotations (edges) to absolute camera rotations (vertices), *i.e*. $\boldsymbol{R}_{ij} = \boldsymbol{R}_j \boldsymbol{R}_i^{-1}$. All real-world datasets tend to have some corrupted $\boldsymbol{R}_{ij}$'s due to a number of reasons including incorrect feature matches, occlusions, repeated structures in scenes, symmetries etc. Robustness to such outliers is a key consideration for all rotation averaging methods. The most commonly used rotation averaging method is [5] that uses an $\ell_{\frac{1}{2}}$ loss function for robustness and optimizes using Iteratively Reweighted Least Squares (IRLS). While the $\ell_{\frac{1}{2}}$ based IRLS method of [5] achieves state-of-the-art in speed and accuracy, in this paper, we argue that:

1. All ℓ_p ($0<p<2$) loss functions suffer from the drawback of unbounded weights in IRLS. While this was implicitly addressed in [5] by clamping large weights, as we will discuss in Sec. 4, such clamping impacts the quality of the estimates. Moreover, the inability of ℓ_p loss functions to adapt to the varying noise characteristics leads to reduced performance.

2. The limitations of the $\ell_{\frac{1}{2}}$-based method of [5] can be overcome by a graduated optimization method using the Geman-McClure loss function. We show in Sec. 6 that our proposed graduated method can outperform the $\ell_{\frac{1}{2}}$-based method of [5] and other state-of-the-art approaches both in terms of speed and accuracy.

2. Related Work

The body of work on rotation averaging is fairly extensive and in this section we only discuss papers of relevance to our approach. 3D rotations form a Lie group known as the Special Orthogonal group $\boldsymbol{R} \in \mathbb{SO}(3)$. We can classify methods based on whether they solve the rotation averaging problem while respecting the geometry of $\mathbb{SO}(3)$ (*intrinsic*

methods) or whether they relax the geometric constraints and project onto $\mathbb{SO}(3)$ after optimization (typically) a convex form (*extrinsic methods*).

Intrinsic Methods: Govindu [13] exploits the Lie-group structure of $\mathbb{SO}(3)$ for the first time in RA. In [14], robustness is embedded by taking a random sampling consensus (RANSAC) approach for removal of outliers in the pairwise relative motion estimates. Hartley *et al*. [16] solve an ℓ_1 cost of RA with an intrinsic metric by iteratively finding the geometric median on the tangent spaces of $\mathbb{SO}(3)$ using the Weiszfeld algorithm. Chatterjee and Govindu [4] propose to solve a robust cost of RA using IRLS technique on the Lie algebra (tangent space) of $\mathbb{SO}(3)$. They optimize an ℓ_1 based cost to serve as an initialization to IRLS. In [5], an $\ell_{\frac{1}{2}}$-based IRLS optimization is proposed as the best empirical choice of loss function. Shi and Lerman [26] propose a message passing algorithm to estimate the noise levels and detect potential outliers in the relative rotation estimates and use it in conjunction with the residuals in the IRLS technique proposed in [4].

Extrinsic Methods: Govindu [12] formulate RA as a linear least squares problem based on the quaternion representation of rotations. Martinec and Pajdla [21] perform linear least squares by relaxing the orthogonality constraints of the rotations, then reproject them back onto the space of rotations. Crandall *et al*. [6] model constraints between pairs of cameras as a Markov Random field (MRF) and use a combination of discrete and continuous optimization to solve for poses. Fredriksson and Olsson [11] use Lagrangian duality to give a certificate for global optimality to the problem considered in [12] in a moderate noise scenario. Eriksson *et al*. [10] use duality and spectral graph theory to obtain certifiably global solutions to RA with the chordal metric. Wang *et al*. [30] use a robust ℓ_1 cost with chordal distance metric, derive a convex relaxation of the problem and solve it with an alternating direction method. Arrigoni *et al*. [1] formulate RA as a low-rank and sparse matrix decomposition problem. Dellaert *et al*. [7] use a semidefinite relaxation of a least squares cost function with chordal metric. They solve the relaxed problem on higher dimensional rotation groups $\mathbb{SO}(p)$ starting from $p = 3$ and increase the value of p until convergence.

Robust cost Optimization: We attempt only to give a glimpse of the vast literature on the optimization of robust cost functions of the form $\sum_{i=1}^{n} \rho\left(x_i(\boldsymbol{\theta})\right)$ [19] which are of relevance in our work. We consider three types of methods: *Direct*, *Lifting*, *Graduated optimization*. Direct methods such as IRLS [19, 15], Triggs correction [28], square-rooting of kernel [9] operate explicitly on the cost function, make a quadratic approximation and reduce it to a non-linear least squares problem, then solve it using Gauss-Newton or Levenberg-Marquardt method. These methods are prone to settling at poor local minima. Lifting based approaches [38, 34, 35] modify the actual cost function by adding auxiliary variables called weights (ϕ_i) which signify the inlier probability for each observation x_i. The optimization is carried out jointly over the actual variables ($\boldsymbol{\theta}$) and the lifting variables (ϕ_i's). It is similar to a line-process formulation which is shown to be equivalent to the actual robust cost function [3]. But the lifting based methods are sensitive to the initialization of ϕ_i's [36, 37]. Graduated optimization methods [36] solve a series of objective functions which are successively better (smoother) approximations of the original cost function and hence easier to optimize. Graduated non-convexity [33, 23], annealing [24, 29, 2], homotopy optimization methods [8], continuation methods [22] fall under this category. Depending on the cost function, these methods generally reach a good local minima, if not the global minima.

Name	$\rho(x)$	$\psi(x)$	$\phi(x)$
ℓ_2	$x^2/2$	x	1
ℓ_p	$\lvert x\rvert^p/p$	$\text{sign}(x)\lvert x\rvert^{p-1}$	$\lvert x\rvert^{p-2}$
Geman-McClure	$\frac{\sigma^2 x^2}{2(x^2+\sigma^2)}$	$\frac{\sigma^4 x}{(\sigma^2+x^2)^2}$	$\frac{\sigma^4}{(\sigma^2+x^2)^2}$

Table 1: Loss functions

3. Mathematical Background

In this section we provide a basic statement of the mathematical concepts that will be used in the remainder of the paper, *i.e*. the IRLS method, properties of 3D rotations, statement of the rotation averaging problem and the optimization step in [5]. The notation presented in this section will be used throughout the paper.

3.1. Iteratively Reweighted Least Squares (IRLS)

Consider an overdetermined system of m linear equations $\boldsymbol{a}_k^\top \boldsymbol{l} = c_k$ where $k = 1, \ldots, m$, where the unknowns to be estimated are $\boldsymbol{l} \in \mathbb{R}^n$. For an overdetermined system of equations, $m > n$. Consider the following cost function:

$$F = \sum_{k=1}^{m} \rho(|\boldsymbol{a}_k^\top \boldsymbol{l} - c_k|) \tag{1}$$

where $\rho(\cdot)$ is a robust loss function chosen to mitigate the effect of noise and outliers. $|\boldsymbol{a}_k^\top \boldsymbol{l} - c_k|$ are referred to as residuals. We wish to minimize the cost F. Assuming $|\boldsymbol{a}_k^\top \boldsymbol{l} - c_k| \neq 0 \; \forall \; k$, taking the gradient we obtain,

$$\nabla F = \sum_{k=1}^{m} \rho'(|\boldsymbol{a}_k^\top \boldsymbol{l} - c_k|) \boldsymbol{a}_k \frac{\boldsymbol{a}_k^\top \boldsymbol{l} - c_k}{|\boldsymbol{a}_k^\top \boldsymbol{l} - c_k|} \tag{2}$$

The *influence* and the *weight* functions are defined as $\psi(x) = \rho'(x)$ and $\phi(x) = \frac{\psi(x)}{x}$ respectively. Table 1

shows the corresponding functions of relevance to this paper. Thus,

$$\nabla F = \sum_{k=1}^{m} \phi(|\boldsymbol{a}_k^\top \boldsymbol{l} - c_k|)\boldsymbol{a}_k(\boldsymbol{a}_k^\top \boldsymbol{l} - c_k) \tag{3}$$

Treating $\phi(|\boldsymbol{a}_k^\top \boldsymbol{l} - c_k|) = \phi_k$ as a constant, solving $\nabla F = 0$ is equivalent to solving a weighted least squares problem given by,

$$\underset{\boldsymbol{l}}{\operatorname{argmin}} \sum_{k=1}^{m} \phi_k |\boldsymbol{a}_k^\top \boldsymbol{l} - c_k|^2 \tag{4}$$

If $\boldsymbol{A}$ is a matrix obtained by stacking $\boldsymbol{a}_k^\top$ as rows, Φ is a diagonal matrix with ϕ_k as its diagonal entries, $\boldsymbol{c} = [c_1, \ldots, c_m]^\top$, then the solution to Eqn. 4 is given by,

$$\boldsymbol{l} = \left(\boldsymbol{A}^\top \Phi \boldsymbol{A}\right)^{-1} \boldsymbol{A}^\top \Phi \boldsymbol{c} \tag{5}$$

We note here that for the problem of rotation averaging, the matrix $\boldsymbol{A}$ is the incidence matrix of the viewgraph $\mathcal{G}$ and $\boldsymbol{A}^\top \Phi \boldsymbol{A}$ is the weighted Laplacian.

The standard IRLS [18] technique can be summarized as follows: in every iteration, ϕ_k's are computed with a current estimate of $\boldsymbol{l}$ and the weighted least squares problem in Eqn. 4 is solved to obtain a new estimate of $\boldsymbol{l}$, and this is repeated until convergence (*i.e.* $|\Delta \boldsymbol{l}| < \epsilon$). We refer the reader to Table 1 in [5] for a comprehensive list of robust loss functions. In this paper, we deal only with l_p ($0<p<2$) and the Geman-McClure (GM) loss functions (Table 1). Henceforth, whenever ℓ_p is used, it is assumed that $0 < p < 2$, unless stated otherwise.

3.2. Representation of 3D Rotations

Every 3×3 rotation matrix $\boldsymbol{R}$ satisfies the constraints $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}_3$ and $det(\boldsymbol{R}) = +1$ and all such rotations form a Lie group, *i.e.* the Special Orthogonal group $\mathbb{SO}(3)$. The local neighbourhood of an element $\boldsymbol{R} \in \mathbb{SO}(3)$ is described by its Lie algebra, *i.e.* the skew-symmetric form $[\boldsymbol{\omega}]_\times \in \mathfrak{so}(3)$ where $\boldsymbol{\omega}$ is the axis-angle representation of rotation $\boldsymbol{R}$. Elements in the Lie group $(\boldsymbol{R})$ and their corresponding Lie algebra $[\boldsymbol{\omega}]_\times$ are related as

$$\boldsymbol{R} = exp([\boldsymbol{\omega}]_\times), \quad [\boldsymbol{\omega}]_\times = log(\boldsymbol{R}) \tag{6}$$

A summary of various representations of 3D rotations can be found in Section 3 in [17] and more details on the group properties of rotations can be found in [13, 20]. The *intrinsic* or geodesic distance between two 3D rotations $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ is given as $d(\boldsymbol{R}_1, \boldsymbol{R}_2) = \frac{1}{\sqrt{2}}\|log(\boldsymbol{R}_1\boldsymbol{R}_2^{-1})\|_F$, whereas *extrinsic* methods typically use the chordal distance $d_{chordal}(\boldsymbol{R}_1, \boldsymbol{R}_2) = \|\boldsymbol{R}_1 - \boldsymbol{R}_2\|_F$. In this work we only use the intrinsic distance. Henceforth we refer to the axis-angle representation of $\boldsymbol{R}$ as $\boldsymbol{\omega}(\boldsymbol{R})$ and the rotation matrix representation of $\boldsymbol{\omega}$ as $\boldsymbol{R}(\boldsymbol{\omega})$.

3.3. Rotation Averaging

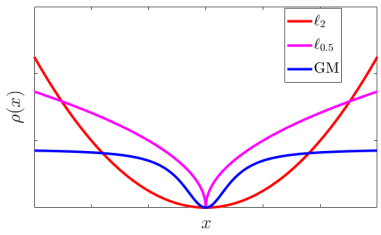
As stated in Sec. 1, rotation averaging methods use all available relative rotations $\{\boldsymbol{R}_{ij}\}$ between pairs of cameras to solve for absolute camera rotations for all cameras, *i.e.* $\boldsymbol{R}_\mathcal{V} = \{\boldsymbol{R}_1, \cdots, \boldsymbol{R}_N\}$. In the presence of noise or outliers, the relationship $\boldsymbol{R}_{ij} \neq \boldsymbol{R}_j\boldsymbol{R}_i^{-1}$. Hence the problem of rotation averaging is one of finding the best possible fit or explanation of all the observed $\{\boldsymbol{R}_{ij}\}$ in terms of the camera rotations $\boldsymbol{R}_\mathcal{V} = \{\boldsymbol{R}_1, \cdots, \boldsymbol{R}_N\}$. We can use the intrinsic distance metric on $\mathbb{SO}(3)$ stated in Sec. 3.2 to evaluate the quality of the fit for each individual edge. By taking all edge relationships and accounting for the presence of noise and outliers using a loss function $\rho(\cdot)$, we define rotation averaging as the problem of optimizing the cost function

$$\boldsymbol{R}_\mathcal{V} = \underset{\{\boldsymbol{R}_1, \boldsymbol{R}_2, \cdots, \boldsymbol{R}_N\}}{\operatorname{argmin}} \sum_{(i,j)\in\mathcal{E}} \rho\left(d\left(\boldsymbol{R}_{ij}, \boldsymbol{R}_j\boldsymbol{R}_i^{-1}\right)\right) \tag{7}$$

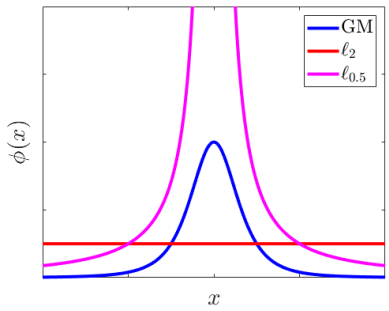
The algorithm to solve the rotation averaging problem of (7) can be briefly sketched as follows: At any iteration we evaluate the residual or difference in fitting between observed relative rotations $\boldsymbol{R}_{ij}$ and the current estimate $\boldsymbol{R}_j\boldsymbol{R}_i^{-1}$, *i.e.* $\boldsymbol{R}_j^{-1}\boldsymbol{R}_{ij}\boldsymbol{R}_i$. We collect their axis-angle forms (for all edges in $\mathcal{E}$) into a single vector denoted $\Delta\boldsymbol{\Omega}_\mathcal{E}$. Further, we denote the angle updates for each camera $i \in \mathcal{V}$ as $\Delta\boldsymbol{\omega}_i$ (or $\boldsymbol{\omega}(\boldsymbol{\Delta R}_i)$) and collect all of them into an update vector $\Delta\boldsymbol{\Omega}_\mathcal{V}$ *i.e.* $\Delta\boldsymbol{\Omega}_\mathcal{V} = [\Delta\boldsymbol{\omega}_1^T, \cdots, \Delta\boldsymbol{\omega}_N^T]$. Thus the averaging problem becomes one of updating all camera rotations $\boldsymbol{R}_i$ by solving the weighted least squares problem $\left(\mathbf{A}^T\Phi\mathbf{A}\right)\Delta\boldsymbol{\Omega}_\mathcal{V} = \mathbf{A}^T\Phi\Delta\boldsymbol{\Omega}_\mathcal{E}$ where Φ is the diagonal matrix of weights which is dependent on $\rho(\cdot)$ and the residuals. We update $\boldsymbol{R}_i \leftarrow \boldsymbol{R}_i \exp([\Delta\boldsymbol{\omega}_i]_\times)$ where $\Delta\boldsymbol{\omega}_i$ is obtained from $\Delta\boldsymbol{\Omega}_\mathcal{V}$. This is repeated till convergence. The reader is referred to [13, 4, 5] for more details. After an extensive evaluation, [5] chose the loss function $\rho(x) = 2|x|^{\frac{1}{2}}$ (denoted $\ell_{\frac{1}{2}}$) for its accuracy on real-world datasets. In Sec. 4 we discuss the limitations of the $\ell_{\frac{1}{2}}$ based approach of [5] and present our proposed solution that overcomes these limitations. In the remainder of this section we present a few preliminaries to keep our discussion self-contained.

Datasets: We demonstrate our findings on the commonly used large-scale 1DSfM datasets [31] taken from `http://www.cs.cornell.edu/projects/1dsfm/` whereas the San Francisco and Artsquad datasets of [6] are taken from `http://www.ee.iisc.ac.in/labs/cvl/research/rotaveraging/` and we follow the same abbreviations as in [5].

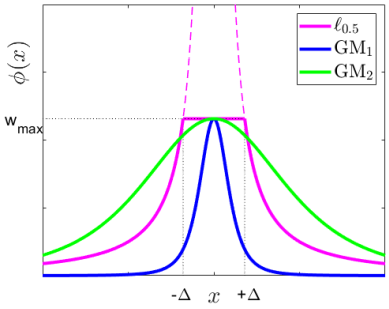
Rotation Error Measures: When we want to compare two sets of rotations (say an estimated set of cameras with ground truth), we need to align them to the same frame of reference. Two sets of rotations $\{\boldsymbol{R}_i\}_{i=1\ldots N}$ and



(a) Loss ($\rho(x)$)



(b) Weight ($\phi(x)$)



(c) Clipped Weight for $\ell_{\frac{1}{2}}$ ($\phi(x)$)

Figure 1: Comparison of loss and weight functions for $\ell_{\frac{1}{2}}$ and Geman-McClure functions.

$\{\boldsymbol{Q}_i\}_{i=1\ldots N}$ are compared after estimating the best aligning rotation $\boldsymbol{S}$ which minimizes the error $\sum_{i=1}^{N} d^2(\boldsymbol{R}_i, \boldsymbol{S}\boldsymbol{Q}_i)$. After alignment we can compute the distances $\boldsymbol{e}_i = d(\boldsymbol{R}_i, \mathbf{S}\mathbf{Q}_i), \forall i = 1, \cdots, N$ where $d(.,.)$ is the intrinsic metric on $\mathbb{SO}(3)$ stated in Sec. 3.2. The mean and median errors of an estimated set of rotations with respect to ground truth (after alignment) is the mean and median of set $\{\boldsymbol{e}_i\}$ and is denoted as $\bar{\boldsymbol{e}}$ and $\tilde{\boldsymbol{e}}$ respectively. Finally, the number of iterations to convergence is denoted by N_{iter}.

4. Analysis of $\ell_{\frac{1}{2}}$ Rotation Averaging

In this section we argue that for rotation averaging based on IRLS minimization, the performance in terms of speed and accuracy is intimately connected with both the nature of the loss function $\rho(\cdot)$ in Eqn. 7 and the initialization used.

We recall that in using IRLS minimization, at any point in the optimization (*i.e.* at any iteration), the rotation updates are estimated by solving a weighted least squares problem in the Lie algebra, *i.e.* $\left(\mathbf{A}^T\Phi\mathbf{A}\right)\Delta\boldsymbol{\Omega}_{\mathcal{V}} = \mathbf{A}^T\Phi\Delta\boldsymbol{\Omega}_{\mathcal{E}}$. Here the individual weights $\phi(.)$ in matrix Φ depend on the choice of the loss function as indicated in Table 1 and illustrated in Fig. 1a and 1b. The corresponding weights indicate the relative confidence that each observation is an inlier (high for inliers and low for outliers). Thus, the least squares approach $\rho(x) = x^2$ is not robust since all observations (including outliers) are given uniform weightage. In the state-of-the-art method of [5], this non-robustness is mitigated by the use of $\ell_{\frac{1}{2}}$ as the loss function which is robust to outliers, as are all loss functions $\ell_p\,(0 < p < 2)$ to different degrees. However, as can be seen in Fig. 1b, the weights of $\ell_{\frac{1}{2}}$ are unbounded and grow rapidly to ∞ as the error approaches zero. This has major implications for the performance of $\ell_{\frac{1}{2}}$-based rotation averaging.

All Lie-algebraic rotation averaging methods are local optimizations that require a reasonable initial guess for convergence to a good solution. If an initialization (say a random initialization) is far from the final answer, the IRLS weights are non-informative and the algorithm may fail to converge to a good solution. Moreover an initialization that is far from the final answer will need a large number of iterations (slow convergence). A remedy to this problem would be to choose a reasonably good initialization which will necessarily have to be derived from the specific dataset instance in question, but also have to be computationally inexpensive. An intuitive answer is to construct an initial estimate ($\boldsymbol{R}_{init}$) of the rotations of all cameras in $\mathcal{V}$, *i.e.* $\{\boldsymbol{R}_{\mathcal{V}}\}$ using a spanning tree (ST) from $\mathcal{G}$. However, it will be immediately realised that for edges (i, j) in this ST, the fitting error will always be zero and the corresponding IRLS weights would be undefined (or ∞ using a logical sleight of hand). This will prevent the estimate $\{\boldsymbol{R}_{\mathcal{V}}\}$ from moving away from $\boldsymbol{R}_{init}$ and the optimization routine would terminate at the initialization.

Practically the problem of undefined (or unbounded) weights is overcome by regularization *i.e.* by clipping the weights ϕ to a maximum value of w_{max}. Hence, we are effectively changing the $\ell_{\frac{1}{2}}$ loss function and the effective weight function is not the $\ell_{\frac{1}{2}}$ weight of Fig. 1b but the one illustrated in Fig. 1c which we denote as the clipped-$\ell_{\frac{1}{2}}$ function. In the implementation of [5] that is widely used[1], $w_{max} = 10^4$. Even then, with a ST initialization, the optimization terminates at the initialization itself in most cases. This problem could be potentially remedied by using an initial guess that does not result in an exact

[1] Available at `http://www.ee.iisc.ac.in/labs/cvl/research/rotaveraging`

fit with respect to most of the edges. Indeed, in [5] the $\ell_{\frac{1}{2}}$ method is initialized using the solution of another ℓ_1 rotation averaging routine (denoted L1RA). However, L1RA entails additional computational cost. Also, there are two important consequences of using the clipped-$\ell_{\frac{1}{2}}$ function. Within the range of $x \in [-\Delta, \Delta]$ indicated in Fig. 1c, the weights for clipped-$\ell_{\frac{1}{2}}$ are constant, *i.e.* all of these errors are treated as inliers in a least squares sense. Thus, when w_{max} is set to a high enough value (as in this case of [5] where $w_{max} = 10^4$), all $\boldsymbol{R}_{ij}$ observations that result in low errors end up being assigned a high weightage. Thus, the gradient estimate is biased by these high weight observations and convergence is slowed down. Setting the threshold to lower w_{max} does not fully allievate such problems as then the range of inliers $(-\Delta, \Delta)$ is broader and we practically end up doing least squares which is non-robust. In summary, while a clipped-$\ell_{\frac{1}{2}}$ function avoids unbounded weights, the act of clipping the weights can cause the method to be slowed down and potentially reduce accuracy.

We emphasise that the problems associated with high weights (upto w_{max} in case of [5]) are at play not only when initialised by a spanning tree but also with a non-ST (where the fitting error on the edges of the viewgraph is not all zero) initialization. In practice it is observed that a subset of edges with high weights (weights $> w_{max}$) arise during the trajectory of optimization irrespective of the initialization, and from then on, the number of edges with high weights increase significantly with every iteration. This is because the optimization routine is "dragged" towards the subset of edges where the high weights arise, thus biasing the final estimate, and hence "averaging" is not carried out in the true sense. We demonstrate the effect on 2 large datasets (Artsquad, Trafalgar from 1DSFM [31]). We take an L1RA initialization and provide it to $\ell_{\frac{1}{2}}$ and plot the number of edges with high weights (weights $> w_{max} = 1\text{e}{+4}$) or equivalently edges with almost zero fitting error (fitting error $< 2.7\text{e}{-4}^\circ$) in every iteration starting from the 6th iteration. In the first few iterations, because of the nature of initialization, there are not many edges with high weights. However, from Fig. 2 we can observe that the optimization routine gets biased by the subset of edges with high weights with increase in the number of iterations. On the other hand, our proposed method of Sec. 5 produces significantly fewer such edges (30 and 449 edges for Artsquad and Trafalgar respectively) and also gives significantly better accuracy and speed.

An additional drawback of using a clipped-$\ell_{\frac{1}{2}}$ loss function is that when the weights are below w_{max} their relative importance is governed by a fixed weighting function corresponding to $\ell_{\frac{1}{2}}$. Thus, the "bandwidth" of the weight function for clipped-$\ell_{\frac{1}{2}}$ is effectively fixed as can be seen in Fig. 1c and cannot be modified in accordance with the noise characterstics of individual datasets.

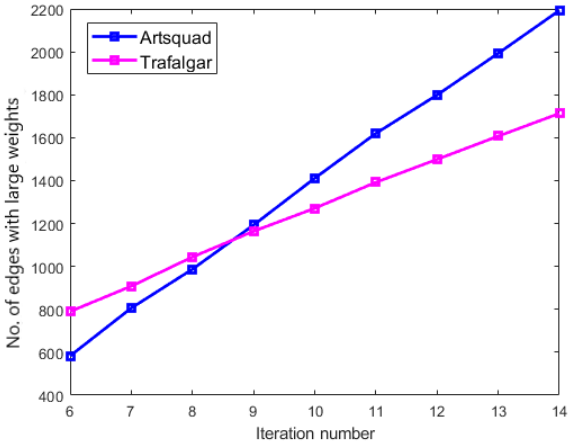


Figure 2: Behaviour of optimization with $\ell_{\frac{1}{2}}$ with L1RA initialization. The number of edges with high weights (equivalently almost zero fitting error) increase significantly as the optimization progresses.

In the first version of robust rotation averaging proposed by Chatterjee and Govindu [4], the authors had used a Geman-McClure loss function (with a fixed $\sigma = 5^\circ$). Subsequently, they used a clipped-$\ell_{\frac{1}{2}}$ loss function in [5] and argued that it was superior to the Geman-McClure loss function in [4]. However, we argue that the relatively poor performance of Geman McClure in [4, 5] can be mitigated by embedding it within a graduated optimization framework which can be shown to achieve superior speed and accuracy compared to the state-of-the-art approach of [5]. As can be seen in Fig. 1c the weights for the Geman-McClure loss function always remain bounded (scaled to w_{max} for illustrative purposes). Moreover, as indicated by the two illustrative examples with different σ, we can effectively control the bandwidth of the Geman-McClure weight function in accordance with the statistics of the residual errors. In Sec. 5 we present our proposed graduated optimization method using the Geman-McClure loss function.

5. Proposed Method

In this section, based on our analysis in Sec. 4, we propose a graduated optimization scheme [36] (GOM) using the Geman-McClure loss function in order to minimize the cost function given by (7). At every iteration, we minimize the cost function of the form:

$$\boldsymbol{\rho}(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{E}} \rho\left(\|\boldsymbol{r}_{ij}(\boldsymbol{\theta})\|\right) \tag{8}$$

In our particular problem, $\boldsymbol{\theta} = \boldsymbol{\Omega}_{\mathcal{V}}$ and $\boldsymbol{r}_{ij}(\boldsymbol{\theta}) = \boldsymbol{\omega}(\boldsymbol{R}^{-1}(\boldsymbol{\Delta\omega_j})\boldsymbol{R}_j^{-1}\boldsymbol{R}_{ij}\boldsymbol{R}_i\boldsymbol{R}(\boldsymbol{\Delta\omega_i}))$. Graduated optimization methods solve a sequence of objective functions $\boldsymbol{\rho}^k(\boldsymbol{\theta})$ starting from $k = k_{max}$ to $k = 0$ successively where the solution of $\boldsymbol{\rho}^{k+1}$ is used as initialization to solve for $\boldsymbol{\rho}^k$. $\boldsymbol{\rho}^k$'s are chosen such that $\boldsymbol{\rho}^{k+1}$ is a smoother approximation of $\boldsymbol{\rho}^k$ and is easier to optimize than $\boldsymbol{\rho}^k$ and eventually should approximate the original cost function, *i.e.*, $\boldsymbol{\rho}^0(\boldsymbol{\theta}) = \boldsymbol{\rho}(\boldsymbol{\theta})$. As in [36], the sequence of $\boldsymbol{\rho}^k$'s are constructed as:

$$\rho^k(r) := s_k^2 \rho\left(r/s_k\right) \tag{9}$$

$$\boldsymbol{\rho}^k(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{E}} \rho^k\left(\|\boldsymbol{r}_{ij}(\boldsymbol{\theta})\|\right) \tag{10}$$

where $s_0 = 1$ and $s_k < s_{k+1}$. We call our proposed method Rotation Averaging with Graduated Optimization (GRA) which is outlined in Algorithm 1.

Our choice of ρ is the Geman-McClure loss function. Thus the loss function at the k^{th} level of optimization is given as:

$$\rho^k(r) := s_k^2 \rho\left(r/s_k\right) = \frac{s_k^2\sigma_0^2 r^2}{2(r^2 + s_k^2\sigma_0^2)} \tag{11}$$

Effectively, $\rho^k(r)$ corresponds to a GM loss with the parameter being $\sigma_k = s_k\sigma_0$ which implies that $\frac{\sigma_{k+1}}{\sigma_k} = \frac{s_{k+1}}{s_k}$. Instead of implementing with a fixed value of k_{max}, we choose an initial value for the parameter σ in the Geman-McClure loss heuristically and with a chosen fixed value of $s = \frac{s_{k+1}}{s_k}$ for all k, we iterate the optimization until σ reaches a minimum threshold σ_0. We use IRLS with Gauss-Newton method to minimize each of $\boldsymbol{\rho}^k(\boldsymbol{\theta})$. We choose the relative stopping criterion as in [36] given by

$$\rho_\Delta^k := \frac{\Delta_\leq^k - \Delta_>^k}{\Delta_\leq^k + \Delta_>^k} = \frac{\boldsymbol{\rho}^k(\boldsymbol{\theta}^-) - \boldsymbol{\rho}^k\left(\boldsymbol{\theta}^+\right)}{\Delta_\leq^k + \Delta_>^k} \leq \eta \tag{12}$$

where in

$$\mathcal{I}_> := \left\{(i,j) \in \mathcal{E} : \|\boldsymbol{r}_{ij}\left(\boldsymbol{\theta}^+\right)\| > \|\boldsymbol{r}_{ij}(\boldsymbol{\theta}^-)\|\right\} \tag{13}$$

$$\boldsymbol{\rho}_>^k(\boldsymbol{\theta}) := \sum_{(i,j)\in\mathcal{I}_>} \rho^k\left(\|\boldsymbol{r}_{ij}(\boldsymbol{\theta})\|\right), \tag{14}$$

$$\boldsymbol{\rho}_\leq^k(\boldsymbol{\theta}) := \sum_{(i,j)\notin\mathcal{I}_>} \rho^k\left(\|\boldsymbol{r}_{ij}(\boldsymbol{\theta})\|\right) \tag{15}$$

$$\Delta_\leq^k := \boldsymbol{\rho}_\leq^k(\boldsymbol{\theta}^-) - \boldsymbol{\rho}_\leq^k\left(\boldsymbol{\theta}^+\right), \Delta_>^k := \boldsymbol{\rho}_>^k\left(\boldsymbol{\theta}^+\right) - \boldsymbol{\rho}_>^k(\boldsymbol{\theta}^-) \tag{16}$$

where $\boldsymbol{\theta}^+$ and $\boldsymbol{\theta}^-$ are the current and previous solutions. Thus, whenever $\rho_\Delta^k \leq \eta$ is satisfied, we graduate from $k+1$ to k in the optimization routine, *i.e.* we *anneal* the σ parameter in the Geman-McClure loss.

Our proposed method can be intuitively interpreted as follows: since an initial guess is relatively far from the final answer, the initial residual errors are large. If the Geman-McClure σ is relatively smaller than the error magnitudes, the weighting function $\phi(x) = \frac{\sigma^4}{(\sigma^2+x^2)^2}$ would tend to zero, leading to poor results. Instead, initially we should remain agnostic about inlier/outlier classification and correspondingly σ should be large. As we progressively move towards convergence, the individual fitting or residual errors for edges are more indicative of an inlier/outlier classification and therefore we should progressively make σ smaller to capture our belief in the relative weights of individual edges.

The GRA method is initialized with rotations using a random spanning tree. We emphasise that unlike the case with the original $\ell_{\frac{1}{2}}$ method, the weights of the Geman-McClure loss function are always bounded. Hence GRA initialized with an ST solution does not suffer from the degeneracies associated with extremely high weights as in the case of $\ell_{\frac{1}{2}}$. The key to obtaining good performance using GRA is to choose appropriate parameters of the initial value of σ in the Geman-McClure loss, the value of s_k and η in the stopping criterion. In our case, in Algorithm 1, for the first iteration, we choose σ heuristically based on the statistics of the residuals of the edges for a given initialization. We choose σ such that the knee or the inflection point (for Geman-McClure it is $GM_{knee} = \sigma/\sqrt{3}$) of the loss function is equal to 95^{th} percentile of the residuals (Lines $8-9$ in Algorithm 1). We choose $s_k = 3^k$ or equivalently $s = \frac{s_{k+1}}{s_k} = 3$. The minimum threshold σ_0 is chosen to be 0.5°. The parameter η is also chosen based on the percentile of the residuals lying within the inflection point of the Geman-McClure loss function. We choose $\eta = [0.5, 0.25, 0.125, 0.1]$ corresponding to 95^{th}, 90^{th}, 80^{th} percentiles of the residuals. For *e.g.*, $\eta = 0.5$ and $\eta = 0.1$ when more than 95% and less than 80% of the residuals respectively lie within the inflection point of Geman-McClure loss. This is because initially, when the importance of the edges (inliers/outliers) is unknown, we don't want the optimization to be strict, hence we choose η liberally in the initial stages. The parameter τ in Line 12 is chosen as $max\{\sigma, 0.08\}$ so that any weight corresponding to a residual above τ radians is diminished by a value of $t = 100$. We note that the same parameter settings are used for all the datasets. In our experiments, we observe that the value of t does not matter very much as long as $t \geq 2$.

6. Results

In this section we demonstrate the efficacy of our proposed GRA method on real-world datasets and compare with the state-of-the-art approaches. For comparing with the method of Chatterjee and Govindu [5] we use their implementation which runs an ℓ_1 cost for 5 iterations (L1RA) followed by an IRLS-based optimization using $\ell_{\frac{1}{2}}$ loss function (*i.e.* the clipped-$\ell_{\frac{1}{2}}$ loss function as discussed in Sec. 4). We refer to this method as L1RA+$\ell_{\frac{1}{2}}$. In addition we compare with two other state-of-the-art

Algorithm 1: GRA

Input: $\{\boldsymbol{R}_{ij}\}_{(i,j)\in\mathcal{E}}$, σ_0, N_{max}
Output: $\boldsymbol{R}_{\mathcal{V}} = \{\boldsymbol{R}_i\}_{i=1..N}$

Initialization: Set $\boldsymbol{R}_{\mathcal{V}} = \boldsymbol{R}_{\mathcal{V},span}$, $N_{iter} = 1$, $\boldsymbol{\Omega}_{\mathcal{V}}^{-} = \mathbf{0}$
while $\sigma \geq \sigma_0$ *AND* $N_{iter} \leq N_{max}$ **do**
  $\Delta\boldsymbol{\omega}_{ij} \leftarrow \boldsymbol{\omega}(\boldsymbol{R}_j^{-1}\boldsymbol{R}_{ij}\boldsymbol{R}_i)$
  $\theta_{ij} \leftarrow \|\Delta\boldsymbol{\omega}_{ij}\|$
  Concatenate $\Delta\boldsymbol{\omega}_{ij}$'s to obtain $\Delta\boldsymbol{\Omega}_{\mathcal{E}}$
  if $N_{iter} == 1$ **then**
    Concatenate θ_{ij}'s to obtain $\boldsymbol{\Theta}$
    $GM_{knee} = prctile(\boldsymbol{\Theta}, 95)$
    $\sigma \leftarrow \sqrt{3}GM_{knee}$
  end
  $\phi_{ij} = \sigma^2/(\sigma^2 + \theta_{ij}^2)$ `// Weights`
  $\phi_{ij} \leftarrow \phi_{ij}/t \quad \forall\ \{(i,j) \in \mathcal{E} : \theta_{ij} \geq \tau\}$
  Form $\boldsymbol{\Phi}$ with ϕ_{ij}'s as diagonal entries
  $\Delta\boldsymbol{\Omega}_{\mathcal{V}}^{+} \leftarrow -\left(\boldsymbol{A}^\top\Phi\boldsymbol{A}\right)^{-1}\boldsymbol{A}^\top\Phi\Delta\boldsymbol{\Omega}_{\mathcal{E}}$ `// IRLS`
  $\boldsymbol{R}_i \leftarrow \boldsymbol{R}_i\boldsymbol{R}(\boldsymbol{\Delta\omega_i}) \quad \forall i \in \{1, \cdots, N\}$
  Compute $\rho_{\Delta}^{k} = \frac{\boldsymbol{\rho}^k(\boldsymbol{\Omega}_{\mathcal{V}}^{-}) - \boldsymbol{\rho}^k(\boldsymbol{\Omega}_{\mathcal{V}}^{+})}{\Delta_{\leq}^{k} + \Delta_{>}^{k}}$.
  if $\rho_{\Delta}^{k} \leq \eta$ **then**
    $\sigma = \sigma/s$
  end
  $\boldsymbol{\Omega}_{\mathcal{V}}^{-} \leftarrow \boldsymbol{\Omega}_{\mathcal{V}}^{+}$
  $N_{iter} \leftarrow N_{iter} + 1$
end

methods that solve for rotation averaging in different ways. Shonan Averaging (Shonan) by Dellaert *et al*. [7] takes a semidefinite relaxation of a non-robust least squares cost function based on chordal distance, then builds a Riemannian staircase to solve the relaxed problem on higher dimensional rotation groups $\mathbb{SO}(p)$ ($p \geq 3$). They prove that under mild assumptions of noise, the solution of the semidefinite relaxation is the global minimizer of the actual robust *extrinsic* cost. Shi and Lerman [26] use a Message Passing Least Squares (MPLS) framework wherein the noise levels of $\boldsymbol{R}_{ij}$'s are estimated using a message passing algorithm based on cycle-consistency loss. These estimates are used in combination with the residuals for calculating the weights ϕ_{ij}'s. All the implementations are available online on the webpages of the respective authors. Finally, all the experiments are conducted on a machine with Intel 3GHz i7-5960 octa core CPU and 32GB RAM and our methods are implemented in MATLAB.

Variants of GRA: We introduce two variants of GRA. The first (GRA_1) is the vanilla version of GRA discussed in Algorithm 1 where a random spanning tree (ST_{random}) is taken as the initialization. However, we note that often the number of feature correspondences between an image pair (i, j) carries useful information on the reliability of the estimated relative rotations $\boldsymbol{R}_{ij}$. Therefore, in the second variant (GRA_{feat}), we use an initialization corresponding to a maximal spanning tree (ST_{feat}) based on edge weights equal to the number of matched features for each edge in the viewgraph $\mathcal{G}$.

6.1. Comparison with State-of-the-art Methods

The performance of L1RA+$\ell_{\frac{1}{2}}$ [5], Shonan [7], MPLS [26] and our methods GRA_1 and GRA_{feat} on the datasets is shown in Table 2. Since the MPLS method of [26] is a randomized procedure, we carry out 50 trials and report the average of the mean and median errors obtained. It should be noted that this is favourable towards the MPLS method as in some cases the variance of the errors is quite significant. We also note that while we report the errors for MPLS averaged over 50 trials, the timing reported in Table 3 is for a *single* trial.

We observe that GRA_{feat} performs better on 9 datasets, MPLS on 3 datasets and GRA_1 on 2 datasets, L1RA+$\ell_{\frac{1}{2}}$ [5] in 2 datasets in terms of the mean errors. In terms of the median errors, GRA_{feat} performs better on 9 datasets, MPLS on 2 datasets and GRA_1 on 5 datasets. We further note that the difference in mean and median errors on the VNC dataset between our method GRA_{feat} and MPLS is negligible and can be attributed to the uncertainty of the average of 50 randomized trials for MPLS. It is only in the case of the Madrid Metropolis (MDR) dataset that MPLS truly outperforms either of our GRA variants. It is also observed that Shonan [7] is outperformed by MPLS and both our GRA variants. In terms of computational speed, both variants of our GRA method significantly outperform all the other methods as shown in Table 3. Specifically on the large datasets ArtsQuad (ARQ), Trafalgar (TFG), and Piccadilly Circus (PIC), we observe that our methods achieve more than one order of magnitude improvement in speed.

Relative impact of initialization: As a final experiment, we examine the relative significance of both the loss functions and initialization in determining the quality of rotation averaging. Specifically, we wish to examine if the improved performance of our GRA method can be attributed to the spanning tree initialization using weights based on feature correspondences or if the Geman-McClure loss function embedded in GOM framework is significant in achieving superior performance. In Table 4 we compare the mean and median errors of our GRA method against the $\ell_{\frac{1}{2}}$ method of [5] on the large-scale Piccadilly dataset. The performance of both methods are compared for an initialization using a random spanning tree (ST_{random}) or a maximal

Dataset	$\bar{e}$					$\tilde{e}$				
	Shonan [7]	L1RA + $\ell_{0.5}$-IRLS [5]	MPLS [26]	Our Methods		Shonan [7]	L1RA + $\ell_{0.5}$-IRLS [5]	MPLS [26]	Our Methods	
				GRA_1	GRA_{feat}				GRA_1	GRA_{feat}
ALM	8.55	4.79	4.28	**4.27**	4.33	4.67	2.14	1.93	**1.69**	1.90
ARQ	NC	4.04	3.18	3.16	**2.98**	NC	2.54	1.81	1.55	**1.52**
ELS	6.76	3.39	3.25	4.48	**2.30**	3.18	1.22	1	1.52	**0.81**
MDR	12.49	8.13	**5.67**	8.58	8.91	8.48	3.08	**1.82**	3.26	3.48
MND	7.33	1.66	1.39	8.17	**1.01**	3.43	0.71	0.63	3.33	**0.45**
NYC	8.41	**3.13**	3.23	3.30	3.35	5.53	1.38	1.36	**1.14**	1.24
ND1	7.95	3.81	**2.93**	3.87	3.00	3.93	0.98	0.92	1.01	**0.76**
PDP	15.14	4.94	4	3.65	**3.40**	8.75	2.61	1.96	1.74	**1.72**
PIC	NC	6.92	4.69	5.75	**4.02**	NC	3.13	2.02	1.74	**1.67**
ROF	13.79	3.19	2.87	2.62	**2.33**	11.6	1.71	1.45	1.30	**1.20**
SNF	7.37	**3.63**	3.96	3.64	4.64	6.36	3.53	3.21	**3.08**	3.67
TOL	6.93	3.98	3.99	4.21	**3.65**	3.97	2.45	2.38	**2.10**	2.11
TFG	NC	3.6	4.50	**3.21**	3.89	NC	2	2.70	**1.58**	1.64
USQ	14.2	10.13	6.29	8.89	**5.96**	10.35	4.97	3.43	3.88	**3.41**
VNC	NC	11.15	**7.97**	11.49	7.99	NC	4.64	**3.69**	4.44	3.76
YKM	8.17	3.55	3.57	3.66	**3.53**	5.71	1.62	1.57	1.49	**1.48**

Table 2: Comparison of mean and median errors (in degrees) for various RA algorithms. Note that the error reported for MPLS [26] is averaged over 50 trials. 'NC' indicates an experiment did not converge.

Dataset	Time in seconds				
	Shonan [7]	L1RA +$\ell_{0.5}$-IRLS [5]	MPLS [26]	Our Methods	
				GRA_1	GRA_{feat}
ALM	36.8	13.3	21.4	**1.6**	**1.1**
ARQ	NC	63.1	86	**3.7**	**3.2**
ELS	6	1.6	3.8	**0.2**	**0.3**
MDR	7.6	1.8	5.1	**0.4**	**0.4**
MND	14.6	3.9	8.4	**0.4**	**0.5**
NYC	2.3	1.3	3.8	**0.3**	**0.3**
ND1	53.7	12.3	21.3	**1.0**	**1.0**
PDP	9.6	1.8	4	**0.3**	**0.2**
PIC	NC	245.9	218.7	**5.6**	**4.9**
ROF	231.2	6.9	12.7	**1.1**	**0.9**
SNF	1471.7	54.5	120.9	**1.3**	**1.0**
TOL	4	1.2	3.6	**0.3**	**0.3**
TFG	NC	375.9	789.2	**9.8**	**9.4**
USQ	12.1	3.1	5.2	**0.5**	**0.5**
VNC	NC	14.2	28.3	**1.0**	**0.8**
YKM	4.9	1.1	4	**0.4**	**0.4**

Table 3: Comparison of time taken for different RA algorithms.

spanning tree based on weights equal to the number of feature correspondences (ST_{feat}). Since any spannning tree initialization with $\ell_{\frac{1}{2}}$ loss function has the problem of unbounded weights as discussed in Sec. 4, we initialize the weights in the first IRLS iteration as all 1's as is implemented by [5] to avoid this problem for the $\ell_{\frac{1}{2}}$ loss function.

As will be observed in a comparative sense from Table 4, initialization using a feature weighted spanning tree does improve performance, but the use of Geman-McClure loss function in a GOM framework has a greater contribution to improving the quality of the rotation averaging estimates

Initialization	$\bar{e}$		$\tilde{e}$	
	$\ell_{\frac{1}{2}}$-IRLS	GRA	$\ell_{\frac{1}{2}}$-IRLS	GRA
ST_{random}	7.14	**5.75**	3.04	**1.74**
ST_{feat}	5.68	**4.02**	2.87	**1.67**

Table 4: Mean and median errors on Piccadilly dataset for $\ell_{\frac{1}{2}}$-IRLS and GRA with different initializations

and plays a role in improvement independent of the type of initialization. In summary, the Geman-McClure loss function embedded in a GOM framework plays a crucial role in achieving significant improvement which can be further enhanced by an intelligent choice of initialization.

7. Conclusion

In this paper we have presented an analysis of the properties and behaviour of the state-of-the-art rotation averaging method [5] based on IRLS optimization of an $\ell_{\frac{1}{2}}$ loss function. Further, we have developed a graduated optimization approach based on Geman-McClure loss function to mitigate the drawbacks of the $\ell_{\frac{1}{2}}$ approach. Through extensive testing on a large number of real-world datasets, we have demonstrated that our proposed approach significantly outperforms [5] and other state-of-the-art methods.

References

[1] F. Arrigoni, L. Magri, B. Rossi, P. Fragneto, and A. Fusiello. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 491–498, 2014.

[2] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

[3] Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International journal of computer vision*, 19(1):57–91, 1996.

[4] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 521–528, 2013.

[5] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):958–972, 2018.

[6] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR 2011*, pages 3001–3008, 2011.

[7] Frank Dellaert, David M. Rosen, Jing Wu, Robert E. Mahony, and Luca Carlone. Shonan rotation averaging: Global optimality by surfing $so(p)^n$. In *ECCV (6)*, volume 12351 of *Lecture Notes in Computer Science*, pages 292–308. Springer, 2020.

[8] Daniel M Dunlavy and Dianne P O'Leary. Homotopy optimization methods for global optimization. *Report SAND2005-7495, Sandia National Laboratories*, 2005.

[9] Chris Engels, Henrik Stewénius, and David Nistér. Bundle adjustment rules. *Photogrammetric computer vision*, 2(32), 2006.

[10] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging and strong duality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–135, 2018.

[11] Johan Fredriksson and Carl Olsson. Simultaneous multiple rotation averaging using lagrangian duality. In *ACCV (3)*, volume 7726 of *Lecture Notes in Computer Science*, pages 245–258. Springer, 2012.

[12] Venu Madhav 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, pages II–II. IEEE, 2001.

[13] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.

[14] Venu Madhav Govindu. Robustness in motion averaging. In *ACCV (2)*, volume 3852 of *Lecture Notes in Computer Science*, pages 457–466. Springer, 2006.

[15] Peter J Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):149–170, 1984.

[16] Richard Hartley, Khurrum Aftab, and Jochen Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *CVPR 2011*, pages 3041–3048. IEEE, 2011.

[17] Richard I. Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *Int. J. Comput. Vis.*, 103(3):267–305, 2013.

[18] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.

[19] Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

[20] Ken-Ichi Kanatani. *Group-theoretical methods in image understanding*, volume 20. Springer Science & Business Media, 2012.

[21] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[22] Hossein Mobahi and John Fisher III. A theoretical analysis of optimization by gaussian continuation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[23] Anand Rangarajan and Rama Chellappa. Generalized graduated nonconvexity algorithm for maximum a posteriori image estimation. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, volume 2, pages 127–133. IEEE, 1990.

[24] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, 1998.

[25] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] Yunpeng Shi and Gilad Lerman. Message passing least squares framework and its application to rotation synchronization. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 8796–8806. PMLR, 2020.

[27] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80(2):189–210, 2008.

[28] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[29] Peter JM Van Laarhoven and Emile HL Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer, 1987.

[30] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2(2):145–193, 2013.

[31] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *European Conference on Computer Vision*, pages 61–75. Springer, 2014.

[32] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013.

[33] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134, 2020.

[34] Christopher Zach. Robust bundle adjustment revisited. In *European Conference on Computer Vision*, pages 772–787. Springer, 2014.

[35] Christopher Zach and Guillaume Bourmaud. Iterated lifting for robust cost optimization. In *BMVC*, 2017.

[36] Christopher Zach and Guillaume Bourmaud. Descending, lifting or smoothing: Secrets of robust cost optimization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 547–562, 2018.

[37] Christopher Zach and Guillaume Bourmaud. Pareto meets huber: Efficiently avoiding poor minima in robust estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10243–10251, 2019.

[38] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (ToG)*, 33(4):1–12, 2014.