Efficient and Robust Large-Scale Rotation Averaging

Avishek Chatterjee Venu Madhav Govindu
Department of Electrical Engineering
Indian Institute of Science, Bengaluru, INDIA
{avishek|venu}@ee.iisc.ernet.in

Abstract

In this paper we address the problem of robust and efficient averaging of relative 3D rotations. Apart from having an interesting geometric structure, robust rotation averaging addresses the need for a good initialization for large-scale optimization used in structure-from-motion pipelines. Such pipelines often use unstructured image datasets harvested from the internet thereby requiring an initialization method that is robust to outliers. Our approach works on the Lie group structure of 3D rotations and solves the problem of large-scale robust rotation averaging in two ways. Firstly, we use modern ℓ_1 optimizers to carry out robust averaging of relative rotations that is efficient, scalable and robust to outliers. In addition, we also develop a two-step method that uses the ℓ_1 solution as an initialisation for an iteratively reweighted least squares (IRLS) approach. These methods achieve excellent results on large-scale, real world datasets and significantly outperform existing methods, i.e. the state-of-the-art discrete-continuous optimization method of [3] as well as the Weiszfeld method of [8]. We demonstrate the efficacy of our method on two large-scale real world datasets and also provide the results of the two aforementioned methods for comparison.

1. Introduction

In this paper we address the problem of robust averaging of 3D relative rotations in the context of structure-from-motion (henceforth SfM) estimation. The canonical SfM solution of nonlinear bundle adjustment that minimizes reprojection error is statistically optimal [15]. Using efficient optimizers and heuristics, current SfM pipelines such as the well-known PhotoTourism [13] and its improvements have successfully solved increasingly larger problems. Such pipelines need to robustly handle outliers that may exist in unstructured image datasets harvested from the internet. Given the high dimensional optimization involved and the presence of outliers, successful convergence critically depends on both a good initial guess as well as the robustness of the optimization methods used. This is often achieved by *incremental* bundle adjustment that robustly grows the solution one image at a time instead of carrying out a single batch optimization.

2. Motion Averaging Preliminaries

In contrast to bundle adjustment, an alternate approach for global camera motion estimation is to average relative motions. Motion averaging was introduced in [5] and further developed in [6] to use the geometric structure of Lie groups. While such a formulation is generic enough to apply to a variety of scenarios, in the context of SfM, motion averaging leverages the observation that in a set of N images, there exist as many as ${}^{N}C_2 = \frac{N(N-1)}{2}$ pairs for which the relative motions can be estimated. We can represent the relationships between all the cameras by means of a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ known as the viewgraph where each vertex in $\mathcal{V}$ represents a camera and an edge $(i, j) \in \mathcal{E}$ implies that the motion between cameras i and j can be estimated.[1] In the following we represent 3D rotations by the 3×3 orthonormal matrix, $\mathbf{R}$, i.e. $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ and $|\mathbf{R}| = +1$. If with respect to a global frame of reference we denote the *absolute* 3D rotation of the k-th camera as $\mathbf{R}_k$, then the *relative* rotation between cameras i and j, $\mathbf{R}_{ij}$, can be written in terms of the global motions of cameras i and j, i.e.

$$\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i{}^{-1}, \ \forall \{i, j\} \in \mathcal{E} \tag{1}$$

All 3×3 rotation matrices form a closed group known as the Special Orthogonal group $SO(3)$ which also has the differentiable properties of a Riemannian manifold, i.e., $SO(3)$ is a Lie group which is the basis for an efficient approach for averaging rotations. Informally speaking, apart from the standard properties of a group, being equipped with the smooth differentiable structure of a Riemannian manifold endows a Lie group with the

[1] Typically, this would imply that we have a sufficient number of point correspondences between cameras i and j to solve for their relative geometry.

additional properties that the product and inverse operations are differentiable mappings [6]. The local neighborhood of a point on a Lie group is topologically equivalent to a vector space, i.e. the neighborhood can be adequately described by the tangent-space at a point known as the Lie algebra. An important property of Lie groups is the existence of direct mappings between the Lie algebra and the group and vice-versa. These mappings are the familiar exponential and logarithm functions respectively. In the case of 3D rotations, the Lie algebra is denoted as $\mathfrak{so}(3)$.

If we denote $\boldsymbol{\omega} = \theta\mathbf{n} \in \mathfrak{so}(3)$ where θ is the angle of rotation about a unit-norm axis $\mathbf{n}$, then the exponential and logarithmic mappings between the Lie group and the corresponding Lie algebra for 3D rotations are given as $\mathbf{R} = e^{[\boldsymbol{\omega}]_\times} \in SO(3)$ and $[\boldsymbol{\omega}]_\times = \log(\mathbf{R}) \in \mathfrak{so}(3)$ where $[\boldsymbol{\omega}]_\times \in \mathfrak{so}(3)$ is the skew-symmetric form of $\boldsymbol{\omega}$. The intrinsic bi-invariant distance on $SO(3)$, $d(\mathbf{R}_1, \mathbf{R}_2) = \frac{1}{\sqrt{2}}||\log(\mathbf{R}_1{\mathbf{R}_2}^{-1})||_F = \frac{1}{\sqrt{2}}||\log(\mathbf{R}_2{\mathbf{R}_1}^{-1})||_F$, where $||.||_F$ is the Frobenius norm. This leads to the Frechet mean or the intrinsic average $\mu \in SO(3)$ of a set of rotations $\{\mathbf{R}_1, \cdots, \mathbf{R}_n\}$ which is defined as

$$\arg \min_{\mu \in SO(3)} \sum_{k=1}^{n} d^2(\mathbf{R}_k, \mu) \tag{2}$$

While there is no closed form solution, an iterative algorithm for estimating this intrinsic average is available as Algorithm A1 of [6]. Analogous to estimating the mean rotation, we may use the intrinsic distance to fit global or absolute rotations to a given set of relative rotation observations. If with respect to a given frame of reference, we define the global rotations as $\mathbf{R}_{global} = \{\mathbf{R}_1, \cdots, \mathbf{R}_N\}$, using Eqn. 1 we can define the global rotation estimate as

$$\arg \min_{\mathbf{R}_{global}} \sum_{(i,j)\in\mathcal{E}} d^2(\mathbf{R}_{ij}, \mathbf{R}_j{\mathbf{R}_i}^{-1}) \tag{3}$$

If we consider a single relationship $\mathbf{R}_{ij} = \mathbf{R}_j{\mathbf{R}_i}^{-1}$ represented by an edge $(i, j) \in \mathcal{E}$, the first-order approximation of the corresponding Lie algebraic relationship can be written as $\boldsymbol{\omega}_{ij} = \boldsymbol{\omega}_j - \boldsymbol{\omega}_i$.[2] We further denote the angle representations of all the rotations as $\boldsymbol{\omega}_{global} = [\boldsymbol{\omega}_1, \cdots, \boldsymbol{\omega}_N]^T$. Consequently, we can write

$$\boldsymbol{\omega}_{ij} = \boldsymbol{\omega}_j - \boldsymbol{\omega}_i = \underbrace{\left[\ \cdots - \mathbf{I} \cdots \mathbf{I} \cdots\ \right]}_{\mathbf{A}_{ij}} \boldsymbol{\omega}_{global} \tag{4}$$

where in $\mathbf{A}_{ij}$, $\mathbf{I}$ and $-\mathbf{I}$ are placed as 3×3 blocks in the appropriate locations of j and i respectively. While Eqn. 4 represents the relationship obtained from a single relative motion edge in $\mathcal{E}$, we can collect all such relationships into a single system of equations as

$$\mathbf{A}\boldsymbol{\omega}_{global} = \boldsymbol{\omega}_{rel} \tag{5}$$

where $\boldsymbol{\omega}_{rel}$ is the vector made by stacking all relative rotation observations $\boldsymbol{\omega}_{ij}$ and $\mathbf{A}$ is made by stacking the corresponding matrices $\mathbf{A}_{ij}$. In other words, Eqn. 5 is obtained by concatenating all the relationships given by Eqn. 4 for each edge in $\mathcal{E}$. This relationship suggests the following algorithm to obtain a global rotation estimate and is adapted from [6]. We may also incorporate appropriate weights in Eqn. 5 to reflect the reliability of individual $\mathbf{R}_{ij}$ estimates.

Algorithm 1 Lie-Algebraic Relative Rotation Averaging

```
Input: ${\mathbf{R}_{ij1}, \cdots, \mathbf{R}_{ijk}}$ ($|\mathcal{E}|$ relative rotations)
Output: $\mathbf{R}_{global} = {\mathbf{R}_1, \cdots, \mathbf{R}_N}$ ($|\mathcal{V}|$ absolute rotations)
Initialisation: $\mathbf{R}_{global}$ to an initial guess
    while $||\Delta\boldsymbol{\omega}_{rel}|| < \epsilon$ do
        1. $\Delta\mathbf{R}_{ij} = {\mathbf{R}_j}^{-1}\mathbf{R}_{ij}\mathbf{R}_i$
        2. $\Delta\boldsymbol{\omega}_{ij} = \log(\Delta\mathbf{R}_{ij})$
        3. Solve $\mathbf{A}\Delta\boldsymbol{\omega}_{global} = \Delta\boldsymbol{\omega}_{rel}$
        4. $\forall k \in [1, N], \mathbf{R}_k = \mathbf{R}_k exp(\Delta\boldsymbol{\omega}_k)$
    end while
```

In Algorithm 1, we have simplified the notation for ease of comprehension, see [6] for further details. In this method for averaging relative rotations, for all the edges in $\mathcal{E}$, the discrepancy between the observations $\mathbf{R}_{ij}$ and the current estimate for the relative rotation as implied by the global estimate, i.e. $\mathbf{R}_j{\mathbf{R}_i}^{-1}$ is averaged in the Lie algebra. Following this averaged estimate (step 3), the individual rotations are updated by mapping the Lie algebraic update back to the rotation group via the exponential mapping. It may be further noted that following the averaging in the Lie algebra, the exponential mapping used to update individual rotations (step 4) ensures that at every point, the estimates are on the rotation manifold, i.e. this algorithm provides an intrinsic estimate for the global rotation $\mathbf{R}_{global}$. We also note that in practice the rotation for any camera is fixed to $\mathbf{I}$ to remove the gauge freedom of $\mathbf{R}_{global}$.

This method of averaging relative rotations can solve for the global rotation of all cameras in an efficient manner, see [6] for details. In the context of SfM, given camera calibration information and a sufficient number of point correspondences between images i and j, we can estimate $\mathbf{R}_{ij}$ either from the epipolar geometry or bundle adjustment between the two images. Given rotations, solving for camera translation and 3D structure is a linear problem. In [12],

[2] The exact relationship is $\boldsymbol{\omega}_{ij} = BCH(\boldsymbol{\omega}_j, -\boldsymbol{\omega}_i)$ where $BCH(.,.)$ is the Baker-Campbell-Hausdorff form [6]. However, the first-order approximation allows us to develop an efficient iterative algorithm.

the authors use relative rotation estimates and information from vanishing point matches to estimate $\mathbf{R}_{global}$. In [11], the authors use rotation averaging in a RANSAC framework to robustly estimate the global rotations. [4] uses a dual formulation on a quaterion representation for rotation averaging while [9] provides a survey on rotation averaging.

2.1. Robust Averaging of Relative Rotations

In [6], step 3 of algorithm A2 is specifically solved in a least squares sense by minimizing the ℓ_2 norm $||\mathbf{A}\Delta\boldsymbol{\omega}_{global} - \Delta\boldsymbol{\omega}_{rel}||^2$, i.e. $\Delta\boldsymbol{\omega}_{global} = \mathbf{A}^{\dagger}\Delta\boldsymbol{\omega}_{rel}$, where $\mathbf{A}^{\dagger}$ is the pseudo-inverse of $\mathbf{A}$. Although this method can efficiently average relative motions, as is well known least squares solutions are non-robust and can give highly erroneous results in the presence of even a single outlier. Such outliers commonly occur in many SfM contexts including outdoor scenes with repeated structures such as doors, windows and columns that are indistinguishable at a local image level, or when we have multiple, independently moving objects in a scene, say cars, pedestrians etc. The problem can be mitigated to some extent by ensuring that the individual relative rotation estimates $\mathbf{R}_{ij}$ are robustly estimated, e.g. by using RANSAC or other robust methods for epipolar geometry estimation. However, in the case of erroneous matches due to repeated structures it may not be possible to disambiguate and recover from errors by observing two images at a time, leading to outlier estimates of $\mathbf{R}_{ij}$. In other words, in many contexts the presence of outliers cannot be fully avoided or it can be prohibitively expensive to remove them. The computational burden can be especially heavy in the case when we need to deal with a very large number of images. In effect, modern SfM systems need a rotation averaging scheme that is efficient, scalable and robust to outliers in the relative rotations.

2.2. Existing Methods

In recent years a few methods have been developed to incorporate robustness into the averaging of relative rotations and they can be classified into two approaches. Some methods detect outliers in the set of relative rotations and remove them before carrying out ℓ_2 norm averaging [7, 18]. In [7], the author uses a RANSAC-based approach for detect and remove outliers. Subsequently, the inliers are averaged using Algorithm 1. In [18], the authors utilize the fact that a loop of transformations should result in the identity transformation if there is no noise or outliers in the loop. They collect statistics of the residual transformation over many overlapping loops which are then used in a loopy belief propagation framework to classify individual edges into inliers and outliers. Both these approaches suffer from the limitation of increased computational complexity with the increased size of the viewgraph $\mathcal{G}$. In additionally, the belief propagation stage in [18] is expensive to compute.

The second category of methods robustly average relative rotations without the need to explicitly detect and remove outliers. Methods presented in [3, 8] and the approach of this paper belong to this category. In [3], the authors use a combination of discrete-continuous optimization (henceforth DISCO) to average relative rotations in a robust manner. Using a robust distance measure between rotations, $d^{\mathbf{R}}(\mathbf{R}_1, \mathbf{R}_2) = \rho_R(||\mathbf{R}_1 - \mathbf{R}_2||)$, DISCO sets up an optimization cost function as

$$D(\mathbf{R}_{global}) = \sum_{(i,j)\in\mathcal{E}} d^{\mathbf{R}}(\mathbf{R}_{ij}, \mathbf{R}_j\mathbf{R}_i{}^{-1}) + \ prior\ terms \tag{6}$$

where the additional prior terms include information from other sensor measurements that provide an approximate estimate for individual rotations in $\mathbf{R}_{global}$.[3] The minimization of the cost in Eqn. 6 is carried out in two stages in [3]. Ignoring the twist component of rotations, [3] parametrizes rotations as a discrete set of labels on a unit sphere. The resultant averaging problem in the discrete labelling form is solved using discrete loopy belief propagation on a Markov Random Field. The approximate discrete solution is used as initialization for a non-linear minimization of Eqn. 6 using Rodrigues parameters.

In [8] (henceforth Weiszfeld), the authors utilize the fact that the ℓ_1 norm is more robust to outliers than the ℓ_2 norm to robustly average relative rotations. Their minimizer is the ℓ_1 analogue of Eqn. 3, i.e. $\sum_{(i,j)\in\mathcal{E}} d(\mathbf{R}_{ij}\mathbf{R}_i, \mathbf{R}_j)$. While the Weiszfeld method can be used to robustly average absolute rotations, it does not directly lend itself to the averaging of relative rotations. To work around this limitation, [8] updates individual rotations one-at-a-time while holding all other rotation estimates fixed. If we hold all rotations in $\mathbf{R}_{global}$ fixed except for $\mathbf{R}_j$, then the Weiszfeld optimization reduces to the ℓ_1 average of the rotations $\mathcal{R}_j = \{\mathbf{R}_{ij}\mathbf{R}_i | \forall i \in \mathcal{N}(j)\}$ where $\mathcal{N}(j)$ is the set of vertices connected to vertex j. This is nothing but the Weiszfeld-based median of $\mathcal{R}_j$. Thus, in [8], $\mathbf{R}_{global}$ is indirectly estimated using nested iteration where the inner loop consists of updating $\mathbf{R}_j$ to the Weiszfeld median of $\mathcal{R}_j$ and the outer loop is iterated till convergence. The reader may refer to [8] for details.

Although the methods of DISCO [3] and Weiszfeld [8] solve the problem of robust averaging of relative rotations, they suffer from significant limitations. DISCO [3] can handle large-scale averaging problems but at a significant cost of implementational and computational complexity.

[3] Note that the form of the cost function given in Eqn. 6 is a modified version of Eqn. 5 of [3] since their notation is different from ours.

By ignoring the twist component of 3D rotations, the authors limit the scope of applicability of their methods. More crucially, disregarding the geometric structure of $SO(3)$ and treating averaging as a complicated discrete labelling MRF problem makes DISCO an extremely expensive problem requiring a significant amount of hardware. In contrast, the Weiszfeld method of [8] elegantly utilises the geometric structure of the $SO(3)$ group for robust estimation. Nevertheless, the overall algorithm of [8] suffers from the significant drawback that it scales poorly with increasing size of the problem. Since the Weiszfeld method cannot update all rotations in $\mathbf{R}_{global}$ simultaneously, it is forced to update the rotation of each camera (i.e. vertex in $\mathcal{V}$) one at a time. This is nothing but a distributed consensus approach where vertex updates on a graph depend only on the neighbours. For a discussion of distributed consensus methods applied to computer vision problems, see [17, 16]. Consequently, in the Weiszfeld method, any change of value at a given vertex takes a long time to propagate itself over the entire viewgraph $\mathcal{G}$. On the average, the number of iterations taken to propagate the information of change will depend on the diameter of the graph, $dia(\mathcal{G})$. As we shall demonstrate in Sec. 3, for datasets that are larger than those considered in [8], the Weiszfeld method does poorly both in accuracy and time.

2.3. ℓ_1 Rotation Averaging

Since neither DISCO [3] nor the Weiszfeld method [8] satisfies both the requirements of a computationally efficient and scalable robust averaging scheme, we propose an alternate approach that can efficiently handle large-scale problems. If we consider Eqn. 4, we recognise that if a specific relative rotation is an outlier, then an ℓ_2 average in the Lie algebra will result in wrong rotation estimates. The remedy lies in robust Lie algebraic averaging so that at each iteration the averaging step in the Lie algebra is robust. In this case the overall estimate of $\mathbf{R}_{global}$ is unaffected by the presence of individual outliers. By exploiting recent advances in convex optimization for ℓ_1 cost functions, we can achieve the twin goal of accuracy and scalability in the presence of outliers.

As the Lie algebra is a vector space, our problem of robust averaging in the Lie algebra is analogous to robust estimation for a linear system of equations. Consider the classical linear algebra formulation of a system of equations, $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$ where $m > n$. If $\mathbf{A}$ is full rank, the input signal $\mathbf{x}$ can be recovered given the observation vector $\mathbf{b}$. However, the difficulty of the problem changes when the observation is corrupted by an unknown additive vector $\mathbf{e}$ that contains both noise and outlier errors, i.e. we have $\mathbf{b} = \mathbf{Ax} + \mathbf{e}$. Evidently our ability to recover $\mathbf{x}$ depends on the nature of the corrupting vector $\mathbf{e}$. Recent work in compressive sensing [2] has shown that we can efficiently and accurately estimate $\mathbf{x}$ in the presence of outliers by solving

$$\arg\min_{\mathbf{x}} ||\mathbf{Ax} - \mathbf{b}||_{\ell_1} \tag{7}$$

Eqn. 7 is an instance of ℓ_1 minimization that is known to be more robust to outliers than ℓ_2 methods. An efficient implementation that solves Eqn. 7 is publicly available in the ℓ_1-magic package [1]. If we consider the presence of outliers in the observed relative rotations in the Lie algebra at any given iteration, we have $\Delta\boldsymbol{\omega}_{rel} = \mathbf{A}\Delta\boldsymbol{\omega}_{global} + \mathbf{e}$. Our ℓ_1 robust rotation averaging method (denoted as **L1RA**) can be stated as Algorithm 1 where step 3 is solved as the minimizer of $||\mathbf{A}\Delta\boldsymbol{\omega}_{global} - \Delta\boldsymbol{\omega}_{rel}||_{\ell_1}$ using Eqn. 7. Here $\Delta\boldsymbol{\omega}_{global} \in \mathbb{R}^{3|\mathcal{V}|}$ and $\Delta\boldsymbol{\omega}_{rel} \in \mathbb{R}^{3|\mathcal{E}|}$ since each rotation angle is a 3-vector. Evidently in the viewgraph $\mathcal{G}$, the number of edges $|\mathcal{E}|$ is far larger than the number of vertices $|\mathcal{V}|$. It will also be noted that each row of $\mathbf{A}$ has only two non-zero entries, specifically $\{-1, +1\}$ making $\mathbf{A}$ exceedingly sparse and our solution efficient as multiplication by $\mathbf{A}$ results in additions.

2.4. IRLS Rotation Averaging

While the ℓ_1 optimization solution outlined above provides an ℓ_1 rotation average estimate in the presence of outliers, we can further improve this solution by treating the problem of robust rotation averaging as one of robust regression or M-estimator modifications of least squares estimation. From Eqn. 3 recall that ideally we would like to carry out an ℓ_2 averaging of relative rotations. Since the observations of relative rotations are corrupted by the presence of outliers, we take recourse to robust estimation using the ℓ_1 norm instead of ℓ_2. However if the solution obtained by ℓ_1 minimization is accurate enough, the fitting error for individual relative rotations gives us a good estimate of the reliability of the input $\mathbf{R}_{ij}$. We utilise this information to iteratively solve for a robust weighted least squares averaging of the relative rotations.

Algorithm 2 Iterativeley Reweighted Least Squares (IRLS)

```
Set $\mathbf{x}$ to initial guess
  while $||\mathbf{x} - \mathbf{x}_{prev}|| < \epsilon$ do
    1. $\mathbf{x}_{prev} \leftarrow \mathbf{x}$
    2. $\mathbf{e} \leftarrow \mathbf{Ax} - \mathbf{b}$
    3. $\mathbf{\Phi} \leftarrow \mathbf{\Phi}(\mathbf{e})$
    4. $\mathbf{x} \leftarrow (\mathbf{A}^T\mathbf{\Phi}\mathbf{A})^{-1}\mathbf{A}\mathbf{\Phi}\mathbf{b}$
  end while
```

Consider the system of equations $\mathbf{Ax} = \mathbf{b}$ that we wish to minimize in a robust ℓ_2 sense. Instead of using the standard least squares cost function $\mathbf{e}^T\mathbf{e}$ where $\mathbf{e} = \mathbf{A}\mathbf{x} - \mathbf{b}$, we choose to minimize a robust version of the cost function, i.e.

$$E = \sum_i \rho(||\mathbf{e}_i||) \tag{8}$$

where $\mathbf{e}_i$ is the i-th element of the error vector $\mathbf{e}$ and $\rho(.)$ is a robust loss function, see Section 6.1.4 of [14]. For this paper we choose the Huber-like loss function of $\rho(x) = \frac{x^2}{x^2+\sigma^2}$, where σ is a tuning parameter that determines the transition from quadratic to fixed loss for increasing magnitude of x. To solve for $\mathbf{x}$ that minimizes the robust cost of Eqn. 8, we observe that

$$\begin{aligned} \min_{\mathbf{x}} E = \min_{\mathbf{x}} \sum_i \rho(||\mathbf{e}_i||) &= \min_{\mathbf{x}} \sum_i \frac{\mathbf{e}_i^2}{\mathbf{e}_i^2 + \sigma^2} \\ \Rightarrow \frac{\partial E}{\partial \mathbf{x}} &= \frac{\partial E}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial \mathbf{x}} = 0 \\ \Rightarrow \mathbf{A}^T \mathbf{\Phi}(\mathbf{e}) \mathbf{A}\mathbf{x} &= \mathbf{A}^T \mathbf{\Phi}(\mathbf{e}) \mathbf{b} \end{aligned} \tag{9}$$

where $\mathbf{\Phi}(\mathbf{e})$ is a diagonal matrix with $\mathbf{\Phi}(i,i) = \frac{\sigma^2}{(\mathbf{e}_i^2+\sigma^2)^2}$. In the derivation shown in Eqn. 9, we have omitted the expressions for the individual terms $\frac{\partial E}{\partial \mathbf{e}}$ and $\frac{\partial \mathbf{e}}{\partial \mathbf{x}}$ for brevity of presentation. The system of equations in Eqn. 9 is a non-linear problem due to the dependency of $\mathbf{\Phi}(\mathbf{e})$ on $\mathbf{x}$ through $\mathbf{e}$. However, such problems are often solved by an iterative approach. If we hold $\mathbf{x}$ fixed, we can compute $\mathbf{e} = \mathbf{A}\mathbf{x} - \mathbf{b}$ and in turn fix $\mathbf{\Phi}$, i.e. we can treat $\mathbf{\Phi}$ as being independent of $\mathbf{e}$. Now, the minimization of E becomes one of solving $\min_{\mathbf{x}} (\mathbf{A}\mathbf{x} - \mathbf{b})^T \mathbf{\Phi}(\mathbf{A}\mathbf{x} - \mathbf{b})$ for which the optimal estimate $\mathbf{x}$ is $(\mathbf{A}^T\mathbf{\Phi}\mathbf{A})^{-1}\mathbf{A}\mathbf{\Phi}\mathbf{b}$. Given this estimate of $\mathbf{x}$ we can in turn re-estimate $\mathbf{\Phi}$. Thus, we can alternate between estimating for $\mathbf{\Phi}$ (for fixed $\mathbf{x}$) and estimating $\mathbf{x}$ (for fixed $\mathbf{\Phi}$) till convergence. This method is known in the literature as *Iteratively Reweighted Least Squares* (IRLS) [10] and can be stated as Algorithm 2. We note that the IRLS method can be used to robustly solve for the Lie-algebraic update in Step 3 of Algorithm 1. While it provides a good solution, the IRLS method of Algorithm 2 is a greedy algorithm and needs a good initial guess for $\mathbf{x}$. Without a good initial guess, the intermediate weighting of $\mathbf{\Phi}$ will not be informative and the algorithm may not converge to a good final estimate. Since our **L1RA** method is efficient and provides a good estimate of $\mathbf{R}_{global}$ we use its output as the initial guess for robust rotation averaging using the IRLS estimator. We can now state our complete robust rotation averaging algorithm (denoted as **L1-IRLS**) in terms of the steps in Algorithm 1 as Algorithm 3.

We have denoted our complete algorithm as **L1-IRLS** to emphasise that both components are crucial for the estimation process. While the **L1RA** method is necessary to provide a good initial guess as it provides an efficient solution for robust estimation, the **IRLS** step is necessary to appropriately weight the information from individual $\mathbf{R}_{ij}$ estimates to give an efficient and accurate solution.

Algorithm 3 Robust Rotation Averaging (**L1-IRLS**)

```
L1RA step:
    • Initialise $\mathbf{R}_{global}$ to initial guess
    • Run Algorithm 1 by solving step 3 using Eqn. 7.
IRLS step:
    • Set $\mathbf{R}_{global}$ to output of L1RA method.
    • Run Algorithm 1 by solving step 3 using Algorithm 2.
```

3. Results

In this Section we compare the results of both our methods (**L1RA** and **L1-IRLS**) with DISCO [3] and the Weiszfeld method of [8] on two real world datasets of different sizes, i.e. Notre Dame and Quad. The Notre Dame dataset is available as raw images[4] and for the Quad dataset, relative rotations $\mathbf{R}_{ij}$ have been provided by the authors of [3][5]. In both cases, the results of bundle adjustment are provided with the datasets and serve as ground truth for all experiments in this Section. For the Notre Dame dataset, we estimate relative rotations by running two-frame bundle adjustment on image pairs using the `bundler` [13] software. In Table 1, we have also indicated the number of cameras ($\mathcal{V}$) and number of relative rotations $\mathbf{R}_{ij}$ ($\mathcal{E}$) for both datasets. These numbers correspond to the largest connected component $\mathcal{G}$ in the original viewgraph.

For all trials both our **L1RA** method and the Weiszfeld method of [8] are initialized by the same solution obtained using a randomly selected spanning tree. For the results in Table 1, both these methods terminate when the maximum change of rotations in $\mathbf{R}_{global}$ between iterations is less than a threshold of $\epsilon = 10^{-3}$. However, in the combined method of **L1-IRLS**, for obtaining an initialisation for the IRLS step, we need not use this termination criterion in the **L1RA** step. Instead, running the **L1RA** method for 5 iterations is sufficient to bring the estimate of $\mathbf{R}_{global}$ within the basin of convergence of the IRLS step. Consequently, since the IRLS iterations are much faster than those of **L1RA**, the combined **L1-IRLS** method is both very fast and highly accurate.

While we have implemented our proposed methods and the Weiszfeld method of [8], the results reported for DISCO

[4] `http://phototour.cs.washington.edu/datasets`

[5] `http://vision.soic.indiana.edu/disco`

# Cameras	# Relative Rotations	Median Rotation Error in degrees				Computational Time in seconds			
$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	DISCO [3]	Weiszfeld [8]	Our Methods		DISCO from [3]	Weiszfeld [8]	Our Methods	
				L1RA	L1-IRLS			L1RA	L1-IRLS
NOTRE DAME									
715	64678	NA	0.50	0.76	**0.54**	NA	40	6.14	**10**
QUAD									
5530	222044	5.02	6.95	4.16	**1.97**	2983 (49m 43s)	5707 (1h 35m 12s) 1271 iters	569 (8m 43s) 24 iters	**232** **(3m 52s)**

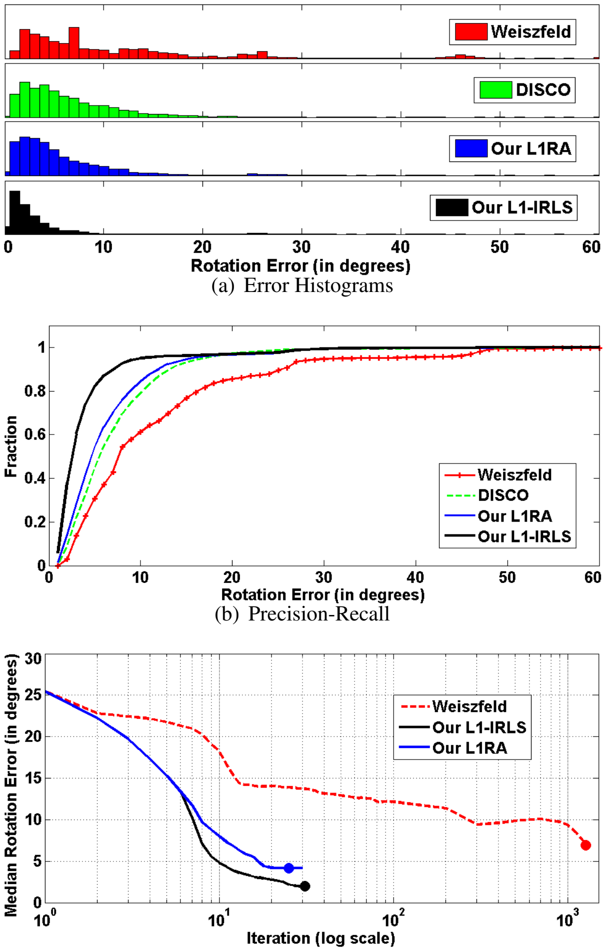
Table 1. Comparison of our methods with DISCO [3] and Weiszfeld [8] methods. The columns denoted as **L1-IRLS** implies the results for the combined method of using **L1RA** followed by **IRLS**. Our methods of **L1RA** and **L1-IRLS** significantly outperform both these methods on both accuracy and computational time. Note that the median error and computational time for DISCO are reproduced from [3] and datasets provided by the authors and that their results were obtained on a computer that is substantially more powerful than ours. [3] does not report results on the Notre Dame dataset (indicated as 'NA' or 'Not Available'). For the Quad dataset, the time is also indicated in hms format along with the number of iterations taken to achieve convergence.

are taken from the datasets provided. Since [3] does not report results on the Notre Dame dataset, the corresponding entries in Table 1 for DISCO are marked as 'not available' (NA). For the Quad dataset, for the results of DISCO [3] we use the aforementioned online dataset provided by the authors. Please note that this dataset is marginally different from the one used in [3] and consequently the corresponding error given in Table 1 differs from that reported in [3]. However, since the online dataset does not provide timing information, for computational time in Table 1 we report the time provided in [3]. We believe this to be reasonable as the difference between the two datasets is marginal.

Speed and Accuracy: In [8], the authors argue that their method "gives excellent results on large data sets". The only dataset considered in [8] is the Notre Dame dataset and as evident from Table 1, for this dataset the Weiszfeld method does marginally better than our methods although both of our methods are much faster. However, the true difference in performance becomes evident only when we consider datasets significantly larger than the Notre Dame dataset used in [8]. The Quad dataset has 5530 cameras, i.e. almost an order of magnitude larger in size than the Notre Dame dataset. Unlike the similar performance for the smaller datasets, for the Quad dataset, from Table 1 it will be immediately observed that both of our methods, i.e. **L1RA** and the combined method of **L1-IRLS** significantly outperform the discrete-continuous optimization of DISCO [3] as well as the Weiszfeld method of [8]. For the Quad dataset, the Weiszfeld method does poorly and has a high median error of 6.95° whereas the DISCO is somewhat better with a median error of 5.02°. In contrast, the median error for our **L1RA** step is 4.16° which is a significant improvement over both DISCO and the Weiszfeld method. This difference is also evident in the computational times involved. Our **L1RA** method provides the significantly more accurate solution in less than 9 minutes whereas the Weiszfeld algorithm of [8] takes more than 1.5 hours to provide a result with poor accuracy.

While, our **L1RA** method outperforms both DISCO and the Weiszfeld method, we note that our combined **L1-IRLS** algorithm is significantly superior both in terms of speed and accuracy. For the larger Quad data set, our **L1-IRLS** method has a very low median error of 1.97°, which is much more accurate than the current state-of-the-art results of [3]. Apart from a higher error than ours, DISCO is also computationally very expensive when compared with our method. Since we use a combination of ℓ_1 iterations followed by IRLS iterations, our combined method is very fast and converges to the final estimate in about 4 minutes! Here the IRLS step converges in 25 iterations in about 50% of the total time of 4 minutes. The remarkable performance of our combined **L1-IRLS** method is particularly evident when we note that our results were obtained using a `matlab` implementation on a desktop with a 2.67GHz processor, whereas the DISCO results of [3] were obtained using a multi-threaded implementation on a 16-core 3.0GHz machine. Even with this significant difference in computational power, for the Quad dataset our **L1-IRLS** averaging method is twice as accurate and is an order of magnitude faster. On a true basis of comparison of computational time, this ratio will be even higher.

Error Distribution: Although we have used the median error as our basis of comparison in Table 1, for a more detailed analysis of relative performance we need to use statistics instead of a single number. In Fig. 1 we present further evidence to demonstrate the superior performance



(a) Error Histograms

(b) Precision-Recall

(c) Convergence Rate

Figure 1. Comparison of our methods with DISCO [3] and Weiszfeld method [8] on the Quad dataset. (a) shows the histograms of errors of individual camera rotation estimates for different methods; (b) represents the fraction of errors below a given level, i.e. precision-recall curves where the methods with higher curves are better; (c) plots the median error vs. iterations for the Weiszfeld method and our methods. The convergence points are indicated with a circle. Note that the scale for the iterations (x-axis) is logarithmic and the number of iterations does not reflect the time taken for the different methods. Please view this figure in colour.

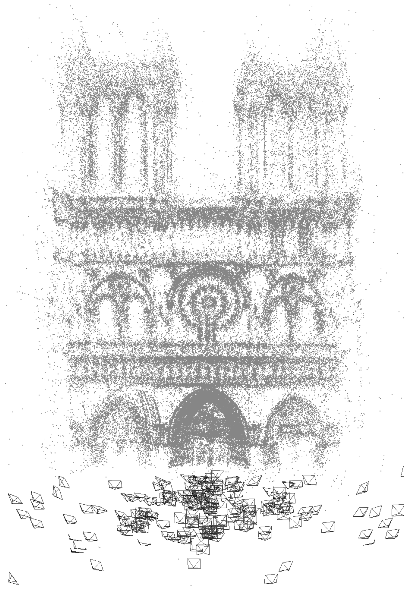


Figure 2. Reconstruction of the Notre Dame dataset using our robust rotation average results.

of our method over both DISCO and the Weiszfeld method. All plots in Fig. 1 pertain to the experiments reported in Table 1 for our largest dataset, i.e. Quad. In Fig. 1(a) we compare the statistical distribution of errors for all four methods. The comparative performance in terms of median error is better represented here as we can observe that the Weiszfeld method of [8] does poorly with a widely spread distribution of error values. It will also be noticed that apart from a large spread of errors, the Weiszfeld method also has two minor peaks in the distribution for very high rotation errors. This clearly indicates that the Weiszfeld method is inferior in comparison with both DISCO and our methods. Compared to these two methods, we note that the error distributions for both our methods are better, especially so for **L1-IRLS** as its distribution is shifted leftward to a significant degree, implying much lower errors on the average. This fact is also well summarized in Fig. 1(b) that is similar to a precision-recall curve. For each method, for a given level of rotation error (θ), we plot the fraction of estimated camera rotations that have an error less than θ. As in precision-recall curves, the better method has a higher curve implying that a larger number of rotation estimates have better accuracy than a given error bound of θ. As is evident from Fig. 1(b), both our methods are better than DISCO and the Weiszfeld method. In particular, our **L1-IRLS** method is significantly superior to these methods. One measure that summarizes the performance of different methods is the area under each curve which is normalized to 1 for a perfect precision-recall curve, i.e. when there are no errors. The normalized areas for Weiszfeld [8], DISCO [3], our **L1RA** and **L1-IRLS** are 0.77, 0.86, 0.88 and 0.92 respectively.

Convergence Rate: In Fig. 1(c) we show the convergence behavior of the Weiszfeld method of [8] and our methods by plotting the median error as a function of the iteration number. Note that the scale for the iterations is logarithmic as the Weiszfeld method takes 1271 iterations to converge while our methods converge in far fewer iterations. In Fig. 1(c), the iterations at which all the methods meet the termination criterion are indicated by dots on the respective convergence curves. It should also be noted that we have plotted the error vs. iteration number to characterise the behaviour of the individual methods in a platform independent manner. In particular, it should be noted that while the **L1-IRLS** takes more iterations than our **L1RA** method, **L1-IRLS** is the faster of the two methods. Evidently, as indicated by the steeper slope on a logarithmic scale, both of our methods converge much faster than the Weiszfeld method of [8] while achieving much higher accuracy. The comparative behaviour of the convergence curves is demonstrable evidence that the Weiszfeld update is unsuitable for large graphs and we should use a joint update of all rotations as is done by our methods.

3D Reconstruction: Finally, as an illustration of the correctness of our robust rotation averaging, we carry out a full 3D reconstruction of the Notre Dame dataset using our rotation estimate. When the rotation is fixed, estimating the 3D structure as well as 3D camera translations can be robustly solved using an SOCP optimization [11], i.e. without any computationally expensive bundle adjustment. The resulting reconstruction is shown in Fig. 2.

4. Discussion

We can now derive some conclusions from our experiments. For large-scale datasets, even a small improvement in rotation estimation has major implications for the speed and accuracy of downstream processing in the SfM pipeline. We also remark here that the inferior quality of DISCO's result is due to its converting a geometric problem into a complicated discrete Bayesian inference problem. On a different note, despite using the geometry of $SO(3)$, the Weiszfeld method of [8] suffers from the bottleneck of carrying out individual updates to solve a global problem. This makes the Weiszfeld method particularly slow and inaccurate for large graphs. In contrast to both these methods, our approach has two crucial attributes. We utilise both the geometric structure of the $SO(3)$ Lie group *and* also carry out a joint update for all rotations simultaneously. By exploiting recent advances in ℓ_1 optimization in the **L1RA** approach, we are able to efficiently solve the relative rotation averaging problem in the presence of outliers. In addition to being efficient and accurate, our **L1RA** method is very robust to the presence of outliers. The speed and accuracy of our **L1-IRLS** is striking given its simplicity. The lesson we draw here is that, given a good initialisation, a weighted least squares approach can effectively solve the problem of robust estimation. However, we do emphasise that the remarkable accuracy of the **L1-IRLS** method cannot be achieved without the powerful capability of the ℓ_1 approaches to provide an accurate initial estimate with a few iterations.

5. Conclusion

In summary, we have developed a robust method for averaging relative rotations that outperforms the state-of-the-art DISCO method of [3] as well as the Weiszfeld method of [8]. Apart from joint estimation on the $SO(3)$ group, a judicious mix of ℓ_1 and IRLS steps can yield a method that is efficient, accurate and scales well with the problem size. We believe that the comparative performance on larger datasets would be even more favorable for our method.

References

[1] E. Candes and J. Romberg. L1-magic : recovery of sparse signals via convex programming. http://users.ece.gatech.edu/~justin/l1magic.

[2] E. Candes and T. Tao. Decoding by linear programming. IEEE Transactions on Information Theory, 51(12), 2005.

[3] D. J. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In CVPR, 2011.

[4] J. Fredriksson and C. Olsson. Simultaneous multiple rotation averaging using lagrangian duality. In ACCV, 2012.

[5] V. M. Govindu. Combining two-view constraints for motion estimation. In CVPR, 2001.

[6] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In CVPR, 2004.

[7] V. M. Govindu. Robustness in motion averaging. In ACCV, 2006.

[8] R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the weiszfeld algorithm. In CVPR, 2011.

[9] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. IJCV, 103(3), 2013.

[10] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. Commun. Statist. Theory Methods, 9, 1977.

[11] C. Olsson and O. Enqvist. Stable structure from motion for unordered image collections. In SCIA, 2011.

[12] S. N. Sinha, D. Steedly, and R. Szeliski. A multi-stage linear approach to structure from motion. In RMLE 2010 Workshop, 2010.

[13] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In SIGGRAPH, 2006.

[14] R. Szeliski. Computer Vision: Algorithms and Applications. Springer-Verlag, 2010.

[15] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In Vision Algorithms: Theory and Practice, 2000.

[16] R. Tron and R. Vidal. Distributed computer vision algorithms through distributed averaging. In CVPR, 2011.

[17] R. Tron, R. Vidal, and A. Terzis. Distributed pose averaging in camera networks via consensus on SE(3). In ICDSC, 2008.

[18] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In CVPR, 2010.