L1 rotation averaging using the Weiszfeld algorithm

Richard Hartley, Khurrum Aftab
Australian National University
Canberra, and National ICT Australia *
Richard.Hartley@anu.edu.au

Jochen Trumpf
Australian National University
Canberra

Abstract

We consider the problem of rotation averaging under the L_1 norm. This problem is related to the classic Fermat-Weber problem for finding the geometric median of a set of points in $\mathbb{R}^n$. We apply the classical Weiszfeld algorithm to this problem, adapting it iteratively in tangent spaces of $\mathrm{SO}(3)$ to obtain a provably convergent algorithm for finding the L_1 mean. This results in an extremely simple and rapid averaging algorithm, without the need for line search. The choice of L_1 mean (also called geometric median) is motivated by its greater robustness compared with rotation averaging under the L_2 norm (the usual averaging process).

We apply this problem to both single-rotation averaging (under which the algorithm provably finds the global L_1 optimum) and multiple rotation averaging (for which no such proof exists). The algorithm is demonstrated to give markedly improved results, compared with L_2 averaging. We achieve a median rotation error of 0.82 degrees on the 595 images of the Notre Dame image set.

1. Introduction

We consider the problem of rotation averaging. The problem takes two forms: single rotation averaging in which several estimates are obtained of a single rotation, which are then averaged to give the best estimate; and multiple rotation averaging, in which relative rotations $\mathtt{R}_{ij}$ are given, and absolute rotations $\mathtt{R}_i$ are computed to satisfy the compatibility constraint $\mathtt{R}_{ij}\mathtt{R}_i = \mathtt{R}_j$.

The problem has significant applications to structure and motion [18, 24, 10, 13, 22, 14] and to non-overlapping camera calibration [4]. It has been studied quite extensively in the past, both in computer vision and in other fields. Significant work in this area includes the work of Govindu [8, 7] and Pajdla [18].

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

We are not aware of any work from outside the vision field in the problem of multiple rotation averaging. However, significant contributions to the single rotation averaging problem have been made by [19, 17, 23] and others. This work belongs to the research field of optimization on manifolds [1].

Most significant from our point of view is the work reported in [6, 30, 2] which considers a version of the Weiszfeld algorithm on classes of Riemannian manifolds, proving convergence theorems in a broad context, which relate directly to our algorithm, and suffice to prove its convergence.

The Weiszfeld algorithm is a well-known algorithm for finding the L_1 mean of a set of points in $\mathbb{R}^n$. Given points $\mathbf{x}_i \in \mathbb{R}^n$, their L_1 mean (more commonly called their geometric median) is the point $\mathbf{y}$ that minimizes $\sum_{i=1}^n \|\mathbf{y} - \mathbf{x}_i\|$, where $\|\cdot\|$ is the Euclidean norm. This problem has been much studied. The simplest, provably convergent algorithm is due to Weiszfeld [28].

Other refinements to the basic algorithm include geometric speed-up methods [21] and Newton methods [16]. However, the simplicity of the basic Weiszfeld algorithm and the rapidity with which its iterative update may be computed make it attractive, and we have used it as our preferred method. The Weiszfeld algorithm may also be generalized to Banach spaces [5] and to Riemannian manifolds [6, 30]. This last case is of relevance to the problem of computing the L_1 geodesic mean on $\mathrm{SO}(3)$.

2. Problem formulation and notation

There are three metrics commonly used for distance measurement in the rotation group $\mathrm{SO}(3)$. These are

1. The geodesic or angle metric $\theta = d_\angle(\mathtt{R}, \mathtt{S})$, which is the angle of the rotation $\mathtt{R}\mathtt{S}^{-1}$.
2. The chordal metric

$$d_{\mathrm{chord}}(\mathtt{R}, \mathtt{S}) = \|\mathtt{R} - \mathtt{S}\|_F = 2\sqrt{2}\sin(\theta/2)$$

where $\|\cdot\|_F$ represents the Frobenius norm.

3. The quaternion metric

$$d_{\text{quat}}(\mathtt{R},\mathtt{S}) = \min(\|\mathbf{r}-\mathbf{s}\|, \|\mathbf{r}+\mathbf{s}\|) = 2\sin(\theta/4)$$

where $\mathbf{r}$ and $\mathbf{s}$ are quaternion representations of the matrices $\mathtt{R}$ and $\mathtt{S}$ respectively, and the norm $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^4$.

All these metrics are bi-invariant, in that they satisfy the condition $d(\mathtt{R},\mathtt{S}) = d(\mathtt{TR},\mathtt{TS}) = d(\mathtt{RT},\mathtt{ST})$ for any rotation $\mathtt{T}$. For small values of $\theta = d_{\angle}(\mathtt{R},\mathtt{S})$ the metrics are the same, to first order, except for a scale factor.

The (single rotation) averaging problem is as follows. Given rotations $\mathtt{R}_i \in \mathrm{SO}(3)$, the L_p mean is equal to

$$\mathtt{S}^* = \operatorname*{argmin}_{\mathtt{S}\in\mathrm{SO}(3)} \sum_{i=1}^{n} d(\mathtt{R}_i,\mathtt{S})^p \ .$$

The L_1 and L_2 means are the two most useful or common cases. Although L_2 averaging has been considered extensively, the L_1 averaging problem has been relatively unexplored. A gradient-descent algorithm using line-search in the tangent space was given in [4]. However, line-search is costly and cumbersome to implement. In addition, no proof of convergence was given in that paper. In this paper, we present an extremely simple geodesic L_1 averaging algorithm for $\mathrm{SO}(3)$, based on the Weiszfeld algorithm for the classic Fermat-Weber problem in $\mathbb{R}^n$. The Weiszfeld algorithm is provably convergent to the global minimum in $\mathbb{R}^n$, and this result carries over to our averaging algorithm in $\mathrm{SO}(3)$, and indeed more generally, as follows from results in [6, 30, 2]. Thus, we have an extremly simple algorithm with guaranteed convergence under simple conditions.

Each of the three bi-invariant metrics presented above has its place in rotation averaging, and different algorithms naturally minimize different distances.

3. L_2 averaging

The rotation averaging problem on $\mathrm{SO}(3)$ under the L_2 norm may be solved in closed form for the chordal and quaternion metrics, under appropriate favourable conditions. There is no closed-form algorithm for L_2 rotation averaging under the geodesic metric, but convergent algorithms have been proposed [17, 4]. We give a brief description of some of these L_2 averaging algorithms, since they may be used for initialization of the L_1 averaging method that is the main topic of this paper.

3.1. Minimization under the chordal metric

Given a set of rotations $\mathtt{R}_i$, the L_2 chordal mean is the rotation $\mathtt{S}$ that minimizes

$$\sum_{i=1}^{n} d_{\text{chord}}(\mathtt{R}_i,\mathtt{S})^2 = \sum_{i=1}^{n} \|\mathtt{R}_i - \mathtt{S}\|_F^2 \qquad (1)$$
$$= \sum_{i=1}^{n} 8\sin^2(\theta_i/2)$$

where $\theta_i = d_{\angle}(\mathtt{R}_i,\mathtt{S})$.

It turns out that there are two different, but equivalent closed form algorithms for finding this minimum. Let $\hat{\mathtt{S}} = \sum_{i=1}^{n} \mathtt{R}_i$. The matrix $\hat{\mathtt{S}}$ is not a rotation matrix, however if we set

$$\mathtt{S}^* = \operatorname*{argmin}_{\mathtt{S}\in\mathrm{SO}(3)} \|\mathtt{S}-\hat{\mathtt{S}}\| = \operatorname*{argmin}_{\mathtt{S}\in\mathrm{SO}(3)} \|\mathtt{S} - \sum_{i=1}^{n} \mathtt{R}_i\|$$

then $\mathtt{S}^*$ is the chordal L_2 mean, minimizing the cost (1). The closest rotation matrix to $\hat{\mathtt{S}}$ may be computed from its singular Value Decomposition (SVD). Let $\hat{\mathtt{S}} = \mathtt{U}\,\mathtt{D}\,\mathtt{V}^\top$ where the diagonal elements of $\mathtt{D}$ are arranged in descending order. If $\det(\mathtt{U}\mathtt{V}^\top) \geq 0$, then set $\mathtt{S} = \mathtt{U}\mathtt{V}^\top$. Otherwise set $\mathtt{S} = \mathtt{U}\,\mathrm{diag}(1,1,-1)\mathtt{V}^\top$. The matrix $\mathtt{S}$ so obtained is the closest rotation to $\hat{\mathtt{S}}$ and hence the required rotation minimizing (1). This algorithm was proposed in [19, 23].

A second algorithm surprisingly uses quaternions to minimize this same metric. Let $\mathbf{r}_i$ be the quaternion representation of the rotation $\mathtt{R}_i$. Both $\mathbf{r}_i$ and $-\mathbf{r}_i$ represent the same rotation, but we may choose either one. We form the 4×4 matrix $\mathtt{A} = \sum_{i=1}^{n} \mathbf{r}_i\mathbf{r}_i{}^\top$, and observe that this does not depend on which of $\mathbf{r}_i$ or $-\mathbf{r}_i$ is chosen. Let $\mathbf{s}$ be the unit eigenvector corresponding to the maximum eigenvalue of $\mathtt{A}$, and let $\mathtt{S}$ be the corresponding rotation matrix. Then it may be verified without great difficulty that $\mathtt{S}$ minimizes the cost (1), and hence is the L_2 chordal mean of the $\mathtt{R}_i$. This is a curious result, in that using quaternions, we nevertheless minimize the chordal mean.

Thus, it is possible to find the global minimizer of (1) under all circumstances.

3.2. Minimization under the quaternion metric

Consider a set of rotations $\mathtt{R}_i$ and a chosen quaternion representation $\mathbf{r}_i$ for each rotation. As a first cut at an algorithm for finding the quaternion mean, we might propose forming the sum $\hat{\mathbf{s}} = \sum_{i=1}^{n} \mathbf{r}_i$ and then normalizing $\hat{\mathbf{s}}$ by setting $\mathbf{s} = \hat{\mathbf{s}}/\|\hat{\mathbf{s}}\|$. The problem with this algorithm is that the choice of which quaternion $\mathbf{r}_i$ or $-\mathbf{r}_i$ to use is not clear, and does affect the result. Nevertheless, a well-specified choice is possible, and leads to the correct result, if the rotations are not too far spread, as stated in the following theorem of Dai *et al*. [4].

Theorem 3.1. *Let* $\mathtt{R}_i$ *be rotations satisfying* $d_\angle(\mathtt{R}_i, \mathtt{S}) < \pi/2$ *for some rotation* $\mathtt{S}$ *and for all* i*. Let* $\mathbf{s}$ *be a quaternion representation of* $\mathtt{S}$ *and let* $\mathbf{r}_i$ *be the quaternion representation of* $\mathtt{R}_i$ *chosen with sign such that* $\|\mathbf{r}_i - \mathbf{s}\|_2 \leq \|\mathbf{r}_i + \mathbf{s}\|_2$*. Then the quaternion* L_2 *mean of the rotations* $\mathtt{R}_i$ *is represented by the quaternion* $\bar{\mathbf{r}}/\|\bar{\mathbf{r}}\|$*, where* $\bar{\mathbf{r}} = \sum_{i=1}^n \mathbf{r}_i$*.*

Thus, the quaternion L_2 mean is easily found, as long as the rotations all lie within a given ball of radius $\pi/2$ in $\mathrm{SO}(3)$. This restriction is quite mild, but it makes the algorithm somewhat less attractive than the algorithm to find the chordal mean, unless finding the quaternion mean is essential.

3.3. Minimization under the geodesic metric

The L_2 geodesic mean of a set of rotations $\mathtt{R}_i$ can only be computed iteratively. The mean is unique provided the given rotations $\mathtt{R}_1, \ldots, \mathtt{R}_n$ do not lie too far apart, more precisely if they lie in a closed ball of geodesic radius $\delta < \pi/2$ about some rotation $\mathtt{S}$, cf. Theorem 3.7 in [9]. For this case Manton [17] has provided the following convergent algorithm where the inner loop of the algorithm is computing the average in the tangent space and then projecting back onto the manifold $\mathrm{SO}(3)$ via the exponential map.

1: Set $\mathtt{R} := \mathtt{R}_1$. Choose a tolerance $\varepsilon > 0$.
2: **loop**
3: Compute $\mathbf{r} := \frac{1}{n}\sum_{i=1}^n \log\left(\mathtt{R}^\top \mathtt{R}_i\right)$.
4: **if** $\|\mathbf{r}\| < \varepsilon$ **then**
5: **return** $\mathtt{R}$
6: **end if**
7: Update $\mathtt{R} := \mathtt{R}\exp(\mathbf{r})$.
8: **end loop**

Algorithm 1: computing the L_2 geodesic mean on $\mathrm{SO}(3)$

In fact, this algorithm is shown to be an instance of simple Riemannian gradient descent. For a Newton-type algorithm to compute this mean see [15].

4. L_1 averaging

The L_1 mean of a set of points $\mathbf{x}_i$ in some metric space is the point $\bar{\mathbf{x}}$ that minimizes the sum of distances to the points. Thus,

$$\bar{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} \sum_{i=1}^n d(\mathbf{x}, \mathbf{x}_i) \ .$$

It is well understood that the L_1 mean of the points is generally more robust than the L_2 mean. For instance, if the points $\mathbf{x}_i$ are points on a line (points in $\mathbb{R}$) then the L_2 mean is the usual mean of the points, whereas the L_1 mean is the median. The median is less affected by the presence of distant (outlier) values. For points in $\mathbb{R}$, the median (L_1 mean) of a set of points may be found in linear time [3], but this is not true in higher dimensions.

The problem of finding the L_1 mean of a set of points in $\mathbb{R}^N$ for $N > 1$ is a classical problem, going back at least to Fermat. The special case of this problem for three points forming a triangle in $\mathbb{R}^2$ was solved by Torricelli. The solution is the so-called Fermat point of the triangle, provided no angle exceeds $120°$. The problem subsequently was studied in some detail by Weber [27]. For this reason, it is sometimes referred to as the Fermat-Weber problem or simply the Weber problem. It is also called the "location" problem. This latter name is related to its interpretation in terms of optimal placement of a factory to minimize the sum of its distances to a set of resources. The solution is commonly referred to as the geometric median of the points, which terminology we will use interchangeably with L_1 mean. A good description of the problem is given in the Wikipedia page on the geometric median [29].

Our interest in this problem relates to its robustness to outliers: "The geometric median has a breakdown point of 0.5. That is, up to half of the sample data may be arbitrarily corrupted, and the median of the samples will still provide a robust estimator for the location of the uncorrupted data." [29]

Unfortunately, there is generally no closed form solution for this problem. Nevertheless, a popular algorithm for solving the problem in $\mathbb{R}^N$ is the Weiszfeld algorithm [28], which will be described next for points $\mathbf{x}_i$ in $\mathbb{R}^N$.

The Weiszfeld algorithm may be considered as a gradient descent algorithm. The L_1 mean of a set of points in $\mathbb{R}^N$ may be written as the point $\bar{\mathbf{x}}$ that minimizes the cost function

$$C(\mathbf{y}) = \sum_{i=1}^n \|\mathbf{y} - \mathbf{x}_i\|. \tag{2}$$

Since each of the individual summands is a convex function, this cost is a convex function of $\mathbf{y}$. This being so, one is led to consider a gradient descent algorithm. This can be guaranteed to converge given a correct choice of step size, or by using line-search for the minimum along the downhill gradient direction.

It is easily computed that the gradient of the cost function (2) is equal to

$$\nabla_C = -\sum_{i=1}^n \frac{\mathbf{x}_i - \mathbf{y}}{\|\mathbf{x}_i - \mathbf{y}\|} \ . \tag{3}$$

Thus, the downhill gradient of the cost at a given point $\mathbf{y}$ is the sum of the unit vectors directed from $\mathbf{y}$ to each of the points $\mathbf{x}_i$. Given a current estimate $\mathbf{y}^t$, the next estimate of the minimum in a gradient descent algorithm is given by

$$y^{t+1} = y^t + \lambda \sum_{i=1}^n \frac{\mathbf{x}_i - \mathbf{y}^t}{\|\mathbf{x}_i - \mathbf{y}^t\|} \ ,$$

where $\lambda > 0$ is some value controlling the step-size along the gradient direction. The choice of step size in the Weiszfeld algorithm is set to be $\lambda = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{y}^t\|^{-1}$. Writing $s_i^t = \|\mathbf{x}_i - \mathbf{y}^t\|$, we then find that

$$\mathbf{y}^{t+1} = \mathbf{y}^t + \frac{\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{y}^t)/s_i^t}{\sum_{i=1}^{n} 1/s_i^t} \tag{4}$$

$$= \frac{\sum_{i=1}^{n} \mathbf{x}_i/s_i^t}{\sum_{i=1}^{n} 1/s_i^t} \ . \tag{5}$$

As long as the intermediate iterates $\mathbf{y}_i^t$ do not concide with any of the points $\mathbf{x}_i$, this algorithm will provably converge to the geometric median of the points [28]. Convergence may not be fast. If in fact $\mathbf{y}^t$ coincides with some point $\mathbf{x}_i$, then the algorithm will get stuck at this point. A simple strategy in this case is to displace the iterate $\mathbf{y}^t$ slightly and continue. It may easily be shown that successive iterates will "escape" from some point $\mathbf{x}_i$, not the minimum, by approximately doubling the distance at each iteration.

5. Geodesic median in $\mathrm{SO}(3)$

We now consider the problem of computing the L_1 geodesic mean in the group of rotations. We will refer to this as the geodesic median. To be able to apply the Weiszfeld algorithm, we transition back and forth between the rotation manifold, and its tangent space centred at the current estimate. Consider a rotation $\mathtt{R}$ in its angle-axis representation $\mathbf{v} = \theta\hat{\mathbf{v}}$ where $\hat{\mathbf{v}}$ is a unit vector, and $\mathtt{R}$ is a rotation through angle θ about the unit axis $\hat{\mathbf{v}}$. The correspondence $\mathtt{R} \leftrightarrow \mathbf{v}$ is the correspondence between the Riemannian rotation manifold and its 3-dimensional tangent space. Writing $[\mathbf{v}]_\times$ to be the 3×3 skew-symmetric matrix corresponding to $\mathbf{v}$, we have $\mathtt{R} = \exp([\mathbf{v}]_\times)$, and $\mathtt{R}$ may be computed from $\mathbf{v}$ using the Rodrigues formula [12]. We extend this notation by writing $\exp : \mathbb{R}^3 \to \mathrm{SO}(3)$ taking $\mathbf{v}$ to $\mathtt{R}$. The inverse map is $\log : \mathrm{SO}(3) \to \mathbb{R}^3$ taking $\mathtt{R}$ to $\mathbf{v}$ and where $\mathbf{v}$ is chosen such that $\|\mathbf{v}\| \leq \pi$.

Given a rotation $\mathtt{S}$, we may write $\log_{\mathtt{S}}(\mathtt{R}) = \log(\mathtt{R}\mathtt{S}^{-1})$ which gives a mapping from $\mathrm{SO}(3)$ to $\mathbb{R}^3$ centred at $\mathtt{S}$. An important property of this is that $\|\log_{\mathtt{S}}(\mathtt{R})\| = d_\angle(\mathtt{S}, \mathtt{R})$, the geodesic distance from $\mathtt{S}$ to $\mathtt{R}$. This expresses the fact that although the logarithm map is not distance-preserving, it does preserve distances from the identity; alternatively speaking, $\log_{\mathtt{S}}$, the logarithm centred at $\mathtt{S}$, preserves distances from $\mathtt{S}$.

The following lemma shows that we can use the geometric median in the tangent space to find the geodesic median in rotation space $\mathrm{SO}(3)$.

Lemma 5.2. *If $\mathtt{S}$ is the geodesic median of rotations $\mathtt{R}_i$, then the origin $\mathbf{0} \in \mathbb{R}^3$ is the geometric median of points $\log_{\mathtt{S}}(\mathtt{R}_i)$. Conversely, if $d_\angle(\mathtt{S}, \mathtt{R}_i) < \pi/2$ for all i, and $\mathbf{0}$ is the geometric median of the points $\log_{\mathtt{S}}(\mathtt{R}_i)$, then $\mathtt{S}$ is the geodesic median of the $\mathtt{R}_i$.*

We give a brief proof of this lemma. A complete proof will be published separately. The converse statement can be extended slightly to the case where all $\mathtt{R}_i$ and $\mathtt{S}$ lie in a convex set (appropriately defined) in $\mathrm{SO}(3)$, but this is much more difficult to prove and lies well beyond the scope of this paper.

Proof. If $\mathtt{S}$ is the geodesic median of the $\mathtt{R}_i$, then the gradient of the cost function, $C(\mathtt{S}) = \sum_{i=1}^{n} d_\angle(\mathtt{R}_i, \mathtt{S})$ is zero at $\mathtt{S}$. The gradient is defined as a vector in the tangent space at $\mathtt{S}$, which may be computed to be

$$\nabla_C = -\sum_{i=1}^{n} \frac{\log_{\mathtt{S}}(\mathtt{R}_i)}{\|\log_{\mathtt{S}}(\mathtt{R}_i)\|} \ .$$

Writing $\mathbf{r}_i = \log_{\mathtt{S}}(\mathtt{R}_i)$, we see that this gradient vector is also the gradient of the cost $C'(\mathbf{s}) = \sum_{i=1}^{n} \|\mathbf{r}_i - \mathbf{s}\|$ at $\mathbf{s} = \mathbf{0} \in \mathbb{R}^3$. In other words, $\mathbf{s} = \mathbf{0}$ is a critical point of C'. However, since this cost function is convex in $\mathbb{R}^3$, it follows that $\mathbf{s} = \mathbf{0}$ is a minimum of the cost function, and so $\mathbf{s} = \mathbf{0}$ is the geometric median of the $\mathbf{r}_i$ in $\mathbb{R}^3$.

On the other hand, if $\mathbf{s} = \mathbf{0}$ is the minimum of $C'(\mathbf{s})$, then by the same argument the gradient of C is zero at $\mathtt{S}$, so $\mathtt{S}$ is at least a critical point of the cost. However, it is not true that the cost function $C(\mathtt{S})$ is convex[1], or has a single minimum on the whole of $\mathrm{SO}(3)$. For this reason, we need the condition that all $\mathtt{R}_i$ lie in a ball of radius $\pi/2$ about $\mathtt{S}$. In this case, it may be shown that the cost $C(\mathtt{S})$ is convex on the ball and in fact the global minimum of $C(\cdot)$ lies within the ball, and hence at $\mathtt{S}$. The proof of this last statement is not entirely trivial, though it is intuitively plausible. A complete proof (in more generality) will be published separately.[2]

Given this lemma, we are led to propose an algorithm for finding the geodesic median in $\mathrm{SO}(3)$, based on the Weiszfeld algorithm on the tangent space. Given rotations $\mathtt{R}_i \in \mathrm{SO}(3)$, we proceed as follows.

1. Find an initial estimate $\mathtt{S}^0$ for the median. Such an estimate may already be know, or else we may take the L_2 mean of the rotations $\mathtt{R}_i$ as a starting point.

2. At any time $t = 0, 1, \ldots$ apply the logarithm map centred at $\mathtt{S}^t$ to compute $\mathbf{v}_i = \log_{\mathtt{S}^t}(\mathtt{R}_i)$.

3. (Weiszfeld step): Set

$$\delta = \frac{\sum_{i=1}^{n} \mathbf{v}_i/\|\mathbf{v}_i\|}{\sum_{i=1}^{n} 1/\|\mathbf{v}_i\|}$$

[1] A function is convex on a set in $\mathrm{SO}(3)$ if its restriction to a geodesic lying in the set is a convex function of arc length.

[2] For the case where all the $\mathtt{R}_i$ lie in a smaller ball of radius $\pi/4$ about $\mathtt{S}$, convexity of the cost $C(\mathtt{S})$ has been shown in [6], and in this case the global mean has been shown to lie within the ball in [2].

4. Set $\mathtt{S}^{t+1} = \exp(\delta)\mathtt{S}^t$.

5. Repeat steps 1 to 3 until convergence.

Thus, this algorithm may be thought of as carrying out successive iterative steps of the Weisfeld algorithm, each step taking place in the tangent space centred at the current estimate.

For computational efficiency, it is simpler to work with the quaternion representations $\mathbf{r}_i$ of the rotations $\mathtt{R}_i$, since mapping between quaternions and angle-axis representation is simpler than computing the exponential and logarithm maps. In addition quaternion multiplication is faster than matrix multiplication. Let Q be the unit quaternions. The mapping $q : \mathbb{R}^3 \to Q$ given by

$$q : \theta\hat{\mathbf{v}} \mapsto (\cos(\theta/2), \sin(\theta/2)\hat{\mathbf{v}})$$

maps between angle-axis and quaternion representation of a rotation. Then steps 2 to 4 of the above algorithm are replaced by

$$\begin{aligned} \theta_i\hat{\mathbf{v}}_i &= q^{-1}(\mathbf{r}_i \cdot \bar{\mathbf{s}}^t) \\ \delta &= \sum_{i=1}^{n} \hat{\mathbf{v}}_i \;/\; \sum_{i=1}^{n} 1/\theta_i \\ \mathbf{s}^{t+1} &= q(\delta) \cdot \mathbf{s}^t \end{aligned}$$

where $\bar{\mathbf{s}}^t$ represents the conjugate (inverse) of the quaternion $\mathbf{s}^t$. A further alternative is to use the Campbell-Baker-Hausdorff formula [8] to work entirely in angle-axis space, but this is essentially equivalent to the use of quaternions.

As is shown in lemma 5.2, a stationary point of this algorithm (for which $S^t = S^{t+1}$) must be the geodesic median of the rotations $\mathtt{R}_i$, provided that $d_\angle(\mathtt{R}^t, \mathtt{S}_i) < \pi/2$ for all i.

The Euclidean metric in a tangent space is related within constant bounds to the angular metric in $\mathrm{SO}(3)$, so it is plausible that this algorithm will converge. However, convergence of this algorithm follows in a more general context from the results of [6, 30, 2]. More precisely, it was shown in [6] that if all the $\mathtt{R}_i$ lie within a ball of radius $\pi/4$, the above algorithm converges to the so-called *solipsistic mean* (the minimum of the cost function within the given ball) provided that (1) not all the $\mathtt{R}_i$ lie on a single geodesic, and (2) the algorithm does not step outside that ball. Restriction (2) can be overcome using step size control and projection techniques [30]. Finally, it was shown in [2] that the solipsistic mean is in fact the global mean if all the $\mathtt{R}_i$ lie within a ball of radius $\pi/4$.

6. L_1 averaging multiple rotations

We now consider the problem of rotation averaging of a set of relative rotations. More specifically, let $\mathtt{R}_i$; $i = 1, \ldots, M$ be a set of rotations denoting the orientation of different coordinate frames in $\mathbb{R}^3$. The rotations are assumed unknown, but a set of relative rotations $\mathtt{R}_{ij}$ are given, for pairs $(i, j) \in \mathcal{N}$, where $\mathcal{N}$ is a subset of all index pairs. If $(i, j) \in \mathcal{N}$, then also $(j, i) \in \mathcal{N}$, and $\mathtt{R}_{ji} = \mathtt{R}_{ij}^{-1}$. These relative rotation matrices $\mathtt{R}_{ij}$ are provided by some measurement process and are assumed to be corrupted by some degree of noise. The required task is to find the absolute rotations $\mathtt{R}_i, \mathtt{R}_j$ such that $\mathtt{R}_{ij} = \mathtt{R}_j\mathtt{R}_i^{-1}$ for all pairs $(i, j) \in \mathcal{N}$. Of course, since this condition can not be fulfilled exactly, given noisy measurements $\mathtt{R}_{ij}$, so the task is to minimize the cost

$$C(\mathtt{R}_1, \ldots, \mathtt{R}_M) = \sum_{(i,j)\in\mathcal{N}} d(\mathtt{R}_{ij}\mathtt{R}_i, \mathtt{R}_j)^p$$

where $p = 1$ or 2. We consider the geodesic distance function $d(\cdot, \cdot) = d_\angle(\cdot, \cdot)$. We may eliminate the obvious gauge freedom (ambiguity of solution) by setting any one of the rotations $\mathtt{R}_i$ to the identity. Generally, minimizing this cost is a difficult problem because of the existence of local minima, but in practice it may be solved in many circumstances with more-or-less acceptable results. In this paper we will consider the L_1 averaging problem, and demonstrate an algorithm that gives excellent results on large data sets.

Our approach is by successive L_1 averaging to estimate each $\mathtt{R}_i$ in turn, given its neighbours. At any given point during the computation, a rotation $\mathtt{R}_i$ will have an estimated value, and so will its neighbors $\mathtt{R}_j$, for $(i, j) \in \mathcal{N}$. Therefore, we may compute estimates $\mathtt{R}_i^{(j)} = \mathtt{R}_{ji}\mathtt{R}_j$, where the superscript (j) indicates that this is the estimate of $\mathtt{R}_i$ derived from its neighbour $\mathtt{R}_j$. We then use our Weiszfeld L_1 averaging method on $\mathrm{SO}(3)$ to compute a new estimate for $\mathtt{R}_i$ by averaging the estimates $\mathtt{R}_i^{(j)}$. In one pass of the algorithm, each $\mathtt{R}_i$ is re-estimated in turn, in some order. Multiple passes of the algorithm are required for convergence.

Since the Weiszfeld algorithm on $\mathrm{SO}(3)$ is itself an iterative algorithm, we have the choice of running the Weiszfeld algorithm to convergence, each time we re-estimate $\mathtt{R}_i$, or else running it for a limited number of iterations leaving the convergence incomplete, and passing on to the next rotation. To avoid nested interation, we choose to run a single iteration of the Weiszfeld algorithm at each step. The complete algorithm is as follows.

1. **Initialization:** Set some node $\mathtt{R}_{i_0}$ with the maximum number of neighbours to the identity rotation, and construct a spanning tree in the neighbourhood graph rooted at $\mathtt{R}_{i_0}$. Estimate the rotations $\mathtt{R}_j$ at each other node in the tree by propagating away from the root using the relation $\mathtt{R}_j = \mathtt{R}_{ij}\mathtt{R}_i$.

2. **Sweep:** For each i in turn, re-estimate the rotation $\mathtt{R}_i$ using one iteration of the Weiszfeld algorithm. (As

each new $\mathtt{R}_i$ is computed, it is used in the computations of the other $\mathtt{R}_i$ during the same sweep.)

3. **Iterate:** Repeat this last step a fixed number of times, or until convergence.

The whole computation is most conveniently carried out using quaternions.

Unlike the single rotation averaging problem considered in section 5 we can not guarantee convergence of this algorithm to a global minimum, but results will demonstrate good performance.

7. Experiments and Applications

We demonstrate the utility and accuracy of the L_1 rotation averaging methods by applying them to a large-scale reconstruction problem, based on the Notre Dame data set [25]. This set has been reconstructed and bundle-adjusted, resulting in estimates of all the camera matrices, which we take to represent ground truth. The set consists of 595 images of $277,887$ points. There exist $42,621$ pairs of images with more than 30 corresponding point pairs, and these were the pairs of images that we used in our tests.

7.1. Single rotation averaging

To test the algorithm for estimating a single rotation from several estimates, we carried out the following procedure.

1. Subsets of five point pairs were chosen and a fast five-point algorithm [20] was used to estimate the essential matrix from the pair of images, and from this the relative rotation and translation were computed. Only those solutions were retained that satisfied the cheirality constraint that all 5 points lie in front of both estimated cameras. This can be done extremely quickly – in our implementation about $35\mu s$ per 5-point sample.

2. The solutions were tested against 3 further points and only solutions which fitted well against these points were retained.

3. From several subsets of 5 points we obtained several estimates of the relative rotation (a subset can lead to more than one rotation estimate).

4. The rotation estimates were then averaged to find their L_1 mean. L_2 rotation averaging was used to find an initial estimate, followed by application of some steps of the Weiszfeld algorithm.

This method was compared with straight L_2 rotation averaging; the L_1 averaging technique gave significantly better results. In addition, the results were compared with those obtained by using non-minimal methods based on the 8-point algorithm, followed by algebraic error or Sampson error minimization, and calibrated bundle adjustment [12].

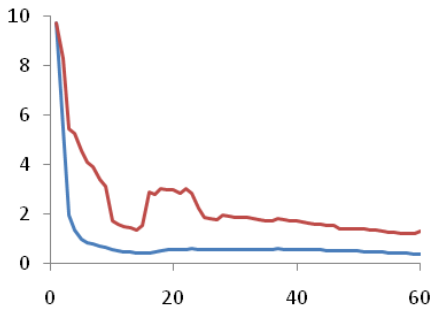

Figure 1. *The graph shows the result of L_2 (top curve) and L_1 (bottom curve) rotation averaging, used in computing the relative orientation of two cameras from repeatedly applying the 5-point algorithm to estimate relative rotation. The plots show the error with respect to ground truth as a function of the number of samples taken. As can be seen, the L_1 algorithm converges in this case to close to ground truth with about 10 samples.*

It is possible that this averaging technique can be used as an alternative to RANSAC in the case of noisy point correspondences, but we emphasize that this was not the purpose of this experiment. Rather, the point was to demonstrate the advantage of L_1 rotation averaging, and investigate it as a means for computing two-view relative pose.

Results We carried out experiments in which the relative rotation of two cameras was computed using the 5-point algorithm, followed by averaging the rotation results from many rotation samples computed in this way. In all cases, the L_1 averaging algorithm worked significantly better. In Fig 1 is shown a typical result of this estimation procedure, comparing L_1 with L_2 averaging algorithms, for increasing numbers of rotations.

7.2. Multiple rotation averaging

The results of pairwise rotation estimates obtained in the previous section were then used as input to the multiple rotation averaging algorithm described in section 6.

In carrying out this test, the two-view relative rotation estimates were obtained using several techniques. Generally speaking, more elaborate methods of computing relative rotation led to better results, but the fast methods were shown to give surprisingly good results very fast. The following methods were used for finding pairwise relative rotations $\mathtt{R}_{ij}$.

1. **E-5pt**(m, n)**:** Rotations were obtained from essential matrices computed from m minimal 5-point sets, then averaged using the L_2-chordal algorithm, followed by n steps of L_1 averaging using the Weiszfeld algorithm.

2. **E-algebraic:** The algebraic cost $\sum_i (\mathbf{x}_i'^{\top} \mathbf{E} \mathbf{x}_i)^2$ was minimized iteratively over the space of all valid essential matrices. This is an adaptation of the method of [11] to essential matrices, and is very efficient and fast.

3. **E-Sampson:** The Sampson error

$$\sum_i \frac{(\mathbf{x}_i'^{\top} \mathbf{E} \mathbf{x}_i)^2}{(\mathbf{E}\mathbf{x})_1^2 + (\mathbf{E}\mathbf{x})_2^2 + (\mathbf{E}^{\top}\mathbf{x}')_1^2 + (\mathbf{E}^{\top}\mathbf{x}')_2^2}$$

was minimized over the space of essential matrices.

4. **E-bundled:** Full 2-view bundle adjustment was carried out, initialized by the results of E-algebraic. This method was expected to give the best results (and it did), but requires substantially more computational effort (cf. Tab 1).

Given the diversity of image-pair configurations, possible small overlap and general instability, no one method gave perfectly accurate relative rotation estimates for all $42,621$ image pairs. However, in all cases the resulting rotation errors for the 595 cameras were quite accurate. For the E-bundled method, the median camera orientation error was 0.82 degrees.

7.2.1 Detailed results

The results of rotation averaging on the Notre Dame data set [26] are given in Fig 2 and Fig 3. Pairs of images from this set were chosen if they shared more that 30 points in common (42,621 such pairs). From these pairs, the essential matrix was computed using various different methods as described above.

Method	per-pair time(msec)	total time(sec)	L1	L2
E-bundled	281	11932	0.82	0.93
E-algebraic	4.07	173	1.21	1.84
E-Sampson	19	839	1.05	1.85
E-5pt(30,20)	7	296	0.98	1.32
E-5pt(20,10)	4	168	0.93	1.46
L1-averaging	–	36		
L2-averaging	–	10		

Table 1. Timing (on a 2.6 GHz laptop) for the computation of the $42,621$ essential matrices using various methods, and also the time taken for L_1 and L_2 averaging over all nodes. This last operation is carried out once only. Columns 2 and 3 show the time per iteration, and total time. The last two columns give the median (over 595 views) rotation error in degrees for L_1 and L_2 averaging. As may be seen, the full bundle adjustment takes a lot more time, though it does lead to slightly better results. We do not count time taken for finding the pairs of overlapping images with sufficiently many matches. Observe that E-5pt(30,20) did better than E-5pt(20,10) for L_1 averaging, but this was by chance.

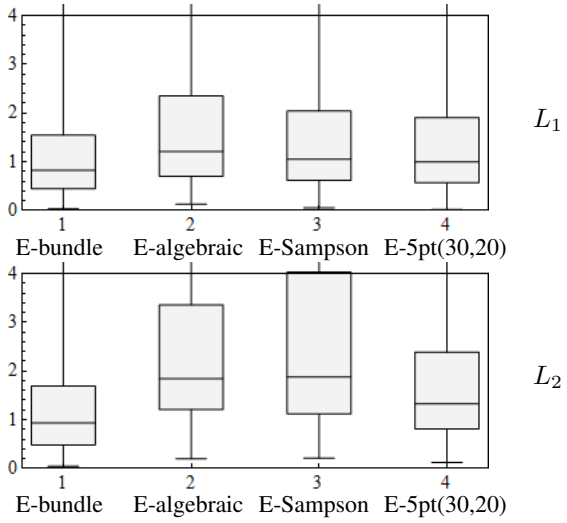

Figure 2. *Whisker plots of the absolute orientation accuracy of the* 595 *images of the Notre Dame data set. The top and bottom of the boxes represent the 25% and 75% marks. The upper graph shows the result of* L_1 *averaging and the lower graph the* L_2 *averaging results. In each graph are shown the results arising from different methods of computing the essential matrices, and hence the rotations.*

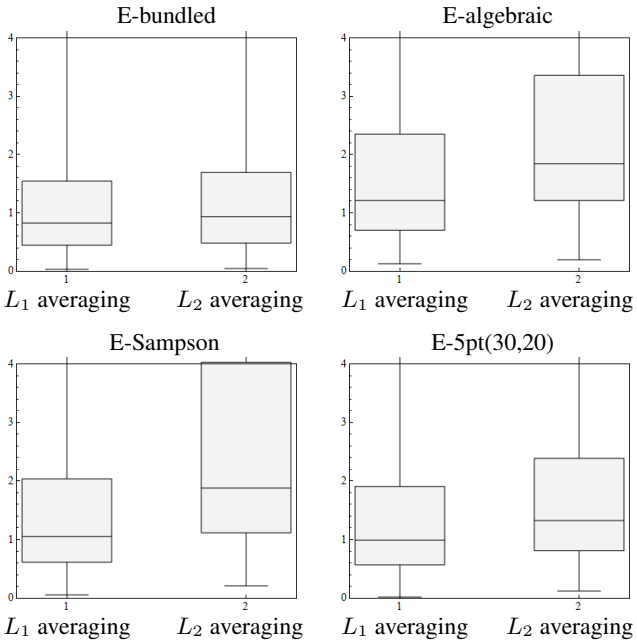

Figure 3. *Side-by-side comparison of the results of* L_1 *and* L_2 *averaging for each of the four methods of computing relative rotations.*

8. Conclusion

The Weiszfeld-based L_1 averaging method gives good results, both for single-view averaging of minimal-case rotation estimates, and iteratively for multiple-view recon-

struction. It is possible to get very good rotation estimates very quickly (3 minutes for the Notre Dame set) with a median accuracy of about one degree. This makes the method suitable as an initialization method for translation estimation and final bundle adjustment.

The L_1 algorithm given here is substantially more simple than the gradient-descent line-search algorithm proposed in [4]. Our experiments strongly confirm the observation of that paper that L_1 averaging gives superior and more robust results than L_2 averaging, and still at very competitive cost. In fact, the time taken for averaging is far smaller than the time required to generate the individual rotation estimates.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[2] B. Afsari. Riemannian L^p center of mass: Existence, uniqueness, and convexity. *Proceedings of the American Mathematical Society*, 139:655–673, 2011.

[3] M. Blum, R. Floyd, V. Pratt, R. Rivest, and R. Tarjan. Time bounds for selection. *J. Comput. System Sci*, 7:448–461, 1973.

[4] Y. Dai, J. Trumpf, H. Li, N. Barnes, and R. Hartley. Rotation averaging with application to camera-rig calibration. In *Proceedings of the ninth Asian Conference on Computer Vision (ACCV), Part II*, pages 335–346, 2009.

[5] U. Eckhardt. Weber's problem and Weiszfeld's algorithm in general spaces. *Mathematical Programming*, 18(1):186–196, 1980.

[6] P. T. Fletcher, S. Venkatasubramanian, and S. Joshi. The geometric median on riemannian manifolds with applications to robust atlas estimation. *Neuroimage*, 45 (1 Suppl):143–152, 2009.

[7] V. M. Govindu. Combining two-view constraints for motion estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 218–225, 2001.

[8] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 684–691, 2004.

[9] K. Grove, H. Karcher, and E. A. Ruh. Jacobi fields and Finsler metrics on compact Lie groups with an application to differentiable pinching problems. *Math. Ann.*, 211:7–21, 1974.

[10] R. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Washington DC*, pages I–504–509, June 2004.

[11] R. I. Hartley. Minimizing algebraic error. *Philosophical Transactions of the Royal Society of London, SERIES A*, 356(1740):1175–1192, 1998.

[12] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision – 2nd Edition*. Cambridge University Press, 2004.

[13] F. Kahl. Multiple view geometry and the L_∞-norm. In *Proc. International Conference on Computer Vision*, pages 1002–1009, 2005.

[14] R. Kaucic, R. I. Hartley, and N. Y. Dano. Plane-based projective reconstruction. In *Proc. 8th International Conference on Computer Vision, Vancouver, Canada*, pages I–420–427, 2001.

[15] K. Krakowski, K. Hüper, and J. Manton. On the computation of the karcher mean on spheres and special orthogonal groups. In *RoboMat 2007, Workshop on Robotics and Mathematics*, Coimbra, Portugal, 2007.

[16] Y. Li. A Newton acceleration of the Weiszfeld algorithm for minimizing the sum of euclidean distances. *Computational Optimization and Applications*, 10:219–242, 1998.

[17] J. H. Manton. A globally convergent numerical algorithm for computing the centre of mass on compact Lie groups. In *Proceedings of the Eighth International Conference on Control, Automation, Robotics and Vision*, pages 2211–2216, Kunming, China, December 2004.

[18] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[19] M. Moakher. Means and averaging in the group of rotations. *SIAM J. Matrix Anal. Appl.*, 24(1):1–16 (electronic), 2002.

[20] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004.

[21] L. Ostresh. Convergence of a class of iterative methods for solving weber location problem. *Operations Research*, 26:597–609, 1978.

[22] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery. In *Proc. 8th International Conference on Computer Vision, Vancouver, Canada*, pages I–42–49, 2001.

[23] A. Sarlette and R. Sepulchre. Consensus optimization on manifolds. *SIAM J. Control Optim.*, 48(1):56–76, 2009.

[24] K. Sim and R. Hartley. Recovering camera motion using L_∞ minimization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, New York City*, pages 1230 – 1237, 2006.

[25] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 835–846, New York, NY, USA, 2006. ACM.

[26] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[27] A. Weber. *Über den Standort der Industrien, Erster Teil: Reine Theorie des Standortes*. 1909.

[28] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Math. Journal*, 43:355–386, 1937.

[29] Wikipedia. `http://en.wikipedia.org/wiki/Geometric_median`. Visited 03 April 2011.

[30] L. Yang. Riemannian median and its estimation. *LMS Journal of Computation and Mathematics*, 13:461–479, 2010.