Generalized Weiszfeld Algorithms for Lq Optimization

Khurrum Aftab, Richard Hartley, Jochen Trumpf

Abstract—In many computer vision applications, a desired model of some type is computed by minimizing a cost function based on several measurements. Typically, one may compute the model that minimizes the L_2 cost, that is the sum of squares of measurement errors with respect to the model. However, the Lq solution which minimizes the sum of the q-th power of errors usually gives more robust results in the presence of outliers for some values of q, for example, $q = 1$. The Weiszfeld algorithm is a classic algorithm for finding the geometric $L1$ mean of a set of points in Euclidean space. It is provably optimal and requires neither differentiation, nor line search. The Weiszfeld algorithm has also been generalized to find the $L1$ mean of a set of points on a Riemannian manifold of non-negative curvature. This paper shows that the Weiszfeld approach may be extended to a wide variety of problems to find an Lq mean for $1 \leq q < 2$, while maintaining simplicity and provable convergence. We apply this problem to both single-rotation averaging (under which the algorithm provably finds the global Lq optimum) and multiple rotation averaging (for which no such proof exists). Experimental results of Lq optimization for rotations show the improved reliability and robustness compared to L_2 optimization.

Index Terms—Weiszfeld algorithm, rotation averaging, L_q mean.

1 Introduction / Literature Review

This paper describes a very simple iterative and provably convergent algorithm for L_q optimization and applies it to different problems, we refer to it as the L_q *Weiszfeld Algorithm*. By modifying the corresponding L_2 problem, the proposed algorithm solves for the L_q solution, where $1 \leq q < 2$. The modification consists of introducing appropriate weighting terms.

We consider in detail the problem of finding the L_q minimum of a set of points on a Riemannian manifold of non-negative sectional curvature. Note that this covers the case of points in $\mathbb{R}^N$, where the curvature is zero and the distance function is the Euclidean distance. Moreover, it is shown that the proposed algorithm can be used to solve for the L_q minimum of a set of rotations. This problem is commonly referred to as the L_q rotation averaging problem and takes two distinct forms: single rotation averaging in which several estimates of a single rotation are 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 proposed algorithm is an extension of our paper [21] which considered L_1 rotation averaging. In this paper, we describe the method in terms of the L_q cost; in most cases this involves no extra complication, and provides a more general form of averaging. Further, the L_q cost has the additional advantage that it is differentiable everywhere, if $q > 1$, which has both theoretical and practical advantages.

Given a set of points $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ in some metric space the L_q cost function takes the following form

$$C(\mathbf{x}) = \sum_{i=1}^{k} d(\mathbf{x}, \mathbf{y}_i)^q \ . \tag{1}$$

We refer to a minimum of this function as the L_q mean of the points.

By using the L_q Weiszfeld algorithm a solution of the problem is found by iteratively solving a weighted L_2 problem,

$$\mathbf{x}^{t+1} = \operatorname*{argmin}_{\mathbf{x}} \sum_{i=1}^{k} w_i^t \, d(\mathbf{x}, \mathbf{y}_i)^2 \ , \tag{2}$$

where w_i^t is a weight associated with $d(\mathbf{x}^t, \mathbf{y}_i)$ at iteration t. We refer to a minimum of this function as a weighted L_2 minimum or weighted L_2 mean. In this paper we show that if we choose $w_i^t = d(\mathbf{x}^t, \mathbf{y}_i)^{q-2}$ where $\mathbf{x}^t$ is the estimate of $\mathbf{x}$ at iteration t and $1 \leq q < 2$ then the L_q Weiszfeld algorithm converges to the L_q solution, that is to $\min_{\mathbf{x}} \sum_{i=1}^{k} d(\mathbf{x}, \mathbf{y}_i)^q$. Thus from an implementation point of view the minimization of the L_q cost function only involves the iterative solution of a weighted L_2 problem, for which a closed-form solution exists in many cases.

For $q = 1$ the problem of finding the L_q mean of a set of points in $\mathbb{R}^N$ is known as the Fermat-Weber problem. A well-known globally convergent algorithm for solving this problem is the Weiszfeld algorithm [51], [52]. The proposed algorithm is named the L_q Weiszfeld algorithm because in addition to the minimization of the L_1 cost the proposed algorithm can also be used to minimize the L_q cost for $1 \leq q < 2$.

In the area of operations research and computational geometry the Fermat-Weber problem is a well studied problem [29], [42]. The Fermat-Weber problem is the simplest form of the facility location problem and solves for the placement of a single facility to reduce the distance to each of the fixed demand points. A Weiszfeld inspired solution strategy to solve an L_q version of the problem has also been proposed in [6], [4], [5]. Note that in [6] the sum of L_q norms is minimized, that is $\min_{\mathbf{x}} \sum_{i=1}^{k} \|\mathbf{x} - \mathbf{y}_i\|_q$. That is quite different from the type of problems we solve in this

- *K. Aftab and R. Hartley are with the Research School of Engineering, Australian National University and NICTA (National ICT Australia).*
E-mail: Khurrum.Aftab@anu.edu.au, Richard.Hartley@anu.edu.au
- *J. Trumpf is with the Research School of Engineering, Australian National University.*
E-mail: Jochen.Trumpf@anu.edu.au

paper, where the sum of the q-th power of distances is minimized, that is $\min_{\mathbf{x}} \sum_{i=1}^{k} d(\mathbf{x}, \mathbf{y}_i)^q$.

The original paper by Weiszfeld [51] (see [52] for an English translation) gave an algorithm for computing the L_1 mean of a given set of points in $\mathbb{R}^N$. Since then it has been generalized to L_1-closest-point problems in Banach spaces [15], rotation space $\mathrm{SO}(3)$ [21], [25] and general Riemannian manifolds [17], [53]. The characteristic of the Weiszfeld algorithm and its generalizations is that they are provably convergent iterative L_1 optimization algorithms that do not require computation of derivatives or line-search. As such, they are very easy to understand, and code. The iterative update is very quick to compute and in practice, the algorithms are quick to converge.

This paper identifies the critical argument in the Weiszfeld algorithm, and shows that it may be generalized to find an L_q solution, where $1 \leq q < 2$. In addition to the L_q solution for points in $\mathbb{R}^N$ the L_q Weiszfeld algorithm can also be applied to a wider class of problems with the same advantages of the Weiszfeld algorithm. The L_2 mean is commonly referred as the Fréchet mean. Several methods exist in the literature to locate the Fréchet mean, for example [20], [31], [34]. A constant step-size gradient descent method, different than the Weiszfeld algorithm and its generalizations, is proposed in [2] to find the L_q mean, for $2 \leq q < \infty$, of a set of points on a Riemannian manifold.

The problem of rotation averaging has significant applications to structure and motion [35], [45], [22], [26], [43], [27] and to non-overlapping camera calibration [12]. It has been studied quite extensively in the past, both in computer vision and in other fields. In the area of information theory the problem of rotation averaging is also known as the synchronization problem [49]. Significant work in this area includes the work of Govindu [19], [18] and Pajdla [35].

Significant contributions to the single rotation averaging problem have been made by [37], [34], [44] and others. Most significant from our point of view is the work reported in [17], [53], [1] which considers L_1 minimization on classes of Riemannian manifolds, proving convergence theorems in a broad context, which relate directly to our algorithm [21] but are not sufficient to show convergence in the desired generality. However, the problem of multiple rotation averaging has been studied outside the vision field in the context of sensor network localization [10], [48] and the molecule problem in structural biology [11].

The L_q Weiszfeld algorithm is the same as the Iterative Reweighted Least Squares (IRLS) technique. IRLS techniques have existed in the literature for a long time, but here we show that for a particular choice of weights the IRLS algorithm converges to the desired L_q minimum. The proposed technique must not be confused with the IRLS technique in compressive sensing (CS), [13], [7], [16], because in this paper we solve a more general class of problems rather than the problems with a sparse solution. In compressive sensing the system of equations is under-determined and has a sparse solution, whereas in this case the system of equations is over-determined and may have a non-sparse solution. The convergence of the IRLS algorithms, in compressive sensing, to the L_1 minimum is achieved only if the solution vector has sparse components, otherwise convergence to the L_1 minimum is not guaranteed. Therefore, the applicability of these techniques is limited only to compressive sensing problems. Furthermore, the technique proposed in this paper not only solves for the L_1 but also solves for the L_q solution of a more general class of problems.

The proposed algorithm can also be applied to the problem of finding the L_q-closest-point to a set of affine subspaces in $\mathbb{R}^N$, [3]. The L_q-closest-point to the subspaces is a point for which the sum of the q-th power of the orthogonal distances to all of the affine subspaces is minimum.

In this paper we provide a proof that the L_q Weiszfeld algorithm converges to the L_q minimum. We also show that the L_q Weiszfeld algorithm can be applied to points on a complete Riemannian manifold of non-negative sectional curvature, for example $\mathrm{SO}(3)$. The simplicity of the L_q Weiszfeld algorithm and the rapidity with which its iterative update may be computed makes it attractive.

2 L_1 Optimization using the Weiszfeld Algorithm

Given a set of points $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$ in some metric space, the L_1 mean or geometric median is a point $\bar{\mathbf{y}}$ that minimizes the sum of distances to all given points. Thus,

$$\bar{\mathbf{y}} = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{k} d(\mathbf{x}, \mathbf{y}_i) \ , \tag{3}$$

where $d(\mathbf{x}, \mathbf{y}_i)$ is the distance between $\mathbf{x}$ and $\mathbf{y}_i$. In the rest of this section we discuss a brief history of the distance minimization problem in $\mathbb{R}^N$, followed by the Weiszfeld algorithm for points in $\mathbb{R}^N$ and on a Riemannian manifold of non-negative sectional curvature.

2.1 History

A special case of the above problem is known as Fermat's problem where the L_1 solution for 3 sample points in a plane is desired. This problem was originally posed by Pierre de Fermat to Evangelista Torricelli, who solved it. Its solution is known as the Fermat point of a triangle with each sample point as a vertex. The more general form of the problem for more than 3 points in $\mathbb{R}^N$ was studied by Alfred Weber [50] and therefore is known as the Fermat-Weber problem.

A mechanical system is shown in fig. 1 to demonstrate the working of the Fermat-Weber problem. In fig. 1 each given point $\mathbf{y}_i$ is represented by a pulley. Strings are passed over the pulleys and unit weights are attached to one end of the strings. The other ends of all the strings are tied together. This mechanical system will reach an equilibrium state. At this point all the forces (due to unit weights) in all directions will cancel each other and there will be no change in the location of the knot connecting the strings. The final location of the knot is the L_1 solution in case of points in $\mathbb{R}^2$, because at the minimum point all the unit magnitude forces cancel each other. It follows that at this point the gradient of (3) is zero; it is shown in (6) that the gradient of (3) is in fact the sum of these unit vectors.

2.2 Weiszfeld Algorithm for Points in $\mathbb{R}^N$

The Weiszfeld algorithm solves for the L_1 mean of a set of points $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$ in $\mathbb{R}^N$. The L_1 mean or geometric median is the point $\mathbf{x}$ that minimizes the cost function

$$C_1(\mathbf{x}) = \sum_{i=1}^{k} \|\mathbf{x} - \mathbf{y}_i\| \ , \tag{4}$$

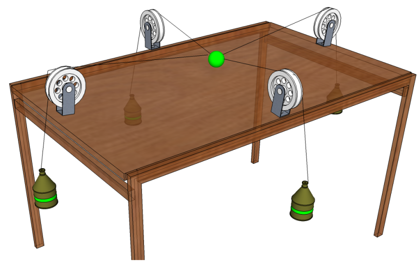


Fig. 1: *Mechanical setup for the Fermat-Weber problem: Each pulley represents a fixed point. Strings are passed over the pulleys and unit weights are attached to one end of string while other ends are tied together at a point (shown in green) The equilibrium position of that point will be the L_1 minimum.*

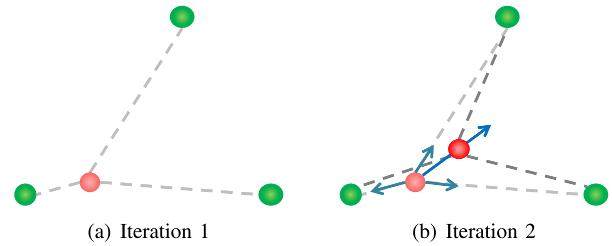


Fig. 2: *Weiszfeld Algorithm (Gradient Descent Form): (a) shows three fixed points (green) and a starting point (red) from which the sum of distances to fixed points (green) is to be minimized. (b) shows an updated point (red) after one iteration of the Weiszfeld algorithm in the descent direction.*

where $\|\cdot\|$ is the Euclidean norm.

The Weiszfeld algorithm updates a current estimate $\mathbf{x}^t$ to

$$\mathbf{x}^{t+1} = \frac{\sum_{i=1}^{k} w_i^t \mathbf{y}_i}{\sum_{i=1}^{k} w_i^t} \ , \tag{5}$$

where $w_i^t = \|\mathbf{x}^t - \mathbf{y}_i\|^{-1}$. If all of the given points are non-collinear then the cost function C_q has a unique minimum, and the sequence of iterates $\mathbf{x}^t$ will converge to the minimum of the cost (4), except if it gets stuck at one of the points $\mathbf{y}_i$, as explained next.

2.2.1 Continuity of the update

If one of the iterations $\mathbf{x}^t$ approaches one of the points $\mathbf{y}_i$, then the weight w_i^t becomes very large, and in the limit, the update step, as given by (5) becomes undefined. However, this is a *removable* singularity. Suppose that one of the points, say $\mathbf{y}_1$ is the one closest to an iteration $\mathbf{x}^t$, then one may replace all the weights by $\tilde{w}_i^t = w_i^t / w_1^t$ and $w_1' = 1$ without altering the update. With these weights, the update step is well-defined, and continuous, even when $\mathbf{x}^t$ is equal to one of the points $\mathbf{y}_i$. Although this renormalization of weights removes the apparent singularity in the definition of the update step, we shall continue to use the formula $w_i^t = \|\mathbf{x}^t - \mathbf{y}_i\|^{-1}$, just so as to avoid complicating the exposition.

Although this trick removes the singularity at the points $\mathbf{y}_i$, the update so defined results in $\mathbf{x}^{t+1} = \mathbf{x}^t = \mathbf{y}_i$ whenever $\mathbf{x}_t$ equals $\mathbf{y}_i$ exactly. Thus, the sequence of iterations gets stuck at $\mathbf{y}_i$. In this case, it cannot be concluded (and is not generally true) that $\mathbf{y}_i$ is the minimum of the cost function.

2.2.2 Getting stuck

This possibility of "getting stuck" at a value $\mathbf{x}^t = \mathbf{y}_i$ is perhaps the main theoretical flaw of the Weiszfeld algorithm. From a practical point of view, however, it is not a significant issue. This eventuality is rarely if ever encountered in practice. If it is, there are ways to handle it.

A simple strategy if $\mathbf{x}^t$ coincides with one of the $\mathbf{y}_i$ is to displace the iterate $\mathbf{x}^t$ slightly and continue. It may be shown that successive iterates will "escape" from some point $\mathbf{y}_i$, not the minimum, by approximately doubling the distance at each iteration. A second possibility is to start the iteration at some point with cost smaller than the cost of any of the points $\mathbf{y}_i$, in which case it is not possible that an iteration will return to approach one of the points.

2.2.3 Different interpretations

The Weiszfeld algorithm can be viewed in various ways as a gradient descent algorithm, an Iterative Re-Weighted Least Squares (IRLS) algorithm and a weighted mean algorithm. Below we discuss two interpretations of the Weiszfeld algorithm.

Gradient Descent Form. The gradient of the cost function (4) is

$$\nabla C_1 = \sum_{i=1}^{k} \frac{\mathbf{x} - \mathbf{y}_i}{\|\mathbf{x} - \mathbf{y}_i\|} \ . \tag{6}$$

Given a current estimate of the L_1 minimum $\mathbf{x}^t$ at iteration t, the next estimate of the minimum in the descent direction is computed as

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \lambda \nabla C_1 \ , \tag{7}$$

where λ is the update step size; see fig. 2. The value of λ in the Weiszfeld algorithm is $\lambda = 1/\sum_{i=1}^{k} w_i^t$, where $w_i^t = 1/\|\mathbf{x}^t - \mathbf{y}_i\|$. By substituting the value of λ and ∇C_1 in (7) one obtains the same formula as (5) for the update. An advantage of the Weiszfeld algorithm is that the descent direction and step size are computed in closed form. Therefore each iteration of the Weiszfeld is fast, compared to other gradient descent algorithms that compute the step size using complex strategies such as line search, etc.

Iterative Re-weighted Least Squares (IRLS) Form. Note that (5) updates a current estimate $\mathbf{x}^t$ by computing a weighted mean of points $\mathbf{y}_i$. An alternative interpretation of the Weiszfeld algorithm is that it can be viewed as an Iterative Re-weighted Least Squares method. At iteration t, the weighted least squares cost function is

$$C^W(\mathbf{x}) = \sum_{i=1}^{k} w_i^t \ \|\mathbf{x} - \mathbf{y}_i\|^2 \ , \tag{8}$$

where $w_i^t = 1/\|\mathbf{x}^t - \mathbf{y}_i\|$. By taking the derivative of the weighted Least Squares function and equating to zero we get an update function, the same as (5). At each step, $\mathbf{x}^{t+1}$ is the exact minimum of the weighted problem. Thus, the Weiszfeld algorithm solves a special type of IRLS cost function to achieve the L_1 solution.

From the above discussion it is obvious that the L_1 solution can be obtained by minimizing either an L_1 cost or a weighted L_2 function as in (8).

2.3 Weiszfeld Algorithm on a Riemannian Manifold

The Weiszfeld algorithm has been generalized to find the L_1 minimum of a set of points on a Riemannian manifold of non-negative sectional curvature [17]. Given a set of points, $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ in a Riemannian manifold $\mathcal{M}$, the L_1 mean or geodesic median is a point $\mathbf{x} \in \mathcal{M}$ for which the sum of geodesic distances

$$C_1(\mathbf{x}) = \sum_{i=1}^{k} d(\mathbf{x}, \mathbf{y}_i) = \| \log_{\mathbf{x}}(\mathbf{y}_i) \| \ , \tag{9}$$

is minimized, where $d(\cdot, \cdot)$ is the geodesic distance between two point on the manifold $\mathcal{M}$ and $\log_{\mathbf{x}}(\mathbf{y})$ is the *logarithm map* that takes a point $\mathbf{y} \in \mathcal{M}$ to the tangent space $T_{\mathbf{x}}\mathcal{M}$ of $\mathcal{M}$ centered at $\mathbf{x} \in \mathcal{M}$.

Given some points on a Riemannian manifold and an initial estimate of their L_1 mean, an updated point is computed by transferring all the given points to the tangent space of the manifold centered at the current estimate; see fig. 3. The point is then transferred back to the manifold. This process is repeated until convergence. This type of technique is convergent on manifolds of non-negative sectional curvature, provided all the points lie in a suitably small set, as will be described later.

The gradient of the cost function (9) is $\nabla C_1(\mathbf{x}) = -\sum_{i=1}^{k} \log_{\mathbf{x}}(\mathbf{y}_i)/\| \log_{\mathbf{x}} \mathbf{y}_i \|$. A current solution $\mathbf{x}^t$ is updated in the descent direction as

$$\mathbf{x}^{t+1} = \exp_{\mathbf{x}^t}(-\lambda \ \nabla C_1(\mathbf{x}^t)) \ , \tag{10}$$

where λ is the step size for the gradient descent algorithm and $\exp_{\mathbf{x}^t}(\mathbf{v})$ is the *exponential map* that maps a vector $\mathbf{v} \in T_{\mathbf{x}^t}\mathcal{M}$ to a point on $\mathcal{M}$.

In case of the Weiszfeld algorithm the step size is $\lambda = 1/\sum_{i=1}^{k} 1/d(\mathbf{x}, \mathbf{y}_i)$. By substituting the value of λ in (10) we get

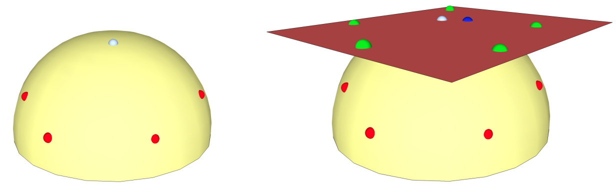
$$\mathbf{x}^{t+1} = \exp_{\mathbf{x}^t}\left(\frac{\sum_{i=1}^{k} \log_{\mathbf{x}^t}(\mathbf{y}_i) \ / \ \| \log_{\mathbf{x}^t}(\mathbf{y}_i) \|}{\sum_{i=1}^{k} 1 \ / \ \| \log_{\mathbf{x}^t}(\mathbf{y}_i) \|} \right) \ . \tag{11}$$

This update may be written more simply as

$$\mathbf{x}^{t+1} = \exp_{\mathbf{x}^t}\left(\frac{\sum_{i=1}^{k} w_i^t \ \log_{\mathbf{x}^t}(\mathbf{y}_i)}{\sum_{i=1}^{k} w_i^t} \right) \ , \tag{12}$$

where $w_i^t = d(\mathbf{x}^t, y_i)^{-1} = \| \log_{\mathbf{x}^t}(\mathbf{y}_i) \|^{-1}$. This update equation is seen to come from finding the weighted average in the tangent space at $\mathbf{x}^t$ of the points $\log_{\mathbf{x}^t}(\mathbf{y}_i)$, followed by mapping back to the manifold by the exponential map.

Conditions for the convergence of the L_1 Weiszfeld algorithm on positively curved manifolds can be found in [17]. However, the L_q Weiszfeld algorithm proposed in this paper solves for the L_q solution $1 \leq q < 2$ of the problem. In addition, the proof given in this paper is more complete than the proof in [17], even for the L_1 case. In particular, in this paper we improve the bounds on the location of the averaged points for which the algorithm will converge, compared to those given in [17], and we fill in some detail concerning the convergence point of the algorithm (for example, what happens when the algorithm converges to some $\mathbf{y}_i$). Furthermore, a technical point concerning the use of the Toponogov's theorem will be addressed. In [53], step-size control algorithms such as line search are used at each step, so the algorithm is not a true Weiszfeld-style algorithm.



(a) Iteration 1 (b) Iteration 2

Fig. 3: *Weiszfeld Algorithm on Manifold: (a) represents a manifold with some given fixed points (red) and a starting point (white). (b), the Weiszfeld algorithm is applied to the transformed points (green) in the tangent space (red plane) and an updated point (blue) is computed in a descent direction. This updated point is then mapped back to the manifold and the procedure is repeated until convergence.*

Our proof of convergence on a Riemannian manifold of non-negative curvature is given in section 6, Theorem 3.6.

3 L_q Weiszfeld Algorithm

The purpose of this paper is to explore and prove the convergence of Weiszfeld-style algorithms in $\mathbb{R}^N$ and on manifolds. We prove the convergence of the L_q Weiszfeld algorithm for points on a Riemannian manifold of non-negative curvature. The curvature of $\mathbb{R}^N$ with the Euclidean metric is zero, so this is a special case of non-negative curvature. As it turns out, the analysis of the L_q Weiszfeld algorithm is scarcely more difficult than for the standard L_1 algorithm. For this reason, we will consider the more general case of minimization of L_q cost functions. Although the case L_1 has received much attention in previous work, there are some advantages to considering the L_q case. First this gives a range of choices between the L_1 cost, favoured for its robustness to noise, and the L_2 cost which is more theoretically justified statistically, assuming Gaussian noise. In addition, considering the L_q cost function avoids the difficulty that the L_1 cost is not differentiable. For $q > 1$ the cost function is differentiable everywhere. A final consideration is that the L_1 mean of a set of points may (with non-zero probability) coincide with one of the points themselves. This requires special care, since the minimum is then a point where the cost-function is non-differentiable. On the other hand, the L_q mean of a set of points will not generically coincide with any of the points, and besides, the cost function is differentiable everywhere. [1]

Given a set of points $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$, $k > 2$, on a Riemannian manifold $\mathcal{M}$, the L_q cost function C_q is

$$C_q(\mathbf{x}) = \sum_{i=1}^{k} d(\mathbf{x}, \mathbf{y}_i)^q = \sum_{i=1}^{k} \| \log_{\mathbf{x}}(\mathbf{y}_i) \|^q \ , \tag{13}$$

where $d(\cdot, \cdot)$ is the geodesic distance between two points on the manifold $\mathcal{M}$. The gradient of the cost function C_q is

$$\nabla C_q(\mathbf{x}) = -\sum_{i=1}^{k} d(\mathbf{x}, \mathbf{y}_i)^{q-2} \log_{\mathbf{x}}(\mathbf{y}_i) \ .$$

1. One can show, using lemma 3.5 that if points $\mathbf{y}_i$ are chosen at random in a ball of given radius, then the probability that their L_q mean corresponds with one of the points $\mathbf{y}_i$ is non-zero for $q = 1$, but is zero for $q > 1$.

In the following section we will state the L_q Weiszfeld algorithm. A detailed proof of convergence of the proposed algorithm is presented in section 6.

3.1 Algorithm

The L_q Weiszfeld algorithm differs from the L_1 algorithm mainly in the choice of weights applied at each step of iteration. In this case, the update equation (12) is with weights given by $w_i^t = d(\mathbf{x}^t, \mathbf{y}_i)^{q-2} = \| \log_{\mathbf{x}^t}(\mathbf{y}_i)\|^{q-2}$. Starting from an initial estimate $\mathbf{x}^0$, the algorithm generates a sequence of estimates $\mathbf{x}^t$ found by solving a weighted least-squares problem, in the current tangent space of a Riemannian manifold. An updated solution is then projected back on the manifold and the process is repeated until convergence.

A current estimate $\mathbf{x}^t$ of the L_q minimum is updated to a new estimate

$$
\begin{aligned}
W(\mathbf{x}^t) &= \exp_{\mathbf{x}^t}\left(\frac{\sum_{i=1}^k w_i^t \log_{\mathbf{x}^t}(\mathbf{y}_i)}{\sum_{i=1}^k w_i^t}\right) \quad \text{if } \mathbf{x}^t \notin \mathcal{Y} \\
&= \mathbf{y}_j \quad \text{if } \mathbf{x}^t = \mathbf{y}_j
\end{aligned}
, \qquad (14)
$$

where

$$
w_i^t = d(\mathbf{x}^t, \mathbf{y}_i)^{q-2} \ .
$$

Starting from a point $\mathbf{x}^0 \in \mathcal{M}$, a sequence of points $(\mathbf{x}^t)$ is obtained using W as $\mathbf{x}^{t+1} = W(\mathbf{x}^t)$. Here, the logarithm map is defined with respect to a weakly convex set, as will be explained in Theorem 3.3. In section 6 we will show that under certain conditions, the sequence of points $(\mathbf{x}^t)$ converges to the L_q minimum or it will stop at some point $\mathbf{x}^t = \mathbf{y}_i$. The conditions required for convergence are that the manifold has non-negative sectional curvature, and that the points $\mathbf{y}_i$ and the initial estimate $\mathbf{x}^0$ lie in a sufficiently restricted region in the manifold.

Remark: Since $\mathbb{R}^N$ is a Riemannian manifold with curvature zero, the algorithm holds there as a special case. In this case the distance function in (13) is simply the Euclidean distance and the update rule (14) may be simplified by omitting the exponential and logarithm maps.

Given a set of points in $\mathbb{R}^N$, $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$. Starting from an initial estimate $\mathbf{x}^0$ of the L_q mean, a current estimate $\mathbf{x}^t$ of the L_q mean is updated as

$$
\begin{aligned}
\mathbf{x}^{t+1} = W(\mathbf{x}^t) &= \frac{\sum_{i=1}^k w_i^t \mathbf{y}_i}{\sum_{i=1}^k w_i^t} \qquad \text{if } \mathbf{x}^t \in \{\mathbf{y}_i\} \\
&= \mathbf{y}_j \qquad \text{if } \mathbf{x}^t = \mathbf{y}_j
\end{aligned}
, \qquad (15)
$$

where $w_i^t = \|\mathbf{x}^t - \mathbf{y}_i\|^{q-2}$.

3.2 Convex sets

The theory of L_q distances in a Riemannian manifold is connected with the concept of a convex set. We discuss this concept before proceeding. It will be used later in section 6 to prove the convergence of the L_q Weiszfeld algorithm.

A *geodesic* is a generalization of the notion of straight line to Riemannian manifolds. A *geodesic segment* joining points $\mathbf{x}$ and $\mathbf{y}$ in $\mathcal{M}$ is an arc length parametrized continuous curve $\gamma : [a, b] \to \mathcal{M}$ such that $\gamma(a) = \mathbf{x}$ and $\gamma(b) = \mathbf{y}$. A geodesic segment is called *minimizing* if it is the minimum length curve joining its end points. By the Hopf-Rinow theorem [38], any two points $\mathbf{x}$ and $\mathbf{y}$ in a complete Riemannian manifold are joined by a minimizing geodesic segment (though this may not be unique). A convex (or strongly convex) set in a manifold $\mathcal{M}$ is a set that contains a unique geodesic joining two points in the set, and that geodesic is the minimizing geodesic.

A function $f : C \to \mathbb{R}$ defined on a convex set C in $\mathcal{M}$ is *convex* if its restriction to a geodesic in C is a convex function of arc-length. Let $B(\mathbf{x}, r)$ denote an open ball of radius r centered at $\mathbf{x} \in \mathcal{M}$ and $\bar{B}(\mathbf{x}, r)$ be the closure of $B(\mathbf{x}, r)$. A ball satisfying the property of a convex set is a *convex ball*.

The injectivity radius at a point $\mathbf{x}$ of a Riemannian manifold is the supremum of the radii r for which the inverse exponential map at $\mathbf{x}$ is a diffeomorphism on $B(\mathbf{x}, r)$. The injectivity radius r_{inj} of a Riemannian manifold, $\mathcal{M}$, is the infimum of the injectivity radii at all points. Open balls of radius r_{inj} or less in the manifold are therefore diffeomorphic to a Euclidean ball.

Another important quantity is the *convexity radius* of the manifold, which is the largest value r_{conv} such that all open balls $B(\mathbf{o}, r)$ of radius $r < r_{\text{conv}}$ are convex, and furthermore, radius $r(\mathbf{x}) = d(\mathbf{x}, \mathbf{o})$ is a convex function on this ball. It is shown in [41], page 177 that the convexity radius of a manifold $\mathcal{M}$ is bounded as follows:

$$
r_{\text{conv}} \geq \frac{1}{2} \min\left(r_{\text{inj}}, \frac{\pi}{\sqrt{\Delta}}\right) \ ,
$$

where Δ is the maximal sectional curvature of the manifold $\mathcal{M}$.

In [17], the convergence of the L_1 Weiszfeld algorithm was shown for manifolds of non-negative sectional curvature, provided that all the points lie inside a ball of radius no greater than $r_{\text{conv}}/2$. However, this is not a tight bound, and in the case of rotation averaging, it is possible to prove convergence within a ball of twice this size. The convexity radius for $\mathrm{SO}(3)$ is equal to $\pi/2$, and hence the result of [17] ensures convergence as long as the points $\mathbf{y}_i$ lie inside a ball of radius $\pi/4$. It will be shown in this paper that convergence to the L_q minimum is assured, as long as the points are within a ball of radius $\pi/2$, that is within a convex ball in $\mathrm{SO}(3)$. To obtain these improved convergence results in the general case, a different concept of convexity is needed.

3.3 Weakly convex sets

A *weakly convex* set in a manifold $\mathcal{M}$ is a set W with the following properties.

1) For two points $\mathbf{x}$ and $\mathbf{y}$ in W, there is a unique geodesic in W from $\mathbf{x}$ to $\mathbf{y}$.
2) This geodesic is the shortest length path in W from $\mathbf{x}$ to $\mathbf{y}$.

If this segment is the minimizing geodesic in $\mathcal{M}$ joining $\mathbf{x}$ and $\mathbf{y}$, then the set is strongly convex, as previously defined. [2]

In a weakly convex set W, we may define a distance $d_W(\mathbf{x}, \mathbf{y})$ equal to the length of the unique geodesic that joins $\mathbf{x}$ to $\mathbf{y}$ in W. In addition, the logarithm map on W is defined in terms of the unique geodesic between two points, in W.

2. Terminology for convexity properties varies in the literature, e.g. [8], [9], [28], [41]. Our definition of weakly convex is closest to the definition of weakly convex in [8]. However, we add an extra condition of uniqueness of the connecting geodesic into our definition of weak convexity to ensure the absence of conjugate points, and injectivity of the exponential map, hence uniqueness of the logarithm map defined on W.

Lemma 3.1. *For an open weakly convex set W in a Riemannian manifold $\mathcal{M}$,*

1) *The distance function $d_W(\mathbf{x}, \mathbf{y})$ satisfies the triangle inequality, and hence W is a metric space under this metric.*
2) *W does not contain any pair of conjugate points.*
3) *For any point $\mathbf{x} \in W$, the logarithm map $\log_{\mathbf{x}}$ maps W diffeomorphically into the tangent space.*

The triangle inequality follows directly from the fact that distance is equal to the shortest path length in W.

If there exist conjugate points $\mathbf{y}$ and $\mathbf{y}'$ in W, then the geodesic from $\mathbf{y}$ to $\mathbf{y}'$ can be extended to a geodesic from $\mathbf{y}$ to $\mathbf{y}''$. However, geodesics are not minimizing beyond conjugate points [32], so the geodesic from $\mathbf{y}$ to $\mathbf{y}''$ is not minimizing in the manifold W, contrary to the definition of a weakly convex set.

The third statement follows from injectivity of the exponential map into W and a standard result about the exponential map on regions without conjugate points [32].

We use Toponogov's theorem to compare distances in W and in its tangent space. Let κ be the sectional curvature of a Riemannian manifold. The following theorem effectively states that the Toponogov's theorem holds in an open weakly convex set with $\kappa \geq 0$.

Theorem 3.2. *Let W be an open weakly convex set in $\mathcal{M}$, a manifold of non-negative sectional curvature. Let $\mathbf{q}$, $\mathbf{p}_1$ and $\mathbf{p}_2$ be three points in W. Then*

$$d_W(\mathbf{p}_1, \mathbf{p}_2) \leq d(\log_{\mathbf{q}}(\mathbf{p}_1), \log_{\mathbf{q}}(\mathbf{p}_2)) = \| \log_{\mathbf{q}}(\mathbf{p}_1) - \log_{\mathbf{q}}(\mathbf{p}_2)\| \ .$$

A proof of the above theorem is provided in appendix A.2. The proof is given, because the usual conditions required by Toponogov's theorem to be true are not satisfied in a weakly convex set, so a proof is required.

Analogous to the definition of convexity radius, we define the *weak convexity radius* r_{wcon} to be the largest value such that all open balls $B(\mathbf{o}, r)$ with $r < r_{\text{wcon}}$ are weakly convex, and $r(\mathbf{x})$ is convex. It is easy to see that $2r_{\text{conv}} \geq r_{\text{wcon}} \geq r_{\text{conv}}$. Indeed, by definition, a ball of radius $\rho < r_{\text{wcon}}/2$ is weakly convex. Therefore, two points $\mathbf{x}$ and $\mathbf{y}$ in $B(\mathbf{o}, \rho)$ are connected by a unique geodesic segment γ in $B(\mathbf{o}, \rho)$. However, this must be a minimizing segment; there cannot be another such segment lying in $B(\mathbf{o}, 2\rho)$, since this is weakly-convex, and any other geodesic from $\mathbf{x}$ to $\mathbf{y}$ that exits the ball $B(\mathbf{o}, 2\rho)$ must be longer than γ. Thus, $B(\mathbf{o}, \rho)$ is convex.

As an example, consider the manifold $\mathrm{SO}(3)$. It is shown in [25] that the convexity radius of $\mathrm{SO}(3)$ is $\pi/2$, whereas the weak-convexity radius is π, which is the maximum distance between points in $\mathrm{SO}(3)$. Note that part of this claim is that $d(\mathbf{x}, \mathbf{o})$ is convex on any ball of radius less than π (and in fact on the open ball $B(\mathbf{o}, \pi)$ itself); see [25]. The closed ball $\bar{B}(\mathbf{o}, \pi)$ is equal to the whole of $\mathrm{SO}(3)$, but any smaller ball is weakly convex.

An essential property of open weakly convex sets is the continuity of the logarithm map. The following theorem says that if the logarithm map is defined in terms of the geodesic that lies inside W, then it is continuous as a function of two variables on this region.

Theorem 3.3. *If W is an open weakly convex set in a complete Riemannian manifold $\mathcal{M}$, and $\mathbf{x}, \mathbf{y}$ are two points in W, define $\log_{\mathbf{x}}(\mathbf{y})$ to be the vector $\mathbf{v}$ in $T_{\mathbf{x}}\mathcal{M} \subset T\mathcal{M}$ such that $\exp_{\mathbf{x}}(\mathbf{v}) = \mathbf{y}$, and $\exp_{\mathbf{x}}(t\mathbf{v}) \in W$ for all $t \in [0, 1]$. Then $\log_{\mathbf{x}}(\mathbf{y})$ as a map from $W \times W$ to $T\mathcal{M}$ is continuous in both variables.*

Our search fails to find this theorem in the literature, so we give a proof of this theorem in appendix A.1.

3.4 Convexity and minima of the L_q cost

The L_q cost function (13) will in general have more than one local minimum on an arbitrary manifold. For example, it was shown in [25] that for n points on the manifold $\mathrm{SO}(3)$, there may be up to $O(n^3)$ local minima. The situation becomes much simpler in the case when (13) is a convex function on some convex region.

It is shown in [1] (proof of Theorem 2.1) that if all points $\mathbf{y}_i$ lie in a ball $\bar{B}(\mathbf{o}, \rho)$ of radius $\rho < r_{\text{conv}}$, then the global minimum of C_q lies in $\bar{B}(\mathbf{o}, \rho)$. Furthermore, if $\mathbf{x}$ is a point not in $\bar{B}(\mathbf{o}, \rho)$ then there exists a point $\mathbf{x}' \in B(\mathbf{o}, \rho)$ such that $C_q(\mathbf{x}') < C_q(\mathbf{x})$. In fact, a specific construction is given for $\mathbf{x}'$, as follows. Suppose that $\mathbf{x} = \exp_{\mathbf{o}}(r\mathbf{v})$ where $\|\mathbf{v}\|$ is a unit vector, and $r = d(\mathbf{x}, \mathbf{o}) > \rho$, then

$$\mathbf{x}' = \begin{cases} \exp_{\mathbf{o}}((2\rho - r)\mathbf{v}) & \text{if } r < 2\rho \\ \mathbf{o} & \text{if } r \geq 2\rho \end{cases} . \tag{16}$$

Thus, for $r < 2\rho$, point $\mathbf{x}'$ is the reflection of $\mathbf{x}$ about the boundary of the ball $B(\mathbf{o}, \rho)$, along the radial geodesic. Since $r_{\text{wcon}}/2 \leq r_{\text{conv}}$, the cost function C_q has appealing properties on a ball of radius $\rho < r_{\text{wcon}}/2$, as follows.

Theorem 3.4. *If all points $\mathbf{y}_i$, $i = 1, \ldots, n$ lie in a ball $\bar{B} = \bar{B}(\mathbf{o}, \rho)$ with $\rho < r_{\text{wcon}}/2$, then*

1) *$C_q(\mathbf{x})$ is convex on $\bar{B}$ and strictly convex unless $q = 1$ and all points $\mathbf{y}_i$ lie on a single geodesic;*
2) *the global minimum of $C_q(\mathbf{x})$ lies in $\bar{B}$;*
3) *the set $S_0 = \{\mathbf{x} \in \mathcal{M} \mid C_q(\mathbf{x}) \leq C_q(\mathbf{o})\}$ is contained in $\bar{B}(\mathbf{o}, 2\rho)$, which is a weakly-convex ball.*

Proof: If $\mathbf{x}$ and $\mathbf{y}_i$ are both in $\bar{B}$, then $d(\mathbf{x}, \mathbf{y}_i) \leq 2\rho < r_{\text{wcon}}$. Hence, $d(\mathbf{x}, \mathbf{y}_i)$ is convex as a function of $\mathbf{x}$ on $\bar{B}$, so $d(\mathbf{x}, \mathbf{y}_i)$ has positive-semidefinite Hessian at $\mathbf{x}$. By a simple calculation it follows that $d(\mathbf{x}, \mathbf{y}_i)^q$ has positive-definite Hessian for $q > 1$, and is hence convex. Summing over all i shows that $C_q(\mathbf{x})$ is convex.

For $q = 1$ the distance $d(\mathbf{x}, \mathbf{y}_i)$ is strictly convex at $\mathbf{x}$ except in the direction pointing towards $\mathbf{y}_i$. Unless the directions to all the points $\mathbf{y}_i$ coincide, the sum of the distance functions will be strictly convex.

The second statement was proved in [1].

The third statement follows from the triangle inequality, since if $C_q(\mathbf{x}) \leq C_q(\mathbf{o})$, then $d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{o}, \mathbf{y}_i)$ for some i, and $d(\mathbf{x}, \mathbf{o}) \leq d(\mathbf{x}, \mathbf{y}_i) + d(\mathbf{o}, \mathbf{y}_i) \leq 2\rho$. □

The minimum of c_q may be characterized in terms of a vanishing sub-gradient. For $q > 1$ the cost is differentiable, so the subgradient is the same as gradient. In the case $q = 1$, the condition may be made explicit. The minima of the cost function C_q may be classified as follows.

Lemma 3.5. *Let D be a subset of $\mathcal{M}$ on which C_q (13) is convex. A point $\mathbf{x}^* \in D$ is the minimum of C_q in D if and only if it satisfies one of the following conditions:*

1) *$\nabla C_q(\mathbf{x})$ vanishes at $\mathbf{x}^*$, or*
2) *$q = 1$, $\mathbf{x}^* = \mathbf{y}_j$ and the gradient $\nabla \widehat{C}_q(\mathbf{x}^*)$ (omitting point $\mathbf{y}_j$) has norm no greater than 1.*

For the case $q > 1$ the cost function is differentiable and convex. Its minimum occurs when the gradient vanishes. In the case $q = 1$, the further possibility exists that the minimum occurs, under the stated condition, at some $\mathbf{y}_j$ where the cost function is non-differentiable. This is the condition given by Weiszfeld for the Euclidean case, and it carries over easily to the case of a Riemannian manifold.

3.5 Convergence theorem

We are now able to state a convergence theorem for the L_q Weiszfeld algorithm on a Riemannian manifold.

Theorem 3.6. *Consider a set of points $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$, $k > 2$, on a complete Riemannian manifold $\mathcal{M}$ with non-negative sectional curvature, such that not all of the given points lie on a single geodesic. Let all points $\mathbf{y}_i$, lie in a ball $B(\mathbf{o}, \rho)$ of radius $\rho < r_{\text{wcon}}/2$ centered at $\mathbf{o}$ and define $\mathcal{D} = \{\mathbf{x} \in \mathcal{M} \mid C_q(\mathbf{x}) \leq C_q(\mathbf{o})\}$.*

Let $(\mathbf{x}^t)$ be a sequence of points starting from $\mathbf{x}^0$ in $\mathcal{D}$, and defined by $\mathbf{x}^{t+1} = W(\mathbf{x}^t)$ where W is defined in (14). Then, the sequence $(\mathbf{x}^t)$ converges to the global minimum of C_q unless $\mathbf{x}^t = \mathbf{y}_i$ for some iteration t and point $\mathbf{y}_i$ (in which case, the sequence remains stuck at $\mathbf{y}_i$).

Proof of this theorem will be deferred until section 6.

4 L_q Optimization on $\mathrm{SO}(3)$

Here we address the problem of L_q rotation averaging, $1 \leq q < 2$, using the proposed L_q optimization method on a Riemannian manifold of non-negative sectional curvature.

Rotation averaging problems can be categorized as either single rotation averaging or multiple rotation averaging; see fig. 4. In single rotation averaging, several estimates of a rotation are found and then the L_q Weiszfeld algorithm is applied to find their L_q mean. In multiple rotation averaging, one is given a set of noisy relative rotations $\mathtt{R}_{ij}$ between frames (cameras) indexed by i and j. The task is to find absolute rotations $\mathtt{R}_i$, $\mathtt{R}_j$ that are consistent with the relative rotations: $\mathtt{R}_{ij} = \mathtt{R}_j\mathtt{R}_i^{-1}$. In applications, the relative rotations $\mathtt{R}_{ij}$ may be computed using single rotation averaging as well.

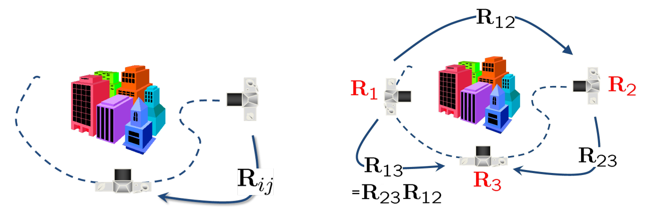
Mathematically, the single rotation averaging problem is as follows. Given rotations $\mathtt{R}_i \in \mathrm{SO}(3)$, the L_q mean, $1 \leq q \leq 2$, is equal to

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

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_q averaging problem in general has been relatively unexplored. A gradient-descent algorithm for L_1 averaging using line-search in the tangent space was given in [12]. 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 a simple geodesic L_q averaging algorithm for $\mathrm{SO}(3)$, based on the proposed L_q Weiszfeld optimization method on a Riemannian manifold.

Metrics: We consider two 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 $\mathtt{R}\mathtt{S}^{-1}$.



(a) Single Rotation Averaging (b) Multiple Rotation Averaging

Fig. 4: *Rotation Averaging: (a) represents two cameras with $\mathtt{R}_{ij}$ as a relative rotation between them. We obtain several estimates of the rotation between these two cameras and then perform averaging on them to get a better estimate. (b), we apply the rotation averaging algorithm to estimate absolute rotations, $\mathtt{R}_i$, from previously computed relative rotations $\mathtt{R}_{ij}$.*

2) The chordal metric

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

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

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.

For a more complete discussion of metrics on rotation space and rotation averaging, see [25].

4.1 L_2 averaging

The L_2 rotation averaging estimate may be used for initialization of the L_q averaging method that is the main topic of this paper. The rotation averaging problem on $\mathrm{SO}(3)$ under the L_2 norm may be solved in closed form for the chordal metric. However, there is no closed-form algorithm for L_2 rotation averaging under the geodesic metric, but convergent algorithms have been proposed [34], [12]. We give a brief description of the L_2 averaging algorithm using the chordal metric.

Computation of the L_2 chordal mean does not require knowledge of a bounding ball for the rotations $\mathtt{R}_i$, and so it is useful as a way to find an initial estimate for an iterative Weiszfeld algorithm.

Let $\mathtt{R}_{\text{sum}} = \sum_{i=1}^{k} \mathtt{R}_i$, the sum of 3×3 rotation matrices. The L_2 chordal mean $\mathtt{S}$ is obtained using the Singular Value Decomposition. Let $\mathtt{R}_{\text{sum}} = \mathtt{U}\,\mathtt{D}\,\mathtt{V}^\top$ 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 $\mathtt{S} = \mathtt{U}\,\mathrm{diag}(1, 1, -1)\mathtt{V}^\top$.

For justification of this algorithm, see [25].

4.2 L_q Geodesic mean in $\mathrm{SO}(3)$

We now consider the problem of computing the L_q geodesic mean in the group of rotations. The L_q Weiszfeld algorithm will be used to compute the minimum of the cost function

$$C_q(\mathtt{S}) = \sum_{i=1}^{k} d(\mathtt{R}_i, \mathtt{S})^q \ . \tag{17}$$

To minimize (17) we transition back and forth between the rotation manifold, and its tangent space centred at the current estimate via the exponential and logarithm maps that will be described next.

The tangent space of $\mathrm{SO}(3)$ may be indentified with the set of skew-symmetric matrices, denoted by $\mathfrak{so}(3)$. The Riemannian logarithm and exponential maps may be written in terms of the matrix exponential and logarithm as follows.

Denote by $[\mathbf{v}]_\times$ the skew-symmetric matrix corresponding to $\mathbf{v}$. The Riemannian exponential and logarithm are then defined as

$$\exp_{\mathtt{S}}([\mathbf{v}]_\times) = \exp([\mathbf{v}]_\times)\mathtt{S}$$
$$\log_{\mathtt{S}}(\mathtt{R}) = \log(\mathtt{R}\mathtt{S}^{-1})$$

where $\log$ and $\exp$ (without subscripts) represent matrix exponential and logarithm. The matrix exponential of a skew-symmetric matrix may be computed using the Rodrigues formula [24]. The distance $d(\mathtt{S}, \mathtt{R})$ is computed by

$$d(\mathtt{S}, \mathtt{R}) = (1/\sqrt{2})\| \log_{\mathtt{S}}(\mathtt{R})\|_F$$

where $\|\cdot\|_F$ represents the Frobenius norm, and the scale factor $1/\sqrt{2}$ is present so that $d(\mathtt{S}, \mathtt{R})$ is equal to the angular distance from $\mathtt{S}$ to $\mathtt{R}$. If $\log_{\mathtt{S}}(\mathtt{R}) = [v]_\times$, then $d(\mathtt{S}, \mathtt{R}) = \|\mathbf{v}\|$, in terms of the Euclidean norm in $\mathbb{R}^3$.

In terms of the matrix exponential and logarithms the update step (14)of L_q optimization, may then be written as

$$\mathtt{S}^{t+1} = \begin{cases} \exp\left(\frac{\sum_{i=1}^k w_i^t \log(\mathtt{R}_i(\mathtt{S}^t)^{-1})}{\sum_{i=1}^k w_i^t}\right)\mathtt{S}^t & \text{if } \mathtt{S}^t \notin \{\mathtt{R}_i\} \\ \mathtt{R}_j & \text{if } \mathtt{S}^t = \mathtt{R}_j \end{cases} \quad , \qquad (18)$$

where $w_i^t = 1/d(\mathtt{S}^t, \mathtt{R}_i)^{2-q}$.

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, $\hat{\mathbf{v}}$ be a unit vector and θ a scalar representing an angle. 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 the angle-axis and quaternion representation of the rotation through angle θ about the axis $\mathbf{v}$. Then the update step above may be expressed as

$$\theta_i\hat{\mathbf{v}}_i = q^{-1}(\mathbf{r}_i \cdot \bar{\mathbf{s}}^t) \ ,$$
$$\delta = \frac{\sum_{i=1}^k \theta_i\hat{\mathbf{v}}_i/\theta_i^{2-q}}{\sum_{i=1}^k 1/\theta_i^{2-q}} \ , \qquad (19)$$
$$\mathbf{s}^{t+1} = q(\delta)\cdot\mathbf{s}^t \ ,$$

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 [19] to work entirely in angle-axis space, but this is essentially equivalent to the use of quaternions.

According to Theorem 3.6 this sequence of iterates will converge to the L_q mean of the rotations $\mathtt{R}_i$, provided all the rotations and the initial extimate $\mathtt{S}^0$ lie within a ball of radius $\pi/2$.

4.3 L_q multiple rotation averaging

We now consider the problem of rotation averaging of a set of relative rotations. More specifically, let $\mathtt{R}_i$; $i = 1, \ldots, k$ 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

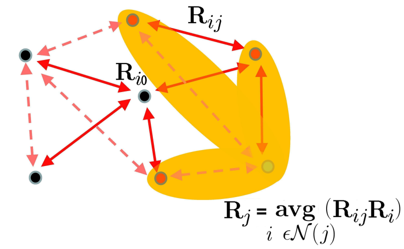


Fig. 5: *Multiple Rotation Averaging: Nodes of the above graph represent absolute rotations $\mathtt{R}_i$ and edges of the graph represent relative rotation $\mathtt{R}_{ij}$. After fixing a root node $\mathtt{R}_{i0}$ we construct a spanning tree of the graph (represented by solid arrows). For each node $\mathtt{R}_j$, we apply a single iteration of the L_q rotation averaging algorithm on its neighboring nodes $\mathcal{N}(j)$ to get an averaged estimate of $\mathtt{R}_j$. This process is repeated for every node of the graph.*

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

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

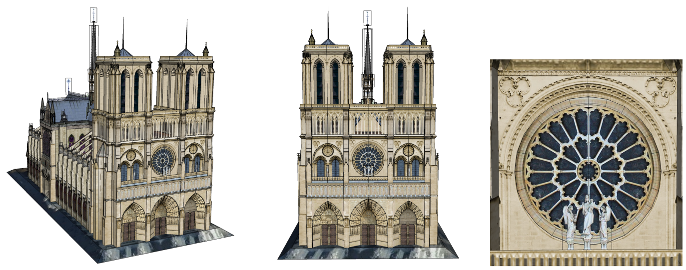
where $1 \leq q < 2$. We consider the geodesic distance function $d(\cdot,\cdot) = d_\angle(\cdot,\cdot)$. We may eliminate the obvious gauge freedom (ambiguity of solution) by setting any one of the rotations $\mathtt{R}_i$ to the identity. Generally, minimizing this cost is a difficult problem because 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_q averaging problem, and demonstrate an algorithm that gives excellent results on large data sets.

Our approach is by successive L_q 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 L_q averaging method on $\mathrm{SO}(3)$ to compute a new estimate for $\mathtt{R}_i$ by averaging the estimates $\mathtt{R}_i^{(j)}$, fig. 5. 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 L_q averaging algorithm on $\mathrm{SO}(3)$ is itself an iterative algorithm, we have the choice of running the L_q averaging 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 iteration, we choose to run a single iteration of the L_q averaging algorithm at each step. The complete algorithm is as follows.

Algorithm 1. *Given a set of relative rotations $\mathtt{R}_{ij}$ we proceed as:*

1) **Initialization:** *Set some node $\mathtt{R}_{i_0}$, with the maximum number of neighbours, to the identity rotation, and construct*



(a) Frame 1 (b) Frame 2 (c) Frame 3

Fig. 6: *Data set: (a), (b) and (c) shows multiple images of the Notre Dame cathedral.*

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 L_q averaging 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 4.2 we can not guarantee convergence of this algorithm to a global minimum, but results will demonstrate good performance.

5 Experimental Results

5.1 L_1 Rotation Averaging

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 [46] (fig. 6). 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.

5.1.1 *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 [39] 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, since the 5-point algorithm may (very rarely) return up to 10 solutions).
4) The rotation estimates were then averaged to find their L_1 mean. A closed form L_2 rotation averaging was used to find an initial estimate, followed by application of some steps of the Weiszfeld algorithm.

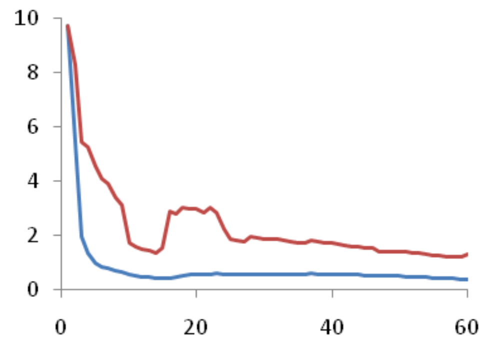


Fig. 7: *The graph shows the result of L_2 (top curve) and L_1 (bottom curve) rotation averaging, used in computing the relative orientation of two cameras from repeatedly applying the 5-point algorithm to estimate relative rotation. The graph shows the results for a single pair of images and is indicative, of the general qualitative behaviour. The plots show the error with respect to ground truth as a function of the number of samples taken. In this example, the L_1 algorithm converges in this case to close to ground truth with about 10 samples.*

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 [24].

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.

Convergence 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. 7 is shown a typical result of this estimation procedure, comparing L_1 with L_2 averaging algorithms, for increasing numbers of rotations.

Robustness for different values of q: In order to show the robustness of the proposed algorithms against outliers we test the proposed algorithms for different values of q in the presence of different proportion of outliers in the data set. We modify some percentage of image point correspondences to represent outliers. and estimate a set of rotations from these correspondences using the 5-point algorithm. We report the median rotation error with respect to ground truth. It is evident from fig. 8 that the L_q averaging algorithm for $q = 1$ is more robust to outliers than

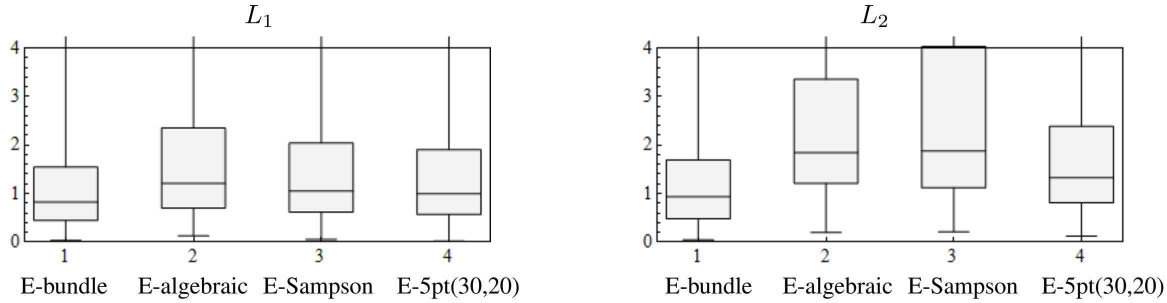


Fig. 9: *Whisker plots of the absolute orientation accuracy of the* 595 *images of the Notre Dame data set. The top and bottom of the boxes represent the 25% and 75% marks. The left graph shows the result of L_1 averaging and the right graph the L_2 averaging results. In each graph are shown the results arising from different methods of computing the essential matrices, and hence the rotations.*

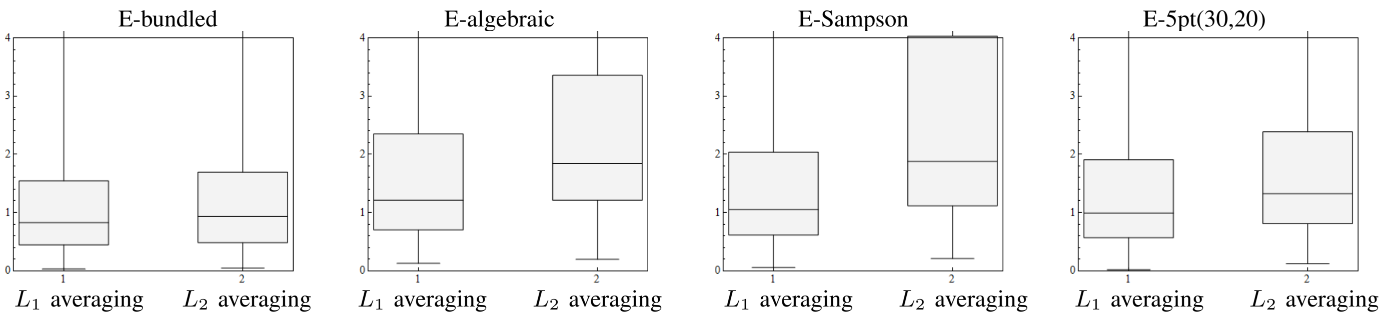


Fig. 10: *Side-by-side comparison of the results of L_1 and L_2 averaging for each of the four methods of computing relative rotations.*

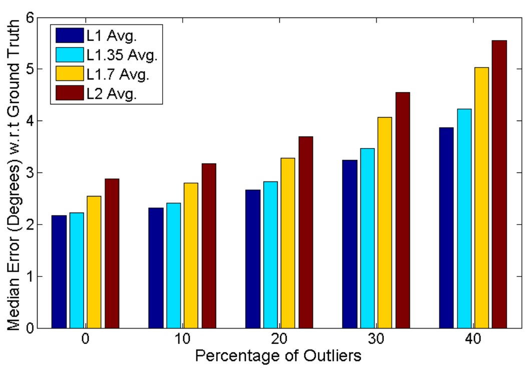


Fig. 8: *The plot shows the result of the L_q averaging method for different values of q in the presence of different percentage of outliers in the dataset. We modify some percentage of image point correspondences to represent outliers and use the* 5*-point algorithm to estimate relative rotation. Errors are computed by using the ground truth values. The above plots show that the L_q method, for $q = 1$, is more robust to outliers than the rest of the values of q and the L_2 method.*

the rest of the values of q. Furthermore, as expected, the L_2 algorithm is the least robust to outliers.

5.1.2 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 4.3.

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 quickly. 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, where each essential matrix was computed by using the five-point algorithm [39] and tested against 3 additional points, as described in the previous section. These rotation estimates were 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 $\sum_i (\mathbf{y}_i'^\top \mathtt{E} \mathbf{y}_i)^2$ was minimized iteratively over the space of all valid essential matrices. This is an adaptation of the method of [23] to essential matrices, and is very efficient and fast.
3) **E-Sampson:** The Sampson error
$$\sum_i \frac{(\mathbf{y}_i'^\top \mathtt{E} \mathbf{y}_i)^2}{(\mathtt{E}\mathbf{x})_1^2 + (\mathtt{E}\mathbf{x})_2^2 + (\mathtt{E}^\top \mathbf{x}')_1^2 + (\mathtt{E}^\top \mathbf{x}')_2^2}$$
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. Table 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.

Detailed results: The results of rotation averaging on the Notre Dame data set [47] are given in fig. 9 and fig. 10. Pairs

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.

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.

6 Proof of convergence of the L_q Weiszfeld algorithm

In this section we prove Theorem 3.6, which ensures the convergence of the proposed L_q Weiszfeld algorithm for points on a complete Riemannian manifold of non-negative sectional curvature.

It will be shown later in this section that updating a current estimate $\mathbf{x}^t$ to $\mathbf{x}^{t+1} = W(\mathbf{x}^t)$ according to (14) results in a decrease in the cost function C_q. The value of C_q decreases at each iteration, except at fixed points where its value remains constant. Of course simply because the value of the cost function decreases, that does not guarantee convergence even to a local minimum; other conditions are needed to ensure this. The Wolf conditions concerning sufficient step length [40] are one way to ensure convergence. A simpler but very general condition also guarantees convergence. Here we identify the conditions under which a descent algorithm converges. Theorem 6.7 follows from the well known Global Convergence Theorem [33, section 6.6] applied to the special case of a single valued, continuous algorithm map. We provide a short and self-contained proof for this special case.

Theorem 6.7. *Let D be a compact topological space, $C : D \to \mathbb{R}$ a continuous function defined on D. Let $W : D \to D$ be a continuous function with the property $C(W(\mathbf{x})) \leq C(\mathbf{x})$ for every $\mathbf{x} \in D$.*

Let $\mathbf{x}^0 \in D$, and $\mathbf{x}^{k+1} = W(\mathbf{x}^k)$ for $k = 0, 1, 2, \ldots$. Then the sequence $(\mathbf{x}^k)$ converges to $S = \{\mathbf{x} \mid C(W(\mathbf{x})) = C(\mathbf{x})\}$, in the sense that if $\mathcal{O}$ is an open set containing S, then there exists an N such that $\mathbf{x}^k \in \mathcal{O}$ for all $k > N$.

Proof: Choose a starting point $\mathbf{x}^0 \in D$, and denote $\mathbf{x}^k = W^k(\mathbf{x})$ for $k > 0$. This theorem states that the sequence of iterates $\mathbf{x}^k$ converges to S, assuming only that the update rule $\mathbf{x}^k \mapsto \mathbf{x}^{k+1} = W(\mathbf{x}^k)$ is continuous on D and strictly decreasing, except on S.

Since D is compact, there exists a subsequence of $(\mathbf{x}^k)$ that is convergent. Let such a subsequence be $(\mathbf{x}^{k_j})$; $j = 1, \ldots, \infty$ and let $\lim_j \mathbf{x}^{k_j} = \mathbf{y}$, which is a point in D. Then, $\lim_j C(\mathbf{x}^{k_j}) = C(\mathbf{y})$, since C is continuous, and $\lim_j C(W(\mathbf{x}^{k_j})) = C(W(\mathbf{y}))$, since $C \circ W$ is continuous.

Now, $C(\mathbf{x})$ is bounded below, for $\mathbf{x} \in D$, since D is compact. So, $C(\mathbf{x}^k)$ is a bounded non-increasing sequence in $\mathbb{R}$, and hence has a limit. Any subsequence of the $C(\mathbf{x}^k)$ must also have the same limit. In particular, $C(\mathbf{x}^{k_j})$ and $C(W(\mathbf{x}^{k_j}))$ are both subsequences of $C(\mathbf{x}^k)$ so

$$C(W(\mathbf{y})) = \lim_j C(W(\mathbf{x}^{k_j})) = \lim_k C(\mathbf{x}^k) = \lim_j C(\mathbf{x}^{k_j}) = C(\mathbf{y}) \ . \tag{20}$$

By definition of S, it follows that $\mathbf{y} \in S$. Since this argument holds for any convergent subsequence of $(\mathbf{x}^k)$, it show that any accumulation point of $\mathbf{x}^k$ lies in S.

Now consider an open set $\mathcal{O}$ containing S. The theorem is proved by showing that at most a finite number of $\mathbf{x}^k$ lie outside of $\mathcal{O}$. Suppose the contrary, and hence that there is a subsequence $(\mathbf{x}^{k_j})$ lying in $\bar{\mathcal{O}} = D - \mathcal{O}$. Since $\bar{\mathcal{O}}$ is a closed subset of a compact set, D, it is itself compact. Therefore, the sequence $(\mathbf{x}^{k_j})$ must itself contain a convergent subsequence, and this subsequence converges to a point in $\bar{\mathcal{O}}$, and hence not in S. This is a contradiction, and the proof is complete. □

The above theorem shows that the sequence $(\mathbf{x}^k)$ converges to the set S but we are interested in the conditions under which $(\mathbf{x}^k)$ is convergent to a point. We show that under the conditions of Theorem 6.8 the sequence $(\mathbf{x}^k)$ is convergent. These conditions are strictly weaker than the usually stated corollaries to the Global Convergence Theorem.

Theorem 6.8. *If in addition to Theorem 6.7, D is a metric space, S a finite or countable set and $W(\mathbf{x}) = \mathbf{x}$ for all $\mathbf{x} \in S$, then the sequence $(\mathbf{x}^k)$ is convergent to a point in S.*

Proof: If $\mathbf{x}^k$ has an accumulation point $\mathbf{y}$, then according to (20) $C(\mathbf{y}) = C(W(\mathbf{y}))$ and $\mathbf{y} \in S$. By the hypothesis of the theorem $W(\mathbf{y}) = \mathbf{y}$.

By assumption, D is compact. Let $\mathbf{y}_0$ be an accumulation point of $\mathbf{x}^k$. If $(\mathbf{x}^k)$ is not convergent, there exists $\epsilon > 0$ such that the sequence $(\mathbf{x}^k)$ enters and exits an open ball $B = B(\mathbf{y}_0, \epsilon)$ infinitely many times. There exists a subsequence $(\mathbf{x}^{k_j})$ such that $\mathbf{x}^{k_j}$ lies inside this ball, whereas $\mathbf{x}^{k_j+1}$ lies outside. Again taking a subsequence, if necessary, it may be assumed that $\mathbf{x}^{k_j}$ converges to a point $\mathbf{y}_1$ and $\mathbf{x}^{k_j+1}$ converges to a point $\mathbf{y}_2$. Clearly, $d(\mathbf{y}_1, \mathbf{y}_0) \leq \epsilon$ and $d(\mathbf{y}_2, \mathbf{y}_0) \geq \epsilon$.

From the continuity of W, we have $W(\mathbf{x}^{k_j}) \to W(\mathbf{y}_1) = \mathbf{y}_1$. However, also, $W(\mathbf{x}^{k_j}) = \mathbf{x}^{k_j+1} \to \mathbf{y}_2$ so $\mathbf{y}_1 = \mathbf{y}_2$, and $d(\mathbf{y}_1, \mathbf{y}_0) = \epsilon$.

The same thing holds for an open ball with any radius $\xi < \epsilon$; there must exist an accumulation point at any distance $\xi < \epsilon$ from $\mathbf{y}_0$, thus $\mathbf{x}^k$ has an uncountable number of accumulation points, and S is uncountable. □

Theorem 6.7 and Theorem 6.8 give simple but widely useful conditions for convergence of a descent algorithm.

6.1 Convergence: proof of Theorem 3.6

The proof of the convergence theorem takes place in several steps, according to the following outline based on Theorem 6.7 and Theorem 6.8.

Outline 1. *Given an update function W and a strictly convex function C_q, to prove that the sequence $(\mathbf{x}^t)$ obtained using $\mathbf{x}^{t+1} = W(\mathbf{x}^t)$ is convergent to the minimum of C_q we proceed as follows:*

1) *The update function W is continuous function, defined on a compact domain $\mathcal{D}$ and maps $\mathcal{D}$ to itself.*
2) *The value of C_q is non-increasing at every iteration.*
3) *The set S of Theorem 6.7 is a finite set containing the L_q minimum point and $\{\mathbf{y}_i\}$.*
4) *Since there is a finite number of accumulation points, the sequence $(\mathbf{x}^t)$ is in fact convergent, see Theorem 6.8.*
5) *If $(\mathbf{x}^t)$ converges to one of the given points, $\mathbf{y}_j$, then this is the minimum, except when $\mathbf{x}^t \in \{\mathbf{y}_i\}$ for any of the intermediate iterates.*
6) *Therefore, unless $\mathbf{x}^t$ gets stuck at $\mathbf{y}_i$, it converges to the L_q minimum.*

The proof will be completed by verifying each step of this outline in the following subsections, numbered in accord with the steps in the outline.

6.1.1 Continuous Update function W

The domain $\mathcal{D}$ is compact and lies in the open weakly convex set $B(\mathbf{o}, 2\rho)$ by Theorem 3.4. Since (as will be shown in the next point) the update function W decreases the cost function at every step, it maps $\mathcal{D}$ into $\mathcal{D}$. The update function W is continuous as a function of $\mathbf{x}$, since according to Theorem 3.3 both $\exp_{\mathbf{x}^t}(\mathbf{y}_i)$ and $\log_{\mathbf{x}^t}(\mathbf{y}_i)$ are continuous functions of $\mathbf{x}^t$. The apparent discontinuity when $\mathbf{x}^t = \mathbf{y}_i$ for some i is a removable singularity, as remarked in section 2.2.1.

6.1.2 Non-Increasing L_q Cost

Below we show in lemma 6.10 that the cost function C_q is a non-increasing function under the update function W, that is $C_q(W(\mathbf{x})) \le C_q(\mathbf{x})$ with equality only when $W(\mathbf{x}) = \mathbf{x}$. In preparation for proving lemma 6.10, the following general result establishes the relation between an L_q cost for $1 \le q < 2$ and a weighted L_2 cost.

Lemma 6.9. *If a_i and b_i are positive real numbers, $0 < q < n$ and $\sum_{i=1}^k a_i^{q-n} b_i^n \le \sum_{i=1}^k a_i^q$ then $\sum_{i=1}^k b_i^q \le \sum_{i=1}^k a_i^q$ with equality only when $a_i = b_i$ for all i.*

Proof: The proof of above lemma depends on the following simple but critical observation, which was implicitly stated (albeit in less generality) in [51]. Consider the following self-evident statement.

- *Let g be a convex function and $0 < q < n$. If $g(n) \le g(0)$ then $g(q) \le g(0)$ with strict inequality if g is strictly convex.*

This statement will now be applied to the function

$$g(t) = \sum_{i=1}^k a_i^{q-t} b_i^t \ ,$$

to prove the above lemma. To show this, one computes the second derivative of g with respect to t. The result is

$$g''(t) = \sum_{i=1}^k a_i^{q-t}\, b_i^t \,(\log a_i - \log b_i)^2 \ .$$

Since all the a_i and b_i are positive, this second derivative is positive unless $\log a_i = \log b_i$ for all i, which proves the lemma. □

Note that the above lemma is true for $0 < n < \infty$ and $q < n$ and is not limited to $n = 2$ and $1 \le q < 2$. We imposed a restriction to $n = 2$ for the problems in this paper to take advantage of the simple solutions of the least squares problems, but this result holds for a general value n.

In the update step, an optimal update is first computed in the tangent space and then this updated point is projected back to the manifold. It is therefore important to show that a decrease in the L_q cost in the tangent space also results in a decrease in the L_q cost on the manifold. The following lemma shows that the update function (14) results in a decrease in the value of C_q.

Lemma 6.10. *For the update function W defined in (14), we have $C_q(W(\mathbf{x})) \le C_q(\mathbf{x})$, where equality holds only when $W(\mathbf{x}) = \mathbf{x}$.*

Proof: The inequality will be shown for any point $\mathbf{x}^t$ and $\mathbf{x}^{t+1} = W(\mathbf{x}^t)$. Let $C_2^{\mathbf{w}}$ be a weighted L_2 function on $\mathcal{M}$,

$$C_2^{\mathbf{w}}(\mathbf{x}) = \sum_{i=1}^k w_i^t\, d(\mathbf{x}, \mathbf{y}_i)^2$$

where $w_i^t = 1/d(\mathbf{x}^t, \mathbf{y}_i)^{q-2}$. Let $\tilde{\mathbf{y}}_i$ and $\tilde{\mathbf{x}}^{t+1}$ be the corresponding points in the tangent space at $\mathbf{x}^t$, under the logarithm map. Note that $\tilde{\mathbf{x}}^t = \mathbf{0}$. A corresponding weighted L_2 cost function $\widetilde{C}_2^{\mathbf{w}}$ is defined in the tangent space $T_{\mathbf{x}^t}\mathcal{M}$ of $\mathcal{M}$, as

$$\widetilde{C}_2^{\mathbf{w}}(\tilde{\mathbf{x}}) = \sum_{i=1}^k w_i^t\, d(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}_i)^2 \ . \tag{21}$$

Note that $d(\mathbf{x}^t, \mathbf{y}_i) = d(\tilde{\mathbf{x}}^t, \tilde{\mathbf{y}}_i)$, so $\widetilde{C}_2^{\mathbf{w}}(\tilde{\mathbf{x}}^t) = C_2^{\mathbf{w}}(\mathbf{x}^t)$. The updated point $\tilde{\mathbf{x}}^{t+1}$ in the tangent space is the global minimizer of (21), so

$$\widetilde{C}_2^{\mathbf{w}}(\tilde{\mathbf{x}}^{t+1}) \le \widetilde{C}_2^{\mathbf{w}}(\tilde{\mathbf{x}}^t) = C_2^{\mathbf{w}}(\mathbf{x}^t) = C_q(\mathbf{x}^t) \ . \tag{22}$$

From the Toponogov Comparison Theorem for open weakly convex sets, Theorem 3.2, the distance between two points on a positively curved manifold is less than the distance between their images under the log map, that is $d(\mathbf{x}^{t+1}, \mathbf{y}_i) \le d(\tilde{\mathbf{x}}^{t+1}, \tilde{\mathbf{y}}_i)$. This implies that

$$C_2^{\mathbf{w}}(\mathbf{x}^{t+1}) \le \widetilde{C}_2^{\mathbf{w}}(\tilde{\mathbf{x}}^{t+1}) \le C_q(\mathbf{x}^t) \ ,$$

the last inequality from (22). The conclusion of the theorem now follows from lemma 6.9 by setting $a_i = d(\mathbf{x}^t, \mathbf{y}_i)$ and $b_i = d(\mathbf{x}^{t+1}, \mathbf{y}_i)$. □

Thus, under the update function W (14) the L_q cost function is a non-increasing function.

6.1.3 Finite set S

Under the update function W (14) the value of C_q remains constant in successive iterations either when $\mathbf{x}^t = \mathbf{y}_j$ where $W(\mathbf{y}_j) = \mathbf{y}_j$, or when $\mathbf{x}^t$ is a minimum of $C_2^{\mathbf{w}}$. By evaluating the gradients of $C_2^{\mathbf{w}}$ and C_q at a point $\mathbf{x}^t$ we get $\nabla C_2^{\mathbf{w}}(\mathbf{x}^t) = \nabla C_q(\mathbf{x}^t)$, assuming the weighted $\mathbf{w}$ are defined at $\mathbf{x}^t$. This means that if $\mathbf{x}^t$ is a minimum of $C_2^{\mathbf{w}}$, then it is also a minimum of C_q. From Theorem 3.4, the L_q minimum $\mathbf{x}^*$ is unique in B, and for this point $W(\mathbf{x}^*) = \mathbf{x}^*$. The set S in Theorem 6.7 is a finite set because it is the union of the L_q minimum point $\mathbf{x}^*$ and $\{\mathbf{y}_i\}$.

6.1.4 Convergent Sequence $(\mathbf{x}^t)$

From Theorem 6.8 the sequence $(\mathbf{x}^t)$ is convergent. When the sequence of points $(\mathbf{x}^t)$ converges to a point other than $\mathbf{y}_j$, then $(\mathbf{x}^t)$ converges to the L_q minimum $\mathbf{x}^*$. However, when the L_q minimum is one of the points $\mathbf{y}_j$ then the following section shows that this point satisfies the minimum point condition of lemma 3.5.

6.1.5 Convergence to a point $\mathbf{y}_j$

In this section we show that when $\mathbf{x}^t$ converges to one of the given points $\mathbf{y}_j$ without getting stuck at $\mathbf{y}_i$, then this point satisfies the conditions of lemma 3.5 and is a stationary point of C_q. When a minimum point is not one of the given points $\mathbf{y}_j$ then the L_q function is differentiable at the minimum point, even for $q = 1$. However, when the L_q minimum is one of the points $\mathbf{y}_j$ then the following lemma shows that for $1 < q < 2$ the gradient of the cost function vanishes at this point, while for $q = 1$ the gradient of the L_q function omitting the entry corresponding to $\mathbf{y}_j$ has a norm no greater than one.

Lemma 6.11. *For $1 \leq q < 2$, if the limit of the sequence $(\mathbf{x}^t)$ is one of the points $\mathbf{y}_j$, then $\mathbf{y}_j$ is the minimum point of C_q, except when any of the intermediate iterates $\mathbf{x}^t$ is equal to one of the $\mathbf{y}_i$ and the iteration gets stuck.*

Proof: Suppose that the sequence $\mathbf{x}^t$ converges to one of the points $\mathbf{y}_i$, which we take to be $\mathbf{y}_1$ for simplicity. Our goal is to invoke lemma 3.5 to show that $\mathbf{y}_1$ is a minimum.

Recall the definition of limit in a topological space: $x_i \to x^*$ if for every open set $\mathcal{O}$ containing x^* there exists an N such that $x_i \in \mathcal{O}$ for $i > N$.

For simplicity, write $\log(\mathbf{x}, \mathbf{y})$ instead of $\log_{\mathbf{x}}(\mathbf{y})$, and recall from Theorem 3.3 that as a function from $D \times D$ into $T\mathcal{M}$, this is continuous in both arguments. The update function is defined as

$$\mathbf{x}^{t+1} = \exp_{\mathbf{x}^t}\left(\frac{\sum_{i=1}^k w_i^t \log(\mathbf{x}^t, \mathbf{y}_i)}{\sum_{i=1}^k w_i^t}\right) ,$$

or equivalently,

$$\log(\mathbf{x}^t, \mathbf{x}^{t+1}) = \frac{\sum_{i=1}^k w_i^t \log(\mathbf{x}^t, \mathbf{y}_i)}{\sum_{i=1}^k w_i^t} ,$$

where $w_i^t = \| \log(\mathbf{x}^t, \mathbf{y}_i)\|^{q-2}$ and $\mathbf{x}^t$ is the estimate of the L_q minimum at iteration t. By a small rearrangement one sees that

$$\begin{aligned}(\log(\mathbf{x}^t, \mathbf{x}^{t+1}) - \log(\mathbf{x}^t, \mathbf{y}_1))\, w_1^t &= \\ \sum_{i=2}^k w_i^t \log(\mathbf{x}^t, \mathbf{y}_i) &- \log(\mathbf{x}^t, \mathbf{x}^{t+1}) \sum_{i=2}^k w_i^t \ .\end{aligned} \tag{23}$$

As $t \to \infty$ the limit of the right-hand side of (23) becomes

$$\sum_{i=2}^k \frac{\log(\mathbf{y}_1, \mathbf{y}_i)}{\| \log(\mathbf{y}_1, \mathbf{y}_i)\|^{2-q}} = \nabla \widehat{C}_q(\mathbf{y}_1), \tag{24}$$

with notation as in lemma 3.5. Note that this step uses the continuity of the logarithm, and also the continuity of the Riemannian metric, and hence norm on $T\mathcal{M}$. Now, turning to the left hand side, one continues:

$$\begin{aligned}\left\|\nabla \widehat{C}_q(\mathbf{y}_1)\right\| &= \lim_{t\to\infty} \left\|(\log(\mathbf{x}^t, \mathbf{x}^{t+1}) - \log(\mathbf{x}^t, \mathbf{y}_1))\, w_1^t\right\| \\ &= \lim_{t\to\infty} \left\|(\log(\mathbf{y}_1, \mathbf{x}^{t+1}) - \log(\mathbf{y}_1, \mathbf{y}_1))\, w_1^t\right\| \\ &= \lim_{t\to\infty} \frac{d(\mathbf{x}^{t+1}, \mathbf{y}_1)}{d(\mathbf{x}^t, \mathbf{y}_1)}\, d(\mathbf{x}^t, \mathbf{y}_1)^{q-1}.\end{aligned} \tag{25}$$

Once more, continuity of the logarithm in the first argument justifies the step to the second line here.

Now, the cases $q = 1$ and $q > 1$ must be dealt differently because when $q > 1$ the cost function C_q is differentiable at $\mathbf{y}_1$, whereas when $q = 1$ it is not. We use a simple observation about convergent sequences, stated here without proof.

Lemma 6.12. *Let $(\mathbf{x}^t)$ be a sequence in a metric space converging to $\mathbf{y}$. Then $\lim_{t\to\infty} d(\mathbf{x}^{t+1}, \mathbf{y})/d(\mathbf{x}^t, \mathbf{y}) \leq 1$ if the limit exists. If g^t is a sequence of real numbers such that $g^t \to 0$, then $\lim_{t\to\infty} g^t\, d(\mathbf{x}^{t+1}, \mathbf{y})/d(\mathbf{x}^t, \mathbf{y}) = 0$ if the limit exists.*

Of course, the given limits may not exist in either case. In applying this lemma to the right hand side of (25), however, the limit is known to exist and equal $\|\nabla \widehat{C}_q(\mathbf{y}_1)\|$.

Consider the case $q > 1$. Then in (25) the term $\|\mathbf{x}^t - \mathbf{y}_1\|^{q-1}$ converges to zero, since $\mathbf{x}^t \to \mathbf{y}_1$. It follows from lemma 6.12 and (25) that

$$\|\nabla C_q(\mathbf{y}_1)\| = \|\nabla \widehat{C}_q(\mathbf{y}_1)\| = 0$$

so $\mathbf{y}_1$ is a stationary point (hence global minimum) of C_q.

In the case $q = 1$, lemma 6.12 and (25) yield

$$\|\nabla \widehat{C}_q(\mathbf{y}_1)\| = \lim_{t\to\infty} \frac{d(\mathbf{x}^{t+1}, \mathbf{y}_1)}{d(\mathbf{x}^t, \mathbf{y}_1)} \leq 1$$

which is the condition given in lemma 3.5 for $\mathbf{y}_1$ to be a minimum of the cost function. □

6.1.6 Convergence to the L_q minimum

Thus, the sequence of points $(\mathbf{x}^t)$ obtained using the update function W (14) converges to the L_q minimum, except when some $\mathbf{x}^t \in \{\mathbf{y}_i\}$ for any of the intermediate iterates.

This completes the proof of Theorem 3.6.

Note: Since $\mathbb{R}^N$ is a Riemannian manifold with curvature zero and injectivity radius equal to infinity, this convergence proof is valid for points in $\mathbb{R}^N$ with the algorithm given in (15). From Theorem 3.6, if all of the given points in $\mathbb{R}^N$ are non-collinear then the cost function C_q has a unique minimum, and the sequence of points $(\mathbf{x}^t)$ either converges to the L_q minimum or gets stuck at some point $\mathbf{x}^t = \mathbf{y}_i$.

7 Remarks and Extensions

7.1 Bounds for convergence on $\mathrm{SO}(3)$

The convergence of the L_q algorithm to a global minimum was shown under the condition that all the points $\mathbf{y}_i$ lie in a convex ball of radius $\rho < r_{\mathrm{wcon}}/2$ around the initial estimate. This is probably the best possible for $\mathrm{SO}(3)$, as it shows that the algorithm will converge to the minimum if the points lie in a ball $B(\mathbf{x}^0, \pi/2)$. If this condition does not hold, then one can easily find examples where the minimum lies outside of a ball containing the $\mathbf{y}_i$; see [25].

7.2 A flexible approach

In the algorithm the logarithm map $\log_{\mathbf{x}^t}(\mathbf{y}_i)$ used is the one defined by geodesics lying in the weakly-convex set $B(\mathbf{o}, 2\rho)$. This is not always the logarithm of smallest norm, unless $\mathbf{x}$ remains inside a convex (not just weakly-convex) set. However, this can be assured as long as the $\mathbf{y}_i$ lie in a ball of radius less than $r_{\text{conv}}/2$. Such a ball is potentially half the size of the one of radius $\rho < r_{\text{wcon}}/2$ used in the theorem.

The condition may possibly be relaxed to a condition that the $\mathbf{y}_i$ lie inside a ball of radius $\rho < r_{\text{conv}}$ by addition of one extra step to the algorithm. Note that $r_{\text{conv}} \geq r_{\text{wcon}}/2$, so this may strengthen the result. The extra step uses (16). If an intermediate estimate $\mathbf{x} = \mathbf{x}^t$ lies outside of the convex ball $B(\mathbf{o}, \rho)$, then it may be replaced by the point $\mathbf{x}'$ defined in (16). The mapping $\mathbf{x} \mapsto \mathbf{x}'$ is continuous. By adding this correction to the update step, all iterations remain inside the convex ball $B(\mathbf{o}, \rho)$. Provided intermediate iterations remain in a weakly convex set, so that cost keeps decreasing, convergence will follow. More detailed analysis would be required to constitute a complete proof for this method.

7.3 More on $\mathrm{SO}(3)$

Finally, in $\mathrm{SO}(3)$, if all points $\mathbf{y}_i$ lie inside a ball of radius $\rho < r_{\text{conv}} = \pi/2$, then the update function W will map $B(\mathbf{o}, \rho)$ into itself always, so the algorithm works without modification in this case, and convergence is assured. This is because the update step may be thought of as finding a weighted centroid in the tangent space $T_{\mathbf{x}^t}\mathcal{M}$ of the points $\log_{\mathbf{x}^t} \mathbf{y}_i$. This weighted centroid must remain inside the convex hull of the $\log_{x^t} \mathbf{y}_i$, and this convex hull is mapped back by $\exp_{\mathbf{x}^t}$ to a point in the convex hull of the points $\mathbf{y}_i$, and hence back into the ball $B(\mathbf{o}, \rho)$.

7.4 More on the initial point

Generally a randomly selected starting point will result in convergence of the L_q Weiszfeld algorithm to the L_q minimum without getting stuck at $\mathbf{y}_i$. However, if such a condition occurs where $\mathbf{x}^t = \mathbf{y}_j$ then the L_q Weiszfeld algorithm, and even the original Weiszfeld algorithm, gets stuck at that point. A simple strategy to escape this situation is to move the current solution $\mathbf{x}^t$ in the descent direction and continue with the algorithm. This condition is not very likely to occur, and moreover can be avoided by a careful selection of a starting point $\mathbf{x}^0$, as explained below in Algorithm 2.

Algorithm 2. *Given a set of points, $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\} \in \mathbb{R}^N$ or $\mathcal{M}$, $k > 1$. Let $d(\mathbf{y}_i, \mathbf{y}_j)$, be the distance between two points, $\mathbf{y}_i$ and $\mathbf{y}_j$.*

1) *Among the $\mathbf{y}_i$ select the one with minimum cost: $\mathbf{x}^* = \operatorname{argmin}_j C_q(\mathbf{y}_j)$, where $C_q(\mathbf{x}) = \sum_{i=1}^{k} d(\mathbf{x}, \mathbf{y}_i)^q$.*
2) *Compute the gradient of C_q and check $\mathbf{x}^* = \mathbf{y}_j$ for the minimality condition according to lemma 3.5. If it satisfies the condition then $\mathbf{x}^*$ is the required minimum and the algorithm is complete.*
3) *Otherwise, displace $\mathbf{x}^*$ in the downhill gradient direction of $\widehat{C}_q(\mathbf{x})$ to obtain $\mathbf{x}^0$. Backtrack if necessary to ensure $C_q(\mathbf{x}^0) < C_q(\mathbf{x}^*)$.*
4) *Repeat, $\mathbf{x}^{t+1} = W(\mathbf{x}^t)$ until convergence, where W is defined in (14).*

The initial point $\mathbf{x}^0$ so found has cost less than any of the points $\mathbf{y}_i$, and iterations of the algorithm from $\mathbf{x}^0$, can not again approach any of the $\mathbf{y}_j$. Thus Algorithm 2 ensures that the non-differentiability condition never occurs by a careful selection of a starting point for the algorithm. Hence the L_q Weiszfeld algorithm is guaranteed to converge to the minimum of $C_q(\mathbf{x})$.

8 Conclusion

This paper presents a theoretical proof for the convergence of the L_q Weiszfeld algorithm that achieves an L_q minimum by iteratively minimizing a weighted L_2 function, where $1 \leq q < 2$. Ease of implementation makes the proposed algorithm attractive wherever L_q optimization is desired.

The L_q Weiszfeld-based averaging method gives good results, both for single-view averaging of minimal-case rotation estimates, and iteratively for multiple-view reconstruction. 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 [12]. 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 of time. In fact, the time taken for averaging is far smaller than the time required to generate the individual rotation estimates.

An interesting observation is that the L_1 methods tend to emphasize (and largely ignore) outliers by allowing large individual errors to occur, whereas L_2 methods will strive to keep all errors low. For this reason the L_1 methods may be useful in identifying outliers. Interesting future work will be to extend the breadth of problems in which the L_q Weiszfeld Algorithm is applicable.

9 Acknowledgments

This research has been funded by National ICT Australia. National ICT Australia 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.

Appendix A

A.1 Continuity of Logarithm map: Proof of Theorem 3.3

In this section we give a proof of Theorem 3.3, which shows the continuity of a logarithm map defined on an open weakly convex set. If W is an open weakly convex set in a complete Riemannian manifold $\mathcal{M}$, and $\mathbf{x}, \mathbf{y}$ are two points in W, define $\log_{\mathbf{x}}(\mathbf{y})$ to be the vector $\mathbf{v}$ in $T_{\mathbf{x}}\mathcal{M} \subset T\mathcal{M}$ such that $\exp_{\mathbf{x}}(\mathbf{v}) = \mathbf{y}$, and $\exp_{\mathbf{x}}(t\mathbf{v}) \in W$ for all $t \in [0, 1]$. Then $\log_{\mathbf{x}}(\mathbf{y})$ as a map from $W \times W$ to $T\mathcal{M}$ is continuous in both variables.

Proof: It is well known (e.g. [30, p. 107]) that the Riemannian exponential is continuous as a map $\exp\colon T\mathcal{M} \to \mathcal{M}$. Let $\pi : T\mathcal{M} \to \mathcal{M}$ be the projection to the base space of the tangent bundle, and define the continuous map $\overline{\exp}\colon T\mathcal{M} \to \mathcal{M} \times \mathcal{M}$ by $X \mapsto (\pi(X), \exp(X))$. Note that X is mapped to the start and end points of the geodesic defined by X.

Now let $W \subset \mathcal{M}$ be an open weakly convex set in $\mathcal{M}$, and for each $s \in [0, 1]$ define

$$O_s^W = \{X \in T\mathcal{M} \mid \overline{\exp}(sX) \in W \times W\},$$

where sX is in the tangent space $T_{\pi(X)}$. Further, define

$$O_{\cup}^{W} = \{(s, X) \in [0,1] \times T\mathcal{M} \mid \overline{\exp}(sX) \in W \times W\} \ .$$

Each O_s^W can be seen as a cross-sectional slice of $O_{\cup}^W$, which itself is a union of all the slices for $s \in [0,1]$. Since $\overline{\exp}(sX)$ is continuous in both s and X, and $W \times W$ is open, $O_{\cup}^W$ is open in $[0,1] \times T\mathcal{M}$. Next, define

$$\begin{aligned} O_{\cap}^{W} &= \{X \in T\mathcal{M} \mid \overline{\exp}(sX) \in W \times W \ \text{ for all } \ s \in [0,1]\} \\ &= \bigcap_{s \in [0,1]} O_s^W \ . \end{aligned}$$

This is an infinite intersection, but since $[0,1]$ is compact, and $O_{\cup}^W$ is open, it follows that $O_{\cap}^W$ is open in $T\mathcal{M}$.

Note that for $s \in [0,1]$, $\exp(sX)$ traces out the (unique) geodesic in W from $\pi(X)$ to $\exp(X)$. Therefore, $\overline{\exp}$ is an injective map from $O_{\cap}^W$ onto $W \times W$, and we can define a *Riemannian logarithm*, $\log_W : W \times W \to O_{\cap}^W$ by

$$\log_W(p, q) = \overline{\exp}^{-1}(p, q).$$

Since $\log_W(p,q) \in T_p\mathcal{M}$ and $\exp_p(\log_W(p,q)) = q$ by definition, this is a well defined two-variable version of the usual pointwise Riemannian logarithm.

We show that $\log_W(\cdot,\cdot)$ is continuous by applying *invariance of domain* (e.g. [14, Proposition IV.7.4]), which states that an injective continuous mapping from $\mathbb{R}^m$ to $\mathbb{R}^m$ has a continuous inverse (restricted to its image). This can be applied to the mapping $\overline{\exp}$, which is an injective mapping between two open sets $O_{\cap}^W$ and $W \times W$. Although these sets are not subsets of any $\mathbb{R}^m$, they are both open subsets of $2n$-dimensional manifolds $T\mathcal{M}$ and $W \times W$ respectively. Since continuity is a local property, and the manifolds are locally homeomorphic to $\mathbb{R}^{2n}$, the result follows. □

A.2 Toponogov's Theorem for $\kappa \geq 0$: Proof of Theorem 3.2

We now give a proof of Theorem 3.2 that compares distances in an open weakly convex set in a Riemannian manifold of non-negative sectional curvature with distances in the tangent space. Let W be an open weakly convex set in $\mathcal{M}$, a manifold of non-negative sectional curvature. Let $\mathbf{q}$, $\mathbf{p}_1$ and $\mathbf{p}_2$ be three points in W. Then, Theorem 3.2 states that the following inequality holds,

$$d(\mathbf{p}_1, \mathbf{p}_2) \leq d(\log_{\mathbf{q}}(\mathbf{p}_1), \log_{\mathbf{q}}(\mathbf{p}_2)) = \| \log_{\mathbf{q}}(\mathbf{p}_1) - \log_{\mathbf{q}}(\mathbf{p}_2) \| \ .$$

Proof of Theorem 3.2: Consider a *hinge* in W consisting of the two geodesics from $\mathbf{q}$ to $\mathbf{p}_1$ and $\mathbf{p}_2$, meeting at $\mathbf{q}$. Under the logarithm map $\log_{\mathbf{q}}$ the three points map to $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{p}}_i$, where $\tilde{\mathbf{q}} = \mathbf{0}$ in the vector space $T_{\mathbf{q}}\mathcal{M}$. These three points form a hinge with the same parameters as the hinge formed by $\mathbf{q}$, $\mathbf{p}_1$ and $\mathbf{p}_2$ in W, namely, $d(\mathbf{q}, \mathbf{p}_i) = d(\tilde{\mathbf{q}}, \tilde{\mathbf{p}}_i)$, and $\angle(\mathbf{q}\mathbf{p}_i, \mathbf{q}\mathbf{p}_2) = \angle(\tilde{\mathbf{q}}\tilde{\mathbf{p}}_i, \tilde{\mathbf{q}}\tilde{\mathbf{p}}_2)$.

The desired conclusion is then seen as a particular case of Toponogov's theorem [9] which states that $d(\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2) \geq d(\mathbf{p}_1, \mathbf{p}_2)$.

However, Toponogov's theorem is true only under certain conditions. One required condition is that one of the geodesics $\mathbf{q}\,\mathbf{p}_i$ is minimizing. In the present case, this is true in the sense that the geodesic is minimizing in W, but not necessarily in $\mathcal{M}$. A further condition, seemingly always present in statements of the theorem, is that the manifold is geodesically complete, which may be true for $\mathcal{M}$, but it is not true for W. Therefore, we cannot apply Toponogov's theorem directly in its usual stated form, either to $\mathcal{M}$ or to W. Consequently, it is necessary to verify that Toponogov's theorem holds in this particular case. In particular, the usual assumptions of geodesic completeness and globally minimizing geodesics are not necessary if we limit the analysis to a weakly convex domain (which excludes the presence of conjugate points).

We sketch a proof below, based on the proof given in [36]. Since the main outline of the proof is the same as in that paper, it is sufficient to give a somewhat brief proof here, referring the reader to [36] for more details.

The proof set out in [36] can be significantly simplified for the present purposes. First, Meyer deals with the case where the sectional curvature satisfies $\kappa \geq \kappa_0$. We are only interested in the case $\kappa \geq 0$, which simplifies things. Secondly, Meyer accounts for the case when the distance function is not differentiable. This can occur when there are conjugate points along the geodesics, and it complicates the proof substantially. In our case, because of the absence of conjugate points in W, this complication can be avoided.

The proof is set out in three steps.

Step 1. Triangle distance inequality. Consider the triangle formed by points $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{q}$ in W and the geodesics joining them. Let $\tilde{\mathbf{p}}_1$, $\tilde{\mathbf{p}}_2$ and $\tilde{\mathbf{q}}$ be the vertices of a triangle in $\mathbb{R}^n$ with the same side lengths (note: this is not a corresponding hinge, in the sense that the angles are not the same). Existence of this triangle is guaranteed because the sides of the triangle in W satisfy the triangle inequality, according to lemma 3.1. Now, parametrize the two edges $\mathbf{p}_1\mathbf{p}_2$ and $\tilde{\mathbf{p}}_1\tilde{\mathbf{p}}_2$ by arc-length; they have the same length by construction. Let $\mathbf{p}_t$ and $\tilde{\mathbf{p}}_t$ be corresponding points with the same parameter value t. The first step of the proof is to show that $d(\mathbf{q}, \mathbf{p}_t) \geq d(\tilde{\mathbf{q}}, \tilde{\mathbf{p}}_t)$.

For a fixed $\mathbf{q}$, define $f(\mathbf{p}) = d(\mathbf{q}, \mathbf{p})$ and $g(\mathbf{p}) = f(\mathbf{p})^2/2$. Similarly, define $\tilde{f}$ and $\tilde{g}$ in terms of distances in $\mathbb{R}^n$. The major technical point concerns the Hessian of the function $g(\mathbf{p})$ in $\mathcal{M}$. Provided $\kappa \geq 0$ in W, the following operator inequality holds.

$$\mathrm{Hess}_g \leq \mathtt{I} \ . \tag{26}$$

where $\mathtt{I}$ is the identity. This is to be interpeted as meaning that if Hess_g is expressed as a matrix in a local Riemannian coordinate system, then, $\mathtt{I} - \mathrm{Hess}_g$ is positive semi- definite. Therefore, for two vector fields X and Y, the inequality $\mathrm{Hess}_g(X,Y) \leq \langle X, Y \rangle$ holds.

This result is shown to hold for a "local" distance function (such at the distance d_W defined as geodesic length in the open set W), provided the path from $\mathbf{q}$ to $\mathbf{p}$ does not extend beyond the first conjugate point. Since there are no conjugate points in W, this inequality holds there. This is the main technical part of the proof (for details see [36]), and the only place where Riemannian geometry, or any information about the curvature is used.

Now, let $\gamma(t)$ be a geodesic in $\mathcal{M}$, parametrized by arc length, and let $h(t) = g(\gamma(t))$; thus, h is a function from $\mathbb{R}$ to $\mathbb{R}$. Its second derivative is given by

$$h''(t) = \mathrm{Hess}_g(\gamma', \gamma') \leq \langle \gamma', \gamma' \rangle = 1 \ .$$

On the other hand, for a straight line in R^n, it is easily computed that $\tilde{h}''(t) = 1$.

Now, apply this to the geodesic γ, defined for $t \in [0, c]$, joining $\mathbf{p}_1$ to $\mathbf{p}_2$. Similarly, let $\tilde{\gamma}$ be the line (of the same length) joining

$\tilde{\mathbf{p}}_1$ to $\tilde{\mathbf{p}}_2$ in R^n. With $h = g \circ \gamma$ and $\tilde{h} = \tilde{g} \circ \tilde{\gamma}$ defined as above, let $\lambda(t) = h(t) - \tilde{h}(t)$.

We wish to prove that $\lambda(t) \geq 0$ for all $t \in [0, c]$. This means that $g(\gamma(t)) \geq \tilde{g}(\tilde{\gamma}(t))$, or $f(\gamma(t)) \geq \tilde{f}(\tilde{\gamma}(t))$. In other words,

$$d(\tilde{\mathbf{q}}, \tilde{\gamma}(t)) \leq d(\mathbf{q}, \gamma(t)) \ . \tag{27}$$

Since the two triangles have sides of the same length, it follows that $h(0) = \tilde{h}(0) = \text{dist}(\mathbf{q}, \mathbf{p}_1)$. Similarly, $h(c) = \tilde{h}(c) = \text{dist}(\mathbf{q}, \mathbf{p}_2)$. Thus $\lambda(0) = \lambda(c) = 0$. The proof is completed by showing that λ is a concave function. This follows from the estimates of $h''(t) \leq 1$ and $\tilde{h}''(t) = 1$, since they imply that $\lambda''(t) \leq 0$.

Step 2. Angle inequality The next step is to extend this length inequality to any chord of the triangle, that is, any geodesic between points on two sides of the triangle. Thus, let $\mathbf{a}_s$ be a point on the edge $\mathbf{q}\mathbf{p}_1$ and $\tilde{\mathbf{a}}_s$ the corresponding point on the side $\tilde{\mathbf{q}}\tilde{\mathbf{p}}_1$. Then applying the argument of the previous section to the triangle $\mathbf{q}\mathbf{p}_1\mathbf{p}_t$ a simple argument (see [36]) shows that $d(\mathbf{a}_s, \mathbf{p}_t) \geq d(\tilde{\mathbf{a}}_s, \tilde{\mathbf{p}}_t)$. By letting these points approach the vertex $\mathbf{p}_1$ one deduces that $\tilde{\alpha}_1 \leq \alpha_1$ where these are the angles of the triangle at $\tilde{\mathbf{p}}_1$ and $\mathbf{p}_1$ respectively.

By symmetry, this inequality holds equally well for all angles of the triangle.

Step 3. Hinge inequality Toponogov's hinge inequality now follows directly from the following lemma.

Lemma A.13. *In $\mathbb{R}^n$, let $\mathbf{a}$ and $\mathbf{b}$ be points on two lines meeting at a point $\mathbf{c}$. Let α be the angle between the two lines. Then $d(\mathbf{a}, \mathbf{b})$ increases monotonically as α increases from 0 to π.*

Since the angle $\angle\tilde{\mathbf{q}} \leq \angle\mathbf{q}$, but $d(\mathbf{p}_1, \mathbf{p}_2) = d(\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2)$, it follows that if the angle $\angle\tilde{\mathbf{q}}$ is increased to equal $\angle\mathbf{q}$, then the distance $d(\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2)$ is increased as well, giving the required result.

References

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

[2] Bijan Afsari, Roberto Tron, and René Vidal. On the convergence of gradient descent for finding the Riemannian center of mass. *SIAM Journal on Control and Optimization*, 2013.

[3] K. Aftab, R. Hartley, and J. Trumpf. Lq-closest-point to affine subspaces using the Generalized Weiszfeld algorithm. in preparation.

[4] J Brimberg and R Chen. A note on convergence in the single facility minisum location problem. *Computers & Mathematics with Applications*, 1998.

[5] Jack Brimberg. Further notes on convergence of the Weiszfeld algorithm. *Yugoslav Journal of Operations Research*, 2003.

[6] Jack Brimberg and Robert F Love. Global convergence of a generalized iterative procedure for the minisum location problem with lp distances. *Operations Research*, 1993.

[7] R. Chartrand and Wotao Yin. Iteratively reweighted algorithms for compressive sensing. In *Proc. IEEE international conference on Acoustics, speech and signal processing*, 2008.

[8] I. Chavel. *Riemannian Geometry*. Cambridge University Press, 2nd edition, 2006.

[9] Jeff Cheeger and David G Ebin. *Comparison theorems in Riemannian geometry*. AMS Bookstore, 1975.

[10] Mihai Cucuringu, Yaron Lipman, and Amit Singer. Sensor network localization by eigenvector synchronization over the euclidean group. *ACM Transactions on Sensor Networks*, 2012.

[11] Mihai Cucuringu, Amit Singer, and David Cowburn. Eigenvector synchronization, graph rigidity and the molecule problem. *Information and Inference*, 2012.

[12] Yuchao Dai, Jochen Trumpf, Hongdong Li, Nick Barnes, and Richard Hartley. Rotation averaging with application to camera-rig calibration. In *Proc. Asian Conference on Computer Vision*, 2009.

[13] I. Daubechies, R. DeVore, M. Fornasier, and S. Gunturk. Iteratively reweighted least squares minimization: Proof of faster than linear rate for sparse recovery. In *Proc. Information Sciences and Systems*, 2008.

[14] A. Dold. *Lectures on Algebraic Topology*. Springer, 1995.

[15] U. Eckhardt. Weber's problem and weiszfeld's algorithm in general spaces. *Mathematical Programming*, 1980.

[16] Y.C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Transactions on Information Theory*, 2009.

[17] P. Thomas Fletcher, Suresh Venkatasubramanian, and Sarang Joshi. The geometric median on riemannian manifolds with applications to robust atlas estimation. *Neuroimage*, 2009.

[18] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[19] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[20] David Groisser. Newton's method, zeroes of vector fields, and the riemannian center of mass. *Advances in Applied Mathematics*, 2004.

[21] R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the Weiszfeld algorithm. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[22] R. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[23] R. I. Hartley. Minimizing algebraic error. *Philosophical Transactions of the Royal Society of London, SERIES A*, 1998.

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

[25] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 2013.

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

[27] R. Kaucic, R. I. Hartley, and N. Y. Dano. Plane-based projective reconstruction. In *Proc. International Conference on Computer Vision*, 2001.

[28] W. Klingenberg. *Riemannian Geometry*. Walter de Gruyter, 1982.

[29] Harold W Kulin and Robert E Kuenne. An efficient algorithm for the numerical solution of the generalized weber problem in spatial economics. *Journal of Regional Science*, 1962.

[30] S. Lang. *Fundamentals of Differential Geometry*. Springer, 1999.

[31] Huiling Le. Locating fréchet means with application to shape spaces. *Advances in Applied Probability*, 2001.

[32] John M Lee. *Riemannian manifolds: an introduction to curvature*. Springer, 1997.

[33] David G Luenberger. *Linear and nonlinear programming*. Springer, 2003.

[34] Jonathan H. Manton. A globally convergent numerical algorithm for computing the centre of mass on compact Lie groups. In *Proc. International Conference on Control, Automation, Robotics and Vision*, 2004.

[35] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[36] Wolfgang Meyer. Toponogovs theorem and applications. *Lecture Notes, Trieste*, 1989.

[37] Maher Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 2002.

[38] S.B. Myers. Arcs and geodesics in metric spaces. *Transactions of the American Mathematical Society*, 1945.

[39] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[40] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Verlag, New York, 1999.

[41] Peter Petersen. *Riemannian geometry, Second Edition*. Springer, 2006.

[42] Frank Plastria and Mohamed Elosmani. On the convergence of the Weiszfeld algorithm for continuous single facility location–allocation problems. *Top*, 2008.

[43] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery. In *Proc. International Conference on Computer Vision*, 2001.

[44] Alain Sarlette and Rodolphe Sepulchre. Consensus optimization on manifolds. *SIAM Journal on Control and Optimization*, 2009.

[45] K. Sim and R. Hartley. Recovering camera motion using L_∞ minimization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[46] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *Proc. ACM SIGGRAPH*, 2006.

[47] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 2008.

[48] Roberto Tron and René Vidal. Distributed image-based 3-d localization of camera sensor networks. In *Proc. IEEE Conference on Decision and Control*, 2009.
[49] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *arXiv preprint arXiv:1211.2441*, 2012.
[50] Alfred Weber. *Über den Standort der Industrien, Erster Teil: Reine Theorie des Standortes*. 1909.
[51] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal*, 1937.
[52] E Weiszfeld and Frank Plastria. On the point for which the sum of the distances to n given points is minimum. *Annals of Operations Research*, 2009.
[53] L. Yang. Riemannian median and its estimation. *LMS Journal of Computation and Mathematics*, 2010.

Khurrum Aftab received Bachelor degree in Electronics Engineering from International Islamic University Islamabad (IIUI), Pakistan, in 2007, and Master degree in Computer Science from the Lahore University of Management Sciences (LUMS), Pakistan, in 2009. He also remained an exchange student at the University of York, UK, in 2009. Since 2010, he is a Ph.D. student at the Australian National University (ANU). He has also been a visiting research fellow at the National Institute of Informatics (NII), Japan. His research interests are computer vision and convex optimization.

Richard Hartley (Fellow) is a member of the computer vision group in the Research School of Engineering, at the Australian National University, where he has been since January, 2001. He is also a member of the computer vision research group in NICTA, a government funded research laboratory.

Dr. Hartley worked at the General Electric Research and Development Center from 1985 to 2001, working first in VLSI design, and later in computer vision. He became involved with Image Understanding and Scene Reconstruction working with GE's Simulation and Control Systems Division.

He is an author (with Andrew Zisserman) of the book Multiple View Geometry in Computer Vision.

Jochen Trumpf (M'04) received the Dipl.-Math. and Dr. rer. nat. degrees in mathematics from the University of Würzburg, Germany, in 1997 and 2002, respectively. He is currently a Fellow in the Research School of Engineering at the Australian National University. His research interests include observer theory and design, linear systems theory and optimisation on manifolds with applications in robotics, computer vision and wireless communication.