

A Survey on Rotation Optimization in Structure from Motion

Roberto Tron*
 Boston University

Xiaowei Zhou*
 University of Pennsylvania

Kostas Daniilidis*
 University of Pennsylvania

Abstract

We consider the problem of robust rotation optimization in Structure from Motion applications. A number of different approaches have been recently proposed, with solutions that are at times incompatible, and at times complementary. The goal of this paper is to survey and compare these ideas in a unified manner, and to benchmark their robustness against the presence of outliers. In all, we have tested more than forty variants of these methods (including novel ones), and we find the best performing combination.

1. Introduction

The problem of reconstructing a 3-D scene from a series of images is known as the Structure from Motion (SfM) problem. It is a classical computer vision problem, with the following standard pipeline [19]:

1. Extract features from the images, and find pairwise matches between every pair of images.
2. Use a robust method (RANSAC) to fit an essential matrix (e.g., [17]), reject outliers and estimate pairwise relative rotations and translation directions.
3. Combine some of the pairwise estimates to obtain initial absolute poses and 3-D geometry.
4. Use bundle adjustment to refine the estimate.

In the last decade, the focus of the research in SfM has been in scaling this basic pipeline to large, heterogeneous datasets (e.g., crowd-sourced image collections from the Internet), producing impressive results (see, for instance, [3, 4, 14, 25]). One problem of high practical relevance is that the second step of the pipeline (RANSAC fitting and outlier rejection) might produce erroneous estimates (i.e., match two unrelated images or give a vastly erroneous pose). This introduces *outliers* in the third step of the pipeline (initial estimation of pose and geometry). If not properly handled, these outliers

introduces biases in the entire reconstruction which cannot be recovered with bundle adjustment.

To avoid this, a number of techniques have been explored. The first solutions (as in the papers cited above) are *sequential*: starting from a small set of “good” images (e.g., images with high inliers counts), they iterate the last two steps of the pipeline while adding one or a few images at every iteration. The results of this method might depend on the sequence used to add the images, and there might be drifts due to error build-up [12]. Another approach is given by *skeletal* solutions [26], where heuristics are used to pick a set of images that “spans” the reconstruction. First, the skeleton is reconstructed, and then all the remaining images are added. Both techniques rely on the use of intermediate 3-D reconstructions, and effectively use only a subset of all the available pairwise poses, while employing heuristics to avoid outliers. A more recent trend is to use a *pose-graph* approach, which does not directly include the 3-D structure, and instead tries to find the absolute poses that best fit the relative pairwise measurements. This technique has emerged not only in computer vision, but also in robotics and control systems [10]. As we will discuss in this paper, it is possible to use robust fitting techniques to use all the pairwise estimates at the same time while reducing the influence of outliers. Moreover, one can further decompose the problem into two subproblems, one for finding the rotations alone and the other for finding translations with given rotations.

The goal of this paper is to survey and benchmark the most promising solutions for the *rotations optimization* subproblem, i.e., finding absolute rotations from relative measurements. We have identified three families of methods:

1. Global, factorization-based methods: all the relative measurements are collected in a matrix which is then factorized into absolute poses. These methods give globally optimal solutions (obtained by solving a convex program or from a Singular Value Decomposition) after some relaxation of the orthonormality constraints.
2. Local, iterative methods: these methods minimize a non-convex robust fitting cost starting from an initial solution while respecting the geometry of the space of rotations. They are based on local gradient information, and require good initializations.

*The authors are grateful for the support of the following grants: NSF-DGE-0966142 (IGERT), NSF-IIS-1317788, NSF-IIP-1439681 (I/UCRC), NSF-IIS-1426840, ARL MAST-CTA W911NF-08-2-0004, ARL RCTA W911NF-10-2-0016, ONR N000141310778.

3. Inference-based methods: these techniques do not output a result in terms of rotations, but label each edge as an outlier or not. They exploit the fact that measurements composed around a cycle in the graph of rotations should (approximately) give the identity rotation.

It is interesting to notice that these families treat the differential geometry of the space of rotations differently. Global methods embed all the rotations into a single subspace, local methods use the Riemannian manifold structure of the space while inference-based methods use its group structure. As such, one might expect that these different families have complementary strengths and weaknesses. In this paper we test a few representatives from each family (including existing and novel variants) and some of their combinations. Our goal is to find the best performing methods in terms of robustness to outliers. The only existing analysis with a similar scope is [10]. That paper, however, did not consider some of the more advanced methods (such as those based on low-rank priors), and did not explicitly evaluate the methods against the presence of outliers.

2. Notation and preliminaries

We model the set of images as an undirected graph $G = (V, E)$, where the vertices $V = \{1, \dots, N\}$ represent the images and the edges $E \subseteq V \times V$ represent pairs of images for which RANSAC was able to find a pose with a large enough support (25 correspondences in our implementation). We use $\deg(i)$ to number of neighbors (degree) of node $i \in V$. We denote as R_i the rotation component of the rigid body transformation from camera i to world coordinates, and as \tilde{R}_{ij} the measured relative rotation from the coordinates in camera j to those in camera i . For ideal (noiseless) measurements, we have $\tilde{R}_{ij} = R_i R_j^T$. We denote the Lie group rotations as $SO(3)$, which has the group structure given by the usual matrix multiplication, and a Riemannian manifold structure giving a geodesic distance $d_{SO(3)}(\cdot, \cdot)$. Moreover, we use $\|\cdot\|_F$ and $\|\cdot\|_1$ to denote, respectively, the L2 (Frobenius) and L1 (sum of absolute values) vector norms of a matrix. We use $[A]_{ij;3 \times 3}$ to indicate the three-by-three block in the i -th block-row and j -th block-column of a matrix A . Finally, we define the projection of a matrix $A \in \mathbb{R}^{3 \times 3}$ to $SO(3)$ as

$$\text{proj}_{SO(3)}(A) = U \text{diag}(1, 1, \det(UV^T)) V^T, \quad (1)$$

where $A = U \Sigma V^T$ is the Singular Value Decomposition (SVD) of A .

3. Problem formulation

The general formulation for the rotation optimization problem is given by:

$$\min_{\{R_i\}_{i \in V} \in SO(3)^N} \sum_{(i,j) \in E} \ell(R_i, \tilde{R}_{ij} R_j), \quad (2)$$

where ℓ is a loss function which is zero when the two arguments are the same. For ideal measurements, and using the group properties of rotations, (2) is equivalent to

$$\min_{\{R_i\}_{i \in V} \in SO(3)^N} \sum_{(i,j) \in E} \ell(R_i R_i^T, \tilde{R}_{ij}) \quad (3)$$

We consider below variants of this general formulation.

4. Datasets and testing protocol

We are interested in evaluating the performance of different methods under a varying percentage of outliers. In order to perform this analysis, we need to use synthetic datasets. However, we would like to use graph topologies and poses that are realistic. For this reason, we use the ground-truth rotations that are provided with the datasets of [27], and obtain the pose-graph from the corresponding images. Table 1 contains a summary of the characteristics of these datasets.

For each dataset, we start from the set of relative rotations obtained from the ground-truth poses and then introduce corruptions with a increasing percentage of outliers (from 0% to 80% in 5% increments). The location of the outliers (*i.e.*, the subset of edges in E) is chosen uniformly at random. The outlying measurements are obtained by corrupting the ground-truth with a rotation with a random angle between 60 and 90 degrees and a random axis (to approximate the distribution of real outliers). For each method we test, we collect all the distances between the relative rotations in the result and the ground truth (we do not use the absolute poses because, in the presence of outliers, the alignment with the ground truth is not generally straightforward). These errors are aggregated over 100 random outlier realizations. In all the plots we will show, we report the mean (in solid lines) and median (in dashed lines) of these aggregated errors.

5. Global, factorization-based methods

All the methods considered in this section share the ideas of considering the unknown absolute rotations in a single stacked matrix $\mathbf{R} = \text{stack}(\{R_i\}_{i \in V}) \in \mathbb{R}^{3N \times 3}$, and neglecting the non-linear constraints $R_i^T R_i = I$ for all $i \in V$.

Name	# poses	# edges	% edges
Fountain-P11	11	23	41.82
Herz-Jesu-P8	8	13	46.43
Herz-Jesu-P25	25	73	24.33
Castle-P19	19	33	19.30
Castle-P30	30	110	25.29
Entry-P10	10	14	31.11

Table 1: Datasets used to obtain the ground truth poses and graph topology. The percentage of edges is computed over the total number of possible pairs.

This leads to optimization problems that can be solved globally (even in the presence of outliers), obtaining a (possibly) approximate solution \mathbf{R} . The final estimates are then obtained by using projections:

$$R_i = \text{proj}_{SO(3)}([\mathbf{R}]_{i;3 \times 3}), \quad i \in V. \quad (4)$$

The various methods in how the matrix \mathbf{R} is obtained and in how the constraints are relaxed.

5.1. Linear methods

These methods obtain \mathbf{R} from the SVD of some matrix containing the measurements $\{\tilde{R}_{ij}\}$, and use the loss $\ell(R_1, R_2) = \|R_1 - R_2\|_F^2$ in (2). Each term of the cost can be expressed in different ways:

$$\|R_i - \tilde{R}_{ij}R_j\|_F^2 \quad (5)$$

$$= \|R_i\|_F^2 - 2\text{tr}(R_i^T \tilde{R}_{ij}R_j) + \|R_j\|_F^2 \quad (6)$$

$$= \text{tr} \left(\begin{bmatrix} R_i \\ R_j \end{bmatrix}^T \begin{bmatrix} I & -\tilde{R}_{ij} \\ -\tilde{R}_{ij}^T & I \end{bmatrix} \begin{bmatrix} R_i \\ R_j \end{bmatrix} \right) \quad (7)$$

$$= 6 - 2\text{tr}(\tilde{R}_{ij}^T R_i R_j^T). \quad (8)$$

Note that (7) and (8) are equivalent only if R_i, R_j are orthonormal matrices. We then define the two matrices $\tilde{L}, \tilde{G} \in \mathbb{R}^{3N \times 3N}$ as

$$[\tilde{L}]_{ij;3 \times 3} = \begin{cases} \deg(i)I & \text{if } i = j, \\ -\tilde{R}_{ij} & \text{if } (i, j) \in E, \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

$$[\tilde{G}]_{ij;3 \times 3} = \begin{cases} I & \text{if } i = j, \\ \tilde{R}_{ij} & \text{if } (i, j) \in E, \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

with the convention that $\tilde{R}_{ji} = \tilde{R}_{ij}^T$. The matrix \tilde{L} is sometimes referred to as the *Graph Connection Laplacian* [24]. We can use (7) and (8) to rewrite (2) as:

$$\min \text{tr}(\mathbf{R}^T \tilde{L} \mathbf{R}), \quad (11)$$

$$\max \text{tr}(\mathbf{R}^T \tilde{G} \mathbf{R}), \quad (12)$$

and use the constraint $\mathbf{R}^T \mathbf{R} = I$ instead of considering $R_i \in SO(3), i \in V$. Note that this effectively changes the domain of the problem from the manifold $SO(3)^N$ to the Stiefel manifold $V_3(\mathbb{R}^{3N})$. We call (11) the *Linear L2* formulation and (12) the *Linear Trace* formulation. They were first introduced by [21] and [5], respectively. Note that (11) and (12) are no longer equivalent (this is because the equality in (8) does not hold). With the relaxed constraint, problem (11) (resp., (12)) can be solved in closed form after computing the SVD of \tilde{L} (resp., \tilde{G}) and setting \mathbf{R} to be equal to the three singular vector corresponding to the bottom (resp., top) singular values.

Variant 1: Inspired by normalized cuts [23], we also consider substituting the Laplacian \tilde{L} with the Normalized Laplacian $D^{-\frac{1}{2}} \tilde{L} D^{-\frac{1}{2}}$. The solution is then obtained by projecting $D^{-\frac{1}{2}} \mathbf{R}$ instead of \mathbf{R} directly (a similar modification is considered with \tilde{G}). In general, this enhances the spectral gap after the top (or bottom) three singular values, leading to a possible performance improvement. These variant have been proposed in [5, 24]. In the experiments, these variants are denoted by the suffix *Norm*.

Variant 2: Inspired by the metric upgrade in affine SfM [28], we tested the additional step of finding a matrix $K \in \mathbb{R}^{3 \times 3}$ such that the blocks in $\mathbf{R}K$ are close to orthonormal:

$$\underset{K \in \mathbb{R}^{3 \times 3}}{\text{argmin}} \|\mathbf{R}\|_{i;3 \times 3} K K^T [\mathbf{R}]_{i;3 \times 3}^T - I\|_F^2. \quad (13)$$

This variant is denoted by the suffix *Upg*.

Numerical tests: Figure 1 shows the results of the comparison of the linear methods. For both the Laplacian and the trace formulations, the normalization step leads to worse performances, and the metric upgrade does not make an appreciable difference. Between the two formulations (without modifications), the trace formulation give slightly better results. Note that these method are not robust to outliers (as expected), but they establish a good baseline.

5.2. Semidefinite and Nuclear Norm relaxations

These methods can be seen as an evolution of the ones above. Instead of recovering \mathbf{R} directly, these method aim to recover the matrix $G = \mathbf{R} \mathbf{R}^T$, which has four characteristics: it is positive-semidefinite, has rank three, each block on the diagonal is equal to I and each block off the diagonal $[G]_{ij;3 \times 3}$ corresponds to the ideal measurement $R_i R_j^T$. Then, the orthogonality constraints of $SO(3)$ are relaxed to either positive-semidefinite constraints on G ($G \succeq 0$, leading to an Semi-Definite Program, SDP, formulation) or to a low-rank prior (leading to a Nuclear Norm formulation) subject to the constraints on the diagonal (which are linear). These relaxation can be paired with different objective functions. The matrix \mathbf{R} is then obtained as in the Linear Trace formulation, where G is used instead of \tilde{G} .

5.2.1 Trace formulation

This formulation has first appeared in [5]. To derive it, note that the cost in (12) can be written as $\text{tr}(\tilde{G} \mathbf{R} \mathbf{R}^T) = \text{tr}(\tilde{G} G)$. Together with the SDP relaxation, this leads to solving

$$\max_{G \succeq 0, [G]_{ii;3 \times 3} = I} \text{tr}(\tilde{G} G). \quad (14)$$

This is a convex problem (because we dropped the rank-three constraint) and it can be efficiently solved (we use CVX [16] in our implementation).

Variant 1: We can constraint each block $[G]_{ij;3 \times 3}$ to be in the convex hull of $SO(3)$ with a minimal number

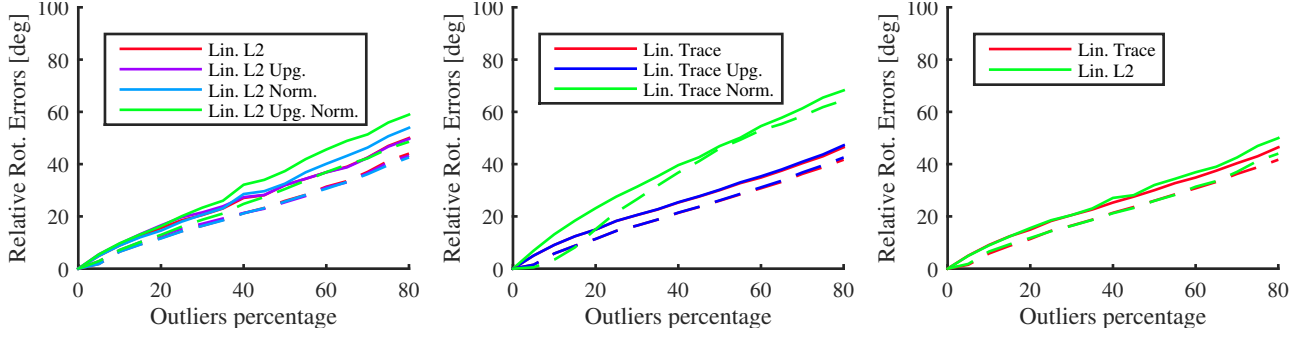


Figure 1: Comparison of linear methods

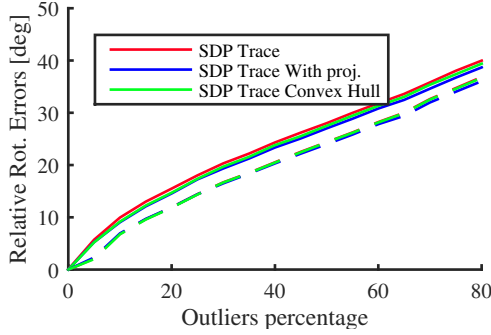


Figure 2: Comparison for the trace SDP formulation

of convex linear matrix inequalities [22]. This solution is denoted with the suffix *Convex Hull*.

Variant 2: Alternatively, we can enforce each block in G to be a rotation using (1) just before computing the SVD of G . This variant is denoted with the suffix *With proj.*

Numerical tests: Figure 2 shows the comparison of the different variants of the SDP Trace method. The use of projections and convex hull constraints slightly improves the results, but not significantly.

5.2.2 Nuclear Norm and SDP formulations

Since the matrix G should contain pairwise poses, we can rewrite (3) in matrix form as

$$\begin{aligned} \min_{G, E} \|E\| \\ \text{s.t. } P_{\Omega}(G + E) = P_{\Omega}(\tilde{G}), \quad [G]_{ii} = I, \quad i \in V, \end{aligned} \quad (15)$$

where $P_{\Omega}(X)$ is the projection to the space of matrices having the same support (in terms of 3×3 blocks) as \tilde{G} , and where the loss ℓ is based on a vector norm $\|\cdot\|$. We can use the L2 norm ($\|E\|_F^2$), thus obtaining a formulation similar to Linear L2), the L1 norm ($\|E\|_1$, promoting sparsity), or the L12 norm ($\sum_{ij} \|[E]_{ij;3 \times 3}\|_F$, promoting block sparsity).

We can pair (15) with either the SDP or Nuclear Norm relaxations. In the first case, we simply solve (15) with

the constraint $G \succeq 0$. When paired with the L12 norm, this corresponds to the formulation in [30]. In the second case, the nuclear norm regularizer leads to the following optimization problem:

$$\begin{aligned} \min_{G, E \in \mathbb{R}^{3N \times 3N}} \|G\|_* + \lambda \|E\|, \\ \text{s.t. } P_{\Omega}(G + E) = P_{\Omega}(\tilde{G}), \quad [G]_{ii} = I, \quad i \in V, \end{aligned} \quad (16)$$

where λ is a tuning parameter. When paired with the L1 norm, this formulation is identical to Principal Component Pursuit (also known as Robust PCA) [9].

5.2.3 Optimization using ADMM

We now briefly review how the SDP and Nuclear Norm formulations above can be solved using the Alternating Direction Method of Multipliers (ADMM, [7]). We first introduce the auxiliary variable Z to split the constraints on G :

$$\min_{Z, G, E} \|Z\|_* + \lambda \|E\|_1, \quad (17)$$

$$\text{s.t. } G = Z, \quad P_{\Omega}(G + E) = P_{\Omega}(\tilde{G}), \quad (18)$$

$$[G]_{ii;3 \times 3} = I_3, \quad i \in V. \quad (19)$$

The augmented Lagrangian is

$$\begin{aligned} \mathcal{L}(Z, G, E, Y_1, Y_2) = & \|Z\|_* + \lambda \|E\|_1 \\ & + \langle Y_1, G - Z \rangle + \frac{\mu_1}{2} \|G - Z\|_F^2 \\ & + \langle Y_2, P_{\Omega}(G + E - \tilde{G}) \rangle + \frac{\mu_2}{2} \|P_{\Omega}(G + E - \tilde{G})\|_F^2. \end{aligned} \quad (20)$$

The matrices Y_1 and Y_2 are dual variables and μ_1 and μ_2 are constants that control the step sizes in the optimization. Note that the equality (19) is kept since, as we will see, it is easy to handle. Then, ADMM alternately updates the primal variables Z , G and E by minimizing \mathcal{L} and the dual variables Y_1 and Y_2 by gradient ascent until convergence. Minimizing \mathcal{L} over Z can be analytically solved by singular value thresholding (in the nuclear norm formulation, [8]) or eigenvalue thresholding (with a threshold equal to zero).

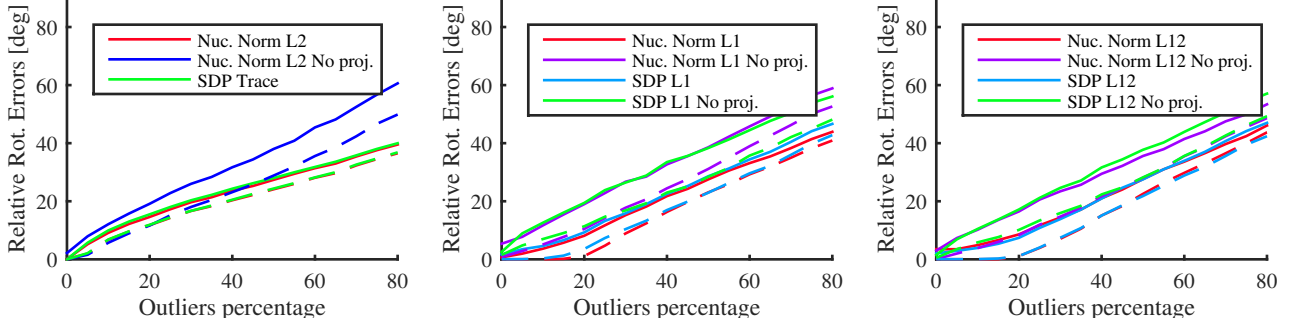


Figure 3: Comparison of nuclear norm and SDP formulations with different losses.

Minimizing \mathcal{L} over E also admits a closed-form solution by elementwise soft-thresholding [6]. Minimizing \mathcal{L} over G is the same as computing a projection to the set of matrices satisfying (19), which is straightforward.

Variant 1: We tested a modification of the basic ADMM algorithm where we constrain $[G]_{ij;3 \times 3} \in SO(3)$ for all i and j . The constraint is enforced by adding the projection (1) after the update of each block in G . Note that this makes the problem non-convex again, and we lose global optimality guarantees. We use the solution given by the original convex optimization as initialization. In the experiments, we use the suffix *No proj.* to denote the original convex formulation, and do not use any suffix for the variant with projections.

Numerical tests: Figure 3 shows the results for the low-rank nuclear norm and SDP formulations of this section using the L2, L1 and L12 losses for the fitting error. We compare the SDP Trace in lieu of the SDP L2 formulation (due to the fact that it was shown to be better in the linear methods). The nuclear norm regularizer cannot be mixed with the trace formulation, as the latter is a maximization problem while the former requires a minimization problem.

For all norms and formulations, the additional iterations using the projections on $SO(3)$ significantly improve the results. In fact, for the L1 and L12 norms, we notice the appearance of some degree of robustness as evinced by the flat region in the curve of the median between 0% and 20% outliers (i.e., in this regime, at least half of the relative poses were estimated correctly). The L12 norm is slightly better than the L1 norm, but the difference is not significant. Regarding the use of the SDP formulation versus the nuclear norm prior, both methods give very similar results. Finally, there is one caveat in the use of these methods: in some cases, even without outliers, the results do not correspond to the ground truth (i.e., the curves do not start from the origin). We investigated this issue, and we found that the problem appears in very sparse datasets (low number of edges). For instance, with the nuclear norm, it is not possible to find a value for λ for which the recovered matrix G matches the ground truth measurements, while being low rank at the same time. For the experiments shown, we used $\lambda = 0.5\sqrt{3N}$.

6. Local, iterative methods

This family of methods considers (2) as an optimization problem on manifolds, and use gradient descent methods [1] or related techniques to find local minima. These methods can also be seen as a postprocessing step applied to an initialization obtained from the global methods of Section 5.

It is common to choose the loss function ℓ to be a function of the distance on $SO(3)$, i.e.,

$$\ell(R_1, R_2) = f(d_{SO(3)}(R_1, R_2)), \quad (21)$$

where $f: \mathbb{R} \rightarrow \mathbb{R}$ can be chosen to reduce the influence of outliers (note that the Frobenius loss $\|R_1 - R_2\|_F$ is a particular case of this). In particular, taking inspiration from M-estimators in the traditional robust fitting literature [32], we can choose any monotonic function which grows slower than x^2 . In this paper, we will consider the two choices proposed in [29] and [18]. In both cases, each rotation R_i , $i \in V$ is updated as

$$R_i \leftarrow \exp \left(\sum_{j:(i,j) \in E} w_{ij} \frac{\log(R_i^T R_j)}{\|\log(R_i^T R_j)\|} \right), \quad (22)$$

where \exp and \log denote the exponential and logarithm maps in $SO(3)$ [20], and w_{ij} are weights determined by the specific formulation. Although not considered here, this formulation can be extended to account for other sources of information, such as vanishing point and priors [13].

6.1. Reshaped cost

In [29], the function f is chosen to be of the form

$$f(x) = a(1 - (1 + bx) \exp(-bx)), \quad (23)$$

where a is a normalization factor, and b controls the influence of outliers (for higher values of b , the function becomes flatter at high angles; we use $b = 5$). The function is twice differentiable almost everywhere, and we can use a gradient descent algorithm with fixed step size to minimize it, leading to the choice of weights

$$w_{ij} = \varepsilon f'(d_{SO(3)}(R_i, R_j)), \quad (24)$$

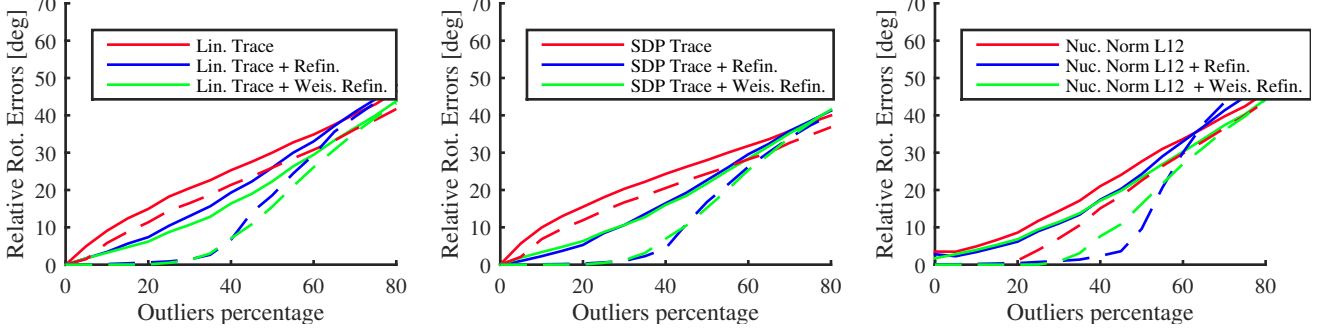


Figure 4: Comparison for iterative methods

where ε is a step size that can be computed in closed form [29] and f' is the derivative of f . All the rotations $\{R_i\}$ can be updated at the same time in parallel. Interestingly, for ideal measurements and with b high enough, this algorithm converges to the correct solution from almost any initial configuration [29]. Unfortunately, these guarantees do not hold in the presence of outliers, which is the case we are interested in this paper.

6.2. Weiszfeld algorithm

In the formulation of [18], the function f is chosen to simply be $f(x) = x$. This leads to a formulation which is the equivalent to an L1 fitting but with the use of the manifold distance. Unfortunately, the problem becomes non-differentiable (when one or more of the distances are equal to zero), and simple gradient techniques cannot be applied. However, if one fixes all the rotations except one, the problem reduces to a generalized median problem, for which the Weiszfeld algorithm has been shown to have good convergence properties [2]. The corresponding choice of weights becomes $w_{ij} = w_i$ for all $j : (i, j) \in E$ (i.e., we have only one weight) and

$$w_i = \sum_{j:(i,j) \in E} \|\log(R_i^T R_j)\|^{-1}. \quad (25)$$

Each rotation needs to be updated in sequence, so that this algorithm performs the minimization of the objective function in a coordinate-descent fashion, ensuring that the cost is reduced at every step. However, since the cost is not differentiable everywhere, there are no strong guarantees of convergence to a local minimum.

We mention here the work [11], which represents a more efficient way to optimize the same cost function.

Numerical tests: Figure 4 shows the effect of using the local iterative methods on top of the Linear Trace, SDP Trace, and Nuclear norm L12 global methods reviewed in Section 5 (the results for the other formulations are similar to the ones shown, and have been omitted due to space limitations).

The introduction of the local refinement introduces a dramatic boost in performances for all methods, including those

based on robust losses. The comparison between the functions for reshaping the distance gives inconclusive results, as the performances are quite similar and depend on the method used for initialization.

7. Outlier inference

The last family of algorithms aims to identify and remove outliers before estimating the rotations. As such, these algorithm are naturally used as a preprocessing step for the other algorithms, and rely on the following idea. Let $L = \{v_1, v_2, \dots, v_l, v_1\}$ describe a *cycle* (or *loop*) in the graph G . Ideally, the composition of the rotations along the cycle, that is, $R_L = R_{v_1 v_2} R_{v_2 v_3} \dots R_{v_l v_1}$ would be equal to the identity transformation for any cycle. We define the *loop closure error* as

$$e_L = d_{SO(3)}(R_L, I). \quad (26)$$

In the presence of small noise, the composed rotation is expected to be near the identity and e_L to be small. However, if the cycle contains one or more outliers, the loop closure error is likely to be very large. This insight was first used in [15], which, however, only aimed to find a spanning tree of inliers. A more recent and complete approach is [31], which follows a two step strategy. The first step is loop sampling. First, choose all loops of length three. Then, compute a Minimum Spanning Tree (MST) of the graph, and sample the loops obtained by adding in sequence the edges that do not belong to the tree. This is repeated for a desired number of times, while adjusting the weights for the computation of the MST to be equal to the number of times each edge appeared across the previous loops. In our implementation, we sample at least $6|E|$ loops. The second step is outlier inference. We define indicator variables $x_{ij}, x_L \in \{0, 1\}$ for each edge $(i, j) \in E$ and each sampled loop L , with the constraint

$$x_L = \max_{ij \in L} x_{ij}. \quad (27)$$

Ideally, the variables should be equal to one if the corresponding edge or the loop contains an outlier. In practice,

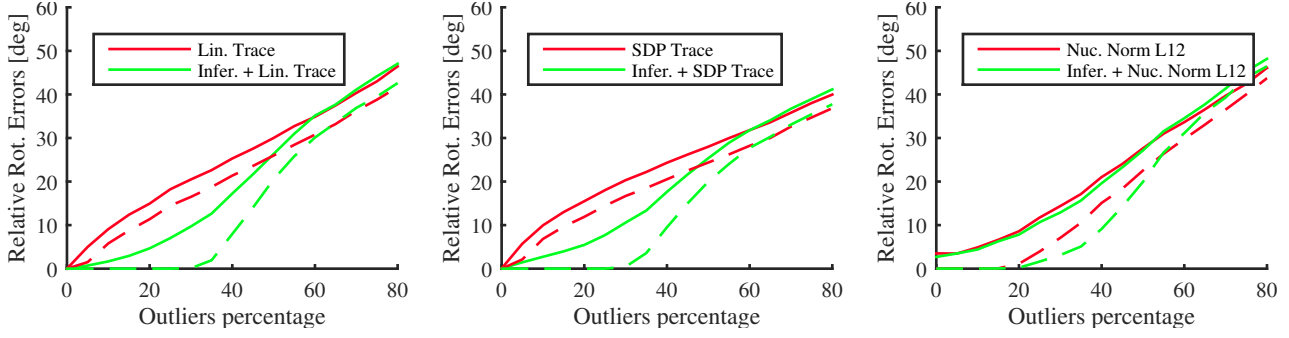


Figure 5: Comparison for methods with inference preprocessing

these are unknown, and they need to be estimated from the closure errors $\{e_L\}$ on the sampled loops by minimizing the following discrete energy:

$$E(\{x_{ij}\}, \{x_L\}) = \sum_{(i,j) \in E} \rho_{ij} x_{ij} + \sum_L \rho_L(e_L) x_L. \quad (28)$$

The parameters $\{\rho_{ij}\}$ are given by a prior on the likelihood of an edge to be an outlier or not (we set $p(x_{ij} = 0) = 0.9$). The parameters $\rho_L(e_L)$ are given by a probabilistic model of the error e_L with and without outliers in the loop. The discrete optimization problem of minimizing (28) subject to (27) can be relaxed to a linear program with linear constraint [31]. In some instances, the solution of the linear program is exact, in the sense that the variables $\{x_{ij}\}$ and $\{x_L\}$ in the solution assume discrete values. If this is not the case, one can find an exact solution by using branch and bound (i.e., fix all the variables that are discrete, fix an additional variable first to zero, then to one, and then recursively solve the reduced problem).

Note that information about loop closures is implicitly used also by the global methods when a low-rank solution is found (see also the discussion in [29]). However, these methods do not perform hard decisions on the location of the outliers, leading to lower performances. In practice, we found that this method is rather conservative, labeling the edges as inliers when the closure errors are inconclusive. We presume that this is due to the strong prior.

Numerical tests: Figure 5 shows the results of applying outlier inference before some of the global methods from Section 5 (again, some of the other methods have a similar behaviour and have been omitted). As with the local iterative methods, the addition of the inference step for removing outliers shows a dramatic boost in performance. Interestingly, this boost is more pronounced when the inference is applied before methods that are originally non-robust.

8. Combined methods

From the results above, we have seen that robust norms, local iterative refinement and outlier inference are all ways

to significantly improve the robustness to outliers in rotation optimization. The natural questions now are: what happens when all these methods are combined, and what is the combination that gives the best results in practice? We now give the answers.

Numerical tests: Figure 6 compares the results of using the outlier inference and local iterative methods. When the two are combined, the performance is better than any one of them alone. Finally, Figure 7 shows the results with the different global optimization methods. The use of pre and postprocessing effectively levels out the differences between the different global optimization methods, and a simple linear method performs surprisingly well. Moreover, the overall performance shows a behaviour very close to the ideal breakpoint of 50% outliers.

9. Conclusions and future work

We evaluated a large number of competing and complementary methods for rotation optimization in a pose graph with applications to SfM. We have shown that the best results are obtained by combining a preprocessing step to remove outliers using loop closure errors, followed by a linear factorization method to obtain a global, approximate solution and a postprocessing step to refine the estimates. The resulting method can tolerate a large number of outliers. This is not entirely surprising, as a similar procedure is commonly used in two-view SfM for fitting an essential matrix: first RANSAC is used to reject outliers, then the eight-point algorithm is used to get an initial, linear estimate, which is then refined using non-linear optimization. On the other hand, approaches based on low-rank factorization (such as Robust-PCA and SDP relaxations), which are more complex and harder to tune for very sparse datasets, have not shown significant improvements.

As future work, we will expand our tests to use larger datasets. We do not expect significantly different results, as the individual steps (outlier inference, global linear solution, local non-linear refinement) have already been shown to be effective on large datasets [2, 10, 31]. We will also evaluate

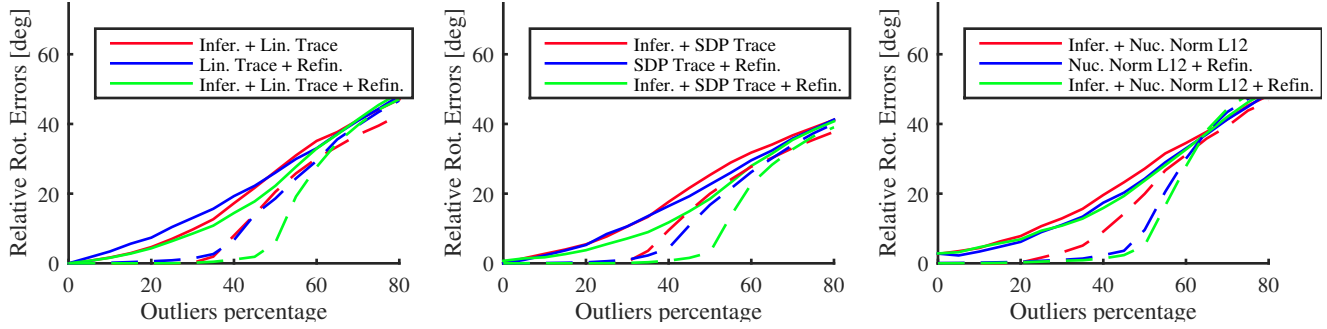


Figure 6: Comparison of methods with inference pre-processing and iterative post-processing combined

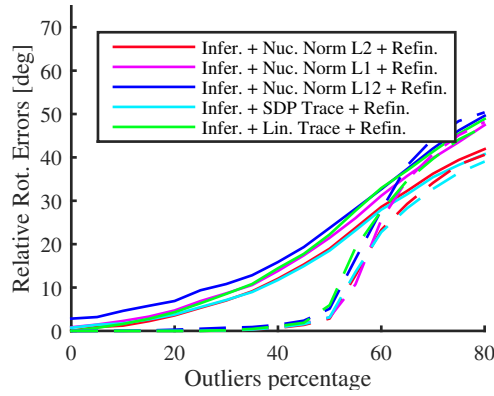


Figure 7: Comparison of all the methods that combine inference pre-processing with iterative post-processing

the effects of noise coupled with outliers (in this paper we focused exclusively on the latter due to space reasons), and the computational costs of each method. Finally, we will investigate theoretical conditions concerning the identifiability of inliers and outliers.

References

- [1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. 5
- [2] K. Aftab, R. Hartley, and J. Trumpf. Generalized weiszfeld algorithms for lq optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):728–745, 2015. 6, 7
- [3] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 1
- [4] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *IEEE European Conference on Computer Vision*, pages 29–42. Springer, 2010. 1
- [5] M. Arie-Nachimson, S. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 81–88, 2012. 3
- [6] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. *Optimization for Machine Learning*, chapter Convex optimization with sparsity-inducing norms, pages 19–53. MIT Press, 2011. 5
- [7] S. Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010. 4
- [8] J. F. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20:1956, 2010. 4
- [9] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011. 4
- [10] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert. Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization. In *IEEE International Conference on Robotics and Automation*, 2015. 1, 2, 7
- [11] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *IEEE International Conference on Computer Vision*, pages 521–528, 2013. 6
- [12] K. Cornelis, F. Verbiest, and L. V. Gool. Drift detection and removal for sequential structure from motion algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1249–1259, 2004. 1
- [13] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3001–3008, 2011. 5

- [14] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building Rome on a cloudless day. In *IEEE European Conference on Computer Vision*, pages 368–381. Springer, 2010. 1
- [15] V. M. Govindu. Robustness in motion averaging. In *Asian Conference on Computer Vision*, volume 3852, pages 457–466. Springer, 2006. 6
- [16] M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming. <http://cvxr.com/cvx/>, 2008. 3
- [17] R. Hartley and H. Li. An efficient hidden variable approach to minimal-case camera motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 1
- [18] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, 2013. 5, 6
- [19] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 1
- [20] Y. Ma. *An invitation to 3-D vision: from images to geometric models*. Springer, 2004. 5
- [21] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 3
- [22] J. Saunderson, P. A. Parrilo, and A. S. Willsky. Semidefinite relaxations for optimization problems over rotation matrices. In *IEEE International Conference on Decision and Control*, 2014. 4
- [23] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 3
- [24] A. Singer and H.-T. Wu. Vector diffusion maps and the connection laplacian. *Communications on pure and applied mathematics*, 65(8):1067–1144, 2012. 3
- [25] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *ACM Transactions on Graphics*, volume 25, pages 835–846, 2006. 1
- [26] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 2, 2008. 1
- [27] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2
- [28] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 3
- [29] R. Tron and R. Vidal. Distributed 3-D localization of camera sensor networks from 2-D image measurements. *IEEE Transactions on Automatic Control*, 2014. 5, 6, 7
- [30] L. Wang and A. Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference*, 2(2):145–193, 2013. 4
- [31] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1426–1433, 2010. 6, 7
- [32] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15:59–76, 1997. 5