Hybrid Rotation Averaging: A Globally Guaranteed Fast and Robust Rotation Averaging Approach

Yu Chen
Peking University
dreamerhacker34@gmail.com

Ji Zhao
TuSimple
zhaoji84@gmail.com

Laurent Kneip
ShanghaiTech University
lkneip@shanghaitech.edu.cn

Abstract

We address rotation averaging and its application to real-world 3D reconstruction. Local optimisation based approaches are the defacto choice, though they only guarantee a local optimum. Global optimizers ensure global optimality in low noise conditions, but they are inefficient and may easily deviate under the influence of outliers or elevated noise levels. We push the envelope of global rotation averaging by formulating it as a semi-definite program that can be solved efficiently by applying the Burer-Monteiro method. Both memory and time requirements are thereby largely reduced through a low-rank factorisation. Combined with a fast view graph filtering as preprocessing, and a local optimiser as post-processing, the proposed hybrid approach is robust to outliers. Compared against state-of-the-art globally optimal methods, our approach is $1 \sim 2$ *orders of magnitude faster while maintaining the same or better accuracy. We apply the proposed hybrid rotation averaging approach to incremental Structure from Motion (SfM) by adding the resulting global rotations as regularizers to bundle adjustment. Overall, we demonstrate high practicality of the proposed method as bad camera poses are effectively corrected and drift is reduced.*

1. Introduction

Rotation averaging is a problem that consists of estimating absolute camera orientations that agree as well as possible with a set of pairwise relative orientations. Errors expressing disagreements between estimated absolute orientations and the measured relative orientations are hereby distributed over each pairwise constraint. Rotation averaging is essential in global or hierarchical Structure from Motion (SfM) [31, 17, 38, 50, 51], as well as Simultaneous Localization and Mapping (SLAM) [10] where it can accelerate camera pose estimation and reduce drift accumulation.

In global SfM, we typically start by constructing a view graph $\mathcal{G}$ that encodes all connections between pairs of views i and j by an edge (i, j), each one including the relative motion between image i and image j. Rotation averaging then gives us the absolute orientation of each view, and it is typically followed by a translation averaging step [27, 33, 23, 52] to also obtain absolute positions. Triangulation of 3D points and joint optimisation over all parameters (i.e. bundle adjustment [40]) completes the reconstruction. In SLAM, rotation averaging has been used in the back-end pose graph optimisation [35, 10] to flexibly encounter large drift accumulations or—more generally—replace the time-consuming bundle adjustment step.

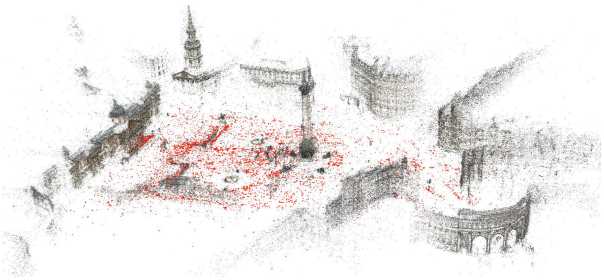


Figure 1. Reconstructions generated from the Trafalgar dataset [46], where 7085 out of 15685 images have been registered.

Rotation averaging was first proposed using the quaternion representation [24]. Later solutions can be categorised into approaches based on either local and global optimisation. Local optimisation approaches such as the one presented by Chatterjee and Govindu [11] are well studied and practical. However, these methods only return the nearest local minimum. To overcome this limitation, the community has also proposed global optimisation approaches [41, 35, 20, 19, 18]. Though the retrieval of global optima can be guaranteed, they have large computational cost and high sensitivity against outliers, and thus are impractical when applied to large-scale SfM problems.

In this paper, we focus on improving the efficiency and robustness of globally optimal rotation averaging, and push

its application to challenging scenes. Rotation averaging based on chordal distances can be reformulated as a semi-definite program (SDP) with a low-rank constraint. Inspired by the Burer-Monteiro method for solving SDP problems [9], we take advantage of this low-rank factorisation to accelerate optimisation. More specifically, the positive semi-definite variable is replaced by an appropriate factorisation, and the minimal rank variable is chosen to enhance computational speed. While the low-rank cost function is non-convex, the saddle point can be escaped, and the Riemannian Staircase framework [5] is adopted to augment the rank until the current solution is identified as a second-order critical point, thus ensuring global optimality [5].

Previous works mainly apply rotation averaging to global SfM. Though global SfM is efficient, translation averaging is often complicated by the unknown scale of relative translations and the difficulty of identifying outliers. In this work, we embed the proposed approach into an incremental SfM pipeline. The strategy is inspired by the work of Cui *et al.* [15], who propose a hybrid SfM scheme where camera rotations are estimated globally, and camera centers are estimated incrementally by a perspective-2-point (P2P) method. Though the approach is efficient, some globally estimated rotations are not correct preventing camera centers from proper registration and scene reconstructions from completeness. Inspired by Cui's approach [15], we apply rotation averaging in a traditional incremental SfM pipeline, however use the perspective-3-point (P3P) to register camera rotations and centers. We further use a novel cost function to optimize camera poses and landmarks that alleviates drift accumulation. Owing to a novel maximum-a-posterior (MAP) formulation, the camera rotations obtained from rotation averaging can notably be used as regularizers within this cost function. The resulting SfM approach (RA-SfM) shows high practicality and at least comparable accuracy, which is demonstrated at the hand of extensive experiments on large-scale real-world datasets.

In summary, the main contributions of our work are:

- We propose a new globally optimal rotation averaging scheme that exploits low-rank factorisations and processes problems faster than state-of-the-art methods by $1 \sim 2$ orders of magnitude.
- We refine the traditional incremental bundle adjustment cost function by adding the obtained global rotations as a regularizing term, which significantly alleviates drift accumulation in incremental SfM.

The practicality and superiority of our approach is demonstrated at the hand of extensive experiments on large-scale real-world datasets.

2. Related Work

Motion averaging [24, 25] is widely used in global SfM pipelines [31, 17, 38, 50, 51] as an answer to the drift problem occurring in incremental SfM [4, 47, 36, 16]. The first solution to rotation averaging goes back to Govindu [24], who uses the quaternion representation and solves the problem by linear least-squares fitting. More reliable results were later on gained by optimising over a Lie algebra [25]. In practice, the problem is complicated by the existence of outliers. To enhance the robustness of rotation averaging, absolute rotations may first be initialised under the L_1-norm, and then refined by Iteratively Reweighted Least Squares (IRLS) [11, 12]. Despite great progress, all aforementioned approaches can only guarantee a locally optimal solution. Another local approach was proposed by Crandall *et al.* [13, 14], who couple the cost function with regularisation terms to enhance robustness. However, the method is computationally demanding as it relies on discrete belief propagation over a Markov random field.

Fredriksson and Olsson [22] exploit Lagrangian duality to become the first to find a globally optimal solution to the rotation averaging problem. In a similar approach, Eriksson *et al.* [19] perform the optimisation directly on the rotation matrix by minimising chordal distances. By removing the determinant constraint on the rotation from the original SDP, they elegantly prove that there is no duality gap between the primal problem and its dual when residual errors are bounded below an angular residual threshold.

Rotation averaging can be converted into an SDP optimisation problem [8]. Wang and Singer [42] solve it by the Alternating Direction Method of Multipliers (ADMM) [7, 45]. Eriksson *et al.* [19] use a row-by-row block coordinate descent method (BCM) [44]. However, due to the slow convergence of ADMM and the repetitive fill-in procedures of BCM, neither approach proves to be practical when applied to large-scale datasets. A seminal work on the solution of SDP problems is presented by Burer and Monteiro [9], where the positive semi-definite variable is replaced by an appropriate factorisation, and the minimal rank variable is chosen to enhance computational speed. The Burer-Monteiro factorisation later inspired Boumal [5], who proposes a general optimisation technique named the Riemannian staircase algorithm, where the rank variable is augmented until the KKT condition is met, thus guaranteeing global optimality. Rosen *et al.* [35] address the SDP problem of pose graph optimisation in the Special Euclidean space (SE(n)). When translation variables are decoupled from rotations, they first find the second-order critical point by the second-order Riemannian trust-region method, and then adopt the low-rank optimisation framework of [5] to guarantee global optimality [35]. Building on SE-Sync [35], Dellaert *et al.* [18] propose *Shonan rotation averaging*, a method in which the rota-

tion matrix is vectorized, thus permitting the use of existing gradient-based optimisation methods (e.g. Gauss-Newton or Levenberg-Marquardt) on the manifold of rotation matrices. However, both SE-Sync and Shonan rotation averaging need to evaluate and maintain the Jacobian matrix, which slows down both approaches.

3. Notations and Preliminaries

Let $\mathcal{G} = \{V, E\}$ be an undirected graph, where V represents the collection of nodes and E the set of edges. Let $m = |E|$ be the number of edges and $n = |V|$ be the number of nodes. Let $\mathrm{tr}(\cdot)$ denote the trace of a square matrix. Given two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times n}$, let $\langle A, B\rangle = \sum_i \sum_j A_{ij}B_{ij}$. We therefore have $\mathrm{tr}(A^TB) = \langle A, B\rangle$. Let $\mathrm{blockdiag}(A)$ represent the block diagonal matrix of A, and $\mathrm{symblockdiag}(A) = \frac{1}{2}\,\mathrm{blockdiag}(A + A^T)$.

The set of rotations in 3D forms the Special Orthogonal Group $\mathrm{SO}(3)$, i.e.,

$$\mathrm{SO}(3) = \{R \in \mathbb{R}^{3\times 3} | R^TR = I, \det(R) = 1\}. \quad (1)$$

Since $\mathrm{SO}(3)$ is a Lie group, there exists an exponential mapping between a rotation R and its Lie algebra $\mathfrak{so}(3)$ representation $\boldsymbol{w}$ [30]:

$$R = \exp([\boldsymbol{w}]_\times). \quad (2)$$

The absolute rotations are grouped in $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$, where $R_i \in \mathrm{SO}(3)$, $i \in [n]$. Relative rotations are represented by $\mathcal{R}_{\text{rel}} = \{R_{ij}\}$, where $R_{ij} \in \mathrm{SO}(3), i, j \in [n], i < j$ is the rotation from R_i to R_j. The chordal distance between two rotations is measured by [26]

$$d_{\text{chord}}(R_1, R_2) = \|R_1 - R_2\|_F\,, \quad (3)$$

where $\|\cdot\|_F$ represents the Frobenius norm of a matrix.

4. Fast Robust Global Rotation Averaging

Globally optimal rotation averaging is sensitive to outliers, thus requiring an additional step to clean the view graph. In this section, we first present an efficient preprocessing step to filter outliers in the view graph. We then outline the application of the block coordinate descent (BCD) method to optimise the low-rank formulation of rotation averaging. Global optimality is guaranteed theoretically. Finally, we apply a local optimisation step to further refine the result in the case of scenes that have many erroneous edges.

4.1. Fast View Graph Filtering

The view graph plays an important role in our SfM pipeline. We clean the view graph for two main reasons: (1) Solutions of global rotation averaging algorithms can be biased by outliers. In addition, global optimality is only guaranteed when the residuals for each edge are bounded below a certain threshold [19]. (2) Some view pairs are redundant and even have negative impact on the quality of SfM results. Zach *et al.* [48] proposed a view graph filtering (VGF) technique to obtain a high quality initial view graph, where loop constraints of rotation triplets are utilised to detect outliers. Specifically, edge (i, j) is an outlier if its angular error satisfies

$$d(R_{ij}R_{jk}R_{ki}, I) > \epsilon. \quad (4)$$

Despite its effectiveness, [48] needs to validate all triplets, which is impractical for large-scale datasets. However, [37] suggests that it is not necessary to check all triplets to distinguish inliers from outliers, and that an increased number of valid 2D-2D image correspondences usually suggests more reliable two-view geometries. We propose an efficient view graph filtering method that relies on this observation. In the following, we denote a group of 3 nodes as a *triplet*, and a triplet with two valid edges and one unverified edge as a *weak triplet*.

Given an initial view graph $\mathcal{G}$, we start by constructing a maximum spanning tree (MST), where the weight of an edge is the number of valid 2D-2D correspondences. The relative rotations from this MST are all treated as valid. We then check the triplets along with the MST. That is, all adjacent edges that share a common node in the MST are used to build triplets. Next, we generate many weak triplets. Now supposing that edges (i, j) and (j, k) are valid and edge (i, k) exists, we use criterion (4) to verify the validity of edge (i, k). An iteration is completed once all such weak triplets have been verified. After the first iteration, new weak triplets are generated based on which we can perform another iteration. We empirically found that 3 iterations are sufficient for successful rotation averaging. Due to space limitation, further explanations of our VGF approach are provided in appendix.

4.2. Global Rotation Averaging using the Burer-Monteiro Method

4.2.1 Global Rotation Averaging Formulation

Rotation averaging can be formulated as

$$\min_{R_1,\cdots,R_n} \sum_{(i,j)\in E} d^p(R_{ij}, R_jR_i^T), \quad (5)$$

where $d^p(\cdot)$ represents a distance measure under a p-norm. When Frobenius norm is chosen as the distance measure, problem (31) can be rewritten as

$$\min_R \; -\mathrm{tr}(R^TGR) \quad \text{s.t.} \quad R \in \mathrm{SO}(3)^n, \quad (6)$$

where $R = [R_1\ R_2\ \cdots\ R_n], G_{ij} = a_{ij}R_{ij}$, $a_{ij} = 1$ if the edge between views i and j exists, and $a_{ij} = 0$ otherwise.

The constraint denotes that the rotations are optimised on $SO(3)$ manifold. Eriksson *et al.* [19] solve the dual problem of (36) with determinant constraint relaxation

$$\begin{aligned} \min_{X} \quad & -\operatorname{tr}(GX) \\ \text{s.t.} \quad & X_{ii} = I_3,\ i = 1, \cdots, n, \quad X \succeq 0, \end{aligned} \tag{7}$$

where X is partitioned as

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nn} \end{bmatrix}. \tag{8}$$

They furthermore prove that there is no duality gap between the primal problem (36) and its dual problem (37) if the maximum residuals stay below a certain threshold. We kindly refer the reader to appendix and [19] for more details.

4.2.2 Block Coordinate Descent with Low-Rank Factorization

Problem (37) is an SDP problem [8]. Inspired by the Burer-Monteiro method [9, 6] and low-rank factorisation in SDP problems [43, 39], we solve the rotation averaging problem by optimisation on a manifold. Since every $X \succeq 0$ can be factored as $Y^T Y$ for some Y [9], and—in the case of rotation averaging—the optimal value of X satisfies $X^\star = R^{\star T} R^\star$ [19], we suggest an implicit constraint on X such that $\operatorname{rank}(X) = 3$. Thus, we reformulate X as

$$X = Y^T Y, \tag{9}$$

where $Y = [Y_1\ Y_2\ \cdots\ Y_n], Y_i^T Y_i = I, \forall i \in [n]$. Substituting (9) into problem (37), we obtain the problem

$$\begin{aligned} \min_{Y} \quad & -\operatorname{tr}(GY^T Y) \\ \text{s.t.} \quad & Y = [Y_1\ Y_2\ \cdots\ Y_n], \quad Y_i^T Y_i = I, \forall i \in [n]. \end{aligned} \tag{10}$$

Next, we derive the block coordinate minimisation (BCM) algorithm for problem (40). Note that

$$\begin{aligned} &\operatorname{tr}(GY^T Y) = \operatorname{tr}(YGY^T) = \langle YG, Y \rangle \\ &= \sum_{j=1}^{n} \langle \sum_{i=1}^{n} Y_i G_{ij}, Y_j \rangle = \sum_{j=1}^{n} \langle \hat{Q}_j, Y_j \rangle, \end{aligned} \tag{11}$$

where $\hat{Q}_j = \sum_{i=1}^{n} Y_i G_{ij}$. Let $f(Y^k) = \sum_{j=1}^{n} \langle \hat{Q}_j^k, Y_j^k \rangle$, where superscript k represents the k-th iteration in BCM. Since $G_{ii} = \mathbf{0}$, using (11) we have

$$\begin{aligned} &\arg\min_{Y} f(Y^k) = \arg\min_{Y} \sum_{j=1}^{n} \langle \hat{Q}_j^k, Y_j^k \rangle \\ &= \arg\min_{Y} \sum_{j=1}^{n} \langle \sum_{i \neq j}^{n} Y_i^k G_{ij}, Y_j^k \rangle = \arg\min_{Y} \sum_{j=1}^{n} \langle Q_j^k, Y_j^k \rangle, \end{aligned}$$

where $Q_j = \sum_{i \neq j}^{n} Y_i G_{ij}$. This leads us to the derivation

$$\begin{aligned} &Y_{j_k}^{k+1} = \arg\min_{Y_{j_k}} \ f(Y_1^k, \cdots, Y_{j_k-1}^k, Y_{j_k}^k, Y_{j_k+1}^k, \cdots, Y_n^k) \\ &= \arg\min_{Y_{j_k}} \ \sum_{j=1}^{n} \langle \sum_{i \neq j}^{n} Y_i^k G_{ij}, Y_j^k \rangle \\ &= \arg\min_{Y_{j_k}} \ \langle Q_j^k, Y_{j_k}^k \rangle + \sum_{j \neq j_k}^{n} \sum_{i \neq j}^{n} \langle Y_i^k G_{ij}, Y_j^k \rangle \\ &= \arg\min_{Y_{j_k}} \ 2\langle Q_j^k, Y_{j_k}^k \rangle + \sum_{j \neq j_k}^{n} \sum_{i \neq j, j_k}^{n} \langle Y_i^k G_{ij}, Y_j^k \rangle \\ &= \arg\min_{Y_{j_k}} \ 2\langle Q_j^k, Y_{j_k}^k \rangle = \arg\min_{Y_{j_k}} \ \frac{1}{2} \left\| Y_{j_k}^k + Q_j^k \right\|_F^2. \end{aligned} \tag{12}$$

Let $U_j \Sigma V_j$ be the singular value decomposition of $-Q_j$. The closed-form solution [29] to problem (12) is

$$Y_j^\star = U_j I_{3\times 3} V_j^T = U_j V_j^T. \tag{13}$$

Once the optimal value Y_j^* is obtained, we need to update Q_j at each inner iteration. The update rule is

$$\begin{aligned} &Q_j^{k+1} = \sum_{i \neq j}^{n} Y_i^{k+1} G_{ij} = Y_{j_k}^{k+1} G_{j_k j} + \sum_{i \neq j, j_k} Y_i^{k+1} G_{ij} \\ &= Y_{j_k}^{k+1} G_{j_k j} + \sum_{i \neq j, j_k} Y_i^k G_{ij} + Y_{j_k}^k G_{j_k j} - Y_{j_k}^k G_{j_k j} \\ &= Y_{j_k}^{k+1} G_{j_k j} + \sum_{i \neq j} Y_i^k G_{ij} - Y_{j_k}^k G_{j_k j} \\ &= Q_j^k + (Y_{j_k}^{k+1} - Y_{j_k}^k) G_{j_k j}. \end{aligned} \tag{14}$$

Our low-rank BCM algorithm is outlined in Algorithm 1. The advantage of our method over [19] is three-fold. First, orthogonality is retained due to the orthogonality of U_j and V_j in (13). Second, as the dimension of our optimised variable Y is $3 \times 3n$, the time complexity in steps 2∼8 is $O(n)$ (this contrasts with the $O(n^2)$ time complexity of the row-by-row approach in [19]). Third, in steps 6∼7 of Algorithm 1, the time complexity of $O(n)$ is only an upper bound occurring for general cases. In practice, due to the commonly sparse structure of SfM problems, time complexity can be further reduced to $O(d)$, where d is the degree of the nodes. This property is important for accelerating the optimisation. We also remark that the proposed low-rank BCM approach for rotation averaging can obtain a global sublinear convergence rate, and the proof is given in appendix.

Algorithm 1 Local low-rank SDP Search Algorithm

Input: relative rotations $\mathcal{R}_{\text{rel}}$, $maxIterNum$, Y^0.
Output: First-order critical point $Y^\star$

$k \leftarrow 0$; $Q_j^0 \leftarrow \sum_{i \neq j}^{n} Y_i G_{ij}, \forall j \in [n]$.
while $k < maxIterNum$ & not converge **do**
  for $i < n$ **do**
    $j_k \leftarrow i$
    Update $Y_{j_k}^{k+1}$ by Eq. (13)
    for $\forall j \neq j_k, G_{j_k j} \neq 0$ **do**
      $Q_j^{k+1} \leftarrow Q_j^k + (Y_{j_k}^{k+1} - Y_{j_k}^k) G_{j_k j}$
  $k \leftarrow k + 1$;
return Y

Algorithm 2 Global Rotation Averaging Algorithm

Input: relative rotations $\mathcal{R}_{\text{rel}}$, r_0
Output: global rotations $\mathcal{R} = \{R_1, R_2, \cdots, R_n\}$

for $r = r_0, \cdots, dn + 1$ **do**
  Compute $Y^\star$ by Algorithm 1.
  if $Y^\star$ passes verification **then**
    break
  else
    Augment Y by Eq. (16)
    $Y \leftarrow EscapeDirection(f, Y)$
Rounding solution by Eq. (17).

4.2.3 Globally Optimal Guarantee

In Sec.4.2.2, we have derived the first-order critical point of (40). However, problem (40) is non-convex, and there is no guarantee that we can obtain the global optimum. Boumal [5] proposes a general framework named the *Riemannian Staircase* algorithm (RS), which can find the global optimum. Given an initial first-order solution, RS first verifies if the solution $X^\star$ is truly the global optimum. We define the explicit matrix

$$S = \nabla f(X) - \text{symblockdiag}(\nabla f(X) X). \tag{15}$$

By *Corollary 3.6* in [5], $X^\star = {Y^*}^T Y^*$ is the global optimum if and only if $S \succeq 0$, and $Y^\star$ is the global minimiser of the non-convex problem (40). Specifically, we only need to verify that the minimum eigenvalue of S is non-negative. We use the general sparse solver of Spectra [2] to efficiently compute the minimum eigenvalue. If $X^\star$ does not pass the verification, we first augment Y as in

$$Y = \begin{bmatrix} Y \\ \mathbf{0}_{1 \times nd} \end{bmatrix}, \tag{16}$$

then leverage the negative eigenvector of S as the escape direction and combine it with a line search to escape the saddle point. After getting the low-rank SDP solution, the final absolute rotations are retrieved by the *Rounding Solution* [35], as in

$$R = [{Y_1}^{\star T} Y^\star]_{1:3,1:3n}, \tag{17}$$

where Y_1^* is the first block in Y^*. The complete procedure for obtaining the absolute rotations is outlined in Algorithm 2.

4.2.4 Local Refinement

The global optimisation presented in Sec. 4.2 assumes that the input relative rotations do not contain any outliers. As a result, it is sensitive to outliers. To further improve the robustness and accuracy of the proposed low-rank BCM method, we follow the *suggest-and-improve* framework of [34]. A local optimizer with a robust loss function is used to further refine the result. Built upon the work of Chatterjee and Govindu [11], rotations R_{ij}, R_i, R_j can be represented by the corresponding Lie algebra elements $\boldsymbol{\omega}_{ij}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_j$, respectively. Using the Baker-Campbell-Hausdorff (BCH) equation [1], we have $\boldsymbol{\omega}_{ij} = \boldsymbol{\omega}_j - \boldsymbol{\omega}_i$ [11]. The collection of all relative constraints can hence be summarised in the linear relationship $A\boldsymbol{\omega}_{\text{global}} = \boldsymbol{\omega}_{\text{rel}}$. A here is a sparse matrix with 3×3 blocks of zeros in each row, except for an entry I and an entry $-I$ at block-indices j and i, respectively. The robust least-squares cost function is depicted by

$$\arg\min_{\boldsymbol{w}_{\text{global}}} \sum \rho(\|A\boldsymbol{\omega}_{\text{global}} - \boldsymbol{\omega}_{\text{rel}}\|), \tag{18}$$

where $\rho(\cdot)$ is the Huber loss.

An ablation study of robustness against outliers is shown in Fig. 2. The outlier ratio ranges from 0 to 50% and is

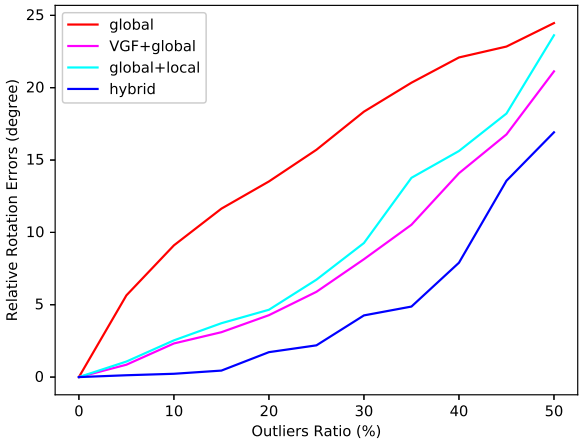


Figure 2. Ablation study of robustness. Outliers are generated by perturbing ground truth by rotation between $60^\circ - 90^\circ$. *global* represents the proposed low-rank BCM method, and *hybrid* represents the proposed hybrid method with VGF and local refinement.

incremented in steps of 5%. We display the mean rotation error over 30 experiments. As can be observed, both VGF and local refinement improve the robustness of the global rotation averaging approach.

5. Application to Structure-from-Motion

In this section, we apply our rotation averaging method to an incremental SfM pipeline, which is known to suffer from the drift problem. Our hybrid SfM pipeline can be summarised as follows: We first construct the view-graph and obtain global rotations from our proposed fast and robust rotation averaging approach. Next, we create a seed reconstruction by selecting two appropriate images. We then continue by incrementally registering adjacent camera poses using a RANSAC-based [21] P3P [28] algorithm, and triangulate landmarks. To reduce the accumulation of errors in our incremental SfM pipeline, we perform local bundle adjustment after each successful registration of an image, and global bundle adjustment whenever the number of recently added views surpasses a certain threshold.

The drift problem is not solved, as each newly computed camera pose is affected by a small error, and these errors accumulate along the graph. Traditional incremental SfM pipelines have no way to rectify these errors. To tackle this problem, we introduce a novel cost function for bundle adjustment which relies on MAP. Let $\mathcal{I}_i$ denote the measurements of image i. Each image observes 3D landmarks given by the set $\mathcal{P}_i$. Let $\mathbf{u}_{il} \in \mathcal{I}_i$ furthermore denote the image keypoint measurement of landmark l in frame i. Grouping image measurements with the pre-computed known rotations, we have

$$\mathcal{Z} = \left\{ \mathcal{I}_i, \hat{\mathcal{R}}_i, \hat{\mathcal{R}}_j | (i,j) \in \mathcal{E} \right\} \tag{19}$$

In SfM, camera poses are estimated as well as 3D points observed by cameras. We have the following unknowns:

$$\mathcal{X}_i = \{R_i, C_i, \mathcal{P}_i\},\ \mathcal{X} = \{\mathcal{X}_i | i \in [0, N]\}, \tag{20}$$

where R_i and C_i are the estimated camera rotation and center, respectively. Note that the sets $\mathcal{P}_i$ are intersection, which—in a slight abuse of notation—is ignored for the sake of simplicity.

5.1. Maximum a Posteriori Estimation

Given the available visual measurements, global rotations $\mathcal{Z}$, and priors $p(\mathcal{X}_0)$, the posterior probability of the variables $\mathcal{X}_i$ is:

$$\begin{aligned} p(\mathcal{X}|\mathcal{Z}) &= p(\mathcal{X}_0) \cdot p(\mathcal{Z}|\mathcal{X}) \\ &= p(\mathcal{X}_0) \cdot \prod_{(i,j)\in\mathcal{E}} p(\mathcal{I}_i, \hat{\mathcal{R}}_i, \hat{\mathcal{R}}_j | \mathcal{X}) \\ &= p(\mathcal{X}_0) \cdot \prod_{i \in N} \prod_{l \in \mathcal{I}_i} p(\mathbf{u}_{il}|\mathcal{X}_i) \cdot \prod_{(i,j)\in\mathcal{E}} p(\hat{\mathcal{R}}_i, \hat{\mathcal{R}}_j | \mathcal{X}_i, \mathcal{X}_j). \end{aligned} \tag{21}$$

The MAP estimate $\mathcal{X}_i^\star$ corresponds to the maximum of (21), or equivalently, the minimum of the negative log-posterior. Under the assumption of zero-mean Gaussian noise, the negative log-posterior can be written as a sum of squared residual errors:

$$\begin{aligned} \mathcal{X}^\star &= \arg\min_{\mathcal{X}} -\log p(\mathcal{X}|\mathcal{Z}) \\ &= \arg\min_{\mathcal{X}} \|\mathbf{r}_0\|^2 \\ &+ \sum_{i \in N} \sum_{l \in \mathcal{I}_i} \rho_v(\|\mathbf{r}_{\mathcal{I}_{il}}\|) + \sum_{(i,j)\in\mathcal{E}} w_{ij} \left\|\mathbf{r}_{\mathcal{R}_{ij}}\right\|^2, \end{aligned} \tag{22}$$

where $\rho_v(\cdot)$ is a robust loss function, and w_{ij} is an individual weighting function for each known rotation term. In this paper, we fix w_{ij} as a constant. This objective divides into two terms which are explained as follows.

Visual Term: We adopt traditional re-projection error in bundle adjustment as our visual term

$$\mathbf{r}_{\mathcal{I}_{il}} = \mathbf{u}_{il} - \mathbf{\Pi}(R_i, C_i, \mathcal{P}_i, l), \tag{23}$$

where $\mathbf{\Pi}(\cdot)$ is the back-projection function that projects landmarks into the image plane. Note that the latter also depends on camera intrincs, which—for the sake of a simplified and general notation—are not specified.

Known Rotation Term: The added known rotation term is

$$\mathbf{r}_{\mathcal{R}_{ij}} = \log(\hat{\mathcal{R}}_j^T \hat{\mathcal{R}}_i R_i^T R_j), \tag{24}$$

where $\log$ is the logarithm map $\mathrm{SO}(3) \rightarrow \mathfrak{so}(3)$. This known rotation term is used as a regularizer in the complete cost function.

6. Experimental Results

Our experiments aim at demonstrating the accuracy, efficiency, and robustness of the proposed methods. We implement Levenberg-Marquardt (LM) [32], row-by-row block coordinate descent (RBR-BCD) [19], and our low-rank factorisation approach in C++. Besides, the implementation of SE-Sync [35] and Shonan [18] are obtained from the given websites of their papers. For HSfM [15] and LUD [33], we use [12] as the rotation averaging solver, and the Ceres solver [3] for bundle adjustment. All approaches are tested on a laptop with a 2.7 GHZ CPU and 8GB RAM.

6.1. Evaluation on Synthetic Datasets

We designed 7 synthetic datasets to evaluate the performance of our rotation averaging approach, The view and relative rotation numbers are shown in Table 1, and denoted by n and #edges respectively. The ground truth absolute rotations are initialised randomly. The relative rotations are constructed by a spanning tree expanded by random edges until the given number of relative poses is reached. All relative rotations are derived from ground truth, and perturbed

Table 1. Comparison of runtime on synthetic datasets. n is the number of rotations, $\bar{R}$ represents the average rotation error (unit: degree).

n	#edges	σ	**LM** [32]		**RBR-BCD** [19]		**Shonan** [18]		**SE-Sync** [35]		**Ours**	
			$\bar{R}$	$time(s)$	$\bar{R}$	$time(s)$	$\bar{R}$	$time(s)$	$\bar{R}$	$time(s)$	$\bar{R}$	$time(s)$
20	30	0.2	1.800e-03	0.001	1.813e-06	0.007	1.719e-06	0.089	**1.499e-06**	**< 1e-06**	1.502e-06	**< 1e-06**
		0.5	1.799e-01	0.002	2.634e-02	0.013	**2.633e-02**	0.112	2.634e-02	**< 1e-06**	**2.633e-02**	0.001
100	300	0.2	5.328e-03	0.007	2.607e-06	0.526	2.596e-06	0.787	2.589e-06	**0.003**	**2.588e-06**	**0.003**
		0.5	5.315e-01	0.014	3.788e-02	0.530	**3.591e-02**	0.870	3.788e-02	**0.003**	3.788e-02	0.011
500	1000	0.2	1.483e-03	0.264	3.008e-06	176.316	2.995e-06	18.47	**2.901e-06**	0.079	2.994e-06	**0.006**
		0.5	1.225e-01	0.330	**4.369e-02**	186.311	4.401e-02	26.60	**4.369e-02**	0.079	**4.369e-02**	**0.051**
1000	4000	0.2	1.800e-01	0.321	2.810e-06	2,362	2.731e-06	22.730	2.641e-06	0.372	**2.490e-06**	**0.008**
		0.5	2.210e-01	0.237	**3.790e-02**	2,579	3.988e-02	29.57	3.930e-02	0.343	4.100e-02	**0.105**
5000	20000	0.2	1.800e-01	7.083	-	-	2.335e-06	159.638	**2.190e-06**	4.63	2.246e-06	**0.052**
		0.5	3.980e-01	7.375	-	-	3.977e-02	340.116	**3.800e-02**	3.95	**3.800e-02**	**0.441**
10000	40000	0.2	1.320e-01	24.256	-	-	2.375e-06	1108.435	**2.012e-06**	13.398	**2.012e-06**	**0.092**
		0.5	2.340e-01	24.071	-	-	NC	NC	3.612e-02	15.205	**3.600e-02**	**1.042**
50000	200000	0.2	-	-	-	-	-	-	-	-	**2.941e-06**	**0.602**
		0.5	-	-	-	-	-	-	-	-	**5.000e-02**	**6.193**

by random angular rotations about randomly selected axes. The perturbation angles are normally distributed with 0 mean and a variance of either $\sigma = 0.2$ rad or $\sigma = 0.5$ rad. Initial absolute rotations are chosen randomly.

The evaluation results are shown in Table 1, where we compare our method against LM [32], RBR-BCD [19], Shonan [18] and SE-Sync [35]. In terms of efficiency, RBR-BCD is the slowest and almost 1000 times slower than others when $n = 1000$. SE-Sync is faster than LM but stays within the same order of magnitude. While SE-Sync is slightly faster than ours when cameras number is below 500, our approach is $1 \sim 2$ orders of magnitude faster than SE-Sync when the number of views grows beyond 1000. While Shonan is also a low-rank method, as well as SE-Sync and ours, it is almost 2 magnitude slower than SE-Sync, and 3 magnitude slower than ours. In terms of the scale of the solved problems, RBR-BCD failed when the camera number increased to $5000, 10000$, or 50000. This is primarily due to insufficient memory for optimisation, and we marked the corresponding cells in the table by "–". LM, Shonan and SE-Sync failed when the camera number reaches 50000, as there is insufficient memory to hold the Jacobian matrix. Besides, Shonan is not converged when camera number is 10000 with variance 0.5, and we marked the corresponding cells by "NC". As our approach only needs to compute the SVD of a small block matrix and evaluate d matrix operations of 3×3 matrices in each iteration, we are able to solve all the evaluated large-scale datasets.

In terms of accuracy, LM achieves the global optimum with certain probability ($30\% - 70\%$ as reported in [19]), and Table 1 only shows the best results. While all the evaluated globally optimal methods have the same accuracy and can both obtain the global optimum for successful cases.

Figure 3. Reconstruction results for the Campus dataset [17]. Left: COLMAP [36], Right: Our RA-SfM.

6.2. Evaluation on Real-World Datasets

We evaluate the performance of our RA-SfM on large scale real-world datasets and compare it against state-of-the-art incremental [36], global [33] and hybrid [15] SfM approaches. Since the quasi-convex SfM approach [49] is sensitive to outlies and extremely slow in such datasets, we did not evaluate it in our experiment. (We also evaluate our RA algorithm in global SfM pipeline individually, and the results are given in appendix.)

Figure 3 shows the reconstruction results of COLMAP and our RA-SfM on the Campus [17] dataset. This dataset, which has a loop, mainly contains plants that can produce lots of wrong matching results. COLMAP [36] fails to reconstruct this dataset, as the camera poses drift and the loop is not closed. Our approach closes the loop successfully, as the known rotation optimisation is able to further constrain camera poses after the initial registration.

We also evaluated our approach on the online datasets from [46], which are collections of challenging unordered images. The datasets contains many wrong epipolar geometries due to extreme viewpoint, scale and illumination changes. The runtime and accuracy results are shown in Table 3. As can be observed, our approach recovers the most camera poses in all online datasets, and has the lowest mean reprojection error (MRE) in most of the online datasets. For our RA-SfM, the time for rotation averaging is separately given. COLMAP [36] is the second best approach in terms

Table 2. Comparison of runtime and accuracy on online datasets [46]. N_i denotes the number of images, N_c denotes the number of registered cameras, N_p denotes the reconstructed 3D landmarks, MRE means mean reprojection error in pixel.

Dataset	N_i	**COLMAP** [36]				**LUD** [33]				**HSfM** [15]				**RA-SfM**				
		N_c	N_p	MRE (px)	$T(s)$	N_c	N_p	MRE (px)	$T(s)$	N_c	N_p	MRE (px)	$T(s)$	N_c	N_p	MRE (px)	$R_t(s)$	$T(s)$
Alamo	2,915	666	94K	0.68	882	578	146K	1.28	260	522	149K	1.62	1,079	895	141K	0.65	3.84	2,784
Ellis Island	2,587	315	64K	0.70	332	234	16K	1.54	24	208	34K	2.53	169	727	146K	0.72	6.4	3,954
Gendarmenmarkt	1,463	861	123K	0.68	627	705	87K	1.51	104	542	74K	1.94	377	1,023	202K	0.70	6.2	3,903
Madrid Metropolis	1,344	368	43K	0.60	251	350	51K	1.08	36	292	51K	1.48	221	438	66K	0.59	2.2	1,440
Montreal N.D.	2,298	506	98K	0.81	723	462	166K	1.64	194	418	155K	1.95	1041	528	105K	0.67	1.7	1,446
Notre Dame	1,431	1408	349K	0.76	22,788	550	262K	2.06	259	526	281K	2.30	2,375	1,409	353K	0.75	1.7	3,323
NYC Library	2,550	453	77K	0.69	420	336	70K	1.52	75	282	74K	1.99	356	519	100K	0.65	2.1	1,734
Piazza del Popolo	2,251	437	47K	0.72	380	329	38K	1.65	62	286	35K	1.92	212	966	122K	0.66	4.3	3,228
Piccadilly	7,351	2,336	260K	0.75	1,961	2,301	202K	1.83	262	1,665	185K	2.09	2,169	3,041	363K	0.80	23.8	15,324
Roman Forum	2,364	1,409	222K	0.70	1,041	1,045	256K	1.71	182	1,071	262K	1.93	2,237	1,460	267K	0.77	4.6	5,502
Tower of London	1,576	578	109K	0.61	678	485	140K	1.65	95	398	149K	1.91	816	672	139K	0.58	2.2	2,040
Trafalgar	15,685	5,211	450K	0.74	5,122	5,044	378K	1.56	713	3,446	318K	1.95	5,761	7,085	597K	0.72	58.5	15,048
Union Square	5,961	763	53K	0.67	693	803	41K	1.65	107	769	38K	1.88	1,763	809	57K	0.52	4.5	1,962
Vienna Cathedral	6,288	933	190K	0.74	1,244	849	203K	1.91	173	662	252K	2.36	2,307	1,173	303K	0.71	9.4	1,611
Yorkminster	3,368	456	105K	0.70	997	421	132K	1.75	135	417	129K	1.93	1,487	614	183K	0.64	5.8	926

Figure 4. Visual reconstruction results for some of the online datasets [46]. For each sub-figure, the top and bottom images are respectively the results obtained from COLMAP [36] and our RA-SfM. (The first two columns are results of two parts of the Ellis Island datasets.)

of recovered camera poses and MRE, which shows the robustness of incremental SfM approaches. While LUD [33] is the most efficient among the evaluated approaches, it has large MRE, and the recovered camera poses is less than ours or even COLMAP. HSfM [15] is faster than COLMAP and our approach, because it only samples 2 correspondences to compute the camera center in each RANSAC iteration. HSfM [15] recovers the fewest camera poses, and fails to recover the correct camera centers.

Some visual results for online datasets are shown in Fig. 4. For each sub-figure, the top and bottom images are the results obtained by COLMAP [36] and our RA-SfM, respectively. For Ellis Island dataset, we showed two different parts in the first two columns of Fig. 4, where the red rectangle area shows the comparison result. For Gendarmenmarkt dataset, the reconstruction result of COLMAP is bad on the left part, which indicates wrong camera poses. For Vienna Cathedral dataset, our approach can recover more scene details than COLMAP, as is again indicated by the red rectangle.

7. Conclusion

This paper presents a novel globally optimal solution to the rotation averaging problem. We extend the BCM method by including the Burer-Monteiro low-rank factorisation, which significantly reduces the computational complexity and the memory footprint. Compared to previous state-of-the-art globally optimal methods, our proposed method is $1 \sim 2$ orders of magnitude faster for large-scale datasets while maintaining the same or better accuracy. A

hybrid combination with view graph filtering and local refinement finally increases robustness against outliers. The exposition is rounded off by a soft embedding into an incremental SfM pipeline leading to accurate, reliable, and highly efficient results.

References

[1] Baker-campbell-hausdorff formula. https://en.wikipedia.org/wiki/Baker-Campbell-Hausdorff_formula. 5

[2] Spectra. https://spectralib.org. 5

[3] Sameer Agarwal and Keir Mierle. Ceres solver. http://ceres-solver.org. 6

[4] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *IEEE 12th International Conference on Computer Vision*, pages 72–79, 2009. 2

[5] Nicolas Boumal. A riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints. *CoRR*, abs/1506.00575, 2015. 2, 5, 11, 12

[6] Nicolas Boumal, Vladislav Voroninski, and Afonso S. Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016. 4

[7] Stephen P. Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. 2

[8] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014. 2, 4

[9] Samuel Burer and Renato D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95(2):329–357, 2003. 2, 4

[10] Álvaro Parra Bustos, Tat-Jun Chin, Anders P. Eriksson, and Ian D. Reid. Visual SLAM: why bundle adjust? In *International Conference on Robotics and Automation*, 2019. 1

[11] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *IEEE International Conference on Computer Vision*, pages 521–528, 2013. 1, 2, 5, 11

[12] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):958–972, 2018. 2, 6, 11, 14, 15

[13] David J. Crandall, Andrew Owens, Noah Snavely, and Dan Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3001–3008, 2011. 2

[14] David J. Crandall, Andrew Owens, Noah Snavely, and Daniel P. Huttenlocher. SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2841–2853, 2013. 2

[15] Hainan Cui, Xiang Gao, Shuhan Shen, and Zhanyi Hu. Hsfm: Hybrid structure-from-motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2393–2402, 2017. 2, 6, 7, 8

[16] Hainan Cui, Shuhan Shen, Xiang Gao, and Zhanyi Hu. Batched incremental structure-from-motion. In *International Conference on 3D Vision*, pages 205–214, 2017. 2

[17] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *IEEE International Conference on Computer Vision*, pages 864–872, 2015. 1, 2, 7

[18] Frank Dellaert, David M. Rosen, Jing Wu, Robert Mahony, and Luca Carlone. Shonan rotation averaging: Global optimality by surfing $SO(p)^n$. In *European Conference on Computer Vision*, volume 12351, pages 292–308. Springer, 2020. 1, 2, 6, 7

[19] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging with the chordal distance: Global minimizers and strong duality. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 1, 2, 3, 4, 6, 7, 11, 14, 15

[20] Anders P. Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging and strong duality. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–135, 2018. 1

[21] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 6

[22] Johan Fredriksson and Carl Olsson. Simultaneous multiple rotation averaging using Lagrangian duality. In *Asian Conference on Computer Vision*, volume 7726, pages 245–258, 2012. 2

[23] Thomas Goldstein, Paul Hand, Choongbum Lee, Vladislav Voroninski, and Stefano Soatto. Shapefit and shapekick for robust, scalable structure from motion. In *Computer Vision - ECCV 2016 - 14th European Conference*, pages 289–304, 2016. 1

[24] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 218–225, 2001. 1, 2, 11

[25] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 684–691, 2004. 2, 11

[26] Richard I. Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, 2013. 3

[27] Nianjuan Jiang, Zhaopeng Cui, and Ping Tan. A global linear method for camera pose registration. In *IEEE International Conference on Computer Vision*, pages 481–488, 2013. 1

[28] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition*, pages 2969–2976, 2011. 6

[29] Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *J. Sci. Comput.*, 58(2):431–449, 2014. 4

[30] Yi Ma, Stefano Soatto, Jana Košecká, and S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Science & Business Media, 2012. 3

[31] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *IEEE International Conference on Computer Vision*, pages 3248–3255, 2013. 1, 2

[32] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999. 6, 7

[33] Onur Özyesil and Amit Singer. Robust camera location estimation by convex programming. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2015. 1, 6, 7, 8, 14

[34] Jaehyun Park and Stephen Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. *arXiv:1703.07870*, 2017. 5

[35] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *International Journal of Robotics Research*, 38(2-3):95–125, 2019. 1, 2, 5, 6, 7, 14, 15

[36] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 2, 7, 8

[37] Tianwei Shen, Siyu Zhu, Tian Fang, Runze Zhang, and Long Quan. Graph-based consistent matching for structure-from-motion. In *European Conference on Computer Vision*, volume 9907, pages 139–155, 2016. 3

[38] Chris Sweeney, Torsten Sattler, Tobias Höllerer, Matthew Turk, and Marc Pollefeys. Optimizing the viewing graph for structure-from-motion. In *IEEE International Conference on Computer Vision*, pages 801–809, 2015. 1, 2

[39] Yulun Tian, Kasra Khosoussi, and Jonathan P. How. Block-coordinate minimization for large SDPs with block-diagonal constraints. *arXiv: 1903.00597*, 2019. 4, 13

[40] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - A modern synthesis. In *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms*, pages 298–372, 1999. 1

[41] Roberto Tron, Xiaowei Zhou, and Kostas Daniilidis. A survey on rotation optimization in structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1032–1040, 2016. 1

[42] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference*, 2(2):145–194, 2013. 2

[43] Po-Wei Wang, Wei-Cheng Chang, and J. Zico Kolter. The mixing method: coordinate descent for low-rank semidefinite programming. *arXiv: 1706.00476*, 2017. 4

[44] Zaiwen Wen, Donald Goldfarb, Shiqian Ma, and Katya Scheinberg. Row by row methods for semidefinite programming. Technical report, 2009. 2

[45] Zaiwen Wen, Donald Goldfarb, and Wotao Yin. Alternating direction augmented lagrangian methods for semidefinite programming. *Math. Program. Comput.*, 2(3-4):203–230, 2010. 2

[46] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *Computer Vision European Conference*, pages 61–75, 2014. 1, 7, 8, 14, 15, 16, 17

[47] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision*, pages 127–134, 2013. 2

[48] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1426–1433, 2010. 3

[49] Qianggong Zhang, Tat-Jun Chin, and Huu Minh Le. A fast resection-intersection method for the known rotation problem. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3012–3021, 2018. 7

[50] Siyu Zhu, Tianwei Shen, Lei Zhou, Runze Zhang, Jinglu Wang, Tian Fang, and Long Quan. Parallel structure from motion from local increment to global averaging. *arXiv: 1702.08601*, 2017. 1, 2

[51] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global SfM by distributed motion averaging. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4568–4577, 2018. 1, 2

[52] Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. Baseline desensitizing in translation averaging. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4539–4547, 2018. 1

APPENDIX

1. Notations and Preliminaries

For clarity, we introduce some notations from [5]. We first introduce the linear operator $\mathcal{A} : \mathbb{R}^{n\times n} \longrightarrow \mathbb{S}^{n\times n}$ that symmetrizes diagonal blocks and zeroes out other blocks:

$$\mathcal{A}(M)_{ij} = \begin{cases} \dfrac{M_{ii} + M_{ii}^T}{2} & i = j, \\ 0 & otherwise. \end{cases} \tag{25}$$

Then we introduce the manifold with an equality constraint as

$$\mathrm{St}(d,p)^n = \{Y \in \mathbb{R}^{p\times m} : \mathcal{A}(YY^T) = I_m\}. \tag{26}$$

where $p \times d$ is the block size and $m = nd$.

The tangent space to $\mathrm{St}(d,p)^n$ at Y is a subspace of $\mathbb{R}^{p\times m}$ obtained by differentiating the equality constraint:

$$\mathrm{T}_Y \,\mathrm{St}(d,p)^n = \{Y \in \mathbb{R}^{p\times m} : \mathcal{A}(YY^T + YY^T) = 0\}. \tag{27}$$

The othogonal projector from the embedding space $\mathbb{R}^{p\times m}$ to the tangent space at Y is

$$\mathrm{Proj}_Y(Z) = Z - \mathcal{A}(ZY^T)Y. \tag{28}$$

The Riemmannian gradient is defined as

$$\mathrm{grad}\, f(Y) = \mathrm{Proj}_Y(\nabla f(Y)). \tag{29}$$

2. Derivation of Globally Optimal Rotation Averaging

Given a set of relative rotations $\{R_{ij}\}$, where $i, j \in [n]$, our aim is to obtain the absolute rotations $\{R_i\}$ that satisfy the constraints

$$R_{ij} = R_j R_i^{-1} = R_j R_i^T \tag{30}$$

between absolute and relative rotations. Usually there are more edges than nodes in an undirected graph, so we have more constraints than unknowns. Rotation averaging can be formulated as

$$\min_{R_1,\cdots,R_n} \sum_{(i,j)\in E} d^p(R_{ij}, R_j R_i^T), \tag{31}$$

where $d^p(\cdot)$ represents a distance measure under a p-norm. While there are a lot of local methods [24, 25, 11, 12] giving a least-squares solution to problem (31), here we exploit a global optimization approach that can obtain the global optimum.

We choose chordal distances as the distance measure in (31). Each residual along an edge of $\mathcal{G}$ will hence read

$$\begin{aligned} &\|R_j - R_{ij}R_i\|_F^2 \\ &= \|R_j\|_F^2 - 2\,\mathrm{tr}(R_j^T R_{ij} R_i) + \|R_i\|_F^2 \\ &= 6 - 2\,\mathrm{tr}(R_j^T R_{ij} R_i). \end{aligned} \tag{32}$$

The set of absolute rotations can be represented by

$$R = [R_1\ R_2\ \cdots\ R_n], \tag{33}$$

and the graph $\mathcal{G}$ can be represented by

$$G = \begin{bmatrix} 0 & a_{12}R_{12} & \cdots & a_{1n}R_{1n} \\ a_{21}R_{21} & 0 & \cdots & a_{2n}R_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}R_{n1} & a_{n2}R_{n2} & \cdots & 0 \end{bmatrix}, \tag{34}$$

where $a_{ij} = 1$ if the edge between views i and j exists, and $a_{ij} = 0$ otherwise. By combining Eqs. (32), (33) and (34), we can rewrite problem (31) as

$$\min_R\ (6 - \mathrm{tr}(R^T G R)) \Leftrightarrow \min_R\ -\mathrm{tr}(R^T G R). \tag{35}$$

The primal problem of rotation averaging is finally given by

$$\min_R\ -\mathrm{tr}(R^T G R) \quad \text{s.t.} \quad R \in \mathrm{SO}(3)^n. \tag{36}$$

Eriksson *et al.* [19] solve the dual problem with determinant constraint relaxation

$$\begin{aligned} &\min_X\ -\mathrm{tr}(GX) \\ &\text{s.t.}\quad X_{ii} = I_3,\ i = 1,\cdots,n, \quad X \succeq 0, \end{aligned} \tag{37}$$

where X is partitioned as

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nn} \end{bmatrix}. \tag{38}$$

They furthermore prove that there is no duality gap between the primal problem (36) and its dual problem (37) if the maximum residuals stay below a certain threshold. If $R^\star$ is the global optimum of problem (36), the solution of problem (37) satisfies

$$X^\star = R^{\star T} R^\star. \tag{39}$$

For readers who are interested in the duality gap and global optimality proof, we kindly refer them to [19] for more details.

As the low-rank formulation of problem 37 has been introduced in our main paper, we simply outline it as follow:

$$\begin{aligned} &\min_Y\ -\mathrm{tr}(GY^TY) \\ &\text{s.t.}\quad Y = [Y_1\ Y_2\ \cdots\ Y_n], \quad Y_i^T Y_i = I, \forall i \in [n]. \end{aligned} \tag{40}$$

3. Fast View Graph Filtering

We present how the fast view graph filtering works in Fig. 5. In Fig. 5, the blue nodes represent images, the *red solid lines* represent the edges in the selected maximum spanning tree (MaxST), the *blue dotted lines* are edges that need to be validated, and the *green solid lines* are edges that passed the verification. In Fig. 5 (a), we deem the edges in the MaxST as valid. Then, we can collect all the weak triplets based on the MaxST: $(a,c,b),(b,c,e),(e,d,b),(e,d,h),(g,f,i),(h,i,e)$ before the first iteration. In Fig. 5 (b), edges $(a,b),(c,e),(e,i),(f,i)$ passed the verification, edges $(d,b),(d,h)$ are deleted as they failed to pass the verification. Based on the first iteration, we also collect all the weak triplets $(c,e,d),(e,i,f),(e,i,g)$. In Fig. 5 (c), $(c,d),(e,g)$ are deleted due to it fails to pass the verification, (e,f) is marked by green solid line as it survives the validation. Fig. 5 (d), we only need to validate weak triplet (b,e,f), suppose (e,g) passed the verification and we mark it by green solid line.

4. Proof of Sublinear Convergence Rate

In this section, we first provide two lemmas upfront, which are needed for our main result. Then we prove the sublinear convergence rate of our low-rank block coordinate descent algorithm

Lemma 4.1. *Each iteration of the BCM with low-rank factorization yields descent on the function value. Specifically,*

$$f(Y^{k+1}) - f(Y^k) = -2\left(||Q_{j_k}^k||_* - \langle Q_{j_k}^k, Y_{j_k}^k\rangle\right) \leq 0. \tag{41}$$

Proof. Suppose we choose the j_k-th block at $(k+1)$-th iteration. From the definition of $f(Y^k)$ and the update rule of Q_j^{k+1}, we can obtain

$$\begin{aligned} f(Y^{k+1}) &= \sum_{j=1}^{n} \langle Q_j^{k+1}, Y_j^{k+1}\rangle \\ &= \langle Q_{j_k}^{k+1}, Y_{j_k}^{k+1}\rangle + \sum_{j\neq j_k}^{n} \langle Q_j^{k+1}, Y_j^{k+1}\rangle. \end{aligned} \tag{42}$$

Let the second term of Eq.(42) represented as $S = \sum_{j\neq j_k}^{n} \langle Q_j^{k+1}, Y_j^{k+1}\rangle$. Since S is independent of Y_{j_k}, we have $Y_j^{k+1} = Y_j^k$. Then we can rewrite S as

$$\begin{aligned} S &= \sum_{j\neq j_k}^{n} \langle Q_j^{k+1}, Y_j^k\rangle \\ &= \sum_{j\neq j_k}^{n} \langle Q_j^k + (Y_{j_k}^{k+1} - Y_{j_k}^k) G_{j_k j}, Y_j^k\rangle \\ &= \sum_{j\neq j_k}^{n} \langle Y_{j_k}^{k+1} - Y_{j_k}^k, Y_j^k\rangle G_{j_k j} + \sum_{j\neq j_k}^{n} \langle Q_j^k, Y_j^k\rangle \\ &= \langle Y_{j_k}^{k+1} - Y_{j_k}^k, Q_{j_k}^k\rangle + \sum_{j\neq j_k}^{n} \langle Q_j^k, Y_j^k\rangle \\ &= \langle Y_{j_k}^{k+1} - Y_{j_k}^k, Q_{j_k}^k\rangle + \sum_{j=1}^{n} \langle Q_j^k, Y_j^k\rangle - \langle Q_{j_k}^k, Y_{j_k}^k\rangle \\ &= \langle Y_{j_k}^{k+1} - 2Y_{j_k}^k, Q_{j_k}^k\rangle + f(Y_k). \end{aligned} \tag{43}$$

Substitute (43) into (42), we obtain

$$\begin{aligned} f(Y^{k+1}) - f(Y^k) &= 2\langle Q_{j_k}^{k+1}, Y_{j_k}^{k+1}\rangle - 2\langle Q_{j_k}^k, Y_{j_k}^k\rangle \\ &= -2\left(\left\|Q_{j_k}^k\right\|_* - \langle Q_{j_k}^k, Y_{j_k}^k\rangle\right). \end{aligned} \tag{44}$$

Using the general Von Neumann trace theorem, we have

$$\begin{aligned} |\langle Q_{j_k}^k, Y_{j_k}^k\rangle| &\leq \sum_i \sigma_i(Q_{j_k}^k)\sigma_i(Y_{i_k}^k) \\ &= \sum_i \sigma_i(Q_{j_k}^k) = \left\|Q_{j_k}^k\right\|_*. \end{aligned} \tag{45}$$

Thus, we can obtain $f(Y^{k+1}) - f(Y^k) = -2\left(\left\|Q_{j_k}^k\right\|_* - \langle Q_{j_k}^k, Y_{j_k}^k\rangle\right) \leq 0$, which concludes the proof of Lemma 3.1. □

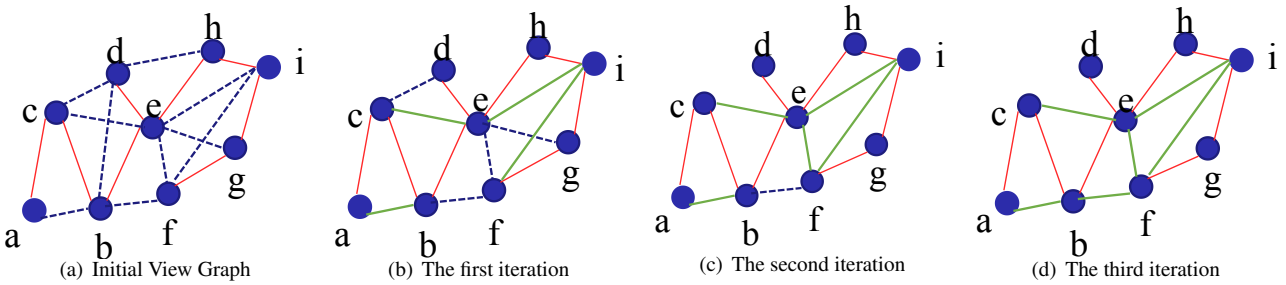
Lemma 4.2. *Let* $A_j = \frac{1}{2}\left(Y_j^T Q_j + Q_j^T Y_j\right), \forall j \in [n]$, *then*

$$\|\operatorname{grad} f(Y)\|_F^2 = 4\sum_{j=1}^{n}\left(\|Q_j\|_F^2 - \|A_j|\,|_F^2\right), \tag{46}$$

where $\operatorname{grad} f(Y)$ *represents the Riemannian gradient of* $f(Y)$ *[5].*

Proof. Let function $B(\cdot)$ represents the block diagonal of matrix.

$$\begin{aligned} \operatorname{grad} f(Y) &= \operatorname{Proj}_Y(\nabla f(Y)) \\ &= 2\operatorname{Proj}_Y(YG) \\ &= 2\left(YG - Y\frac{1}{2}B\left(Y^T YG + GY^Y\right)\right). \end{aligned} \tag{47}$$



(a) Initial View Graph (b) The first iteration (c) The second iteration (d) The third iteration

Figure 5. The procedure of fast view graph filtering, where the blue nodes represent images, the *red solid lines* represent the edges in the selected maximum spanning tree (MaxST), the *blue dotted lines* are edges that need to be validated, and the *green solid lines* are edges that passed the verification.

The j-th block of $Y^TYG + GY^TY$ is

$$\begin{aligned}
&\frac{1}{2}B\left(Y^TYG + GY^Y\right)_{jj} \\
=&\frac{1}{2}\sum_{i=1}^{n}\left(Y_j^TY_iG_{ij} + G_{ji}Y_i^TY_j\right) \\
=&\frac{1}{2}Y_j^T\sum_{i\neq j}Y_iG_{ij} + \frac{1}{2}(\sum_{i\neq j}G_{ji}Y_i^T)^TY_j \\
=&\frac{1}{2}Y_j^TQ_j + \frac{1}{2}Q_j^TY_j = A_j.
\end{aligned} \tag{48}$$

Then, (47) can be rewritten as

$$\operatorname{grad} f(Y) = 2Y\begin{bmatrix} -A_1 & G_{12} & \cdots & G_{1n} \\ G_{21} & -A_2 & \cdots & G_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ G_{n1} & G_{n2} & \cdots & -A_n \end{bmatrix}. \tag{49}$$

We take $\operatorname{grad} f(Y)$ as a row-block matrix, and the j-th block can be written as

$$\operatorname{grad} f(Y)_j = 2(-Y_jA_j + Q_j). \tag{50}$$

Then, we have

$$\begin{aligned}
&\|\operatorname{grad} f(Y)_j\|_F^2 \\
=&4\operatorname{tr}\left(\left(-Y_jA_j + Q_j\right)^T\left(-Y_jA_j + Q_j\right)\right) \\
=&4\operatorname{tr}(A_j^TY_j^TY_jA_j + Q_j^TQ_j - Q_j^TY_jA_j - A_j^TY_j^TQ_j) \\
=&4(\|A_j\|_F^2 + \|G_j\|_F^2 - \operatorname{tr}((G_j^TY_j + Y_j^TG_j)A_j)) \\
=&4(\|Q_j\|_F^2 - \|A_j\|_F^2).
\end{aligned} \tag{51}$$

This yields Lemma 3.2

$$\begin{aligned}
\|\operatorname{grad} f(Y)\|_F^2 &= \sum_{j=1}^{n}\|\operatorname{grad} f(Y)_j\|_F^2 \\
&= 4\sum_{j=1}^{n}\left(\|Q_j\|_F^2 - \|A_j|\,|_F^2\right).
\end{aligned} \tag{52}$$

□

Next, we prove a global sublinear convergence rate of the proposed low-rank BCM approach for rotation averaging.

Theorem 1. *Let f^* be the optimal value of problem (40), and $C(G) = \max_j\sum_{i\neq j}\|G_{ij}\|_*$. Then, for any $K \geq \lceil 2dnC(G)\left(f(Y^0) - f^*\right)/\epsilon\rceil$ iterations, the low-rank BCM approach is guaranteed to return a solution Y^k for some $k \in [K-1]$ that satisfies $\mathbb{E}\left(\left\|\operatorname{grad} f(Y^k)\right\|_F^2\right) \leq \epsilon$, where $\mathbb{E}$ represents the expectation over j_k given Y^k. Equivalently, for any $K \geq 1$, the low-rank BCM yields the guarantee*

$$\min_{k\in[K-1]}\mathbb{E}\left\|\operatorname{grad} f(Y^k)\right\|_F^2 \leq \frac{2dnC(G)(f(Y^0) - f^*)}{K}. \tag{53}$$

Proof. According to Lemma 4.1, we have

$$\begin{aligned}
&f(Y^{k+1}) - f(Y^k) = -2(||Q_{j_k}^k||_* - \langle Q_{j_k}^k, Y_{j_k}^k\rangle) \\
=&-2\frac{||Q_{j_k}^k||_*^2 + ||Q_{j_k}^k||_*\langle Q_{j_k}^k, Y_{j_k}^k\rangle}{||Q_{j_k}^k||_*} \\
\leq&-2\frac{||Q_{j_k}^k||_*^2 - \langle Q_{j_k}^k, Y_{j_k}^k\rangle^2}{||Q_{j_k}^k||_*} \\
=&-2\frac{||Q_{j_k}^k||_*^2 - \operatorname{tr}(A_{j_k}^k)^2}{||Q_{j_k}^k||_*} \\
\leq&-2\frac{||Q_{j_k}^k||_F^2 - \left\|A_{j_k}^k\right\|_F^2}{||Q_{j_k}^k||_*},
\end{aligned} \tag{54}$$

where $\|\cdot\|_*$ represents the nuclear norm. The third term of inequality (54) is obtained from the general Von Neumann trace theorem [**?**], and the last term of (54) is obtained from the lemma of [39]. Let the expectation of $f(Y^{k+1})$ be $\mathbb{E}_k(f(Y^{k+1}))$ and $D = \mathbb{E}_k(f(Y^{k+1})) - f(Y^k)$. We can

prove that

$$\left\|Q_j^k\right\|_* = \left\|\sum_{i\neq j}^{n} Y_i^k G_{ij}\right\|_* \leq \sum_{i\neq j}^{n} \left\|Y_i^k G_{ij}\right\|_*$$
$$\leq \sum_{i\neq j}^{n} \left\|Y_i^k\right\|_* \left\|G_{ij}\right\|_* = d\sum_{i\neq j}^{n} \left\|G_{ij}\right\| = dC(G), \quad (55)$$

and it leads to the inequality

$$D \leq 2\sum_{j=1}^{n} p_i \frac{\left\|Q_{j_k}^k\right\|_F^2 - \left\|A_{j_k}^k\right\|_F^2}{\langle Q_{j_k}^k, Y_{j_k}^k\rangle}$$
$$\leq \frac{2}{n\max_j \left\|Q_j^k\right\|_*} \sum_{j=1}^{n} \left(\left\|Q_{j_k}^k\right\|_F^2 - \left\|A_{j_k}^k\right\|_F^2\right)$$
$$= \frac{2}{n\max_j \left\|Q_j^k\right\|_*} \left\|\operatorname{grad} f\left(Y^k\right)\right\|_F^2$$
$$\leq \frac{1}{2dnC(G)} \left\|\operatorname{grad} f\left(Y^k\right)\right\|_F^2. \quad (56)$$

The second term is obtained from Lemma 4.2.

In order to prove inequality (53), we assume the contrary that $\mathbb{E}\left(\left\|\operatorname{grad} f\left(Y^k\right)\right\|_F^2\right) > \epsilon, \forall k \in [K-1]$ and some integer K. Then we have

$$f\left(Y^0\right) - f^* \geq f\left(Y^0\right) - \mathbb{E}\left(f\left(Y^k\right)\right)$$
$$= \sum_{k=0}^{K-1} \mathbb{E}\left(f\left(Y^k\right) - f\left(Y^{k+1}\right)\right)$$
$$= \sum_{k=0}^{K-1} \mathbb{E}\left(f\left(Y^k\right) - \mathbb{E}_k\left(f\left(Y^{k+1}\right)\right)\right) \quad (57)$$
$$\geq \frac{1}{2dnC(G)} \sum_{k=0}^{K-1} \mathbb{E}\left\|\operatorname{grad} f(Y^k)\right\| > \frac{K\epsilon}{2dnC(G)}.$$

By contradiction, the algorithm returns a solution with $\mathbb{E}\left(\left\|\operatorname{grad} f\left(Y^k\right)\right\|_F^2\right) \leq \sigma$ for some $k \in [K-1]$. Thus we complete the proof that

$$K > \frac{2dnC(G)\left(f\left(Y^0\right) - f^*\right)}{\epsilon}. \quad (58)$$

□

From Theorem 1, it can be seen that the low-rank BCM approach attains a sublinear convergence rate.

5. Experiment

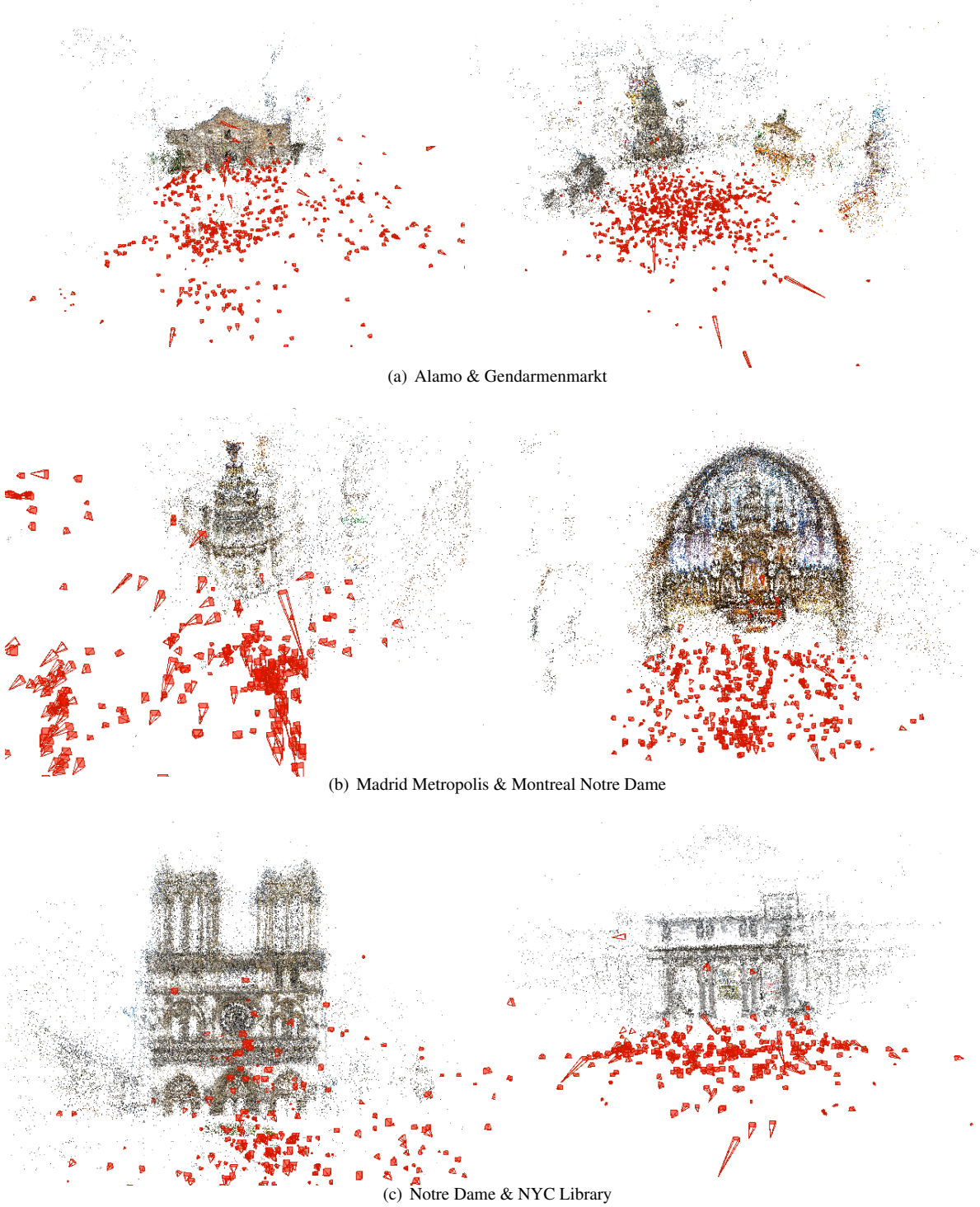
We additionally embed our rotation averaging approach in a global SfM pipeline, where LUD [33] is adopted as the translation averaging approach. In this section, we provide the statistics results that reconstructed from internet datasets [46], where our method is compared against the local rotation averaging solver [12], Eriksson's method [19], and SE-Sync [35]. The visual results of our method are shown in Fig.6 and Fig.7.

For all global rotation averaging approaches (Eriksson's method, SE-Sync and our method), we stop the solver when the relative function error change is less than $1e{-}7$. For SE-Sync and our method, the initial rank and maximum rank are respectively 3 and 10.

In terms of registered camera number and solver efficiency, our approach obtained the best results among most datasets, and Chatterjee's method [12] can obtain the second best results in most datasets. As both our approach and SE-Sync [35] need to validate the global optimality, and restart the low-rank local solver when $Y^\star$ is not the second-critical point, thus our approach behaves a little slower than Chatterjee's method. In terms of accuracy, Chatterjee's method can obtain most best results, and our approach has comparable accuracy to Chatterjee's method. In Trafalgar dataset, Eriksson's method spares more than a week to solve the problem, thus we stopped the method and mark the corresponding table cells by '–'.

Dataset	N_i	**Chatterjee** [12]				**Eriksson** [19]				**SE-Sync** [35]				**Ours**			
		N_c	$time(s)$	$\bar{R}$	$\tilde{R}$	N_c	$time(s)$	$\bar{R}$	$\tilde{R}$	N_c	$time(s)$	$\bar{R}$	$\tilde{R}$	N_c	$time(s)$	$\bar{R}$	$\tilde{R}$
Alamo	627	558	3.314	0.041	0.024	509	2701.060	0.081	0.045	556	4.454	0.075	0.059	581	2.948	0.058	0.033
Ellis Island	247	232	0.475	0.157	0.115	206	94.259	0.195	0.176	210	3.466	0.120	0.158	226	0.125	0.102	0.096
Gendarmenmarkt	742	705	1.950	1.033	0.159	583	6220.350	0.895	0.350	703	4.790	1.102	0.406	697	2.862	0.945	0.289
Madrid Metropolis	394	358	0.801	0.063	0.037	335	617.105	0.122	0.059	350	0.883	0.079	0.051	354	0.150	0.078	0.045
Montreal N.D.	474	461	1.600	0.018	0.010	349	1124.830	0.044	0.018	448	2.218	0.016	0.011	464	0.267	0.015	0.009
Notre Dame	553	549	3.608	0.013	0.009	547	86.145	0.034	0.015	550	4.094	0.022	0.014	550	0.558	0.014	0.010
NYC Library	376	341	0.416	0.130	0.120	317	665.065	0.162	0.140	336	4.236	0.134	0.124	343	0.105	0.106	0.093
Piazza del Popolo	354	328	0.788	0.148	0.145	309	470.836	0.187	0.148	326	5.275	0.169	0.150	333	0.538	0.108	0.097
Piccadilly	2508	2300	21.119	0.122	0.059	1842	277967.500	0.241	0.155	2287	22.151	0.201	0.168	2311	10.897	0.183	0.136
Roman Forum	1134	1061	1.520	0.087	0.076	529	23745.200	0.328	0.367	1045	10.730	0.097	0.083	1069	11.083	0.067	0.056
Tower of London	508	478	0.727	0.049	0.031	297	1325.855	0.084	0.082	487	4.506	0.067	0.048	485	0.206	0.059	0.039
Trafalgar	5433	5057	56.767	0.074	0.046	-	-	-	-	5063	95.381	0.081	0.073	5068	110.573	0.066	0.052
Union Square	930	812	0.955	0.163	0.129	693	6882.200	0.245	0.189	803	4.047	0.235	0.184	824	2.081	0.172	0.137
Vienna Cathedral	918	851	3.195	0.238	0.092	684	12046.450	0.740	0.685	850	13.374	0.320	0.260	862	5.151	0.285	0.156
Yorkminster	458	431	0.601	0.083	0.070	390	1033.385	0.152	0.090	421	0.547	0.139	0.114	421	0.522	0.096	0.093

Table 3. Comparison of runtime and accuracy on internet datasets [46]. For runtime and the registered camera number, the first, second and third best results are highlighted in color. For rotation error, we only highlight the best. N_i denotes the number of images in the largest connected component of the viewing graph given in [46], N_c denotes the number of registered cameras, $\bar{R}$ denotes the mean of rotation errors, and $\tilde{R}$ denotes the median of rotation errors (unit: radian).

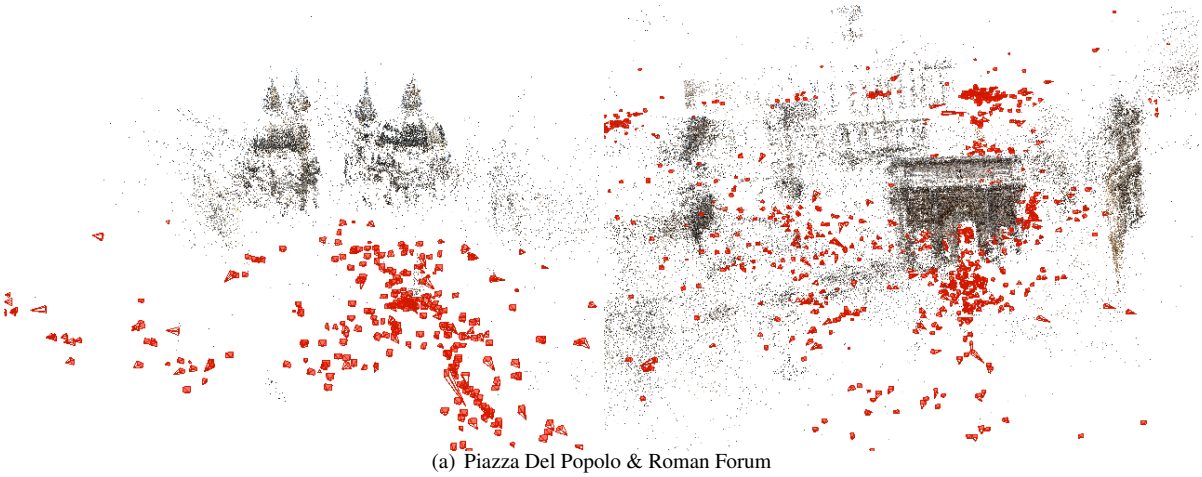


(a) Alamo & Gendarmenmarkt

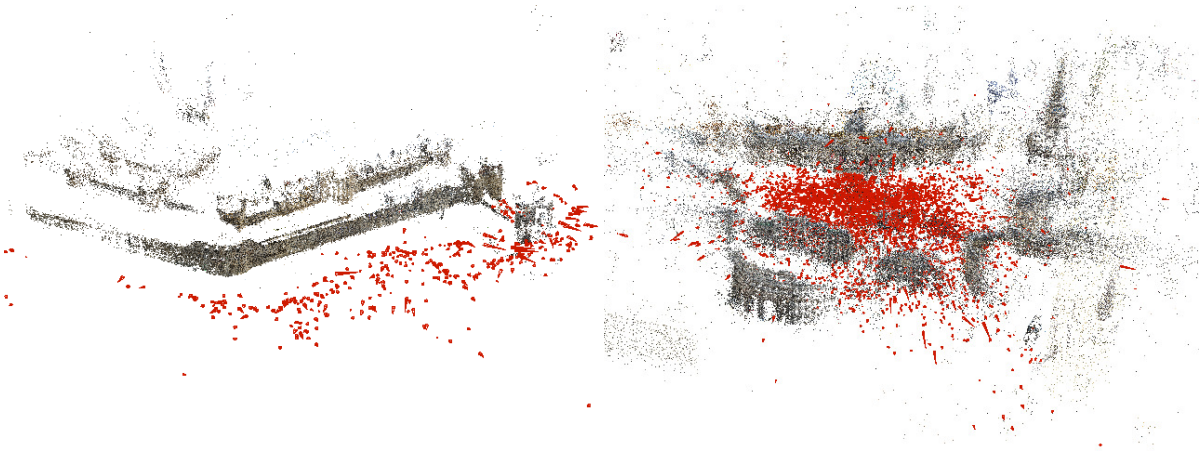
(b) Madrid Metropolis & Montreal Notre Dame

(c) Notre Dame & NYC Library

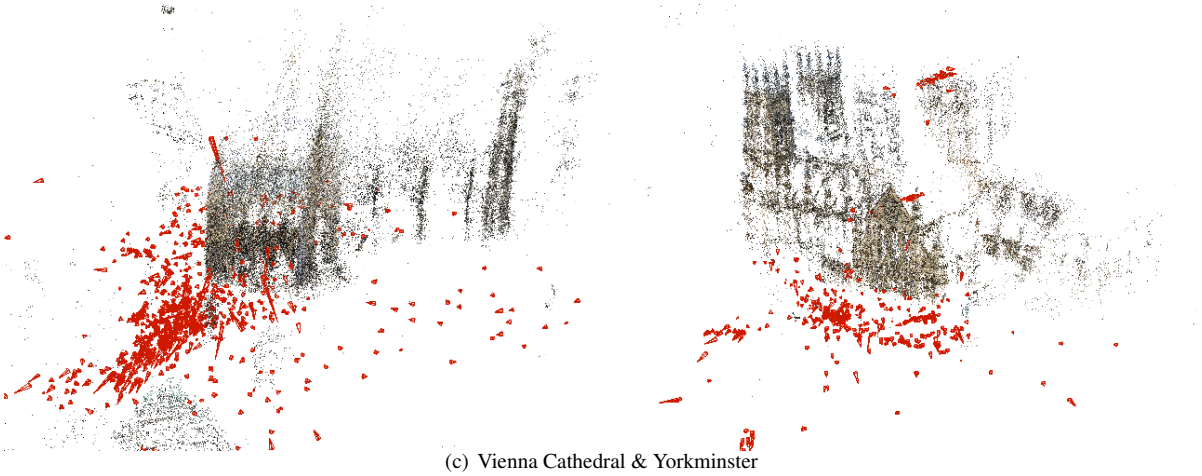
Figure 6. Reconstruction results of internet datasets [46]. From top to bottom are Alamo ,Gendarmenmarkt, Madrid Metropolis, Montreal Notre Dame, Notre Dame, NYC Library, respectively.



(a) Piazza Del Popolo & Roman Forum



(b) Tower of London & Trafagar



(c) Vienna Cathedral & Yorkminster

Figure 7. Reconstruction results of internet datasets [46]. From top to bottom are Piazza Del Popolo, Roman Forum, Tower of London, Trafagar, Vienna Cathedral, Yorkminster, respectively.