

# Hybrid Rotation Averaging: A Fast and Robust Rotation Averaging Approach

Yu Chen

Peking University

hackerdreamer34@gmail.com

Ji Zhao

TuSimple

zhaoji84@gmail.com

Laurent Kneip

ShanghaiTech University

lkneip@shanghaitech.edu.cn

## Abstract

We address rotation averaging (RA) and its application to real-world 3D reconstruction. Local optimisation based approaches are the *de facto* choice, though they only guarantee a local optimum. Global optimisers 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 rotation averaging by leveraging the advantages of a global RA method and a local RA method. Combined with a fast view graph filtering as preprocessing, the proposed hybrid approach is robust to outliers. We further apply the proposed hybrid rotation averaging approach to incremental Structure from Motion (SfM), the accuracy and robustness of SfM are both improved by adding the resulting global rotations as regularisers 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) [29, 15, 36, 49, 50], as well as Simultaneous Localization and Mapping (SLAM) [7] 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 [25, 31, 21, 51] to also obtain absolute positions. Triangulation of



Figure 1. Reconstructions generated from the Trafalgar dataset [45], where 7085 out of 15685 images have been registered, and the rotation averaging step only took 4.2 s.

3D points and joint optimisation over all parameters (i.e. bundle adjustment [39]) completes the reconstruction. In SLAM, rotation averaging has been used in the back-end pose graph optimisation [33, 7] to flexibly encounter large drift accumulations or—more generally—replace the time-consuming bundle adjustment step.

Rotation averaging was first proposed using the quaternion representation [22]. 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 [8] 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 [40, 33, 18, 17, 16]. 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 the rotation averaging method, and on pushing its application to challenging scenes. We make a combination of a global solver and a local solver to solve rotation averaging, which guarantees global optimality and has strong resilience against outliers. Rotation averaging based on chordal distances can be reformulated as a semi-definite program (SDP) with a low-rank constraint. Taking advantage of the low-rank factorisation of the original SDP, we

can apply globally optimal Riemannian-Staircase-based [3] methods. In principle, any global solver [33, 37, 16] can be used in our hybrid approach, and we adopt the block coordinate minimisation method [37] to better leverage the view graph sparsity. By preprocessing the graph with fast view graph filtering, graph sparsity can be further exploited to accelerate the optimisation.

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.* [13], who proposes 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 estimated absolute rotations are not correct, thus it prevents camera centers from proper registration and scene reconstructions from completeness. Inspired by Cui’s approach [13], 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 propose a novel cost function to optimise camera poses and landmarks that alleviates drift accumulation, in which camera rotations obtained from rotation averaging are used as regularisers. The resulting SfM approach, named RA-SfM (Rotation-Averaged Structure from Motion), shows high practicality and surpasses state-of-the-art methods in accuracy, which is demonstrated at the hand of extensive experiments on large-scale real-world datasets. The reconstruction result of the largest dataset is shown in Fig. 1.

In summary, the main contributions of our work are:

- We propose an outlier-resilient hybrid rotation averaging approach, which combines a global optimiser with fast view graph filtering and a local optimiser.
- We refine the traditional incremental bundle adjustment cost function by adding the obtained global rotations as a regularisation term, which significantly alleviates drift accumulation in incremental SfM.

The practicality and superiority of the proposed scheme is demonstrated by extensive experiments on synthetic datasets and challenged internet datasets.

## 2. Related Work

Motion averaging [22, 23] is widely used in global SfM pipelines [29, 15, 36, 49, 50] as an answer to the drift problem occurring in incremental SfM [2, 46, 34, 14]. The first solution to rotation averaging goes back to Govindu [22], 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 [23]. 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) [8, 9]. Despite great progress, all aforementioned approaches can only guarantee a locally optimal solution. Another local approach was proposed by Crandall *et al.* [11, 12], 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 [20] 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.* [17] 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 [5]. Wang and Singer [41] solve it by the Alternating Direction Method of Multipliers (ADMM) [4, 44]. Eriksson *et al.* [17] use a row-by-row block coordinate descent method (BCM) [43]. 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 [6], 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 [3], 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.* [33] 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 [3] to guarantee global optimality [33]. Inspired by Wang’s work *et al.* [42], which solves the low-rank SDP problem by block coordinate descent method, Tian *et al.* [37] extends this work to Steifel manifold, and further applied a Riemannian BCM method to pose graph optimisation in distributed settings [38]. Building on SE-Sync [33], Dellaert *et al.* [16] propose *Shonan rotation averaging*, a method in which the rotation matrix is vectorized, thus permitting the use of existing gradient-based optimisation methods on the manifold

of rotation matrices.

### 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  $\text{tr}(\cdot)$  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  $\text{tr}(A^T B) = \langle A, B \rangle$ . Let  $\text{blockdiag}(A)$  represent the block diagonal matrix of  $A$ , and  $\text{symblockdiag}(A) = \frac{1}{2} \text{blockdiag}(A + A^T)$ .

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

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

Since  $\text{SO}(3)$  is a Lie group, there exists an exponential mapping between a rotation  $R$  and its Lie algebra  $\mathfrak{so}(3)$  representation  $w$  [28]:

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

The absolute rotations are grouped in  $\mathcal{R} = \{R_1, R_2, \dots, R_n\}$ , where  $R_i \in \text{SO}(3)$ ,  $i \in [n]$ . Relative rotations are represented by  $\mathcal{R}_{\text{rel}} = \{R_{ij}\}$ , where  $R_{ij} \in \text{SO}(3)$ ,  $i, j \in [n]$ ,  $i < j$  is the rotation from  $R_i$  to  $R_j$ . The chordal distance between two rotations is measured by [24]

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

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

## 4. Hybrid 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 apply a block coordinate descent (BCD) method [37] to optimise the low-rank formulation of rotation averaging. Its global optimality can be 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. Also, global optimality is only guaranteed when the residuals for each edge are bounded below a certain threshold [17]. (2) Some view pairs are redundant and even harm the quality of SfM results. Zach *et al.* [47] 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 below a given threshold  $\epsilon$

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

Despite its effectiveness, [47] needs to validate all triplets, which is impractical for large-scale datasets. However, [35] 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.

### 4.2. Global Rotation Averaging

In this section, we first review a globally optimal guaranteed rotation averaging method [37], then the sparsity pattern of the view graph is further exploited to accelerate the algorithm.

Given a set of relative rotations  $\{R_{ij}\}$ , where  $i, j \in [n]$ , the aim of rotation averaging is to obtain the absolute rotations  $\{R_i\}$  that minimises the cost function below:

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

where  $d^p(\cdot)$  represents a distance measure under a  $p$ -norm. While there are a lot of local methods [22, 23, 8, 9] giving a least-squares solution to problem (5), here we exploit a global optimisation approach that can obtain the global optimum. Adopting the chordal distances, the primal problem of rotation averaging is finally given by <sup>1</sup>

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

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

<sup>1</sup>See supplementary material for the complete derivation.

Eriksson *et al.* [17] solve the dual problem with determinant constraint relaxation to the primal problem (6)

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

where  $X$  can be written as a block matrix  $X_{ij}$ , with  $i, j \in [n]$  and  $X_{ij} \in \mathbb{R}^3$ .

Problem (7) is an SDP problem [5]. Since every  $X \succeq 0$  can be factored as  $Y^T Y$  for some  $Y$  [6], and—in the case of rotation averaging—the optimal value of  $X$  satisfies  $X^* = R^{*T} R^*$  [17], there is an implicit constraint on  $X$  such that  $\text{rank}(X) = 3$ . Thus,  $X$  can be reformulated as

$$X = Y^T Y, \quad (8)$$

where  $Y = [Y_1 \ Y_2 \ \dots \ Y_n]$ ,  $Y_i^T Y_i = I, \forall i \in [n]$ . By substituting (8) into problem (7), a new problem is obtained

$$\begin{aligned} \min_Y & -\text{tr}(GY^T Y) \\ \text{s.t.} & Y = [Y_1 \ Y_2 \ \dots \ Y_n], \quad Y_i^T Y_i = I, \forall i \in [n]. \end{aligned} \quad (9)$$

While Tian *et al.* [37] proposed a block coordinate minimisation (BCM) method to solve the low-rank SDP problem (9), we make a detailed derivation to the BCM solution, which is further accelerated by exploring graph sparsity. Note that

$$\begin{aligned} \text{tr}(GY^T Y) &= \text{tr}(YGY^T) = \langle YG, Y \rangle \\ &= \sum_{j=1}^n \langle \sum_{i=1}^n Y_i G_{ij}, Y_j \rangle = \sum_{j=1}^n \langle \hat{Q}_j, Y_j \rangle, \end{aligned} \quad (10)$$

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 (10) 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

$$\begin{aligned} Y_{j_k}^{k+1} &= \arg \min_{Y_{j_k}} f(Y_1^k, \dots, Y_{j_k-1}^k, Y_{j_k}^k, Y_{j_k+1}^k, \dots, Y_n^k) \\ &= \arg \min_{Y_{j_k}} \sum_{j=1}^n \langle \sum_{i \neq j}^n Y_i^k G_{ij}, Y_j^k \rangle \\ &= \arg \min_{Y_{j_k}} \langle Q_j^k, Y_{j_k}^k \rangle + \sum_{j \neq j_k} \sum_{i \neq j} \langle Y_i^k G_{ij}, Y_j^k \rangle \\ &= \arg \min_{Y_{j_k}} 2 \langle Q_j^k, Y_{j_k}^k \rangle + \sum_{j \neq j_k} \sum_{i \neq j, j_k} \langle Y_i^k G_{ij}, Y_j^k \rangle \\ &= \arg \min_{Y_{j_k}} 2 \langle Q_j^k, Y_{j_k}^k \rangle = \arg \min_{Y_{j_k}} \frac{1}{2} \|Y_{j_k}^k + Q_j^k\|_F^2. \end{aligned} \quad (11)$$

By solving problem (11), the update of  $Y_j$  in problem (9) can be determined by [27, 37]

$$Y_j^* = U_j I_{3 \times 3} V_j^T = U_j V_j^T, \quad (12)$$

where  $U_j \Sigma V_j$  is the singular value decomposition of  $-Q_j$ .

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

$$\begin{aligned} Q_j^{k+1} &= \sum_{i \neq j}^n Y_i^{k+1} G_{ij} = Y_{j_k}^{k+1} G_{j_k j} + \sum_{i \neq j, j_k}^n Y_i^{k+1} G_{ij} \\ &= Y_{j_k}^{k+1} G_{j_k j} + \sum_{i \neq j, j_k} Y_i^k G_{ij} + Y_{j_k}^k G_{j_k j} - Y_{j_k}^k G_{j_k j} \\ &= Y_{j_k}^{k+1} G_{j_k j} + \sum_{i \neq j} Y_i^k G_{ij} - Y_{j_k}^k G_{j_k j} \\ &= Q_j^k + (Y_{j_k}^{k+1} - Y_{j_k}^k) G_{j_k j}. \end{aligned} \quad (13)$$

In Algorithm 1, we outline the BCM with graph sparsity. 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. Notice that, with our fast view graph filtering, the graph can be more sparse and we can gain more acceleration.

---

#### Algorithm 1 BCM for SDP [37] with Graph Sparsity

---

**Input:** relative rotations  $\mathcal{R}_{\text{rel}}$ ,  $\text{maxIterNum}$ ,  $Y^0$ .

**Output:** First-order critical point  $Y^*$

- 1:  $k \leftarrow 0$ ;  $Q_j^0 \leftarrow \sum_{i \neq j}^n Y_i G_{ij}, \forall j \in [n]$ .
  - 2: **while**  $k < \text{maxIterNum}$  AND not converge **do**
  - 3:     **for**  $i < n$  **do**
  - 4:          $j_k \leftarrow i$
  - 5:         Update  $Y_{j_k}^{k+1}$  by Eq. (12)
  - 6:         **for**  $\forall j \neq j_k$  AND  $G_{j_k j} \neq \mathbf{0}$  **do**
  - 7:             Update  $Q_j^{k+1}$  by Eq. (13)
  - 8:      $k \leftarrow k + 1$ ;
  - 9: return  $Y$
- 

**Discussion of Global Optimality:** Problem (9) is non-convex, and there is no guarantee that we can obtain the global optimum. In [42], the global optimum is guaranteed by selecting an appropriate step size and random initialisation (Theorem 3.4). However, the theorem only holds for scalar variables. For the RA problem, global optimality is not held since we optimise on the manifold. Boumal [3] proposes a general framework named the *Riemannian Staircase* algorithm (RS), which can find the global optimum. As previous work has applied a Riemannian based method to ensure the global optimum [33, 38, 16], we recommend interested readers to refer to them.

### 4.3. Local Optimization 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 low-rank BCM method, we follow the *suggest-and-improve* framework of [32]. The global optimisation approaches can obtain a good solution close to the global optimum. Still, it could be further refined by a gradient descent algorithm. We adopt this framework and use the method of Chatterjee and Govindu [8] as a local optimiser. This method performs an Iteratively Reweighted Least Square (IRLS) under Lie algebra, which leads to an efficient and robust optimiser. The rotation  $R_{ij}, R_i, R_j$  can be represented by the corresponding Lie algebra  $\omega_{ij}, \omega_i, \omega_j$ , respectively. Using the Baker-Campbell-Hausdorff (BCH) equation, a single constraint in Eq. (5) can be converted to

$$\omega_{ij} = \omega_j - \omega_i. \quad (14)$$

By collecting the relative constraints, we obtain

$$A\omega_{\text{global}} = \omega_{\text{rel}}. \quad (15)$$

Here  $A$  is a sparse matrix, in which all consecutive  $3 \times 3$  blocks are zeros except two matrices  $I$  and  $-I$ . Encapsulating Eq. (15) with a Huber loss  $\rho(x) = \frac{x^2}{x^2 + \sigma^2}$ , we can optimise a robust cost function under least square meaning

$$\arg \min_{\omega_{\text{global}}} \sum \rho(\|A\omega_{\text{global}} - \omega_{\text{rel}}\|). \quad (16)$$

### 4.4. Hybrid of Global Rotation Averaging

We outline our hybrid rotation averaging algorithm in Algorithm 2. An ablation study of robustness against outliers is shown in Fig. 2. The outlier ratio ranges from 0 to 50% and is 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.

---

#### Algorithm 2 Hybrid Rotation Averaging Algorithm

---

**Input:** relative rotations  $\mathcal{R}_{\text{rel}}$

**Output:** global rotations  $\mathcal{R} = \{R_1, R_2, \dots, R_n\}$

- 1: Perform fast VGF as described in Sec. 4.1.
  - 2: Calculate global rotations using Algorithm 1 (or any other global rotation averaging method).
  - 3: Refine global rotations by solving problem (16).
- 

## 5. Rotation-Averaged 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

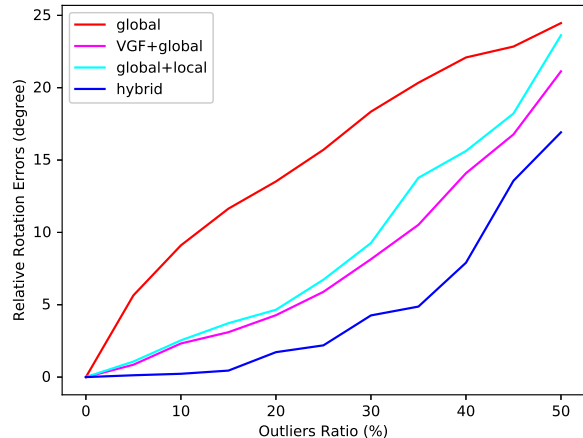


Figure 2. Ablation study of robustness. Outliers are generated by perturbing ground truth by rotation between  $60^\circ - 90^\circ$ . *global* represents the proposed low-rank BCM method, and *hybrid* represents the proposed hybrid method with VGF and local refinement.

summarised as follows: We first construct the view graph and obtain global rotations from our proposed hybrid 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 [19] P3P [26] 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 with averaged rotations as regularisers for bundle adjustment. Let  $\mathcal{I}_i$  denote the measurements of image  $i$ . 3D landmarks observed by image  $i$  are denoted as set  $\mathcal{P}_i$ . Note that the sets  $\{\mathcal{P}_i | i \in \mathcal{I}\}$  might have repetitive elements, which—in a slight abuse of notation—is ignored for the sake of simplicity. Let  $\mathbf{u}_{il} \in \mathcal{I}_i$  furthermore denote the image keypoint measurement of landmark  $l$  in frame  $i$ . the pre-computed known rotation with respect to image  $i$  is denoted as  $\hat{\mathcal{R}}_i$ .

The proposed cost function is given by

$$\sum_{i \in \mathcal{I}} \sum_{l \in \mathcal{I}_i} \rho_v(\|\mathbf{r}_{\mathcal{I}_{il}}\|^2) + \sum_{(i,j) \in \mathcal{E}} w_{ij} (\|\mathbf{r}_{\mathcal{R}_{ij}}\|^2), \quad (17)$$

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. The objective divides into two terms which are explained as follows.

**Visual Term:** We adopt the traditional re-projection er-

ror in bundle adjustment as our visual term

$$\mathbf{r}_{\mathcal{I}_{il}} = \mathbf{u}_{il} - \Pi(R_i, C_i, \mathcal{P}_i, l), \quad (18)$$

where  $R_i$  and  $C_i$  are respectively the estimated camera rotation and center, and  $\Pi(\cdot)$  is the back-projection function that projects landmarks into the image plane. Note that the latter also depends on camera intrinsics, which—for the sake of a simplified and general notation—are not specified.

**Known Rotation Term:** The added known rotation term is

$$\mathbf{r}_{\mathcal{R}_{ij}} = \log(\hat{\mathcal{R}}_j^T \hat{\mathcal{R}}_i R_i^T R_j), \quad (19)$$

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

To better demonstrate the effectiveness of (17), we further make an explanation for our cost function, and draw it as a toy example in Fig.3. In Eq. (17), the first term corresponds to the reprojection error, the second term is used to penalize the large RPE that is caused by pose drift. Note that the reprojection error may remain small even if camera poses are drifting. This can be seen from Fig.3 (a): Suppose  $C_0$  is correctly registered at first, and  $C_1, C_2, C_3$  are registered sequentially. When  $C_1$  is wrongly registered, the error will be passed to  $C_2$ , then  $C_3$ . But traditional BA can not optimise camera poses to the correct place, because the triangulated 3D points are geometrically coherent with the camera poses, and reprojection errors are small (measured by the red and green dots in Fig.3). However, in this situation, the known rotation term can measure the discrepancy between the averaged rotations and incrementally recovered rotations (See from Fig.3 (b)). And our optimiser tries to minimise this discrepancy and thereby alleviate pose drift.

## 6. Experimental Results

Our experiments aim at demonstrating the accuracy, efficiency, and robustness of the proposed methods. We implement Levenberg-Marquardt (LM) [30], row-by-row block coordinate descent (RBR-BCD) [17], and our hybrid rotation averaging in C++. Besides, the implementations of SE-Sync [33] and Shonan [16] are provided by the authors and publicly available. For HSfM [13] and LUD [31], we use [9] as the rotation averaging solver, and the Ceres solver [1] for bundle adjustment. All approaches are tested on a laptop with a 2.7 GHz CPU and 8GB RAM.

### 6.1. Evaluation of Hybrid Rotation Averaging 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

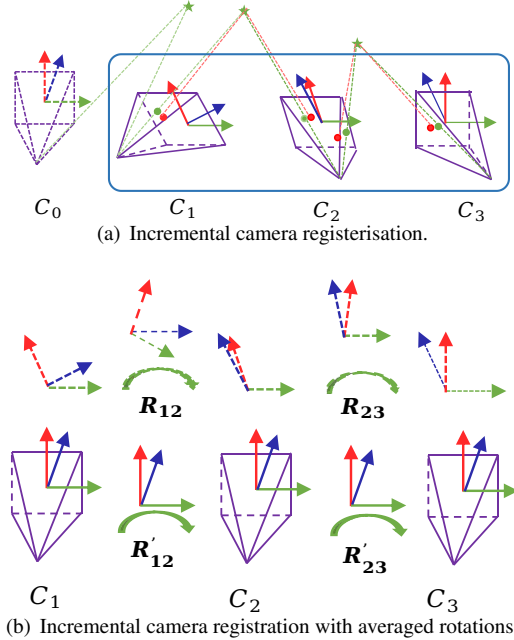


Figure 3. A toy example to explain our RA-SfM. Camera rotations are drawn by three arrows with colors in red, green, and blue. In (a), landmarks are denoted by  $\star$ , green dots in the image plane represent keypoints, and red dots are reprojected coordinates. In (b), we show the correctly registered camera poses for  $C_1, C_2, C_3$ .  $R_{12}$  and  $R_{23}$  denote the relative rotations, which are obtained from (a).  $R'_{12}$  and  $R'_{23}$  denote the relative rotations obtained from averaged rotations.

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 by random angular rotations about randomly selected axes. The perturbation angles are normally distributed with 0 mean and 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 [30], RBR-BCD [17], Shonan [16] and SE-Sync [33]. 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 the camera's number is below 500, the hybrid rotation averaging 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 4 times slower than SE-Sync, and  $2 \sim 3$  orders of magnitude slower than ours, when the number of the cameras goes above 5000. 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

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

$n$	#edges	$\sigma$	LM [30]		RBR-BCD [17]		Shonan [16]		SE-Sync [33]		hybrid RA	
			$\bar{R}$	$time(s)$	$\bar{R}$	$time(s)$	$\bar{R}$	$time(s)$	$\bar{R}$	$time(s)$	$\bar{R}$	$time(s)$
20	30	0.2	5.201e-05	0.001	1.219e-05	0.039	9.803e-06	0.032	8.975e-06	< <b>1e-06</b>	<b>7.307e-06</b>	< <b>1e-06</b>
		0.5	1.492e-01	<b>0.002</b>	7.200e-02	0.028	1.955e-01	0.038	2.033e-01	0.003	<b>1.550e-01</b>	<b>0.002</b>
100	300	0.2	9.490e-06	0.095	6.351e-06	0.651	<b>5.374e-06</b>	0.144	5.383e-06	<b>0.005</b>	5.938e-06	0.006
		0.5	8.771e-01	0.089	1.160e-01	0.813	1.108e-01	0.078	1.336e-01	<b>0.079</b>	<b>9.400e-02</b>	0.189
500	1000	0.2	5.413e-06	0.845	5.381e-06	208.372	5.433e-06	0.886	5.209e-06	0.403	<b>5.184e-06</b>	<b>0.159</b>
		0.5	7.351e-01	0.781	1.130e-01	245.818	1.255e-01	0.533	<b>9.417e-02</b>	<b>0.255</b>	1.080e-01	0.270
1000	4000	0.2	1.800e-01	1.213	<b>6.754e-06</b>	2,274	9.835e-06	1.883	8.739e-06	0.821	1.021e-05	<b>0.127</b>
		0.5	8.956e-01	1.372	1.120e-01	2,153	8.974e-02	1.141	1.310e-01	0.985	<b>8.200e-02</b>	<b>0.949</b>
5000	20000	0.2	1.260e-01	4.414	-	-	8.341e-06	14.870	<b>6.371e-06</b>	9.083	7.159e-06	<b>0.331</b>
		0.5	2.787e-01	5.183	-	-	1.516e-01	12.338	1.408e-01	3.699	<b>8.200e-02</b>	<b>0.809</b>
10000	40000	0.2	1.410e-01	23.714	-	-	1.838e-05	45.680	9.037e-06	10.335	<b>7.884e-06</b>	<b>0.362</b>
		0.5	3.240e-01	27.265	-	-	1.209e-01	42.627	1.386e-01	11.128	<b>9.100e-02</b>	<b>1.704</b>
50000	200000	0.2	-	-	-	-	6.013e-06	956.821	-	-	<b>6.310e-06</b>	<b>0.515</b>
		0.5	-	-	-	-	1.933e-01	905.294	-	-	<b>7.500e-02</b>	<b>7.124</b>

corresponding cells in the table by “-”. LM and SE-Sync failed when the camera number reaches 50000, as there is insufficient memory to perform the CHOLMOD [10] factorisation. As our approach only needs to compute the SVD of a small block matrix and evaluate  $d$  matrix operations of  $3 \times 3$  matrices in each iteration, we can solve all the evaluated large-scale datasets.

In terms of accuracy, LM achieves the global optimum with certain probability (30% – 70% as reported in [17]), 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.

## 6.2. Evaluation of RA-SfM 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 [34], global [31] and hybrid [13] SfM approaches. Since the quasi-convex SfM approach [48] is sensitive to outliers and extremely slow in such datasets, we did not evaluate it in our experiment.

Figure 4 shows the reconstruction results of COLMAP and our RA-SfM on the Campus [15] dataset. This dataset, which has a loop, mainly contains plants that can produce lots of wrong matching results. COLMAP [34] 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 can further constrain camera poses after the initial registration.

We also evaluated our approach on the online datasets from [45], which are collections of challenging unordered images. The datasets contain many wrong epipolar geometries due to the extreme viewpoint, scale, and illumination changes. The runtime and accuracy results are shown in

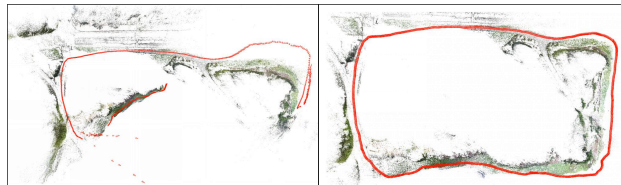


Figure 4. Reconstruction results for the Campus dataset [15]. Left: COLMAP [34], Right: Our RA-SfM.

Table 2. As is observed, COLMAP [34] recovers the most camera poses in most online datasets. However, our method has the lowest mean reprojection error (MRE) in most of the online datasets, which indicates RA-SfM is more robust and accurate than COLMAP. For our RA-SfM, the time for rotation averaging is separately given in the penultimate column (denoted as  $T_{\bar{R}}$ ). While LUD [31] is the most efficient one among the evaluated methods, it has large MRE, and the number of recovered camera poses is less than ours and COLMAP. HSfM [13] is faster than COLMAP and RA-SfM because it only samples 2 correspondences to compute the camera centers in each RANSAC iteration. Besides, HSfM [13] recovers the fewest camera poses and fails to recover the correct camera centers.

Some visual results for online datasets are shown in Fig. 5. For each subfigure, the top and bottom images are the results obtained by COLMAP [34] and our RA-SfM, respectively. For the Ellis Island dataset, we showed two different parts in the first two columns of Fig. 5, where the red rectangle area shows the comparison result. For the Gendarmenmarkt dataset, the reconstruction result of COLMAP is bad on the left part, which indicates the wrong camera poses. For the Vienna Cathedral dataset, though COLMAP recovers more camera poses than ours, our approach reconstructed more scene details than COLMAP, as is indicated

Table 2. Comparison of runtime and accuracy on online datasets [45].  $N_i$  and  $N_c$  denotes the number of images and registered cameras respectively,  $N_p$  denotes the reconstructed 3D landmarks, MRE means mean reprojection error in pixel.  $T$  and  $T_R$  denotes the total reconstruction time and hybrid rotation averaging time respectively (in seconds). The best MREs are marked in bold font.

Dataset	$N_i$	COLMAP [34]				LUD [31]				HSfM [13]				RA-SfM				
		$N_c$	$N_p$	MRE	$T(s)$	$N_c$	$N_p$	MRE	$T(s)$	$N_c$	$N_p$	MRE	$T(s)$	$N_c$	$N_p$	MRE	$T_R(s)$	$T(s)$
Alamo	2,915	906	138K	0.69	3,180	578	146K	1.28	260	522	149K	1.62	1,079	895	141K	<b>0.65</b>	0.340	2,771
Ellis Island	2,587	801	154K	0.73	4,307	234	16K	1.54	24	208	34K	2.53	169	727	146K	<b>0.72</b>	2.290	3,920
Gendarmenmarkt	1,463	1,040	209K	0.71	3,737	705	87K	1.51	104	542	74K	1.94	377	1,023	202K	<b>0.70</b>	1.997	3,931
Madrid Metropolis	1,344	460	60K	0.62	1,320	350	51K	1.08	36	292	51K	1.48	221	438	66K	<b>0.59</b>	0.420	1,417
Montreal N.D.	2,298	554	107K	<b>0.67</b>	1,902	462	166K	1.64	194	418	155K	1.95	1041	528	105K	0.68	0.115	1,423
Notre Dame	1,431	1,408	349K	0.76	22,788	550	262K	2.06	259	526	281K	2.30	2,375	1,409	353K	<b>0.75</b>	0.131	19,943
NYC Library	2,550	556	101K	0.72	1,698	336	70K	1.52	75	282	74K	1.99	356	519	100K	<b>0.65</b>	0.515	1,715
Piazza del Popolo	2,251	1,011	122K	0.68	2,676	329	38K	1.65	62	286	35K	1.92	212	966	122K	<b>0.66</b>	0.360	3,258
Piccadilly	7,351	3,129	362K	<b>0.73</b>	16,590	2,301	202K	1.83	262	1,665	185K	2.09	2,169	3,041	363K	0.80	1.422	15,109
Roman Forum	2,364	1,594	284K	<b>0.71</b>	5,388	1,045	256K	1.71	182	1,071	262K	1.93	2,237	1,460	267K	0.77	1.938	5,408
Tower of London	1,576	707	140K	0.61	2,767	485	140K	1.65	95	398	149K	1.91	816	672	139K	<b>0.58</b>	0.800	1,979
Trafalgar	15,685	6,980	581K	0.81	14,790	5,044	378K	1.56	713	3,446	318K	1.95	5,761	7,085	597K	<b>0.72</b>	4.213	14,831
Union Square	5,961	937	69K	0.66	2,604	803	41K	1.65	107	769	38K	1.88	1,763	809	57K	<b>0.52</b>	1.304	1,962
Vienna Cathedral	6,288	1,185	290K	0.74	9,714	849	203K	1.91	173	662	252K	2.36	2,307	1,173	303K	<b>0.71</b>	1.959	16,111
Yorkminster	3,368	1,022	259K	0.71	10,806	421	132K	1.75	135	417	129K	1.93	1,487	614	183K	<b>0.64</b>	3.183	9,299

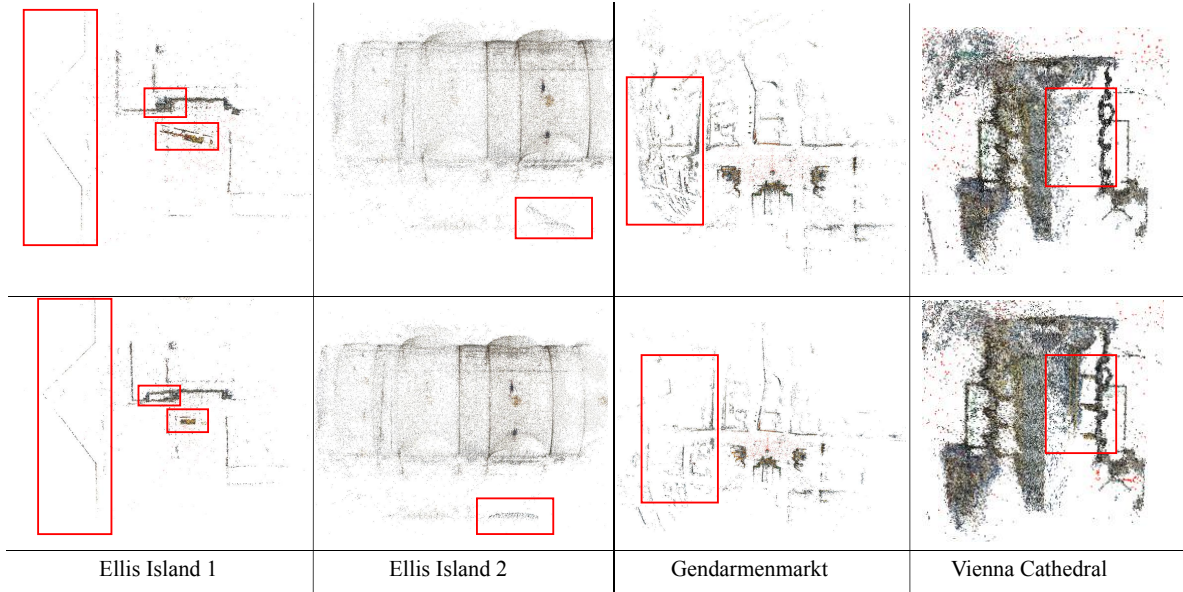


Figure 5. Visual reconstruction results for some of the online datasets [45]. For each subfigure, the top and bottom images are respectively the results obtained from COLMAP [34] and our RA-SfM. (The first two columns are results of two parts of the Ellis Island dataset.)

by the red rectangle. From Table 2 and Fig. 5, we demonstrate our RA-SfM can effectively correct the wrongly registered camera poses in the incremental SfM pipeline, and also achieves the state-of-the-art robustness and accuracy.

## 7. Conclusion

This paper presents a hybrid rotation averaging method that is robust to outliers. We combine fast view graph filtering to increase graph sparsity with state-of-the-art implementations of both global and local optimization methods. The exposition is rounded off by a soft embedding into an incremental SfM pipeline leading to accurate, reliable, and

highly efficient results. However, our method solves the rotation averaging problem all in once, thus it can also meet memory limitation in larger scenes. In our future work, we are interested in extending this work in larger scenes with a more efficient manner.

**Acknowledgement:** We sincerely thank Prof. Frank Daelart and Jing Wu for the discussion of the experimental details, and also for their help in improving the experimental results. L. Kneip furthermore acknowledges the support of the Natural Science Foundation of Shanghai (grant number 19ZR1434000), and his affiliation with the Shanghai Engineering Research Center of Intelligent Vision and Imaging.



## References

- [1] Sameer Agarwal and Keir Mierle. Ceres solver. <http://ceres-solver.org>. 6
- [2] 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
- [3] Nicolas Boumal. A riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints. *CoRR*, abs/1506.00575, 2015. 2, 4
- [4] 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
- [5] Stephen P. Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2014. 2, 4
- [6] 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
- [7] 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
- [8] 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, 3, 5
- [9] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):958–972, 2018. 2, 3, 6
- [10] Yanqing Chen, Timothy A. Davis, William W. Hager, and Sivasankaran Rajamanickam. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw.*, 35(3):22:1–22:14, 2008. 7
- [11] 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
- [12] 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
- [13] 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
- [14] 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
- [15] 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
- [16] 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, 4, 6, 7
- [17] 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
- [18] 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
- [19] 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. 5
- [20] 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
- [21] 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
- [22] 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, 3
- [23] 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, 3
- [24] Richard I. Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, 2013. 3
- [25] 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
- [26] 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. 5
- [27] Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *J. Sci. Comput.*, 58(2):431–449, 2014. 4
- [28] Yi Ma, Stefano Soatto, Jana Košecá, and S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Science & Business Media, 2012. 3
- [29] 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
- [30] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999. 6, 7

- [31] 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
- [32] Jaehyun Park and Stephen Boyd. General heuristics for non-convex quadratically constrained quadratic programming. *arXiv:1703.07870*, 2017. 5
- [33] 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, 4, 6, 7
- [34] 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
- [35] 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
- [36] 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
- [37] Yulun Tian, Kasra Khosoussi, and Jonathan P. How. Block-coordinate minimization for large SDPs with block-diagonal constraints. *arXiv: 1903.00597*, 2019. 2, 3, 4
- [38] Yulun Tian, Kasra Khosoussi, David M. Rosen, and Jonathan P. How. Distributed certifiably correct pose-graph optimization. *arXiv: 1911.03721*, 2019. 2, 4
- [39] 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
- [40] 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
- [41] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference*, 2(2):145–194, 2013. 2
- [42] Po-Wei Wang, Wei-Cheng Chang, and J. Zico Kolter. The mixing method: coordinate descent for low-rank semidefinite programming. *arXiv: 1706.00476*, 2017. 2, 4
- [43] Zaiwen Wen, Donald Goldfarb, Shiqian Ma, and Katya Scheinberg. Row by row methods for semidefinite programming. Technical report, 2009. 2
- [44] 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
- [45] Kyle Wilson and Noah Snavely. Robust global translations with ldsfm. In *Computer Vision European Conference*, pages 61–75, 2014. 1, 7, 8
- [46] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision*, pages 127–134, 2013. 2
- [47] 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
- [48] Qiangong 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
- [49] 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
- [50] 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
- [51] 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