This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2024.3416057

Cayley Rotation Averaging: Multiple Camera Averaging Under the Cayley Framework

Qiulei Dong, Shuang Deng, Yuzhen Liu

***Abstract*—Rotation averaging, which aims to calculate the absolute rotations of a set of cameras from a redundant set of their relative rotations, is an important and challenging topic arising in the study of structure from motion. A central problem in rotation averaging is how to alleviate the influence of noise and outliers. Addressing this problem, we investigate rotation averaging under the Cayley framework in this paper, inspired by the extra-constraint-free nature of the Cayley rotation representation. Firstly, for the relative rotation of an arbitrary pair of cameras regardless of whether it is corrupted by noise/outliers or not, a general Cayley rotation constraint equation is derived for reflecting the relationship between this relative rotation and the absolute rotations of the two cameras, according to the Cayley rotation representation. Then based on such a set of Cayley rotation constraint equations, a Cayley-based approach for Rotation Averaging is proposed, called CRA, where an adaptive regularizer is designed for further alleviating the influence of outliers. Finally, a unified iterative algorithm for minimizing some commonly-used loss functions is proposed under this approach. Experimental results on 16 real-world datasets and multiple synthetic datasets demonstrate that the proposed CRA approach achieves a better accuracy in comparison to several typical rotation averaging approaches in most cases.**

I. INTRODUCTION

Structure from Motion (SfM) aims to recover the 3D structure of a scene by computing the camera motion from an input set of images, which is an important problem for 3D reconstruction [1], [2], [3], [4], [5] and has many potential applications in the fields of computer vision and robotics. In recent years, a category of two-stage SfM methods, which computes all the cameras' global rotations at first and then computes their global locations with these obtained global rotations, has attracted more and more attention [6], [7], [8], [9], [10], [11], [12], [13], [14].

The aforementioned two-stage SFM methods have to confront the following fundamental problem, i.e. the **rotation averaging** problem: how to calculate the absolute rotations $R_i (i = 1, 2, ..., n)$ of n cameras from a given set of m relative rotations R_{ij} among these cameras. In the noiseless case, the absolute and relative rotations satisfy the *basic rotation constraint equation* or its variant:

$$\begin{aligned} &R_{ij}R_i = R_j \text{ or } R_{ij} = R_j R_i^{-1}\ , \\ &\text{s.t. } \ R_i^T R_i = I_3,\ |R_i| = 1,\ i = 1, 2, ..., n \end{aligned} \tag{1}$$

Qiulei Dong, Shuang Deng, and Yuzhen Liu are with (i) State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, China, (ii) University of Chinese Academy of Sciences, China, and (iii) Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, China.

where I_3 is the 3-order identity matrix, and $|\bullet|$ represents the determinant of a matrix.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be an epipolar geometry graph for representing the relationships among the $|\mathcal{V}| = n$ cameras with the given $|\mathcal{E}| = m$ relative rotations, where each vertex in $\mathcal{V}$ represents a camera and each edge $(i, j) \in \mathcal{E}$ implies that the relative rotation between cameras i and j has been given (without loss of generality, we assume that $i < j$ in this paper). Let $\mathrm{SO}(3)$ be the 3D special orthogonal group, i.e., the set of 3×3 rotation matrices that are orthogonal matrices with determinant 1. Then due to the fact that the input relative rotations (which could be obtained by some relative rotation estimation methods [15], [16], [17], [18]) are generally inaccurate, according to the basic rotation constraint equation (1), many rotation averaging methods have been proposed for estimating the absolute rotations by minimizing the following unified-form distance metric:

$$\min_{\{R_i\}_{i=1}^n \in \mathrm{SO}(3)} \sum_{(i,j)\in\mathcal{E}} \mathcal{F}(R_{ij}R_i, R_j), \tag{2}$$

where $\mathcal{F}(\cdot, \cdot)$ represents an arbitrary distance metric.

In general, there are two main rotation averaging techniques: incremental rotation averaging [19], [20], [21] and non-incremental rotation averaging [22], [23], [24], [25], [26]. The incremental rotation averaging technique firstly builds an epipolar geometry sub-graph of an initial triplet of cameras, and then iteratively adds the other cameras to the epipolar geometry sub-graph one (or multiple) at a time. The non-incremental rotation averaging technique computes the absolute rotations of all the cameras together as a whole. In fact, the non-incremental rotation averaging technique is the basis of the incremental rotation averaging technique, and the incremental rotation averaging technique could be considered as such an iterative process where the non-incremental rotation averaging technique is implemented on an epipolar geometry sub-graph repeatedly at each iteration.

In this work, the non-incremental rotation averaging problem is investigated, and the related works on this problem are introduced in the following subsection.

A. Related Works

According to whether pre-processing operations are used or not, the existing non-incremental rotation averaging methods in literature can be roughly divided into two categories as follows.

The first category generally employs the following two-stage strategy: (i) At the pre-processing stage, considering that

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2024.3416057

the input set of relative rotations is inevitably infiltrated by some outliers in real scenarios, an outlier filtering operation is implemented for detecting and removing the outliers, resulting in a relatively reliable subset of relative rotations; (ii) At the optimization stage, the obtained subset of relative rotations are used for estimating the absolute rotations by minimizing the objective function (2) or its variants [11], [12], [13], [27], [28]. For example, Zach et al. [28] collected statistics of the residual transformation over many overlapping loops, and then used them to detect outliers in the input set of relative rotations for further rotation averaging. Purkait et al. [11] employed a view-graph cleaning network for detecting outliers and rectifying noisy measurements, and then introduced a fine-tuning network for rotation averaging. Melekhov et al. [12] proposed a hybrid method for rotation averaging by combining a global solver and a local solver, where an additional view graph filtering approach was introduced to remove outliers as a pre-processing step. It has to be pointed out that the preprocessing techniques used in the above methods could not guarantee to identify and remove all the outliers among the given relative rotations in theory, hence, some rotation averaging methods (e.g., [12], [13]) have to utilize some specific optimization techniques (e.g., iterative reweighting) to further alleviate the influence of the remaining outliers at their optimization stage.

Unlike the above two-stage methods, the second category of rotation averaging methods employs a single-stage strategy, which straightforward estimates the absolute rotations from the input relative rotations. Crandall et al. [6] proposed a rotation averaging method (called DISCO) using a combination of discrete-continuous optimization, where extra prior terms were added into the unified formulation (2). Hartley et al. [22] used an L_1-norm-based objective function for alleviating the influence of outliers, and proposed an iterative method based on the Weiszfeld algorithm for rotation averaging. Govindu [29] proposed a Lie-Algebraic rotation averaging method, where the axis-angle form of 3D rotation was used for representing the basic rotation constraint equation (1). Then, Chatterjee and Govindu [25] extended the Lie-Algebraic rotation averaging method [29] for pursuing a higher accuracy of rotation estimation. They firstly introduced an L_1-norm-based solver for averaging relative rotations according to the Lie-Algebraic method [29], and then proposed an iterative method by minimizing a Huber-like cost function, where the L_1 solution was used as an initial guess. Furthermore, they proposed a generalized computational framework of rotation averaging [26], where different loss functions could be utilized. They found that under this framework, the $L_{1/2}$-norm loss function performs better than the other comparative loss functions(here, their method with the $L_{1/2}$-norm loss function is denoted as $\text{FRRA}_{1/2}$). Based on the spectral graph theory, Eriksson et al. [30] provided a theoretical analysis of Lagrangian duality in rotation averaging, and proposed a first-order algorithm for semidefinite cone programming. Dellaert et al. [31] proposed the Shonan rotation averaging method, which could guarantee globally optimal solutions under mild conditions on the measurement noise. Yang and Carlone [32] proposed a general framework to explore certifiable algorithms for geometric perception where outliers are involved, and accordingly, the explored algorithms could be applied to rotation averaging. Arrigoni et al. [33] investigated the synchronization problem[34] that was formulated as a low-rank and sparse matrix decomposition problem in the presence of missing data, outliers, and noise. And they proposed a minimization strategy, called R-GoDec, which could deal with the rotation averaging task. It has to be pointed out that the single-stage methods do not oppose to the preprocessing techniques used in the two-stage methods, and they could be straightforwardly integrated with an additional preprocessing technique to form new two-stage methods.

B. Motivation and Contributions

Despite the rapid progress in rotation averaging as discussed above, it is noted that the estimation accuracies of the existing methods on some challenging datasets where severe noise and outliers are involved, are still quite low as reported in many existing works [12], [26].

Addressing this problem, we investigate how to alleviate the influence of noise and outliers on rotation averaging in this work. Considering that some outliers in the input data inevitably remain regardless of whether additional preprocessing techniques are used or not, we aim to explore a new single-stage rotation averaging method from the following two points of view, motivated by several basic observations in the optimization community:

1 It is observed that (i) for the input data that is corrupted by noise and outliers, when the iterative optimization approaches could not be guaranteed to converge to a globally optimal solution, two different parameterizations of the same loss function might lead to result with different numerical inaccuracies; (ii) Cayley transformation is a mapping between skew-symmetric matrices and rotation matrices [35], which has shown its numerical stability to noise and outliers for handling the camera self-calibration problem [36], [37]. The Cayley rotation representation, which is obtained by transforming a 3×3 rotation matrix under the Cayley transformation, is an extra-constraint-free 3D vector whose elements are independent from each other, and any Cayley rotation representation is mathematically equivalent to its corresponding rotation-matrix representation. Motivated by the two observations, our first goal is to construct a new rotation constraint with the Cayley representation, which is mathematically equivalent to the original one (i.e., Eq. (1)), but is easy to solve and more robust to noise and outliers.

2 It is observed that for an optimization problem where some outliers are involved, if some additional attentions could be paid to the outliers adaptively during the optimization process, an improved solution could be generally obtained. Motivated by this observation, our second goal is to design an adaptive mechanism to alleviate the negative influence of outliers during the optimization process.

By combining the aforementioned two goals, we propose a single-stage computational approach for rotation averaging under the Cayley transformation in this paper. We firstly derive a general Cayley rotation constraint equation according to the Cayley rotation representation. This general constraint equation is mathematically equivalent to the traditional matrix-form rotation constraint equation (1), but it does not require both the orthogonal constraint and matrix determinant constraint that are indispensable for the matrix-form rotation constraint equation. For an input set of relative rotations, a set of general Cayley rotation constraint equations could be obtained accordingly. Then, by utilizing the obtained constraint equations, a Cayley-based approach for Rotation Averaging is explored, called CRA. In the proposed CRA approach, an adaptive regularizer is designed to weaken such Cayley constraint equations that are corrupted by outliers. Finally, we explore a unified algorithm for solving some commonly-used objective functions under the proposed CRA in an iterative optimization manner.

In sum, the main contributions in this work include:

- We propose the CRA approach for rotation averaging, based on the general Cayley rotation constraint equation. To the best of our knowledge, this work is the first attempt to investigate the rotation averaging problem under the Cayley framework.
- By embedding different loss functions into the proposed CRA approach, different rotation averaging methods could be straightforwardly derived, and a unified iterative algorithm is presented for solving them.
- The effectiveness of the proposed CRA approach is demonstrated by the experimental results on 16 public datasets and 3 synthetic datasets in Section IV .

The remainder of this work is organized as follows: Section II proposes the Cayley-based approach CRA for rotation averaging. Section III presents the iterative algorithm for optimizing the derived methods from CRA in detail. Extensive experimental results are reported in Section IV, and followed by some concluding remarks in Section V.

II. Cayley Rotation Averaging

In this section, we elaborate the Cayley-based approach for rotation averaging. Firstly, some preliminary knowledge on the Cayley transformation and the Cayley rotation representation is introduced. Then, we describe the general Cayley rotation constraint equation and the proposed approach in detail.

A. Cayley Transformation and Rotation Representation

Considering that the Cayley transformation was used infrequently in computer vision and the proposed Cayley-based approach for rotation is dependent on the Cayley rotation representation, some basic knowledge of the Cayley transformation and the Cayley rotation representation is introduced here, and more details could be found in [35], [36], [37].

Assume a matrix $X \in \mathcal{R}^{v\times v}$ is subject to $|I_v + X| \neq 0$, where '$|\bullet|$' represents the determinant of a matrix and I_v is the v-order identity matrix, then the following function $\psi(\bullet)$ is called the *Cayley transformation* that transforms the matrix X into another matrix $Y \in \mathcal{R}^{v\times v}$:

$$Y = \psi(X) = (I_v - X)(I_v + X)^{-1} = (I_v + X)^{-1}(I_v - X). \tag{3}$$

In addition, it is noted that if Y is the Cayley transformation matrix of X (i.e. $Y = \psi(X)$), then according to (3), it holds:

$$X = \psi(Y) = (I_v - Y)(I_v + Y)^{-1} = (I_v + Y)^{-1}(I_v - Y). \tag{4}$$

This means that X is also the Cayley transformation matrix of Y.

Let $c = [c_x, c_y, c_z]^T \in \mathcal{R}^{3\times 1}$ be a 3D vector, and $[c]_\times \in \mathcal{R}^{3\times 3}$ be the skew-symmetric matrix derived from c, i.e.

$$[c]_\times = \begin{pmatrix} 0 & -c_z & c_y \\ c_z & 0 & -c_x \\ -c_y & c_x & 0 \end{pmatrix}. \tag{5}$$

Let $\mathrm{C}(3)$ be the set of all the skew-symmetric matrices, and $\mathrm{SO}_\pi(3)$ be the set of 3×3 rotation matrices whose rotation angles equal to π. The Cayley transformation of an arbitrary rotation matrix $R \in \mathrm{SO}(3)\backslash\mathrm{SO}_\pi(3)$ subject to $|I_3 + R| \neq 0$ (i.e. the rotation angle of R is not equal to π. In this paper, we assume this condition always holds. In fact, it also holds in most real-world datasets), is a 3×3 skew-symmetric matrix [37] as:

$$[c]_\times = \psi(R) \quad \text{and} \quad R = \psi([c]_\times). \tag{6}$$

The 3D vector c is called the *Cayley rotation representation* of a rotation matrix R.

It has to be pointed out that the Cayley rotation representation is one of various parametrization representations (e.g., rotation matrix, angle-axis representation, Rodrigues representation, etc.) for representing 3D rotations. Mathematically, the Cayley transform defines the same map as the gnomonic projection[24]. However, by comparing with the Rodrigues representation derived via the gnomonic projection, the Cayley rotation representation is more concise. By comparing with the 3×3 rotation matrix R, where both the orthogonal constraint and the matrix determinant constraint $|R| = 1$ have to be imposed, the 3×1 Cayley representation does not introduce any extra constraint.

According to (3) and (6), a rotation matrix R could be further represented by its Cayley rotation representation c as:

$$R = (I_3 - [c]_\times)(I_3 + [c]_\times)^{-1} = \frac{(1 - c^T c)I_3 - 2[c]_\times + 2cc^T}{1 + c^T c}. \tag{7}$$

The detailed derivation of Eq. (7) could be found in Appendix A1.

B. General Cayley Rotation Constraint

Given an input set of m relative rotations (corresponding to n cameras), an epipolar geometry graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ could be constructed as described in Section I. Without loss of generality, let c_{ij} be the Cayley representation of the input relative rotation R_{ij} in the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, and c_i be the Cayley representation of the unknown absolute rotation $R_i (i = 1, 2, ..., n)$.

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2024.3416057

By substituting the matrix-form representations $\{R_i, R_j, R_{ij}\}$ in Eq. (1) with their corresponding Cayley representations according to Eq. (7), the basic matrix-form rotation constraint equation (i.e., Eq. (1)) could be re-formulated as the following *basic Cayley rotation constraint equation* via matrix/vector computation and algebraic reduction (Detailed derivation could be found in Appendix A2):

$$c_{ij} = ([c_{ij}]_\times - I_3)c_i + (1 - c_{ij}^T c_i)c_j,\ \forall (i,j)\ \in \mathcal{E}\ . \quad (8)$$

As is seen, the basic Cayley rotation constraint equation (8) is mathematically equivalent to the original rotation equation (1) without any information loss.

Moreover, it is noted that there exist more or less outliers and noise in real data, and the basic Cayley rotation constraint equation (8) does not hold strictly in this case. Hence, a residual e_{ij}, representing the difference between the two sides of (8), is introduced for constructing the *general Cayley rotation constraint equation* as:

$$e_{ij} = ([c_{ij}]_\times - I_3)c_i + (1 - c_{ij}^T c_i)c_j - c_{ij}\ \ . \quad (9)$$

C. Formulation of Cayley Rotation Averaging

Our goal is to compute the $n = |\mathcal{V}|$ absolute Cayley rotations $\{c_i\}_{i=1}^n$ with the given $m = |\mathcal{E}|$ relative Cayley rotations c_{ij} in the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Here, without loss of generality, the absolute Cayley rotation c_1 of the first camera is fixed as $c_1 = [0,0,0]^T$ for removing the rotation ambiguity.

Let $\mathbf{c} = [c_2; c_3; ...; c_n] \in \mathcal{R}^{(3n-3)\times 1}$ be the vector consisting of the Cayley representations of all the absolute rotations except the first one. According to (9), the following objective function that is an instantiation of Eq. (2) could be straightforwardly obtained by minimizing the sum of the residuals e_{ij} defined in (9):

$$\begin{aligned} \min_{\mathbf{c}} \mathcal{F}(\mathbf{c}) &= \sum_{(i,j)\in\mathcal{E}} f(e_{ij}) \\ &= \sum_{(i,j)\in\mathcal{E}} f\bigg(([c_{ij}]_\times - I_3)c_i + (1 - c_{ij}^T c_i)c_j - c_{ij}\bigg) \\ \text{s.t. } & c_1 = [0,0,0]^T \end{aligned} \quad (10)$$

where $f(\cdot)$ indicates an arbitrary loss function. Eq. (10) provides a general objective form but not a special distance metric, where different distance metrics could be used as the metric function $f(\cdot)$, such as the commonly used L_2, L_1, $L_{1/2}$ distances.

It is noted that Eq. (10) consists of a lot of measurements, and each measurement is used to constrain the residual of a specific relative rotation. The measurements corresponding to the outliers (unknown in advance) would generally deteriorate the estimation of absolute rotations. In other words, in the case of outliers, the solution obtained by minimizing (10) might seriously deviate from the correct absolute rotations. Addressing this problem, an adaptive weighting mechanism is explored to dynamically assign a 0/1 weight $w_{ij} \in \{0,1\}$ to each measurement in (10) during the optimization process: $w_{ij} = 0$ indicates that the corresponding measurement is corrupted by outliers and this measurement is unavailable, while $w_{ij} = 1$ indicates that the corresponding measurement is not corrupted by outliers. Let $\mathbf{w} = [w_{ij}] \in \{0,1\}^{m\times 1}$ be the m-dimensional weight vector, and its element w_{ij} corresponds to the Cayley rotation measurement with regard to c_{ij} (i.e., R_{ij}). Then, a sparsity regularizer $Reg_s(\mathbf{w}) = ||1 - \mathbf{w}||_0$ on the vector $\mathbf{1} - \mathbf{w}$, where '$||\bullet||_0$' indicates the L_0 norm and $\mathbf{1}$ is an all-one vector, is added into (10) for both avoiding a degenerated all-zero solution and resisting outliers. Accordingly, the Cayley-based approach for rotation averaging (CRA) is derived from (10) as:

$$\begin{aligned} & \min_{\mathbf{c},\mathbf{w}} \sum_{(i,j)\in\mathcal{E}} w_{ij} f(e_{ij}) + \beta Reg_s(\mathbf{w}) \\ \Rightarrow & \min_{\mathbf{c},\mathbf{w}} \sum_{(i,j)\in\mathcal{E}} w_{ij} f(e_{ij}) + \beta \sum_{(i,j)\in\mathcal{E}} ||1 - w_{ij}||_0 \\ \Rightarrow & \min_{\mathbf{c},\mathbf{w}} \sum_{(i,j)\in\mathcal{E}} w_{ij} f\bigg(([c_{ij}]_\times - I_3)c_i + (1 - c_{ij}^T c_i)c_j - c_{ij}\bigg) \\ & \qquad + \beta||1 - \mathbf{w}||_0 \\ \text{s.t. } & c_1 = [0,0,0]^T, \mathbf{w} \in \{0,1\}^{m\times 1} \end{aligned} \quad (11)$$

where β is a given weight. It has to be explained that each two-term measurement "$\mathcal{F}_{ij}^r = w_{ij} f(e_{ij}) + \beta||1 - w_{ij}||_0$ " in Eq. (11), which contains a basic term and a weighted outlier regularizer, is equivalent to the following single-term measurement without a weight as indicated in [38]:

$$\mathcal{F}_{ij}^r = \begin{cases} \beta & \text{if} f(e_{ij}) \geq \beta \\ f(e_{ij}) & \text{else} \end{cases} \quad (12)$$

However, the proposed two-term measurement could reflect our design motivation for resisting outliers more intuitively in comparison to its single-term counterpart.

III. Algorithm for CRA

Note that different loss functions could be used for restraining the first term of (11) under the proposed CRA. In this section, we propose a unified iterative algorithm for solving three commonly-used loss functions (including the L_2, L_1, and $L_{1/2}$ losses) under CRA as defined in the second column of Table I. Then, the convergence criterion of the proposed algorithm is given.

A. Algorithm

As seen from (11), this optimization problem is a non-linear problem with multiple variables. It could be solved via various optimization techniques in principle, such as the Augmented Lagrange Multiplier(ALM)-based method[39], [40], the Graduated Non-Convexity method[41], etc. Here, we explore an ALM-based algorithm for solving (11) as follows:

For computational convenience, we introduce auxiliary variables d_{ij} to replace $1 - c_{ij}^T c_i$. Since $c_1 = [0,0,0]^T$, we have $d_{1j} = 1$ for all $(1,j) \in \mathcal{E}$. Let k be the number of d_{1j} (i.e. the edges connecting the first camera node) in $\mathcal{E}$, and $\mathbf{d} = [d_{ij}]_{i\neq 1} \in \mathcal{R}^{(m-k)\times 1}$ where $m = |\mathcal{E}|$ is the number of all the given relative rotations. Let $\mathbf{e} = [e_{ij}] \in \mathcal{R}^{3m\times 1}$ be the vector consisting of the residuals corresponding to all the

TABLE I
LOSS FUNCTIONS AND THE CORRESPONDING SOLUTIONS TO $\mathbf{e}^{t+1}$ TO (22): './' INDICATES THE ELEMENTWISE DIVISION, 'sgn(·)' REPRESNETS THE SIGN FUNCTION.

	Loss Function $f(x)$	Optimal solution $\mathbf{e}^{t+1}$ to (22)
L_2	x^2	$(Q\mathbf{c}^t + F\mathbf{c}^t + P - \frac{\Upsilon^{\mathbf{e},t}}{\eta^t})./(\frac{2W}{\eta^t}+1)$
L_1	$\|x\|$	$\text{sgn}(Q\mathbf{c}^t + F\mathbf{c}^t + P - \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}) \cdot \max(\|Q\mathbf{c}^t + F\mathbf{c}^t + P - \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}\| - \frac{W}{\eta^t}, 0)$
$L_{1/2}$	$\|x\|^{1/2}$	$e_i^{t+1} = \begin{cases} B_i & \text{if } 4\eta^t\sqrt{(\frac{\|A_i\|}{3})^3} > W_i \text{ and } W_i\sqrt{\|B_i\|} + \frac{\eta^t}{2}(B_i - A_i)^2 < \frac{\eta^t}{2}A_i^2 \\ 0 & \text{otherwise} \end{cases}$ where e_i^{t+1} and A_i are the i-th elements of e^{t+1} and A $(= Q\mathbf{c}^t + F\mathbf{c}^t + P - \frac{\Upsilon^{\mathbf{e},t}}{\eta^t})$ respectively, $B_i = \frac{4}{3}A_i\cos^2\theta$, and $\theta = \frac{1}{3}\arccos(-\frac{W_i}{4\eta^t\sqrt{(\frac{\|A_i\|}{3})^3}})$.

given relative rotations, and then the minimization problem (11) is reformulated as:

$$\begin{aligned} &\min_{\mathbf{c},\mathbf{d},\mathbf{e},\mathbf{w}} \sum_{(i,j)\in\mathcal{E}} w_{ij} f(e_{ij}) + \beta\|\mathbf{1}-\mathbf{w}\|_0 \\ &\text{s.t. } \forall(i,j)\in\mathcal{E},\ e_{ij} = ([c_{ij}]_\times - I_3)c_i + d_{ij}c_j - c_{ij}; \\ &c_1 = [0,0,0]^T;\ \forall(i=1,j)\in\mathcal{E}, d_{ij}=1; \\ &\forall(i\neq 1,j)\in\mathcal{E}, d_{ij} = 1 - c_{ij}^T c_i; \mathbf{w}\in\{0,1\}^{m\times 1}. \end{aligned} \tag{13}$$

We employ the Augmented Lagrange Multiplier to solve (13), and the corresponding loss function is:

$$\begin{aligned} &L(\mathbf{c},\mathbf{d},\mathbf{e},\mathbf{w},\Upsilon^{\mathbf{d}},\Upsilon^{\mathbf{e}},\eta) = \sum_{(i,j)\in\mathcal{E}} w_{ij}f(e_{ij}) + \beta\|\mathbf{1}-\mathbf{w}\|_0 \\ &+ \sum_{(i,j)\in\mathcal{E}} \langle \Upsilon_{ij}^{\mathbf{e}}, e_{ij} - ([c_{ij}]_\times - I_3)c_i - d_{ij}c_j + c_{ij}\rangle \\ &+ \sum_{(i\neq 1,j)\in\mathcal{E}} \langle \Upsilon_{ij}^{\mathbf{d}}, d_{ij} - 1 + c_{ij}^T c_i\rangle \\ &+ \frac{\eta}{2}\Bigg(\sum_{(i,j)\in\mathcal{E}} \|e_{ij} - ([c_{ij}]_\times - I_3)c_i - d_{ij}c_j + c_{ij}\|_2^2 \\ &+ \sum_{(i\neq 1,j)\in\mathcal{E}} \|d_{ij} - 1 + c_{ij}^T c_i\|_2^2 \Bigg) \\ &\text{s.t. } c_1 = [0,0,0]^T; \forall(i=1,j)\in\mathcal{E}, d_{ij}=1; \mathbf{w}\in\{0,1\}^{m\times 1} \end{aligned} \tag{14}$$

where $\Upsilon^{\mathbf{d}} = [\Upsilon_{ij}^{\mathbf{d}}]_{(i\neq 1,j)\in\mathcal{E}} \in \mathcal{R}^{(m-k)\times 1}$ and $\Upsilon^{\mathbf{e}} = [\Upsilon_{ij}^{\mathbf{e}}]_{(i,j)\in\mathcal{E}} \in \mathcal{R}^{3m\times 1}$ denote the Lagrange multipliers, η is a positive penalty parameter, and '$\langle\cdot,\cdot\rangle$' is the inner product operator. Theoretically, (14) can be solved by the classical Lagrange multiplier method [42] as:

$$\begin{cases} \{\mathbf{c}^{t+1}, \mathbf{d}^{t+1}, \mathbf{e}^{t+1}, \mathbf{w}^{t+1}\} = \\ \quad \arg\min_{\mathbf{c},\mathbf{d},\mathbf{e},\mathbf{w}} L(\mathbf{c},\mathbf{d},\mathbf{e},\mathbf{w},\Upsilon^{\mathbf{e},t},\Upsilon^{\mathbf{d},t},\eta^t) \\ \Upsilon^{\mathbf{d},t+1} = \Upsilon^{\mathbf{d},t} + \eta^t[d_{ij}^{t+1} - 1 + c_{ij}^T c_i^{t+1}]_{(i\neq 1,j)\in\mathcal{E}} \\ \Upsilon^{\mathbf{e},t+1} = \Upsilon^{\mathbf{e},t} + \\ \quad \eta^t[e_{ij}^{t+1} - ([c_{ij}]_\times - I_3)c_i^{t+1} - d_{ij}^{t+1}c_j^{t+1} + c_{ij}]_{(i,j)\in\mathcal{E}} \\ \eta^{t+1} = \rho\eta^t \end{cases} \tag{15}$$

where $[x_{ij}]_{(i,j)\in\mathcal{E}}$ represents the column vector consisting of all the terms x_{ij}, ρ is a constant, and the subscript t indicates the iteration index.

Then, the alternating direction scheme is adopted to minimize (15) iteratively as done in [3], [39]: at each iterative step of the optimization process, each variable in (15) is updated while the others are fixed, and after one sweep of alternating minimization with regard to all the variables, these Lagrange multipliers are updated. The complete algorithm is listed in Algorithm 1, and it has to be pointed out that the minimization in the first equation of (15) is approximated as a block-wise minimization, which might be suboptimal. The detailed performance of the iterative procedure is described as follows:

For notational convenience, let $r_{ij} \in \{1,2,...,m\}$ indicate the index of the given relative rotation c_{ij}, and we define the following intermediate variable matrices $\{Q, F, P, W, S, H\}$ as follows:

Q is a $3m \times (3n-3)$ sparse matrix, and at its r_{ij}-th($r_{ij} = k+1,...,m$, where k represents the number of the edges connecting the first camera node as defined above) row triplet, the elements at the $(3i-5)$-th, $(3i-4)$-th, $(3i-3)$-th columns are defined as

$$Q_{(3r_{ij}-2):3r_{ij},(3i-5):(3i-3)} = [c_{ij}]_\times - I_3 \tag{16}$$

and the rest elements in Q are set to zero.

F is also a $3m \times (3n-3)$ sparse matrix, and at its r_{ij}-th($r_{ij} = 1,2,...,m$) row triplet, the elements at the $(3j-5)$-th, $(3j-4)$-th, $(3j-3)$-th columns are defined as

$$F_{(3r_{ij}-2):3r_{ij},(3j-5):(3j-3)} = d_{ij}I_3 \tag{17}$$

and the rest elements in F are set to zero.

P is a $3m$-dimensional column vector, whose r_{ij}-th($r_{ij} = 1,2,...,m$) element is defined as:

$$P_{r_{ij}} = -c_{ij} \tag{18}$$

W is a $3m$-dimensional column vector, whose r_{ij}-th($r_{ij} = 1,2,...,m$) element is defined as:

$$W_{r_{ij}} = w_{ij}\mathbf{1}_3 \tag{19}$$

where '$\mathbf{1}_3$' is the 3-dimensional all-one column vector.

S is a $(m-k)\times(3n-3)$ sparse matrix, and at its $(r_{ij}-k)$-th($r_{ij} = k+1,...,m$) row, the elements at the $(3i-5)$-th, $(3i-4)$-th, $(3i-3)$-th($i = 2,3,...,n$) columns are defined as

$$S_{r_{ij}-k,(3i-5):(3i-3)} = c_{ij}^T \tag{20}$$

and the rest elements in S are set to zero.

H is a $3m \times m$ sparse matrix, and at its r_{ij}-th($r_{ij} = 1,2,...,m$) row triplet, the elements at the r_{ij}-th column are defined as

$$H_{(3r_{ij}-2):3r_{ij},r_{ij}} = c_j \tag{21}$$

and the rest elements in H are set to zero.

Optimizing $\mathbf{e}$: Here, we investigate how to obtain $\mathbf{e}$ when the L_2, L_1, and $L_{1/2}$ losses are used respectively as the loss function $f(\cdot)$ in (13).

Given $\{\mathbf{c}^t, \mathbf{d}^t, \mathbf{w}^t, \eta^t, \Upsilon^{\mathbf{e},t}\}$, according to the definitions on $\{Q, F, P, W\}$ in (16), (17), (18), and (19), the optimization problem (15) under the L_2, L_1, and $L_{1/2}$ cases becomes:

$$\arg\min_{\mathbf{e}} \sum W \odot f(\mathbf{e}) + \frac{\eta^t}{2}||\mathbf{e} - Q\mathbf{c}^t - F\mathbf{c}^t - P + \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}||_2^2 \tag{22}$$

where '$\odot$' indicates the Hadamard product, $f(\cdot)$ represents an arbitrary one of the L_2, L_1, and $L_{1/2}$ losses. And the closed-form solutions to (22) under the three cases are listed in Table I, and more computational details could be found in Appendixes A3-A5.

Optimizing $\mathbf{c}$: Given $\{\mathbf{e}^{t+1}, \mathbf{d}^t, \eta^t, \Upsilon^{\mathbf{e},t}, \Upsilon^{\mathbf{d},t}\}$, according to the definitions on $\{Q, F, P, S\}$ in (16), (17), (18), and (20), the optimization problem (15) becomes:

$$\begin{aligned}\arg\min_{\mathbf{c}} \ & ||\mathbf{e}^{t+1} - Q\mathbf{c} - F\mathbf{c} - P + \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}||_2^2 \\ & + ||S\mathbf{c} + \mathbf{d}^t - \mathbf{1}_{m-k} + \frac{\Upsilon^{\mathbf{d},t}}{\eta^t}||_2^2\end{aligned} \tag{23}$$

where '$\mathbf{1}_{m-k}$' is the $(m-k)$-dimensional all-one column vector. It is a standard least-squares problem, and its closed-form solution is:

$$\begin{aligned}\mathbf{c}^{t+1} = & \Big((Q+F)^T(Q+F) + S^T S\Big)^{-1} \Big((Q+F)^T(\mathbf{e}^{t+1} - \\ & P + \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}) - S^T(\mathbf{d}^t - \mathbf{1}_{m-k} + \frac{\Upsilon^{\mathbf{d},t}}{\eta^t})\Big).\end{aligned} \tag{24}$$

Optimizing $\mathbf{d}$: Given $\{\mathbf{c}^{t+1}, \mathbf{e}^{t+1}, \mathbf{w}^t, \eta^t, \Upsilon^{\mathbf{e},t}, \Upsilon^{\mathbf{d},t}\}$, according to the definitions on $\{Q, P, S, H\}$ in (16), (18), (20), and (21), the optimization problem (15) becomes:

$$\begin{aligned}\arg\min_{\mathbf{d}} \ & ||\mathbf{e}^{t+1} - H_1\mathbf{1}_k - H_2\mathbf{d} - Q\mathbf{c}^{t+1} - P + \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}||_2^2 + \\ & ||S\mathbf{c}^{t+1} + \mathbf{d} - \mathbf{1}_{m-k} + \frac{\Upsilon^{\mathbf{d},t}}{\eta^t}||_2^2\end{aligned} \tag{25}$$

where H_1 represents the sub-matrix consisting of the left k columns of H, H_2 represents the sub-matrix consisting of the right $m-k$ columns of H, and '$\mathbf{1}_k$' is the k-dimensional all-one column vector.

It is a standard least-squares problem, and its closed-form solution is:

$$\begin{aligned}\mathbf{d}^{t+1} = & (H_2^T H_2 + I_{m-k})^{-1}\Big(H_2^T(\mathbf{e}^{t+1} - H_1\mathbf{1}_k - Q\mathbf{c}^{t+1} - \\ & P + \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}) - (S\mathbf{c}^{t+1} - \mathbf{1}_{m-k} + \frac{\Upsilon^{\mathbf{d},t}}{\eta^t})\Big)\end{aligned} \tag{26}$$

where I_{m-k} represents the $(m-k)\times(m-k)$ identity matrix.

Optimizing $\mathbf{w}$: Given $\mathbf{e}^{t+1}$, the optimization problem (15) becomes:

$$\begin{aligned}& \arg\min_{\mathbf{w}} \sum_{(i,j)\in\mathcal{E}} w_{ij} f(e_{ij}^{t+1}) + \beta||\mathbf{1} - \mathbf{w}||_0 \\ & \text{s.t. } \mathbf{w} \in \{0,1\}^{m\times 1}\end{aligned} \tag{27}$$

Algorithm 1: CRA

Input: $\{c_{ij}\}_{(i,j)\in\mathcal{E}}$ and initials $\{\mathbf{c}^0, \mathbf{w}^0, \Upsilon_{ij}^{\mathbf{e},0}, \Upsilon_{ij}^{\mathbf{d},0}, \eta^0\}$

Output: $\mathbf{c} = [c_2; c_3; ...; c_n]$

Calculate $d_{ij}^0 = 1 - c_{ij}^T c_i^0, \forall(i \neq 1, j) \in \mathcal{E}$;
Set $t = 0$;
repeat
  Update $\mathbf{e}^{t+1}$ based on the solution reported in Table I;
  Update $\mathbf{c}^{t+1}$ according to (24);
  Update $\mathbf{d}^{t+1}$ according to (26);
  Update $\mathbf{w}^{t+1}$ according to (28);
  Update $\Upsilon^{\mathbf{e},t+1}, \Upsilon^{\mathbf{d},t+1}, \eta^{t+1}$
    $\Upsilon^{\mathbf{d},t+1} = \Upsilon^{\mathbf{d},t} + \eta^t[d_{ij}^{t+1} - 1 + c_{ij}^T c_i^{t+1}]_{(i\neq 1,j)\in\mathcal{E}}$;
    $\Upsilon^{\mathbf{e},t+1} = \Upsilon^{\mathbf{e},t} +$
    $\eta^t[e_{ij}^{t+1} - ([c_{ij}]_\times - I_3)c_i^{t+1} - d_{ij}^{t+1} c_j^{t+1} + c_{ij}]_{(i,j)\in\mathcal{E}}$;
    $\eta^{t+1} = \min(\rho\eta^t, \eta_{\max})$;
  $t = t + 1$;
until *convergence*;

Since $\mathbf{w} \in \{0,1\}^{m\times 1}$, it always holds that $||\mathbf{1} - \mathbf{w}||_0 = \mathbf{1} - \mathbf{w}$. Then, the solution to (27) is:

$$w_{ij}^{t+1} = \begin{cases} 0 & \text{if } f(e_{ij}) \geq \beta \\ 1 & \text{if } f(e_{ij}) < \beta \end{cases} \tag{28}$$

B. Convergence Criterion

The explored Algorithm 1 is an iterative ALM(Augmented Lagrange Multiplier)-based algorithm for solving a multiple-block-variable optimization problem, and as far as we know, there has been a lack of theoretical proof on the convergence of such kind of ALM-based algorithms(e.g. [43], [44]). However, at each iteration during the optimization process of Algorithm 1, a closed-form solution of each variable could be obtained according to the aforementioned description in Section III-A, and we have empirically found that this algorithm always converges under the used three losses (i.e., L_2, L_1, and $L_{1/2}$ losses) in our experiments. Its termination criterion is that the ratio of the values of Func. (13) at two continuous iterations is approximately equal to 1, i.e. $\Big|\Big(\sum_{(i,j)\in\mathcal{E}} w_{ij}^{t+1} f(e_{ij}^{t+1}) + \beta||\mathbf{1}-\mathbf{w}^{t+1}||_0\Big)/\Big(\sum_{(i,j)\in\mathcal{E}} w_{ij}^t f(e_{ij}^t) + \beta||\mathbf{1}-\mathbf{w}^t||_0\Big) - 1\Big| \leq \delta$ (δ is a tiny threshold, and it is set to be 10^{-5} here).

Remark: As noted from Algorithm 1, other than the used L_2, L_1, and $L_{1/2}$ losses, many convex loss functions could be used as $f(\cdot)$ and easily embedded in the proposed CRA approach. Only the variable $\mathbf{e}$ needs to be optimized in a corresponding manner to a new choice of $f(\cdot)$, while the rest variables could be optimized as done in Algorithm 1.

IV. Experimental Results

In this section, three Cayley rotation averaging methods are derived from the proposed CRA by embedding the L_2, L_2, and $L_{1/2}$ losses, denoted as CRA$_2$, CRA$_1$, CRA$_{1/2}$ respectively. All the derived methods are implemented and evaluated using MATLAB on a 3.80GHz desktop.

A. Datasets

Real datasets: The proposed methods are evaluated on 16 benchmark datasets, including *Ellis Island*, *Piazza Del Popolo*, *NYC Library*, *Madrid Metropolis*, *Yorkminster*, *Montreal Notre*

TABLE II
ROTATION ERRORS (IN DEGREES) BY THE PROPOSED METHODS WITH DIFFERENT CHOICES OF β.

Methods	Datasets	$\beta = 0$	$\beta = 10^{-3}$	$\beta = 10^{-2}$	$\beta = 10^{-1}$
CRA_2	Ellis Island	3.48	1.58	1.53	1.90
	Piazza Del Popolo	7.51	4.85	4.80	4.76
	NYC Library	4.91	2.48	2.32	2.73
CRA_1	Ellis Island	1.24	1.17	1.08	1.01
	Piazza Del Popolo	4.89	4.46	4.39	4.27
	NYC Library	2.09	1.40	1.27	1.27
$\text{CRA}_{1/2}$	Ellis Island	0.89	1.08	1.07	1.07
	Piazza Del Popolo	3.76	3.75	3.75	3.75
	NYC Library	1.54	1.11	1.11	1.12

TABLE III
ROTATION ERRORS (IN DEGREES) BY THE PROPOSED METHODS WITH DIFFERENT CHOICES OF δ.

Methods	Datasets	$\delta = 10^{-4}$	$\delta = 10^{-5}$	$\delta = 10^{-6}$
CRA_2	Ellis Island	1.54	1.53	1.54
	Piazza Del Popolo	4.80	4.80	4.81
	NYC Library	2.32	2.32	2.32
CRA_1	Ellis Island	1.09	1.08	1.09
	Piazza Del Popolo	4.39	4.39	4.39
	NYC Library	1.27	1.27	1.26
$\text{CRA}_{1/2}$	Ellis Island	1.07	1.07	1.07
	Piazza Del Popolo	3.75	3.75	3.75
	NYC Library	1.11	1.11	1.11

Dame,*Tower of London*, *Alamo*, *Notre Dame*, *Vienna Cathedral*, *Union Square*, *Roman Forum*, *Piccadilly*, *Trafalgar*, *Arts Quad*, *San Francisco*. And the relative rotations on the first 14 datasets are provided by the authors of [8], while the relative rotations on the last two datasets are provided by the authors of [6]. The second column of Table IV lists the number $n = |\mathcal{V}|$ of the cameras in the largest connected component of these datasets, while the third column of Table IV lists the number $m = |\mathcal{E}|$ of the given relative rotations.

As done in [25], [26], we perform incremental BA(Bundle Adjustment) on all the used datasets, and then the obtained results are used as the ground truths in our experiments. Here, two points have to be explained: (i) The ground-truth results are not available for all the cameras in each dataset, and the fourth column of Table IV lists the number n_g of the cameras with the ground-truth global rotations; (ii) As done in [26], the Gendarmenmarkt dataset is not used for evaluation in this work, because this dataset contains repetitive scene structures and incremental BA could not obtain high-quality absolute camera rotations from this dataset as indicated in [8], [10]. In addition, since there generally exists an unknown non-identical rotation transformation between the estimated set of global rotations by a rotation averaging method and the corresponding ground-truth rotations, we use the sum of squares alignment approach as suggested in [25], [26] to find this rotation transformation and then align the estimated set of global rotations to the ground-truth set before algorithmic evaluation.

Synthetic datasets: In addition to the above real-world datasets, we build three sets of synthetic datasets in controlled-but-challenging conditions for more comprehensive evaluation as follows:

For evaluation in the cases of noise and outliers, we build the first set *SD1* of synthetic datasets: we firstly synthesize 100 absolute camera rotations randomly as ground truths, and accordingly, we obtain totally 4950 relative rotations among these cameras. Then, we add random Gaussian noise with mean of 0 and standard deviation of 30^o into each relative rotation. Next, we use the selection strategy suggested in [26] to automatically selected 20% of these noisy relative rotations as the observed relative rotations. Finally, in order to investigate the performances of the proposed method and the comparative methods with increasing amounts of outliers, a specific outlier percentage $p = \{10\%, 15\%, 20\%\}$ of the observed relative rotations are replaced with other rotations so that they could be used as outliers. Under each configuration of p, we synthesize 10 datasets independently for evaluation.

For evaluation in the challenging case where some input relative rotations are close to π, we build the second set *SD2* of synthetic datasets: we firstly synthesize 100 absolute camera rotations randomly as ground truths, and accordingly, we obtain totally 4950 relative rotations among these cameras. Then, we add random Gaussian noise with mean of 0 and standard deviation of 30^o into each relative rotation. Next from these noisy relative rotations, we select the ones whose absolute rotation difference from π is smaller than $D = \{\pi/4, \pi/6, \pi/12\}$ respectively as three initial synthetic datasets. Finally, for each initial synthetic dataset, we randomly keep a percentage $s = \{10\%, 20\%, 30\%\}$ of relative rotations but replace the rest relative rotations with randomly selected relative rotations whose absolute rotation difference from π is larger than or equal to D, resulting in the synthetic datasets where the percentage of cases near π and the distance between the rotation angle and π can be controlled. Under each of the above parameter configurations, we synthesize 10 datasets independently for evaluation.

In order to further evaluate the performance of the proposed method for handling the above challenging case with more synthetic cameras, we build the third set *SD3* of synthetic datasets with 1000 cameras by utilizing the same data synthesis manner of *SD2*.

B. Implementation Details

Similar to other iterative algorithms, Algorithm 1 needs a set of initial values of the involved variables: As done in [26], we use the estimated raw rotations by the L1RA method [25] to initialize the rotation variable $\mathbf{c}$. The maximum number of iterations in the L1RA initialization is preset to 5. The weight variable $\mathbf{w}$ is simply initialized with an all-one vector, and the variables $\{\mathbf{d}, \mathbf{e}\}$ are not initialized manually but updated automatically. The iterative parameters $\{\Upsilon^{\mathbf{e},t},\ \Upsilon^{\mathbf{d},t},\ \eta^t\}$ in Algorithm 1 are simply initialized as: $\Upsilon^{\mathbf{e},0} = \Upsilon^{\mathbf{d},0} = 0$, $\eta^0 = 10$. The updating orders for all the variables are outlined in Algorithm 1.

C. Influence of the Weight β in (11)

Here, we evaluate the influence of the weight β on the proposed three methods, including CRA_2, CRA_1, and $\text{CRA}_{1/2}$. The three methods with $\beta = \{0, 10^{-3}, 10^{-2}, 10^{-1}\}$ are tested respectively on the first three datasets in Table IV, and the corresponding results are reported in Table II. Here, for the case of $\beta = 0$, the weight vector $\mathbf{w}$ in Eq. (11) are set to be an all-one vector constantly during the optimization process of Algorithm 1.

Two points could be observed from Table II:

- The angle errors by the three proposed methods with $\beta = 0$ are higher than those with $\beta = \{10^{-3}, 10^{-2}, 10^{-1}\}$ in most cases, indicating the effectiveness of the introduced sparsity regularizer on the vector $\mathbf{1} - \mathbf{w}$ in Eq. (11). It is also noted that the angle error of $\text{CRA}_{1/2}$ with $\beta = 0$ on the *Ellis Island* dataset is exceptionally lower than those of $\text{CRA}_{1/2}$ with the other values of β. The reason is two-fold: (i) The used $L_{1/2}$ norm in $\text{CRA}_{1/2}$ is relatively insensitive to outliers than both the L_2 and L_1 losses, consistent with the reported results with different losses in [26]; (ii) The input set of relative rotations on the *Ellis Island* dataset are corrupted by slight noise and few outliers, and the introduced weights with $\beta > 0$ probably weaken some available ones from the constructed Cayley measurements in Eq. (11) to some extent in this case.
- It is noted that when β ranges from 10^{-3} to 10^{-1}, the corresponding angle errors change slightly. This issue demonstrates the proposed model is not quite sensitive to the choice of the tuning parameter β. In the rest experiments, β is set to be 10^{-2} all the time.

D. Influence of the Threshold δ in Termination Criterion

In order to verify whether the proposed methods are sensitive to the threshold δ in the termination criterion, we evaluate the proposed methods with $\delta = \{10^{-4}, 10^{-5}, 10^{-6}\}$ respectively on the first three datasets. The corresponding median angle errors are reported in Table III . As seen from this table, when δ ranges from 10^{-4} to 10^{-6}, the corresponding angle errors change slightly. These results demonstrate the proposed methods are not quite sensitive to the choice of the threshold δ. In the rest experiments, δ is set to be 10^{-5} all the time.

E. Accuracy Comparison

1) Comparison on Real Datasets: In this subsection, we evaluate the proposed methods in comparison to five typical global rotation averaging methods, including $\text{FRRA}_{1/2}$[26], DISCO[6], Weiszfeld[22], RA-SfM[12], and NeuRoRA[11]. Considering that the reported results of NeuRoRA in [11] are obtained by implementing the absolute sum alignment approach, which is not recommended in [26] in comparison to the sum of squares alignment approach, Table IV reports the median rotation errors by the first four methods on the aforementioned datasets. It has to be further pointed out that the rotations errors by DISCO [6], Weiszfeld [22], and $\text{FRRA}_{1/2}$ [26] in this table, are straightforwardly cited from [26]. The results by RA-SfM [12] are obtained by implementing the C++ code released by its authors[1].

As seen from Table IV, the performance of the explored method CRA_2 by minimizing the L_2 loss is inferior to the other two explored methods with the L_1 and $L_{1/2}$ losses, mainly because the L_2 loss is generally more sensitive to outliers, as indicated in many existing works [45], [46] for handling other visual tasks. The proposed CRA_1 method performs better than DISCO[6] and Weiszfeld[22], and it achieves close performances to $\text{FRRA}_{1/2}$[26] and RA-SfM[12]. In addition, it is noted that all the used 16 datasets are obtained from real-world scenarios, and there exists remarkable differences among these datasets. It is quite difficult for a given rotation averaging method to achieve high-accuracy performances on all of them. However, the proposed $\text{CRA}_{1/2}$ method achieves either the first place or the second place in most cases. If comparing $\text{CRA}_{1/2}$ with each comparative method singly, $\text{CRA}_{1/2}$ performs better in most cases. For example, if comparing the proposed $\text{CRA}_{1/2}$ with RA-SfM singly, RA-SfM performs better on 6 of the total 16 datasets, while $\text{CRA}_{1/2}$ performs better on the rest 10 datasets; if comparing with $\text{FRRA}_{1/2}$, $\text{CRA}_{1/2}$ performs better on the rest 10 datasets. These results demonstrate the effectiveness of the proposed Cayley-based methods.

Moreover, it is noted that $\text{FRRA}_{1/2}$[26] and RA-SfM[12] perform much better than DISCO[6] and Weiszfeld[22]. Table V further reports the median rotation errors by $\text{FRRA}_{1/2}$, RA-SfM, NeuRoRA[11], and the proposed $\text{CRA}_{1/2}$ on the aforementioned datasets under the absolute sum alignment approach. It has to be pointed out that NeuRoRA is a learning-based method, which is trained by utilizing the leave-one-out strategy in their original paper, and its rotations errors in this table are straightforwardly cited from [11]. As seen from the last two rows of Table V, when handling the ArtsQuad and SanFran datasets that come from a different source from the other evaluation datasets, the performance of NeuRoRA decreases significantly, while the other comparative methods perform much better than NeuRoRA on the two datasets. As seen from Table V and Table IV, the performance of the proposed $\text{CRA}_{1/2}$ method under the absolute sum alignment approach is consistent with its performance under the sum of squares alignment approach. If comparing $\text{CRA}_{1/2}$ with each comparative method singly, $\text{CRA}_{1/2}$ still performs better in

[1]https://github.com/PeterZs/GraphOptim

This article has been accepted for publication in IEEE Transactions on Image Processing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIP.2024.3416057

TABLE IV

COMPARISON OF ROTATION ERRORS (IN DEGREES) ON THE BENCHMARK DATASETS UNDER THE SUM OF SQUARES ALIGNMENT APPROACH: m DENOTES THE NUMBER OF THE INPUT RELATIVE ROTATIONS, n DENOTES THE TOTAL NUMBER OF GLOBAL ROTATIONS, n_g DENOTES THE NUMBER OF THE GROUND-TRUTH GLOBAL ROTATIONS. THE BEST AND THE SECOND BEST RESULTS ARE IN BOLD AND UNDERLINED TYPES RESPECTIVELY. '–' INDICATES THAT DISCO FAILED ON THE 'NOTRE DAME' DATASET AS INDICATED IN [26]

Dataset	n	m	n_g	DISCO[6]	Weiszfeld[22]	$\text{FRRA}_{1/2}$[26]	RA-SfM[12]	CRA_2	CRA_1	$\text{CRA}_{1/2}$
Ellis Island	247	20297	227	1.82	1.66	1.15	**1.03**	1.53	1.08	$\underline{1.07}$
Piazza Del Popolo	354	24710	338	5.25	$\underline{3.35}$	**2.62**	3.82	4.80	4.39	3.75
NYC Library	376	20680	332	2.59	2.43	1.40	1.39	2.32	$\underline{1.27}$	**1.11**
Madrid Metropolis	394	23784	341	6.64	4.37	$\underline{3.08}$	**2.38**	8.00	4.51	3.84
Yorkminster	458	27729	437	2.34	2.73	1.62	1.68	2.27	**1.31**	$\underline{1.34}$
Montreal Notre Dame	474	52424	450	1.03	0.92	0.71	0.76	1.22	$\underline{0.65}$	**0.51**
Tower of London	508	23863	472	2.73	2.73	2.45	2.58	2.61	$\underline{2.36}$	**1.92**
Alamo	627	97206	577	4.21	3.57	$\underline{2.14}$	2.20	2.82	2.49	**2.11**
Notre Dame	715	64678	715	–	0.50	$\underline{0.49}$	0.52	0.62	**0.43**	$\underline{0.49}$
Vienna Cathedral	918	103350	853	14.57	5.14	4.64	**4.47**	5.36	4.61	$\underline{4.59}$
Union Square	930	25561	789	7.50	13.54	$\underline{4.97}$	5.02	5.42	5.00	**4.39**
Roman Forum	1134	70187	1084	13.69	2.11	$\underline{1.70}$	**1.61**	5.25	2.52	2.21
Piccadilly	2508	319257	2152	14.66	7.65	3.12	$\underline{2.58}$	9.31	6.66	**2.53**
Trafalgar	5433	680012	5058	91.62	13.20	2.03	14.73	3.10	2.18	**2.00**
Arts Quad	5530	222044	4978	88.58	6.95	$\underline{2.54}$	**2.12**	4.17	3.83	3.91
San Francisco	7866	101512	7866	54.38	15.85	$\underline{3.56}$	**3.26**	4.46	4.01	4.11

TABLE V

COMPARISON OF MEDIAN ROTATION ERRORS (IN DEGREES) ON THE BENCHMARK DATASETS UNDER THE ABSOLUTE SUM ALIGNMENT MANNER. THE BEST AND THE SECOND BEST RESULTS ARE IN BOLD AND UNDERLINED TYPES RESPECTIVELY.

Datasets	$\text{FRRA}_{1/2}$[26]	RA-SfM[12]	NeuRoRA[11]	$\text{CRA}_{1/2}$
Ellis Island	$\underline{0.52}$	0.58	0.60	**0.39**
Piazza Del Popolo	0.90	0.95	**0.70**	$\underline{0.74}$
NYC Library	1.37	1.34	$\underline{1.10}$	**1.05**
Madrid Metropolis	1.31	**1.06**	$\underline{1.10}$	1.58
Yorkminster	1.59	1.68	**0.9**	$\underline{1.26}$
Montreal Notre Dame	$\underline{0.52}$	0.59	0.60	**0.38**
Tower of London	2.44	2.55	**1.40**	$\underline{1.85}$
Alamo	$\underline{1.07}$	1.09	1.20	**0.99**
Notre Dame	**0.42**	0.52	0.60	$\underline{0.43}$
Vienna Cathedral	$\underline{1.27}$	1.38	1.50	**1.18**
Union Square	$\underline{3.93}$	4.14	**2.0**	4.31
Roman Forum	1.60	$\underline{1.53}$	**1.30**	2.31
Piccadilly	2.94	2.42	**1.9**	$\underline{2.26}$
Trafalgar	$\underline{2.02}$	13.19	2.2	**1.98**
Arts Quad	$\underline{2.53}$	**2.12**	7.3	3.13
San Francisco	$\underline{3.31}$	**3.20**	12.6	3.89

most cases. These results further demonstrate the effectiveness of the proposed Cayley-based methods.

2) Comparison on Synthetic Datasets: Here, considering that the comparative methods $\text{FRRA}_{1/2}$ [26] and RA-SfM[12] perform much better than DISCO[6] and Weiszfeld[22], we evaluate $\text{FRRA}_{1/2}$, RA-SfM, and the proposed method $\text{CRA}_{1/2}$ on the synthetic dataset in controlled-but-challenging conditions for comparison.

Firstly, in order to verify the effectiveness of the proposed method for handling the cases of noise and outliers, we evaluate the three methods on the *SD1* set of synthetic datasets, where the relative rotations are corrupted by Gaussian noise and different percentages $p = \{10\%, 15\%, 20\%\}$ of outliers as described in Section IV-A. The corresponding mean, median, and maximum rotation errors are reported in Table VI. It has to be explained that this table also reports the results of the three methods on noisy relative rotations without outliers (i.e., $p = 0$) for comparison. As seen from this table, when the input relative rotations are only corrupted by Gaussian noise without outliers, the proposed $\text{CRA}_{1/2}$ outperforms the two comparative methods. With the increase of outliers, the performances of all the three methods decrease to some extent, and the proposed method always performs better than $\text{FRRA}_{1/2}$ and RA-SFM under different outlier percentages. These results demonstrate the effectiveness of the proposed method for resisting outliers and noise to some extent.

Secondly, as indicated in Section II-A, the Cayley rotation representation assumes that the rotation angle is not equal to π. Although this assumption holds in most real-world datasets, in order to understand the sensitivity of the Cayley representation to π, we evaluate the three methods on both the *SD2* and *SD3* sets of synthetic datasets, where the percentage $s = \{10\%, 20\%, 30\%\}$ of cases near π and the distance $D = \{\pi/4, \pi/6, \pi/12\}$ between the rotation angle and π can be controlled. The corresponding mean, median, and maximum rotation errors are reported in Table VII and Table VIII. As seen from the two table, although the two sets of datasets have different camera numbers (i.e., 100 cameras and 1000 cameras), the corresponding results on the two sets of datasets show the following consistent phenomenon: by fixing the percentage s, with the decrease of D, the calculated rotation errors of the three methods are both increased, demonstrating that a given percentage of relative rotations that are closer to π would lead to a worse result. Moreover, by fixing D, with the increase of the percentage s, the calculated rotation errors of the proposed $\text{CRA}_{1/2}$ varies slightly. Moreover, the proposed method outperforms the other two comparative methods in all the cases. These results do not only demonstrate that the Cayley rotation representation is not quite sensitive to π, but also demonstrate the superiority of the proposed Cayley-based method.

F. Time Comparison

In order to compare the computational speeds of the comparative methods, the running times of these methods on all the datasets are reported in Table IX. In addition, for the proposed methods, the iteration numbers of their L1RA initialization and Cayley optimization, as well as their corresponding times, are reported in Table X.

Here, the following points on the two tables have to be explained: (i) The reported computational time for each dataset in Table IX is the sum of the times of the L1RA initialization and the Cayley optimization listed in Table X. (ii) As done in [26], the running times of Weiszfeld[22] (by implementing our reproduced code), FRRA$_{1/2}$[26] (by implementing the code released by the authors), as well as the proposed methods, are obtained via a MATLAB implementation on a 3.80GHz desktop; (iii) As indicated in [26], DISCO [6] is computationally expensive and it needs a cluster to deal with a large dataset, rather than a desktop. Hence, Table IX straightforwardly cites the running times of DISCO from [26], which were obtained via a mixed implementation (C++, MATLAB) on a 36 node cluster, and each node contains 2 2.67 GHz quad-core processors; (iv) As introduced above, the authors in [12] provide the C++ code for RA-SfM, and accordingly, the running times of RA-SfM in Table IX are obtained by implementing the C++ code on a desktop.

As seen from Table IX, all the proposed methods (CRA$_2$, CRA$_1$, CRA$_{1/2}$) run faster than DISCO[6] and Weiszfeld[22]. Compared with FRRA$_{1/2}$, the three proposed methods run relatively slowly on the datasets that contain a small/moderate number of relative rotations. However, it is noted that when dealing with both the Piccadilly and Trafalgar datasets that contain a relatively large number of relative rotations, the three proposed methods run faster than FRRA$_{1/2}$. In addition, RA-SfM [12] runs fastest among all the comparative methods, mainly because C++ (for RA-SfM) generally has a faster implementation speed than its counterpart MATLAB (for FRRA$_{1/2}$, CRA$_2$, CRA$_1$, and CRA$_{1/2}$).

Moreover, it is noted from Table IX that although the derived three methods (CRA$_2$, CRA$_1$, CRA$_{1/2}$) have similar optimization processes according to Algorithm 1, they spend different times on most of the datasets, mainly because of their different iteration steps when meeting the termination condition of Algorithm 1.

G. Influence of Initialization

It is noted that in the above experiments, we use the estimated rotations by the L1RA method [25] as initials as done in [26]. Here, in order to evaluate the sensitivity and convergence of the proposed method CRA$_{1/2}$ to the initialization, we evaluate the proposed CRA$_{1/2}$ with random initialization on each of the first five datasets listed in Table IV 5 times independently. The corresponding mean and standard deviation values of the median rotation errors are reported in Table XI. In addition, the initial rotation errors of L1RA are also reported for comparison in Table XI. As seen from this table, three points could be observed:

1 With different initials, the performance of the proposed method varies in a moderate range.

2 In most cases, the initial errors of L1RA are larger than those of the proposed CRA$_{1/2}$ by a clear margin ($> 30\%$), demonstrating that the effectiveness of CRA$_{1/2}$ mainly comes from its own specific design, rather than its initial estimator L1RA. In addition, the performance of the proposed method with the L1RA initialization is better than or close to that with random initialization in most cases. These results demonstrate that a strong initialization could improve the performance of the proposed method than the random initialization to some extent. Moreover, we also like to indicate that by comparing the above results in Table IV and Table XI, the performance of the proposed CRA$_{1/2}$ with random initialization is still competitive in comparison to these comparative methods.

3 For the dataset "Piazza Del Popolo", the proposed CRA$_{1/2}$ with the L1RA initialization obtains a poorer result than CRA$_{1/2}$ with 5 random initializations. The reason is probably that: the problem for optimizing the proposed CRA$_{1/2}$ is a non-linear optimization problem, and the designed algorithm for optimizing CRA$_{1/2}$ is an iterative algorithm, whose convergence solution is dependent on its initial value. As indicated in the numerical optimization theory, for a given non-linear optimization algorithm, a relatively better initial is generally helpful to search for a convergence solution, but the better initial value could not guarantee that this algorithm could surely obtain a more accurate solution, since this algorithm with the better initial might converge to a poorer local minimum occasionally. Hence, although the L1RA initialization approach could provide a better initial value than the random initialization approach, the proposed CRA$_{1/2}$ with the L1RA initialization might obtain a worse result on some data (e.g., Piazza Del Popolo) than CRA$_{1/2}$ with a few random initializations. However, as discussed in the second point, a strong initialization is beneficial to the proposed CRA$_{1/2}$ in most cases.

V. Conclusions

In this paper, we propose the Cayley-based approach CRA for rotation averaging. Specifically, a general Cayley rotation constraint equation is introduced by utilizing the Cayley rotation representation, which could characterize the relationship between the relative rotations of pairwise cameras and their global rotations in the cases of noise and outliers. Then, the CRA approach is explored by integrating a set of Cayley rotation constraints. In addition, a sparsity regularizer is introduced into CRA for alleviating the influence of outliers further. Experimental results on both real and synthetic datasets demonstrate the effectiveness of the proposed approach.

Currently, the proposed CRA is explored under the non-incremental rotation averaging strategy. In the future, we

TABLE VI
COMPARISON OF ROTATION ERRORS (IN DEGREES) ON THE *SD1* SYNTHETIC DATASETS WITH DIFFERENT OUTLIER PERCENTAGES $p = \{0, 10\%, 15\%, 20\%\}$. MN INDICATES THE MEAN ROTATION ERROR, MD INDICATES THE MEDIAN ERROR, AND MAX INDICATES THE MAXIMUM ERROR. THE BEST RESULTS ARE IN BOLD.

	$p=0$			$p=10\%$			$p=15\%$			$p=20\%$		
Methods	Mn	Md	MAX	Mn	Md	MAX	Mn	Md	MAX	Mn	Md	MAX
$\text{FRRA}_{1/2}$ [26]	13.80	13.29	31.22	14.96	14.39	33.11	15.08	14.53	36.96	15.55	14.65	42.40
RA-SfM[12]	11.06	10.73	27.01	11.86	11.54	29.56	12.27	11.82	30.31	12.92	12.12	33.81
$\text{CRA}_{1/2}$	**9.26**	**8.94**	**21.13**	**10.46**	**9.82**	**24.41**	**11.14**	**10.70**	**27.71**	**12.36**	**11.42**	**31.03**

TABLE VII
COMPARISON OF ROTATION ERRORS (IN DEGREES) ON THE *SD2* SYNTHETIC DATASETS WHERE THE PERCENTAGE $s = \{10\%, 20\%, 30\%\}$ OF CASES NEAR π AND THE DISTANCE $D = \{\pi/4, \pi/6, \pi/12\}$ BETWEEN THE ROTATION ANGLE AND π. MN INDICATES THE MEAN ROTATION ERROR, MD INDICATES THE MEDIAN ERROR, AND MAX INDICATES THE MAXIMUM ERROR. THE BEST RESULTS ARE IN BOLD.

		$D=\pi/4$			$D=\pi/6$			$D=\pi/12$		
s	Methods	Mn	Md	MAX	Mn	Md	MAX	Mn	Md	MAX
$s=10\%$	$\text{FRRA}_{1/2}$ [26]	6.27	5.75	15.30	8.23	8.00	19.71	11.98	11.23	29.03
	RA-SfM[12]	9.61	9.47	22.62	11.40	10.99	26.30	15.12	14.06	36.25
	$\text{CRA}_{1/2}$	**5.85**	**5.44**	**14.27**	**6.98**	**6.59**	**16.31**	**10.24**	**9.85**	**24.92**
$s=20\%$	$\text{FRRA}_{1/2}$ [26]	6.55	6.23	15.05	7.88	7.30	19.17	12.65	11.71	31.03
	RA-SfM[12]	10.07	9.81	23.38	11.64	11.23	26.30	15.15	13.81	36.46
	$\text{CRA}_{1/2}$	**5.92**	**5.66**	**13.85**	**7.13**	**6.56**	**17.37**	**10.56**	**10.15**	**25.77**
$s=30\%$	$\text{FRRA}_{1/2}$ [26]	6.43	6.01	16.71	7.98	7.63	20.27	12.40	11.55	31.33
	RA-SfM[12]	9.57	9.13	21.15	11.85	11.37	25.74	15.20	14.31	34.25
	$\text{CRA}_{1/2}$	**5.96**	**5.53**	**14.99**	**7.16**	**6.97**	**16.54**	**10.70**	**10.11**	**25.33**

TABLE VIII
COMPARISON OF ROTATION ERRORS (IN DEGREES) ON THE *SD3* SYNTHETIC DATASETS WHERE THE PERCENTAGE $s = \{10\%, 20\%, 30\%\}$ OF CASES NEAR π AND THE DISTANCE $D = \{\pi/4, \pi/6, \pi/12\}$ BETWEEN THE ROTATION ANGLE AND π. MN INDICATES THE MEAN ROTATION ERROR, MD INDICATES THE MEDIAN ERROR, AND MAX INDICATES THE MAXIMUM ERROR. THE BEST RESULTS ARE IN BOLD.

		$D=\pi/4$			$D=\pi/6$			$D=\pi/12$		
s	Methods	Mn	Md	MAX	Mn	Md	MAX	Mn	Md	MAX
$s=10\%$	$\text{FRRA}_{1/2}$ [26]	**1.83**	**1.74**	5.42	2.23	2.14	**5.90**	3.17	3.03	9.00
	RA-SfM[12]	4.45	4.29	12.19	5.06	4.82	13.53	6.37	6.15	16.27
	$\text{CRA}_{1/2}$	**1.83**	**1.74**	**5.12**	**2.20**	**2.11**	5.94	**3.09**	**2.98**	**8.17**
$s=20\%$	$\text{FRRA}_{1/2}$ [26]	1.88	1.80	5.50	2.26	2.16	6.38	3.24	3.08	10.18
	RA-SfM[12]	4.47	4.27	11.81	5.02	4.92	13.15	6.45	6.18	16.10
	$\text{CRA}_{1/2}$	**1.84**	**1.77**	**5.26**	**2.24**	**2.14**	**6.15**	**3.13**	**3.00**	**8.35**
$s=30\%$	$\text{FRRA}_{1/2}$ [26]	1.89	1.80	5.49	2.28	2.20	6.26	3.30	3.14	10.68
	RA-SfM[12]	4.46	4.30	11.24	5.11	4.95	12.87	6.49	6.21	16.85
	$\text{CRA}_{1/2}$	**1.86**	**1.76**	**5.48**	**2.24**	**2.14**	**5.97**	**3.19**	**3.05**	**8.73**

TABLE IX
COMPUTATIONAL TIME (IN SECONDS) COMPARISON ON THE BENCHMARK DATASETS.

Dataset	DISCO[6]	Weiszfeld[22]	$\text{FRRA}_{1/2}$[26]	RA-SfM[12]	CRA_2	CRA_1	$\text{CRA}_{1/2}$
Ellis Island	470	13	1	1	2	2	3
Piazza Del Popolo	583	19	2	1	3	3	6
NYC Library	446	14	1	1	2	3	5
Madrid Metropolis	560	18	1	1	3	3	6
Yorkminster	641	23	1	1	3	3	6
Montreal Notre Dame	1608	70	4	2	6	6	11
Tower of London	479	19	1	1	3	3	6
Alamo	3917	212	12	5	11	12	26
Notre Dame	–	99	4	3	6	8	17
Vienna Cathedral	4085	243	12	8	10	14	27
Union Square	466	23	3	2	4	4	7
Roman Forum	1559	120	6	6	32	10	16
Piccadilly	15604	5869	164	82	63	56	123
Trafalgar	43616	27519	404	223	530	229	348
Arts Quad	5227	3140	129	80	201	131	164
San Francisco	1413	565	201	39	212	190	200

would investigate how to extend the current version of CRA into an incremental version under the incremental rotation averaging strategy, for pursuing higher accuracy.

APPENDIX

A1. Derivation on Eq. (7)

According to the definition of Cayley transformation (i.e., Eq. (3)), Eq. (6 could be reformulated as:

$$R = \psi([c]_\times) = (I_3 - [c]_\times)(I_3 + [c]_\times)^{-1} \quad . \tag{29}$$

And according to the definition of the skew-symmetric matrix

TABLE X
ITERATION NUMBERS OF THE L1RA INITIALIZATION (DENOTED AS INIT) AND THE CAYLEY OPTIMIZATION (DENOTED AS CAY) AND THE CORRESPONDING TIMES (SECONDS) BY IMPLEMENTING THE PROPOSED METHODS (CRA_2, CRA_1, $CRA_{1/2}$).

Datasets	CRA_2				CRA_1				$CRA_{1/2}$			
	No. Iterations		Time		No. Iterations		Time		No. Iterations		Time	
	Init	Cay	Init	Cay	Init	Cay	Init	Cay	Init	Cay	Init	Cay
Ellis Island	5	38	0.4	2	5	40	0.4	2	5	67	0.4	3
Piazza Del Popolo	5	37	1	2	5	41	1	2	5	91	1	5
NYC Library	5	35	1	1	5	41	1	2	5	89	1	4
Madrid	5	36	1	2	5	37	1	2	5	93	1	5
Yorkminster	5	36	1	2	5	41	1	2	5	72	1	5
Montreal ND	5	34	1	5	5	39	1	5	5	71	1	10
Tower of London	5	35	1	2	5	42	1	2	5	92	1	5
Alamo	5	34	2	9	5	38	2	10	5	90	2	24
Notre Dame	3	28	1	5	3	41	1	7	3	88	1	16
Vienna Cathedral	5	26	3	7	5	39	3	11	5	83	3	24
Union Square	5	42	1	3	5	42	1	3	5	89	1	6
Roman Forum	5	150	3	29	5	40	3	7	5	66	3	13
Piccadilly	5	39	21	42	5	34	21	35	5	91	21	102
Trafalgar	5	150	106	424	5	45	106	123	5	84	106	242
Arts Quad	5	150	98	103	5	49	98	33	5	92	98	66
San Francisco	4	150	172	40	4	68	172	18	4	98	172	28

TABLE XI
COMPARISON OF ROTATION ERRORS (IN DEGREE) BETWEEN L1RA INITIALIZATION AND RANDOM INITIALIZATION FOR THE PROPOSED $CRA_{1/2}$. MEAN AND SD INDICATE THE MEAN VALUE AND THE STANDARD DEVIATION OF THE ROTATION ERRORS COMPUTED 5 TIMES INDEPENDENTLY.

Datasets	$CRA_{1/2}$ with L1RA initialization		$CRA_{1/2}$ with random initialization	
	L1RA error	$CRA_{1/2}$ error	Mean	Sd
Ellis Island	1.38	1.07	1.14	0.39
Piazza Del Popolo	4.62	3.75	2.44	0.38
NYC Library	2.17	1.11	1.29	0.14
Madrid Metropolis	4.71	3.84	3.98	0.38
Yorkminster	1.89	1.34	1.33	0.07

in Eq. (5), it holds

$$(I_3 + [c]_\times)^{-1} = \frac{cc^T + I_3 - [c]_\times}{1 + c^T c} \quad . \tag{30}$$

Then according to Eq. (29) and Eq. (30), we have

$$\begin{aligned} R &= (I_3 - [c]_\times)(I_3 + [c]_\times)^{-1} \\ &= (I_3 - [c]_\times) \cdot \frac{cc^T + I_3 - [c]_\times}{1 + c^T c} \\ &= \frac{cc^T - [c]_\times cc^T + I_3 - 2[c]_\times + [c]_\times [c]_\times}{1 + c^T c} \quad . \end{aligned} \tag{31}$$

It is noted from Eq. (31) that $[c]_\times cc^T = 0_{3\times 3}$ and $[c]_\times [c]_\times = cc^T - c^T c I_3$. Hence, Eq. (31) could be reformulated as the following form (i.e., Eq. (7)):

$$\begin{aligned} R &= \frac{cc^T + I_3 - 2[c]_\times + cc^T - c^T c I_3}{1 + c^T c} \\ &= \frac{(1 - c^T c) I_3 - 2[c]_\times + 2cc^T}{1 + c^T c} \quad . \end{aligned} \tag{32}$$

A2. Derivation on Eq. (8)

Let $\{R_i, R_j\}$ be an arbitrary pair of absolute rotation matrices, and let R_{ij} be the corresponding relative rotation matrix. Accordingly, it holds that $R_j = R_{ij} R_i$ (i.e., Eq. (1)). Let $\{c_i, c_j, c_{ij}\}$ be the corresponding Cayley representations to $\{R_i, R_j, R_{ij}\}$. According to Eq. (7), we have

$$R_i = \frac{(1 - c_i^T c_i) I_3 - 2[c_i]_\times + 2c_i c_i^T}{1 + c_i^T c_i} \tag{33}$$

$$R_{ij} = \frac{(1 - c_{ij}^T c_{ij}) I_3 - 2[c_{ij}]_\times + 2c_{ij} c_{ij}^T}{1 + c_{ij}^T c_{ij}} \tag{34}$$

According to Eq. (3) and Eq. (6), we have:

$$[c_j]_\times = \psi(R_j) = \psi(R_{ij} R_i) = (I_3 - R_{ij} R_i)(I_3 + R_{ij} R_i)^{-1} \tag{35}$$

where I_3 represents the 3-order identity matrix, and $\psi(\bullet)$ is the Cayley transformation that is defined as Eq. (3).

Let $c_{ij} = [c_{ij,x}, c_{ij,y}, c_{ij,z}]^T$ and $c_i = [c_{i,x}, c_{i,y}, c_{i,z}]^T$, then by substituting Eq. (33) and Eq. (34) into Eq. (35), we have Eq. (36) as shown on the top of the following page. According to Eq. (36), it holds that $c_j = \frac{c_{ij} + c_i - [c_{ij}]_\times c_i}{1 - c_{ij}^T c_i}$. Then by multiplying the two sides of this equation with $1 - c_{ij}^T c_i$, we obtain Eq. (8), i.e.,

$$c_{ij} = ([c_{ij}]_\times - I_3) c_i + (1 - c_{ij}^T c_i) c_j .$$

A3. Solution to (22) under the L_2 loss

It is noted that the elements in $W \in \mathcal{R}^{3m\times 1}$ (as defined in (19)) are either 1 or 0. When $f(\cdot)$ represents the L_2 loss, Func. (22) is re-formulated as:

$$\arg\min_{\mathbf{e}} \; ||W \odot \mathbf{e}||_2^2 + \frac{\eta^t}{2} ||\mathbf{e} - Q\mathbf{c}^t - F\mathbf{c}^t - P + \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}||_2^2 \tag{37}$$

where '$\odot$' indicates the Hadamard product.

The closed-form solution to (37) could be obtained by calculating the stationary point where the derivative of (37) equals to zero:

$$\mathbf{e}^{t+1} = (Q\mathbf{c}^t + F\mathbf{c}^t + P - \frac{\Upsilon^{\mathbf{e},t}}{\eta^t})./(\frac{2W}{\eta^t} + 1) \tag{38}$$

where './' indicates the element-wise division.

$$[c_j]_\times = \begin{pmatrix} 0 & -\frac{c_{ij,z}+c_{i,z}-c_{ij,x}c_{i,y}+c_{ij,y}c_{i,x}}{1-c_{ij}^T c_i} & \frac{c_{ij,y}+c_{i,y}+c_{ij,x}c_{i,z}-c_{ij,z}c_{i,x}}{1-c_{ij}^T c_i} \\ \frac{c_{ij,z}+c_{i,z}-c_{ij,x}c_{i,y}+c_{ij,y}c_{i,x}}{1-c_{ij}^T c_i} & 0 & -\frac{c_{ij,x}+c_{i,x}-c_{ij,y}c_{i,z}+c_{ij,z}c_{i,y}}{1-c_{ij}^T c_i} \\ -\frac{c_{ij,y}+c_{i,y}+c_{ij,x}c_{i,z}-c_{ij,z}c_{i,x}}{1-c_{ij}^T c_i} & \frac{c_{ij,x}+c_{i,x}-c_{ij,y}c_{i,z}+c_{ij,z}c_{i,y}}{1-c_{ij}^T c_i} & 0 \end{pmatrix} = [\frac{c_{ij}+c_i-[c_{ij}]_\times c_i}{1-c_{ij}^T c_i}]_\times \tag{36}$$

A4. Solution to (22) under the L_1 loss

Similar to the L_2 case, when $f(\cdot)$ represents the L_1 loss, Func. (22) is re-formulated as:

$$\arg\min_{\mathbf{e}} \ ||W \odot \mathbf{e}||_1 + \frac{\eta^t}{2}||\mathbf{e} - Q\mathbf{c}^t - F\mathbf{c}^t - P + \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}||_2^2 \tag{39}$$

where '$\odot$' indicates the Hadamard product.

The closed-form solution to (39) is

$$\mathbf{e}^{t+1} = \mathcal{D}(Q\mathbf{c}^t + F\mathbf{c}^t + P - \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}, \frac{W}{\eta^t}) \tag{40}$$

where $\mathcal{D}(x,\varrho) = \operatorname{sgn}(x)\max(|x| - \varrho, 0)$ is the soft-thresholding operator, and $\operatorname{sgn}(\cdot)$ is the sign function.

A5. Solution to (22) under the $L_{1/2}$ loss

When $f(\cdot)$ represents the $L_{1/2}$ loss, Func. (22) is re-formulated as:

$$\arg\min_{\mathbf{e}} \ ||W \odot \mathbf{e}||_{1/2} + \frac{\eta^t}{2}||\mathbf{e} - Q\mathbf{c}^t - F\mathbf{c}^t - P + \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}||_2^2 \tag{41}$$

where '$\odot$' indicates the Hadamard product.

Let $A = Q\mathbf{c}^t + F\mathbf{c}^t + P - \frac{\Upsilon^{\mathbf{e},t}}{\eta^t}$, then (41) could be solved by dealing with $3m$ minimization subproblems independently, whose objective functions have the same form as:

$$\arg\min_{\mathbf{e}_i} \ W_i\sqrt{|\mathbf{e}_i|} + \frac{\eta^t}{2}||\mathbf{e}_i - A_i||_2^2 \tag{42}$$

where $\{W_i, \mathbf{e}_i, A_i\}$ are the i-th element in $\{W, \mathbf{e}, A\}$ respectively, and $W_i \in \{0, 1\}$. The above problem could be solved under the following two cases respectively:

Case #1: When $\mathbf{e}_i \geq 0$, let $y = \sqrt{\mathbf{e}_i} \geq 0$, and we obtain the following objective function by substituting $\mathbf{e}_i$ with y^2 in (42):

$$g(y) = W_i y + \frac{\eta^t}{2}||y^2 - A_i||_2^2 \tag{43}$$

Accordingly, the first and second derivatives of $g(y)$ are:

$$g'(y) = 2\eta^t y^3 - 2\eta^t A_i y + W_i \tag{44}$$

$$g''(y) = 6\eta^t y^2 - 2\eta^t A_i \tag{45}$$

(1.1) If $A_i \leq 0$, then it holds $g''(y) \geq 0$ according to (45) (η^t is a positive parameter), indicating that $g'(y)$ is nondecreasing for $y \in [0, +\infty)$. Furthermore, since $g'(0) = W_i \geq 0$, it holds $g'(y) \geq g'(0) \geq 0$ for $y \in [0, +\infty)$. Hence, $y = 0$ is the minimum solution to (43) for $y \in [0, +\infty)$ under this case ($A_i \leq 0$).

(1.2) If $A_i > 0$, then $y_0 = \sqrt{A_i/3} > 0$ is the solution to $g''(y) = 0$ for $y \in [0, +\infty)$. Due to the fact that $g'(0) = W_i \geq 0$ and $g'(y)$ is a cubic function, $g'(y)$ is nonincreasing for $y \in [0, y_0)$ and nondecreasing for $y \in [y_0, +\infty)$, indicating that the first derivative $g'(y)$ achieves the local minimum in $y \in [0, +\infty)$ at the point y_0, i.e.:

$$g'(y_0) = W_i - 4\eta^t\sqrt{(\frac{A_i}{3})^3} \tag{46}$$

- If $g'(y_0) \geq 0$ (i.e., $4\eta^t\sqrt{(\frac{A_i}{3})^3} \leq W_i$), then $g'(y) \geq g'(y_0) \geq 0$ for $y \in [0, +\infty)$, indicating that $y = 0$ is the minimum solution to (43) for $y \in [0, +\infty)$ under this case ($A_i > 0$).
- If $g'(y_0) < 0$ (i.e., $4\eta^t\sqrt{(\frac{A_i}{3})^3} > W_i$), there exist the following three solutions $\{y_1, y_2, y_3\}$ to $g'(y) = 0$:

$$y_1 = 2\sqrt{\frac{A_i}{3}}\cos\theta \tag{47}$$

$$y_2 = 2\sqrt{\frac{A_i}{3}}\cos(\theta + \frac{2\pi}{3}) \tag{48}$$

$$y_3 = 2\sqrt{\frac{A_i}{3}}\cos(\theta + \frac{4\pi}{3}) \tag{49}$$

where $\theta = \frac{1}{3}\arccos(-\frac{W_i}{4\eta^t\sqrt{(\frac{A_i}{3})^3}}) \in [0, \frac{\pi}{3}]$ (in fact, $\theta \in [\frac{\pi}{6}, \frac{\pi}{3}]$ more strictly). Then, it holds that $\cos(\theta + \frac{2\pi}{3}) \leq \cos(\theta + \frac{4\pi}{3}) \leq \cos\theta$ and $\cos(\theta + \frac{2\pi}{3}) < 0$, and accordingly, it holds that $y_2 \leq y_3 \leq y_1$ and $y_2 < 0$. Furthermore, since $g(y)$ is a quartic function, $y = y_1$ is the local minimum solution to (43) for $y \in [0, +\infty)$ under this case ($A_i > 0$), indicating that either $g(y_1)$ or $g(0)$ is the minimum value of $g(y)$ for $y \in [0, +\infty)$. This means that if $W_i y_1 + \frac{\eta^t}{2}(y_1^2 - A_i)^2 < \frac{\eta^t}{2}A_i^2$ (i.e., $g(y_1) < g(0)$), $y = y_1$ is the minimum solution to (43) for $y \in [0, +\infty)$; otherwise, $y = 0$ is the minimum solution.

The above proofs show that under Case #1 (i.e., $\mathbf{e}_i \geq 0$), if $W_i y_1 + \frac{\eta^t}{2}(y_1^2 - A_i)^2 < \frac{\eta^t}{2}A_i^2$ and $4\eta^t\sqrt{(\frac{A_i}{3})^3} > W_i$, the minimum solution to (42) is $\mathbf{e}_i^{t+1} = y_1^2$, otherwise $\mathbf{e}_i^{t+1} = 0$.

Case #2: When $\mathbf{e}_i \leq 0$, let $\bar{\mathbf{e}}_i = -\mathbf{e}_i \geq 0$ and $\bar{A}_i = -A_i$, then (42) is re-formulated as:

$$\arg\min_{\bar{\mathbf{e}}_i} \ W_i\sqrt{|\bar{\mathbf{e}}_i|} + \frac{\eta^t}{2}||\bar{\mathbf{e}}_i - \bar{A}_i||_2^2 \tag{50}$$

It is noted that (50) has the same formulation and condition as (42) in the aforementioned Case #1. Hence, for the defined y_1 and θ in Case #1, their corresponding forms in Case #2 are $y_1 = 2\sqrt{-\frac{A_i}{3}}\cos\theta$ and $\theta = \frac{1}{3}\arccos(-\frac{W_i}{4\eta^t\sqrt{(-\frac{A_i}{3})^3}})$ with $A_i < 0$. Accordingly, in Case #2, if $W_i y_1 + \frac{\eta^t}{2}(y_1^2 + A_i)^2 < \frac{\eta^t}{2}A_i^2$ and $4\eta^t\sqrt{(-\frac{A_i}{3})^3} > W_i$, the minimum solution to (42) is $\mathbf{e}_i^{t+1} = -y_1^2$, otherwise $\mathbf{e}_i^{t+1} = 0$.

Considering both the above two cases, the minimum solution to (42) is:

$$\mathbf{e}_i^{t+1} = \begin{cases} B_i & \text{if } 4\eta^t\sqrt{(\frac{|A_i|}{3})^3} > W_i \\ & \text{and } W_i\sqrt{|B_i|} + \frac{\eta^t}{2}(B_i - A_i)^2 < \frac{\eta^t}{2}A_i^2 \\ 0 & \text{otherwise} \end{cases} \tag{51}$$

where $B_i = \frac{4}{3}A_i\cos^2\theta$, and $\theta = \frac{1}{3}\arccos(-\frac{W_i}{4\eta^t\sqrt{(\frac{|A_i|}{3})^3}})$

References

[1] Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2016)

[2] Chen, P.H., Yang, H.C., Chen, K.W., Chen, Y.S.: Mvsnet++: Learning depth-based attention pyramid features for multi-view stereo. IEEE Transactions on Image Processing **29** (2020) 7261–7273

[3] Dong, Q., Hong, W.: Latent-smoothness nonrigid structure from motion by revisiting multilinear factorization. IEEE Transactions on Cybernetics **49**(4) (2019) 3557–3570

[4] Cui, H., Tu, D., Tang, F., Xu, P., Liu, H., Shen, S.: Vidsfm: Robust and accurate structure-from-motion for monocular videos. IEEE Transactions on Image Processing **31** (2022) 2449–2462

[5] Wang, P., Shi, L., Chen, B., Hu, Z., Qiao, J., Dong, Q.: Pursuing 3d scene structures with optical satellite images from affine reconstruction to euclidean reconstruction. IEEE Transactions on Geoscience and Remote Sensing **60** (2022) 1–14

[6] Crandall, D.J., Owens, A., Snavely, N., Huttenlocher, D.: Discrete-continuous optimization for large-scale structure from motion. In: Proc. CVPR. (2011) 3001–3008

[7] Tron, R., Vidal, R.: Distributed 3-d localization of camera sensor networks from 2-d image measurements. IEEE Transactions on Automatic Control **59**(12) (2014) 3325–3340

[8] Wilson, K., Snavely, N.: Robust global translations with 1dsfm. In: Proc. ECCV Part III. (2014) 61–75

[9] Özyesil, O., Singer, A.: Robust camera location estimation by convex programming. In: Proc. CVPR. (2015) 2674–2683

[10] Cui, Z., Tan, P.: Global structure-from-motion by similarity averaging. In: Proc. ICCV. (2015) 864–872

[11] Purkait, P., Chin, T.J., Reid, I.: Neurora: Neural robust rotation averaging. In: Proc. European Conference on Computer Vision. (2020) 137–154

[12] Chen, Y., Zhao, J., Kneip, L.: Hybrid rotation averaging: A fast and robust rotation averaging approach. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2021) 10353–10362

[13] Lee, S.H., Civera, J.: HARA: A hierarchical approach for robust rotation averaging. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2022) 15777–15786

[14] Gao, X., Zhu, L., Fan, B., Liu, H., Shen, S.: Incremental translation averaging. IEEE Transactions on Circuits and Systems for Video Technology **32**(11) (2022) 7783–7795

[15] Nister, D.: An efficient solution to the five-point relative pose problem. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(6) (2004) 756–770

[16] Melekhov, I., Ylioinas, J., Kannala, J., Rahtu, E.: Relative camera pose estimation using convolutional neural networks. In: Proc. International Conference on Advanced Concepts for Intelligent Vision Systems. (2017) 675–687

[17] En, S., Lechervy, A., Jurie, F.: Rpnet: An end-to-end network for relative camera pose estimation. In: Proc. European Conference on Computer Vision Workshops. (2018)

[18] Cai, R., Hariharan, B., Snavely, N., Averbuch-Elor, H.: Extreme rotation estimation using dense correlation volumes. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. (2021) 14566–14575

[19] Bustos, .P., Chin, T., Eriksson, A., Reid, I.: Visual slam: Why bundle adjust? In: Proc. International Conference on Robotics and Automation. (2019) 2385–2391

[20] Chng, C.K., Parra, A., Chin, T.J., Latif, Y.: Monocular rotational odometry with incremental rotation averaging and loop closure. In: Digital image computing: techniques and applications. (2020)

[21] Gao, X., Zhu, L., Xie, Z., Liu, H., Shen, S.: Incremental rotation averaging. International Journal of Computer Vision **129** (2021) 1202–1216

[22] Hartley, R., Aftab, K., Trumpf, J.: L1 rotation averaging using the weiszfeld algorithm. In: Proc. CVPR. (2011)

[23] Fredriksson, J., Olsson, C.: Simultaneous multiple rotation averaging using lagrangian duality. In: Proc. ACCV. (2013) 245–258

[24] Hartley, R.I., Trumpf, J., Dai, Y., Li, H.: Rotation averaging. International Journal of Computer Vision **103**(3) (2013) 267–305

[25] Chatterjee, A., Govindu, V.M.: Efficient and robust large-scale rotation averaging. In: Proc. ICCV. (2013) 521–528

[26] Chatterjee, A., Govindu, V.: Robust relative rotation averaging. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4) (2018) 958–972

[27] Govindu, V.M.: Robustness in motion averaging. In: Proc. ACCV. (2006)

[28] Zach, C., Klopschitz, M., , Pollefeys, M.: Disambiguating visual relations using loop constraints. In: Proc. CVPR. (2010)

[29] Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: Proc. CVPR (1). (2004) 684–691

[30] Eriksson, A., Olsson, C., Kahl, F., Chin, T.J.: Rotation averaging with the chordal distance: Global minimizers and strong duality. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(1) (2021) 256–268

[31] Dellaert, F., Rosen, D.M., Wu, J., Mahony, R., Carlone, L.: Shonan rotation averaging: Global optimality by surfing $\mathrm{SO}(p)^n$. In: Proc. European Conference on Computer Vision. (2020)

[32] Yang, H., Carlone, L.: Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(3) (2023) 2816–2834

[33] Arrigoni, F., Rossi, B., Fragneto, P., Fusiello, A.: Robust synchronization in SO(3) and SE(3) via low-rank and sparse matrix decomposition. Computer Vision and Image Understanding **174** (2018) 95–113

[34] Boumal, N., Singer, A., Absil, P., Blondel, V.: Cramer-rao bounds for synchronization of rotations. Journal of the IMA (2013)

[35] Courant, R., Hilbert, D.: Methods of mathematical physics. Wiley, New York (1989)

[36] Wu, F., Wang, Z.H., Hu, Z.: Cayley transformation and numerical stability of calibration equation. International Journal of Computer Vision **82**(2) (2009) 156–184

[37] Wu, F., Zhang, M., Hu, Z.: Self-calibration under the cayley framework. International Journal of Computer Vision **103**(3) (2013) 372–398

[38] Black, M., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. International Journal of Computer Vision **19** (1996) 57–91

[39] Lin, Z., Chen, M., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrix. Technical Report UILU-ENG-09-2215, UIUC (2009)

[40] Bertsekas, D.: Constrained Optimization and Lagrange Multiplier Method. Academic Press (1982)

[41] Yang, H., Antonante, P., Tzoumas, V., Carlone, L.: Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. IEEE Robotics and Automation Letters **5**(2) (2020) 1127–1134

[42] Hestenes, M.R.: Multiplier and gradient methods. Journal of Optimization Theory and Applications **4**(5) (1969) 303–320

[43] Yang, J., Zhang, Y.: Alternating direction algorithms for l_1-problems in compressive sensing. SIAM Journal on Scientific Computing **33**(1-2) (2011) 250–278

[44] Shen, Y., Wen, Z., Zhang, Y.: Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. Technical Report CAAM TR11-02, Rice University (2011)

[45] Eriksson, A., van den Hengel, A.: Efficient computation of robust low-rank matrix approximations in the presence of missing data using the L_1 norm. In: Proc. CVPR. (2010) 771–778

[46] Dong, Q., Gao, X., Cui, H., Hu, Z.: Robust camera translation estimation via rank enforcement. IEEE Transactions on Cybernetics **52**(2) (2022) 862–872