Synchronization of Projective Transformations

Rakshith Madhavan[1], Andrea Fusiello[2], and Federica Arrigoni[1]

[1] DEIB - Politecnico di Milano, Italy
[2] DPIA - University of Udine, Italy

Abstract. Synchronization involves the task of inferring unknown vertex values (belonging to a group) in a graph, from edges labeled with vertex relations. While many matrix groups (e.g., rotations or permutations) have received extensive attention in Computer Vision, a complete solution for projectivities is lacking. Only the 3×3 case has been addressed so far, by mapping the problem onto the Special Linear Group, but the 4×4 projective case has remained unexplored and is the focus here. We propose novel strategies to address this task, and demonstrate their effectiveness in synthetic experiments, as well as on an application to projective Structure from Motion.

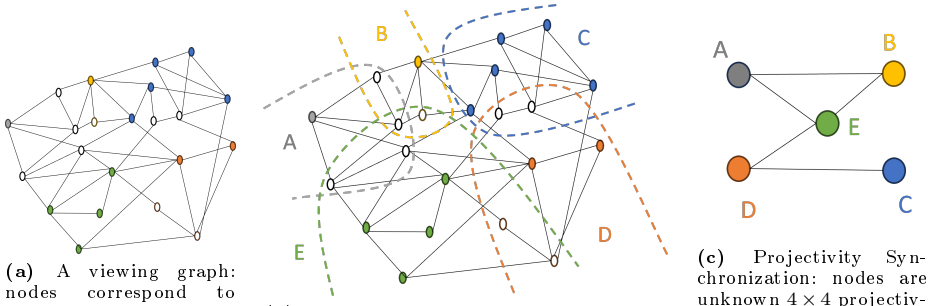
Keywords: Projective Structure from Motion · Multiview geometry · Synchronization · Projectivity · Homography

1 Introduction

A powerful tool for modelling and solving problems in Computer Vision is representing the associated entities (e.g., cameras or 3D point clouds) and their pairwise relations as a graph, while enforcing global consistency: a process also known as *synchronization* [4]. More precisely, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. Supposing each vertex is associated with an unknown value, and each edge with a measure of the ratio between the two incident vertices, and all these values belong to a group (e.g., the set of rotations or permutations): the task is to infer the unknown values associated with vertices in the graph. Assuming that such node values are denoted by $X_1, \ldots, X_n$ and Z_{ij} denotes the measure for edge (i, j), the consistency relation at the basis of synchronization can be stated as:

$$Z_{ij} = X_i X_j^{-1}. \tag{1}$$

The problem has been studied for several matrix groups representing relevant transformations in Computer Vision, including rotations, rigid transformations and permutations (see [4] for a review). The minimal case is represented by a spanning tree, in which case the solution can be found by sequential concatenation, thereby accumulating errors. In a general graph, instead, the idea is to exploit redundant measures. Typically, the discrepancy between the left and the right side in Eq. (1) is minimized with appropriate tools (see Sec. 2).



(a) A viewing graph: nodes correspond to cameras and edges to the fundamental matrices.

(b) The viewing graph is partitioned into sub-graphs. In each sub-graph projective SfM is solved: each reconstruction is up to an unknown 4x4 projective transformation.

(c) Projectivity Synchronization: nodes are unknown 4×4 projectivities and edges are the relative projectivities between partitions.

Fig. 1: Parititioned projective SfM and Projectivity Synchronization.

However, a general solution for the case of projectivities (also known as collinearities, homographies or projective transformations) is still missing. In this scenario, the consistency constraint takes the following form:

$$Z_{ij} \simeq X_i X_j^{-1} \tag{2}$$

where $\simeq$ denotes equality up to a scale. In addition to typical challenges of synchronization problems, such as the presence of outlier measures and missing edges in the graph, the case of projectivities presents additional complexity due to the scale ambiguity in Eq. 2. This problem has been studied only for 3×3 matrices and is solved satisfactorily in the Special Linear Group $SL(3)$ [55] where the projectivities are represented as matrices with unit determinant which has the effect of eliminating the scale ambiguity, i.e., turning Eq. (2) into a standard synchronization problem represented by Eq. (1). The case of 4×4 projective transformations has not been addressed yet and is explored here. It is worth mentioning that the trick used in [55] does not work for 4×4 matrices (see Sec. 3), demanding a new paradigm for projectivity synchronization. To fill in this gap, we propose a new iterative framework for synchronizing projective transformations in a graph. Inspired by [28], we propose to update each node value in turn as the average of its neighbors. Note that defining a proper average in projective space requires careful investigation. We analyze different approaches to accomplish such a task, namely: vectorizing projectivities and representing them as directions; modelling projective transformations as points in the unit sphere, analyzing both an instrinsic and an extrinsic solution.

Projectivity synchronization is related to projective (i.e., uncalibrated) structure from motion (SfM) [33,56], where the objective is to reconstruct the camera matrices and 3D points, up to a projectivity. This problem lends itself to insightful exploration when analyzed with the aid of the *viewing graph* [6, 37], where the nodes correspond to images or cameras, while the edges denote the geometric relationships (fundamental matrices) between them. Specifically, projectivity synchronization has applications in *partitioned* projective SfM, where the

viewing graph is partitioned into sub-graphs (for the sake of efficiency) where projective SfM is solved independently; therefore there are multiple independent reconstructions, each in different projective frames and synchronization of projectivities becomes relevant in order to merge them into a single consistent reconstruction (see Fig. 1). In this context, we have chosen to investigate a recent approach for projective structure-from-motion named GPSFM [33], which represents a specific instance of partitioned projective SfM where the viewing graph is partitioned into triplets. Such triplets are then stored in a *triplet graph*: a graph whose nodes are triplets of cameras, which are connected by edges if they share two cameras. The authors of GPSFM recover a globally consistent set of projection matrices by concatenating relative projective transformations along a spanning tree of this triplet graph. We incorporate our method into the GPSFM pipeline by replacing this spanning tree based approach, with our method described in Sec 4, and experimentally show that accuracy increases by exploiting the redundancies in the triplet graph.

To summarize, our contributions are three-fold:

- we introduce a new problem named projectivity synchronization;
- we define an effective iterative approach for such a task;
- we demonstrate the usefulness of our framework on synthetic graphs and real data in the context of projective structure from motion.

2 Related Work

Many synchronization methods have been proposed so far, which can be categorized based on the specific problem that is being addressed. We focus here on scenarios where the variables/unknowns can be represented as matrices.

Rotations. Several authors addressed synchronization of rotations [64], leveraging the rich mathematical structure of the rotation space (i.e., the Special Orthogonal Group) within the context of (calibrated) structure from motion [47]. Early works include approximate solutions where the entire rotation structure is neglected while taking into account only a subset of the constraints, in order to derive simplified formulations (linear least squares [25, 42], spectral decomposition [2, 59], semidefinite programming [2, 59], low-rank decomposition [7]). A different paradigm is employed in [28] where an iterative method is developed that updates each absolute rotation (i.e., node value) in turn based on its neighbors (i.e., measures on adjacent edges). Other authors focus on devising a suitable optimization objective, which is then minimized using (e.g.) the Levenberg-Marquardt algorithm [21] or Lie-group optimization [19, 58]. More recent works focus on global optimality [22, 23, 44, 49], including uncertainty into the model [12, 72] or how to produce a good initialization for nonlinear optimization [34]. Other authors also explored a deep learning framework either in a supervised fashion (using graph neural networks [38, 40, 51, 70]) or in an unsupervised way (in terms of deep matrix factorization [61]). Finally, there also exist notable works providing theoretical analysis [16, 29, 67, 68].

Rigid Transformations. Synchronization of rigid transformations (that form the Special Euclidean Group) has also received a lot of attention. Typical applications of this task are 3D point cloud registration [27] and simultaneous localization and mapping [18]. Based on the observation that a rigid transformation can be decomposed into rotation and translation, most works propose proper extensions of the methodologies and algorithms originally developed for rotations: spectral decomposition [3,8,24,31,66]; Lie-group optimization [11,26]; iterative update based on neighbors [62]; semidefinite programming [32,52,53,69]; modelling uncertainty [15,63]. A different paradigm (named group contraction) is used in [41, 46], where the d-dimensional group of rigid transformations is approximated with the group of rotations in $d+1$ dimensions.

Permutations. Another example of synchronization is the case of permutations, which form the so-called Symmetric Group, which are a convenient and compact way to represent matches (e.g., across multiple images). Several authors investigated this task, proposing solutions based on spectral decomposition [10, 48, 57], Gauss-Seidel relaxation [71], distributed optimization [35], Riemannian optimization [14] and quantum annealing [13]. Although synchronization has a well-established theory for the case where unknowns/measures belong to a group, specific routines can be developed when the variables belong to a weaker structure, as in the case of *partial* permutations, that form the so-called Symmetric Inverse Semigroup, which is used to model partiality (*i.e.*, missing correspondences) in the context of multi-view matching. Several works address this problem, including generalizations of the spectral solution [43], semidefinite programming [20], low-rank decomposition [36, 73], non-negative matrix factorization [9] and message-passing [39].

Other Scenarios. Other synchronization problems involve binary matrices in the context of motion segmentation [5, 30] and affine transformations with application to color correction [54]. The scenario of a multi-graph has also been studied in the literature [50, 60], namely the case where multiple measures are available between two nodes. Finally, it is worth mentioning the task of image mosaicking [55], which was cast to a synchronization problem by using 3×3 projective transformations, which in turn are converted into matrices with unit determinant: this represents the most related method to our approach, which will be reviewed in Sec. 3.

3 Problem Formulation and Motivation

Our objective is to address the problem of projectivity synchronization, namely, to solve $Z_{ij} \simeq X_i X_j^{-1}$ where $\simeq$ denotes equality up to the scale, $X_1, \ldots, X_n$ represent *unknown* $4{\times}4$ projectivities and Z_{ij} represents the *known* $4{\times}4$ projectivity associated to edge $(i,j) \in \mathcal{E}$. The available measures are therefore represented as edges in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes. All these projective transformations are invertible, i.e., they belong to the General Linear Group $GL(4)$.

In general, synchronization of $d \times d$ projective transformations can be cast to a synchronization in $SL(d)$ by normalizing the matrices to unit determinant. This works perfectly well for planar homographies (i.e., elements of $GL(3, \mathbb{R})$) because any real 3×3 matrix can be normalized to unit determinant by dividing it by the third root of its determinant, *which is always real*, thus ending up in $SL(3, \mathbb{R})$ [55]. Note that this operation is, in general, closed only in $\mathbb{C}$: in other words, for any dimension d, projectivity synchronization can be cast to a synchronization in $SL(d, \mathbb{C})$. Only for odd d this translates into a synchronization in $SL(d, \mathbb{R})$. When we consider, e.g., $d = 4$, then not all 4×4 matrices can be brought to $SL(4, \mathbb{R})$ by dividing by the 4-th root of the determinant, which will be complex, in general, so this normalization would bring to $SL(4, \mathbb{C})$. In addition, there are four possible roots which may be different in general.

In fact, as long as we get a final result that is real, intermediate results may be complex. So one can – in principle – run spectral synchronization [3] in $SL(4, \mathbb{C})$ and use a *complex* scale factor at the end to force matrices to be real. This is reminiscent of the projection step that is customarily taken when working with subgroups of $GL(d)$. We have empiric evidence that a *unique* complex scale exists that makes all the entries of the resulting matrices real, in a noiseless case. This property, however, gets disrupted by noise, and so this method is hardly applicable in practice (see experiments in Sec. 5.2).

4 Proposed Method

In this section we derive a novel approach for projectivity synchronization that is inspired by the iterative algorithm developed in [28] for rotations. The starting point is rewriting Eq. (2) as

$$X_i \simeq Z_{ij} X_j. \tag{3}$$

Such an equation means that – if all node values were known except for X_i – then we have *multiple* measures for X_i, one for each neighbor j connected to node i through an edge. Specifically, such measures are given by $Z_{ij} X_j$ for varying $j \in \mathcal{N}(i)$. We denote with $X_{i|j}$ the estimate of X_i given neighbor X_j, namely $X_{i|j} = Z_{ij} X_j$. Hence the idea is to update each unknown projectivity in turn as the "average" of its neighbors, where a suitable average operation has to be defined to manage the scale ambiguity inherent to the problem.

The scheme is fairly simple and works on a graph:

1. select a node i for updating;
2. compute new estimates of X_i given each neighbour X_j, namely:

$$X_{i|j} = Z_{ij} X_j; \tag{4}$$

3. average the $X_{i|j}$ over j to obtain the new estimate of X_i;
4. repeat from Step 1.

This procedure is repeated for all the nodes until convergence or a maximum number of iterations is reached. See Sec. 5.1 for implementation details. To

complete the algorithm for the synchronization of projectivities, we need to define a suitable "average" operation (Step 3), namely a method to compute a new estimate of X_i given the set of its neighbour estimates represented by $\{X_{i|j} \text{ for } j \in \mathcal{N}(i)\}$, that takes into account the scale ambiguity.

This is an instance of the *single averaging* problem, with reference to the terminology used in [29] for rotations. The task can be stated as follows in a general form for projectivities: given multiple[3] measures $H_1, \ldots, H_p \in GL(d)$ find a unique $C \in GL(d)$ (named *centroid*) that best satisfies $\forall\, k = 1, \ldots, p$:

$$H_k \simeq C \quad \Longleftrightarrow \quad \mathbf{h}_k \simeq \mathbf{c} \tag{5}$$

where $\mathbf{h}_k = \text{vec}(H_k)$ and $\mathbf{c} = \text{vec}(C)$. Several options for this operation are available, detailed in the next sections. Our proposals exploit different strategies for representing projective transformations and managing the scale ambiguity:

- by reasoning on the geometric meaning of parallel vectors, it is possible to derive a set of equations that is equivalent to (5) but with equality instead of equality up to scale, therefore resulting in a least-squares problem (Sec. 4.1);
- by representing vectorized projective transformations as points in the unit sphere, the single averaging problem is tantamount to finding the average on the unit sphere, for which an iterative approach can be derived (Sec. 4.2);
- with reference to the previous strategy, it is also possible to represent projectivities as points in the sphere, then ignore the sphere constraint and average these measures as standard points in 16 dimensions (Sec. 4.3).

4.1 Direction-based Averaging

Note that $\mathbf{h}_k \simeq \mathbf{c}$ – see Eq. (5) – means that the two vectors have the same direction. This can be rephrased into the property that the projection of $\mathbf{c}$ onto the orthogonal complement of $\mathbf{h}_k$ is zero. Hence:

$$\mathbf{h}_k \simeq \mathbf{c} \iff \left(I_{16} - \frac{\mathbf{h}_k \mathbf{h}_k^\top}{\mathbf{h}_k^\top \mathbf{h}_k} \right) \mathbf{c} = 0 \tag{6}$$

because the matrix between parentheses is the projector onto the the orthogonal complement of $\mathbf{h}_k$. Here I_{16} denotes the 16×16 identity matrix. Recall that $\mathbf{h}_k$ is the vectorized version of a 4×4 matrix, therefore it is a vector in 16 dimensions.

A solution can be found by stacking equations like (6) for $k = 1, \ldots, p$ and solving for $\mathbf{c}$ (up to a scale). Specifically, in the presence of noise the solution is computed in the least-squares sense, which can be derived as the eigenvector corresponding to the *minimum* eigenvalue of the following matrix:

$$\left[I_{16} - \frac{\mathbf{h}_1 \mathbf{h}_1^\top}{\mathbf{h}_1^\top \mathbf{h}_1}, \ldots, I_{16} - \frac{\mathbf{h}_p \mathbf{h}_p^\top}{\mathbf{h}_p^\top \mathbf{h}_p} \right] \begin{bmatrix} I_{16} - \frac{\mathbf{h}_1 \mathbf{h}_1^\top}{\mathbf{h}_1^\top \mathbf{h}_1} \\ \vdots \\ I_{16} - \frac{\mathbf{h}_p \mathbf{h}_p^\top}{\mathbf{h}_p^\top \mathbf{h}_p} \end{bmatrix} = p I_{16} - \sum_{ik1}^{p} \frac{\mathbf{h}_k \mathbf{h}_k^\top}{\mathbf{h}_k^\top \mathbf{h}_k} \tag{7}$$

[3] With reference to projectivity synchronization, the number of measures used for updating a node X_i is equal to the degree of node i, that is the number of edges having such a node as endpoint.

which is the eigenvector[4] corresponding to the *maximum* eigenvalue of the 16×16 matrix $\sum_{k=1}^{p}\frac{\mathbf{h}_k\mathbf{h}_k^\top}{\mathbf{h}_k^\top\mathbf{h}_k}$. Observe that this approach works with directions, i.e., it is independent on the scale of a particular $\mathbf{h}_k$. Note also that antipodal points are seamlessly dealt with, since the dyad $\mathbf{h}_k\mathbf{h}_k^\top$ does not depends on the sign of $\mathbf{h}_k$.

4.2 Spherical Averaging

Instead of using a scale-invariant approach, as done in the previous section, another strategy is explored here. The idea is to represent the measures to be averaged as points on the unit sphere. Let $\mathcal{S}=\{\mathbf{x}\in\mathbb{R}^{16}$ such that $\|\mathbf{x}\|=1\}$ be the unit sphere, let $\gamma(\mathbf{x},\mathbf{y})=\arccos(\mathbf{x}^\top\mathbf{y})$ denote the geodesic distance on the sphere, and let $\bar{\mathbf{h}}_k\in\mathcal{S}$ be the unit-norm representative of $\mathbf{h}_k$, i.e., $\bar{\mathbf{h}}_k=\mathbf{h}_k/||\mathbf{h}_k||$. In order to be consistent with this representation of projective transformations, we need to identify antipodal points. This can be done by selecting one of the $\bar{\mathbf{h}}_k$ (e.g., the first) and changing the sign of the others such that the scalar product with the first is positive, bringing all of them to the same half-space.

The *L1 geodesic mean* is defined as the point $\mathbf{c}$ of minimum average geodesic distance (it generalizes the concept of median), and can be found using the method of Lagrange multipliers. Specifically, the task is to minimize the average distance to the sought centroid $\mathbf{c}$, which is given by $\frac{1}{p}\sum_{k=1}^{p}\arccos\left(\mathbf{c}^\top\bar{\mathbf{h}}_k\right)$. The Lagrangian that constrains $\mathbf{c}$ to the unit sphere is given by

$$L(\mathbf{c},\lambda)=\frac{1}{p}\sum_{k=1}^{p}\arccos\left(\mathbf{c}^\top\bar{\mathbf{h}}_k\right)+\lambda(1-\mathbf{c}^\top\mathbf{c}) \tag{8}$$

and the partial derivatives are:

$$\frac{\partial L}{\partial c_h}(\mathbf{c},\lambda)=-\sum_{k=1}^{p}\frac{\bar{h}_{hk}}{\sqrt{1-(\mathbf{c}^\top\mathbf{h}_k)^2}}-2\lambda c_h,\quad \frac{\partial L}{\partial\lambda}(\mathbf{c},\lambda)\quad=1-\mathbf{c}^\top\mathbf{c} \tag{9}$$

where $\bar{h}_{hk}$ is the h-th coordinate of the k-th point. By zeroing these partial derivatives we get:

$$\mathbf{c}=\alpha\sum_{i=1}^{p}\frac{\bar{\mathbf{h}}_k}{\sqrt{1-(\mathbf{c}^\top\bar{\mathbf{h}}_k)^2}} \tag{10}$$

where α is a normalizing constant so that $\mathbf{c}$ lies on the unit sphere. The sought average $\mathbf{c}$ appears on both sides of the equation, so it is defined as a fixed point, which can be estimated iteratively.

It is worth mentioning that spherical averaging is a well-studied topic that counts notable works (such as [17]). We tested [17] and obtained comparable results with respect to the method of Lagrangian multipliers described above, therefore it is not included in our experiments.

[4] It is easy to see that A and $A-\alpha I$ for any scalar α have the same eigenvectors. Moreover, $\lambda(A)=-\lambda(-A)$.

4.3 Euclidean Averaging

Ignoring that the result of averaging unit vectors must be intrinsic (i.e. a unit vector in turn), leads to a simpler solution. In other terms, we require the average $\mathbf{c}$ to minimize the L1 distance in the Euclidean space (instead of geodesic) and we do not impose any constraint on the result, so the objective becomes:

$$\frac{1}{p}\sum_{k=1}^{p}\left\|\bar{\mathbf{h}}_k - \mathbf{c}\right\|. \tag{11}$$

A solution to this L1 optimization problem can be found using the classical Weiszfeld algorithm [1]. The result – in general – will not belong to $\mathcal{S}$, but can be simply brought to the unit sphere by normalization at the end.

5 Experiments

In this section we report results on synthetic/real data and we also discuss limitations of our approach. We ran all experiments on a Lenovo Legion 5 laptop, with an AMD Ryzen 5 4600H processor and an 8GB RAM. The synchronization framework and the synthetic experiments are implemented in Julia[5], while the experiments with the real datasets are in MATLAB, with a *system* call to Julia. We analyzed the performance of our synchronization framework combined with the three averaging methods discussed in Sec. 4, resulting in three variants of our approach (named Direction, Sphere, Euclidean). Since our work is the first that explicitly addresses synchronization of 4×4 projective transformations, there are no competitors in the literature, apart from the spanning tree solution (named Tree hereafter), that was used in [33] in the context of structure from motion. In order to enrich the evaluation, we include an additional method in the comparison, namely an extension of the Spectral method [55], that was developed for 3×3 projectivities, to manage 4×4 matrices (see Sec. 3).

Observe that the solution to projectivity synchronization is defined up to a global transformation. Therefore, in order to compare a solution (denoted by $\hat{X}_1,\ldots,\hat{X}_n$) with the ground-truth node projectivities (denoted by $X_1,\ldots,X_n$), we have to find a homography $C\in GL(4)$ such that $\hat{X}_iC\simeq X_i$ for all $i=1,\ldots,n$. Note that this is an averaging problem, with measures given by $H_k=\hat{X}_k^{-1}X_k$, therefore it can be solved with an averaging method from Sec. 4. Then, the error for node i is measured as the angular distance between the vectorized estimated and ground truth projectivities, brought to a common projective frame, namely:

$$e_i=\min(\phi,\pi-\phi)\quad\text{where}\quad\phi=\arccos\big(\mathrm{vec}(\hat{X}_iC),\mathrm{vec}(X_i)\big). \tag{12}$$

5.1 Implementation details

We now specify some implementation details of our approach. The nodes are initialized to $I\in\mathbb{R}^{4\times4}$, and updated in descending order of their degree (i.e.,

[5] `https://github.com/rakshith95/projective_synchronization.jl`

the number of incident edges), based on the idea that the vertex with the highest degree is the most stable and hence a good candidate to start the procedure.

Concerning robustness to outliers[6], we propose to embed our synchronization framework into an IRLS-like scheme, motivated by the fact that adapting Iteratively Reweighted Least Squares (IRLS) to synchronization is a popular choice also in other scenarios [3,19]. The idea is to assign a weight w_{ij} to each input transformation Z_{ij}. It is easy to see that all averaging strategies from Sec. 4 can be straightforwardly extended to manage *weighted* measures. At the beginning, all such weights are set to one. Then, after a complete round of projectivity synchronization, such weights are updated as follows: for each edge, we compute the residual between the obtained node transformations and the input relative one, namely $r_{ij} = error(Z_{ij}, X_i X_j^{-1})$; the error can be computed as the angle in $\mathbb{R}^{16}$ between the vectorized matrices, similar to Eq. (12); then, we set $w_{ij} = f\big(\min(r_{ij}, r_{ji})\big)$ where f is a robust loss function (we used Cauchy in our experiments) and the term $\min(r_{ij}, r_{ji})$ is computed to guarantee symmetric weights in the graph. Hence we alternate between solving projectivity synchronization in a weighted graph and updating the weights.

5.2 Synthetic Data

Given a value n for the number of nodes in the graph, we set up the synthetic environment by generating random ground-truth projective transformations $X_i \in GL(4)$ for $i = 1, \dots n$. Concerning the edge set $\mathcal{E}$ in the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we remove from the complete graph a random fraction of edges according to a "holes density" $0 \leq \rho < 1$, while ensuring connectivity ($\rho = 0$ produces a complete graph). For each edge $(i, j) \in \mathcal{E}$, we compute its measure as $Z_{ij} = X_i X_j^{-1}$, which is then normalized and perturbed by noise. Specifically, noise is introduced by perturbing $\mathrm{vec}(Z_{ij})$ by some angle θ_{ij}. This is done in the space of the hyperplane tangent to $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^{16} \,|\, \|\mathbf{x}\| = 1\}$ at the point $\bar{\mathbf{z}}_{ij} = \mathrm{vec}(Z_{ij})/||\,\mathrm{vec}(Z_{ij})||$. In this tangent space, a vector $\mathbf{t}_{ij}$ is generated such that $\|\mathbf{t}_{ij}\| = \theta$, which is then projected back to $\mathcal{S}$ with the exponential map [17]. Outliers are added by replacing Z_{ij} with a random $R_{ij} \in GL(4)$, with the fraction of outlying edges determined by an outlier density γ. The edge measures are given as input to all the analyzed methods. The IRLS scheme is used only in experiments with outliers (i.e., $\gamma > 0$). For each configuration, the test was repeated 1000 times and median results were reported.

First, we test the effects of varying the number of nodes n, while keeping the other parameters fixed ($\theta = 0.1$ rad $= 5.73°, \rho = 0.5, \gamma = 0.0$). Results are shown in Fig. 2, reporting both the errors and execution times of the analyzed methods. We can observe that, by increasing the number of nodes, the performance of all methods except Tree improve, since there is more redundancy in the data. The

[6] It is worth observing that both Sphere and Euclidean are robust to outliers (being based on the L1 norm), whereas Direction is not (being based on the L2 norm). Therefore, for the first two methods, the IRLS-like scheme has the effect of *improving* robustness, whereas for the third approach it has the effect of *gaining* robustness.

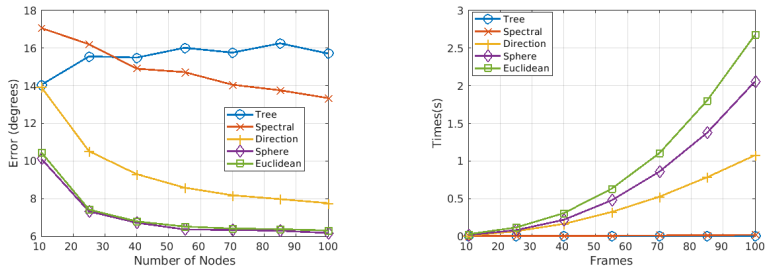


Fig. 2: Error (left) and execution times (right) for the analyzed methods with respect to varying number of nodes in the graph, with $\theta = 5.73°, \rho = 0.5, \gamma = 0.0$.

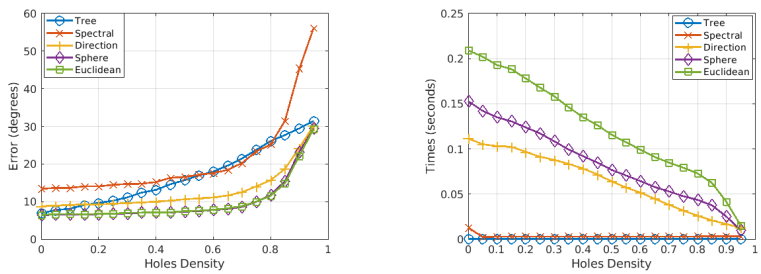


Fig. 3: Error (left) and execution times (right) for the analyzed methods with respect to varying holes density in the graph, with $n = 25$, $\theta = 5.73°, \gamma = 0.0$.

bad behaviour of the spanning tree approach is caused by the fact that it chains relative transformation from the root to the leaves, resulting in increased error accumulation for a bigger graph. In addition, we can observe that our framework is significantly better than Spectral, therefore justifying the need of a specific approach for synchronization of 4×4 projective transformations – developed in this paper – as anticipated in Sec. 3. Among the different variants of our method, the best are Sphere and Euclidean, whereas Direction is slightly worse, probably due to the fact that the latter is not robust in the averaging phase. From the execution times in Fig. 2, as we would expect, the global methods (Tree and Spectral) are significantly faster than iterative methods (Direction, Sphere and Euclidean). Direction performs the fastest out of the latter, since the average is computed with a closed-form expression, while in Sphere and Euclidean the average is obtained iteratively.

In a second experiment, we analyze the behaviour of the competing methods with respect to varying the holes density ρ (with $n = 25, \theta = 5.73°, \gamma = 0.0$). Results are given in Fig. 3, showing that the accuracy of our synchronization methods get worse with increasing holes density, even reaching Tree on the left-most scenario with 0.95 fraction of missing edges in the graph. This is an expected behaviour, since synchronization naturally exploits redundant measures to achieve error compensation. It is worth noting that Spectral is even worse

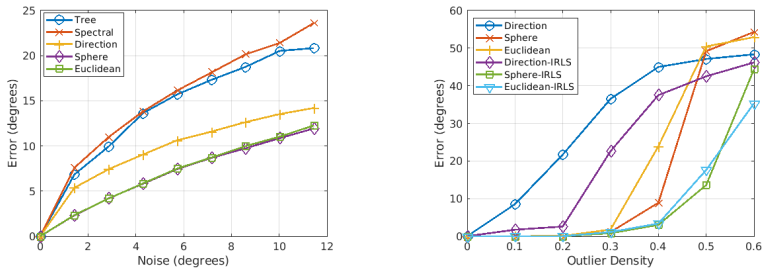


Fig. 4: Error for the analyzed methods with respect to varying noise (left) and outliers (right), with $n = 25$ and $\rho = 0.5$. In the left experiment there are no outliers ($\gamma = 0.0$) whereas in the right experiment there is no noise ($\theta = 0°$).

than the spanning tree in most cases, further validating the intuition that it does not represent a valid solution to projectivity synchronization. The most accurate approaches are Sphere and Euclidean, as before. The global methods are the fastest, while the iterative ones start orders of magnitude slower, but approach the execution times of the global ones, as ρ increases (making the graph sparser).

In a third experiment, we test the effects of varying noise θ with $n = 25, \rho = 0.5, \gamma = 0.0$. Results are given in Fig. 4 (left), showing that all methods become less accurate as the amount of noise increases, to different extents. Specifically, both Spectral and Tree are significantly worse than our synchronization approach, and the best methods are Sphere and Euclidean (without significative differences), in agreement with our previous experiments.

In our last experiment, we test the effect of varying outlier density γ, with $n = 25, \theta = 0.0, \rho = 0.5$. In this scenario we do not consider Spectral and Tree, since they have been shown to be inferior to our approach in the previous experiments. Specifically, for each variant of our approach, we consider both the plain method (without IRLS) and the one endowed with the IRLS scheme. Results are given in Fig. 4 (right), showing the effectiveness of the IRLS approach to manage outliers in the data. Note that Direction achieves the worst results since it uses the L2 norm (which is not robust) in the averaging step, as already observed previously; the IRLS scheme improves its robustness, but still it is not comparable to methods that, by employing the L1 norm, are robust *also* in the averaging phase (namely Sphere and Euclidean).

From the results of the synthetic experiments, we choose Sphere with the IRLS-like scheme for the experiments with real data.

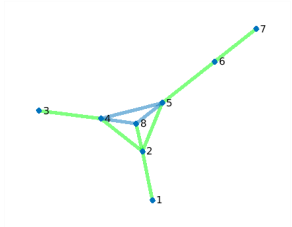
5.3 Application to Projective Structure from Motion

In order to show the practical benefits of projectivity synchronization, we consider the scenario of projective structure from motion, with particular focus on a method – named Gpsfm [33]– that recovers cameras from fundamental matrices (represented as a viewing graph). Gpsfm employs several steps:

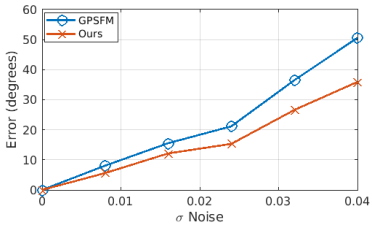
1. A *triplet graph* is constructed from the viewing graph: each node in the triplet graph corresponds to a triplet of cameras and an edge is present between two nodes if the triplets share a pair of cameras;
2. Fundamental matrices are globally optimized by enforcing consistency over triplets; then, cameras are estimated independently for each triplet;
3. Cameras are brought to a common projective frame as follows; a tree in the triplet graph is constructed and a homography is found for each edge in the tree, such that the common camera pair in the two triplets are aligned; such projective transformations are then concatenated along the tree.

With reference to Step 2, note that *in theory* consistency over triplets implies global consistency (see Theorem 2 in [33]). However, *in practice* such consistency is not completely enforced, as the ADMM method (which is used for such a task) employs soft constraints. For this reason, cameras estimated over different triplets may not be coherent, demanding a principled approach to compensate for these errors. In this context, it is worth observing that Step 3 is tantamount to solving a projectivity synchronization problem defined on the triplet graph. Specifically, the method employed by the authors of [33] coincides with Tree. We propose to replace this step, with our proposed pipeline with the Sphere averaging combined with IRLS (see Fig. 5a for an example of a triplet graph: Gpsfm uses only minimal edge measures, whereas our method would additionally exploit available redundancies).

We first consider a synthetic scenario where a set of noisy fundamental matrices are given as input. Specifically, we consider a full viewing graph and generate the data as follows. First, n finite 3×4 cameras P_i for $i = 1, \ldots, n$ are randomly generated. For each pair (i, j) of cameras such that $j > i$, the corresponding fundamental matrix is created as $F_{ij} = [\mathbf{e}_{ij}]_\times P_j P_i^+$, where P_i^+ denotes the pesudo-inverse, $\mathbf{e}_{ij}$ represents the epipole (i.e., $\mathbf{e}_{ij} = P_j \mathbf{c}_i$ with $P_i \mathbf{c}_i = \mathbf{0}$), and $[\mathbf{e}_{ij}]_\times$ denotes the skew-symmetric matrix that defines the cross-product. Then, we add Gaussian noise $\mathcal{N}(0, \sigma)$ to the entries of F_{ij} and project it to its rank-2



(a) A triplet graph from the *House* dataset [45]. The edge values used by Gpsfm are in Green, ignoring the ones in Blue, while our method uses all edges.



(b) Error for the analyzed methods in camera recovery in a complete viewing graph with 20 nodes, with respect to varying noise in the input fundamental matrices.

Fig. 5: Left: example of a triplet graph constructed by Gpsfm [33]. Right: synthetic experiments on partitioned projective SfM.

Table 1: Results on real structure from motion data [45, 65]. For each sequence, the number of nodes in the triplet graph is reported in addition to its holes density. The reprojection error [pixels] before bundle adjustment is reported for the competing methods. Times [seconds] refers only to the camera recovery step after optimization.

Dataset	Nodes	Density	Error (px)		Time (s)	
			Ours	GPSFM	Ours	GPSFM
Dino 319	34	0.94	**4.89**	5.21	10.2	0.12
Dino 4983	34	0.93	**1.59**	**1.59**	11.0	0.12
Corridor	9	0.75	**0.67**	0.71	10.2	0.03
House	8	0.64	**1.40**	1.68	10.3	0.03
Gustav Vasa	17	0.88	**1.99**	**1.99**	10.2	0.06
Folke Filbyter	38	0.94	**1.83**	1.96	10.7	0.14
Park Gate	37	0.94	**13.70**	18.45	10.8	0.15
Nijo	18	0.88	**25.83**	29.85	10.7	0.08
Drinking Fountain	13	0.82	**1.34**	1.54	10.5	0.04
Golden Statue	16	0.85	**0.90**	0.95	10.7	0.05
Jonas Ahls	39	0.94	**33.06**	36.83	10.5	0.13
De Guerre	35	0.93	1.36	**1.31**	10.4	0.12
Dome	90	0.97	6.41	**4.17**	11.5	0.31
Alcatraz Courtyard	140	0.98	**14.96**	36.73	11.2	0.48
Alcatraz Water Tower	188	0.99	19.72	**19.10**	12.1	0.7
Cherub	66	0.96	**14.98**	16.87	10.3	0.25
Pumpkin	222	0.99	**9.27**	9.57	13.4	0.94
Sphinx	78	0.97	**6.16**	8.46	11.4	0.27
Toronto University	80	0.97	20.10	**13.85**	10.5	0.3
Sri Thendayuthapani	104	0.98	16.09	**14.55**	10.9	0.39
Porta San Donato	158	0.99	41.22	**32.41**	11.1	0.57
Buddah Tooth	172	0.99	**19.07**	25.49	11.9	0.67
Tsar Nikolai I	104	0.98	15.26	**9.58**	10.4	0.37
Smolny Cathedral	143	0.98	182.83	**122.82**	11.8	0.55
Skansen Kronan	141	0.98	**9.75**	13.71	11.5	0.52

approximation. For $j < i$ we set $F_{ij} = F_{ji}^{\mathsf{T}}$. We consider $n = 20$ and evaluate the effects of increasing noise σ in Fig. 5b. The error between a solution and the ground-truth is evaluated as the angle between the vectorized cameras, after bringing them to a common projective frame, similarly to Eq. (12). Results show that our synchronization framework outperforms GPSFM thanks to the fact that it considers the entire triplet graph for error compensation, in agreement with the outcome of our previous experiments.

For the experiments with real structure-from-motion data, we test on the same datasets used by the authors of GPSFM [33]: a total of 25 image collections from [45, 65][7]; for each dataset, the authors of [33] have made available the fundamental matrices. We compare the mean reprojection error (in pixels)

[7] The datasets can be downloaded from `https://www.maths.lth.se/matematiklth/personal/calle/dataset/dataset.html`

across all scene points, before Bundle Adjustment. Results are given in Tab. 1, which also reports the execution times of the competing methods, confirming the outcome of our previous experiments. Despite the sparsity of the triplet graphs, we are able to reduce the reprojection error in most of the datasets (15 out of the 25) using SPHERE in the IRLS-like scheme, in comparison to the method used in [33]. In a few datasets, we see worse errors, which can be attributed to the presence of outliers in graphs with high holes density ($0.97 < \rho < 1.0$) where there are fewer redundancies to mitigate their influence; note also that the high reprojection error achieved by both methods on these cases, confirms the difficulty of those sequences in terms of noise and outlier level. We do not report results after Bundle Adjustment (BA), since in almost all cases the two methods converge to the same values; the effect of the projectivity synchronization, in this context, is in providing better starting estimates for the BA.

5.4 Limitations

The main limitation of our synchronization framework is that it requires redundant measures to work well in practice. Performance might degrade significantly in the case of minimal data, especially if corrupted by outliers. In this respect, the choice of the input graph might be critical in real scenarios. This is in line with other synchronization tasks [4]. In addition, we do not, currently, claim a proof on the local/global convergence of our iterative scheme.

6 Conclusion

In this paper we considered the problem of projectivity synchronization with focus on 4×4 matrices: considering a graph, the task is to find unknown projective transformations (associated with nodes), starting from measures (up to scale) of their pairwise ratios (associated with edges). Previous solutions are either specific for 3×3 matrices [55] (and do not generalize to the 4×4 case) or do not exploit the redundancy in the data [33]. Motivated by this, we presented a novel framework to address the challenges of projectivity synchronization, based on *scale-invariant* averaging of convenient representations of projective transformations. We showed the effectiveness of our iterative method on a variety of synthetic scenarios as well as on experiments using real datasets. For the latter, we embedded our method into a pipeline for *partitioned* projective structure from motion that recovers cameras from fundamental matrices by partitioning the viewing graph, computing a reconstruction for each partition independently, and finally integrating them in a globally consistent reconstruction. We hope that our preliminary results can serve as the basis for further research in projectivity synchronization, both theoretical and in terms of new application scenarios.

Acknowledgements. This paper is supported by PNRR-PE-AI FAIR project funded by the NextGeneration EU program. The authors would like to thank Gaia Trebucchi for her help on a preliminary project related to this research.

References

1. Aftab, K., Hartley, R., Trumpf, J.: Generalized Weiszfeld algorithms for l_q optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **4**(37), 728 – 745 (2015)
2. Arie-Nachimson, M., Kovalsky, S.Z., Kemelmacher-Shlizerman, I., Singer, A., Basri, R.: Global motion estimation from point matches. Proceedings of the Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission (2012)
3. Arrigoni, F., Rossi, B., Fusiello, A.: Spectral synchronization of multiple views in SE(3). SIAM Journal on Imaging Sciences **9**(4), 1963 – 1990 (2016)
4. Arrigoni, F., Fusiello, A.: Synchronization problems in computer vision with closed-form solutions. International Journal of Computer Vision **128**, 26–52 (2020)
5. Arrigoni, F., Pajdla, T.: Motion segmentation via synchronization. In: IEEE International Conference on Computer Vision Workshops (ICCVW) (2019)
6. Arrigoni, F., Pajdla, T., Fusiello, A.: Viewing graph solvability in practice. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8147–8155 (2023)
7. 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**, 95–113 (2018)
8. Bernard, F., Thunberg, J., Gemmar, P., Hertel, F., Husch, A., Goncalves, J.: A solution for multi-alignment by transformation synchronisation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
9. Bernard, F., Thunberg, J., Goncalves, J., Theobalt, C.: Synchronisation of Partial Multi-Matchings via Non-negative Factorisations. Pattern Recognition **92**, 146 – 155 (2019)
10. Bernard, F., Thunberg, J., Swoboda, P., Theobalt, C.: HiPPI: Higher-order projected power iterations for scalable multi-matching. In: Proceedings of the International Conference on Computer Vision (2019)
11. Bhattacharya, U., Govindu, V.M.: Efficient and robust registration on the 3d special euclidean group. In: Proceedings of the International Conference on Computer Vision (2019)
12. Birdal, T., Arbel, M., Simsekli, U., Guibas, L.J.: Synchronizing probability measures on rotations via optimal transport. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1566–1576 (2020)
13. Birdal, T., Golyanik, V., Theobalt, C., Guibas, L.: Quantum permutation synchronization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
14. Birdal, T., Simsekli, U.: Probabilistic permutation synchronization using the riemannian structure of the birkhoff polytope. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11105–11116 (2019)
15. Birdal, T., Simsekli, U., Eken, M.O., Ilic, S.: Bayesian pose graph optimization via bingham distributions and tempered geodesic mcmc. In: Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
16. Boumal, N., Singer, A., Absil, P.A., Blondel, V.D.: Cramer-Rao bounds for synchronization of rotations. Information and Inference: A Journal of the IMA **3**(1), 1 – 39 (2014)
17. Buss, S.R., Fillmore, J.P.: Spherical averages and applications to spherical splines and interpolation. ACM Trans. Graph. **20**(2), 95–126 (2001)

18. Carlone, L., Tron, R., Daniilidis, K., Dellaert, F.: Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In: Proceedings of the IEEE International Conference on Robotics and Automation (2015)
19. Chatterjee, A., Govindu, V.M.: Efficient and robust large-scale rotation averaging. In: Proceedings of the International Conference on Computer Vision (2013)
20. Chen, Y., Guibas, L., Huang, Q.: Near-optimal joint object matching via convex relaxation. In: Proceedings of the International Conference on Machine Learning. pp. 100–108 (2014)
21. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.P.: Discrete-continuous optimization for large-scale structure from motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3001–3008 (2011)
22. Dellaert, F., Rosen, D.M., Wu, J., Mahony, R., Carlone, L.: Shonan rotation averaging: Global optimality by surfing $so(p)^n$. In: Computer Vision – ECCV 2020. pp. 292–308. Springer International Publishing (2020)
23. Eriksson, A., Olsson, C., Kahl, F., Chin, T.J.: Rotation averaging and strong duality. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 127–135 (2018)
24. Gojcic, Z., Zhou, C., Wegner, J.D., Guibas, L.J., Birdal, T.: Learning multiview 3D point cloud registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
25. Govindu, V.M.: Combining two-view constraints for motion estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2001)
26. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 684–691 (2004)
27. Govindu, V.M., Pooja, A.: On averaging multiview relations for 3D scan registration. IEEE Transactions on Image Processing **23**(3), 1289–1302 (2014)
28. Hartley, R., Aftab, K., Trumpf, J.: L1 rotation averaging using the Weiszfeld algorithm. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp. 3041–3048 (2011)
29. Hartley, R.I., Trumpf, J., Dai, Y., Li, H.: Rotation averaging. International Journal of Computer Vision (2013)
30. Huang, J., Wang, H., Birdal, T., Sung, M., Arrigoni, F., Hu, S.M., Guibas, L.J.: Multibodysync: Multi-body segmentation and motion estimation via 3d scan synchronization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7108–7118 (2021)
31. Huang, X., Liang, Z., Zhou, X., Xie, Y., Guibas, L.J., Huang, Q.: Learning transformation synchronization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2019)
32. Iglesias, J.P., Olsson, C., Kahl, F.: Global optimality for point set registration using semidefinite programming. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
33. Kasten, Y., Geifman, A., Galun, M., Basri, R.: GPSfM: Global projective SFM using algebraic constraints on multi-view fundamental matrices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3259–3267 (2019)
34. Lee, S.H., Civera, J.: Hara: A hierarchical approach for robust rotation averaging. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)

35. Leonardos, S., Zhou, X., Daniilidis, K.: Distributed consistent data association via permutation synchronization. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 2645–2652 (2017)
36. Leonardos, S., Zhou, X., Daniilidis, K.: A low-rank matrix approximation approach to multiway matching with applications in multi-sensory data association. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 8665–8671 (2020)
37. Levi, N., Werman, M.: The viewing graph. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 518 – 522 (2003)
38. Li, H., Cui, Z., Liu, S., Tan, P.: Rago: Recurrent graph optimizer for multiple rotation averaging. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
39. Li, S., Shi, Y., Lerman, G.: Fast, accurate and memory-efficient partial permutation synchronization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022)
40. Li, X., Ling, H.: Pogo-net: Pose graph optimization with graph neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5895–5905 (October 2021)
41. Mankovich, N., Birdal, T.: Chordal averaging on flag manifolds and its applications. In: Proceedings of the International Conference on Computer Vision (2023)
42. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2007)
43. Maset, E., Arrigoni, F., Fusiello, A.: Practical and efficient multi-view matching. In: Proceedings of IEEE International Conference on Computer Vision. pp. 4568–4576 (2017)
44. Moreira, G., Marques, M., Costeira, J.a.P.: Rotation averaging in a split second: A primal-dual method and a closed-form for cycle graphs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5452–5460 (2021)
45. Olsson, C., Enqvist, O.: Stable structure from motion for unordered image collections. In: Proceedings of the 17th Scandinavian conference on Image analysis (SCIA'11). pp. 524–535. Springer-Verlag (2011)
46. Ozyesil, O., Sharon, N., Singer, A.: Synchronization over cartan motion groups via contraction. SIAM Journal on Applied Algebra and Geometry **2**(2), 207 – 241 (2018)
47. Ozyesil, O., Voroninski, V., Basri, R., Singer, A.: A survey of structure from motion. Acta Numerica **26**, 305 – 364 (2017)
48. Pachauri, D., Kondor, R., Singh, V.: Solving the multi-way matching problem by permutation synchronization. In: Advances in Neural Information Processing Systems, vol. 26, pp. 1860–1868 (2013)
49. Parra, A., Chng, S.F., Chin, T.J., Eriksson, A., Reid, I.: Rotation coordinate descent for fast globally optimal rotation averaging. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4298–4307 (2021)
50. Porfiri Dal Cin, A., Magri, L., Arrigoni, F., Fusiello, A., Boracchi, G.: Synchronization of group-labelled multi-graphs. In: IEEE International Conference on Computer Vision (ICCV) (2021)
51. Purkait, P., Chin, T.J., Reid, I.: Neurora: Neural robust rotation averaging. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 137–154. Springer International Publishing (2020)

52. Rosen, D.M., DuHadway, C., Leonard, J.J.: A convex relaxation for approximate global optimization in simultaneous localization and mapping. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 5822 – 5829 (2015)
53. Rosen, D.M., Carlone, L., Bandeira, A.S., Leonard, J.J.: Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. The International Journal of Robotics Research **38**, 95–125 (2019)
54. Santellani, E., Maset, E., Fusiello, A.: Seamless image mosaicking via synchronization. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences **IV-2**, 247–254 (2018)
55. Schroeder, P., Bartoli, A., Georgel, P., Navab, N.: Closed-form solutions to multiple-view homography estimation. In: IEEE Workshop on Applications of Computer Vision (WACV). pp. 650–657 (2011)
56. Sengupta, S., Amir, T., Galun, M., Goldstein, T., Jacobs, D.W., Singer, A., Basri, R.: A new rank constraint on multi-view fundamental matrices, and its application to camera location recovery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2413–2421 (2017)
57. Shen, Y., Huang, Q., Srebro, N., Sanghavi, S.: Normalized spectral map synchronization. In: Advances in Neural Information Processing Systems, vol. 29, pp. 4925–4933. Curran Associates, Inc. (2016)
58. Shi, Y., Lerman, G.: Message passing least squares framework and its application to rotation synchronization. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 8796–8806. PMLR (2020)
59. Singer, A.: Angular synchronization by eigenvectors and semidefinite programming. Applied and Computational Harmonic Analysis **30**(1), 20 – 36 (2011)
60. Sun, Y., Huang, Q.: Pose synchronization under multiple pair-wise relative poses. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2023)
61. Tejus, G., Zara, G., Rota, P., Fusiello, A., Ricci, E., Arrigoni, F.: Rotation synchronization via deep matrix factorization. In: 2023 IEEE International Conference on Robotics and Automation (ICRA) (2023)
62. Torsello, A., Rodolà, E., Albarelli, A.: Multiview registration via graph diffusion of dual quaternions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2441 – 2448 (2011)
63. Tron, R., Danilidis, K.: Statistical pose averaging with varying and non-isotropic covariances. In: Proceedings of the European Conference on Computer Vision (2014)
64. Tron, R., Zhou, X., Daniilidis, K.: A survey on rotation optimization in structure from motion. In: Computer Vision and Pattern Recognition Workshops (CVPRW) (2016)
65. Visual Geometry Group - University of Oxford: Multiview datasets, `https://www.robots.ox.ac.uk/~vgg/data/`
66. Wang, H., Liu, Y., Dong, Z., Guo, Y., Liu, Y.S., Wang, W., Yang, B.: Robust multiview point cloud registration with reliable pose graph initialization and history reweighting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
67. Wilson, K., Bindel, D.: On the distribution of minima in intrinsic-metric rotation averaging. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6030–6038 (2020)

68. Wilson, K., Bindel, D., Snavely, N.: When is rotations averaging hard? In: Proceedings of the European Conference on Computer Vision. pp. 255 – 270 (2016)
69. 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), 2816–2834 (2023)
70. Yang, L., Li, H., Rahim, J.A., Cui, Z., Tan, P.: End-to-end rotation averaging with multi-source propagation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11774–11783 (June 2021)
71. Yu, J.G., Xia, G.S., Samal, A., Tian, J.: Globally consistent correspondence of multiple feature sets using proximal Gauss–Seidel relaxation. Pattern Recognition **51**, 255 – 267 (2016)
72. Zhang, G., Larsson, V., Barath, D.: Revisiting rotation averaging: Uncertainties and robust losses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2023)
73. Zhou, X., Zhu, M., Daniilidis, K.: Multi-image matching via fast alternating minimization. In: Proceedings of the International Conference on Computer Vision. pp. 4032 – 4040 (2015)