Noname manuscript No.
(will be inserted by the editor)

Synchronization Problems in Computer Vision with Closed-form Solutions

Federica Arrigoni · Andrea Fusiello

the date of receipt and acceptance should be inserted later

Abstract In this paper we survey and put in a common framework several works that have been developed in different contexts, all dealing with the same abstract problem, called *synchronization* by some authors, or averaging, or graph optimization by others. The problem consists in recovering some variables from a set of pairwise relation measurements. In particular, we concentrate on instances where the variables and the measures belong to a *(semi-)group* and the measures are their mutual differences (or ratios, depending on how the group operation is called). The groups we deal with have a matrix representation, which leads to an elegant theory and closed-form solutions.

Keywords synchronization · averaging · graph optimization · multiple point-set registration · structure from motion · multi-view matching

1 Introduction

Consider a network of nodes where each node is characterized by an unknown state, and suppose that pairs of nodes can measure the ratio (or difference) between their states. The goal is to infer the unknown states from the pairwise measures. This is a general statement of the *synchronization* problem [63, 49, 99]. States are represented by elements of a group Σ, that is why the problem is actually referred to as *group synchronization*. The groups of interest are defined in Table 1, while Figure 2 represents their inclusion.

The problem can be usefully modelled by introducing a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which is referred to as the *measurement graph*, where nodes correspond to the unknown states and edges correspond to the pairwise measures, and it is well-posed only if such a graph is connected. Synchronization can be seen as upgrading from relative (pairwise) information, which involves two nodes at a time, onto absolute (global) information, which involves all the nodes simultaneously. In the literature, the same problem is also referred to as *averaging* [53, 117, 58] or *graph optimization* [34] in some cases.

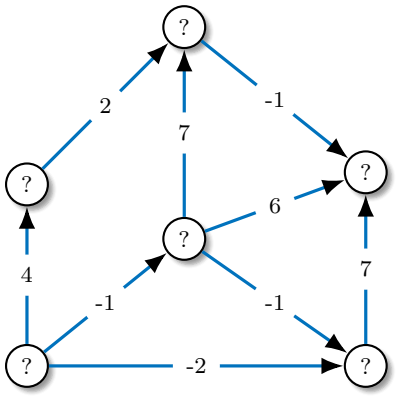

Fig. 1: Synchronization over $(\mathbb{Z}, +)$.

As an example, consider the graph in Figure 1, where nodes and edges are labelled with integer numbers: the task is to recover the unknown numbers in the nodes by measuring their differences (on the edges). Two things can be immediately observed: a solution exists only if

F. Arrigoni · A. Fusiello
DPIA, University of Udine, Via Delle Scienze 208, Udine, Italy
E-mail: arrigoni.federica@spes.uniud.it, andrea.fusiello@uniud.it

F. Arrigoni
CIIRC, Czech Technical University in Prague, Czech Republic

Table 1: Matrix groups/semigroups considered in this paper.

$GL(d)$	The General Linear Group is the set of invertible matrices: $GL(d) = \{M \in \mathbb{R}^{d\times d} \ s.t. \ \det(M) \neq 0\}$
$SL(d)$	The Special Linear Group is the set of matrices with unit determinat: $SL(d) = \{M \in \mathbb{R}^{d\times d} \ s.t. \ \det(M) = 1\}$
$O(d)$	The Orthogonal Group is the set of rotations and reflections: $O(d) = \{M \in \mathbb{R}^{d\times d} s.t. \ M^\mathsf{T}M = MM^\mathsf{T} = I_d\}$
$SO(d)$	The Special Orthogonal Group is the set of rotations: $SO(d) = \{M \in O(d) \ s.t. \ \det(M) = 1\}$
$GA(d)$	The General Affine Group is the set of affine maps: $GA(d) = \left\{ \begin{bmatrix} M & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}, \ s.t. \ M \in GL(d), \mathbf{t} \in \mathbb{R}^d \right\}$
$SE(d)$	The Special Euclidean Group is the set of direct isometries: $SE(d) = \left\{ \begin{bmatrix} M & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}, \ s.t. \ M \in SO(d), \ \mathbf{t} \in \mathbb{R}^d \right\}$
$\mathcal{S}_d$	The Symmetric Group is the set of total permutations: $\mathcal{S}_d = \{M \in \{0,1\}^{d\times d} \ s.t. \ M\mathbf{1} = \mathbf{1}, \ \mathbf{1}M = \mathbf{1}\}$
$\mathcal{I}_d$	The Symmetric Inverse Semigroup is the set of partial permutations: $\mathcal{I}_d = \{M \in \{0,1\}^{d\times d} \ s.t. \ M\mathbf{1} \leq \mathbf{1}, \ \mathbf{1}M \leq \mathbf{1}\}$

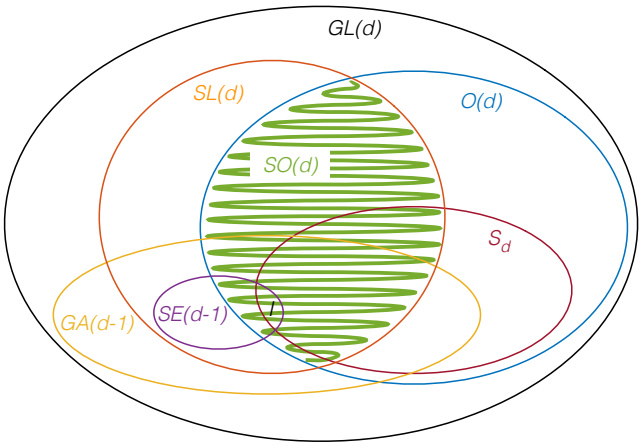

Fig. 2: Subgroups of $GL(d)$. The identity is in the intersection of all of them.

the sum of the differences along any cycle is zero, and, when it exists, the solution is not unique, for adding a constant to the nodes does not change the differences.

Measures are typically corrupted by errors, which can be gross errors (outliers) and/or a diffuse noise with small variance. If $\mathcal{G}$ is a tree then these errors will creep in the solution, however, as soon as redundant measures are considered (i.e. the graph has at least one cycle), they are exploited by synchronization to globally compensate the errors. The solution minimizes a suitable cost function which evaluates the coherence between the unknown states and the pairwise measures. By construction synchronization enforces *cycle consistency* [44,121], namely the property that the composition of relative measures along any cycle in the measurement graph should return the identity.

Several instances of synchronization have been studied in the literature. $\Sigma = \mathbb{R}$ yields *clock synchronization* (with offset) [63,49], from which the term *synchronization* originates, where all the nodes in a network are synchronized to a common clock. The same problem is studied under the name *levelling* in topography and surveying [22]. If the clock model includes also a drift (besides the offset), the problem can be addressed in $\Sigma = GA(1)$ [102]. Synchronization over $GA(3)$ has also been applied to RGB colour matching in image mosaicking [92]. $\Sigma = \mathbb{Z}_2$ gives *sign synchronization* [40], which was used to identify communities in a network where the interaction between the nodes is described by two dichotomous values, e.g., agreement/disagreement, and the network has a natural partition into two communities. $\Sigma = \mathbb{R}^d$ is a *translation synchronization* [13,91,108,80,2], namely the problem of localizing a set of nodes in space from pairwise differences. $\Sigma = SO(d)$ corresponds to *rotation synchronization* (also known as rotation averaging) [97,75,99,104,39, 47,24,58,36,116,7,111] and $\Sigma = SE(d)$ results in *rigid-motion synchronization* (also known as motion averaging or pose graph optimization) [48,53,107,109,18,11, 10,90,89], which find application in structure from motion, registration of 3D point sets and simultaneous localization and mapping (SLAM). $\Sigma = SL(d)$ produces *homography synchronization* [95], which is an essential step in the context of image stitching or mosaicking. Finally, $\Sigma = \mathcal{S}_d$ and $\Sigma = \mathcal{I}_d$ give rise to *permutation synchronization* [85,98,120] and *partial permutation synchronization* [8], respectively, which are related to multi-view matching, as it will be clarified in Section 8.2.

While it is clear that synchronization over $\mathbb{R}^d$ can be reduced to a linear system of equations, the other instances of synchronization involve the minimization of a non convex cost function, which make the problem difficult to solve.

However, if Σ admits a matrix representation, i.e. it can be embedded in $\mathbb{R}^{d\times d}$, then synchronization reduces to an eigenvalue decomposition, resulting in an efficient and closed-form solution. Specifically, the unknown states are derived from the top eigenvectors of a matrix constructed from the pairwise measures. An equivalent null-space formulation can also be derived.

This procedure was introduced in [99] for $\Sigma = SO(2)$, extended in [3,100] to $\Sigma = SO(3)$, and further generalized in [11,18] to $\Sigma = SE(3)$. The same formulation appeared in [95] and [85] for $\Sigma = SL(d)$ and $\Sigma = \mathcal{S}_d$ respectively. The latter was extended to $\Sigma = \mathcal{I}_d$ in [8].

Other techniques exploiting a matrix representation of the group can be found in the literature, which express the synchronization problem in terms of well studied mathematical tools, such as semidefinite programming [99,116,90] or matrix completion [9], resulting in iterative solutions. These approaches, however, are not applicable to all groups. For example, the semidefinite programming formulation derives from specific properties of $O(d)$ and $SE(d)$, namely the fact that the matrix constructed from the pairwise measures is positive semidefinite when considering the Orthogonal Group [99,116], and the property that the convex hull of the Special Euclidean Group admits a semidefinite representation [90]. The matrix completion formulation [9], instead, is based on the property that the matrix containing the pairwise measures is low-rank and it is incoherent (see [29]). Unfortunately in the $\Sigma = \mathcal{S}_d$ case the matrix is sparse, being composed of permutation matrices, and hence it does not satisfy the incoherence assumption.

Other approaches include iterating local solutions on the measurement graph or explicit minimization of a cost function. The former comprises [97], where the error is distributed over a set of cycles, and [107,1], where each state is updated in turn in a distributed fashion. The latter includes Quasi-Newton iterations [48], the Levenberg-Marquardt algorithm [39], Riemannian trust-region optimization [24], Riemannian gradient descent [109], integer quadratic programming [33], and the Gauss-Seidel method [120]. These techniques, however, heavily depend on the chosen group and its parametrization. For instance, unit quaternions are used in [35] and dual quaternions in [107], which represent rotations and rigid-motions in 3-space, respectively. See also the references in [33].

1.1 Scope and outline

In this paper we provide a comprehensive survey on closed-form solutions to group synchronization, namely linear least squares [13,91] and spectral decomposition [99,95,100,3,85,18,11,8]. The former solves synchronization over $\mathbb{R}^d$ in an optimal way. The latter, although suboptimal, is theoretically appealing since it can be applied to any group admitting a matrix representation (e.g. homographies, rigid motions, permutations, ...), as opposed to other techniques which are based on ad-hoc minimizations of specific cost functions. These solutions are extremely fast and they easily cope with weights on individual relative measures, allowing a robust extension via Iteratively Reweighted Least Squares (IRLS) [60].

We also set forth a theoretical unified framework where several synchronization problems are seen as instances of a more abstract principle, gathering several works that were developed in different communities (including Computer Vision, Photogrammetry, Robotics and Graph Theory), and we show how this framework can be profitably used in several applications.

The paper is organized as follows. Section 2 formally defines the synchronization problem, which is grounded on the notion of group-labelled graph. Section 3 is devoted to synchronization over $(\mathbb{R}^d, +)$, which is expressed as a linear system of equations. Section 4 addresses synchronization over $(GL(d), \cdot)$, which can be cast to a spectral decomposition or a null-space problem. Then, several subgroups of $GL(d)$ are analysed in Section 5, namely $SL(d)$, $O(d)$, $SE(d-1)$, $GA(d-1)$ and $\mathcal{S}_d$, in which cases the solution needs to be projected onto the group, as closure is not guaranteed. Section 6 shows that the spectral solution can be extended to synchronization over $\mathcal{I}_d$, which is an inverse monoid and a subsemigroup of $\mathcal{S}_d$. Some considerations about the optimization problem associated with the spectral solution are reported in Section 7. Finally, Section 8 describes how some Computer Vision problems can be addressed in terms of synchronization and Section 9 briefly discusses pros and cons of spectral synchronization. The results presented in this paper require some basic notions from graph theory, which are covered in Appendix B, and the definitions of the Kronecker, Hadamard and Khatri-Rao products, which are given in Appendix A.

2 Theoretical framework

Let us start by introducing the notion of *group-labelled graph* [43]. Let $(\Sigma, *)$ be a group with unit element 1_Σ, and let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple directed graph with vertex set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edge set $\mathcal{E}$, with $m = |\mathcal{E}|$. A *Σ-labelled* graph is a directed graph with a labelling of its edge set by elements of Σ, that is a tuple $\Gamma = (\mathcal{V}, \mathcal{E}, z)$ where

$$z : \mathcal{E} \to \Sigma \tag{1}$$

is such that if $(i, j) \in \mathcal{E}$ then $(j, i) \in \mathcal{E}$ and

$$z(j, i) = z(i, j)^{-1}. \tag{2}$$

Thus, we may also view $\mathcal{G}$ as an undirected graph.

Definition 1 Let $\Gamma = (\mathcal{V}, \mathcal{E}, z)$ be a Σ-labelled graph. We say that a circuit $\{(i_1, i_2), (i_2, i_3), \dots, (i_\ell, i_1)\}$ is a *null cycle*[1] if and only if the composition of the edge labels along the circuit returns the identity, namely

$$z(i_1, i_2) * z(i_2, i_3) * \dots * z(i_\ell, i_1) = 1_\Sigma. \tag{3}$$

The "null" term clearly refers to an additive notation for the group (which is used in [65,38,57]), without implying that Σ needs to be Abelian. Note that if the group is not commutative, then it may happen that cyclic shifts of the same circuit yield different elements of the group. Nevertheless the notion of null cycle is well defined, as either all of the cyclic shifts are equal to 1_Σ or none of them, as observed also in [57]. As noted in [14], the concept of null cycle resembles Kirchoff's voltage law, stating that the electrical potential differences around any cycle sum to zero.

Definition 2 Let $\Gamma = (\mathcal{V}, \mathcal{E}, z)$ be a Σ-labelled graph. Let $x : \mathcal{V} \to \Sigma$ be a vertex labelling. We say that x is a *consistent labelling* if and only if

$$z(e) = x(i) * x(j)^{-1} \quad \forall\ e = (i, j) \in \mathcal{E}. \tag{4}$$

A Σ-labelled graph admitting a consistent labelling is also called *balanced* [62]. Equation (4) means that each edge label is the ratio of the corresponding vertex labels, as shown in Figure 3. Such condition is referred to as *consistency constraint* and it is equivalent to

$$z(e) * x(j) = x(i) \quad \forall\ e = (i, j) \in \mathcal{E}. \tag{5}$$

It is understood that a consistent labelling is defined up to a global (right) product with any group element, in the sense that if $x : \mathcal{V} \to \Sigma$ is consistent then also $y : \mathcal{V} \to \Sigma$, $y(i) = x(i) * s$, is consistent, for any (fixed) $s \in \Sigma$.

The notion of consistent labelling is strictly related to that of null cycle, as stated by the following result.

Proposition 1 ([57]) *Let $\Gamma = (\mathcal{V}, \mathcal{E}, z)$ be a Σ-labelled graph. There exists a polynomial algorithm which either finds a non-null cycle in Γ or finds a consistent labelling of Γ.*

The following procedure draws an outline of the proof. First, compute a spanning tree (see Appendix B) and use Equation (5) to label nodes, starting from the root labelled with the identity 1_Σ: this is a consistent labelling by construction. Then add one by one the edges not belonging to the spanning tree, thereby creating a circuit. If the cycle is null then the edge can be added and leave the labelling consistent, otherwise a non-null cycle has been found.

[1] A circuit is also a cycle; definitions are given in Appendix B.

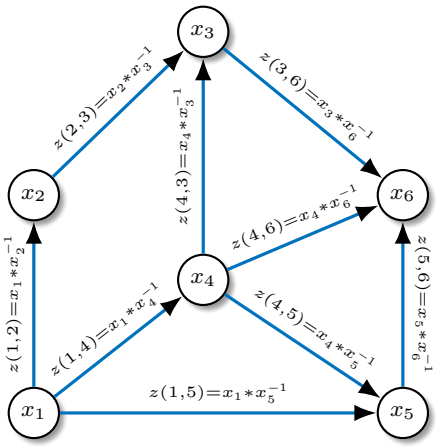

Fig. 3: Consistent labelling.

Corollary 1 ([57]) *The Σ-labelled graph $\Gamma = (\mathcal{V}, \mathcal{E}, z)$ has a consistent labelling if and only if it does not contain a non-null cycle.*

One direction is an immediate consequence of Proposition 1. The opposite direction follows from the observation that, if Equation (4) holds, then all the terms in the left side in Equation (3) simplify, yielding the identity.

2.1 Group Feedback Edge Set

The problem of finding non-null cycles in a group-labelled graph is studied in Graph Theory community under the name of "group feedback edge set" problem [57]. Specifically, the goal is to break non-null cycles by deleting k edges, where $k \in \mathbb{N}$ is assumed to be known.

Definition 3 Let $\Gamma = (\mathcal{V}, \mathcal{E}, z)$ be a Σ-labelled graph. The *Group Feedback Edge Set* (GFES) problem is defined as follows: on input (Γ, k) for some $k \in \mathbb{N}$, decide whether there exists a subset of the edges $\mathcal{S} \subseteq \mathcal{E}$ with $|\mathcal{S}| \leq k$ such that the labelled graph of the remaining edges $\Gamma' = (\mathcal{V}, \mathcal{E} \setminus \mathcal{S}, z)$ does not contain a non-null cycle.

With some abuse of notation, in Definition 3 we denote with $(\mathcal{V}, \mathcal{E} \setminus \mathcal{S}, z)$ the Σ-labelled graph with edges in $\mathcal{S}$ removed from $\mathcal{E}$, even though formally z has in its domain edges that do not exist in $\mathcal{E} \setminus \mathcal{S}$.

The set $\mathcal{S}$ satisfying Definition 3 (if it exists) is called the *feedback edge set* of Γ. The interpretation is that $\mathcal{S}$ identifies edges with *outlying labels* that prohibit a consistent labelling to be found. Note that in the presence of noise we have to relax Equation (3) and consider the following

$$\delta\big(z(i_1, i_2) * z(i_2, i_3) * \dots * z(i_\ell, i_1), 1_\Sigma\big)/\sqrt{\ell} \leq \tau \tag{6}$$

where $\tau \geq 0$ is a given threshold and it is assumed that Σ admits a metric function $\delta : \Sigma \times \Sigma \to \mathbb{R}^+$. The normalization factor $\sqrt{\ell}$ takes into account error propagation when considering long cycles [44].

Outlying labels can be detected through the methods in [115,42], which come from Graph theory community. Alternatively, Computer Vision solutions can be used, which include outlier rejection heuristics [44,12], Random Sample Consensus (RANSAC) [54,82], and Bayesian inference [121,81,26]. However, these strategies are computationally demanding and do not scale well with the size of the graph.

2.2 Group synchronization

Let us assume that Σ is equipped with a metric function $\delta : \Sigma \times \Sigma \to \mathbb{R}^+$ and let $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ be a non-negative non-decreasing function with a unique minimum at 0 and $\rho(0) = 0$. Some instances are the quadratic loss function $\rho(y) = y^2$ or robust loss functions used in M-estimators [60].

Definition 4 Let $\Gamma = (\mathcal{V}, \mathcal{E}, z)$ be a Σ-labelled graph. Let $\widetilde{x} : \mathcal{V} \to \Sigma$ be a vertex labelling. We define the *consistency error* of $\widetilde{x}$ as the quantity

$$\epsilon(\widetilde{x}) = \sum_{(i,j)\in\mathcal{E}} \rho\Big(\delta\big(\widetilde{z}(i,j), z(i,j)\big)\Big) \tag{7}$$

where $\widetilde{z}$ is the edge labelling induced by $\widetilde{x}$, namely $\widetilde{z}(i,j) = \widetilde{x}(i) * \widetilde{x}(j)^{-1}$.

A vertex labelling is consistent if and only if it has zero consistency error. In practical applications a labelling with zero error hardly exists, since the edge labels are corrupted by noise, thus the goal is to address the following problem.

Definition 5 Given a Σ-labelled graph $\Gamma = (\mathcal{V}, \mathcal{E}, z)$, the *group synchronization problem* consists in finding a vertex labelling with minimum consistency error.

In other words, one wants to recover the unknown group elements (vertex labels) given a redundant set of noisy measurements of their ratios (edge labels), as shown in Figure 4.

A related approach (e.g. [97]) consists in minimizing the cost function (7) with respect to the edge labelling $\tilde{z}$ while imposing the constraint that all the cycles are null (a.k.a. *cycle consistency*). For the $\Sigma = \mathbb{R}^d$ case this is equivalent to synchronization (Proposition 2), whereas analogous results are not known in general.

The synchronization problem requires the graph to be connected, but *error compensation* happens only

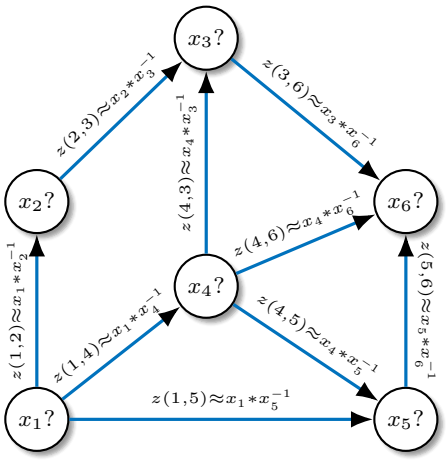

Fig. 4: The synchronization problem.

within cycles. The minimum number of relative measures is $n-1$, which makes $\mathcal{G}$ a tree. In this case every vertex can be labeled by simply propagating Equation (5) along the tree, starting from the root labeled with the identity element. In this case, however, there is no remedy to error propagation: the error affecting an edge label propagates down to the leaves of the tree without compensation, as shown (e.g.) by experiments in [53]. In the synchronization problem, instead, the goal is to exploit *redundant* relative measures in a global fashion to improve the final estimate.

If the measures are also corrupted by outliers, one needs to solve a GFES problem beforehand, using a relaxed notion of null cycle, i.e., Equation (6). Alternatively, a robust loss function can be used in (7) without detecting outliers explicitly (as done in Section 9.5).

2.3 Synchronization over an inverse monoid

As observed in [8], the notion of synchronization can be extended to the case where Σ is an *inverse monoid*. One example is $\Sigma = \mathcal{I}_d$, which is a subsemigroup of $\mathcal{S}_d$, resulting in *partial permutation synchronization* [8].

Definition 6 An *inverse semigroup* $(\Sigma, *)$ is a semigroup in which for all $s \in \Sigma$ there exists an element $t \in \Sigma$ such that $s = s * t * s$ and $t = t * s * t$. In this case, we write $t = s^{-1}$ and call t the *inverse* of s. If Σ has an identity element 1_Σ (i.e. it is a *monoid*), then it is called an *inverse monoid*.

Remark 1 Inverses in an inverse semigroup have many of the same properties as inverses in a group, for instance, $(a * b)^{-1} = b^{-1} * a^{-1}$ for all $a, b \in \Sigma$.

If Σ is an inverse monoid, then Equations (4) and (7) still make sense, with the provision that $x(j)^{-1}$

now denotes the inverse of $x(j)$ in the semigroup. Note that $x(j)^{-1} * x(j)$ and $x(j) * x(j)^{-1}$ are not necessarily equal to the identity, thus Equation (4) is not equivalent to (5). The solution to the synchronization problem over an inverse monoid is defined up to a global (right) product with any element $y \in \Sigma$ such that $y * y^{-1} = 1_\Sigma = y^{-1} * y$.

As concerns the notion of null-cycle, Equation (3) still makes sense in the case of an inverse monoid, since it only involves products between elements of the set and the existence of a unit element 1_Σ. However, Proposition 1 (and hence one direction of Corollary 1) does not straightforwardly extend to this case since the proof exploits Equation (5), which is no longer equivalent to the consistency constraint. Regarding the other direction of Corollary 1, note that, if the consistency constraint holds, then the left side in Equation (3) rewrites

$$\begin{aligned} &z(i_1,i_2) * z(i_2,i_3) * \ldots * z(i_\ell,i_1) = \qquad (8)\\ &= x(i_1) * \underbrace{x(i_2)^{-1} * x(i_2)}_{\neq 1_\Sigma} * x(i_3)^{-1} * \ldots * x(i_\ell) * x(i_1)^{-1} \end{aligned}$$

which, in general, does not coincide with the identity. Developing a complete theory for synchronization over an inverse monoid is outside the scope of this paper.

3 Synchronization over $\mathbb{R}^d$

In this section we consider the synchronization of real vectors with addition, namely $(\Sigma, *) = (\mathbb{R}^d, +)$, which is also known as *translation synchronization*, and we derive a direct solution following [13,91].

A vertex labelling $\mathbf{x} : \mathcal{V} \to \mathbb{R}^d$ is consistent with a given edge labelling $\mathbf{z} : \mathcal{E} \to \mathbb{R}^d$ if and only if[2]

$$\mathbf{x}_i - \mathbf{x}_j = \mathbf{z}_{ij} \;\forall\; (i,j) \in \mathcal{E}. \qquad (9)$$

Note that we can view each component in Equation (9) as a synchronization over $(\mathbb{R}, +)$.

If we denote the *incidence vector* of the edge (i,j) with

$$\boldsymbol{b}_{ij} = [\, 0, \ldots, \underset{\uparrow}{1}_{i}, \ldots, \underset{\uparrow}{-1}_{j}, \ldots, 0 \,]^\mathsf{T} \qquad (10)$$

then Equation (9) can be written as

$$[\, \mathbf{x}_1, \ldots, \mathbf{x}_n \,]\, \boldsymbol{b}_{ij} = \mathbf{z}_{ij} \quad \forall\; (i,j) \in \mathcal{E} \qquad (11)$$

or, equivalently, in matrix form

$$XB = Z \qquad (12)$$

[2] For simplicity of notation, hereafter we will use subscripts instead of parenthesis to denote indices of a node/edge labelling.

where B is the $n \times m$ incidence matrix of the *directed* graph $\mathcal{G}$, which has the vectors $\boldsymbol{b}_{ij}$ as columns, X is the $d \times n$ matrix obtained by juxtaposing all the vertex labels, namely $X = [\mathbf{x}_1 \ldots \mathbf{x}_n]$, and Z is the $d \times m$ matrix obtained by juxtaposing all the edge labels (ordered as in B), namely $Z = [\mathbf{z}_{12} \;\ldots\; \mathbf{z}_{ij} \;\ldots]$. See Appendix B.1 for the definition of incidence matrix and related properties.

Applying the vectorization operator $\mathrm{vec}(\cdot)$ to both sides in (12) and using formula (87) we get

$$(B^\mathsf{T} \otimes I_d)\, \mathrm{vec}(X) = \mathrm{vec}(Z) \qquad (13)$$

where I_d denotes the $d \times d$ identity matrix and $\otimes$ denotes the Kronecker product, defined in Appendix A, which has the effect of "inflating" the incidence matrix in order to cope with the vector representation of the group elements.

Under the assumption that the graph is connected we have $\mathrm{rank}(B) = n-1$ and hence, using (89), we have $\mathrm{rank}(B^\mathsf{T} \otimes I_d) = dn - d$. The rank deficiency corresponds to the translation ambiguity. Since the solution to the synchronization problem is defined up to a global group element, we are allowed w.l.o.g. to arbitrarily set $\mathbf{x}_k = \mathbf{0} = 1_\Sigma$ for a chosen $k \in \mathcal{V}$. Removing $\mathbf{x}_k$ from the unknowns and the corresponding row in B leaves a full-rank matrix.

With a suitable choice of $\delta(\cdot,\cdot)$ and ρ in Equation (7), the consistency error of the synchronization problem writes

$$\epsilon(X) = \left\| (B^\mathsf{T} \otimes I_d)\, \mathrm{vec}(X) - \mathrm{vec}(Z) \right\|^2 \qquad (14)$$

where $\|\cdot\|$ denotes the Euclidean norm. Thus the least-squares solution of Equation (13) solves the synchronization problem. This method is adopted in [13,91] to localize a group of agents in a sensor network and in [6] to recover camera positions in a structure-from-motion pipeline.

Remark 2 If $\mathbf{c} \in \{-1,0,1\}^m$ denotes the indicator vector of a circuit in $\mathcal{G}$, the cycle is null if and only if

$$Z\mathbf{c} = \mathbf{0}. \qquad (15)$$

If the equations coming from all the circuits in a *cycle basis* are stacked, then the cycle consistency writes:

$$ZC^\mathsf{T} = 0 \qquad (16)$$

where $C \in \{-1,0,1\}^{(m-n+1)\times m}$ denotes the cycle matrix associated to the basis. See Appendix B or [31] for the definitions of cycle basis and cycle matrix. Using the vectorization operator and formula (87), cycle consistency can also be expressed as

$$(C \otimes I_d)\, \mathrm{vec}\,(Z) = \mathbf{0}. \qquad (17)$$

It can be shown that the edge labels produced by the synchronization process are the closest to the input edge labels among those that yield null-cycles.

Proposition 2 ([91]) *If $\widetilde{X}$ is the least-squares solution to Equation* (13)*, then the induced edge labelling $\widetilde{Z} = \widetilde{X}B$ solves the following constrained minimization problem*

$$\min_{\widetilde{Z}} \|Z - \widetilde{Z}\|^2 \quad s.t.\ \widetilde{Z}C^{\mathsf{T}} = 0 \tag{18}$$

where C denotes the cycle matrix associated to a cycle basis of $\mathcal{G}$.

This result was also known in topography, in the context of the least-squares adjustment of leveling networks [22].

4 Synchronization over $GL(d)$

In this section we consider the synchronization problem over the General Linear Group $GL(d)$, which is the set of all $d \times d$ invertible matrices, where the group operation $*$ reduces to matrix multiplication and $1_\Sigma = I_d$. A vertex labelling $X : \mathcal{V} \to \mathbb{R}^{d\times d}$ is consistent with a given edge labelling $Z : \mathcal{E} \to \mathbb{R}^{d\times d}$ if and only if $Z_{ij} = X_i \cdot X_j^{-1}$.

All the vertex/edge labels can be collected in two matrices $X \in \mathbb{R}^{dn\times d}$ and $Z \in \mathbb{R}^{dn\times dn}$ respectively, which are "matrices of matrices" defined as follows

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix}, \quad Z = \begin{bmatrix} I_d & Z_{12} & \dots & Z_{1n} \\ Z_{n1} & I_d & \dots & Z_{2n} \\ \dots & & & \dots \\ Z_{n1} & Z_{n2} & \dots & I_d \end{bmatrix}. \tag{19}$$

For a complete graph, the consistency constraint can be expressed in matrix form as

$$Z = XX^{-\flat} \tag{20}$$

where $XX^{-\flat}$ contains the edge labels induced by X and $X^{-\flat} \in \mathbb{R}^{d\times dn}$ denotes the block-matrix containing the inverse of each $d \times d$ block of X, i.e. $X^{-\flat} = [X_1^{-1}\ X_2^{-1}\ \dots\ X_n^{-1}]$. Note that Equation (20) implies that $\operatorname{rank}(Z) = d$.

Remark 3 By computation it can be verified that

$$X^{-\flat}X = nI_d \tag{21}$$

and hence Z/n is idempotent.

If the graph is not complete then Z is not fully specified. In this case missing edges are represented as zero entries, i.e. $Z_A := Z \circ (A \otimes \mathbf{1}_{d\times d})$ represents the matrix of the available measures, where $\circ$ is the Hadamard product and A is the adjacency matrix of the graph $\mathcal{G}$, which gets "inflated" by the Kronecker product with $\mathbf{1}_{d\times d}$ to match the block structure of the measures. Being a matrix of 0/1, the effect of its entry-wise product with Z is to zero the unspecified entries and leave the others unchanged. See Appendices B and A for the definitions of Hadamard product and adjacency matrix, respectively. Hence the consistency constraint writes

$$Z_A = (XX^{-\flat}) \circ (A \otimes \mathbf{1}_{d\times d}). \tag{22}$$

With a suitable choice of $\delta(\cdot,\cdot)$ and ρ in Equation (7) the consistency error of the synchronization problem writes

$$\epsilon(X) = \|Z_A - (XX^{-\flat}) \circ (A \otimes \mathbf{1}_{d\times d})\|_F^2 \tag{23}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The Hadamard product with $A \otimes \mathbf{1}_{d\times d}$ mirrors the summation over the edges in $\mathcal{E}$ in the definition of the consistency error. The minimization of ϵ is a non-linear least squares problem, for which closed-form solutions do not seem to exist. However, two direct solutions to a related version of the problem exist [99,100,3], which can be derived by considering the exact (noiseless) case.

4.1 Spectral solution

Let us consider the noiseless case, i.e. $\epsilon = 0$, and let us start assuming that the graph is complete. Using the consistency constraint and (21) we obtain

$$ZX = nX \tag{24}$$

which means that – in the absence of noise – the columns of X are d (independent) eigenvectors of Z corresponding to the eigenvalue n. Note that, since Z has rank d, all the other eigenvalues are zero, thus n is also the *largest* eigenvalue of Z.

We now consider the case of missing edges in which the graph is not complete and the adjacency matrix comes into play.

Proposition 3 ([99,100,3]) *The columns of X are d (independent) eigenvectors of $(D \otimes I_d)^{-1}Z_A$ associated to the eigenvalue* 1*.*

Proof In the case of missing data, it can be seen that Equation (24) generalizes to

$$Z_A X = (D \otimes I_d)X \tag{25}$$

where D is the degree matrix of the graph (see Appendix B.1). Indeed, the i-th block-row in the above equation is

$$\sum_{j \ s.t. \ (i,j)\in\mathcal{E}} Z_{ij}X_j = [D]_{ii}X_i \tag{26}$$

which is satisfied since $Z_{ij} = X_i X_j^{-1}$. □

Note that the incomplete data matrix Z_A has full rank in general, thus 1 is not the unique nonzero eigenvalue of $(D \otimes I_d)^{-1}Z_A$, in contrast to the case of Equation (24). However, it can be shown that 1 is the *largest* eigenvalue of such a matrix.

Proposition 4 ([11]) *The matrix $(D \otimes I_d)^{-1}Z_A$ has real eigenvalues. The largest eigenvalue is 1 with multiplicity d.*

Proof By computation it can be verified that

$$Z_A = \operatorname{blkdiag}(X)(A \otimes I_d)\operatorname{blkdiag}(X)^{-1} \tag{27}$$

where $\operatorname{blkdiag}(X)$ produces a $dn \times dn$ block-diagonal matrix with $d \times d$ blocks $X_1, \ldots, X_n$ along the diagonal. Note that the diagonal matrix $(D \otimes I_d)^{-1}$ commutes with $\operatorname{blkdiag}(X)$, since each $d \times d$ block along its diagonal is a multiple of the identity matrix. Thus

$$\begin{aligned}&(D \otimes I_d)^{-1}Z_A = \\ &= \operatorname{blkdiag}(X)(D \otimes I_d)^{-1}(A \otimes I_d)\operatorname{blkdiag}(X)^{-1} = \\ &= \operatorname{blkdiag}(X)((D^{-1}A) \otimes I_d)\operatorname{blkdiag}(X)^{-1}\end{aligned} \tag{28}$$

where the last equality follows from properties (85) and (86). Hence $(D \otimes I_d)^{-1}Z_A$ is similar to the matrix $(D^{-1}A) \otimes I_d$, i.e., they have the same eigenvalues. The matrix $D^{-1}A$ is the transition matrix of the graph $\mathcal{G}$ (see Appendix B.1), which – as a consequence of the Perron-Frobenius theorem – has real eigenvalues and 1 is the largest eigenvalue (with multiplicity 1), if the graph is connected. Since the eigenvalues of the Kronecker product of two matrices are the product of the eigenvalues of the matrices, we conclude that the largest eigenvalue of $(D^{-1}A) \otimes I_d$ is 1 with multiplicity d. □

The proof of Proposition 4 has pointed out that – provided that Z is decomposable as $Z = XX^{-\flat}$ – the matrix $(D \otimes I_d)^{-1}Z_A$ has a particular structure that yields real eigenvalues, although it is not symmetric. In particular, the eigenvalues do not depend on the measured data, but they depend only on the structure of the graph $\mathcal{G}$ (through the matrices D and A).

When noise is present, i.e. $\epsilon \neq 0$, the eigenvectors of $(D \otimes I_d)^{-1}Z_A$ corresponding to the d largest eigenvalues are an estimate of the vertex labelling X. The presence of noise, however, cripples the structure of Z_A, i.e. $Z_A \neq (XX^{-\flat}) \circ (A \otimes \mathbf{1}_{d\times d})$, thus the eigenvalues and the eigenvectors may be complex. As a consequence, after computing the leading eigenvectors, the imaginary part is zeroed. This approach is adopted in [6] in the $d = 1$ case to synchronize distances between camera pairs in a structure-from-motion pipeline. To the best of our knowledge, no general results are known linking this spectral solution to the synchronization cost function. Results in the special case where $\Sigma = O(d)$ are reported in Section 7.1.

Ambiguity. Since the eigenvalue 1 is repeated, the corresponding eigenvectors span a linear subspace, and hence any basis for such a space is a solution. However, a change of the basis in the eigenspace corresponds to right-multiply the eigenvectors by an invertible $d \times d$ matrix, i.e., the solution to synchronization is defined up the action of an element of $GL(d)$, as expected.

Projection. Let U be the matrix containing the d leading eigenvectors of $(D \otimes I_d)^{-1}Z_A$, where the imaginary part (if any) is zeroed. The blocks of U are in general non-singular, hence they belong to $GL(d)$. This means that the solution is *intrinsic*, and no projection is required.

Remark 4 The top eigenvector can be computed by the power iteration method, which, considering (e.g.) the case of a complete graph, starts with a random vector $\mathbf{x}_0 \in \mathbb{R}^{dn}$ and iterates the relation $\mathbf{x}_{k+1} = Z\mathbf{x}_k / \|Z\mathbf{x}_k\|$, thus it requires to compute Z^k, for $k = 1, 2, \ldots, k_{\max}$. It is observed in [99] that multiplying the matrix Z by itself integrates the consistency relation of triplets, while high order iterations exploit consistency relations of longer cycles. Indeed

$$Z^2_{ij} = \sum_{k=1}^{n} Z_{ik}Z_{kj} \tag{29}$$

$$Z^3_{ij} = \sum_{k=1}^{n}\sum_{h=1}^{n} Z_{ik}Z_{kh}Z_{hj} \ \ldots \tag{30}$$

Thus the top eigenvector integrates the consistency relation of all cycles.

4.2 Null-space solution

We now show that synchronization over $(GL(d), \cdot)$ can also be expressed as a null-space problem. If $\epsilon = 0$ Equation (24) is equivalent to

$$(nI_{dn} - Z)X = 0 \tag{31}$$

which means that, if the graph is complete, the vertex labelling X coincides with the d-dimensional null-space

of $nI_{dn} - Z$. In the case of missing edges, let us rewrite (25) as

$$(D \otimes I_d - Z_A)X = 0 \tag{32}$$

thus X belongs to the null-space of $D \otimes I_d - Z_A$.

Let us observe that the matrix $D \otimes I_d$ coincides with $(D \otimes \mathbf{1}_{d\times d}) \circ Z$, since Z has identity blocks along its diagonal and $D \otimes \mathbf{1}_{d\times d}$ is block-diagonal. Using the distributive property of the involved products, we obtain an equivalent expression for $D \otimes I_d - Z_A$

$$D \otimes I_d - Z_A = ((D-A) \otimes \mathbf{1}_{d\times d}) \circ Z = (L \otimes \mathbf{1}_{d\times d}) \circ Z \tag{33}$$

where $L = D - A$ is the Laplacian matrix of $\mathcal{G}$ (see Appendix B.1), which gets inflated to a $d \times d$-block structure by the Kronecker product with $\mathbf{1}_{d\times d}$, to match the block structure of Z.

Note that in practice one cannot measure the matrix $(L \otimes \mathbf{1}_{d\times d}) \circ Z$, since the full Z is not available. In fact, only the product $Z \circ (A \otimes \mathbf{1}_{d\times d})$ is available. Therefore, the left side in (33) will be used in real scenarios. However, the right side emphasizes the presence of the Laplacian matrix, which is useful to prove that the null-space of $D \otimes I_d - Z_A$ is d-dimensional, as happens in the case of a complete graph.

Proposition 5 ([11]) *The matrix $D \otimes I_d - Z_A$ has a d-dimensional null-space.*

Proof By computation it can be verified that

$$\begin{aligned} D \otimes I_d - Z_A &= (L \otimes \mathbf{1}_{d\times d}) \circ Z = \\ &= \operatorname{blkdiag}(X)(L \otimes I_d) \operatorname{blkdiag}(X)^{-1} \end{aligned} \tag{34}$$

which means that $D \otimes I_d - Z_A$ and $L \otimes I_d$ are similar, thus they have the same rank. The rank of the Laplacian matrix is $n - 1$, under the assumption that the graph is connected (see Appendix B.1). Since the rank of the Kronecker product of two matrices is the product of the rank of the matrices, we obtain

$$\operatorname{rank}(D \otimes I_d - Z_A) = \operatorname{rank}(L) \operatorname{rank}(I_d) = dn - d \tag{35}$$

thus we have the thesis. □

When noise is present, an estimate of X is given by the right singular vectors of $D \otimes I_d - Z_A$ corresponding to the d least singular values, which solve the following problem

$$\min_{X^\intercal X = nI_d} \|(D \otimes I_d - Z_A)X\|_F^2. \tag{36}$$

A formal relationship between this cost function and the consistency error of the synchronization problem has still to be found.

The considerations about ambiguity/projection made for the spectral method apply also to the null-space solution, modulo the fact that singular vectors are real even in the absence of noise, so no rounding is needed.

4.3 Additive solution

We observe that synchronization over the General Linear Group can be cast to a translation synchronization, exploiting the fact that $GL(d)$ has the structure of a Lie group [114], where the associated Lie algebra consists of all $d \times d$ real matrices with the commutator operator serving as the Lie bracket, namely $[Y, W] = YW - WY$. Informally, a Lie group can be locally viewed as topologically equivalent to a vector space, and the local neighbourhood of any group element can be adequately described by its tangent space, whose elements form a Lie algebra. The Lie algebra and the Lie group are related by the exponential mapping, and the inverse mapping from the Lie group to the Lie algebra is given by matrix logarithm.

By taking the logarithm, the consistency constraint of the synchronization problem over $GL(d)$, that is $Z_{ij} = X_i X_j^{-1}$, can be transformed into the consistency constraint of an additive group, namely

$$\log(Z_{ij}) = \log(X_i) - \log(X_j) \tag{37}$$

assuming that the each of the above matrices admits a unique real logarithm. Specifically, by vectorizing each side in (37), a relation of the form (9) is obtained, which defines a translation synchronization problem. Thus the solution can be found by solving a linear system in the least-squares sense, as done in Section 3, or via IRLS (to gain robustness to outliers), as explained in Section 9.5. In other words, the synchronization problem is addressed in the Lie algebra rather than in the group.

However, as observed in [53], the Euclidean distance in the Lie algebra does not coincide with the Riemannian distance in the group, but it constitutes a first-order approximation, as stated by the Baker-Campbell-Hausdorff formula [114]. For this reason, in [53,54,36] the solution is found by iterating between solving the linear system in the Lie algebra and remapping onto the group.

As a final note, it is straightforward to see that the approach of Section 4 also applies to synchronization over $(\mathbb{R} \setminus \{0\}, \cdot)$, for it coincides with $(GL(1), \cdot)$.

5 Synchronization over subgroups of $GL(d)$

The analysis carried out in Section 4 can be extended to the case where Σ is a subgroup of $GL(d)$, i.e., it can be embedded in $\mathbb{R}^{d\times d}$, where the group operation $*$ reduces to matrix multiplication and $1_\Sigma = I_d$. In this case Propositions 4 and 5 still hold, and the synchronization problem can be addressed either via the spectral solution, which computes the top d eigenvectors of

$(D \otimes I_d)^{-1} Z_A$ (which may be complex in the presence of noise), or via the null-space solution, which computes the least d right singular vectors of $D \otimes I_d - Z_A$. Alternatively, the (approximate) null-space of $I_{dn} - (D \otimes I_d)^{-1} Z_A$ can be computed, as done in [95].

Let U be the $dn \times d$ matrix containing either the output of the spectral method or the null-space solution. Recall that U is not uniquely determined (even in the absence of noise) but it is defined up to (right) multiplication by an element of $GL(d)$, since any basis for the null-space of $D \otimes I_d - Z_A$ (or, equivalently, any basis for the eigenspace of $(D \otimes I_d)^{-1} Z_A$ associated with eigenvalue 1) is a solution. Thus a procedure that reduces the ambiguity up to an element of Σ is required, since the solution to the synchronization problem is inherently defined up to an element of the group.

Note that closure is not always guaranteed. In other words, the spectral and the null space methods produce an *extrinsic* estimate of the vertex labelling X which needs to be eventually projected onto Σ. This approach is clearly suboptimal with respect to working *in* the group, and represents the price to be paid for simplicity and computational efficiency.

In the following sections we will analyze synchronization is some subgroups of $GL(d)$ (see Table 1 and Figure 2). They basically differ from each other by the ambiguity fixing and the projection stages.

Remark 5 Synchronization over a subgroup of $GL(d)$ can also be addressed via the approach detailed in Section 4.3, as done in [53,54,36]. Particularly interesting are the $\Sigma = SO(3)$ and $\Sigma = SE(3)$ cases, where the associated Lie algebras are described by 3 and 6 parameters, respectively, and the exponential and logarithm maps admit closed form expressions [79,30].

5.1 Synchronization over $SL(d)$

The Special Linear Group $SL(d)$ is the set of $d \times d$ matrices with unit determinant. Synchronization over $SL(3)$ is studied in [95] within the context of multiple-view homography estimation, that is why the problem is referred to as *homography synchronization*.

Ambiguity. U is the solution up to multiplication by element of $GL(d)$, which can be reduced to $SL(d)$ after permutation of two columns of U s.t. $\det(U_1) > 0$ and division by $\sqrt[d]{\det(U_1)}$, where U_1 denotes the first $d \times d$ block in U.

Projection. In order to obtain elements of $SL(d)$ from U, each $d \times d$ block in U, denoted by U_i, must be scaled to unit determinant, which can be done by dividing U_i by $\sqrt[d]{\det(U_i)}$. In the case of the spectral solution, before performing such projection, the imaginary part of the eigenvectors is zeroed.

5.2 Synchronization over $O(d)$

The Orthogonal Group $O(d)$ is the set of orthogonal transformations in d-space, which admits a matrix representation through $d \times d$ orthogonal matrices. An important subgroup of $O(d)$ is the Special Orthogonal Group $SO(d)$, that is the set of orthogonal matrices with determinant 1, which represent rotations in d-space. Synchronization over $SO(d)$ is also known as *rotation (angular) synchronization* [99] or *multiple rotation averaging* [58,117]. A comprehensive survey on existing solutions can be found in [34,111].

From the theoretical perspective, synchronization over $SO(3)$ is analyzed in depth in [58,117]. In [58] the consistency error (7) is studied under the choice $\rho(y) = y^p$ (with $p \geq 1$) and several distance measures are considered, including quaternion, angular (geodesic) and chordal distances, where each metric is related to a particular parametrization of the rotation space. In [117] it is shown that smaller and well-connected graphs are easier than larger and noisy ones, based on a local convexity analysis. Further theoretical analysis is reported in [25] where Cramèr-Rao bounds for synchronization over $SO(d)$ are derived, namely lower bounds on the variance of unbiased estimators, assuming a certain noise model.

The spectral method was introduced in [99] for $\Sigma = SO(2)$ and extended in [100,3] to $\Sigma = SO(3)$. A related approach is adopted in [75], where the $\Sigma = SO(3)$ case is considered and the least eigenvectors of $D \otimes I_3 - Z_A$ are computed (instead of the least right singular vectors).

Remark 6 Note that both Z and Z_A are symmetric even in the presence of noise, since, by assumption, a group-labelled graph satisfies $Z_{ji} = Z_{ij}^{-1}$ for all $(i,j) \in \mathcal{E}$, which becomes $Z_{ji} = Z_{ij}^{\mathsf{T}}$ in the $\Sigma = O(d)$ case. The matrix $(D \otimes I_d)^{-1} Z_A$ is not symmetric, but it is similar to the *symmetric* matrix $(D \otimes I_d)^{-1/2} Z_A (D \otimes I_d)^{-1/2}$, thus its eigenvalues are real and it admits an orthonormal basis of real eigenvectors.

Ambiguity. The solution U is defined up to an element of $O(d)$, since a change of the orthonormal basis in the eigenspace (or null-space) corresponds to (right) multiplication by an orthogonal $d \times d$ matrix. This reduces to $SO(d)$ after permutation of columns of U such that $\det(U_1)$ is positive.

Projection. Each $d \times d$ block of U is not guaranteed to belong to $O(d)$ and has to be projected onto the group. Such projection can be obtained by solving an orthogonal Procrustes problem, e.g. via Singular Value Decomposition (SVD) [66]. Specifically, if $Q \in \mathbb{R}^{d\times d}$ is a given matrix, then the nearest orthogonal matrix (in the Frobenius norm sense) is given by

$$R = WV^{\mathsf{T}} \in O(d) \tag{38}$$

where $Q = WSV^{\mathsf{T}}$ denotes the singular value decomposition of Q. If the synchronization problem over $\Sigma = SO(d)$ is considered, the projection step is slightly different. Specifically, the nearest rotation matrix (in the Frobenius norm sense) is given by

$$R = W\,\mathrm{diag}([1,\dots,1,\det(WV^{\mathsf{T}})])V^{\mathsf{T}} \in SO(d). \tag{39}$$

Note that the final rounding step which is required for $GL(d)$, i.e., zeroing the imaginary part of the eigenvectors, is not necessary here due to Remark 6.

5.3 Synchronization over $GA(d)$

The General Affine Group $GA(d)$ is the set of invertible affine transformations in d-space, which admits a matrix representation through $(d+1)\times(d+1)$ matrices

$$GA(d) = \left\{ \begin{bmatrix} M & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix},\ s.t.\ M \in GL(d),\ \mathbf{t} \in \mathbb{R}^d \right\}. \tag{40}$$

Synchronization over $GA(d)$ is referred to as *affine synchronization* [18,92].

Ambiguity. The solution U is unique up to the action of an element of $GL(d+1)$, which can be reduced to $GA(d)$ by computing a linear combination of the columns of U such that the output matrix has the vector $[\mathbf{0}_{1\times d}\ 1]$ in rows multiple of $d+1$. More precisely, let $F \in \mathbb{R}^{n\times(d+1)n}$ be the 0/1-matrix such that $FU \in \mathbb{R}^{n\times(d+1)}$ consists of the rows of U with indices multiple of $d+1$. The coefficients $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{d+1}$ of the linear combination are solution of

$$FU\mathbf{a} = \mathbf{0}_{n\times 1}, \quad FU\mathbf{b} = \mathbf{1}_{n\times 1} \tag{41}$$

where the first equation has a d-dimensional solution space. If $\mathbf{a}_1,\dots,\mathbf{a}_d$ denotes a basis for the null-space of FU, then U is transformed into $U[\mathbf{a}_1,\dots,\mathbf{a}_d,\mathbf{b}]$. In the presence of noise, Equation (41) is solved in the least squares sense.

Projection. In order to project the solution onto $GA(d)$, the rows of U multiple of $d+1$ are forced to $[\mathbf{0}_{1\times d}\ 1]$, and the imaginary part of the eigenvectors is zeroed (in the case of the spectral solution).

5.4 Synchronization over $SE(d)$

The Special Euclidean Group $SE(d)$ is the set of direct isometries (or rigid motions) in d-space, which admits a matrix representation through $(d+1)\times(d+1)$ matrices

$$SE(d) = \left\{ \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix},\ s.t.\ R \in SO(d), \mathbf{t} \in \mathbb{R}^d \right\}. \tag{42}$$

Synchronization over $SE(d)$ is also known as *rigid-motion synchronization* [11,18] or *motion averaging* [53,55] or *pose-graph optimization* [34]. The spectral and null-space solutions can be regarded as the extension of the rotation synchronization approach introduced in [99,100,3], and were developed independently by [18] and [11].

Ambiguity. The solution U is unique up to the action of an element of $GL(d+1)$. In order to reduce the ambiguity up to an element of $SE(d)$ we must design a suitable transformation that maps, e.g., the first block of U onto an element of $SE(d)$. In order to do that let us apply a linear combination of the columns of U such that the $(d+1)^{th}$ row becomes $[\mathbf{0}_{1\times d}\ 1]$, as in the $\Sigma = GA(d)$ case, and subsequently transform U such that U_1, the first $d\times d$ block of U, becomes an element of $SO(d)$. Let $U_1 = RP$ be the polar decomposition of U_1, with $R \in O(d)$ and P symmetric positive definite. Since U_1 is invertible, then $R = U_1P^{-1}$, and P^{-1} is the sought transformation. In the presence of noise, a least square solution that brings every row multiple of $d+1$ as close as possible to $[\mathbf{0}_{1\times d}\ 1]$ is sought, as in the affine case.

Projection. In order to project the solution onto $SE(d)$ – as in [16] – the rows of U multiple of $d+1$ are forced to $[\mathbf{0}_{1\times d}\ 1]$ and the $d\times d$ rotation blocks are projected onto $SO(d)$. In the case of the spectral solution, before performing such projection, the imaginary part of the eigenvectors is zeroed.

Two-step synchronization. Since the Special Euclidean Group is the the semi-direct product of $SO(d)$ and $\mathbb{R}^d$, synchronization over $SE(d)$ can be alternatively addressed by breaking the problem into rotation and translation and solving the two sub-problems separately, as done (e.g.) in [6]. Let $X_i \in SE(d)$ denote the (unknown) label of node i

$$X_i = \begin{bmatrix} R_i & \mathbf{t}_i \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \tag{43}$$

where $R_i \in SO(d)$ and $\mathbf{t}_i \in \mathbb{R}^d$ denote the rotation and translation components of the rigid motion, respectively. Similarly, each edge label $Z_{ij} \in SE(d)$ can be

expressed as

$$Z_{ij} = \begin{bmatrix} R_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \tag{44}$$

with $R_{ij} \in SO(d)$ and $\mathbf{t}_{ij} \in \mathbb{R}^d$. Using this notation, the consistency constraint for synchronization over $SE(d)$, namely $Z_{ij} = X_i X_j^{-1}$, can be equivalently rewritten as

$$R_{ij} = R_i R_j^\mathsf{T} \tag{45}$$
$$\mathbf{t}_{ij} = -R_i R_j^\mathsf{T} \mathbf{t}_j + \mathbf{t}_i. \tag{46}$$

Note that Equation (45) defines a rotation synchronization problem, thus the rotation components of the unknown vertex labels can be recovered as explained in Section 5.2. Equation (46) can be equivalently written as

$$-R_i^\mathsf{T} \mathbf{t}_{ij} = R_j^\mathsf{T} \mathbf{t}_j - R_i^\mathsf{T} \mathbf{t}_i = \mathbf{x}_i - \mathbf{x}_j \tag{47}$$

using the substitution $\mathbf{x}_i = -R_i^\mathsf{T} \mathbf{t}_i$. Thus – assuming that rotations have been computed beforehand – recovering the translation components of the vertex labels can be reduced to a translation synchronization problem (as defined in Section 3), where the edge labels are given by $\mathbf{z}_{ij} = -R_i^\mathsf{T} \mathbf{t}_{ij}$.

5.5 Synchronization over $\mathcal{S}_d$

The Symmetric Group $\mathcal{S}_d$ is the set of bijections between d objects, which admits a matrix representation through $d \times d$ permutation matrices. A permutation matrix is such that exactly one entry in each row and column is equal to 1 and all other entries are 0. Synchronization over $\mathcal{S}_d$ is also known as *permutation synchronization*, which finds application in multi-view matching (see Section 8.2). The spectral solution was introduced in [85] for a complete graph (based on [99]) and subsequently extended in [98] to the case of missing data.

Ambiguity. The solution U is defined up to an element of $O(d)$, as a consequence of Remark 6. However, the solution to permutation synchronization is inherently defined up to an element of $\mathcal{S}_d$. Let Q be the unknown orthogonal transformation such that $X = UQ$, where X is the solution, a matrix whose blocks are in $\mathcal{S}_d$. Let U_1 be the first d rows of U, then from $X_1 = U_1 Q$ we have $Q = U_1^\mathsf{T} X_1$, i.e., $Q = U_1^\mathsf{T}$ up to a permutation. U_1 is indeed orthogonal because $U_1 U_1^\mathsf{T} = X_1 X_1^\mathsf{T} = I_d$.

Projection. Even after fixing the ambiguity, the blocks of U will not be permutation matrices (in general). We need to project them onto $\mathcal{S}_d$ by solving the so-called permutation Procrustes problem, e.g., with the Kuhn-Munkres algorithm [71]. Note that the spectral solution returns real eigenvectors, as in the $\Sigma = O(d)$ case.

6 Synchronization over $\mathcal{I}_d$

Let us consider now the Symmetric Inverse Semigroup $\mathcal{I}_d$, that is the set of bijections between (different) subsets of d objects, which admits a matrix representation through $d \times d$ partial permutation matrices. A *partial* permutation matrix – which represents matches between different objects (see Figure 6) – has at most one nonzero entry in each row and column, and these nonzero entries are all 1. The synchronization problem over $\mathcal{I}_d$ is also known as *partial permutation synchronization* [8].

The set $\mathcal{I}_d$ is an inverse monoid with respect to matrix multiplication (and a subsemigroup of $\mathcal{S}_d$) where the inverse is given by matrix transposition (see Definition 6). Let $X_i \in \mathcal{I}_d$ denote the (unknown) label of vertex i and let $Z_{ij} \in \mathcal{I}_d$ denote the (known) label of edge $(i, j) \in \mathcal{E}$, which are linked by the consistency constraint

$$Z_{ij} = X_i X_j^\mathsf{T}. \tag{48}$$

If $[X_i]_{(h,k)} = 1$ for some index h we say that "node i sees object k". Note that $X_i^\mathsf{T} X_i$ is not equal, in general, to the identity, unless $X_i \in \mathcal{S}_d$. Indeed, $[X_i^\mathsf{T} X_i]_{(k,k)} = 1$ if node i sees object k and $[X_i^\mathsf{T} X_i]_{(k,k)} = 0$ otherwise. However, it can be checked that $X_i^\mathsf{T} X_i \leq I_d$.

Remark 7 If $\{(i_1, i_2), (i_2, i_3), \ldots, (i_\ell, i_1)\}$ denotes a circuit in $\mathcal{G}$, then – assuming that the consistency constraint (48) holds – the composition of edge labels along its edges becomes

$$\begin{aligned} &Z_{i_1 i_2} Z_{i_2 i_3} \cdot \ldots \cdot Z_{i_\ell i_1} = \\ &= X_{i_1} \underbrace{X_{i_2}^\mathsf{T} X_{i_2}}_{\leq I_d} X_{i_3}^\mathsf{T} \cdot \ldots \cdot X_{i_\ell} X_{i_1}^\mathsf{T} \leq I_d. \end{aligned} \tag{49}$$

In contrast to the case of total permutations, where synchronization implies that compositions of edge labels over circuits must be equal to the identity (see Corollary 1), in the case of partial permutations we obtain that compositions of edge labels over circuits must be a subset of the identity, as observed also in [19]. In other words, with reference to the multi-view matching application (see Section 8.2), due to potential missing matches (i.e. zero rows/columns in Z_{ij}) along a cyclic path, some of the original matches may vanish, and only those matches that are seen in all the images survive.

6.1 Spectral solution

Despite the fact that the group structure is missing[3], it can be shown that a spectral solution can be derived under the assumption that the graph is complete. Let X and Z be two block-matrices containing the vertex labels and edge labels respectively – defined as in (19) – so that the consistency constraint becomes $Z = XX^\mathsf{T}$ with Z of rank d. Note that here the diagonal of Z is not filled with identity matrices, in general. When all the objects seen by node i are different from those seen by node j we have $X_i X_j^\mathsf{T} = 0$, resulting in a zero block in Z.

Proposition 6 ([8]) *The columns of X are d (orthogonal) eigenvectors of Z and the corresponding eigenvalues are contained in the diagonal of the following $d \times d$ matrix*

$$V := X^\mathsf{T} X = \sum_{i=1}^{n} X_i^\mathsf{T} X_i. \tag{50}$$

Proof Using (50) and the consistency constraint, we obtain

$$ZX = XV \tag{51}$$

which is a spectral decomposition, i.e. the columns of X are d eigenvectors of Z and the corresponding eigenvalues are on the diagonal of V. Recall that Z admits an orthonormal basis of real eigenvectors since it is symmetric. □

Although $\mathcal{I}_d$ is not a group, an eigenvalue decomposition problem has been obtained, where the non-zero eigenvalues are contained in the diagonal of V. Specifically, the k-th eigenvalue counts how many nodes see object k, thus all the eigenvalues are integer numbers lower than or equal to n. This implies that, when the number of objects is larger than the number of nodes (i.e., $d > n$) – which is likely to happen in the multi-view matching application – the eigenvalues are repeated. In the case of total permutations (i.e. $\Sigma = \mathcal{S}_d$) all the nodes see all the objects, thus $V = nI_d$ and all the eigenvalues are equal, hence Equation (51) reduces to (24). In the presence of noise, the eigenvectors of Z corresponding to the d largest eigenvalues are computed.

This result holds for a complete graph. When it is not, Proposition 6 can not be straightforwardly extended as we did for $\Sigma = GL(d)$ case.

[3] Note that $0 \in \mathcal{I}_d$, thus we can not distinguish between the case of missing measures and the case of missing correspondences between nodes, as it was the case for the subgroups of $GL(d)$.

Remark 8 Equation (51) could also be expressed as a null-space problem, but in that case the matrix V, which is unknown, have to be estimated somehow.

Ambiguity. Note that the reverse of Proposition 6 is not true in general, i.e., the matrix U is not necessarily equal to the vertex labelling X. Indeed, U is not uniquely determined if the eigenvalues of Z are repeated. So we have to face the problem of how to select, among the infinitely many Us, the one that resembles X, a matrix composed of partial permutations. Note that the reasoning reported in Section 5.5 does not apply here, since the first block U_1 is not orthogonal in general, due to the presence of zero rows. A key observation is reported in the following proposition, suggesting that such a problem can be solved via clustering techniques.

Proposition 7 ([8]) *Let U be the $nd \times d$ matrix composed by the d leading eigenvectors of Z; then U has $d+1$ different rows (in the absence of noise). One of these is the zero row.*

Proof Let $\lambda_1, \lambda_2, \dots, \lambda_\ell$ denote all the *distinct* eigenvalues of Z (with $\ell \leq d$), and let $m_1, m_2, \dots, m_\ell$ be their multiplicities such that $\sum_{k=1}^{\ell} m_k = d$. Let U_{λ_k} denote the m_k columns of U corresponding to the eigenvalue λ_k, and let X_{λ_k} be the corresponding columns of X. Up to a permutation of the columns, we have

$$U = [U_{\lambda_1}\, U_{\lambda_2}\, \dots\, U_{\lambda_\ell}], \quad X = [X_{\lambda_1}\, X_{\lambda_2}\, \dots\, X_{\lambda_\ell}]. \tag{52}$$

Since U_{λ_k} and X_{λ_k} are (orthogonal) eigenvectors corresponding to the same eigenvalue, there exists an orthogonal matrix $Q_k \in \mathbb{R}^{m_k \times m_k}$ representing a change of basis in the eigenspace of λ_k, such that $U_{\lambda_k} = X_{\lambda_k} Q_k$. In matrix form this rewrites

$$U = X \underbrace{\operatorname{blkdiag}(Q_1, Q_2, \dots, Q_\ell)}_{Q}. \tag{53}$$

Note that the rows of X are the rows of I_d plus the zero row. Since Q is invertible (hence injective), $U = XQ$ has only $d+1$ different rows as well. □

Specifically, an estimate of the vertex labelling can be obtained by clustering the rows of U into $d+1$ clusters (e.g. with k-means), then assigning the centroid which is closest to zero to the zero row, and arbitrarily assigning each of the other d centroids to a row of I_d. This arbitrary assignment corresponds to the fact that the solution to partial permutation synchronization is defined up to an element of $\mathcal{S}_d$.

Projection. Even after fixing the ambiguity, valid permutation matrices may not be obtained. Indeed, since there are no constraints in the clustering phase, it may happen that different rows of a $d \times d$ block in U are assigned to the same cluster, resulting in more than one entry per column equal to 1. For this reason, for each $d \times d$ block in U, the partial permutation matrix that best maps such block into the set of centroids has to be computed (e.g. via the Kuhn-Munkres algorithm [71]), and such permutation is output as the sought solution.

7 Spectral and other relaxations

In this section we concentrate mainly on the $\Sigma = O(d)$ case. We frame the spectral solution as an instance of a constraint relaxation pattern by setting it side by side with rank relaxation [7,10,124] and semidefinite programming [99,37,90], which are not closed-form, though. Moreover we highlight the link between the spectral solution and the consistency error of the synchronization problem (which is available only for $\Sigma = O(d)$).

Note that this is a special case, since the inverse equals matrix transposition, thus the consistency constraint rewrites $Z_{ij} = X_i X_j^\mathsf{T}$ which, if the graph is complete, is equivalent to

$$Z = XX^\mathsf{T}. \tag{54}$$

Such a decomposition implies that, if $\epsilon = 0$, the matrix Z is *symmetric and positive semidefinite*, besides being low-rank. In the case of missing edges, the consistency constraint translates into

$$Z_A = (XX^\mathsf{T}) \circ (A \otimes \mathbf{1}_{d\times d}) \tag{55}$$

and the consistency error of the synchronization problem becomes

$$\begin{aligned}\epsilon(X) &= \sum_{(i,j)\in\mathcal{E}} \|Z_{ij} - X_i X_j^\mathsf{T}\|_F^2 = \\ &= \|Z_A - (XX^\mathsf{T}) \circ (A \otimes \mathbf{1}_{d\times d})\|_F^2.\end{aligned} \tag{56}$$

In particular, since the Frobenius norm of a matrix can be defined in terms of its trace, Equation (56) can be expressed as

$$\begin{aligned}&\epsilon(X) = \\ &= \sum_{i,j=1}^{n} \mathrm{tr}(Z_{ij}^\mathsf{T} Z_{ij}) + \mathrm{tr}(X_j X_i^\mathsf{T} X_i X_j^\mathsf{T}) - 2\mathrm{tr}(X_i^\mathsf{T} Z_{ij} X_j) = \\ &= \sum_{i,j=1}^{n} \mathrm{tr}(Z_{ij}^\mathsf{T} Z_{ij}) + d - 2\mathrm{tr}(X_i^\mathsf{T} Z_{ij} X_j)\end{aligned} \tag{57}$$

where a complete graph is considered and the last equality holds since $X_i^\mathsf{T} X_i = I_d = X_j^\mathsf{T} X_j$. Therefore

$$\begin{aligned}\min_{X\in O(d)^n} \epsilon(X) \Longleftrightarrow \max_{X_1,\dots,X_n \in O(d)} \sum_{i,j=1}^{n} \mathrm{tr}(X_i^\mathsf{T} Z_{ij} X_j) = \\ \max_{X\in O(d)^n} \mathrm{tr}(X^\mathsf{T} Z X).\end{aligned} \tag{58}$$

Solving the synchronization problem over $O(d)$ is difficult since the feasible set is non-convex, and the cost function may have multiple local minima in different regions of attraction, as shown in [58]. Several relaxations will be considered.

7.1 Spectral relaxation

Let us start with the case of a complete graph and let us consider the following minimization problem

$$\min_{X^\mathsf{T}X = nI_d} \|Z - XX^\mathsf{T}\|_F^2 \tag{59}$$

where the columns of X are constrained to be orthogonal rather than imposing that each $d \times d$ block in X belongs to $O(d)$. This is called the *spectral relaxation*, and, reasoning as in Equation (58), it can be shown to be equivalent to the following generalized Rayleigh problem

$$\max_{X^\mathsf{T}X = nI_d} \mathrm{tr}(X^\mathsf{T} Z X). \tag{60}$$

Proposition 8 *Equation* (60)*, with Z symmetric, admits a closed-form solution given by the d leading eigenvectors of Z.*

Proof Let $\mathcal{F}$ be the unconstrained cost function corresponding to problem (60), namely

$$\mathcal{F}(X) = \mathrm{tr}(X^\mathsf{T} Z X) + \mathrm{tr}(\Lambda(X^\mathsf{T}X - nI_d)) \tag{61}$$

where $\Lambda \in \mathbb{R}^{d\times d}$ is a symmetric matrix of unknown Lagrange multipliers. Setting to zero the partial derivatives of $\mathcal{F}$ with respect to X we obtain

$$\frac{\partial \mathcal{F}}{\partial X} = 2ZX + 2X\Lambda = 0 \Rightarrow ZX = -X\Lambda. \tag{62}$$

Let $\mathbf{u_i}$ denote d eigenvectors of Z (normalized so that $\|\mathbf{u_i}\| = \sqrt{n}$) for $i = 1,\dots,d$ and let λ_i be the corresponding eigenvalues. Then $X = [\mathbf{u_1}\ \mathbf{u_2}\ \dots\ \mathbf{u_d}]$ satisfies both (62) and the constraint $X^\mathsf{T}X = nI_d$, with $\Lambda = -\mathrm{diag}([\lambda_1, \lambda_2, \dots, \lambda_d])$. In other words, any set of d eigenvectors is a stationary point for the objective function $\mathcal{F}$. The corresponding stationary value is given by $n(\lambda_1 + \lambda_2 + \dots \lambda_d)$, hence the maximum is attained if $\mathbf{u_i}$ are the eigenvectors of Z corresponding to the d largest eigenvalues. □

If the graph is not complete, we consider a different definition for the consistency error

$$\epsilon(X) = \sum_{(i,j)\in\mathcal{E}} [D]_{ii}^{-1} \|Z_{ij} - X_i X_j^{\mathsf{T}}\|_F^2 \tag{63}$$

in which the term associated to the edge (i,j) is weighted with the inverse of the degree of node i. Reasoning as before we get

$$\begin{aligned} &\min_{X\in O(d)^n} \epsilon(X) \iff \\ &\max_{X_1,\dots,X_n\in O(d)} \sum_{(i,j)\in\mathcal{E}} [D]_{ii}^{-1} \mathrm{tr}(X_i^{\mathsf{T}} Z_{ij} X_j) = \\ &\max_{X\in O(d)^n} \mathrm{tr}(X^{\mathsf{T}}(D\otimes I_d)^{-1} Z_A X) \end{aligned} \tag{64}$$

which, if the spectral relaxation is adopted, becomes

$$\max_{X^{\mathsf{T}}X = nI_d} \mathrm{tr}(X^{\mathsf{T}}(D\otimes I_d)^{-1} Z_A X) \tag{65}$$

whose solution is given by the d leading eigenvectors of $(D\otimes I_d)^{-1} Z_A$.

We have shown, following [3], that the spectral solution minimizes the consistency error of the synchronization problem under relaxed constraints. Note that this result is due to the special structure of $O(d)$, it also holds for subgroups of $O(d)$, and it is not available in general for synchronization over $GL(d)$.

Concerning the Symmetric Inverse Semigroup, since Z is symmetric, Proposition 8 holds. However, the computation above linking Problem (60) to the synchronization cost function is no longer valid since the equation $X_i^{\mathsf{T}} X_i = I_d = X_j^{\mathsf{T}} X_j$ does not hold for $\Sigma = \mathcal{I}_d$.

7.2 Rank relaxation

Let $\Omega = A \otimes \mathbf{1}_{d\times d}$ denote the *pattern* (also known as the *sampling set*) of Z_A, that is the index set of available entries. Using this notation, the synchronization problem can be expressed as

$$\min_{X\in\Sigma^n} \epsilon(X) = \min_{X\in\Sigma^n} \|(Z - XX^{-\flat})\circ\Omega\|_F^2 \iff \tag{66}$$

$$\min_{\widetilde{Z}} \|(Z-\widetilde{Z})\circ\Omega\|_F^2 \quad s.t.\ \widetilde{Z} = XX^{-\flat},\ X\in\Sigma^n \tag{67}$$

where the problem of finding a consistent vertex labelling X is reduced to that of finding an edge labelling $\widetilde{Z}$ induced by X.

If the *rank relaxation* is adopted [12], i.e. the matrix $\widetilde{Z}$ is enforced to have rank (at most) d (while the remaining properties on $\widetilde{Z}$ are not enforced), then Problem (67) becomes

$$\min_{\widetilde{Z}} \|(Z-\widetilde{Z})\circ\Omega\|_F^2 \quad s.t.\ \mathrm{rank}(\widetilde{Z}) \le d \tag{68}$$

which is a *matrix completion* problem [29], that is the problem of recovering a low-rank matrix starting from an incomplete subset of its entries (possibly corrupted by noise), which can be solved via (e.g.) the OptSpace algorithm [68].

In order to handle outliers, a *robust matrix completion* framework can be considered instead of (68), namely

$$\begin{aligned} &\min_{\widetilde{Z},S} \|(Z-\widetilde{Z}-S)\circ\Omega\|_F^2 \\ &s.t.\ \mathrm{rank}(\widetilde{Z}) \le d,\ S \text{ is sparse in } \Omega \end{aligned} \tag{69}$$

where the additional variable S represents outliers, which are sparse over the measurement graph (by assumption). Available algorithms to solve problem (69) include R-Godec [7], Grasta [59] and L1-Alm [123]. This approach was introduced in [7] for $\Sigma = SO(d)$ and extended in [10] to $\Sigma = SE(d)$.

In the $\Sigma = \mathcal{S}_d$ case, however, the optimization variable $\widetilde{Z}$ is sparse, being composed of binary matrices, and hence it does not satisfy the incoherence assumption (see [29]) that make "generic" matrix completion algorithms work in practice. To overcome this drawback, the authors of [124] consider the following problem instead of (68)

$$\begin{aligned} &\min_{\widetilde{Z}} \|(Z-\widetilde{Z})\circ\Omega\|_F^2 + \alpha\|\widetilde{Z}\|_1 \\ &s.t.\ \mathrm{rank}(\widetilde{Z}) \le d,\ 0 \le \widetilde{Z} \le 1 \end{aligned} \tag{70}$$

where the regularization term $\alpha\|\widetilde{Z}\|_1$ is included to promote a sparse solution, and the optimization variable is enforced to lie in the interval $[0,1]$ (while the binary constraints are not enforced). The resulting cost function is minimized via the Alternating Direction Method of Multipliers (ADMM) [27].

7.3 Semidefinite relaxation

Let us consider the $\Sigma = O(d)$ case. Reasoning as in Equation (67), we can express the synchronization problem as

$$\min_{\widetilde{Z}} \|(Z-\widetilde{Z})\circ\Omega\|_F^2 \quad s.t.\ \widetilde{Z} = XX^{\mathsf{T}},\ X\in\Sigma^n. \tag{71}$$

If the *semidefinite relaxation* is employed [99,100,3], i.e. the optimization variable $\widetilde{Z}$ is constrained to be symmetric positive semidefinite and covered by identity blocks along its diagonal (while the remaining properties on $\widetilde{Z}$ are not enforced), then Problem (71) reduces to a semidefinite program

$$\min_{\widetilde{Z}} \|(Z-\widetilde{Z})\circ\Omega\|_F^2 \quad s.t.\ \widetilde{Z} \succeq 0, \widetilde{Z}_{ii} = I_d \tag{72}$$

which can be solved (e.g.) though interior point methods [119]. In [116] the ℓ_1-norm is used in (72) in place of the ℓ_2-norm, exploiting the fact that the former is more robust to outliers than the latter.

In order to deal with the $\Sigma = \mathcal{S}_d$ case, the authors of [37] introduce in (72) the non-negative constraint $\widetilde{Z} \geq 0$ and a sparsifying regularization term $\alpha\|\widetilde{Z}\|_1$, similarly to (70), resulting in the following problem

$$\begin{aligned} &\min_{\widetilde{Z}} \|(Z - \widetilde{Z}) \circ \Omega\|_F^2 + \alpha\|\widetilde{Z}\|_1 \\ &s.t.\ \widetilde{Z} \succeq 0,\ \widetilde{Z}_{ii} = I_d,\ \widetilde{Z} \geq 0 \end{aligned} \tag{73}$$

which is solved via the ADMM algorithm.

Concerning the Special Euclidean Group, in [90] a cost function tightly related to (67) is considered and the feasible set $SE(d)$ is relaxed to its convex hull, which admits a semidefinite representation [93]. A convex relaxation is also employed in [89,32] where the authors, using the theory of Lagrangian duality, develop an algorithm for certifying the global optimality of a candidate solution to rigid-motion synchronization.

It should be noted that although semidefinite relaxation solutions are not closed-form, they are nonetheless convex problems, which are relatively easy to compute.

8 Synchronization in Computer Vision

Among the several applications that have been mentioned in the introduction (image mosaicking, multiview matching, synchronization, etc.) we concentrate here on the few ones where the formulation of the problem in terms of synchronization might require some clarification.

8.1 Homography synchronization

When elements of $SL(d)$ are identified with homographies of the $(d-1)$-dimensional projective space, synchronization over $SL(3)$ can be easily seen as a convenient way to align multiple images into a mosaic, starting from pairwise homographies. This works because any real non-singular 3×3 matrix can be scaled to a *real* unit determinant matrix, as first mentioned in [95]. The same trick does not apply when d is even because of complex roots.

8.2 Permutation synchronization

Consider a set of d objects, which is attached to each node in the measurement graph in a random order, i.e., each node has its own local labelling of the objects with integers in the range $\{1, \dots, d\}$, represented as a permutation. It is assumed that pairs of nodes can match these objects, establishing which objects are the same in the two nodes, despite the different naming, and the goal is to infer a global labelling of the objects, such that the same object receives the same label in all the nodes (see Figure 5).

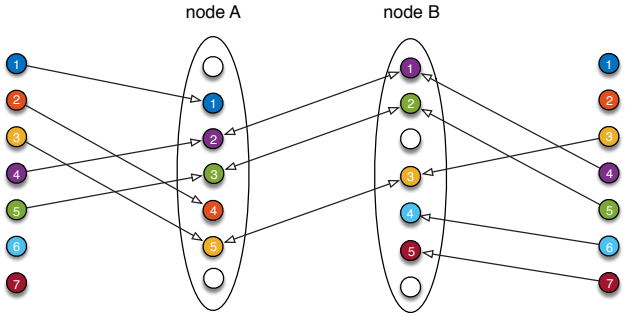

Fig. 5: In the center, two nodes with partial visibility match their three common objects. At the extrema the ground truth ordering of the objects. Each node sees some of the objects (white circles are missing objects) and puts them in a different order, i.e., it gives them different numeric labels.

A more concrete problem statement can be given in terms of *multi-view matching* [37,124,76], where nodes are *images* and objects are *features*. A set of matches between pairs of images is computed in isolation, and the goal is to jointly update them so as to maximize their consistency. In general, not all the features are visible (or matchable) in all the images, so each matching is modelled as a partial permutation (Figure 6).

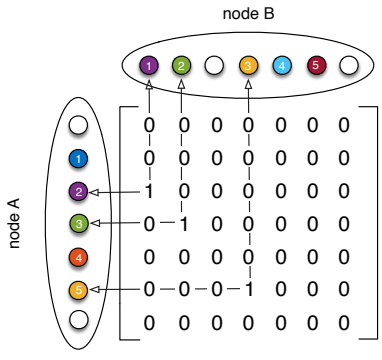

Fig. 6: Partial permutation matrix representing the matches between two nodes.

Specifically, the permutation matrix P representing the matching between node B and node A is con-

structed as follows:

$$[P]_{h,k} = \begin{cases} 1 \text{ if obj. } k \in \text{B is matched with obj. } h \in A; \\ 0 \text{ otherwise.} \end{cases}$$

If row $[P]_{h,\cdot}$ contains only zeros, then object h in node A does not have a matching object in node B. If column $[P]_{\cdot,k}$ is made of zeros, then object k in node B does not have a matching object in node A.

Let Z_{ij} denote the permutation matrix representing the matching between node j and node i, and let X_i (resp. X_j) denote the unknown permutation that reveals the true identity of the objects in node i (resp. j). It can be easily verified that $Z_{ij} = X_i X_j^\mathsf{T}$, thus the problem of finding the global labelling of the objects can be modelled as a synchronization problem over the Symmetric Inverse Semigroup $\mathcal{I}_d$ (if permutations are partial), or over the Symmetric Group $\mathcal{S}_d$ (if permutations are total).

8.3 Rigid-motion Synchronization

Elements of $SE(d)$ represent the angular attitude and position of a d-dimensional reference frame. These two properties are collectively referred to as *motion* in Computer Vision, *orientation* in Photogrammetry, or *pose* in Robotic (although pose is also used in Vision). Synchronization over $SE(d)$ is tantamount to recovering the location and attitude of a set of reference frames organized in a network, where the links of this network are relative transformations of one frame with respect to (some of) the others. This is also called *network orientation* [46], *pose graph optimization* [34], or *sensor network localization* [41]. If we restrict the attention to the angular attitude (leaving out the position) then we get a rotation synchronization. Similarly, if position only is considered, it results in translation synchronization.

Such local frames can be local coordinates where $3D$ points are represented, in which case we are dealing with *multiple point-set registration* [87], or camera reference frames, in which case we are in the context of (global) *structure-from-motion* [84].

In the first case, the goal is to find the rigid transformation that brings multiple ($n > 2$) 3D point sets into alignment. The problem can be solved in *point space* or in *frame space*. In the former case all the transformations are simultaneously optimized with respect to a cost function that depends on the distance between corresponding points [86,17,70,105,45]. In the latter case the optimization criterion is related to the internal coherence of the network of transformations applied to the local coordinate frames [97,48,107,56,10]. This is exactly an instance of rigid-motion synchronization, as shown in Figure 7, where the input edge labels are typically computed via the Iterative Closest Point algorithm [20].

In the structure from motion application the goal is to recover both scene structure (3D coordinates of scene points) and camera motion starting from a set of images. In this case $\mathcal{G}$ is known as the *epipolar graph* [81,5] or the *viewing graph* [72] (see Figure 8).

Structure from motion is a well studied problem that can be addressed in several ways (see the recent survey by Ozyesil et al. [84]). One stream of research focuses on methods that compute the motion *before* recovering the structure, and in this paper we are specifically interested in frame-space methods, that do not make use of points in order to compute the global motion. These methods are usually faster than sequential and hierarchical methods (e.g. [101,94,106]), while ensuring a fair distribution of the errors among the cameras, being global. On the other hand, accuracy is usually worse than that achieved by point-space methods, such as bundle adjustment.

We are interested in these "structure-free" methods because they reduce to a rigid-motion synchronization, modulo the fact that the relative translations are only known as directions, as the magnitude is unknown.

8.3.1 Localization

We consider here the two-step formulation of synchronization over $SE(3)$, described in Section 5.4. Since the magnitude of translations are unknown, we are required to estimate such magnitudes either directly or indirectly (i.e., by computing camera locations from the directions only). We will only hint at these solutions here, since they fall outside the domain of synchronization, although being related to it. As a matter of fact, the starting point is the translation synchronization equation

$$(B^\mathsf{T} \otimes I_3) \operatorname{vec}(X) = \operatorname{vec}(Z) \tag{74}$$

and from here two paths can be followed:

- first recover the *magnitude* of translations [5,108] and then solve a *translation synchronization*;
- solve the problem straight from the direction information: *bearing-only localization* [28,122].

Direct computation of magnitudes. If the relative measures $\mathbf{z}_{ij}$ are expanded into magnitude $\alpha_{ij} = \|\mathbf{z}_{ij}\|$ and direction $\mathbf{u}_{ij} = \mathbf{z}_{ij}/\|\mathbf{z}_{ij}\|$, then the matrix Z rewrites

$$Z = U \operatorname{diag}(\boldsymbol{\alpha}) \tag{75}$$

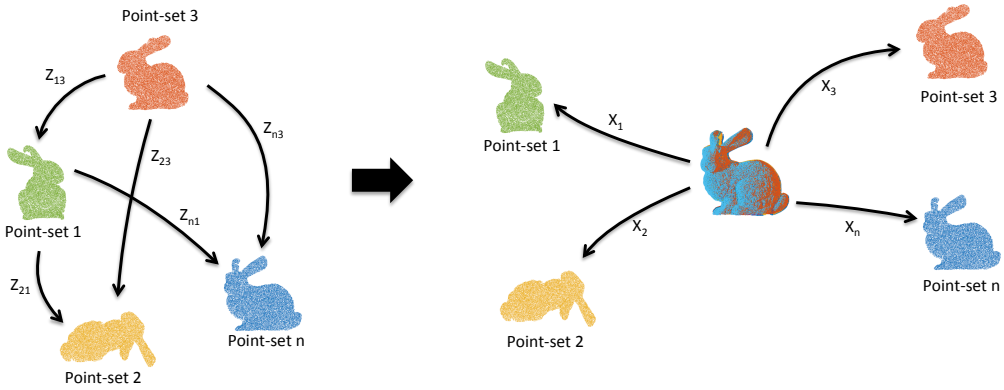

Fig. 7: The goal of the multiple point-set registration problem is to find the rigid transformations that bring multiple 3D point sets into alignment, where each rigid transformation is represented by a direct isometry.

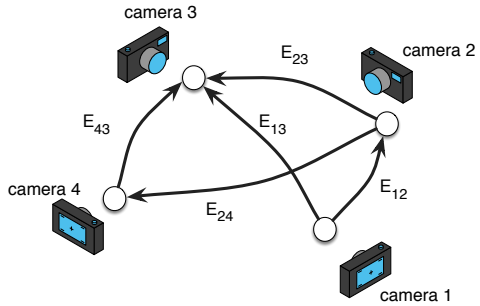

Fig. 8: The epipolar graph: each node represents one camera and each edge is labelled with the essential matrix of the corresponding camera pair, which encodes the relative rotation and translation (up to scale).

where U denotes the $3\times m$ matrix obtained by juxtaposing all the edge directions, namely $U = [\mathbf{u}_{12} \ \dots \ \mathbf{u}_{ij} \ \dots]$, and $\boldsymbol{\alpha} \in \mathbb{R}^m$ denotes the vector containing all the *unknown* magnitudes, namely $\boldsymbol{\alpha} = [\alpha_{12} \ \dots \ \alpha_{ij} \ \dots]^\mathsf{T}$. Thus, using Property (92), we get

$$\operatorname{vec}(Z) = (I_m \odot U)\boldsymbol{\alpha} \tag{76}$$

where $\odot$ denotes the Khatri-Rao product (see Appendix A). Hence, Equation (74) rewrites

$$(B^\mathsf{T} \otimes I_3) \operatorname{vec}(X) = (I_m \odot U)\boldsymbol{\alpha}. \tag{77}$$

This equation is also called the *edge-based bearing constraint* in [108]. Let us consider the cycle matrix $C \in \{-1,0,1\}^{(m-n+1)\times m}$ associated with a cycle basis of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and let us multiply left and right sides in (77) by $(C \otimes I_3)$

$$(C \otimes I_3)(B^\mathsf{T} \otimes I_3) \operatorname{vec}(X) = (C \otimes I_3)(I_m \odot U)\boldsymbol{\alpha}. \tag{78}$$

Using Properties (86), (91) and (108) we get

$$(CB^\mathsf{T} \otimes I_3) \operatorname{vec}(X) = \mathbf{0} = (C \odot U)\boldsymbol{\alpha}. \tag{79}$$

Note that Equation (79) is nothing else but the cycle consistency (Equation (17)) rewritten in terms of directions and magnitudes.

Proposition 9 ([5, 108]) *The unknown translation magnitudes can be uniquely (up to a global scale) recovered if and only if* $\operatorname{rank}(C \odot U) = m - 1$. *In this case the solution is given by the 1-dimensional null-space of* $C \odot U$.

Bearing-only localization. Let us multiply (74) by the block-diagonal matrix

$$S = \operatorname{blkdiag}\left(\{[\mathbf{u}_{ij}]_\times\}_{(i,j)\in\mathcal{E}}\right) \tag{80}$$

where $[\mathbf{u}_{ij}]_\times$ denotes the 3×3 skew-symmetric matrix corresponding to the cross-product with $\mathbf{u}_{ij}$, yielding

$$S(B^\mathsf{T} \otimes I_3) \operatorname{vec}(X) = S \operatorname{vec}(Z) = \mathbf{0}. \tag{81}$$

This step has the effect of substituting Z, which is unknown, with S (derived from U) which is known instead. Expanding this equation for a single edge (i, j) of the graph yields the more custom expression:

$$\mathbf{u}_{ij} \times (\mathbf{x}_i - \mathbf{x}_j) = \mathbf{0}. \tag{82}$$

This equation is also called the *node-based bearing constraint* in [108]. Its solution yields the node locations X, hence implicitly recovering the magnitudes. It is observed in [67] that least squares solution of (81), which is used (e.g.) in [52], is equivalent to the method presented in [28]. A bearing-only formulation is also adopted in [81,61,118,110,83,50] within the context of structure from motion.

This last section led us to glimpse the problem of *localization*. The goal of localization is to compute the

position of n nodes in d-space given measures on the edges. As in a translation synchronization problem the nodes correspond to positions, i.e., elements of $\mathbb{R}^d$, but the available measures are *not* differences of states, i.e. translations, for they can be directions (bearings) or distances. Figure 9 hopefully helps to clarify this relationship.

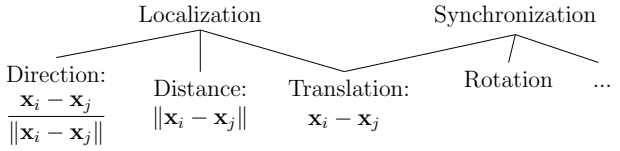

Fig. 9: The relationship between synchronization and localization.

The problem of localization from directions and the conditions under which an epipolar graph is *localizable* are discussed with continuity of notation by the same authors of this paper in [4].

9 Discussion

In this section we report some considerations about the pros and cons of the closed-form solutions reviewed in this paper.

9.1 Simplicity

These methods are particularly attractive for their simplicity. As an example we report the Matlab code of rotation and translation synchronization in 3-space (Listings 1 and 2).

In particular, rotation synchronization can be taken as a prototype of synchronization in any subgroup of $GL(d)$: the only step that changes is the projection. In fact, the removal of the ambiguity with division by the first block works in all cases, since the identity belongs to any subgroup.

This does not hold for $\mathcal{I}_d$, which is not a subgroup of $GL(d)$. In this case, removal of the ambiguity reduces to a clustering problem, as explained in Section 6. See also [112] for the relation between clustering and partial permutation synchronization. Note that synchronization over $\mathcal{I}_d$ requires to know the value of d in advance, which is not available in the multi-view matching application, since it corresponds to the total number of features (or tracks) present in the images. To overcome this drawback, a practical solution to multi-view matching is developed in [76], where the authors compute refined edge labels (instead of vertex labels), in order to avoid the ambiguity issue.

Listing 1: Rotation synchronization

```
function R = rotation_synch(Z,A)
  n = size(A,1);
  iD = diag(1./sum(A,2));
  [Q,~]=eigs( kron(iD,eye(3))*Z, 3);

  % Remove ambiguity
  Q = Q/(Q(1:3, 1:3));

  % Projection onto SO(3)
  R=cell(1,n);
  for i=1:n
    [U,~,V] = svd(Q(3*i-2:3*i,:));
    R{i} = U*diag([1,1,det(U*V')])*V';
  end
end
```

Listing 2: Translation synchronization

```
function T = translation_synch(U,B)

    B(1,:) = []; % remove node 1
    F=kron(B',eye(3));
    X=F\U(:);
    X=[0;0;0;X]; % add node 1
    X=reshape(X,3,[]);

    T = num2cell(reshape(X,3,[]),...
        [1,size(X,2)]);
end
```

9.2 Spectral solution versus null-space solution

In the presence of noise the null-space and spectral solutions do not coincide, in general. In the case of a complete graph, for instance, they coincide if and only if the unique non zero eigenvalue of Z is exactly n, which is unlikely to happen in practice.

An empirical comparison between the two approaches – for the $\Sigma = SE(3)$ case – is reported in [11] where it is shown that the spectral solution achieves the same accuracy as the null-space method but it is faster. In particular, it turns out that the final rounding step, i.e. zeroing the imaginary part of the eigenvectors, do not compromise the accuracy achieved by the spectral method.

Note that in both cases the matrices inherit the same sparsity pattern as the adjacency matrix A, thus sparse solvers can be exploited, e.g., `eigs` for the spec-

tral method and `svds` for the null-space solution, as done in Listing 1. As a matter of fact, `svds(F)` calls `eigs([0 F; F' 0])`, as reported in the function documentation, and consequently it runs more slowly, for the dimension of the matrix is double. From the computational complexity point of view, every iteration of the Lanczos method (implemented by `eigs`) is linear in n [51], if the matrix is sparse, but the number of iterations cannot be bounded by a constant.

9.3 Efficiency

The considered instances of synchronization translate into efficient closed form solutions, such as spectral decomposition or linear least squares. As a matter of fact, in the experiments reported in the literature [3,18,11] they are consistently the fastest method. For instance, the rigid-motion synchronization pipeline in [6] takes 7 seconds for the Madrid Metropolis dataset [118], whose epipolar graph contains about 300 nodes and 65% of missing edges (with respect to the complete graph).

Note also that the spectral/null-space solutions aim at recovering the vertex labelling X, thus they require to store a $dn \times d$ matrix in addition to the $dn \times dn$ matrix Z_A containing the input measures, which is *sparse* if the graph is not complete. Semidefinite and rank relaxations, instead, aim at computing a refined edge label for each pair of nodes, represented by the optimization variable $\widetilde{Z}$ (see Sections 7.2 and 7.3), thus they require an additional amount of memory to store a *full* $dn \times dn$ matrix.

9.4 Accuracy

The least-squares solution to translation synchronization is statistically optimal, whereas the spectral (or, equivalently, null-space) solution to the other instances of synchronization provides an extrinsic estimate, whose quality is – in general – inferior to those provided by intrinsic methods. Note also that the spectral technique is a weaker relaxation than semidefinite programming (namely it enforces less constraints on the optimization variable), as explained in Section 7, and it does not guarantee exact recovery (in contrast to [116,89,32]).

In [11] the spectral method is compared to [90], which minimizes a geometric error tightly related to the synchronization cost function with a convex relaxation. The latter returns a more accurate solution, but it requires a significant amount of time (for instance, it takes around ten minutes on a synthetic scenario with $n = 100$ and 30% of missing edges, whereas the spectral approach is able to compute a solution in less than 2 seconds with up to $n = 1000$).

Experiments on multiple point-set registration conducted in [9] – graphs with up to 95% of missing edges – demonstrate that the spectral solution and the semidefinite relaxation proposed in [89] are comparable in accuracy but the latter requires additional time. However, the same comparison[4] on the SLAM dataset used in [89] – with more than 99% of missing edges – reveals that [89] achieves a running time comparable with the spectral method on these very sparse graphs. The authors of [88] suggest that some computational enhancements can be incorporated in semidefinite relaxations in order to speed up the computation further. Finally, the same study performed in [9] indicates also that methods based on matrix completion [7], instead, fail when the percentage of missing data is higher than 90%.

Although the accuracy obtained by the spectral method is not optimal, it is however high, as demonstrated by experiments performed in [11] in a variety of scenarios (for instance, it gets an average rotation error of the order of 1 degree on the popular Stanford 3D Scanning Repository [103]). As a consequence, it can be seen as a good and fast initialization for a subsequent local refinement (e.g. bundle adjustment in structure from motion).

9.5 Robustness

The least-squares solution for translation synchronization of Section 3 can be made resistant to outliers (i.e. rogue edge labels) by replacing $\rho(y) = y^2$ in Equation (14) with another function $\rho(y)$ with sub-quadratic growth, and solving the resulting minimization problem, e.g., with Iteratively Reweighted Least Squares (IRLS) [60]. This technique iteratively solves weighted least squares problems where the weights are computed at each iteration as a function of the residuals of the current solution.

As for synchronization over $GL(d)$, it is easy to see that the analysis of Section 4 can be extended to handle weighted measurements, which translates in relaxing the entries of the adjacency matrix A to assume real values in $[0, 1]$, where 0 still indicates a missing measurement and the other values reflect the reliability of the edge labels. This allows to apply an IRLS-like scheme: first, an estimate for the vertex labelling with given edge weights is obtained via either the spectral approach or the null-space solution; then, the weights are updated based on the current estimate of the vertex

[4] Unpublished experiments made by the authors

labels, and these steps are iterated until convergence or a maximum number of iterations is reached.

Experiments in [11] report good performances in the $\Sigma = SE(d)$ case (resilience to about 40% of wrong edge labels) and it is shown in [9] that the spectral relaxation is comparable to methods performing robust matrix completion in terms of robustness to outliers.

9.6 Limitations and open questions

In summary, the spectral and null-space methods provide fast, sub-optimal (but fairly accurate) solutions in the presence of noise/outliers.

Further efficiency is needed in certain applications involving very large-scale problems, such as multi-view matching. For instance, the method in [76], which provides a practical approach to partial permutation synchronization based on a spectral decomposition, took one and half hour to compute a solution in the Dino dataset [96], where $d = 493$ and $n = 363$ (resulting in a matrix of size ≈ 180000). A possible way to speed-up the computation is to split the synchronization problem into smaller subproblems in a distributed fashion, similarly to [41,21].

Thanks to the group-labelling interpretation, the formalism of synchronization permits to address a variety of Computer Vision applications using the same Linear Algebra approach without relying on features or points, since the problem is formulated in frame space, or, more abstractly, in a group/semigroup. An unexplored application involves projective frames, which are represented by 4×4 invertible matrices defined up to scale. As mentioned in Section 8.1, the idea of scaling each matrix by the d-th root of its determinant, which was used in [15], does not apply here because complex roots may appear, hence further research is required in order to formulate the problem in terms of synchronization.

From the theoretical perspective, interesting open questions include the link between the spectral solution and the consistency error when the data matrix is not symmetric, and the relation between the notion of null-cycle and the synchronization problem over an inverse monoid.

10 Conclusion

In this paper we gathered several disparate works within the common framework of synchronization and showed how this framework can be profitably used in several Computer Vision applications. Besides exhibiting a nice and clean formulation, synchronization can also benefit from efficient and closed-form solutions such as spectral decomposition or linear least squares. We hope that this survey will serve as a starting point for more research in this field.

Acknowledgements

The authors would like to thank Carlo Comin and Romeo Rizzi for pointing out the connection with the Group Feedback Edge Set problem, and Eleonora Maset who co-authored the works on permutation synchronization. We are also grateful to Fabio Crosilla for drawing our attention to the leveling problem in topography.

This work was supported by the European Regional Development Fund under the project IMPACT (reg. no CZ.02.1.01/0.0/0.0/15_003/0000468).

A Kronecker, Hadamard and Khatri-Rao products

This appendix is devoted to the Kronecker, Hadamard and Khatri-Rao products [113,78,73], which are widely used in this paper.

Let A and B be two real matrices of dimension $m \times r$ and $n \times s$ respectively. The *Kronecker product* of A and B [113], denoted by $A \otimes B$, is defined as

$$A \otimes B = \begin{bmatrix} [A]_{1,1}B & [A]_{1,2}B & \dots & [A]_{1,r}B \\ [A]_{2,1}B & [A]_{2,2}B & \dots & [A]_{2,r}B \\ \dots & & & \dots \\ [A]_{m,1}B & [A]_{m,2}B & \dots & [A]_{m,r}B \end{bmatrix} \tag{83}$$

where each $[A]_{i,j}B$ is a block of dimension $n \times s$, thus $A \otimes B$ has dimension $mn \times rs$. The Kronecker product is associative, distributive (with respect to the sum of matrices), but not commutative, and it satisfies the following properties

$$(A \otimes B)^{\mathsf{T}} = A^{\mathsf{T}} \otimes B^{\mathsf{T}} \tag{84}$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \tag{85}$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{86}$$

$$\mathrm{vec}(AXB) = (B^{\mathsf{T}} \otimes A)\mathrm{vec}(X) \tag{87}$$

where $\mathrm{vec}(\cdot)$ denotes the vectorization operator which transforms a matrix into a vector by stacking the columns of the matrix one underneath the other.

Let $A = U_A \Sigma_A V_A^{\mathsf{T}}$ and $B = U_B \Sigma_B V_B^{\mathsf{T}}$ be the singular value decompositions of A and B, respectively, then

$$A \otimes B = (U_A \otimes U_B)(\Sigma_A \otimes \Sigma_B)(V_A \otimes V_B)^{\mathsf{T}} \tag{88}$$

which implies

$$\mathrm{rank}(A \otimes B) = \mathrm{rank}(A)\ \mathrm{rank}(B). \tag{89}$$

Thus the Kronecker product of two matrices is invertible if and only if both the factors are invertible.

Consider now two real matrices A and B of dimension $m \times r$ and $n \times r$ respectively, and denote the columns of A by $\mathbf{a}_1, \dots, \mathbf{a}_r$ and those of B by $\mathbf{b}_1, \dots, \mathbf{b}_r$. The *Khatri-Rao product* of A and B [69,73], denoted by $A \odot B$, is defined as

$$A \odot B = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_r \otimes \mathbf{b}_r \end{bmatrix} \tag{90}$$

where each $\mathbf{a}_i \otimes \mathbf{b}_i$ is a vector of dimension mn, thus $A \odot B$ has dimension $mn \times r$. The Khatri-Rao product is associative, distributive, but not commutative, and it satisfies the following equalities

$$(A \otimes B)(C \odot D) = (AC) \odot (BD) \tag{91}$$

$$\operatorname{vec}(A \operatorname{diag}(\mathbf{x}) B) = (B^\mathsf{T} \odot A)\mathbf{x} \tag{92}$$

where $\operatorname{diag}(\mathbf{x})$ transforms the vector $\mathbf{x} = [x_1 \ \dots \ x_r]^\mathsf{T}$ into a diagonal matrix with elements $x_1, \dots, x_r$ along the diagonal.

To the best of our knowledge, equalities expressing the rank of $A \odot B$ in terms of the rank of the factors are not present in the literature, in contrast to the case of the Kronecker product. Some inequalities are reported in [69], where it is shown, for instance, that $\operatorname{rank}(A \odot B) \geq \operatorname{rank}(A)$, if all the columns of B corresponding to independent columns of A are non-null.

Let A and B be two real matrices of dimension $m \times r$. The *Hadamard product* (or *entry-wise product*) of A and B [78], denoted by $A \circ B$, has dimension $m \times r$ as well, and it is simply the product of the corresponding elements

$$A \circ B = \begin{bmatrix} [A]_{1,1}[B]_{1,1} & [A]_{1,2}[B]_{1,2} & \dots & [A]_{1,r}[B]_{1,r} \\ [A]_{2,1}[B]_{2,1} & [A]_{2,2}[B]_{2,2} & \dots & [A]_{2,r}[B]_{2,r} \\ \dots & & & \dots \\ [A]_{m,1}[B]_{m,1} & [A]_{m,2}[B]_{m,2} & \dots & [A]_{m,r}[B]_{m,r} \end{bmatrix}. \tag{93}$$

The Hadamard product is associative, distributive, commutative, and it satisfies the following properties

$$(A \odot B) \circ (C \odot D) = (A \circ C) \odot (B \circ D) \tag{94}$$

$$(A \odot B)^\mathsf{T}(A \odot B) = (AA^\mathsf{T}) \circ (BB^\mathsf{T}) \tag{95}$$

$$\operatorname{vec}(A \circ B) = \operatorname{diag}(\operatorname{vec}(A)) \operatorname{vec}(B) \tag{96}$$

$$\operatorname{diag}(\mathbf{x}) \ A \ \operatorname{diag}(\mathbf{y}) = A \circ (\mathbf{x}\mathbf{y}^\mathsf{T}) \tag{97}$$

$$\operatorname{rank}(A \circ B) \leq \operatorname{rank}(A) \operatorname{rank}(B). \tag{98}$$

B Results from Graph Theory

In this section we review some useful concepts from graph theory. A complete treatment of this subject can be found in [23, 64].

A *graph* is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a finite set and $\mathcal{E}$ is a family of pairs of elements of $\mathcal{V}$. We use n and m to denote the number of vertices and edges respectively, namely $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$. A *weighted graph* is a graph together with a weight function $\omega : \mathcal{E} \to \mathbb{R}^+$. If the graph is unweighted, we set $\omega : \mathcal{E} \to 1$ and call w the *uniform weight function*. An edge occurring more than once is referred to as a *multiple edge*, and a graph without multiple edges is called *simple*. An edge of the form (v, v) is called a *loop*. In an undirected graph, the *degree* of a vertex v is the number of times that v occurs as an endpoint of an edge. In a directed graph, the *outdegree* and *indegree* of a vertex v are the number of times that v occurs as the tail and head of an edge, respectively.

A *subgraph* $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of $\mathcal{G}$ is a graph with $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. If $\mathcal{E}'$ is a subset of $\mathcal{E}$, then $\mathcal{G} \setminus \mathcal{E}'$ denotes the graph obtained by removing all the edges in $\mathcal{E}'$ from $\mathcal{G}$. If $\mathcal{V}'$ is a subset of $\mathcal{V}$, then $\mathcal{G} \setminus \mathcal{V}'$ denotes the graph obtained by removing all the vertices in $\mathcal{V}'$ and their incident edges from $\mathcal{G}$. A *path* from v to w is a subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ with $\mathcal{V}' = \{v_0 = v, v_1, \dots, v_k = w\}$ and $\mathcal{E}' = \{(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)\}$. An undirected graph is called *connected* if there exists a path from each vertex to any other, and a directed graph is called connected if the underlying undirected graph is connected. Any maximal connected subgraph $\mathcal{H}$ is called a *connected component*. A graph is a *tree* if it is connected and it has $n-1$ edges. The disjoint union of trees is called a *forest*. The number of edges in a forest is $n - cc$, where cc denotes the number of connected components in $\mathcal{G}$. A subgraph $\mathcal{G}'$ of a connected graph $\mathcal{G}$ is called a *spanning tree* if it has the same vertices of $\mathcal{G}$ and it is a tree. If $\mathcal{G}$ is not connected, any union of spanning trees for each connected component is called a *spanning forest*.

A *cycle* in a graph $\mathcal{G}$ is a vector $\mathbf{c} \in \mathbb{Q}^m$ such that for any vertex $v \in \mathcal{V}$ it holds

$$\sum_{e \in \delta_+(v)} [\mathbf{c}]_e = \sum_{e \in \delta_-(v)} [\mathbf{c}]_e \tag{99}$$

where $\delta_+(v)$ and $\delta_-(v)$ denote the edges leaving and entering v, respectively, and $[\mathbf{c}]_e$ denotes the component of $\mathbf{c}$ indexed by edge e. A cycle is *simple* if $[\mathbf{c}]_e \in \{-1, 0, 1\}$ for all $e \in \mathcal{E}$, and a simple cycle is a *circuit* if its support (i.e. the set of edges with $[\mathbf{c}]_e \neq 0$) is connected and for any vertex $v \in \mathcal{V}$ there are at most two edges in the support incident to v. The set of cycles forms a vector space over $\mathbb{Q}$, which is called the *cycle space* of $\mathcal{G}$, and a *cycle basis* is a set of circuits forming a basis of such a space. It can be shown [23, 64] that if $\mathcal{G}$ is connected the dimension of the cycle space is given by the *cyclomatic number*

$$\nu = m - n + 1. \tag{100}$$

B.1 Matrices associated with graphs

The *adjacency matrix* A of a graph $\mathcal{G}$ is the $n \times n$ matrix whose elements indicate whether pairs of vertices are adjacent or not, namely

$$[A]_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \tag{101}$$

If $\mathcal{G}$ does not contain loops, then A has zero diagonal. Note that the adjacency matrix is symmetric if the graph is undirected.

The *incidence matrix* B of a directed graph $\mathcal{G}$ is the $n \times m$ matrix defined by

$$[B]_{k,e} = \begin{cases} -1 & \text{if } k \text{ is the head of edge } e, \\ 1 & \text{if } k \text{ is the tail of edge } e, \\ 0 & \text{otherwise.} \end{cases} \tag{102}$$

The rows of B correspond to vertices and the columns correspond to edges. Note that each column has exactly two non zero entries, which correspond to the endpoints of the edge associated to that column. The *incidence matrix* B of an undirected graph $\mathcal{G}$ is defined considering a particular orientation of the edges. It is shown in [23] that, if $\mathcal{G}$ is connected, then

$$\operatorname{rank}(B) = n - 1. \tag{103}$$

The *degree matrix* D of an undirected graph $\mathcal{G}$ is the $n \times n$ diagonal matrix such that $[D]_{i,i}$ contains the degree of node i. Equivalently, it can be defined as

$$D = \operatorname{diag}(A\mathbf{1}_{n \times 1}) \tag{104}$$

where $\mathbf{1}_{n\times 1}$ denotes a $n\times 1$ matrix filled by ones, thus $A\mathbf{1}_{n\times 1}$ is the sum of the rows of A. In the case of a directed graph, either the indegree or the outdegree can be used. The *transition matrix* P is defined as

$$P = D^{-1}A. \tag{105}$$

The *Laplacian matrix* L of a graph $\mathcal{G}$ is defined as

$$L = D - A. \tag{106}$$

It can be checked that, independently of the orientation of the edges, the following equation holds

$$L = BB^{\mathsf{T}} \tag{107}$$

which implies that L is symmetric and positive semidefinite, and, if the graph is connected, $\operatorname{rank}(L) = \operatorname{rank}(B) = n - 1$, hence L is singular. Note that the vector $\mathbf{1}_{n\times 1}$ is in the nullspace of L.

The notion of adjacency matrix can be extended to the case of a weighted graph, which translates in letting the entries of A to assume values in $[0, 1]$. Specifically, $[A]_{i,j}$ contains the weight of edge (i, j), and $[A]_{i,j} = 0$ still indicates that $(i, j) \notin \mathcal{E}$. In this case Equations (104), (105) and (106) still make sense, which define the degree matrix, the transition matrix and the Laplacian matrix of a weighted graph, respectively.

The *cycle matrix* C corresponding to a cycle basis of a connected graph $\mathcal{G}$ is the $(m - n + 1) \times m$ matrix having the incidence vectors of the circuits in the basis in its rows. The following equation [23] expresses the relation between the cycle matrix and the incidence matrix

$$CB^{\mathsf{T}} = 0. \tag{108}$$

Note that, if the graph is undirected, the matrices A and L are symmetric, thus their eigenvalues are real. The matrix P is not symmetric, but it is similar to the symmetric matrix N defined as

$$N = D^{-1/2}AD^{-1/2} \tag{109}$$

since $P = D^{-1/2}ND^{1/2}$. The matrices N and P have the same eigenvalues, so P has real eigenvalues.

Theorem 1 (Perron-Frobenius [77]) *If an $n \times n$ matrix has non-negative entries then it has a non-negative real eigenvalue λ which has maximum absolute value among all the eigenvalues. This eigenvalue has a non-negative real eigenvector. If, in addition, the matrix has no block-triangular decomposition, then λ has multiplicity* 1 *and the corresponding eigenvector is positive.*

As explained in [74], the Perron-Frobenius theorem implies that, if $\mathcal{G}$ is connected, the largest eigenvalue of A has multiplicity 1. Likewise, the largest eigenvalue of the transition matrix is 1 and it ha multiplicity 1. It is easy to check that the eigenvector associated to such eigenvalue is $\mathbf{1}_{n\times 1}$.

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. Aragues, R., Carlone, L., Sagues, C., Calafiore, G.: Distributed centroid estimation from noisy relative measurements. Systems and Control Letters **61**(7), 773 – 779 (2012)
3. 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)
4. Arrigoni, F., Fusiello, A.: Bearing-based network localizability: A unifying view. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2018)
5. Arrigoni, F., Fusiello, A., Rossi, B.: On computing the translations norm in the epipolar graph. In: Proceedings of the International Conference on 3D Vision (3DV), pp. 300–308 (2015)
6. Arrigoni, F., Fusiello, A., Rossi, B.: Camera motion from group synchronization. In: Proceedings of the International Conference on 3D Vision (3DV), pp. 546–555 (2016)
7. Arrigoni, F., Magri, L., Rossi, B., Fragneto, P., Fusiello, A.: Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In: Proceedings of the International Conference on 3D Vision (3DV), pp. 491–498 (2014)
8. Arrigoni, F., Maset, E., Fusiello, A.: Synchronization in the symmetric inverse semigroup. In: International Conference on Image Analysis and Processing, pp. 70–81. Springer (2017)
9. 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)
10. Arrigoni, F., Rossi, B., Fusiello, A.: Global registration of 3D point sets via LRS decomposition. In: Proceedings of the 14th European Conference on Computer Vision, pp. 489–504 (2016)
11. Arrigoni, F., Rossi, B., Fusiello, A.: Spectral synchronization of multiple views in SE(3). SIAM Journal on Imaging Sciences **9**(4), 1963 – 1990 (2016)
12. Arrigoni, F., Rossi, B., Malapelle, F., Fragneto, P., Fusiello, A.: Robust global motion estimation with matrix completion. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **XL-5**, 63–70 (2014)
13. Barooah, P., Hespanha, J.P.: Estimation on graphs from relative measurements. IEEE Control Systems **27**(4), 57 – 74 (2007)
14. Barooah, P., Hespanha, J.P.: Estimation from relative measurements: Electrical analogy and large graphs. IEEE Transactions on Signal Processing **56**(6), 2181 – 2193 (2008)
15. Bartoli, A., Sturm, P.: Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. International Journal of Computer Vision **52**(1), 45–64 (2003)
16. Belta, C., Kumar, V.: Euclidean metrics for motion generation on SE(3). Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science **216**(1), 47–60 (2002)
17. Benjemaa, R., Schmitt, F.: A solution for the registration of multiple 3D point sets using unit quaternions. In: Proceedings of the European Conference on Computer Vision, pp. 34–50 (1998)
18. 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)
19. Bernard, F., Thunberg, J., Goncalves, J., Theobalt, C.: Synchronisation of Partial Multi-Matchings via Non-negative Factorisations. arXiv e-prints **1803.06320** (2018)
20. Besl, P., McKay, N.: A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(2), 239–256 (1992)
21. Bhowmick, B., Patra, S., Chatterjee, A., Govindu, V.M., Banerjee, S.: Divide and conquer: Efficient large-scale structure from motion using graph partitioning. In: 12th Asian Conference on Computer Vision (ACCV 2014) (2014)
22. Bjerhammar, A.: Theory of errors and generalized matrix inverses. Elsevier (1973)
23. Bollobas, B.: Modern Graph Theory. Springer (1998)
24. Boumal, N., Singer, A., Absil, P.A.: Robust estimation of rotations from relative measurements by maximum likelihood. In: Proceedings of the IEEE International Conference on Robotics and Automation (2013)
25. 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)
26. Bourmaud, G., Megret, R., Giremus, A., Berthoumieu, Y.: Global motion estimation from relative measurements in the presence of outliers. In: Proceedings of the Asian Conference on Computer Vision (2014)
27. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning **3**(1), 1–122 (2011)
28. Brand, M., Antone, M., Teller, S.: Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In: Proceedings of the European Conference on Computer Vision (2004)
29. Candès, E.J., Tao, T.: The power of convex relaxation: near-optimal matrix completion. IEEE Transactions on Information Theory **56**(5), 2053–2080 (2010)
30. Cardoso, J., Leite, F.S.: On computing the logarithm in the special Euclidean group of motions in $\mathbb{R}^n$, (1999). Preprint, Departamento de Matematica, 99-01, Universidade de Coimbra
31. Carlone, L., Aragues, R., Castellanos, J., Bona, B.: A linear approximation for graph-based simultaneous localization and mapping. In: Robotics: Science and Systems (RSS), pp. 41–48 (2011)
32. Carlone, L., Calafiore, G.C., Tommolillo, C., Dellaert, F.: Planar pose graph optimization: Duality, optimal solutions, and verification. IEEE Transactions on Robotics **32**(3), 545–565 (2016)
33. Carlone, L., Censi, A.: From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization. IEEE Transactions on Robotics **30**(2), 475–492 (2014)
34. 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)
35. Castellani, U., Fusiello, A., Murino, V.: Registration of multiple acoustic range views for underwater scene reconstruction. Computer Vision and Image Understanding **87**(1), 78–89 (2002)
36. Chatterjee, A., Govindu, V.M.: Efficient and robust large-scale rotation averaging. In: Proceedings of the International Conference on Computer Vision (2013)
37. 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)
38. Chudnovsky, M., Cunningham, W.H., Geelen, J.: An algorithm for packing non-zero a-paths in group-labelled graphs. Combinatorica **28**(2), 145–161 (2008)
39. 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)
40. Cucuringu, M.: Synchronization over Z_2 and community detection in signed multiplex networks with constraints. Journal of Complex Networks **3**(3), 469–506 (2015)
41. Cucuringu, M., Lipman, Y., Singer, A.: Sensor network localization by eigenvector synchronization over the Euclidean group. ACM Transactions on Sensor Networks **8**(3), 19:1 – 19:42 (2012)
42. Cygan, M., Pilipczuk, M., Pilipczuk, M.: On group feedback vertex set parameterized by the size of the cutset. In: Graph-Theoretic Concepts in Computer Science - 38th International Workshop, pp. 194–205 (2012)
43. Edelman, P.H., Saks, M.: Group labelings of graphs. Journal of Graph Theory **3**(2), 135–140 (1979)
44. Enqvist, O., Kahl, F., Olsson, C.: Non-sequential structure from motion. In: Eleventh Workshop on Omnidirectional Vision, Camera Networks and Non-classical Camera (2011)
45. Fantoni, S., Castellani, U., Fusiello, A.: Accurate and automatic alignment of range surfaces. In: Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission, pp. 73 – 80 (2012)
46. Fraser, C.: Network orientation models for image-based 3D measurement. ISPRS Archives **XXXVI-5/W17** (2005)
47. Fredriksson, J., Olsson, C.: Simultaneous multiple rotation averaging using lagrangian duality. In: Proceedings of the Asian Conference on Computer Vision (2012)
48. Fusiello, A., Castellani, U., Ronchetti, L., Murino, V.: Model acquisition by registration of multiple acoustic range views. In: Proceedings of the European Conference on Computer Vision, pp. 805–819 (2002)
49. Giridhar, A., Kumar, P.: Distributed clock synchronization over wireless networks: Algorithms and analysis. Proceedings of the IEEE Conference on Decision and Control pp. 4915–4920 (2006)
50. Goldstein, T., Hand, P., Lee, C., Voroninski, V., Soatto, S.: ShapeFit and ShapeKick for robust, scalable structure from motion. In: Proceedings of the European Conference on Computer Vision, pp. 289 – 304 (2016)
51. Golub, G.H., Van Loan, C.F.: Matrix computations (3rd ed.). Johns Hopkins University Press (1996)
52. Govindu, V.M.: Combining two-view constraints for motion estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2001)
53. 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)
54. Govindu, V.M.: Robustness in motion averaging. In: Proceedings of the Asian Conference on Computer Vision, pp. 457–466 (2006)

55. Govindu, V.M.: Motion averaging: a framework for efficient and accurate large-scale camera estimation in 3D vision. Tutorial at CVPR (2017). http://www.ee.iisc.ac.in/labs/cvl/cvpr2017/tutorial/
56. Govindu, V.M., Pooja, A.: On averaging multiview relations for 3D scan registration. IEEE Transactions on Image Processing **23**(3), 1289–1302 (2014)
57. Guillemot, S.: FTP algorithms for path-traversal and cycle-traversal problems. Discrete Optimization **8**(1), 61 – 71 (2011). Parameterized Complexity of Discrete Optimization
58. Hartley, R., Trumpf, J., Dai, Y., Li, H.: Rotation averaging. International Journal of Computer Vision (2013)
59. He, J., Balzano, L., Szlam, A.: Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1568 – 1575 (2012)
60. Holland, P.W., Welsch, R.E.: Robust regression using iteratively reweighted least-squares. Communications in Statistics - Theory and Methods **6**(9), 813–827 (1977)
61. Jiang, N., Cui, Z., Tan, P.: A global linear method for camera pose registration. In: Proceedings of the International Conference on Computer Vision (2013)
62. Joglekar, M., Shah, N., Diwan, A.A.: Balanced group-labeled graphs. Discrete Mathematics **312**(9), 1542 – 1549 (2012). Recent Trends in Graph Theory and Combinatorics
63. Karp, R., Elson, J., Estrin, D., Shenker, S.: Optimal and global time synchronization in sensor nets. Tech. rep., Center for Embedded Networked Sensing: University of California, Los Angeles (2003)
64. Kavitha, T., Liebchen, C., Mehlhorn, K., Michail, D., Rizzi, R., Ueckerdt, T., Zweig, K.: Cycle bases in graphs: Characterization, algorithms, complexity, and applications. Computer Scienze Review **3**(4), 199–243 (2009)
65. Kawarabayashia, K., Wollan, P.: Non-zero disjoint cycles in highly connected group labelled graphs. Journal of Combinatorial Theory, Series B **96**(2), 296 – 301 (2006)
66. Keller, J.: Closest unitary, orthogonal and Hermitian operators to a given operator. Mathematics Magazine **48**, 192–197 (1975)
67. Kennedy, R., Daniilidis, K., Naroditsky, O., Taylor, C.J.: Identifying maximal rigid components in bearing-based localization. In: Proceedings of the International Conference on Intelligent Robots and Systems, pp. 194 – 201 (2012)
68. Keshavan, R.H., Montanari, A., Oh, S.: Matrix completion from a few entries. IEEE Transactions on Information Theory **56**(6), 2980–2998 (2010)
69. Khatri, C.G., Rao, C.R.: Solutions to some functional equations and their applications to characterization of probability distributions. Sankhya: The Indian Journal of Statistics, Series A (1961-2002) **30**(2), pp. 167–180 (1968)
70. Krishnan, S., Lee, P.Y., Moore, J.B., Venkatasubramanian, S.: Optimisation-on-a-manifold for global registration of multiple 3D point sets. International Journal of Intelligent Systems Technologies and Applications **3**(3/4), 319–340 (2007)
71. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2 **2**, 83 – 97 (1955)
72. Levi, N., Werman, M.: The viewing graph. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 518 – 522 (2003)
73. Liu, S., Trenkler, G.: Hadamard, Khatri-Rao, Kronecker and other matrix products. International Journal of Information and Systems Sciences **4**(1), 160 – 177 (2008)
74. Lovász, L.: Eigenvalues of graphs. Tech. rep., E otv os Loránd University, Institute of Mathematics (2007)
75. 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)
76. 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)
77. Meyer, C.D.: Matrix Analysis and applied linear algebra. SIAM (2000)
78. Minka, T.: Old and new matrix algebra useful for statistics. MIT Media Lab note (2000). Http://research.microsoft.com/ minka/papers/matrix/
79. Moakher, M.: Means and averaging in the group of rotations. SIAM Journal on Matrix Analysis and Applications **4**(1), 1–16 (2002)
80. Molavi, P., Jadbabaie, A.: A topological view of estimation from noisy relative measurements. In: IEEE American Control Conference (2011)
81. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motions for robust, accurate and scalable structure from motion. In: Proceedings of the International Conference on Computer Vision, pp. 3248–3255 (2013)
82. 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)
83. Ozyesil, O., Singer, A.: Robust camera location estimation by convex programming. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2674 – 2683 (2015)
84. Ozyesil, O., Voroninski, V., Basri, R., Singer, A.: A survey of structure from motion. Acta Numerica **26**, 305 – 364 (2017)
85. Pachauri, D., Kondor, R., Singh, V.: Solving the multiway matching problem by permutation synchronization. In: Advances in Neural Information Processing Systems 26, pp. 1860–1868. Curran Associates, Inc. (2013)
86. Pennec, X.: Multiple registration and mean rigid shape: Applications to the 3D case. In: 16th Leeds Annual Statistical Workshop, pp. 178–185 (1996)
87. Pulli, K.: Multiview registration for large data sets. In: Proceedings of the International Conference on 3-D Digital Imaging and Modeling, pp. 160–168 (1999)
88. Rosen, D.M., Carlone, L.: Computational enhancements for certifiably correct SLAM. In: Proceedings of the International Conference on Intelligent Robots and Systems (2017)
89. 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**(2-3), 95–125 (2019)
90. 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)
91. Russel, W., Klein, D., Hespanha, J.: Optimal estimation on the graph cycle space. IEEE Transactions on Signal Processing **59**(6), 2834 – 2846 (2011)

92. 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)
93. Saunderson, J., Parrilo, P.A., Willsky, A.S.: Semidefinite descriptions of the convex hull of rotation matrices. SIAM Journal on Optimization **25**(3), 1314 – 1343 (2015)
94. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4104 – 4113 (2016)
95. Schroeder, P., Bartoli, A., Georgel, P., Navab, N.: Closed-form solutions to multiple-view homography estimation. In: Applications of Computer Vision (WACV), 2011 IEEE Workshop on, pp. 650–657 (2011)
96. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 519–528 (2006)
97. Sharp, G.C., Lee, S.W., Wehe, D.K.: Multiview registration of 3D scenes by minimizing error between coordinate frames. In: Proceedings of the European Conference on Computer Vision, pp. 587–597 (2002)
98. Shen, Y., Huang, Q., Srebro, N., Sanghavi, S.: Normalized spectral map synchronization. In: D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, R. Garnett (eds.) Advances in Neural Information Processing Systems 29, pp. 4925–4933. Curran Associates, Inc. (2016)
99. Singer, A.: Angular synchronization by eigenvectors and semidefinite programming. Applied and Computational Harmonic Analysis **30**(1), 20 – 36 (2011)
100. Singer, A., Shkolnisky, Y.: Three-dimensional structure determination from common lines in cryo-EM by eigenvectors and semidefinite programming. SIAM Journal on Imaging Sciences **4**(2), 543 – 572 (2011)
101. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques, pp. 835–846 (2006)
102. Solis, R., Borkar, V.S., Kumar, P.R.: A new distributed time synchronization protocol for multihop wireless networks. In: Proceedings of the 45th IEEE Conference on Decision and Control, pp. 2734–2739 (2006)
103. `http://graphics.stanford.edu/data/3Dscanrep/`
104. Thunberg, J., Montijano, E., Hu, X.: Distributed attitude synchronization control. In: 2011 50th IEEE Conference on Decision and Control and European Control Conference, pp. 1962–1967 (2011). DOI 10.1109/CDC.2011.6161295
105. Toldo, R., Beinat, A., Crosilla, F.: Global registration of multiple point clouds embedding the generalized procrustes analysis into an ICP framework. In: International Symposium on 3D Data Processing, Visualization and Transmission, pp. 109–122 (2010)
106. Toldo, R., Gherardi, R., Farenzena, M., Fusiello, A.: Hierarchical structure-and-motion recovery from uncalibrated images. Computer Vision and Image Understanding (2015)
107. 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)
108. Tron, R., Carlone, L., Dellaert, F., Daniilidis, K.: Rigid components identification and rigidity enforcement in bearing-only localization using the graph cycle basis. In: IEEE American Control Conference (2015)
109. Tron, R., Danilidis, K.: Statistical pose averaging with varying and non-isotropic covariances. In: Proceedings of the European Conference on Computer Vision (2014)
110. Tron, R., Vidal, R.: Distributed 3-D localization of camera sensor networks from 2-D image measurements. IEEE Transactions on Automatic Control **59**(12), 3325–3340 (2014)
111. Tron, R., Zhou, X., Daniilidis, K.: A survey on rotation optimization in structure from motion. In: Computer Vision and Pattern Recognition Workshops (CVPRW) (2016)
112. Tron, R., Zhou, X., Esteves, C., Daniilidis, K.: Fast multi-image matching via density-based clustering. In: Proceedings of the International Conference on Computer Vision, pp. 4077–4086 (2017)
113. Van Loan, C.: The ubiquitous Kronecker product. J. Comput. Appl. Math **123**(1-2), 85–100 (2000)
114. Varadarajan, V.S.: Lie Groups, Lie Algebras, and Their Representations, *Graduate Texts in Mathematics*, vol. 102. Springer (1984)
115. Wahlström, M.: Half-integrality, LP-branching and FPT algorithms. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1762–1781 (2014)
116. Wang, L., Singer, A.: Exact and stable recovery of rotations for robust synchronization. Information and Inference: a Journal of the IMA **2**(2), 145–193 (2013)
117. Wilson, K., Bindel, D., Snavely, N.: When is rotations averaging hard? In: Proceedings of the European Conference on Computer Vision, pp. 255 – 270 (2016)
118. Wilson, K., Snavely, N.: Robust global translations with 1DSfM. In: Proceedings of the European Conference on Computer Vision, pp. 61–75 (2014)
119. Wright, S.: Primal-Dual Interior-Point Methods. Society for Industrial and Applied Mathematics (1997)
120. 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)
121. Zach, C., Klopschitz, M., Pollefeys, M.: Disambiguating visual relations using loop constraints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1426 – 1433 (2010)
122. Zhao, S., Zelazo, D.: Localizability and distributed protocols for bearing-based network localization in arbitrary dimensions. Automatica **69**, 334 – 341 (2016)
123. Zheng, Y., Liu, G., Sugimoto, S., Yan, S., Okutomi, M.: Practical low-rank matrix approximation under robust L_1-norm. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1410–1417 (2012)
124. 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)