Camera Motion from Group Synchronization

Federica Arrigoni, Andrea Fusiello
DPIA - University of Udine
Via Delle Scienze, 208 - Udine (Italy)
arrigoni.federica@spes.uniud.it

Beatrice Rossi
AST Lab - STMicroelectronics
Via Olivetti, 2 - Agrate Brianza (Italy)
beatrice.rossi@st.com

Abstract

This paper deals with the problem of estimating camera motion in the context of structure-from-motion. We describe a pipeline that consumes relative orientations and produces absolute orientations (i.e. camera position and attitude in an absolute reference frame). This pipeline exploits the concept of "group synchronization" in most of its stages, all of which entail direct solutions such as eigenvalue decompositions or linear least squares. A comprehensive introduction to the group synchronization problem is provided, and the proposed pipeline is evaluated on standard real datasets.

1. Introduction

Structure from motion (SfM) consists in recovering the 3D *structure* of the scene and camera *motion* or *orientation*[1] from point correspondences. The most common approach consists in incrementally growing a partial reconstruction by adding cameras and points iteratively (resection/intersection) [46]. A different paradigm in which *first* the motion is recovered and *then* the structure is computed is receiving a growing attention in Computer Vision community. All these methods start from the relative orientation of a subset of camera pairs (or triplets), computed from point correspondences, then solve for the absolute orientation of all the cameras (*globally*), reconstruct 3D points by intersection, and finally run bundle adjustment to refine the reconstruction. The step from relative to absolute orientation is the core of these methods. The problem can be usefully modelled by considering the *epipolar graph*, where nodes correspond to cameras, labelled with their absolute orientations, and edges correspond to relative orientations. Owing to the depth-speed ambiguity, the magnitude of relative translations is undefined.

Most of these global techniques first solve for rotations and then for translations [16, 30, 1, 33, 22, 35]. Alternatively, rotations and translations can be recovered *simultaneously*, by working on the manifold of rigid motions SE(3) [17]. This approach, although being more principled, is less explored, probably because essential matrices do not fully specify elements of SE(3), due to the scale indeterminacy.

In [17] the scale problem is bypassed by alternation: at each iteration current estimates of absolute orientations are used to fix the scales of the corresponding relative translations. In [1, 52, 10, 23, 30] the scale issue is sidestepped by exploiting implicit or explicit point triangulation, whereas in [16, 8, 22, 33, 35] translation recovery is expressed as a bearing-only network localization where the relative translation directions are regarded as bearing measures that globally constraint the position of the cameras. In [2, 48] the scales are explicitly recovered, which can be traced back again to a bearing-only localization.

In this paper we present a complete pipeline which estimates camera motion starting from the relative orientation of a subset of camera pairs. Our method is grounded on the concept of *group synchronization*, namely the problem of finding group elements from noisy measurements of their ratios (or differences). The pipeline is composed of three stages. We first compute absolute rotations by solving a synchronization problem over $SO(3)$ following the approach in [1]. In the second stage, we estimate the scales by partitioning the epipolar graph into smaller subgraphs: we compute the translations norm locally – using the technique in [2] – and then we globally derive all the scales by solving a synchronization problem over $\mathbb{R}$. In the third stage, we recover absolute translations by solving a synchronization problem over $\mathbb{R}^3$. All the considered synchronization instances translate into direct solutions, namely eigenvalue decompositions (rotations and scales) or linear least squares (translations), which are coupled with iteratively reweighted least squares (IRLS) to gain robustness to outliers. Experimental results show that our pipeline compares favourably to the state of the art in terms of precision and speed.

The **contribution of this paper** is threefold: i) in a comprehensive introduction on group synchronization we

[1] The term "orientation" refers to angular attitude *and* position of a reference frame, as customary in Photogrammetry.

978-1-5090-5407-7/16 $31.00 © 2016 IEEE
DOI 10.1109/3DV.2016.64

set forth a theoretical unified framework were several synchronization problems are seen as instances of a more abstract principle; ii) we propose a divide and conquer technique for computing the scales that exploits a synchronization instance; iii) we demonstrate a complete pipeline for the recovery of camera motion where each stage is based on group synchronization and admits a direct solution.

2. Background on Synchronization

In a network of nodes, suppose that each node has an unknown state and that (noisy) measures of differences (or ratios) of states are available. The goal is to infer the unknown states from the available measures. This is a general statement of the *synchronization problem*. Typically, states are represented by group elements, that is why the problem is actually referred to as *group synchronization*.

2.1. Notation and Preliminary Results

Let $\vec{G} = (V, E)$ be a finite simple digraph with vertex set $V = \{1, \dots, n\}$ and edge set $E \subseteq \{1, \dots, n\} \times \{1, \dots, n\}$, $|E| = m$. Throughout this paper we assume that $\vec{G}$ is connected. In the SfM case $\vec{G}$ represents the *epipolar graph*, which has a vertex for each camera and an edge for each available relative orientation. A denotes the $n \times n$ *adjacency matrix* of $\vec{G}$, i.e. $A_{ij} = 1$ if $(i, j) \in E$ and $A_{ij} = 0$ otherwise, and B denotes the $n \times m$ *incidence matrix* of $\vec{G}$, i.e. $B_{ie} = 1$ and $B_{je} = -1$ for $e = (i, j) \in E$. D denotes the $n \times n$ *degree matrix* of $\vec{G}$, i.e. the diagonal matrix such that D_{ii} is the degree of node i.

A *cycle* in a undirected graph is a subgraph in which every vertex has even degree. A *circuit* is a connected cycle where every vertex has degree two. The set of cycles of a undirected graph can be described algebraically as a vector space over $\mathbb{Z}_2$, where each cycle is represented by its indicator vector and the sum of two cycles is a cycle where the common edges vanish. In a digraph cycles are represented by *signed* indicator vectors, where the sign indicates whether the orientation of the cycle is concordant with the edge orientation or not. A *cycle basis* is a minimal set of circuits (of dimension $m - n + 1$) such that any cycle can be expressed as linear combination of the circuits in the basis [25]. The *integral cycle basis matrix* C is the $(m - n + 1) \times m$ matrix obtained as the stack of signed indicator vectors of the circuits that form the basis.

Let $(\Sigma, *)$ be a group with unit element 1_Σ. In this paper we are particularly interested in $\Sigma = SO(3)$ and $\Sigma = \mathbb{R}^3$, which model angular attitude and position of the cameras, respectively. A *Σ-labelled* graph is a digraph with a labelling of its edge set by elements of Σ, that is a t-uple $\Gamma = (V, E, z)$ where $z : E \to \Sigma$ is such that if $(i, j) \in E$ then $(j, i) \in E$ and $z(j, i) = z(i, j)^{-1}$. So, we may also consider G, the undirected version of $\vec{G}$. In the SfM case, each edge in the epipolar graph is labelled with the relative orientation between the corresponding camera pair.

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

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

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

$$z(e) = x(i) * x(j)^{-1} \ \ \forall e = (i, j) \in E. \tag{2}$$

In other words, each edge label is the ratio of the corresponding vertex labels. Such condition is referred to as *consistency constraint*. It is understood that a consistent labelling is defined up to a global (right) product with any group element.

Proposition 1. *[19] Let $\Gamma = (V, E, z)$ be a Σ-labelled graph. There exists a polynomial algorithm which either finds a non-null cycle in Γ or a consistent labelling of Γ.*

Corollary 1. *The Σ-labelled graph Γ has a consistent labelling if and only if it does not contain a non-null cycle.*

Non-null cycles arise because of *outlying* labels that prohibit a ground truth consistent labelling to be found. In the SfM case such outliers derive from repetitive structures in the images, which cause mismatches leading to false epipolar geometries. Several solutions have been proposed in the literature to detect them, *e.g.* [14, 18, 34, 53, 5]. However, these strategies are computationally demanding and do not scale well with the number of nodes/edges.

2.2. Group Synchronization

Let us assume that the group Σ is equipped with a metric function $d : \Sigma \times \Sigma \to \mathbb{R}^+$ and let $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ be a symmetric positive-definite non-decreasing function with a unique minimum in 0 and $\rho(0) = 0$. Some instances are the quadratic loss function $\rho(y) = y^2$ or robust loss functions used in M-estimators [21].

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

$$\epsilon(\tilde{x}) = \sum_{(i,j) \in E} \rho\Big(d\big(\tilde{z}(i, j), z(i, j)\big)\Big) \tag{3}$$

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

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.

[Group synchronization] Given a Σ-labelled graph $\Gamma = (V, E, z)$, 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 Fig. 1. In the SfM case such unknowns are the absolute orientations of the cameras.

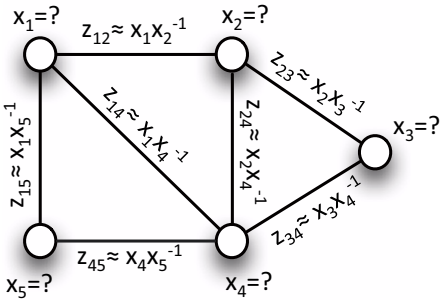

Figure 1: The group synchronization problem.

Several instances of synchronization have been studied in the literature:

- $\Sigma = \mathbb{Z}_2$ *sign synchronization* [11];
- $\Sigma = \mathbb{R}$ *time synchronization* [24, 15] (from which the term *synchronization* originates);
- $\Sigma = \mathbb{R}^d$ state / *translation synchronization* [39, 48];
- $\Sigma = SO(d)$ *rotation synchronization* (a.k.a. rotation averaging) [41, 30, 44, 20, 9, 50, 3];
- $\Sigma = SE(d)$ *rigid-motion synchronization* (a.k.a. motion averaging) [17, 47, 49, 38, 6, 4];
- $\Sigma = SL(d)$ *homography synchronization* [40];
- $\Sigma = \mathbb{S}_d$ *permutation synchronization* [37].

In the next sections we will derive solutions for synchronization problems where the underlying group is $\mathbb{R}$ and show that these solutions can be easily generalized to $\mathbb{R}^d$ and matrix groups such as $SO(3)$. With the aim of building a fast motion estimation pipeline, we aim for *direct* solutions that involve a matrix formulation of the problem.

2.3. Synchronization over $(\mathbb{R}^d, +)$

Let us start by considering the synchronization of real numbers with the addition, namely $(\Sigma, *) = (\mathbb{R}, +)$ (a.k.a time synchronization). A vertex labelling $x : V \to \mathbb{R}$ is consistent with a given edge labelling $z : E \to \mathbb{R}$ if and only if[2]

$$x_i - x_j = z_{ij} \quad \forall\, (i,j) \in E. \tag{4}$$

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

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

$$\boldsymbol{b}_{ij} = [\, 0, \ldots, \underset{\uparrow}{1}_{i}, \ldots, \underset{\uparrow}{-1}_{j}, \ldots, 0\,] \tag{5}$$

then Eq. (4) can be written as $\boldsymbol{b}_{ij}\, [\, x_1, \ldots, x_n\,]^\intercal = z_{ij}$, or equivalently, in matrix form

$$B^\intercal \mathbf{x} = \mathbf{z} \tag{6}$$

where B is the incidence matrix of the *directed* graph $\vec{G}$, $\mathbf{x} \in \mathbb{R}^n$ is the vector containing all the vertex labels and $\mathbf{z} \in \mathbb{R}^m$ is the vector containing all the edge labels (ordered as in B).

We assume that the graph is connected, hence $\operatorname{rank}(B) = n - 1$. Since the solution of the group synchronization problem is defined up to a global group element, we are allowed w.l.o.g. to arbitrarily set $x_k = 0 = 1_\Sigma$ for a chosen $k \in V$. Removing x_k from the unknowns and the corresponding row in B leaves a full-rank $n-1 \times m$ matrix.

With a suitable choice of $d(\cdot, \cdot)$ and ρ in Eq. (3) the consistency error of the synchronization problem writes

$$\epsilon(\mathbf{x}) = \sum_{(i,j)\in E} |x_i - x_j - z_{ij}|^2 = \|B^\intercal \mathbf{x} - \mathbf{z}\|^2 \tag{7}$$

where $\|\cdot\|$ denotes the Euclidean norm, thus the least squares solution of Eq. (6) solves the synchronization problem. Russel *et al*. [39] proved the following result.

Proposition 2. *If $\tilde{\mathbf{x}}$ is the least-squares solution of Eq.* (6)*, then the induced edge labelling $\tilde{\mathbf{z}} = B^\intercal \tilde{\mathbf{x}}$ solves the following constrained minimization problem*

$$\min_{\tilde{\mathbf{z}}} \|\mathbf{z} - \tilde{\mathbf{z}}\|^2 \quad \textit{s.t. } C\tilde{\mathbf{z}} = \mathbf{0} \tag{8}$$

where C is the integral cycle basis matrix.

The interpretation is that the edge labels obtained with Eq. (6) are the closest to the input edge labels among those that yield null-cycles. Indeed, $C\tilde{\mathbf{z}} = \mathbf{0}$ is the stack of the null-cycle constraints (Eq. (1)) for all the cycles in a basis.

Let us now consider the synchronization of real vectors with the addition, namely $(\Sigma, *) = (\mathbb{R}^d, +)$ (a.k.a. translation synchronization). A vertex labelling $\mathbf{x} : V \to \mathbb{R}^d$ is consistent with a given edge labelling $\mathbf{z} : E \to \mathbb{R}^d$ if and only if

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

It is easy to see (reasoning as in the scalar case) that the equation above can be expressed in matrix form as

$$XB = Z \tag{10}$$

where X is the $d \times n$ matrix obtained by juxtaposing all the vertex labels, Z is the $d \times m$ matrix obtained by juxtaposing all the edge labels, and B is the $n \times m$ incidence

matrix of the directed graph $\vec{G}$. By the same token as before, removing one vertex label from the unknowns and the corresponding row in B leaves a full-rank $n-1 \times m$ matrix.

By vectorizing each side of Eq. (10) and using the Kronecker product $\otimes$ [29] we get

$$(B^\intercal \otimes I_d) \operatorname{vec} X = \operatorname{vec} Z \tag{11}$$

where I_d denotes the $d \times d$ identity matrix. This is a generalization of Eq. (6), where the incidence matrix B gets "inflated" by the Kronecker product with I_d in order to cope with the vector representation of the group elements. Thus the vertex labelling $\mathbf{x}$ is computed as the least-squares solution of Eq. (11), as in the case of time synchronization.

It is straightforward to see that Proposition 2 extends to the case of synchronization over $\mathbb{R}^d$, since we can view each component in Eq. (9) as a synchronization over $(\mathbb{R}, +)$.

Robust synchronization. Resistance to outliers (*i.e.* wrong edge labels) can be obtained by replacing $\rho(y) = y^2$ in Eq.(7) with another function $\rho(y)$ with sub-quadratic growth, and solving the resulting minimization problem, *e.g.*, with Iteratively Reweighted Least Squares (IRLS) [21]. 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.

Several weight functions have been proposed in the literature, that correspond to different ρ functions; among them the Cauchy weight function is one of the most popular

$$w_{ij} = \frac{1}{1 + (r_{ij}/c)^2} \tag{12}$$

where $r_{ij} = d\big(\tilde{z}_{ij}, z_{ij}\big)$ and the tuning constant c is chosen so as to yield a reasonably high efficiency in the normal case, and still offer protection against outliers. In particular, $c = 2.385\sigma$ produces 95-percent efficiency when the errors are normal with standard deviation σ. The latter is robustly estimated from the median absolute deviation (MAD) of the residuals as $\sigma = \mathrm{MAD}/0.6745$.

2.4. Synchronization over $(\mathbb{R} \setminus \{0\}, \cdot)$

We now consider the synchronization of real numbers with the multiplication, namely $(\Sigma, *) = (\mathbb{R} \setminus \{0\}, \cdot)$. A vertex labelling $x : V \to \mathbb{R}$ is consistent with a given edge labelling $z : E \to \mathbb{R}$ if and only if

$$z_{ij} = x_i \cdot x_j^{-1} \quad \forall\, (i,j) \in E. \tag{13}$$

Let $\mathbf{x} \in \mathbb{R}^n$ be the vector containing the vertex labels and let $Z \in \mathbb{R}^{n \times n}$ be the matrix containing the edge labels

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & z_{12} & \dots & z_{1n} \\ z_{21} & 1 & \dots & z_{2n} \\ \dots & & & \dots \\ z_{n1} & z_{n2} & \dots & 1 \end{bmatrix}. \tag{14}$$

For a complete graph, the consistency constraint rewrites

$$Z = \mathbf{x}\mathbf{x}^{-\intercal} \tag{15}$$

where $\mathbf{x}\mathbf{x}^{-\intercal}$ contains the edge labels induced by $\mathbf{x}$ and $\mathbf{x}^{-\intercal}$ denotes the row-vector containing the inverse of each vertex label, namely $\mathbf{x}^{-\intercal} = [x_1^{-1}\ x_2^{-1}\ \dots\ x_n^{-1}]$. Note that Eq. (15) implies that $\operatorname{rank}(Z) = 1$.

If the graph is not complete then Z is not fully specified. In this case missing edges are represented as zero entries, i.e. $Z_A := Z \circ A$ represents the matrix of the available measures, where $\circ$ is the Hadamard (or entry-wise) product and A is the adjacency matrix of the graph G. 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. Hence the consistency constraint writes

$$Z_A = (\mathbf{x}\mathbf{x}^{-\intercal}) \circ A. \tag{16}$$

With a suitable choice of $d(\cdot,\cdot)$ and ρ in Eq. (3) the consistency error of the synchronization problem writes

$$\epsilon(\mathbf{x}) = \sum_{(i,j)\in E} |z_{ij} - x_i \cdot x_j^{-1}|^2 = \|Z_A - (\mathbf{x}\mathbf{x}^{-\intercal}) \circ A\|_F^2 \tag{17}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The minimization of ϵ is a non-linear least squares problem. However, a direct solution to a related version of the problem exists, which can be derived by considering the exact (noiseless) case.

If $\epsilon = 0$ then the consistency constraint – after some rewriting – can be expressed as

$$Z_A = \operatorname{diag}(\mathbf{x}) A \operatorname{diag}(\mathbf{x})^{-1} \tag{18}$$

which implies that

$$Z_A \mathbf{x} = \operatorname{diag}(\mathbf{x}) A \mathbf{1}_{n\times 1} = \operatorname{diag}(A\mathbf{1}_{n\times 1})\mathbf{x} = D\mathbf{x} \tag{19}$$

where $\mathbf{1}_{n\times 1}$ is a vector of ones and $D = \operatorname{diag}(A\mathbf{1}_{n\times 1})$ is the *degree matrix* of the graph ($A\mathbf{1}_{n\times 1}$ is the sum of the rows of A). Eq. (19) means that the vertex labelling $\mathbf{x}$ is the eigenvector of $D^{-1}Z_A$ associated to the eigenvalue 1.

Proposition 3. *The matrix $D^{-1}Z_A$ has real eigenvalues. The largest eigenvalue is 1 and it has multiplicity 1.*

Proof. Since diagonal matrices commute, it follows from Eq. (18) that

$$D^{-1}Z_A = \operatorname{diag}(\mathbf{x})(D^{-1}A)\operatorname{diag}(\mathbf{x})^{-1} \tag{20}$$

hence $D^{-1}Z_A$ and $D^{-1}A$ are similar, *i.e.* they have the same eigenvalues. The matrix $D^{-1}A$ is the *transition matrix* of the graph G, which – as a consequence of the Perron-Frobenius theorem [31] – has real eigenvalues and 1 is the largest eigenvalue (with multiplicity 1), if the graph is connected. □

When noise is present, *i.e.* $\epsilon \neq 0$, the eigenvector of $D^{-1}Z_A$ corresponding to the largest eigenvalue is an estimate of the vertex labelling $\mathbf{x}$. In [43] this eigenvalue solution is linked to an algebraic cost function, by generalizing the extremal properties of the standard Rayleigh Quotient known for Hermitian matrices to the case of non-Hermitian matrices with real eigenvalues.

Robust Synchronization. It is easy to see that the above analysis can be extended to handle weighted measurements, which translates in letting the entries of A to assume non-negative values, where 0 still indicates a missing measurement and the other values reflect the reliability of the edge labels. This allows a straightforward extension to gain resilience to outliers via an IRLS-like scheme, i.e. an estimate for the vertex labelling with given weights – stored in the symmetric adjacency matrix A – is obtained from the top eigenvector of $D^{-1}Z_A$, then the weights are updated through, *e.g.*, the Cauchy weight function (12), and these steps are iterated until convergence.

2.5. Synchronization over Matrix Groups

Suppose that Σ is a group which admits a matrix representation through $d \times d$ matrices (*i.e.* Σ can be embedded in $\mathbb{R}^{d\times d}$), where the group operation $*$ reduces to matrix multiplication and $1_\Sigma = I_d$. Some instances are $SO(d)$, $SE(d-1)$ and $SL(d)$. A vertex labelling $x : V \to \mathbb{R}^{d\times d}$ is consistent with a given edge labelling $z : E \to \mathbb{R}^{d\times d}$ 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", namely they are defined as in Eq. (14) with the provision that each element is now a $d \times d$ matrix. In particular, the diagonal of Z is filled with identity matrices I_d. For a complete graph, the consistency constraint rewrites $Z = XX^{-\flat}$ (which implies that rank$(X) = d$), where $X^{-\flat} \in \mathbb{R}^{d\times dn}$ is the block-matrix containing the inverse of each $d \times d$ block of X, i.e. $X^{-\flat} = [x_1^{-1}\ x_2^{-1}\ \dots\ x_n^{-1}]$.

If the graph is not complete then the available measures are represented by $Z_A := Z \circ (A \otimes \mathbf{1}_{d\times d})$, where the adjacency matrix gets "inflated" by the Kronecker product with $\mathbf{1}_{d\times d}$ to match the block structure of the measures. Accordingly, the consistency constraint becomes

$$Z_A = (XX^{-\flat}) \circ (A \otimes \mathbf{1}_{d\times d}) \tag{21}$$

which generalizes Eq. (16).

It can be seen that Eq. (18) and Eq. (19) become

$$Z_A = \mathrm{blkdiag}(X)(A \otimes I_d)\,\mathrm{blkdiag}(X)^{-1} \tag{22}$$

$$Z_A X = (D \otimes I_d)X \tag{23}$$

where $\mathrm{blkdiag}(X)$ produces a $dn \times dn$ block-diagonal matrix with $d \times d$ blocks $x_1, \dots, x_n$ along the diagonal. Thus the columns of X are d eigenvectors of $(D \otimes I_d)^{-1}Z_A$ associated to the eigenvalue 1.

Proposition 4. *The matrix $(D\otimes I_d)^{-1}Z_A$ has real eigenvalues, where Z_A is defined in Eq.* (21)*. The largest eigenvalue is 1 and it has multiplicity d.*

Proof. Similarly to the proof of Proposition 3, we get that $(D \otimes I_d)^{-1}Z_A$ is similar to the matrix $(D \otimes I_d)^{-1}(A \otimes I_d) = (D^{-1}A)\otimes I_d$. 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 and it has multiplicity d. □

In the presence of noise the eigenvectors of $(D \otimes I_d)^{-1}Z_A$ corresponding to the d largest eigenvalues are an estimate of the vertex labelling x. Closure is not always guaranteed: depending on the structure of the group the solution might need to be projected onto Σ.

This solution was introduced in [44] for $\Sigma = SO(2)$, extended in [1, 45] to $\Sigma = SO(3)$, and further generalized in [4, 6] to $\Sigma = SE(3)$. The same formulation appeared in [40] and [37] for $\Sigma = SL(d)$ and $\Sigma = \mathbb{S}_d$ respectively.

To the best of our knowledge, no general results are known linking the eigenvalue solution to the synchronization cost function, which writes

$$\epsilon(X) = \|Z_A - (XX^{-\flat}) \circ (A \otimes \mathbf{1}_{d\times d})\|_F^2 \tag{24}$$

with a suitable choice of $d(\cdot,\cdot)$ and ρ in Eq. (3).

However, results in the case of $\Sigma = SO(3)$ are reported in [1]. In this case, the rank-3 matrix Z containing all the edge labels is *symmetric* and *positive semidefinite*. As a consequence, the consistency error for *rotation* synchronization is

$$\epsilon(X) = \sum_{(i,j)\in E} \|x_{ij} - x_i x_j^{\intercal}\|_F^2. \tag{25}$$

It is shown in [1] that the eigenvalue solution minimizes Eq. (25) under the constraint $X \in V_3(\mathbb{R}^{3n})$, where $V_3(\mathbb{R}^{3n})$ denotes the Stiefel manifold, *i.e.* the columns of X are enforced to be orthonormal rather than imposing that each 3×3 block in X belongs to $SO(3)$. Therefore, after computing the eigenvectors, such blocks are projected onto $SO(3)$ through Singular Value Decomposition [26].

This procedure follows the custom pattern to relax the constraints and subsequently project onto $SO(3)$. Indeed, solving the synchronization problem over $SO(3)$ (a.k.a. *multiple rotation averaging*) is difficult since the feasible set is non-convex, and the cost function may have multiple local minima in different regions of attraction [20]. Other examples of relaxations include semidefinite programming [44, 1] and rank relaxation [3] which, however, produce iterative solutions.

3. Problem formulation

Consider n pinhole cameras that capture the same (stationary) 3D scene. Each camera orientation is described by a rigid motion (or direct isometry), *i.e.* an element of the Special Euclidean Group $SE(3)$, which admits a linear space representation, namely the orientation of camera i is described by the matrix

$$M_i = \begin{bmatrix} R_i & \mathbf{t}_i \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \in SE(3) \tag{26}$$

where $R_i \in SO(3)$ and $\mathbf{t}_i \in \mathbb{R}^3$ represent the rotation and translation component respectively. Similarly, the relative transformation between cameras i and j is represented as

$$M_{ij} = \begin{bmatrix} R_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \in SE(3) \tag{27}$$

with $R_{ij} \in SO(3)$ and $\mathbf{t}_{ij} \in \mathbb{R}^3$.

Let $G = (V, E)$ denote the *epipolar graph* (also known as the *viewing graph* [28]), which has a vertex for each camera and edges in correspondence of the available pairwise transformations. A $SE(3)$-labelled graph, Γ, is obtained from G by labelling the edges with the relative orientations.

The goal of the motion recovery stage of structure from motion is to find a consistent labelling for Γ, where the consistency constraint for the edge (i, j) is $M_{ij} = M_i M_j^{-1}$.

This problem can be tackled directly as a synchronization over $SE(3)$, as done in [17, 47, 49, 6, 4, 38], or by breaking it into rotation and translation, as done in [16, 30, 1, 33, 22, 35] and in this paper, and solving the two sub-problems separately, according to the respective consistency definitions

$$R_{ij} = R_i R_j^\mathsf{T} \tag{28}$$

$$\mathbf{t}_{ij} = -R_i R_j^\mathsf{T} \mathbf{t}_j + \mathbf{t}_i. \tag{29}$$

Note that Eq. (28) defines a *rotation synchronization* problem, thus the angular attitudes of the cameras can be recovered as explained in Sec. 2.5 with $\Sigma = SO(3)$. Equation (29) can be written equivalently as

$$-R_i^\mathsf{T} \mathbf{t}_{ij} = R_j^\mathsf{T} \mathbf{t}_j - R_i^\mathsf{T} \mathbf{t}_i = \mathbf{x}_i - \mathbf{x}_j \tag{30}$$

where $\mathbf{x}_i := -R_i^\mathsf{T} \mathbf{t}_i$ is the centre (position) of the i-th camera. Thus – assuming that rotations have been computed beforehand – recovering camera centres is a *translation synchronization* problem (as defined in Sec. 2.3), where the edge labels are the *baselines* $\mathbf{u}_{ij} := -R_i^\mathsf{T} \mathbf{t}_{ij}$.

In practice, however, due to the depth-speed ambiguity, there is a scale indeterminacy in the relative translations, *i.e.* what can be measured are the *directions* of relative translations $\hat{\mathbf{t}}_{ij} = \mathbf{t}_{ij} / \|\mathbf{t}_{ij}\|$. In other words, the scale factors $\alpha_{ij} = \|\mathbf{t}_{ij}\| = \|\mathbf{u}_{ij}\|$ are unknown. So, in order to exploit the group synchronization framework, one needs to recover these unknown scales (a.k.a. *epipolar scales*), at some point. This issue is addressed in the next section.

4. Epipolar scales recovery

Let $\hat{\mathbf{u}}_{ij}$ be the unit vector that corresponds to the direction of the baseline, also called *bearing*: it is the direction of relative translation expressed in the global reference frame (whereas $\hat{\mathbf{t}}_{ij}$ is local.) Let us rewrite the constraint in Eq. (30) with explicit scale

$$\mathbf{x}_i - \mathbf{x}_j = \alpha_{ij} \hat{\mathbf{u}}_{ij}. \tag{31}$$

Reasoning as in Sec. 2.3 one obtains

$$(B^\mathsf{T} \otimes I_3) \operatorname{vec} X = (I_m \odot U) \boldsymbol{\alpha} \tag{32}$$

where $U \in \mathbb{R}^{3 \times m}$ contains the bearings in columns, $\boldsymbol{\alpha} \in \mathbb{R}^m$ contains all the scale factors and $\odot$ denotes the Khatri-Rao product [27], such that $(I_m \odot U) = \operatorname{blkdiag}(\{\hat{\mathbf{u}}_{ij}\}_{(i,j)\in E})$.

If the scales $\boldsymbol{\alpha}$ where known, this would reduce to Eq. (11) with $(I_m \odot U)\boldsymbol{\alpha} = \operatorname{vec} Z$. We now see how such scales can be recovered linearly, following the derivation given in [2]. The same formulation was derived independently in [48] starting from Eq. (32) and using the identity $CB^\mathsf{T} = 0$.

Let $\{(i_1, i_2), (i_2, i_3), \ldots, (i_\ell, i_1)\}$ be a circuit of G of length ℓ, the null-cycle property in terms of bearings can be written as

$$\sum_{k=1}^{\ell-1} \alpha_{k,k+1} \hat{\mathbf{u}}_{k,k+1} = \alpha_{1\ell} \hat{\mathbf{u}}_{1\ell} \tag{33}$$

which is a homogeneous linear equation in the unknown scales. This equation can also be expressed with the Khatri-Rao product as

$$(\mathbf{c}^\mathsf{T} \odot U)\, \boldsymbol{\alpha} = \mathbf{0}. \tag{34}$$

where $\mathbf{c}$ is the $m \times 1$ signed indicator vector of the circuit.

In a generic epipolar graph, we can stack the equations coming from a cycle basis, thus recovering the translations norm is equivalent to the resolution of a single homogeneous linear system

$$(C \odot U)\, \boldsymbol{\alpha} = \mathbf{0} \tag{35}$$

where C is the integral cycle basis matrix.

Proposition 5. *[2] The unknown epipolar scales can be uniquely (up to a global scale) recovered if and only if* $rank(C \odot U) = m - 1$. *In this case the solution is given by the 1-dimensional null-space of* $(C \odot U)$.

In the special case where the epipolar graph is made of a *single circuit*, then Proposition 5 states that it is possible to recover the translations norm if and only $\ell = 3$ or $\ell = 4$ (provided that the cameras are in a general configuration).

[Parallel-rigidity] A graph for which the dimension of the null space of $(C \odot U)$ is one is said to be parallel rigid.

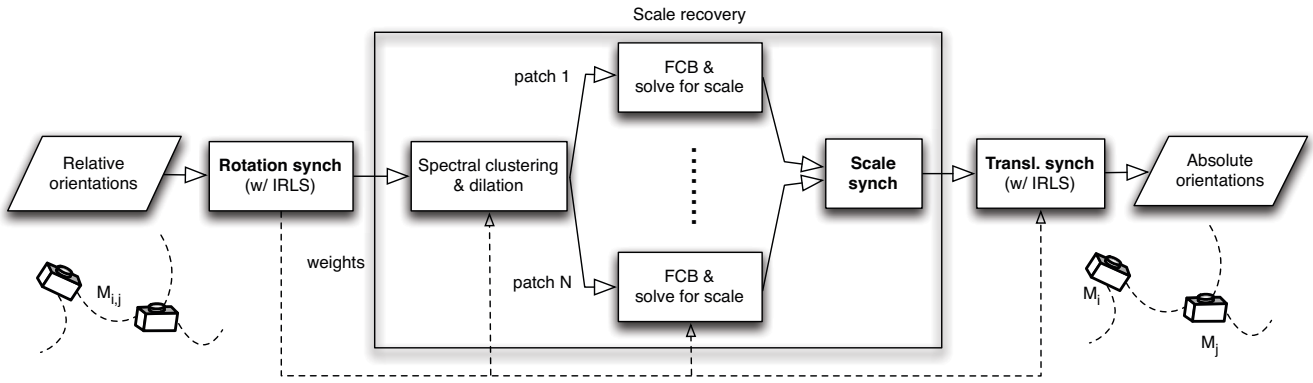

Figure 2: Overview of the GSP. The group synchronization stages are highlighted in bold.

The notion of parallel rigidity emerges in the context of bearing-only network localization [36], but in [48] this problem has been linked to the epipolar scales recovery, hence our definition above is equivalent to the classical one [51].

Regarding the construction of the matrix C, a cycle basis for the graph G is needed. Among several types of basis, the easiest to compute is the *fundamental cycle basis* (FCB). It can be seen (*e.g.* [25]) that if G is connected and T is a spanning tree of G, then adding any edge from $G \setminus T$ to T generates a circuit. The set of such circuits forms a cycle basis, which is referred to as the FCB.

5. Proposed Method

In this paper we demonstrate a complete pipeline (sketched in Fig. 2) that consumes relative orientations of a set of cameras, represented as an epipolar graph, and produces absolute orientations of those cameras. The pipeline – henceforth dubbed GSP, for Group Synchronization Pipeline – exploits group synchronization in most of its stages, which translates into direct solutions such as eigenvalue decompositions (rotations and scales) or linear least squares (translations).

The first stage of GSP is rotation synchronization which is solved through the eigenvalue approach detailed in Sec. 2.5, coupled with IRLS to gain resilience to outliers. Whereas in the IRLS iteration we used a "soft-redescender" – namely the Cauchy function, the final weights are computed with a "hard-redescender" such as the bisquare function (a.k.a. biweight) [32] that assigns zero weight to edges with residual higher than a threshold; in such a way outliers are definitively removed. A fixed threshold on the angular error (5° in our experiments) is also used as a safeguard against high outliers contamination. The final weights are attached to the edges of G, and subsequently used throughout the pipeline wherever it makes sense.

In the second stage the epipolar scales are computed. Since a FCB is subject to error propagation in the presence of long cycles, as observed in [2], we take a *divide-et-impera* approach: partition the graph, solve for the epipolar scales locally, and then globally synchronize all the scales. The idea of partitioning the computation into smaller subproblems is also present, *e.g.*, in [12] and [7].

The weighted epipolar graph is partitioned with spectral clustering, in particular using normalized cuts [42] that tend to produce clusters of approximately the same size. In each cluster the largest parallel-rigid subgraph is extracted as explained in [36]. In order to obtain *overlapping* subgraphs, which are called *patches*, each cluster adopts some nodes of the other clusters, including orphan nodes left out by the parallel-rigid subgraph extraction. More precisely, the adoptable nodes must be connected to the cluster with at least two edges, in order to keep the rigidity of the patch (by the principle of triangulation). They are chosen among the ones with the highest degree and up to a fixed maximum number.

In each patch the scales are computed as described in Sec. 4, using a maximum-weight spanning tree to obtain the FCB. Note that each patch has an unknown global scale and – thanks to the overlap among patches – such scales can be synchronized by running the method described in Sec. 2.4 on the *patch graph* G^P whose vertices are the patches and two patches are adjacent if and only if they share at least one edge of the epipolar graph G. Every such edge in common introduces a measure of the ratio of the scales of the two adjacent patches: a single ratio is attached to the edge of G^P by least squares fitting.

Translation synchronization (as detailed in Sec. 2.3) is run in the end of GSP to compute the position of the cameras in the absolute reference system. The linear system of equations is solved via IRLS, initialized with the weights computed by rotation synchronization.

The subsequent stages (after GSP) are fairly standard and consist in triangulation to initialize the structure and bundle adjustment to refine all the parameters.

		GSP				1DSfM			CLS			LUD			Cui *et al.*		
Dataset	miss %	n	rot.	tra.	time	n	tra.	time	n	tra.	time	n	tra.	time	n	tra.	time
Vienna Cathedral	74	684	1.0	**2.8**	69	836	6.6	302	836	8.8	578	836	5.4	787	578	3.5	242
Alamo	47	499	1.0	0.6	40	577	1.1	158	577	1.3	239	577	**0.4**	385	500	0.6	259
Notre Dame	32	530	0.6	1.5	27	553	10	154	553	0.8	512	553	**0.3**	707	539	**0.3**	366
Tower of London	80	408	2.5	**1.6**	10	572	11	78	572	16	55	572	4.7	88	393	4.4	100
Montreal Notre Dame	52	423	0.5	**0.4**	14	450	2.5	114	450	1.1	180	450	0.5	271	426	0.8	125
Yorkminster	72	386	1.6	**1.4**	10	437	3.4	122	437	6.2	62	437	2.7	103	341	3.7	45
Madrid Metropolis	65	268	2.7	7.5	7	341	9.9	43	341	6.4	46	341	**1.6**	67	–	–	–
NYC Library	68	295	1.5	**1.1**	8	332	2.5	76	332	5.0	52	332	2.0	102	288	1.4	42
Piazza del Popolo	58	297	0.8	**1.0**	14	328	3.1	58	328	3.5	62	328	1.5	88	294	2.6	51

Table 1: Median errors (rotation in degrees, translation in metres) on the datasets from [52] *before* bundle adjustment. Times (in seconds) are net of bundle adjustment. The percentage of missing pairs refers to the largest parallel-rigid component of the input graph. The lowest translation errors are highlighted in boldface.

6. Experiments

We implemented GSP in MATLAB[3] and tested it on irregular large-scale datasets taken from [52], for which recovering camera orientation is challenging. Following the experiments in [52], we used the output of BUNDLER [46] as reference solution, and we computed the optimal rigid transformation between this solution and our estimate with least median of squares (LMedS), using correspondences between camera centres.

We compared GSP with 1DSfM [52], which first removes outlier directions by solving simpler low-dimensional sub-problems, and then computes absolute translations through the Levenberg-Marquardt algorithm. We also included in the comparison the constrained-least-squares (CLS) technique [49], where rotations and translations are initialized separately and then they are jointly refined through Riemannian gradient descent, the least-unsquared-deviations (LUD) solver [35], where absolute translations are derived from a convex program robust to outlier directions, and the method by Cui *et al.* [13], which extends the technique [22] by introducing feature tracks. Our method and 1DSfM use as input the relative orientations provided in [52] as they are, whereas the remaining pipelines internally refine the pairwise directions, hence they require points in input as well.

Results are shown in Table 1, which reports the median translation errors (rotation errors are not analysed in all these papers) *before* applying bundle adjustment, the number of reconstructed cameras, and the running times. The figures of the competing methods are taken from the respective papers [52, 35, 13] (including running times). Table 1 indicates that GSP qualifies among the most accurate solutions. In particular, it outperforms 1DSfM and CLS, and it places itself in the same range of LUD and Cui *et al.* (it compares favourably in 6 out of 9 cases).

The number of cameras reconstructed by GSP is in general smaller than the other methods. Indeed, some nodes/edges are discarded either because IRLS detects them as outliers or during the clustering phase, which involves rigid components extraction. As a matter of fact, these dataset are taken "in the wild", so one cannot expect to reconstruct all the cameras. The high accuracy achieved by our method *without any refinement of the input epipolar geometries* confirms that it correctly removes cameras with outlier measures or low connectivity to the rest of the graph.

Although the running times are not not directly comparable to each other owing to the different hardware, it is worth noting that the running times of GSP are the lowest among the competitors and they have been measured on a less powerful hardware (a MacBook Air with i5 dual-core 1.5 GHz processor and 4GB RAM).

7. Conclusion

In this paper we described a pipeline for computing camera motion from relative orientations that is deeply rooted into the notion of group synchronization. It exploits three different types of group synchronization which admit direct solutions such as eigenvalue decompositions (rotations and epipolar scales) or linear least squares (translations). A comprehensive introduction to the subject is one of the contributions of this paper. In addition, we proposed a divide and conquer technique for computing the epipolar scales: the epipolar graph is partitioned into smaller subgraphs where local scales are computed, and then they are globally synchronized. The experimental evaluation of the pipeline on standard real datasets reveals that it is very fast while achieving competing figures in terms of accuracy.

[3]Code available at http://www.diegm.uniud.it/fusiello/demo/gmf/

References

[1] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. *Proceedings of the Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2012.

[2] F. Arrigoni, A. Fusiello, and B. Rossi. On computing the translations norm in the epipolar graph. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2015.

[3] F. Arrigoni, L. Magri, B. Rossi, P. Fragneto, and A. Fusiello. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 491–498, 2014.

[4] F. Arrigoni, B. Rossi, and A. Fusiello. Spectral synchronization of multiple views in SE(3). *SIAM Journal on Imaging Sciences*, 2016. To appear.

[5] F. Arrigoni, B. Rossi, F. Malapelle, P. Fragneto, and A. Fusiello. Robust global motion estimation with matrix completion. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5:63–70, 2014.

[6] F. Bernard, J. Thunberg, P. Gemmar, F. Hertel, A. Husch, and J. Goncalves. A solution for multi-alignment by transformation synchronisation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[7] B. Bhowmick, S. Patra, A. Chatterjee, V. M. Govindu, and S. Banerjee. Divide and conquer: Efficient large-scale structure from motion using graph partitioning. In *12th Asian Conference on Computer Vision (ACCV 2014)*, 2014.

[8] M. Brand, M. Antone, and S. Teller. Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In *Proceedings of the European Conference on Computer Vision*, 2004.

[9] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the International Conference on Computer Vision*, 2013.

[10] D. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3001–3008, 2011.

[11] M. Cucuringu. Synchronization over Z_2 and community detection in signed multiplex networks with constraints. *Journal of Complex Networks*, 3(3):469–506, 2015.

[12] M. Cucuringu, Y. Lipman, and A. Singer. Sensor network localization by eigenvector synchronization over the Euclidean group. *ACM Transactions on Sensor Networks*, 8(3):19:1 – 19:42, 2012.

[13] Z. Cui, N. Jiang, C. Tang, and P. Tan. Linear global translation estimation with feature tracks. In *British Machine Vision Conference*, 2015.

[14] O. Enqvist, F. Kahl, and C. Olsson. Non-sequential structure from motion. In *Eleventh Workshop on Omnidirectional Vision, Camera Networks and Non-classical Camera*, 2011.

[15] A. Giridhar and P. Kumar. Distributed clock synchronization over wireless networks: Algorithms and analysis. *Proceedings of the IEEE Conference on Decision and Control*, pages 4915–4920, 2006.

[16] V. M. Govindu. Combining two-view constraints for motion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[17] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 684–691, 2004.

[18] V. M. Govindu. Robustness in motion averaging. In *Proceedings of the Asian Conference on Computer Vision*, pages 457–466, 2006.

[19] S. Guillemot. FTP algorithms for path-traversal and cycle-traversal problems. *Discrete Optimization*, 8(1):61 – 71, 2011. Parameterized Complexity of Discrete Optimization.

[20] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International Journal of Computer Vision*, 2013.

[21] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827, 1977.

[22] N. Jiang, Z. Cui, and P. Tan. A global linear method for camera pose registration. In *Proceedings of the International Conference on Computer Vision*, 2013.

[23] F. Kahl and R. Hartley. Multiple-view geometry under the l_∞-norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1603–1617, 2008.

[24] R. Karp, J. Elson, D. Estrin, and S. Shenker. Optimal and global time synchronization in sensornets. Technical report, Center for Embedded Networked Sensing: University of California, Los Angeles.

[25] T. Kavitha, C. Liebchen, K. Mehlhorn, D. Michail, R. Rizzi, T. Ueckerdt, and K. Zweig. Cycle bases in graphs: Characterization, algorithms, complexity, and applications. *Computer Scienze Review*, 3(4):199–243, 2009.

[26] J. Keller. Closest unitary, orthogonal and Hermitian operators to a given operator. *Mathematics Magazine*, 48:192–197, 1975.

[27] C. G. Khatri and C. R. Rao. 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.

[28] N. Levi and M. Werman. The viewing graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 518 – 522, 2003.

[29] S. Liu and G. Trenkler. Hadamard, Khatri-Rao, Kronecker and other matrix products. *International Journal of Information and Systems Sciences*, 4(1):160 – 177, 2008.

[30] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[31] C. D. Meyer. *Matrix Analysis and applied linear algebra*. SIAM, 2000.

[32] F. Mosteller and J. W. Tukey. *Data analysis and regression: a second course in statistics.* Addison-Wesley Series in Behavioral Science: Quantitative Methods, 1977.

[33] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Proceedings of the International Conference on Computer Vision*, pages 3248–3255, 2013.

[34] C. Olsson and O. Enqvist. Stable structure from motion for unordered image collections. In *Proceedings of the 17th Scandinavian conference on Image analysis (SCIA11)*, pages 524–535. Springer-Verlag, 2011.

[35] O. Ozyesil and A. Singer. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674 – 2683, 2015.

[36] O. Ozyesil, A. Singer, and R. Basri. Stable camera motion estimation using convex programming. *SIAM Journal on Imaging Sciences*, 8(2):1220 – 1262, 2015.

[37] D. Pachauri, R. Kondor, and V. Singh. Solving the multiway matching problem by permutation synchronization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1860–1868. Curran Associates, Inc., 2013.

[38] D. M. Rosen, C. DuHadway, and J. J. Leonard. A convex relaxation for approximate global optimization in simultaneous localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5822 – 5829, 2015.

[39] W. Russel, D. Klein, and J. Hespanha. Optimal estimation on the graph cycle space. *IEEE Transactions on Signal Processing*, 59(6):2834 – 2846, 2011.

[40] P. Schroeder, A. Bartoli, P. Georgel, and N. Navab. Closed-form solutions to multiple-view homography estimation. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 650–657, Jan 2011.

[41] G. C. Sharp, S. W. Lee, and D. K. Wehe. Multiview registration of 3D scenes by minimizing error between coordinate frames. In *Proceedings of the European Conference on Computer Vision*, pages 587–597, 2002.

[42] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[43] R. R. D. Silva. A new generalized field of values. *ArXiv e-prints*, 1012.3835, 2010.

[44] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30(1):20 – 36, 2011.

[45] A. Singer and Y. Shkolnisky. 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.

[46] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*, pages 835–846, 2006.

[47] A. Torsello, E. Rodola, and A. Albarelli. Multiview registration via graph diffusion of dual quaternions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2441 – 2448, 2011.

[48] R. Tron, L. Carlone, F. Dellaert, and K. Daniilidis. Rigid components identification and rigidity enforcement in bearing-only localization using the graph cycle basis. In *IEEE American Control Conference*, 2015.

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

[50] L. Wang and A. Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: a Journal of the IMA*, 2(2):145–193, 2013.

[51] W. Whiteley. Matroids from discrete geometry. In *Matroid Theory*, AMS Contemporary Mathematics, pages 171–313. American Mathematical Society, 1997.

[52] K. Wilson and N. Snavely. Robust global translations with 1DSfM. In *Proceedings of the European Conference on Computer Vision*, pages 61–75, 2014.

[53] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1426 – 1433, 2010.