

Global Network Orientation by Synchronization

ISPRS TC II Symposium 2018

Andrea Fusiello*
University of Udine

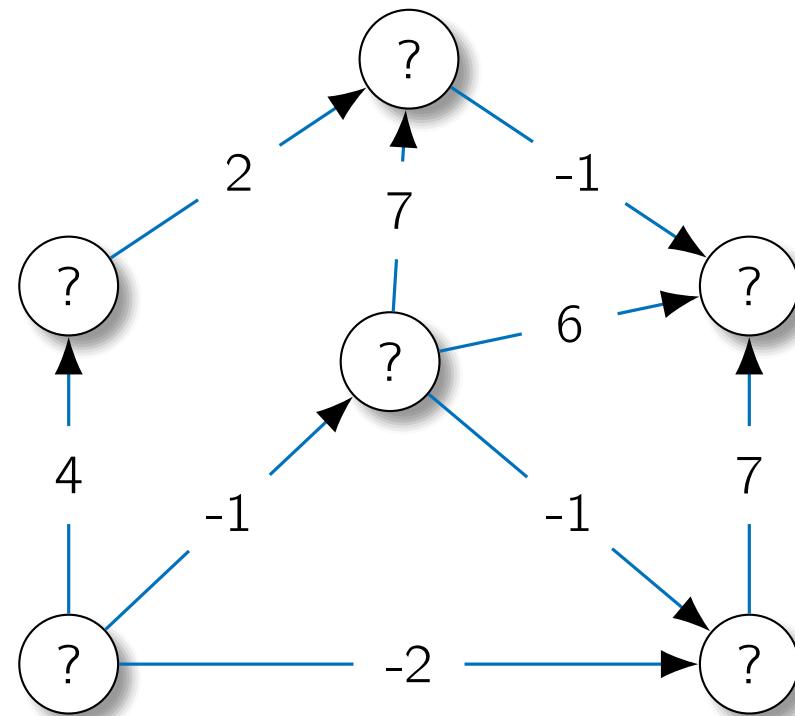
*joint work with Federica Arrigoni

Part 1: theory

Introduction

In a network of nodes, each node has an unknown state and measures of differences (or ratios) of states are available.

Example in \mathbb{Z} :

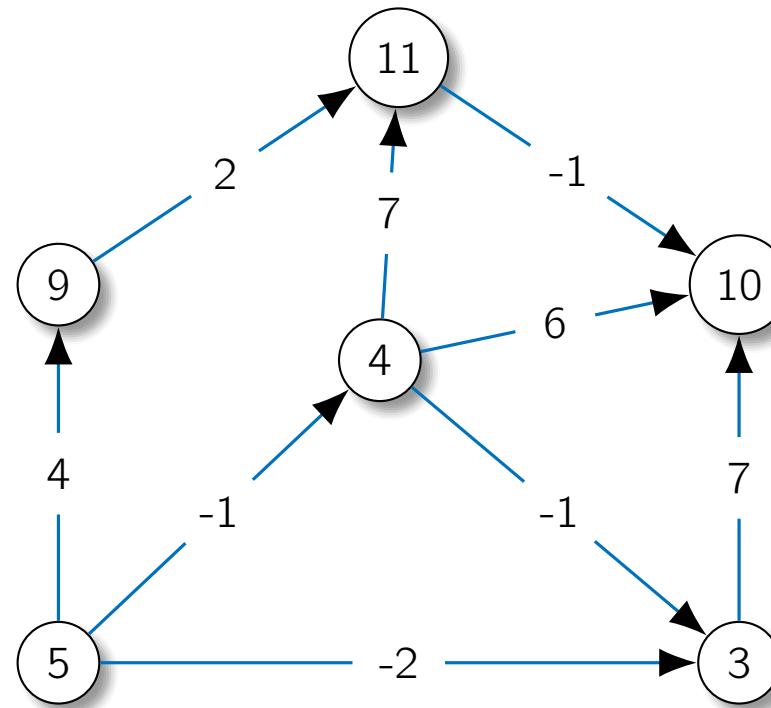


The goal is to guess the unknown states from the available measures.

Introduction

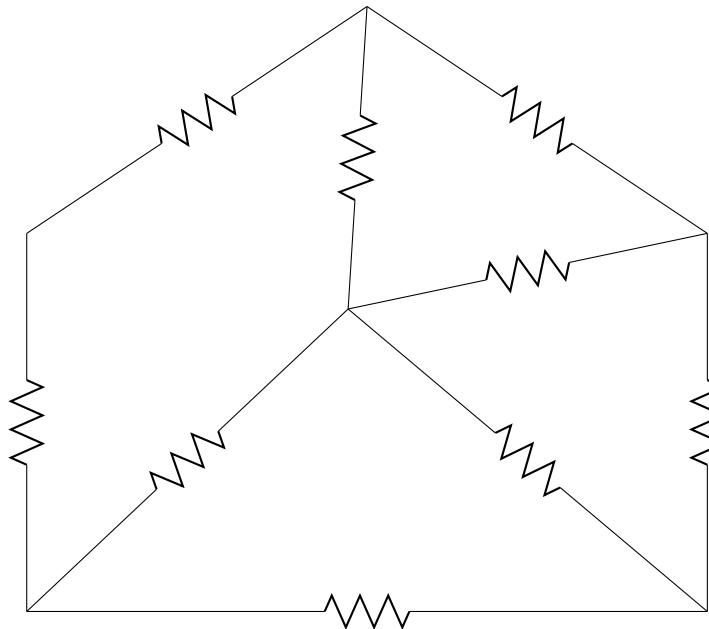
In a network of nodes, each node has an unknown state and measures of differences (or ratios) of states are available.

Example in \mathbb{Z} :



The goal is to guess the unknown states from the available measures.

- Adding a constant to the solution yields another valid solution
- Not every edge labeling produces a solvable problem: circuits must have zero sum.
- Kirchhoff's voltage law (the directed sum of the electrical potential differences around any closed network is zero).



This is an instance of the **synchronization** problem. In general, states can be elements of any **group**, possibly with noisy or wrong measures.

A group is a set Σ with an operation s.t.

- it is closed wrt the operation
- there is an identity 1_Σ
- every element has an inverse.

A related problem is that of **localization**, where the state is a position in 3D space and measures are relative displacements.

1 Consistent labelling

Let Σ be a group with unit element 1_Σ .

States are elements of Σ .

Let $G = (V, E)$ be a finite simple digraph, with $n = |V|$ vertices and $m = |E|$ edges.

Consider a labelling of the edges

$$z : E \rightarrow \Sigma$$

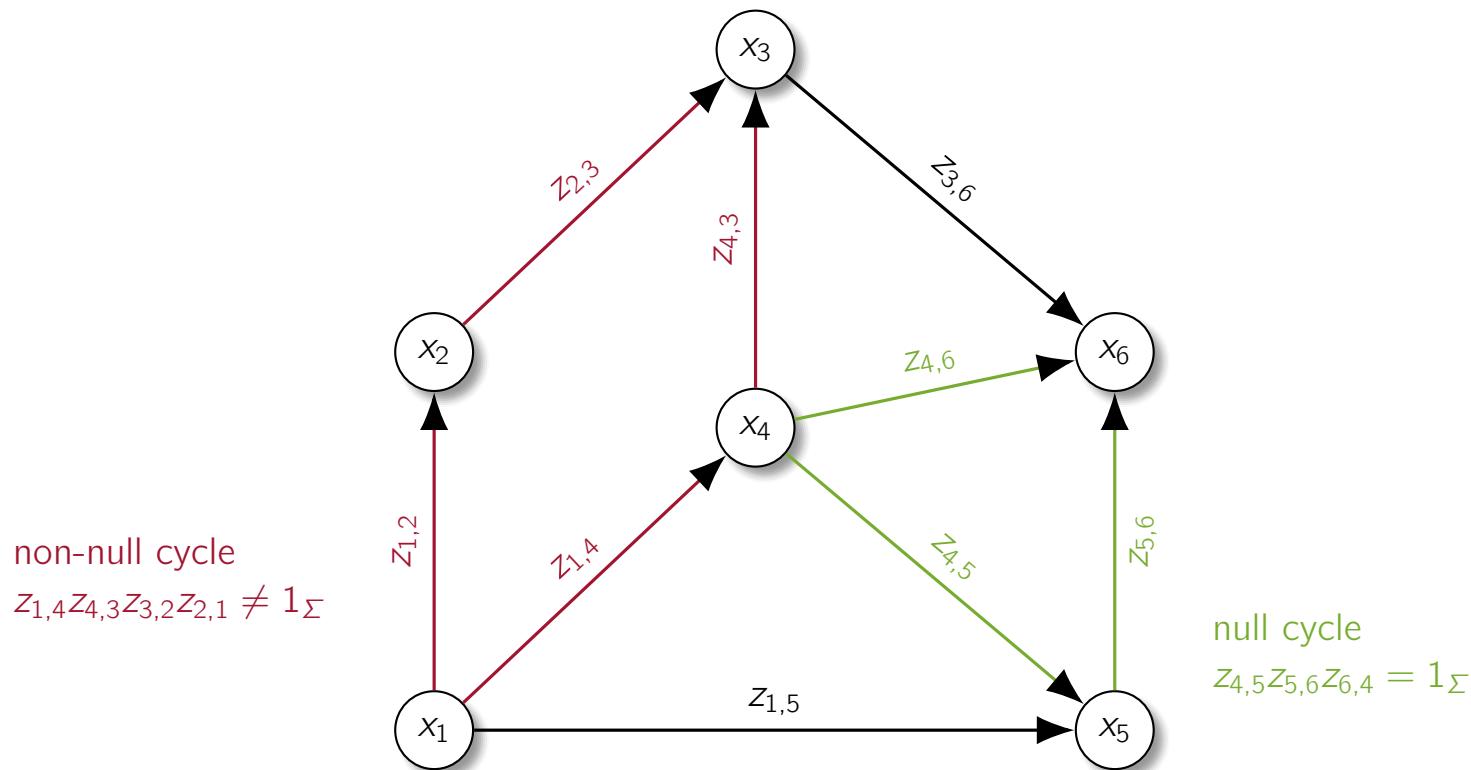
such that if $(u, v) \in E$ then $(v, u) \in E$ and $z(v, u) = z(u, v)^{-1}$.

We say that $\Gamma = (V, E, z)$ is a **Σ -labelled graph**.

A **cycle** in a undirected graph is a subgraph in which every vertex has even degree.

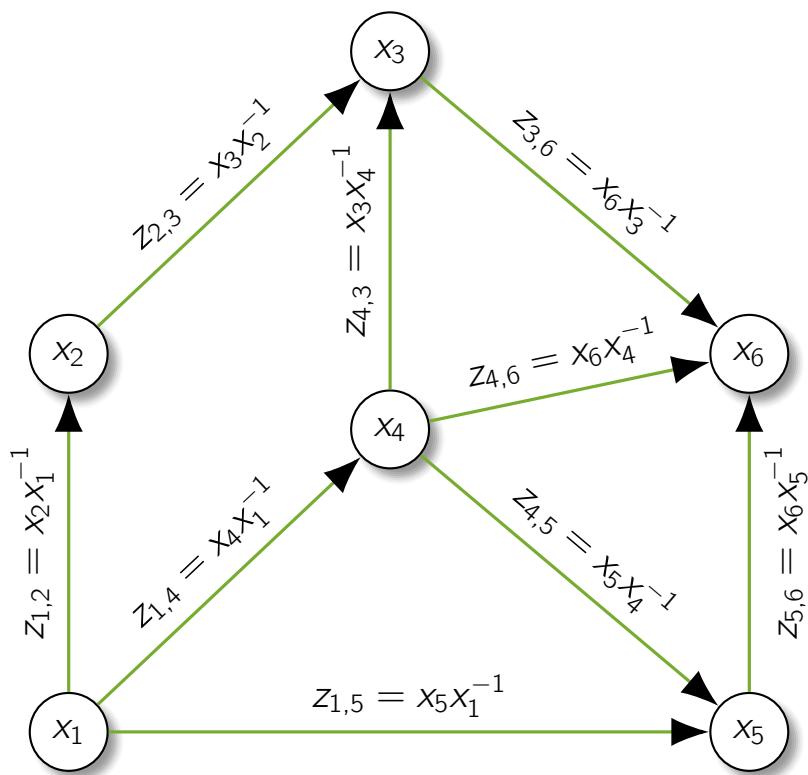
A **circuit** is a connected cycle where every vertex has degree two.

Definition 1.1 (Null cycle) We say that a **cycle** $v_1v_2, v_2v_3, \dots, v_\ell v_1$ in Γ is **null** if and only if $z(v_1, v_2) \cdot z(v_2, v_3) \cdot \dots \cdot z(v_\ell, v_1) = 1_\Sigma$.



Definition 1.2 (Consistent labelling) Let $\Gamma = (G, z)$ be a Σ -labelled graph for $G = (V, E)$. Let $x : V \rightarrow \Sigma$ be a vertex labeling. We say that x is a *consistent labelling* iff

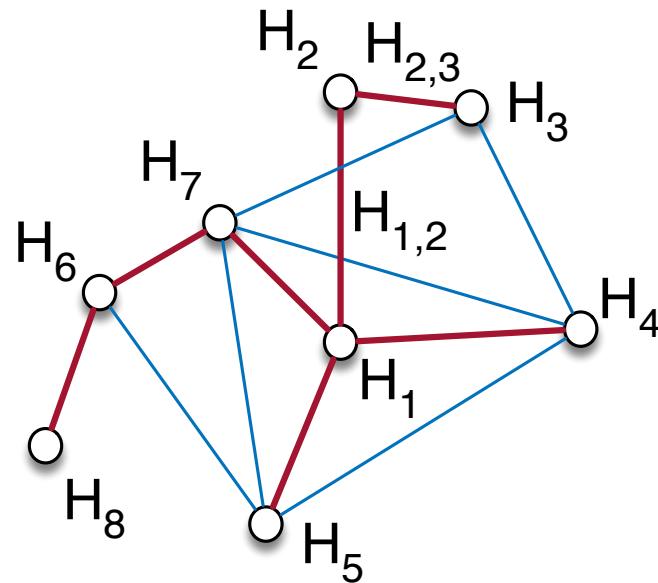
$$z(e) = x(u)^{-1} \cdot x(v) \quad \forall e = (u, v) \in E$$



Proposition 1.3 Let $\Gamma = (G, z)$ be a Σ -labelled graph. Then there exists a polynomial algorithm which either finds a non-null cycle in Γ or finds a consistent labelling of Γ .

Proof. (sketch) Use a spanning tree to label vertices: this is a consistent labelling by construction. Then add one by one the edges not belonging to the spanning tree, thereby creating a cycle. If the cycle is null the edge can be added and leave the labelling consistent, otherwise a non-null cycle has been found.

Corollary 1.4 *The graph Γ has a consistent labeling if and only if all its cycles are null.*



1.1 Finding non-null cycles: GFES

Definition 1.5 (GFES) Let $\Gamma = (G, z)$ be a Σ -labelled graph with labelling z . 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 $S \subseteq E$ with $|S| \leq k$ such that the labelled graph of the remaining edges $\Gamma' = (V, E \setminus S, z)$ does not contain a non-null cycle (i.e. it has a consistent labelling).

The interpretation is that S identifies edges with **outlying labels** that prohibit a ground truth consistent labelling to be found.

Solutions:

- from graph theory community: (Cygan et al., 2012; Wahlström, 2014)
- from CV community: based on outliers rejection heuristics: (Enqvist et al., 2011); (Arrigoni et al., 2014b); based on RANSAC: (Govindu, 2006); based on bayesian inference: (Zach et al., 2010).

2 Synchronization

Find approximately consistent vertex labels given a redundant set of noisy measurements of their ratios stored as edge labels.

Let us assume now that we are given a symmetric positive definite function $f : \Sigma \rightarrow \mathbb{R}$ with a unique minimum at 1_Σ and $f(1_\Sigma) = 0$.

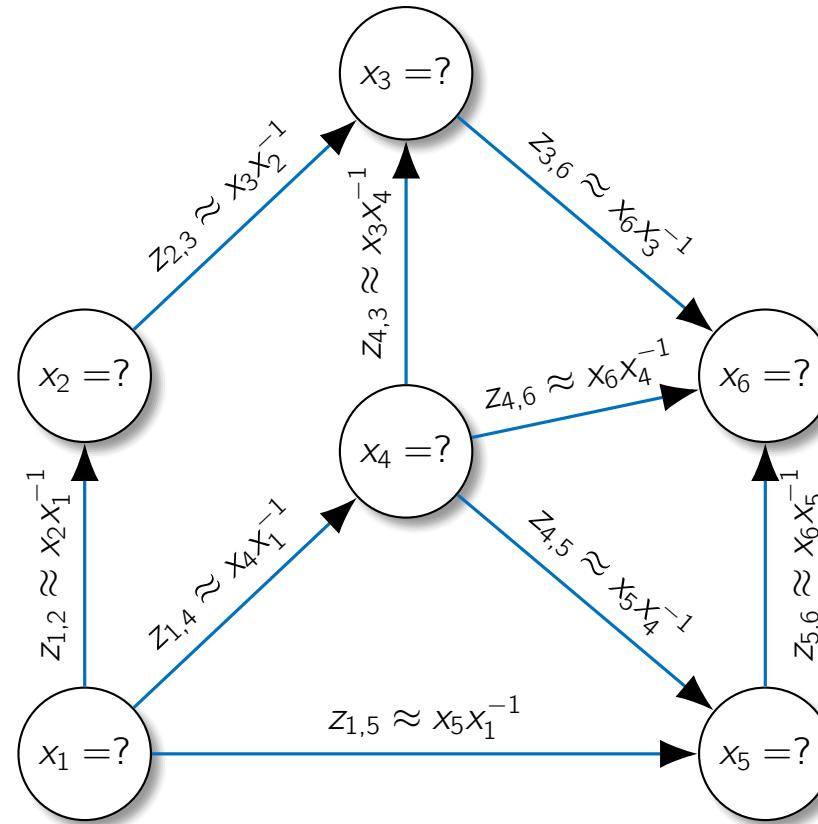
Definition 2.1 (Consistency error) Let $\Gamma = (G, z)$ be a Σ -labelled graph for $G = (V, E)$. Let $\tilde{x} : V \rightarrow \Sigma$ be a vertex labeling. The **consistency error** of \tilde{x} is defined as

$$\epsilon(\tilde{x}) = \sum_{(u,v) \in E} f(\tilde{z}(u, v) \cdot z(u, v)^{-1}) \quad (1)$$

where \tilde{z} is the edge labeling induced by \tilde{x} : $\tilde{z}(u, v) = \tilde{x}(u)^{-1} \cdot \tilde{x}(v)$.

A (vertex) labeling is consistent iff it has zero consistency error.

Definition 2.2 (Synchronization) The *synchronization problem* consists in finding a vertex labeling of G with minimum consistency error, given a (edge) labeling z .



Synchronization can be solved as soon as the graph is connected, but errors compensation happens only within cycles.

Some instances:

- $\Sigma = \mathbb{R}$ time synchronization: (Giridhar and Kumar, 2006)
- $\Sigma = \mathbb{Z}_2$ sign synchronization: (Cucuringu, 2015)
- $\Sigma = \mathbb{R}^d$ state synchronization / translation synchronization: (Barooah and Hespanha, 2007; Russel et al., 2011)
- $\Sigma = SO(3)$ rotation synchronization (averaging): (Martinec and Pajdla, 2007; Singer, 2011; Fredriksson and Olsson, 2012; Hartley et al., 2013; Chatterjee and Govindu, 2013; Arrigoni et al., 2014a; Tron et al., 2016; Wilson et al., 2016)
- $\Sigma = SE(3)$ motion synchronization (averaging): (Fusiello et al., 2002; Sharp et al., 2002; Govindu, 2004; Torsello et al., 2011; Tron and Danilidis, 2014; Rosen et al., 2016; Arrigoni et al., 2015b, 2016c)
- $\Sigma = SL(d)$ homography synchronization: (Schroeder et al., 2011)
- $\Sigma = \mathbb{S}_d$ permutation synchronization: (Pachauri et al., 2013; Yu et al., 2016; Shen et al., 2016; Arrigoni et al., 2017).

2.1 Cycle basis

Viewing cycles as vectors indexed by edges, addition of cycles corresponds to modulo-2 sum of vectors, and the cycles of a graph form a vector space in \mathbb{Z}_2^m .

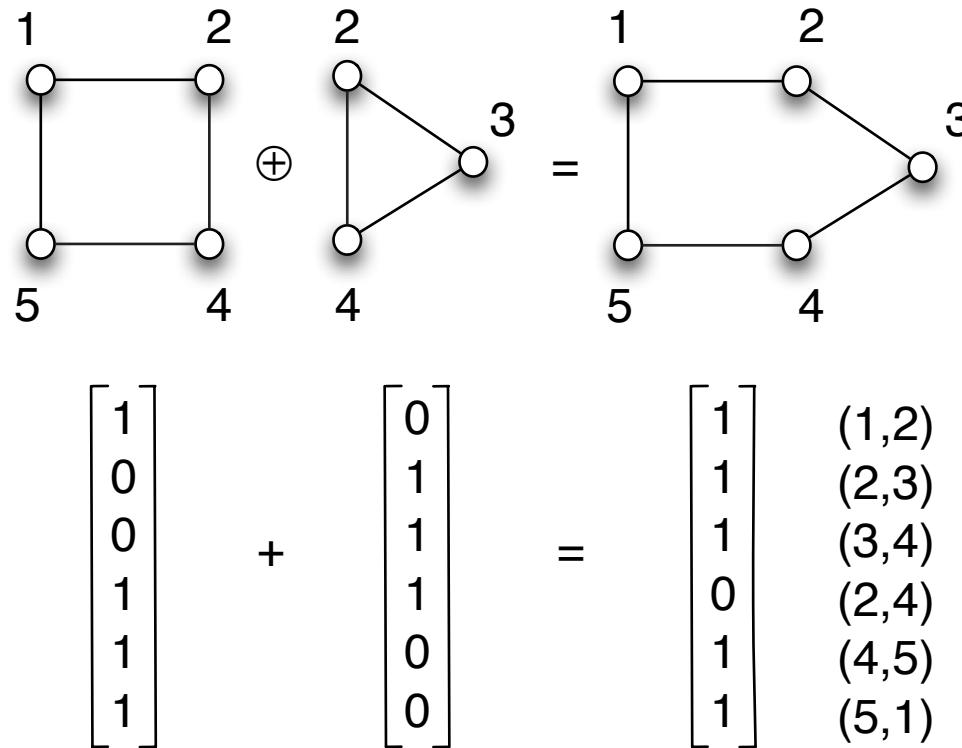
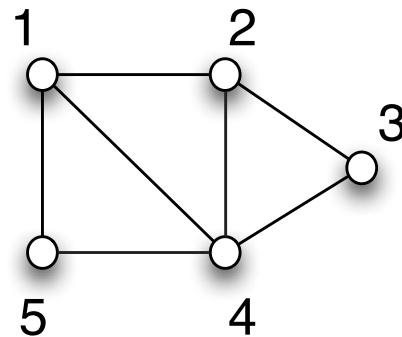


Fig. 1: The sum of two cycles is a cycle where the common edges vanish.

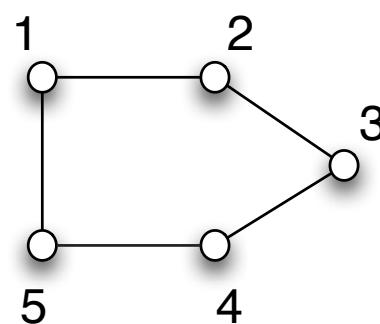
The dimension of such a space is $m - n + c$, where c denotes the number of connected components in $G = (V, E)$.

A **cycle basis** is a minimal set of **circuits** such that any cycle can be written as linear combination of the circuits in the basis.

If we stack the indicator vectors of the circuits of a basis in a matrix C (by rows) we obtain the **cycle basis matrix**.



(a) Graph $G = (V, E)$.



(b) Cycle Basis.

Fig. 2: Example of a cycle basis associated to a given graph $G = (V, E)$. In general, a cycle basis is not unique.

2.2 Synchronization in $(\mathbb{R}, +)$

In $\Sigma = \mathbb{R}$ a vertex labeling $x : V \rightarrow \mathbb{R}$ is consistent with a given edge labeling $z : E \rightarrow \mathbb{R}$ iff

$$x(v) - x(u) = z(u, v) \quad \forall (u, v) \in E \quad (2)$$

Let us denote the incidence vector of the edge (u, v) with

$$\mathbf{b}_{(u,v)} = (0, \dots, \underset{\uparrow}{-1}, \dots, \underset{\downarrow}{1}, \dots, 0)^T \quad (3)$$

Let \mathbf{x} be the vector containing all the vertex labels and \mathbf{z} the vector containing the edge labels (ordered as in B). Equation (2) can be written as

$$\mathbf{x}^T \mathbf{b}_{(u,v)} = \mathbf{z}^T \quad (4)$$

Let B be the $n \times m$ incidence matrix of G , which has the $\mathbf{b}_{(u,v)}$ as columns; it is easy to see that for all the edges the equation above writes $\mathbf{x}^T B = \mathbf{z}^T$, or

$B^T \mathbf{x} = \mathbf{z}$

(5)

Considering $f(\cdot) = \|\cdot\|^2 : \Sigma \mapsto \mathbb{R}^+$, the consistency error of the synchronization problem writes

$$\epsilon(\mathbf{x}) = \sum_{(u,v) \in E} \|x(v) - x(u) - z(u, v)\|^2 \quad (6)$$

The least squares solution of $B^T \mathbf{x} = \mathbf{z}$ solves the synchronization problem.

Oss. If \mathbf{c} is the indicator vector of a cycle, the cycle is null iff $\mathbf{c}^T \mathbf{z} = 0$.

Proposition 2.3 (Russel et al. (2011)) *If $\hat{\mathbf{x}}$ is the least-squares solution of (5), then the induced edge labeling $\tilde{\mathbf{z}} = B^T \hat{\mathbf{x}}$ solves the following constrained minimization problem*

$$\min_{\tilde{\mathbf{z}}} \|\mathbf{z} - \tilde{\mathbf{z}}\|^2 \quad s.t. \quad C\tilde{\mathbf{z}} = 0 \quad (7)$$

where C is the cycle basis matrix.

The interpretation is that: the edge labels produced by the synchronization are the closest to the input edge labels among those that yield null-cycles.

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

In $\Sigma = \mathbb{R}^d$ a vertex labeling $\mathbf{x} : V \rightarrow \mathbb{R}^d$ is consistent with a given edge labeling $\mathbf{z} : E \rightarrow \mathbb{R}^d$ iff

$$\mathbf{x}(v) - \mathbf{x}(u) = \mathbf{z}(u, v) \quad \forall (u, v) \in E \quad (8)$$

Reasoning as in the scalar case, this becomes

$$XB = Z \quad (9)$$

where the columns of X and Z are vectors of \mathbb{R}^d and B is the $n \times m$ incidence matrix of G .

Equivalently, using the **Kronecker product**:

$$(B^T \otimes I_d) \text{vec } X = \text{vec } Z. \quad (10)$$

This is indeed a synchronization problem 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.

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

In $\Sigma = (\mathbb{R} \setminus \{0\}, \cdot)$ a vertex labeling $x : V \rightarrow \mathbb{R}$ is consistent with a given edge labeling $z : E \rightarrow \mathbb{R}$ iff

$$z(u, v) = x(u)^{-1} \cdot x(v) \quad \forall (u, v) \in E \quad (11)$$

The consistency constraint can be expressed in an equivalent compact matrix form.

Let \mathbf{x} be the vector containing the vertex labels and let Z be the matrix containing the edge labels

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

For a complete graph, the consistency constraint rewrites

$$Z = \mathbf{x}\mathbf{x}^{-T} \quad (13)$$

where $\mathbf{x}\mathbf{x}^{-T}$ contains the edge labels induced by \mathbf{x} (with a little abuse of notation, we define \mathbf{x}^{-1} as the vector containing the inverse of each element of $\mathbf{x} \in \Sigma$).

If the graph G is not complete (missing measures) we shall write the constraint as

$$\underbrace{(Z \circ A)}_{Z_A} = (\mathbf{x}\mathbf{x}^{-T}) \circ A \quad (14)$$

where A is the (0-1) adjacency matrix of G and \circ is the Hadamard product.

The synchronization problem rewrites:

$$Z_A = A \circ (\mathbf{x}\mathbf{x}^{-T}) \Rightarrow Z_A \mathbf{x} = D\mathbf{x} \iff \boxed{D^{-1} Z_A \mathbf{x} = \mathbf{x}} \iff \boxed{(D - Z_A)\mathbf{x} = 0} \quad (15)$$

where $D = \text{diag}(A\mathbf{1})$ is the degree matrix of the graph.

- The solution \mathbf{x} is the eigenvector of $D^{-1}Z_A$ associated to eigenvalue 1;
- The solution \mathbf{x} belongs to the null-space of $(D - Z_A)$.

Considering $f(\cdot) = \|1 - \cdot\|^2 : \Sigma \mapsto \mathbb{R}^+$, the consistency error of the synchronization problem writes

$$\epsilon(\mathbf{x}) = \sum_{(u,v) \in E} \|z(u, v) - x(u)^{-1} \cdot x(v)\|^2 \quad (16)$$

In matrix form, the cost function of the synchronization rewrites

$$\epsilon(\mathbf{x}) = \|(Z - \mathbf{x}\mathbf{x}^{-T}) \circ A\|_F^2 \quad (17)$$

To the best of our knowledge, no general results are known linking the algebraic solutions (eigen and SVD) to the synchronization cost function, in the noisy case.

Note also that the two algebraic solutions do not coincide in this case.

3 Synchronization in $GL(d)$

In this case Σ is the General Linear Group, i.e., the group of $d \times d$ invertible matrices over \mathbb{R} . Hence the (multiplicative) synchronization problem can be instantiated with:

$$X = \begin{bmatrix} X_1^{-1} \\ X_2^{-1} \\ \dots \\ X_n^{-1} \end{bmatrix}, \quad Y = [X_1, X_2, \dots, X_n], \quad Z = \begin{bmatrix} I & X_{12} & \dots & X_{1n} \\ X_{21} & I & \dots & X_{2n} \\ \dots & \dots & \dots & \dots \\ X_{n1} & X_{n2} & \dots & I \end{bmatrix} \quad (18)$$

The consistency constraint rewrites (complete graph):

$$Z = XY \quad (19)$$

The solution X is recovered as (general graph):

- the d top eigenvectors of $(D \otimes I_d)^{-1} Z_A$;
- the d -dimensional null-space of $(D \otimes I_d) - Z_A$.

where $Z_A = Z \circ (A \otimes \mathbf{1}_{d \times d})$.

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., solution is unique up to the action of an element of $GL(d)$.

Projection Blocks of U are in general non-singular, hence they belong to $GL(d)$. This is an **intrinsic solution**: no projection is needed.

The analysis carried out in this section 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 the solution is not intrinsic any more: it need to be projected onto Σ .

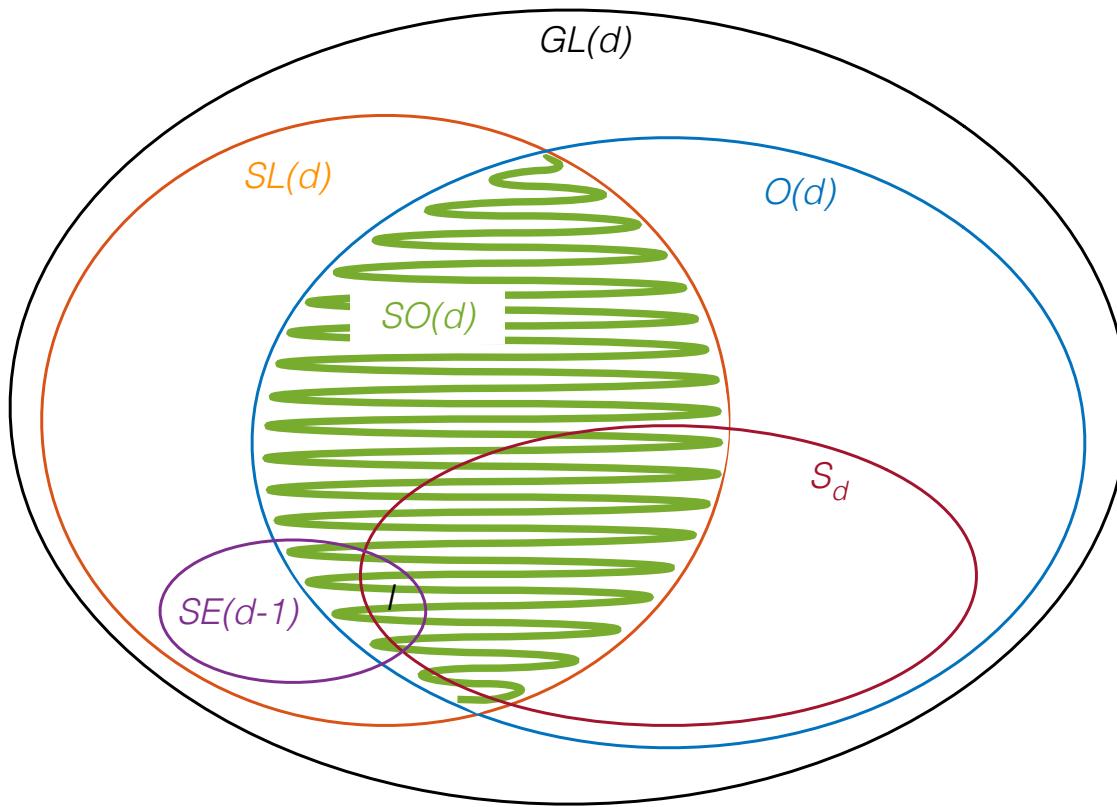


Fig. 3: Subgroups of $GL(d)$. The identity is in the intesection of all of them.

Let us summarize...

- If the group is additive, solve the linear system

$$B^T \mathbf{x} = \mathbf{z}.$$

- The measures (state differences) are stacked in \mathbf{z} according to the edge ordering; B is the **incidence** matrix of the graph.
- If the group is multiplicative, solve the eigenvalues problem

$$(D^{-1} Z_A) \mathbf{x} = \mathbf{x}$$

- The measures (state ratios) are in a matrix $Z_A = Z \circ A$ where A is the **adjacency** matrix of the graph. ($D = \text{diag}(A\mathbf{1})$)
- This holds for scalar states; if the state is a matrix, everything must be suitably “inflated” with a wise use of Kronecker products.
- The eigen or null-space solution is **intrinsic** in $GL(d)$ but (we will see) it is **extrinsic** in its subgroups.

Break

3.1 Synchronization over $SL(d)$

Let us consider the Special Linear Group $SL(d)$, that is the set of $d \times d$ matrices with unit determinant

$$SL(d) = \{R \in \mathbb{R}^{d \times d} \text{ s.t. } \det(R) = 1\}. \quad (20)$$

Since $SL(d)$ is a subgroup of $GL(d)$, the synchronization problem can be addressed either via the spectral solution, or via the null-space solution.

Alternatively, the null-space of $I_{dn} - (D \otimes I_d)^{-1}Z_A$ can be computed, as done in Schroeder et al. (2011).

Ambiguity Let U be the $dn \times d$ matrix containing either the output of the spectral method or the null-space solution.

U is the solution up to multiplication element of $GL(d)$, which can be reduced to $SL(d)$ after permutation of two columns of U s.t. $\det(U_1)$ is positive and division by $\sqrt[d]{\det(U_1)}$.

Projection 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 $\sqrt[d]{\det(U_i)}$.

3.2 Synchronization over $O(d)$

$O(d)$ is the group of orthogonal matrices, i.e.,

$$O(d) = \{R \in \mathbb{R}^{d \times d} \text{ s.t. } R^T R = R R^T = I_d\}. \quad (21)$$

In matrix form:

$$X = \begin{bmatrix} R_1^T \\ R_2^T \\ \dots \\ R_n^T \end{bmatrix}, \quad Y = [R_1, R_2, \dots, R_n], \quad Z = \begin{bmatrix} I & R_{12} & \dots & R_{1n} \\ R_{21} & I & \dots & R_{2n} \\ \dots & \dots & \dots \\ R_{n1} & R_{n2} & \dots & I \end{bmatrix} \quad (22)$$

With respect to the general case, $O(3)$ is a special one, for $Y = X^T$ and Z is symmetric and positive semidefinite.

Therefore, the consistency constraint $R_{ij} = R_i^T R_j$ becomes:

$$Z = XY = XX^T. \quad (23)$$

In this case, Z admits an **orthonormal basis of real eigenvectors**.

Ambiguity A change of the orthonormal basis in the eigenspace corresponds to right-multiplication by an orthogonal $d \times d$ matrix, and this agrees with the fact that the solution is defined up to an element of $O(d)$.

Projection Blocks of U are not guaranteed to be orthogonal matrices (only columns of U are orthogonal vectors).

For each block of U find the closest orthogonal matrix (orthogonal procrustes problem).

3.3 Synchronization over \mathcal{S}_d

Let us consider the Symmetric Group \mathcal{S}_d , that 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.

Since \mathcal{S}_d is a subgroup of $O(d)$, then solution is as in $O(d)$. and Z_A is symmetric.

Ambiguity Let us consider the spectral method and suppose that the eigenvectors of $(D \otimes I_d)^{-1} Z_A$ corresponding to the d largest eigenvalues are collected in a $nd \times d$ matrix U .

Any orthogonal basis for the eigenspace is a solution: the solution is defined up to multiplication by an element of $O(d)$.

However, the solution to permutation synchronization is *not* invariant to multiplication by an element of $O(d)$. Indeed, it is defined up to an element of \mathcal{S}_d .

Let Q the unknown orthogonal transformation s.t. $U = XQ$, 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 $U_1 U_1^T = X_1 X_1^T = I_d$, i.e. $U_1 \in O(d)$;

Moreover, from $U = XQ$ we have $U_1 = X_1 Q$, so $Q = U_1$ up to a permutation.

Projection Even after fixing the ambiguity, the blocks of U will not be permutation matrices (in general). We need to project onto permutation matrices: permutation procrustes problem.

3.4 Rotation synchronization

It is also known as **multiple rotation averaging** (Hartley et al., 2013).

Rotation synchronization is a particular case of the synchronization problem in the group of rotations $SO(3) = \{R \in \mathbb{R}^{3 \times 3} \text{ s.t. } R^T R = I, \det(R) = 1\}$.

Rotation synchronization can be cast as the following problem:

$$\begin{aligned} & \text{minimize} && \sum_{(i,j) \in E} \|R_{ij} - R_i^T R_j\|_F^2 \\ & \text{subject to} && R_i \in SO(3) \end{aligned} \tag{24}$$

Singer (2011); Arie-Nachimson et al. (2012) prove that taking the top 3 eigenvectors of $(D \otimes I_3)^{-1} Z_A$ corresponds to solving a relaxed rotation synchronization with orthonormality of the columns of X (instead of each 3×3 block.)

Ambiguity The solution is defined up to an element of $O(d)$, which reduces to $SO(d)$ after permutation of columns of U s.t. $\det(U_1)$ is positive.

Projection Blocks of U are not guaranteed to be orthogonal matrices (only columns of U are orthogonal vectors).

For each block of U find the closest matrix in $SO(3)$ (orthogonal procrustes problem). Solve with SVD.

3.5 Motion synchronization.

Motion synchronization is an instance of the $GL(d)$ synchronization with $\Sigma = SE(3)$.

Each group element is described by a homogeneous rigid transformation

$$M_i = \begin{pmatrix} R_i & \mathbf{x}_i \\ \mathbf{0} & 1 \end{pmatrix} \in SE(3) \quad M_{ij} = \begin{pmatrix} R_{ij} & \mathbf{x}_{ij} \\ \mathbf{0} & 1 \end{pmatrix} \in SE(3) \quad (25)$$

where $R_i, R_{ij} \in SO(3)$ and $\mathbf{x}_i, \mathbf{x}_{ij} \in \mathbb{R}^3$ represent the rotation and translation components of the isometry.

The vertex labeling is consistent iff $M_{ij} = M_i^{-1}M_j$, which is equivalent to

$$R_{ij} = R_i^T R_j \quad (26)$$

$$\mathbf{x}_{ij} = R_i^T \mathbf{x}_j - R_i^T \mathbf{x}_i \quad (27)$$

by considering separately the rotation and translation terms.

Ambiguity Solution is unique up to the action of an element of $GL(d)$. However we would like to have an ambiguity up to an element of $SE(d)$:

- apply a linear combination of the columns of U such that the fourth row becomes $[0 \ 0 \ 0 \ 1]$
- apply the transform that bring the first 3×3 block onto $SO(3)$, using the polar decomposition of U_1 , the first 3×3 block of U : $U_1 = RP$ with $R \in O(d)$ and P symmetric p.d. Since U_1 is invertible, $R = U_1 P^{-1}$

Projection The 4×4 blocks in U are not guaranteed to belong to $SE(3)$. In order to project the solution onto $SE(3)$:

- fix every fourth row to be $[0 \ 0 \ 0 \ 1]$
- project each 3×3 block onto $SO(3)$.

3.6 Translation synchronization.

The consistency constraint for translations (27):

$$\mathbf{x}_{ij} = R_i^T \mathbf{x}_j - R_i^T \mathbf{x}_i \quad (28)$$

can be written equivalently as

$$R_i \mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i := \mathbf{u}_{ij} \quad (29)$$

where \mathbf{x}_i is the centre of the i -th camera and \mathbf{u}_{ij} is the **baseline**.

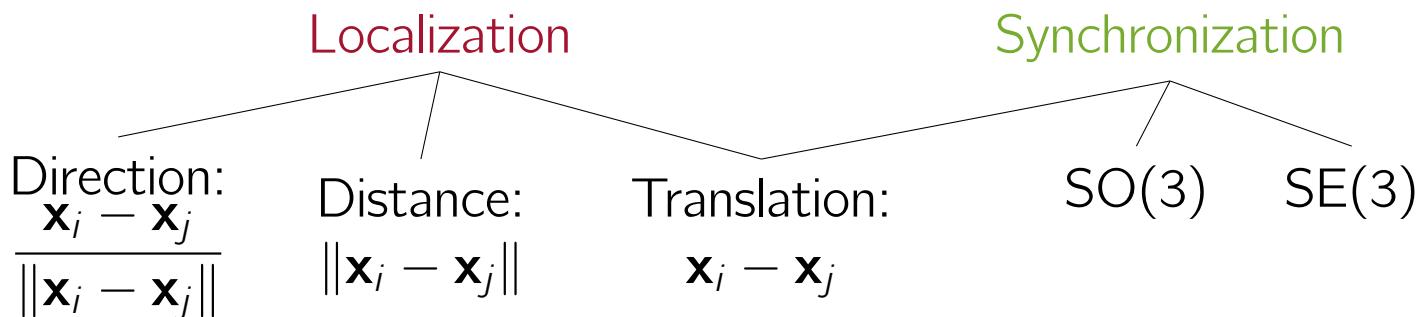
By juxtaposing all the m baselines \mathbf{u}_{ij} in one $3 \times m$ matrix U , we obtain an instance of synchronization in $(\mathbb{R}^d, +)$, where the node labels are the position of the cameras:

$$(B^T \otimes I_3) \text{vec } X = \text{vec } U. \quad (30)$$

4 Localization

The goal of localization is to compute the position of n nodes in d -space given measures on the edges.

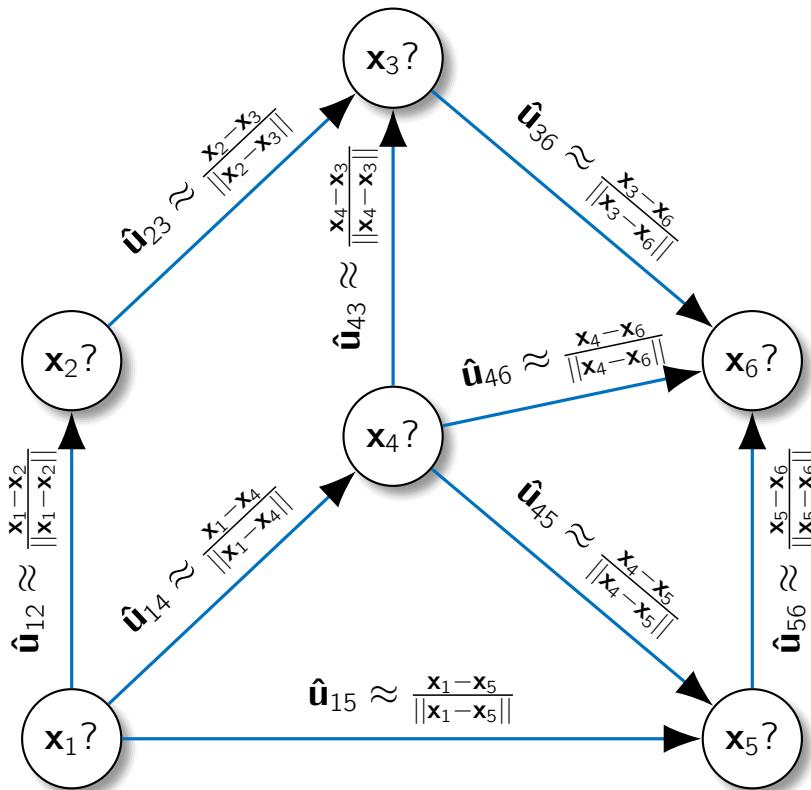
As in a translation synchronization problem the states of the nodes \mathbf{x}_i are positions, but the available measures are *not* differences of the states, in general.



- distance-based: only magnitude of translation is known;
- direction-based: only direction of translation is measured (AOA, bearings);

4.1 Bearing-based localization

The goal is to recover the position of n nodes in d -space, where pairs of nodes can measure the direction of the line joining their locations.



The solution is defined up to translation and scale.

Bearings $\hat{\mathbf{u}}_{ij}$ are defined as

$$\hat{\mathbf{u}}_{ij} = \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|}. \quad (31)$$

They are available after solving for relative orientation and **rotation synchronization**, since they are directions expressed in a global frame.

Notation recap.

The **relative translation** $\hat{\mathbf{x}}_{ij}$ is defined in a local reference frame;

The **baseline** $\mathbf{u}_{ij} = R_i^T \hat{\mathbf{x}}_{ij}$ is defined in a global reference frame;

The **bearing** $\hat{\mathbf{u}}_{ij} = \mathbf{u}_{ij}/\|\mathbf{u}_{ij}\|$ is the direction of the baseline.

4.1.1 Incidence-based formulation

Let us multiply the translation synchronization equation:

$$(B^T \otimes I_3) \text{vec } X = \text{vec } U \quad (32)$$

by the block diagonal matrix

$$\widehat{S} = \text{blkdiag}(\{[\widehat{\mathbf{u}}_{ij}]_\times\}_{(i,j) \in E})$$

yielding

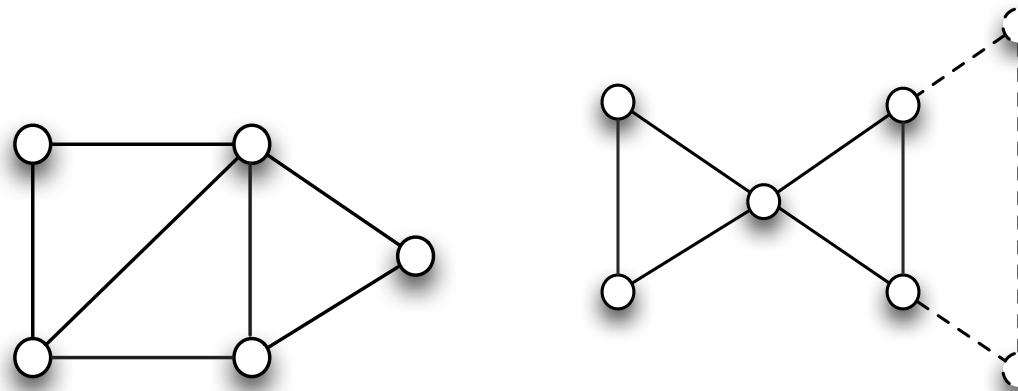
$$\widehat{S}(B^T \otimes I_3) \text{vec } X = \widehat{S} \text{vec } U = 0 \quad (33)$$

This step has the effect of substituting U , which is unknown, with \widehat{S} (derived from \widehat{U}) which is known instead.

The solution is the null space of $\widehat{S}(B^T \otimes I_3)$ (Tron et al., 2015).

Requiring that the underlying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is connected is not sufficient for unique localizability.

In general, the graph must be **parallel rigid** (Whiteley, 1997).



Parallel Rigid

Flexible

4.1.2 Parallel rigidity

The classical characterization of parallel rigidity is given in terms of the rank of the **parallel rigidity matrix**, which is defined with reference to a particular embedding \mathbf{x} (coordinates) of the graph $G = (V, E)$:

$$Q_{p,G} = \widehat{S}_{\mathbf{x}}(B^T \otimes I_3)$$

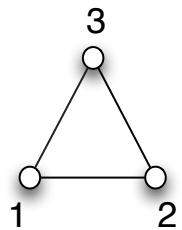
where $\mathbf{x} \in \mathbb{R}^{nd}$ ($n = |V|$) is an embedding of the graph, and $\widehat{S}_{\mathbf{x}}$ contains the induced bearings.

Whiteley (1997) demonstrates that a point formation (i.e., graph + embedding) in d -dimensions is parallel rigid iff $\text{rank}(Q_{p,G}) = d|V| - (d + 1)$ (in our case, $3n - 4$)

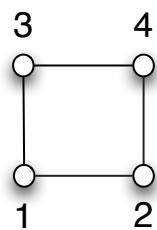
The notion of **generic parallel rigidity**, instead, is a property of the graph G without reference to a specific embedding; A graph $G = (V, E)$ is generically parallel rigid in d dimensions iff

$$\max_{\mathbf{x} \text{ is generic}} \text{rank}(Q_{\mathbf{x},G}) = d|V| - (d + 1)$$

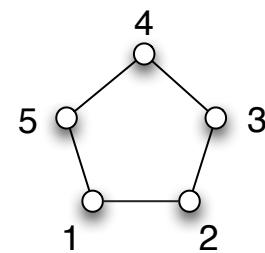
where \mathbf{x} is a **generic** (i.e., no algebraic dependency among coordinates) embedding of the graph.



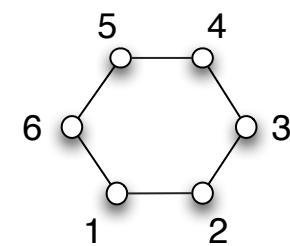
parallel rigid



parallel rigid

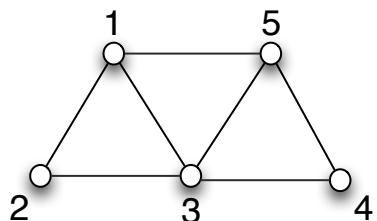


flexible

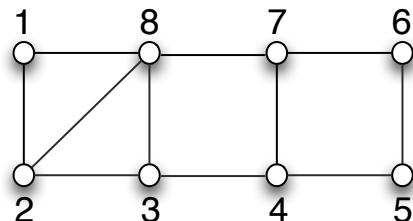


flexible

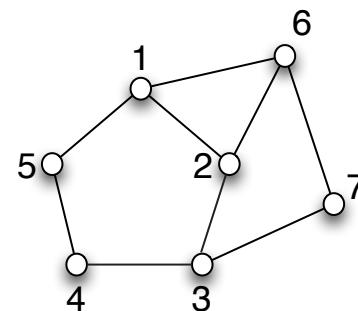
...



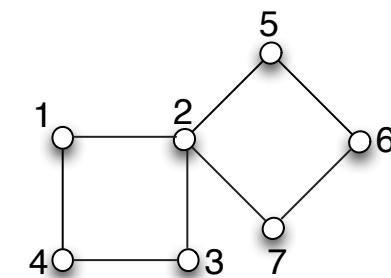
parallel rigid



parallel rigid



parallel rigid



flexible

4.1.3 Cycle-based formulation

Let us rewrite the consistency constraint of synchronization over \mathbb{R}^3 in terms of directions $\hat{\mathbf{u}}$ and magnitudes α .

$$\mathbf{x}_i - \mathbf{x}_j = \mathbf{u}_{ij} \quad \forall (i, j) \in \mathcal{E} \iff (B^T \otimes I_3) \text{vec}(X) = \text{vec}(U)$$

$$\mathbf{x}_i - \mathbf{x}_j = \alpha_{ij} \hat{\mathbf{u}}_{ij} \quad \forall (i, j) \in \mathcal{E} \iff (B^T \otimes I_3) \text{vec}(X) = (I_m \odot \hat{U})\boldsymbol{\alpha}$$

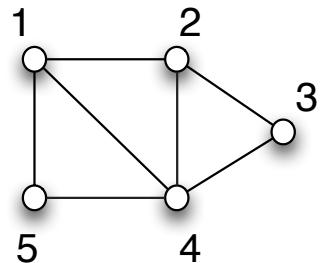
where $\hat{U} = [\underbrace{\hat{\mathbf{u}}_{12} \dots \hat{\mathbf{u}}_{ij} \dots}_m]$ and \odot is the Khatri-Rao product.

Let us consider a cycle matrix C and multiply both sides by $C \otimes I_3$.

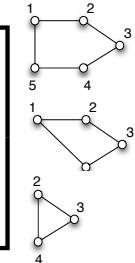
$$(CB^T \otimes I_3) \text{vec}(X) = (C \odot \hat{U})\boldsymbol{\alpha} \underset{CB^T=0}{\implies} 0 = (C \odot \hat{U})\boldsymbol{\alpha}$$

Example.

1. Compute a cycle basis for the epipolar graph.



$$C = \begin{bmatrix} (1,2) & (2,4) & (4,1) & (2,3) & (3,4) & (4,5) & (5,1) \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$



2. Solve a homogeneous linear system.

$$(C \odot \hat{U})\alpha = 0 \iff \begin{bmatrix} \hat{u}_{12} & 0 & 0 & \hat{u}_{23} & \hat{u}_{34} & \hat{u}_{45} & \hat{u}_{51} \\ \hat{u}_{12} & 0 & \hat{u}_{41} & \hat{u}_{23} & \hat{u}_{34} & 0 & 0 \\ 0 & -\hat{u}_{24} & 0 & \hat{u}_{23} & \hat{u}_{34} & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_{12} \\ \alpha_{24} \\ \alpha_{41} \\ \alpha_{23} \\ \alpha_{34} \\ \alpha_{45} \\ \alpha_{51} \end{bmatrix} = 0$$

Proposition 4.1 (Arrigoni et al. (2015a)) *Node locations can be uniquely (up to translation and scale) determined from pairwise directions if and only if edge magnitudes can be uniquely (up to scale) recovered from pairwise directions.*

$$\text{parallel rigidity} \iff \text{rank}(C \odot \hat{U}) = m - 1$$

4.1.4 Laplacian formulation

The least squares solution of bearing-based localization (both methods) is equivalent to minimize the following quadratic form

$$\min_{\|\mathbf{x}\|=1} \text{vec}(X)^T H \text{vec}(X). \quad (34)$$

where

$$H = \begin{bmatrix} \sum_j [\hat{\mathbf{u}}_{1,j}]_x^2 & & & \\ & \sum_j [\hat{\mathbf{u}}_{2,j}]_x^2 & & \\ & & \ddots & \sum_j [\hat{\mathbf{u}}_{n,j}]_x^2 \\ & & & \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \dots & [\hat{\mathbf{u}}_{1,n}]_x^2 \\ \vdots & \ddots & \vdots \\ [\hat{\mathbf{u}}_{n,1}]_x^2 & \dots & \mathbf{0} \end{bmatrix} \quad (35)$$

The matrix H resembles the definition of a Laplacian where the edge labels are the matrices on the diagonal blocks of \widehat{S}^2 .

This formulation has the same structure of a Laplacian eigenmap for distance-based localization, where the measures are the matrices $[\hat{\mathbf{u}}_{i,j}]_x^2$ instead of distances.

Part II: applications

5 Time synchronization.

In a wireless network, nodes must often act in coordinated or synchronized fashion. This requires global clock synchronization, wherein all nodes in the network are synchronized to a common clock.

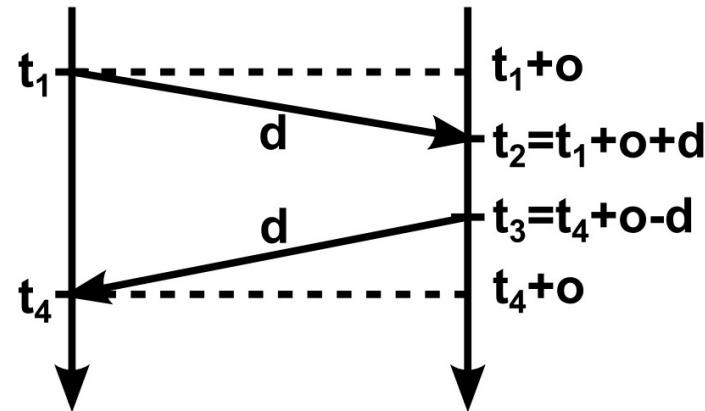
The network is modeled as a directed graph of $n + 1$ nodes $\{0, 1, 2, \dots, n\}$, where each edge represents the ability to transmit and receive packets between the corresponding pair of nodes.

Each node i has a clock $c_i(t)$, where t represents the reference time variable of node 0, which is the designated reference node, and $c_i(\cdot)$ is some unknown function. A simple first order model of such a function would be

$$c_i(t) = t + o_i.$$

The **clock synchronization problem** would then consist of finding the offset o_i of each node i from the reference node 0.

Estimates of clock differences between pair of nodes connected by an edge can be obtained by exchanging of time-stamped packets



The differences between the time of transmission and the time of reception is the sum of the propagation delay d , the clock offsets of the respective clocks, and random effects (eliminated by averaging).

The effect of propagation delay can be eliminated by subtracting the one-way difference from the reverse-way difference.

$$o = (t_2 - t_1 + t_3 - t_4)/2$$

At the end, we are able to obtain for each edge ij in the network, an estimate o_{ij} of the true clock offset.

These estimates must then be processed by the network to obtain estimates of the clock offsets of all nodes from the common reference clock od node 0.

This is a simple synchronization in $(\mathbb{R}^d, +)$ where $x_i = o_i$ and $z_{ij} = o_{ij}$.

Alternately, one could employ a two parameter model for the node clock:

$$c_i(t) = d_i t + o_i$$

where the skew parameter d_i represent the clock drift relative to node 0.

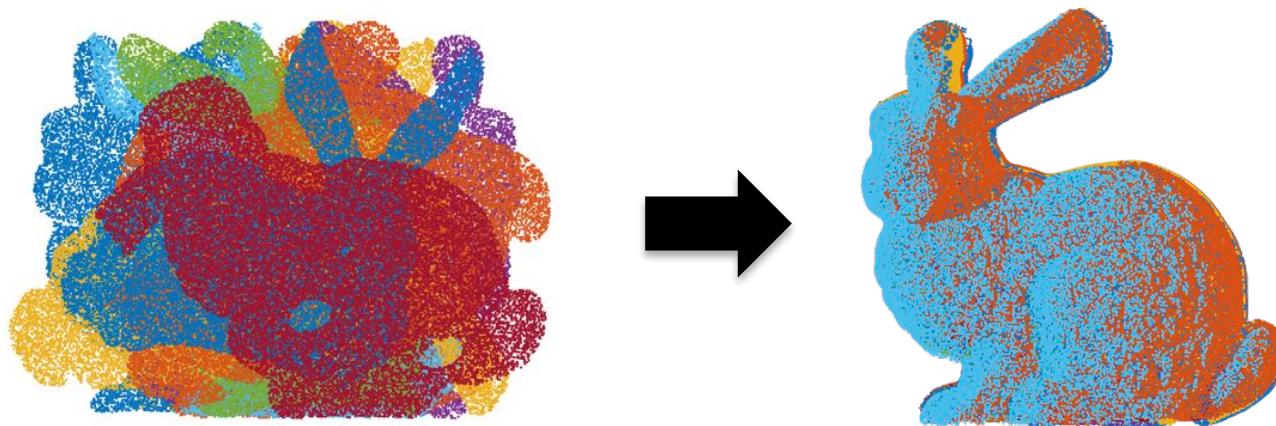
This is an istance of syncronization in the group of 1-d affinities, a subgroup of $GL(2)$.

In reality, these parameters will be time-varying. Thus, it is necessary to recompute their values periodically in order to maintain a bounded estimation error.

6 Point sets registration

Global registration (of 3D models) (a.k.a. *n*-view point set registration problem) consists in finding the rigid transformation that brings multiple ($n > 2$) 3-D point sets into alignment.

Each rigid transformation is an element of $SE(3)$.



Global registration can be solved in **point space** or in **frame space**.

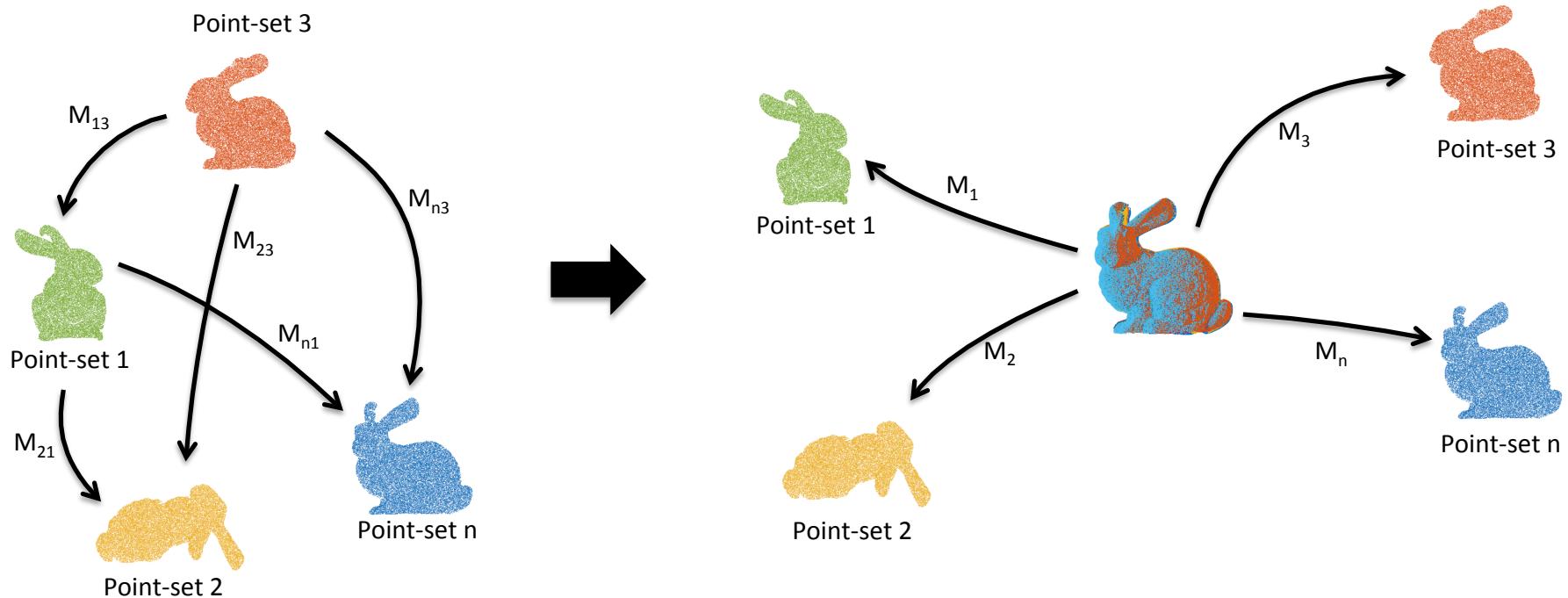
In **point space** all the transformations are simultaneously optimized with respect to a cost function that depends on the distance of corresponding points.

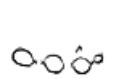
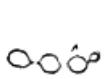
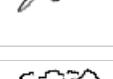
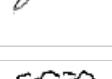
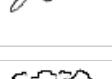
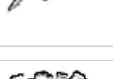
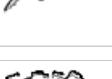
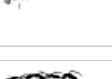
In **frame space** the optimization criterion is related to the internal coherence of the network of transformations applied to the local coordinate frames.

Ref. (Arrigoni et al., 2016c,b).

1. Registration between pairs of views is performed in isolation via the Iterative Closest Point Algorithm (ICP).
2. All the rigid transformations are globally optimized without using points, by solving a synchronization over $\text{SE}(3)$.

$$Z_{ij} = X_i^{-1} X_j$$



Dataset	R-GoDEC	GRASTA	L1-ALM	Govindu	DIFFUSION	Sharp et al.	EIG-SE(3)	EIG-SE(3)-IRLS	Rosen et al.
Bunny n=10									
Dragon n=15									
Sheep n=20									
Kitten n=24									
Frog n=24									
Gargoyle n=27									
Capital n=100									
									

Stanford 3D Scanning Repository <http://graphics.stanford.edu/data/3Dscanrep/>

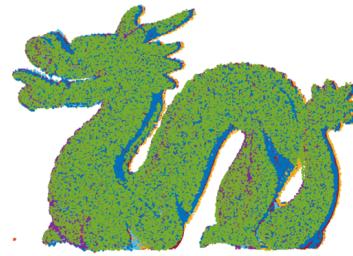
Visionair Shape Repository <http://visionair.ge.imati.cnr.it/ontologies/shapes/>



Bunny



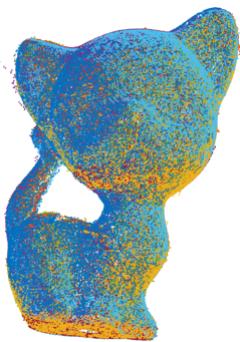
Buddha



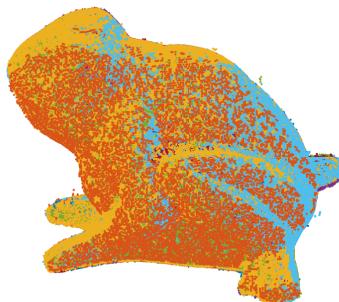
Dragon



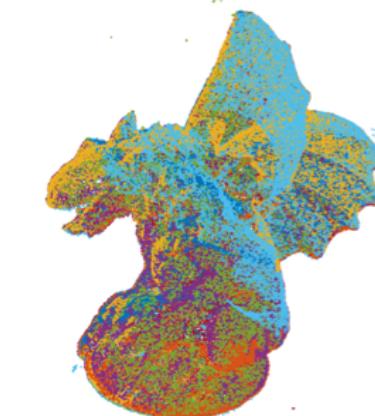
Sheep



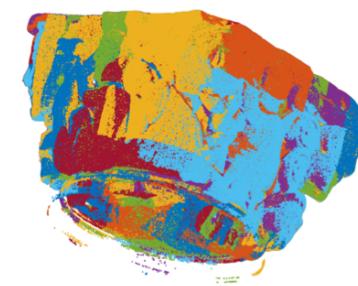
Kitten



Frog



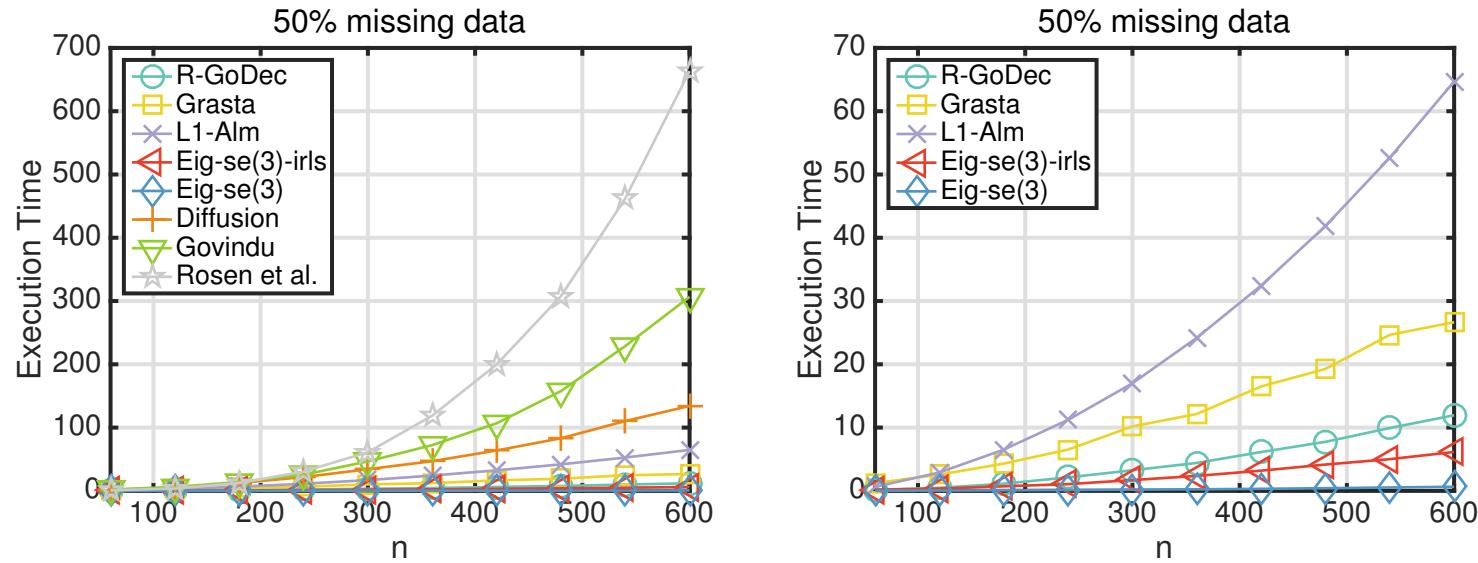
Gargoyle



Capital

Stanford 3D Scanning Repository <http://graphics.stanford.edu/data/3Dscanrep/>

Visionair Shape Repository <http://visionair.ge.imati.cnr.it/ontologies/shapes/>



S/W analysis:

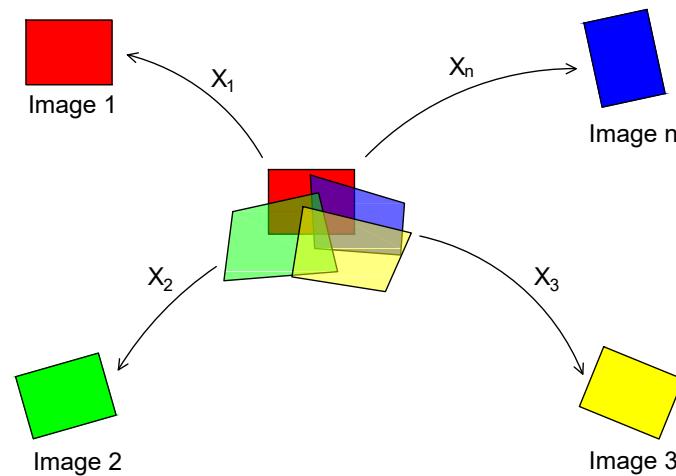
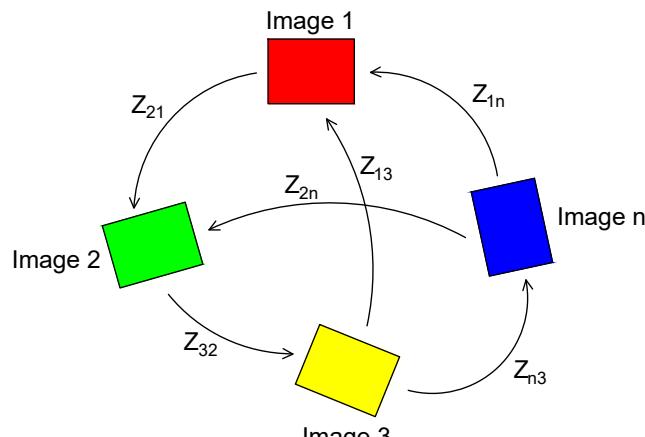
- ✓ This approach is extremely fast, as the synchronization problem is cast to a (partial) eigenvalue decomposition.
- ✓ It easily copes with weights on individual relative motions, allowing a straightforward extension to Iteratively Reweighted Least Squares (IRLS).
- ✗ The relaxation brings out of $SE(3)$, hence the method is suboptimal as compared to working **in** the manifold.

7 Mosaicing

Find the linear transformation of the projective plane (homography) that brings multiple ($n > 2$) images into alignment.

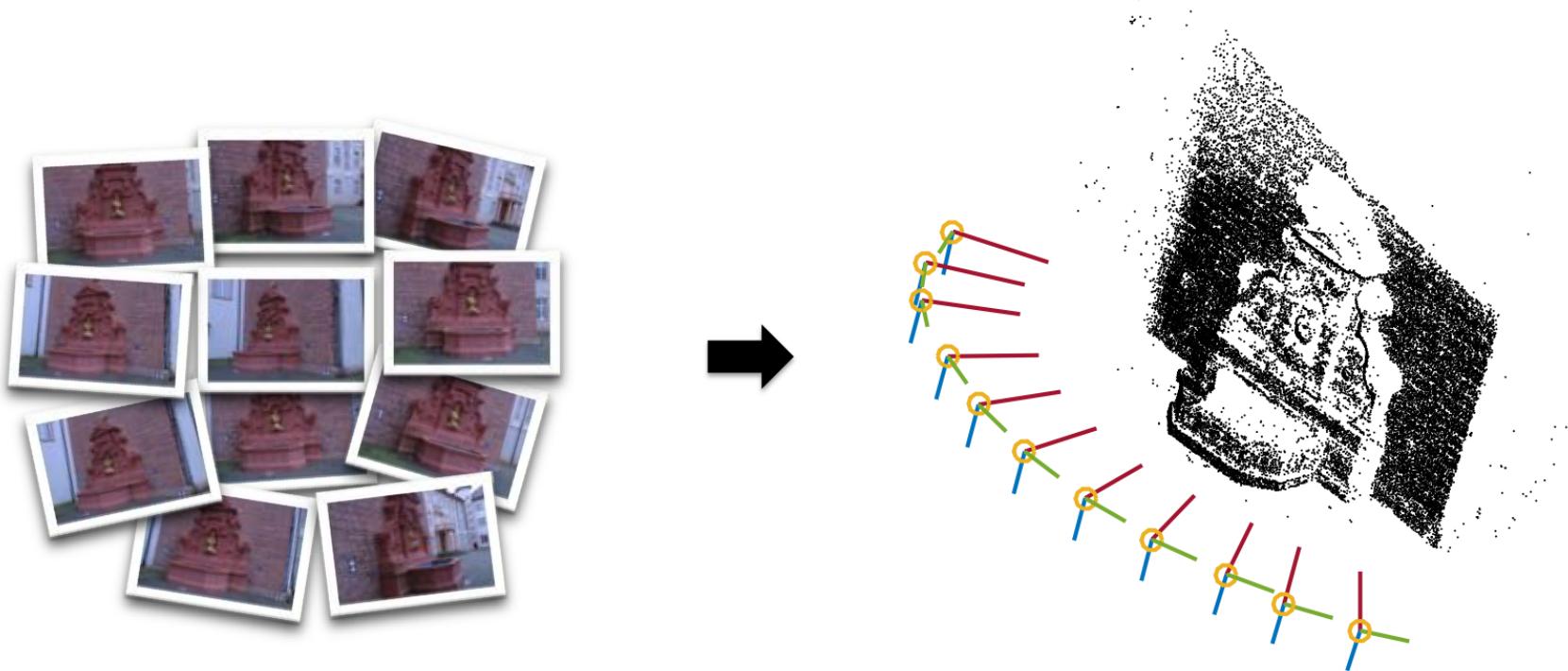
Each transformation is an element of $SL(3)$ (scale does not matter, so we force $\det = 1$)

Similarly to point set registration, solving in frame space is an instance of synchronization on $SL(3)$.



8 Structure-from-motion

The goal is to compute both scene structure (3D coordinate of scene points) and camera motion (angular attitudes and positions of the cameras), starting from a set of images.



Camera motion is represented as an element of $SE(3)$.

1. Starting from known interior parameters and matching points, the epipolar geometry is computed.
2. The angular attitudes/positions of the cameras are estimated with respect to an external reference system.

$$M_{ij} = M_i^{-1} M_j$$

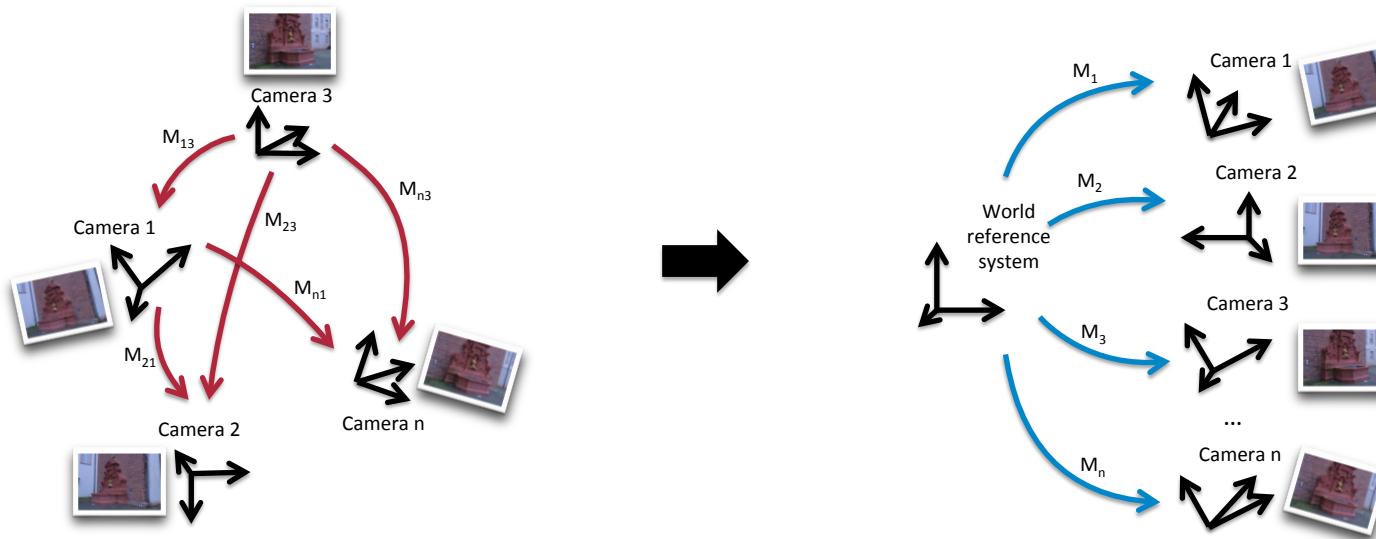
3. The structure is computed (via triangulation).
4. Bundle adjustment is applied at the end.

Synchronization brings from relative to absolute orientations (up to an arbitrary transformation, as in a free-network BA)

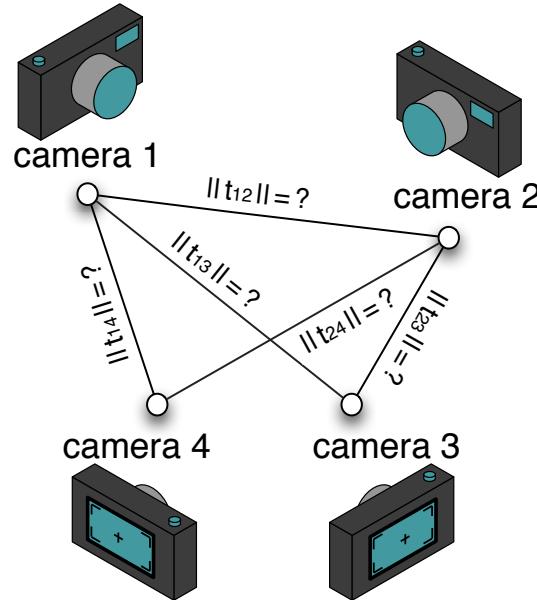
The underlying graph $G = (V, E)$ that represent the camera network is referred to as the **epipolar graph**:

- vertices correspond to cameras/images
- edges correspond to pairs of cameras sharing a sufficient number of tie-points.

The unknown vertex labels represent **absolute orientations** of cameras, while edge labels represent (measured) **relative orientations**.

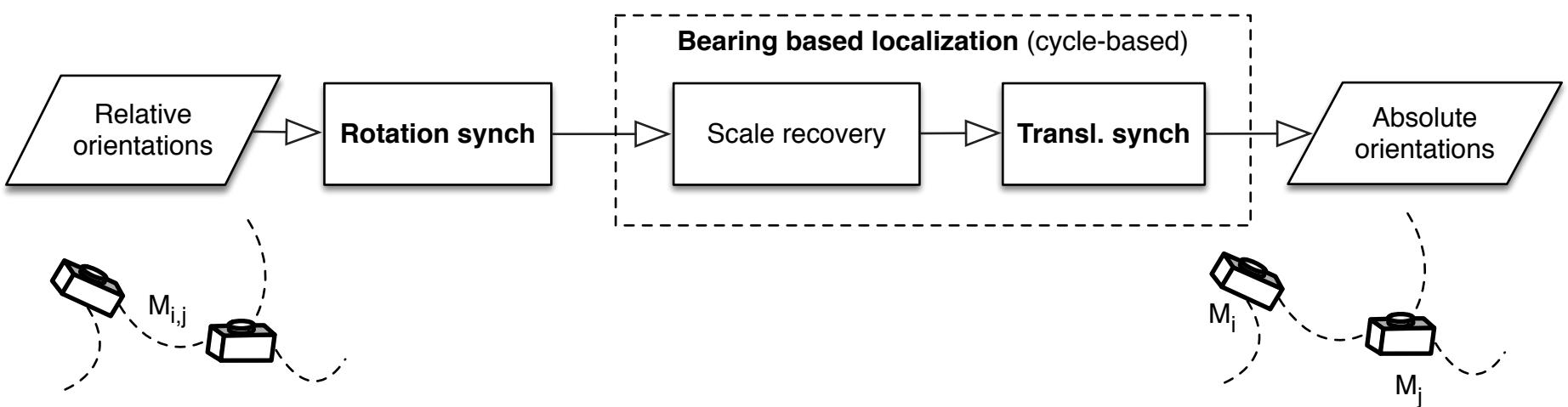
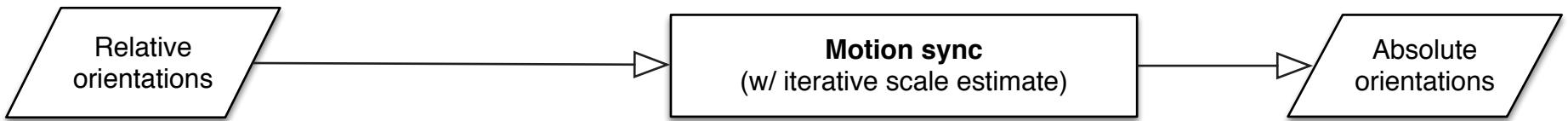


Problem: the magnitudes of pairwise translations (epipolar scales) are unknown: only directions (bearings) can be computed.



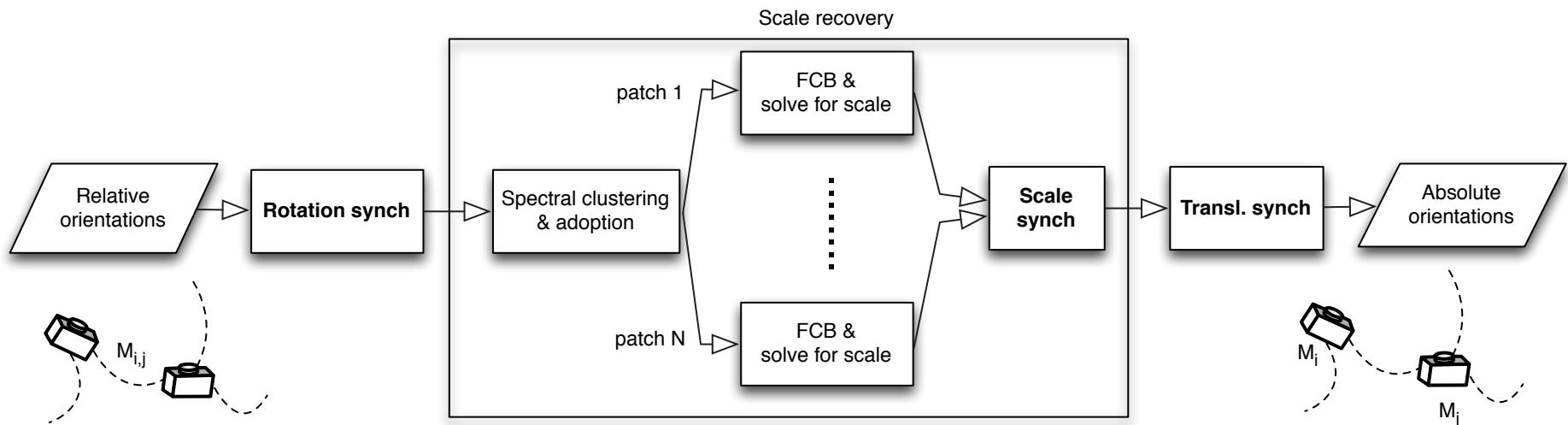
The motion stage of structure-from-motion (a.k.a. camera network orientation) can not be straightforwardly solved as a motion synchronization.

We need to explicitly compute epipolar scales or solve directly from bearings.



In the cycle-based approach the number of unknowns for scale recovery is equal to the number of edges, which is of the order $O(n^2)$, if the graph is not sparse.

1. Rotation synchronization to obtain the angular attitudes of the cameras.
2. The scales are recovered via a divide-et-impera approach.
3. Synchronization over \mathbb{R}^3 is run at the end to compute the position of the cameras.



Ref. Arrigoni et al. (2016a)

Translation Errors [meters] and Execution Times [seconds]

Dataset	miss %	GSP			1DSfM			LUD			Cui et al.			ShapeKick		
		n	tra.	time	n	tra.	time	n	tra.	time	n	tra.	time	n	tra.	time
Vienna Cathedral	74	684	2.8	69	836	6.6	302	836	5.4	787	578	3.5	242	836	1.9	156
Alamo	47	499	0.6	40	577	1.1	158	577	0.4	385	500	0.6	259	577	0.9	68
Notre Dame	32	530	1.5	27	553	10	154	553	0.3	707	539	0.3	366	553	0.2	68
Tower of London	80	408	1.6	10	572	11	78	572	4.7	88	393	4.4	100	472	2.3	24
Montreal Notre Dame	52	423	0.4	14	450	2.5	114	450	0.5	271	426	0.8	125	450	0.8	32
Yorkminster	72	386	1.4	10	437	3.4	122	437	2.7	103	341	3.7	45	—	—	—
Madrid Metropolis	65	268	7.5	7	341	9.9	43	341	1.6	67	—	—	—	341	6.0	19
NYC Library	68	295	1.1	8	332	2.5	76	332	2.0	102	288	1.4	42	332	1.4	18
Piazza del Popolo	58	297	1.0	14	328	3.1	58	328	1.5	88	294	2.6	51	338	3.6	17

1DSfM datasets <http://www.cs.cornell.edu/projects/1dsfm/>

S/W analysis:

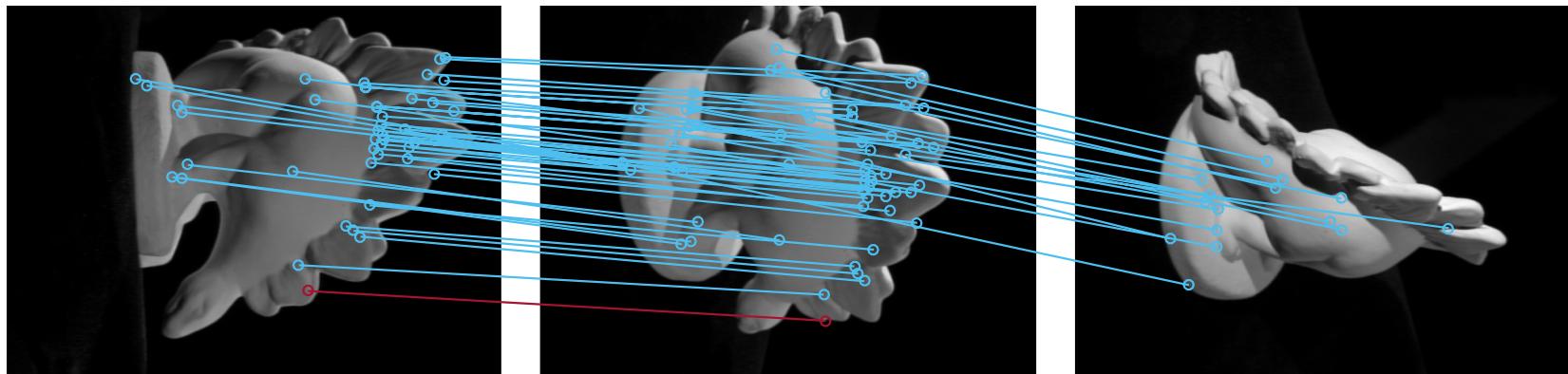
- ✓ These methods are usually faster than others, while ensuring a fair distribution of the errors among the cameras, being global.
- ✗ Accuracy is suboptimal, worse than that achieved by bundle adjustment/point-space methods.

These global methods can be seen as an effective and efficient way of computing approximate orientations to be subsequently refined by bundle adjustment.

9 Multi-view Matching

The goal is to establish correspondences between multiple feature sets.

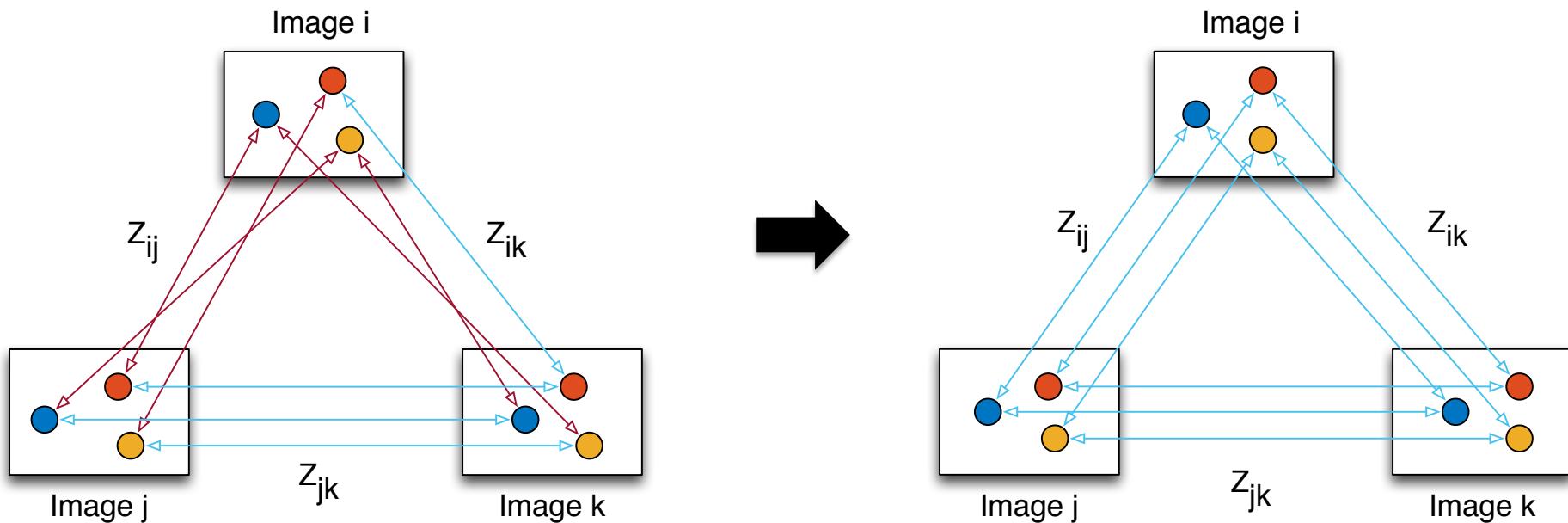
Each matching is represented as a partial permutation matrix.



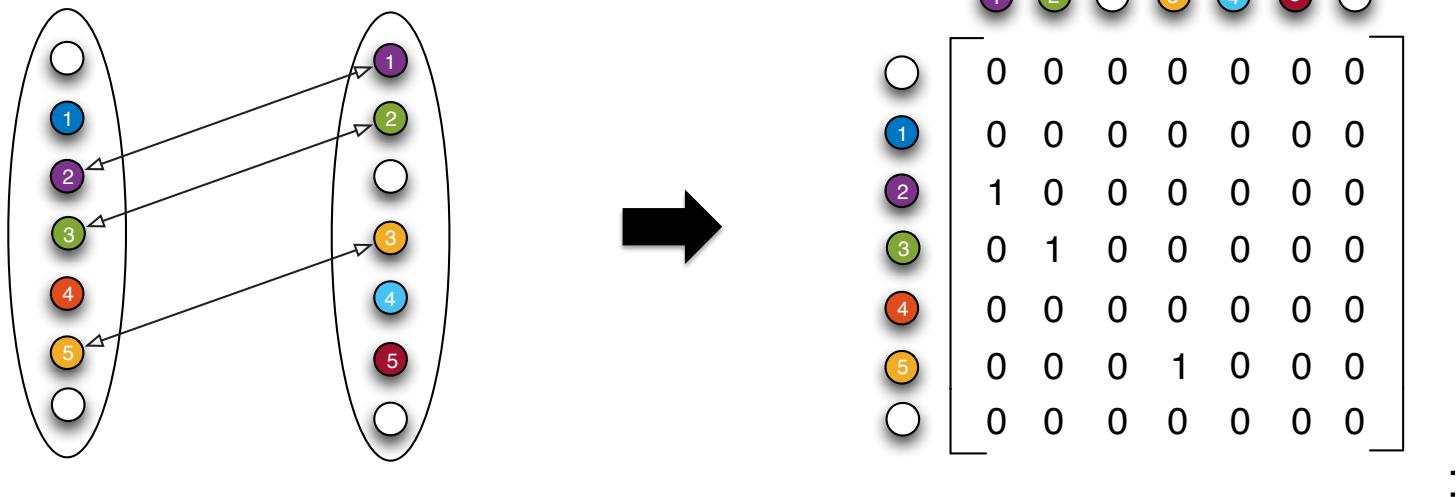
- One-step matching: the cost function depends on the features extracted in all the images.
- Two-step matching: the optimization criterion is related to the internal coherence of permutation matrices.

1. Matching between pairs of images is performed in isolation (e.g. using SIFT).
2. Pairwise correspondences are improved by globally optimizing their internal coherence, without relying on the value of the features.

$$Z_{ij} = X_i^T X_j$$



Matches – however – are best represented by **partial permutations**

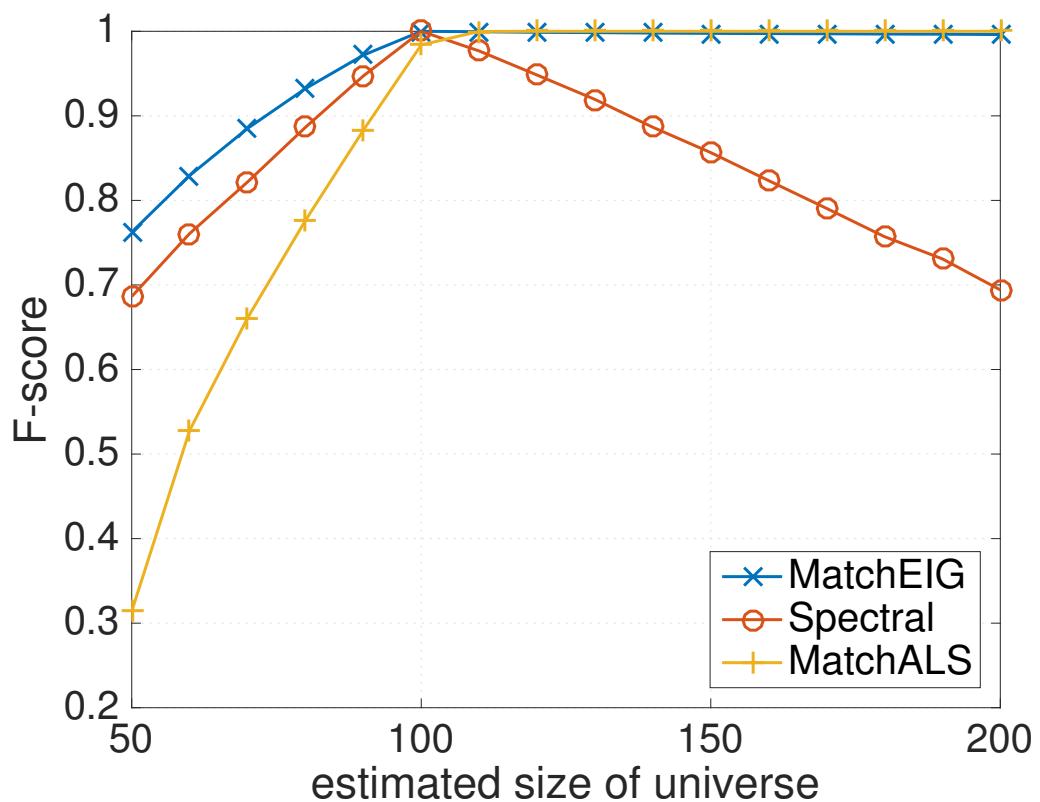


Synchronization is similar to the case of total permutations, with a few caveats. See (Arrigoni et al., 2017) for details.

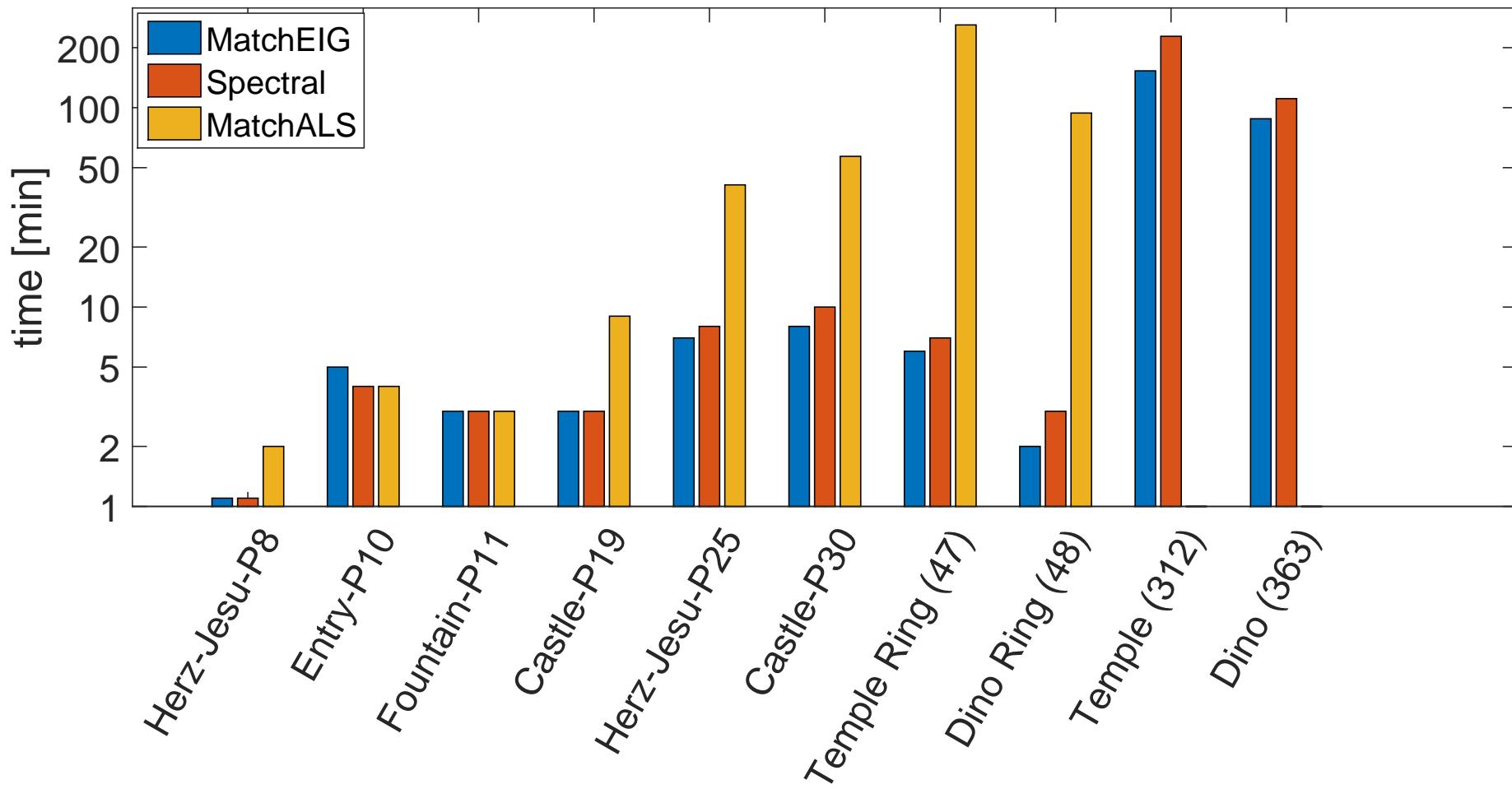
This approach however requires the exact size of the feature's universe (d).

In (Maset et al., 2017) we proposed a variation that is insensitive to d , provided it is an overestimate.

F-score versus Estimated Size of Universe

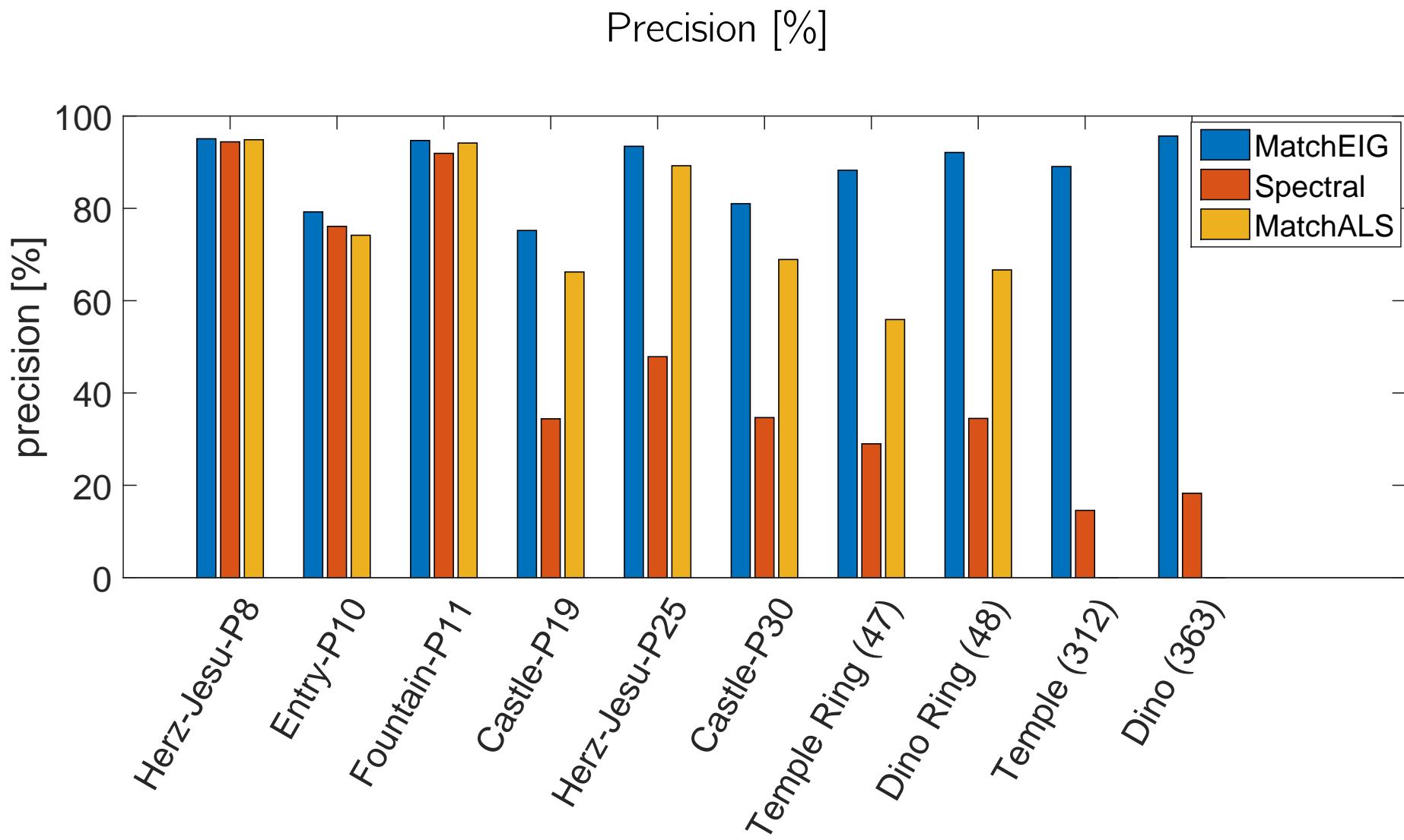


Execution Times [minutes]



EPFL multi-view stereo dataset <https://icwww.epfl.ch/~marquez/multiview/denseMVS.html>

Middlebury multi-view stereo dataset <http://vision.middlebury.edu/mview/data/>



EPFL multi-view stereo dataset <https://icwww.epfl.ch/~marquez/multiview/denseMVS.html>

Middlebury multi-view stereo dataset <http://vision.middlebury.edu/mview/data/>

References

- Arie-Nachimson, M., Kovalsky, S. Z., Kemelmacher-Shlizerman, I., Singer, A., Basri, R., 2012. Global motion estimation from point matches. Proceedings of the Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission.
- Arrigoni, F., Fusiello, A., Rossi, B., 2015a. On computing the translations norm in the epipolar graph. In: Proceedings of the International Conference on 3D Vision (3DV). pp. 300–308.
- Arrigoni, F., Fusiello, A., Rossi, B., 2016a. Camera motion from group synchronization. In: Proceedings of the International Conference on 3D Vision (3DV). pp. 546–555.
- Arrigoni, F., Magri, L., Rossi, B., Fragneto, P., Fusiello, A., 2014a. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In: Proceedings of the International Conference on 3D Vision (3DV). pp. 491–498.
- Arrigoni, F., Maset, E., Fusiello, A., 2017. Synchronization in the symmetric inverse semigroup. In: Proceedings of the International Conference on Image Analysis and Processing.
- Arrigoni, F., Rossi, B., Fusiello, A., 2015b. Robust and efficient camera motion synchronization via matrix decomposition. In: Proceedings of the International Conference on Image Analysis and Processing. Vol. 9279 of Lecture Notes in Computer Science. Springer International Publishing, pp. 444–455.
- Arrigoni, F., Rossi, B., Fusiello, A., 2016b. Global registration of 3D point sets via LRS decomposition. In: Proceedings of the 14th European Conference on Computer Vision. pp. 489–504.
- Arrigoni, F., Rossi, B., Fusiello, A., 2016c. Spectral synchronization of multiple views in $SE(3)$. SIAM Journal on Imaging Sciences 9 (4), 1963 – 1990.

- Arrigoni, F., Rossi, B., Malapelle, F., Fragneto, P., Fusiello, A., 2014b. Robust global motion estimation with matrix completion. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-5, 63–70.
- Barooah, P., Hespanha, J. P., 2007. Estimation on graphs from relative measurements. *IEEE Control Systems* 27 (4), 57 – 74.
- Chatterjee, A., Govindu, V. M., 2013. Efficient and robust large-scale rotation averaging. In: *Proceedings of the International Conference on Computer Vision*.
- Cucuringu, M., 2015. Synchronization over Z_2 and community detection in signed multiplex networks with constraints. *Journal of Complex Networks* 3 (3), 469–506.
- Cygan, M., Pilipczuk, M., Pilipczuk, M., 2012. 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.
- Enqvist, O., Kahl, F., Olsson, C., 2011. Non-sequential structure from motion. In: *Eleventh Workshop on Omnidirectional Vision, Camera Networks and Non-classical Camera*.
- Fredriksson, J., Olsson, C., 2012. Simultaneous multiple rotation averaging using Lagrangian duality. In: *Proceedings of the Asian Conference on Computer Vision*.
- Fusiello, A., Castellani, U., Ronchetti, L., Murino, V., 2002. Model acquisition by registration of multiple acoustic range views. In: *Proceedings of the European Conference on Computer Vision*. pp. 805–819.
- Giridhar, A., Kumar, P., 2006. Distributed clock synchronization over wireless networks: Algorithms and analysis. *Proceedings of the IEEE Conference on Decision and Control*, 4915–4920.
- Govindu, V. M., 2004. Lie-algebraic averaging for globally consistent motion estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 684–691.

- Govindu, V. M., 2006. Robustness in motion averaging. In: Proceedings of the Asian Conference on Computer Vision. pp. 457–466.
- Hartley, R., Trumpf, J., Dai, Y., Li, H., 2013. Rotation averaging. International Journal of Computer Vision.
- Martinec, D., Pajdla, T., 2007. Robust rotation and translation estimation in multiview reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Maset, E., Arrigoni, F., Fusiello, A., 2017. Practical and efficient multi-view matching. In: Proceedings of the International Conference on Computer Vision.
- Pachauri, D., Kondor, R., Singh, V., 2013. Solving the multi-way matching problem by permutation synchronization. In: Advances in Neural Information Processing Systems 26. pp. 1860–1868.
- Rosen, D., Carbone, L., Bandeira, A., Leonard, J., 2016. A certifiably correct algorithm for synchronization over the special Euclidean group. In: International Workshop on the Algorithmic Foundations of Robotics (WAFR).
- Russel, W., Klein, D., Hespanha, J., 2011. Optimal estimation on the graph cycle space. IEEE Transactions on Signal Processing 59 (6), 2834 – 2846.
- Schroeder, P., Bartoli, A., Georgel, P., Navab, N., Jan 2011. Closed-form solutions to multiple-view homography estimation. In: Applications of Computer Vision (WACV), 2011 IEEE Workshop on. pp. 650–657.
- Sharp, G. C., Lee, S. W., Wehe, D. K., 2002. Multiview registration of 3D scenes by minimizing error between coordinate frames. In: Proceedings of the European Conference on Computer Vision. pp. 587–597.
- Shen, Y., Huang, Q., Srebro, N., Sanghavi, S., 2016. Normalized spectral map synchronization. In: Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 29. Curran Associates, Inc., pp. 4925–4933.
- Singer, A., 2011. Angular synchronization by eigenvectors and semidefinite programming. Applied and Computational Harmonic Analysis 30 (1), 20 – 36.

- Torsello, A., Rodolà, E., Albarelli, A., 2011. Multiview registration via graph diffusion of dual quaternions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2441 – 2448.
- Tron, R., Carlone, L., Dellaert, F., Daniilidis, K., 2015. Rigid components identification and rigidity enforcement in bearing-only localization using the graph cycle basis. In: IEEE American Control Conference.
- Tron, R., Daniilidis, K., 2014. Statistical pose averaging with varying and non-isotropic covariances. In: Proceedings of the European Conference on Computer Vision.
- Tron, R., Zhou, X., Daniilidis, K., 2016. A survey on rotation optimization in structure from motion. In: Computer Vision and Pattern Recognition Workshops (CVPRW).
- Wahlström, M., 2014. Half-integrality, LP-branching and FPT algorithms, 1762–1781.
- Whiteley, W., 1997. Matroids from discrete geometry. In: Matroid Theory. AMS Contemporary Mathematics. American Mathematical Society, pp. 171–313.
- Wilson, K., Bindel, D., Snavely, N., 2016. When is rotations averaging hard? In: Proceedings of the European Conference on Computer Vision. pp. 255 – 270.
- Yu, J.-G., Xia, G.-S., Samal, A., Tian, J., 2016. Globally consistent correspondence of multiple feature sets using proximal Gauss–Seidel relaxation. Pattern Recognition 51, 255 – 267.
- Zach, C., Klopschitz, M., Pollefeys, M., 2010. Disambiguating visual relations using loop constraints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1426 – 1433.

10 Graphs basics

Let $G = (V, E)$ a finite simple *undirected* graph with n nodes and m vertices. The **adjacency matrix** of G is defined as the $n \times n$ matrix $A(G)$ in which:

$$A(G)_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases}$$

The **incidence matrix** of a finite simple *directed* graph $\vec{G} = (V, E)$ with n nodes and m edges is defined as:

$$B(\vec{G})_{ij} = \begin{cases} 1, & \text{if } i \text{ is the head of } e_j \\ -1, & \text{if } i \text{ is the tail of } e_j \\ 0, & \text{otherwise} \end{cases}$$

The rows of the incidence matrix correspond to vertices of G and its columns to edges of G .

The **degree matrix** of the graph is the diagonal matrix defined as:

$$D(G)_{ij} = \begin{cases} \deg(v_i) = \sum_j A(G)_{i,j}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

or, equivalently: $D = \text{diag}(A\mathbf{1})$.

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.

Viewing cycles as vectors indexed by edges, addition of cycles corresponds to modulo-2 sum of vectors, and the cycles of a graph form a vector space in \mathbb{Z}_2^m .

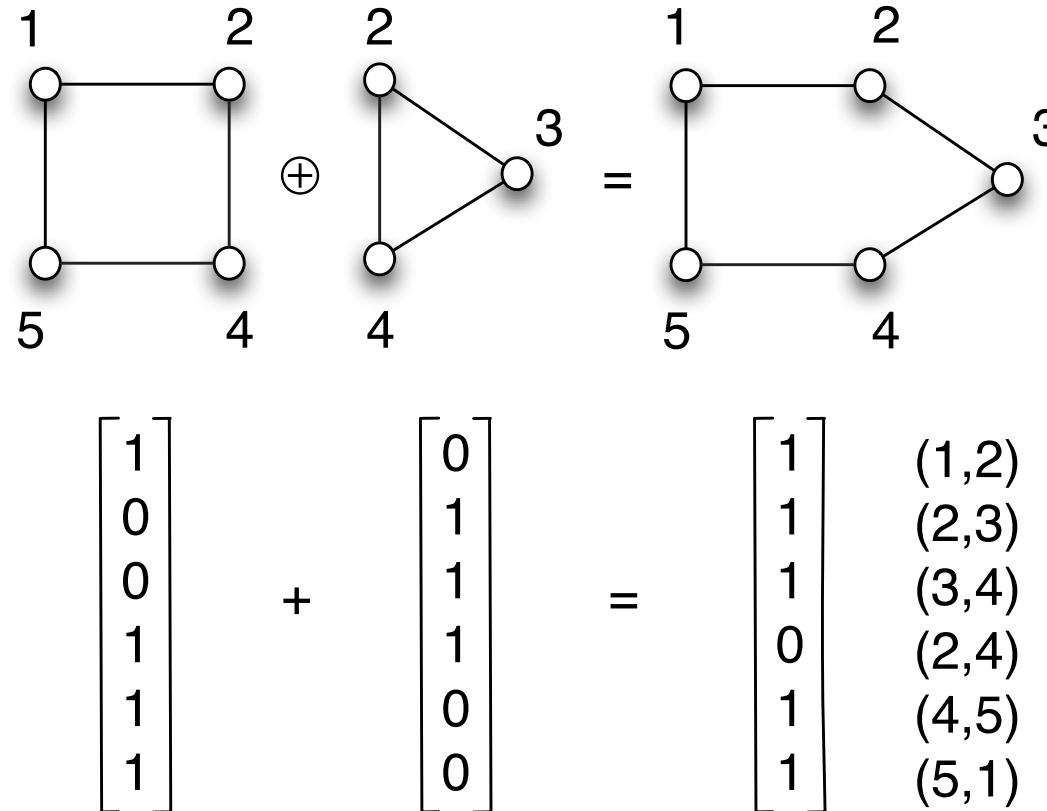
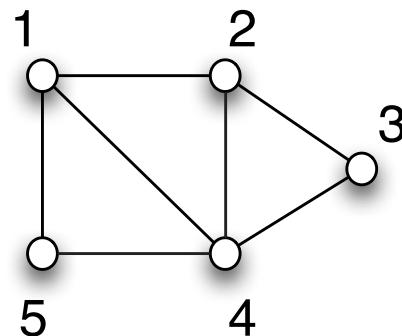


Fig. 4: The sum of two cycles is a cycle where the common edges vanish.

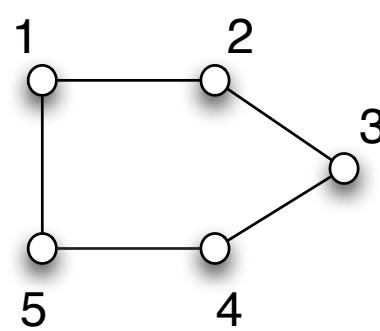
A **cycle basis** is a minimal set of **circuits** such that any cycle can be written as linear combination of the circuits in the basis.

If we stack the indicator vectors of the circuits of a basis in a matrix C (by rows) we obtain the **cycle basis matrix**.

The dimension of the cycle space is $m - n + c$, where c denotes the number of connected components in $G = (V, E)$.



(a) Graph $G = (V, E)$.



(b) Cycle Basis.

Fig. 5: Example of a cycle basis associated to a given graph $G = (V, E)$. In general, a cycle basis is not unique.

It can be proven that $CB^T = 0$.

11 Kronecker product

Let A be a $m \times n$ matrix and B a $p \times q$ matrix. The *Kronecker product* of A and B is the $mp \times nq$ matrix defined by

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}. \quad (36)$$

The Kronecker product is defined for any pair of matrices A and B . It is associative and distributive with respect to matrix sum and product, but it is not commutative. The transpose of a Kronecker product is $(A \otimes B)^T = A^T \otimes B^T$.

A very important property concerns the eigenvalues of the Kronecker product: the eigenvalues of $A \otimes B$ are the outer product of the eigenvalues of A and B . This implies that:

$$\text{rank}(A \otimes B) = \text{rank}(A) \text{rank}(B). \quad (37)$$

Vectorization

The *vectorization* of a matrix is a linear transformation which converts the matrix into a column vector. Specifically, the vectorization of the matrix A , denoted by $\text{vec}(A)$, is the vector obtained by stacking the columns of A one underneath the other.

The basic connection between the vec operator and the Kronecker product is

$$\text{vec}(\mathbf{a}\mathbf{b}^T) = \mathbf{b} \otimes \mathbf{a} \quad (38)$$

for any column vectors \mathbf{a} and \mathbf{b} . The generalization of this is the following important property:

$$\text{vec}(AXB) = (B^T \otimes A) \text{vec}(X) \quad (39)$$

for matrices A, B, X of compatible dimensions.

The *half-vectorization*, $\text{vech}(A)$, of a symmetric $n \times n$ matrix A is the $n(n+1)/2 \times 1$ column vector obtained by vectorizing only the lower triangular part of A .

The *duplication matrix* D_n is the unique $n^2 \times n(n+1)/2$ matrix which, transforms $\text{vech}(A)$ into $\text{vec}(A)$: $D_n \text{vech}(A) = \text{vec}(A)$.

12 Khatri-Rao product

The Khatri-Rao product (?), denoted by \odot , is in some sense a partitioned Kronecker product, where by default the column-wise partitioning is considered.

Let us consider two matrices A of order $p \times r$ and B of order $q \times r$ and denote the columns of A by $\mathbf{a}_1 \cdots \mathbf{a}_r$ and those of B by $\mathbf{b}_1 \cdots \mathbf{b}_r$. The Khatri-Rao product is defined to be the partitioned matrix of order $pq \times r$:

$$A \odot B = [\mathbf{a}_1 \otimes \mathbf{b}_1, \cdots, \mathbf{a}_r \otimes \mathbf{b}_r] \quad (40)$$

where \otimes denotes the Kronecker product.

If X is diagonal, then

$$\text{vec}(AXB) = (B^T \odot A) \text{diag}^{-1}(X) \quad (41)$$

where diag^{-1} returns a vector containing the diagonal elements of its argument.

With $B = I$ one obtains

$$\text{vec}(AX) = (I \odot A) \text{diag}^{-1}(X). \quad (42)$$

It it is easy to see that

$$(I \odot A) = \text{blkdiag}(\mathbf{a}_1 \dots \mathbf{a}_n) \quad (43)$$

where $\mathbf{a}_1 \dots \mathbf{a}_n$ are the columns of A and blkdiag is the operator that construct a block diagonal matrix with its arguments as blocks.

13 Orthogonal Procrustes Problem

Given $A, B \in \mathbb{R}^{m \times n}$ find the orthogonal transformation Q that best align B onto A , i.e., solve

$$\min\{\|A - BQ\|_F : QQ^T = I_n\} \quad (44)$$

This is equivalent to maximize $\text{tr}(Q^T B^T A)$. Let us consider the SVD of $B^T A$:

$$\text{tr}(Q^T B^T A) = \text{tr}(Q^T UDV^T) = \text{tr}(V^T Q^T U D) \leq \sum_i \sigma_i \quad (45)$$

Equality is achieved for $Q = UV^T$

Setting $B = I$ in (44), the solution Q is the closest orthogonal matrix to A .

If $\det(Q) = 1$ is required in addition (rotation matrix), the solution is

$$Q = U \text{diag}(1, 1, \dots, \det(V^T U)) V^T$$

14 Permutation Procrustes Problem

Given $A, B \in \mathbb{R}^{m \times n}$ find the permutation Q that best align B onto A , i.e., solve

$$\min\{\|A - BQ\|_F : Q \text{ a permutation matrix}\} \quad (46)$$

This is equivalent to maximize $\text{tr}(PC)$ with $P = Q^T$ and $C = B^T A$. Permutation matrices are special cases of double stochastic one, so the relaxed problem writes

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n \sum_{j=1}^n p_{ij} c_{ij} \\ & \text{subject to} && p_{ij} \geq 0 \\ & && \sum_{i=1}^n p_{ij} = 1, \quad \sum_{j=1}^n p_{ij} = 1. \end{aligned} \quad (47)$$

This is a linear programming problem, hence its solution must occur at a vertex of the feasible region, which correspond to a permutation matrix.

This is in fact an assignment problem, which can be solved by the Hungarian method ($-c_{ij}$ is the cost of assigning agent i to task j , and p_{ij} represent the assignment).

Setting $B = I$ in (46), the solution Q is the closest permutation matrix to A .

In order to cater for partial permutation the problem is relaxed to doubly *substochastic* matrices, namely the constraint is changed to $\sum_{i=1}^n p_{ij} \leq 1$ (and likewise for the columns).

Moreover, a soft thresholding is applied to c_{ij} with a small threshold, otherwise even a negligible c_{ij} would justify a $p_{ij} = 1$, which would compel the solution to be always a total permutation.