Synchronization problems

in Computer Vision

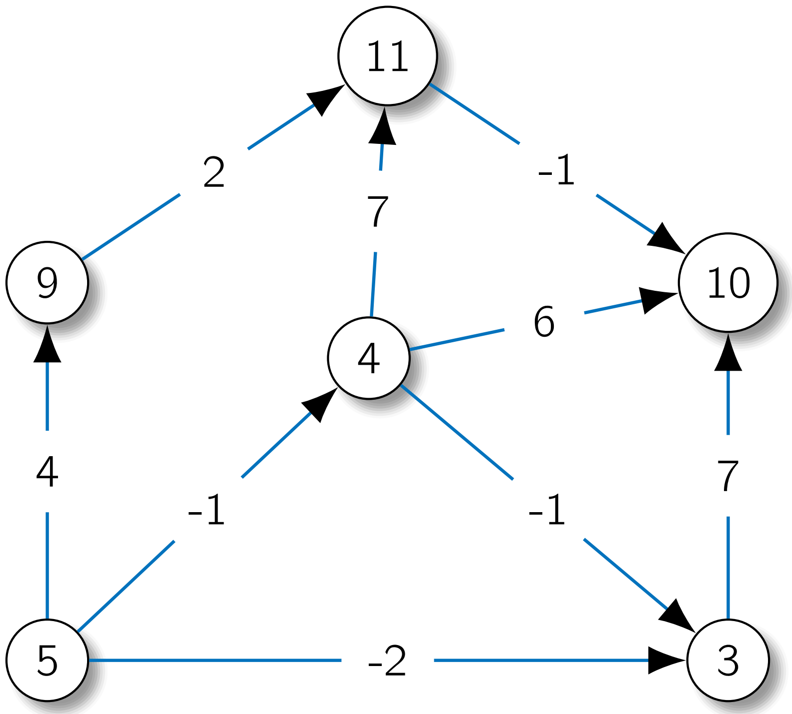
Andrea Fusiello*
University of Udine

*joint work with Federica Arrigoni

Introduction

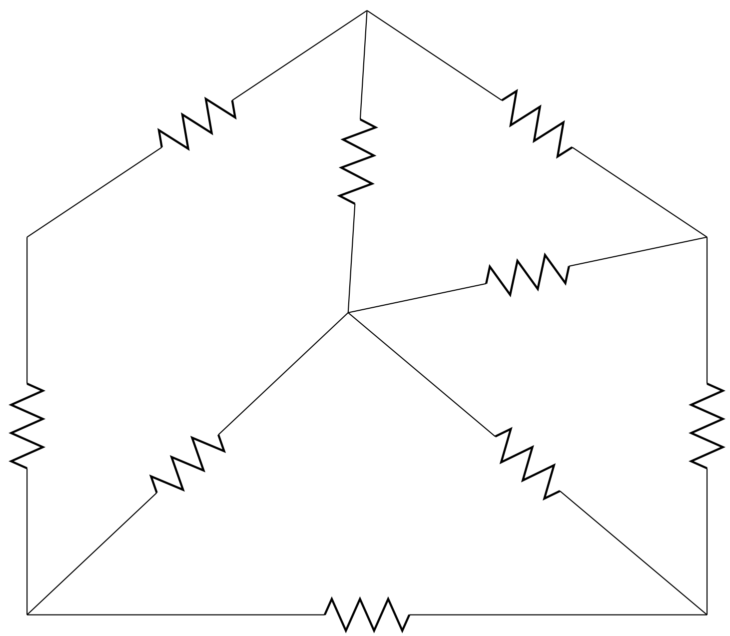
In a network of nodes, each node has an unknown state and measures of differences (or ratios) of states are available.

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 set of measures 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 istance of the synchronization problem. In general, states can be elements of any group, possibly with noisy or wrong measures.

In Computer Vision the state is the origin and/or attitude of a local reference frame (e.g., attached to a camera).

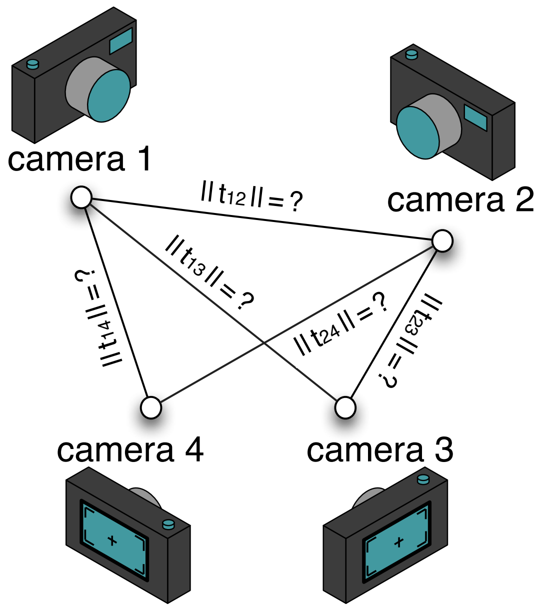
Starting from known interior orientation and tie-points compute epipolar geometry which results in relative rotations and relative translations (up to a scale).

Synchronization brings from relative to absolute orientations.

Displacements are known only partially, as directions. Let us assume for the moment that the magnitude of translation is known, we deal with this at the end.

The underlying graph $G = (V, E)$ is referred to as the epipolar graph (in the following $n = |V|$ and $m = |E|$):

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



The unknown vertex labels M_i represent absolute orientations of cameras, while edge labels M_{ij} represent (measured) relative orientations.

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

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 rigid motion.

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

$$R_{ij} = R_i^T R_j \tag{2}$$

$$\mathbf{x}_{ij} = R_i^T \mathbf{x}_j - R_i^T \mathbf{x}_i \tag{3}$$

by considering separately the rotation and translation terms.

Matrix M_i is the inverse of the usual matrix G_i found in the definition of the perspective projection matrix: $P_i = [I|0]G_i$ (assuming normalized coordinates).

In order to have $M_{ij} = M_i^{-1} M_j$ encoded in E_{ij}, then the essential matrix must defined by $\mathbf{p}_i^T E_{ij} \mathbf{p}_j$. Then: $E_{ij} = [\mathbf{x}_{ij}]_\times R_{ij}$.

1 Rotation synchronization

It is also known as multiple rotation averaging (?).

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

In matrix form:

$$X = \begin{bmatrix} R_1^T \\ R_2^T \\ \dots \\ R_n^T \end{bmatrix}, \quad X^T = \begin{bmatrix} R_1, R_2, \dots R_n, \end{bmatrix}, \quad Z = \begin{bmatrix} I & R_{12} & \dots & R_{1n} \\ R_{21} & I & \dots & R_{2n} \\ \dots & & & \dots \\ R_{n1} & R_{n2} & \dots & I \end{bmatrix} \tag{4}$$

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

$$Z = XX^T. \tag{5}$$

The rank-3 matrix Z containing all the edge labels is symmetric and positive semidefinite.

Since $X^T X = nI$, the consistency can be rewriten as:

$$ZX = nX. \tag{6}$$

Hence, the 3 columns of X are the eigenvectors of Z corresponding to the 3 nonzero eigenvalues of Z.

This was for a complete graph; in general Z has zero blocks in correspondence of missing edges, and the solution X is recovered as the 3 top eigenvectors of

$$\boxed{(D \otimes I_3)^{-1} Z_A,} \tag{7}$$

where A is the adjacency matrix of the epipolar graph, $D = \text{diag}(A\mathbf{1})$ is the degree matrix, and Z_A (with zero blocks) contains the available measures.

At the end, each 3×3 block of X is projected onto $SO(3)$ through SVD.

2 Translation synchronization.

The consistency constraint for translations (3):

$$\mathbf{x}_{ij} = R_i^T \mathbf{x}_j - R_i^T \mathbf{x}_i \tag{8}$$

can be written equivalently as

$$R_i \mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i := \mathbf{u}_{ij} \tag{9}$$

where $\mathbf{x}_i$ is the centre of the i-th camera and $\mathbf{u}_{ij}$ is the baseline (available only after rotation synchronization)

Let us denote the incidence vector of the edge (i, j) with

$$\mathbf{b}_{ij} = (0, \ldots, \underset{\substack{\uparrow \\ i}}{-1}, \ldots, \underset{\substack{\uparrow \\ j}}{1}, \ldots, 0)^T \tag{10}$$

Equation (9) writes:

$$X \mathbf{b}_{ij} = \mathbf{u}_{ij} \tag{11}$$

where the columns of X are the centres $\mathbf{x}_i$.

Let B be the $n \times m$ incidence matrix of G, which has the $\mathbf{b}_{ij}$ as columns; it is easy to see that for all the edges the equation above writes

$$XB = U \tag{12}$$

where all the m baselines $\mathbf{u}_{ij}$ are juxtaposed in one $3 \times m$ matrix U.

Equivalently, using the Kronecker product:

$$\boxed{(B^T \otimes I_3)\operatorname{vec} X = \operatorname{vec} U.} \tag{13}$$

If we assume that the epipolar graph is connected, $\operatorname{rank}(B) = n - 1$. Since the solution is defined up to a global translation, we are allowed w.l.o.g. to arbitrarily set $\mathbf{x}_j = \mathbf{0}$. Removing $\mathbf{x}_j$ from the unknowns and the corresponding row in B leaves a full-rank $n - 1 \times m$ matrix B_j.

Is that all? No, the magnitude of translations are unknown.

3 Magnitude revovery

Node-based. Let us multiply the translation synchronization equation:

$$(B^T \otimes I_3)\,\mathrm{vec}\,X = \mathrm{vec}\,U \tag{14}$$

by the block diagonal matrix

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

yielding

$$\widehat{S}(B^T \otimes I_3)\,\mathrm{vec}\,X = \cancel{\widehat{S}\,\mathrm{vec}\,U} = 0 \tag{15}$$

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

This equation is also called the node-based bearing constraint in ?. Its solution yields the locations X, hence implicitly recovering the scales.

Egde-based. Let us start from the translation synchronization:

$$(B^T \otimes I_3)\,\mathrm{vec}\,X = \mathrm{vec}\,U \tag{16}$$

If the baselines $\mathbf{u}_{ij}$ are expanded into magnitude α_{ij} and direction $\hat{\mathbf{u}}_{ij}$: $\mathbf{u}_{ij} = \alpha_{ij}\hat{\mathbf{u}}_{ij}$, the matrix U writes:

$$U = \boldsymbol{\alpha}^T \odot \widehat{U} \tag{17}$$

where $\widehat{U}$ contains the baseline directions (or bearings) in columns, $\boldsymbol{\alpha}$ is a vector containing the magnitudes and $\odot$ denotes the Khatri-Rao product. Therefore:

$$(B^T \otimes I_3)\,\mathrm{vec}\,X = \mathrm{vec}(\boldsymbol{\alpha}^T \odot \widehat{U}) = (I \odot \widehat{U})\boldsymbol{\alpha} \tag{18}$$

Let us consider a cycle basis matrix C and multiply left and right by $(C \otimes I_3)$:

$$(C \otimes I_3)(B^T \otimes I_3)\,\mathrm{vec}\,X = (C \otimes I_3)(I \odot \widehat{U})\boldsymbol{\alpha} \tag{19}$$

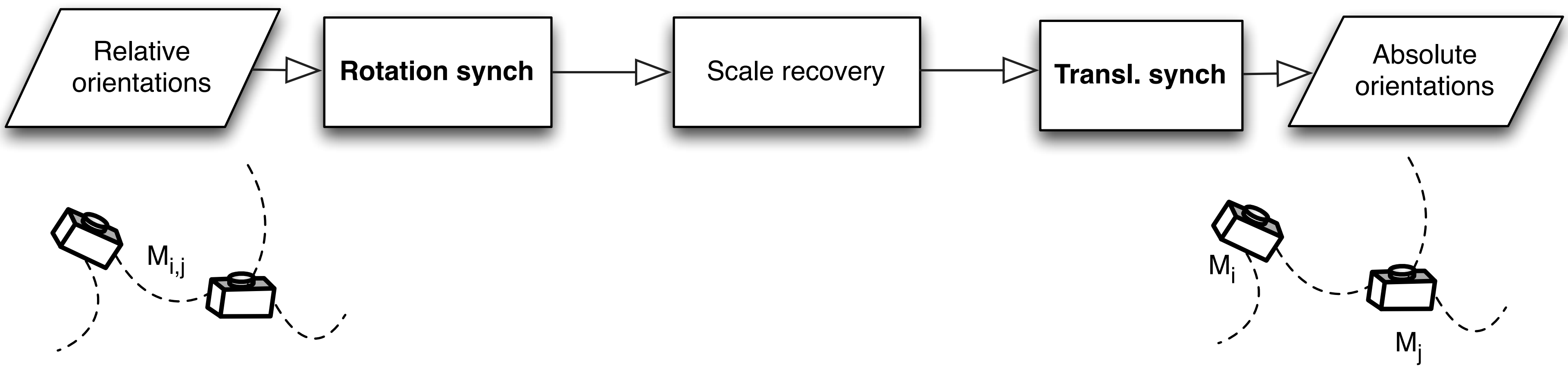
$$\cancel{(CB^T \otimes I_3)}\,\mathrm{vec}\,X = (C \odot \widehat{U})\boldsymbol{\alpha} \tag{20}$$

because $CB^T = 0$ for any cycle basis matrix C, leaving

$$\boxed{(C \odot \widehat{U})\boldsymbol{\alpha} = 0} \tag{21}$$

It can be seen that the above equation express the condition that with the correct $\boldsymbol{\alpha}$ the bearings sums up to zero in every cycle.

Baseline motion synchronization pipeline



4 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:

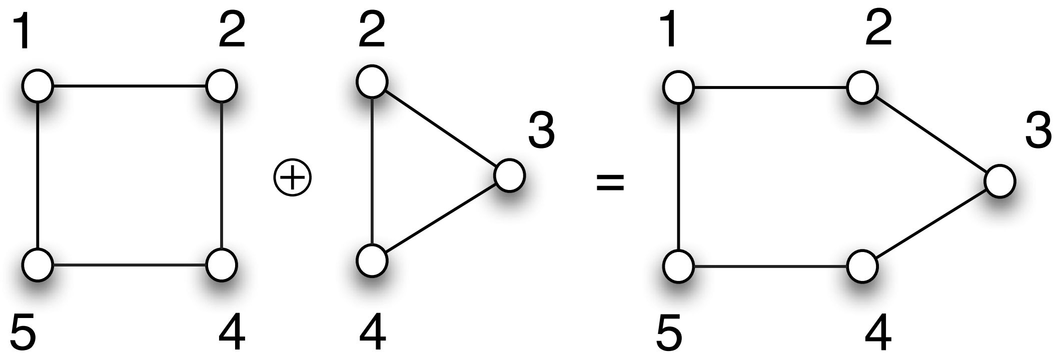
$$D(G)_{ij} = \begin{cases} \deg(v_i) = \sum_j A(G)_{i,j}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

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

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 $\mathbb{Z}_2^m$.



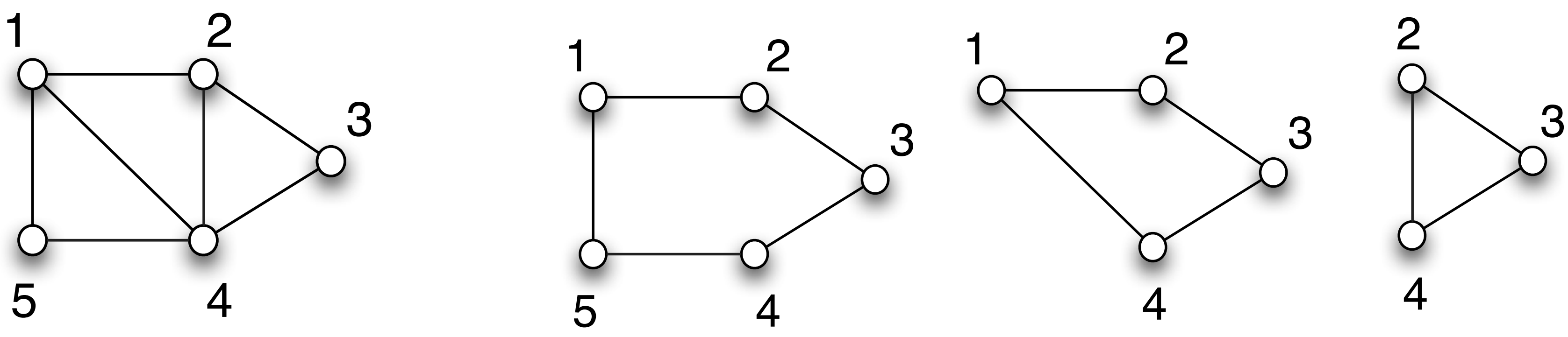
$$\begin{bmatrix}1\\0\\0\\1\\1\\1\end{bmatrix} + \begin{bmatrix}0\\1\\1\\1\\0\\0\end{bmatrix} = \begin{bmatrix}1\\1\\1\\0\\1\\1\end{bmatrix} \begin{matrix}(1,2)\\(2,3)\\(3,4)\\(2,4)\\(4,5)\\(5,1)\end{matrix}$$

Fig. 1: 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. 2: 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 $CB^T = 0$.

5 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 the those of B by $\mathbf{b}_1 \cdots \mathbf{b}_r$. The Khatri-Rao product is defined to be the partitioned matrix of order $pq \times r$:

$$A \odot B = [\mathbf{a}_1 \otimes \mathbf{b}_1, \cdots \mathbf{a}_r \otimes \mathbf{b}_r] \tag{22}$$

where $\otimes$ denotes the Kronecker product.

If X is diagonal, then

$$\mathrm{vec}(AXB) = (B^T \odot A)\, \mathrm{diag}^{-1}(X) \tag{23}$$

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

With $B = I$ one obtains

$$\mathrm{vec}(AX) = (I \odot A)\,\mathrm{diag}^{-1}(X). \tag{24}$$

It it is easy to see that

$$(I \odot A) = \mathrm{blkdiag}(\mathbf{a}_1 \ldots \mathbf{a}_n) \tag{25}$$

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

Property: $(C \otimes D)(A \odot B) = CA \odot DB$ (provided sizes are compatible)