Synchronization and Cycle-Consistency: Application to Motion Segmentation and Localization

Federica Arrigoni

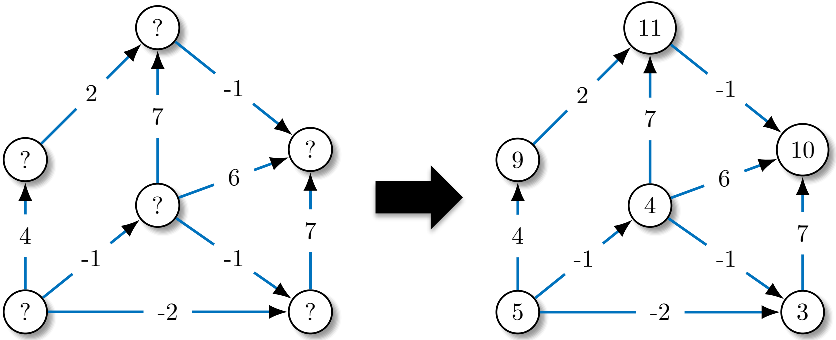
Department of Information Engineering and Computer Science
University of Trento (Italy)

Tutorial on Cycle Consistency and Synchronization in Computer Vision

In Conjunction with CVPR 2020

Time Synchronization

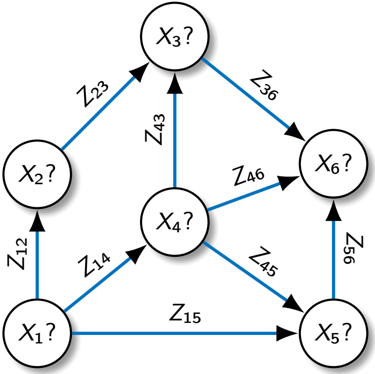
Consider a network of nodes where each node is characterized by an unknown state and pairs of nodes can measure the **difference** between their states.



The goal is to infer the unknown states from the pairwise measures.

Synchronization

The goal of synchronization is to recover elements of a **group**, given a certain number of their mutual differences (or **ratios**).



Consistency Constraint

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

The problem is well-posed only if the graph is **connected**.

Spectral Solution

If the group admits a matrix representation, the unknown elements can be recovered via **eigenvalue decomposition**.

$$Z_{ij} = X_i X_j^{-1} \quad \Longrightarrow \quad \boxed{ZX = nX}$$

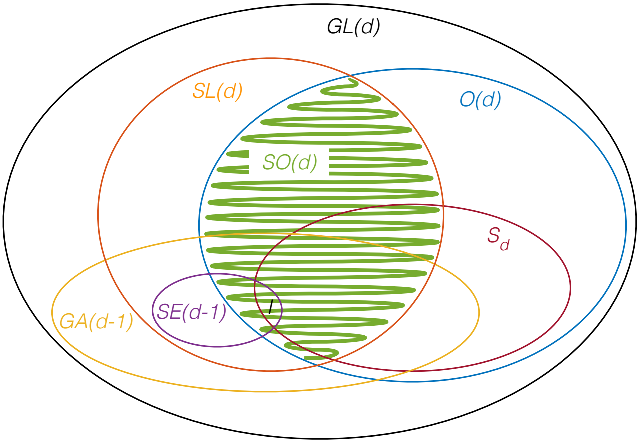
$$Z = \begin{bmatrix} I & Z_{12} & \dots & Z_{1n} \\ Z_{21} & I & \dots & Z_{2n} \\ \dots & & & \dots \\ Z_{n1} & Z_{n2} & \dots & I \end{bmatrix}, \quad X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix}$$

At the end the solution is **projected** onto the group.

Extensions

- Missing data $\longrightarrow$ adjacency/degree matrix
- Outliers $\longrightarrow$ iteratively reweighted least squares (IRLS)

Examples



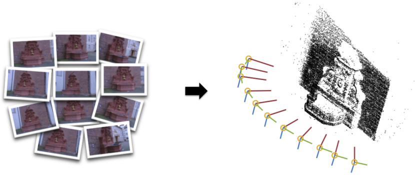
F. Arrigoni and A. Fusiello Synchronization problems in Computer Vision with closed-form solutions International Journal of Computer Vision (2020)

Example: rotations

$$SO(d) = \{M \in \mathbb{R}^{d\times d} s.t.\ M^\mathsf{T} M = MM^\mathsf{T} = I_d,\ \det(M) = 1\}$$

- A. Singer Angular synchronization by eigenvectors and semidefinite programming Applied and Computational Harmonic Analysis (2011)
- 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 (2011)
- M. Arie-Nachimson, S.Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer and R. Basri Global motion estimation from point matches Joint 3DIM/3DPVT Conference (2012)

Application: structure from motion



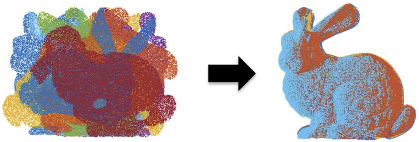
Example: rigid motions

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

F. Bernard, J. Thunberg, P. Gemmar, F. Hertel, A. Husch, and J. Goncalves A solution for multi-alignment by transformation synchronisation. CVPR (2015)

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

Application: registration



Example: permutations

$$\mathcal{S}_d = \{M \in \{0,1\}^{d\times d} \ s.t. \ M\mathbf{1} = \mathbf{1}, \ \mathbf{1}M = \mathbf{1}\}$$
$$\mathcal{I}_d = \{M \in \{0,1\}^{d\times d} \ s.t. \ M\mathbf{1} \leq \mathbf{1}, \ \mathbf{1}M \leq \mathbf{1}\}$$

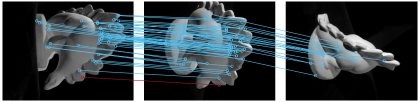
D. Pachauri, R. Kondor, and V. Singh Solving the multi-way matching problem by permutation synchronization. NIPS (2013)

Y. Shen, Q. Huang, N. Srebro and S. Sanghavi Normalized spectral map synchronization. NIPS (2016)

E. Maset, F. Arrigoni and A. Fusiello Practical and efficient multi-view matching ICCV (2017)

F. Bernard, J. Thunberg, P. Swoboda, C. Theobalt HiPPI: Higher-Order Projected Power Iterations for Scalable Multi-Matching ICCV (2019)

Application: multi-view matching

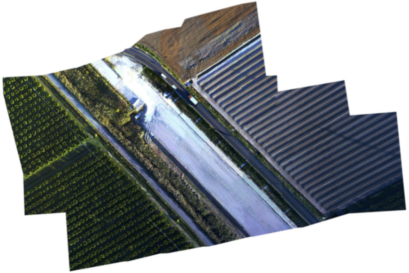


Example: homographies

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

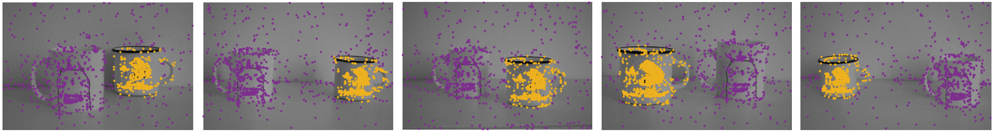
- P. Schroeder, A. Bartoli, P. Georgel and N. Navab Closed-form solutions to multiple-view homography estimation. WACV (2011)
- E. Santellani, E. Maset, and A. Fusiello Seamless image mosaicking via synchronization ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (2018)

Application: mosaicking

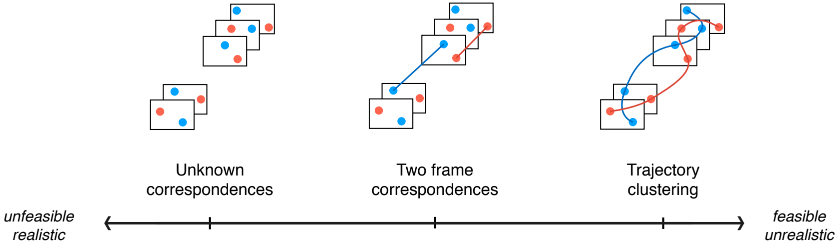


Application to Motion Segmentation

The goal is to classify points in multiple images based on the moving object they belong to.

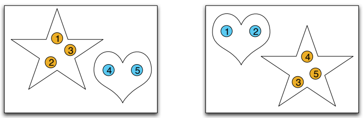


Two-frame correspondences (i.e. pairwise matches) are assumed as input.



Problem Formulation

Motion segmentation can be seen as a synchronization of **binary** matrices.



$$\begin{array}{c|c} & \text{image j} \\ & 1\ \ 2\ \ 3\ \ 4\ \ 5 \\ \text{image i} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad \begin{array}{c|c} & \text{universe} \\ & \star\ \ \heartsuit \\ \text{image i} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array} \qquad \begin{array}{c|c} & \text{universe} \\ & \star\ \ \heartsuit \\ \text{image j} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \end{array}$$

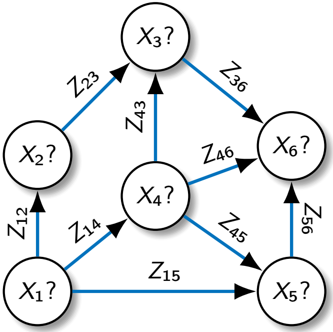
Partial segmentation

- $[Z_{ij}]_{h,k} = 1$ if point h in image i and point k in image j belong to the same motion;
- $[Z_{ij}]_{h,k} = 0$ otherwise.

Total segmentation

- $[X_i]_{h,k} = 1$ if point h in image i belongs to motion k;
- $[X_i]_{h,k} = 0$ otherwise.

Problem Formulation



Consistency constraint

$$S_{ij} = S_i S_j^{\mathsf{T}}$$

1. Each partial segmentation is computed by fitting multiple fundamental matrices to corresponding points.

   L. Magri and A. Fusiello Robust multiple model fitting with preference analysis and low-rank approximation. BMVC (2015)

2. Total segmentations are derived via synchronization (spectral solution).

Spectral Solution

Consistency constraint $\boxed{Z = XX^{\mathsf{T}}} \implies \operatorname{rank}(Z) = d$

$$\textbf{input} \longleftarrow \quad Z = \begin{bmatrix} Z_{11} & Z_{12} & \dots & Z_{1n} \\ Z_{21} & Z_{22} & \dots & Z_{2n} \\ \dots & & & \dots \\ Z_{n1} & Z_{n2} & \dots & Z_{nn} \end{bmatrix} \qquad X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix} \longrightarrow \quad \textbf{output}$$

Remark. $X_i^{\mathsf{T}} X_i$ is a $d \times d$ diagonal matrix s.t. the (k,k)-entry counts the number of points in image i that belong to motion k. Thus $X^{\mathsf{T}} X = \sum_{i=1}^{n} (X_i^{\mathsf{T}} X_i)$ is a diagonal matrix s.t. (k,k)-entry counts the number of points over all the images that belong to motion k.

$$ZX = XX^{\mathsf{T}}X = X \underbrace{\sum_{i=1}^{n} (X_i^{\mathsf{T}} X_i)}_{\Lambda} = X \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{bmatrix}$$

Spectral Solution

Proposition

The columns of X are d (orthogonal) eigenvectors of Z, where d denotes the number of motions. If each motion contains a different number of points, then the eigenvalues are distinct.

Algorithm - Synch

1. Compute the top d eigenvectors of Z, collected in a matrix U.
2. Transform U into a binary matrix:
   - $[X]_{h,k} = 1$ if the following conditions are satisfied:
     a) $[U]_{h,k}$ is the maximum value over row h;
     b) $[U]_{h,k} \neq 0$;
     c) $[U]_{h,k}/[U]_{h,l} > \theta$ where $[U]_{h,l}$ is the 2nd-maximum value over row h;
   - $[X]_{h,k} = 0$ otherwise.

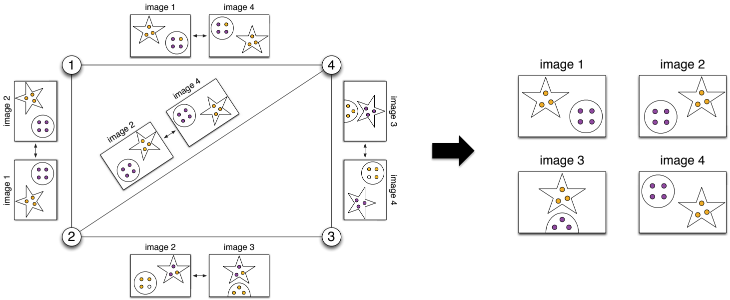
F. Arrigoni and T. Pajdla Motion segmentation via synchronization ICCV Workshops (2019)

Local solution

1. Segmentation of corresponding points is performed on each image pair in isolation (e.g. by fitting multiple fundamental matrices).

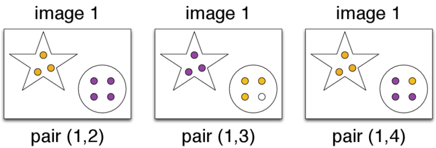
   L. Magri and A. Fusiello Robust multiple model fitting with preference analysis and low-rank approximation. BMVC (2015)

2. Segmentation of image points is computed without relying on the feature locations, using only the classification of matching points derived in Step 1.

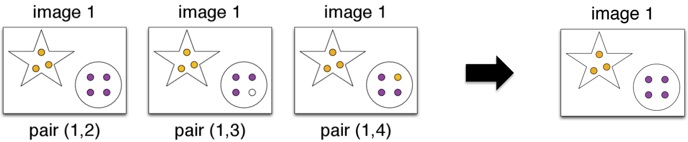


Local solution

- Idea: all the two-frame segmentations involving a fixed image provide (up to a permutation) an estimate for the segmentation of that image.

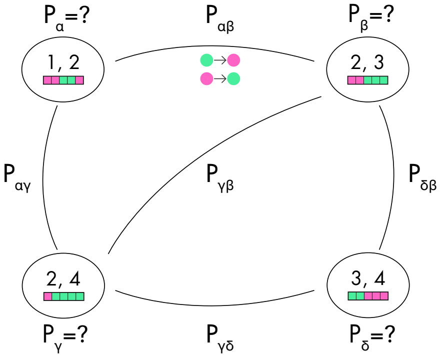


- The permutation ambiguity can be solved via **permutation synchronization** (next slide).
- For each point in each image, several putative labels are available: the most frequent label (**mode**) is chosen.



Local solution

Graph representation: each vertex corresponds to one pair of images; an edge is present between two vertices if and only if the associated pairs have one image in common.



- Compute a permutation for each edge: *linear assignment problem*

  H. W. Kuhn The Hungarian method for the assignment problem Naval Research Logistics Quarterly 2 (1955)

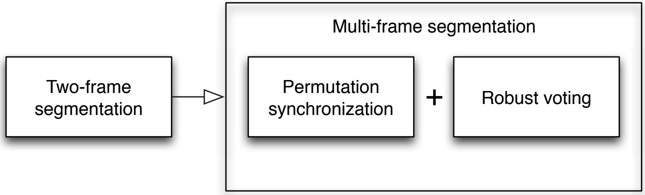
- Compute permutations for all the nodes: *permutation synchronization*

  D. Pachauri, R. Kondor, and V. Singh Solving the multi-way matching problem by permutation synchronization NIPS (2013)

Remark: the size of permutation matrices is equal to the number of motions (assumed known).

Local solution

Algorithm - Mode



F. Arrigoni and T. Pajdla Robust motion segmentation from pairwise matches ICCV (2019)

Remark: this approach is **local**, as each image is segmented based on its neighbors only. Local solutions appeared also in the case of rotations and rigid-motions:

R. Hartley, K. Aftab and J. Trumpf L1 rotation averaging using the Weiszfeld algorithm. CVPR (2011)

A. Torsello, E. Rodolá and A. Albarelli Multiview registration via graph diffusion of dual quaternions CVPR (2011)

Experiments - Trajectory Clustering

Hopkins155 – Misclassification Error [%]

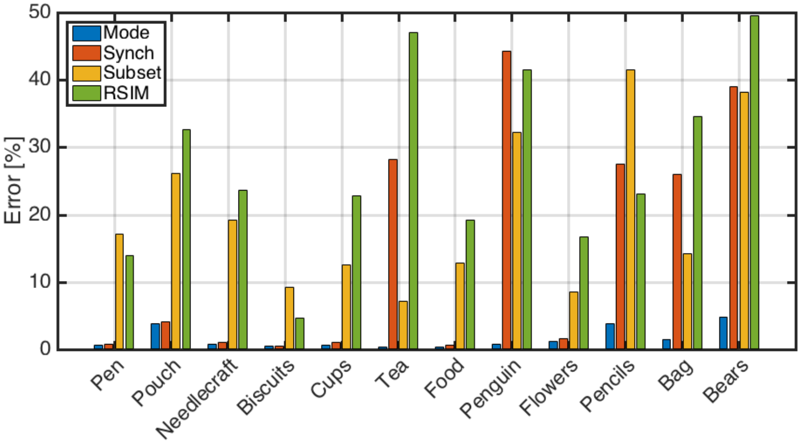
	LSA	GPCA	ALC	SSC	TPV	LRR	T-Linkage	S^3C	RSIM	MSSC	KerAdd	Coreg	Subset	Synch	Mode
2 Motions	4.23	4.59	2.40	1.52	1.57	1.33	0.86	1.94	0.78	0.54	0.27	0.37	**0.23**	2.70	1.00
3 Motions	7.02	28.66	6.69	4.40	4.98	4.98	5.78	4.92	1.77	1.84	0.66	0.75	**0.58**	6.99	2.67
All	4.86	10.02	3.56	2.18	2.34	1.59	1.97	2.61	1.01	0.83	0.36	0.46	**0.31**	3.67	1.37

Hopkins12 – Misclassification Error [%]

	PF	PF+ALC	RPCA+ALC	ℓ_1+ALC	SSC-R	SSC-O	RSIM	KerAdd	Coreg	Subset	Synch	Mode
Mean	14.94	10.81	13.78	1.28	3.82	8.78	0.61	0.11	**0.06**	**0.06**	5.46	4.33
Median	9.31	7.85	8.27	1.07	0.31	4.80	0.61	**0.00**	**0.00**	**0.00**	0.57	0.38

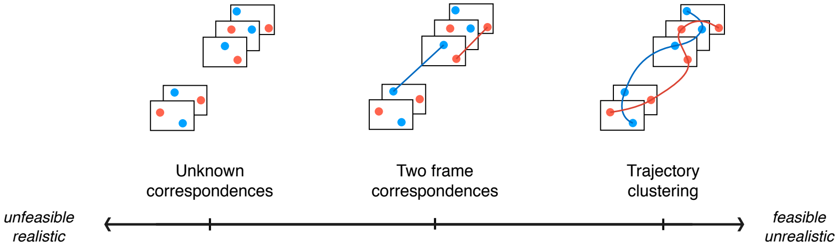
http://www.vision.jhu.edu/data/hopkins155/

Experiments - Pairwise Matches



https://github.com/federica-arrigoni/ICCV_19

Experiments - Comments



- Methods belonging to a specific category are **sub-optimal** when applied to the task associated to another category.
- The **local solution** (Mode) is more accurate than the spectral solution (Synch) for motion segmentation with two-frame correspondences.

Summary

Spectral solution

✓ It is very **general**: it can be applied to any group admitting a matrix representation (rotations/rigid-motions/permutations/homographies) and also to sets that do not have the structure of a group (binary matrices).

✓ It involves a variety of **applications** in Computer Vision (structure from motion, registration, matching, mosaicking, motion segmentation, ...).

✓ It entails a **simple** implementation: spectral decomposition + projection.

✗ It is an **approximate** solution that does not enforce geometric constraints.

Local solution

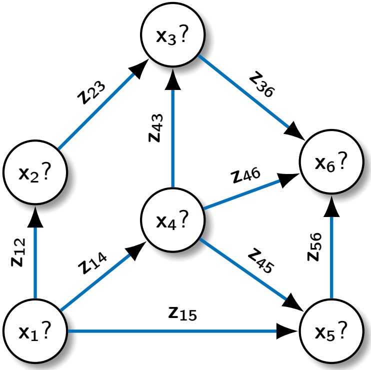
✓ It is more **accurate** than the spectral solution on motion segmentation datasets.

✓ The framework is **modular**: it can be easily generalized to more practical scenarios (e.g. the case of unknown number of motions).

✗ The algorithm is **specific** for the motion segmentation problem.

Translation synchronization

Let us consider the **translation synchronization** problem: the task is to recover the position of n nodes in d-space, starting from pairwise differences.



Consistency Constraint

$$\boxed{\mathbf{z}_{ij} = \mathbf{x}_i - \mathbf{x}_j}$$

$\mathbf{x}_i \in \mathbb{R}^d$ unknown

$\mathbf{z}_{ij} \in \mathbb{R}^d$ known

The solution is defined up to a global translation.

Translation synchronization

The unknown node locations can be recovered as the solution of a **linear system** of equations:

$$\mathbf{x}_i - \mathbf{x}_j = \mathbf{z}_{ij} \quad \forall\, (i,j) \in \mathcal{E} \quad \iff \quad \boxed{(B^{\mathsf{T}} \otimes I_d)\,\mathrm{vec}(X) = \mathrm{vec}(Z)}$$

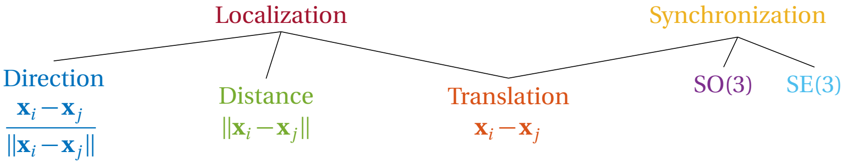
$$\begin{aligned} B &= \text{ incidence matrix} \\ \otimes &= \text{ Kronecker product} \\ I_d &= d \times d \text{ identity matrix} \\ X &= [\mathbf{x}_1 \dots \mathbf{x}_n] \\ Z &= [\mathbf{z}_{12} \ \dots \ \mathbf{z}_{ij} \ \dots] \end{aligned}$$

Remark. Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ connected $\Longrightarrow \mathrm{rank}(B) = n - 1 \Longrightarrow \mathrm{rank}(B^{\mathsf{T}} \otimes I_d) = dn - d$.
The rank deficiency corresponds to the **translation ambiguity**.

F. Arrigoni and A. Fusiello Synchronization problems in Computer Vision with closed-form solutions International Journal of Computer Vision (2020)

Localization

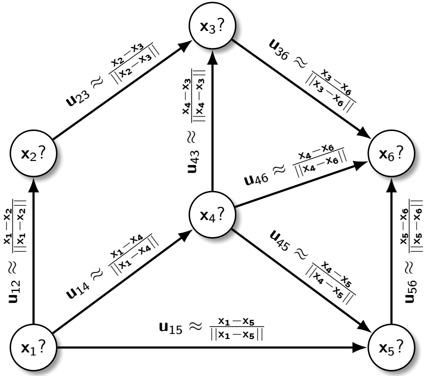
The goal of localization is to compute the position of n nodes in d-space given measures (directions/distances/differences) on the edges.



We are interested here in **direction-based** localization (a.k.a. bearing-based localization), which is **not** a synchronization problem.

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**.

Bearing-based localization

Let us rewrite the consistency constraint of translation synchronization in terms of **directions** (known) and **magnitudes** (unknown).

$$\mathbf{x}_i - \mathbf{x}_j = \mathbf{z}_{ij} \quad \forall\, (i,j) \in \mathcal{E} \qquad \Longleftrightarrow \qquad \boxed{(B^{\mathsf{T}} \otimes I_d)\,\mathrm{vec}(X) = \mathrm{vec}(Z)}$$

$$\mathbf{x}_i - \mathbf{x}_j = \alpha_{ij}\mathbf{u}_{ij} \quad \forall\, (i,j) \in \mathcal{E} \qquad \Longleftrightarrow \qquad \boxed{(B^{\mathsf{T}} \otimes I_d)\,\mathrm{vec}(X) = (I_m \odot U)\boldsymbol{\alpha}}$$

$$m = \text{number of edges}$$
$$\odot = \text{Khatri-Rao product}$$
$$U = [\mathbf{u}_{12} \;\ldots\; \mathbf{u}_{ij} \;\ldots]$$

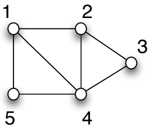
Let us consider a **cycle matrix** C and multiply both sides by $C \otimes I_d$.

$$(CB^{\mathsf{T}} \otimes I_d)\,\mathrm{vec}(X) = (C \odot U)\boldsymbol{\alpha} \quad \underset{CB^{\mathsf{T}}=0}{\Longrightarrow} \quad \boxed{0 = (C \odot U)\boldsymbol{\alpha}}$$

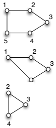
F. Arrigoni, A. Fusiello and B. Rossi On computing the translations norm in the epipolar graph *3DV* (2015)

Bearing-based localization

Given a graph we can compute a **cycle basis**, represented as a cycle matrix.



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



The equation $(C \odot U)\alpha = 0$ means that we are imposing **cycle-consistency**: *the sum of differences along any cycle in a basis is zero.*

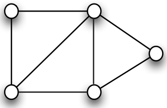
$$
(C \odot U)\alpha = 0 \iff \begin{bmatrix} \mathbf{u}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{u}_{23} & \mathbf{u}_{34} & \mathbf{u}_{45} & \mathbf{u}_{51} \\ \mathbf{u}_{12} & \mathbf{0} & \mathbf{u}_{41} & \mathbf{u}_{23} & \mathbf{u}_{34} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{u}_{24}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

Bearing-based localizability

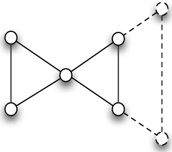
Requiring that the underlying graph is **connected** is **not** sufficient for unique localizability.

The graph is required to be **parallel rigid**: *all the configurations with parallel edges differ by translation and scale*. Otherwise it is called flexible.

Parallel Rigid



Flexible



W. Whiteley Matroids from Discrete Geometry *American Mathematical Society* (1997)

T. Eren and W. Whiteley and P. N. Belhumeur Using angle of arrival (bearing) information in network localization *IEEE Conference on Decision and Control* (2006)

Bearing-based localizability

Proposition. ***Node locations*** *can be uniquely (up to translation and scale) determined from pairwise directions if and only if* ***edge lengths*** *can be uniquely (up to scale) recovered from pairwise directions.*

Corollary. *A graph is parallel rigid in d-space if and only if* $\operatorname{rank}(C \odot U) = m - 1$, *where*

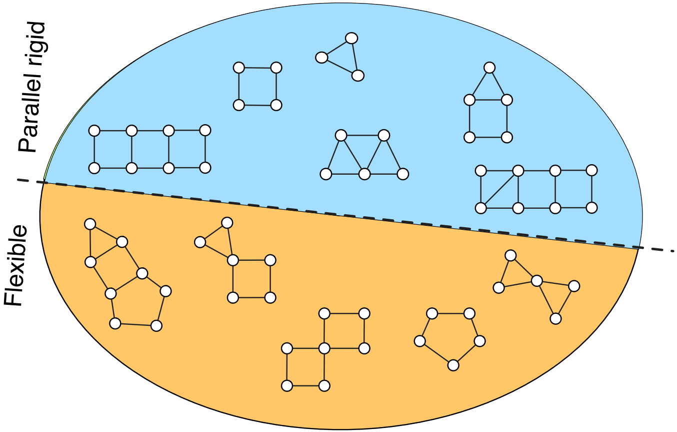
$$
\begin{aligned}
C &= \text{cycle matrix} \\
\odot &= \text{Khatri-Rao product} \\
U &= [\mathbf{u}_{12} \;\; \dots \;\; \mathbf{u}_{ij} \;\; \dots] \\
m &= \text{number of edges}
\end{aligned}
$$

F. Arrigoni and A. Fusiello Bearing-based network localizability: a unifying view IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)

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

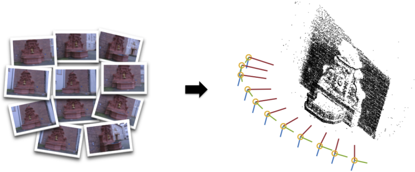
Bearing-based localizability

Examples (d=3)



Application to Structure from Motion

Estimating **camera positions** in structure from motion is an instance of bearing-based localization in 3-space.



- If the **viewing graph** is parallel rigid, then the problem is well-posed.
- If the graph is flexible, then the **largest rigid component** has to be extracted.

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

R. Kennedy, K. Daniilidis, O. Naroditsky, and C. J. Taylor Identifying maximal rigid components in bearing-based localization, Proc. Int. Conf. Intell. Robots Syst. (2012)

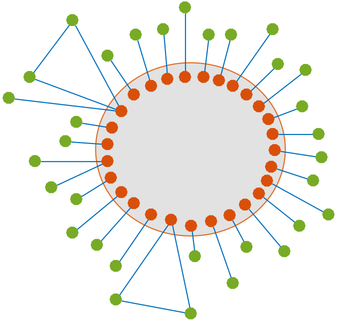
Application to Structure from Motion

Dataset	nodes	% edges	rigid	articulation	bridges
Arts Quad	5530	2	✗	30	10
Piccadilly	2508	10	✗	59	62
Roman Forum	1134	11	✗	28	28
Union Square	930	6	✗	60	68
Vienna Cathedral	918	25	✗	19	20
Alamo	627	50	✗	17	19
Notre Dame	553	68	✓	–	–
Tower of London	508	19	✗	19	19
Montreal N. Dame	474	47	✗	7	7
Yorkminster	458	26	✗	9	10
Madrid Metropolis	394	31	✗	17	15
NYC Library	376	29	✗	17	18
Piazza del Popolo	354	40	✗	8	9
Ellis Island	247	67	✗	6	7

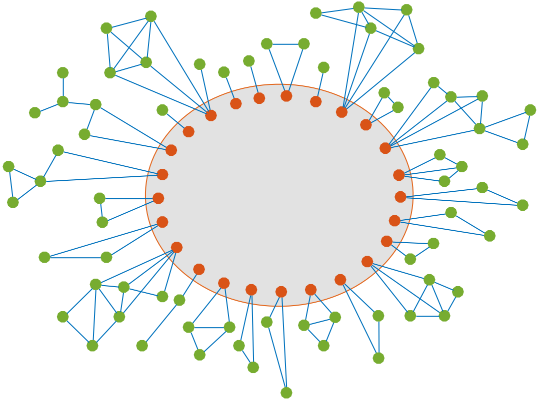
https://research.cs.cornell.edu/1dsfm/

Application to Structure from Motion

Roman Forum



Arts Quad



More information at

https://synchinvision.github.io/