Image Mosaicking via Synchronization

Image Analysis and Computer Vision

Federica Arrigoni – federica.arrigoni@polimi.it

POLITECNICO
MILANO 1863

Exercise Session – October 30, 2024

Outline

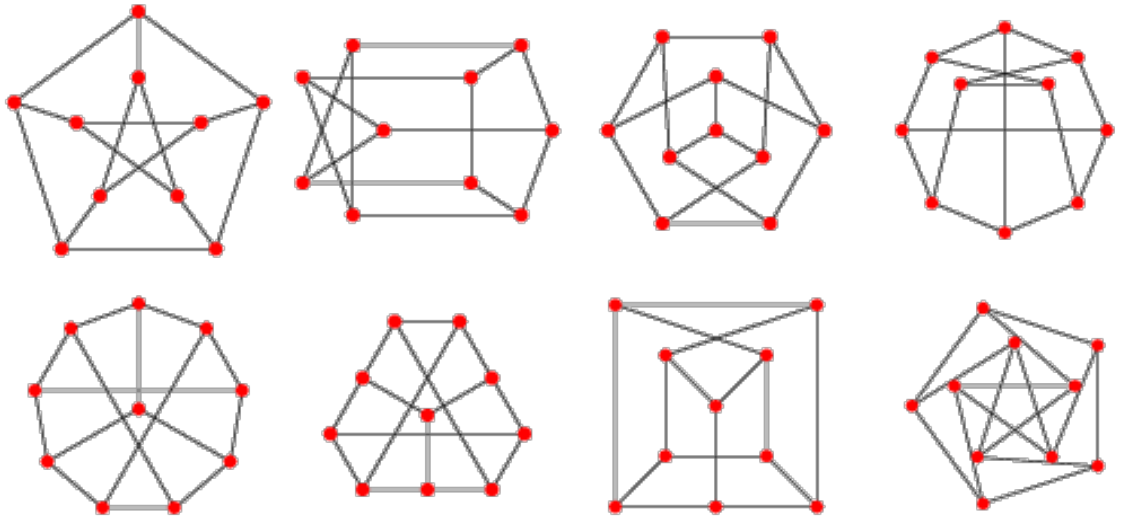
- Background
- Introduction
- Spectral solution
- Application to image mosaicking
- Conclusion & extensions
- Matlab demo

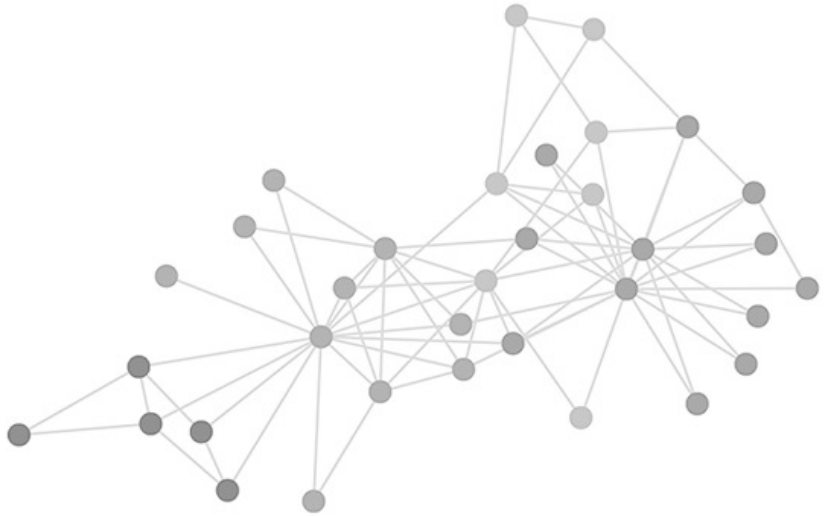
Outline

- **Background**
- Introduction
- Spectral solution
- Application to image mosaicking
- Conclusion & extensions
- Matlab demo

Elements of Graph Theory

Definition. *A graph is a pair (V,E) where where V is a set whose elements are called* **vertices/nodes** *and E is a set of paired vertices, which are called* **edges**.

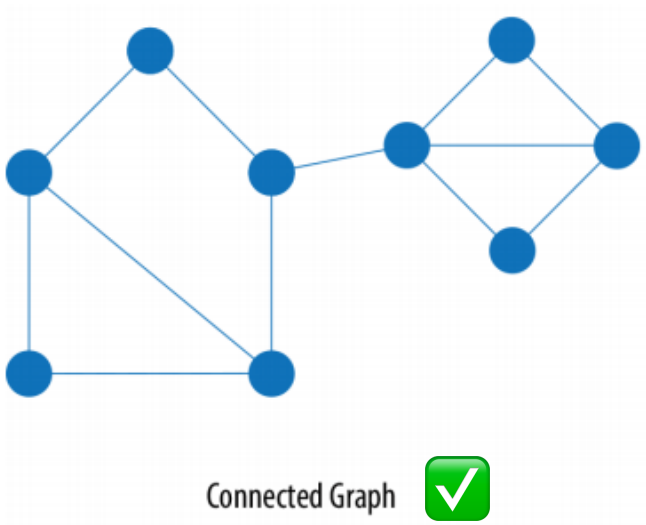


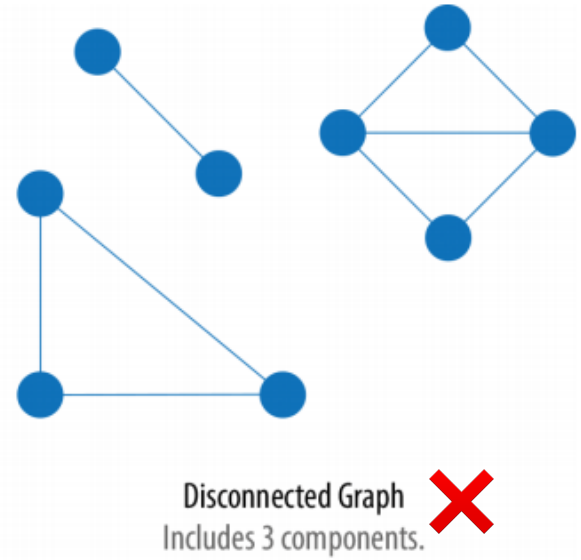


Notation: n = #V, m=#E

Elements of Graph Theory

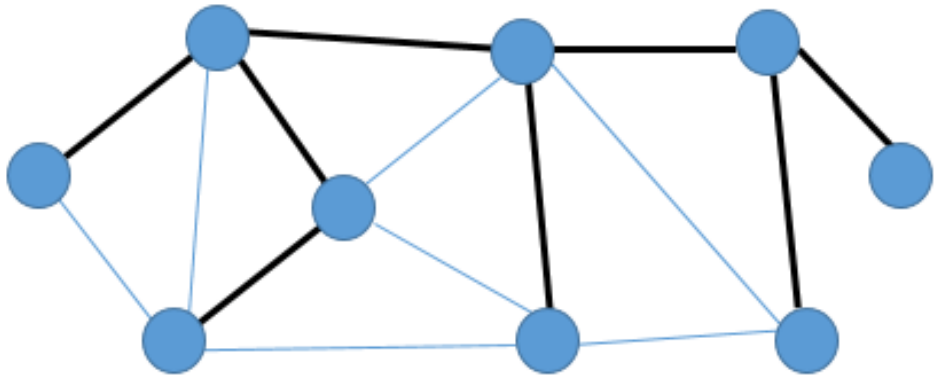
Definition. *A graph is called* **connected** *if there exists a path from each vertex to any other.*





Elements of Graph Theory

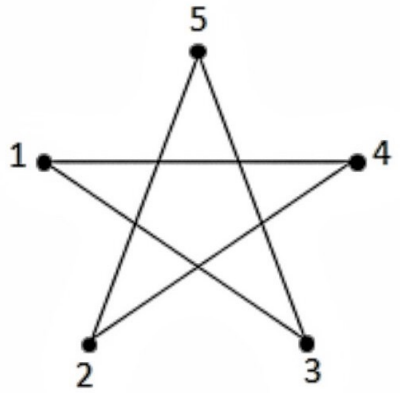
Definition. *A* **spanning tree** *in a graph is a minimal set of edges that connect all vertices.*



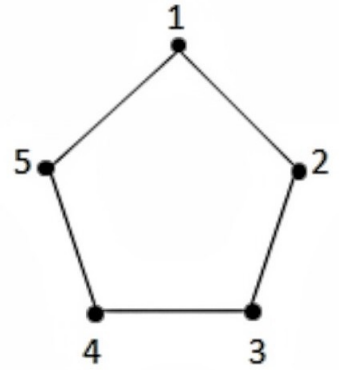
Elements of Graph Theory

Definition. *The* **adjacency matrix** *A of a graph (V,E) with n nodes and m edges has dimension n x n and it is constructed as follows:*

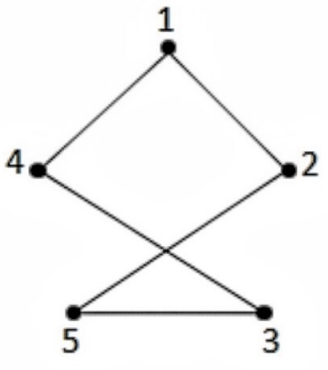
$$[A]_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$



	1	2	3	4	5
1			1	1	
2				1	1
3	1				1
4	1	1			
5		1	1		



	1	2	3	4	5
1		1			1
2	1		1		
3		1		1	
4			1		1
5	1			1	

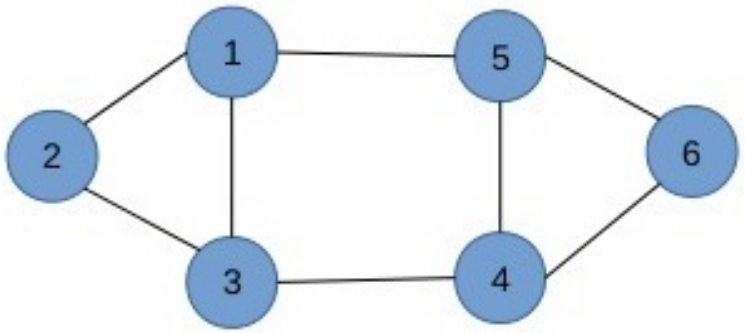


	1	2	3	4	5
1		1		1	
2	1				1
3				1	1
4	1		1		
5		1	1		

Elements of Graph Theory

Definition. *The **degree** **matrix** D of a graph (V,E) with n nodes and m edges is a diagonal matrix with dimension n x n such that:*

$$[D]_{i,i} = \#\{\text{edges connected to node } i\}$$



$$D = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Kronecker product

Definition. Let A and B be two matrices of dimension m × r and n × s respectively. The Kronecker product of A and B is a mn x rs matrix denoted by A ⊗ B :

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

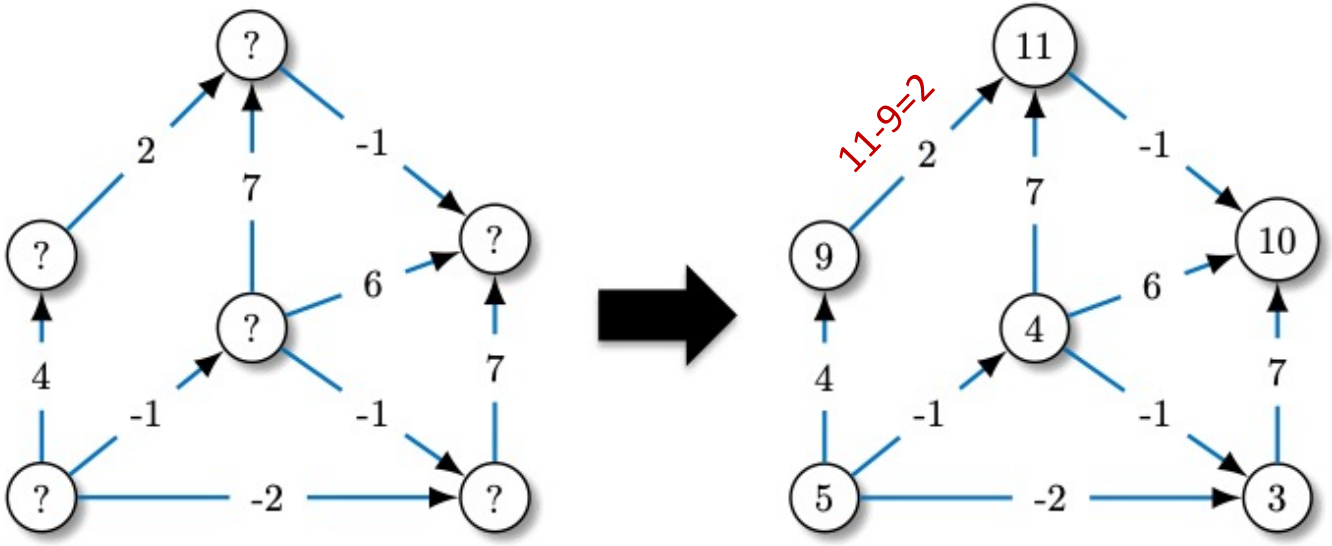
Example: $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 7 & 5 \\ 6 & 0 \end{bmatrix} = \begin{bmatrix} 1\begin{bmatrix} 7 & 5 \\ 6 & 0 \end{bmatrix} & 2\begin{bmatrix} 7 & 5 \\ 6 & 0 \end{bmatrix} \\ 3\begin{bmatrix} 7 & 5 \\ 6 & 0 \end{bmatrix} & 4\begin{bmatrix} 7 & 5 \\ 6 & 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 7 & 5 & 14 & 10 \\ 6 & 0 & 12 & 0 \\ 21 & 15 & 28 & 20 \\ 18 & 0 & 24 & 0 \end{bmatrix}$

Outline

- Background
- **Introduction**
- Spectral solution
- Application to image mosaicking
- Conclusion & extensions
- Matlab demo

Example

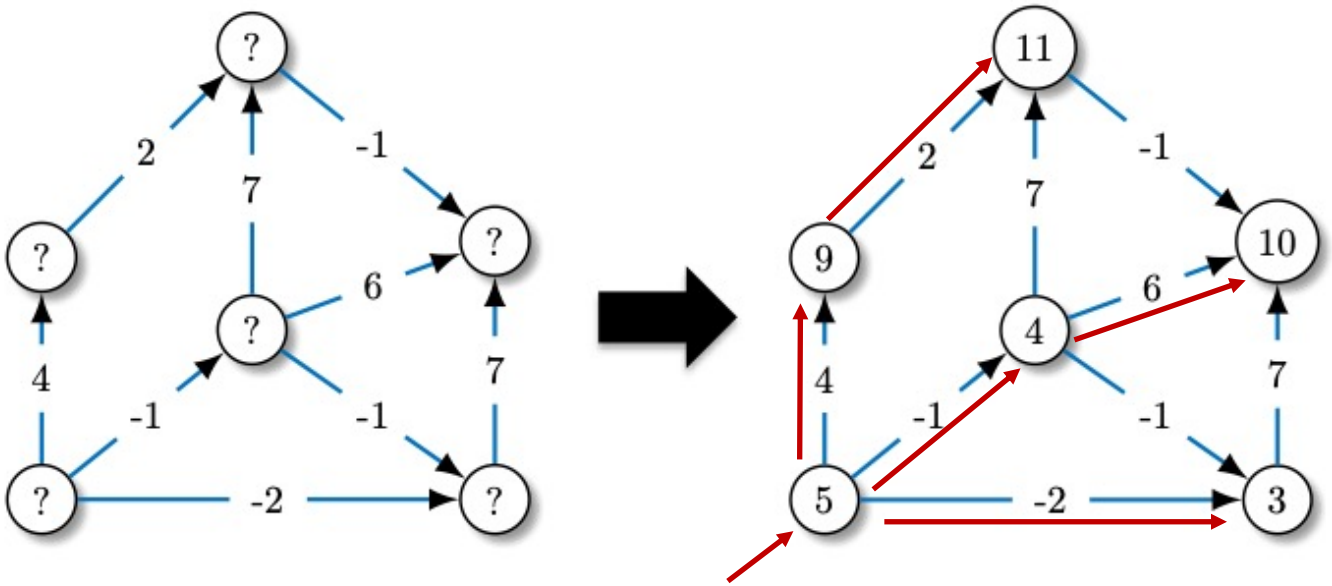
Consider a network of nodes where each node is characterized by an unknown state, and pairs of nodes can measure the **difference** between their states.



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

Example

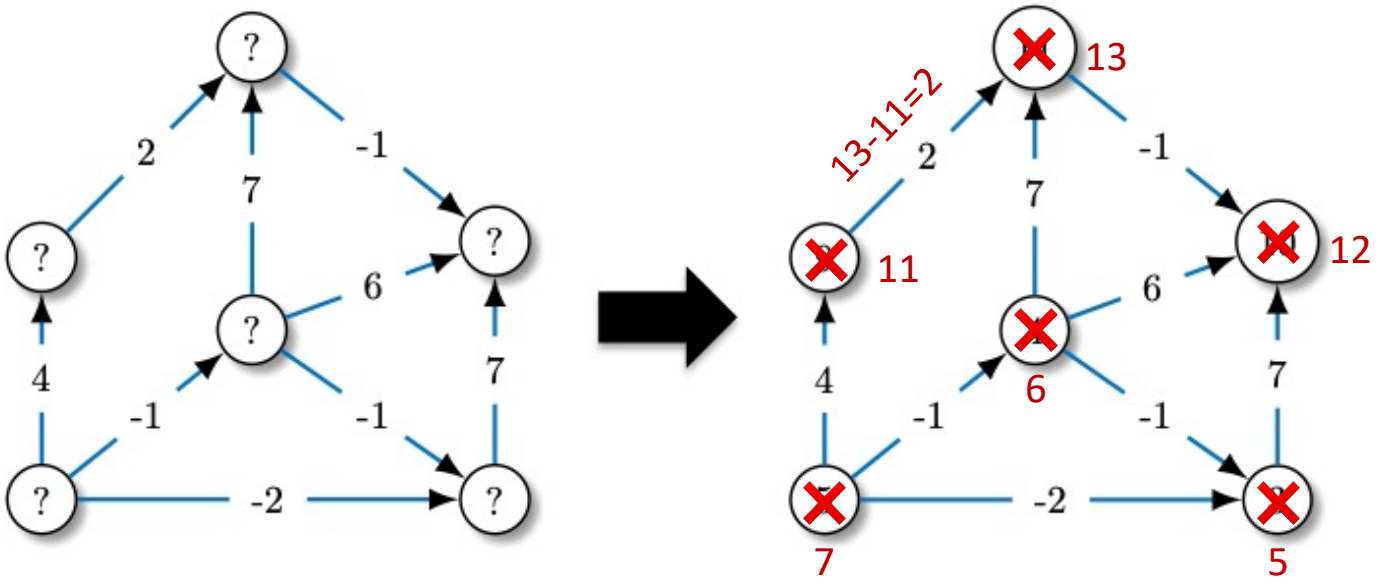
Consider a network of nodes where each node is characterized by an unknown state, and pairs of nodes can measure the **difference** between their states.



👉 A solution can be obtained by **sequential propagation**, starting from any node. The value of the initial node is **arbitrary**.

Example

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.

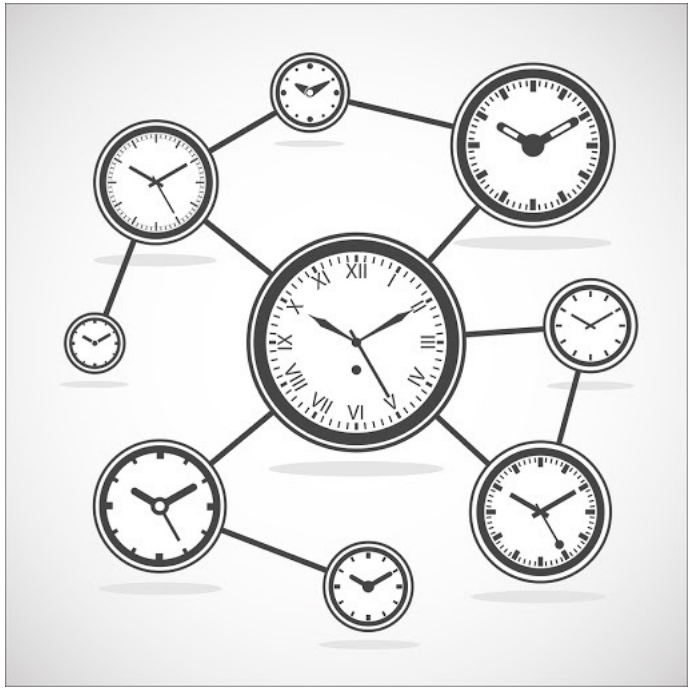


👉 The solution is **not unique**: adding the same constant to all the nodes gives another valid solution.

Example

Clock/time synchronization

- Each node has its own clock that tells the **time**.
- Each edge represents the ability to transmit and receive messages between the corresponding pair of nodes, which enables to measure clock **differences**.



The task is to enable each node in the network to convert its own time to that of a **unique common clock** (e.g. the clock of node 1).

A. Giridhar, P. Kumar. *Distributed clock synchronization over wireless networks: algorithms and analysis*. Conference on Decision and Control (2006)

Problem Definition

The problem of clock synchronization can be generalized to the case where states are elements of a **group**.

Definition. *A group is a set Σ equipped with a binary operation $*$ that combines any two elements to form a third element, such that:*

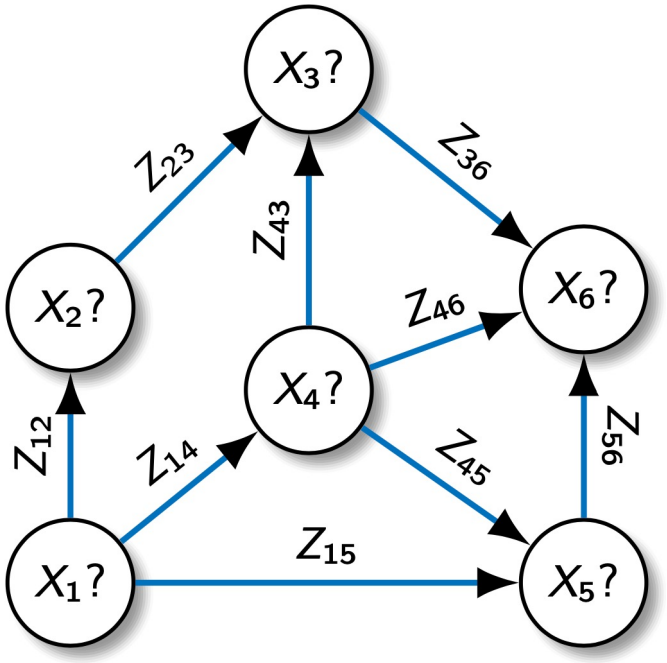
- **Associativity:** *for all $x, y, z \in \Sigma$ we have $(x * y) * z = x * (y * z)$.*
- **Identity:** *there exists an element 1_Σ such that for every $x \in \Sigma$ we have $1_\Sigma * x = x = x * 1_\Sigma$*
- **Inverse:** *for each $x \in \Sigma$ there exists an element $u \in \Sigma$ such that $x * u = 1_\Sigma = u * x$. Such element is commonly denoted by x^{-1}.*

Clock synchronization: integer (real) numbers

Examples in Computer Vision: rotations, permutations, …

Problem Definition

Graph representation: Nodes = unknowns, Edges = observations



Time synchronization

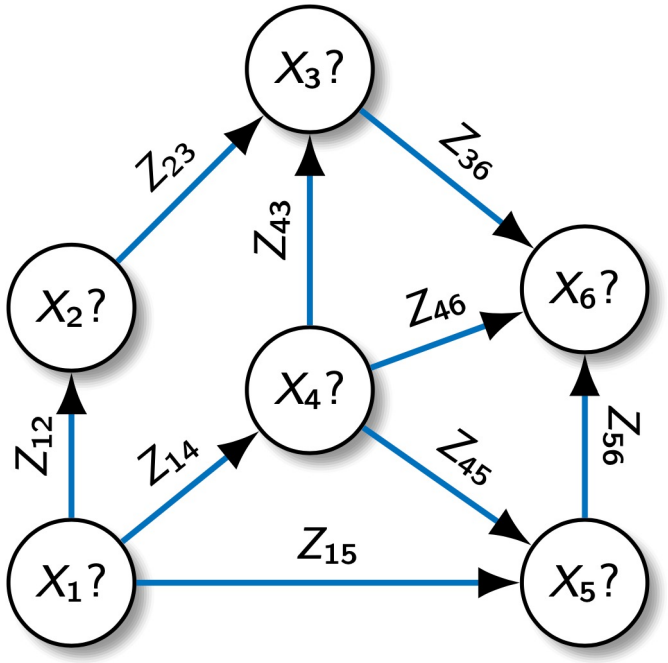
$$Z_{ij} = X_i - X_j = X_i + (-X_j)$$

General case (group)

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

Problem Definition

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



Graph representation:

Nodes = unknowns

Edges = observations

Consistency Constraint

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

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

Problem Definition

🤔 Is the solution **unique**?

The solution is defined up to a global **group element**:

$$\underbrace{Z_{ij} = X_i * X_j^{-1}} = X_i * \underbrace{(S * S^{-1})} * X_j^{-1} = (\underbrace{X_i * S}) * (\underbrace{X_j * S})^{-1}$$

Consistency constraint

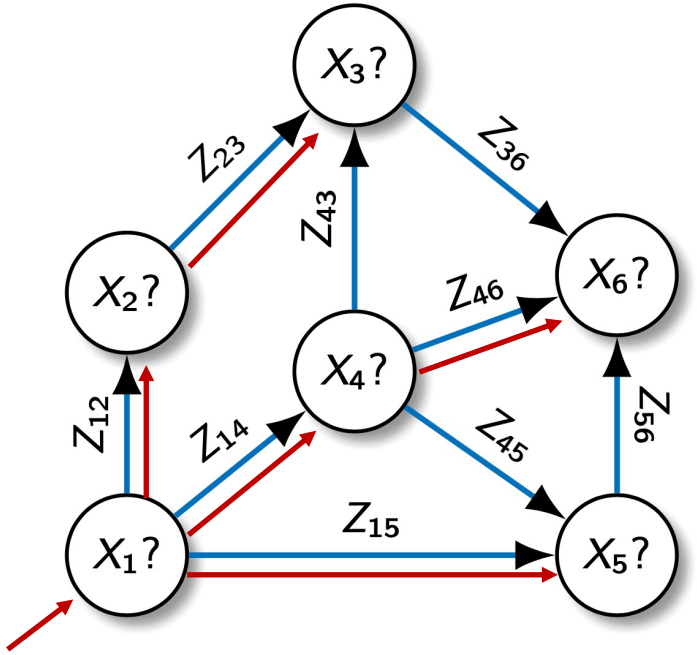
Identity

New group elements (change of variables)

👉 The **ambiguity** can be fixed by arbitrarily choosing one node (e.g. node 1) and assigning it to the **identity**.

Problem Definition

A solution can be found via sequential propagation along a **spanning tree**.



Equivalent Expression for the Consistency Constraint

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

$$X_1 = 1_\Sigma$$

$$X_4 = Z_{41} * X_1 = Z_{41}$$

$$X_6 = Z_{64} * X_4 = Z_{64} * Z_{41}$$

$$\dots$$

👉 Each measure satisfies: $Z_{ji} = Z_{ij}^{-1}$

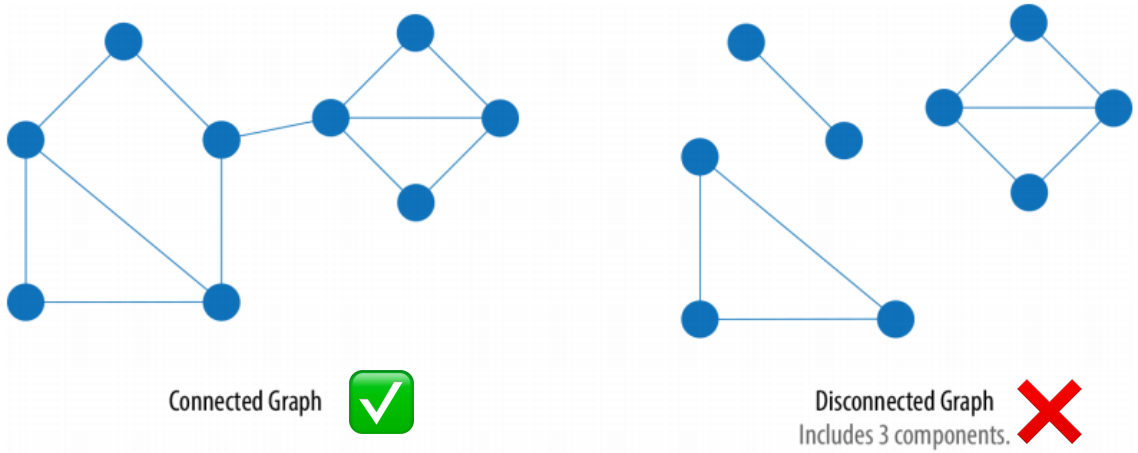
Problem Definition

❌ The spanning tree solution is subject to **error accumulation**.

✅ The key component for achieving error compensation is to exploit **redundancy**, namely considering the whole graph.

🤔 Under which assumption synchronization is **well-posed**?

The graph must be **connected** (= there is a path between any pair of nodes).



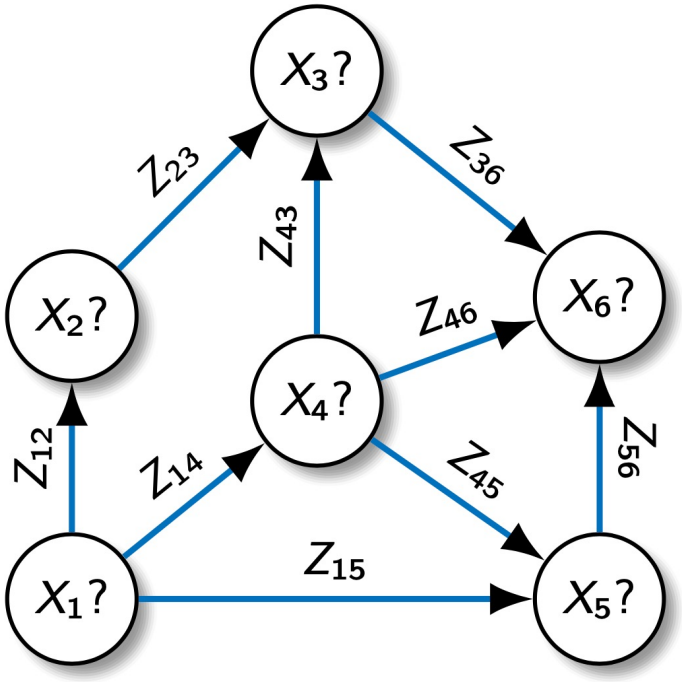
Outline

- Background
- Introduction
- **Spectral solution**
- Application to image mosaicking
- Conclusion & extensions
- Matlab demo

Problem Formulation

Let us consider the **General Linear Group** with matrix multiplication:

$$GL(d) = \{M \in \mathbb{R}^{d \times d} \text{ such that } \det(M) \neq 0\}$$ ------> d x d invertible matrices



Graph representation:

Nodes = unknowns

Edges = observations

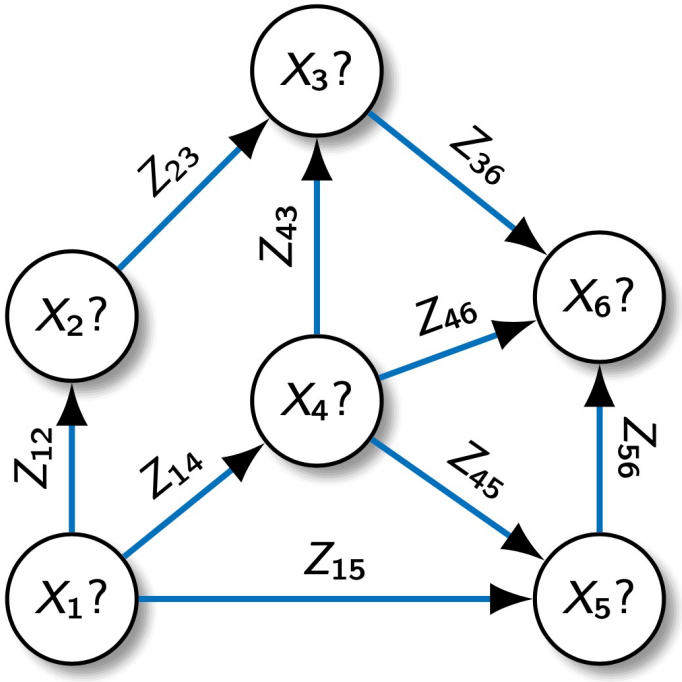
Consistency Constraint

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

Problem Formulation

The consistency constraint holds for a **single edge** in the graph: $Z_{ij} = X_i X_j^{-1}$

When considering the **entire graph**, we have multiple equations:



known unknown

$$Z_{12} = X_1 X_2^{-1}$$

$$Z_{23} = X_2 X_3^{-1}$$

$$Z_{14} = X_1 X_4^{-1}$$

...

The case of a complete graph

The consistency constraint holds for a **single edge** in the graph: $Z_{ij} = X_i X_j^{-1}$

When considering the **entire graph**, an equivalent compact form can be obtained.

Let us consider the auxiliary variable: $Y = \begin{bmatrix} X_1^{-1} & X_2^{-1} & \dots & X_n^{-1} \end{bmatrix}$ → Unknown d x dn

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

Known dn x dn

Unknown dn x d

$$Z = XY$$

Z is low-rank: rank(Z)=d

The case of a complete graph

Known ← $Z = \begin{bmatrix} I_d & Z_{12} & \dots & Z_{1n} \\ Z_{21} & I_d & \dots & Z_{2n} \\ \dots & & & \dots \\ Z_{n1} & Z_{n2} & \dots & I_d \end{bmatrix}$ $X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix}$ → Unknown

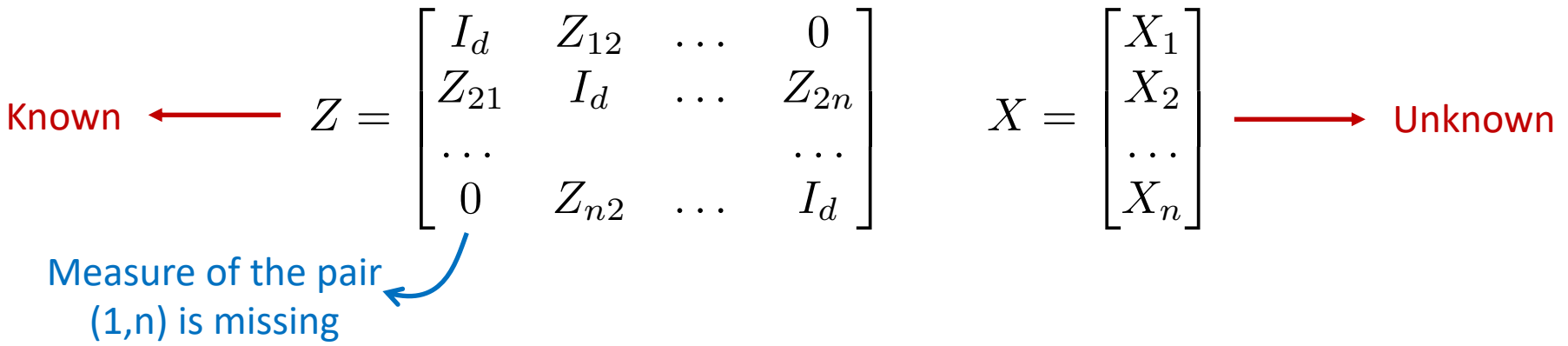
Proposition. *The columns of X are d eigenvectors of Z with eigenvalue n, where n is the number of nodes.*

Proof. $ZX = \underbrace{XY}_{\text{Consistency constraint}} X = X \underbrace{\sum_{i=1}^{n} (X_i^{-1} X_i)}_{X_i^{-1} X_i = I} = nX$ ➡ $\boxed{ZX = nX}$

A. Singer. *Angular synchronization by eigenvectors and semidefinite programming*. Applied and Computational Harmonic analysis (2011).

M. Arie-Nachimson, S. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, R. Basri. *Global motion estimation from point matches*. 3DIM/3DPVT (2012).

The case of missing data



👉 Missing measures translate into **zero blocks** in Z.

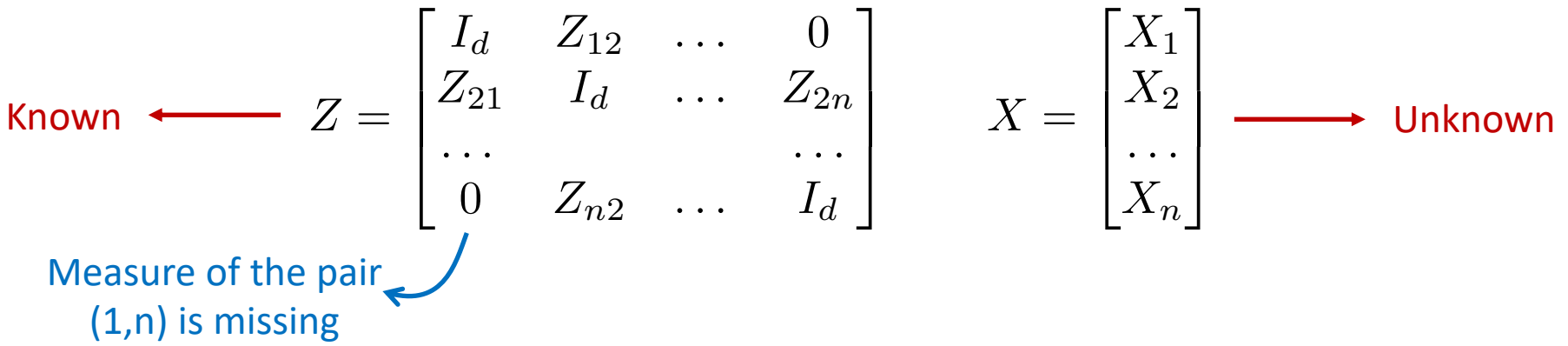
Proposition. *The unknown matrix X satisfies the following equalities.*

$$ZX = (D \otimes I_d)X \iff \underbrace{(D \otimes I_d)^{-1} Z}_{W} X = X$$

Degree matrix n x n

Scaled measurement matrix

The case of missing data



👉 Missing measures translate into **zero blocks** in Z.

Proposition. *The columns of X are d eigenvectors of the matrix* $W = (D \otimes I_d)^{-1} Z$ *with eigenvalue 1, which is the largest eigenvalue.*

$$WX = X$$

The case of missing data

🤔 Who is the matrix $D \otimes I_d$?

$$D = \begin{matrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{matrix} \qquad I_2 = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \qquad D \otimes I_2 = \begin{matrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{matrix}$$

Algorithm

The **spectral method** for synchronization can be detailed as follows:

$$W = (D \otimes I_d)^{-1} Z$$

$$Z = \begin{bmatrix} I_d & Z_{12} & \dots & 0 \\ Z_{21} & I_d & \dots & Z_{2n} \\ \dots & & & \dots \\ 0 & Z_{n2} & \dots & I_d \end{bmatrix} \quad \Rightarrow \text{Compute Eigenvectors} \quad U = \begin{bmatrix} U_1 \\ U_2 \\ \dots \\ U_n \end{bmatrix} \quad \Rightarrow \text{Fix group ambiguity} \quad X = UU_1^{-1} = \begin{bmatrix} I_d \\ U_2 U_1^{-1} \\ \dots \\ U_n U_1^{-1} \end{bmatrix}$$

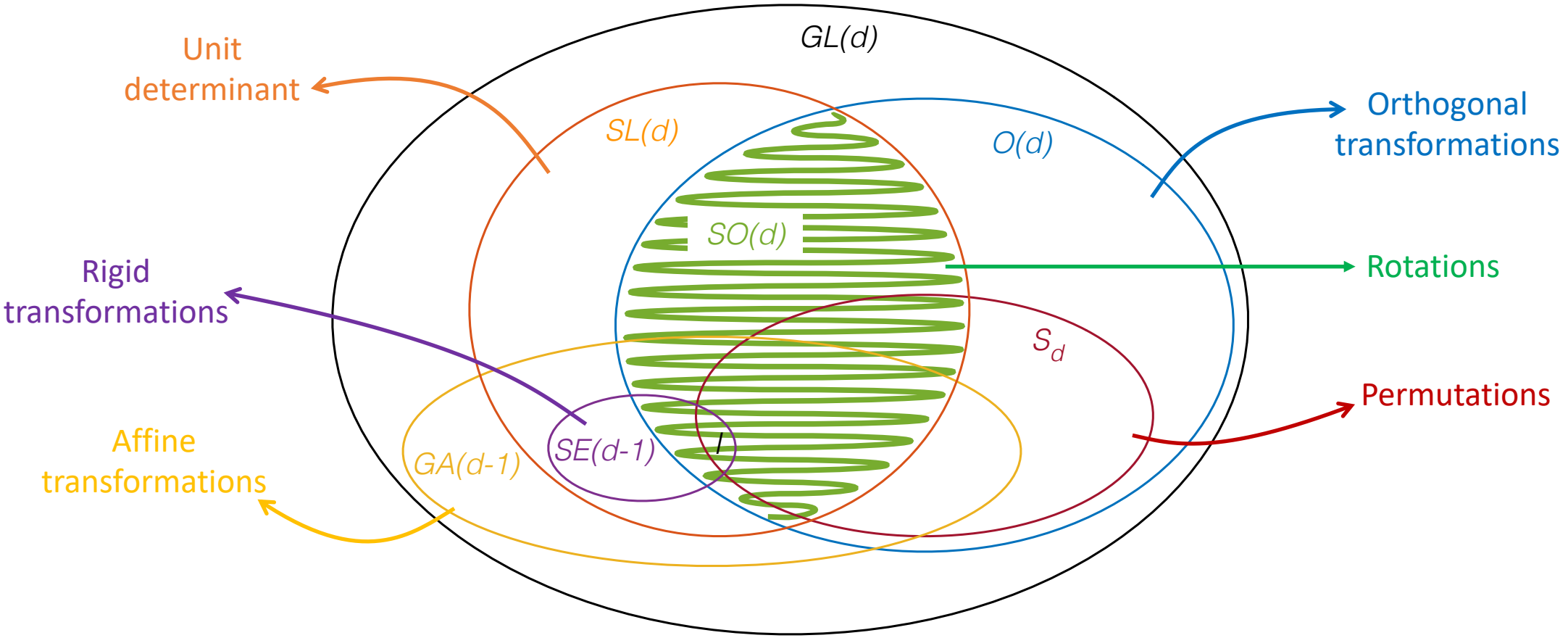
👉 We do not need to compute all the eigenvectors, but only the d **leading** ones.

👉 The scaled measurement matrix has size dn x dn: **sparse** eigen-solvers can be exploited (Matlab `eigs`) to manage large-scale scenarios with missing data.

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

Algorithm

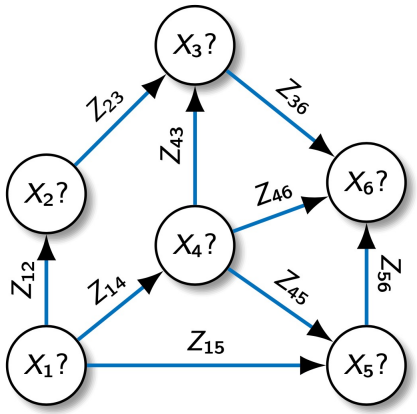
The spectral method works for any **subgroup** of GL(d).



Algorithm

The spectral method works for any **subgroup** of GL(d).

Input:
graph with observations



1

Store observations in a **block-matrix** where zero blocks correspond to missing measures.

$$Z = \begin{bmatrix} I_d & Z_{12} & \dots & 0 \\ Z_{21} & I_d & \dots & Z_{2n} \\ \dots & & & \dots \\ 0 & Z_{n2} & \dots & I_d \end{bmatrix}$$

2

Rescale the measurement matrix using the degree matrix of the graph.

$$W = (D \otimes I_d)^{-1} Z$$

3

Compute the d leading **eigenvectors.**

$$U = \begin{bmatrix} U_1 \\ U_2 \\ \dots \\ U_n \end{bmatrix}$$

4

Fix the group **ambiguity**: the first element is set to the identity.

$$V = \begin{bmatrix} I_d \\ V_2 \\ \dots \\ V_n \end{bmatrix}$$

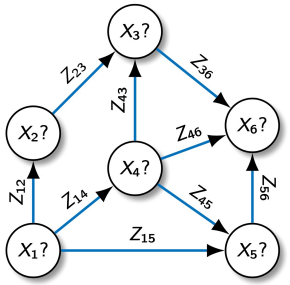
5

Project the solution onto the group.

$$X = \begin{bmatrix} I_d \\ X_2 \\ \dots \\ X_n \end{bmatrix}$$

Algorithm

Graph with observations on the edges



Compute eigenvectors

Use existing solvers (Matlab eigs)

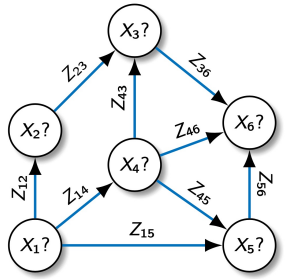
Fix ambiguity

Right-multiply by the inverse of the first block

Final projection

It depends on the group

Values on the nodes



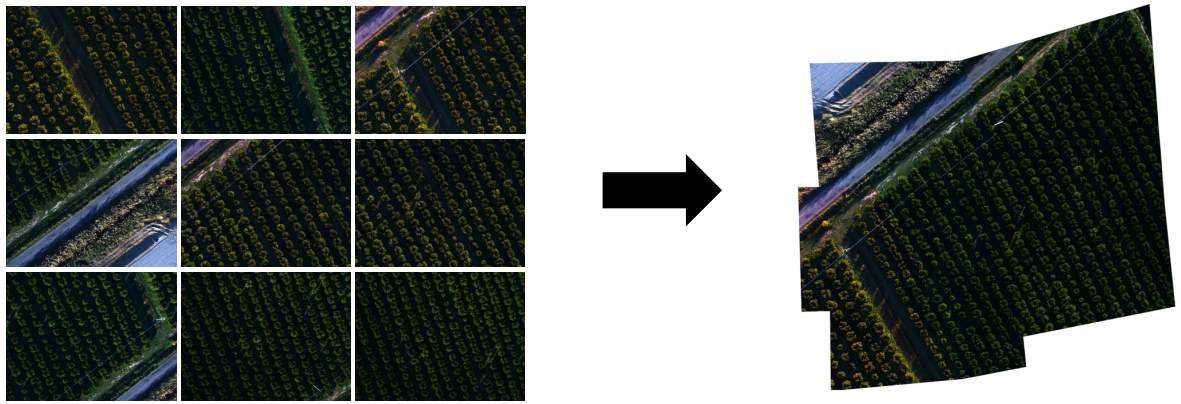
👉 At the end the solution needs to be projected onto the group: the spectral method does not enforce specific **constraints** of group elements.

Outline

- Background
- Introduction
- Spectral solution
- **Application to image mosaicking**
- Conclusion & extensions
- Matlab demo

Problem Formulation

Image mosaicking: the task is to align multiple images into a mosaic.



The underlying transformations are **homographies:**

- The scene is (approximately) planar
- The camera is rotating only

P. Schroeder, A. Bartoli, P. Georgel, N. Navab. *Closed- form solutions to multiple-view homography estimation*. WACV (2011).

Problem Formulation

🤔 How can we deal with the **scale ambiguity**?

$$Z_{ij} \simeq X_i X_j^{-1}$$

Equality up to scale

We need to properly choose a normalization:

- We would like to have a **strict equality**, in order to apply the synchronization framework (spectral solution).
- We need a normalization that preserves **group** properties (e.g. multiplication of two group elements belongs to the group).

Problem Formulation

Homographies can be conveniently represented as matrices with **unit determinant,** which form the **Special Linear Group** SL(3):

$$SL(3) = \{M \in \mathbb{R}^{3\times 3} \;\; \text{such that} \;\; \det(M) = 1\}$$

Fix scale ambiguity

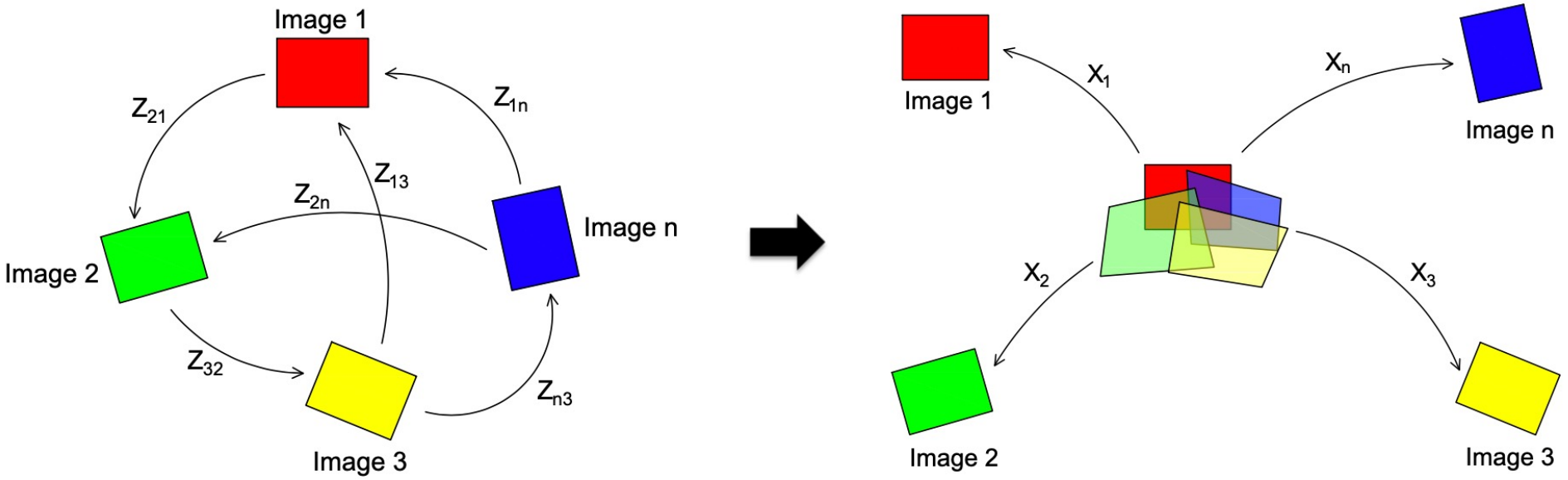
- Closure: $\forall A, B \in SL(3): \quad \det(AB) = \det(A)\det(B) = 1 \quad \Longrightarrow \quad AB \in SL(3)$
- Identity: $\det(I_3) = 1 \quad \Longrightarrow \quad I_3 \in SL(3)$
- Inverse: $\forall A \in SL(3): \quad \det(A^{-1}) = \det(A)^{-1} = 1 \quad \Longrightarrow \quad A^{-1} \in SL(3)$

Properties of determinant

Two-step Approach

The problem can be solved in **two steps**:

1. Align **pairs** of images in isolation;
2. Compute a **global** alignment starting from pairwise transformations.



Two-step Approach

The problem can be solved in **two steps:**

1. Align **pairs** of images in isolation;
2. Compute a **global** alignment starting from pairwise transformations.

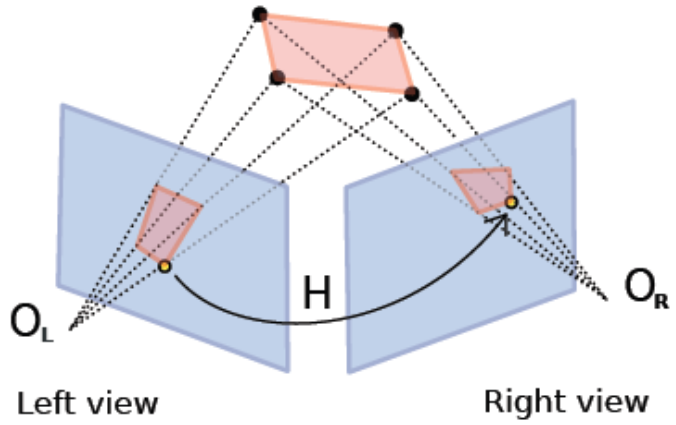
👉 **Remarks:**

- In Step 1 the transformations are **local/relative** whereas in Step 2 they are **global/absolute.**
- Correspondences are used in Step 1 only, but they are not used in Step 2.

Two-step Approach

The problem can be solved in **two steps:**

1. Align **pairs** of images in isolation;
2. Compute a **global** alignment starting from pairwise transformations.



STEP 1

Standard techniques can be used:

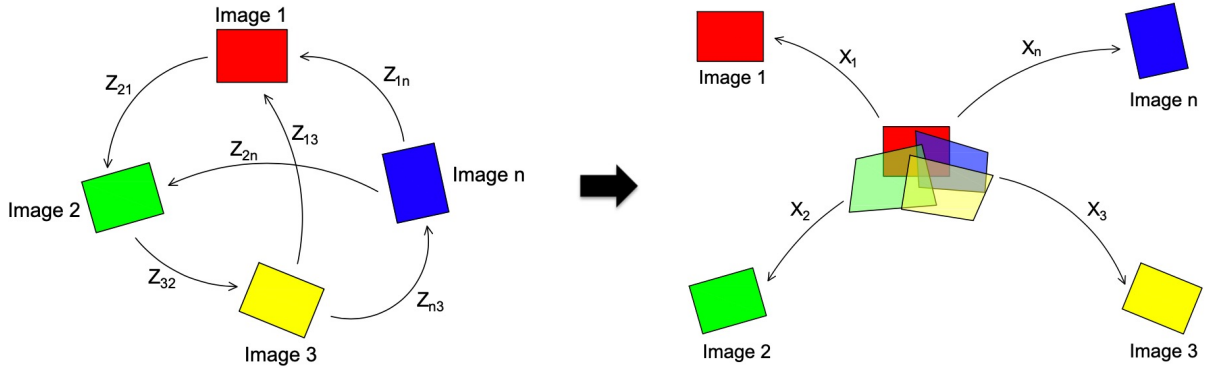
- Compute the **homography** from image point correspondences (DLT).
- Scale the homography to unit determinant, to get an element in SL(3).

R. Hartley, A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, second edition (2004)

Two-step Approach

The problem can be solved in **two steps**:

1. Align **pairs** of images in isolation;
2. Compute a **global** alignment starting from pairwise transformations.



STEP 2

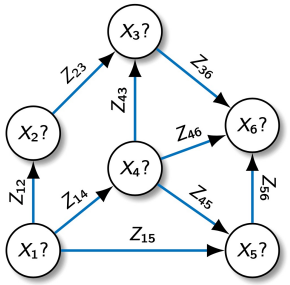
This is a synchronization problem in the Special Linear Group SL(3):

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

P. Schroeder, A. Bartoli, P. Georgel, N. Navab. *Closed- form solutions to multiple-view homography estimation*. WACV (2011).

Spectral Solution

Graph with observations on the edges



→ Compute eigenvectors

Use existing solvers (Matlab eigs)

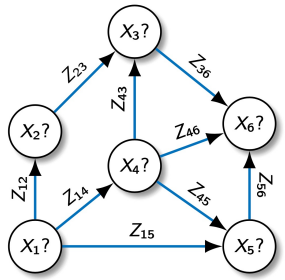
→ Fix ambiguity

Right-multiply by the inverse of the first block

→ Final projection

It depends on the group

→ Values on the nodes



👉 Projection can be done by dividing H by $\sqrt[3]{\det(H)}$, which has unit determinant.

In general: $\det(aU) = a^3 \det(U)$

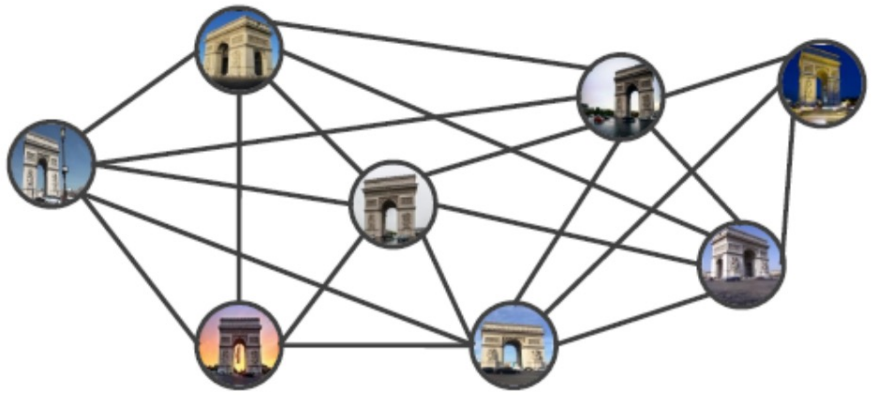
Outline

- Background
- Introduction
- Spectral solution
- Application to image mosaicking
- **Conclusion & extensions**
- Matlab demo

Extensions

There are many applications of synchronization in Computer Vision!

👉 The key structure to represent these problems is a **graph**.



👉 All the variables are elements of a **group** (rotations, permutations, …).

Extensions

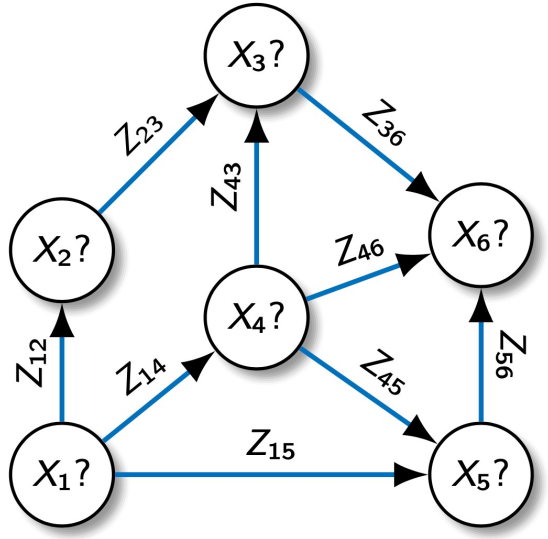
There are many applications of synchronization in Computer Vision!

🤔 *What changes between them?*

- Tool used to get measures on edges
- Group (matrices) where the problem is formulated → projection

🤔 *What remains the same?*

- Tool used to recover unknowns on the nodes → spectral method



Extensions

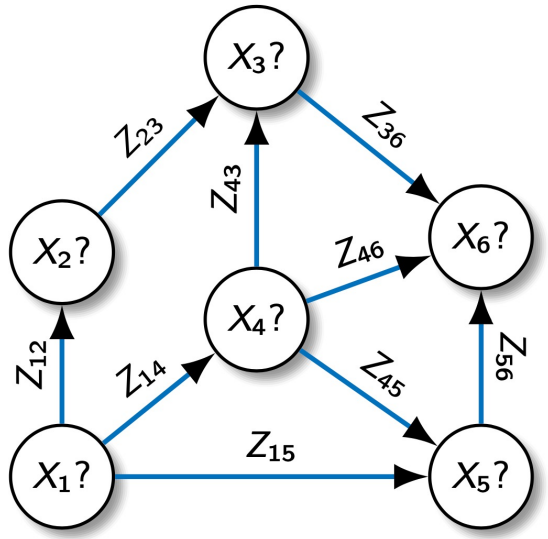
There are many applications of synchronization in Computer Vision!

🤔 *What changes between them?*

- Tool used to get measures on edges ⟶ Homography estimation
- Group (matrices) where the problem is formulated → projection
  - ⇣ Matrices with unit determinant (Special Linear Group)
  - ⇣ Scale to unit determinant

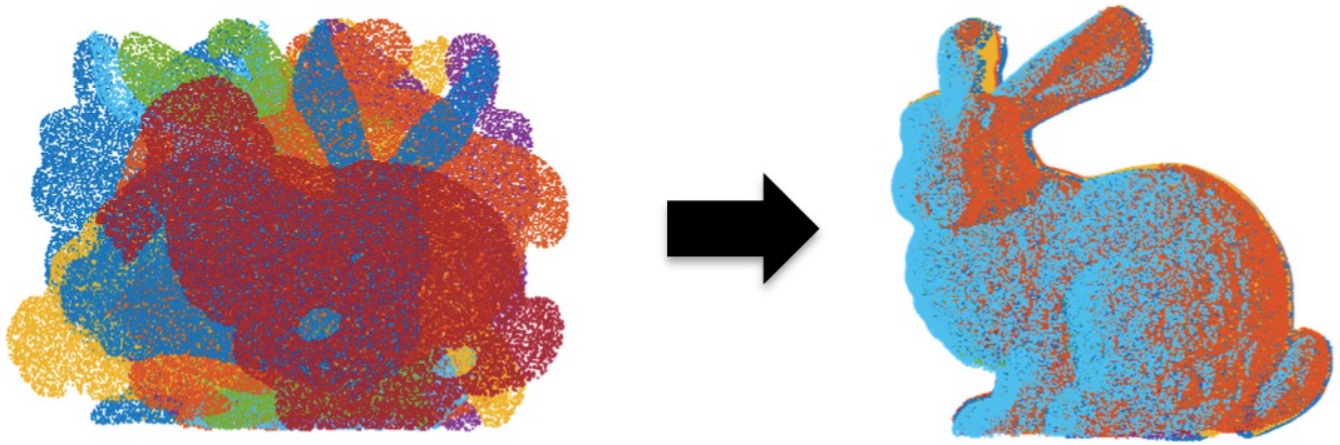
🤔 *What remains the same?*

- Tool used to recover unknowns on the nodes → spectral method



Other Scenarios
3D Registration

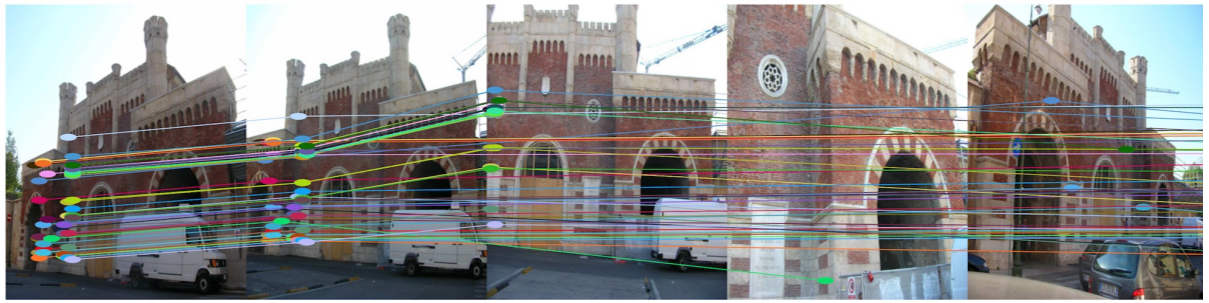
The task is to find the **rigid transformations** (rotations+translations) that bring multiple 3D point-clouds into alignment.

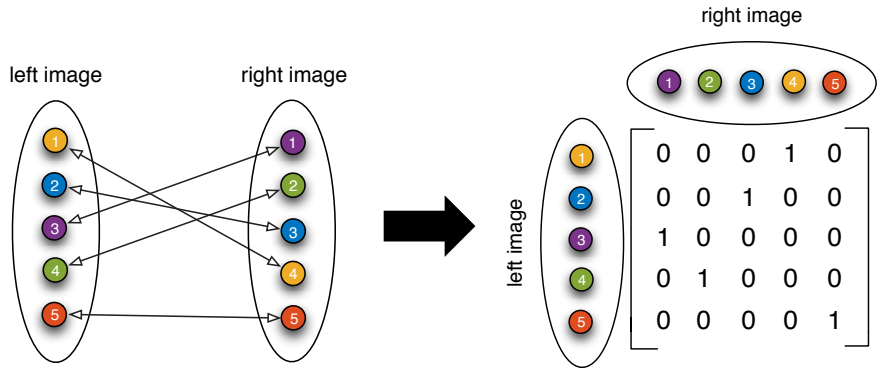


F. Arrigoni, B. Rossi, A. Fusiello. *Spectral synchronization of multiple views in SE(3)*. SIAM Journal on Imaging Sciences (2016)

Other Scenarios
Multi-image Matching

The task is to find point correspondences between multiple images.



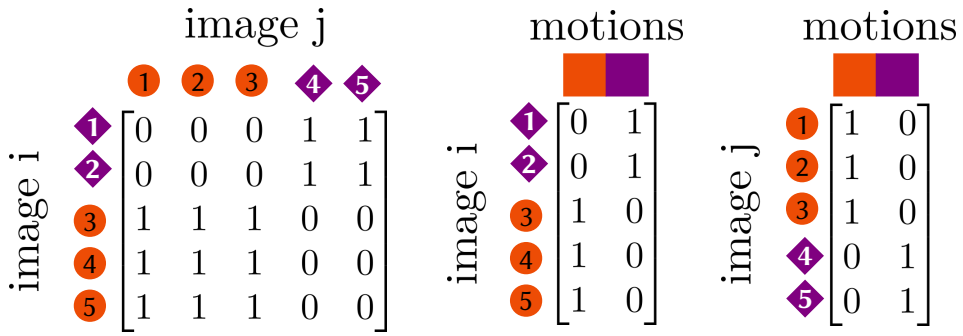


Matches can be represented as **permutation matrices**.

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

Other Scenarios
Motion Segmentation

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

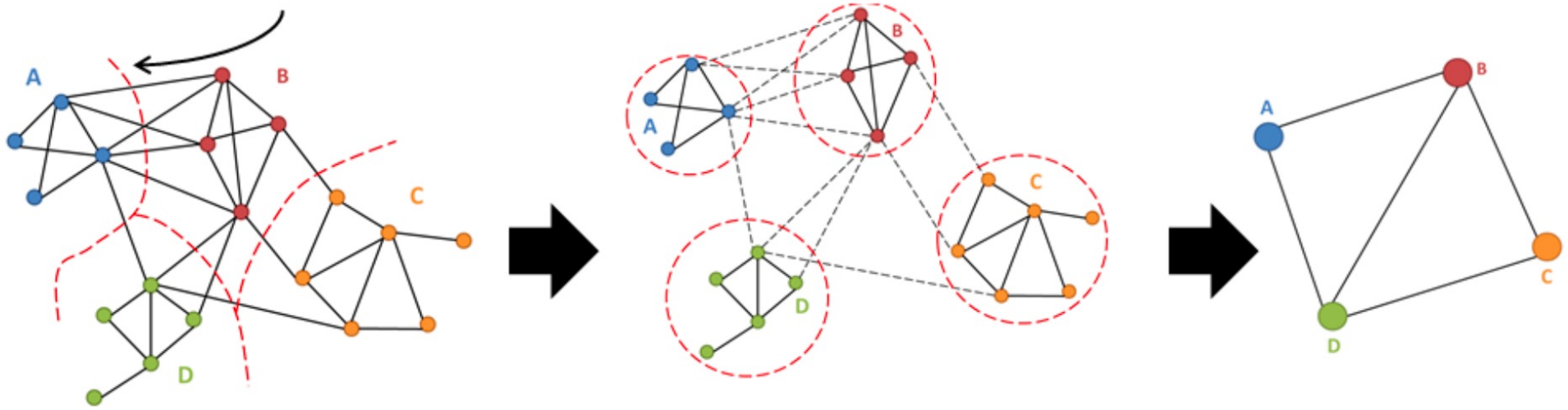


It can be viewed as a synchronization of **binary matrices**.

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

Other Scenarios
Partitioned Synchronization

A synchronization task is partitioned into subproblems for the sake of efficiency.



Each subgraph has its own local **ambiguity**: finding consistency between the local ambiguities is an example of synchronization.

Bhowmick, Patra, Chatterjee, Govindu, Banerjee. *Divide and conquer: Efficient large-scale structure from motion using graph partitioning.* ACCV (2014)

Concluding remarks
Spectral Solution

❌ The spectral method is an **approximate** solution that does not enforce geometric constraints.

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

✅ It is **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 (partial permutations/binary matrices).

👉 *There exist also other solutions, but they are specific for the chosen group.*

Concluding remarks
Synchronization

✅ The **two-step framework** permits to:

- exploit two-view tools, hence the problem is split into subproblems which are **easier** to solve (in closed-form sometimes);
- exploit a more **compact** representation as the problem is solved in frame-space (points are not considered after pairwise transformations);

❌ Results can be suboptimal as points are not considered.

✅ It promotes error compensation, being global.

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

Outline

- Background
- Introduction
- Spectral solution
- Application to image mosaicking
- Conclusion & extensions
- **Matlab demo**