Viewing Graph Solvability
in Structure from Motion

Federica Arrigoni

federica.arrigoni@polimi.it

POLITECNICO
MILANO 1863

ISPRS TC II online talk series – September 20, 2023

Outline

- Introduction
- Calibrated Case
- Uncalibrated Case
- Calibrated vs Uncalibrated
- Conclusion

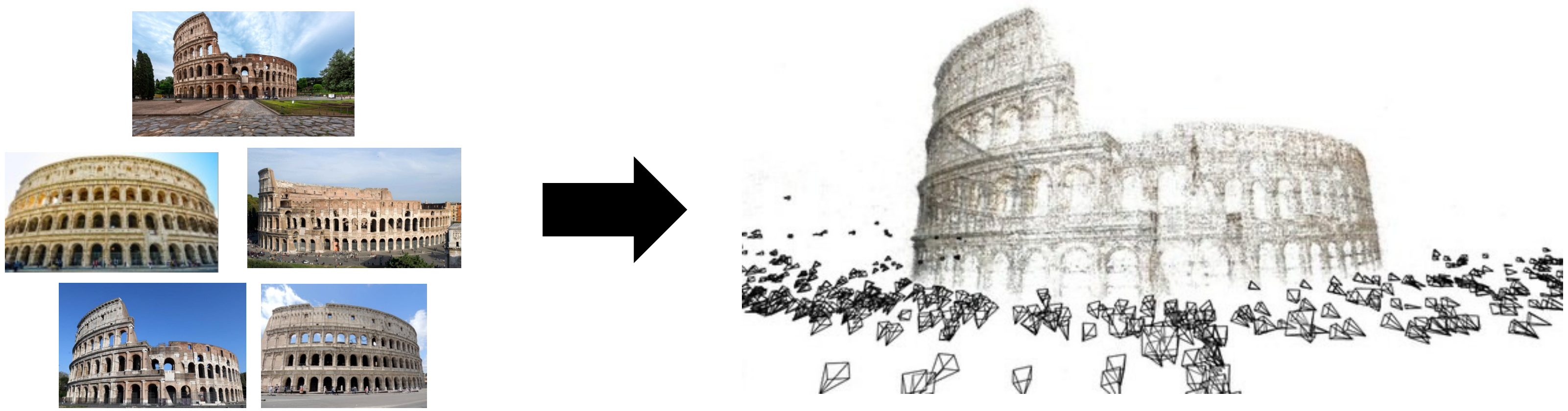
Outline

- **Introduction**
- Calibrated Case
- Uncalibrated Case
- Calibrated vs Uncalibrated
- Conclusion

Introduction

The goal of **structure from motion** (SfM) is to recover both camera motion and scene structure, starting from point correspondences in multiple images:

- camera motion = camera matrices/poses;
- scene structure = 3D coordinates of points.



O. Ozyesil, V. Voroninski, R. Basri, A. Singer. *A survey of structure from motion*. Acta Numerica (2017).

Introduction

Formally, the task is to compute **camera matrices** P_i and **coordinates of 3D points** M_j starting from image points m_{ij} such that the following equation is best satisfied:

$$\underbrace{m_{ij}}_{\substack{\text{Projection of}\\ \text{point j in image i}\\ \text{KNOWN}}} \simeq \underbrace{P_i}_{\substack{\text{Projection matrix}\\ \text{of camera i}\\ \text{UNKNOWN}}} \underbrace{M_j}_{\substack{\text{3D coordinates}\\ \text{of point j}\\ \text{UNKNOWN}}}$$

In the calibrated case, calibration matrices are known and projection matrices consist of **rotations** and **translations**: $P_i = \underbrace{K_i}_{\text{Known}} [\underbrace{R_i \ \mathbf{t}_i}_{\text{Unknown}}]$

Introduction

*Is 3D reconstruction **unique**?* 🤔

The solution is defined (at least) up to a global **projective transformation**:

$$m_{ij} \simeq P_i M_j = P_i \underbrace{QQ^{-1}}_{\text{identity}} M_j = \underbrace{P_i Q}_{\text{new cameras}} \underbrace{Q^{-1} M_j}_{\text{new points}}$$

If cameras are calibrated, then the reconstruction ambiguity is represented (at least) by a global **rotation, translation and scale.**

Introduction

The task of solvability is to analyse the **ambiguities** inherent to the SfM problem:

- single transformation → well-posed problem ✅
- multiple transformations → ill-posed problem ❌

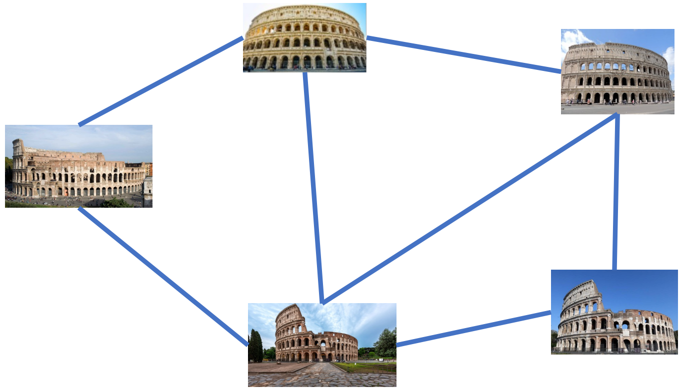
There are many ways to approach SfM! ⚠️

Here we focus on a framework that recovers **camera motion** from two-view relationships only (no points):

- Essential matrix (calibrated)
- Fundamental matrix (uncalibrated)

Introduction

The problem can be represented as a **viewing graph:**

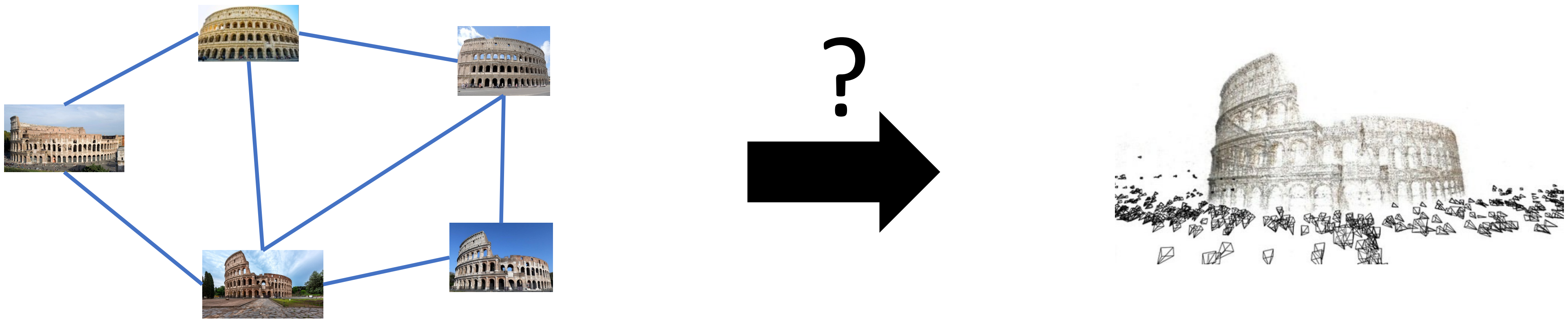


- Nodes = cameras/images
- Edges = two-view relations

Levi & Werman. *The viewing graph*. CVPR 2003.

Introduction

For which graphs do we have a **well-posed** problem? 🤔



✅ A graph is called **solvable** if and only if the available two-view relationships **uniquely** (up to a single transformation) determine the cameras → *unique solution*

❌ Otherwise it is called **non solvable** → *multiple (infinitely many) solutions*

Introduction

Here we focus on **solvability** only (*we do not address reconstruction*).

	Calibrated	Uncalibrated
Solvability	📕 Arrigoni & Fusiello. *Bearing-based network localizability: a unifying view.* IEEE TPAMI (2019).	📕 Levi & Werman. *The viewing graph*. CVPR 2003. 📕 Rudi, Pizzoli & Pirri. *Linear solvability in the viewing graph*. ACCV 2011. 📕 Trager, Osserman, & Ponce. *On the solvability of viewing graphs*. ECCV 2018. 📕 Arrigoni, Fusiello, Ricci & Pajdla. *Viewing graph solvability via cycle consistency.* ICCV (2021).
Reconstruction	📕 Ozyesil, Voroninski, Basri & Singer. *A survey of structure from motion*. Acta Numerica (2017).	📕 Kasten, Geifman, Galun & Basri. *GPSfM: global projective SfM using algebraic constraints on multi-view fundamental matrices*. CVPR (2019)

It is important to check solvability **before running SfM:**

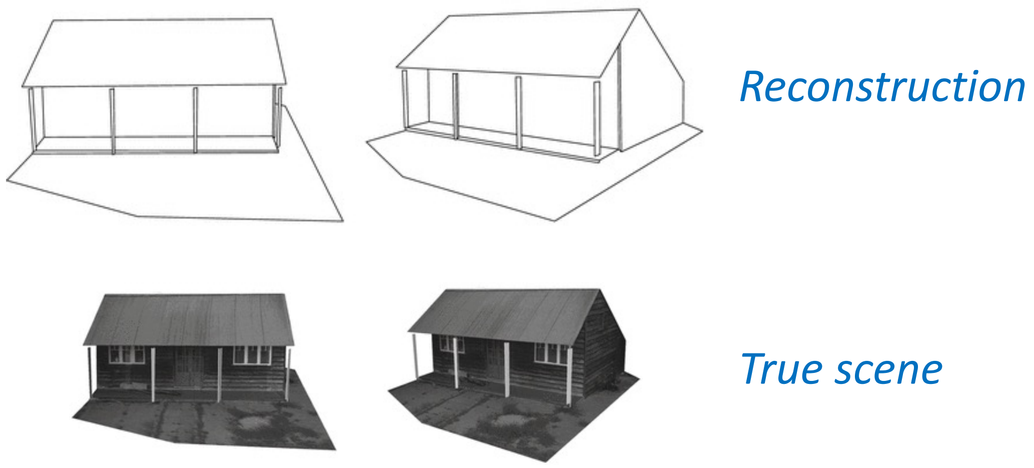
✅ If the graph is solvable, the SfM problem is well-posed.

❌ If the graph is not solvable, the problem is ill-posed: no method will return a useful solution.

Outline

- Introduction
- **Calibrated Case**
- Uncalibrated Case
- Calibrated vs Uncalibrated
- Conclusion

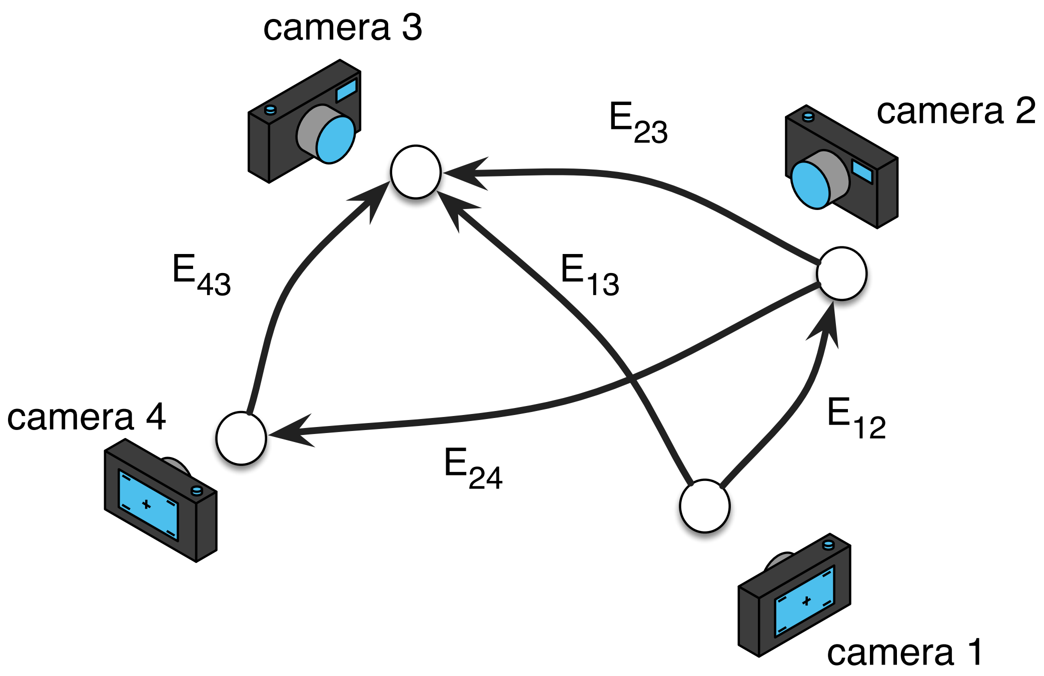
- Calibration matrix is required in advance
- Reconstruction is **metric** (up to scale)



The Calibrated Case

Problem Formulation

The **viewing graph** is a graph where vertices correspond to cameras and edges represent essential matrices.



Each essential matrix can be decomposed into:

- *Relative rotation* R_{ij}
- *Relative translation* t_{ij} *(known up to scale)*

The Calibrated Case

Problem Formulation

Solvable graph $\Leftrightarrow$ two-view transformations uniquely (up to a *single* rotation, translation & scale) determine the camera poses.

- We consider a **noiseless**-case
- We split the problem into **rotation and translation**:

$$\begin{aligned} R_{ij} &= R_i R_j^{\mathsf{T}} \\ \mathbf{t}_{ij} &= -R_i R_j^{\mathsf{T}} \mathbf{t}_j + \mathbf{t}_i \end{aligned} \iff \begin{aligned} R_{ij} &= R_i R_j^{\mathsf{T}} \\ \underbrace{-R_i^{\mathsf{T}} \mathbf{t}_{ij}}_{\mathbf{z}_{ij}} &= \underbrace{-R_i^{\mathsf{T}} \mathbf{t}_i}_{\mathbf{x}_i} + \underbrace{R_j^{\mathsf{T}} \mathbf{t}_j}_{-\mathbf{x}_j} \end{aligned}$$

Relative displacement

Centre of camera i

Centre of camera j

Consistency constraint between relative and absolute poses

The Calibrated Case
Problem Formulation

Solvable graph $\Longleftrightarrow$ two-view transformations uniquely (up to a *single* rotation, translation & scale) determine the camera poses.

- We consider a **noiseless**-case
- We split the problem into **rotation and translation**:

$$\begin{aligned} R_{ij} &= R_i R_j^{\mathsf{T}} \\ \mathbf{t}_{ij} &= -R_i R_j^{\mathsf{T}} \mathbf{t}_j + \mathbf{t}_i \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} R_{ij} &= R_i R_j^{\mathsf{T}} \\ \underbrace{-R_i^{\mathsf{T}} \mathbf{t}_{ij}}_{\mathbf{z}_{ij}} &= \underbrace{-R_i^{\mathsf{T}} \mathbf{t}_i}_{\mathbf{x}_i} + \underbrace{R_j^{\mathsf{T}} \mathbf{t}_j}_{-\mathbf{x}_j} \end{aligned}$$

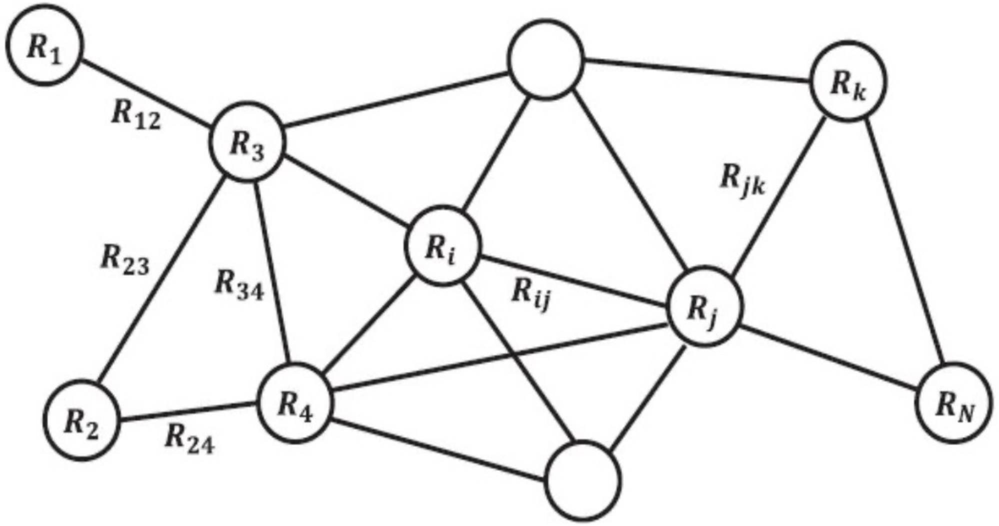
Consistency constraint between relative and absolute poses

👉 The magnitude of relative translations are **unknown**: $||\mathbf{t}_{ij}|| = ||\mathbf{z}_{ij}|| = ?$

The Calibrated Case

Rotations

In which cases can we uniquely (up to a global rotation) *recover camera rotations starting from relative rotations?* 🤔



Given a **spanning tree,** a solution can be found by setting the root to the identity and propagating the consistency constraint:

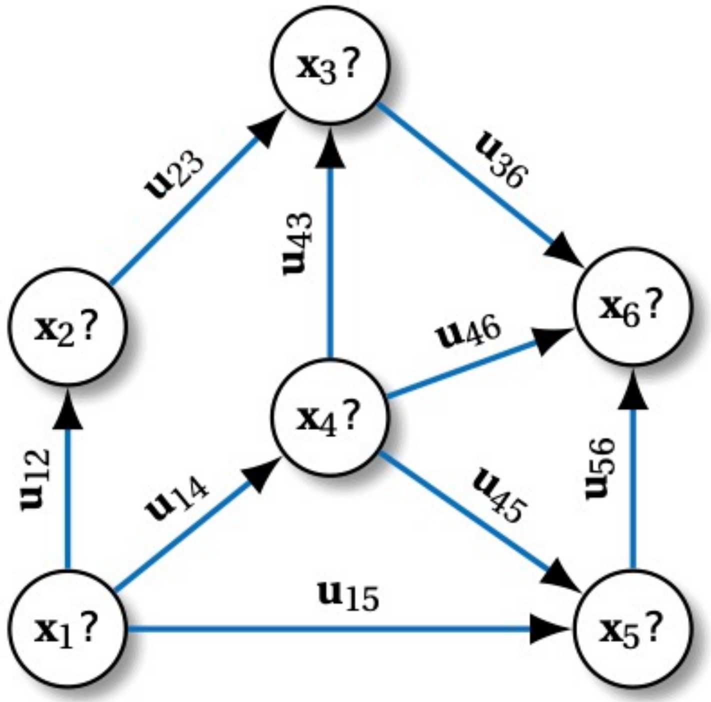
$$R_i = R_{ij} R_j \iff R_{ij} = R_i R_j^T$$

Solvability for rotations $\iff$ **connected** viewing graph

The Calibrated Case

Translations

In which cases can we uniquely (up to translation & scale) *recover camera positions from pairwise directions?*



- **Nodes** = unknown locations
- **Edges** = known directions

$$\mathbf{u}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{||\mathbf{x}_i - \mathbf{x}_j||} \iff \mathbf{u}_{ij} \times (\mathbf{x}_i - \mathbf{x}_j) = 0$$

*A solution can be found from the direction constraint, which is a **linear** equation!*

The Calibrated Case

Translations

Theorem. *A graph is solvable if and only if* rank(S)=3n-4

Localization Equation: Sx=0

Translation & scale ambiguity

✅ If the viewing graph is **solvable**, then the problem is well-posed.

❌ Otherwise, the problem is ill-posed: the **largest solvable component** has to be extracted ⟺ clustering rows in the null-space of S

📕 F. Arrigoni, A. Fusiello. *Bearing-based network localizability: a unifying view.* IEEE TPAMI (2019).

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

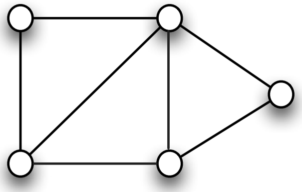
📕 R. Kennedy, K. Daniilidis, O. Naroditsky, C. J. Taylor. *Identifying maximal rigid components in bearing-based localization*. IROS (2012)

The Calibrated Case

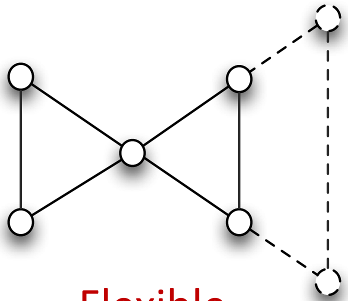
Translations

Solvability for translations $\Leftrightarrow$ **parallel rigid** viewing graph

Definition. *A graph is **parallel rigid** when all the configurations with parallel edges differ by translation and scale. Otherwise it is called **flexible**.*



Parallel rigid



Flexible

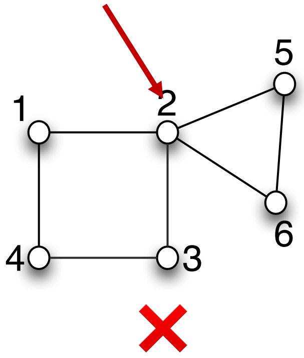
This is a well studied task! 😄

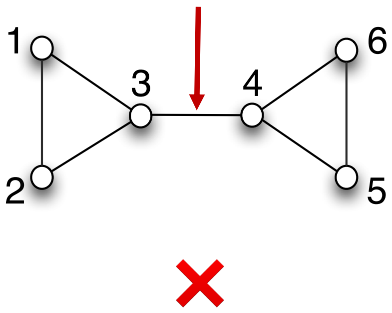
O. Ozyesil, A. Singer. *Robust camera location estimation by convex programming.* CVPR (2015).

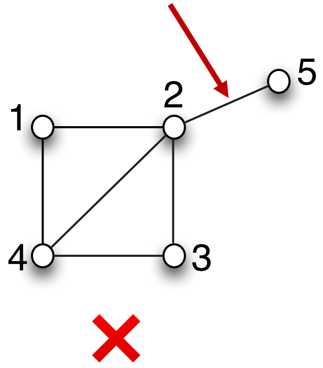
The Calibrated Case
Translations

A parallel-rigid graph must satisfy the following **necessary conditions**:

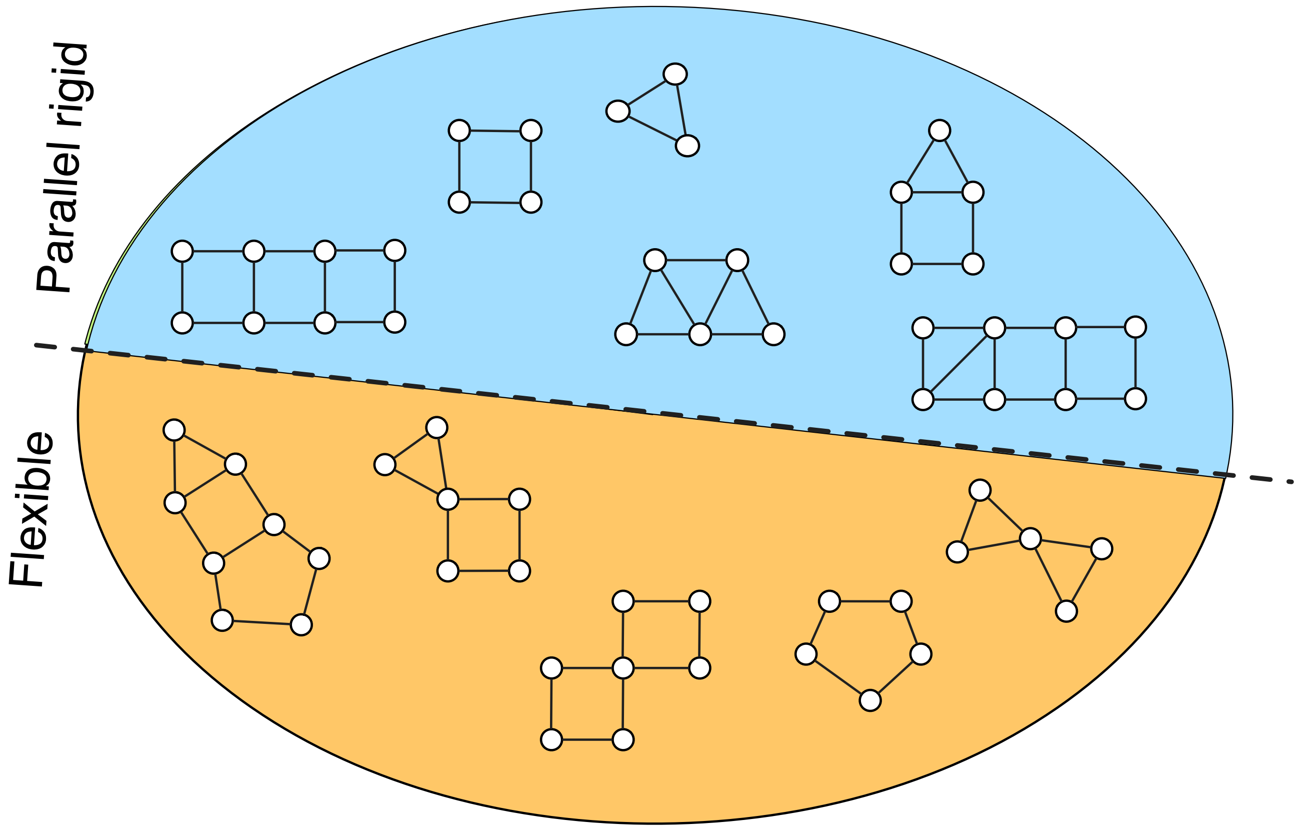
- it has at least **(3n-4)/2 edges**
- It is **bridgeless** (i.e., it remains connected after removing any edge).
- It is **biconnected** (i.e. it does not have **articulation points** meaning that it remains connected after removing any node).







The Calibrated Case
Examples



- A single cycle of **length 3 or 4** is parallel rigid, whereas longer cycles are flexible
- Union of rigid graphs with a common edge is also rigid ⟹ **sufficient conditions**

The Calibrated Case
Examples

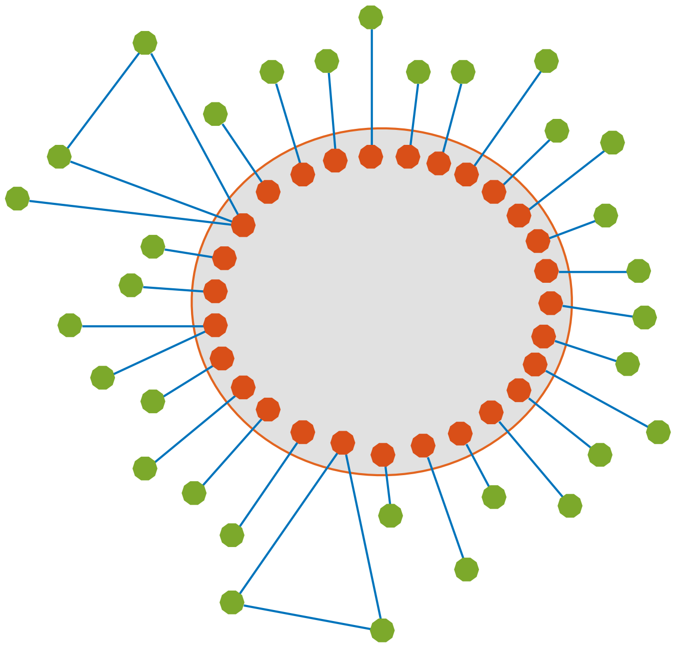
Dataset	nodes	% edges	rigid	articulation	bridges
Arts Quad	5530	2	✗	30	10
Piccadilly	2508	10	✗	59	62
Roman Forum	1134	11	✗	28	28
Union Square	930	6	✗	60	68
Vienna Cathedral	918	25	✗	19	20
Alamo	627	50	✗	17	19
Notre Dame	553	68	✓	–	–
Tower of London	508	19	✗	19	19
Montreal N. Dame	474	47	✗	7	7
Yorkminster	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

Cornell ArtsQuad http://vision.soic.indiana.edu/projects/disco/

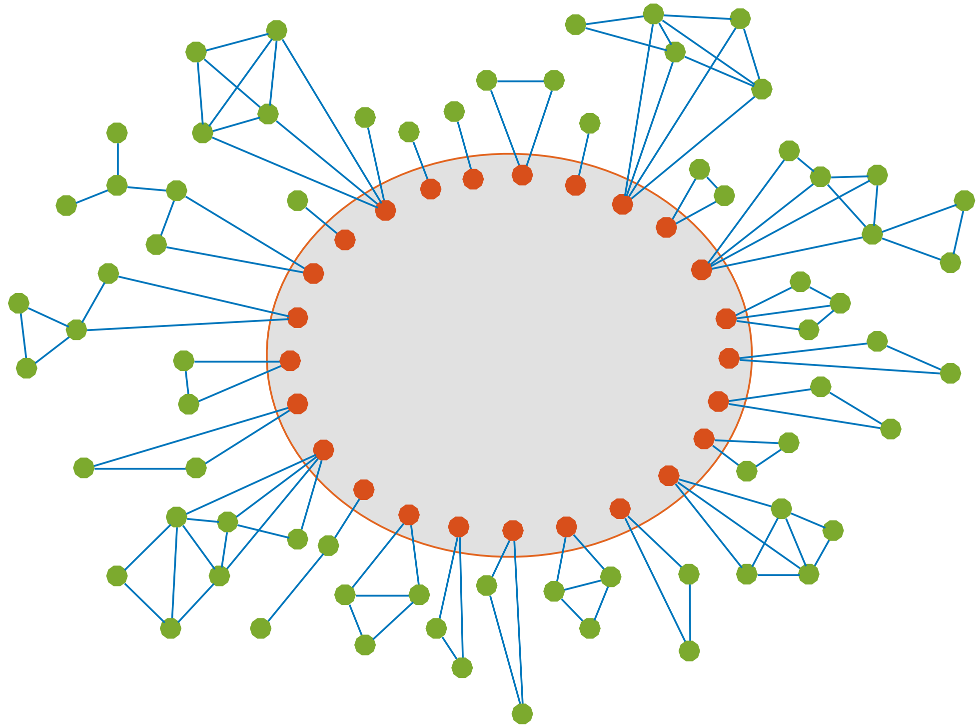
1DSfM datasets http://www.cs.cornell.edu/projects/1dsfm/

The Calibrated Case
Examples

Simplified representation: edges outside the largest rigid component are drawn.



Roman Forum



Arts Quad

The Calibrated Case

Summary

Solvability for rotations $\Longleftrightarrow$ **connected** viewing graph

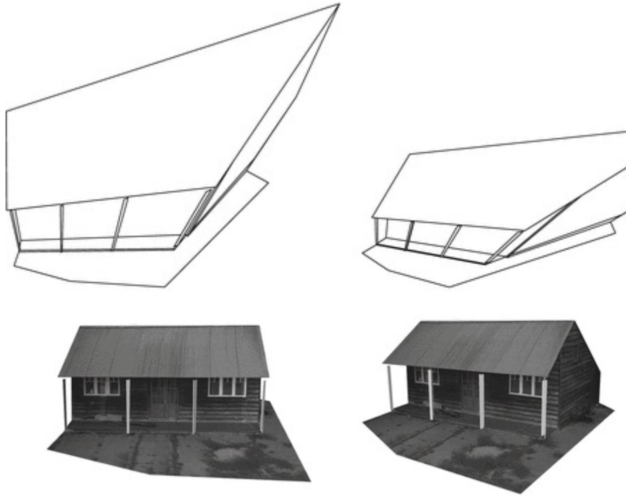
Solvability for translations $\Longleftrightarrow$ **parallel rigid** viewing graph

- Parallel rigidity can be tested from the rank of a **linear system.**
- Maximal components can be extracted from the **null-space** of such a system.
- **Large-scale** datasets can be processed. 😄

Outline

- Introduction
- Calibrated Case
- **Uncalibrated Case**
- Calibrated vs Uncalibrated
- Conclusion

- No assumptions
- Reconstruction is **projective**

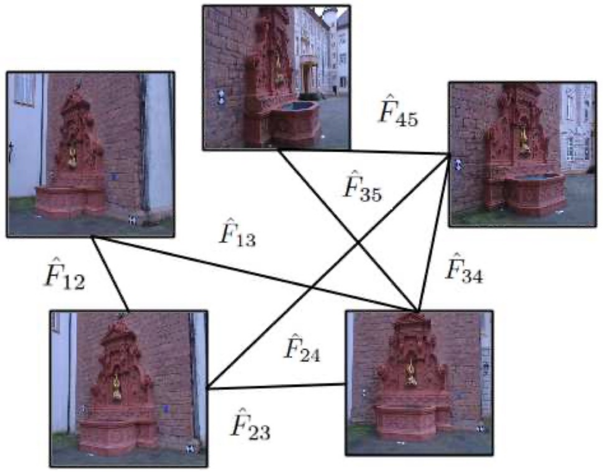


Reconstruction

True scene

The Uncalibrated Case
Problem Formulation

The **viewing graph** is a graph where vertices correspond to cameras and edges represent fundamental matrices.

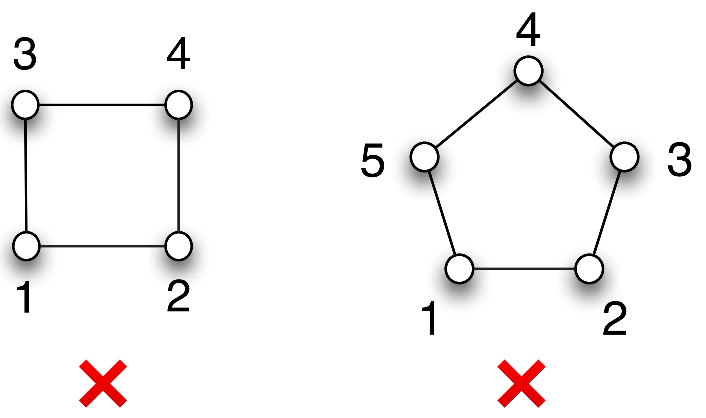


- Solvability depends on the **graph** and **camera centres** only.
- It can be reduced to a property of the graph only if we assume **generic** centres.

Solvable graph $\Leftrightarrow$ it uniquely (up to a *single* projective transformation) determines a projective configuration of cameras.

The Uncalibrated Case
Necessary Conditions

- A solvable graph has **at least (11n-15)/7 edges**.
- In a solvable graph, **all the vertices have degree at least two and no two adjacent vertices have degree two** (if $n > 3$).

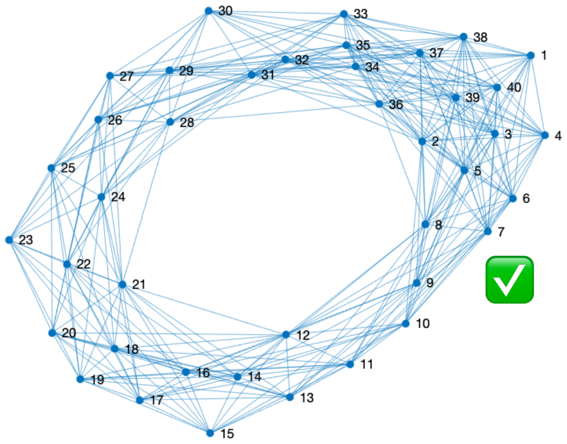


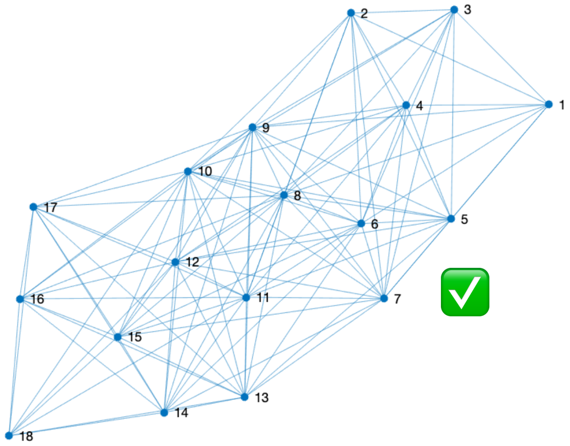
M. Trager, B. Osserman, and J. Ponce. *On the solvability of viewing graphs*. ECCV 2018.

N. Levi and M. Werman. *The viewing graph*. CVPR 2003

The Uncalibrated Case
Sufficient Conditions

- **Triangulated** graphs are solvable
- **Constructive** approaches are also available





📕 M. Trager, M. Hebert, and J. Ponce. *The joint image hand-book*. ICCV 2015.

📕 A. Rudi, M. Pizzoli, and F. Pirri. *Linear solvability in the viewing graph*. ACCV 2011.

The Uncalibrated Case
Algebraic Characterization

Idea: *characterize the set of projective transformations that represent all possible ambiguities of the problem.*

First, let us identify the family of transformations that leave a **single camera** fixed.

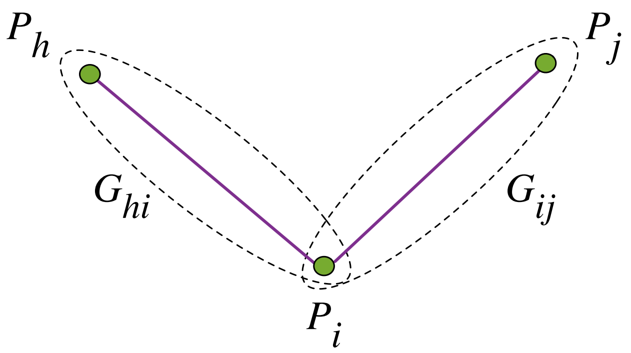
Proposition. *Let P be a camera with centre c. All the solutions to* $PG = aP$ for $G \in GL(4,\mathbb{R})$ and $a \in \mathbb{R}_{\neq 0}$ *are given by* $\boxed{G = aI_4 + \mathbf{c}\mathbf{v}^{\mathsf{T}}}$ $\forall a \in \mathbb{R}_{\neq 0}, \mathbf{v} \in \mathbb{R}^4$

M. Trager, B. Osserman, and J. Ponce. *On the solvability of viewing graphs*. ECCV 2018.

The Uncalibrated Case
Algebraic Characterization

*What happens when we have **multiple cameras**, represented as a viewing graph?* 🤔

Let us assign an unknown projective transformation G_{ij} to every edge, and let us consider two edges (h,i) and (i,j) with a common vertex i.



Compatibility Condition

$$G_{hi}G_{ij}^{-1} = a_{hij} I_4 + \mathbf{c}_i \mathbf{v}_{hij}^{\mathsf{T}}$$

$G_{hij} \in GL(4)$ is unknown

$a_{hij} \in \mathbb{R}_{\neq 0}$ and $\mathbf{v}_{hij} \in \mathbb{R}^4$ are unknown

$\mathbf{c}_i \in \mathbb{R}^4$ is known (camera center)

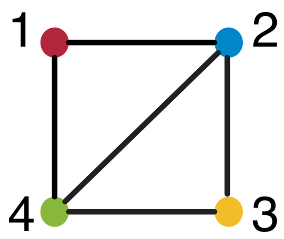
Solvable graph $\Leftrightarrow$ $G_{ij} = s_{ij}\, H$ → **Single** projective transformation

M. Trager, B. Osserman, and J. Ponce. *On the solvability of viewing graphs*. ECCV 2018.

The Uncalibrated Case
Algebraic Characterization

Input Graph

$$G_{hi} G_{ij}^{-1} = a_{hij} I_4 + \mathbf{c}_i \mathbf{v}_{hij}^{\mathsf{T}}$$



- **Polynomial** system of equations with many unknowns

$$G_{hi} \in GL(4) \text{ is unknown}$$

$$a_{hij} \in \mathbb{R}_{\neq 0} \text{ and } \mathbf{v}_{hij} \in \mathbb{R}^4 \text{ are unknown}$$

$$\mathbf{c}_i \in \mathbb{R}^4 \text{ is known (camera center)}$$

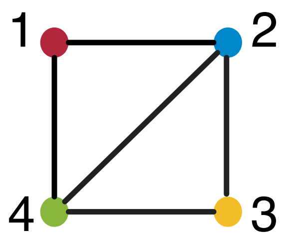
$$(h, i) \text{ and } (i, j) \text{ are adjacent edges}$$

M. Trager, B. Osserman, and J. Ponce. *On the solvability of viewing graphs*. ECCV 2018.

The Uncalibrated Case
Reduced Formulation

Input Graph

$$G_{hi} G_{ij}^{-1} = a_{hij} I_4 + \mathbf{c}_i \mathbf{v}_{hij}^{\mathsf{T}}$$



- **Polynomial** system of equations with many unknowns

$$G_{hi} \in GL(4) \text{ is unknown}$$

$$\boxed{a_{hij} \in \mathbb{R}_{\neq 0} \text{ and } \mathbf{v}_{hij} \in \mathbb{R}^4 \text{ are unknown}}$$

$$\mathbf{c}_i \in \mathbb{R}^4 \text{ is known (camera center)}$$

$$(h, i) \text{ and } (i, j) \text{ are adjacent edges}$$

M. Trager, B. Osserman, and J. Ponce. *On the solvability of viewing graphs*. ECCV 2018.

- It is possible **eliminate variables** 😄

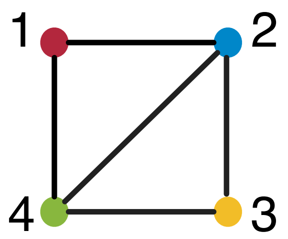
Arrigoni, Fusiello, Ricci & Pajdla. *Viewing graph solvability via cycle consistency.* ICCV (2021).

The Uncalibrated Case

Reduced Formulation

Input Graph

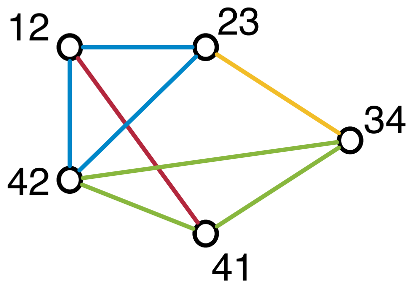
$$G_{hi}G_{ij}^{-1} = a_{hij}I_4 + \mathbf{c}_i\mathbf{v}_{hij}^{\mathsf{T}}$$



Line Graph

$$G_{\tau}G_{\upsilon}^{-1} = Z_{\tau\upsilon}$$

$$Z_{\tau\upsilon} = a_{\tau\upsilon}I_4 + \mathbf{c}_i\mathbf{v}_{\tau\upsilon}^{\mathsf{T}}$$



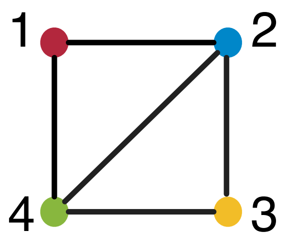
- Each node is an edge in the input graph;
- Two nodes are linked if the corresponding edges are adjacent in the input graph.
- There is one equation for each edge in the line graph.

The Uncalibrated Case

Reduced Formulation

Input Graph

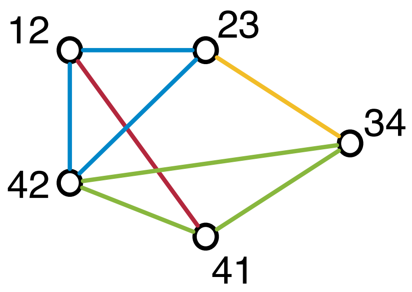
$$G_{hi}G_{ij}^{-1} = a_{hij}I_4 + \mathbf{c}_i\mathbf{v}_{hij}^{\mathsf{T}}$$



Line Graph

$$G_{\tau}G_{\upsilon}^{-1} = Z_{\tau\upsilon}$$

$$Z_{\tau\upsilon} = a_{\tau\upsilon}I_4 + \mathbf{c}_i\mathbf{v}_{\tau\upsilon}^{\mathsf{T}}$$



- Each node is an edge in the input graph;
- Two nodes are linked if the corresponding edges are adjacent in the input graph.
- There is one equation for each edge in the line graph.

How can we **eliminate the G variables?**

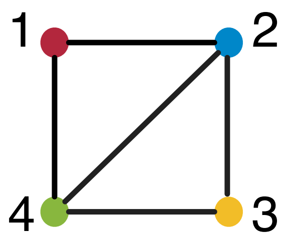
Idea: $$Z_{12,23} \cdot Z_{23,42} \cdot Z_{42,12} = G_{12}\underbrace{G_{23}^{-1}G_{23}}_{I}\underbrace{G_{42}^{-1}G_{42}}_{I}G_{12}^{-1} = I$$

The Uncalibrated Case

Reduced Formulation

Input Graph

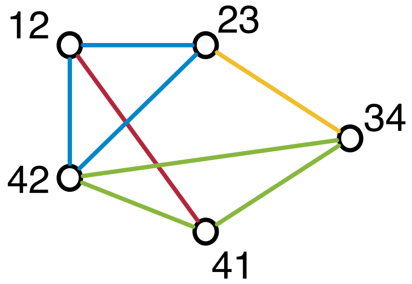
$$G_{hi}G_{ij}^{-1} = a_{hij}I_4 + \mathbf{c}_i\mathbf{v}_{hij}^{\mathsf{T}}$$



Line Graph

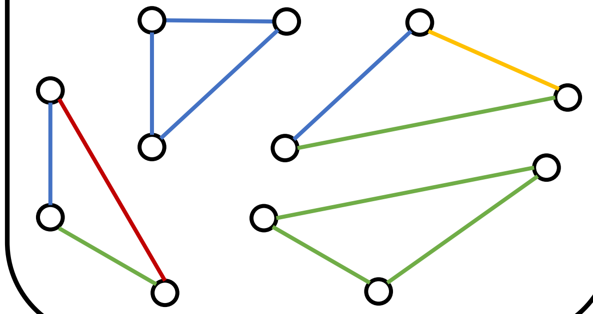
$$G_\tau G_\upsilon^{-1} = Z_{\tau\upsilon}$$

$$Z_{\tau\upsilon} = a_{\tau\upsilon}I_4 + \mathbf{c}_i\mathbf{v}_{\tau\upsilon}^{\mathsf{T}}$$



Cycle Consistency

$$Z_{\tau_1\tau_2}Z_{\tau_2\tau_3}\cdot\cdots\cdot Z_{\tau_\ell\tau_1} = I_4$$



cycle consistency (on all cycles) $\Longleftrightarrow$ cycle consistency (on a basis)

The Uncalibrated Case

Algorithm

Algorithm 1 Viewing Graph Solvability

Input: undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Output: solvable or not solvable

1. randomly sample the camera centres
2. compute the line graph $\mathcal{L}(\mathcal{G})$
3. compute a cycle consistency basis for $\mathcal{L}(\mathcal{G})$
4. set up equations
5. compute the number s of real solutions

if $s = 1$ **then**
  solvable ✅
else
  not solvable ❌
end if

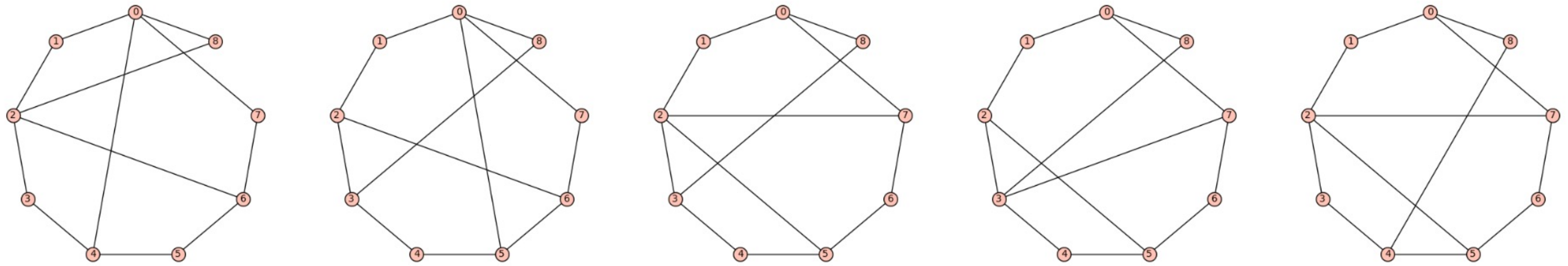
Gröbner basis
(symbolic computation)

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

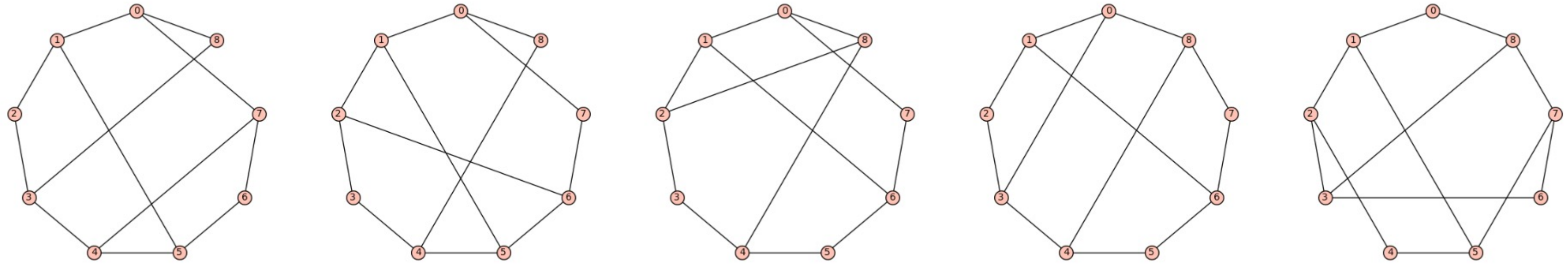
The Uncalibrated Case

Examples

Minimal viewing graphs with 9 vertices



Solvable ✅

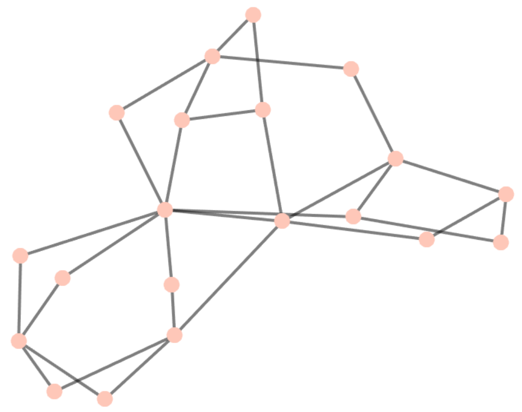


Not solvable ❌

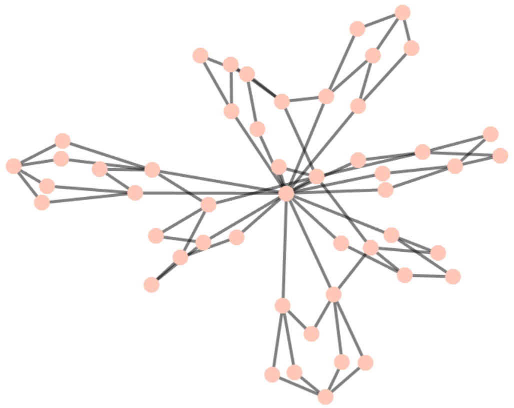
The Uncalibrated Case
Examples

Execution times on minimal graphs

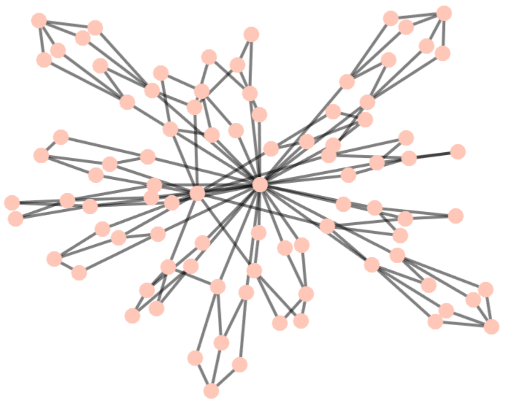
Nodes	10	20	30	40	50	60	70	80	90
Time	1.6 s	9 s	93 s	3 min	15 min	35 min	1 h	≈ 2 h	> 4 h



Solvable graph with 20 nodes



Solvable graph with 50 nodes

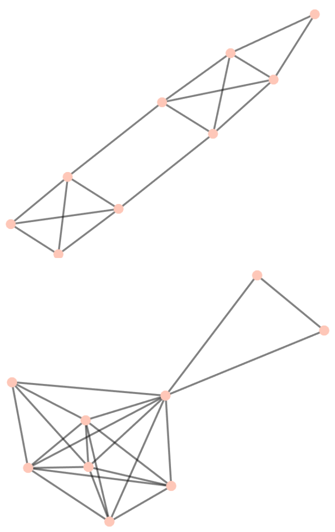


Solvable graph with 90 nodes

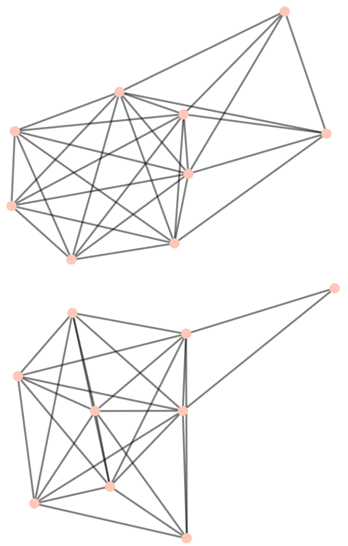
The Uncalibrated Case

Examples

Subgraphs with 9 nodes sampled from **real structure-from-motion viewgraphs**



Unsolvable

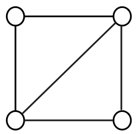
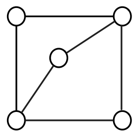
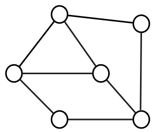
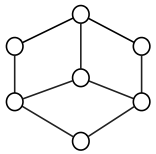
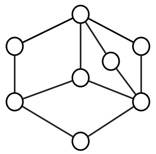
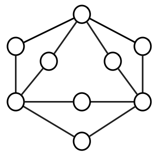


Solvable

	Solvable			Unsolvable		
Data set	by suff.	by Alg. 1	Tot.	by nec.	by Alg. 1	Tot.
Alcatraz Courtyard	200	0	200	0	0	0
Buddah Tooth	178	20	198	2	0	2
Pumpkin	169	22	191	8	1	9
Skansen Kronan	179	8	187	13	0	13
Tsar Nikolai I	196	0	196	4	0	4
Alamo	136	16	152	48	0	48
Ellis Island	136	30	166	34	0	34
Gendarmenmarkt	128	11	139	61	0	61
Madrid Metropolis	88	28	116	84	0	84
Montreal Notre Dame	140	12	152	48	0	48
Notre Dame	165	18	183	17	0	17
NYC Library	110	19	129	71	0	71
Piazza del Popolo	105	22	127	73	0	73
Piccadilly	109	23	132	68	0	68
Roman Forum	114	28	142	58	0	58
Tower of London	123	18	141	59	0	59
Trafalgar	86	16	102	98	0	98
Union Square	74	19	93	107	0	107
Vienna Cathedral	122	8	130	70	0	70
Yorkminster	116	14	130	70	0	70
Cornell Arts Quad	76	23	99	101	0	101

The Uncalibrated Case
Summary

- Thanks to cycle consistency, **less unknowns** are involved than previous work:

												
	#Eq.	#Var.	#Eq.	#Var.	#Eq.	#Var.	#Eq.	#Var.	#Eq.	#Var.	#Eq.	#Var.
Our formulation	64	36	64	40	112	63	112	67	192	100	208	109
Trager et al.	128	120	144	141	224	198	240	219	352	286	384	312

- It is possible to classify **previously undecided** viewing graphs and extend solvability testing up to minimal graphs **with 90 nodes.**
- *Larger/denser graphs can not be processed* ☹️

Outline

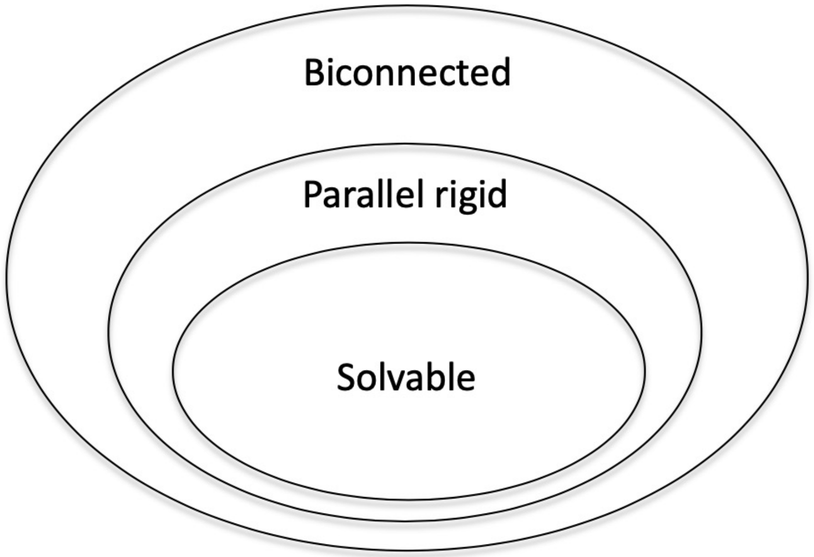
- Introduction
- Calibrated Case
- Uncalibrated Case
- **Calibrated vs Uncalibrated**
- Conclusion

🤔

Uncalibrated solvability $\Longrightarrow$ calibrated solvability

Calibrated vs Uncalibrated

Proposition. *A solvable (uncalibrated) graph is parallel rigid.*



Expected result! 😅

Well-posed with uncalibrated cameras
$\Longrightarrow$ well-posed with calibrated cameras

📕 Arrigoni, Fusiello, Rizzi, Ricci & Pajdla. *Revisiting viewing graph solvability: an effective approach based on cycle consistency.* TPAMI (2022).

Calibrated vs Uncalibrated

Proposition. *A solvable (uncalibrated) graph is parallel rigid.*

Proof [sketch].

Parallel rigid graph $\Leftrightarrow$ for any partition of the edges: $\sum_{i=1}^{k} (3|\mathcal{V}_i| - 4) \geqslant 3n - 4$

Solvable graph $\Longrightarrow$ for any partition of the edges: $\sum_{i=1}^{k} (11|\mathcal{V}_i| - 15) \geqslant 11n - 15$

Only necessary condition!
Unknown if the opposite holds

Arrigoni, Fusiello, Rizzi, Ricci & Pajdla. *Revisiting viewing graph solvability: an effective approach based on cycle consistency.* TPAMI (2022).

Outline

- Introduction
- Calibrated Case
- Uncalibrated Case
- Calibrated vs Uncalibrated
- **Conclusion**

Conclusion

	Calibrated	Uncalibrated
Formulation	Linear system	Polynomial system
Datasets	Large-scale	Small-scale
Interpretation	Connected + Parallel rigid	?
Components	Null-space computation	?

"Solved"

Open issues

References

📕 F. Arrigoni, T. Pajdla & A. Fusiello. *Viewing graph solvability in practice.* ICCV (2023).

📕 F. Arrigoni, A. Fusiello, R. Rizzi, E. Ricci & T. Pajdla. *Revisiting viewing graph solvability: an effective approach based on cycle consistency*. IEEE TPAMI (2022).

📕 F. Arrigoni, A. Fusiello, E. Ricci & T. Pajdla. *Viewing graph solvability via cycle consistency.* ICCV (2021). **Best paper honourable mention** 🏆

📕 F. Arrigoni & A. Fusiello. *Bearing-based network localizability: a unifying view*. IEEE TPAMI (2019).

Thank you for your attention!

Viewing Graph Solvability
in Structure from Motion

Federica Arrigoni

federica.arrigoni@polimi.it

POLITECNICO
MILANO 1863

ISPRS TC II online talk series – September 20, 2023