Viewing Graph Solvability in Structure from Motion

Federica Arrigoni

federica.arrigoni@polimi.it

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
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.
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:

$$m_{ij} \approx P_i M_j$$

In the calibrated case, calibration matrices are known and projection matrices consist of rotations and translations: $P_i = K_i [R_i \ t_i]$
Introduction

Is 3D reconstruction *unique*?

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

\[ m_{ij} \simeq P_i M_j = P_i Q Q^{-1} M_j = P_i Q Q^{-1} M_j \]

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

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 \( \rightarrow \text{unique solution} \)

❌ Otherwise it is called **non solvable** \( \rightarrow \text{multiple (infinitely many) solutions} \)
**Introduction**

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

<table>
<thead>
<tr>
<th>Solvability</th>
<th>Calibrated</th>
<th>Uncalibrated</th>
</tr>
</thead>
</table>

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** $\iff$ 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*:

\[
R_{ij} = R_i R_j^T \\
t_{ij} = -R_i R_j^T t_j + t_i
\]

\[
R_{ij} = R_i R_j^T \\
-z_{ij} = -R_i^T t_i + R_j^T t_j
\]

Consistency constraint between relative and absolute poses

- Relative displacement
- Centre of camera $i$
- Centre of camera $j$
The Calibrated Case

Problem Formulation

**Solvable graph** $\iff$ 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**:

  \[
  \begin{align*}
  R_{ij} &= R_i R_j^T \\
  t_{ij} &= -R_i R_j^T t_j + t_i \\
  \end{align*}
  \]

  Consistency constraint between relative and absolute poses

  \[
  \begin{align*}
  R_{ij} &= R_i R_j^T \\
  -R_i^T t_{ij} &= -R_i^T t_i + R_j^T t_j \\
  \end{align*}
  \]

  The magnitude of relative translations are **unknown**: $||t_{ij}|| = ||z_{ij}|| = ?$
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:

\[ R_i = R_{ij}R_j \iff R_{ij} = R_iR_j^T \]

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

\[ u_{ij} = \frac{x_i - x_j}{||x_i - x_j||} \iff u_{ij} \times (x_i - x_j) = 0 \]

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 \( \text{rank}(S) = 3n - 4 \)

- ✔️ 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 \( \iff \) clustering rows in the null-space of \( S \)

The Calibrated Case
Translations

Solvability for translations $\iff$ **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**.

This is a well studied task!

---

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 \( \Rightarrow \) sufficient conditions.
## The Calibrated Case

### Examples

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


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 $\iff$ \textit{connected} viewing graph
Solvability for translations $\iff$ \textit{parallel rigid} viewing graph

• Parallel rigidity can be tested from the rank of a \textit{linear system}.
• Maximal components can be extracted from the \textit{null-space} of such a system.
• \textit{Large-scale} datasets can be processed. 😊
Outline

• Introduction
• Calibrated Case
• **Uncalibrated Case**
• Calibrated vs Uncalibrated
• Conclusion

- No assumptions
- Reconstruction is **projective**

20/09/23 Viewing Graph Solvability 24
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 \(n > 3\)).


The Uncalibrated Case
Sufficient Conditions

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


20/09/23 Viewing Graph Solvability
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 $G = aI_4 + cv^T \quad \forall a \in \mathbb{R}_{\neq 0}, v \in \mathbb{R}^4$

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

$$G_{hi}G_{ij}^{-1} = a_{hij}I_4 + c_i v_{hij}^T$$

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

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

$c_i \in \mathbb{R}^4$ is known (camera center)

$\text{Solvable graph} \iff G_{ij} = s_{ij} H$

Single projective transformation

The Uncalibrated Case
Algebraic Characterization

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

\[ G_{hi}G_{ij}^{-1} = a_{hi}I_4 + c_i v_{hij}^T \]

- \( G_{hi} \in GL(4) \) is unknown
- \( a_{hij} \in \mathbb{R} \neq 0 \) and \( v_{hij} \in \mathbb{R}^4 \) are unknown
- \( c_i \in \mathbb{R}^4 \) is known (camera center)
- \((h, i)\) and \((i, j)\) are adjacent edges

The Uncalibrated Case
Reduced Formulation

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

\[ G_{hi}G_{ij}^{-1} = a_{hij}I_4 + c_i v_{hij}^T \]

\( G_{hi} \in GL(4) \) is unknown
\( a_{hij} \in \mathbb{R} \neq 0 \) and \( v_{hij} \in \mathbb{R}^4 \) are unknown
\( c_i \in \mathbb{R}^4 \) is known (camera center)

\((h, i)\) and \((i, j)\) are adjacent edges


• It is possible **eliminate variables** 😄

The Uncalibrated Case
Reduced Formulation

- 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

How can we eliminate the $G$ variables? 😐

Idea: 

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

- 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

\[ G_{hi}G^{-1}_{ij} = a_{hi}I_4 + c_i v_{hi}^T \]

Line Graph

\[ G_{\tau}G^{-1}_{\tau'} = Z_{\tau\tau'} \]
\[ Z_{\tau\tau'} = a_{\tau\tau'}I_4 + c_i v_{\tau\tau'}^T \]

Cycle Consistency

\[ Z_{\tau_1\tau_2}Z_{\tau_2\tau_3} \cdots Z_{\tau_{\ell}\tau_1} = I_4 \]

cycle consistency (on all cycles) ⇔ cycle consistency (on a basis)
The Uncalibrated Case
Algorithm

Algorithm 1 Viewing Graph Solvability

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

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

\[
\text{if } s = 1 \text{ then } \\
\quad \text{solvable } \checkmark \\
\text{else } \\
\quad \text{not solvable } \times \\
\text{end if}
\]

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

<table>
<thead>
<tr>
<th>Nodes</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>1.6 s</td>
<td>9 s</td>
<td>93 s</td>
<td>3 min</td>
<td>15 min</td>
<td>35 min</td>
<td>1 h</td>
<td>≈ 2 h</td>
<td>&gt; 4 h</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Data set</th>
<th>Solvable</th>
<th>Unsolvable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>by suff.</td>
<td>by Alg. 1</td>
</tr>
<tr>
<td></td>
<td>Tot.</td>
<td>by nec.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tot.</td>
</tr>
<tr>
<td>Alcatraz Courtyard</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>Buddah Tooth</td>
<td>178</td>
<td>20</td>
</tr>
<tr>
<td>Pumpkin</td>
<td>169</td>
<td>22</td>
</tr>
<tr>
<td>Skansen Kronan</td>
<td>179</td>
<td>8</td>
</tr>
<tr>
<td>Tsar Nikolai I</td>
<td>196</td>
<td>0</td>
</tr>
<tr>
<td>Alamo</td>
<td>136</td>
<td>16</td>
</tr>
<tr>
<td>Ellis Island</td>
<td>136</td>
<td>30</td>
</tr>
<tr>
<td>Gendarmenmarkt</td>
<td>128</td>
<td>11</td>
</tr>
<tr>
<td>Madrid Metropolis</td>
<td>88</td>
<td>28</td>
</tr>
<tr>
<td>Montreal Notre Dame</td>
<td>140</td>
<td>12</td>
</tr>
<tr>
<td>Notre Dame</td>
<td>165</td>
<td>18</td>
</tr>
<tr>
<td>NYC Library</td>
<td>110</td>
<td>19</td>
</tr>
<tr>
<td>Piazza del Popolo</td>
<td>105</td>
<td>22</td>
</tr>
<tr>
<td>Piccadilly</td>
<td>109</td>
<td>23</td>
</tr>
<tr>
<td>Roman Forum</td>
<td>114</td>
<td>28</td>
</tr>
<tr>
<td>Tower of London</td>
<td>123</td>
<td>18</td>
</tr>
<tr>
<td>Trafalgar</td>
<td>86</td>
<td>16</td>
</tr>
<tr>
<td>Union Square</td>
<td>74</td>
<td>19</td>
</tr>
<tr>
<td>Vienna Cathedral</td>
<td>122</td>
<td>8</td>
</tr>
<tr>
<td>Yorkminster</td>
<td>116</td>
<td>14</td>
</tr>
<tr>
<td>Cornell Arts Quad</td>
<td>76</td>
<td>23</td>
</tr>
</tbody>
</table>
The Uncalibrated Case

Summary

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

<table>
<thead>
<tr>
<th></th>
<th>#Eq.</th>
<th>#Var.</th>
<th>#Eq.</th>
<th>#Var.</th>
<th>#Eq.</th>
<th>#Var.</th>
<th>#Eq.</th>
<th>#Var.</th>
<th>#Eq.</th>
<th>#Var.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our formulation</td>
<td>64</td>
<td>36</td>
<td>64</td>
<td>40</td>
<td>112</td>
<td>63</td>
<td>112</td>
<td>67</td>
<td>192</td>
<td>100</td>
</tr>
<tr>
<td>Trager et al.</td>
<td>128</td>
<td>120</td>
<td>144</td>
<td>141</td>
<td>224</td>
<td>198</td>
<td>240</td>
<td>219</td>
<td>352</td>
<td>286</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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 $\Rightarrow$ calibrated solvability
Calibrated vs Uncalibrated

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

Expected result! Well-posed with uncalibrated cameras \(\Rightarrow\) well-posed with calibrated cameras

Calibrated vs Uncalibrated

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

**Proof [sketch].**

Parallel rigid graph $\iff$ for any partition of the edges:

$$
\sum_{i=1}^{k} (3|\mathcal{V}_i| - 4) \geq 3n - 4
$$

Solvable graph $\implies$ for any partition of the edges:

$$
\sum_{i=1}^{k} (11|\mathcal{V}_i| - 15) \geq 11n - 15
$$

*Only necessary condition!*

*Unknown if the opposite holds*

---

Outline

• Introduction

• Calibrated Case

• Uncalibrated Case

• Calibrated vs Uncalibrated

• Conclusion
## Conclusion

<table>
<thead>
<tr>
<th></th>
<th><strong>Calibrated</strong></th>
<th><strong>Uncalibrated</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formulation</strong></td>
<td>Linear system</td>
<td>Polynomial system</td>
</tr>
<tr>
<td><strong>Datasets</strong></td>
<td>Large-scale</td>
<td>Small-scale</td>
</tr>
<tr>
<td><strong>Interpretation</strong></td>
<td>Connected + Parallel rigid</td>
<td>?</td>
</tr>
<tr>
<td><strong>Components</strong></td>
<td>Null-space computation</td>
<td>?</td>
</tr>
</tbody>
</table>

- **“Solved”**
- **Open issues**
References


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