Introduction to Structure from Motion

Image Analysis and Computer Vision

Federica Arrigoni – federica.arrigoni@polimi.it

POLITECNICO MILANO 1863

Exercise Session – December 16, 2024

Outline

- Introduction to SfM
  - Problem definition
  - Bundle adjustment
- Selected SfM Approaches
  - Sequential approach
  - Factorization approach

Outline

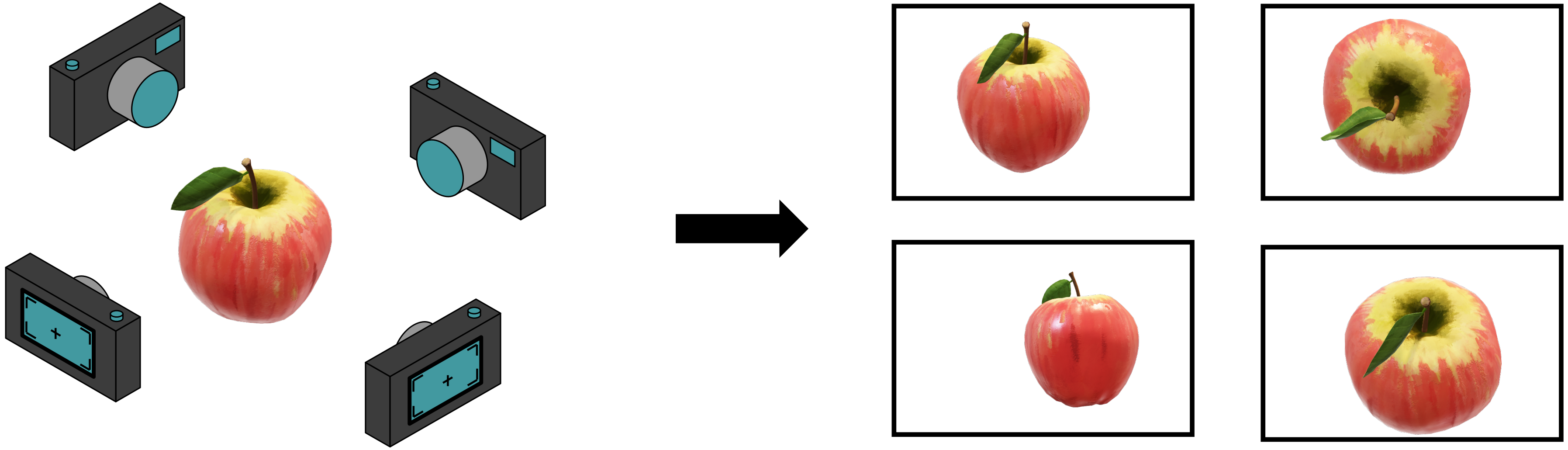
- **Introduction to SfM**
  - Problem definition
  - Bundle adjustment

- Selected SfM Approaches
  - Sequential approach
  - Factorization approach

Introduction

Problem Formulation

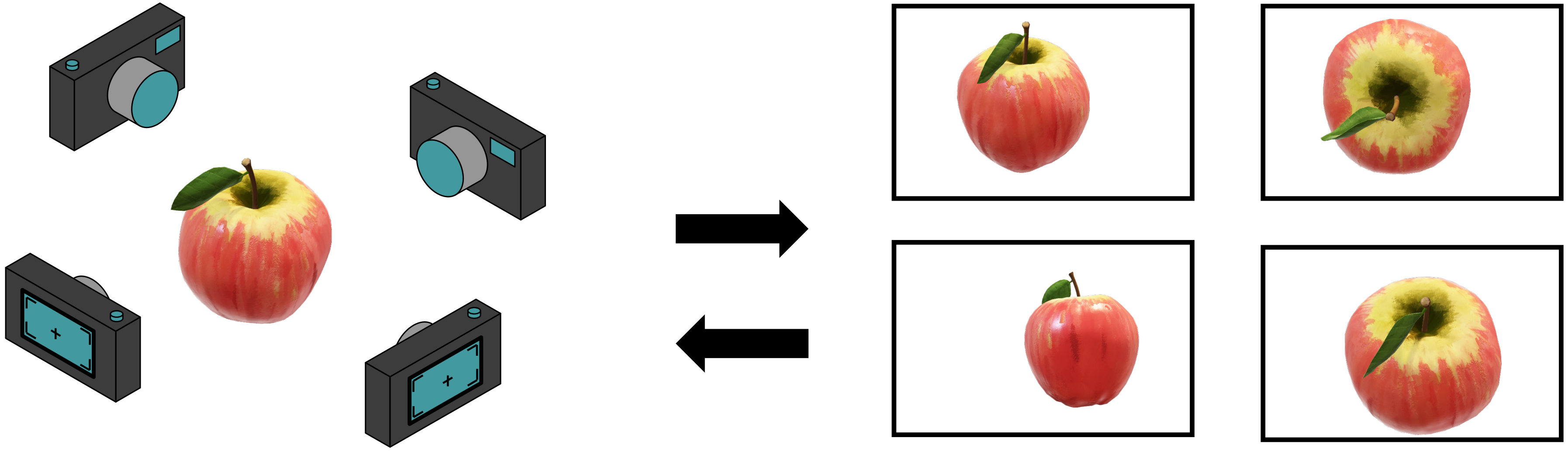
Let us consider a **static scene** (no moving objects) in 3D and suppose that the scene is captured by **multiple cameras** at different positions and viewing directions, thus producing a set of **images**.



Introduction

Problem Formulation

Let us consider a **static scene** (no moving objects) in 3D and suppose that the scene is captured by **multiple cameras** at different positions and viewing directions, thus producing a set of **images**.

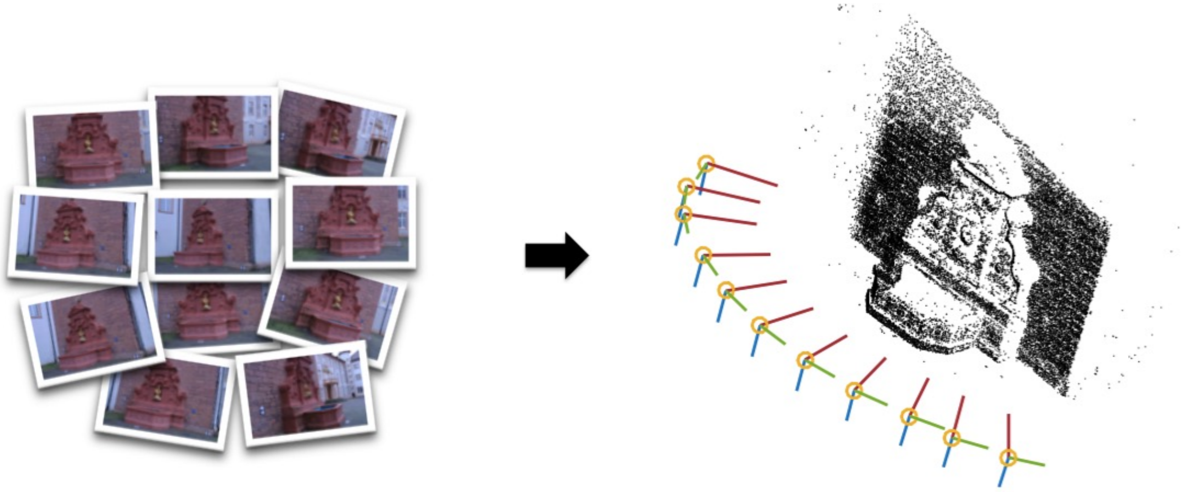


We are interested in the **inverse problem,** namely 3D reconstruction from images.

Introduction
Problem Formulation

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

- **camera motion** = camera matrices (angular attitudes and positions);
- **scene structure** = coordinates of 3D points.



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

Introduction
Problem Formulation

The starting point of a SfM pipeline is a set of **correspondences/matches**, namely image points that correspond to the **same** (unknown) 3D point.

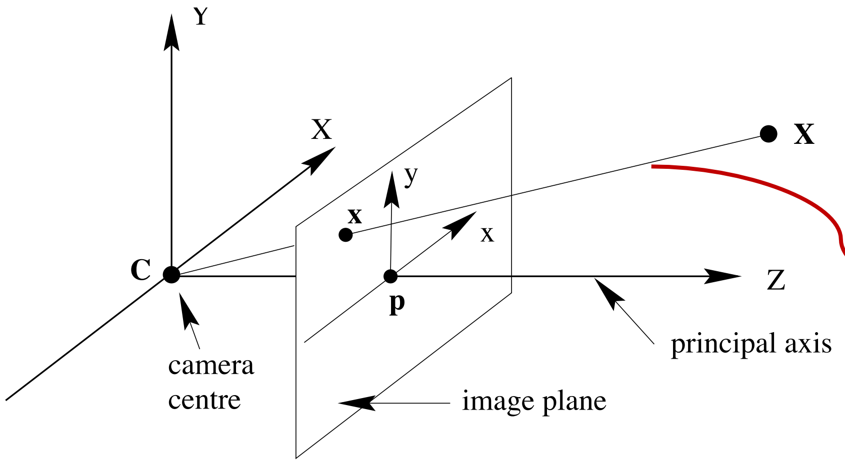
👉 In this lecture we assume that correspondences are **given**.

👉 The final 3D reconstruction is **sparse** (3D point cloud).

Introduction
Problem Formulation

🤔 Can we perform 3D reconstruction from a **single image**?

This problem is **ill-posed** since there is no unique solution for it.



There is an infinite number of 3D points (**optical ray**) that project to the same 2D image point.

The problem is feasible when **additional assumptions** are made.

📕 G. Fahim, K. Amin, S. Zarif. *Single-view 3D reconstruction: a survey of deep learning methods*. Computers&Graphics (2021).

Introduction
Problem Formulation

Structure from motion (SfM) is more **difficult** than other problems in multi-view geometry since **both** cameras and 3D points are unknown!

	Measures	3D Points	Cameras
Triangulation/Intersection	2D-2D	Unknown	Known
Resection/PnP	2D-3D	Known	Unknown
Reconstruction/SfM	2D-2D	Unknown	Unknown

👉 Triangulation and resection can be used as **building blocks** in SfM pipelines.

Introduction
Notation

Equality
up to scale

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

Projection of
point j in image i
(3 x 1 vector)

Projection matrix
of camera i
(3 x 4 matrix)

i=1,…,n
n=number of
cameras

3D coordinates
of point j
(4 x 1 vector)

j=1,…,p
p=number of
points

Introduction

Notation

Equality
up to scale

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

Projection of
point j in image i
(3 x 1 vector)

Projection matrix
of camera i
(3 x 4 matrix)

i=1,…,n
n=number of
cameras

3D coordinates
of point j
(4 x 1 vector)

j=1,…,p
p=number of
points

Calibrated Case

$$P_i = K_i [R_i \ \mathbf{t}_i]$$

Calibration matrix
of camera i
(3 x 3 matrix)

Translation vector
of camera i
(3 x 1 vector)

Rotation matrix of
camera i
(3 x 3 matrix)

Introduction

Notation

Equality up to scale

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

Known Unknown Unknown

Calibrated Case

$$P_i = K_i [R_i \ \mathbf{t}_i]$$

Known Unknown

Unknown

Introduction
Notation

The task of Structure from Motion (SfM) is to compute both **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:

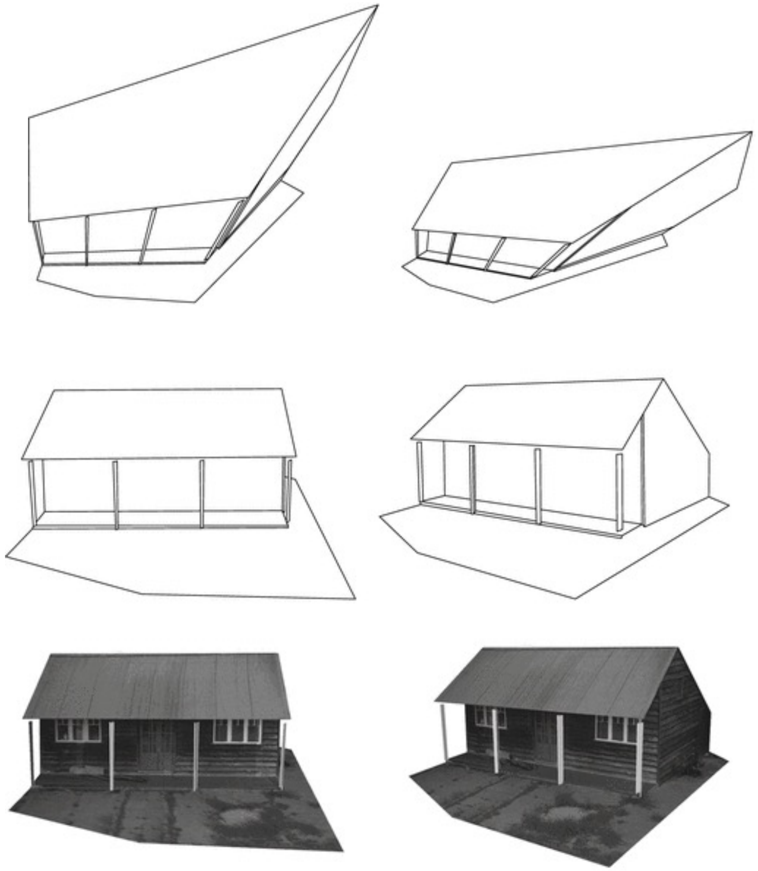
$$m_{ij} \simeq P_i M_j$$

In the calibrated case, calibration matrices are known and projection matrices consist of **rotation matrices** and **translation vectors**:

$$P_i = K_i [R_i \ \mathbf{t}_i]$$

Introduction
Remarks

🤔 What is the meaning of **calibrated vs uncalibrated** reconstruction?



Uncalibrated reconstruction
- No assumptions
- Reconstruction is **projective**
- Metric upgrade is possible at the end (*self-calibration*)

Calibrated reconstruction
- Calibration matrix is required in advance (can be estimated e.g. from a *checkerboard*)
- Reconstruction is **metric**

Introduction
Bundle Adjustment

🤔 *How can we solve SfM?* Let us consider the projection equation:

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

A suitable cost function for SfM is based on the **reprojection error**:

$$\min_{\substack{P_1,\dots,P_n \\ M_1,\dots,M_p}} \sum_{i=1}^{n} \sum_{j=1}^{p} d(P_i M_j, m_{ij})^2$$

Distance in the image plane

We are **adjusting the bundle of rays** between each camera centre and the set of 3D points (or equivalently between each 3D point and the set of camera centres).

Introduction
Bundle Adjustment

👉 A more **general** cost function can be used:

$$\min_{\substack{P_1,\ldots,P_n \\ M_1,\ldots,M_p}} \sum_{i=1}^{n} \sum_{j=1}^{p} w_{ij}\ \rho\big(||\mathcal{F}(P_i M_j) - m_{ij}||\big)$$

Indicator variable

$$w_{ij} = \begin{cases} 1 & \text{if point } j \text{ is visible in image } i \\ 0 & \text{otherwise} \end{cases}$$

L_2 loss or robust loss

Perspective projection

$$\mathcal{F}\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} x/z \\ y/z \end{bmatrix}$$

The number of unknowns is extremely high!

Introduction
Bundle Adjustment

❌ The cost function is non-linear and closed-form solutions do not exist.

⬇

The unknown structure and motion can be recovered by minimizing the reprojection error with an **iterative** method.

👉 The most popular one is **Levenberg-Marquardt,** that is a combination of **Gradient Descent** and **Gauss-Newton**.

rapid decrease in the cost function

rapid convergence in the neighbourhood of the solution

📕 B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon. *Bundle adjustment - a modern synthesis*. International Workshop on Vision Algorithms (2000).

Introduction
Bundle Adjustment

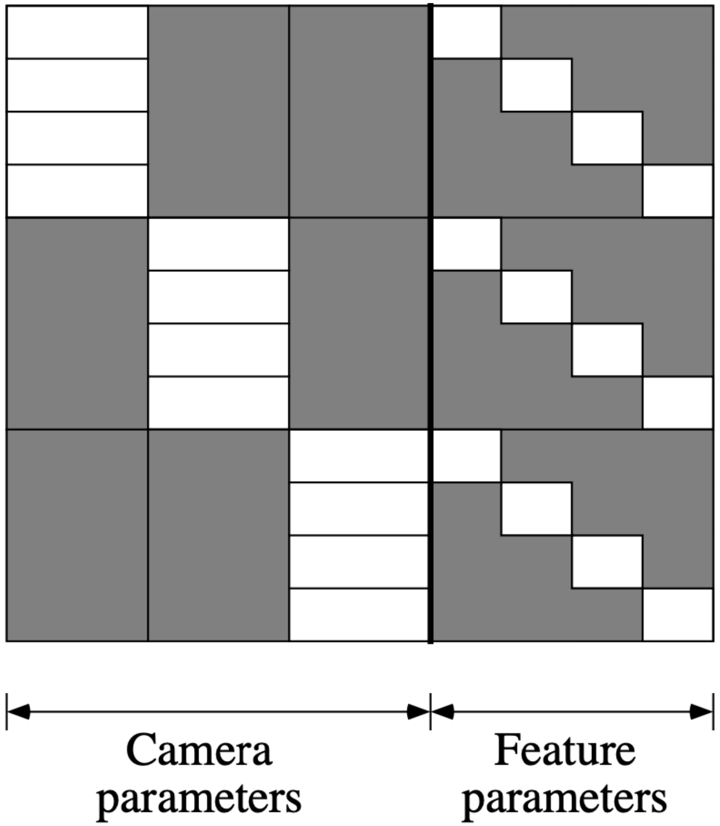
❌ Extremely **large minimization** problem (number of parameters involved).

✅ The Jacobian matrix has a **sparse structure**:

- *primary structure (general):* each term in the cost involves only one camera/point;
- *secondary structure (depends on the data):* each point may be visible only in a subset of cameras.

⬇

reduced system



Introduction
Bundle Adjustment

👉 Several efforts were made to improve performances:

- 📕 S. Agarwal et al. *Ceres Solver.* http://ceres-solver.org/nnls_tutorial.html#bundle-adjustment
- 📕 S. Weber, N. Demmel, T. C. Chan, D. Cremers. *Power Bundle Adjustment for Large-Scale 3D Reconstruction*. CVPR (2023)
- 📕 J. Ren, W. Liang, R. Yan, L. Mai, X. Liu. *MegBA: A High-Performance and Distributed Library for Large- Scale Bundle Adjustment*. ECCV (2022)
- 📕 N. Demmel, C. Sommer, D. Cremers, V. Usenko. *Square Root Bundle Adjustment for Large-Scale Reconstruction*. CVPR (2021)
- 📕 J. Ortiz, M. Pupilli, S. Leutenegger, A. J. Davison. *Bundle Adjustment on a Graph Processor*. CVPR (2020)
- 📕 L. Zhou, Z. Luo, M. Zhen, T. Shen, S. Li, Z. Huang, T. Fang, L. Quan. *Stochastic bundle adjustment for efficient & scalable 3D Reconstruction*. ECCV (2020)
- 📕 J. H. Hong, C. Zach. *pOSE: Pseudo Object Space Error for Initialization-Free Bundle Adjustment*. CVPR (2018)
- 📕 K. N. Ramamurthy, C. Lin, A. Aravkin, S. Pankanti, R. Viguier. *Distributed bundle adjustment*. ICCV (2017)

❌ It requires to be **initialized** close to the solution in order to work in practice (it may converge to a local minima).

⬇

*Bundle adjustment is used as **final refinement**.*

Introduction

Reconstruction Ambiguity

*Is 3D reconstruction **unique**?*

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

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

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

Introduction

Reconstruction Ambiguity

Is 3D reconstruction ***unique****?* 🤔

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

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

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

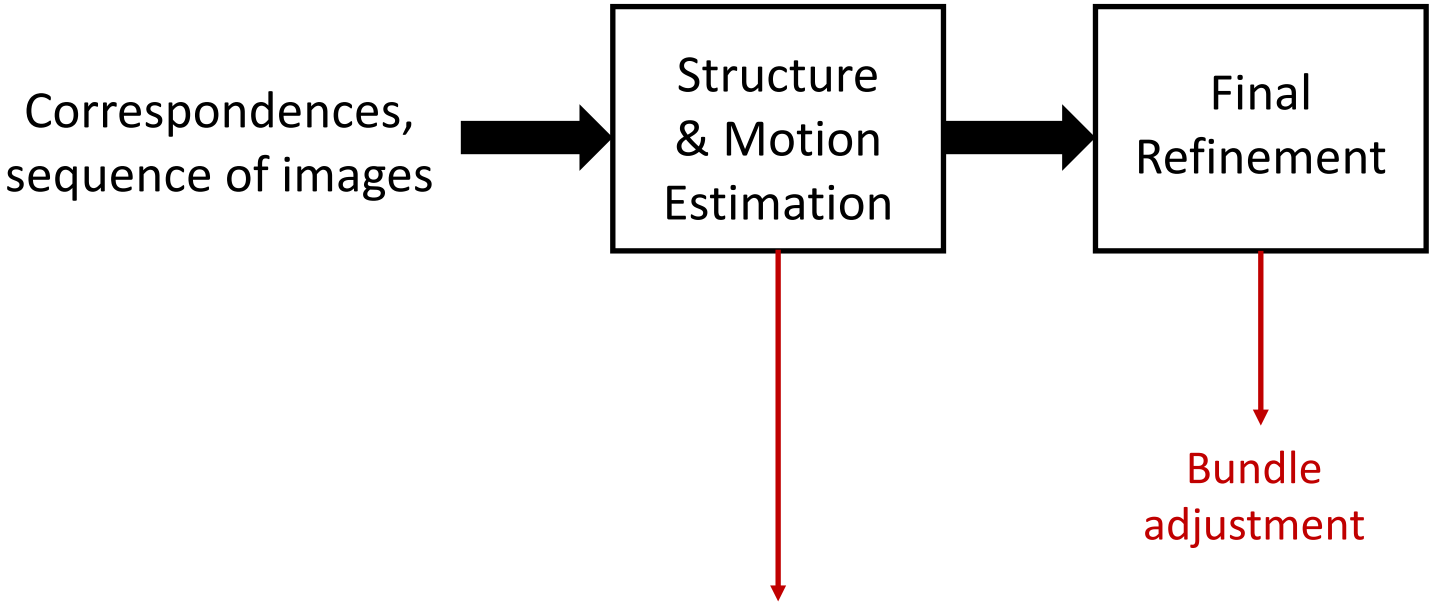
It is important is to analyse the **ambiguities** inherent to the SfM problem:

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

Outline

- Introduction to SfM
  - Problem definition
  - Bundle adjustment
- **Selected SfM Approaches**
  - Sequential approach
  - Factorization approach

Incremental SfM
Problem Formulation

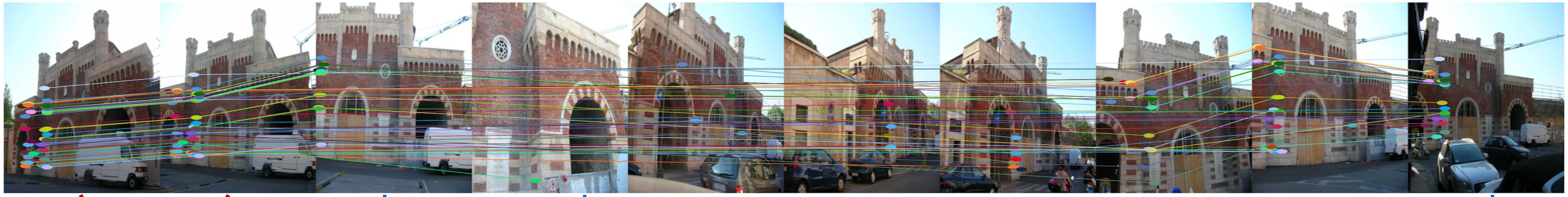


Sequential SfM progressively builds an **increasing model**.

Incremental SfM
Problem Formulation

The idea is to **iteratively grow a partial model** (composed of cameras and points).

👉 It is assumed that images are organized in a **sequence.**



INITIAL PAIR
Initializing the model

NEW IMAGE
Updating the model

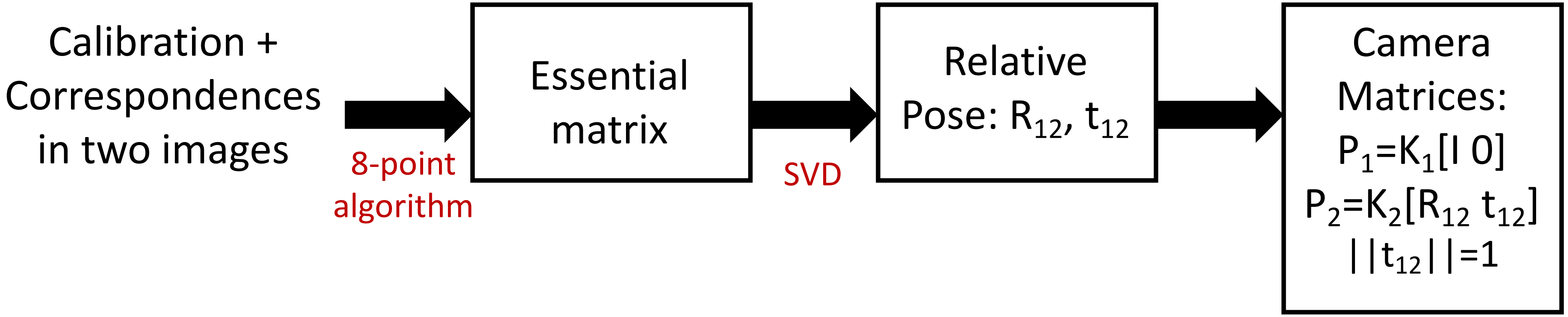
NEW IMAGE
Updating the model

...

NEW IMAGE
Updating the model

Incremental SfM
Algorithm: Initializing the Model

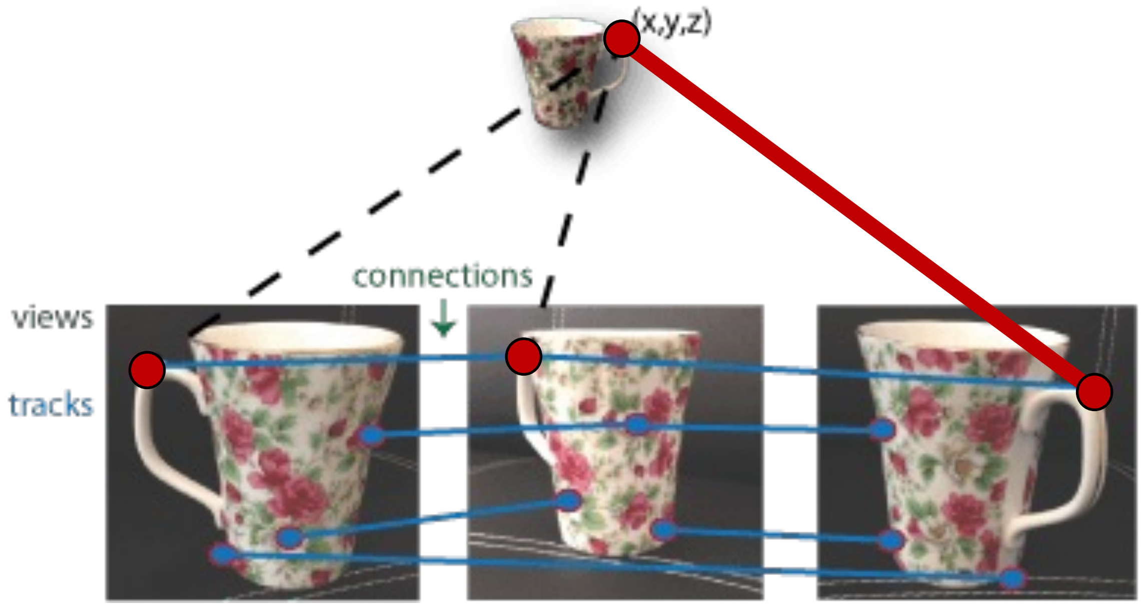
Given the initial image pair, we first compute its **camera motion**:



Once the two camera matrices have been fully determined, the 3D points can be reconstructed via **intersection**/triangulation.

Incremental SfM
Algorithm: Updating the Model

After model initialization we have structure and motion for the first two cameras:



When adding a new image we exploit the available image correspondences: thanks to the initial reconstruction, we get **2D-3D correspondences** for the third camera.

Incremental SfM

Algorithm: Updating the Model

1. The camera projection matrix of the 3rd image is derived via **resection/PnP** (starting from 2D-3D correspondences);

2. The structure is updated via **intersection/triangulation** (starting from the camera matrices of the three cameras and available 2D correspondences):
   - the position of 3D points that are observed in the new image is refined;
   - a new 3D point is initialized when we have a correspondence not related to an existing point.

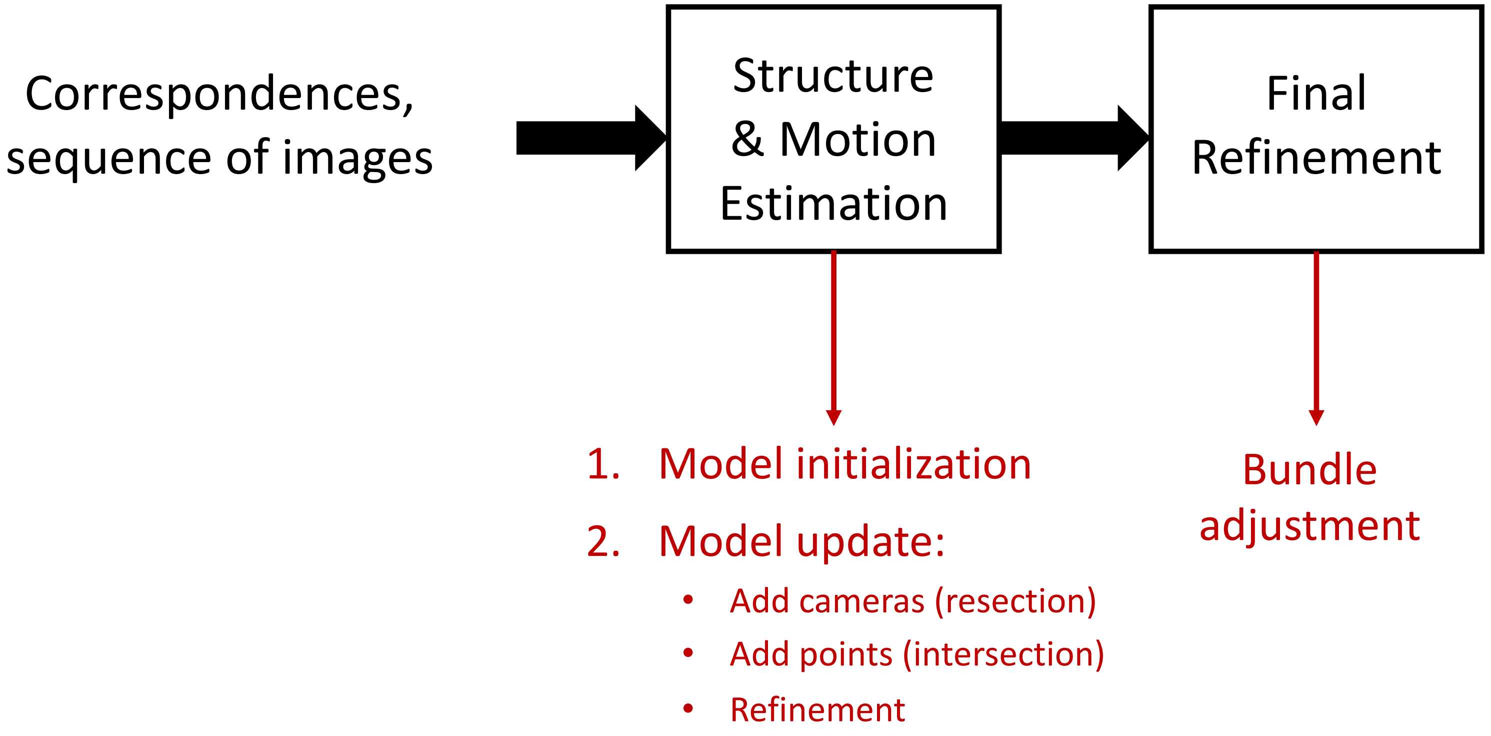
👉 This procedure is applied to all the images **sequentially.**

👉 It can be extended to uncalibrated cameras.

Incremental SfM
Summary

👉 This approach is also called **resection-intersection**.

👉 Frequent **bundle adjustment** is needed to contain error accumulation.



Incremental SfM

Summary

- This approach is also called **resection-intersection**.
- Frequent **bundle adjustment** is needed to contain error accumulation.

Correspondences, sequence of images → Structure & Motion Estimation → Final Refinement

Theoretical Assumptions:
Images are organized in a **sequence** (enough correspondences between the new image and previous ones)

Structure & Motion Estimation:

1. Model initialization
2. Model update:
   - Add cameras (resection)
   - Add points (intersection)
   - Refinement

Final Refinement:

Bundle adjustment

Incremental SfM
Summary

✅ This approach naturally deals with missing data, as it does not require points to be visible in all the images.

✅ It represents the **most successful** pipeline in practice.

❌ The method is not global, so it may suffer from error accumulation.

❌ The choice of the initial pair is critical.

📕 N. Snavely, S. Seitz, and R. Szeliski. *Photo tourism: exploring photo collections in 3D*. ACM TOG (2006)

📕 S. Agarwal, N. Snavely, I. Simon, S. Seitz and R. Szeliski. *Building Rome in a day*. ICCV (2009)

📕 C. Wu. *Towards linear-time incremental structure from motion*. 3DV (2013)

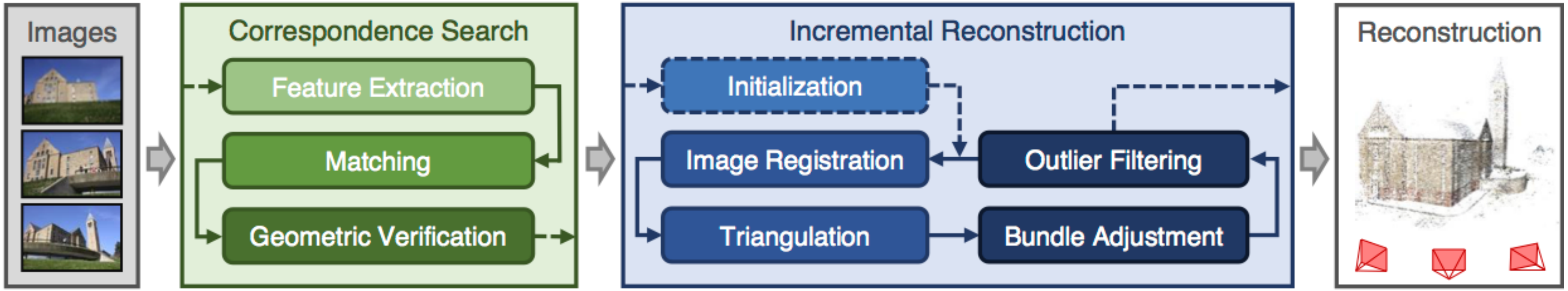
📕 J. L. Schonberger, J. Frahm. *Structure-from-Motion Revisited*. CVPR (2016)

Incremental SfM

Software

There exist several **sequential** SfM systems that work well in practice.

One of the most popular is **COLMAP** https://colmap.github.io



👉 Details are important!

Incremental SfM
Software

There exist several **sequential** SfM systems that work well in practice.

One of the most popular is **COLMAP** https://colmap.github.io

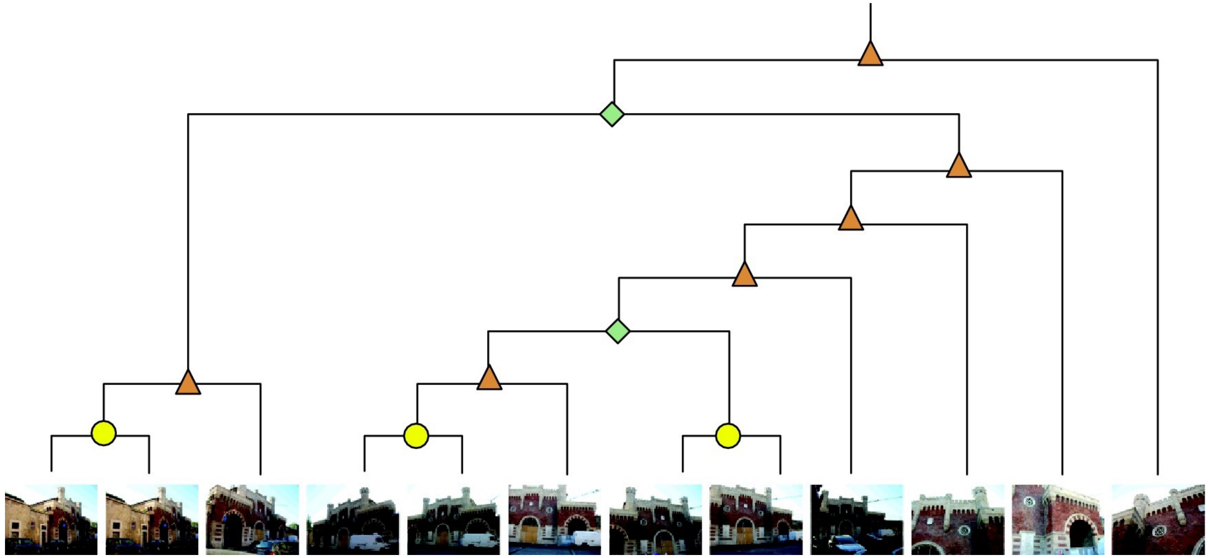
- **Next best view selection:** it tends to favour images with more visible points and more uniform distribution.
- **Local Bundle Adjustment:** it is applied on the set of most connected images.
- **Redundant view mining:** redundant cameras are clustered into groups and cameras within each group are collapsed into a single camera.

👉 Details are important!

🏆 PAMI Mark Everingham Prize (2020) - For the COLMAP SFM and MVS software library. *The Everingham Prize is awared for a selfless contribution of significant benefit to other members of the computer vision community.*

Hierarchical Variation

Incremental reconstruction can be generalized to the case where images are organized in a **tree** instead of a sequence.



1. Solve many two-view problems at the leaves.
2. Traverse the tree. At each node one of these operations takes place:
   - Add one image (resection-intersection).
   - Merge two independent reconstructions (3D registration).

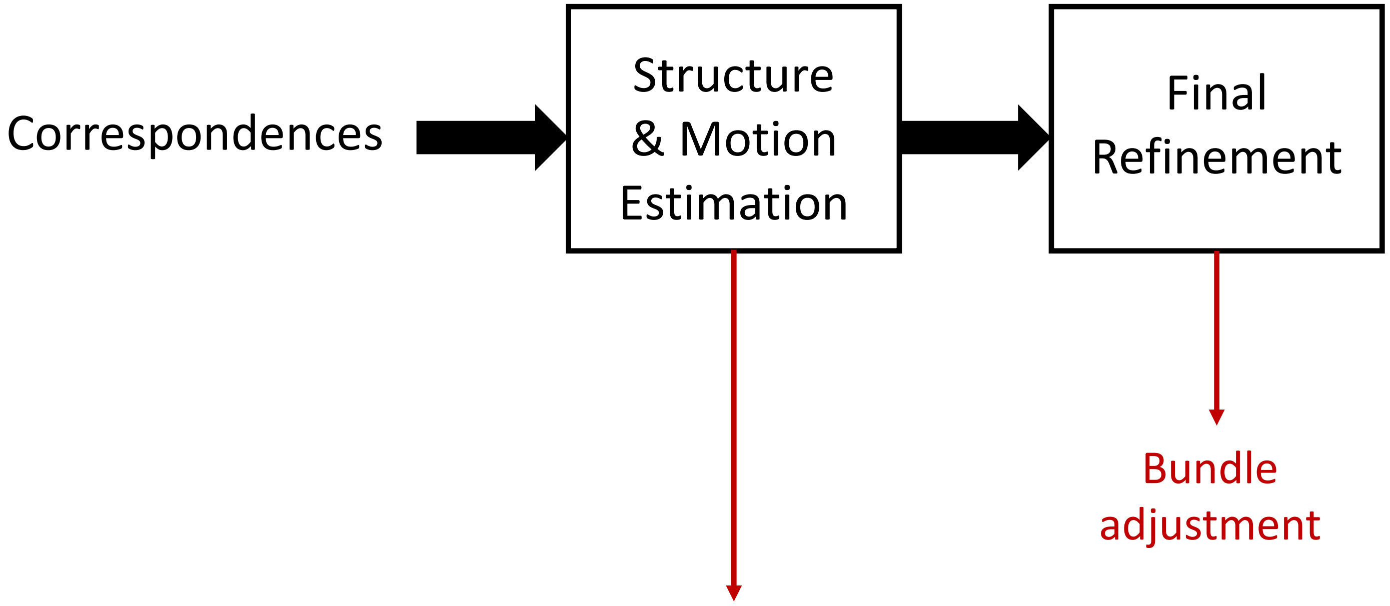
R. Toldo, R. Gherardi, M. Farenzena, A. Fusiello. *Hierarchical structure-and-motion recovery from uncalibrated images*. CVIU (2015).

Outline

- Introduction to SfM
  - Problem definition
  - Bundle adjustment
- Selected SfM Approaches
  - Sequential approach
  - **Factorization approach**

Projective Factorization

Problem Formulation



Factorization methods recover both **structure and motion** simultaneously.

Projective Factorization
Problem Formulation

Factorization is an elegant method that recovers both structure and motion in a **single step**, under suitable assumptions. Cameras are **uncalibrated**.

The projection equation can be expressed as an equality by using scales:

$$\underbrace{m_{ij} \simeq P_i M_j}_{\simeq\ \text{Equality up to scale}} \quad \Longleftrightarrow \quad \underbrace{d_{ij}}_{\text{Projective depth}} m_{ij} \underbrace{=}_{\text{Equality}} P_i M_j$$

👉 In general, projective depths are **unknown**.

👉 For simplicity of exposition, we first consider the case where depths are given.

Projective Factorization
Problem Formulation

The projection equation holds for a **single** image point: $d_{ij}m_{ij} = P_i M_j$

When considering **all the points**, an equivalent compact form can be used:

$$\underbrace{\begin{bmatrix} d_{11}m_{11} & d_{12}m_{12} & \dots & d_{1p}m_{1p} \\ d_{21}m_{21} & d_{22}m_{22} & \dots & d_{2p}m_{2p} \\ \dots & & & \dots \\ d_{n1}m_{n1} & d_{n2}m_{n2} & \dots & d_{np}m_{np} \end{bmatrix}}_{\text{Measurement matrix W, } 3n \times p} = \underbrace{\begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_n \end{bmatrix}}_{\text{Motion matrix P, } 3n \times 4} \underbrace{\begin{bmatrix} M_1 & M_2 & \dots & M_p \end{bmatrix}}_{\text{Structure matrix S, } 4 \times p} \iff \boxed{W = PS}$$

👉 W can be factored into the product of a 3n x 4 matrix P and a 4 x p matrix S. This means that **W has rank 4**.

Projective Factorization
Problem Formulation

👉 We are assuming **known** projective depths:

$$W = PS$$

known unknown unknown

👉 The presence of noise alters the structure of W so that it does not satisfy the above equation exactly. In particular, W does not have rank 4 in practice.

We can consider the following **optimization problem**:

$$\min_{P,S} ||W - PS||_F^2$$

We are looking for the best rank-4 approximation of W

We can obtain a solution in **closed-form** with Singular Value Decomposition (SVD).

📕 P. F. Sturm and B. Triggs. *A factorization based algorithm for multi-image projective structure and motion*. ECCV (1996)

Projective Factorization

Algorithm

1) Compute the SVD of the measurement matrix *W:*

$$W = U\Sigma V^\top = U \begin{bmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \sigma_3 & & & \\ & & & \sigma_4 & & \\ & & & & \sigma_5 & \\ & & & & & \ldots \end{bmatrix} V^\top$$

2) Compute $\widetilde{W}$ that is the rank-4 approximation of *W:*

$$\widetilde{W} = U \begin{bmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \sigma_3 & & & \\ & & & \sigma_4 & & \\ & & & & 0 & \\ & & & & & \ldots \end{bmatrix} V^\top = \underbrace{U_{1:4}}_{\text{First 4 columns}} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \sigma_3 & \\ & & & \sigma_4 \end{bmatrix} \underbrace{V_{1:4}^\top}_{\text{First 4 rows}}$$

3) Recover motion matrix *P* and structure matrix *S* from $\widetilde{W}$:

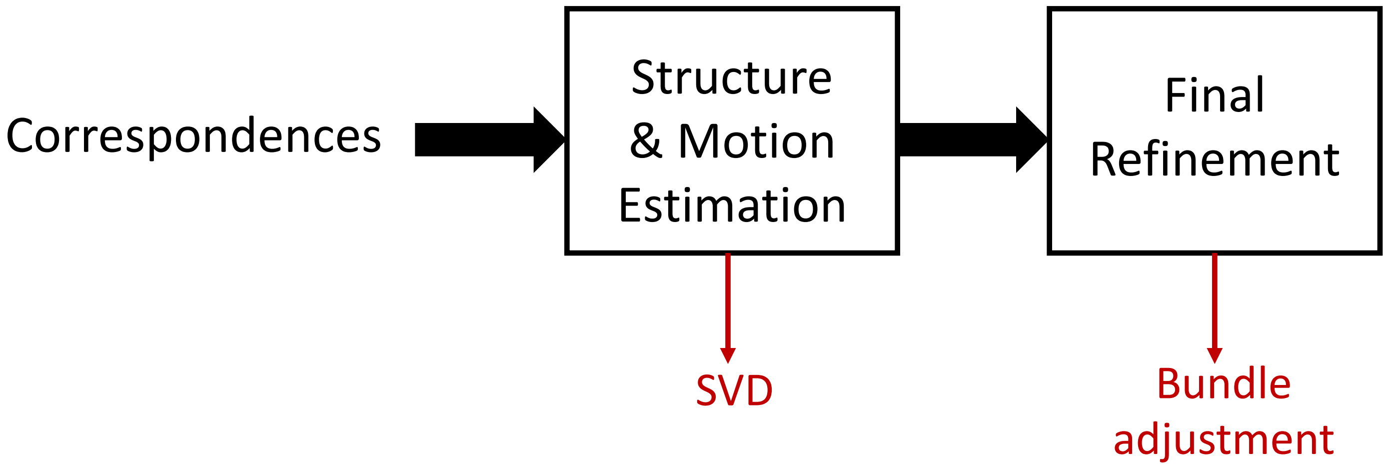
$$P = \underbrace{U_{1:4} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \sigma_3 & \\ & & & \sigma_4 \end{bmatrix}}_{\text{Projective ambiguity}}$$

$$S = V_{1:4}^\top$$

👉 the projective ambiguity could be placed in S as well.

Projective Factorization

Summary



Factorization is an elegant method that recovers both structure and motion in a **single step**, under suitable assumptions:

- **Known projective depths**
- **All points visible in all images**

👉 *In practice such assumptions are not satisfied!*

Projective Factorization
Extensions

🤔 How can we deal with **unknown** projective depths?

The problem can be solved by **alternation**:

- With **known depths**, structure and motion can be recovered via SVD;
- With **known structure and motion**, projective depths can be recovered by solving a linear system: $\underbrace{d_{ij}}_{\text{unknown}} \underbrace{m_{ij} = P_i M_j}_{\text{known}}$
- The above steps are iterated until convergence (or a maximum number of iterations) is reached.

👉 The projective depths can be initialized as: $d_{ij} = 1$

📕 J. Oliensis and R. Hartley. *Iterative extensions of the Sturm/Triggs algorithm: convergence and nonconvergence*. IEEE TPAMI (2007)

Projective Factorization
Extensions

🤔 How can we deal with **missing** correspondences?

The problem can be solved by **matrix completion**:

$$\begin{bmatrix} d_{11}m_{11} & ? & \dots & d_{1p}m_{1p} \\ d_{21}m_{21} & d_{22}m_{22} & \dots & ? \\ \dots & & & \dots \\ ? & d_{n2}m_{n2} & \dots & d_{np}m_{np} \end{bmatrix} \xrightarrow[\text{Rank minimization}]{} \begin{bmatrix} d_{11}m_{11} & d_{12}m_{12} & \dots & d_{1p}m_{1p} \\ d_{21}m_{21} & d_{22}m_{22} & \dots & d_{2p}m_{2p} \\ \dots & & & \dots \\ d_{n1}m_{n1} & d_{n2}m_{n2} & \dots & d_{np}m_{np} \end{bmatrix}$$

Incomplete matrix

Full matrix

📕 Y. Dai, H. Li, and M. He. *Projective Multiview Structure and Motion from Element-Wise Factorization*. IEEE PAMI (2013)

📕 H. Jia and A. M. Martinez. *Low-rank matrix fitting based on subspace perturbation analysis with applications to structure from motion*. IEEE TPAMI (2009)

📕 D. Martinec and T. Pajdla. *3D reconstruction by fitting low-rank matrices with missing data*. CVPR (2005)

Projective Factorization
Conclusion

✅ This approach is **global**: all the cameras/points are considered simultaneously, thus promoting error compensation.

✅ It works under **weak assumptions**: the most general camera model (projective).

❌ It is memory demanding, requiring to store all the points at once.

Alternative approaches

- *Variable projection method* (rewrite the objective in a reduced set of unknowns)

📕 J. Hong, C. Zach, A. Fitzgibbon, R. Cipolla. *Projective Bundle Adjustment from Arbitrary Initialization Using the Variable Projection Method*. ECCV (2016)

📕 J. P. Iglesias, A. Nilsson, C. Olsson. *expOSE: Accurate Initialization-Free Projective Factorization Using Exponential Regularization*. CVPR (2023)

- *Deep learning* (encoder architecture)

📕 D. Moran, H. Koslowsky, Y. Kasten, H. Maron, M. Galun, R. Basri. *Deep permutation equivariant structure from motion*. ICCV (2021)

Projective Factorization
Theoretical Conditions

🤔 *Is projective factorization well-posed?* **NO**

In the absence of noise/missing data, the solution is not unique (up to a single projective transformation), i.e., **false solutions** may appear:

$$
\begin{array}{r|cccccc}
 & \overbrace{\hat{\mathbf{X}}_1}^{=\bar{\mathbf{C}}_1} & \multicolumn{5}{c}{\overbrace{\hat{\mathbf{X}}_2 \quad \hat{\mathbf{X}}_3 \quad \hat{\mathbf{X}}_4 \quad \hat{\mathbf{X}}_5 \quad \hat{\mathbf{X}}_6}^{\hat{\mathbf{X}}_j = (\mathtt{I} - \bar{\mathbf{C}}_1 \bar{\mathbf{C}}_1^T)\mathbf{X}_j}} \\
\hline
\mathtt{P}_1 = \hat{\mathtt{P}}_1 & 0 & \lambda_{12} & \lambda_{13} & \lambda_{14} & \lambda_{15} & \lambda_{16} \\
\mathtt{P}_2 \mathbf{X}_1 \bar{\mathbf{C}}_1^T = \hat{\mathtt{P}}_2 & \lambda_{21} & 0 & 0 & 0 & 0 & 0 \\
\mathtt{P}_3 \mathbf{X}_1 \bar{\mathbf{C}}_1^T = \hat{\mathtt{P}}_3 & \lambda_{31} & 0 & 0 & 0 & 0 & 0 \\
\mathtt{P}_4 \mathbf{X}_1 \bar{\mathbf{C}}_1^T = \hat{\mathtt{P}}_4 & \lambda_{41} & 0 & 0 & 0 & 0 & 0 \\
\mathtt{P}_5 \mathbf{X}_1 \bar{\mathbf{C}}_1^T = \hat{\mathtt{P}}_5 & \lambda_{51} & 0 & 0 & 0 & 0 & 0 \\
 & \multicolumn{6}{c}{\underbrace{\hspace{8cm}}_{\hat{\Lambda}}}
\end{array}
$$

It satisfies the projection equation but it is not projectively equivalent to the true solution.

📕 B. Nasihatkon, R. Hartley, and J. Trumpf. *A generalized projective reconstruction theorem and depth constraints for projective factorization*. IJCV (2015)

Projective Factorization
Theoretical Conditions

🤔 *Is projective factorization well-posed?* **NO**

In the absence of noise/missing data, the solution is not unique (up to a single projective transformation), i.e., **false solutions** may appear:

- Depth matrix has zero rows
- Depth matrix has zero columns
- Depth matrix is cross-shaped

Depth matrix = n x p matrix containing projective depths

$$\begin{bmatrix} & & a & & & \\ & & b & & & \\ c & d & x & e & f & g \\ & & h & & & \end{bmatrix} \quad \begin{bmatrix} a & b & c & x & d & e \\ & & & f & & \\ & & & g & & \\ & & & h & & \end{bmatrix} \quad \begin{bmatrix} a & & & & & \\ b & & & & & \\ c & & & & & \\ x & d & e & f & g & h \end{bmatrix}$$

📕 B. Nasihatkon, R. Hartley, and J. Trumpf. *A generalized projective reconstruction theorem and depth constraints for projective factorization*. IJCV (2015)

Projective Factorization
Theoretical Conditions

🤔 *Is projective factorization well-posed?* **YES (under additional constraints)**

Projective Reconstruction Theorem. *If cameras/points are in generic configuration and the* **projective depths are nonzero**, *then projective factorization is well-posed.*

📕 R. Hartley, and A. Zissermann. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition (2004).

❌ hard to implement

Generalized Projective Reconstruction Theorem. *If cameras/points are in generic configuration and* **the depth matrix does not have zero columns/rows and it is not cross-shaped**, *then projective factorization is well-posed.*

📕 B. Nasihatkon, R. Hartley, and J. Trumpf. *A generalized projective reconstruction theorem and depth constraints for projective factorization*. IJCV (2015)

✅ easy to implement

Projective Factorization
Theoretical Conditions

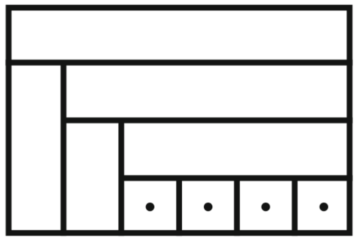
🤔 *Is projective factorization well-posed?* **YES (under additional constraints)**

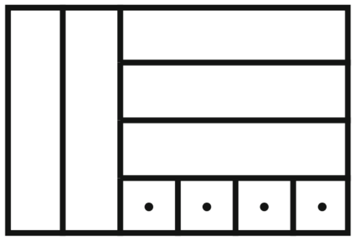
Examples

Step-like mask constraint: fixing certain elements of the depth matrix (*linear equality constraints* → *easy to implement*)

$$\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & 1 & & \\ & & & 1 & 1 & \\ & & & & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & & & \\ & & 1 & & & \\ & & 1 & & & \\ & & 1 & 1 & 1 & 1 \end{bmatrix}$$

Fixing norms of tiles: tiling the depth matrix and requiring each tile to have fixed norm (*compact constraints* → *convergence properties*)



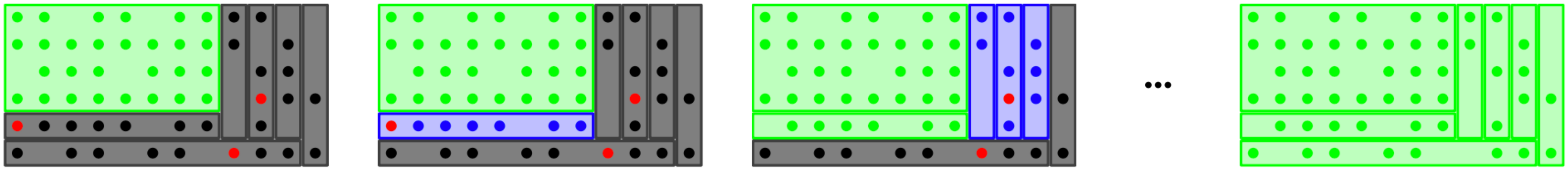


📕 B. Nasihatkon, R. Hartley, and J. Trumpf. *A generalized projective reconstruction theorem and depth constraints for projective factorization*. IJCV (2015)

Projective Factorization
Theoretical Conditions

🤔 *How can we incorporate the constraints?*

- Let’s start from an initial **sub-problem** (*no cross-shape*).
- Missing tiles are **incrementally** added: each tile corresponds to either a view (row) or a point (a column).



Initial sub-problem | Adding 1 view | Adding 3 points | Final reconstruction

📕 L. Magerand and A. Del Bue. *Revisiting Projective Structure from Motion: A Robust and Efficient Incremental Solution.* IEEE PAMI (2020)

Projective Factorization
Theoretical Conditions

🤔 *How can we incorporate the constraints?*

- Let's start from an initial **sub-problem** (*no cross-shape*).
- Missing tiles are **incrementally** added: each tile corresponds to either a view (row) or a point (a column).
  - ➢ With known points, solving for cameras is **linear** (resection).
  - ➢ With known cameras, solving for points is **linear** (intersection).
  - ➢ **Constrains** compliant with the Theorem are used: *the average depth in a tile is fixed to 1.* This is a linear equation that can be used to substitute one of the parameters in the system.
- To prevent error accumulation: after each inclusion, outliers are removed via **RANSAC** and the reconstruction is refined by **re-estimating** all the points/views.

📕 L. Magerand and A. Del Bue. *Revisiting Projective Structure from Motion: A Robust and Efficient Incremental Solution.* IEEE PAMI (2020)

Outline

- Introduction to SfM
  - Problem definition
  - Bundle adjustment

- Selected SfM Approaches
  - Sequential approach
  - Factorization approach