19/12

overview of this topic and different approaches to it!

Introduction to Structure from Motion

wide area of research

Image Analysis and Computer Vision

Federica Arrigoni – federica.arrigoni@polimi.it

POLITECNICO MILANO 1863

Exercise Session – December 16, 2024

Outline

introduction on the PROBLEM

- Introduction to SfM
  - Problem definition   in formal term
  - Bundle adjustment   relevant approach ⇒ rule of approach
- Selected SfM Approaches
  - Sequential approach
  - Factorization approach

examples of possible approaches (most classical)

Outline

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

- Selected SfM Approaches
  - Sequential approach
  - Factorization approach

Introduction

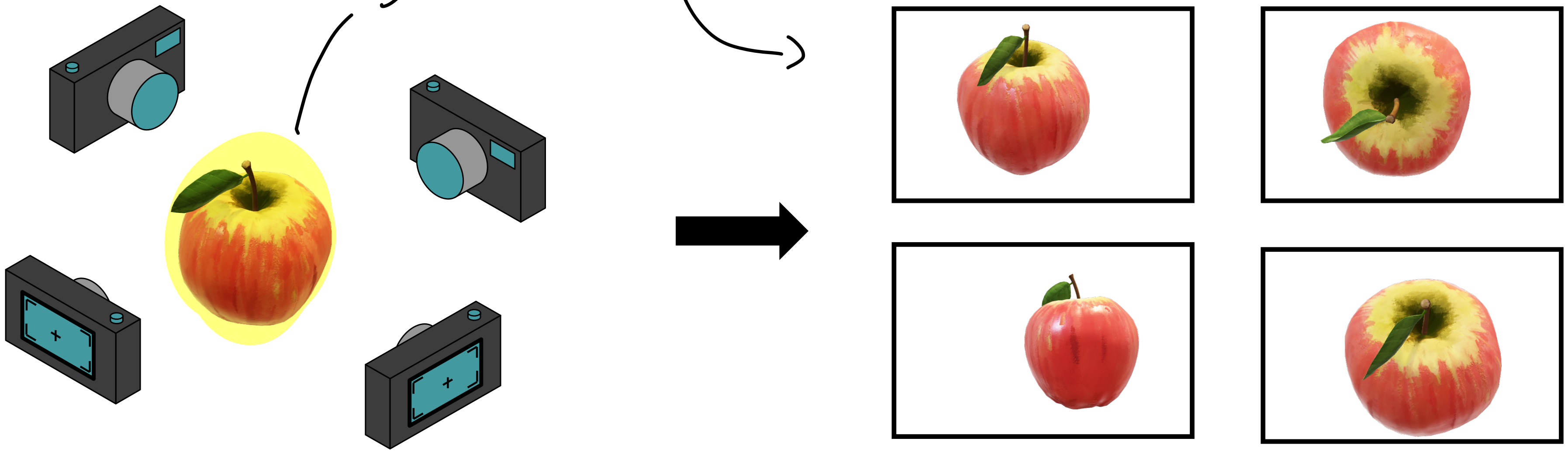
Problem Formulation

scenario

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**.

3D scene

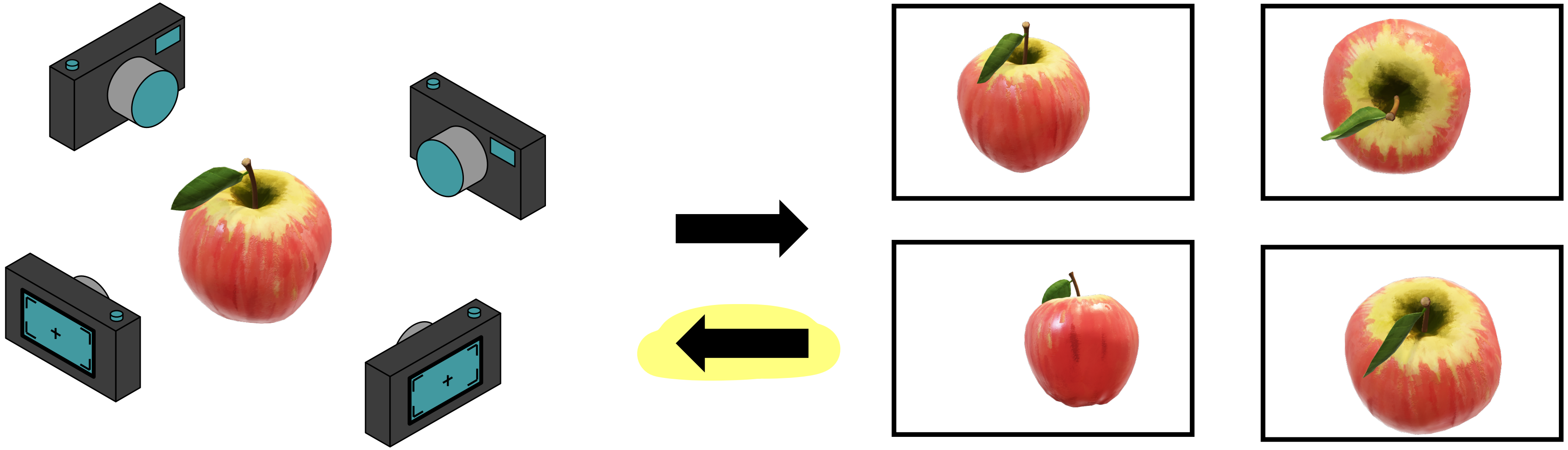
images produces



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**.



given a set of images

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

inverse of image capturing

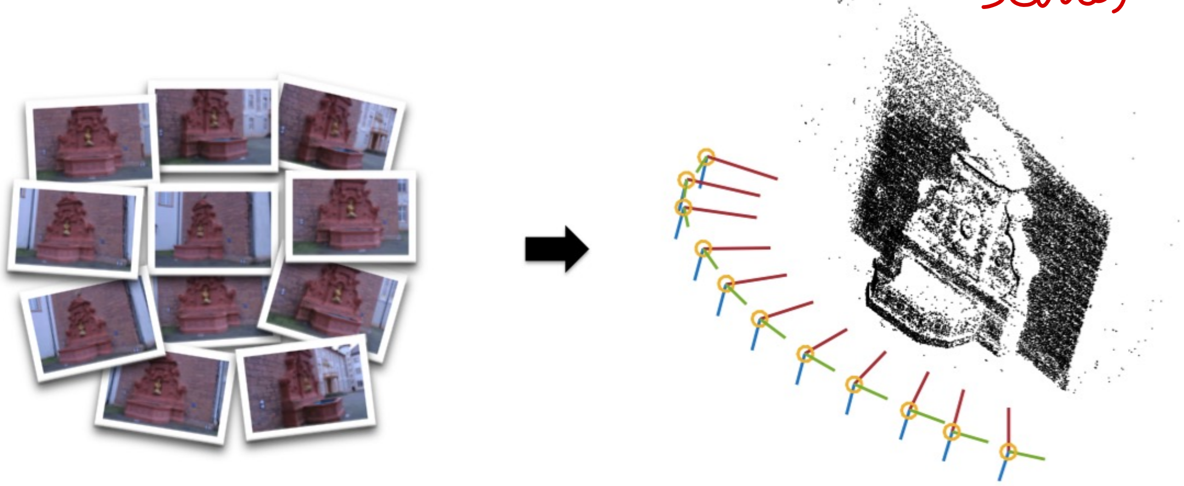
GOAL: recover both

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); ← how camera is moving, different camera matrix (in calibrated cases is just angles/positions)
- **scene structure** = coordinates of 3D points. (points in 3D scene)



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.

(coordinates)

can be visualized as joining them

In SfM corresp. are starting point

- In this lecture we assume that correspondences are **given**.
- The final 3D reconstruction is **sparse** (3D point cloud).

the INPUT to SfM is the set of correspondency (NOT image itself)

(we can reconstruct R via correspondences, detected point sparse)

Introduction

Problem Formulation

we always assume we have MANY IMAGES

NOT UNIQUE solution

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

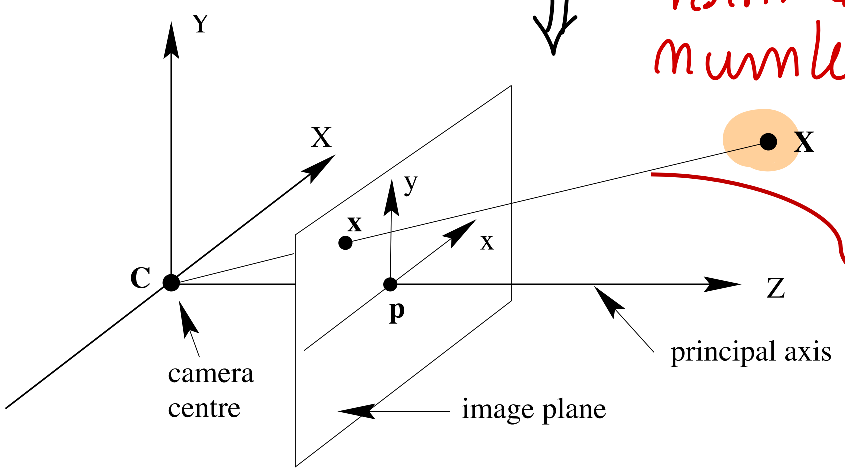
but recall that...

NO

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

We focus on more than 2 images

from only one image... infinite number of 3D points projecting to same image point



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.

we need additional assumption on shape...

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

with different approach! NOT by geometry only

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!

on this issue, we recover 3D points knowing cameras (linear probl)

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

relationship on the image (2D to 2D relation)

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

combine their solut.

(when calibrated cameras, PnP (Perspective n Points))
recover a single-view camera knowing image-space corresp.
approach used in camera calibration... isolate unknowns
and solved by linear systems

Introduction

Notation

Image i

Equality up to scale

point j

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

basic equation of 3D reconstruction is the one of camera model... with 3D point M_j projected via camera P_i generating m_{ij} (proj of point j in image i)

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

homogeneus on image plane

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) = homogeneus coordinates in 3D

j=1,…,p
p=number of points

Introduction

Notation

We assume K_i is pre-computed for example using a checker board

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

⇓

im SfM problem

Equality
up to scale

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

Known — Unknown — Unknown

we only know image points;

when camera calibrated

Calibrated Case

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

Known — Unknown — Unknown

Introduction

Notation

summarize
⇓

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:
(projection)

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

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

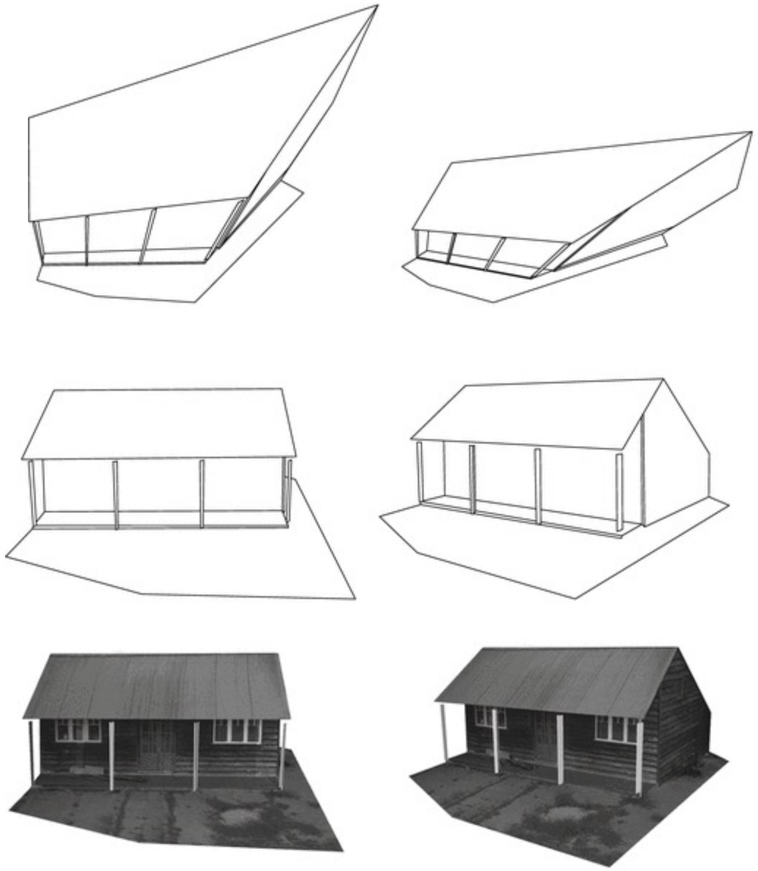
$$P_i = K_i [R_i \ \mathbf{t}_i]$$

Introduction

Remarks

difference ↓

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



most general case P[3x4] matrix

Uncalibrated reconstruction

- No assumptions
- Reconstruction is **projective** → we can have distorsion, being projective no preserve parallelism
- Metric upgrade is possible at the end (*self-calibration*)

Calibrated reconstruction

require assumption

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

corresponding to true scene up to rotation/translation/scaling how to solve ?

Introduction

Bundle Adjustment

process of optimizing the cost function below

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

↓ set up some optimization problem based on projection equation ...

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

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

so we have to define some distances on that equation

summ considering all points and cameras! MANY UNKNOWNS (cameras, points)

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

Distance in the image plane ⇒ compute distance each m_{ij} an projection using P_i

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).

this process minimize reprojection error

⇒ we can formulate MORE in details... —▷

summ for all camera and points ⇒ we have many UNKNOWNS to deal with!

Introduction

Bundle Adjustment

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

example of distance based on perspective projection

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

indicator that consider the contribution only of 3D points visible in a given image!

Indicator variable

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

tell us if point is visible on image

L_2 loss or robust loss

Perspective projection (divide by III coord)

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

evaluate distance as euclidean norm

The number of unknowns is extremely high!

in principle instead of quadratic function we can use others, like ROBUST LOSS (good choice) ⇐ wheight differently points with high residuals... usefull for manage outlyers

Introduction
Bundle Adjustment

$\Longrightarrow$ how to minimize this... $\longrightarrow$

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

we optimize over
P_i, M_j $\quad i = 1..n$
$\quad j = 1..p$

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

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.

(iterative methods!)

⬇

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

ITERATIVE METHOD — typically used in computer vision!

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

take the advantage of both

Gradient Descent → rapid decrease in the cost function

Gauss-Newton → rapid convergence in the neighbourhood of the solution

Inherits rapid decrease

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

Considerations...

Introduction
Bundle Adjustment

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

to optimize resolution, improvemt considering structure of problem

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

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

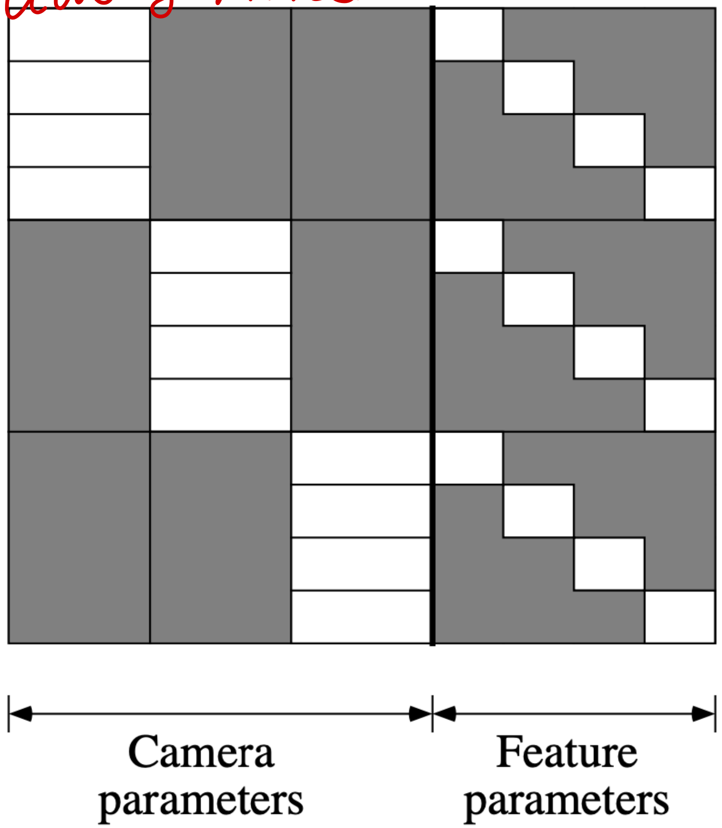
on specific problem

since we use gradient based method, we compute Jacobian, that is sparse

⬇ EXPLOIT SPARSITY!

reduced system

exploit to simplify



this is used @ end or at intermediate step to refine solution

Introduction

Bundle Adjustment

- approximation
- distributed etc...

- still an open problem

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).

⬇

this optimization is NOT complete solution

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

SfM techniques focus on providing initialization ⟹ example of initial.

consideration on the problem

Introduction

Reconstruction Ambiguity

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

NO

NOT UNIQUE! because

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

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

Q 4x4 invertible matrix.. NOT visible impact

from one solution we can derive NEW ones using Q

} coupled matrix

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

global...

create new elements satisfying same eq. NOT unique solution!

NOT UNIQUE

however Q might be NOT only ambiguity ⟹

Introduction

Reconstruction Ambiguity

*Is 3D reconstruction **unique**?* 🤔

In some case we have ill-posed problem ☹

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

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

uniqueness up to global transform

If cameras are calibrated, then the ambiguity is represented (<u>at least</u>) 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 ✅ (usually up to single transformation)
- multiple transformations → ill-posed/degenerate problem ❌ (study on degeneracies!)

applied to camera.. Multiple scale

Outline

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

(classical old approaches, still used)

examples of possible approaches to solve SfM which can be viewed as STARTING POINTS for Bundle adjustment

Incremental SfM (sequential approach)

Problem Formulation

starting point

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

we assume images organized in sequence
"ORDERING" of images

simultaneously considered! ⇒ interaction between them

Bundle adjustment

(as seen before..)

↳ also used in intermediate step

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

INCREMENTAL way

Incremental SfM
Problem Formulation

start from simple small model, and add thing...

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

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

ORDER of how to process images

INITIAL PAIR
Initializing the model

Using first images ← two camera initial

NEW IMAGE
Updating the model

NEW IMAGE
Updating the model

...

New camera and point, increase model each time

NEW IMAGE
Updating the model

proceed over ALL images

how to INITIALIZE and UPDATE...

by combining concept of CV...

Incremental SfM

Algorithm: Initializing the Model

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

between two cameras

set up global coordinate system

from corresp.

Calibration + Correspondences in two images → (8-point algorithm) → Essential matrix → (SVD) → Relative Pose: R_{12}, t_{12} → Camera Matrices: $P_1=K_1[I\ 0]$, $P_2=K_2[R_{12}\ t_{12}]$, $||t_{12}||=1$

defines essential matrix from points

decompose essential matrix

up to scale

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

INITIALIZE: compute P_1, P_2 and 3D points viewed ⇒ first: motion(P_1, P_2) & cameras...

Incremental SfM

Algorithm: Initializing the Model

produce SfM from first two images..
STARTING POINT

Initialization

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

Calibration + Correspondences in two images → (8-point algorithm) → Essential matrix → (SVD) → Relative Pose: R_{12}, t_{12} → Camera Matrices: $P_1=K_1[I\ 0]$, $P_2=K_2[R_{12}\ t_{12}]$, $||t_{12}||=1$

this is simple because for couple of images closed-form solutions exists... make original task simple

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

↳ also 3D points initialization needed

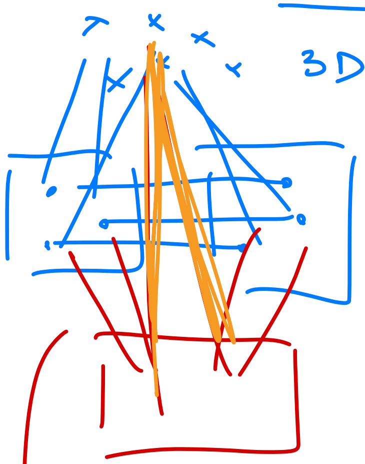
process of computing 3D points from camera knowledge! ⇒ 2-images, we can recover motion then triangulation

Incremental SfM

Algorithm: Updating the Model

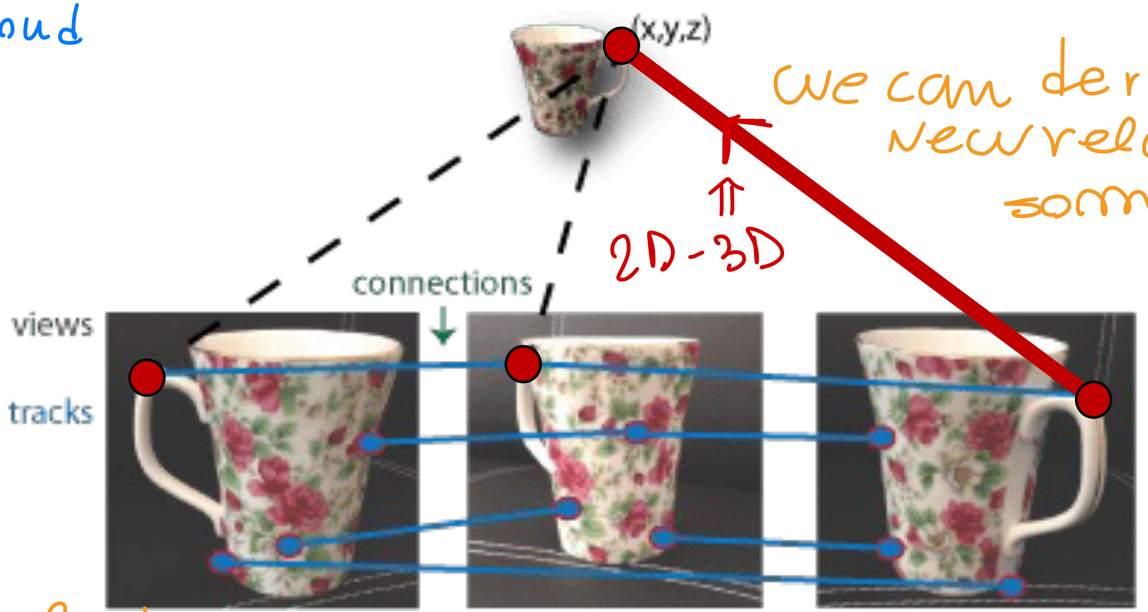
given a new image

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



3D point-cloud by first images

new image



2D-3D

we can derive new relation... some 3D coord on point-cloud

combine informations and relate to point cloud

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.

between images corresp to 3D scene

possible to be solved in closed-form by RESECTION/PnP to take camera

Incremental SfM

Algorithm: Updating the Model

P_1, P_2 ← P_3 reconstructed + update model

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

then we update in 3D points by triangulation for new KNOWN camera

2. The structure is updated via **intersection/triangulation** (starting from the camera matrices of the three cameras and available 2D correspondences):
   - (1) the position of 3D points that are observed in the new image is refined; situation 1
   - (2) a new 3D point is initialized when we have a correspondence not related to an existing point.

situation 2

what can occur

⇓ iteratively...

new points + refinement

- 👉 This procedure is applied to all the images **sequentially.**
- 👉 It can be extended to uncalibrated cameras.

same model update, what change is INITIALIZATION and we rely on foundamental matrix! ⇒ also here closed-form solution exists

Incremental SfM

Summary

alternate the two step ⇒ add new camera ↓ add new point

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

NOT only at the end but also at intermediate step

INPUT

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

Structure & Motion Estimation:

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

Final Refinement:

Bundle adjustment

in sequence ∀ images

sequential update may be subject to error propagation

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**

Structure & Motion Estimation:

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

Final Refinement:

Bundle adjustment

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

guarantees of working only in this case

Incremental SfM

Summary

PRO/CONS

✅ This approach naturally deals with missing data, as it does not require points to be visible in all the images. NO additional info/ visible points in all images NOT

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

how 3D reconstr is NOWDAYS solved

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

because sequential process

❌ The choice of the initial pair is critical. influence solution

- 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

nice GUI to use!
easy to use with
interface to do reconstruction
and so on...

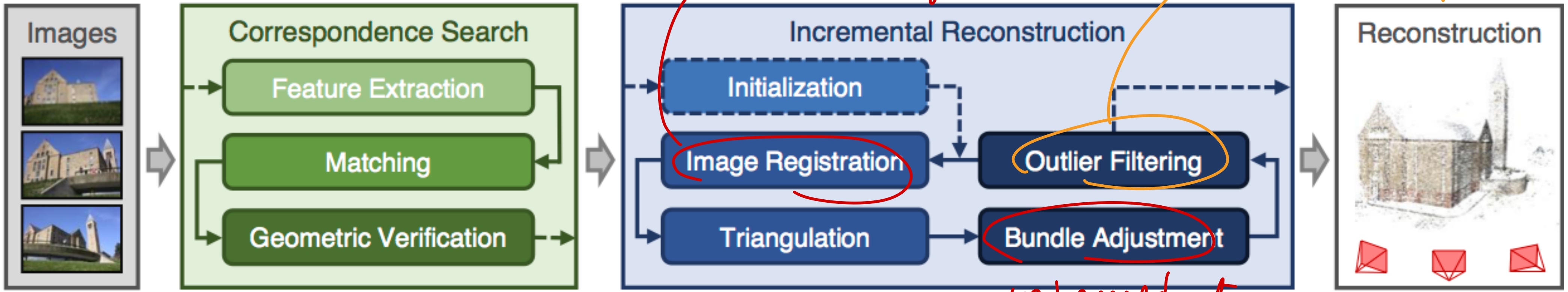
There exist several **sequential** SfM systems that work well in practice.

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

improve
result

remove correspondence
to improve result

new image



intermediate
adj to contain error propag.

👉 Details are important!

It follows the seen approach, INCREMENTAL
reconstruction from correspondence

done by feature
extraction + matching
to see coherent points ⇒ allow points discard

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

↳ additional improvement

with already matched ones

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

↳ when close to be identical, we group cameras similar (no additional info)

👉 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.*

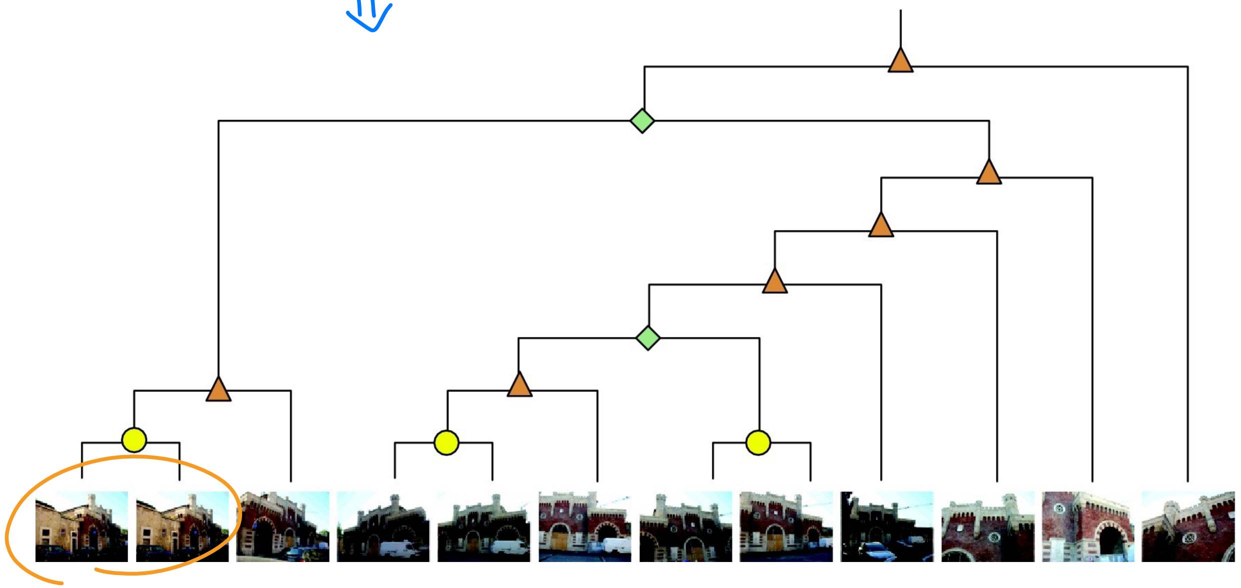
Hierarchical Variation

in general:

of this approach...

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

⇓



phase of initialization @ tree leaves

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). as sequential approach
   - Merge two independent reconstructions (3D registration). when merging different pairs

align 3D models!

still use other tools

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

Outline

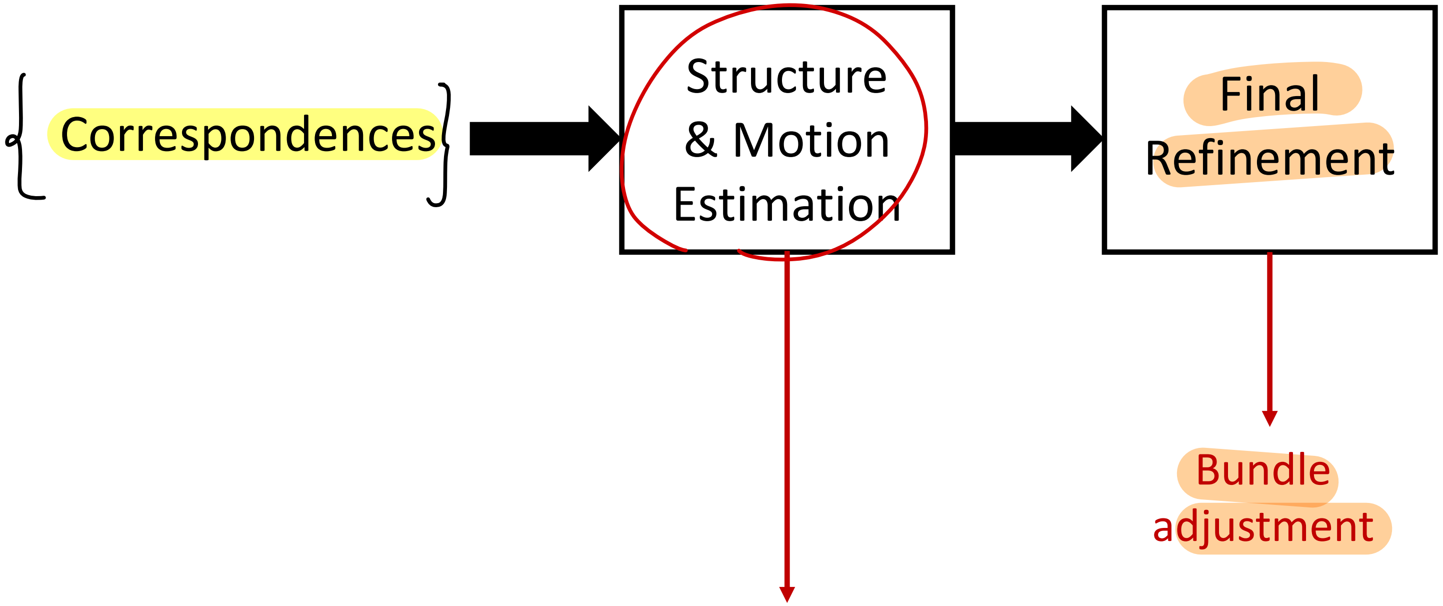
- Introduction to SfM
  - Problem definition
  - Bundle adjustment
- Selected SfM Approaches
  - Sequential approach
  - **Factorization approach** (II approach) => D

Projective Factorization

Problem Formulation

idea of solving STRUCTURE + MOTION simult.

uncalibrated



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

initial guess before minimization ⟹

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:

remove up-to scale

$$m_{ij} \simeq P_i M_j \iff d_{ij} m_{ij} = P_i M_j$$

scale transforming $\simeq$ into "="

Equality up to scale

Projective depth

Equality (using explicit scale)

each image point has unknown scale as projective depth

(*)

- In general, projective depths are **unknown**.
- For simplicity of exposition, we first consider the case where depths are given.

Idea of rewriting as matrix form

Projective Factorization

Problem Formulation

as if all points visible in all images

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:

$$\begin{bmatrix} d_{11}m_{11} & d_{12}m_{12} & \dots & d_{1p}m_{1p} \\ d_{21}m_{21} & d_{22}m_{22} & \dots & d_{2p}m_{2p} \\ \dots & & & \dots \\ d_{n1}m_{n1} & d_{n2}m_{n2} & \dots & d_{np}m_{np} \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_n \end{bmatrix} \begin{bmatrix} M_1 & M_2 & \dots & M_p \end{bmatrix} \iff \boxed{W = PS}$$

compact form:

1 column ∀ 3D point (p)
3 rows ∀ camera (3n)

Measurement matrix W
[3n x p]

Motion matrix P
[3n x 4]
collect all cameras

Structure matrix S
[4 x p]
collect all coordinates of 3D point 1..p of $\mathbb{R}^4$ in homogeneous space

👉 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**.

↳ low rank wrt its dimension ⇒ key of factorization

Projective Factorization
Problem Formulation

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

$$W = PS$$

known unknown unknown

in absence of noise this holds

BUT...

👉 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)

best low rank in F norm sense can be used by SVD

Projective Factorization

Algorithm

diagonal matrix
~ projective ambiguity,
doesn't impact P
defined up to global
proj transform

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:

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

First 4 columns
FIRST columns of U

First 4 rows

we cut SVD from 4TH, ordered by σ

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

$$P = U_{1:4} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \sigma_3 & \\ & & & \sigma_4 \end{bmatrix}$$

size of MATRIX

Projective ambiguity

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

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

Projective Factorization

Summary

Correspondences → Structure & Motion Estimation → Final Refinement

SVD

Bundle adjustment

closed form solution

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**

(full visibility scenario)

⇒ Relevant assumptions that must hold!

👉 *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: $d_{ij} m_{ij} = P_i M_j$

unknown known

linear system

- The above steps are iterated until convergence (or a maximum number of iterations) is reached.

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

initialization! to perform proj factoriz. when dij unkn

📕 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:**

uncomplete W

area of research, recover missing part of matrix

low rank matrix.. redundant entry, we can recover it by rank minimiz.

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

Incomplete matrix

some blocks with non visible points on image

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)

exploit low rank

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.

in a single matrix

Alternative approaches

still based of Factorization

scalability

- *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

imply alternative approach...

🤔 *Is projective factorization well-posed?* **NO** (NOT unique solution)

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

even in simple scenario, NOT unique

chosen example to have ambiguous case!

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

$\hat{\Lambda}$

depth matrix storing all scale has special structure

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

pathological cases with false solutions

pathological example!

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

Projective Factorization

Theoretical Conditions

We need additionae assumptions to be well posed ⇒

🤔 *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) to solve this issue..

Projective Factorization
Theoretical Conditions

$\Downarrow$

🤔 *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).

≥ 0 enforcing this is HARD!

❌ hard to implement

$\Downarrow$ extended version...

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)**

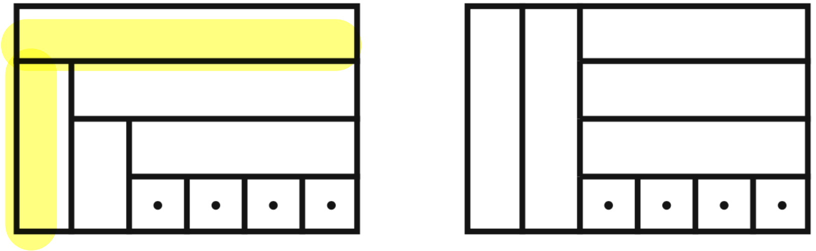
to force this form:

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*)



≃ as incremental approach

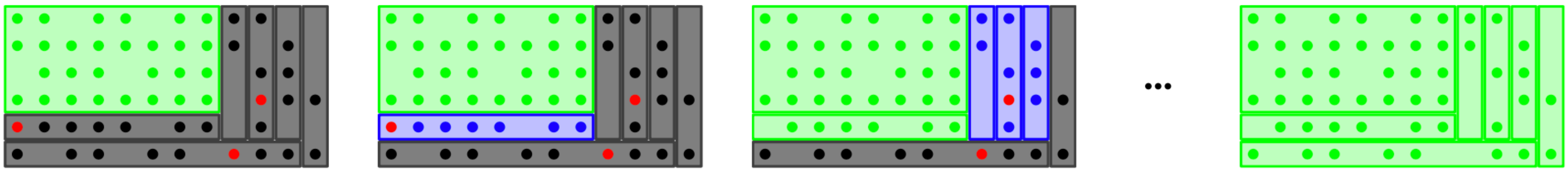
B. Nasihatkon, R. Hartley, and J. Trumpf. *A generalized projective reconstruction theorem and depth constraints for projective factorization*. IJCV (2015)

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). simi lar to sequential approach



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.

exploits theorem, enforce constraint to ensure

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

well posing of factorization approach!

Outline

- Introduction to SfM
  - Problem definition
  - Bundle adjustment

- Selected SfM Approaches
  - Sequential approach
  - Factorization approach