清華大學 Tsinghua University
交叉信息研究院 Institute for Interdisciplinary Information Sciences

Lecture 4: Learning-based SFM

Li Yi

2025.03.13

Recap

- Transformations and homogeneous coordinates
- Rotation and SO(n)
- 3D Rotation representation
  - Euler Angles
  - ***Axis-Angle***
  - Quaternion

Given $\hat{\omega}$ and θ, what is $R \in SO(3)$?

- Definition of Matrix Exponential:

$$e^{[\hat{\omega}]\theta} = I + \theta[\hat{\omega}] + \frac{\theta^2}{2!}[\hat{\omega}]^2 + \frac{\theta^3}{3!}[\hat{\omega}]^3 + \cdots$$

- Sum of infinite series? **Rodrigues Formula**
  - Can prove that $[\hat{\omega}]^3 = -[\hat{\omega}]$
  - Then, use Taylor expansion of **sin** and **cos**
  - $e^{[\hat{\omega}]\theta} = I + [\hat{\omega}]\sin\theta + [\hat{\omega}]^2(1 - \cos\theta)$

Slides credit: Hao Su

Is Axis-Angle a Unique Parametrization for SO(3)

- First question: Is there a **unique** parametrization?
  - No:
    1. $(\hat{\omega}, \theta)$ and $(-\hat{\omega}, -\theta)$ give the same rotation
    2. when $R = I$, $\theta = 0$ and $\hat{\omega}$ can be arbitrary

- When 2 does not happen, and if we also restrict $\theta \in (0, \pi]$, a unique parameterization exists:
  - when $\mathrm{tr}(R) \neq -1$, can be computed by

$$\theta = \arccos \frac{1}{2}[\mathrm{tr}(R) - 1], \quad [\hat{\omega}] = \frac{1}{2\sin\theta}(R - R^T)$$

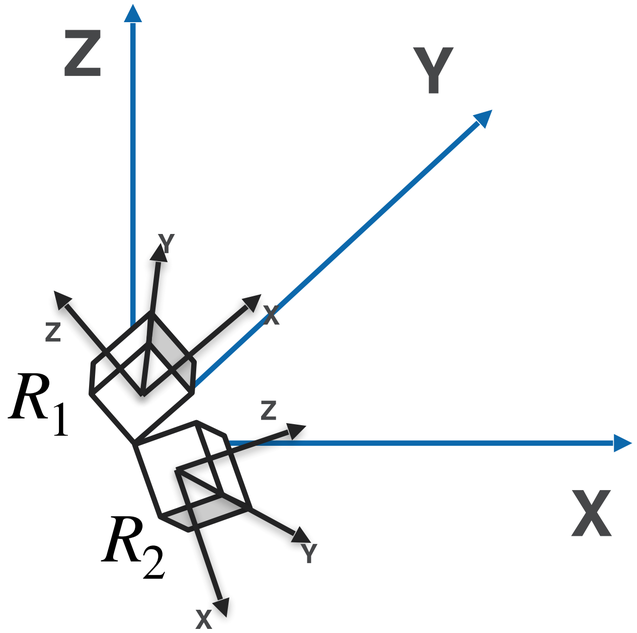
  - when $\mathrm{tr}(R) = -1$, they are the cases that $\theta = \pi$ for rotations around x/y/z axis

Slides credit: Hao Su

Distance between Rotations

- How to measure the distance between rotations (R_1, R_2)?
- A natural view is to measure the (minimal) effort to rotate the body at R_1 pose to R_2 pose:

$$\because (R_2 R_1^T) R_1 = R_2 \quad \therefore \text{dist}(R_1, R_2) = \theta(R_2 R_1^T) = \arccos \frac{1}{2}[\text{tr}(R_2 R_1^T) - 1]$$



Slides credit: Hao Su

Inspection from Learning Perspective

- When used in networks, one prominent issue is:
  - Suppose that you are estimating $\theta\hat{\omega}$ as a 3D vector
  - To keep a unique parameterization, you assume that $\theta \in (0,\pi]$
  - Your current solution is $\pi\hat{\omega}$
  - $(\pi - \epsilon)(-\hat{\omega})$ is mapped to a neighborhood point in $SO(3)$, but it is not in the neighborhood of the domain, hence gradient descent could not achieve it

Slides credit: Hao Su

Summary of Axis-Angle

- Axis-Angle is an intuitive rotation representation
- By adding a constraint to the domain of θ , the parameterization can be unique at most points
- Can be converted to and from rotation matrices by exponential map and its inverse (when possible)
- Induced a distance between rotations which is a metric in $\mathrm{SO}(3)$ (independent of parameterization)

Recap

- Transformations and homogeneous coordinates
- Rotation and SO(n)
- 3D Rotation representation
  - Euler Angles
  - Axis-Angle
  - ***Quaternion***

Mathematical Definition

- Recall the complex number $a + b\mathbf{i}$

- Quaternion is a more generalized complex number:

  $$q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

  - w is the real part and $\overrightarrow{v} = (x, y, z)$ is the imaginary part
  - Imaginary: $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$
  - anti-commutative :
    $\mathbf{ij} = \mathbf{k} = -\mathbf{ji},\ \mathbf{jk} = \mathbf{i} = -\mathbf{kj},\ \mathbf{ki} = \mathbf{j} = -\mathbf{ik}$

Slides credit: Hao Su

Properties of General Quaternions

- In vector-form, the product of two quaternions:
  For $q_1 = (w_1, \vec{v}_1)$ and $q_2 = (w_2, \vec{v}_2)$

$$q_1 q_2 = (w_1 w_2 - \vec{v}_1^T \vec{v}_2, w_1 \vec{v}_2 + w_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2)$$

- Conjugate: $q^* = (w, -\vec{v})$
- Norm: $\|q\|^2 = w^2 + \vec{v}^T \vec{v} = qq^* = q^* q$
- Inverse: $q^{-1} := \dfrac{q^*}{\|q\|^2}$

Slides credit: Hao Su

Unit Quaternion as Rotation

- A **unit** quaternion $\|q\| = 1$ can represent a rotation
  - Four numbers plus one constraint $\rightarrow$ 3 DoF

- Geometrically, the shell of a 4D sphere

Slides credit: Hao Su

Unit Quaternion as Rotation

- Rotate a vector $\overrightarrow{x}$ by quaternion q:
  1. Augment $\overrightarrow{x}$ to $x = (0, \overrightarrow{x})$
  2. $x' = qxq^{-1}$
- Compose rotations by quaternion:
  - $(q_2(q_1 x q_1^*)q_2^*)$: first rotate by q_1 and then by q_2
  - Since $(q_2(q_1 x q_1^*)q_2^*) = (q_2 q_1)x(q_1^* q_2^*)$, we conclude that composing rotations is as simple as multiplying quaternions!

Slides credit: Hao Su

Conversation between Quaternions and Angle-Axis

- Exponential coordinate → Quaternion:

$$q = [\cos(\theta/2), \sin(\theta/2)\hat{\omega}]$$

Quaternion is very close to angle-axis representation!

- Exponential coordinate ← Quaternion:

$$\theta = 2\arccos(w), \qquad \hat{\omega} = \begin{cases} \dfrac{1}{\sin(\theta/2)}\overrightarrow{v} & \theta \neq 0 \\ 0 & \theta = 0 \end{cases}$$

Slides credit: Hao Su

Conversation between Quaternion and Rotation Matrix

- Rotation ← Quaternion

$$R(q) = E(q)G(q)^T$$

$$\text{where } E(q) = [-\vec{v}, wI + [\vec{v}]] \text{ and}$$

$$G(q) = [-\vec{v}, wI - [\vec{v}]]$$

- Rotation → Quaternion
  - Rotation → Angle-Axis → Quaternion

Slides credit: Hao Su

Inspection from Learning Perspective

- Each rotation corresponds to two quaternions (“double-covering”)
- Need to normalize to unit length in networks. This normalization may cause big/small gradients in practice

Slides credit: Hao Su

More about Quaternion

- Quaternion is computationally cheap:
  - Internal representation of Physical Engine and Robot
  - Pay attention to convention (w, x, y, z) or (x, y, z, w)?
  - (w, x, y, z): SAPIEN, transforms3d, Eigen, blender, MuJoCo, V-Rep
  - (x, y, z, w): ROS, PhysX, PyBullet

Slides credit: Hao Su

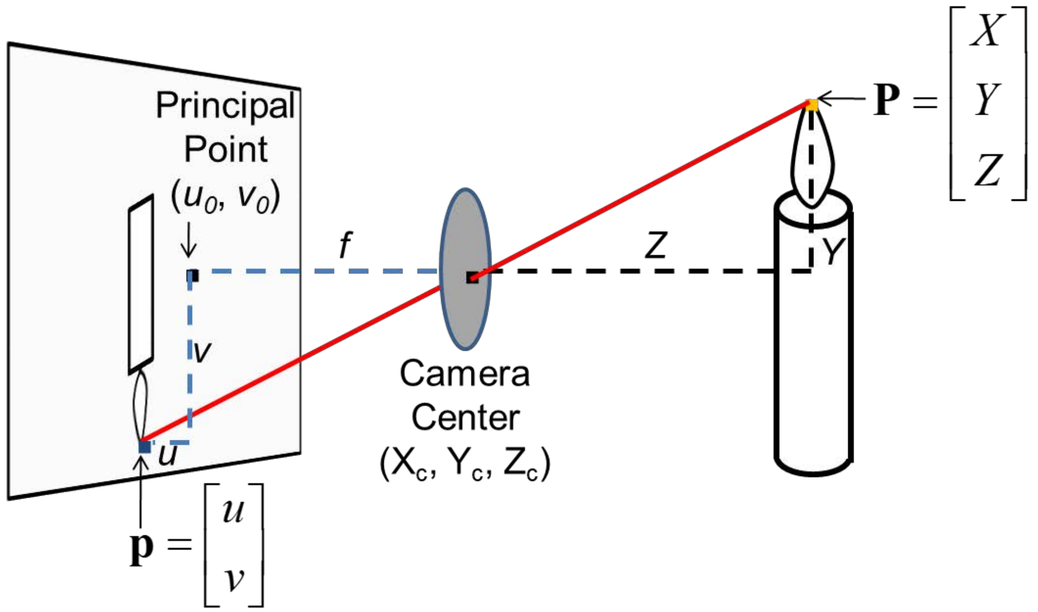
Summary of Quaternion

- Very useful and popular in practice
- 4D parameterization, compact and efficient to compute
- Good numerical properties in general

Outline Today

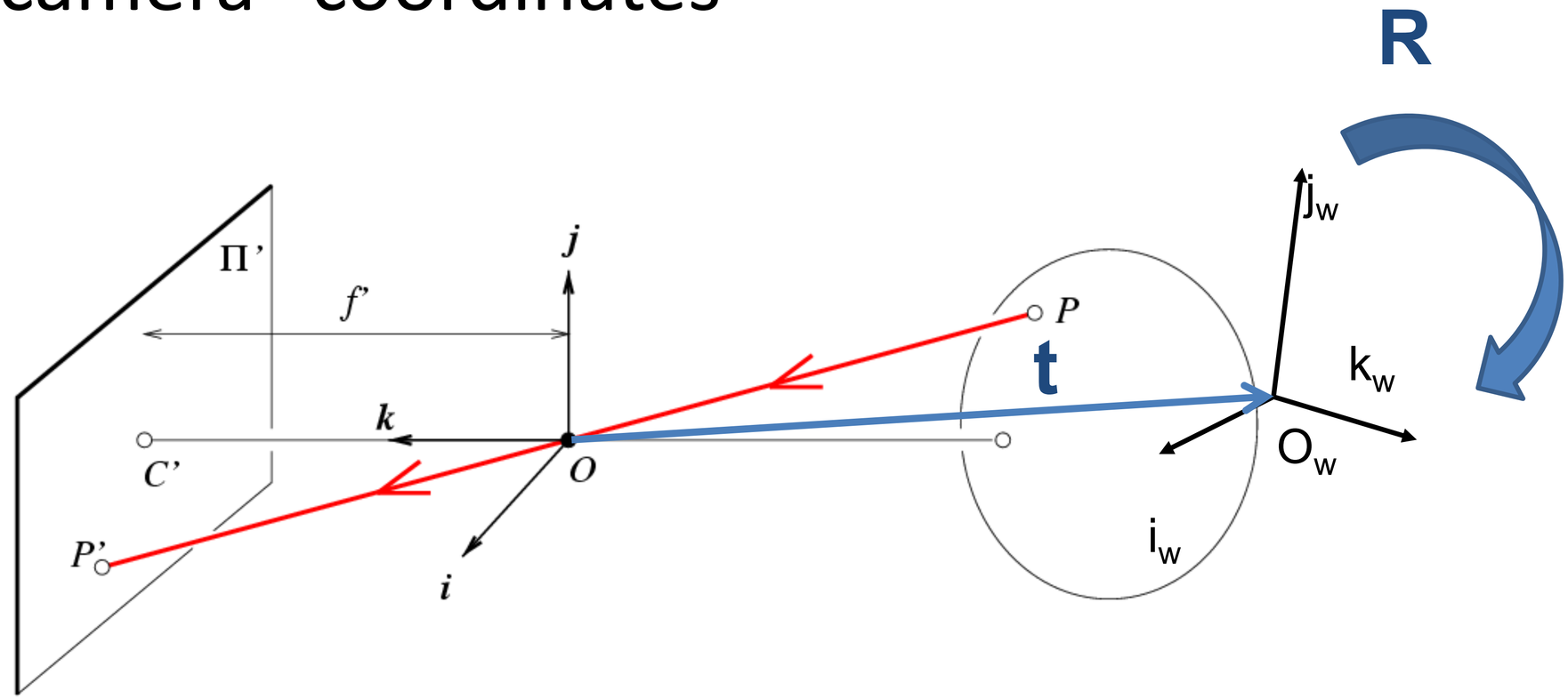
- 2D-3D basics
- Structure from motion
- Learning-based structure from motion

How do we map from 3D to 2D?



$$\mathbf{p} = \mathbf{K}\,\mathbf{P} \quad \Rightarrow \quad w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

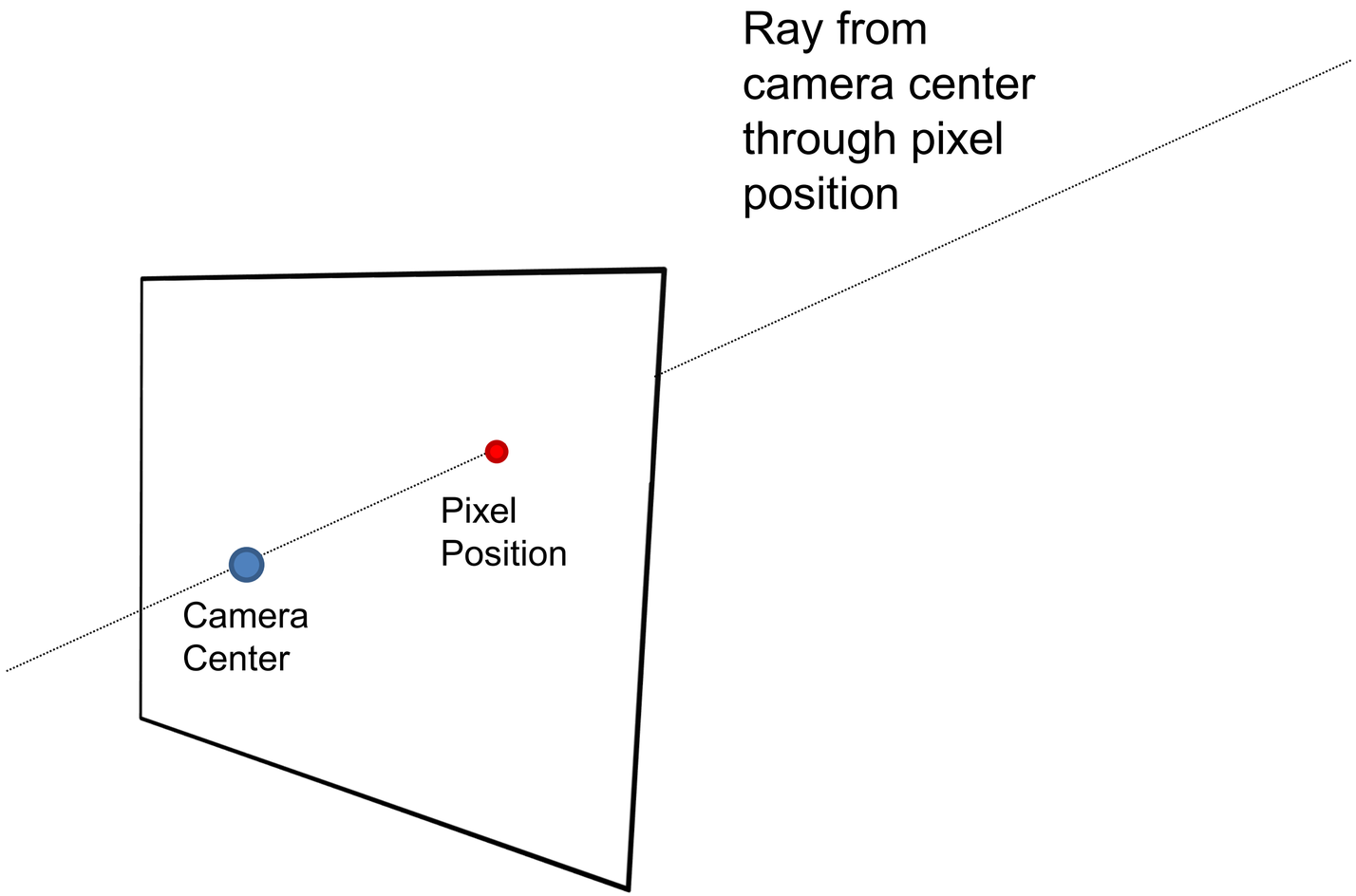
Rotation and translation map from “world” coordinates to “camera” coordinates



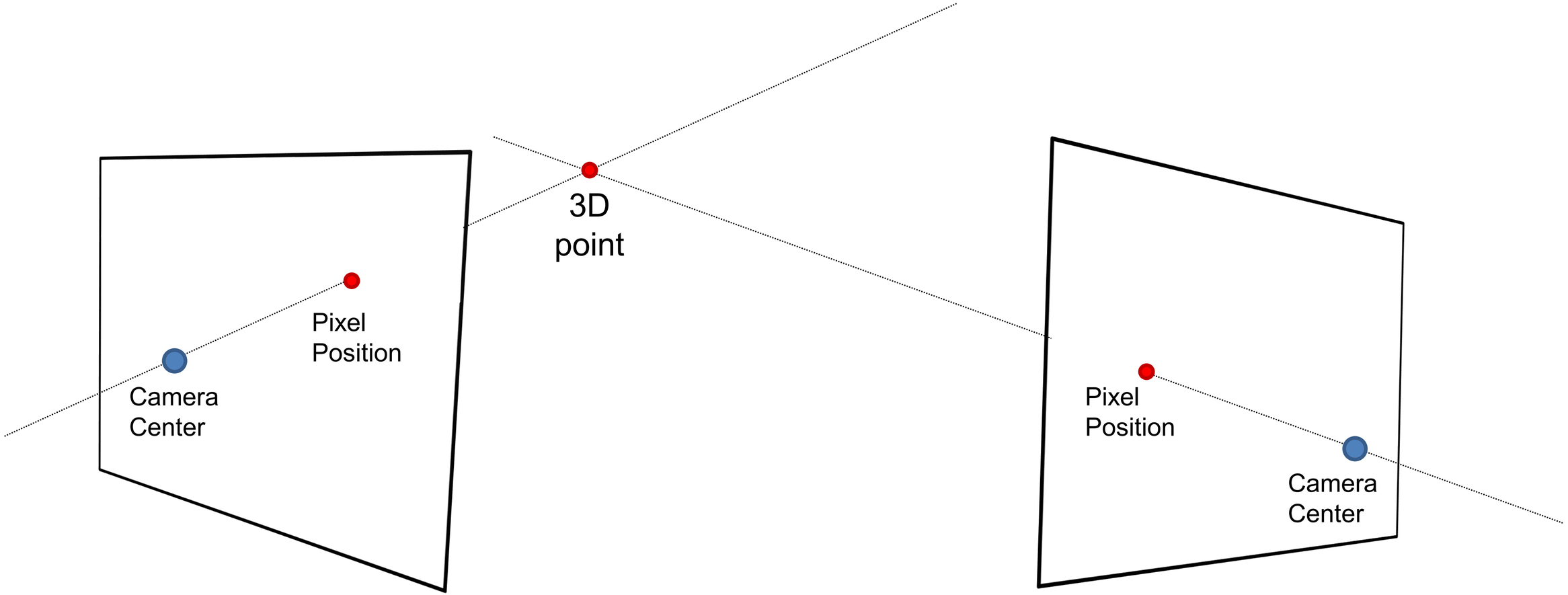
$$\mathbf{X}_c = \begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}_w \qquad \mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}_w$$

x: Image Coordinates: (u,v,1)
K: Intrinsic Matrix (3x3)
R: Rotation (3x3)
t: Translation (3x1)
$\mathbf{X}_w$: World Coordinates: (X,Y,Z,1)

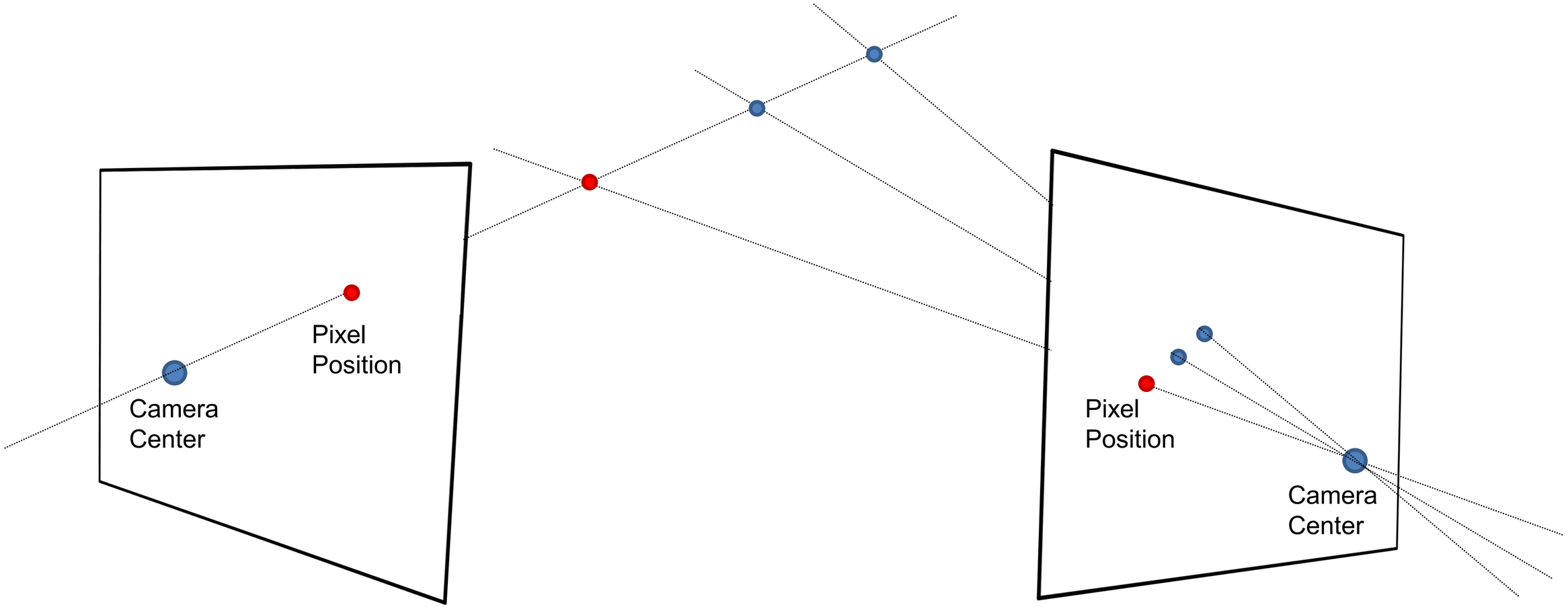
2D Pixel → 3D Ray through camera center and pixel



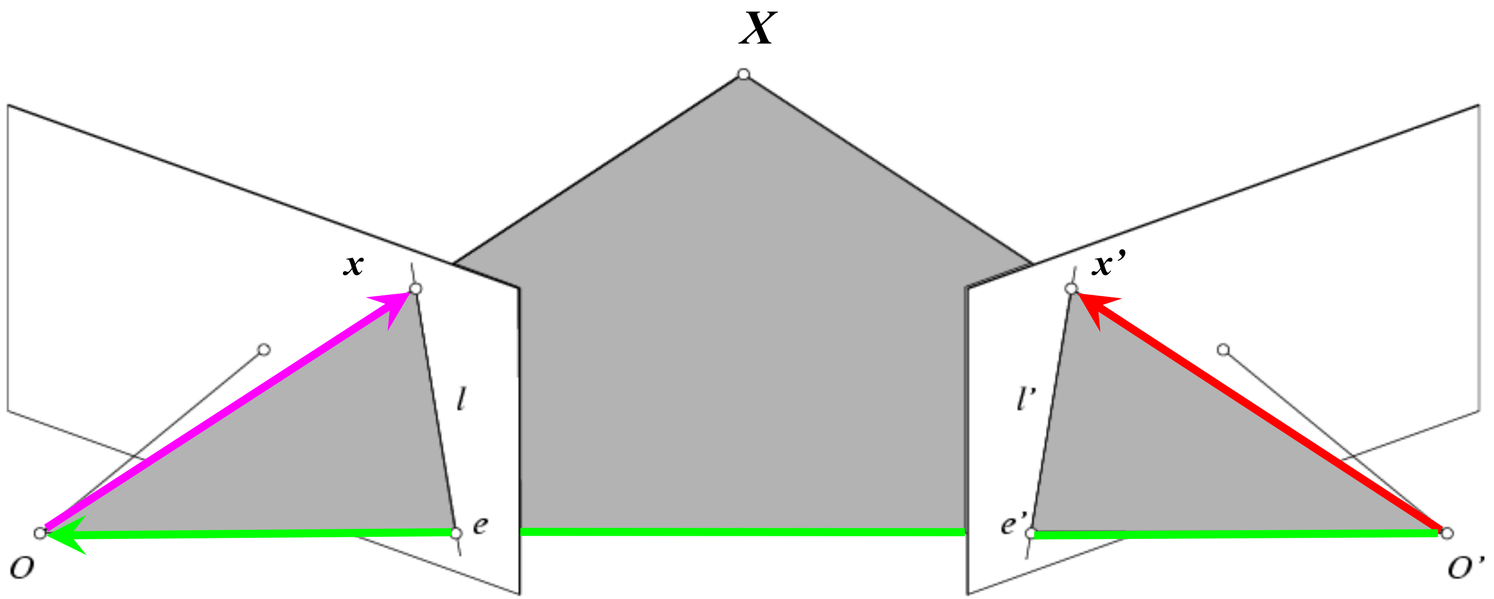
Intersection of two rays determines depth



All points on the ray from first camera project onto line in second camera



Essential matrix



$$\hat{x}\cdot[t\times(R\hat{x}')]=0 \quad \Rightarrow \quad \hat{x}^T E\hat{x}'=0 \quad \text{with} \quad E=[t]_\times R$$

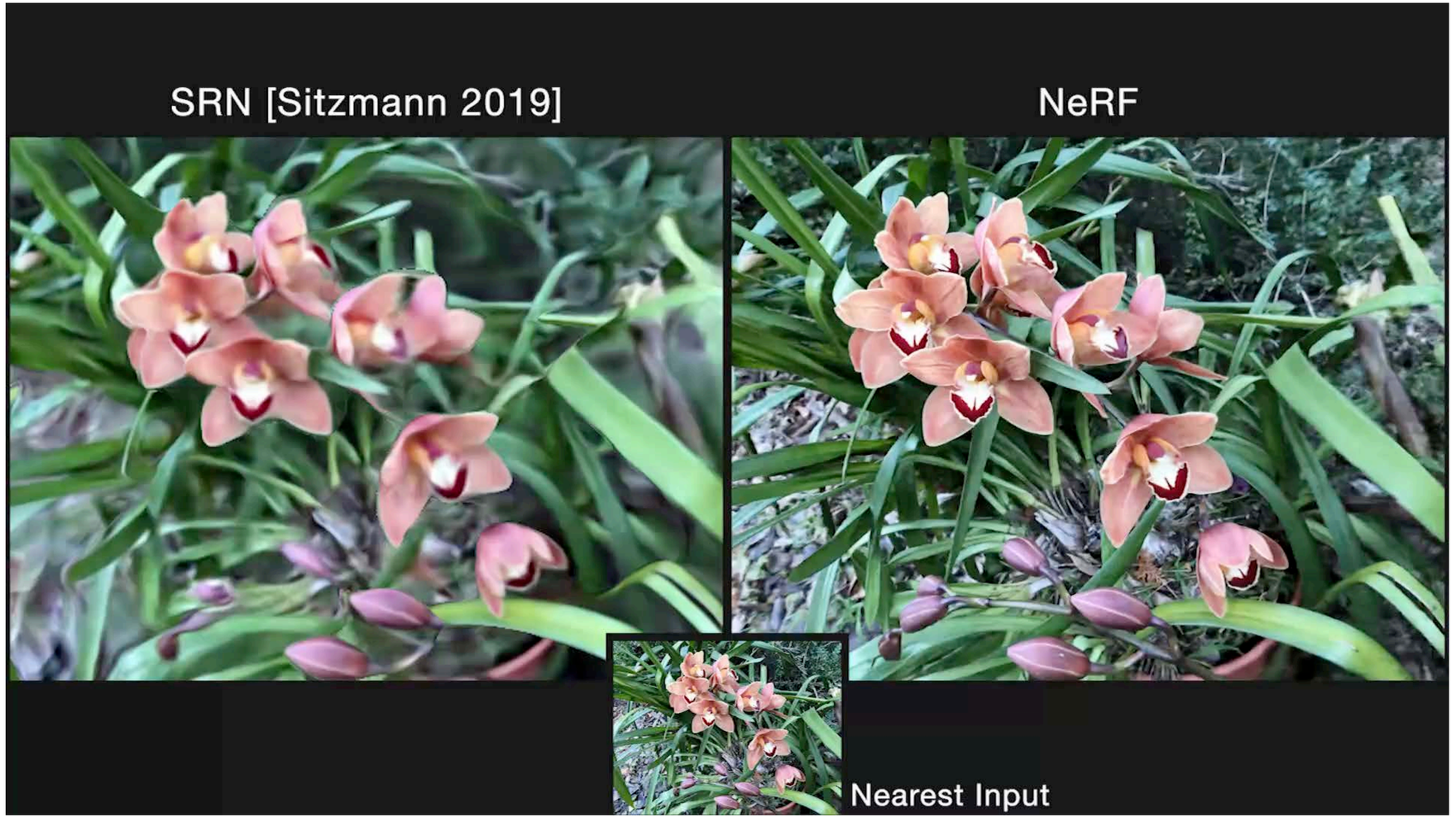
Essential Matrix
(Longuet-Higgins, 1981)

Outline Today

- 2D-3D basics
- Structure from motion
- Learning-based structure from motion

Colosseum

SfM systems are used for 3D reconstruction at scale



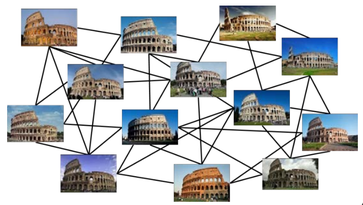
and are a critical preprocessing step in recent neural reconstruction

SfM Pipeline

Unstructured Images

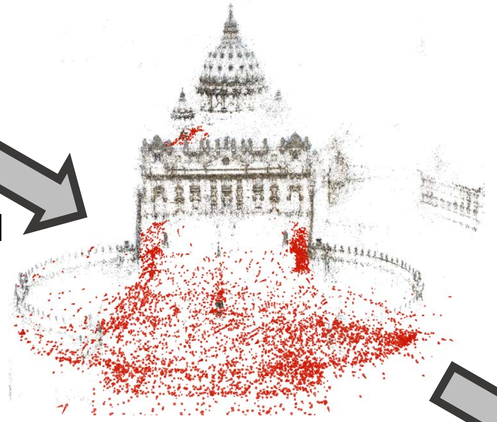
Assoc.

Scene Graph



SFM

Sparse Model

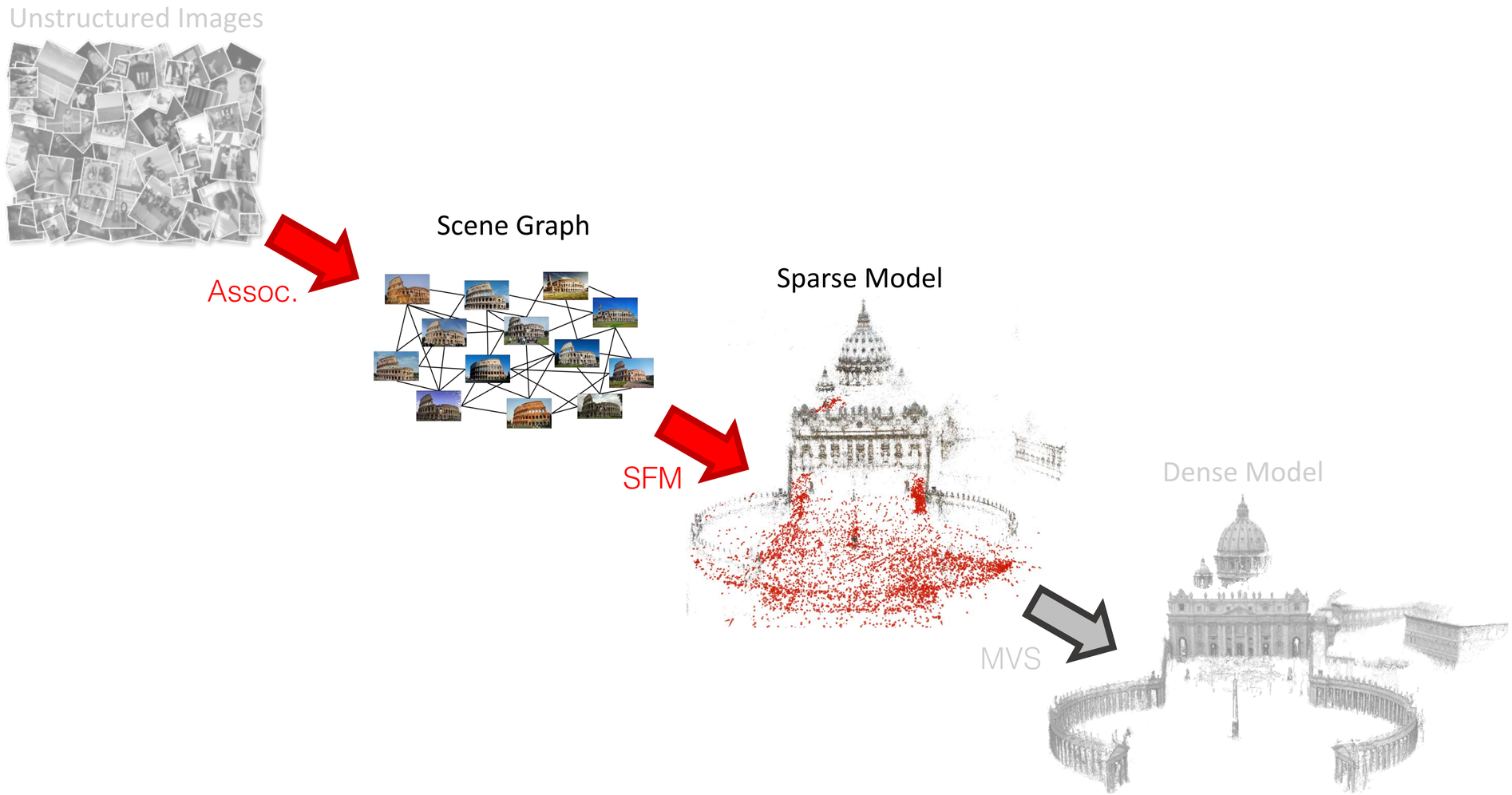


MVS

Dense Model

Slide credits: Johannes Schönberger

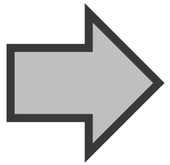
SfM Pipeline



Slide credits: Johannes Schönberger

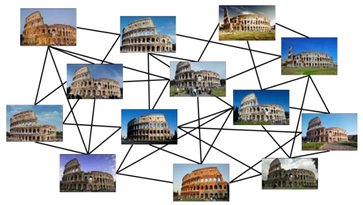
Data Association

Unstructured Images



Connected Components

Scene Graph



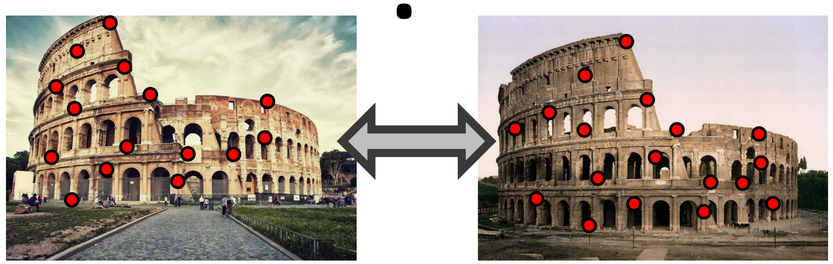
Slide credits: Johannes Schönberger

Data Association

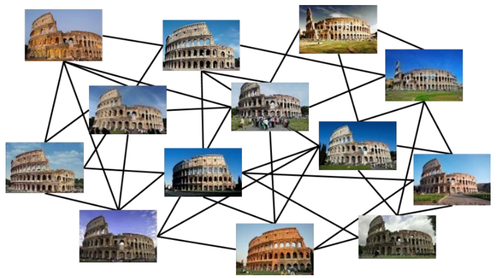
Unstructured Images

Two-View Geometry

?



Scene Graph



Outputs:

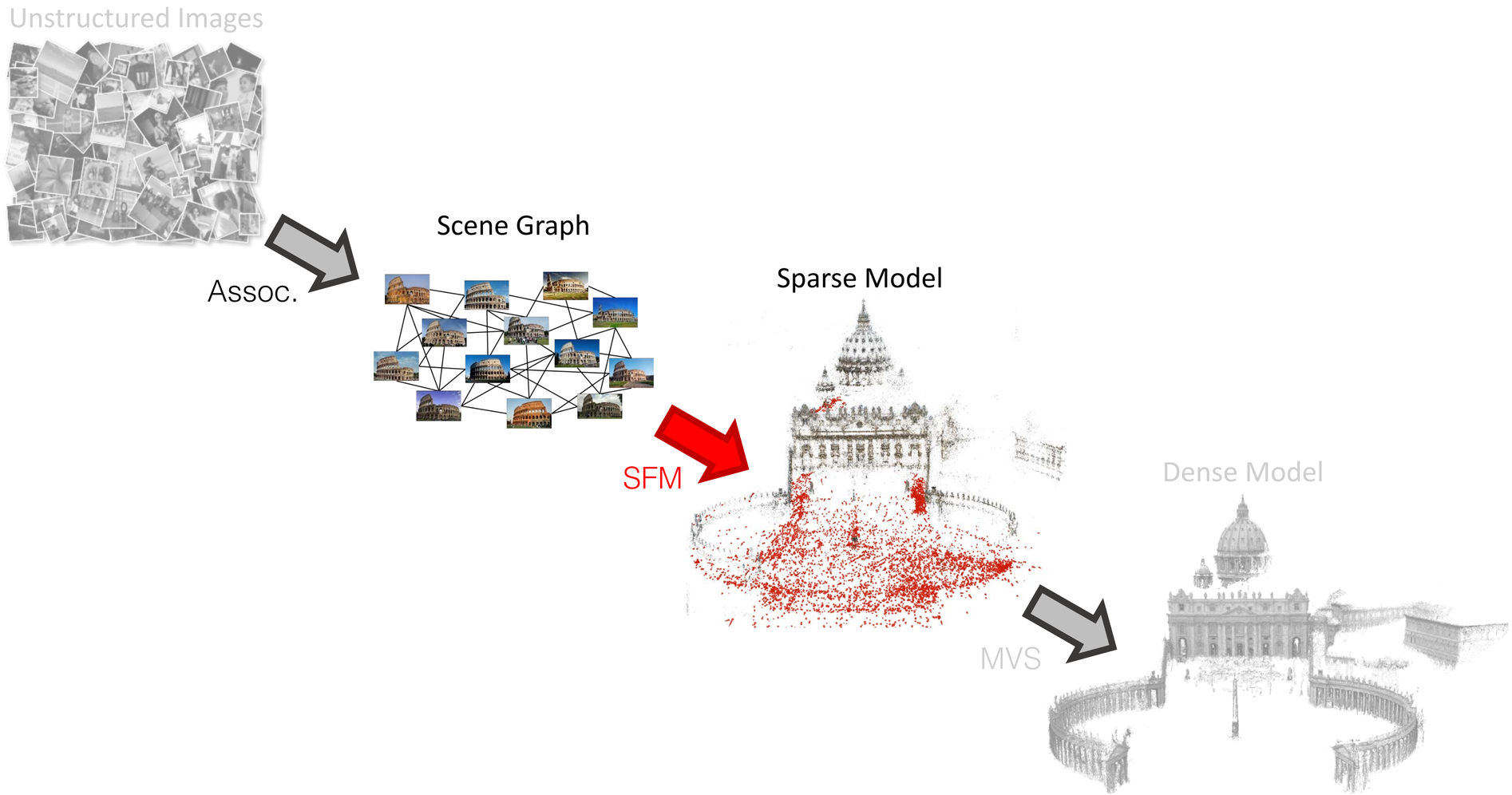
1) identified pairs of overlapping images

2) geometrically verified inlier matches (and optionally, feature descriptors for later use)

3) related camera poses (if known calibration)

Slide adapted from Johannes Schönberger

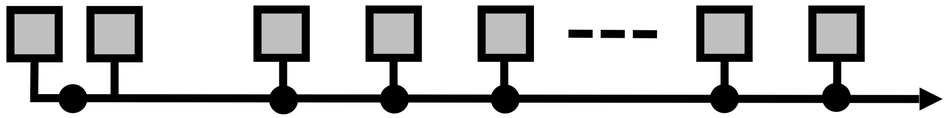
SfM Pipeline



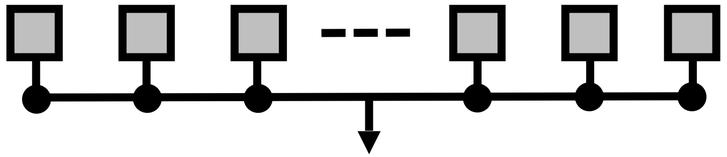
Slide credits: Johannes Schönberger

Structure from Motion

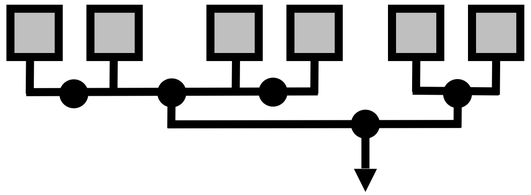
- 3 paradigms
  - Incremental



  - Global

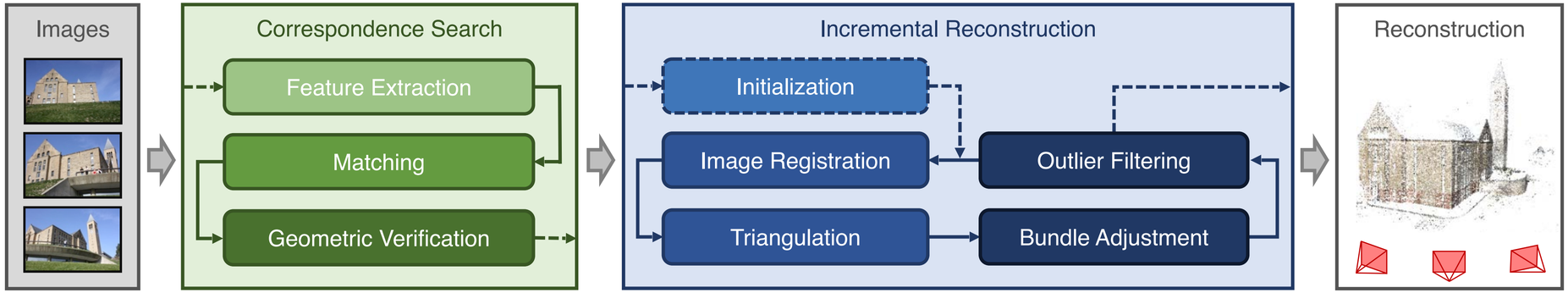


  - Hierarchical



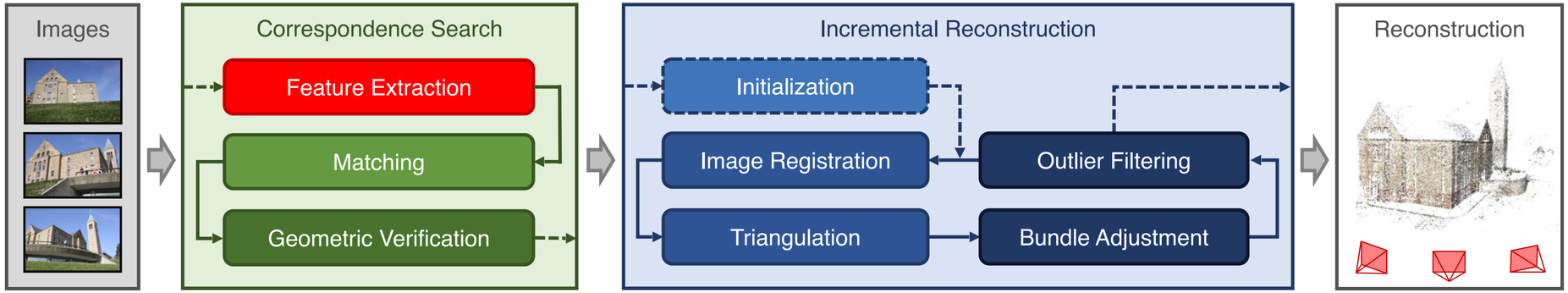
Slide credits: Johannes Schönberger

Incremental SfM



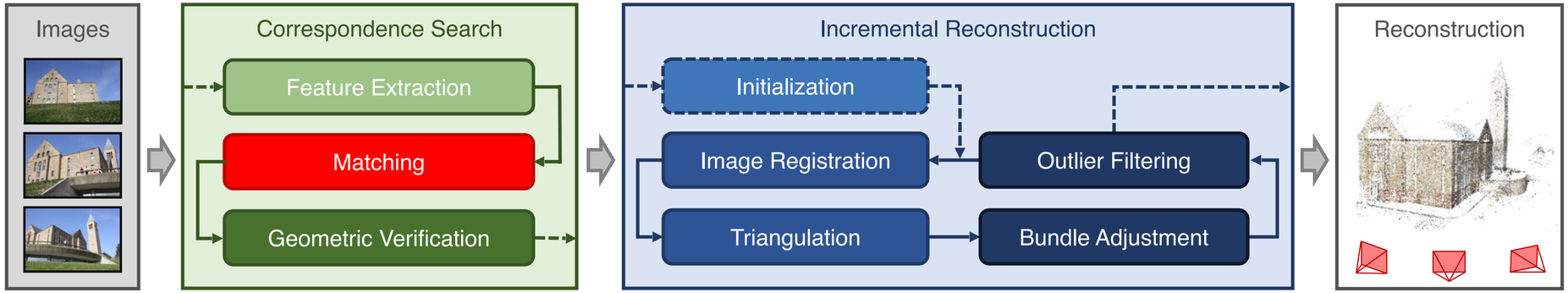
Slide credits: Johannes Schönberger

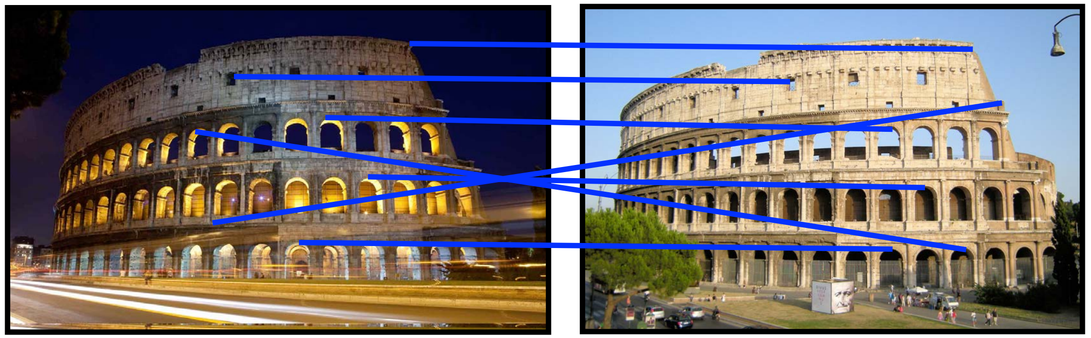
Incremental SfM



Slide credits: Johannes Schönberger

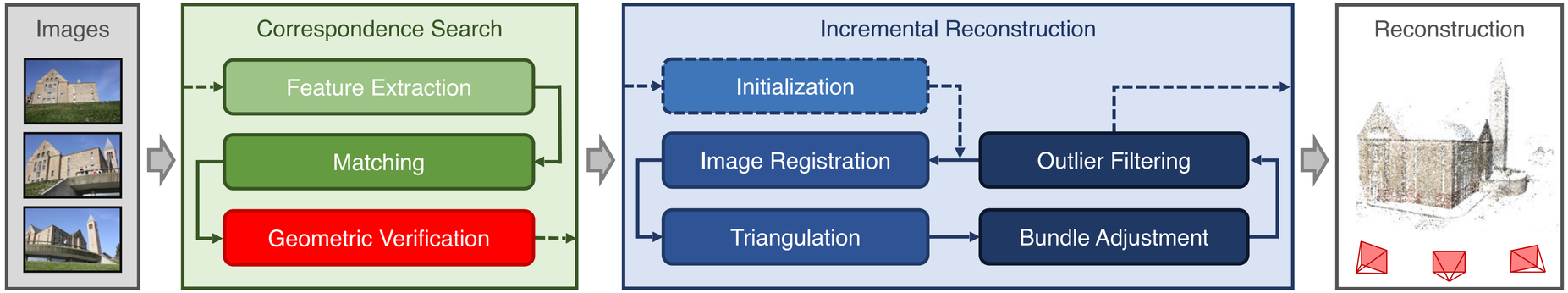
Incremental SfM



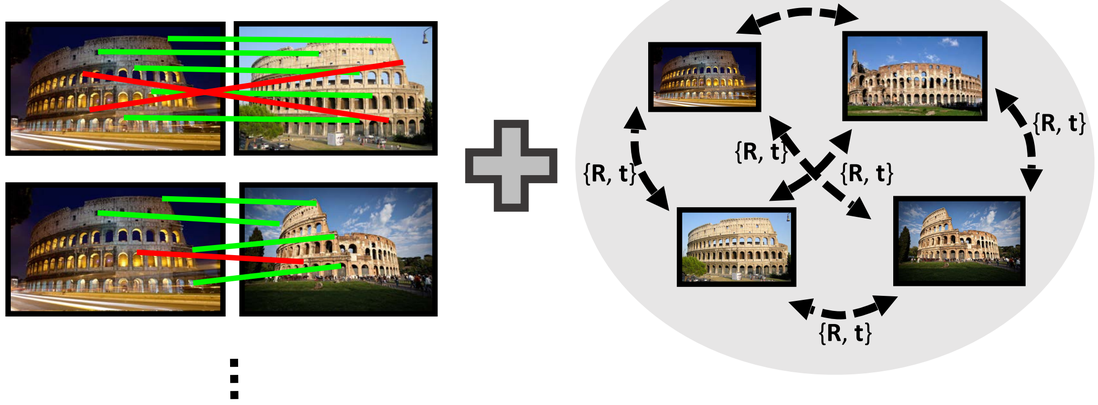


Slide credits: Johannes Schönberger

Incremental SfM

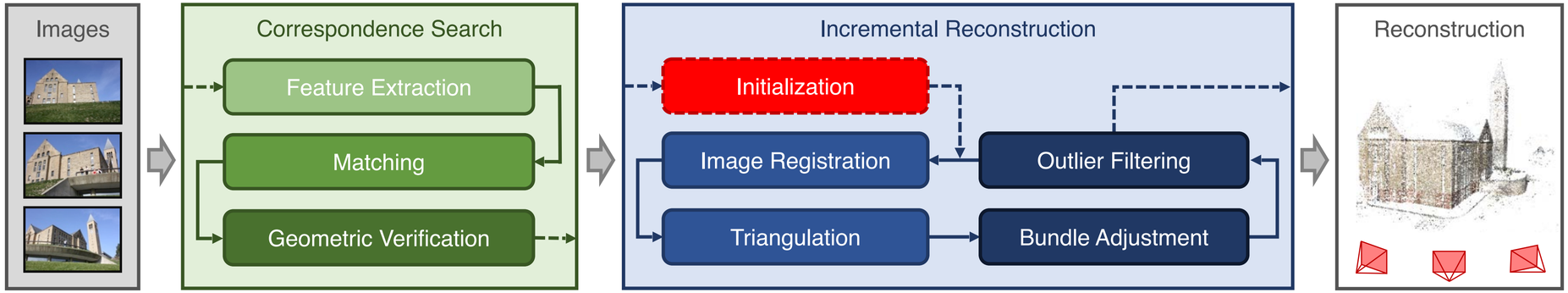


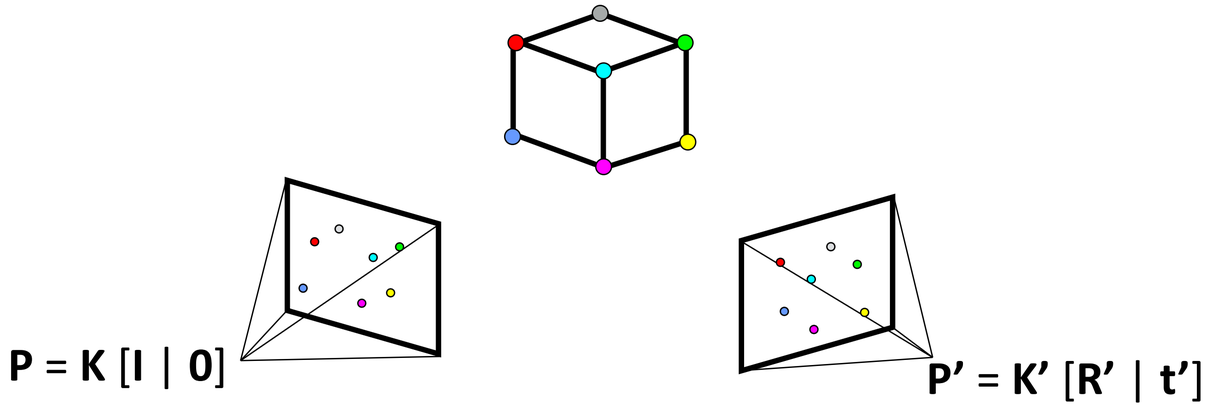
Inlier/outlier correspondences



Slide credits: Johannes Schönberger

Incremental SfM

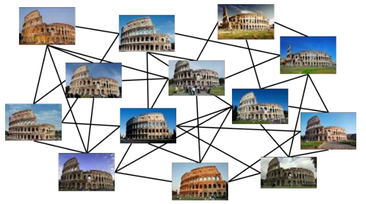


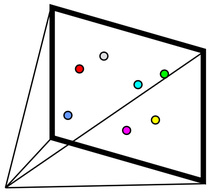


Slide credits: Johannes Schönberger

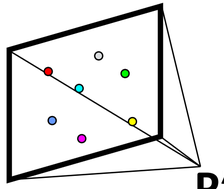
Incremental SfM

- Initialization
  1. Choose two non-panoramic views ($\|t\| \neq 0$)





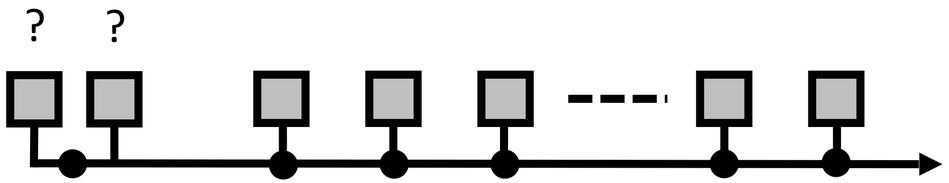
P = K [I | 0]



P’ = K’ [R’ | t’]

Slide credits: Johannes Schönberger

Incremental SfM: Initial-view Selection



Trade-off:

- Triangulation angle
- Num. correspondences

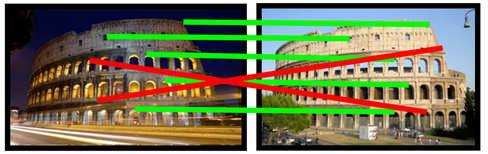
Should have sufficient correspondences

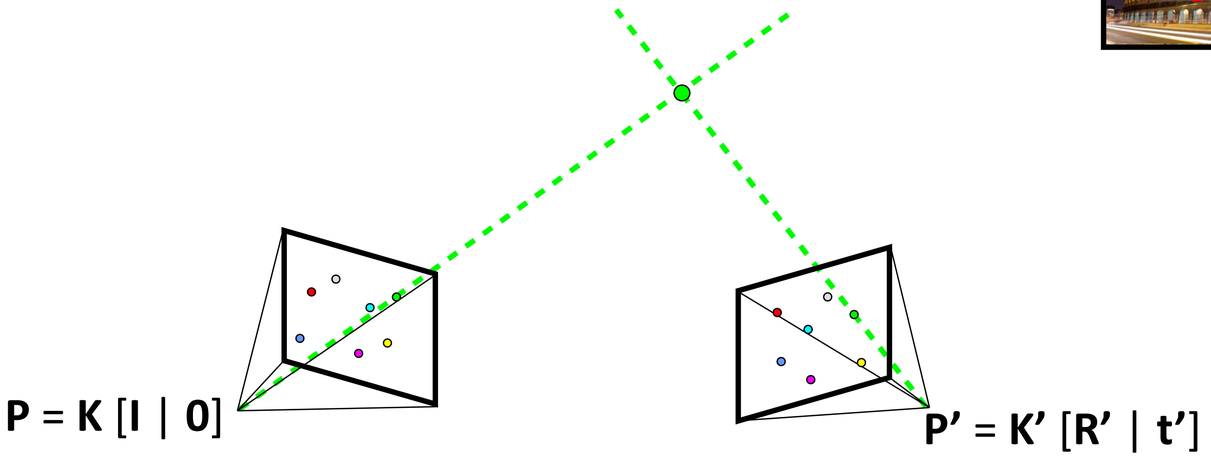
But views should also be slightly far apart — low ‘triangulation angle’ i.e. 3D point viewed from similar directions can lead to errors in triangulation

Slide credits: Johannes Schönberger

Incremental SfM

- Initialization
  1. Choose two non-panoramic views ($\|t\| \neq 0$)
  2. Triangulate inlier correspondences

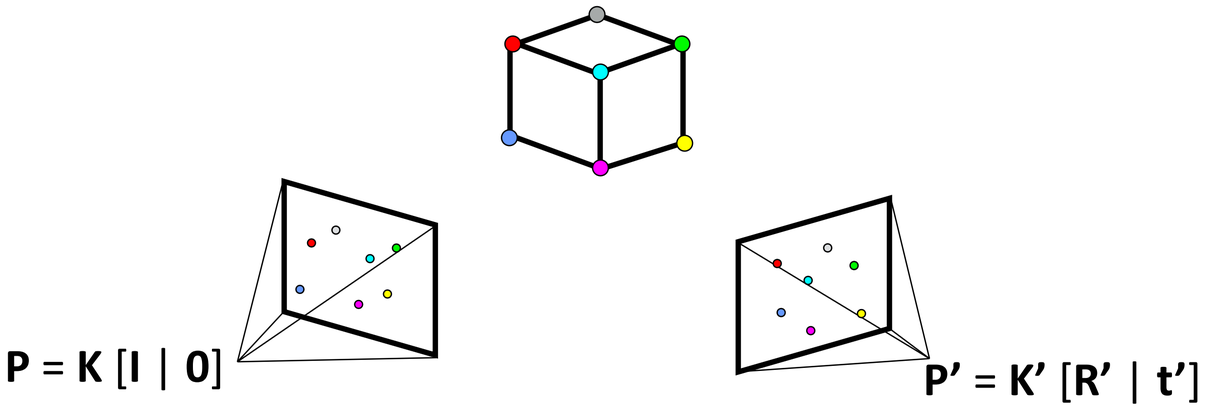




Slide credits: Johannes Schönberger

Incremental SfM

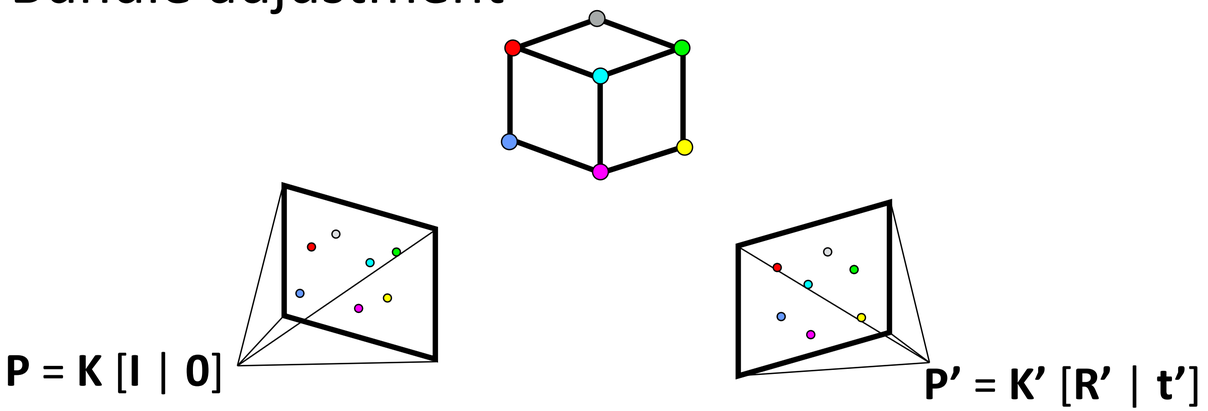
- Initialization
  1. Choose two non-panoramic views ($\|t\| = 1$)
  2. Triangulate inlier correspondences



Slide credits: Johannes Schönberger

Incremental SfM

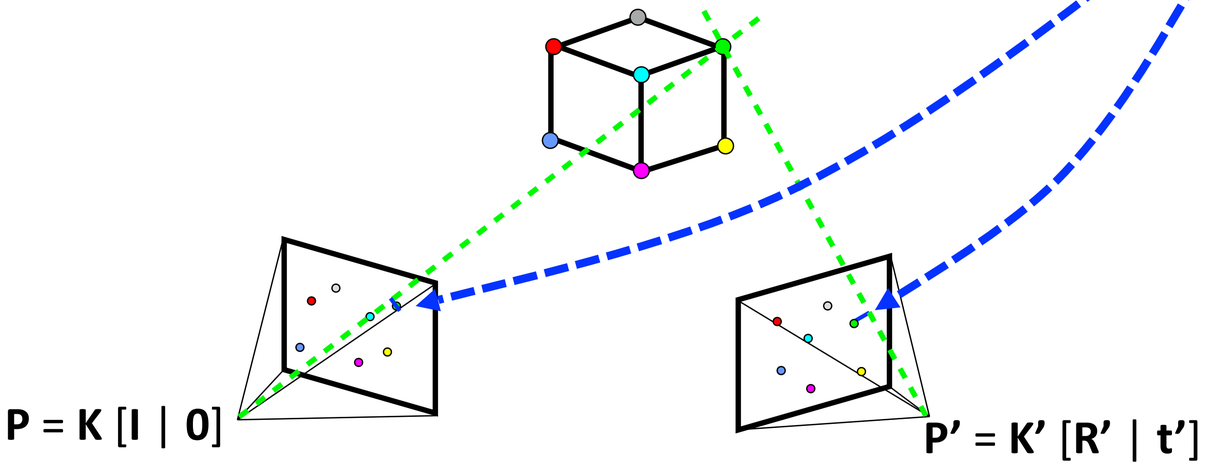
- Initialization
  1. Choose two non-panoramic views ($\|t\| = 1$)
  2. Triangulate inlier correspondences
  3. Bundle adjustment



Slide credits: Johannes Schönberger

Incremental SfM

- Bundle adjustment
  - Non-linear refinement of structure and motion
  - Minimize reprojection error: $\min_{P,\ X} \left\| x - \pi(P,\ X) \right\|$

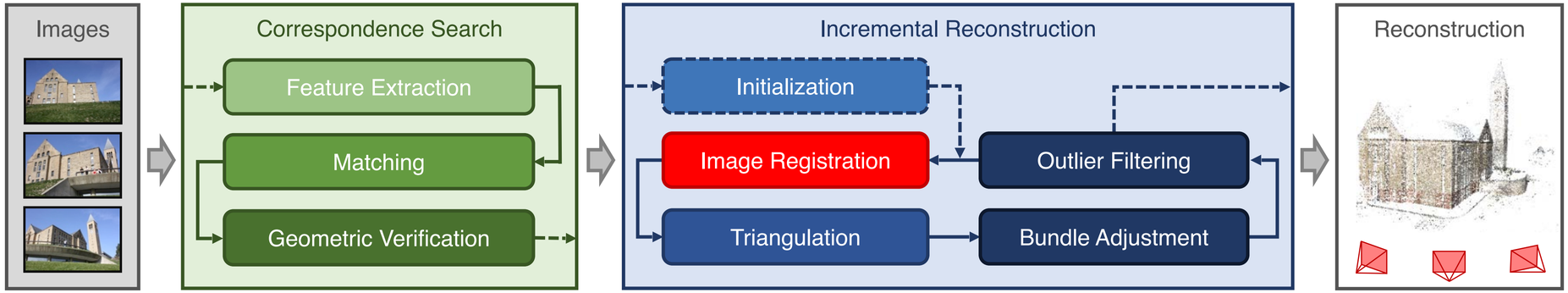


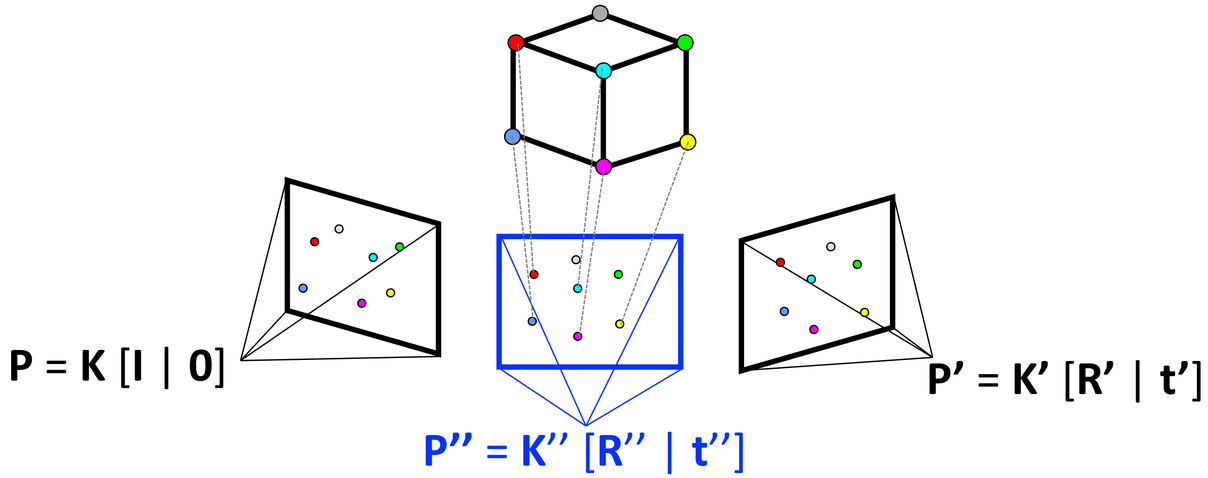
Ceres-Solver, http://ceres-solver.org/

Triggs et al., "Bundle Adjustment – A Modern Synthesis"

Slide credits: Johannes Schönberger

Incremental SfM

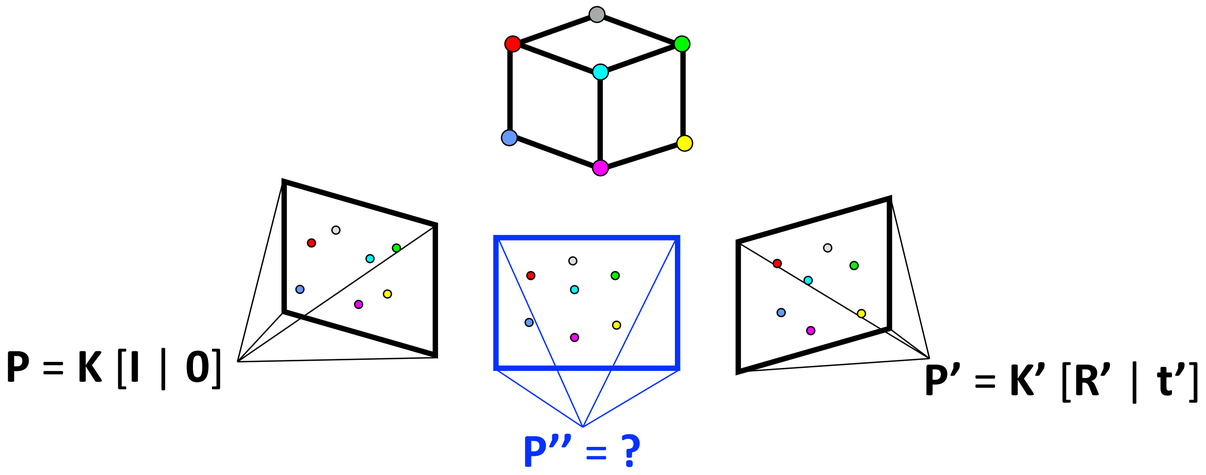




Slide credits: Johannes Schönberger

Incremental SfM

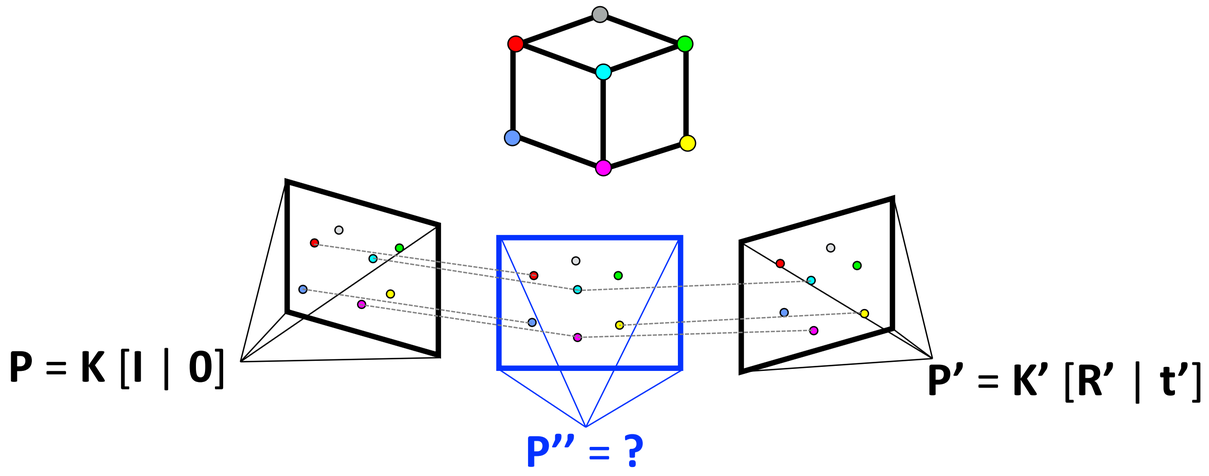
- Absolute camera registration



Slide credits: Johannes Schönberger

Incremental SfM

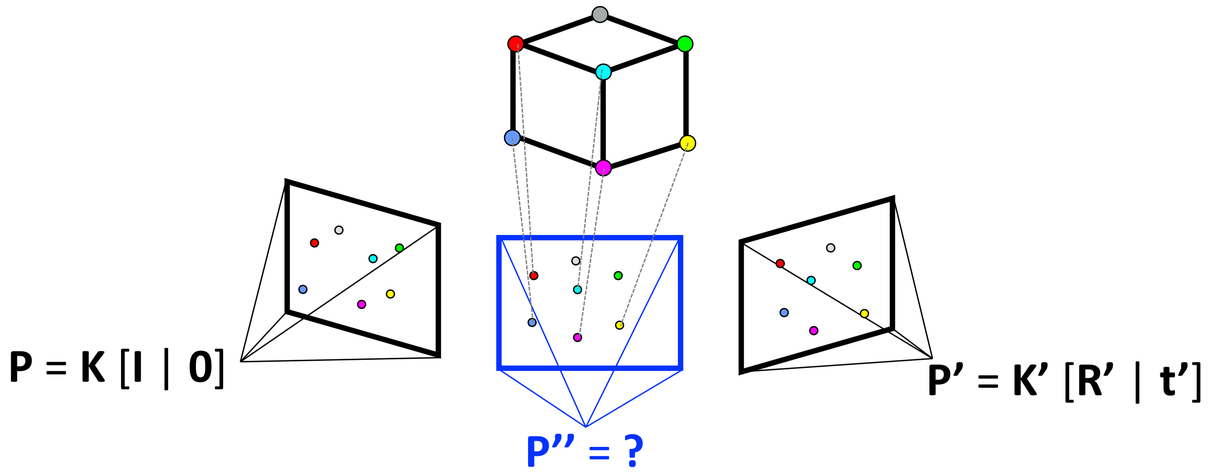
- Absolute camera registration
  1. Find 2D-3D correspondences



Slide credits: Johannes Schönberger

Incremental SfM

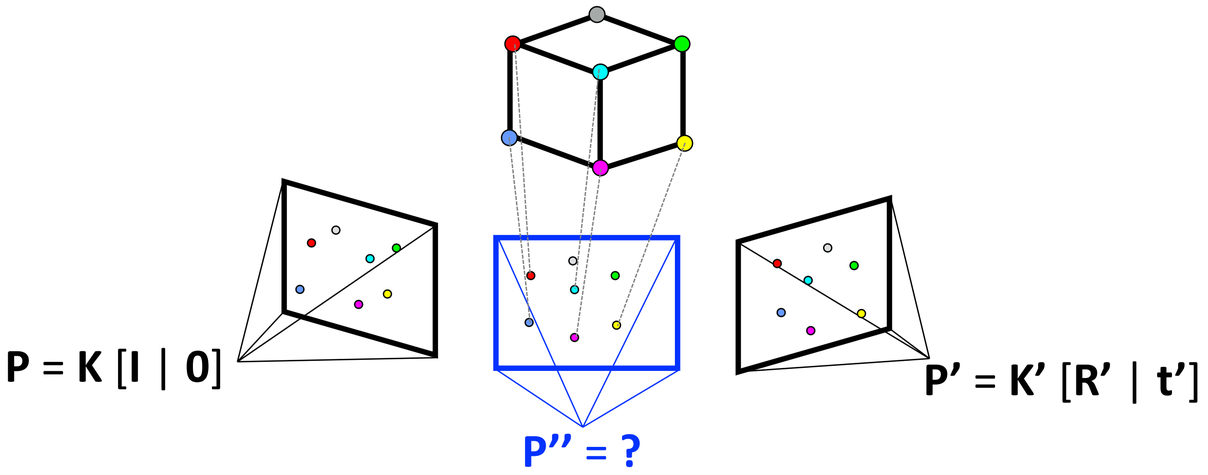
- Absolute camera registration
  1. Find 2D-3D correspondences



Slide credits: Johannes Schönberger

Incremental SfM

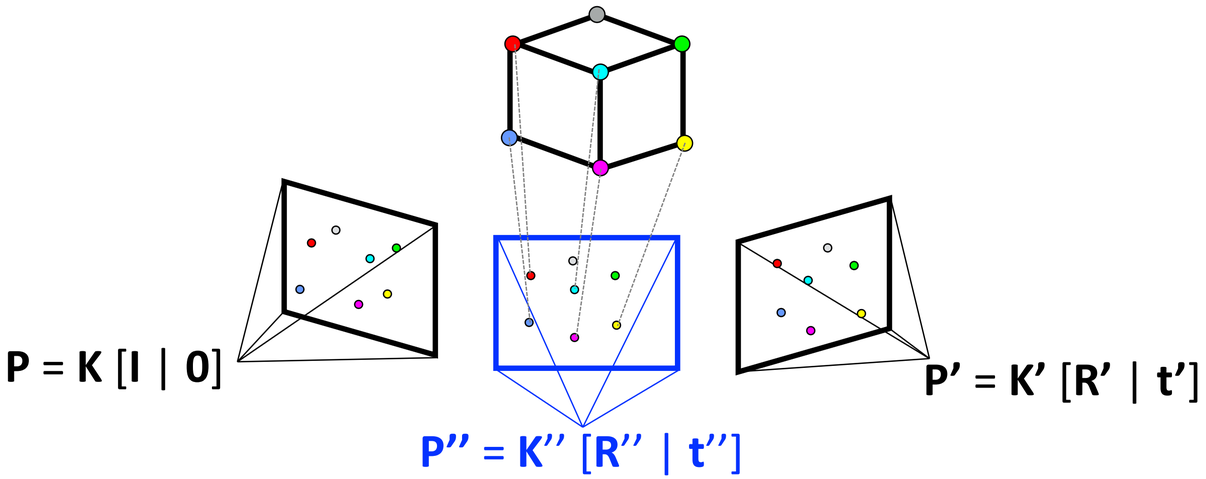
- Absolute camera registration
  1. Find 2D-3D correspondences
  2. Solve Perspective-n-Point problem



Slide credits: Johannes Schönberger

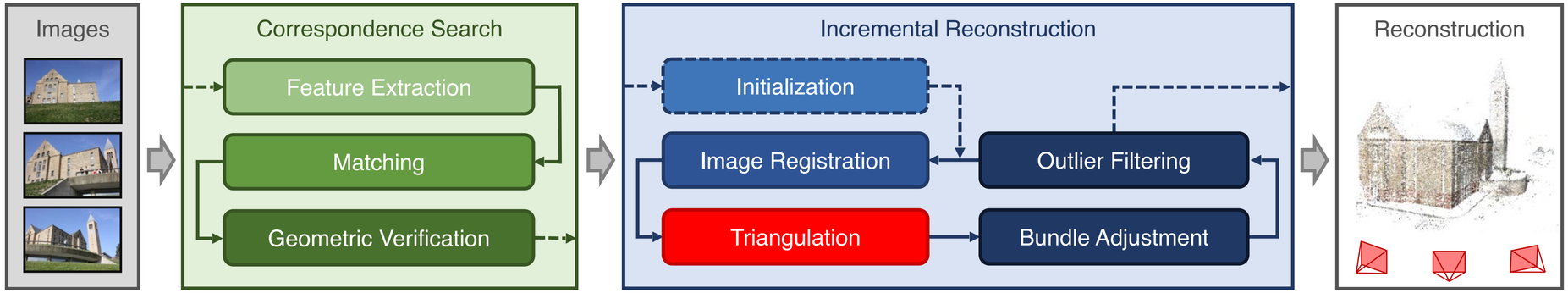
Incremental SfM

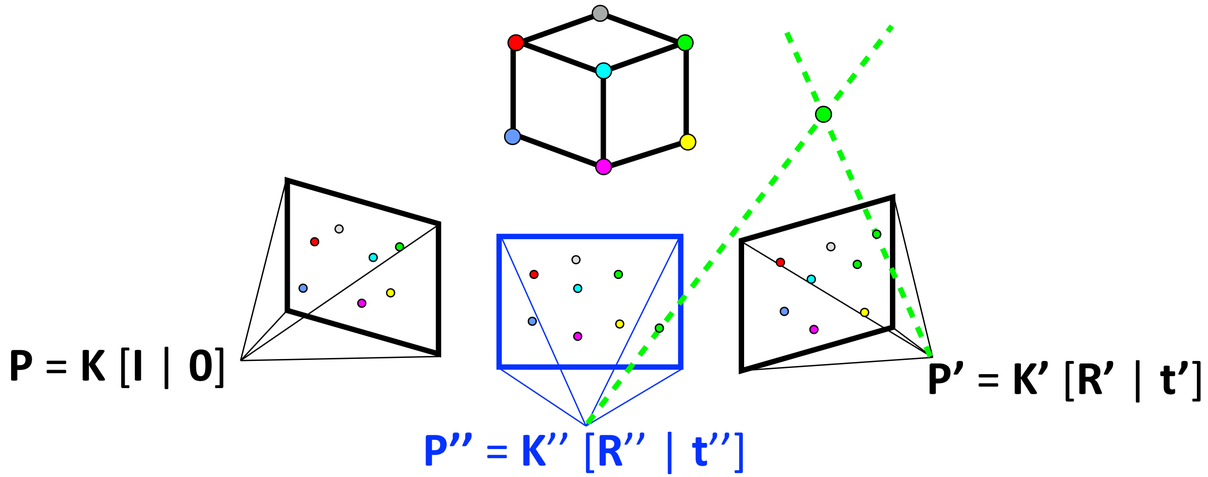
- Absolute camera registration
  1. Find 2D-3D correspondences
  2. Solve Perspective-n-Point problem



Slide credits: Johannes Schönberger

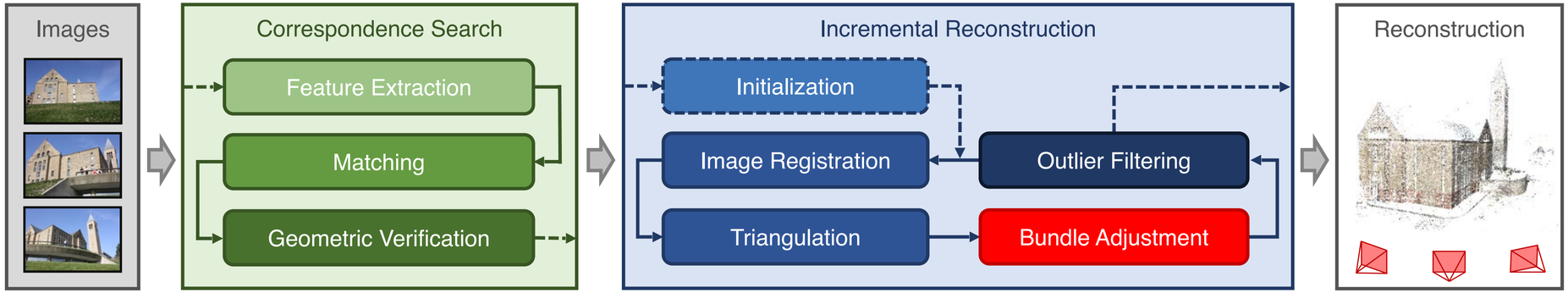
Incremental SfM





Slide credits: Johannes Schönberger

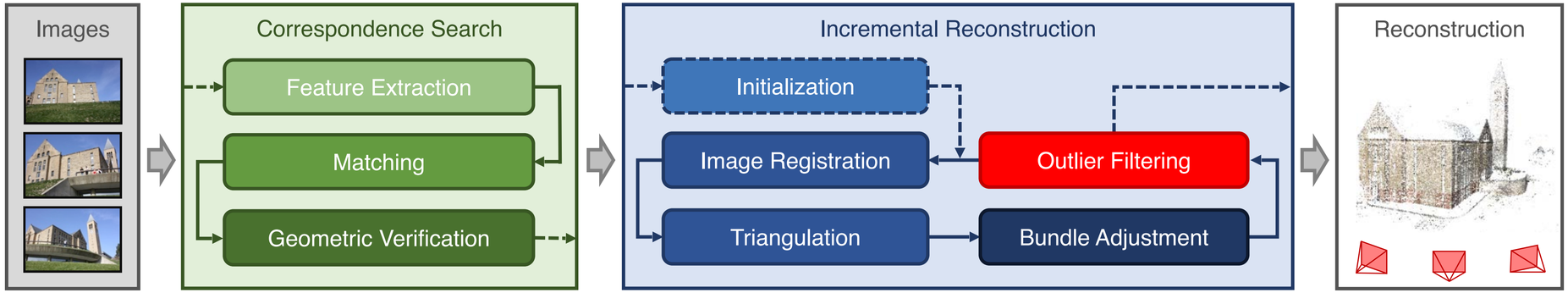
Incremental SfM

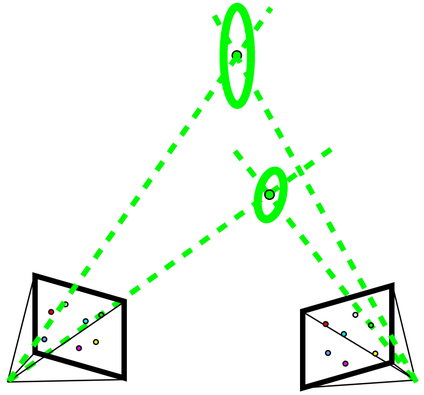


$$\min_{P,\ X} \left\| x - \pi(P,\ X) \right\|$$

Slide credits: Johannes Schönberger

Incremental SfM

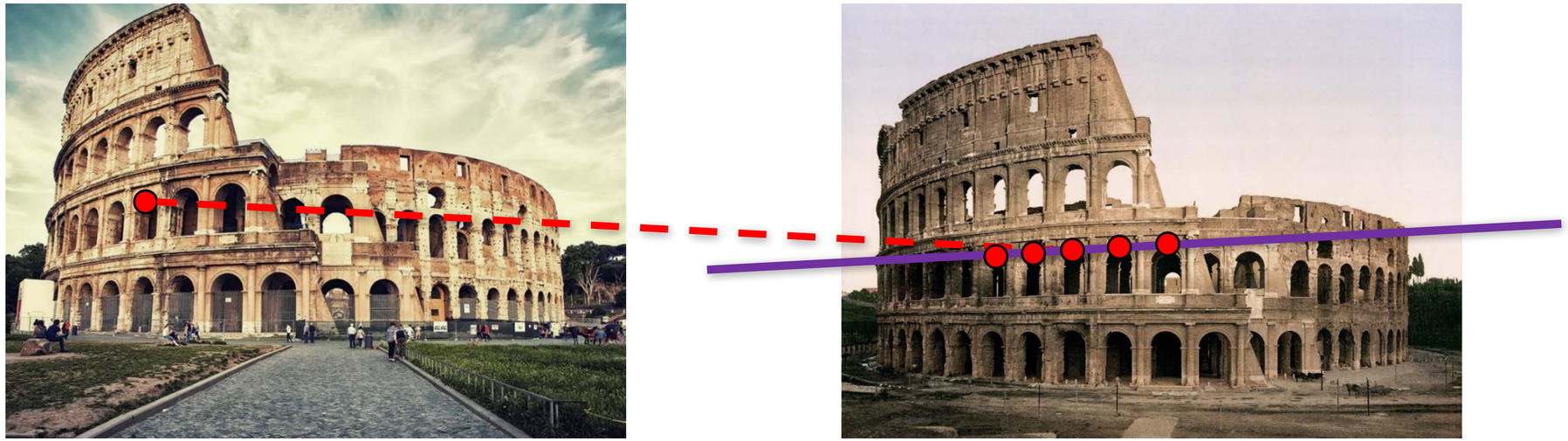




Slide credits: Johannes Schönberger

Incremental SfM

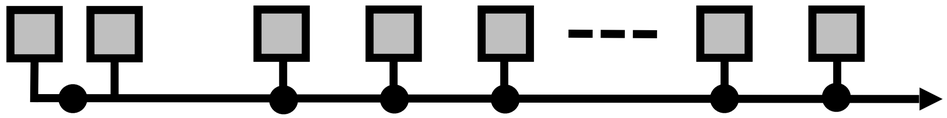
- Outlier filtering
  - Remove points with large reprojection error (and points at infinity)



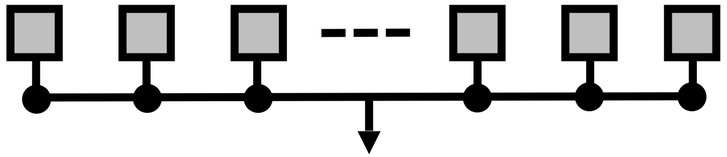
Slide credits: Johannes Schönberger

Structure from Motion

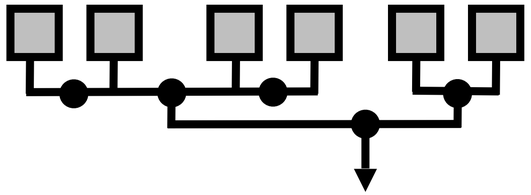
- 3 paradigms
  - Incremental



  - Global

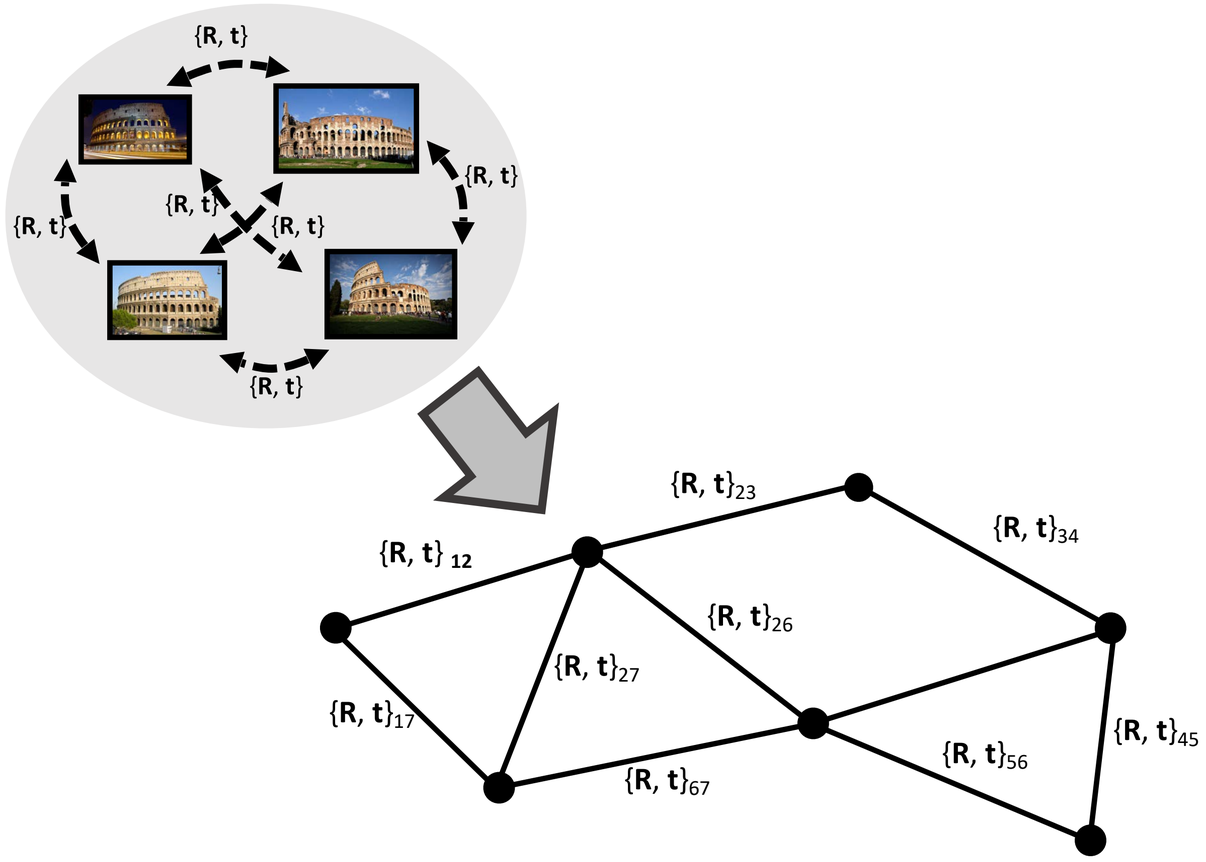


  - Hierarchical



Slide credits: Johannes Schönberger

Global SfM



Slide credits: Johannes Schönberger

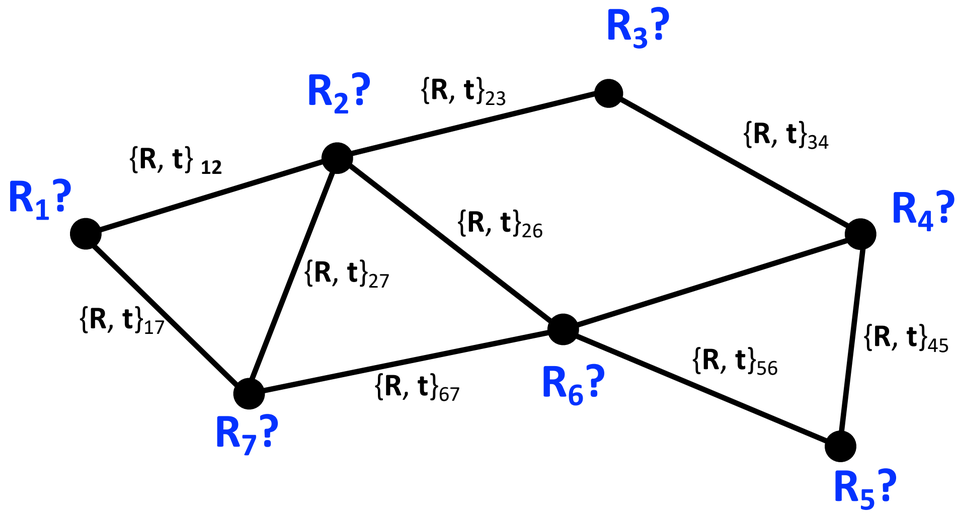
Global SfM

Image

Two-View Geometry

- Estimate global rotations: $\min_{R} \left\| Rij - RjRiT \right\|$

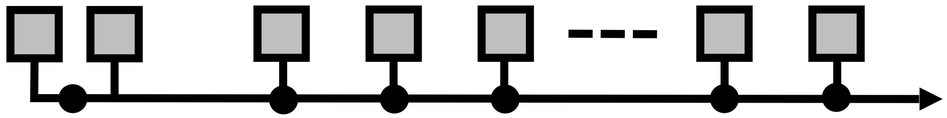
Chatterje and Govindu 2013, “Efficient and Robust Large-Scale Rotation Averaging”



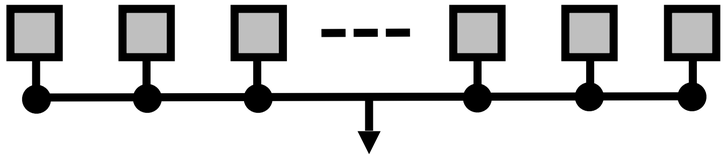
Slide credits: Johannes Schönberger

Structure from Motion

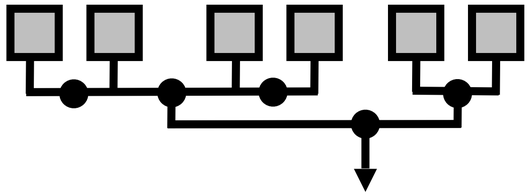
- 3 paradigms
  - Incremental

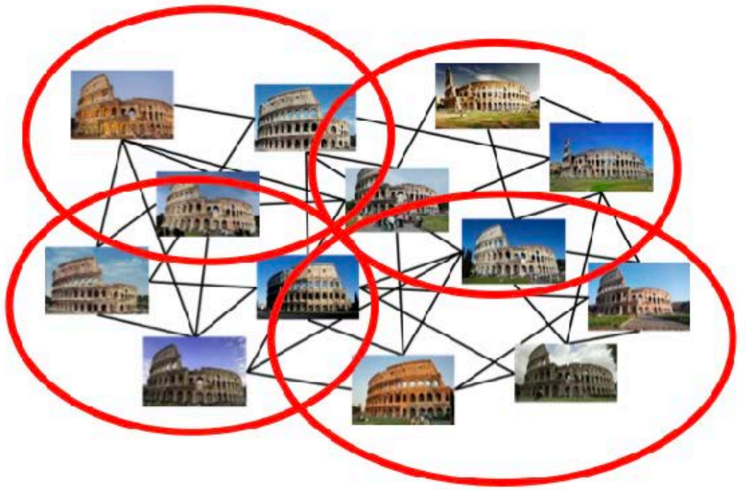


  - Global



  - Hierarchical





Slide credits: Johannes Schönberger

SfM and SLAM: similarities

- Solve for camera poses and 3D scene points given images

- Correspondence, registration, outlier rejection, and bundle adjustment are core problems

SfM vs. SLAM: differences

SfM

- Input is unordered set of images
- Focus is on precision, with aim to produce a good 3D model
- Offline, one-time process
- Published mainly in vision conferences
- Complicated

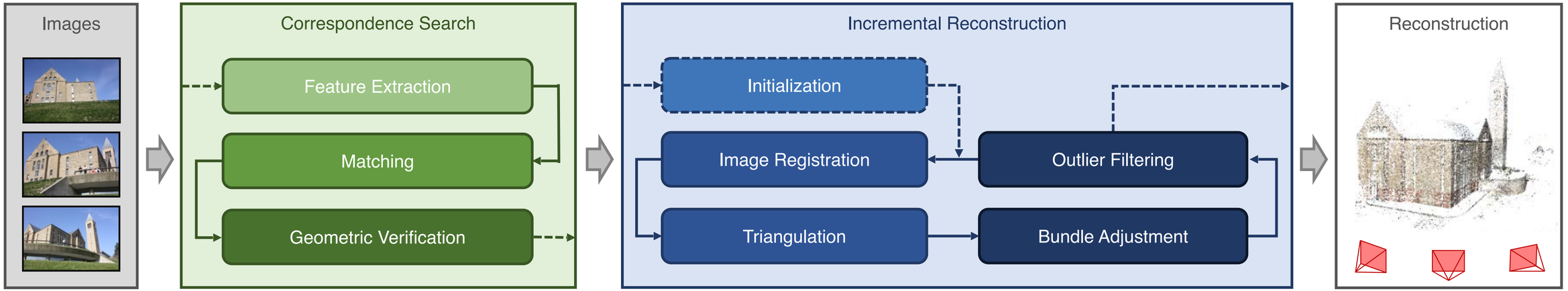
SLAM

- Input is stream of images, stereo, or depth and sometimes IMU
- Focus is on speed and robustness, with aim to localize camera or robot
- Online process, possibly with relocalization
- Published mainly in robotics conferences
- Very complicated

Outline Today

- 2D-3D basics
- Structure from motion
- Learning-based structure from motion

Learning for SfM



Can we improve the robustness/precision via data-driven learning?

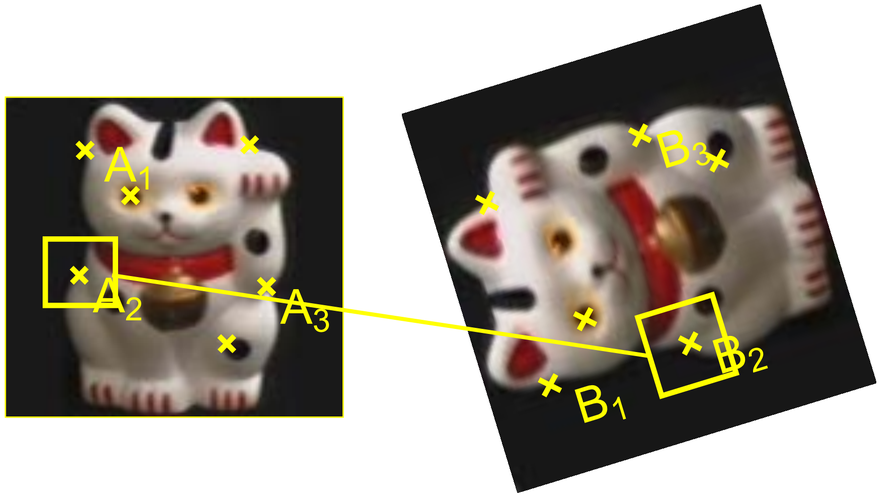
Example 1: Improving features and keypoints for matching

Example 2: Improving the matching process via global reasoning

Image credits: Johannes Schönberger

What makes for good keypoints?

Points should be
repeatable and *distinctive*



Detection

More Repeatable

Robust detection
Precise localization

More Points

Robust to occlusion
Works with less texture

Description

More Distinctive

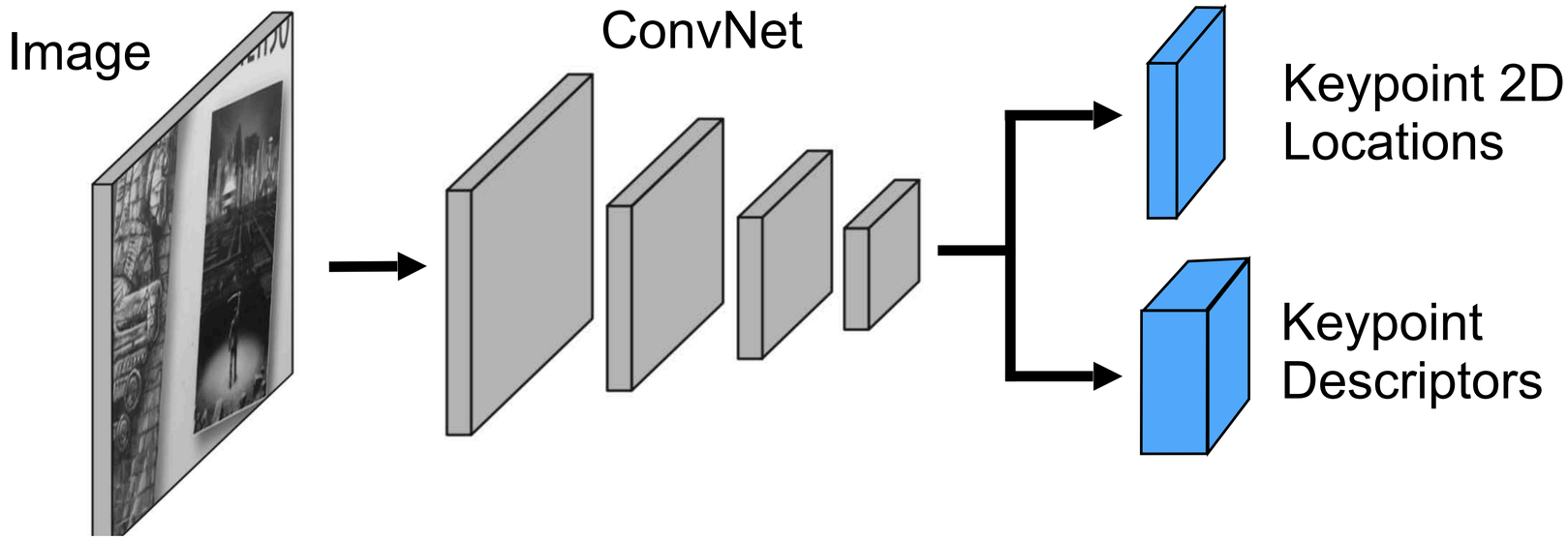
Minimize wrong matches

More Flexible

Robust to expected variations
Maximize correct matches

Slide adapted from: Derek Hoiem

SuperPoint: A Learned Detector and Descriptor



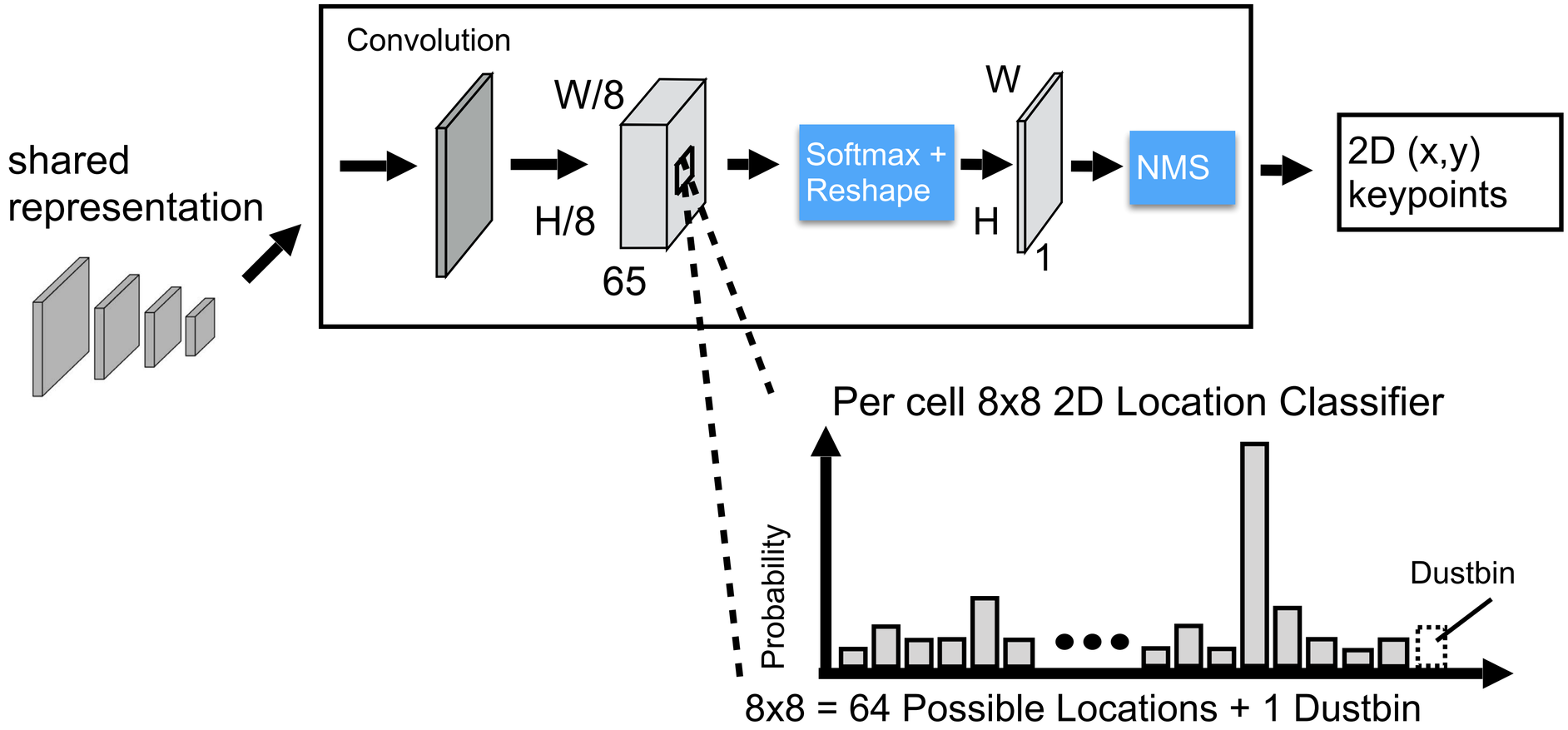
Key Challenges

How to get training data?

How to maintain precision i.e. detection at high resolution?

Slide adapted from: Derek Hoiem & T. Malisiewicz

SuperPoint: Detector

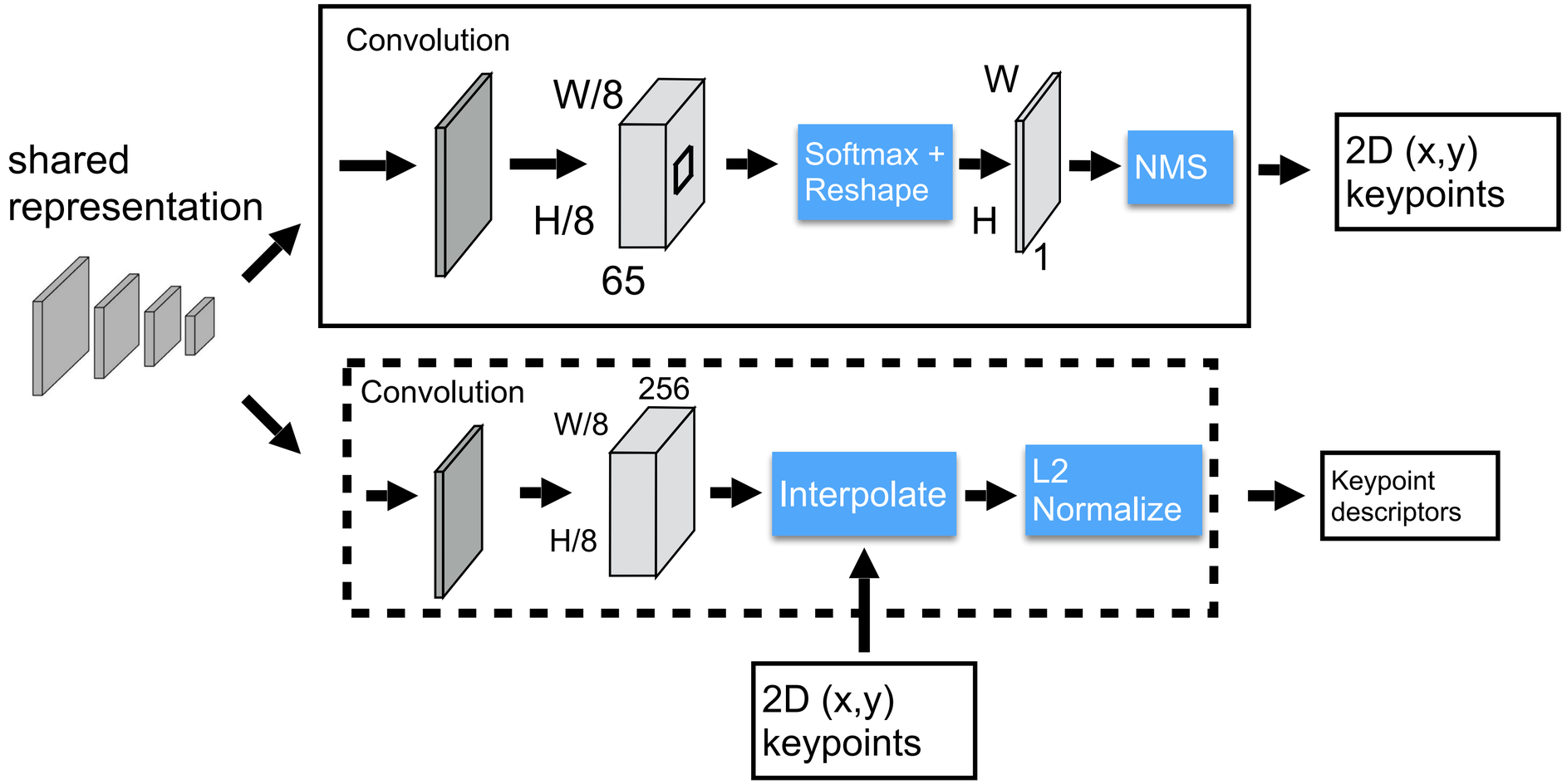


No upsampling layers

Each output cell responsible for a 8 X 8 pixel patch

Slide adapted from: Derek Hoiem & T. Malisiewicz

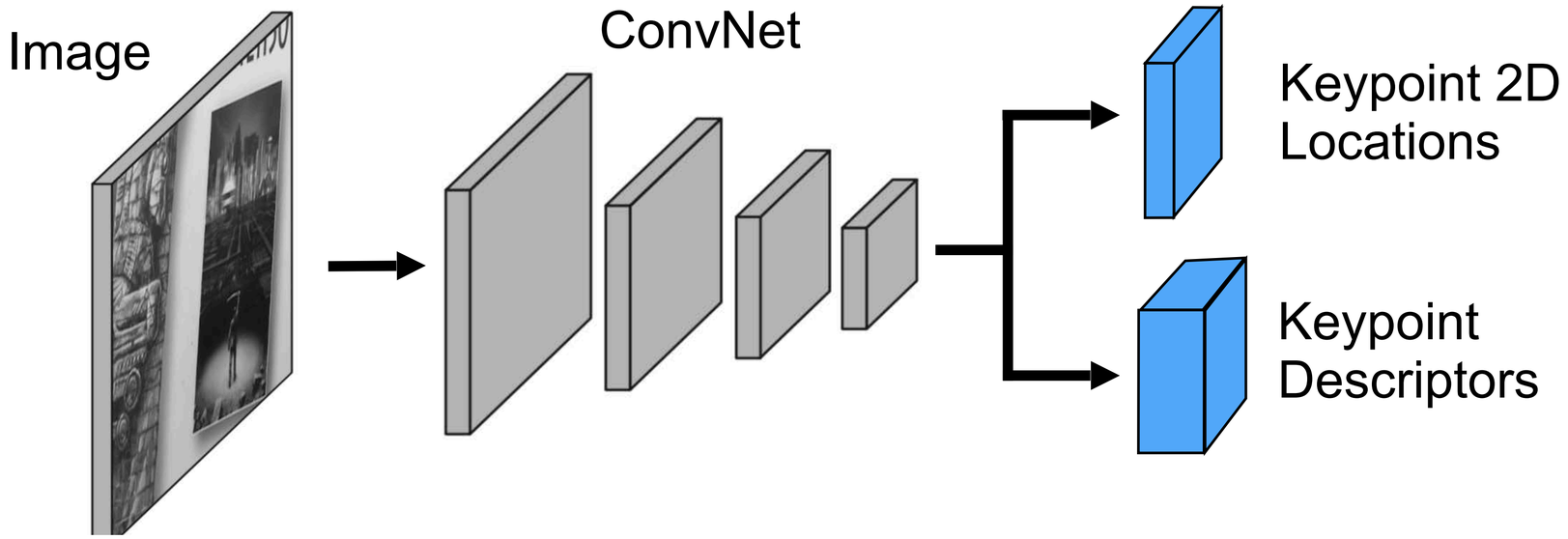
SuperPoint: Detector



Bilinear interpolation using keypoint locations to get descriptors

Slide adapted from: Derek Hoiem & T. Malisiewicz

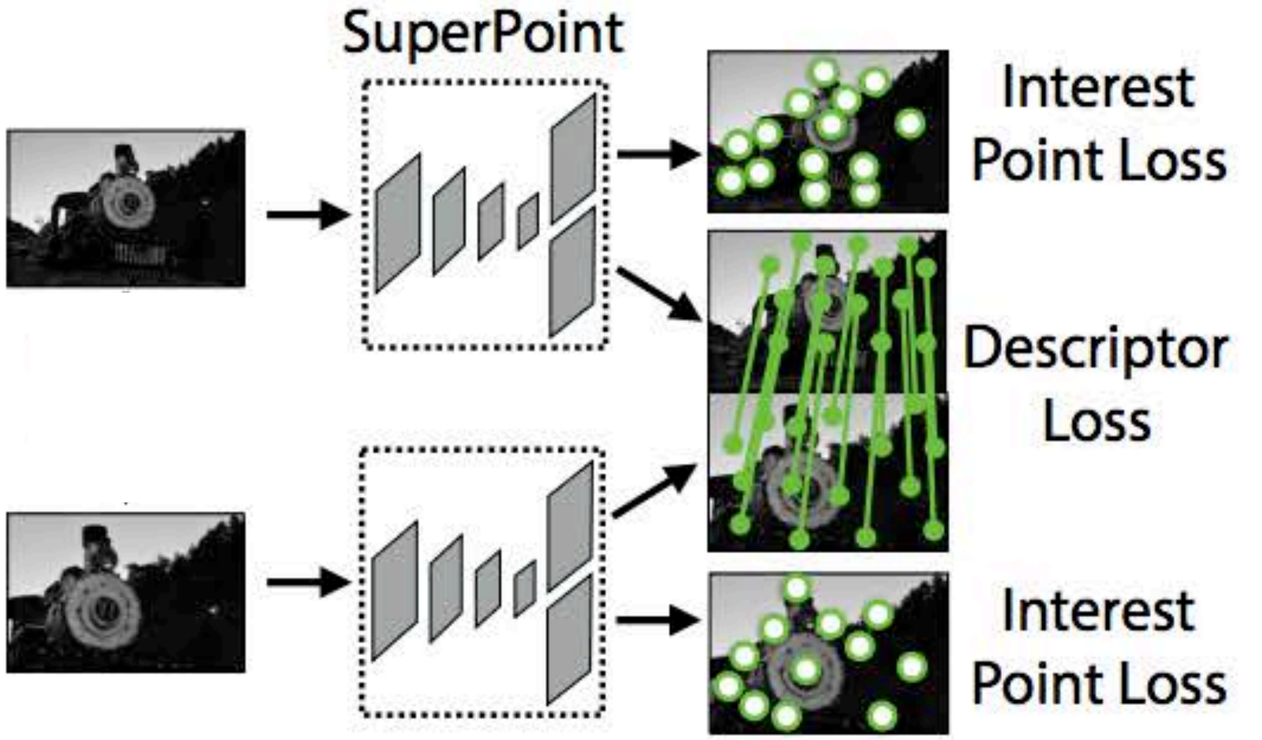
SuperPoint: A Learned Detector and Descriptor



How to train SuperPoint?

Slide adapted from: Derek Hoiem & T. Malisiewicz

SuperPoint: Training Overview

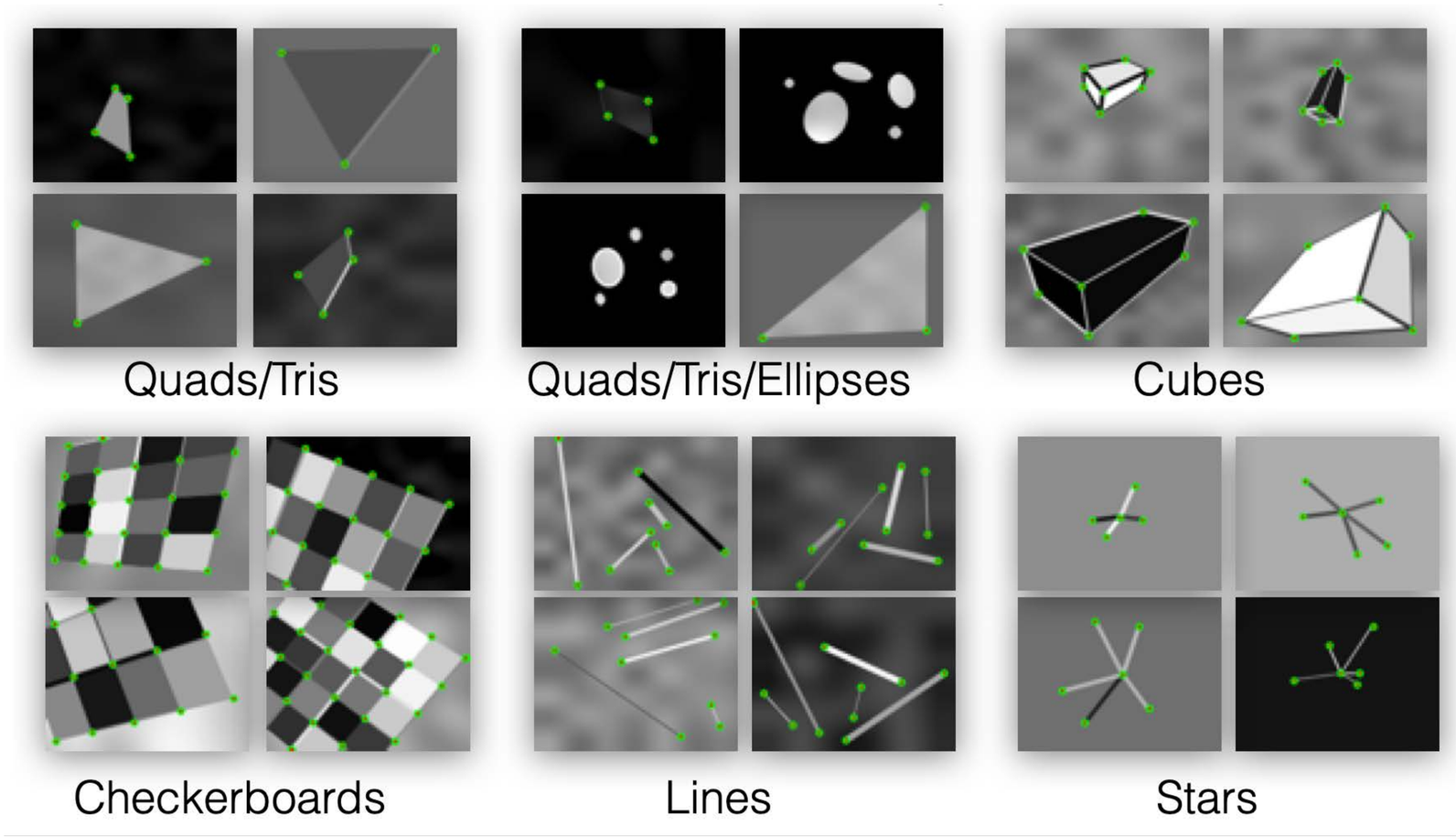


Keypoints trained using (supervised and self-supervised) labels

Descriptor training: metric learning objective

Slide adapted from: Derek Hoiem & T. Malisiewicz

Training Keypoint Detectors: Synthetic Training

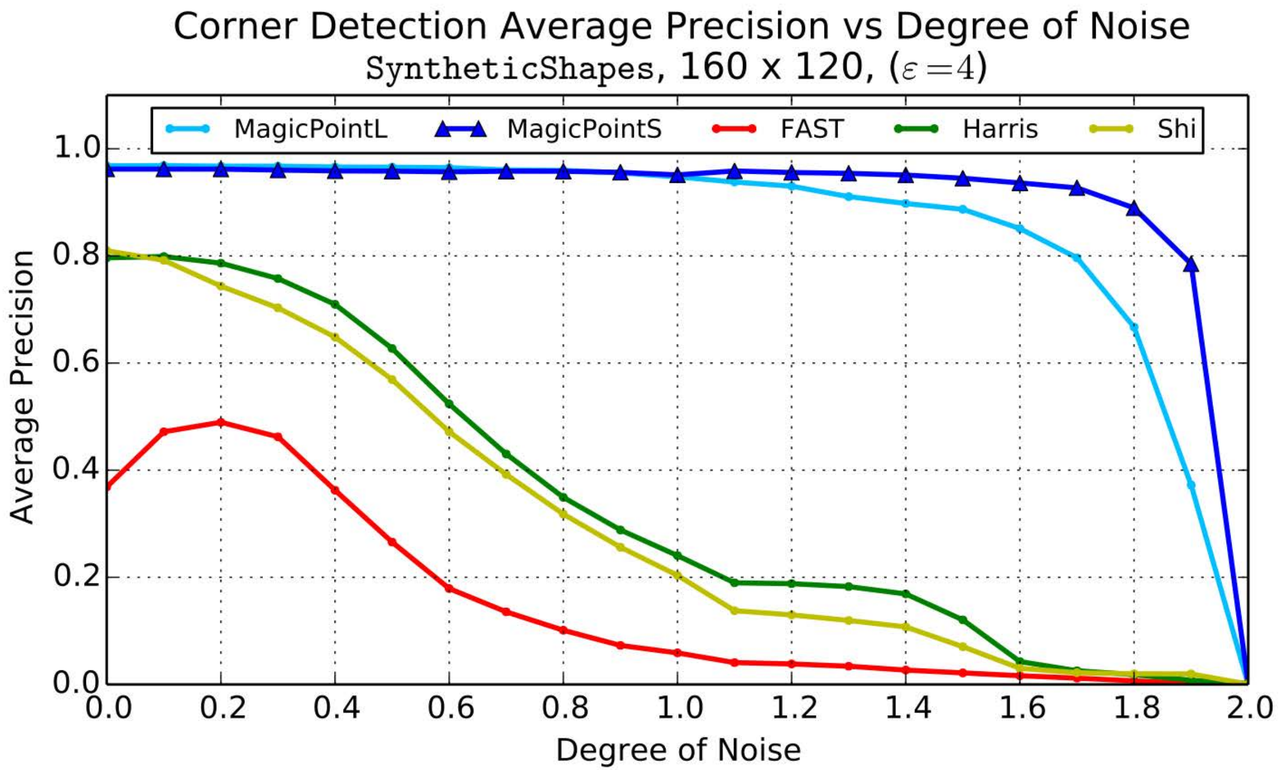


- Non-photorealistic shapes
- Heavy noise
- Effective and easy

Slide adapted from: Derek Hoiem & T. Malisiewicz

Early Version of SuperPoint: MagicPoint

"Toward Geometric Deep SLAM"
DeTone et. al. 2017



Worked well on synthetic data, but not real data

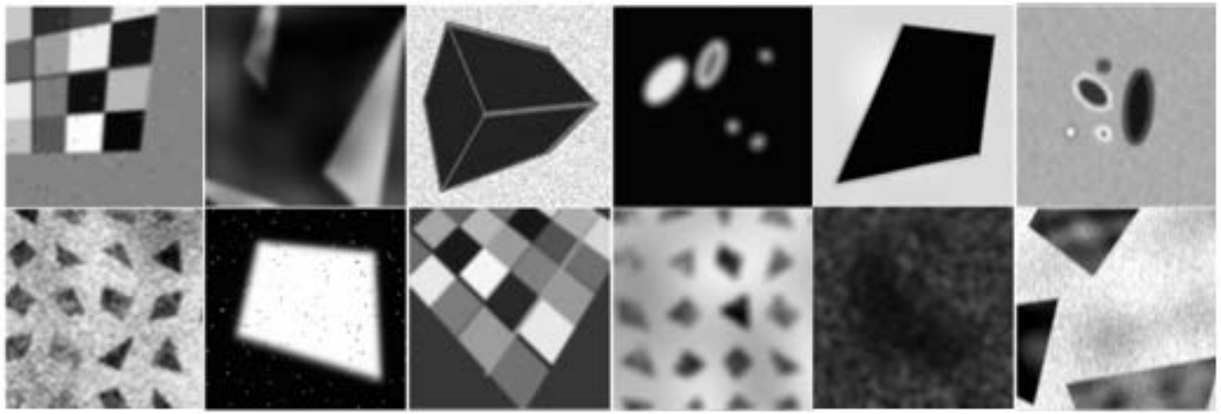
Q: How to finetune on natural images?

Idea: Self-supervised adaptation

Slide adapted from: Derek Hoiem & T. Malisiewicz

Training Keypoint Detectors

Synthetic Shapes (has interest point labels)



First train on this

MS-COCO (no interest point labels)

“Homographic Adaptation”

Use resulting detector to label this

Slide adapted from: Derek Hoiem & T. Malisiewicz

Homographic Adaptation

Unlabeled Input Image

Synthetic Warp + Run Detector

Point Set #1

Point Set #2

Point Set #3

Point Aggregation

Detected Point Superset

- Simulate planar camera motion with homographies
- Self-labelling technique
  - Suppress spurious detections
  - Enhance repeatable points

Slide adapted from: Derek Hoiem & T. Malisiewicz

Training Details

- Pretrain on synthetic images for 200000 iterations
- Generate 100 random homographies in 240x320 COCO images and pool keypoints
- Train to predict these pooled keypoints on original image

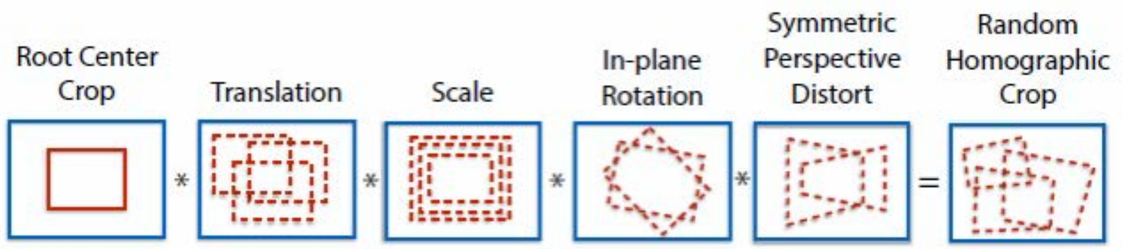
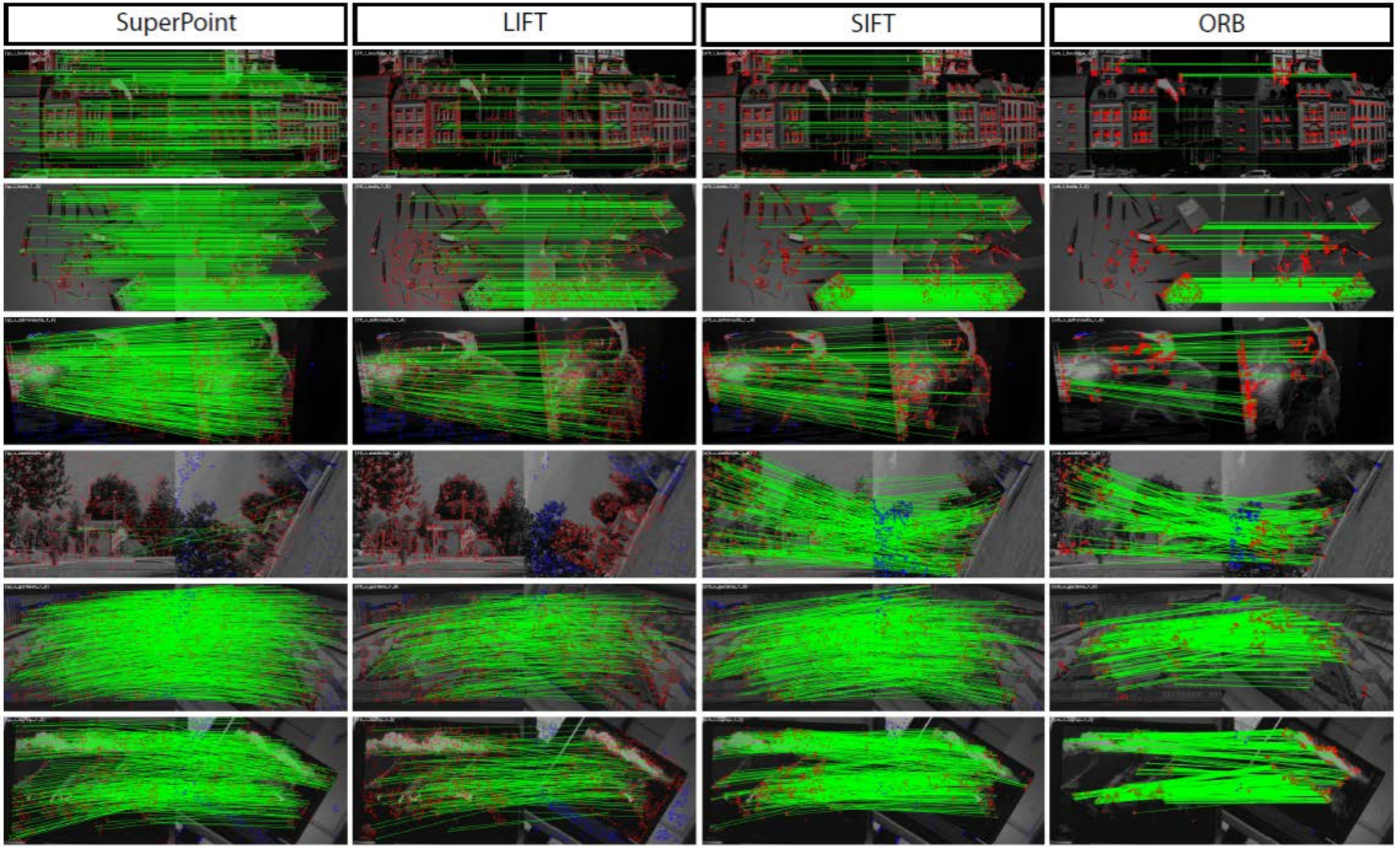


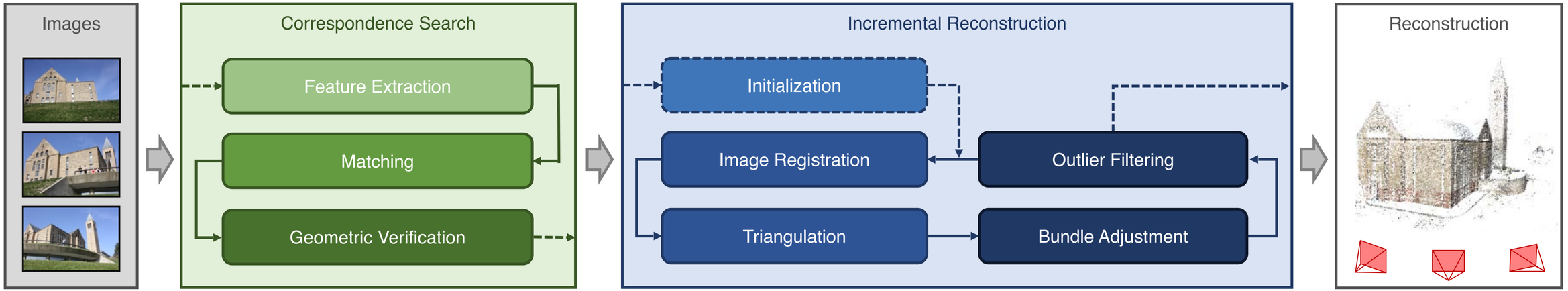
Figure 6. **Random Homography Generation.** We generate random homographies as the composition of less expressive, simple transformations.

Slide adapted from: Derek Hoiem & T. Malisiewicz



Slide adapted from: Derek Hoiem & T. Malisiewicz

Learning for SfM



Can we improve the robustness/precision via data-driven learning?

Example 1: Improving features and keypoints for matching

Example 2: Improving the matching process via global reasoning

Image credits: Johannes Schönberger

A minimal matching pipeline

SuperGlue: context aggregation + matching + filtering

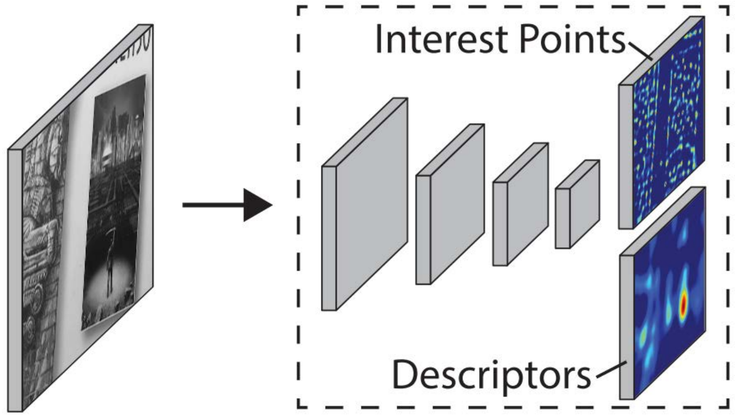
image pair

detection → description → feature matching → outlier filtering → pose estimation

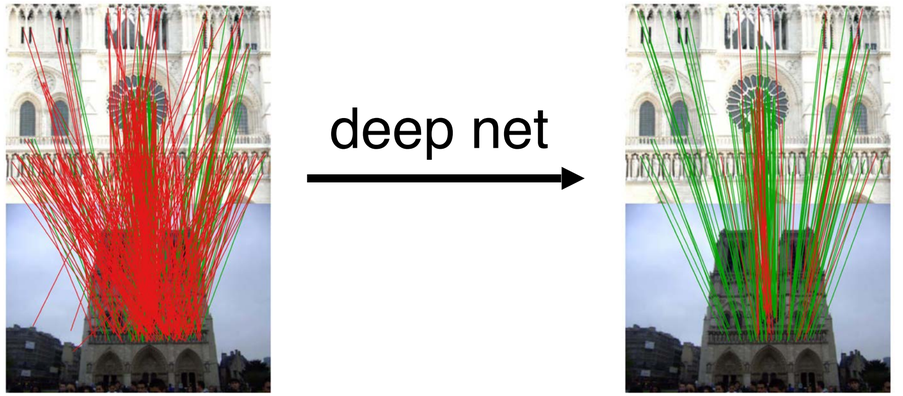
> Classical: SIFT, ORB
> Learned: SuperPoint, D2-Net

Nearest Neighbor Matching

> Heuristics: ratio test, mutual check
> Learned: classifier on set



[DeTone et al, 2018]

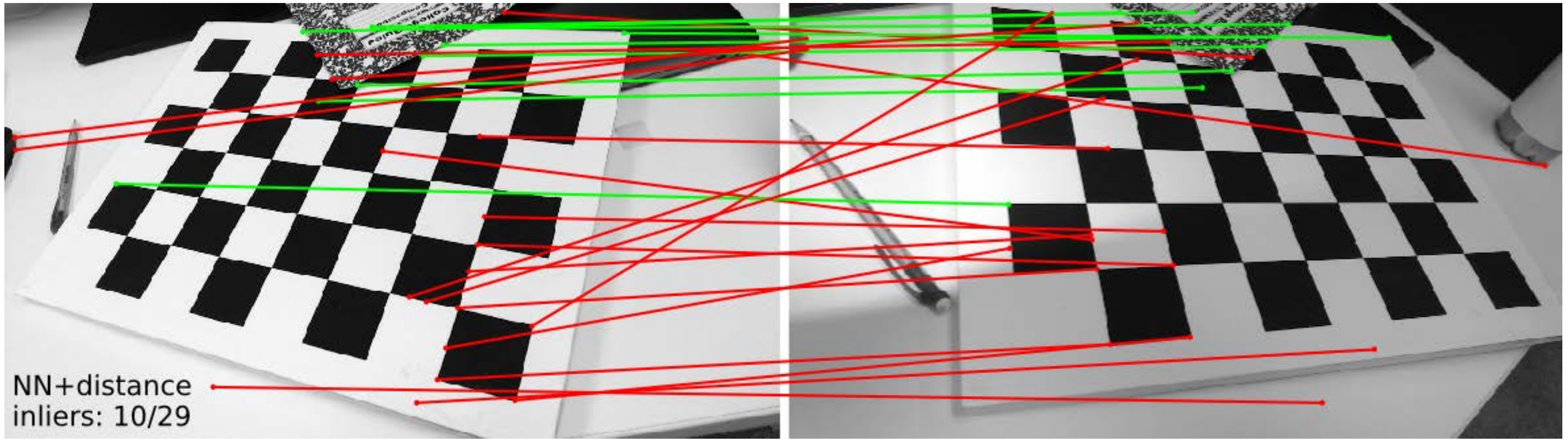


[Yi et al, 2018]

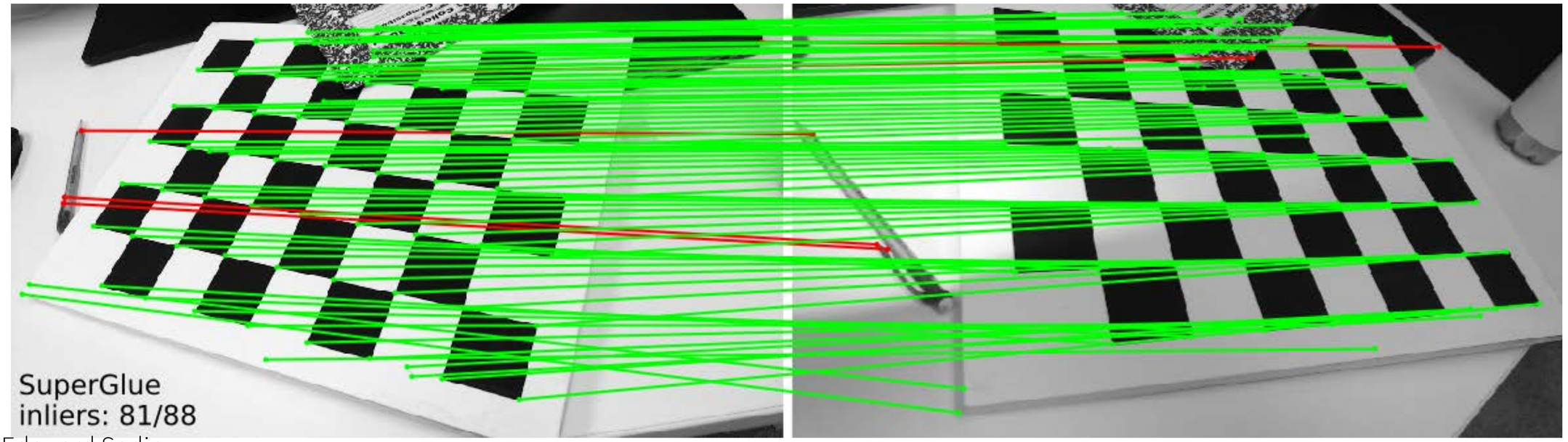
Slide credits: Paul-Edouard Sarlin

The importance of context

no SuperGlue



with SuperGlue



Slide credits: Paul-Edouard Sarlin

Problem formulation

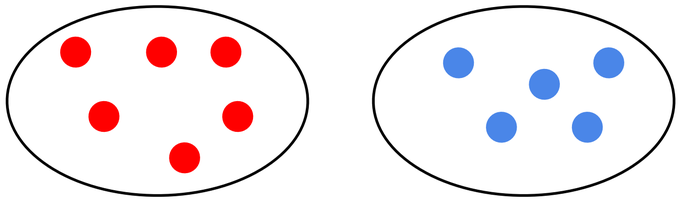
Inputs → 

→ **Outputs**

- Images **A** and **B**
- **2 sets** of **M**, **N** **local features**
  - Keypoints:
    - Coordinates
    - Confidence
  - Visual descriptors:

$$\mathbf{p}_i := (x, y, c)_i$$

$$\mathbf{d}_i$$

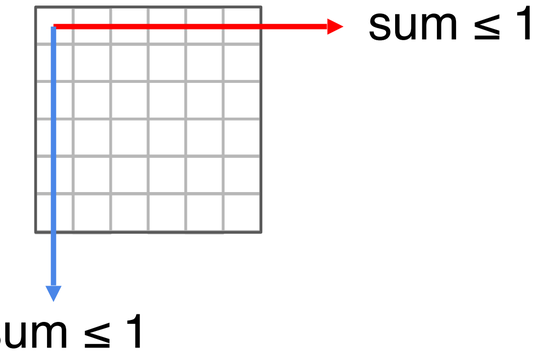


Single a match per keypoint

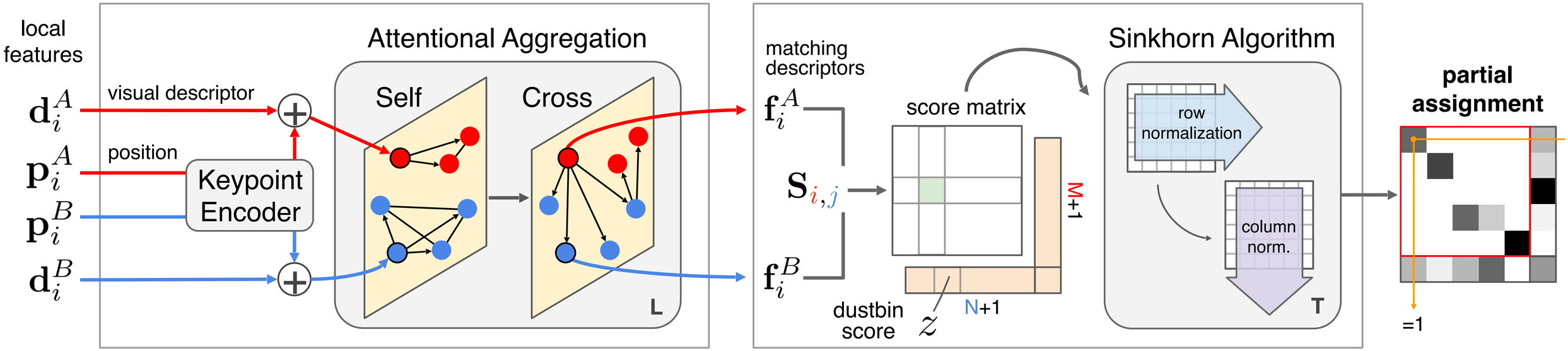
+ occlusion and noise

→ a **soft partial assignment**:

$$\mathbf{P} \in [0, 1]^{M \times N}$$



Slide credits: Paul-Edouard Sarlin



A Graph Neural Network with attention

Encodes **contextual cues** & priors

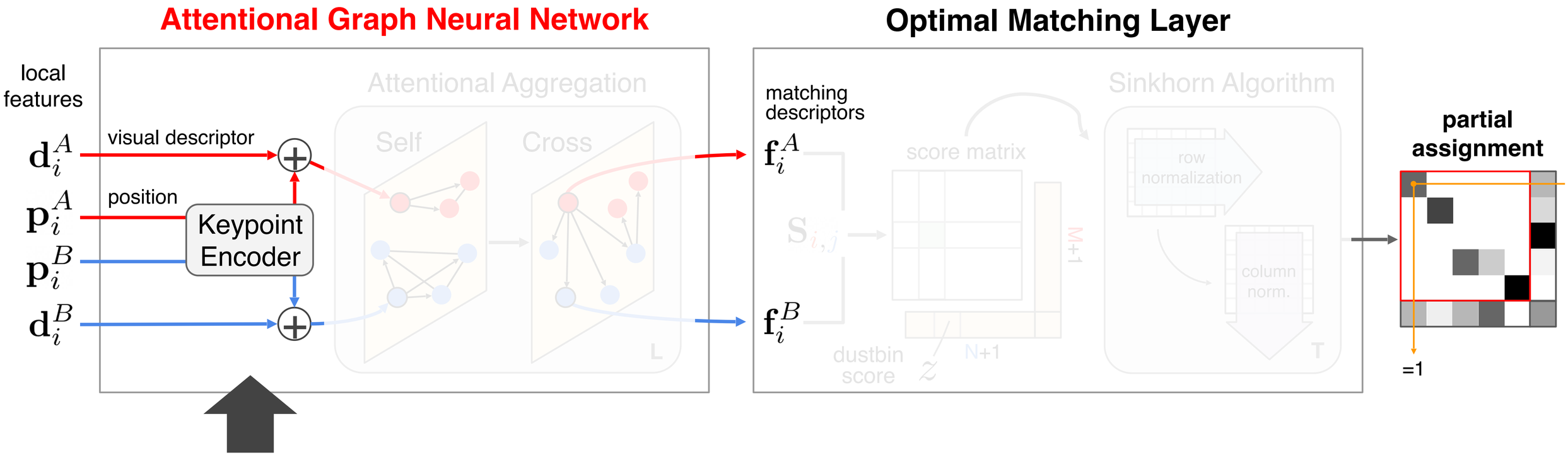
Reasons about the 3D scene

Solving a partial assignment problem

Differentiable **solver**

Enforces the assignment constraints
= **domain knowledge**

Slide credits: Paul-Edouard Sarlin

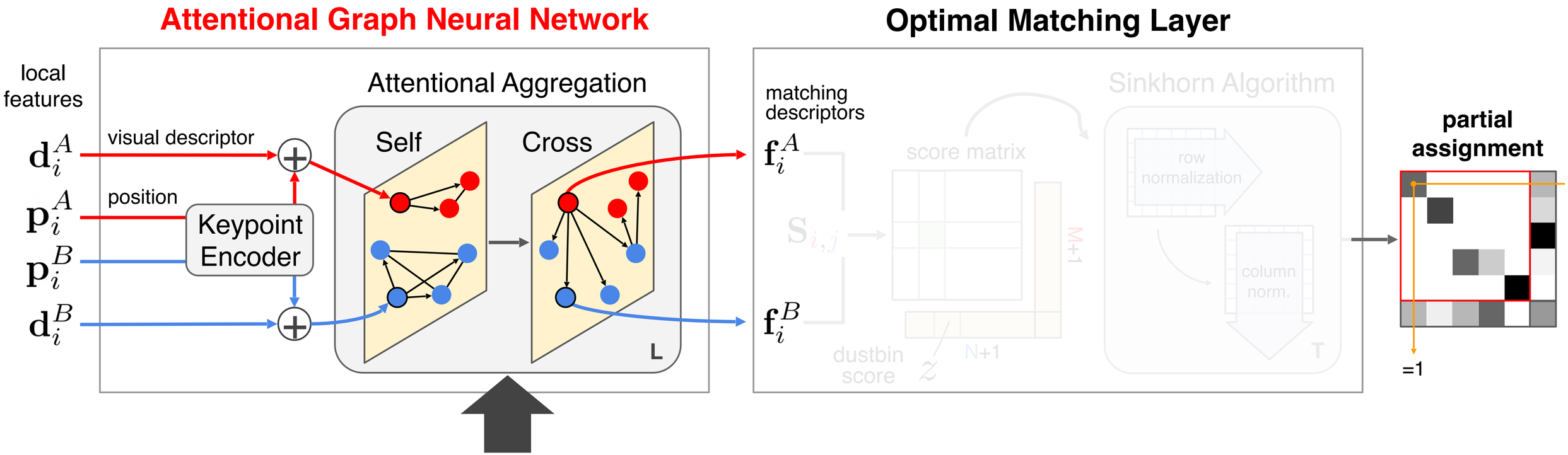


- Initial representation for each keypoints i: ${}^{(0)}\mathbf{x}_i$
- Combines visual appearance and position with an MLP:

$$ {}^{(0)}\mathbf{x}_i = \mathbf{d}_i + \mathrm{MLP}\left(\mathbf{p}_i\right) $$

Multi-Layer Perceptron

Slide credits: Paul-Edouard Sarlin



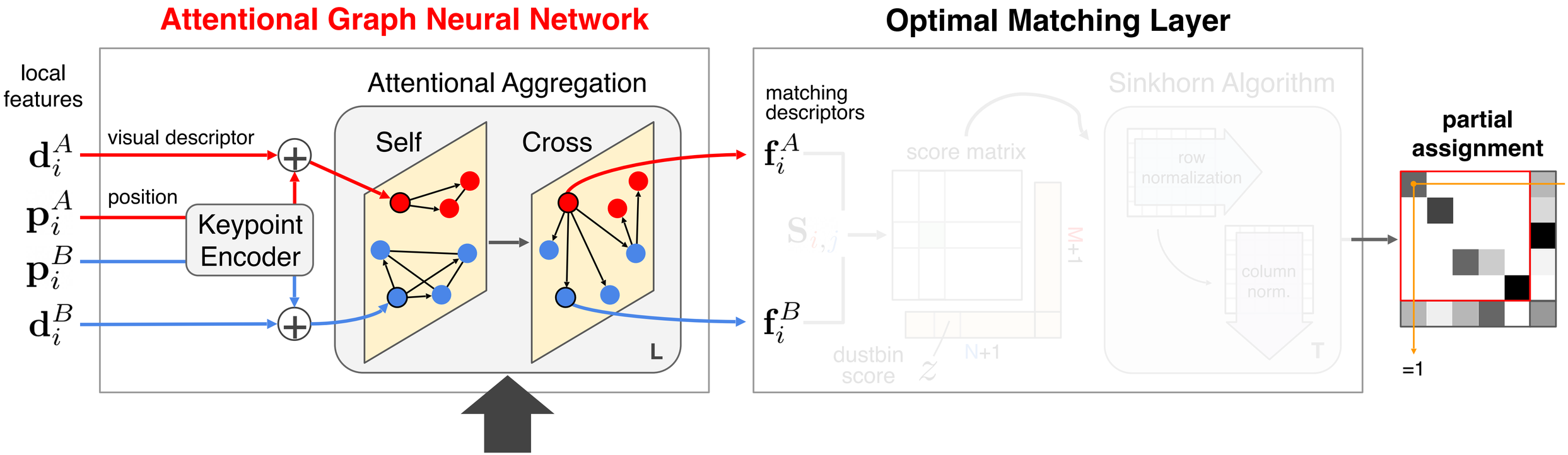
Update the representation based on other keypoints:

- in the same image: "**self**" edges
- in the other image: "**cross**" edges

→ A complete **graph** with two types of edges

$$ {}^{(\ell)}\mathbf{x}_i^A \longrightarrow {}^{(\ell+1)}\mathbf{x}_i^A $$

Slide credits: Paul-Edouard Sarlin



Update the representation using a **Message Passing Neural Network**

$$ {}^{(\ell+1)}\mathbf{x}_i^A = {}^{(\ell)}\mathbf{x}_i^A + \mathrm{MLP}\left(\left[{}^{(\ell)}\mathbf{x}_i^A \,||\, \mathbf{m}_{\mathcal{E}\to i}\right]\right) $$

Similar to a single-head transformer layer

the message

Slide credits: Paul-Edouard Sarlin

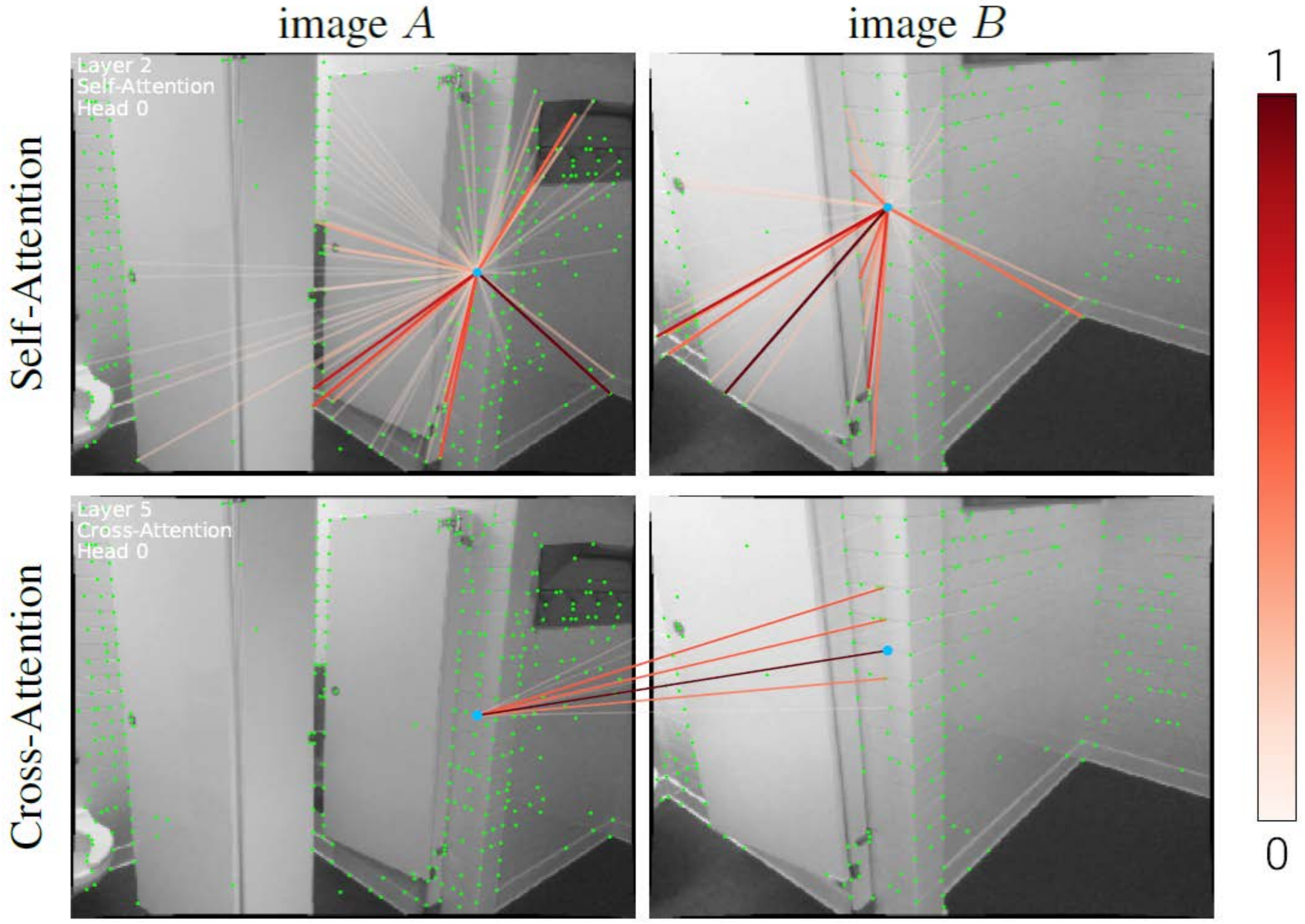
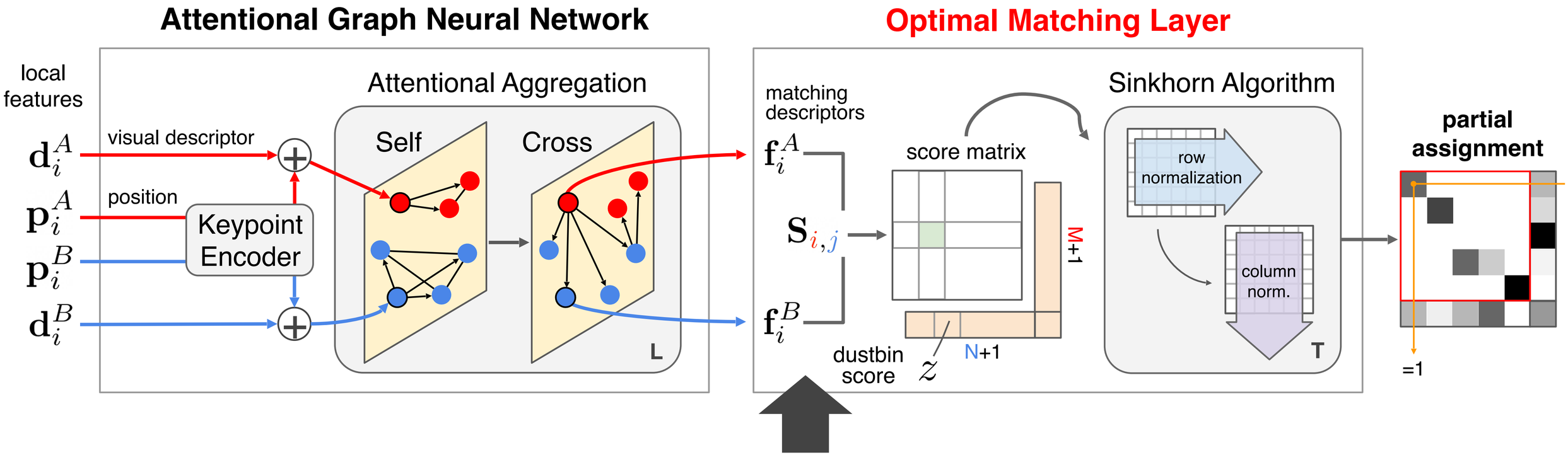


image A
image B
Self-Attention
Cross-Attention
Layer 2
Self-Attention
Head 0
Layer 5
Cross-Attention
Head 0
1
0

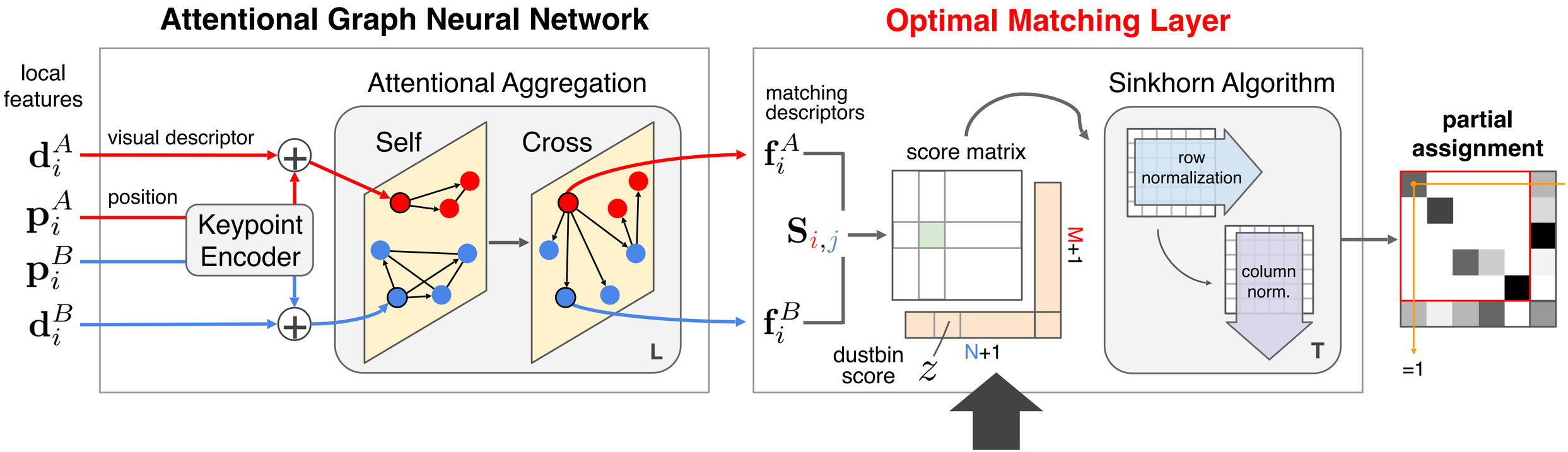


Compute a **score matrix** $\mathbf{S} \in \mathbb{R}^{M \times N}$ for all matches:

$$\mathbf{f}_i^A = \mathbf{W} \cdot {}^{(L)}\mathbf{x}_i^A + \mathbf{b}$$

$$\mathbf{S}_{i,j} = < \mathbf{f}_i^A, \mathbf{f}_j^B >$$

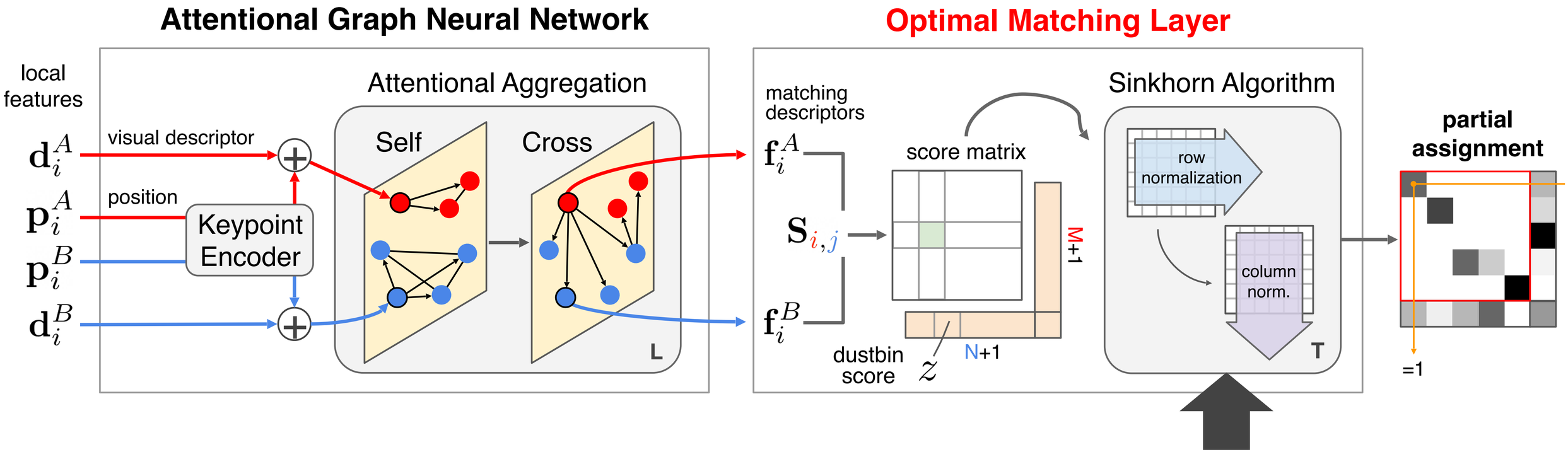
Slide credits: Paul-Edouard Sarlin



- Occlusion and noise: unmatched keypoints are assigned to a **dustbin**
- **Augment** the scores with a learnable dustbin score z

$$\bar{\mathbf{S}}_{i,N+1} = \bar{\mathbf{S}}_{M+1,j} = \bar{\mathbf{S}}_{M+1,N+1} = z \in \mathbb{R}$$

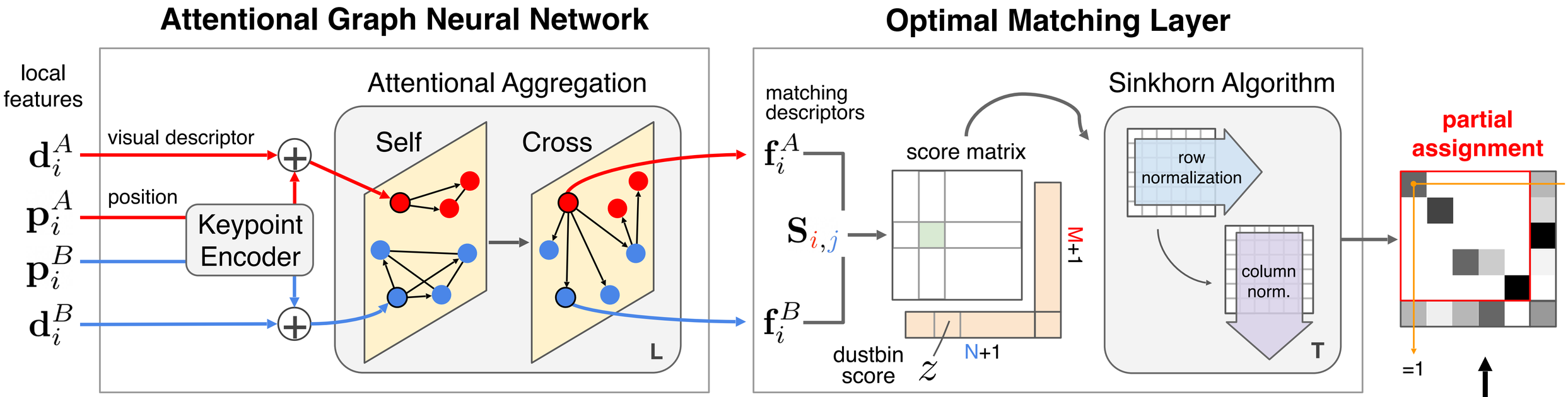
Slide credits: Paul-Edouard Sarlin



- Compute the assignment $\bar{\mathbf{P}}$ that maximizes $\sum_{i,j} \bar{\mathbf{S}}_{i,j} \bar{\mathbf{P}}_{i,j}$
- Solve an **optimal transport** problem
- With the **Sinkhorn algorithm**: differentiable & soft Hungarian algorithm

[Sinkhorn & Knopp, 1967]

Slide credits: Paul-Edouard Sarlin

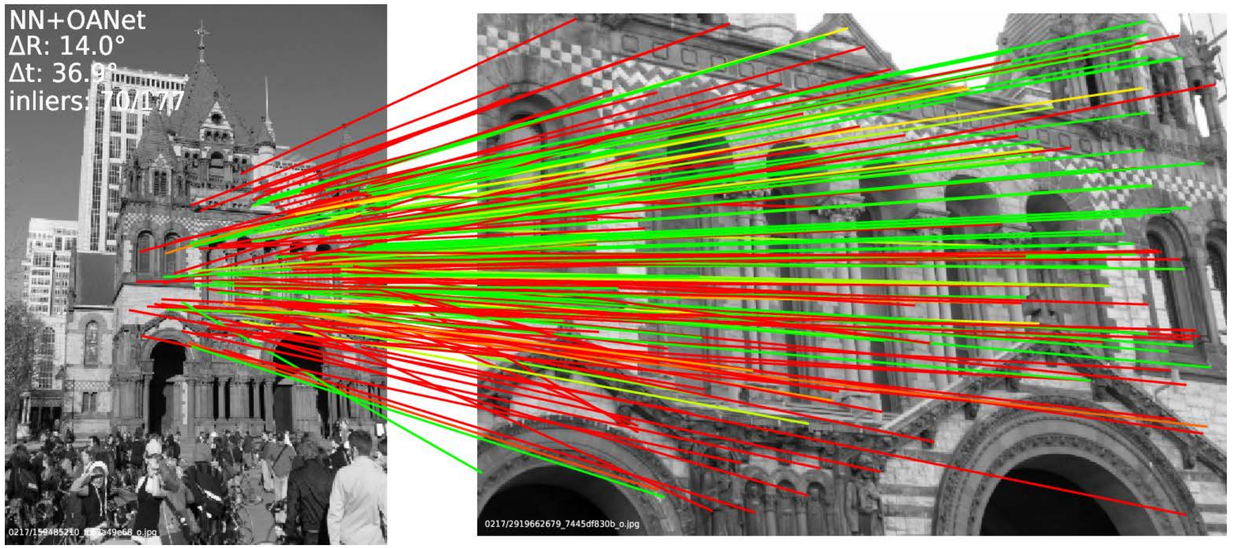


- Compute **ground truth correspondences** from pose and depth
- Find which keypoints should be **unmatched**
- Loss: maximize the log-likelihood $\bar{\mathbf{P}}_{i,j}$ of the GT cells

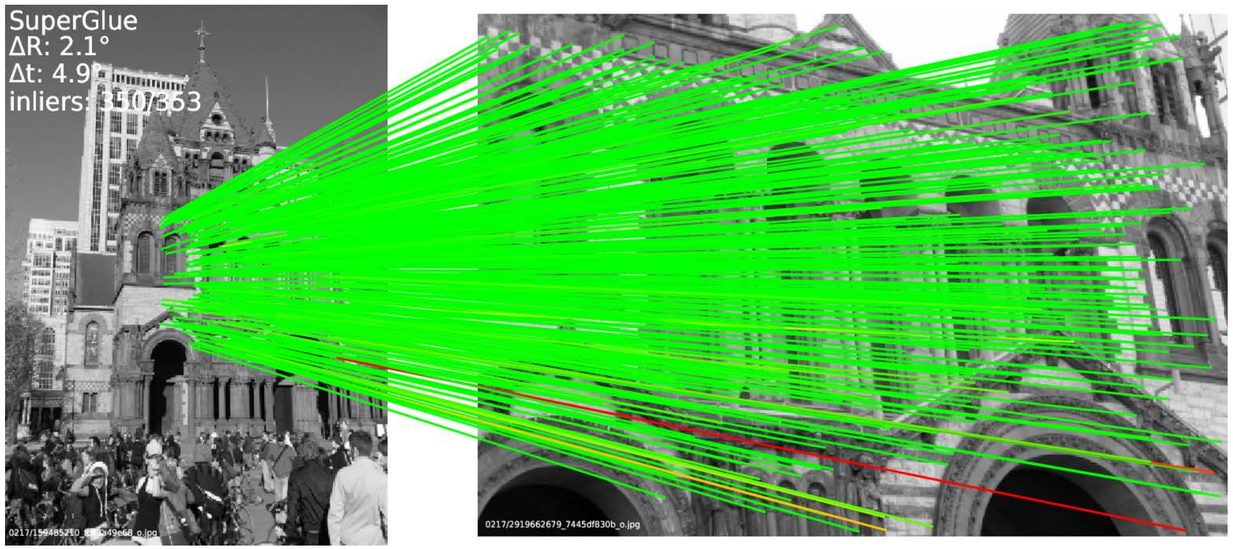
Slide credits: Paul-Edouard Sarlin

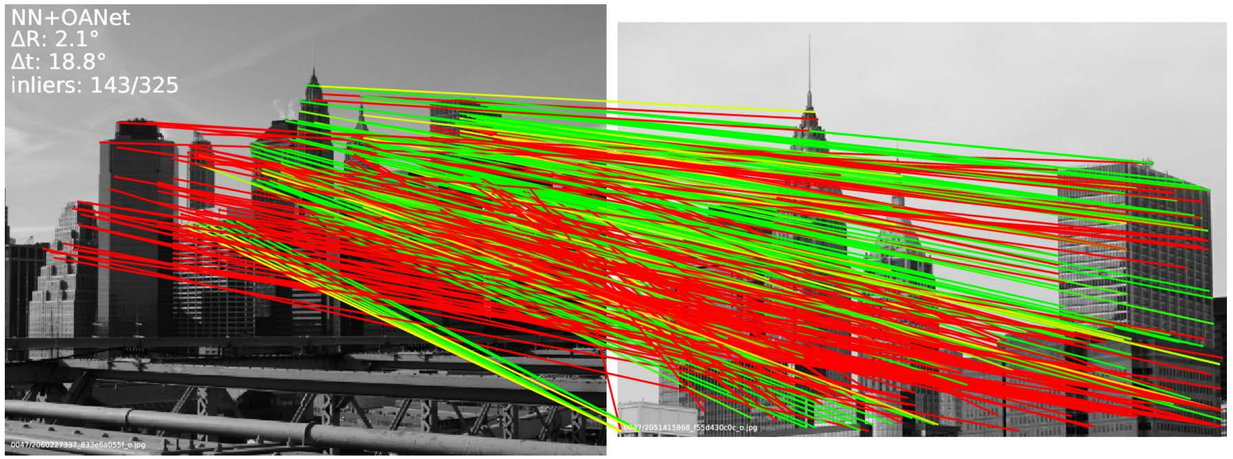
Results: outdoor - SfM

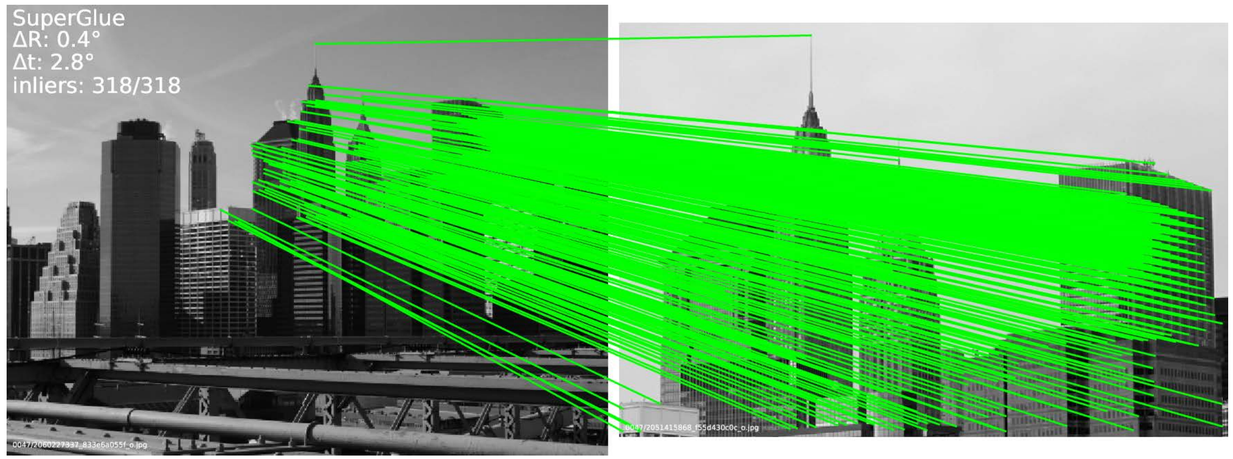
SuperPoint + NN + OA-Net (inlier classifier)



SuperPoint + **SuperGlue**



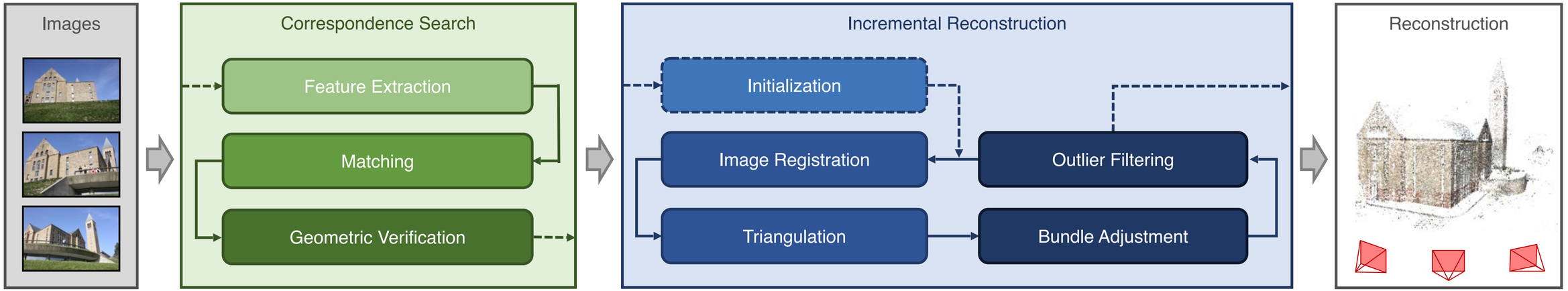




SuperGlue: more **correct matches** and fewer **mismatches**

Slide credits: Paul-Edouard Sarlin

Learning for SfM



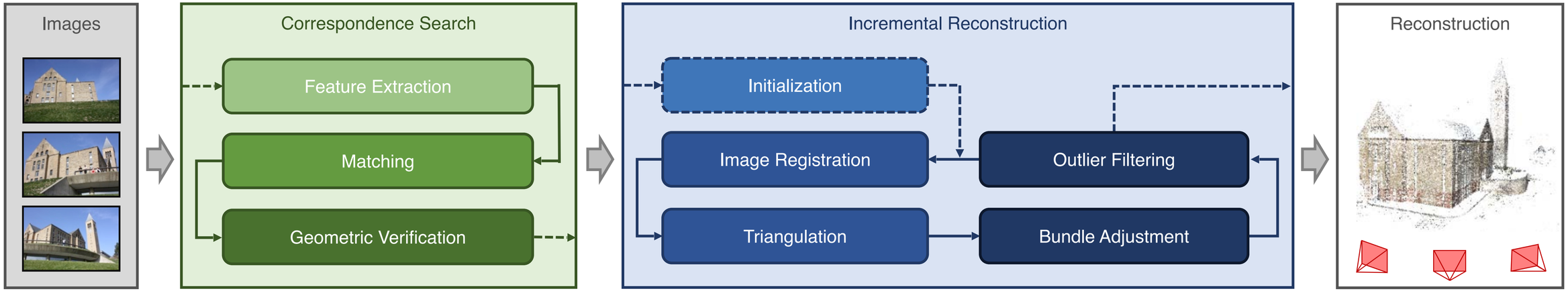
Can we improve the robustness/precision via data-driven learning?

Example 1: Improving features and keypoints for matching

Example 2: Improving the matching process via global reasoning

Image credits: Johannes Schönberger

Learning for SfM



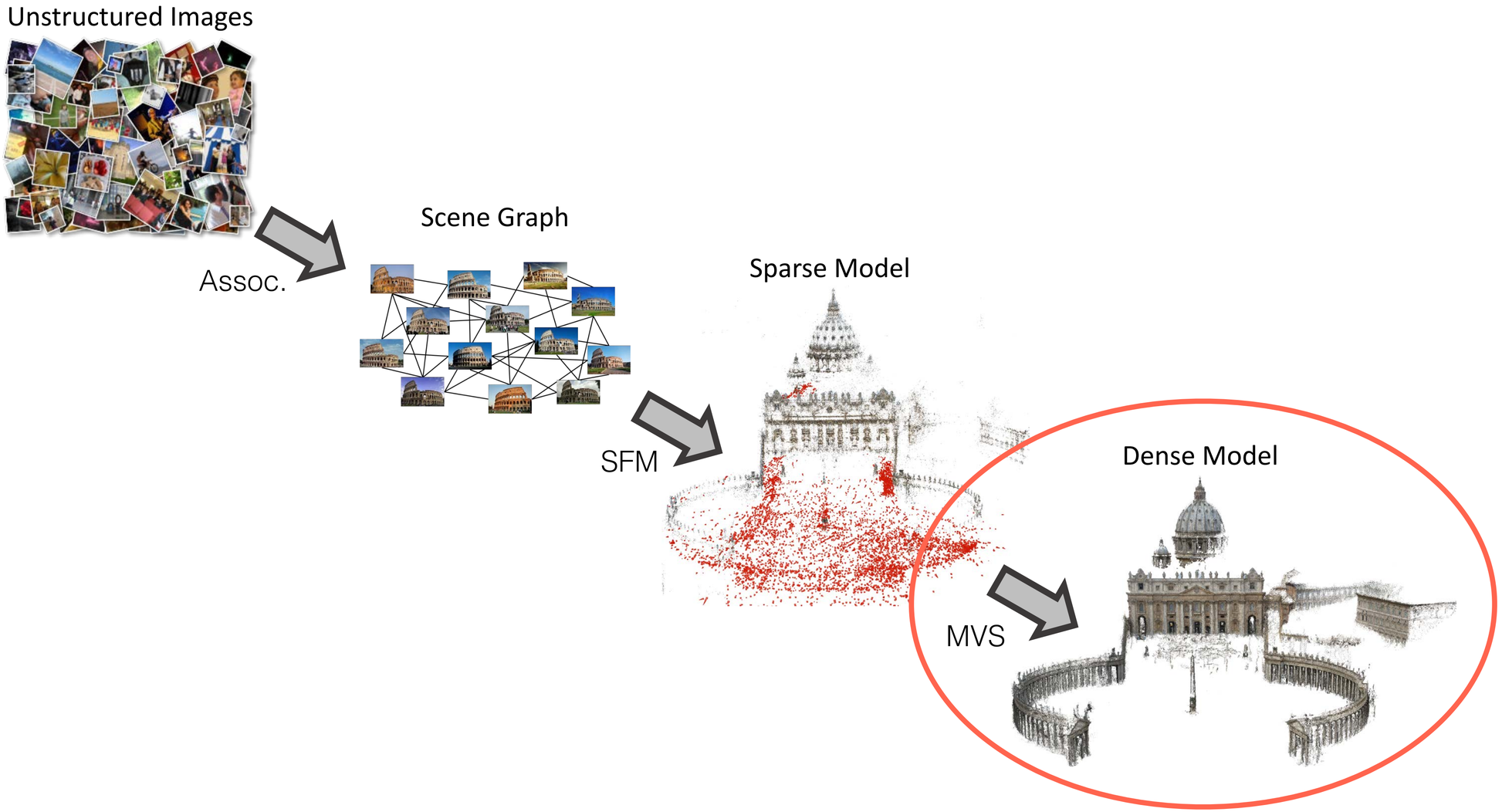
Why only learn modules? Why not rethink the pipeline?

Well-engineered systems that work well at scale — high entry barrier for new approach

But, lots of scope for alternate approaches in few-view settings!

Long-term — maybe other solutions will emerge?

Next Time


Unstructured Images
Assoc.
Scene Graph
SFM
Sparse Model
MVS
Dense Model