NYU TANDON SCHOOL OF ENGINEERING

Robot Perception

SfM

Dr. Chen Feng

cfeng@nyu.edu

ROB-GY 6203, Fall 2022

Overview

++ Marker-based SfM

* Bundle adjustment

+ Error/uncertainty propagation

++ Feature-based SfM

* COLMAP

*: know how to code (or how to use tools)
++: know how to derive (more than just the concept)
+: know the concept

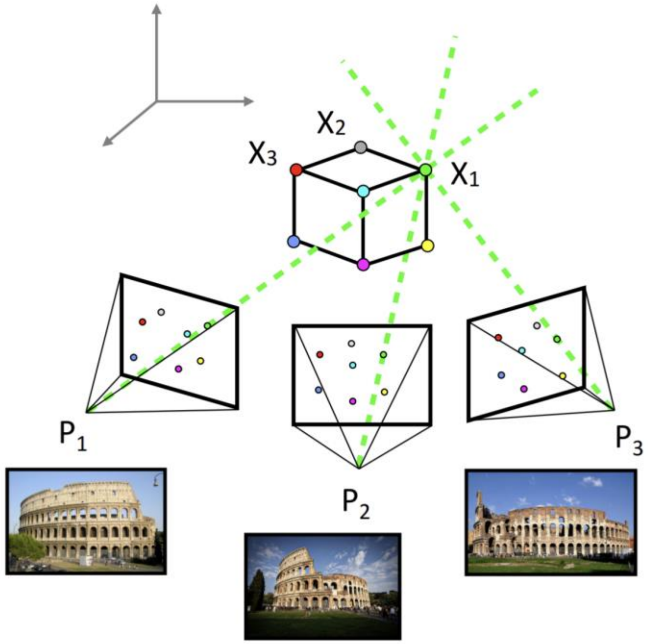
References

- Sz2022:
  - Chapter 11
- HZ2003:
  - Section 5.2, 18.1, A6.6

- https://colmap.github.io/
  - https://demuc.de/tutorials/cvpr2017/

SfM: Structure from Motion

- Joint estimation of …
  - Structure X_i
  - Cameras P_j
- … from motion, i.e.
  - images at different viewpoints

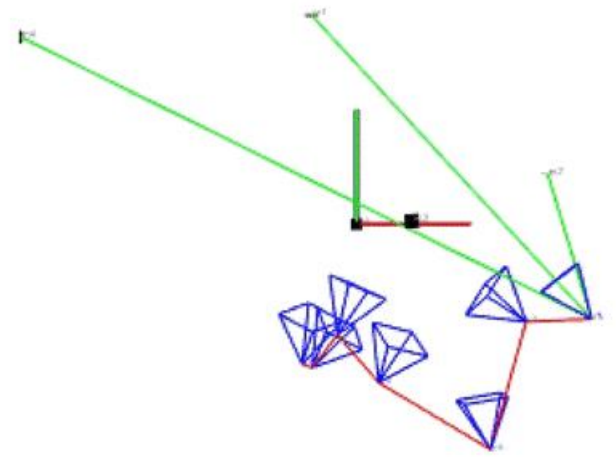


Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf, https://colmap.github.io/tutorial.html

Let’s Start from Something Simple and Useful

- Marker-based SfM
  - Given *n* images observing *m* markers by a calibrated camera
  - Find out the *m* marker poses (the structure), and the *n* image poses (the motion)

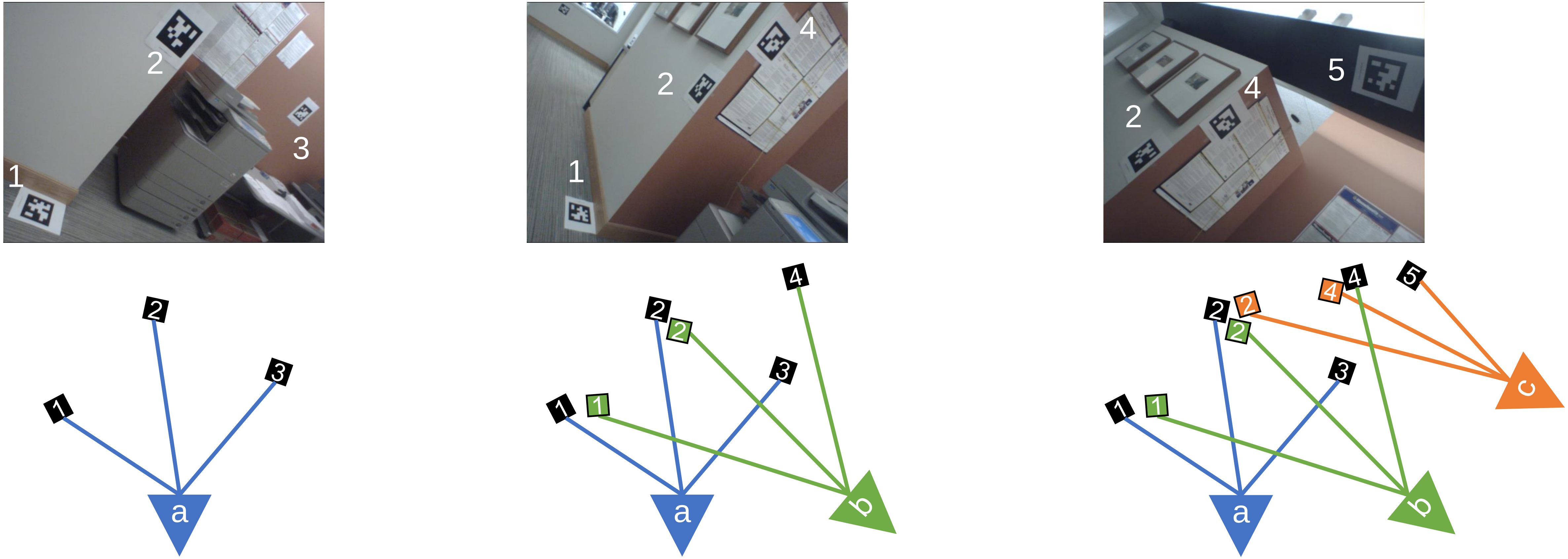


Feng, Chen. "Camera Marker Networks for Pose Estimation and Scene Understanding in Construction Automation and Robotics." (2015).

Marker-based SfM

- Challenge: how to find the best estimation of all the marker & image poses?

Image a   Image b   Image c

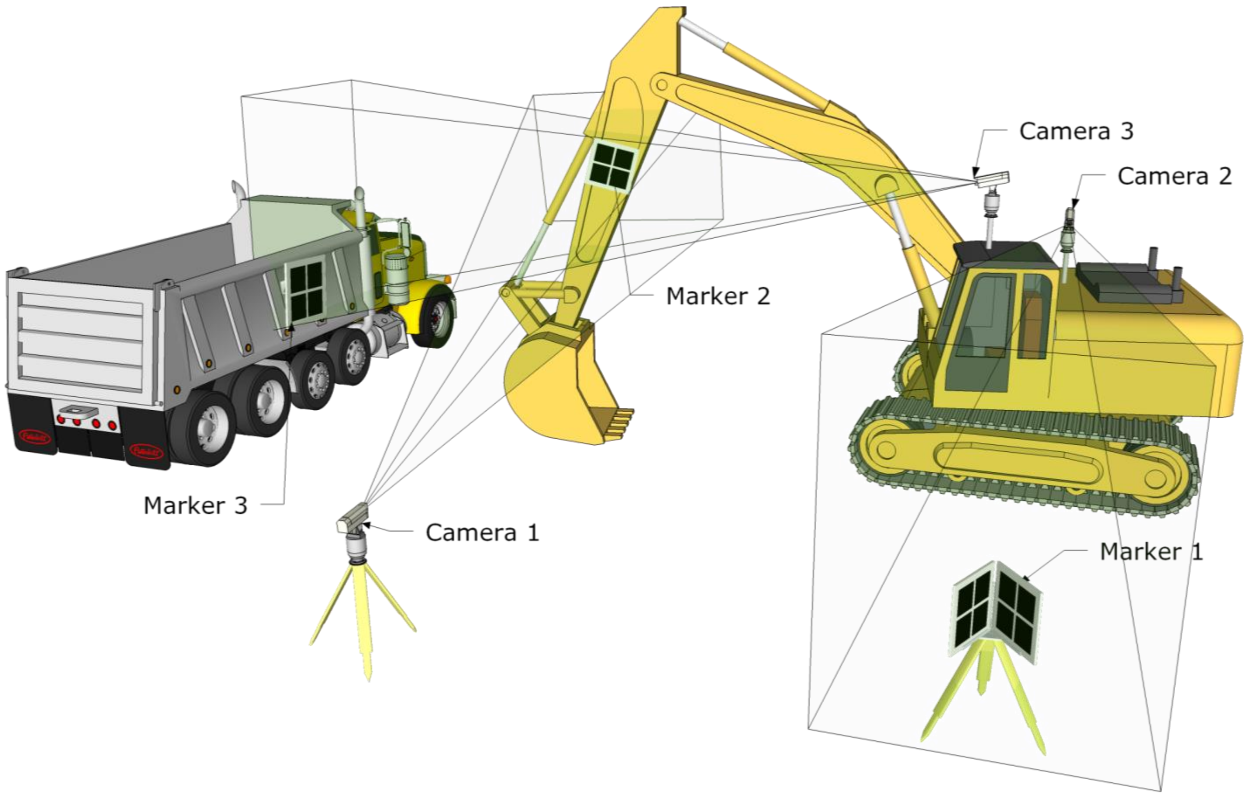


Feng, Chen. "Camera Marker Networks for Pose Estimation and Scene Understanding in Construction Automation and Robotics." (2015).

Camera Marker Networks

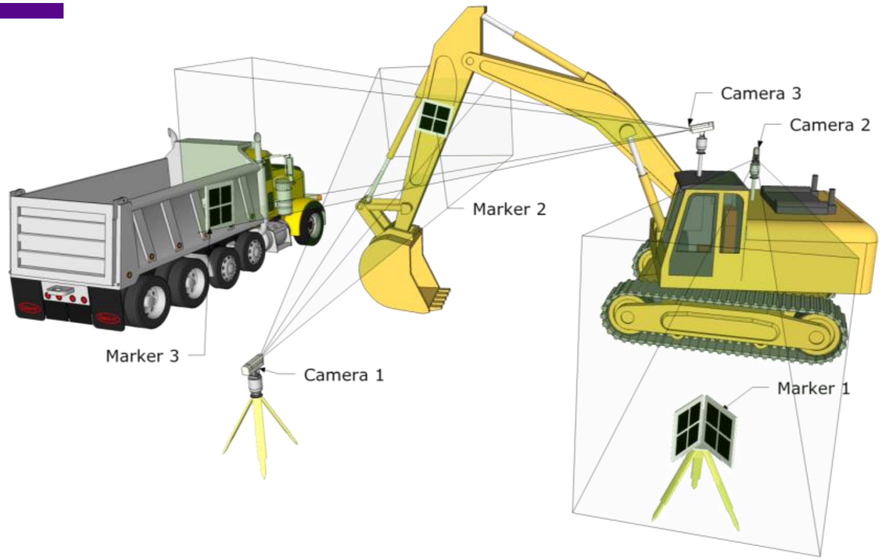
- Definition
  - Observation system
  - Multiple cameras or markers
  - Pose estimation of embedded object

- Multiple Cameras and Views



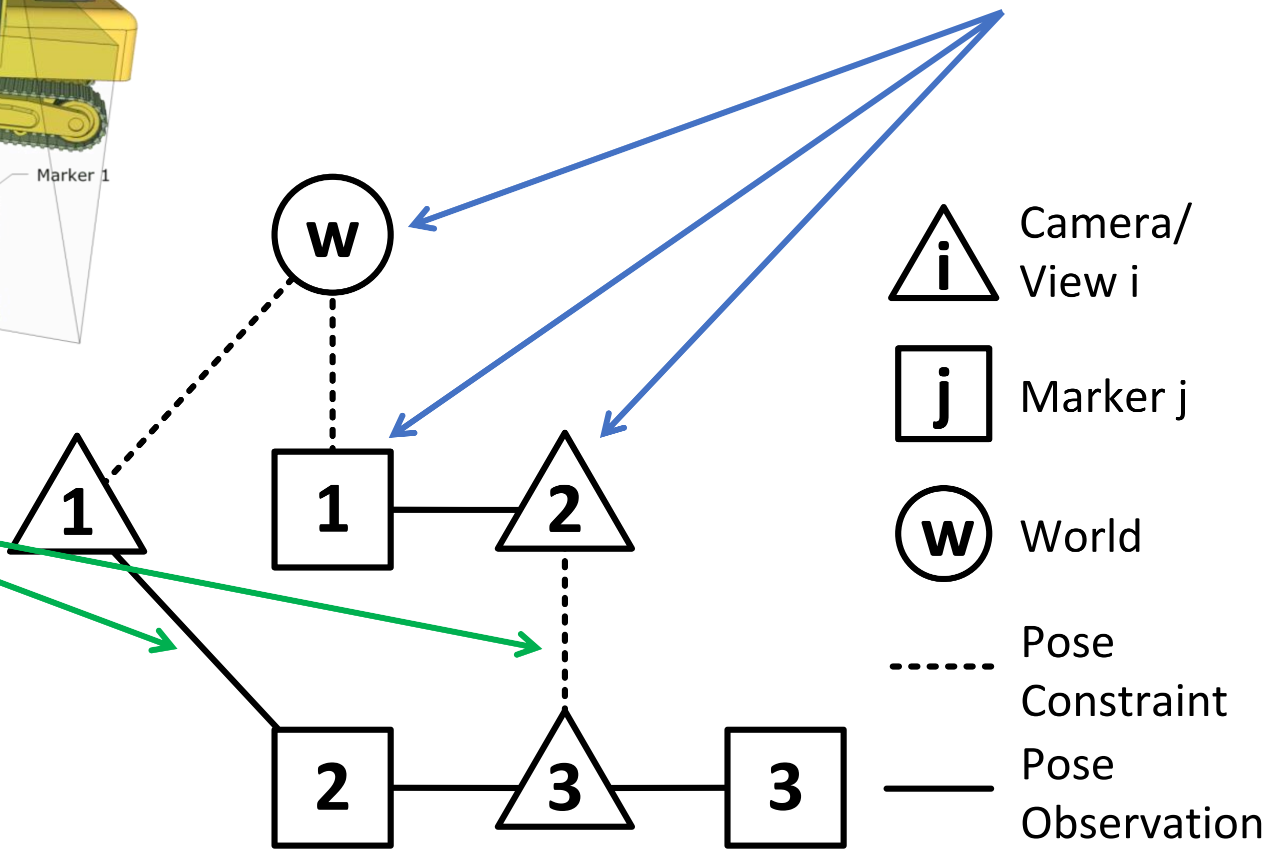
Feng, Chen, Vineet R. Kamat, and Carol C. Menassa. "Marker-Assisted Structure from Motion for 3D Environment Modeling and Object Pose Estimation." (2016).

Graph Abstraction: A Unified Framework

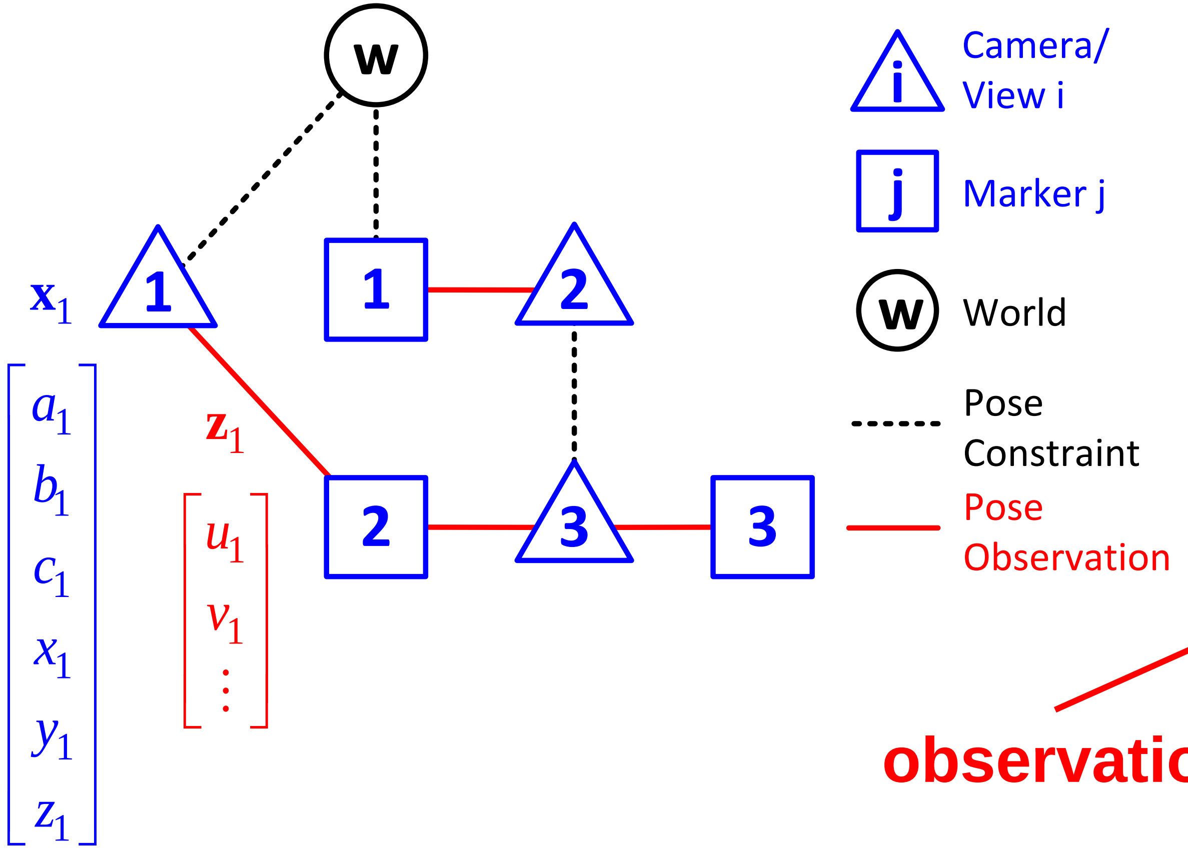


Node: pose of an object

Edge: relative relationship



Mathematical Solution: Observation Model



Perspective projection **Markers' points** **Cameras' intrinsics**

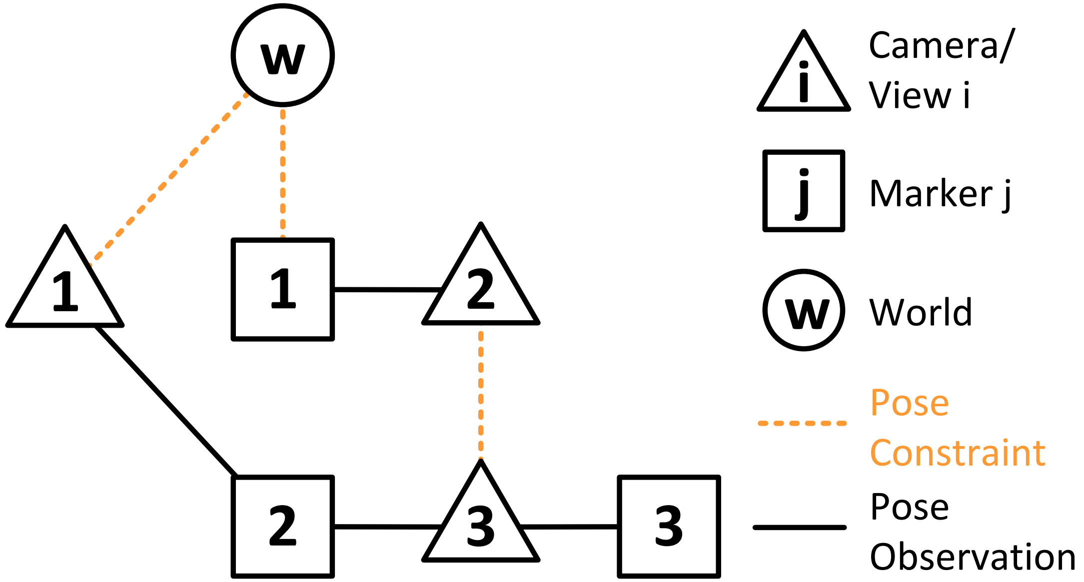
$$\mathbf{Z} = \mathbf{F}(\mathbf{X} ; \mathbf{Y} , \mathbf{K})$$

$$\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \end{bmatrix} \quad \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \end{bmatrix} \begin{bmatrix} m_1^x \\ m_1^y \\ m_1^z \\ \vdots \end{bmatrix} \begin{bmatrix} c_1^x \\ c_1^y \\ f_1^x \\ f_1^y \\ \vdots \end{bmatrix}$$

observations

poses

Mathematical Solution: Constraint Model



$$\mathbf{G}(\mathbf{X}) = \begin{bmatrix} g_1(\mathbf{x}_{s_1}, \mathbf{x}_{e_1}) \\ \vdots \end{bmatrix} = \mathbf{0}$$

Constraint types

- Fixed-node
- Parallelism
- Perpendicularity
- Coplanarity

Mathematical Solution: Optimization

- Final solution from bundle adjustment (Triggs et al. 2000):

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \left\| \hat{\mathbf{Z}} - \mathbf{F}(\mathbf{X}; \mathbf{Y}, \mathbf{K}) \right\|^2_{\mathbf{C}_{\hat{\mathbf{Z}}}} + \left\| \mathbf{G}(\mathbf{X}) \right\|^2_{\mathbf{C}_{\mathbf{K}}}$$

Actual observations

A priori covariance matrix of actual observations

Weighted constraint residuals

- Solve by Levenberg-Marquardt algorithm
  - For large problem:
    - Ceres (Agarwal and Mierle 2012)
    - g2o (Kummerle et al. 2011)
  - For small problem:
    - minimize or least_squares (scipy.optimize)
    - fminunc or lsqnonlin (matlab)

Uncertainty Propagation

Forward Computation
Observation model $\mathbf{f}$, from given state $\mathbf{x}$

For linear functions

$$f(\boldsymbol{x}) = \mathbf{J}\boldsymbol{x}$$

$$\boldsymbol{C}_f = \mathbf{J}\boldsymbol{C}_{\boldsymbol{x}}\mathbf{J}^{\mathrm{T}}$$

For non-linear functions, we can approximate via Taylor's Expansion

$$f(x) \approx f(x_0) + \mathbf{J}(x_0)(x - x_0)$$

Note: Discard higher order terms

$$\boldsymbol{C}_f = \mathbf{J}(x_0)\boldsymbol{C}_{\boldsymbol{x}}\mathbf{J}(\mathrm{x}_0)^{\mathrm{T}}$$

Uncertainty Propagation

Backward Computation
Estimate state $\mathbf{x}$, from given observation $\mathbf{f}$

$$f(\boldsymbol{x}) = \mathbf{J}\boldsymbol{x}$$

$$\mathbf{P}_f = \mathbf{C}_f^{-1}$$

Note: **P** is also known as the **information matrix, precision matrix**

We can solve for **x** via Weighted Least Square Estimation, minimizing the **Mahalanobis Distance**

$$\boldsymbol{x} = (\mathbf{J}^{\mathrm{T}}\mathbf{P}_f\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{P}_f f$$

$$\boldsymbol{C}_x = \left[(\mathbf{J}^{\mathrm{T}}\mathbf{P}_f\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{P}_f\right]\boldsymbol{C}_f\left[(\mathbf{J}^{\mathrm{T}}\mathbf{P}_f\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{P}_f\right]^{\mathrm{T}}$$

Uncertainty Propagation

Resource for revision on Linear Algebra
Appendix A, Convex Optimization by Boyd and Vandenberghe

Backward Computation
Estimate state $\mathbf{x}$, from given observation $\mathbf{f}$

$$\boldsymbol{C}_{x} = \left[(\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\right]\boldsymbol{C}_{f}\left[(\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\right]^{\mathrm{T}}$$

$$= (\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{C}_{f}\mathbf{P}_{f}^{\mathrm{T}}\mathbf{J}(\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{J})^{-\mathrm{T}}$$

$$= (\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{J}(\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{J})^{-\mathrm{T}}$$

$$= (\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{J})^{-\mathrm{T}}$$

$$= (\mathbf{J}^{\mathrm{T}}\mathbf{P}_{f}\mathbf{J})^{-1}$$

Uncertainty Analysis is Critical

- Reason 1: A **measure of confidence** level
  - Backward propagation of $\mathbf{C}_{\hat{\mathbf{Z}}}$, the actual observation's covariance matrix

$$\mathbf{C}_{\hat{\mathbf{X}}} = (\mathbf{J}^T \mathbf{C}_{\hat{\mathbf{Z}}}^{-1} \mathbf{J})^{-1}$$

$$\mathbf{J} = \left.\frac{\partial \mathbf{F}}{\partial \mathbf{X}}\right|_{\hat{\mathbf{X}}}$$

Jacobian matrix of **F**
evaluated at the final solution

Estimation Uncertainty can be Visualized as Ellipsoid

https://github.com/simbaforrest/masfm

Uncertainty Analysis is Critical

- Reason 2: A **tool to evaluate stability** vs. state
  - Backward propagation at any state **X**

$$\mathbf{C}_{\mathbf{X}}(\mathbf{X}) = \left(\mathbf{J}(\mathbf{X})^{\mathrm{T}} \mathbf{C}_{\hat{\mathbf{Z}}}^{-1} \mathbf{J}(\mathbf{X})\right)^{-1}$$

    - The **theoretically best/smallest** pose estimation **uncertainty** that one can expect at **X**
    - The system stability at **X**

Optimize Configuration to Reduce Uncertainty

- Improve camera marker network designs by:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \operatorname{Cost}(\mathbf{C}_{\mathbf{X}}(\mathbf{X}))$$

  - Trust region reflective (Coleman and Li 1996), etc.
  - MATLAB's *fmincon*
- **Intractable** for large network if using Monte Carlo simulation (Luhmann 2009)

Feature-based SfM Pipeline – Overview

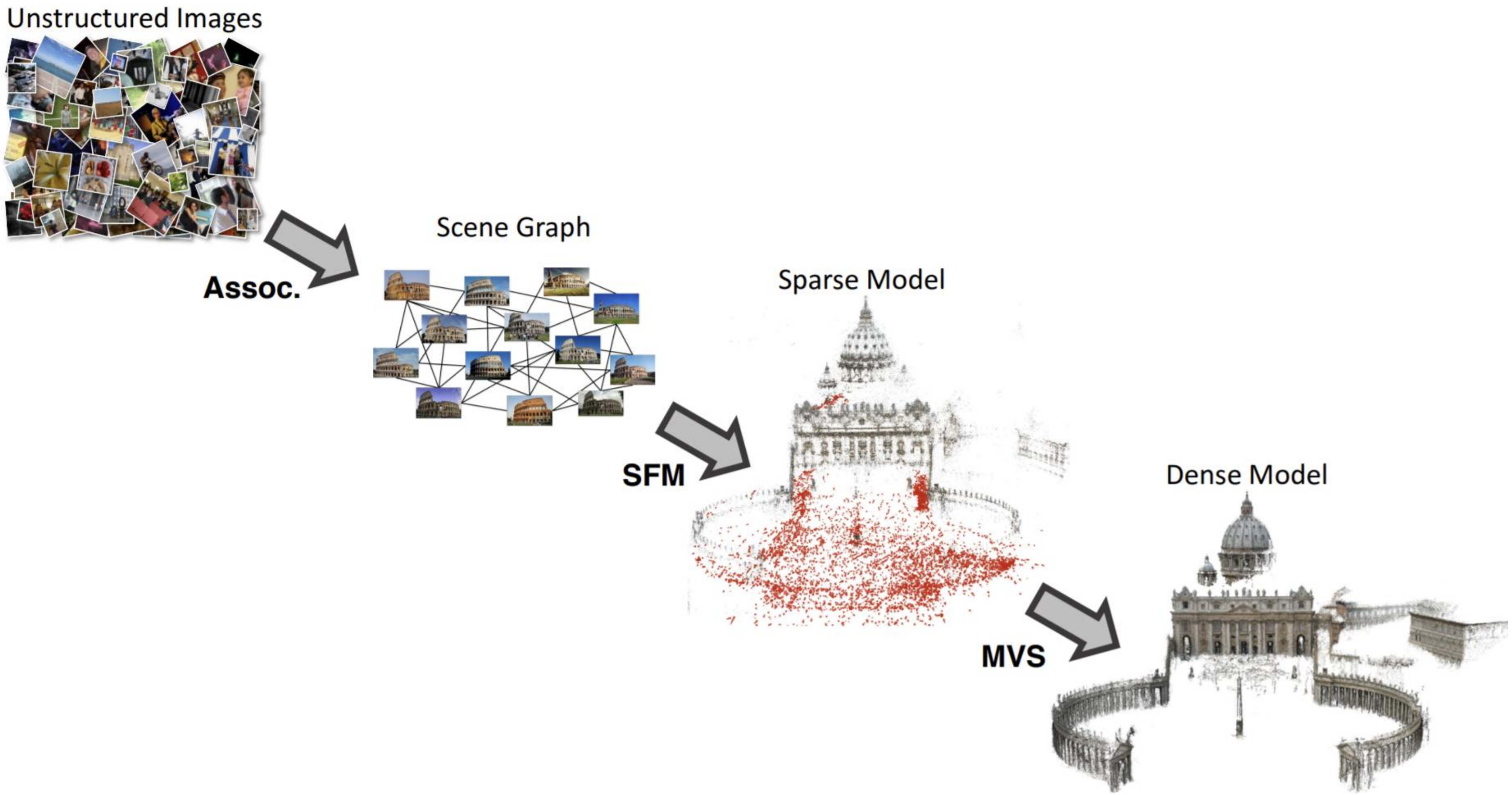
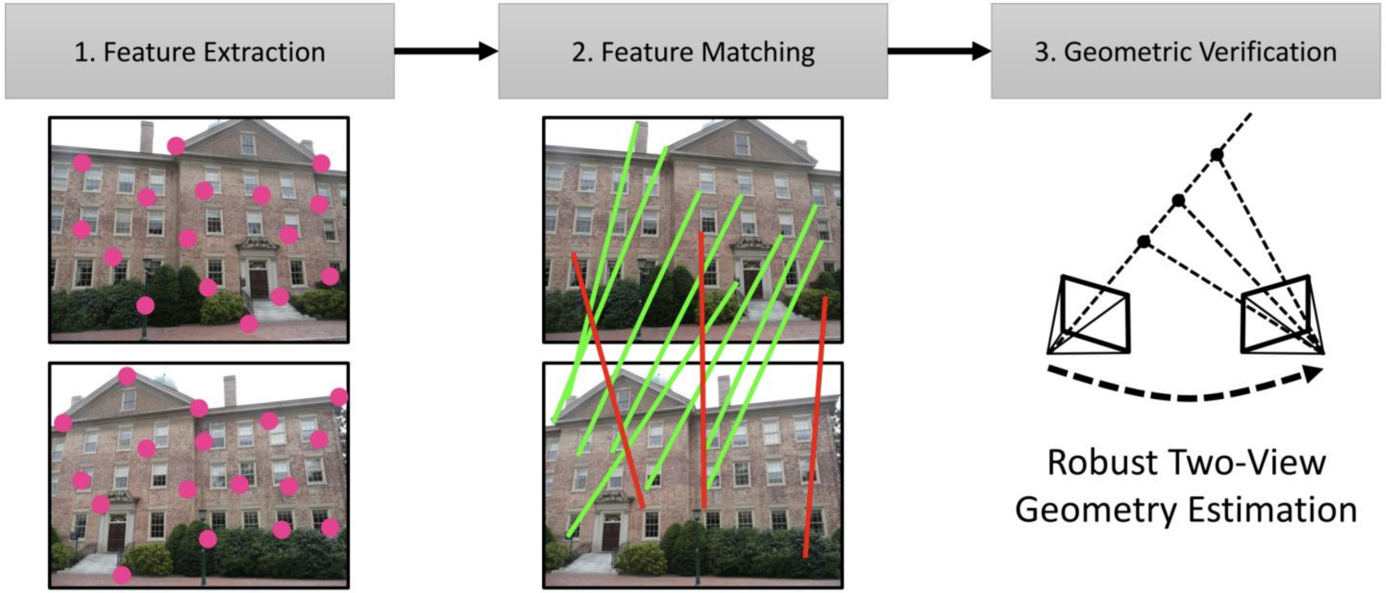
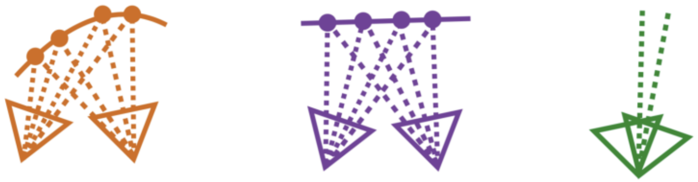


Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

Data Association



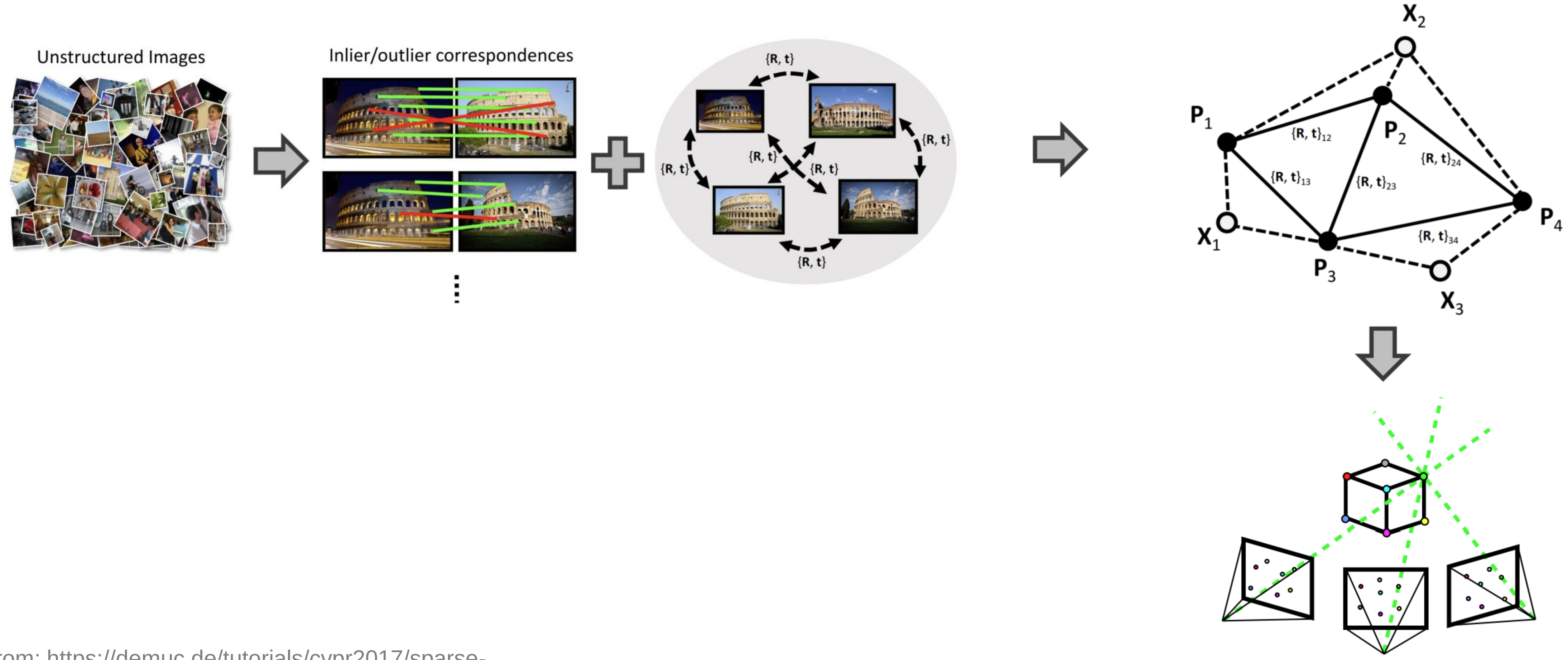
General	Planar	Panoramic
• Fundamental matrix F (*uncalibrated*) • Essential matrix E (*calibrated*)	• Homography H	• Homography H
• 7 correspondences • 5 correspondences	• 4 correspondences	• 4 correspondences



Images from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

Data Association

- Data association creates a graph of cameras/views and landmark points
  - Similar to the camera marker graph



Images from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

The Math Core of SfM

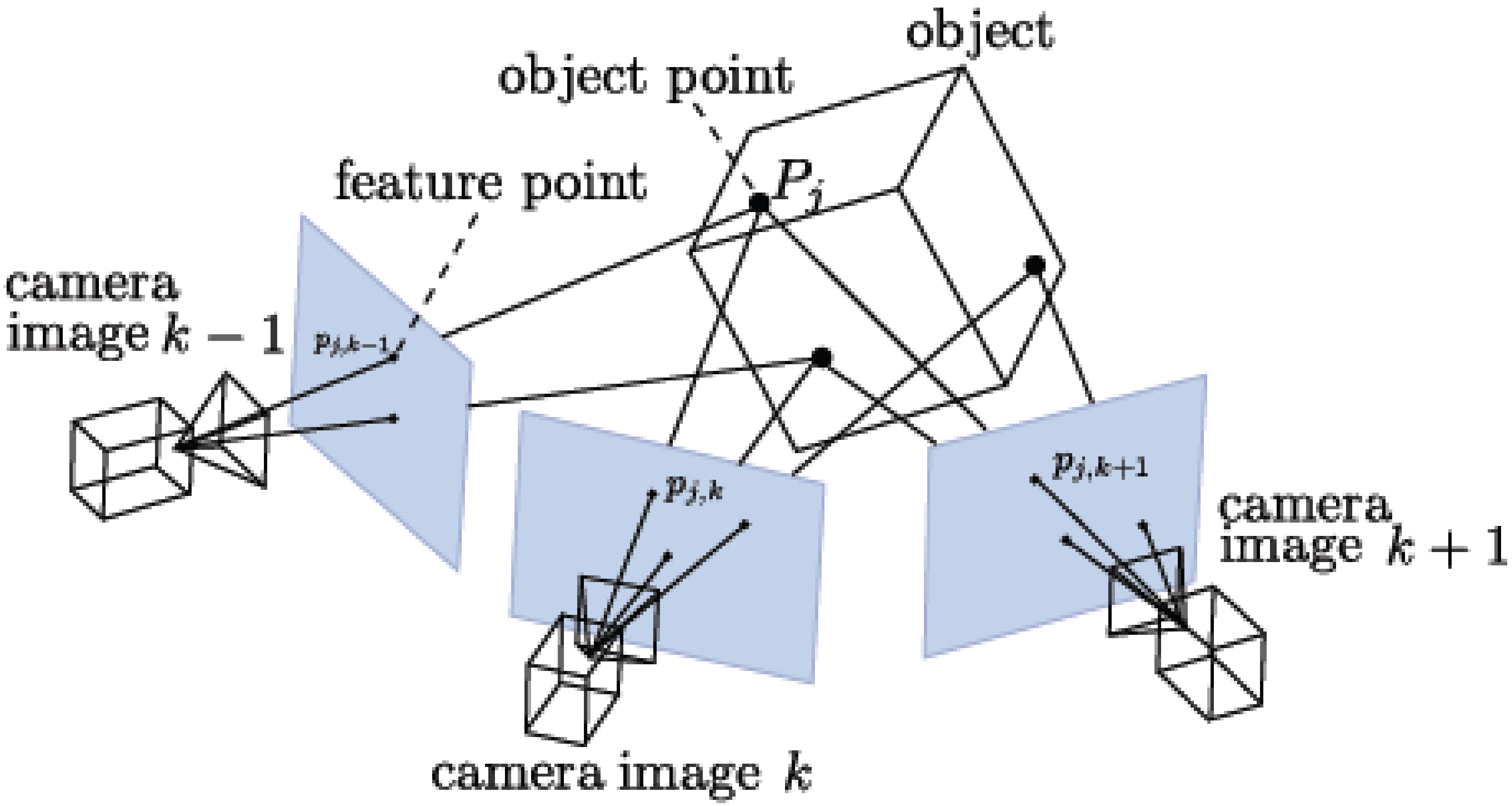
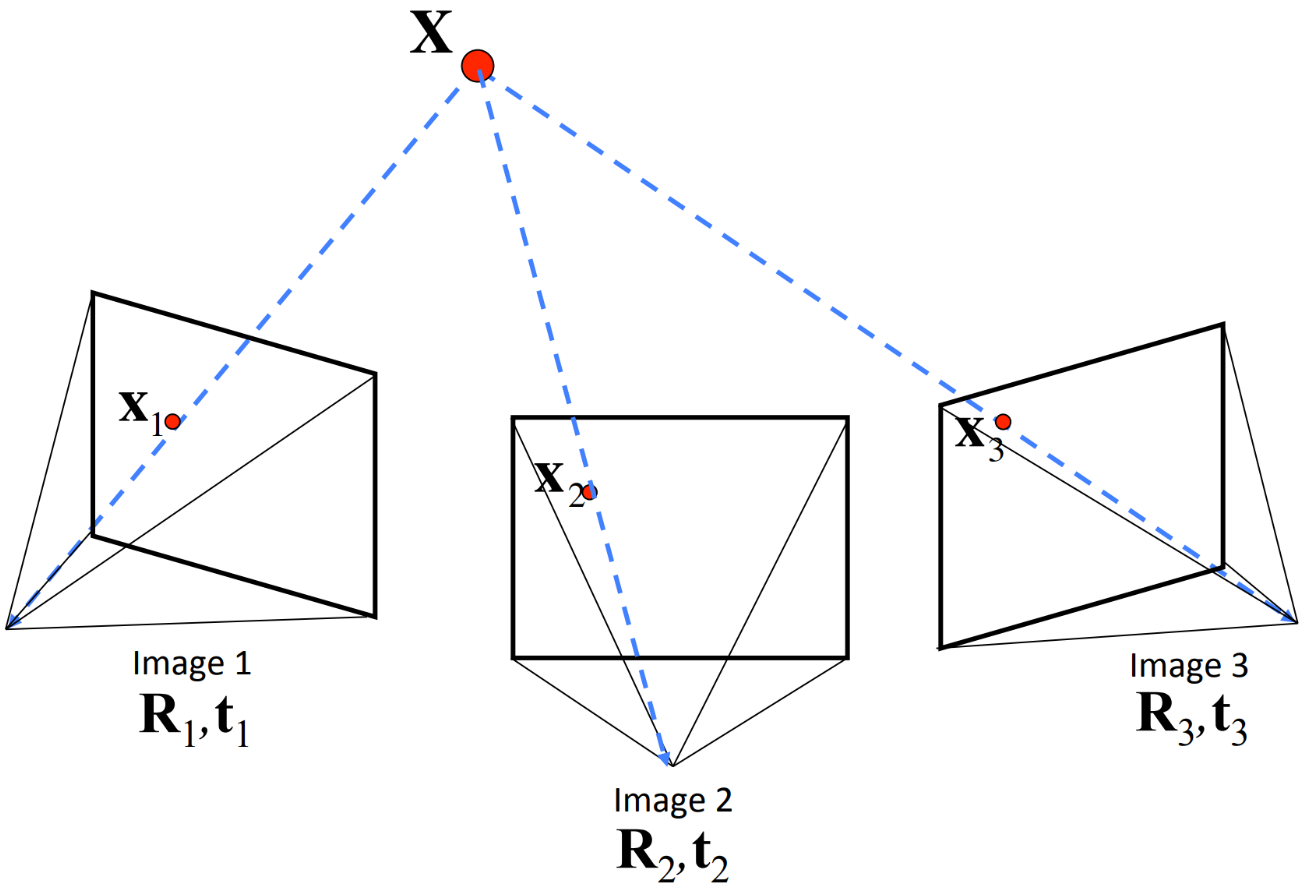


Image from:
https://openmvg.readthedocs.io/en/latest/openMVG/sfm/sfm/

The Math Model of Multiple Photos

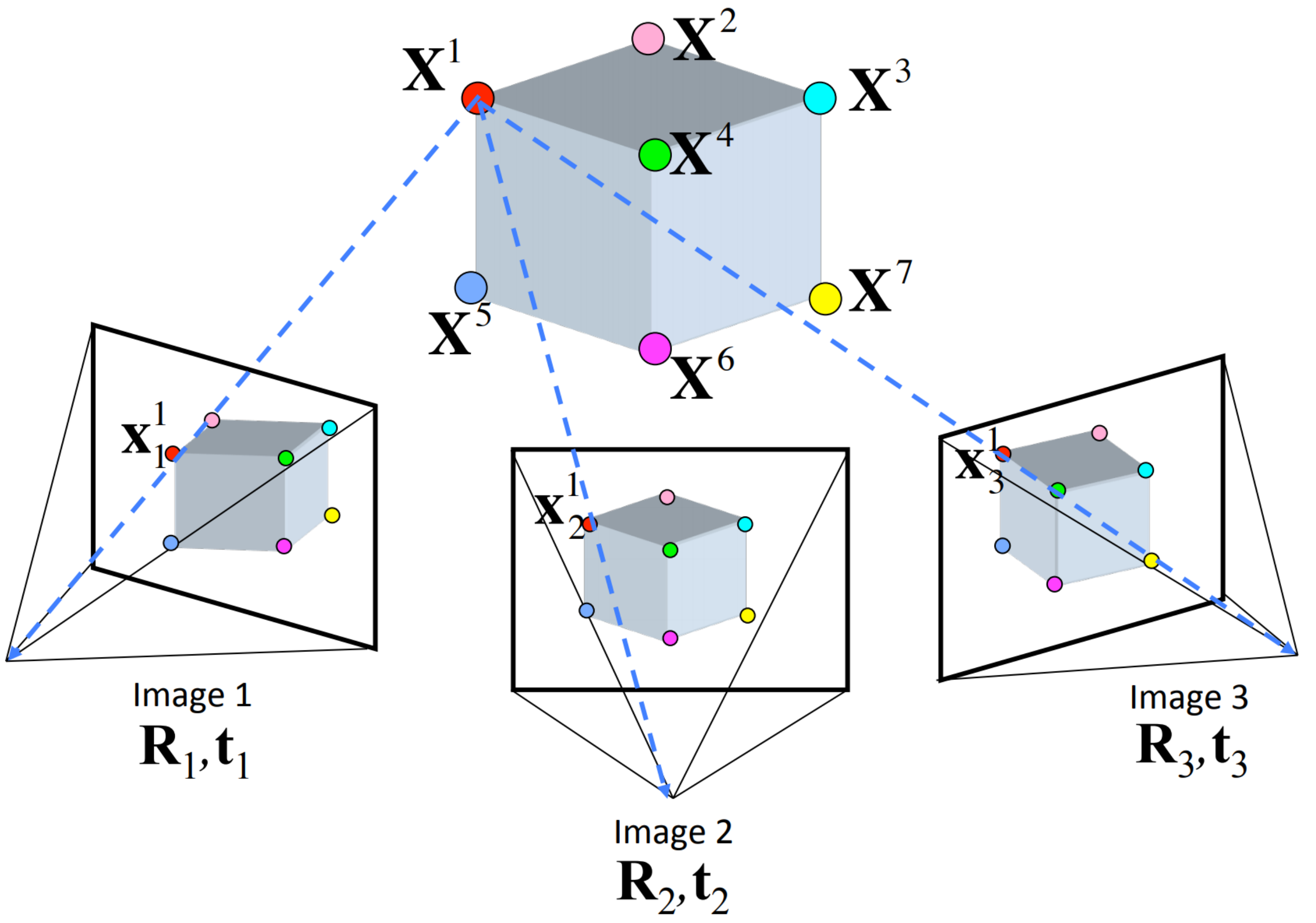


$$\mathbf{x}_1 = \mathbf{K}\left[\mathbf{R}_1 \middle| \mathbf{t}_1\right]\mathbf{X}$$

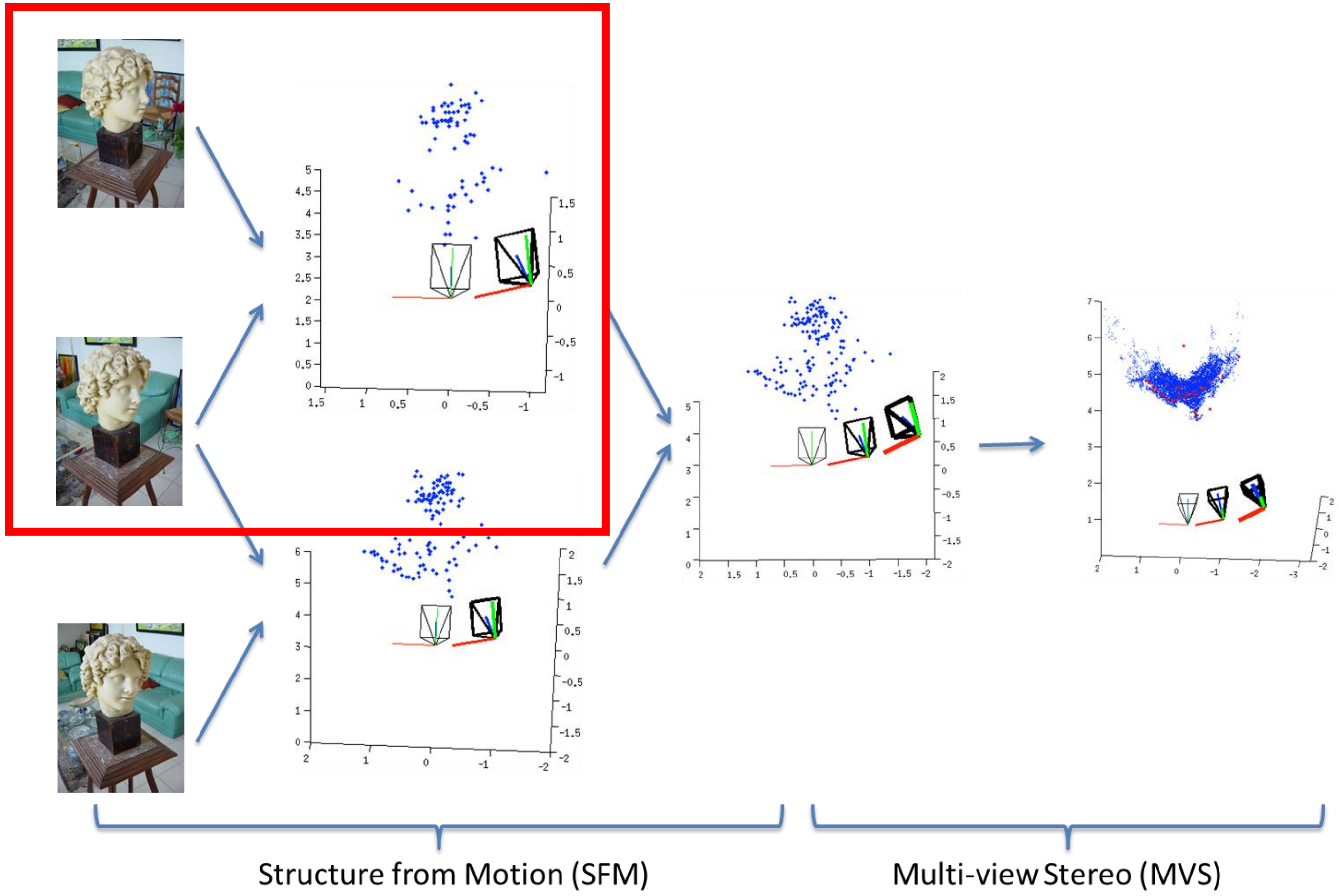
$$\mathbf{x}_2 = \mathbf{K}\left[\mathbf{R}_2 \middle| \mathbf{t}_2\right]\mathbf{X}$$

$$\mathbf{x}_3 = \mathbf{K}\left[\mathbf{R}_3 \middle| \mathbf{t}_3\right]\mathbf{X}$$

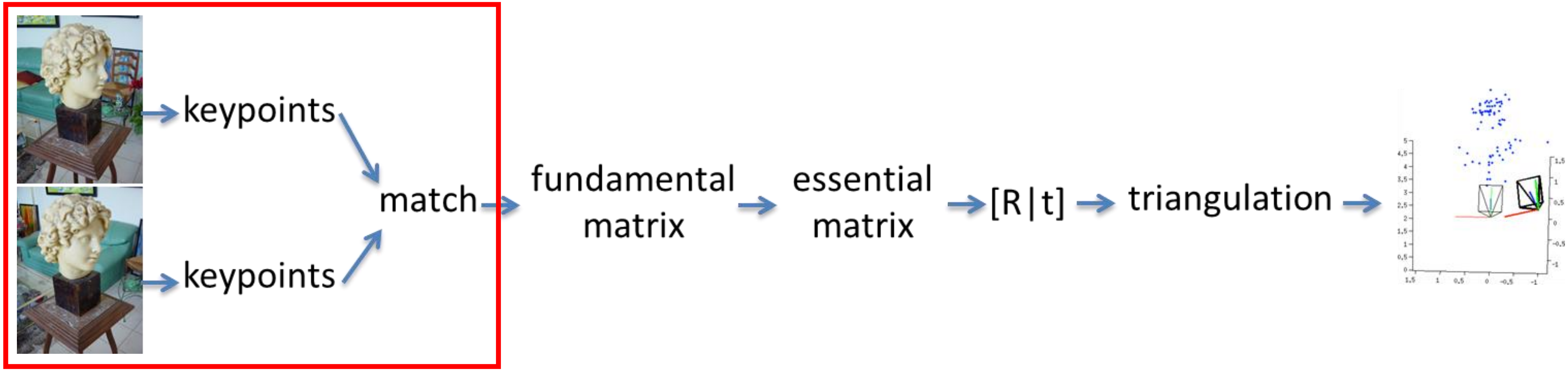
Multiple Photos of Multiple Points



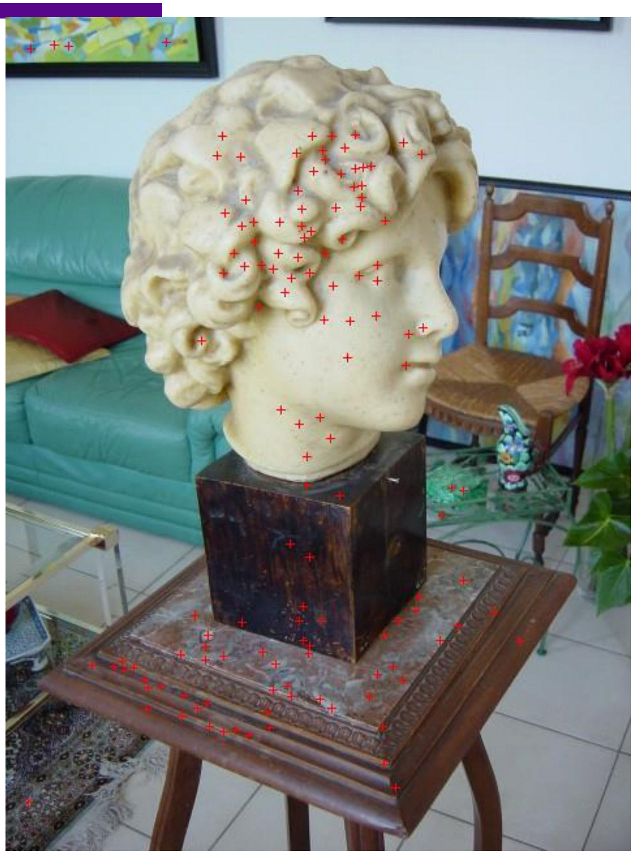
Let’s Look At a Simple Example

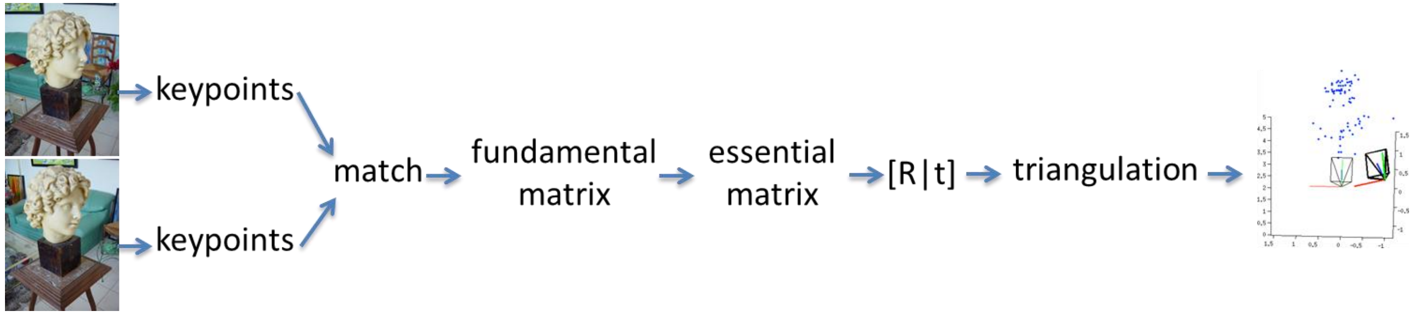


Two-view Reconstruction

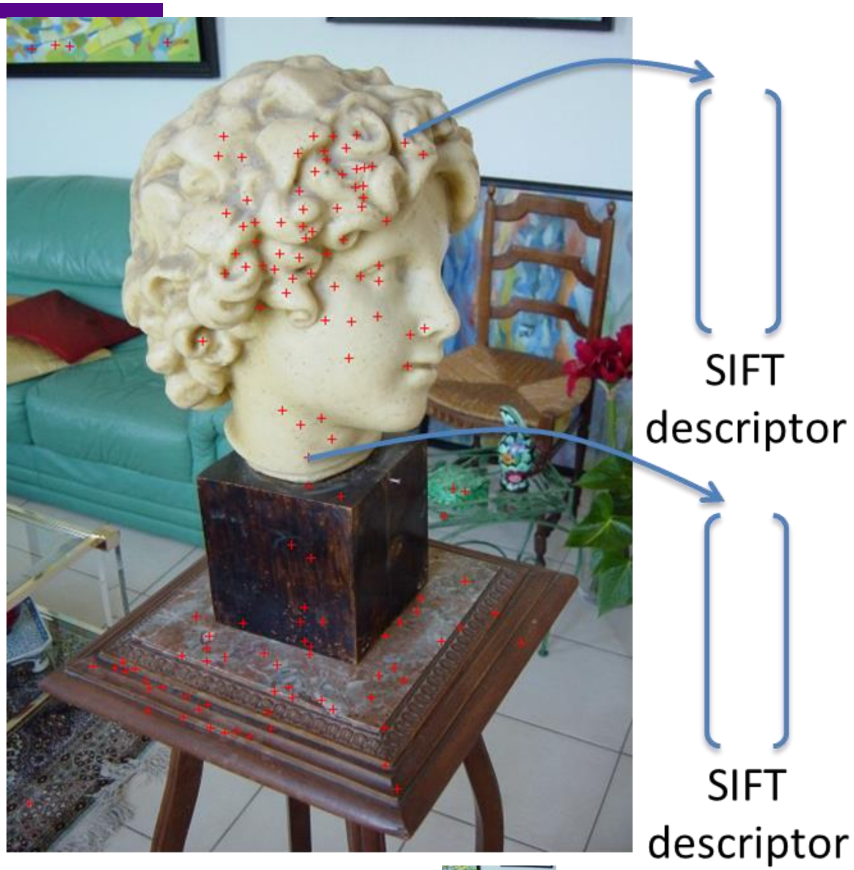


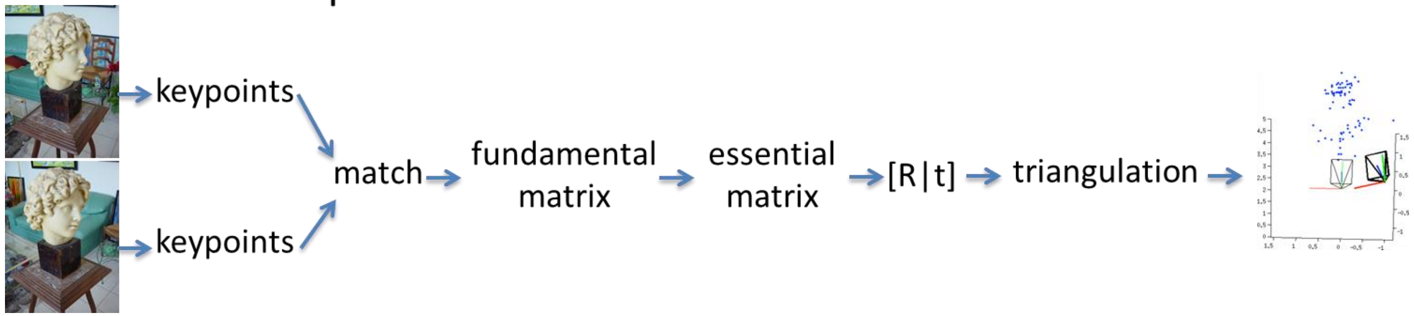
Keypoint Detection



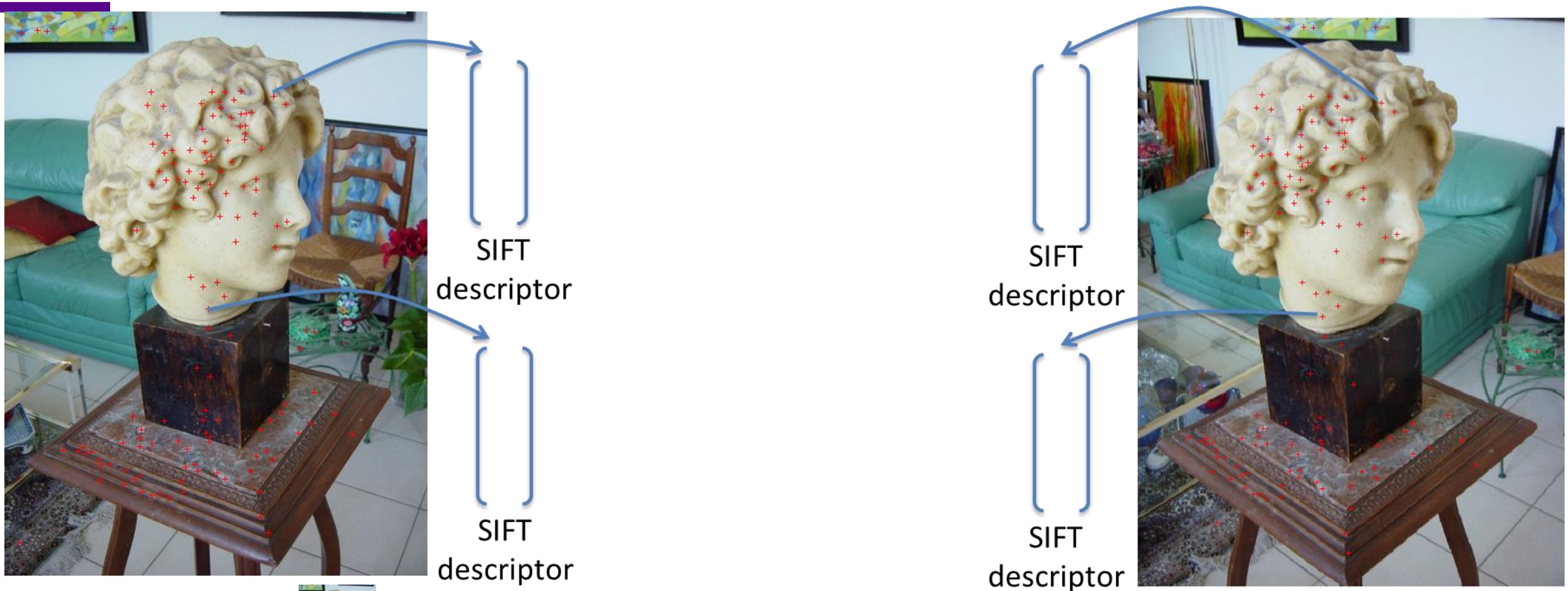


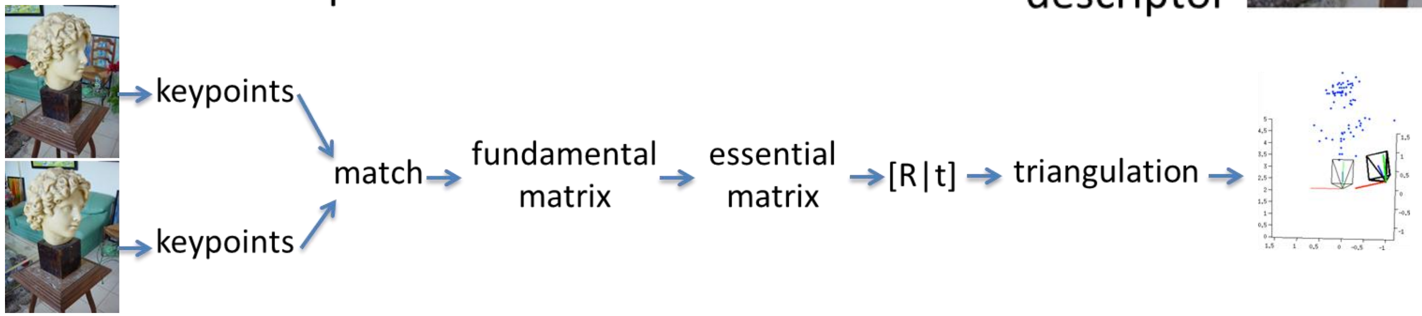
Keypoint Description



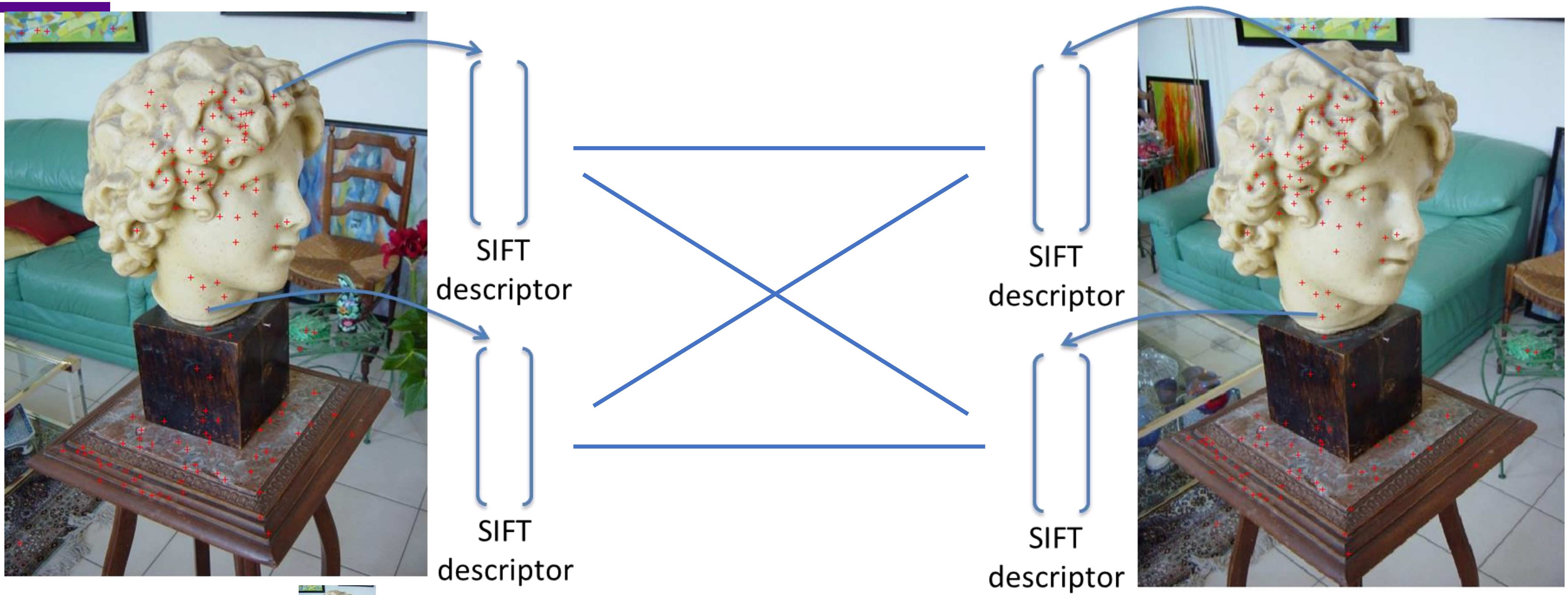


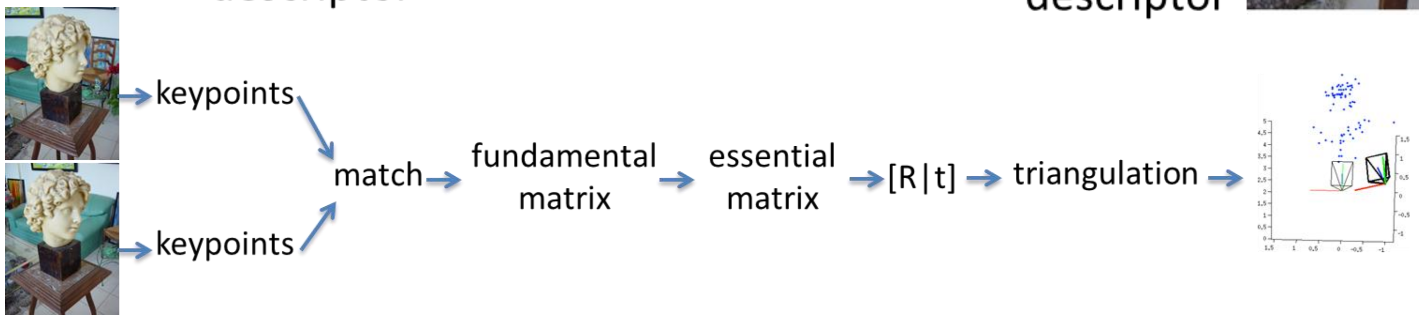
Repeat for the other image



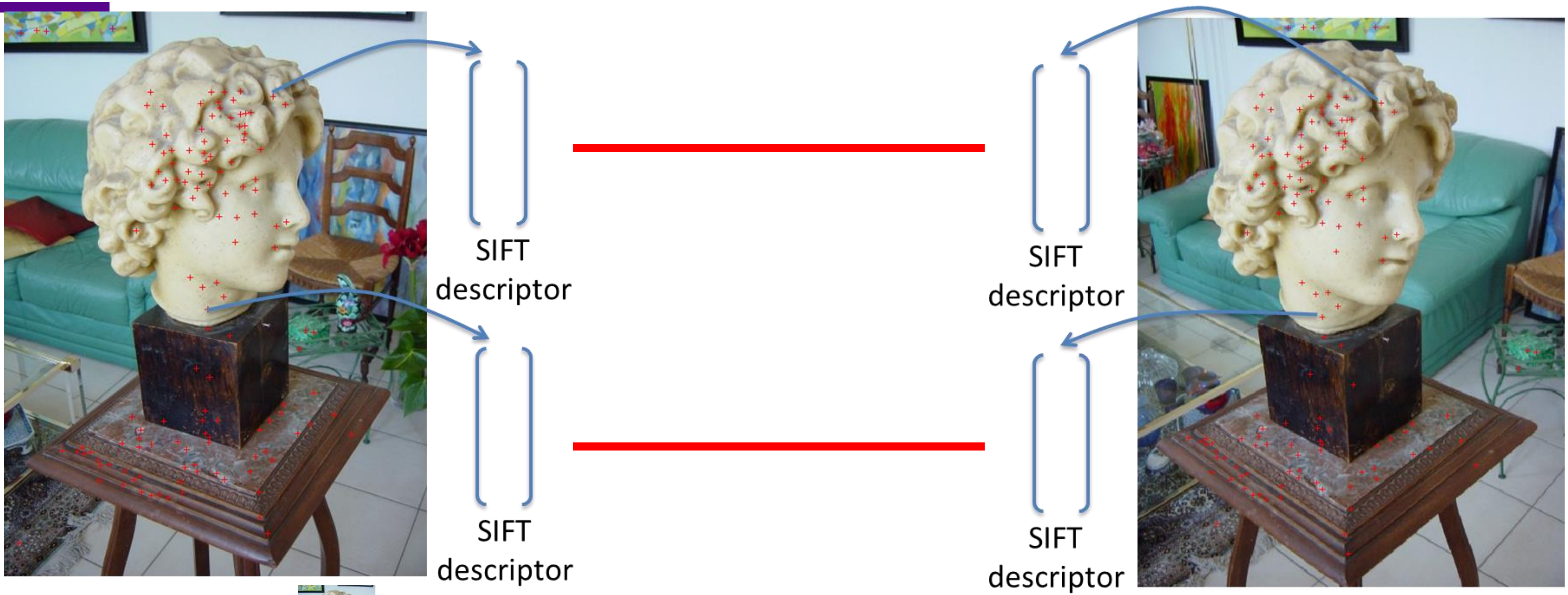


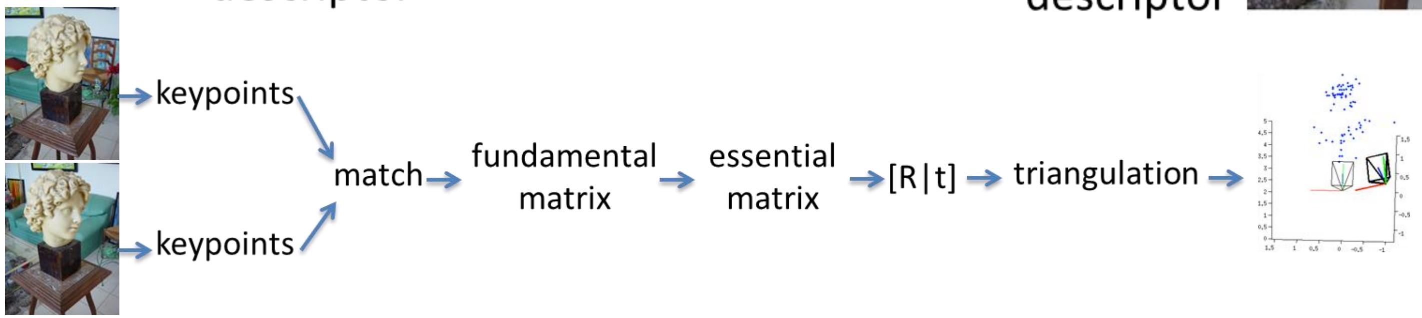
Keypoint Matching



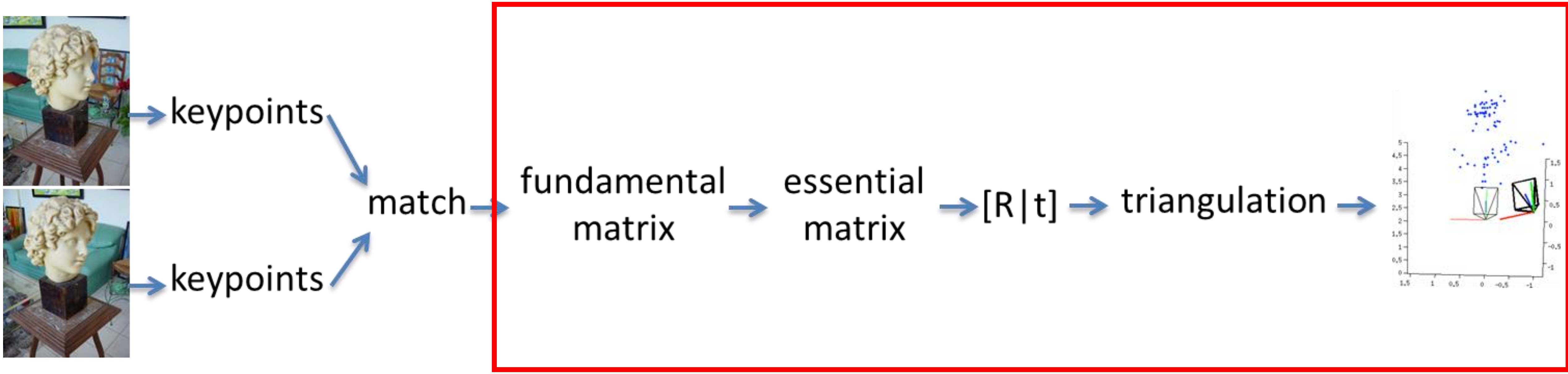


Find Keypoint Correspondences

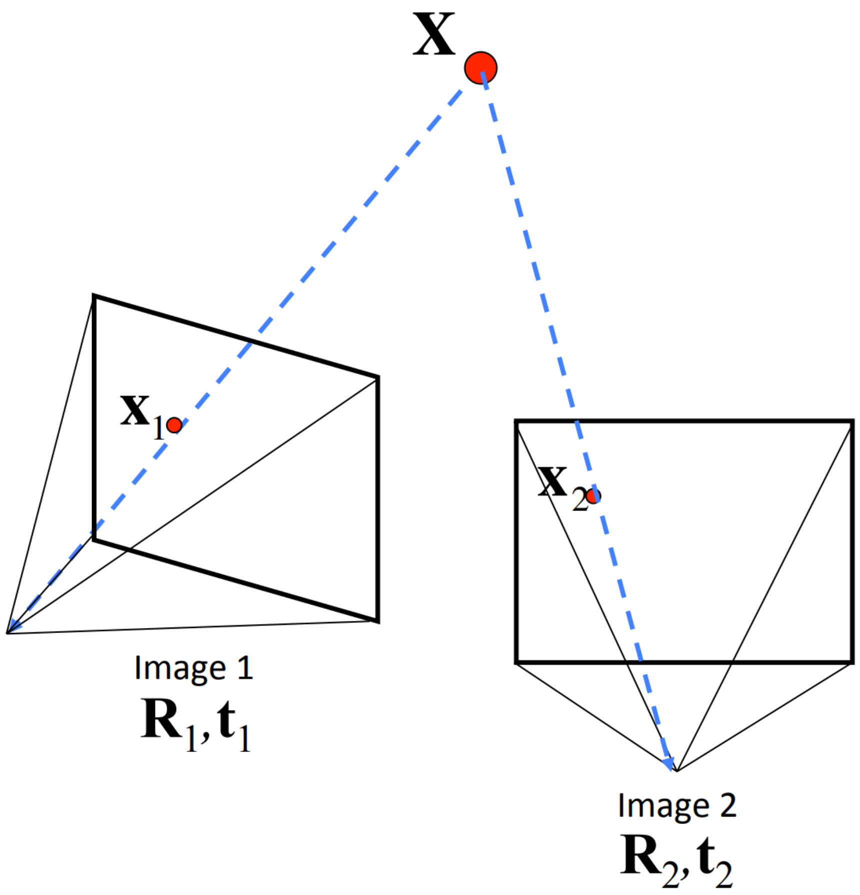




Two-view Reconstruction



Relative Pose ←→ Fundamental Matrix

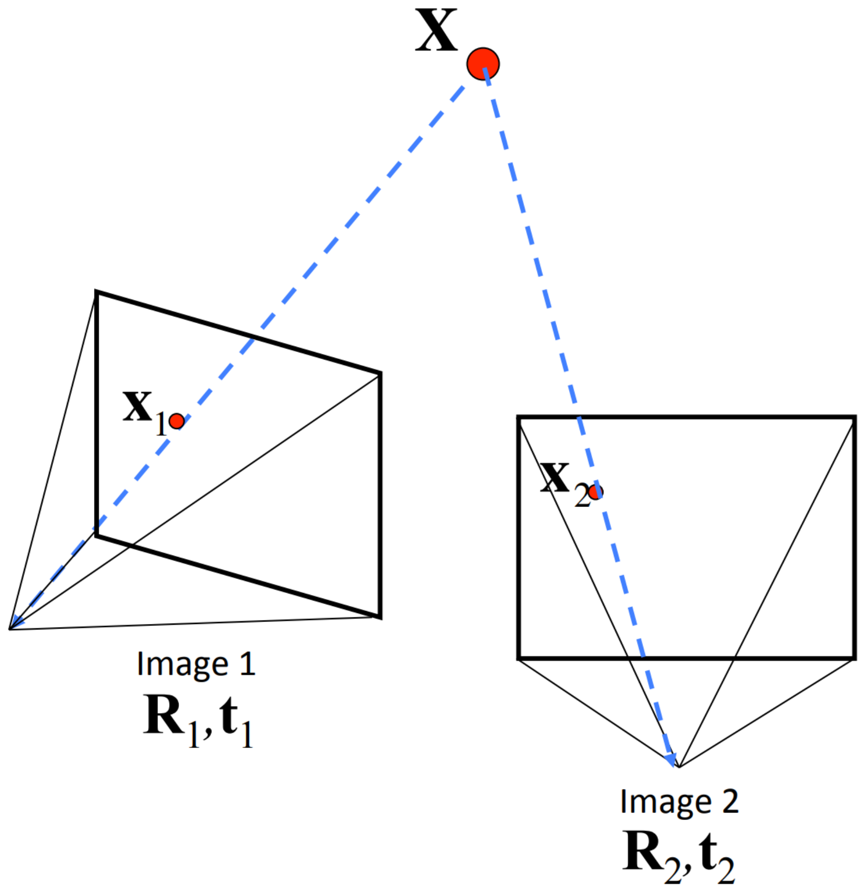


$$\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$$

$$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$$

F-matrix are estimated by finding multiple pairs of such corresponding keypoints ($\mathbf{x}_1$, $\mathbf{x}_2$)

Fundamental Matrix ←→ Essential Matrix

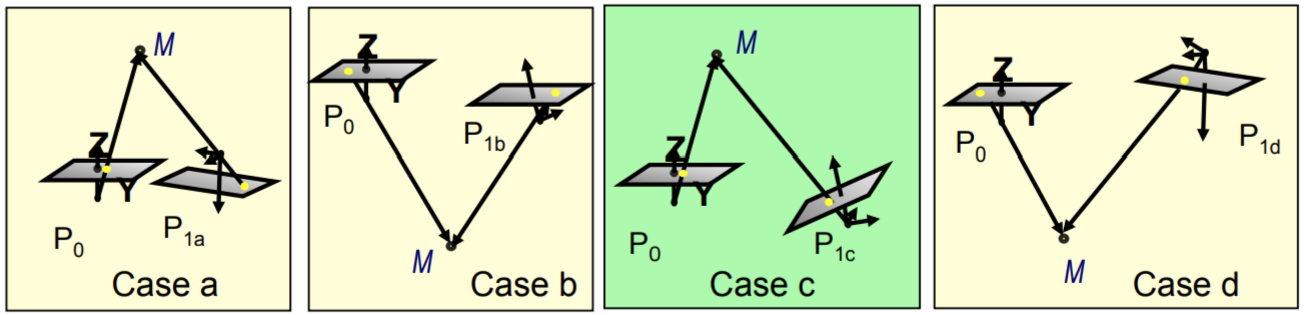


$$\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$$

$$\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$$

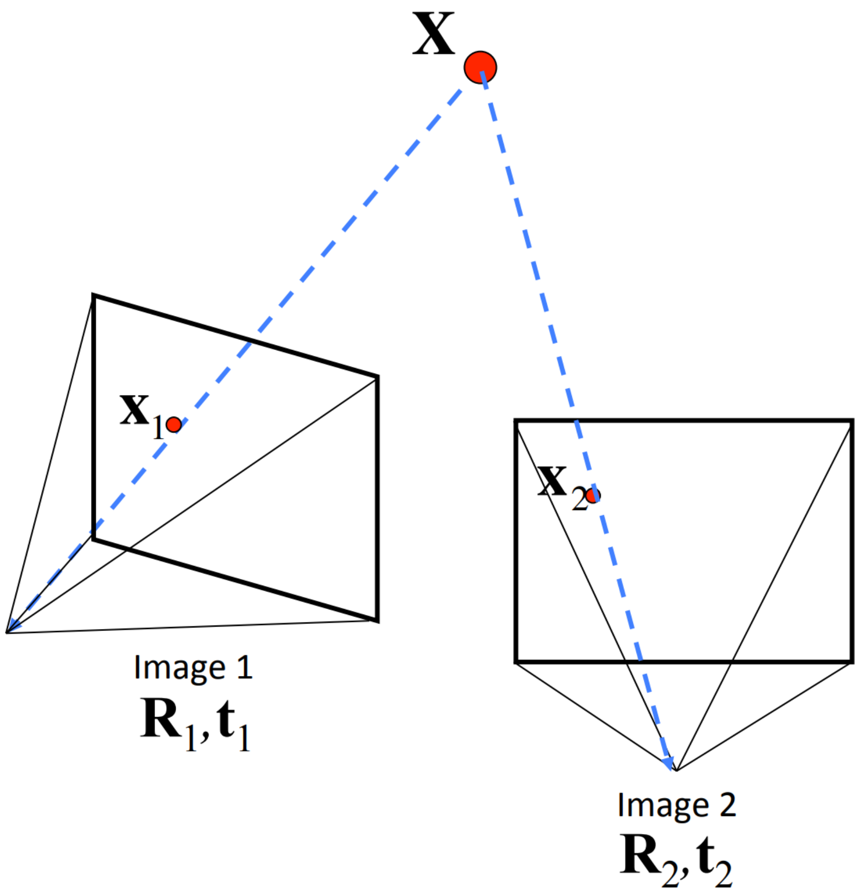
$$\mathbf{E} = \mathbf{K}_1^T \mathbf{F} \mathbf{K}_2$$

E-matrix are estimated by F-matrix with camera intrinsic parameters, and are used to estimate relative pose via decomposition $[t]_x R$



Triangulation

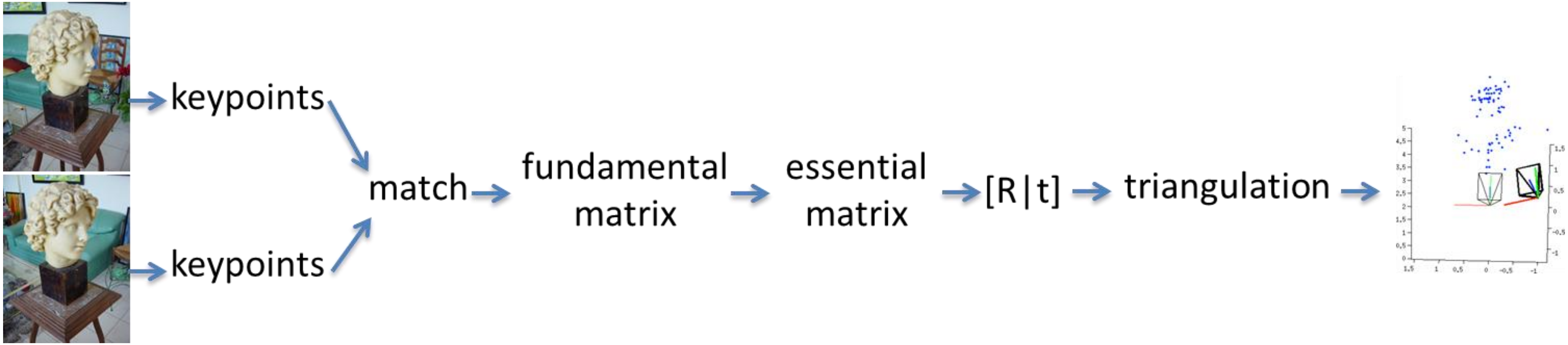
- Solve X=?



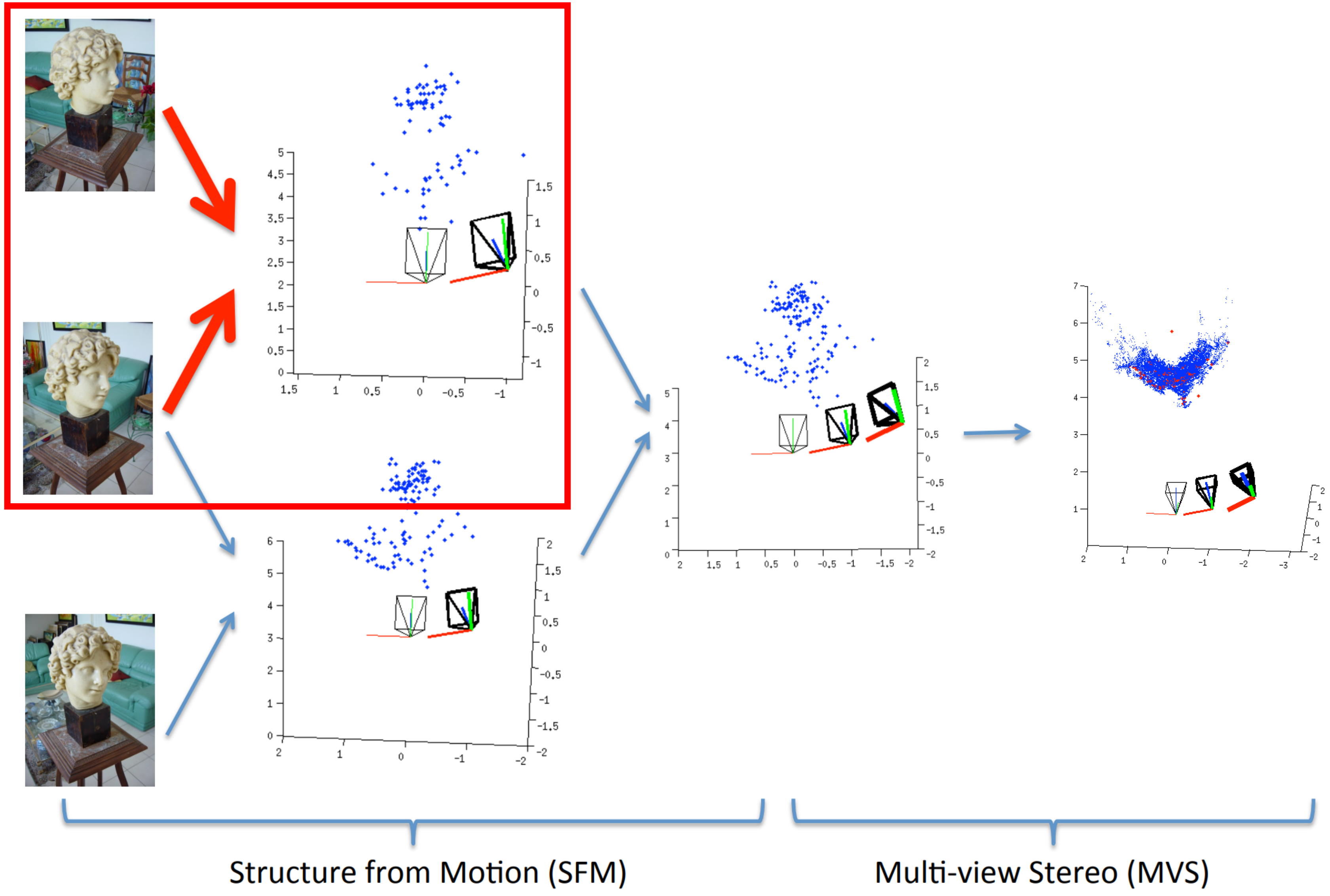
$$\mathbf{x}_1 = \mathbf{K}\left[\mathbf{R}_1 \middle| \mathbf{t}_1\right]\mathbf{X}$$

$$\mathbf{x}_2 = \mathbf{K}\left[\mathbf{R}_2 \middle| \mathbf{t}_2\right]\mathbf{X}$$

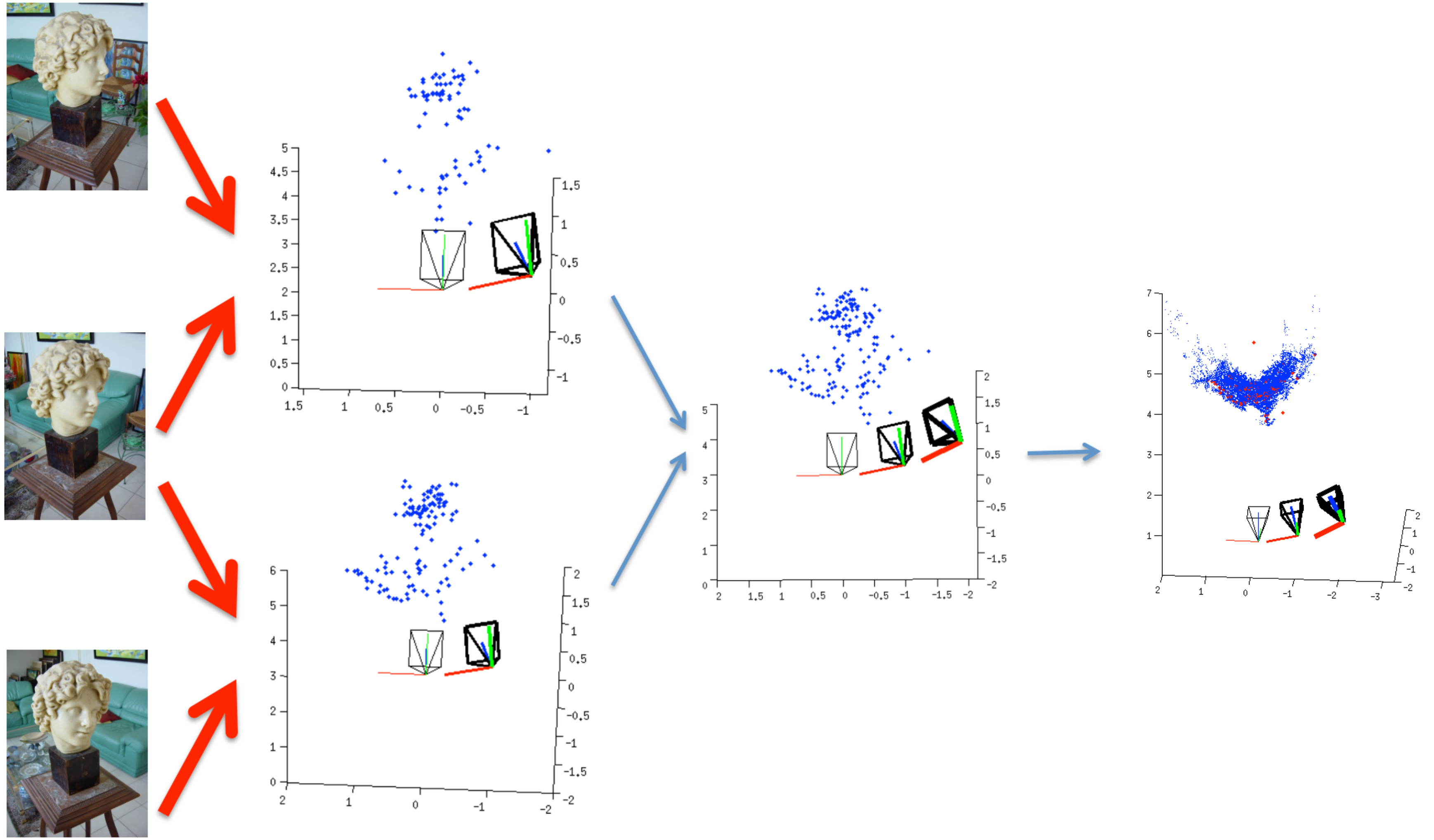
Recap: Two-view Reconstruction



Multi-view Reconstruction

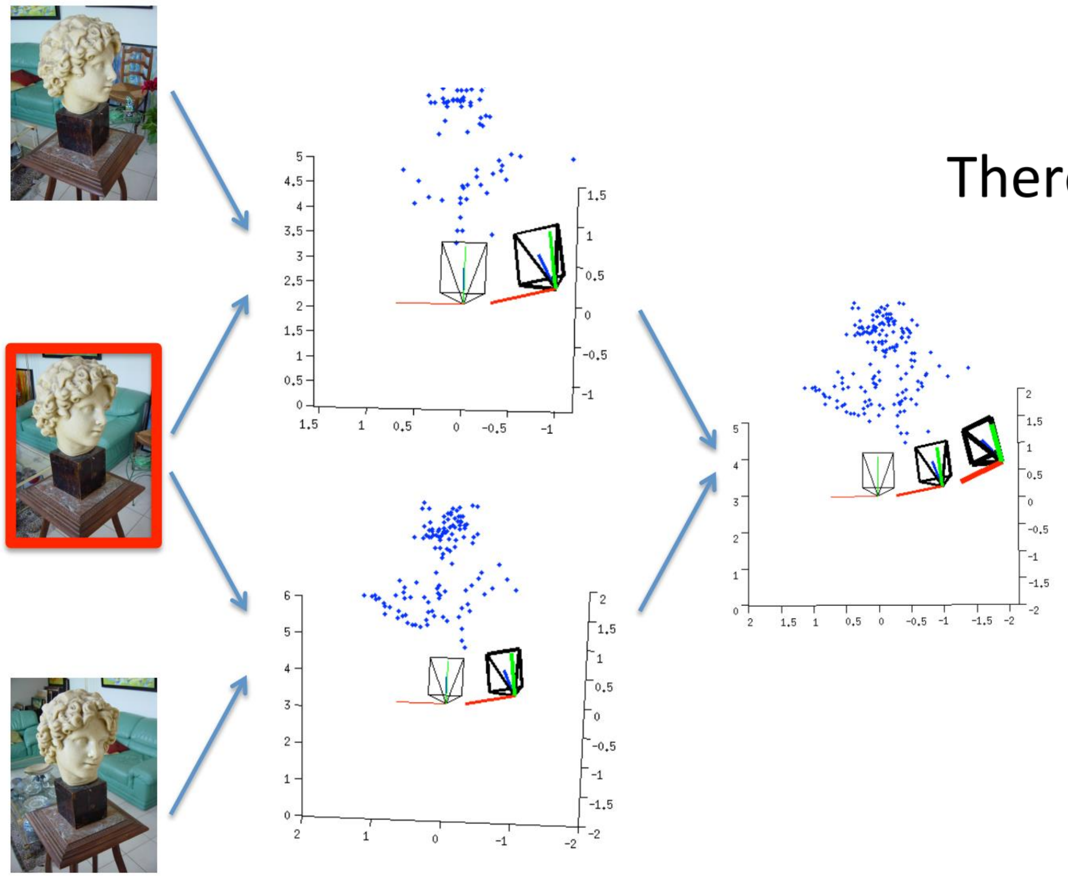


Multi-view Reconstruction



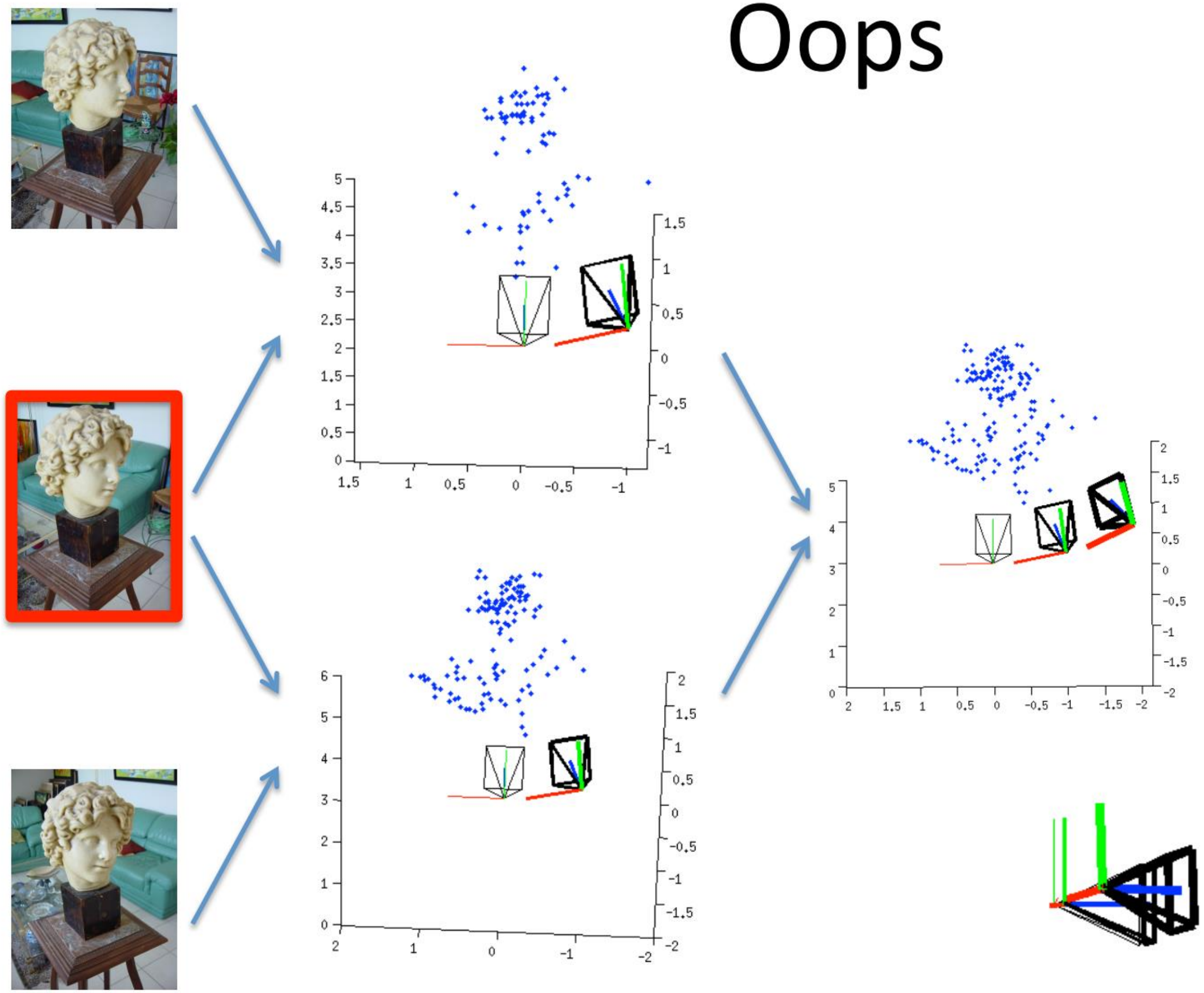
Taught Next

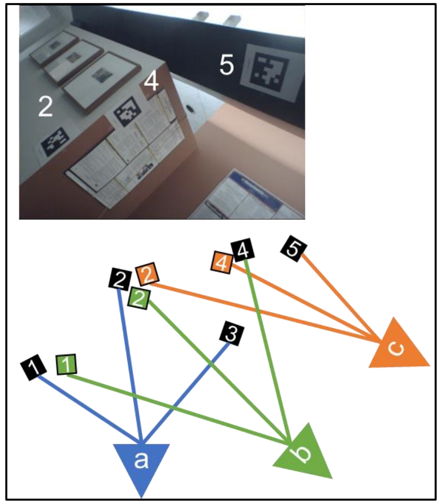
How to Merge Two Point Clouds?

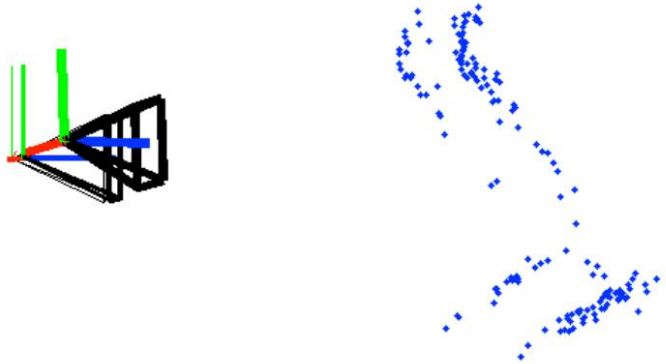


There can be only one $\left[\mathbf{R}_2 | \mathbf{t}_2\right]$

Oops

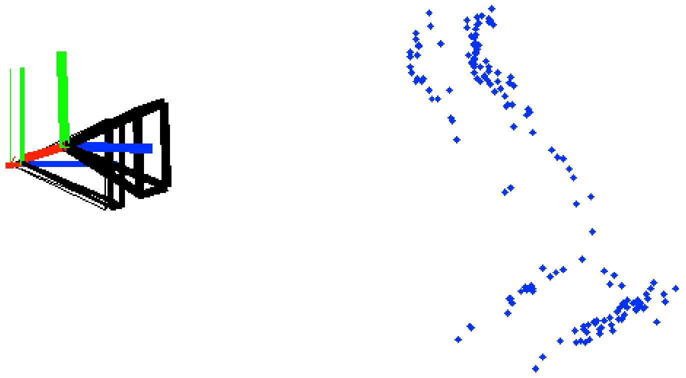


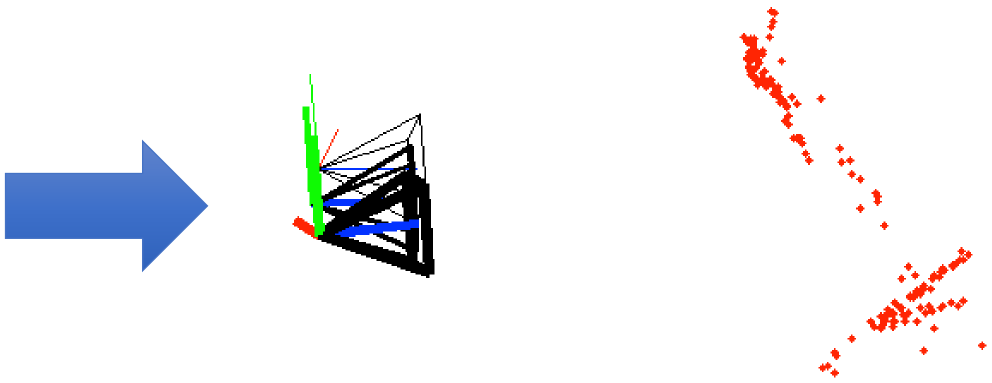




See From a Different Angle

Bundle Adjustment Come to the Rescue

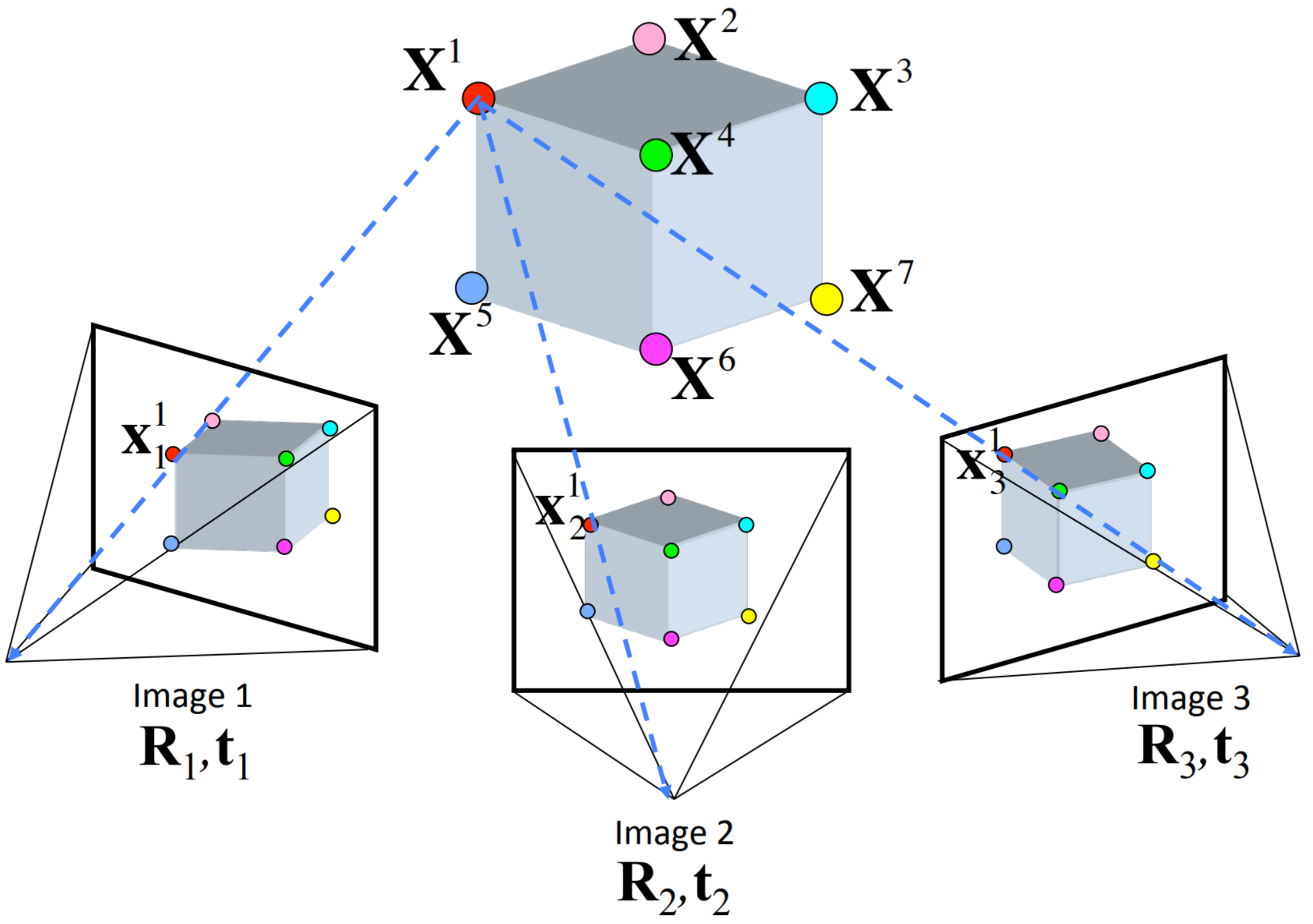




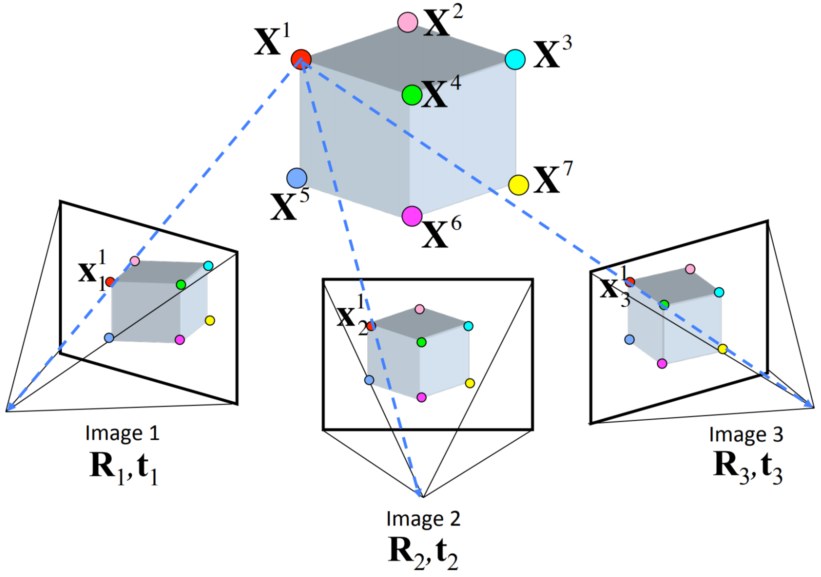
If assume all [R,t] are correct, error are all shown in point cloud.

Jointly optimize all [R,t] and points

Recall: Multiple Photos of Multiple Points



Writing in Math



	Point 1	Point 2	Point 3
Image 1	$\mathbf{x}_1^1 = \mathbf{K}\left[\mathbf{R}_1 \middle\vert \mathbf{t}_1\right]\mathbf{X}^1$	$\mathbf{x}_1^2 = \mathbf{K}\left[\mathbf{R}_1 \middle\vert \mathbf{t}_1\right]\mathbf{X}^2$	
Image 2	$\mathbf{x}_2^1 = \mathbf{K}\left[\mathbf{R}_2 \middle\vert \mathbf{t}_2\right]\mathbf{X}^1$	$\mathbf{x}_2^2 = \mathbf{K}\left[\mathbf{R}_2 \middle\vert \mathbf{t}_2\right]\mathbf{X}^2$	$\mathbf{x}_2^3 = \mathbf{K}\left[\mathbf{R}_2 \middle\vert \mathbf{t}_2\right]\mathbf{X}^3$
Image 3	$\mathbf{x}_3^1 = \mathbf{K}\left[\mathbf{R}_3 \middle\vert \mathbf{t}_3\right]\mathbf{X}^1$		$\mathbf{x}_3^3 = \mathbf{K}\left[\mathbf{R}_3 \middle\vert \mathbf{t}_3\right]\mathbf{X}^3$

Formulating the SfM Problem Mathematically

- Input: Observed 2D image position

$$
\begin{array}{ccc}
\tilde{\mathbf{x}}_1^1 & \tilde{\mathbf{x}}_1^2 & \\
\tilde{\mathbf{x}}_2^1 & \tilde{\mathbf{x}}_2^2 & \tilde{\mathbf{x}}_2^3 \\
\tilde{\mathbf{x}}_3^1 & & \tilde{\mathbf{x}}_3^3
\end{array}
$$

- Output:

Unknown Camera Parameters (with some guess)

$$[\mathbf{R}_1|\mathbf{t}_1],[\mathbf{R}_2|\mathbf{t}_2],[\mathbf{R}_3|\mathbf{t}_3]$$

Unknown Point 3D coordinate (with some guess)

$$\mathbf{X}^1,\mathbf{X}^2,\mathbf{X}^3,\cdots$$

Bundle Adjustment

A valid solution $[\mathbf{R}_1|\mathbf{t}_1],[\mathbf{R}_2|\mathbf{t}_2],[\mathbf{R}_3|\mathbf{t}_3]$ and $\mathbf{X}^1,\mathbf{X}^2,\mathbf{X}^3,\cdots$ must let

Re-projection

$$\begin{cases}
\mathbf{x}_1^1 = \mathbf{K}[\mathbf{R}_1|\mathbf{t}_1]\mathbf{X}^1 & \mathbf{x}_1^2 = \mathbf{K}[\mathbf{R}_1|\mathbf{t}_1]\mathbf{X}^2 & \\
\mathbf{x}_2^1 = \mathbf{K}[\mathbf{R}_2|\mathbf{t}_2]\mathbf{X}^1 & \mathbf{x}_2^2 = \mathbf{K}[\mathbf{R}_2|\mathbf{t}_2]\mathbf{X}^2 & \mathbf{x}_2^3 = \mathbf{K}[\mathbf{R}_2|\mathbf{t}_2]\mathbf{X}^3 \\
\mathbf{x}_3^1 = \mathbf{K}[\mathbf{R}_3|\mathbf{t}_3]\mathbf{X}^1 & & \mathbf{x}_3^3 = \mathbf{K}[\mathbf{R}_3|\mathbf{t}_3]\mathbf{X}^3
\end{cases}$$

=

Observation

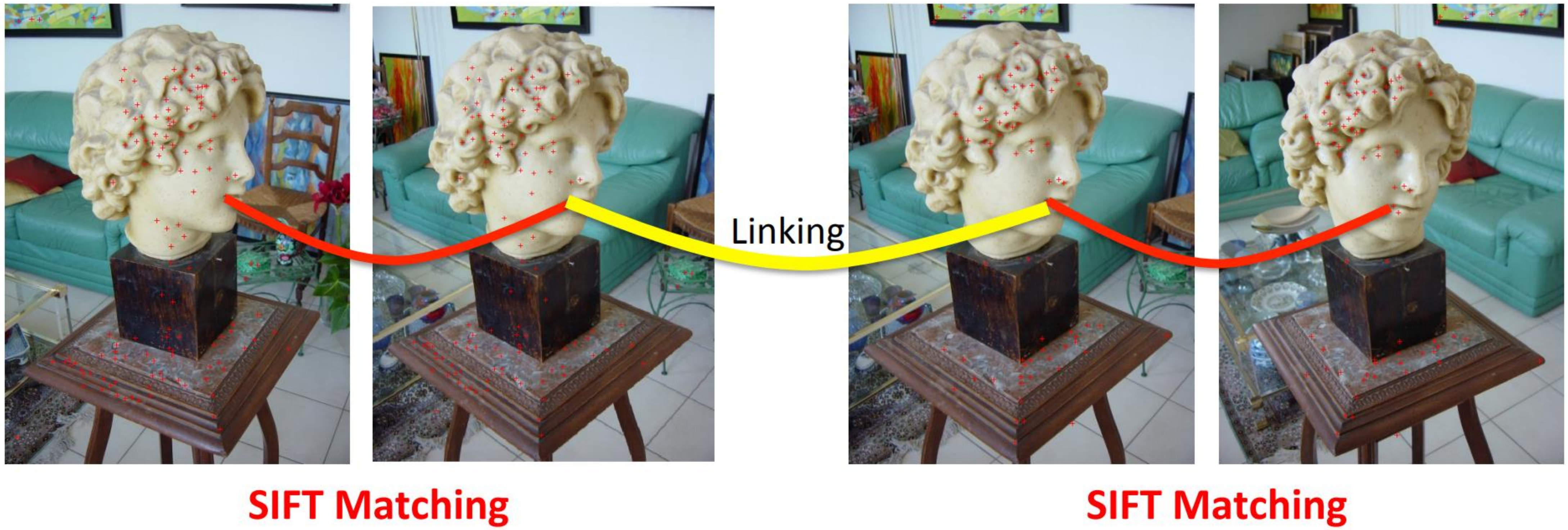
$$\begin{cases}
\tilde{\mathbf{x}}_1^1 & \tilde{\mathbf{x}}_1^2 & \\
\tilde{\mathbf{x}}_2^1 & \tilde{\mathbf{x}}_2^2 & \tilde{\mathbf{x}}_2^3 \\
\tilde{\mathbf{x}}_3^1 & & \tilde{\mathbf{x}}_3^3
\end{cases}$$

Bundle Adjustment is a Least Squares Problem

A valid solution $[\mathbf{R}_1|\mathbf{t}_1],[\mathbf{R}_2|\mathbf{t}_2],[\mathbf{R}_3|\mathbf{t}_3]$ and $\mathbf{X}^1,\mathbf{X}^2,\mathbf{X}^3,\cdots$ must let the Re-projection close to the Observation, i.e. to minimize the reprojection error

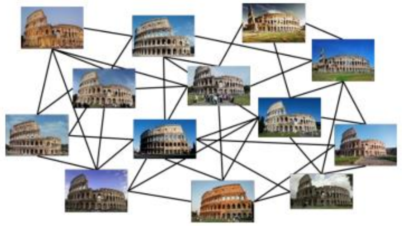
$$\min \sum_i \sum_j \left(\tilde{\mathbf{x}}_i^j - \mathbf{K}[\mathbf{R}_i|\mathbf{t}_i]\mathbf{X}^j\right)^2$$

Keypoint Linking: Build the Objective Function



What We Just Learnt: Incremental SfM

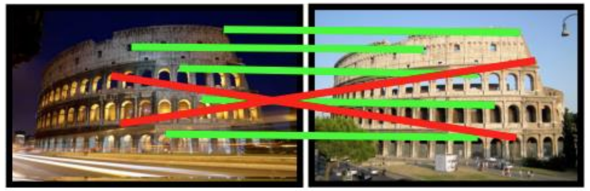
- Initialization
  1. Choose two non-panoramic views ($\|t\| \neq 0$)

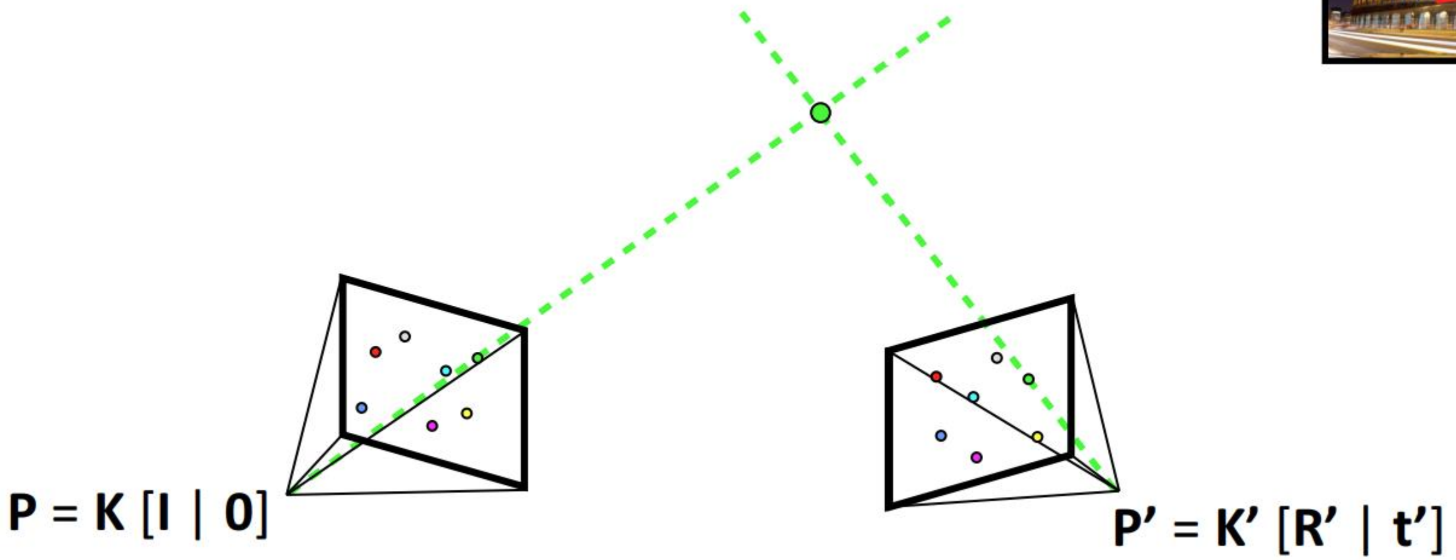


Slides from Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

Incremental SfM

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

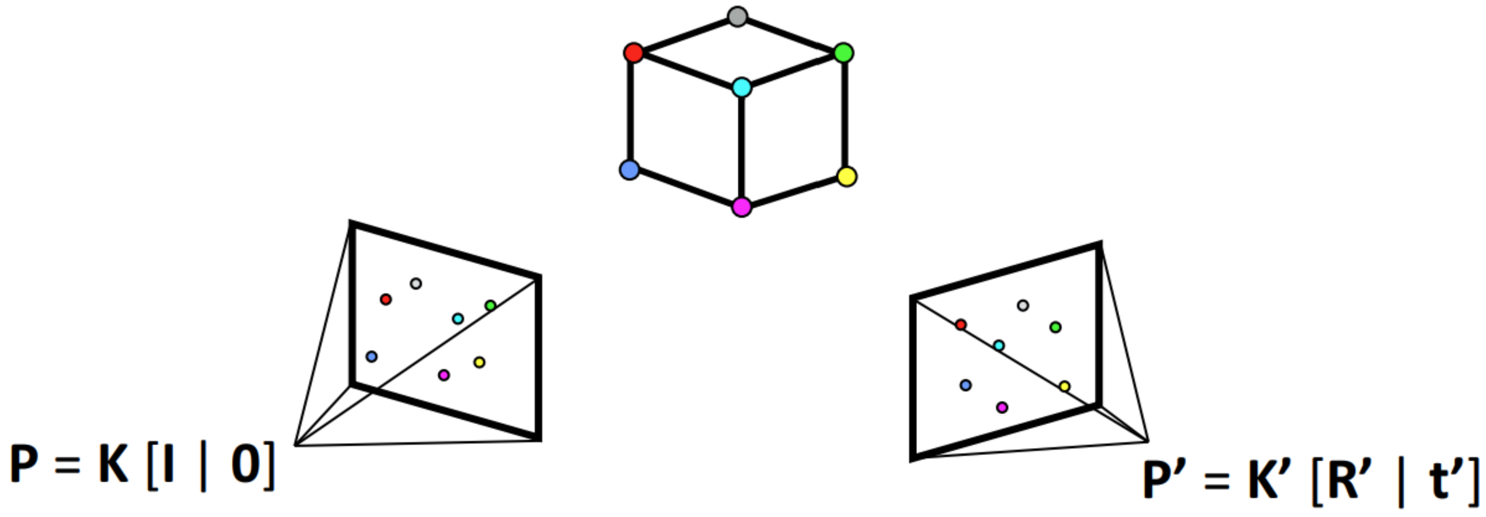




Slides from Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

Incremental SfM

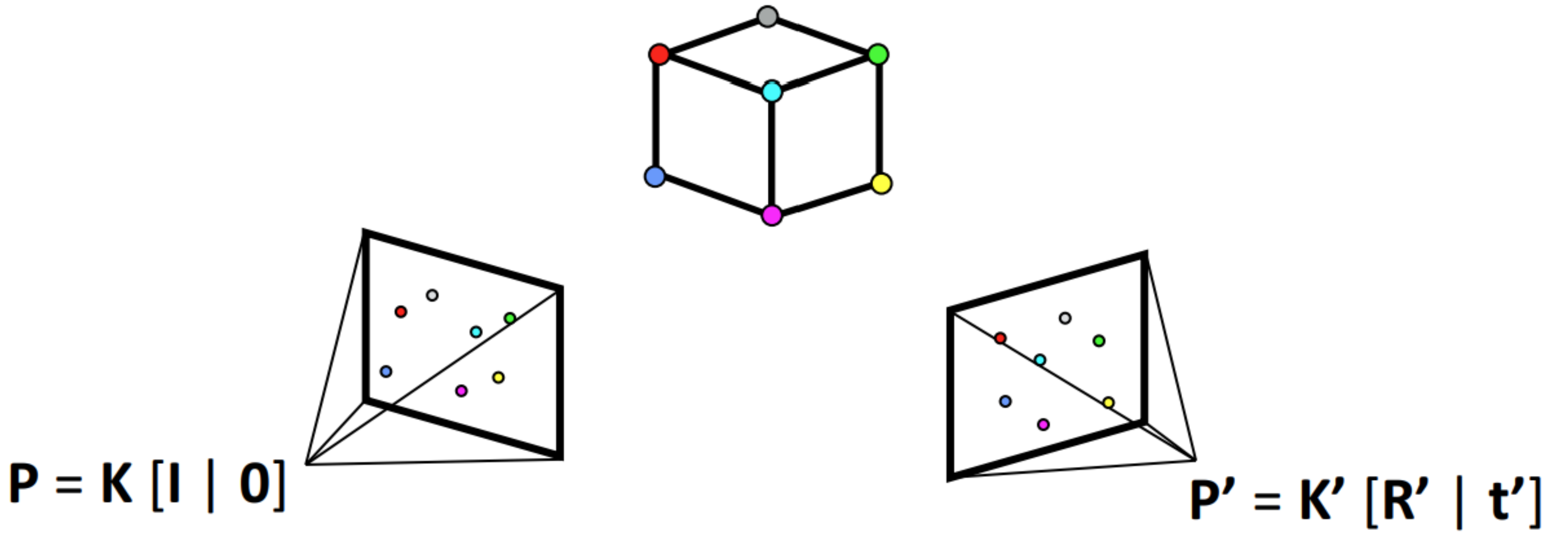
- Initialization
  1. Choose two non-panoramic views ($||t|| = 1$)
  2. Triangulate inlier correspondences



Slides from Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

Incremental SfM

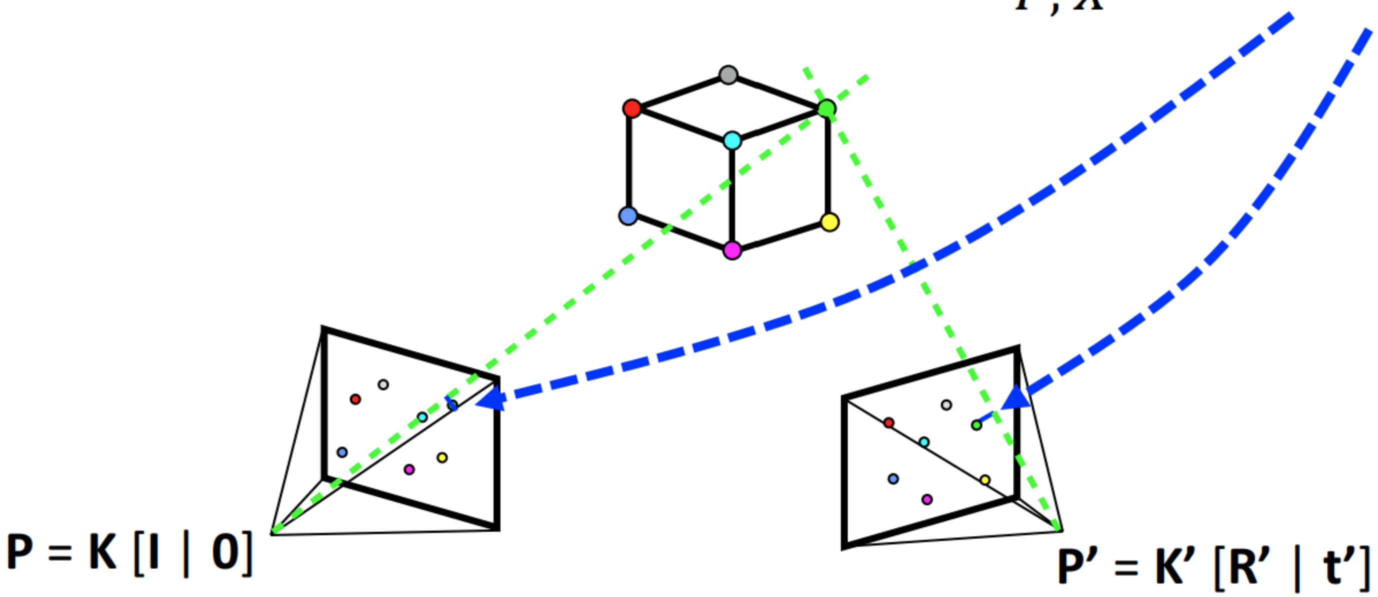
- Initialization
  1. Choose two non-panoramic views ($||t|| = 1$)
  2. Triangulate inlier correspondences
  3. Bundle adjustment



Slides from Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

Incremental SfM

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

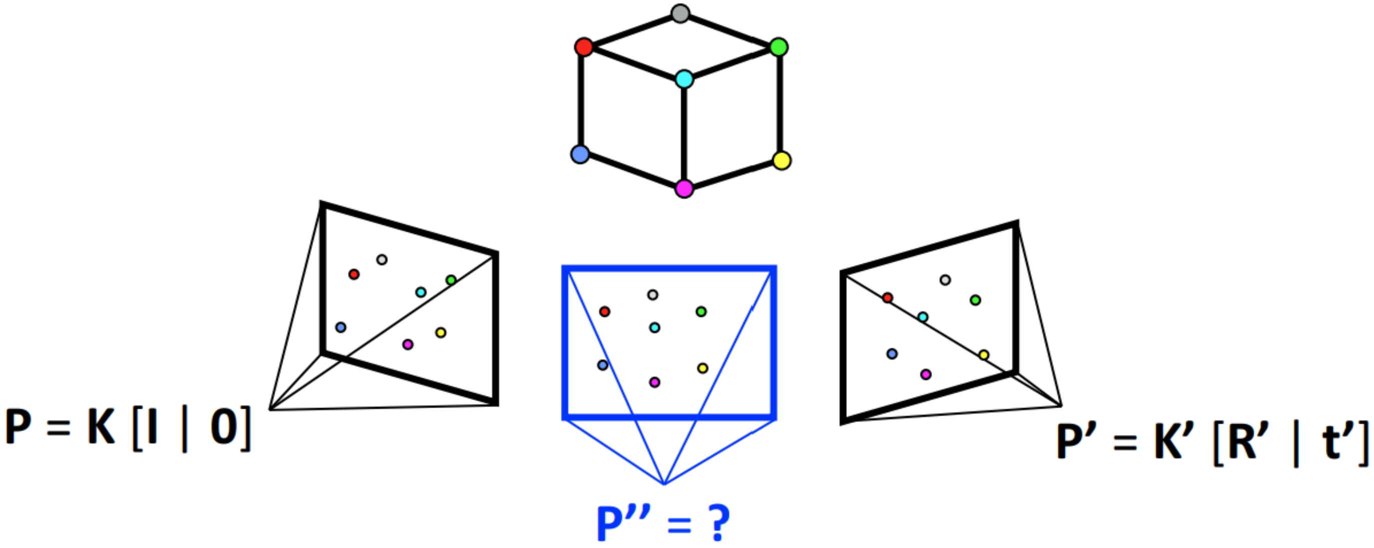


Ceres-Solver, http://ceres-solver.org/

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

Incremental SfM

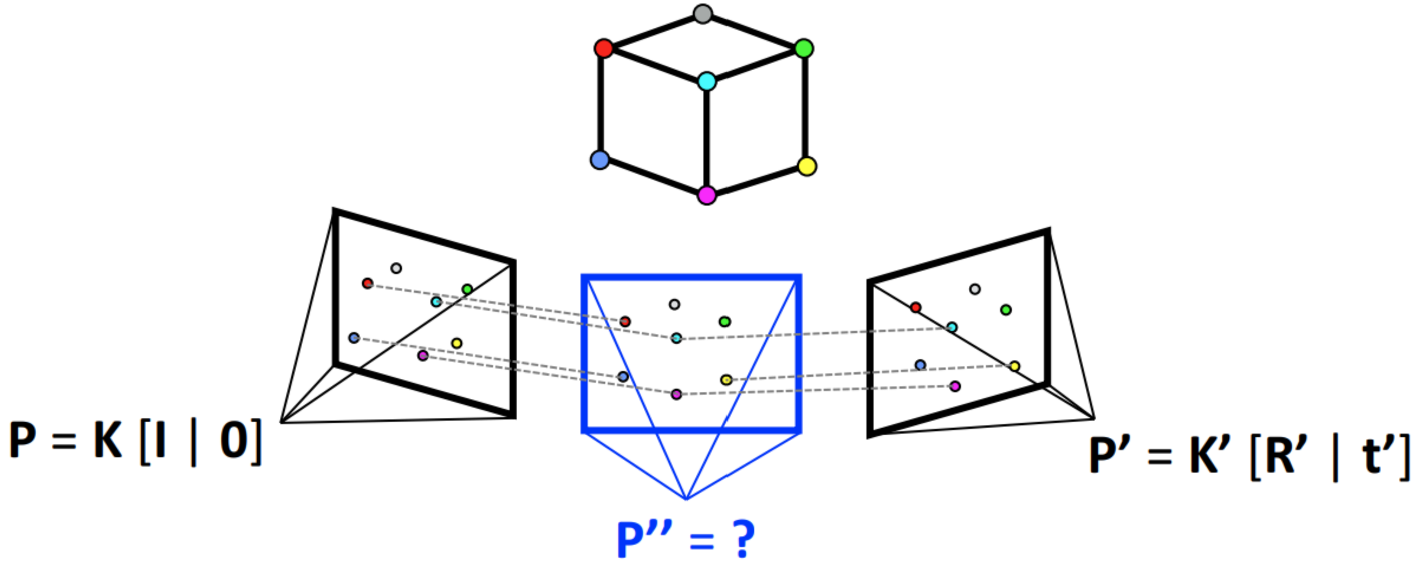
- Absolute camera registration



Slides from Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

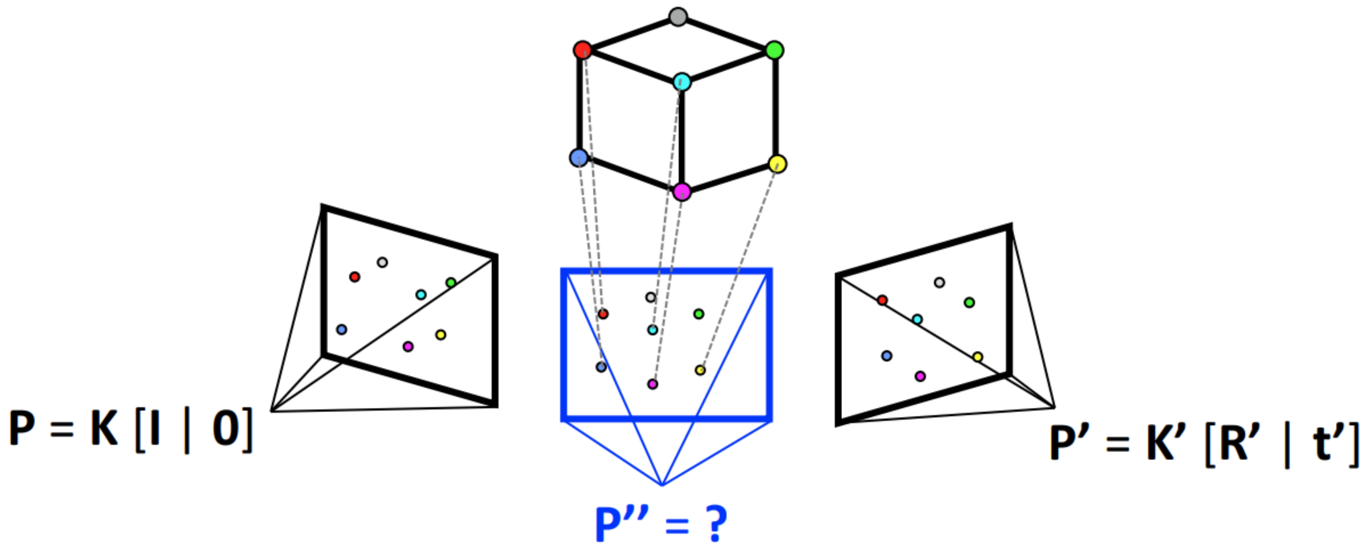
Incremental SfM

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



Incremental SfM

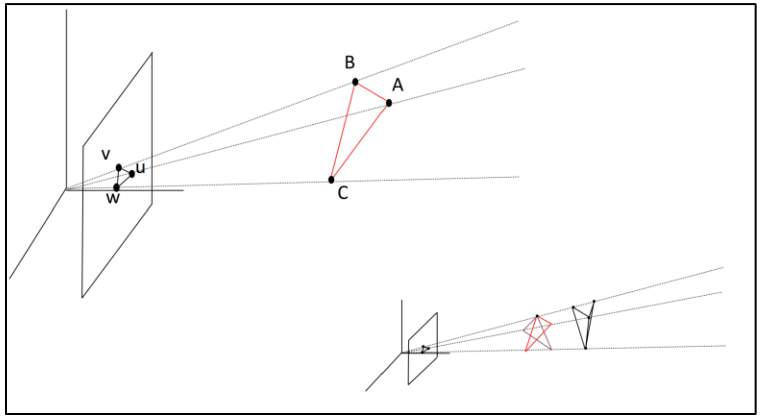
- Absolute camera registration
  1. Find 2D-3D correspondences

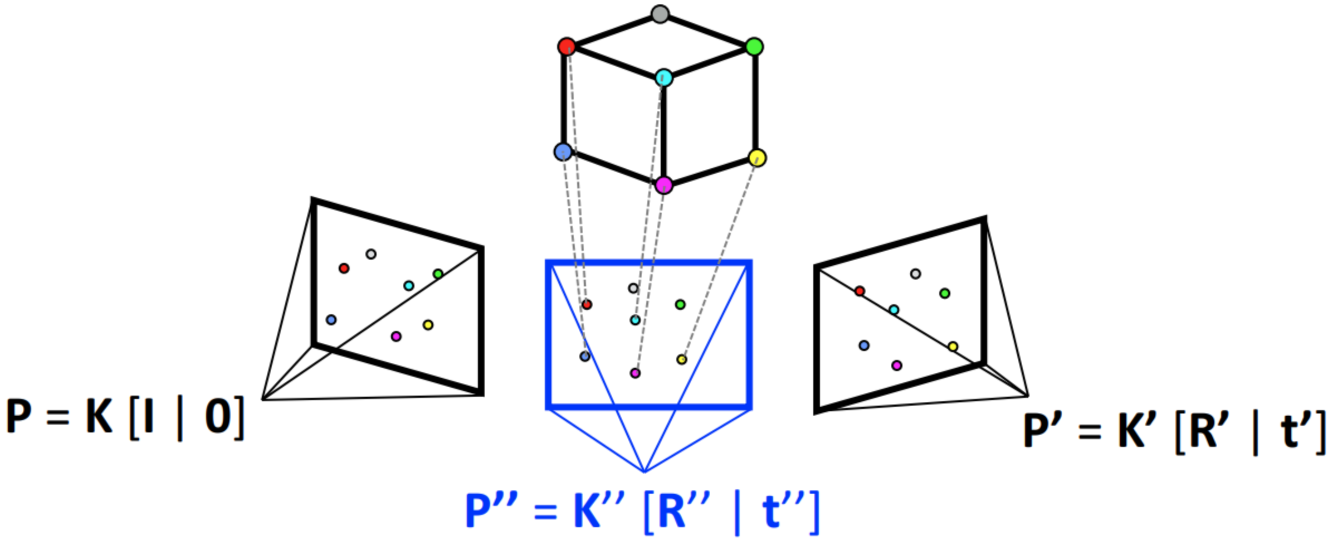


Slides from Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

Incremental SfM

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

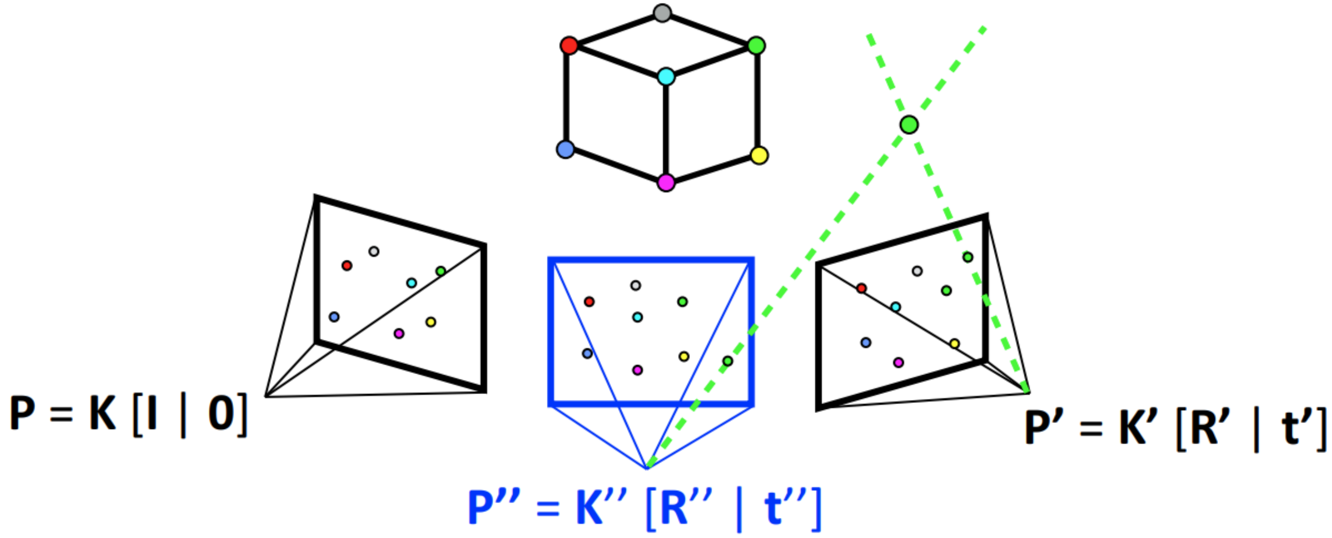




Slides from Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

Incremental SfM

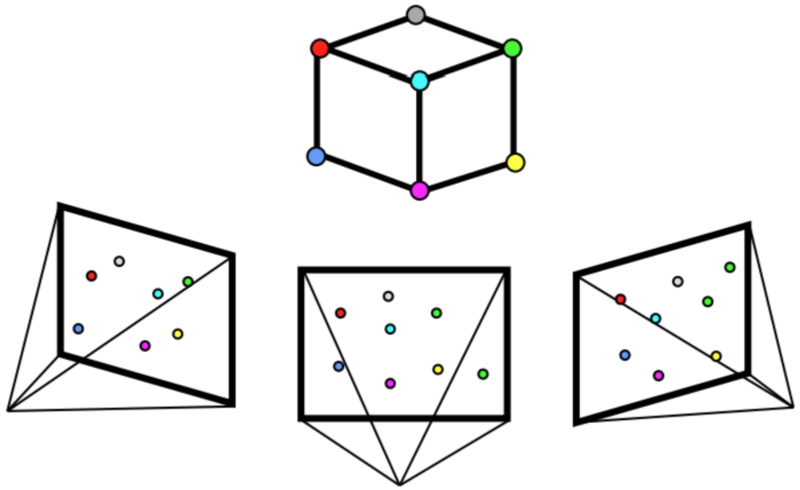
- Absolute camera registration
  1. Find 2D-3D correspondences
  2. Solve Perspective-n-Point problem
  3. Triangulate new points



Slides from Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

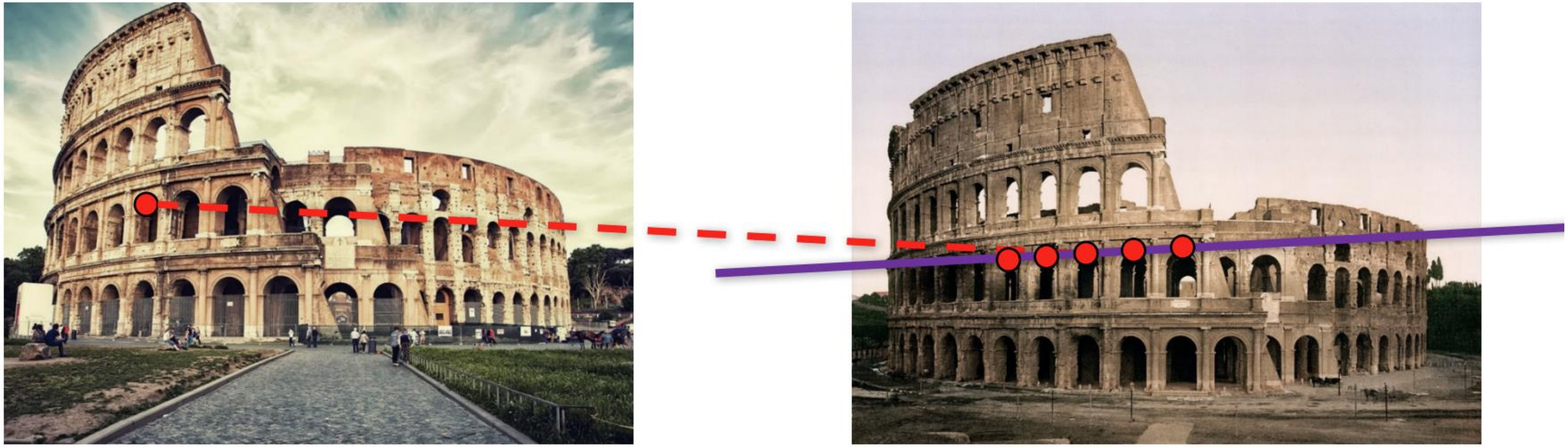
Incremental SfM

- Bundle adjustment $\min_{P,X} \|x - \pi(P, X)\|$



Incremental SfM: Important Details

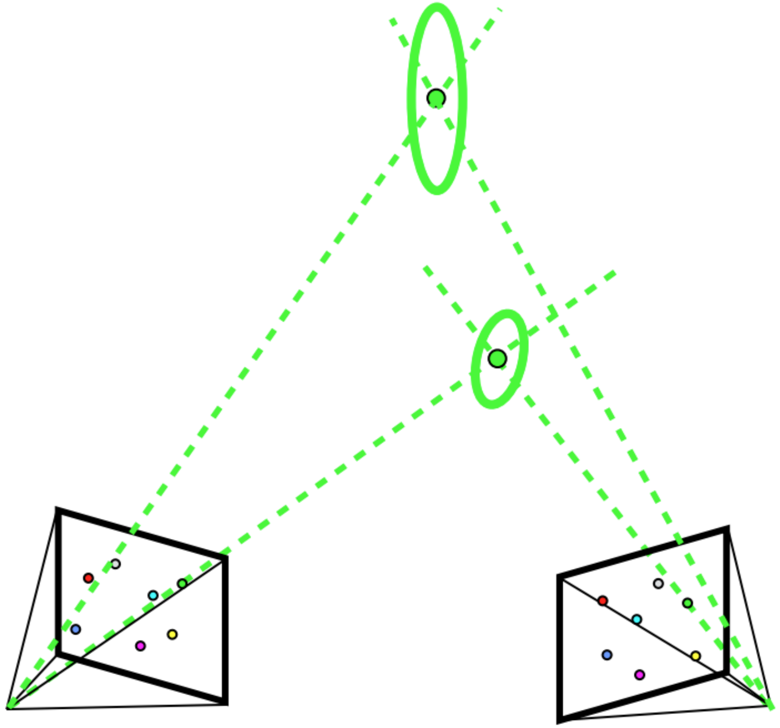
- Outlier filtering
  - Remove points with large reprojection error



Slides from Image from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

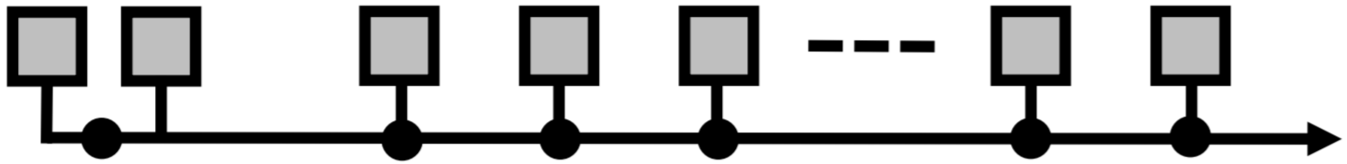
Incremental SfM: Important Details

- Outlier filtering
  - Remove points with large reprojection error
  - Remove points at “infinity”

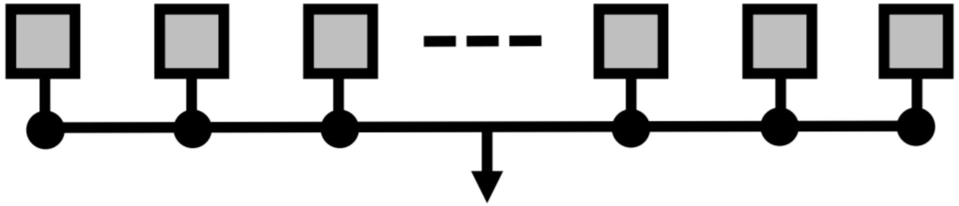


SfM Pipeline – Sparse 3D Reconstruction

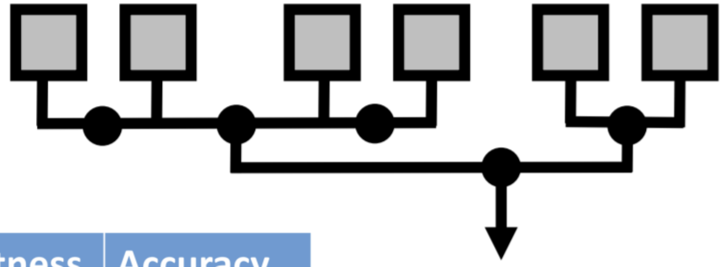
- 3 paradigms
  - Incremental



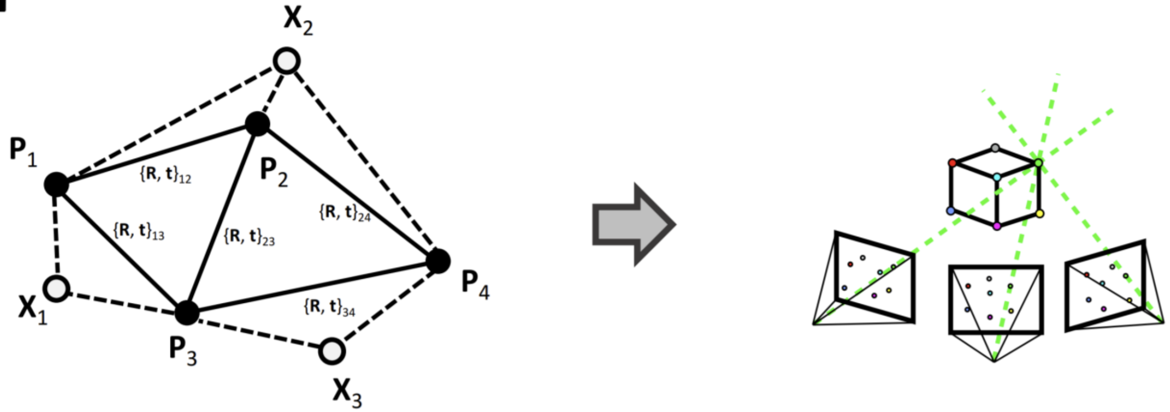
  - Global



  - Hierarchical

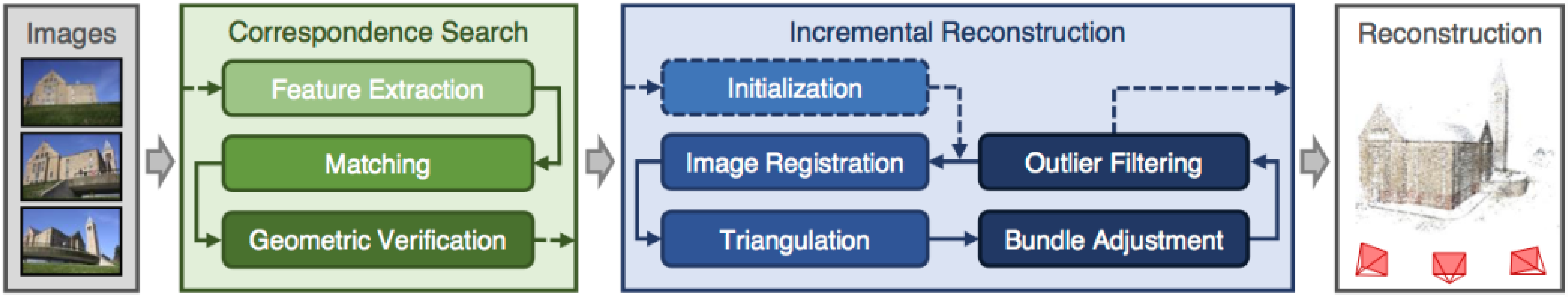


Method	Efficiency	Robustness	Accuracy
Incremental	-	++	+
Global	+	+	+
Hierarchical	++	-	-



Images from: https://demuc.de/tutorials/cvpr2017/sparse-modeling.pdf

SfM Pipeline



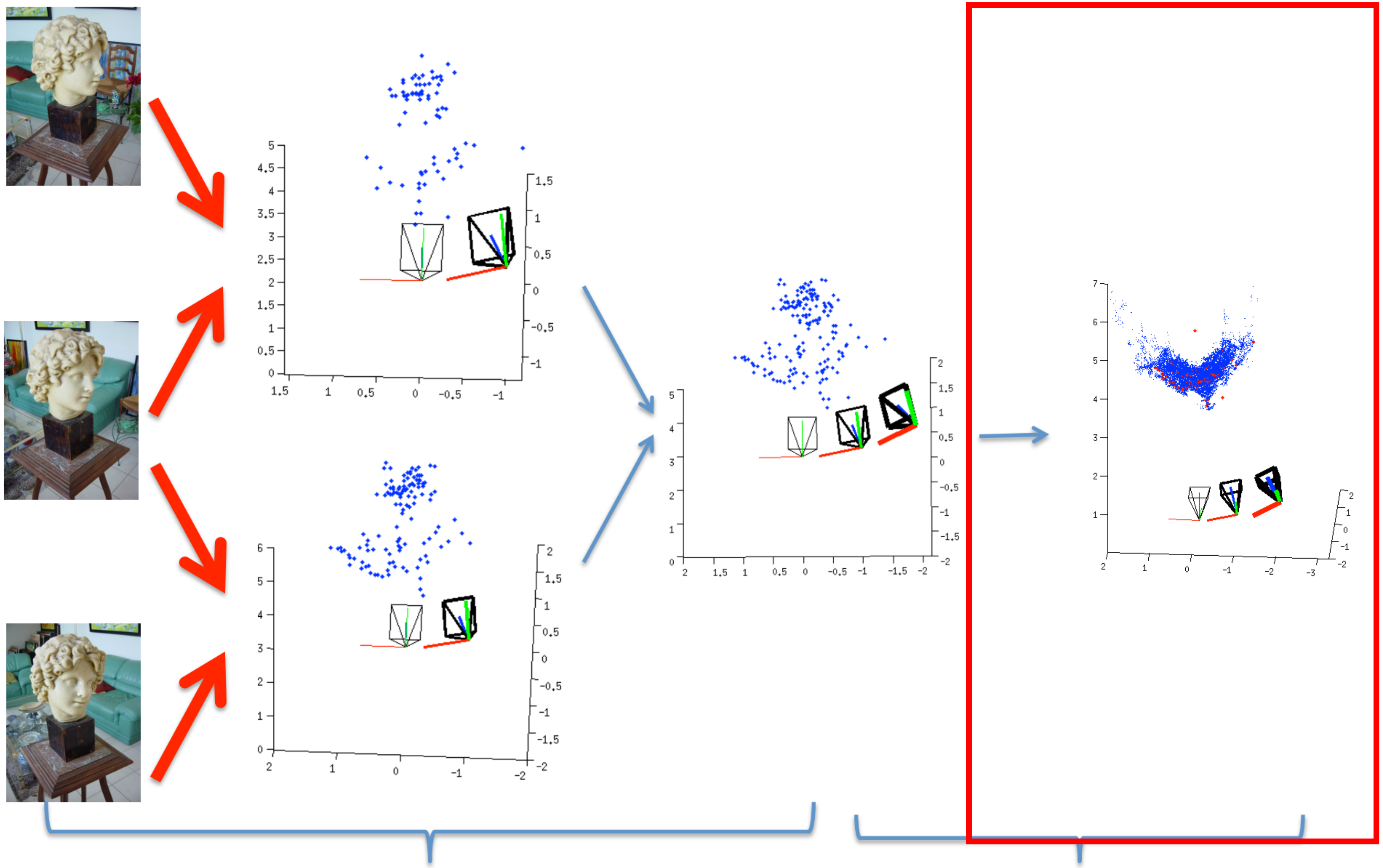
COLMAP's incremental Structure-from-Motion pipeline.

Image from: https://colmap.github.io/tutorial.html

COLMAP Example

COLMAP Reconstruction of Dubrovnik – 58k Images

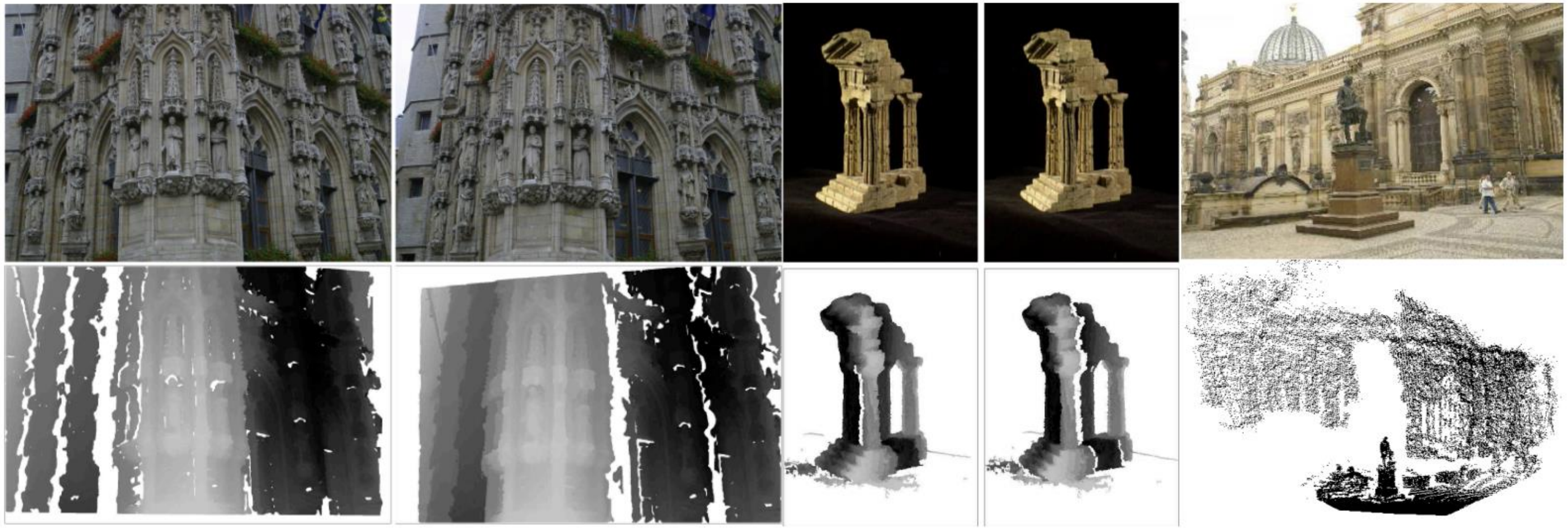
https://grail.cs.washington.edu/rome/



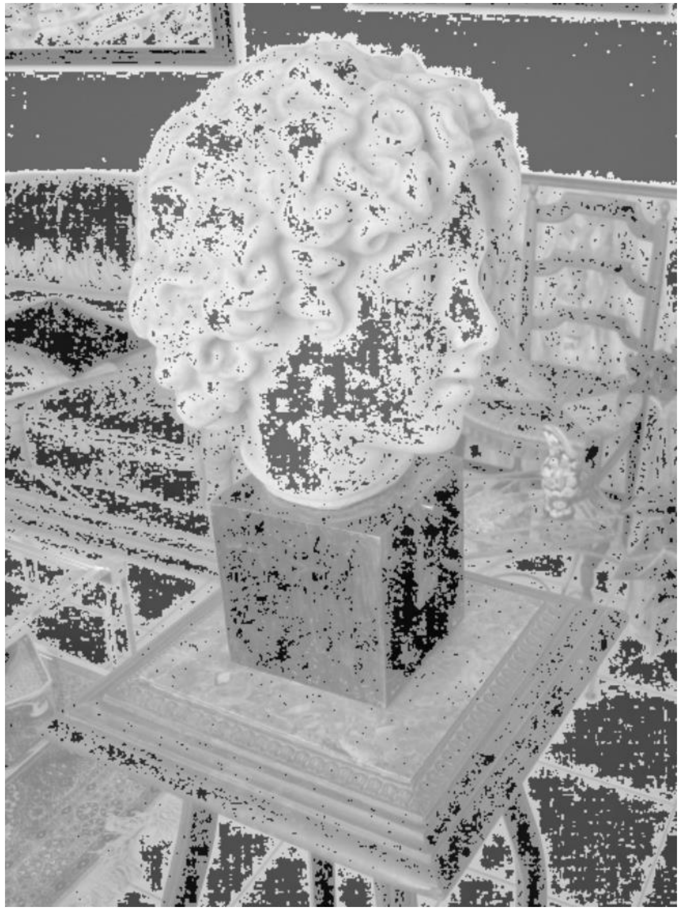
Taught

Next

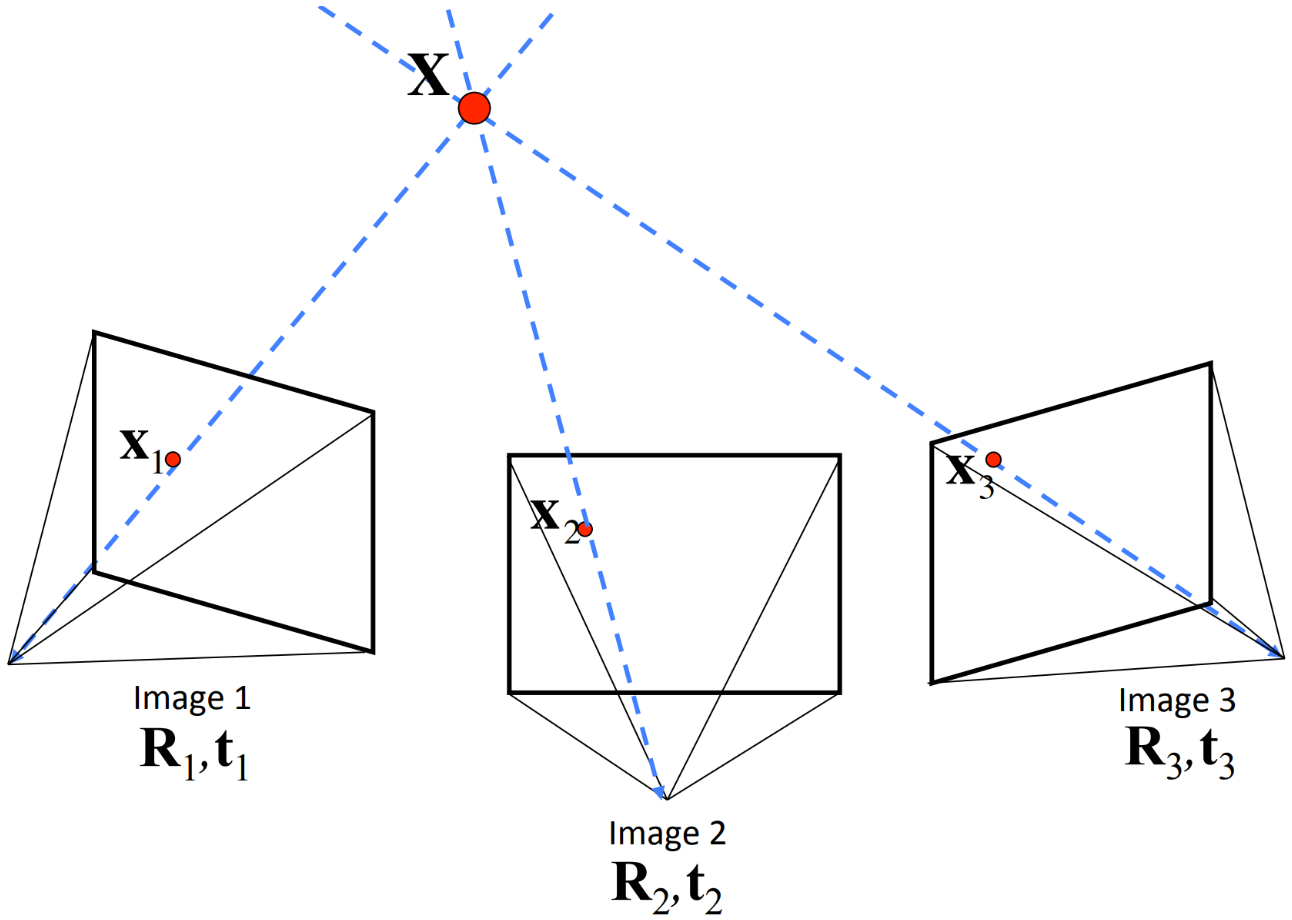
Multi-View Stereo: Matching Propagation



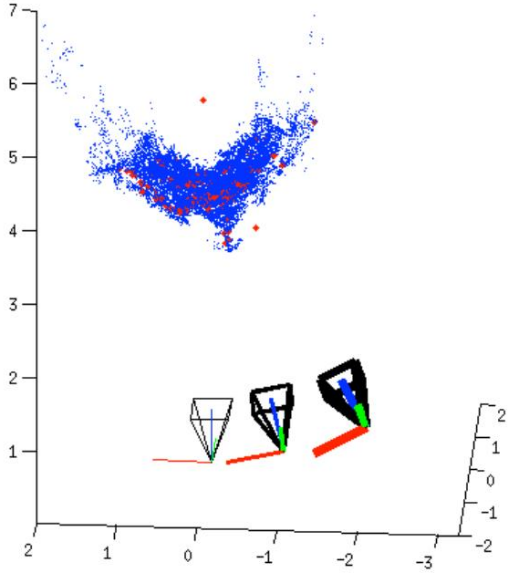
Start With the Seeding Pairs from SfM

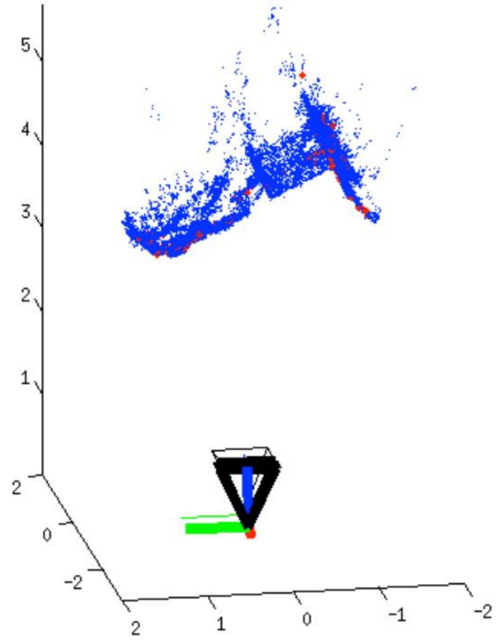


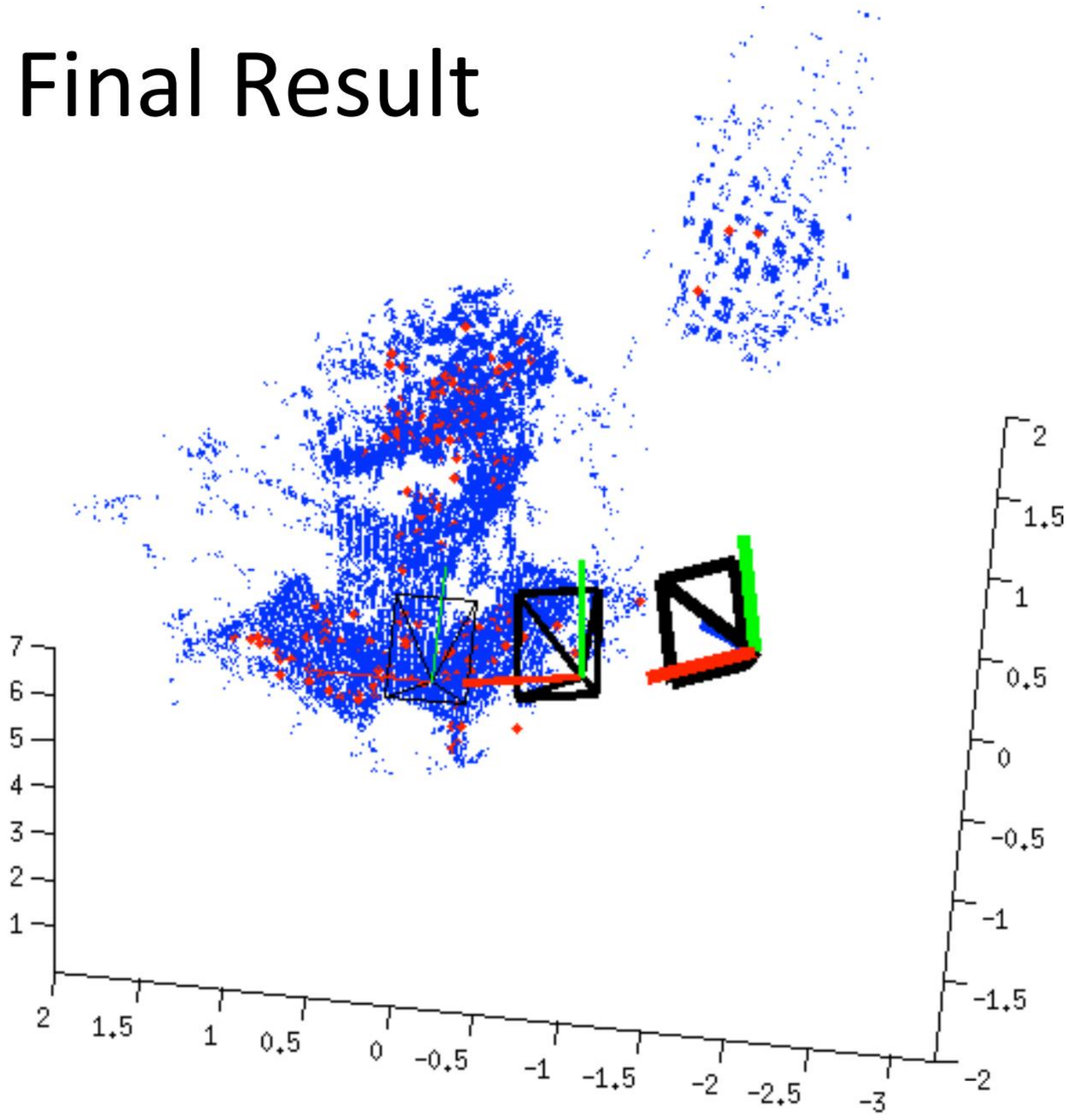
Triangulation from Multiple Views



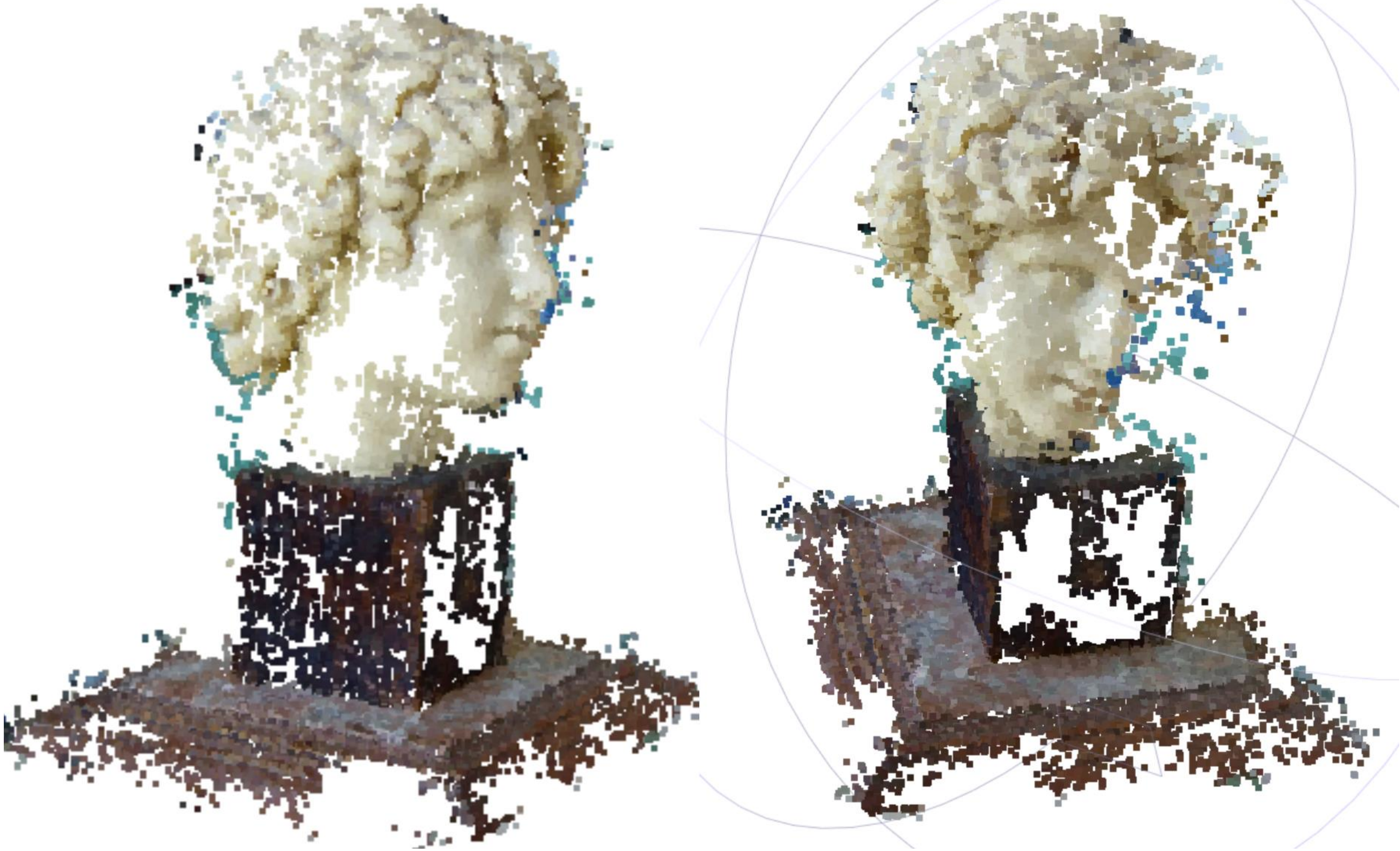
Final Result



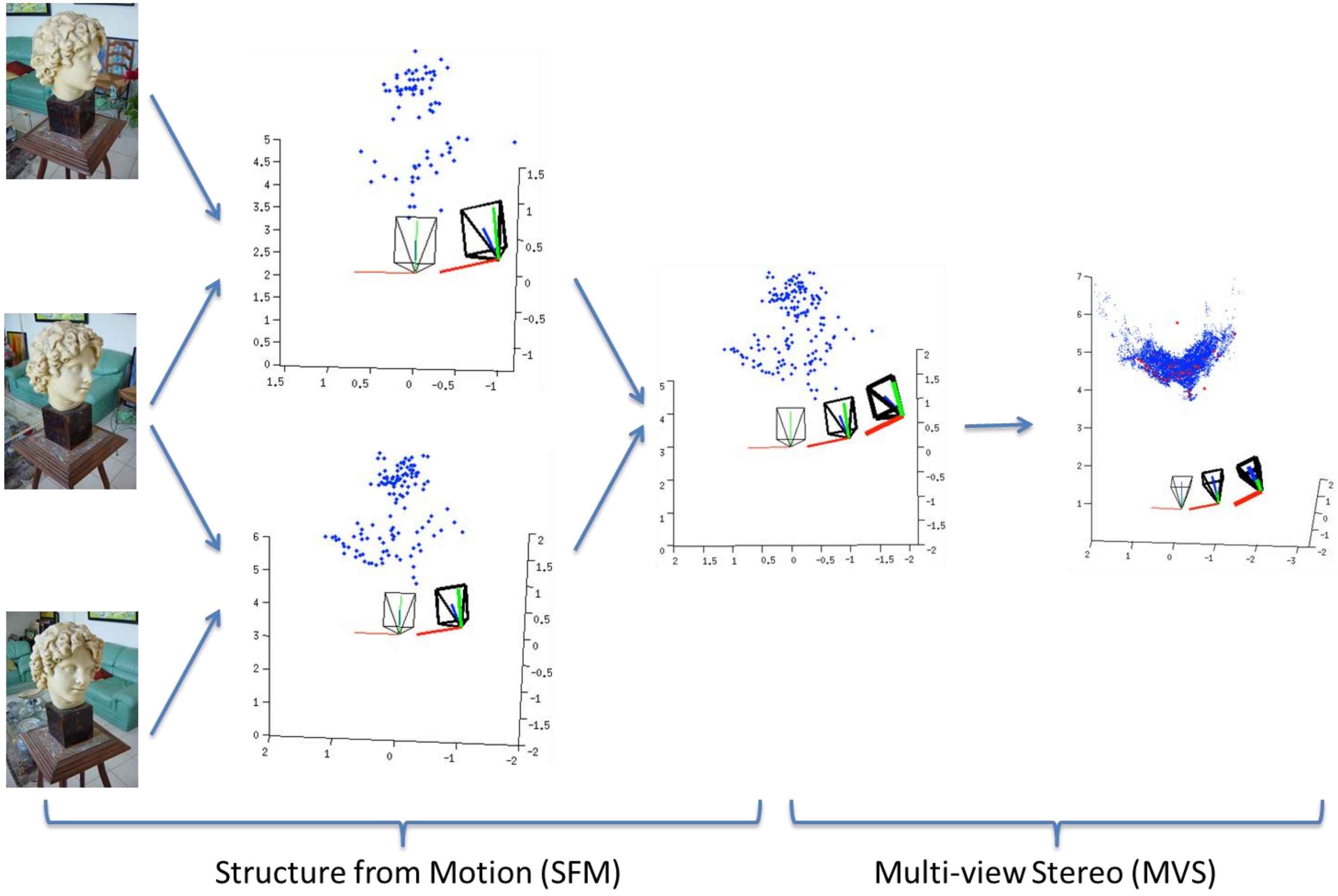




Colorize the Point Cloud



Recap: SfM + MVS

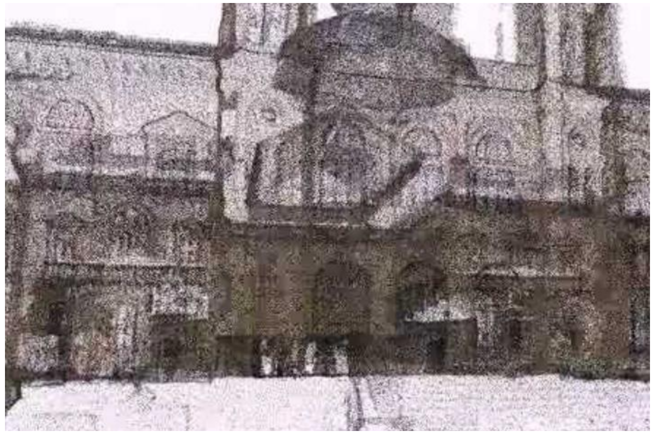


Major Steps of Image-based 3D Modeling

- Images → Sparse Points
  - SfM
- Images → Dense Points
  - Multiple View Stereo

Major Steps of Image-based 3D Modeling

- Images → Sparse Points
  - SfM
- Images → Dense Points
  - Multiple View Stereo
- Points → Meshes
  - Mesh Generation
- Meshes → Textured Meshes
  - Texture Mapping



https://www.youtube.com/watch?v=dXeXtj0PPVI

Recent work with Deep Learning - BA-Net

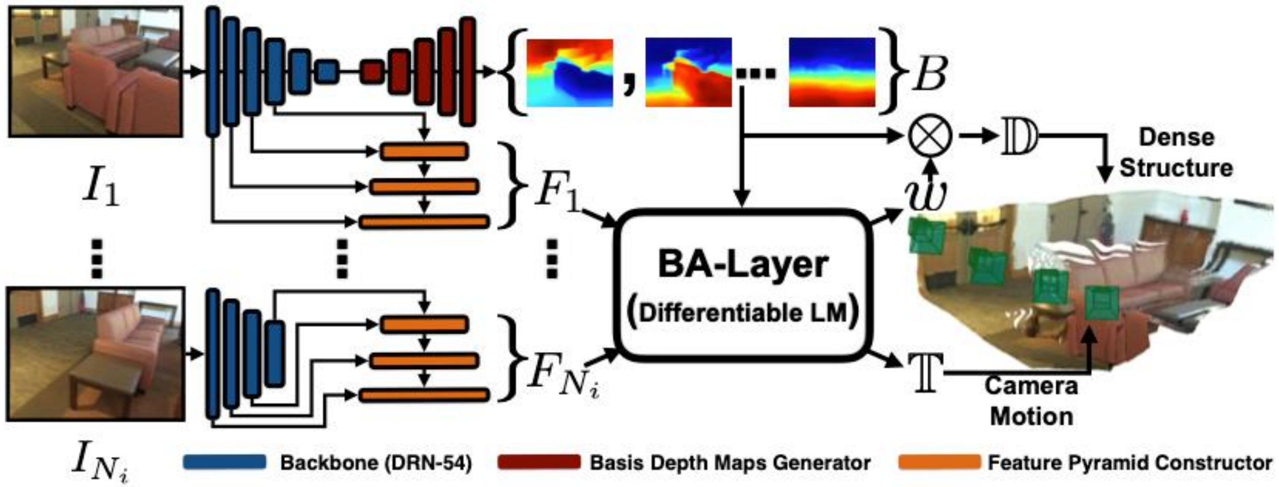


Figure 1: Overview of our BA-Net structure, which consists of a DRN-54 (Yu et al., 2017) as the backbone network, a **Basis Depth Maps Generator** that generates a set of basis depth maps, a **Feature Pyramid Constructor** that constructs multi-scale feature maps, and a **BA-Layer** that optimizes both the depth map and the camera poses through a novel differentiable LM algorithm.

Bundle Adjustment BA

Reprojection Error

Pixel coord

$$e^g_{i,j}(\mathcal{X}) = \pi(\boldsymbol{T}_i, \boldsymbol{p}_j) - \boldsymbol{q}_{i,j}$$

Camera Poses

3D point coord

Objective

$$\mathcal{X} = \operatorname{argmin} \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \|e^g_{i,j}(\mathcal{X})\|$$

Feature-based BA

$$e^f_{i,j}(\mathcal{X}) = F_i(\pi(\boldsymbol{T}_i, d_j \cdot \boldsymbol{q}_j)) - F_1(\boldsymbol{q}_j)$$

Key Idea: Instead of performing Bundle Adjustment (BA) in pixel space, why not do it in feature space? Feature space should be more robust and invariant towards non-semantic changes (e.g. changes in lighting).

1. Pixel space BA + RANSAC still produces outliers, BA-Net tackles this by providing a better conditioned input (features) instead.
2. Uses basis depth map in order reduce search space for their Levenberg-Marquardt (LM) algorithm during optimization of depth map.

BA-net: Dense Bundle Adjustment Networks, Chengzhou Tang et al., 2018

Recent work with Deep Learning - BA-Net

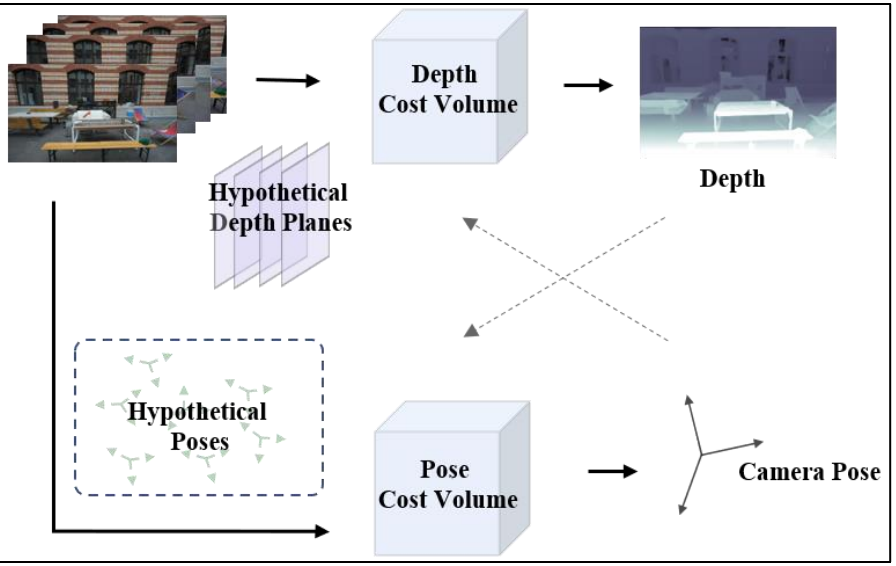
	Ours	Ours*	DeMoN*	Photometric BA	Geometric BA
Rotation (degree)	**1.018**	1.587	3.791	4.409	8.56
Translation (cm)	**3.39**	10.81	15.5	21.40	36.995
Translation (degree)	**20.577**	31.005	31.626	34.36	39.392
abs relative difference	**0.161**	0.238	0.231	0.268	0.382
sqr relative difference	**0.092**	0.176	0.520	0.427	1.163
RMSE (linear)	**0.346**	0.488	0.761	0.788	0.876
RMSE (log)	**0.214**	0.279	0.289	0.330	0.366
RMSE (log, scale inv.)	**0.184**	0.276	0.284	0.323	0.357

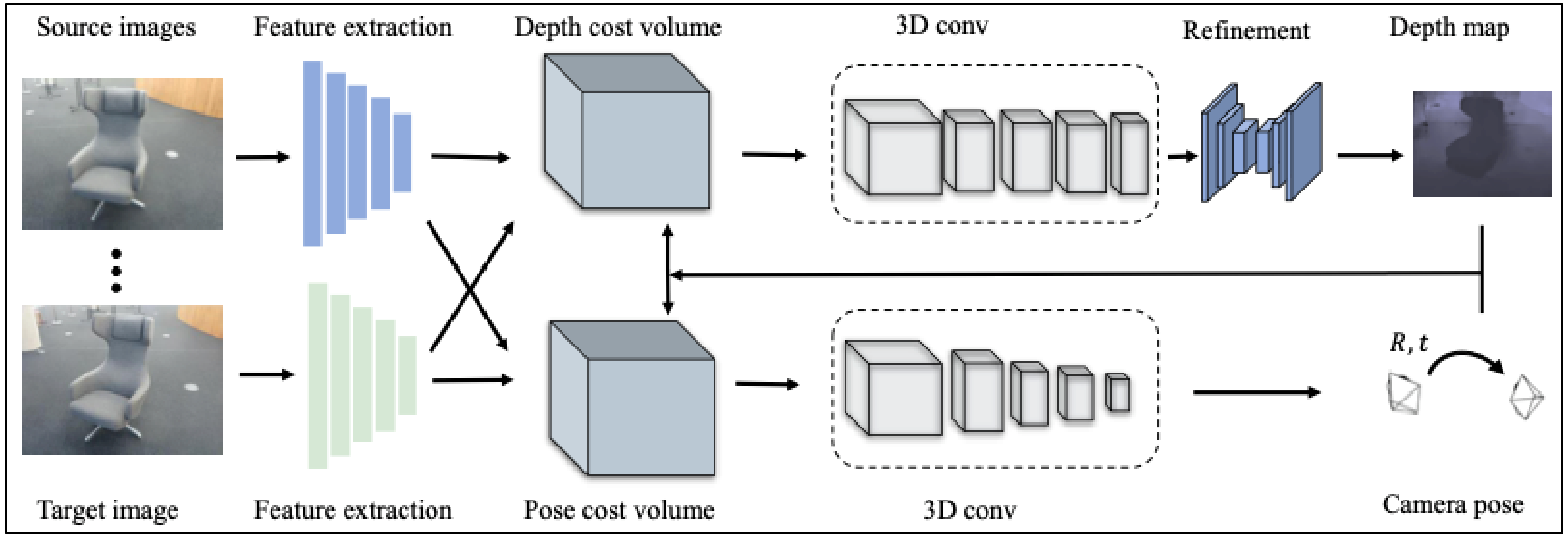
Table 1: Quantitative comparisons with DeMoN and classic BA. The superindex * denotes that the model is trained on the trainning set described in Ummenhofer et al. (2017).

Works pretty well compared to **Photometric BA** and **Geometric BA**!

BA-net: Dense Bundle Adjustment Networks, Chengzhou Tang et al., 2018

Recent work with Deep Learning - DeepSFM





Key Idea: Similar to BA-Net, performs refinement in feature-space. Main difference is the use of Depth + Pose cost volumes instead of a differentiable LM model like BA-Net, which is very compute and memory intensive.

1. Effectively enforces both geometric and photometric consistency with the cost volumes via iterative refinement.
2. To construct cost volumes, they
   1. Warp + reproject depth maps
   2. Warp feature maps
   3. Sample hypothetical camera poses via random translation and rotation
   4. Loss is simply a linear sum of the scaled *rotation*, *translation* and *depth estimation* loss measured via Huber Loss.

DeepSFM: Structure From Motion Via Deep Bundle Adjustment, Xingkui Wei et al., 2020

Recent work with Deep Learning - DeepSFM

Table 1. Results on DeMoN datasets, the best results are noted by **Bold**.

MVS	Depth			Motion		**Scenes11**	Depth			Motion	
Method	L1-inv	sc-inv	L1-rel	Rot	Trans	Method	L1-inv	sc-inv	L1-rel	Rot	Trans
Base-Oracle	0.019	0.197	0.105	0	0	Base-Oracle	0.023	0.618	0.349	0	0
Base-SIFT	0.056	0.309	0.361	21.180	60.516	Base-SIFT	0.051	0.900	1.027	6.179	56.650
Base-FF	0.055	0.308	0.322	4.834	17.252	Base-FF	0.038	0.793	0.776	1.309	19.426
Base-Matlab	-	-	-	10.843	32.736	Base-Matlab	-	-	-	0.917	14.639
DeMoN	0.047	0.202	0.305	5.156	14.447	DeMoN	0.019	0.315	0.248	0.809	8.918
LS-Net	0.051	0.221	0.311	4.653	11.221	LS-Net	0.010	0.410	0.210	4.653	8.210
BANet	0.030	0.150	0.080	3.499	11.238	BANet	0.080	0.210	0.130	3.499	10.370
Ours	**0.021**	**0.129**	**0.079**	**2.824**	**9.881**	Ours	**0.007**	**0.112**	**0.064**	**0.403**	**5.828**

Better than BA-Net and has no rollout limitation as well unlike BA-Net (due to compute constraints from integrated LM module)

Next Week

++ Graph-based SLAM

+ SLAM Formalism (state, observation, error/observation model)

* 1D SLAM example

+ A probabilistic perspective of SLAM

+ Parallel Tracking And Mapping (PTAM)

+ Visual Place Recognition (BoVW, VLAD, NetVLAD)

+ Deep-learning-based SLAM and NeRF

*: know how to code (or how to use tools)
++: know how to derive (more than just the concept)
+: know the concept

References for Next week

- Visual SLAM Tutorial, CVPR'14
- Dellaert, Frank. "Visual SLAM Tutorial: Bundle Adjustment." (2014).
- Grisetti, Giorgio, et al. "A tutorial on graph-based SLAM." IEEE Intelligent Transportation Systems Magazine 2.4 (2010): 31-43.
- Klein, G. and Murray, D., 2007, November. Parallel tracking and mapping for small AR workspaces. In 2007 6th IEEE and ACM international symposium on mixed and augmented reality (pp. 225-234). IEEE.
- https://sites.google.com/view/lsvpr2019/home
- Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R. and Ng, R., 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1), pp.99-106.