

Computer Vision (CMPT 412)

Lecture 9 – Structure from Motion (SfM)

(slides courtesy of Andreas Geiger)

Dr. Andrea Tagliasacchi

Associate Professor

Visual Computing Research Chair

Associate Professor (Status Only)

Staff Research Scientist

Simon Fraser University

Simon Fraser University

University of Toronto

Google DeepMind

Agenda

3.1 Preliminaries

3.2 Two-frame Structure-from-Motion

3.3 Factorization

3.4 Bundle Adjustment

3.1

Preliminaries

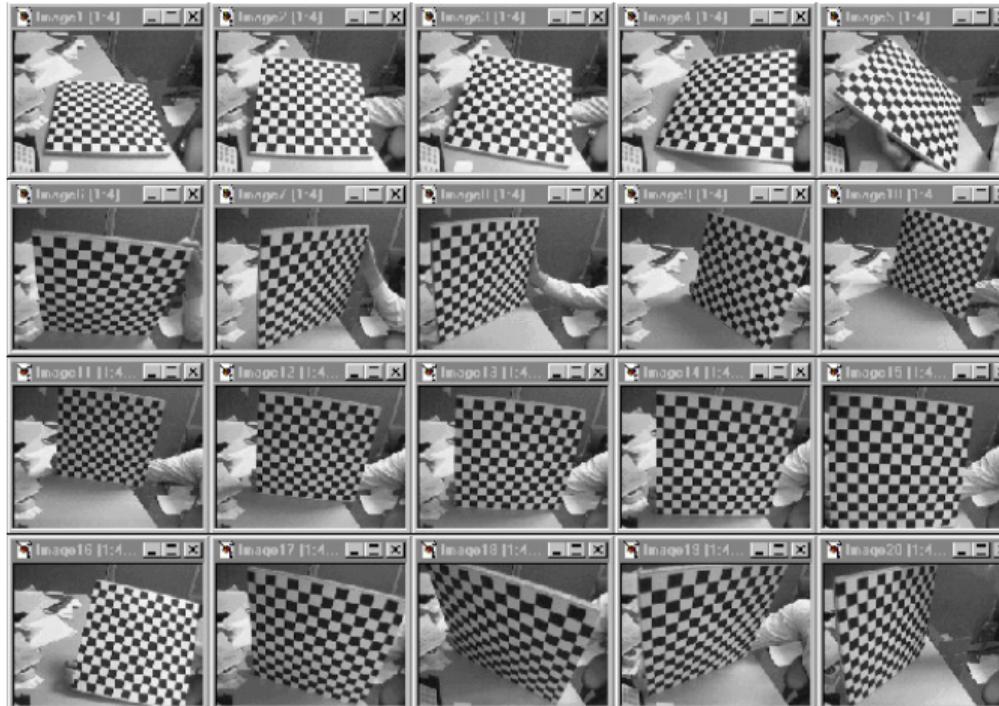
Camera Calibration

Camera Calibration



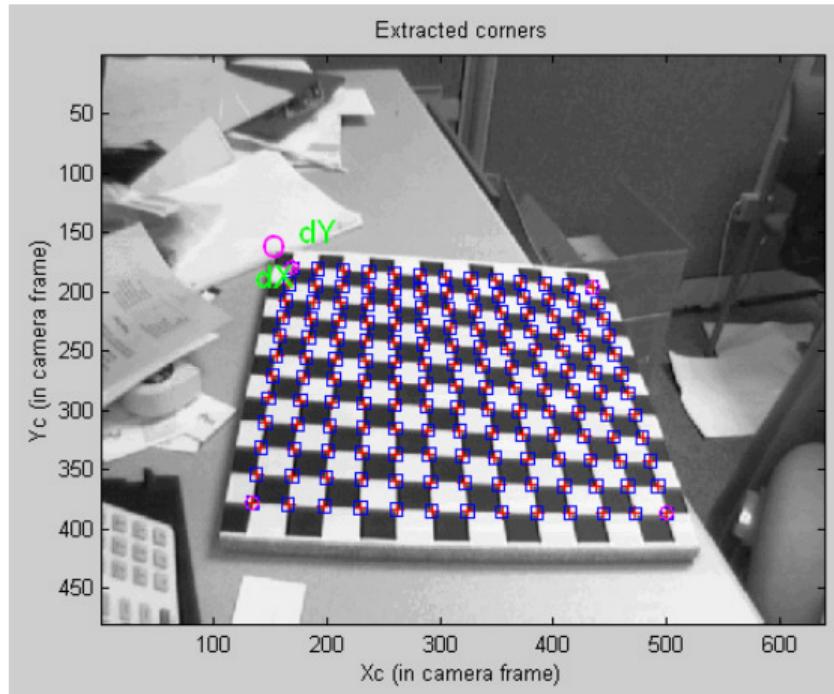
- ▶ **Camera calibration** is the process of finding the **intrinsic/extrinsic parameters**
- ▶ Most commonly, a **known calibration target** (image, checkerboard) is used

Camera Calibration



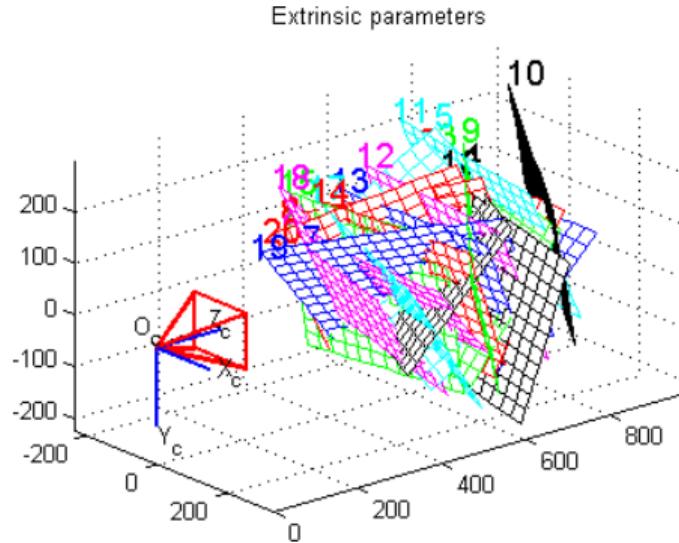
- ▶ First, the known calibration target is **captured in different poses**

Camera Calibration



- Second, **features** (e.g., corners) on the target are detected in the images

Camera Calibration



Finally, the **camera intrinsics and extrinsics** (=poses) are **jointly optimized**:

- ▶ **Closed-form solution** initializes all parameters except for distortion parameters
- ▶ **Non-linear optimization** of all parameters by minimizing reprojection errors

Camera Calibration

Remarks:

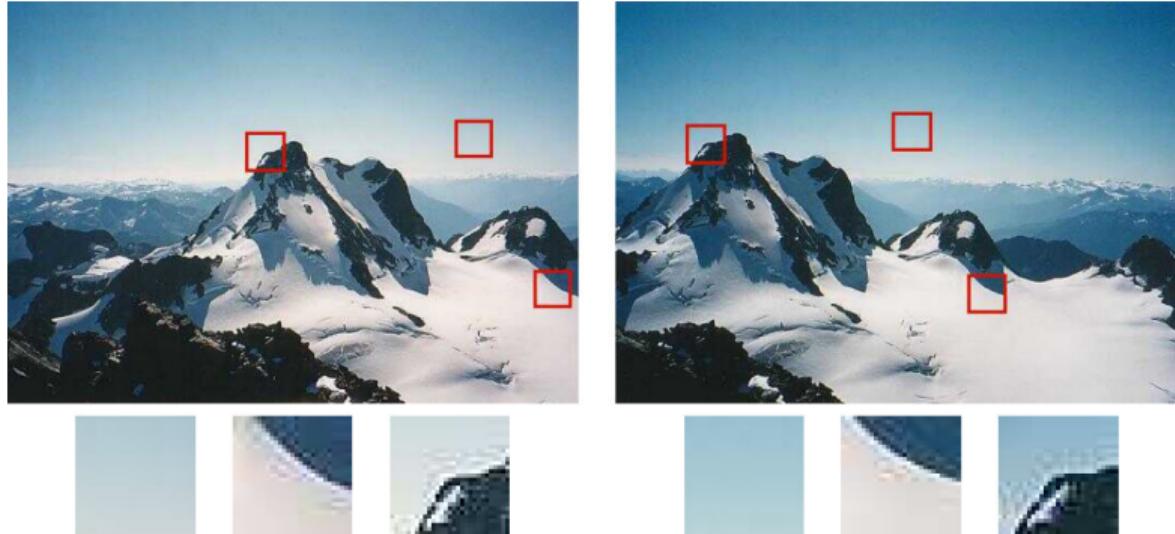
- ▶ There exists a variety of calibration techniques that are used in different settings
- ▶ These methods differ algorithmically, but also in the type of assumptions and calibration targets they use: 2D/3D targets, planes, vanishing points, etc.

Further Readings:

- ▶ http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
- ▶ https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html
- ▶ See also: Szeliski Book, Chapter 11.1

Feature Detection and Description

Point Features



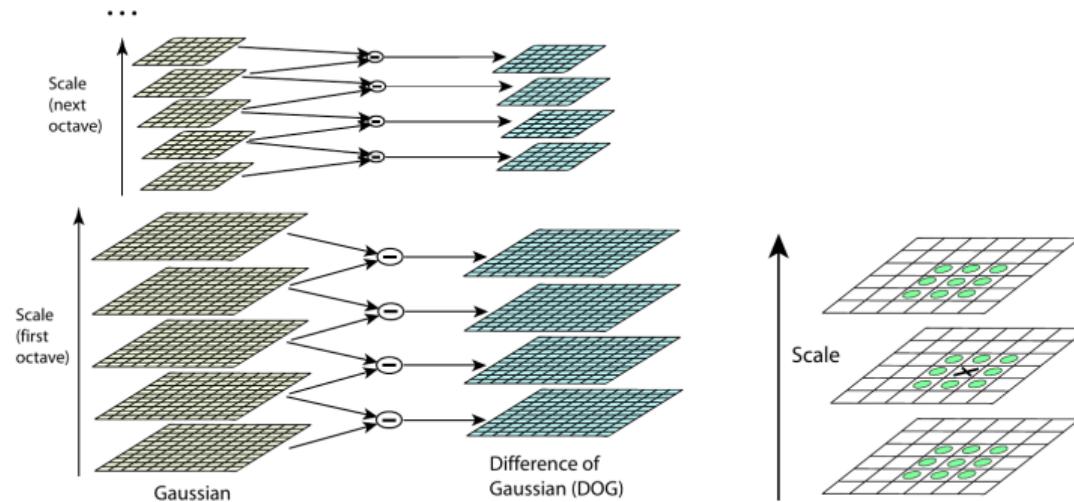
- ▶ **Point features** describe the appearance of **local, salient regions** in an image
- ▶ They can be used to **describe and match images** taken from different viewpoints
- ▶ They form the **basis of sparse 3D reconstruction methods** covered in this lecture

Point Features



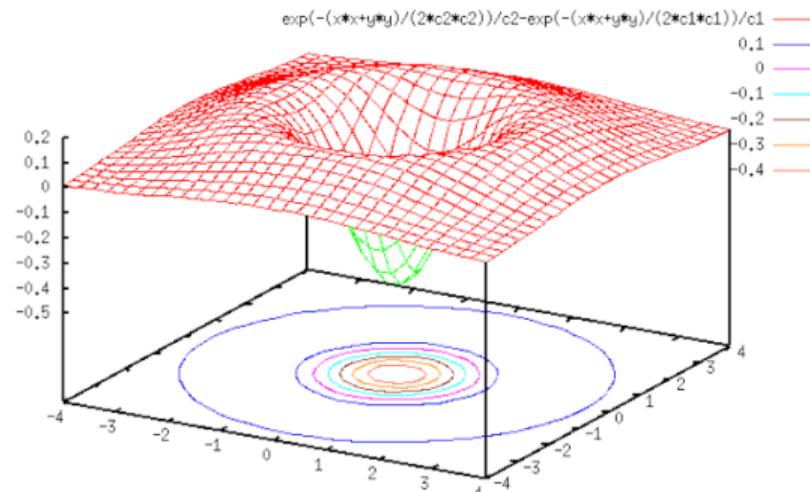
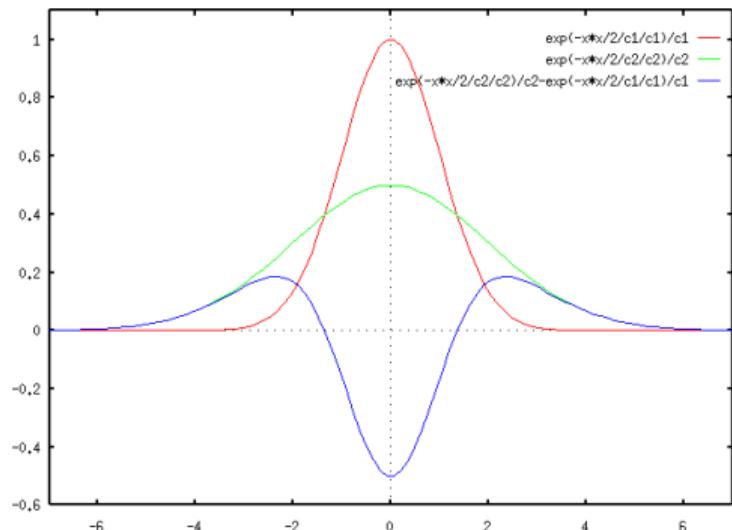
- ▶ Features should be **invariant** to **perspective** effects and **illumination**
- ▶ The same point should have **similar vectors** independent of pose/viewpoint
- ▶ Plain RGB/intensity patches will not have this property, we need something better

Scale Invariant Feature Transform (SIFT)



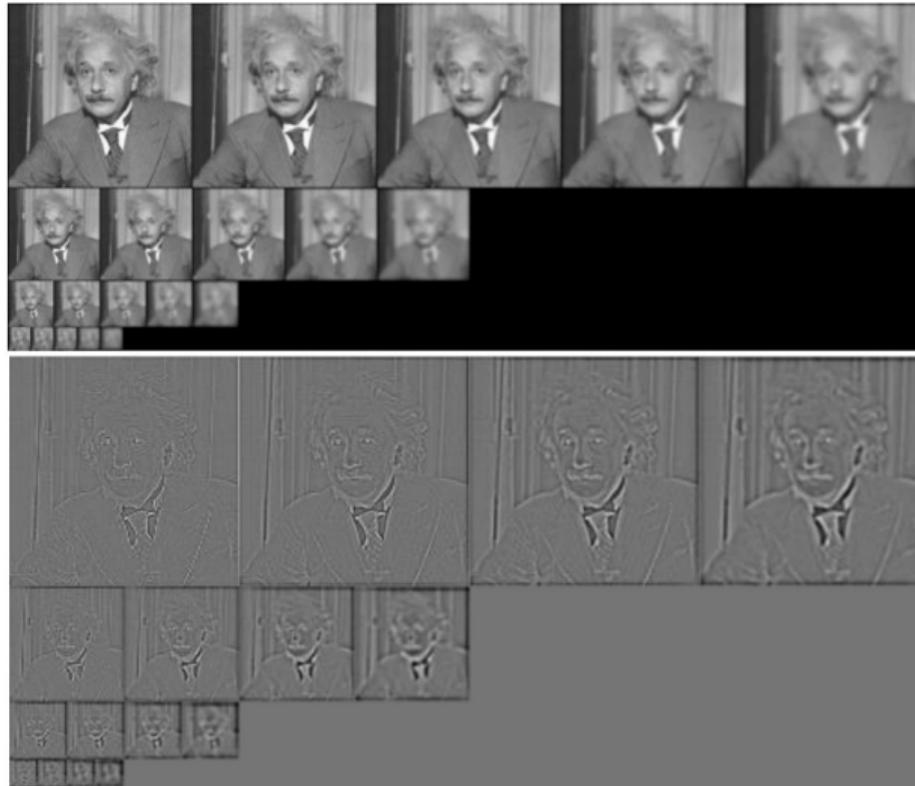
- ▶ SIFT constructs a **scale space** by iteratively filtering the image with a Gaussian
- ▶ Adjacent scales are subtracted, yielding **Difference of Gaussian (DoG)** images
- ▶ **Interest points** (=blobs) are detected as **extrema** in the resulting scale space

Scale Invariant Feature Transform (SIFT)

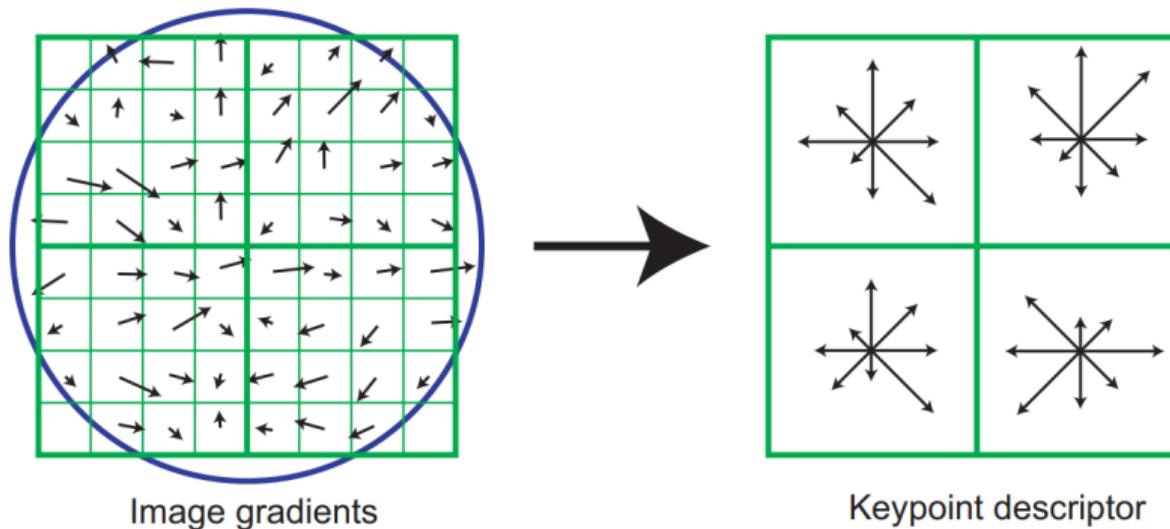


- ▶ SIFT constructs a **scale space** by iteratively filtering the image with a Gaussian
- ▶ Adjacent scales are subtracted, yielding **Difference of Gaussian (DoG)** images
- ▶ **Interest points** (=blobs) are detected as **extrema** in the resulting scale space

Scale Invariant Feature Transform (SIFT)



Scale Invariant Feature Transform (SIFT)

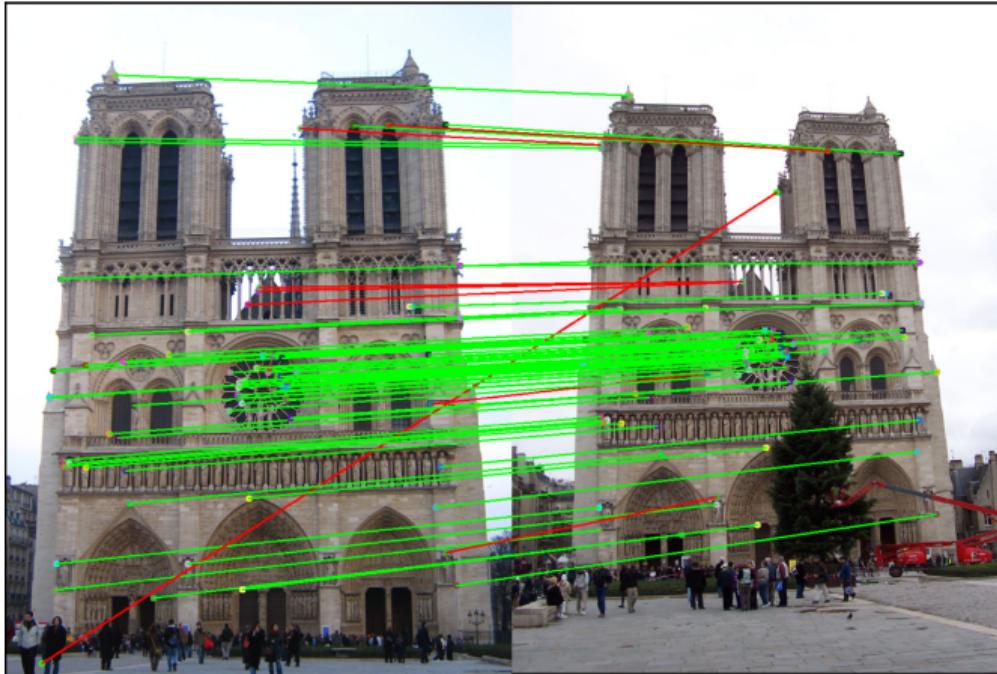


- ▶ SIFT **rotates** the descriptor to **align with the dominant gradient orientation**
- ▶ **Gradient histograms** are computed for local sub-regions of the descriptor
- ▶ All histograms are concatenated and normalized to form a **128D feature vector**

Feature Detection and Description

- ▶ Many algorithms for feature detection and description have been developed, e.g., SIFT, SURF, U-SURF, BRISK, ORB, FAST, and recently deep learning based ones
- ▶ SIFT was a **seminal** work due to its invariance and robustness which revolutionized recognition and in particular matching and enabled the development of large-scale SfM techniques which we discuss in this lecture
- ▶ Despite >20 years old, SIFT is **still used today** (e.g., in the SfM pipeline COLMAP)
- ▶ Feature correspondences can be retrieved with **efficient nearest neighbor search**
- ▶ **Ambiguous matches** are typically **filtered** by computing the ratio of distance from the closest neighbor to the distance of the second closest
- ▶ A large ratio (>0.8) indicates that the found match might not be the correct one

Scale Invariant Feature Transform (SIFT)



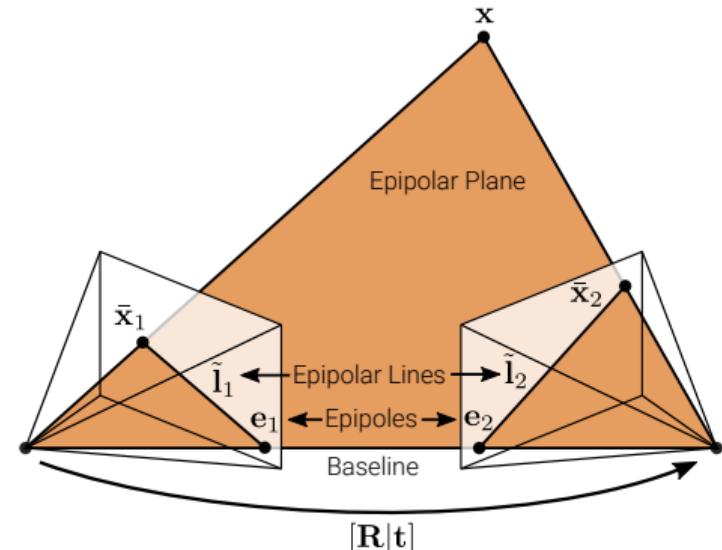
- ▶ Example of two images with SIFT correspondences (green: correct, red: wrong)

Epipolar Geometry

Epipolar Geometry

Goal: Recovery of camera pose (and 3D structure) from image correspondences.
The required relationships are described by the two-view **epipolar geometry**.

- ▶ Let \mathbf{R} and \mathbf{t} denote the relative pose between **two perspective cameras**
- ▶ A 3D point \mathbf{x} is projected to pixel $\bar{\mathbf{x}}_1$ in image 1 and to pixel $\bar{\mathbf{x}}_2$ in image 2
- ▶ The 3D point \mathbf{x} and the two camera centers span the **epipolar plane**
- ▶ The correspondence of pixel $\bar{\mathbf{x}}_1$ in image 2 must lie on the **epipolar line** $\tilde{\mathbf{l}}_2$ in image 2
- ▶ All epipolar lines pass through the **epipole**



Epipolar Geometry

Let $\mathbf{K}_i \in \mathbb{R}^{3 \times 3}$ denote the **camera matrix** of camera i .

Let $\tilde{\mathbf{x}}_i = \mathbf{K}_i^{-1} \bar{\mathbf{x}}_i$ denote the **local ray direction** of pixel $\bar{\mathbf{x}}_i$ in camera i . We have:

$$\tilde{\mathbf{x}}_2 \propto \mathbf{x}_2 = \mathbf{R} \mathbf{x}_1 + \mathbf{t} \propto \mathbf{R} \tilde{\mathbf{x}}_1 + s \mathbf{t}$$

Taking the **cross product** of both sides with \mathbf{t} we obtain:

$$[\mathbf{t}]_{\times} \mathbf{R} \tilde{\mathbf{x}}_1 \propto [\mathbf{t}]_{\times} \tilde{\mathbf{x}}_2$$

Taking the **dot product** of both sides with $\tilde{\mathbf{x}}_2^T$ yields (triple product):

$$\tilde{\mathbf{x}}_2^T [\mathbf{t}]_{\times} \mathbf{R} \tilde{\mathbf{x}}_1 \propto \tilde{\mathbf{x}}_2^T [\mathbf{t}]_{\times} \tilde{\mathbf{x}}_2 = 0 \quad \Rightarrow \quad \tilde{\mathbf{x}}_2^T [\mathbf{t}]_{\times} \mathbf{R} \tilde{\mathbf{x}}_1 = 0$$

Remark: The symbol \propto denotes proportionality.

Remark: Triple product is zero whenever two vectors are the same (volume of trapezoid)

Epipolar Geometry

We arrive at the **epipolar constraint**

$$\tilde{\mathbf{x}}_2^\top \tilde{\mathbf{E}} \tilde{\mathbf{x}}_1 = 0$$

with **essential matrix**:

$$\tilde{\mathbf{E}} = [\mathbf{t}]_\times \mathbf{R}$$

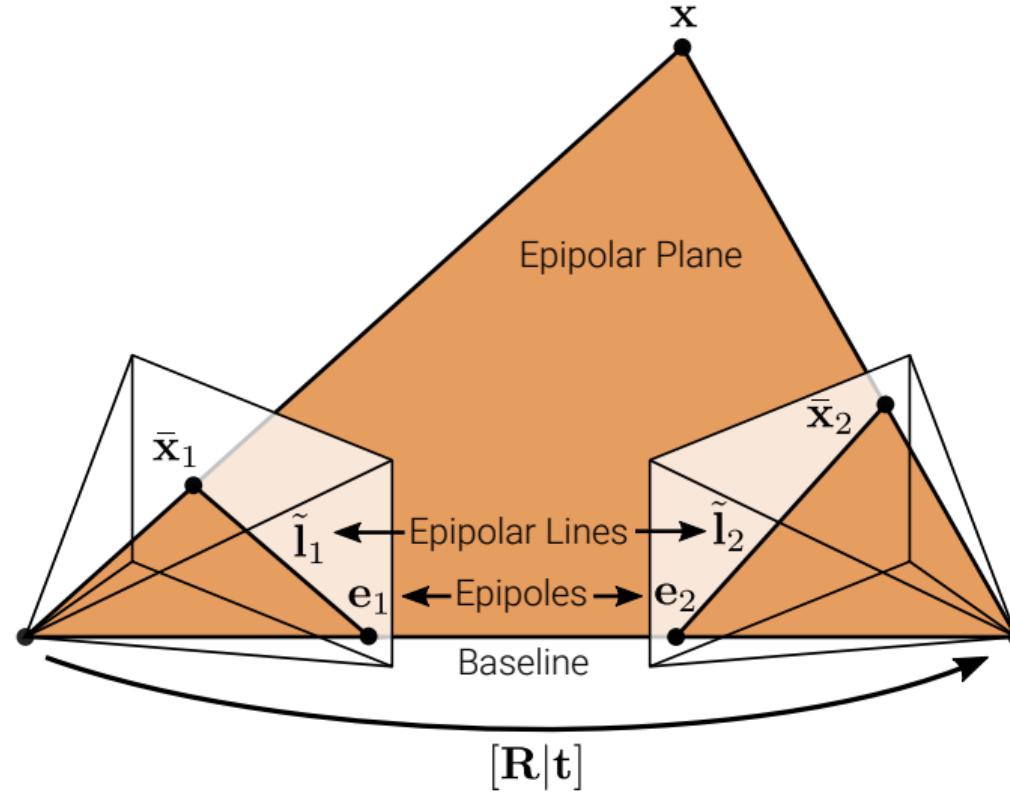
$\tilde{\mathbf{E}}$ maps $\tilde{\mathbf{x}}_1$ in image 1 to the corresponding **epipolar line in image 2** (recall $\tilde{\mathbf{x}}_2^\top \tilde{\mathbf{l}}_2 = 0$)

$$\tilde{\mathbf{l}}_2 = \tilde{\mathbf{E}} \tilde{\mathbf{x}}_1$$

Similarly, by transposition, we obtain the **epipolar line in image 1**:

$$\tilde{\mathbf{l}}_1 = \tilde{\mathbf{E}}^\top \tilde{\mathbf{x}}_2$$

Epipolar Geometry



Epipolar Geometry

For any point $\tilde{\mathbf{x}}_1$ in the first image, the corresponding **epipolar line** $\tilde{\mathbf{l}}_2 = \tilde{\mathbf{E}} \tilde{\mathbf{x}}_1$ in the second image passes through the so-called **epipole** $\tilde{\mathbf{e}}_2$ which therefore satisfies

$$\tilde{\mathbf{e}}_2^\top \tilde{\mathbf{l}}_2 = \tilde{\mathbf{e}}_2^\top \tilde{\mathbf{E}} \tilde{\mathbf{x}}_1 = 0 \quad \forall \tilde{\mathbf{x}}_1.$$

This allows us to compute the position of the epipole, as from the above we have:

$$\tilde{\mathbf{e}}_2^\top \tilde{\mathbf{E}} = \mathbf{0}$$

Thus, $\tilde{\mathbf{e}}_2^\top$ is the **left null-space** (left singular vector with singular value 0) of $\tilde{\mathbf{E}}$. Further:

$$\tilde{\mathbf{E}} \tilde{\mathbf{e}}_1 = \mathbf{0}$$

Thus, $\tilde{\mathbf{e}}_1^\top$ is the **right null-space** (right singular vector with singular value 0) of $\tilde{\mathbf{E}}$.

Estimating the Epipolar Geometry

We can **recover the essential matrix $\tilde{\mathbf{E}}$** from N **image correspondences** forming N homogeneous equations in the nine elements of $\tilde{\mathbf{E}}$ (note: linear wrt. the terms in $\tilde{\mathbf{E}}$):

$$\tilde{\mathbf{x}}_2^\top \tilde{\mathbf{E}} \tilde{\mathbf{x}}_1 = 0$$

As $\tilde{\mathbf{E}}$ is homogeneous we use **singular value decomposition** to constrain the scale.

Note: some terms are products of two image measurements and hence amplify measurement noise asymmetrically. Thus, the **normalized 8-point algorithm** whitens the observations to have zero-mean and unit variance before the calculation and back-transforms the matrix recovered by SVD accordingly.

Estimating the Epipolar Geometry

From $\tilde{\mathbf{E}}$, we can recover the **direction** $\hat{\mathbf{t}}$ of the **translation** vector \mathbf{t} (why only the direction? because problem is only defined up to scale). We have:

$$\hat{\mathbf{t}}^\top \tilde{\mathbf{E}} = \hat{\mathbf{t}}^\top [\mathbf{t}]_\times \mathbf{R} = \mathbf{0}$$

Thus, $\tilde{\mathbf{E}}$ is singular and we obtain $\hat{\mathbf{t}}$ as the left singular vector associated with singular value 0. In practice the singular value will not be exactly 0 due to measurement noise, and we choose the smallest one. The other two singular values are roughly equal.

The rotation matrix \mathbf{R} can also be calculated, see Szeliski, Section 11.3 (p. 683).

Remark: The essential matrix has 5 DoF (3 for rotation \mathbf{R} , 2 for translation direction $\hat{\mathbf{t}}$).

Estimating the Epipolar Geometry with unknown Intrinsics

If the camera calibration \mathbf{K}_i is unknown, we cannot use the local ray directions

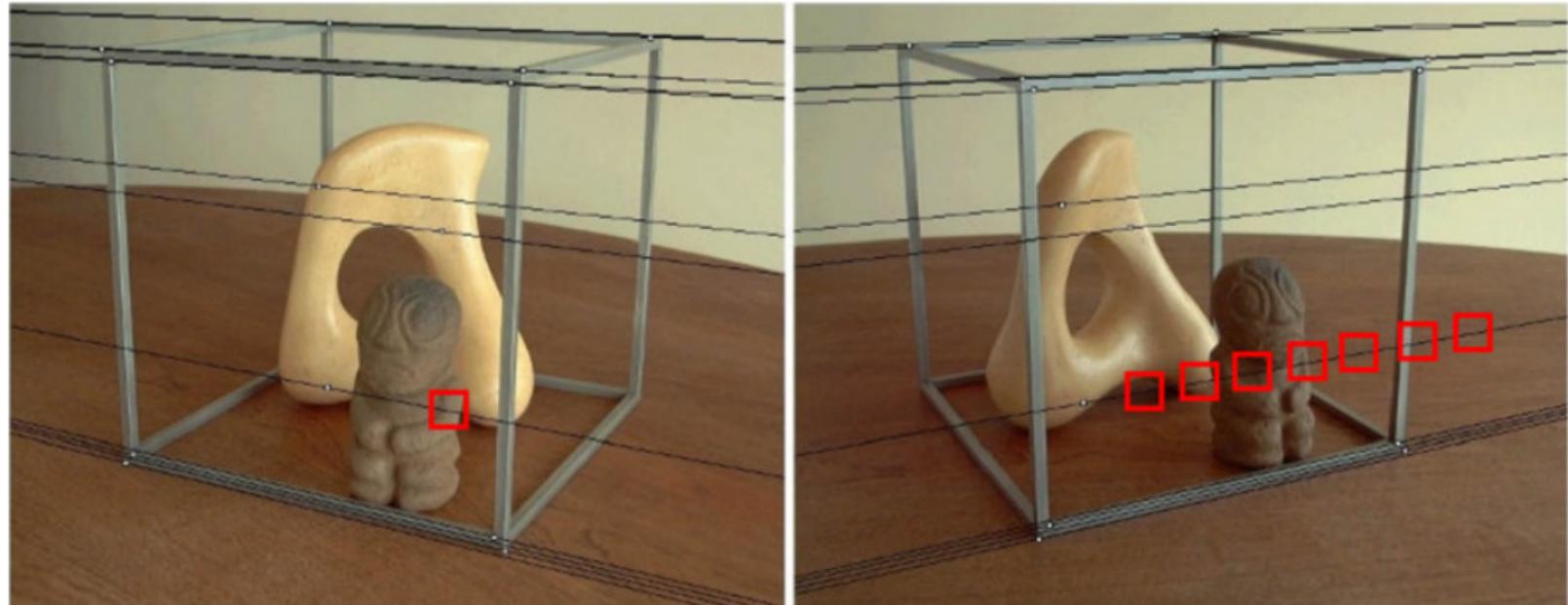
$\tilde{\mathbf{x}}_i = \mathbf{K}_i^{-1} \bar{\mathbf{x}}_i$ as we only know the pixel coordinates $\bar{\mathbf{x}}_i$. The essential matrix becomes

$$\tilde{\mathbf{x}}_2^\top \tilde{\mathbf{E}} \tilde{\mathbf{x}}_1 = \bar{\mathbf{x}}_2^\top \mathbf{K}_2^{-\top} \tilde{\mathbf{E}} \mathbf{K}_1^{-1} \bar{\mathbf{x}}_1 = \bar{\mathbf{x}}_2^\top \tilde{\mathbf{F}} \bar{\mathbf{x}}_1 = 0$$

where $\tilde{\mathbf{F}} = \mathbf{K}_2^{-\top} \tilde{\mathbf{E}} \mathbf{K}_1^{-1}$ is called the **fundamental matrix**. Like $\tilde{\mathbf{E}}$, $\tilde{\mathbf{F}}$ is (in absence of noise) rank two and the epipoles can be recovered in the same way. However, the intrinsic parameters cannot be directly determined, i.e., we obtain only a **perspective reconstruction** and not a metric one.

Additional information (vanishing points, constancy of \mathbf{K} across time, zero skew, aspect ratio) can be used to upgrade a perspective reconstruction to a metric one.

Epipolar Geometry

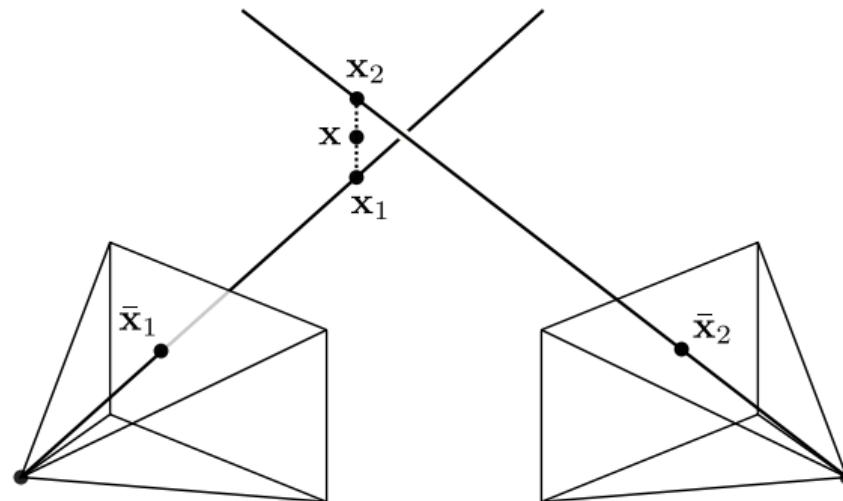


- ▶ **Epipolar lines** for two real images based on estimated epipolar geometry
- ▶ Note that corresponding points lie on the corresponding epipolar lines

Triangulation

Triangulation

Given the camera intrinsics and extrinsics, how can we recover 3D geometry?



Given noisy 2D image observations \bar{x}_1 and \bar{x}_2 , the two rays might not intersect in one point. We like to recover the 3D point x that (in some sense) is closest to the two rays.

Triangulation

Let $\tilde{\mathbf{x}}_i^s = \tilde{\mathbf{P}}_i \tilde{\mathbf{x}}_w$ denote the projection of a 3D world point $\tilde{\mathbf{x}}_w$ onto the image of the i 'th camera $\tilde{\mathbf{x}}_i^s$. As both sides are homogeneous, they have the same direction but may differ in magnitude. To account for this, we consider the cross product $\tilde{\mathbf{x}}_i^s \times \tilde{\mathbf{P}}_i \tilde{\mathbf{x}}_w = \mathbf{0}$. Using $\tilde{\mathbf{p}}_{ik}^\top$ to denote the k 'th row of the i 'th camera's projection matrix $\tilde{\mathbf{P}}_i$, we obtain:

$$\underbrace{\begin{bmatrix} x_i^s \tilde{\mathbf{p}}_{i3}^\top - \tilde{\mathbf{p}}_{i1}^\top \\ y_i^s \tilde{\mathbf{p}}_{i3}^\top - \tilde{\mathbf{p}}_{i2}^\top \end{bmatrix}}_{\mathbf{A}_i} \tilde{\mathbf{x}}_w = \mathbf{0}$$

Stacking $N \geq 2$ observations of a point, we obtain a linear system $\mathbf{A} \tilde{\mathbf{x}}_w = \mathbf{0}$. As $\tilde{\mathbf{x}}_w$ is homogeneous this leads to a constrained least squares problem. The solution to this problem is the **right singular vector** corresponding to the smallest singular value of \mathbf{A} .

Triangulation

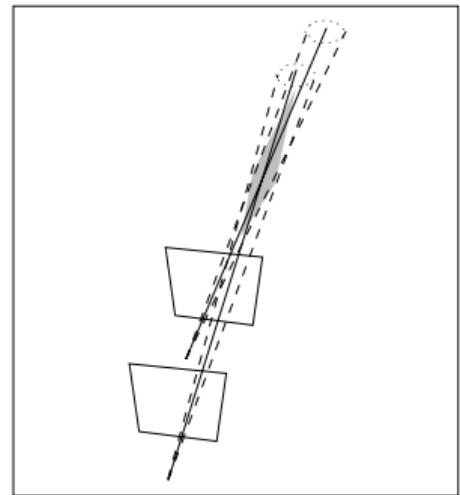
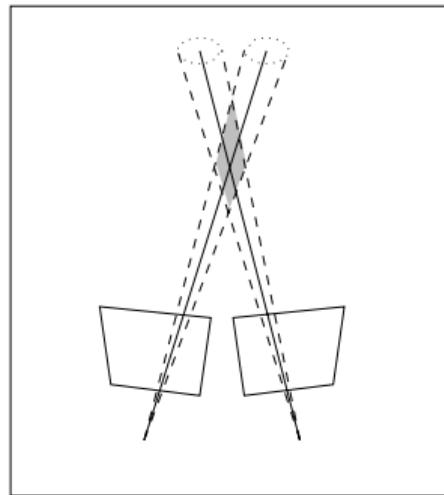
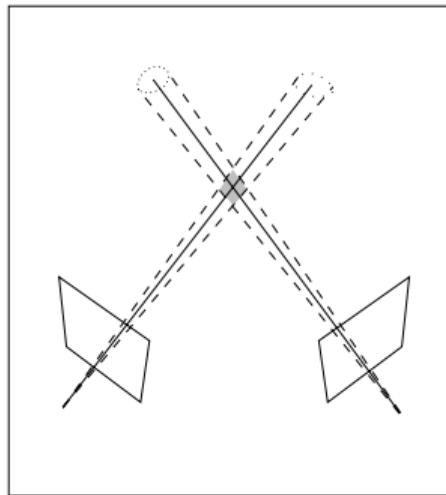
Reprojection Error Minimization:

- ▶ While this often works well, it is *not* invariant to perspective transformations (i.e. noise in the correspondence position not properly taken into account)
- ▶ The gold standard is to minimize the *reprojection* error using numerical methods:

$$\bar{\mathbf{x}}_w^* = \operatorname{argmin}_{\bar{\mathbf{x}}_w} \sum_{i=1}^N \|\bar{\mathbf{x}}_i^s(\bar{\mathbf{x}}_w) - \bar{\mathbf{x}}_i^o\|_2^2 \quad \text{with observation } \bar{\mathbf{x}}_i^o$$

- ▶ This allows to take measurement noise appropriately into account
- ▶ The minimum can also be obtained in closed form as the solution of a sixth degree polynomial, see Hartley & Zisserman, Section 12.5 for details

Triangulation Uncertainty



Triangulation works differently well depending on the relative camera pose:

- ▶ **Uncertainty** (shaded region) increases as the rays become more parallel
- ▶ **Tradeoff:** feature matching easier for nearby views, but triangulation is harder

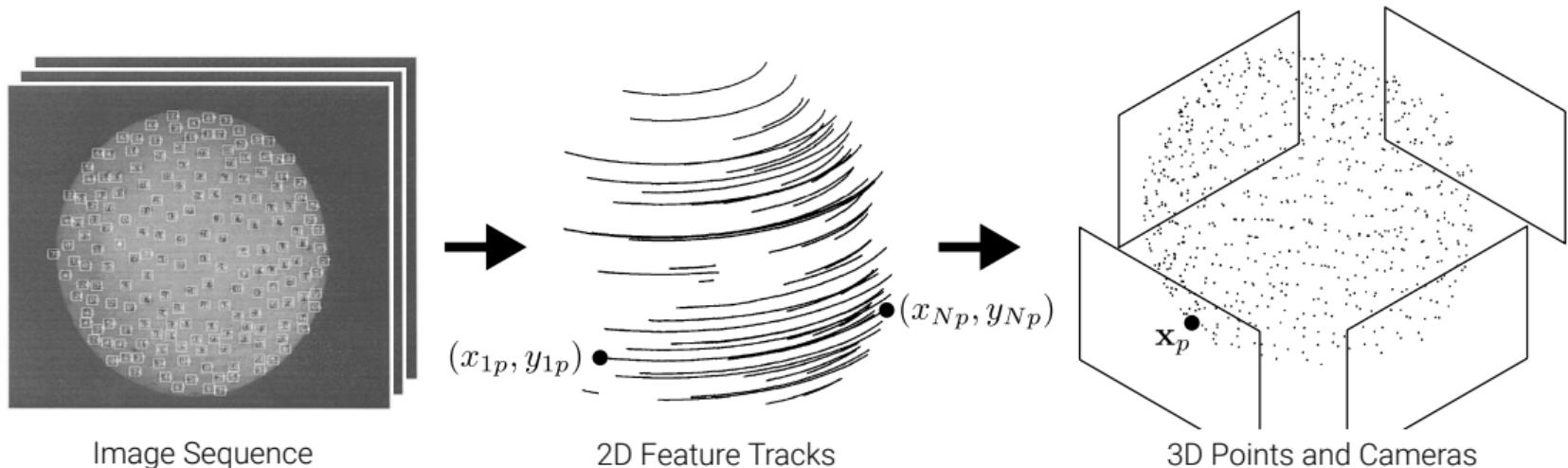
3.3

Factorization

Orthographic Factorization

How can we use more than two views for structure-from-motion?

Let $\mathcal{W} = \{(x_{ip}, y_{ip}) \mid i = 1, \dots, N, p = 1, \dots, P\}$ denote P feature points tracked over N frames. Given \mathcal{W} and assuming **orthographic projection**, our goal is to recover both **camera motion** (rotation) and the **structure** (3D points \mathbf{x}_p corresponding to (x_{ip}, y_{ip})):



Orthographic Factorization

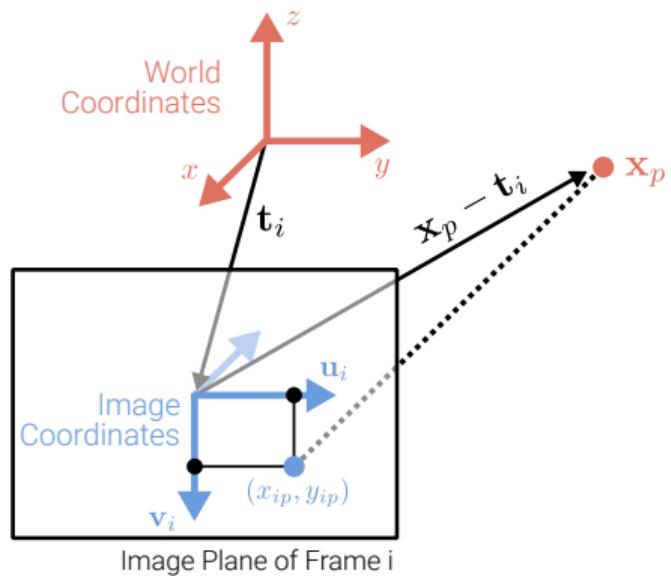
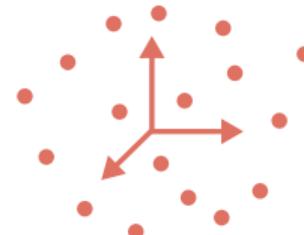
Under **orthographic projection**, a 3D point \mathbf{x}_p maps to a pixel (x_{ip}, y_{ip}) in frame i as:

$$x_{ip} = \mathbf{u}_i^\top (\mathbf{x}_p - \mathbf{t}_i)$$

$$y_{ip} = \mathbf{v}_i^\top (\mathbf{x}_p - \mathbf{t}_i)$$

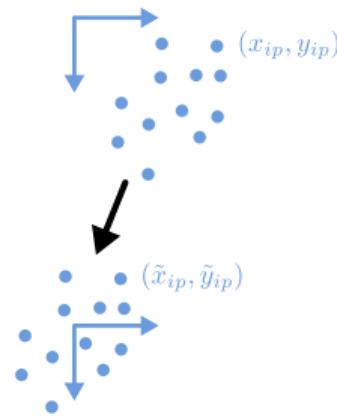
Without loss of generality, we assume that the **3D coordinate system** is at the center:

$$\frac{1}{P} \sum_{p=1}^P \mathbf{x}_p = 0$$



Orthographic Factorization

Let (x_{ip}, y_{ip}) denote the 2D location of feature p in frame i . Centering the features per frame (zero-mean) and collecting them yields the **centered measurement matrix** $\tilde{\mathbf{W}}$:



$$\tilde{x}_{ip} = x_{ip} - \frac{1}{P} \sum_{q=1}^P x_{iq}$$

$$\tilde{y}_{ip} = y_{ip} - \frac{1}{P} \sum_{q=1}^P y_{iq}$$

$$\tilde{\mathbf{W}} = \begin{bmatrix} \tilde{x}_{11} & \dots & \tilde{x}_{1P} \\ \vdots & & \vdots \\ \tilde{x}_{N1} & \dots & \tilde{x}_{NP} \\ \tilde{y}_{11} & \dots & \tilde{y}_{1P} \\ \vdots & & \vdots \\ \tilde{y}_{N1} & \dots & \tilde{y}_{NP} \end{bmatrix}$$

Remark: Here, the \sim notation denotes centered, not homogeneous coordinates.

Orthographic Factorization

As $x_{ip} = \mathbf{u}_i^\top (\mathbf{x}_p - \mathbf{t}_i)$, the **centered image x-coordinate** is given by:

$$\begin{aligned}\tilde{x}_{ip} &= x_{ip} - \frac{1}{P} \sum_{q=1}^P x_{iq} \\ &= \mathbf{u}_i^\top (\mathbf{x}_p - \mathbf{t}_i) - \frac{1}{P} \sum_{q=1}^P \mathbf{u}_i^\top (\mathbf{x}_q - \mathbf{t}_i) \\ &= \mathbf{u}_i^\top (\mathbf{x}_p - \mathbf{t}_i) - \mathbf{u}_i^\top \frac{1}{P} \sum_{q=1}^P \mathbf{x}_q + \mathbf{u}_i^\top \mathbf{t}_i \\ &= \mathbf{u}_i^\top \left(\mathbf{x}_p - \frac{1}{P} \sum_{q=1}^P \mathbf{x}_q \right) = \mathbf{u}_i^\top \mathbf{x}_p\end{aligned}$$

Orthographic Factorization

We obtain a similar equation for **centered image y-coordinate**, thus we have:

$$\tilde{x}_{ip} = \mathbf{u}_i^\top \mathbf{x}_p \quad \tilde{y}_{ip} = \mathbf{v}_i^\top \mathbf{x}_p$$

The **centered measurement matrix** $\tilde{\mathbf{W}}$ can therefore be expressed as follows:

$$\tilde{\mathbf{W}} = \mathbf{R} \mathbf{X} \quad \text{with} \quad \tilde{\mathbf{W}} = \begin{bmatrix} \tilde{x}_{11} & \dots & \tilde{x}_{1P} \\ \vdots & & \vdots \\ \tilde{x}_{N1} & \dots & \tilde{x}_{NP} \\ \tilde{y}_{11} & \dots & \tilde{y}_{1P} \\ \vdots & & \vdots \\ \tilde{y}_{N1} & \dots & \tilde{y}_{NP} \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \mathbf{u}_1^\top \\ \vdots \\ \mathbf{u}_N^\top \\ \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_N^\top \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_P \end{bmatrix}$$

Orthographic Factorization

$$\tilde{\mathbf{W}} = \mathbf{R} \mathbf{X} \quad \text{with} \quad \tilde{\mathbf{W}} = \begin{bmatrix} \tilde{x}_{11} & \dots & \tilde{x}_{1P} \\ \vdots & & \vdots \\ \tilde{x}_{N1} & \dots & \tilde{x}_{NP} \\ \tilde{y}_{11} & \dots & \tilde{y}_{1P} \\ \vdots & & \vdots \\ \tilde{y}_{N1} & \dots & \tilde{y}_{NP} \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \mathbf{u}_1^\top \\ \vdots \\ \mathbf{u}_N^\top \\ \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_N^\top \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_P \end{bmatrix}$$

Here, \mathbf{R} represents the camera **motion** (rotation), and \mathbf{X} the **structure** of the 3D scene. As $\mathbf{R} \in \mathbb{R}^{2N \times 3}$ and $\mathbf{X} \in \mathbb{R}^{3 \times P}$, in the absence of noise, the centered measurement matrix $\tilde{\mathbf{W}}$ has at most **rank 3**. When adding noise, the matrix becomes **full rank**.

Orthographic Factorization

$$\underbrace{\tilde{W}}_{\mathbb{R}^{2N \times P}} = \underbrace{R}_{\mathbb{R}^{2N \times 3}} \underbrace{X}_{\mathbb{R}^{3 \times P}}$$

Given observed measurements with full rank \tilde{W} , we can find a rank 3 approximation \hat{W} minimizing $\|\hat{W} - \tilde{W}\|_F$ by using **singular value decomposition (SVD)**:

$$\hat{W} = U \Sigma V^\top$$

We obtain the **rank 3 factorization** by considering the singular vectors corresponding to the top 3 singular values (the others should be small and capture noise primarily):

$$\underbrace{\hat{R}}_{\mathbb{R}^{2N \times 3}} = \underbrace{U}_{\mathbb{R}^{2N \times 3}} \underbrace{\Sigma^{\frac{1}{2}}}_{\mathbb{R}^{3 \times 3}} \quad \underbrace{\hat{X}}_{\mathbb{R}^{3 \times P}} = \underbrace{\Sigma^{\frac{1}{2}}}_{\mathbb{R}^{3 \times 3}} \underbrace{V^\top}_{\mathbb{R}^{3 \times P}}$$

Orthographic Factorization

$$\hat{\mathbf{W}} = \mathbf{U}\Sigma\mathbf{V}^\top \quad \Rightarrow \quad \hat{\mathbf{R}} = \mathbf{U}\Sigma^{\frac{1}{2}} \quad \hat{\mathbf{X}} = \Sigma^{\frac{1}{2}}\mathbf{V}^\top$$

However, this decomposition is **not unique** as there exists a matrix $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ such that

$$\hat{\mathbf{W}} = \mathbf{U}\Sigma\mathbf{V}^\top = \hat{\mathbf{R}} \hat{\mathbf{X}} = (\hat{\mathbf{R}}\mathbf{Q})(\mathbf{Q}^{-1}\hat{\mathbf{X}})$$

To find \mathbf{Q} we observe that the rows of \mathbf{R} are unit vectors and the first half are orthogonal to the corresponding second half of \mathbf{R} . We obtain the **metric constraints**:

$$\hat{\mathbf{u}}_i^\top \mathbf{Q} (\hat{\mathbf{u}}_i^\top \mathbf{Q})^\top = \hat{\mathbf{u}}_i^\top \mathbf{Q} \mathbf{Q}^\top \hat{\mathbf{u}}_i = 1$$

$$\hat{\mathbf{v}}_i^\top \mathbf{Q} (\hat{\mathbf{v}}_i^\top \mathbf{Q})^\top = \hat{\mathbf{v}}_i^\top \mathbf{Q} \mathbf{Q}^\top \hat{\mathbf{v}}_i = 1$$

$$\hat{\mathbf{u}}_i^\top \mathbf{Q} (\hat{\mathbf{v}}_i^\top \mathbf{Q})^\top = \hat{\mathbf{u}}_i^\top \mathbf{Q} \mathbf{Q}^\top \hat{\mathbf{v}}_i = 0$$

Orthographic Factorization

Metric Constraints:

$$\hat{\mathbf{u}}_i^\top \mathbf{Q} (\hat{\mathbf{u}}_i^\top \mathbf{Q})^\top = \hat{\mathbf{u}}_i^\top \mathbf{Q} \mathbf{Q}^\top \hat{\mathbf{u}}_i = 1$$

$$\hat{\mathbf{v}}_i^\top \mathbf{Q} (\hat{\mathbf{v}}_i^\top \mathbf{Q})^\top = \hat{\mathbf{v}}_i^\top \mathbf{Q} \mathbf{Q}^\top \hat{\mathbf{v}}_i = 1$$

$$\hat{\mathbf{u}}_i^\top \mathbf{Q} (\hat{\mathbf{v}}_i^\top \mathbf{Q})^\top = \hat{\mathbf{u}}_i^\top \mathbf{Q} \mathbf{Q}^\top \hat{\mathbf{v}}_i = 0$$

This gives us a large set of linear equations for the entries in the matrix $\mathbf{Q} \mathbf{Q}^\top$, from which the matrix \mathbf{Q} can be recovered using standard Cholesky decomposition.

Orthographic Factorization

Algorithm:

1. Take measurements $\hat{\mathbf{W}}$
2. Compute SVD $\hat{\mathbf{W}} = \mathbf{U}\Sigma\mathbf{V}^\top$ and keep the top 3 SVs
3. Define $\hat{\mathbf{R}} = \mathbf{U}\Sigma^{\frac{1}{2}}$ and $\hat{\mathbf{X}} = \Sigma^{\frac{1}{2}}\mathbf{V}^\top$
4. Compute $\mathbf{Q}\mathbf{Q}^\top$ and from this \mathbf{Q}
5. Compute $\mathbf{R} = \hat{\mathbf{R}}\mathbf{Q}$ and $\mathbf{X} = \mathbf{Q}^{-1}\hat{\mathbf{X}}$

Remarks:

- Advantage: **closed form solution** (determined up to an arbitrary global rotation)
- Disadvantage: **complete feature tracks** required (\Rightarrow cannot handle occlusions)
- Solution: Apply to subsets of features/frames and propagate (see T&K, Sec. 5)

Orthographic Factorization: Results



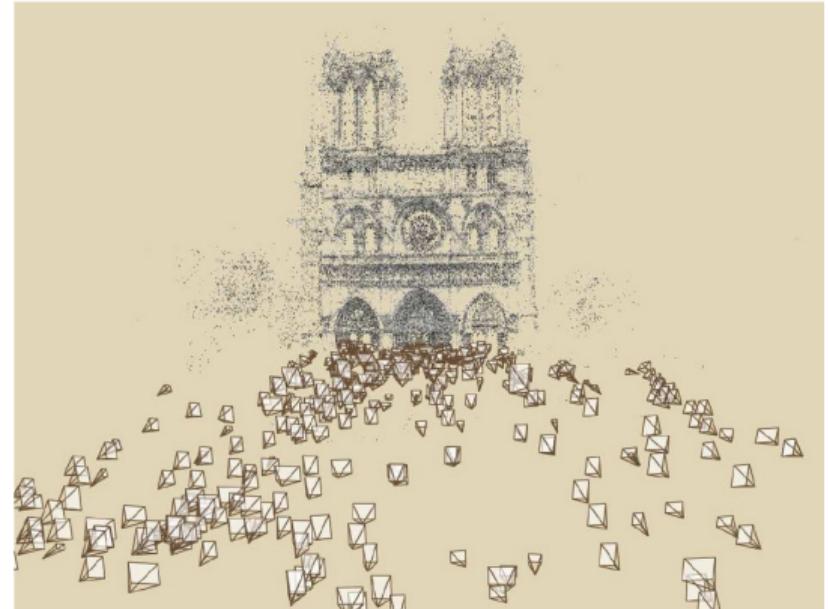
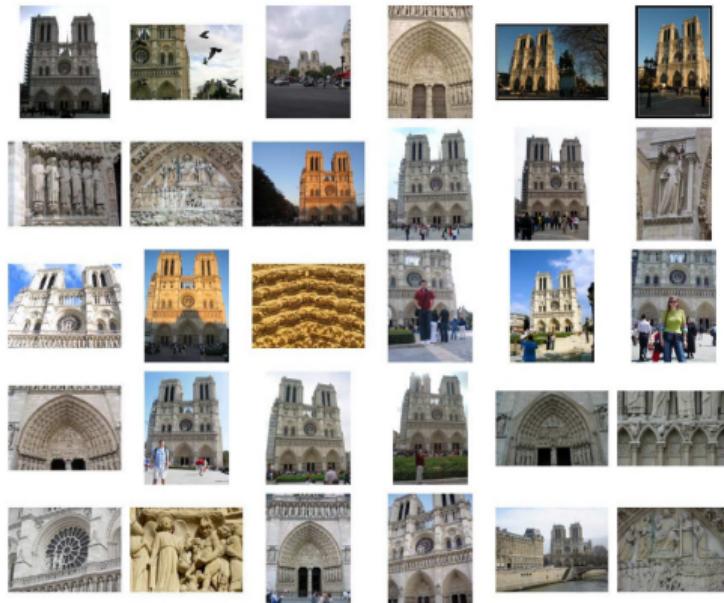
Perspective Factorization

- ▶ Tomasi and Kanade's original factorization approach assume **orthography**
- ▶ Christy and Horaud (TPAMI, 1996) perform an initial orthographic reconstruction and then correct perspective in an iterative manner
- ▶ Triggs (CVPR, 1996) performs **projective factorization**, iteratively updating depth
- ▶ Even though inaccurate, factorization methods can provide a **good initialization** for iterative techniques such as bundle adjustment
- ▶ However, modern SfM approaches (e.g., COLMAP) often perform **incremental bundle adjustment**, initializing with a carefully selected two-view reconstruction and iteratively adding new images/cameras to the reconstruction

3.4

Bundle Adjustment

Bundle Adjustment



- ▶ **Goal: Optimize reprojection errors** (distance between observed feature and projected 3D point in image plane) **wrt. camera parameters and 3D point cloud**

Bundle Adjustment

Let $\Pi = \{\pi_i\}$ denote the N cameras including their intrinsic and extrinsic parameters.

Let $\mathcal{X}_w = \{\mathbf{x}_p^w\}$ with $\mathbf{x}_p^w \in \mathbb{R}^3$ denote the set of P 3D points in world coordinates.

Let $\mathcal{X}_s = \{\mathbf{x}_{ip}^s\}$ with $\mathbf{x}_{ip}^s \in \mathbb{R}^2$ denote the image (screen) observations in all i cameras.

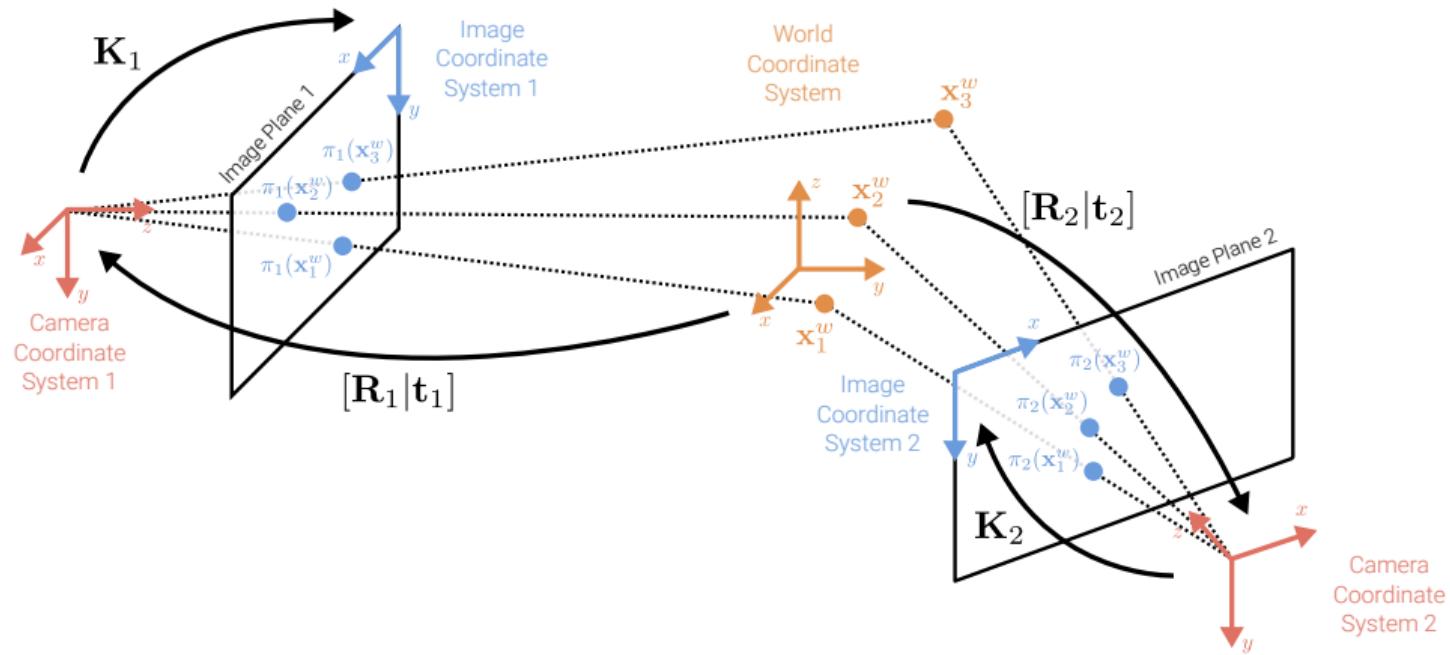
Bundle adjustment minimizes the reprojection error of all observations:

$$\Pi^*, \mathcal{X}_w^* = \underset{\Pi, \mathcal{X}_w}{\operatorname{argmin}} \sum_{i=1}^N \sum_{p=1}^P w_{ip} \|\mathbf{x}_{ip}^s - \pi_i(\mathbf{x}_p^w)\|_2^2$$

Here, w_{ip} indicates if point p is observed in image i and $\pi_i(\mathbf{x}_p^w)$ is the 3D-to-2D projection of 3D world point \mathbf{x}_p^w onto the 2D image plane of the i 'th camera, i.e.:

$$\pi_i(\mathbf{x}_p^w) = \begin{pmatrix} \tilde{x}_p^s / \tilde{w}_p^s \\ \tilde{y}_p^s / \tilde{w}_p^s \end{pmatrix} \quad \text{with} \quad \tilde{\mathbf{x}}_p^s = \mathbf{K}_i(\mathbf{R}_i \mathbf{x}_p^w + \mathbf{t}_i)$$

Bundle Adjustment



\mathbf{K}_i and $[\mathbf{R}_i | \mathbf{t}_i]$ are the intrinsic and extrinsic parameters of π_i , respectively. During bundle adjustment, we optimize $\{(\mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i)\}$ and $\{\mathbf{x}_p^w\}$ jointly.

Challenges of Bundle Adjustment

Initialization:

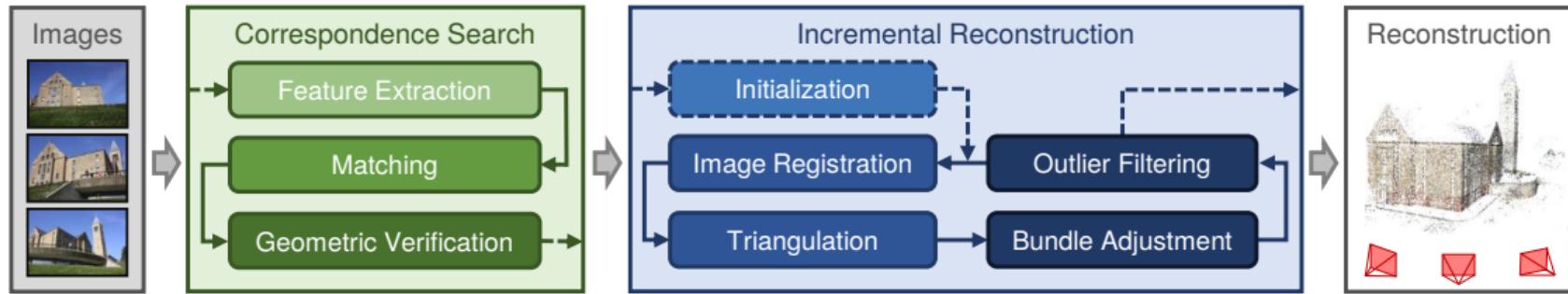
- ▶ The energy landscape of the bundle adjustment problem is highly **non-convex**
- ▶ A good **initialization** is crucial to avoid getting trapped in bad local minima
- ▶ As initializing all 3D points and cameras jointly is difficult (occlusion, viewpoint, matching outliers), **incremental bundle adjustment** initializes with a carefully selected two-view reconstruction and iteratively adds new images/cameras

Optimization:

- ▶ Given millions of features and thousands of cameras, large-scale bundle adjustment is **computationally demanding** (cubic complexity in #unknowns)
- ▶ Luckily, the problem is **sparse** (not all 3D points are observed in every camera), and efficient sparse implementations (e.g., Ceres) can be exploited in practice

Incremental Structure-from-Motion

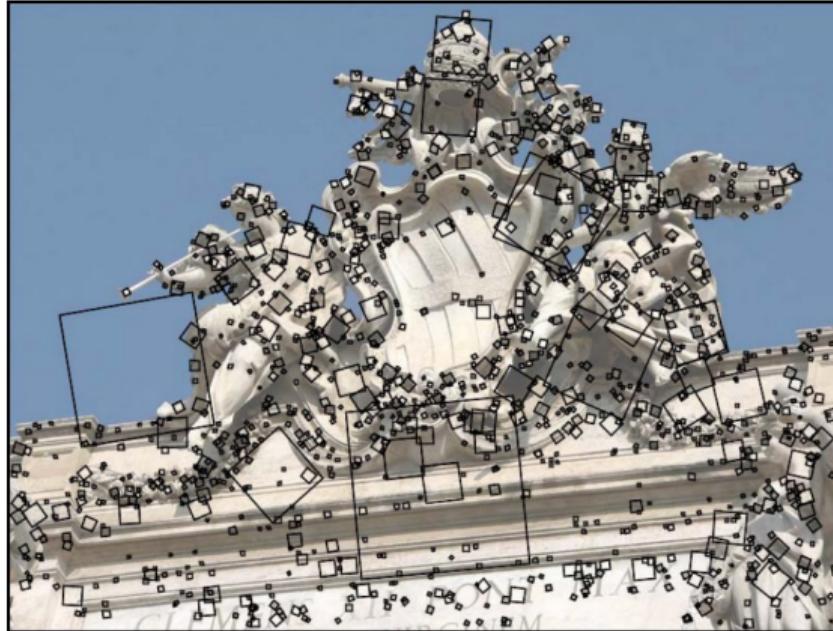
Incremental Structure-from-Motion



Feature matching and reconstruction pipeline (COLMAP):

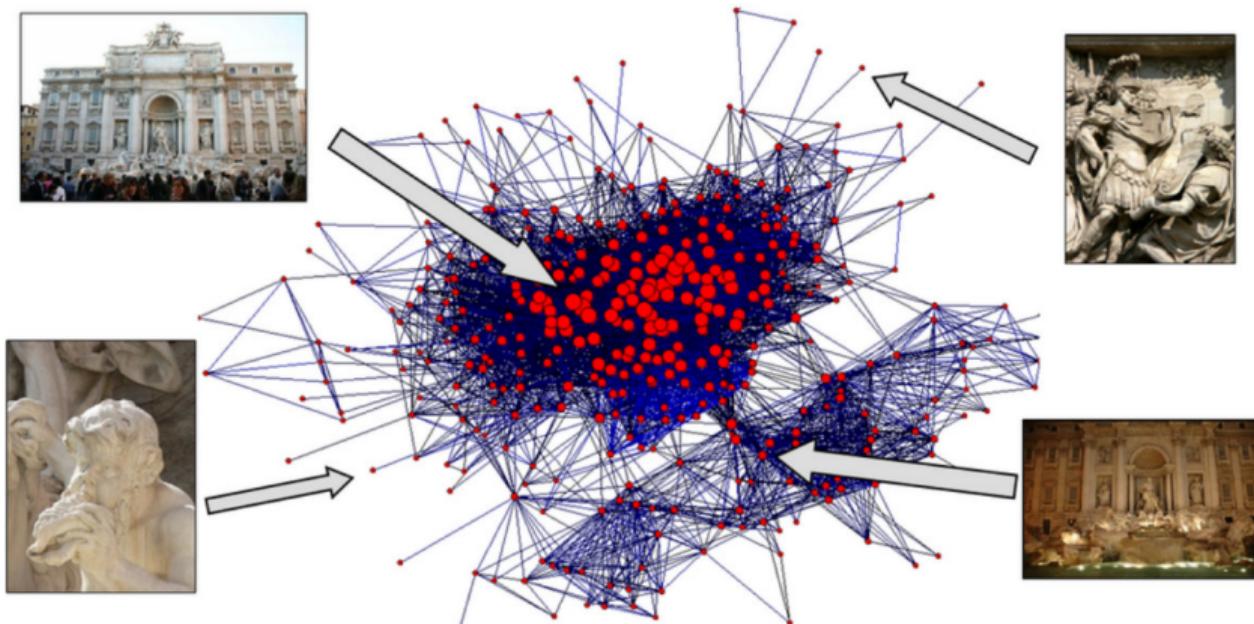
- ▶ **Correspondence Search:** Find and match robust 2D features across images
- ▶ **Incremental Reconstruction:** Start with 2 views, incrementally add cameras

Feature Extraction



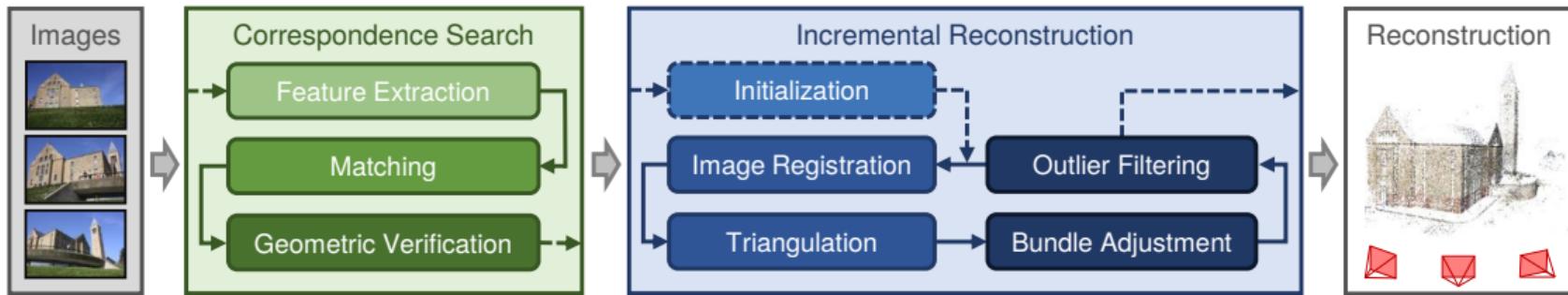
- ▶ **Detect features** (eg, SIFT, SURF, BRISK) in all input images

Feature Matching & Geometric Verification



- Find **overlapping image pairs** and **associated feature correspondences**

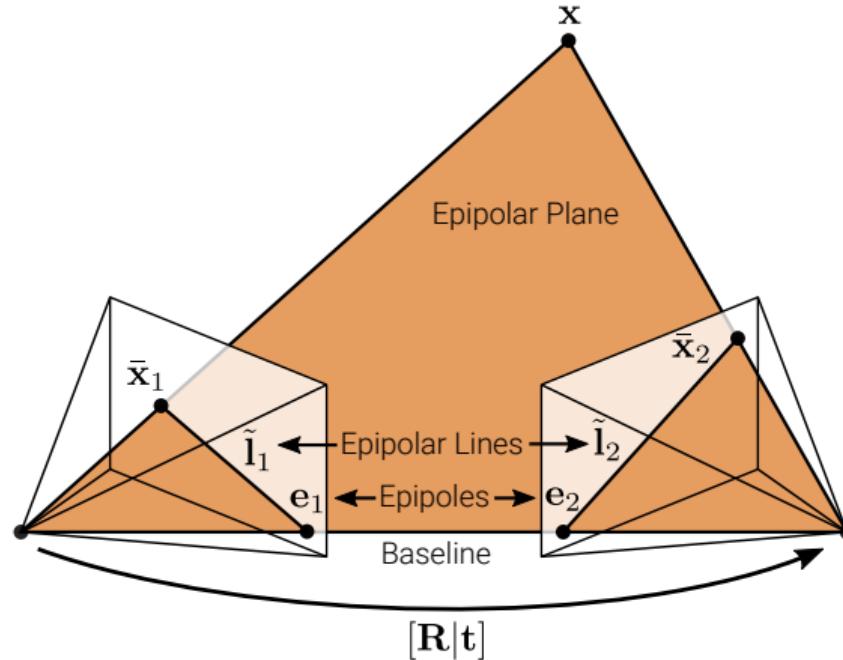
Incremental Structure-from-Motion



Feature matching and reconstruction pipeline (COLMAP):

- ▶ **Correspondence Search:** Find and match robust 2D features across images
- ▶ **Incremental Reconstruction:** Start with 2 views, incrementally add cameras

Initialization



- ▶ Select **two views** with many correspondences and **estimate their geometry**

Image Registration

Find a **new image** with correspondences to the current set and **estimate camera pose**:

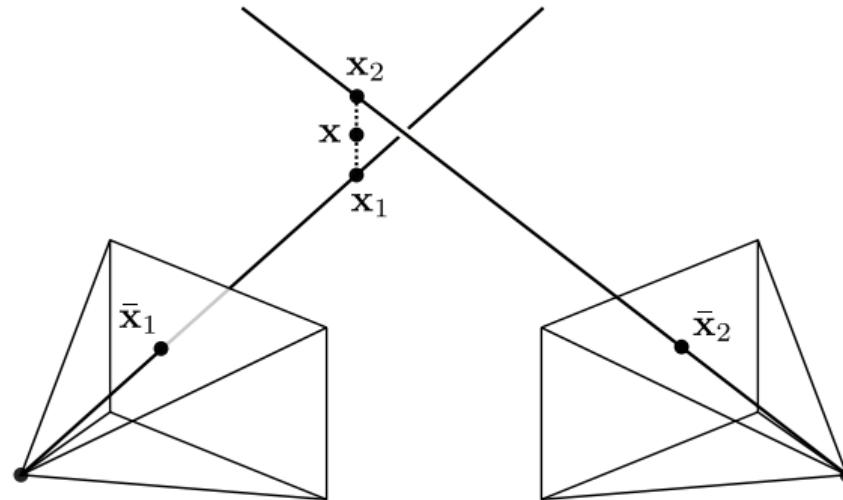
Let $\mathcal{X} = \{\bar{\mathbf{x}}_i^s, \bar{\mathbf{x}}_i^w\}_{i=1}^N$ be a set of N 3D-to-2D correspondences related by $\bar{\mathbf{x}}_i^s = \mathbf{P} \bar{\mathbf{x}}_i^w$.

As the correspondence vectors are homogeneous, they have the same direction but differ in magnitude. Thus, the equation above can be expressed as $\bar{\mathbf{x}}_i^s \times \mathbf{P} \bar{\mathbf{x}}_i^w = \mathbf{0}$.

Using the **Direct Linear Transform**, this can be written as a linear equation in the entries of \mathbf{P} . The solution to the constrained system (fixing \mathbf{P} 's scale) is given by SVD.

- ▶ Given that $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ and \mathbf{K} is upper-triangular, both \mathbf{K} and \mathbf{R} can be easily obtained from the front 3×3 submatrix of \mathbf{P} using standard RQ factorization
- ▶ If \mathbf{K} is known, we can even estimate \mathbf{P} from only three points (P3P algorithm)
- ▶ In practice, random sampling consensus (RANSAC) is used to remove outliers

Triangulation



- ▶ Given the newly registered image, **new correspondences** can be **triangulated**
- ▶ In COLMAP, a robust triangulation method is proposed that handles outliers

Bundle Adjustment & Outlier Filtering

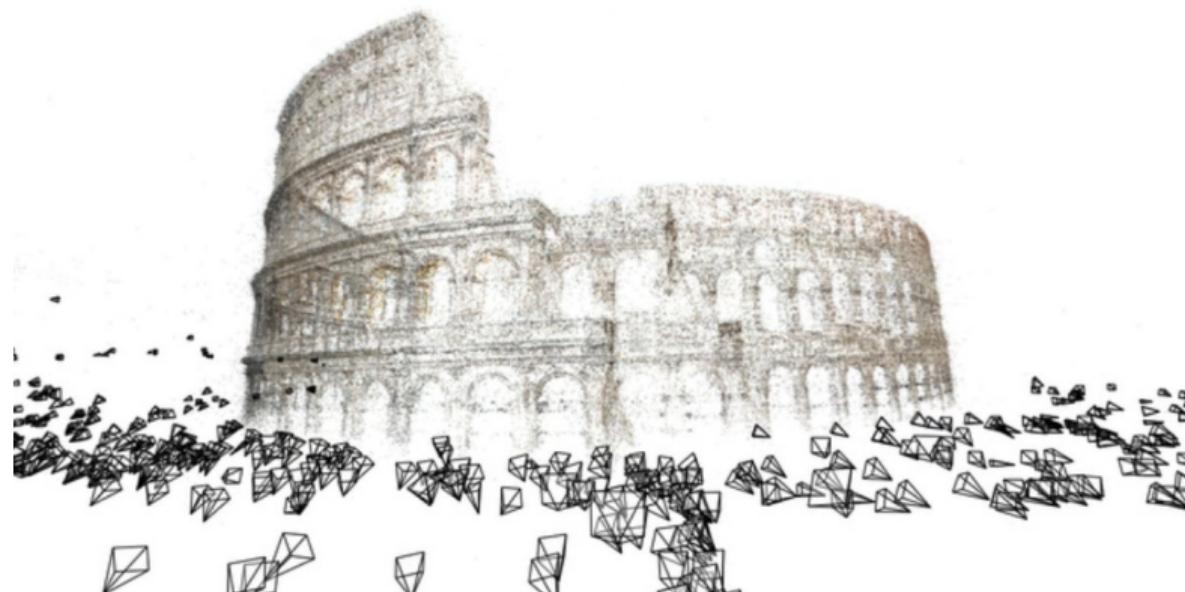
Minimize reprojection error of all observations wrt. all cameras and 3D points:

$$\Pi^*, \mathcal{X}_w^* = \operatorname{argmin}_{\Pi, \mathcal{X}_w} \sum_{i=1}^N \sum_{p=1}^P w_{ip} \|\mathbf{x}_{ip}^s - \pi_i(\mathbf{x}_p^w)\|_2^2$$

- ▶ Since incremental SfM only affects the model locally, COLMAP performs **local BA** on the locally connected images, and **global BA** only once in a while for efficiency
- ▶ For solving the sparse large-scale optimization problem, COLMAP uses Ceres
- ▶ After each BA, observations with large reprojection errors and cameras with abnormal field of views or large distortion coefficients are removed

Results and Applications

Building Rome in a Day



- First large-scale SfM from unstructured images: **Building Rome in a Day**

COLMAP SfM



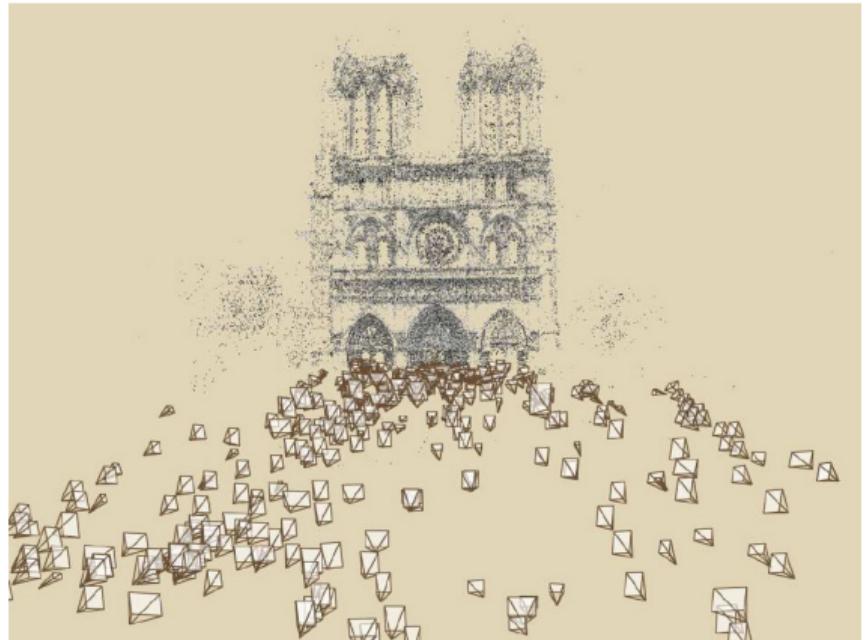
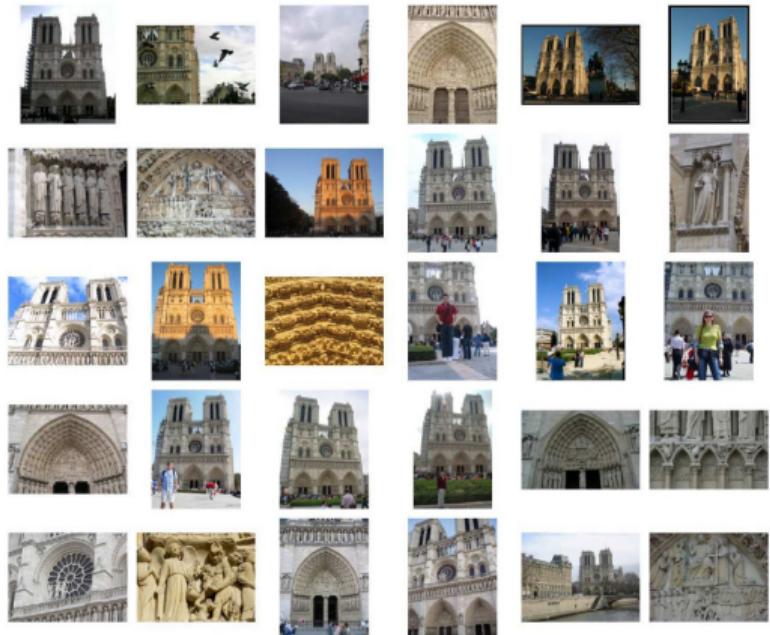
- **COLMAP** significantly improves accuracy and robustness compared to prior work

COLMAP MVS



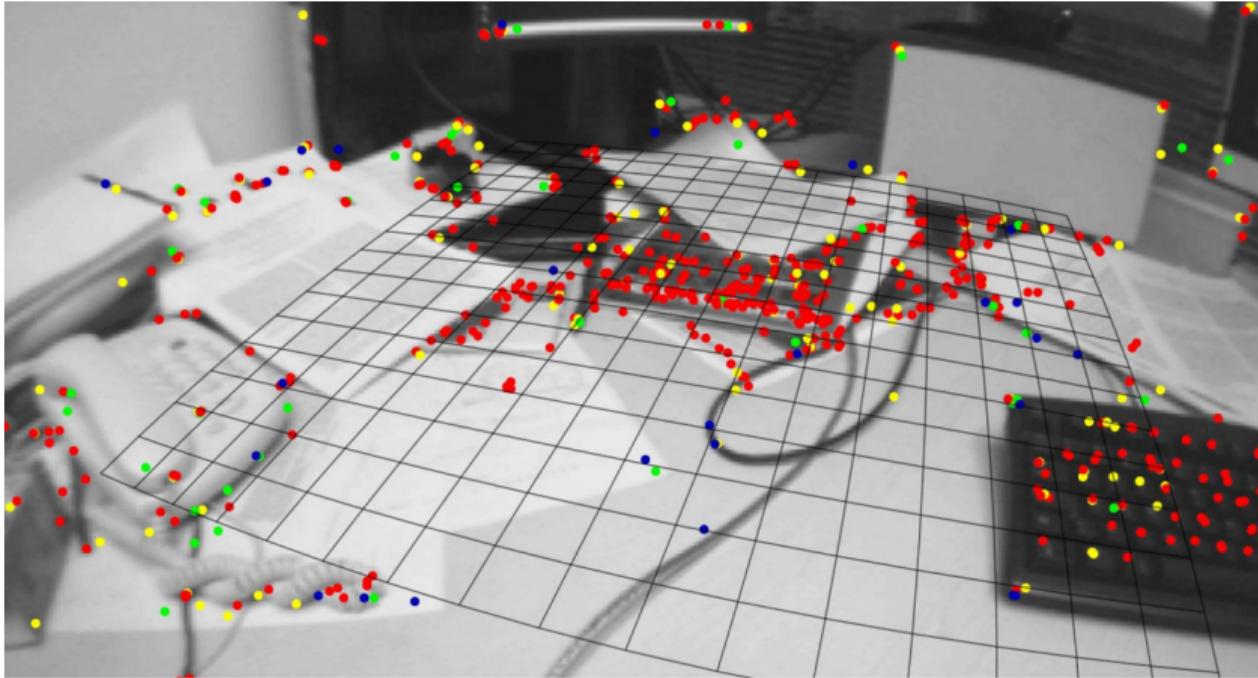
- COLMAP features a second **multi-view stereo stage** to obtain dense geometry

Photo Tourism



- **Photo Tourism / PhotoSynth** allows for exploring photo collections in 3D

Parallel Tracking and Mapping (PTAM)



- ▶ **PTAM** demonstrates real-time tracking and mapping of small workspaces

Match Move



- 2d3's automated camera motion tracker **Boujou** won an Emmy award in 2002