MAEG 5720: Computer Vision in Practice

Lecture 14a:

Introduction to Structure from Motion (SFM)

Dr. Terry Chang

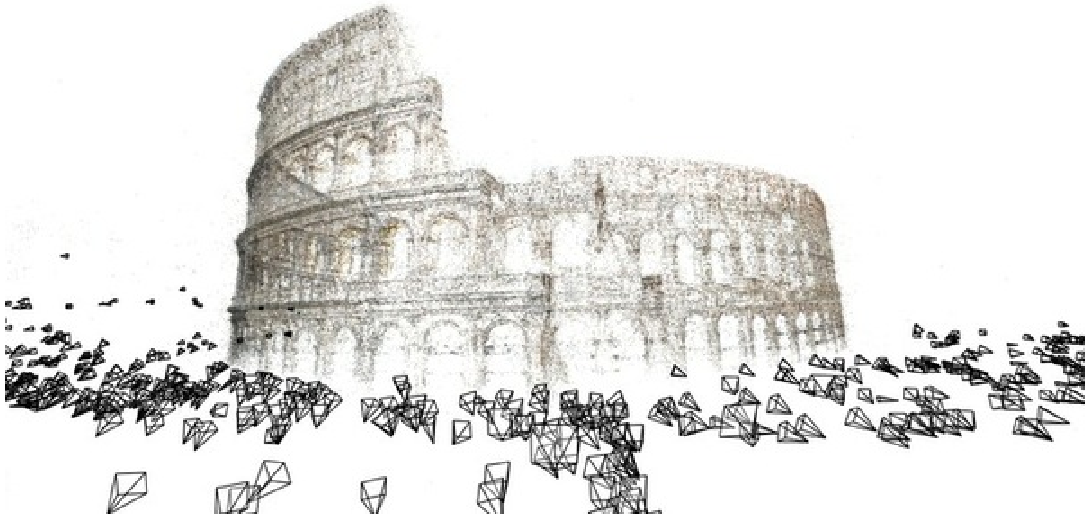
2021-2022

Semester 1

香港中文大學
The Chinese University of Hong Kong

Department of Mechanical and Automation Engineering
機械與自動化工程學系

Structure from motion



N. Snavely, S. Seitz, and R. Szeliski, Photo tourism: Exploring photo collections in 3D, SIGGRAPH 2006.

Image Credit: Building Rome in a Day (washington.edu)

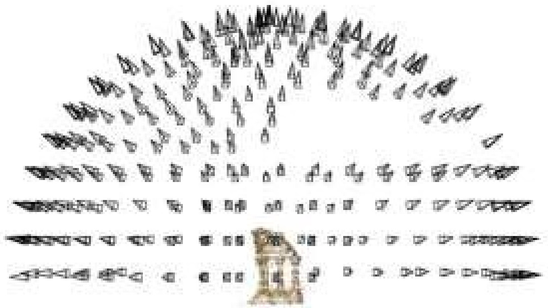
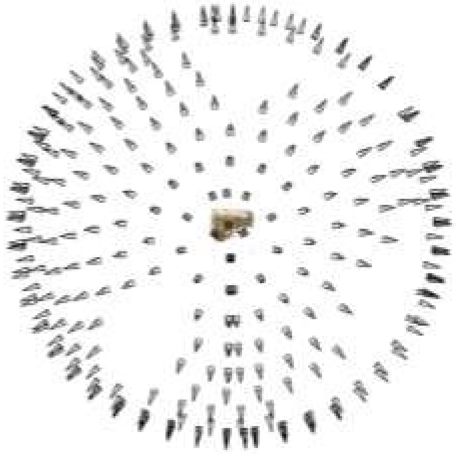
Two views

- Search for correspondences
- Compute fundamental matrix/Essential matrix
- Factorize into camera intrinsic, rotation and translation
- Triangulate the 3D points

What about more than two view?

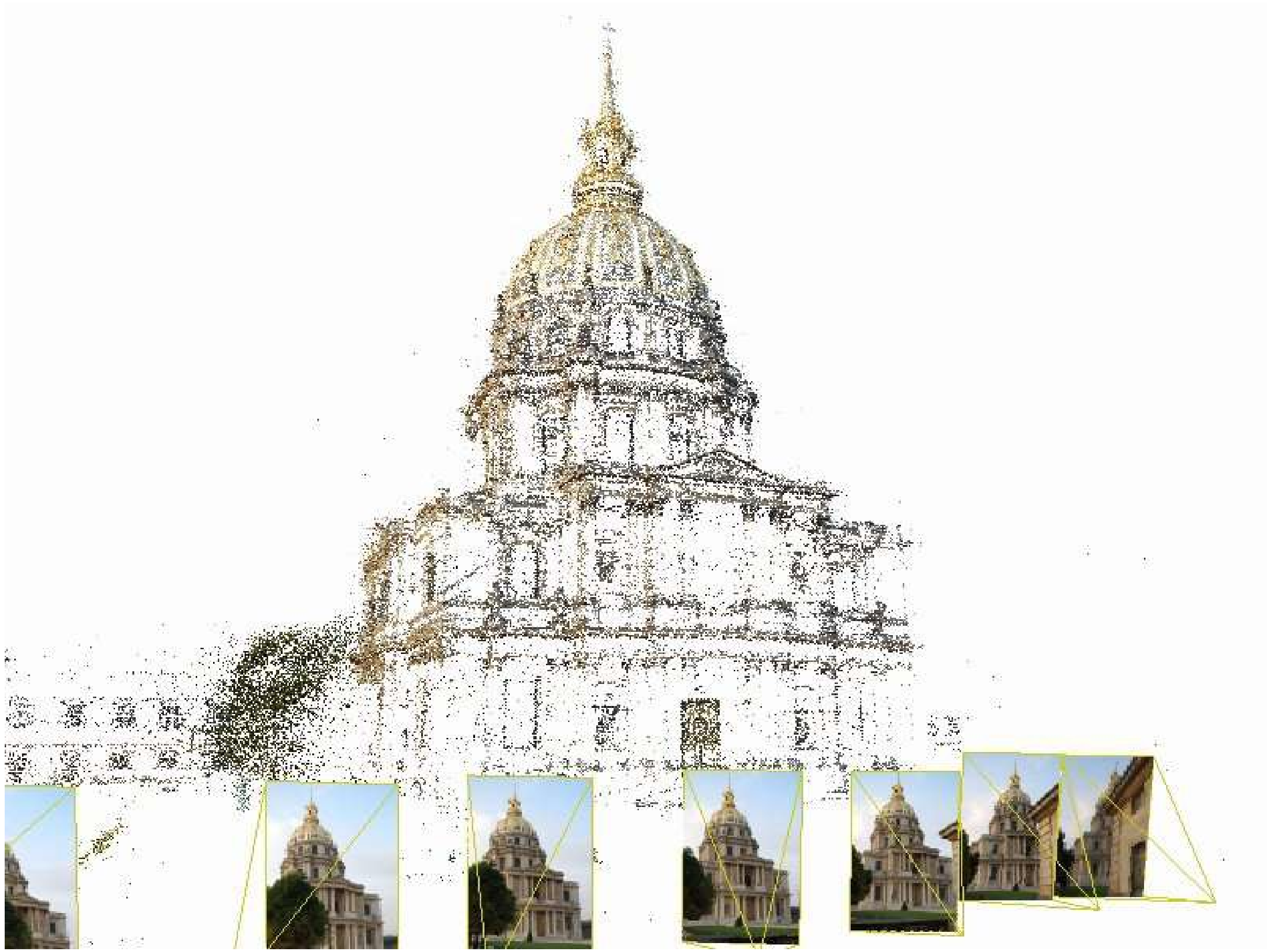
- The geometry of three views is described as *trifocal tensor*
- The geometry of four views is described by *quadrifocal tensor*
- How about more camera views?
  - How can we figure out where are the cameras?
  - How to reconstruct the 3D model of the scene?
  - This is the *structure from motion* problem.

Structure from motion

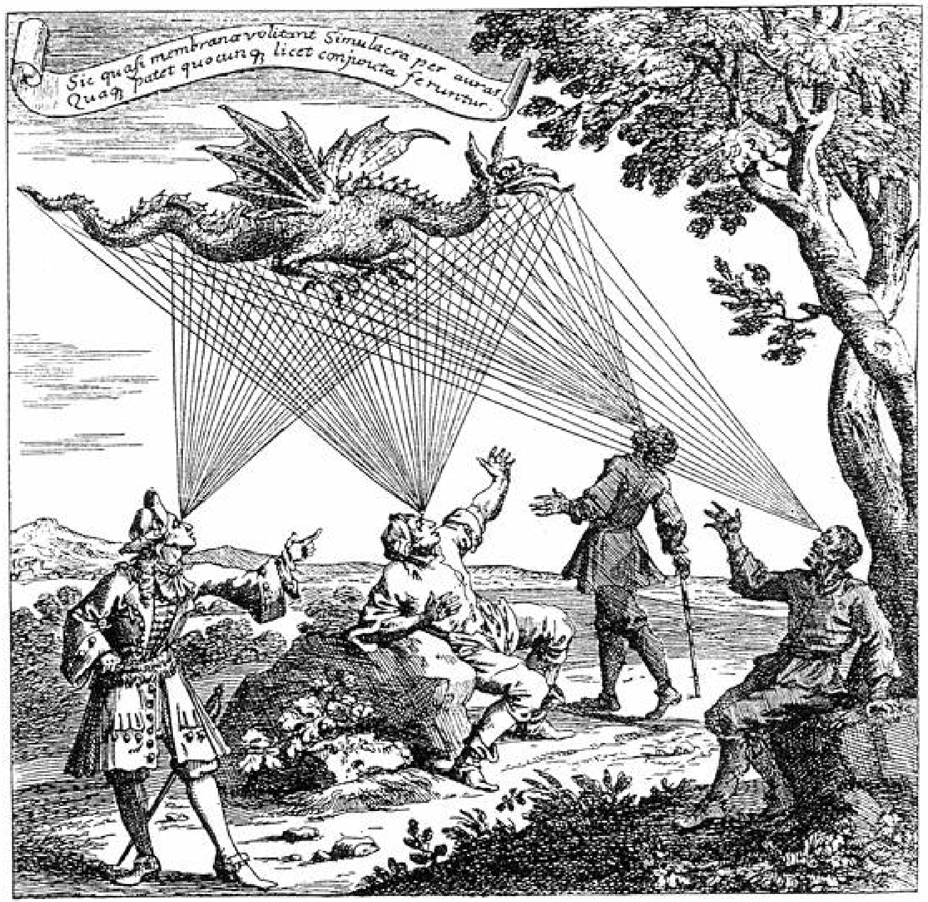




- Input:
  - Images with points in *correspondence* $p_{i,j} = (u_{i,j}, v_{i,j})$
- Output:
  - Structure: 3D location x_i for each p_i
  - Motion: camera parameters R_j, $\mathbf{t}_j$ and possibly K_j
- Objective minimize reprojection error

Also doable by video



Structure from motion



Драконъ, видимый подъ различными углами зрѣнія
По гравюрѣ на мѣди изъ „Oculus artificialis teledioptricus" Цана. 1702 года.

Camera calibration & triangulation

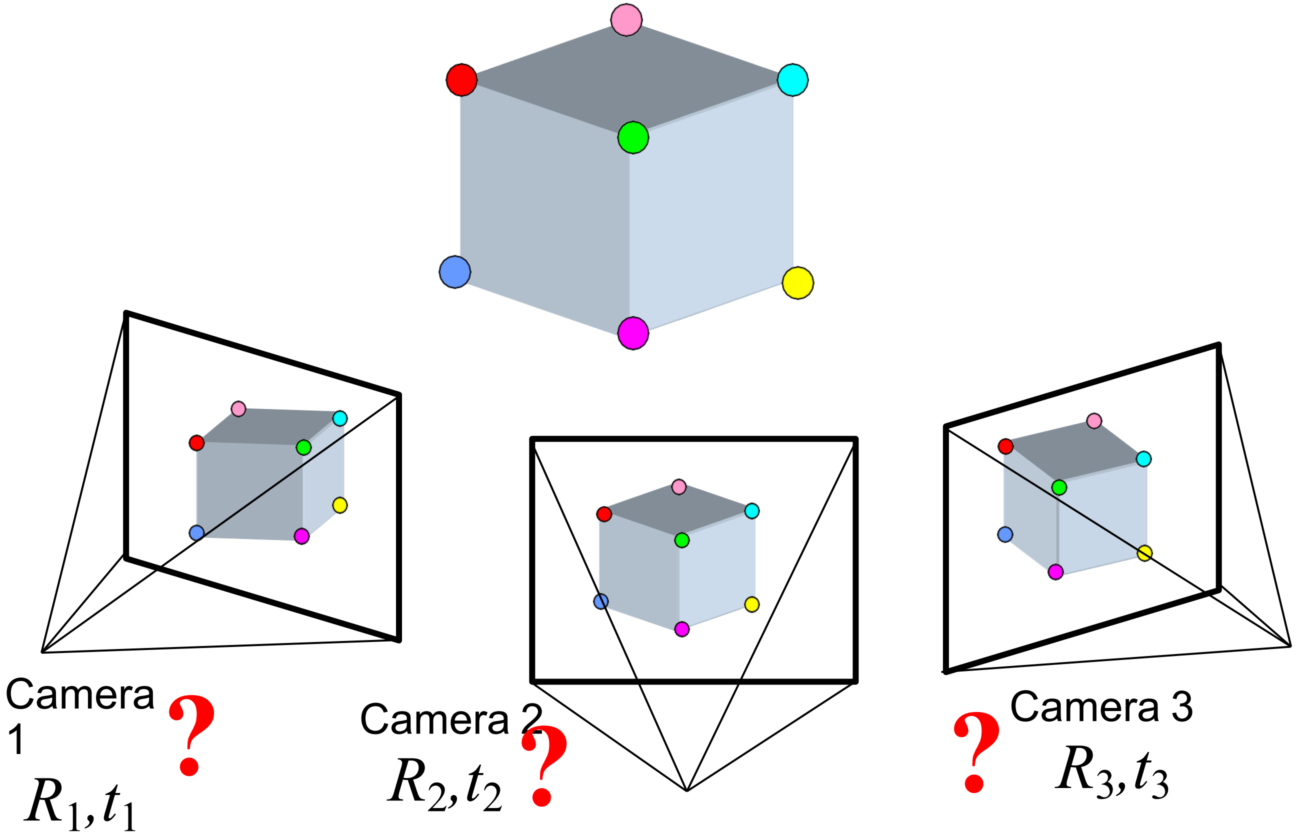
- Suppose we *know 3D points*
  - And have matches between these points and an image
  - How can we *compute the camera parameters*?

- Suppose we have *know camera parameters*, each of which observes a point
  - How can we *compute the 3D location* of that point?

- *SFM* solves both of these problems *at once*
- A kind of chicken-and-egg problem
  - (but solvable)

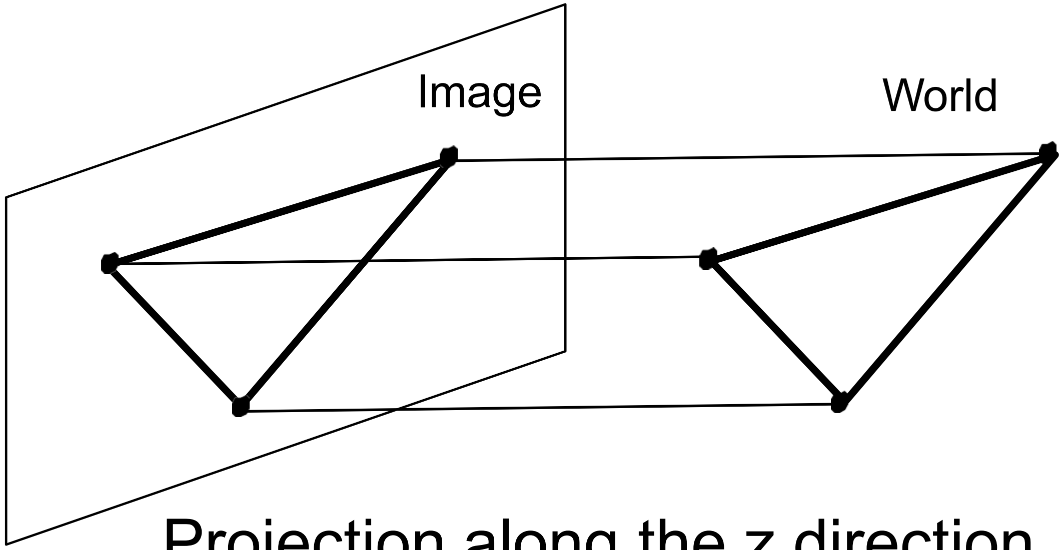
Structure from motion

- Given a set of corresponding ***2D image points*** $(u_{f,p}, v_{f,p})$ in two or more images, compute the camera parameters and the ***3D point coordinates*** (P_p)



Slide credit: Noah Snavely

Let's look at a simple orthographic projection

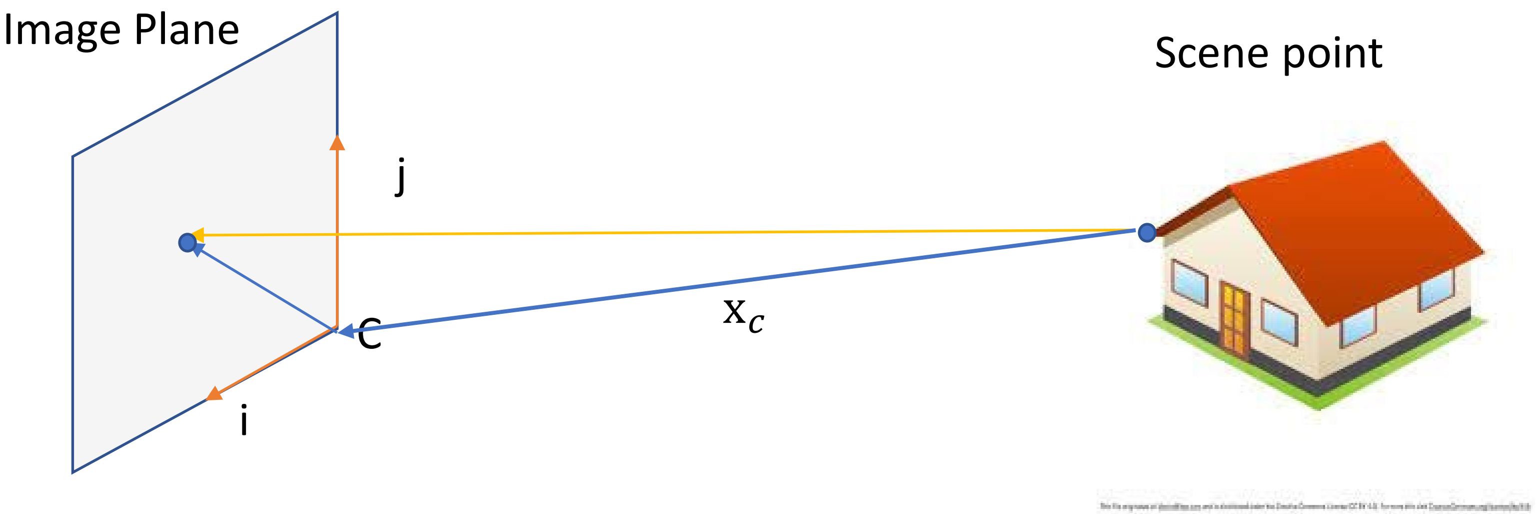


Projection along the z direction

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

Slide from L. Lazebnik.

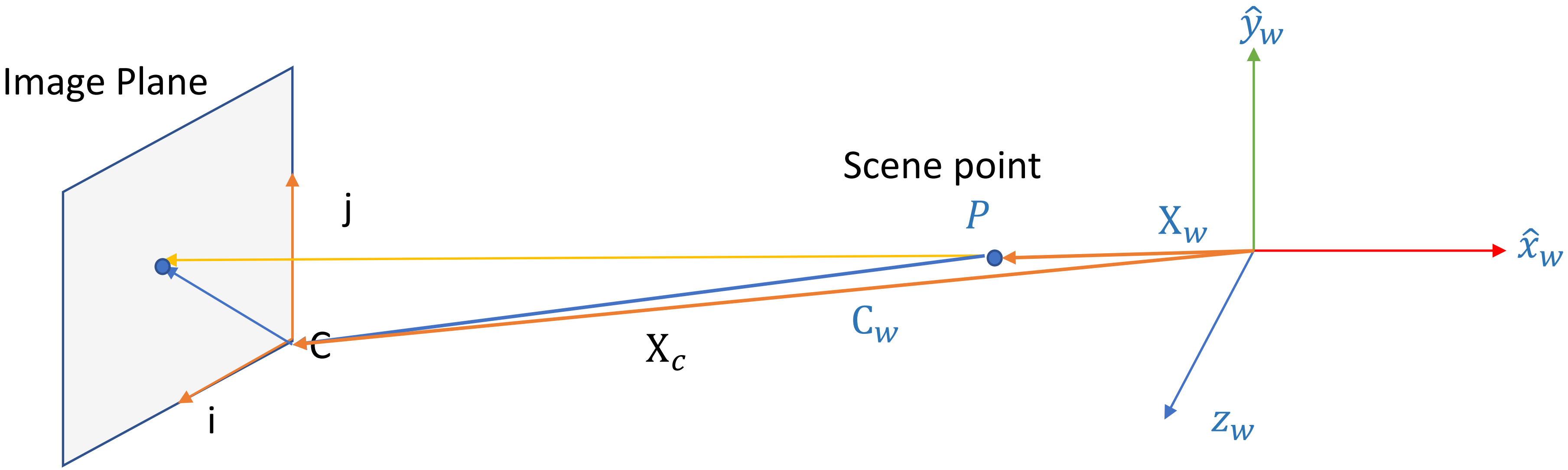
From 3D to 2D: Orthographic Projection



$$u = \mathrm{i} \cdot \mathrm{x}_c = \mathrm{i}^{\mathrm{T}} \mathrm{x}_c$$
$$v = \mathrm{j} \cdot \mathrm{x}_c = \mathrm{j}^{\mathrm{T}} \mathrm{x}_c$$

When the distance of scene from the camera centre is large compared to the depth of the object, we can use orthographic project for approximation

From 3D to 2D: Orthographic Projection

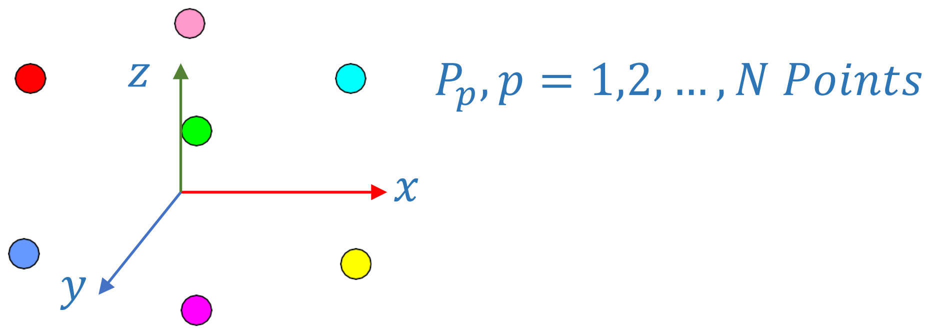


$$u = \mathrm{i} \cdot X_c = \mathrm{i}^T X_c = \mathrm{i}^T (P - C)$$
$$v = \mathrm{j} \cdot X_c = \mathrm{j}^T X_c = \mathrm{j}^T (P - C)$$

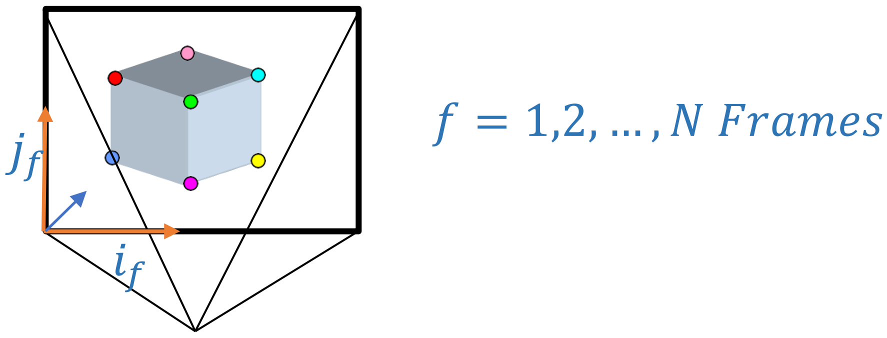
$$\boxed{\begin{aligned} u &= \mathrm{i}^T (P - C) \\ v &= \mathrm{j}^T (P - C) \end{aligned}}$$

Orthographic Structure from motion

Given: a set of corresponding *2D image points* $\{u_{f,p}, v_{f,p}\}$ in two or more images



Compute: the camera parameters and the ***Scene Point*** $\{P_p\}$, Camera position $\{C_f\}$, camera orientation $\{i_f, j_f\}$



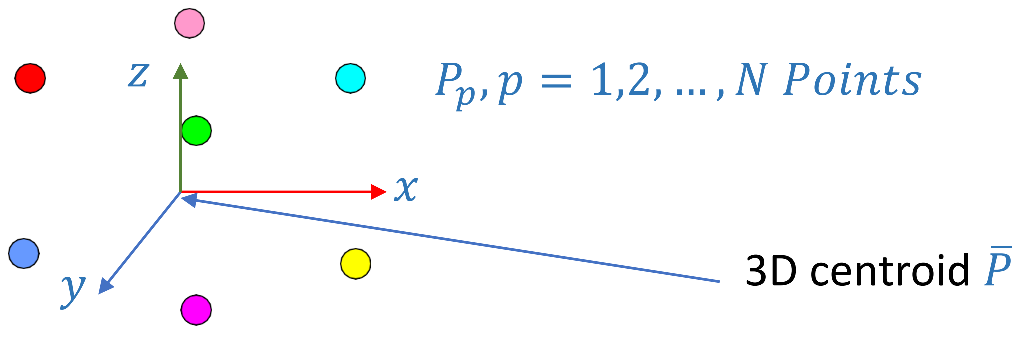
Orthographic Structure from motion

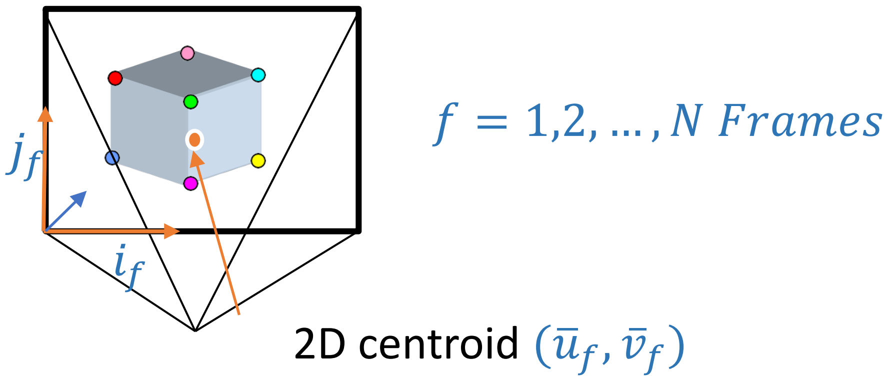
Image of point P_p in camera frame f :

$$u_{f,p} = \mathrm{i}_f^{\mathrm{T}} \left(P_p - C_f\right)$$
$$v_{f,p} = \mathrm{j}_f^{\mathrm{T}} \left(P_p - C_f\right)$$

Assume origin of world at centroid of the scene point

$$\frac{1}{N}\sum_{p=1}^{N} P_p = \bar{P} = 0$$





How to eliminate the centroid?

Centroid $(\bar{u}_f, \bar{v}_f)$ of the image pints in frame f :

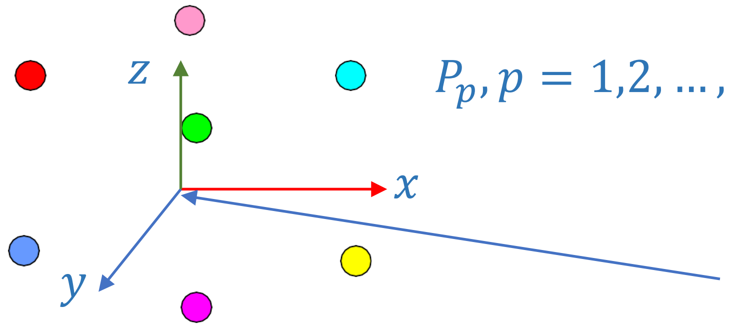
$$\bar{u}_f = \frac{1}{N}\sum_{p=1}^{N} u_{f,p} = \frac{1}{N}\sum_{p=1}^{N} \mathrm{i}_f^{\mathrm{T}}\,(P_p - C_f)$$

$$\bar{u}_f = \frac{1}{N}\mathrm{i}_f^{\mathrm{T}}\sum_{p=1}^{N} P_p - \frac{1}{N}\mathrm{i}_f^{\mathrm{T}}\sum_{p=1}^{N} C_f$$

$$\Rightarrow \quad \boxed{\bar{u}_f = \mathrm{i}_f^{\mathrm{T}} C_f}$$

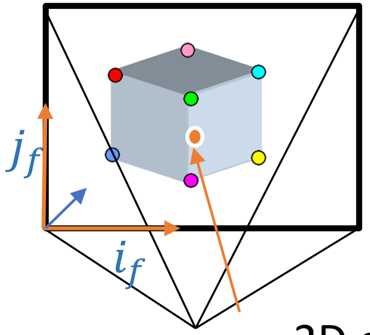
Similarly we have

$$\boxed{\bar{v}_f = \mathrm{j}_f^{\mathrm{T}} C_f}$$



$P_p, p = 1,2,\ldots,N$ *Points*

3D centroid $\bar{P}$



$f = 1,2,\ldots,N$ *Frames*

2D centroid $(\bar{u}_f, \bar{v}_f)$

How to eliminate the centroid?

Shift the camera origin to the centroid $(\bar{u}_f, \bar{v}_f)$

Image points with respect to $(\bar{u}_f, \bar{v}_f)$

$$\tilde{u}_{f,p} = u_{f,p} - \bar{u}_f$$

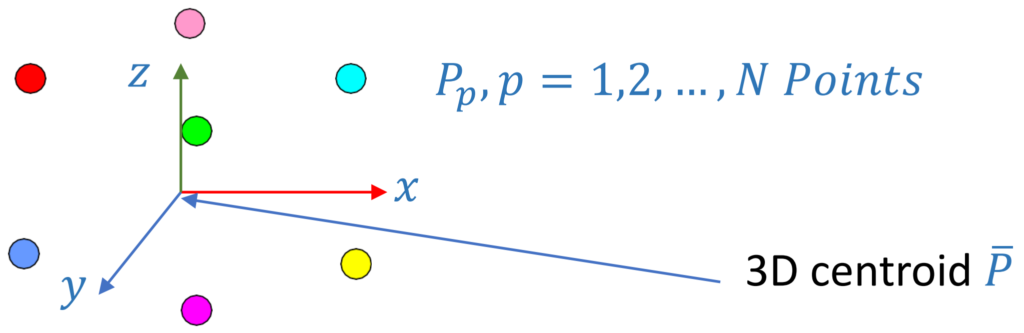
$$\tilde{u}_{f,p} = \mathrm{i}_f^{\mathrm{T}} (P_p - C_f) - \mathrm{i}_f^{\mathrm{T}} C_f$$

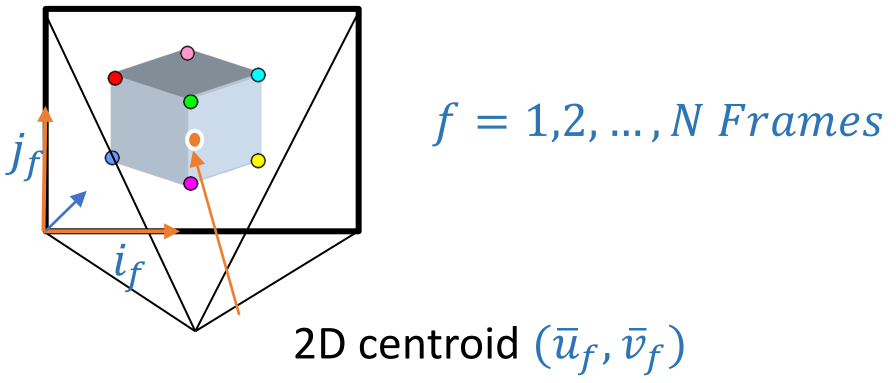
$$\tilde{u}_{f,p} = \mathrm{i}_f^{\mathrm{T}} P_p$$

Similarly

$$\tilde{v}_{f,p} = \mathrm{j}_f^{\mathrm{T}} P_p$$

Camera location removed from the equations





Observation matrix W

We have

$$\tilde{u}_{f,p} = \mathrm{i}_f^{\mathrm{T}} P_p$$
$$\tilde{v}_{f,p} = \mathrm{j}_f^{\mathrm{T}} P_p$$

In matrix form

$$\begin{bmatrix} \tilde{u}_{f,p} \\ \tilde{v}_{f,p} \end{bmatrix} = \begin{bmatrix} \mathrm{i}_f^{\mathrm{T}} \\ \mathrm{j}_f^{\mathrm{T}} \end{bmatrix} P_p$$

- Putting all frame and points we have

$$\begin{bmatrix} \tilde{u}_{1,1} & \tilde{u}_{1,2} & \dots & \tilde{u}_{1,N} \\ \tilde{u}_{2,1} & \tilde{u}_{2,2} & \dots & \tilde{u}_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{u}_{F,1} & \tilde{u}_{F,2} & \dots & \tilde{u}_{F,N} \\ \tilde{v}_{1,1} & \tilde{v}_{1,2} & \tilde{v}_{1,3} & \tilde{v}_{1,4} \\ \tilde{v}_{2,1} & \tilde{v}_{2,2} & \tilde{v}_{2,3} & \tilde{v}_{2,4} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{v}_{F,1} & \tilde{v}_{F,2} & \tilde{v}_{F,3} & \tilde{v}_{F,4} \end{bmatrix} = \begin{bmatrix} i_1^T \\ i_2^T \\ \vdots \\ i_F^T \\ j_1^T \\ j_2^T \\ \vdots \\ j_F^T \end{bmatrix} \begin{bmatrix} P_1 & P_2 & \dots & P_N \end{bmatrix}$$

$S_{3\times N}$
Scene Structure
(UnKnown)

$W_{2F\times N}$
Centroid-subtracted
image points (Known)

$M_{2F\times 3}$
Camera Motion
(UnKnown)

Rank of Observation Matrix

$$W = M \times S$$

$2F \times N \qquad 2F \times 3 \qquad 3 \times N$

Therefore

$$Rank(W) = Rank(M \times S) \leq \min(3, N, 2F)$$

Since N and $2F$ always >3, we can assume

$$Rank(W) \leq 3$$

Using SVD

- $W = U\Sigma V^T$

$$SVD(W) = [U] \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix} [V^T]$$

$2F \times 2F$ $\qquad$ $2F \times N$ $\qquad$ $N \times N$

Using SVD

- $W = U\Sigma V^T$

$$SVD(W) = [\underset{3}{U_1} \quad | \quad \underset{2F-3}{U_2}] \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix} \begin{array}{l} \begin{bmatrix} V_1^T \\ - \\ V_2^T \end{bmatrix} \end{array} \begin{array}{l} 3 \\ \\ N-3 \end{array}$$

$$2F \times 2F \qquad\qquad 2F \times 3 \qquad\qquad N \times N$$

$$\boxed{\begin{array}{c} W = U_1 \Sigma V_1^T \\ (2F \times 3)(3 \times 3)(3 \times P) \end{array}}$$

Factorization (Finding M, S)

The observation matrix is

$$W = U_1 \Sigma_1 V_1^T$$

$$W = U_1 (\Sigma_1)^{1/2} (\Sigma_1)^{1/2} \Sigma_1$$

M?

S?

The decomposition is not unique, we can put any 3x3 non-singular matrix Q

$$W = \underbrace{\boxed{U_1 (\Sigma_1)^{1/2} Q}}_{=M} \underbrace{\boxed{Q^{-1} (\Sigma_1)^{1/2} \Sigma_1}}_{=S}$$

If we solve Q

Solving Q

- The motion matrix M is

$$M = \begin{bmatrix} i_1^T \\ i_2^T \\ \vdots \\ i_F^T \\ j_1^T \\ j_2^T \\ \vdots \\ j_F^T \end{bmatrix} = U_1\ (\Sigma_1)^{1/2} Q = \begin{bmatrix} i_1^T Q \\ i_2^T Q \\ \vdots \\ i_F^T Q \\ j_1^T Q \\ j_2^T Q \\ \vdots \\ j_F^T Q \end{bmatrix}$$

- The orthonormal Constraints:

- $i_f \cdot i_f = i_f^T \cdot i_f = 1$
- $j_f \cdot j_f = j_f^T \cdot j_f = 1$
- $i_f \cdot j_f = i_f^T \cdot j = 0$

- Therefore
- $i_f^T Q Q^T i_f = 1$
- $j_f^T Q Q^T j_f = 1$
- $i_f^T Q Q^T j_f = 0$

Solving Q

- When have for each frame 3 equations (where Q is unknown)

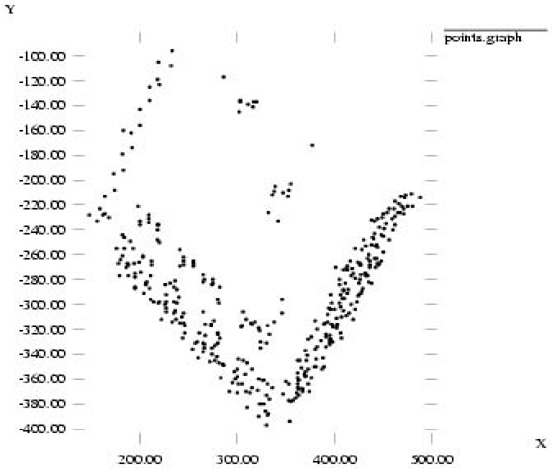
$$i_f^T Q Q^T i_f = 1$$
$$j_f^T Q Q^T j_f = 1$$
$$i_f^T Q Q^T j_f = 0$$

- Q is 3x3 matrix, 9 variables, For F frame, we have3F quadratic equations
- Q can be solved with 3 or more images using Newton's Method.
- Final Solution
  - $M = U_1(\Sigma_1)^{1/2}Q$
  - $S = Q^{-1}Q\,(\Sigma_1)^{1/2}V^T$

Reconstruction results

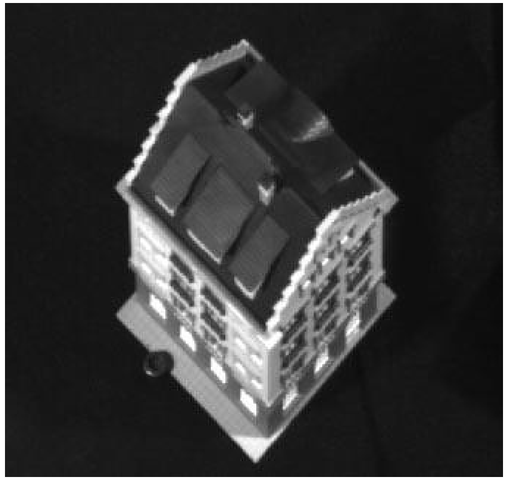
1

60



120

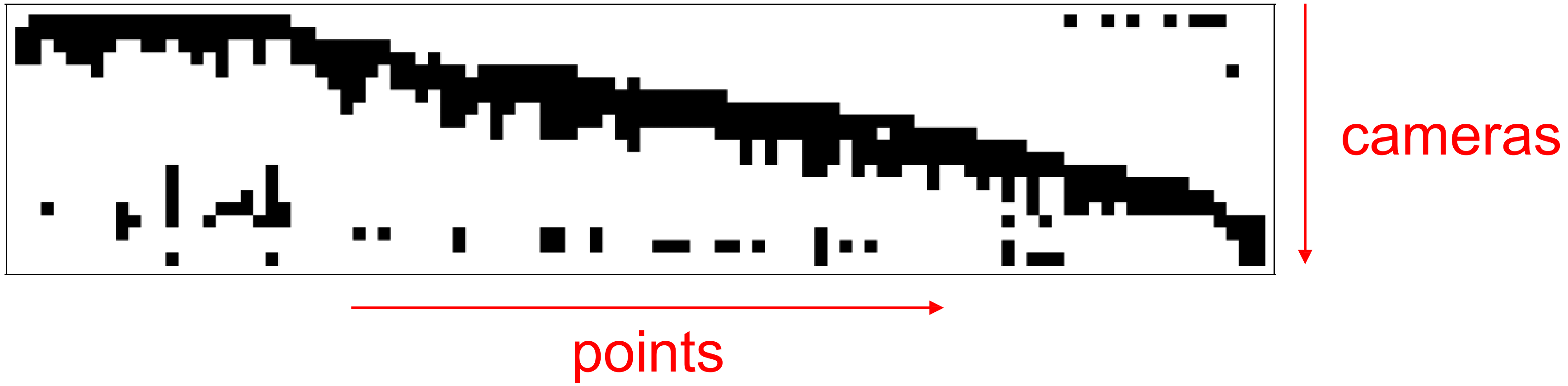
150



C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Dealing with missing data

- So far, we have assumed that all points are visible in all views
- In reality, the measurement matrix typically looks something like this:

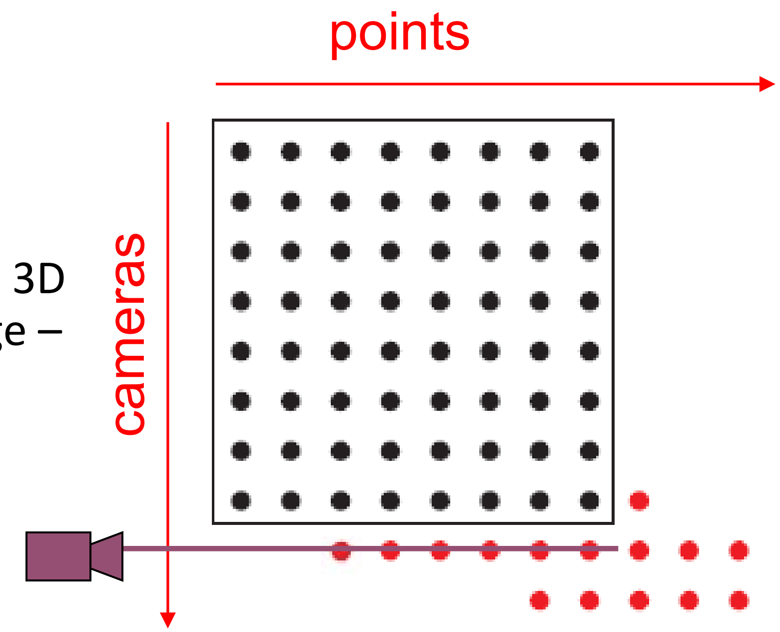


Sequential structure from motion

- Initialize motion from two images using fundamental matrix

- Initialize structure by triangulation

- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*



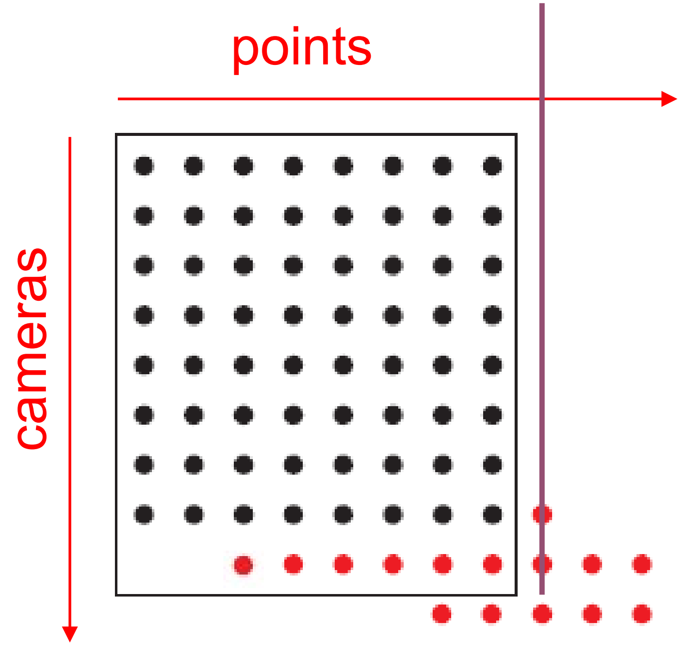
Sequential structure from motion

- Initialize motion from two images using fundamental matrix

- Initialize structure by triangulation

- For each additional view:
  - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*

  - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*



Sequential structure from motion

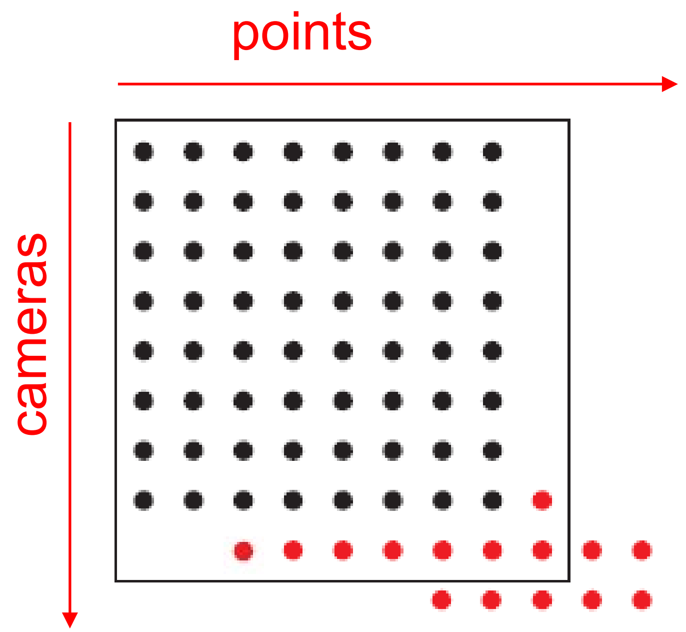
- Initialize motion from two images using fundamental matrix

- Initialize structure by triangulation

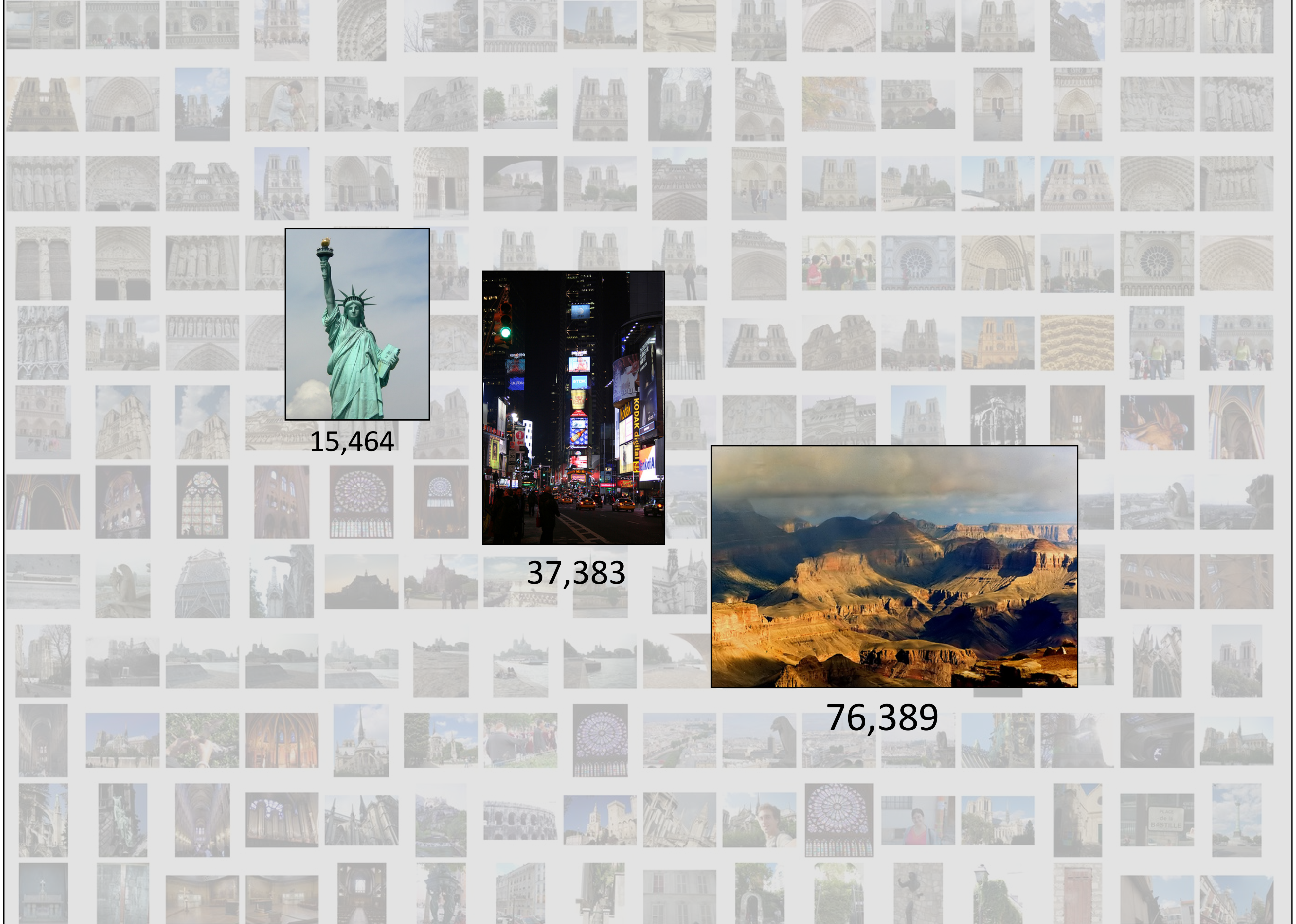
- For each additional view:
    - Determine projection matrix of new camera using all the known 3D points that are visible in its image – *calibration*

    - Refine and extend structure: compute new 3D points, re-optimize existing points that are also seen by this camera – *triangulation*

- Refine structure and motion: bundle adjustment



Large-scale structure from motion


15,464
37,383
76,389

Standard way to view photos

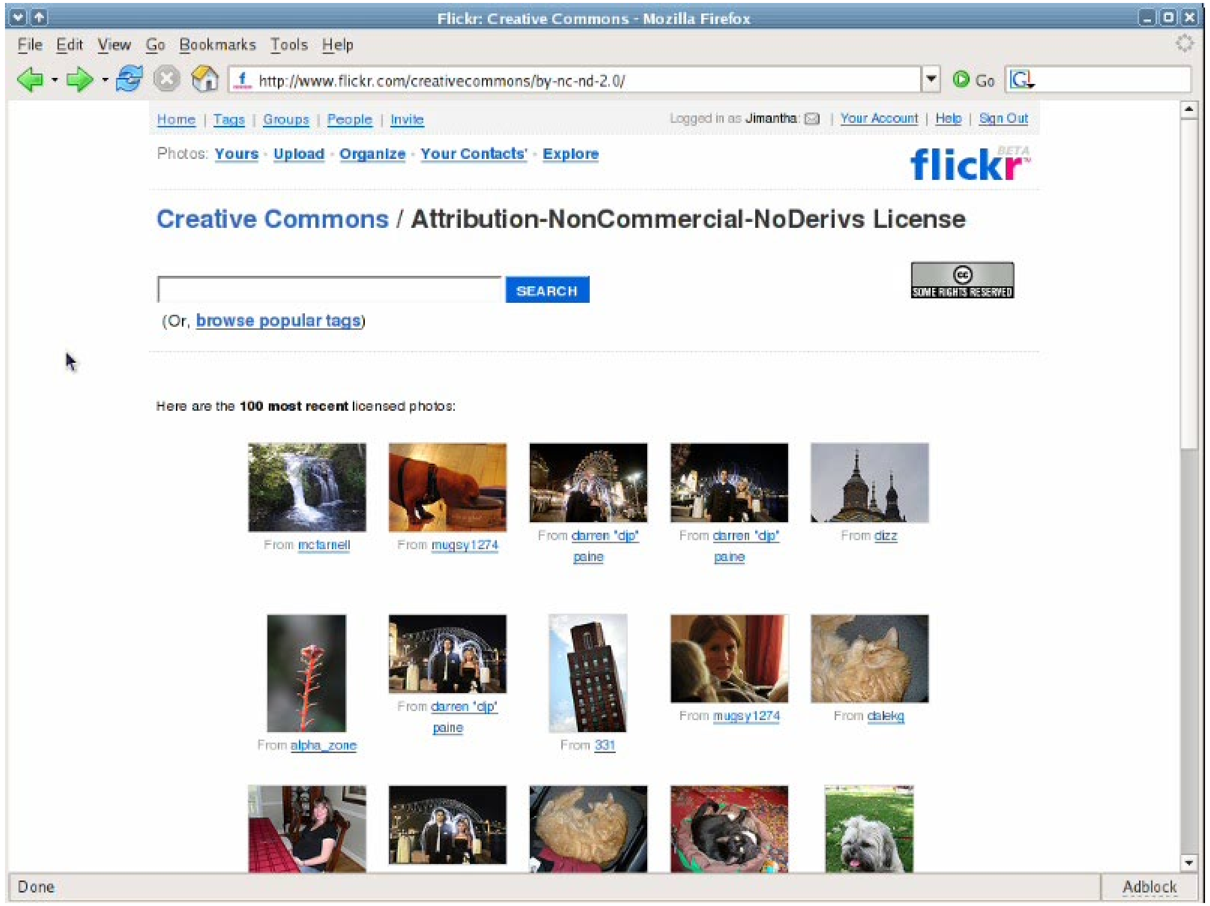
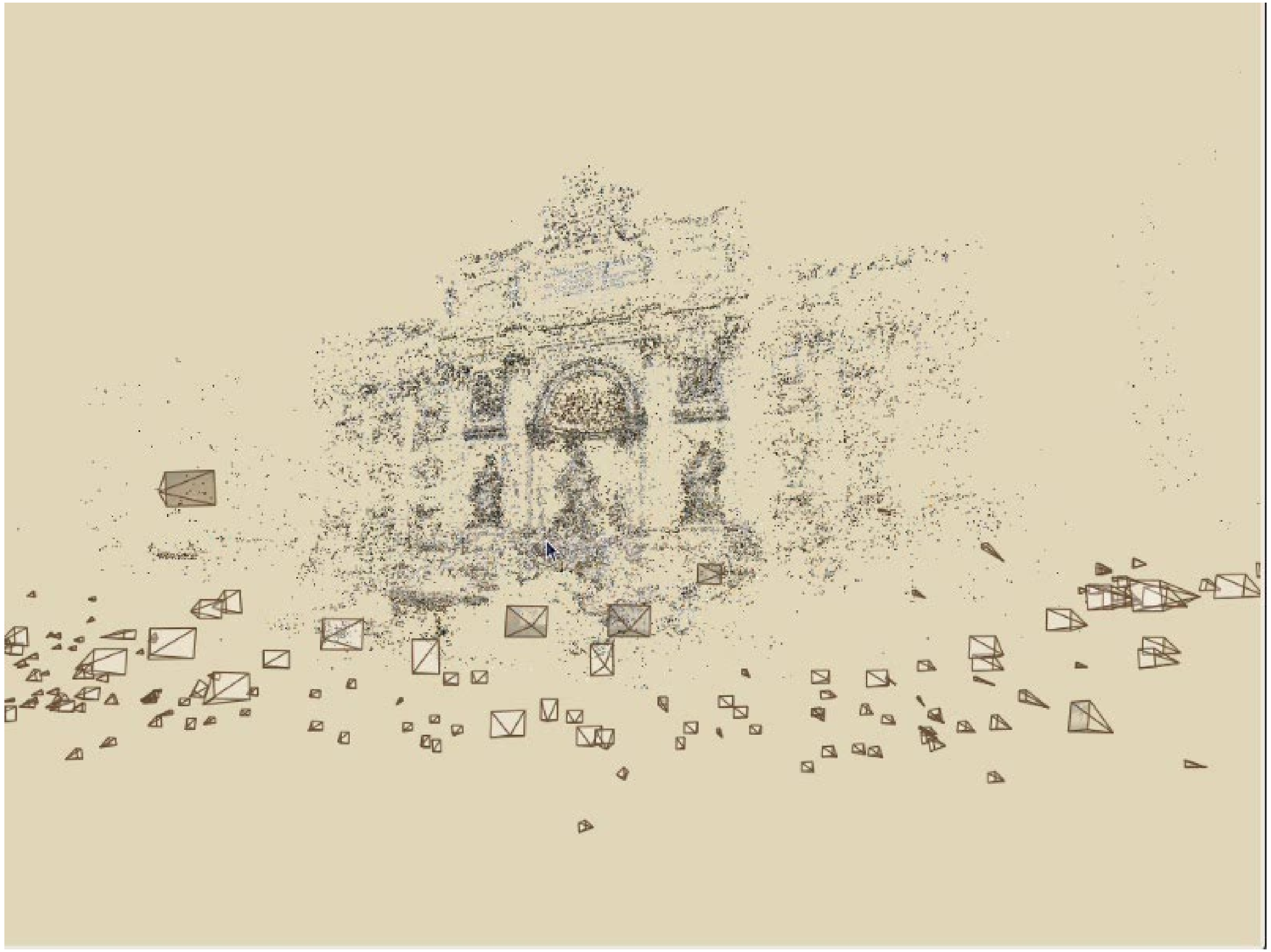
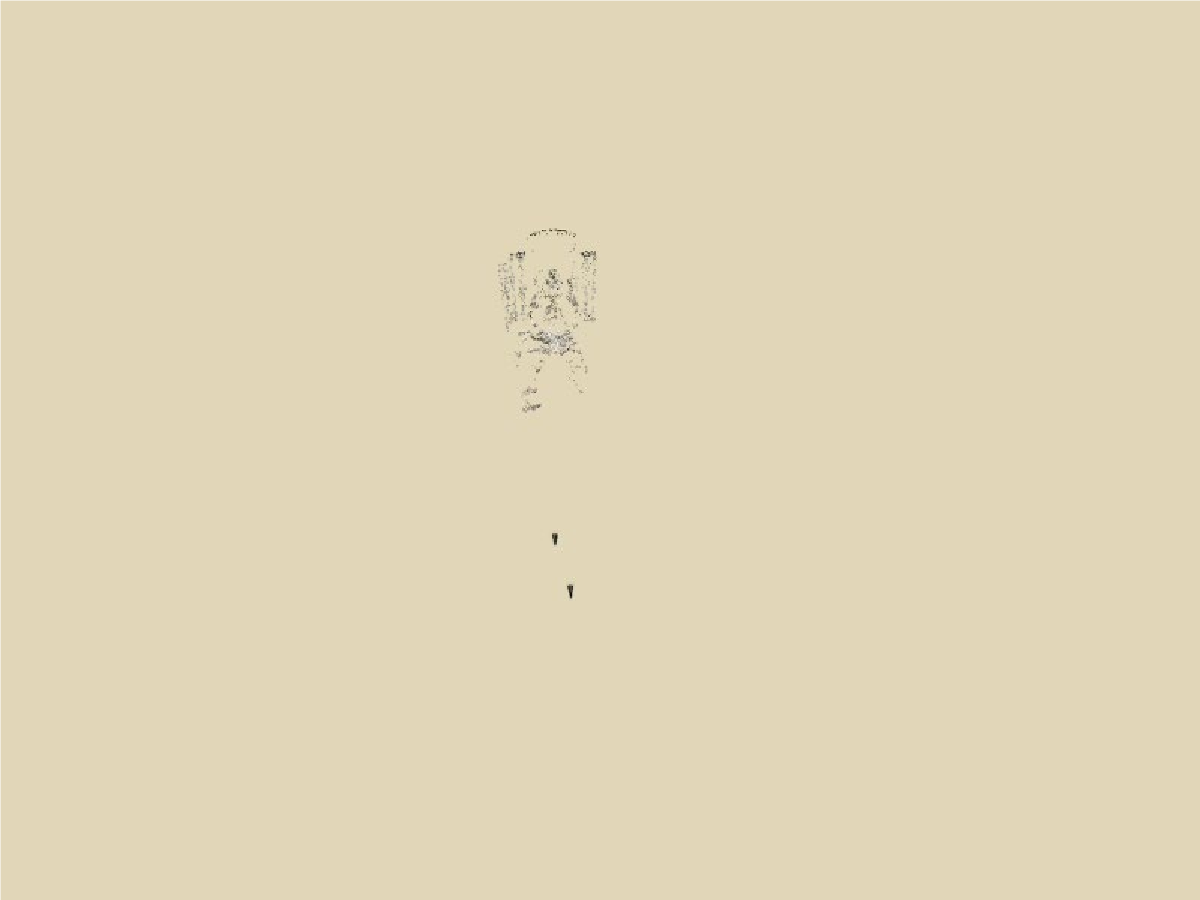


Photo Tourism



Slide Source: Ioannis (Yannis) Gkioulekas

Incremental structure from motion



Slide Source: Ioannis (Yannis) Gkioulekas

Final reconstruction

Slide Source: Ioannis (Yannis) Gkioulekas

More examples

Slide Source: Ioannis (Yannis) Gkioulekas

More examples

[Slide Source: Ioannis (Yannis) Gkioulekas]

More examples

Slide Source: Ioannis (Yannis) Gkioulekas

SFM Softwares

- Bundler
- OpenSfM
- OpenMVG
- VisualSFM
- See also Wikipedia's list of toolboxes

Slide from L. Lazebnik.

Reference

- Richard Szeliski, Computer Vision: Algorithms and Applications, Springer 2010, Chapter 7