Computer Vision

---Structure from motion

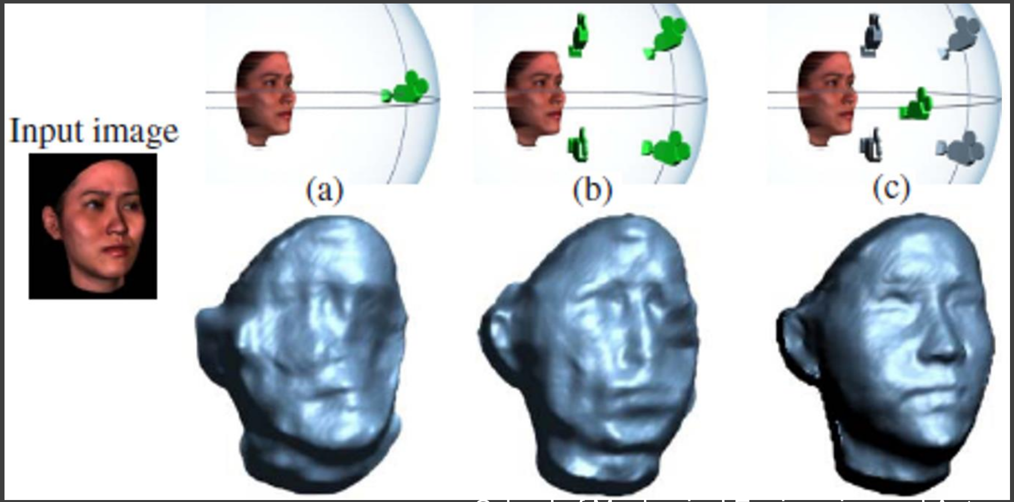
Dr. WU Xiaojun

2020.11.06

3D Reconstruction

- One image: 2D-to-3D reconstruction method
  - Difficult and with ambiguity

  - Using prior knowledge (e.g. face)



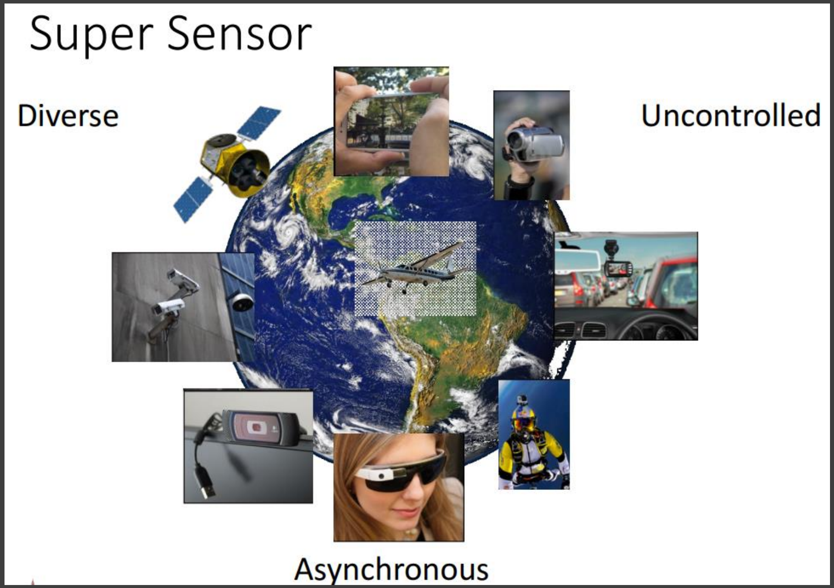
http://www.wisdom.weizmann.ac.il/~ronen/papers/Hassner Basri - Example Based 3D Reconstruction from Single 2D Images.pdf

3D Reconstruction

- Two images: 2D-to-3D reconstruction method
  - Basic idea of stereo vision
  - Stereo reconstruction by epipolar geometry
    - Stereo camera pair calibration (find Fundamental matrix F)
    - Construct the 3D (graphic) model from 2 images

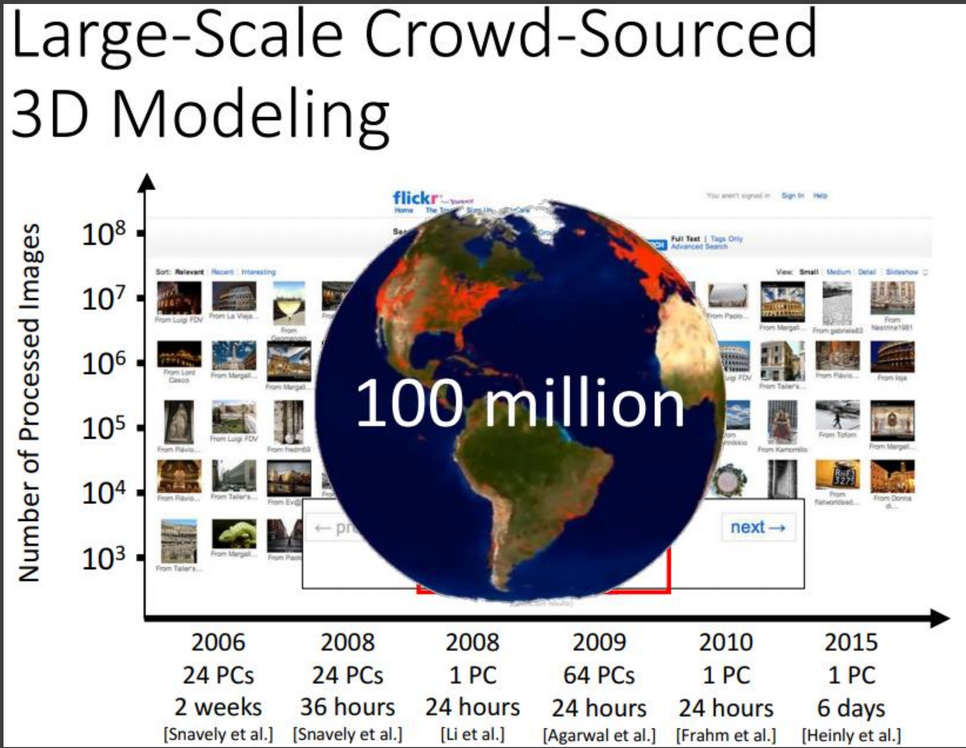
3D Reconstruction

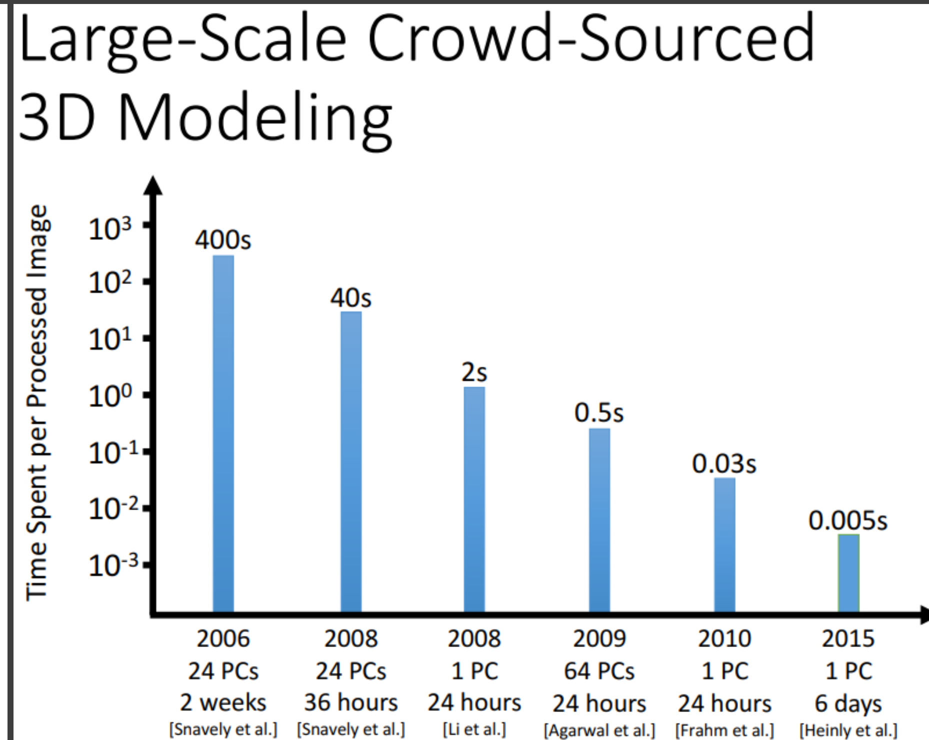
- M images: 2D-to-3D reconstruction method



3D Reconstruction

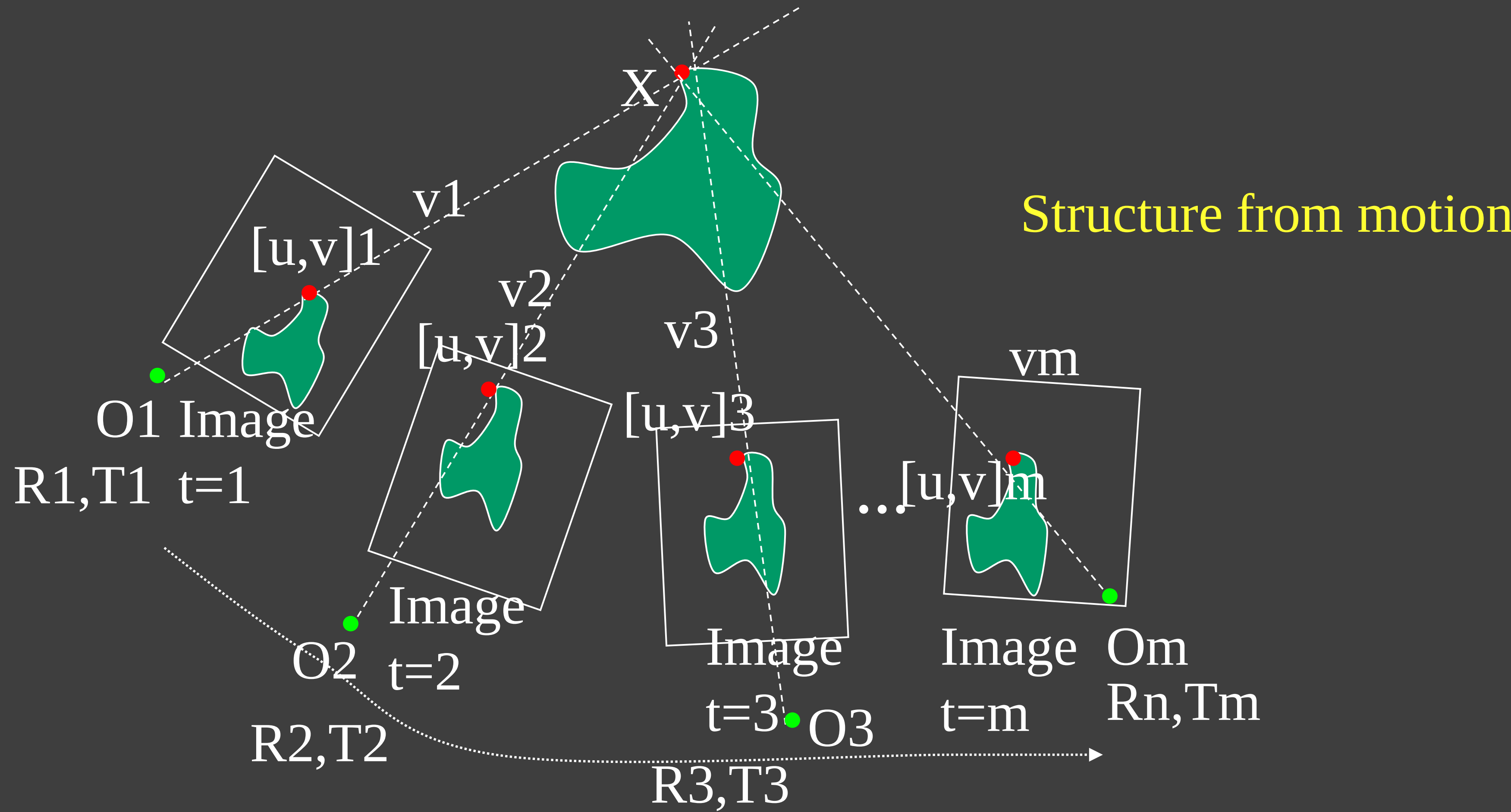
- M images: 2D-to-3D reconstruction method





3D Reconstruction

- M images: 2D-to-3D reconstruction method

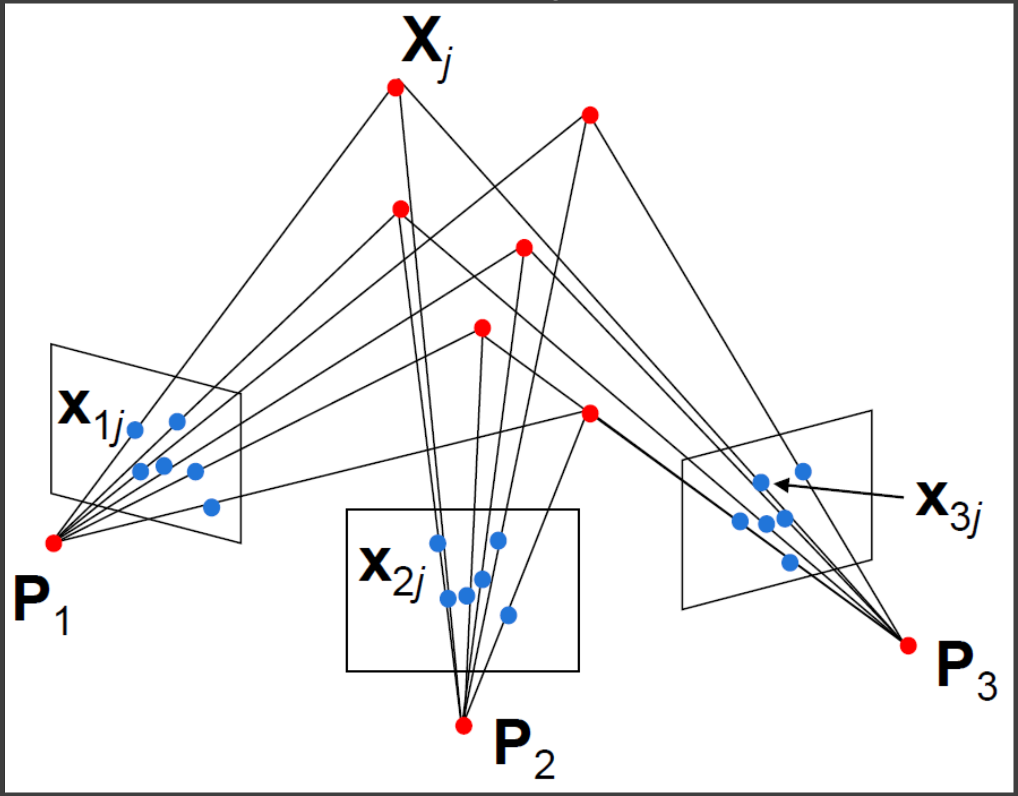


Structure from motion

- Given: m images of n fixed 3D points

$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j \qquad i = 1, \cdots, m.\ j = 1, \cdots, n$$

- Problem: estimate m projection matrices $\mathbf{P}_i$ and n 3D points X_j from the mn correspondences $\boldsymbol{x}_{ij}$



Structure from motion

- Structure from motion ambiguity
- If we scale the entire scene by some factor k and, at the same time, scale the camera matrices by the factor of $1/k$, the projections of the scene points in the image remain exactly the same:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\frac{1}{k}\mathbf{P}\right)(k\mathbf{X})$$

It is impossible to recover the absolute scale of the scene!

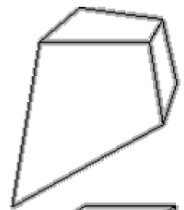
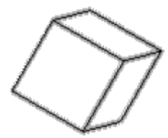
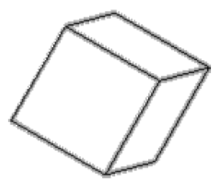
Structure from motion

- Structure from motion ambiguity
- If we scale the entire scene by some factor k and, at the same time, scale the camera matrices by the factor of $1/k$, the projections of the scene points in the image remain exactly the same:
- More generally: if we transform the scene using a transformation $\mathbf{Q}$ and apply the inverse transformation to the camera matrices, then the images do not change.

$$\boldsymbol{x} = \mathbf{P}\mathbf{X} = (\mathbf{P}\mathbf{Q}^{-1})(\mathbf{Q}\mathbf{X})$$

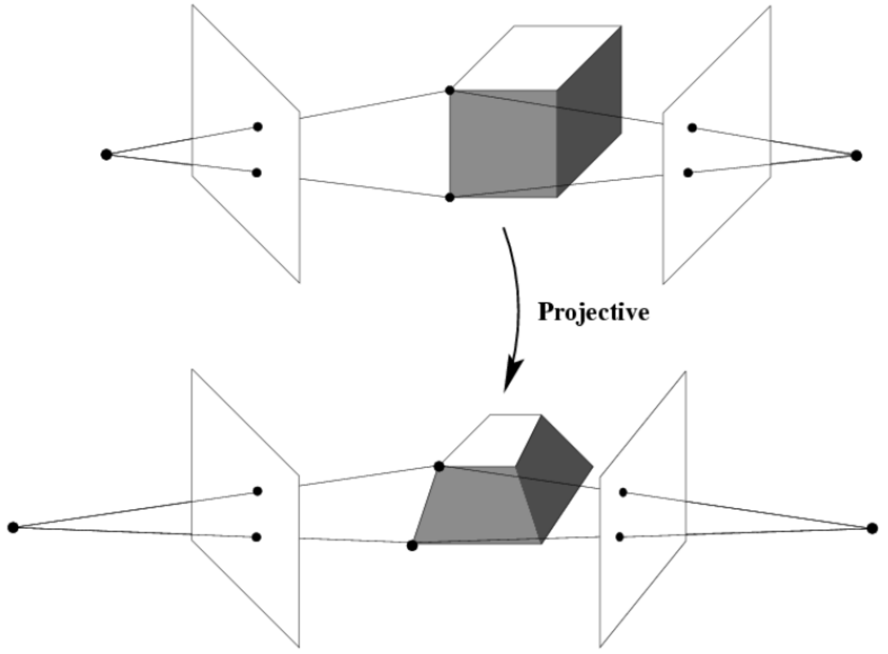
Structure from motion

- Structure from motion ambiguity
- Types of ambiguity

Projective 15dof	$\begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$		Preserves intersection and tangency
Affine 12dof	$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$		Preserves parallellism, volume ratios
Similarity 7dof	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$		Preserves angles, ratios of length
Euclidean 6dof	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$		Preserves angles, lengths

Structure from motion

- Structure from motion ambiguity
- Types of ambiguity---Projective ambiguity

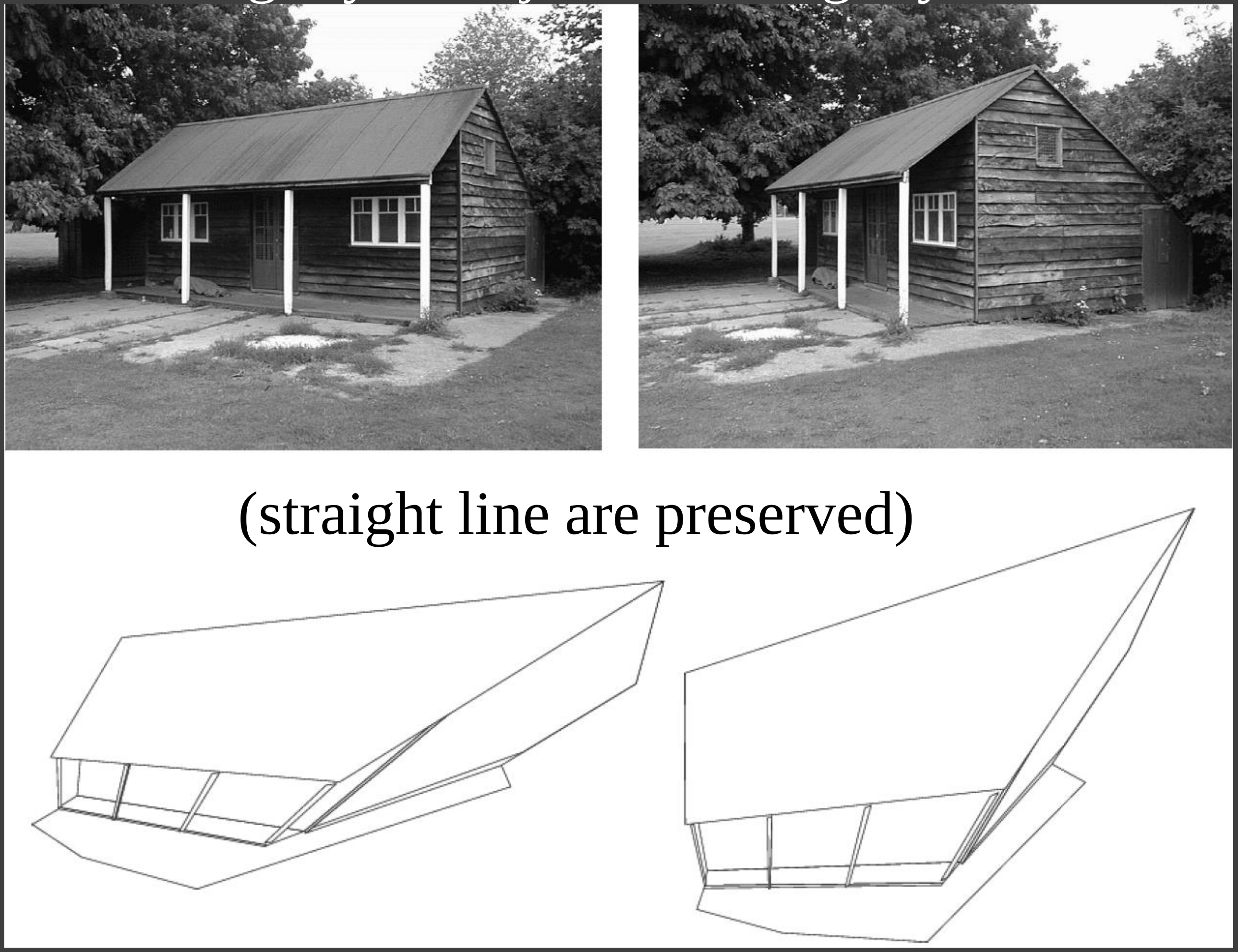


$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_{\mathrm{P}}^{-1}\right)\left(\mathbf{Q}_{\mathrm{P}}\mathbf{X}\right)$$

$$\mathbf{Q}_{\mathbf{p}} = \begin{bmatrix} \mathrm{A} & \mathrm{t} \\ \mathrm{v}^{\mathrm{T}} & v \end{bmatrix}$$

Structure from motion

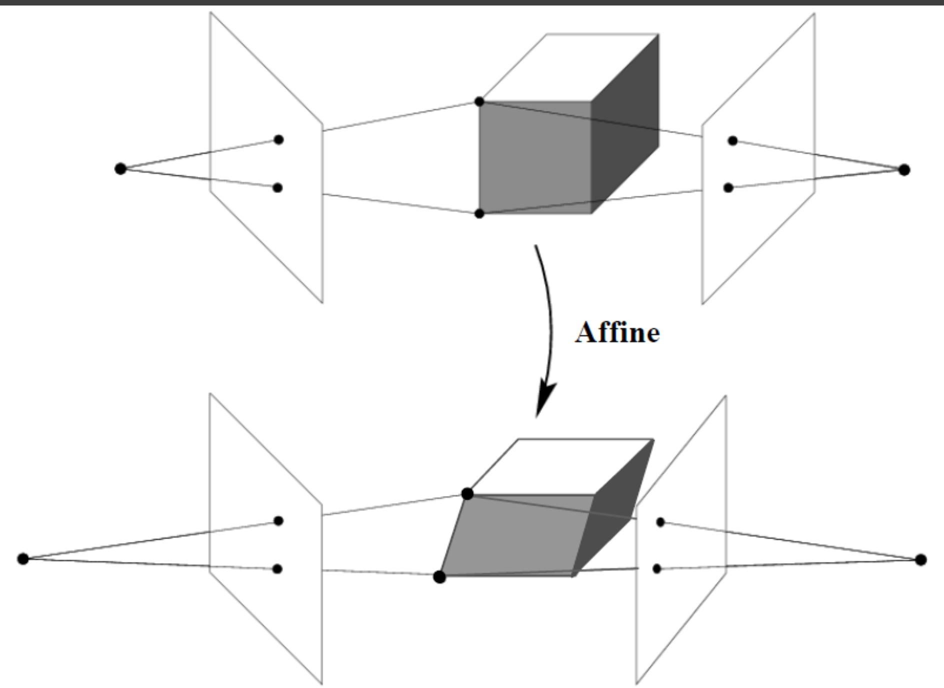
- Structure from motion ambiguity
- Types of ambiguity---Projective ambiguity



(straight line are preserved)

Structure from motion

- Structure from motion ambiguity
- Types of ambiguity---Affine ambiguity

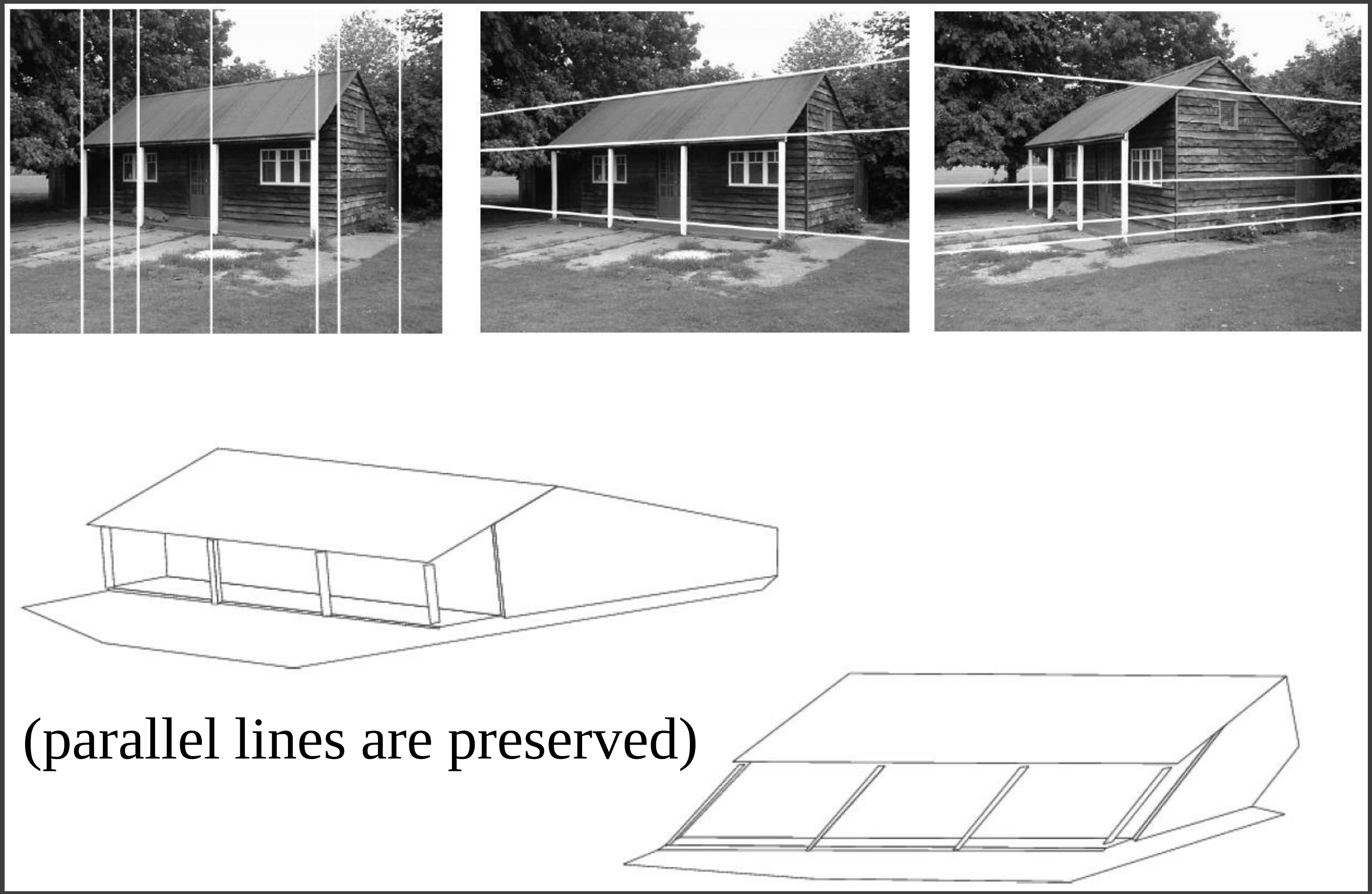


$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_A^{-1}\right)\left(\mathbf{Q}_A\mathbf{X}\right)$$

$$\mathbf{Q}_A = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$$

Structure from motion

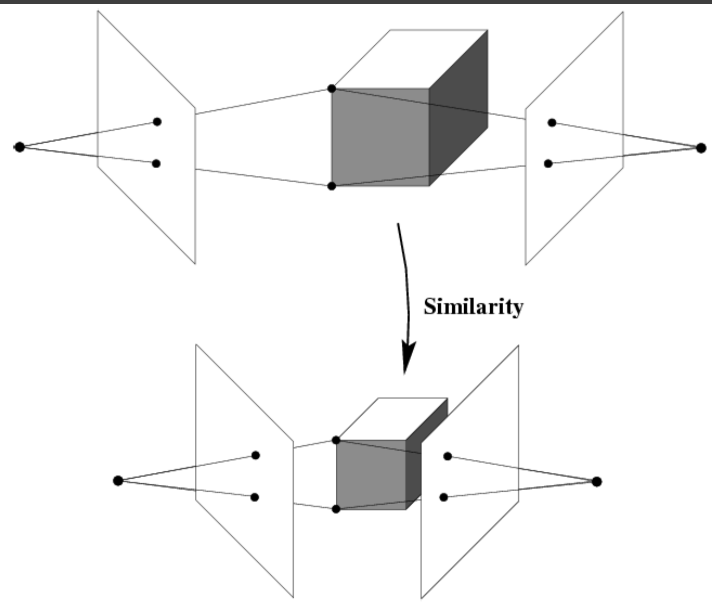
- Structure from motion ambiguity
- Types of ambiguity---Affine ambiguity



(parallel lines are preserved)

Structure from motion

- Structure from motion ambiguity
- Types of ambiguity---Similarity ambiguity

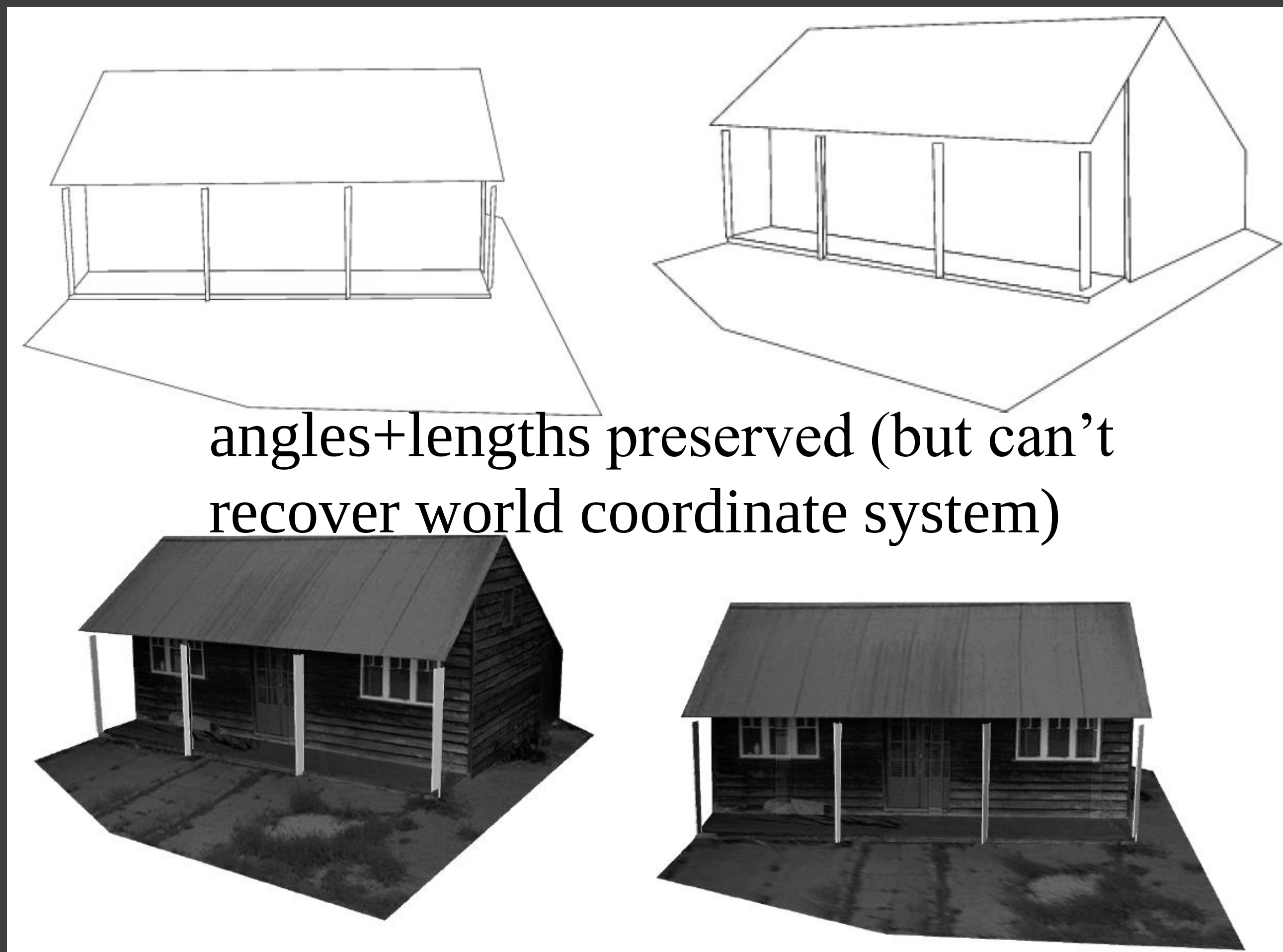


$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\mathbf{P}\mathbf{Q}_S^{-1}\right)\left(\mathbf{Q}_S\mathbf{X}\right)$$

$$\mathbf{Q}_s = \begin{bmatrix} s\mathrm{R} & \mathrm{t} \\ 0^{\mathrm{T}} & 1 \end{bmatrix}$$

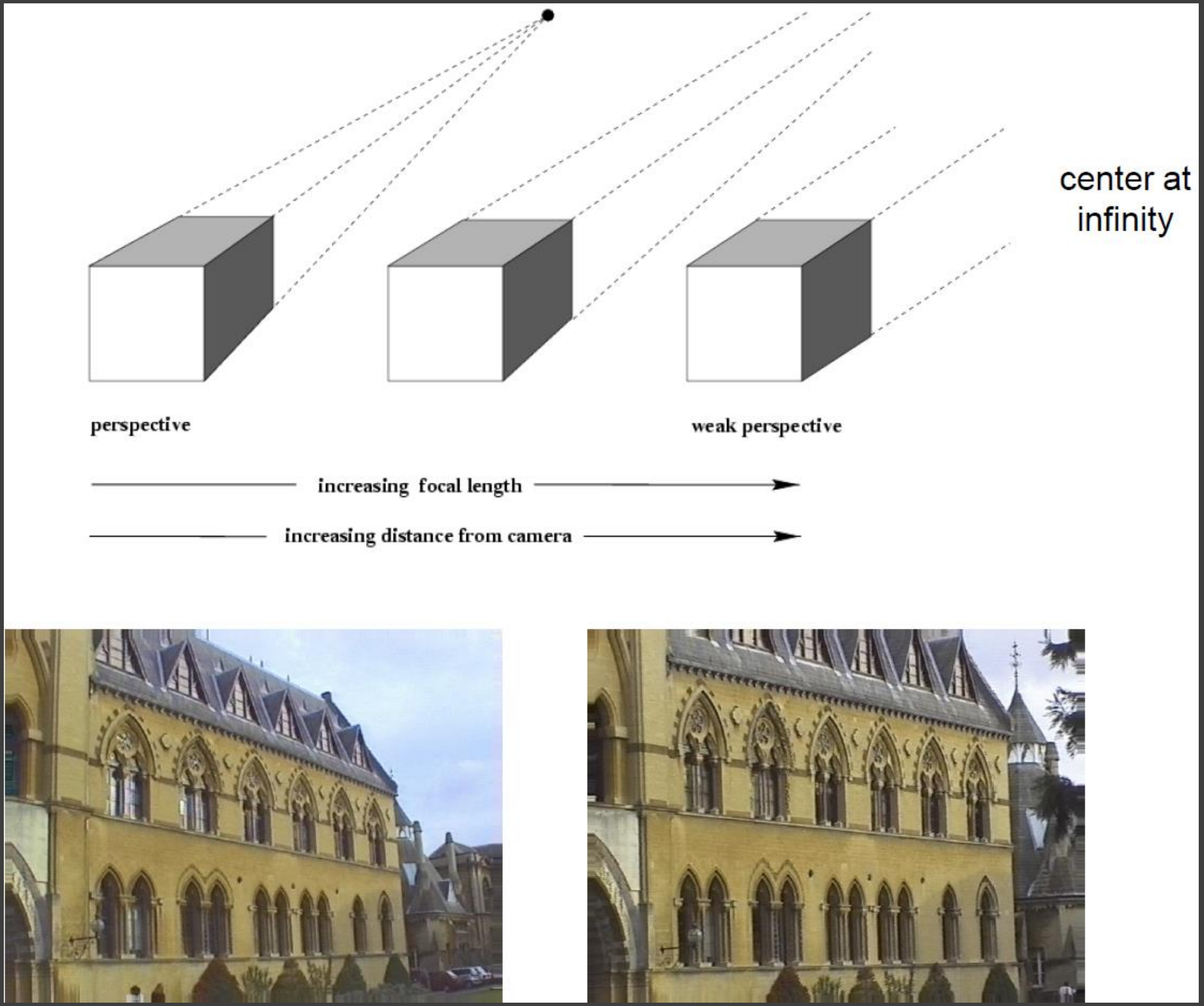
Structure from motion

- Structure from motion ambiguity
- Types of ambiguity---Similarity ambiguity



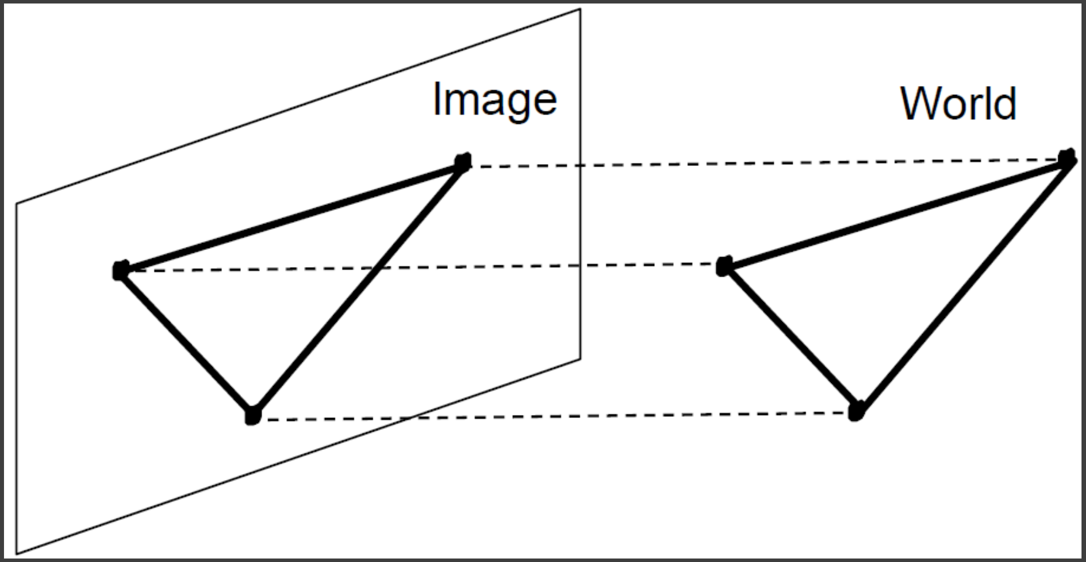
Structure from motion

- Let's start with affine cameras (the math is easier)



Structure from motion

- Recall: Orthographic Projection
- Special case of perspective projection
  - Distance from center of projection to image plane is infinite

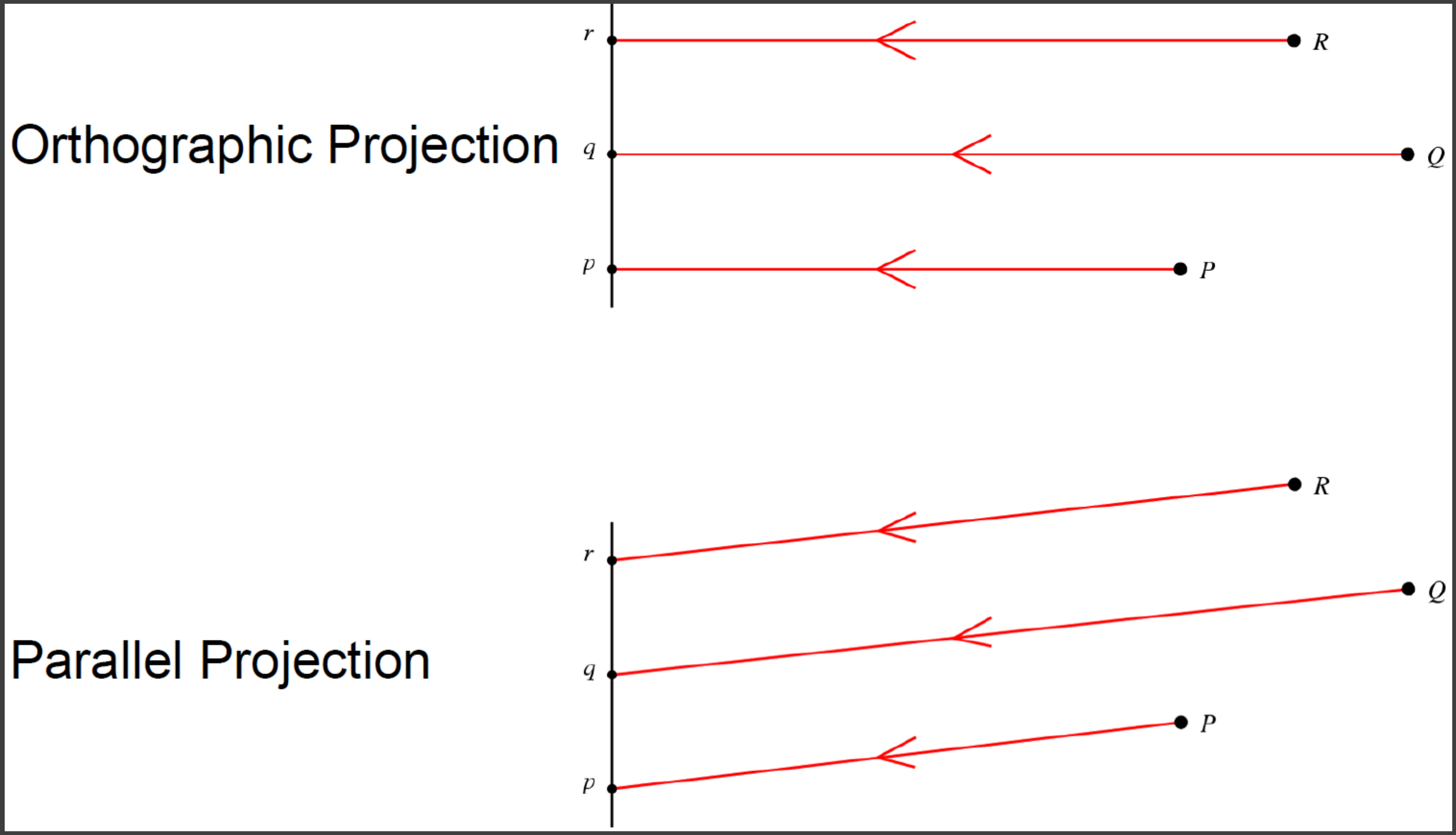


- Projection matrix:

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

Structure from motion

- Affine cameras



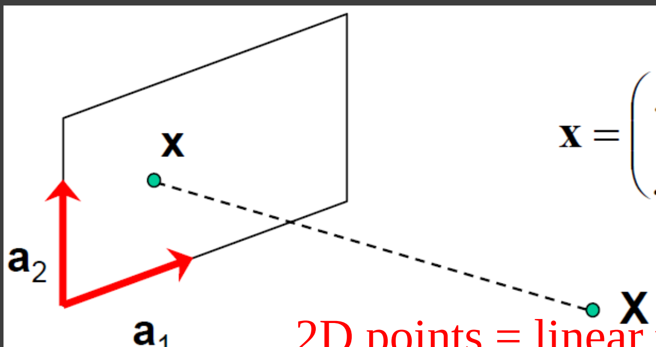
Structure from motion

- Affine cameras
- A general affine camera combines the effects of an affine transformation of the 3D space, orthographic projection, and an affine transformation of the image:

$$\mathbf{P} = [3\times 3\,\text{affine}]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}[4\times 4\,\text{affine}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Affine camera defined by 8 parameters

- Affine projection is a linear mapping + translation in inhomogeneous coordinates



$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \mathbf{AX} + \mathbf{b}$$

Projection of world origin

2D points = linear transformation of 3D points + 2D translation

Affine Structure from motion

- Given: m images of n fixed 3D points:

$$\mathbf{x}_{ij} = \mathbf{A}_i \mathbf{X}_j + \mathbf{b}_i, \qquad i = 1, \cdots, m, \qquad j = 1, \cdots, n$$

- Problem: use the mn correspondences $\mathbf{x}_{ij}$ to estimate m projection matrices $\mathbf{A}_i$ and translation vectors $\mathbf{b}_i$, and n points $\mathbf{X}_j$.
- The reconstruction is defined up to an arbitrary 3D affine transformation $\mathbf{Q}$ (12 degrees of freedom):

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{Q}^{-1}, \qquad \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} \rightarrow \mathbf{Q} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}$$

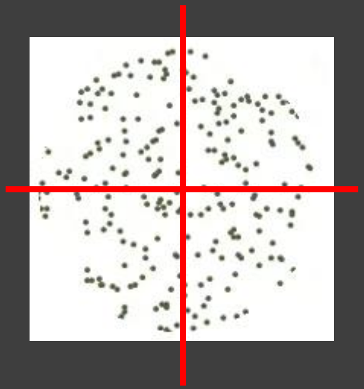
- We have 2mn knowns and 8m + 3n unknowns (minus 12 dof for affine ambiguity)
- Thus, we must have $2mn >= 8m + 3n - 12$
- For two views(m=2), we need four point correspondences(n=4)

Affine Structure from motion

- Centering: subtract the centroid of the image points

$$\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik} = \mathbf{A}_i\mathbf{X}_j + \mathbf{b}_i - \frac{1}{n}\sum_{k=1}^{n}\left(\mathbf{A}_i\mathbf{X}_k + \mathbf{b}_i\right)$$

$$= \mathbf{A}_i\left(\mathbf{X}_j - \frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k\right) = \mathbf{A}_i\hat{\mathbf{X}}_j$$



- For simplicity, assume that the origin of the world coordinate system is at the centroid of the 3D points
- After centering, each normalized point $\mathbf{x}_{ij}$ is related to the 3D point $\mathbf{X}j$ by

$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i\mathbf{X}_j$$

Affine Structure from motion

- Let's create a $2m \times n$ data (measurement) matrix of image points:

$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i \mathbf{X}_j$$

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix}$$

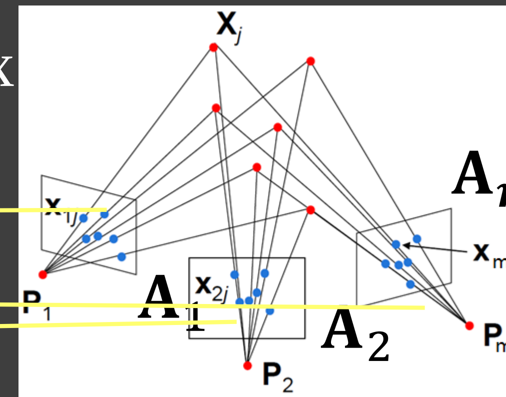
cameras (2m)

Points (n)

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

Affine Structure from motion

- Let's create a $2m \times n$ data (measurement) matrix



$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}$$

points ($3 \times n$)

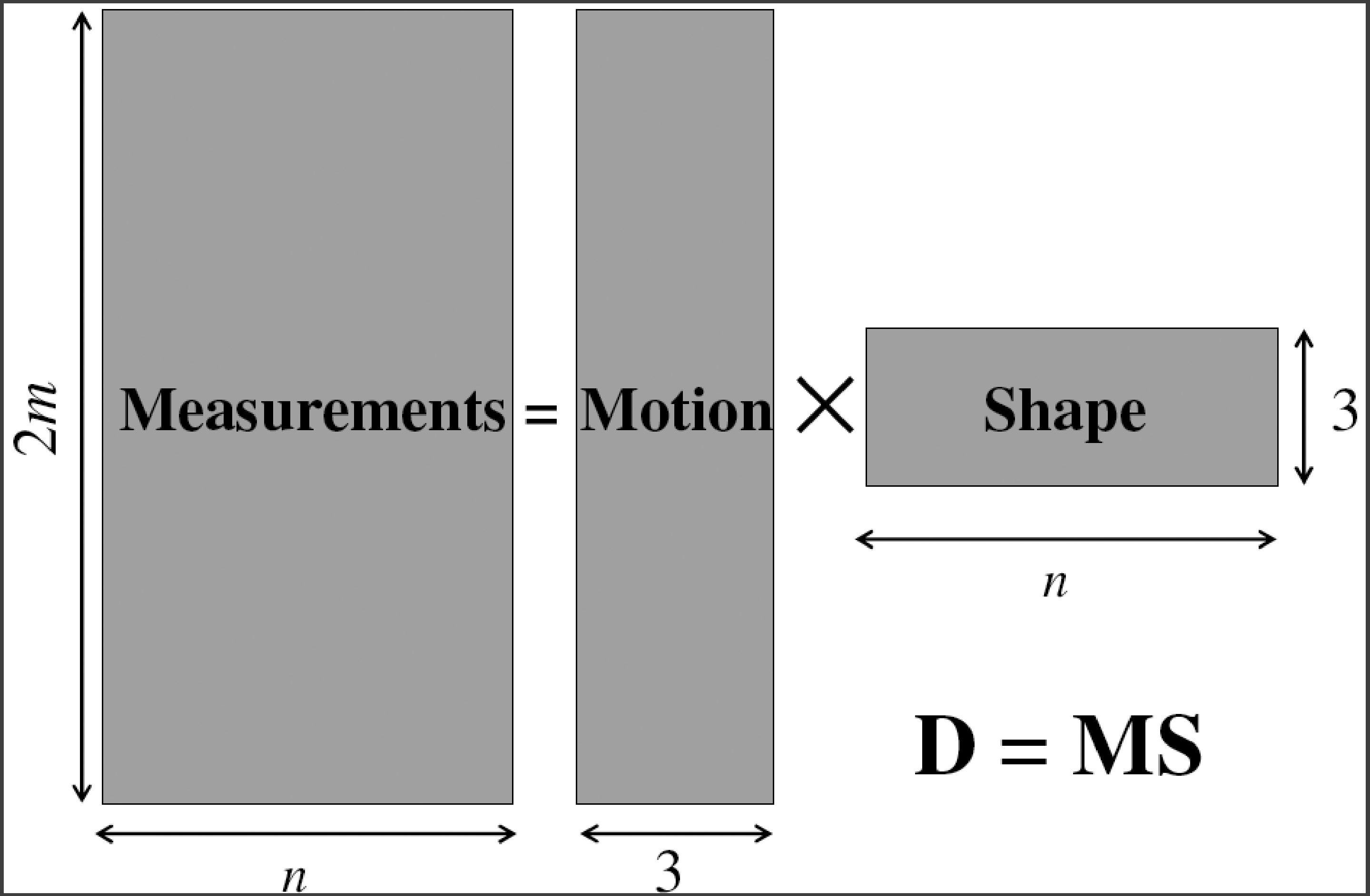
cameras ($2m \times 3$)

$\mathbf{M}$ $\mathbf{S}$

$$\hat{\mathbf{x}}_{ij} = \mathbf{A}_i \mathbf{X}_j$$

Affine Structure from motion

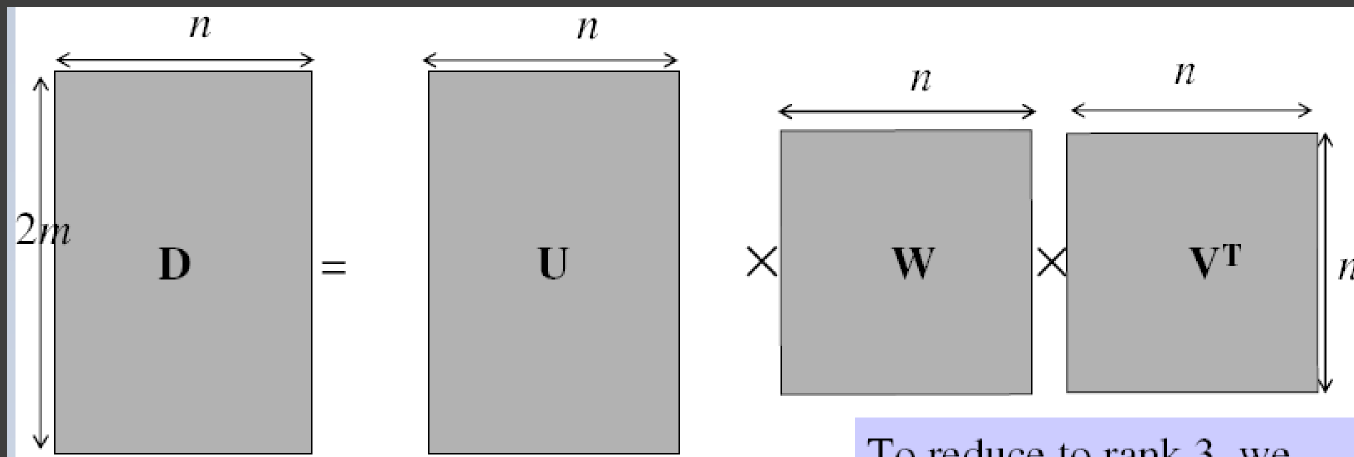
- Factorizing the measurement matrix



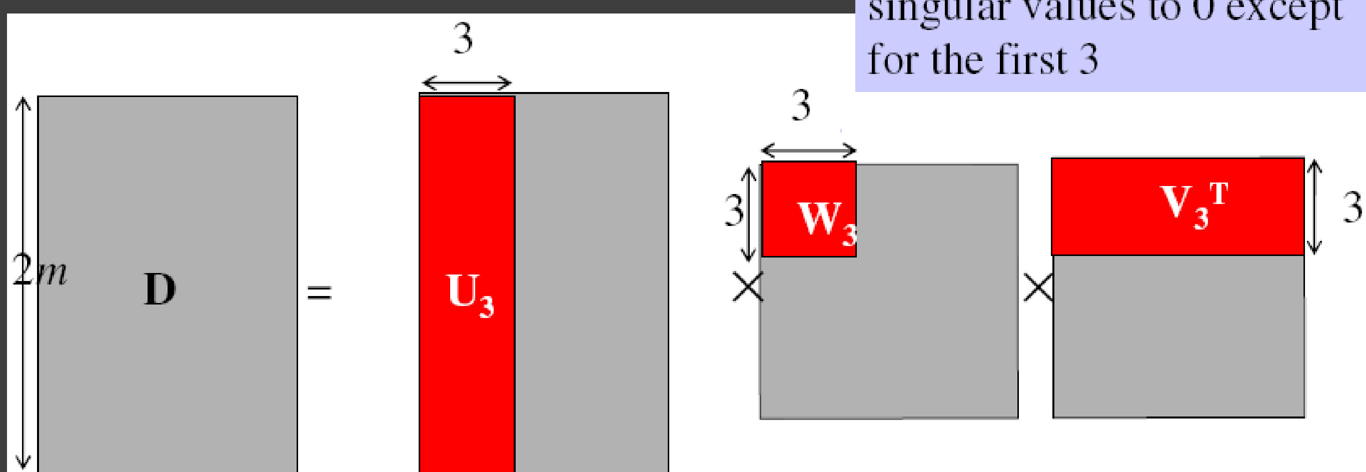
The measurement matrix **D** = **MS** must have rank 3!

Affine Structure from motion

- Factorizing the measurement matrix
- Singular value decomposition of D

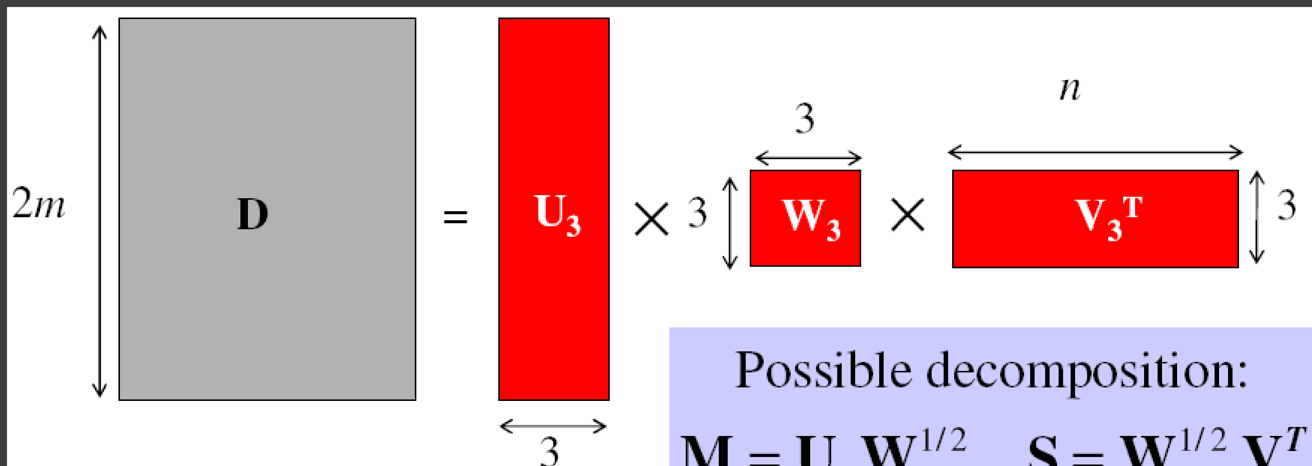


To reduce to rank 3, we just need to set all the singular values to 0 except for the first 3



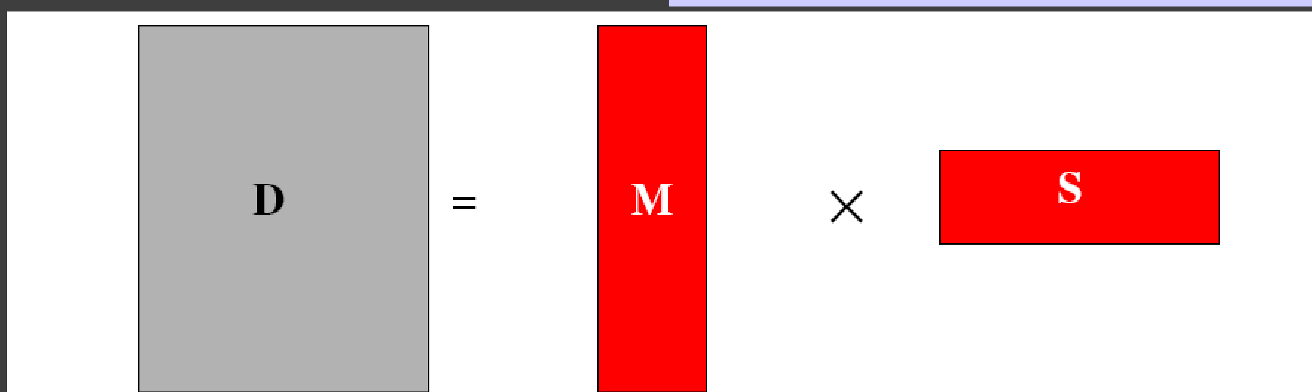
Affine Structure from motion

- Factorizing the measurement matrix
- Obtaining a factorization from SVD:



Possible decomposition:

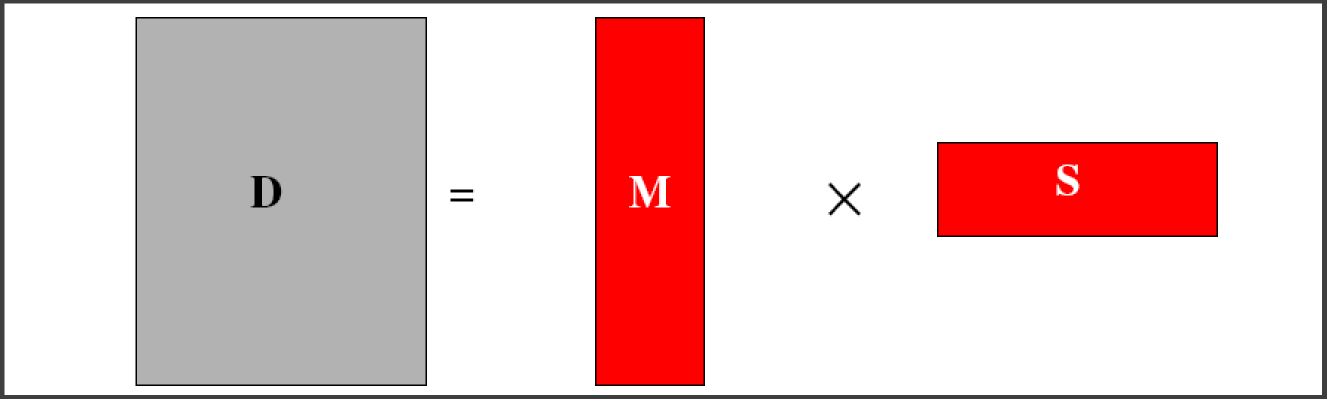
$$\mathbf{M} = \mathbf{U}_3 \mathbf{W}_3^{1/2} \qquad \mathbf{S} = \mathbf{W}_3^{1/2}\,\mathbf{V}_3^{T}$$



This decomposition minimizes $|\mathbf{D}\text{-}\mathbf{MS}|^2$

Affine Structure from motion

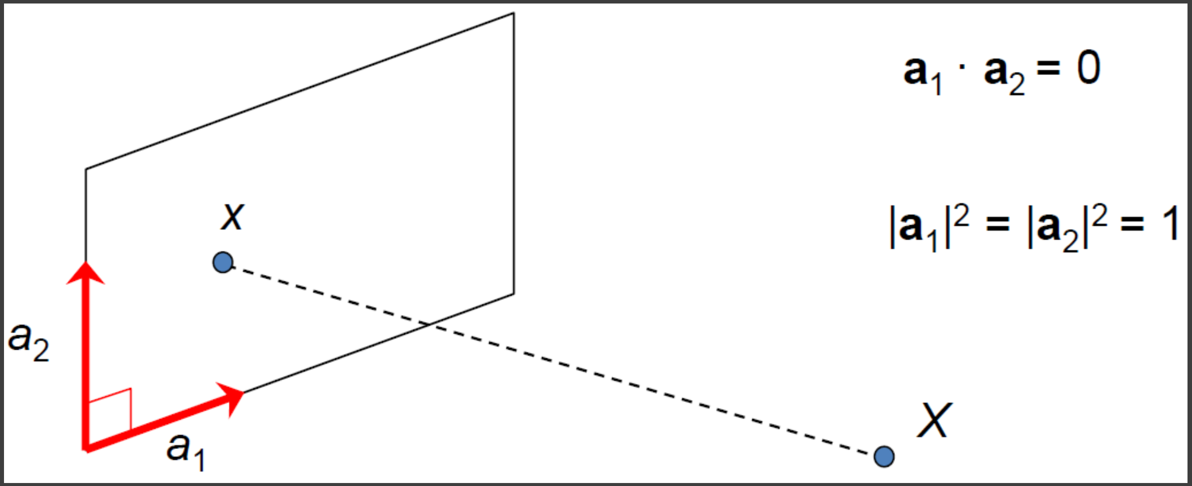
- Affine ambiguity



- The decomposition is not unique. We get the same **D** by using any 3×3 matrix **C** and applying the transformations $\mathbf{M} \rightarrow \mathbf{MC}$, $\mathbf{S} \rightarrow \mathbf{C}^{-1}\mathbf{S}$
- That is because we have only an affine transformation and we have not enforced any Euclidean constraints (like forcing the image axes to be perpendicular, for example)

Affine Structure from motion

- Eliminating the affine ambiguity
- Orthographic: image axes are perpendicular and of unit length



Affine Structure from motion

- Eliminating the affine ambiguity
- Solve for orthographic constraints
  - Three equations for each image *I*

$$\begin{aligned} \tilde{\mathbf{a}}_{i1}^T \mathbf{C}\mathbf{C}^T \tilde{\mathbf{a}}_{i1}^T &= 1 \\ \tilde{\mathbf{a}}_{i2}^T \mathbf{C}\mathbf{C}^T \tilde{\mathbf{a}}_{i2}^T &= 1 \\ \tilde{\mathbf{a}}_{i1}^T \mathbf{C}\mathbf{C}^T \tilde{\mathbf{a}}_{i2}^T &= 0 \end{aligned} \quad \text{where} \quad \tilde{\mathbf{A}}_i = \begin{bmatrix} \tilde{\mathbf{a}}_{i1}^T \\ \tilde{\mathbf{a}}_{i2}^T \end{bmatrix}$$

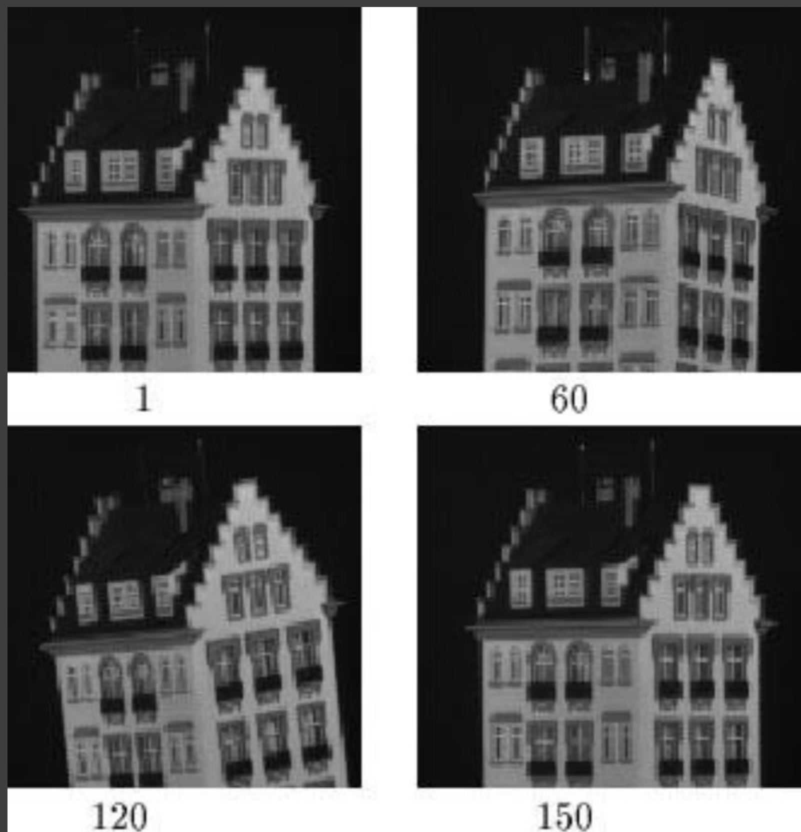
  - Solve for $\mathbf{L} = \mathbf{C}\mathbf{C}^{\mathrm{T}}$
  - Recover $\mathbf{C}$ from $\mathbf{L}$ by Cholesky decomposition: $\mathbf{L} = \mathbf{C}\mathbf{C}^{\mathrm{T}}$
  - Update $\mathbf{M}$ and $\mathbf{S}$: $\mathbf{M} = \mathbf{MC}, \boldsymbol{S} = \mathbf{C}^{-1}\boldsymbol{S}$

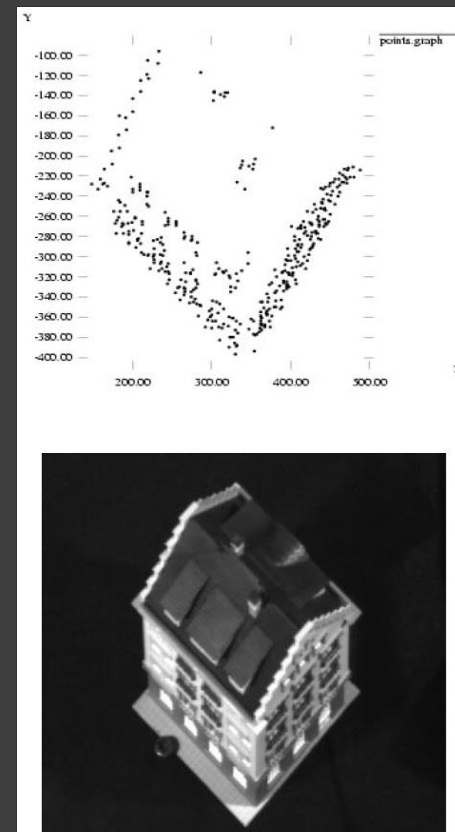
Affine Structure from motion

- Given: m images and n features $\mathbf{x}_{ij}$
- For each image i, center the feature coordinates
- Construct a $2m \times n$ measurement matrix **D**:
  - Column j contains the projection of point j in all views
  - Row i contains one coordinate of the projections of all the n points in image I
- Factorize **D**:
  - Compute SVD: $\mathbf{D} = \mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}}$
  - Create U_3 by taking the first 3 columns of **U**
  - Create V_3 by taking the first 3 columns of **V**
  - Create W_3 by taking the upper left 3×3 block of **W**
- Create the motion and shape matrices:
  - $\mathbf{M} = \mathbf{U}_3\mathbf{W}_3^{1/2}$ and $\mathbf{S} = \mathbf{W}_3^{1/2}\mathbf{V}_3^{\mathrm{T}}$ (or $\mathbf{M} = \mathbf{U}_3$ and $\mathbf{S} = \mathbf{W}_3\mathbf{V}_3^{\mathrm{T}}$)
- Eliminate affine ambiguity

Affine Structure from motion

- Reconstruction results





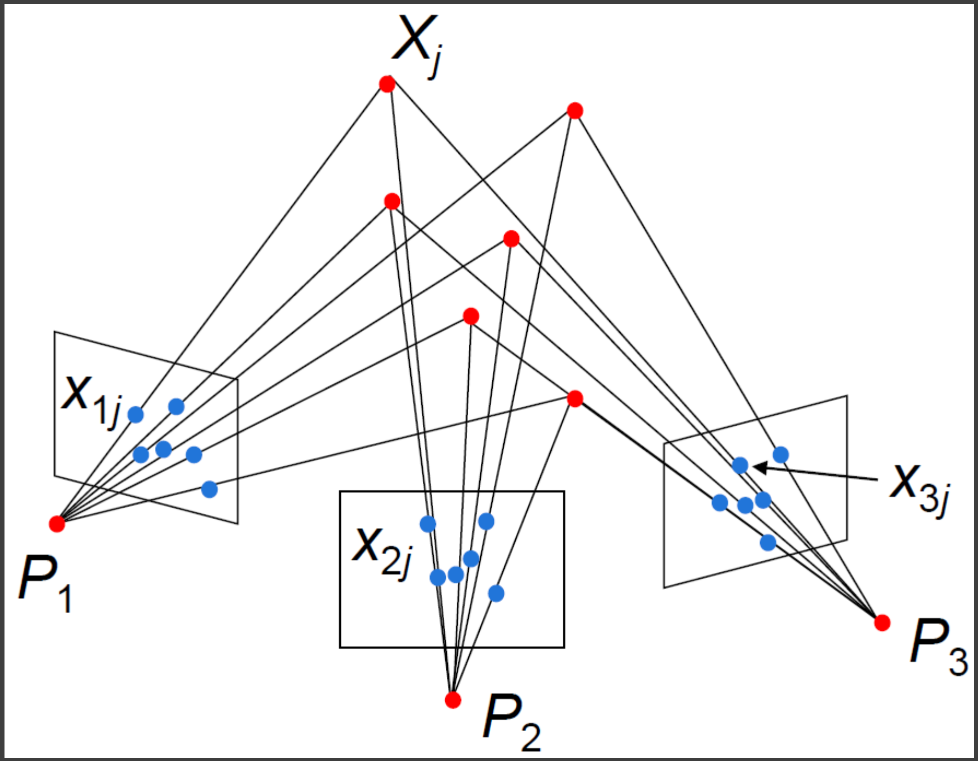
C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137-154, November 1992.

Projective structure from motion

- Given: m images of n fixed 3D points

$$z_{ij}\mathbf{x}_{ij} = \mathbf{P}_i\mathbf{X}_j, \qquad i = 1, \cdots, m, \qquad j = 1, \cdots, n$$

- Problem: estimate m projection matrices $\mathbf{P}_i$ and n 3D points $\mathbf{X}_j$ from the mn correspondences



Projective structure from motion

- Given: *m* images of *n* fixed 3D points

$$z_{ij}\mathbf{x}_{ij} = \mathbf{P}_i\mathbf{X}_j, \qquad i = 1,\cdots,m, \qquad j = 1,\cdots,n$$

- Problem: estimate *m* projection matrices Pi and n 3D points Xj from the *mn* correspondences
- With no calibration info, cameras and points can only be recovered up to a 4x4 projective transformation **Q**:

$$\mathbf{X}\rightarrow\mathbf{QX},\ \mathbf{P}\rightarrow\mathbf{PQ}^{-1}$$

- We can solve for structure and motion when

$$2mn >= 11m + 3n - 15$$

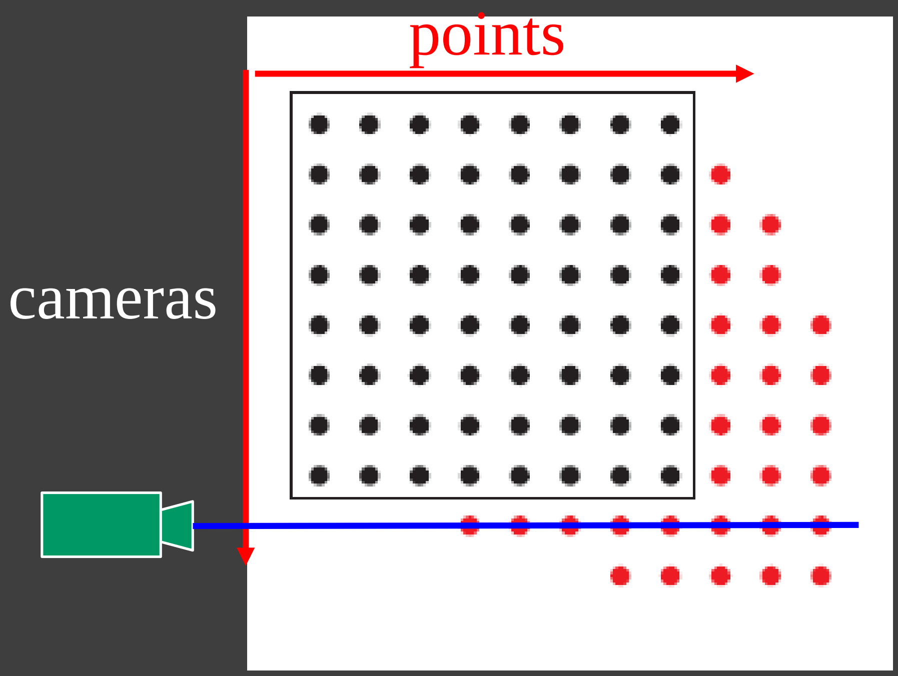
- For two cameras, at least 7 points are needed

Projective SFM: Two-camera case

- Compute fundamental matrix **F** between the two views
- First camera matrix: $[\mathbf{I}|0]$
- Second camera matrix: $[\mathbf{A}|\mathbf{b}]$
- Then **b** is the epipole ($\mathbf{F}^T\mathbf{b} = 0$), $\mathbf{A} = -[\mathbf{b}_\times]\mathbf{F}$

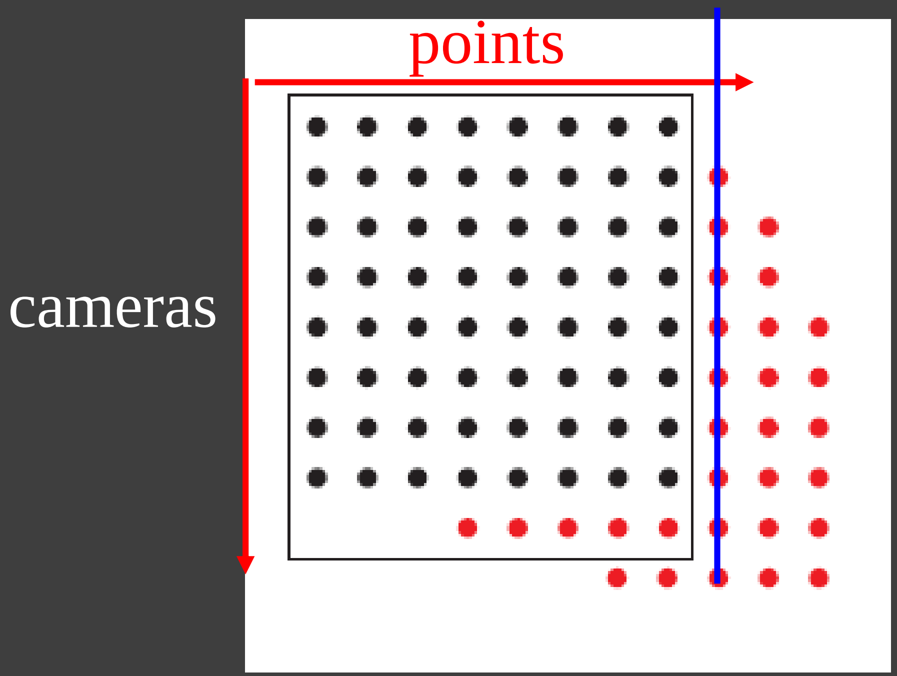
Projective SFM: Two-camera case

- 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



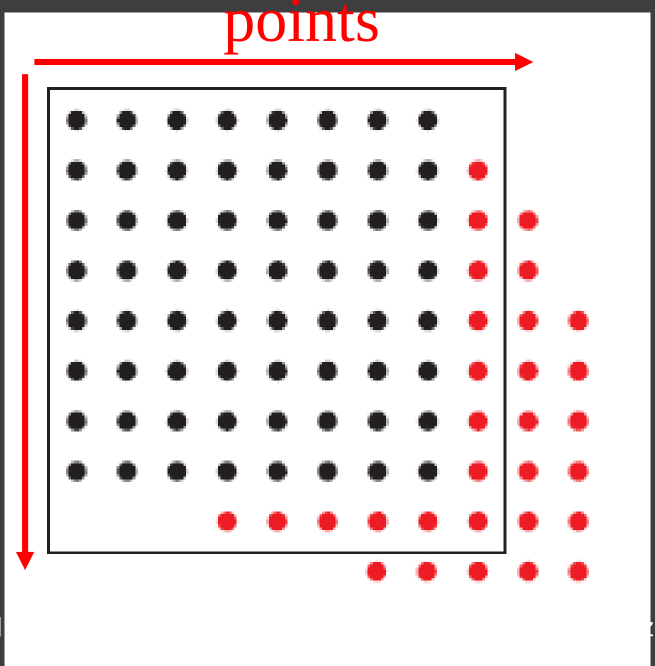
Projective SFM: Two-camera case

- 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



Projective SFM: Two-camera case

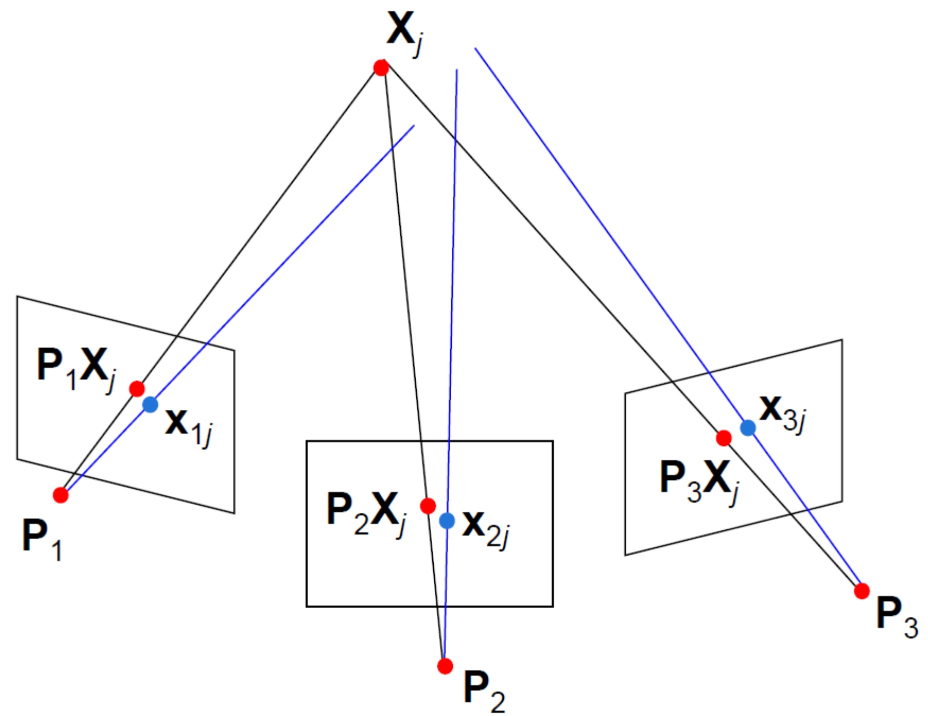
- Sequential structure from motion
  - Initialize motion from two images using fundamental matrix
  - Initialize structure by triangulation
  - For each additional view:
    - Determine projection matrix of new camera using all the known 3D points that are visible in its image –calibration
    - Refine 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



Projective SFM: Two-camera case

- Bundle adjustment
  - Non-linear method for refining structure and motion
  - Minimizing reprojection error

$$E(\mathbf{P},\mathbf{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} D\left(\mathbf{x}_{ij}, \mathbf{P}_i\mathbf{X}_j\right)^2$$



Projective SFM: Two-camera case

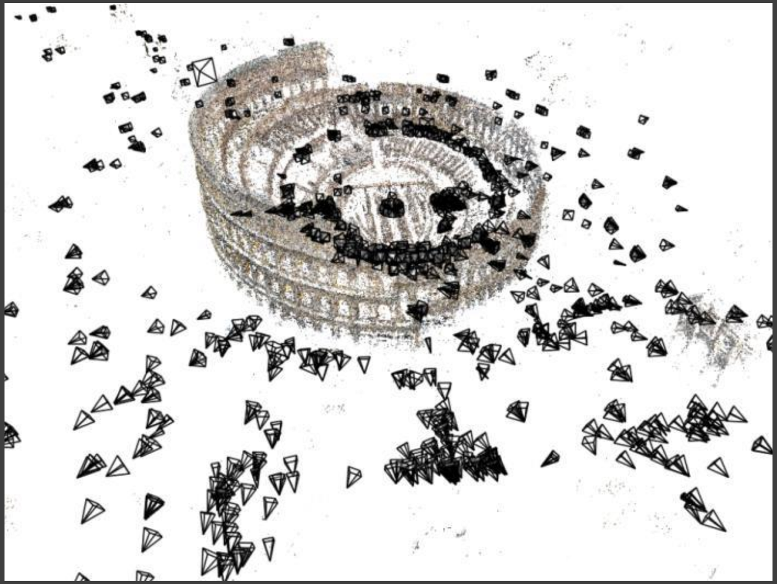
- Self-calibration
  - Self-calibration (auto-calibration) is the process of determining intrinsic camera parameters directly from uncalibrated images
  - For example, when the images are acquired by a single moving camera, we can use the constraint that the intrinsic parameter matrix remains fixed for all the images
    - Compute initial projective reconstruction and find 3D projective transformation matrix Q such that all camera matrices are in the form $\mathbf{P}_i = \mathbf{K}[\mathbf{R}_i|\mathbf{t}_i]$
  - Can use constraints on the form of the calibration matrix: zero skew

Review: Structure from motion

- Ambiguity
- Affine structure from motion
  - Factorization
- Dealing with missing data
  - Incremental structure from motion
- Projective structure from motion
  - Bundle adjustment
  - Self-calibration

Large-scale Structure from motion

- Given many images from photo collections how can we
  - figure out where they were all taken from?
  - build a 3D model of the scene?



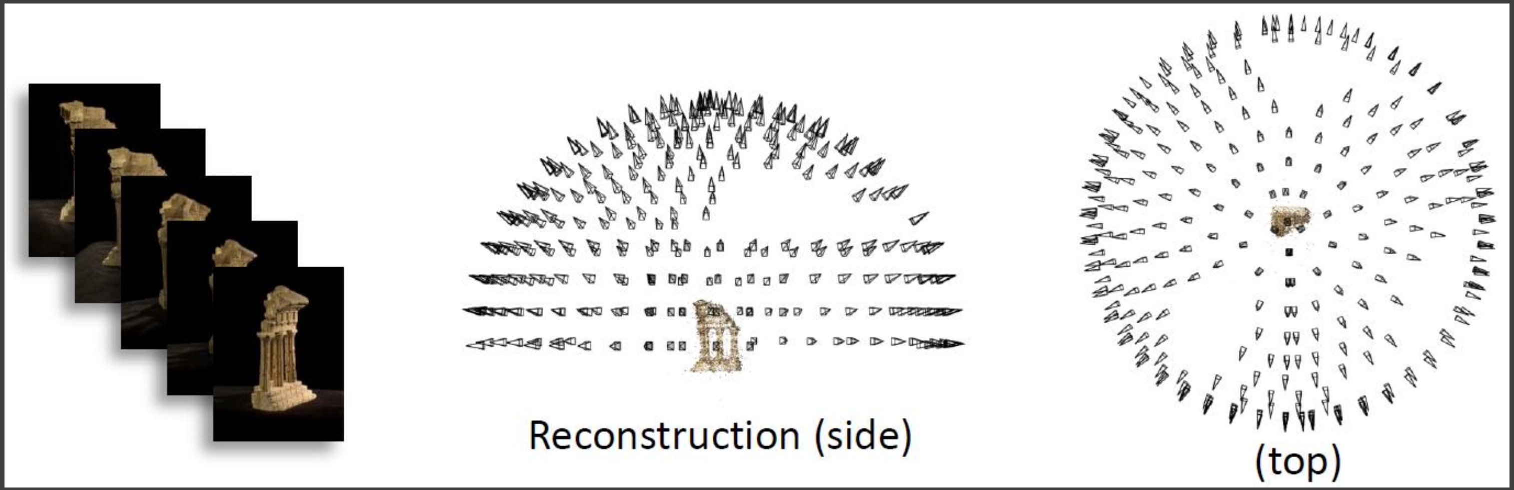
This is (roughly) the **structure from motion** problem

Large-scale Structure from motion

Dubrovnik, Croatia. 4,619 images (out of an initial 57,845).
Total reconstruction time: 23 hours
Number of cores: 352

Large-scale Structure from motion

- Structure from motion

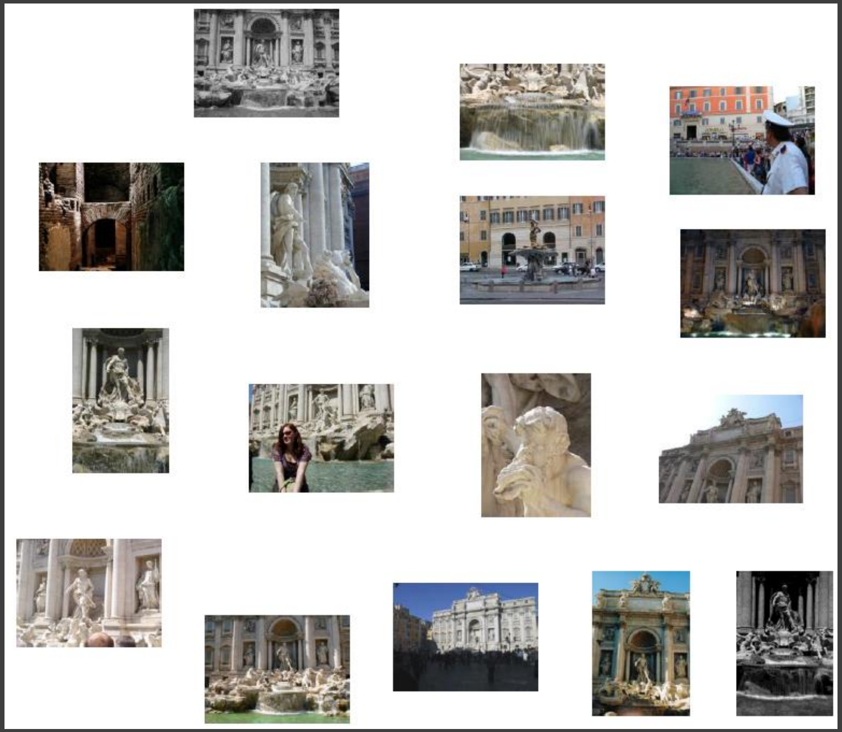


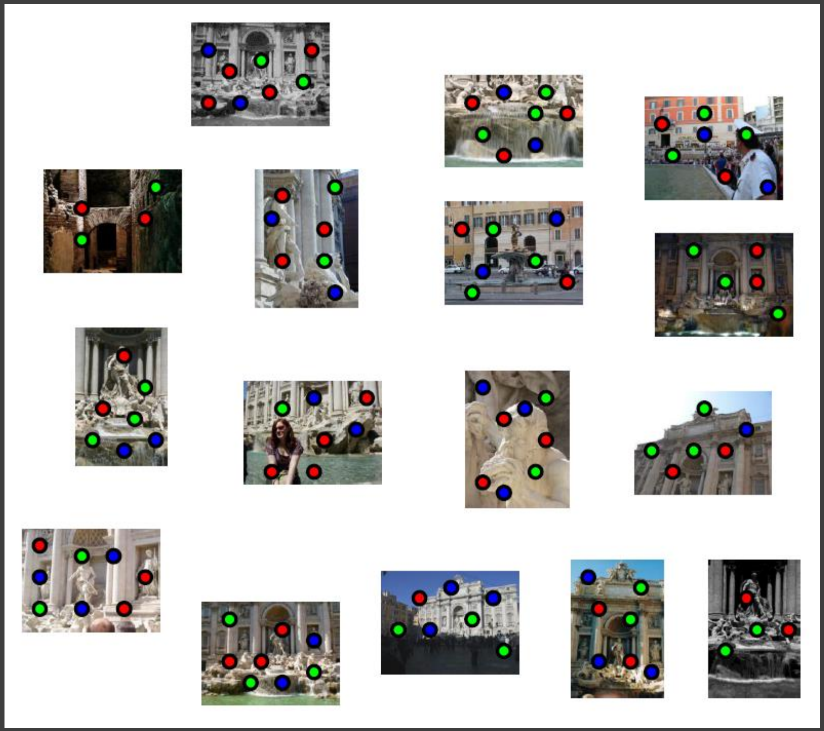
- Input: images with points in correspondence $p_{i,j} = (u_{i,j}, v_{i,j})$
- Output:
  - structure: 3D location $\mathbf{x}_i$ for each point $\boldsymbol{p}_i$
  - motion: camera parameters $\mathbf{R}_j$, $\mathbf{t}_j$ possibly $\mathbf{K}_j$

- Objective function: minimize reprojection error

Large-scale Structure from motion

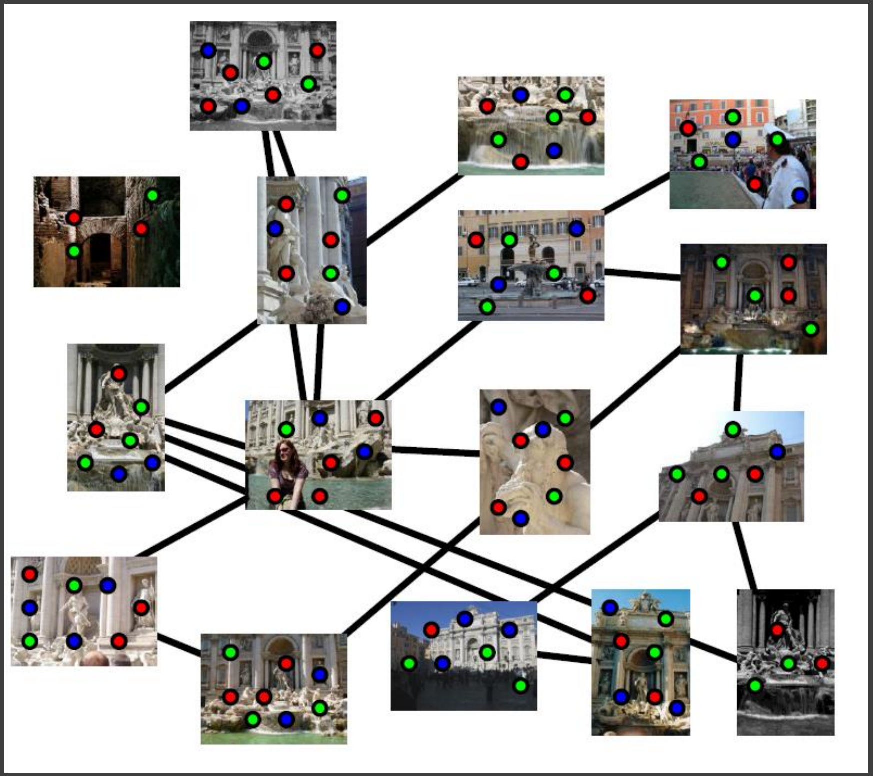
- First step: how to get correspondence?
  - Feature detection and matching
  - Detect features using SIFT[Lowe, IJCV2004]



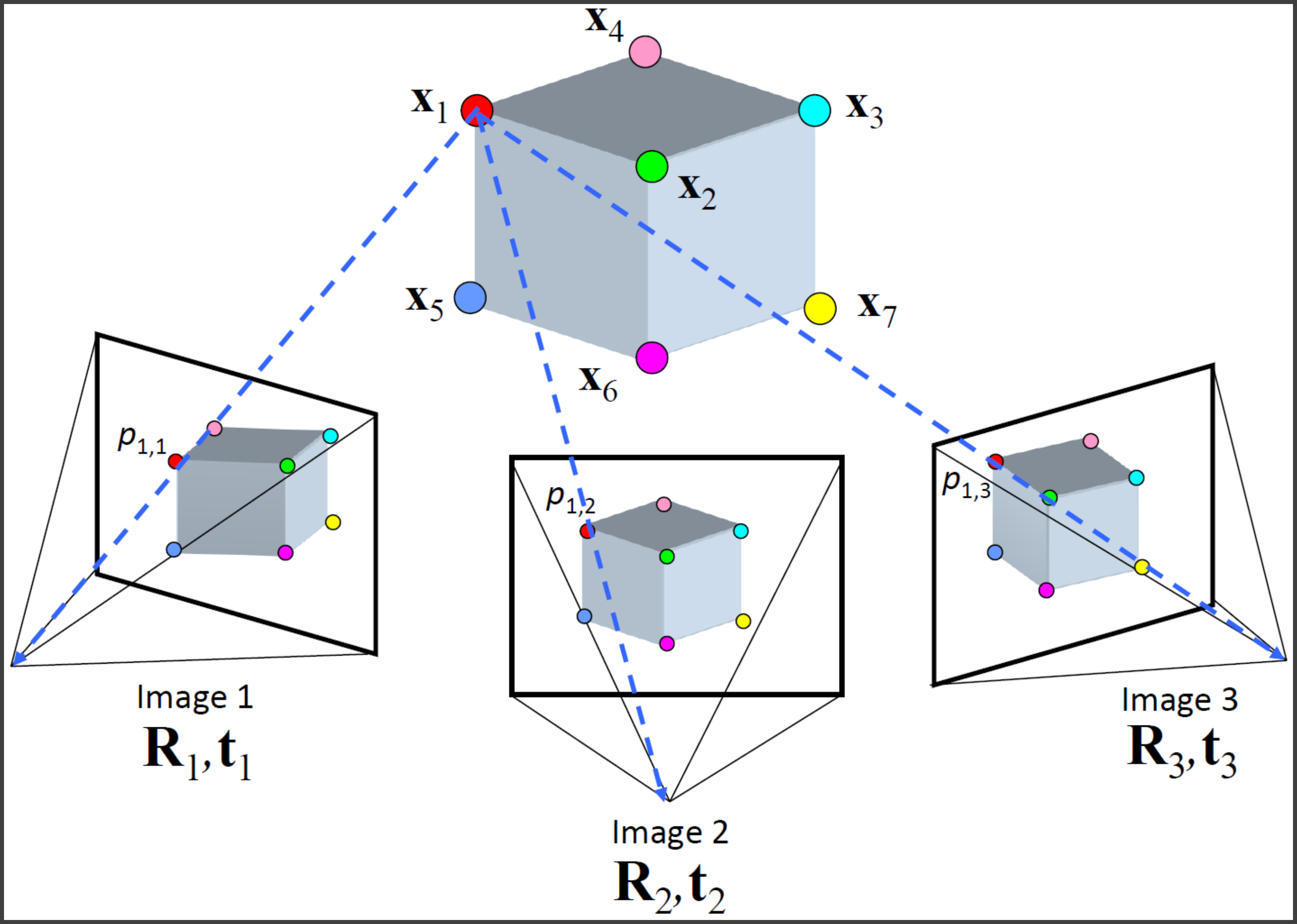


Large-scale Structure from motion

- First step: how to get correspondence?
  - Feature detection and matching
  - Detect features using SIFT[Lowe, IJCV2004]
  - Match features between each pair of images
  - Refine matching using RANSAC to estimate fundamental matrix between each pair

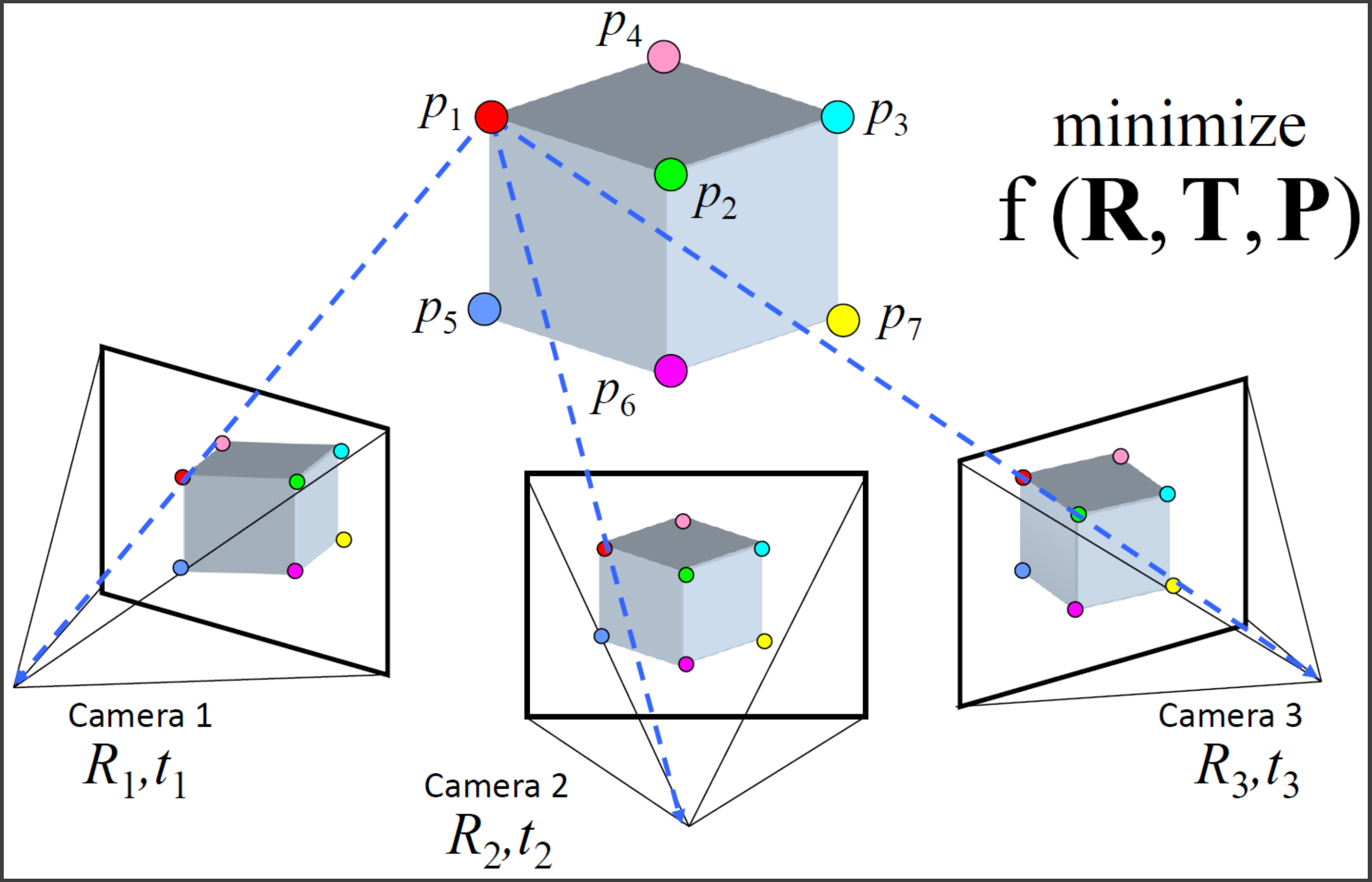


Large-scale Structure from motion



Large-scale Structure from motion

- Structure from motion

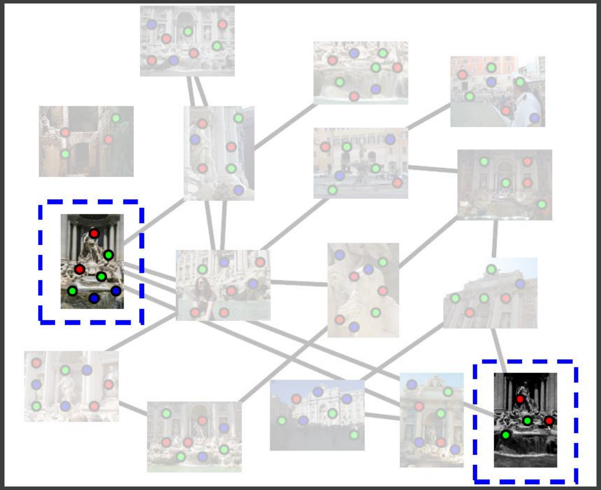


Problem size: Trevi Fountain collection

466 input photos + > 100,000 3D points = very large optimization problem

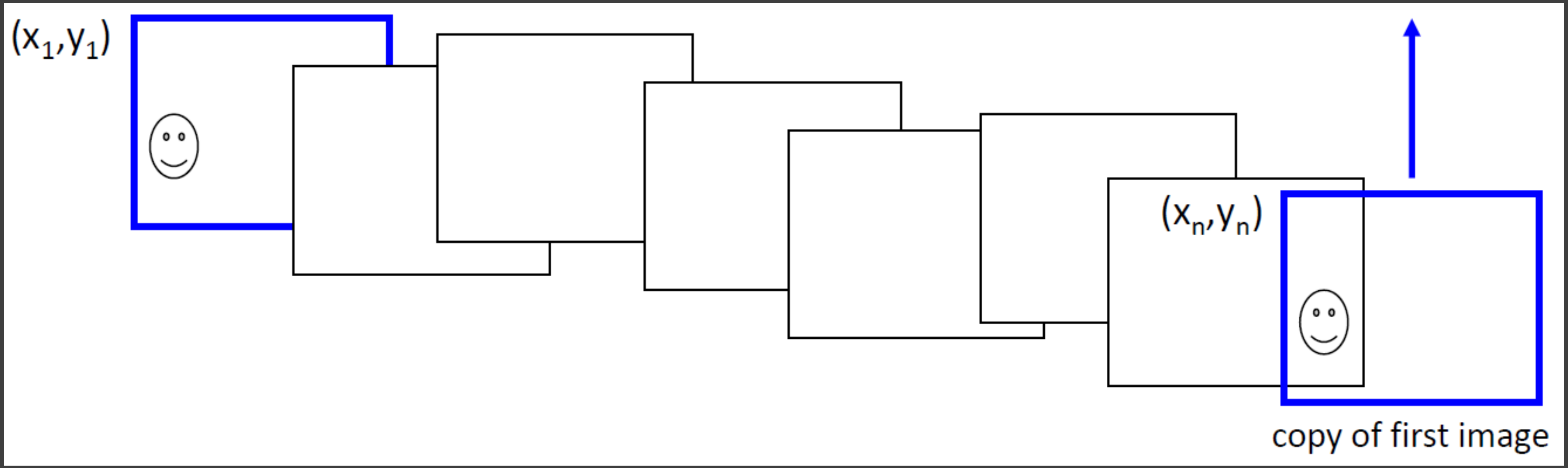
Large-scale Structure from motion

- Incremental structure from motion



Large-scale Structure from motion

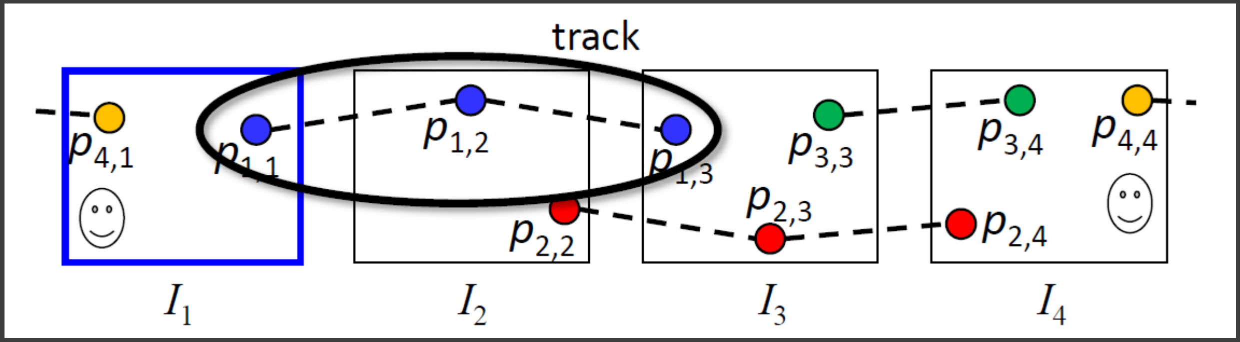
- Related topic: Drift



- add another copy of first image at the end
- this gives a constraint: $y_n = y_1$
- there are a bunch of ways to solve this problem
  - add displacement of $(y_1 – y_n)/(n - 1)$ to each image after the first
  - compute a global warp: y’ = y + ax
  - run a big optimization problem, incorporating this constraint
    - –best solution, but more complicated
    - –known as “bundle adjustment”

Large-scale Structure from motion

- Global optimization

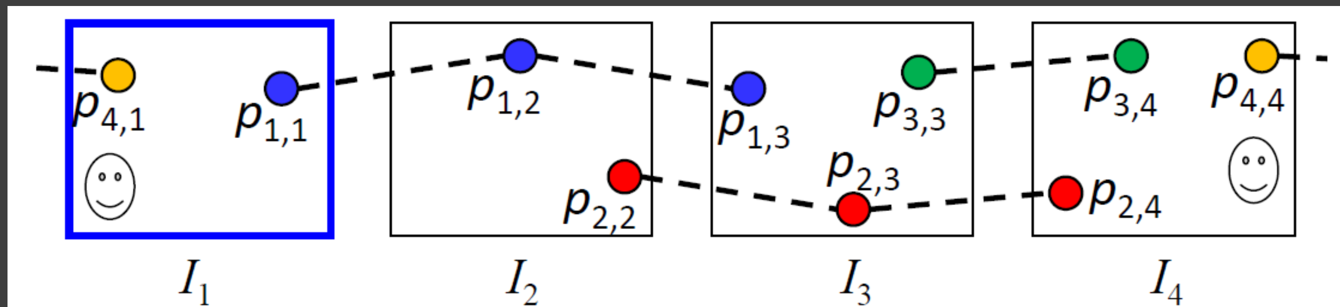


Minimize a global energy function:

- What are the variables?
  - The translation $t_j = (x_j, y_j)$ for each image I_j
- What is the objective function?
  - We have a set of matched features $p_{i,j} = (u_{i,j}, v_{i,j})$
    - » We'll call these *tracks*
  - For each point match $(p_{i,j}, p_{i,j+1})$: $p_{i,j+1} - p_{i,j} = t_{j+1} - t_j$

Large-scale Structure from motion

- Global optimization



$w_{ij} = 1$ if track i is visible in images j and j+1
$w_{ij} = 0$ otherwise

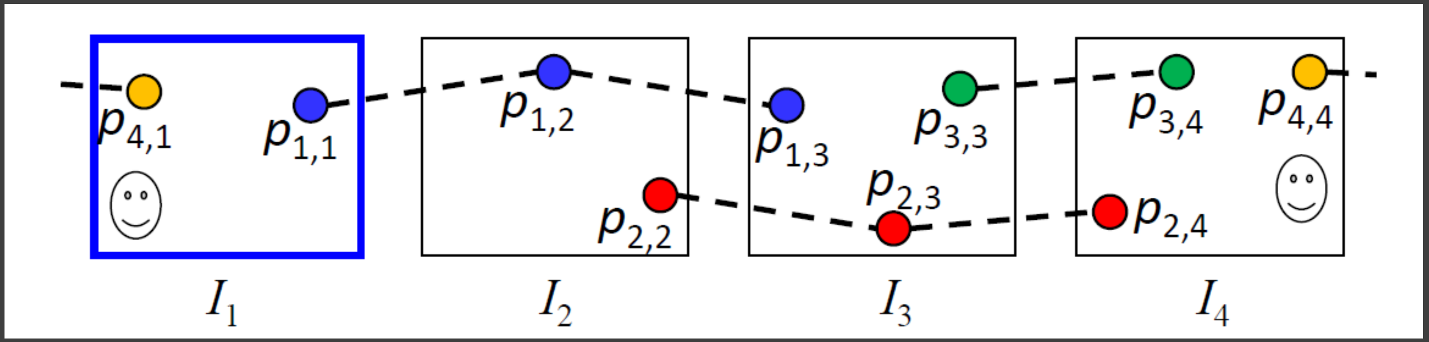
$$
\begin{aligned}
p_{1,2} - p_{1,1} &= t_2 - t_1 \\
p_{1,3} - p_{1,2} &= t_3 - t_2 \\
p_{2,3} - p_{2,2} &= t_3 - t_2 \\
&\cdots \\
v_{4,1} - v_{4,4} &= y_1 - y_4
\end{aligned}
$$

Minimize

$$
\sum_{i=1}^{m} \sum_{j=1}^{n-1} w_{ij} \cdot \left\| (p_{i,j+1} - p_{i,j}) - (t_{j+1} - t_j) \right\|^2 + \sum_{i=1}^{m} w_{in} \cdot \left\| (v_{i,1} - v_{i,n}) - (y_1 - y_n) \right\|^2
$$

Large-scale Structure from motion

- Global optimization



$$\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ & & & \cdots & & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix} = \begin{bmatrix} u_{1,2} - u_{1,1} \\ v_{1,2} - v_{1,1} \\ \vdots \\ v_{4,1} - v_{4,4} \end{bmatrix}$$

A 2m x 2n **x** 2n x 1 **b** 2m x 1

Large-scale Structure from motion

- Global optimization

$$\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ & & & \cdots & & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix} = \begin{bmatrix} u_{1,2} - u_{1,1} \\ v_{1,2} - v_{1,1} \\ \vdots \\ v_{4,1} - v_{4,4} \end{bmatrix}$$

$\mathbf{A}$ 2m x 2n $\quad$ $\mathbf{x}$ 2n x 1 $\quad$ $\mathbf{b}$ 2m x 1

Defines a least squares problem: minimize $\|\mathbf{Ax} - \mathbf{b}\|$

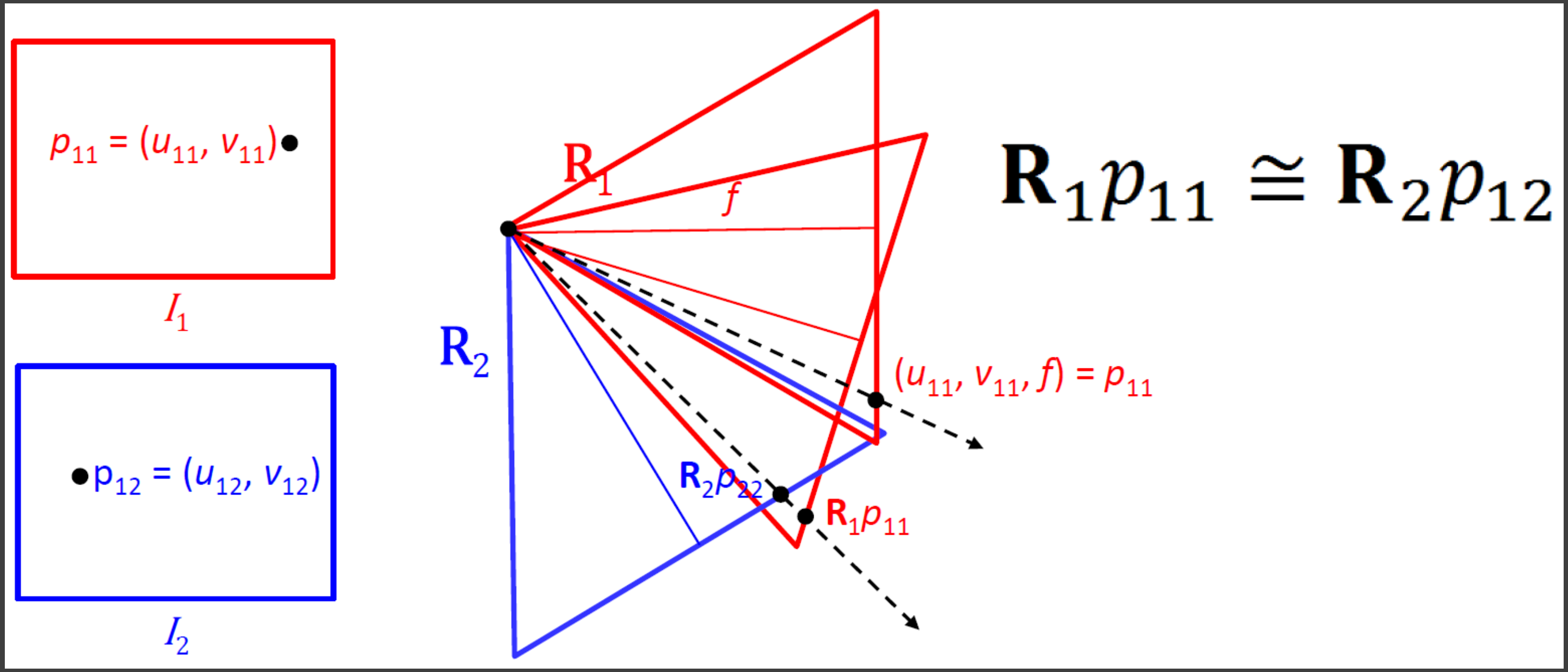
- Solution: $\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$
- Problem: there is no unique solution for $\hat{\mathbf{x}}$! (det$(\mathbf{A}^T\mathbf{A})$ = 0)
- We can add a global offset to a solution $\hat{\mathbf{x}}$ and get the same error

Large-scale Structure from motion

- Solving for camera rotation
- Instead of spherically warping the images and solving for translation, we can directly solve for the rotation R_j of each camera.
- Can handle tilt / twist.

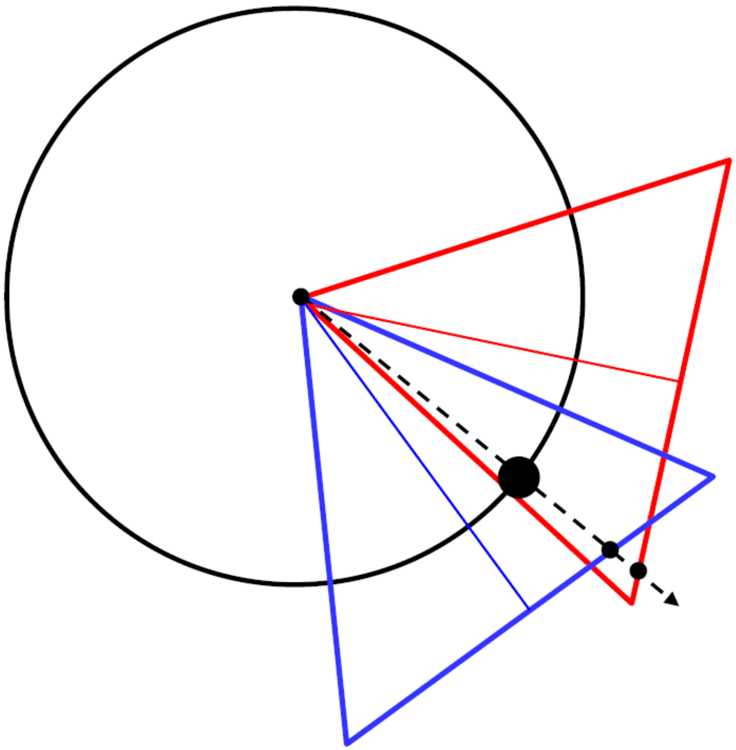
Large-scale Structure from motion

- Solving for camera rotation
- Instead of spherically warping the images and solving for translation, we can directly solve for the rotation R_j of each camera
- Can handle tilt / twist



Large-scale Structure from motion

- Solving for camera rotation



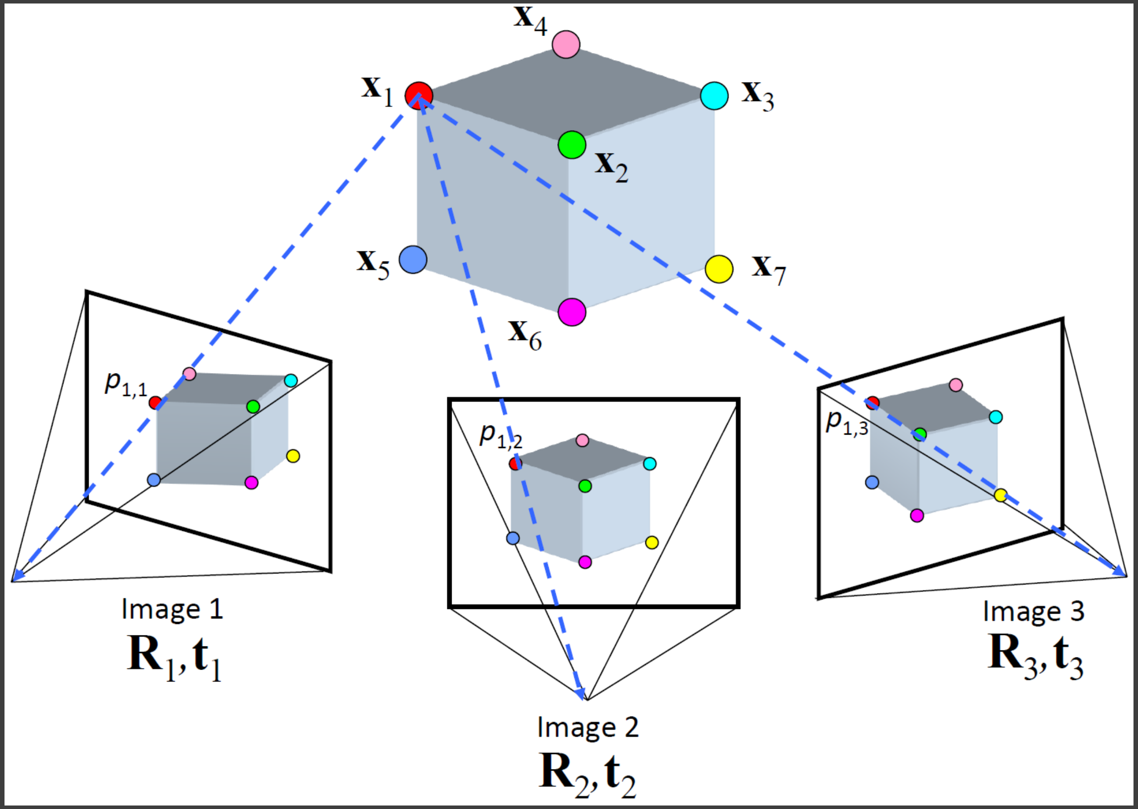
$$\mathbf{R}_1 p_{11} \cong \mathbf{R}_2 p_{12}$$

$$\mathbf{R}_1 \hat{p}_{11} = \mathbf{R}_2 \hat{p}_{12}$$

$$\text{minimize} \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| \mathbf{R}_{j+1} \hat{p}_{i,j+1} - \mathbf{R}_j \hat{p}_{i,j} \right\|^2$$

Large-scale Structure from motion

- 3D rotations
- How many degrees of freedom are there? How do we represent a rotation?
  - Rotation matrix (too many degrees of freedom)
  - Euler angles (e.g. yaw, pitch, and roll) –bad idea
  - Quaternions (4-vector on unit sphere)
- Usually involves non-linear optimization



SfM objective function

- Given point **x** and rotation and translation **R**, **t**

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{Rx} + \mathbf{t} \qquad u' = \frac{fx'}{z'} \qquad v' = \frac{fy'}{z'}$$

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \mathbf{P}(\mathbf{x}, \mathbf{R}, \mathbf{t})$$

- Minimize sum of squared reprojection errors:

$$g(\mathbf{X}, \mathbf{R}, \mathbf{T}) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \cdot \left\| \underbrace{\mathbf{P}(\mathbf{x}_i, \mathbf{R}_j, \mathbf{t}_j)}_{\substack{\textit{predicted} \\ \text{image location}}} - \underbrace{\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}}_{\substack{\textit{observed} \\ \text{image location}}} \right\|^2$$

Solving structure from motion

- Minimizing g is difficult
  - g is non-linear due to rotations, perspective division
  - Lots of parameters: 3 for each 3D point, 6 for each camera
  - Difficult to initialize
  - Gauge ambiguity: error is invariant to a similarity transform (translation, rotation, uniform scale)

- Many techniques use non-linear least-squares (NLLS) optimization (bundle adjustment)
  - Levenberg-Marquardt is one common algorithm for NLLS
  - Lourakis, The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm, http://www.ics.forth.gr/~lourakis/sba/
  - http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm

- Large scale 3D modeling from images https://demuc.de/tutorials/cvpr2017/

structure from motion

- Examples

From feature matching to dense stereo

1. Extract features
2. Get a sparse set of initial matches
3. Iteratively expand matches to nearby locations
4. Use visibility constraints to filter out false matches
5. Perform surface reconstruction

Yasutaka Furukawa and Jean Ponce, [Accurate, Dense, and Robust Multi-View Stereopsis](), CVPR 2007.

structure from motion

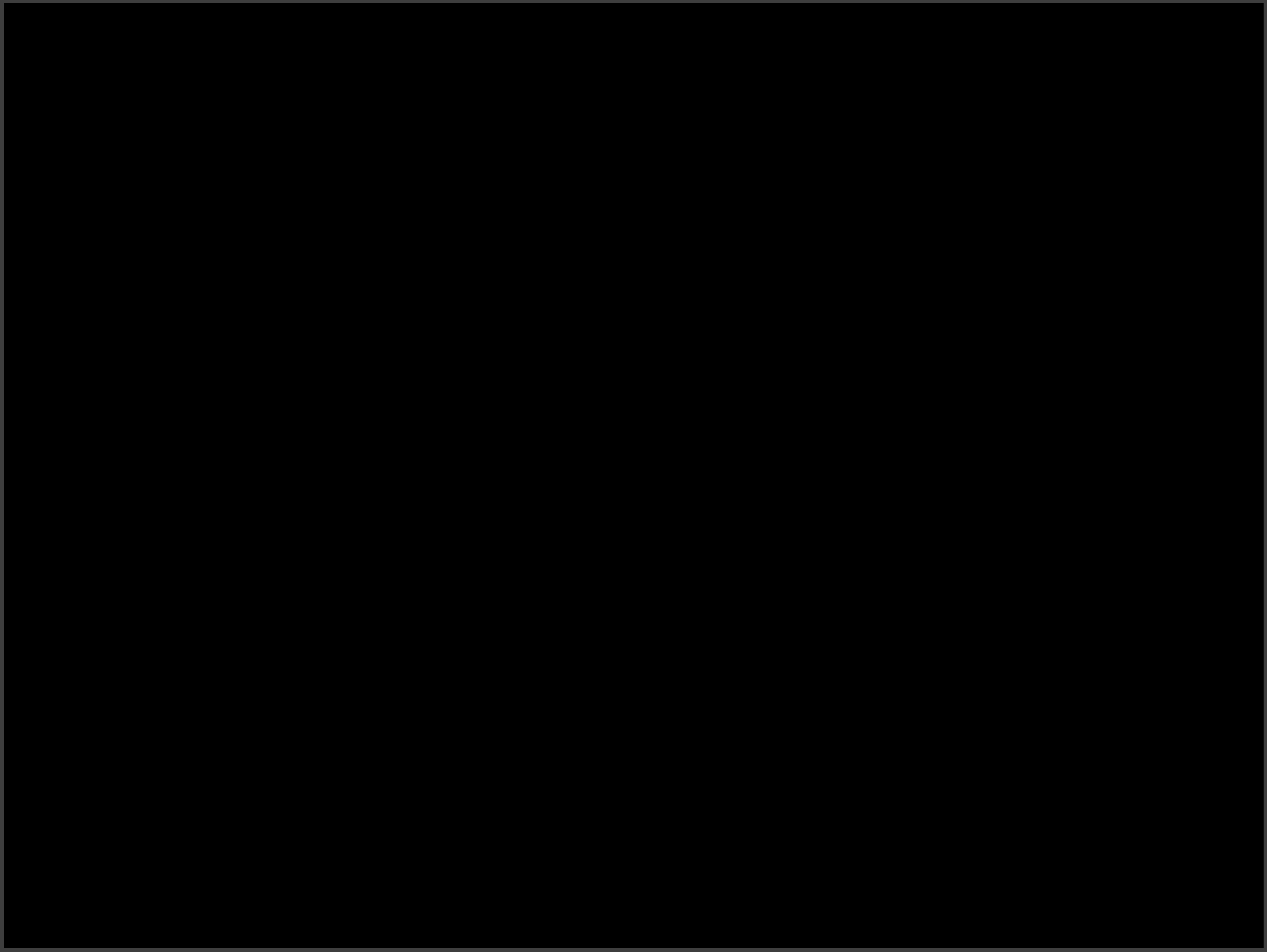
- Examples

http://www.cs.washington.edu/homes/furukawa/gallery/

Yasutaka Furukawa and Jean Ponce, **Accurate, Dense, and Robust Multi-View Stereopsis**, CVPR 2007.

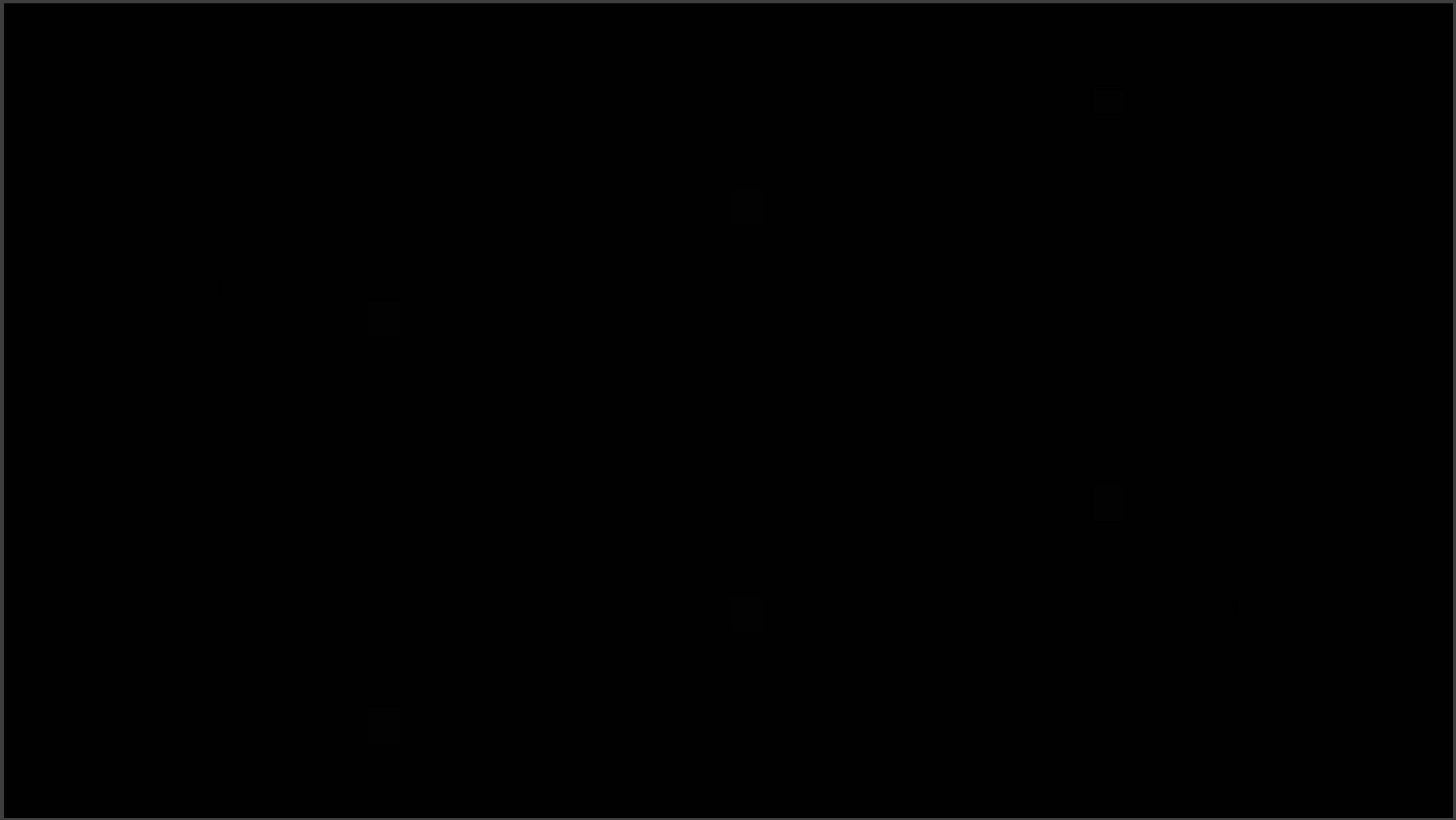
structure from motion

- Examples



structure from motion

- Examples



My pleasure to give this talk, and thanks for your cooperation!

See You