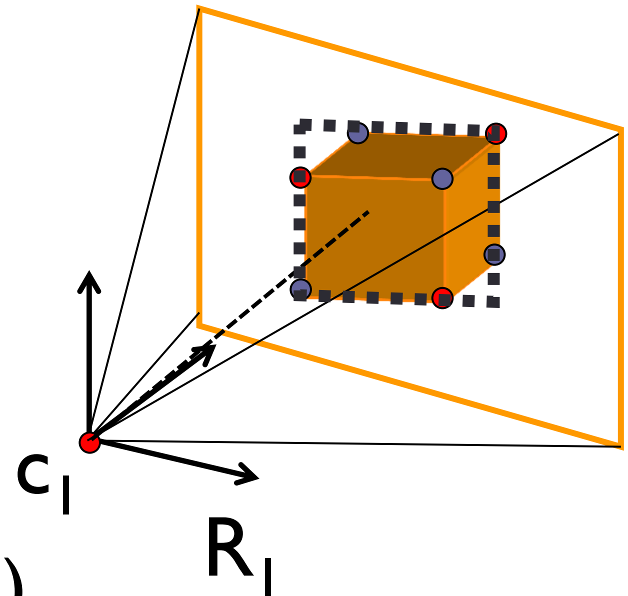
CEE598 - Visual Sensing for Civil Infrastructure Eng. & Mgmt.

Session 21 – Structure from Motion Used in D4AR Modeling Approach

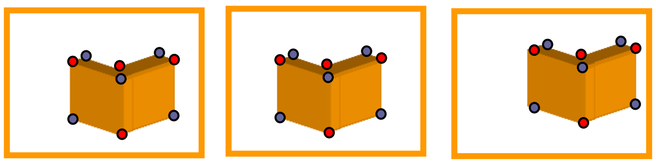
Mani Golparvar-Fard
Department of Civil and Environmental Engineering
3129D, Newmark Civil Engineering Lab
e-mail: mgolpar@illinois.edu

Points and cameras

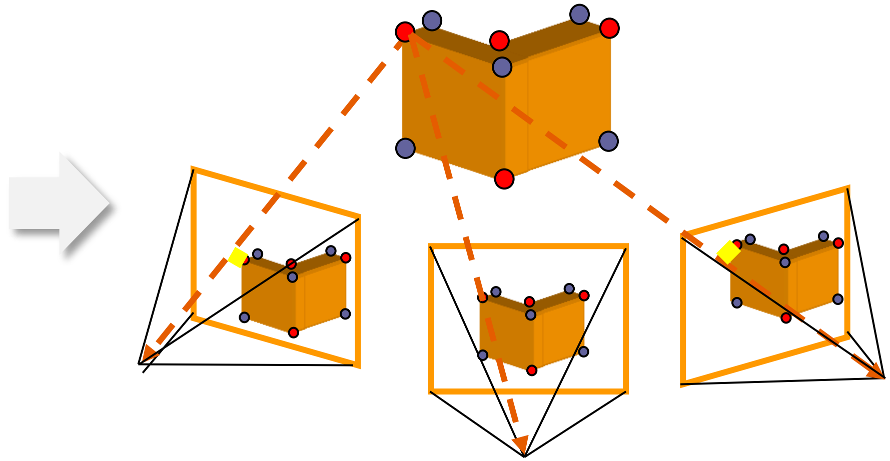
- Point: 3D position in space (X_j)
- Camera (C_i):
  - A 3D position (C_i)
  - A 3D orientation (R_i)
  - Intrinsic parameters (**focal length**, aspect ratio, …)
  - 7 parameters (3+3+1) in total



Solving SfM



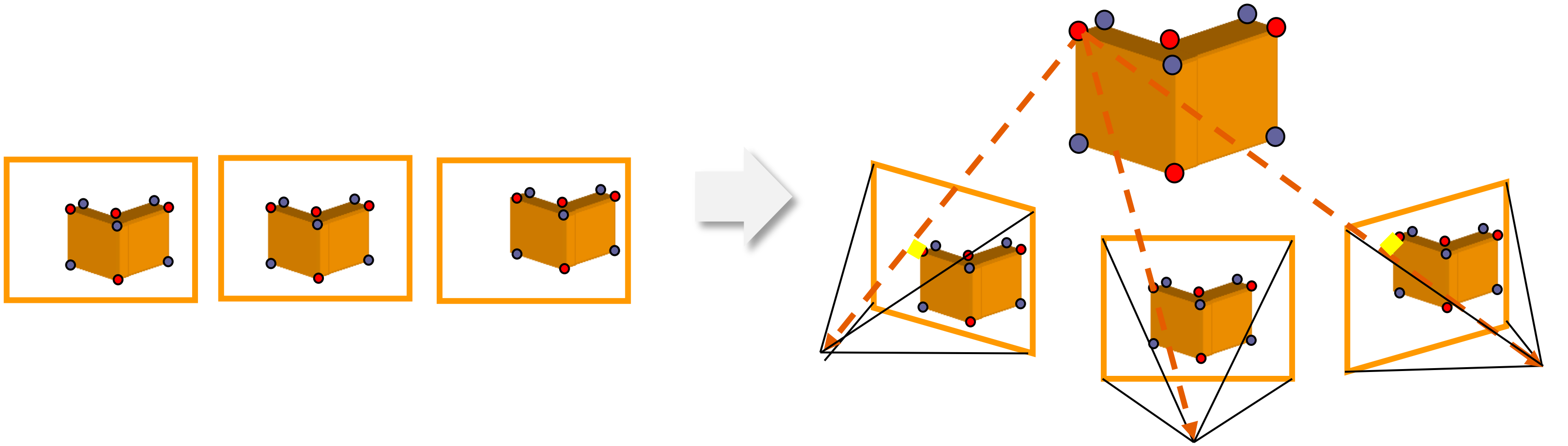
Inputs: feature tracks



Outputs: 3D cameras and points

- How do we solve the SfM problem?
- Challenges:
  - Large number of parameters (1000's of cameras, millions of points)
  - Very non-linear objective function

Solving SfM

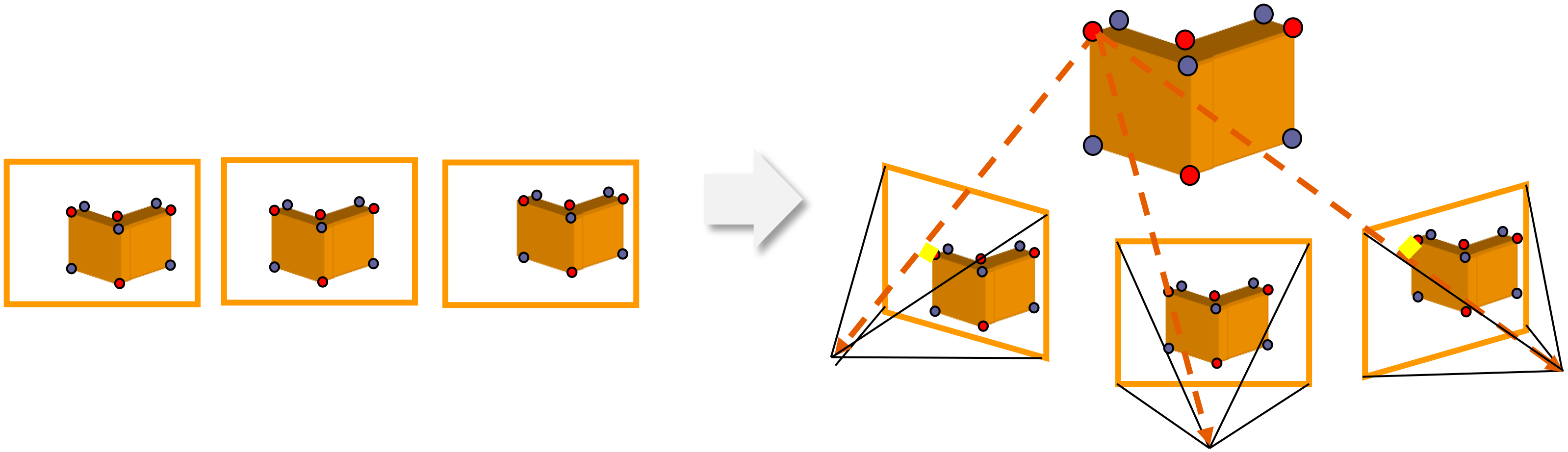


Inputs: feature tracks

Outputs: 3D cameras and points

- Important tool: Bundle Adjustment [Triggs *et al.* '00]
  - Joint non-linear optimization of both cameras and points
  - Very powerful, elegant tool
- The bad news:
  - Starting from a random initialization is very likely to give the wrong answer
  - Difficult to initialize all the cameras at once

Solving SfM



Inputs: feature tracks

Outputs: 3D cameras and points

- The good news:
  - Structure from motion with two cameras is (relatively) easy
  - Once we have an initial model, it's easy to add new cameras
- Idea:
  - Start with a small seed reconstruction, and grow

Incremental SfM: Algorithm

1. Pick a strong initial pair of images
2. Initialize the model using two-frame SfM
3. While there are connected images remaining:
   a. Pick the image which sees the most existing 3D points
   b. Estimate the pose of that camera
   c. Triangulate any new points
   d. Run bundle adjustment

1. Picking the initial pair

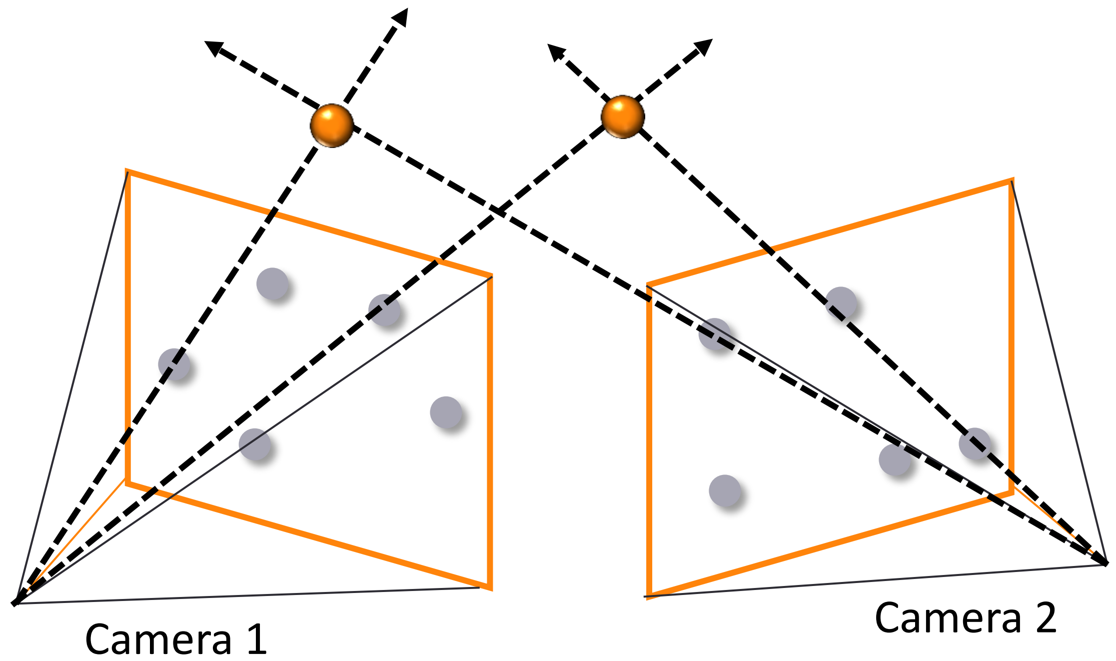
- We want a pair with many matches, but which has as large a baseline as possible

large baseline
very few matches

lots of matches
small baseline

large baseline
lots of matches

Initial Image Pair reconstruction



- Initial pair selection heuristics
  - Choose an image pair with min 100 matches
  - The ratio is as small as possible

- Find Calibration Information from EXIF tag of photos
- Estimate the pose using 5-point algorithm (David Nister, PAMI '04)

2. Two-frame reconstruction

- Input: two images with correspondence
- Output: camera parameters, 3D points

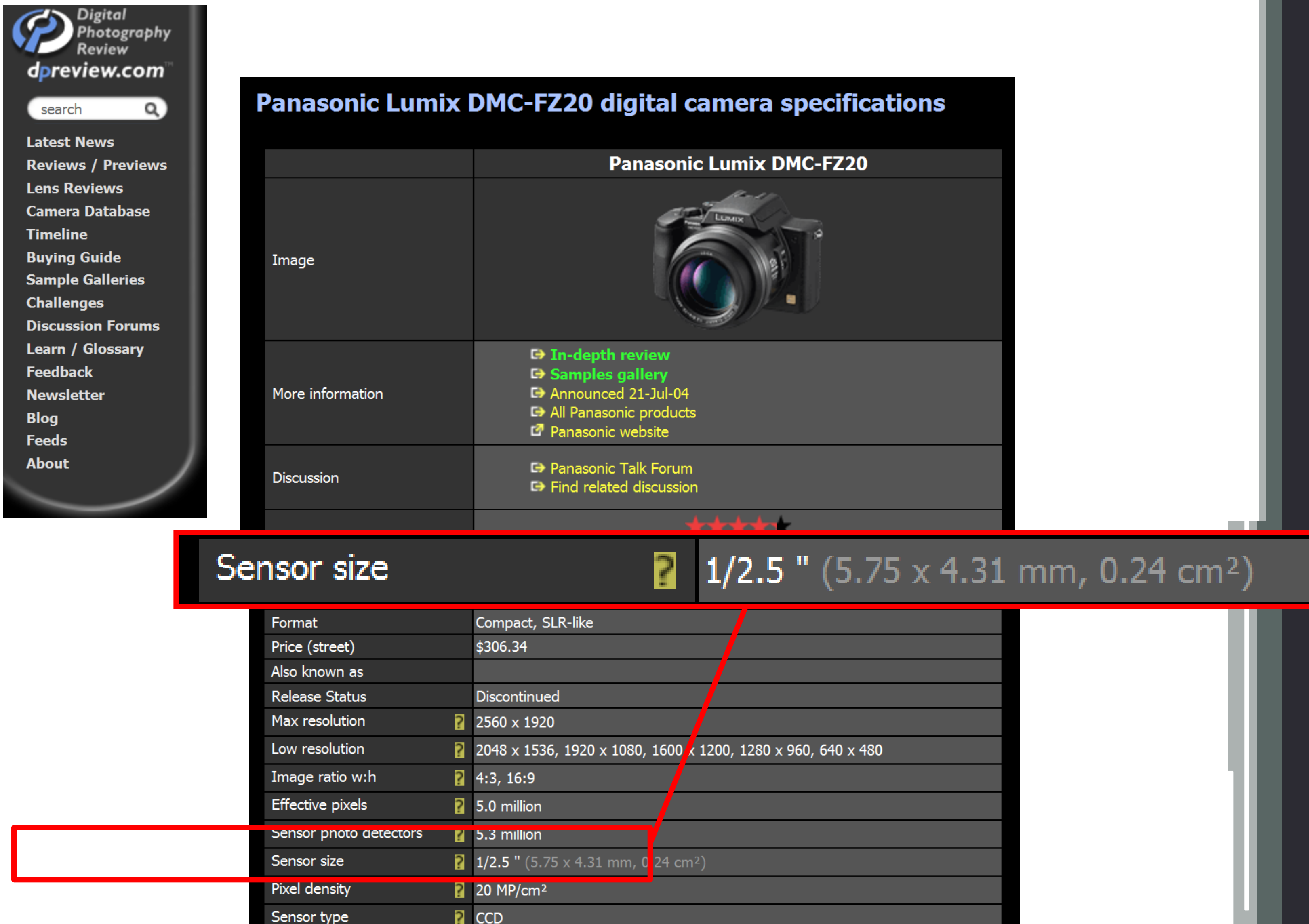
- In general, there can be ambiguities if the cameras are un calibrated (camera intrinsic: unknown)
- We assume that the only intrinsic parameter is an unknown focal length

Finding calibration information

- Many cameras list the focal length of a photo in its EXIF metadata

File size : 85111 bytes
File date : 2008:07:08 15:17:12
Camera make : Nikon
Camera model : D300
Date/Time : 2008:07:08 15:17:12
Resolution : 2414 x 1424
Flash used : No
Focal length : 6.0mm
Exposure time: 0.0012 s (1/800)
Aperture : f/5.6
ISO equiv. : 80
Whitebalance : Auto
Metering Mode: matrix
Exposure : program (auto)

Extract Focal Length from the image meta-data



http://www.dpreview.com/reviews/specs/Panasonic/panasonic_dmcfz20.asp

Finding calibration information

```
File size     : 85111 bytes
File date     : 2008:07:08 15:17:12
Camera make   : Nikon
Camera model  : D300
Date/Time     : 2008:07:08 15:17:12
Resolution    : 2414 x 1424
Flash used    : No
Focal length  : 6.0mm
Exposure time: 0.0012 s  (1/800)
Aperture      : f/5.6
ISO equiv.    : 80
Whitebalance  : Auto
Metering Mode: matrix
Exposure      : program (auto)
```

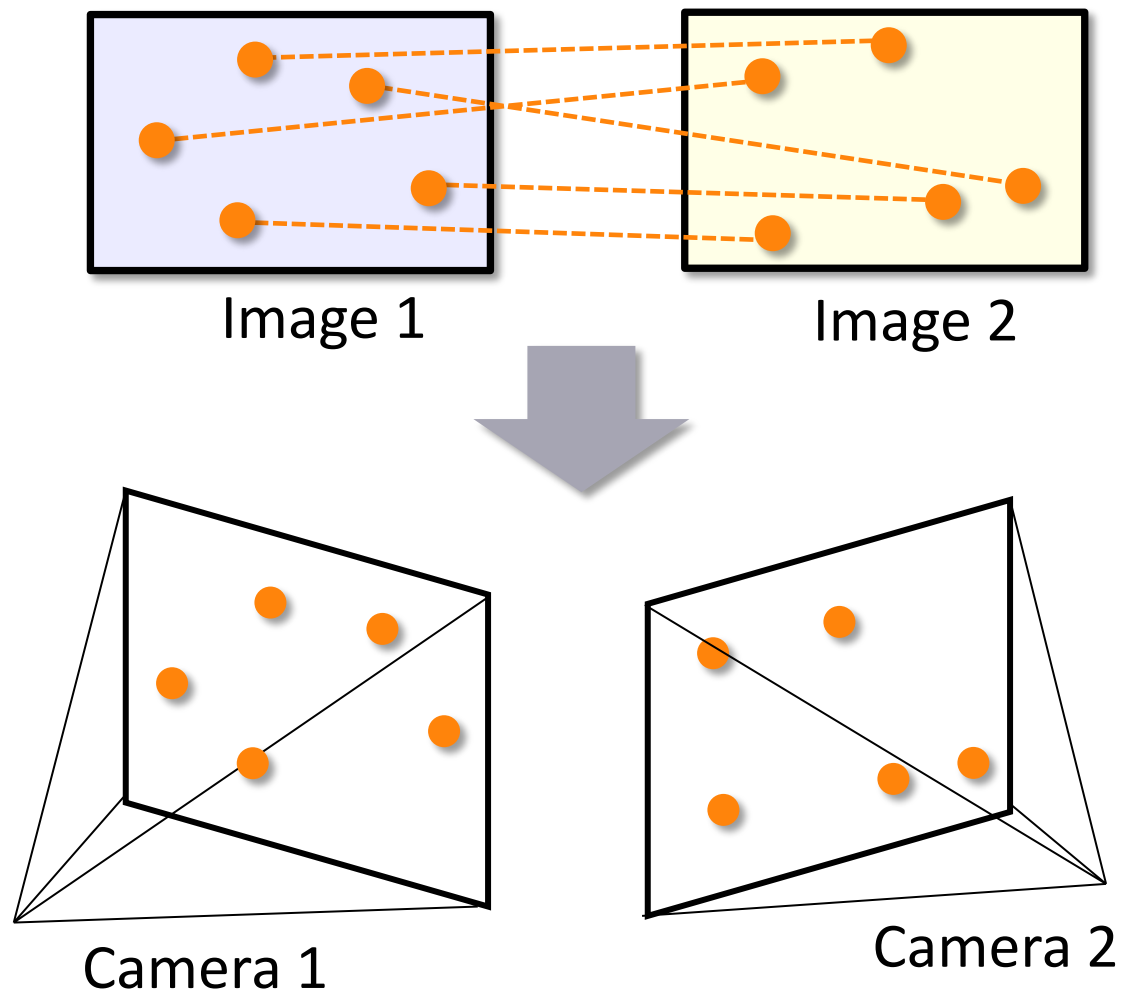
Focal length (pixels) = Focal length (mm) x Image width (pixels) / Sensor size (mm)

= 6.0 mm x 600 pixels / 5.75 mm = 626.1 pixels

2. Two-view reconstruction

- Two-view SfM: Given two calibrated images with corresponding points, compute the camera and point positions
- Solved by finding the essential matrix between the images
- Best approach is the 5-point algorithm (as opposed to the 6-, 7-, or 8- point algorithms)

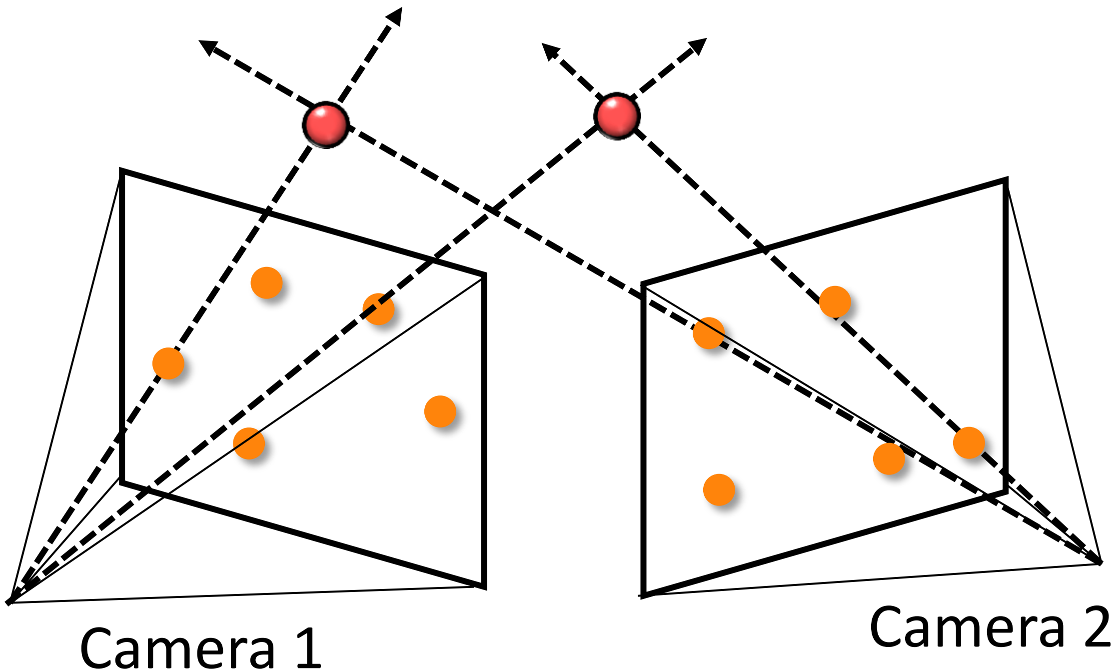
Five-point algorithm



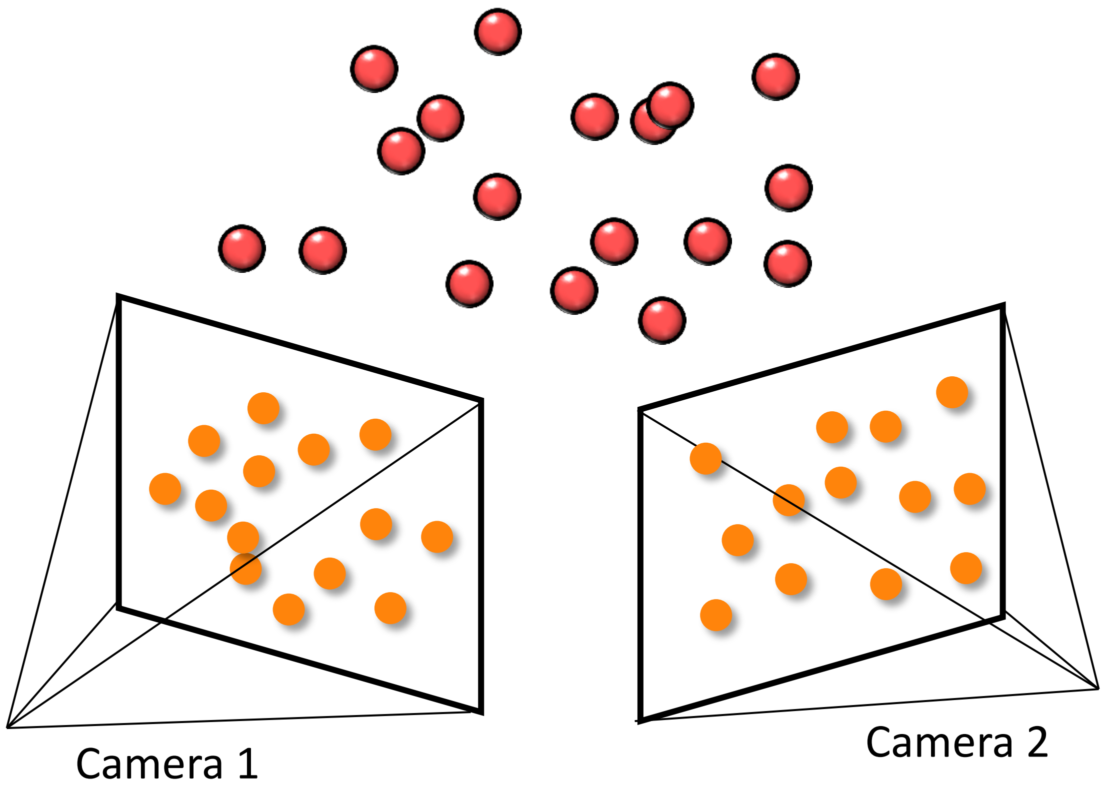
Five-point algorithm

- First practical solution to the 5-point algorithm: [Nister, "An efficient solution to the 5-point relative pose problem," PAMI '04]
- See also:
  - [Li and Hartley, "Five-Point Motion Estimation Made Easy," ICPR '06]

Two-view reconstruction

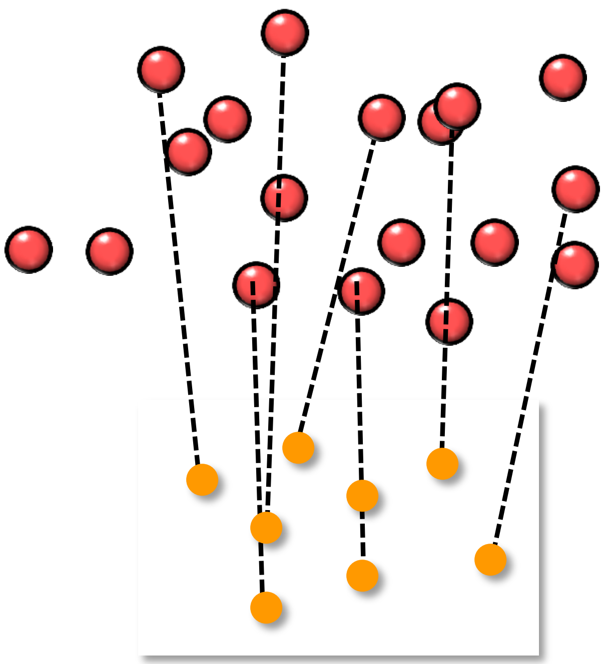


Two-view reconstruction

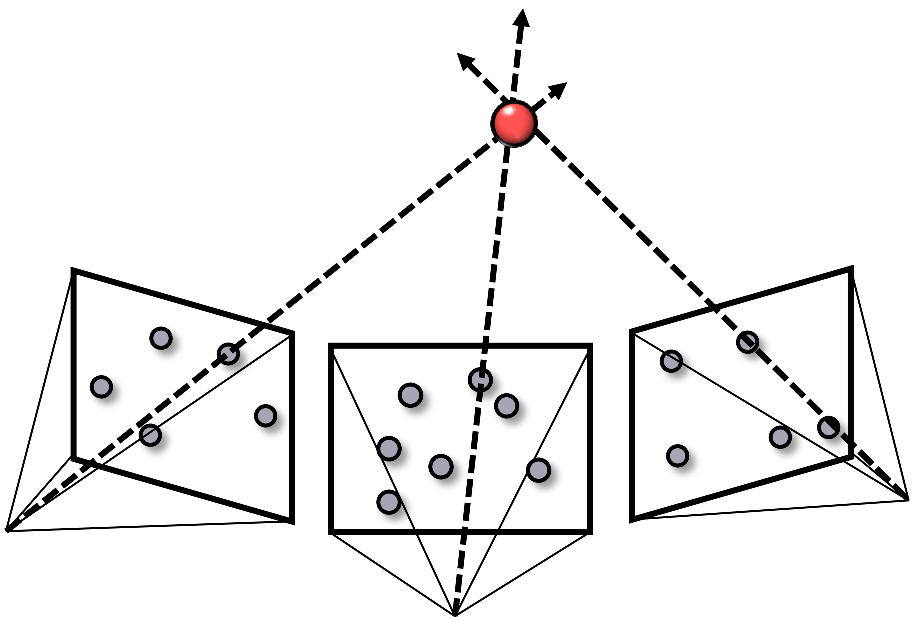


3bc. Pose estimation and Triangulation

- Next step: grow the reconstruction by adding another image, triangulating new points



Pose estimation: 2D -> 3D



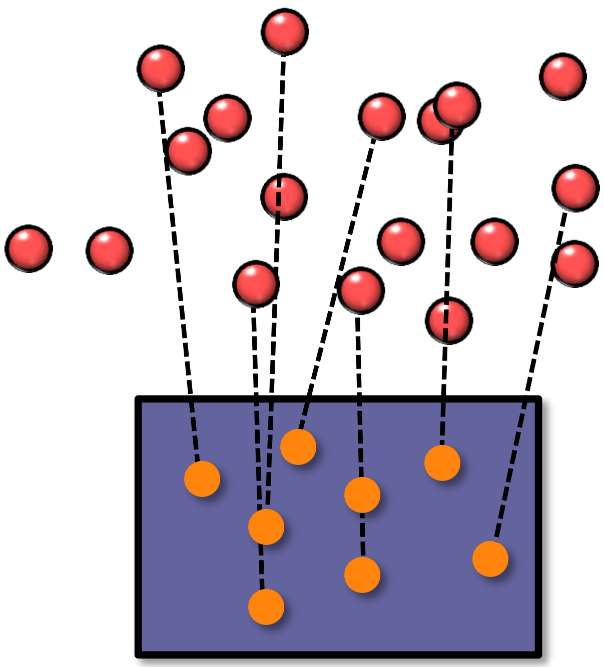
n-view triangulation

3bc. Pose estimation and triangulation

- Next step: grow the reconstruction by adding another image, triangulating new points

- Both of these problems can be solved approximately using linear systems (Direct Linear Transformation (DLT))

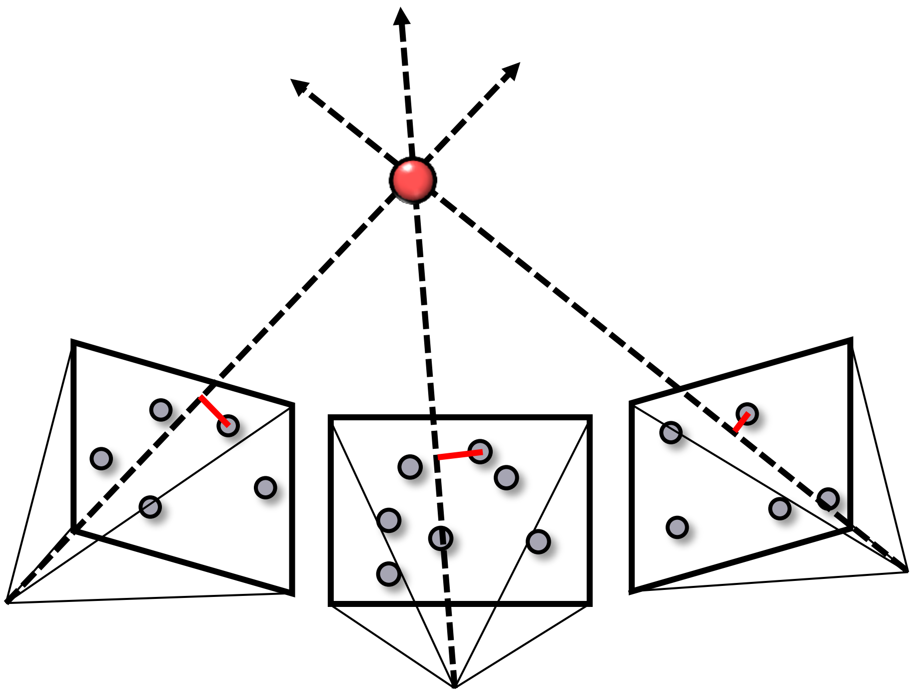
3b. Pose estimation



Pose estimation: 2D -> 3D

- Choose the image with the most matches to existing 3D points
- Linear 6-point algorithm for finding the 3x4 projection matrix $\mathbf{\Pi}$
- $\mathbf{\Pi}$ can then be decomposed into $\mathbf{K}[\mathbf{R}|\mathbf{t}]$ (intrinsics + rotation and translation) using RQ decomposition
- Use non-linear polishing to snap the camera into place
- For calibrated cameras, there is also a 3-point algorithm

3c. *n*-view triangulation



- Objective function: sum of squared reprojection errors
- Also solvable (approximately) using a simple linear system
- Follow with a non-linear polishing

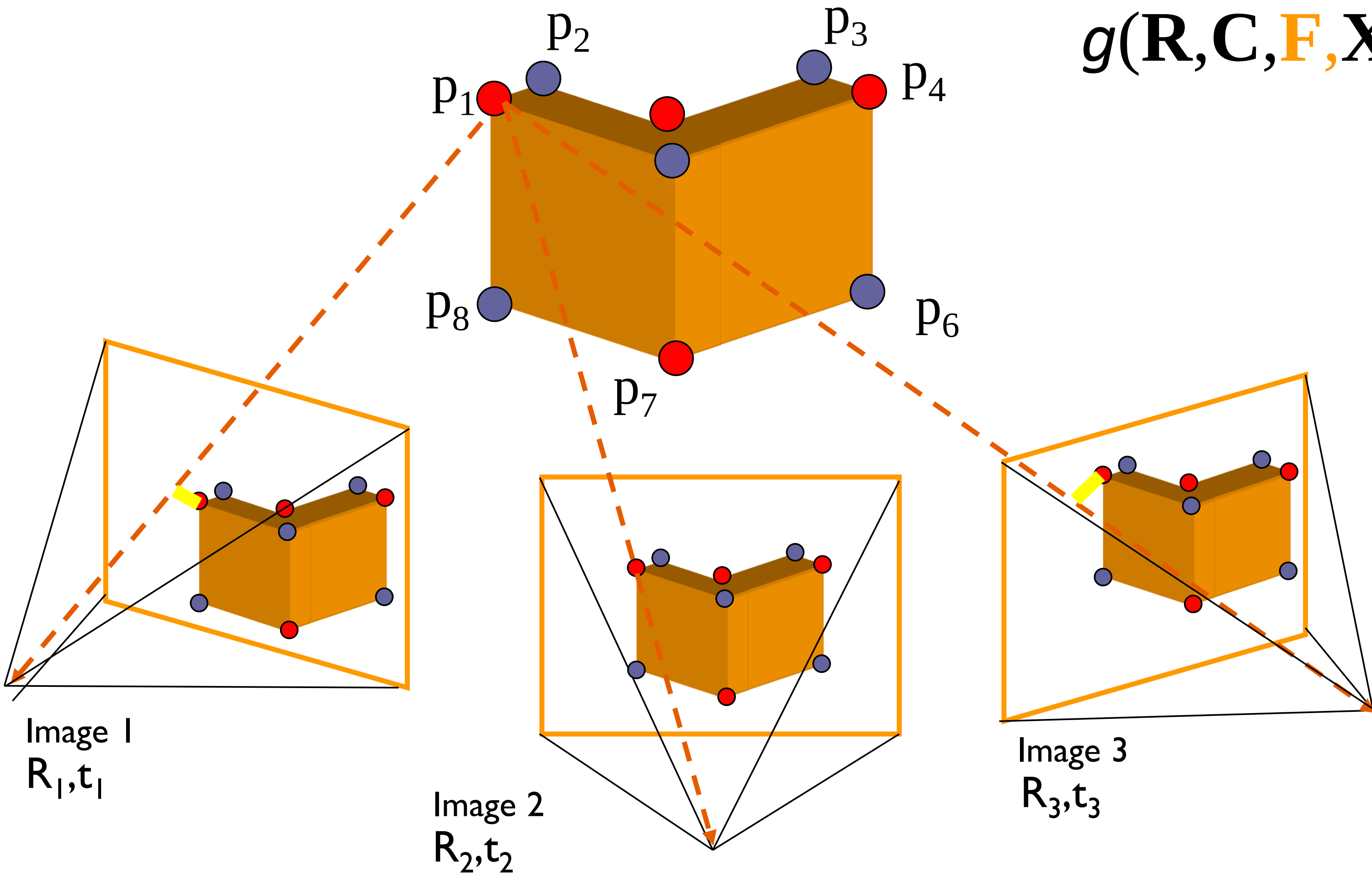
3bc. Pose estimation and triangulation

- In practice, multiple images can be added at once
- If the highest-matching image has *N* matches, add all images with at least 0.75 *N* matches (or at least 500 matches)

3d. Bundle adjustment

minimize

$$g(\mathbf{R},\mathbf{C},\mathbf{F},\mathbf{X})$$



3d. Bundle adjustment

- Given:
  - Vectors of cameras and 3D points

$$C = (C_1, C_2, \ldots, C_n)$$
$$X = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m)$$

  - A set of observed point projections

$\mathbf{q}_{ij}$ – the observed 2D location of point *j* in image *i*

adjust the cameras and points to minimize g, the sum of squared reprojection errors

Reprojection Error

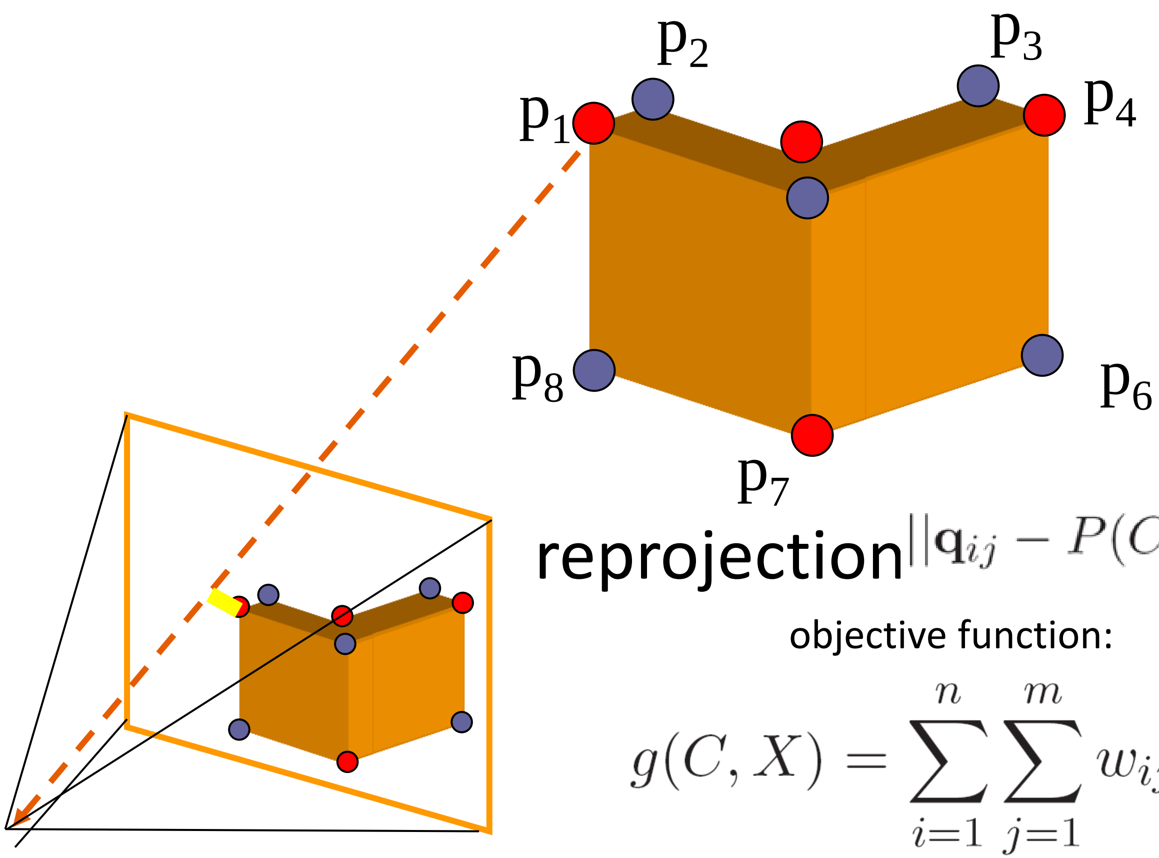


Image 1
R_1, t_1

reprojection $||\mathbf{q}_{ij} - P(C_i, \mathbf{X}_j)||$

objective function:

$$g(C, X) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} ||\mathbf{q}_{ij} - P(C_i, \mathbf{X}_j)||^2$$

indicator variable:
- 1 if point *j* is visible in camera *i*
- 0 otherwise

Objective function

$$g(C, X) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} \|\mathbf{q}_{ij} - P(C_i, \mathbf{X}_j)\|^2$$

Projection equation
(simplified version) :

$$P(C_i, \mathbf{X}_j) = \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix}$$

$$u_{ij} = -f_i \frac{[\mathbf{R}_i(\mathbf{X}_j - \mathbf{c}_i)]_x}{[\mathbf{R}_i(\mathbf{X}_j - \mathbf{c}_i)]_z}$$

$$v_{ij} = -f_i \frac{[\mathbf{R}_i(\mathbf{X}_j - \mathbf{c}_i)]_y}{[\mathbf{R}_i(\mathbf{X}_j - \mathbf{c}_i)]_z}$$

26

Bundle adjustment

$$\underset{C,X}{\text{minimize}} \quad g(C,X) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}||\mathbf{q}_{ij} - P(C_i, \mathbf{X}_j)||^2$$

$$g(C,X) = ||\mathbf{Q} - \mathbf{P}(C,X)||^2$$

- Minimizing g is a sparse non-linear least squares problem
- Usual approach: approximate **P** with a linear function , minimize using linear least squares, and repeat u $\tilde{\mathbf{P}}$ l convergence

Bundle adjustment

- Usual approach: approximate **P** by linearizing around a current guess C_0, X_0

$$\mathbf{P}(C_0 + \Delta C, X_0 + \Delta X) \approx \underbrace{\mathbf{P}(C_0, X_0) + \mathbf{J} \begin{bmatrix} \Delta C \\ \Delta X \end{bmatrix}}_{\tilde{\mathbf{P}}}$$

where **J** is the Jacobian (matrix of partials), $\tilde{\mathbf{P}}$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial C} & \frac{\partial \mathbf{P}}{\partial X} \end{bmatrix}$$

Bundle adjustment

- Linearized problem: find the step $\Delta C, \Delta X$ that minimizes:

$$
\begin{aligned}
\tilde{g}(C_0 + \Delta C, X_0 + \Delta X) &= ||\mathbf{Q} - \tilde{\mathbf{P}}(C_0 + \Delta C, X_0 + \Delta X)||^2 \\
&= \left\| \mathbf{Q} - \mathbf{P}(C_0, X_0) - \mathbf{J} \begin{bmatrix} \Delta C \\ \Delta X \end{bmatrix} \right\|^2
\end{aligned}
$$

$$
\begin{aligned}
C_1 &= C_0 + \Delta C \\
X_1 &= X_0 + \Delta X
\end{aligned}
$$

Bundle adjustment

- How do we minimize:

$$\left\| \mathbf{Q} - \mathbf{P}(C_0, X_0) - \mathbf{J} \left[\begin{array}{c} \Delta C \\ \Delta X \end{array} \right] \right\|^2 \quad ?$$

- Least-squares solution to the overconstrained linear system

$$\mathbf{J} \left[\begin{array}{c} \Delta C \\ \Delta X \end{array} \right] = \mathbf{P}(C_0, X_0) - \mathbf{Q}$$

Bundle adjustment

$$\mathbf{J}\left[\begin{array}{c}\Delta C\\ \Delta X\end{array}\right] = \mathbf{P}(C_0, X_0) - \mathbf{Q}$$

- (Over-constrained as long as
  2 x numObservations > 7 x numCameras + 3 x numPoints)

- Solved using the *normal equations*

$$\mathbf{J}^T\mathbf{J}\left[\begin{array}{c}\Delta C\\ \Delta X\end{array}\right] = \mathbf{J}^T\left(\mathbf{P}(C_0, X_0) - \mathbf{Q}\right)$$

Bundle adjustment

- Guess an answer
- Linearize and compute an optimal step
- Relinearize and repeat

- This algorithm is known as *Gauss-Newton*
- In practice, a modified algorithm known as *Levenberg-Marquardt* is used

Bundle adjustment

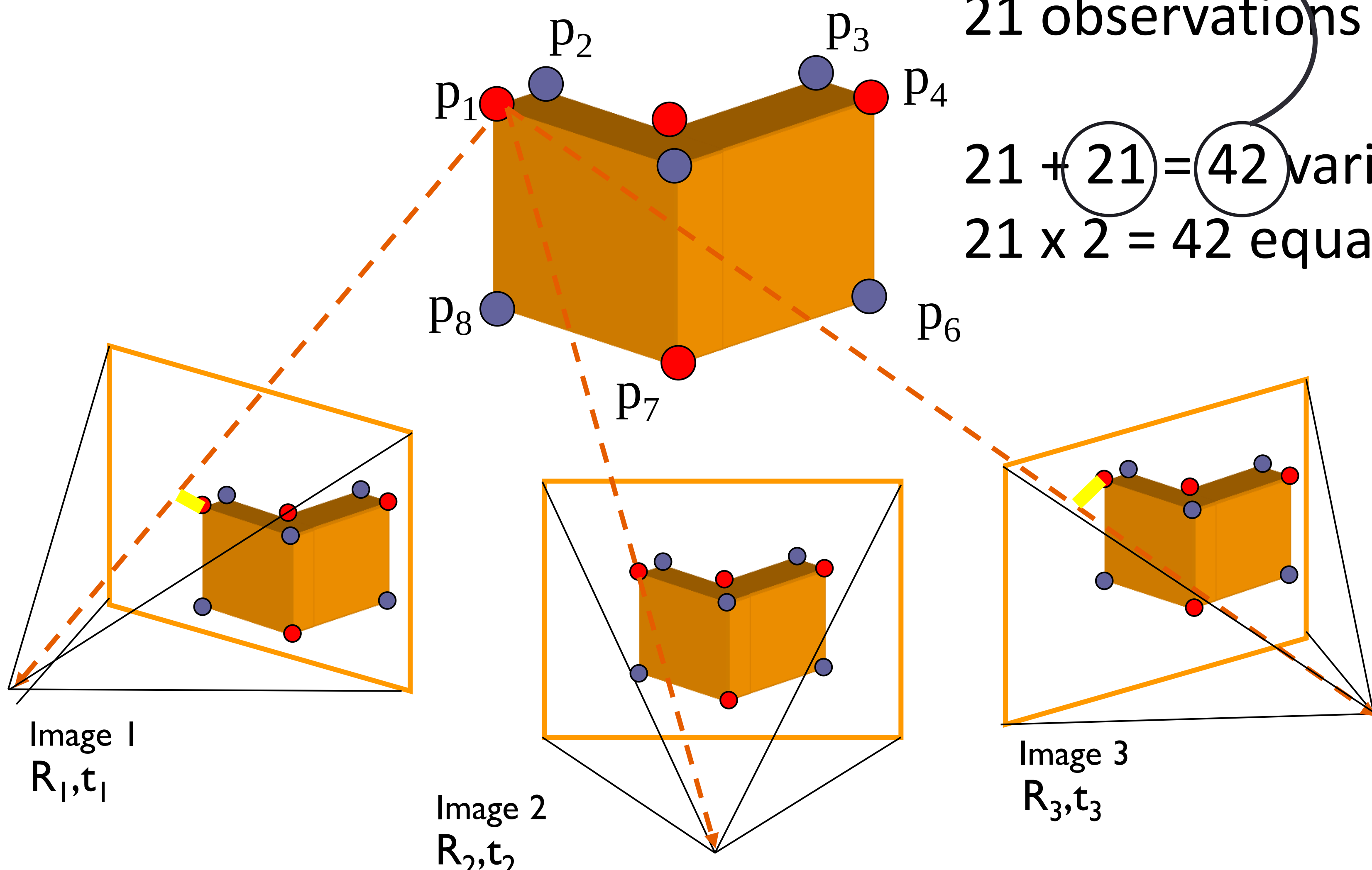
7 points

3 cameras

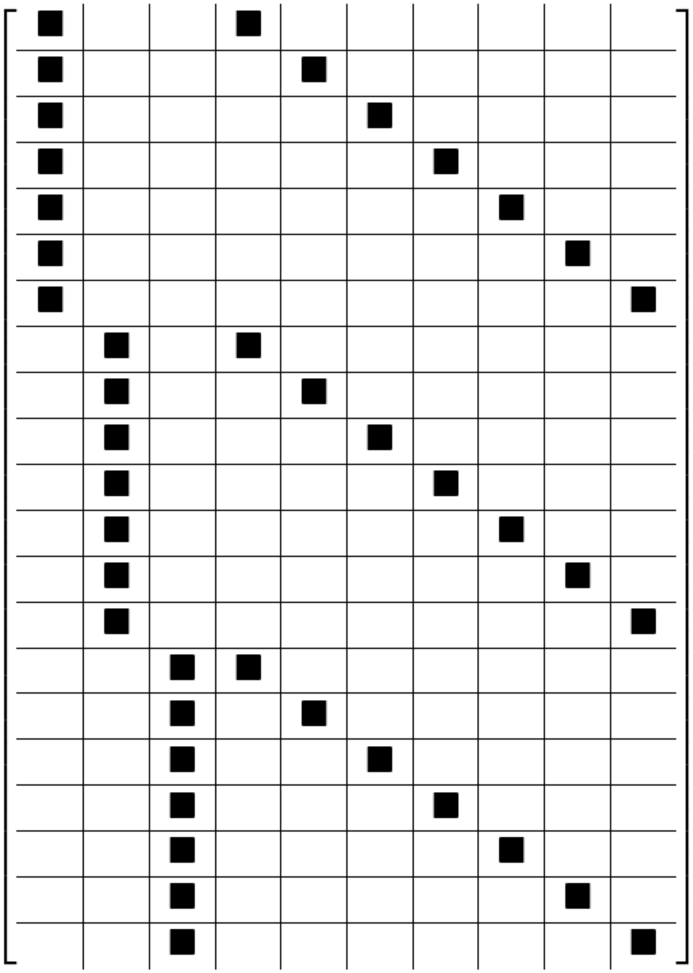
21 observations

21 + 21 = 42 variables

21 x 2 = 42 equations



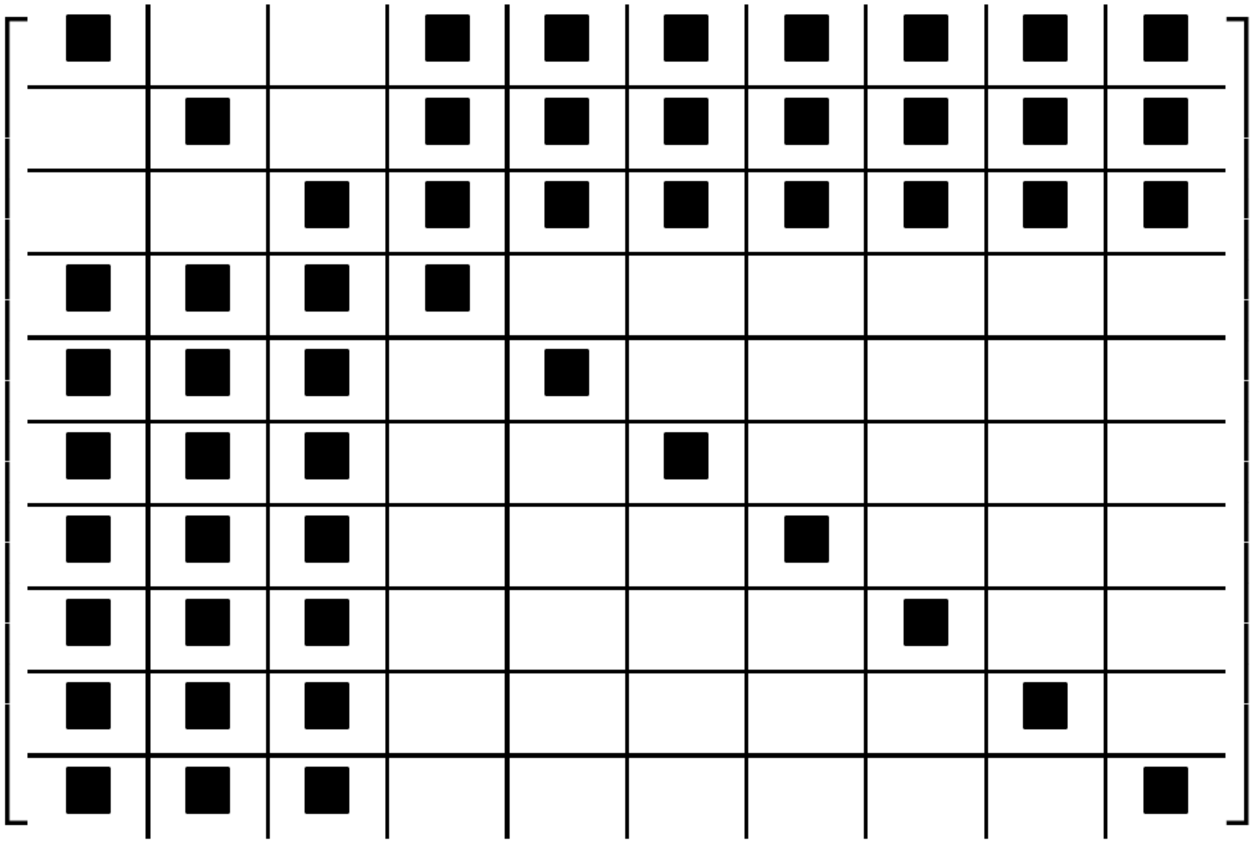
Bundle adjustment



$$\begin{bmatrix} \Delta C_1 \\ \Delta C_2 \\ \Delta C_3 \\ \Delta X_1 \\ \Delta X_2 \\ \Delta X_3 \\ \Delta X_4 \\ \Delta X_5 \\ \Delta X_6 \\ \Delta X_7 \end{bmatrix} = \begin{bmatrix} P(C_1, \mathbf{X}_1) - \mathbf{q}_{11} \\ P(C_1, \mathbf{X}_2) - \mathbf{q}_{12} \\ P(C_1, \mathbf{X}_3) - \mathbf{q}_{13} \\ \vdots \\ P(C_2, \mathbf{X}_1) - \mathbf{q}_{21} \\ P(C_2, \mathbf{X}_2) - \mathbf{q}_{22} \\ P(C_2, \mathbf{X}_3) - \mathbf{q}_{23} \\ \vdots \\ P(C_3, \mathbf{X}_1) - \mathbf{q}_{31} \\ P(C_3, \mathbf{X}_2) - \mathbf{q}_{32} \\ P(C_3, \mathbf{X}_3) - \mathbf{q}_{33} \\ \vdots \end{bmatrix}$$

$$\mathbf{J} \quad \begin{bmatrix} \Delta C \\ \Delta X \end{bmatrix} = \mathbf{P}(C_0, X_0) - \mathbf{Q}$$

Bundle adjustment

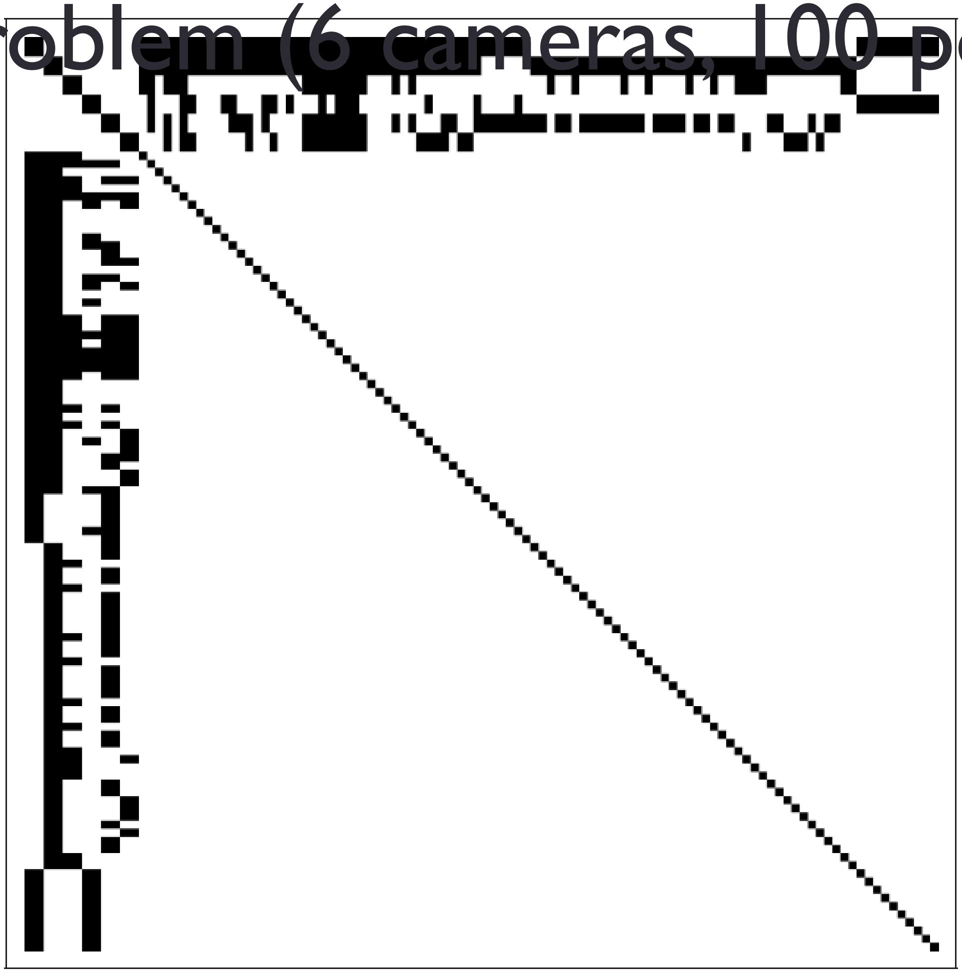


$$\begin{bmatrix} \Delta C_1 \\ \Delta C_2 \\ \Delta C_3 \\ \Delta X_1 \\ \Delta X_2 \\ \Delta X_3 \\ \Delta X_4 \\ \Delta X_5 \\ \Delta X_6 \\ \Delta X_7 \end{bmatrix}$$

$$\mathbf{J}^T\mathbf{J}$$

$$\begin{bmatrix} \Delta C \\ \Delta X \end{bmatrix}$$

Typical problem (6 cameras, 100 points)



36