Computer Vision and Image Processing

Gil Ben-Artzi

Structure

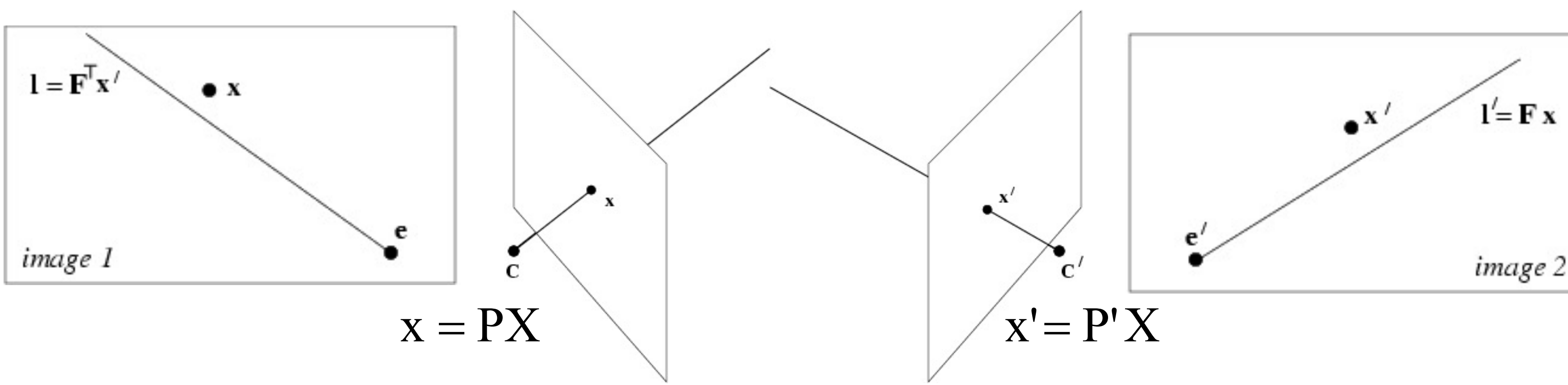
3D Structure

Slides:Derek Hoiem

Triangulation

Triangulation

- Given camera matrices P,P’ and corresponding image pixels, we would like to recover the 3D point that projects onto the pixels
- The problem – the pixels matching is not exact and therefore their rays do not intersect



Triangulation: Linear Solution

- How can we model the problem?
- How can we solve it?

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \mathbf{x}' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{bmatrix} \quad \mathbf{P}' = \begin{bmatrix} \mathbf{p}_1'^T \\ \mathbf{p}_2'^T \\ \mathbf{p}_3'^T \end{bmatrix}$$

$$\mathrm{x}' = \mathrm{P}'\mathrm{X} \qquad \mathbf{x} \times (\mathbf{P}\mathbf{X}) = 0$$

$$\mathrm{x} = \mathrm{P}\mathrm{X} \qquad \mathbf{x}' \times (\mathbf{P}'\mathbf{X}) = 0$$

$$\Rightarrow \quad \mathbf{A}\mathbf{X} = \mathbf{0} \quad \mathbf{A} = \begin{bmatrix} u\mathbf{p}_3^T - \mathbf{p}_1^T \\ v\mathbf{p}_3^T - \mathbf{p}_2^T \\ u'\mathbf{p}_3'^T - \mathbf{p}_1'^T \\ v'\mathbf{p}_3'^T - \mathbf{p}_2'^T \end{bmatrix}$$

See: HZ p. 312-313

Triangulation: Linear Solution

Given **P**, **P'**, **x**, **x'**

1. Precondition points and projection matrices
2. Create matrix **A**
3. [U, S, V] = svd(A)
4. **X** = V(:, end)

Pros and Cons

- It often provides acceptable results
- Works for any number of corresponding images
- But it is not projectively invariant (P->PH)

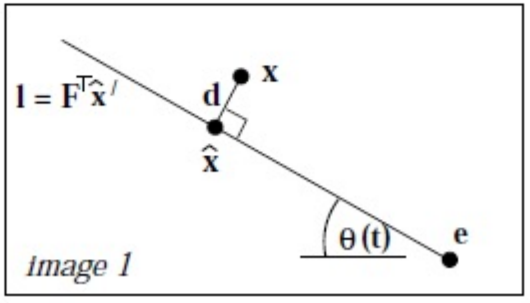
Code: http://www.robots.ox.ac.uk/~vgg/hzbook/code/vgg_multiview/vgg_X_from_xP_lin.m

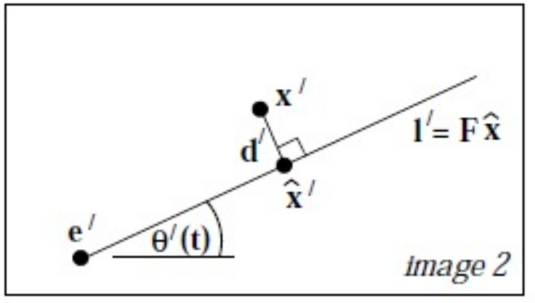
Triangulation: Non-linear Solution

- Minimize projected error

$$d(\mathrm{x}, \hat{\mathrm{x}})^2 + d(\mathrm{x}', \hat{\mathrm{x}}')^2 \text{ subject to } \hat{\mathrm{x}}'^{\mathrm{T}}\, \mathrm{F}\hat{\mathrm{x}} = 0$$

$$\text{or equivalently subject to } \hat{\mathrm{x}} = \mathrm{P}\hat{\mathrm{X}} \text{ and } \hat{\mathrm{x}}' = \mathrm{P}'\hat{\mathrm{X}}$$





- Solution is a 6-degree polynomial of *t*, minimizing

$$d(\mathbf{x}, \mathbf{l}(t))^2 + d(\mathbf{x}', \mathbf{l}'(t))^2$$

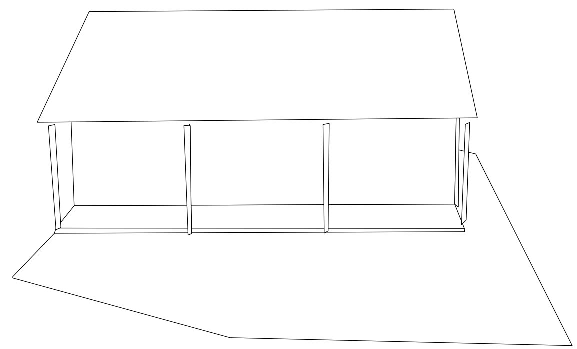
Further reading: HZ p. 318

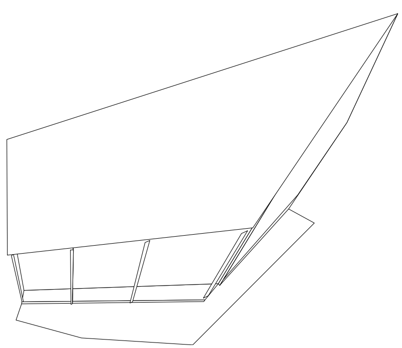
Projective Ambiguity

- In the general/uncalibrated case, cameras and points can only be recovered up to a projective ambiguity $\boldsymbol{x} = PQ^{-1}Q\boldsymbol{X}$
- In the calibrated case, they can be recovered up to a similarity (scale)
  - Known as Euclidean/metric reconstruction

Projective Ambiguity

- Projective vs. Euclidean





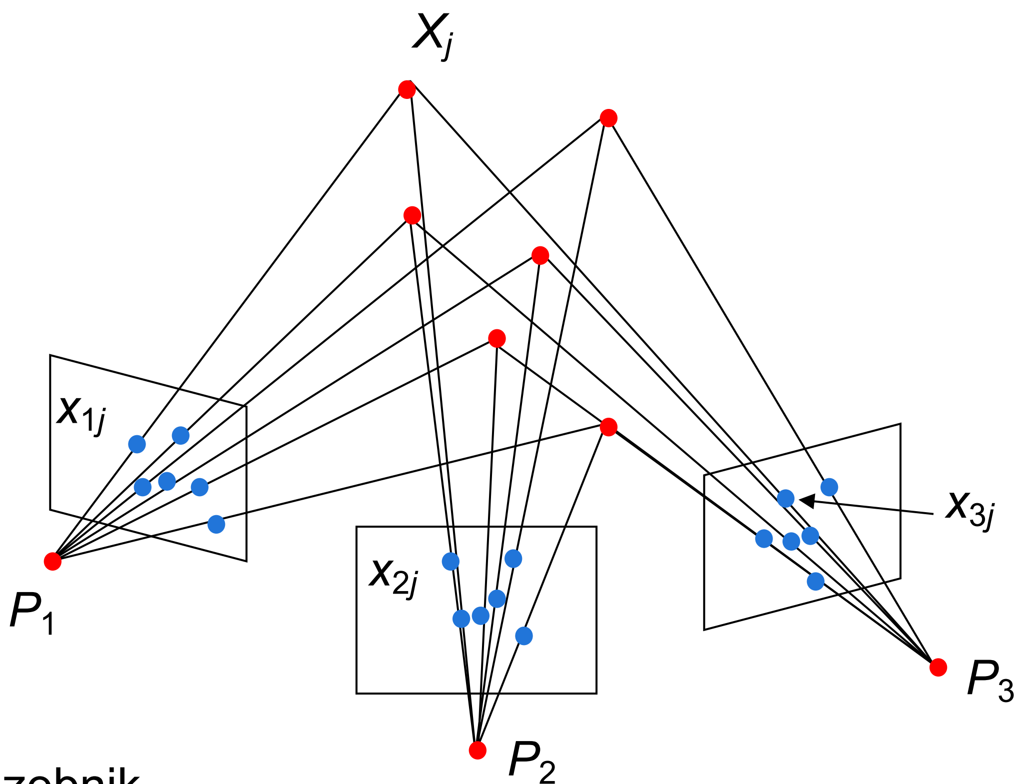
Structure From Motion (SfM)

Structure from Motion

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

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

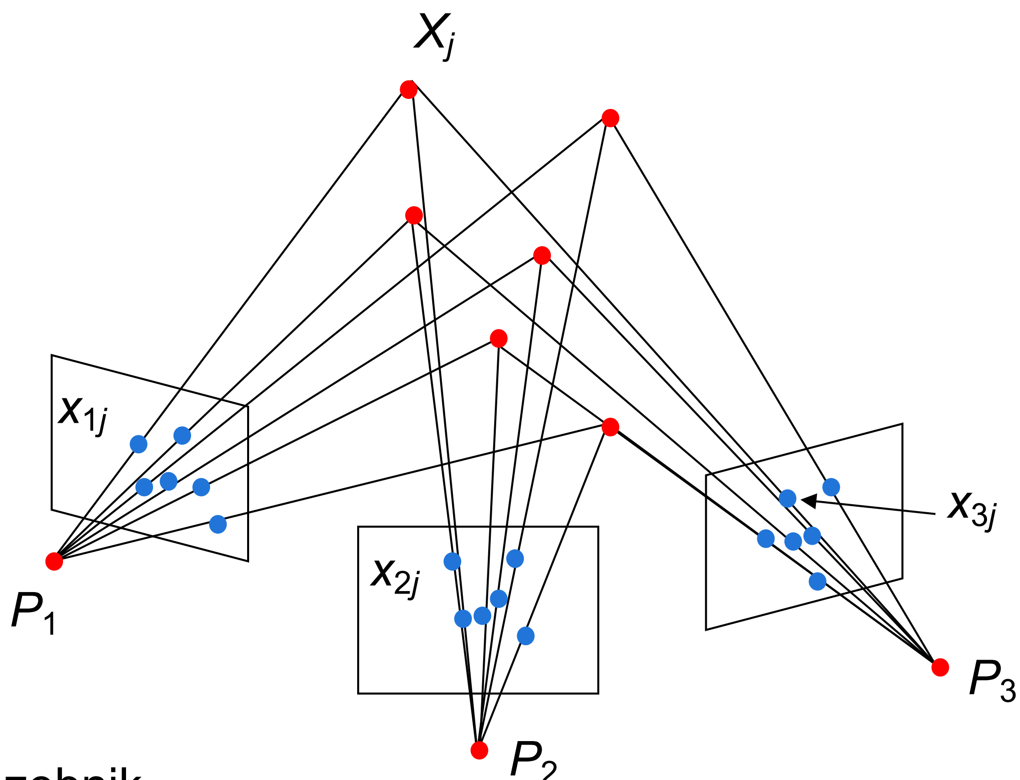
- Goal: estimate 3D points $\mathbf{X}_j$ from the *mn* corresponding 2D points $\mathbf{x}_{ij}$ and *m* projection matrices $\mathbf{P}_i$



Slides from Lana Lazebnik

Structure from Motion

- Structure are $\mathbf{X}_j$
- Motion are the rotation and translation (pose) extracted from the matrices $\mathbf{P}_i$



Slides from Lana Lazebnik

Structure from Motion

- Recall that 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

N-Views Linear Triangulation

- We can extend the 2 views triangulation approach to N cameras viewing the same points

$$\mathbf{x} \times (\mathbf{P}\mathbf{X}) = 0 \qquad \mathbf{x}' \times (\mathbf{P}'\mathbf{X}) = 0 \qquad \mathbf{x}'' \times (\mathrm{P}''\mathrm{X}) = 0$$

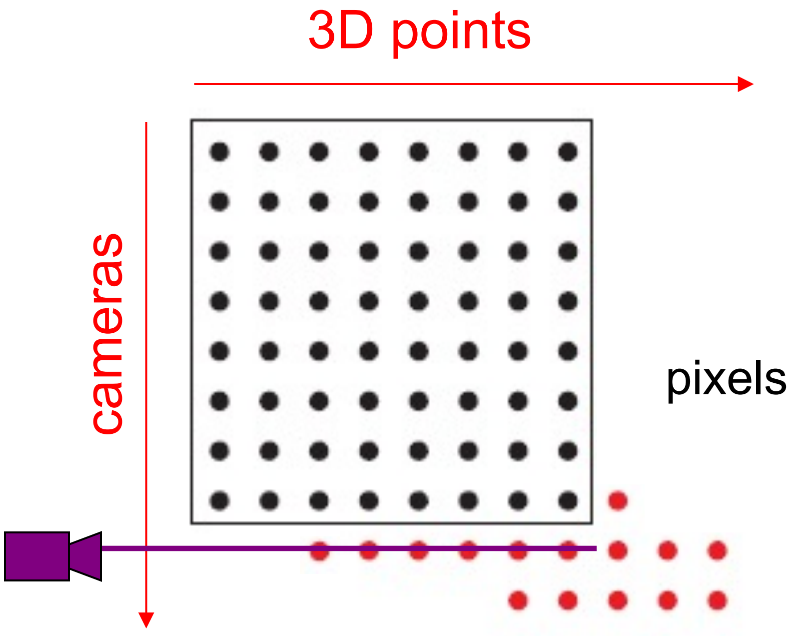
- Stack equations in a matrix to get homogenous LS

Sequential Sfm

- Initialize motion (calibration) from two images using fundamental matrix/essential 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/resectioning*



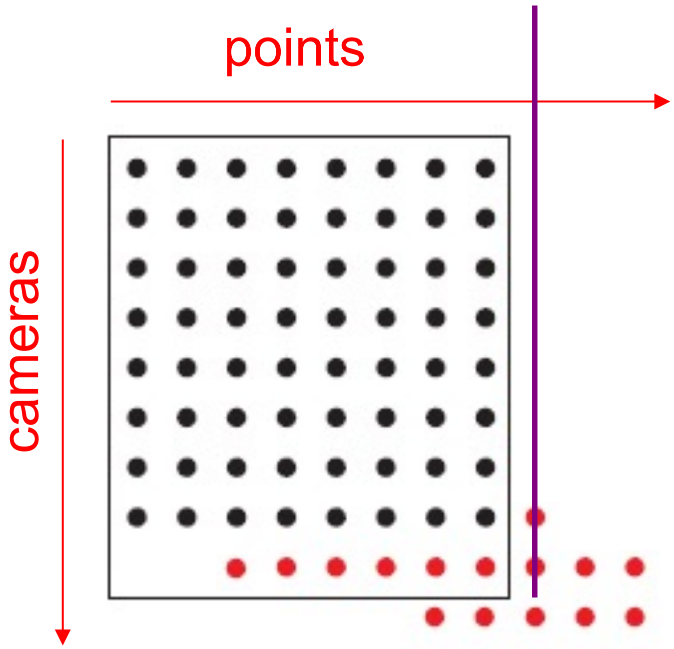
Sequential Sfm

- 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*



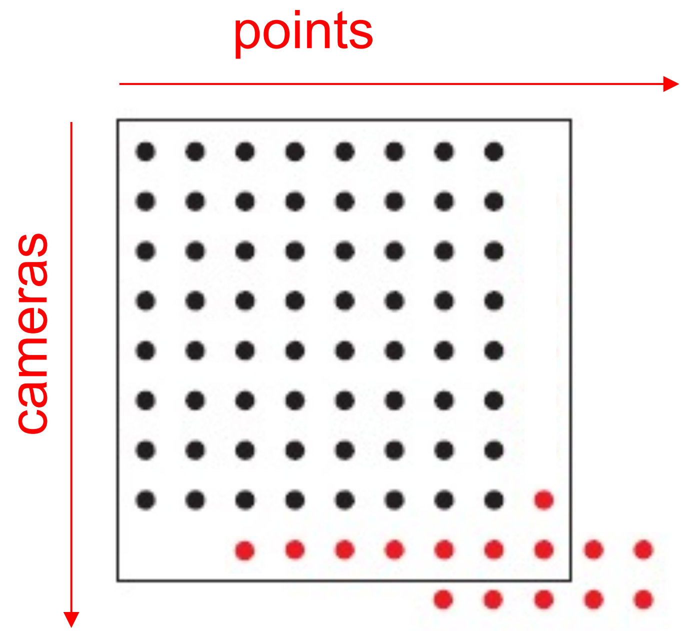
Sequential Sfm

- 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*

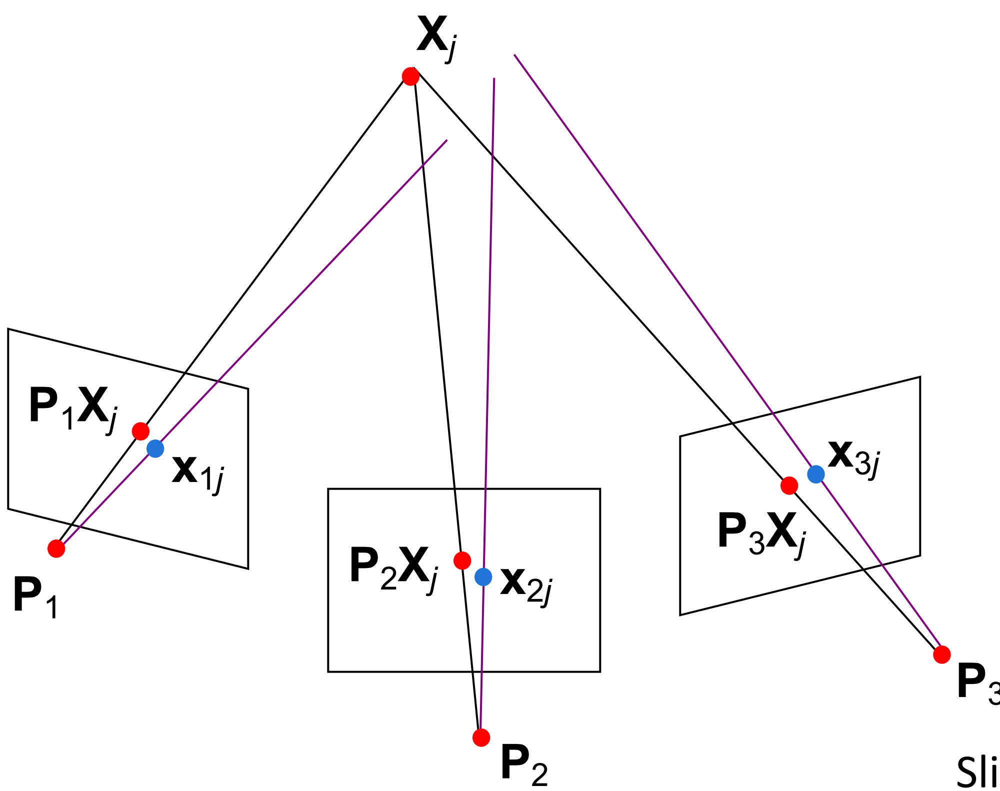


- Refine structure and motion: bundle adjustment

Bundle adjustment

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

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



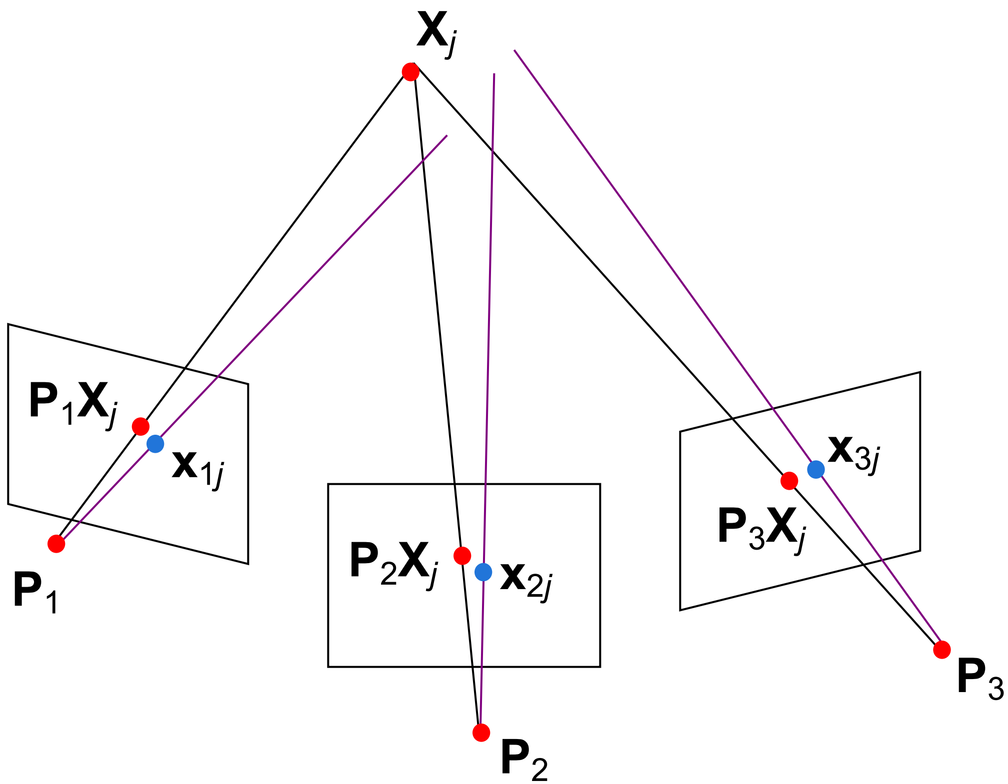
Slides from Derek Hoiem

Bundle adjustment

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

$$(x_{ij}^1, x_{ij}^2) = \left(\frac{P_i^1 \mathbf{X}_j}{P_i^3 \mathbf{X}_j}, \frac{P_i^2 \mathbf{X}_j}{P_i^3 \mathbf{X}_j} \right)$$

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| \left(x_{ij}^1 - \frac{P_i^1 \mathbf{X}_j}{P_i^3 \mathbf{X}_j},\ x_{ij}^2 - \frac{P_i^2 \mathbf{X}_j}{P_i^3 \mathbf{X}_j} \right) \right\|^2 .$$



Important recent papers and methods for SfM

- OpenMVG
  - https://github.com/openMVG/openMVG
  - http://imagine.enpc.fr/~moulonp/publis/iccv2013/index.html (Moulin et al. ICCV 2013)
  - Software has global and incremental methods

- OpenSfM (software only): https://github.com/mapillary/OpenSfM
  - Basis for my description of incremental SfM

- Visual SfM: Visual SfM (Wu 2013)
  - Used to be the best incremental SfM software (but not anymore and closed source); paper still very good

Reconstruction of Cornell (Crandall et al. ECCV 2011)

Where does SfM fail?

- Not enough images with enough overlap
  - Disconnected reconstructions
- No matches or bad matches
  - Repeated structures (buildings or bridges)
  - reflecting surfaces
- Images with pure rotations
  - Recovery of “F” can fail or bad pose reconstruction

Robust Estimation

Robust Model Estimation

- Goal
  - Estimate any computer vision/image processing model in the presence of large error in the data

Robust Model Estimation

- We have seen several models
  - Camera Calibration: K
  - Camera Pose: R,T
  - Essential Matrix
  - Fundamental Matrix
  - Homography
  - Triangulation
  - Structure from Motion

Robust Model Estimation

- What is the general algorithm for the estimation of such model in the presence of outliers
- Let's take the case of camera calibration
- We need
  - Model: P matrix
  - Data: matching 3D and 2D(pixels): $\{(X_i,x_i)\}_{i=1..N}$
  - Equations based on the data x=PX
  - Error function: given the model P, how can we estimate its quality? for (X,x), |PX-x| should be minimum
- We can then use RANSAC
  - Sample minimal set of data points
  - Infer model
  - Evaluate its quality using the error function
  - Do it N times and select the best model