CS231A Midterm Review

Friday 5/6/2016

Outline

- General Logistics
- Camera Models
- Non-perspective cameras
- Calibration
- Single View Metrology
- Epipolar Geometry
- Structure from Motion
- Active Stereo and Volumetric Stereo
- Fitting and Matching
  - RANSAC
  - Hough Transform
- Detectors and Descriptors

Midterm Logistics

- In class midterm at Skilling Aud. 3:00pm-4:20pm on **Monday** 5/9/2016
- SCPD students not taking exam at Stanford should have already set up a proctor.
- Exam covers material from lecture 1-10 (through detectors and descriptors)
- Approximately 10 TF, 5 MC, 8 Short Answer.
- Open book, open notes, open computer without network access

Camera Models

Homogeneous coordinates

E→H

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

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

homogeneous scene coordinates

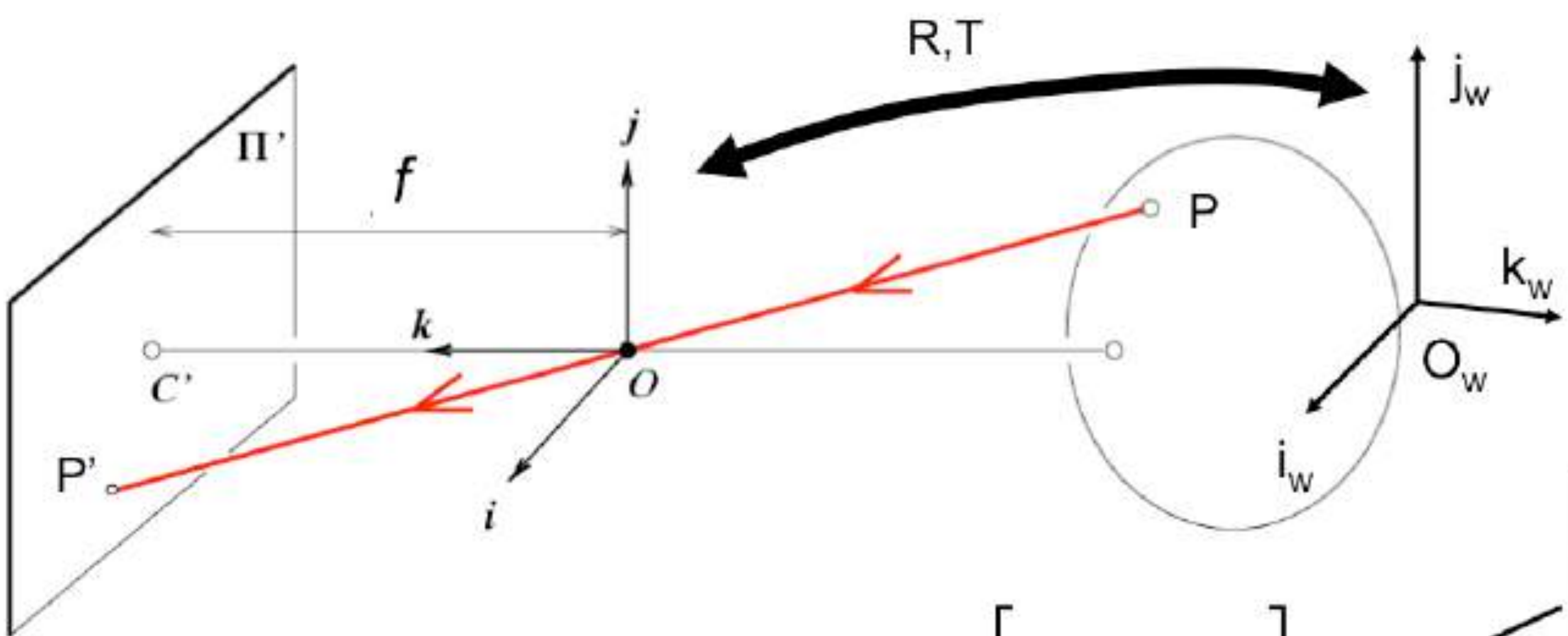
- Converting back *from* homogeneous coordinates

H→E

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

World reference system



In 4D homogeneous coordinates: $P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times 4} P_w$, where $P_w = \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$

Internal parameters

External parameters

$$P' = K\begin{bmatrix} I & 0 \end{bmatrix} P = K\begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times 4} P_w = \boxed{K\begin{bmatrix} R & T \end{bmatrix}} P_w$$

M

[Eq.11]

Projective camera

$$P' = M\,P_w = \underbrace{K}_{\text{Internal parameters}}\underbrace{\begin{bmatrix} R & T \end{bmatrix}}_{\text{External parameters}} P_w$$

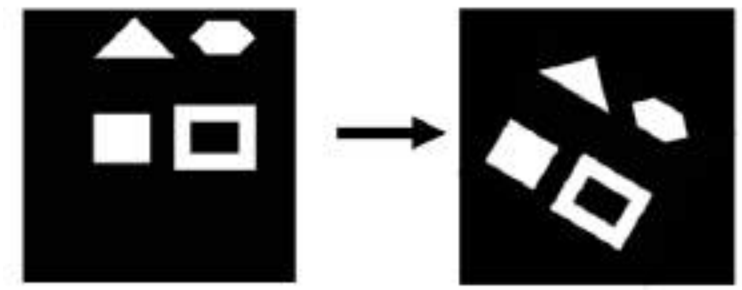
$$\boxed{\mathcal{M} = \begin{pmatrix} \alpha \boldsymbol{r}_1^T - \alpha \cot\theta \boldsymbol{r}_2^T + u_0 \boldsymbol{r}_3^T & \alpha t_x - \alpha \cot\theta t_y + u_0 t_z \\ \dfrac{\beta}{\sin\theta}\boldsymbol{r}_2^T + v_0 \boldsymbol{r}_3^T & \dfrac{\beta}{\sin\theta} t_y + v_0 t_z \\ \boldsymbol{r}_3^T & t_z \end{pmatrix}_{3\times 4}}$$

$$K = \begin{bmatrix} \boldsymbol{\alpha} & -\boldsymbol{\alpha}\cot\boldsymbol{\theta} & u_o \\ 0 & \dfrac{\boldsymbol{\beta}}{\sin\boldsymbol{\theta}} & v_o \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Transformations in 2D

Isometric

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

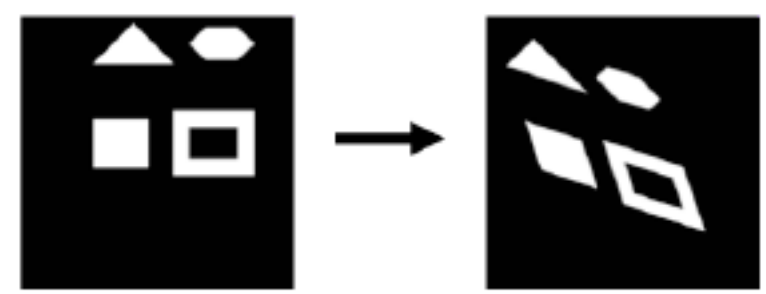


Similarity

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} SR & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Affine

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Projective

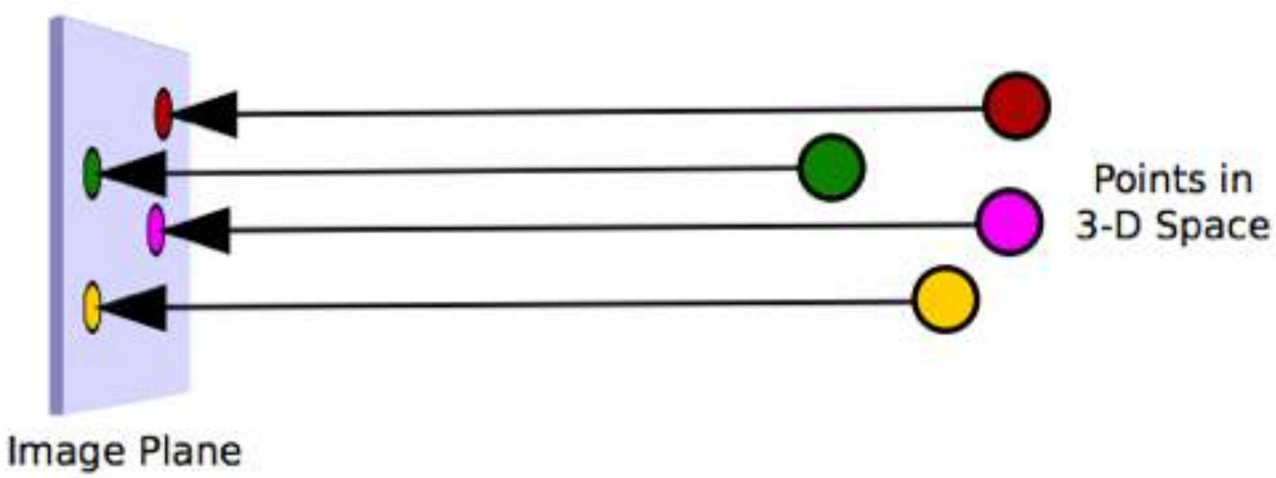
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A & t \\ v & b \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Properties of projective transformations

- points project to points
- lines project to lines
- distant objects look smaller
- angles are not preserved
- parallel lines in 3D meet in the image

Non-perspective cameras

- Orthographic



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

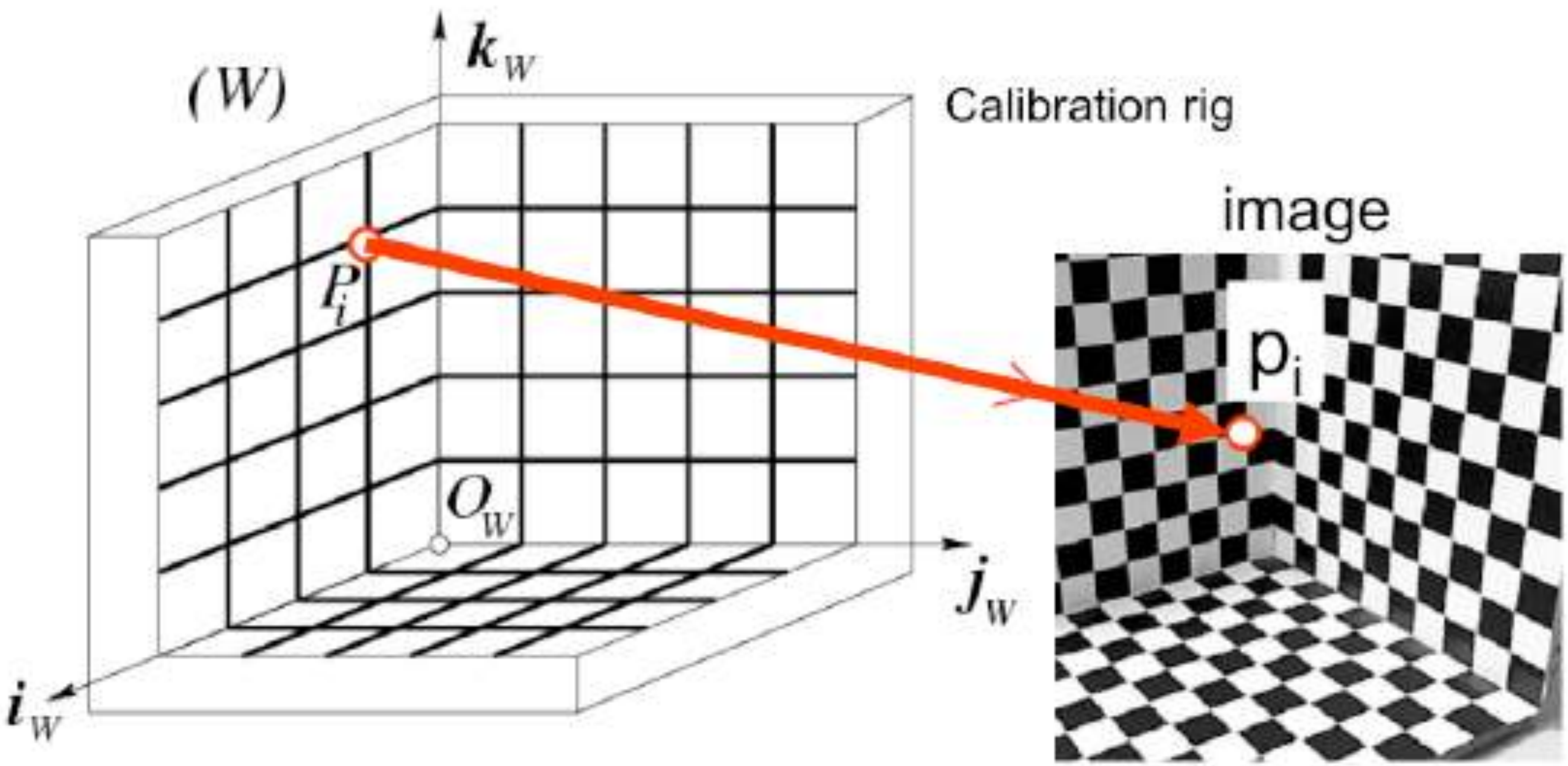
- Weak Perspective

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Affine

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

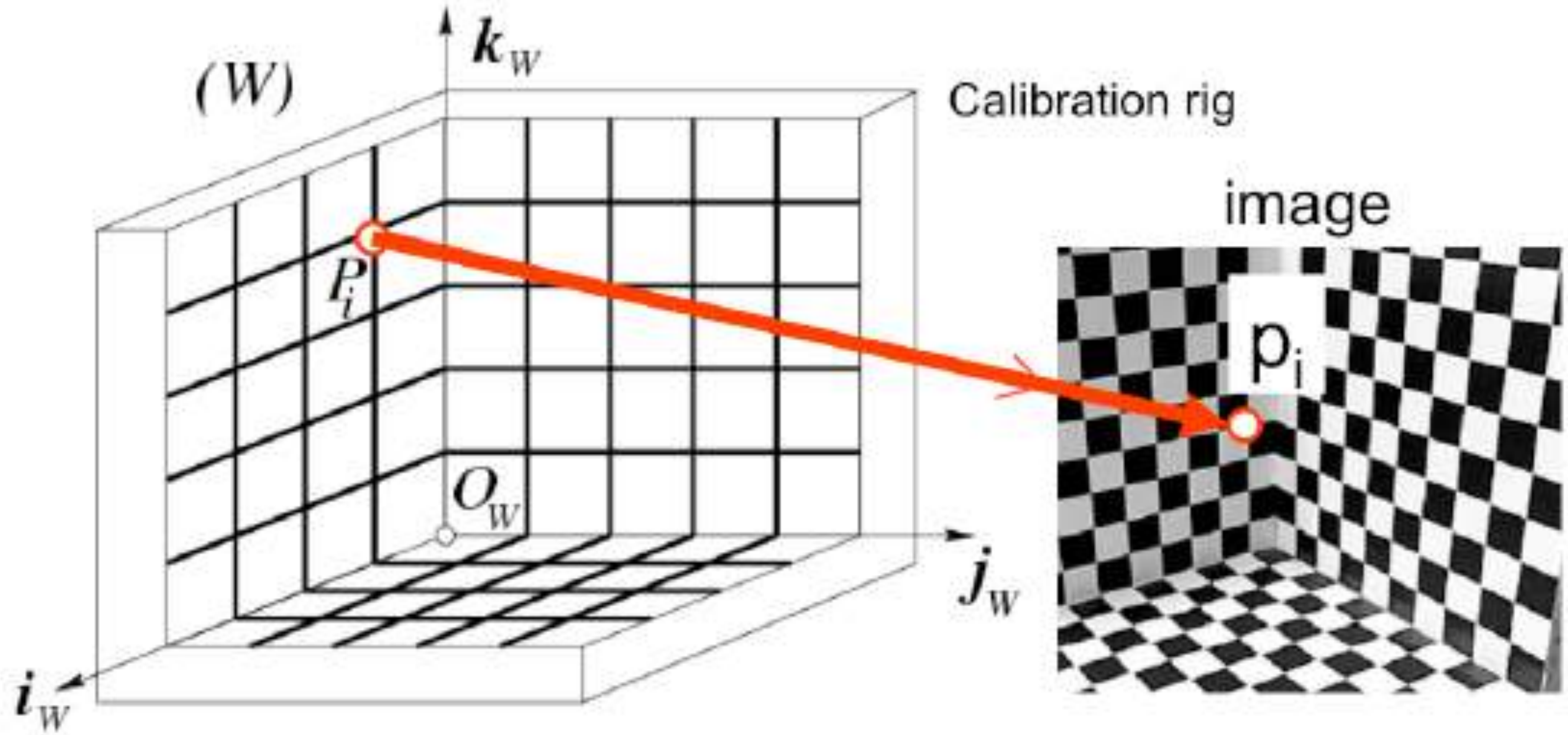
Calibration Problem



- $P_1 \ldots P_n$ with known positions in $[O_w, i_w, j_w, k_w]$
- $p_1, \ldots p_n$ known positions in the image

Goal: compute intrinsic and extrinsic parameters

Calibration Problem



$$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1 \, P_i}{\mathbf{m}_3 \, P_i} \\ \dfrac{\mathbf{m}_2 \, P_i}{\mathbf{m}_3 \, P_i} \end{bmatrix} = M \, P_i$$

in pixels

[Eq. 1]

$$M = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \mathbf{m}_3 \end{bmatrix}$$

Calibration Problem

[Eq. 1] $$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \\ \dfrac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \end{bmatrix}$$

$$u_i = \frac{\mathbf{m}_1 P_i}{\mathbf{m}_3 P_i} \rightarrow u_i(\mathbf{m}_3 P_i) = \mathbf{m}_1 P_i \rightarrow u_i(\mathbf{m}_3 P_i) - \mathbf{m}_1 P_i = 0$$

$$v_i = \frac{\mathbf{m}_2 P_i}{\mathbf{m}_3 P_i} \rightarrow v_i(\mathbf{m}_3 P_i) = \mathbf{m}_2 P_i \rightarrow v_i(\mathbf{m}_3 P_i) - \mathbf{m}_2 P_i = 0$$

Calibration Problem

$$\begin{cases} -u_1(\mathbf{m}_3\, P_1) + \mathbf{m}_1\, P_1 = 0 \\ -v_1(\mathbf{m}_3\, P_1) + \mathbf{m}_2\, P_1 = 0 \\ \vdots \\ -u_n(\mathbf{m}_3\, P_n) + \mathbf{m}_1\, P_n = 0 \\ -v_n(\mathbf{m}_3\, P_n) + \mathbf{m}_2\, P_n = 0 \end{cases} \longrightarrow \boxed{\underbrace{\mathbf{P}}_{\text{known}}\, \underbrace{\mathbf{m}}_{\text{unknown}} = 0}$$

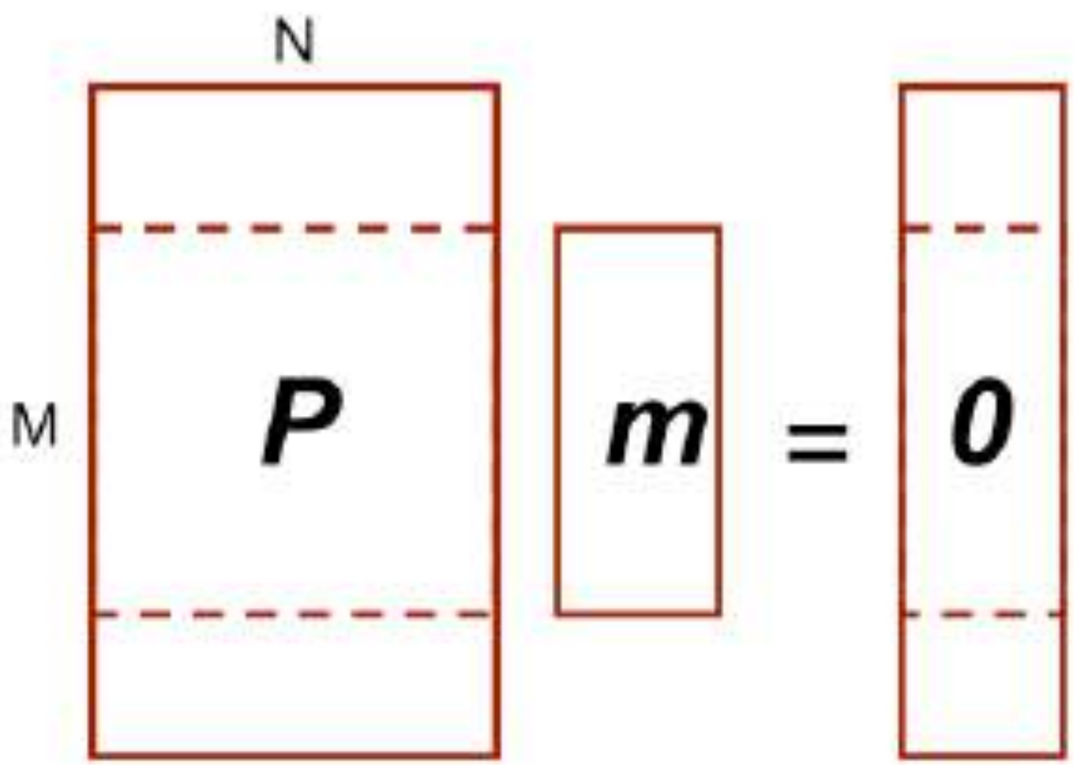
[Eq. 4]

Homogenous linear system

$$\mathbf{P} \stackrel{\text{def}}{=} \begin{pmatrix} \mathrm{P}_1^T & \mathbf{0}^T & -u_1\overbrace{\mathrm{P}_1^T}^{1\text{x}4} \\ \mathbf{0}^T & \mathrm{P}_1^T & -v_1\mathrm{P}_1^T \\ & \vdots & \\ \mathrm{P}_n^T & \mathbf{0}^T & -u_n\mathrm{P}_n^T \\ \mathbf{0}^T & \mathrm{P}_n^T & -u_n\mathrm{P}_n^T \end{pmatrix}_{2\text{n x }12}
\qquad
\boldsymbol{m} \stackrel{\text{def}}{=} \begin{pmatrix} \overbrace{\mathbf{m}_1^{\mathrm{T}}}^{4\text{x}1} \\ \mathbf{m}_2^{\mathrm{T}} \\ \mathbf{m}_3^{\mathrm{T}} \end{pmatrix}_{12\text{x}1}$$

Homogeneous MxN Linear System

M=number of equations = 2n
N=number of unknown = 11



Rectangular system (M>N)

- 0 is always a solution
- To find non-zero solution

  Minimize $|\mathbf{P}\ \mathbf{m}|^2$

  under the constraint $|\mathbf{m}|^2 = 1$

Quick SVD recap

Any real mxn matrix A can be decomposed uniquely as $A = U\Sigma V^T$ where U & V are orthogonal and Σ is diagonal.

If A has rank r,

$$A = U_r \Sigma_r V_r^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T \quad \begin{aligned} \Sigma_r &= diag(\sigma_1, \sigma_2, ..., \sigma_r) \\ U &= [u_1, u_2, ..., u_r] \\ V &= [v_1, v_2, ..., v_r] \end{aligned}$$

Be familiar with how to use SVD to solve Ax=0 and Ax=b

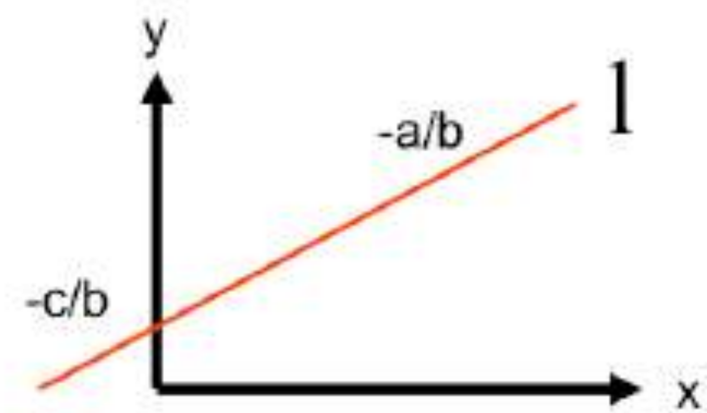
Single View Metrology

- Vanishing points
- Vanishing lines
- Construction of lines from points
- Directions and normals of vanishing points and vanishing lines.

Lines in a 2D plane

$$ax + by + c = 0$$

$$\mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



If $x = [\,x_1, x_2]^T \in l$

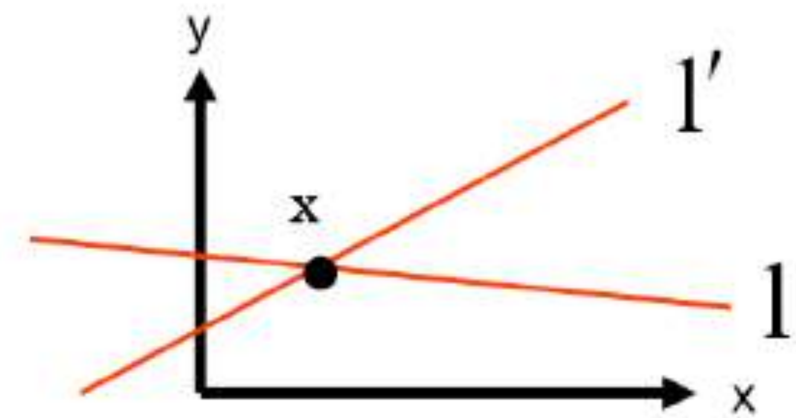
$$\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}^T \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0$$

[Eq. 10]

Points at infinity

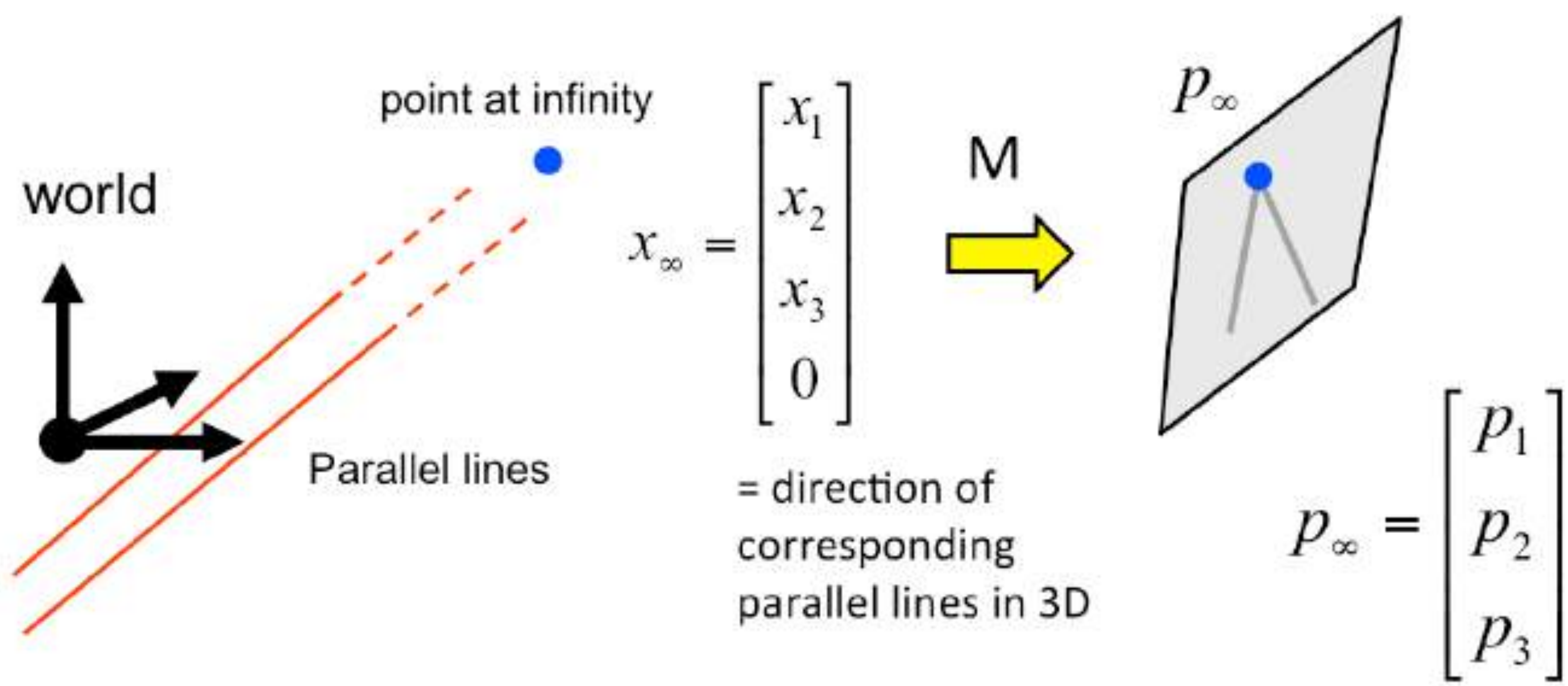
Intersection of two lines $x = l \times l'$

Line through two points $l = x \times x'$

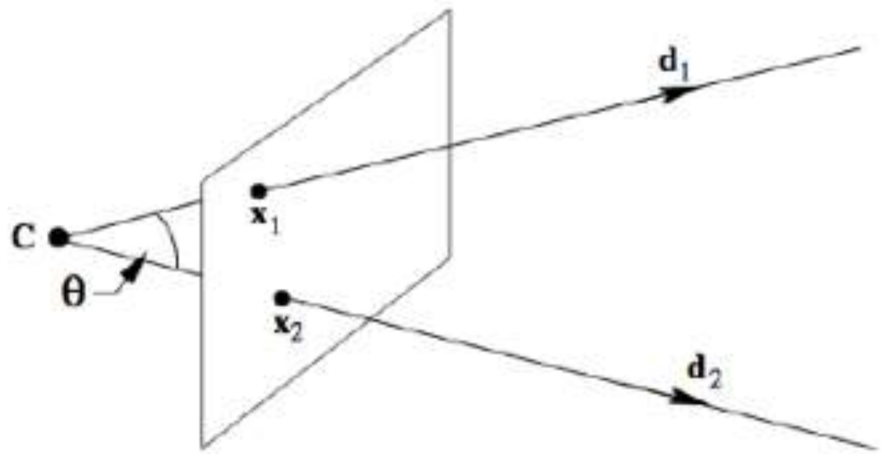


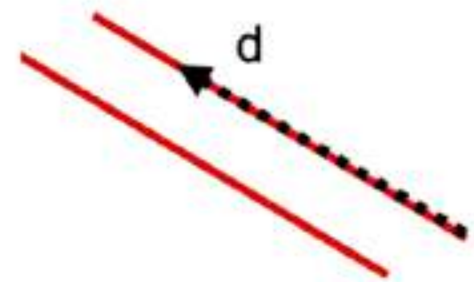
Point at infinity x_∞ Line at infinity l_∞

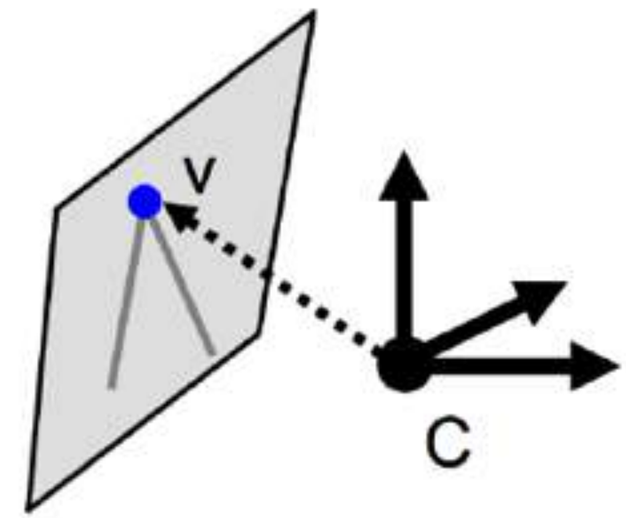
Vanishing point:
the projective projection of a point at infinity into the image plane



Vanishing points and directions





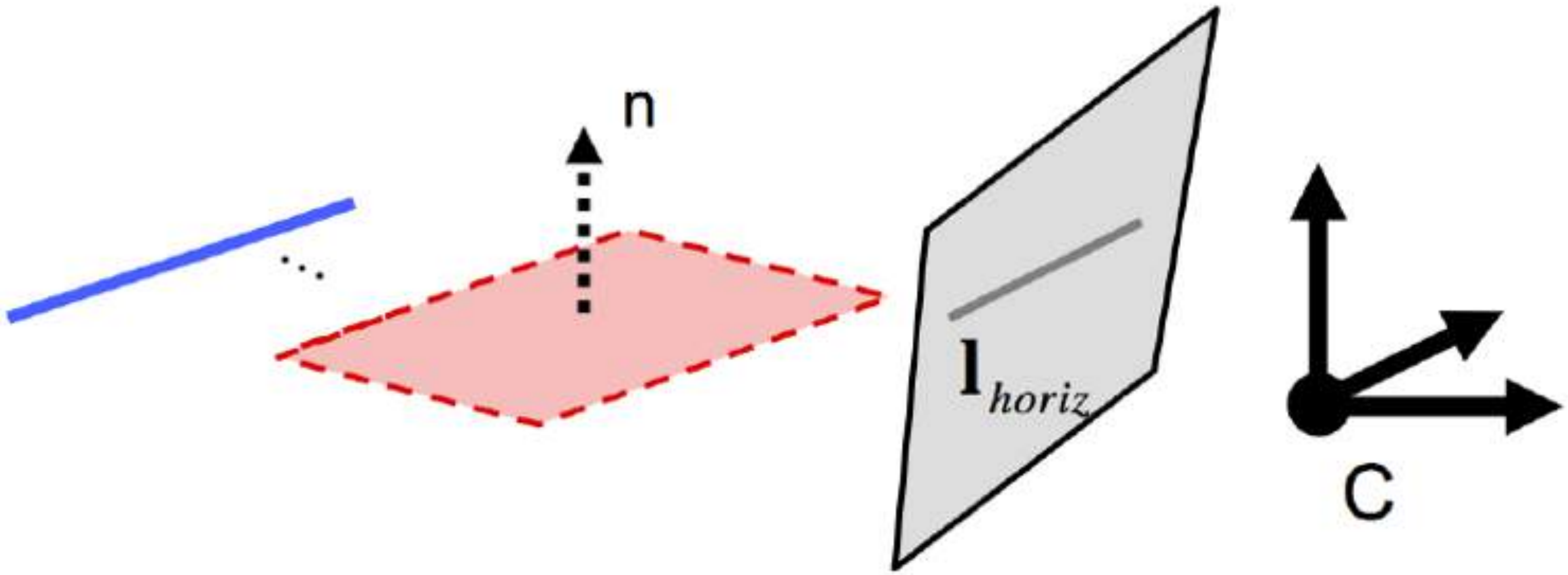


$$\mathbf{d}_1 = \frac{\mathrm{K}^{-1}\ \mathbf{v}_1}{\left\|\mathrm{K}^{-1}\ \mathbf{v}_1\right\|}$$

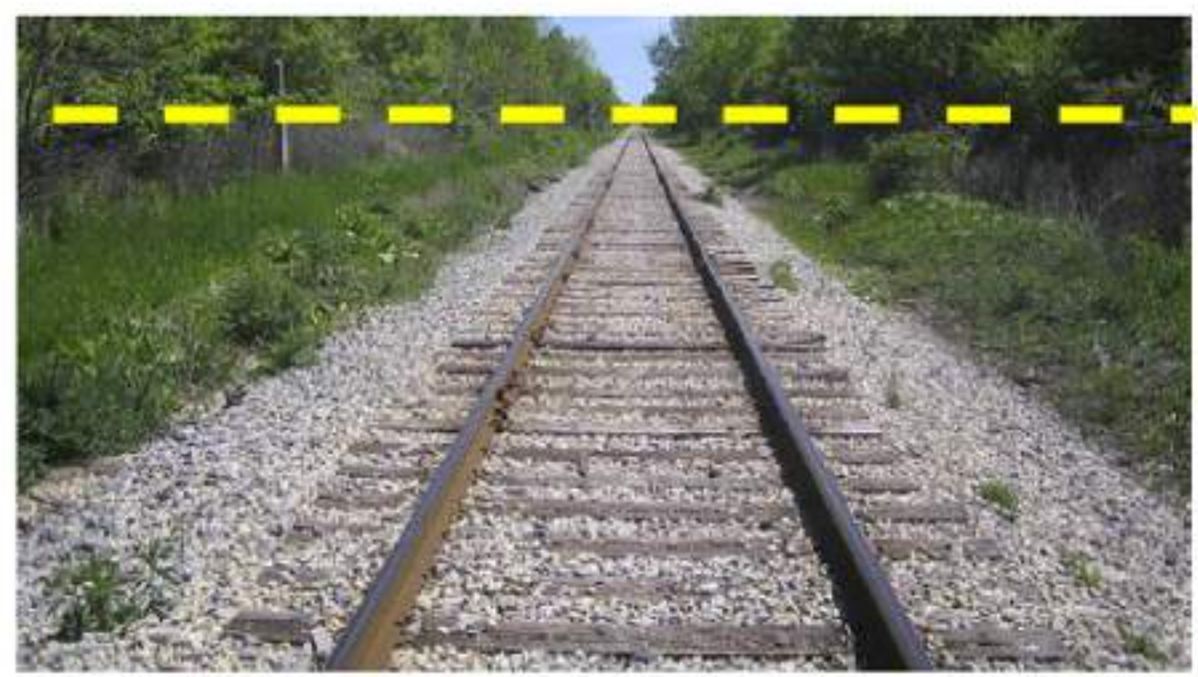
$$\mathbf{v} = \mathrm{K}\ \mathbf{d}$$

$$\mathbf{x}_\infty = \overset{\mathbf{d}}{\begin{bmatrix} \boxed{\begin{matrix} a \\ b \\ c \end{matrix}} \\ 0 \end{bmatrix}} \xrightarrow{M} \mathbf{v} = \mathbf{X}_\infty\ \mathbf{M} = \mathbf{K}[\mathbf{I} \quad \mathbf{0}]\begin{bmatrix} a \\ b \\ c \\ 0 \end{bmatrix} = \mathbf{K}\begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Vanishing lines



$$\mathbf{n} = \mathrm{K}^{T}\mathbf{l}_{\text{horiz}}$$



Single View Calibration

- As we saw in problem set 1, we can calculate the internal camera matrix K provided a single view and knowledge of the the scene geometry.

$$\omega = (K\,K^{T})^{-1}$$

$$\mathrm{v}_1^{\mathrm{T}} \boldsymbol{\omega}\, \mathrm{v}_2 = 0$$

$$\begin{cases} \mathrm{v}_1^{\mathrm{T}} \boldsymbol{\omega}\, \mathrm{v}_2 = 0 \\ \mathrm{v}_1^{\mathrm{T}} \boldsymbol{\omega}\, \mathrm{v}_3 = 0 \\ \mathrm{v}_2^{\mathrm{T}} \boldsymbol{\omega}\, \mathrm{v}_3 = 0 \end{cases}$$

Multi-view Geometry

- **Camera Geometry:** given corresponding points in two images, find camera matrices, position and pose.
- **Scene Geometry:** Find correspondences of 3D points from its projection into 2 or more images.
- **Correspondence:** Given a point p in one image, how can I find the corresponding point p’ in another?

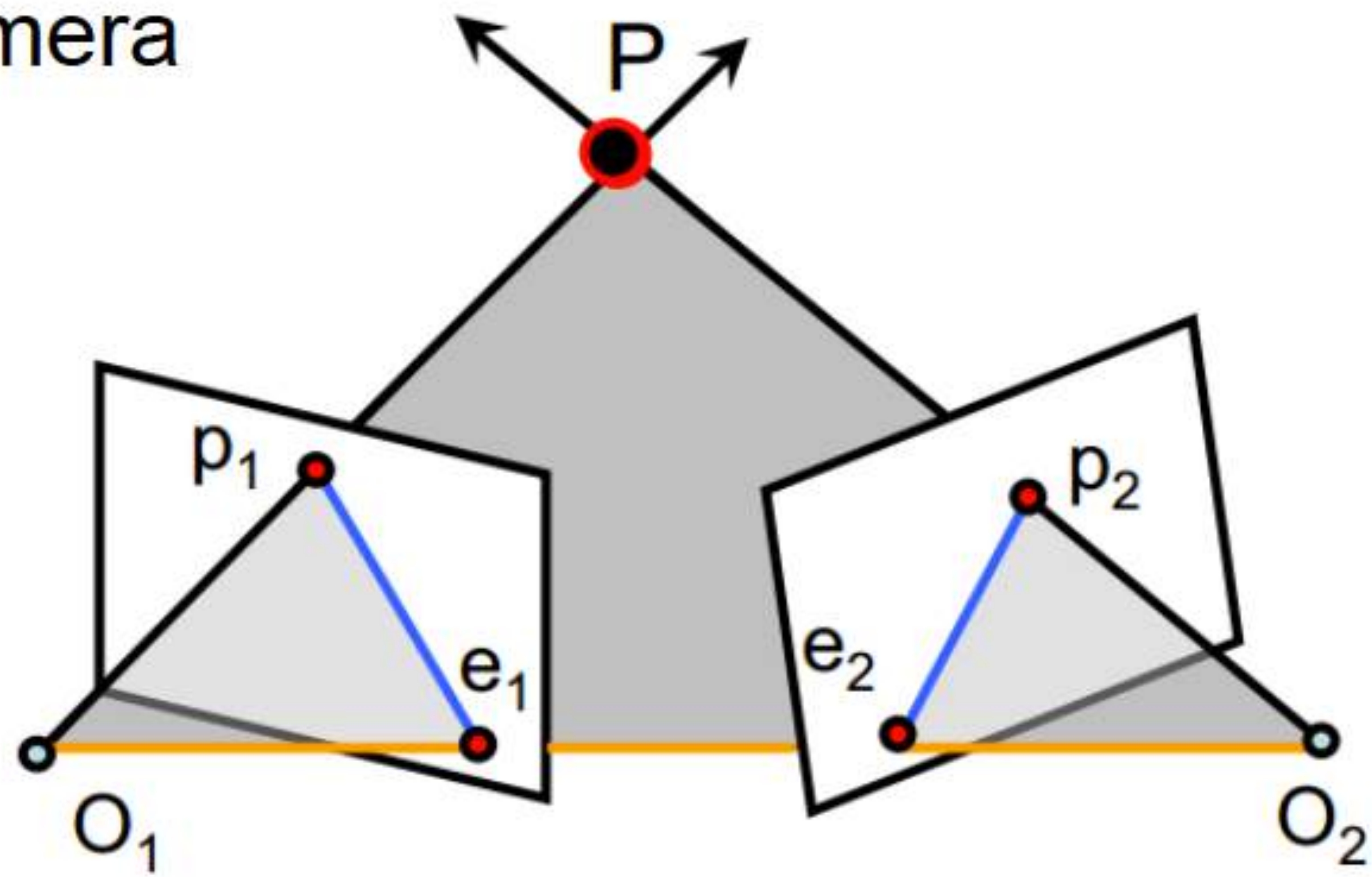
Epipolar Geometry

P: object

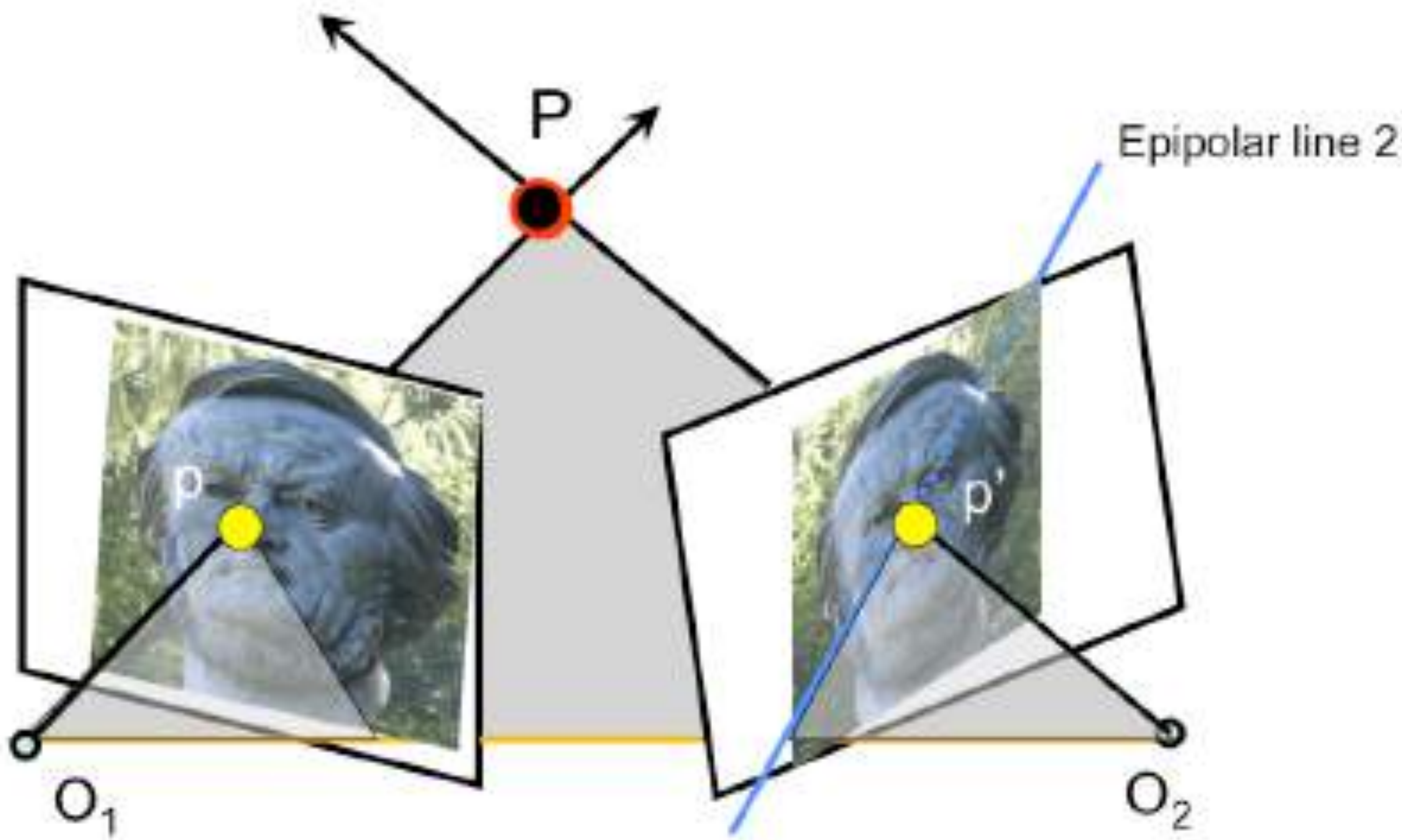
O: center of camera

p: image point

e: epipole



Epipolar Geometry



Fundamental Matrix

$$F = K_1^{-T} \cdot [T_x] \cdot R K_2^{-1}$$

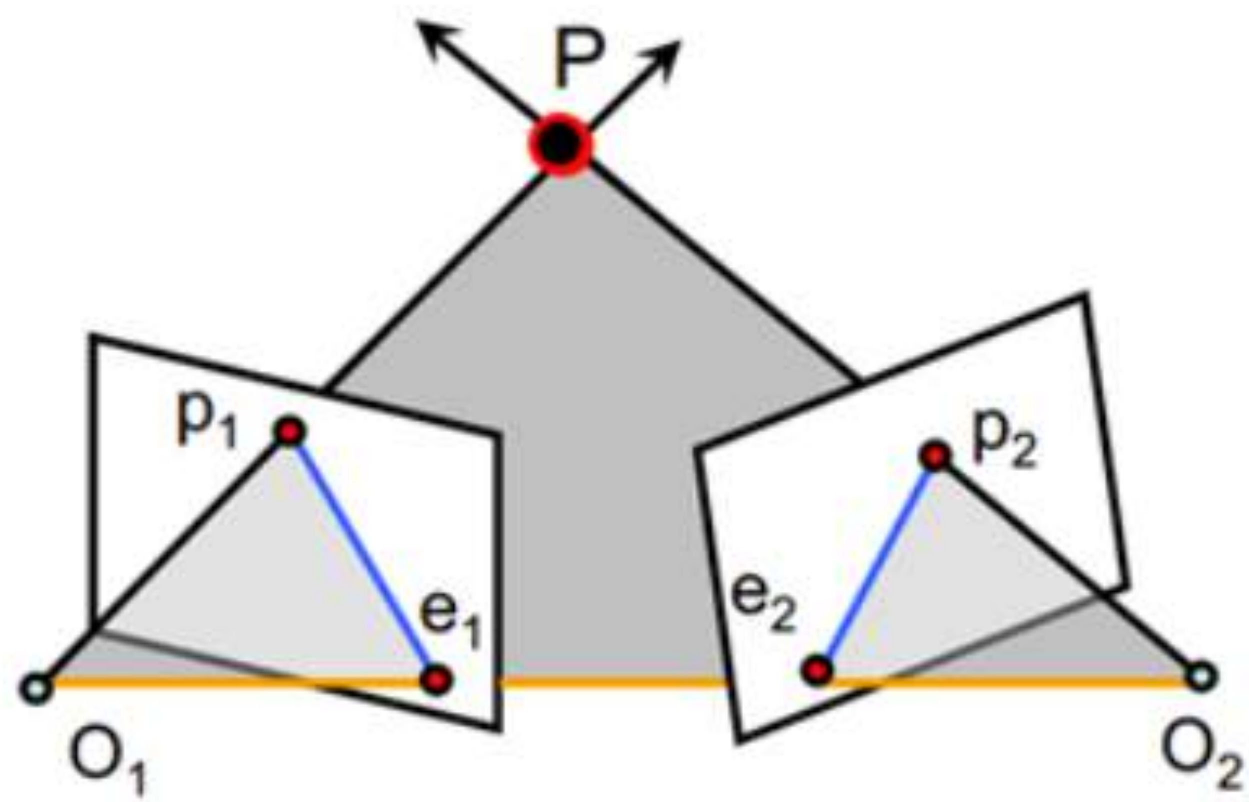
$$p_1^T \cdot F p_2 = 0$$

$$l_1 = F p_2$$

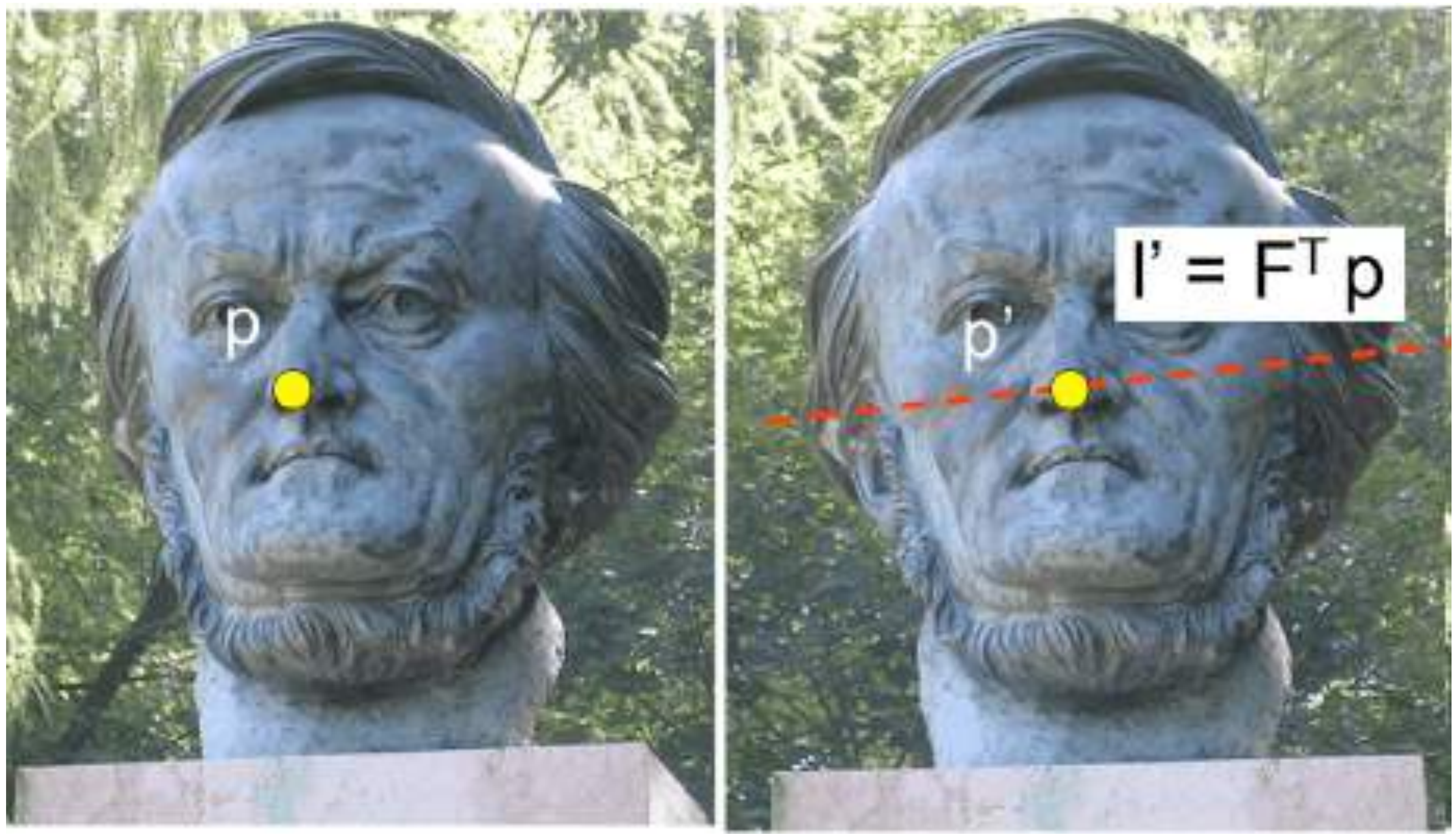
$$l_2 = F p_1$$

F is 3x3 matrix; 7 DOF

F is singular (rank 2)



Epipolar Geometry



Only need F to establish a relationship between the two corresponding points in the image
No knowledge of position of P in 3D, or intrinsic/extrinsic parameters.

8-point algorithm

$$p_1^{T} \cdot F \; p_2 = 0$$

$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

corresponding points

(u', v', 1), (u, v, 1):

$$(uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

8-point algorithm

8-point algorithm

Normalize: $q_i = Tp_i$, $q_i' = T'p_i'$

8-point algorithm to solve F_q' from → SVD

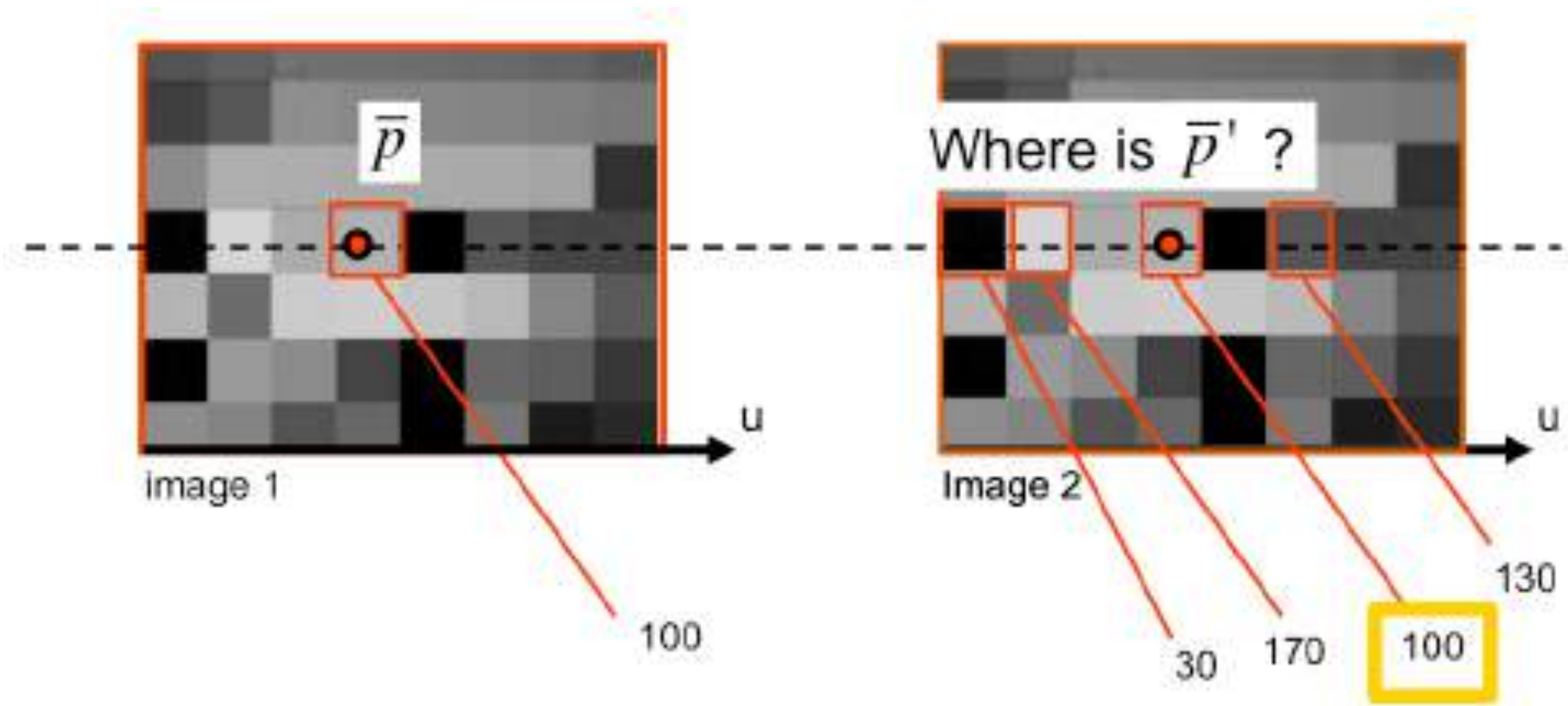
$$q_i'^T F_q' q_i = 0$$

Force F_q' to have rank 2 → SVD

De-normalize F_q to get F

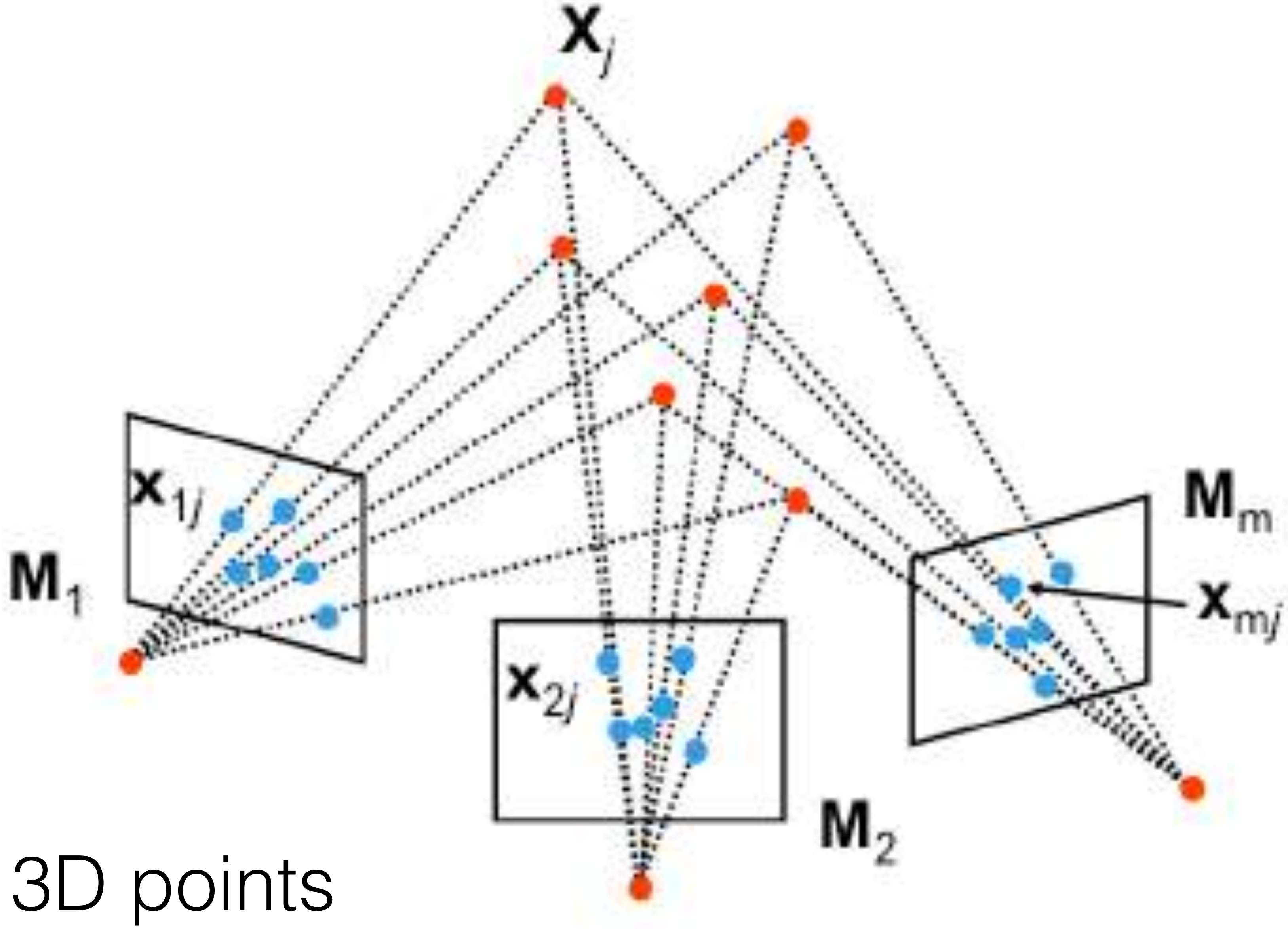
$$F = T'^T F_q T$$

Solving the Correspondence



issues:
occlusion
illumination
foreshortening
homogeneous
regions
repetitive patterns

Structure From Motion



Structure = 3D points

Motion = projection matrices

Affine Structure From Motion

Problem: estimate the m matrices Ai, m matrices bi, and the n positions Xj from the mxn observations xij.

Affine camera

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} A & b \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$x_{ij} = A_i X_j + b_i$$

$$\hat{x}_{ij} = A_i \hat{X}_j$$ normalized

Affine Structure From Motion

Factorization method

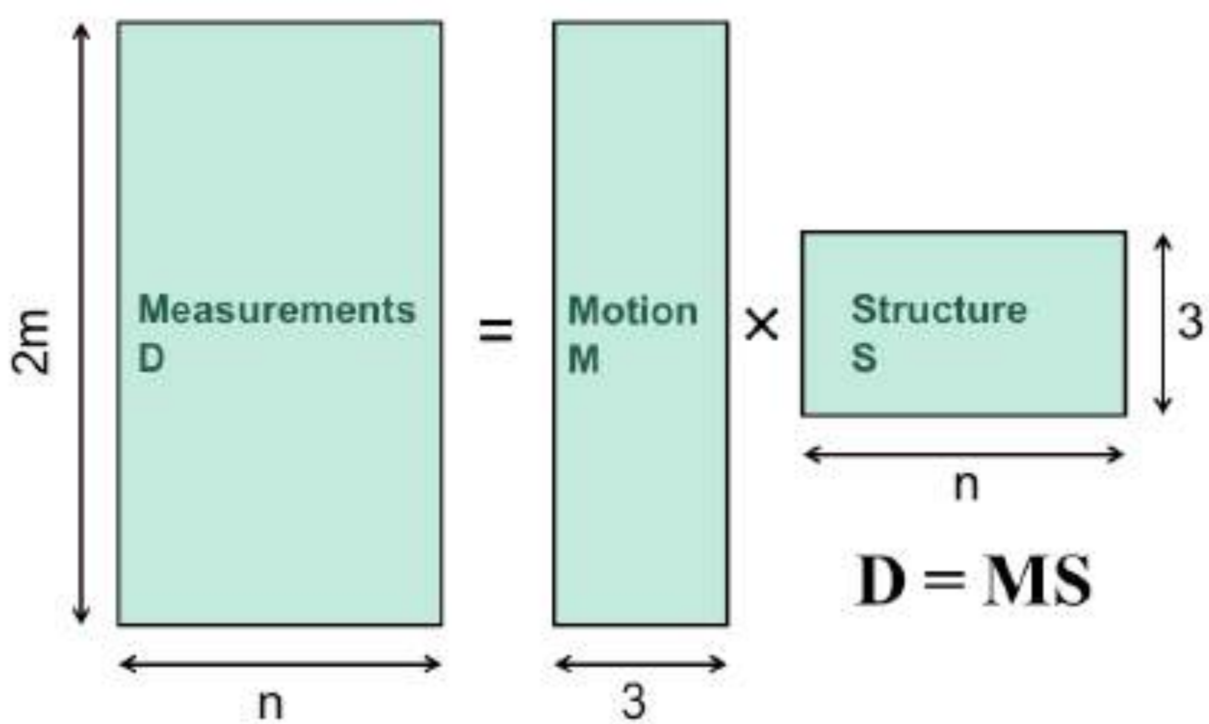
$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{x}}_{11} & \hat{\mathbf{x}}_{12} & \cdots & \hat{\mathbf{x}}_{1n} \\ \hat{\mathbf{x}}_{21} & \hat{\mathbf{x}}_{22} & \cdots & \hat{\mathbf{x}}_{2n} \\ & & \ddots & \\ \hat{\mathbf{x}}_{m1} & \hat{\mathbf{x}}_{m2} & \cdots & \hat{\mathbf{x}}_{mn} \end{bmatrix}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

($2m \times n$)

cameras ($2m \times 3$) M

points ($3 \times n$) S

[Eq. 10]



Affine camera

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

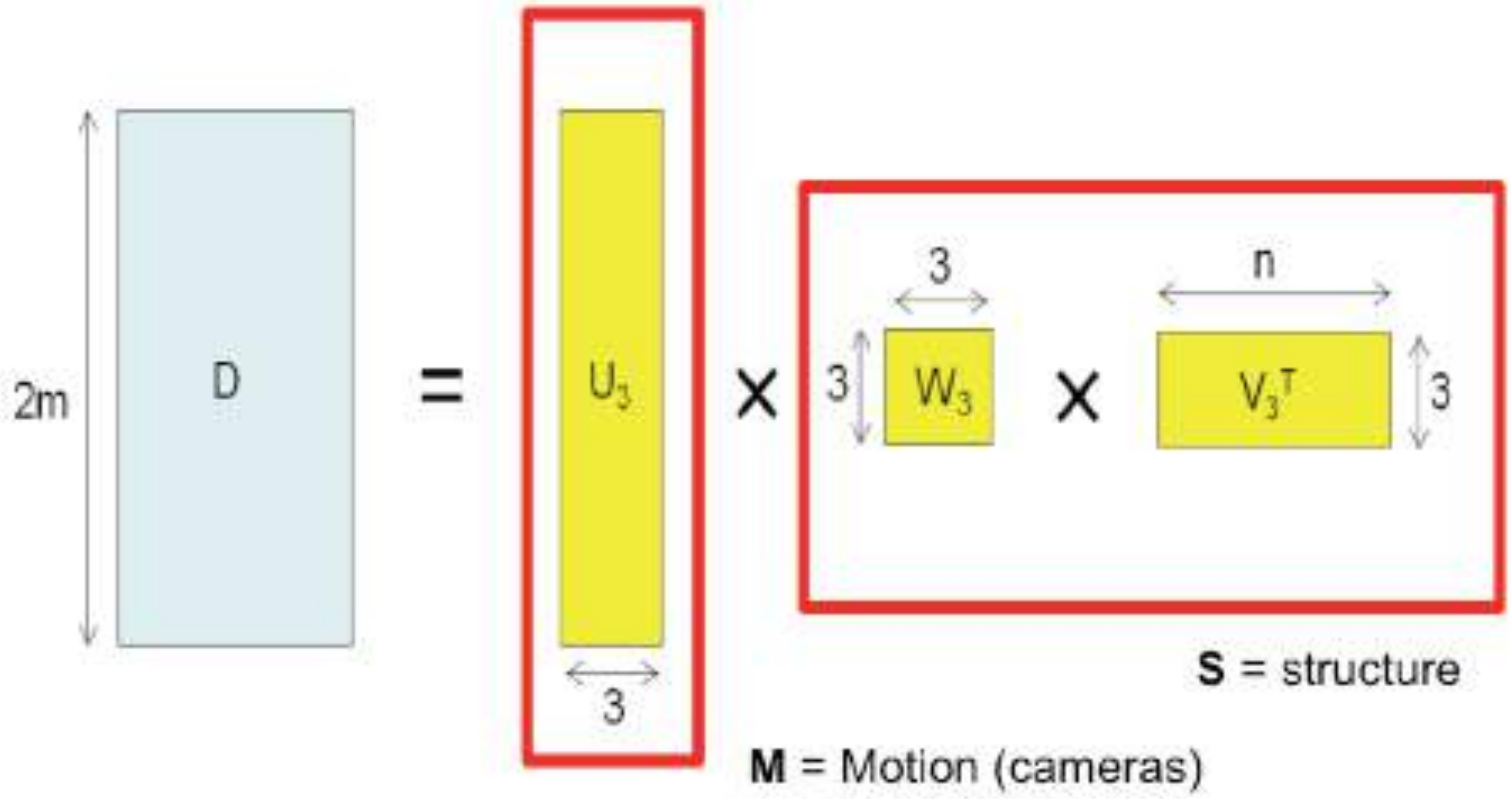
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} A & b \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$x_{ij} = A_i X_j + b_i$$

$$\hat{x}_{ij} = A_i \hat{X}_j \quad \text{normalized}$$

Factorization method

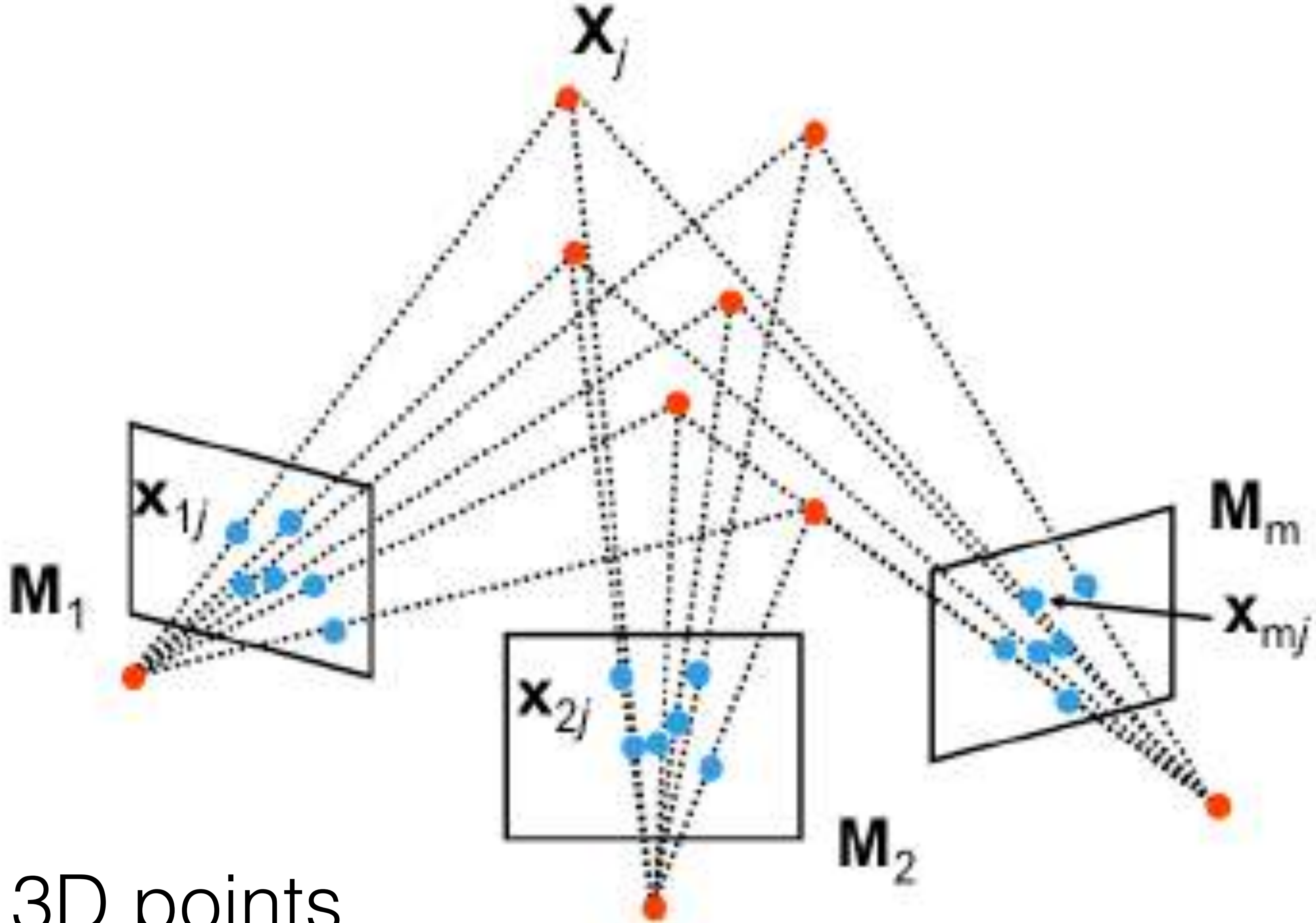
use SVD!



$$\mathbf{D} = \mathbf{U}_3\,\mathbf{W}_3\,\mathbf{V}_3^T = \mathbf{U}_3\,(\mathbf{W}_3\,\mathbf{V}_3^T) = \mathbf{M}\,\mathbf{S}$$ [Eq. 12]

D should be rank 3

Structure From Motion



Structure = 3D points

Motion = projection matrices

Structure From Motion

1. Recover structure and motion up to perspective ambiguity
    - Algebraic approach using the Fundamental matrix
    - Factorization method (by SVD)
    - Bundle adjustment

2. Resolving the perspective ambiguity

Structure = 3D points

Motion = projection matrices

Structure From Motion

Limitations:

1. Algebraic approach using the Fundamental matrix
   - Yields pairwise solutions.
2. Factorization method (by SVD)
   - Assumes that all points are visible in every image, doesn't handle occlusions.

Bundle adjustment via optimization

- Attempts to minimize the re-projection error (pixel distance between the projection of a reconstructed point into the estimated cameras for all the cameras and all the points).

Active Stereo and Volumetric Stereo

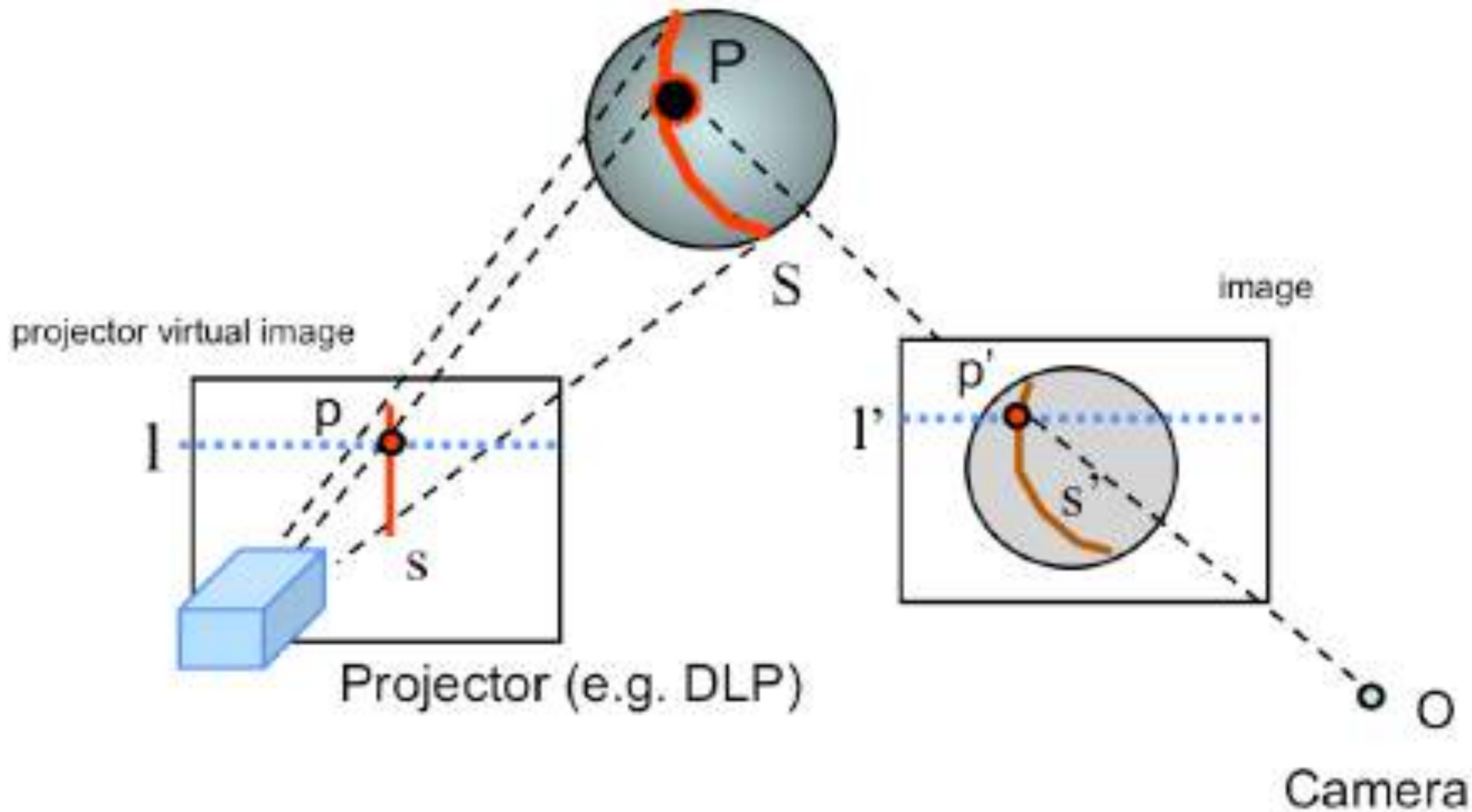
Active Stereo

- Structured lighting
- Depth Sensing

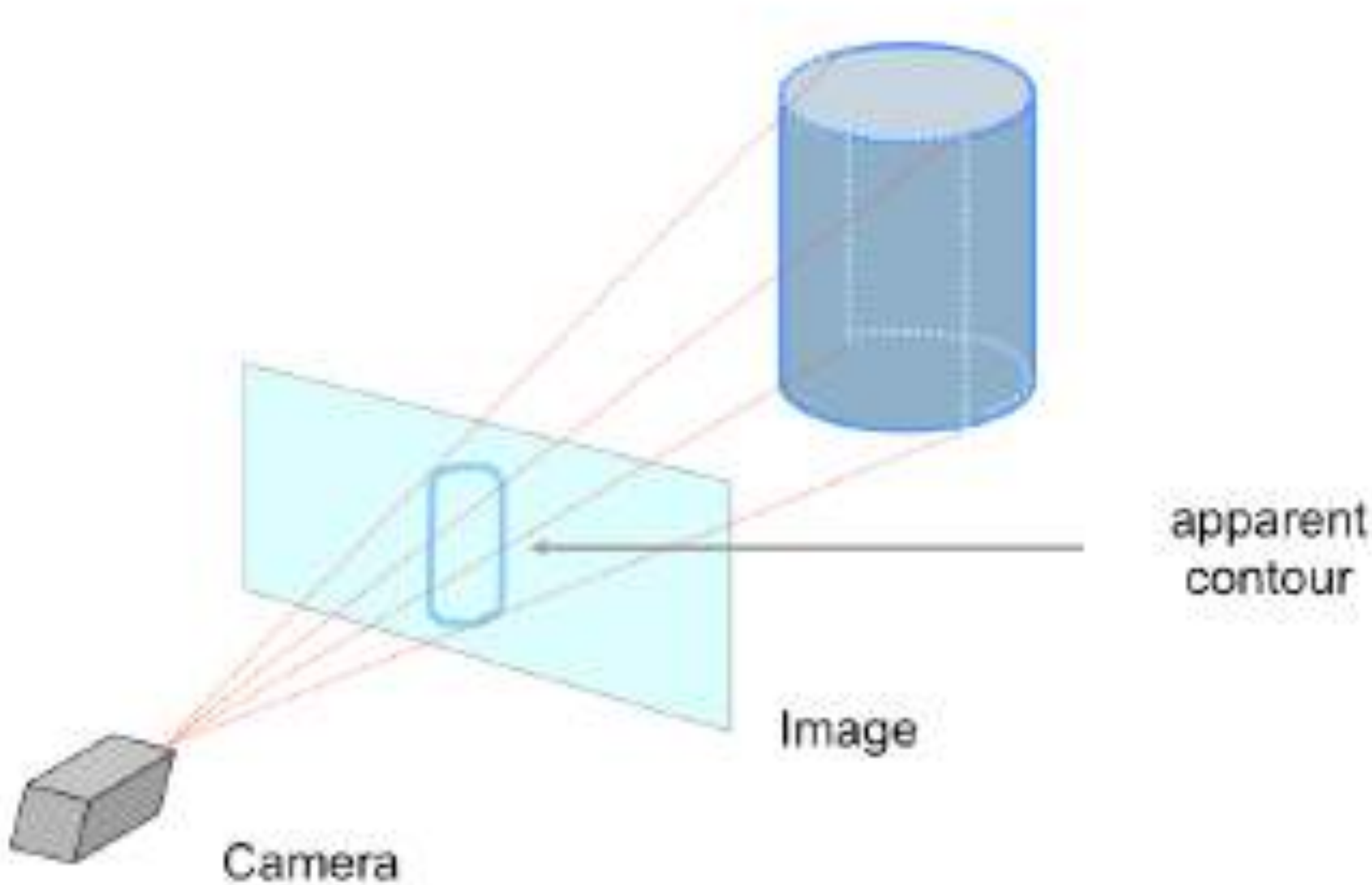
Volumetric Stereo

- Space carving
- Shadow carving
- Voxel coloring

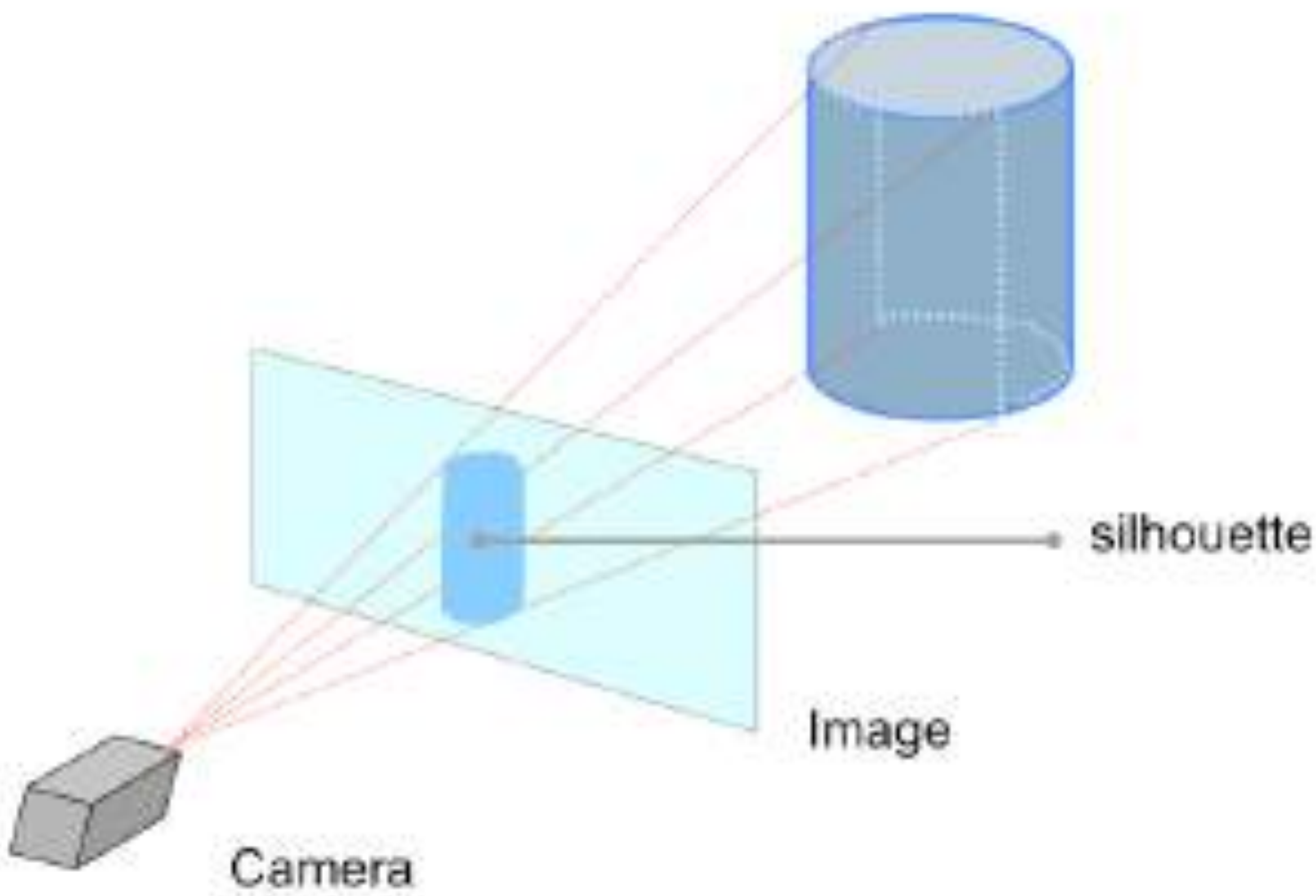
Active Stereo



Volumetric Stereo



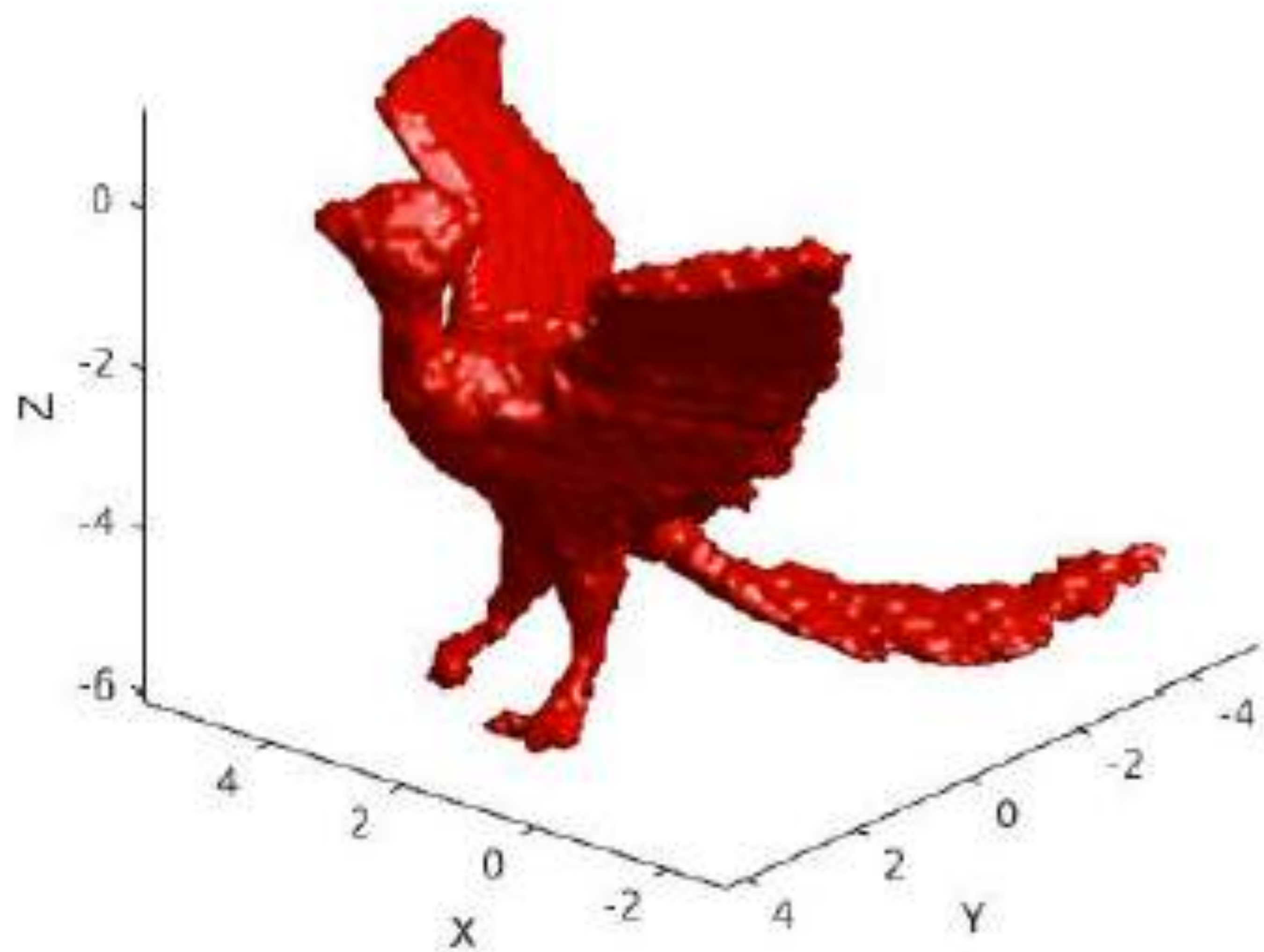
Volumetric Stereo



Volumetric Stereo

Space Carving

- remove voxels via multiple views.



Fitting and Matching

- Least Squares methods
- RANSAC
- Hough transform

Fitting

Goal: Find the best model parameters to fit the data.

Things we might want to fit include

- Lines
- Curves
- Homographic transformations
- Fundamental matrices

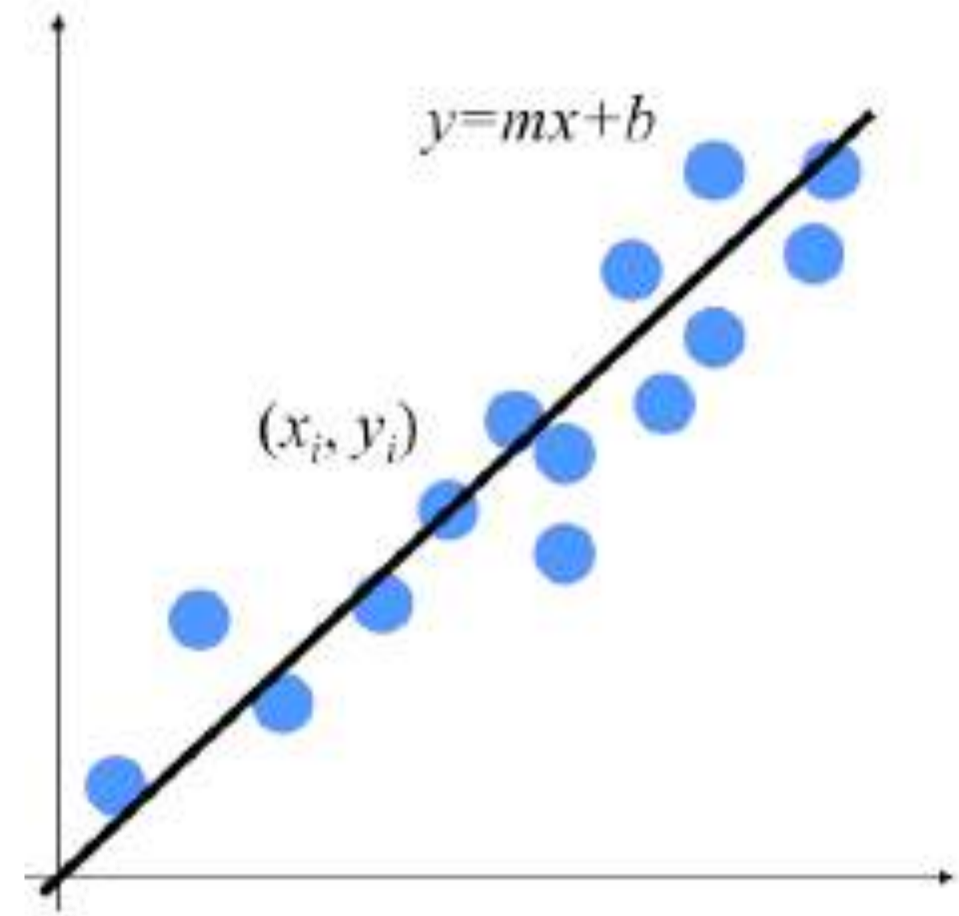
Issues with Fitting

- Noisy data
- Outliers
- Missing data

Least squares methods

$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

Goal is the find the line that minimizes the residuals.

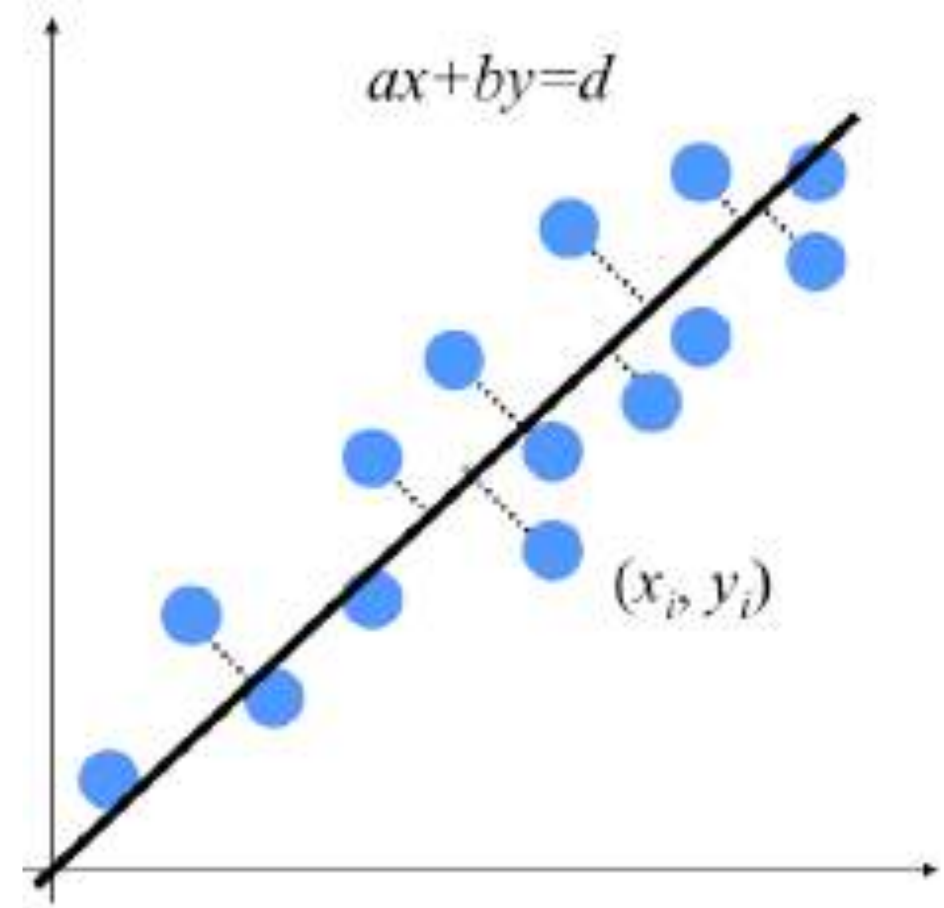


Least squares methods

$$E = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

$$Ah = 0 \qquad h = \begin{bmatrix} a \\ b \\ d \end{bmatrix}$$

data parameters



Robustness estimator to handle outliers

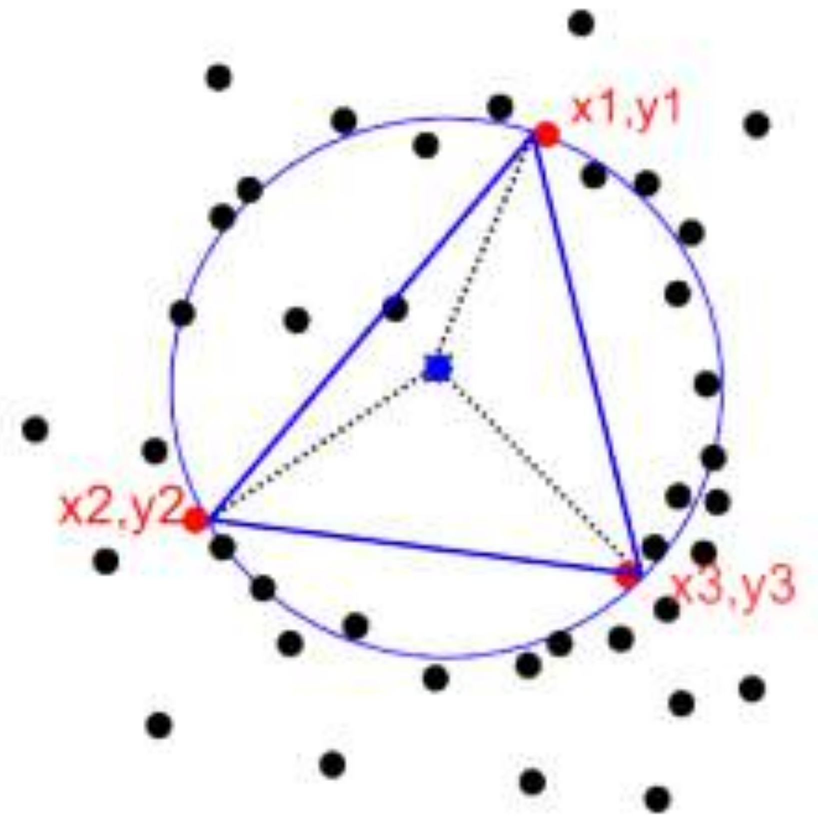
RANSAC

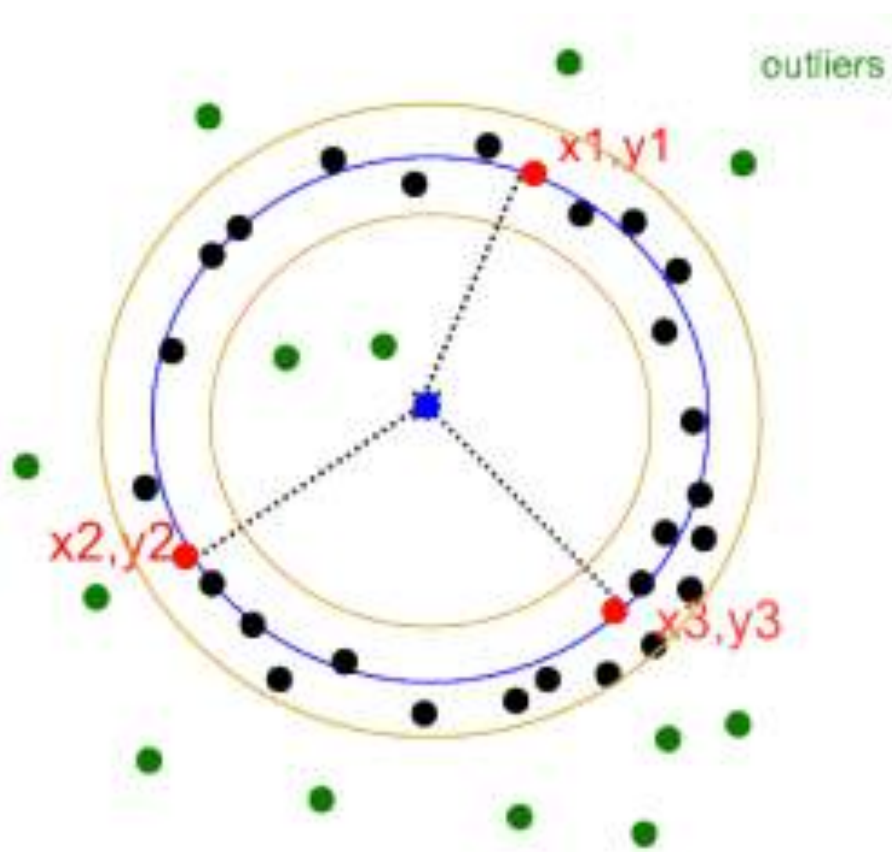
Randomized iterative method to fit a parametric model to data based on random sampling of data.

The typical approach

1. Sample the minimum number of points required to fit the model.
2. Fit a model to the sample.
3. Compute inliers.
4. Refine model based on all points.

RANSAC: fitting a circle





1. Select 3 random points. Find center and radius of the circumscribed circle to the triangle formed by the 3 points.

2. Compute distances of all the points to the center of the circle, and calculate inliers and outliers

3. Repeat at most N times

RANSAC

Randomized iterative method to fit a parametric model to data based on random sampling of data.

Pros:

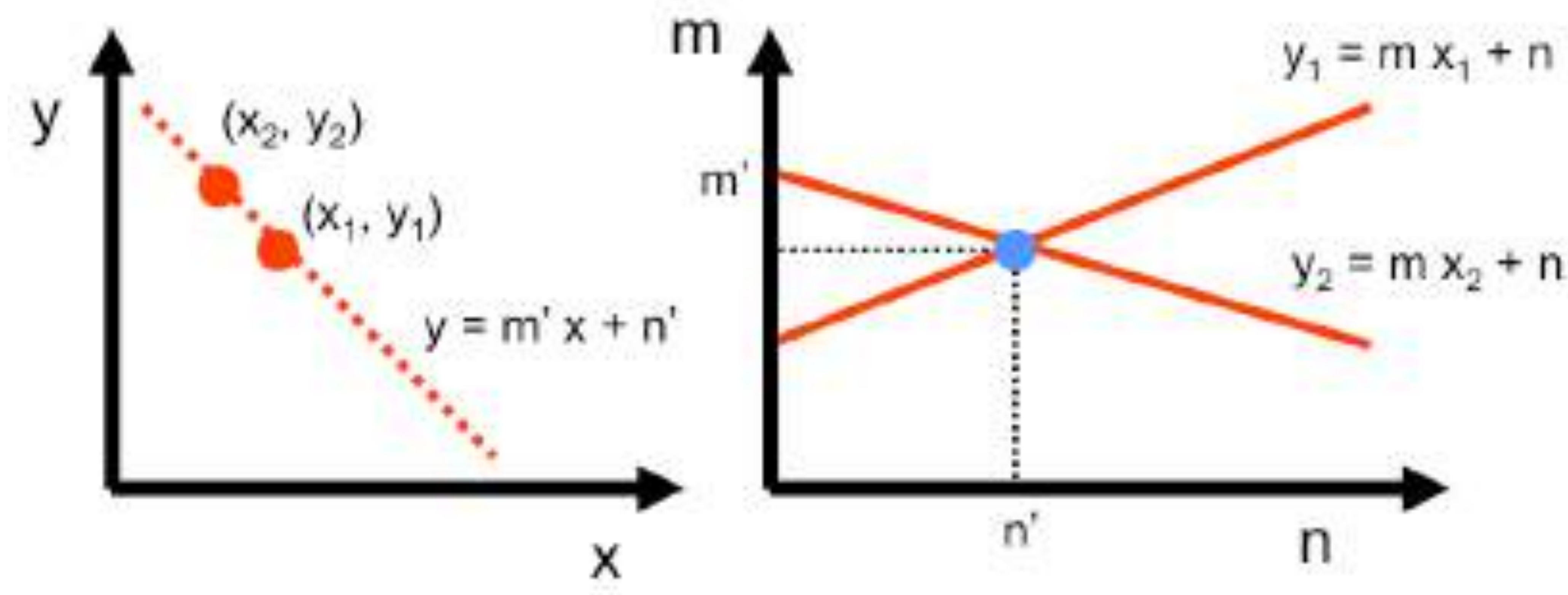
- General method suited for a wide range of model fitting problems
- Easy to implement

Cons:

- Only handles a moderate percentage of outliers
- lots of parameters to tune

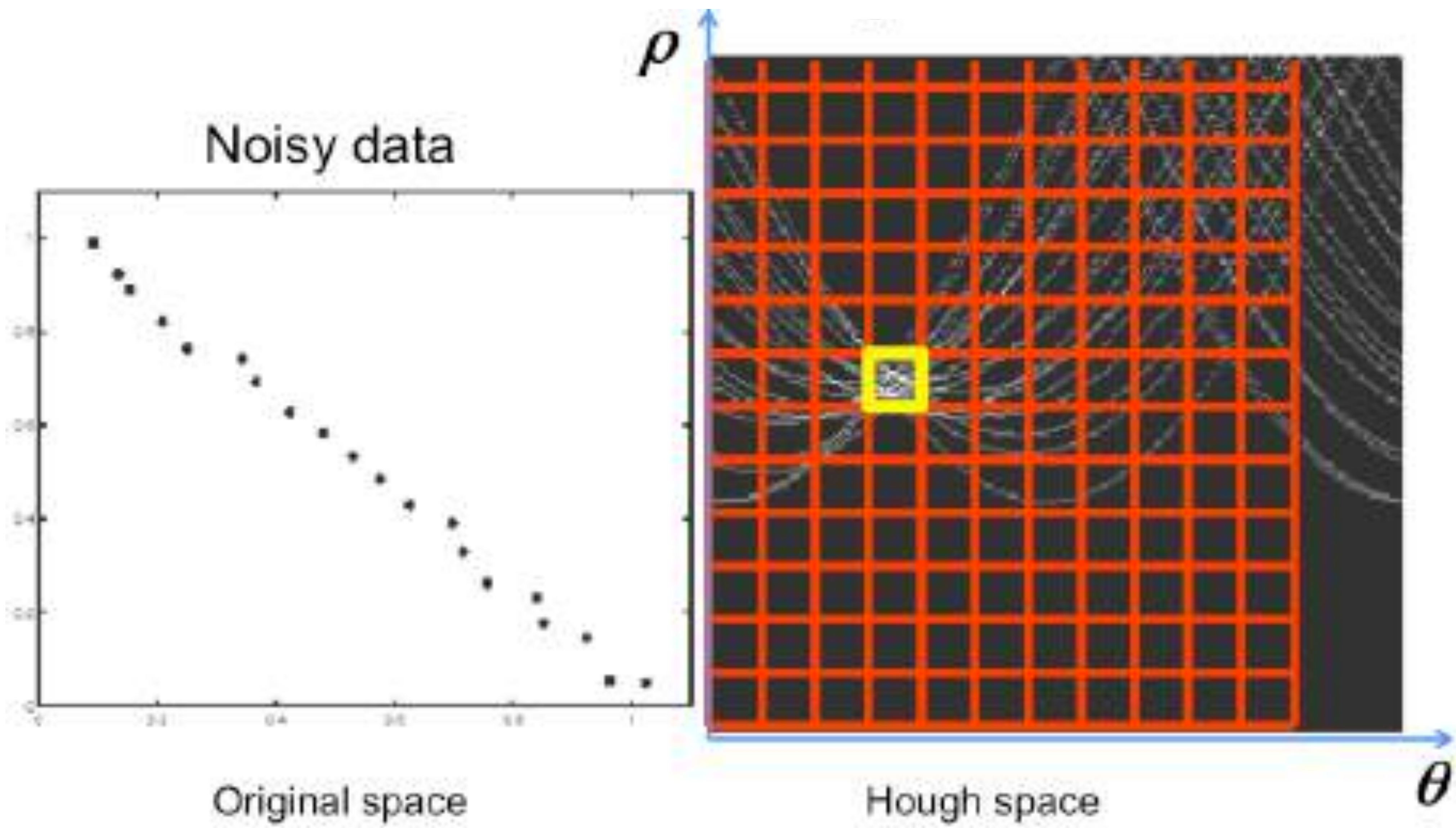
Hough Transform

A voting scheme to fit a parametric model to data by selecting the model with the most votes.



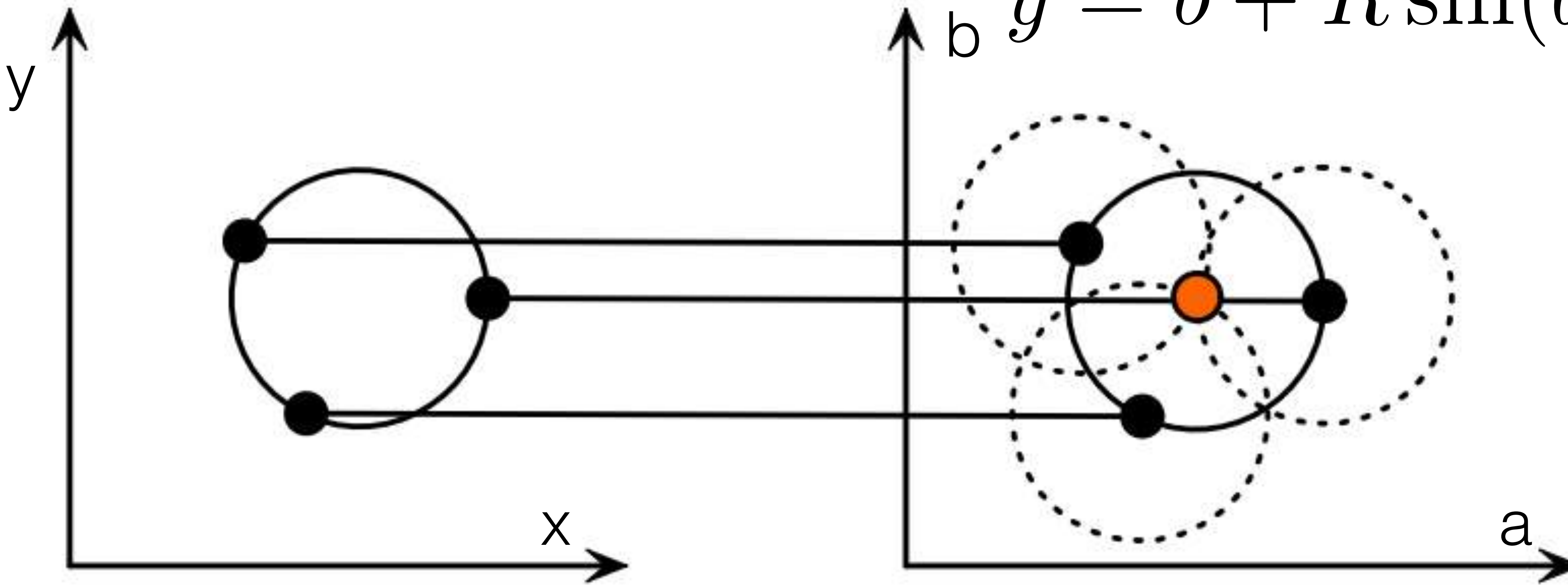
Hough Transform

A voting scheme to fit a parametric model to data by selecting the model with the most votes.



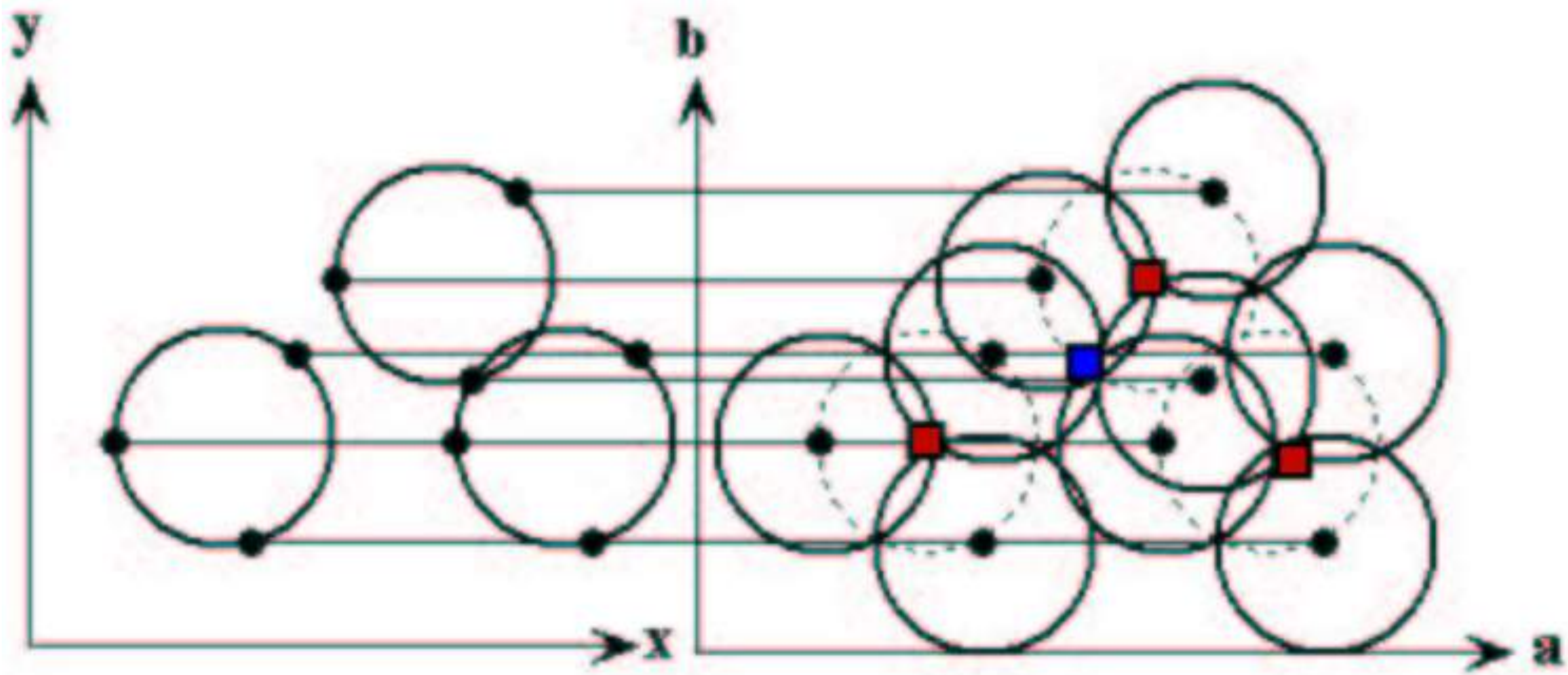
Hough Transform to fit a Circle with known radius R

$$x = a + R\cos(\theta)$$
$$y = b + R\sin(\theta)$$



Each point in the geometric space (left) generates a circle in parametric space (right). The circles in parametric space intersect at the (a,b) that is the center in the geometric space.

Hough Transform to fit a Multiple Circles with known R



Each point in the geometric space (left) generates a circle in parametric space (right). The circles in parametric space intersect at the (a,b) that is the center in the geometric space.

Hough transform

Pros:

- All points are processed independently, so can handle occlusions
- Some robustness to noise: noisy points unlikely to contribute consistently to any single bin.
- Can detect multipole instances of a model in a single pass

Cons:

- Complexity of search time increases exponentially with the number of model parameters
- Spurious peaks due to uniform noise
- Quantization: hard to pick a good grid size.

Detectors and Descriptors

Properties of detectors
- Edge detectors (Canny edge detector)
- Harris Corner detector
- Blob detectors
- Difference of Gaussians (DoG)

Properties of descriptors
- SIFT
- HOG

Good Luck!