CS231A Midterm Review

Overview

- General Logistics
- Hough Transform and RANSAC examples and overviews
- Single View Metrology: Vanishing lines, points, etc.
- Camera Matrix and geometry
- Fundamental matrix and Affine SFM

Midterm Logistics

- We will post the midterm on the course website at ~12:15pm Tuesday 2/25.
- Midterms must be submitted by Thursday 11am, either in class or in the Gates dropbox.
- SCPD can submit through the SCPD office.
- Late submissions will get a 0%
- Piazza will be frozen(no new posts), clarifications can be requested via the staff mailing list.
- Open book, open notes, no collaboration.

Midterm Structure

- 1-2 Implementation questions
- 3-4 Theory questions
- Coverage is up to and including PS3 material.
- 6 hour estimated time.

RANSAC + Hough Transform

RANSAC: A randomized iterative method to fit a parametric model to data based on a random sampling of data

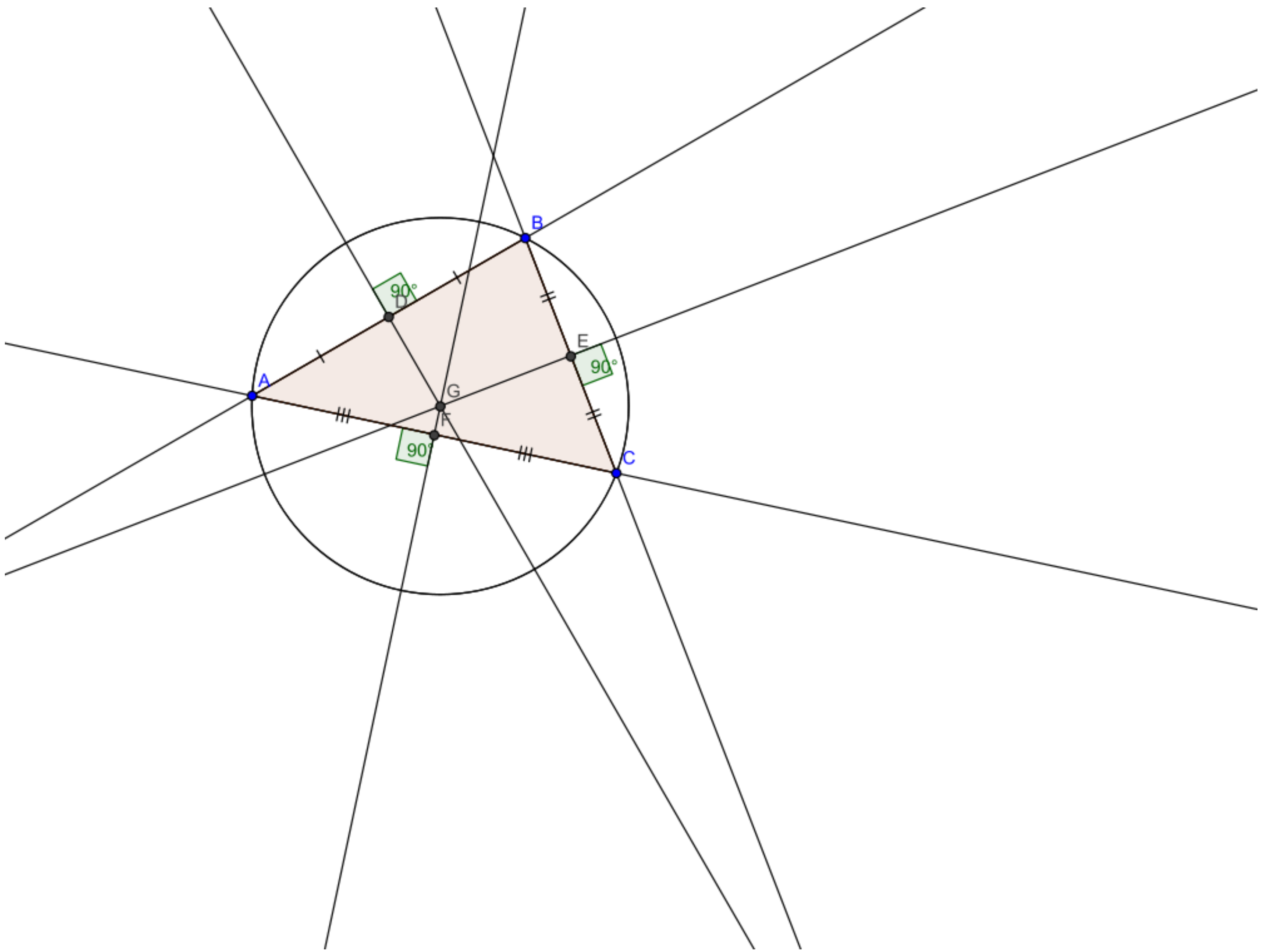
Hough Transform: A voting scheme to fit a parametric model to data by selecting the model with the most votes.

Example Problem

Finding a circle using RANSAC:

1. Decide how many points that we need
2. Fit a model to the data
3. Compute inliers
4. Refine model based on all points

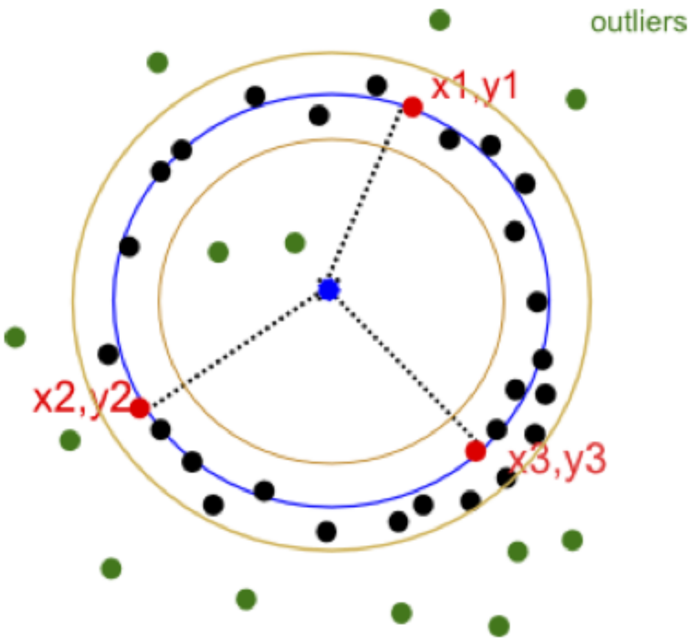
Finding the circle from 3 points



Distance to circle?

$$d = \left| \sqrt{(x_c - x)^2 + (y_c - y)^2} - r \right|$$

Outliers?



Figures: Brehar Raluca

RANSAC Pros/Cons

<u>Pros</u>:

•General method suited for a wide range of model fitting problems

•Easy to implement and easy to calculate its failure rate

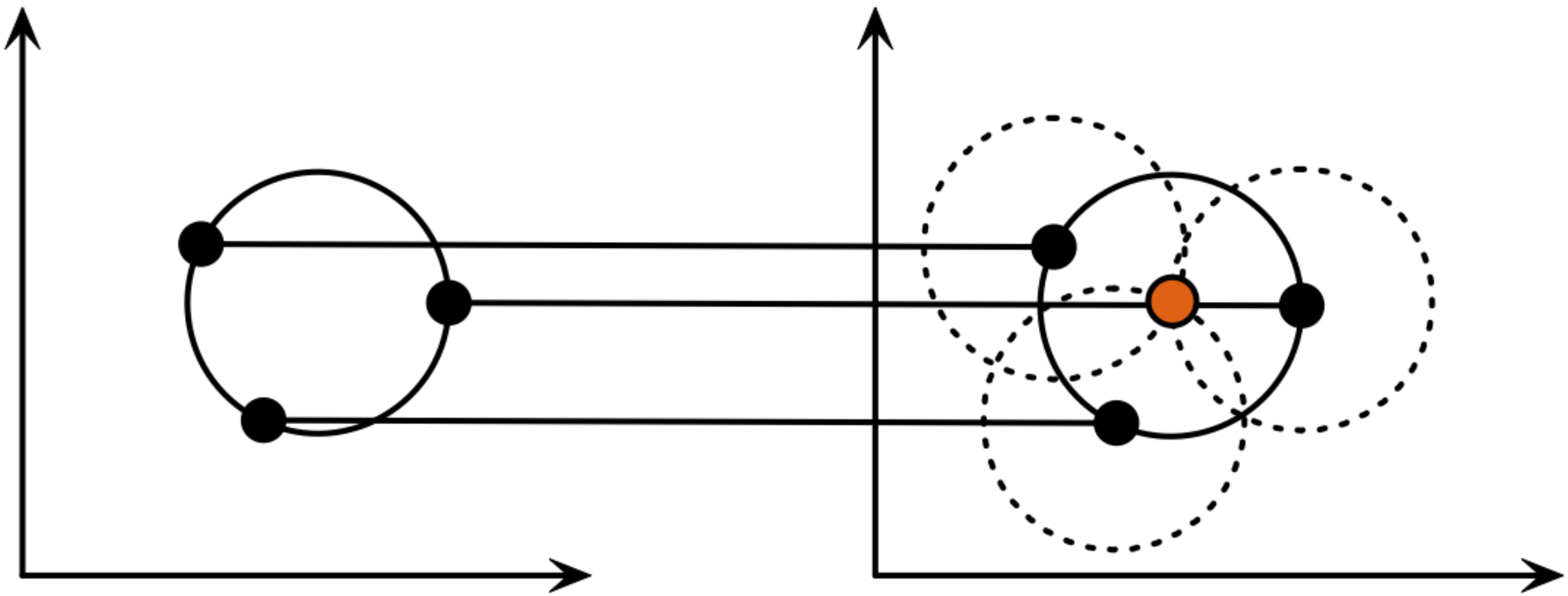
<u>Cons</u>:

•Only handles a moderate percentage of outliers without cost blowing up

•Many real problems have high rate of outliers (but sometimes selective choice of random subsets can help)

Hough Transform to find a circle

We can start with a simplified scenario where the radius is known

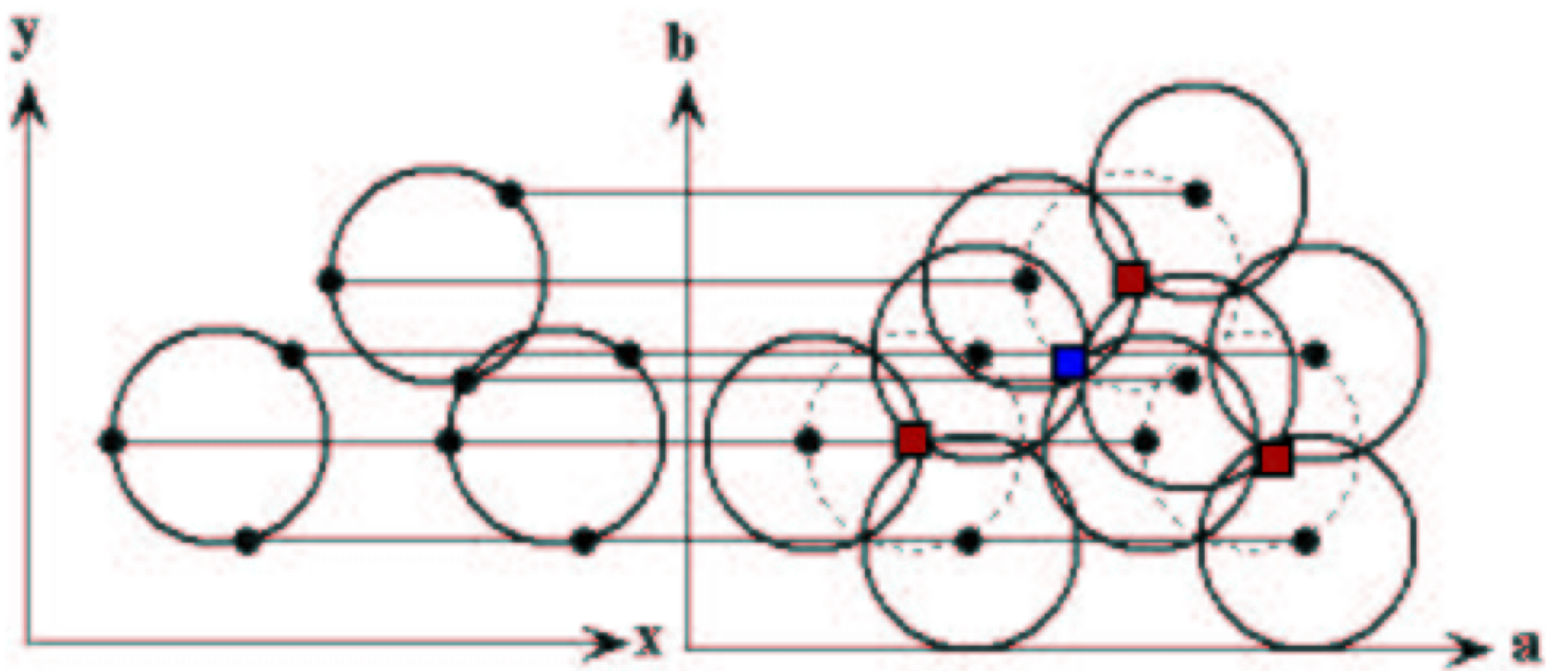


Figures: Dr. Harvey Rhody, RIT

Hough Transform: Circles

Each point maps to a circle in the parameter space spanned by candidate circle centers.

What if there are multiple circles?

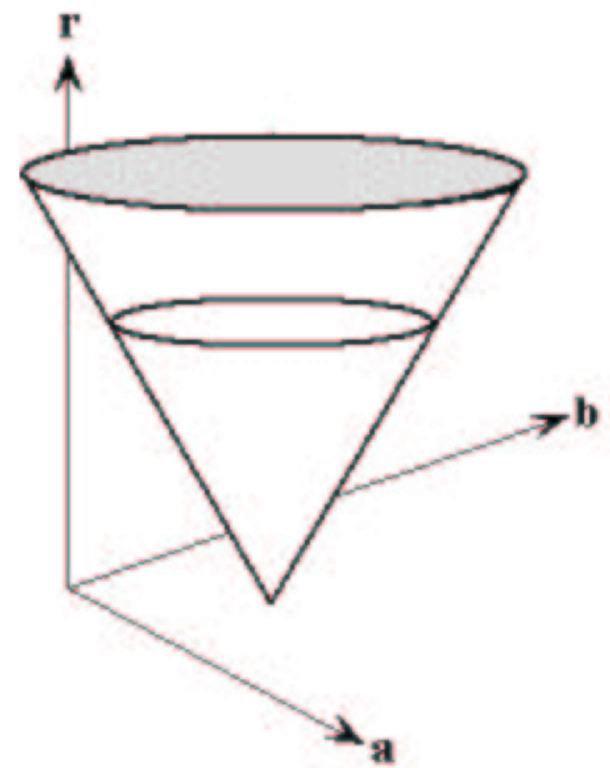


Hough Transform: Circles

We can then vote on the most likely circle(s)

What if we don't know the radius?

- We have a 3d parameter space, each point maps to a hollow cone.



Hough Transform Pros/Cons

Pros

- All points are processed independently, so can cope with occlusion
- Some robustness to noise: noise points unlikely to contribute consistently to any single bin
- Can detect multiple instances of a model in a single pass

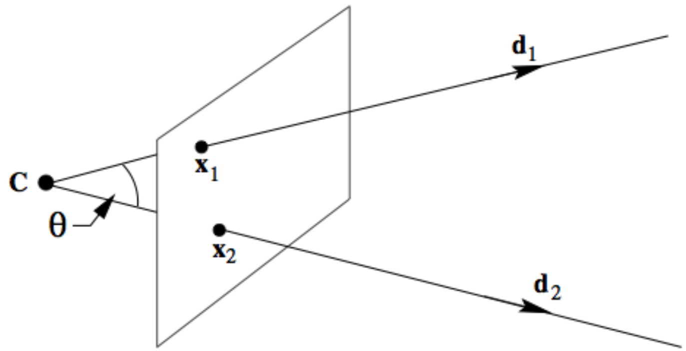
Cons

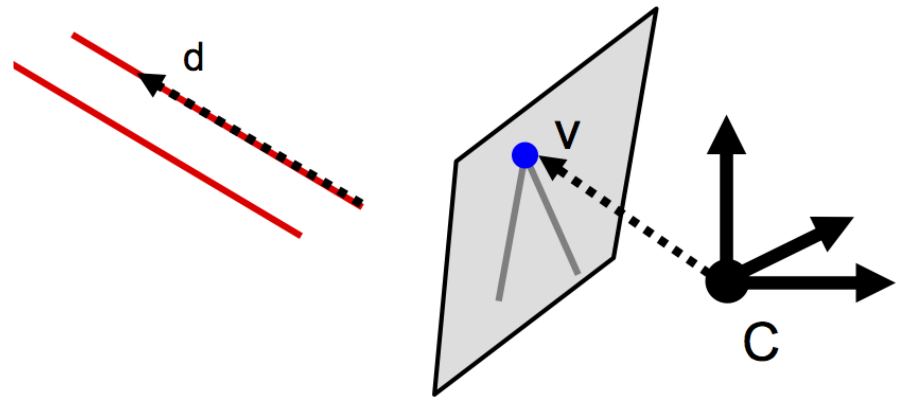
- Complexity of search time increases exponentially with the number of model parameters
- Non-target shapes can produce spurious peaks in parameter space
- Quantization: hard to pick a good grid size

Single View Metrology

- Vanishing points
- Vanishing lines
  - construction of lines from points
  - directions and normals of vanishing points/planes

Vanishing Points



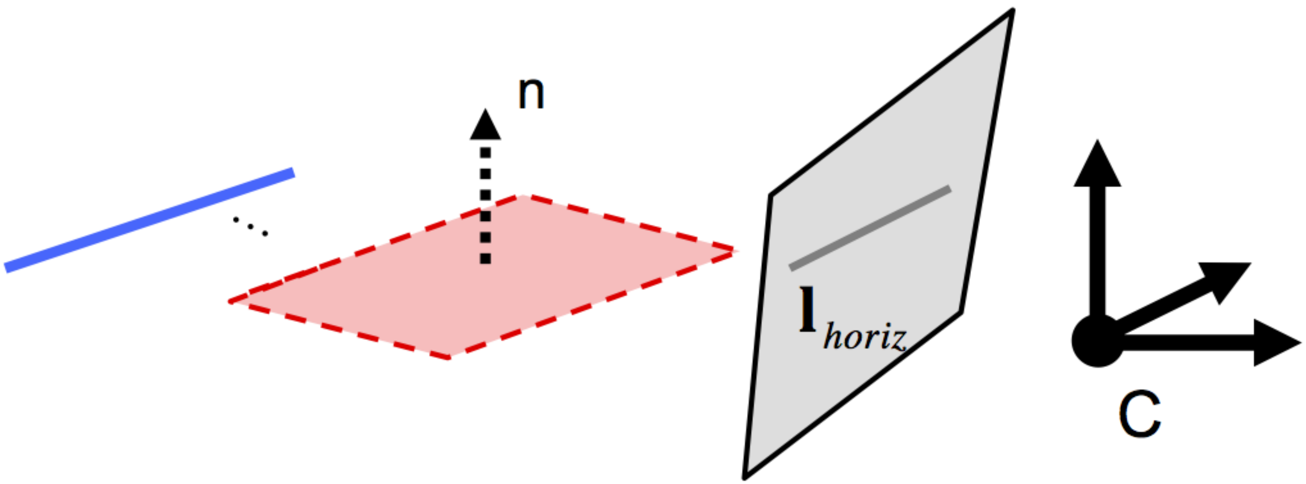


$$\mathbf{d}_1 = \frac{K^{-1}\,\mathbf{v}_1}{\left\| K^{-1}\,\mathbf{v}_1 \right\|}$$

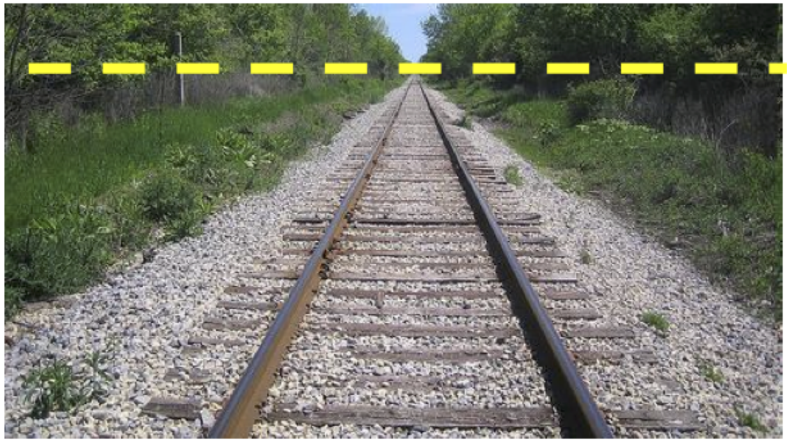
$$\mathbf{v} = K\,\mathbf{d}$$

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

Vanishing Lines

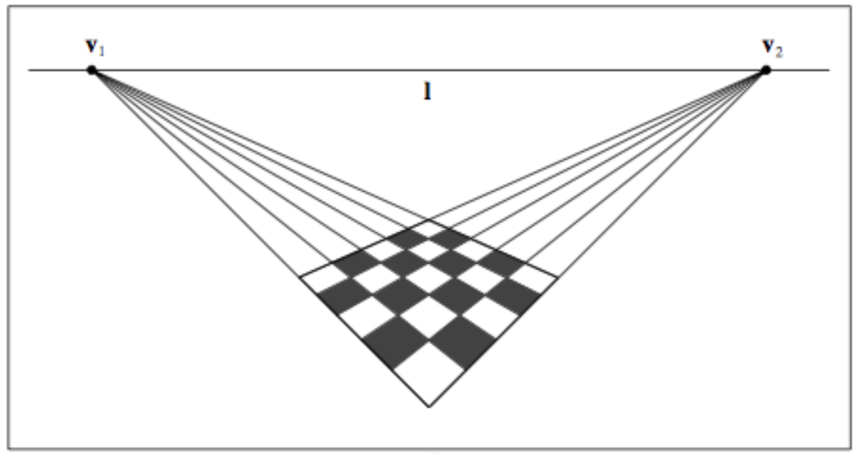


$$\mathbf{n} = K^T \mathbf{l}_{\text{horiz}}$$

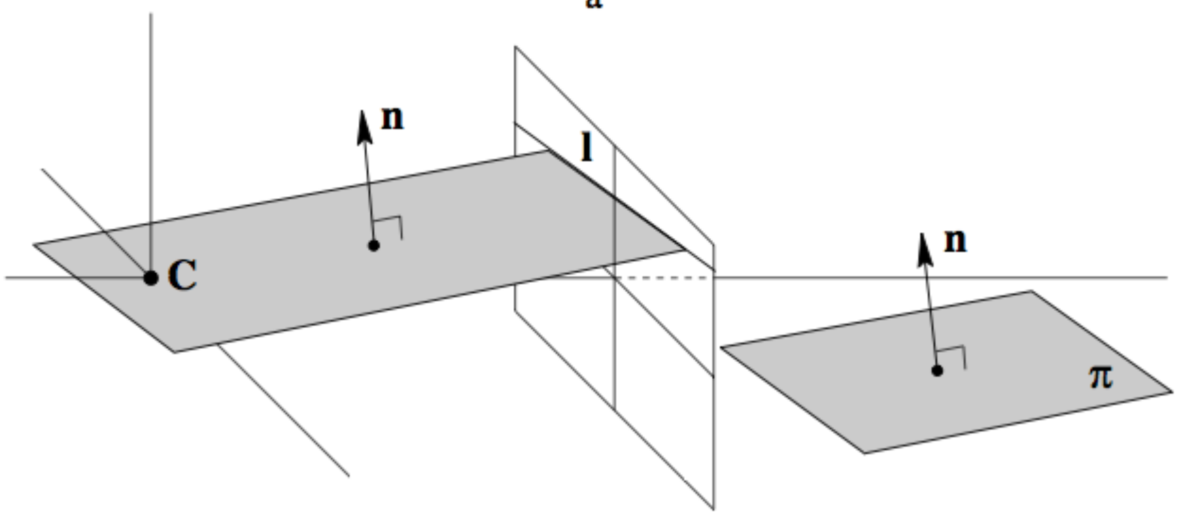


Vanishing Lines

$$l = v_1 \times v_2$$



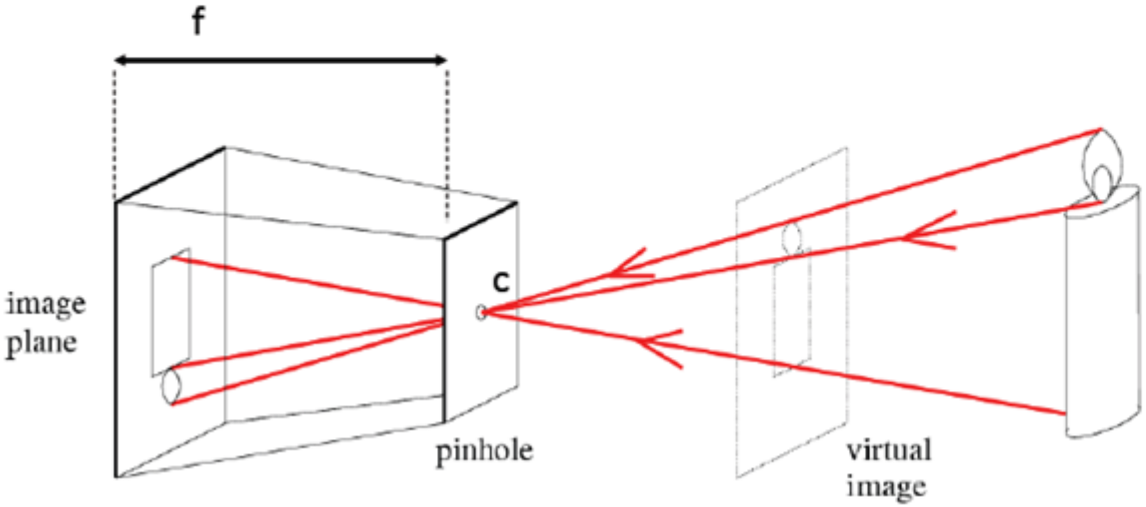
a



b

Projection

From pinhole camera



$$(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$$

This is not linear

Homogeneous Coordinates

Cartesian → Homogeneous

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

camera

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

scene

Homogeneous → Cartesian

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

camera

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

scene

Projective Camera

intrinsic matrix extrinsic matrix

$$P'_{3\times1} = M \; P_w = K_{3\times3}\begin{bmatrix} R & T \end{bmatrix}_{3\times4} P_{w4\times1}$$

camera matrix

$$K = \begin{bmatrix} \alpha & s & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

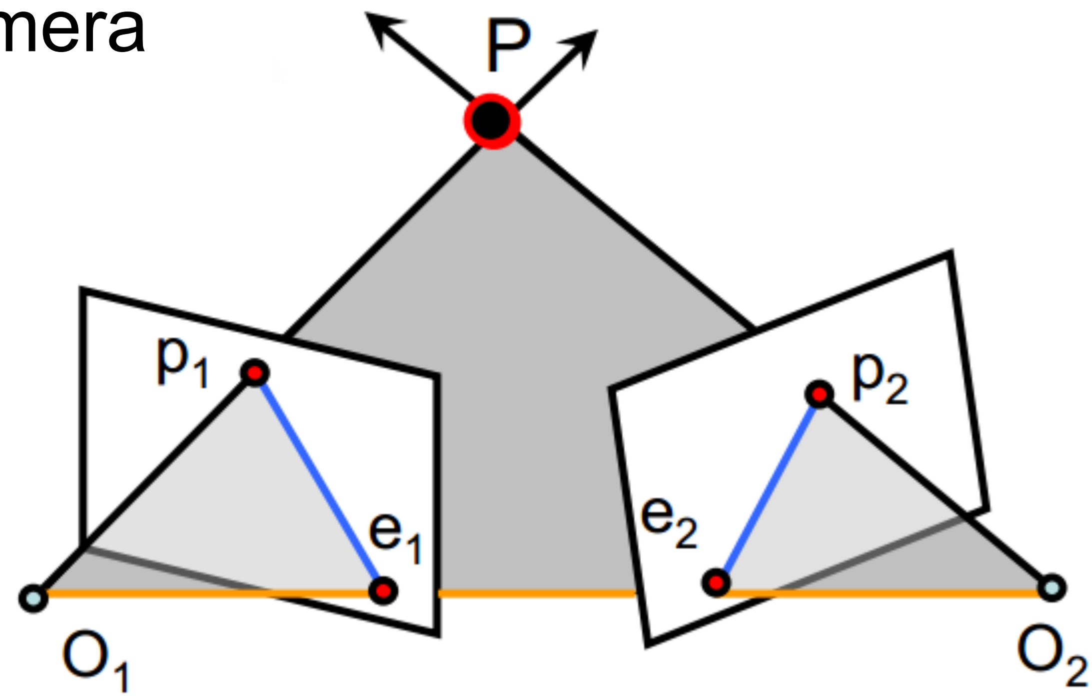
Epipolar geometry

P: object

O: center of camera

p: image point

e: epipole

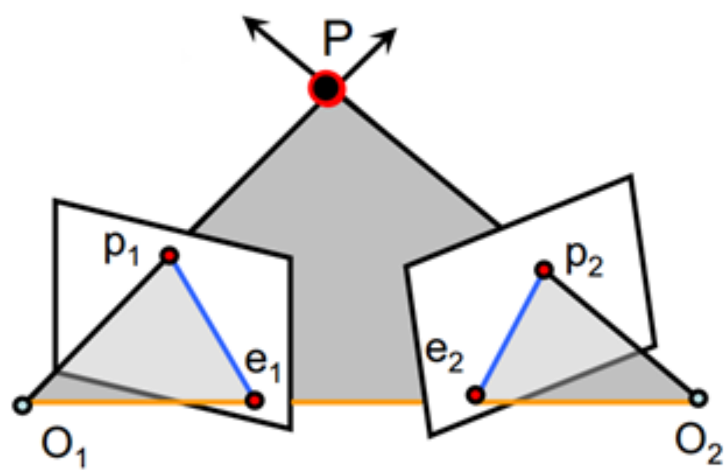


Fundamental Matrix F

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

F is rank 2, with 7 degree of freedom

Epipolar lines: $l_1 = F\,p_2$



Computation of F

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

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

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

Normalized 8-point Algorithm

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

8-point algorithm to solve F_q' from → SVD

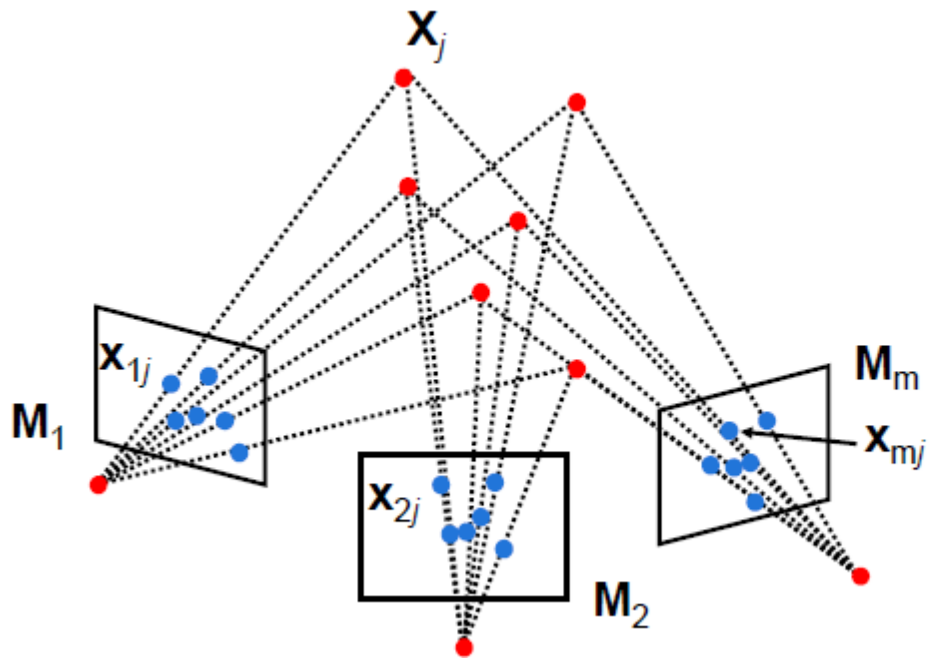
$$q_i'^T F_q' q_i = 0$$

Force F_q' to have rank 2 → SVD

De-normalize F_q to get F

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

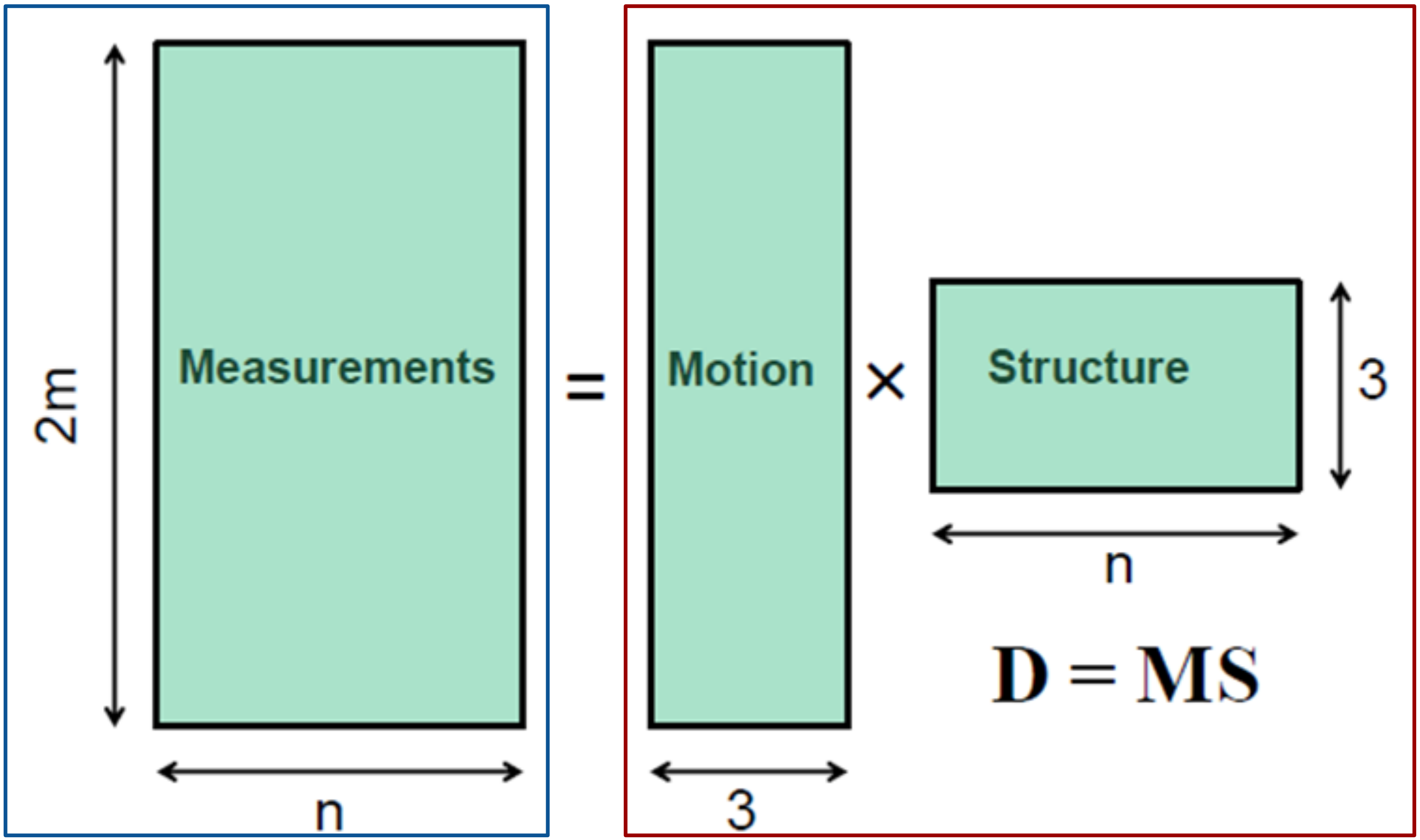
Structure From Motion



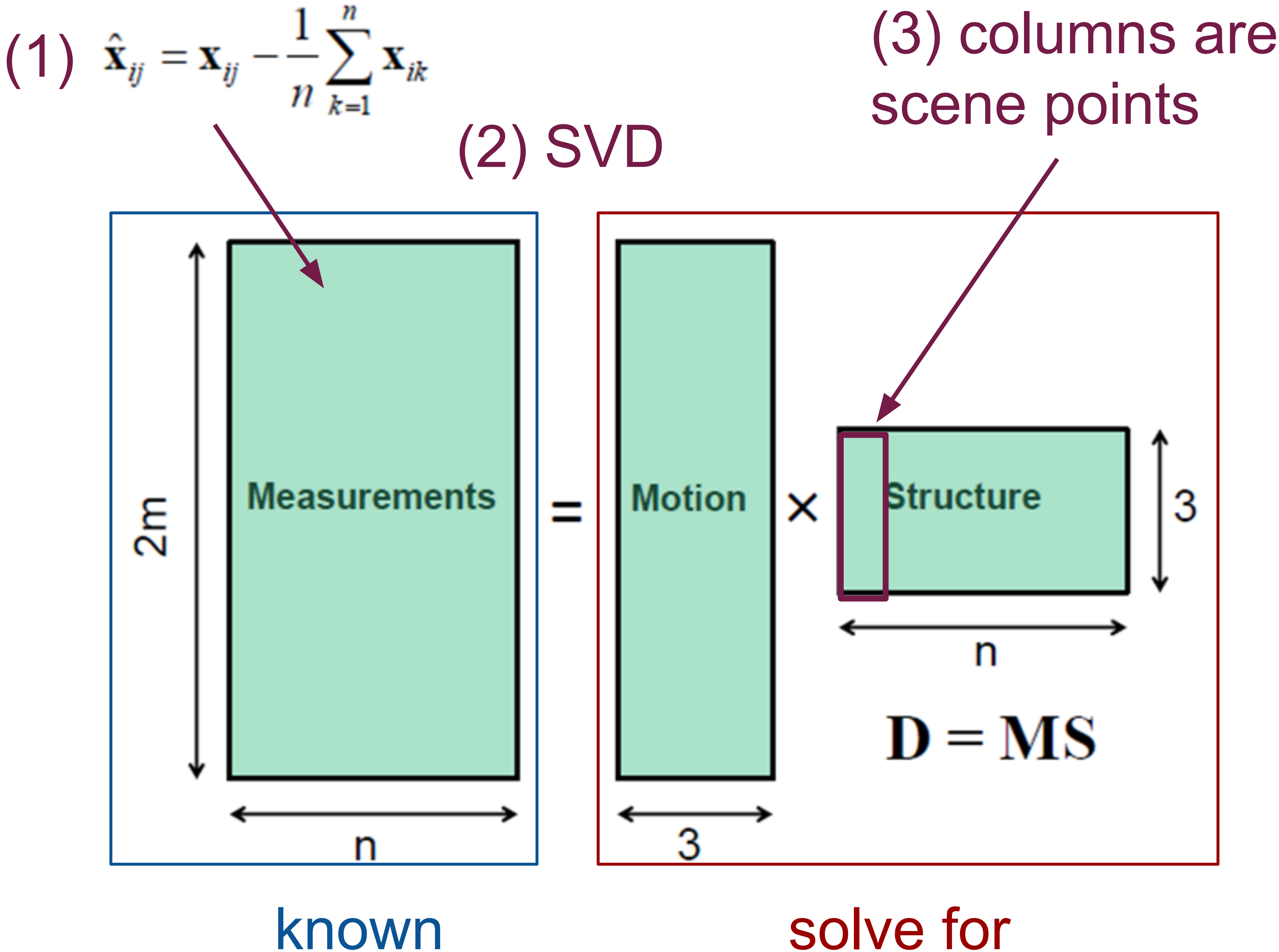
$$\underline{\mathbf{x}_{ij}} = \underline{\mathbf{M}_i \mathbf{X}_j}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

known   solve for

Factorization



known solve for


(1) $\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij} - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik}$
(2) SVD
(3) columns are scene points
2m
Measurements
n
=
Motion
3
×
Structure
3
n
$\mathbf{D} = \mathbf{MS}$
known
solve for