Model Fitting

Computer Vision I

CSE 252A

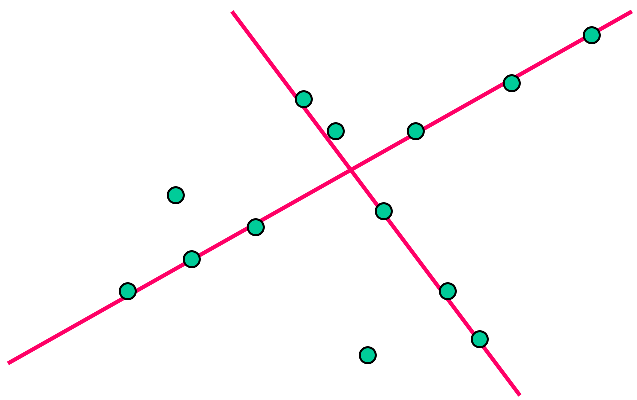
Lecture 11

Announcements

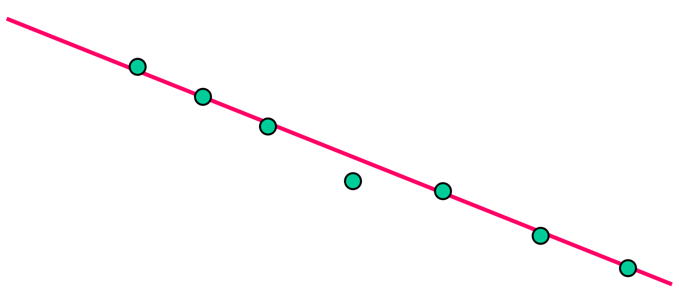
- Assignment 2 is due Nov 8, 11:59 PM
- Assignment 3 will be released Nov 8
  - Due Nov 22, 11:59 PM

Model fitting example

- Segment linked edge chains into curve features (e.g., line segments)
- Group unlinked or unrelated edges into lines (or curves in general)

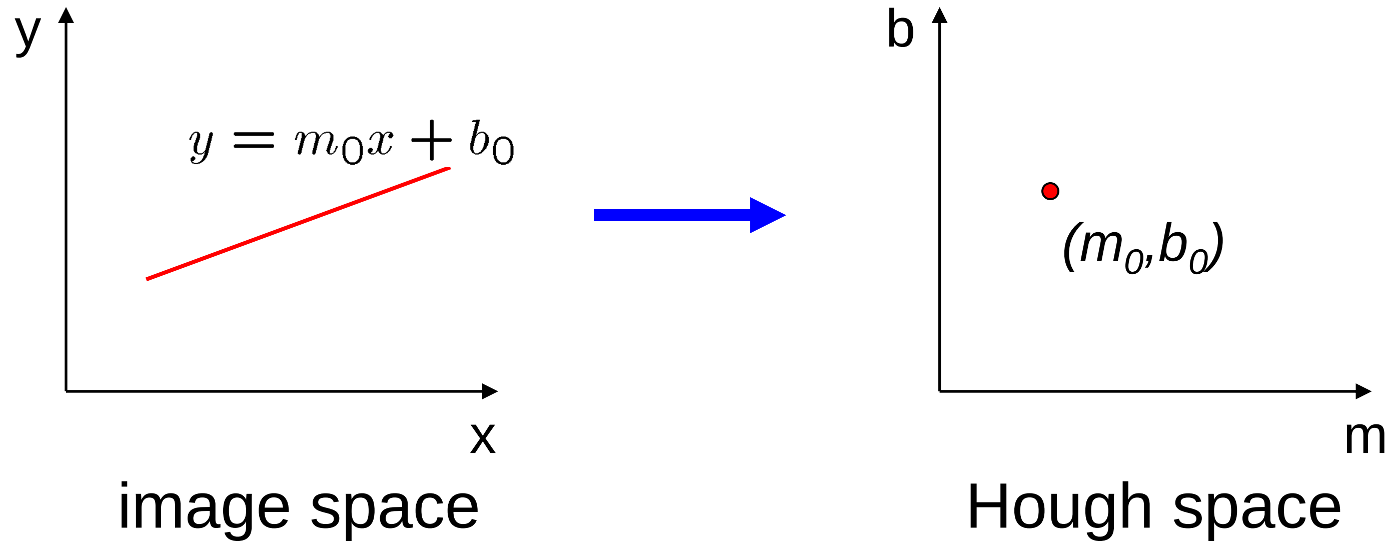


- Accurately fitting parametric curves (e.g., lines) to grouped edge points



Hough Transform
[Patented 1962]

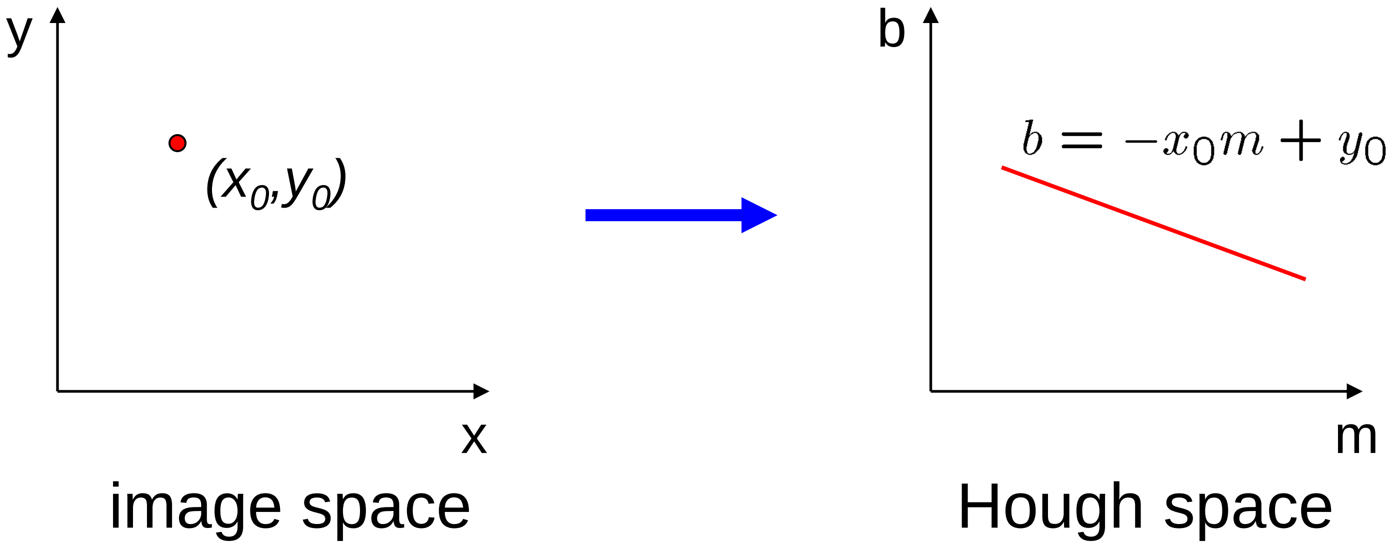
Finding lines in an image



Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space

Finding lines in an image



Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space
- What does a point (x_0, y_0) in the image space map to?

The equation of any line passing through (x_0, y_0) has form

$$y_0 = mx_0 + b$$

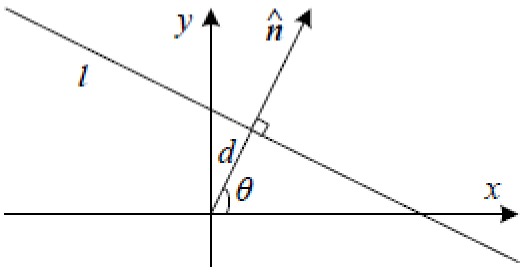
This is a line in Hough space: $b = -x_0 m + y_0$

Hough Transform Algorithm

- Typically use a different parameterization

$$d = x cos\theta + y sin\theta$$

  – d is the perpendicular distance from the line to the origin
  – θ is the angle this perpendicular makes with the x axis



- Basic Hough transform algorithm
  1. Initialize H[d, θ]=0 ; H is called accumulator array
  2. for each edge point I[x,y] in the image

     for θ = 0 to 180

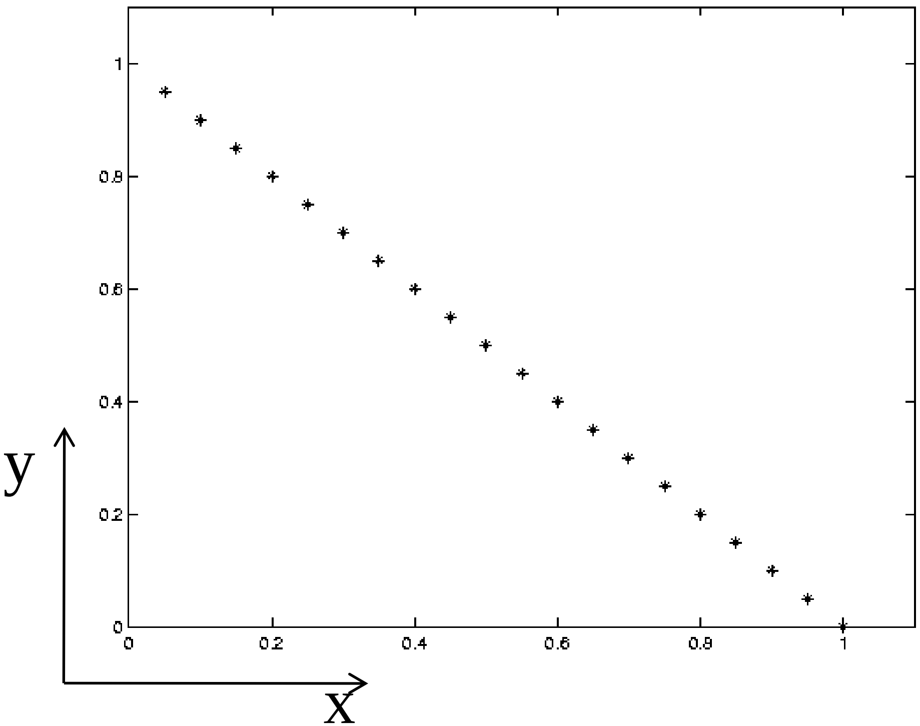
     $d = x cos\theta + y sin\theta$

     H[d, θ] += 1
  3. Find the value(s) of (d, θ) where H[d, θ] is the global maximum
  4. The detected line in the image is given by $d = x cos\theta + y sin\theta$

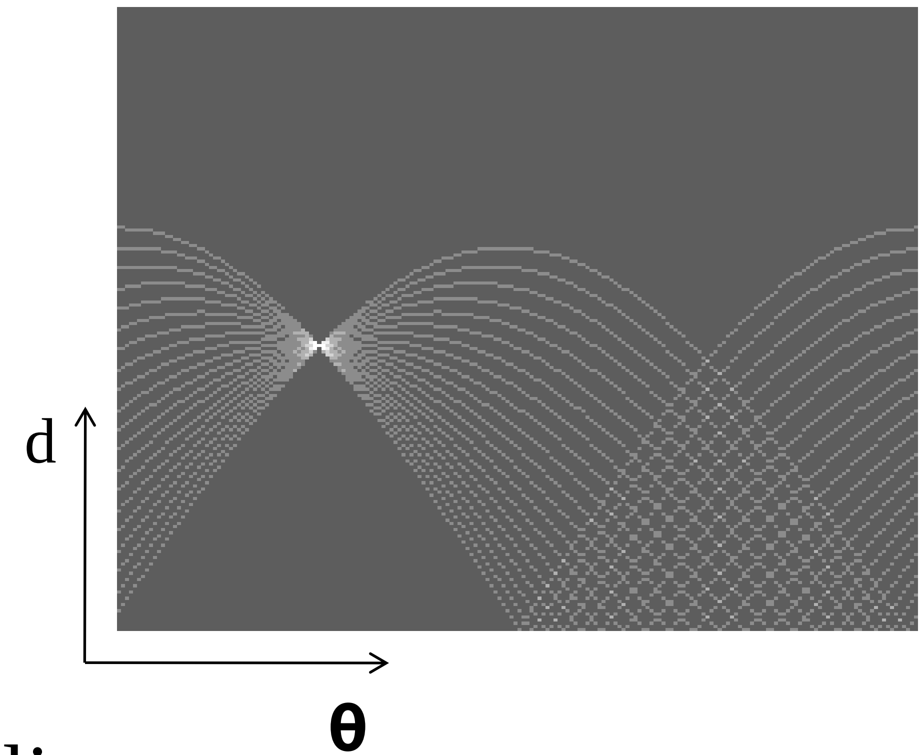
- What's the running time (measured in # votes)?

Hough Transform: 20 colinear points

Image



Accumulator

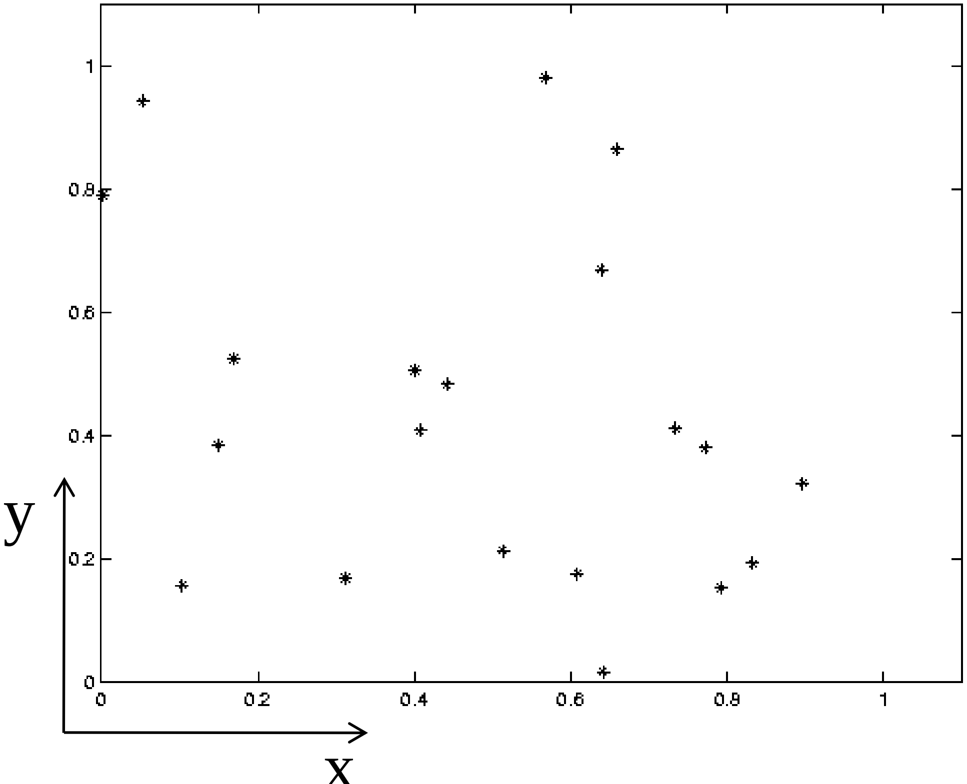


- d, θ representation of line
- Drawn as: $d = |x\cos\theta + y\sin\theta|$
- Maximum accumulator value is 20

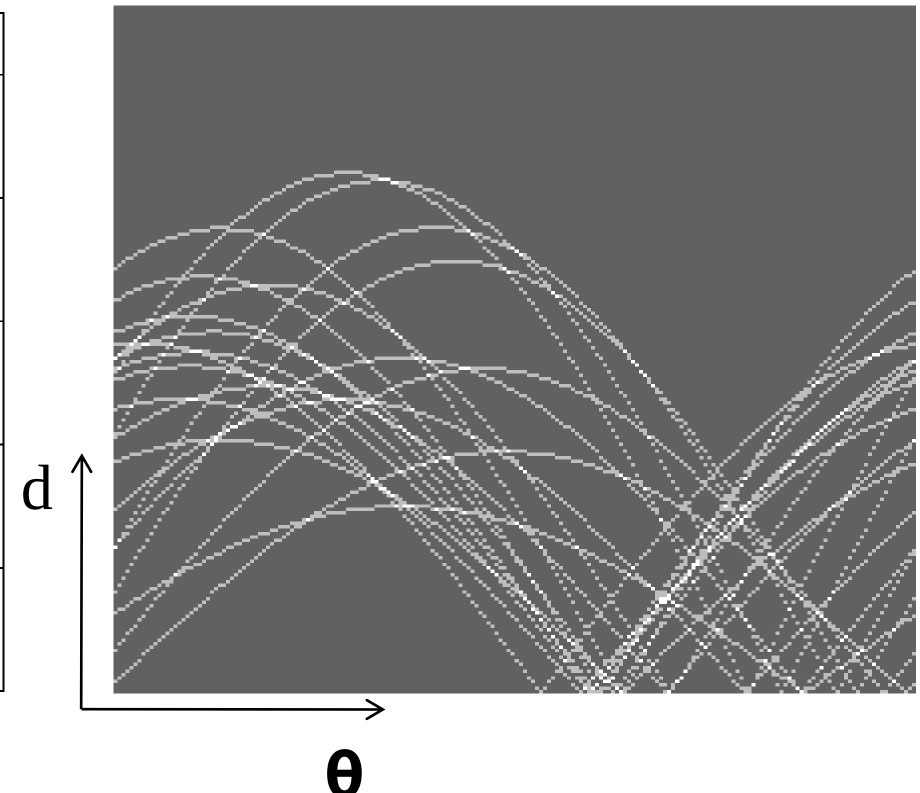
Note: accumulator array range: theta: 0-360, d: positive

Hough Transform: Random points

Image



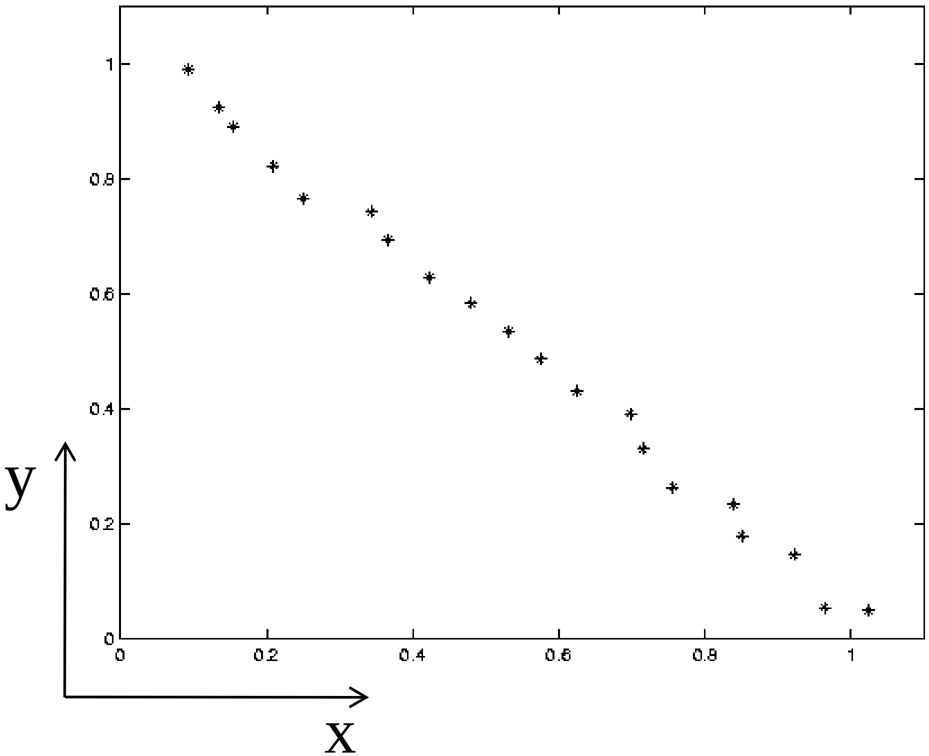
Accumulator



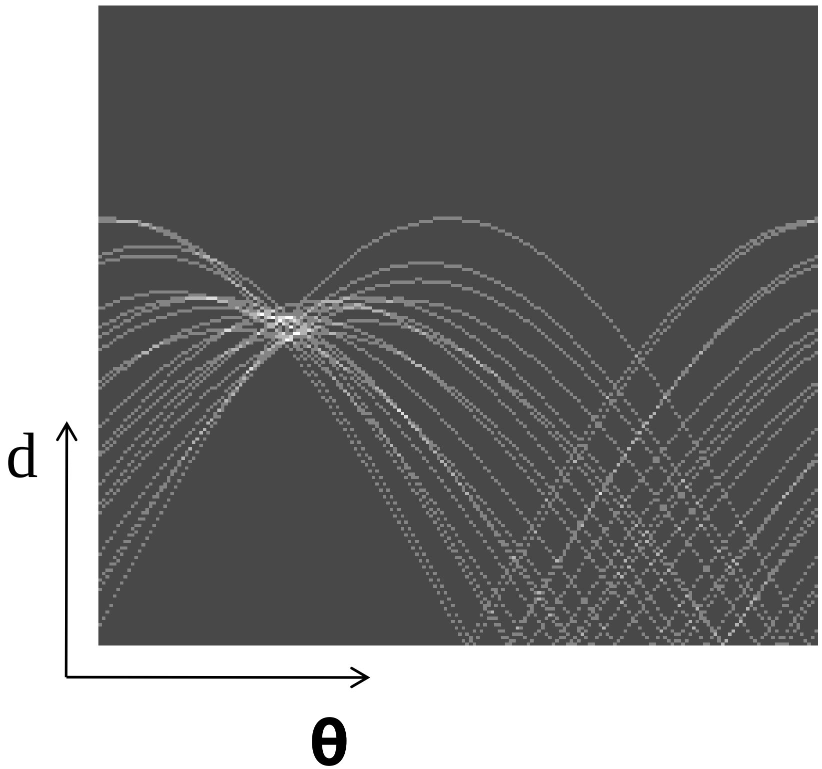
- d, θ representation of line
- Drawn as: $d = |x\cos\theta + y\sin\theta|$
- Maximum accumulator value is 4

Hough Transform: "Noisy line"

Image



Accumulator



- d, θ representation of line
- Drawn as: $d = |x\cos\theta + y\sin\theta|$
- Maximum accumulator value is 6

Hough Transform for Curves

Generalized Hough Transform

- The Hough transform can be generalized to detect any curve that can be expressed in parametric form:

$$y = f(x, a_1, a_2, \ldots a_p)$$

or

$$g(x, y, a_1, a_2, \ldots a_p) = 0$$

  – $a_1, a_2, \ldots a_p$ are the parameters
  – The parameter space is p-dimensional
  – The accumulating array is *large*

Example: Finding circles

Equation for circle is

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

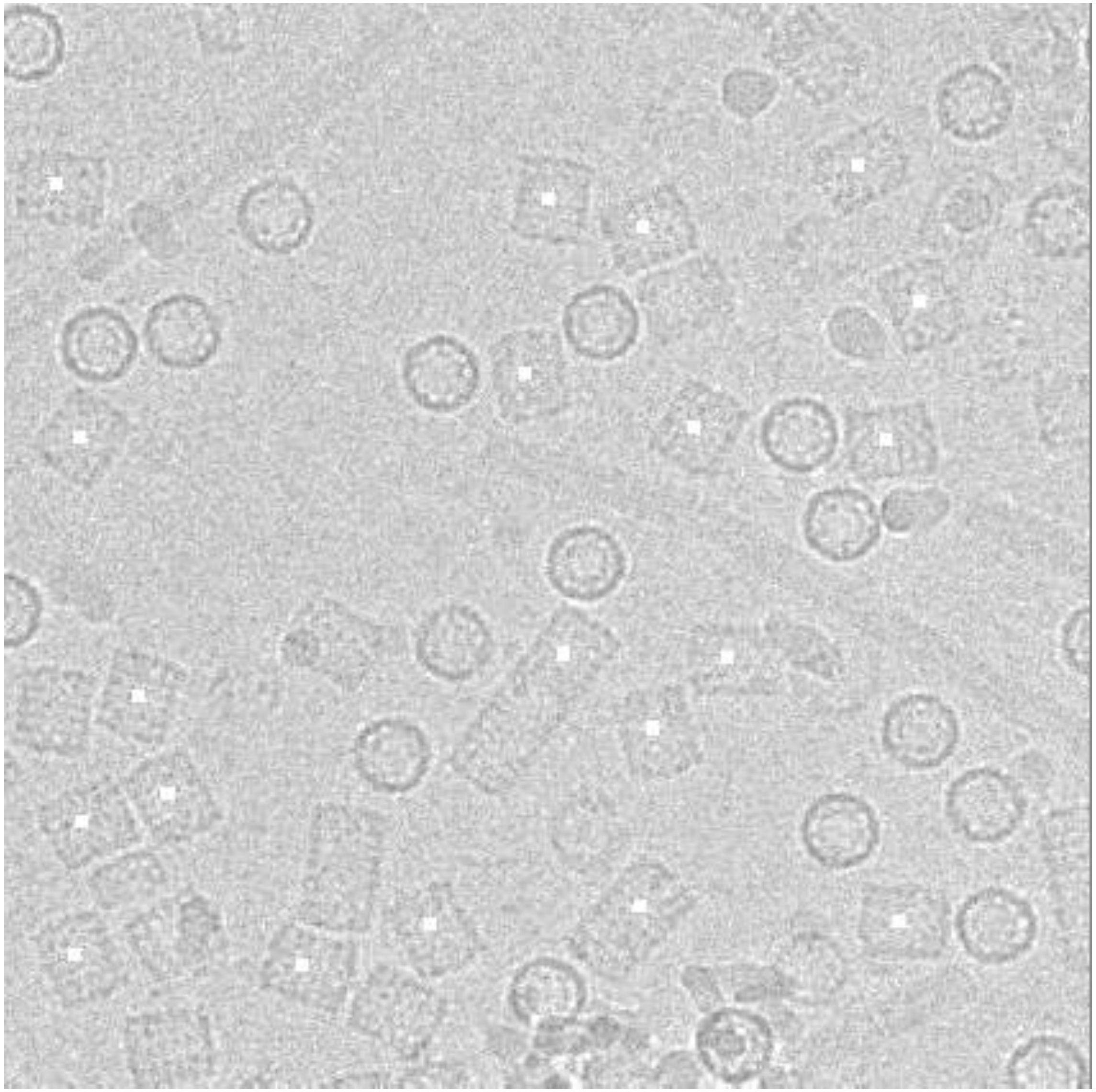
where the parameters are the circle's center (x_c, y_c) and radius r.

Three dimensional generalized Hough space.

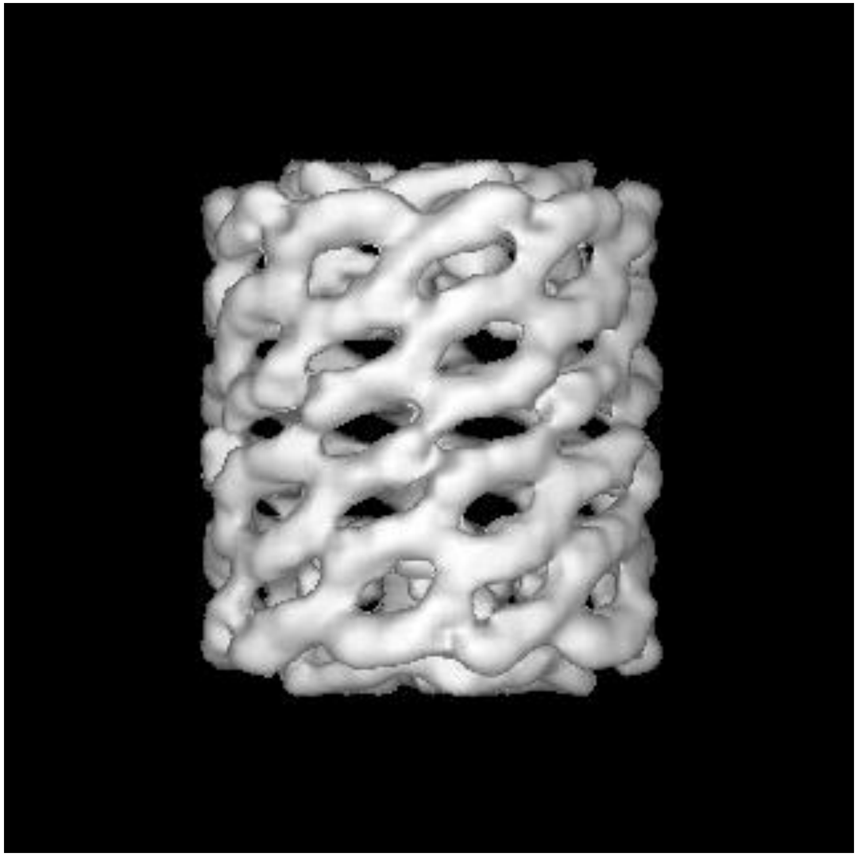
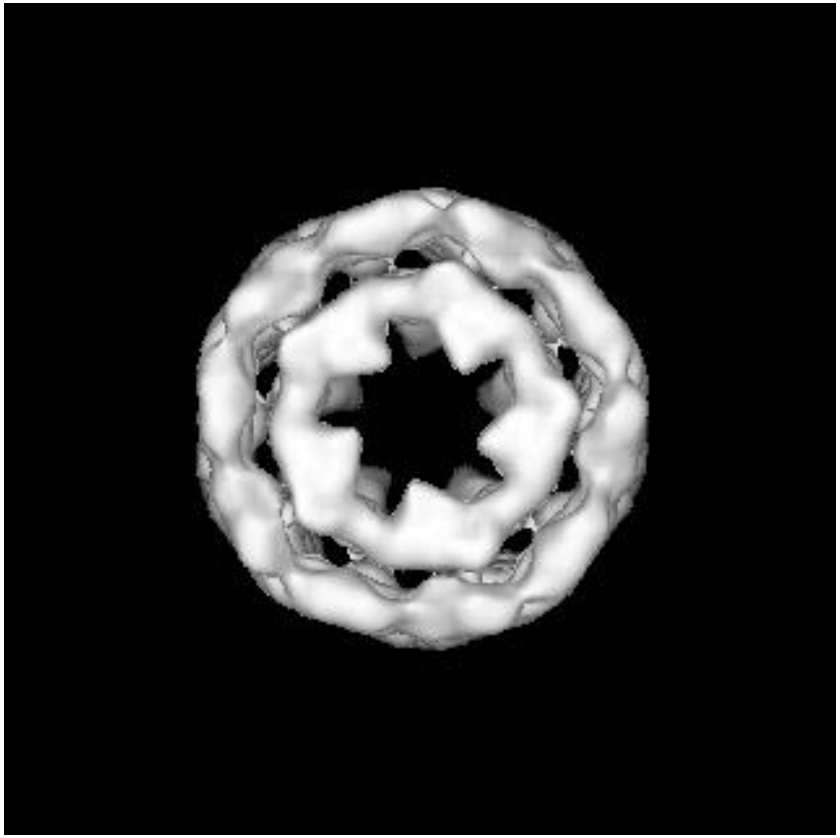
Given an edge point (x,y),

1. Loop over all values of (x_c, y_c),
2. Compute r
3. Increment $H(x_c, y_c, r)$

Transmission Electron Microscopy (TEM) Image of Keyhole Limpet Hemocyanin (KLH) with detected particles



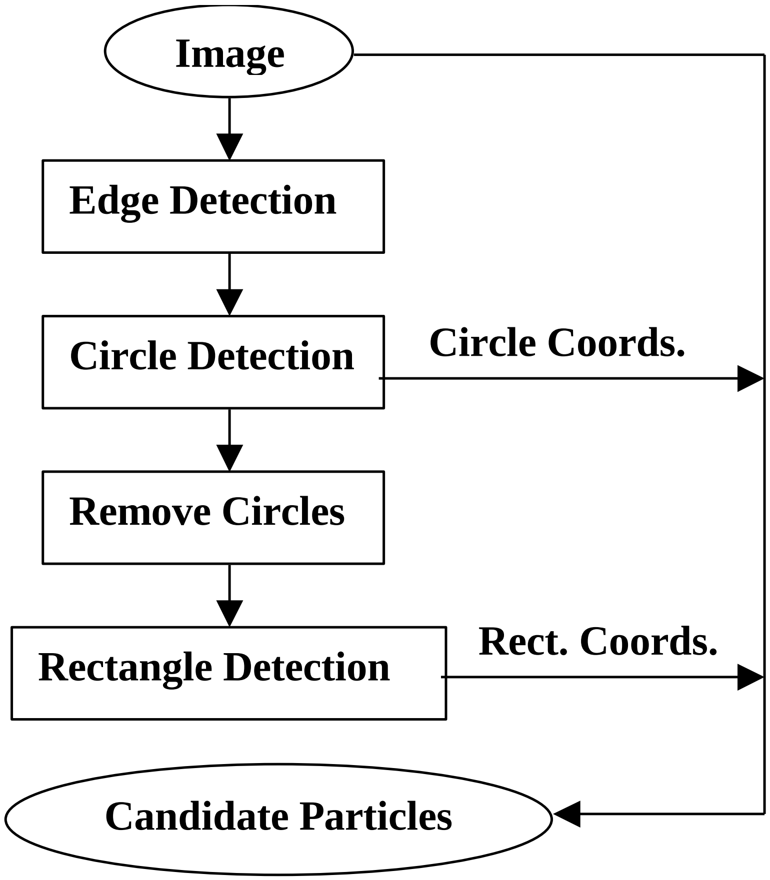
3D Maps of KLH

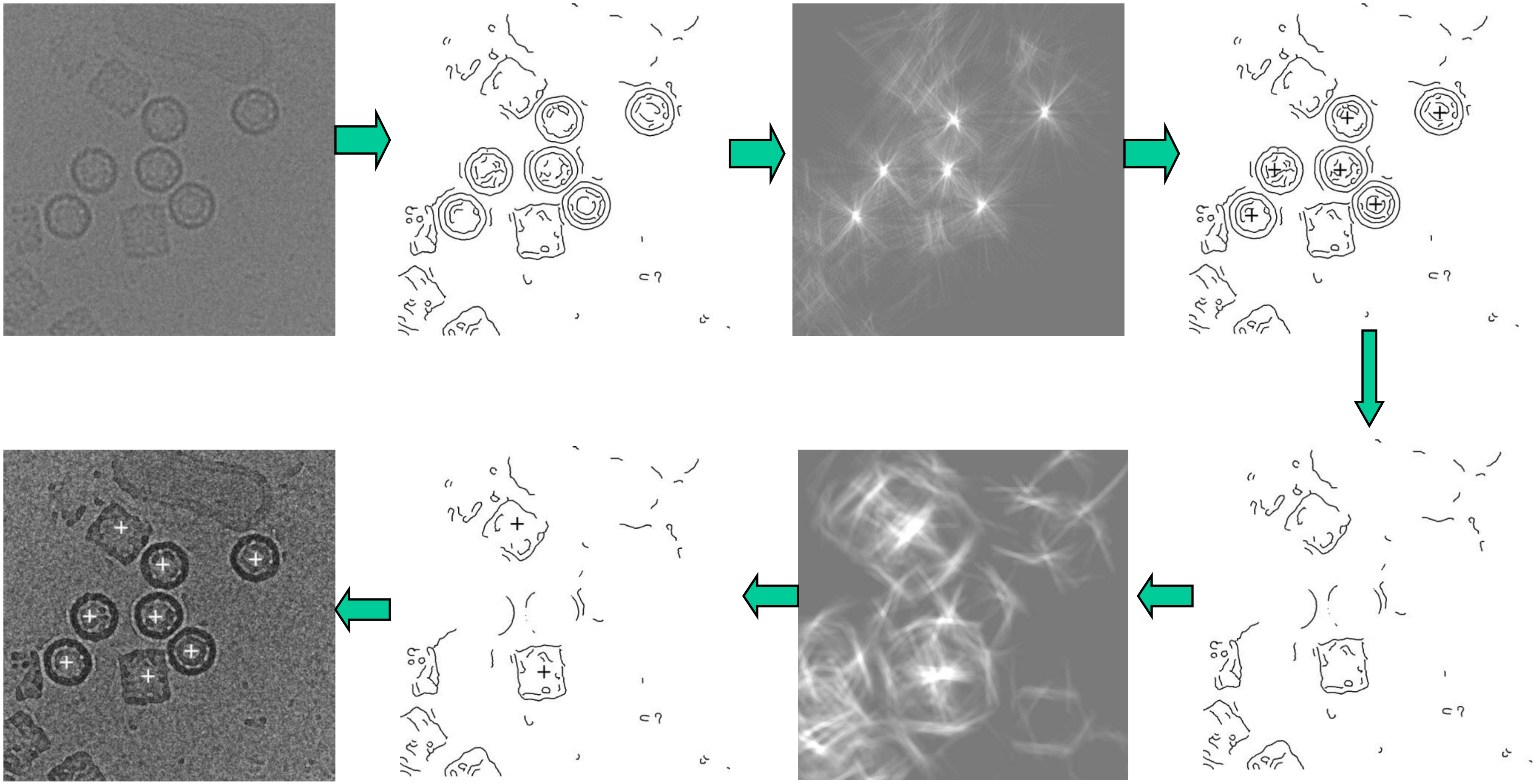
Three-dimensional maps of KLH at a resolution of 23.5 Å reconstructed using particles extracted either manually or automatically as described in the text. (a), (b) The side- and top- view of a 3D map reconstructed from a set of 1042 manually selected particle images.

Processing in Stage 1 for KLH

- Canny edge detection.
- A sequence of ordered Hough transforms (HTs) is applied in order from the computationally simplest one to the most complex one.
- Edges covered by the detected shapes are removed immediately from edge images following the application of the last HT.

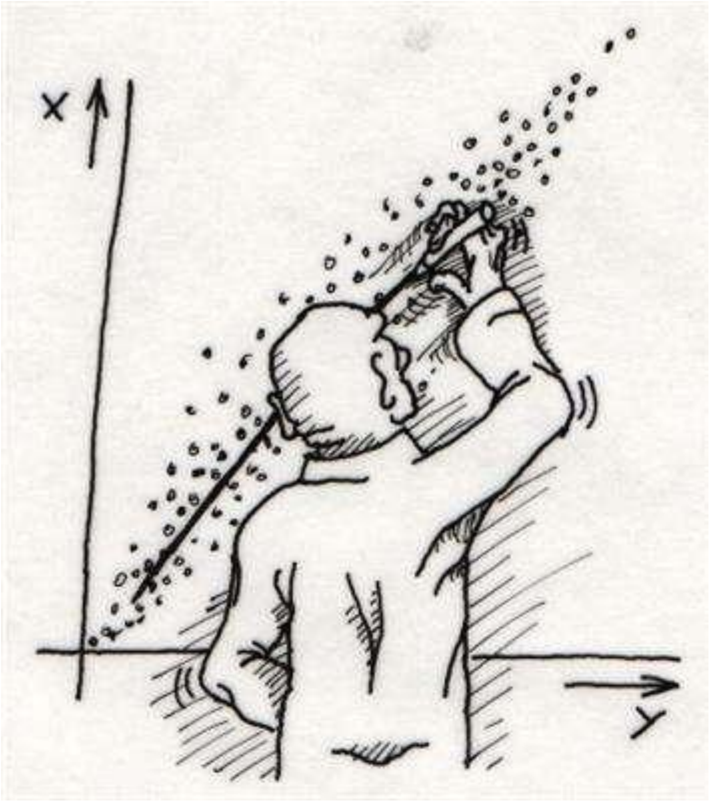


Picking KLH Particles in Stage 1

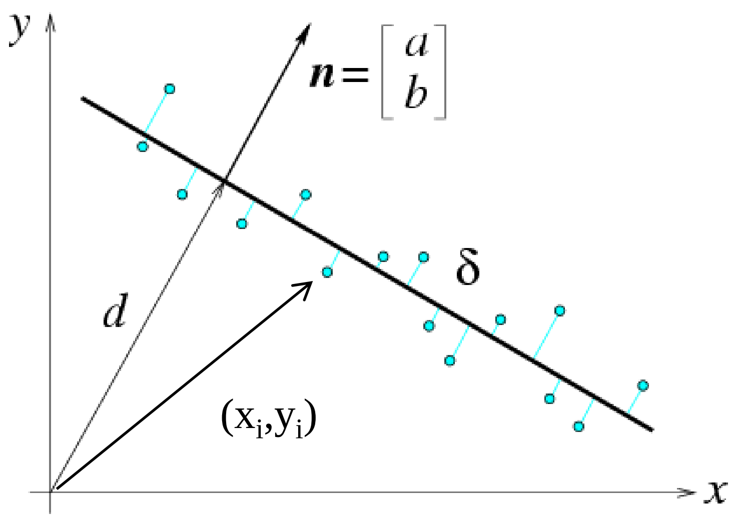


Zhu et al., IEEE Transactions on Medical Imaging, 2003

Line Fitting



Line Fitting



Given n points (x_i, y_i), estimate parameters of line

$$ax_i + by_i - d = 0$$

subject to the constraint that

$$a^2 + b^2 = 1$$

Note: $ax_i + by_i - d$ *is distance from* (x_i, y_i) to line.

Problem: minimize

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

Cost Function:
Sum of squared distances between each point and the line

with respect to (a,b,d).

1. Minimize E with respect to d:

$$\frac{\partial E}{\partial d} = 0 \Rightarrow d = \frac{1}{n}\sum_{i=1}^{n} ax_i + by_i = a\bar{x} + b\bar{y}$$

Where $(\bar{x}, \bar{y})$ is the mean of the data points

Line Fitting

2. Substitute *d* back into E

$$E = \sum_{i=1}^{n} [a(x_i - \bar{x}) + b(y_i - \bar{y})]^2 = |\mathcal{U}\boldsymbol{n}|^2$$

where $\mathcal{U} = \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \dots & \dots \\ x_n - \bar{x} & y_n - \bar{y} \end{pmatrix}$

where $\mathbf{n}=(a\ b)^T$.

3. Minimize $E=|Un|^2=\boldsymbol{n}^T U^T U\boldsymbol{n}=\boldsymbol{n}^T S\boldsymbol{n}$ with respect to *a*, *b* subject to the constraint $\mathbf{n}^T\mathbf{n} = 1$. Note that S is given by

$$S = \mathcal{U}^T\mathcal{U} = \begin{pmatrix} \sum_{i=1}^{n} x_i^2 - n\bar{x}^2 & \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} \\ \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} & \sum_{i=1}^{n} y_i^2 - n\bar{y}^2 \end{pmatrix}$$

which is real, symmetric, and positive definite

Line Fitting

4. This is a constrained optimization problem in ***n***. Solve with Lagrange multiplier

$$L(\boldsymbol{n}) = \mathbf{n}^T S \mathbf{n} - \lambda(\mathbf{n}^T\mathbf{n} - 1)$$

Take partial derivative (gradient) w.r.t. **n** and set to 0.

$$\nabla \mathbf{L} = \mathbf{2}S\mathbf{n} - \mathbf{2\lambda n} = 0$$

or

$$S\mathbf{n} = \boldsymbol{\lambda}\mathbf{n}$$

n*=(a,b)* is an Eigenvector of the symmetric matrix *S* (the one corresponding to the smallest Eigenvalue).

5. d is computed from Step 1.

RANdom Sample Consensus (RANSAC)

Slides adapted from

Frank Dellaert and Marc Pollefeys

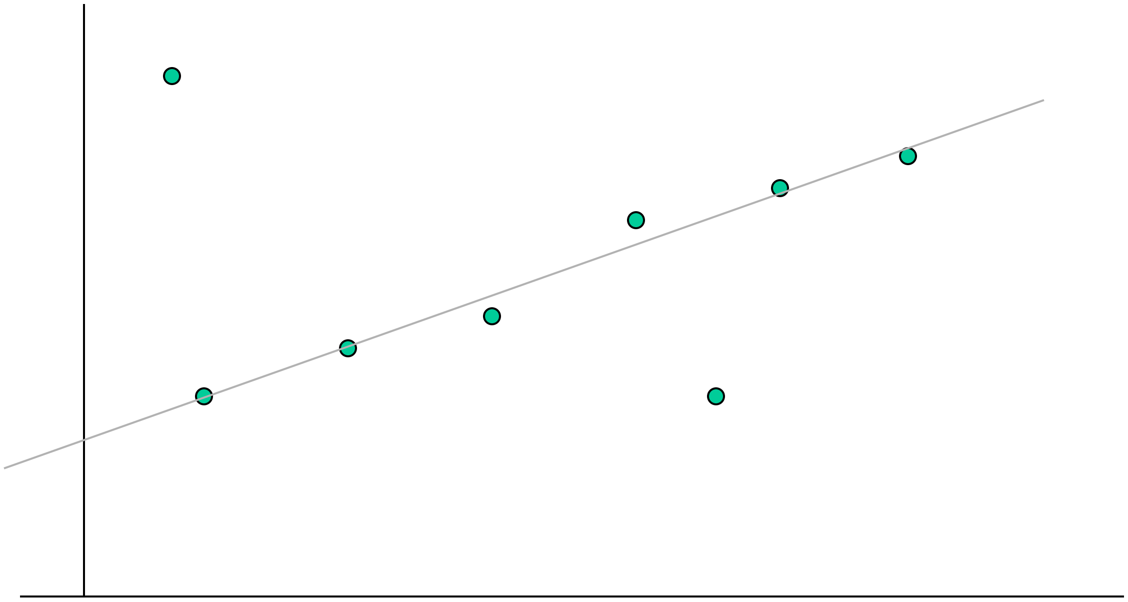
Motivation

- Estimating parameters of models in the presence of outlier data points
  - Lines (two parameters)
  - Homographies for mosaicing or rectification (8 parameters)
  - Essential matrix
  - And other models (circle, ellipses)

- For SFM: keypoints in two images

Simpler Example

- Fitting a straight line



- Inliers
- Outliers

Discard Outliers

- No point with d > t
- RANSAC:
  - RANdom SAmple Consensus
  - Fischler & Bolles 1981
  - Copes with a large proportion of outliers

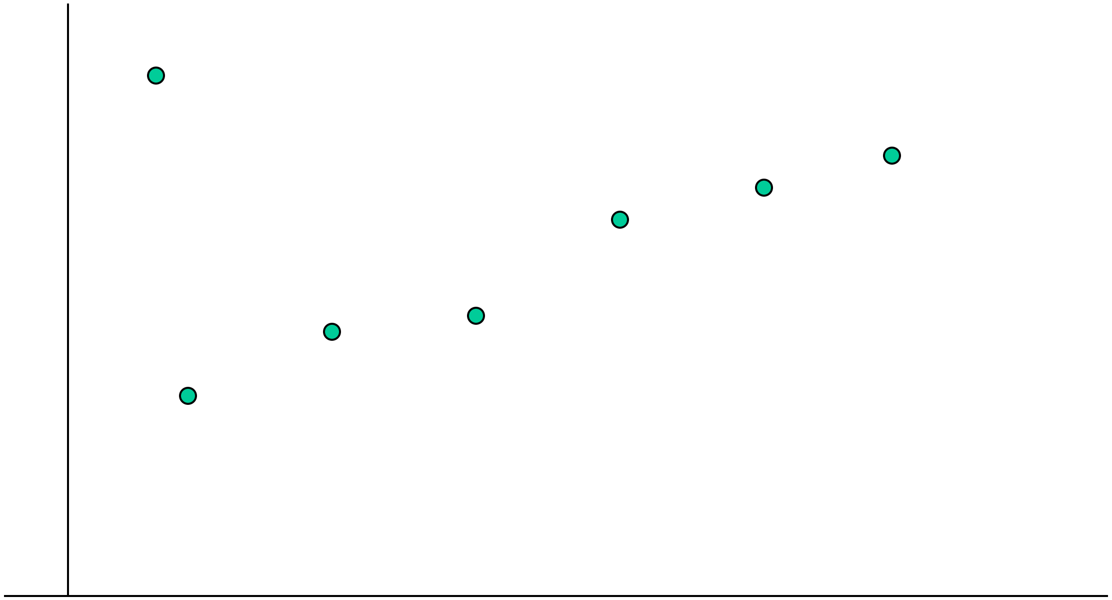
RANSAC Idea applied to line fitting

Problem: Given S points and threshold τ, determine best fit line in presence of outliers

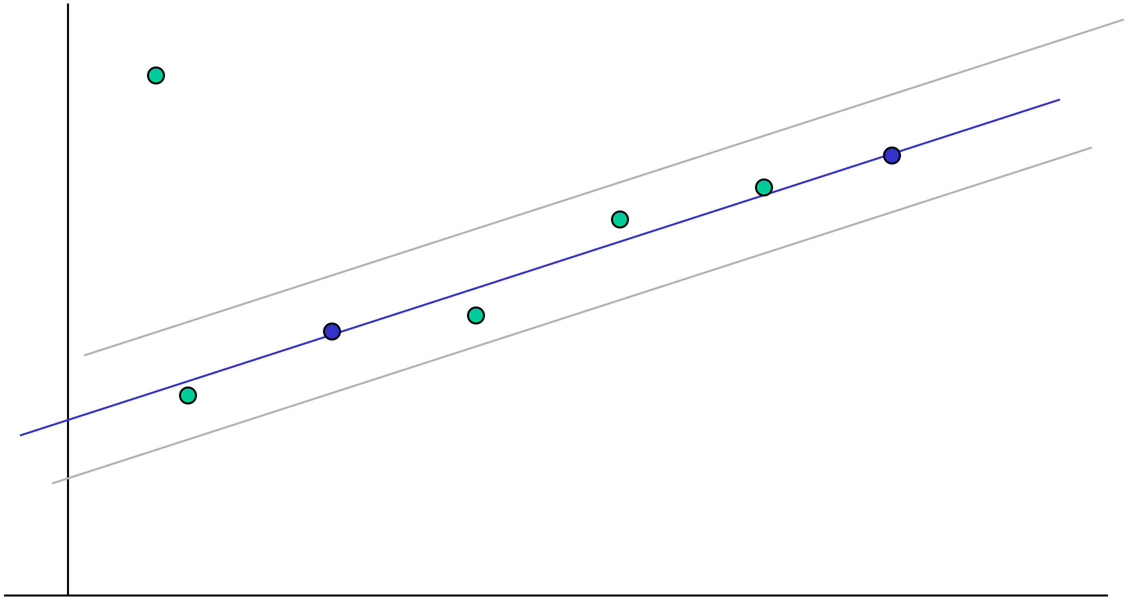
Repeat N times

- Select two points at random
- Determine line equation from the two points
- Count number of points that are within distance τ from the line. This is called the "support" of the line and it's the number of inliers
- Line with the greatest support wins

Why will this work ?

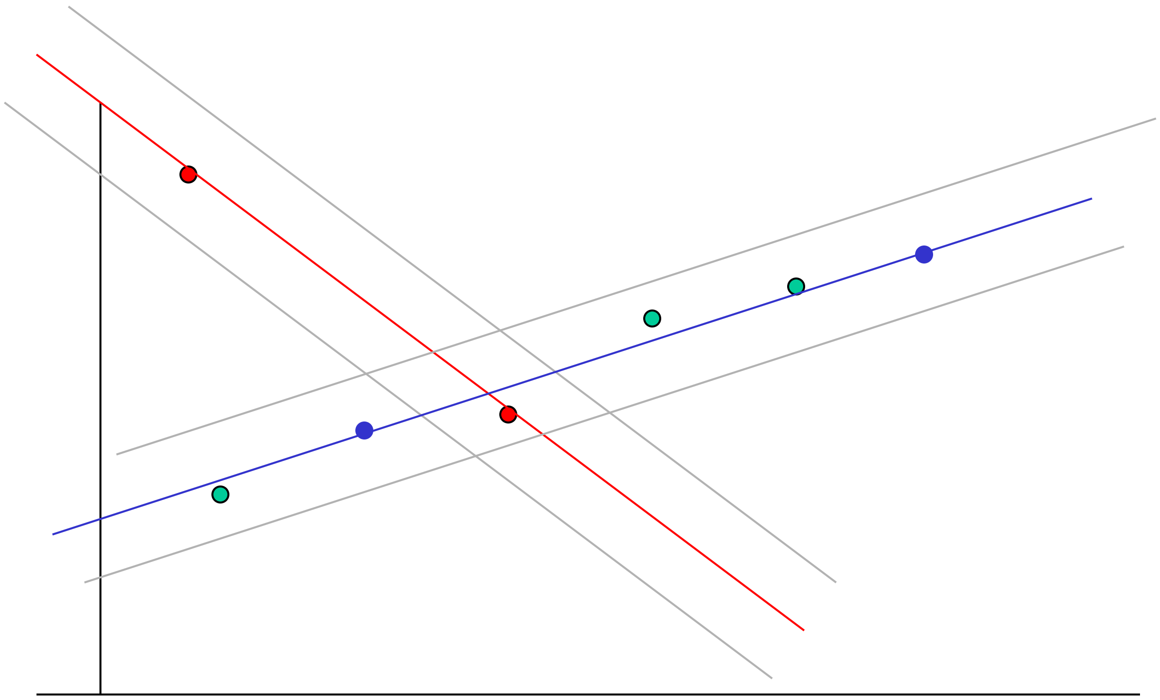


Why will this work ?



- Best line has most support
  - More support -> better fit

Why will this work ?



- Best line has most support
  - More support -> better fit

RANSAC More Generally

- What do we need to apply RANSAC
  1. A parameterized model
  2. A way to estimate the model parameters from s data points $\{x_1, ..., x_s\}$
  3. Given the parameters of the model, a way to estimate the distance from a data point x_i to the model

RANSAC More Generally

Objective

Robust fit of model to data set S which contains outliers

Algorithm

REPEAT

(i) Randomly select a sample of *s* data points from S

(ii) Instantiate the model from this sample.

(iii) Determine the set of data points S_i which are within a distance threshold ***t*** of the model. The set S_i is the **consensus set** of samples and defines the inliers of S.

(iv) $S_{largest} = S_i$ if S_i is larger than $S_{largest}$

UNTIL (The size of S_i is greater than some threshold *T) OR*

(There have been N samples)

The model is re-estimated using all the points in $S_{largest}$

Using RANSAC to estimate the fundamental matrix

- What is the model?

    Fundamental matrix

- What is the sample size and where do the samples come from?

    8 points in each image or 8 matched pairs (e.g., SIFT matches)

- What distance do we use to compute the consensus set?
    1. L^2 distance of points to epipolar line
    2. Epipolar constraint
    3. Reprojection error or its Sampson error approximation

- How often do outliers occur

    Usually not known in advance

Feature points extracted by a corner detector

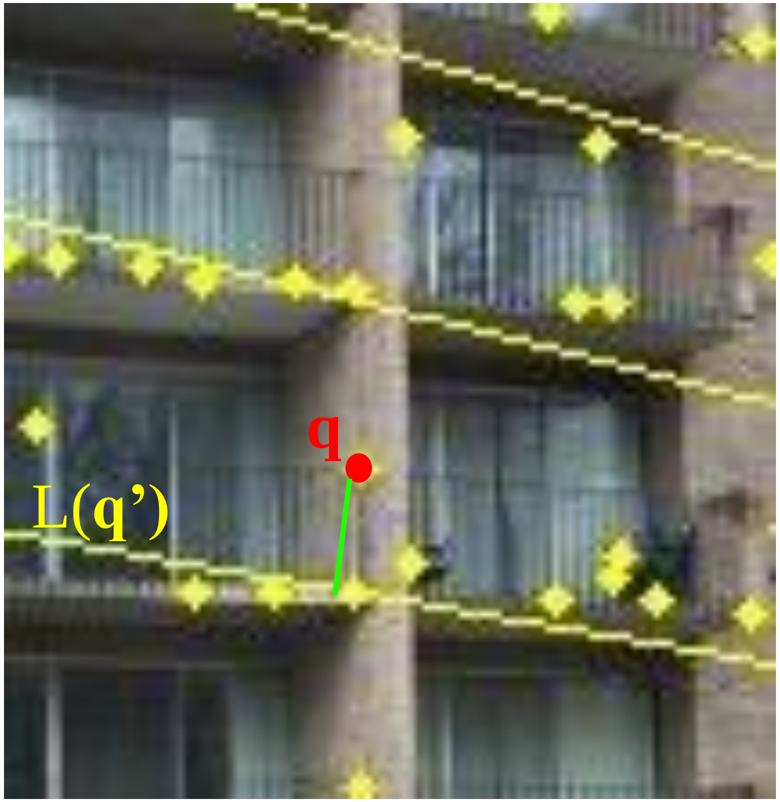
Matched points by RANSAC

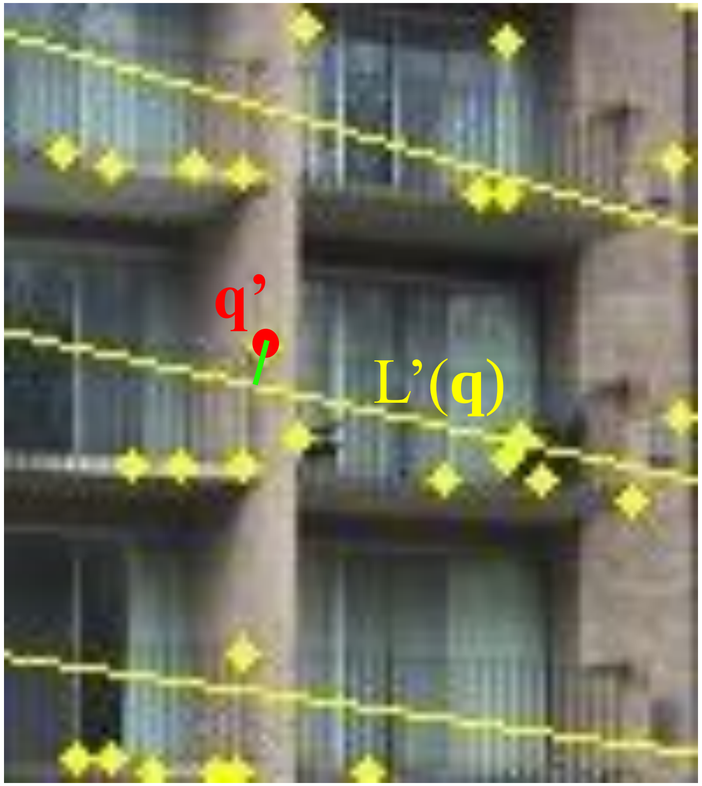
Yellow: Inliers (correct matches)
Cyan: Outliers (mismatches)

Putative matches of the feature points in both images are computed by using a correlation measure for points in one image with a features in the other image. Only features within a small window are considered to limit computation time. Mutually best matches are retained. RANSAC is used to robustly determine F from these putative matches.

Putative Epipolar Geometry during an iteration of RANSAC

Distance 1 for computing consensus set

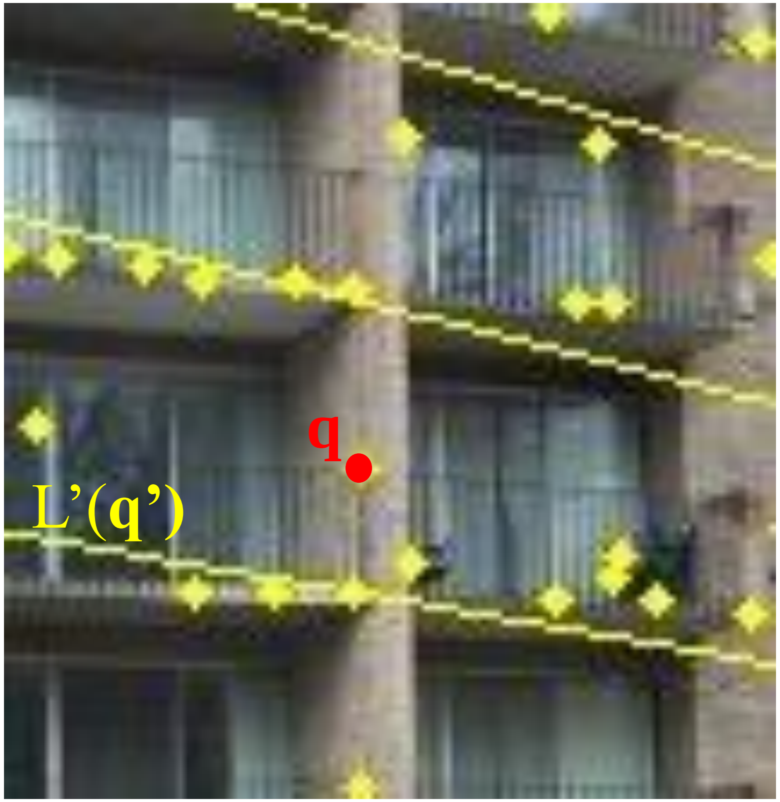


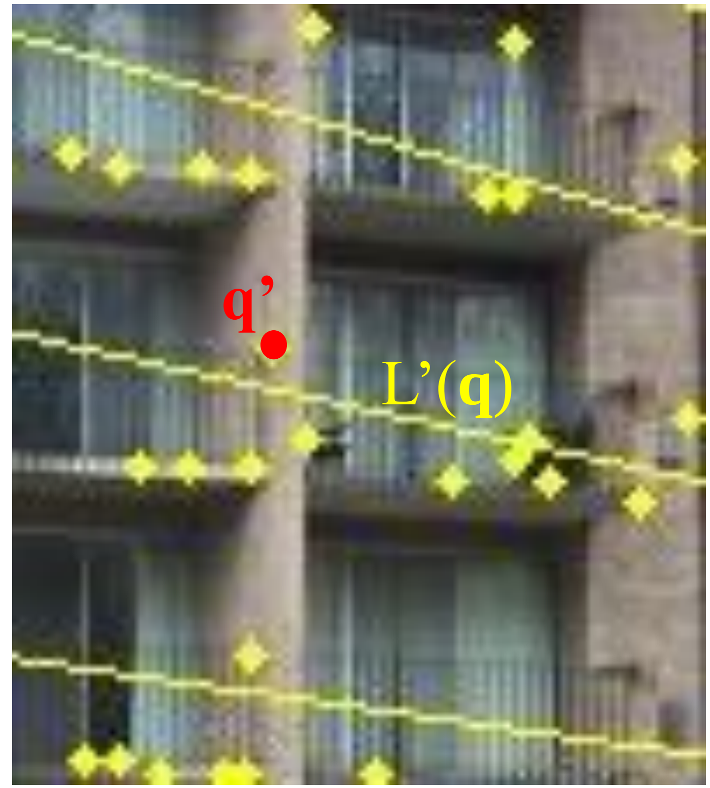


Distance between matching feature points **q** and **q'** using point-line distance

$$\text{Dist1}(\mathbf{q},\mathbf{q}') = \text{dist}(\mathbf{q}, L(\mathbf{q}')) + \text{dist}(\mathbf{q}', L'(\mathbf{q}))$$

Distance 2 for computing consensus set





Distance between matching feature points **q** and **q'** using epipolar constraint

$$\text{Dist2}(\mathbf{q},\mathbf{q}') = \mathbf{q}'^T\mathbf{F}\mathbf{q}$$

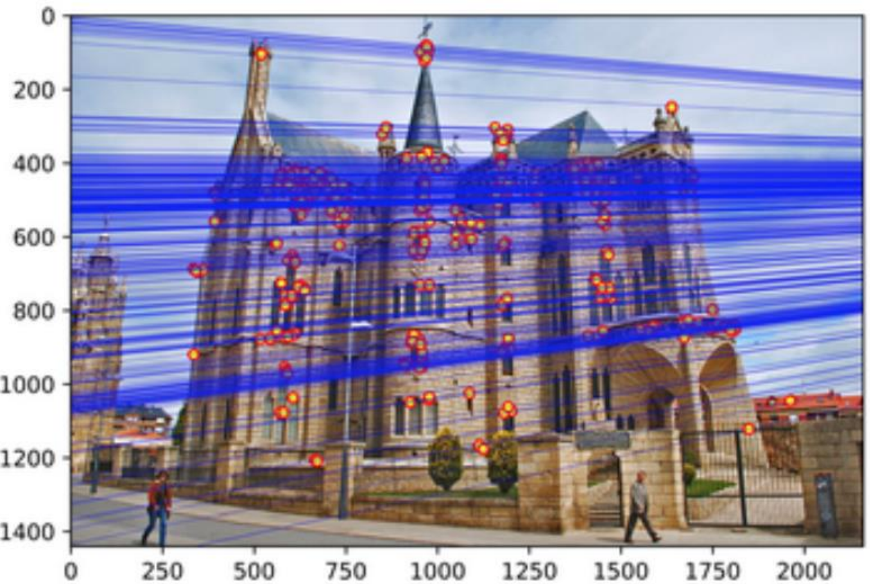
Where **F** is the fundamental matrix.

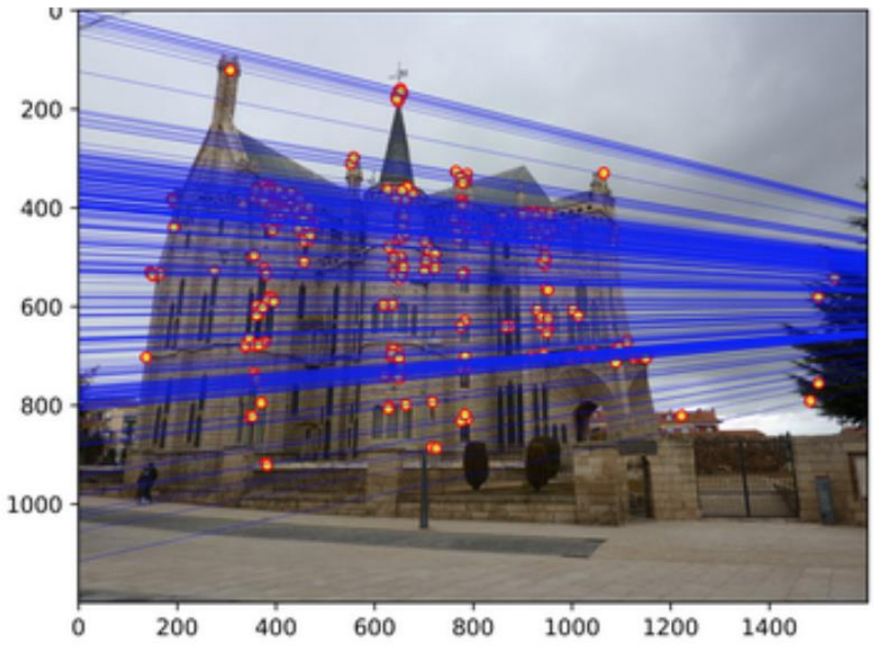
Note: **F** must be normalized, perhaps by dividing by ||**F**||

Two input images with features points

Matching points computed with RANSAC

Epipolar lines





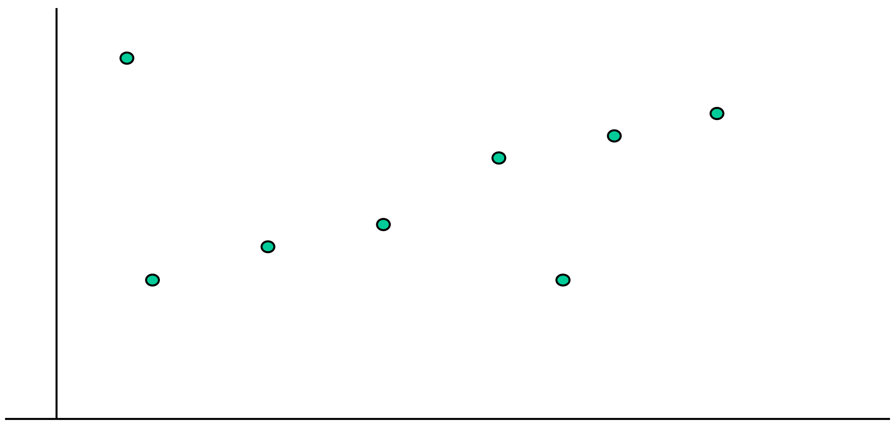
How many samples are needed to be confident that you have found only inliers?

Choose N (number of samples) so that, with probability p, at least one of N random samples is free from outliers, e.g., p=0.99

Let

s: sample size (i.e., number of points needed for the model)

e: proportion of outliers in the data



- For line fitting: $s = 2$
- For this example, there are 6 inliers and 2 outliers: $e = 2/8 = 0.25$

RANSAC applied to estimating Fundamental Matrix

- The 8 point algorithm requires having 8 correctly matching **pairs** of points in a pair of images.
- Hypothetically, if a matching method using SIFT descriptors leads to 40% of pairs being incorrect, Then the chance of randomly selecting 8 pairs, all of which are correct, is $(1\text{-}0.4)^8 = 0.016$
- To be 99% sure that at least one of the randomly selected group of 8 pairs of points is correct implies that you must draw 272 pairs (i.e., random samples).

How many samples are needed to be confident that you have found only inliers?

Choose N (number of samples) so that, with probability p, at least one of N random samples is free from outliers, e.g., p=0.99

Let

s: sample size (i.e., number of points needed for the model)

e: proportion of outliers in the data

$$\left(1-(1-e)^s\right)^N = 1-p \quad \text{Solve for } N$$

$$N = \log(1-p)/\log\left(1-(1-e)^s\right)$$

Where does this equation come from?

$$\left(1 - (1 - e)^{s}\right)^{N} = 1 - p$$

- p: desired probability that at least one of N random samples is free from outliers
- $1- p$: probability that none of the samples is free from outliers
- s: sample size (i.e., number of points needed for the model)
- e: proportion of outliers in the data
- $1- e$: probability of a data sample being an inlier
- $(1- e)^s$: probability that s samples are all inliers
- $1 - (1 - e)^s$: probability that s samples contain at least one outlier
- $(1 - (1 - e)^s)^N$: probability that none of the samples is free from outliers

How many samples are needed to be confident that you have found only inliers?

Choose N (number of samples) so that, with probability p, at least one of N random samples is free from outliers, e.g., p=0.99

Let s: sample size (i.e., number of points needed for the model)

e: proportion of outliers in the data

$$\left(1-\left(1-e\right)^{s}\right)^{N}=1-p \quad \text{Solve for } N$$

$$N=\log\left(1-p\right)/\log\left(1-\left(1-e\right)^{s}\right)$$

	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Distance threshold

Choose threshold t so probability for inlier is α (e.g., $\alpha = 0.95$)

- Often empirically
- Zero-mean Gaussian noise σ then $d_\perp^2$ follows χ_m^2 distribution with m = codimension of model

(codimension of subspace = dimension of space – dimension of subspace)

Codimension	Model	t^2
1	E, F, 2D line	$3.84\sigma^2$
2	P	$5.99\sigma^2$

Number of inliers threshold

- Typically, terminate when inlier ratio reaches expected ratio of inliers

$$T = (1 - e)N$$

  - N: number of samples
  - e: proportion of outliers in the data

Next Lecture

- Optical flow and motion
- Additional, optional reading in Course Reserves
  - Introductory Techniques for 3-D Computer Vision, Trucco and Verri
    - Chapter 8: Motion