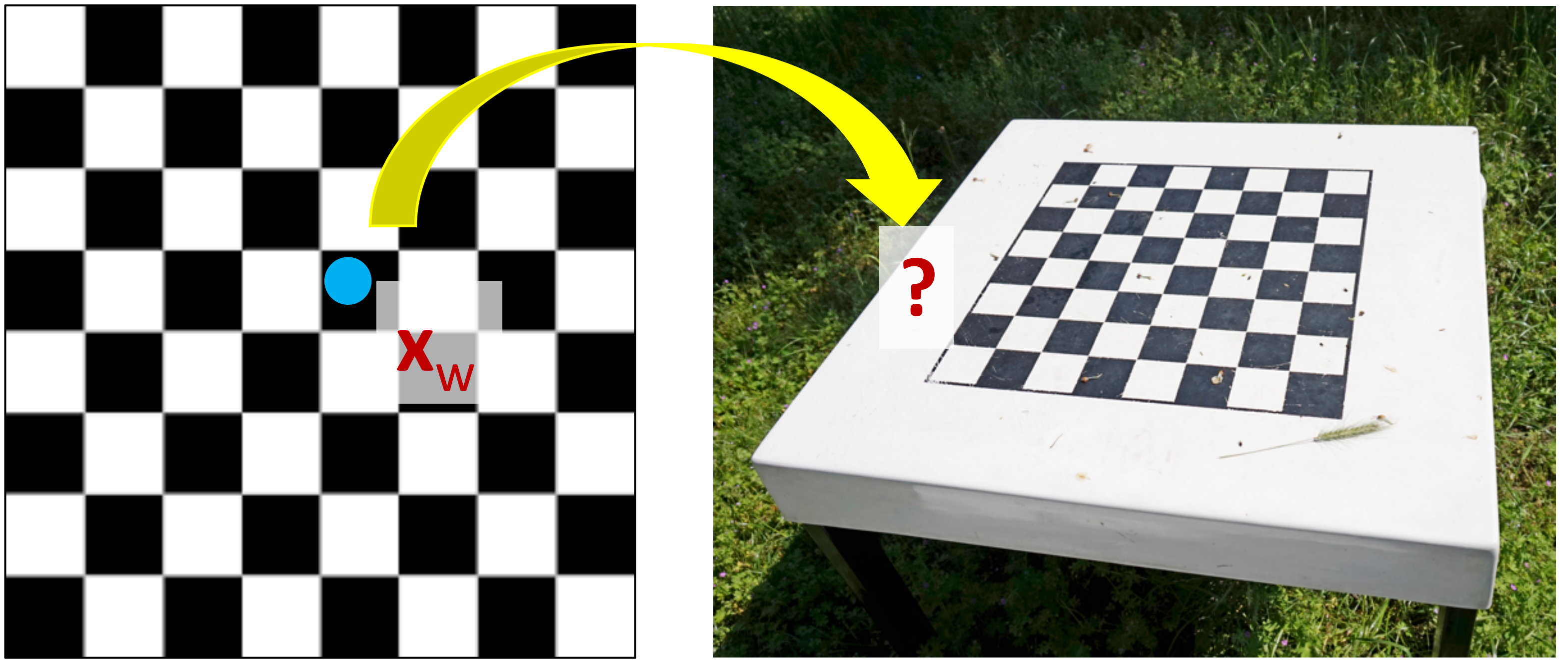
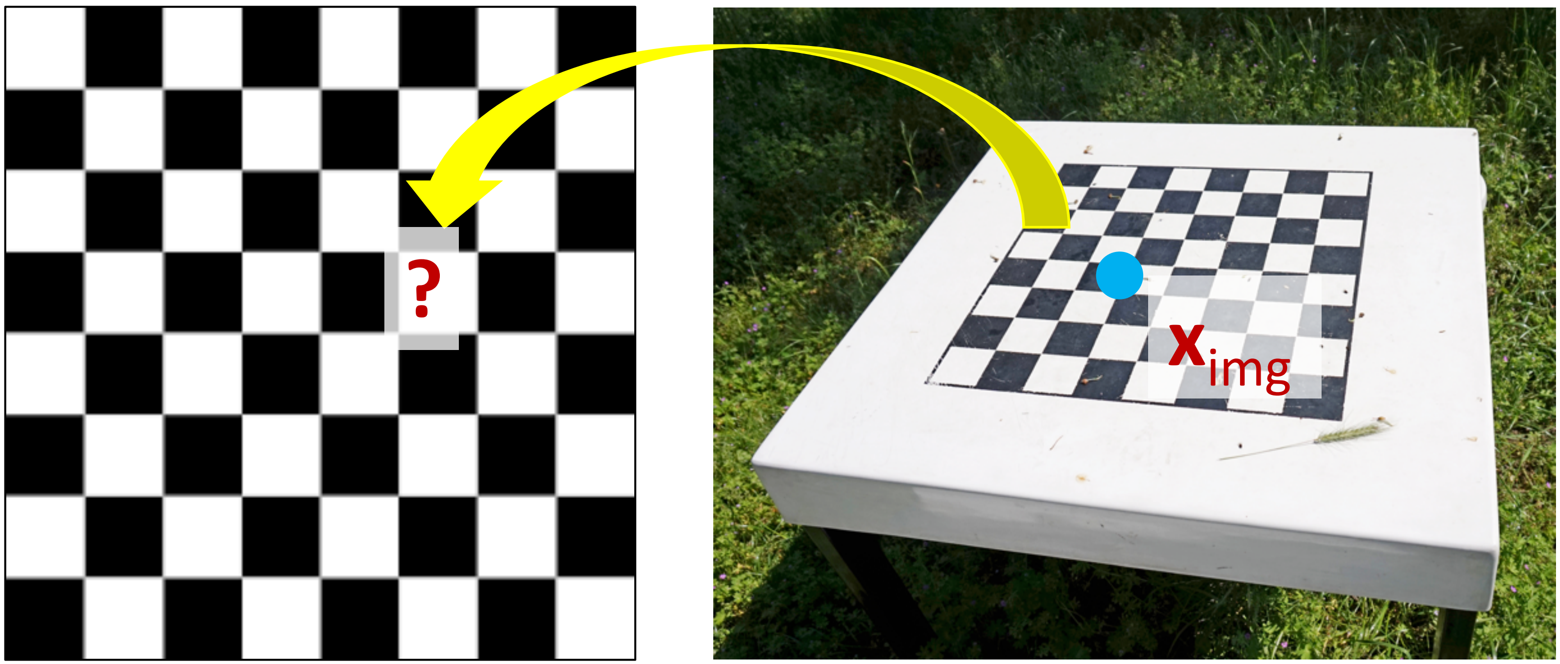
RANSAC continued

Homography estimation



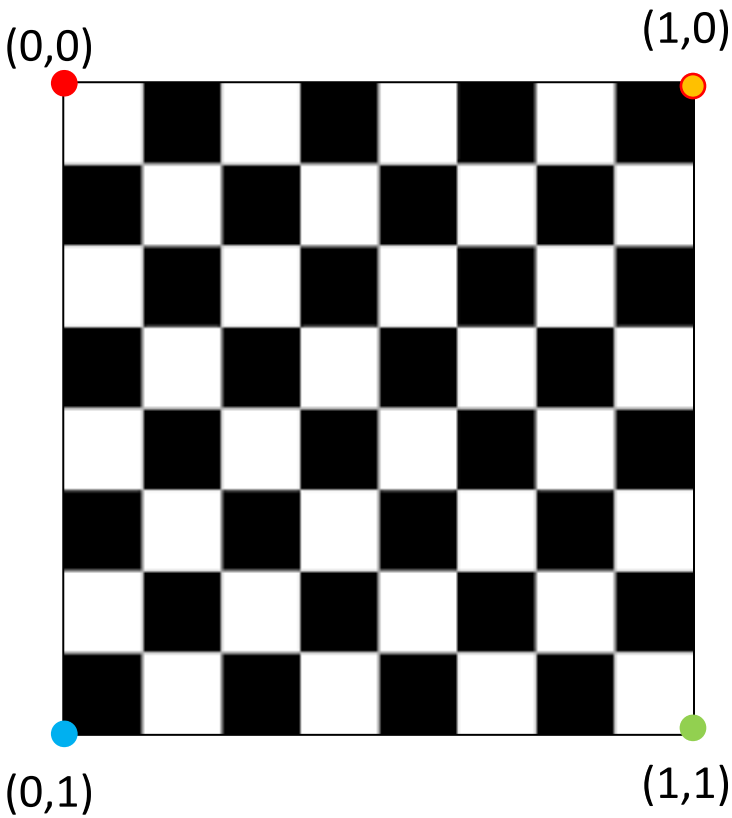
$$\vec{\mathbf{x}}_{img} \equiv H\vec{\mathbf{x}}_w$$

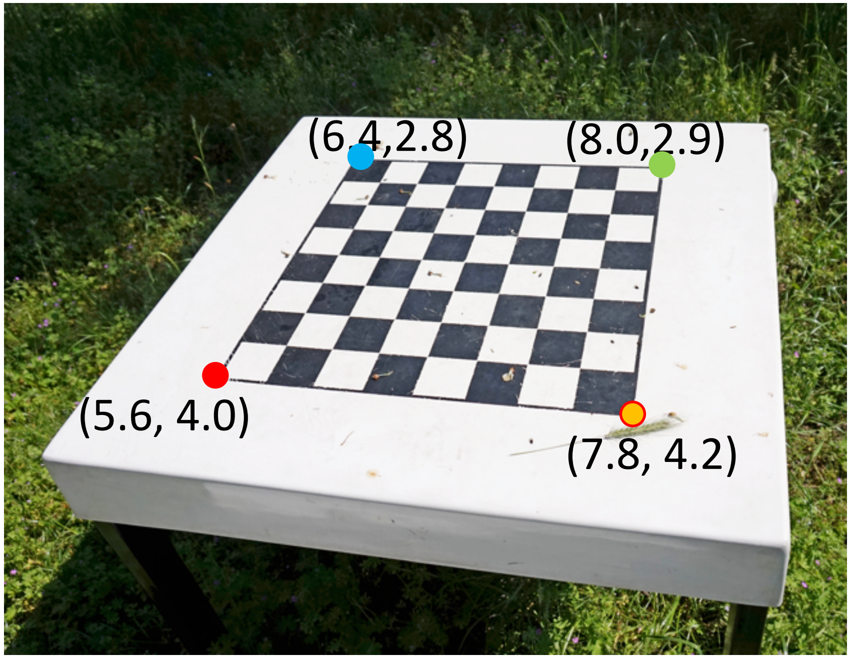
Homography estimation



$$\vec{\mathbf{x}}_w = H^{-1}\vec{\mathbf{x}}_{img}$$

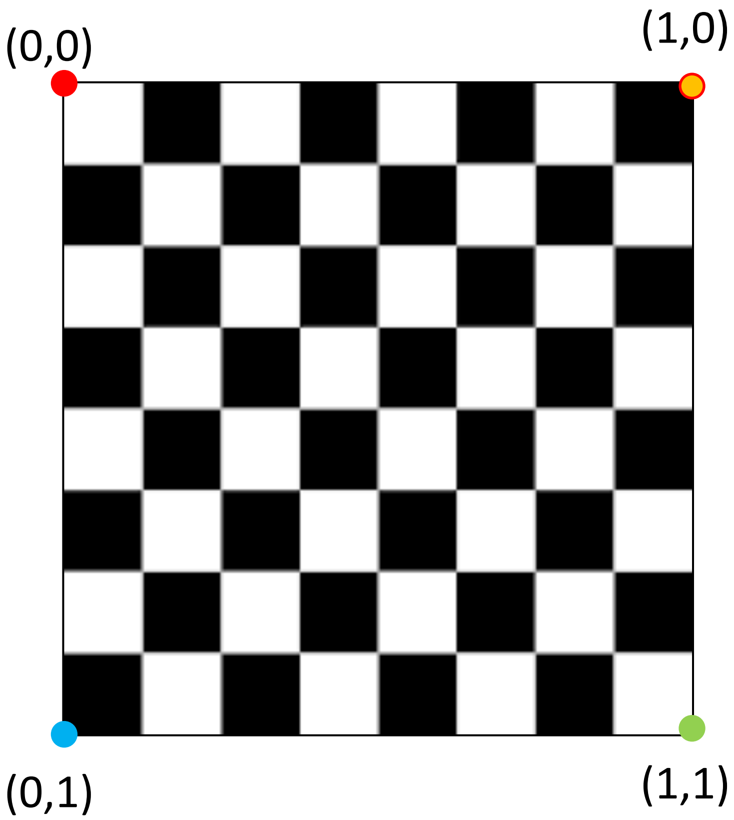
Homography estimation

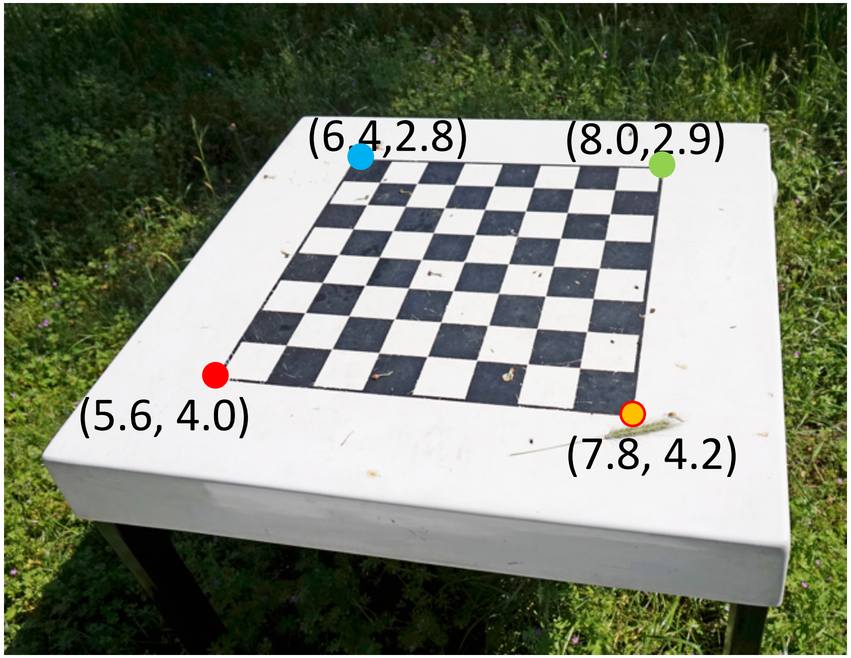




$$A\mathbf{h} = 0 \text{ s.t } \|\mathbf{h}\| = 1$$

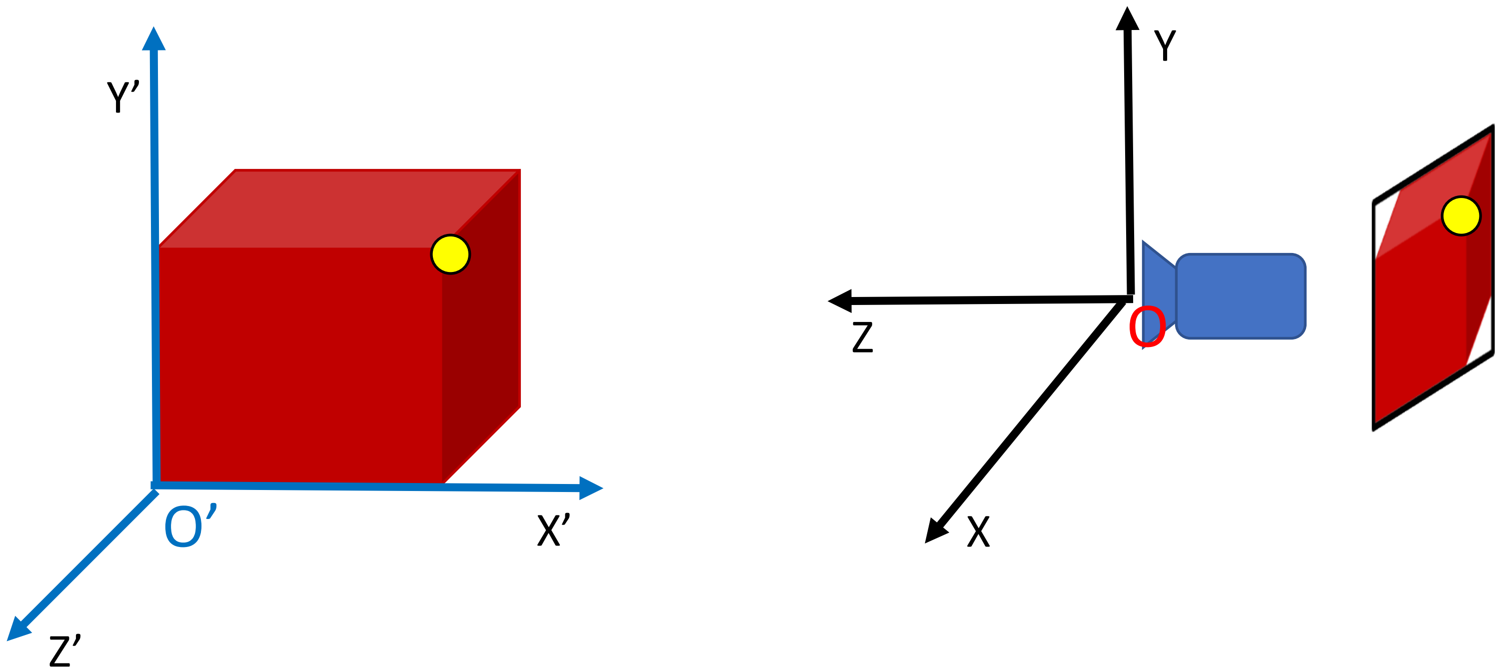
Homography estimation





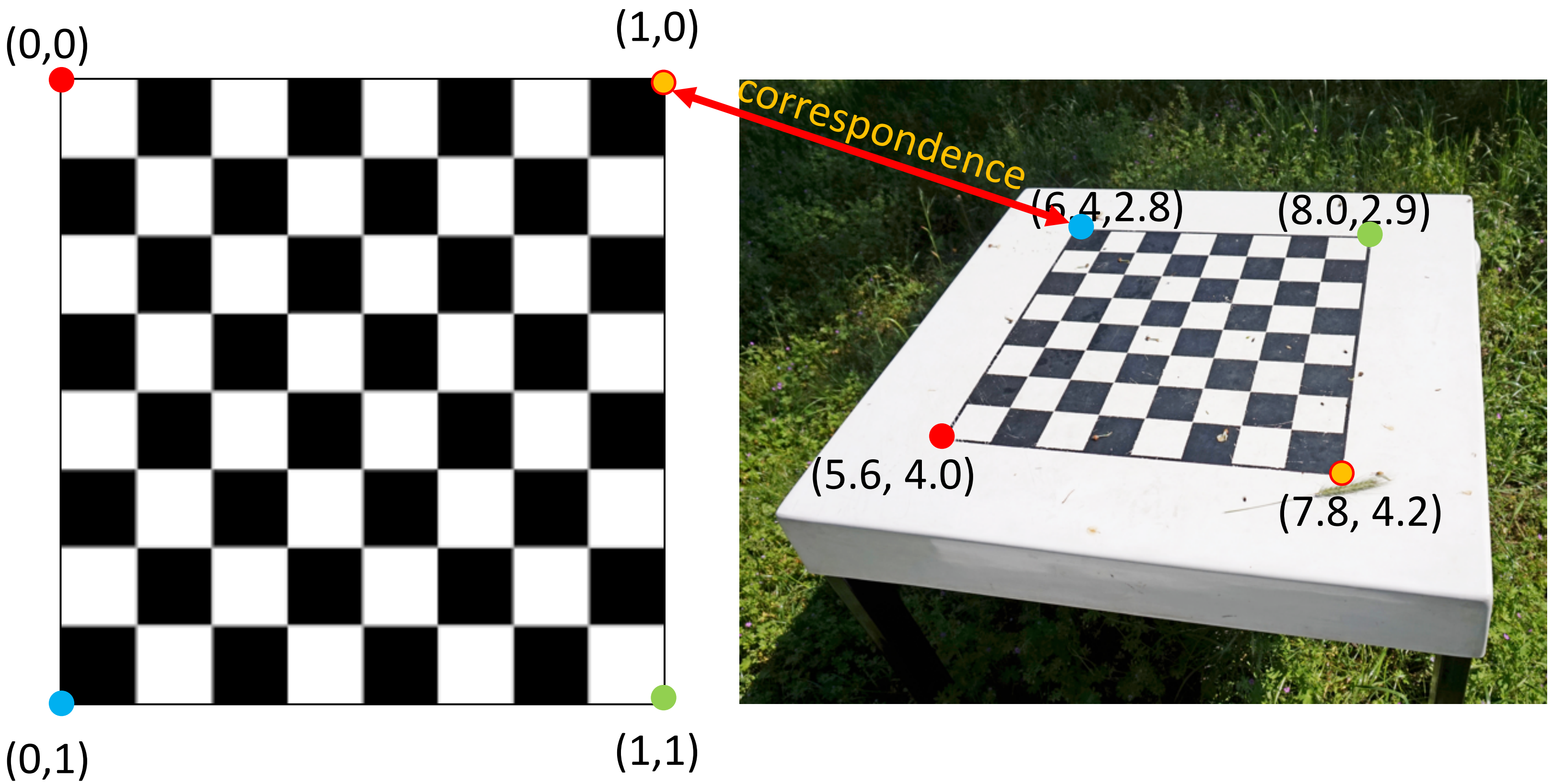
$$\min_{\mathbf{h}} \|A\mathbf{h}\|^2 \text{ s.t } \|\mathbf{h}\| = 1$$

Camera calibration



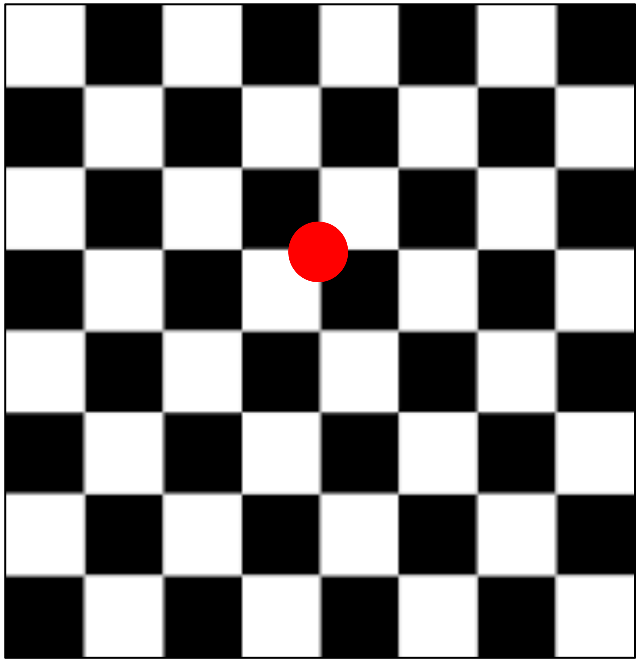
$$\min_{\mathbf{p}} \|A\mathbf{p}\|^2 \text{ s.t } \quad \|\mathbf{p}\| = 1$$

Homography estimation



$$\min_{\mathbf{h}} \|A\mathbf{h}\|^2 \text{ s.t } \|\mathbf{h}\| = 1$$

Homography estimation: obtaining correspondences



$$\min_{\mathbf{h}} \|A\mathbf{h}\|^2 \text{ s.t } \|\mathbf{h}\| = 1$$

Homography estimation

Given images A and B

1. Compute image features for A and B
2. Match features between A and B
3. Compute homography between A and B

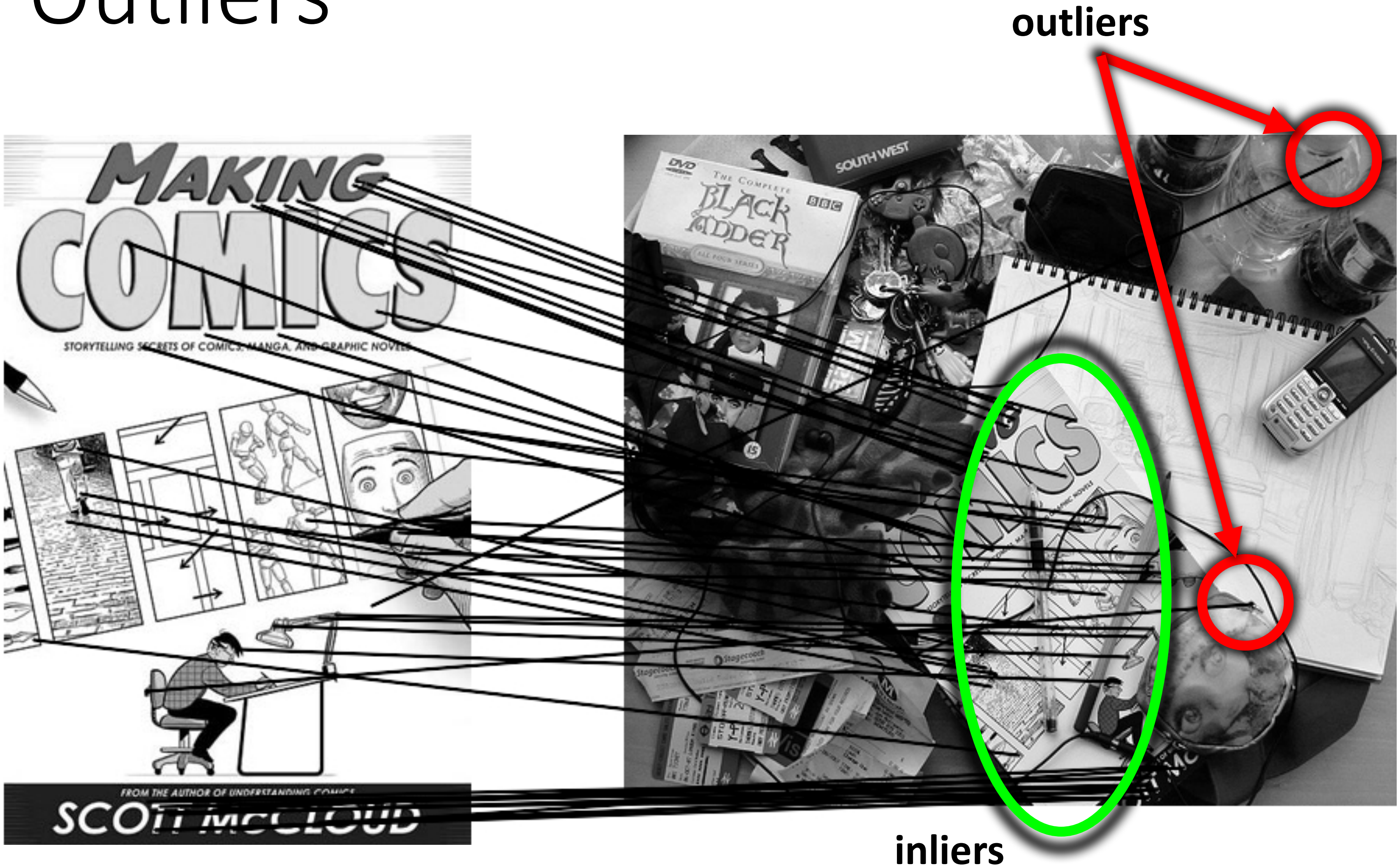
What do we do when correspondences are incorrect?

Homography fitting and incorrect correspondences

$$\min_{\mathbf{h}} \|A\mathbf{h}\|^2 \text{ s.t } \|\mathbf{h}\| = 1$$

- Correspondences create matrix A
- What if many correspondences are actually incorrect?
- Even true H cannot satisfy constraint!
- *Outliers*

Outliers

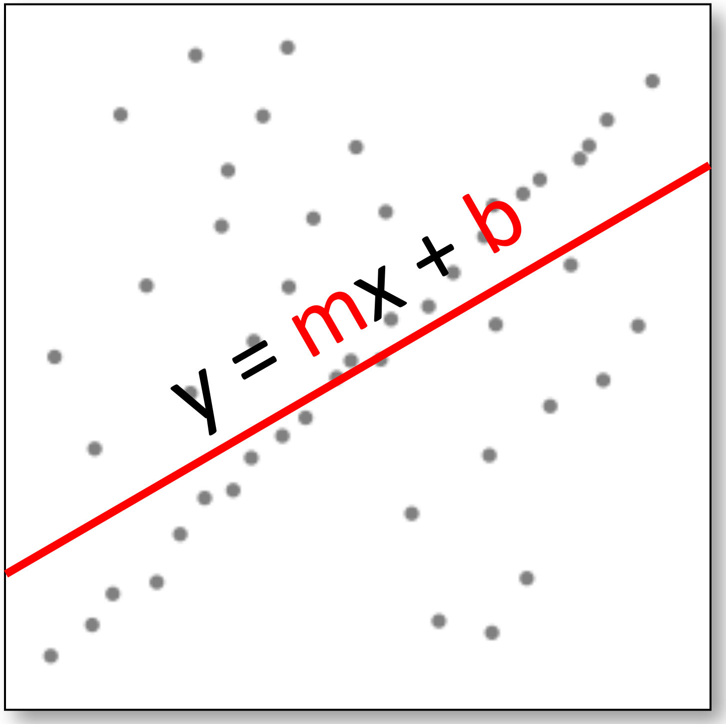

outliers
inliers
MAKING
COMICS
STORYTELLING SECRETS OF COMICS, MANGA, AND GRAPHIC NOVELS
FROM THE AUTHOR OF UNDERSTANDING COMICS
SCOTT McCLOUD
DVD
THE COMPLETE
BLACK ADDER
BBC
ALL FOUR SERIES
SOUTH WEST

A general class of problems

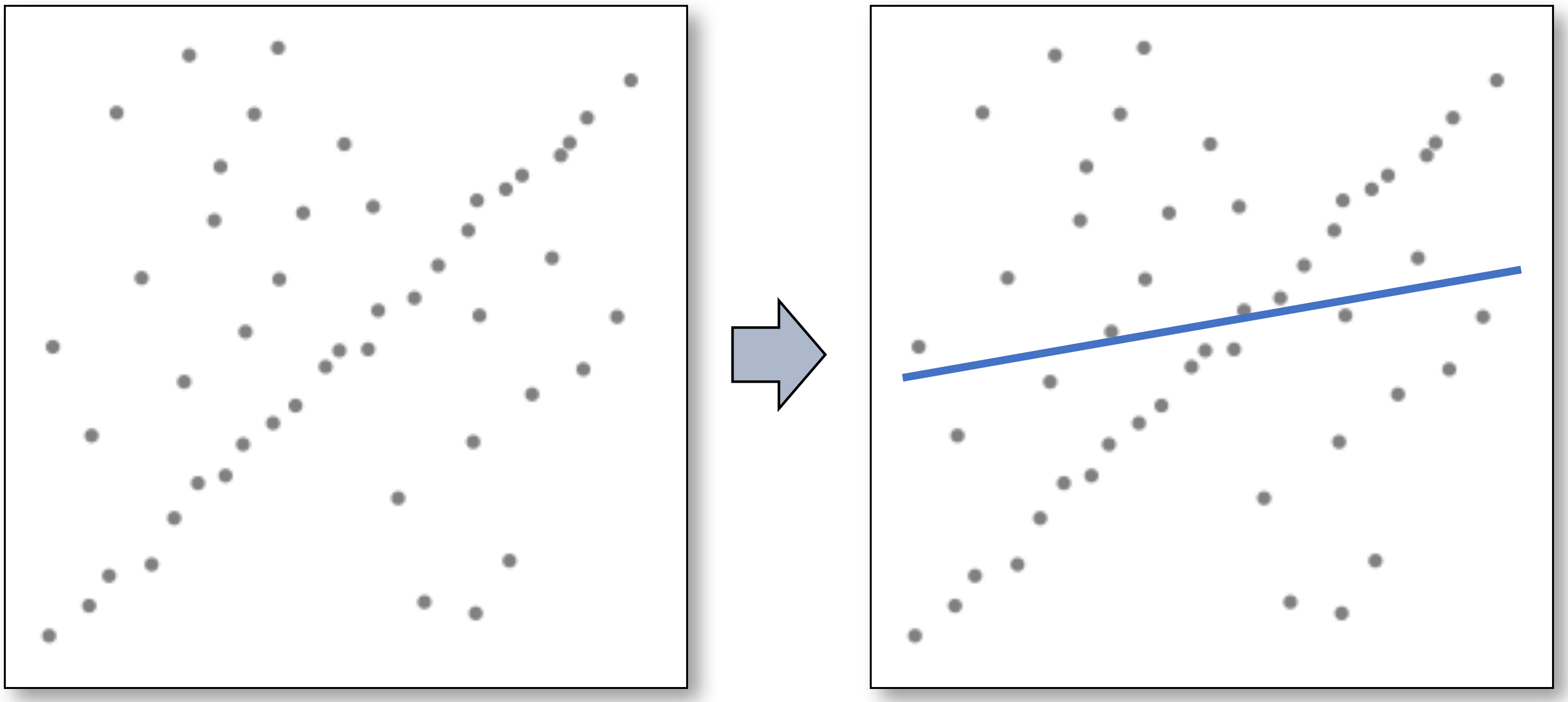
- Need to “fit a model”, i.e., ”find parameters”
  - e.g., H for homography
- Have some data points to find parameters
  - e.g. correspondences (x_w, x_{img})
- Need at least k data points to find parameters
  - e.g., 4 correspondences for homography
- Many data points might be completely incorrect, i.e., even correct model won’t fit them
  - e.g., incorrect correspondences

Another example

- Need to "fit a model", i.e., "find parameters"
  - e.g., m,b for line fitting
- Have some data points to find parameters
  - e.g. points (x, y)
- Need at least k data points to find parameters
  - e.g., 2 points for line
- Many data points might be completely incorrect, i.e., even correct model won't fit them



Robustness



Problem: Fit a line to these datapoints

Least squares fit

Robust model fitting

- Correct data = “inliers”, incorrect data = “outliers”
- If we knew inliers, fitting model is easy
  - e.g., for homography, set up matrix A

$$\min_{\mathbf{h}} \|A\mathbf{h}\|^2 \text{ s.t } \|\mathbf{h}\| = 1$$

- If we knew model, identifying inliers is easy
  - Inliers agree with model, outliers disagree

- Chicken and egg problem!

Key idea

- A *single model* will satisfy all inliers
- No single model will satisfy all outliers
    - Outliers will all *disagree* on model they like

"Happy families are all alike; every unhappy family is unhappy in its own way."

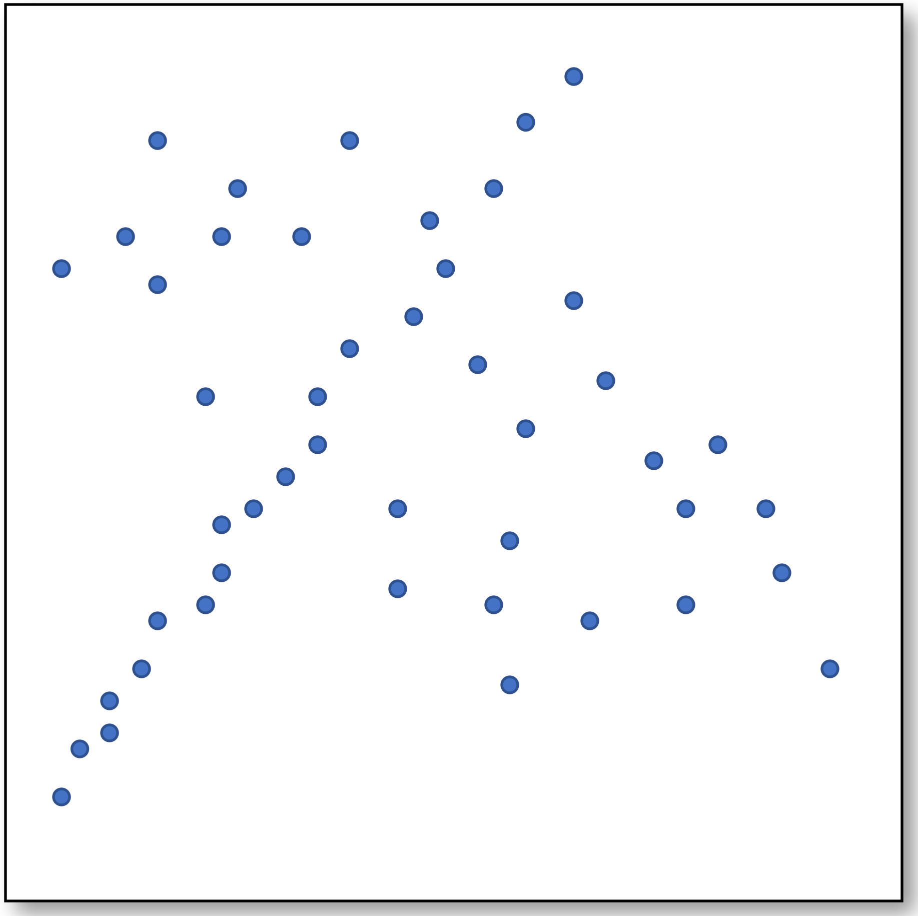
-*Leo Tolstoy,* Ana Karenina

Key idea

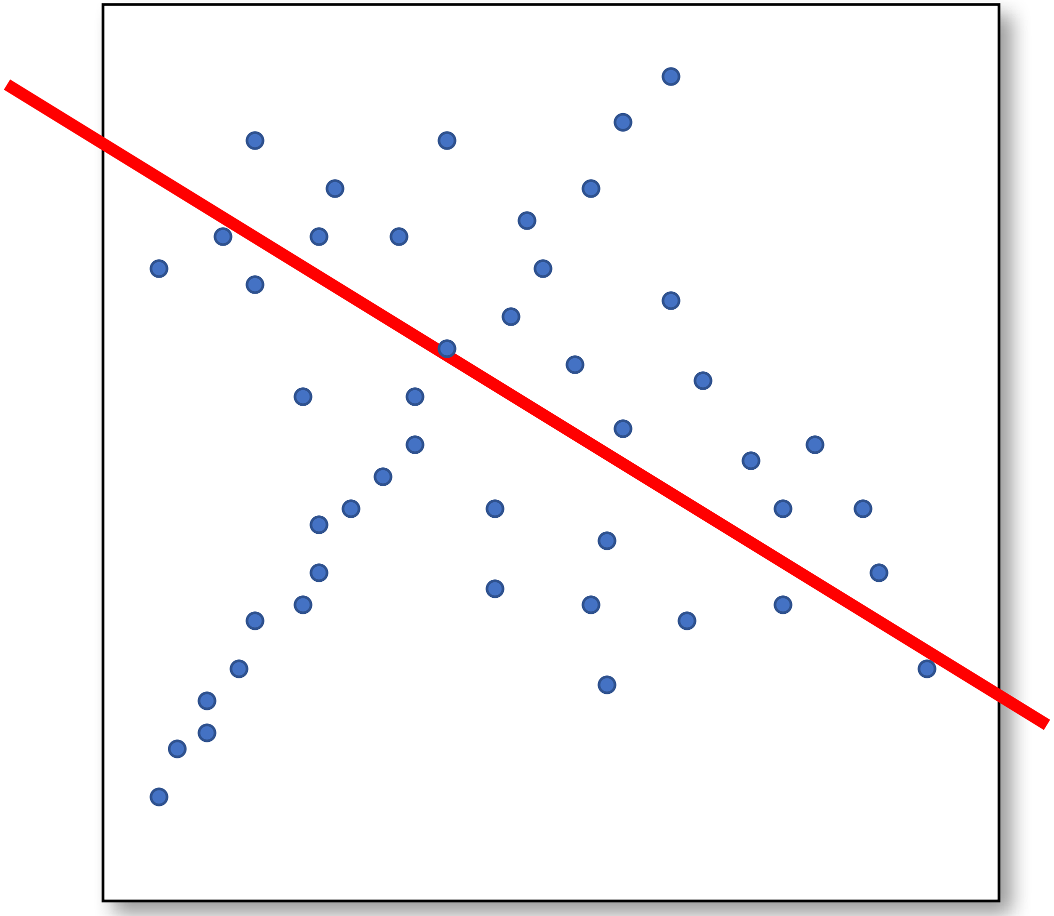
- Identify model that agrees with most points
- Given a hypothesized line
- Count the number of points that “agree” with the line
    - “Agree” = within a small distance of the line
    - I.e., the **inliers** to that line

- For all possible lines, select the one with the largest number of inliers

Counting inliers

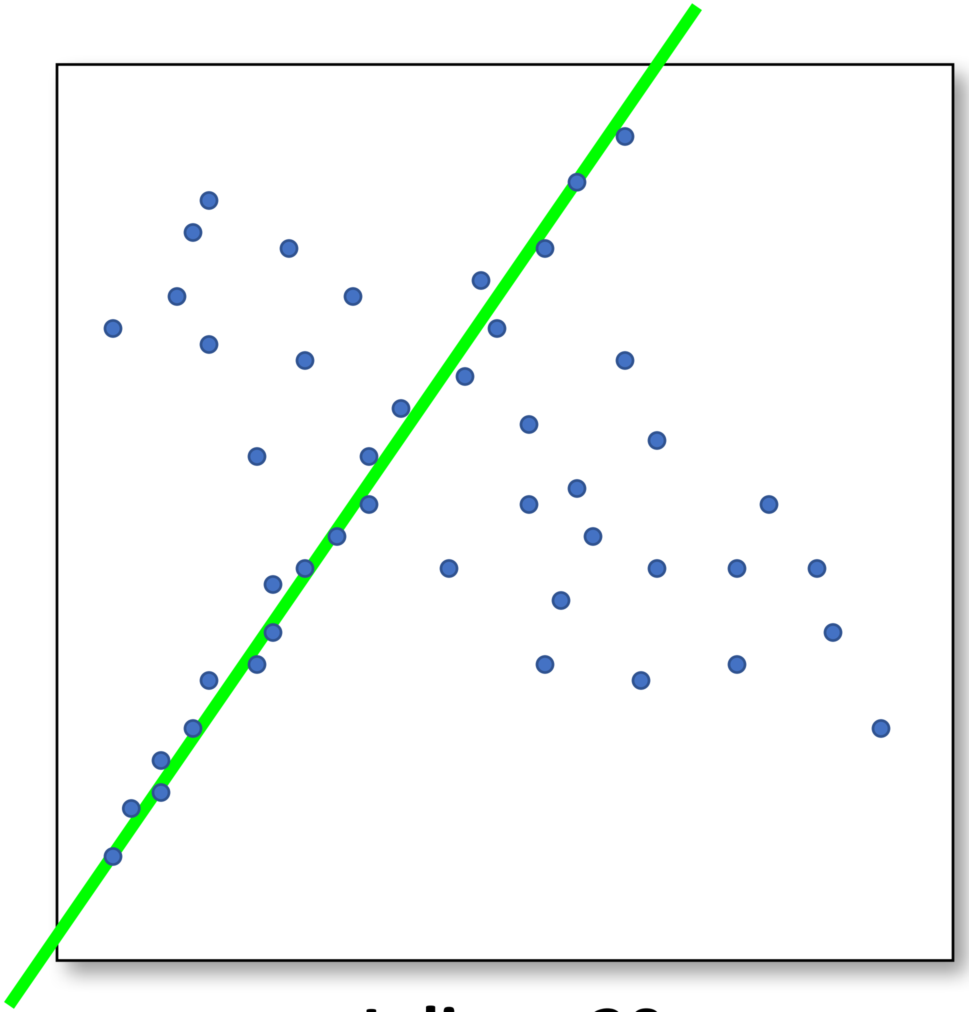


Counting inliers



Inliers: 3

Counting inliers



Inliers: 20

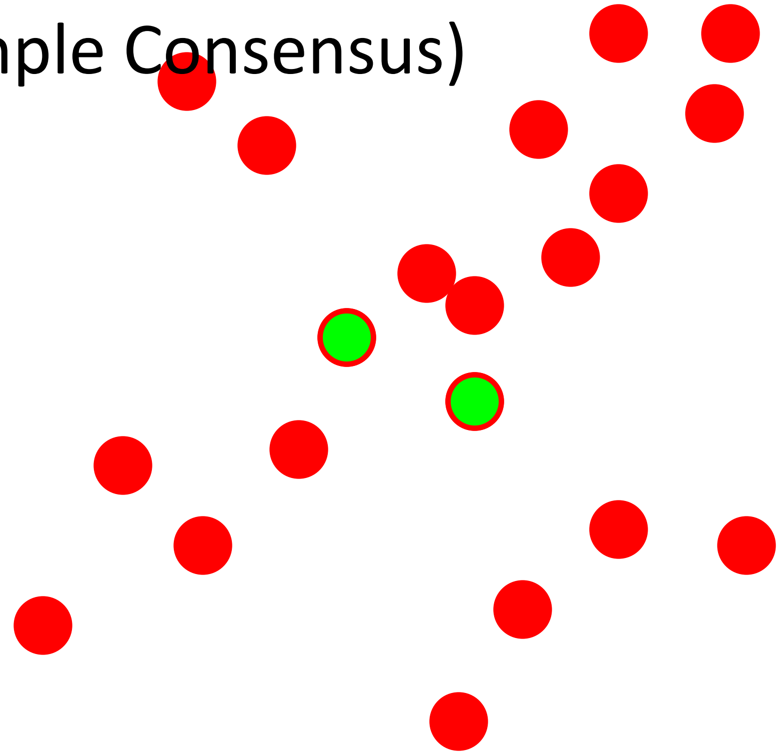
How do we find the best line?

- Unlike least-squares, no simple closed-form solution

- Hypothesize-and-test
  - Try out many lines, keep the best one
  - Which lines?

RANSAC (Random Sample Consensus)

Line fitting example



Algorithm:

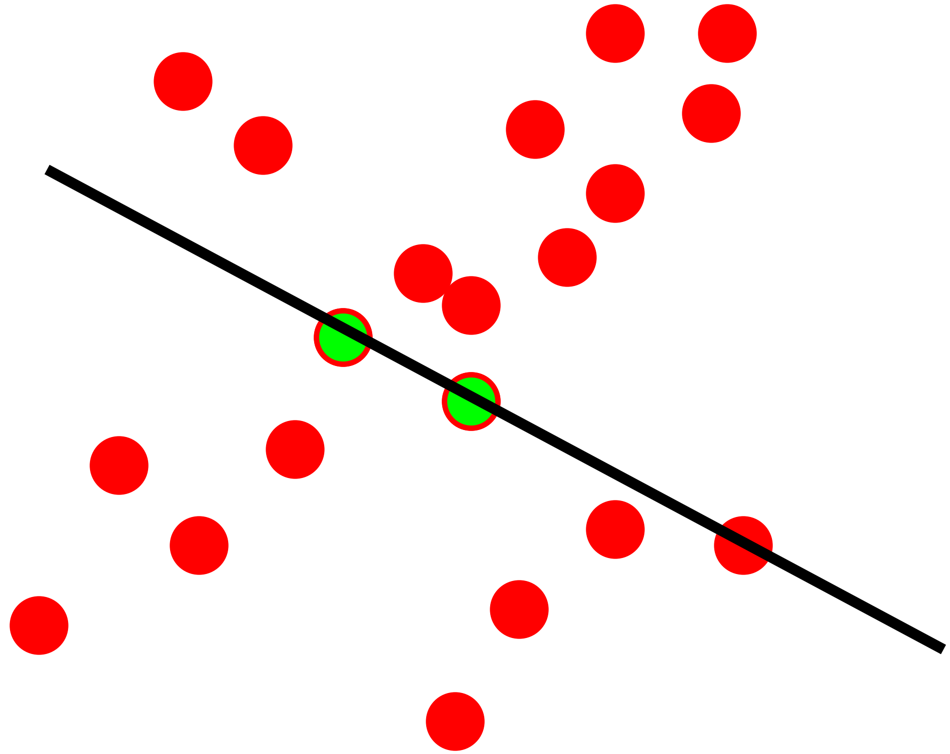
1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

Illustration by Savarese

RANSAC

Line fitting example



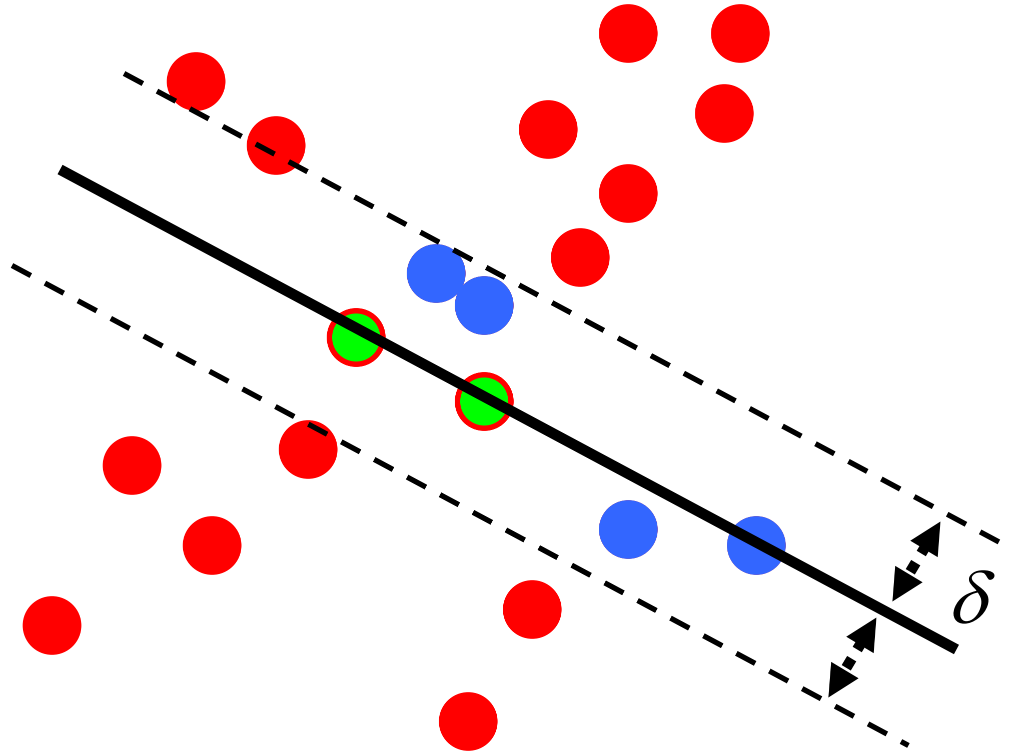
Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC

Line fitting example



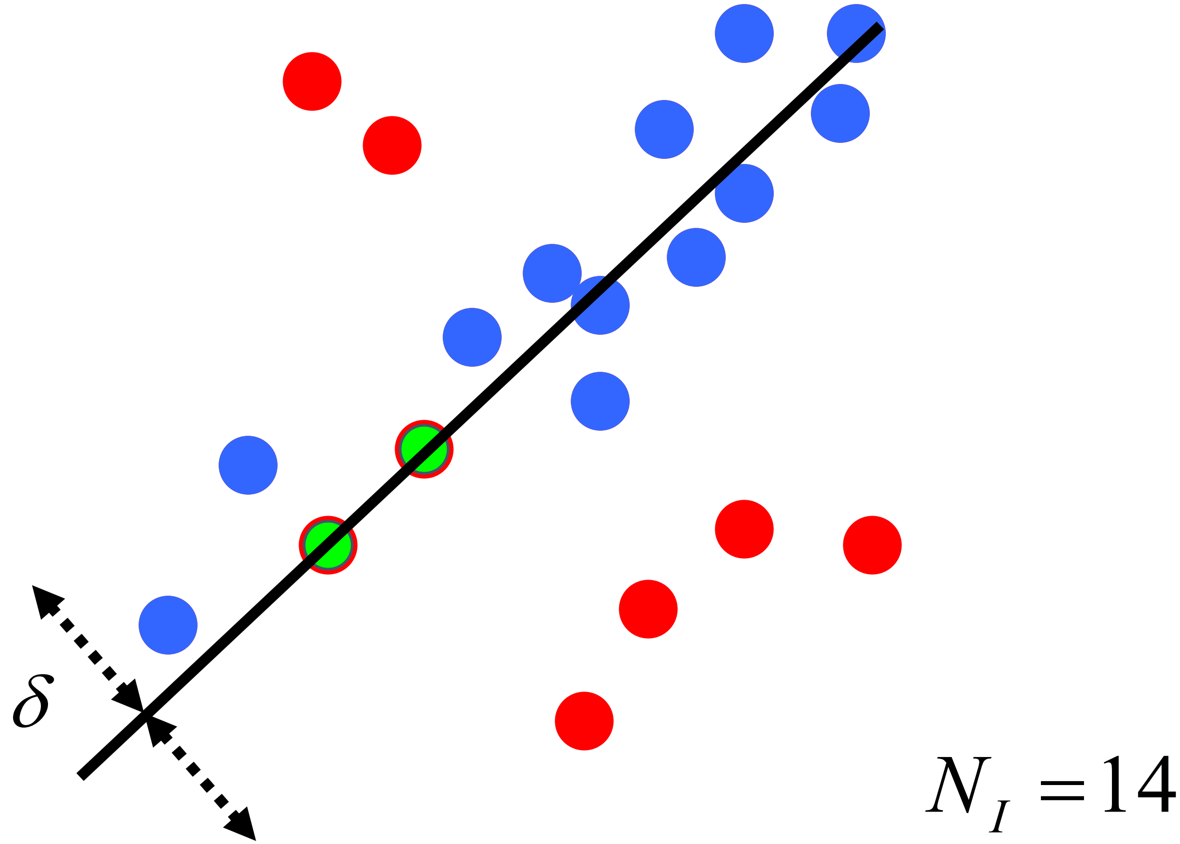
$$N_I = 6$$

Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC



Algorithm:

1. **Sample** (randomly) the number of points required to fit the model (#=2)
2. **Solve** for model parameters using samples
3. **Score** by the fraction of inliers within a preset threshold of the model

Repeat 1-3 until the best model is found with high confidence

RANSAC

- Idea:
  - All the inliers will agree with each other on the translation vector; the (hopefully small) number of outliers will (hopefully) disagree with each other
    - RANSAC only has guarantees if there are < 50% outliers

  - "All good matches are alike; every bad match is bad in its own way."

– Tolstoy via Alyosha Efros

RANSAC

- **Inlier threshold** related to the amount of noise we expect in inliers
  - Often model noise as Gaussian with some standard deviation (e.g., 3 pixels)
- **Number of rounds** related to the percentage of outliers we expect, and the probability of success we'd like to guarantee
  - Suppose there are 20% outliers, and we want to find the correct answer with 99% probability
  - How many rounds do we need?

How many rounds?

- If we have to choose *k* samples each time
  - with an inlier ratio p
  - and we want the right answer with probability *P*

	proportion of inliers *p*						
k	95%	90%	80%	75%	70%	60%	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

$P = 0.99$

Source: M. Pollefeys

To ensure that the random sampling has a good chance of finding a true set of inliers, a sufficient number of trials S must be tried. Let p be the probability that any given correspondence is valid and P be the total probability of success after S trials. The likelihood in one trial that all k random samples are inliers is p^k. Therefore, the likelihood that S such trials will all fail is

$$1 - P = (1 - p^k)^S \tag{6.29}$$

and the required minimum number of trials is

$$S = \frac{\log(1 - P)}{\log(1 - p^k)}. \tag{6.30}$$

	proportion of inliers *p*						
k	95%	90%	80%	75%	70%	60%	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

$P = 0.99$

RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Parameters to tune
  - Sometimes too many iterations are required
  - Can fail for extremely low inlier ratios

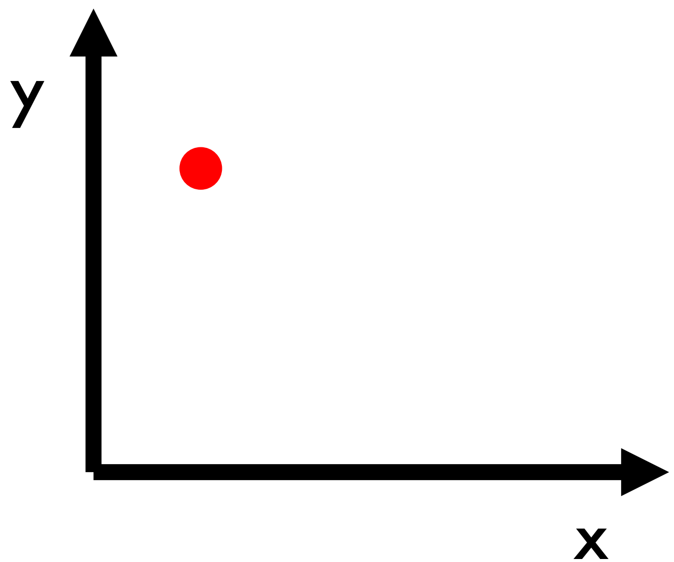
RANSAC

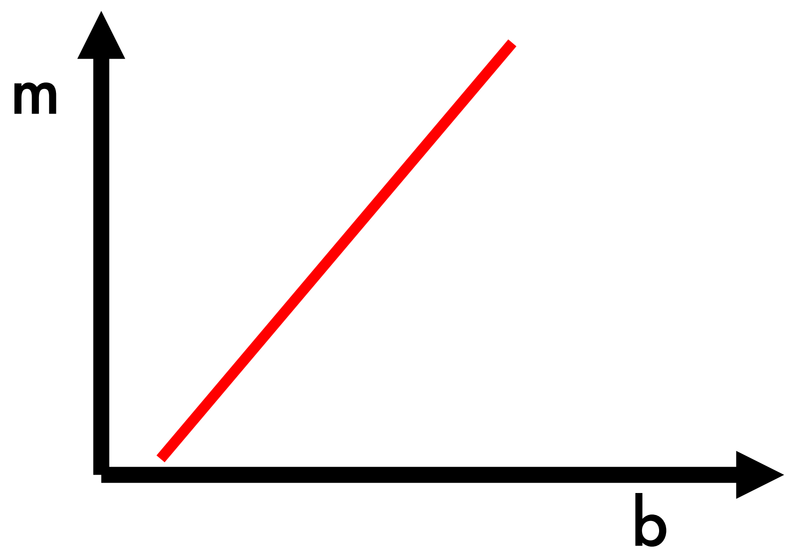
- An example of a “voting”-based fitting scheme
- Each hypothesis gets voted on by each data point, best hypothesis wins

- There are many other types of voting schemes
  - E.g., Hough transforms…

Hough transform

- What possible lines can this point lie on?
- (m,b) must satisfy: $y = mx + b$
  - This is the equation of a line in m and b

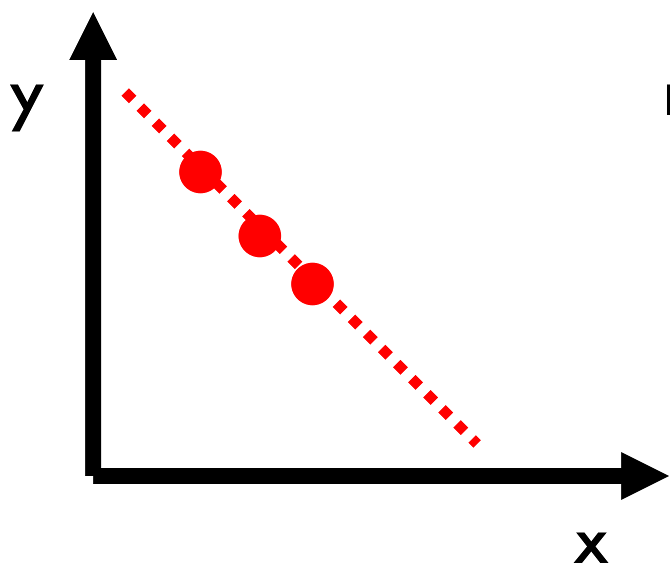


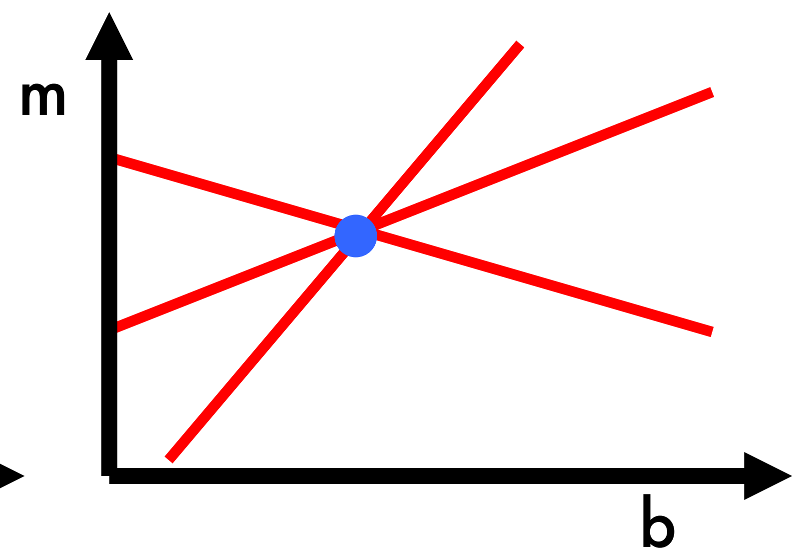


Hough transform

P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures,* Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

Given a set of points, find the curve or line that explains the data points best





Hough space

$$y = m\,x + b$$

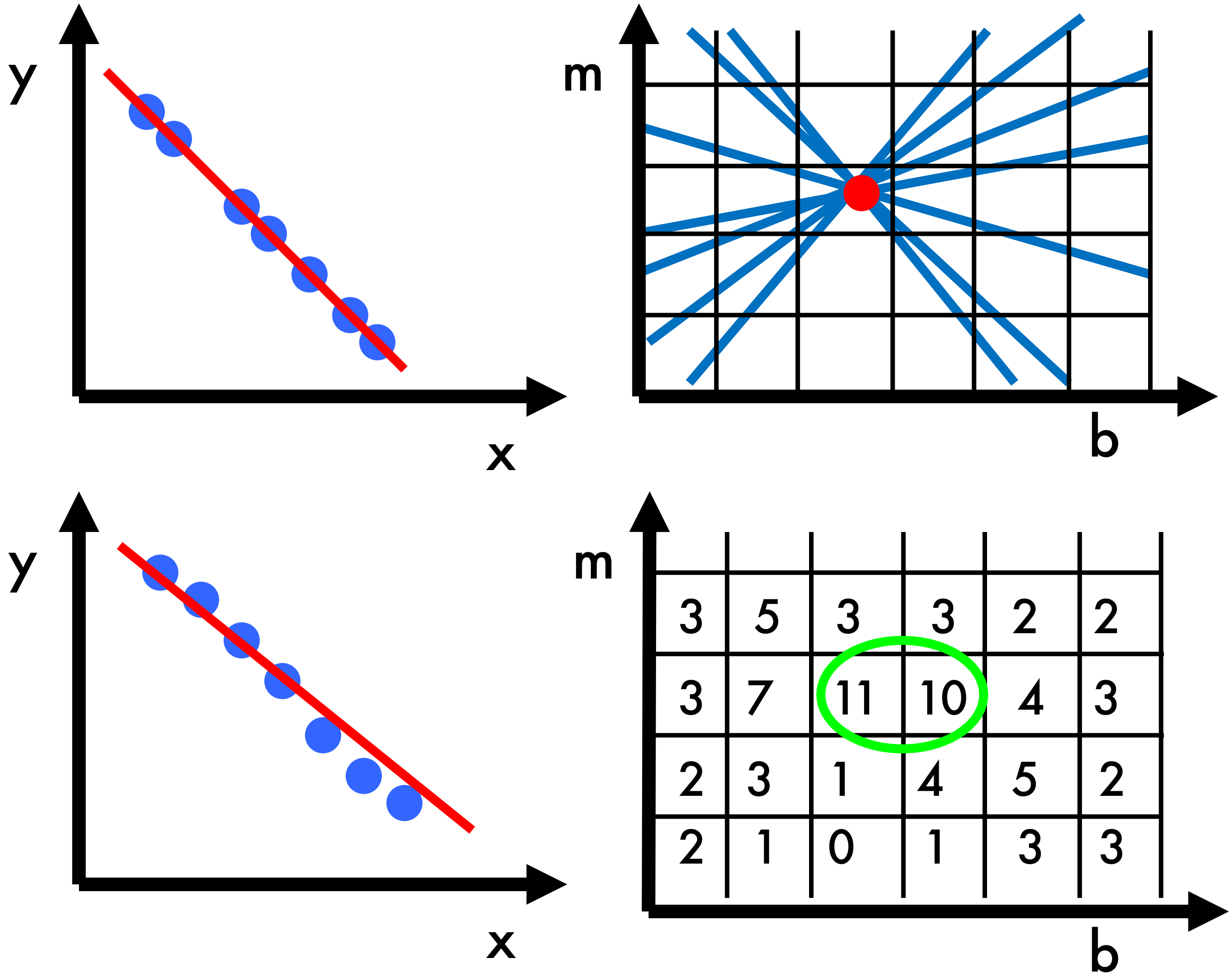
Hough Transform: Outline

1. Create a grid of parameter values

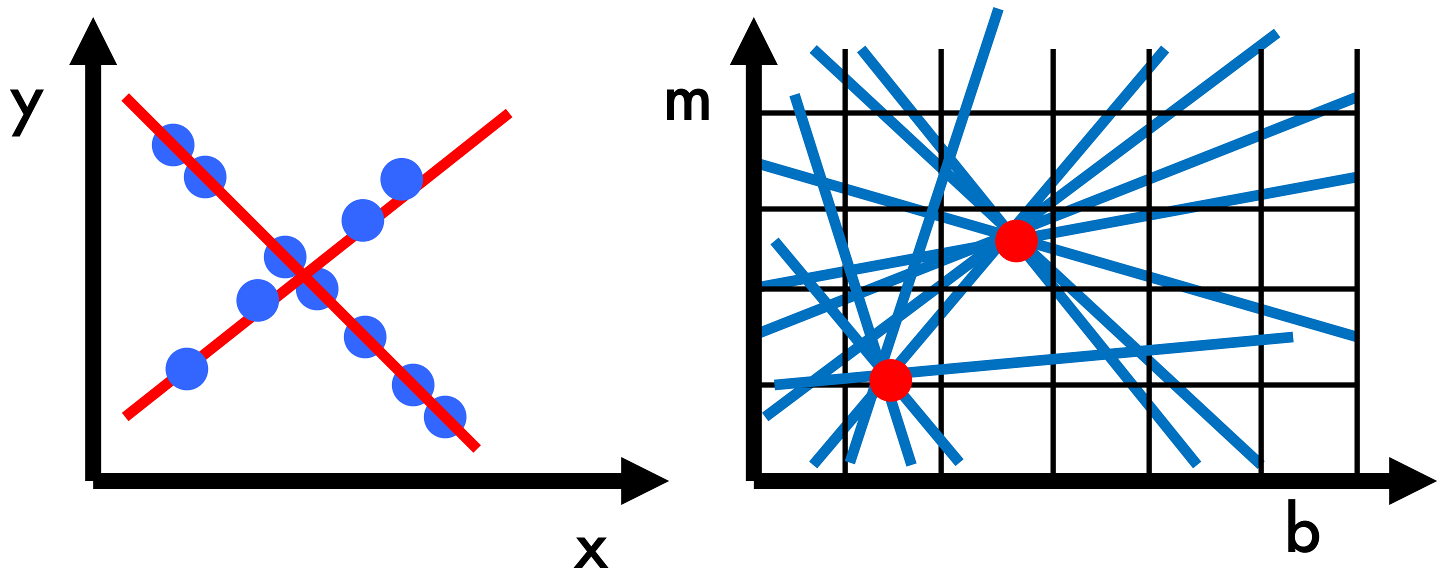
2. Each point votes for a set of parameters, incrementing those values in grid

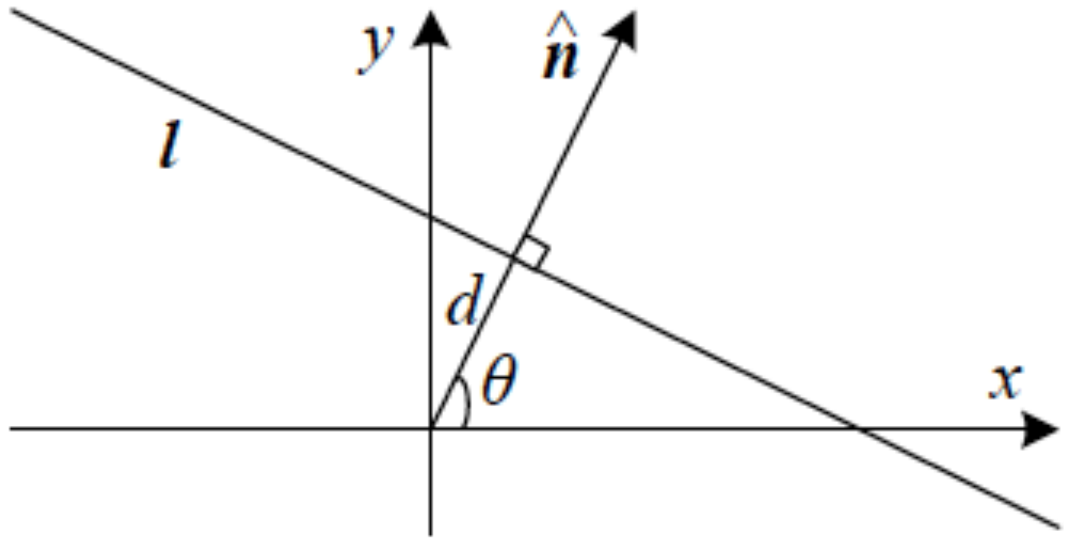
3. Find maximum or local maxima in grid

Hough transform



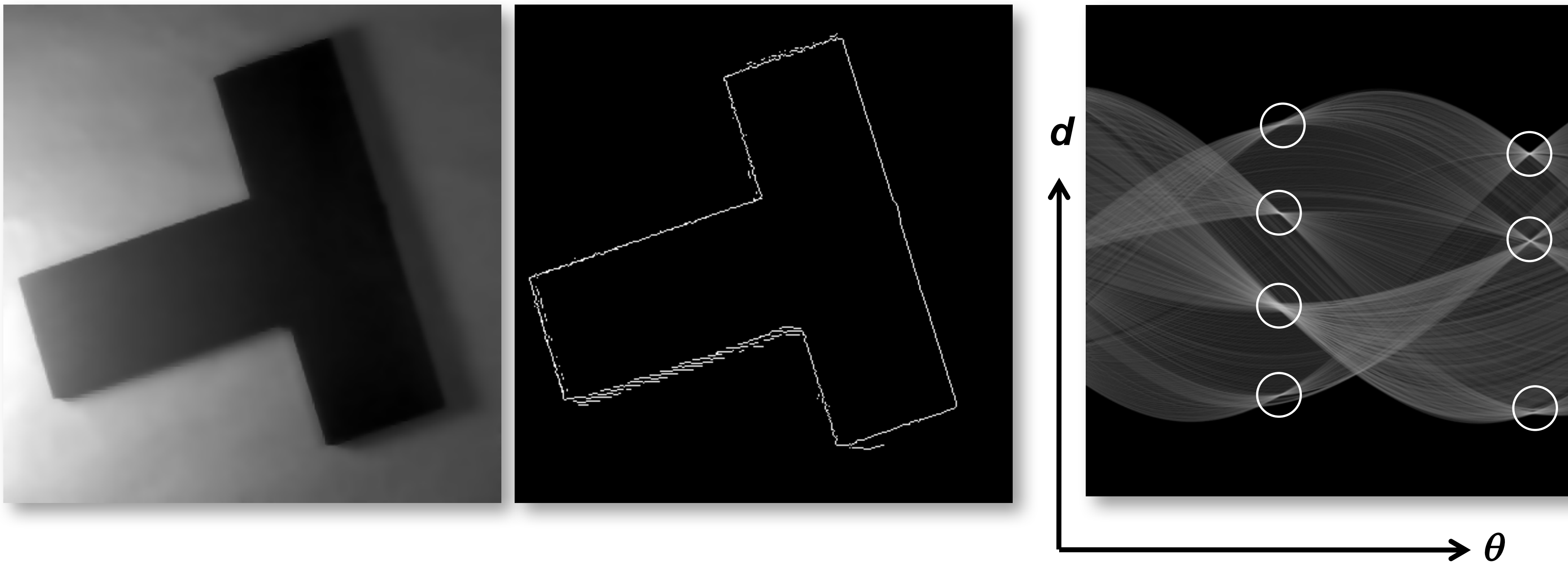
Hough transform





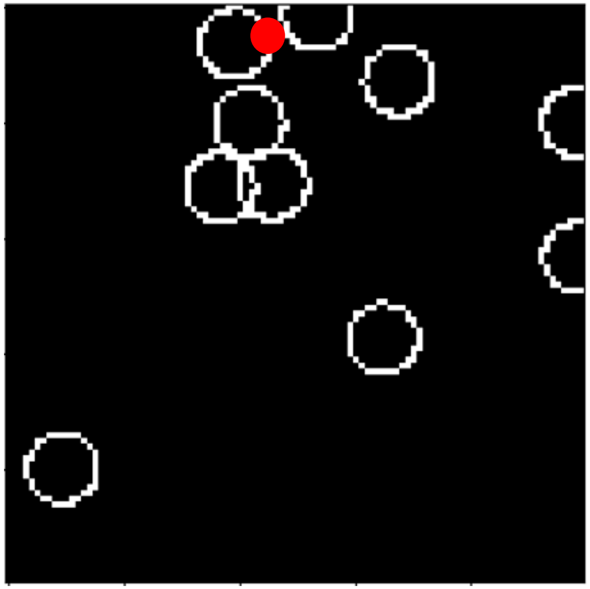
$$d = x cos\theta + y sin\theta$$

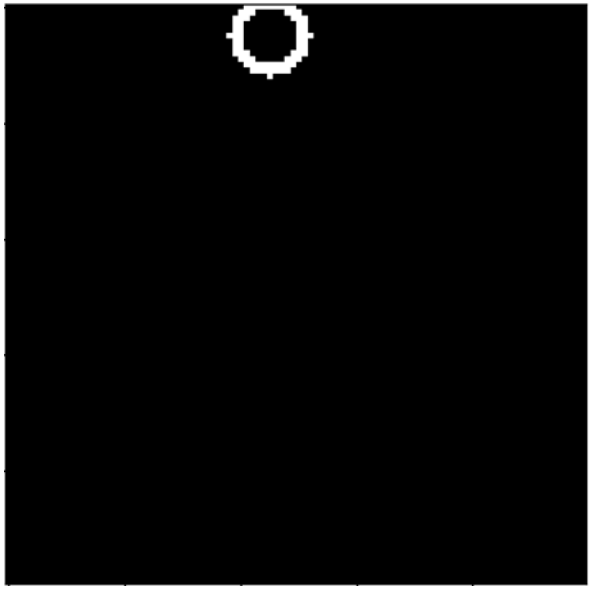
Hough transform



Hough transform : circles

- Suppose we want to fit circles to points
- Assume we know the radius, but don't know the centres

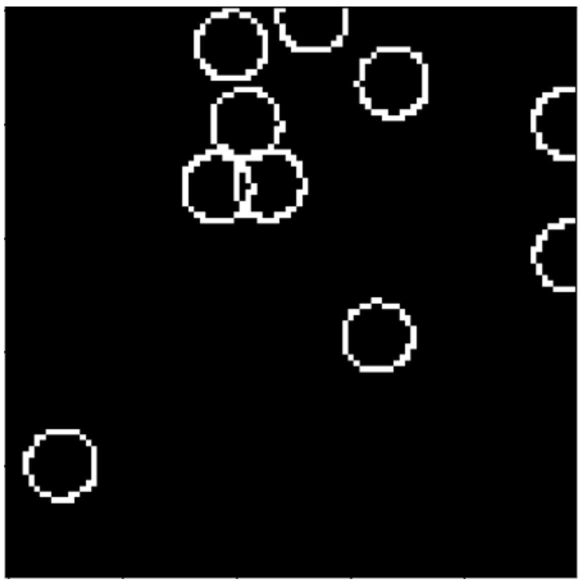


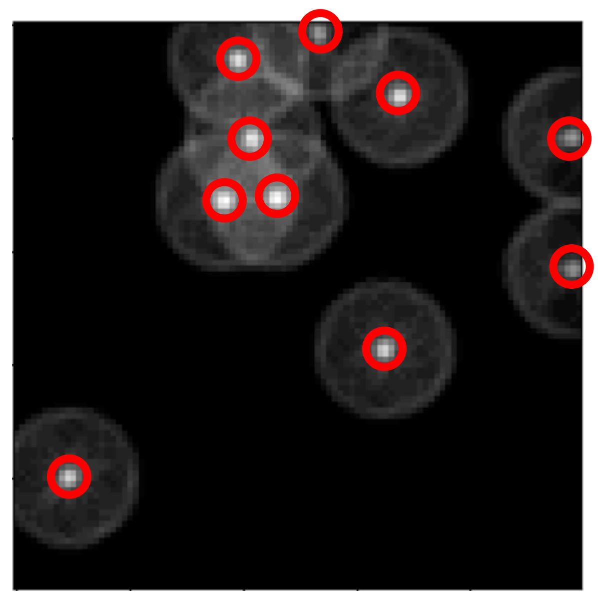


- Given a point (x,y), what possible circles of radius r can it lie on?
  - center (c_x, c_y) must satisfy $(x - c_x)^2 + (y-c_y)^2 = r^2$
  - center must lie on a circle!

Hough transform : circles

- Suppose we want to fit circles to points
- Assume we know the radius, but don't know the centres





- Given a point (x,y), what possible circles of radius r can it lie on?
  - center (c_x, c_y) must satisfy $(x - c_x)^2 + (y-c_y)^2 = r^2$
  - center must lie on a circle!

Hough transform: circles

- What happens if we don’t know the radius of the circle?
  - How big should the Hough space be?

Hough transform: general form

- Suppose we have to fit a model with some parameters to some data
- Construct a grid of parameter values
  - d parameters with k possible values for each: $\underbrace{k \times k \times \ldots \times k}_{\text{d times}}$ array
  - k^d elements
- Each data point puts in a vote for all sets of parameter values it likes
- Look for parameter settings with votes>threshold

Hough transform

- Simple
- Can deal with multiple correct answers
- Can only work if only a small number of parameters to fit (e.g. 2-3)
- Can we use this for homography fitting?