Lecture 5. Features and Fitting

RANSAC

Juan Carlos Niebles and Jiajun Wu

CS131 Computer Vision: Foundations and Applications

What will we learn today?

- A model fitting method for line detection
  - RANSAC

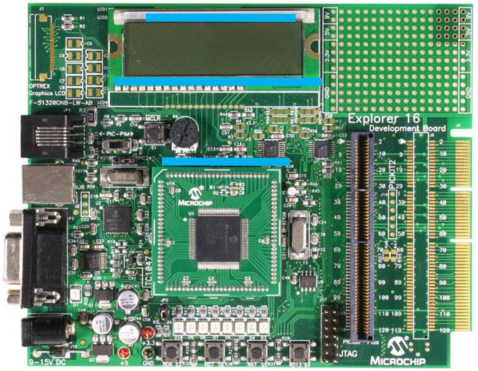
Fitting as Search in Parametric Space

- Let’s say we have chosen a parametric model for a set of features
  - For example, we have a line equation that we want to fit to a set of edge points
- We can ’search’ in parameter space by trying many potential parameter values and see which set of parameters ‘agree’/fit with our set of features
- Three main questions:
  - What model represents this set of features best?
  - Which of several model instances gets which feature?
  - How many model instances are there?
- Computational complexity is important
  - It is infeasible to examine every possible set of parameters and every possible combination of features

Source: L. Lazebnik

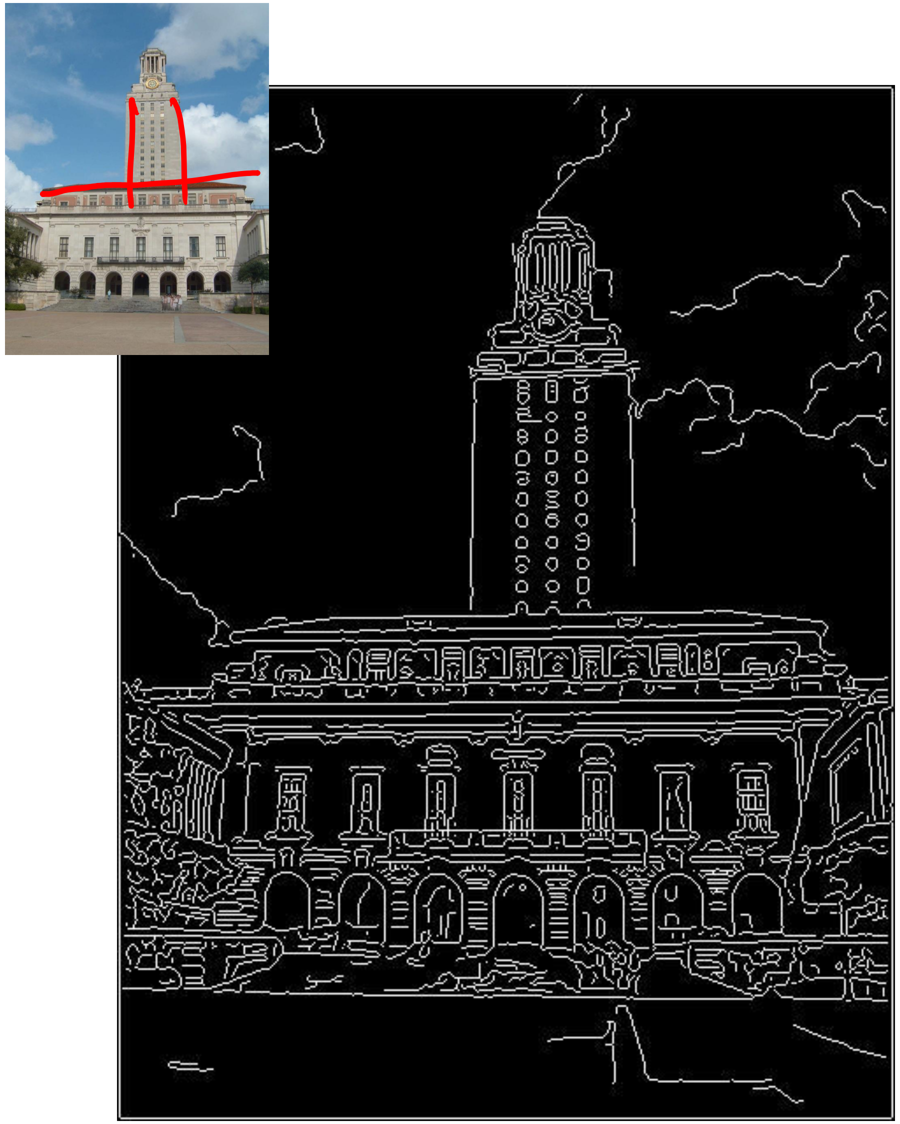
Example: Line Fitting

- Why fit lines? Many objects characterized by presence of straight lines

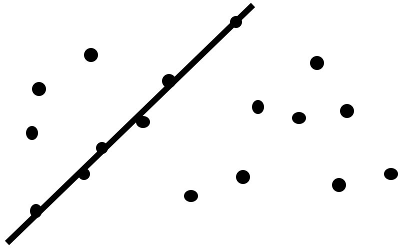


Slide credit: Kristen Grauman

Difficulty of Line Fitting



- Extra edge points (clutter), multiple models:
  - Which points go with which line, if any?



- Only some parts of each line detected, and some parts are missing:
  - How to find a line that bridges missing evidence?
- Noise in measured edge points, orientations:
  - How to detect true underlying parameters?

Slide credit: Kristen Grauman

Voting as a fitting technique

- It's not feasible to check all combinations of features by fitting a model to each possible subset. For example, the naïve line fitting we saw last time was $O(N^2)$.
- Voting is a general technique where we let the features vote for all models that are compatible with it.
  - Cycle through features, cast votes for model parameters.
  - Look for model parameters that receive a lot of votes.
- Noise & clutter features will cast votes too, *but* typically their votes should be inconsistent with the majority of "good" features.
- Ok if some features not observed, as model can span multiple fragments.

Slide credit: Kristen Grauman

RANSAC [Fischler & Bolles 1981]

- RANdom SAmple Consensus
- Approach: we want to avoid the impact of outliers, so let’s look for “inliers”, and use only those.
- Intuition: if an outlier is chosen to compute the current fit, then the resulting line won’t have much support from rest of the points.

Slide credit: Kristen Grauman

RANSAC [Fischler & Bolles 1981]

RANSAC loop:

Repeat for k iterations:

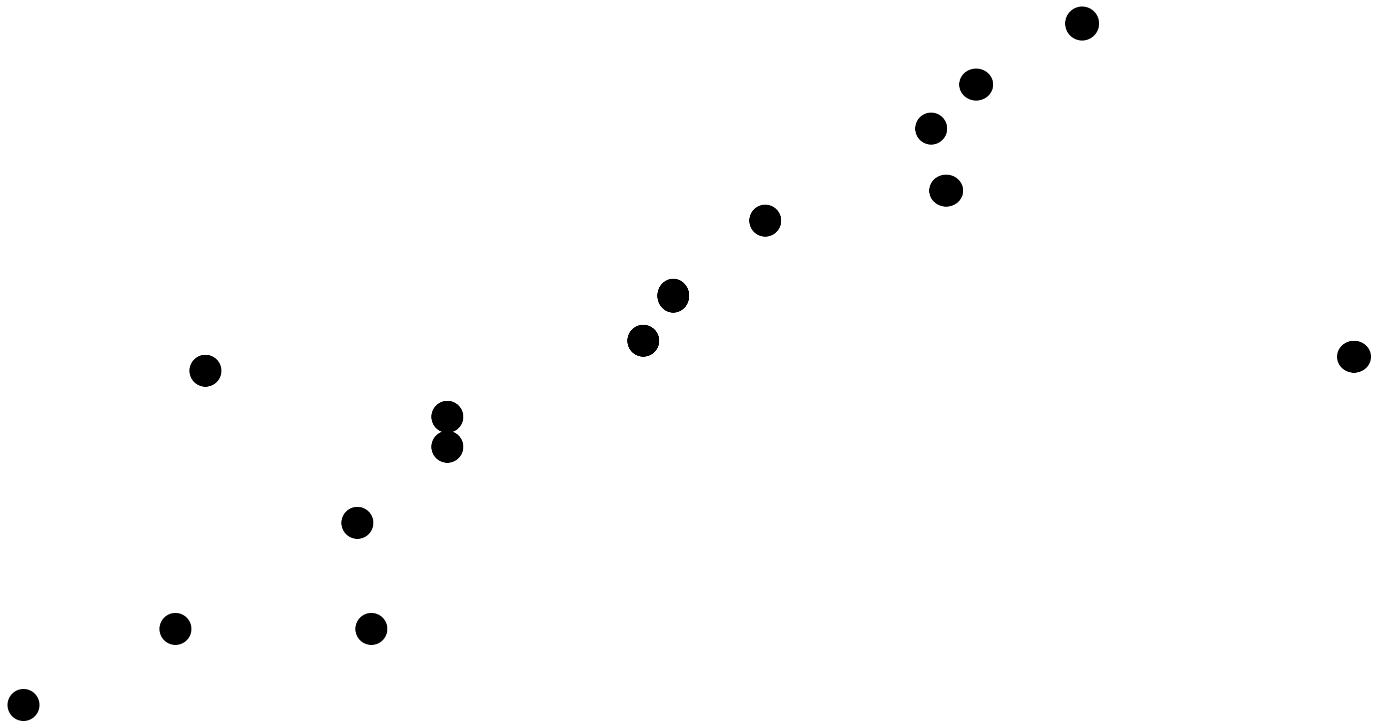
1. Randomly select a *seed group* of points on which to perform a model estimate (e.g., a group of edge points)
2. Compute model parameters from seed group
3. Find *inliers* to this model
4. If the number of inliers is sufficiently large, re-compute least-squares estimate of model on all of the inliers

– Keep the model with the largest number of inliers

Slide credit: Kristen Grauman

RANSAC Line Fitting Example

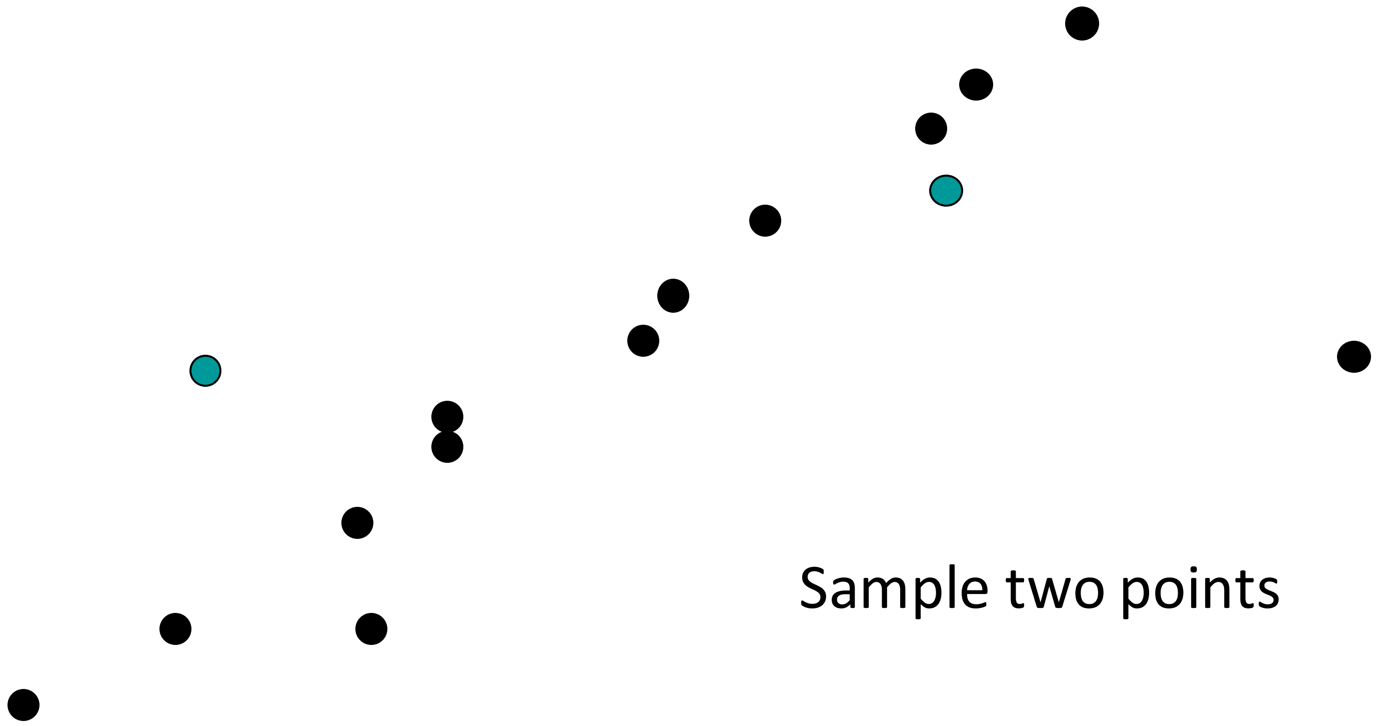
- Task: Estimate the best line
  - *How many points do we need to estimate the line?*



Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

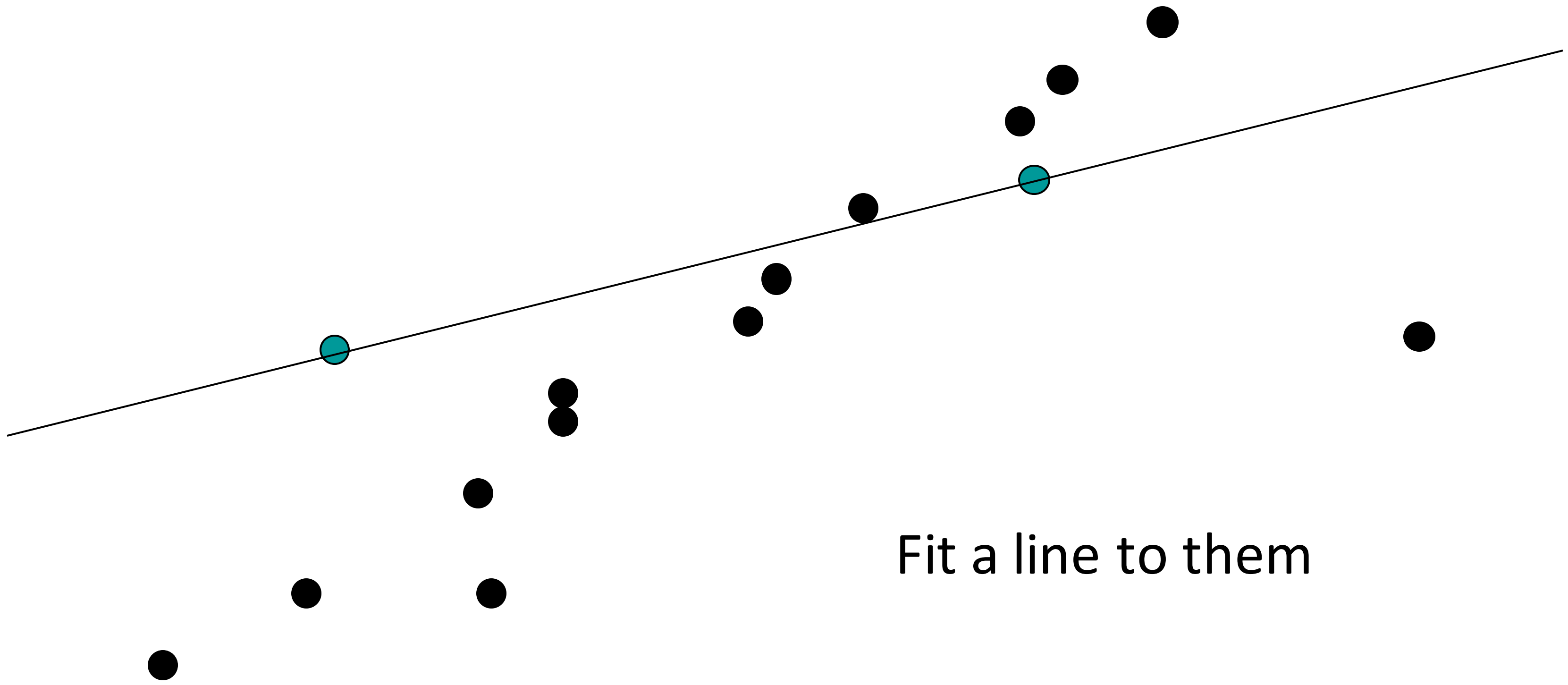


Sample two points

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

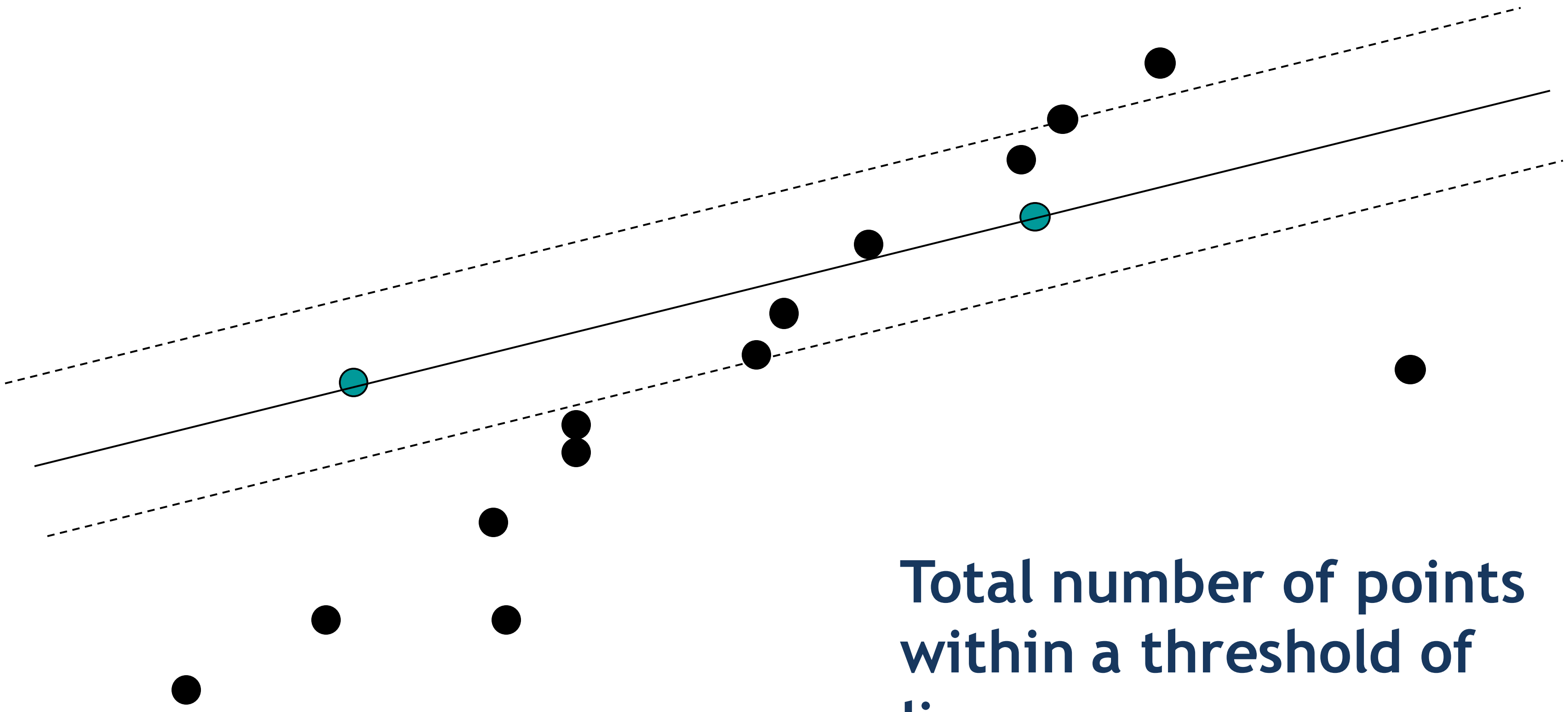


Fit a line to them

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

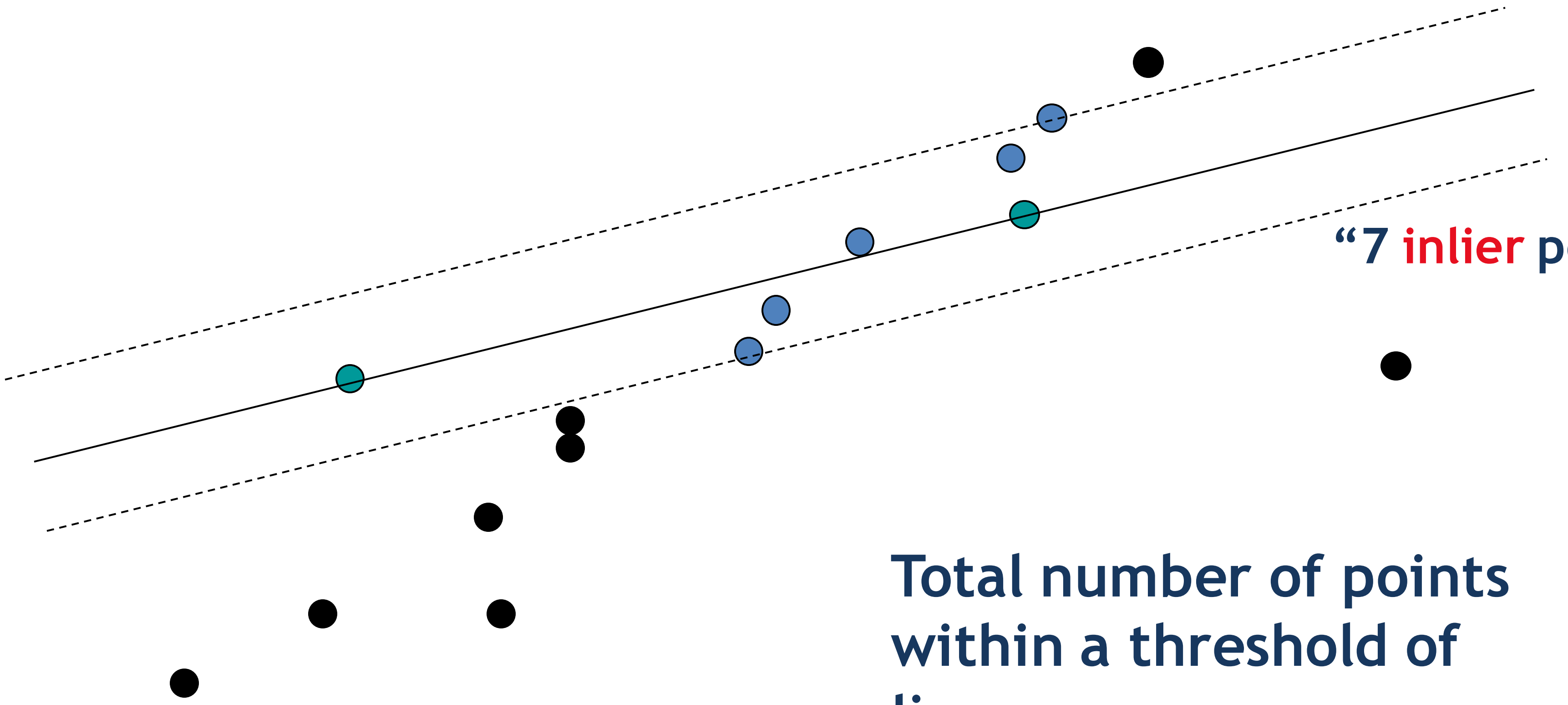


Total number of points within a threshold of line.

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line



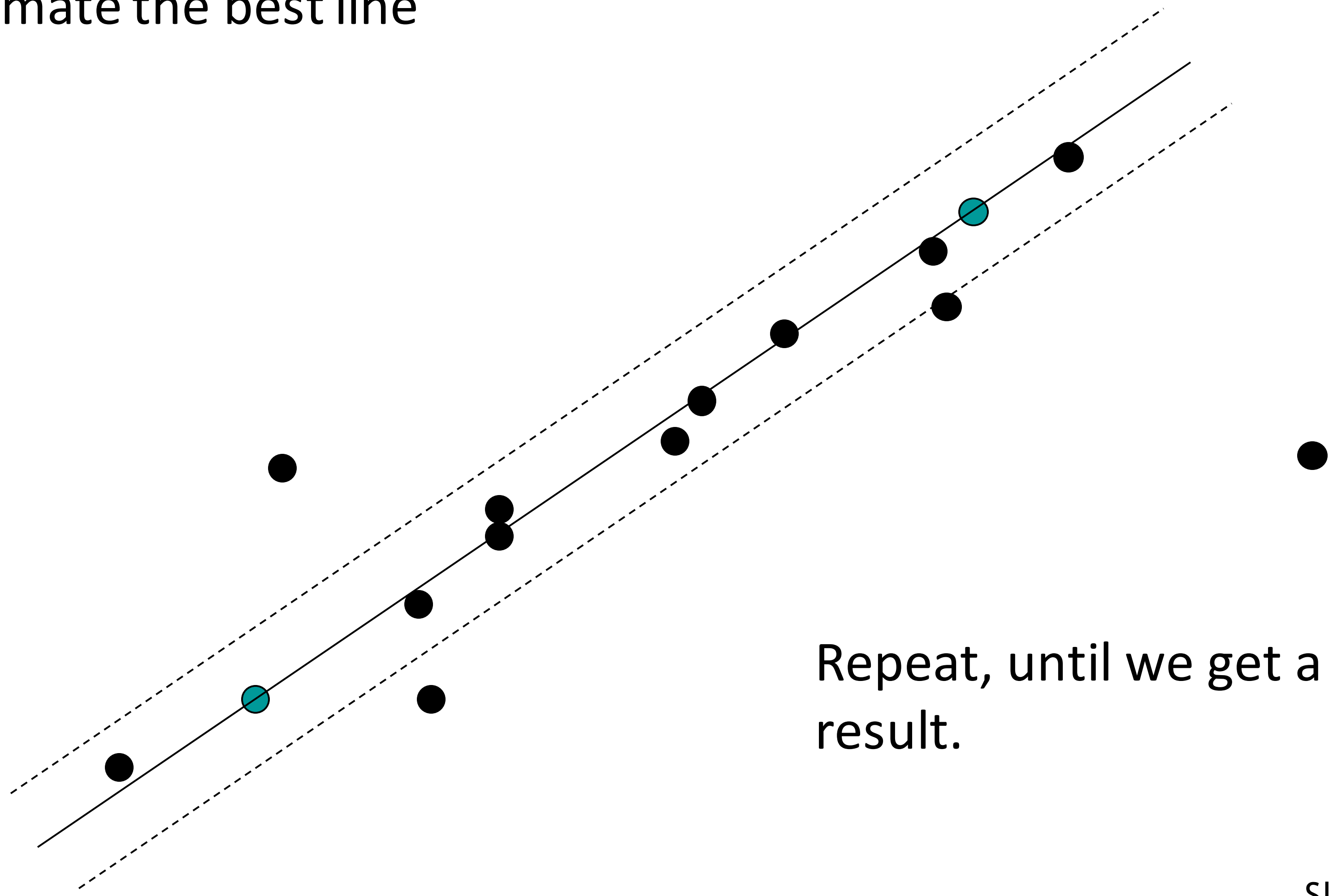
"7 **inlier** points"

Total number of points within a threshold of line.

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

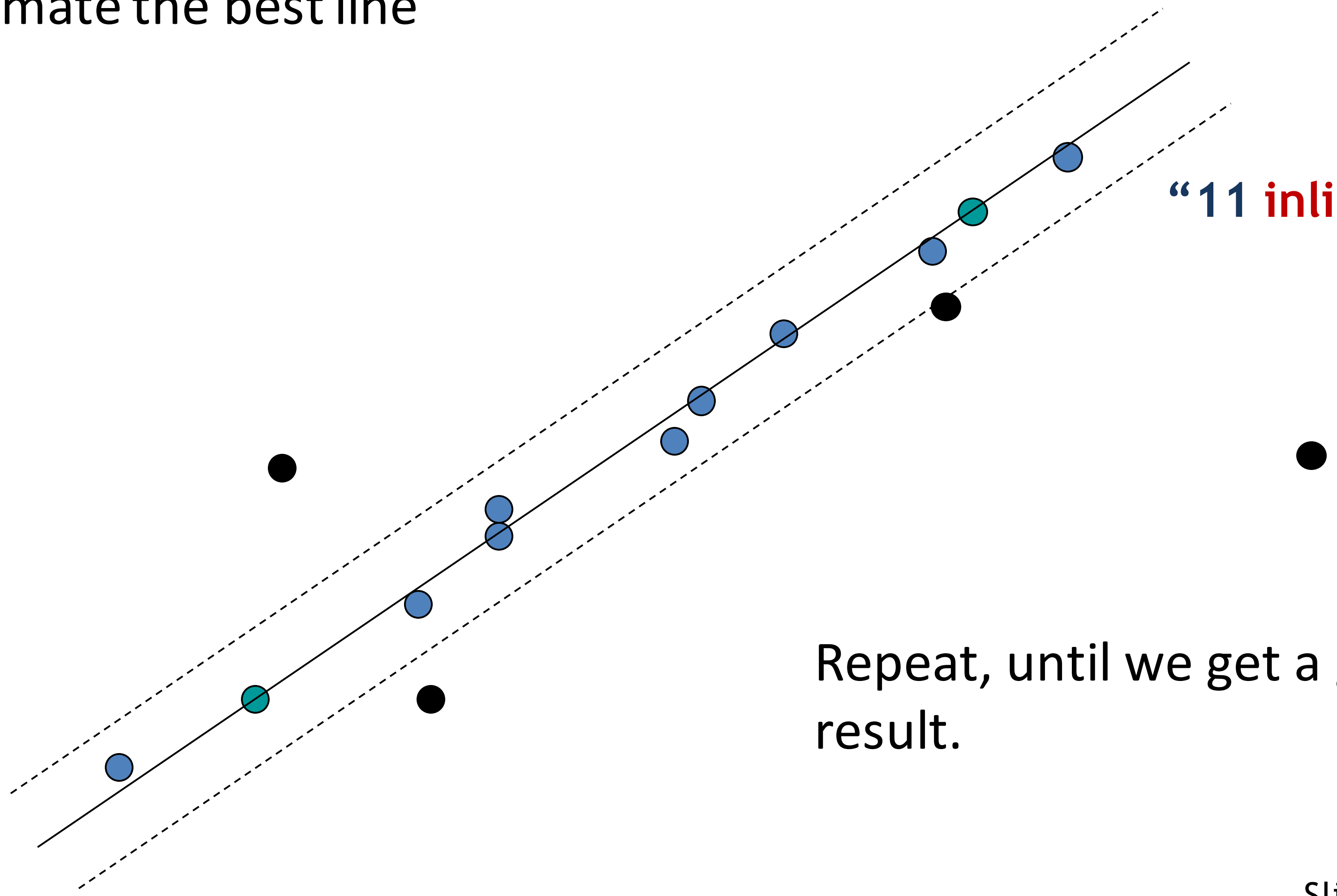


Repeat, until we get a good result.

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line



"11 **inlier** points"

Repeat, until we get a good result.

Slide credit: Jinxiang Chai

Algorithm 15.4: RANSAC: fitting lines using random sample consensus

Determine:
- n — the smallest number of points required
- k — the number of iterations required
- t — the threshold used to identify a point that fits well
- d — the number of nearby points required to assert a model fits well

Until k iterations have occurred
- Draw a sample of n points from the data uniformly and at random
- Fit to that set of n points
- For each data point outside the sample
  - Test the distance from the point to the line against t; if the distance from the point to the line is less than t, the point is close
- end
- If there are d or more points close to the line then there is a good fit. Refit the line using all these points.

end

Use the best fit from this collection, using the fitting error as a criterion

RANSAC: How many iterations “k”?

- How many samples are needed?
  - Suppose w is fraction of inliers (points from line).
  - n points needed to define hypothesis (2 for lines)
  - k samples chosen.
- Prob. that a single sample of n points is correct: w^n
- Prob. that a single sample of n points fails: $1 - w^n$
- Prob. that all k samples fail is: $(1 - w^n)^k$
- Prob. that at least one of the k samples is correct: $1 - (1 - w^n)^k$

$\Rightarrow$ Choose k high enough to keep this below desired failure rate.

Slide credit: David Lowe

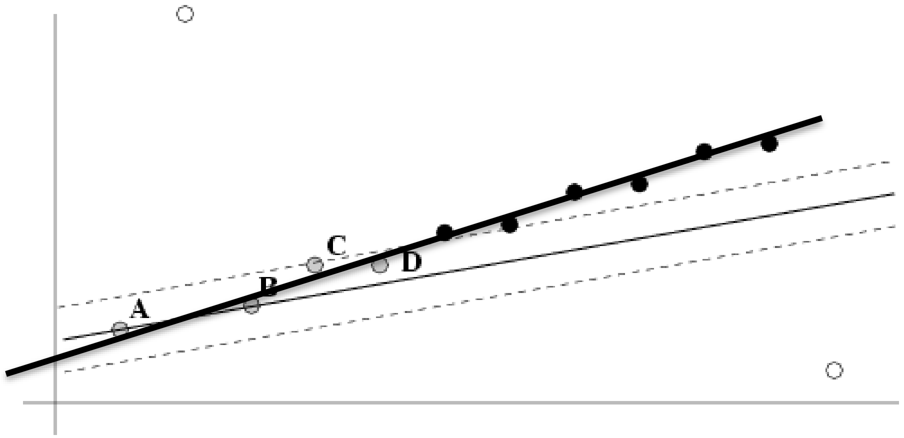
RANSAC: Computed k (p=0.99)

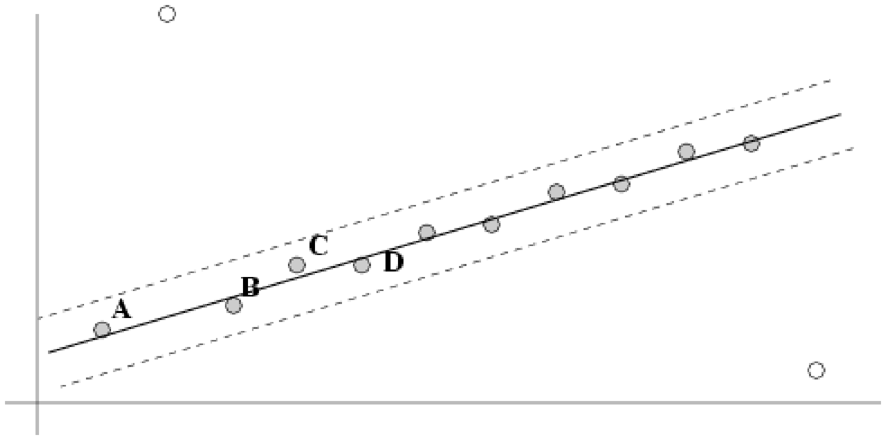
Sample size n	Proportion of outliers						
	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Slide credit: David Lowe

Refining RANSAC estimate

- RANSAC computes its best estimate from a minimal sample of n points, and divides all data points into inliers and outliers using this estimate.
- We can improve this initial estimate by estimation over all inliers (e.g. with standard least-squares minimization).
- But this may change inliers, so alternate fitting with re-classification as inlier/outlier.





Slide credit: David Lowe

RANSAC: Pros and 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)
- A voting strategy, The Hough transform, can handle high percentage of outliers

Summary

- RANSAC
  - Algorithm
  - Analysis
    - Number of samples
    - Pros and cons