COMP 408/508

Computer Vision
Fall 2017

RANSAC for Homography Estimation

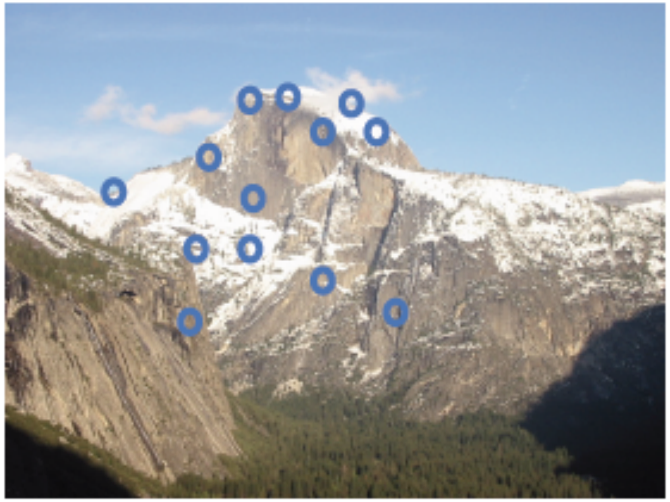
Mosaics: Stitching images together

Mosaics: Stitching images together

Algorithm:

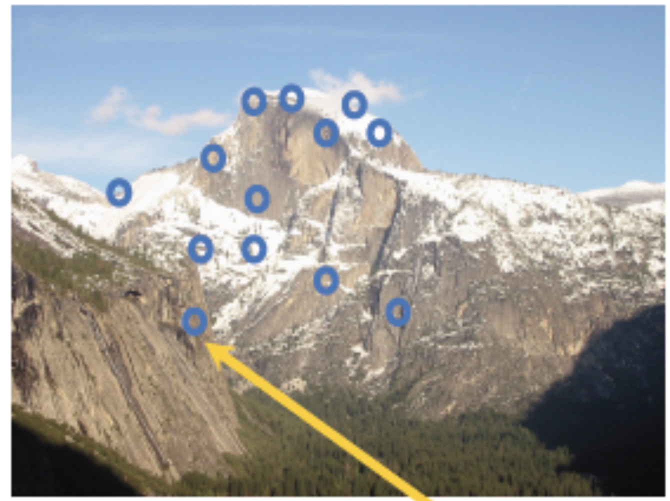
- Detect key points on images
- Build SIFT-like descriptors at key points
- Match SIFT-like descriptors
- Estimate homography transformation between images
    - Use RANSAC
- Warp images using homography transformation and stitch them together to create a mosaic

Detect key points



Slide credit: R. Krishna

Build SIFT descriptors

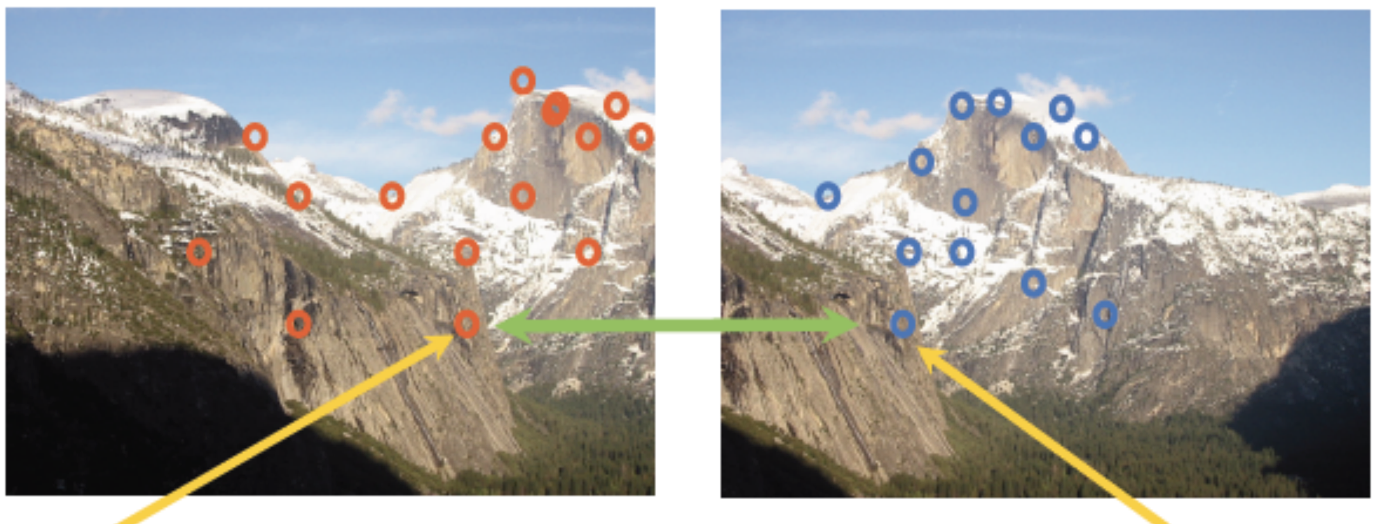


$$(u_1, u_2, \ldots, u_{128})$$

$$(v_1, v_2, \ldots, v_{128})$$

Slide credit: R. Krishna

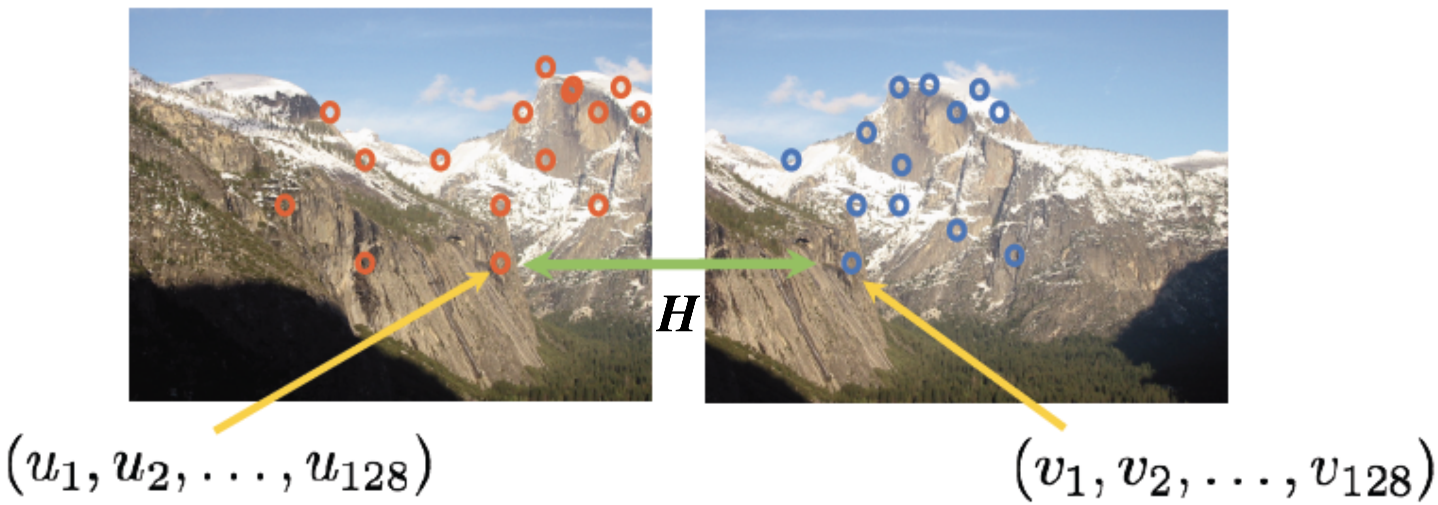
Match SIFT descriptors



$(u_1, u_2, \ldots, u_{128})$ $(v_1, v_2, \ldots, v_{128})$

Slide credit: R. Krishna

Find homography transformation H



H is a 3x3 transformation matrix that maps a pixel on one image to the corresponding pixel on the other

Slide credit: R. Krishna

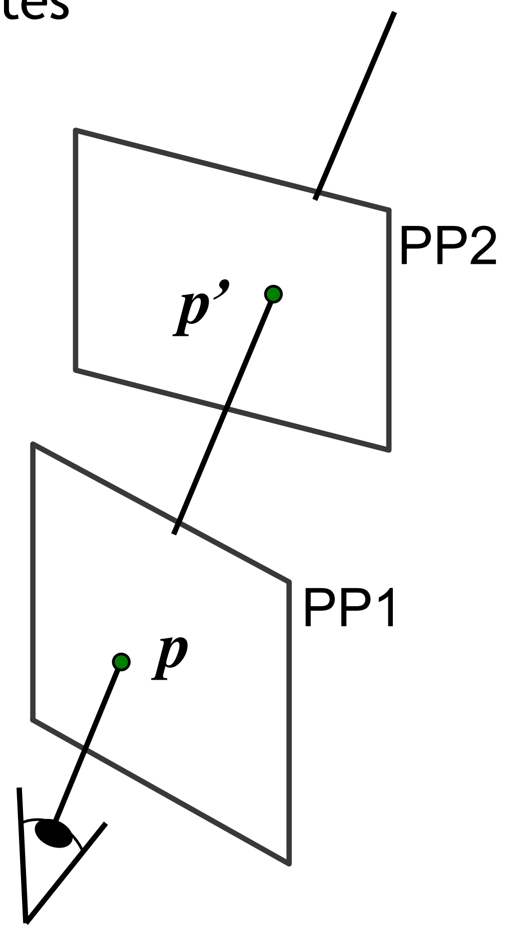
Warp images with H and stitch with blending

Slide credit: R. Krishna

What is homography H exactly?

- Projective mapping between any two projection planes with the same center of projection
- called Homography
- represented as 3x3 matrix in *homogenous* coordinates

$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\boldsymbol{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\boldsymbol{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\boldsymbol{p}}$$



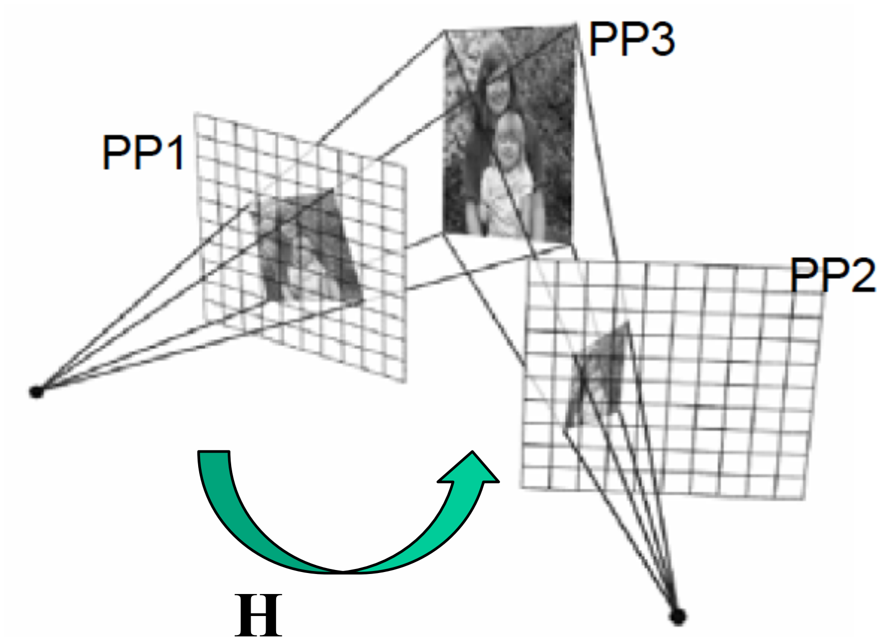
To apply a homography ***H***

- Compute $\boldsymbol{p'} = \boldsymbol{Hp}$ (regular matrix product)
- Convert $\boldsymbol{p'}$ from homogeneous to image coordinates (divide by w)

Homography between images

Homography $\boldsymbol{H}$ remains the same for all corresponding pixel pairs in either of the following cases:

- planar (or far away) scenes
- rotating cameras



How to estimate H?

- Estimate homography $\boldsymbol{H}$ from a set of given corresponding key points $\boldsymbol{p}_i$ and $\boldsymbol{p}_i'$
- Solve equations of the form: $w\boldsymbol{p}_i' = \boldsymbol{H}\boldsymbol{p}_i$
- Linear in unknowns: w and entries of $\boldsymbol{H}$
- $\boldsymbol{H}$ is defined up to an arbitrary scale factor
- At least 4 point matches are necessary
- Turns out to be a least squares estimation problem!

$$\begin{bmatrix} \mathrm{w}x_i' \\ \mathrm{w}y_i' \\ \mathrm{w} \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

RANSAC for estimating homography

RANSAC loop:

Select four feature pairs (at random)

Compute homography $\boldsymbol{H}$ (exact)

Compute *inliers* where $||\boldsymbol{p}_i' - \boldsymbol{H}\,\boldsymbol{p}_i|| < \varepsilon$

Keep largest set of inliers

Re-compute least-squares $\boldsymbol{H}$ estimate using all inliers

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:

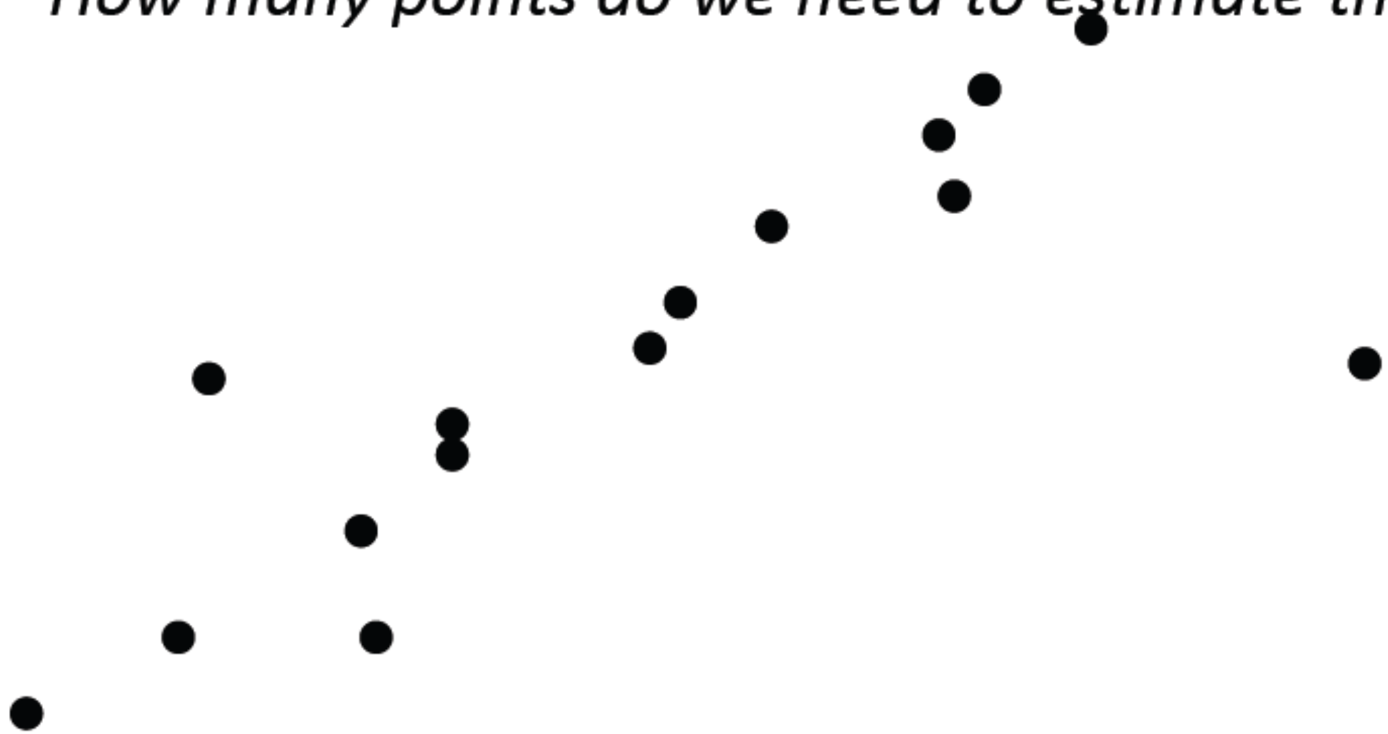
1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on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

- Keep the transformation with the largest number of inliers

Slide credit: Kristen Grauman

RANSAC Line Fitting Example

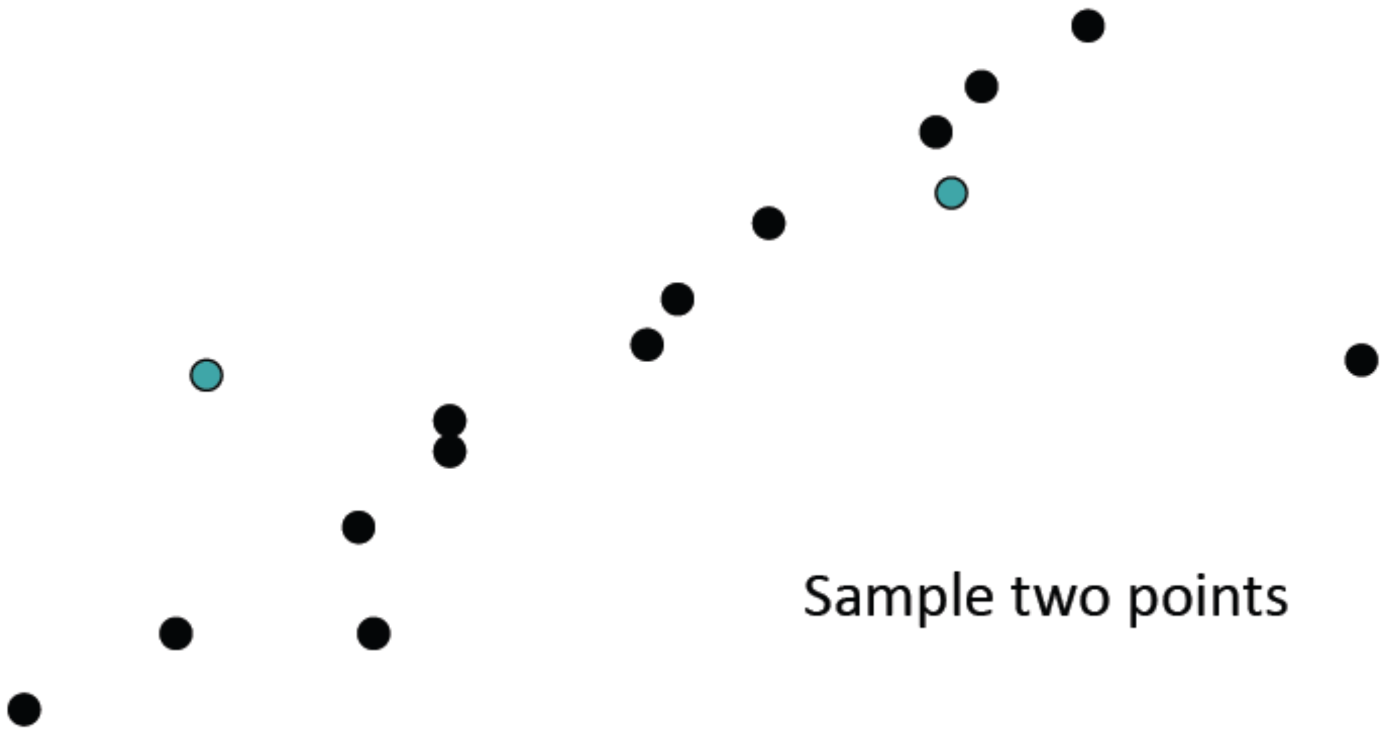
- Task: Estimate the best line
  - *How many points do we need to estimate the line?*



Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

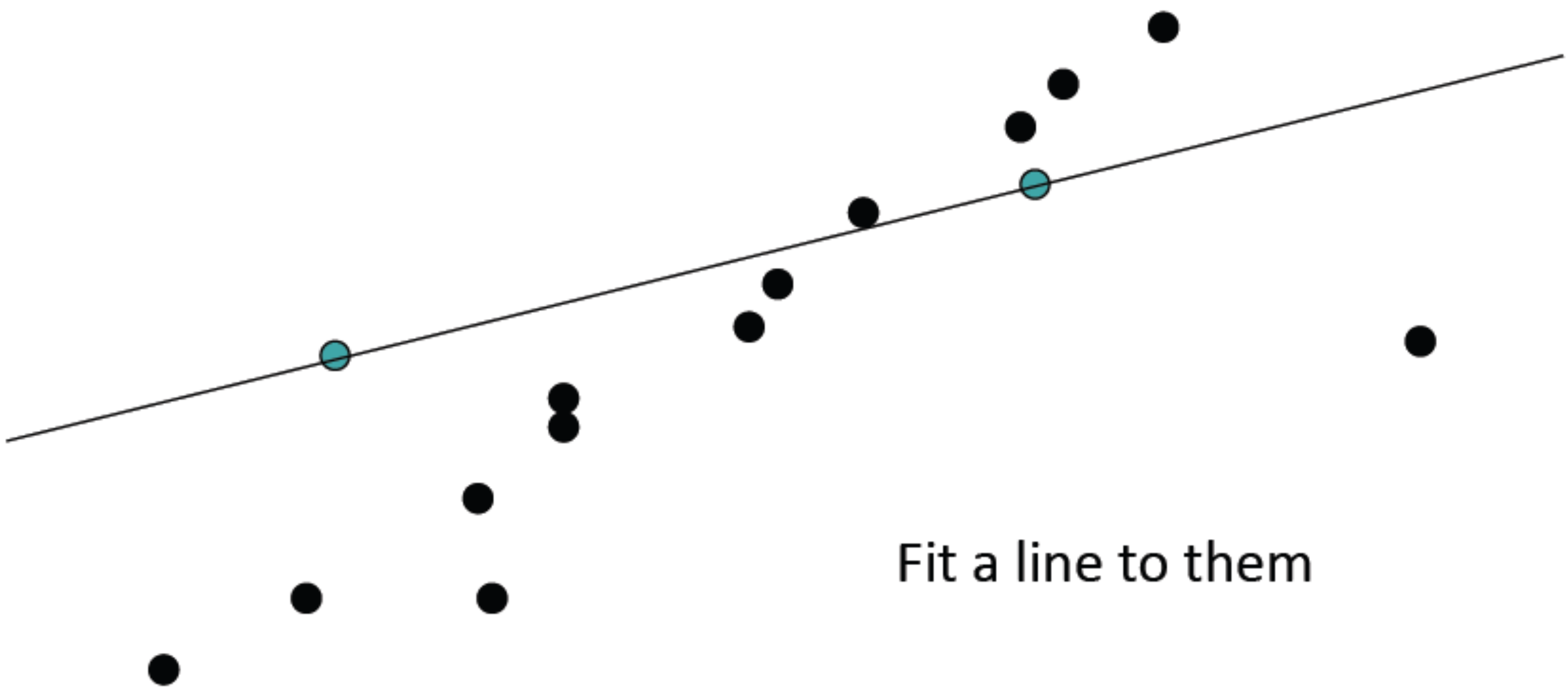


Sample two points

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

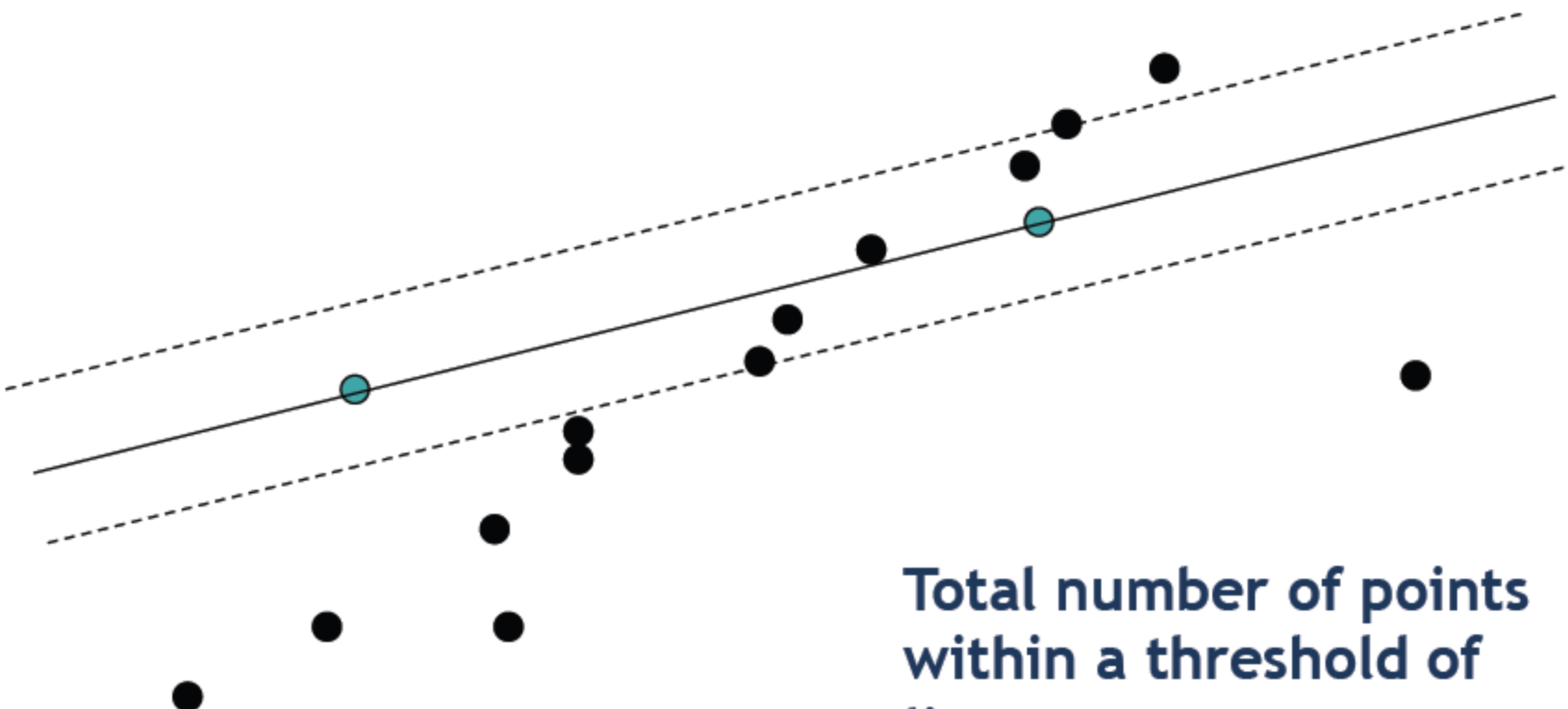


Fit a line to them

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

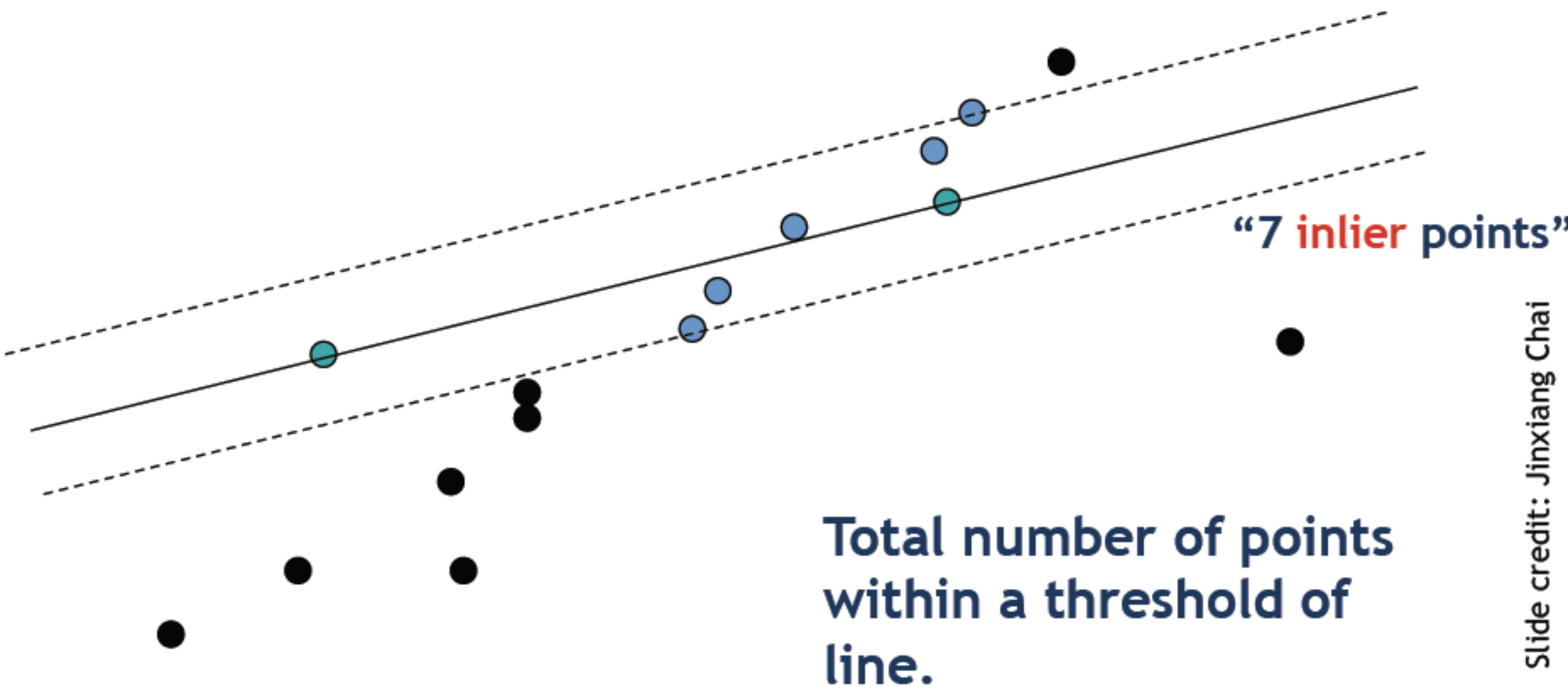


Total number of points within a threshold of line.

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

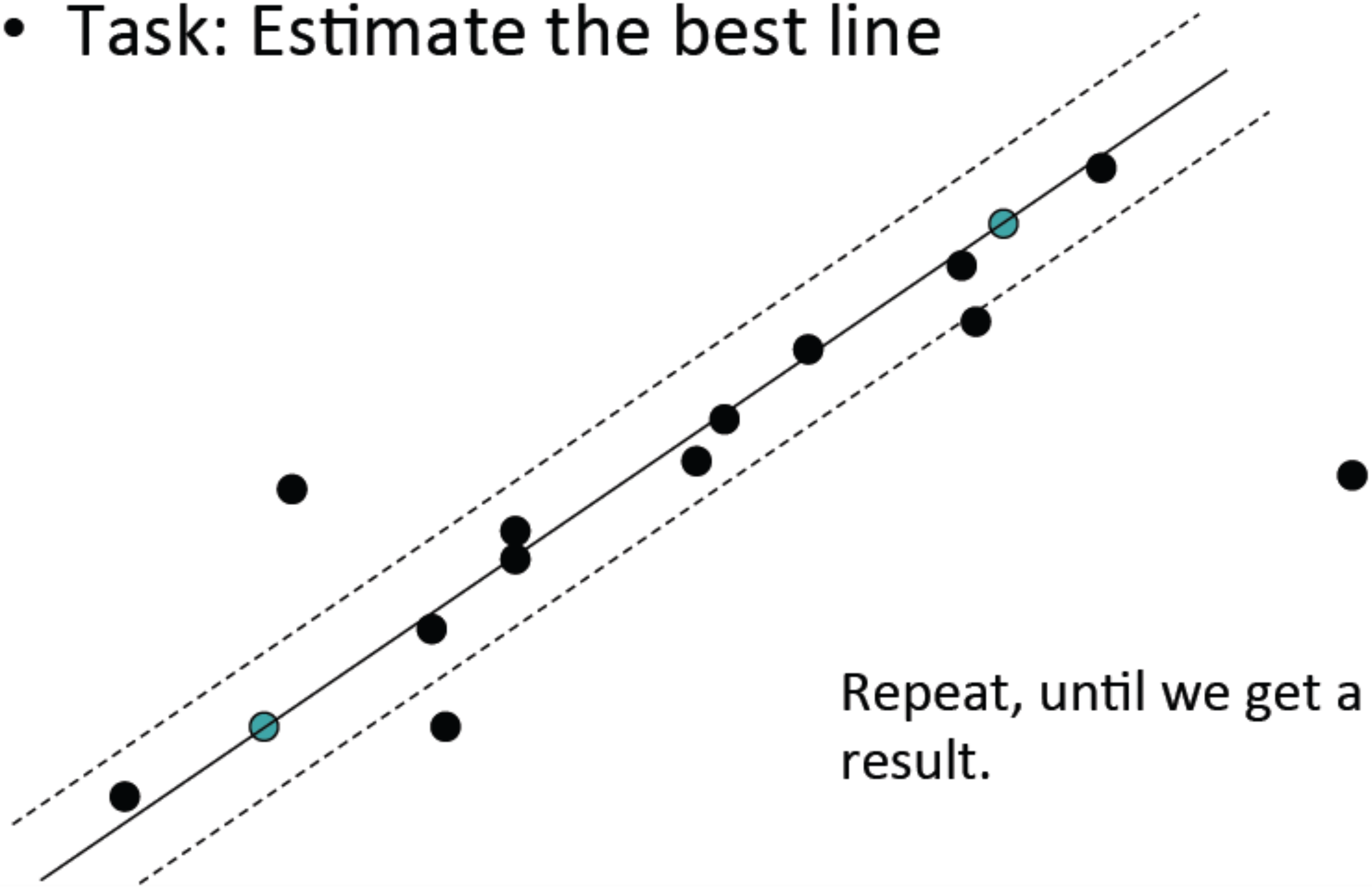
- Task: Estimate the best line



Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line

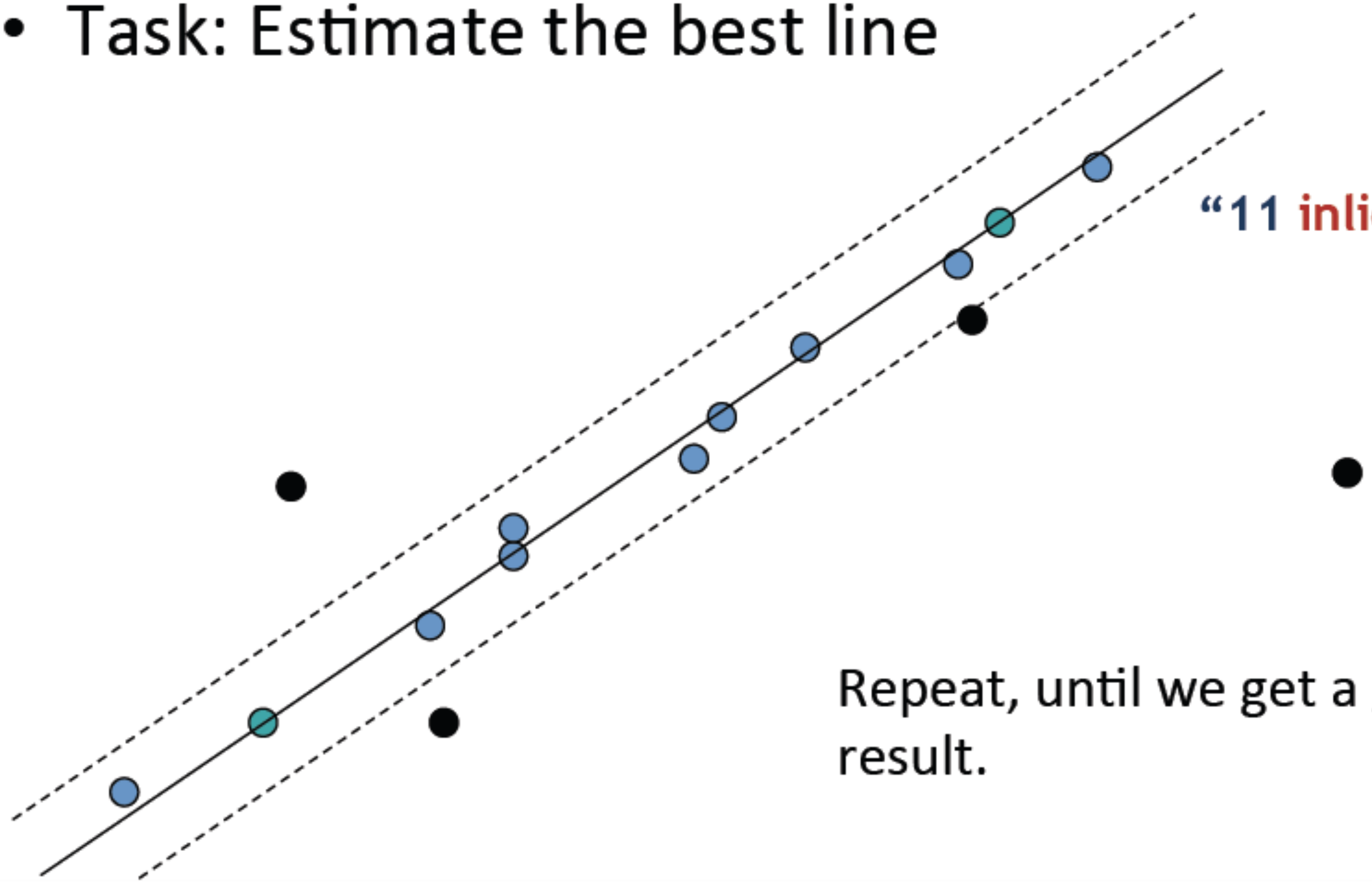


Repeat, until we get a good result.

Slide credit: Jinxiang Chai

RANSAC Line Fitting Example

- Task: Estimate the best line



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 samples?

- 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 all k samples fail is: $(1-w^n)^k$

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

Slide credit: David Lowe

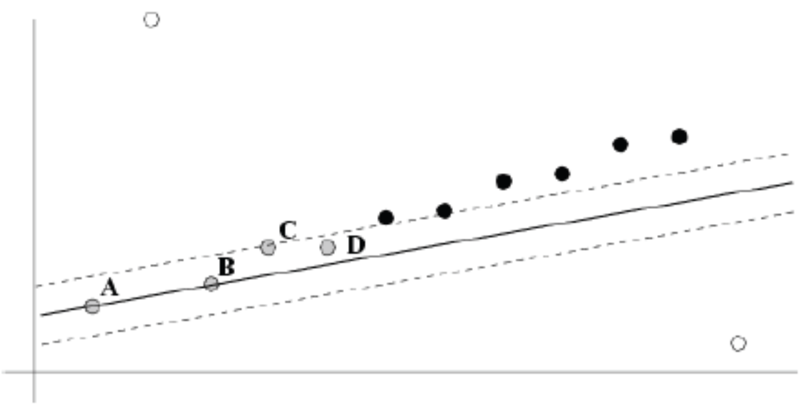
RANSAC: Computed k (p=0.99)

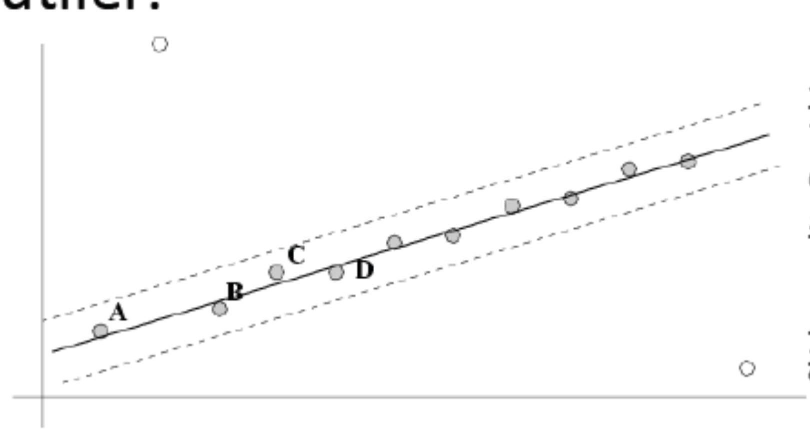
Sample size n	Proportion of outliers						
	5%	10%	20%	25%	30%	40%	50%
2	**2**	**3**	**5**	**6**	**7**	**11**	**17**
3	**3**	**4**	**7**	**9**	**11**	**19**	**35**
4	**3**	**5**	**9**	**13**	**17**	**34**	**72**
5	**4**	**6**	**12**	**17**	**26**	**57**	**146**
6	**4**	**7**	**16**	**24**	**37**	**97**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

Slide credit: David Lowe

After RANSAC

- RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers.
- Improve this initial estimate with 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