Line fitting with RANSAC

D.A. Forsyth

University of Illinois at Urbana Champaign

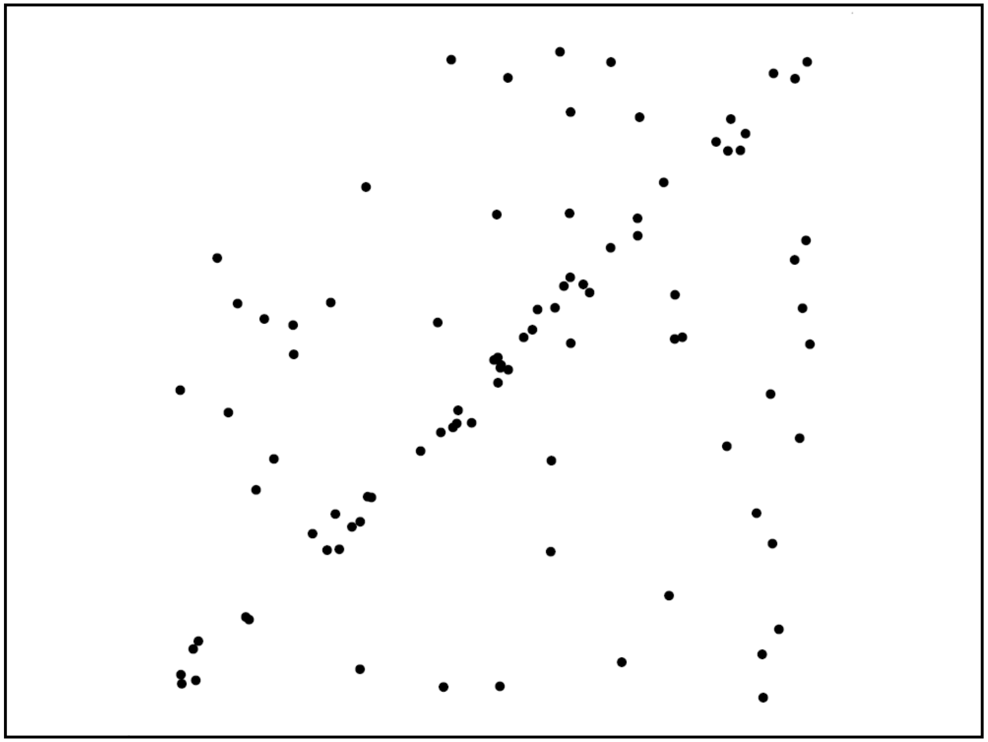
RANSAC

- Random sample consensus:
  - very general framework
- Outline:
  - Randomly choose a small initial subset of points
  - Fit a model to that subset
  - Find all inlier points that are “close” to the model and reject the rest as outliers
  - Do this many times and choose the model with the most inliers

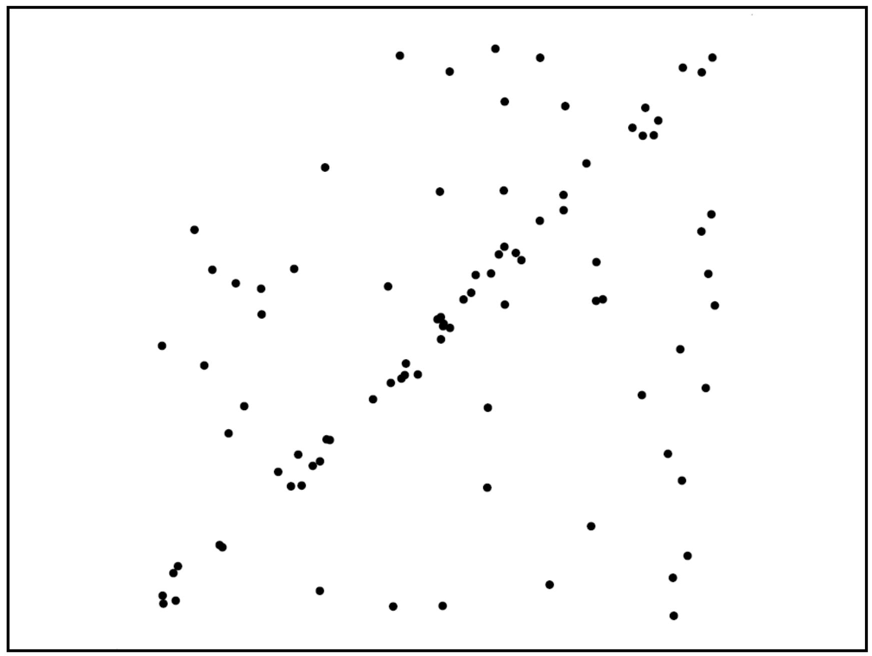
M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981

Voting schemes

- What if there are many outliers?
- let each point *vote* for all compatible models
  - Hopefully, outliers will not vote consistently for any single model
  - The model that receives the most votes is the best fit

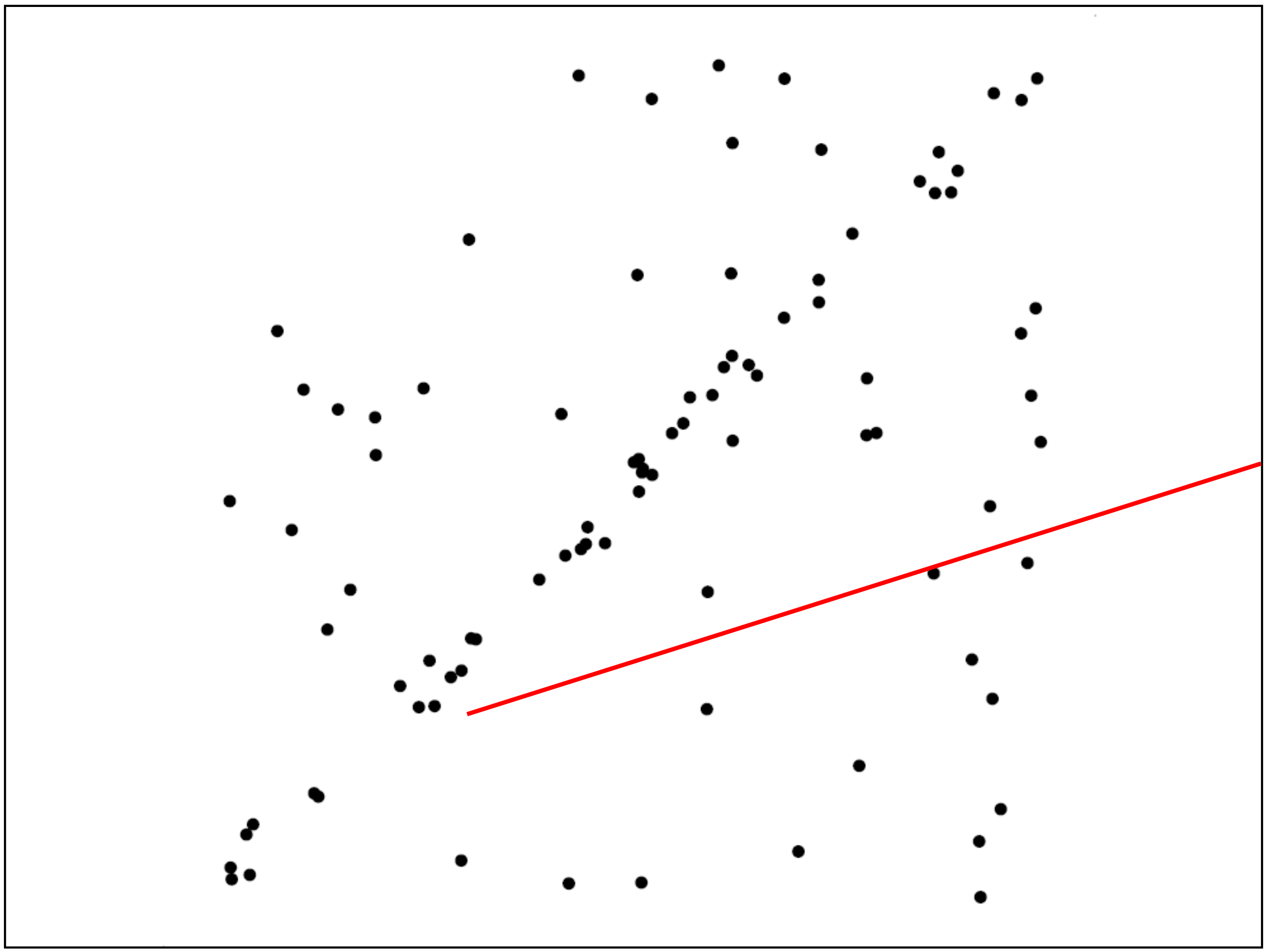


RANSAC for line fitting example



Source: R. Raguram

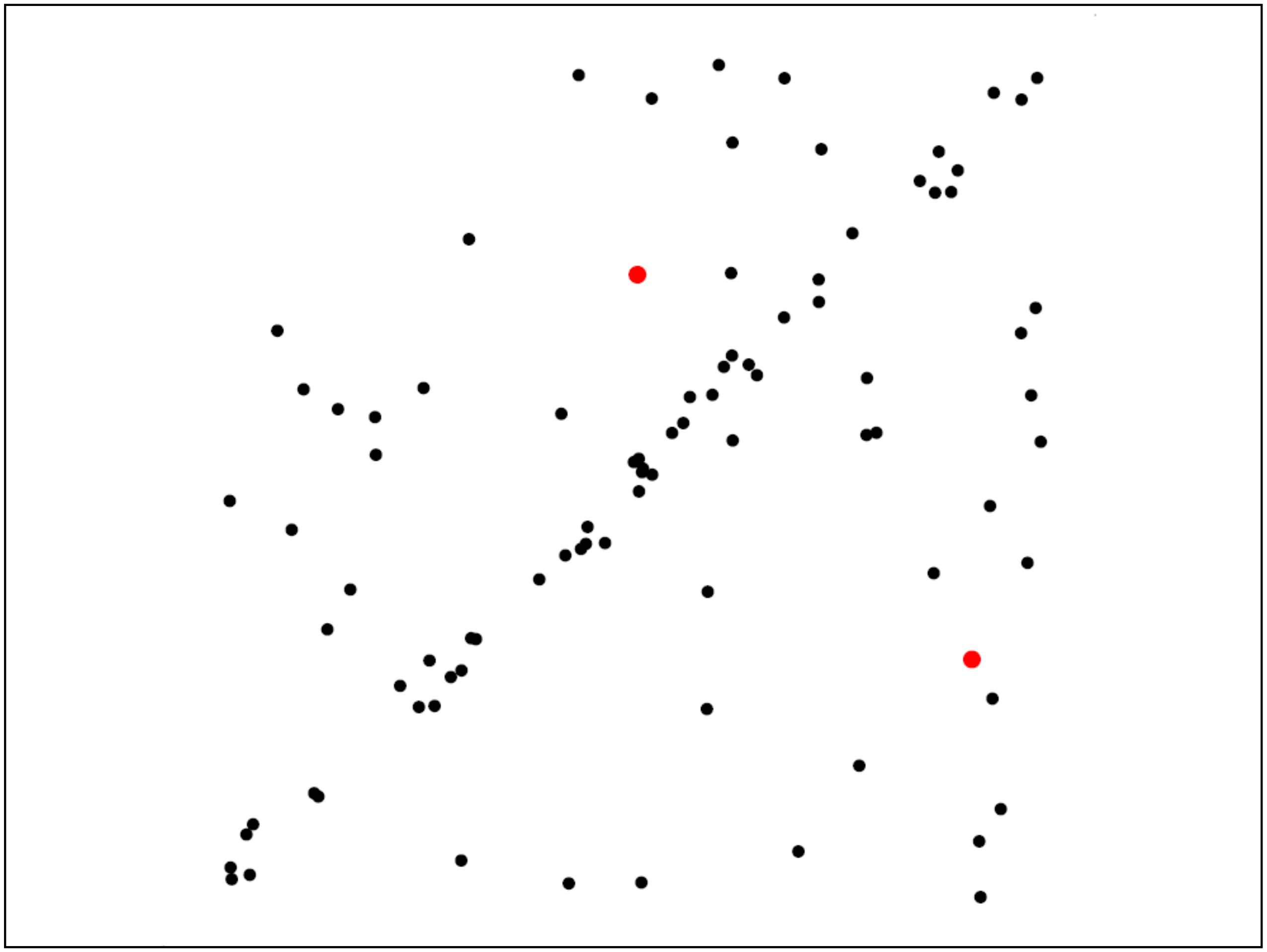
RANSAC for line fitting example



Least-squares fit

Source: R. Raguram

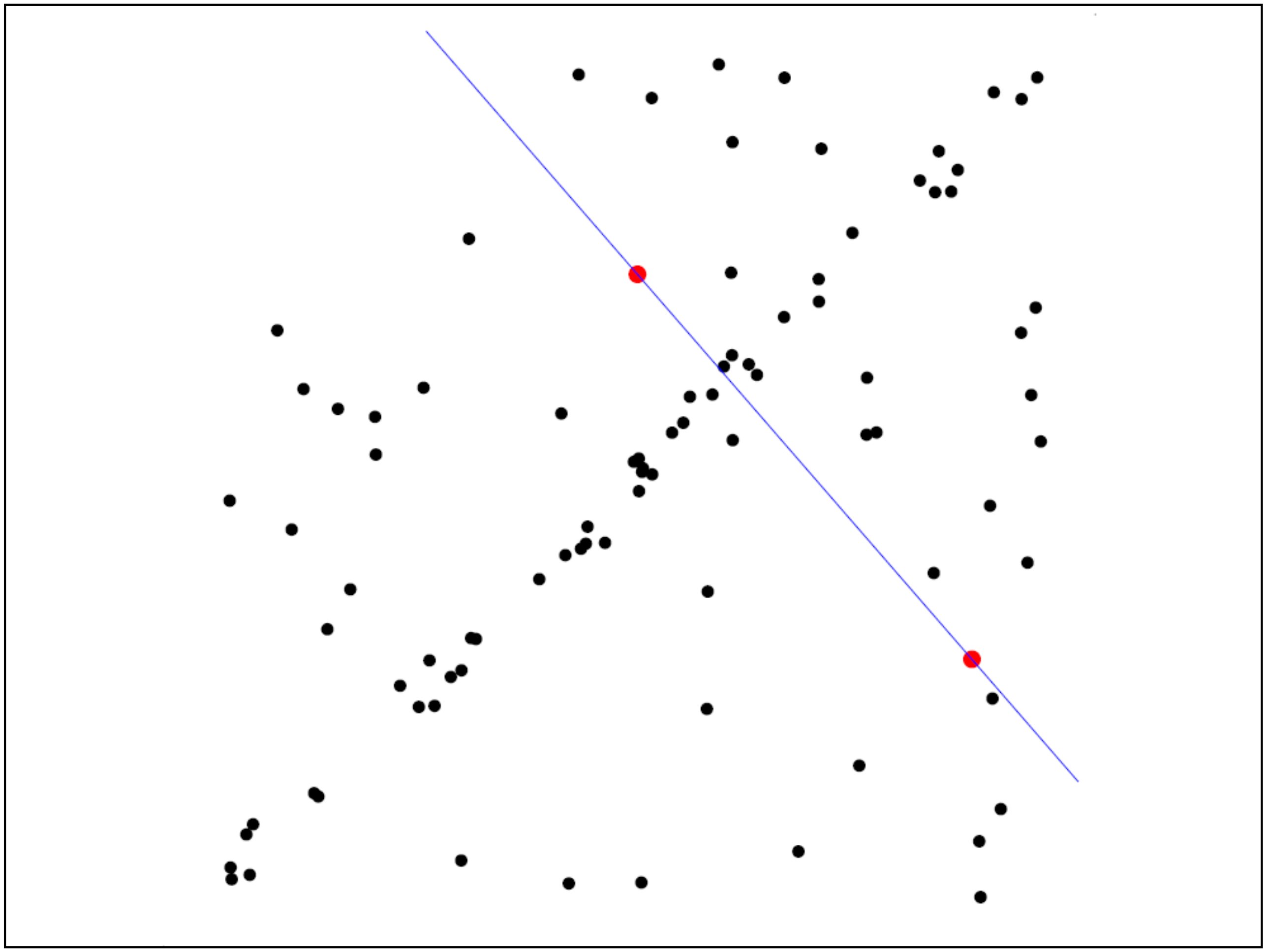
RANSAC for line fitting example



Source: R. Raguram

1. Randomly select minimal subset of points

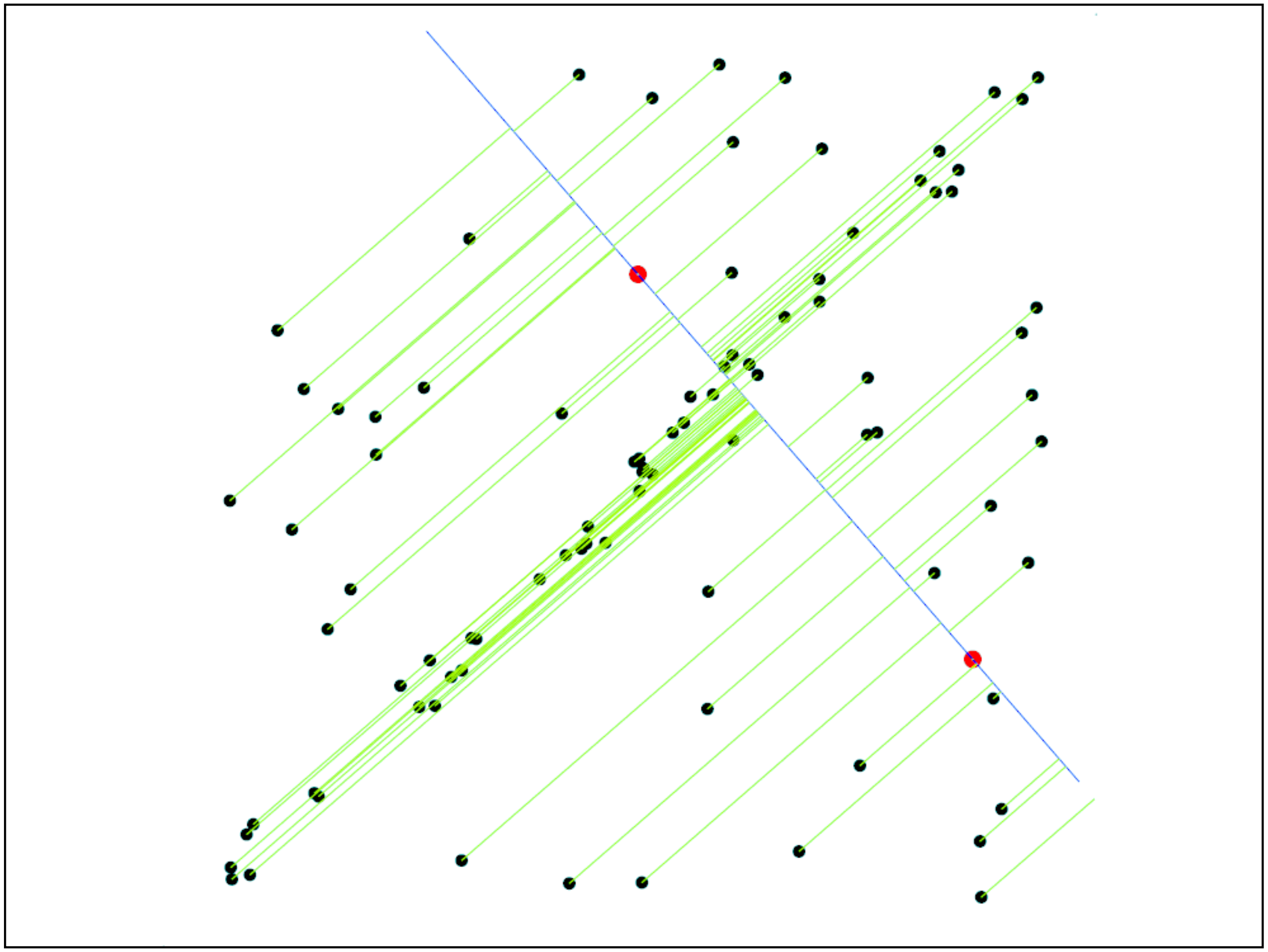
RANSAC for line fitting example



Source: R. Raguram

1. Randomly select minimal subset of points
2. Hypothesize a model

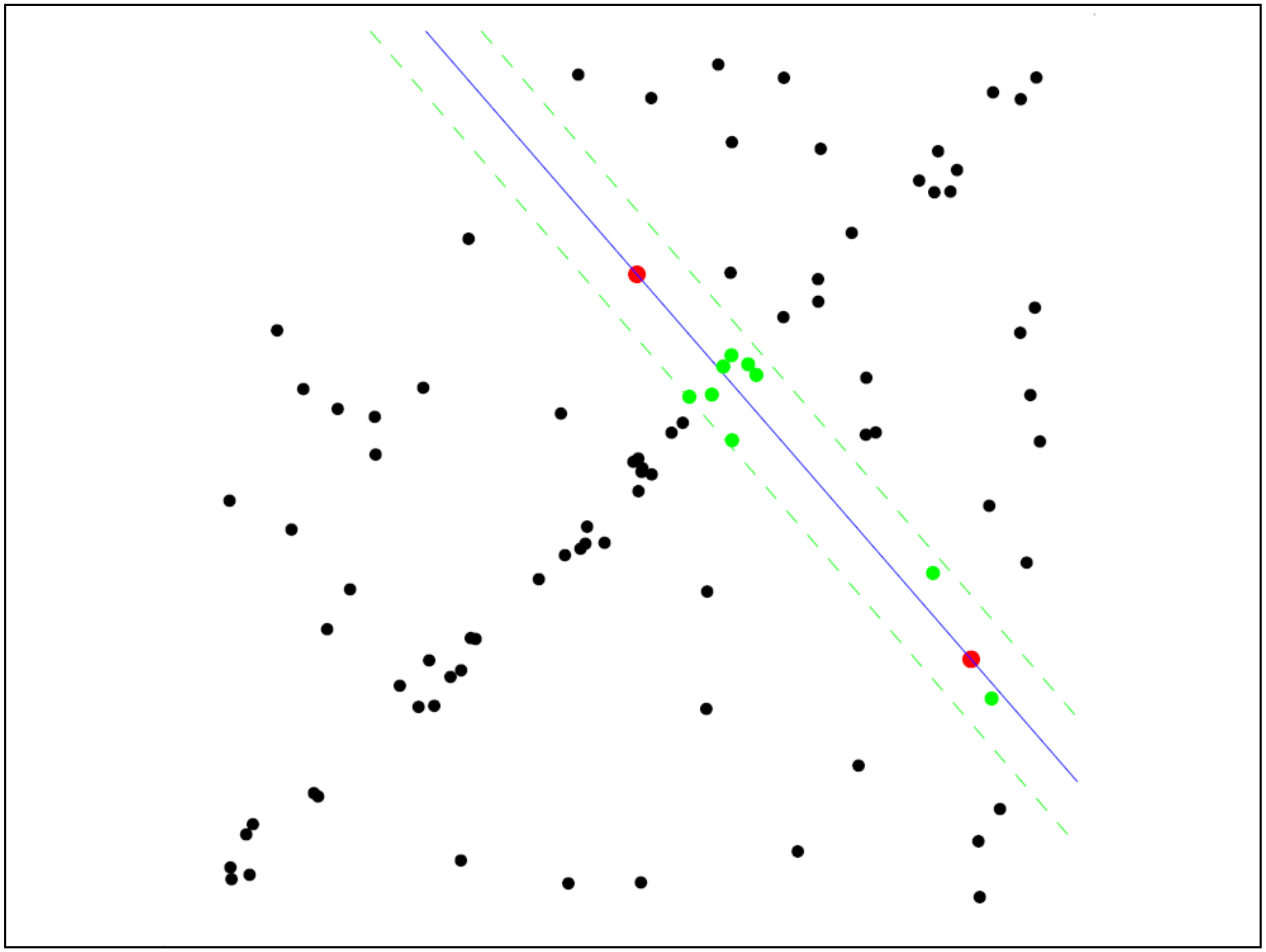
RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function

Source: R. Raguram

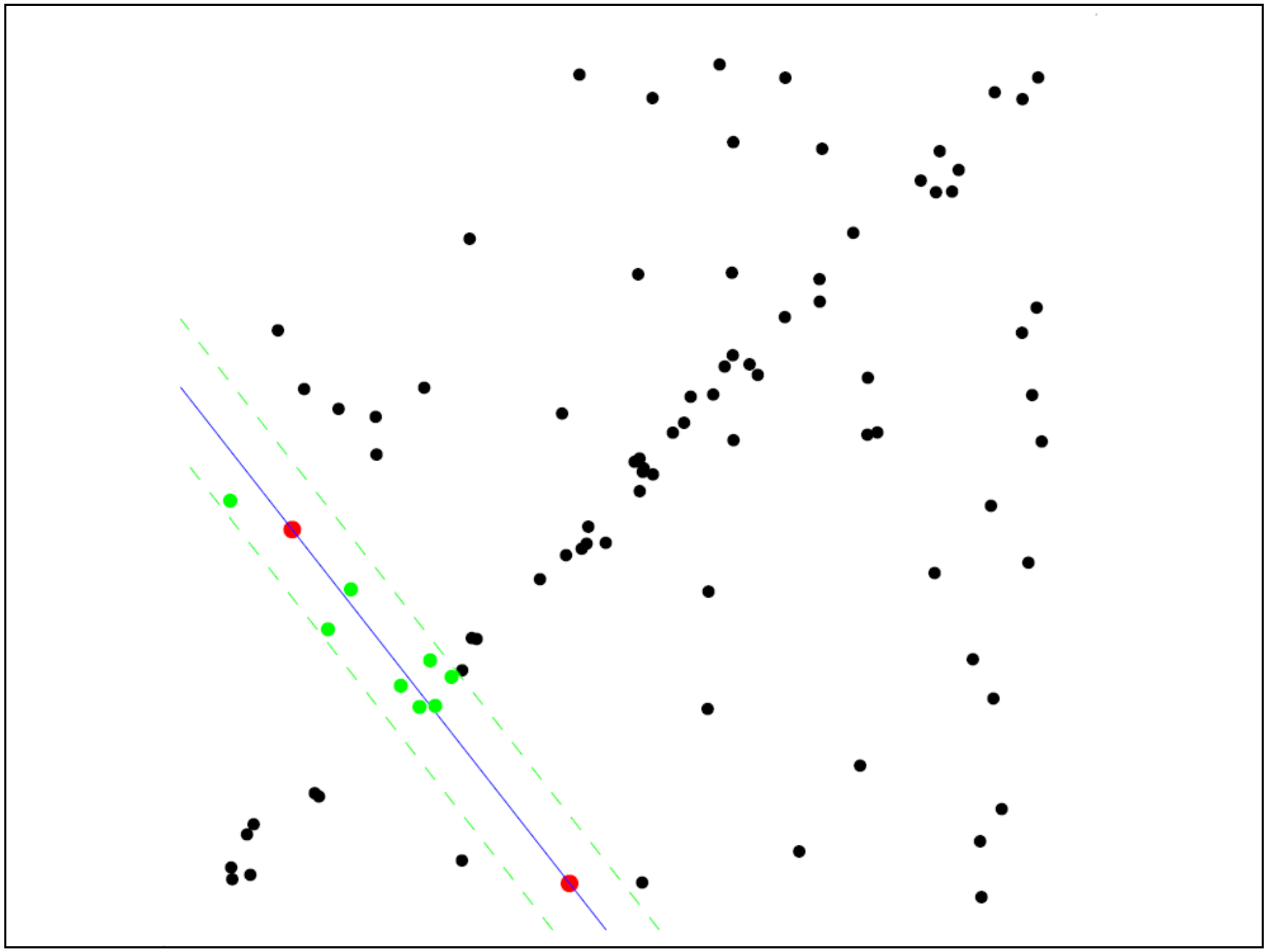
RANSAC for line fitting example



Source: R. Raguram

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model

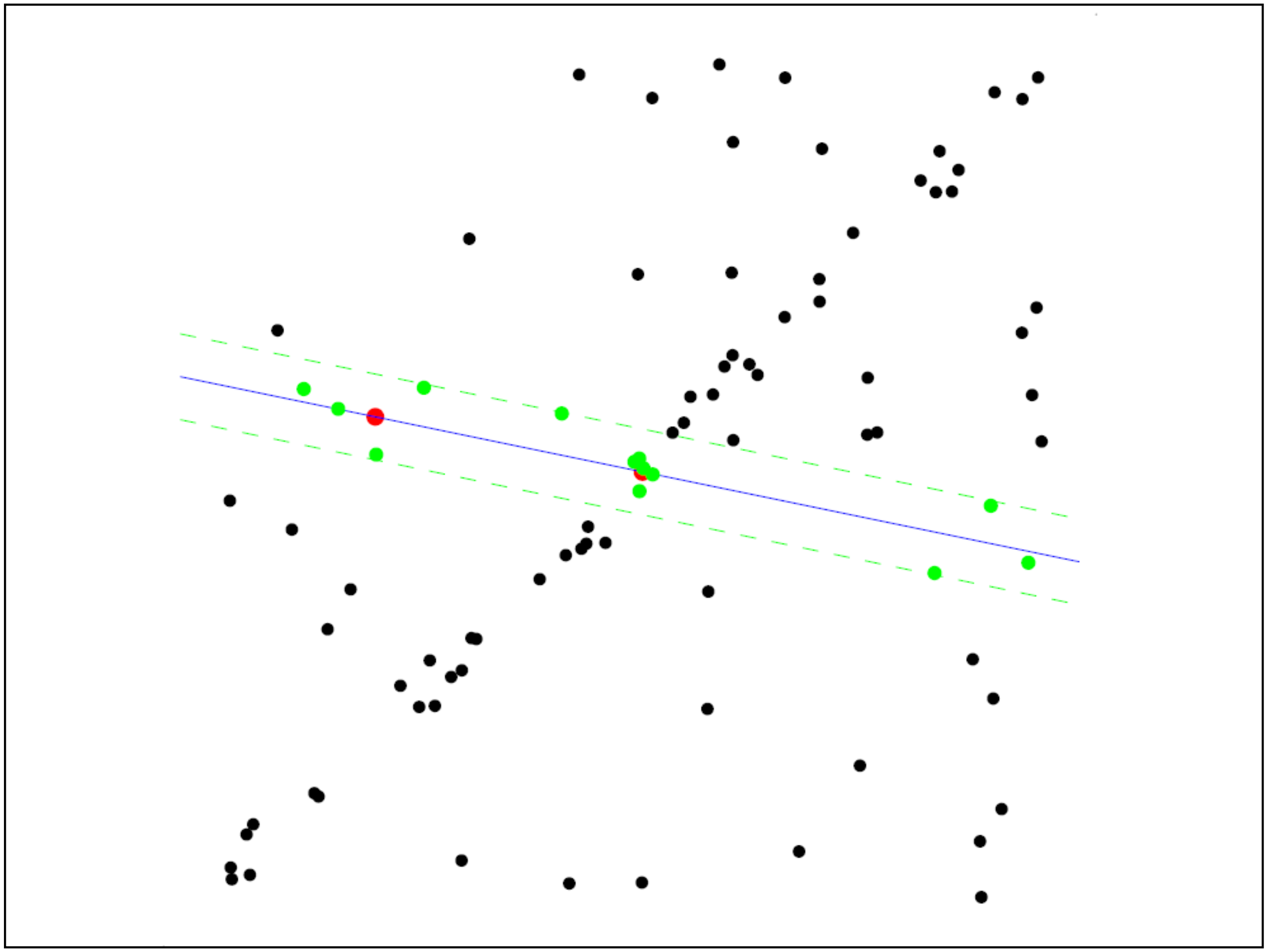
RANSAC for line fitting example



Source: R. Raguram

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

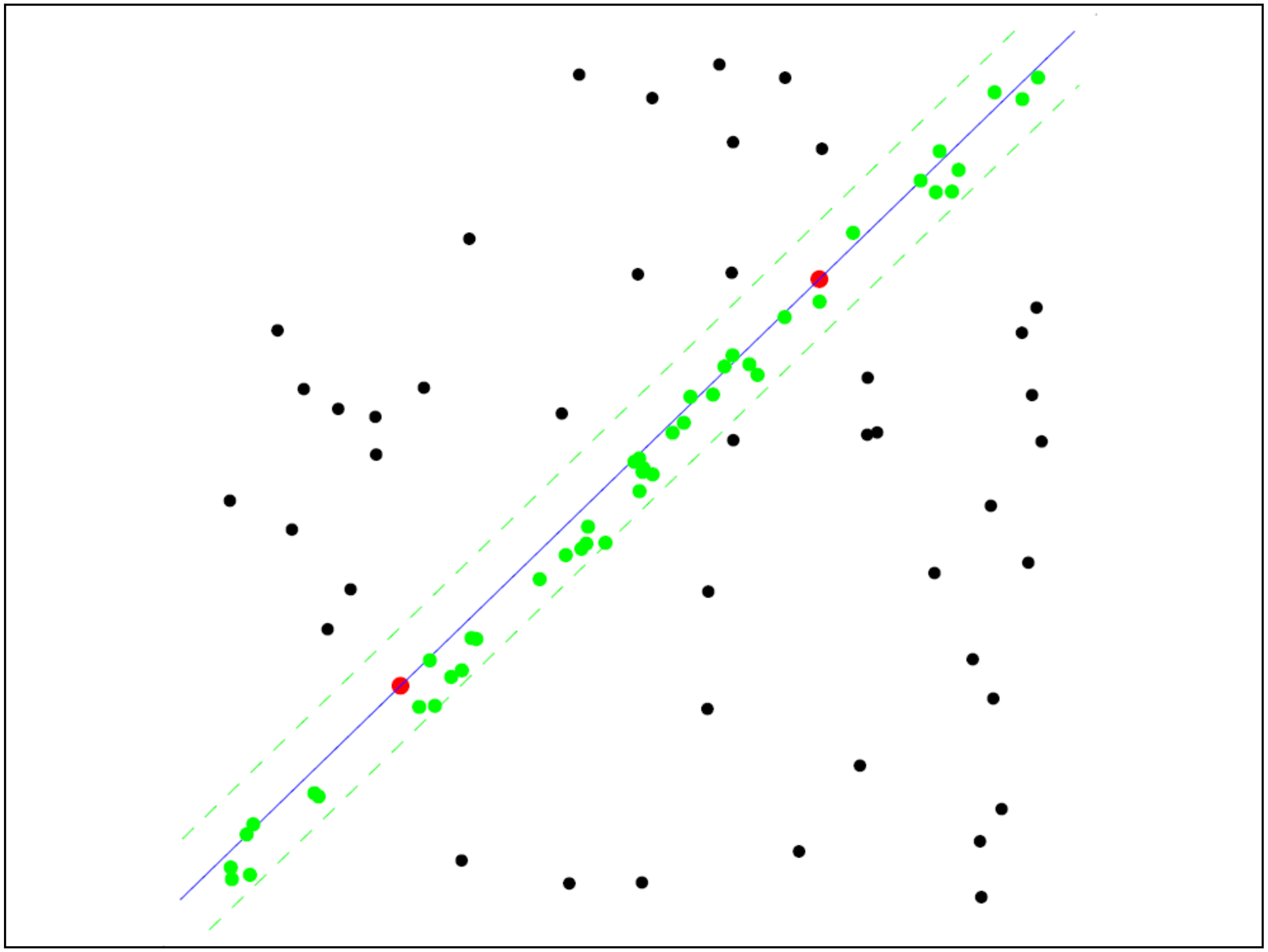
RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

RANSAC for line fitting example

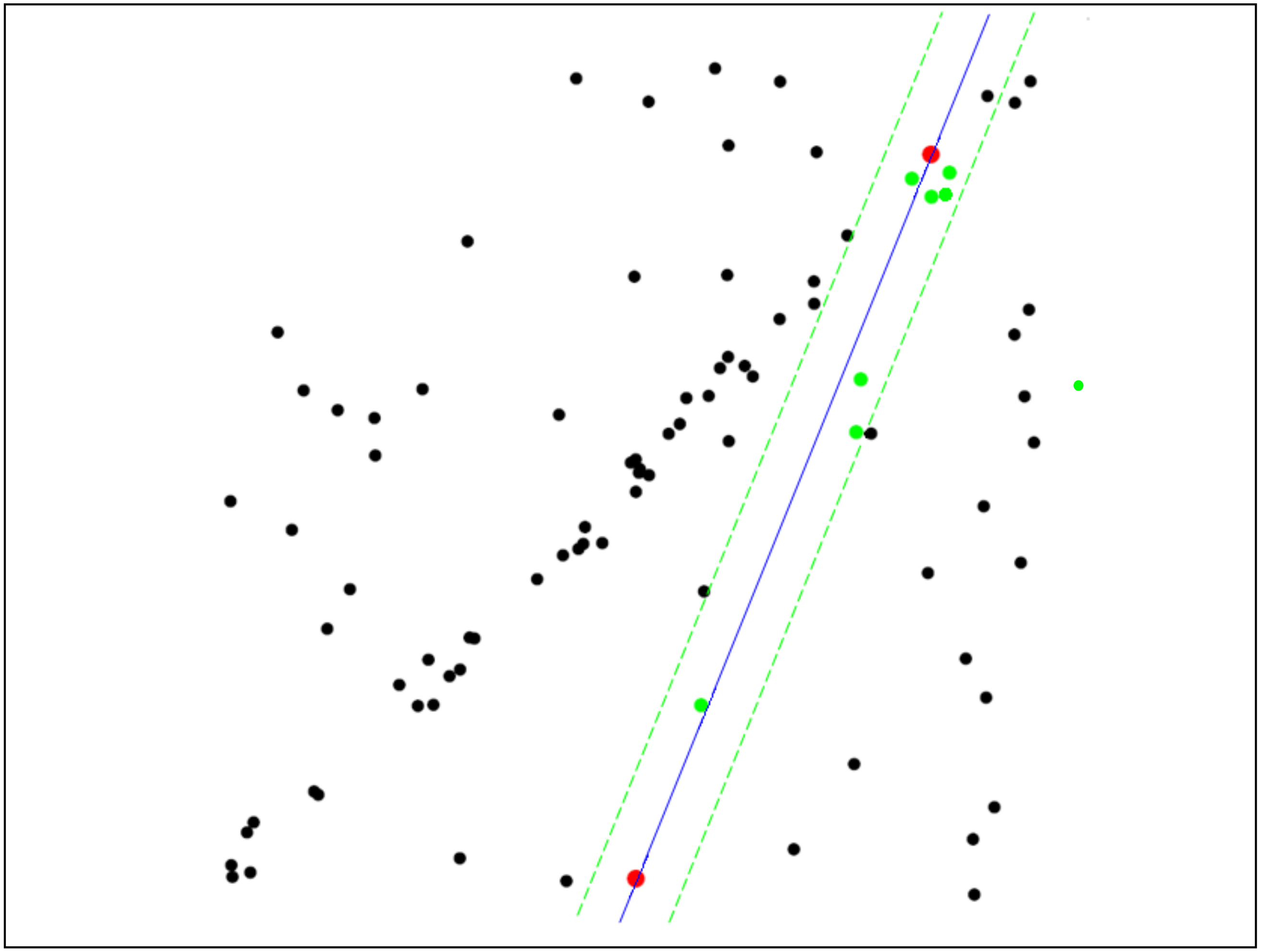


Source: R. Raguram

Uncontaminated sample

1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

RANSAC for line fitting example



1. Randomly select minimal subset of points
2. Hypothesize a model
3. Compute error function
4. Select points consistent with model
5. Repeat *hypothesize-and-verify* loop

Source: R. Raguram

RANSAC loop

- Repeat N times:
  - Draw s points uniformly at random
  - Fit model to these s points
  - Find *inliers* among the remaining points
    (distance or residual w.r.t. model is less than t)
  - If there are d or more inliers, accept the model and refit using all inliers

RANSAC: Choosing the parameters

- Initial number of points s
  - Typically minimum number needed to fit the model
- Distance threshold t for inliers
  - Need suitable assumptions, e.g., given zero-mean Gaussian noise with std. dev. σ, $t = 1.96\sigma$ will give ~95% probability of capturing all inliers
- Consensus set size d
  - Should match expected inlier ratio

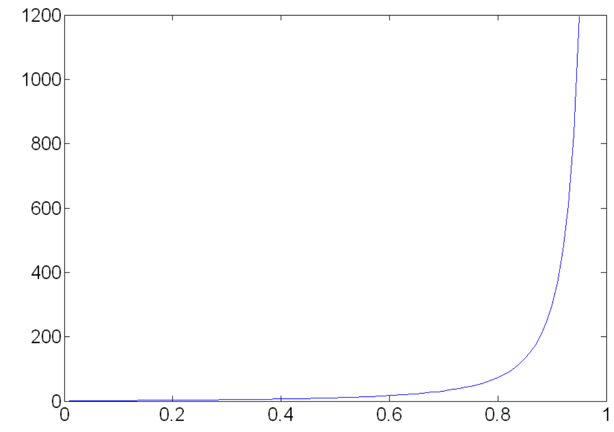
Adapted from M. Pollefeys

RANSAC: Choosing the parameters

- Choosing the number of iterations (initial samples) N:
  - Choose N so that, with probability p (e.g. 99%), at least one initial sample is free from outliers
  - Assuming an outlier ratio of e:

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

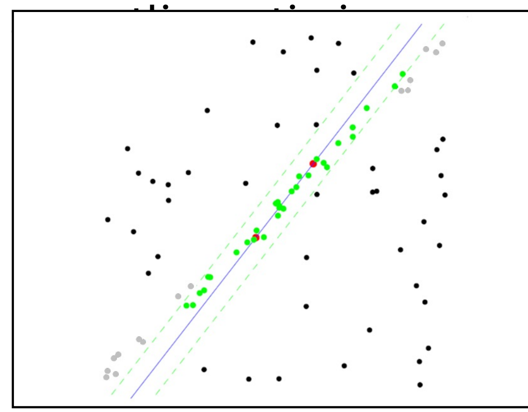
	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177



Source: M. Pollefeys

RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Lots of parameters to set
  - Number of iterations grows exponentially
  - Can’t always get a good initialization of the model based on the minimum number of samples

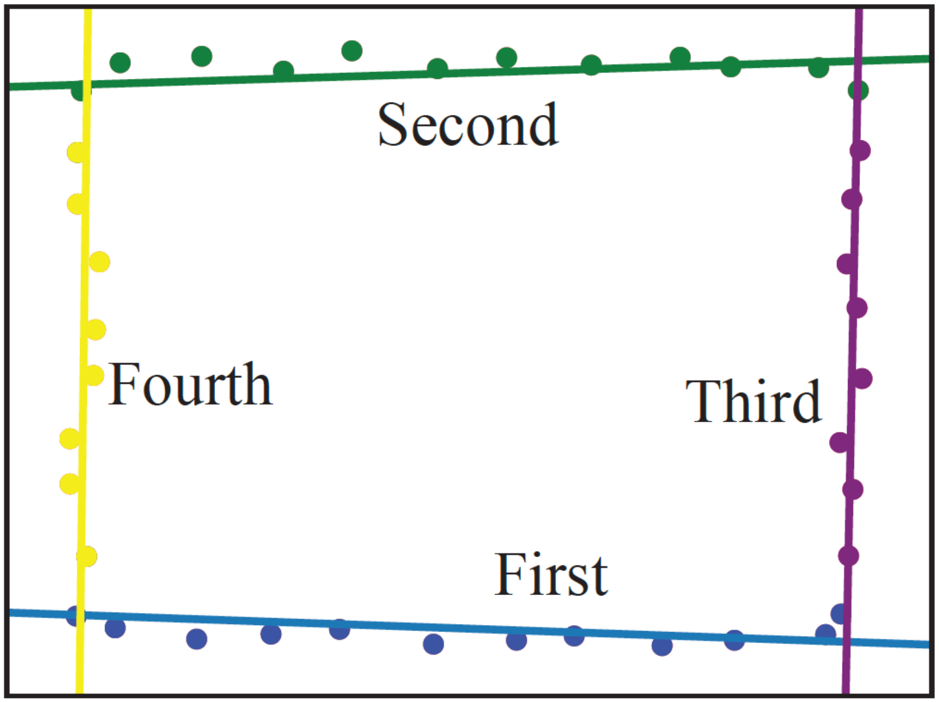


Incremental RANSAC

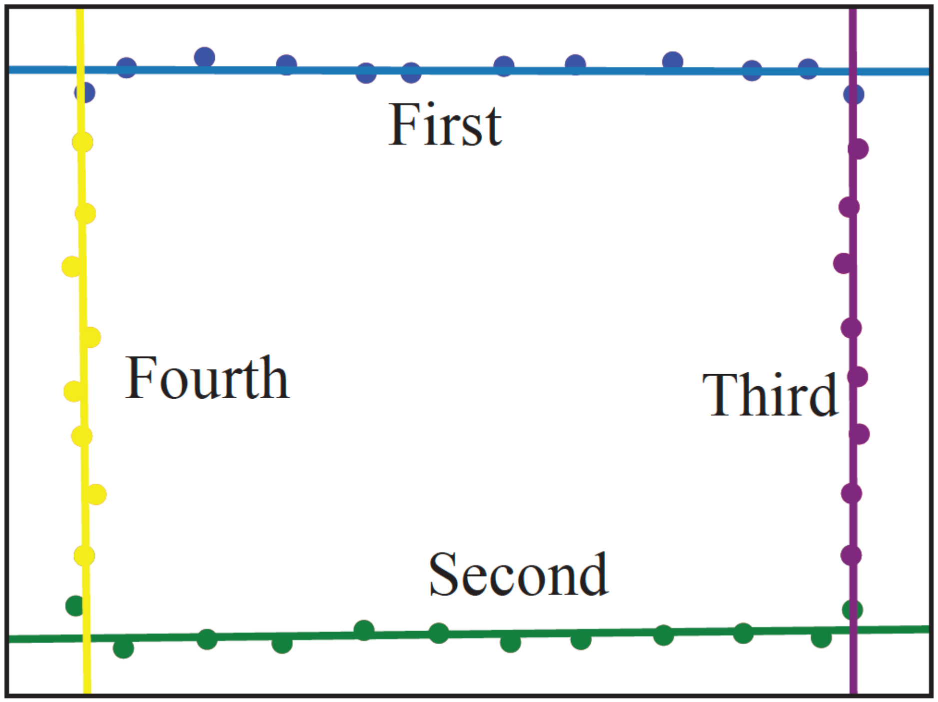
- To fit many lines to a dataset:
- Iterate:
  - Fit a line with RANSAC using IRLS (crucial!)
  - Remove inliers from dataset

- Q: when to stop?
- A:
  - when you have the right number of lines
  - when too little data is left
  - when the last line has few inliers

IRLS matters...



Least squares on inliers



IRLS squares on inliers

Things to think about…

14.4. Section 14.3.1 has:"The problem of updating the estimate of w reduces to estimating the probability that a coin comes up heads or tails given a sequence of flips" Explain.

14.5. A hyperplane in 4D is a set of points (x_1, x_2, x_3, x_4) such that $ax_1 + bx_2 + cx_3 + dx_4 + e = 0$ for some (a, b, c, d, e). You want to use RANSAC to fit a hyperplane to 4D data. How many points are there in a sample?

14.6. Which takes fewer samples: use RANSAC to fit a line to a collection of points that contains 20% outliers, with a probability of $1e-5$ that you see a good sample at least once; or use RANSAC to fit a plane to a collection of points that contains 20% outliers, with a probability of $1e-5$ that you see a good sample at least once? Why?

14.7. What will happen if you use RANSAC to fit a line to a dataset that contains no outliers?

14.8. What will happen if you use RANSAC to fit a line to a dataset that contains only outliers?