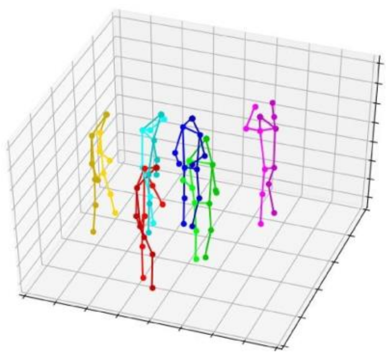
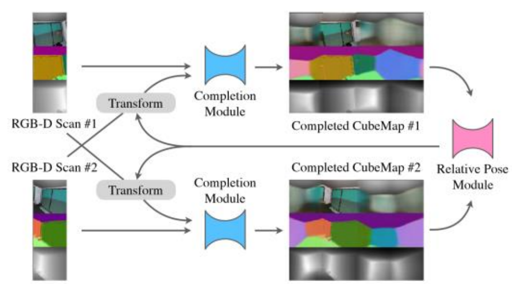
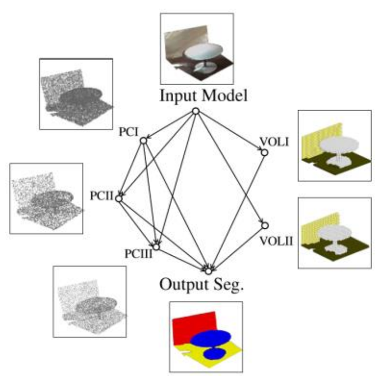
CS376 Computer Vision
Lecture 8: RANSAC + Robust Fitting







Qixing Huang

Feb. 18th 2019

Last Lecture

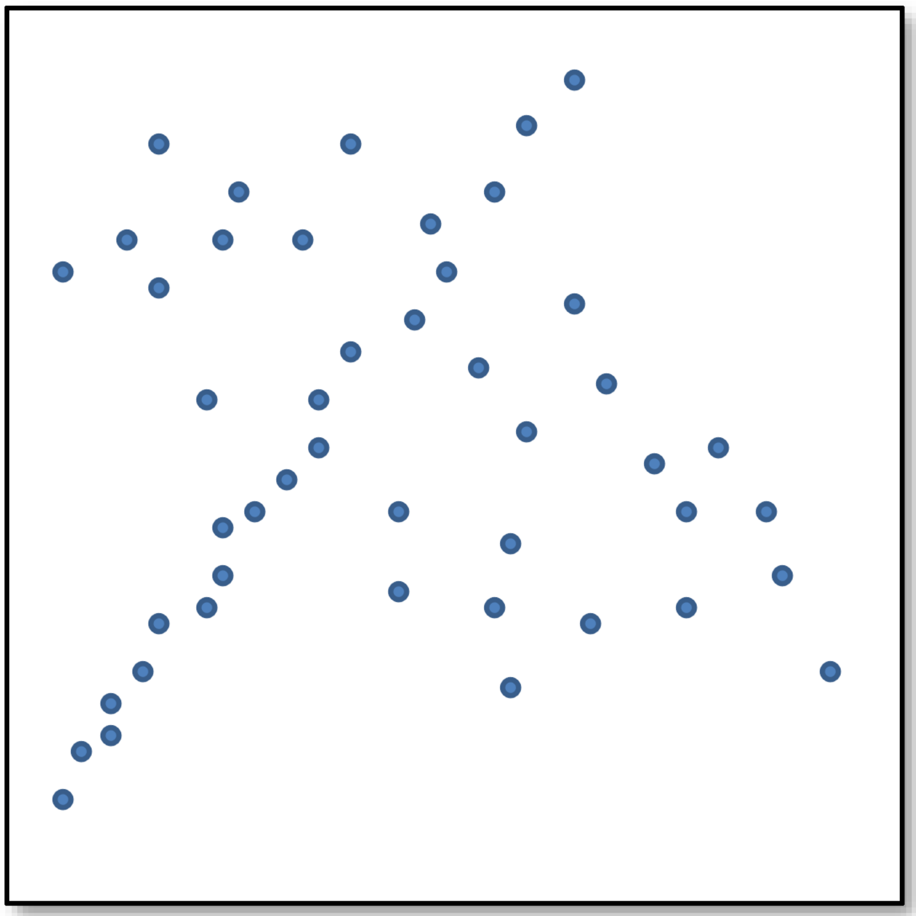
- Hough transform for model fitting (e.g., line)
- Pros:
  - Detecting multiple lines whose number is not fixed
  - Input may contain outliers
- Cons:
  - May be affected by noisy edge points

This lecture – two other model fitting techniques

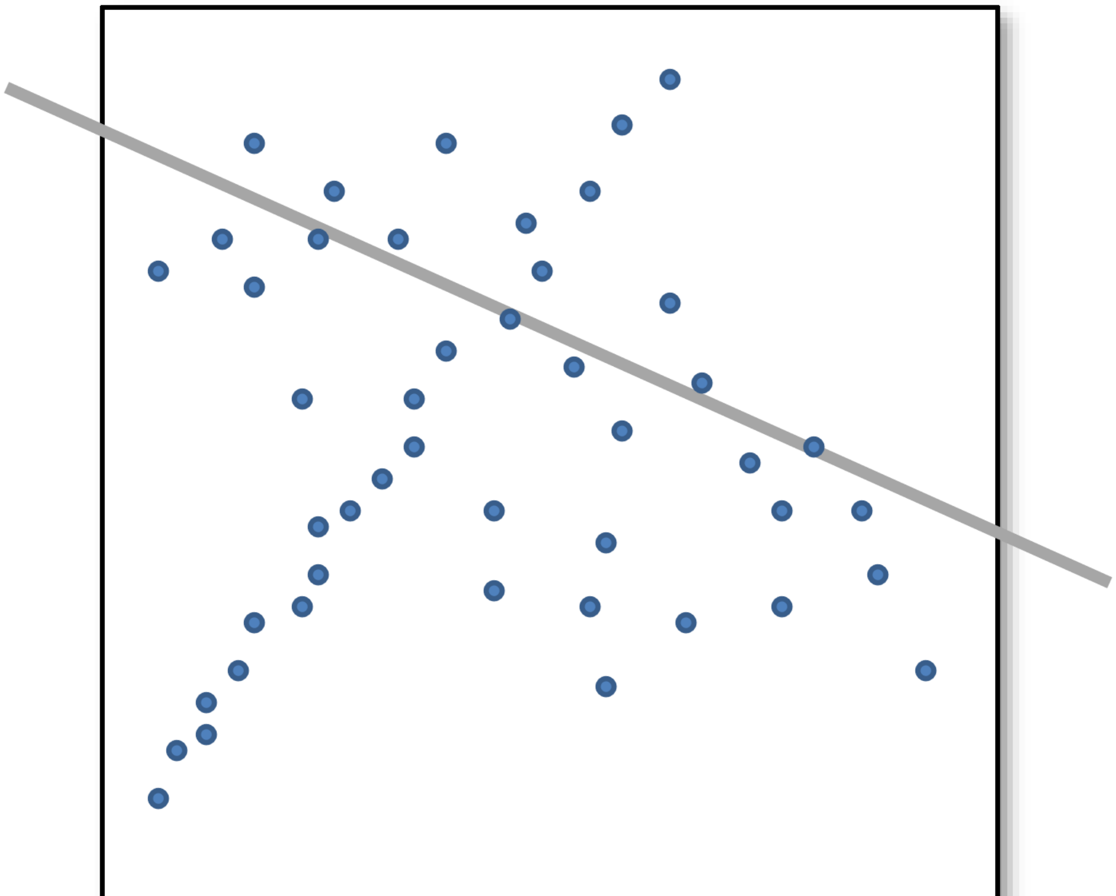
- RANSAC

- Robust fitting

Counting inliers

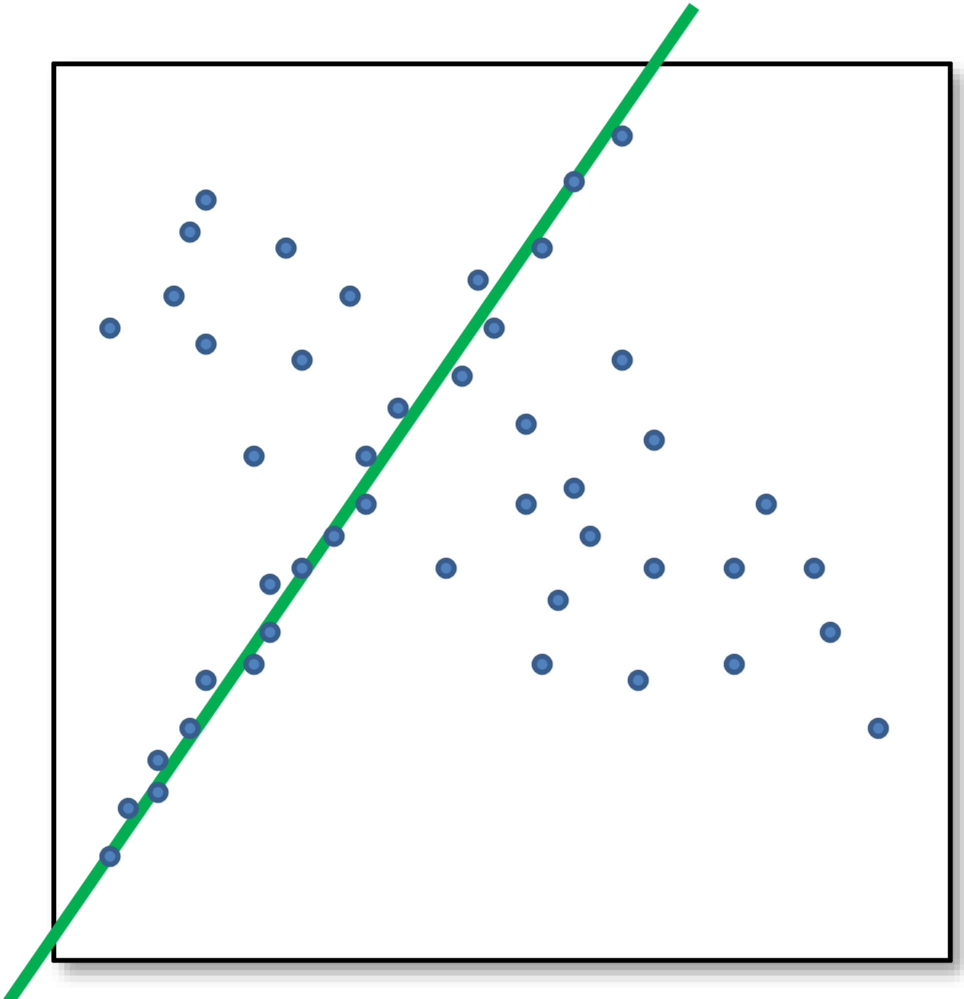


Counting inliers



Inliers: 4

Counting inliers



Inliers> 20

How de we find the best line?

- Unlike least-squares, no simple closed-form solution – we will get back to this, e.g., using robust norms

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

RANSAC

- General version:
  - Randomly choose s samples
    - Typically s= minimum sample size that lets you fit a model
  - Fit a model (e.g., line) to those samples
  - Count the number of inliers that approximately fit the model
  - Repeat N times
  - Choose the model that has the largest set of inliers

Analysis of RANSAC

however, can be determined as a function of the desired probability of success p using a theoretical result. Let p be the desired probability that the RANSAC algorithm provides a useful result after running. RANSAC returns a successful result if in some iteration it selects only inliers from the input data set when it chooses the n points from which the model parameters are estimated. Let w be the probability of choosing an inlier each time a single point is selected, that is,

w = number of inliers in data / number of points in data

A common case is that w is not well known beforehand, but some rough value can be given. Assuming that the n points needed for estimating a model are selected independently, w^n is the probability that all n points are inliers and $1 - w^n$ is the probability that at least one of the n points is an outlier, a case which implies that a bad model will be estimated from this point set. That probability to the power of k is the probability that the algorithm never selects a set of n points which all are inliers and this must be the same as $1 - p$. Consequently,

$$1 - p = (1 - w^n)^k$$

which, after taking the logarithm of both sides, leads to

$$k = \frac{\log(1 - p)}{\log(1 - w^n)}$$

This result assumes that the n data points are selected independently, that is, a point which has been selected once is replaced and can be selected again in the same iteration. This is often not a reasonable approach and the derived value for k should be taken as an upper limit in the case that the points are selected without replacement. For example, in the case of finding a line which fits the data set illustrated in the above figure, the RANSAC algorithm typically chooses two points in each iteration and computes `maybe_model` as the line between the points and it is then critical that the two points are distinct.

To gain additional confidence, the standard deviation or multiples thereof can be added to k. The standard deviation of k is defined as

$$\mathrm{SD}(k) = \frac{\sqrt{1 - w^n}}{w^n}$$

https://en.wikipedia.org/wiki/Random_sample_consensus

Basic RANSAC

Comments

$$N = \frac{log(1-p)}{log(1-\omega^n)} \text{ with } p = 0.99$$

ω

N	90	80	70	60	50
2	3	5	7	11	17
3	4	7	11	19	35
4	5	9	17	34	72
5	6	12	26	57	146
6	7	16	37	97	293
7	8	20	54	163	588
8	9	26	78	272	1177

n

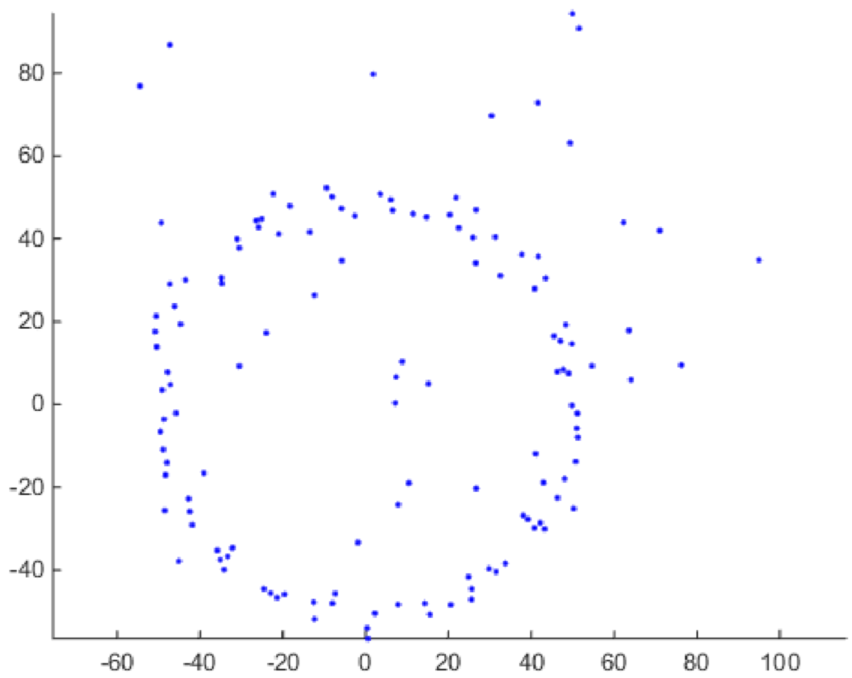
- Typically we do not know the ratio of outliers in our data set, hence we do not know the probability w or the number N
- Instead of operating with a larger than necessary N we can modify RANAC to adaptively estimate N as we perform the iterations

How to estimate the Inlier ratio?

Adaptive RANSAC

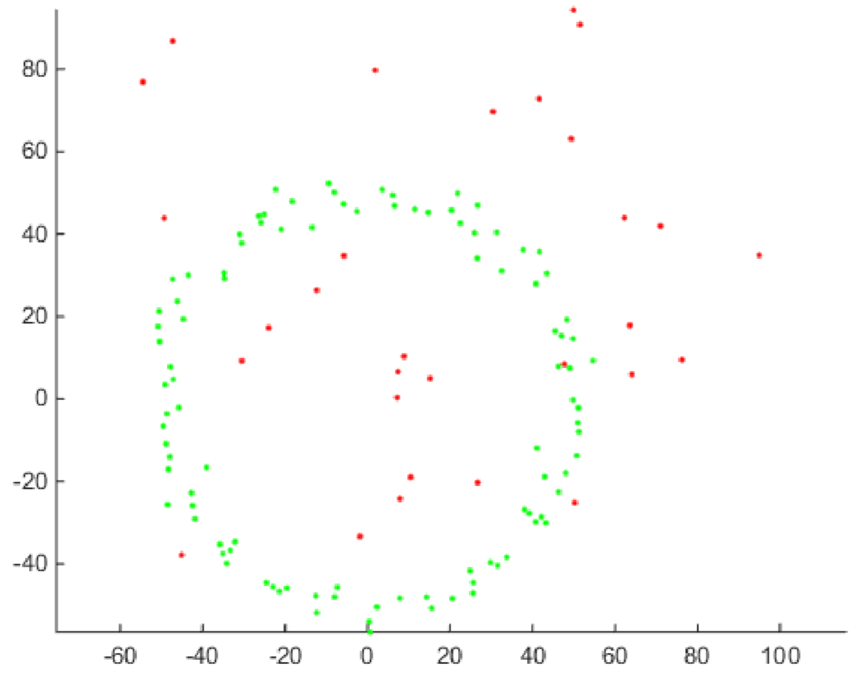
- Objective
  - To robustly fit a model y = f(x,a) to a data set S containing outliers
- Algorithm
  - Let N = infty, S_{IN}= null and #iterations = 0
  - While N > #iterations repeat 3-5
  - Estimate parameters a_{tst} from a random n-tuple from S
  - Determine inlier set S_{tst}, i.e., data points within a distance t of the model $y = f(x; a_{tst})$
  - If $|S_{tst}| > |S_{IN}|$, set $S_{IN} = S_{tst}$, $a = a_{tst}$, $w = |S_{IN}|/|S|$ and $N = \log(1-p)/\log(1-w^n)$ with p = 0.99. Increase #iterations by 1

Example



Fit a circle $(x - x_0)^2 + (y - y_0)^2 = r^2$

Example



Circle + Gaussian noise

Random points

Fit a circle $(x - x_0)^2 + (y - y_0)^2 = r^2$

Least square approach

Separate observables from parameters:

$$(x-x_0)^2+(y-y_0)^2=r^2$$

$$x^2-2xx_0+x_0^2+y^2-2yy_0+y_0^2=r^2$$

$$2xx_0+2yy_0+r^2-x_0^2-y_0^2=x^2+y^2$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix}\begin{bmatrix} 2x_0 \\ 2y_0 \\ r^2-x_0^2-y_0^2 \end{bmatrix}=\begin{bmatrix} x^2+y^2 \end{bmatrix}$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}=\begin{bmatrix} x^2+y^2 \end{bmatrix}$$

So for each observation (x_i, y_i) we ge equation

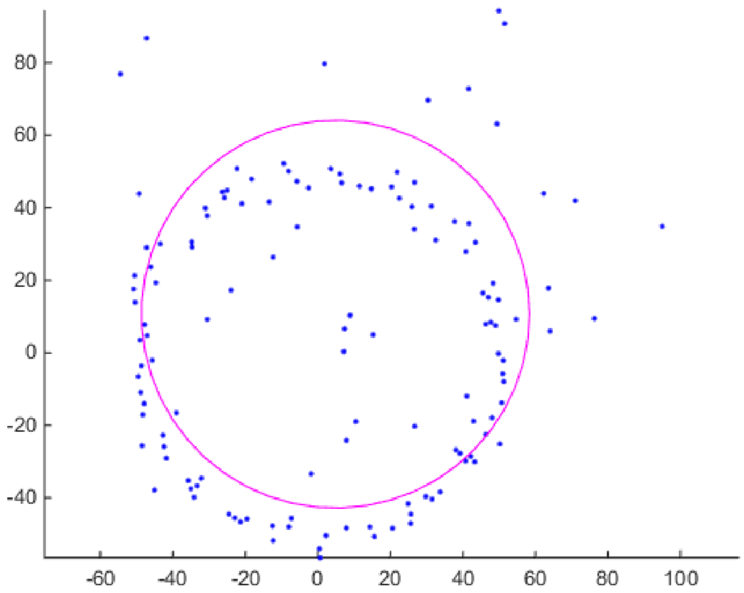
$$\begin{bmatrix} x_i & y_i & 1 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}=\begin{bmatrix} {x_i}^2+{y_i}^2 \end{bmatrix}$$

From all our N observations we get a system of linear equations

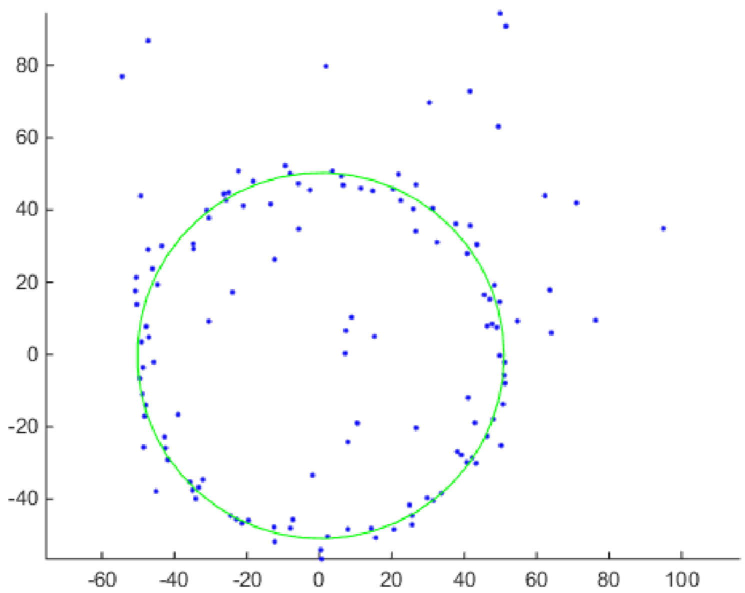
$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \vdots & \\ x_N & y_N & 1 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}=\begin{bmatrix} x_1^2+y_1^2 \\ x_2^2+y_2^2 \\ \vdots \\ x_N^2+y_N^2 \end{bmatrix}$$

$$A\boldsymbol{p}=\boldsymbol{b}$$

Comparison



80
60
40
20
0
-20
-40
-60
-40
-20
0
20
40
60
80
100



80
60
40
20
0
-20
-40
-60
-40
-20
0
20
40
60
80
100

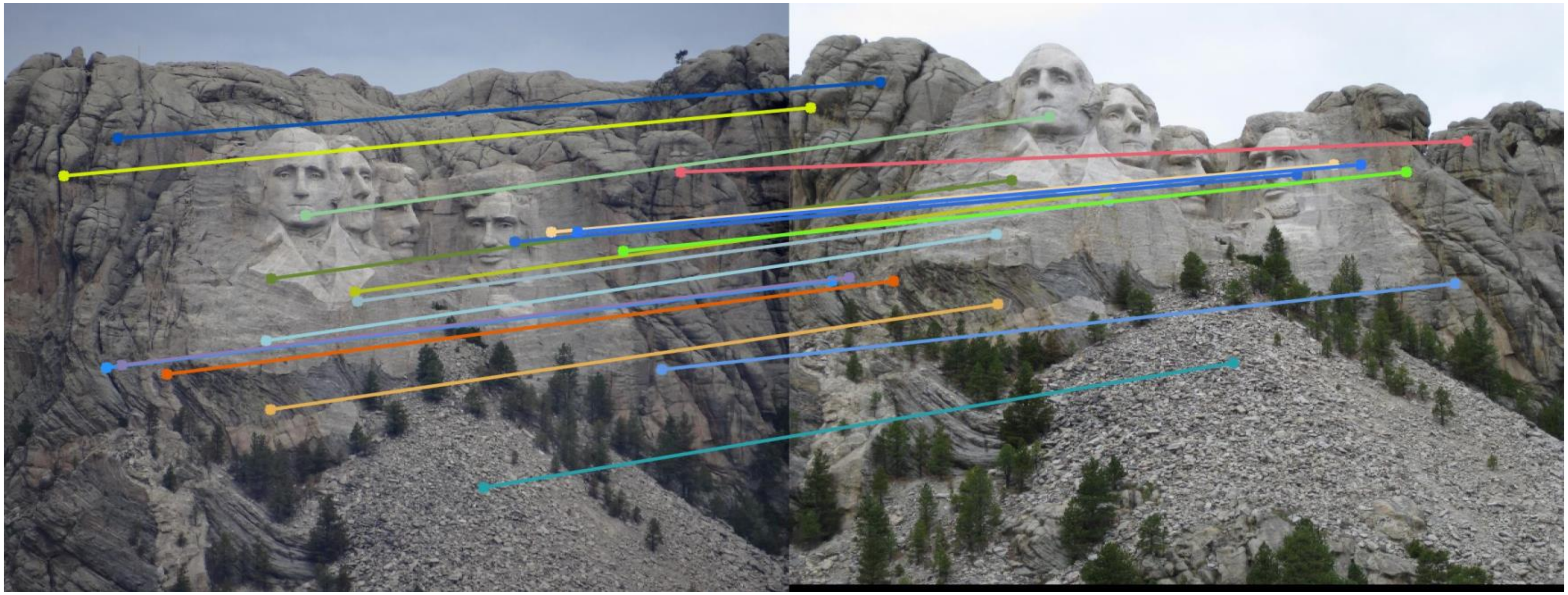
Summary

- RANSAC
  - A robust iterative method for estimating the parameters of a mathematical model from a set of observed data containing outliers
  - Separates the observed data into “inliers”and “outliers”
  - Can be applied in an iterative manner to obtain multiple models
  - Not perfect

Application of RANSAC

Recognising Panoramas [Brown and Lowe’ 03]

Application of RANSAC



Camera calibration

Application of RANSAC

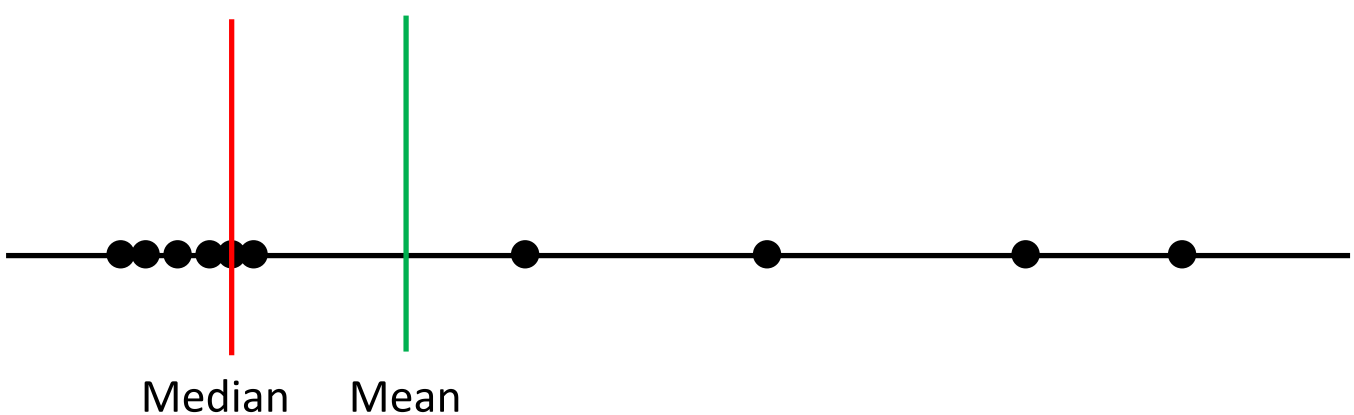
Structure-from-motion [Snavely et al. 06]

Reweighted Least Squares

1D case

- Mean $$\min_x \sum_{i=1}^{n} (x - x_i)^2$$

- Median $$\min_x \sum_{i=1}^{n} |x - x_i|$$



General formulation

L^p norm linear regression [edit]

To find the parameters $\boldsymbol\beta = (\beta_1, \ldots, \beta_k)^T$ which minimize the L^p norm for the linear regression problem,

$$\underset{\boldsymbol\beta}{\arg\min} \big\| \mathbf{y} - X\boldsymbol\beta \big\|_p = \underset{\boldsymbol\beta}{\arg\min} \sum_{i=1}^{n} |y_i - X_i\boldsymbol\beta|^p,$$

the IRLS algorithm at step $t + 1$ involves solving the weighted linear least squares problem:[4]

$$\boldsymbol\beta^{(t+1)} = \underset{\boldsymbol\beta}{\arg\min} \sum_{i=1}^{n} w_i^{(t)} |y_i - X_i\boldsymbol\beta|^2 = (X^{\rm T} W^{(t)} X)^{-1} X^{\rm T} W^{(t)} \mathbf{y},$$

where $W^{(t)}$ is the diagonal matrix of weights, usually with all elements set initially to:

$$w_i^{(0)} = 1$$

and updated after each iteration to:

$$w_i^{(t)} = \big|y_i - X_i\boldsymbol\beta^{(t)}\big|^{p-2}.$$

https://en.wikipedia.org/wiki/Iteratively_reweighted_least_squares

Optimization

- For L^p-norm where $p>=1$, the objective function is convex and we can apply convex optimization

- For other robust norms, a popular approach is reweighted least squares

When the fraction of inliers > 50%

where $W^{(t)}$ is the diagonal matrix of weights, usually with all elements set initially to:

$$w_i^{(0)} = 1$$

and updated after each iteration to:

$$w_i^{(t)} = \left|y_i - X_i \boldsymbol{\beta}^{(t)}\right|^{p-2}.$$

In the case $p = 1$, this corresponds to least absolute deviation regression (in this case, the problem would be better approached by use of linear programming methods,[5] so the result would be exact) and the formula is:

$$w_i^{(t)} = \frac{1}{\left|y_i - X_i \boldsymbol{\beta}^{(t)}\right|}.$$

To avoid dividing by zero, regularization must be done, so in practice the formula is:

$$w_i^{(t)} = \frac{1}{\max\left\{\delta, \left|y_i - X_i \boldsymbol{\beta}^{(t)}\right|\right\}}.$$

where δ is some small value, like 0.0001.[5] Note the use of δ in the weighting function is equivalent to the Huber loss function in robust estimation.

https://en.wikipedia.org/wiki/Iteratively_reweighted_least_squares