Lecture 11: Robust Fitting and Matching II

[AIX7021] Computer Vision

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

Tentative schedule

week	topic	date
1	Introduction and Basics	09.01
2	Image Process I	09.08
3	Image Process II	09.15
4	(휴강)	09.22
5	Feature Detection and Matching I	09.29
6	Feature Detection and Matching II	10.06
7	Clustering and Segmentation I & II	10.13 & 10.16
8	**Mid-Term Exam**	**10.20**
9	Mid-Term Solution	10.27
10	Dimensional Reduction I	11.03
11	Dimensional Reduction II & Robust Fitting and Matching I	11.10
12	Robust Fitting and Matching II	11.17
13	Object Recognition & Motion and Tracking	11.24
14	Image Classification	12.01
15	**Final exam**	**12.08**

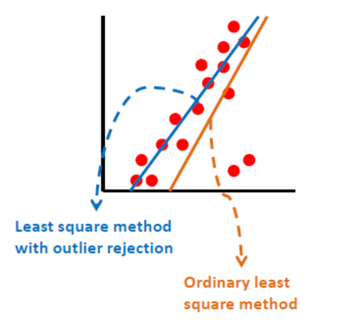
Fitting data with outliers

Effect of outliers

- Ordinary least square methods are very sensitive to noises
- We need to reject the outliers effectively

Handling outliers

- Robust statistics
- RANSAC
- Hough transform



Robust Statistics

Robust statistics

Limitation of classical methods

- The data errors assumed to be normally distributed
- But, outliers may be from arbitrary distributions

Motivation and implementation

- The estimators should not be unduly affected by small departures from model assumptions
- We reduce the effect of samples with high errors

Objective function for robust estimation

$$\min \rho(u(\mathbf{x}_i\,;\,\mathbf{p}), \theta) \quad \text{instead of} \quad \min u(\mathbf{x}_i\,;\,\mathbf{p}) \tag{1}$$

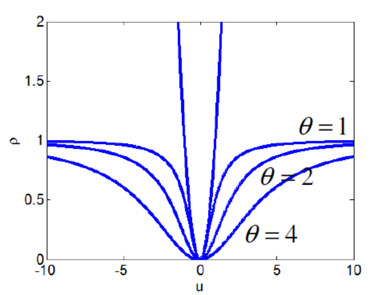
$$u(\mathbf{x}_i\,;\,\mathbf{p}) = \sum_{i=1}^{n} (a x_i + b - y_i)^2 \quad \text{where } \mathbf{p} = (a, b)^T \tag{2}$$

M-estimator

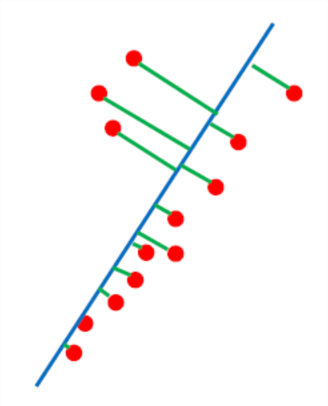
Objective function for M-estimator

$$\min \rho(u(\mathbf{x}_i\,;\,\mathbf{p}), \theta) \tag{3}$$

where penalty function $\rho(u, \theta)$, residual u, model parameter p and scale parameter θ

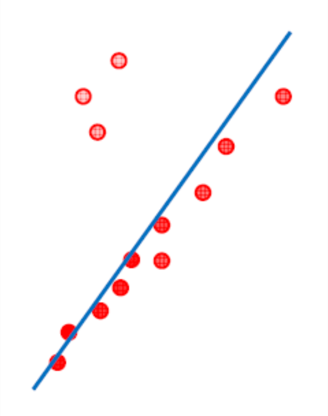


Robust least square with M-estimator 1/3

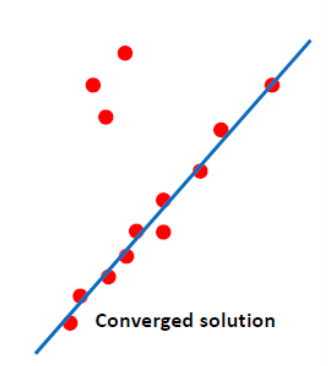


1. Set the scale parameter θ
2. Find the initial solution
3. Compute the residual for each sample $u(\mathbf{x}_i\,;\,\mathbf{p})$

Robust least square with M-estimator 2/3



1. Set the scale parameter θ
2. Find the initial solution
3. Compute the residual for each sample $u(\mathbf{x}_i\,;\,\mathbf{p})$
4. Assign a weight for each sample based on the residual
5. Solve the weighted least square problem
6. Go to step3 or stop if the solution converges



1. Set the scale parameter θ
2. Find the initial solution
3. Compute the residual for each sample $u(\mathbf{x}_i\,;\,\mathbf{p})$
4. Assign a weight for each sample based on the residual
5. Solve the weighted least square problem
6. Go to step3 or stop if the solution converges

RANSAC

RANSAC

RANSAC: RANdom SAmple Consensus

- Handles a large number of outliers
- Reduces model estimation time
- Needs iterative procedure: hypothesizing and testing
- Is a sort of voting scheme

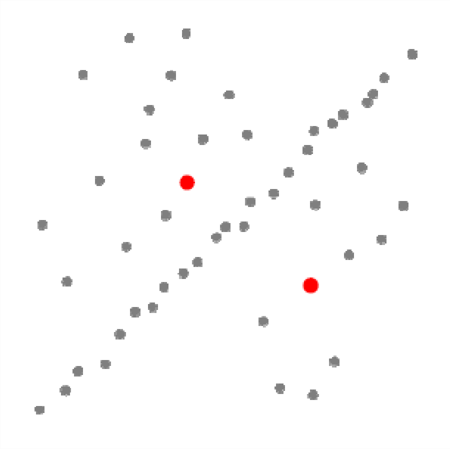
Assumptions

- Few outliers: noise features will not be consistent to vote any single model
- Few missing data: there are sufficient data to agree on a good model

RANSAC procedure 1/8

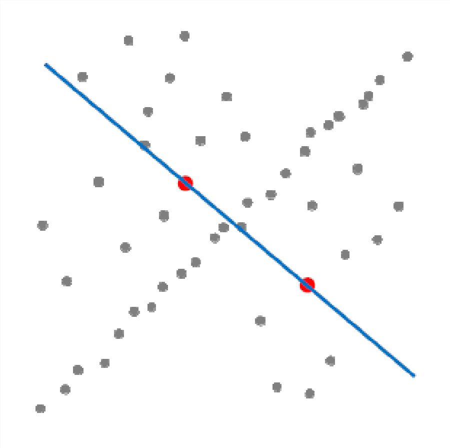
1. Given data

RANSAC procedure 2/8

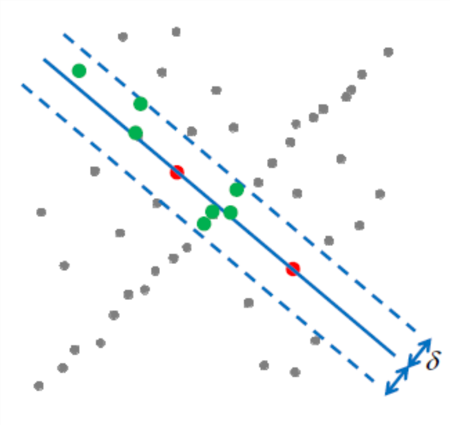


1. Given data
2. Randomly select minimal subset of points

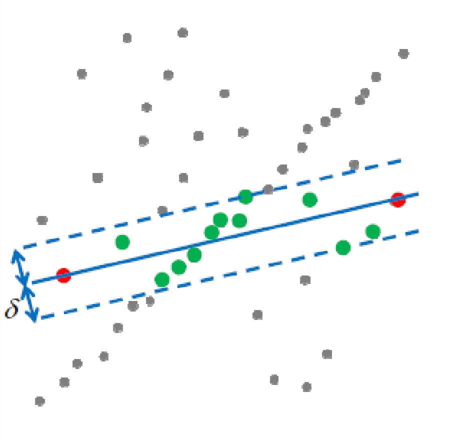
RANSAC procedure 3/8



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

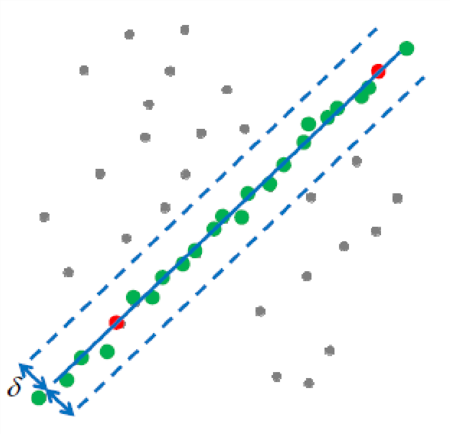


1. Given data
2. Randomly select minimal subset of points
3. Hypothesize a model
4. Select samples consistent with the model, and compute the ratio



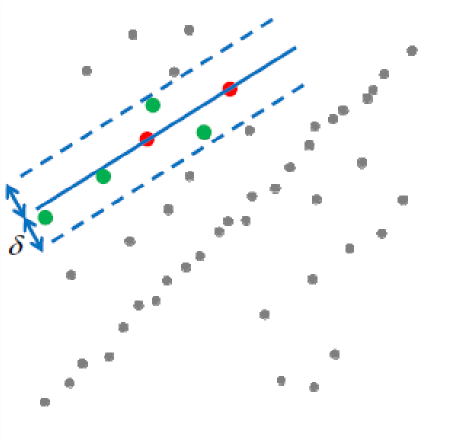
1. Given data
2. Randomly select minimal subset of points
3. Hypothesize a model
4. Select samples consistent with the model, and compute the ratio
5. Repeat the hypothesizing and testing procedure for the predefined number of iterations

RANSAC procedure 6/8



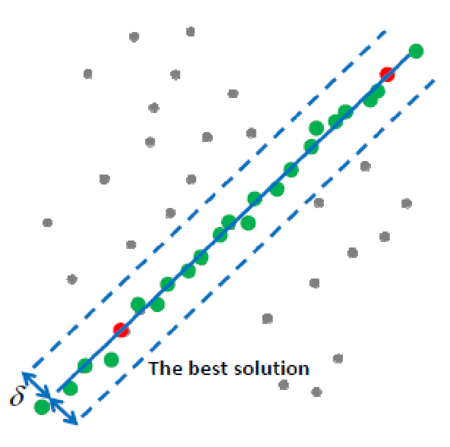
1. Given data
2. Randomly select minimal subset of points
3. Hypothesize a model
4. Select samples consistent with the model, and compute the ratio
5. Repeat the hypothesizing and testing procedure for the predefined number of iterations

RANSAC procedure 7/8



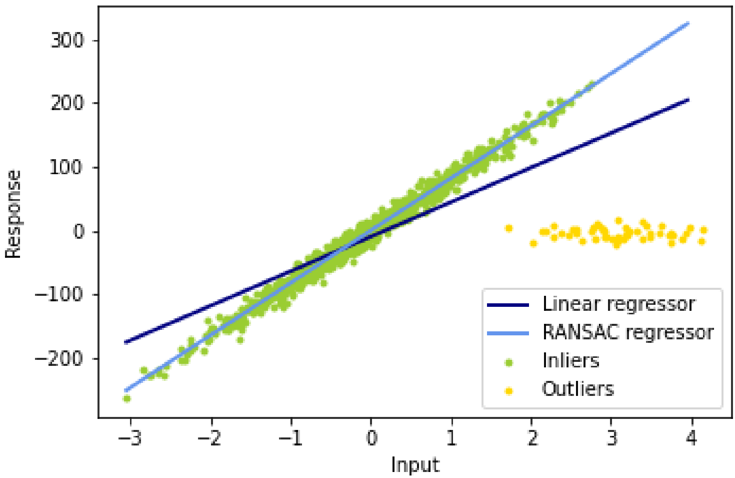
1. Given data
2. Randomly select minimal subset of points
3. Hypothesize a model
4. Select samples consistent with the model, and compute the ratio
5. Repeat the hypothesizing and testing procedure for the predefined number of iterations

RANSAC procedure 8/8



1. Given data
2. Randomly select minimal subset of points
3. Hypothesize a model
4. Select samples consistent with the model, and compute the ratio
5. Repeat the hypothesizing and testing procedure for the predefined number of iterations
6. Choose the best solution, and compute the model parameters

RANSAC example



Parameters for RANSAC

- Number of samples for building a hypothesis, s
  - Typically, the minimum number needed to fit the model
- Number of iterations, N
  - More iterations are required generally when outlier ratio is high, but does not mean the better solution necessarily
  - It depends on the probability of noise-free parameter estimation, p, with outlier ratio, e

$$\left(1-(1-e)^s\right)^N = 1-p \tag{4}$$

$$N = \log(1-p)/\log\left(1-(1-e)^s\right) \tag{5}$$

- Distance threshold, δ
- Consensus set size
  - How much inliers in percentage is sufficient to accept the model?

Characteristics of RANSAC

Pros

- Simple and general
- Often works well in practice

Cons

- Lots of parameters to tune
- Does not work well for low inlier ratios
  - Too many iterations
  - Failing completely
- Can't always get a good initialization of the model based on the minimum number of samples
- Does not guarantee the globally optimal solution

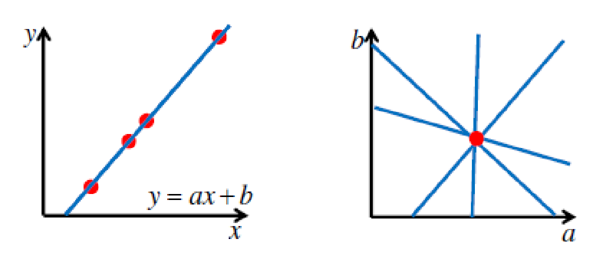
However, RANSAC is a good robust parameter estimation technique, and widely used in many computer vision problems.

Hough transform

Hough transform

Voting for parameter estimation

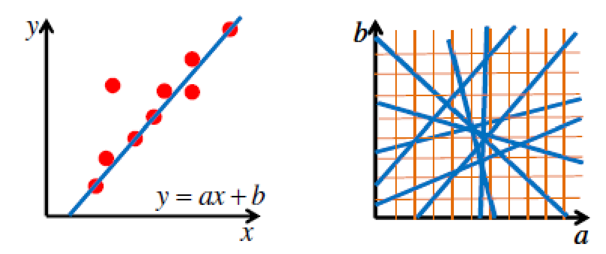
- An early type of voting scheme
- Convert from image space to parameter space



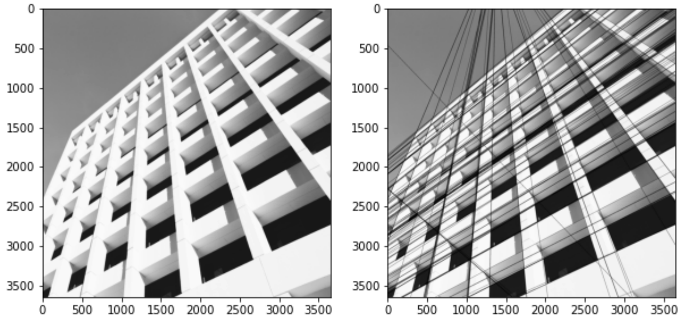
Parameter estimation by Hough transform

Procedure

- Discretize parameter space into bins
- For each feature point in the image, put a vote in every bin in the parameter space that could have generated this point
- Find bins that have the most votes



Hough transform example



Characteristics of Hough transform

Assumptions

- The noise features will not vote consistently for any single model.
- There are enough data to agree on a good model.

Pros

- Can deal with non-locality and missing data
- Can detect multiple instances of a model
- Some robustness to noise: noise points unlikely to contribute consistently to any single bin

Cons

- Complexity of search time increases exponentially with the number of model parameters: more than 2D is not practical.
- It's hard to pick a good grid size.

Appendix

Reference and further reading

- “Lecture10 | Robust Fitting and Matching” of Bohyung Han, CSED441: Introduction to Computer Vision, POSTECH (2011)
- “Robust linear model estimation using RANSAC” of scikit-learn (link)
- “Hough line transform” of OpenCV-Python Tutorials (link)