DMIP - Exercise:

RANSAC

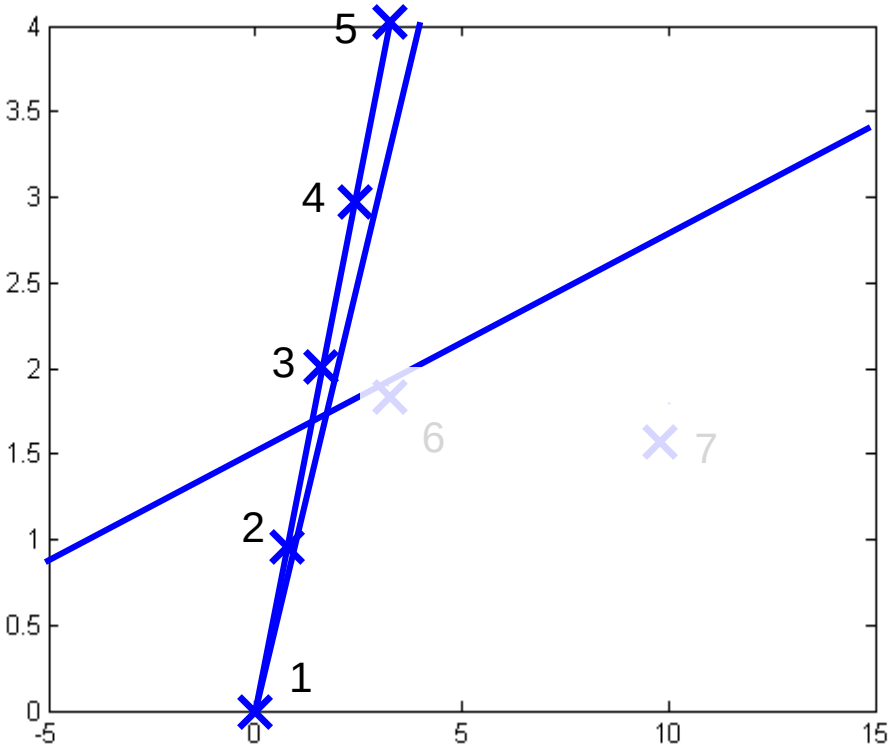
Marco Bögel, Bastian Bier
Pattern Recognition Lab (CS 5)

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Problem in calibration: inaccuracies in observations and outliers.

- Badly localized points (noise)
- Wrong correspondence

Linear Regression



RANSAC – RANdom Sample Consensus

RANSAC assumes that a model built with a minimum number of data points **does not contain outliers**.

Algorithm:

- Determine the minimum number n_{mdl} of data points required to build the model
  → A line is completely defined by two points → $n_{mdl} = 2$

- For n_{it} iterations do
  - a) Choose randomly n_{mdl} points out of your data to estimate the model
  - b) Determine the error of the current model using all data points

- Choose model with lowest error

RANSAC

Find the line parameter *m* and *t,* so that all points (x_i, y_i), i = 1,…,7, approximately fulfill the line equation

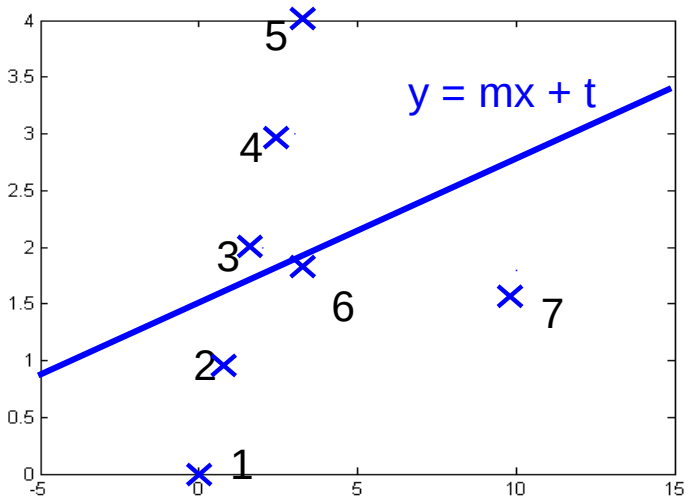
→ Solve the following optimization problem

$$y_i = mx_i + t$$

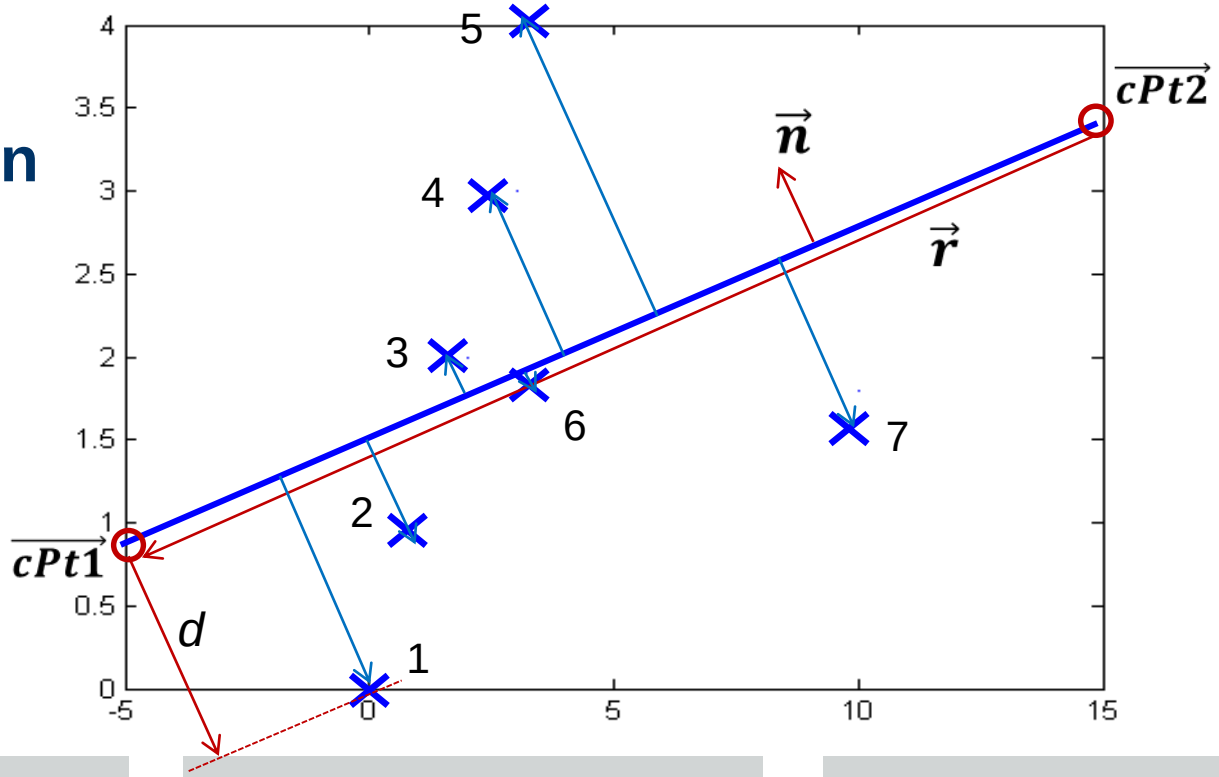
$$\left\| [X\ 1] \cdot \begin{pmatrix} m \\ t \end{pmatrix} - Y \right\| = \left\| M \cdot \begin{pmatrix} m \\ t \end{pmatrix} - Y \right\| \to 0$$

The least square solution of this equation is given (**Moore-Penrose pseudo-inverse)**

$$\begin{pmatrix} m \\ t \end{pmatrix} = M^{\dagger} Y$$



Error Function



1) Calculate the normal vector of the line

$$\vec{n}$$

2) Distance to origin is given by the scalar product of some point on the line and the normal:

$$d = c\vec{P}t1^{T} \cdot \vec{n}$$

3) Calculate distance from each point to the line

$$ds = p\vec{t}s_i^{T} \cdot \vec{n} - d$$

RANSAC

Number of iterations

Probability for an outlier

p_o

RANSAC

Number of iterations

Probability for an outlier

p_o

Probability for not having outliers in the minimum number of points required to build the model

$(1 - p_o)^{n_{mdl}}$

RANSAC

Number of iterations

Probability for an outlier

p_o

Probability for not having outliers in the minimum number of points required to build the model

$(1 - p_o)^{n_{mdl}}$

Probability of having at least one outlier in the minimum number of points for given iterations

$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}}$

RANSAC

Number of iterations

Probability for an outlier

p_o

Probability for not having outliers in the minimum number of points required to build the model

$(1 - p_o)^{n_{mdl}}$

Probability of having at least one outlier in the minimum number of points for given iterations

$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}}$

This should not be higher than a given probability

$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}} \leq 1 - P_{corr}$

RANSAC

Number of iterations

Probability for an outlier

$$p_o$$

Probability for not having outliers in the minimum number of points required to build the model

$$(1 - p_o)^{n_{mdl}}$$

Probability of having at least one outlier in the minimum number of points for given iterations

$$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}}$$

This should not be higher than a given probability

$$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}} \leq 1 - P_{corr}$$

$$\Rightarrow n_{it} = \left\lceil \frac{log(1 - P_{corr})}{log(1 - (1 - p_o)^{n_{mdl}})} \right\rceil$$

RANSAC

Number of iterations

Probability for an outlier

p_o

Probability for not having outliers in the minimum number of points required to build the model

$(1 - p_o)^{n_{mdl}}$

Probability of having at least one outlier in the minimum number of points for given iterations

$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}}$

This should not be higher than a given probability

$(1 - (1 - p_o)^{n_{mdl}})^{n_{it}} \leq 1 - P_{corr}$

$$\Rightarrow n_{it} = \left\lceil \frac{log(1 - P_{corr})}{log(1 - (1 - p_o)^{n_{mdl}})} \right\rceil$$

Estimate probability for an outlier using relative frequencies. Minimum number of points for the model is given.

→ Choose probability for having at least one iteration without outliers

RANSAC

Task: `commonransac`: In `it` iterations choose randomly `mn` points out of `points`. Use them to estimate the model. Estimate the error for this model.

For each iteration, do

1. Randomly choose `mn` points from data
2. Use them to estimate the model: fitline(…)
3. Compute the error for this model: lineError(…)