Robot Vision:
Robust Estimation

Prof. Friedrich Fraundorfer

SS 2023

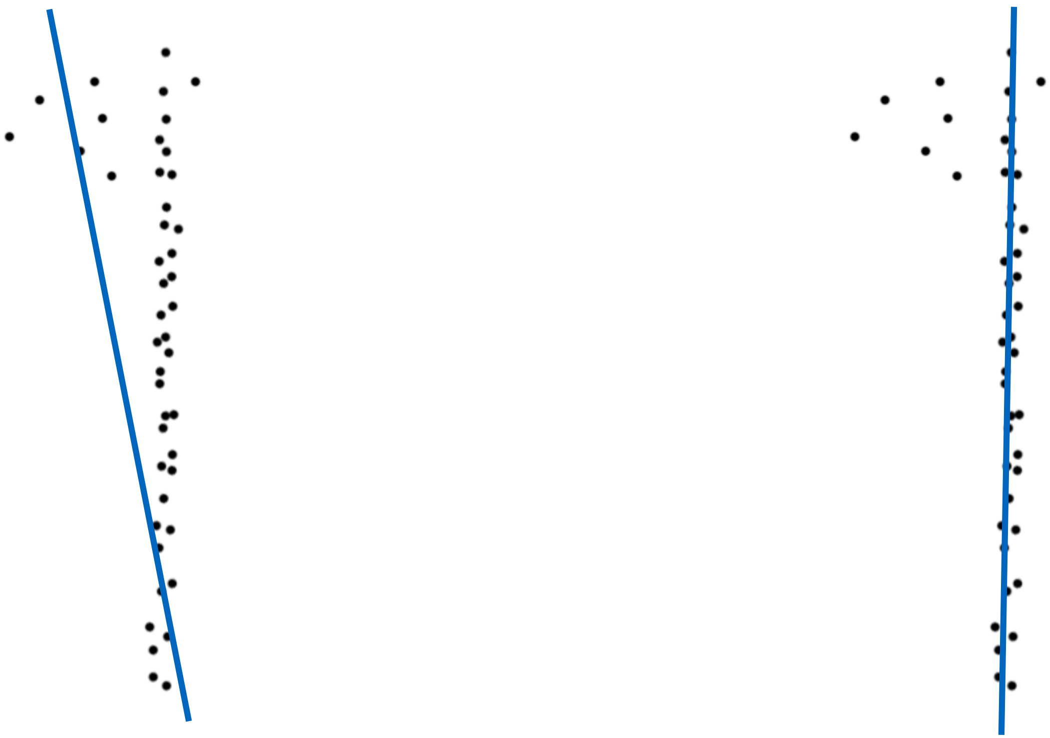
Outline

- Ransac concept
- Ransac for fundamental matrix estimation
- Robust optimization (robust cost functions)

Learning goals

- Understand the Ransac concept
- Be able to apply the equation for Ransac iterations
- Understand the use of Ransac for fundamental matrix estimation
- Understand the problem of least squares estimation with outliers
- Understand how robust cost functions work

Robust estimation

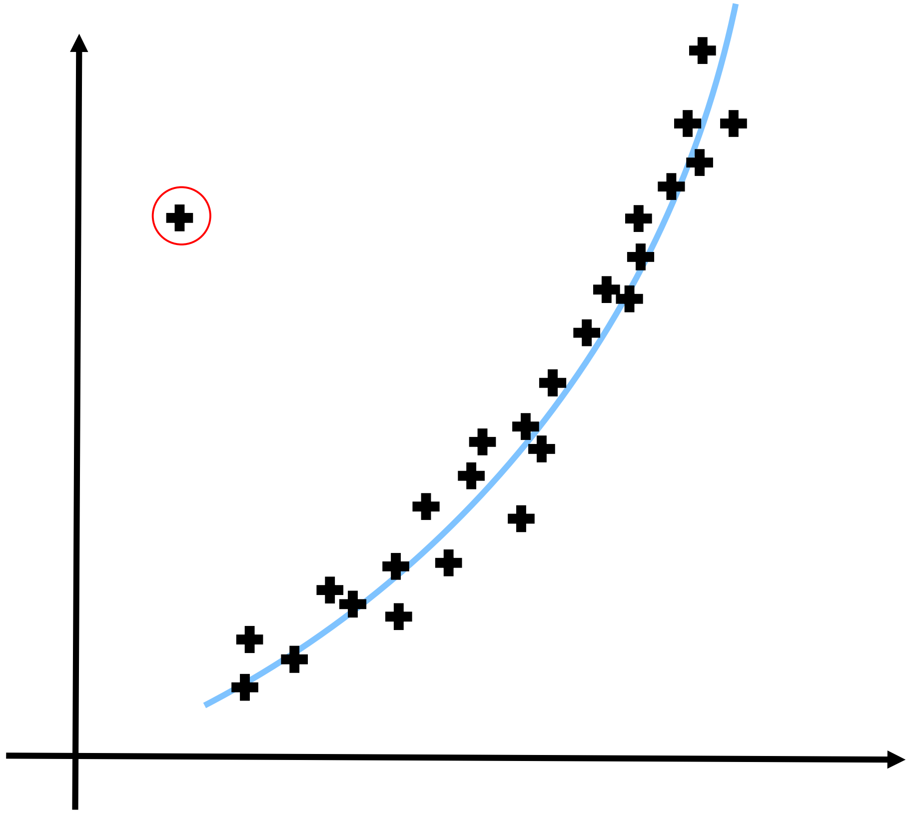


Non-robust line estimate   Robust line estimate

- In computer vision the analogy to the line would be the fundamental matrix

Robust estimation in computer vision

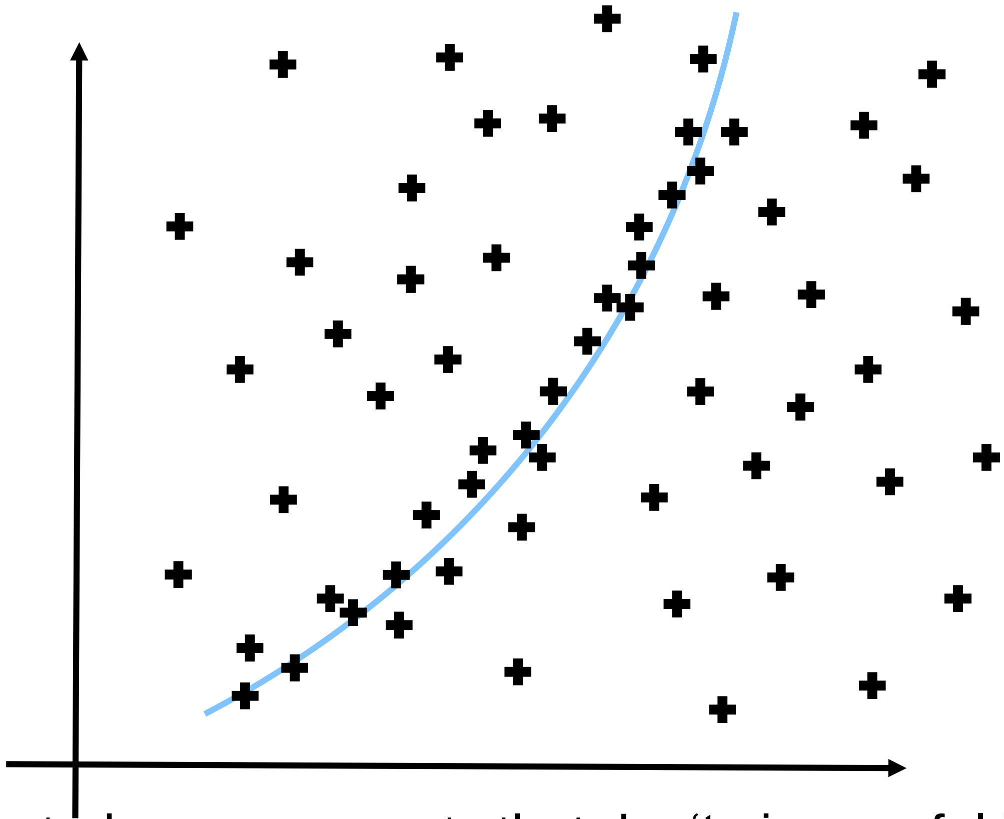
- Outliers in general estimation



- Rare unexpected measurements that don‘t fit the model

Robust estimation in computer vision

- Outliers in computer vision



- Frequent expected measurements that don‘t give useful information

Robust estimation in computer vision

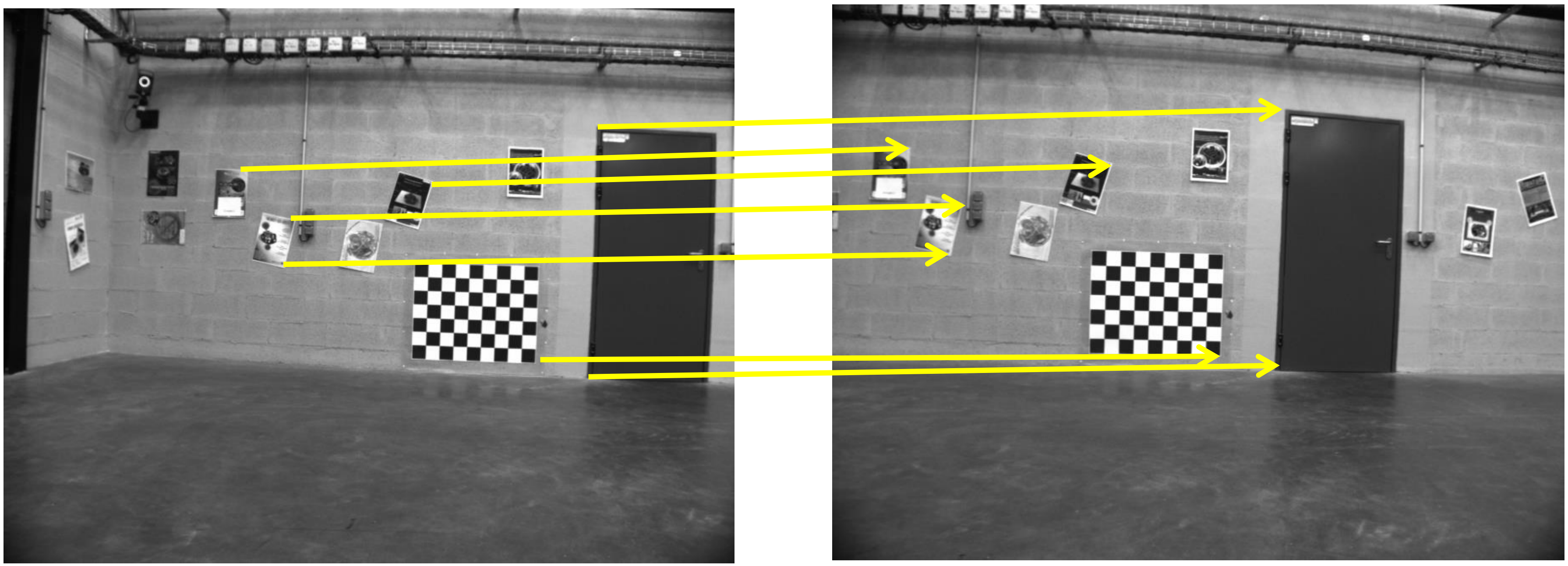
- Multiple areas in computer vision require robust estimation techniques

  - Fundamental matrix estimation
  - Essential matrix estimation
  - Camera pose estimation
  - Camera calibration
  - Triangulation
  - Bundle adjustment
  - 2D Registration
  - 3D Registration
  - 3D Model fitting
  - Line detection

- Techniques:
  - Ransac
  - Robust loss functions for optimization

Camera motion estimation

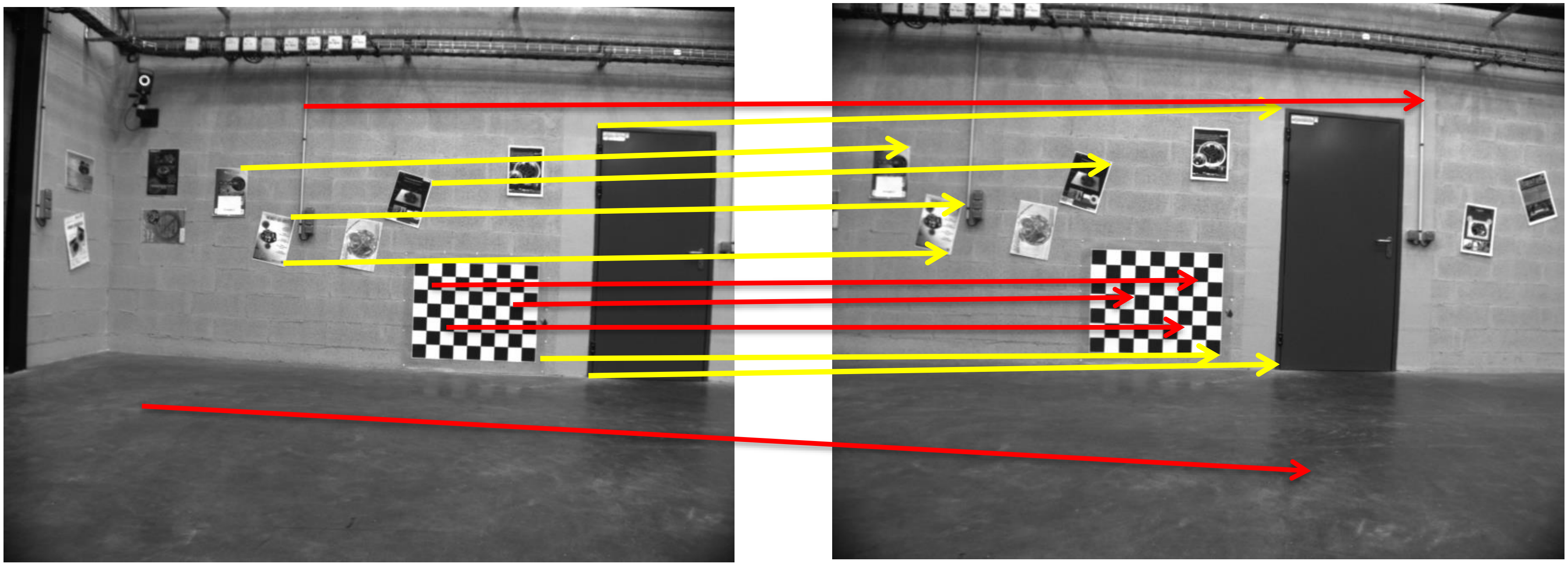
$$x'^T \boldsymbol{E} x = 0$$



x image matches x'

Camera motion estimation

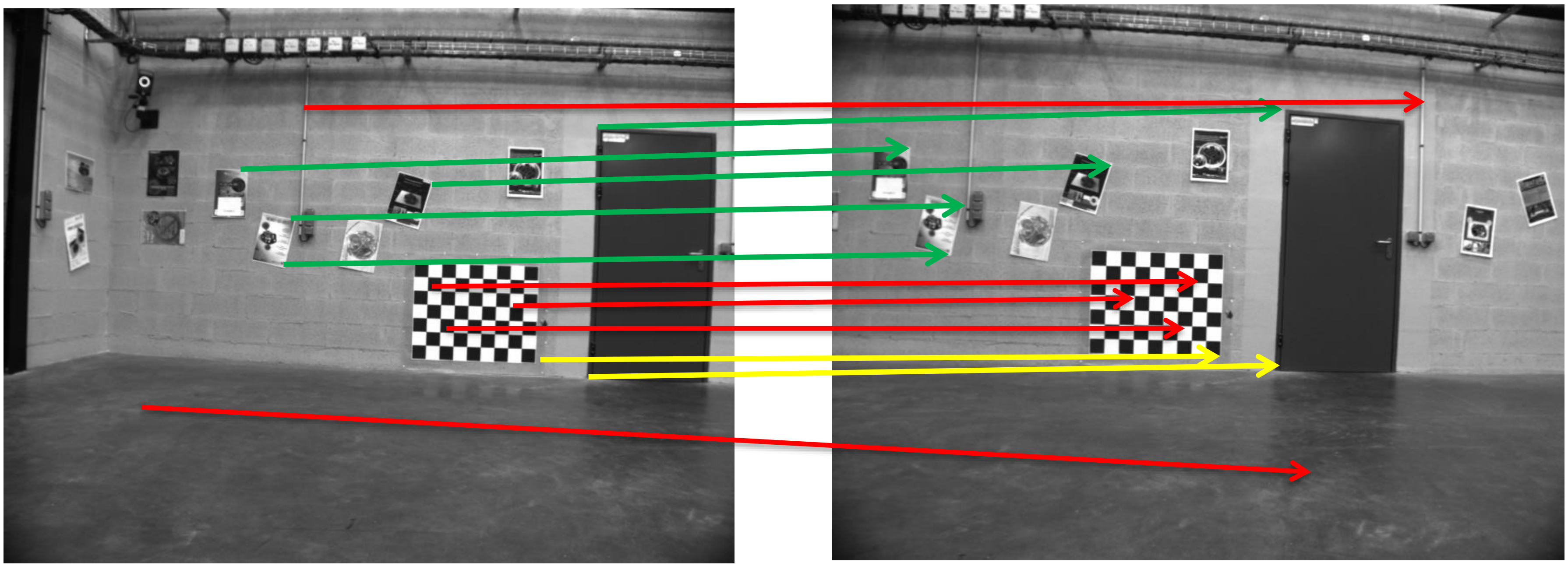
$$x'^{T}\boldsymbol{E}x = 0$$



x image matches with mis-matches x'

Camera motion estimation

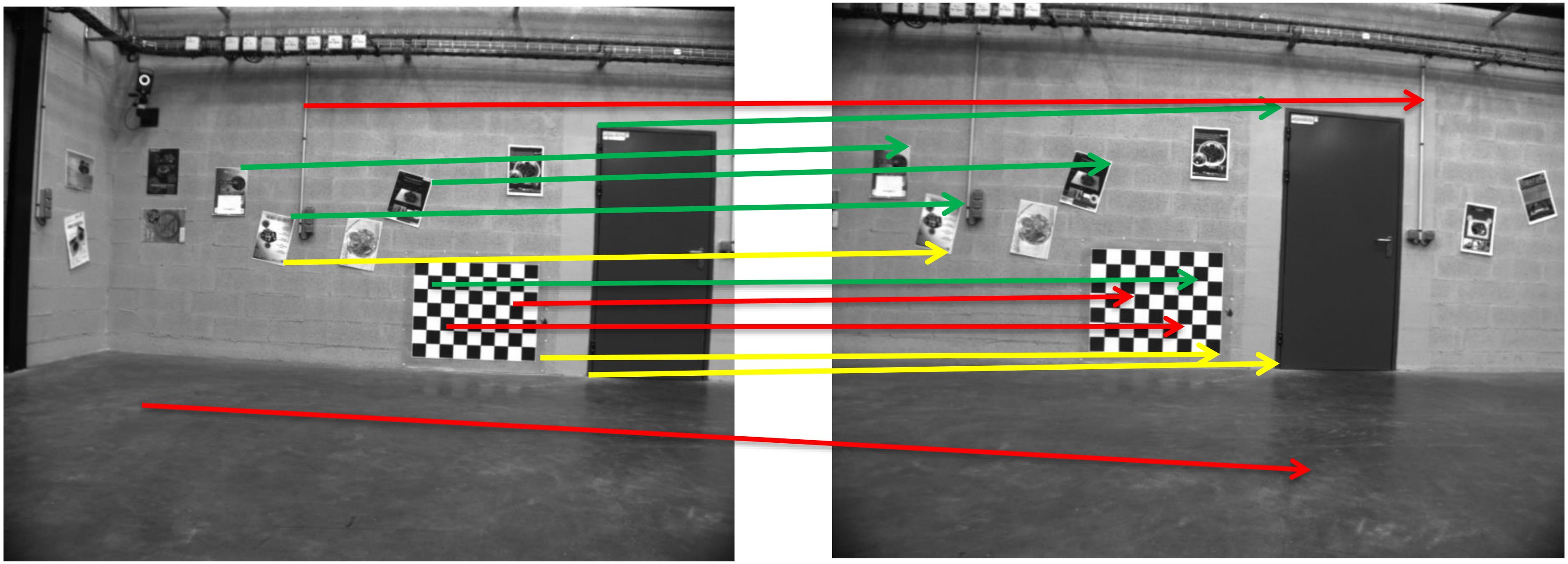
$$x'^{T}\boldsymbol{E}x = 0$$



x image matches with mis-matches x'

Camera motion estimation

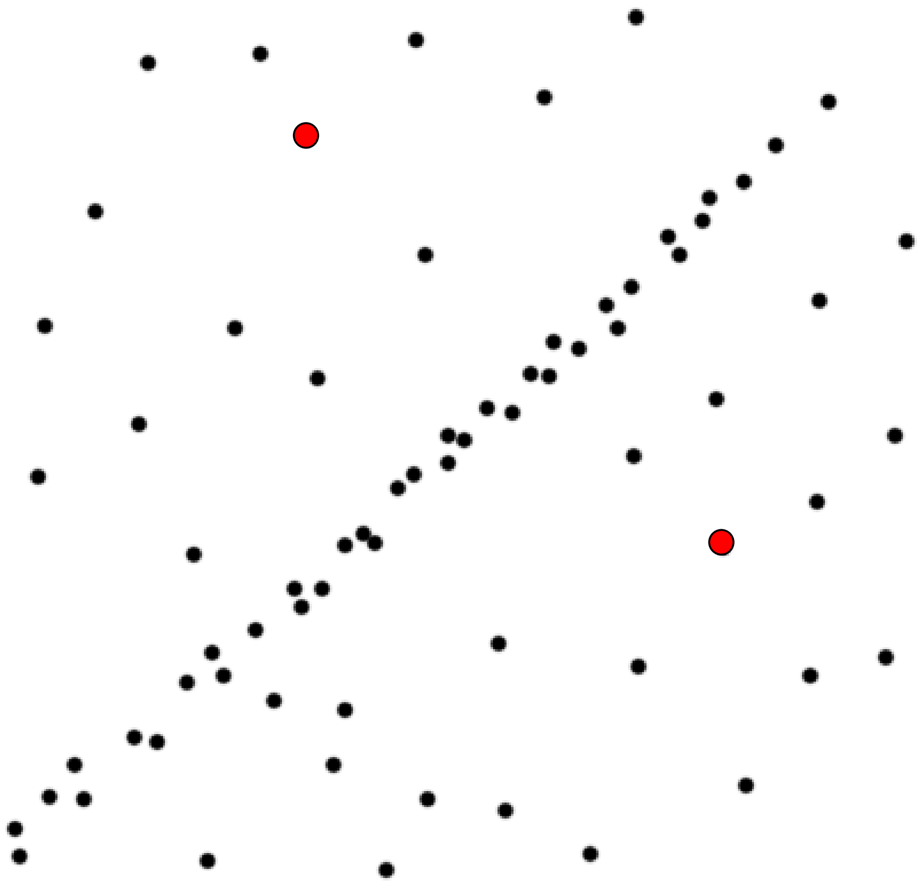
$$x'^{T}\boldsymbol{E}x = 0$$



x image matches with mis-matches x'

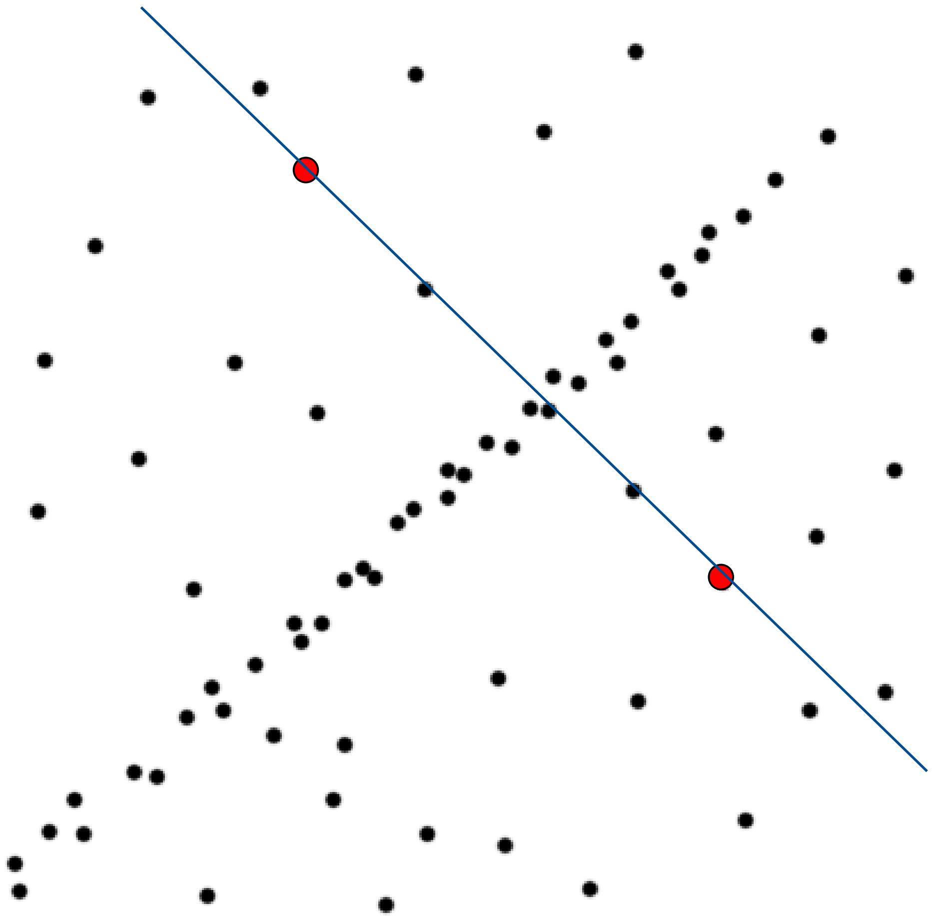
RANSAC Example: Line extraction

RANSAC Example: Line extraction



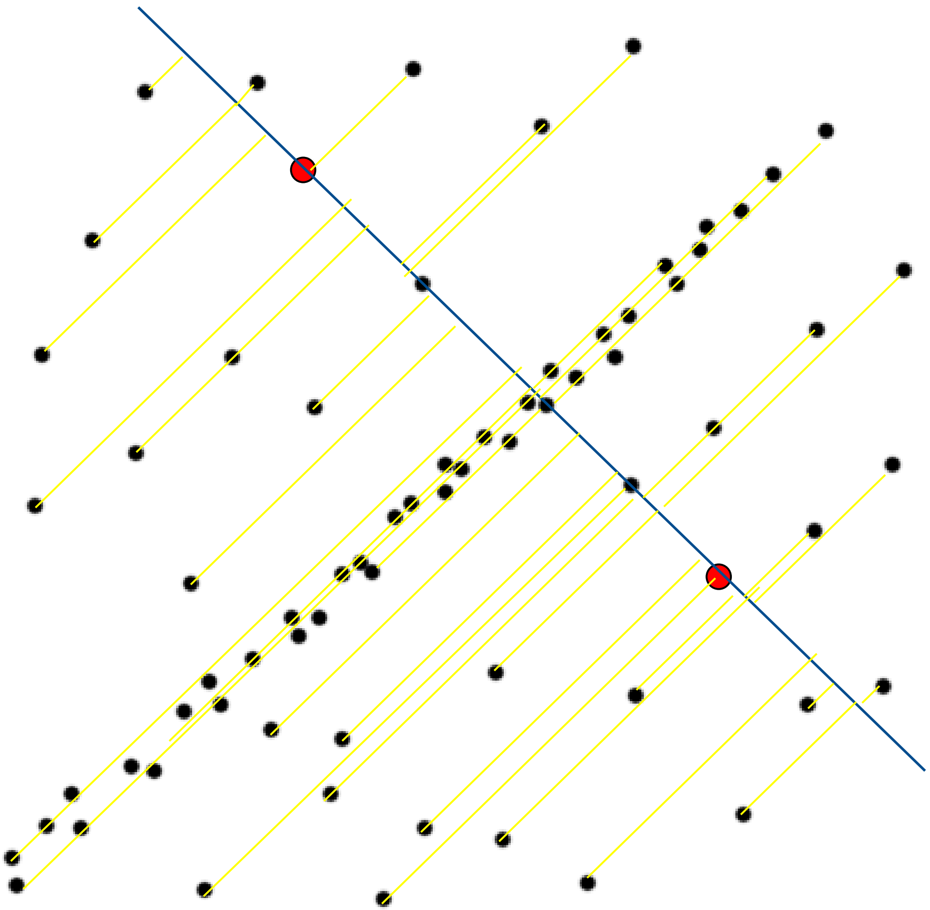
- Random sampling of 2 points

RANSAC Example: Line extraction



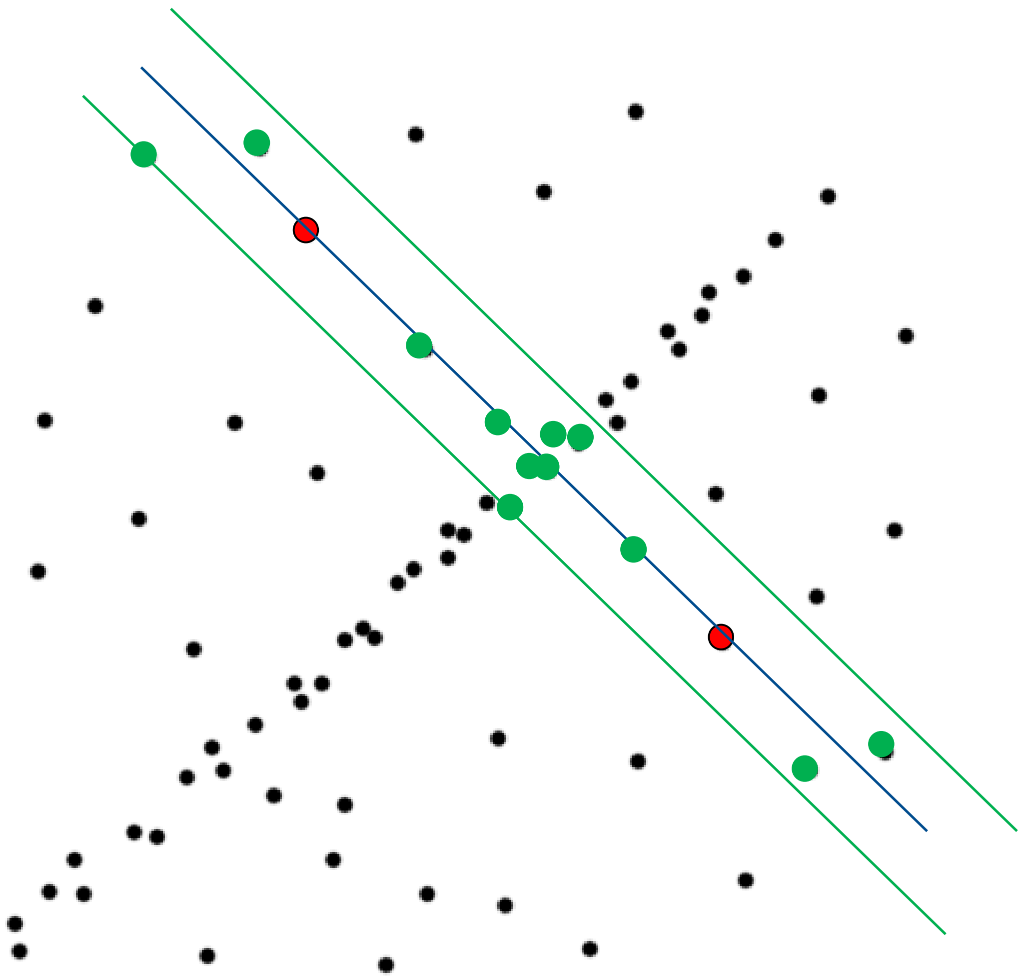
- Random sampling of 2 points
- Calculate line model from this 2 data points

RANSAC Example: Line extraction



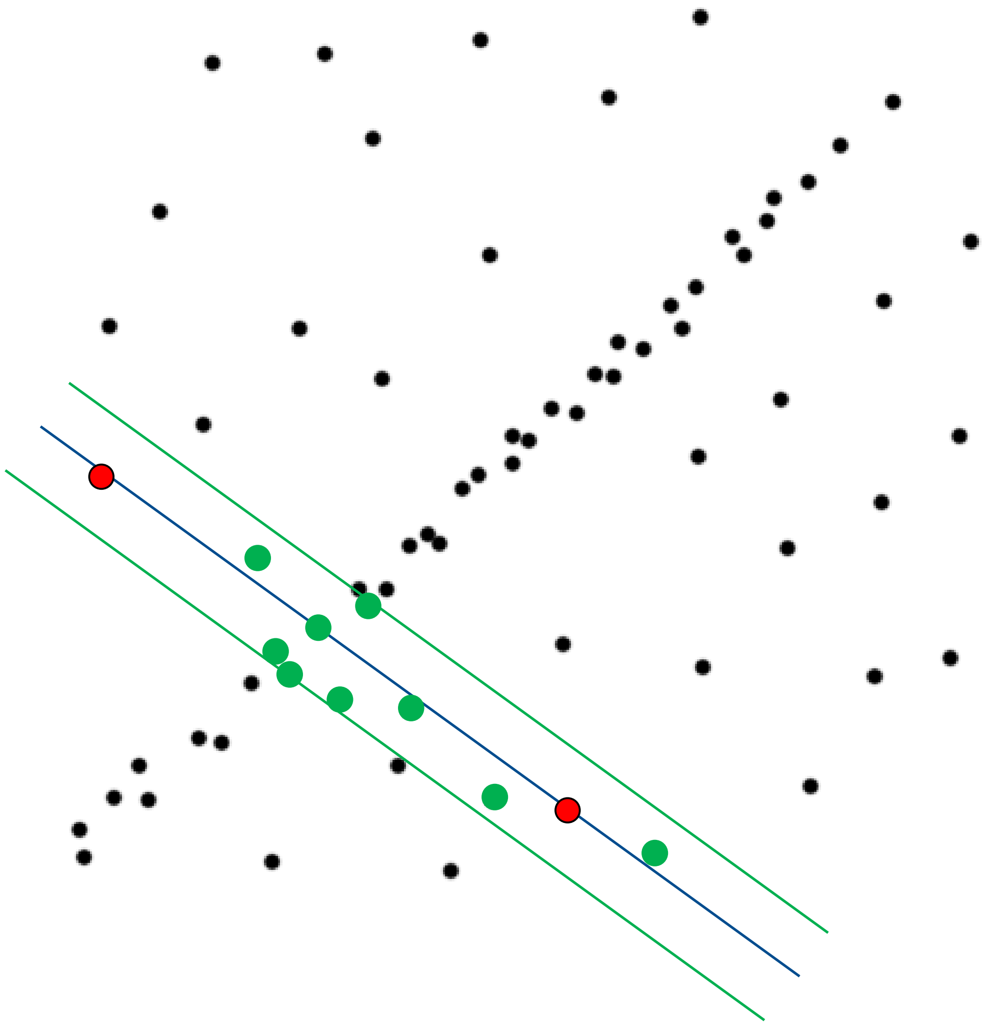
- Random sampling of 2 points
- Calculate line model from this 2 data points
- Calculate residual error for each data point (normal distance to line)

RANSAC Example: Line extraction



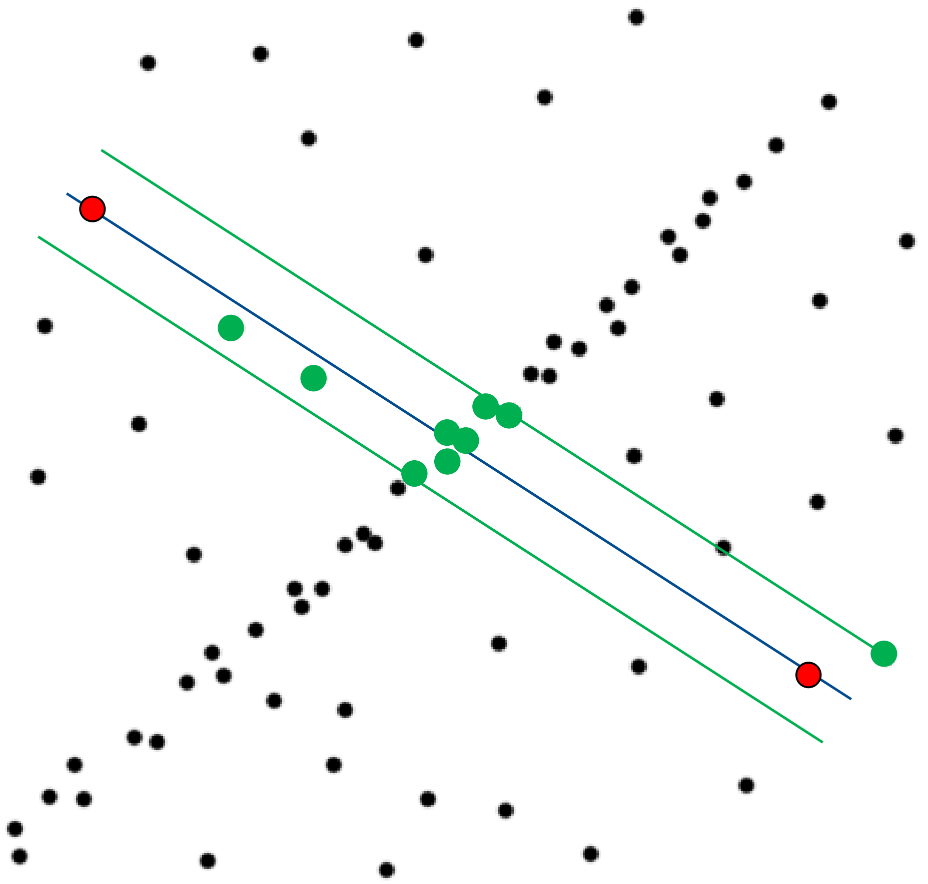
- Random sampling of 2 points
- Calculate line model from this 2 data points
- Calculate residual error for each data point (normal distance to line)
- Select data points that support current hypothesis

RANSAC Example: Line extraction



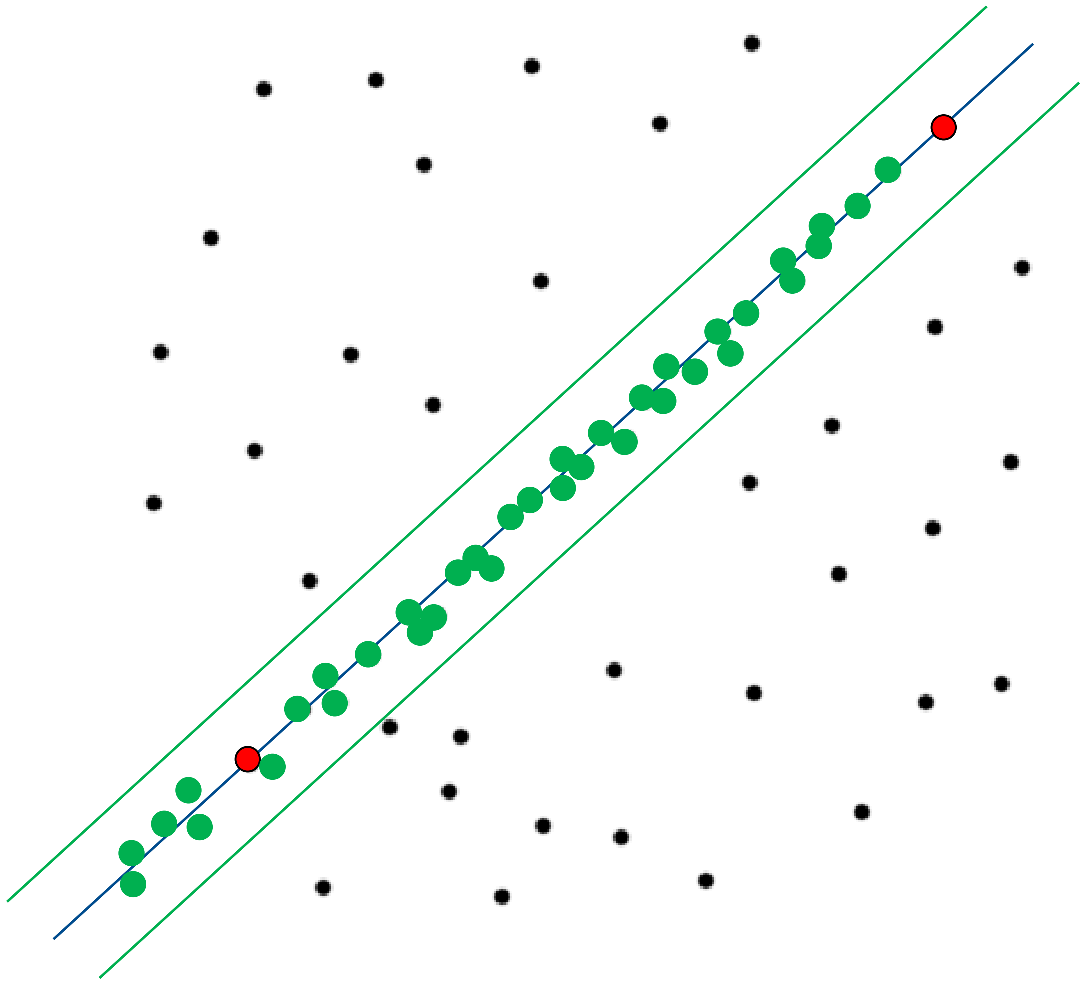
- Random sampling of 2 points
- Calculate line model from this 2 data points
- Calculate residual error for each data point (normal distance to line)
- Select data points that support current hypothesis
- Repeat sampling

RANSAC Example: Line extraction



- Random sampling of 2 points
- Calculate line model from this 2 data points
- Calculate residual error for each data point (normal distance to line)
- Select data points that support current hypothesis
- Repeat sampling

RANSAC Example: Line extraction



- Random sampling of 2 points
- Calculate line model from this 2 data points
- Calculate residual error for each data point (normal distance to line)
- Select data points that support current hypothesis
- Repeat sampling
- Until a hypothesis with a maximum number of inliers has been found

RANSAC Algorithm

1. Initial: Let A be a set of N data points
2. Repeat
    1. Randomly select a sample of s data points from A
    2. Fit a model to these points
    3. Compute the distance of all other points to this model
    4. Construct the inlier set (i.e. count the number of data points whose distance from the model < threshold d
    5. Store these inliers
3. Until maximum number of iterations reached
4. The model with the maximum number of inliers is chosen as the solution to the problem
5. Re-estimate the model using all the inliers

How many iterations of RANSAC ?

- The number of iterations N which is necessary to guarantee that at least a single correct solution is found can be computed by

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)}$$

- s is the number of data points from which a model can be minimally computed
- ε is the percentage of outliers in the data (can only be guessed)
- p is the requested probability of success

- Example: p = 0.99, s = 5, ε = 0.5 -> N = 145

How many iterations of RANSAC ?

- RANSAC is an iterative method and it is non deterministic. It returns different solutions on different runs.

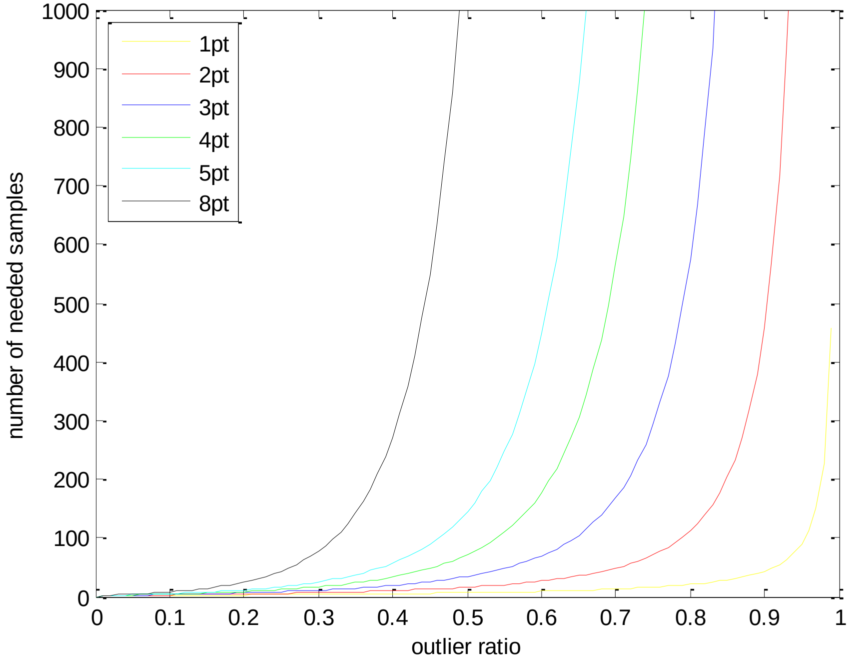
- For reasons of reliability, in many practical implementations N is usually multiplied by a factor of 10

- More advanced implementations of RANSAC estimate the fraction of inliers adaptively, for every iteration, and use it to update N.

RANSAC iterations

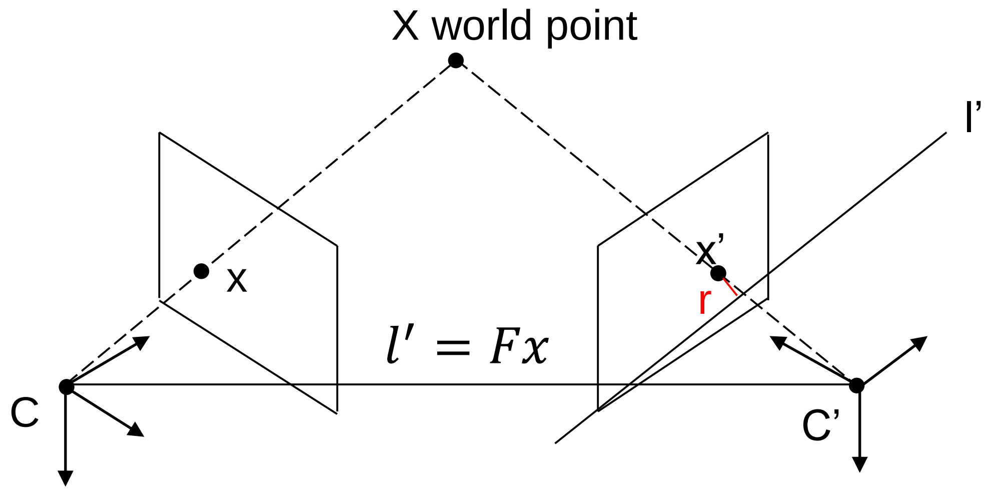
- N is exponential in the number of data points s necessary to estimate the model
- Therefore, it is very important to use a minimal parameterization of the model

Number of points (s) p=0.99, e=0.5	8	7	6	5	4	2	1
Number of iterations (N)	1177	587	292	145	71	16	7



Ransac for fundamental matrix estimation

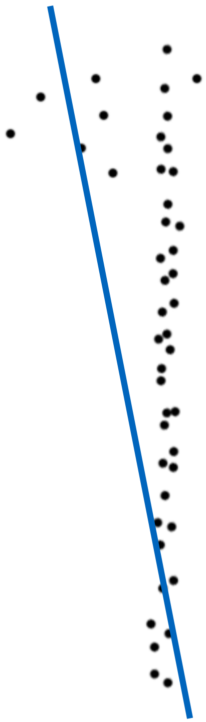
- Inlier set for fundamental matrix estimation is set of correct image matches
- Cost function for Ransac: Epipolar distance

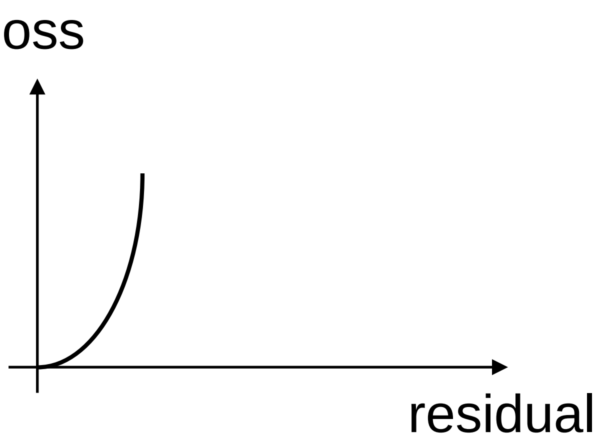


$$r_i = d(x_i', Fx_i)^2 + d(x_i\ , F^T x_i')^2$$
$$= \left(x_i'^T F x_i\right)^2 \left(\frac{1}{(Fx_i)_1^2 + (Fx_i)_2^2} + \frac{1}{\left(F^T x_i'\right)_1^2 + \left(F^T x_i'\right)_2^2} \right)$$

Why is least-squares not robust to outliers?

$$Ax = b \rightarrow \min_x \|Ax - b\|$$
$$x = (A^T A)^{-1} A^T b$$





$$\min_x \sum_i r_i^2(x)$$

- Least squares means quadratic loss function
- Robust estimation can be achieved by different loss functions than quadratic

Robust optimization

- Iteratively reweighted least squares (IRLS)
  - Estimates weights in every iteration to down-weigh outliers

$$\min_x \|w(Ax - b)\|$$

- Non-linear optimization with robust loss function
  - A robust loss function is used that down-weighs the influence of outliers

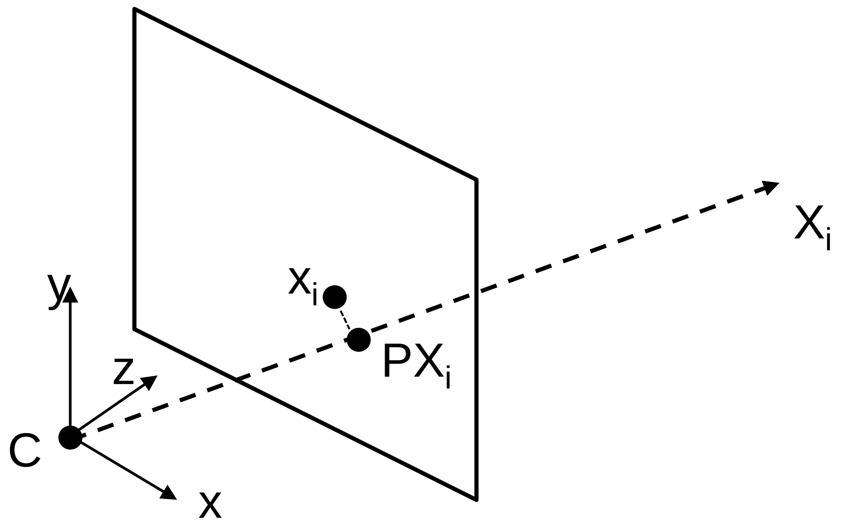
$$\min_x \sum_i \rho_i(\|f_i(x_i)\|^2)$$

$$\min_x \|A(x) - b\|$$

Example: Gold standard method for fundamental matrix

$$\min_{P,P',X_i} \sum_i d(x_i, PX_i)^2 + d(x'_i, P'X_i)^2$$

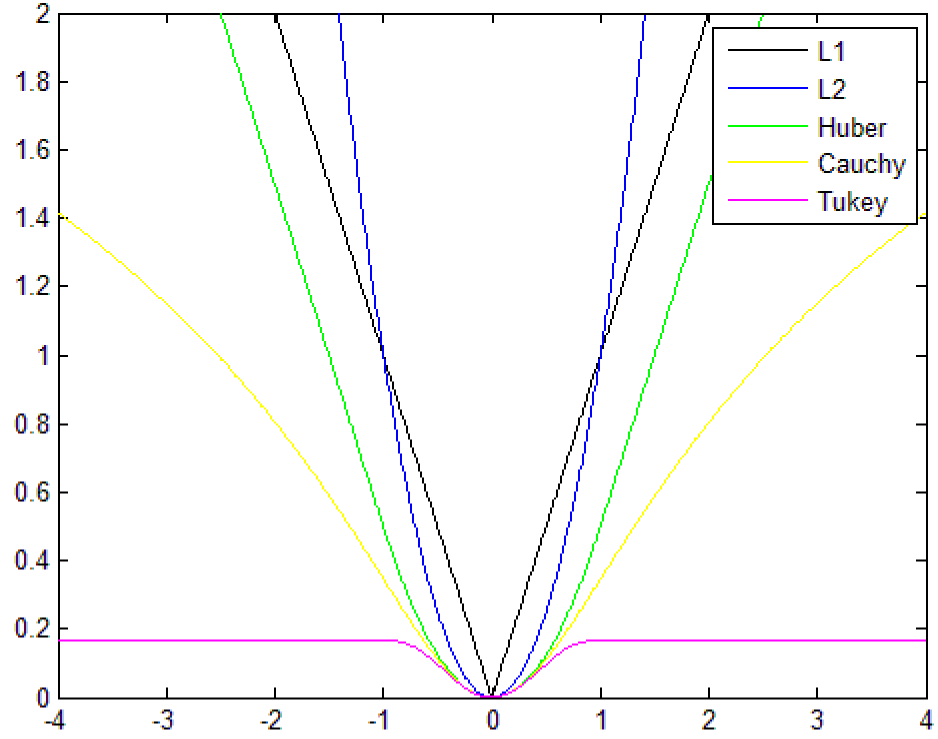
- x_i … image measurement
- P, P’ … camera matrices (will be optimized)
- X_i … 3D points (will be optimized)



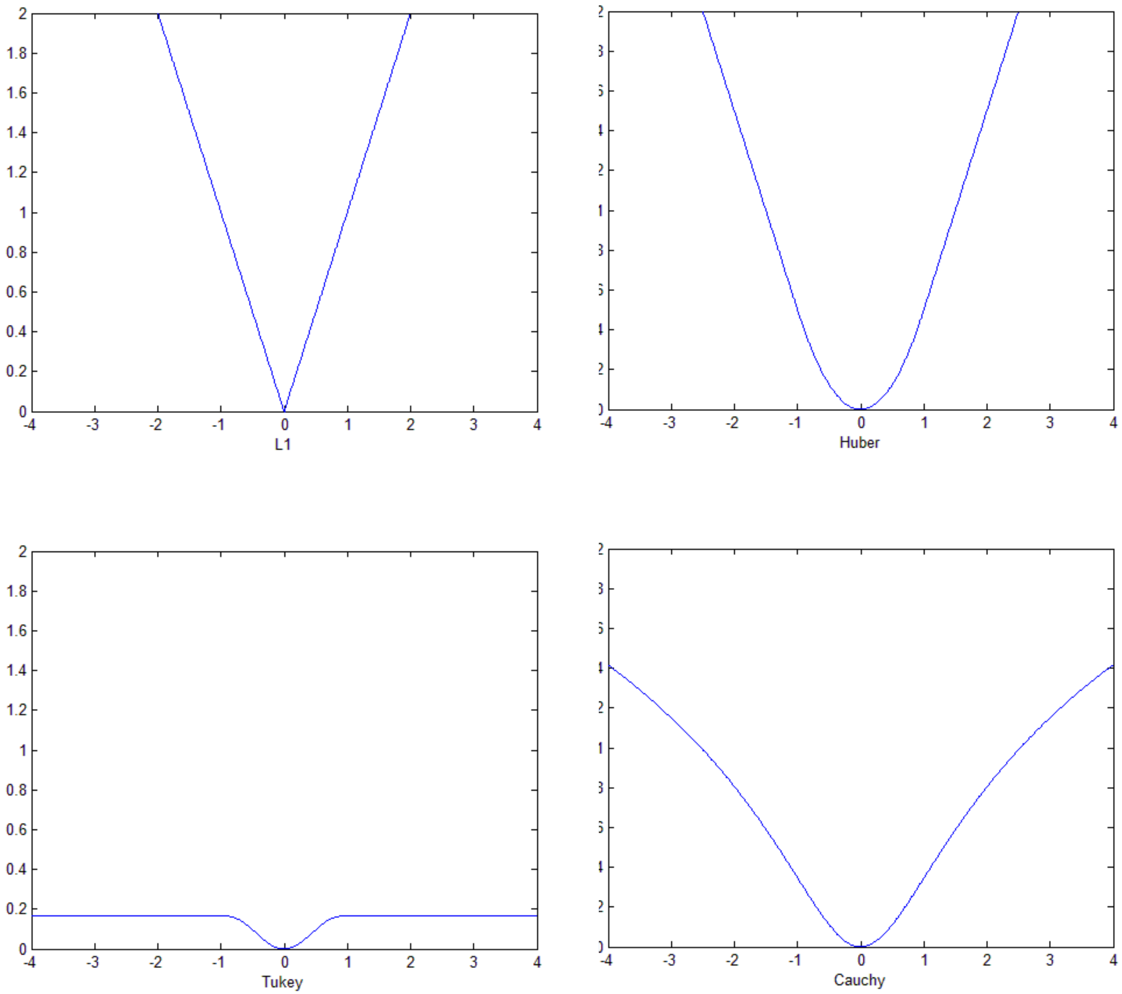
Non-linear estimation with robust loss function

$$\min_x \sum_i \rho_i(\|f_i(x_i)\|^2)$$

- Non-linear optimization (e.g. Levenberg-Marquard)
- Squared error will be modified by robust loss function $\rho(x)$
- Loss function should be differentiable
- Jacobian needs to be calculated



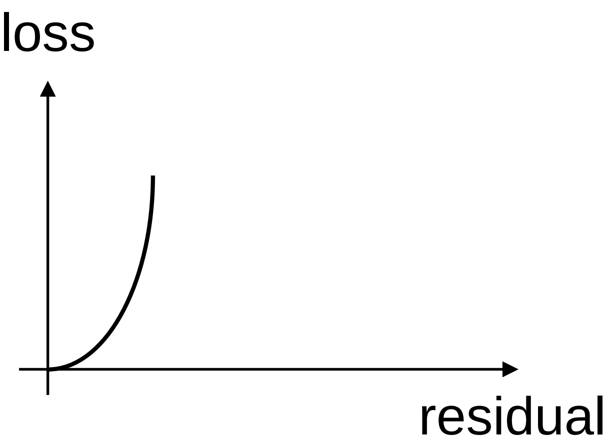
More robust loss functions



loss function	p(x)
L_1	$\lvert x\rvert$
$Huber \begin{cases} if\ \lvert x\rvert \le k \\ if\ \lvert x\rvert > k \end{cases}$	$\begin{cases} x^2/2 \\ k(\lvert x\rvert - \frac{k}{2}) \end{cases}$
$Tukey \begin{cases} if\ \lvert x\rvert \le k \\ if\ \lvert x\rvert > k \end{cases}$	$\begin{cases} k^2/6(1-\left(1-\left(\frac{x}{c}\right)^2\right)^3) \\ k^2/6 \end{cases}$
$Cauchy$	$\frac{k^2}{2}\log(1+(x/k)^2)$

Effects of different loss functions

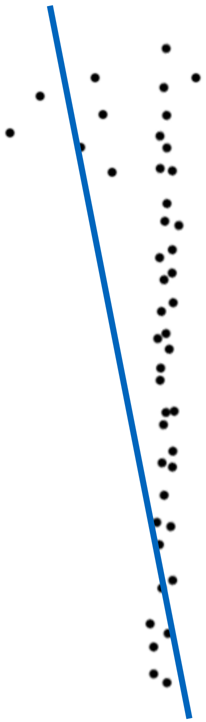
L_2 solution



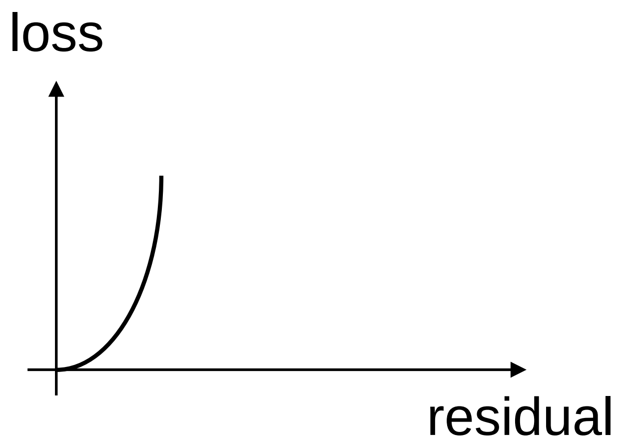
$$\min_{x} \sum_{i} r_i^2(x)$$

- L_2 (quadratic) loss works fine for outlier-free data

Effects of different loss functions



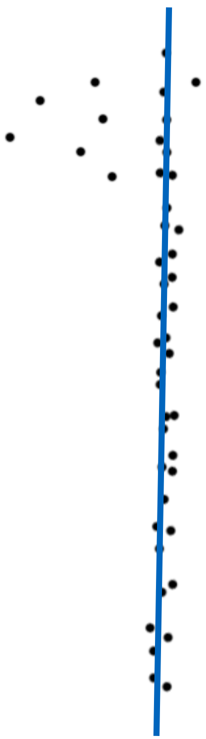
L_2 solution



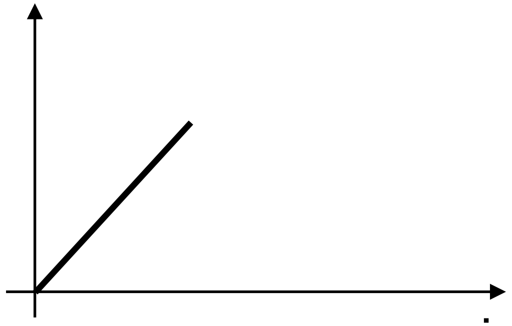
$$\min_x \sum_i r_i^2(x)$$

- Outliers lead to wrong estimate

Effects of different loss functions



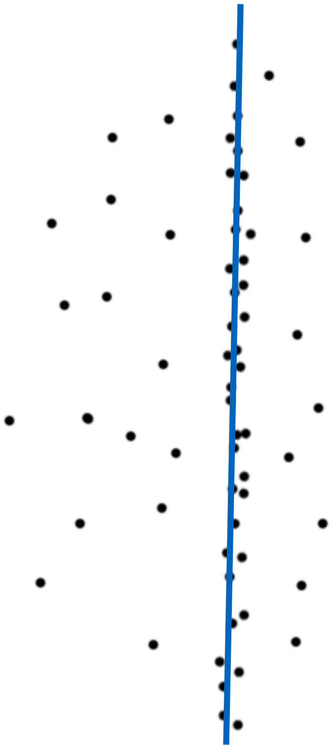
L_1 solution



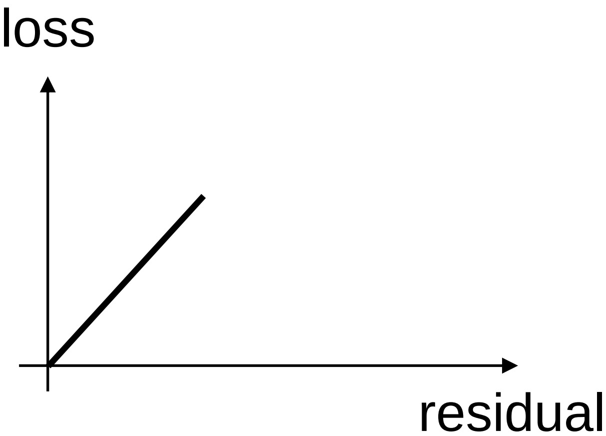
$$\min_x \sum_i |r_i(x)|$$

- L_1 loss leads to correct solution (estimation robust to outliers)

Effects of different loss functions



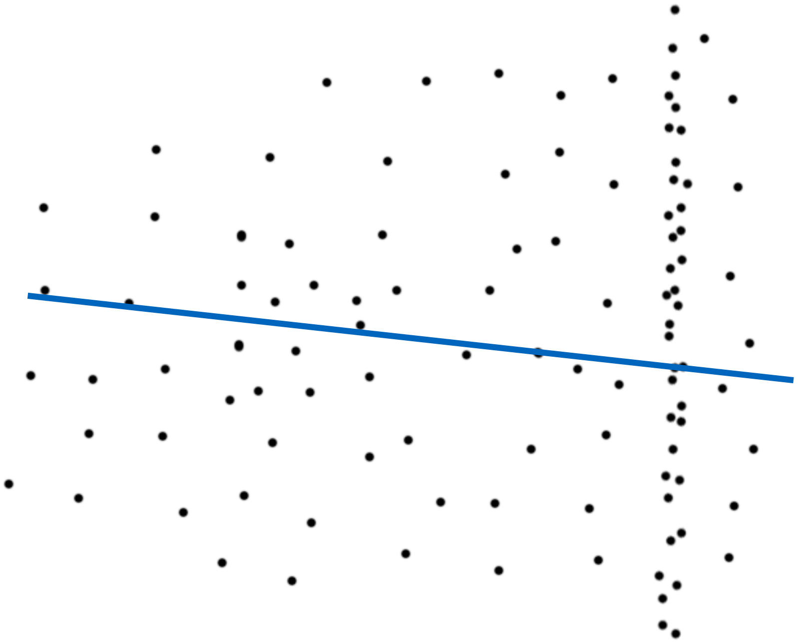
L_1 solution



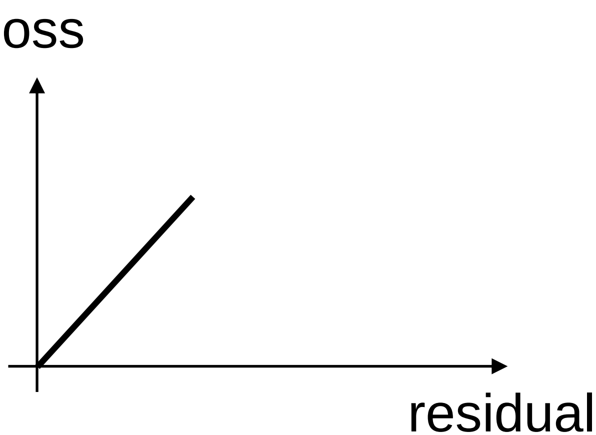
$$\min_x \sum_i |r_i(x)|$$

- Does work for larger number of outliers as well

Effects of different loss functions



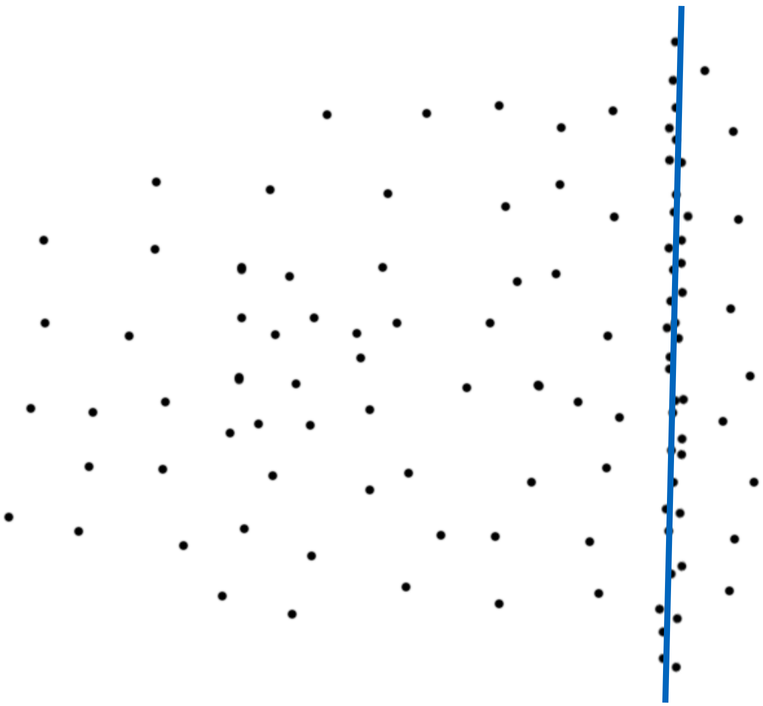
L_1 solution

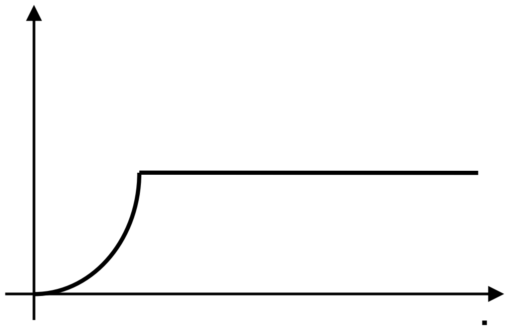


$$\min_x \sum_i |r_i(x)|$$

- Breaks down eventually (> 50% outliers)

Effects of different loss functions





$$\min_x \sum_i \min(r_i^2(x), t^2)$$

- However, can be solved with different cost function (e.g. Truncated L_2)
- However, not easy to predict this behaviour

Learning goals - Recap

- Understand the Ransac concept
- Be able to apply the equation for Ransac iterations
- Understand the use of Ransac for fundamental matrix estimation
- Understand the problem of least squares estimation with outliers
- Understand how robust cost functions work