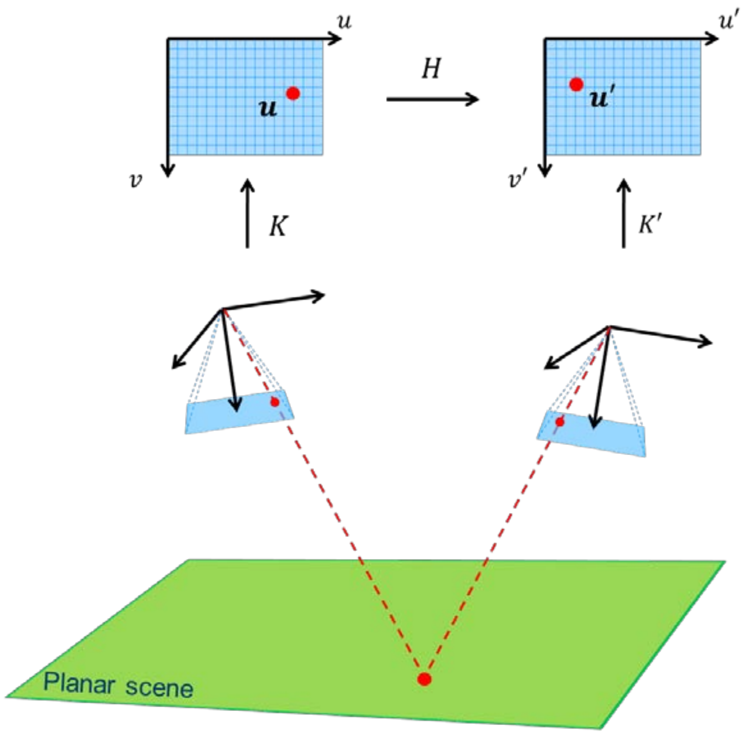
Lecture 3.3
Robust estimation with RANSAC

Thomas Opsahl

Motivation



- If two perspective cameras captures an image of a planar scene, their images are related by a homography H

Motivation

- If two perspective cameras captures an image of a planar scene, their images are related by a homography H

- It can be estimated if we know at least 4 point-correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u}'_i$

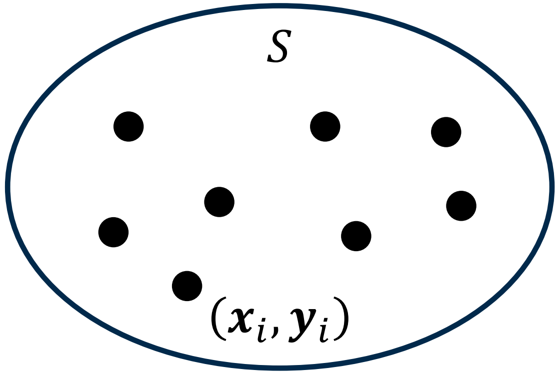
Motivation

- If two perspective cameras captures an image of a planar scene, their images are related by a homography H
- It can be estimated if we know at least 4 point-correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u'}_i$
- Correspondences can be found automatically, but typically some of them will be wrong

Motivation

- If two perspective cameras captures an image of a planar scene, their images are related by a homography H
- It can be estimated if we know at least 4 point-correspondences $\boldsymbol{u}_i \leftrightarrow \boldsymbol{u'}_i$
- Correspondences can be found automatically, but typically some of them will be wrong
- A robust estimation method provides a good estimate of H despite the presence of these wrong correspondences

RANdom SAmple Consensus - RANSAC



Observed data

$\longrightarrow$

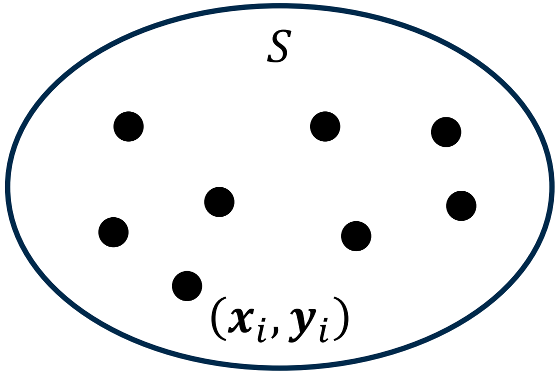
Estimation method

$$\boxed{\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha})}$$

Mathematical model with parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$

- RANSAC is an iterative method for estimating the parameters of a mathematical model from a set of observed data containing outliers

RANdom SAmple Consensus - RANSAC



Observed data

$\longrightarrow$

Estimation method

$$\boxed{\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha})}$$

Mathematical model with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$

$\boxed{y = ax + b}$

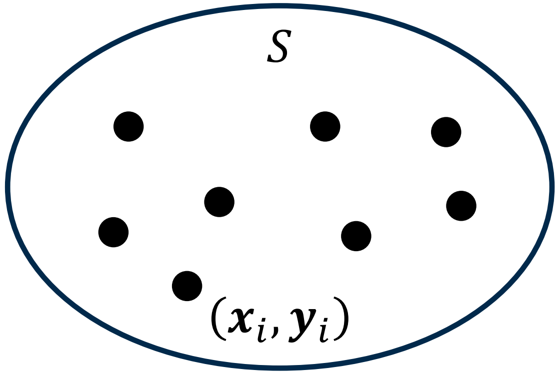
$\boxed{ax + by + c = 0}$

$\boxed{y = ax^2 + bx + c}$

$\boxed{\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1}$

- RANSAC is an iterative method for estimating the parameters of a mathematical model from a set of observed data containing outliers

RANdom SAmple Consensus - RANSAC



Observed data

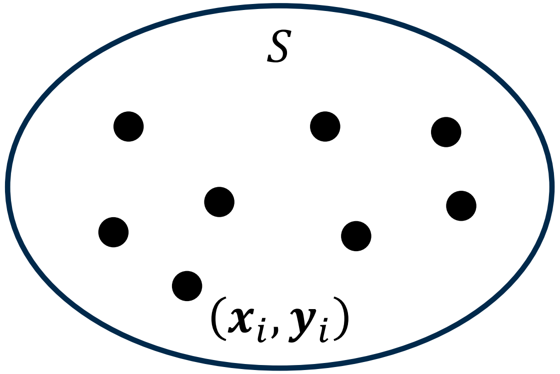
Estimation method

$$\boxed{\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha})}$$

Mathematical model with parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$

- RANSAC is an iterative method for estimating the parameters of a mathematical model from a set of observed data containing outliers
  - Robust method (handles up to 50% outliers)

RANdom SAmple Consensus - RANSAC



Observed data

$\longrightarrow$

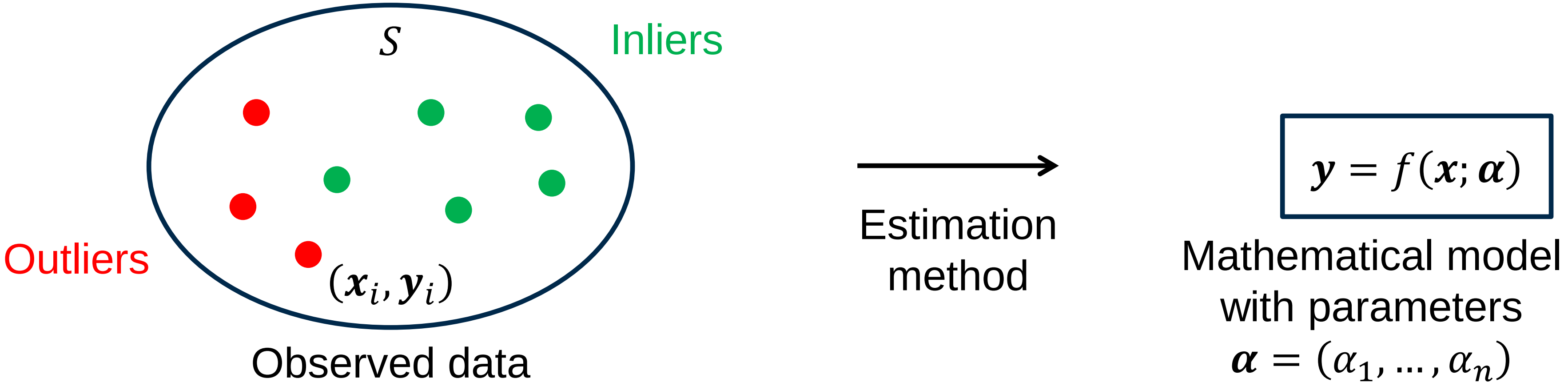
Estimation method

$$\boxed{\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha})}$$

Mathematical model with parameters
$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$

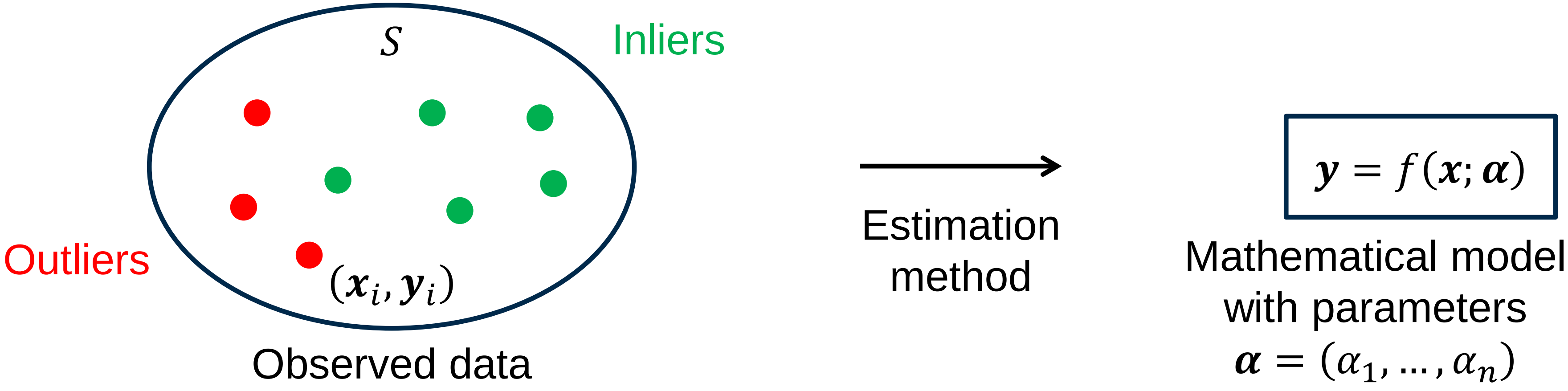
- RANSAC is an iterative method for estimating the parameters of a mathematical model from a set of observed data containing outliers
  - Robust method (handles up to 50% outliers)
  - The estimated model is random but reasonable

RANdom SAmple Consensus - RANSAC



- RANSAC is an iterative method for estimating the parameters of a mathematical model from a set of observed data containing outliers
  - Robust method (handles up to 50% outliers)
  - The estimated model is random but reasonable
  - The estimation process divides the observed data into inliers and outliers

RANdom SAmple Consensus - RANSAC



- RANSAC is an iterative method for estimating the parameters of a mathematical model from a set of observed data containing outliers
  - Robust method (handles up to 50% outliers)
  - The estimated model is random but reasonable
  - The estimation process divides the observed data into inliers and outliers
  - Usually an improved estimate of the model is determined based on the inliers using a less robust estimation method, e.g. least squares

Basic RANSAC

Objective

Robustly fit a model $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha})$ to a data set $S = \{\boldsymbol{x}_i\}$

Algorithm

1. Determine a test model $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha}_{tst})$ from n random data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$

2. Check how well each individual data point in S fits with the test model
   - Data points within a distance t of the model constitute a set of inliers $S_{tst} \subseteq S$
   - Data points outside a distance t of the model are outliers

3. If S_{tst} is the largest set of inliers encountered so far, we keep this model
   - Set $S_{IN} = S_{tst}$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{tst}$

4. Repeat steps 1-3 until N models have been tested

Basic RANSAC

Comments

- The number of random samples, n, is typically the smallest number of data points required to estimate the model
- Assuming Gaussian noise in the data, the threshold value t should be in the region of 2σ were σ is the expected noise in the data set
- The maximal number of tests, N, can be chosen according to how certain we want to be of sampling at least one data set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ with no outliers
- If p is the desired probability of sampling at least one n-tuple with no outliers and ω is the probability of a random data point to be an inlier, then

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

Basic RANSAC

Comments

- Standard value $p = 0.99$
- We rarely know the ratio of inliers in our set of data points, so in most situations, ω is unknown
- Instead of maximizing ω, leading to a larger than necessary N, we can modify RANAC to adaptively estimate N as we perform the iterations

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

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

$$p = 0.99$$

Adaptive RANSAC

Objective

Robustly fit a model $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha})$ to a data set $S = \{\boldsymbol{x}_i\}$

Algorithm

1. Let $N = \infty$, $S_{IN} = \emptyset$

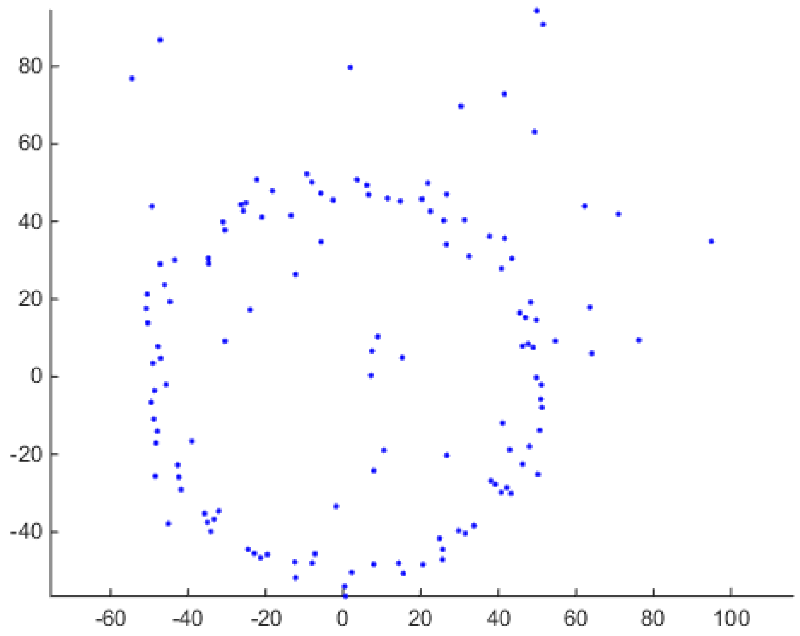
2. As long as the number of iterations are smaller than N repeat steps 3-5

3. Determine a test model $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\alpha}_{tst})$ from n random data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$

4. Check how well each individual data point in S fits with the test model
   - Data points within a distance t of the model constitute a set of inliers $S_{tst} \subseteq S$

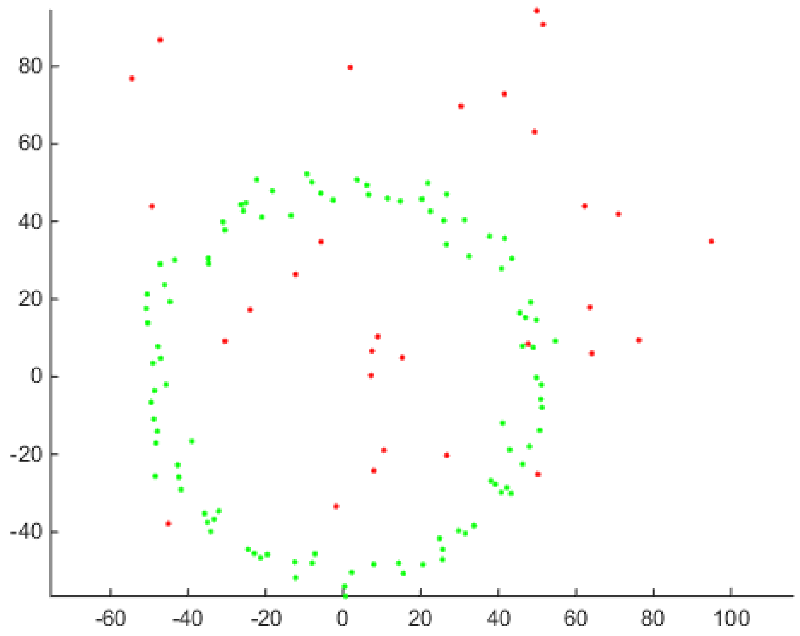
5. If S_{tst} is the largest set of inliers encountered so far, we keep this model
   - Set $S_{IN} = S_{tst}$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{tst}$
   - Compute $N = \frac{log(1-p)}{log(1-\omega^n)}$ using that $\omega = \frac{|S_{IN}|}{|S|}$ and $p = 0.99$

Example



- Fit a circle $(x - x_0)^2 + (y - y_0)^2 = r^2$ to these data points by estimating the 3 parameters x_0, y_0 and r

Example



Circle + Gaussian noise
Random points

- Fit a circle $(x - x_0)^2 + (y - y_0)^2 = r^2$ to these data points by estimating the 3 parameters x_0, y_0 and r
- The data consists of some points on a circle with Gaussian noise and some random points

Example

Least-squares 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 get one 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

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

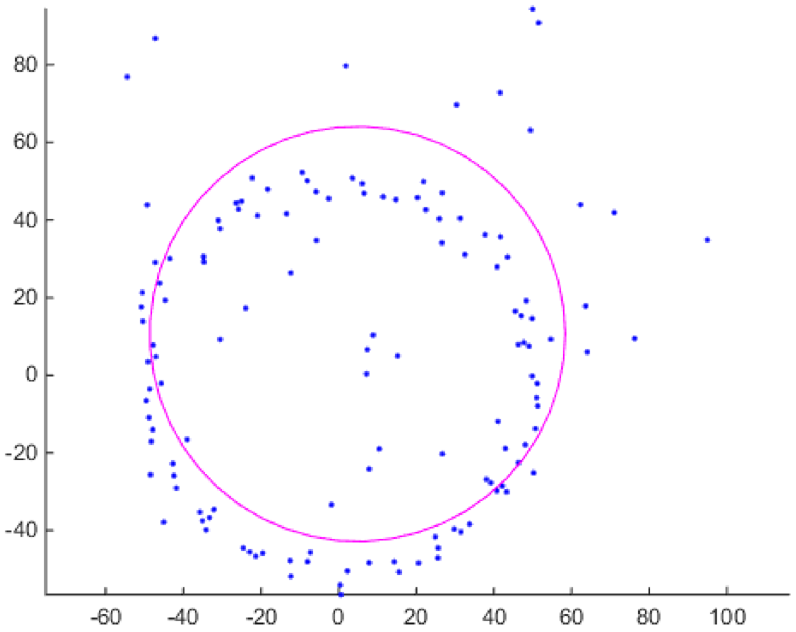
$$A\boldsymbol{p} = \boldsymbol{b}$$

Example

- One way of solving the equation $A\boldsymbol{p} = \boldsymbol{b}$ is to take the pseudo inverse $\boldsymbol{p} = (A^T A)^{-1} A^T \boldsymbol{b}$
  - This give us the solution that minimizes $||A\boldsymbol{p} - \boldsymbol{b}||$

Example

- One way of solving the equation $A\boldsymbol{p} = \boldsymbol{b}$ is to take the pseudo inverse $\boldsymbol{p} = (A^T A)^{-1} A^T \boldsymbol{b}$
  - This give us the solution that minimizes $||A\boldsymbol{p} - \boldsymbol{b}||$



- NOT GOOD! All points are treated equally, so the random points shifts the estimated circle away from the desired solution

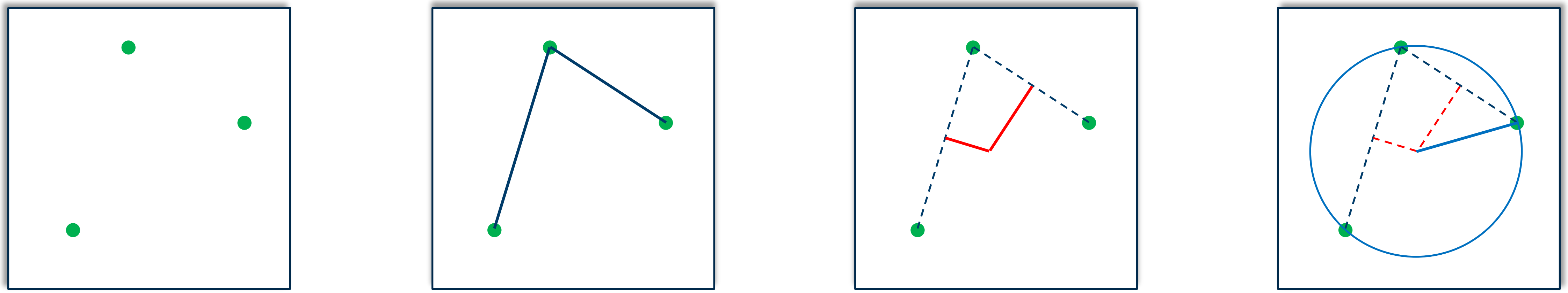
Example

- To estimate the circle using RANSAC, we need two things
  1. A way to estimate a circle from n points, where n is as small as possible
  2. A way to determine which of the points are inliers for an estimated circle

Example

- To estimate the circle using RANSAC, we need two things
  1. **A way to estimate a circle from n points, where n is as small as possible**
  2. A way to determine which of the points are inliers for an estimated circle

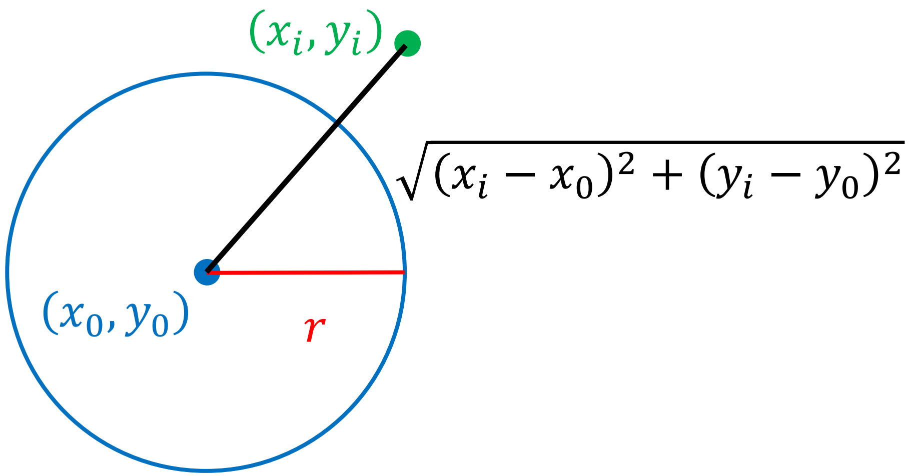
- The smallest number of points required to determine a circle is 3, i.e. $n = 3$, and the algorithm for computing the circle is quite simple



Example

- To estimate the circle using RANSAC, we need two things
  1. A way to estimate a circle from n points, where n is as small as possible
  2. **A way to determine which of the points are inliers for an estimated circle**

- The distance from a point (x_i, y_i) to a circle $(x - x_0)^2 + (y - y_0)^2 = r^2$ is given by $\left|\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r\right|$



Example

- To estimate the circle using RANSAC, we need two things
  1. A way to estimate a circle from n points, where n is as small as possible
  2. **A way to determine which of the points are inliers for an estimated circle**

- The distance from a point (x_i, y_i) to a circle $(x - x_0)^2 + (y - y_0)^2 = r^2$ is given by $\left|\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r\right|$

- So for a threshold value t, we say that (x_i, y_i) is an inlier if $\left|\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r\right| < t$

Example

Objective

To robustly fit the model $(x - x_0)^2 + (y - y_0)^2 = r^2$ to our data set $S = \{(x_i, y_i)\}$

Algorithm

1. Let $N = \infty$, $S_{IN} = \emptyset$, $p = 0.99$, $t = 2 \cdot expected\ noise$

2. As long as the number of iterations are smaller than N repeat steps 3-5

3. Determine parameters $(x_{tst}, y_{tst}, r_{tst})$ from three random points from S

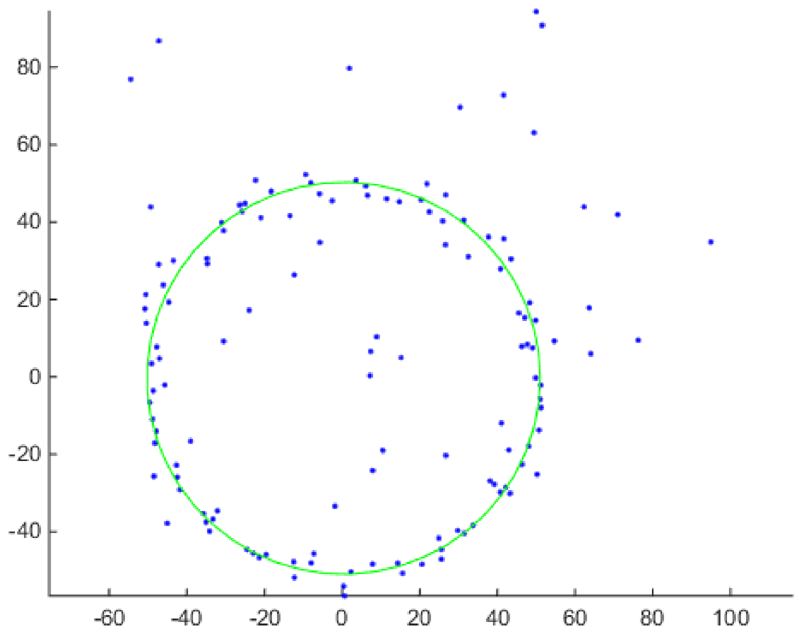
4. Check how well each individual data point in S fits with the test model

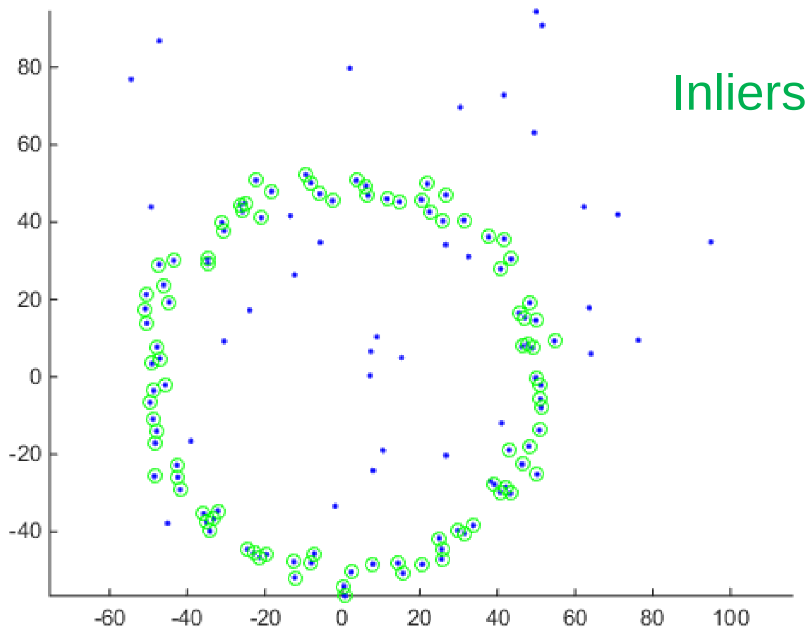
$$S_{tst} = \left\{ (x_i, y_i) \in S \text{ such that } \left| \sqrt{(x_i - x_{tst})^2 + (y_i - y_{tst})^2} - r_{tst} \right| < t \right\}$$

5. If S_{tst} is the largest set of inliers encountered so far, we keep this model
   - Set $S_{IN} = S_{tst}$ and $(x_0, y_0, r) = (x_{tst}, y_{tst}, r_{tst})$
   - Recompute $N = \frac{log(1-p)}{log(1-\omega^n)}$ using that $\omega = \frac{|S_{IN}|}{|S|}$

Example

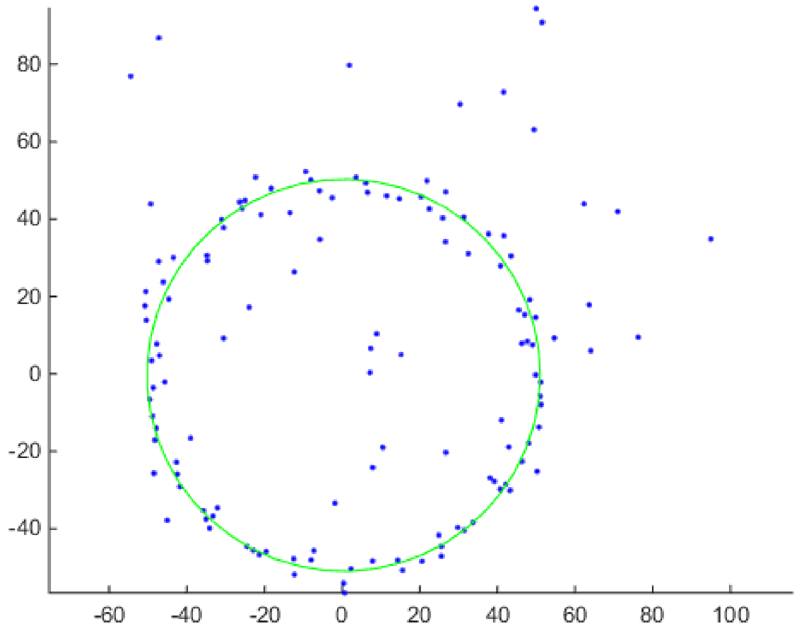
- The RANSAC algorithm evaluates many different circles and returns the circle with the largest inlier set

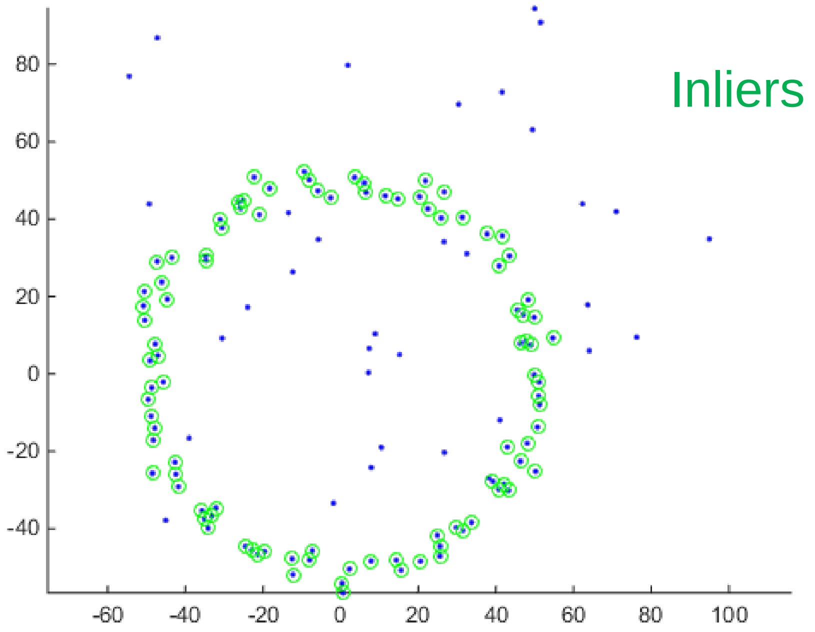




Example

- The RANSAC algorithm evaluates many different circles and returns the circle with the largest inlier set

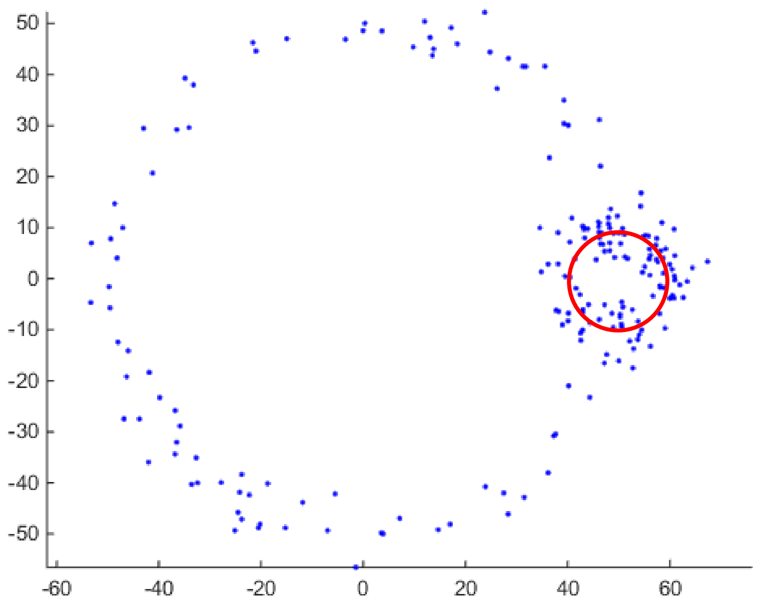




- An improved estimate for the circle can found from the set of inliers using a less robust algorithm e.g. least squares

Robust estimation

- But RANSAC is not perfect…



- Several other robust estimation methods exist
  - Least Median Squares (LMS)
  - Preemptive RANSAC
  - PROgressive Sample and Consensus (PROSAC)
  - M-estimator Sample and Consensus (MSAC)
  - Maximum Likelihood Estimation Sample and Consensus (MLESAC)
  - Randomized RANSAC (R-RANSAC)
  - KALMANSAC
  - +++

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” which is very useful if we want to use better, but less robust, estimation methods

- Additional reading
  - Szeliski: 6.1.4

- Homework?
  - Implement a RANSAC algorithm for estimating a line