UiO : Department of Technology Systems
University of Oslo

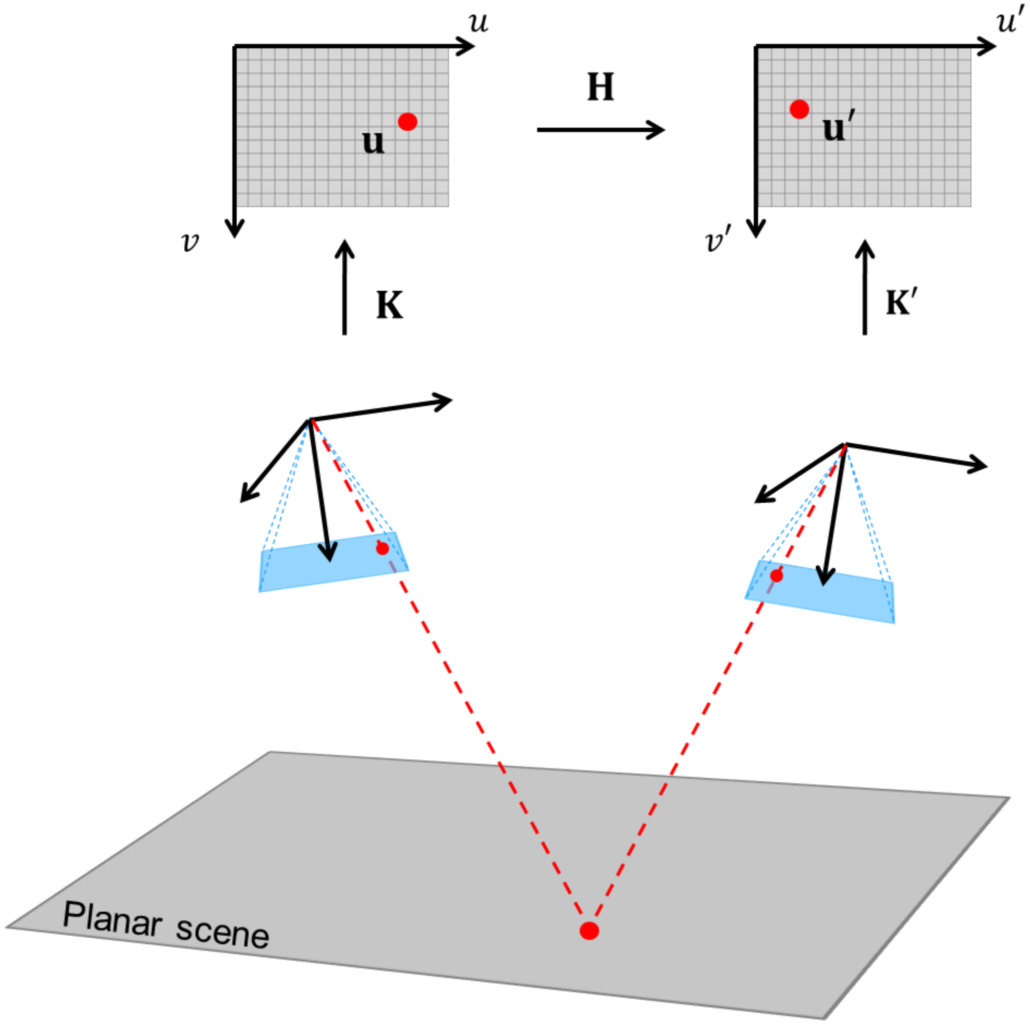
# Robust estimation with RANSAC

Thomas Opsahl

2023

# Motivation

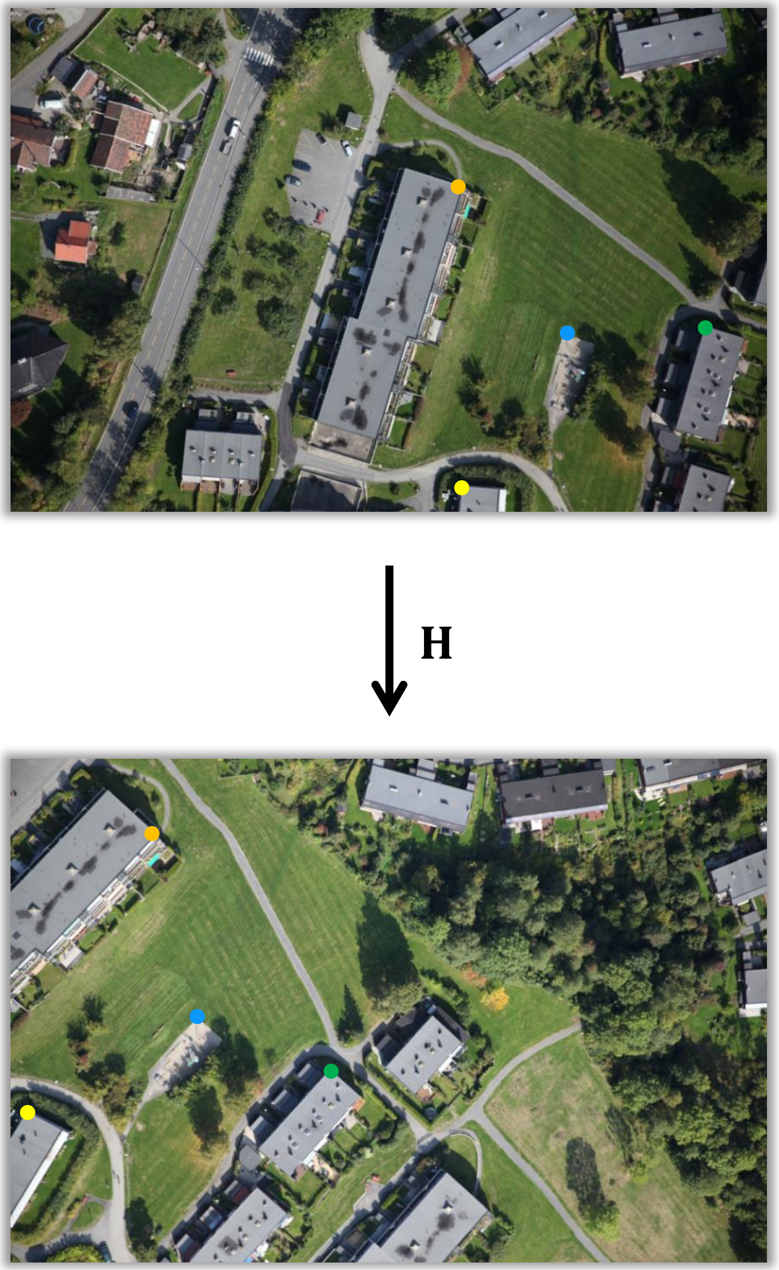
- If two perspective cameras captures images of a planar scene, their images are related by a homography $\mathbf{H}$



# Motivation

- If two perspective cameras captures images of a planar scene, their images are related by a homography $\mathbf{H}$

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

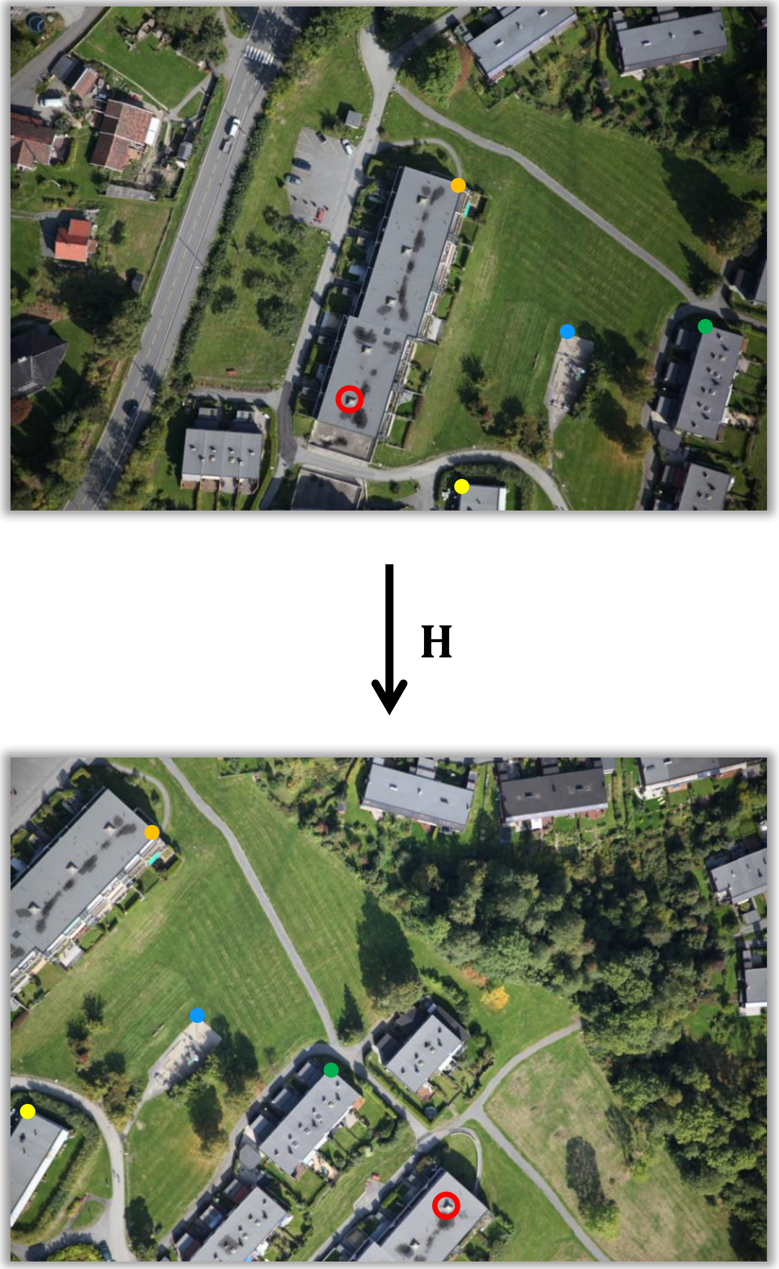


# Motivation

- If two perspective cameras captures images of a planar scene, their images are related by a homography $\mathbf{H}$

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

- Correspondences can be found automatically, but typically some of them will be wrong



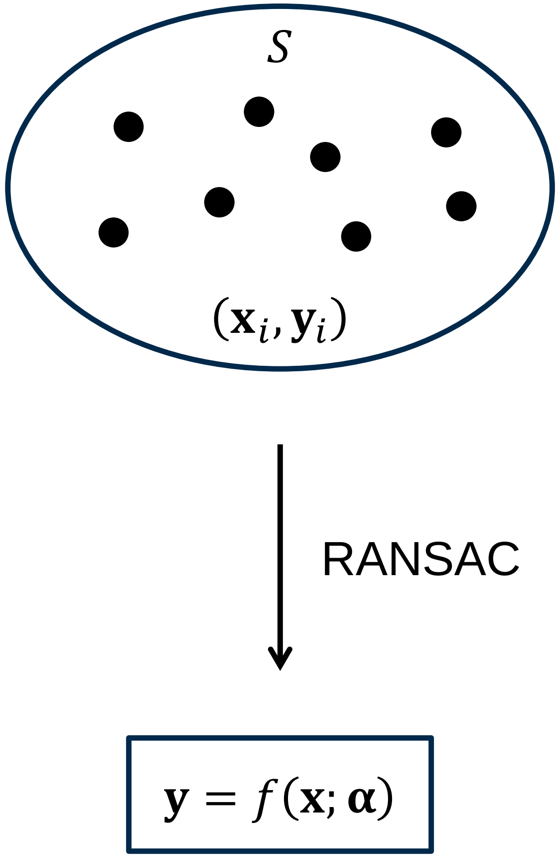
# Motivation

- If two perspective cameras captures images of a planar scene, their images are related by a homography $\mathbf{H}$
- It can be estimated if we know at least 4 point-correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u'}_i$
- Correspondences can be found automatically, but typically some of them will be wrong
- RANSAC is a general method for detecting bad data that we can use to remove bad correspondences and robustly estimate $\mathbf{H}$

# RANdom SAmple Consensus - RANSAC

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

Observed data

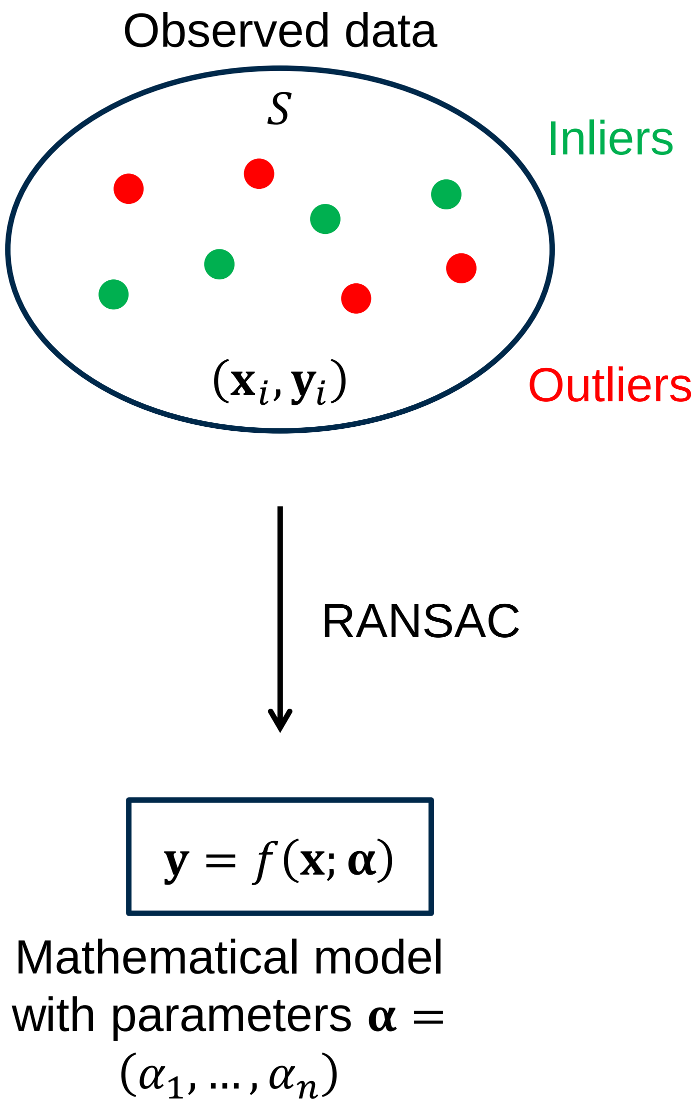



Mathematical model with parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$

# RANdom SAmple Consensus - RANSAC

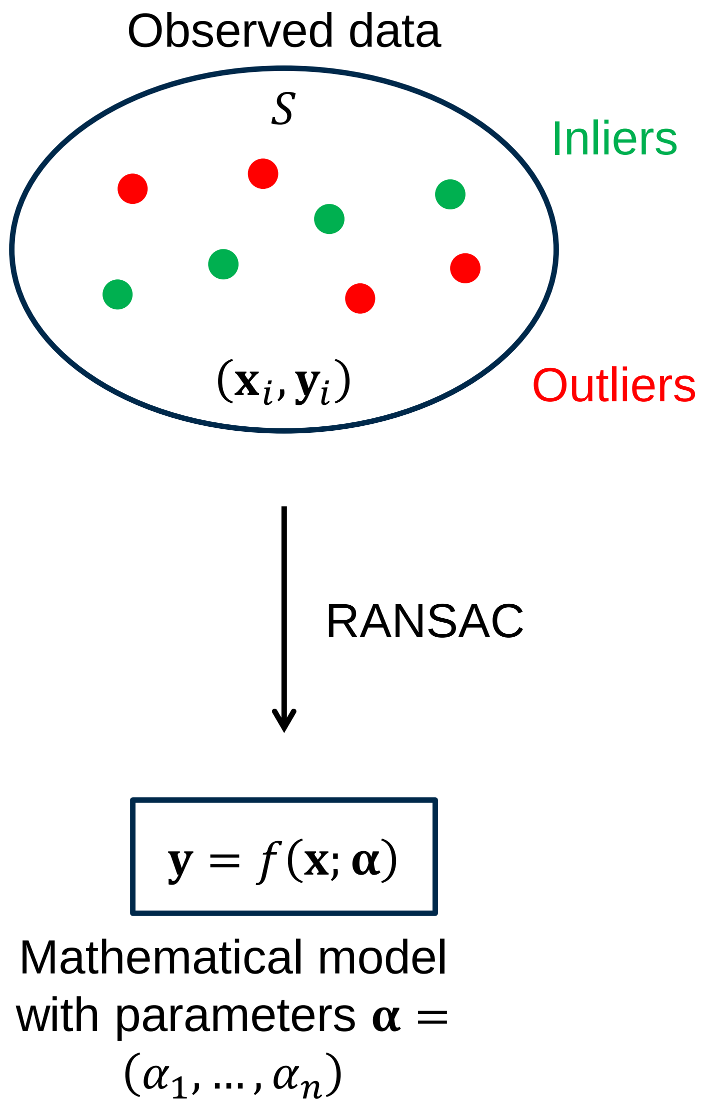
- RANSAC is an iterative method for estimating the parameters of a mathematical model from a set of observed data that contains outliers

- The RANSAC estimation process divides the observed data into inliers and outliers, so it can also be regarded as an outlier detection method

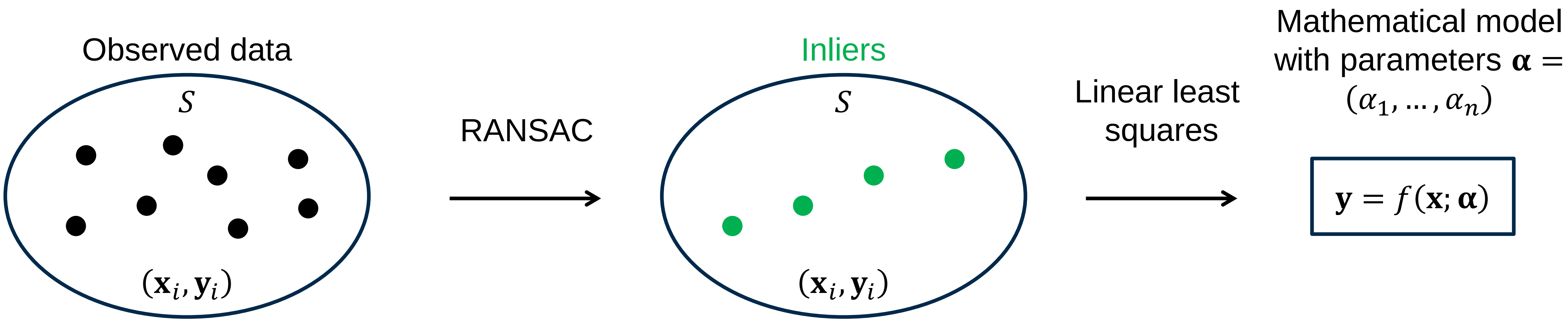


# RANdom SAmple Consensus - RANSAC

- RANSAC is an iterative method for estimating the parameters of a mathematical model from a set of observed data that contains outliers
  - Secondary application!
- The RANSAC estimation process divides the observed data into inliers and outliers, so it can also be regarded as an outlier detection method
  - Main application!



# RANdom SAmple Consensus - RANSAC




- In order to estimate a model from data containing outliers, it is common to use RANSAC in combination with another estimation method e.g. linear least squares

- The model estimated in RANSAC is usually ignored
  - It is based on a small subset of the observed data
  - Different RANSAC estimations will typically return different models (RANSAC is non-deterministic)

# Basic RANSAC

**Objective**

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

**Algorithm**

Repeat steps 1-3 until $N$ models have been tested

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

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

3. If $S_{tst}$ is larger than all previous set of inliers, we update the RANSAC model $f(\mathbf{x}; \boldsymbol{\alpha}) = f(\mathbf{x}; \boldsymbol{\alpha}_{tst})$ and the set of inliers $S_{IN} = S_{tst}$

# Comments

- The number of tests, $N$, is directly related to the probability of sampling at least one random $n$-tuple $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ with no outliers in it

  The test model and inlier set corresponding to such an $n$-tuple should be an acceptable result of the RANSAC estimation

- If $\omega$ is the probability of a random data point being an inlier, then the number of test $N$ and the probability $p$ to sample at least one random $n$-tuple with no outliers is related by

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

- By keeping $n$ as small as possible, we also minimize the number of required tests $N$ for a desired level of confidence $p$

# Basic RANSAC

**Comments**

- Standard value $p = 0.99$

- We rarely know the ratio of inliers in our dataset, so in most situations, $\omega$ is unknown

- Instead of choosing an large $\omega$ just to be on the safe side, leading to a larger than necessary $N$, we can modify RANAC to adaptively estimate $N$ as we perform the iterations

$\omega = P(inlier)$

| $n$ \ $N$ | 0.9 | 0.8 | 0.7 | 0.6 | 0.5 |
|---|---|---|---|---|---|
| 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 |

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

$$p = 0.99$$

# Adaptive RANSAC

## Objective

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

**Algorithm**

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

While ($num\_iterations < N$) repeat steps 1-4

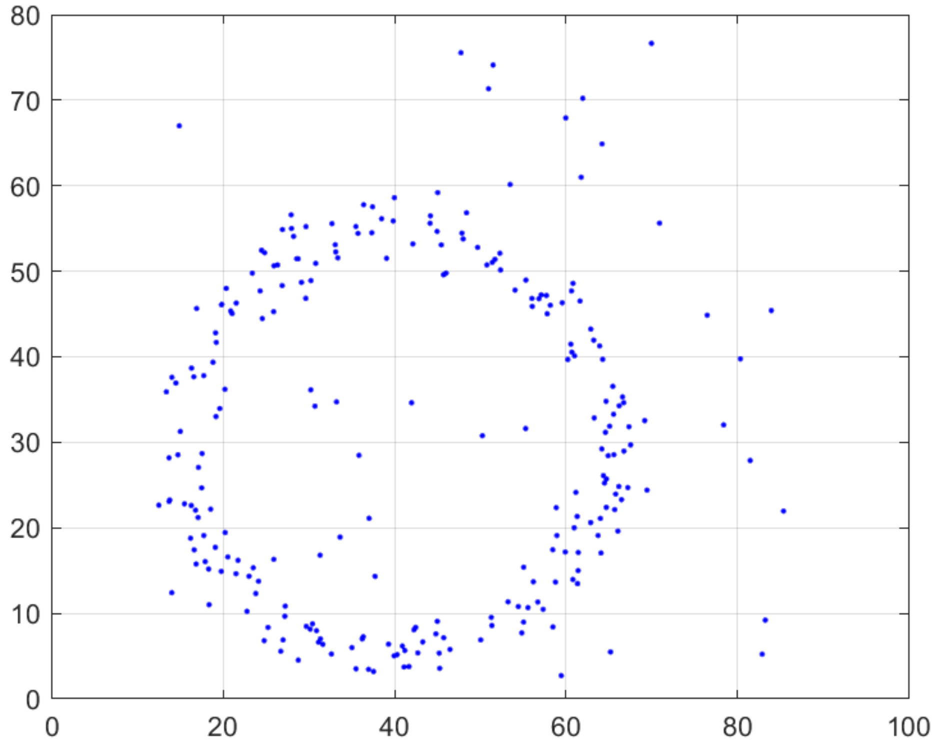
1. Determine a test model $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\alpha}_{tst})$ from $n$ random data points $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$

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

3. If $S_{tst}$ is larger than $S_{IN}$, we update the RANSAC model $f(\mathbf{x}; \boldsymbol{\alpha}) = f(\mathbf{x}; \boldsymbol{\alpha}_{tst})$ and the set of inliers $S_{IN} = S_{tst}$

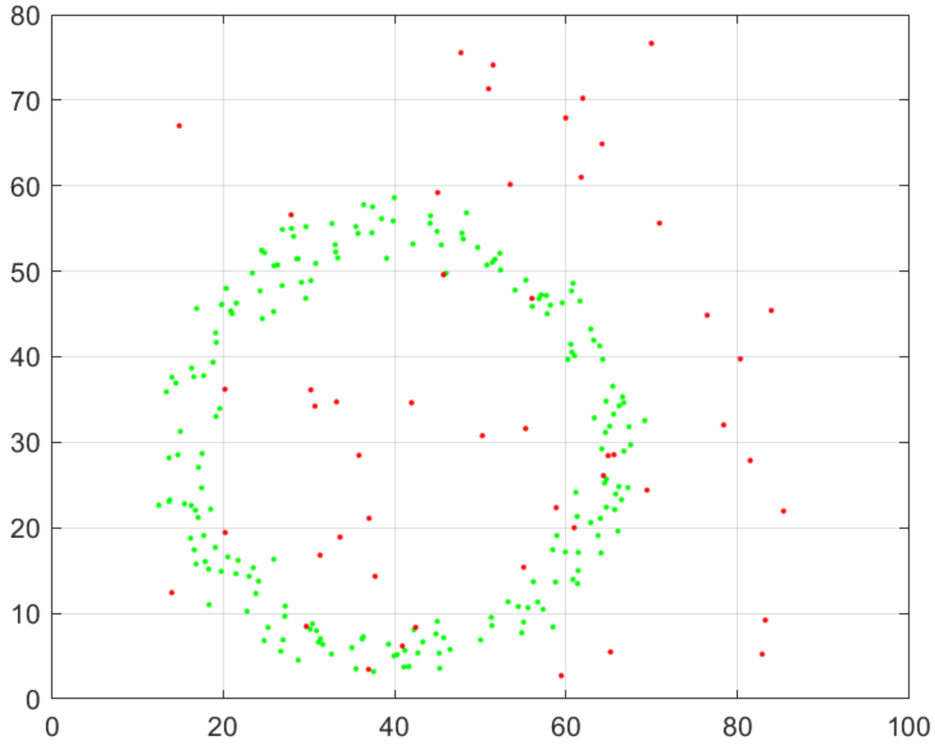
4. Compute $N = \frac{log(1-p)}{log(1-\omega^n)}$ using that $\omega = \frac{|S_{IN}|}{|S|}$ and $p = 0.99$

# 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




Random points on a 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 set consists of random points on a circle with some Gaussian noise added to them and some additional random points

# Example

**Linear 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

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

$$\mathbf{Ap} = \mathbf{b}$$

# Example

- Here we have $n > 3$ data points and only $3$ parameters
  - Overdetermined set of equations
  - Typically no exact solution

- The linear least squares solution to the problem is the parameter $\mathbf{p}^*$ that minimizes the sum of squares of residuals

$$\mathbf{p}^* = \underset{\mathbf{p}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{p} - \mathbf{b}\|^2$$

- This can be found by solving the following equation

$$\frac{\partial}{\partial \mathbf{p}}\left(\|\mathbf{A}\mathbf{p} - \mathbf{b}\|^2\right) = \mathbf{0}$$

# Example

- This leads to the so called normal equations

$$\frac{\partial}{\partial \mathbf{p}}(\|\mathbf{Ap}-\mathbf{b}\|^2) = \mathbf{0}$$

$$2\mathbf{A}^T(\mathbf{Ap}-\mathbf{b}) = \mathbf{0}$$

$$\mathbf{A}^T\mathbf{Ap} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$$

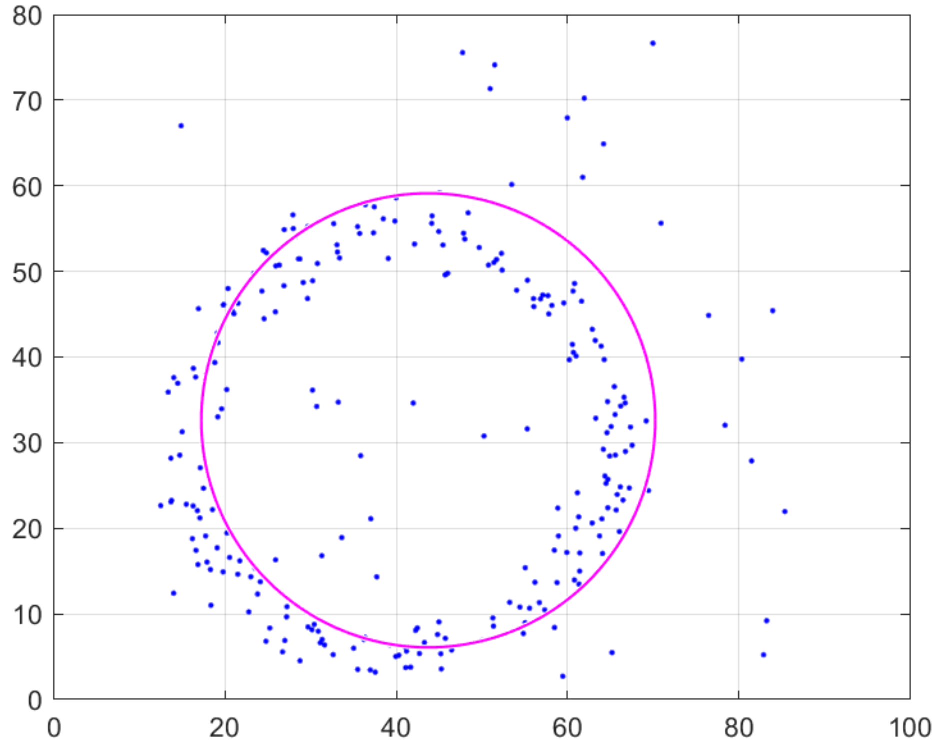
- Hence the linear least squares solution is

$$\mathbf{p}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

# Example

- The linear least squares solution for our problem looks like this...

$$\underbrace{\begin{bmatrix} 2x_0 \\ 2y_0 \\ r^2 - x_0^2 - y_0^2 \end{bmatrix}}_{\mathbf{p}^*} = \underbrace{\left( \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \vdots & \\ x_n & y_n & 1 \end{bmatrix}^T \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \vdots & \\ x_n & y_n & 1 \end{bmatrix} \right)^{-1}}_{(\mathbf{A}^T\mathbf{A})^{-1}} \underbrace{\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \vdots & \\ x_n & y_n & 1 \end{bmatrix}^T}_{\mathbf{A}^\mathrm{T}} \underbrace{\begin{bmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ \vdots \\ x_n^2 + y_n^2 \end{bmatrix}}_{\mathbf{b}}$$




NOT GOOD!

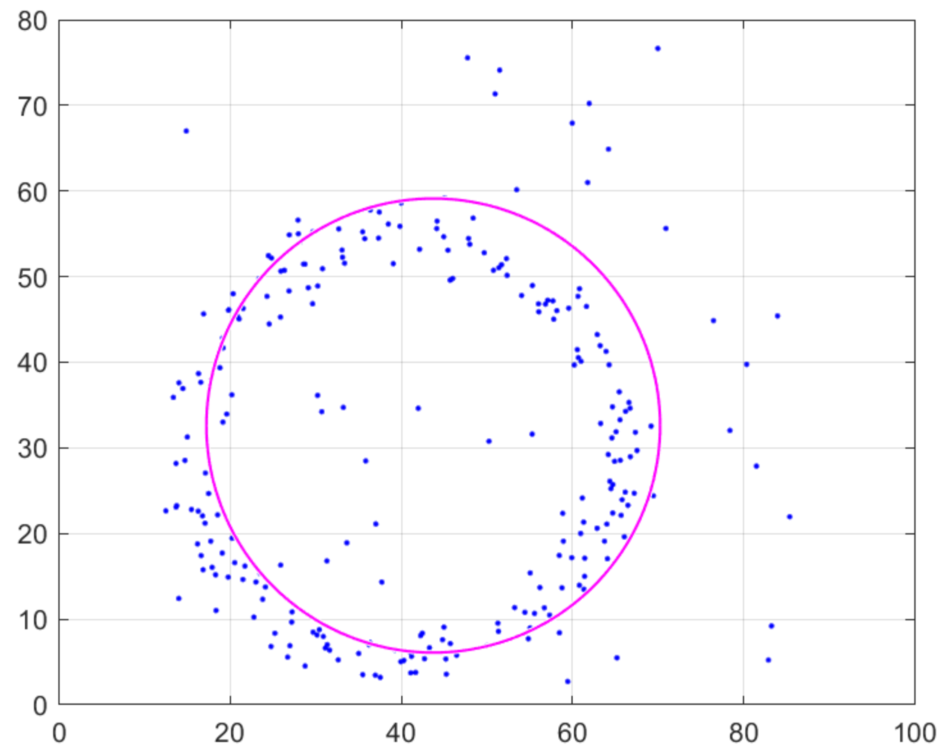
# Example

- The linear least squares solution for our problem looks like this...

$$\underbrace{\begin{bmatrix} 2x_0 \\ 2y_0 \\ r^2 - x_0^2 - y_0^2 \end{bmatrix}}_{\mathbf{p}^*} = \underbrace{\left( \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \vdots & \\ x_n & y_n & 1 \end{bmatrix}^T \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \vdots & \\ x_n & y_n & 1 \end{bmatrix} \right)^{-1}}_{(\mathbf{A}^T\mathbf{A})^{-1}} \underbrace{\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \vdots & \\ x_n & y_n & 1 \end{bmatrix}^T}_{\mathbf{A}^\mathrm{T}} \underbrace{\begin{bmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ \vdots \\ x_n^2 + y_n^2 \end{bmatrix}}_{\mathbf{b}}$$

- Since all points are treated equal, the random points shift the estimated circle away from the desired solution

- Now let us try RANSAC




NOT GOOD!

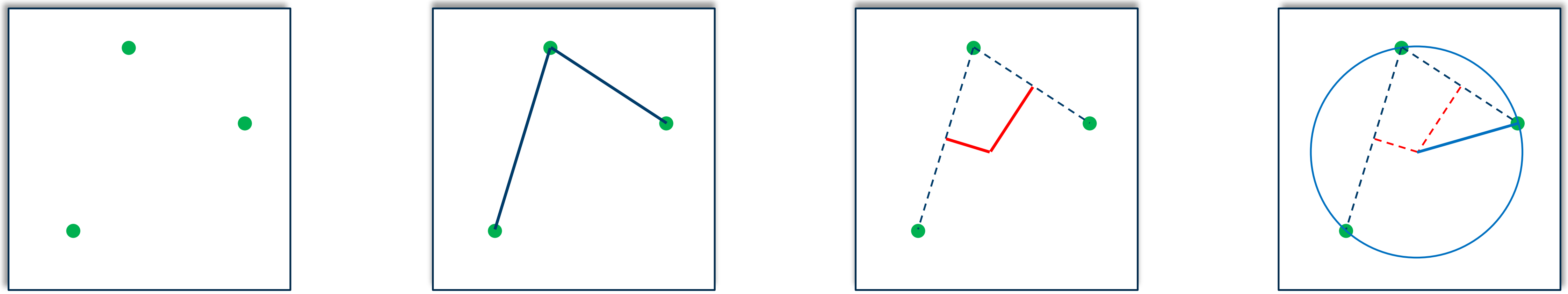
# Example

- RANSAC requires 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

- RANSAC requires 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

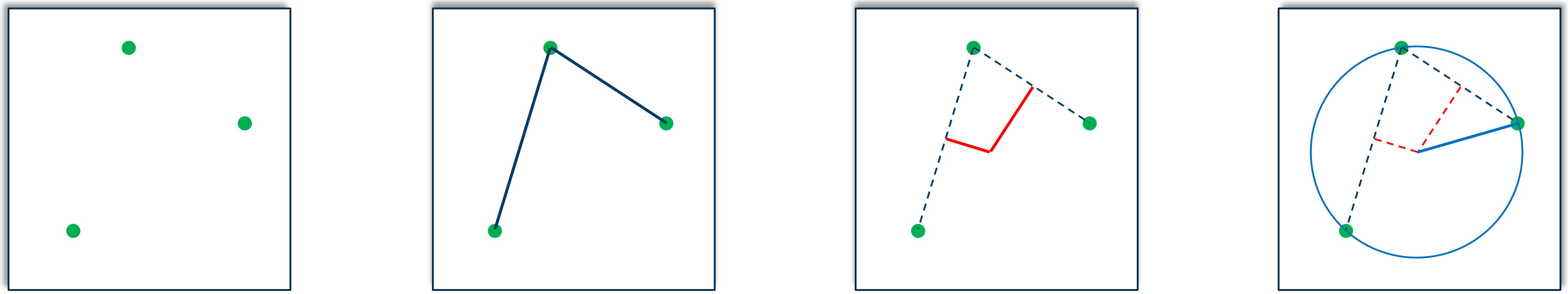


# Example

- RANSAC requires 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

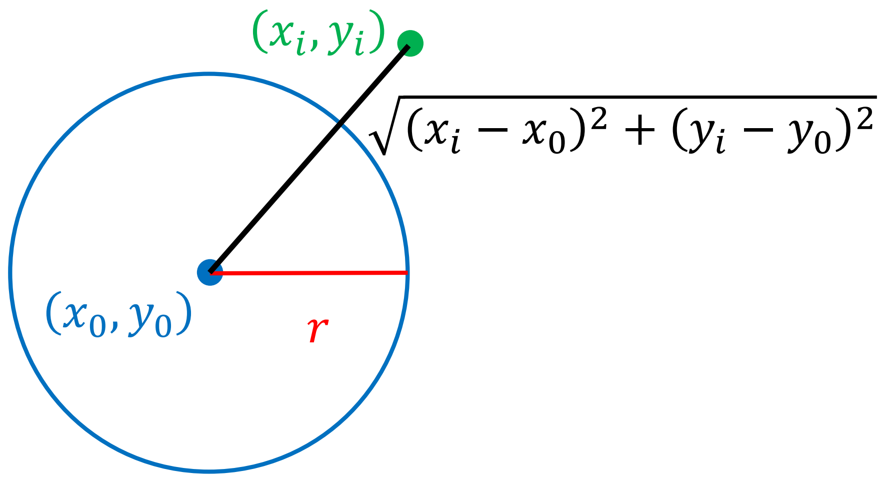
We could also have used the least squares approach from earlier, just with three points!



# Example

- RANSAC requires 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

- RANSAC requires 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 $\left|\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r\right| < t$

- The value of $t$ should be chosen according to the noise/uncertainty we expect in the data points $(x_i, y_i)$
  - In the case of Gaussian noise with standard deviation $\sigma = \sigma_x = \sigma_y$, $t = 3\sigma$ should enable us to find a large set of inliers

# 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

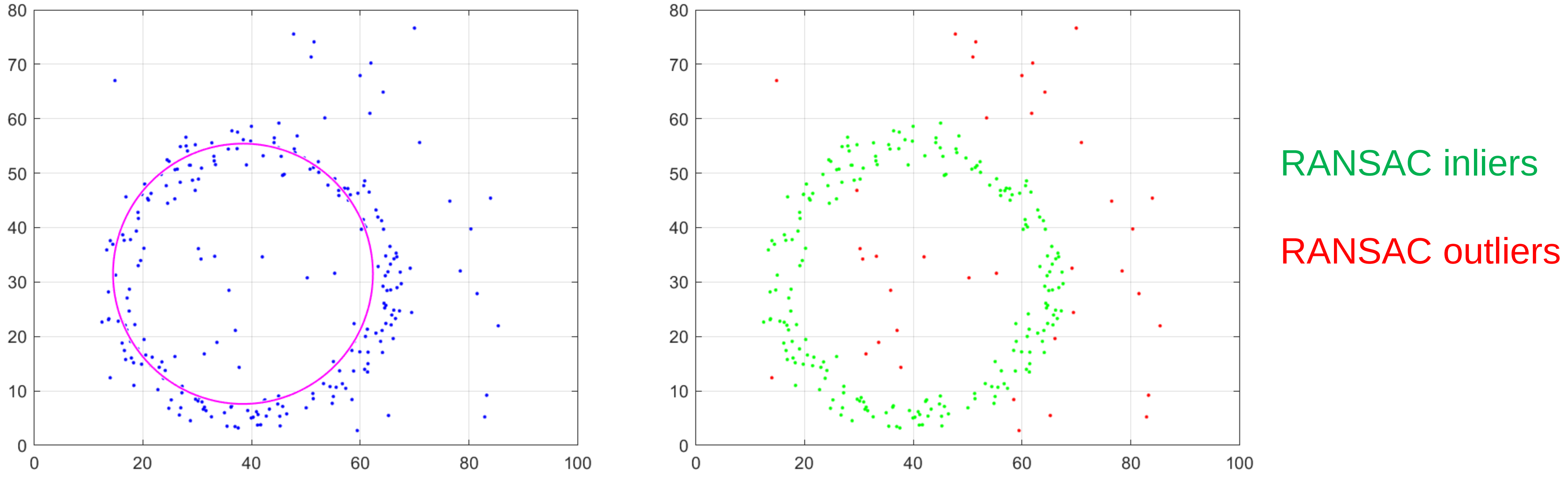
3. Determine parameters $\left(x_{0,tst}, y_{0,tst}, r_{tst}\right)$ from three random points from $S$

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

$$S_{tst} = \left\{(x_i, y_i) \in S \text{ such that } \left|\sqrt{\left(x_i - x_{0,tst}\right)^2 + \left(y_i - y_{0,tst}\right)^2} - r_{tst}\right| < t \right\}$$

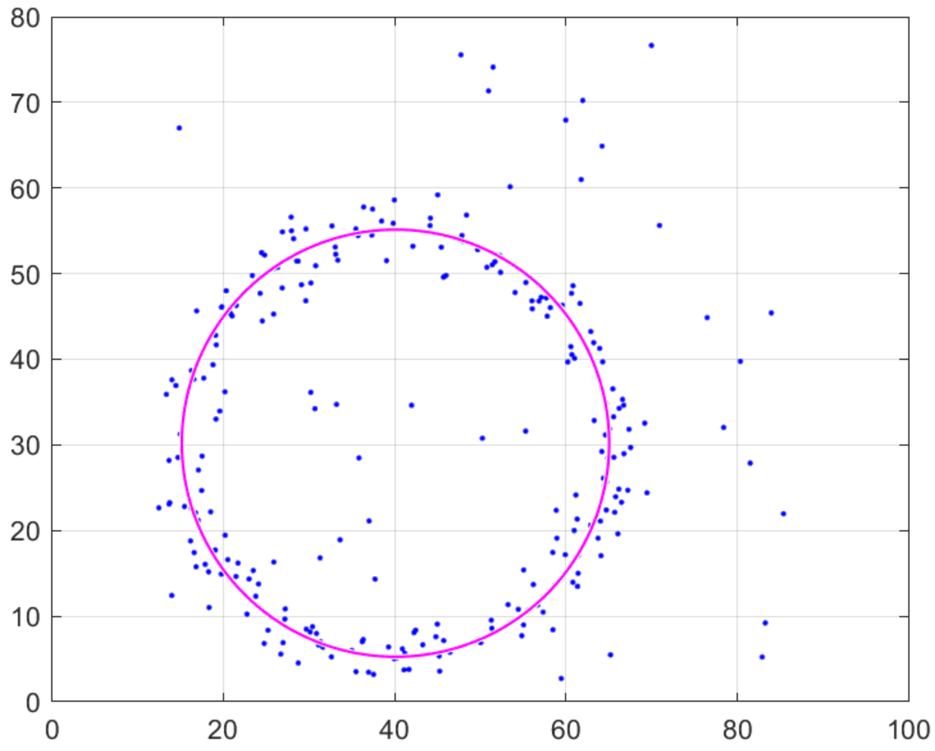
5. If $S_{tst}$ is the largest set of inliers encountered so far, we keep this model
   - Set $S_{IN} = S_{tst}$ and $(x_0, y_0, r) = \left(x_{0,tst}, y_{0,tst}, r_{tst}\right)$
   - Recompute $N = \frac{log(1-p)}{log(1-\omega^n)}$ using that $\omega = \frac{|S_{IN}|}{|S|}$

# Example



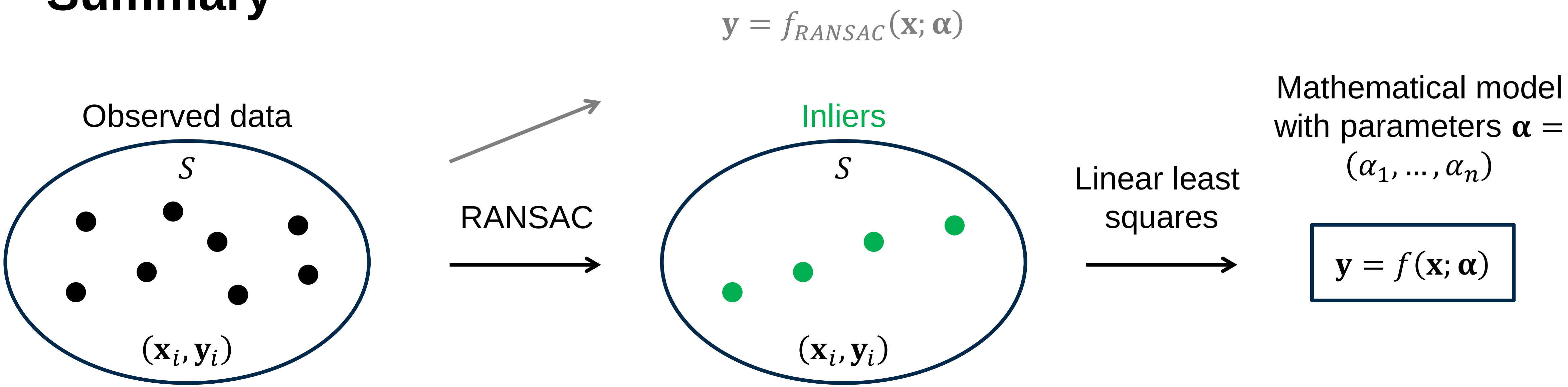

- RANSAC output (an example)
  - The RANSAC estimated circle typically changes from one estimation to another
  - The RANSAC estimated inliers are more consistent

# Example




- Linear least squares solution based on RANSAC inliers

# Summary




- RANSAC is an inlier detection method commonly used in combination with an estimation method like linear least squares to estimate a mathematical model from a dataset containing outliers

- RANSAC also provides an estimate for the mathematical model,
  - Typically estimated from only a small subset of the inliers
  - Typically different from one estimation to another

# Supplementary material

**Recommended**

- *Richard Szeliski: Computer Vision: Algorithms and Applications 2nd ed*
  - Chapter 8 “Image alignment and stitching”, in particular section 8.1.4 “Robust least-squares and RANSAC”