

# **RANSAC- RANdom Sample Consensus**

Chul Min Yeum

Assistant Professor

Civil and Environmental Engineering

University of Waterloo, Canada

CIVE 497 – CIVE 700: Smart Structure Technology



**UNIVERSITY OF WATERLOO**  
FACULTY OF ENGINEERING

Last updated: 2020-03-01

# Overview

- If we know which points (with errors) belong to the line, how do we find the “optimal” line parameters?

**Least squares**

- What if there are outliers?

**Robust fitting, RANSAC**

- What if we're not even sure it's a line?

**Model selection (topics in machine learning)**

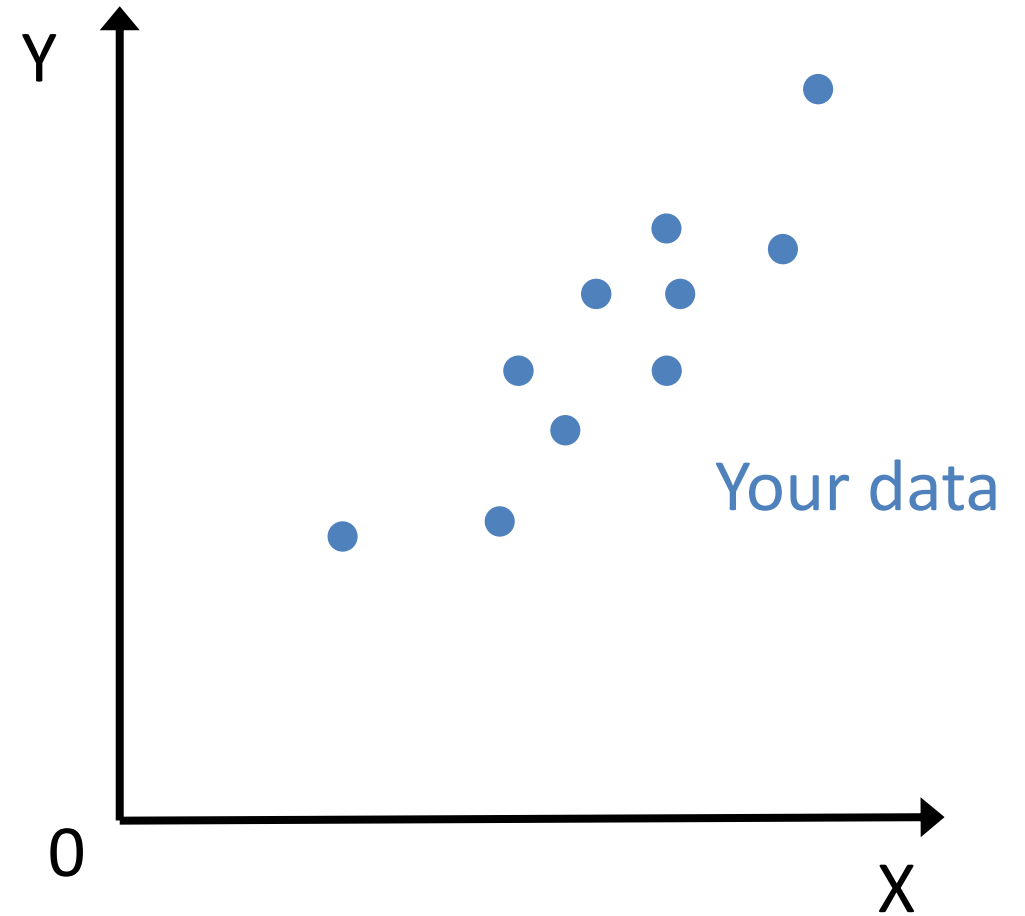


# Line Fitting

Data (measurement):  $(x_1, y_1), \dots, (x_n, y_n)$

**Known** model: Line  $(y_i = m x_i + b)$

We will find  $m$  and  $b$ .



# Least Squares Line Fitting (Approach 1)

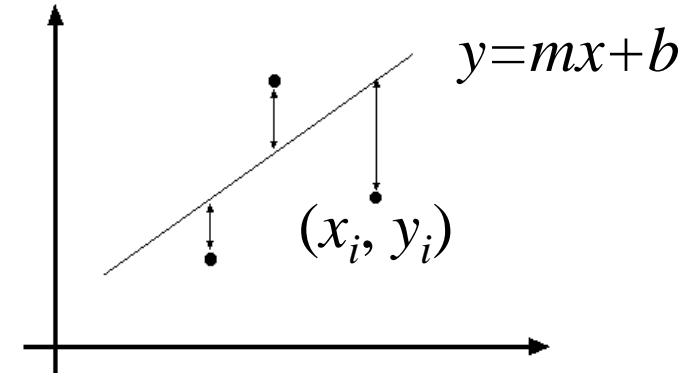
Data (measurement):  $(x_1, y_1), \dots, (x_n, y_n)$

Model: Line ( $y_i = mx_i + b$ )

Task: Find  $(m, b)$

Minimize

$$E = \sum_{i=1}^n (y_i - mx_i - b)^2$$



$$\frac{\partial(E)}{\partial m} = -2 \sum_{i=1}^n [y_i - mx_i - b]x_i = 0$$

$$b \sum_{i=1}^n x_i + m \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

$$\frac{\partial(E)}{\partial b} = -2 \sum_{i=1}^n [y_i - mx_i - b] = 0$$

$$nb + m \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

# Least Squares Line Fitting (Approach 1) (Continue)

$$nb + m \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$b \sum_{i=1}^n x_i + m \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$



$$\begin{bmatrix} \sum_{i=1}^n x_i & n \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$\begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_i & n \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}$$

$$\begin{bmatrix} \sum_{i=1}^n x_i & n \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \end{bmatrix}^{-1}$$

$$= \frac{1}{\sum_{i=1}^n x_i \sum_{i=1}^n x_i - n \sum_{i=1}^n x_i^2} \begin{bmatrix} \sum_{i=1}^n x_i & -n \\ -\sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \end{bmatrix}$$

# Least Squares Line Fitting (Approach 1) (Continue)

$$\begin{bmatrix} \sum_{i=1}^n x_i & n \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \end{bmatrix}^{-1} = \frac{1}{\sum_{i=1}^n x_i \sum_{i=1}^n x_i - n \sum_{i=1}^n x_i^2} \begin{bmatrix} \sum_{i=1}^n x_i & -n \\ -\sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \end{bmatrix}$$

$$\begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n x_i & n \\ \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}$$

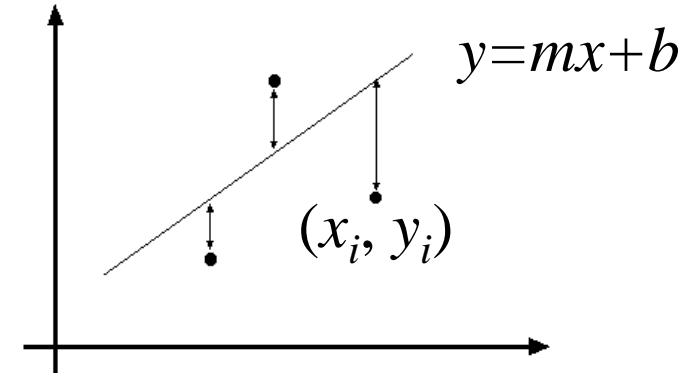
$$m = \frac{\sum_{i=1}^n x_i y_i - 1/n (\sum_{i=1}^n x_i \sum_{i=1}^n y_i)}{\sum_{i=1}^n x_i^2 - 1/n (\sum_{i=1}^n x_i)^2}$$
$$b = \frac{1/n (\sum_{i=1}^n y_i) (\sum_{i=1}^n x_i^2) - 1/n \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2 - 1/n (\sum_{i=1}^n x_i)^2}$$

# Least Squares Line Fitting (Approach 2)

Data (measurement):  $(x_1, y_1), \dots, (x_n, y_n)$

Model: Line  $(y_i = mx_i + b)$

Task: Find  $(m, b)$



Minimize  $E = \sum_{i=1}^n (y_i - mx_i - b)^2$

$$E = \|Y - XB\|^2 \quad \text{where} \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \quad B = \begin{bmatrix} m \\ b \end{bmatrix}$$

$$E = \|Y - XB\|^2 = (Y - XB)^T (Y - XB) = Y^T Y - 2(XB)^T Y + (XB)^T (XB)$$

## Least Squares Line Fitting (Approach 2) (Continue)

$$E = \|Y - XB\|^2 = (Y - XB)^T (Y - XB) = Y^T Y - 2(XB)^T Y + (XB)^T (XB)$$

$$\frac{dE}{dB} = 2X^T XB - 2X^T Y = 0 \quad \boxed{X^T XB = X^T Y}$$

Least squares solution to  $XB=Y$

$$\text{where } Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \quad B = \begin{bmatrix} m \\ b \end{bmatrix}$$

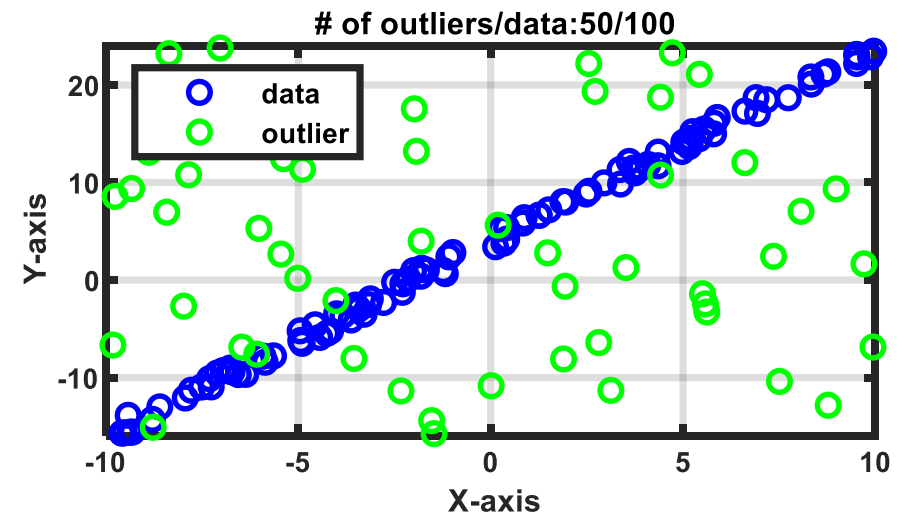
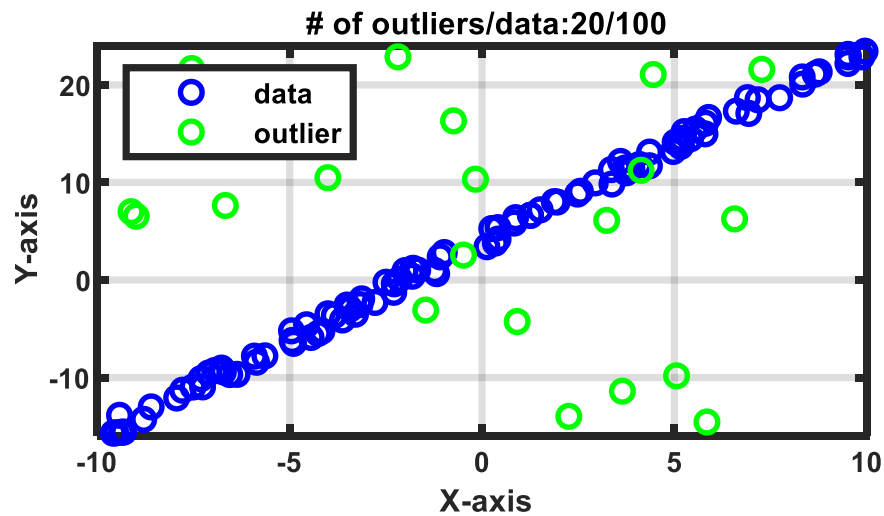
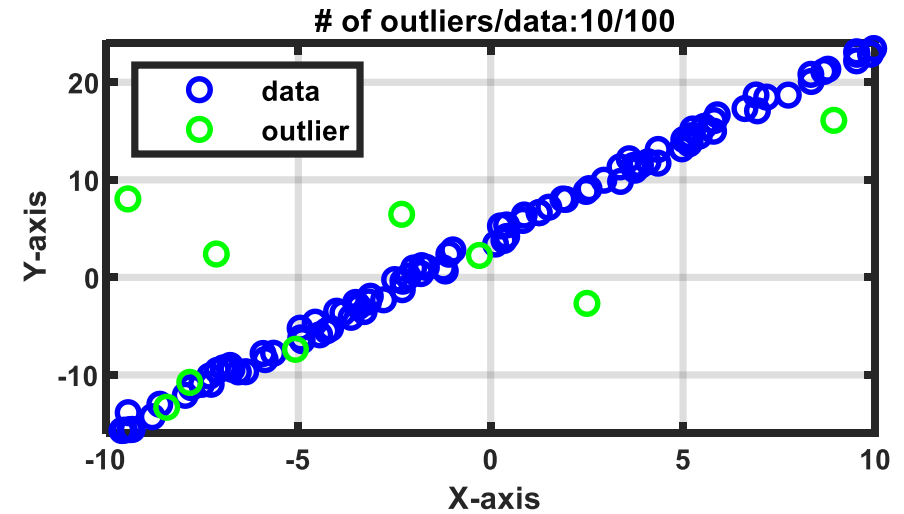
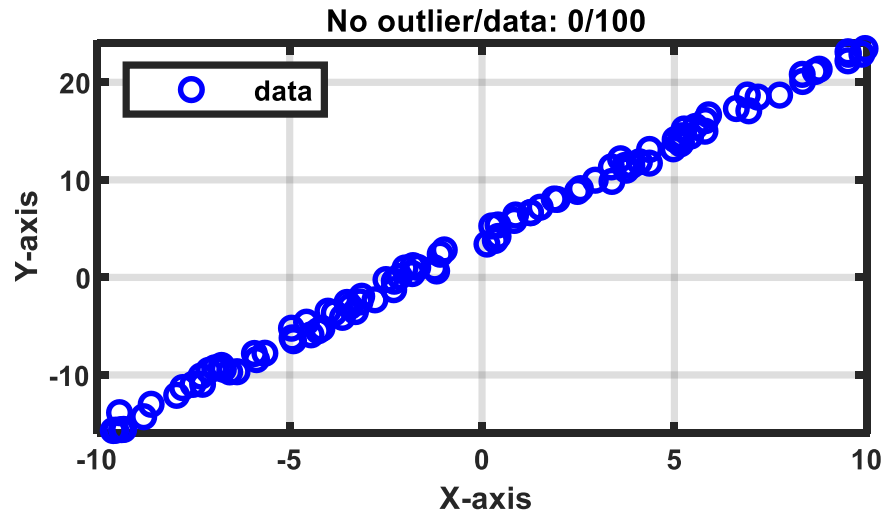
$$B \approx X^* Y$$

$$\text{where } X^* = (X'X)^{-1}X'$$

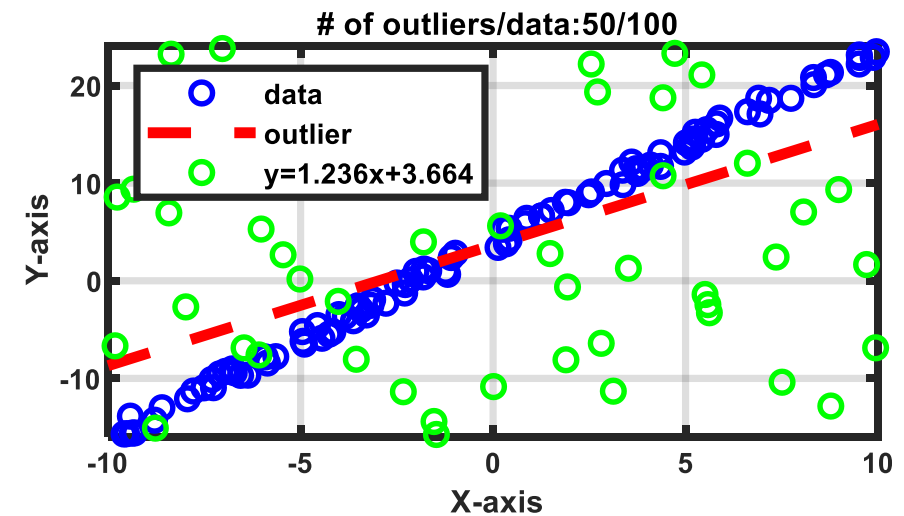
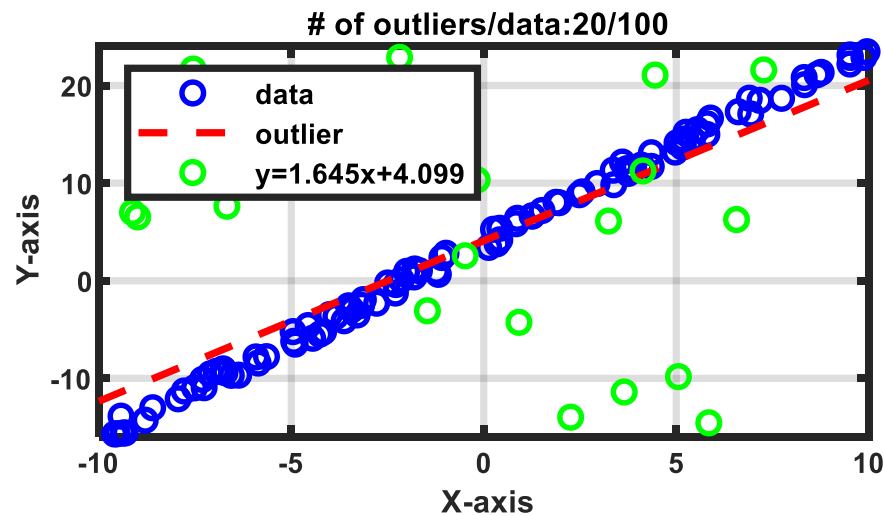
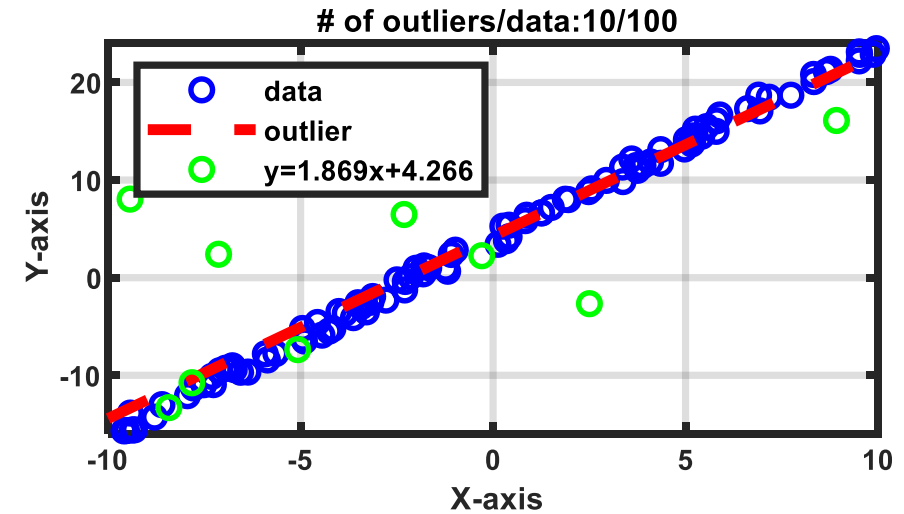
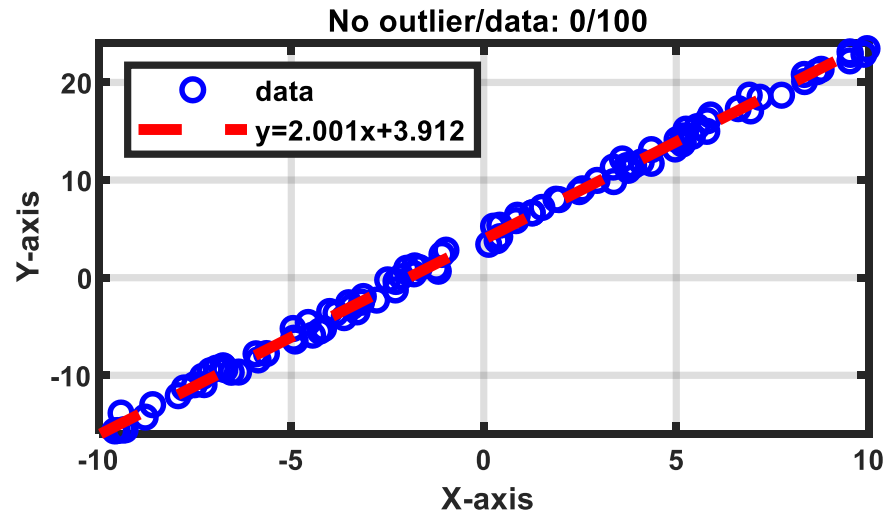
pseudoinverse



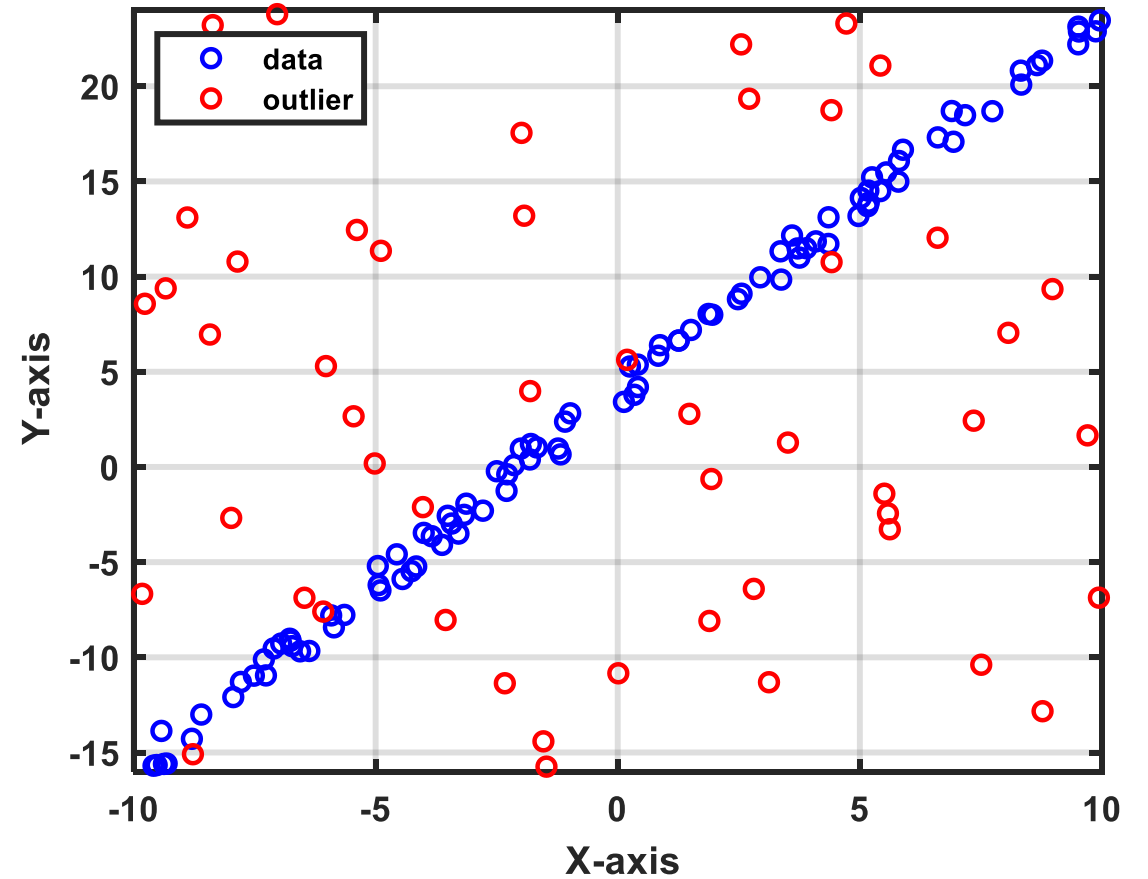
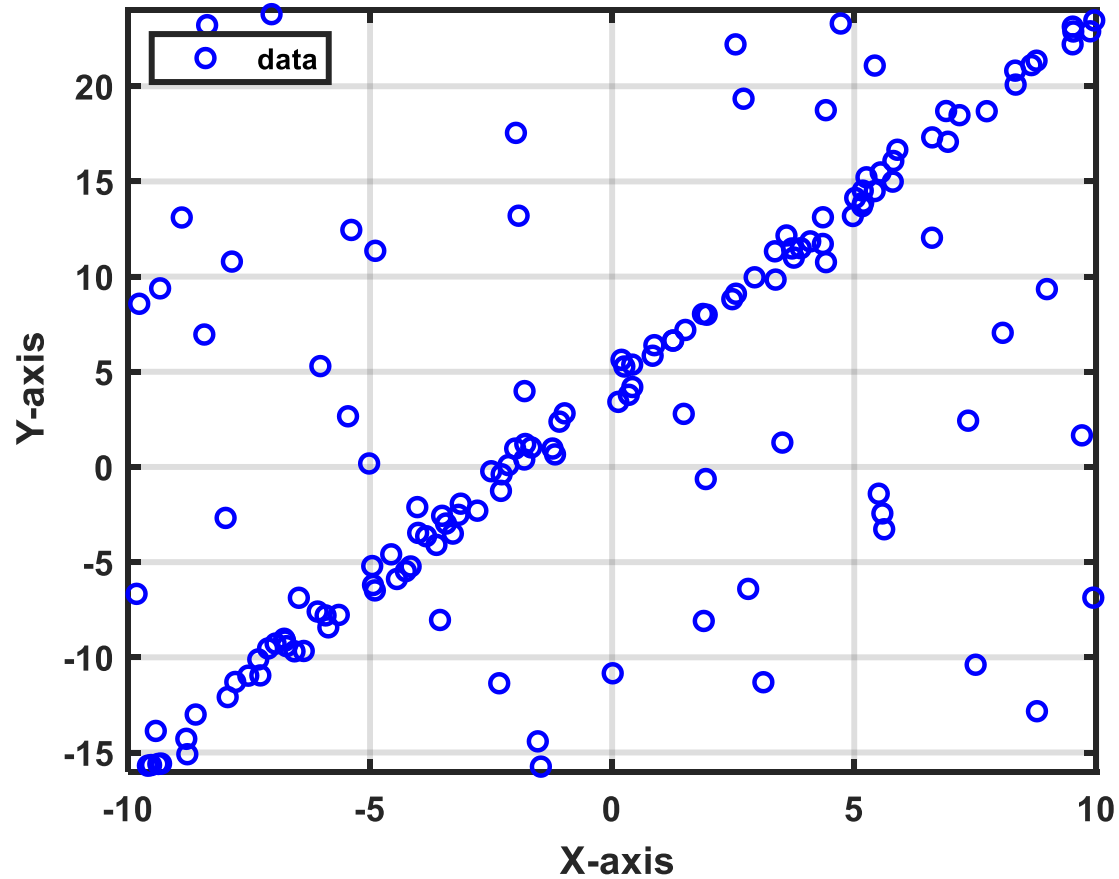
# What if There are Outliers in Your Data



# What if There are Outliers in Your Data (Continue)



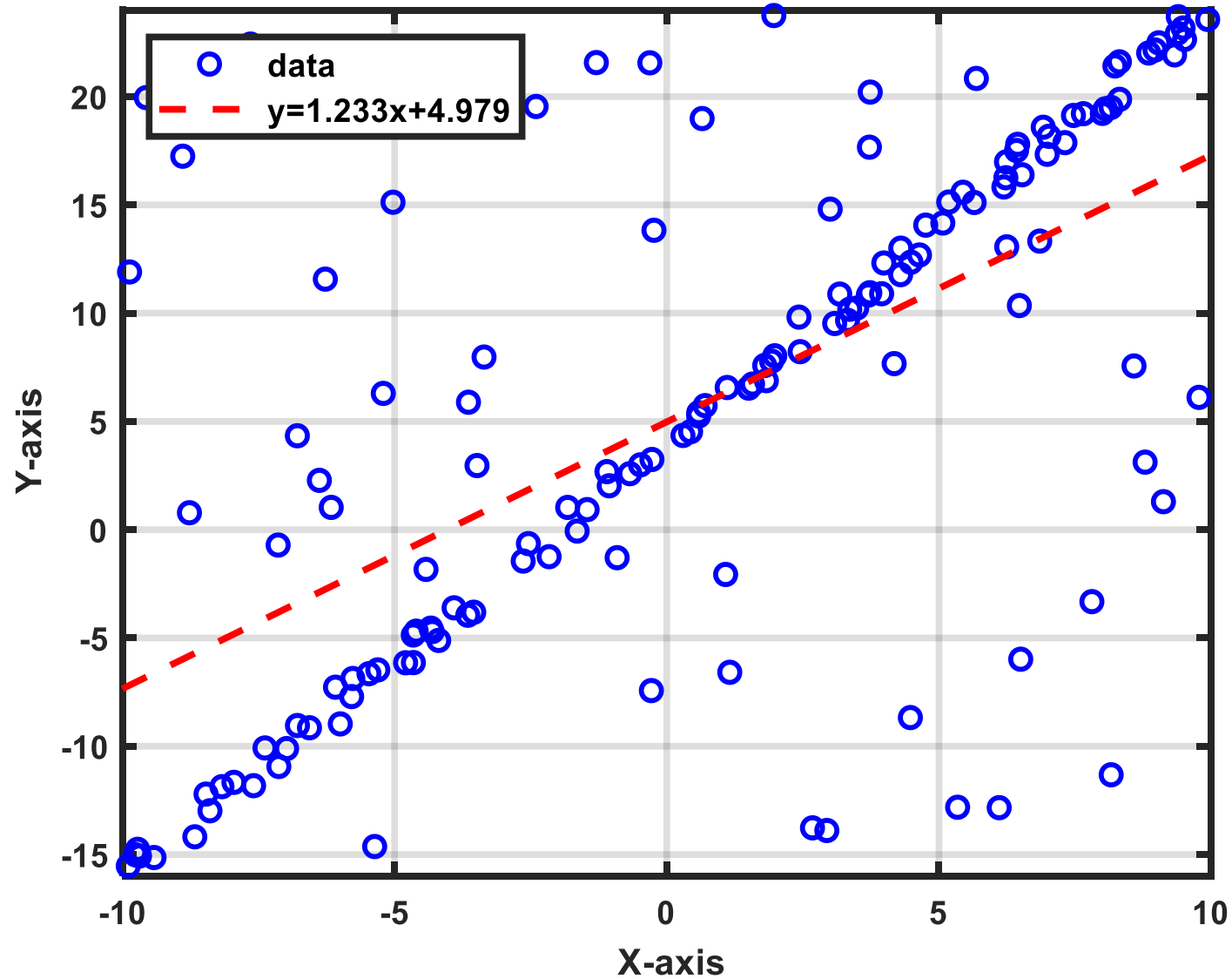
# What's Your Solution?



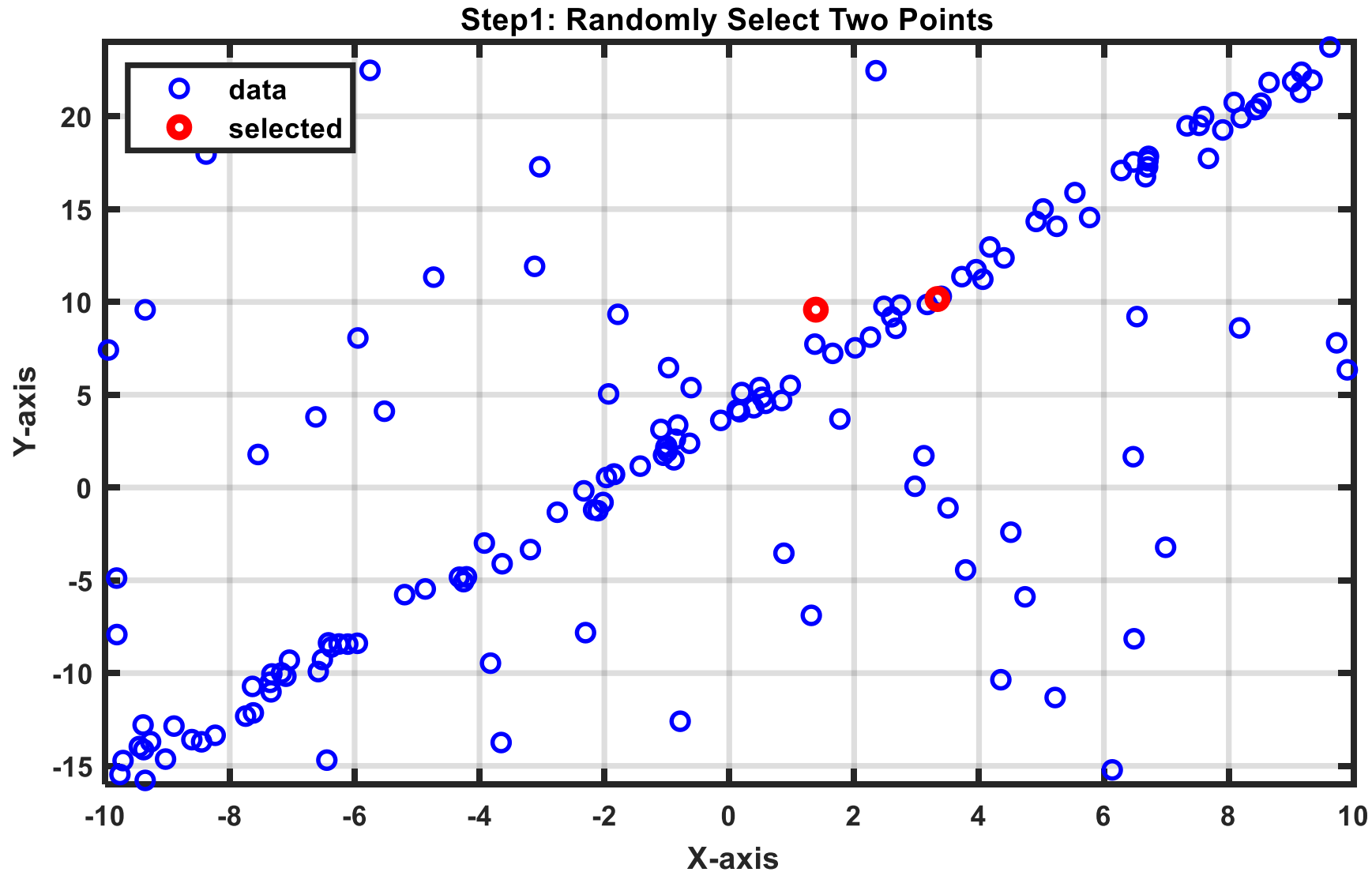
# Random Sample Consensus (RANSAC)

Fundamental to RANSAC algorithm is using randomly-selected least amount of data to construct an estimate (parameters of a known mathematical model) and to then ascertain the number of inliers from the rest of the data. We accept inliers only if the error (difference between the estimate and data) does not exceed a threshold. We repeat this process many times and choose the best estimate having the greatest number of inliers.

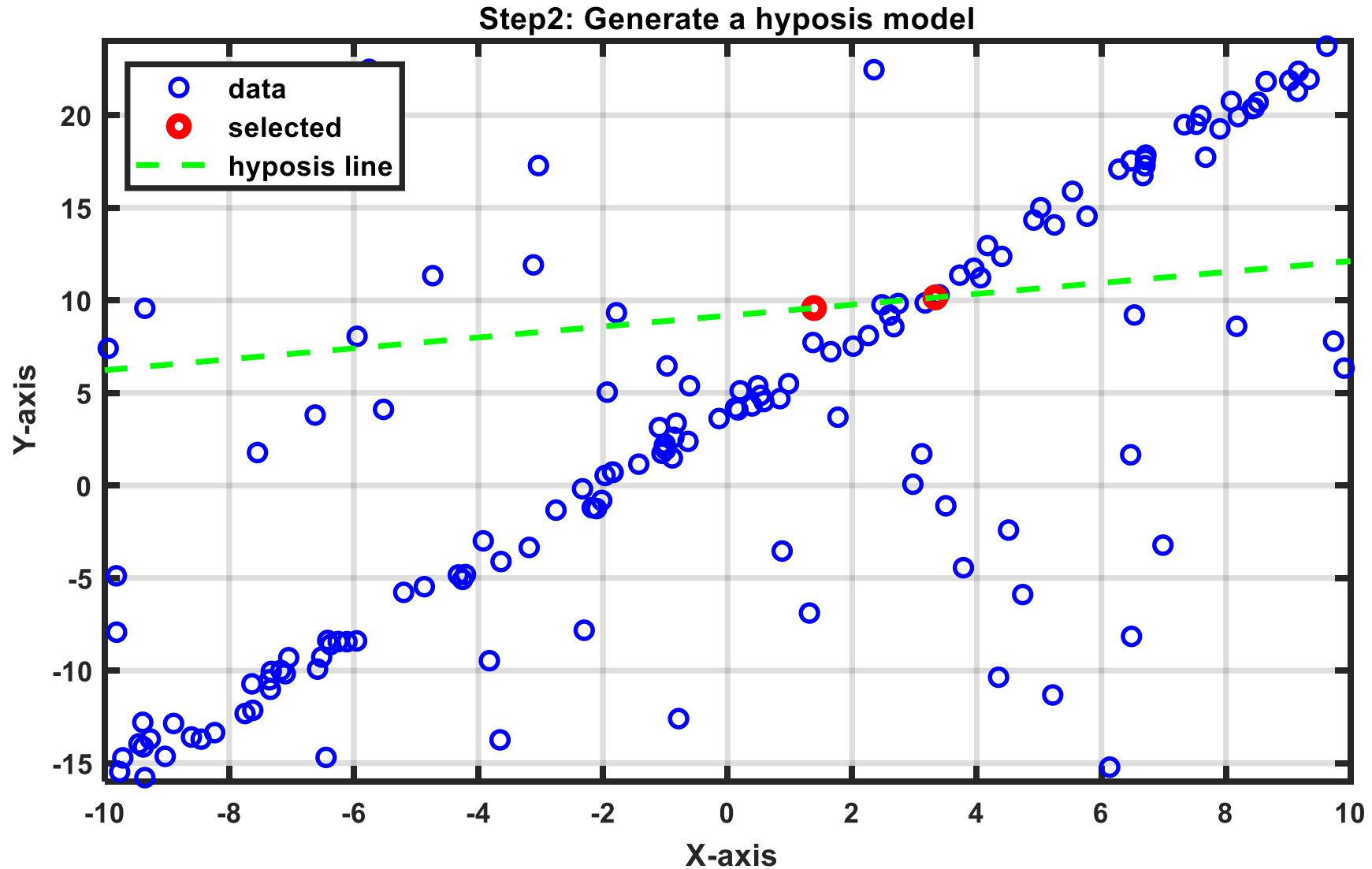
# Linear Fitting using Dataset Having Outliers



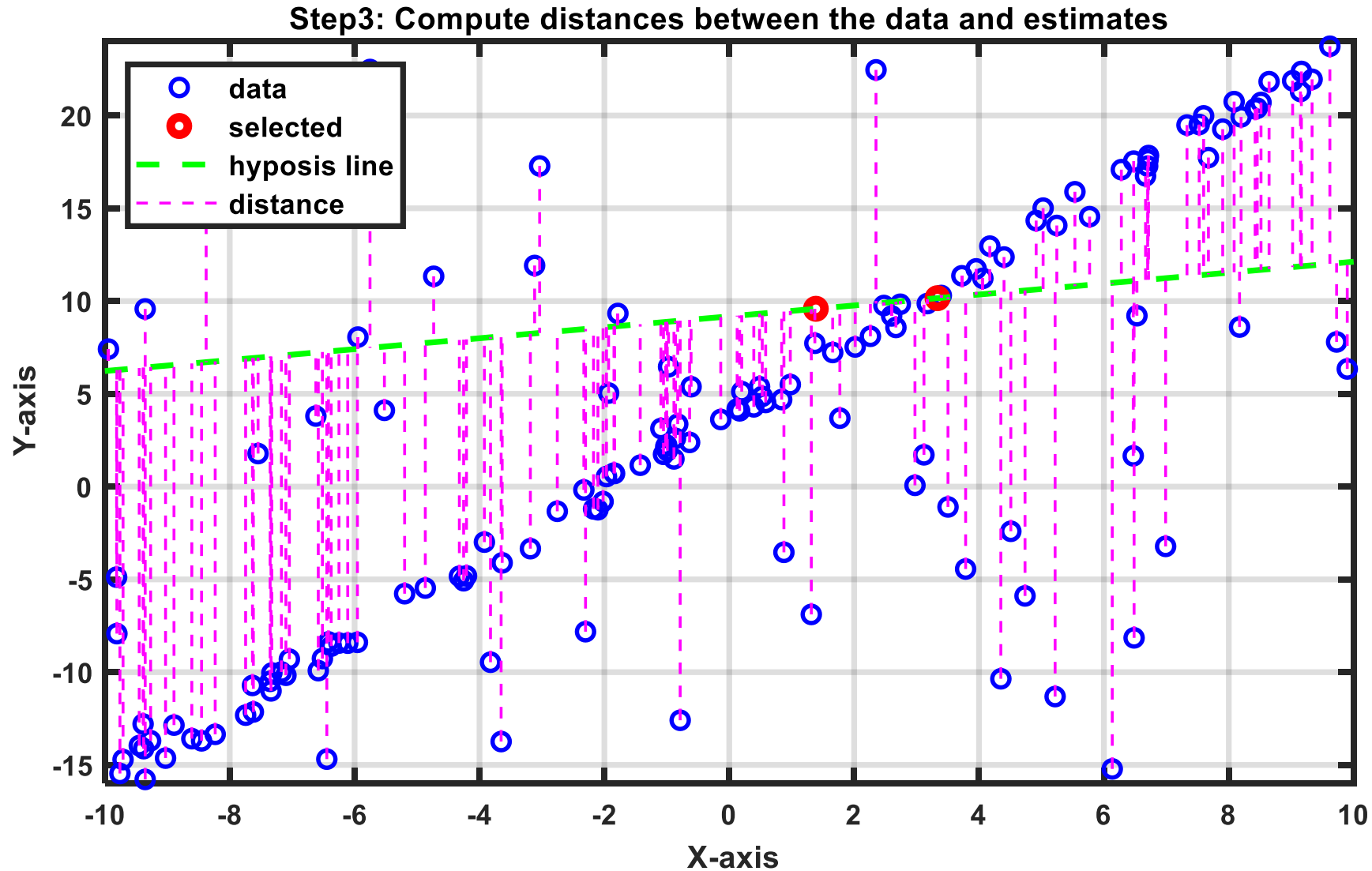
# Step 1: Randomly Select a Minimal Subset of Points



## Step 2: Generate a Hypothesis Model

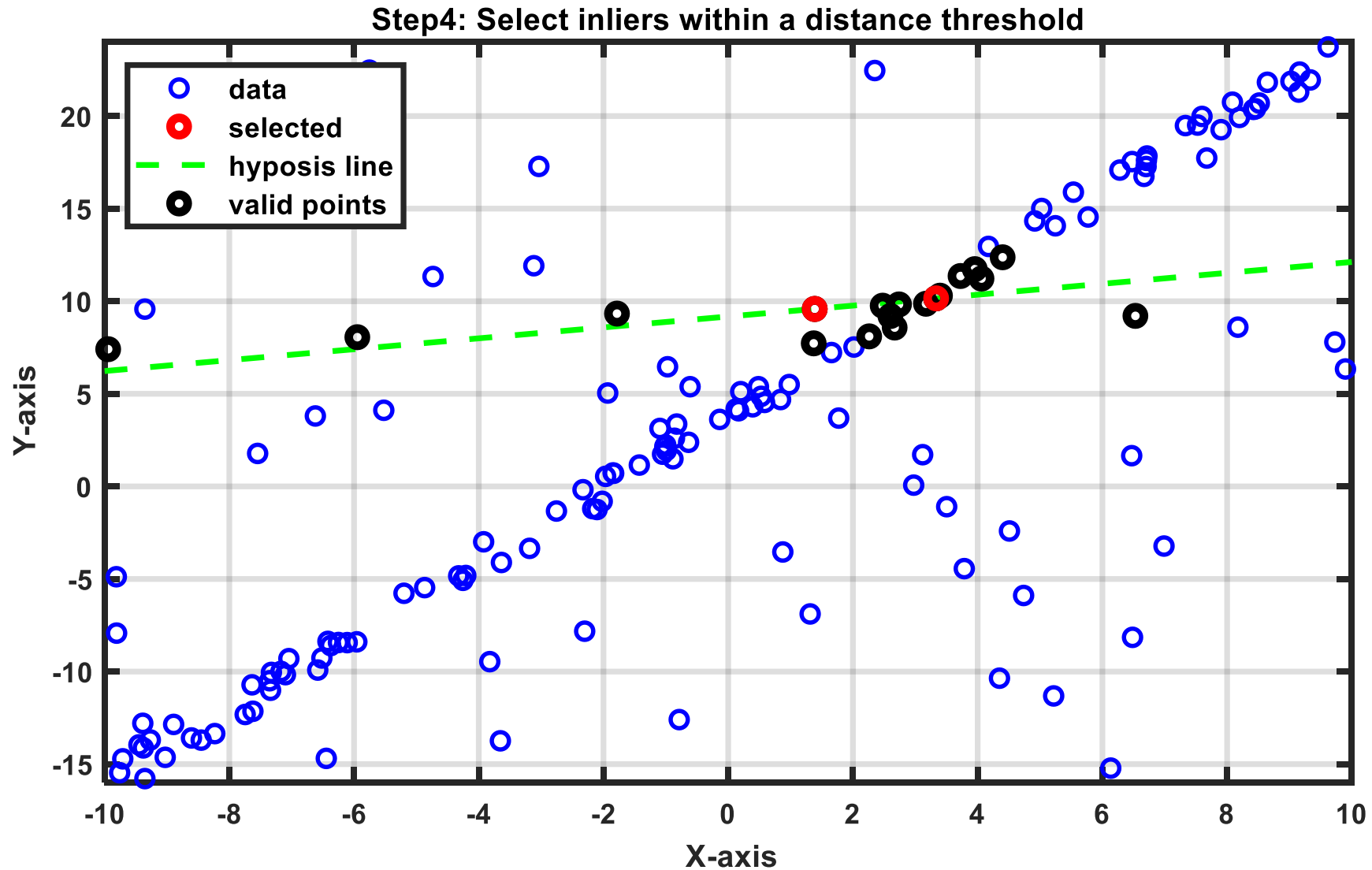


## Step 3: Compute Distances between the Data and Estimates from the Hypothesis Model

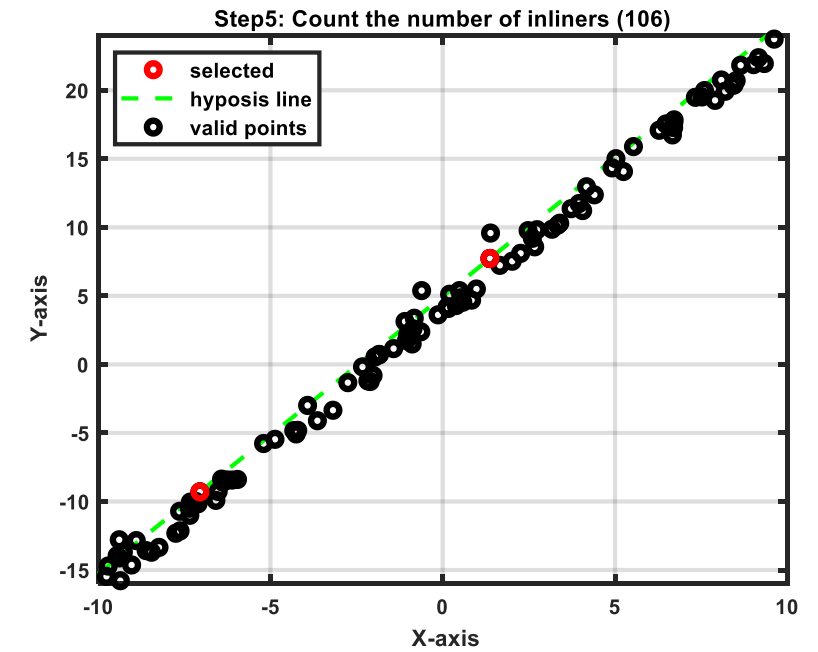
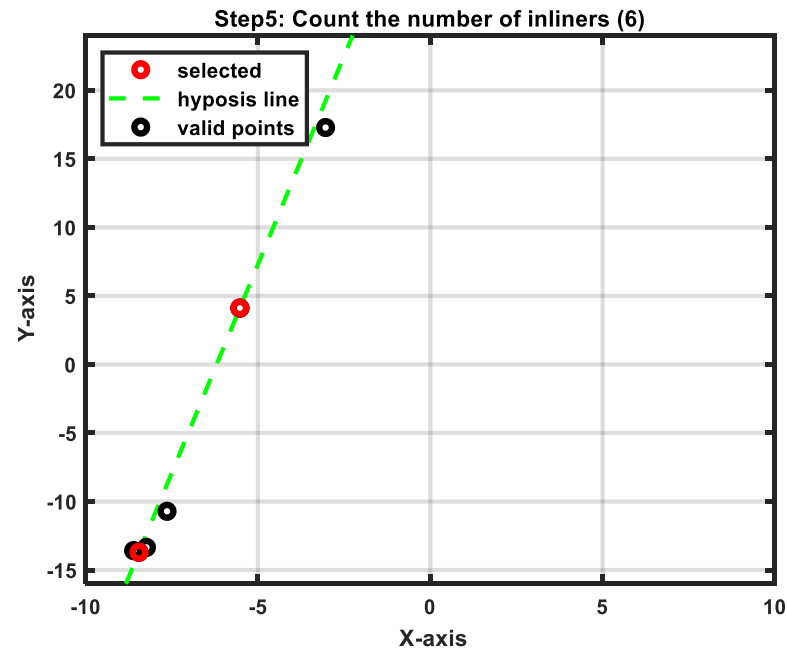
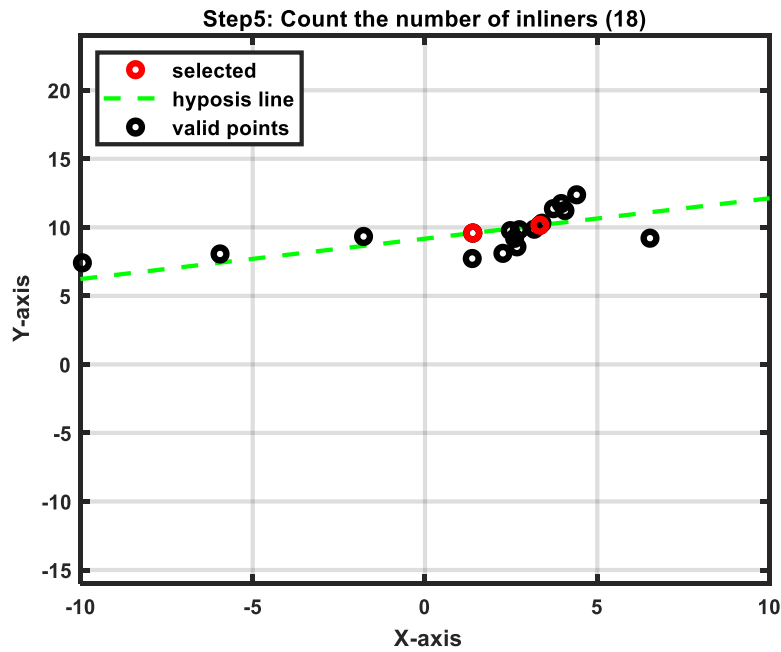




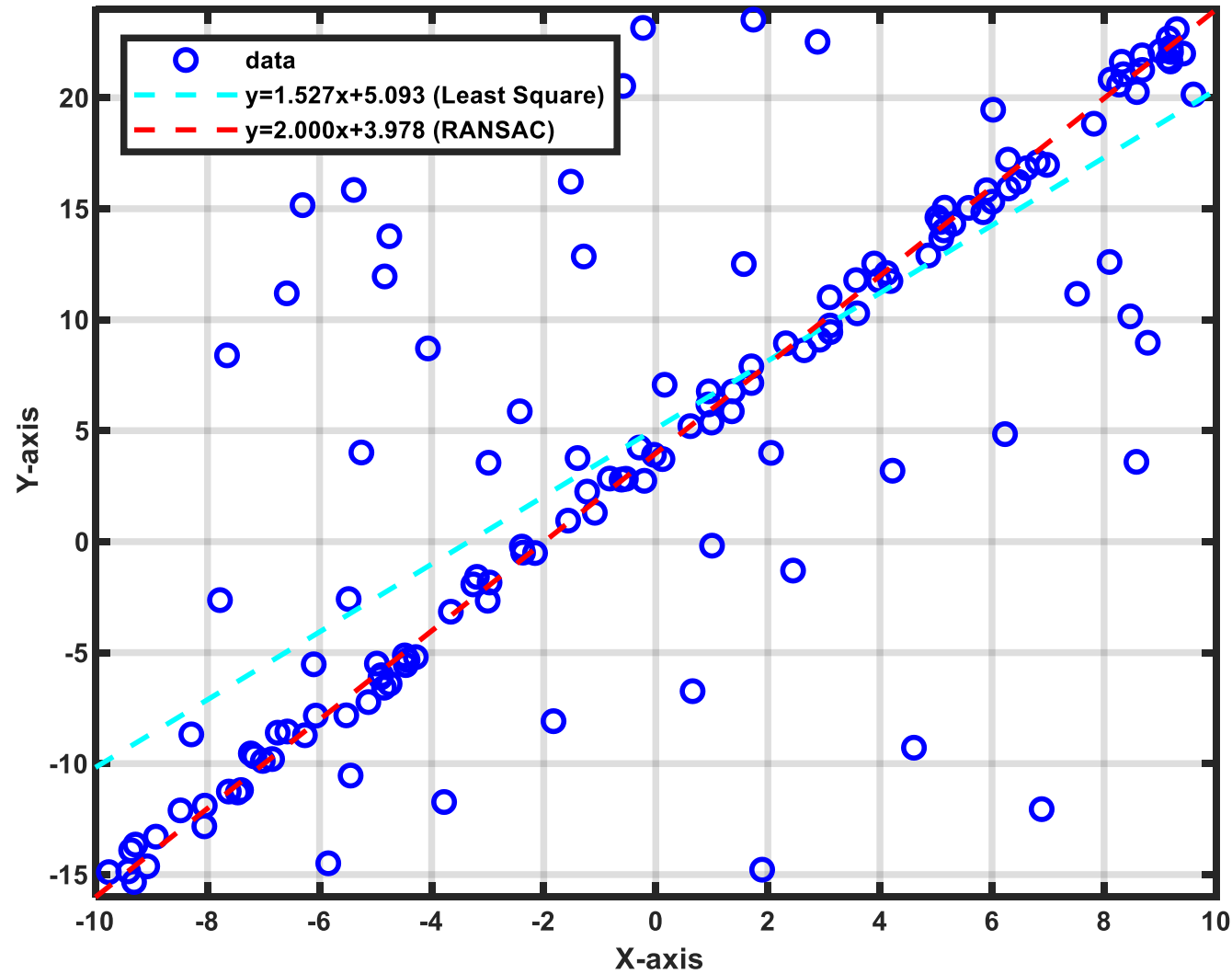
## Step 4: Select Inliers that are within a Distance Threshold



# Step 5: Repeat This Process and Count the Number of Inliers in each Hypothesis Model



# Demo: Line Fitting using RANSAC



[ransac\\_demo.mlx](#)

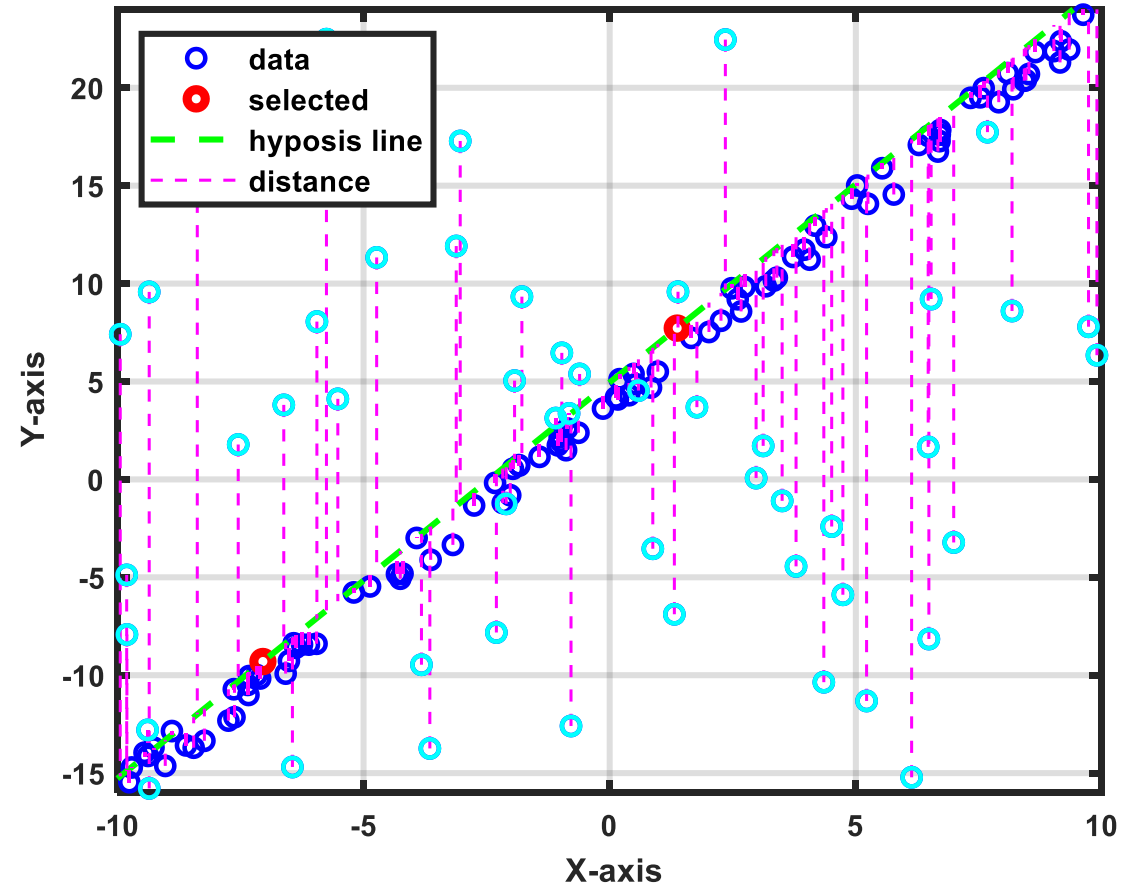
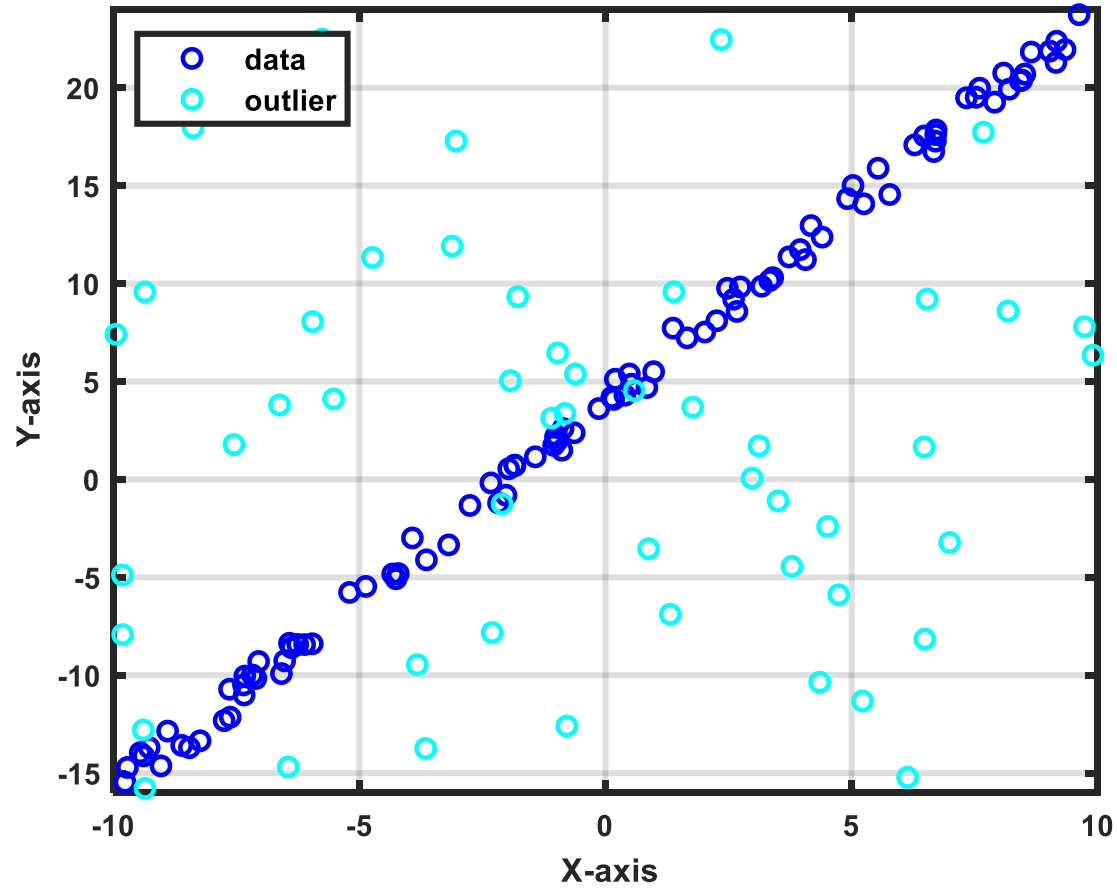
# Three Important Parameters for the Use of RANSAC

$\delta$ : Decision threshold to construct the inlier set.

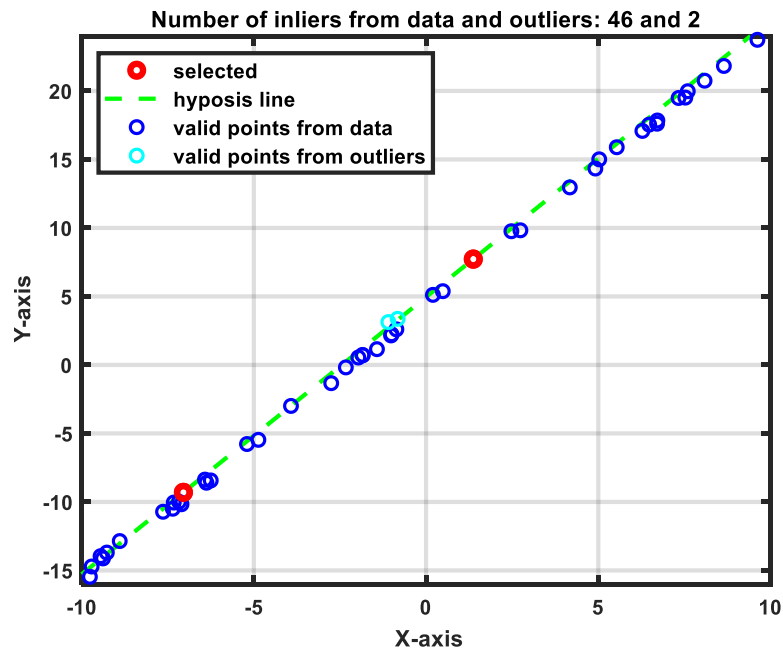
$N$ : Maximum number of trials to generate hypothesis models

$p$ : Confidence that the final solution finds the maximum number of inliers (finds a correct answer)

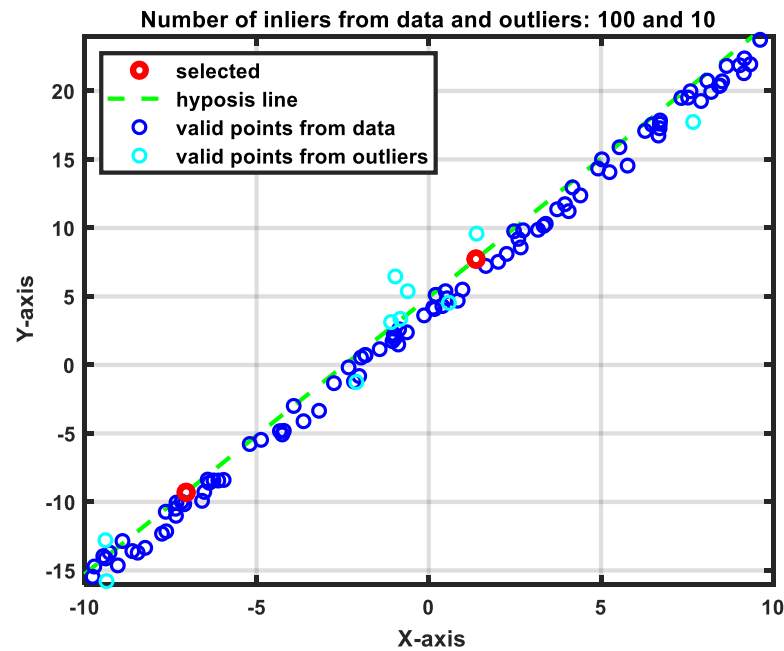
# Parameter: Decision Distance Threshold ( $\delta$ )



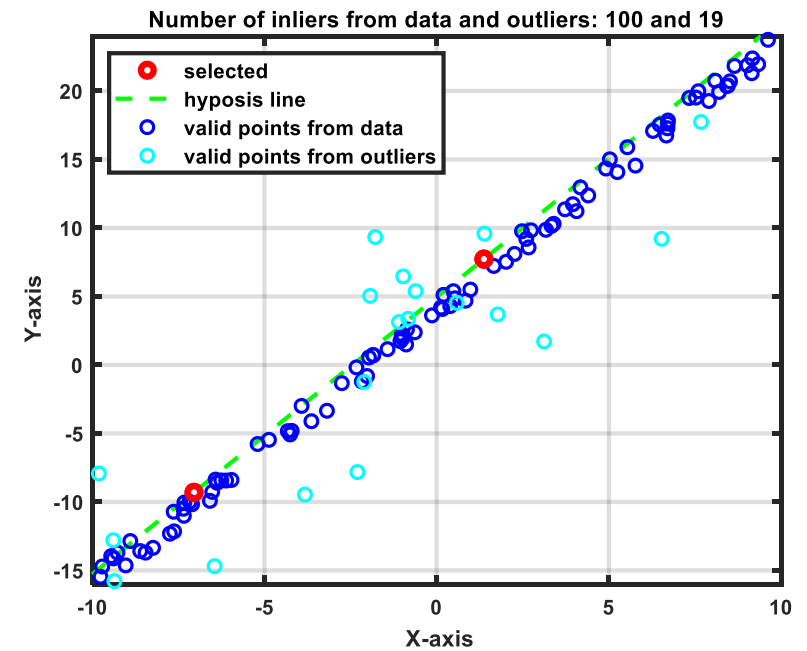
# Parameter: Decision Distance Threshold ( $\delta$ ) (Continue)



$\delta = 1$



$\delta = 4$



$\delta = 10$

# Parameter: Maximum Number of Trials (N) & Confidence (p)

Binomial combinations or all combinations

The binomial coefficient  $\binom{n}{k}$  is the number of ways of picking k unordered outcomes from n.

$$\binom{100}{2} = 4950 : 2 \text{ sample point picking among 100 data (line model)}$$

$$\binom{1000}{4} = 4e10 : 4 \text{ sample point picking among 1000 data (homography estimation)}$$

Computationally expensive

# Parameter: Maximum Number of Trials (N) & Confidence (p) (Continue)

$\epsilon$  : Probability of the false correspondence

$1 - \epsilon$  : Probability of the true correspondence

$(1 - \epsilon)^n$  : Probability of all n true correspondences in a trial

$1 - (1 - \epsilon)^n$  : Probability of at least one false correspondence in n correspondences in a trial

$(1 - (1 - \epsilon)^n)^N$  : Probability of at least one false correspondence in n correspondences in N trial

$1 - (1 - (1 - \epsilon)^n)^N$  : Probability of at least one trial that has all true n correspondence in N trial

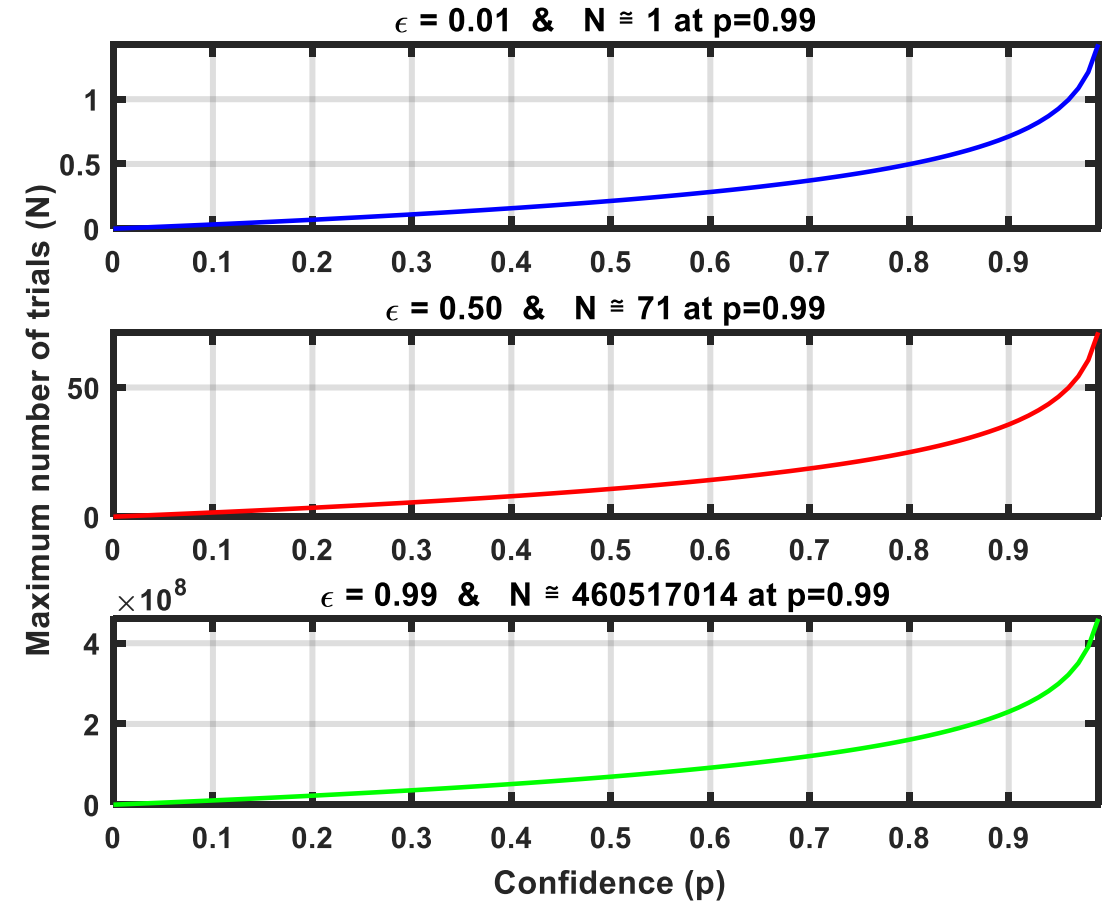
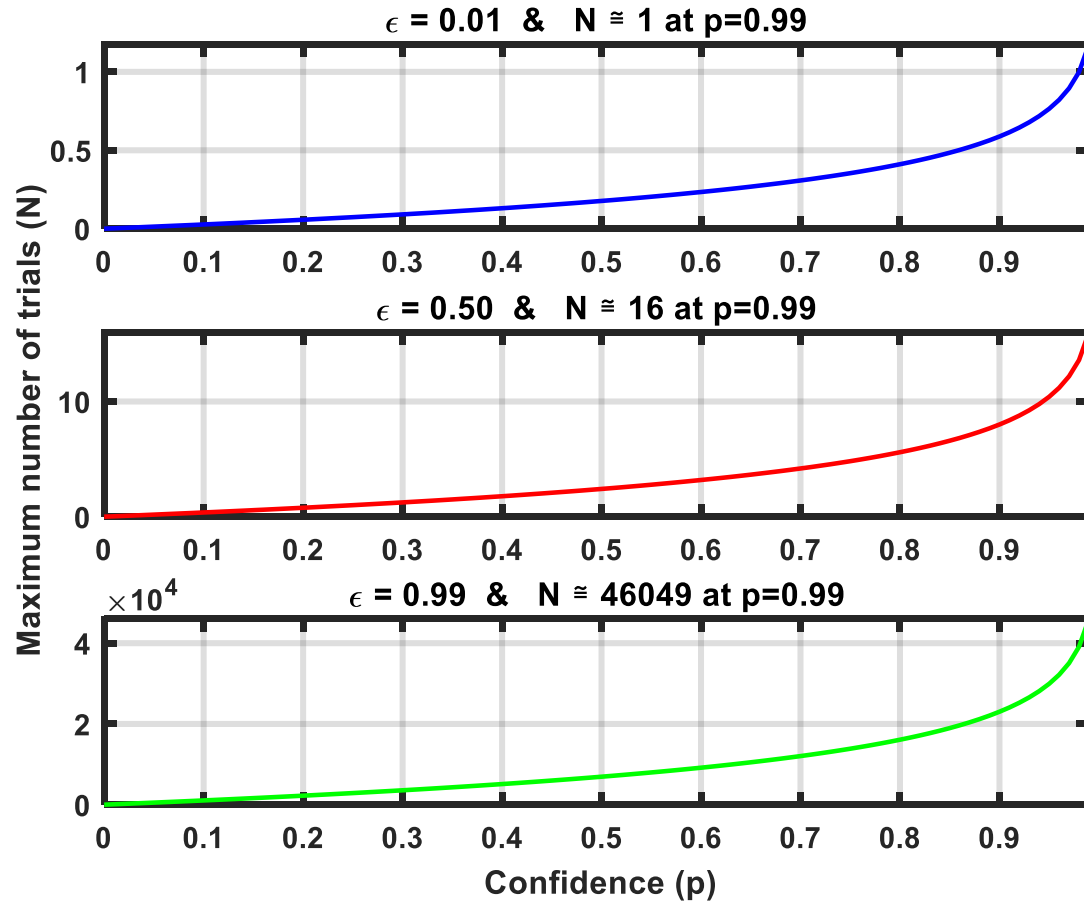
$$p = 1 - (1 - (1 - \epsilon)^n)^N$$

$$\ln(1 - p) = N \ln(1 - (1 - \epsilon)^n)$$

$$N = \frac{\ln(1 - p)}{\ln(1 - (1 - \epsilon)^n)}$$



# Parameter: Maximum Number of Trials (N) & Confidence (p) (Continue)



# How to Use a Confidence (p) Parameter in the Program (MATLAB)

```
while idxTrial <= numTrials && skipTrials < maxSkipTrials
    % Random selection without replacement
    indices = randperm(numPts, sampleSize);

    % Compute a model from samples
    samplePoints = allPoints(indices, :, :);
    modelParams = funcs.fitFunc(samplePoints, varargin{:});

    % Validate the model
    isValidModel = funcs.checkFunc(modelParams, varargin{:});

    if isValidModel
        % Evaluate model with truncated loss
        [model, dis, accDis] = evaluateModel(funcs.evalFunc, modelParams, ...
            allPoints, threshold, varargin{:});

        % Update the best model found so far
        if accDis < bestDis
            bestDis = accDis;
            bestInliers = dis < threshold;
            bestModelParams = model;
            inlierNum = cast(sum(dis < threshold), 'like', allPoints);
            num = vision.internal.ransac.computeLoopNumber(sampleSize, ...
                confidence, numPts, inlierNum);
            numTrials = min(numTrials, num);
        end

        idxTrial = idxTrial + 1;
    else
        skipTrials = skipTrials + 1;
    end
end
```

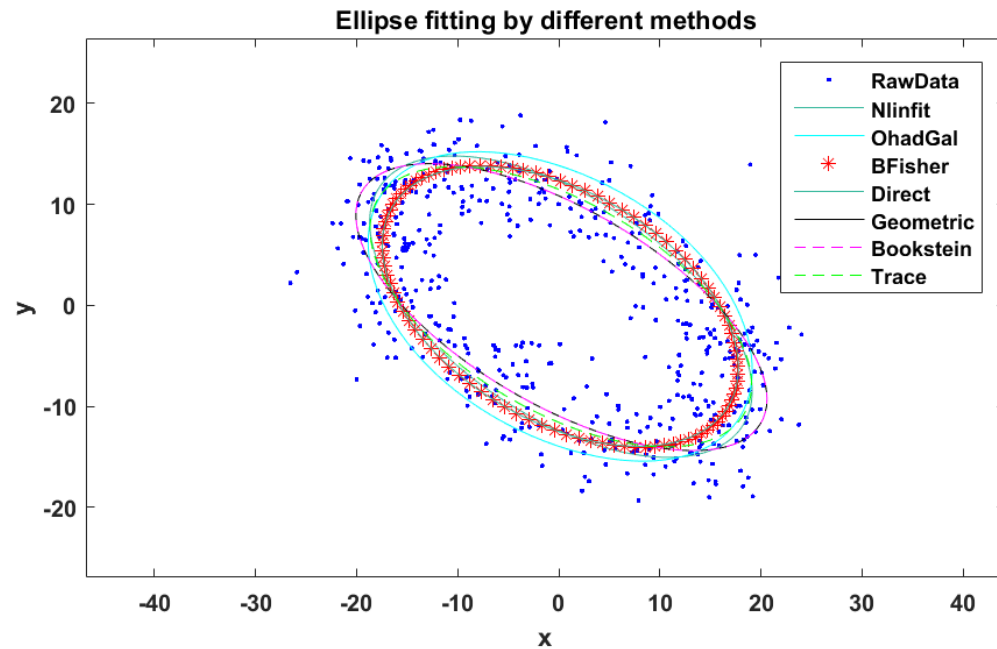
Early stop?

# Advantages and Disadvantage

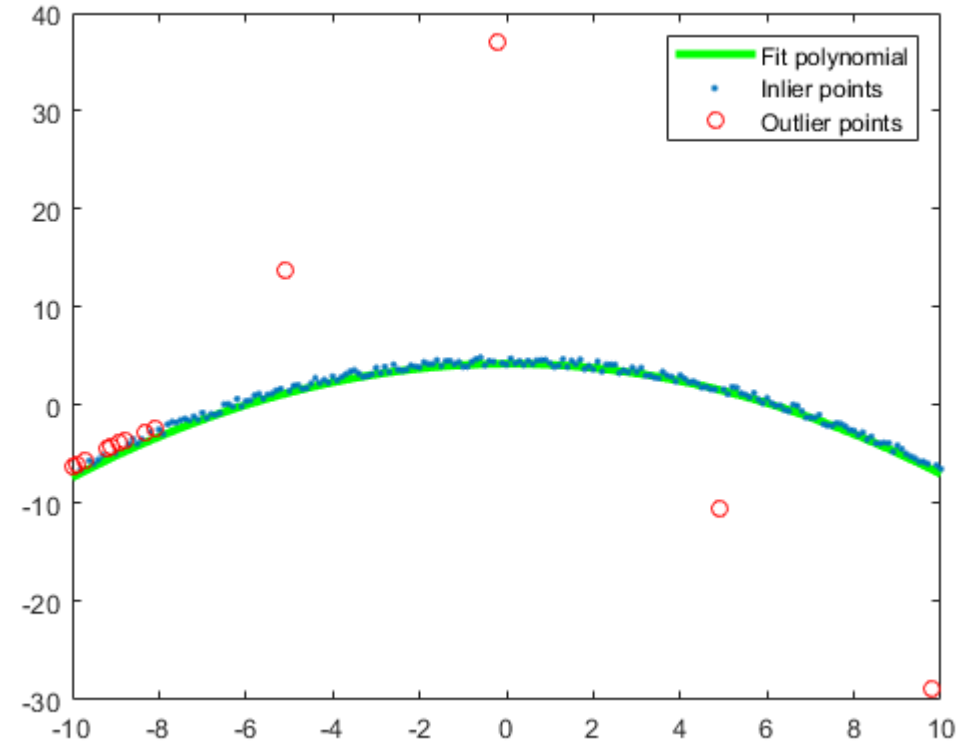
An advantage of RANSAC is its ability to do [robust estimation](#)<sup>[2]</sup> of the model parameters, i.e., it can estimate the parameters **with a high degree of accuracy even when a significant number of outliers are present in the data set**. A disadvantage of RANSAC is that there is **no upper bound on the time** it takes to compute these parameters (except exhaustion). When the number of iterations computed is limited the solution obtained may not be optimal, and it may not even be one that fits the data in a good way. In this way RANSAC offers a trade-off; by computing a greater number of iterations the probability of a reasonable model being produced is increased. Moreover, RANSAC is not always able to find the optimal set even **for moderately contaminated sets and it usually performs badly when the number of inliers is less than 50%**. Optimal RANSAC <sup>[3]</sup> was proposed to handle both these problems and is capable of finding the optimal set for heavily contaminated sets, even for an inlier ratio under 5%. Another disadvantage of RANSAC is that it requires the setting of **problem-specific thresholds**.

RANSAC can only estimate one model for a particular data set. As for any one-model approach when two (or more) model instances exist, RANSAC may fail to find either one. The [Hough transform](#) is one alternative robust estimation technique that may be useful when more than one model instance is present. Another approach for multi model fitting is known as PEARL,<sup>[4]</sup> which combines model sampling from data points as in RANSAC with iterative re-estimation of inliers and the multi-model fitting being formulated as an optimization problem with a global energy functional describing the quality of the overall solution.

# Application: Ellipse Fitting and Polynomial Fitting



<https://github.com/seisgo/EllipseFit>



<https://www.mathworks.com/help/vision/ref/fitpolynomialran.html>

# (Review) Point Correspondences for Estimating a Homography

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} \cong \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \cong \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

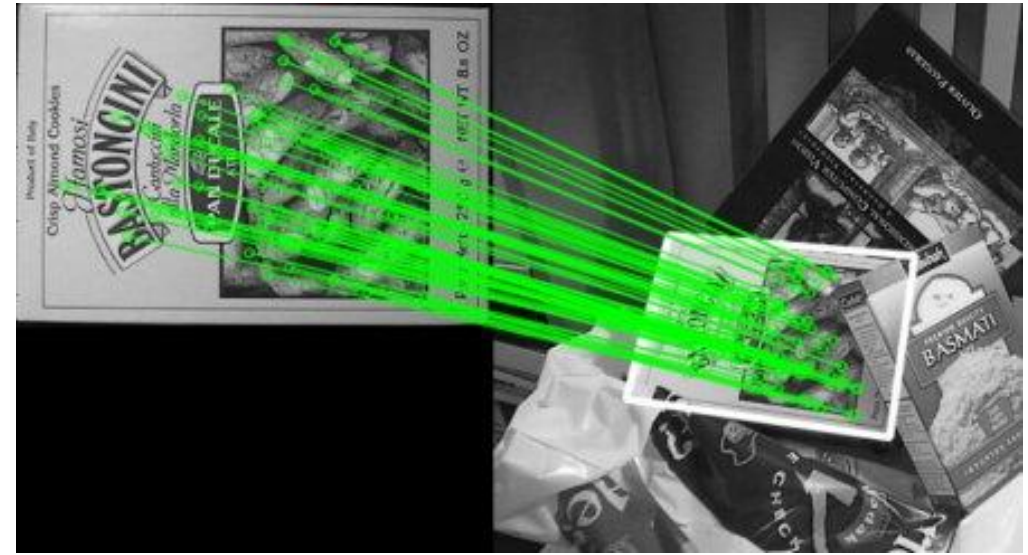
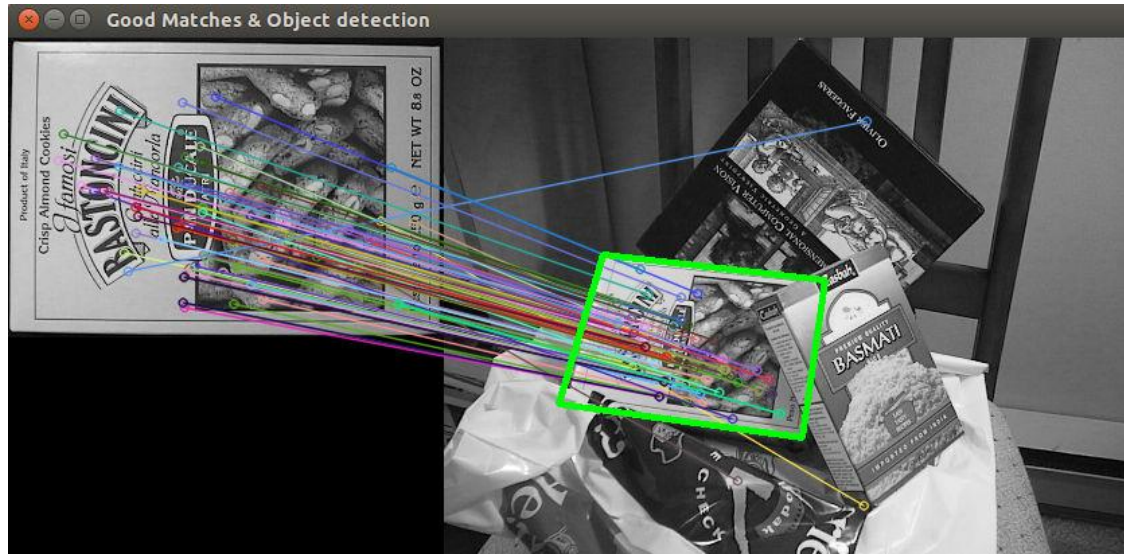
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \cong \begin{pmatrix} xh_{11} + yh_{12} + h_{13} \\ xh_{21} + yh_{22} + h_{23} \\ xh_{31} + yh_{32} + h_{33} \end{pmatrix}$$

$$x' = \frac{xh_{11} + yh_{12} + h_{13}}{xh_{31} + yh_{32} + h_{33}}$$

$$y = \frac{xh_{21} + yh_{22} + h_{23}}{xh_{31} + yh_{32} + h_{33}}$$

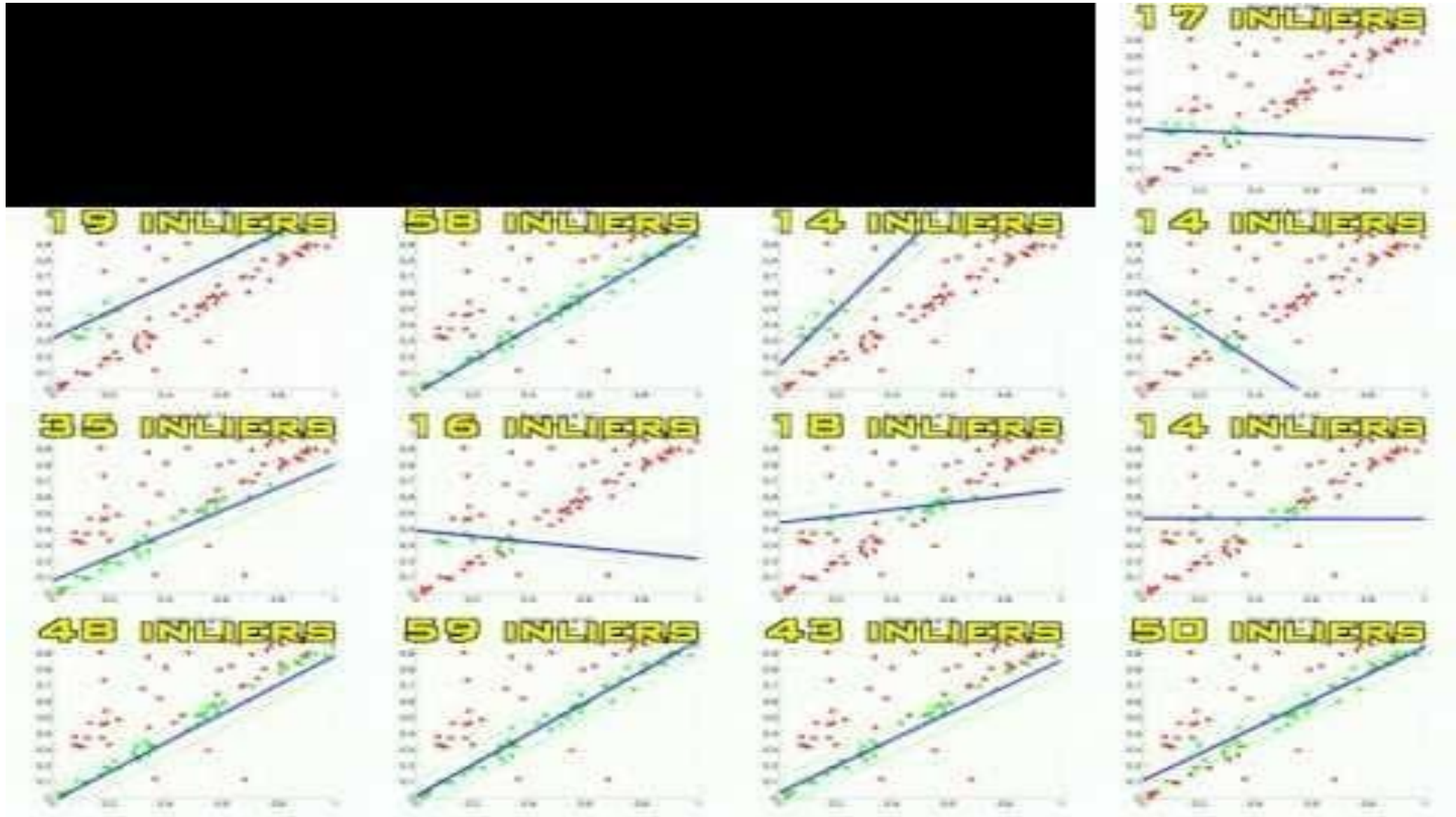
$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -xx' & -x'y & -x' \\ 0 & 0 & 0 & x & y & 1 & -xy' & -y'y & -y' \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0$$

# Homography Estimation





# RANSAC Song



# Slide Credits and References

- Lecture notes: Svetlana Lazebnik
- Lecture notes: S. Seitz
- Lecture notes: M. Pollefeys
- Lecture notes: A. Effros