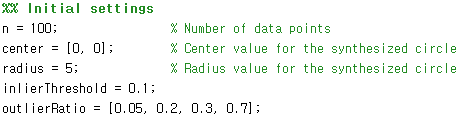
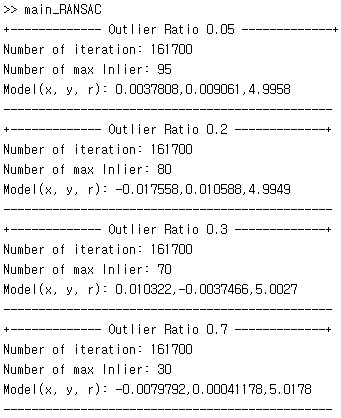
# EXERCISE 1: ROBUST ESTIMATION AND NORMS

GCT722 MATHEMATICAL METHODS FOR VISUAL COMPUTING  
20183151 Chaelin Kim

## PART 1: RANSAC FOR CIRCLE FITTING

### **Description of implementation**

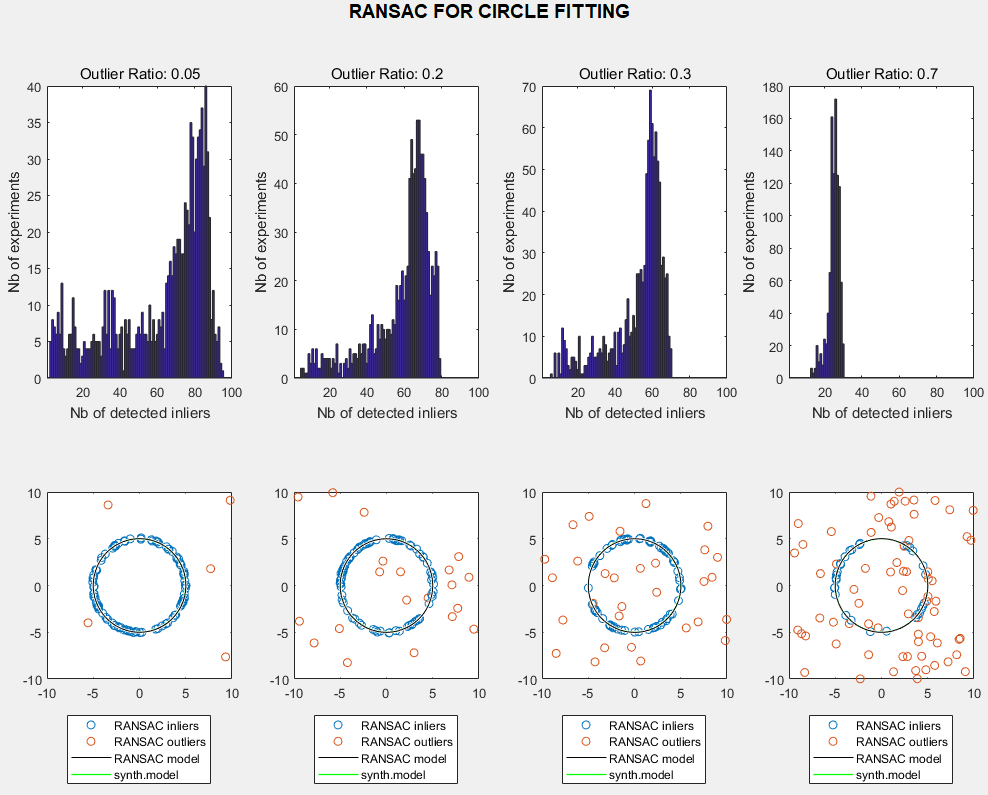
There are 1 script file & 4 function files for the exercise RANSAC.

* **Script file**
* main\_RANSAC.m  
  : This is the script for doing RANSAC for circle fitting. It calls functions for RANSAC (genCircleData, doRANSAC, doExhaustiveSearch, drawRANSACPlot). Details of these functions are explained below. We can set the initial settings in this script.  
  
* **Function files**
* genCircleData.m  
  : This function generates data points on a synthesized circle with inliers and outliers. The number of inliers and outliers are decided by the given outlier ratio. The inlier data is made with random noise (between -0.1 and 0.1). In the part of outlier generation, *while* loop is executed until the number of made outlier data meets the given number of outliers.  
  
  + Input Value
    - *n*: The number of data points
    - *center*: The center (x, y) value of the synthesized circle
    - *radius*: The radius value of the synthesized circle
    - *inlierThreshold*: The inlier distance threshold
    - *outlierRatio*: The ratio of outliers in data
  + Output
    - *data*: data in the form of circle with inliers and outliers according to the outlier ratio.
* doRANSAC.m  
  : This function runs RANSAC for the given data. In this function, the number of RANSAC iteration is computed and used in *for* loop for RANSAC (How to calculate this number is explained in Questions category below (What about the number of RANSAC iterations with r = 5%, 20%, 30% and 70%?)).  
  
  + Input Value
    - *data*: The data made from genCircleData function
    - *M*: The number of Iteration for re-apply RANSAC
    - *inlierThreshold*: The inlier distance threshold.
    - *outlierRatio*: Ratio of outliers in data
  + Output
    - *bestModel*: The computed data [center\_x\_value, center\_y\_value, radius]
    - *detectedInliers*: The number of computed inliers
* doExhaustiveSearch.m  
  : This function runs exhaustive search for the given data. It has the code for displaying the result of exhaustive searching  
  
  + Input Value
    - *data*: data made from genCircleData function
    - *k*: how many points are selected for searching
    - *inlierThreshold*: Inlier distance threshold
    - *outlierRatio*: Ratio of outliers in data
  + Output
    - No output.
    - 
* drawRANSACPlot.m  
  :  
  
  + Input Value
    - *center*: Center (x, y) value of the synthesized circle
    - *radius*: Radius value of the synthesized circle
    - *data*: data made from genCircleData function
    - *ransacResult*:
    - *histResult*:
    - *outlierRatio*: Ratio of outliers in data
  + Output
    - No output

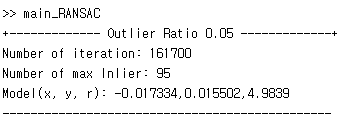
**Instructions for running**

1. Open the file “main\_RANSAC.m” in Matlab.
2. Execute that file.
   1. You can see the result of exhaustive search in command window.
   2. The another window that shows plots of RANSAC results is opened.

**Screenshots**



**Questions**

* How many combinations (exhaustive search) exist for N = 100 points?
* Ans: 161700  
    
   
* What about the number of RANSAC iterations with r = 5%, 20%, 30% and 70%?
* Ans  
    
  🡪 
* What about when N = 10,000 points?
* Exhaustive search
* RANSAC iterations

**Discuss the results**

## PART 2: IRLS AND NORMS FOR LINE FITTING

**Description of implementation**

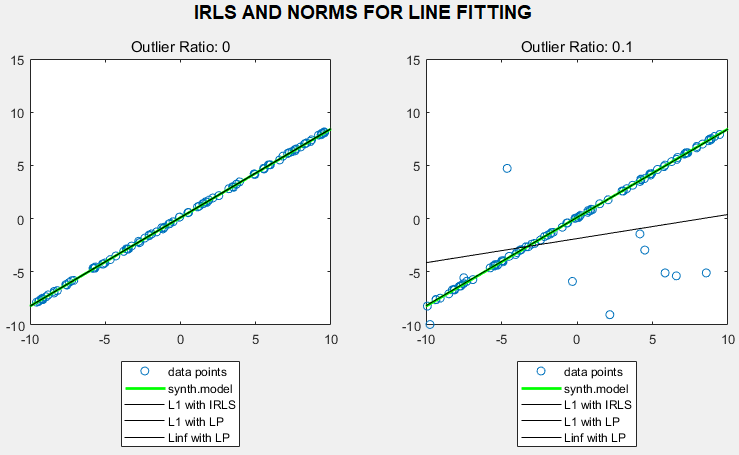
There are 1 script file & 4 function files for the exercise Line fitting.

* Script file
* main\_LineFitting.m
* Function files
* genLineData.m
* doIRLS.m
* doLP.m
* drawLineFittingPlot.m

**Instructions for running**

1. Open the file “main\_LineFitting.m” in Matlab.
2. Execute that file
   1. The another window that shows plots of IRLS with and LP with  and  norms results is opened.

**Screenshots**



**Discuss the results**