Exercise 1. Robust Estimation and Optimization

Dongho Kang

March 12, 2017

The "code" directory contains followings:

- ***part1.m*** Script .m file for part1.
- ***part2.m*** Script .m file for part2.
- ***functions dir*** Function .m files
    - ...
    - ...

1 EXERCISE PART 1: RANSAC FOR CIRCLE FITTING

1.1 DESCRIPTION

In exercise part 1, RANSAC algorithm was used for circular model fitting with N number of 2D data points which were corrupted by noise and certain ration of outliers. The implementation of part 1 corresponds to the .m script file ***part1.m*** and the script consists of several parts which can be executed separately:

- Data generation and verification
- Run RANSAC algorithm for circular model fitting
- Run exhaustive search for circular model fitting
- Plot the result

1.1.1 Data generation and verification

For certain outlier ratio r(%) and N number of 2D data points, the number of inliers $(n_{inlier})$and outliers$(n_{outlier})$ are as follows:

$$\begin{aligned} n_{inlier} &= N \times \tfrac{1-r}{100} \\ n_{outlier} &= N \times \tfrac{r}{100} \end{aligned}$$

$$N = 100,\ r \in \{5, 20, 30, 70\}$$

In order to generate n_{inlier} inliers and $n_{outlier}$ outliers, the function **GenerateInlierData** and **GenerateOutlierData** are implemented. **GenerateInlierData** generates noise-added data points which is not over the inlier distance threshold $\tau = 0.1$ from synthetic model and **GenerateOutlierData** generates data points which is over τ. It can be described as follows. $d\big(p_1, p_2\big)$ is euclidean distance function between two 2D points $p_1 and p_2$:

$$\begin{aligned} \Big|d\big((x_i, y_i), (x_c, y_c)\big) - R_c\Big| &\leq \tau \qquad \text{for } (x_i, y_i) \in Set_{inlier} \\ \Big|d\big((x_j, y_j), (x_c, y_c)\big) - R_c\Big| &> \tau \qquad \text{for } (x_i, y_i) \in Set_{outlier} \end{aligned}$$

After generating inliers/outliers, verified whether generated inliers is threshold and also generated outliers . For this purpose, the function **VerifyInlier** is used. The specific descriptions for each functions are following:

Inlier generation   A 2D point $(\tilde{x_i}, \tilde{y_i})$ on the circular model $(x - x_c)^2 + (y - y_c)^2 = R_c^2$ which of center (x_c, y_c), and radius R_c can be described as follows:

$$(\tilde{x_i}, \tilde{y_i}) = (x_c, y_c) + R_c \times (\cos\theta_i, \sin\theta_i) \qquad \theta_i \in [0, 2\pi],\ i \in \{1, 2, \ldots n_{inlier}\}$$

Thus, picked n_{inlier} number of random float numbers from the range of $[0, 2\pi]$ by **rand** and as using these numbers as θ_i, generated (x_i, y_i) for $i \in \{1, 2, \ldots n_{inlier}\}$.
Random noise $(\sigma_{x_i}, \sigma_{y_i})$ is added on the data point (x_i, y_i) as follows:

$$\begin{gathered} (x_i, y_i) = (\tilde{x_i}, \tilde{y_i}) + (\sigma_{x_i}, \sigma_{y_i}) \\ \text{s.t. } \sigma_{x_i}, \sigma_{y_i} \in [-0.1, 0.1], \quad \big|d((\tilde{x_i}, \tilde{y_i}), (x_c, y_c)) - R_c\big| \leq \tau \end{gathered}$$

To implement this, two random float numbers by **rand** were picked for $(\sigma_{x_i}, \sigma_{y_i})$ iteratively, until it satisfies$\big|d((x_i, y_i), (x_c, y_c)) - R_c\big| \leq \tau$ and repeated it for $i = 1 ... n_{inlier}$ to generate n_{inlier} number of inliers.

outlier generation   In order to generate one outlier point (x_j, y_j), two random float numbers were generated by **rand** in the domain of $[-10, 10] \times [-10, 10]$ and generated (x_j, y_j) was checked whether it satisfies $\big|d((x_i, y_i), (x_c, y_c)) - R_c\big| > \tau$. If not, generated (x_j, y_j) again. Repeated this for $n_{outlier}$ times to generate $n_{outlier}$ number of outliers.

DATA VERIFICATION Checked whether synthesized inliers are indeed inliers and outliers are indeed outliers. The function **VerifyInlier** was used. The function calculates $\left|d((\tilde{x}_i,\tilde{y}_i),(x_c,y_c))-R_c\right|$ for each data points and check if there is a inlier point which of $\left|d((\tilde{x}_i,\tilde{y}_i),(x_c,y_c))-R_c\right| > \tau$ or an outlier point which of $\left|d((\tilde{x}_i,\tilde{y}_i),(x_c,y_c))-R_c\right| \leq \tau$. In this case, error is returned and the script halts.

1.2 Running

parameters debug

1.3 Result

$$\begin{aligned}(x+y)^3 &= (x+y)^2(x+y)\\ &= (x^2+2xy+y^2)(x+y)\\ &= (x^3+2x^2y+xy^2)+(x^2y+2xy^2+y^3)\\ &= x^3+3x^2y+3xy^2+y^3\end{aligned} \tag{1.1}$$

Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies

1.4 Heading on level 2 (subsection)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

$$A = \begin{bmatrix} A_{11} & A_{21} \\ A_{21} & A_{22} \end{bmatrix} \tag{1.2}$$

Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem.

1.4.1 Heading on level 3 (subsubsection)

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

HEADING ON LEVEL 4 (PARAGRAPH) Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

2 LISTS

2.1 EXAMPLE OF LIST (3*ITEMIZE)

- First item in a list
    - First item in a list
        - First item in a list
        - Second item in a list
    - Second item in a list
- Second item in a list

2.2 EXAMPLE OF LIST (ENUMERATE)

1. First item in a list
2. Second item in a list
3. Third item in a list