Assignment 3: Perfecting Your Drawing

2021 Fall EECS205002 Linear Algebra

Due: 2021/12/15

Many drawing tools, such as iPad, have a function that when you are drawing a shape by hand, they will guess what it is, and make the best fit one to replace it. For example, if you draw a lopsided circle, it will produce a perfect one for it.

In this assignment, you will learn how to do that using the magic from linear algebra. Figure 1 shows to examples. Figure 1(a) and Figure 1(c) are hand drawn shapes, and Figure 1(b) and Figure 1(d) are the drawn shapes with the best fit circle. The used algorithm is based on the least square method, as the application 3 in the textbook Section 5.3 for Coordinate Metrology. To fit a circle,

$$(x - c_1)^2 + (y - c_2)^2 = r^2, \tag{1}$$

we need to find out what c_1, c_2 and r are so that

$$f(c_1, c_2, r) = \sum_{i=1}^{m} \left((x_i - c_1)^2 + (y_i - c_2)^2 - r^2 \right)^2$$

is minimum for the given data $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$.

The above problem is nonlinear, which is hard to solve. But we can approximate it by rewriting (1) with

$$2c_1 x + 2c_2 y + c_3 = x^2 + y^2, \tag{2}$$

for $c_3 = r^2 - c_1^2 - c_2^2$, and making c_1, c_2, c_3 independent variables.

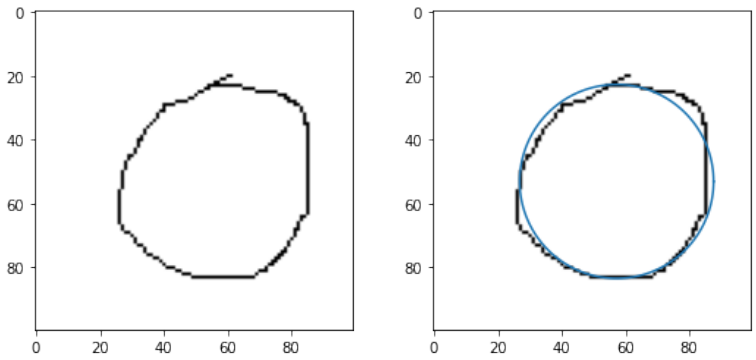
If you have learned the least square problem in Sec 5.3, you will know how to transform the $\min_{c_1,c_2,c_3} f(c_1, c_2, c_3)$ into matrix form. Let

$$A = \begin{bmatrix} 2x_1 & 2y_1 & 1 \\ 2x_2 & 2y_2 & 1 \\ \vdots & \vdots & \vdots \\ 2x_m & 2y_m & 1 \end{bmatrix}, x = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, b = \begin{bmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ \vdots \\ x_m^2 + y_m^2 \end{bmatrix}.$$

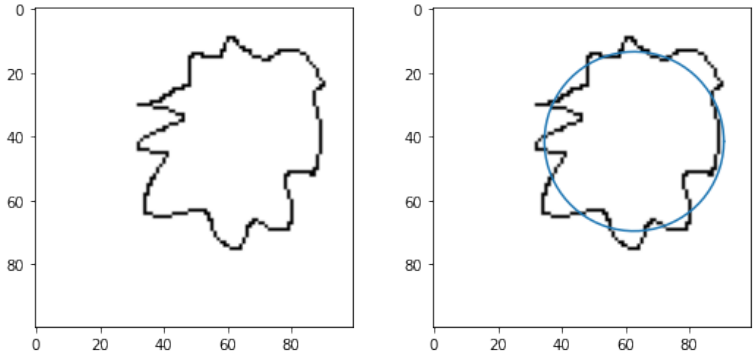
The original problem is to solve an over-determinded system, $Ax = b$.

The problem is also called a least square problem, which can be solved by the normal equation

$$A^T A x = A^T b. \tag{3}$$



(a) Drawn circle (b) Fitted circle



(c) Shaky circle (d) Fitted circle

Figure 1: Example of circle fitting.

1 Assignments

1. (10%) Hand draw at least 3 "circle" plots of resolution 100×100 pixels, and use them as inputs for the given program.

2. (30%) The given program uses `numpy.linalg.lstsq` to solve the least square problem.

   (a) Write a python code to implement the normal equation, as shown in (3), and use your own drawing to compare the results with those by `numpy.linalg.lstsq`.

   (b) How to measure the closeness of your drawings to a perfect circle? For example, the drawing in Figure 1(a) is better than that of Figure 1(c), because it is more close to a perfect circle. How to obtain this measurement from the returned values of `numpy.linalg.lstsq`? And how to obtain it from the normal equation you implemented?

   (c) There is a return value of `numpy.linalg.lstsq` called `rank`. What

are the ranks of your examples? Create an image whose rank output by `numpy.linalg.lstsq` is less than 3.

3. (20%) Design and implement an algorithm to find the best fitting ellipse using least square algorithm. Test the program with three hand drawing ellipses. You can use this formula,

$$c_1x^2 + c_2xy + c_3y^2 + c_4x + c_5y = 1.$$

   The formula can actually fit all kinds of curves from conic section. We will discuss the details later.

4. (20%) The given code uses all the drawn points for circle fitting, which is too many. Design and implement an algorithm to reduce the number of points for the input of least square, so that the computed circle is similar to the one computed using all data points. Note your algorithm CANNOT compute the solution using all data, including the center and the radius, for reference. Your algorithm will be judged from three aspect.

   (a) Efficiency: which is the time complexity of your algorithm. You should provide a simple analysis for that.

   (b) Correctness: how close if the solution using the sample data given by your code to that of using all data?

   (c) The number of sampled data points, which is the few the better.

5. (20%) Look up what *Voronoi diagram* and *Farthest Point Voronoi Diagram* are, and how to use them to find the minimum enclosing circle for the given points. Explain how it works, and implement the algorithm. (See `https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Voronoi.html`)

2 Submission

1. Write a report in PDF file that includes the answers of question (1), (2), (3), (4), and (5), including the figures of your hand drawing circles and ellipes and the fitted results.

2. The code of (2), (3), (4), (5).

3. Zip them and submit to eeclass.