

Assignment 11: Shape Finder

CS 730/830, Fall 2015

Electronic submission due at **11:30pm on Wed, Dec 2**

Hardcopy submission due at **start of class on Thu, Dec 3**

Overview

You will write a program to find line segments in simulated laser rangefinder data given as an occupancy grid. Graduate students will extend their program to also find circles. Your program will be given on standard input a bitmap image (in ASCII format). It will emit the set of lines (and for grad students, circles) that it found in the image. We recommend that you use the RANSAC algorithm. We provide an image generator so that you can tune the algorithm's parameters to achieve good performance. Some reasonable starting values might be:

minimum number of inliers: 5

minimum density of inliers: 0.7

maximum distance of inlier from model: 2

Background

You'll be dealing with numbers, so don't forget to watch for things like division by 0. You should usually be able to deal with these by choosing new random points.

1. You will find yourself wanting to fit a line to two or more points. Many line fitting methods assume that, for particular known values of x , you have measured some y values which might be corrupted by noise. Hence they find the line that minimizes the squared error in y at each x value. But in our situation here, both the x and y values for a particular obstacle could be subject to noise. We want to minimize the distance of each point from the fitted line, and this distance should be measured by the shortest distance to the line from the point (not just the distance in the y direction).

To fit a line in this way to two or more points, it's actually best to work in polar coordinates. To convert each data point (x_i, y_i) to polar form (r_i, θ_i) , we have $r_i = \sqrt{x_i^2 + y_i^2}$ and $\theta_i = \text{atan2}(y_i, x_i)$. The best fit line (r, θ) to N data points is then:

$$\begin{aligned} A &= \sum_i r_i^2 \sin(2\theta_i) - \frac{2}{N} \sum_i \sum_j r_i r_j \cos(\theta_i) \sin(\theta_j) \\ B &= \sum_i r_i^2 \cos(2\theta_i) - \frac{1}{N} \sum_i \sum_j r_i r_j \cos(\theta_i + \theta_j) \\ \theta &= \text{atan2}(A, B)/2 \\ r &= \frac{\sum_i r_i \cos(\theta_i - \theta)}{N} \end{aligned}$$

To convert this line to slope-intercept form, let $x = r \cos(\theta)$ and $y = r \sin(\theta)$. Then the slope is $-x/y$ and the y -intercept is $y + \frac{x^2}{y}$.

2. Given a line, you will want to compute the distance from a point (x, y) to the nearest point lying on a line $y = ax + b$. This is $|ax - y + b|/\sqrt{a^2 + 1}$.

3. Given a fitted line, you will need to determine the endpoints of the best line segment. We recommend taking the simple approach of testing each possible line segment that starts and ends at inlier data points and keeping the one that explains the most data (has the most inliers between the endpoints, inclusive) while still having density above the threshold. To compute density, just find the distance between the endpoints, don't worry about the exact distance along the fitted line.

4. To find the center (x, y) and radius r of a circle from three points (x_i, y_i) :

$$m_a = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\begin{aligned}
m_b &= \frac{y_3 - y_2}{x_3 - x_2} \\
x &= \frac{m_a m_b (y_1 - y_3) + m_b (x_1 + x_2) - m_a (x_2 + x_3)}{2(m_b - m_a)} \\
y &= -\frac{1}{m_a} \left(x - \frac{x_1 + x_2}{2} \right) + \frac{y_1 + y_2}{2} \\
r &= \sqrt{(x_1 - x)^2 + (y_1 - y)^2}
\end{aligned}$$

Don't worry about refitting the circle to all the inliers. Finding the distance of a point from the circle is easy: find its distance from the center and compare it to the radius.

Input/Output

The bitmap will be provided on standard input in a format similar to the vacuum world (assignments 1 and 2) and motion planning (assignment 3) maps. All cells will be either empty (`_`) or blocked (`#`). There may be substantial noise in the image data. Lines starting with `/` are comment lines and can be ignored.

Your program must emit all the circles it finds (zero, in the case of undergrads) and then all the line segments it finds. Circles are specified by center $\langle x, y \rangle$, and radius, lines by the $\langle x, y \rangle$ coordinates of the endpoints. As in:

```

number of circles: 1
34.1 56.7 5.0
number of lines: 2
34.1 45.5 162.6 73.2
10.8 23.3 72.5 116.1

```

Coordinates are in map units (as in assignment 3), where $\langle 0, 0 \rangle$ is the bottom left corner.

Supplied Utilities

We supply:

image-generator generates sample data. Command line arguments include **-height**, **-width**, **-numlines**, **-numcircles**, and **-noise** (a percentage given as a float). Try **-help** for details.

ransac-reference a sample solution.

ransac-tester a test harness to evaluate your program. It assumes that the image contains special comments listing the shapes used to generate it. It scores your output based on precision, recall, and accuracy. Try **-help** to learn more about its arguments.

Write-up

Submit a brief write-up with your hardcopy solution answering the following questions:

1. Describe any implementation choices you made that you felt were important. Mention anything else that we should know when evaluating your program.
2. What sorts of input will cause your program to fail? In other words, what does it have trouble with?
3. What suggestions do you have for improving this assignment in the future?