ELEC 474 – Machine Vision
Lab #3 – Circle RANSAC
Revised Sept. 2019

RANSAC Methods

1.0 :: Introduction

The premise for this lab is that you have been hired by the one and only, Ellen Dusk, to be her new Chief of Tracking. She needs some help with two of her companies, Tessler and SpaceY. At Tessler, the lane tracking algorithm on their self-driving vehicles has gone haywire and you have been tasked with developing a system to track lane markings on the road for their upcoming Tessler Model Q. Ellen has kindly provided you some sample images to ensure that your line tracking algorithm works as expected. SpaceY has had some issues with their new interplanetary Starship. The ship's guidance system is having difficulty recognizing planets as it's crossing the galaxy. Ellen has tasked you with developing a circle tracking algorithm that can identify these planets in outer space. If both of these new algorithms are successful, then you and Ellen will become lifelong friends!

Per the software specification, Tessler has requested that the developed software correctly identifies and illuminates both edges along the highway. The specification does not call for 100% correct identification of the lanes, but it should be clear where the vehicle should travel down the road.

SpaceY has provided you with an image taken during their last flight. They have requested that at a minimum, the sun in the center of the image be found. The specification does not call for all of the planets and stars in the image to be identified, simply identifying the most prominent planets will do just fine.

The purpose of this lab is to introduce the concepts of RANSAC line detection and RANSAC circle detection. To correctly complete both requirements, you will need to implement the required algorithms as well as use appropriate filtering techniques as discussed in class.

2.0 :: Pre-Lab [1 Mark]

***Required Files*:** lines.jpg

Your task for the pre-lab component is to implement (from scratch) a RANSAC line extraction routine. First, load a greyscale image (covert if necessary) and generate a binary edge image using Canny Edge Detection. Using the Canny edge map, implement your RANSAC line extraction algorithm to detect the significant lines in the image.

To achieve this, the Tessler has provided you with an image of a couple colored lines to test your algorithm (**lines.jpg**). After all, you have to prove the accuracy and reliability of your algorithm before Ellen will allow it to be tested on the Model Q. Tessler's standards and safety criteria is that you can successfully display these lines, either overlaid (in color) in the original or edge image, or in a separate image. Once Ellen and her directors (the TA's) have verified your initial work, you may proceed to testing the Model Q on the road and SpaceY's guidance system for the Flamingo VI spacecraft.

Tips:

1. For an OpenCV example of Canny Edge Detection using cv::Canny see the following link:
    a. https://docs.opencv.org/4.1.0/dd/d1a/group__imgproc__feature.html#ga04723e007ed888ddf11d9ba04e2232de
2. For help with implementing the RANSAC line extraction routine pseudocode can be found in the lectures:
    a. 06-ELEC 474 Geometric Primitive Extraction (slides 12-26)

3.0 :: Lab [6 Marks]

***Required Files*:** road_1.jpeg, road_2.jpeg, road_3.jpeg, road_4.jpeg, Planets.jpg
Optional Files: Circle Reference Files

- circle.jpg
- concentric_circles.jpg

Commented [IM1]: ???

The goal for the lab component is to implement your RANSAC line extraction algorithm on the road images such that the Model Q can successfully navigate down the road and to implement a RANSAC circle extraction algorithm on the Planets image such that the spacecraft navigation system can navigate the galaxy. This will be accomplished in four steps, starting with pre-processing of the required images to extract the required edge data. Your RANSAC line extraction algorithm will be run on Tessler road images to identify the lanes of the road. When successful, you will created a modified version of this algorithm to create a RANSAC circle extraction algorithm that allows you to identify planets in space.

Task #1 – Pre-processing on Road images:

In order to perform line extraction, the edges in the image must be identified. Same as in the prelab, you will need to perform Canny Edge Detection on the road images to extract all of the edges. You may want to filter the image before running edge detection to achieve better results. This can be achieved by eroding or degrading the image, using functions such as cv::gaussianBlur or cv::medianBlur if they help.

Task #2 – Line extraction on Road images:

With the edges now extracted from the road images, you can now perform the same RANSAC line extraction algorithm on this set of images. Once each line is identified you should then project the lines, either overlaid (in color) in the original or edge image, or in a separate image.

Tips:

1. Experiment with different stopping conditions, such as an iteration threshold or a consensus threshold.
2. Write your routine so that it returns the top-k results, sorted from best to worst.
3. You will need to write a subroutine that calculates the distance of a point to a line. The algebra to do this can be found below

$$distance(P_1, P_2, (x_0, y_0) = \frac{|(y_2 - y_1)x_0 - (x_2 - x_1)y_0 + x_2y_1 - y_2x_1|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}$$

   Where $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are two points on the line and x_0 and y_0 are the point in question
4. [Optional] Calculate the inliers in the edge map that contribute to each resulting line and highlight these inliers with a different color (e.g. red highlight)

Commented [IM2]: And, and/or, or?

Task #3 – Pre-processing on Planet images:

In order to perform circle extraction, the edges in the image must be identified. Same as in the prelab, you will need to perform Canny Edge Detection on the planet images to extract all of the edges. You may want to filter the image before running edge detection to achieve better results. This can be achieved by eroding or degrading the image, using functions such as cv::gaussianBlur or cv::medianBlur if they help.

Task #4 – Circle extraction on Planet images:

With the edges now extracted from the planet images, you can now implement your own RANSAC circle extraction algorithm. When developing the algorithm, it may be useful to use a simpler image such as the circle.jpg image. For the final result, the algorithm will need to be run on the Planets.jpg file. Once each circle is identified correctly, you should display the circles found, either overlaid (in color) in the original or edge image, or in a separate image.

Tips:

1. An easy way to generate random numbers is using the cv::RNG class, more information can be found here:
    a. https://docs.opencv.org/4.1.0/d1/dd6/classcv_1_1RNG.html
2. For help with implementing the RANSAC circle extraction routine, see the lectures slides:
    a. 06-ELEC 474 Geometric Primitive Extraction (slides 12-26)
3. Experiment with different stopping conditions, such as an iteration threshold or a consensus threshold.
4. Write your routine so that it returns the top-k results, sorted from best to worst.
5. You will need to write a subroutine that calculates the distance of a point to a circle. The algebra to do this can be found below

$$distance(C, r, (x_0, y_0) = \left|\sqrt{(x_0 - x_c)^2 + (y_0 - y_c)^2} - r\right|$$

Where $C = (x_c, y_c)$ which is the center of the circle, r is the radius of the circle, and x_0 and y_0 are the point in question.

6. [Optional] Calculate the inliers in the edge map that contribute to each resulting line and highlight these inliers with a different color (e.g. red highlight)

Task #5 – Improve RANSAC Algorithm:

Now that the basic routines are running correctly, implement some additional computational feature(s) to improve the efficiency your algorithms. Compare the number of iterations that are required for the original algorithm and the improved version. You must be able to demonstrate and explain how/why these improvements help.

4.0 :: Mark Distribution

Commented [IM3]: Correct slides?

Commented [IM4]: And, and/or, or?

	Detailed Distribution			
Pre-Lab	Complete [1 Mark]	Incomplete [0 Marks]		**/1**
Task 1	Display filtered and clear edges in all road images [1 Mark]	Partially display edges in some road images [0.5 Marks]	Don't display edges in road images [0 Marks]	**/1**
Task 2	Clearly mark both sides of road in all road images [1 Mark]	Clearly mark one side of road in some road images [0.5 Marks]	Don't mark sides of road in road images [0 Marks]	**/1**
Task 3	Display filtered and clear edges on all planets in the image [1 Mark]	Partially display edges on some planets in the image [0.5 Marks]	Don't display edges on planets in the image [0 Marks]	**/1**
Task 4	Clearly mark the predominant planet in the image [2 Marks]	Partially mark predominant planet in the image [1 Mark]	Don't mark any planets in the image [0 Marks]	**/2**
Task 5	Demonstrate, Explain and prove that the RANSAC algorithm has been improved [1 Mark]	Demonstrate, or explain, or prove that the RANSAC algorithm has been partially improved [0.5 Marks]	Don't improve the RANSAC algorithm [0 Marks]	**/1**
		Total:		**/7**