

TRC3500 Assignment 1 Report

Tan Jin Chun (32194471), Cheryl Ooi Sze Wei (32611196)

General Algorithm

The general algorithm involves generating two shapes (ovals or rectangles) side by side, one oriented correctly and the other with a slight deviation. An image of these shapes is taken using a phone camera or a webcam, after printing the image out in MATLAB. This image is then processed through our code to determine the orientation of each shape. The process includes calculating central moments for each shape to understand its area, centroid, and orientation. The orientation, converted from radians to degrees if necessary, helps in plotting a line representing the shape's orientation centered at its centroid. By comparing the calculated orientations with the original, we identify the correctly oriented shape based on the smallest orientation difference, while the largest difference points to the deviated one. Finally, we fit a sigmoidal curve to the calculated values based on the different test cases to interpolate the curve's value, facilitating a concise comparison and identification of the shapes' orientations.

1) How to differentiate the original blob and the orientated blob?

We will differentiate the original blob and the orientated blob by using the original orientation used to generate the original blob and subtract it with the calculated orientation from the pictures (with the code that has been written for the DEMO). The largest calculated orientation difference will be the orientated blob, the lowest calculated orientation difference will be the original blob.

2) Outline something you could change to improve the performance of your system and why it will help.

We could optimize image acquisition by using proper lighting and camera settings to capture a high-contrast image to minimize post-processing. This would ensure that the image resolution is not higher than necessary for analysis to reduce processing time. Built-in functions like 'rgb2gray' and 'graythresh' could also be used instead of the pre-defined

function for image conversion. Connectivity Analysis could also be carried out using the bwlabel or bwconncomp function instead of manual implementation of those functions, improving the run-time of the code. The built-in function such as 'regionprops' could also be used to calculate the blob properties which will help to detect the blobs much more accurately. In a nutshell, these changes will help to reduce computational time, improve the accuracy of the analysis and create a much more efficient system.

3) Results

3.1 The orientation of the shapes generated

The two figures below show the orientation of the shapes. The one on the left (50 degrees) is the accurate orientation, the other angles (on the right) have slight deviations of 0.5 degrees and 0.8 degrees.

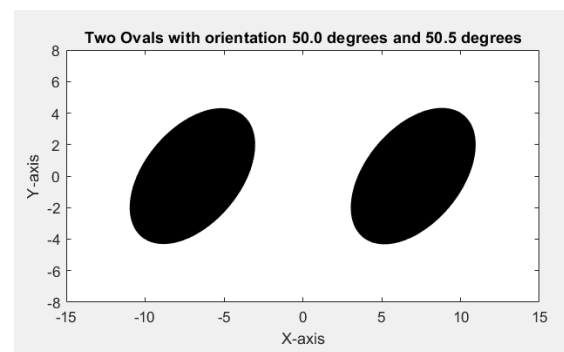


Figure 3.1.1

Figure 3.1.1 shows the orientation of the two ovals. The left side shows 50 and the right side shows 50.5 degrees.

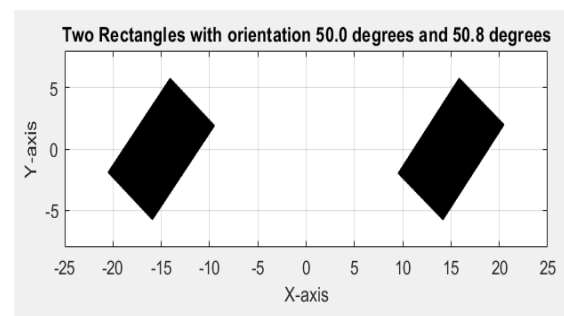


Figure 3.1.2

Figure 3.1.2 shows the orientation of the two rectangles. The left side shows 50 degrees and the right side shows 50.8 degrees.

3.2 Sigmoid curve

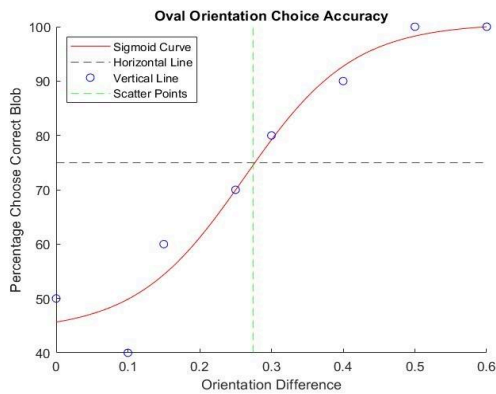


Figure 3.3.1

Figure 3.2.1 shows the sigmoid curve for the choice accuracy for the oval shape.

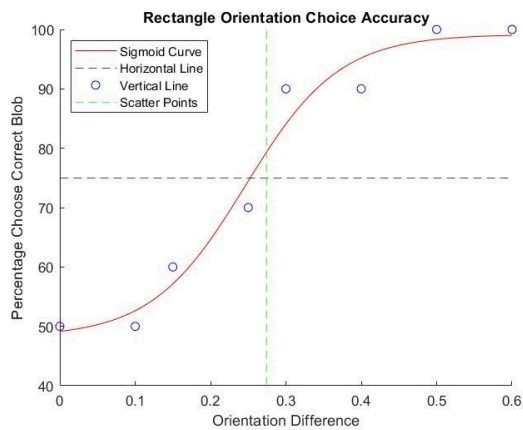


Figure 3.3.2

Figure 3.2.2 shows the sigmoid curve for the choice accuracy for the rectangle shape.

The table below shows the values obtained from the experiments.

Test Case 1 (Oval)

Orientation Difference	Percentage Choose Correct Blob
0.0	50
0.1	40
0.15	60
0.25	70
0.3	80
0.4	90
0.5	100
0.6	100

Table 3.1

Test Case 2 (Rectangle)

Orientation Difference	Percentage Choose Correct Blob
0.0	50
0.1	50
0.15	60
0.25	70
0.3	90
0.4	90
0.5	100
0.6	100

Table 3.2