

Assignment 1

Problem Statement:

“Write a program that reads in an image, makes a determination about the location of the eyes, and output the positions of the eyes.”

Software Used:

MATLAB R2018b

Solution:

This solution assumes that the strongest circular shapes in images would be the eyes and thus by finding the circles, we can deduce the position of the eyes. First the image is acquired from the main file. The stored image is a color image and will have three values for each pixel corresponding to the intensity values of the Red, Green and Blue channels. This image must be converted to a Grayscale image. For this we use the function **rgb2gray()**.

If the image size is large, we must resize it. The **imresize()** function is used to do this. Additionally, this is done iteratively to ensure that the image size is smaller than a threshold value (set here at 2000x2000 pixels)

Next, Gaussian filtering is done on the image to eliminate noise. This is done using the **imgaussfilt()** function. After filtering, we have to detect the edges in the image. For this implementation, I have used the Canny method of edge detection. This method finds edges by finding the local maxima of the gradient of the image. It makes use of an additional Gaussian filter and two threshold values so as not to be fooled by noise. The function **edge()** is used to perform edge detection. The threshold values for the edges was found using trial and error method.

Edge detection returns a binary image with the edges represented using white pixels while the rest of the image is black. Once this is done, we use a circular Hough transform to find any circular shapes in the image. In most of the given images, the eyes were the strongest circular shapes, thus finding them would consequently lead to finding the eyes themselves. I used **imfindcircles()** to do this. This function has a parameter that specifies the sensitivity of the algorithm while finding circles in the image. If the value is large, then the number of circles will be larger. However, it also increases the possibility of finding false positives. The sensitivity is adjusted till the number of circles found is at least two. We can also specify the range of radii between which the circles must lie. Through **imfindcircles()**, the centers and radii of the circles are found.

The coordinates of the centers of the circles are considered to be the positions of the eyes. The left eye will have a smaller x value than the right eye.

Algorithm:

- 1) Store the image in a variable 'img'. Image will be in RGB format.
- 2) Convert RGB image to Grayscale.
- 3) Find the size of the image.
- 4) If the resolution of the image is too large, resize it.
- 5) Perform Gaussian filtering to eliminate noise.
- 6) Perform edge detection using the Canny Algorithm. Edge detection returns a binary image.
- 7) Find the circles in the binary image using circular Hough transform.
- 8) Adjust the sensitivity of the Hough transform till the number of circles found in the image is greater than or equal to 2.
- 9) Assign the x and y coordinates of the circles to the points representing the eyes. The right eye has a larger x value than the left eye.
- 10) Adjust the x and y coordinates according to scale with which the image was resized.
- 11) Return the coordinates of the eyes.

Results:

The main file computes the normalized distance between the eye positions find by the program and the actual positions of the eyes. The results for the given data set are:

0.0207

0.0093

0.4780

0.2479

0.0122

0.0328

0.2150

0.0246

0.2140

0.0291