Computer Vision Assignment 2 - Canny Edge Detection and Other Analysis

Love Mehta
2014csb1018@iitrpr.ac.in

Department of Computer Science and Engineering,
Indian Institute of Technology - Ropar

In this assignment we have three different parts to be implemented, as discussed below.

1 Canny Edge Detection

MyCannyEdgeDetector is the function designed for the Canny edge detection process. It takes as input an image matrix (not the address) and a threshold value (b/w 0 to 1) and returns the edge detected image. It also displays the image in a seperate window. We follow the certain steps of canny edge detection as given below.The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Categorize the continuous gradient directions into a small set of discrete directions 0, 45, 90, 135
4. Apply non-maximum suppression to get rid of spurious response to edge detection
5. Apply double threshold to determine potential edges
6. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

2 Comparing the Outputs

MyCompareOutput is the function designed to compare the output of my implementation and matlab inbuilt function "edge". It takes as input an image matrix (not the address) and a threshold value (b/w 0 to 1) and displays the eucledian and psnr distance maps. The function compares the output of MyCannyEdgeDetector with the Matlab edge function. Using the same threshold value for the two functions. We use the inbuilt PSNR and Euclidean distance measures. The function output are - a colored difference map of the inbuilt and our Canny detector output. A blockwise summary matrix (divide image into 3 x 3 non-overlapping) of the distances between the Matlab inbuilt and MyCannyEdgeDetector outputs.

3 Corner Detection

MyDetectInterest is the function designed for the interest point detection process. It takes as input an image matrix (not the address) and a threshold value (b/w 0 to 1). We use the output of MyCannyEdgeDetector() and detect interest points in it by following the steps for Harris Interest Point Detection as given below:

1. Compute image gradient in x and y.
2. Find the intensity gradients of the image
3. Categorize the continuous gradient directions into a small set of discrete directions 0, 45, 90, 135
4. Apply non-maximum suppression to get rid of spurious response to edge detection
5. Apply double threshold to determine potential edges
6. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

In the end we display the corners detected on the original input image and save it too.

4 References

1. https://en.wikipedia.org/wiki/Canny_edge_detector
2. https://en.wikipedia.org/wiki/Sobel_operator