VISUAL PERCEPTION

Lab report 1: Corner Detection

Author: **RAABID HUSSAIN**

Summary:-

The coursework was related to an implementation of a common corner detection algorithm called 'Harris corner detector'. The algorithm works by computing the smoothed image derivatives (matrix M) to infer the degree of corner-ness at the image position/pixel. The algorithm proposes an approximation to this algorithm by simplifying the calculations with the help of determinant and trace. The coursework involved comparing the original method with the approximation proposed by Harris and Stephens using MATLAB.

The first task was to generate a matrix E that would constitute the smaller eigenvalues of M for each pixel and a similar matrix R but with the Harris detector method. In terms of visual comparison the image produced by R was blurred owing to the fact that it is an approximation to the actual matrix E. However, the approximation turned out to produce reasonable results later. Another point to note here was since Harris reduced the computational cost of the original algorithm, contrarily the MATLAB script was taking more time to compute the R matrix. This was solved by manually calculating the determinants and the traces rather than using the MATLAB in built functions (as function calls take more time and the MATLAB functions are taking care of all possible input parameters, thus increasing the complexity). This reduced to the time to calculate R by approximately 10 times.

The next task was to extract the corners from the matrices. This was achieved by reducing the solutions to the 81 (number of actual corners in the image) most salient points from the matrix. When the results were displayed on the screen with the original image in the background, it was found that the 8 most salient points did not represent the true corners in the image rather they were clustered together in different corners in the image (Note: the points given by E and R were not exactly same). To solve this, an 11x11 neighborhood was used to filter out the points around the salient points (already selected in previous iterations) selected so that the clusters would be replaced by just one point thus extracting the 81 local maxima instead of the 81 highest value pixels. After a few experiments on all the images provided it was necessary to add a clause when selecting the most salient points. This stated that if the new chosen salient point is connected to the 11x11 neighborhood of any of the previously selected salient points, it should be ignored. This solved the problem in all the images except 'chessboard03', which had white lines in between the boxes. Increasing the 11x11 neighborhood to 17x17 solved the problem. A point to notice here is that the algorithms do not give exact corner (when you zoom and see). This is probably due to the nature of the edges in the image.

The last task was to determine the corners with sub-pixel accuracy. This was done by taking the 3x3 neighborhood around the maximal suppressed salient points and approximating them to a parabola and finding the exact (actually approximated) maxima of the region. Initially the entire 11x11 neighborhood was being used, however the results returned with a change in pixel as the intensity distribution in the image was not in an ideal pattern. So, the points were reduced to a 3x3 neighborhood. Linear least square was used to determine the equation of the parabola and the solution was inverted (as there is no inbuilt function for finding maxima) and 'fminsearch' function in MATLAB was used to find the maxima location. However, after seeing the results (more than 0.5 difference) on all the images, it was decided to add a new condition while approximating the parabola: adding more weight to the central pixel so that the results were within 0.5 difference.