Homework 4 Writeup

Instructions

- Describe any interesting decisions you made to write your algorithm.
- Show and discuss the results of your algorithm.
- Feel free to include code snippets, images, and equations.

get_interest_point

I approximated the gradient of image by applying sobel filter. From the gradient, I could get the second momentum matrix M. I computed the determinant and trace of M. $C = det(M) - \alpha * trace(M)^2$, and if C is greater than threshold, which I decided as 0.006, the point represent corner.

get_descriptors

I first padded the image to avoid using zero pixel values for window when finding descriptors for pixel close to the end of image. Then for the given pixel, I retrieved surrounding window's derivatives, x and y direction each. From the derivatives, the histogram for edge direction can be found, to get the major direction shown in the window. Using the direction, Total direction of the window is normalized to get orientation invariance. Local histogram is formed using the normalized directions, and the feature of the point is formed using each of the local histograms. 16 locals are formed from the window, and histogram is formed by dividing angle into 8 partitions, so that the dimension of the feature is 16*8 = 128. The feature is normalized, and any value over 0.2 got fixed to 0.2 to avoid error from specular point. The feature get normalized again, and the stack of features for all the given points get returned.

match_features

For a given two features, the difference is computed. I used cosine difference, which is 1 - (cosine similarity) as the feature represent angular values. For a point on one image, I found 2 matching points on the other image, point with most closest feature and second closest feature. Then the difference between features is compared. If the ratio of difference, (distance to most closest point)/(distance to second closest point), is less than threshold, which I choose as 0.75, the most closest point is decided to match with the given point. The match point pairs and confidence, (distance to second closest point)(distance to most closest point) is returned.