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.
- Use as many pages as you need, but err on the short side If you feel you only need to write a short amount to meet the brief, th
- **Please make this document anonymous.**

get_interest_points.m

For the function get_interest_points, I used a normal Harris corner detection method. I followed the algorithms below to implement Harris corner detection:

1. Compute image derivatives by applying sobel filters to image.
2. Compute M components as squares of derivatives by finding I_x^2, I_y^2, and $I_x I_y$.
3. Apply a Gaussian filter of size $[4, 4]$ and width 2 to each of I_x^2, I_y^2, and $I_x I_y$.
4. Compute cornerness C $=$ det(M) $- \alpha$trace(M)^2.
5. Apply threshold on C to pick high cornerness.
6. Non-maxima suppression to pick peaks. I used function colfilt() to find maximum peaks.

get_descriptors.m

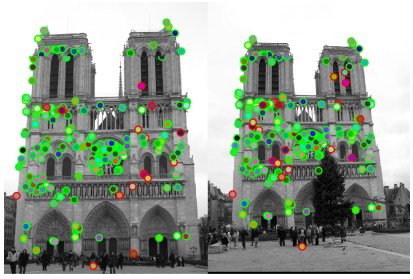
When using normalized patches, for Notre Dame, the accuracy on all points was 73.95$\%$ for 215 points, and 83$\%$ for top 100 points. For Mount Rushmore, the accruacy on all points was 84.5$\%$ for 129 submitted poitns, and 92$\%$ for top 100 points. For Gaudi's Episcopal Palace, the accuracy on all points was 12.5$\%$ for 8 points, and 1$\%$ for top 100 points.

This performance generally increased by implementing SIFT method. For Notre Dame, the accuracy on all points was 90.79$\%$ for 228 points, and 100$\%$ for top 100 points. For Mount Rushmore, the accruacy on all points was 98.5$\%$ for 133 submitted poitns, and

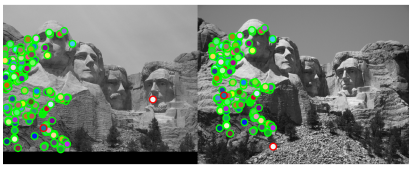
99% for top 100 points. For Gaudi's Episcopal Palace, the performance was worse. The accuracy on all points was 0% for 2 points, and 0% for top 100 points.

For SIFT, I implemented N x 128 features matrix, where N is the number of feature points. I divided 16 x 16 pixel window into 4 x 4 cells, with each cell consisting of 8-directional histogram. After finding features descriptors, I normalized each of the descriptor. Then, I capped all magnitudes higher than 0.2 to 0.2, after which I normalized the descriptor again.

Figures below show the results obtained for the three images - Notre Dame, Mount Rushmore, and Episcopal Gaudi:



(a) Notre Dame



(b) Mount Rushmore

(c) Episcopal Gaudi

match_features.m

I used NNDR test to implement feature matching. I calculated Euclidean distance between all possible combinations of feature points from the two images. Then, I sorted the matrix to find smallest distance and the next smallest distance. Taking quotient of those two numbers gave me NNDR, which I used to determine if that feature point is a valid match by comparing the ratio with the threshold.