

# Homework Writeup

## Instructions

- This write-up is intended to be ‘light’; its function is to help us grade your work and not to be an exhaustive report.
- Be brief and precise.
- Please describe any non-standard or interesting decisions you made in writing your algorithm.
- Show your results and discuss any interesting findings.
- List any extra credit implementation and its results.
- Feel free to include code snippets, images, and equations. Below are useful markup templates for these.
- **Please make this document anonymous.**

## Tasks

- **Task 1: get feature points**

In get feature points, I found the gradients in x and y directions of the images through a sobel filter. Then, took the squares of the derivatives and applied a Gaussian blur. From there I used those values to calculate the cornerness score for every pixel. Then I used a threshold value of (0.1 times the maximum cornerness score value) and (half the window width) to peak local max and eliminate clusters.

- **Task 2: plot feature points**

For plot feature points I used plt.scatter to show where interest points are.

- **Task 3: get feature descriptors**

For image patch, I flattened each window around the interest point to get a descriptor of dimension 256 by 1.

For SIFT, I took a window around the interest point and calculated the derivatives of the window, in order to map the gradient magnitude to the feature descriptor according to its gradient orientation. After calculating each feature descriptor, I normalized the values and clamped it to less than 0.2.

- **Task 4: match features**

I calculated the Euclidean distances between each descriptor by using numpy functions on matrices. Then sorting the distances to take the two closest points, nearest neighbor 1 (nn1) and nearest neighbor 2 (nn2). Then I used a threshold value of 0.89 to evaluate the nearest neighbor distance ratio. Values less than the threshold were valid matches.

## Result

1. First three images are of mt\_rushmore. The match features has highest accuracy: 65%
2. Second three images are of notre\_dame. The match features has the second highest accuracy: 65%
3. Last three images are of e\_gaudi. The feature matching didn't really work for this one, the accuracy was 0%

