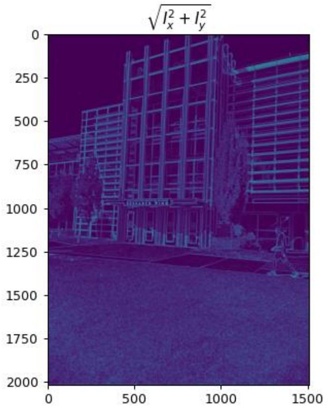
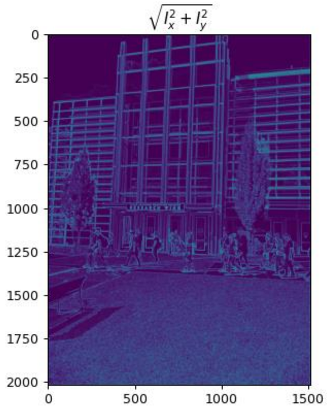
CS x476 Project 4

Bipin Koirala
bkoirala3@gatech.edu
bkoirala3
GTID: 903715285

Part 1: Harris corner detector

[insert visualization of \sqrt(I_x^2 + I_y^2) for Klaus image pair from proj2.ipynb here]





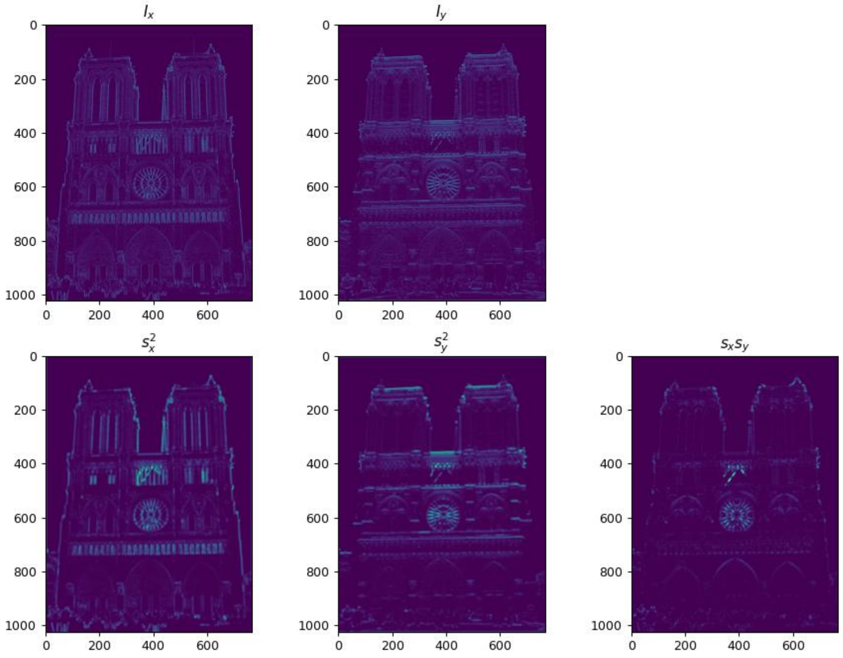
[Which areas have highest magnitude? Why?]

The building (frameworks) and the area where people are walking shows high magnitude in image gradient because these are the areas that depict many changes in gradient of image along vertical and horizontal direction.

In other words, edges have high magnitude for the same reason.

Part 1: Harris corner detector

[insert visualization of I_x, I_y, s_x^2, s_y^2, s_xs_y for Notre Dame image pair from proj2.ipynb here]



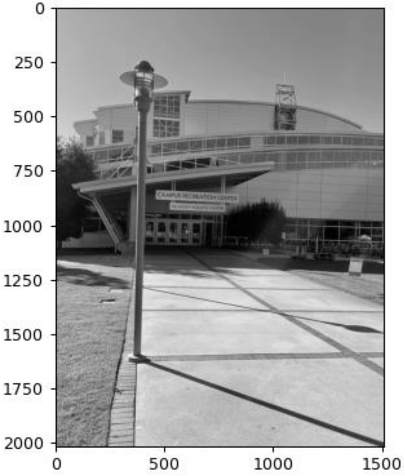
Part 1: Harris corner detector

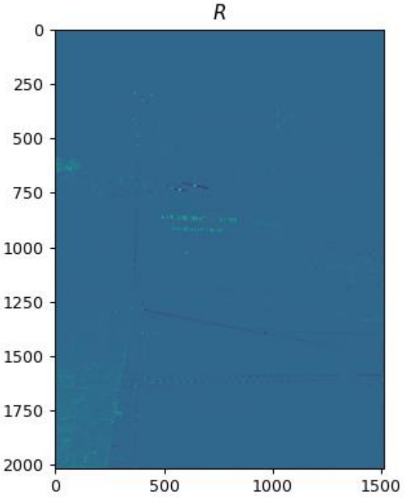
[Describe your implementation of generating the Harris cornerness score response map]

At first image gradient along x, y and xy direction were calculated and then convolved with the gaussian kernel to get the weighted moment matrix. Then ‘R’ score was obtained using the formula. $R = \det M - k\left(\operatorname{trace} M\right)^2$ The ‘R’ value depended only on the eigenvalues of ‘M’ and was large for a corner and small for flat region. Here; ‘k’ is empirically determined constant. In the project; k = 0.05 was taken.

Part 1: Harris corner detector

[insert visualization of corner response map of CRC image from proj2.ipynb here]



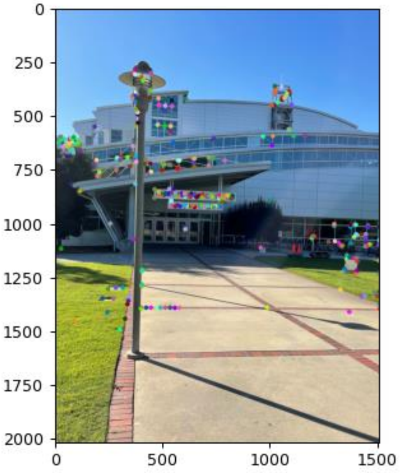


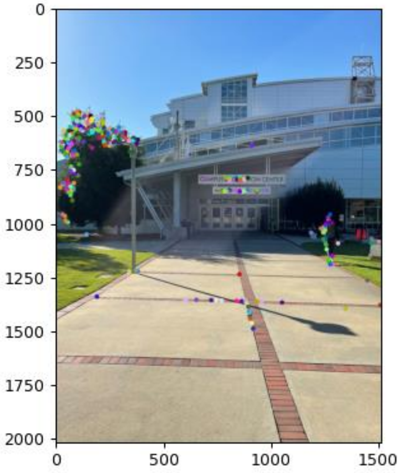
[Are gradient features invariant to both additive shifts (brightness) and multiplicative gain (contrast)? Why or why not?]

Gradient features are invariant to additive shifts. However, they are not invirant to multiplicative gain. If we see the histogram plot for the RGB channels, the plots do not change under additive shifts but in the case of multiplicative gian, the image becomes sharper.

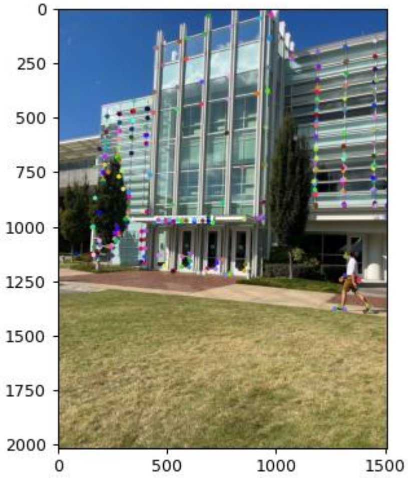
Part 1: Harris corner detector

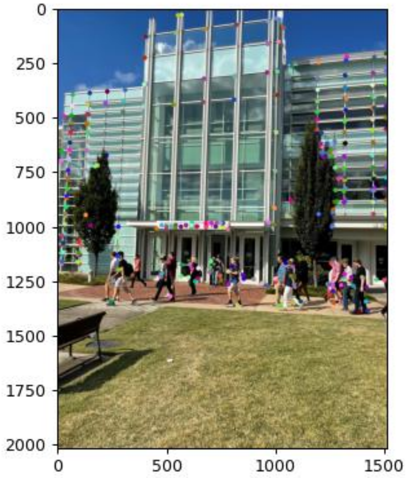
[insert visualization of CRC interest points from proj2.ipynb here]





[insert visualization of Klaus interest points from proj2.ipynb here]





Part 1: Harris corner detector

[What are the advantages and disadvantages of using maxpooling for non-maximum suppression (NMS)?]

Using Maxpool in non-maximum supression (NMS) results in the improvement of computation speed by performing a spatial compression to obtain respectable number of interest points. However, doing so results in performance decrease because Maxpooling yields a local maxima for a given kernel and might not always yield an interest point with a higher value that does not fall inside the window size.

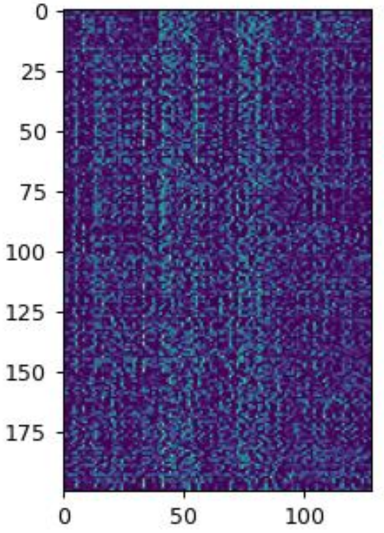
Part 1: Harris corner detector

[What is your intuition behind what makes the Harris corner detector effective?]

In an image, corners are invariant to rotation and translation and can be accounted as a good feature candidate for dectection taks. Harris corner detector first uses gradients to calculate second moment matrix and gets the ‘R’ score based on sliding window approach. The ‘R’ values classifies exactly where the corner based on the eigenvalues, which are invariant under transformation, of the second moment matrix. ‘R’ values are large when there is a window containing a corner and translation of this window in any orientation causes large change in gradients.

Part 2: SIFT feature descriptor

[insert visualization of SIFT feature descriptor from proj2.ipynb here]



[Describe your implementation of SIFT feature descriptors here]

First image gradients are obtained along x and y direction. Then the magnitude and orientation of each corresponding gradient are at each pixel location is obtained using squaring of elements of gradient matrices.

Then feature vectors are computed based on magnitude, orientation, feature width and shape of window for operation for each interest points. SIFT descriptor is obtained using a 16x16 window with its center at each interest point to obtain 128-D vector of gradient histogram. This is done by splitting the window to 4x4 grid with 8 bins. Thus, obtained vector is normalized and raised to the power of 1/2 , and this gives me SIFT descriptor.

Part 2: SIFT feature descriptor

[Why aren't our version of SIFT features rotation- or scale-invariant? What would you have to do to make them so?]

Our version of SIFT features do not exhibit rotational and scale invariance because we have implemented the key points as a normalized orientation distribution histogram. In order to establish rotational invariance, key locations are chosen at maxima and minima of Gaussian function in scale space followed by determining the descriptor’s orientation and assigning the angle to the respective key point. For scale invariance, we compute the Gaussian and utilize it to compute scale invariance Laplacian of Gaussian estimation using the resulting scale space [1].

[1] https://www.cs.ubc.ca/~lowe/keypoints/

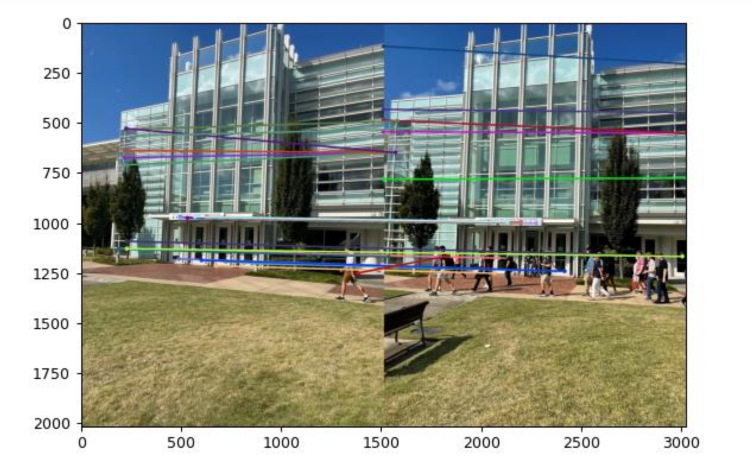
Part 2: SIFT feature descriptor

[Why are SIFT features better descriptors than simplying normalizing image intensities in a local window?]

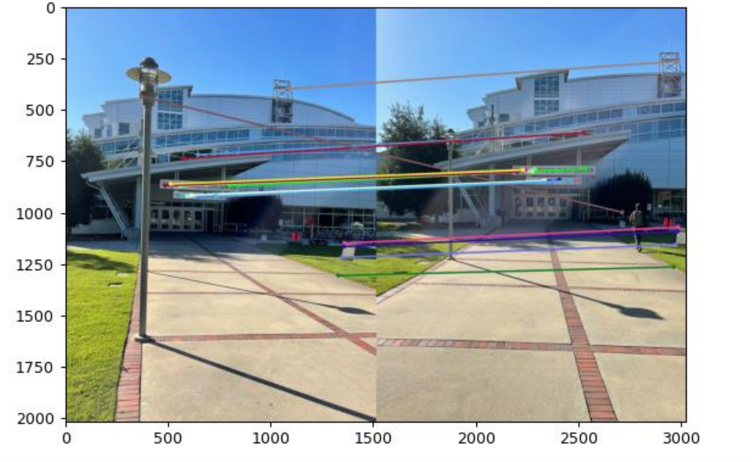
SIFT features are better descriptors because they convert an image into feature vectors that contain an array of local features. Each such feature vectors are invariant to rotation, scaling, translation and less invariant to light intensities and projections.

Part 3: Feature matching

[insert visualization of matches for Klaus image pair here with num_pts_to_visualize = 30]



[insert visualization of matches for CRC image pair here with num_pts_to_visualize = 30]



Part 3: Feature matching

[Describe your implementation of feature matching here]

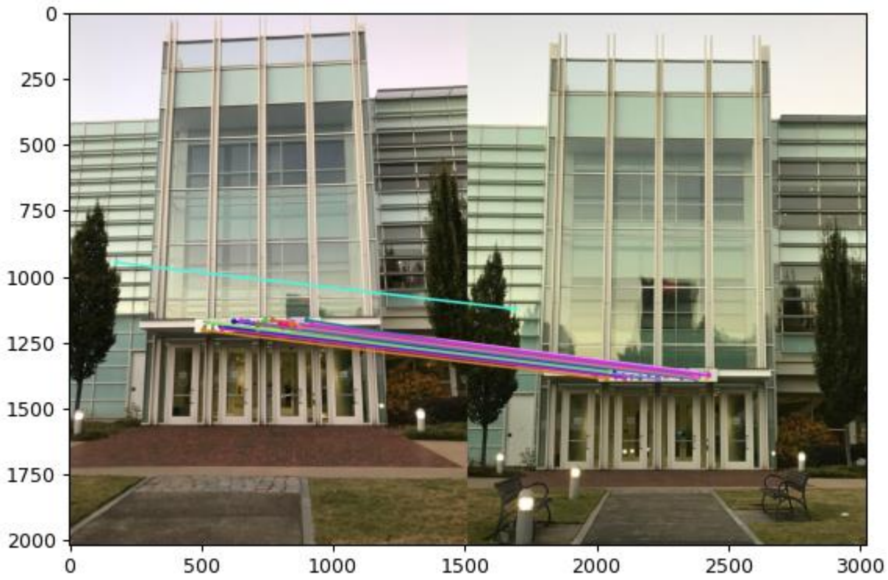
First, I computed the 2D-distance (Euclidean) between every feature in one array to another feature. This was followed by ratio test which computes the nearest neighbor distance ratio for feature matching. In ratio test, I arranged every distance in an incremental order and ratio computed. The threshold for test was set to 0.81 and the pairs that passed the threshold were grouped in an array and passed as output.

[Look at some of the mismatched features in your picture. Why might this have occurred?]

There are some mismatch in the feature map. I believe, this issue is caused due to pooling layer used in the computation of feature map. Furthermore, our version of SIFT feature is not rotation/scale invariant.

Part 4: SIFT feature descriptor (Extra Credit)

[insert visualization of matches for your own image pair here]



What makes our feature matching pipeline work well or poorly for your image pair?