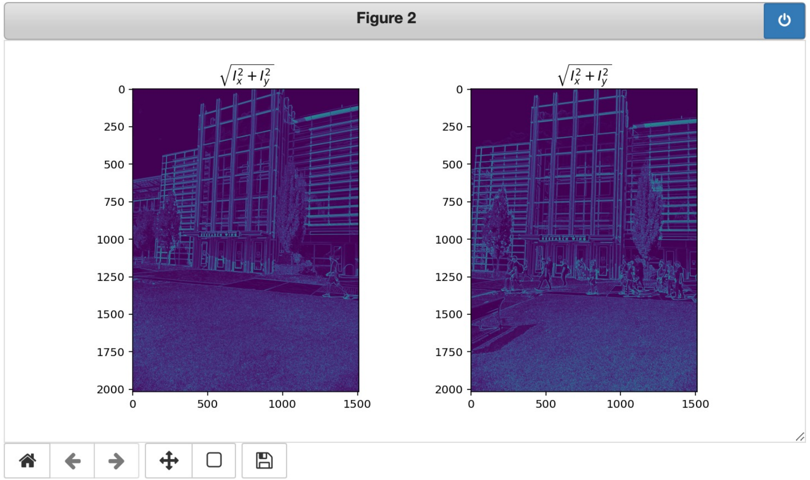
CS x476 Project 4

Aaron Lopes
alopes7@gatech.edu
alopes7
903407727

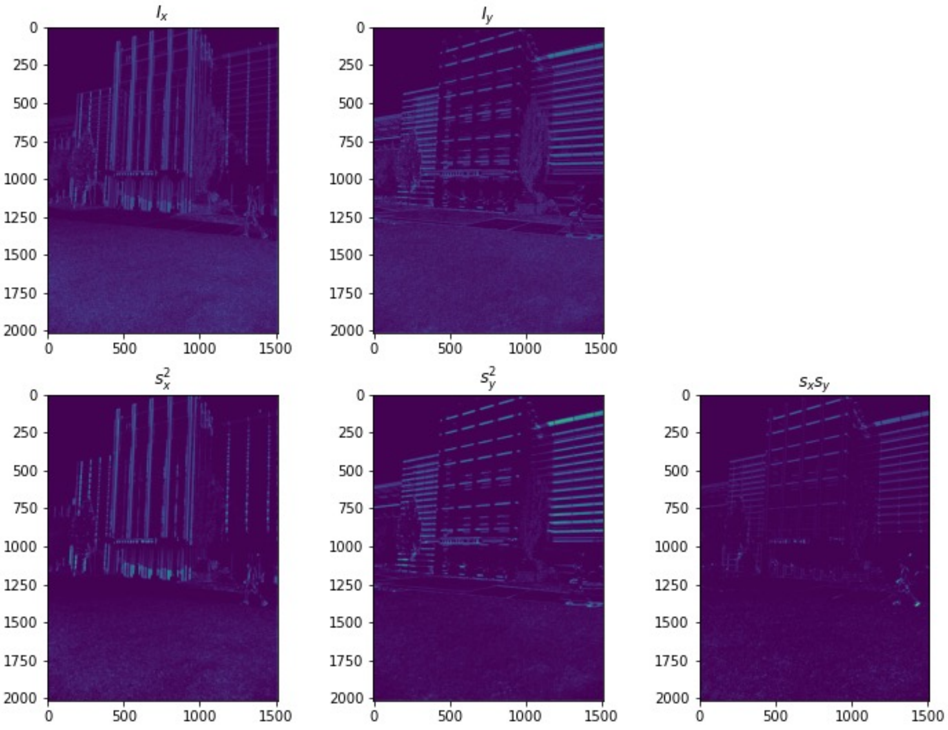
Part 1: Harris corner detector

```
compute_image_gradients():  "Correct"
```



The area’s with the highest magnitude are the vertical and horizontal edges, because these are the locations of maximum contrast between the surrounding areas with respect to the image orientation; where the gradient is pointing towards the most rapid increase in intensity

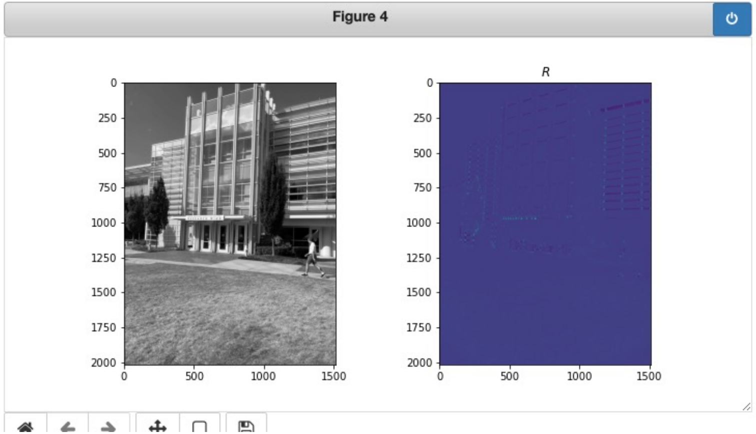
Part 1: Harris corner detector



Part 1: Harris corner detector

First, the black and white image is passed through the Sobel filter and the x and y gradients are obtained. From these gradients, the values of the second moment matrix are calculated by convoluting the squares and mixed gradients with a generated Gaussian filter. Once these values are found, the score can be calculated using the det(M) - alpha(trace(M))^2 where M is the second moment matrix.

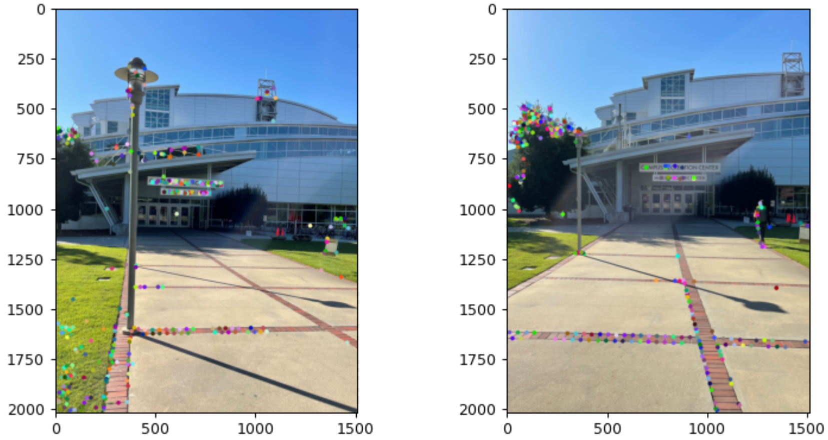
Part 1: Harris corner detector



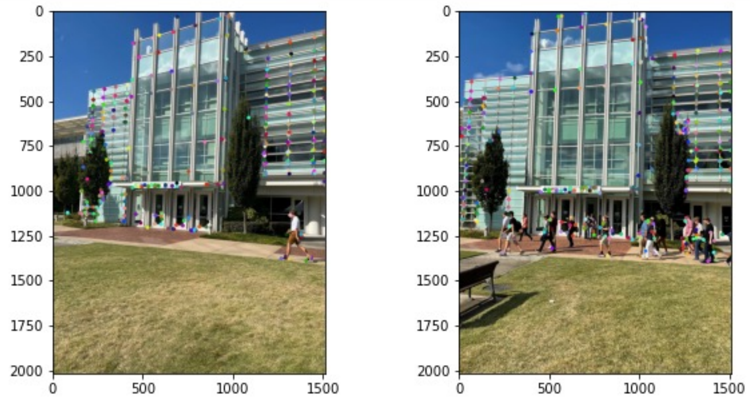
Yes, because the image is black and white changes in brightness and contrast with not effect the intensity gradients.

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)?]

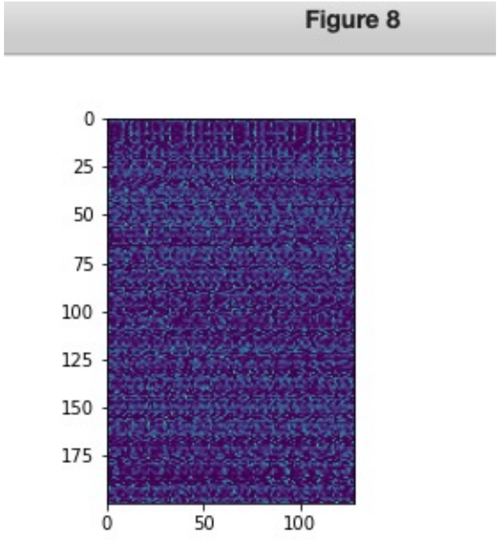
Advantage: setting a threshold for the image allows for uniformity in evaluation
Disadvantage: Not adaptable to images with varying contrast ratios

Part 1: Harris corner detector

As the detector is simply looking for significant changes in the x and y directions, the computational power and complexity needed to perform this task is relatively low.

Part 2: SIFT feature descriptor

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



[Describe your implementation of SIFT feature descriptors here]

First compute the image gradients of the given black and white image, then obtain the magnitudes and orientations of the gradient vectors. Once this information is computed, feature vectors of size 16x16 are obtained from the image. These are stored in a (1, 128) tensor.

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?]

They are not rotation of scale invariant because the SIFT features take orientation and magnitude into account while feature mapping. To make it invariant, a maximal orientation would need to be applied to the image.

Part 2: SIFT feature descriptor

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

By adding two more dimensions (magnitude and orientation) to the equation, the descriptors are more pertinent to image content rather than simply light distribution.

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, the distances between features is computed to obtain matches and confidence. Then, matches below the threshold are added to a list of matches, and returned.

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

There are repetitive features in the pictures.

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?