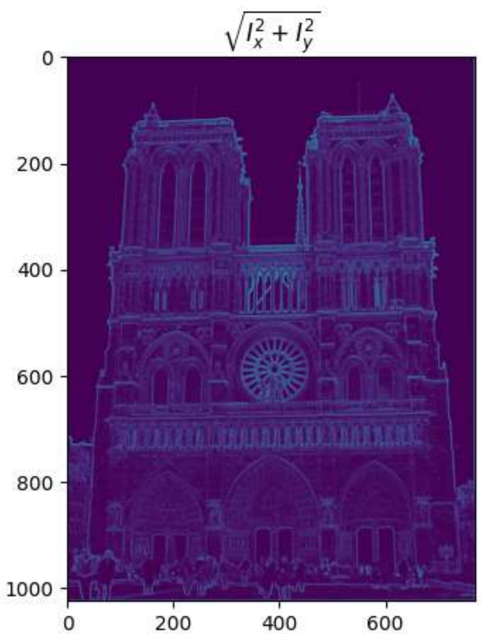
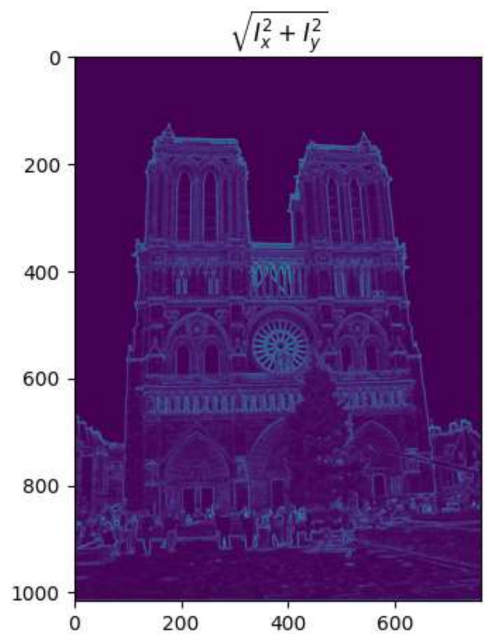
CS 5330 Programming Assignment 2

Dipak Sairamesh
sairamesh.d@northeastern.edu
sairamesh.d
001778611

Part 1: Harris corner detector

[insert visualization of \sqrt(I_x^2 + I_y^2) for Notre Dame image pair from pa2.ipynb here]



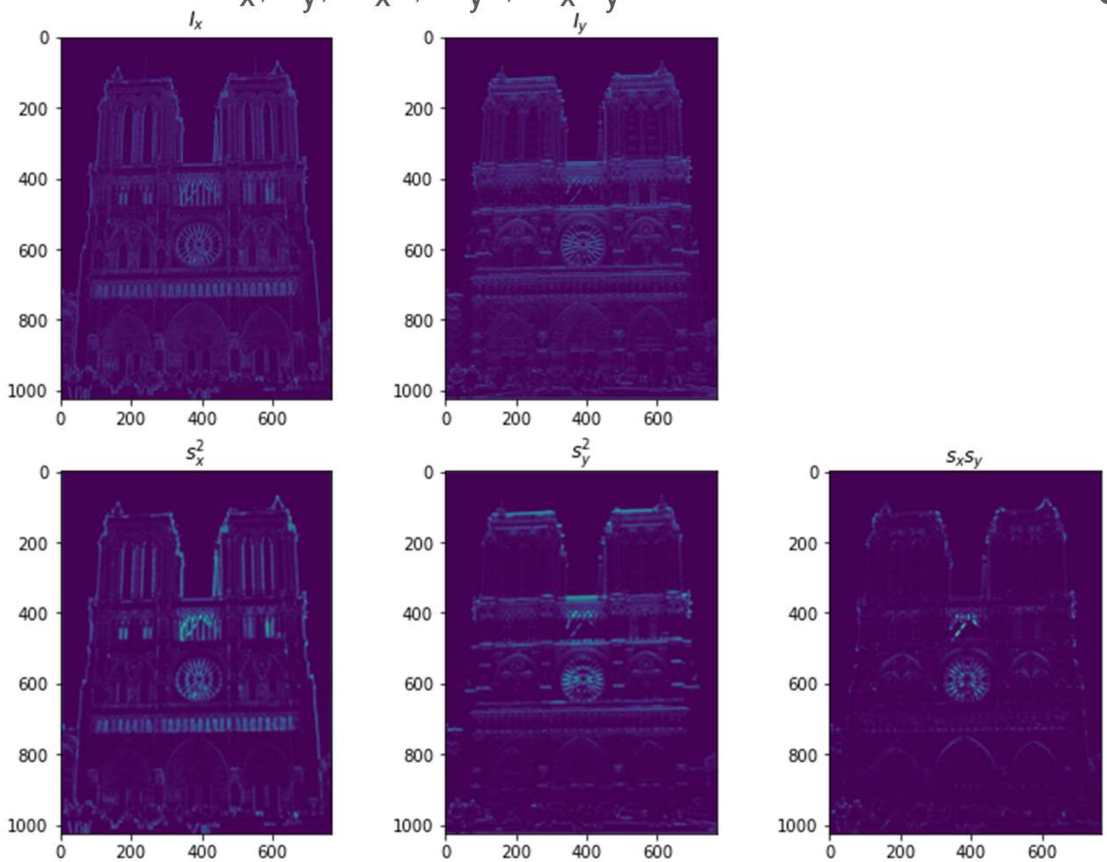


[Which areas have highest magnitude? Why?]

- Edges of objects in images have the highest magnitude. Sobel filter is used to calculate the image intensity at each pixel.
- It is also used to find the direction and rate of change of the steepest increase from light to dark regions.
- We also get to know how likely the edge will be oriented. Convolution of pixels with this filter helps with edge detection.

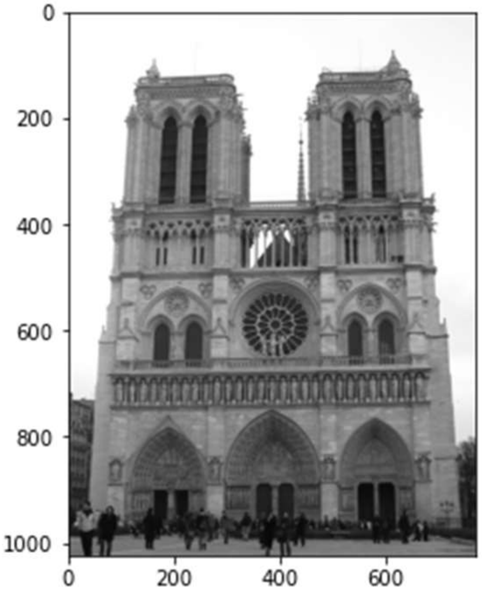
Part 1: Harris corner detector

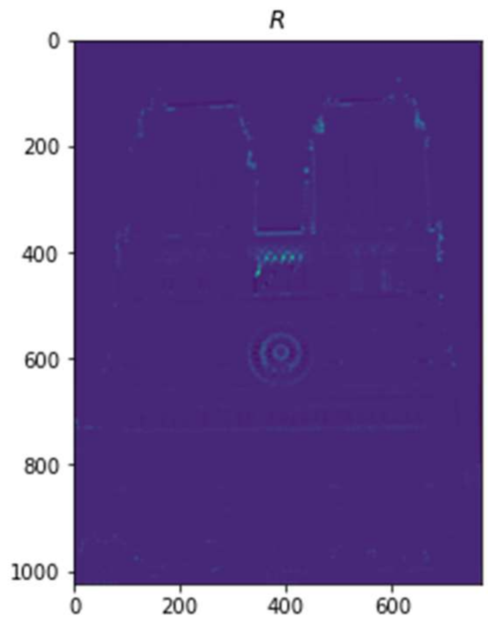
[insert visualization of I_x, I_y, s_x^2, s_y^2, s_xs_y for Notre Dame image pair from pa2.ipynb here]



Part 1: Harris corner detector

[insert visualization of corner response map of Notre Dame image from pa2.ipynb here]





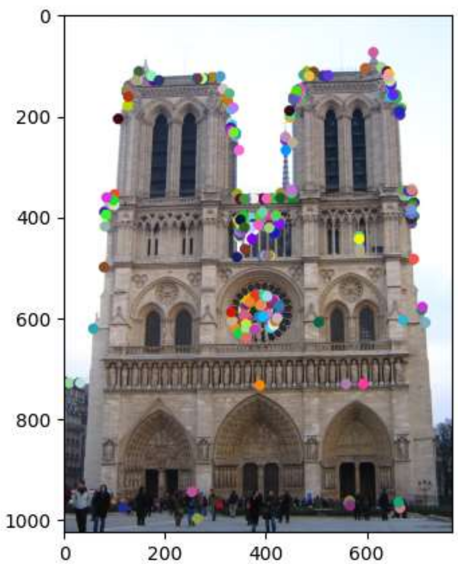
[Are gradient features invariant to both additive shifts (brightness) and multiplicative gain (contrast)? Why or why not? See Szeliski Figure 3.2]

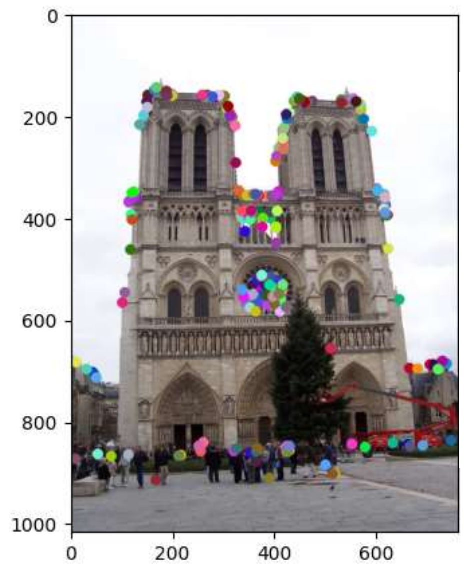
- Gradient features are not invariant to contrast and this is why we must perform some form of processing such as SIFT. Sift features are robust to changes in illumination or noise.
- A brightness change where a constant is added to each image pixel will not affect the gradient values but a change in intensity would make the feature variant.

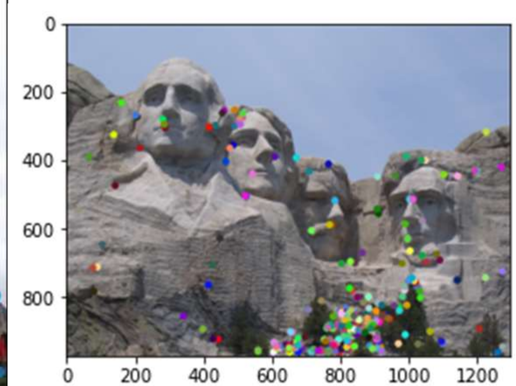
Part 1: Harris corner detector

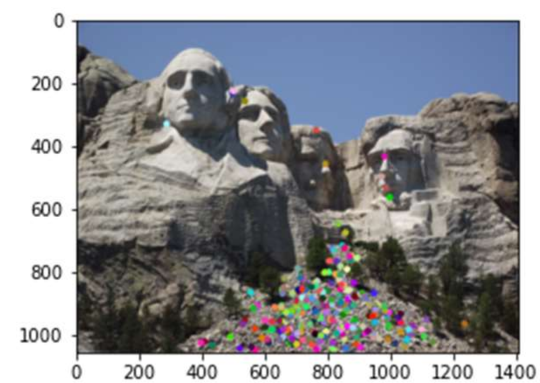
[insert visualization of Notre Dame interest points from pa2.ipynb here]

[insert visualization of Mt. Rushmore interest points from pa2.ipynb here]



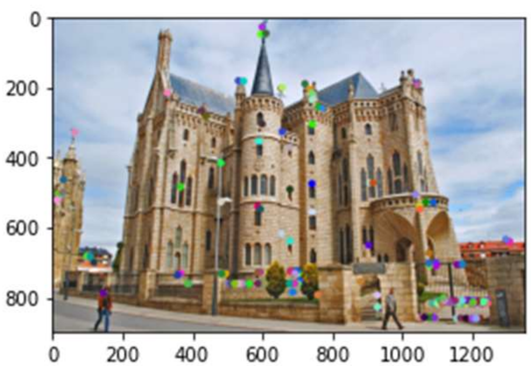


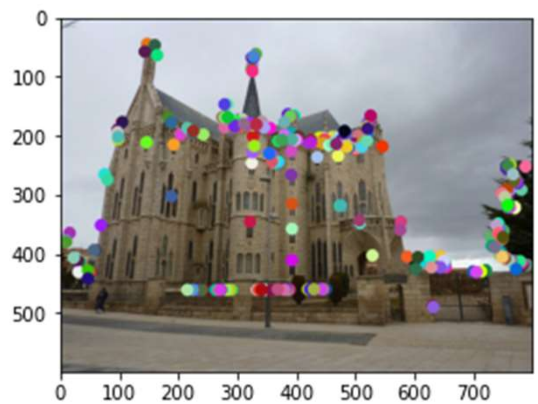




Part 1: Harris corner detector

[insert visualization of Gaudi interest points from pa2.ipynb here]





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

- Max pooling is used to save computational time by controlling the number of parameters that needs to be focused on. By filtering out these values and parameters, the most accurate boundary box is selected.
- Max pooling only considers the local maxima even if there are a lot of high values centered around it. And as such, there is also a loss of information for such values.

Part 1: Harris corner detector

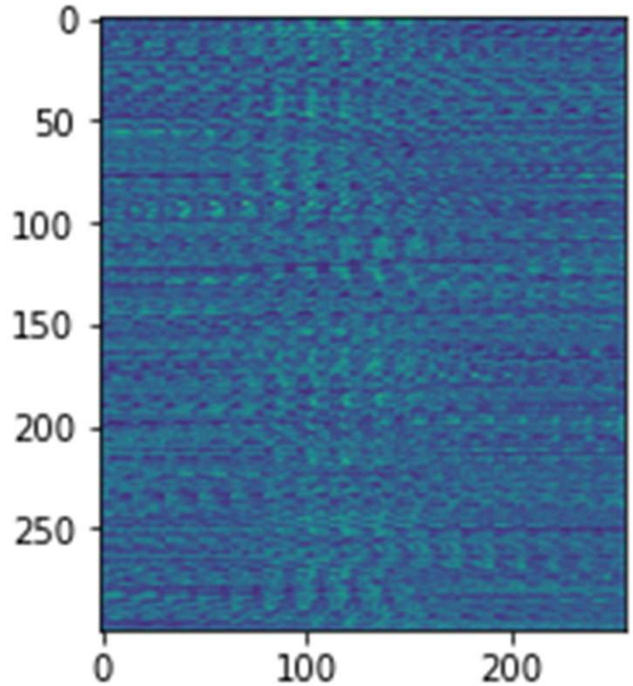
[What is your intuition behind what makes the Harris corner detector effective?]

- Harris corner detector is used to detect corners that are invariant to both translation and rotation while being differentiable and itegrateable. Edges don't offer the same level of confidence. We find derivatives of the image and then later remove edge features through non maxima suppression.

- We can more effectively and easily perform this by sliding a small window over the entire image to infer specific details about particular coordinates.

Part 2: Normalized patch feature descriptor

[insert visualization of normalized patch descriptor from pa2.ipynb here]

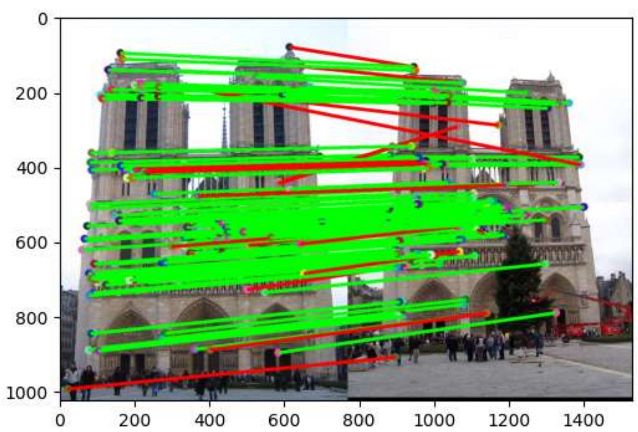


[Why aren't normalized patches a very good descriptor?]

- Normalized patches changes the range of the pixel intensity values in the entire image. They are normalized to be insensitive to illumination changes.
- Normalized patch descriptors don't do too well with changes in scale and is not invariant with respect to rotation as well.

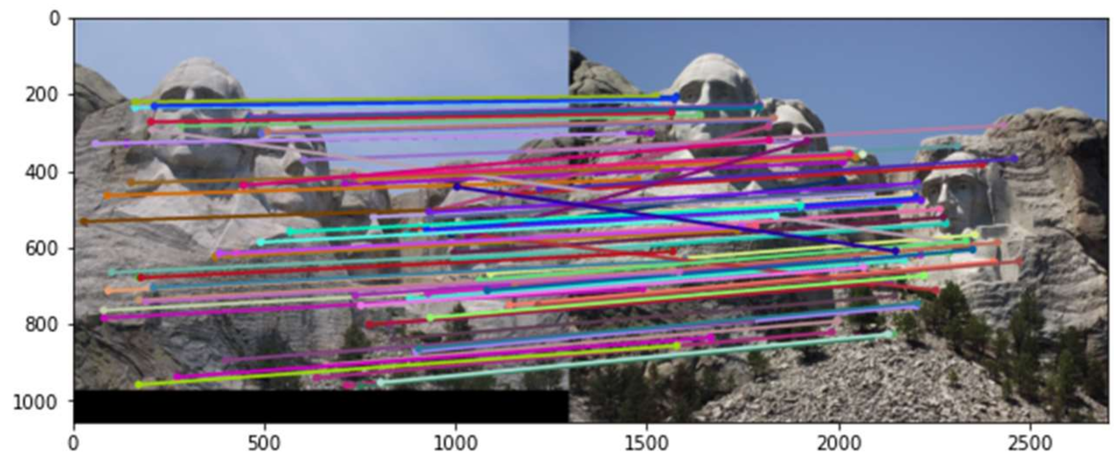
Part 3: Feature matching

[insert visualization of matches (with green/red lines for correct/incorrect correspondences) for Notre Dame image pair from pa2.ipynb here]



matches (out of 100): 160
Accuracy: 0.875

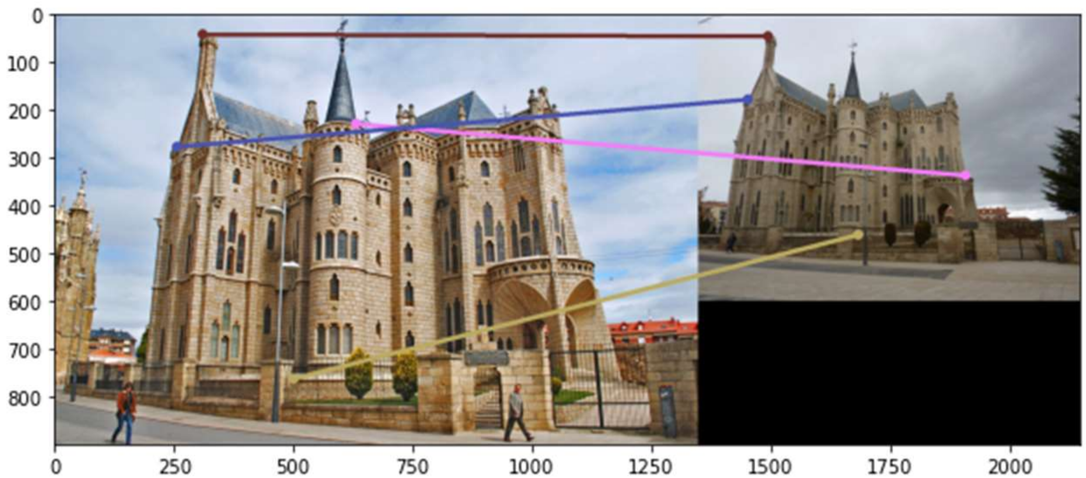
[insert visualization of matches for Mt. Rushmore image pair from pa2.ipynb here]



matches: 88
Accuracy: 0.76

Part 3: Feature matching

[insert visualization of matches for Gaudi image pair from pa2.ipynb here]



matches: 4

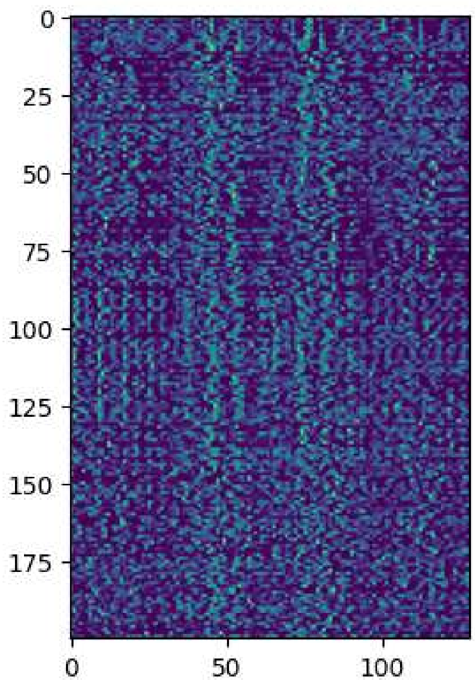
Accuracy: 0.0

[Describe your implementation of feature matching here]

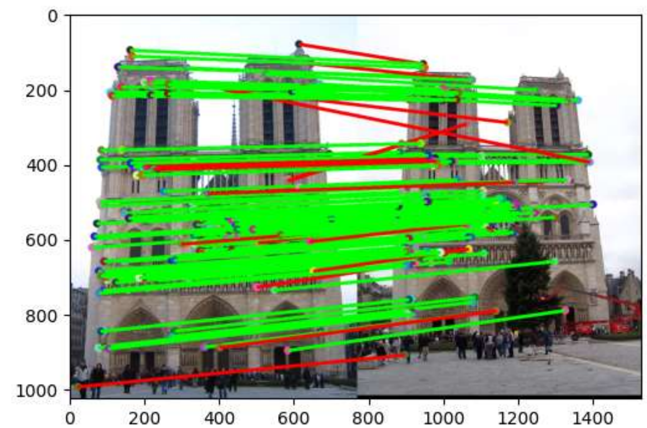
- To get the pairwise feature distances, iterate through list of features in features1.
- Compute the feature distances to every feature in features2 by using np.linalg.norm.
- Sort the array of distances in descending order, now the first column will be the feature distance to nearest neighbour and second column will be the distance to 2^{nd} nearest neighbour. Ratio of distances < threshold.
- 1/ratio is our confidence values and the indices of these confidence values provides matches.

Part 4: SIFT feature descriptor

[insert visualization of SIFT feature descriptor from pa2.ipynb here]



[insert visualization of matches (with green/red lines for correct/incorrect correspondences) for Notre Dame image pair from pa2.ipynb here]

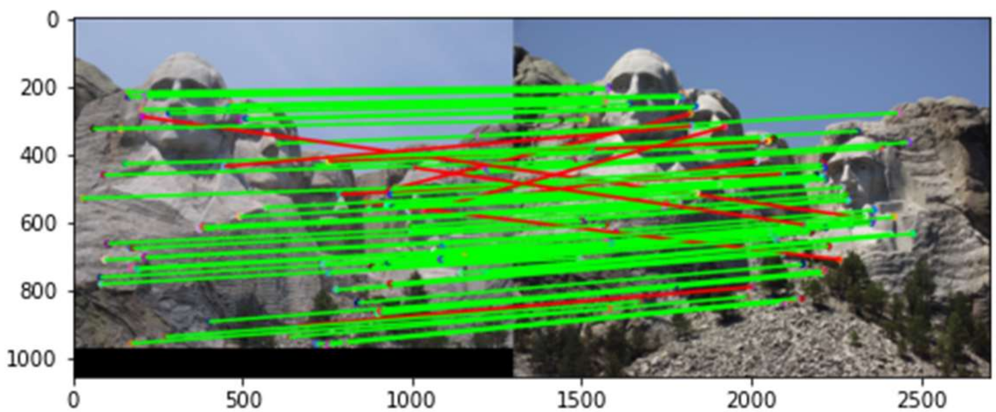


matches (out of 100): 160

Accuracy: 0.875

Part 4: SIFT feature descriptor

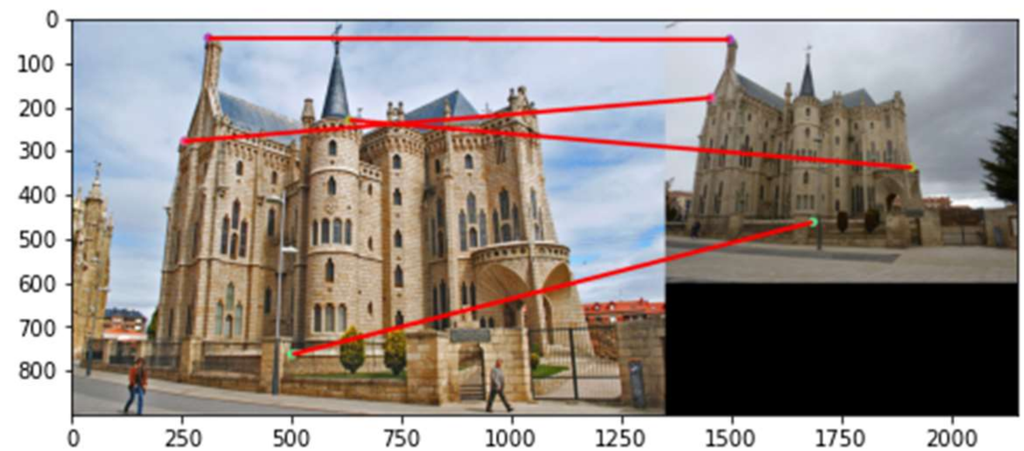
[insert visualization of matches for Mt. Rushmore image pair from pa2.ipynb here]



matches: 88

Accuracy: 0.76

[insert visualization of matches for Gaudi image pair from pa2.ipynb here]



matches: 4

Accuracy: 0.0

Part 4: SIFT feature descriptor

[Describe your implementation of SIFT feature descriptors here]

- Compute magnitude and gradients at each pixel location. Get a local patch of grids from the image and compute a histogram of weighted gradients, weighting each of the patches to produce a weighted histogram gradient vector.
- Next get the grids centred around keypoints to perform square root SIFT. Get the patches corresponding to our window coordinates of magnitudes and directions to compute the hog, which is normalized and then taken square root of.

[Why are SIFT features better descriptors than the normalized patches?]

- SIFT features are invariant to rotation, translation and scaling transformations in the image domain.
- It generates a large number of densely covered features over the full range of scales and locations in images.
- It computes features from orientations and histograms which makes it better as feature descriptors.

Conclusion

[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 is effective for edge detection through the use of Gaussian derivatives but to get better accuracy with rotation or scale invariance, a Laplacian derivative or higher order differential could be used. We find the most dominant locations of bin values in the patch histograms. Using a Difference of Gaussian or Laplacian filter will be more effective.

We also do not compare overlapping grids in our feature descriptors that could have similar features but this data might be lost.