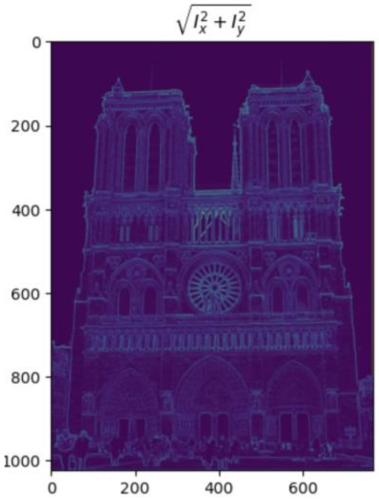
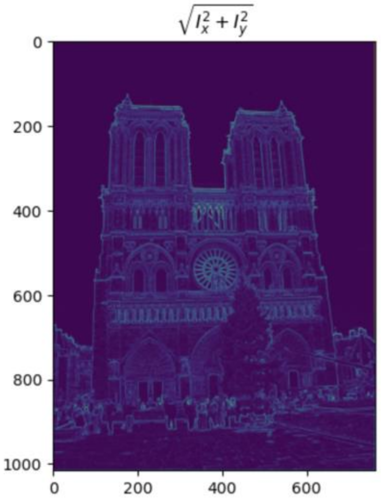
CS 4476 Project 2

Arjun Janakiraman
ajanakiraman7@gatech.edu
ajanakiraman7
903856569

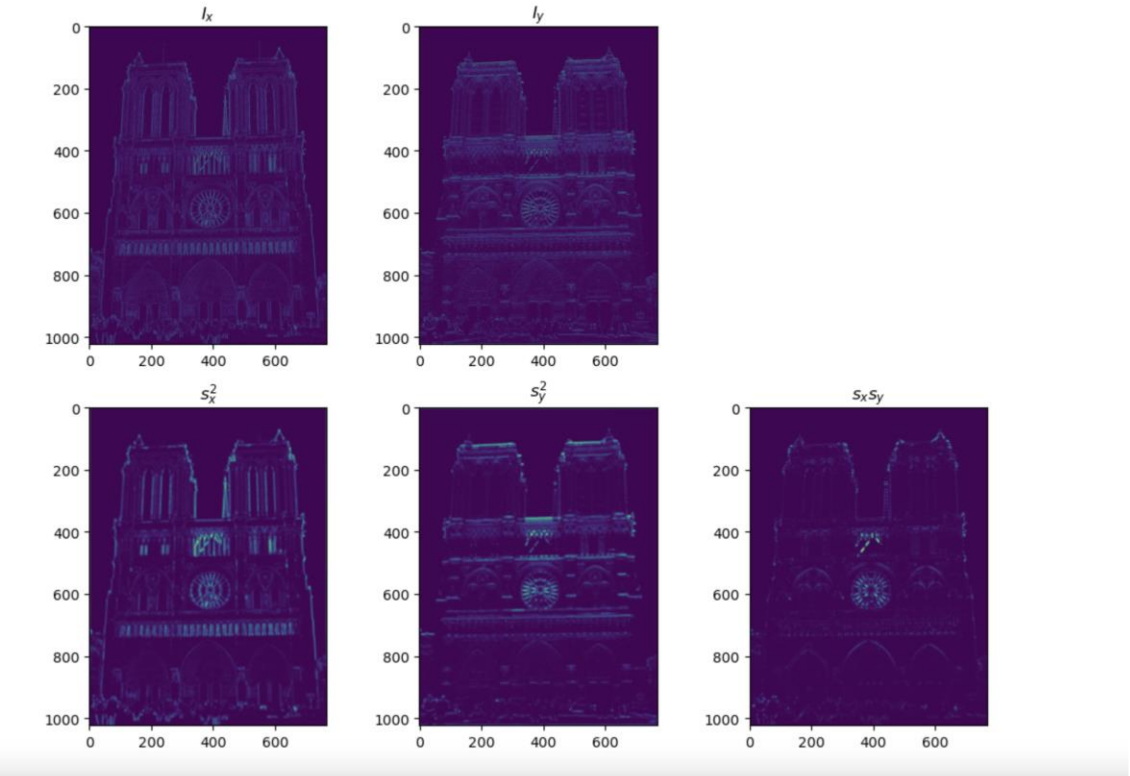
Part 1: Harris corner detector



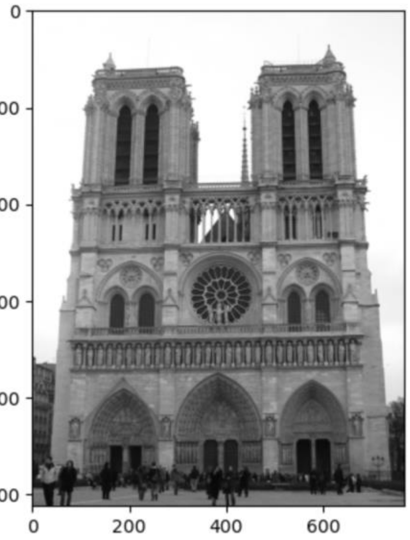


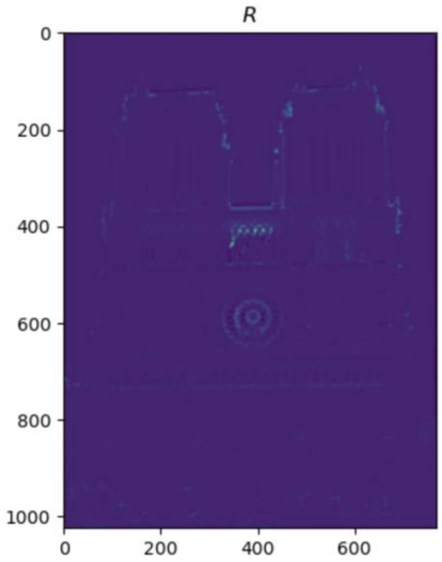
The areas with the highest gradient magnitude are the corner areas. This is because the formula for gradient of a vector involves the change in the X direction AND the change in the Y direction squared then sum together. At corner points BOTH the change and in X direction and change in Y direction will be high, by the definition of a corner, as a result the gradient will be high. You can contrast this with a line. In a line that is aligned with the X or Y axis, only ONE of the change in X direction and change in Y direction will be large, thus leading to (generally speaking) a lower magnitude than corners.

Part 1: Harris corner detector



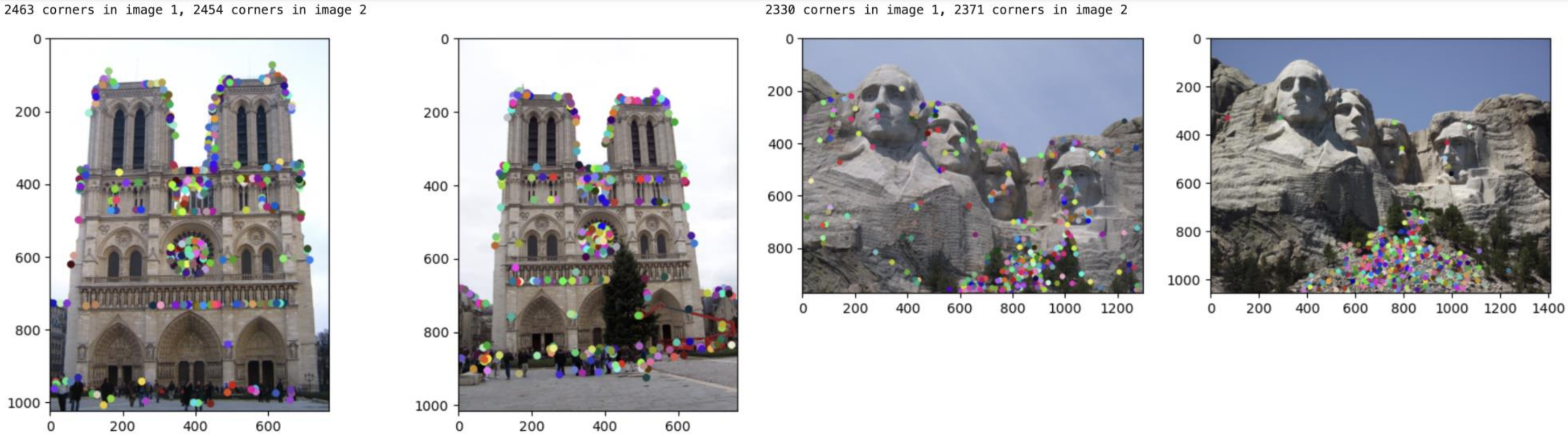
Part 1: Harris corner detector



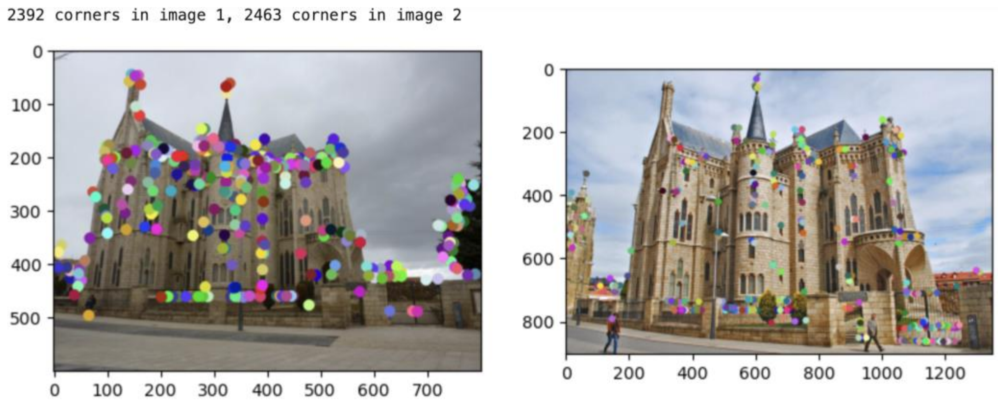


The gradient features ARE invariant to additive shifts (brightness) because they depend on the first derivative (change in values of neighboring pixels). Increasing or decreasing the brightness shifts every pixel value the same amount, so the change between them remains the same. However, gradients are NOT invariant to multiplicative gain, as when you multiply everything by some constant, the difference between pixel values either increases (if |constant| >1) or decreases (if |constant| < 1). This change in gradient magnitudes affects the second moment matrix, which depends on squared gradient values and influences the cornerness score in detectors in Harris Corner Detection

Part 1: Harris corner detector



Part 1: Harris corner detector



Advantages of using Max-pooling for non maximum suppression

Reduces redundant corner points

- For a given corner in a photo, there will be several pixels that exhibit ”cornerness”. With Max Pooling NMS, we only considered the local maxima
- Suppresses weaker responses caused by noise by only considering local maxima

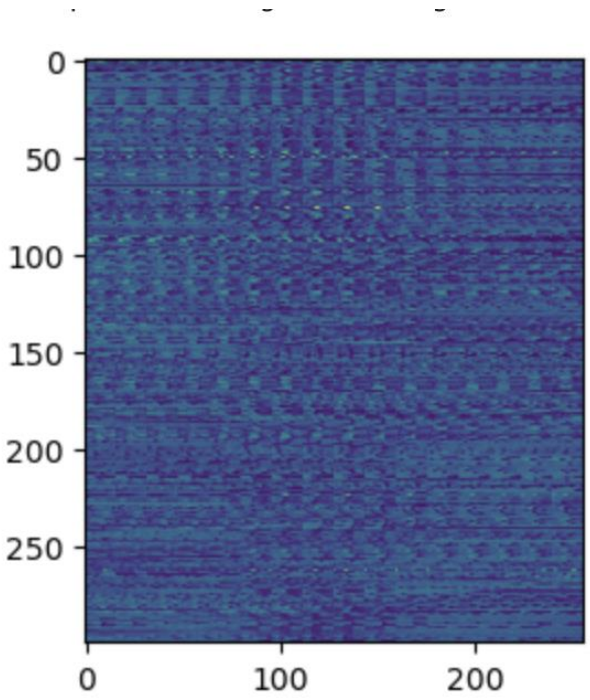
Disadvantages

- Too large of a window may cause 2 distinct corner points to be considered by just one local maxima
- Too small of a window, could cause the same corner to leak into multiple regions, causing redundancies

Part 1: Harris corner detector

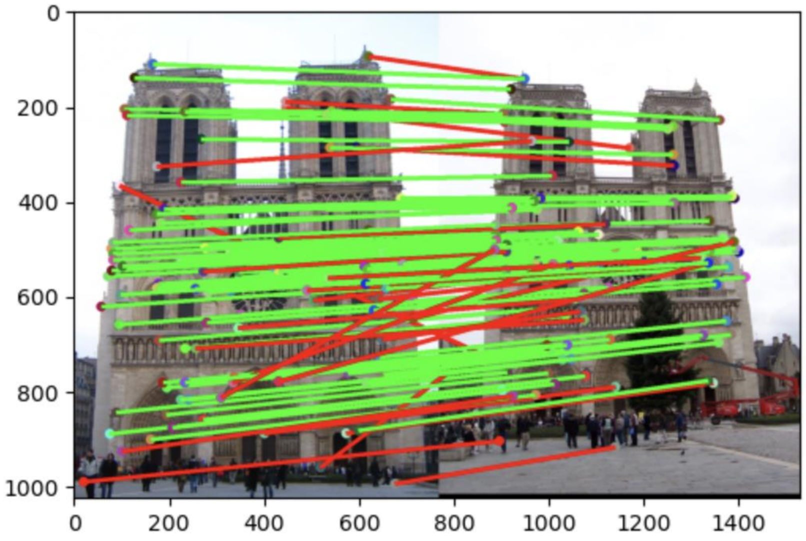
The Harris corner detector is effective due to its ability to concisely determine the qualities of a corner. As mentioned in previous slide, it leverages the fact if a window is around a corner, shifting a window in any direction will yield a large change in intensity. In encapsulates this idea in the second moment matrix which leverages both the x and y gradient. By checking if the eigen values are sufficiently large (or equivalently/more efficiently checking det(M) – a Trace(M)^2), we can accurately extract corner points.

Part 2: Normalized patch feature descriptor

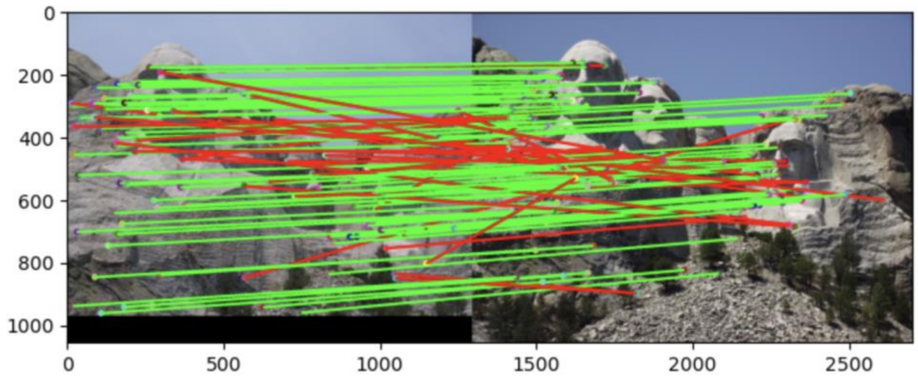


Normalized patch feature detectors don’t handle small shifts or rotations very well. These work by flattening a region into a vector and normalizing them. For example, if two corresponding regions in different images are identical but are slightly shifted (e.g., one region is moved down by a single row), their pixel values will no longer directly align in the descriptor. Even though the regions are essentially visually identical, their descriptors will appear drastically different, leading to a misleading similarity score.

Part 3: Feature matching

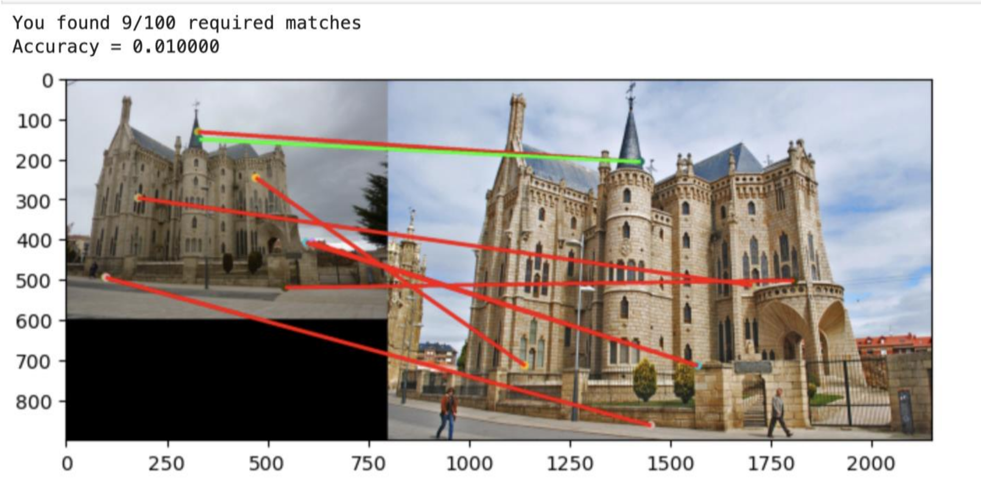


matches (out of 100): 107
Accuracy: 0.766355



matches: 107
Accuracy: 0.728972

Part 3: Feature matching



matches: 9

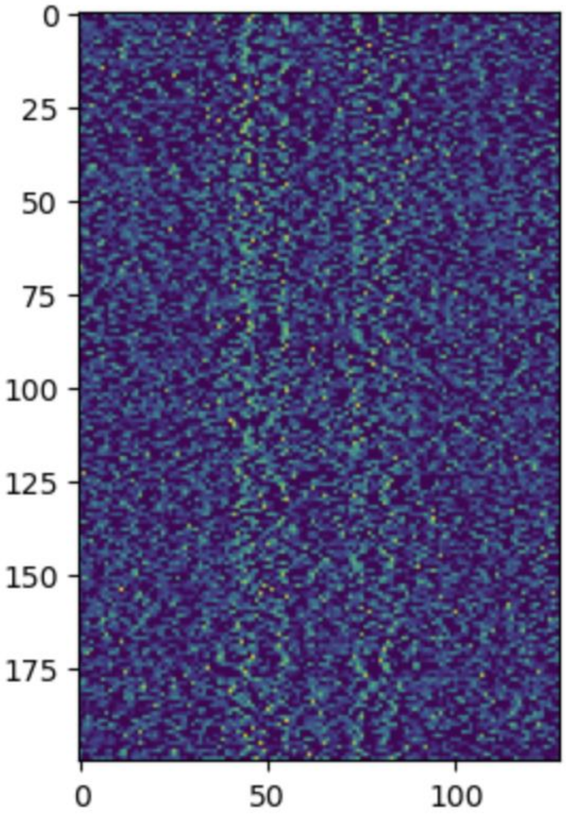
Accuracy: 0.01

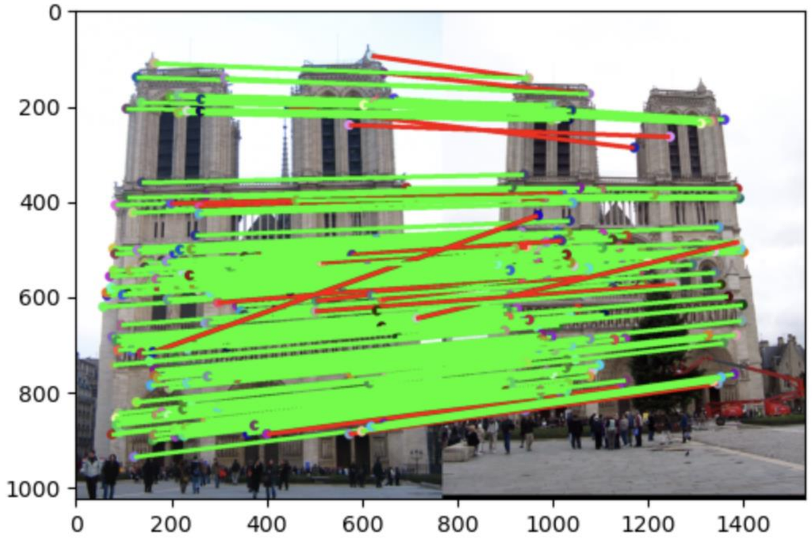
Args: features1 (MxD) and feature2 (Nx) which are both lists of feature vectors for regions of interests in two images.

Create 2d np array dists, where dists[i,j] = Euclidean distance between the ith feature in feature1 and the jth feature in feature2

Leveraging dists array, for each feature in feature1, I found the two features the least distance away from it in feature2. If the ratio between the smallest distance and the second smallest distance was less than a threshold (.8), I considered it a match between the current feature in feature1 and its closest feature in feature2. (leveraged argsort for dists to determine 2 closest features cleanly)

Part 4: SIFT feature descriptor

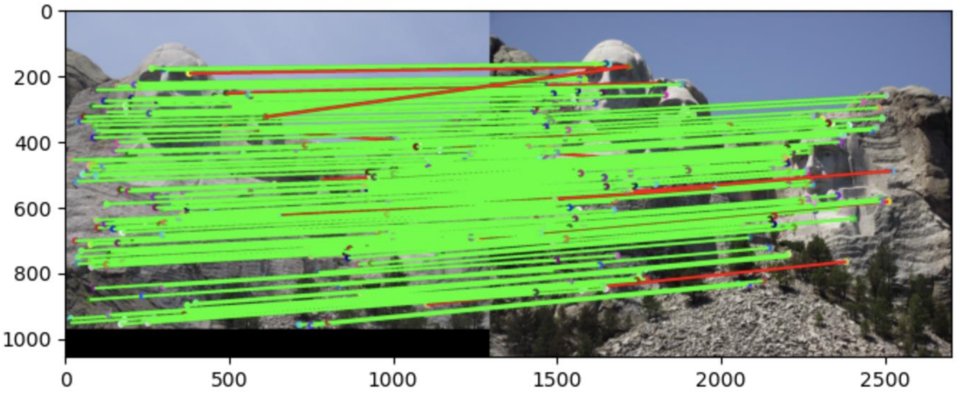


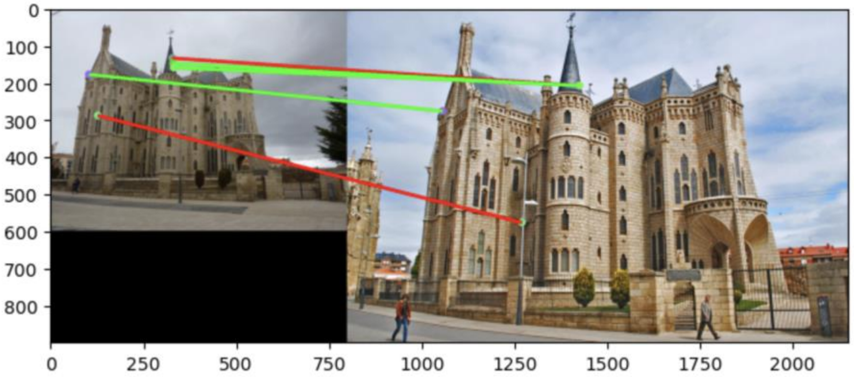


matches (out of 100): 198

Accuracy: 0.919192

Part 4: SIFT feature descriptor





matches: 177/100

Accuracy:0.937853

matches: 5/100

Accuracy:0.03

Part 4: SIFT feature descriptor

1) Leveraged sobel kernel to get gradient in x & y direction for every pixel in input image
2) Used gradients from (1) to determine gradient magnitude and angle for each pixel
3) For each pixel of INTEREST in the image, I extracted region of dim feature_width x feature_width centered around pixel of interest from the image
4) For each of these regions, created a histogram outlining the frequency (weighted by gradient magnitude) of different gradient orientations for each pixel in the region
5) Combined all histograms into one flattened vector and returned

SIFT descriptors are better than normalized patches as they generalize what occurs in a region as opposed to looking at specific element wise similarities. By evaluating regions via a histogram, the order of values in a region do not matter. All that matters is the magnitude/orientation of gradients. As a result, unlike normalized patches, smalls shifts up/down/left/right do not drastically affect the description of the region leading to more accurate similarity scores

Part 4: SIFT feature descriptor

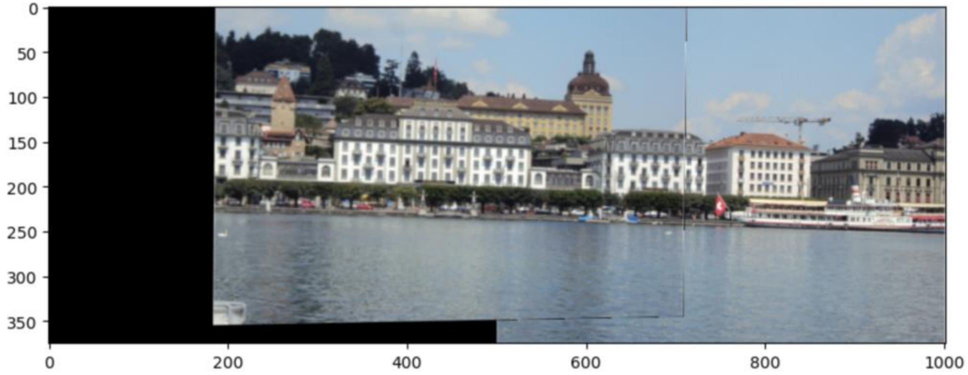
The differences in lighting between the two images in gaudi images have large differences in lighting which makes it harder to classify. Furthermore, in our implementation, the sift algorithm is not scale invariant. In the Gaudi example the photos are taken at very different scales (ie different distances away from the subject). On the other hand, the photos of Notre dame and Rushmore, have more similar scales, making it easier for SIFT to accurately describe regions such that they can be correctly matched.

Part 5: Panorama Stitching

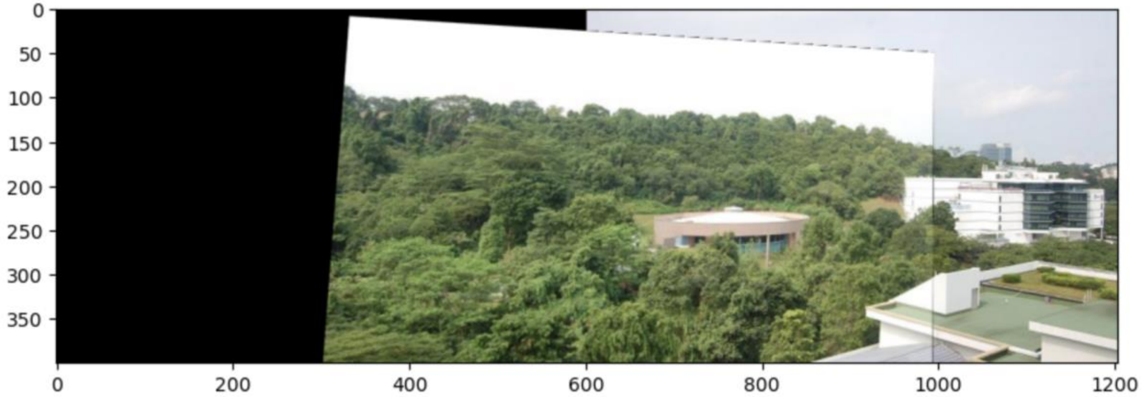
This implementation performs panorama stitching by detecting interest points, matching features, computing a homography matrix, and manually warping one image into the coordinate space of the other. First, the input images are converted to grayscale to extract feature points using the Harris corner detector. The SIFT descriptor is then applied to these interest points to generate feature vectors, which are subsequently matched using the ratio test.

A blank canvas is initialized with dimensions set to accommodate both images side by side, where the second image is copied directly onto the right portion. The matched feature coordinates are stored separately for both images, with the second image's points shifted to the right to align properly in the final stitched output. Next, the homography matrix is computed using cv.findHomography() with RANSAC to filter outliers. The first image is then warped onto this canvas using cv.warpPerspective(), mapping its coordinates according to the computed homography matrix. Finally, a mask is applied to ensure that only valid, non-black pixels from the warped image overwrite the corresponding pixels in the result. This approach ensures a smooth blend while preserving the structural integrity of the panorama without relying on pre-built warping functions. To recreate the results shown on previous slides, simply run the cells under part 5: panaroma stiching in the project-2 jupyter notebook

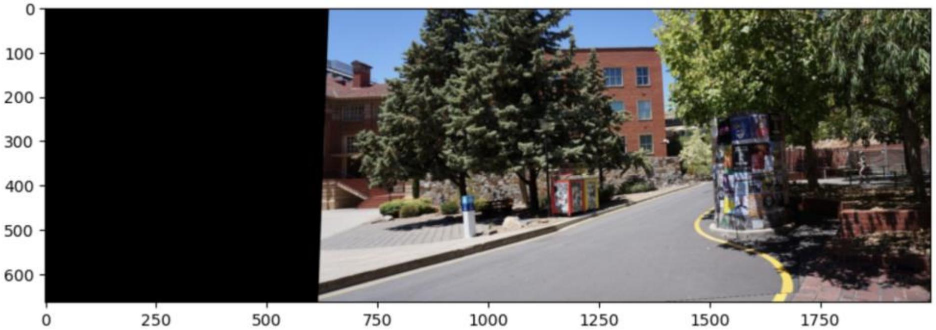
Part 5: Panorama Stitching: Image Pair 1



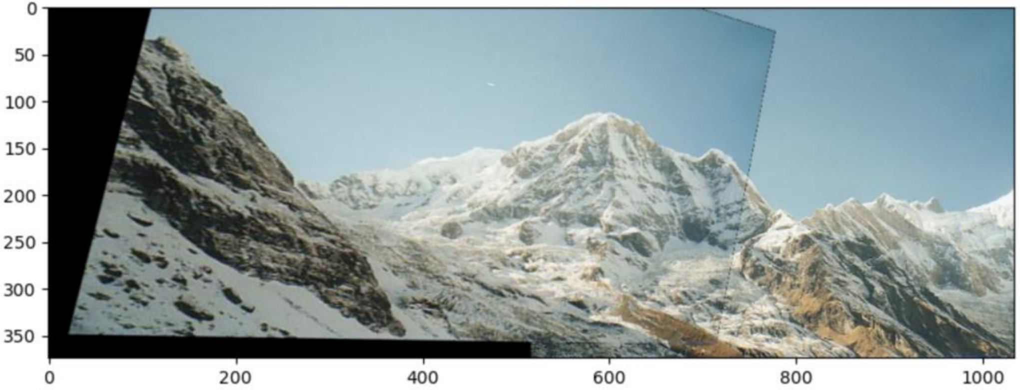
Part 5: Panorama Stitching: Image Pair 2



Part 5: Panorama Stitching: Image Pair 3



Part 5: Panorama Stitching: Custom Image Pair



Extra Credit: Warp Implementation

[Please add a README style documentation here for your implementation of warping in panaroma stitching: description of what you implemented, instructions on how to replicate the results in clear steps that can be followed by course staff. Failure to replicate results by following this documentation will result in point penalties on this question of the assignment.]