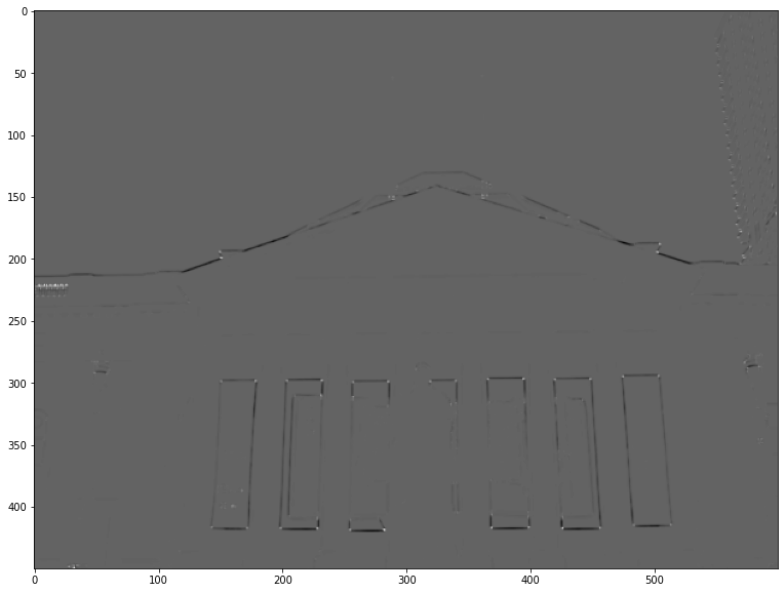
Corner detection

- I selected the corners using cv2.cornerHarris. The function returns a matrix of corner scores, I showed it as an image.
- We can see that the white spots are corners. The corners are not spread across the entire image so we need to run ANMS on it to select a more diverse set of corners.

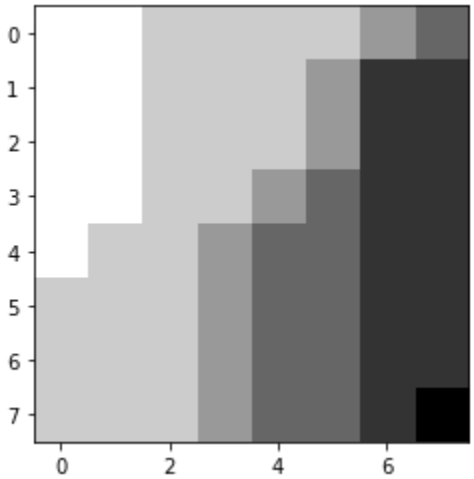


ANMS

- First get the values using list comprehension. This vectorized operation reduces the runtime and prevents us from needing to get the values in each iteration
- The comparison in the for loop is done as boolean indexing. Only the indices where it is true (the value is significantly smaller) has a true value, so only those are returned for the strong points.
- Use broadcasting the compute the sum of squared distance for each pair of points. Use np.min to get the min.

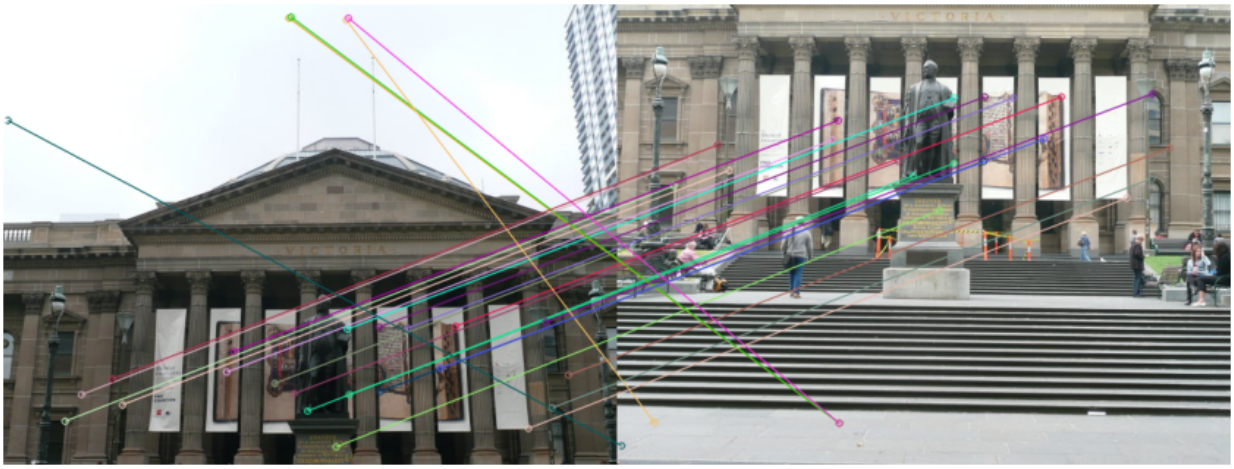
Feature descriptor

- Follow the instructions as stated in the article
- Subsample the 40x40 region with a spacing of 5 after applying gaussian blur
- The result is a 64x1 vector that has been reshaped (8,8) to display



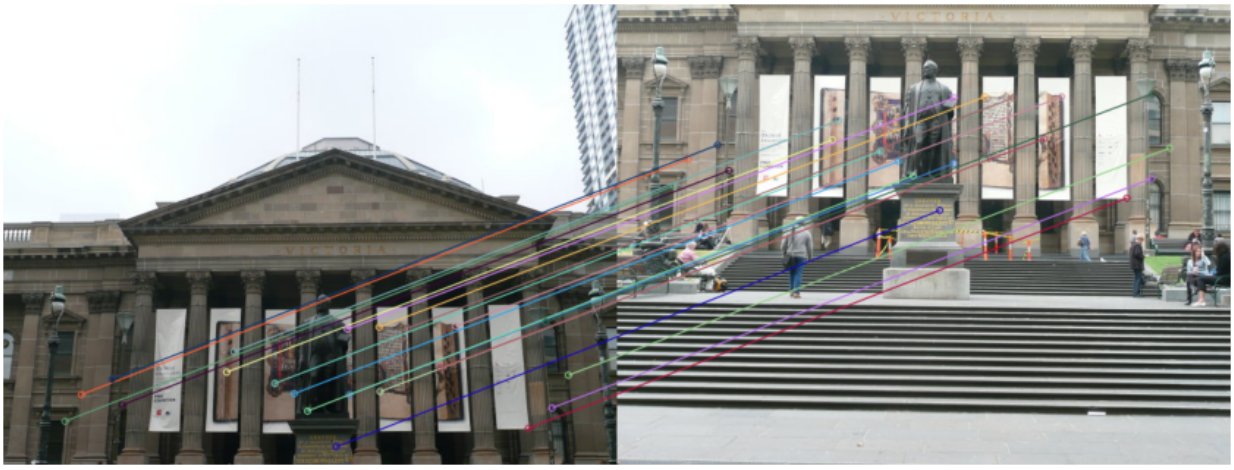
Feature matching

- For each feature point compute the distance to each other feature point with broadcasting. We can then take the shortest 2 distances using sorting then selecting the first 2 indices
- Run the function for each feature vector. If there is no match (i.e., the ratio of the smallest is not large enough) then None is returned and the feature vector is ignored.
- The result is a list of matching indices between the two lists of interest points
- We get a good number of matches, about ¼ or ⅛ of the original interest points



RANSAC

- The ransac algorithm is also implemented as shown in the article
- A function try homography to select a random 4 pairs of points and try the homography, returns a list of inliers
- Run n_iter times and get the largest set of inliers
- Chose n_iter=300 because that seemed to be sufficient to get a consistent accurate homography



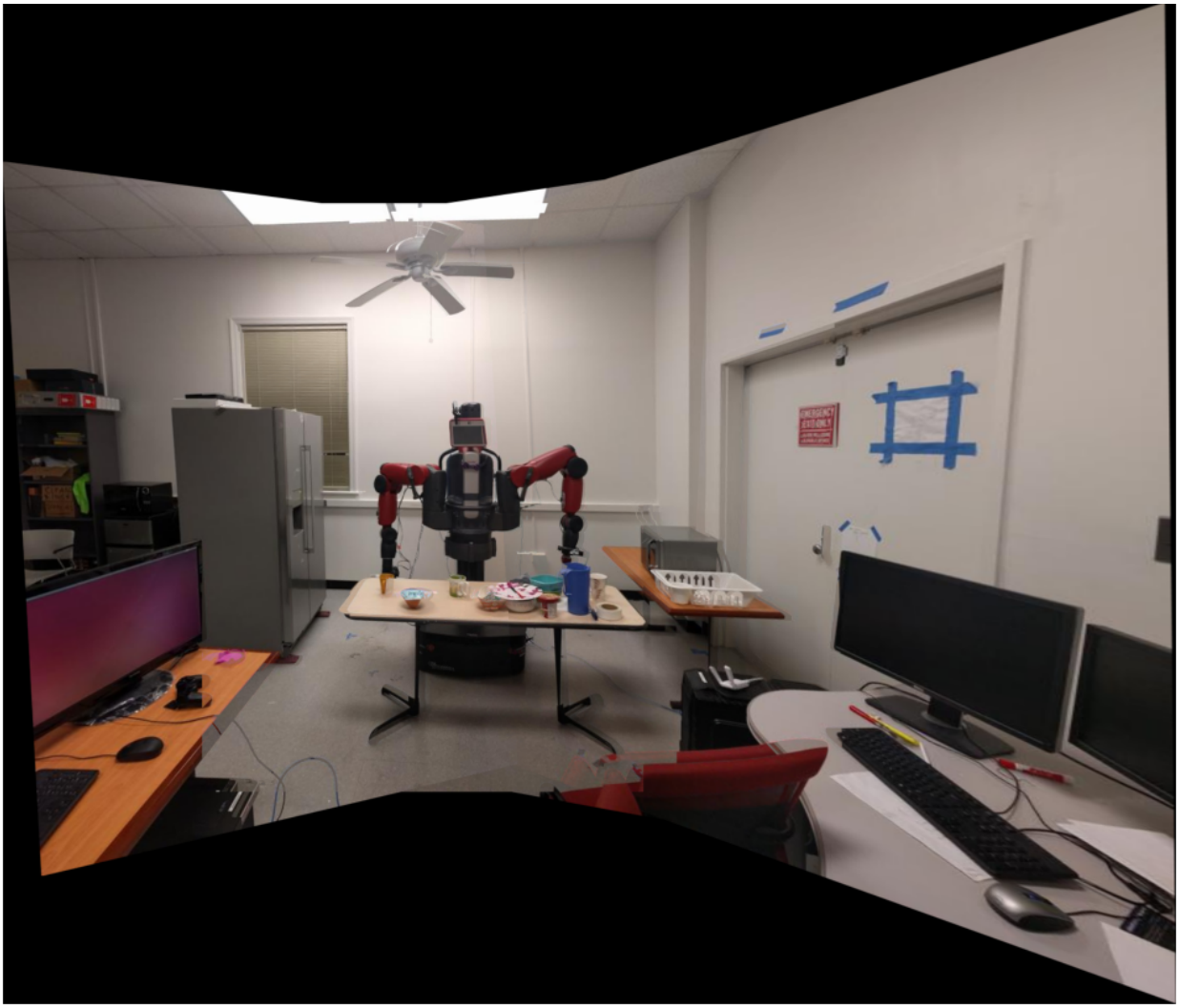
Blending

- To blend the images I created a new homography matrix that was transformed by the amount the canvas moved in the x and y directions. The new matrix was then multiplied to the original homography matrix.
- We apply the new H to the first image and apply the transformation matrix to the second image. Since the second image only needs to translate x and y the matrix will do it
- Blend the images using np.max, although there are still visible seams

Final stitched image


VICTORIA


HAMARKSHR
35JOMILUR


EMERGENCY
EXIT ONLY