P1-Phase1
AutoPano

Smit M Shah
Email: smshah1@wpi.edu
Worcester Polytechnic Institute

Rigved Sanku
Email: rsanku@wpi.edu
Worcester Polytechnic Institute

***Abstract*—There are two parts to this paper, Phase-1 focuses on implementing Panorama stitching from a series of images. Phase-2 focuses on developing a deep-learning model to estimate homography between 2 images. The model is trained using two different approaches, namely, Supervised and Unsupervised Learning.**

I. Phase-1 : Traditional Approach

A. Introduction

The traditional approach of generating a panorama from numerous images includes the following steps: 1) Corner Detection, 2) Adaptive Non-Maximal Suppression(ANMS), 3) Feature Description generation, 4) Feature Matching, 5) Random Sample Consensus(RANSAC) and 6) Stitching and blending

B. Corner Detection

We are using corners to identify unique features of the images as they are an intersection of two edges. Harris Corner is used as the algorithm to extract corners. The *cv2.cornerHarris* function is used, it uses grey-scale images hence images were converted to grey scale before the function. The before and after image(colored) is as shown in the Figure 1.

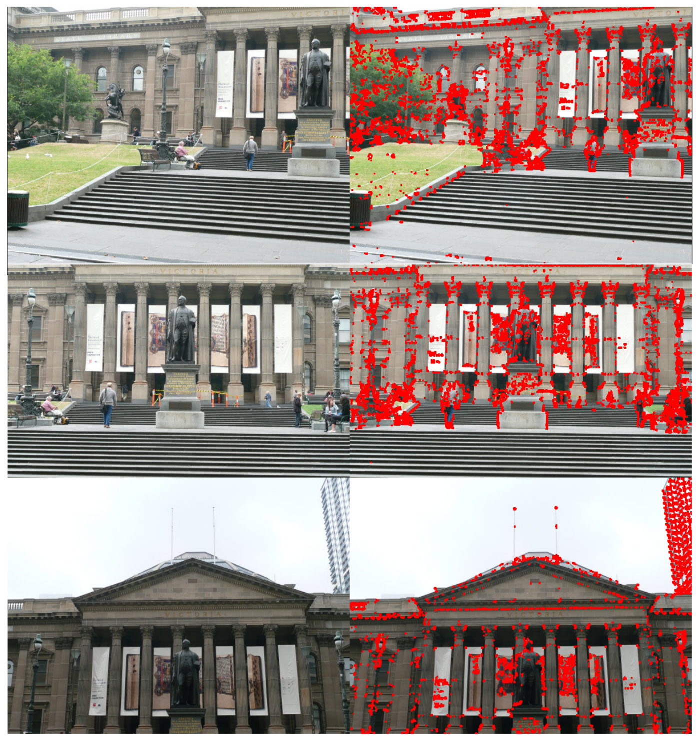

Fig. 1. Before and After Harris Corner Detection

C. Adaptive Non-Maximal Suppression

The corners detected by Harris Corner are a alot in quantity and are not the best corners as well. Therefore, ANMS is used to select N_{best} corners that are uniformly distributed across the image. A minimum distance of 10 pixels is selected. *peak local max* function of *skimage.feature* was used to find addresses of local maxima peaks. The corners selected are displayed in Figure 2.

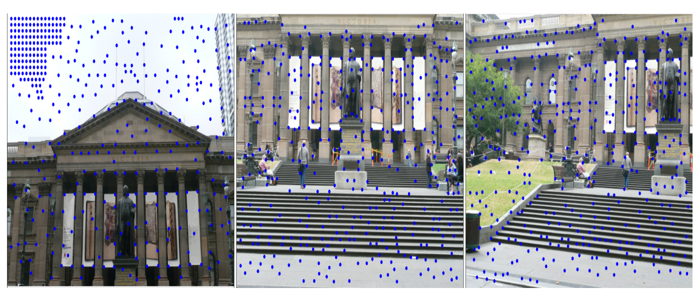

Fig. 2. ANMS

D. Feature Description Generation

With the N_{best} strongest corners obtained after ANMS, A patch of 41x41 with each corner as centre is used to extract features around each corner. To include corners near the edge of image a padding of 20 pixels was made around the image. A Gaussian filter of kernel size=5x5 σ=0 was used to smoothen the features of the patch. Initially, this patch was then downsampled to 8x8 but later to 16x16 as it improved the accuracy of the features. It is then flattened into a 256x1 element vector, which was standardized and normalized.

E. Feature Matching

Next, our objective is to align the features in the two images intended for stitching. For each feature vector extracted from the first image, we compute the sum of squared differences with all the feature vectors in the second image. If the ratio between the top-matching pair and the second-best matching pair is below a certain threshold, we retain the best matching pair. This process is then iterated for all feature vectors in the first image. These matched pairs of features can be seen in Figure 3.

Fig. 3. Feature Matching

F. Random Sample Consensus(RANSAC)

From Figure 3 it is visible that certain matches are wrong and hence need to be removed. This function is conducted using RANSAC method. 4 corner pairs are selected at random and used to calculate the Homography between the two images using *cv2.findHomography* function. Multiplying corner points of source image with the estimated homography, transformed points are calculated. A Sum of Square of Differences(SSD) of transformed points and the destination corner points and maximum number of corners with SSD under a threshold are recorded and the Homography estimated at that point. These steps are repeated over 1000 times, the best Homography is selected and the filtered corners are recorded. The new pairs are displayed in Figure 4.

Fig. 4. Corner Pairs After RANSAC

G. Stitching and Blending

Number of **Unique** corners of source and destination image each are calculated, if the ratio between the counts is less then 0.6 or 60% then the image is dropped and further stitching is stopped as there are not enough common features to perform stitching.
The Homography opted from RANSAC is used to first transform the image corners of source image using *cv2.perspectiveTransform*. The min and max values among the image corners of both images is extracted. The source image is then warped *cv2.warpPerspective* using the Homography and translation calculated from min and max edge values. The destination image is then overlaid over the warped image.

Fig. 5. Image 1 and 2 stitched

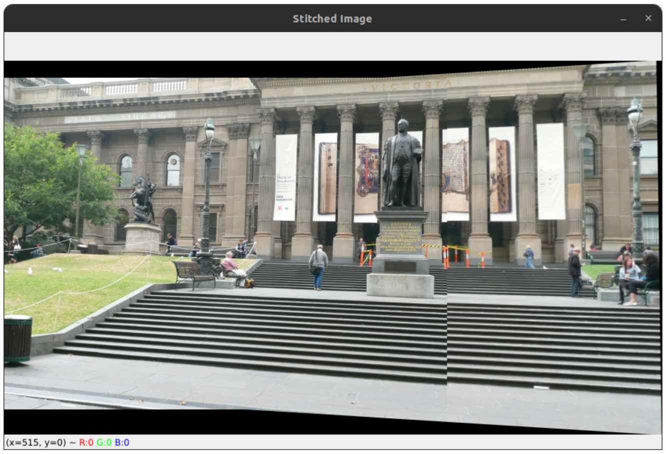

Fig. 6. Image 2 and 3 stitched

H. Stitching Multiple Images

The order used to stitch multiple images is by selecting 2 simultaneous images, perform stitching and saved. Other 2 images are selected and stitched together and then saved. The process is repeated, once images are exhausted, the stitched images are then selected and stitched together. Though the algorithm is slow compared to stitching every alternate image, the results obtained were better in quality.

Fig. 7. Image 1,2 and 3 stitched

I. Results

Our algorithm was able to stitch together museum, mountain and the office(struggled) dataset. From the Test Case uploaded recently, our algorithm could create a Panorama successfully of the chess board, corridor and the office. To stitch the office with white board dataset we had to reorder the images, as the order received was continuous which resulted in failure in generating a proper panorama. The algorithm was also able to reject stitching unrelated images together. The algorithm however struggled to generate good panoramas when the number of images exceeded 5 images.

Fig. 8. Set1 - Museum

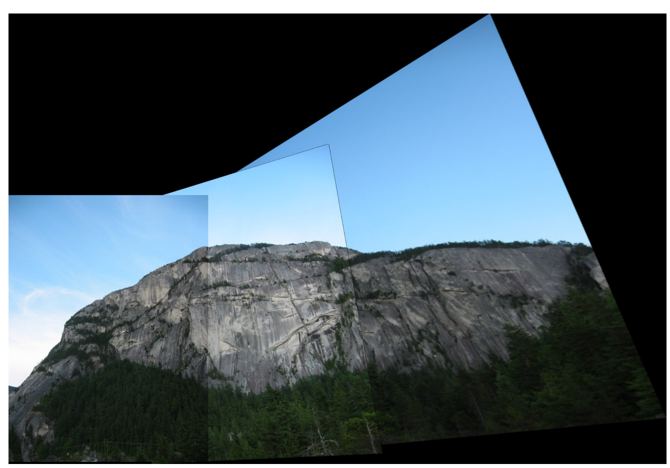

Fig. 9. Set2 - Hill/Mountain

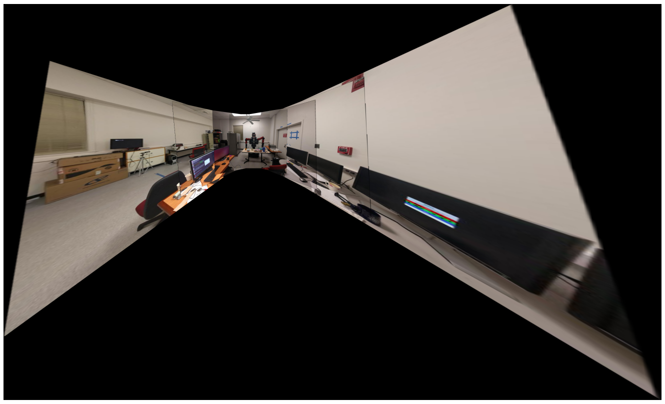

Fig. 10. Set3 - Office with robot

Fig. 11. TestSet1 - Chessboard

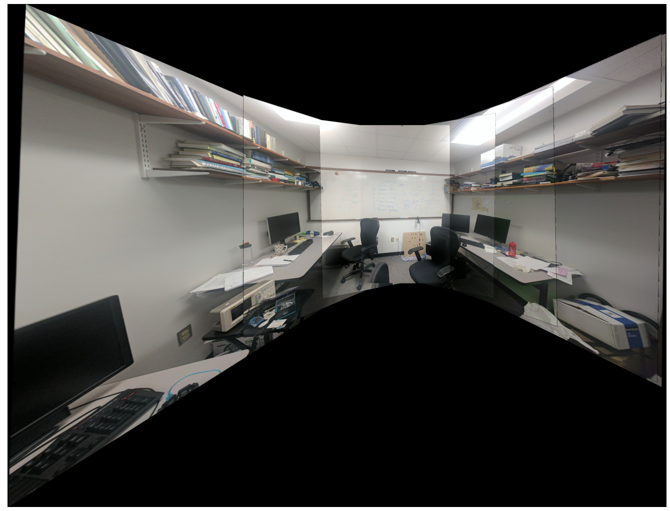

Fig. 12. TestSet2 - Office with board(rearranged)

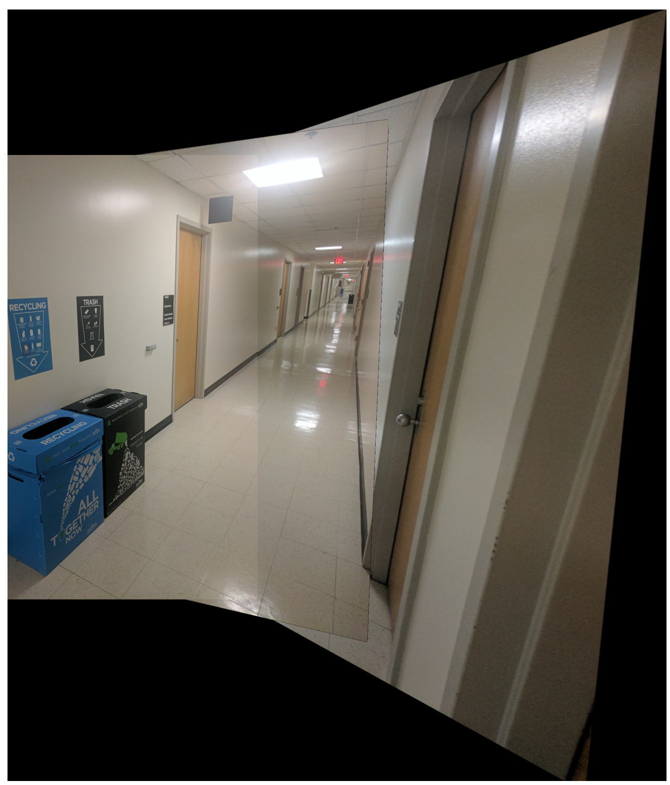

Fig. 13. TestSet3 - Corridor

Fig. 14. TestSet4 - Unrelated Images