# CMSC733: Project 1, AutoPano!

Using 4 Late Days

Naveen, Mangla
nmangla@umd.edu

Bhargav Kumar Soothram
bsoothra@umd.edu

***Abstract*—We know panorama as an unbroken view of the entire region surrounding an observer. In this project, we worked on an algorithm by which we created a panorama using feature matching and filtering the matched feature using RANSAC, We also used the deep learning technique to get the results.**

## I. Phase 1

### A. Algorithm

The Algorithm can be divided into following steps.

1) Extracting Features
2) ANMS [Adaptive Non-maximal Suppression]
3) Matching Features in two images
4) RANSAC
5) Stitching Using matched corners

Steps are explained as follows:

### B. Extracting Features or Corners

The feature of the image is termed as the part with maximum intensity variation in different directions. These are essentially the corners of the image. The two methods of detecting corners are Harris corners detection and Shi-Tomasi Corner Detector. We are using the latter one for our project.

Fig. 1: Corners Set 1

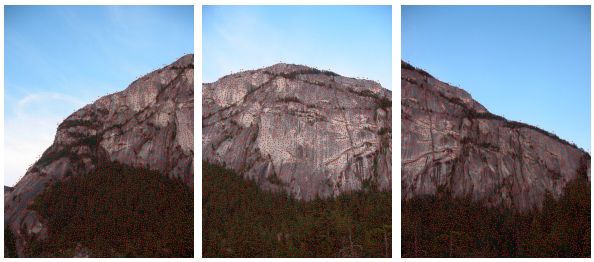


Fig. 2: Corners Set 2

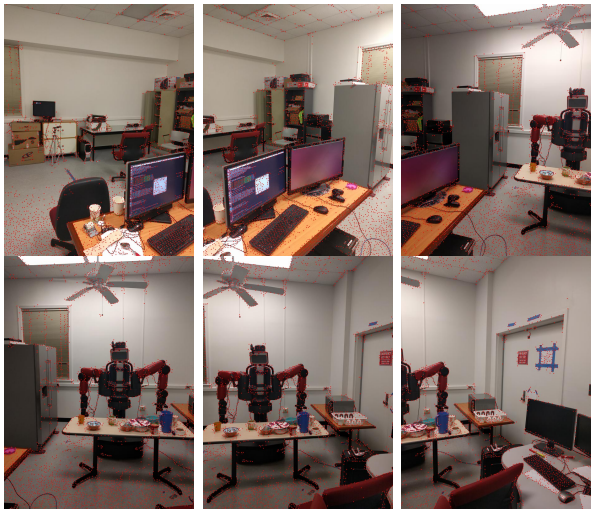
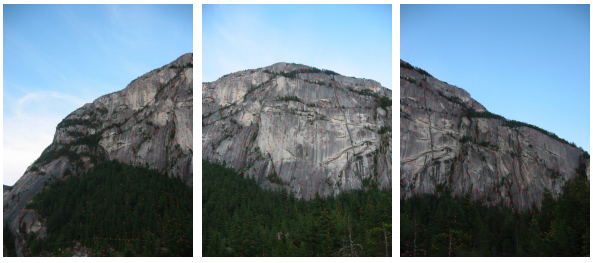


Fig. 3: Corners Set 3

### C. ANMS

ANMS -Adaptive Non-Maximal Suppression, is a non-maximum suppression algorithm that applies a dynamic suppression threshold to an instance according to the target density. We use the ANMS algorithm to detect peak, or local maxima of the corners and suppress all nearby corners.

(a) Corners Set 3



(b) ANMS Set 2

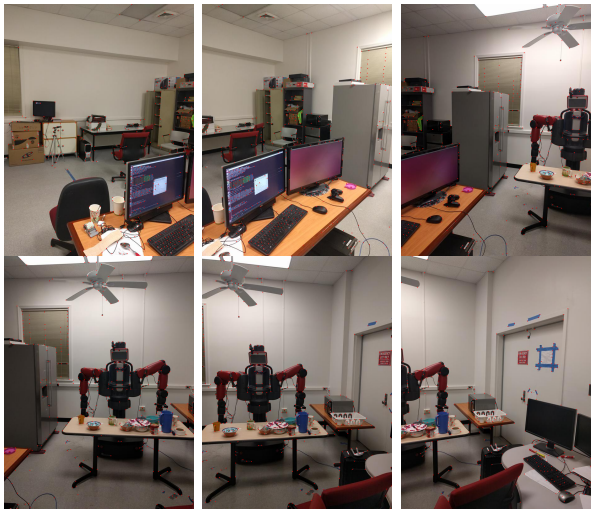


Fig. 5: ANMS Set 3

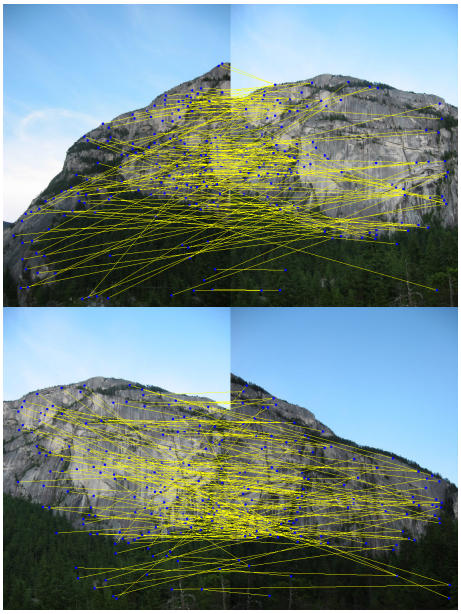


Fig. 8: Matching features SET 2

### *D. Matching Features*

In order to compare features, a patch of 41×41 was extracted with the feature location as the center, a Gaussian filter of scale 5 was than applied. This patch then reshaped into 64×1 vector and SSD (Sum of square difference was calculated). By this way.The lowest difference was than compared with second lowest by taking ratio. This ratio is than compared with 0.9 (arbitrary value), pair with SSD ratio less than 0.9 was kept ans other were discarded. Since they were too many, few of them are shown below.

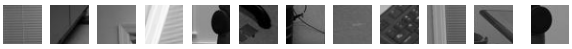


Fig. 6: Some Feature Patches

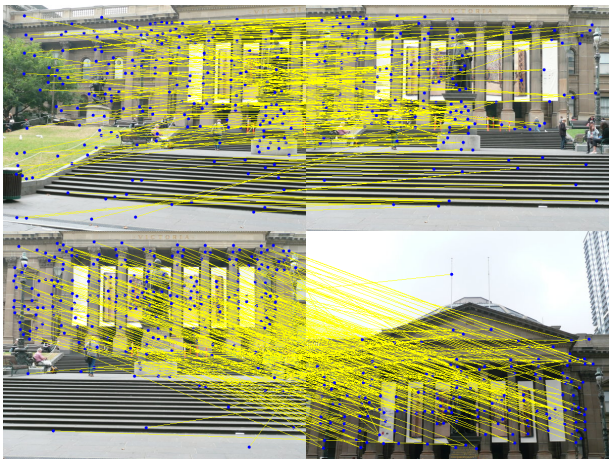


Fig. 7: Matching features SET 1

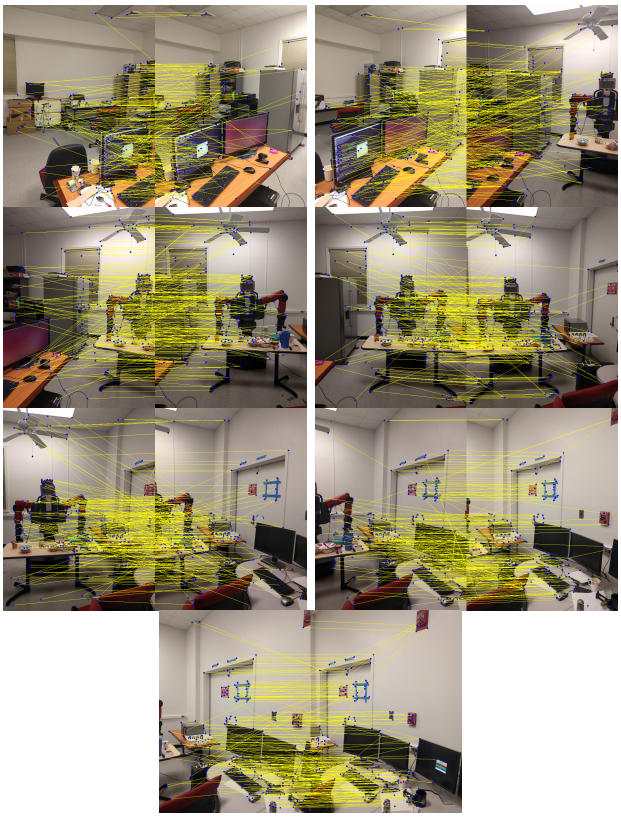


Fig. 9: Matching features SET 3

### *E. RANSAC*

After having all the Matching pairs, we need to filter them out, as many matching pairs are nothing but noise, this noise is known as Outliers. To get rid of these outliers, RANSAC was used. The response after RANSAC is given below.

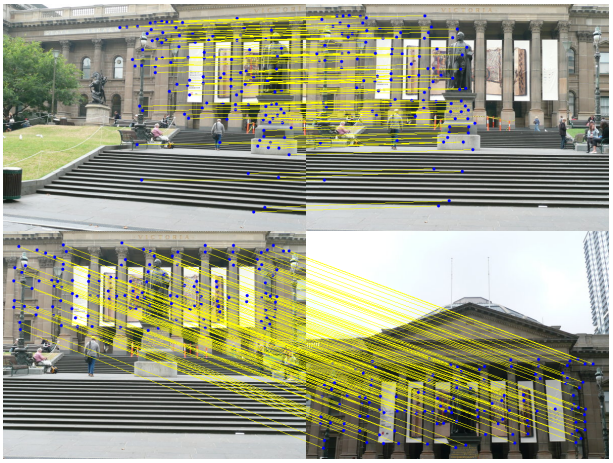


Fig. 10: RANSAC SET 1

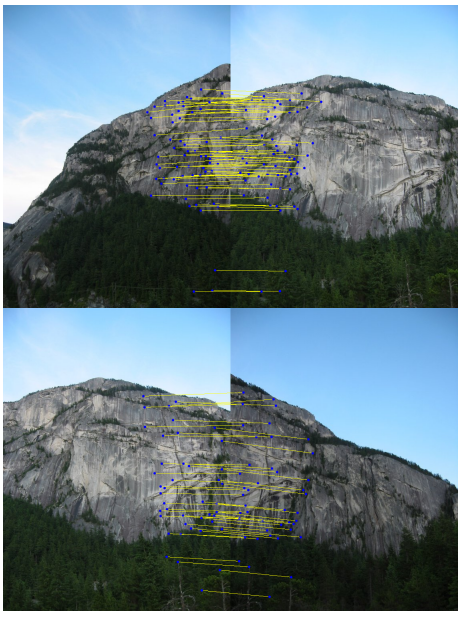


Fig. 11: RANSAC SET 2

Fig. 12: RANSAC SET 3

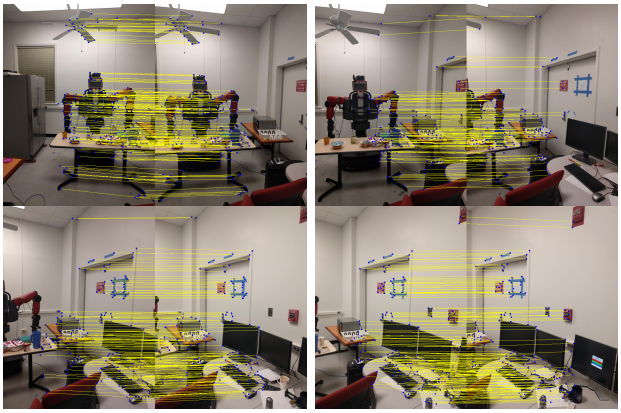


Fig. 13: RANSAC SET 3

### *F. Panorama*

The last thing need is the Panorama, For this we calculated the homography matrix between $1^{st}$ and $2^{nd}$ and the resultant is treated as $1^{st}$. The cycle goes until the final Panorama reached. However, we are not able to get perfect results. One of the panorama is given below.(As Discussed, we were getting a *Hard Pair* after RANSAC)

Fig. 14: Panorama SET 2

Our algorithm does not work when images are not in a constant order. Thus we had to drop image 3.jpg from Set 1.

Fig. 15: Panorama SET 1

With this, primitive approach and having trouble getting some outliers after RANSAC, we are not able to blend Sets with large number of images. Thus Set 3 is not posted here.

## II. Phase 2

Phase II includes getting homography and stitching images using deep learning approach.

### A. Data Extraction

The image from data is resized to $340 \times 240$. Than a random patch of size $128 \times 128$ was extracted from resized image. We used a threshold value 10 to avoid cornering effect. The patch is then transformed using initial corners and corners resulted from random perturbation in range $[-\rho, \rho]$, ($\rho = 32$ in our case). These patches are than stacked one upon the other to get the final 2 channel input image.Ten such patches were created for each image.

### B. Architecture

We use following network architecture for our supervised model. The architecture can be seen in figure 16. Input is the stack of of patch A and patch B, and the output is the 4-point homography

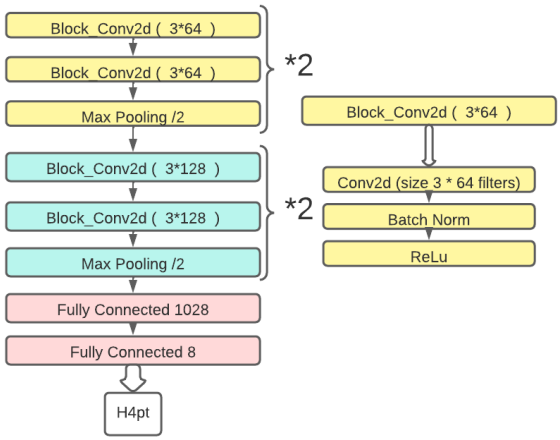



Fig. 16: Architect Followed

We trained the model for 25 epochs and got the following results. Due to lack of time, we were unable to test the model and work on the unsupervised part of the model. However the loss function in the model seen a significant decrease. We have created the Network and Train files for the model. The loss went down from 342 to 210 in 25 epochs.


Fig. 17: Loss

### C. References


1) https://arxiv.org/abs/1709.03966
2) https://arxiv.org/pdf/1606.03798.pdf
3) https://machinelearningmastery.com/tensorflow-tutorial-deep-learning-with-tf-keras/
4) https://machinelearningmastery.com/tensorflow-tutorial-deep-learning-with-tf-keras/
5) https://arxiv.org/pdf/1606.03798.pdf
6) https://cs231n.github.io/