RBE549 : Project1 MyAutoPano

Chinmay Kate
M.S. Robotics Engineering
Worcester Polytechnic Institute (WPI)
Worcester, MA 01609
Email: cskate@wpi.edu

Mandeep Singh
M.S. Robotics Engineering
Worcester Polytechnic Institute (WPI)
Worcester, MA 01609
Email: msingh2@wpi.edu

***Abstract*—In this Project we propose methods to stitch images in order to create seamless Panorama image. Given Pair of Images should have common regions, Our goal is to stitch them to produce Panoramic Image scene. We implemented both Classical Computer vision Method as well as Deep Learning methods to get the desire output. In Classical method called Feature based technique which determines relation between images through features. DL techniques focus more on the Intensity and Homography matrices which will be explained more in the respective sections. We also computed output with the unknown dataset. This was tricky as our hyperparameters of our algorithm in Phase 1 were tuned with the known dataset, still manage to produce interesting panoramic scene. Github Link**

***Index Terms*—MyAutoPano, Mosaicing, Panorama stitching, Image Stitching.**

I. PHASE 1: CLASSICAL FEATURE BASED TECHNIQUE

Feature Based Methods are used to extract features that shares in the common regions of the images. We match these feature keypoints and estimate Homography between these set of points from two images. We can use this Homography to warp the second image with respect to the first which will then be use in stitching the images with the reference image. To counter illumination problem or unexpected generation of artifacts after image stitching we seamlessly blend the images with blending techniques.

In Summary, this work will have following contributions:

1) Corner Detection using Harris corner Detection as Feature Keypoints.
2) Adaptive Non-Maximal Suppression using sum of Sqaure distance to avoid artifacts during warping and get best corners out of Harris corner detection.
3) Feature Descriptor standardize the keypoints and it's neighbouring pixels and uses Gaussian blur to achieve illlumination invarience at certain level.
4) Feature matching finds feature correspondences between two images, taking distance between first best match and second best match(lowest).
5) RANSAC is use to ignore the outliers by setting the threshold and Compute Homography matrix of the inliers of both the Images.
6) With the Homography we can convert one Image to the Perspective of another which overlaps the common regions giving the stitched Image.
7) We finally blend the overlapping images to avoid illumination invarience after stitching.

II. PROPOSED METHOD

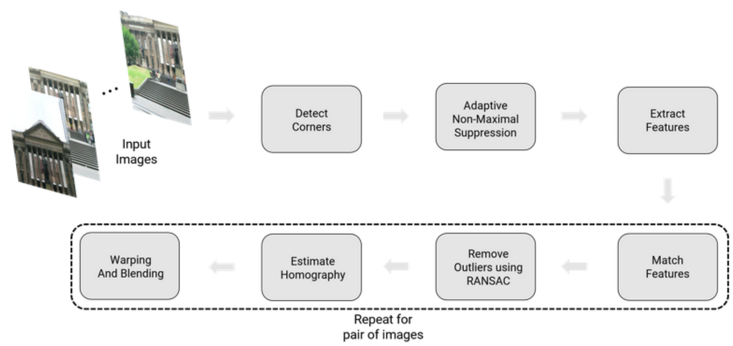

Fig. 1. Pipeline for Panorama stitching

Fig. 2. Sample Images having common regions

A. Corner Detection

Corners are one of the best features in the Images but these features are illumination, rotation and scale invariant. We implemented Harris corner detection and Good Feature Detection function from OpenCV. Both has different results on different Image-sets. Good Feature Detector uses Shi Thomasi corners detection, suppresses the corners in the region and gives the maximum from the region. This gives best results. Again there are different image dataset where good features to track gives worst features and can't even detect feature corners in low lighting conditions.

B. ANMS

ANMS is used to suppress the corner points that are close to one another, This will give you equally distributed points in the image by taking local maxima. This feature is heavily required in Harris corner detection. It takes the regional maximum and give maximum in that window. Good features to track function already does this and don't need to use ANMS in this case.

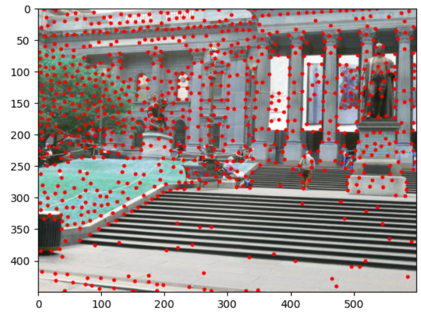

Fig. 3. Features detected from Good feature to track.

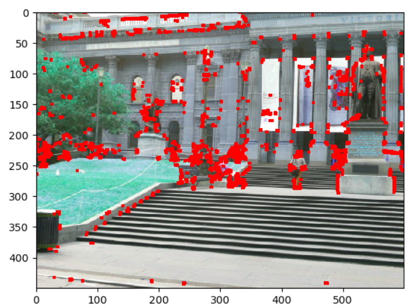

Fig. 4. Features Detected from Harris Detection

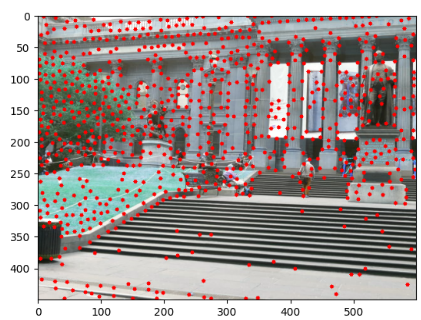

Fig. 5. Good features to track after ANMS

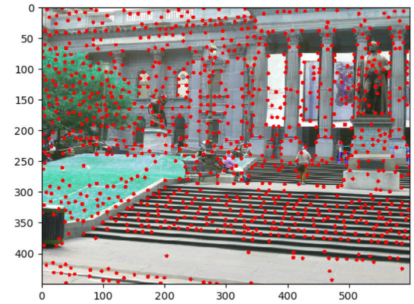

Fig. 6. Harris Corners after ANMS

C. Feature Descriptor

Here each Feature points around their neighbors are vectorized, Gaussian blur and normalization is applied to counter the illumination invarience and artifacts at some level during stitching. There are various feature descriptor used which addresses the problem of scale, rotation, illumination.

D. Feature Matching

In Feature matching we find the feature correspondence between the two images with the sum of square distance. We take the ratio of lowest first to lowest second and compare the ration with the threshold to get the best matches. Threshold can be varied according the acceptance level of the feature mapping. We draw the matches between images as shown in the figure.

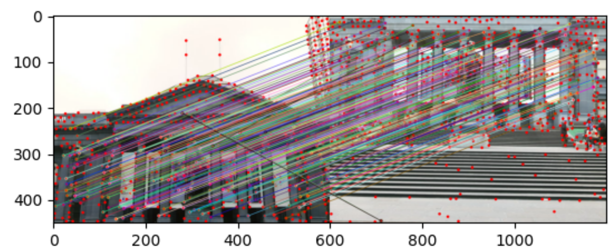

Fig. 7. Feature matching

E. RANSAC

We know that all matches we get are not important which are called outliers. If we don't remove these outliers they will cause problem in stitching. We also compute Homography matrix after eliminating outliers. Homography is the transformation between 2 images. Homography has rotational and translation components in 2D space. Thus we finally find Homography matrix of inliers of both the match pairs. We keep on iterating till we get good inliers from the match pairs.

1) For Outliers Rejection: In Outliers rejection we compute Homography matrix and multiply its inverse with the image to get the second image reference to first image, We compute predicted match with the target match and threshold the outliers.

2) To Compute Homography: We compute Homography again on the inliners after we reject them earlier on basis of

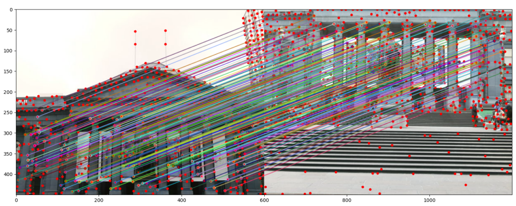

Fig. 8. Feature matching after Ransac

threshold. We pass this homography matrix to get the warp perspective with the image 1.

F. Warping and Stitching

In Warping we compute warp perspective with respect to image 1 by simple multiplying inverse of Homography matrix with image 2. We make the container by finding the xmin,xmax and ymin,ymax of the newly formed image. It will store the newly formed image in the container. Thus we get the stitched image.

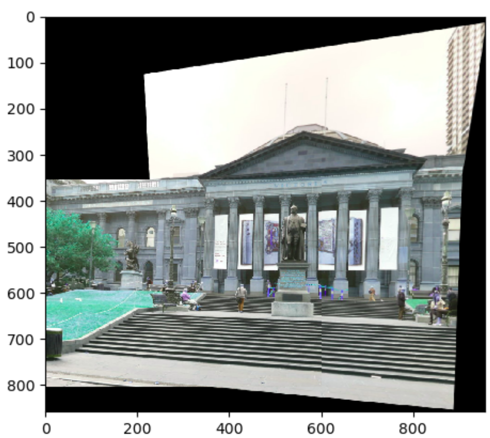

Fig. 9. Images in Set After stitched.

G. Blending

The image stitched needs to be blend here we counter the illumination invarience and artifacts that occurred in stitching. We use function from OpenCV called weighted alpha blending. It helps in some sort and gets rid of the artifacts.

H. Conclusion and Results

First most important is feature detection, Which is very vital in any Panoramic stitching scene. We could have used better features which counters illumination, rotation and scale problems like SIFT, FAST, ORB. Our Harris Corner detector gave good results when used with ANMS. Secondly we believe

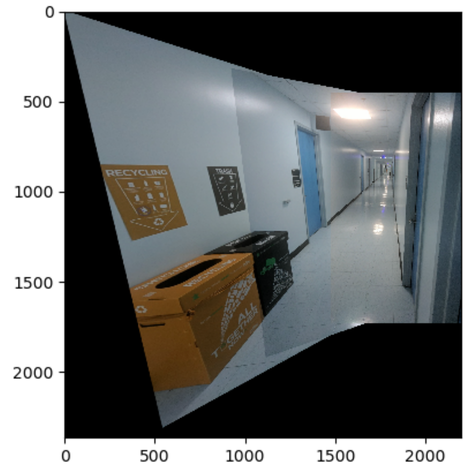

Fig. 10. Stitched for Testing-Set

RANSAC to be more efficient as it has many number of parameters that needs to be hand-tuned. Some set of parameters works with one but doesn't work with rest. Also Blending was not as perfect as we expected it to be. There are many good Third party blending functions which has better results like poission blending.

III. Phase 2: Deep Learning Method

Deep learning methods estimates the Homography matrix between two images. We implemented on both learning methods (Supervised and unsupervised learning). This models learns the corners of the patches and tries to predict the warp of the same patch which is perturbed as label set. This method is called 4 points parameterization.

A. DataGeneration

This is common to both supervised and unsupervised learning models. Here we need data for our Homography-Net in pair of images. We have subset of MSCOCO dataset. In data generation we select random patch from the Image-1 we get get 4 corners with the image, we perturbe the corners with some range and get the warped patch. Then we stack them into the channels and pass to the Homography Network.

B. Network

Our Networks uses 3 X 3 convolution block with batch norms and RELUs, Both Network takes 128X128 size grayscale images as inputs. It has 8 convolution layers and 2 fully connected layers. It has dropout layers with probability of 0.5. STL Network used in the unsupervised network has localization layer which computes H inverse matrix from the H4pt. We build affine mattrix layer with fully connected layers to get the predicted patch B.

C. Supervised Model

In this model we have patch A and patch B, we can calculate H4pt matrix which is the ground truth. We compare this with predicted H4pt matrix and find the L2 loss. This network architecture is modified VGGNet model.

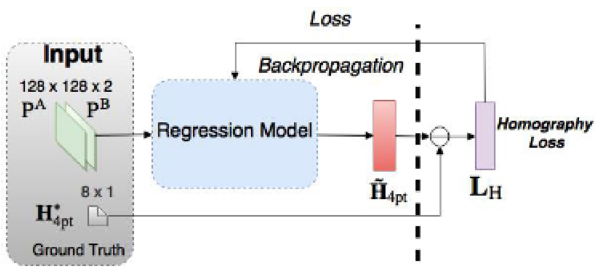

Fig. 11. Supervised Learning pipeline

D. Un-Supervised Model

Supervised model is more biased so there is more scope in improvement, In this model We implemented the Tensor-DLT algorithm as suggested in the paper 'Unsupervised Deep Homography: A Fast and Robust Homography Estimation Model' This Converts H4pt predicted to Homogeneous matrix. Also Spatial Transformation layer is implemented to compute Photometric loss which is the L1 loss. We back propagates to optimize the weights. This model is not biased like the supervised model.

TABLE I
HYPERPARAMETERS FOR HOMOGRAPHY-NET MODEL ARCHITECTURE

Hyper-parameter	Value
Optimizer	SGD
Learning rate	0.0001
Mini Batch Size	10
No. of Epochs	50
Dropout	0.5

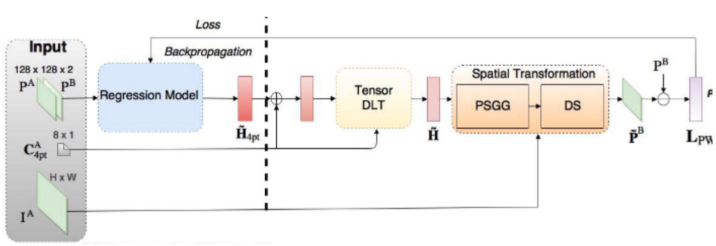

Fig. 12. Un-Supervised Learning pipeline

IV. Conclusion

Supervised model has good output after epoch 15, we trained our model till epoch 50. It has considerable low loss and goes till 14.4 from 89.2. But the Validation loss kept on increasing due to overfitting. Also general trend of validation loss kept on increasing. Something is wrong with the validation. Unsupervised model sees better output in terms of loss, It also goes low right from starting and goes low till 14.4 at epoch 32 from 93.5. It generalize better than the supervised model.

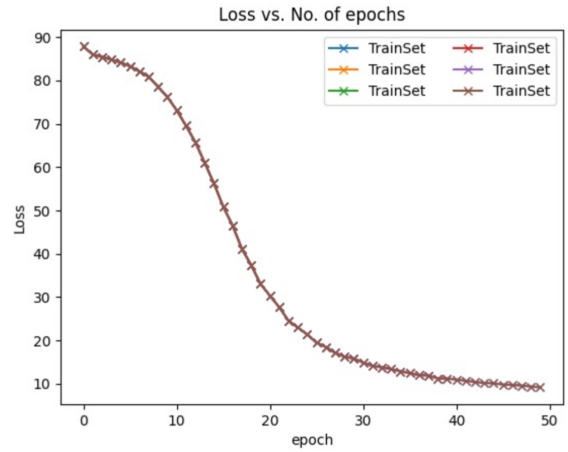

Fig. 13. Supervised Loss vs Epochs

Fig. 14. Supervised Prediction vs Target

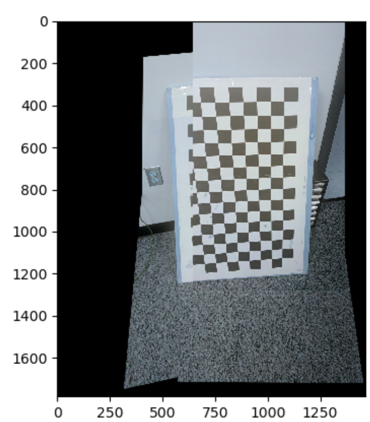

Fig. 15. Checkers Panorama

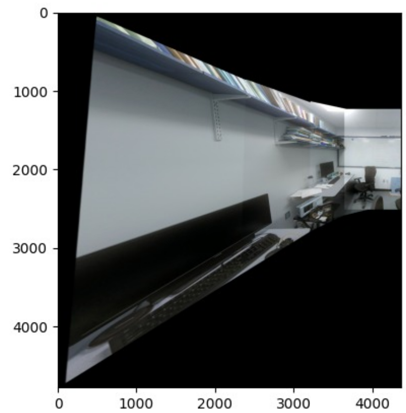

Fig. 16. Stitching for another Testing Set

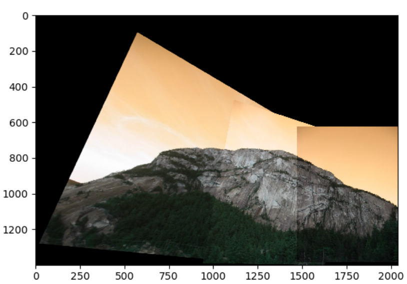

Fig. 17. Outputs after Stitching on Train Set

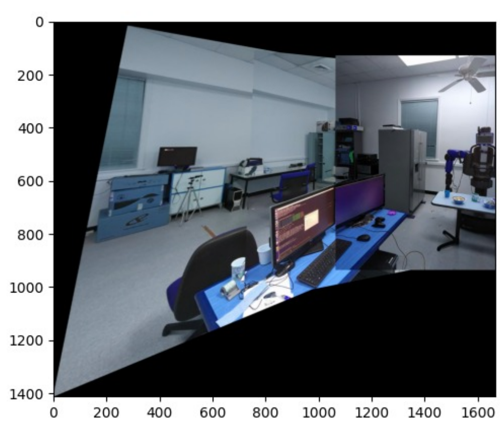

Fig. 18. Stitching for another Testing Set