# CMSC733 Project 1: MyAutoPano

Vikram Setty
University of Maryland, College Park
Email: vikrams@umd.edu

Vinay Lanka
University of Maryland, College Park
Email: vlanka@umd.edu

Mayank Deshpande
University of Maryland, College Park
Email: msdeshp4@umd.edu

***Abstract*—This report overviews various panorama stitching techniques and goes over the mathematical foundations, methodology, and implementation. In the first phase, a traditional feature-matching-based approach is discussed followed by a deep-learning-based homography estimation technique using augmented and generated data using supervised and unsupervised techniques in phase two.**

## I. Phase 1: Traditional Methods

The traditional approach of panorama stitching in our implementation includes Harris Corner Detection on each image as a first step. Adaptive Non-Max Suppression is then used to filter out duplicate corners representing the same feature in the image. After that, features between images are matched using a brute-force method with matched features having satisfied the ratio test. Then, using RANSAC (random sampling consensus), the best homography relating the two images is calculated and the first image is warped and stitched onto the perspective of the second image. This process is repeated to stitch all images in a set into a single panoramic view.

### A. Corner Detection

Feature detection is done using Harris Corners on the grayscale version of the color image. All the Harris corners are thresholded and masked to get the truly prominent/probable features in the image. With each corner having a strength value associated with it, they are further passed through an Adaptive Non-Max Suppression Algorithm to filter out duplicate representations of the same corner feature. All the corners with strength greater than the set threshold for a sample image are shown in Figure 1.

### B. Adaptive Non-Maximal Suppression (ANMS)

To filter out duplicates, the local maxima (using the Harris Corner strength) from each area in the image are identified and isolated. To further limit the number of features in the image and avoid too many features having a lot of overlap in their descriptor patches, each feature is associated with an R-value computed using the distance of each feature with the closest neighbor. The features having the highest R-values or are the most isolated are given preference and the 'N' most prominent/preferred features are then selected as the final feature set or the given image. The final set of corner features output by the Adaptive Non-Maximal Suppression (ANMS) Algorithm [1] on the same sample image displayed in Figure 1 are shown in Figure 2.

Fig. 1. Corner locations output by the Harris Corner Detector for a sample image

Fig. 2. Corner features remaining after using ANMS

### C. Feature Descriptors

For each key point/feature location in the image, a feature descriptor is generated by blurring, downsampling, and reshaping a patch around each feature keypoint location into a fixed-size vector.

### D. Feature Matching

A brute-force methodology is adopted to match features between images. For each feature in the first image, the sum of squared differences (SSD) between each feature in the second image is computed and ordered. If the SSD ratio between the best and second best match falls within a specified threshold, the match is said to be a good one and added to the set of matched features. All the feature matches detected by using this algorithm on two images with plenty of overlapping features/areas are shown in Figure 3.

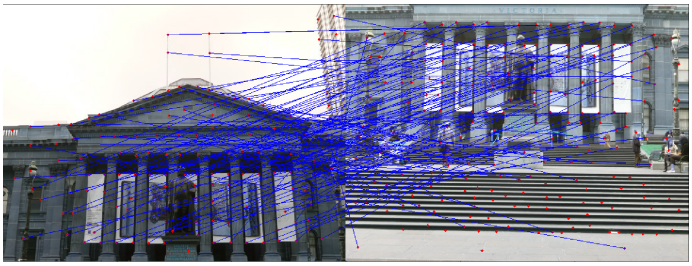

Fig. 3. All feature matches detected between two images using the SSD ratio method

### E. RANSAC

The Random Sampling Consensus (RANSAC) algorithm is used to find a reasonable set of matched features to find the best homography from the first to the second image. Over a fixed number of iterations, four matches are randomly sampled and the corresponding homography between them is computed. The norm of the distance between the actual location of the respective features and the homography predicted location is used to classify each of the good matches identified previously as an inlier or not depending on if it falls under a pre-defined threshold. The random sample with the most number of inliers is chosen to calculate the best fitting homography taking into account each of the inliers. Figure 4 displays the final chosen RANSAC matches between the same two sample images. As can be seen, only the truly desirable/correct matches are selected by using an appropriate set of hyperparameters.

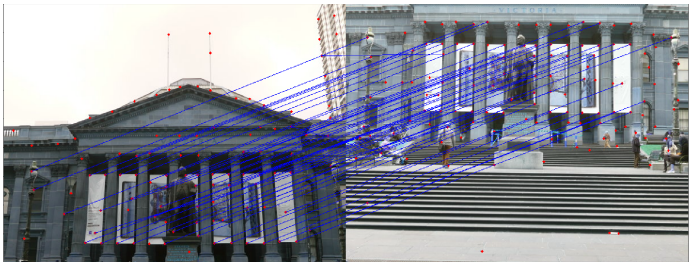

Fig. 4. Final RANSAC feature matches between the same two sample images

### F. Image Warping and Stitching

*1) Approach 1:* The first approach explored for stitching images together includes sequentially stitching the previous set of images together with the latest image till all images in the set are stitched together into a single panorama.

Using the computed best-fitting homography matrix, the first image is warped to the perspective of the second image to a size bounded by the corner locations of the transformed first image and the corner locations of the second image. The second image is then stitched onto the corresponding appropriate location of the warped/transformed first image.

To stitch together a panorama from a set of individual images, the first image is considered to be the base image. Looping over the rest of the images in the set, the base image is warped to the perspective of the other image and then stitched together. The stitched image is then taken as the base image for the next loop iteration. After stitching together all the images from the set, the final image is smoothed and resized to a desirable shape/size to obtain the final panoramic view from the set of individual images.

The result of using this stitching approach to make a panoramic view of images from the first and second training set provided as a part of this assignment is shown in Figure 5 and Figure 6 respectively.

Fig. 5. Final panoramic view of images in training set 1

Though effective, the downside of this approach is that the differences in brightness and contrast that may occur while stitching two images together may make feature matching in subsequent images harder leading to successive degradation. This method thus works best when there are many features and not much difference in brightness and contrast variation between images in the set. [2]

### G. Approach 2

The second approach involves calculating the consecutive homographies between pairs of unstitched/original images. Then using matrix multiplication between consecutive homographies, the tomography between non-consecutive images in the set can be obtained. Thus, by using this idea, one image

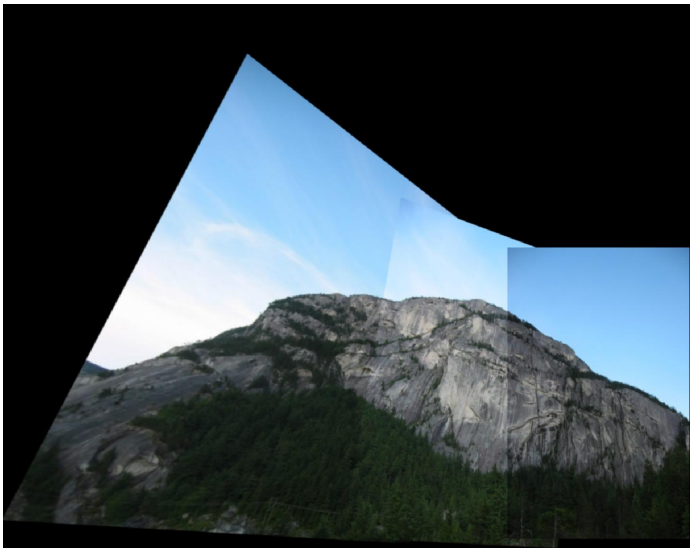

Fig. 6. Final panoramic view of images in training set 2

can consecutively be warped onto the next by calculating the homographies beforehand rather than just before warping. The only disadvantage of this approach is the fact that inaccuracies in tomography calculation propagate through the set and can cause large variations between the actual homographies, especially between images placed far apart from one another in the sequence of the image set. For this reason, leading to worse performances overall than Approach 1, Approach 1 was chosen over Approach 2.

### *H. Performance Analysis*

The final result/stitched panorama obtained by using Approach 1 on different testing sets is shown in Figure 7-Figure 10.

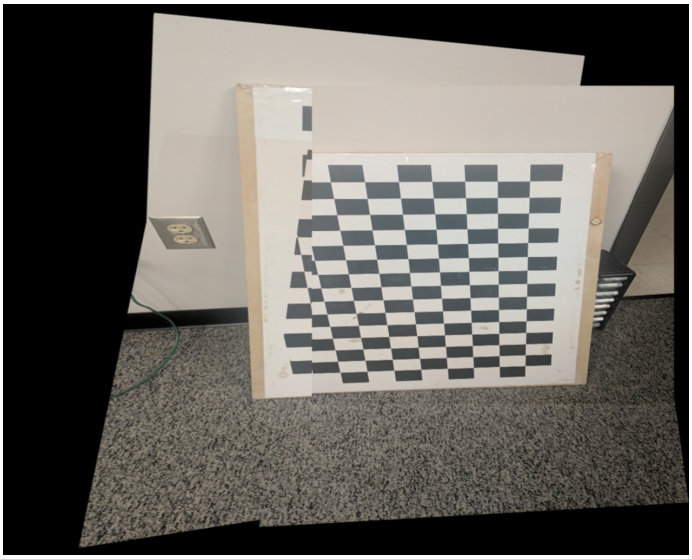

Fig. 7. Final panoramic view of images in test set 1

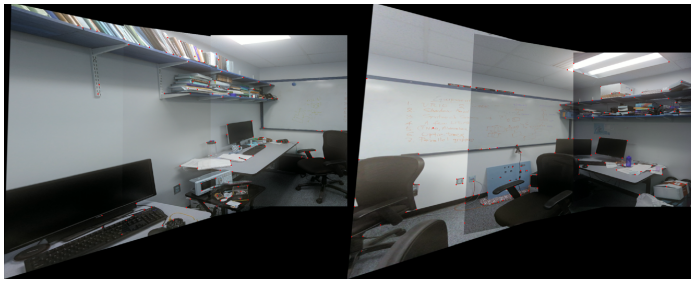

Fig. 8. Final panoramic view of the two major regions in test set 2


Fig. 9. Final panoramic view of images in test set 3

One thing to notice includes near-perfect stitching for test set 3 displayed in Fig 9. This can be attributed to an abundance of good features and not too much variation in brightness and contrast between individual images. On the other hand, some of the images in test set 4 are mostly unrelated to each other leading to no matching feature and an error in finding the panoramic view reflected by a random output generated in Fig 10. The checkerboard pattern in test set 1 has many similar-looking features leading to sub-optimal image stitching reflected in the output shown in Figure 7. Finally, with test set 2 having nine images, the scale and variations in brightness and contrast propagate through the stitching process. As a result, two major parts of the panoramic scene individually get stitched together well but when trying to combine these major regions, there is a lack of matching features. The two major regions that stitch well together individually are shown in Figure 8.

The various hyperparameters used in stitching the images from different test sets are shown in Table I. As can be seen, most of the hyperparameters used are the same except for the SSD threshold used in feature matching. The choice of an appropriate SSD threshold depends on the image set. When there is an abundance of features or when features may be very

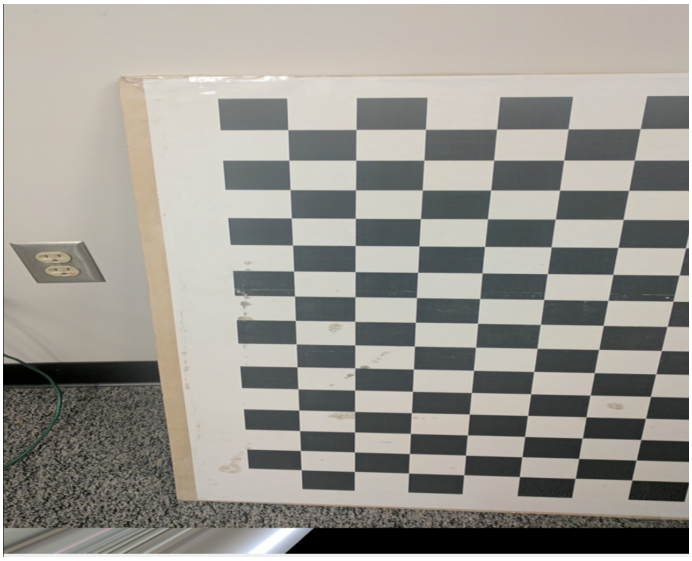

Fig. 10. Final panoramic view of images (that generates an error because of no features between unrelated images) in test set 4

similar to one another, it is better to use a low SSd threshold to enable more precise matching. However, in case of a low number of features overall, a higher SSD threshold allows for a higher number of initial matches to get generated which would get filtered to the best set of matches using RANSAC anyway.

### I. Inferences

The underlying inefficiencies in using the traditional approach mostly arise in non-optimal feature descriptors. Instead of using Harris corner features, using more effective feature descriptors like SIFT, SURF, ORB, and BRISK would give better feature locations and representations, taking into account the effect of feature rotation and illumination as well which would directly translate to a much higher performance level, especially for tricky situations like checkerboard patterns (difficult feature matching) and high number of images in a set. However, for the most part, the current implementation that uses Harris corners and ANMS works well too.

## II. Phase 2: Deep Learning Approach

Deep learning techniques are utilized to calculate the Homography matrix, which represents the transformation between two images. We applied both supervised and unsupervised learning approaches to this task. The models are trained to identify the corners of patches and predict the transformation of those patches, which are perturbed to serve as the label set. This approach is referred to as the ”4 points parameterization” method.

### A. Data Generation

This process applies to both supervised and unsupervised learning models. For our Homography-Net, we require image pairs as data input. We use 5000 images from the MSCOCO dataset [4]. During data generation, we randomly select patches from Image-1, extract four corners from the image, perturb these corners with a perturbation factor of 16, and obtain the warped patch. These patches are then concatenated into channels and fed into the Homography Network.

### B. Network

Initially, we employed the identical network architecture as described in the original HomographyNet paper [3]. This architecture bears resemblance to Oxford’s VGG Net, employing 3x3 convolutional blocks alongside BatchNorm and ReLUs (refer to Figure X). Our network comprises 8 convolutional layers, with a max pooling layer (2x2, stride 2) following every two convolutions. The convolutional layers are configured with the following number of filters per layer: 64, 64, 64, 64, 128, 128, 128, 128. Two fully connected layers follow the convolutional layers. The initial fully connected layer contains 1024 units. We incorporate dropout with a probability of 0.5 after the final convolutional layer and the first fully-connected layer. The network accepts a two-channel grayscale image sized 128x128x2 as input. The output of this network is predicted H4pt values.

### C. Supervised Approach

Within this model, we possess patch A and patch B, from which we can compute the H4pt matrix, representing the ground truth. This matrix is then compared with the predicted H4pt matrix, and the Root Mean Squared Error (RMSE) loss is determined through this comparison. The two input images, connected by a homography, are stacked channel-wise and passed through the network. The loss function employed is RMSE loss, and the Adam optimizer is utilized. A learning rate of 0.0001 is used, as decreasing it from 0.001 to 0.0001 led to a significant reduction in loss. Increasing the number of epochs resulted in a substantial reduction in loss over time.

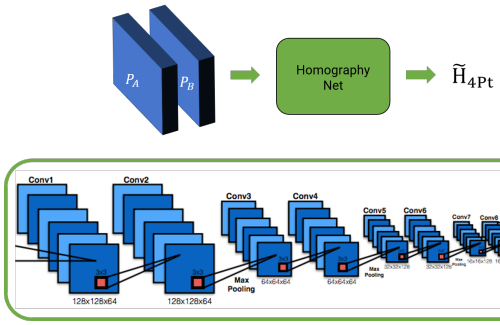


Fig. 11. 1) Supervised Learning Pipeline, 2) Architecture Diagram of Both Models

### D. Unsupervised Approach

Both supervised and unsupervised deep learning models were developed, each with its unique characteristics and methods. The supervised model, acknowledging its inherent bias, offers significant potential for improvement. The unsupervised approach utilized data as of supervised and, employing the

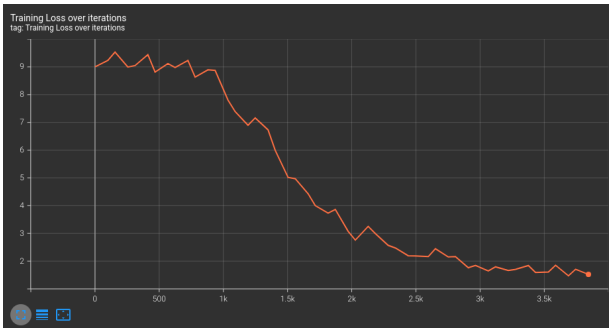


Fig. 12. Supervised Network Training Loss over Iterations. We observe it decreases and slows down at 25 epochs.

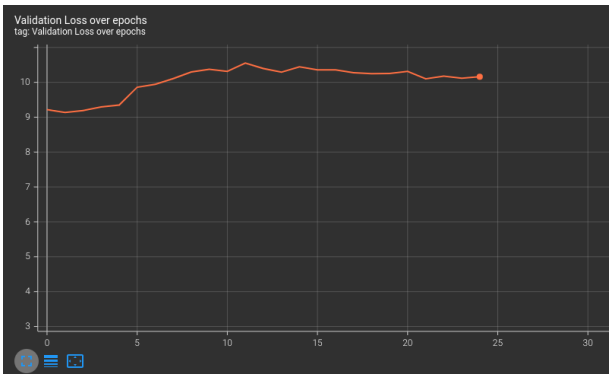


Fig. 13. Supervised Network Validation Loss over Epochs. After an initial increase, it flat lines.

same architecture as HomographyNet. Following the methodology proposed in "Unsupervised Deep Homography: A Fast and Robust Homography Estimation Model" [5], the model derives the 3x3 Homography matrix from the H4pt output, ensuring differentiability for gradient propagation through the network. Upon acquiring the 3x3 homography matrix using tensorDLT, the original image is warped through a Spatial Transformer Network (STN) layer to obtain the warped image, facilitating photometric loss calculation against ground truth input data.

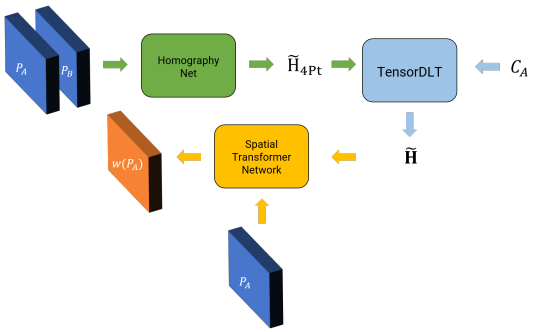


Fig. 14. Unsupervised deep learning system for homography estimation.

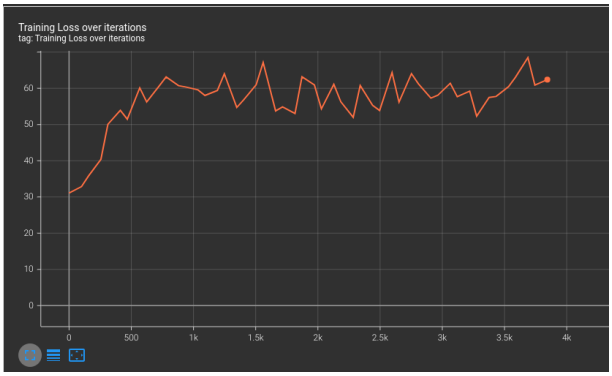


Fig. 15. Unsupervised Network Training Loss over Iterations. It does not converge even after 3.5k iterations / 25 epochs

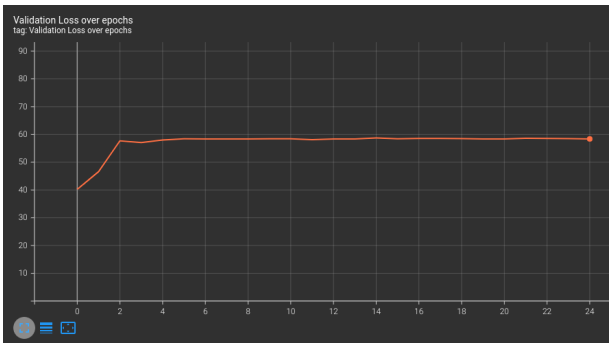


Fig. 16. Unsupervised Network Validation Loss over Epochs. After an initial increase, it flat lines.

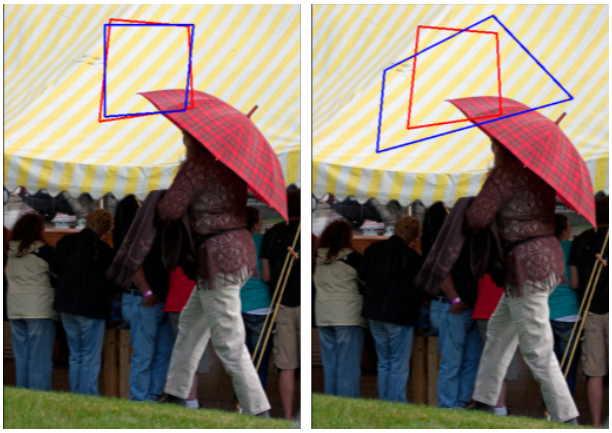

Fig. 17. Test Set Image 1 - Ground Truth Patch B (in red) and network predictions (in blue), Supervised (Left) Unsupervised (Right)

## REFERENCES

[1] Cmsc733 project 1 webpage. https://cmsc733.github.io/2022/proj/p1/.
[2] Image stitching inspiration.
[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *CoRR*, abs/1606.03798, 2016.
[4] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

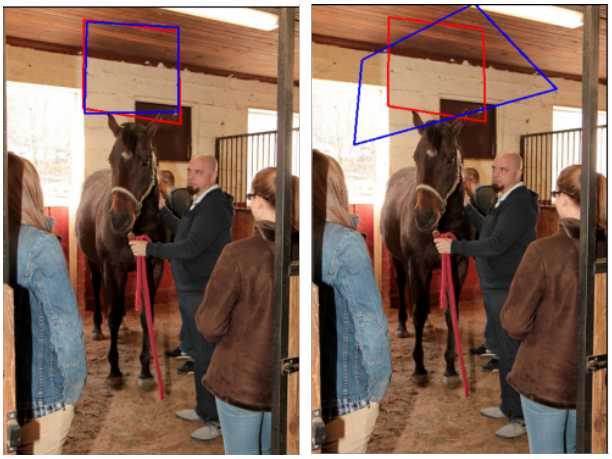

Fig. 18. Test Set Image 2 - Ground Truth Patch B (in red) and network predictions (in blue), Supervised (Left) Unsupervised (Right)

Fig. 19. Test Set Image 3 - Ground Truth Patch B (in red) and network predictions (in blue), Supervised (Top) Unsupervised (Bottom)

[5] Ty Nguyen, Steven W. Chen, Shreyas S. Shivakumar, Camillo J. Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *CoRR*, abs/1709.03966, 2017.

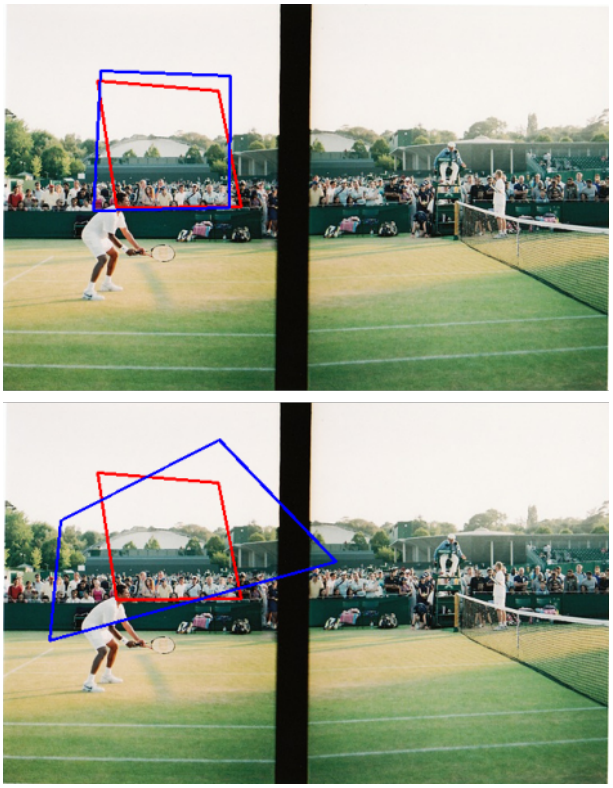

Fig. 20. Test Set Image 4 - Ground Truth Patch B (in red) and network predictions (in blue), Supervised (Top) Unsupervised (Bottom)

TABLE I
HYPERPARAMETERS USED THE TEST SETS FOR PHASE 1

| Images | Harris K-Value | Harris Max-Ratio | SSD Thresh | RANSAC Epochs |
|---|---|---|---|---|
| Test Set 1 | 0.04 | 0.05 | 0.75 | 10000 |
| Test Set 2 | 0.04 | 0.05 | 3.0 | 10000 |
| Test Set 3 | 0.04 | 0.05 | 4.5 | 10000 |
| Test Set 4 | 0.04 | 0.05 | 10.0 | 10000 |

TABLE II
HYPERPARAMETERS FOR NEURAL NETWORK TRAINING

| Hyper-parameter | Value |
|---|---|
| Optimizer | Adam |
| Learning rate | 0.0001 |
| Mini Batch Size | 32 |
| No. of Epochs | 25 |
| Dropout | 0.5 |

TABLE III
RMSE PIXEL ERRORS FOR PREDICTIONS IN TEST SET SHOWN

| Image | Supervised | Unsupervised |
|---|---|---|
| 1 | 7.4190 | 52.6489 |
| 2 | 7.1347 | 54.9761 |
| 3 | 12.3078 | 50.2396 |
| 4 | 11.9091 | 52.4801 |