# CMSC 733 Project I: MyAutoPano
# Using 2 late days

Naitri Rajyaguru
email: nrajyagu@umd.edu

Angelos Mavrogiannis
email: angelosm@umd.edu

***Abstract*—Phase I consists of traditional approach used for Panorama Stitching. It uses the concepts of feature detection like corners, feature correspondence, applying RANSAC and estimating Homography between two images for stitching. Phase II consists of a set of two deep learning approaches to estimate the homography between a pair of images. It combines the components of the classical approach (corner detection, ANMS, feature extraction, feature matching, RANSAC) in an end-to-end manner. We begin by generating a synthetic dataset of images with known homographies and use them as ground truth for two deep learning-based approaches. One of them is built upon a VGG Convolutional Neural Network architecture that minimizes the difference between corresponding points and the other expands this approach by adding a direct linear transform (DLT) layer and a spatial transformer layer, minimizing a pixel-wise photometric loss function.**

## I. Phase I: Traditional Approach

This approach takes in depth steps into consideration for Automated Image Stitching for more images. The traditional approach is divided into six steps:
1) Corner Detector
2) Adaptive Non-Maximal Suppression
3) Feature Descriptor
4) Feature Matching
5) RANSAC
6) Image Stitching/Blending

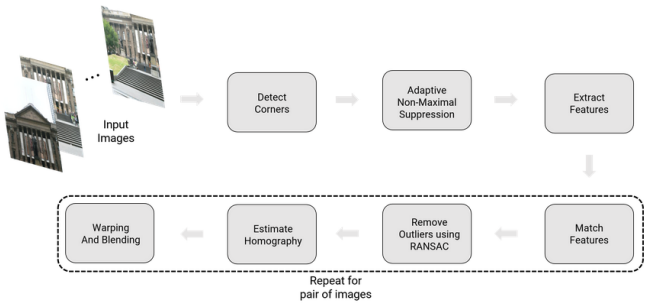


Fig. 1: Panorama Stitching Pipeline Overview

### A. Corner Detection

Corners are considered to be the most unique features for extraction of image information. Usually, corners are detected using corner score also referred as autocorrelation. In this implementation, Shi-Tomasi Corner Detector is used as it is more efficient than Harris Corner detector. Though formulation for both is similar but R score(cornerness) is different.

$$R = \min(\lambda_1, \lambda_2)$$

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

The corners detected in the images are huge in number and not all contribute to better results. The corners are supposed to be more spatially spaced in an image with only high corner score corners are kept for further calculation. However, Shi-Tomasi detector function available in OpenCV filters the corners. The following algorithm has been followed to achieve ANMS:

### C. Feature Descriptor

For each refined corner after ANMS, a descriptor for the same is required which is achieved by taking a patch of image surrounding each corner point. This patch is blurred and sampled into $8 \times 8$ dimension which is flattened into $64 \times 1$ vector. Normalization is taken by subtracting each value with the mean and dividing it by standard deviation.

### D. Feature Matching

For feature matching, each corner point from first image, we compute sum of square differences between all points in second image. A Lowe's ratio test is being taken over lowest first and second SSD from the pair of corners. The output for the same is shown below:

**Input** : Corner score Image ($C_{img}$ obtained using `cornermetric`), $N_{best}$ (Number of best corners needed)
**Output:** $(x_i, y_i)$ for $i = 1 : N_{best}$

Find all local maxima using `imregionalmax` on $C_{img}$;
Find $(x, y)$ co-ordinates of all local maxima;
($(x, y)$ for a local maxima are inverted row and column indices i.e., If we have local maxima at $[i, j]$ then $x = j$ and $y = i$ for that local maxima);

Initialize $r_i = \infty$ for $i = [1 : N_{strong}]$

**for** $i = [1 : N_{strong}]$ **do**
  **for** $j = [1 : N_{strong}]$ **do**
    **if** $(C_{img}(y_j, x_j) > C_{img}(y_i, x_i))$ **then**
      ED $= (x_j - x_i)^2 + (y_j - y_i)^2$
    **end**
    **if** $ED < r_i$ **then**
      $r_i =$ ED
    **end**
  **end**
**end**

Sort $r_i$ in descending order and pick top $N_{best}$ points

Fig. 3: ANMS Algorithm

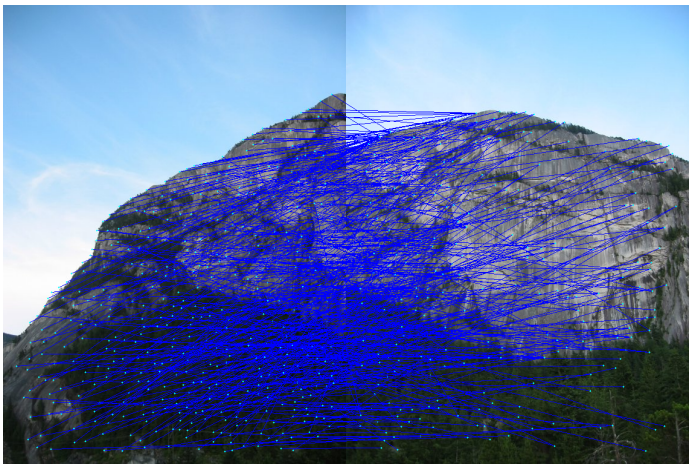

Fig. 5: Feature Matching for set 2

### E. RANSAC and Estimating Homography

We need Homography H matrix to recover projective transformation between set of images. H is $3 \times 3$ matrix with 8 degrees of freedom which means to solve for pairs of correspoinding points we need 4 pairs of points. The result of this will be an overdetermined system and hence many outliers will be present. RANSAC is used for outliers removal and keeping only set of points where number of inliers is maximum.

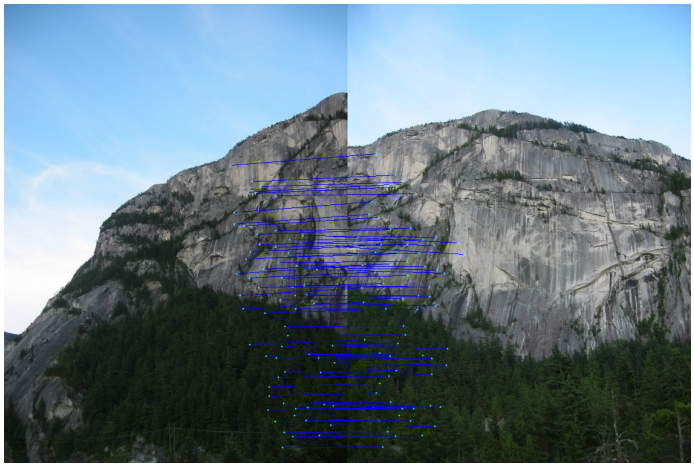

Fig. 6: Feature Matching for set 2

### F. Image Stitching

As we have homography matrix with us, we can wrap one image on another using the same. We use homography matrix to transform all points in one image to the coordinate system of another.

Fig. 7: Panorama for 2 images in set 2

## II. Phase II: Deep Learning Approach

### A. Data Generation

The first deep learning approach [1] that we explore requires a set of ground truth homographies, which is also required as an input to the second approach [2]. Due to the lack of such a dataset, we manually generate a synthetic dataset of

Fig. 8: Panorama of CustomSet1

Fig. 9: Panorama of Set2

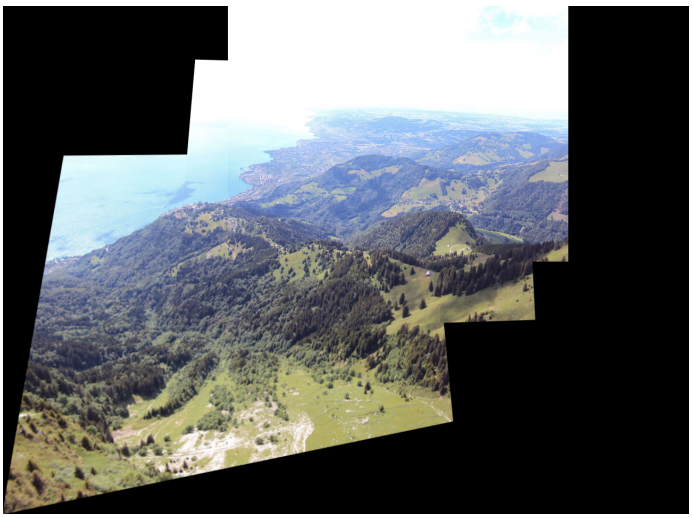

Fig. 10: Panorama of CustomSet2

pairs of images from the MS COCO dataset [3] and compute the homography between them. First, we resize all the given images to $320 \times 240$ since our Convolutional Neural Network (CNN) expects images of the same size. Next, we choose a random point at a position $p$ of an image and we extract a square patch $I_p$ of size $128 \times 128$. We randomly perturb the four corners of the patch within the range $[-\rho, \rho]$ with $\rho = 32$, following the implementation of the paper, thereby defining a homography $H^{AB}$. We compute the inverse homography $H_{BA}$ and apply it to the initial image, creating a new warped image

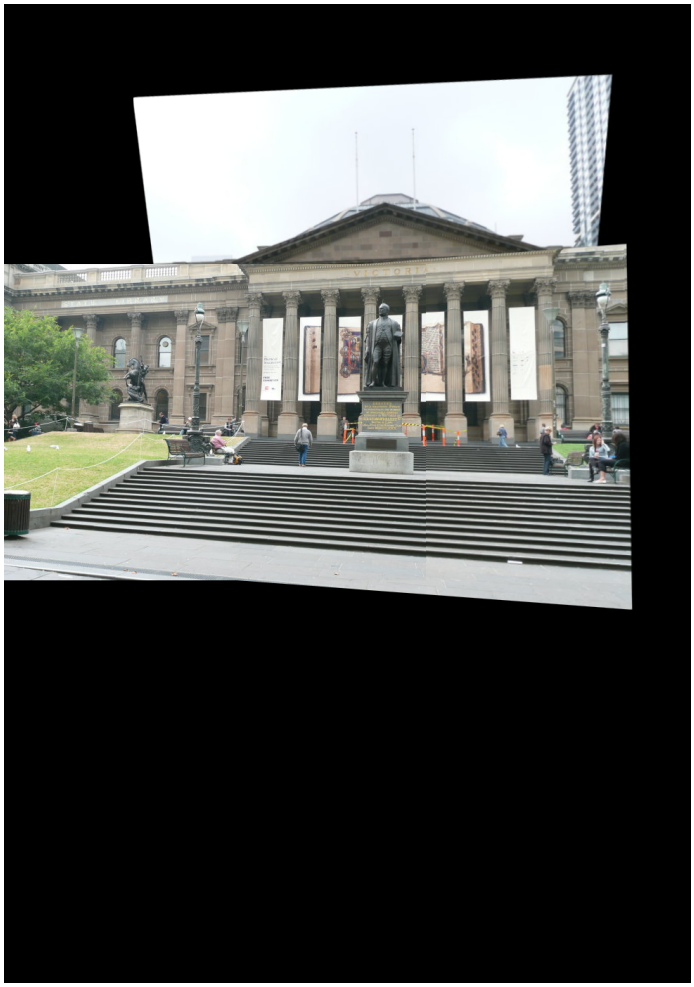

Fig. 11: Panorama of Set1

$I'$. Finally, we crop a square patch of the same size ($128 \times 128$) from the new image.

### *B. Supervised Approach*

The supervised approach [1] consists of a CNN built following Oxford's VGG Net architecture [4]. It receives an input of two stacked $128 \times 128$ square patches derived from the Data Generation part, passes them from eight convolutional layers with each of them followed by a relu activation [5] while applying a max pooling after every two of these layers. It finally results in two fully-connected layers, with the last one corresponding to the estimated homography. We also apply batch normalization [6] between every layer and its activation function, except for the last dense layer, and dropout [7] with a probability of $0.5$ between the two fully-connected layers. The full architecture of the network is shown in Figure 12. Each of the four first convolutional layers has $64$ $3 \times 3$ filters, each of the next four has $128$ $3 \times 3$ filters, the max pooling layers consists of $2 \times 2$ kernels with a stride of 2, and the final fully-connected layers have $1024$ and $8$ layers each, respectively. The ground truth corresponds to a $4 \times 2$ matrix $H_{4point}$ that

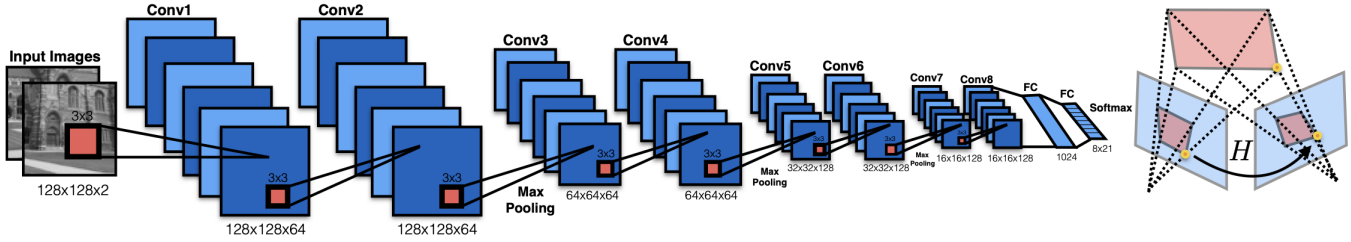


Fig. 12: Architecture of the Supervised Approach, as seen in [1].

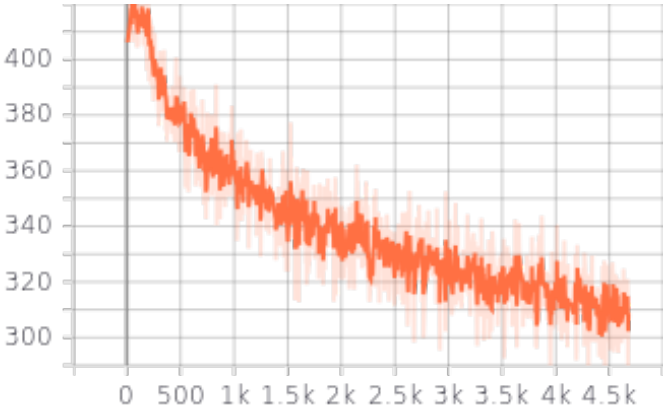


Fig. 13: Training loss of the supervised model for 30 epochs with a batch size of 64 and 10000 training samples, leading to 4680 total iterations.

includes the differences between the corner points of the initial image and the warped image, as computed in Section II-A:

$$H_{4point} = \begin{bmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{bmatrix} \tag{1}$$

We use the Adam optimizer [8] to minimize the Euclidean $L_2$ difference between the estimated and the ground truth values of the homography, and we train for 30 epochs with a batch size of 64, resulting to 4680 iterations. We are given a training set of 5000 images but augment it to a set of 10000 images, for better performance. A graph showing the training loss from Tensorboard can be seen in Figure 13. We show sample output from the validation set in Figure 14 and from the test set in Figure 15. The ground truth is shown in red and the corresponding points based on the estimated homography are shown in yellow.

### C. Unsupervised Approach

The unsupervised approach [2] is based on the supervised approach in Section II-B with the addition of some new layers. It receives as input the stacked patches from Section II-B which correspond to the initial images $I^A$ and $I^B$, the four corners in $I^A$, denoted as $C^A_{4pt}$, and the image $I_A$, which is needed for warping. First, a Tensor Direct Linear Transform (Tensor

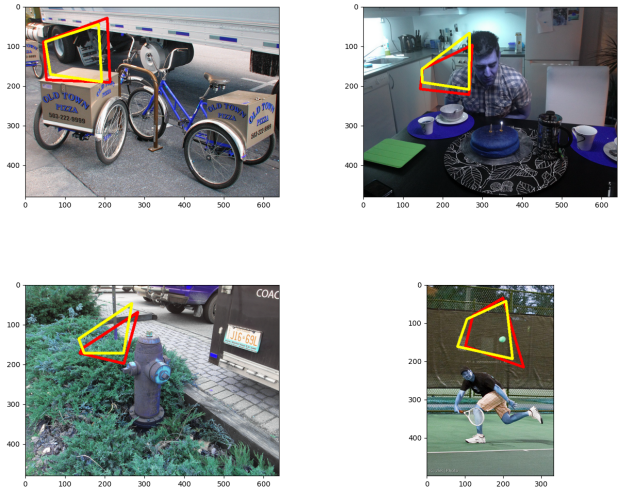

Fig. 14: Sample output from testing the supervised network on the validation set.

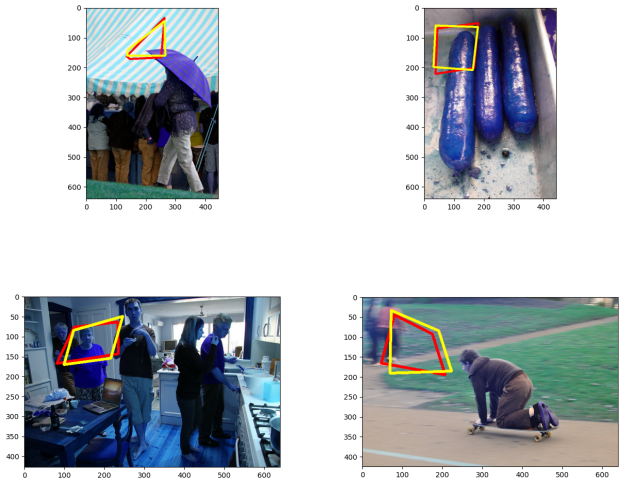

Fig. 15: Sample output from testing the supervised network on the test set.

DLT) layer is implemented to transform the 4-point homography parameterization into a $3 \times 3$ parameterization matrix $\tilde{H}$. This layer applies the DLT algorithm [9] to tensors, keeping them differentiable and hence applicable for backpropagation. The matrix $\tilde{H}$ is then passed from a spatial transformation layer [10] and is applied to the pixel coordinates $x_i$ of the initial input image to get warped coordinates $\mathcal{H}(x_i)$. Finally, the network returns a homography parameterization $\tilde{H}_{4pt}$ that minimizes the $L_1$ pixel-wise photometric loss:

$$L_{PW} = \frac{1}{|x_i|} \sum_{x_i} |I^A(\mathcal{H}(x_i)) - I^B(x_i)| \tag{2}$$

The architecture can be seen in Figure 16. Again we use the Adam optimizer and a learning rate of $\eta = 0.0001$. We show sample output from the validation set in Figure 24 and from the test set in Figure 19. The ground truth is shown in red and the corresponding points based on the estimated homography are shown in yellow.

## III. Conclusion

For Phase I, Blending of images is difficult because of while balance present in the images. Panorama stitching is becomes tricky when number of images increases and a good way to combine can be dividing images into a group and combining after stitching each group of images.The problem just mentioned was occurred during Panorama stitching of Set3 images.For Phase II, the supervised approach shows promising results and is easier to implement and train than the unsupervised approach. However, it requires manual ground truth generation and a large training dataset. In theory, the unsupervised approach should produce better results, but we were unable to achieve that during our experiments, and most of the times the supervised approach gives superior results. This could also be because of the fact that we used data augmentation to improve the training and performance of the supervised network. What is clear though is that the unsupervised approach has a much faster inference speed than the supervised, and is more robust due to its resistance to illumination variation.

## References


[1] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation, 2016.

[2] Ty Nguyen, Steven W. Chen, Shreyas S. Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, 3(3):2346–2353, 2018.

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[5] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[7] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.


## Appendix

Output for Phase1 set1
Output for Phase1 set2
Output for Phase1 set3
Output for Phase1 CustomSet1
Output for Phase1 CustomSet2

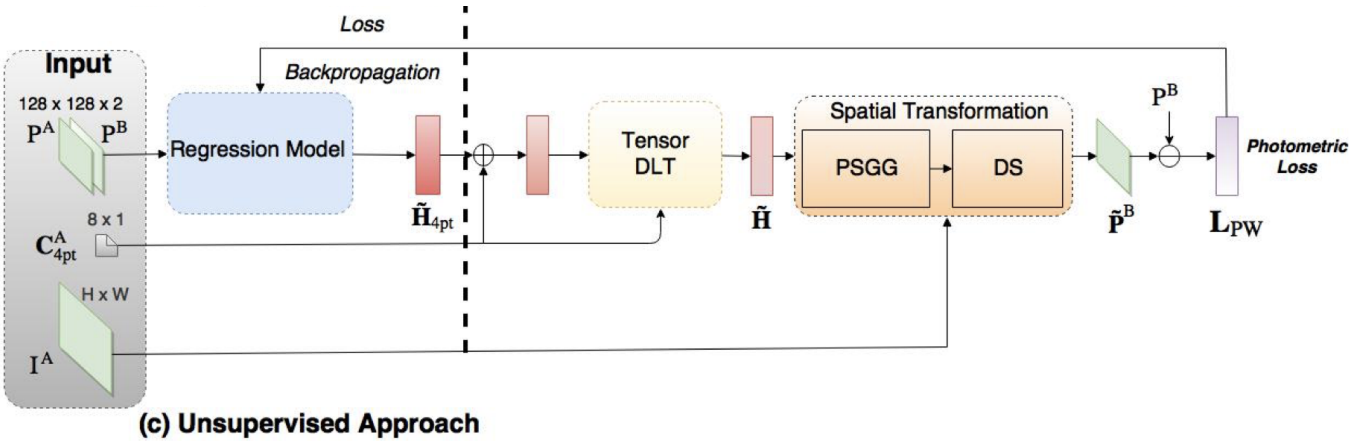


Fig. 16: Architecture of the Unsupervised Approach, as seen in [2].

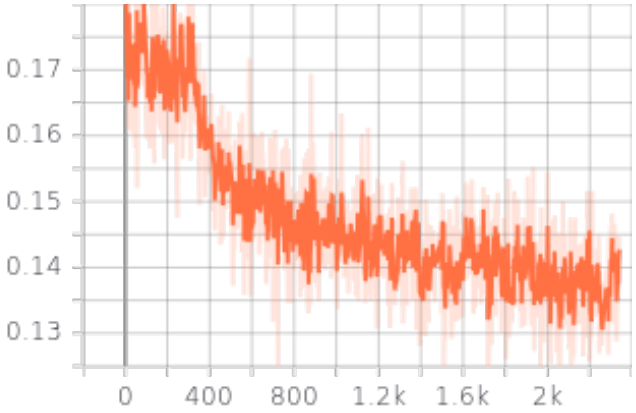


Fig. 17: Training loss of the supervised model for 30 epochs with a batch size of 64 and 5000 training samples, leading to 2340 total iterations.

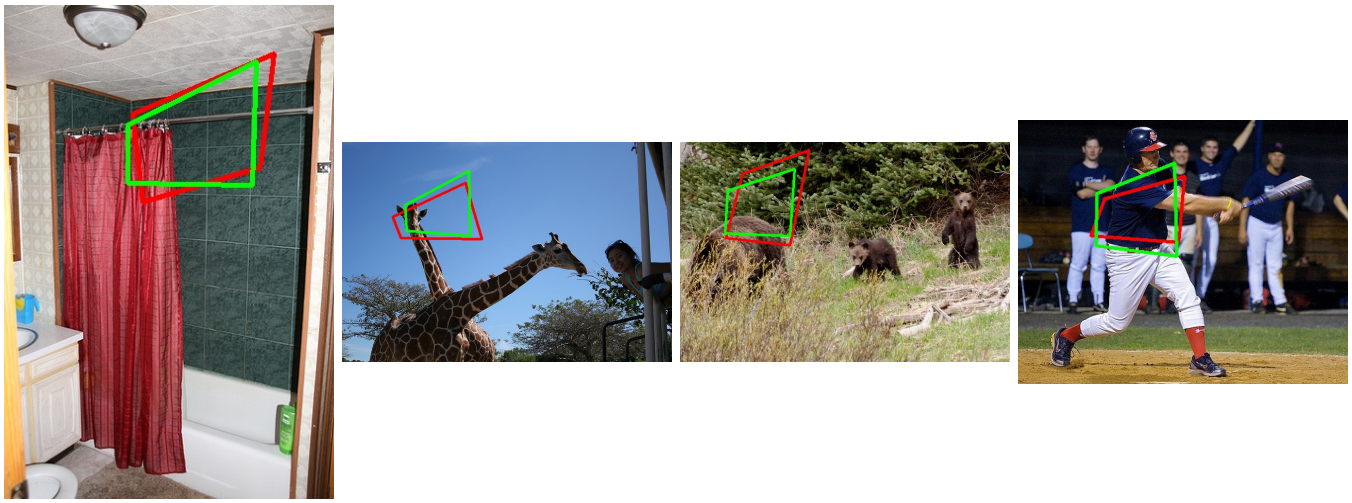

Fig. 18: Sample output from testing the unsupervised network on the validation set.

Fig. 19: Sample output from testing the unsupervised network on the test set.

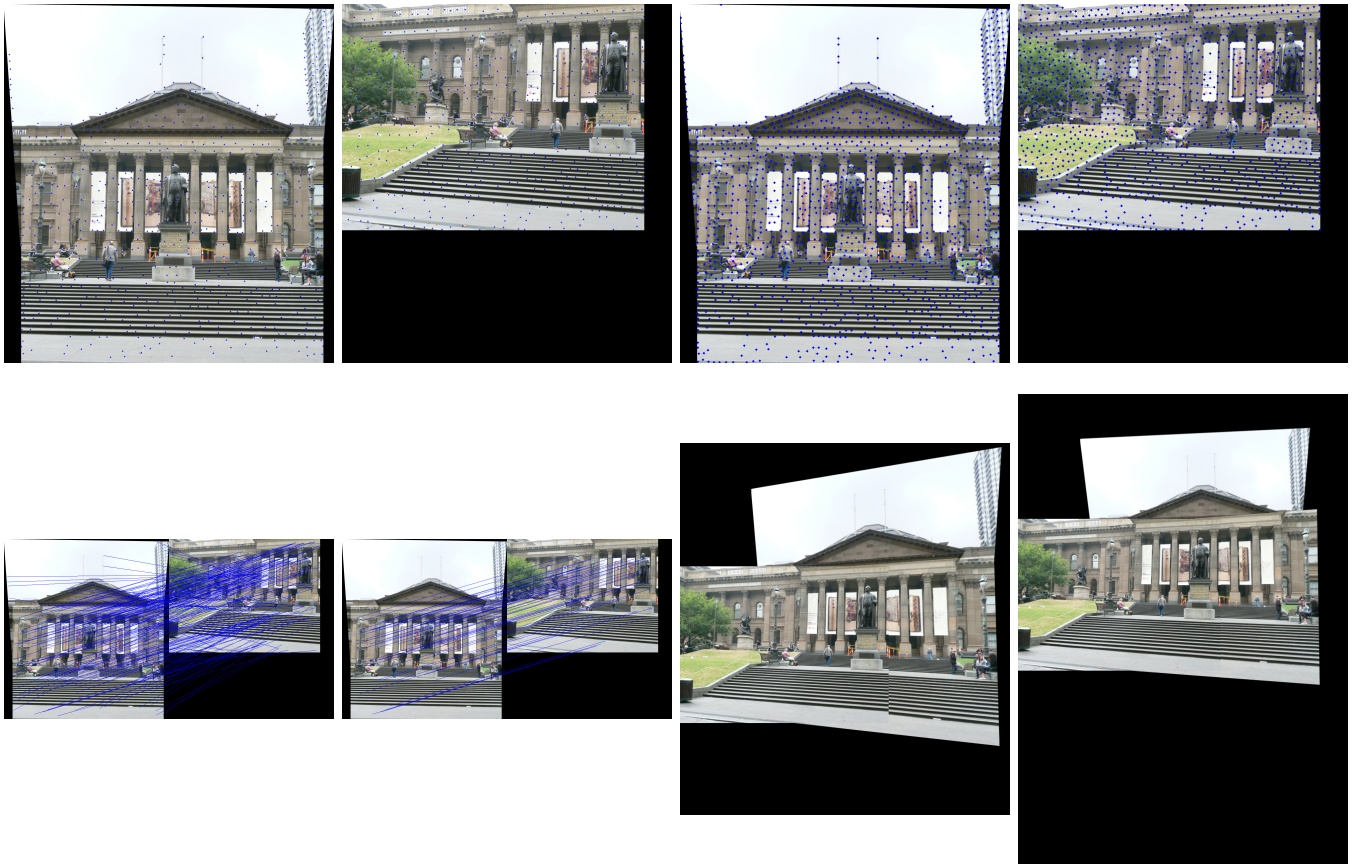

Fig. 20: Output of all steps for Set 1

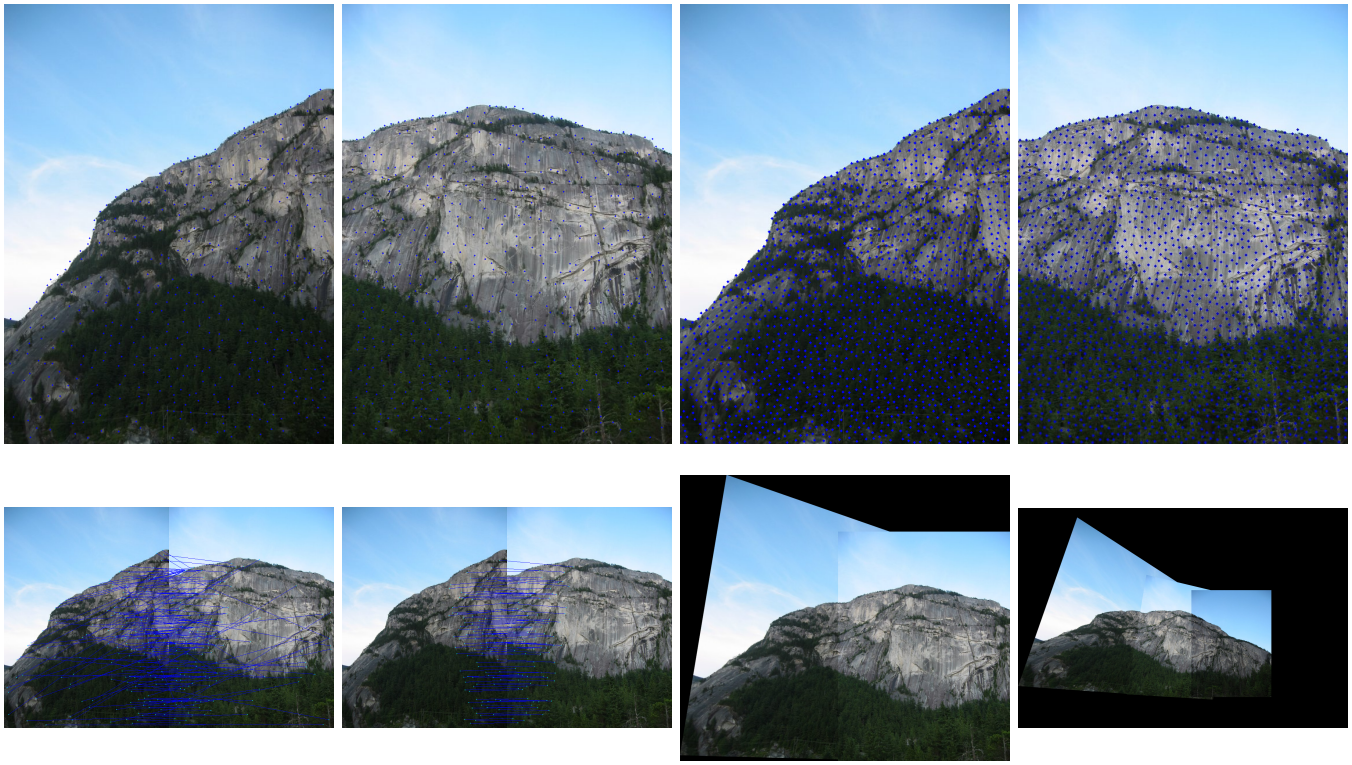

Fig. 21: Output of all steps for Set 2

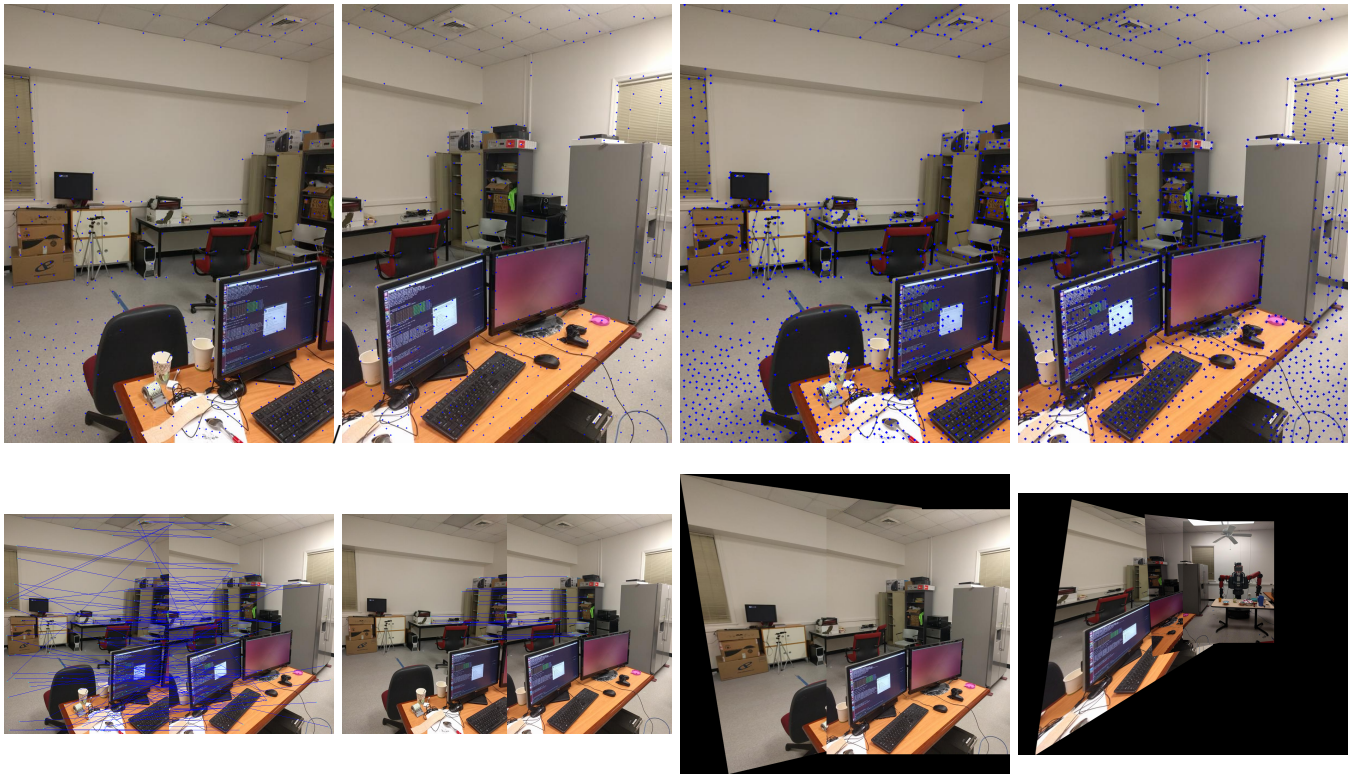

Fig. 22: Output of all steps for Set 3

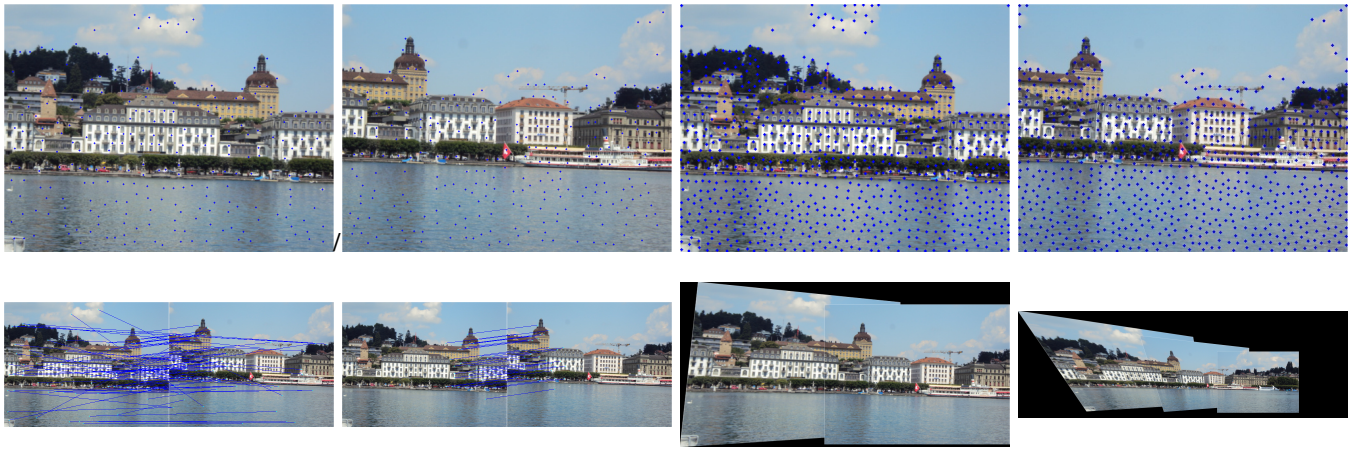

Fig. 23: Output of all steps for CustomSet1

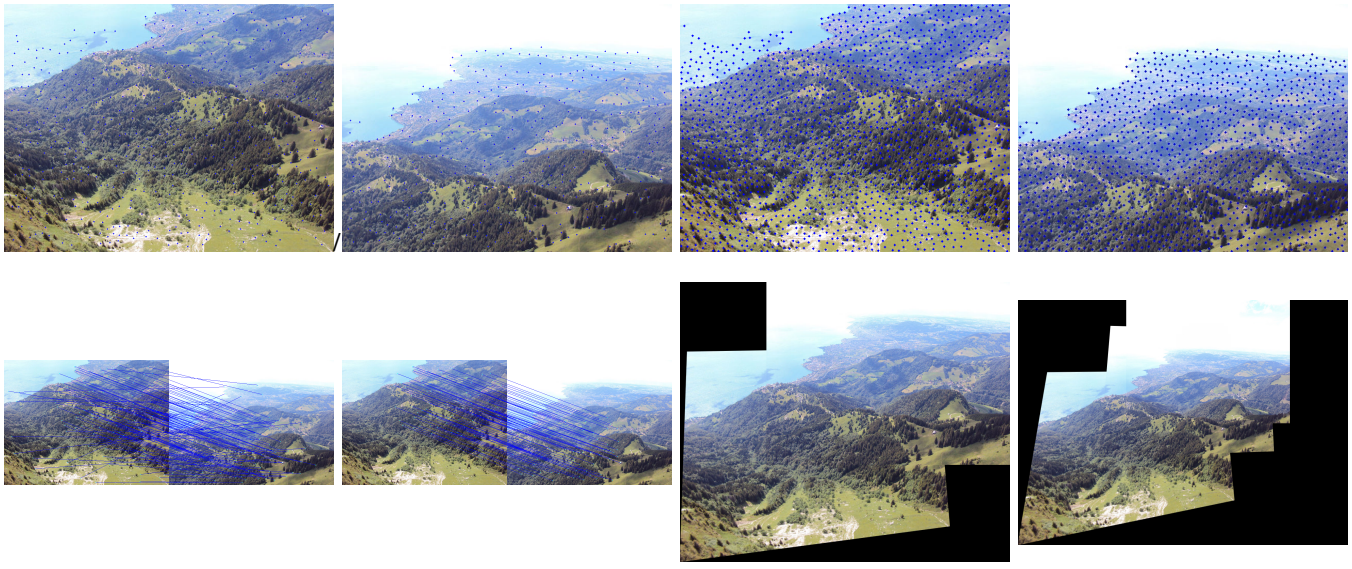

Fig. 24: Output of all steps for CustomSet2