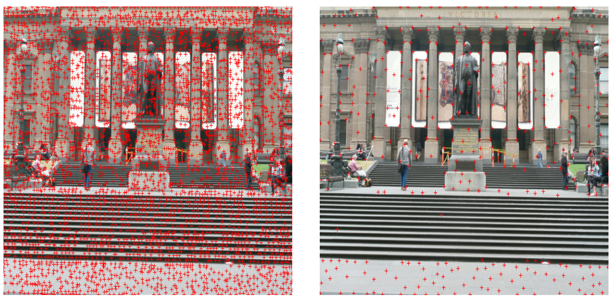
Computer Processing of Pictorial Information Project 1

Huai-Jen Liang
Department of Electrical and Computer Engineering
University of Maryland
College Park, Maryland 20740
Email: r81810g.umd@gmail.com



(a) Before applying ANMS (b) After applying ANMS

Fig. 1: Corner detection.

I. Introduction

This project contains an end-to-end pipeline to do image panorama stitching. To form a harder problem, we handled a set of unordered images. Before stitching a set of images, we need to find the order of images to stitch. The results are presented in Fig.7.

II. Method

In this section, we describe the pipeline of stitching images. At first, we assume we are going to stitch a pair of images. Then, we describe the way we handled a set of unordered images.

A. Feature Extraction

In order to determine the matches between two images, we need to find features first. In our method, we used Harris corner detector[1] to extract the corner inside an image as features. After extracting corners, we applied Adaptive Non-maximal Suppression(ANMS)[2] to find corners distributed much evenly over the image. The difference between before and after applying ANMS with taking the max 400 responses are shown in Fig.1. To represent the feature of each corner, we simply extracted a patch of size 40×40 centered around the corner, applied gaussian filter, and downsampled to 8×8. This 64-dimensions feature descriptor is used to match keypoints between two stitching images.

Fig. 2: Features Matching.

B. Robust Homography Estimation

After extracting features, we matched each feature in image 1 to a feature in image 2. This is done by picking a keypoint in image 1 and computing $L_2 - norm$ between all points in image 2. We accepted or rejected the match based on Eq.1

$$Accept\ if\ \frac{D_{best}}{D_{second}} < 0.1 \quad (1)$$

where D_{best} is the lowest distance. The matching result is shown in Fig.2. Picking four correspondences between two images, we can estimate a homography. There could be many homography since there are many sets of four points. We applied RANSAC[3] to compute a robust homography.

C. Blending two images

With homography, we can transform image 2 and combine them together. In order to combine two images without strange artifacts in overlapping region, we have to blend two images. We applied Maximum Blending, Average Blending, Feathering, and Laplacian Pyramid Blending. The pixel values in overlapping region for each method are described as follows:

1) Maximum Blending: Taking the maximum value.
2) Average Blending:
$$P(x, y) = \frac{P_1(x, y) + P_2(x, y)}{2} \quad (2)$$
3) Feathering: The weights for each pixel are expressd in Fig.3.
4) Laplacian Pyramid Blending: We first created $weights$ as in Feathering. Second, we built Gaussian Pyramid for $image$ 1, $image$ 2, and $weights$. Third, we created

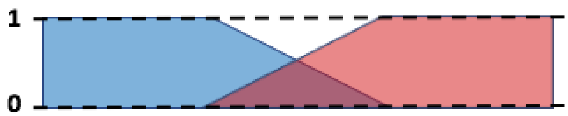


Fig. 3: Weights for Feathering.

Laplacian Pyramid for $image$ 1 and $image$ 2. Last, we combined these two pyramid with $weights$ and collapsed whole pyramid to get the output image.

The comparison between each method is shown in Fig.4. We can easily observe that there are artifacts in Fig.4a and Fig.4b.

D. Cylindrical Projection

When we tried to stitch a lot of images with translation, a simple projective transformation will give bad results if we assume all images lie on a plane. There will be lots of distortion problems at edges. Therefore, it will be useful if we first project images on two a cylindrical. This can be done be doing the following projection:

$$
\begin{aligned}
x' &= f \cdot tan\left(\frac{x - x_c}{f}\right) + x_c \\
y' &= \left(\frac{y - y_c}{cos\left(\frac{x - x_c}{f}\right)}\right) + y_c
\end{aligned}
\tag{3}
$$

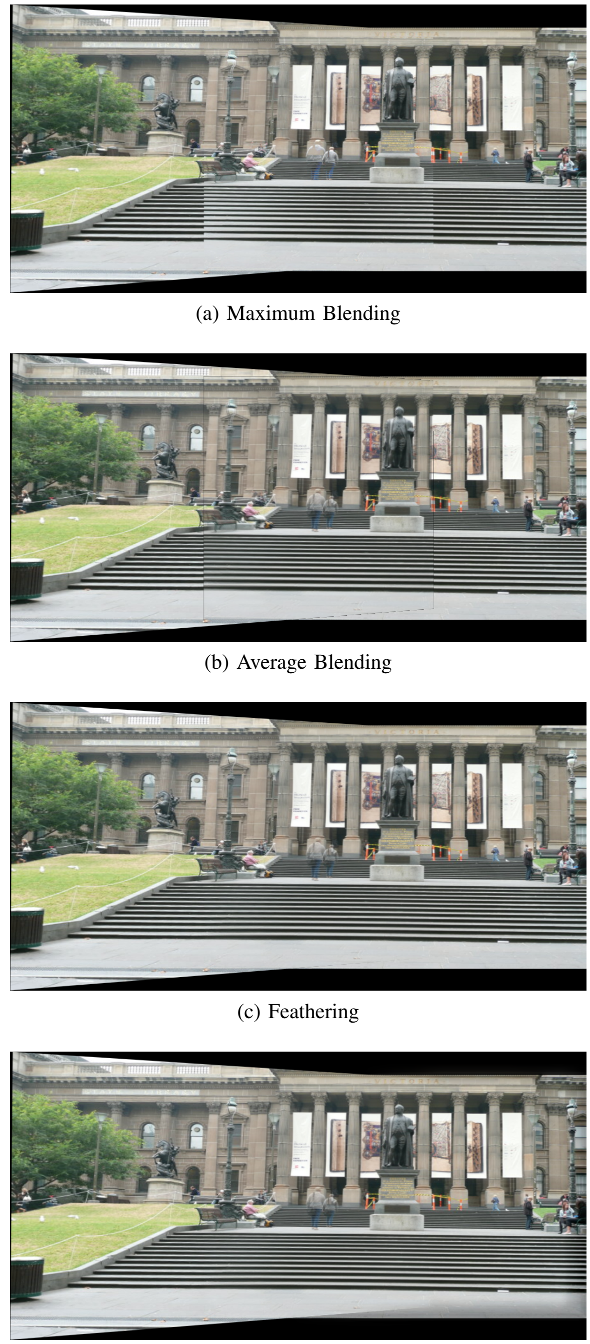
where (x', y') is in original image co-ordinates and (x, y) is in cylindrical co-ordinates. It should be noted that, in this project, we first extract features and match them in original image co-ordinates, then project them to cylindrical co-ordinates. If we do feature extraction and matching in cylindrical co-ordinates, we will get bad features and matching. The compare of stitching images in original image co-ordinates and cylindrical co-ordinates is shown in Fig.6. We only show the result of stitching the first three images in Fig.6c because the edge distortion is too large and software will break down if we try to stitch the rest.

E. Automatic Panorama Recognition (Extra Credit)

We developed an automatic panorama recognition algorithm that will cluster images into different panoramas (if there are multiple panorama set) and order them. We first check each pair of images whether they are connected or not by the following condition:

$$
L(I_a, I_b) = \begin{cases} 1, & \text{if } inliers'\ ratio > 0.3 \\ 0, & \text{otherwise} \end{cases} \tag{4}
$$

where L is a $n_{images} \times n_{images}$ matrix. After having L, we start $BreathFirstSearch$ from a random image and cluster images which can be reached from this root into the same panorama.



(a) Maximum Blending

(b) Average Blending

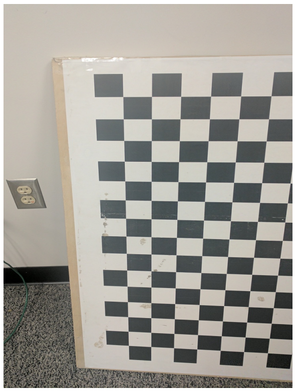
(c) Feathering

(d) Laplacian Pyramid Blending

Fig. 4: Comparison of different blending.

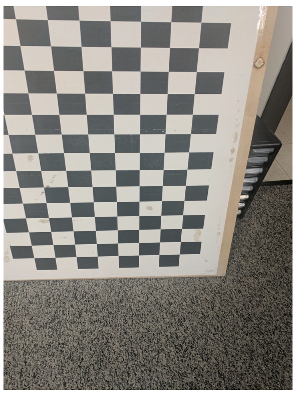
III. Results

For four test sets, we show the results in Fig.7. We can not get a good result for test set 1, since feature matching is not working. We can observe that in Fig.5, there are many similar corners in each images, so matching procedure is confused and the inliers' ratio will be very low. For test set 2, Fig.7f, the images contain whiteboard will give a bad homography, because the feature on the whiteboard is not discriminant and we cannot find good feature matching. For test set 4, Fig.7h,

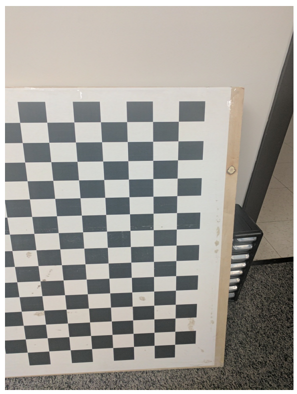


(a) Original co-ordinates

(b) Cylindrical co-ordinates



(c) Original co-ordinates



(d) Cylindrical co-ordinates

Fig. 5: Effect of cylindrical projection.

there are two images inside the folder not belong to any panorama and our algorithm can automatically reject these two images. With these experiments, we can observe that the edge is sometime a good feature to align two images. However, in this project, we only consider corners as our features, we lost many information. Even we find some corners on an edge, matching them will be a disaster.

IV. Acknowledgement

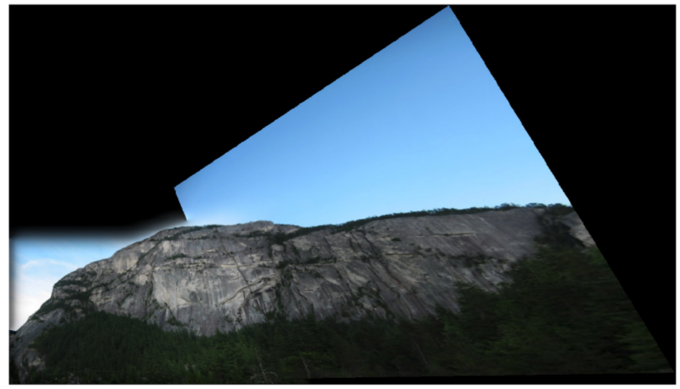
Special thanks Wei-An Lin for discussion.

References

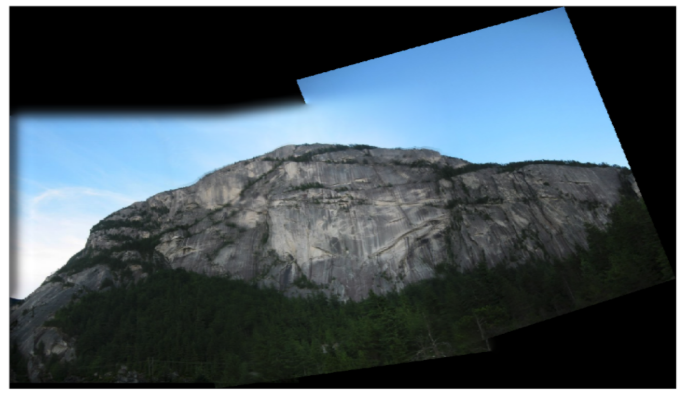
[1] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, 1988.

[2] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005.

[3] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, June 1981.

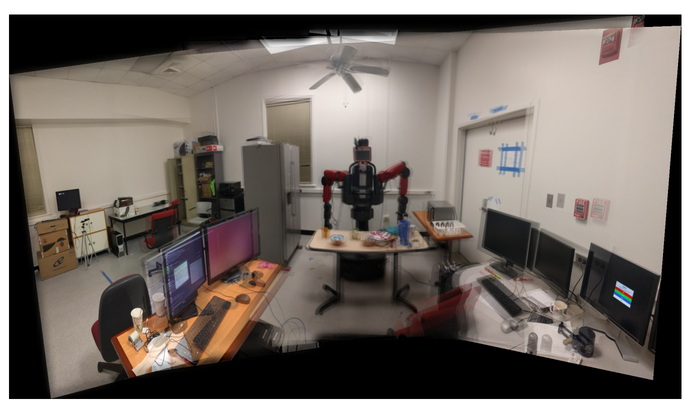


(a) Original co-ordinates



(b) Cylindrical co-ordinates

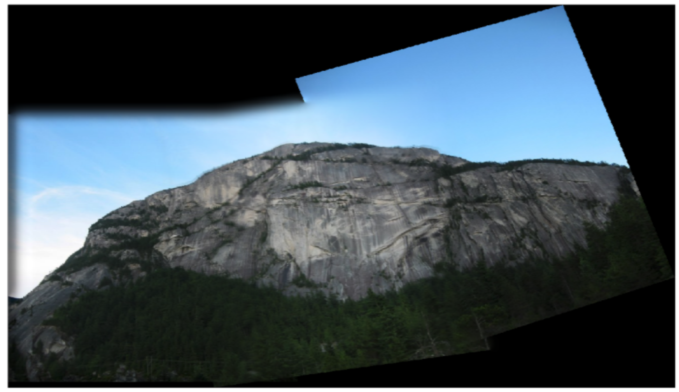
(c) Original co-ordinates



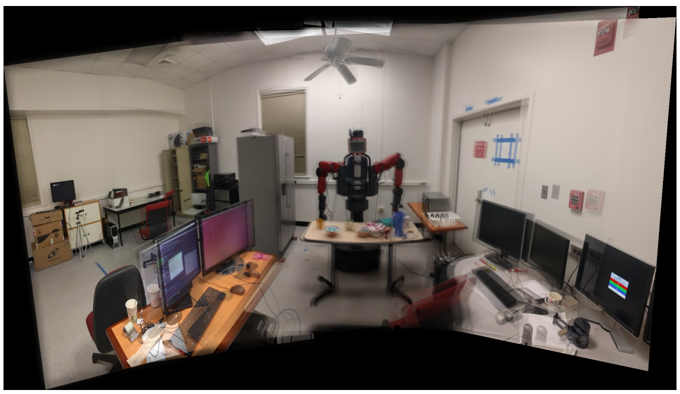
(d) Cylindrical co-ordinates

Fig. 6: Effect of cylindrical projection.

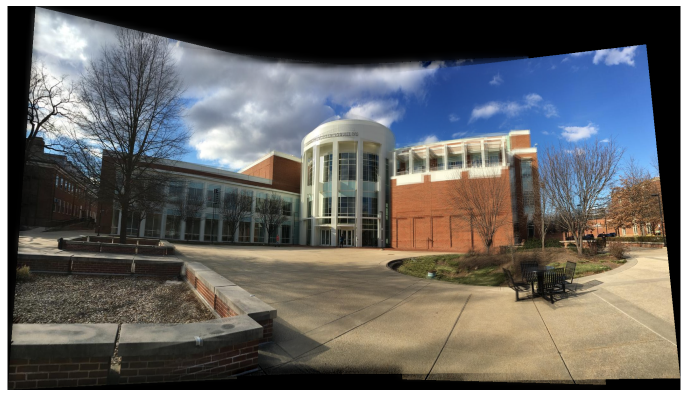
(a) Set 1



(b) Set 2

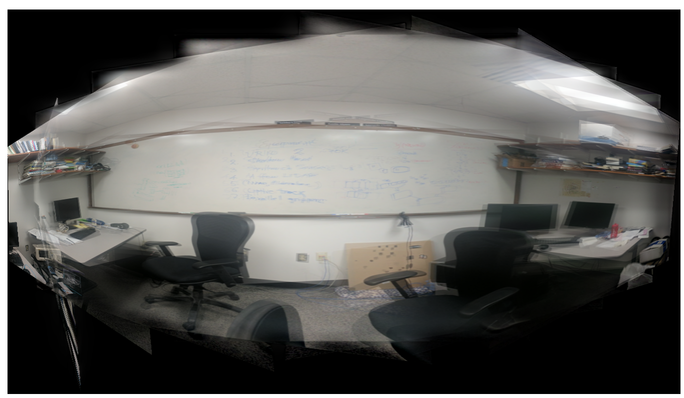


(c) Set 3



(d) Custom Set 1

(e) Custom Set 2



(f) Test Set 2

(g) Test Set 3

(h) Test Set 4

Fig. 7: Stitching Results.