Introduction and Implementation

For this assignment, we implemented an algorithm to find the infinite homography associated with each image in a dataset and eventually stitch them together to form a panoramic image. A simple outline of the algorithm is given below:

- Step 1: Choose the middle image of the dataset as the reference image
- Step 2: Find the SIFT features for every image with respect to the reference image
- Step 3: Apply the feature matching algorithm using Euclidean distance as the metric
- Step 4: Using the features evaluated above find the homography associated with each image with respect to the reference image
- Step 5: Create a new panoramic image and use alpha blending with a binary mask to stitch the images

Step 1: This one is simple but an important step of the algorithm since all homographies will be evaluated with respect to this image. If the first image is chosen as the reference image then I noticed in my experiments that the subsequent images further from the reference image became distorted and the panorama was highly inaccurate. So the optimal solution would actually be the middle image since the relative homographies that would be evaluated would be less prone to distortion.

Step 2: For the sift feature points I used the implementation provided by VL Feat [10]. The SIFT features were found using the peak threshold as 0. The *peak threshold* filters peaks of the DoG scale space that are too small (in absolute value). The higher this value, the lesser features we will obtain in our image. I set this value to the default 0 since that gave me the number of extracted features ~ 1000. Since the subsequent part of the algorithm relied on matching the descriptors, most of them were filtered out during the process making this is reasonable value for the threshold.

Step 3: For the feature matching algorithm, the approach used was the same as the on suggested by David Lowe. **Vl_ubcmatch** uses the specified threshold THRESH. A descriptor D1 is matched to a descriptor D2 only if the distance d(D1,D2) multiplied by THRESH is not greater than the distance of D1 to all other descriptors. The default value of THRESH is 1.5 which was giving matched for images in the order of 300 which were sufficient for an estimation of the homography in the next step.

Step 4: In this part, I used the ransacfithomography function provided by Peter Kovesi [6]. The idea was to limit the contribution of outliers to the final homography by select points that form the inlier set. A specific threshold had to be defined for the homography to evaluate which points should be included in the outlier set and which should not. This was an interesting part of the assignment since the reasoning behind selecting the threshold, T has to be backed by

mathematics. As mentioned in Chapter 4 of Multi View Geometry [11], for the estimation of the homography using ransac if the sample set of feature correspondences has zero mean error then T can be chosen empirically however, if there is a probability distribution defining the set of matches then it may be calculated to find the optimal threshold. This specific value was evaluated to be 0.001 for my program.

Step 5: The images were stitched together using the alpha blending object of MATLAB. A binary mask was used to find the section of the image with similarity after which the overlay was carried out. A bilinear interpolation followed after which the result was stored in one image. The stitching that was carried out in each iteration was done with respect to the reference image which was in effect transformed as well using the identity matrix.

Results

I evaluated the results of my algorithm using 4 datasets. The two movie datasets were stitched together using two different sequences as they are present in the datasets folder. It is important to mention that difference sequences were taking for the same dataset using an increment of 3 and 4 images respectively. For the datasets provided this was an effective approach since there were a large number of images present. There are a total of 4 datasets out which 2 are the ones I collected with my iPhone 6S. It is also important to mention that these images had to be downscaled since the original number of pixels in the image were 12, 265, 344. The images were downscaled to 40 % of their original size to get about 5 million pixels which is still easier to process than the former case. I also noticed the **ransacfithomography** function fail to accurately estimate the homography even after 1000 trials since a higher number of pixels causes the corresponding points to be inaccurate to some sense due to a high level of detail in those images. The results are presented on the next page.

Fnu Tulha

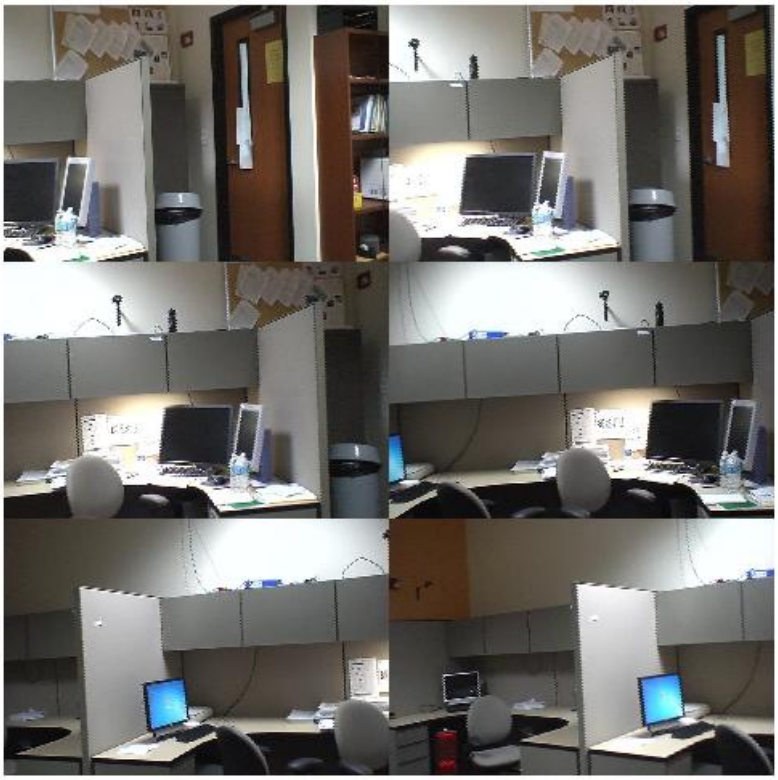
Input images for Dataset 1 sequence 1

Panorama image for Dataset 1 sequence 1

Input images for Dataset 1 sequence 1

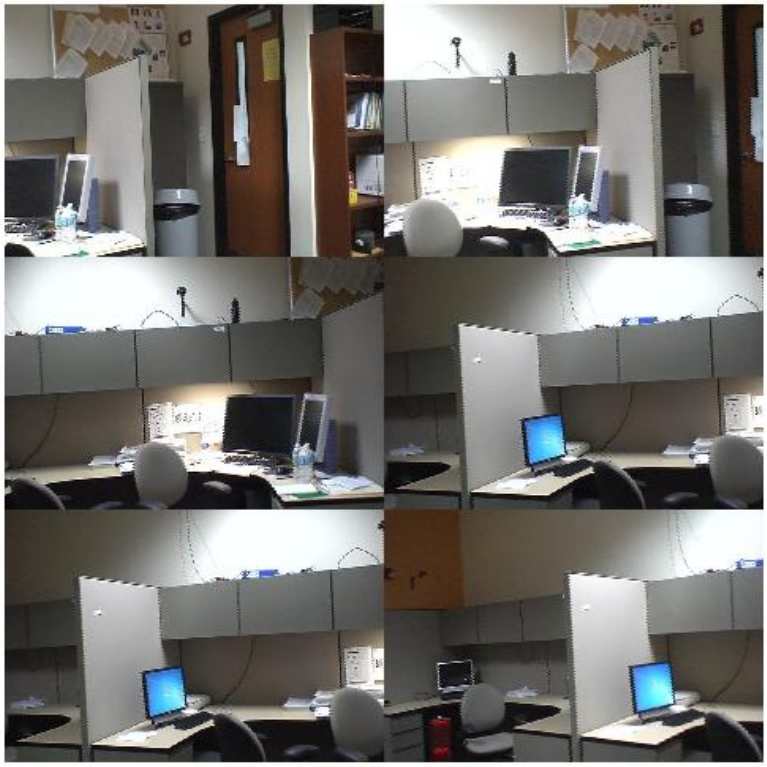
Panorama image for Dataset 1 sequence 2

Fnu Tulha

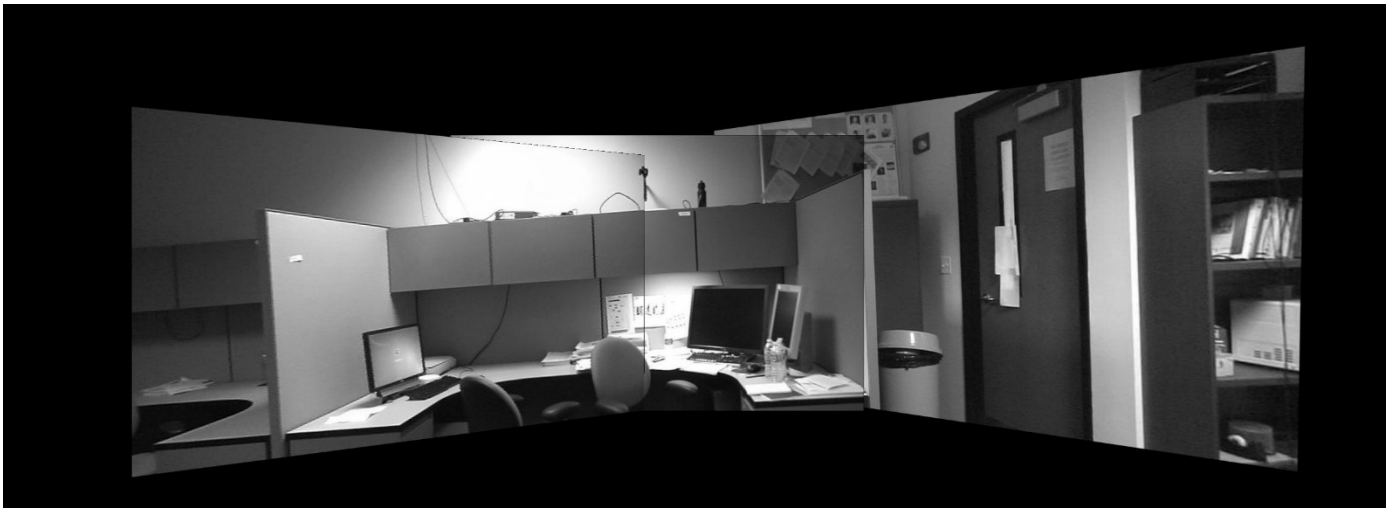


Input images for Dataset 2 sequence 1

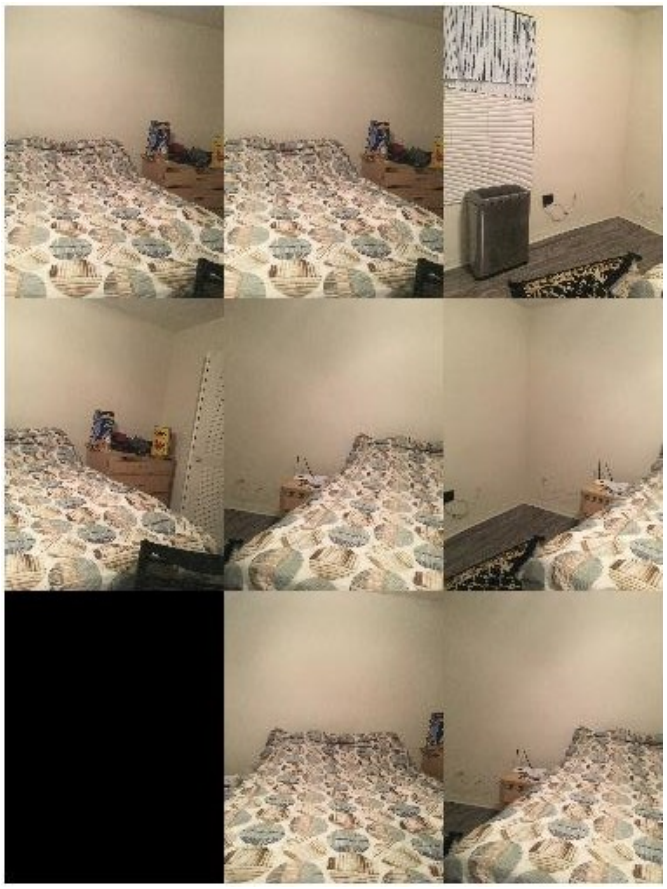
Panorama image for Dataset 2 sequence 1



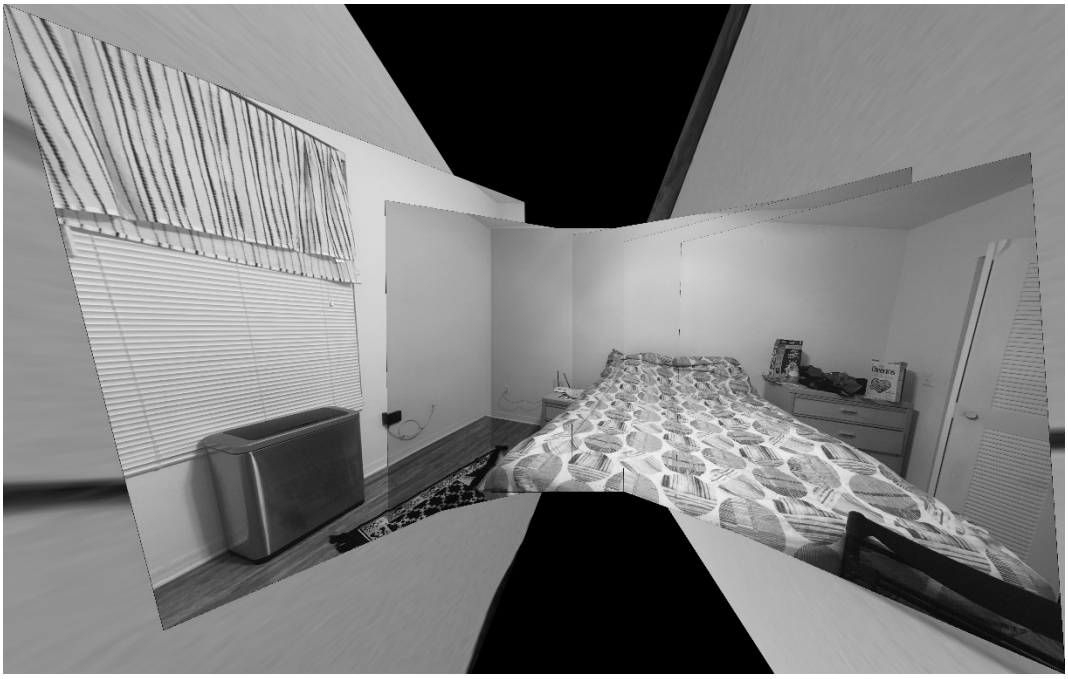
Input images for Dataset 2 sequence 2



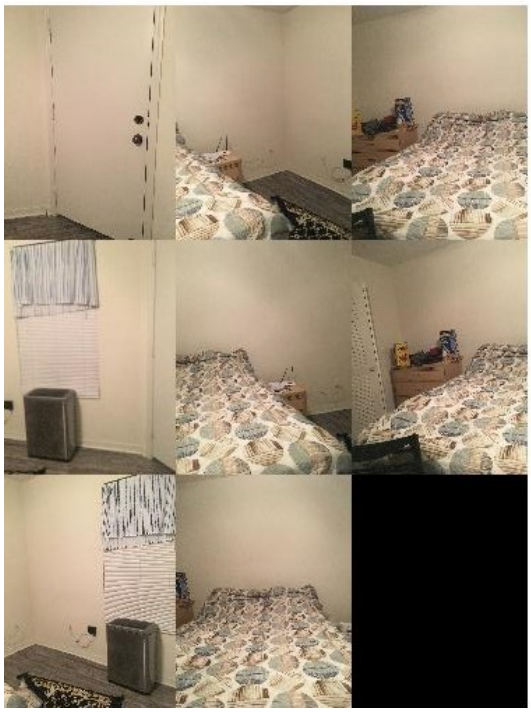
Panorama image for Dataset 2 sequence 2



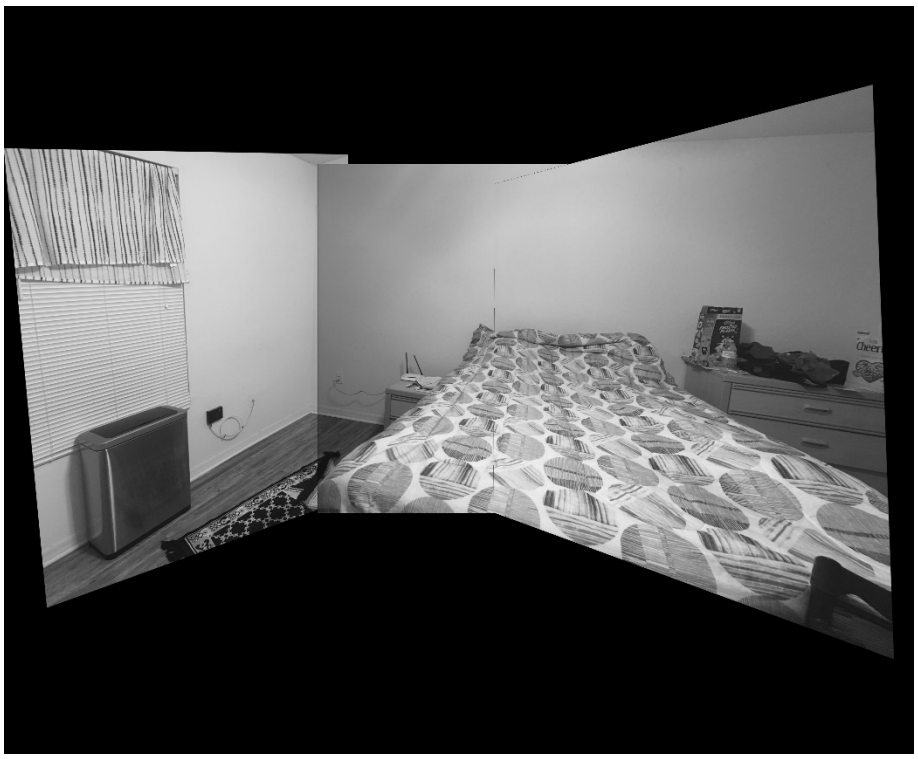
Input images for Dataset 3 sequence 1



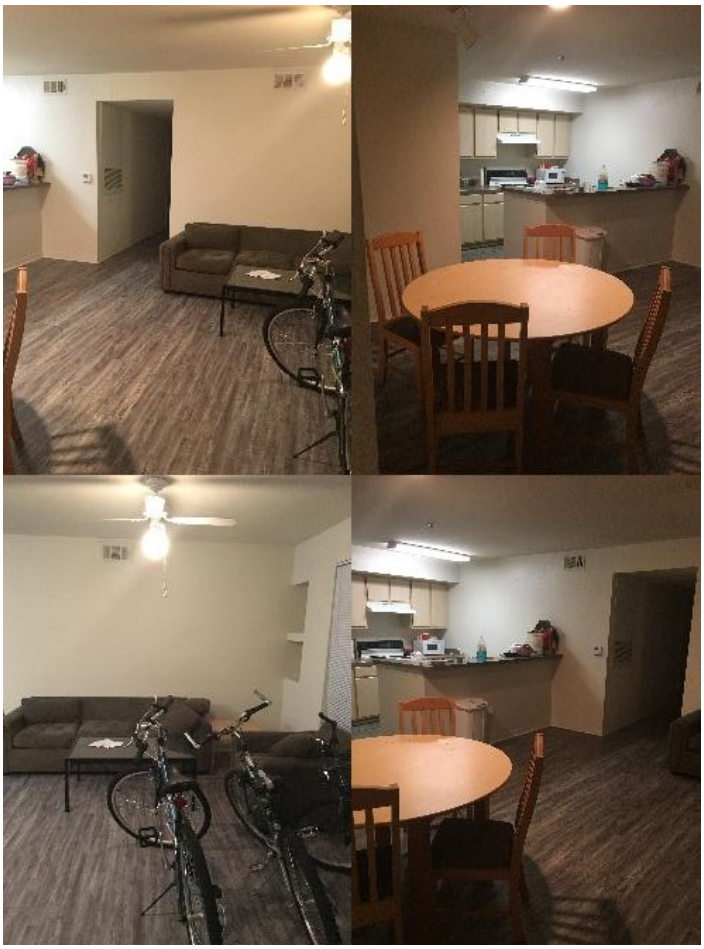
Panorama image for Dataset 3 sequence 1



Input images for Dataset 3 sequence 2



Panorama image for Dataset 3 sequence 2



Input images for Dataset 4

Panorama image for Dataset 4

References

1. Multi View Geometry in Computer Vision, Richard Hartley and Andrew Zisserman
2. http://www.cs.ubc.ca/~lowe/keypoints/
3. https://www.mathworks.com/help/matlab/matlab_prog/cell-vs-struct-arrays.html
4. https://www.mathworks.com/help/vision/examples/feature-based-panoramic-image-stitching.html
5. https://www.mathworks.com/help/vision/ug/use-surf-features-to-find-corresponding-points.html
6. http://www.peterkovesi.com/projects/index.html
7. https://www.mathworks.com/help/vision/ref/extractfeatures.html
8. https://www.mathworks.com/matlabcentral/answers/385-how-to-read-images-in-a-folder
9. https://www.mathworks.com/help/images/ref/montage.html
10. http://www.vlfeat.org/
11. Multi View Geometry, Richard Hartley and Andrew Zisserman