3D Computer Vision – Homework 3

In this homework, we will be extensively working on one of the popular features on every feature, so called “Panaroma Stitching” or simply “Image Stitching”. Going ahead in this assignment, the steps involved in performing image stitching are explained.

First step was to capture a series of custom pictures for the demonstration of the image stitching. The images were captured on an iPhone; hence 2 major pre-processing’s were needed before sending the images for image stitching namely

1. Conversion of the. HEIC to JPG (MATLAB doesn’t support .HEIC extensions)
2. The images were captured at a resolution of 3024x4032, hence required resizing to a lower resolution, provided aspect ratios were maintained.

While capturing the images, it was taken care to maintain the uniform centre. A total of 12 images were captured in series. Once the image was read using the “imread” function in MATLAB, the resize operation was performed to reduce the computation by resizing the image from 3024x4032 – 600x800. Sample images captured have been attached below in the Fig 1.

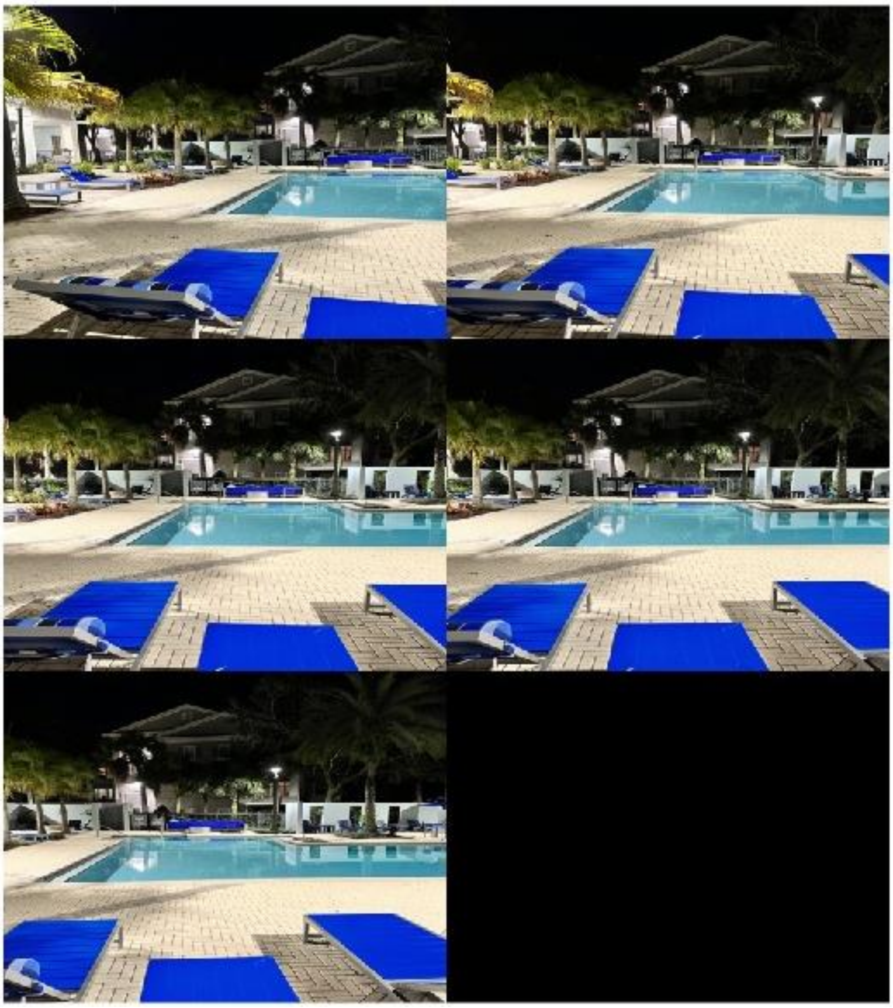

Fig 1. Sample images captured from the iPhone

Now that we already have the images to proceed with, the next important thing is the SIFT+RANSAC mechanism. To explain in short about the SIFT, it basically applied gaussian filters on the image with different sigma values and get the key points from the images comparing it with it's 26 neighbours from the adjacent images of same size with different sigma values. Finally, the descriptors of size 128 bits are generated for all the key points detected using the Difference of Gaussian mechanism considering the key point and its precise surroundings.

In the MATLAB code that we have written, a popular library called vlfeat was used, which provided a function namely 'vl_sift' to get the key points as well as the descriptors for the same.

Going to RANSAC, it stands for Random Sampling Consensus. It is a simple mechanism to construct a model based purely on the inliers and separating out the outliers. In the MATLAB code provided a custom function has been written down to consider a set of 4 matched points for performing the RANSAC operation. The steps involved in the SIFT + RANSAC are explained below

1. Key points + Descriptors are calculated for every image passed to the code
2. Now say we want to stitch the initial set of 2 images, then the features and key points are passed on to the matching functions, which gets us the set of matched points on both the images usually based on a distance.
3. Once these matched key points are obtained, the next step is to model a transformation/homography between them. This is where RANSAC comes into picture.
4. In the code provided, we have initially set the sample points to 4, which means a subset of 4 points would be chosen to calculate the transformation matrix.
5. Now the operation DLT is performed, where in the matrix is created out of the selected points. Now the SVD is performed to get the final homography matrix.
6. For every homograph matrix created using the subset of points, a score is generated to see the level of fit.
7. The score
8. This RANSAC Operation is run for prescribed number of times. The more the runs, the better the output, as the algorithm will be able to try its luck on different combination of key points.

Experimentation factors

1. Varying the number of RUNS for RANSAC
2. Changing the distance threshold for RANSAC Score
3. Modifying the threshold for matching function
4. Number of points to be considered for RANSAC

Next coming to stacking of the images to form a panorama, 2 specific ways have been tried.

1. First approach involves considering the first image as the reference and calculating the homography.

2. Second approach involves considering the central image as reference.

Output of both the approaches has been attached here with this report.

The above-mentioned approaches are run all the images either first image approach or the central image approach. The outputs are demonstrated in the images below Fig 2. – Fig 4.

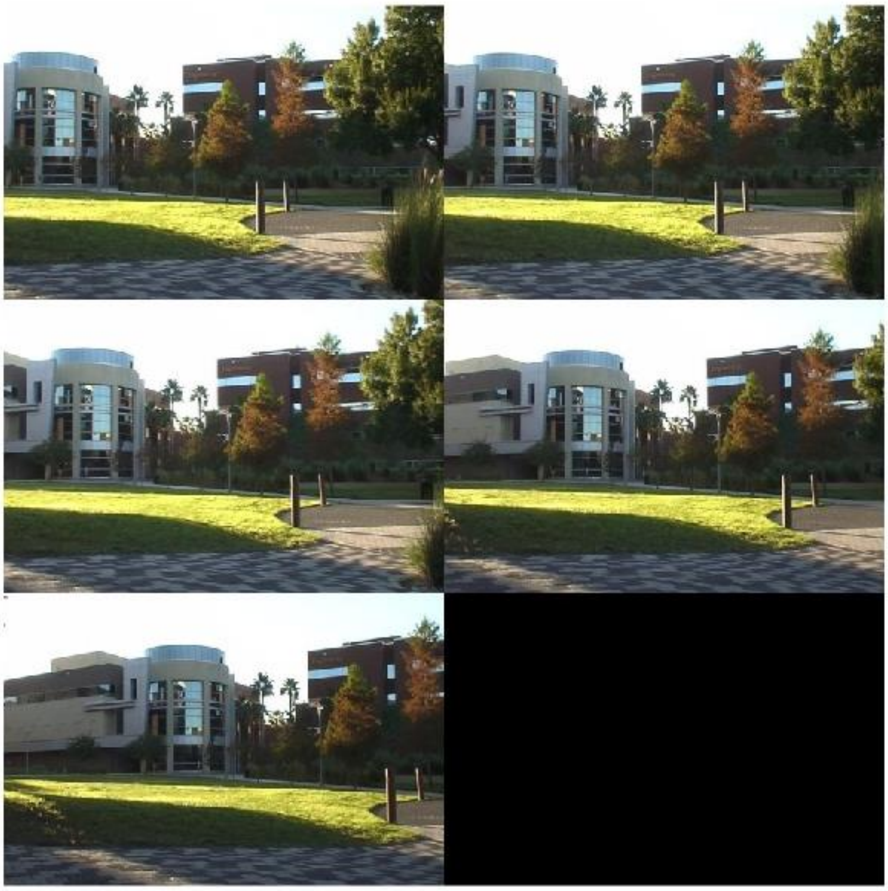

Fig 2. Input images-1

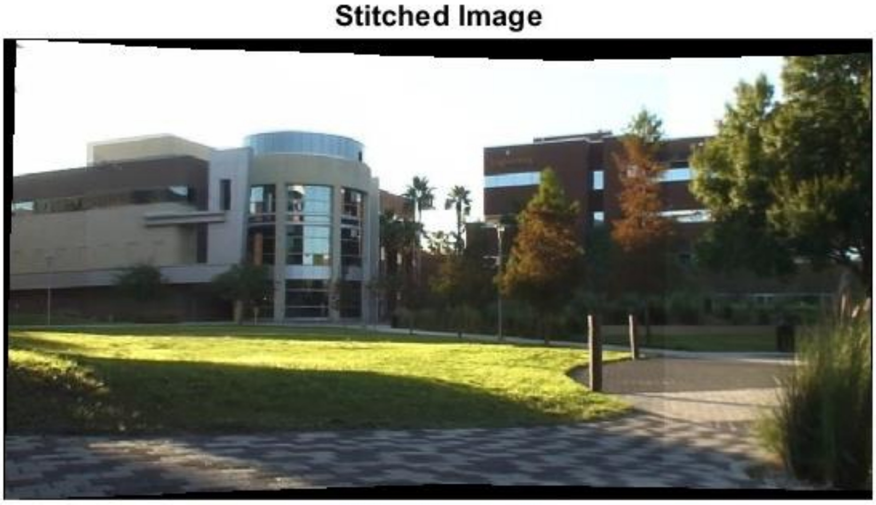

Fig 3. Panorama Image -1

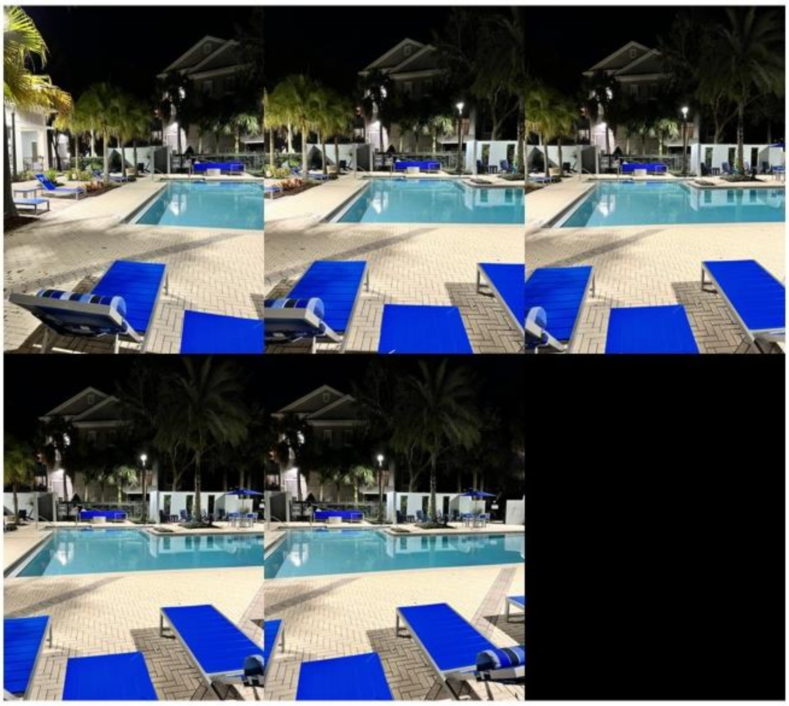

Fig 3. Input images -2

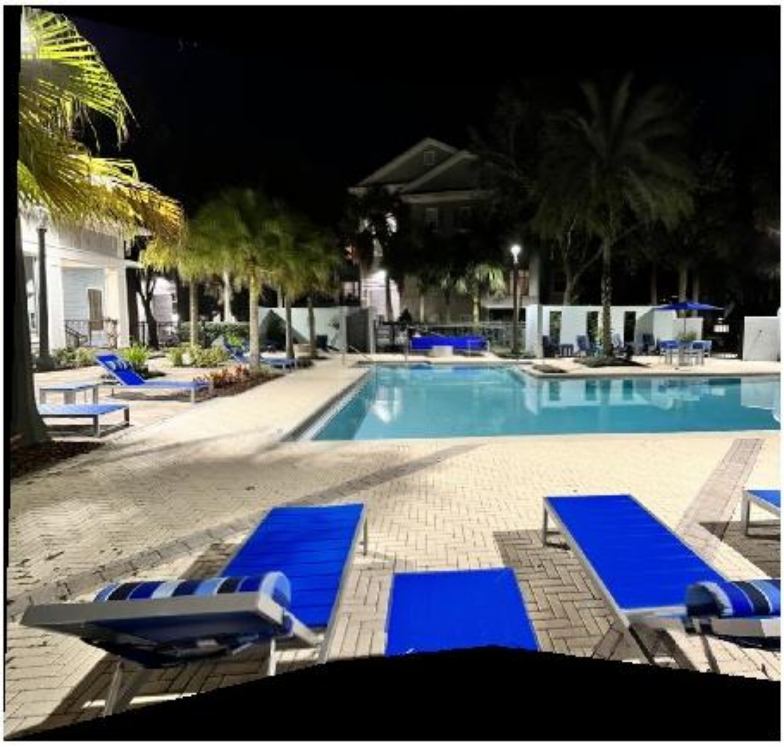

Fig 4. Panorama Image -2

Above figures from Fig 2. – Fig 4 portrays the functionality of the image stitching based 2 set of images. Following were the default parameters set

1. RANSAC iterations = 100

2. Minimum number of points = 4
3. Feature matching distance threshold = 1.5
4. Number of images used = 5

Observations

1. RANSAC Iterations: Coming to RANSAC Iterations, it was noticed that as the iterations are reduced below 10, there is a sudden degradation in the quality of stitching of image. And, just as the RANSAC follows a random sampling procedure, the stitching output varies every time we run the code. The reason being that the RANSAC choses different set of key points to construct a homography every single iteration. The below images describe the same phenomenon Fig 5 – Fig 8

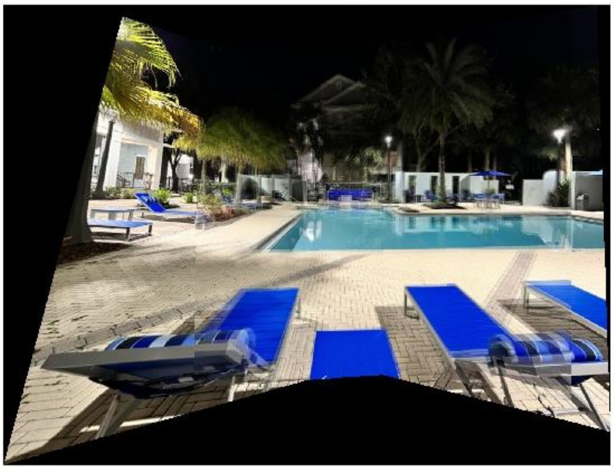

Fig 5. Iter = 5

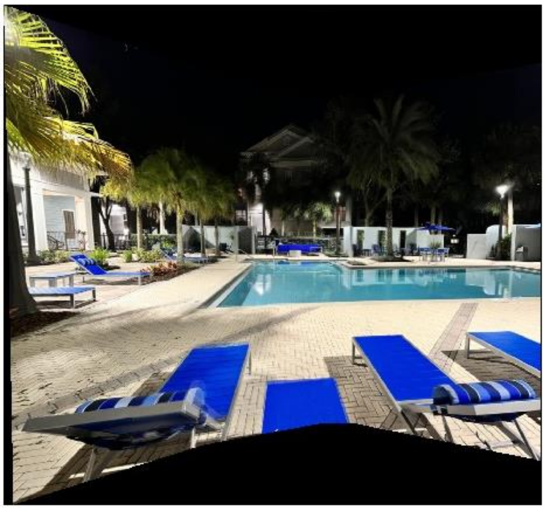

Fig 6. Iter = 10

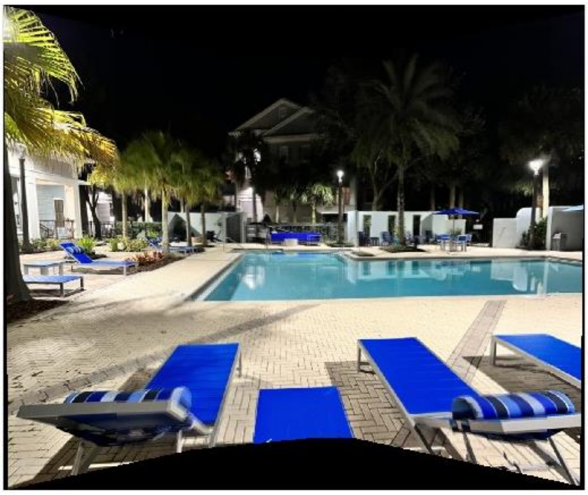

Fig 7. Iter = 100

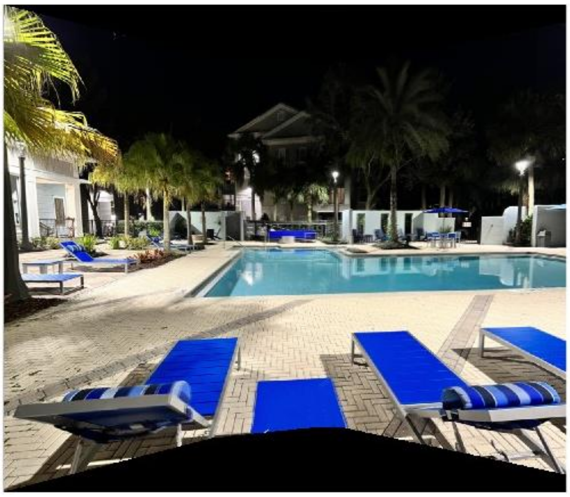

Fig 8. Iter = 1000

2. Closely captured images vs less overlaping images : A simple experiement was performed to see how the stitching works with difference in spacing of images. It was found out that , the panorama's were more clear when the images were closely

spaced to each other as compared to less overlapping images. The simulation of less overlapping images were obtained by skipping 1 or 2 images in a sequence of closely captured images. The below shown images describes the phenomenon Fig 9 & Fig 10

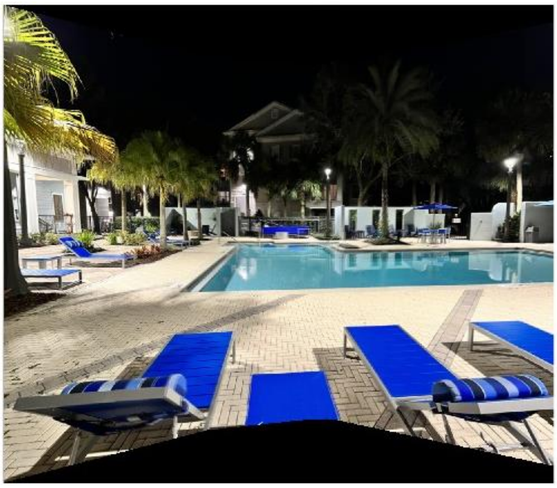

Fig 9. Less overlapping

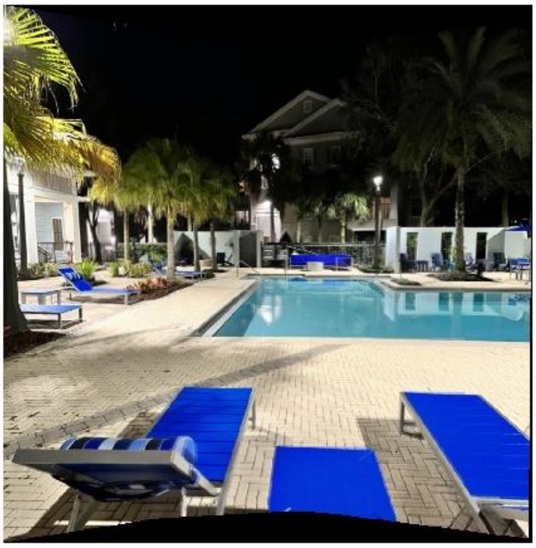

Fig 10. Closely placed

3. Feature matching threshold
   It was experimented to see if the threshold used in the vlfeat's 'vl_ubcmatch' really mattered. On running the code with different set of thresholds, it was noticed that there was no difference in the quality of the images, provided the number of RANSAC iterations was set to 300 and the number of points for homography was set to 4. It could mean that even though the the distance threshold was set high, RANSAC was doing a really good job in refining the outliers and presenting us with only the best homograhy estimation.

4. Reference image consideration – First image or the center image
   First 2 codes have been written to check the quality of images obtained using first image as reference and center image as reference. It was found out that :

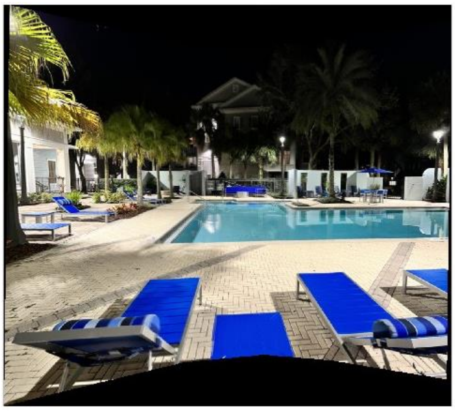

Fig 11. Center

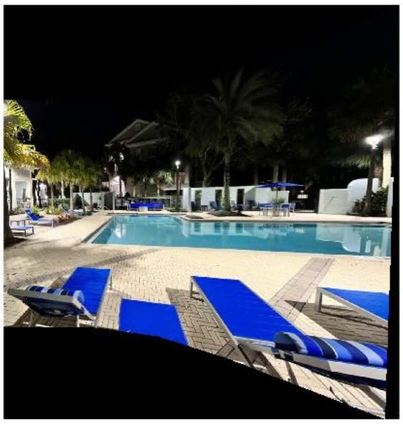

Fig 12. First

1. When first image was used as reference, stitched image was slightly blurry and more distorted as compared to the output of stitched image with center as reference. The images are shown in the figures above in Fig 11 and Fig 12. It has to be noted that both the outputs were obtained from the same parameters.

5. Ransac Sample points: Coming to DLT, which is currently being used in the code to calculate the homography matrix H, is a simple operation where the transformation matrix is found between the point correspondences. According to the textbook, we need atleast 4 such correspondences to calculate the unknowns in our H matrix (8 unknowns, 1 is left out as homography is defined upto a scale). But it doesn't really mean that we need only 4 points, we can actually consider all the point corresondences to calculate the H.
   When only 4 points are considered for H calculation, our linear equation looks like this : Ah=0 where A= 8x9 h=9x1, but this can be increased to say A= 16x9 h=9x1 as the multiplocation is still possible. A similar experiment is tried in this homework to see if there is any improvement in the stitching. Here are the observations for the same

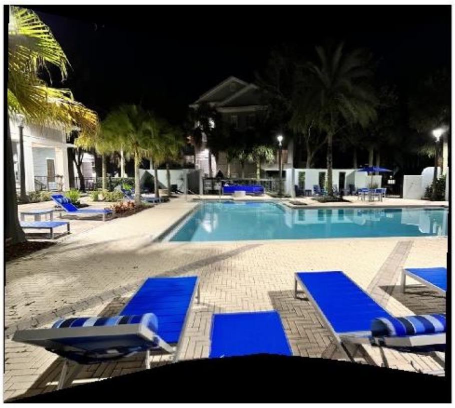

Fig 13. RANSAC Iter = 100 , num points = 16

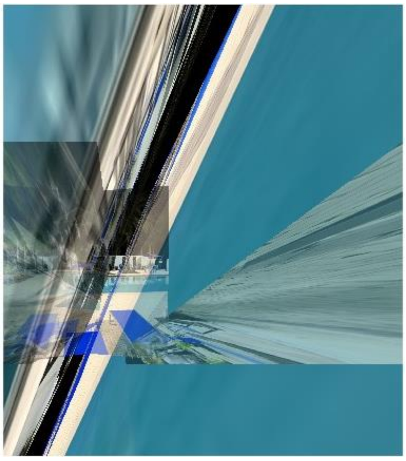

Fig 14. RANSAC Iter = 100 , num points = 50

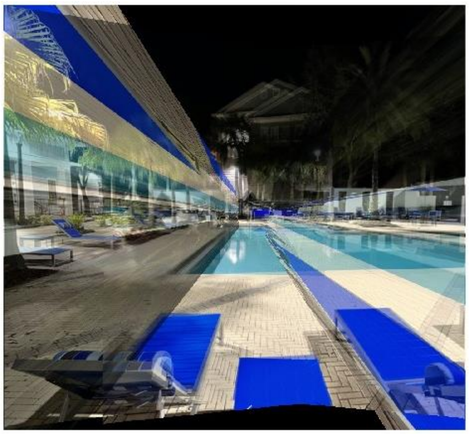

Fig 15. RANSAC Iter=1000 , num points = 16

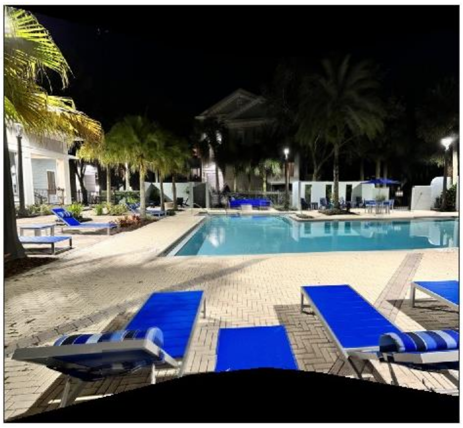

Fig 16. RANSAC Iter=2000 , num points = 16

1. MATLAB gives out an error if all the points are used out. Hence the experiment was tried on smaller number of sample points namely 16 and 50
2. The experiment was performed using 2 parameters RANSAC Iterations and number of Sample points. The above images portarys the output of the experiment Fig 13 – Fig 16
3. It was inferred that:
   - It takes slightly longer to reproduce the result when the number of sample points are reduced
   - Another point being, if number of points are increased, then it clearly requires larger number of RANSAC iterations to reproduce better results (which is depicted in the images above). As with lesser iterations, it can be that the RANSAC algorithm is not able to collect required number of inliers, which is needed to produce a H matrix of best score.

References

- https://www.cse.psu.edu/~rtc12/CSE486/lecture15.pdf -- Ransac explanation
- https://pages.cs.wisc.edu/~dyer/cs534/hw/hw4/hw4.pdf -- Homework Explanation
- https://www.mathworks.com/help/vision/ug/feature-based-panoramic-image-stitching.html -- Mathworks image stitching
- https://www.mathworks.com/help/vision/ref/estimategeometrictransform.html - geometric transform
- https://github.com/rahul-kothari/Image-Stitching -- main code for image stitching
- https://theailearner.com/2020/11/06/perspective-transformation/ -- perspective transform
- https://www.vlfeat.org/install-matlab.html -- VLfeat for vlsift or other fucntions
- http://weitz.de/sift/#:~:text=The%20scale%2Dinvariant%20feature%20transform,be%20used%20for%20object%20recognition
- https://www.youtube.com/watch?v=l_qjO4cM74o – Columbia University

Note:

Following are the contents of the zip file

1. Image stitching code with center referenece image
2. Image stitching code with first referenece image
3. Custom images captured by me
4. Report for Homework-3