PURDUE
UNIVERSITY®
FORT WAYNE

ECE 66100: Computer Vision Project 2 Report

Mann Raval
(ravam01@pfw.edu)

Course Coordinator:
Dr. Bin Chen

ECE 66100

Problem Statement

Panoramic Image Stitching of the images. This project is to implement a pipeline for panorama stitching, combining individual photos with overlapping fields of view to produce a panorama image.

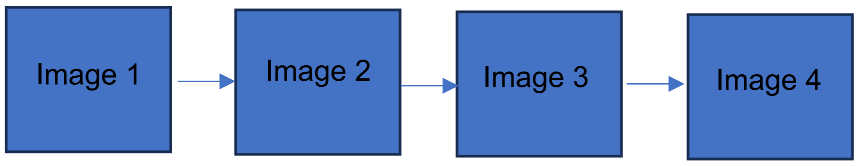
Design Methodology

There are four images in the given project. The images are taken from a singular point with a camera which is rotated about a vertical axis. There is also a small overlap in each image which helps in getting the features and the key descriptors in each photo. Assuming there is no rotation along the horizontal axis, the images should be stitched together. If there would have been some rotation along the horizontal axis, the first task would have been to preprocess the image into a same orientation of image, then the image would have been stitched together. The images should combine and form a panorama.

Procedure

The images are in a sequence going in horizontal direction. For the simplicity of programming, the stitching is done in left to right direction. The first step is to get the images in order. Conversion of the images into *double* data type is done and then it is converted into gray scale image. Since the OpenCV functions work efficiently on the grayscale images, they are converted into grayscale images. Using Scale Invariant Feature Transform or SIFT, the key points and the descriptors are found. These help in getting the good features that can be utilized in the feature matching process. Using a predefined number of matchers, the features are mapped. The images need to be warped to make the images into the same coordinate system. Using Random Sample Consensus or RANSAC, the numbers of inliers are computed for the stitching of the images. The RANSAC algorithm can take a maximum of 50% of the outliers so it is very efficient to use it. Using the RANSAC Algorithm, we can get the homographic matrix. Using the homographic matrix, we can approximate the matrix using the Image warping which will convert both the images into the same coordinate system. Then we overlap the images on top of each other. Often this will create a seam on the image, which will deteriorate the quality of the image. To enhance the quality of the image, we use a method called image blending in which we blend and merge the images to enhance the look of the photo.

There are four images that need to be stitched together. The flow is in such a way that we stitch the first two images by computing the descriptors and key points of the images. Then we merge image 1 and image 2 together to get the stitched image. Then using the stitched image using image 1 and image 2, we stitch it to image 3 following the same methodology of computing the key points and the descriptors. Then using the matching features, we merge and blend the images together to form a stitched image of images one, two and three. Then following the same procedure, we can stitch the image four with the same method. The resulting image is the panoramic image of the scene.



The diagram above shows the image stitching process. The same method is applied on the code.

Implementation

The implementation of this project was done in Python. The tools used were Jupyter Notebook. The libraries used are OpenCV, NumPy and Matplotlib. Initially, the libraries are imported, and images are read into the program. A user defined function of *Keypoints* is defined that returns the key points of the images with the descriptors and the number of matches. Using the RANSAC homography the homographic matrix is calculated. This helps in warping the image into the next image.
The procedure is repeated four times in the program to stitch the four images together since there are 4 images in the problem statement. The functionality can be increased and can be repeated for a greater number of pictures also.

Results

The results of the project are as follows:

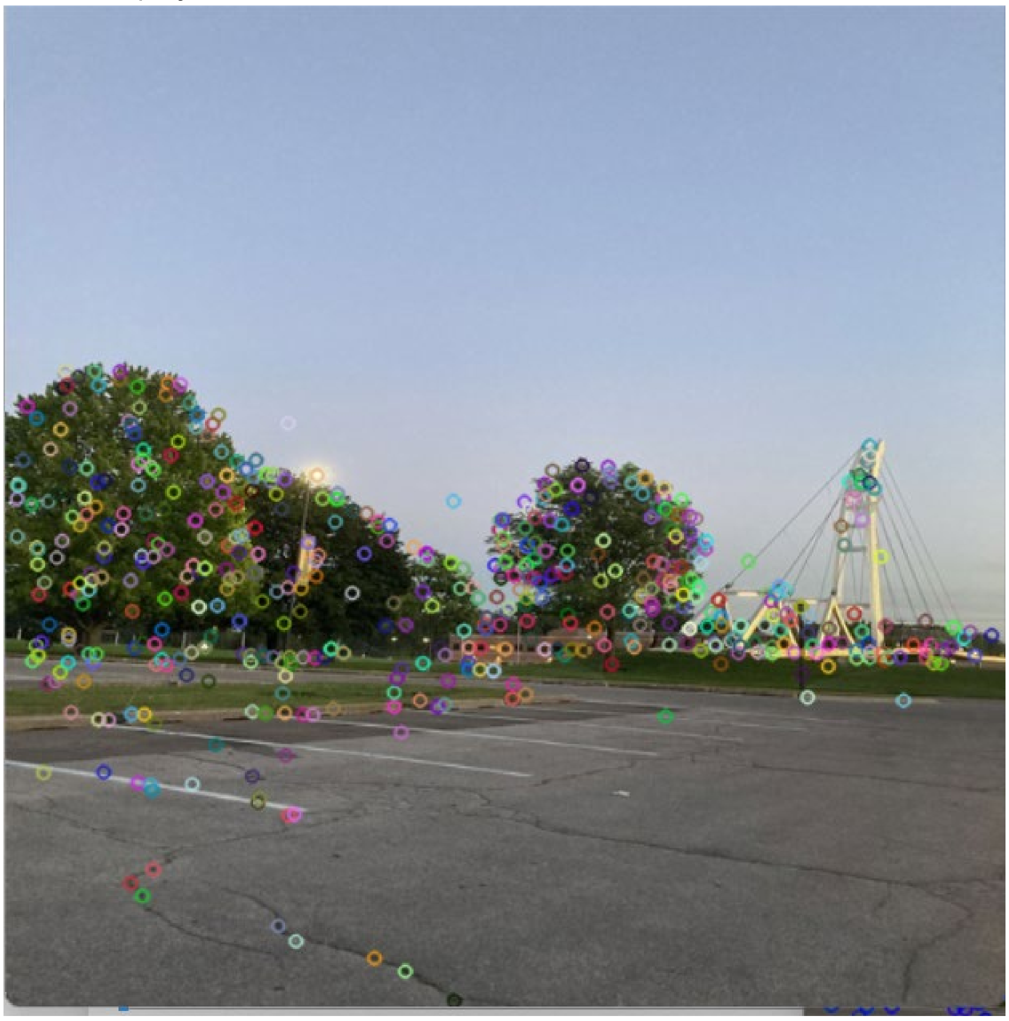


Figure 1. The Key points in Image 1

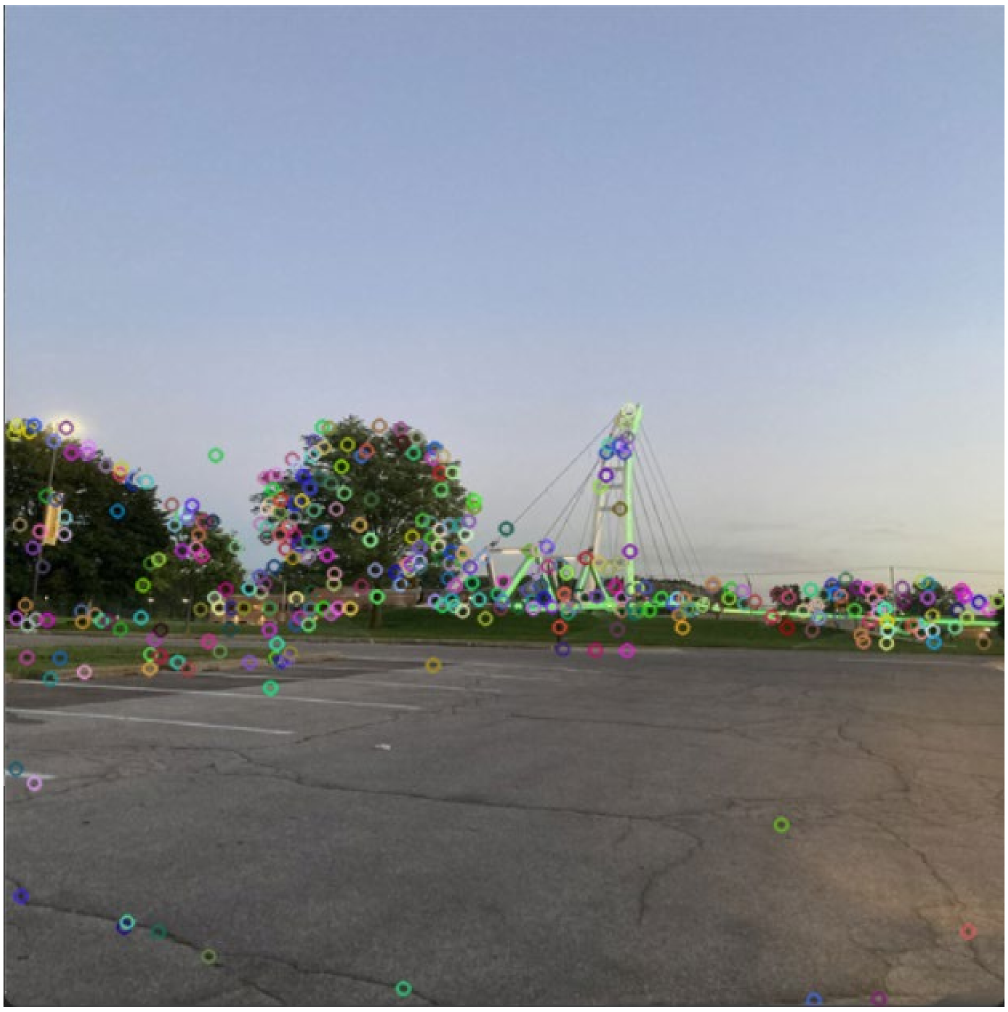


Figure 2. Key points in Image 2

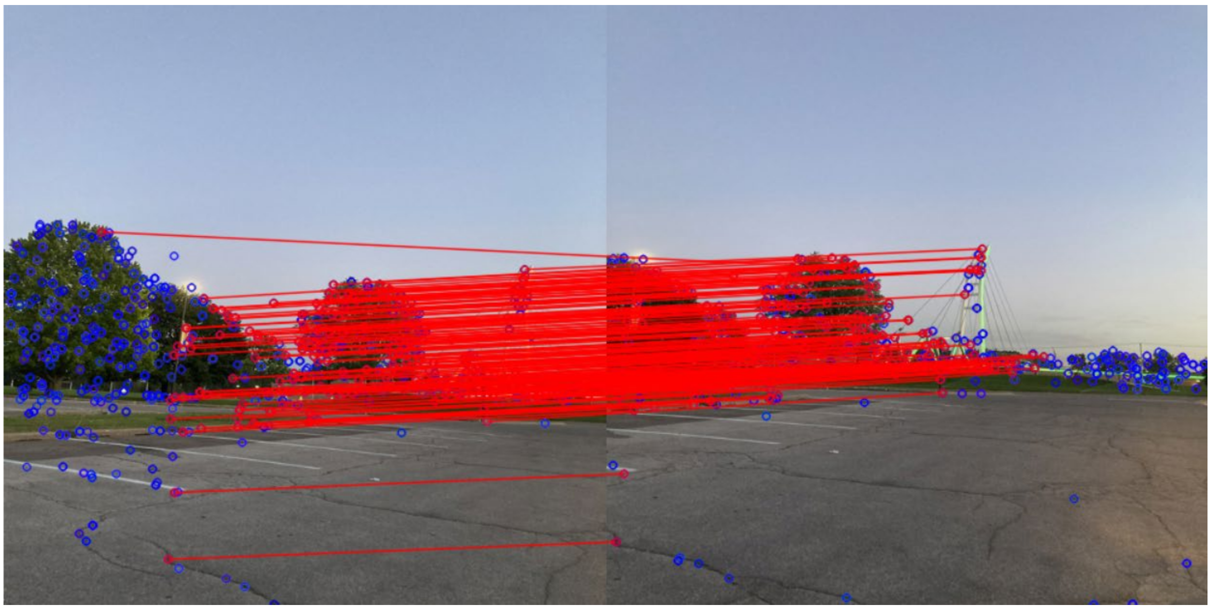


Figure 3. Matched key points on Image 1 and Image 2

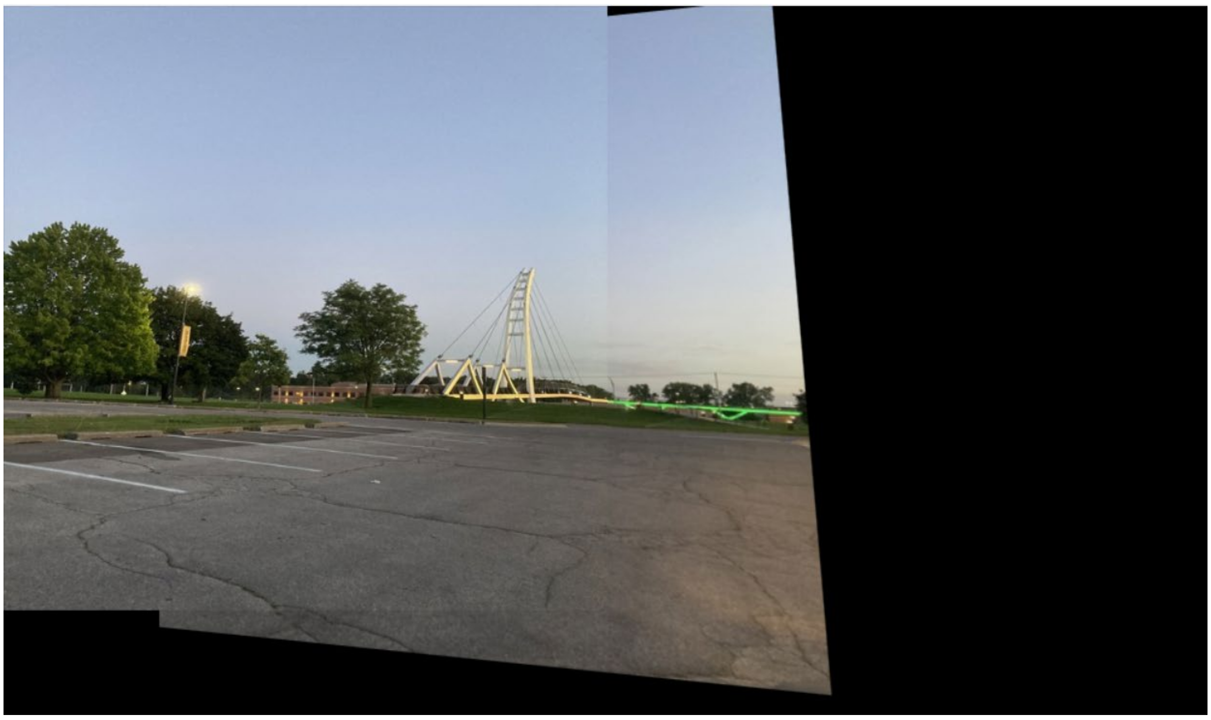


Figure 4. Merged Image of Images 1 and Image 2

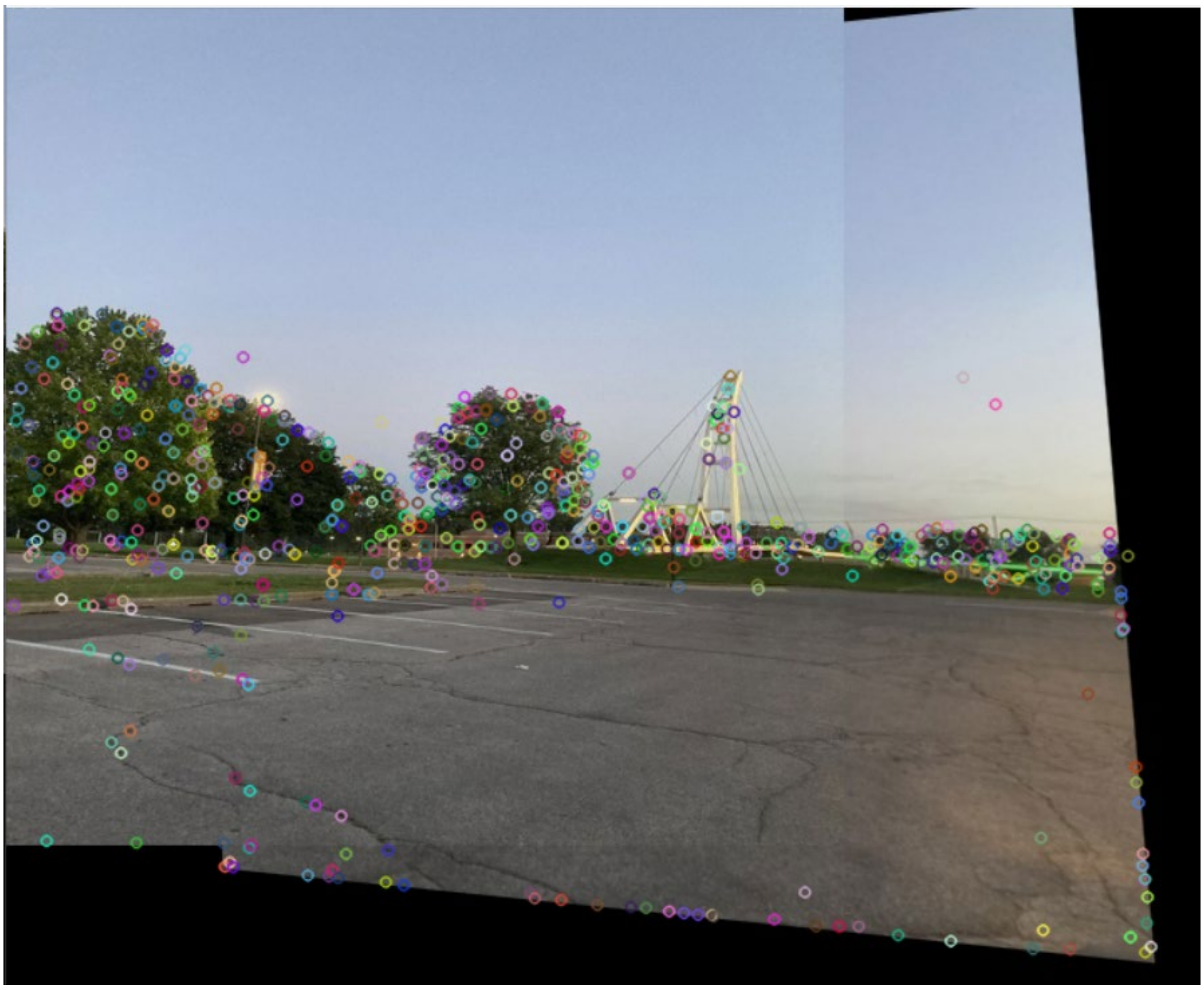


Figure 5. Key points in Image 1 and 2

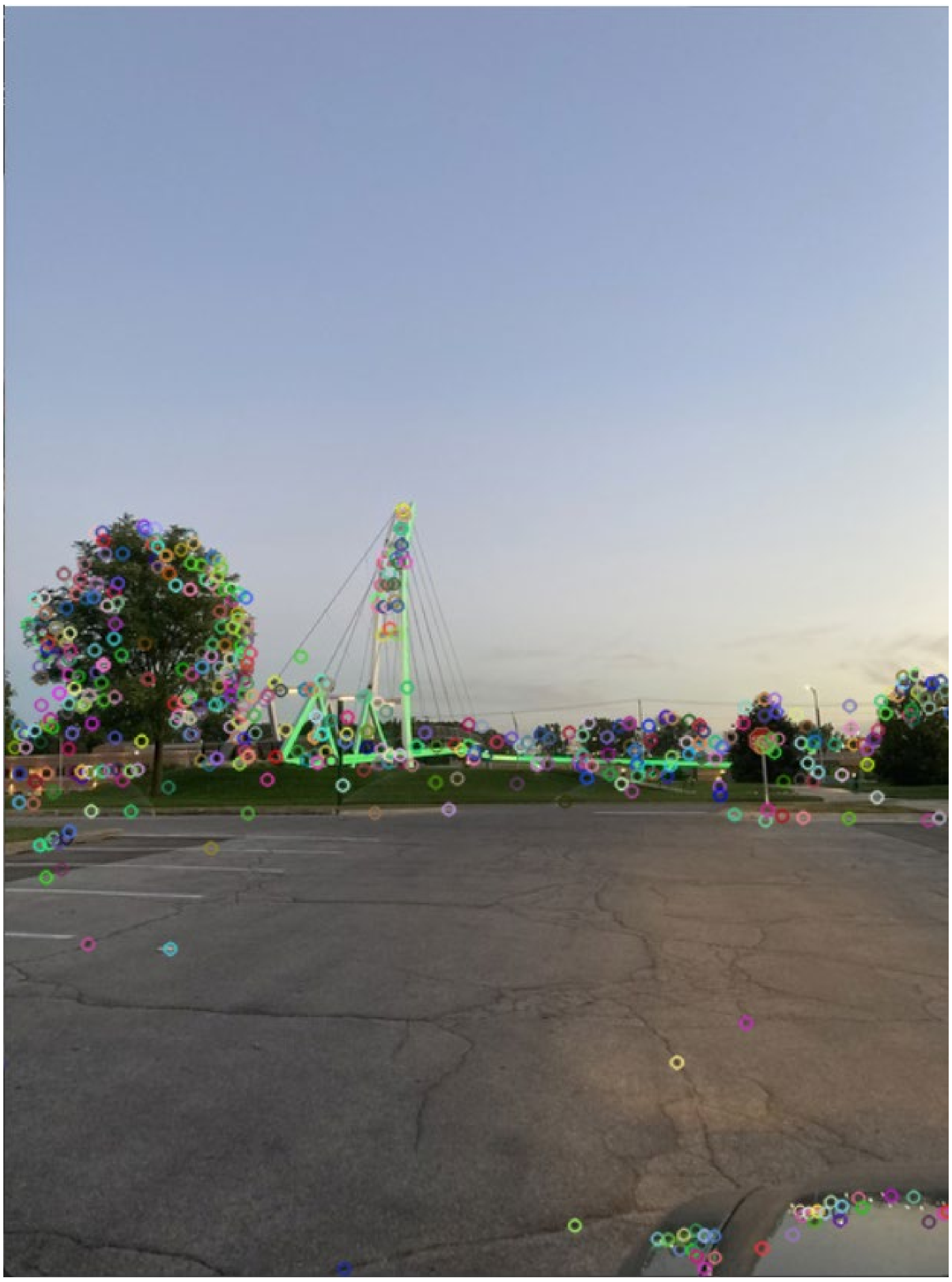


Figure 6. Key points in Image 3

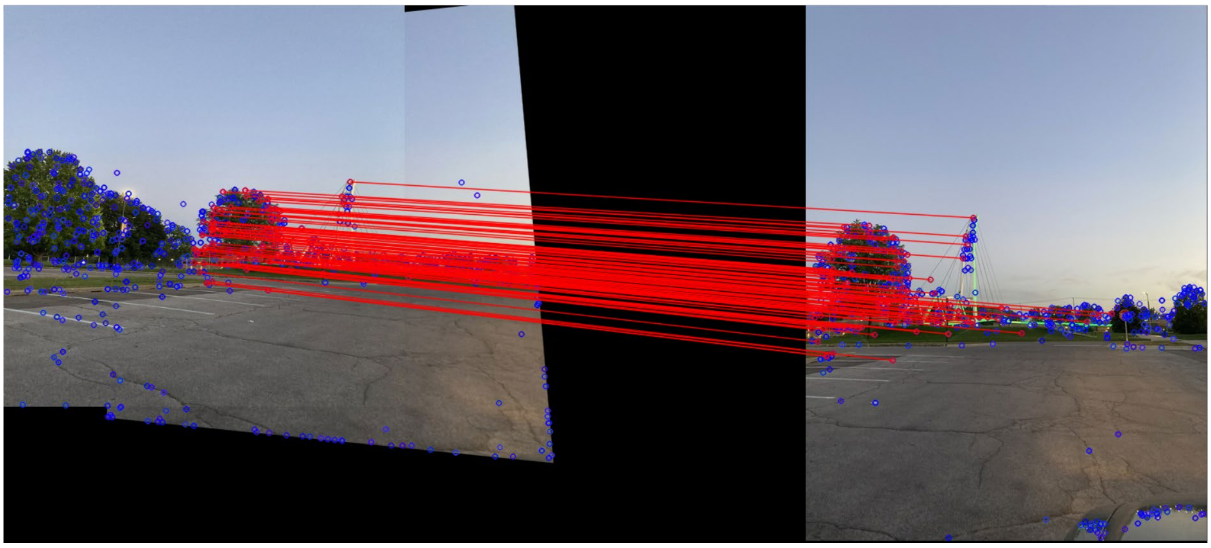


Figure 7. Matched key points in Image 1 and 2 and Image 3

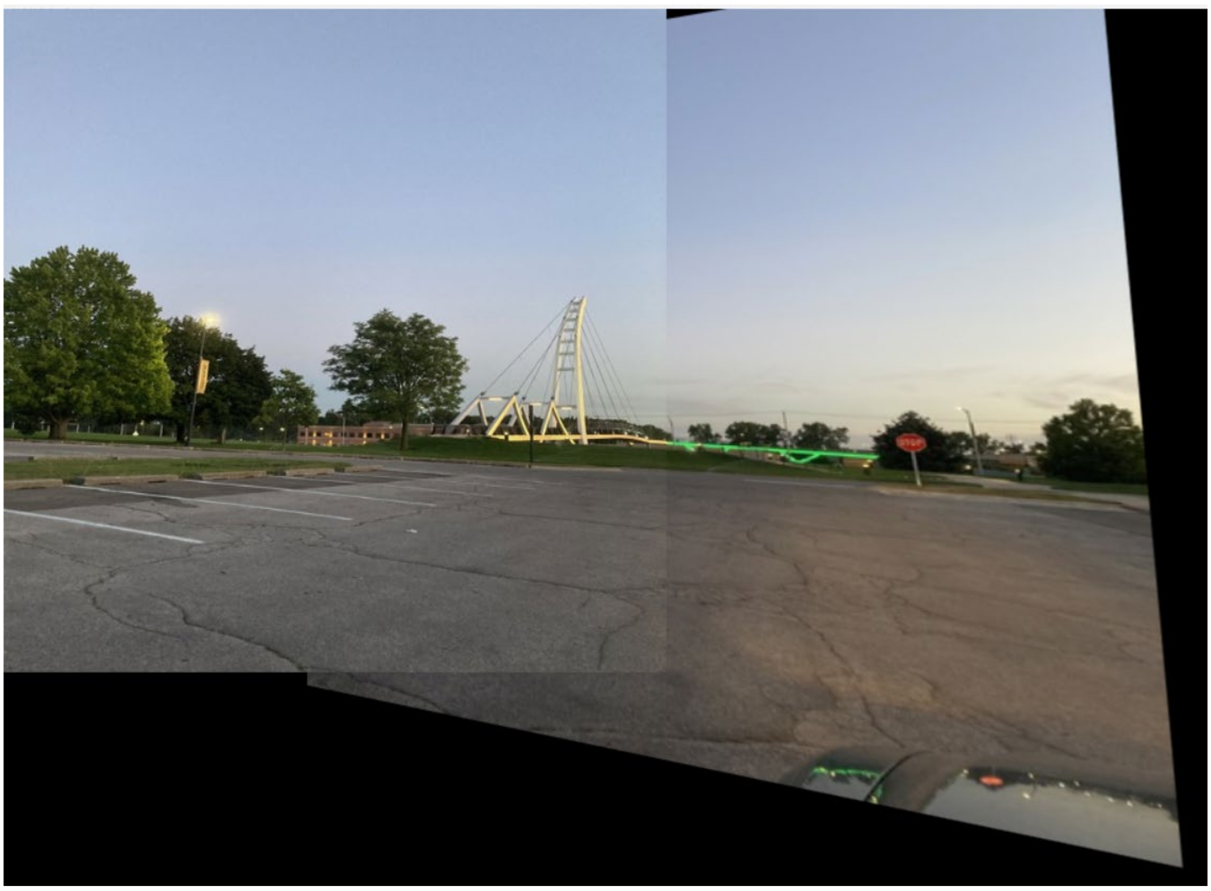


Figure 8. Merged Image of Image 1, 2 and 3

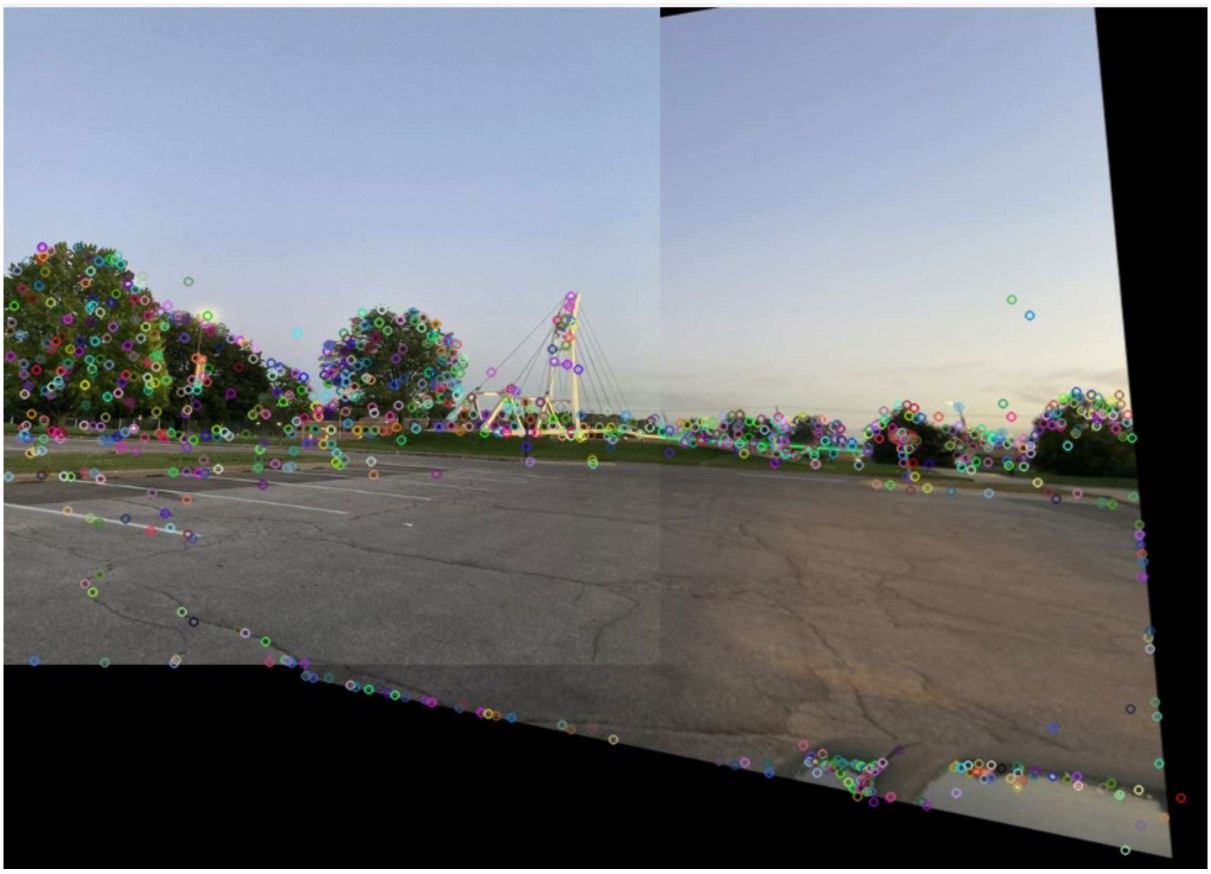


Figure 9. Key points in Image 1, 2 and 3

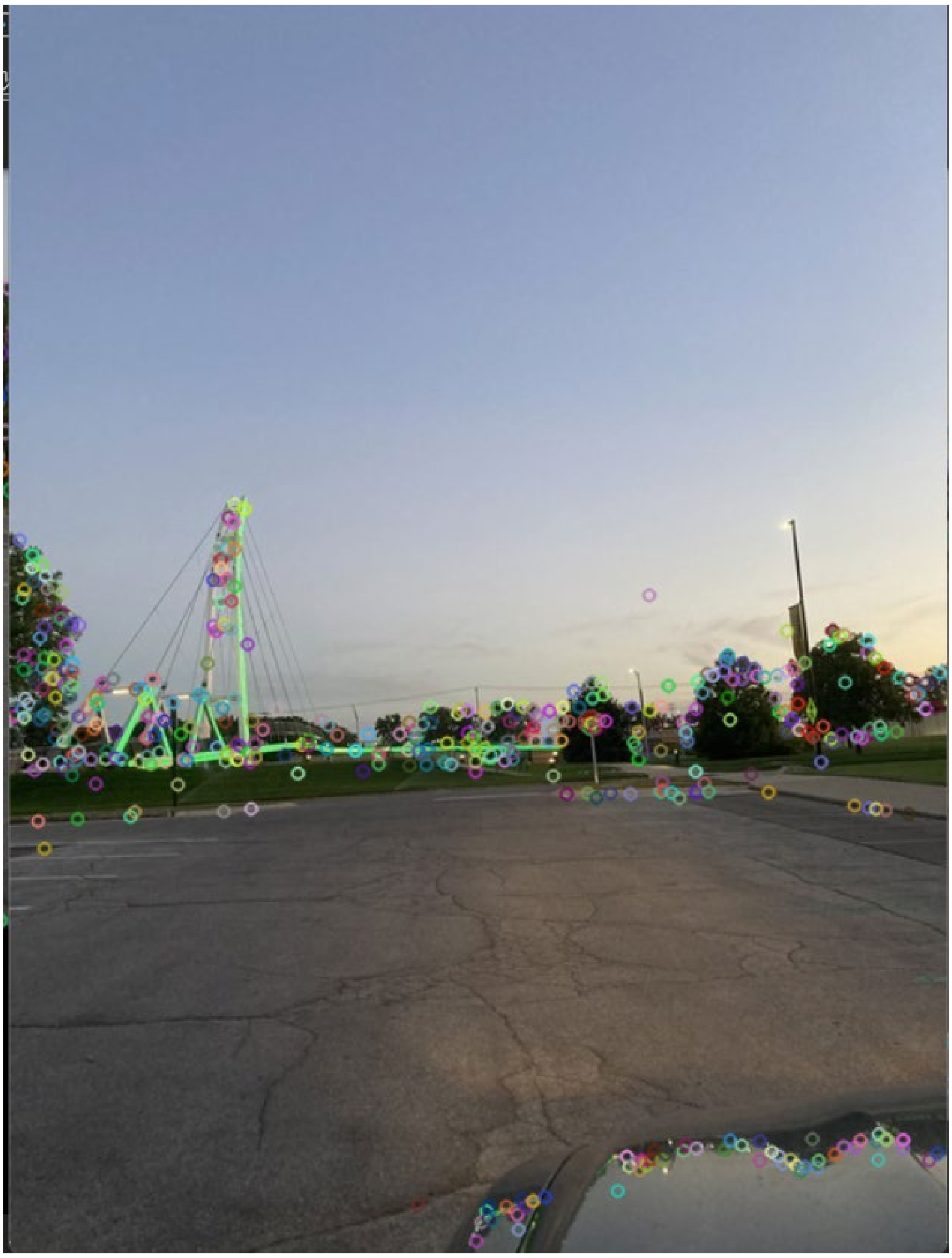


Figure 10. Key points in Image 4

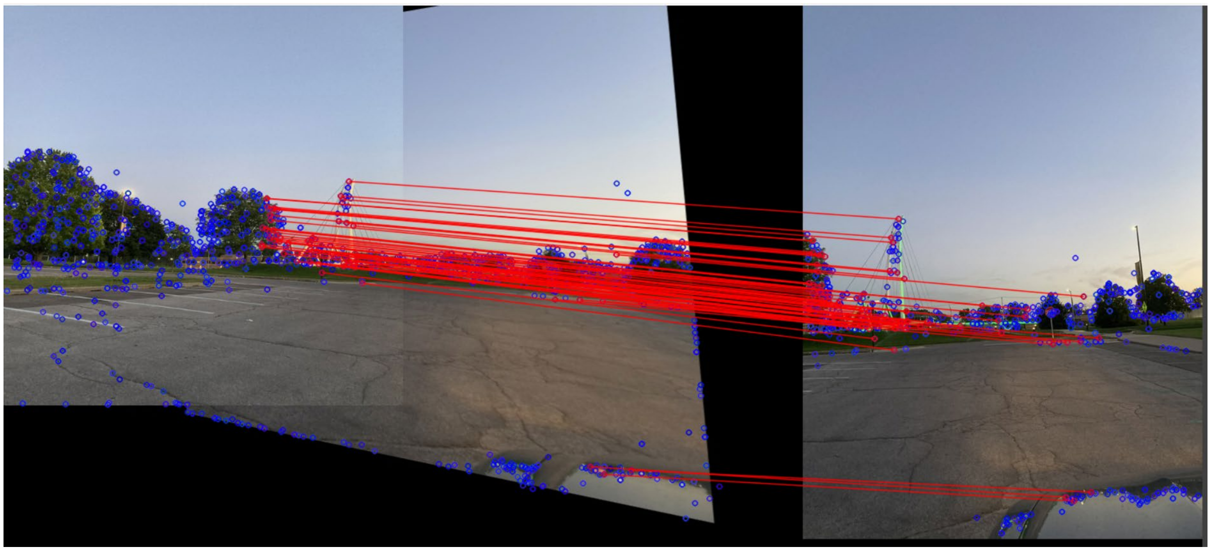


Figure 11. Matched key points in Image 1, 2, 3 and Image 4

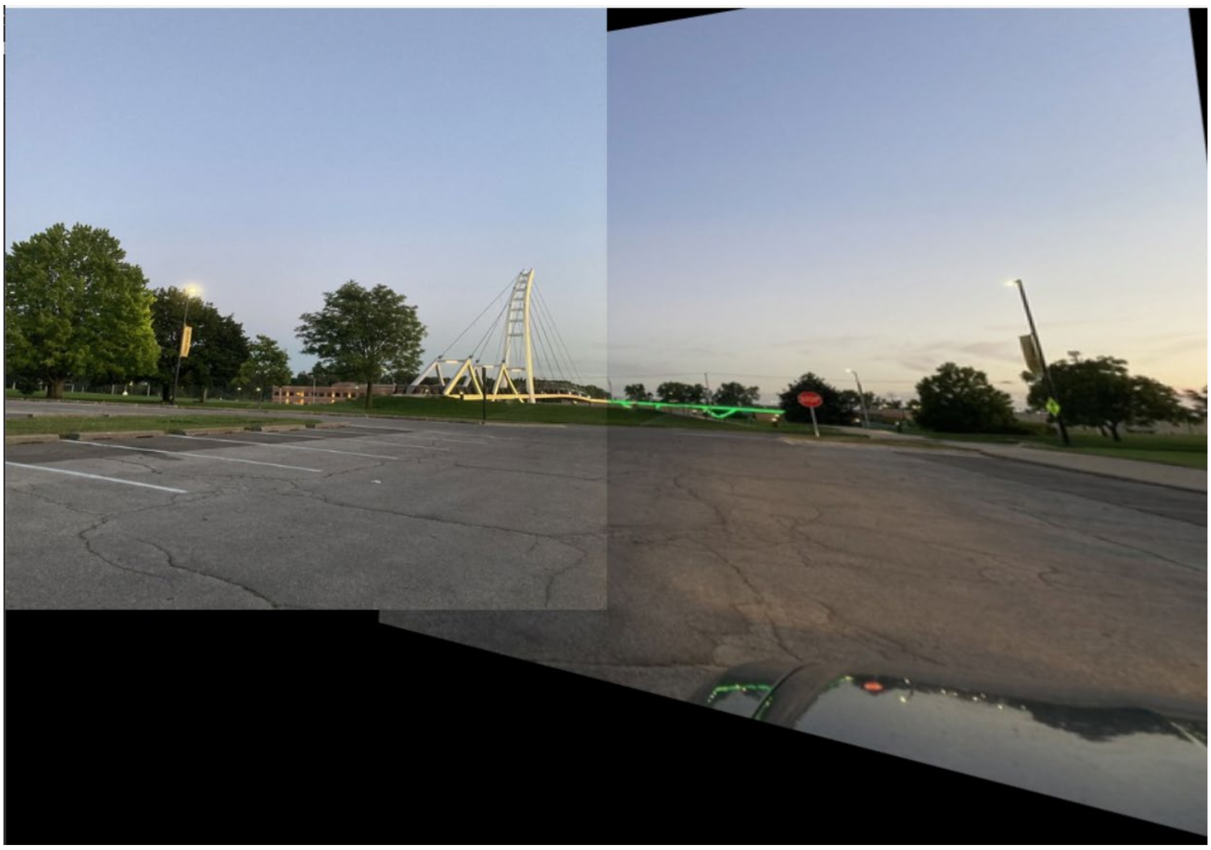


Figure 12. Final panoramic image of Image 1 and Image 2 and Image 3 and Image 4

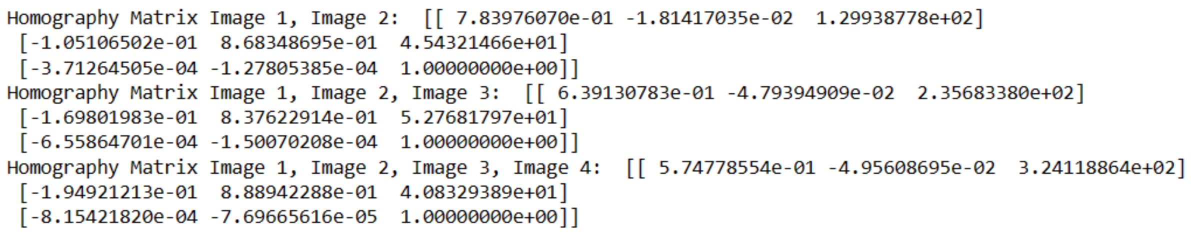

```
Homography Matrix Image 1, Image 2:  [[ 7.83976070e-01 -1.81417035e-02  1.29938778e+02]
 [-1.05106502e-01  8.68348695e-01  4.54321466e+01]
 [-3.71264505e-04 -1.27805385e-04  1.00000000e+00]]
Homography Matrix Image 1, Image 2, Image 3:  [[ 6.39130783e-01 -4.79394909e-02  2.35683380e+02]
 [-1.69801983e-01  8.37622914e-01  5.27681797e+01]
 [-6.55864701e-04 -1.50070208e-04  1.00000000e+00]]
Homography Matrix Image 1, Image 2, Image 3, Image 4:  [[ 5.74778554e-01 -4.95608695e-02  3.24118864e+02]
 [-1.94921213e-01  8.88942288e-01  4.08329389e+01]
 [-8.15421820e-04 -7.69665616e-05  1.00000000e+00]]
```

Figure 13. Homographic Matrix used for Image Warping

Conclusions

- This was a challenging project.
- Prior knowledge of Image Processing would have been an advantage.
- A review of MATLAB and Python was done.
- Implementation of algorithm such as RANSAC, SIFT was done.
- The accuracy of clicking on the objects on the image can change the accuracy by some degree.

References

1. https://www.kaggle.com/code/victorvianaom/sift-and-ransac-algorithms/log
2. https://learnopencv.com/image-alignment-feature-based-using-opencv-c-python/
3. https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html
4. https://docs.opencv.org/4.x/dd/d1a/group__imgproc__feature.html
5. https://www.mathworks.com/matlabcentral/fileexchange/45719-harris-corner-detector
6. http://www.ipol.im/pub/art/2014/82/article.pdf