University at Buffalo
Department of Computer Science and Engineering
CSE 473/573 - Computer Vision and Image Processing

Spring 2023

Project #2
Due Date: 4/4/23, 11:59PM

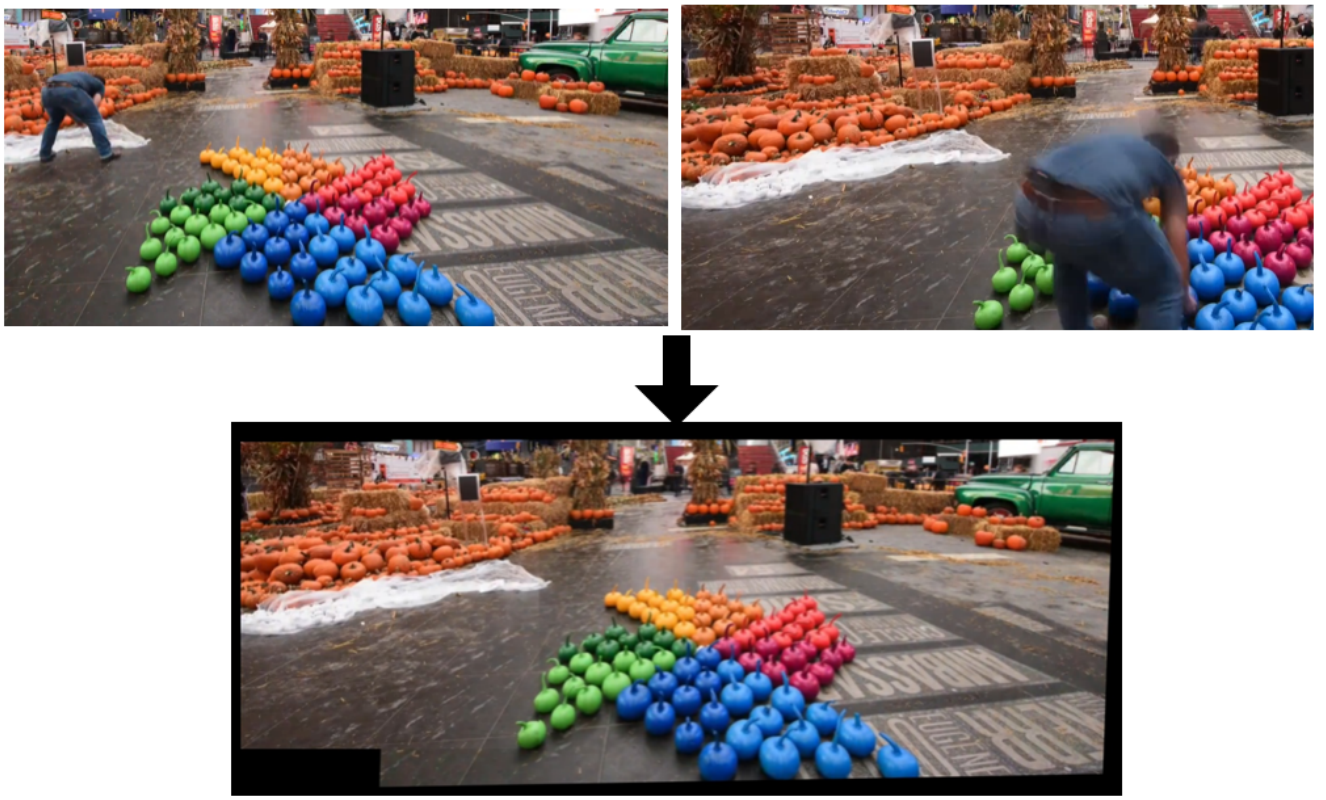

Figure 1: Example of background stitching.

1 Background Stitching (50pts)

The goal of this task is to experiment with image stitching methods. Two images may have the same background but different foreground. For example, a moving person may be moving in the

scene. You need to stitch the two images into one image eliminating foreground objects that move in the scene. You may assume that the background region is behind the foreground object is visible in two or more images. See the example in Fig 1. The shape of the output image after stitching might be irregular since you are not allowed to crop either of the transformed images. There are no restrictions regarding the method you use to stitch photos into one photo.
Expected Steps:

- Extract features from each image.
- Match features and use matches to determine if there is overlap between given pairs of images.
- Compute the homography between the overlapping pairs as needed.
- Transform the images and stitch the two images into one mosaic, eliminating the foreground as described above.
- Save the resulting mosaic according to the instructions specified in the script. (50 points)

2 Image Panorama (50pts)

This task aims to stitch multiple images into one panoramic photo. As shown in Fig. 2, the given images might be non-overlapping or multiply-overlapped (overlapping two or more other images). The shape of the output image after stitching might be irregular since you are not allowed to crop either of the transformed images. There are no restrictions regarding the method you use to stitch photos into a panoramic photo. For this project, you can assume the following:

- Your code will need to be able to stitch together four or more images and you will not know that in advance.
- You can assume that IF an image is to be part of the panorama, it will overlap at least one other image and by at least 20%.
- Images that do not overlap with any other image can be ignored.
- Images can overlap with multiple images.
- Although the Figure below shows horizontal panoramas, your five images can be stitched together in any way.
- You are only expected to produce one overall image.
- While some of the most modern techniques may use a spherical projection for better panoramas, you are free to assume that basic 2D planer transformations are sufficient for this project.

Steps:

- Extract features for each image, match features, and determine the spatial overlaps of the images automatically. For instance, among four images, if image1 overlaps with image3, image4, then your overlap array should be $\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$. Follow the instructions in the provided script and save the overlap result as a $N * N$ one-hot array in the JSON file. (20 points)
- Conduct image transformation and stitch all into one panoramic photo. Save the photo as the instructions specified in the script. (30 points)

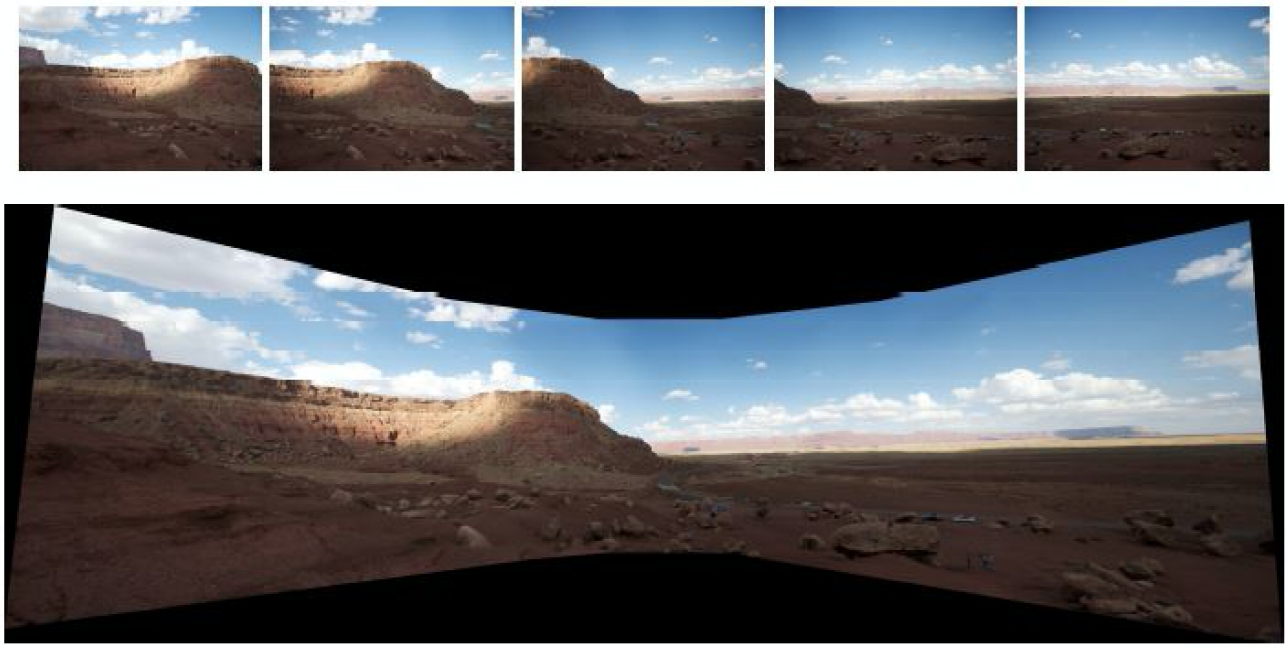

Figure 2: Example of image panoramas.

3 Image Panorama Example (10pts Bonus)

Find something that is Buffalo or UB-related (suggested by not required), photograph it and demonstrate that your code works on regions where at least four images overlap. Set the number of your images N, name your input images as specified in the script, store your images, run the Image Panorama functions with your images.

4 Project Guidelines and Submission

- Please implement all your code in the file "stitching.py". Please do NOT make any changes to any file except "stitching.py".

- Please do NOT read/write any files in your code. The file reading and output part are already given in "task1.py" and "task2.py". In your implementation, you should ONLY use the input parameters of the function and give the output in a required data structure. The data structure is given in the code.
- Please do not use cv2, numpy, PIL, imageio for any image-related operations. We have provided the required helper functions. If you see any custom methods being useful for your implementation, then please feel free to implement them with PyTorch.
- You can only use the given libraries provided in the code. You can not make any new imports.
- Unlimited number of submissions is allowed and only the latest submission will be used for grading.
- To submit your code and result, Please run "pack submission.sh " to pack your code and result into a zip file. You can find the command line in "README.md" Note that when packing your submission, the script would run your code before packing. The resulting zip file is the only file you need to submit.
- The packed submission file should be named "submission YourUBITName.zip", and it should contain 5 files, named "task2.json", "stitching.py" "bonus.json", and images folder and outputs folder. If not, there is something wrong with your code/filename, please go back and check.
- For code raising "RuntimeError", the grade will be ZERO for the project if it can not be corrected. Not following the name rules "UBID.zip" will result in a reduction in grade.
- You have up to 3 late days for all the project/homework submission throughout the semester without points deducted.
- You may NOT copy code from other projects on the internet.
- Identical code will be treated as plagiarism. Please work it out independently. Checks will be done for AI violations. If your submission is evaluated as plagiarism, your grade will be 0 for that task.