COL780 Assignment 2
Creating Panoramas using Image Stitching

Ananya Mathur : 2020CS50416

March 2024

1 Directory Structure

```
2020CS50416.zip
├── main.py
├── stitcher.py
├── homography.py
├── warp.py
├── match.py
├── pysift.py
├── processvideo.py
├── report.pdf
└── results
```

2 Approach

2.1 Image Preprocessing

I transformed all images to a cylindrical projection before stitching to make the view wide angle and image stitching more uniform and easier.

2.2 Image Registration

The stitch function takes two images (base and sec) and stitches them to return output. For image registration, keypoints as well as their matches for both images are calculated using the get_matches function. Then, the invoke_RANSAC function is called which takes the keypoints of both images as well as their matches and returns the homography matrix between the two images. It internally calls the RANSAC function which implements the RANSAC algorithm (for 1000 iterations and threshold value 4) and computes homography between the images.

2.3 Panorama Creation from Images

Finally, I invoke the compute_newsize function which finds the size of the new frame of stitched images and updates the homography matrix accordingly. Then I call the warp function which places the images upon one another. To make warping more robust, I utilise the mask of both images as well using some bitwise operations between warped images and(or) image masks.

2.4 Video Processing

For panorama creation from videos, the capture_frames_at_intervals function in process_video.py captures frames equal to the duration of the video (in s). After thorough testing I realised that the number of frames captured in this way is sufficient to create panoramic view of the entire video. Then the usual steps for panorama creation of images are performed on the captured frames.

2.5 Blending

I tried implementing a few blending techniques including alpha and Laplacian, however for my case better results were coming with warping without blend and thus I did not perform blending during warping of images.

3 Challenges faced

- Implementing every single algorithm in the pipeline manually was extremely cumbersome as there are numerous optimisations that the inbuilt algorithms of opencv perform which cannot be replicated.
- Since the manually written functions were inefficient, making even a tiny change resulted in 10-15 minutes of wait to see its effect on output.
- Some images have a very high number of features leading to SIFT (manual implementation, hence slower) taking a very long time, close to 35-40 minutes for a pair of image stitching.
- Of the given 6 images to generate a panorama few were redundant and were giving no extra information about the scene thus increasing time for code to run.

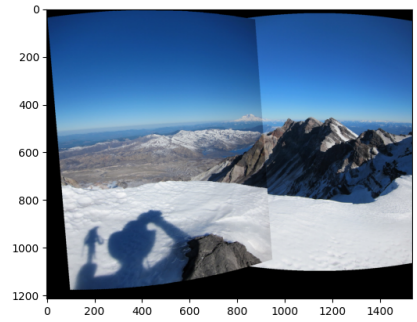
4 References

- I referred to `https://github.com/rmislam/PythonSIFT` for the implementation of SIFT.
- For some other implementations, I also referred to
  - `https://github.com/KEDIARAHUL135/PanoramaStitchingP2`
  - `https://github.com/WillBrennan/ImageStitching`
  - `https://github.com/fidansamet/image-stitching`
  - `https://github.com/nirmal-25/Image-Panorama-Stitching`
  - `https://github.com/Avinash793/panoramic-image-stitching`
- I also referred to the following paper for my image stitiching algorithm :
  Brown, M., Lowe, D.G. Automatic Panoramic Image Stitching using Invariant Features. Int J Comput Vision 74, 59–73 (2007). https://doi.org/10.1007/s11263-006-0002-3

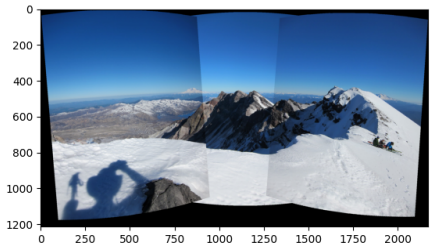
5 Results

(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4 (e) Image 5 (f) Image 6

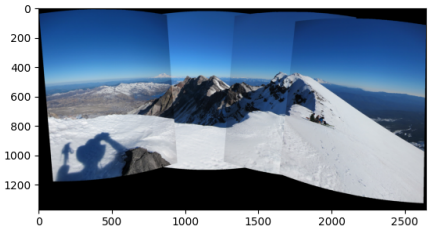
Figure 1: Mountain



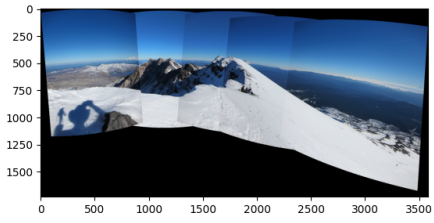
(a) Stage 1



(b) Stage 2



(c) Stage 3



(d) Stage 4

Figure 2: Stages of Image Stitching

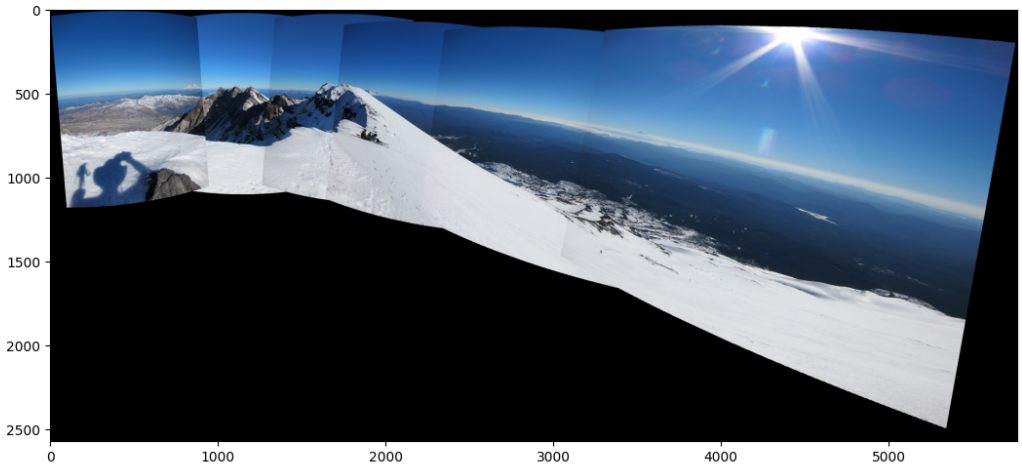


Figure 3: Panorama Created

(a) Image 1

(b) Image 2

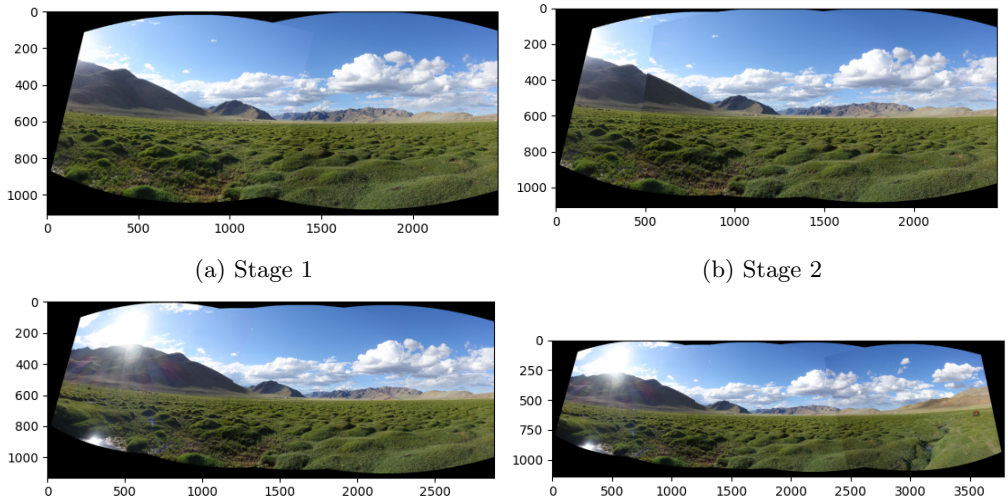
(c) Image 3

(d) Image 4

(e) Image 5

(f) Image 6

Figure 4: Field



(a) Stage 1

(b) Stage 2

(c) Stage 3

(d) Stage 4

Figure 5: Stages of Image Stitching

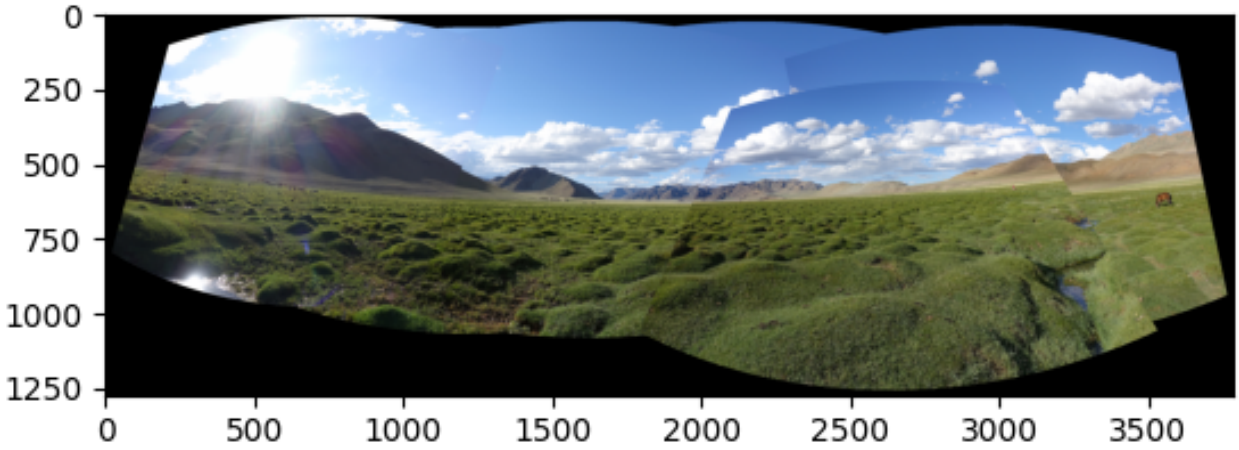
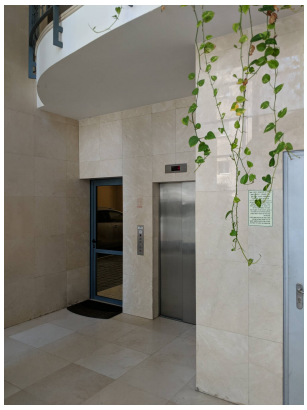


Figure 6: Panorama Created

(a) Image 1

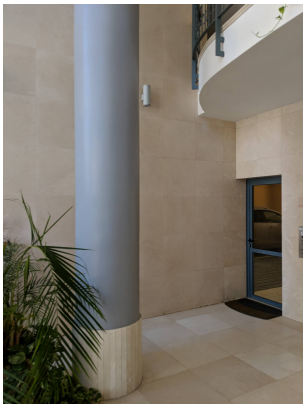


(b) Image 2

(c) Image 3

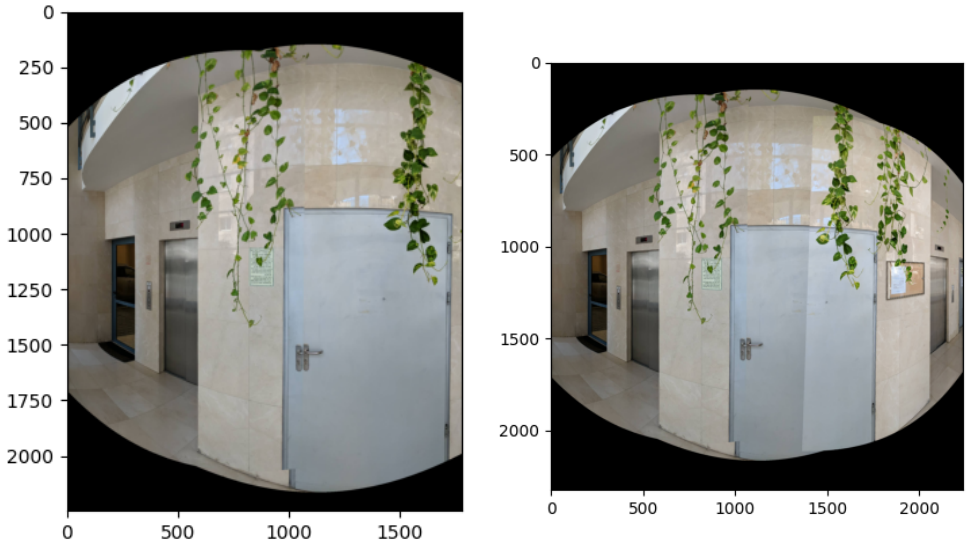
(d) Image 4

(e) Image 5



(f) Image 6

Figure 7: Office



(a) Image 1 (b) Image 2

Figure 8: Stages of Image Stitching: After stitching 3 major images together addition of 4^{th} image caused smudging of entire panorama so I stopped at this stage

(a) Image 1 (b) Image 2 (c) Image 3

Figure 9: Images extracted from Video 1

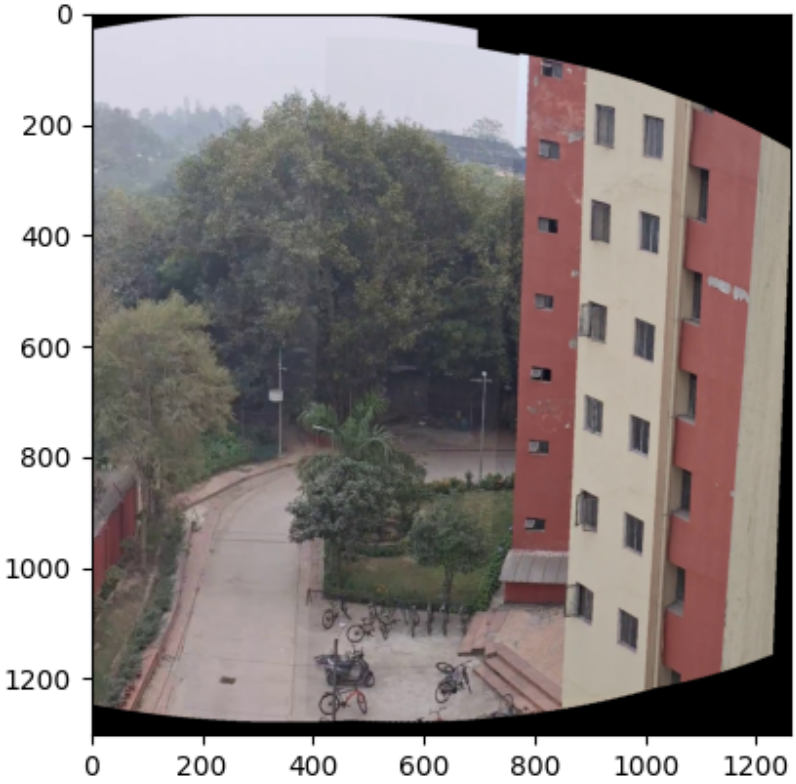


Figure 10: Panorama Created

(a) Image 1 (b) Image 2 (c) Image 3

Figure 11: Images extracted from Video 3

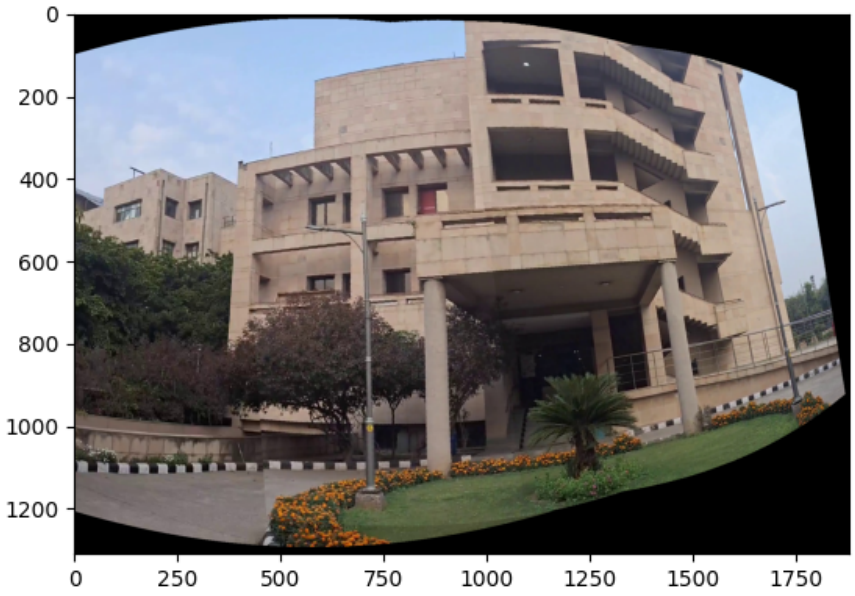
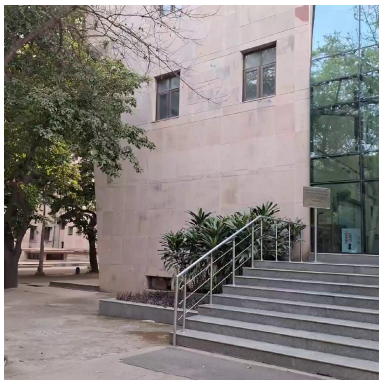


Figure 12: Panorama Created



(a) Image 1

(b) Image 2

(c) Image 3

(d) Image 3

Figure 13: Images extracted from Video 2

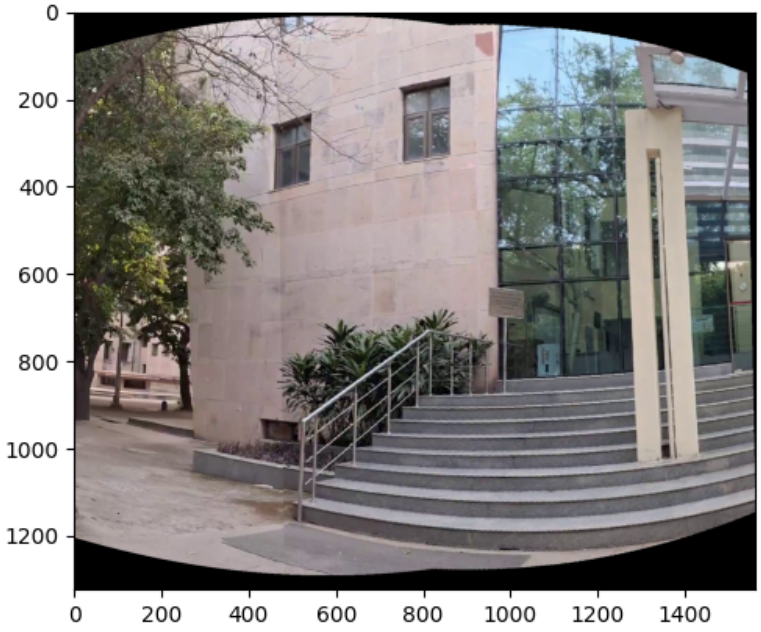


Figure 14: Stages of Image Stitching: After stitching 2 images together addition of 3^{rd} image caused smudging of entire panorama so I stopped at this stage. This is probably due to the large presence of glass each part of which is identical to another and thus keypoint matching does not perform too well