INDIAN INSTITUTE OF TECHNOLOGY, DELHI

REPORT

Assignment 2: Corner Detection and Panorama Stitching

ARNAV TULI | ENTRY NO. 2019CS10424

Course - COL780 | Prof. Anurag Mittal

Due March 7, 2023

Contents

Chapter 1

Corner Detection and Panorama Stitching

1.1 Introduction

My entry number ends with $4 \equiv 0 \pmod 2$. So, I have implemented Harris Corner Detector to detect interest points (corners) in the image. The algorithm used is same as mentioned in the paper by Harris and Stephens (1988). For panorama stitching, I first matched interest points between two images using proximity and sum of squared differences (SSD) approach. After matching, I found the affine transformation between the two sets of points and warped one on the other to get an image stitch.

Note: OpenCV's estimateAffine2D() (RANSAC) and warpAffine() methods are used to find the transformation between two images, and warp one onto the other respectively.

Following datasets are used for generating panorama images (*New Dataset*):

- 1 (New Hostel Building)
- 2 (IIT Hospital)
- 3 (SIT Building)
- 4 (A Statue)
- 5 (David Warner DC)

The generated panoramas can be found with their respective names here.

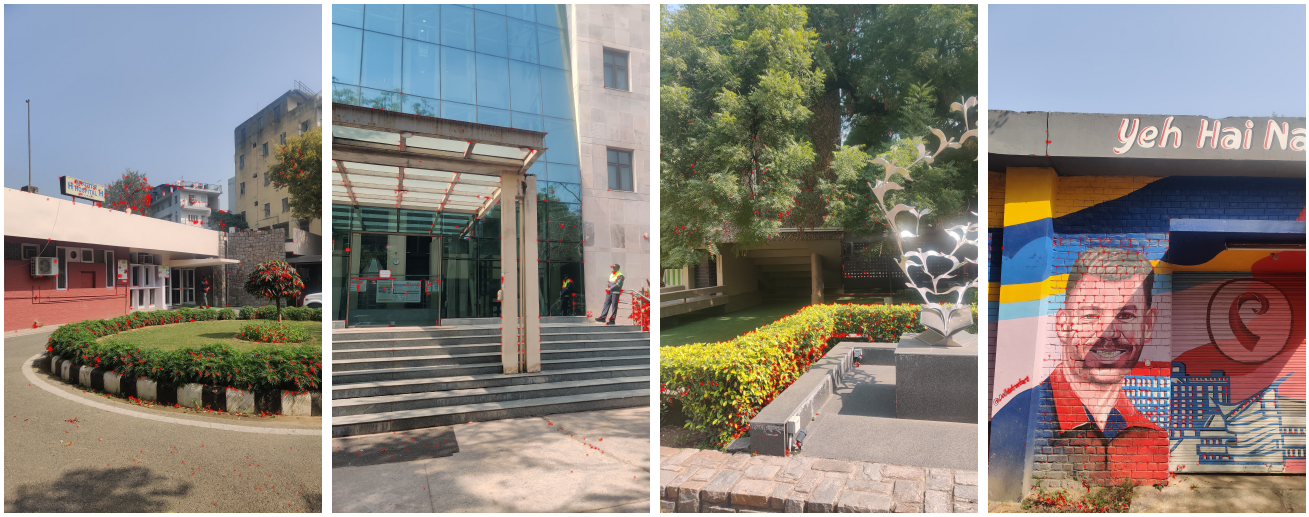
1.2 Results

The solution has two major components: Harris corner detector (HCD) and image stitcher (IS). The following parameters are used for these components:

- Corner window size (HCD): 5
- Corner window weight (HCD): Gaussian
- Harris response parameter (HCD): 0.04
- Harris response threshold fraction (HCD): 0.01 (1% of maximum response as cut-off)
- Harris response threshold count (HCD): 3000 (only top-3000 of the remaining points (after fraction cut-off) are taken as corners)
- Non-maximum suppression window size (HCD): 5
- Proximity window size (IS): 601
- Neighbourhood window size (IS): 5 (used in SSD calculation)
- Number of corners matched to get affine transform (IS): 100
- RNG seed (IS): 1 (used in sampling 100 points)
- Cropping tolerance: 10 pixels (to avoid *stitch* lines in the cropped image after warping)

1.2.1 Harris Corners (Top-3000 above 1% of maximum)

Figure 1.1: New Hostel Building (Image 0)



(a) (b) (c) (d)

Figure 1.2: (a) IIT Hospital, (b) SIT Building, (c) A Statue and (d) David Warner DC (all Image 0)

1.2.2 Panoramas

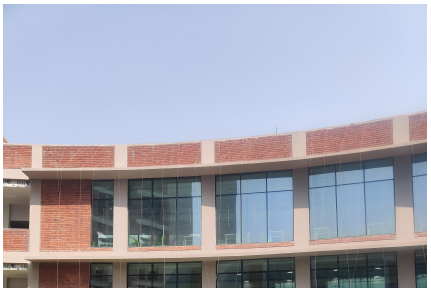
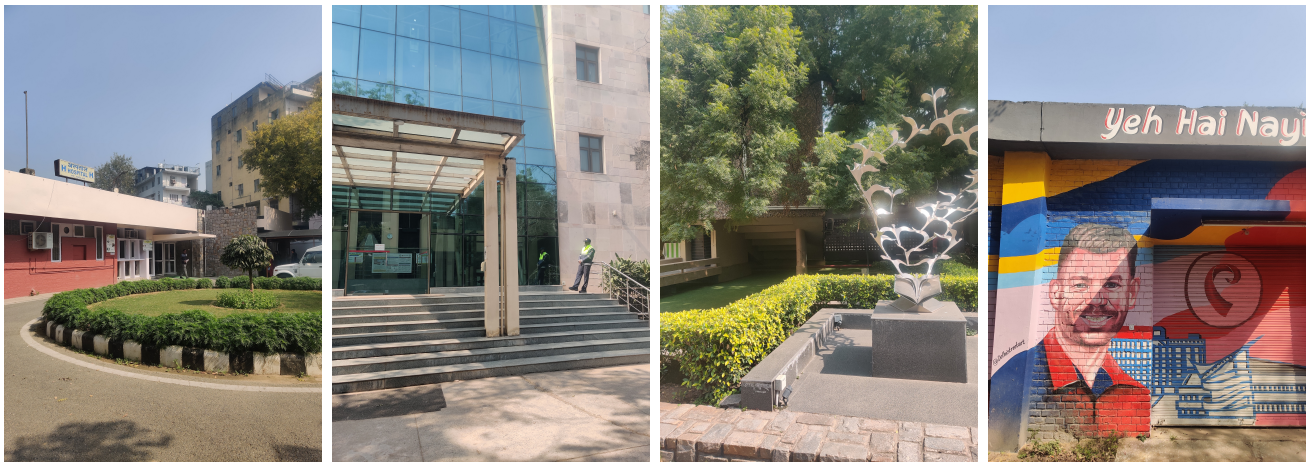
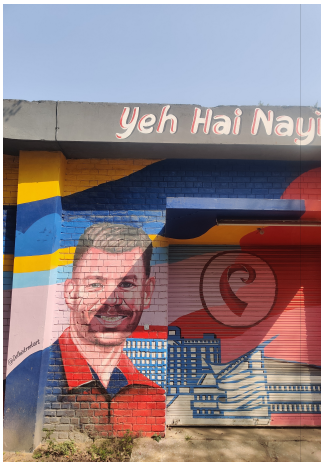


Figure 1.3: New Hostel Building



(a) (b) (c) (d)

Figure 1.4: (a) IIT Hospital, (b) SIT Building, (c) A Statue and (d) David Warner DC



(a)

Figure 1.5: David Warner DC but with *zero* crop tolerance (one *stitch* line is visible)

Chapter 2

Appendix: Code Overview

There are three code files `main.py`, `cornerDetector.py` and `imageStitcher.py`. The code assumes the following directory structure:

```
directory
├── Dataset/
│   ├── 1/
│   ├── 2/
│   ├── 3/
│   ├── 4/
│   └── 5/
├── main.py
├── cornerDetector.py
└── imageStitcher.py
```

`main.py` has support for the following command line arguments:

- `--dir`: Dataset directory (1, 2 etc.)
- `--p_radius`: Proximity radius for brute-force matching
- `--n_radius`: Neighbourhood radius for SSD calculation
- `--sample_count`: Number of points matched per stitch
- `--window_size`: Harris corner detector window size (for matrix calculation)
- `--k`: Harris response parameter ($det(H) - k * tr(H)$)
- `--thresh`: Harris response threshold fraction
- `--top_k`: Harris response threshold count (top k)
- `--nms_radius`: Radius used for non-maximal suppression
- `--seed`: RNG seed
- `--tolerance`: Crop tolerance
- `--gaussian`: Flag used to turn on gaussian weights in Harris matrix calculation

Run command: `python main.py` ⟨ `args` ⟩
Output: Final panorama image is saved in the current working `directory`.