Project 3 - ENPM 673
Spring 2022

Nishad Kulkarni
A. James Clark, School of Engineering
University of Maryland, College Park
UID - 117555431 Email: nkulkar2@umd.edu

***Abstract*—The following document is the report for Project 3 for the class ENPM 673, held in the Spring 2022 Semester. There were four sections in the Project pipeline, thus the sections are divided according to these sections.**

I. Introduction

Readers must first note that we will first see the directory structure of the submission in this section. The submission is titled `nkulkar2_prj3.zip`. The assignment codes were written in Python(3.8.10) and used the packages numpy(1.22.3), matplotlib(3.5.1), and opencv(4.5.5, opencv-contrib-python).

```
nkulkar2_prj2
├── code
│   └── p1.py
├── Datasets
├── Output
│   ├── curule_epipolar.png
│   ├── ...
│   └── pendulum_epipolar.png
├── nkulkar2_prj3_report.pdf
└── README.md
```

II. Calibration

For calibration between two successive poses, the 8 point algorithm is employed, the features for which are detected using `cv2.ORB` detector. For the dataset *'curule'*, 500 features were detected, for *'octagon'* 1500, and for the *'pendulum'* dataset, 5000 features were detected. The top 64% best matches are selected.

The detected features are used to estimate the Fundamental matrix '***F***' using both Least squares and RANSAC method. The latter is used for further calculations.

Next, the Essential Matrix '***E***' is calculated, which is then decomposed to get the '***T***' and '***R***' matrices.

III. Rectification

Moving on, the perspective transforms for the images are found using the inbuilt function `cv2.stereoRectifyUncalibrated()`. This gives us the Homographies of the left and right images.

It must be noted that the code assumes that images are passed in the order left to right. For ensuring this, the output of `glob` is sorted.

If the epipolar lines are horizontal, then it may be assumed that our warping is correct. This can be seen in the following images.

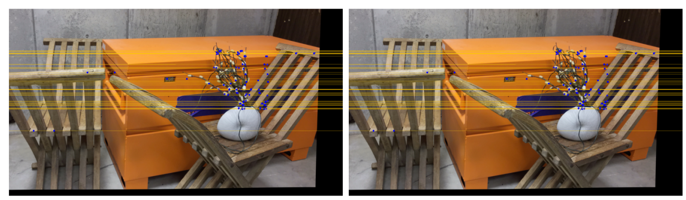

Fig. 1. Epipolar lines of 'curule' dataset

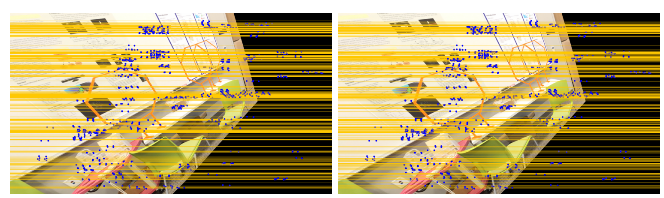

Fig. 2. Epipolar lines of 'octagon' dataset

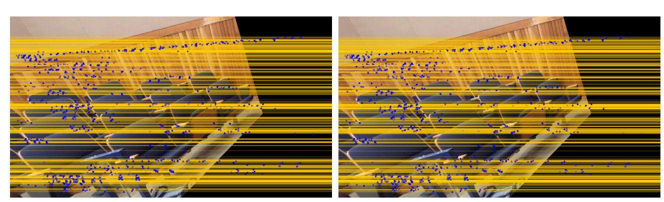

Fig. 3. Epipolar lines of 'epipolar' dataset

The number of features was progressively increased as the scenes progressivbely get indistinct in terms of feature variation. This is the reason, the pendulum dataset is the hardest to process, curule the easiest, and octagon lies in the middle.

IV. Correspondance

The Sum of Squared distances (SSD) is calculated between corrresponding features to generate a disparity map. This is done by chosing a square block of size 6 and sliding it over 90 consequent pixels. These values were selected by trial and error and were originally selected for the 'curule' dataset. The checks are performed cumulatively.

The information gathered by the SSD is then used to generate the Disparity map. These maps are passed through two denoising filters namely, the `cv2.fastNlMeansDenoisingColored()` and the `cv2.medianBlur()`. This gives a rather smooth output.

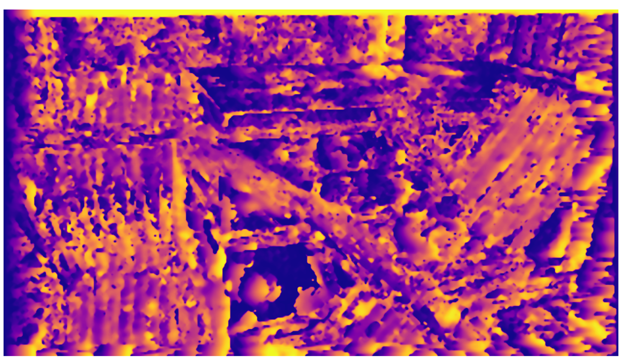

Fig. 4. Disparity map of 'curule' dataset

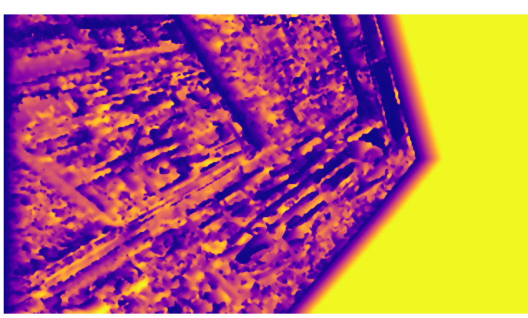

Fig. 5. Disparity map of 'octagon' dataset

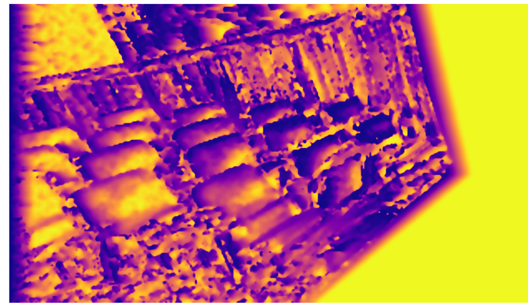

Fig. 6. Epipolar lines of 'pendulum' dataset

It may now be observed that the texture of the scene in view also makes a significant contribution on the calculation of the disparity. For example, even though the pendulum dataset is the hardest to process, it gives the best disparity map.

V. Depth

Finally, the depth is calculated using the camera intrinsics value '***baseline***' and '***f***'.

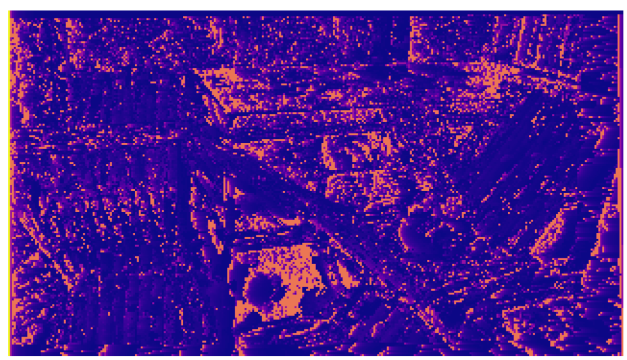

Fig. 7. Depth map of 'pendulum' dataset

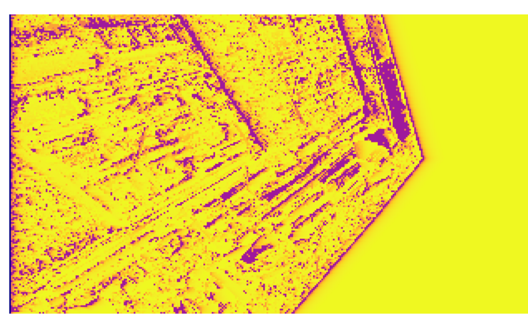

Fig. 8. Depth map of 'octagon' dataset

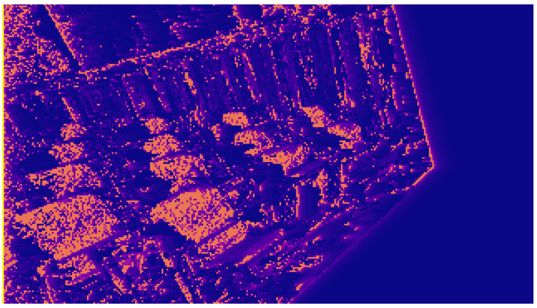

Fig. 9. Depth map of 'pendulum' dataset

Thus, the depth information is estimated using two subsequent images of 3 scenes.

It is obvious from the images that to calculate depth more accurately, more images would be required.