ENPM673 – Perception for Autonomous Robots – Spring 2021

Project 3

Srikumar Muralidharan, Samer Charifa

Due date: 26th April 2021, 11:59PM

Submission guidelines:

- This project is to be done and submitted individually.
- Your submission on ELMS/Canvas must be a zip file, following the naming convention **YourDirectoryID_proj3.zip**. If you email ID is abc@umd.edu or abc@terpmail.umd.edu, then your Directory ID is **abc**. Remember, this is your directory ID and NOT your UID.
- Please submit only the python script(s) you used to compute the results, the PDF report you generate for the project and a detailed README.md file.
- For each section of the project, explain briefly what you did, and describe any interesting problems you encountered and/or solutions you implemented.
- Include intermittent step outputs in your report.
- Please don't include the dataset provided to you and your video/image results in your submission, rather provide the link to your video/image output in the report.
- Disallowed function (In general, you are not allowed to use in-built functions unless you are instructed otherwise):
  – any inbuilt function that lets you compute the fundamental and essential matrices directly.
  – any other inbuilt function that computes the disparity or stereo correspondence.

Project Description

In this project, we are going to implement the concept of Stereo Vision. We will be given 3 different datasets, each of them contains 2 images of the same scenario but taken from two different camera angles. By comparing the information about a scene from 2 vantage points, we can obtain the 3D information by examining the relative positions of objects.

Dataset Preparation

Please use this link to download the 3 Datasets that you will try to implement stereo vision on, individually.

A brief explanation of the terms used in the ground truth files:

```
cam0,1:        camera matrices for the rectified views, in the form [f 0 cx; 0 f cy; 0 0 1], where
  f:           focal length in pixels
  cx, cy:      principal point  (note that cx differs between view 0 and 1)

doffs:         x-difference of principal points, doffs = cx1 - cx0

baseline:      camera baseline in mm

width, height: image size

ndisp:         a conservative bound on the number of disparity levels;
               the stereo algorithm MAY utilize this bound and search from d = 0 .. ndisp-1

isint:         whether the GT disparites only have integer precision (true for the older datasets;
               in this case submitted floating-point disparities are rounded to ints before evaluating)

vmin, vmax:    a tight bound on minimum and maximum disparities, used for color visualization;
               the stereo algorithm MAY NOT utilize this information

dyavg, dymax:  average and maximum absolute y-disparities, providing an indication of
               the calibration error present in the imperfect datasets.
```

Pipeline for creating a Stereo Vision System

1. Calibration (40 points)

- First, we need to compare the two images in each dataset and select a set of matching features. You **can** use any inbuilt function for feature matching. SIFT and corner detection methods are recommended feature matching techniques.
- Estimate the Fundamental matrix using the features obtained in the previous step. Refer to section 3.2.2 in this link to get an overall understanding of Fundamental matrix estimation. You can use inbuilt SVD function to solve for the fundamental matrix. Note that you have the choice of using RANSAC method or the straight least square method to estimate the fundamental matrix.
- Estimate Essential matrix E from the Fundamental matrix F by accounting for the calibration parameters. You should implement the functions to estimate the Essential matrix and also to recover the rotation/translational matrices.
- Decompose E into a translation T and rotation R.

2. Rectification (20 points)

- Apply perspective transformation to make sure that the epipolar lines are horizontal for both the images.
- You **can** use inbuilt functions for this purpose.
- Print H1 and the homography matrices for both left and right images that will rectify the images.
- Plot epipolar lines on both images along with feature points.

3. Correspondence (20 points)

- For each epipolar line, apply the matching windows concept (such as SSD or Cross correlation).
- Calculate Disparity
- Rescale the disparity to be from 0-255 and save the resulting image.
- You need to save the disparity as a gray scale **and** color image using heat map conversion.

4. Compute Depth Image (20 points)

- Using the disparity information obtained above, compute the depth information for each pixel image. The resulting depth image has the same dimensions of the disparity image, but it has depth information instead.
- You need to save the depth image as a gray scale **and color image** using heat map conversion.