AER 1515 - Perception for Robotics

Assignment 2

Submission guidelines: This assignment is due on Friday, November 20, 2020 at 11:59 pm. Include the following items in a `.zip` file and upload to Quercus:

- PDF of assignment solution write-up
- output file `P3_result.txt`
- documented Python that generated your images and `P3_result.txt`
- `README` detailing how to run the code and any dependencies the code has

Feature Point Detection and Correspondences

Feature point detection and matching is an important task in many robotics applications, such as image stitching, 3D reconstruction, visual odometry and object tracking. This assignment will cover feature point detection in images, matching features between stereo camera pairs, and filtering out incorrect correspondences using an outlier rejection algorithm. The rectified stereo image pairs are from the KITTI dataset. You are provided with 10 image pairs along with the ground truth depth map to validate your algorithm (the training set). Your code will be tested on the 5 provided image pairs without ground truth depth map (the test set).

Setting up your development environment

Starter code provided is for Python 3.5+. You will need the following Python packages for this assignment: OpenCV for working with image features, NumPy for working with arrays and Matplotlib for displaying images. For students who are not familiar with Python and OpenCV, there are good tutorials here (make sure the version number in the dropdown menu on the top-left matches your OpenCV version).

1 Feature Detection

Using a detector and a descriptor of your choice, detect 1000 keypoints and compute their descriptors in every image pair in both the training set and the test set. Present the left image with keypoints for each image pair in the test set. Briefly discuss any observations you make as to the appearance of unreliable keypoints (ones that may lead to difficulty in performing matching later in the assignment).

2 Feature Matching

Use the keypoints and their descriptors to find their correspondences between image pairs. Describe how you matched these features and present your feature matches for the images in the test set like the sample below. Given that the image pairs are rectified, your feature matches should follow an epipolar constraint (i.e. matches lie on a horizontal line). To evaluate your results, calculate disparity based on keypoint matches, then find the depth for the detected keypoints using camera calibration information. The ground truth depth map for the training set is given for you to evaluate your estimated depths. Note that the ground truth depth map is generated using LiDAR-based depth completion, so it does not cover the entire image. Pixels without a corresponding depth are labelled with a depth of 0. The ground truth depth map is with respect to the left camera frame. **Note:** the starter code includes functions to load the calibration for the stereo cameras. These functions return a class object with multiple camera calibration matrices, but you will only need to use the calibration information for the left camera (**p2**) and the right camera (**p3**).

Figure 1: Feature points matching

3 Outlier Rejection

In the previous section, we were able to apply an epipolar constraint because the images are rectified. This may not always be applicable. Perform outlier rejection on your original feature correspondences, prior to applying the epipolar constraint, using an algorithm of your choice (covered in class or from a paper). Tune your algorithm on the training set to achieve optimal results, and then test your algorithm on the test set. In your write-up, present the feature correspondences after outlier filtering (on the test set) and describe the algorithm you used, including any necessary equations. Briefly discuss the performance of your outlier rejection algorithm compared to using the epipolar constraint. Also, please present your feature matches on the test set in a *P3_result.txt* file with the same format as the *example.txt*. Code to generate this file is provided. Your results will be evaluated based on the correct number of feature matches and true positive rates, the RMSE of your depth estimates on the test set, and the explanation of your algorithm.

Marking guidelines:

- Code Clarity [10 pts]
- Feature Detection [20 pts]
- Feature Matching [20 pts]
- Outlier Rejection [50 pts]