Intelligent mobile robotics
Assignment 2

Sami Sellami

March 13, 2019

Problem1:

The task is related to line extraction, we have radar data form the scanning of a room (room 303 in the university) and we need to extract the line in the point cloud using different methods, namely; split and merge, line regression and hough transform

split and merge: This is a popular algorithm which uses recursive procedure of fitting and splitting, in our case we used the so called Iterative end point fit which simply connects the end points for line fitting, the splitting is done until the distance from the line to the farthest point is less than a predifined threshold, at the end we used the information of slopes to merge the line that are colinear enough (wrt a specific threshod)
Figure show the result of the line fitting:

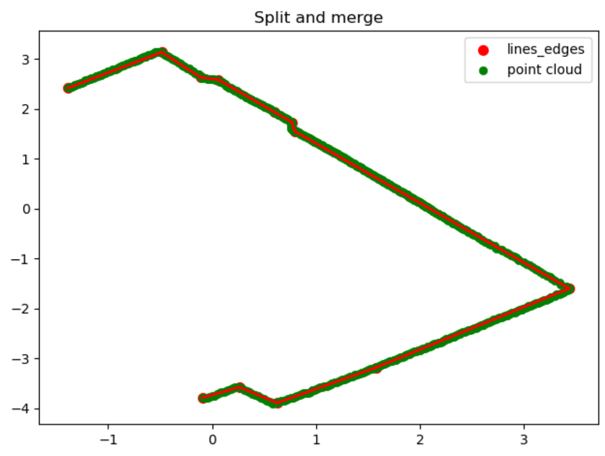

Figure 1: Split and merge line fitting

line regression: Line regression uses a sliding windows of size N_f, when the windows is sliding, a line is fitted to the points inside the window using least square method , also a fidelity array is computed wich contains the mahalalobis distance between the last two windows, when all the points have been analized, we merge the lines that have similarities in the fidelity array, meaning that distance between them is not too high
Figure show the result of the procedure

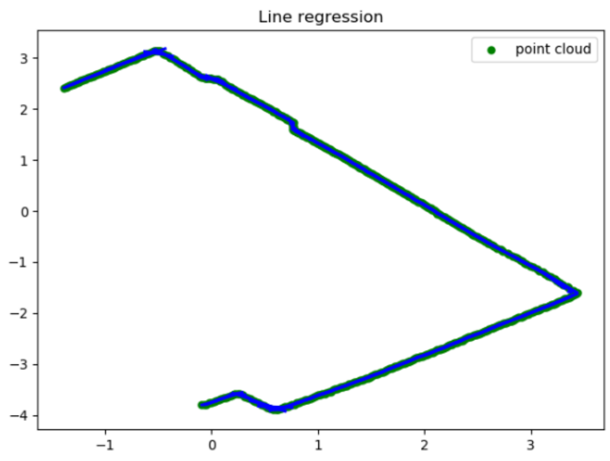

Figure 2: Line regression

Hough transform: Hough transfom maps image space into hough space it uses the proporty that a line in image space is a point in the hough space, OpenCV provide a function to compute the hough transform of an image (***cv2.HoughLines***), to uses the funciton, we needed to transform our point cloud into a matrix corresponding to an image. The result is show in the following image

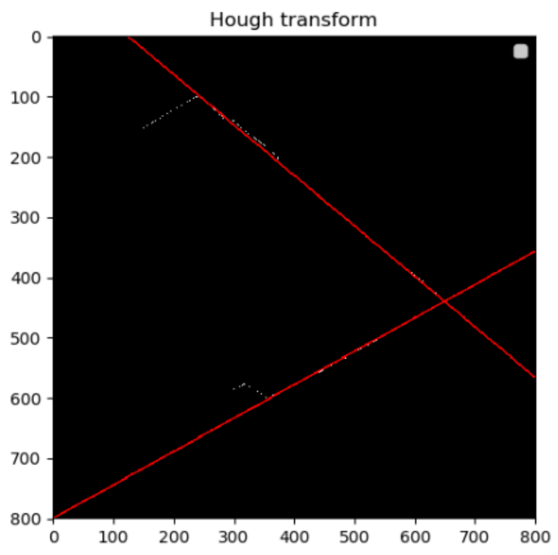

Figure 3: Hough transform

Checking the time exectution: The next diagram shows the time execution of each algorithm, we can see that line regression takes much time to compute, this is due to the fact that its two step algorithm and the computations are higher

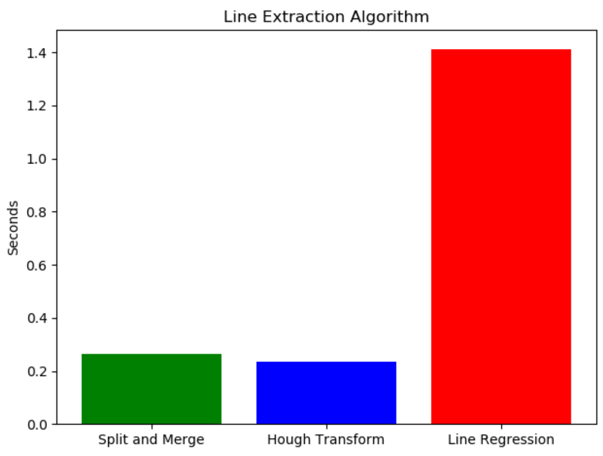

Figure 4: Time of execution

Problem2:

The task is to use **SIFT** feature desctiptor to detect and match features between five successive images, for this we located a bouding box around our object of interest (car moving in the driveway) and then we used OpenCV library to detect the keypoints in the box, the image below shows the result;

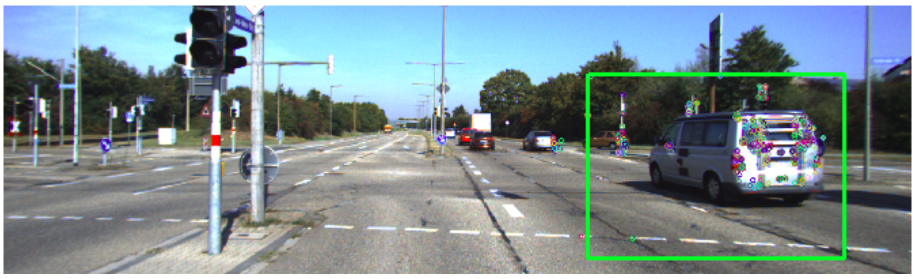

Figure 5: Keypoints detection

After that we implemented our feature descriptor which consist of of a 128 vector array of the histogram of orentation in 8 directions of the neighbouring region, the correspondance between the keypoints detected in two successive images is shown below:

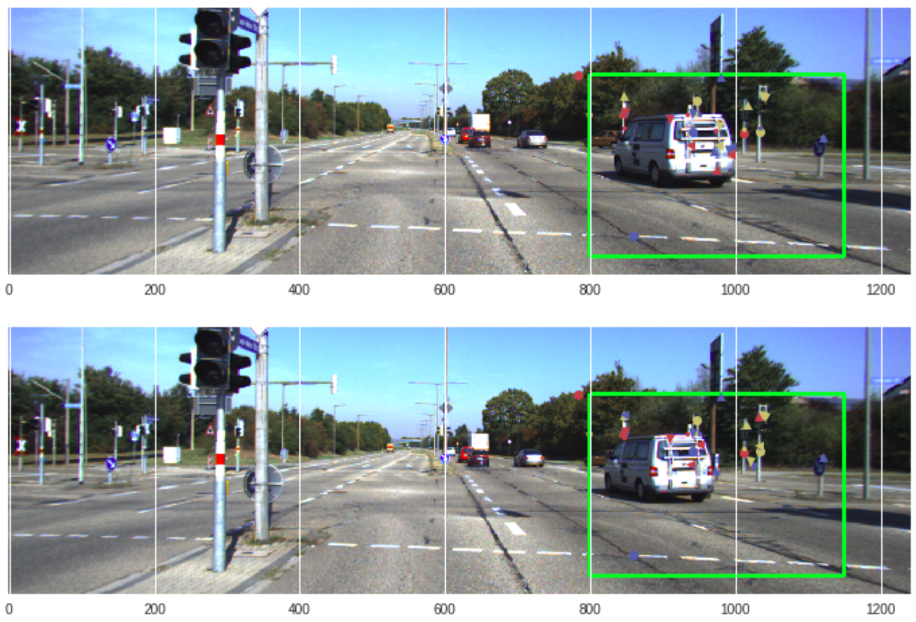

Figure 6: Keypoints detection and matching