Georgia Institute of Technology

CS 8803
Machine Learning for Robotics

LAB 2 Report

Author:
Bilal Ghader Ayush Shukla

September 4, 2018

1 Computation Time and Load

The typical values of computation time are significantly different for the two solvers. In order to be able to visualize this differences, we modified the code adding a snippet to print out the time needed for each solver. To begin with, we notice that the linear solver has a $0.003s$ average time while the non linear ceres solver has an average time of 0.09 to $0.13s$. Thus, in general the ceres solver is expected to be 10 to 100 times slower than the linear implementation. To be noted, these values are highly dependent on the environment and notably on the number of points the solver is processing during the iteration. The linear solver showcases less variation than the non-linear one which showcased around 50% variation in the time required for one iteration.

2 Precision

Both solvers have the same precision when it comes to the final solution. This is due to the fact that both solvers use the same cost function. However, the ceres non-linear solver tends to converge slower than the linear implementation (refer to section 1).
Both approaches are sensitive to the environment. When no obstacles are in front of the robot, the result equation of the plan is $z = -0.01x + 0.0y + 0.0$. However, when an obstacle is detected the precision of the result decreases, the result equation reaches $z = -0.18x + 0.02y - 0.09$. This is due to the bias introduced by the vertical points of the wall.

3 Robustify

Even with –robustify argument, we don't observe any noticeable difference in the output.

4 Solution

- One way to solve the problem we have is to filter out noisy points based on a threshold on the z value (maybe only take points which are within a particular threshold to the minimum observed z value).
- Another heuristic is to ignore the obstacles in calculations. In fact, obstacles can be thought of as points in the point cloud with exactly the same (x, y) values, but different z values. Thus we can filter for these values to keep only one of them.