MEEG 667-010: Homework Project

Patrick Geneva

May 28, 2019

1 Introduction

Simultaneous localization and mapping (SLAM) is trying to concurrently solve the state estimation problem while also building a structured map of the environment. This is challenging since both the environment and agent's motion through it is unknown and can only be perceived through sensors that measure the two. For this project, the problem is formulated in a graph-based setting where a graph is constructed to represent the measurements and states that the robot traveled through at different times. Here we consider the case that we have already compressed measurements into relative pose measurements and have reconstructed a set of initial guess states from these "odometry" measurements. As derived by Grisetti et al. [1], the problem can be simplified into repeatably solving a linear system which is built based on the measurement connections and their information.

2 Measurement Model

The measurements provided in the g2o format, see [2], are 2D relative position measurements. These have been originally reconstructed from 2D LiDAR scan alignment and thus also has a covariance matrix which fully correlates the orientation and position changes. The measurement can be written as a function of the two bounding poses:

$$
{}^{1}_{2}\mathbf{R} = {}^{G}_{1}\mathbf{R}^{\top}\,{}^{G}_{2}\mathbf{R} \tag{1}
$$

$$
{}^{1}\mathbf{p}_{2} = {}^{G}_{1}\mathbf{R}^{\top}({}^{G}\mathbf{p}_{2} - {}^{G}\mathbf{p}_{1}) \tag{2}
$$

or in the case of 2D rotations, we have the following:

$$
{}^{1}_{2}\theta = {}^{G}_{2}\theta - {}^{G}_{1}\theta \tag{3}
$$

$$
{}^{1}x_{2} = \cos({}^{G}_{1}\theta)({}^{G}x_{2} - {}^{G}x_{1}) + \sin({}^{G}_{1}\theta)({}^{G}y_{2} - {}^{G}y_{1}) \tag{4}
$$

$$
{}^{1}y_{2} = -\sin({}^{G}_{1}\theta)({}^{G}x_{2} - {}^{G}x_{1}) + \cos({}^{G}_{1}\theta)({}^{G}y_{2} - {}^{G}y_{1}) \tag{5}
$$

Given the above, we can find the Jacobians by taking the derivative in respect to each state variable. We will get two Jacobians, one in respect to the $\{1\}$ frame and the $\{2\}$ frame of reference. These can then directly be used in the derivation of Grisetti et al. [1] to solve for a delta that can update the current state estimate.

3 Experiments

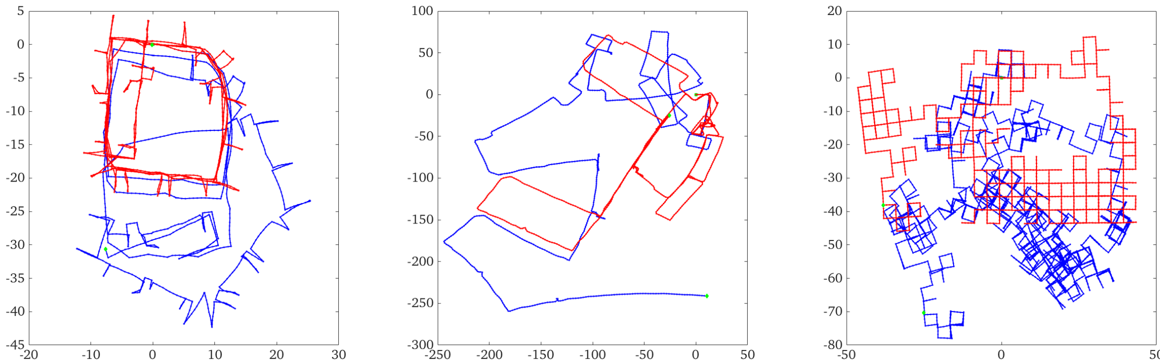

Figure 1: Results for the Intel (left), MIT (middle), and M3500 (right) datasets. The initial node guess are shown in blue while the optimized poses are shown in red. The M3500 took the longest to solve due to its size.

As shown by the generated trajectories in Figure 1, the problem is able to converge to a good estimate from very poor initial values. The Intel dataset was able to converge within a few optimization steps while the M3500 dataset took over 20 iterations to converge to low update values. In practice one would need to leverage the sparsity of the problem as compared to my use of using the matlab linear system solver which shouldn't exploit the sparsity.

References

[1] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. "A tutorial on graph-based SLAM". In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010), pp. 31–43.

[2] Luca Carlone and Andrea Censi. "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization". In: *IEEE Transactions on Robotics* 30.2 (2014), pp. 475–492.