Robostats Project Proposal
Sparse Planning Graphs for Information Driven Simultaneous Localization and Mapping

Vishnu Desaraju, John Yao, and Erik Nelson

The Robotics Institute, Carnegie Mellon University
{rajeswar, johnyao, enelson}@cmu.edu

November 6, 2014

1 Motivation

Guided by the desire to enable ultra-fast robot exploration, we plan to focus our project towards developing an information-based belief representation for rapid planning and localization. In general, the shortest trajectory over which a robot can greedily explore an environment is the trajectory formed by choosing actions which maximally reduce uncertainty in the environmental belief representation. One strategy to accomplish this is to choose control actions that maximize the expected mutual information between the robot's map at a future timestep and the robot's current map. This solution is one variant of a broad category of exploration strategies known as active Simultaneous Localization and Mapping (SLAM) [**?**].

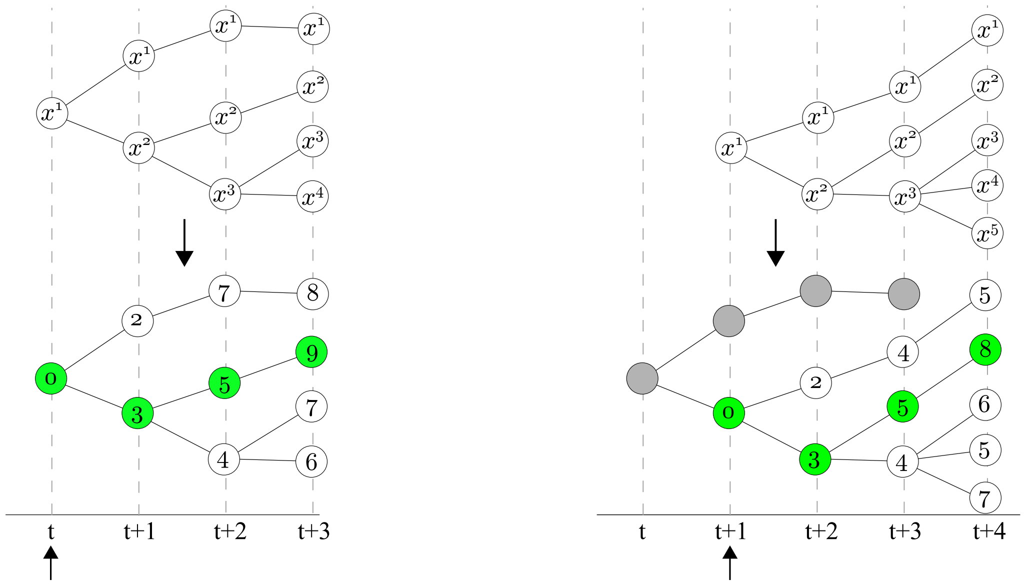
State of the art active SLAM implementations generally only compute actions over a one-step planning horizon due to the extremely high computational cost of determining expected mutual information over all potential future locations [**?**], [**?**]. **However, we believe that plans over much longer horizons can be generated** if these expected mutual information values can be computed once, cached, and updated efficiently based on new information. An even more efficient approach would utilize sparse planning graphs to limit expected mutual information calculations to a select few feasible future locations. Rapidly Exploring Random Trees (RRT), and lattice graphs are two examples of sparse planning graphs which decompose the reachable space into a sparse set of goal states based on a set of motion primitives for planning purposes.

We propose to investigate mutual information propagation through sparse planning graphs. **A substantial impact of this work** would be to develop a recursive formulation for efficiently and intelligently updating expected mutual information over a finite horizon as the robot navigates and obtains information through its sensors. **This project is deeply intertwined with the concepts of probability and uncertainty** due to the fact that at every update step, our algorithm must predict future measurements and future maps based on current sensor models and environmental belief.

2 Problem Definition

The active SLAM exploration problem can be framed as determining the control actions which guide a robot to a state that maximizes mutual information between its current and future maps. For this project, we will model the environment as an occupancy grid map, and represent the map as a conglomeration of cells: $m = \{m^i\}_{i=1}^N$. The probability that an individual cell is occupied at t is given by $p\left(m^i \mid x_{1:t}, z_{1:t}\right)$, where $x_{1:t}$ denotes the history of states of the vehicle, and $z_{1:t}$ denotes the history of range observations accumulated by the vehicle. Additionally we assume that cell occupancies are independent of one another: $p\left(m \mid x_{1:t}, z_{1:t}\right) = \prod_i p\left(m^i \mid x_{1:t}, z_{1:t}\right)$. For notational simplicity we write the map conditioned on random variables $x_{1:t}$ and $z_{1:t}$ as $p\left(m_t\right) \equiv p\left(m \mid x_{1:t}, z_{1:t}\right)$.

The optimal plan over a one step horizon will guide the robot to a state, x^*_{t+1}, in which the mutual information between



(a) (b)

Figure 1: Planning trees (top) and expected mutual information trees (bottom) at times t (a) and $t+1$ (b) over a 3-step horizon. As the robot moves forward, new nodes are added to the planning tree, the expected mutual information is updated based on the new sensor measurement, and the plan is refined. Superscripts on x denote the nodes that comprise each plan at a given timestamp, while the numbers in the mutual information tree represent an information gain "score" that we seek to maximize.

m_t and m_{t+1} is maximized.

$$
\begin{aligned}
x^*_{t+1} &= \underset{x_{t+1}}{\operatorname{argmax}}\ \mathrm{IG}\left[m_t; m_{t+1}\right] \\
&= \underset{x_{t+1}}{\operatorname{argmax}}\ \mathrm{H}\left[m_t\right] - \underset{z_{t+1}}{\mathbb{E}}\left[\mathrm{H}\left[m_{t+1}\right]\right] \\
&= \underset{x_{t+1}}{\operatorname{argmin}}\ \underset{z_{t+1}}{\mathbb{E}}\left[\mathrm{H}\left[m_{t+1}\right]\right]
\end{aligned} \tag{1}
$$

This is the formulation for a single step horizon, which is computed by sampling laser scans from the future state, updating the occupancy grid accordingly, and estimating the entropy of the resulting map. The goal of our project is to compute the optimal plan over multiple step horizons, which is simply an expansion of Eq. (**??**). We expect to be able to leverage recursive update rules to store and update expected information gain by propagating new information down a mutual information tree (Fig. **??**).

3 Metrics for Success

In order to reflect on our final result, we have defined the following success metrics:

1. Development of a recursive formulation for efficiently solving expected mutual information over several time steps.
2. Expected mutual information is similar to ground truth mutual information (evaluated in simulation).
3. Implementation on a quadrotor system demonstrates that the quadrotor navigates towards unexplored areas.
4. Implementation is real-time for a 2-step horizon, at minimum.

4 Required Resources

Software

- C++ / ROS Implementation of a laser-based SLAM pipeline (for collecting datasets)
- Eigen / Armadillo math libraries
- MATLAB for idea prototyping and plotting

Hardware

- Vicon motion capture system (for ground truth in datasets)
- Hokuyo UTM-30LX laser scanner
- Sensor-equipped quadrotor platform

5 Timeline

Date	**Task / Deliverable**
Oct 2	Written proposal
Oct 3 - 8	Derive bounds on convergence of the predicted information gain to the actual information gain
Oct 9	Initial presentation
Oct 9 - 14	Develop MATLAB simulation using a toy example (state-only, no map)
Oct 15 - 21	Implement strategy on quadrotor system with C++/ROS
Oct 22 - 29	Debug implementation
Oct 30 - Nov 5	Prepare midterm report
Nov 6	Midterm report
Nov 7 - Nov 13	Test on quadrotor-mounted laser scanner
Nov 14 - Nov 19	Prepare 3 / 4 report
Nov 20	3 / 4 report
Nov 21 - Nov 27	Refine and optimize implementation
Nov 28 - Dec 3	Prepare final presentation and video
Dec 4	Final presentation
Dec 5 - Dec 9	Prepare final report
Dec 10	Final report