Sparse Planning Graphs for Information Driven Exploration

Erik Nelson, Vishnu Desaraju, John Yao
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15217
{enelson, rajeswar, johnyao}@cmu.edu

***Abstract*—...**

I. Introduction

Exploration is a key capability that enables robotic vehicles to operate in unknown environments. In this project we develop an active perception policy for robotic exploration. Active perception exploration formulations choose control actions which optimize an information-theoretic objective function such as Shannon's mutual information or entropy [] over the robot's map, given a sensor measurement model. Other common exploration techniques, such as Frontier exploration [], use geometric reasoning to infer explorative paths. While these strategies work well in practice, they operate on a maximum likelihood estimate of the map, and apply heuristics to determine the most uncertain locations in the environment. In contrast, active perception strategies do not utilize geometric or maximum likelihood assumptions, and instead interpret the map as a binary random variable, choosing actions which directly minimize the random variable's uncertainty. Julian et al. prove that maximizing mutual information between a robot's map and expected future map naturally yields explorative behaviors [].

Active perception formulations seek to optimize information-theoretic objectives. While this optimization is real-time for short planning horizons, these metrics are often expensive to compute online, requiring double integration over possible future robot states and measurements, or Monte Carlo sampling from the distribution of measurements. In this project, we aim to develop an efficient active perception exploration strategy which evaluates the information-theoretic objective in a sparse set of poses across the configuration space. This strategy allows one to evaluate the objective function a limited number of times, while still generating paths that explore the space.

To achieve real-time active perception exploration, we use a Rapidly-Exploring Random Tree (RRT) to generate sets of dynamically feasible actions over a finite planning horizon []. RRT planners trade trajectory optimality for efficiency, allowing for evaluation of many potential future locations in the configuration space during a single planning step. In addition, RRT planners are anytime, allowing one to generate potential trajectories for a prespecified amount of time before evaluating the most optimal sampled trajectory. Our strategy evaluates each RRT leaf-node using the information-theoretic objective function, and stores the resulting reward in the tree. After planning for a specified amount of time, the maximum reward leaf-node is chosen as the next location to visit, and the RRT is traversed to generate a dynamically feasible trajectory to that location.

A recent work by Charrow et al. [] has proposed the Cauchy-Schwarz Quadratic Mutual Information (CSQMI) as an efficient information-theoretic objective function. CSQMI is theoretically well-motivated: it is derived from Renyi's Quadratic Entropy, a generalization of Shannon's entropy. However, in contrast to Shannon's mutual information (which is derived directly from Shannon's entropy), CSQMI is shown to have superior computational efficiency.

A. *Planning with RRT*

To make use of this information cost function to guide exploration, we consider a sampling-based planning approach that can evaluate the predicted information gain throughout the environment. In addition, we wish to use the occupancy grid that is being updated online (as described in Sect. **??**) to guide the vehicle around obstacles in the environment. The rapidly-exploring random tree (RRT) algorithm is well suited to planning paths through these types of large environments, and works as follows. The planner starts from the vehicle's current state and samples a point x in the environment. Using the occupancy grid, we can reject samples that lie in cells with a sufficiently high probability of being occupied. If the sample is valid, we find the closest node in the tree of paths (initially just the vehicle state), where closeness is measured in terms of Euclidean distance, and add a new edge to the tree connecting the sample point to the nearest node. Then a new sample is drawn and the process repeats to grow a tree of path segments through the environment. This tree-growing process terminates after a specified time, and the minimum cost path is returned.

Figure **??** shows snapshots of the system planning through the environment while updating the occupancy grid. The edges in the tree are smooth since they are generated by forward simulating the closed-loop vehicle dynamics toward the sample point, resulting in a variant of the RRT algorithm known as Closed-loop RRT (CL-RRT) [2]. This approach is traditionally used to ensure dynamic feasibility and dense collision checking. However, the forward simulation also means we have full state information for the system at the end of each segment. This allows us to evaluate the predicted information gain at that point and assign a corresponding cost to candidate trajectory.

To change CL-RRT from a goal-directed planner to an exploration-driven planner, we first define the sampling distribution to be a Gaussian centered about the root of the tree, with no bias toward any direction (unlike standard sampling-based planners that will sample the goal some small probability). We also define the cost of each branch segment to just be the information metric computed at its endpoint (as opposed to a more traditional setup where the cost is the total distance traveled from the root plus a cost to go based on an admissible heuristic, such as Euclidean distance to a goal). Finally, since there is no goal to guide the selection of the best branch from the tree, we simply select the branch with the minimum cost endpoint in the entire tree. This enables the planner to compute paths that aim to maximize the predicted information gain.

Algorithm 1 CL-RRT: Tree Expansion

```
Sample point x_sample from the environment State Identify nearest node N_near in tree
k ← 0
x̂(t+k) ← last state of N_near
while x̂(t+k) ∈ X_free(t+k) and x̂(t+k) has not reached x_sample do
    Compute reference input r̂(t+k) from x_sample
    Compute control input û(t+k) from control law
    Compute next state x̂(t+k+1) from propagation model
    k ← k+1
end while N ← r_final
for each feasible node N produced do
    Update cost estimates for N
    Add N to tree
end for
```

References

[1] Frederic Bourgault, Alexei A Makarenko, Stefan B Williams, Ben Grocholsky, and Hugh F Durrant-Whyte. Information based adaptive robotic exploration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 540–545. IEEE, 2002.

[2] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, Sepetember 2009.

[3] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Robotics: Science and Systems*, volume 2, 2005.

[4] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.