Risk Averse Motion Planning for a Mobile Robot

Neil MacMillan, River Allen, Dimitri Marinakis[1], Sue Whitesides[2]
Computational Geometry and Robotics Lab[3]
Department of Computer Science, University of Victoria
neilrqm@gmail.com, riverallen@gmail.com, dmarinak@gmail.com, sue@uvic.ca

Abstract— **Here we consider a pragmatic sample-based motion planning approach for a robot operating in a fixed, range-only beacon field. We define and calculate *entropy* values for *regions of interest* and provide a method for finding "safe," risk averse, low-entropy paths between these regions. We include an experimental, real-world assessment of the approach.**

I. INTRODUCTION

Given the widespread prevalence of wireless sensor networks for automated data acquisition and control purposes, many industrial environments will have *a priori*, statically deployed wireless devices that can be used by a mobile robot for navigation purposes.

Here we propose a pragmatic, sample-based motion planning approach that allows a mobile robot to exploit information provided by wireless devices already present in the environment. We seek to identify *safely navigable* regions and routes in the instrumented environment that enable the robot to successfully execute its tasks. By safely navigable regions, we mean areas where the robot's pose uncertainty is relatively low. We use the term *navigation graph* to refer to a set of paths between safely navigable regions. We assume only that the robot can: 1) determine a range estimate to beacons in the environment and 2) use odometry information to aid its pose estimate. For our initial studies we designed and built a custom range-based navigation system using a low-cost off-the-shelf commercial robot and wireless components.

Sampling based motion planning approaches based on the classic paper of Kavraki *et al.* [1] aim to build a road map consisting of collision free configurations in the configuration space of a robot moving in an obstacle-cluttered environment, with free configurations joined by edges that correspond to collision-free positions. A large body of literature elaborates this approach (see [2] ch. 5). The objective is primarily to find a collision free path. Our approach likewise precomputes a map; however, the objective is to find good paths, via randomly sampled intermediate locations, between regions where the robot is expected to travel. The "goodness" of a path is determined by a measure of robot localization ease and likelihood of traversing the path successfully.

Many graph model approaches to path planning use vertices to represent cells in a cell decomposition of configuration space or physical space [2]. If cell adjacencies are known, and methods for moving within a cell are available, then finding a path becomes a graph theoretic problem. Graph edges may be weighted, and shortest-path algorithms can be used. By contrast, our graph theoretic model does not decompose the physical space. Instead, the graph associates vertices with regions within which the robot can be localized accurately and to which the robot will often be commanded to travel. Edges in the graph are associated with reasonably reliable routes between the regions. Thus, the focus is not on finding a path, but rather, on finding what we call a *navigation graph*, *i.e.*, a reliable set of paths that connect regions of frequent operation for the robot.

II. NAVIGATION GRAPH CONSTRUCTION

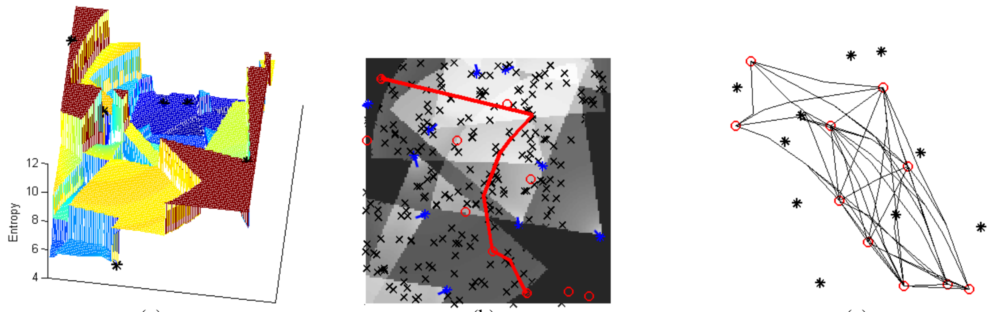
We consider the problem of constructing a *navigation graph* $G = (V, E)$ for a beacon-instrumented environment. The vertices $V = R \cup \Gamma$ of G represent regions of interest in the environment. Our regions of interest include both initially provided *operating regions* R in the environment that the robot will be expected to visit frequently during routine operations, and additional random locations Γ' (from which Γ is selected to form V) that allow the robot to navigate between the operating regions using straight-line transits. The edges $E = \{e_{ij}\}$ of G represent paths through the environment, weighted in proportion to the *cost* of the path between the two neighbouring vertices. Once a cost-weighted navigation graph is constructed for an environment, it can be used for fast, on-line path planning.

To construct a navigation graph we:

1) Estimate a discretized *entropy map* of the environment;
2) Sample candidate navigation locations (x, y) to form Γ';
3) Estimate the *navigation cost* (risk) between each pair of locations in $R \cup \Gamma'$;
4) Produce an initial navigation graph G' in which the vertices are $R \cup \Gamma'$;
5) For each pair of operating regions $r_i, r_j \in R$ find the lowest cost path p_{ij};
6) Construct a final navigation graph $G = R \cup \Gamma$ using $P = \bigcup p_{ij}$.

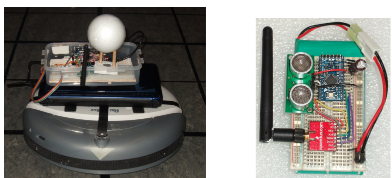
A fundamental input to our approach is a beacon model that specifies when the range to a beacon can typically be obtained and what error is associated with a range estimate. Additionally, an entropy map of the environment must be computed as a function of the beacon locations. This can be accomplished by precomputing an estimate for the entropy

[1]Research supported by an NSERC post-doctoral fellowship
[2]Research supported by NSERC and University of Victoria, BC
[3]CGAR Lab, http://cgar.cs.uvic.ca



(a) (b) (c)

Fig. 1. a) estimated entropy throughout the region; black stars depict beacon locations b) a path of minimum entropy; higher entropy is represented by a darker shade c) full navigation graph; black stars depict beacons, red circles operating regions, and black lines low entropy paths.



(a) (b)

Fig. 2. (a) iRobot Roomba instrumented with navigation components and (b) time-difference-of-arrival ranging beacon.

of the probability distribution for the robot's position at fixed locations throughout the environment; *e.g.* see [3].

Given the discretized *entropy map* Z of the environment, we find a set of paths that connect our operating regions R. Motivated by classical sample-based motion planners, *e.g.* [4], we uniformly sample N candidate points from our environment. We then assess paths through these points for robot navigation purposes. Similar to the approach of finding minima in a potential field, our approach tends to select paths through low entropy regions to improve navigation.

Given the N sampled points Γ' and the original operating regions R, we obtain a set of vertices $V' = \Gamma' \cup R$ and compute the pairwise *navigation cost* w_{ij} between each pair i, j of vertices; *i.e.* we assign a weight to the undirected edge e_{ij}. Our weight calculation takes as input: 1) the entropy that would be observed in the PDF of the robot if it were able to execute a straight line transit between the vertices i and j and 2) the distance between i and j:

$$w_{ij} \approx \int_{e_{ij}} f(H(x))dx$$

where $H(x)$ returns the value for the grid square in the entropy map Z that contains x, and f is some cost function.

Once the pairwise costs among all vertices in V' are computed, the lowest cost path p_{ij} is found between each pair of operating regions $r_i, r_j \in R$. The final navigation graph G consists of the union of all the edges and vertices in this collection of paths between all operating regions.

III. Preliminary Results

We have implemented a simulation framework to evaluate our navigation graph algorithm. We model the beacons as angular sections radiating from the beacon location with fixed range. Figure 1 shows the result of one simulation result from a trial with 10 randomly placed beacons and 10 randomly placed operating regions. Beacons are represented as blue stars, navigation locations are black crosses, and operating regions are red circles.

Experimentation on hardware (Figure 2) using our navigation system demonstrates the successful fusion of range data obtained from the beacons into our state estimation system. Due to the poor quality of the odometry data, range data are essential; without the range data, the error in the position estimate grows to the point that the robot cannot successfully navigate a standard office environment. Within a beacon-instrumented laboratory of 60 square meters, the robot could successfully execute predefined sequences of moves and avoid major obstacles.

IV. Conclusion and Open Problem

Here we have considered navigation issues for a robot carrying out routine tasks in an environment instrumented with range-only beacons. We have aimed at finding low-entropy routes among predefined regions in the environment. In future work we will incorporate into our current framework algorithms that use various environmental parameters for localizing purposes [5].

More generally, we ask how to build on the empirical success of sample-based algorithms to plan and execute high-quality paths for error-prone systems guiding robots through real-world environments.

References

[1] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[2] S. M. LaValle, *Planning algorithms*. Cambridge: Cambridge University Press, 2006. [Online]. Available: http://planning.cs.uiuc.edu

[3] N. MacMillan, R. Allen, D. Marinakis, and S. Whitesides, "Range-based navigation system for a mobile robot," in *In Proc. of Canadian Conference on Computer and Robot Vision*, St. John's, Canada, May 2011.

[4] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Springer-Verlag, 2008.

[5] D. Marinakis, N. MacMillan, R. Allen, and S. Whitesides, "Simultaneous localization and environmental mapping with a sensor network," in *Proc. of ICRA*, Shanghai, China, May 2011.