

Kartik V. Sastry

Advisor: Prof. Erik Verriest

Group Name: NavX (Team 2)

Review of Environmental Representation Techniques for Mobile Robot Path Planning

Introduction

The objective of robotic path planning is to find a feasible path for a mobile robot to move between two points in an unknown environment while optimizing travel time, energy expenditure or other application-relevant metrics. This paper is a review of representation techniques from recent literature. Hardware/software tradeoffs and requirements associated with the techniques, which affect the cost of implementation, are discussed when possible. There are several autonomous mobile robot platforms available commercially; however, the path planning algorithms and representations they use are not available in public.

To compute a path, it is necessary to represent the robot's physical environment in a computer. Any such representation must address at least four primary concerns – time, space, determinism and worldview. Other considerations such as maintainability, information adequacy, efficiency and scalability are also useful when evaluating a representation. This paper focuses on discrete-time representations since they are best suited for digital implementation. The building blocks, or basic methods, for representation are *Occupancy Grid Maps*, *Geometric Maps* and *Landmark-based Maps* [1]. Any combination of these is also a valid representation [2].

Occupancy Grid Maps

An Occupancy Grid Map (OGM) is a two-dimensional, discrete space, stochastic, global-observer model of a robot's environment. In this scheme, the world is viewed as a collection of cells; each cell is assigned a probability of occupancy that represents the presence of obstacles. The occupancy values are periodically updated using sensor data and Bayes' Rule. While scalability is a concern for OGM, the simplicity and maintainability of the representation make it an attractive choice.

An OGM can be stored as an array of floating point numbers or as a graph [2]. If the world has sparse obstacles, the graph model would be dense. Adjacency matrix representations are recommended for dense graphs [3]. The number of cells in the OGM grows quadratically with linear increase in the spatial resolution of the cells. Thus, it is necessary to select a suitable cell size to trade-off spatial resolution and memory usage. To mitigate this storage problem, [2] proposes the use of k-d tree structures to encode the array of cells. This decreases the space usage from $O(n^2)$ to $O(n)$. However, the time to access a cell increases from $O(1)$ to $O(\log n)$ [3]. Again, it is necessary to select the data structure to trade-off responsiveness of the robot with memory usage.

The Bayesian update procedure time is known to be exponential in the number of cells over which the sensor range spans [4]. This is prohibitive when spatial resolution is increased. Earlier work in OGM reduced this computation to $O(n)$ time by assuming independence of cells but this assumption introduced error [4]. Recently, an $O(n)$ approximation presented in [4] was shown to have a peak Kullback-Leibler divergence of 0.1 from the ideal Bayesian update, making it a closer approximation than previous work by a factor of 5.

Geometric Maps

A Geometric Map is a continuous space, stochastic, global-observer representation that eliminates the positional quantization error that is inherent in discrete space representations [5]. Current applications include using LIDAR point cloud data to synthesize a 3D model with Delaunay Triangulation. This can be completed in $O(n \log n)$ time and $O(n)$ space [6]. The representation can be updated for new obstacles using the C++ CGAL library, which performs 3D operations on polyhedra in $O(n \log n)$ time [7]. However, $O(n \log n)$ computations are still non-trivial for large sets of data and detract from real-time performance. [8]. Furthermore, the CGAL library and its dependencies require over 1 GB of storage [7]. Thus, geometric mapping is not intended to be implemented on embedded platforms [1], [8].

Landmark-Based Maps

A landmark-based map is a continuous space, deterministic, global-observer model of a robot's environment that is concerned with precisely locating obstacles. One way to represent the environment is with a meromorphic complex-valued function of a complex variable. A rational, open loop transfer function, $H(s)$, is an example of such a function from control theory [9]. The mobile robot and obstacles may be modeled as poles of $H(s)$. The destination location may be modeled with zeros of $H(s)$ [10]. Given a fixed representation, the computation of a path using Root Locus is equivalent to finding the 0° or 180° level curves of $\arg(H(s))$, which can be computed as in [9], [11]. A property of the root locus is that trajectories start at open loop poles (the robot) and end in open loop zeroes (the destination).

Perception Space Modelling: The Robot's Perspective

Perception-Space Modelling (PSM) is a memory-efficient, discrete space, stochastic representation that maintains the perspective of the robot [8]. This representation is useful for reactive or local planning, which focuses on computing collision-avoiding trajectories in real time. A depth imaging camera is used to collect 2D images of the robot environment. Both environmental obstacles and the robot are represented as objects in this image space. The motion controller works by comparing simulated images of the robot following course with real images collected from the depth camera. A multi-core processor with 2 GB RAM was used in to obtain real-time planning results under this representation in [8].

References

- [1] J. O. Wallgrun, "Robot Mapping," in *Hierarchical Voronoi Graphs: Spatial Representation and Reasoning for Mobile Robots*, Heidelberg, Springer-Verlag, 2010, pp. 11-43.
- [2] N. Correll, "Path Planning," in *Introduction to Autonomous Robots*, Magellan Scientific, 2016, pp. 73-75.
- [3] A. V. Aho and J. D. Ullman, *Foundations of Computer Science: C Edition*. New York, NY: W. H. Freeman, 1994.
- [4] R. S. Merali and T. D. Barfoot, "Optimizing Online Occupancy Grid Mapping to Capture the Residual Uncertainty," In Proc. 2014 IEEE International Conference on Robotics & Automation (ICRA), 2014, pp. 6070-6076.
- [5] S. M. LaValle, "Geometric Representations and Transformations," in *Planning Algorithms*, Cambridge University Press, 2006.
- [6] L. J. Guibas, D. E. Knuth and M. Sharir, "Randomized Incremental Construction of Delaunay and Voronoi Diagrams," *Algorithmica*, vol. 7, pp. 381-413, 1992.
- [7] L. Kettner, "3D Polyhedral Surface," in *CGAL User and Reference Manual*, CGAL Editorial Board, 2018.
- [8] J. S. Smith and P. Vela, "PiPS: Planning in perception space," In Proc. 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 6204-6209.
- [9] E. I. Verriest, *Lecture notes on Signals, Systems and Control, ECE3550*. Atlanta, GA: In preparation to publish, 2014.
- [10] N. s. P. Hyun, E. I. Verriest and P. A. Vela, "Optimal obstacle avoidance trajectory generation using the root locus principle," In Proc. 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 626-631.
- [11] D. Breda, S. Maset and R. Vermiglio, "An adaptive algorithm for efficient computation of level curves of surfaces," *Numerical Algorithms*, vol. 52, pp. 605-628, 2009.