Search-Based Path Planning with Homotopy Class Constraints

Subhrajit Bhattacharya
Department of Mechanical Engineering and Applied Mechanics
University of Pennsylvania
Philadelphia, PA 19104

Vijay Kumar
Department of Mechanical Engineering and Applied Mechanics
University of Pennsylvania
Philadelphia, PA 19104

Maxim Likhachev
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104 *

Introduction

Homotopy classes of trajectories, arising due to the presence of obstacles, are defined by the set of trajectories joining same start and end points which can be smoothly deformed into one another by gradual bending and stretching without colliding with obstacles.

Despite being mostly an uncharted research area, homotopy class constraints often appear in path planning problems. For example, in multi-agent planning problems (Zhang, Kumar, and Ostrowski 1998; Karabakal and Bean 1995), the trajectories often need to satisfy certain proximity or resource constraints or constraints arising due to tasks allocated to agents, which translates into restricting the solution trajectories to certain homotopy classes. In exploration and mapping problems (Bourgault et al. 2002), agents often need to plan trajectories based on their mission or part of the environment they are assigned for mapping or exploration, and hence restrict their trajectories to certain homotopy classes. Other examples of path planning with homotopy class constraints include predicting paths for dynamic entities and computing heuristics for path planning with dynamic constraints.

Motion planning with homotopy class constraints have been studied in the past using geometric approaches (Grigoriev and Slissenko 1998; Hershberger and Snoeyink 1991) and probabilistic road-map construction (Schmitzberger et al. 2002) techniques. Such techniques suffer from complexity of representation of homotopy classes and are not immediately integrable with standard graph search techniques. While comparing trajectories in different homotopy classes and finding the different homotopy classes in an environment is possible using such techniques, optimal path planning with homotopy class constraints is not achievable in an efficient way. The *abstract triangle graph* in triangulation-based path planning (Demyen and Buro 2006) technique can be used to represent various homotopy classes in an environment. However the construction of the graph and planning in it relies on the assumptions that the obstacles in the environment are polygonal, cost function is always the Euclidean length of the trajectories, and the algorithm does not scale well if there are many small obstacles.

In this paper we propose a compact way of representing homotopy classes of trajectories which is independent of the geometry, discretization, cost function or search algorithm. A full paper describing our method can be found at (Bhattacharya, Kumar, and Likhachev 2010). Our method is based on Complex Analysis and exploits the Cauchy Integral theorem to characterize homotopy classes. We show that this representation can be seamlessly weaved into the standard graph search techniques in arbitrarily discretized environments and impose the desired homotopy class constraints. It is to be noted that the method we propose is quite independent of the discretization scheme used for the environment, the nature of the cost function that needs to be optimized, or the search algorithm used. Hence this method can be incorporated into many existing planners, giving them the capability of imposing homotopy class constraints. Also, we can choose not to include certain obstacles if their sizes are too small for them to contribute towards creating homotopy classes.

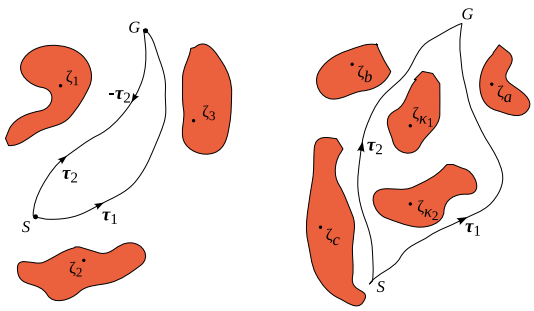
Overview

The basic principle of our proposed method is based on the *Residue Theorem* from *Complex Analysis*. We represent the two dimensional plane in which the robot's path is to be planned by the complex plane. We define a complex function called *Obstacle Marker Function* (Equation (1)) that is analytical everywhere in the complex plane, except for distinct points, ζ_i, which we call *representative points*, placed on the obstacles (Figure 1), where the function has *poles*.

$$\mathcal{F}(z) = \frac{f_0(z)}{(z-\zeta_1)(z-\zeta_2)\cdots(z-\zeta_N)} \qquad (1)$$

Upon making such a construction we note that according to the *Residue Theorem*, complex integrals of the function, $\int_\tau \mathcal{F}(z)dz$, from a fixed start to a fixed goal point have the same values when the paths along which the integral is performed lie in the same homotopy class, whereas they assume distinct values when the paths lie in different homotopy classes. We exploit this observation in order to characterize the homotopy classes. We call the complex integral value over a path the "*L-value*" of the path. We also observe

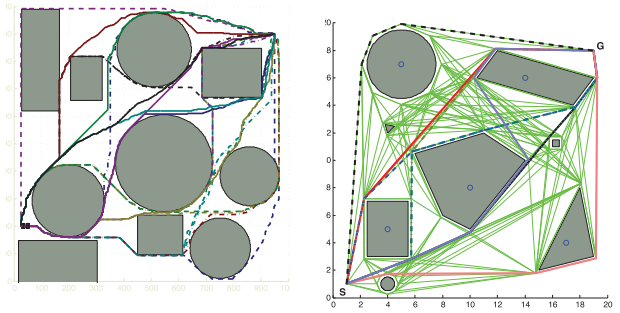
*We gratefully acknowledge support from ONR grant no. N00014-09-1-1052, NSF grant no. IIS-0427313, ARO grant no. W911NF-05-1-0219, ONR grants no. N00014-07-1-0829 and N00014-08-1-0696, and ARL grant no. W911NF-08-2-0004.



(a) In same Homotopy class, forming a closed contour

(b) In different Homotopy classes, enclosing obstacles

Figure 1: Two trajectories in same and different homotopy classes



(a) Exploring homotopy classes in a 1000 × 1000 uniformly discretized environment

(b) Exploring homotopy classes using a Visibility Graph

Figure 2: Exploring homotopy classes

that we have significant amount of freedom in choosing the *Obstacle Marker Function*, including the ability to choose only *significant* obstacles for creation of homotopy classes.

We discretize the environment or lay down a roadmap to construct a graph, in which we intend to search for the least cost path. Based on this graph we construct a *L-augmented graph* by augmenting each state with the L-value of the paths leading to it from the start of the state expansions. By doing this, a particular coordinate in space arrived using paths in different homotopy classes can be distinguished from each other as different states. Hence planning in this L-augmented graph lets one impose homotopy class constraints very efficiently just by specifying the goal state as a combination of goal coordinate and the desired (or a set of blocked) L-value(s) of the state. We have also shed light on the topology of the graph hence formed in order to obtain better insight. We have shown that the L-value of the edges of the graphs can be computed efficiently and accurately using an analytical expression when the edges are small (or discretized into small parts) (Bhattacharya, Kumar, and Likhachev 2010).

The most important advantage of using this approach as compared to more involved geometric approaches is that our method can be seamlessly woven in a standard graph search algorithm for planning least cost path in an environment. The search graph can be created by any form of discretization of the environment, as well as graphs like visibility graphs (Lozano-Pérez and Wesley 1979) can be employed. Planning in such a graph while keeping track of the L-value of trajectories joining nodes of the graphs to the start node lets one keep track of the homotopy classes during the expansion of the graph, and hence without too much additional computational or memory burden lets one impose homotopy class constraints.

Experimental Results

We have demonstrated the use of the proposed method in several different examples. The two primary kinds of graphs on which we have made our implementation are the uniformly discretized 8-connected grid-world (Figure 2(a)) and the visibility graph (Figure 2(b)). As cost function, we have demonstrated the use of Euclidean length as well as more complicated cost functions. We have also demonstrated how the method can be easily extended to environments with additional coordinates like time. Our experiments using very large environments demonstrate the efficiency and scalability of the method.

References

Bhattacharya, S.; Kumar, V.; and Likhachev, M. 2010. Search-based path planning with homotopy class constraints. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Bourgault, F.; Makarenko, A. A.; Williams, S. B.; Grocholsky, B.; and Durrant-Whyte, H. F. 2002. Information based adaptive robotic exploration. In *in Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, 540–545.

Demyen, D., and Buro, M. 2006. Efficient triangulation-based pathfinding. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, 942–947. AAAI Press.

Grigoriev, D., and Slissenko, A. 1998. Polytime algorithm for the shortest path in a homotopy class amidst semi-algebraic obstacles in the plane. In *ISSAC '98: Proceedings of the 1998 international symposium on Symbolic and algebraic computation*, 17–24. New York, NY, USA: ACM.

Hershberger, J., and Snoeyink, J. 1991. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl* 4:331–342.

Karabakal, N., and Bean, J. C. 1995. A multiplier adjustment method for multiple shortest path problem. Technical report, The University of Michigan.

Lozano-Pérez, T., and Wesley, M. A. 1979. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM* 22(10):560–570.

Schmitzberger, E.; Bouchet, J.; Dufaut, M.; Wolf, D.; and Husson, R. 2002. Capture of homotopy classes with probabilistic road map. In *International Conference on Intelligent Robots and Systems*, volume 3, 2317–2322.

Zhang, H.; Kumar, V.; and Ostrowski, J. 1998. Motion planning under uncertainty. In *IEEE International Conference on Robotics and Automation*. Leuven, Belgium: IEEE.