Robotics and Autonomous Systems 64 (2015) 44–58

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A comparison of homotopic path planning algorithms for robotic applications

Emili Hernandez *, Marc Carreras, Pere Ridao

Department of Computer Engineering, University of Girona, Girona 17071, Spain

HIGHLIGHTS

- We present three path planners to generate solutions that follow homotopy classes.
- Homotopy classes provide an added value to the path planning problem.
- Our method generates paths with the topology of the optimal solution much faster.
- We show extensive results in synthetic scenarios and on a bathymetric map.

ARTICLE INFO

Article history:
Received 23 March 2014
Received in revised form
26 August 2014
Accepted 28 October 2014
Available online 7 November 2014

Keywords:
Path planning
Homotopy
Marine robotics

ABSTRACT

This paper addresses the path planning problem for robotic applications using homotopy classes. These classes provide a topological description of how paths avoid obstacles, which is an added value to the path planning problem. Homotopy classes are generated and sorted according to a lower bound heuristic estimator using a method we developed. Then, the classes are used to constrain and guide path planning algorithms. Three different path planners are presented and compared: a graph-search algorithm called Homotopic A* (HA*), a probabilistic sample-based algorithm called Homotopic RRT (HRRT), and a bug-based algorithm called Homotopic Bug (HBug). Our method has been tested in simulation and in an underwater bathymetric map to compute the trajectory of an Autonomous Underwater Vehicle (AUV). A comparison with well-known path planning algorithms has also been included. Results show that our homotopic path planners improve the quality of the solutions of their respective non-homotopic versions with similar computation time while keeping the topological constraints.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Path planning is one of the most studied problems in robotics. The goal of the algorithms is to find a safe path to guide a robot in the Configuration Space (C-Space) [1]. Graph-based search algorithms look for the global optimal path in the C-Space which is obtained with an exhaustive exploration of the search space [2,3]. Most of these algorithms use a heuristic function to speed up the exploration process by first selecting the most promising states according to the heuristics [4]. On the other hand, probabilistic sample-based path planners, most of them based on the Rapidly-exploring Random Tree (RRT) [5], perform the exploration by growing a tree incrementally until the goal is reached. These algorithms do not perform an exhaustive exploration in C-Space, therefore, they provide a solution very quickly at the expense of its quality. Even bug-based algorithms [6], initially developed to perform reactive motion planning, can also be considered for path planning purposes. Their simple navigation strategies alternate "go straight" and "follow boundary" behaviors when an obstacle is detected are suitable to generate paths in a C-Space.

Although the aforementioned strategies generate paths towards a goal state taking into account the non-traversable states due to obstacles, none of them report the topological information of *how* they are avoided. Because of this, we use homotopy classes to provide a topological description of the paths.

A *homotopy class* is the set of all possible trajectories from a start to a goal point that avoids obstacles in the same manner [7,8]. Given two paths, they are said to be homotopic if one can be deformed into the other without encroaching any obstacle. Providing this extra layer of topological information is an added valued to the path planning problem, and knowing the homotopy class before the computation of the path itself has several advantages for robotic applications. First, it is possible to constrain the

* Correspondence to: Autonomous Systems, Digital Productivity and Services (DPAS) Flagship, CSIRO, Pullenvale, QLD 4069, Australia. Tel.: +61 7 33274062.
E-mail addresses: Emili.Hernandez@csiro.au (E. Hernandez), marcc@eia.udg.edu (M. Carreras), pere@eia.udg.edu (P. Ridao).

http://dx.doi.org/10.1016/j.robot.2014.10.021
0921-8890/© 2014 Elsevier B.V. All rights reserved.

path search into those areas of the search space that satisfy the homotopy class speeding up the path computation, which is very useful for low-powered navigation computers commonly used in autonomous robots. Second, if the homotopy class of the optimal path is known, it is possible to generate a good solution very quickly. Third, homotopy classes allow avoidance of obstacles in a specific manner which is of interest in surveillance, coverage and map exploration applications. For instance, homotopy classes can be selected to reinforce surveillance in specific areas, to avoid navigation patterns and to ensure that different areas of interest are explored from a topological point of view. They can also be used to constrain specific parts of the environment to avoid dangerous or unwanted areas in coverage applications. Multi-robot map exploration can also be improved using homotopy classes if each robot explores the environment following a different homotopy class.

The work presented in this paper is focused on, but not constrained to, path planning for Autonomous Underwater Vehicle (AUV) applications. Although the problem is naturally formulated in 3d, for certain scenarios of interest the problem can be simplified to 2d. For instance, a survey and/or search mission where the robot is supposed to fly at a fixed altitude, in bottom-following mode, while acquiring opto-acoustic imagery. Under these conditions, we can consider a 2d map parallel to the seafloor, where any area with a slope greater than a certain threshold behaves as a 2d obstacle. This is the case for applications like benthic habitat mapping, underwater archeology or cable/pipe inspection, being also the target for the system proposed.

In this paper we use homotopy classes to guide path planning algorithms topologically. We generate the homotopy classes that can be followed in any 2d workspace using the method presented in [9] and improved in [10]. Using the topological information, path planners do not have to explore the whole space but the space confined in a homotopy class. Then, using a lower bound heuristic estimator, the homotopy classes that most probably contain the lower cost solutions are known. Thus, the algorithm can generate some good solutions quickly. Three well known algorithms have been adapted to compute paths for each homotopy class generated, expanding on our preliminary work: the Homotopic A* (HA*) [11], the Homotopic RRT (HRRT) [10] and the Homotopic Bug (HBug) [12,13]. The completeness of our method is ensured because in case the goal is not reachable, no homotopy classes will exist and, consequently, no paths will be generated. The homotopy class of the global optimal path is guaranteed to be generated by the algorithm. Constraining the path search to the topology of the optimal solution allows generating good solutions with non-exhaustive search-based methods at a fraction of their computation time. This is an advantage in large environments because these algorithms can be very slow.

The paper also presents an extension of our previous results. The efficiency and scalability of our method has been tested in synthetic scenarios. Furthermore, the path planners have been compared with their non-homotopic versions and with themselves. Finally, the paper extends the results of an experiment under the scope of the TRIDENT European project (EU FP7 ICT-248497), in which we initially explored the generation of homotopic trajectories on a bathymetric map using the HA* [14]. The bathymetric map was generated by means of a Multibeam Profiling Sonar (MPS) and a Differential Global Positioning System (DGPS), which was used to generate a 2d Occupancy Grid Map (OGM) at the depth in which the AUV navigates. Our method has been applied to compute paths that the AUV must follow to reach a goal position avoiding obstacles in different manners. Extended analysis of the HA* execution and new results using the HRRT and the HBug on the bathymetric map are provided, leading to a discussion on the suitability of each path planner for robotic applications.

The paper is structured as follows: Section 2 details the relevant literature about homotopy classes for robotic applications; Section 3 describes the method to generate the homotopy classes from the workspace; Section 4 details the path planners that we propose which follow the homotopy classes previously generated; Section 5 reports the results; Section 6 presents the conclusions and future work.

2. Related work with homotopy classes

From the path planning point of view, the literature about homotopy classes can be classified in three different groups: the computation of the shortest homotopic path solutions that require a homotopy class as an input; those solutions that constrain the search topologically based on previous paths generated in the C-Space; and the automated generation of homotopy classes approaches that do not require generating any prior path in the C-Space.

2.1. The shortest homotopic path problem

Computing the shortest homotopic path using a non optimal path or homotopy class as an input has been studied since the 1980s. The *funnel algorithm* [15] is a well-known reference for triangulated environments. Although it was not explicitly designed to be used with homotopy classes, it generates the shortest path in a simple polygonal environment provided as an input which implicitly constrains it topologically. Other approaches generate the shortest homotopic solution for paths defined by closed curves [16] and adding constraints such as weighted regions [17].

In non-triangulated environments some authors proposed algorithms that efficiently compute the shortest homotopic path for a set of input paths that do not self-intersect [18]. This restriction was overcome in [19] and additional constraints such as path thickness were added in [20]. In the same vein, Grigoriev and Slissenko proposed a method to construct the shortest path for a given homotopy class that does not intersect in a scenario with semi-algebraic obstacles [21].

Despite the set of proposals that perform the computation efficiently for any possible path, the problem becomes intractable when there is no input homotopy class or path. Because of this, these solutions are difficult to apply in robot path planning. On the other side, some of these algorithms are suitable for an optimization process, when a path has been already generated by a path planner.

2.2. Constraining path search topologically

There is also a group of methods that compute the shortest path and then identify its homotopy class. This process can be done during the path search or once the whole path has been computed. Then, the topology of the path is encoded in order to restrict the next path search, which ensures that the new shortest path will have a different topology and hence, belong to another homotopy class. By repeating this process, it is possible to obtain the k-shortest paths of k-homotopy classes. In [22,23], two step methods were proposed: first, for each node of the environment they compute the shortest path that passes through it and then, the number of paths are pruned to generate the shortest path for each homotopy class. Other approaches generate a graph based on a Voronoi diagram computed from the areas to avoid which is then traversed using Depth-First Search (DFS)-based algorithms [24]. Bhattacharya et al. proposed a method to perform path planning with homotopy class constraints using graph-search algorithms. The graph that represents the environment is expanded with Complex Analysis values to characterize homotopy classes while computing the path [25]. This method has recently been extended to work on 3d environments and formulated for nd C-Spaces [26].

Download English Version:

https://daneshyari.com/en/article/413366

Download Persian Version:

https://daneshyari.com/article/413366

Daneshyari.com