Graph Search-Based Path Planning with moving Obstacles using Persistent Homology

Daniel Bencic

28.10.2022

1 Introduction

One of the crucial tasks for every autonomous robot is finding a collision free path through free space. Due to it's importance this problem has been studied for decades and is know as the path planning or the "piano movers" problem [31, 32]. While there exist polynomial time algorithms for very simple planning problems, the general path planning problem is PSPACE-complete [29]. The problem becomes computationally even harder if obstacles are not static anymore and the additional dimension of time is added to the problem. It has already been shown that the understanding of the topological strucuture of the configuration space can be used to improve algorithm performance by reducing the search space and solving the path planning problem on a more abstract level.

2 Problem Formulation

Preliminaries

This section introduces some fundamental concepts of topology and path planning [25, 24, 15, 28].

Homeomorphism If there exists a bijective function $f : X \mapsto Y$ such that f and it's inverse f^{-1} are continuous functions, then the topological spaces X and Y are homeomorphic. Homeomorphism implies that both X and Y share the same topological properties.

Configuration Space Given a metric space $\mathcal{W}$ with metric d and a rigid body $\mathcal{A}$, a rigid body transformation is a function $f : \mathcal{A} \mapsto \mathcal{W}$ such that $d(a_1, a_2) = d(f(a_1), f(a_2))$ and no reflection occurs. This holds for rotations and translations. Given $GL(n)$ by the set of all invertible $n \times n$ matrices, $O(n)$ is a subgroup of $GL(n)$ such that $QQ^T = Q^T Q = I$ for all $Q \in O(n)$. The subgroup $SO(n)$ of $O(n)$ which contains all rotations matrices meaning $det(P) = 1$ for all $P \in SE(n)$. Combining arbitrary rotations and translations gives the special euclidean group $SE(n)$ which is homeomorphic to $\mathbb{R}^n \times SO(n)$.

The configuration space for a rigid robot is therefore $\mathcal{C} \cong \mathbb{R}^2 \times \mathbb{S}^1$ and $\mathcal{C} \cong \mathbb{R}^3 \times \mathbb{RP}^3$ in 2D space and 3D space, respectively.

Path Given two points $x_0, x_1 \in X$, a path is a continous function $f : [0,1] \mapsto X$ such that $f(0) = x_0$ and $f(1) = x_1$. Paths for which $f(0) = f(1) = x_0$ are loops with basepoint x_0.

Homotopy Two paths f and g with the same initial and endpoints are homotopic if one can be continuously deformed into the other. Formally, there exists a familiy $f_t : [0,1] \mapsto X, 0 \leq t \leq 1$ such that $f_t(0) = x_0$, $f_t(1) = x_1$ and $f_0(s) = f(s)$, $f_1(s) = g(s)$ and the function $F(s,t) : [0,1] \times [0,1] \mapsto X$ defined by $F(s,t) = f_t(s)$ is continuous (Figure 1).

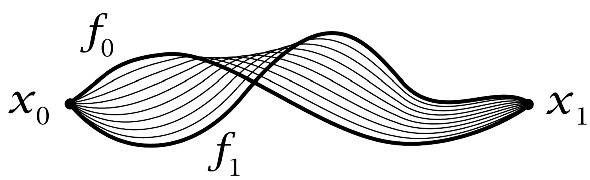

Figure 1: Homotopy [15].

Simplicial Complex A simplicial complex is a collection K of subsets of a set K_0 such that $\{v\} \in K$ for all $v \in K_0$ and if $\tau \subset \sigma$ and $\sigma \in K$, $\tau \in K$. The elements of K_0 are called vertices and the elements of K are called simplices. A simplex $\sigma \in K$ is called p-simplex if $|\sigma| = p+1$. The collection of all p-simplices is denoted as K_p. The k-skeleton of K is the union of all K_p for $p = \{1, 2, ..., k\}$. A simplex τ is called a face of simplex σ if $\tau \subset \sigma$. Figure 2 shows and example for $K = \{\{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,c\}, \{a,d\}, \{b,c\}, \{c,d\}, \{a,b,c\}\}$.

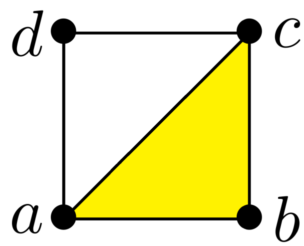

Figure 2: Simplicial complex [28].

Persitent Homology

The Path Planning Problem

We use the definition of a path planning problem introduced by Farber [10]. Given a map $\pi : PX \mapsto X \times X$ that associates with every path $\gamma \in PX$ the pair of it's initial and endpoints $\pi(\gamma) = (\gamma(0), \gamma(1))$, path planning is finding a function $s : X \times X \mapsto PX$ such that the composition $\pi \circ s = id$. A continuous path planning is then given if close intial-end pairs produce close movements (see Figure 3).

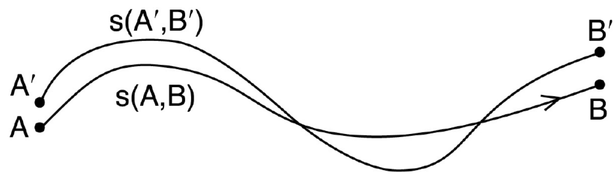

Figure 3: Continuous path planning [10].

An algorithm that solves the path planning problem is said to be *complete* if it finds a solution if there exists one and returns failure otherwise. It is called *optimal* if the returned path minimizes a cost metric. For sampling-based algorithms the theoretical guarantees are weakened. An algorithm is called *probabilistic complete* if the probability that it fails to return a solution if one exists decays to zero as the number of samples approaches infinity. It is called *asymptotically optimal* if the solution converges to the optimal solution as the number of samples approaches infinity.

Related Work

Static Obstacles

Graph search. These approaches assume that $\mathcal{C}$ is in the form of a graph. They then apply graph search techniques to find the path π. A widely used algorithm to compute shortest paths in a graph is Dijkstra's algorithm [8]. It is a greedy algorithm that computes the shortest path from every node to every other node in the graph. The next node for expansion is selected based on the lowest cost-to-come $\hat{g}(x)$. It is *complete*, meaning it finds a solution if one exists and reports failure otherwise. It is also *optimal* in the sense that the computed path is the shortest possible path.

A* [14] improves Dijkstra's algorithm by reducing the number of expanded nodes in the graph. This is achieved by using a different function $\hat{f}(x) = \hat{g}(x)+\hat{h}(x)$ to select the next node for expansion. The term $\hat{h}(x)$ is a heuristic for the least possible cost-to-go in a metric space. Many path planning algorithms in the current literature build upon A* (Figure 4).

Stentz et al. developed D*, an incremental algorithm based on A* that is able to find shortest paths on maps with changing costs, i.e. due to a robot moving trough the environment and updating the map. D* computes a path backwards from the goal to the robot. Koenig et al. [20] also proposed an incremental version of A* (LPA*), where successive runs only recalculate locally inconsistent nodes. They combined it with the backward search of D* and created D*-Lite

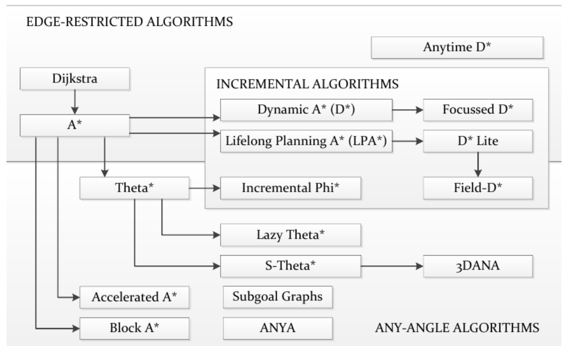

Figure 4: Evolution of graph search algorithms for the path planning problem [30].

[21], a less complex version of D* with at least the same performance therefore making D* obsolete.

Daniel et al. [7] showed an approach specifically for grid-maps that is not limited to predefined angles for the transition to other grid cells. It is based on A* and uses line of sight to determine successor nodes. Another any-angle approach proposed by Ferguson et al. [11] uses linear interpolation to find the least cost path trough a cell and produces globally smooth paths.

A comprehensive review of graph search algorithms for the path planning problem can be found in the literature [1, 30, 35, 26].

Sampling. Due to the curse of dimensionality [2], graph search in high dimensional configuration spaces can quickly become computationally intractable. Sampling-based approaches discretize the high-dimensional configuration space by generating random samples in $\mathcal{C}_{free}$. One highly influential contribution from Kavraki et. al. [19] is called Probabilistic Roadmap Method (PRM). It is a solution to the multi-query path planning problem in high dimensional configuration spaces. The method consists of a learning and a query phase. The former repeatedly generates a random configuration in $\mathcal{C}_{free}$ and connects this sample to neighboring nodes in a given radius with a fast local planner. In addition, a heuristic is used to generate extra nodes in "difficult" regions of $\mathcal{C}_{free}$. The method is therefore influenced by the topological properties of $\mathcal{C}_{free}$. The result is a graph representation of $\mathcal{C}_{free}$ in the form of a forest of trees.

Another highly cited approach using only a single tree is the Rapidly-exploring Random Tree (RRT) algorithm by Lavalle [23]. The algorithm grows a tree in $\mathcal{C}_{free}$ by generating a sample from a uniform distribution and growing the nearest node of the tree in the direction of the sample. Since the probability that a node is selected for expansion is proportional to the size of it's Voronoi cell (see Figure 5), the tree grows with a bias towards unexplored regions in $\mathcal{C}_{free}$.

Karaman et. al. [17] showed that both PRM and RRT algorithms are not asymptotically optimal. That means that with a growing number of nodes the probability that the algorithm finds an optimal solution converges to zero. In

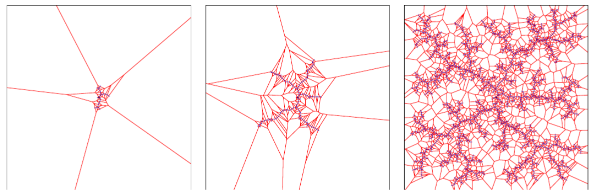

Figure 5: Voronoi diagram of a RRT tree.

their paper they also proposed two new algorithms PRM* and RRT* which are shown to be asymptotically optimal. The former was improved by calculating the radius, in which a sample is connected to it's neighbors, based on the number of present samples. The latter was improved by selecting the node for expansion based on the cost-to-come in the neighborhood of a sample and rewiring the tree after insertion of a new node.

There are many algorithms improving and extending RRT and RRT* in the literature. With RRT-Connect, Kuffner et. al. [22] introduced a bidirectional RRT that grows one tree from q_I and another tree from q_G while trying to connect the two trees. Anytime RRT* [18] returns a fast initial solution and a robot commits to execute a subset of the full path $\pi_{com} : [0, t_{com}] \mapsto C_{free}$. While executing π_{com} the remaining path is further improved until the robot commits to the next subset of the path. Realizing that there is a subset of C_{free} that contains configurations that are guaranteed to improve the current solution, Informed RRT* [12] uses a ellipsoidal heuristic to sample from this set and improve the convergence rate of RRT*. BIT* [13] does not sample a single configuration, instead it samples a batch of configurations and grows a tree from this batch. After finding a solution or no further possible expansion, the next batch is sampled. If a solution has been found with the previous batch, this batch is sampled from the ellipsoidal subset introduced by Informed RRT* and the tree is updated by identifying locally inconsistent nodes.

A comprehensive review of RRT* variants has been published by Noreen et. al. [27] and a review of more sampling-based approaches for the path planning problem can be found in [9].

Homotopy Invariants

Since the homotopy of paths forms an equivalence relation [15] it partitions the set of paths for a given initial-endpoint pair.

Bhattachary et. al. [5, 6] construct an augmented graph that can then be searched with classic graph search algorithms. Given a graph $G = (V, E)$, the augmented graph is a lift of G into the covering space of X. In configuration spaces homeomorphic to $\mathbb{R}^2$ they use Cauchy's Integral Theorem and the Residue Theorem from complex analysis to calculate a L-value which is equal for paths in the same homtopy classes and not equal otherwise. In configuration spaces homeomorphic to $\mathbb{R}^3$ they use laws from electromagnetism to calculate a H-value. These values are then used to augment the respective graphs. They propose another method to augment a graph in [3]. Here homotopy invariants for a D-dimensional manifold X are constructed by introducing $(D-1)$-submanifolds. Words are then formed by tracing a path f and inserting a letter every time f intersects one of the submanifolds.

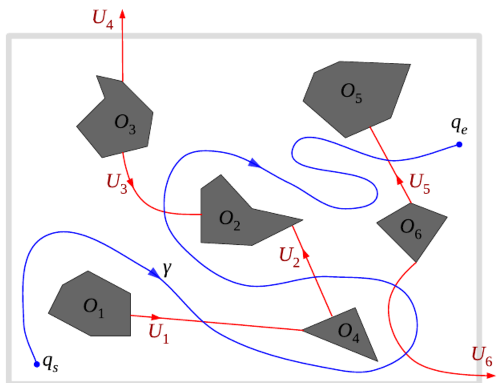

Figure 6: The blue path yields the constructed word $h(\gamma) = u_1^{-1}u_6u_6^{-1}u_2u_3u_5^{-1} = u_1u_2u_3u_5^{-1}$ [3].

Hershberger et. al. restrict the search of $\mathcal{C}$ to specific homotopy classes [16].

Yang et. al. [36] propose a method to calculate the k-shortest non-homotopic paths by simplifiying the topology and discarding $2^n - k$ of the 2^n possible paths in a two dimensional environment with n obstacles.

Wu et. al. [34] propose a method to deconflict paths of multiple robots in cluttered environments. Sequencially each robot computes it's path. The order is determined by the number of homology classes of paths of each robot.

A solution for a similar problem is shown by Wang et. al. [33]. They show a solution to multi-robot path planning in complex cluttered environments without inter-robot communication or coordination. The approach assigns robots stochastically to different topological classes and uses a potential field based controller to avoid local obstacles.

Persistent Homology

In [4] the authors try to solve the problem of how to threshold an occupancy grid map to get a binary map for the use with classic path planning techniques. They use persistent homology to get the homology class of paths that is persistent over the widest range of threshold values.

Configuration Spaces with moving Obstacles

In the thesis we want to apply the concept of persistent homology to the problem of planning paths with moving obstacles using graph search-based algorithms.

References

[1] Amylia Ait Saadi, Assia Soukane, Yassine Meraihi, Asma Benmessaoud Gabis, Seyedali Mirjalili, and Amar Ramdane-Cherif. UAV Path Planning Using Optimization Approaches: A Survey. *Archives of Computational Methods in Engineering*, April 2022.

[2] Richard Bellman. *Dynamic Programming*. Princeton Univ. Pr, Princeton, NJ, 1984.

[3] Subhrajit Bhattacharya and Robert Ghrist. Path homotopy invariants and their application to optimal trajectory planning. *Annals of Mathematics and Artificial Intelligence*, 84(3):139–160, December 2018.

[4] Subhrajit Bhattacharya, Robert Ghrist, and Vijay Kumar. Persistent Homology for Path Planning in Uncertain Environments. *IEEE Transactions on Robotics*, 31(3):578–590, June 2015.

[5] Subhrajit Bhattacharya, Vijay Kumar, and Maxim Likhachev. Search-based Path Planning with Homotopy Class Constraints. page 8, 2010.

[6] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. Search-Based Path Planning with Homotopy Class Constraints in 3D. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1):2097–2099, 2012.

[7] K. Daniel, A. Nash, S. Koenig, and A. Felner. Theta*: Any-Angle Path Planning on Grids. *Journal of Artificial Intelligence Research*, 39:533–579, October 2010.

[8] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959.

[9] Mohamed Elbanhawi and Milan Simic. Sampling-Based Robot Motion Planning: A Review. *IEEE Access*, 2:56–77, 2014.

[10] Farber. Topological Complexity of Motion Planning. *Discrete & Computational Geometry*, 29(2):211–221, January 2003.

[11] Dave Ferguson and Anthony Stentz. The Field D* Algorithm for Improved Path Planning and Replanning in Uniform and Non-uniform Cost Environments. Technical report, 2005.

[12] Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2997–3004, September 2014.

[13] Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Batch Informed Trees (BIT): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3067–3074, May 2015.

[14] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[15] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge ; New York, 2002.

[16] John Hershberger and Jack Snoeyink. Computing minimum length paths of a given homotopy class. *Computational Geometry*, 4(2):63–97, June 1994.

[17] Sertac Karaman and Emilio Frazzoli. Sampling-based Algorithms for Optimal Motion Planning, May 2011.

[18] Sertac Karaman, Matthew R. Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime Motion Planning using the RRT*. In *2011 IEEE International Conference on Robotics and Automation*, pages 1478–1483, May 2011.

[19] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, August 1996.

[20] S. Koenig and M. Likhachev. Incremental A*. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

[21] Sven Koenig and et al. D* Lite. 2002.

[22] James J Kuffner and Steven M LaValle. RRT-Connect: An Efficient Approach to Single-Query Path Planning. page 7.

[23] Steven M. Lavalle. Rapidly-Exploring Random Trees: A New Tool for Path Planning. Technical report, 1998.

[24] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, 2006.

[25] James Raymond Munkres. *Topology*. Pearson, Harlow, 2. ed., pearson new internat. ed edition, 2014.

[26] Alex Nash and Sven Koenig. Any-Angle Path Planning. *AI Magazine*, 34(4):85–107, September 2013.

[27] Iram Noreen, Amna Khan, and Zulfiqar Habib. Optimal Path Planning using RRT* based Approaches: A Survey and Future Directions. *International Journal of Advanced Computer Science and Applications*, 7(11), 2016.

[28] Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, December 2017.

[29] John H. Reif. Complexity of the mover's problem and generalizations. In *20th Annual Symposium on Foundations of Computer Science (Sfcs 1979)*, pages 421–427, October 1979.

[30] José Ricardo Sánchez-Ibáñez, Carlos J. Pérez-del-Pulgar, and Alfonso García-Cerezo. Path Planning for Autonomous Mobile Robots: A Review. *Sensors*, 21(23):7898, January 2021.

[31] Jacob T. Schwartz and Micha Sharir. On the "piano movers"' problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics*, 36(3):345–398, 1983.

[32] Jacob T Schwartz and Micha Sharir. On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics*, 4(3):298–351, September 1983.

[33] Xiaolong Wang, Alp Sahin, and Subhrajit Bhattacharya. Coordination-free Multi-robot Path Planning for Congestion Reduction Using Topological Reasoning, May 2022.

[34] Wenying Wu, Subhrajit Bhattacharya, and Amanda Prorok. Multi-Robot Path Deconfliction through Prioritization by Path Prospects. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9809–9815, May 2020.

[35] Bin Yan, Tianxiang Chen, Xiaohui Zhu, Yong Yue, Bing Xu, and Kai Shi. A Comprehensive Survey and Analysis on Path Planning Algorithms and Heuristic Functions. In Kohei Arai, Supriya Kapoor, and Rahul Bhatia, editors, *Intelligent Computing*, Advances in Intelligent Systems and Computing, pages 581–598, Cham, 2020. Springer International Publishing.

[36] Tong Yang, Li Huang, Yue Wang, and Rong Xiong. Efficient Search of the k Shortest Non-Homotopic Paths by Eliminating Non-k-Optimal Topologies, July 2022.