From Homotopy to Local Minima: Efficient Path Space Partitioning

Andreas Orthey

November 2023

Outline. This document explores the concepts of homotopies and local minima in robot motion planning. Homotopies describe continuous deformations between paths, partitioning the path space into homotopy classes. Local minimum classes, defined by path optimization methods, offer a finer, more practical partitioning. I compare those concepts and discuss why local minima classes can be more useful to partition the path space.

Let X be a configuration space of a robot for which we like to generate motions. In point-to-point motion planning [LaValle, 2006], an initial configuration a and a goal configuration b are given, and we are tasked to find a path between them. This defines a point-to-point motion planning problem, which can be written as a tuple (X, a, b). To solve such a motion planning problem, we need a motion planning algorithm (also called a *motion planner*) which takes (X, a, b) as input and which produces a path. A path is defined as a continuous function on X as

$$p : [0, 1] \to X. \tag{1}$$

The space of paths on X is then denoted as

$$\mathcal{F} = \{p : [0, 1] \to X \mid p \text{ is continuous}\}. \tag{2}$$

Using the path space, we can define a motion planner as a function which takes two configurations as input and produces a path [Farber, 2003]. This can be written as

$$s : X \times X \to \mathcal{F}. \tag{3}$$

Given initial and goal configurations $a, b \in X$ the solution space (codomain) of a motion planner will be the subspace

$$\mathcal{F}_{ab} = \{p \in \mathcal{F} \mid p(0) = a, p(1) = b\}. \tag{4}$$

We like to investigate this path space $\mathcal{F}_{ab}$ for a given motion planning problem and explore different concepts to partition this space.

1 Homotopies

The first concept to partition a path space is homotopy. A homotopy is a continuous deformation of a path into another path. It can be shown that the concept of homotopy creates equivalence classes which in turn partition the path space. Let us see how this works.

Let $f, g \in \mathcal{F}_{ab}$ be two paths in the subspace $\mathcal{F}_{ab}$. We say that f and g are deformable into each other if there exists a continuous function $H(s,t) : [0,1] \times [0,1] \to X$ with $H(0,t) = f(t)$, $H(1,t) = g(t)$. This function H is called a homotopy [Munkres, 2000, Farber, 2003, 2008].

If there exists a homotopy between two functions f and g, we write this as $f \simeq g$ and say that f is homotopic to g. It is important to note that the relation $\simeq$ is an equivalence relation on the space of paths, i.e. it is symmetric, reflexive, and transitive[1]. This is important, because an equivalence relation has the property that it will partition the underlying space into equivalence classes, thereby creating distinct regions [Munkres, 2000].

The equivalence classes which are defined by the homotopic equivalence relation are called homotopy classes and we denote them as H_α. We can then write $\mathcal{F}_{ab}$ as

$$\mathcal{F}_{ab} = \bigsqcup_{\alpha \in I} H_\alpha, \tag{5}$$

whereby we use the $\bigsqcup$ operator to emphasize that the H_α sets do not overlap, i.e. if $\alpha, \beta \in I$, $\alpha \neq \beta$, then $H_\alpha \cap H_\beta = \emptyset$. The index I denotes the set of all homotopies on $\mathcal{F}_{ab}$.

2 Linear Homotopies

A homotopy is one way to deform a path into another path. However, it is not the only way. For example, we could define a weaker version of homotopy, where two paths are only deformable into each other when they can "see" each other, i.e. there is a straight line connection between them. We call this a *linear (or straight line) homotopy* [Munkres, 2000, Jaillet and Siméon, 2008].

Linear homotopies are defined as follows. We say that f and g are linearly deformable into each other denoted as $f \sim g$ if there exists a linear homotopy $H(s,t) = (1-s)f(t) + sg(t)$. In that case we say that f is linearly homotopic to g. Note that the relation $\sim$ is not an equivalence relation anymore. It is symmetric and reflexive but not transitive. To see this, imagine a path for a point in 3-dimensional space. Let us imagine that there is a tall mountain on a planar field and that we define three paths starting from the same start position

[1] A relation $\simeq$ is an equivalence relation on a set X [Munkres, 2000], if the following properties hold for any elements $a, b, c \in X$.

- Reflexivity: $a \simeq a$.
- Symmetry: If $a \simeq b$, then $b \simeq a$.
- Transitivity: If $a \simeq b$ and $b \simeq c$ then so is $a \simeq c$.

in front of the mountain to the same goal position behind the mountain. Let us define the first path p_a going left around the mountain, a second path p_b going over the mountain, and a third path p_c going right around the mountain. We can linearly deform p_a to p_b (because they can see each other), and p_b to p_c, but we cannot linearly deform p_a to p_c, because the mountain blocks the way. This violates the transitive property of equivalence relations. The linear homotopy thus does not create a partitioning anymore, but instead it creates something we call a covering, where sets can overlap. Such a covering is related to the homotopy classes as $H_\alpha = \bigcup_{\beta \in J_\alpha} L_{\alpha\beta}$ such that

$$\mathcal{F}_{ab} = \bigsqcup_{\alpha \in I} \bigcup_{\beta \in J_\alpha} L_{\alpha\beta}, \tag{6}$$

whereby the $\bigcup$ operator is used to denote possible overlapping sets and J_α is the set of linear homotopies in homotopy class α.

It can be readily seen that the linear homotopy relation induces a finer decomposition of the path space. This intuition can be visualized by analyzing the path space of a simple spherical spacecraft in 3-dimensional space, when it has to navigate through a small asteroid field. While each path of this spacecraft might go around different asteroids, make turns, and meander around, in the end we can always stretch each path until it can be moved outside the asteroid field [Orthey et al., 2020]. In this case, any path can be continuously deformed into any other, and we end up with a single homotopy class. However, the asteroids would split the paths into linear homotopy classes (because paths might fail to see each other) which together would cover the path space of our spacecraft. We can say that the linear homotopy induces a finer decomposition as the homotopy decomposition.

While linear homotopies are more practical than homotopies [Jaillet and Siméon, 2008], they do not create equivalence classes, which makes them often difficult to deal with.

3 From Homotopies to Local Minima

While homotopy is an intuitive, rigorous concept, it is, however, difficult to compute efficiently. Finding a homotopy to deform one path into another can be almost as hard as solving a motion planning problem itself [Bhattacharya et al., 2018]. It is therefore difficult to talk about homotopy classes, because computing them is computationally impracticable. Moreover, as some of the above examples have shown (point in asteroid belt), many realistic scenarios have a single homotopy class, thereby making it of limited value for motion planning, especially when we want to scale to higher-dimensional spaces.

Instead of homotopies, I like to shift our focus to a similar, but more efficient operation. This operation is path optimization. Path optimization is already used by many motion planning algorithms, often as a post-processing step to deform a path into a more optimal path [Zucker et al., 2013, Toussaint, 2014, Kamat et al., 2022]. It turns out that this is a computationally efficient way to achieve a practical partitioning of the path space.

To understand this, let us look closer at path optimization. Let $\phi : \mathcal{F}_{ab} \to \mathcal{F}_{ab}$ be a path optimizer. This path optimizer takes as input a path in $\mathcal{F}_{ab}$ and returns another path in $\mathcal{F}_{ab}$ which is hopefully more optimal with respect to some internal optimization criterion. One can think about such a path optimizer ϕ as having multiple regions of attraction in $\mathcal{F}_{ab}$, where every path is optimized to a common locally minimal (optimal) path p^*. Note that those regions do not overlap.

Given two paths f and g in $\mathcal{F}_{ab}$, we can thus define an equivalence relation $f \overset{\phi}{\simeq} g$. This equivalence relation defines both paths to be equivalent, if, after application of ϕ, they converge to the same local minima path p^*. Note that this is again symmetric, reflexive, and transitive. Thus we arrive at a decomposition of the path space as

$$\mathcal{F}_{ab} = \bigsqcup_{\alpha \in I} \bigsqcup_{\beta \in J_\alpha} P_{\alpha,\beta}, \tag{7}$$

whereby each equivalence class $P_{\alpha,\beta}$ is associated with a specific local minima path $p^*_{\alpha,\beta}$ to which any path in $P_{\alpha,\beta}$ converges to. This is a finer partitioning than homotopy classes, but more efficient. Given any two paths, one can readily check if the associated path optimizer converges to the same path [Orthey et al., 2021].

4 Conclusion and Summary

Homotopies have been well-studied in motion planning [Jaillet and Siméon, 2008, Bhattacharya et al., 2018], but they have limited applicability to path planning because they are often impracticable to compute. Instead, the concept of local minima shows a better alternative. To this end, here is some actionable advice.

- Homotopies, especially in higher-dimensional spaces, are difficult to compute and often highly non-intuitive. Let us therefore replace homotopy with the concept of local minima classes.
- If we want to generate a minimal set of paths to cover the path space, it should be our goal to have paths in each local minimum class, not in each homotopy class [Orthey et al., 2021].
- If we want to learn a diverse set of paths, local minima classes are an efficient way to accomplish this.

References

Subhrajit Bhattacharya, Robert Ghrist, and Vijay Kumar. Path homotopy invariants and their application to optimal trajectory planning. *Annals of*

Mathematics and Artificial Intelligence, 84(3-4):139–160, 2018. doi: 10.1007/s10472-018-9602-8.

Michael Farber. Topological complexity of motion planning. *Discrete & Computational Geometry*, 29(2):211–221, 2003.

Michael Farber. *Invitation to Topological Robotics*. EMS Press, Zürich, 2008. ISBN 978-3-03719-554-3.

Léonard Jaillet and Thierry Siméon. Path deformation roadmaps: Compact graphs with useful cycles for motion planning. *The International Journal of Robotics Research*, 27(11-12):1175–1188, 2008. doi: 10.1177/0278364908097583.

Jay Kamat, Joaquim Ortiz-Haro, Marc Toussaint, Florian T Pokorny, and Andreas Orthey. Bitkomo: Combining sampling and optimization for fast convergence in optimal motion planning. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4492–4497, 2022.

Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

James Raymond Munkres. *Topology*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 2000.

Andreas Orthey, Benjamin Frész, and Marc Toussaint. Motion planning explorer: Visualizing local minima using a local-minima tree. *IEEE Robotics and Automation Letters*, 5(2):346–353, apr 2020.

Andreas Orthey, Florian T. Pokorny, and Marc Toussaint. Approximate topological optimization using multi-mode estimation for robot motion planning. RSS Workshop on Geometry and Topology in Robotics, 2021.

Marc Toussaint. KOMO: Newton methods for k-order Markov constrained motion problems, 2014.

Matthew Zucker, Nathan Ratliff, Anca Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher Dellin, J. Andrew (Drew) Bagnell, and Siddhartha Srinivasa. CHOMP: Covariant Hamiltonian Optimization for Motion Planning. *International Journal of Robotics Research*, 2013.