Project Proposal

Benj Cape

September 27, 2021

1 Goal

The goal of this project is to devise and potentially write an algorithm that can determine the most walk-able path from some starting point to a summit.

More specifically: Given a terrain graph (a planar graph where each vertex v is given a value $h(v) \geq 0$), some starting point v_0 and ending point v_1, determine the most walk-able path $v_0 \to v_1$ through the terrain.

Given that this topic has been researched before, a more attainable goal might to be implement this work on some of Dartmouth's hiking mountains to compare mathematical trail development with actual development in the Upper Valley.

2 Relevant Work

2.1 Non-Topological Work

The idea criteria for trail placement have been described by the American Trail organization.[1]

2.2 Topological Work

This problem, I think, is closely related to the shortest path problem. Which has been described extensively on 2 dimensional graphs through algorithms such as A*, Dijkstra's, and many others. 2-D work is not entirely applicable thought because the problem at hand regards graphs in space, and cannot simply use euclidean distance, or a weight function on edges to determine distance. Nonetheless, some of the principles might be applicable.

Some related work on shortest paths within the terrain model of planar graphs:

1. A Fast Shortest Path Algorithm on Terrain-like Graphs
2. Finding shortest path on land surface
3. Finding Shortest Paths on Terrains by Killing Two Birds with One Stone
4. Ridge-valley path planning for 3D terrains

[1] https://www.americantrails.org/resources/criteria-for-trail-placement-control-points-and-layout

What's different here, is that edges do not need to be a part of the path, nor does an edge in the path described need to be strictly discrete across an edge, or a face. This means that we can introduce new edges, as well as new vertices in the path that we generate from this algorithm. The second link above solves a very similar problem, restricting the path creation on some θ which is the maximum angle that the path can ever realize, and some ϵ that determines the difference that the path can have from the true shortest path. This work is incredibly pertinent, but lacks other elements of hiking trails such as visibility, lack of switchbacks, avoiding ridge-tops, that the θ is not consistent throughout the trail, and intersections with streams and water. I see potential to apply this algorithm possible with additional information such as topographical maps and AI to read the maps and create trails that not only are comfortable, but also enjoyable.

3 Detailed Plan

For this project, I first plan on familiarizing myself with the content of the "Finding shortest path on land surface" paper. Understanding there methods, there underlying assumptions, and the motivations for their techniques.

The next step will be understanding how to extend their algorithm to provide for other constraints that are more pertinent to hiking trails. Or identifying possible points where the algorithm can be tweaked to enable faster, or more precise calculation using tools we have learned in class.

I would also be interested in testing their algorithm on data sets such as Mount Moosilauke, the Dartmouth SkiWay, and other mountains in the area if data sets are available to determine if the trails built in fact follow, or are close to following the "ideal" trail.

3.1 Materials

I will not need any extra materials to accomplish this project.

3.2 Timeline

1. Gather information on current research by **October 30th**
2. Investigate extensions, simplifications, or re-definitions of current algorithm until **November 14th**
3. Experimentation **November 14-20th**
4. Presentation **November 20-23rd**

COSC 249, Fall 2021: Introduction to Computational Topology

Ankita Sarkar `f004nwn`

1 Project Goal and Related Work

The Mapper algorithm [SMC07] in topological data analysis exploits the Nerve Theorem [ner21] in topological data analysis to visualize high-dimensional data as a simplicial complex in low dimensions. Such visualizations can lead to unexpected features of the data being exposed, like in the case of Muthu Alagappan's reimagining [Coh12] of basketball positions. However, a key roadblock in the application of Mapper is that it requires hand-tuning of certain parameters in its input, which can affect the usefulness of the resulting visualization. It is, hence, important to automate the optimization of these parameters, and to characterize the "best" choices of parameters. In the 1-dimensional case, where the visualization is produced as a graph (i.e. a simplicial complex with simplices of dimension at most 1), there is an established notion [CMO18] of optimality of parameter selection. This notion is achieved by approximating the original dataset as a "nice and smooth" (paracompact) topological space, and then viewing the Mapper as a "pixellated" approximation of the Reeb graph of that space. A distance function is established to measure how close the Mapper is to the Reeb graph, and the parameters are tuned to minimize this distance.

For this project, one of my main goals is to understand how the above idea generalizes to higher dimensions, i.e. **how a general Mapper compares to the Reeb space**. This is also a well-studied area, including an approach involving category theory tools [MW16], and some other recent work that I have not read yet. To the best of my knowledge, none of these have managed to create a uniform notion of the *optimal* Mapper in terms of the Reeb space. I want to write a survey of these papers, and hopefully narrow down on a concrete open question regarding the optimality of Mapper as an approximation to the Reeb space.

Another goal for this project is that it might give me ideas to **improve upon two related notions called Multiscale Mapper [DMW] and Multimapper [DSK+18]**, both of which combine Mappers in multiple codomains to produce an ultimately "better" visualization. The notion of Multimapper is also not well-understood theoretically, and I would like to explore that.

Finally, the **selection of a cover for the codomain in the Mapper algorithm**, is often simply a overlapping intervals (in 1 dimension), grid (in 2 dimensions), or the higher dimension equivalent. This is not always the best choice for the "readability" and space complexity of the resulting simplicial complex, and in the work on Multimapper [DSK+18], a better method is suggested for 2 dimensions. I would like to formalize what "better" means, and possibly create algorithms to construct such better covers.

2 Project Plan

I will start off reading papers and working on a survey, and then continue on either a theoretical or experimental track (or a mixture) depending on what kinds of questions I narrow down on. Here is a 4-week plan, but I will adapt it to my evolving interest and emerging open problems.

Week 1. Survey the relationship of Mapper and Reeb space, and look out for theoretical gaps. The papers cited above, and some papers that they cite, are the reading plan. The broad question is "how to characterize when a Mapper is created with optimal parameters, and how does Reeb space help in achieving this?"

Week 2. Expand the survey scope to Multiscale Mapper and Multimapper, and continue identifying possible theoretical open questions. Try to create the theoretical backing of Multimapper and the cover selection problem. Replicate relevant experiments.

Week 3. Prove or disprove that Multiscale Mapper and/or Multimapper are always closer to the Reeb space than Mapper. Prove or disprove that cover selection "matters", i.e. can be a bottleneck, in how close a Mapper (or one of its extensions) is to the Reeb space. Produce visualizations and examples.

Week 4. Continue working on open question, if any. Add to the survey about persistence features of Mapper. (I cannot find the paper I had read on this – I will add a citation when I find it). Create an intuitive presentation of the category-theory viewpoint of this topic.

3 Resources needed

I anticipate that being able to discuss/present regularly will be the main resource I need. I will try to use department resources for experimental and creative tasks, and ask if they do not suffice.

References

[CMO18] Mathieu Carrière, Bertrand Michel, and Steve Oudot. Statistical analysis and parameter selection for mapper. *Journal of Machine Learning Research*, 19(12):1–39, 2018.

[Coh12] Ben Cohen. Muthuball: How to build an nba championship team - the mit sloan sports analytics conference, Jun 2012.

[DMW] Tamal K. Dey, Facundo Mémoli, and Yusu Wang. *Multiscale Mapper: Topological Summarization via Codomain Covers*, pages 997–1013.

[DSK+18] Bishal Deb, Ankita Sarkar, Nupur Kumari, Akash Rupela, Piyush Gupta, and Balaji Krishnamurthy. Multimapper: Data density sensitive topological visualization. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1054–1061, 2018.

[MW16] E. Munch and Bei Wang. Convergence between categorical representations of reeb space and mapper. In *SoCG*, 2016.

[ner21] Nerve of a covering, Jul 2021.

[SMC07] Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, editors, *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 2007.

Fleet Command: Hierarchical Multi-ASV Control and Topological Motion Planning System for Adversarial ASV Detainment

Mingi Jeong[1*] and Julien Blanchet[1†]

[1]Department of Computer Science, Dartmouth College, Hanover, 03785, NH, USA.

Contributing authors: mingi.jeong.gr@dartmouth.edu, julien.b.blanchet.gr@dartmouth.edu;
†These authors contributed equally to this work.

1 Goal

We propose to develop a solution to a *multi-robot Autonomous Surface Vehicle (ASV) detainment problem*, in which the goal is for a robotic naval fleet to surround and detain adversarial enemy ships. We define the 'detainment' as a situation where the enemy is unable to move a distance of δ without coming within ϵ of a some member of our friendly fleet (or some point along the shoreline). We plan to refine work from a previous class, where we presented a system in simulation (Fig. 1) capable of solving these issues using *virtual force*-based controllers. Specifically, we propose to explore one or both of the following: (1) topological motion planning, allowing for improved system performance in realistic scenarios, e.g., no information about mapping and localization of the enemy. (2) topological herding, in which we attempt to force enemy ships into dead ends, corners, or surrounds such that they are unable to escape (per the definition above).

2 Related Works

Our approach is related to works in terms of human robot interaction (HRI), multi-robot coordination and behaviors, multi-robot defense system, and topological motion planning in robotics. As per the principal topic – *computational*

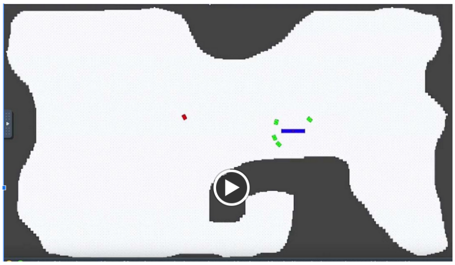

Fig. 1 Fleet Command simulation, developed in COSC 269.

topology – in this class, we mainly focus on the works having applications of topological motion planning aligned with the 'detainment' problem we try to address.

2.1 Topological Exploration in Unknown Environments

To detain an adversarial ASV in unknown environments, exploration and mapping of the environment is a key to the pre-requisite task – targeting and identifying the enemy. By using a distributed swarm robotic system, Dirafzoon and Lobaton [1] developed algorithms based on estimation of a topological model with limited information – localization – retrieved from the biotic agents. The method extracted topological information of the environment from persistent features and encounter times of the team members. Kim et al [2] proposed a method to identify topological classes (*homotopy and homology class invariants*) of the trajectories (*embedded*) connecting between the explored region and the unexplored region. This topological representation of the environment, having a role of a coarse map, helps a single robot or coordination of multiple robots.

2.2 Topological Representation of Separating Objects

Detaining and capturing of an enemy is analogous to separating and manipulating objects. Bhattacharya et al [3] presented automated separating objects by ships – cleaning oil spills or removing debris – based on the theoretical foundation derived from topological invariants (*homology* and *homotopy*). Their method finds the necessary topological conditions (*separating configuration* identified by its homology class) for separation of objects; then, finds optimal robot trajectories leading to the separating configuration with the use of physical cables. Although it is not directly related to separating objects problem, Kim et al [4] presented path planning for a tethered mobile robot by adopting the information about the *homotopy class* of the cable, i.e., constrained by curves surrounding the nearby environment (obstacle). By constructing a *homotopy augmented graph*, their method finds the shortest paths from the initial robot-cable configuration to a final robot position.

3 Ideas and Plans

3.1 Ideas

As we identified in the related works, the principal idea is potentially comprised of two components:

1. **Exploration and coverage of map** : as surveyed in [5], realistic scenarios for 'pursuit-evasion' and 'autonomous search' in mobile robotics need to consider the deployment of ASVs under partially or fully unknown environment. If the enemy is identified by a single ASV, while the multiple ASVs are exploring the environment with the help of topological concepts in Section 2.1, the ASV asks for help from the team. However, since the enemy is also a mobile robot, it might not be identified during the search or it might have moved from the last detected position. We plan to implement a 'retrial' of the exploration, until the ASVs finds the enemy in the region.
2. **Detaining the enemy**: Although the topological invariant is introduced based on the physical wire connection [3, 4], we can alternatively think of a virtual connection among the friendly ASVs detaining the enemy. This theoretical foundation derived from the topological concept is expected to obtain and identify the detainment faster than our previous work – regular convex polygon. We expect that the detainment task is significantly related to the Jordan Curve Theorem and rotation (or winding) number, while the ASVs are surrounding the enemy.

3.2 Plans

The following plans including the timelines and direction of the work are subject to change as per a discussion with Prof. Chang. We aim to extend this study in connection with our work presented at COSC 269 (Multi Robot Systems) by Prof. Quattrini Li. Our goal for this term is to investigate topological techniques for the multi-robot detainment problem and develop a proof-of-concept implementation. Ultimately (assuming this path of exploration is fruitful), we'd like to build this line of work into a publishable study after this term is over.

- Rigorous survey of the related works and understanding the concepts for topological motion planning aligned with our main detainment task: week 1, 2 and on-wards
- Set-up of the simulated platform and ROS (Robot Operating System) implementation: week 3, 4
- Analysis of the proposed algorithm and comparison with the state-of-the-art methods: week 4
- Paper-worthy output: continue polishing after the term ends and submit in 2022.

References

[1] Dirafzoon, A., Lobaton, E.: Topological mapping of unknown environments using an unlocalized robotic swarm. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5545–5551 (2013)

[2] Kim, S., Bhattacharya, S., Ghrist, R., Kumar, V.: Topological exploration of unknown and partially known environments. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3851–3858 (2013)

[3] Bhattacharya, S., Kim, S., Heidarsson, H., Sukhatme, G.S., Kumar, V.: A topological approach to using cables to separate and manipulate sets of objects. The International Journal of Robotics Research **34**(6), 799–815 (2015)

[4] Kim, S., Bhattacharya, S., Kumar, V.: Path planning for a tethered mobile robot. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1132–1139 (2014)

[5] Chung, T., Hollinger, G., Isler, V.: Search and pursuit-evasion in mobile robotics. Auton. Robots **31** (2011)