A Topological Approach to Finding Coarsely Diverse Paths

Aakriti Upadhyay[1] Boris Goldfarb[2] Chinwe Ekenna[1]

***Abstract*— We present a topological method for finding coarsely diverse pathways. The use of pre-computed paths for online planning in a dynamic context reduces the overhead of re-planning alternate routes. Our algorithm applied the notion of discrete Morse theory to identify critical points incident on the obstacles and used this information to identify and return a diverse set of coarse paths. Three sampling-based planning approaches are converted to topology-aware planners and compared to another that employs the SPARS2 path planning algorithm. We report on the number of coarse pathways found, computation time, and average path length and show that our approach outperformed previously published path diversity algorithms.**

I. Introduction

Path planning is needed to successfully maneuver autonomous robots while avoiding obstacles and banging into walls in a small passage. Many sampling-based algorithms are used in motion planning, including RRT (Rapidly Exploring Random Trees) [1] and Probabilistic Roadmaps (PRM) [2]. Because these methods create random samples in space, they are easier to implement than discrete approaches like A*. Due to the randomness of these sampling-based procedures, probabilistic completeness is the best that can be accomplished. For every failed path due to an unexpected new obstacle, a re-computation is required with just a probabilistic completeness guarantee and large time overhead.

Unfortunately, little effort has gone into providing robots with valid paths (alternative routes). Kavraki et al. termed this as path diversity in [3]. It saves time and delivers more information on space's properties and topology. It can help self-driving cars by offering alternate routes from a single map if one becomes invalid, and it also helps to service robots like the Roomba [4].

We present a new approach to identify and provide path sets during a single roadmap generation. We achieve this by applying discrete Morse theory on the topological roadmap to identify critical points which contain information about different regions of the space. Figure 1 gives an illustration of our approach. Using this information, we identify different path set incidents to these critical points and formally define path diversity for our work. We perform experiments in the configuration space ($\mathcal{C}_{space}$) using different sampling methods to generate our dense map, i.e. Uniform [2], Gaussian [5] and Bridge-Test [6] and provide analysis of the path sets returned. We compare our method with a baseline method presented in [3]. Our results show that we can generate more diverse paths with reduced path length. We also show that our algorithm can provide a set of coarsely-diverse paths that can be distinguished based on critical points information. This capability can have application in real-world scenarios where alternate energy-efficient routes are needed.

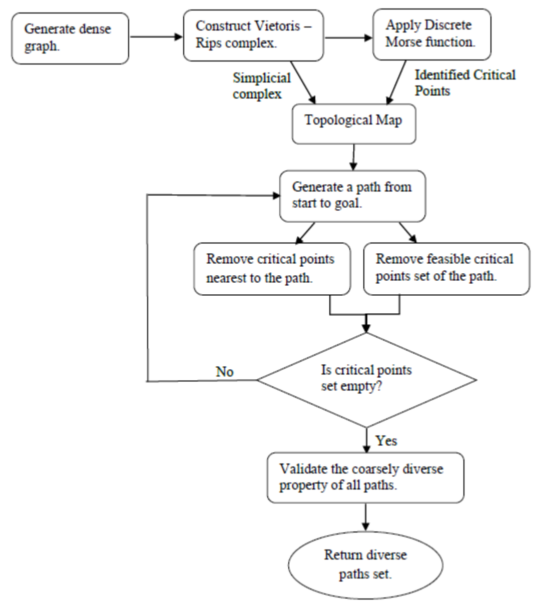


Fig. 1: Proposed Approach

II. Related Work

A. Motion Planning Preliminaries

A robot's placement, or configuration, can be uniquely described by a point $(x_1, x_2, ..., x_n)$ in a n dimensional space (x_i being the i^{th} degree of freedom (DOF)). The space consisting of collection of feasible configurations ($\mathcal{C}_{free}$) and unfeasible configurations ($\mathcal{C}_{obst}$) of robot is called configuration space ($\mathcal{C}_{space}$) [7]. The motion planning problem can be defined as finding a valid solution (e.g., collision-free and satisfying all joint limit and/or loop closure constraints) in $\mathcal{C}_{free}$ on successfully connecting start and goal configurations [8].

B. Sampling Based Motion Planning

A number of strategies have been proposed to modify the sampling strategy to increase the number of nodes sampling in narrow and difficult regions of the environments. Uniform sampling method [2] generates nodes uniformly at random in $\mathcal{C}_{space}$ retaining valid ones, however, show inability to sample in narrow passages efficiently. Obstacle-Based PRM

This research is supported in part by NSF awards CRII-IIS-1850319.

[1]A.Upadhyay and [1]C.Ekenna are with the Department of Computer Science and [2]B.Goldfarb is with the Department of Mathematics and Statistics, University at Albany, SUNY, USA. {aupadhyay, bgoldfarb, cekenna}@albany.edu

(OBPRM) [9] samples configurations near $\mathcal{C}_{obst}$ surfaces either by pushing configurations to the $\mathcal{C}_{obst}$ boundary or by finding surface intersections of randomly placed line segments. Even if OBPRM excels in narrow passages, it can be expensive because it requires many validity tests. Gaussian [5] and Bridge-Test [6] filter samples with inexpensive tests to find samples near $\mathcal{C}_{obst}$ boundaries or directly in narrow passages, respectively. However, both methods perform the same basic sampling as uniform random sampling and suffer from needing many samples to find one in a narrow passage. We use these auxiliary samplers to generate the topological maps for our approach except OBPRM, which continuously failed to generate a dense map to meet our stopping criteria.

C. Prior Work in Path Diversity

Knepper and Matthew in [10] introduced an important concept about path set as being a collection of feasible paths and their corresponding control sequences. A path set contains high path diversity if there is an improved performance in the presence of obstacles and goal-seeking behaviors. The importance of path diversity cannot be overemphasized, and the ability of a motion planning algorithm to return path sets, thus, giving the robot or a self-driving car choice of alternative paths reduces complexity and replanning scenarios. Voss et al. in [3] proposed an algorithm that returned low-cost diverse paths from a sparse graph by calculating the closest distance between path curves using Fréchet distance measure. However, the method required tuning of two parameters to return paths set.

For a graph with a broad explicit set of paths, Branicky et al. [11] provided two approximate algorithms (Inner-Product and Inclusion-Exclusion) that pruned large sets of candidate paths or trajectories down to smaller subsets while maintaining desirable characteristics in terms of overall reachability and path length. Extended work in [12] focused discussion on the survivability of the path from the collection of diverse paths w.r.t. random placement of the robot into an obstacle field in a static or dynamic space. In this paper, we instead produce a topologically rich graph, which implicitly provides critical point information, and our method generates paths with different path class representations.

Recent work in [13] iteratively found diverse paths in the $\mathcal{C}_{space}$ for a simplified multiple path problem by reducing the size of the robot (called inhibited regions). Although their method provided an increased success rate for new solutions, the oversimplification of the robot's dimension reduced the robustness of the approach. In this work, we propose a topology-based solution to maximize the use of non-degenerate critical points derived from discrete Morse theory to compute diverse paths in the environment.

III. Preliminaries

Definition 1: (Vietoris-Rips complex) Given a set X of points in a Euclidean space E, the Vietoris-Rips complex $R_\varepsilon(X)$ is the abstract simplicial complex whose k-simplices are the subsets of $k+1$ points in X of diameter at most ε.

In [14], a simplicial collapse, on constructed simplicial complex, removed redundant information to provide a space approximation of $\mathcal{C}_{free}$ as a pre-processing step. An extension in work [15], we applied discrete Morse theory to the same simplicial complex to identify critical points on the boundary of $\mathcal{C}_{obst}$. We make the following formal definitions.

Definition 2: Let D be the Euclidean distance function that measures the distance between the point $x \in \mathcal{C}_{free}$ and the nearest point y on the closest obstacle $O_i \in \mathcal{C}_{obst}$, that is, $D(x) = \min_{y \in O_i} \|x - y\|$.

Definition 3: Let $\Gamma(y, \varrho)$ be a density function where $\varrho > 0$ and y is the point on the obstacle. The function Γ counts all neighbors close to y in S within distance ϱ.

Definition 4: Let f be a discrete Morse function on $R(S)$ restricted to the vertices of the Vietoris-Rips complex. One option was defined in [15]. In our case, this is also the restriction of a Morse function formally defined at any point in $\mathcal{C}_{space}$ by

$$f(x) = D(x) \times \Gamma(y, \varrho). \tag{1}$$

Definition 5: (Critical points) The set of critical points is defined as the set of non-degenerate points on the convex hull of $\mathcal{C}_{obst}$ when the given discrete Morse function f reaches its extreme values, i.e. local minima or maxima.

Here, we denote critical points set as C. The derived feasible critical points information near $\mathcal{C}_{obst}$ was obtained within $\mathcal{C}_{free}$ by essentially shifting the critical points radially at a distance ϱ from $\mathcal{C}_{obst}$ to within S, thus providing a set of vertices in $S \subseteq \mathcal{C}_{free}$. The computation of ϱ value does not affect the identification of feasible critical points, and, thus, can be of any choice. Here, we calculate the value of ϱ as defined in [15].

Definition 6: (Feasible critical points) This set is defined as all vertices in S at a radial clearance of ϱ from a critical point of $\mathcal{C}_{obst}$. In other words, it is the union of intersections of vertices in S within the metric balls of radius ϱ centered at some critical point.

Let c_i denote an element of the set C, i.e. $c_i \in C$, and let v_j denote an element of the set S, $\forall i, j > 0$. We denote the list of the feasible critical points within ϱ from c_i by $F(c_i) = \{v_0, ..., v_q\}$, $\forall q, 0 \leq q < size(S)$.

To this end let us think of F as a function from critical points C to the power set of S, that is the set of its subsets. If all $F(c_i)$ are disjoint, which happens when all critical points are at least 2ϱ apart, then there is a well-defined function F^{-1} from feasible critical points $F(C)$ back to C. Even when this is not true, we can think of $F^{-1}(v)$ as the pre-image of v under the multi-valued function F. For each path p, we then have $F^{-1}(p)$ which is the union $\bigcup_{v \in p} F^{-1}(v)$ but can also be thought of as a sequence in C because it inherits the order from p.

Definition 7: (Path diversity) Given a set of paths P, two paths p_a and p_b in P are *diverse* if $p_a \neq p_b$. They are called *coarsely diverse*, or more precisely *ϱ-coarsely diverse* if for some time $t > 0$ we have distance $d(p_a(t), p_b(t)) > \varrho$. Two paths that are not coarsely diverse are called *fellow travelers*.

d is the Hausdorff distance function that measures the distance between two paths at time t.

Proposition 1: Suppose the distance between any two critical points is at least 3ϱ. Then two paths p_a and p_b in P are coarsely diverse if $F^{-1}(p_a) \neq F^{-1}(p_b)$.

Proof: The inequality means that for some $t > 0$, $F^{-1}(p_a(t)) \neq F^{-1}(p_b(t))$, so $d(F^{-1}(p_a(t)), F^{-1}(p_b(t))) > 3\varrho$. Since $d(F^{-1}(p_a(t)), p_a(t)) < \varrho$ and similarly $d(F^{-1}(p_b(t)), p_b(t)) < \varrho$, by the triangle inequality $d(p_a(t), p_b(t)) > \varrho$ as needed. ∎

Proposition 2: The equation $F^{-1}(p_a) = F^{-1}(p_b)$ defines an equivalence relation on paths. All paths in the same equivalence class are pairwise fellow travelers within the bound 2ϱ.

Proof: The first statement is clear. For the second notice that $d(p_a(t), p_b(t)) \leq 2\varrho$ for all $t > 0$ as $d(p_a(t), F^{-1}(p_a(t)) \leq \varrho$, $d(p_b(t), F^{-1}(p_b(t)) \leq \varrho$, and by the assumption $F^{-1}(p_a(t)) = F^{-1}(p_b(t))$ for all $t > 0$. ∎

We can conclude that to obtain a collection of diverse paths, or even better, coarsely diverse paths, it suffices to makes sure the collection contains paths that map via F^{-1} to distinct sets of critical points.

Corollary 1: Suppose c is in $F^{-1}(p_a)$ but is not in $F^{-1}(p_b)$. Then p_a and p_b are coarsely diverse.

IV. Methodology

In this algorithm, multiple paths map to a different collection of critical points incidents to them, as defined in Def.7. Critical points represent the important intersection points on the surface of $\mathcal{C}_{obst}$, i.e., high or low curve points. These critical points are non-degenerate, and, so, do not change with change in the dimensionality of the robot, as $\mathcal{C}_{obst}$ are rigid (non-deformable). The number of critical points for a $\mathcal{C}_{obst}$ in 2D space, like square, can be a small countable value, but a polyhedral representation of $\mathcal{C}_{obst}$ can have enormous critical points due to its intricate structure in 3D space. In such a framework, the possibility of having multiple paths increases around a single $\mathcal{C}_{obst}$ using its critical points information. So, the pruning process removes only those critical points close to the path within distance 2ϱ, from the identified critical points set. After pruning, there exists a possibility that different paths map to the untouched critical points of the same $\mathcal{C}_{obst}$ in the future. Therefore, it also validates the output paths to make sure that they pass along different groups of $\mathcal{C}_{obst}$ using *proposition 1*.

Algorithm 1 finds coarsely-diverse paths using identified critical points information and pre-computed simplicial complex from our previous work [14], [15], in lines 2-5. The algorithm calls *ConstructComplex* method to construct simplicial complex S (Def.1), on a verified Hausdorff distance measure of the dense graph G, and performs a sequence of simplicial collapse on S in the method *TopologicalCollapse*. The discrete Morse function f on S identifies critical points as the computed morse values in eq.(1) reaches its extrema, in method *IdentifyCriticalPoints*. Finally, it executes *GetFeasiblePoints* method to get vertices in S at ϱ-clearance (Def.6) from the critical point of $\mathcal{C}_{obst}$. The resulting topology map is a complete connected graph combined with necessary topological information, i.e., vertices and edges, of the $\mathcal{C}_{free}$ and the critical point information. The algorithm solves a query from start to goal position and uncovers different paths in the $\mathcal{C}_{space}$, using this map. At each iteration (line 6), the algorithm updates the graph while pruning the sets of feasible critical points and constituting critical points of the already found path such that no future path passes through the same groupset.

As a result, when the local planner begins to build a path for every updated graph complex S, it provides a path incident on feasible critical points representing a different collection of critical points compared to all other paths previously obtained (line 8). The algorithm terminates when all possible combinations of the paths for the environment have been visited, and, no more query can be solved with the available vertices in the S (line 18).

Algorithm 1 Planning coarsely diverse paths

Input: G: A dense graph comprising of vertices set V and edges set E where $G = \{V, E\}$, Q: query to be solved from start to goal position, P: set of n-distinct paths, p: vector array of path vertices.

1: Let $n = 0$, $P \leftarrow \{\phi\}$, $p = 0$.
2: $S \leftarrow ConstructComplex(G)$; ◁ Refer Def. 1
3: $TopologicalCollapse(S)$; ◁ Refer [14]
4: $C \leftarrow IdentifyCriticalPoints(S)$; ◁ Refer Def. 4, 5
5: $F \leftarrow GetFeasiblePoints(S, C)$; ◁ Refer Def. 6
6: **while** $|C| > 0$ **do**
7:   **if** Q == false **then**
8:     $p = PlanPath(S)$
9:     **if** Q = true **then**
10:       $P = P \bigsqcup p$
11:       **for all** vertex $x \in$ p **do**
12:         $C = C/F^{-1}(x)$
13:         $y = F^{-1}(x)$
14:         $S = S/F(y)$
15:         Q $\leftarrow$ false
16:       $p.clear()$
17:     **else**
18:       No more path exists. Assign $C \leftarrow \phi$
19: $n = |P|$
20: **for all** $i = 0$ to $n - 1$ **do**
21:   **for all** $j = 1$ to n **do**
22:     **if** $\bigcup_{z \in P[i]} F^{-1}(z) \neq \bigcup_{z \in P[j]} F^{-1}(z)$ **then**
23:       list all $\mathcal{C}_{obst}$ covered by $P[i]$
24:     **else**
25:       $P = P/P[i]$
26: update $n = |P|$
27: return $\{P, n\}$

Once all the possible paths are found, the algorithm performs a validation to accept only paths that map via F^{-1} to distinct sets of critical points of coarsely-diverse path classes, and lists all the $\mathcal{C}_{obst}$ a particular path covers (lines 20-25). Finally, the algorithm outputs the total number of paths, i.e., n, and the set of coarsely-diverse paths, i.e., P, for a given $\mathcal{C}_{space}$.

V. Environment setup

The experiments were executed on a Dell Optiplex 7040 desktop machine running OpenSUSE operating system, and the algorithms were implemented in C++ language.

We performed experiments in 3 different environments:

- **3D Cluttered environment**: Obstacles are cluttered around the room as shown in Figure 5a. The robot has to traverse through these obstacles successfully to reach its goal.
- **House environment**: A L-shaped robot is placed in a house-like space with four different rooms, see Figure 5b. The obstacles, like, a box and two tables, are placed randomly in the different rooms, and the robot is required to pass through these obstacles to reach the goal position.
- **Kuka YouBot environment**: An 8 DOF robot in an environment with four different rooms, see Figure 5c. This robot is a simulation replica of Kuka YouBot [16]. The robot moves through different rooms within narrow passages and arrives at its destination where it performs an action (grasps or puts an object down).

VI. Experimental results

This section provides a discussion on the results obtained for Uniform [2], Gaussian [5] and Bridge-Test [6] planners using our approach, and compared with the Voss method [3] which used the SPARS2 algorithm [17]. RAPID [18] was used as our collision detection method during the sampling, connection and query stages. The experiment results were averaged over the value of 10 runs for topological map generation and 10 runs for diverse path planning for each method in all environments.

A. Generating a Dense Topology Map

We generate a dense graph in all three environments using the different planning methods. Table I shows the size of the maps needed to meet the criteria defined in the definitions and proposition as discussed in Section III.

TABLE I: Number of samples in the topology map

Environments	Uniform	Gaussian	Bridge-Test
3D Cluttered	5090	5918	6136
House	4985	6590	6504
Kuka YouBot	4908	6304	6601

Figure 2 show the number of edges formed on connecting nodes in a topological map for all planning methods.

B. Number of Diverse Paths

Figure 3 gives the information about the number of diverse paths returned after running Algorithm 1 on our different environment scenarios. The distance between the pair of paths was measured using the Hausdorff distance metric. We observe that the Voss method returned fewer paths in the 3D cluttered environment (3) compared to our approach that returned 5 diverse paths for all 3 planners. In the House environment, the Voss returned 8 paths while our method returned 5 in all planners' cases. We increase 100 units more for the path threshold T to obtain 8 paths for all planners, as shown in Figure 3b. Interestingly, the average path length of our coarse paths in all planners' cases was lower than

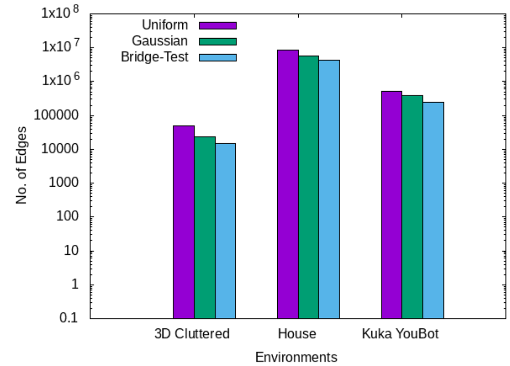


Fig. 2: Total edges in the Topological maps

Voss, as can be seen in Figure 4b. Finally, in the Kuka YouBot environment, the Voss method failed to complete and returned 0 paths while our approach returned 4 paths for all 3 planners. Voss method was unsuccessful due to its dependency on the SPARS2 that is limited to handle only planar and rigid body configurations (SE(2) and SE(3)), as discussed in [17]. It can also be viewed, from the Figure 3 that as the number of paths increases, the distance between paths decreases, and our method in all 3 planners maintained good diversity between obtained paths.

C. Setting a Threshold

We set a threshold for returned paths based on the range from the shortest diverse path length to a path of length T, where T is the mean of all shortest path lengths returned. It is due to the infinitely large number of diverse paths that can be generated in any environment depending on the mapping of paths to a plethora of critical point group combinations (detailed discussion available in Section IV). Table II to IV give information about our threshold and the diverse paths returned.

TABLE II: Paths generated in 3D Cluttered environment

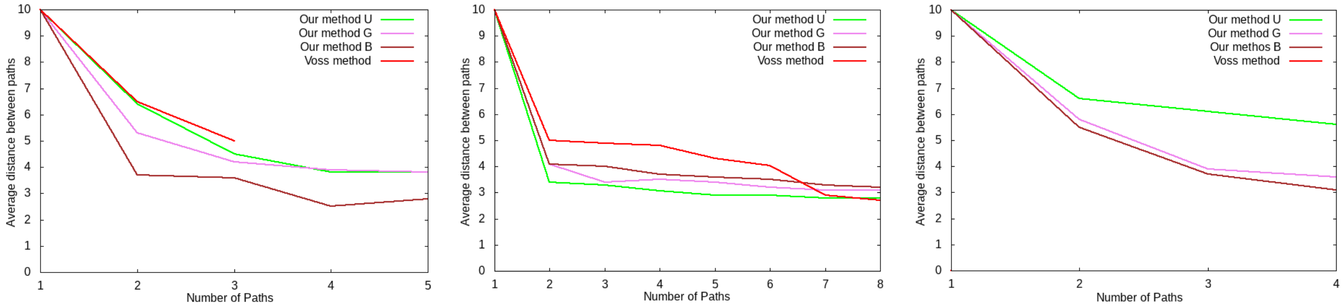
	Diverse Paths	Path length threshold
Uniform topological map	5	**300-400**
Gaussian topological map	5	350-500
Bridge-Test topological map	5	500-850
Voss method	3	N/A

TABLE III: Paths generated in House environment

	Diverse Paths	Path length threshold
Uniform topological map	5	**522-600**
Gaussian topological map	5	500-700
Bridge-Test topological map	5	500-650
Voss method	8	N/A

TABLE IV: Paths generated in Kuka YouBot environment

	Diverse Paths	Path length threshold
Uniform topological map	4	**1020-1500**
Gaussian topological map	4	1050-2000
Bridge-Test topological map	4	1200-2500
Voss method	0	N/A



(a) **3D Cluttered environment** (b) **House environment** (c) **Kuka Youbot environment**

Fig. 3: We show a decrease in the average hausdorff distance between paths as the number of coarsely diverse paths increase.For our method U = Uniform topological planner, G = Gaussian topological planner, and B = Bridge-Test topological planner.

D. *Computation Time and Average Path Length*

In Figure 4a, we can see that the time needed to build the maps and generate the diverse paths differ in the environments. The Voss method uses less time in the 3D cluttered environment than our method but was outperformed by the Uniform topological planner in the House environment. No result was recorded for Voss in the Kuka YouBot environment since it failed to complete. However, comparing the results of the topology methods in the Kuka YouBot environment, we see that the Uniform topology method uses less time than Bridge-Test and Gaussian and returns paths with a smaller average length, as seen in Figure 4b. We can postulate that Uniform planner worked best with our topology approach due to the need to form a dense map in the environment as quickly as possible, which is a strength Uniform (Basic PRM) has over the other methods. Although Gaussian and Bridge-Test provide samples closer to the $\mathcal{C}_{obst}$, the approaches require enormous samples to cover the entire space, resulting in oversampling around the $\mathcal{C}_{obst}$. Thus, Gaussian and Bridge-Test operate effectively in maze-like situations when combined with our technique. A detailed description of the properties of these different planners is available in the related work section.

E. *Critical Points Analysis and the Environment Topology*

We show some interesting analyses to aid an appreciation for the critical points and their role in identifying these diverse paths. Figures 5a to 5c show examples of paths returned in the 3D Cluttered, House, and Kuka YouBot environments, showing the paths that map to a different set of critical points within a close distance of ϱ.

To better explain the relationship between the obstacles and the critical points, we create a plot that depicts the number of $\mathcal{C}_{obst}$ in the environment v/s the diverse paths that pass through the various groups of critical points of these obstacles, as seen in Figure 5. Recall that we define our diversity based on the distinct collections of critical points each path constitutes. The number of obstacles in the 3D Cluttered environment is 23 with 170 identified critical points, in the House environment is 3 obstacles with 95 identified critical points and in the Kuka YouBot environment is 26 obstacles with 210 identified critical points. We also previously proved in *Proposition 1* that our critical points determine the number of coarsely diverse paths we can generate. In summary, we have successfully provided a new methodology to extract coarsely diverse paths from the topological information of the $\mathcal{C}_{space}$ and compared it with a baseline method. Our approach has shown successfully that diverse paths of shorter lengths can be possible.

VII. Conclusion and Future Work

While path diversity is already a fruitful topic in robotics, the work in this paper points to methods for the solution of a more compelling problem of finding diverse paths that represent diverse homotopy classes of paths. Suppose the $\mathcal{C}_{free}$ is two-dimensional and not contractible. In this case, we can assume that there is at least one obstacle that is homeomorphically a disk. Construction of non-homotopic paths can be achieved by successively selecting feasible critical points as goalposts near critical points on the boundary of that same disk. For example, suppose one obstacle in a 2D environment is a free-floating regular polygon (that is, not touching a wall). We posit that alternate selection of feasible critical points on diametrically opposite points of the boundary circle will result in the generation of two paths by Algorithm 1 that are not homotopic to each other, i.e. touching on parallel tangential sides of the disk.

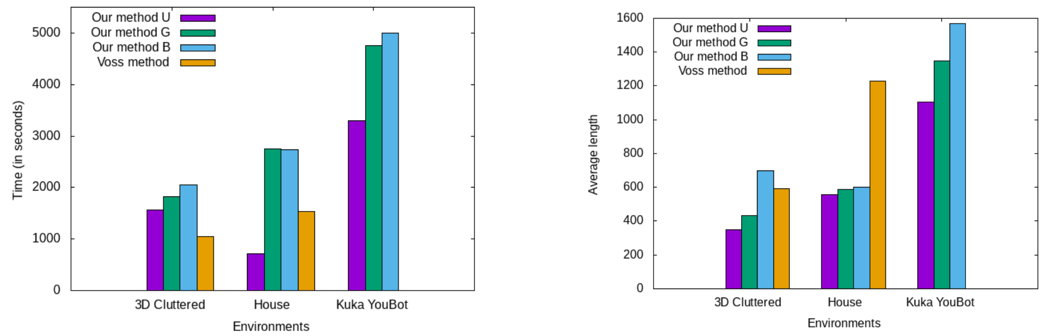
Looking to higher dimensions, the key property of the disk in a 2D environment for this purpose of creating homotopically diverse paths is the fact that the core of the disk is dimension 1, and so has *codimension 1* in $\mathcal{C}_{space}$. These ideas generalize directly to obstacles in any dimension that are co-dimension 1. The application of 1-dimensional persistent homology from topological data analysis to higher dimension robots and more complex obstacles will be presented in our future work.

References

[1] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1999, pp. 473–479.

[2] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, August 1996.

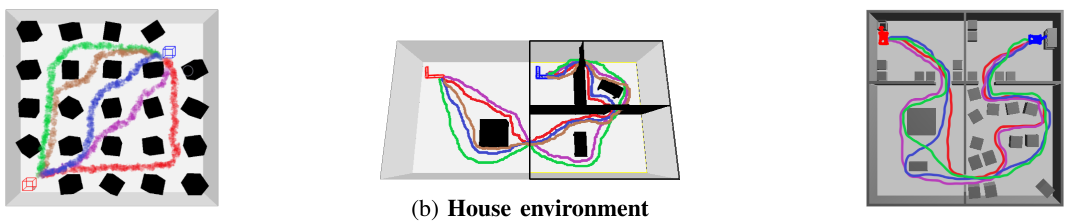
[3] C. Voss, M. Moll, and L. E. Kavraki, “A heuristic approach to finding diverse short paths,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4173–4179.



(a) **Computation Time**

(b) **Average Path Length**

Fig. 4: Computation Time and Average Path Length Comparisons.



(a) **3D Cluttered environment**

(b) **House environment**

(c) **Kuka Youbot environment**

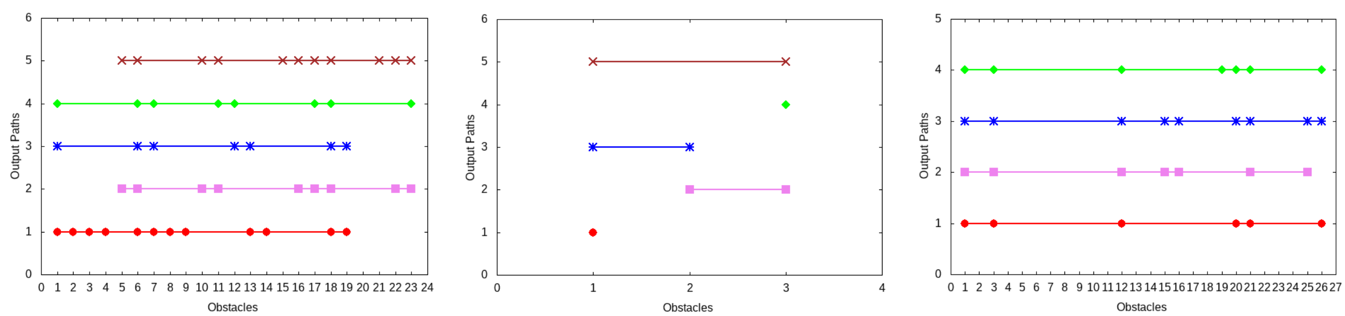


Fig. 5: The plots below each respective environment show the set of critical points of $\mathcal{C}_{obst}$ that map to different coarsely diverse paths within a close distance ϱ, for Uniform topological planner. Different colors denote the output path's color in these environments, and the dots on each horizontal line represent the mapped critical points of the different $\mathcal{C}_{obst}$ for each path, from start to goal positions.

[4] Roomba, "Roomba — Wikipedia, the free encyclopedia," 2002, accessed: 25-February-2021. [Online]. Available: https://en.wikipedia.org/wiki/Roomba

[5] V. Boor, M. H. Overmars, and A. F. van der Stappen, "The Gaussian sampling strategy for probabilistic roadmap planners," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, May 1999, pp. 1018–1023.

[6] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "Bridge test for sampling narrow passages with probabilistic roadmap planners," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2003, pp. 4420–4426.

[7] T. Lozano-Perez, "Spatial planning: A configuration space approach," *Computers, IEEE Transactions on*, vol. 100, no. 2, pp. 108–120, 1983.

[8] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, June 2005.

[9] Y. Wu, "An obstacle-based probabilistic roadmap method for path planning," Master's thesis, Department of Computer Science, Texas A&M University, 1996.

[10] R. A. Knepper and M. T. Mason, "Path diversity is only part of the problem," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3224–3229.

[11] M. S. Branicky, R. A. Knepper, and J. J. Kuffner, "Path and trajectory diversity: Theory and algorithms," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1359–1364.

[12] L. H. Erickson and S. M. LaValle, "Survivability: Measuring and ensuring path diversity," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2068–2073.

[13] V. Vonásek, R. Pěnička, and B. Kozlíková, "Computing multiple guiding paths for sampling-based motion planning," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 374–381.

[14] A. Upadhyay, W. Wang, and C. Ekenna, "Approximating cfree space topology by constructing vietoris-rips complex." in *IROS: IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 2019, pp. 2517–2523.

[15] A. Upadhyay, B. Goldfarb, W. Wang, and C. Ekenna, "Planning among obstacles: using discrete morse theory for topological extraction of geometric features," State University of New York, Albany, NY (United States), Tech. Rep., 2020. [Online]. Available: https://aakupadhyay.github.io/assets/images/DMT_in_cspace.pdf

[16] Kuka YouBot, "Research and application development in mobile robotics," 2021, accessed: 28-February-2021. [Online]. Available: https://www.generationrobots.com/img/Kuka-YouBot-Technical-Specs.pdf

[17] A. Dobson and K. E. Bekris, "Improving sparse roadmap spanners," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4106–4111.

[18] S. Gottschalk, M. Lin, and D. Manocha, "Obb-tree: A hierarchical structure for rapid interference detection," University of N. Carolina, Chapel Hill, CA, Technical Report TR96-013, 1996.