Planning among obstacles: using discrete Morse theory for topological extraction of geometric features

Aakriti Upadhyay* Boris Goldfarb◇ Weifu Wang† Chinwe Ekenna*

***Abstract*— In this work, we apply discrete Morse theory to a simplicial complex built from a sampled configuration space. This allows us to extract and exploit the properties of the configuration space with near-optimal path guarantees. Our algorithm leverages advantages provided by discrete Morse theory applied to density-based functions to detect critical points on the boundaries of the obstacles. We analyze the non-degenerate properties of these critical points to improve the dimension representation of the configuration space. Using this topological and geometric information, we derive path classes and produce a roadmap in less time than proven optimal methods like PRM* and RRT*. We perform experiments in different obstacle present environments to show the performance improvement in comparison with other methods.**

I. Introduction

A robot is a movable object whose position and orientation can be described by d parameters or degree of freedoms (DOFs), each corresponding to an object component (e.g., object positions, object orientations, link angles, or link displacements). Hence, a robot's placement, or configuration, can be uniquely described by a point $(x_1, x_2, ..., x_d)$ in a d dimensional space (x_i being the ith DOF). This space consisting of all possible robot configurations (feasible or not) is called the configuration space ($\mathcal{C}_{space}$) [1]. The subset of all feasible configurations is the free space ($\mathcal{C}_{free}$), while the union of the unfeasible configurations is the obstacle space ($\mathcal{C}_{obst}$). Sampling-based methods [2] represent the $\mathcal{C}_{space}$ with a roadmap of sampled configurations. The configurations are retained in $\mathcal{C}_{free}$ if the connection edge between two samples is in $\mathcal{C}_{free}$. These algorithms have been successful in high-dimensional space and are known to be probabilistic complete. These feasible paths, however, depend on the robot's geometry, its motion capabilities, the workspace geometry, and the topology of the underlying space. While solutions abound that help determines the geometry of the robot, there is still a fundamental challenge in being able to get an accurate description and analysis of the underlying space.

In this work, we propose an analytical and computational method that exploits both the topological and geometrical representations of the configuration space ($\mathcal{C}_{space}$). Our framework is composed of a pre-processing step, the graph-collapse method, which we previously developed in [3]. Our graph-collapse method extracts the connectivity information of an η-offset of the $\mathcal{C}_{free}$ space using Vietoris-Rips complex and performs simplicial collapse to prune redundant edges and samples. It then proves that a certain upper bound of samples is always sufficient to provide the topological information about the given sub-space in $\mathcal{C}_{free}$, i.e. memory-efficient information. We extend this work in this paper by building and implementing an algorithm that extracts geometrical information of the obstacle using discrete Morse function. This function identifies non-degenerate critical points in $\mathcal{C}_{space}$, classifies paths, and produces path with near-optimal properties.

The contribution of this paper includes:

- A means to extract a geometrical representation of the space, i.e. critical points on the boundary of $\mathcal{C}_{obst}$, to help in obstacle avoidance planning with minimum clearance.
- A new methodology that includes both topological and geometrical representations of the space to provide good quality near-optimal path.
- A new way to characterise available paths into classes which will aid in having a more informed representation of the planning space.

We perform experiments in the 2D environment (used by Karaman et. al in [4]) and 3D environments for different sampling-based techniques i.e., RRT, RRT*, PRM, and PRM* [5]. We analyze the performance in terms of time taken to plan a path, the number of nodes needed, path length/cost, and path clearance, and show improvements using our approach.

II. Preliminaries

A. Discrete Morse theory

Discrete Morse theory, as defined by Robin Forman [6], investigates homotopy type and homology groups of finite simplicial complexes. It is a discrete analog of the classical, smooth Morse theory, introduced in [7]. As described later in [8], the smooth Morse theory shows the relationship between critical points of a smooth map defined on a manifold and the topology of the manifold. In particular, given a compact manifold M and a smooth map $f : M \to \mathbb{R}$ whose critical points are all non-degenerate called a Morse map, M is homotopy equivalent to a simplicial complex X whose d-cells are in bijective correspondence with critical points of f of index d, for all $0 \leq d \leq dimM$. Moreover, if f has no critical values in some interval [a,b], then the manifolds $M(a) = f^{-1}((-\infty, a])$ and $M(b) = f^{-1}((-\infty, b])$ are diffeomorphic (a diffeomorphism is a homeomorphism that is smooth in both directions), and is a deformation retract of $M(b)$.

* are with the Department of Computer Science. † is with the Department of Electrical and Computer Engineering and ◇ is with the Department of Mathematics and Statistics, University at Albany, SUNY. {aupadhyay, bgoldfarb, wwang8, cekenna}@albany.edu

B. Space approximation using Vietoris-Rips complex

From our previous work [3], we defined two important concepts, i.e. abstract simplicial complex and Vietoris-Rips complex. We applied these mathematical concepts to perform a memory-efficient path planning in a given $\mathcal{C}_{space}$ on generating a homotopy-equivalent topological map of $\mathcal{C}_{free}$.

Definition 1: (Abstract Simplicial complex) An abstract simplicial complex K, i.e., a collection of sets closed under the subset operation, is a generalization of a graph useful in representing higher-than-pairwise connectivity relationships. The elements of the set are called vertices, and the set itself is a simplex.

Definition 2: (Vietoris-Rips complex) Given a set X of points in Euclidean space E, the VR complex $R(X)$ is the abstract simplicial complex whose k-simplices are determined by subsets of $k+1$ points in X with a diameter that is at most ε.

We constructed the VR complex for $\mathcal{C}_{free}$ on verifying the Hausdorff distance property (i.e., the distance between two compact sets is minimum and the intersection of two sets is ϕ) between simplicial complex set and the boundary set of $\mathcal{C}_{space}$. A simplicial collapse removed redundant information to provide a space approximation measure of the $\mathcal{C}_{free}$ in the generated topological map as a pre-processing step. In this paper, we apply the discrete Morse theory to the constructed simplicial complex and generate samples in proximity to the extracted geometrical representation of $\mathcal{C}_{obst}$ via topological information about the region of the samples.

III. Related Work

A. Sampling-Based Motion Planning (SBMP) methods

Sampling-based methods are broadly classified into two main classes: graph-based methods such as the Probabilistic Roadmap Method (PRM) [9] and tree-based methods such as Expansive-Space tree planner (ESTs) [10] and Rapidly-exploring Random Tree (RRT) [11]. PRM variants exist that sample near obstacles [12]–[16], with constraints placed on the robots [17] and representing uncertainty in the environment [18]. Other methods exist that investigate the heterogeneous nature of the planning environment using machine learning or reinforcement learning techniques [19]–[25]. Most sampling-based approaches, however, extract information that is either isolated or do not provide much information about the underlying space.

The work in [26] combined graph-search and sampling-based planning techniques through RGG (Random geometric graph) theory. A set of samples was defined as an implicit RGG batch with connected edges as an input to find an optimal path to the goal. The proposed algorithm, BIT*, used heuristics of previous information about path cost to prioritize the search of high-quality paths and focus the search for improvements as the number of samples reaches infinity.

B. Critical Points and Motion Planning

The need to plan safe and optimal paths for the robots in the presence of obstacles has been an important area of research. Early research work in the 90s [27] presented an online motion planning algorithm in 2D $\mathcal{C}_{space}$ for right-handed manipulators. Critical points on the boundaries of obstacles in $\mathcal{C}_{space}$ were identified using line-of-sight and wall-following methods. Improvements to path planning problems were made by modifying line-of-sight and wall-following algorithms by using the critical point graph to find exit points to goal locations. The work further extended to 3D $\mathcal{C}_{space}$ in [28] showing the unaffected performance of manipulators in higher-dimensional space. Dakin et. al. [29] presented a methodology for generating and testing fine-motion plans in generalized contact space. They assumed small intersection clearance to form a simplified, hyper polyhedral approximation of contact space around the critical points in a nominal assembly path.

In [30] an exact cellular decomposition was proposed, Morse function critical points provided the location of cell boundaries. They described a conventional slice algorithm where a slice is a flat plane defined by the pre-image of a real-valued function whose restriction to the boundary of the free space is a Morse function, i.e., has no non-degenerate critical points. The topology of the slice in the free space changes at these critical points. These critical points were used to form the cell boundaries such that the structure of each cell enables a planner to use simple control strategies such as back-and-forth motions for coverage tasks. Another work by [31] introduced a new probabilistic algorithm called Crawling Probabilistic RoadMap (CPRM) for real-time motion planning in the configuration space. The work defined a varying potential field f on ∂O as a Morse function where O is the obstacle in $\mathcal{C}_{space}$ and combining the naive PRM algorithm with the potential field method reduced the time needed. However, a CPRM algorithm is applicable for mechanisms whose $\mathcal{C}_{space}$ are algebraically known.

More recently, research in [32] presented an algorithm to find a stream function within a sampling-based planner (PRM*) such that two given points are connected by a trajectory. The work applied a Morse function to induce a stopping condition if two points P and Q lie on the same trajectory with a net velocity sufficiently large over the trajectory. In the above-cited works, the authors implemented algorithms to generate critical points in different scenarios like obstacle mapping, generating fine-motion planes, or for cell boundary indication. However, the functions are limited to the pre-known algebraic information of the $\mathcal{C}_{space}$.

IV. Methodology

A. The discrete Morse Function in $\mathcal{C}_{space}$

In our previous work [3], we constructed a simplicial complex in the $\mathcal{C}_{space}$ using the VR-complex. As an extension of the work, we apply the results of discrete Morse theory to the simplicial complex to identify critical points on the boundary of $\mathcal{C}_{obst}$.

Definition 3: 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 4: Let $\Gamma(y, \varrho)$ be a density function where $\varrho > 0$ and y is the point on the obstacle. The function Γ records all neighbors close to y within distance ϱ.

Theorem 1: f is a discrete Morse function when restricted to the vertices of the Vietoris-Rips complex. The Morse function formally defined at any point in $\mathcal{C}_{space}$ can be given as

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

Proof: Let $\omega \geq 0$. We consider any closest obstacle O. Let us consider $X = D^{-1}([0, \omega])$. Then we will denote by $Hull(X)$ the convex hull of the set X at scale ω, where t and s are the values of vertices/points in X.

$$Hull(X, \omega) = \bigcup_{t,s \in X} [t, s]. \tag{2}$$

Given an obstacle O, the local maxima and minima of the function f occur on the surface of the obstacle O. These are non-degenerate critical points in $\tau \subset O$ (as objects are rigid). The $Hull(\tau)$ determines the boundary of the obstacle surface containing these critical points. Let us take point $p \in X$; the value of $D^{-1}(\omega)$ determines the critical points of f if the distance from the surface of an obstacle boundary to the point p reaches its extreme value. When $D(p) = 0$, i.e. $\omega \to 0$, the distance between point p and closest obstacle O becomes null or zero. This corresponds to the decreasing density of points in $\mathcal{C}_{free}$ on approaching closer to the obstacle O, as shown in Figure 1a.

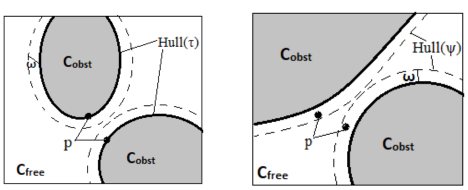
On satisfying the two conditions, i.e.,

1) $\forall$ points $p \in X$; $\exists$ point $y \in O$; $f(p) \leq f(y)$,
2) $\forall$ points $y \in O$; $\exists$ point $p \in X$; $f(y) \geq f(p)$,

the function f becomes a discrete Morse function and the equation for the identified critical point p can be given as,

$$\lim_{\omega \leftarrow 0} f(p) = D(p) \times \Gamma(y, \omega) \subset \text{Hull}(\tau). \tag{3}$$

■



(a) Critical point p (b) Feasible critical point p

Fig. 1: Critical and Feasible Critical Point Illustration

1) Defining feasible critical points in $\mathcal{C}_{space}$: Let Ψ denote the compact set of points in $\mathcal{C}_{free}$ at distance ϱ from the obstacle boundary such that the computed value can be given as,

$$\varrho = \frac{1}{n} \sum_{i=0}^{n-1} D(c_i, s); \forall c_i \in C, \forall s \in S \tag{4}$$

where C denotes the set of identified critical points and S denotes simplicial complex vertices set. The size of set C is given by n and distance function D comes from Def. 3.

By increasing $\omega > 0$ such that the value of $\omega \leq \varrho$, the critical point p (from eq.(3)) shifts towards a feasible region ($\mathcal{C}_{free}$) in $\mathcal{C}_{space}$ at distance ω from the obstacle boundary, thus, presenting a set of points in Ψ. So, the morse value equation becomes as

$$\lim_{0 < \omega < \varrho} f(p) = D(p) \times \Gamma(\omega) \subseteq \text{Hull}(\Psi). \tag{5}$$

Therefore, p becomes a feasible critical point with a ϱ-clearance from the obstacle boundary as shown in Figure 1b.

The Generalized Voronoi Diagram (GVD) has been considered as a roadmap to extract high-clearance paths. The GVD defines the maximum clearance for a path from the $\mathcal{C}_{obst}$ as utilized in the Medial-Axis PRM [33]. Unfortunately, an exact computation of the medial-axis distance is not practical for problems involving many DOFs and many obstacles as this require an expensive and intricate computation of the configuration space obstacles. In our previous work, we proved that on reaching a proven sampling condition, the VR-complex provides a topologically-equivalent map of the space to the Čech complex. On collapse, the resulting simplicial complex is a sparse sub-sampled graph that reconstructs the surfaces equivalent to the Delaunay complex as in [34]. Likewise, instead of taking the medial-axis distance from $\mathcal{C}_{obst}$ to the boundary of Voronoi cell, we consider the closest distance from each critical point to the boundary of simplicial complex and take the average of all these distance values. The computed mean value is used as the clearance value to the $\mathcal{C}_{obst}$, i.e. ϱ, in this work.

B. Classifying ϱ- clearance samples in the $\mathcal{C}_{space}$

Algorithm 1 provides configurations in the $\mathcal{C}_{free}$ that are closer to the obstacle at a distance ϱ using discrete Morse function f on the constructed simplicial complex. Our algorithm considers $\mathcal{C}_{obst}$ present in the $\mathcal{C}_{space}$ and computes the critical morse values to determine the critical points on the boundary of the $\mathcal{C}_{obst}$, as given in equation (1). The computed values for each $\mathcal{C}_{obst}$ determines the identified critical points on it as the extreme values of the function f is reached. The algorithm returns the identified critical points for each $\mathcal{C}_{obst}$ as well as a new graph G_{cp} with the configurations in the $\mathcal{C}_{free}$ at ϱ-clearance to the $\mathcal{C}_{obst}$.

Algorithm 1 ϱ-clearance algorithm

Input: G: complete sampled graph from [3]; O: Obstacle set, D: distance function, Γ: density function, f: Morse values set function, ϱ: value from eq.(4), N: set of configurations around identified critical points.

```
if G is not empty then
  for each obstacle o_i ∈ O do
    for all sample x ∈ G do
      D(x, y) = min_{y∈o_i} ||x − y||        ◁ Refer Def. 3
    for all node y ∈ o_i do
      Γ(y, ϱ) = ⋃_{||x−y||<ϱ} x ∈ G          ◁ Refer Def. 4
      f(y) = D(x, y) × Γ(y, ϱ)
    if f(y) > 0 then
      N = N ⊔ Γ(y, ϱ)
for each neighbor n ∈ N do
  G_cp ← G[n]
return G_cp
```

a) Significance of critical points: In Algorithm 1, we use the identified critical points on the $\mathcal{C}_{obst}$ to keep configurations in $\mathcal{C}_{free}$ close to these critical points within a distance ϱ, i.e. feasible critical points. The feasible critical points obtained from the resulting map is used to connect different sets of paths between start and goal position on a random selection. Each critical point generates a unique set of feasible critical points around these obstacles, and we have used this information and the number of paths incident on the feasible critical point to define path classes in the $\mathcal{C}_{space}$ as illustrated in Figure 2. Path 1 and Path 2 represent different path classes based on the set of feasible critical points incident on them from either on red or green side of $\mathcal{C}_{obst}$.

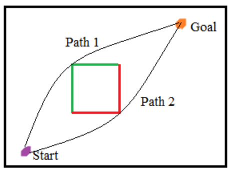


Fig. 2: Path Class Illustration

V. Experimental Setup

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

We performed experiments in 3 different environments with 4 different robots.

- **2D environment**: The 2D environment consists of a point robot with random obstacles in the space, as shown in Figure 3a. This environment was taken from the RRT* paper [4].
- **Mixed-Object environment**: The 3D environment consists of 4 obstacles with different shapes such as a cube, prism, trapezoid, and cylinder placed in it, as shown in Figure 3b. The robot is a 6 DOF cube.
- **Zig-Zag environment**: This environment consists of cluttered obstacles in the environment and has an articulated robot to pass through it. Two different robots are tested in this environment; a 4 DOF robot with a single joint link and a 6 DOF robot with two joint links (Figure 3c).

VI. Experiment Results

In this section, we discuss the results obtained using a pre-processed roadmap built using our method and compare it with different planning methods RRT, RRT*, PRM, and PRM* generating the same number of samples before attempting to query the roadmap to find a path.

A. Pre-processed RoadMap

On applying Algorithm 1 to the roadmap from [3], we output a topological map for each environment with information about the critical points and feasible critical points (i.e., configurations at ϱ-clearance to the $\mathcal{C}_{obst}$).

Once the sampling condition is satisfied, the final sample nodes generated in the 2D environment was 15000 and in the remaining environments was 20000. This map, when processed by our algorithm, generated 8382 nodes in the 2D environment, 12918 nodes in the Mixed-Object environment, 10049 nodes in the 4DOF ZigZag environment, and 9966 nodes in the 6DOF ZigZag environment.

The generated feasible critical points in 2D environment, Mixed-Object environment and ZigZag environment are shown in Figures 3a, 3b, and 3c respectively. The time needed to generate the critical points is very negligible i.e, less than 2 seconds in most of the environment studied. The information gained for all environments during the pre-processing step was used to perform obstacle-avoidance guided path planning, as discussed in the next section.

B. Path Analysis Results

As discussed in section VI-A, we input a pre-processed map of each environment to the different planning strategies and attempt to generate a path from the start to the goal positions of the robot. For the methods we compare with, we set them to produce the same set of samples as with our method (i.e., 15000 in the 2D environment and 20000 in the other environments) before attempting to connect, query and find a path in the environment. This was done to have a fair comparison. In addition, using our critical and feasible critical point information, we categorize all possible paths generated in the 4 testbeds into classes. These classes are based on the paths that include the different set of feasible critical points incident on them. We run 100 random trials that randomly select feasible critical points for each defined path class and generate different path groups of distinguished path classes.

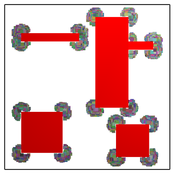
In Figure 4, the total time taken to generate the roadmap and plan a path using our method is smaller than than other methods we compare to. This is in part to the quality of samples returned after using our approach which in turn reduces the number of collision checks.

*1) **Mixed-Object Environment**:* In Table I, the path cost and clearance generated using our method are compared with other methods. Table II show the path classes generated using our approach (paths incident on identified feasible critical points). In this environment, we observe 4 classes (see Figure 5) and also record the average and minimum cost for all paths generated corresponding to each class. We observe that the path returned using our method is closer to the minimum cost for class 1, which records the smallest minimum cost than other classes. This indicates that our method is returning near-optimal paths.

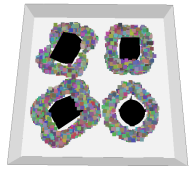
TABLE I: Mixed-Object Environment

	Our Approach map		PRM Sampler map	
Strategy	Path Cost	Clearance	Path Cost	Clearance
RRT	214.9	0.55	215.4	0.65
RRT*	212.8	0.46	211.7	0.63
PRM	249	0.64	246	0.63
PRM*	247	0.58	242	0.69

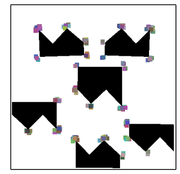
*2) **4DOF ZigZag Environment**:* In Table III, we observe a slight difference in the path cost, whereas the clearance of the paths has a negligible difference between them. From



(a) 2D environment

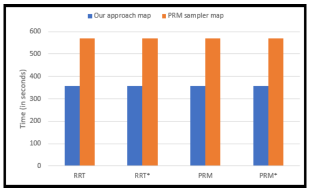


(b) Mixed-Object environment

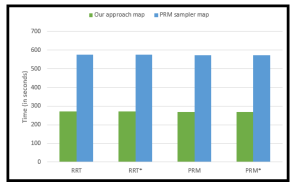


(c) ZigZag environment

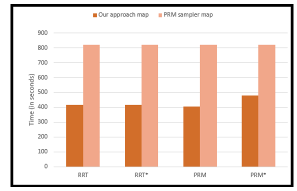
Fig. 3: Environments studied showing feasible critical points around the identified critical points on the boundaries of the $\mathcal{C}_{obst}$.



(a) Mixed-Object environment

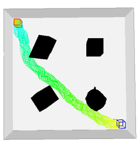


(b) 4DOF ZigZag environment

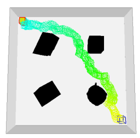


(c) 6DOF ZigZag environment

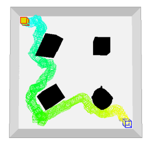
Fig. 4: Time Taken to Build a Roadmap and Plan a Path



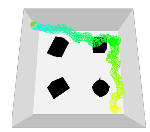
(a) **Class 1**



(b) **Class 2**



(c) **Class 3**



(d) **Class 4**

Fig. 5: Mixed-Object environment Path Classes

TABLE II: Classified paths in Mixed-Object environment

Path Classes	Average Cost	Minimum Cost	Minimum Clearance
Class 1	350.1	211.7	0.48
Class 2	339.3	330	0.64
Class 3	409.8	396	0.54
Class 4	402.8	375	0.54

Figure 6, we observe that all methods generate paths of Class 3 in this environment. Referring to Tables III and IV, we observe a similar behavior as with the Mixed-Object environment; the paths generated with our method and the other methods pick the paths closer to minimum cost of 709.8.

TABLE III: 4DOF ZigZag Environment

	Our Approach map		PRM Sampler map	
Strategy	Path Cost	Clearance	Path Cost	Clearance
RRT	765	0.48	720.1	0.52
RRT*	759.9	0.48	709.8	0.52
PRM	798	0.53	760	0.52
PRM*	786	0.53	757	0.52

Again our method return near-optimal paths, and as seen in Figure 4(b), we generate this path with a fraction of the time needed.

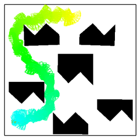
TABLE IV: Classified paths in 4DOF ZigZag environment

Path Classes	Average Cost	Minimum Cost	Minimum Clearance
Class 1	1153.5	957	0.60
Class 2	1441.5	1276	0.51
Class 3	909.48	709.8	0.48

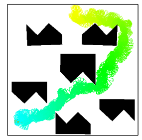
*3) **6DOF ZigZag Environment**:* Table V shows varying results using our method, while RRT and RRT* provide paths in Class 2, PRM, and PRM* generate a path of Class 1 (paths incident on identified feasible critical points). Table VI records the minimum path cost as 900 for Class 1 and 1017 for Class 2 and similar in terms of the minimum clearance recorded as 0.61 for Class 1 and 0.70 for Class 2, which is closely achieved by the methods using our approach map. However, the paths observed for the PRM sampler map were Class 3 for all methods. Additionally, we test incremental path planning for methods without any input map. As seen in Table V, we observe that the path cost for the RRT and RRT* methods are comparable with our approach though slightly better, but, our method enables us to characterise paths with topology information in the $\mathcal{C}_{space}$ which is unavailable with these classical methods. We also notice that our method outperforms the PRM and PRM* with lower path cost and optimal clearance results. The noted time for PRM* on average after 10 runs is 9897.29 seconds with 10494 samples, but our method used 479.58 seconds using a pre-processed map of 9966 samples to generate a path.

TABLE V: 6DOF ZigZag Environment

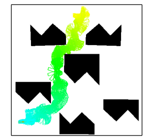
	Our Approach map		PRM Sampler map		Incremental planning	
Strategy	Path Cost	Clearance	Path Cost	Clearance	Path Cost	Clearance
RRT	1107	0.74	772.3	0.60	1019.68	0.6
RRT*	1030	0.73	754.8	0.60	979.25	0.59
PRM	992	0.62	780	0.60	1402.4	1.13
PRM*	968	0.63	776	0.60	1538.6	1.06



(a) **Class 1**

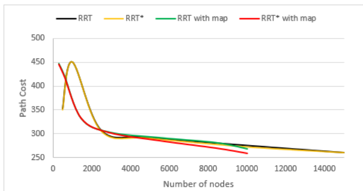


(b) **Class 2**

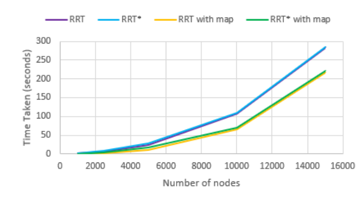


(c) **Class 3**

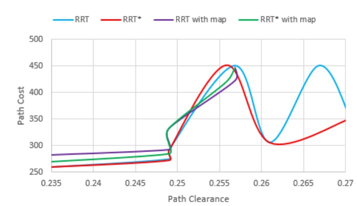
Fig. 6: ZigZag environment Path Classes



(a) **Path cost v/s Number of Nodes**

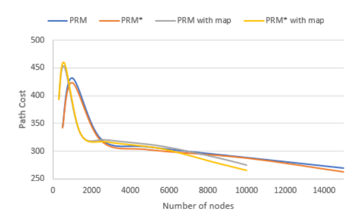


(b) **Time taken v/s Number of Nodes**

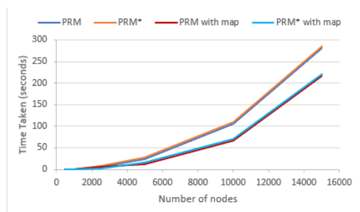


(c) **Path Cost v/s Clearance**

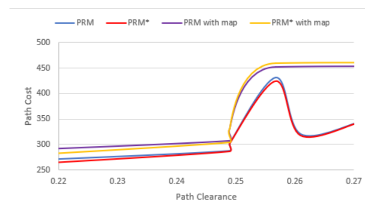
Fig. 7: Plots showing performance of RRT and RRT* in 2D environment.



(a) **Path cost v/s Number of Nodes**



(b) **Time taken v/s Number of Nodes**



(c) **Path Cost v/s Clearance**

Fig. 8: Plots showing performance of PRM and PRM* in 2D environment.

TABLE VI: Classified paths in 6DOF ZigZag environment

Path Classes	Average Cost	Minimum Cost	Minimum Clearance
Class 1	1358.2	900	0.61
Class 2	1781	1017	0.70
Class 3	1376.38	754.8	0.55

*4) **2D Environment**:* For this environment, we compare the performance for the different planners, i.e. RRT, RRT*, PRM, and PRM* using our pre-processed maps with the results from [4] to show the improvement in path quality as the sampling density increases in the 2D environment. The experiment was performed for a sampling density of 500, 1000, 2500, 5000, 10000, and 15000.

a) Path Cost and Time: In Figure 7a, we observe that as the number of samples increases, the path cost decreases and reaches an optimal value using a uniform sampling strategy approach (PRM) map. However, the methods were able to show a similar pattern with a decrease in path cost using our method map, and, as the sampling condition in the last sampling density is achieved, the path cost attains an optimal value. From Figure 7b, we notice that with our pre-processed map, the methods make the faster connection between the nodes (i.e., less collision check) with lesser computation time to plan paths compared to results using the uniform sampling method. A similar trend was observed for PRM and PRM* methods in Figures 8a and 8b.

b) Path Clearance v/s Path Cost: In Figure 7c, as expected, the path clearance decreases with a decrease in path cost using both maps. However, the pattern shows consistency as the values decrease using our method map than compared to the sinusoidal pattern observed using the uniform sampler method (PRM) map. A similar trend was shown in Figure 8c.

VII. Discussion and Future Work

This paper presented an algorithm that applies discrete Morse theory to the vertices of the simplicial complex that help identify critical points on the boundaries of $\mathcal{C}_{obst}$. In this work, we generated a pre-processed map from a sampled graph of the space that provides an approximate measure of the space along with the configuration nodes at the proximity of the obstacles. The result shows that the performance of RRT, RRT*, PRM, and PRM* planners using our algorithm-generated roadmaps provide near-optimal paths with a reduced computation time. Using the feasible critical points, we can provide path classes that are potentially very useful in planning ahead of time based on what paths the robot should take. In our future work, we plan to generate our pre-processed map using an incremental and targeted approach to identify and sample in high priority regions. We plan to utilize these critical points information to provide waypoints towards trajectory planning in state spaces.

References

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

[2] 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.

[3] A. Upadhyay, W. Wang, and C. Ekenna, "Approximating cfree space topology by constructing vietoris-rips complex," in *Proceedings 2019 IROS: IEEE/RSJ International Workshop on Intelligent Robots and Systems (Under publication)*. IEEE, 2019.

[4] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[5] ——, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[6] R. Forman, "A user's guide to discrete morse theory," *Sém. Lothar. Combin*, vol. 48, p. 35pp, 2002.

[7] M. Morse, "Relations between the critical points of a real function of n independent variables," *Transactions of the American Mathematical Society*, vol. 27, no. 3, pp. 345–396, 1925.

[8] M. Kukieła, "The main theorem of discrete morse theory for morse matchings with finitely many rays," *Topology and its Applications*, vol. 160, no. 9, pp. 1074–1082, 2013.

[9] 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.

[10] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," *Int. J. Comput. Geom. & Appl.*, pp. 495–517, 1999.

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

[12] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo, "OBPRM: an obstacle-based PRM for 3d workspaces," in *Proceedings of the third Workshop on the Algorithmic Foundations of Robotics*. Natick, MA, USA: A. K. Peters, Ltd., 1998, pp. 155–168, (WAFR '98).

[13] N. M. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1996, pp. 113–120.

[14] 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.

[15] 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.

[16] H.-Y. C. Yeh, J. Denny, A. Lindsey, S. Thomas, and N. M. Amato, "UMAPRM: Uniformly sampling the medial axis," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, P. R. China, June 2014, pp. 5798–5803.

[17] T. McMahon, S. Thomas, and N. M. Amato, "Motion planning with reachable volumes," Parasol Lab, Department of Computer Science, Texas A&M University, Tech. Rep. 13-001, Jan. 2013.

[18] L. Jaillet, J. Hoffman, J. van den Berg, P. Abbeel, J. M. Porta, and K. Goldberg, "EG-RRT: Environment-guided random trees for kinodynamic motion planning with uncertainty and obstacles," in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, 2011.

[19] A. Upadhyay and C. Ekenna, "Investigating heterogeneous planning spaces," in *Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), 2018 IEEE International Conference on*. IEEE, 2018, pp. 108–115.

[20] C. Ekenna, "Improved sampling based motion planning through local learning," Ph.D. dissertation, Texas A&M University, College Station, Texas, 2016.

[21] C. Ekenna, D. Uwacu, S. Thomas, and N. M. Amato, "Improved roadmap connection via local learning for sampling based planners," in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, Hamburg, Germany, October 2015, pp. 3227–3234.

[22] ——, "Studying learning techniques in different phases of prm construction," in *Machine Learning in Planning and Control of Robot Motion Workshop (IROS-MLPC)*, Hamburg, Germany, October 2015.

[23] C. Ekenna, S. Thomas, S. A. Jacobs, and N. M. Amato, "Adaptive neighbor connection for PRMs, a natural fit for heterogeneous environments and parallelism," Texas A&M, Tech. Rep. TR13-006, May 2013.

[24] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "Prm-rl: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5113–5120.

[25] B. Ichter, E. Schmerling, T.-W. E. Lee, and A. Faust, "Learned critical probabilistic roadmaps for robotic motion planning," *arXiv preprint arXiv:1910.03701*, 2019.

[26] J. D. Gammell, S. S. Srinivasa, and T. 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)*. IEEE, 2015, pp. 3067–3074.

[27] V. Krishnaswamy and W. S. Newman, "Online motion planning using critical point graphs in two-dimensional configuration space," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*. IEEE, 1992, pp. 2334–2339.

[28] V. K. Krishnaswamy, "On-line motion planning in three dimensional configuration space for a robotic manipulator," Ph.D. dissertation, Case Western Reserve University, 1991.

[29] G. Dakin and R. Popplestone, "Simplified fine-motion planning in generalized contact space," in *Proceedings of the 1992 IEEE International Symposium on Intelligent Control*. IEEE, 1992, pp. 281–286.

[30] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *The international journal of robotics research*, vol. 21, no. 4, pp. 331–344, 2002.

[31] N. Shvalb, B. B. Moshe, and O. Medina, "A real-time motion planning algorithm for a hyper-redundant set of mechanisms," *Robotica*, vol. 31, no. 8, pp. 1327–1335, 2013.

[32] K. C. To, K. M. B. Lee, C. Yoo, S. Anstee, and R. Fitch, "Streamlines for motion planning in underwater currents," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4619–4625.

[33] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, 1999, pp. 1024–1031.

[34] T. K. Dey, F. Fan, and Y. Wang, "Graph induced complex on point data," in *Proceedings of the twenty-ninth annual symposium on Computational geometry*, 2013, pp. 107–116.