Approximating C_{free} space topology by constructing Vietoris-Rips complex

Aakriti Upadhyay[1], Weifu Wang[2] and Chinwe Ekenna[1]

***Abstract*— We present a new way of constructing *sparse* roadmaps using point clouds that approximates and measures the underlying topology of the C_{free} space. The main advantage of the constructed roadmap is its homotopy equivalence to the η-offset of the C_{free} space. Though only used to plan paths as a regular roadmap in this work, because the roadmap preserves the topology of the underlying sampled space, the information can be used to plan paths beyond the simple connection of graph vertices. To construct the roadmap, we first sample the configuration space so that the resulting graph is a *n-skeleton* graph that constructs a Vietoris-Rips (VR) complex. Then, we perform a series of topological collapses to remove vertices from the graph while still preserving its topological properties. The resulting roadmaps are used to plan paths for different robots and the experimental results show that the proposed topological approach is faster and more feasible in complex high-dimensional spaces.**

I. INTRODUCTION

One of the fundamental challenge in robot motion planning is the description of the underlying configuration space ($\mathcal{C}_{space}$). Motion planning can become trivial if the geometry and topology of the $\mathcal{C}_{space}$ is fully known. On one hand, geometric information of arbitrary continuous space is hard to describe using a finite amount of fundamental geometric structures. On the other hand, the topological information is hard to represent using tools that can also be used for planning. Even the connectivity is hard to know in the configuration space if either robot or the obstacles are complex.

In this work, we present a new roadmap construction approach where the resulting roadmaps preserve the topological properties of the underlying space. In addition, the resulting roadmap is sparse, yielding a memory-efficient representation. The first stage of the proposed approach is sampling based, which generate samples to construct the Vietories-Rips (VR) complex. In addition to being collision-free, the samples also need to satisfy topological properties that we will introduce in Section III-D, so that the constructed VR-complex using these samples will be topological equivalent to the underlying space. The samples of the VR complex is then trimmed using topological collapses, result in a sparse representation still captures the topological information of the space while being sparse.

The proposed approach take advantage of the *simplicity* of the sampling based approach, where the ease of generating samples is one of main attractions over the previous geometry-based approaches. Modifications have been made to sampling-based approaches to preserve more information of the underlying space. Algorithms such as PRM* and RRT* [26] gained much attraction partially due to the asymptotically optimal properties. However, to provide near-optimal paths, many more samples are needed, and still the roadmaps provide little information about the $\mathcal{C}_{free}$ spaces.

Extracting topological properties of the configuration space is not a new idea and many good results have been presented in [14], [31] and [37]. One of the properties most frequently sought is *persistent homology*, which describes holes, i.e., obstacles, in the configuration space. The aim of our work is not to identify objects and obstacles in the configuration space, but to provide a tool to efficiently describe properties of the $\mathrm{C}_{\mathrm{free}}$ space.

Our approach can briefly be described as follows, First, generate and connect samples, ensuring they satisfy a set of properties describing the *convexity* of the occupied subspace. These properties guarantee that the constructed VR-complexes are homotopy equivalent to the underlying space. Then, remove samples by performing topological collapses while preserving the topological properties. The resulting set of samples gives a roadmap representation of the $\mathrm{C}_{\mathrm{free}}$ space. Finally, we use the roadmap to successfully build a path trajectory with start and goal positions of the robot. In addition, our roadmap is spare in the sense that no edge needs to be stored. Since each vertex belongs to one or more cliques, it is sufficient to only store the associated cliques to fully represent the graph structure without storing any edge.

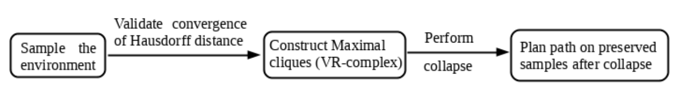


Fig. 1: Our approach

We admit that this is still a preliminary work on using topological tools for motion planning. There are many potential extensions that can be built upon the current results. For example, even though our approach is not currently incremental, the capability of preserving topological information of the sampling space while being memory-efficient can be beneficial for planning in complex spaces. Also, existing approaches currently only finds a path and makes no guarantee about the optimality of the path. Interestingly, we can change/refine the sampling parameters used to generate samples for VR-complex construction if a path that we know exists is not returned by the current roadmap. Increasing the sampling density in the space approximated by the VR-complexes will eventually include this path. We perform experiments in three different environments and compare

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, University at Albany. [2]Weifu Wang is with the Department of Electrical and Computer Engineering, University at Albany. {aupadhyay,wwang8,cekenna}@albany.edu

results with an optimal path planning algorithm; we show improvement in time needed to generate path trajectories for different robot scenarios.

II. Related Work

Topological features are defined as the basic representation of mathematical or geometrical space and refer to features that supports continuity, connectivity, and convergence that is established and maintained based on geometric coincidence. These topological features can be extracted using various mathematical concepts such as sheaf theory [13], persistent homology [16], Vietoris-Rips (VR) complexes and landmarking approach [36]. Past results have shown the benefits of using topological features for improved behaviors or actions of machines in areas like signal processing and cohomology in [32] and, topological motion planning in [31], etc.

To understand the application of homotopy classes for 2D and 3D space objects, Bhattacharya et al. in [6], [8], [9], proposed the use of homology classes for 2D objects, as homotopy classes cannot be practically applied to robot path planning problems. Whereas, they proposed an application of complex analysis and electromagnetism for path planning through 3D objects with genus K (holes in the obstacles) using the concept of homotopy classes. Later in a more practical approach, Bhattacharya et al. in [7], used the concept of persistent homology to find the homology class of trajectories that are most persistent for a given probability map. The work proposed persistent homology to solve the fundamental problem of goal-directed path planning in an uncertain environment represented by a probability map.

Research by Pokorny et. al. in [30], considered homotopy classes of trajectories in general configuration space using Delaunay-Cech Complex filtration and abstracted the global information about trajectories using persistent homology. Our work avoids the use of Delaunay-Cech Complex filtration due to the difficulty in computing them for complex spaces, i.e. the curse of dimensionality. Pokorny et. al. further showed in [31], the application of a sampling-based approach to topological motion planning that is fully data-driven in nature. The work also uses the Delaunay-Cech method to filter the data from the point-cloud dataset and improvises Dijkstra's algorithm to generate distance-vector terminology for a source vertex.

The above-cited research has shown improvements inclined towards extracting a topological description of the space and then performing approximate sampling with performance guarantees. These methods, however, do not provide a measure of the approximation that has been performed.

Sampling-based methods [15] are a state-of-the-art approach to solving motion planning problems. These methods are known to be probabilistic complete because the probability of finding a solution if it exists tends towards 1 as the number of samples generated increases. Sampling-based methods are broadly classified into two main classes: graph-based methods such as the Probabilistic Roadmap Method (PRM) [27] and tree-based methods such as Expansive-Space tree planner (ESTs) [24] and Rapidly-exploring Random Tree (RRT) [28]. PRM variants include topologies such as uniformly generating samples in the environment [27], sampling near obstacles [2], [4], [12], [23], [35], sampling with constraints placed on the robots [29] and planning with uncertainty in the environment [25]. Other methods exist that investigate the heterogeneous nature of the planning environment using reinforcement learning [17]–[20], [34].

III. Preliminaries

A. VR-complex and Čech-complex

Attali et. al., [5] have proven that VR complexes can provide topologically correct approximations of shapes utilizing the notion of distances between points in the metric space. The research provides conditions under which the VR complex of the point set at some scale reflects the homotopy type of the shape for a finite point set that samples a shape. Formally, the VR and $\check{C}$ech complex can be defined as follows:

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, whereas the $\check{C}$ech-complex C(X) is the abstract simplicial complex where a subset of k + 1 points in X determines a k-simplex if and only if they lie in a ball of radius ε/2.

B. Simplicial collapses

An abstract simplicial complex K, i.e., a collection of sets closed under the subset operation, is a generalization of a graph and is useful in representing higher-than-pairwise connectivity relationships. The elements of any set are called vertices and the set itself is called a simplex. Topological thinning (simplicial collapse) [10] is an operation that shrinks simplicial complexes to homotopy-equivalent subcomplexes. In this work, the simplicial collapse will be used to reduce the complexity of maximal simplices through vertex deletion down to a core simplex on maintaining the topological structure of the configuration space.

C. Hausdorff Distance

The Hausdorff distance measures how far two subsets of a metric space are from each other [1]. In this work, we measure Hausdorff distance (ϵ) between set P – sampled points, and set X – the $\mathcal{C}_{free}$ space. The algorithm uses a convex hull method to find the boundary points of set P to compute the closest distance ϵ between sample points and $\mathcal{C}_{space}$ boundary. In figure 2, the blue line represents the boundary of the $\mathcal{C}_{space}$ and the green line as the boundary of point cloud set P (calculated using convex hull). As the sample points get denser in the $\mathcal{C}_{space}$, the value of ϵ decreases and becomes constant above a certain sampling density.

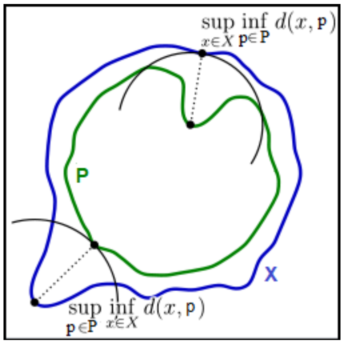


Fig. 2: Hausdorff distance for set P and X

D. From VR complex to sampled-space topology

Generally, a VR-complex does not preserve the topology of the underlying sampled space. However, in [5], the authors showed that a VR complex can be retracted to a Čech complex to approximate the topology of the underlying sampled space. Let us define the *flag complex* of a graph G, denoted Flag G as the maximal simplicial complex whose 1-skeleton is G. More precisely, this is the largest simplicial complex sharing with the Čech complex the same 1-skeleton. In addition, let us denote the VR-complex $R(P,t)$ the abstract simplicial complex whose k-simplices correspond to subsets of $k+1$ points in P with a diameter that is at most $2t$. The Čech complex $C(P,t)$ as the abstract simplicial complex whose k-simplices correspond to subsets of $k+1$ points that can be enclosed in a ball of radius t. Define α as an *inert value* of P if $\mathrm{Rad}(\delta) \neq \alpha$ for all non-empty subsets $\delta \subset P$.

Then, given any point set $P \in \mathbb{R}^n$ and any real numbers $\alpha, \beta \geq 0$ with $\alpha \leq \beta$, define the flag complex of any graph G satisfying $R(P,\alpha) \subset \text{Flag } G \subset R(P,\beta)$ an (α,β)-quasi Rips complex of P. Also, let $v_n = \sqrt{\frac{2n}{n+1}}$. We can have the following property, which is Theorem 7 from [5].

Theorem 1. *Let $P \subset \mathbb{R}^n$ be a finite set of points. For any real numbers $\beta \geq \alpha \geq 0$ such that α is an inert value of P and $c_P(v_n\beta) < 2\alpha - v_n\beta$, there exists a sequence of collapses from any (α,β)-almost Rips complex of P to the Cech complex $C(P,\alpha)$.*

The measure of convexity defects of X at a given scale is determined by function c_p as given below.

$$c_p(t) = d_H(\mathrm{Centers}(X,t)|X) \tag{1}$$

Further, the graph can be shown to be homotopy equivalent to η-offset of the sampling space X, from Theorem 10 in [5].

Theorem 2. *Let ϵ, α and β be three non-negative real numbers such that $\alpha \leq \beta$ and $\eta = 2\alpha - v_n\beta - 2\epsilon > 0$. Let P be a finite set of points whose Hausdorff distance to a compact subset X is ϵ or less. Then, any (α,β)-quasi Rips complex of P is homotopy equivalent to the η-offset of X whenever α is an inert value of P and $h_X(v_n\beta+\epsilon) < 2\alpha - v_n\beta - 2\epsilon$.*

where $\mathrm{Hull}(X)$ denotes the convex hull of X, and

$$h_X(t) = d_H(\mathrm{Hull}(X,t)|X) \tag{2}$$

$$Hull(X,t) = \bigcup_{\substack{\emptyset \neq \delta \subset X \\ \mathrm{Rad}(\delta) < t}} \mathrm{Hull}(\delta) \tag{3}$$

From the theorem, we can derive that in order to use a graph-like structure to approximate the homotopy of the sampling space, we need to first have sufficiently dense samples, so that P is no more than ϵ away from the set X based on Hausdorff distance. Here, X is the set we would like to approximate using samples in P. Recall, Hausdorff distance $d_H(X,Y)$ is

$$\begin{aligned} d_H(X,Y) &= \max\{\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(x,y)\} \\ d(y,X) &= \inf_{x\in X} d(y,x) \\ d_H(Y|X) &= \sup_{y\in Y} d(y,X) \end{aligned}$$

Therefore, if the samples P satisfy the above properties, we can construct a graph based on P and use the relations to approximate the underlying homotopy of X, even when the number of samples is small. Compared to the sampling-based motion planning approaches, where the connectivity is *guaranteed* when the number of samples reaches infinity, the proposed method yields a bound on the number of samples. On the other hand, given a set of samples P, we can also compute the relevant parameters to derive how much of the sample space X has the samples covered, where X can be C_{free} in the case of motion planning.

IV. Proposed Roadmap Construction

To construct the proposed roadmaps that approximates the topology of the underlying configuration space, we first densely place samples that will satisfy the parameters presented in Theorem 2. We then use the samples to construct VR complex, which approximates the homotopy of the underlying $\mathcal{C}_{free}$ space. Finally, we perform topological collapses to remove the unnecessary samples from the VR-complexes to output a sparse roadmap.

The sampling process is similar to that of a PRM algorithm, with additional requirements, mainly the conditions mentioned in Theorem 1 and 2. During the sampling process, the Hausdorff distances are computed between the samples and the set X, which is the $\mathcal{C}_{free}$ space in our application. In [5], the authors stated that if the Hausdorff distance is smaller than the parameters measuring the space, the resulting point cloud (sampled points) approximates the topology of the space. Taking this into consideration (Theorem 2), we validate the expression $2\epsilon < 2\alpha - v_n\beta$, where $\beta = \alpha$ in our experiments. On verifying the sampling condition in the workspace, the output is a densely sampled $\mathcal{C}_{space}$ graph G.

A. Collapsing a VR-complex

The convex hull of any nonempty subset of the $n+1$ points that define an n-simplex is called a face of the simplex (complex). A maximal face (facet) is any simplex in a complex that is *not* a face of any larger simplex. Given $\tau, \delta \in K$, if $\tau \subset \delta$, in particular $\dim \tau < \dim \delta$, and δ is a maximal face of K and no other maximal face of K contains τ, then τ is called a *free face*. A *simplicial collapse* of K is the removal of all simplices γ such that $\tau \subseteq \gamma \subset \delta$. The work in [36] and [37] explains the equivalence between maximal faces in abstract simplicial complexes and maximal cliques in graph theory.

Given a simplicial complex K of dimension $n \geq d$, a d-skeleton of K is the subcomplex of K consisting of all the faces of K that have dimension at most d. Then, a graph can be used to represent the 1-skeleton of K, and let us refer to the graph as the *underlying graph* and denote the graph as G_K. For simplicity in this work, we will refer to the 0-skeleton of K as vertices of G_K, and 1-skeleton of K as edges of G_K. Then, we can derive the following results.

Lemma 1. *Given a complex K and its underlying graph G_K, let δ be a maximal face of K, if a vertex v of G_K is a subset of δ ($v \subset \delta$) and no other maximal face of K contains v, then there exist a sequence of simplicial collapses on K that can remove vertex v.*

Proof: Let there exist a sequence of free faces $s_0, s_1, s_2, \ldots, s_m$, so that $s_0 \subset s_1 \subset s_2 \subset \ldots \subset s_m \subset \delta$ and $s_0 = v$. Let s_1 be one of the edges on G_K with v being one endpoint of the edge, let s_2 be the tetrahedron containing s_1, etc. Because each s_i is a free face, a simplicial collapse can remove it. Then, let the sequence of collapse start from s_m, and move towards s_0. Each collapse of s_i will not change the fact that s_i still is a free face of δ. Therefore, v can be removed. □

Then, we can extend the results to get the following theorem.

Theorem 3. *Given a complex K and its underlying graph G_K, let δ be a maximal face of K, and let V_s be the set of all the vertices v where v is a subset of δ and no other maximal face of K contains v. Then, after removing all vertices in V_s, there are no free faces on δ.*

Proof: Let us assume that after removing all vertices in V_s, there still exists at least one free face $\tau \subset \delta$. If τ is of dimension 0, then it is a vertex that only belongs to δ, so it must have been part of V_s, so τ can not be of dimension 1 or above. If τ is of dimension 1, i.e. an edge on G_K, then at least one vertex of the edge will belong only to δ otherwise the edge cannot be a free face. Therefore, removing all vertices of V_s will remove this edge. Inductively, we can extend this to higher dimensions. Therefore, there cannot be any free face left after removing all vertices in V_s. □

B. Removing topologically unimportant vertices

Similar to the work in [36], Algorithm 1 first constructs VR-complex using maximal clique technique for faster computation. After the cliques are computed, we transform the representation of the cliques to binary. Each node in the graph was represented in binary form based on the clique in which it belongs. Given a graph with n nodes, the binary representation of a clique (or a sub-graph) is the binary string of length n in which the i^{th} character is 1 if the clique (sub-graph) contains the i^{th} node and 0 otherwise. We then perform bit-wise and operation to find potential simplicial collapses using results from Lemma 1. We remove the vertices labeled as 1 after the operations among cliques.

The algorithm returns a sampled graph G_{new} with vertices of non-colliding regions of $\mathcal{C}_{space}$ after completing topological collapse on the graph. This resulting graph gives an approximate topological shape representation of the objects and available free region in the $\mathcal{C}_{space}$.

Algorithm 1 Graph-Collapse(G)

Input: G: sampled graph from the point cloud; M: maximal clique, B: set of binary representation for each clique; T: set of vertices after topological collapse.

for all nodes in graph G **do**
  compute maximal clique M.
while M is not empty **do**
  for each clique in M **do**
    if node in clique **then**
      Set binary value '1' for node in B
    else
      Set binary value '0' for node in B
if B is not empty **then**
  T = B $\oplus$ B
  for each node in T **do**
    project node in graph G_{new}.
return G_{new}

V. Experiments and Results

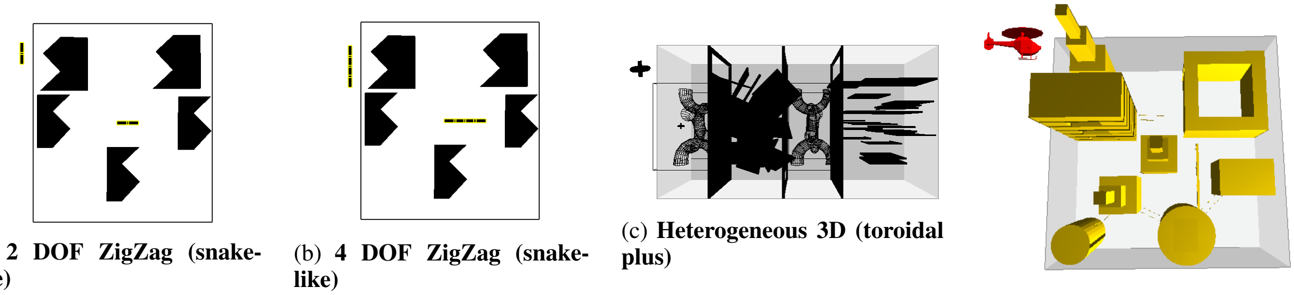
A. Experimental Setup

All experiments were executed on a Dell Optiplex 7040 desktop machine running OpenSUSE operating system and were implemented in C++. We performed experiments in three different environments as shown in Figure 3 and generated samples ranging from 100 to 10,000. The environments are taken from the Parasol Lab benchmarks at Texas A & M University [3].

- **ZigZag environment**: 2D environment with structured obstacles placed randomly as shown in Figure 3a and 3b. We tested two configurations, one with a 2 DOF robot and one with a 4 DOF robot.
- **Heterogeneous 3D**: 3D maze environment with walls and narrow passages between the walls. A robot with a toroidal shape has to pass through maze-like tunnels to reach the goal as shown in Figure 3c.
- **Helico**: A city representation with tall buildings and wires between buildings (Figure 3d). The robot is a rigid body representation of a helicopter and can change its vertical position based on the goal position.

B. VR Complex computation

We performed preliminary experiments with two libraries that generate VR-complexes. We compared results to determine which library is most suited for our approach.



(a) **2 DOF ZigZag (snake-like)** (b) **4 DOF ZigZag (snake-like)** (c) **Heterogeneous 3D (toroidal plus)** (d) **Helico (helicopter)**

Fig. 3: Environments Studied

We used the VR-complex package of GUDHI library [11] to construct simplicial complexes. The time complexity of the algorithm is $O(v^2d + m^2d)$, where d is the dimension of the complex, v is the number of vertices, and m is the number of maximal simplices in the graph.

The Quick-cliques library [21], [22] generates maximal cliques using a modified Bron-Kerbosch algorithm by Tomita et. al. [33]. We used a hybrid algorithm that applies a VR-complex approach to construct simplices. The time complexity of the algorithm is $O(3^{d/3}nd)$ with n vertices and degeneracy d. Since VR-complexes are also known as clique complexes, the algorithm tries to generate maximal cliques as a result.

Table I and II compares results for GUDHI and Quick-cliques library in 2 DOF ZigZag environment with and without obstacles present. One can see that Quick-cliques library computes maximal cliques faster than the GUDHI library as the number of nodes increases hence we chose to use the Quick-cliques library for the remaining cases.

TABLE I: Constructing Rips complex in a 2 DOF ZigZag environment without obstacles

Library	Number of Nodes	Cliques	Time taken (sec)
GUDHI	100	210	**0.01**
Quick-Cliques	100	51	0.042274
GUDHI	10000	33552695	289.31
Quick-Cliques	10000	892190	**0.014901**

TABLE II: Constructing Rips complex in 2 DOF ZigZag environment with obstacles

Library	Number of Nodes	Cliques	Time taken (sec)
GUDHI	100	268	**0.02**
Quick-Cliques	100	1378	0.042939
GUDHI	10000	81463172	675.6
Quick-Cliques	10000	1443062	**50.072735**

C. Sampling at different densities

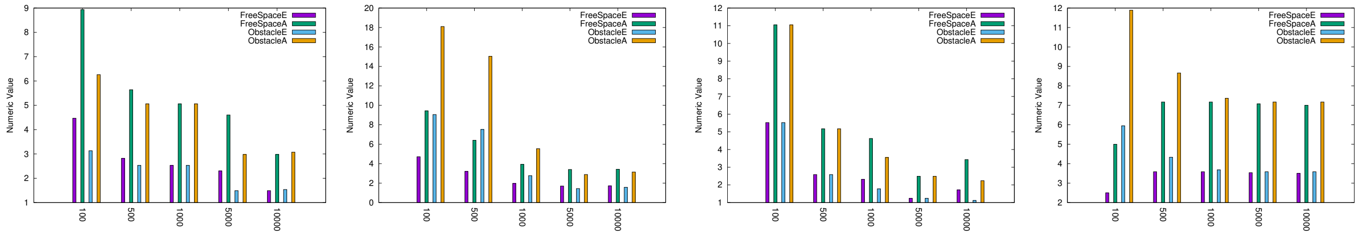
We performed two sets of experiments on our three testbeds, all with and without obstacles in the environment. We first performed experiments for the sampling conditions of P based on the $2\epsilon < 2\alpha - v_n\beta$, where $\beta = \alpha$ preconditions as previously discussed in Section IV. Another condition as defined in [5], states that as the sampled space becomes denser, the Hausdorff distance (ϵ) reduces or approaches a constant value. Secondly, we constructed a (space) graph G from a point cloud that densely sampled the space and then performed topology collapse. Our results show that after a topology collapse, the coverage of $\mathcal{C}_{space}$ is not compromised.

1) Sampling conditions: In Figure 3 and 4, the Hausdorff distance (ϵ) decreases in an empty environment as well as in an environment with obstacles. The trend as shown in Figure 4d clearly satisfy the conditions stated in [5] which state "the value of ϵ will become constant above the radius of the circle covering the $\mathcal{C}_{space}$". The purple and blue bars (2ϵ)(E) and the green and yellow bars ($2\alpha - v_n\beta$) (A) in the histogram represented in Figure 4a to 4d show that in all cases both the above conditions are satisfied.

In the particular case of the Helico environment as seen in Figure 4d, the ϵ value is initially low and subsequently increases as the graph becomes denser before leveling off and then becoming constant. The position of a robot in this environment is at the corner of the $\mathcal{C}_{space}$, so when samples are generated initially, they are generated only near the boundary of the $\mathcal{C}_{space}$ and hence ϵ value is low until the number of samples increases in the environment to produce better coverage. The values of ϵ converges to constant as it reaches 10000 sampled nodes in all the environments as shown in Figure 5.

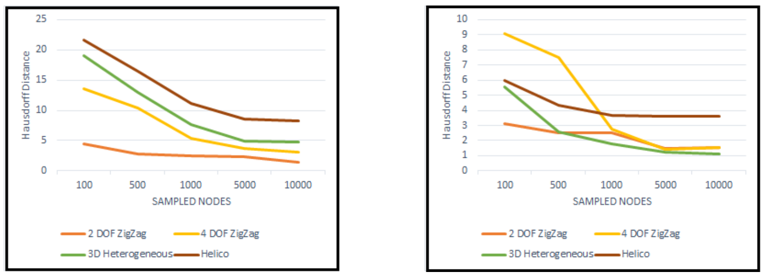
2) Topology Collapse: Table III contains results for topology collapse experiments that utilize theorems and algorithms presented in Section IV-A and IV-B. The results substantiate the ability to delete vertices thus confirming Lemma 1. We show a 40 to 90% reduction across all the environments which indicate that our method can delete vertices while retaining the topological information of the space.

Figure 6 gives a pictorial representation for one studied environment after the graph topology collapse. The sub figures show the process from performing a topology collapse and getting a path for a simple robot. Figure 6 (i) a $\mathcal{C}_{space}$ with obstacles, (ii) a 1-skeleton with five or few samples in $\mathcal{C}_{free}$, (iii) a 0-skeleton densely sampled graph, (iv) the structure that remains after the topological collapse, and (v) a successful path through the $\mathcal{C}_{space}$.



(a) **2 DOF ZigZag** (b) **4 DOF ZigZag** (c) **Heterogeneous 3D** (d) **Helico**

Fig. 4: ϵ and α trends in obstacle and free environments



(a) $\mathcal{C}_{free}$ **environment** (b) $\mathcal{C}_{obst}$ **environment**

Fig. 5: Convergence of Hausdorff distance in obstacle and free environments

Environment	Nodes Before	Nodes After- Free	% Reduction	Nodes After- Obstacle	% Reduction
2DOF Zig Zag	10,000	5081	49.2	4826	51.7
4DOf Zig Zag	10,000	637	93.6	896	91.1
Heterogeneous 3D	10,000	4968	50.3	5061	49.3
Helico	10,000	5041	49.6	5023	49.8

TABLE III: **Results after the Topology Collapse in the Free and Obstacle Environment**

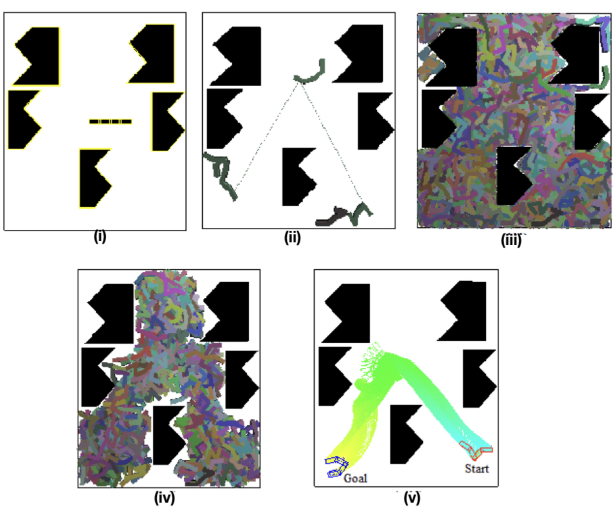


Fig. 6: Phases of Topological path planning

In addition, because the samples we generated were used to construct a VR-complex, which consists of locally complete subgraphs (cliques), we can skip the storage of all the edges and store only which cliques a vertex belongs to. Therefore, the storage needed to store the entire resulting roadmap scales linearly with the number of samples left after the collapse, which is comparable to k-nearest neighbor PRM but provides much richer topological information [36].

D. Planning with homology equivalent samples

Table IV and V compare paths generated by PRM* [26] using (i) the original point cloud and (ii) the vertices of the VR-complex after topological collapse in different environments in terms of total path cost and time needed to build a path. We report time to connect and query the environment alone to allow for fairness in our comparisons. The results show an order of magnitude improvement in all environments studied.

VI. Discussion and Future Work

The work presented derives from studies that show the circumstances under which Vietoris-Rips Complex and the $\check{C}$ech Complex have homotopy equivalence beneficial to improving approximate sampling algorithms and gives a much-needed measure of this approximation. The reconstructed $\mathcal{C}_{space}$ has proven to be helpful in path planning while reducing the computation time and memory.

The approach can have application in a dynamic real-world environment where the path planning of a robot can be performed with minimum computation time on smaller portions of the environment that a robot can view at the

$\mathcal{C}_{free}$ Environment	Our Approach	PRM*	$\mathcal{C}_{obst}$ Environment	Our Approach	PRM*
2DOF ZigZag	**62.1342**	229.922	2DOF ZigZag	**53.0861**	129.904
4DOf ZigZag	**64.7357**	11146.4	4DOf ZigZag	**2.52541**	24089.9
Heterogeneous 3D	**62.6602**	DNF	Heterogeneous 3D	DNF	DNF
Helico	**55.8919**	DNF	Helico	**58.4688**	82967

TABLE IV: **Path planning time (in seconds) in the Free and Obstacle Environments**

$\mathcal{C}_{free}$ Environment	Our Approach	PRM*	$\mathcal{C}_{obst}$ Environment	Our Approach	PRM*
2DOF ZigZag	**1003**	1438	2DOF ZigZag	**827**	1553
4DOf ZigZag	**916**	1324	4DOf ZigZag	**893**	1258
Heterogeneous 3D	**3714**	DNF	Heterogeneous 3D	DNF	DNF
Helico	**1806**	DNF	Helico	**1338**	2698

TABLE V: **Path planning cost in the Free and Obstacle Environments**

time of traversal and combine together to get a better understanding of the actual environment.

In future work, we will further enhance the approach to identify critical points in $\mathcal{C}_{space}$, i.e. sample points closest to the $\mathcal{C}_{space}$ curvature, and using the properties of Vietoris-Rips, perform path planning on smaller sized graphs.

REFERENCES

[1] Hausdorff distance. `https://en.wikipedia.org/wiki/Hausdorff_distance`, accessed: 2018-01-30

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

[3] Amato, N.M.: Motion planning benchmarks, http://parasol.tamu.edu/groups/amatogroup/benchmarks/

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

[5] Attali, D., Lieutier, A., Salinas, D.: Vietoris–rips complexes also provide topologically correct reconstructions of sampled shapes. Computational Geometry 46(4), 448–465 (2013)

[6] Bhattacharya, S.: Identification and representation of homotopy classes of trajectories for search-based path planning in 3d (2011)

[7] Bhattacharya, S., Ghrist, R., Kumar, V.: Persistent homology for path planning in uncertain environments. IEEE Transactions on Robotics 31(3), 578–590 (2015)

[8] Bhattacharya, S., Likhachev, M., Kumar, V.: Topological constraints in search-based robot path planning. Autonomous Robots 33(3), 273–290 (2012)

[9] Bhattacharya, S., Lipsky, D., Ghrist, R., Kumar, V.: Invariants for homology classes with application to optimal search and planning problem in robotics. Annals of Mathematics and Artificial Intelligence 67(3-4), 251–281 (2013)

[10] Björner, A.: Topological methods. Handbook of combinatorics 2, 1819–1872 (1995)

[11] Boissonnat, J.D., Pritam, S., Pareek, D.: Strong collapse for persistence. arXiv preprint arXiv:1809.10945 (2018)

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

[13] Bredon, G.E.: Sheaf theory, vol. 170. Springer Science & Business Media (2012)

[14] Chambers, E.W., De Silva, V., Erickson, J., Ghrist, R.: Vietoris–rips complexes of planar point sets. Discrete & Computational Geometry 44(1), 75–90 (2010)

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

[16] Edelsbrunner, H., Morozov, D.: Persistent homology: theory and practice. Tech. rep., Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States) (2012)

[17] Ekenna, C.: Improved Sampling Based Motion Planning Through Local Learning. Ph.D. thesis, Texas A&M University, College Station, Texas (2016)

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

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

[20] Ekenna, C., Uwacu, D., Thomas, S., Amato, N.M.: 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)

[21] Eppstein, D., Löffler, M., Strash, D.: Listing all maximal cliques in large sparse real-world graphs. Journal of Experimental Algorithmics (JEA) 18, 3–1 (2013)

[22] Eppstein, D., Strash, D.: Listing all maximal cliques in large sparse real-world graphs. In: International Symposium on Experimental Algorithms. pp. 364–375. Springer (2011)

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

[24] Hsu, D., Latombe, J.C., Motwani, R.: Path planning in expansive configuration spaces. Int. J. Comput. Geom. & Appl. pp. 495–517 (1999)

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

[26] Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. The international journal of robotics research 30(7), 846–894 (2011)

[27] Kavraki, L.E., Švestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Automat. 12(4), 566–580 (August 1996)

[28] LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA). pp. 473–479 (1999)

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

[30] Pokorny, F.T., Hawasly, M., Ramamoorthy, S.: Multiscale topological trajectory classification with persistent homology. In: Robotics: science and systems (2014)

[31] Pokorny, F.T., Kragic, D.: Data-driven topological motion planning with persistent cohomology. In: Robotics: Science and Systems (2015)

[32] Robinson, M.: Understanding networks and their behaviors using sheaf theory. In: Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE. pp. 911–914. IEEE (2013)

[33] Tomita, E., Tanaka, A., Takahashi, H.: The worst-case time complexity for generating all maximal cliques and computational experiments. Theoretical Computer Science 363(1), 28–42 (2006)

[34] Upadhyay, A., Ekenna, C.: Investigating heterogeneous planning spaces. In: Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), 2018 IEEE International Conference on. pp. 108–115. IEEE (2018)
[35] Yeh, H.Y.C., Denny, J., Lindsey, A., Thomas, S., Amato, N.M.: UMAPRM: Uniformly sampling the medial axis. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA). pp. 5798–5803. Hong Kong, P. R. China (June 2014)
[36] Zomorodian, A.: Fast construction of the vietoris-rips complex. Computers & Graphics 34(3), 263–271 (2010)
[37] Zomorodian, A.: Topological data analysis. Advances in applied and computational topology 70, 1–39 (2012)