Investigating Heterogeneous Planning Spaces

Aakriti Upadhyay and Chinwe Ekenna

Abstract— **With the continuous improvement of the capabilities of robots and the increasing complexity of the environments they successfully traverse, this paper presents useful concepts and definitions about the heterogeneous nature of planning spaces within the context of motion planning. Our methodology uses the property of visibility, expansiveness and homotopy class to develop algorithms that represent the heterogeneity of the planning space. Our algorithm also include a machine learning technique that identifies sub regions and then intelligently applies necessary existing strategies to create well connected maps in that sub region. We make comparisons with two other machine learning methods in a variety of simulated robot environments ranging from simple homogeneous rooms to complicated maze environments. Our method outperforms the other two methods in terms of time to build a roadmap, the number of nodes needed and the number of connected components generated.**

I. Introduction

An important research area in robotics is motion planning, navigation and localization which aids in the quest for precision and accuracy in robots. Planning motions is needed in many disciplines such as planning for deformable robots [18], [33], [36], manipulation planning [24], character animation for games and movies, and virtual prototyping. A robot is a movable object where the position and orientation is defined by a set of parameters such as the degrees of freedom, angles and displacement of its links. The robot's configuration can be described based on the configuration space (the set of all possible placement of the robot) which contains both feasible and infeasible regions of the space.

These research areas and tasks present experts with a challenge of deciding the best algorithms to use given varying heterogeneous environments that exist in nature e.g., a surgical needle navigating organs to get to the heart during surgery, a rover moving among debris to rescue or retrieve humans in disaster areas. Unfortunately, most of these algorithms are not adaptable to different scenarios. Previous work [2], [9], [22] show the need to explicitly define parameters used in different algorithms ahead of time which is difficult to do.

There is increasing concern about the development of a variety of motion planning algorithms [26], [29], [34], perception [17], [42] and localization [4], that have very little scalability for success outside of predefined and explicitly defined parameters. There is some evidence to show that combining different algorithms [2], [9], [22] in a heterogeneous environment improve performance [16] but this has been applied with only intuitive knowledge of the environment.

There are algorithms that look into the different states, regions and features of the environments for these robots developed overtime [8], [14], [38], with a specific interest e.g., narrow passage, homotopy classes, but not necessarily considering the heterogeneity. The placement of discrete objects (obstacles) (i.e. varying in shape, size, volume and dimension) randomly in C-Space forming combination of either C-free, narrow passage, cluttered or blocked regions defines the term "heterogeneity" within the context of this paper.

Ekenna et. al in [16], presented an algorithm that used machine learning concepts to help partition the configuration space (C-Space) into homogeneous pieces which produced some improved results but the underlying problem of determining if an environment is heterogeneous remains. In another work Bhattacharya et. al [5], attempts to generate homotopy classes of the environment which helped determine the feasibility of an object transitioning from one state to the next in a given environment.

In this paper, we use the property of visibility [20], expansiveness and homotopy classes [5], to analyse and investigate the heterogeneity of the planning space. This informed knowledge of the planning space is used in conjunction with a reinforcement learning technique to determine the best planning strategy to apply to different identified regions. We compare our work with HybridPRM [16], [22] and UtilityGuidedSampling [10], [11] strategies, and our results indicate better performance in terms of time to solve query, number of nodes/configuration generated and number of connected components present in the map.

II. Related Work

In this section we discuss motion planning primitives related to our work to include C-Space properties such as expansiveness and visibility, and the notion of homotopy class in relation to the C-Space.

A. Motion Planning Preliminaries

The motion planning problem involves finding a valid path (e.g., collision-free and satisfying all joint limit and/or loop closure constraints) for a movable object starting from its start configuration to a goal configuration in an environment [13]. A single configuration is defined based on the movable object's d independent parameters or DOFs (degrees of freedom).

A robot is a movable object whose position and orientation can be described by n parameters, or degrees of freedom(DOFs), each corresponding to an object component

C.Ekenna, is affliated with the Department of Computer Science and Engineering, Texas A & M University, Texas, USA.

C.Ekenna, A.Upadhyay are with the Department of Computer Science, University at Albany, New York, 12222, USA. {cekenna, aupadhyay}@albany.edu

(e.g., object positions, object orientations, link angles, link displacements). Hence, 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 ith DOF). This space consisting of all possible robot configurations (feasible or not) is called configuration space (C-Space) [28]. The subset of all feasible configurations is the free space (C-free), while the union of the unfeasible configurations is the blocked or obstacle space (C-Obstacle). Thus, the motion planning problem becomes that of finding a continuous trajectory in C-free connecting start and goal pair of configurations.

1) Sampling Based Motion Planning: Sampling-based motion planning methods [13] are a state-of-the-art approach to solve motion planning problems. These methods are known to be probabilistically complete because even though there is no guarantee to find a solution if one exists, the probability of finding a solution if it exists increases as the number of samples generated also increase. Sampling-based methods are broadly classified into two main classes: graph-based methods such as the Probabilistic Roadmap Method (PRM) [25] and tree-based methods such as Expansive-Space tree planner (ESTs) [21] and Rapidly-exploring Random Tree (RRT) [27]. PRM variants consider different topology which include uniformly generating samples in the environment [25], sampling near obstacles [2], [3], [9], [19], [41], sampling with constraints placed on the robots [31] and planning with uncertainty existing in the environment [23].

B. C-Space Properties

Two configurations q and q' forming a path or a continuous sequence of adjacent configurations existing between them is termed as visible. Two configurations q and q' are therefore connectable if there exists a continuous sequence of configurations that are also visible to each other. In such a case, we define connectable (q, q') = true. We define a connected component CC of C-free as a maximal subset of configurations $CC \subseteq$ C-free such that $\forall q, q' \in CC$, $connectable(q, q') = true$. C-free may comprise of one or more connected components.

1) Visibility Properties: The degree to which two configurations can be said to be connectable defines the visibility of two or more nodes in the C-Space. This could be based on e.g., distance, radius or topology between configurations. Any configuration q is said to be ε - good [20], if it is connected to another visible random configuration q_{rand} using the same sampling method for q e.g., uniform sampling. such that an edge connecting configuration points is collision-free or exists in the absence of obstacle.

2) β - goodness: The volume of β-LOOKOUT(G) is $\alpha\mu$(G), where G is the subset of connected components in the C-Space and μ represents the volume of subset G. If either α or β is small, then it would be difficult to sample points and build roadmap between the connected components.

C-Space is said to be ε, α, and β-expansive, if every point q in connected components is ε-good and its connected components are all ε, α, and β-expansive. When ε, α and β are reasonably large, it guarantees larger lookout or expansiveness of C-Space.

Research by Nissoux in [39], outlines visibility for a mechanical system moving in a workspace. The algorithm incrementally constructs visibility roadmaps by randomly sampling the configuration space and connects collision-free samples using the local planner that considers the free space representation of the configuration space. Denny et al [15] introduces visibility during RRT construction and define visibility as an estimation of how easy it is to connect a configuration q to the other configurations in its local surroundings.

C. Homotopy Classes

A set of trajectories (or paths), q_0 to q_1, through q_2, q_3, q_4, q_5, q_6 as shown in Figure 1 are said to be in the same homotopy class [5] iff one can be smoothly deformed into the other without intersecting obstacles, otherwise they belong to different homotopy classes.

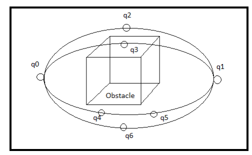


Fig. 1: **Homotopy Class of trajectories**

In Figure 1, the paths connecting points q_0 to q_1 through q_2 and q_3 or q_4, q_5 and q_6 are in the same homotopy classes respectively whereas paths connecting $q_0q_2q_1$ and $q_0q_4q_5q_1$ are in different homotopy classes as they enclose object between them.

Homotopy class definitions of the C-Space has been investigated in [6], [7] where representative trajectories are identified.

Pokorny and Kragic [35] use cohomology as a tool for classifying preexisting trajectories and create a topological motion planing method. Salzman et. al [38] propose an algorithmic framework that combines geometric methods for exact and complete analysis of low dimensional C-Space with sampling based approaches that are also appropriate in higher dimensions. Manifolds are decomposed into cells in C-free and C-Obstacle regions and using the notion of homotopy class definitions. This work however gives no clear evidence for any number of clearance path in any given region of the planning space.

The algorithms in the cited papers has a foundation in expansiveness, visibility and homotopy classifications of the C-Space, our algorithm uses these properties to formally define the heterogeneity of the planning space. Our heterogeneous properties in this paper can be integrated into any sampling based planning method and helps to inform in an automated fashion the best methods suitable for any environment scenario. In this paper we integrate these properties with a machine learning inspired motion planing framework.

D. Planning Space Decomposition

In this work we implement an algorithm using machine learning concepts similar to the following discussed research.

Feature Sensitive Motion Planning [32] uses machine learning to help partition and characterize planning problems. Here, the planning space is subdivided in a recursive manner, then each region is classified and assigned an appropriate planning method. One main strength of this approach is its ability to map workspace/C-Space topologies for a particular planner. However, it is not able to adapt sampling methods to needed regions over time.

HybridPRM [22] employs a reinforcement learning approach to select a sampling strategy with a greater chance for success. However, these samplers are applied globally over the entire problem, and the features of the planning space, such as topology, are not used when deciding where to apply the selected method.

The Unsupervised Adaptive Strategy (UAS) [40] is similar to feature sensitive motion planning because it identifies regions and specifies a planner to the region. UAS also considers the topology of the space. In UAS, the K-means clustering method is used to partition the space using a training roadmap and then HybridPRM [22] is applied in each region. This method showed an improvement in speed and quality in the roadmaps generated, but does not consider all aspects of the planning process in particular, the edge creation process.

Utility Guided Sampling [10], [11] uses information from previous experiences to guide sampling to more relevant areas of the C-Space. Every exploration of the C-Space provides information to the motion planner. They construct an approximate model of the C-Space. Their model captures and maintains information from each configuration to predict the state of unobserved configurations and reduce collision detection calls.

RESAMPL [37] uses local region information (e.g., entropy of neighboring samples) to make decisions about both how and where to sample, and which samples to connect together. This use of spatial information about the planning space enables RESAMPL to increase sampling in regions identified as narrow and decreases sampling in regions identified as free. These approaches do not consider the topology that is discovered within the explored space.

Task and Motion planning (TAMP) [12] integrates logical search over high level actions by observing and making decisions based on the motion and task being currently performed. It uses the markov decision process to learn the appropriate inverse kinematics for robots. TAMP however considers the entire dynamics of the environment and the finite state of the machine as an input to the reward parameter. This method however does not give an assurance for the best method being used consistently in an environment.

The methods cited aim to subdivide and simplify the planning space using different machine learning techniques. We utilize these ideas and include a machine learning paradigm in our algorithm.

III. Heterogeneous Property of the C-space

This section discuss the properties of different regions in the C-Space i.e., free space region, obstacle region, narrow passage, cluttered region and the heterogeneity of the C-Space as integrated in our algorithm.

Free Space Region: Two configuration points in a region are in C-free, if the visibility of points allow them to connect to each other forming connected components in the region, i.e. region is $(\varepsilon, \alpha, \beta)$-expansive, and the trajectories connecting these points lie in the same homotopy class.

Obstacle Regions: With no visibility and expansiveness in the region, the configuration points fails to connect each other reporting existence of no trajectories connecting the points in homotopy class. This results in a blocked (C-Obstacle) region i.e. region is not $(\varepsilon, \alpha, \beta)$-expansive.

Narrow Passage Regions: Two configuration points in region, where region is $(\varepsilon, \alpha, \beta)$-expansive, are in a narrow passage if there are at least some trajectories in the homotopy class connecting the points with low visibility. The low visibility of points infer partial expansiveness of configuration points in F.

Cluttered Region: For any set of configuration points, we say the points are in a cluttered region if the position of multiple C-Obstacle interrupts the existence of trajectories in the same or different homotopy classes connecting the configuration points with varying visibility and expansiveness of region (F) in C-Space. Region F in Figure 2 defines cluttered region, where F1 represents blocked region, F2 represents narrow passage, F3 represents C-free and F4 represents an environment implicitly comprising the properties of homotopy classes and expansiveness. The varying expansiveness of F within each sub-region depends on the position of the C-Obstacle in F as the points connecting each other may fail or succeed in doing that.

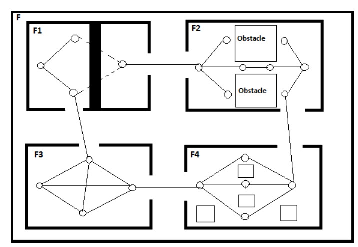


Fig. 2: **Cluttered Region**

IV. Heterogeneous Planning Space Algorithm

To achieve the division of the C-Space into homogeneous regions based on defined properties, we developed HPS (Heterogeneous Planning Space) algorithm that can recognize the regions as C-free, blocked (C-Obstacle) and narrow passage, and calculates the reward for the sampler to be applied in the region using reinforcement learning approach.

A. *Identifying Heterogeneous Spaces*

The HPS algorithm as seen in Algorithm 1 identifies each region of the heterogeneous space using the concept of visibility, homotopy classes and expansiveness i.e. the robot placed in an environment, identifies its current position as either free, narrow or blocked based on identified properties of the space.

Algorithm 1 Heterogeneous Planning Space (HPS)

Input. Let S and G be start and goal configuration points where S$\neq$G, visibility parameter ε, connected component parameter β, Homotopy class H, region parameters cfree (C-free), cnarr (narrow passage) and blocked (C-Obstacle).

Initialize: cfree$\leftarrow$False, cnarr$\leftarrow$False, blocked$\leftarrow$False
while $\beta \geq 1$ and H(S,G) exists **do**
  if ε = 1 **then**
    $cfree \leftarrow true$
    CalculateReward(ε)
  else if $0 < \varepsilon < 1$ **then**
    $cnarr \leftarrow true$
    CalculateReward(ε)
  else
    $blocked \leftarrow true$
  end if
end while

Algorithm 2 CalculateReward(ε)

Input. Visibility parameter ε, learning rate parameter ρ. S denotes the set of motion planning samplers and matrix M stores the time for each sampler in a region.

Output. Map N containing region, sampler and reward.

$\rho \leftarrow 0.1$, n = size(S), $i \leftarrow 0$
declare M[region][sampler]
while $i \leq n-1$ **do**
  $s_i[t] \leftarrow GetExecutionTime(s_i)$
  $M[\varepsilon][s_i] = s_i[t]$
  $i \leftarrow i+1$
end while
$r \leftarrow GetRegion(\varepsilon)$
if M not NULL **then**
  $k \leftarrow min(M[\varepsilon][s_i])$
  $\gamma \leftarrow e^{(\rho*\varepsilon*k)}$
  $N.push(r, s, \gamma)$
end if
return N

The algorithm here takes the calculated values of ε-goodness for each configuration point and β-LOOKOUT for connected components in the environment, i.e. $\forall$ point q $\in$ C-Space, it sees at least ε-fraction of C-Space.

$$\mu(V(q) \geq \varepsilon\mu(C - Space) \tag{1}$$

The β-LOOKOUT for the set of connected component C is

$$\mu(\beta - LOOKOUT(V(C))) \geq \alpha\mu(V(C)) \tag{2}$$

such that each point in C sees β-fraction of the complement of C, where *V(C)* refers to the visibility for set C. It investigates the existence of connected components in the environment and then identifies the region based on the value of ε for each sample point while there exists path from the start to goal configuration in the homotopy class, as shown in Algorithm 1. It identifies three regions i.e., C-free, narrow passage and blocked (C-Obstacle) region. In C-free region, a configuration point could successfully connect to all nearby points easily, i.e. ε = 1. In the narrow passage region, a configuration point can connect to only some point due to low visibility as expansion will either fail or occur in small increments in the region. Otherwise a region is said to be a blocked region if points fails to connect its neighbors. The robot's internal state continues to update information about the visited regions and the similarities to the new regions it encounters.

B. Machine Learning and Heterogeneous Spaces

Algorithm 2 uses a reinforcement learning approach to calculate rewards for all sampling methods being used based on the type of region it encounters and it subsequent performance (success rate).

From the available set of sampling methods, Algorithm 2 calculates the execution time for randomly selected samplers in C-free, narrow passage or blocked regions. Based on the performance of samplers in a region, the reward is computed for the sampler with minimum execution time, i.e.

$$\gamma = e^{(\rho*\varepsilon*k)} \tag{3}$$

The reward is an exponential factor of the type of region and execution time of the sampler, where ε refers to the type of region, k is the minimum time taken by a sampler in a region and ρ is constant learning rate. The computed reward (3) is stored in a nested map N for the respective region and sampler. A suitable sampler for an identified region is chosen based on the learning reward scheme and information stored in N. The reward scheme tracks the success and failure of the different planning methods being applied, which helps in building a self-learning capability for the robot in applying the best method in any environment scenario.

V. Experimental Results

We discuss our experimental setup, results and comparisons with HybridPRM and UtilityGuidedSampling.

A. Experimental Setup

We perform experiments in seven different environments. These environments consists of four homogeneous regions i.e. free space, narrow passage, blocked region, cluttered region and three environment with combination of two or more homogeneous regions.

- **Empty room**: The environment has a room with no obstacles in it as shown in Figure 3a.
- **Cluttered Environment**: Obstacles are cluttered around the room as shown in Figure 3c. The robot has to traverse these obstacles successfully to reach its goal.
- **Blocked environment**: The environment has a solid block covering the space diagonally in the room as shown in Figure 3b.
- **Zig-Zag tunnel**: This environment consists of a narrow tunnel that the robot has to traverse in order to move from one side of the room to the other as shown in Figure 3d.
- **6 DOF Heterogeneous Environment**: This environment comprises of a free space, narrow passage, cluttered region and blocked region within it as shown in

Figure 4a. The robot is a 6 DOF articulated linkage that has to pass through a hole in the wall and encounter obstacles at the other end of the wall.

- **Kukayoubot** : An 8 DOF robot in an environment with four different rooms, see Figure 4b. Its base has 5 DOFs that allow it to move forward, backward and rotate, and its arm has 3 DOFs. 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).
- **Maze3D** : A cylindrical robot in a tunnel-like environment structured as maze searches for the best path to traverse to reach goal as shown in Figure 4c.

The environments are taken from the Parasol Lab benchmarks at Texas A & M University [1]. We use the brute force K-closest neighbor finding technique [30], the euclidean distance metric and a straight line local planner. The experiment aims to solve a query (start to goal) for all environments except the blocked region where we calculate the time taken to generate 100 nodes in its free regions. The environments were designed to simulate and study the behavior of HPS in the different regions which make up our definition about heterogeneous spaces. Experimental results for each environment are averaged over 10 runs with different random seeds and run on a Dell Optiplex 7040 desktop machine. Sample nodes and generated paths for 6 DOF Heterogeneous, KukaYouBot and Maze3D environments are shown in Figure 4.

B. Querying the Environment

The performance of the different strategies i.e., HPS, HybridPRM and UtilityGuidedSampling are compared based on the number of nodes generated (see Figure 5a), number of connected components formed (see Figure 5b) and the time taken to solve a query as shown in Table I. We generate plots to show the learning trend for HPS based on the reward probabilities generated for each sampler used.

In the **empty room** (see Figure 3a), UtilityGuidedSampling strategy performs better than HybridPRM and HPS strategies in terms of time as seen in Table I. This is a simple room with no obstacles that requires no learning and so the learning methods incur an unnecessary overhead which is reflected in the time and number of nodes generated. We do not produce learning plots for this environment since no learning occurred. The aim of this experiment was to show the trend in performance of the methods being compared across different representative regions of a heterogeneous environment. UtilityGuidedSampling strategy did not finish for other environments apart for the empty room and the cluttered environment and our results in Table I indicates DNF (did not finish) for the other experiments.

In the **blocked environment**, we perform an experiment to determine how HPS performs in an environment where it is not possible to solve a query due to a obstacle completely obstructing any trajectory from the start to a goal position. HPS and HybridPRM returns similar time needed as seen in Table I and the number of connected components as seen in Figure 5b while UtilityGuidedSampling was not successful in generating any feasible node. The learning plot in Figure 6a shows a trend with higher probabilities being recorded for the BasicPRM and Gauss Sampling method in HPS. OBPRM and Bridge Sampling method depreciates in performance due to the constant failures that would occur in this environment. There is a zero chance of connecting nodes on the opposite side of the obstacle and so methods that tend to take into cognizance the obstacles in the environment will fail.

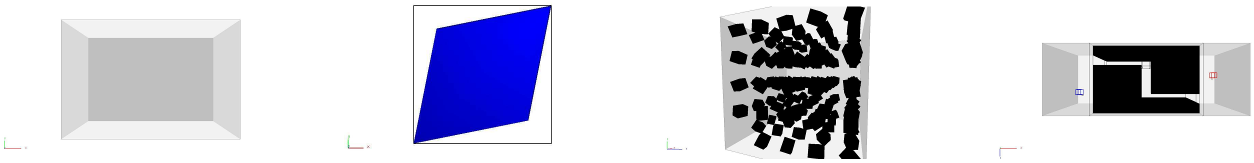
In the **cluttered environment**, we have similar performance between HPS and HybridPRM in terms of time needed to solve the query. HPS however produces less connected components as seen in Figure 5b which is an improvement over HybridPRM even in this simple environment. In the learning plot in Figure 6b, HPS utilizes more of BasicPRM and OBPRM sampling strategies which according to literature perform well in free and obstacle regions respectively and HPS is able to capture this.

In the **zig-zag tunnel environment**, here we see a more complicated environment with an a narrow passage and the aim is to traverse this narrow tunnel to the goal position. HPS shows an improvement in performance in comparison to HybridPRM in terms of time (see Table I) while using similar number of nodes and connected component. This result indicate our algorithm's ability to improve performance as the environment gets more heterogeneous. The learning plot in Figure 7a and 7b shows the learning trend of HPS in the free and narrow region of the zig-zag environment and we see that the Gauss and Bridge has a higher probability than OBPRM and BasicPRM. The strengths of Gauss and Bridge according to literature comes into play in this environment and HPS utilizes this methods during its learning phase.

In the **6 DOF heterogeneous environment**, HPS is not the best performing method in terms of time but this is comparable with HybridPRM with only a 1 second difference. HPS however uses less nodes to solve the query and equal connected components produced. In the learning plot as seen in Figure 8a and 8b, we see that HPS has a higher probability for Gauss and Bridge in the free region and in the cluttered region OBPRM and BasicPRM sampling methods have higher probability. This again is indicative of the properties of these different sampling methods being used. In the narrow region, the learning plot for HPS as seen in Figure 8b shows a higher probability for OBPRM and BasicPRM in most cases.

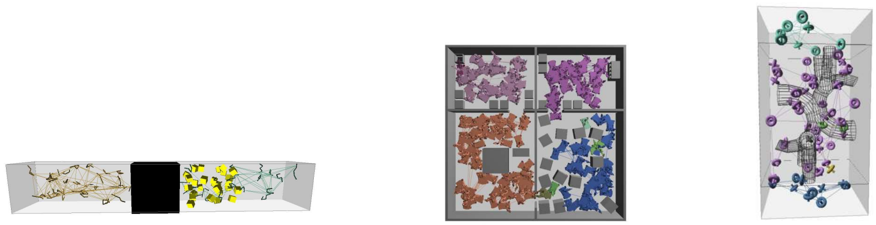
In the **KukaYouBot environment** we see an improved performance of HPS in terms of time needed to solve the query. This environment has been proven to be difficult to plan for as seen in [16] but, HPS cuts down the time needed by almost an 8th the time needed by HybridPRM. HPS also uses less nodes and produced less connected components and this is a vast improvement in comparison to previous methods. In the learning plot, the Bridge Sampling method has a higher probability in most cases with OBPRM coming a close second. This two strategies tend in work well in obstacle ridden environments and HPS utilizes them while also having simpler methods like BasicPRM to use in simple regions of the environment.

In the **Maze3D environment** we see another improvement



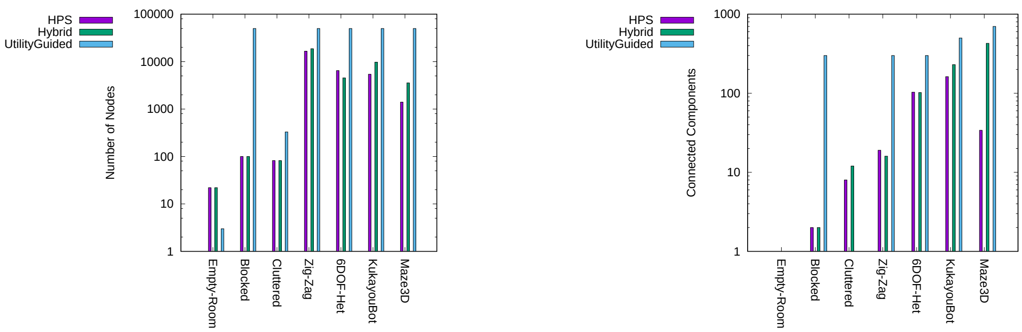
(a) **Empty Room** (b) **Blocked** (c) **Cluttered** (d) **Zig-Zag Tunnel**

Fig. 3: Homogeneous Environments Studied



(a) **6 DOF Heterogeneous** (b) **KukaYouBot** (c) **Maze3D**

Fig. 4: Heterogeneous Environments Studied

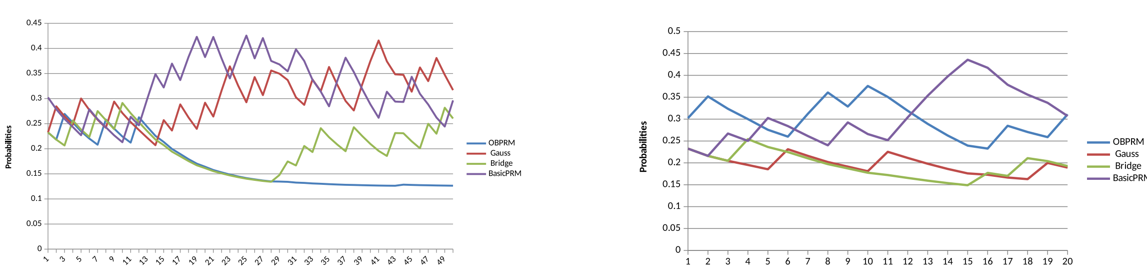


(a) **Number of Nodes** (b) **Number of Connected Component**

Fig. 5: Performance Plots

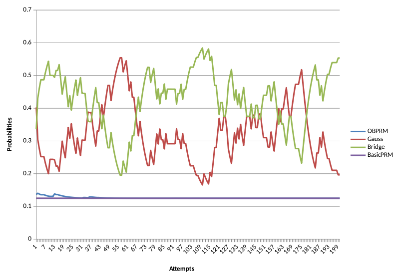
Environment	HPS	HybridPRM	UtilityGuidedSampling
Empty room	0.014	0.010	**0.003**
Blocked environment	0.169	0.168	DNF
Zig-Zag Tunnel	**89.32**	100.02	DNF
Cluttered environment	**0.25**	0.29	12.08
6 DOF Heterogeneous Environment	17.41	**16.90**	DNF
KukaYouBot environment	**1711.30**	9035.62	DNF
Maze3D	**102.91**	135.35	DNF

TABLE I: **Map Generation time for each sampler in a environment**

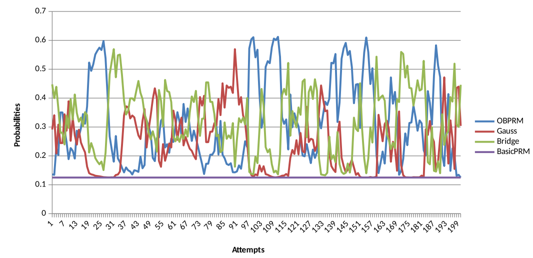


(a) **Blocked Environment** (b) **Cluttered Environment**

Fig. 6: **Homogeneous regions**

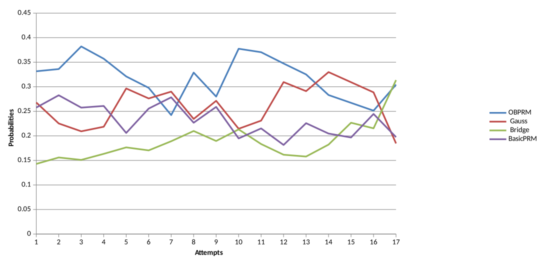


(a) Free region

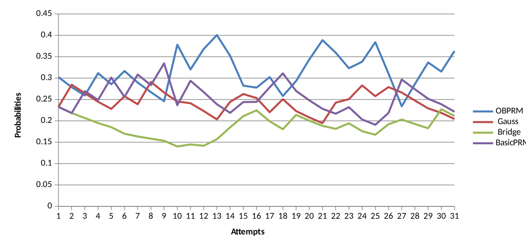


(b) Narrow passage

Fig. 7: **Zig-Zag Tunnel Environment**

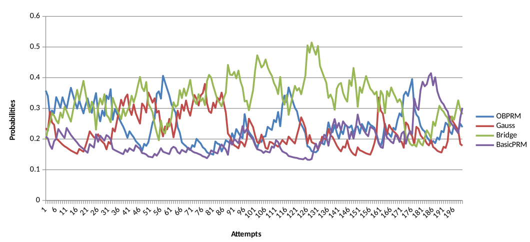


(a) Free region

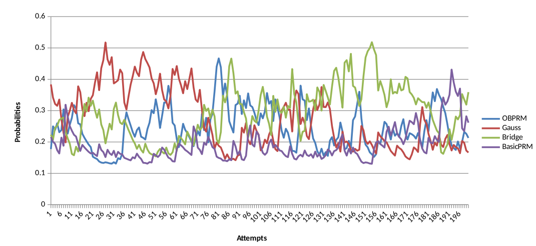


(b) Narrow passage

Fig. 8: **6 DOF Heterogeneous Environment**



(a) Free region



(b) Narrow passage

Fig. 9: **KukaYouBot Environment**

of HPS in terms of time needed to solve the query, fewer nodes needed and fewer number of connected components. The learning plot in Figure 10a and 10b, shows that HPS has a higher probability for BasicPRM in the free region and a mix of Gauss and OBPRM in the free region. HPS once again utilizes the capabilities of Gauss and Bridge samplers in the narrow region.

VI. Conclusion

In this paper we have presented formal definition about the heterogeneity of the planning space from previously defined properties e.g., visibility. Using this definition, we have proposed an efficient methodology that take cares of such heterogeneous environments and intelligently apply suitable sampling methods in the different regions. We have experimentally demonstrated its efficiency, versatility and application in varying environment scenarios. Our method is applicable to any robot dimension and can have application in life-saving and real world application such as search and rescue and agricultural robot monitoring applications.

References

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

[2] 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*, pages 155–168, Natick, MA, USA, 1998. A. K. Peters, Ltd. (WAFR '98).

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

[4] S. Atiya and G. Hager. Real-time vision-based robot localization. *IEEE Trans. Robot. Automat.*, 9(6):785–800, 1991.

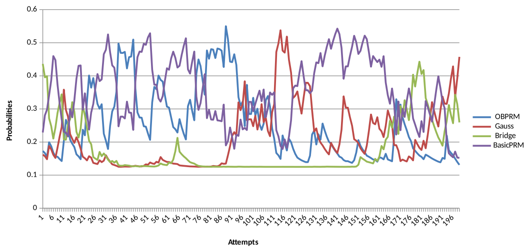
[5] S. Bhattacharya. Search-based path planning with homotopy class constraints, 2010.

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

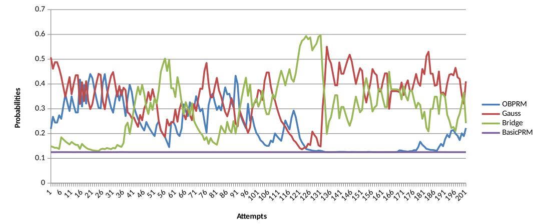
[7] S. Bhattacharya, D. Lipsky, R. Ghrist, and V. Kumar. Invariants for homology classes with application to optimal search and planning problem in robotics. *Annals of Mathematics and Artificial Intelligence*, 67(3-4):251–281, 2013.

[8] R. Bohlin and L. E. Kavraki. Path planning using Lazy PRM. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 521–528, 2000.

[9] V. Boor, M. H. Overmars, and A. F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE*



(a) Free region



(b) Narrow passage

Fig. 10: **Maze3D Environment**

Int. Conf. Robot. Autom. (ICRA), volume 2, pages 1018–1023, May 1999.

[10] B. Burns and O. Brock. Sampling-based motion planning using predictive models. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3313–3318, 2005.

[11] B. Burns and O. Brock. Toward optimal configuration space sampling. In *Proc. Robotics: Sci. Sys. (RSS)*, pages 105–112, 2005.

[12] R. Chitnis, D. Hadfield-Menell, A. Gupta, S. Srivastava, E. Groshev, C. Lin, and P. Abbeel. Guided search for task and motion plans using learned heuristics. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 447–454. IEEE, 2016.

[13] 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*. MIT Press, Cambridge, MA, June 2005.

[14] J. Denny and N. M. Amato. Toggle PRM: simultaneous mapping of c-free and c-obstacle - a study in 2d -. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2011, San Francisco, CA, USA, September 25-30, 2011*, pages 2632–2639, 2011.

[15] J. Denny, M. Morales, S. Rodriguez, and N. M. Amato. Adapting rrt growth for heterogeneous environments. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1772–1778. IEEE, 2013.

[16] 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)*, pages 3227–3234, Hamburg, Germany, October 2015.

[17] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[18] R. Gayle, M. C. Lin, and D. Manocha. Constraint-based motion planning of deformable robots. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1046–1053, April 2005.

[19] 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)*, pages 4420–4426. IEEE, 2003.

[20] D. Hsu, J.-C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *Int. J. Robot. Res.*, 25:627–643, July 2006.

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

[22] D. Hsu, G. Sánchez-Ante, and Z. Sun. Hybrid PRM sampling with a cost-sensitive adaptive strategy. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3874–3880, Barcelona, Spain, 2005. IEEE.

[23] 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.

[24] M. Kallmann, A. Aubel, T. Abaci, and D. Thalmann. Planning collision-free reaching motion for interactive object manipulation and grasping. *Computer Graphics Forum*, 22(3):313–322, Sept. 2003.

[25] 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.*, 12(4):566–580, August 1996.

[28] T. Lozano-Perez. Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, 100(2):108–120, 1983.

[26] S. M. Lavalle. Sampling-based motion planning. 2006.

[27] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 473–479, 1999.

[29] J. D. Marble and K. E. Bekris. Asymptotically near-optimal is good enough for motion planning. In *The International Symposium on Robotics Research (ISRR)*, 2011.

[30] T. McMahon, S. Jacobs, B. Boyd, L. Tapia, and N. M. Amato. Evaluation of the k-closest neighbor selection strategy for prm construction. Technical Report TR12-002, Texas A&M, College Station Tx., 2011.

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

[32] M. Morales, L. Tapia, R. Pearce, S. Rodriguez, and N. M. Amato. A machine learning approach for feature-sensitive motion planning. In *Algorithmic Foundations of Robotics VI*, Springer Tracts in Advanced Robotics, pages 361–376. Springer, Berlin/Heidelberg, 2005. (WAFR ‘04).

[33] T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi. Knot planning from observation. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 3, pages 3887–3892, sept. 2003.

[34] M. H. Overmars and P. Švestka. A probabilistic learning approach to motion planning. In *Algorithmic Foundations of Robotics*. A. K. Peters, Wellesley, MA, 1995.

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

[36] S. Rodriguez, J.-M. Lien, and N. M. Amato. Planning motion in completely deformable environments. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2466–2471, May 2006.

[37] S. Rodriguez, S. Thomas, R. Pearce, and N. M. Amato. (RESAMPL): A region-sensitive adaptive motion planner. In *Algorithmic Foundation of Robotics VII*, Springer Tracts in Advanced Robotics, pages 285–300. Springer, Berlin/Heidelberg, 2008. (WAFR ‘08).

[38] O. Salzman, M. Hemmer, and D. Halperin. On the power of manifold samples in exploring configuration spaces and the dimensionality of narrow passages. *IEEE Transactions on Automation Science and Engineering*, 12(2):529–538, 2015.

[39] T. Siméon, J.-P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics*, 14(6):477–493, 2000.

[40] L. Tapia, S. L. Thomas, B. Boyd, and N. M. Amato. An unsupervised adaptive strategy for constructing probabilistic roadmaps. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009*, pages 4037–4044, 2009.

[41] 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)*, pages 5798–5803, Hong Kong, P. R. China, June 2014.

[42] J. Zacharias. Pedestrian behavior pedestrian behavior and perception in urban walking environments. *Journal of Planning Literature*, 16(1):3–18, 2001.