Metrics for Efficient Environment Exploration in Robot Motion Planning

Regina Rex[1], Diane Uwacu[2], Shawna Thomas[2], Nancy M. Amato[2] *

[1] Mathematics and Computer Science Department,
University of Wisconsin-Superior, Superior, WI, USA,
rrex@uwsuper.edu.

[2] Parasol Lab, Department of Computer Science and Engineering,
Texas A&M University, College Station, TX, USA,
{duwacu, sthomas, amato}@tamu.edu.

Abstract. Although current state-of-the-art planners utilize properties of the environment such as the workspace topology to guide exploration, more efficient methods and metrics that bias exploration based on desired properties of the robot and its workspace are needed.
In response to this, we present a method that biases the exploration of a Rapidly-exploring Random Tree (RRT) based on the clearance-the free space between the obstacle boundaries of the workspace. The exploration can be biased towards regions with maximum or minimum clearance value. In particular, this method is applied to Dynamic Region-biased RRT (DR-RRT), a sampling-based planner that uses the environment's topology to guide the growth of an RRT. We show our approach generates safer paths in less time compared to the regular DR-RRT.

1 Introduction

Motion planning refers to the classical problem of finding a collision-free path for a robot given a starting point and a goal destination in an environment containing obstacles. Outside the universal robotics application, motion planning is used to find the minimal invasive path for surgical operations, and also in computational biology to study the folding pathways and motions of proteins, to name a few. Except for motion planning problem with robots that have few degrees of freedom (DoFs), it is computationally hard to find a collision-free path [1].

According to J. Denny et al., state-of-the-art motion planners rely on randomized sampling to construct an approximate model of the problem space that is then searched for a valid path [2]. Motion planning for robotics and computational biology problems could be improved by using properties specific to the workspace and robot(s) during planning.

* This work was performed at the Parasol Lab during Summer 2018 and supported in part by the CRA-W Distributed Research Experience for Undergraduates (DREU).

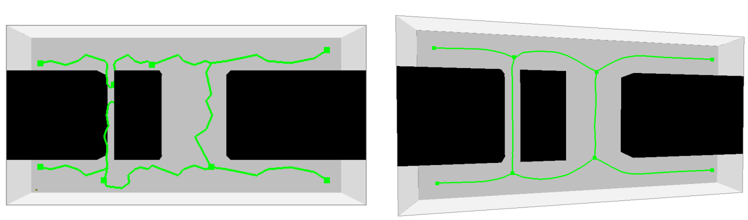
Many sampling-based planners exploit workspace properties like guiding sampling with collision information [3, 4], breaking planning into sub-problems for better planning [5,6], biasing sampling using obstacle geometry [7,8], and making a representation of the workspace topology to direct the planner to unexplored areas [2]. However, it would be desirable to design more methods and metrics to target exploration of the workspace using properties specific to the problem. Take the case of predicting accessibility of protein binding sites. It would be convenient to guide the exploration of the protein sites based on the energy threshold level. A more general example would be in robotics, for safety purposes it would be desirable to explore passages in the workspace with wider clearance value. Clearance value refers to the value of free space between obstacles of the workspace. Taking advantage of robot and workspace properties would also be efficient in narrow passage problems. In the case of narrow passages-parts of the environment where the probability of sampling a valid object position is low [2], biasing environment exploration towards regions with minimum clearance values would be valuable in exploring these passages.

In our work, we exploit the workspace property, clearance value, to guide exploration. In particular, our method is applied to Dynamic Region-biased Rapidly-exploring Random Tree (DR-RRT), a sampling-based planner that encodes the environment's topology to guide RRT growth in the workspace. Using workspace topology guidance is beneficial in problems that have a strong correlation with the workspace geometry [2], as the issues explained above.

We designed a method that biases environment exploration based on the value of the free space between obstacles of the workspace. Our experiments show that our algorithm explores the workspace based on the clearance value of its passages. Hence we can generate safer paths and optimize the cost and runtime of the planning process.

2 Related Work

In this section, we discuss related work to our method and explain the terms and concepts used in the paper.



(a) Reeb Graph

(b) MCS Workspace Skeleton

Fig. 1: Workspace Skeletons

2.1 Workspace Skeleton

A Workspace Skeleton is an undirected graph of the free workspace that represents the environment's topology. Since the Workspace Skeleton satisfies certain topological properties [2], workspace properties like clearance and connectivity can be computed from it.

For our method, we used two types of Workspace Skeleton, the Reeb Graph [9], and the Mean Curvature Skeleton (MCS) [10]. The Reeb Graph is a graph where the vertices are represented by the Morse function and its edges are topologically transitionally to one another [9] (see Figure 1(a)). While the Mean Curvature Skeleton is constructed from a mesh-based algorithm which computes the skeletal representation from the mean curvature flow of some surfaces in the workspace [11] (see Figure 1(b)).

2.2 Dynamic Region-biased Rapidly-exploring Random Trees

Rapidly-exploring Random Tree (RRT) [12] is a single-query planner which at each extension attempt, randomly samples a configuration, q_{rand} and the nearest configuration in the tree, q_{near} is found, If there are no obstacles between them, q_{rand} and q_{near} are then connected.

DR-RRT is a sampling-based planner that uses the Workspace Skeleton for guidance to choose an RRT expansion direction. It uses the Workspace Skeleton to guide RRT growth based on the probability of an RRT exploration to reach the next region in the Workspace Skeleton [2]. Compared to RRT, DR-RRT produces more productive samples and explores more regions in the Workspace Skeleton than regular RRT. DR-RRT avoids false passages and is more robust in complex environments with multiple homotopy classes [2].

Although DR-RRT exploits the Workspace topology, there are instances where it does not explore the workspace as efficiently as one would expect. For example, it would be efficient for DR-RRT to first explore the wider passages in a workspace for a feasible path before exploring narrow passages. But DR-RRT explores the workspace based on the probability of sampling valid configurations in each region of the Query Skeleton [2]. In this case, DR-RRT spends time exploring both wide and narrow passages, even though exploring the wider passages first is more efficient because they would be fewer collision checks.

3 Clearance-Biased Exploration

In this section, we explain clearance value biased exploration.

3.1 Algorithm Overview

Our clearance-biased exploration method is depicted in Algorithm 1 and Figure 2. The main idea is to compute a Property Map for the Workplace Skeleton. A Property Map is an unordered map that contains the clearance value for every node and edge in the Workspace Skeleton.

Algorithm 1 Clearance-Biased Exploration

Input: *env*: the environment
Output: *g*: the free-space roadmap
$WS \leftarrow$ `BuildWorkspaceSkeleton`(env)
$QS \leftarrow$ `GetQuerySkeleton`(WS)
$PM \leftarrow$ `GeneratePropertyMap`(QS)
$curRegion \leftarrow$ `CreateDynamicSamplingRegion`(PM_0)
while $!done$ **do**
  $g \leftarrow$ `RRT`$(curRegion)$
  $children \leftarrow$ `curRegion.GetChildren`$()$
  $curRegion \leftarrow$ `maxvar`(`children.Clearance`$()$)
return g

Using DR-RRT, a Workspace Skeleton is created for the environment. The Workspace Skeleton is then pruned only to contain edges that connects the start configuration to the goal configuration. The Workspace skeleton is pruned to avoid exploring regions that do not solve the query. The pruned skeleton is called a Query Skeleton [2] (Figure 2(b)). After pruning, a Property Map is created.

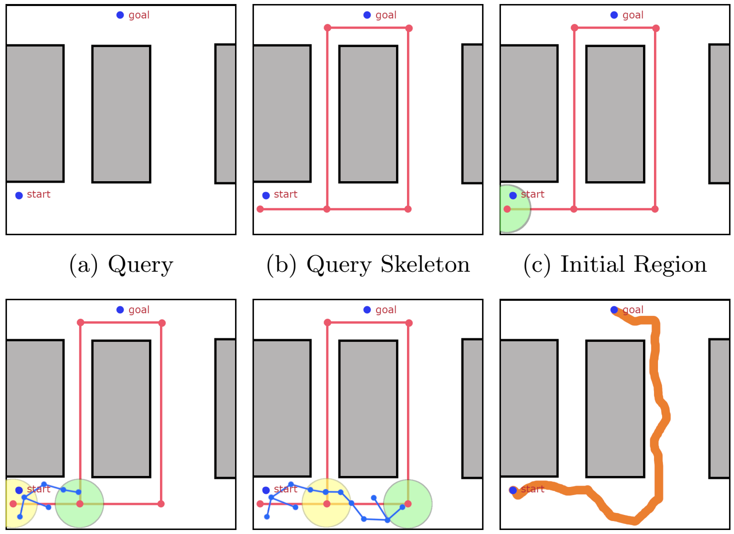
From the Query skeleton, DR-RRT creates a dynamic sampling region at the node closest to the start point of the query (Figure 2(c)). At the initial region, it begins to grow an RRT (Figure 2(d)). On each iteration, a region with the maximum clearance value is chosen as the next growth target (Figure 2(e)). In a workspace, clearance-biased DR-RRT first explores regions with higher clearance value (Figure 2(f)). Exploration can also be targeted to regions with smaller clearance values (narrow passages).

Method	Workspace Skeleton	Runtime	Number of Collision Checks	Average Path Clearance
Clearance bias DR-RRT	Reeb Graph	0.4164 sec	43105	9.14524
	MCS	4.1565 sec	45373	8.15253
Regular DR-RRT	Reeb Graph	0.5259 sec	73219	9.08872
	MCS	4.1518 sec	41827	6.09615

Table 1: Experiment Results

4 Experiments

Our goal for these experiments is to demonstrate how clearance-biased DR-RRT utilizes the information from the Workspace Skeleton during sampling to guide RRT growth. We ran our tests in the 3D environment shown in Figure 3. We measure test success by the exploration of the RRT towards regions with the specified clearance value. For comparison, we check the number of collision



(a) Query (b) Query Skeleton (c) Initial Region

(d) RRT growth (e) Biased growth (f) Path

Fig. 2: Example of clearance-biased DR-RRT: (a) environment and query q_s, q_g (b) Query Skeleton (c) Dynamic Sampling Region created at Initial region near q_s(d) RRT growth (e) exploration biased towards region with larger clearance value (f) Collision-free path from q_s, q_g

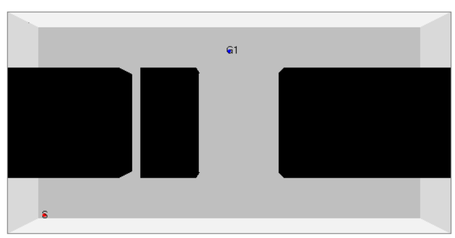
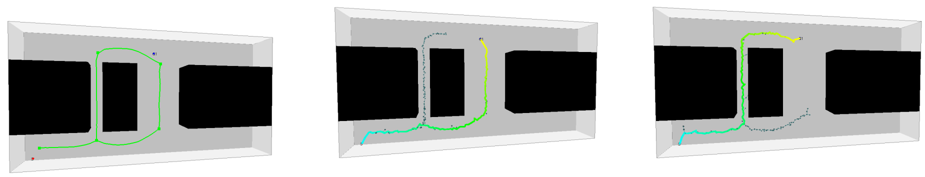


Fig. 3: 3D Test Environment With q_s and q_g

detection calls, which indicates the total number of collision check it takes for the query to be solved, the total time taken to solve the query, and the clearance value of the collision-free path.

4.1 Experimental setup

We demonstrate our method by running tests on the 3D environment in Figure 3. We compared regular DR-RRT and the clearance-biased DR-RRT on the workspace with a Reeb Graph Query Skeleton (Figure 4(a)) and a Mean Curvature Query Skeleton (Figure 5(a)).

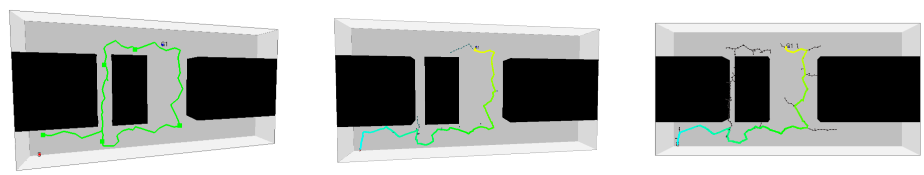


(a) MCS Query Skeleton

(b) Path - Clearance-biased DR-RRT

(c) Path - Regular DR-RRT

Fig. 4: Test Result on 3D Environment using MCS



(a) Reeb Graph Query Skeleton

(b) Path - Clearance-biased DR-RRT

(c) Path - Regular DR-RRT

Fig. 5: Test Result on 3D Environment using Reeb Graph

We ran our tests on a Unix system, and we use VIZMO++, a motion planning visualization tool, to debug and visualize all the ran experiments. We also implemented all our methods in Parasol Motion Planning Library, (PMPL) a C++ motion planning library developed in the Parasol Lab.

4.2 Results

From Table 1, our method shows faster planning time and less number of collision detection checks than the regular DR-RRT. The average clearance value from the path gotten from the clearance-biased DR-RRT is greater than that of regular DR-RRT because exploration is biased towards regions with higher clearance value. Thus the paths found with clearance-biased DR-RRT is safer. Also, it can be noted that the clearance-biased method is robust with the underlying skeleton as the difference between the number of collision detection calls is not as vast as the difference in regular DR-RRT.

5 Conclusion

We introduced a new method for biasing DR-RRT based on the Workspace Skeleton clearance value. Our contribution is the use of the clearance value property from the Workspace Skeleton to bias how regions of DR-RRT are explored. Our experiment result shows an improvement in overall planning time and less number of collision checks as compared to regular DR-RRT.

Some future directions for this work would include designing more metrics for biasing exploration based on other properties like energy threshold levels. More efficient methods and metrics can be developed to guide exploration for better exploitation of the Workspace Skeleton.

6 Acknowledgments

I want to express my gratitude to Dr. Nancy Amato, my faculty mentor, and Diane Uwacu, my graduate student mentor, for their support and guidance throughout this project. This research is supported in part by the Computing Research Association-Women (CRA-W) Distributed Research Experience for Undergraduate (DREU) and by the Texas A&M Department of Computer Science and Engineering.

References

1. J. C. Latombe, "Motion planning: A journey of robots, molecules, digital actors, and other artifacts," *Int. Journal of Robotics Research*, vol. 18, no. 11, pp. 1119–1128, 1999.
2. J. Denny, R. Sandstrom, A. Bregger, and N. M. Amato, "Dynamic region-biased exploring random trees," in *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, (San Francisco, CA), December 2016.
3. P. Cheng and S. LaValle, "Reducing metric sensitivity in randomized trajectory design," in *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, vol. 1, pp. 43–48 vol.1, 2001.
4. A. Yershova, L. Jaillet, T. Simeon, and S. M. Lavalle, "Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 3856–3861, April 2005.
5. S. Rodriguez, X. Tang, J.-M. Lien, and N. M. Amato, "An obstacle-based rapidly-exploring random tree," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2006.
6. L. Zhang and D. Manocha, "An efficient retraction-based RRT planner," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2008.
7. H. Kurniawati and D. Hsu, "Workspace-based connectivity oracle - an adaptive sampling strategy for prm planning," in *Algorithmic Foundation of Robotics VII*, pp. 35–51, Berlin/Heidelberg: Springer, 2008. Book contains the proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR), New York City, 2006.
8. 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, pp. 361–376, Berlin/Heidelberg: Springer, 2005. (WAFR '04).
9. G. Reeb, "Sur les points singuliers dune forme de pfaff compl'ement integrable ou dune fonction numerique," in *Comptes Rendus de LAcademie ses Seances, Paris, 222*, pp. 847–849, 1946.
10. D. Uwacu, E. Yang, S. Thomas, and N. M. Amato, "Using motion planning to evaluate protein binding site accessibility," tech. rep., Department of Computer Science, Texas A&M University, July 2018.

11. A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang, "Mean curvature skeletons," *Eurographics Symposium on Geometry Processing*, p. 27(1), 2012.
12. S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 473–479, 1999.