Workspace Skeleton Tools for Motion Planning

Kiffany Lyons, Diego Ruvalcaba, Mukulika Ghosh, Shawna Thomas, and Nancy Amato

***Abstract*— Motion planning is the ability to find a valid path from a start to a goal through an environment containing obstacles. Its applications reach into many different fields which include robotics, minimally invasive surgical planning, graphic animation, and even molecular design. In motion planning, the workspace is the physical environment where the robots and the obstacles reside. Complexity of a workspace (e.g. complex geometry) might affect the performance of motion planning algorithms. We wanted to create an intuitive and sparse representation of the workspace in the form of a workspace skeleton that can be used to improve motion planning algorithms. In particular, we developed a workspace skeleton that encodes important topographical information regarding the workspace and properties of the workspace that could be used to guide the planner and increase effectiveness. We also developed a visual representation of the skeleton such that users can view, manipulate, and debug the skeleton. We demonstrate the usability and the application of our skeleton tools in a pre-existing motion planning algorithm that utilizes a workspace skeleton.**

I. INTRODUCTION

Motion planning is a problem to find a valid path for an object through an environment containing obstacles [4]. It can be used for robots performing surgery or simulations to try and find the most efficient and least invasive path during a procedure. This could obviously help patients with recovery time and make surgeries much more efficient. A few other applications of motion planning include molecular design, assembly sequencing, and animation.

In our research we focus on the workspace, which is the physical environment where the robots and obstacles reside. Workspace complexity is a serious problem in motion planning because as the complexity (e.g., complex geometry) of the workspace increases, like in Fig. 1, the performance of the motion planner can be negatively affected. Besides, we can use the inherent workspace information such as passage width to guide the motion planning algorithms.

We developed tools for a user intuitive and sparse representation of the workspace, known as workspace skeleton. A workspace skeleton is a graph data structure that overlays a workspace. The tools allows easy manipulation of the workspace skeleton and also storage of properties that can be associated with each edge and vertex in the skeleton.

We added tools that the user can use to view, manipulate, and debug the skeleton. The user can modify existing skeletons many different ways or even add their own. These edits allow the user to provide feedback on the skeleton which eventually helps in finding better paths and plans than the automatic planner.

This research supported in part by NSF awards CNS-0551685, CCF 0702765, CCF-0833199, CCF-1439145, CCF-1423111, CCF-0830753, IIS-0916053, IIS-0917266, EFRI-1240483, RI-1217991, by NIH NCI R25 CA090301-11,and by DOE awards DE-AC02-06CH11357, DE-NA0002376, B575363.

Fig. 1. Complex Environment

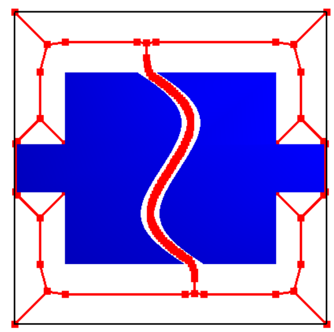

Fig. 2. Workspace Skeleton

We also provided a way to associate attributes of the workspace to the skeleton using property maps. Information regarding the workspace such as passage width can be stored as property associated to a corresponding edge or vertex in the skeleton. The property maps allows creation of different filtered views of the same skeleton based on various requirement (For example, view of only those edges in the skeleton that passes through regions which are wide enough for the robot). There are many different metrics that can be stored as properties. There is no constraint on specification properties of the workspace property map. We hope this allows future applications to be uninhibited in planning for multitudes of variables utilized in the planning process.

The skeleton with the proposed tool can be used to guide motion planning algorithms through more desirable pathways (such as those with high clearance or provided by user). We can also use the same workspace skeleton across different robots to plan in the same environment since the information of the workspace does not change. Finally, since the skeleton can store properties of the workspace in its edges and vertices, we can use planners to create a different view of the skeleton to not plan on regions that have undesirable properties such as too long or not wide enough for the robot to pass through like in Fig. 3.

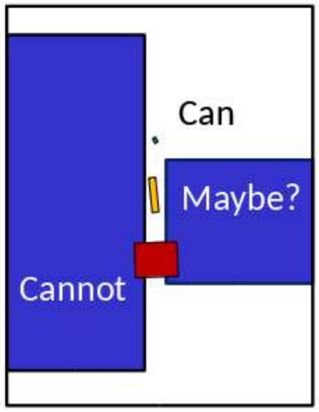

Fig. 3. Use for Property Map

In our experiments, we use the skeleton tools to edit the skeleton used in an existing motion planning strategy and compared its performance against the one with the un-edited skeleton. By using the information from the workspace we intend to increase the effectiveness of the planner by preventing the it from trying to plan through regions that are inaccessible to the robot or undesirable to the user. We used two different approaches to edit the skeleton: manual and automatic skeleton editing.

1) User-based Skeleton Editing : The user can update an existing skeleton to guide the planner through specific regions and paths.
2) Property-based Skeleton Editing : The property map can generate a view of the skeleton that can guide the planner through accessible regions and paths based on the system requirements of the robot.

To summarize our contributions we developed the following:

1) Created visual tools to view, manipulate and debug the workspace skeleton
2) Allowed the skeleton to encode important information regarding the workspace through property maps.

The rest of the paper will talk about related work regarding our research, our approach to the workspace complexity problem, and our results from our contribution.

II. Related Work

In this section, we review existing methods and explain concepts and terms that we will use in the rest of the paper such as Motion Planning, Medial Axis Planning, Dynamic Region-biased RRT (DRRT), User-Guided Motion Planning, Workspace Informed Sampling (WIS), and Visualization Tools.

A. Preliminaries

The workspace is the physical environment wherein the robots and obstacles reside. The difference between the workspace and the configuration space is that there are many constraints such as dimension which inhibit the workspace and make the configuration space necessary in motion planning. The configuration space is an abstract representation of an environment in which the robot is represented as a single point with multiple coordinates that make up a configuration which describes its position. what is workspace. In motion planning the environment must have as many degrees of freedom as the robot. If the type of robot is known it is easy to discern how many variables are needed to describe a configuration. Although planning is done in c-space, the workspace can play a role in the complexity of the problem.

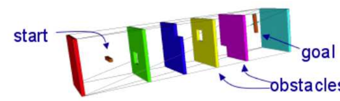

Fig. 4. Motion Planning

B. Workspace Based Motion Planning

Medial Axis planning strategically tries to find the path that gives the agent as much clearance as possible. The medial axis is a type of skeleton that represents the parts of the free space that are equidistant to two or more obstacles or boundaries [5]. The way it works is, a random sample is placed into the environment, if the sample is valid or not colliding with an obstacle, it finds its nearest obstacle or boundary and moves the be so that it is centered. In medial axis planning nodes are generated near the medial axis to increase the probability of finding a path even more likely of finding a path with increased robot clearance than other planners, thanks to the medial axis skeleton.

Dynamic Region-biased Rapidly-expanding Random Tree Planning also known as DRRT, is one of the few motion planning algorithms that actually utilizes some form of skeleton. Regular RRT is just a graph that starts at the start configuration and generates a random sample that is in the direction of the goal, as long as that sample is valid and can be connected it is added to the graph. RRT will continue to do this until it reaches the end configuration and the problem is solved [1]. DRRT and RRT are very similar only instead of generating random samples pointing toward the goal, DRRT creates regions near points on a skeleton and generates a random sample within the region. DRRT samples continuously follow along the skeleton towards the goal until the tree has extended itself enough to reach the end configuration. With DRRT, the skeleton allows the tree to be constantly kept near the skeleton as it moves towards the goal.

Workspace informed sampling is very useful in motion planning because it conserves resources by utilizing information about the workspace to sample more efficiently. Workspace informed sampling expedites the path finding process by creating ideal situations wherein paths are deleted without the necessity of testing.

User-Guided planning consists of an active interaction between a user and an algorithm. The user analyzes the workspace to determine a solution while they allow the planner to take care of the high-precision computations [2], [3]. In our project, we allow the user to modify the skeleton with out visualization tools. Due to our use of the DRRT planner, it allows the user to guide the planning based on the modifications made to the skeleton by the user. The user can insure that the planner avoids areas by deleting members of the skeleton and it can encourage sampling by adding more members to the skeleton.

Vizmo++ is a visualization tool which allows users to view, manipulate, and store environments, paths, queries, and roadmaps. Once you run a motion planning sequence in Parasol Motion Planning Library(PMPL), you can use vizmo to view the procedure and debug if necessary. Vizmo allows users to generate road maps and animate the sampling process which help users understand and debug motion planning strategies.

III. Methodology

Our research goals were to provide a way to store various important topographical information about the workspace and utilize them during the motion planning process. By developing the property map on the skeleton we achieved this goal. We enable users to load, create, and manipulate the skeleton graph overlay which associates properties to the skeleton's vertices and edges with information regarding the workspace. The implementation of our property map allows easy storage of information regarding the workspace. In addition, we also set out to create visual tools for our workspace skeleton so that users are able to visualize and manipulate the skeleton as well as inspect and analyze the skeleton being used in the motion planning process. Users are able to import and export a skeleton as well as select components of the graph for manipulation. Manipulation tools include coloring, adding and deleting of edges and vertices, edge merging, and edge collapsing. We achieved this by the inherent features of the graph structure of the skeleton. We performed basic graph manipulations the same way any other graph would do it, only our nodes and edges had properties that needed to be constant through revisions of the skeleton.

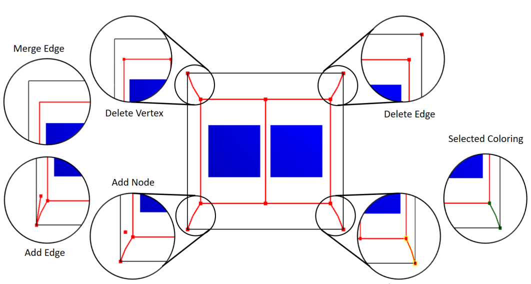

Fig. 5. Workspace Skeleton Tools

Our long term objective for this project is to allow storage and retrieval of multiple information associated with the same workspace such as passage width and length. For our visual tools, we wanted to allow the user to analyze, update, or manipulate these data structures visually to better their use and applications in motion planners.

IV. Experimental Results

We demonstrate the use of our skeleton tools in two different approaches. For our first experiment, we edit the skeleton using our visual tools to guide the planner along user-defined/user-allowed paths or regions. For our second experiment, we allowed the automatically pruned skeleton based on the clearance requirement of the robot by utilizing the property tools.

Our goal for these experiments is to test, demonstrate and utilize the workspace skeleton tools in improving planning using DRRT algorithm (a workspace skeleton based motion planning algorithm).

We measured the success of each test based on its efficiency and its ability to guide the RRT toward the user defined skeleton. Efficiency in motion planning can be determined by time, amount of samples generated, and collision detections. Obviously, if time is decreased in a trial then it is inherently more efficient and that usually has a correlation to the amount of samples generated in the environment. If more samples are needed to find a path to solve the query then it is less efficient. Finally, collision detections is usually the best representation of how effective a planner is because it measures how many checks for collisions it takes for the motion planner to solve the query. We use all three of these metrics to show if our method is more efficient.

All of our visual tools we created were added to a software called Vizmo++, which is a visualization software used to test motion planning strategies as well as help debug and visualize newly developing motion planning strategies. We coded all of the visual tools with C++ and added it to the PMPL library, which is Parasol Lab's Motion Planning library. All of our test were ran on a unix based operating system.

A. *User-based Skeleton Editing*

In this section we demonstrate that we can use our visual tools to guide our planner to only plan where the user would like it to and find the desired solution path faster and more efficiently in large environments than through using an unedited skeleton.

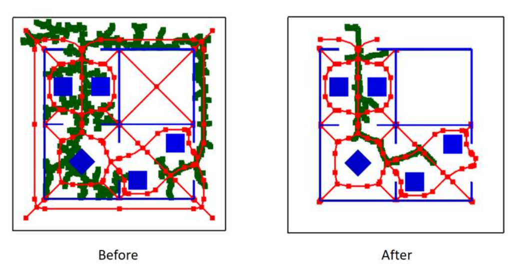

Fig. 6. Experiment 1

In this experiment involving user modification of the skeleton, we simulate a robot's movement in an office environment that had to get from one point to the other. We created the environment that had two obvious paths that could be chosen by the planner, one that went around the office, and one that maneuvered through it. First, we allowed the planner to plan with DRRT using the un-edited skeleton (as shown on the left in Figure 6). Next, we delete some of the skeleton edges outside the office using our visual tools such that solution path goes through the office rooms and not around them (as shown on the right in Figure 6) because the robot needed to accomplish some tasks along that route.

After planning in the same environments using two different skeletons we found our claim to be correct. In the simulation with the full unmodified skeleton we saw the DRRT planner creates paths through both outside and inside of the office to find a path. It finally finds the solution path (the first one to reach the goal) outside the office rooms which might not be desirable. After manually modifying the skeleton and running the strategy again, we saw a very large difference in the number of samples it took to find the desired path.

We found that for the unpruned skeleton environment, it took 1,006 samples, 19,882 collision detections, as well as just under a 10 second run time. In contrast, our manually pruned skeleton environment, it took only 267 samples, 4,622 collision detections, and ran in just under 1 second (0.642 sec). It is very obvious to tell that even this small impact of guiding the skeleton toward the right direction greatly impacted the performance of the planner.

With the user-modified skeleton, the planner was able to only expand the tree through user-defined paths/regions and was able to find the path quickly and with many less samples.

B. Property-based Skeleton Editing

In our second experiment, we used the automatic modification or pruning of skeleton using the property tools of the skeleton. The purpose of this experiment is to try and see if our property tools could use the clearance requirement of the robot to prune out the edges in the skeleton to realize the paths not wide enough as non accessible one and therefore not plan in those region. We created an environment (as shown in

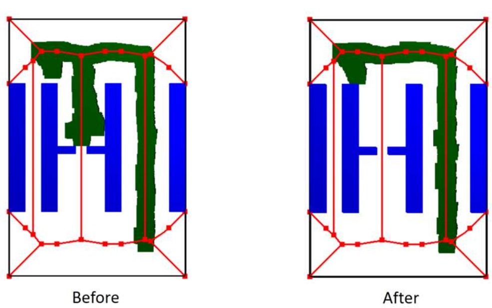

Fig. 7. Experiment 2

Figure 7) with three different paths to reach our goal. One narrow enough such that it did not allow the robot to fit at all, another that could fit the robot through the entire path except for a narrow bottle neck at the middle making it difficult for the robot to pass through. Similar to the previous experiment, we first ran the experiment with the skeleton with DRRT algorithm. We would then compare that to the simulation of the same environment but with the automatically pruned skeleton based on the clearance requirement of the robot. The skeleton edges whose minimum clearance are less than the radius of the bounding sphere of the robot are deleted. With our property map, we expect that our pruning strategy will be able to guide the planner away from narrow paths and paths with the clearance bottleneck.

In our experiment we found that for the DRRT that did not use the clearance paths property to prune the planner generated 316 samples, called the collision detector 6,775 times, and took about a second and half to run (1.57 sec). In contrast, with the DRRT running using the clearance property to prune itself, the planner generated 233 samples, called 5,312 collision detections, and ran in about three-fourths of a second (0.782 sec). Although the results aren't as obvious as the as in the first experiment you can obviously see that the planner that pruned the second path did not waste time by generating samples in the region. Visually, it was much more oriented towards reaching the goal because it did not waste time trying to find samples in paths that are unaccessible.

V. Conclusion

Workplace skeleton tools can be very useful in helping guide planners toward paths that are more desirable and steer away paths that may be unaccessible and a waste of resources. With the creation of our property map on our workspace skeleton, we are able now encode important topographical information that previously may not have been even considered. With the use of that topographical information such as clearance we showed that we can improve DRRT strategies by allowing the planner to take into considerations whether or not the robot could even get through the path. We also show that with the use of user-guided motion planning we can use the skeleton that we create or randomly generate to guide the DRRT planner towards paths that are more desirable.

Acknowledgments

We would like to thank the USRG and DREU programs for sponsoring our stay at Texas A&M University as well as our research. We would also like to thank Dr. Nancy Amato for letting us do research with her lab, Parasol Labs, as well as with the help of post-doc Shawna Thomas. This research wouldn't have been possible without the guidance of Mukulika Ghosh, our mentor, and all of her support on our project.

References

[1] J. Denny, R. Sandstrom, A. Bregger, and N. M. Amato. Dynamic region-biased rapidly-exploring random trees. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, San Francisco, CA, December 2016.

[2] C. Guo, T. Tarn, N. Xi, and A. Bejczy. Fusion of human and machine intelligence for telerobotic systems. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3110–3115, 1995.

[3] I. Ivanisevic and V. J. Lumelsky. Configuration space as a means for augmenting human performance in teleoperation tasks. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 30(3):471–484, Jun 2000.

[4] J. C. Latombe. Motion planning: A journey of robots, molecules, digital actors, and other artifacts. *Int. Journal of Robotics Research*, 18(11):1119–1128, 1999.

[5] S. A. Wilmarth. *A Probabilistic Method for Rigid Body Motion Planning Using Sampling from the Medial Axis of the Free Space*. PhD thesis, Dept. of Math., Texas A&M Univ., College Station, TX, 1999.