Exploiting collision information in probabilistic roadmap planning

Serene W. H. Wong and Michael Jenkin
Department of Computer Science and Engineering
York University, 4700 Keele Street
Toronto, Ontario, M3J 1P3
{swong, jenkin}@cse.yorku.ca

Abstract—This paper develops a novel approach to combining probabilistic motion planners. Rather than trying to develop a single planner that works over a wide range of environments, we develop a strategy for combining different motion planners within a single framework. Specifically we examine how planners designed for open spaces and those designed for narrow passages can be integrated within a single planning framework. Information that is normally discarded in the planning process is used to identify regions as being potentially ‘narrow’ or ‘cluttered’, and we then apply the planner most suited for that region based on this information. Experimental results demonstrate our approach outperforms the basic PRM approach as well as a Gaussian sampler designed for narrow regions in three test environments.

I. INTRODUCTION

Probabilistic roadmap methods (PRM) [1] have proven very successful in solving the motion planning problem for high dimensional configuration spaces. Although these methods have proven to be effective, no ‘best’ strategy within the family of PRM algorithms has emerged. Different PRM planners perform better under different environmental conditions. For example, the performance of simple PRM strategies degrades when a robot’s workspace contains narrow passages [2], [3]. Thus, many variants of PRM have been developed e.g. [2], [3], [4], [5], to operate in cluttered and other less-open environments. As no single PRM seems ideally suited for all possible environments, a number of researchers (e.g., [6], [7], [9]) have integrated different motion planners within a single framework. Following this approach, here we develop a method to combine different motion planners within an integrated planning framework. We apply individual motion planners in regions that they are likely to perform well and then to combine these ‘locally best’ planners to plan globally. In open regions, a basic PRM may be appropriate while in clustered areas, an algorithm tuned to cluttered regions should be used instead. In order to merge these local planners, the problem becomes one of determining regions that are ‘open’ versus regions that are ‘cluttered’ or ‘narrow’.

This paper examines how information about configuration space can be used to control the region subdivision process and drive the selection of different planners in different regions. The rest of the paper is organized as follows. Section II reviews related work in PRM. Section III describes how collision information can be used when integrating local planners within a single framework. Section IV presents experimental results, and Section V summarizes this work and suggests some directions for future work.

II. RELATED WORK

There has been significant work in integrating different motion planners together within a PRM. Dale et al. [6] proposed a metaplanner for using different PRMs. This metaplanner first characterizes the workspace into different regions based on predefined characteristics. Then, based on the region type, one or more planners are assigned to each subregion. They define several characteristics of the environment that can be used to divide the region including, the free space ratio, the number of connected components, the size of connected components, and the surface ratio which is used to determine whether the surface is rough or smooth. The subdivision of regions based on the characteristics is suggested, but no automatic mechanism is proposed. Morales et al. [7] developed a machine learning approach for feature-sensitive motion planning. The basic idea here is to categorize and identify regions using a machine learning approach. Regions are classified to be either a narrow passage, cluttered, free, or non-homogeneous. After regions are identified, appropriate planners are assigned to each sub-region. However, this approach has two basic drawbacks as observed in [8]. First, when partitioning regions, information that was already obtained from characterizing the region is not used. Instead, random subdivisions are performed. Second, components are connected using a brute-force method where a k-closest connections are created over all nodes. This accounts for 50% to 90% of the collision-detection calls in the approach. In order to address these problems [8] proposes that only nodes in the overlapping areas between different regions should be considered and to randomly create connections over all nodes.

An alternate approach to these static combination approaches is to combine different samplers in an adaptive fashion during planning. Hsu et al. [9] proposed an adaptive hybrid sampling method to systematically combine samplers. In this approach multiple sampling planners are used. Each planner is assigned a probability and the planner to use is picked based on this probability. Depending on the effectiveness of the chosen planner, its weight is adjusted. Thus, the more useful the planner is, the more likely it is that it will be chosen.

The problem with the approach described in [9] is that it assumes a stationary likelihood of ‘best planner’. It does not recognize that one planner may work well in one portion of the environment, while another might work well somewhere else.
III. COMBINING PROBABILISTIC ROADMAP PLANNERS

The goal of this work is to develop a planning framework that enables the combination of different planners such that the strengths of planners manifest in individual regions can be exploited. Our approach uses history information obtained during the planning process to characterize regions. Furthermore, this method is designed such that the characterization of regions is simple, and can be performed automatically.

A. The use of history information

In PRM, configurations are sampled randomly (or sampled randomly subject to some bias function). If the configuration is in a free space, then it is added onto the roadmap. If the configuration is not in the free space, then this sample is discarded. This information is valuable as it actually provides information about the workspace. In particular, the process of determining that a configuration is occupied provides information about the specific obstacle (or obstacles) that causes this sample to lie outside of C_free. Assuming that this information is retained when performing the intersection of some configuration of the robot with configuration space, then this information is ‘free’. How can this information be exploited during planning?

Assume that the determination that a particular pose $s$ is in configuration space determines not only if $s$ is in C_free but if it is not in C_free that it also determines the specific object that $s$ intersects. We can use this information to identity regions that are likely to be associated with narrow passages between different objects (and hence difficult to probe with simple randomized path planners (RPP’s)), and then apply appropriately tuned sampling algorithms in these regions.

B. The approach

Our method first randomly samples the workspace using a sampler designed for ‘open’ regions (e.g., a uniform distribution sampler). Two sets of configurations, $C_{free}, C_{obstacle}$ are maintained. If the sampled configuration intersects an obstacle, this configuration is assigned to $C_{obstacle}$. If the sampled configuration is collision free then it is assigned to $C_{free}$.

The set $C_{obstacle}$ is used to identify narrow or cluttered regions. For each pair of configurations in $C_{obstacle}$, the distance between them is calculated. If the distance between them is less than a predefined value $2r$, and their collisions with obstacles are not from the same obstacle, then between these two obstacles a potential narrow or cluttered region may exist. Let A and B be elements in $C_{obstacle}$ that correspond to the identified obstacles A and B. A potential narrow or cluttered region is defined using the centroid of A and B. Denote this centroid as N(A,B).

After potential narrow or cluttered regions have been identified, different planners are applied in these regions. The basic approach is shown in Fig. 1. p1, p2, p3, p4 belong to $C_{obstacle}$. p5, p6, p7, p8, p9, p10 belong to $C_{free}$. The
cluttered region sampler (here a Gaussian [4] sampler is used for performance evaluation) generates configurations near \(N(A,B)\). In the example shown in Fig. 1, two configurations \(x_1\) and \(x_2\) in the narrow or cluttered region are generated. \(x_1\) and \(x_2\) are added into \(C_{\text{free}}\) and \(C_{\text{free}}\) now has the following elements: \(p_5, p_6, p_7, p_8, p_9, p_{10}, x_1, x_2\). A roadmap based on \(C_{\text{free}}\) is then created based on the full set of configurations.

C. Computational Overhead

The computational cost of this approach is not much more than the basic PRM (or its variants) approach. We have used information that the basic PRM normally discards in order to identify narrow or cluttered regions. The only additional cost is in computing the narrow or cluttered region, which involves maintaining the identity of the obstacles that were found during the random configuration phase and coding the obstacles in the environment so that individual obstacles (or portions of obstacles) are labeled separately.

IV. EXPERIMENTAL RESULTS

In this section, we first illustrate our approach using a simple 2D environment. We then examine the performance of the algorithm on more complex environments.

A. 2D Illustration

The motion planning program provided by Garlos Guestrin [11], and the graph theory package provided by B. W. Bush [10] were used to construct a simple motion planner. Fig. 2 shows the workspace. In Fig. 2 the black solid circles denote obstacles in the workspace. The robot is to go from the start to the goal passing through the narrow passages. First, the basic PRM algorithm is run. If the sampled configuration intersects an obstacle, the configuration is assigned to \(C_{\text{obstacle}}\). If the sampled configuration is collision free, the configuration is assigned to \(C_{\text{free}}\). Based on the configurations in \(C_{\text{obstacle}}\), narrow regions are determined. In Fig. 3, the white circles show potential narrow/cluttered regions that were identified in the workspace. There are four narrow/cluttered regions identified in this execution. A second planner is used to generate configurations within the identified narrow/cluttered regions. The dark grey circles inside the white circles in Fig. 3 represent configurations generated in the narrow/cluttered regions. A roadmap is then generated using both sets of configurations.

B. Performance evaluation

In order to evaluate the performance of the integrated planner, we compared its performance against a basic PRM and a Gaussian sampler. Basic PRM’s tend to work well in open spaces while Gaussian samplers tend to work well in cluttered regions. We tested our implementation on an Intel Centrino Duo Core Duo U2500 1.2GHz with 1GB of RAM, running Fedora 6 OS. We used the motion planning kit provided by Stanford University [12] and the GNU Scientific Library [13] to implement PRM, the Gaussian sampler and our approach.

We have chosen three environments that contain different kinds of obstacles to compare the different algorithms. The

<table>
<thead>
<tr>
<th>Environment</th>
<th>PRM</th>
<th>Gaussian</th>
<th>Combined Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0%</td>
<td>10%</td>
<td>40%</td>
</tr>
<tr>
<td>2</td>
<td>10%</td>
<td>10%</td>
<td>30%</td>
</tr>
<tr>
<td>3</td>
<td>10%</td>
<td>20%</td>
<td>40%</td>
</tr>
</tbody>
</table>

TABLE I

The percentage of times that the algorithm is able to find a path. Each experiment was repeated ten times for each environment.
first environment consists of a plate with a hole in the middle (Fig. 4). The plate is built out of four objects so that the determination of collision with separate objects was possible. This plate partitions the workspace into two portions, the top and the bottom portion. In order for the robot to move from the top portion to the bottom portion, it has to go through the hole in the middle of the plate, which is a narrow passage. The second environment contains four rectangular bars with narrow spaces between each of them (Fig. 5). This environment is highly cluttered. The third environment contains two plates with a hole in the middle of each of them (Fig. 6). As in the first environment, the plates are each built out of four objects. These two plates partitions the workspace into three portions. In order for the robot to move from the top portion to the bottom portion of the workspace, it has to go through two holes. The robot used in all three environments is a complex rigid object.

Fig. 7 shows the process of moving the robot through the hole in sample environment 1. The robot must undergo complex maneuvers in order to move it through the small opening. Planning a path that finds a collision free path through the hole is quite difficult, and as will be shown below, was not possible with the basic PRM algorithm.

We applied the basic PRM, the Gaussian sampler [4], and our combined approach to each of the three test environments. We evaluated the percentage of times that each algorithm was able to find a path for the robot. Each experiment was repeated ten times for each environment, and the results are shown in Table I.

For environment one, 30,000 attempted roadmap sample point and edge creations were made in all three methods. Even with 30,000 possible sample points and edges, the basic PRM method was not able to locate a path. This is because the hole in the wall creates a narrow passage for the robot and the basic PRM method tends to perform poorly in this condition. The Gaussian sampler performs better than the basic PRM approach and was able to locate a path 10% of the time. However, the combined approach performs best, and was able to locate a path for the robot 40% of the time.

In environment two, 1,000 attempted roadmap sample point and edge creations were made. Both the basic PRM and the Gaussian sampler found a path 10% of the time. The combined approach was able to locate a path in 30% of the time. In environment three, 2,000 attempted roadmap sample point and edge creations were made. 10% of the time, the basic PRM was able to find a path, 20% of the time, the Gaussian sampler was able to find a path. However, the integrated approach was able to find a solution 40% of the time. These results show that in workspaces that contain narrow or cluttered region, our approach outperforms the basic PRM approach as well as the use of Gaussian sample approach by itself.

These results show that the framework of using history information to characterize regions is beneficial. It outperforms the basic PRM approach, and outperforms the use of a tuned cluttered region algorithm in dealing with cluttered region by itself.

V. CONCLUSIONS

This work develops a framework that combine multiple probabilistic roadmap planners such that the strengths of the individual planners can be exploited in different portions of the environment. The approach uses history information that is normally discarded to characterize regions. The results show that this integrated approach outperforms the basic PRM method as well as an algorithm designed for cluttered regions in all of the three test environments.

Although the preliminary results for this approach is promising, future work is needed. One issue is how to establish the predefined distance value for determining whether there exists a narrow passage/cluttered region between different obstacles. This distance is domain specific. For example, in a hospital, the door entrance from a hallway could be a narrow passage. However, the predefined distance in the molecular targeting domain would be much smaller than the door entrance in a hospital. Another potential direction for future research is to explore if this history information might be used to characterize other properties of the environment to drive the use of other types of planners.

The approach described here only identifies narrow regions between obstacles. It does not deal with narrow regions caused by the same obstacle. This is perhaps illustrated best by the example shown in Environment 3 (Fig. 6). The hole in the plate was constructed by joining together four simple objects. This makes the small opening a narrow region between objects not a narrow region associated with a single object. The process of establishing narrow regions associated with a single object is also an interesting direction for future research. Two approaches seem promising. One approach would be to subdivide the single object into multiple objects and then process these multiple objects using the approach described...
Fig. 7. An animation of the robot moving through the narrow passage. Moving through the opening requires complex maneuvering of the robot through the opening.

An alternative would be to use a different planner (e.g., [7], [8], or [9]) on these individual objects in isolation and then to combine a ‘narrow object planner’ with a ‘narrow regions between objects planner’ when constructing paths in the full environment.

REFERENCES