

Reachability Analysis of Sampling Based Planners*

Roland Geraerts and Mark H. Overmars
Institute of Information and Computing Sciences
Utrecht University
3584 CH Utrecht, the Netherlands
{roland,markov}@cs.uu.nl

Abstract—The last decade, sampling based planners like the Probabilistic Roadmap Method have proved to be successful in solving complex motion planning problems. We give a reachability based analysis for these planners which leads to a better understanding of the success of the approach and enhancements of the techniques suggested. This also enables us to study the effect of using new local planners.

Index Terms—Reachability analysis, potential field local planner, PRM, motion planning

I. INTRODUCTION

A central problem in robotics is planning a collision-free path for a moving object in a rigid environment. In 1979, Reif showed that path planning for a polyhedral robot among a finite set of polyhedral obstacles was PSPACE-hard [1]. Four years later, Schwartz and Sharir proposed a complete general-purpose algorithm based on an algebraic decomposition of the configuration space of any fixed dimension d . When the space of collision-free placements is a set defined by n polynomial constraints of maximal degree m , a path can be computed by an algorithm whose time complexity is doubly exponential in d and polynomial in both n (geometrical complexity) and m (algebraic complexity) [2]. In 1986, this algorithm was improved to a single exponential time algorithm [3]. Due to their complexity, these algorithms have never been used in practice. In 1988, Canny found a PSPACE algorithm for the general mover's problem and showed that it was PSPACE-complete [4], showing that exact planners have little chance of solving complicated problems. In 1991, Barraquand created the Randomized Path Planner, which was an artificial potential field planner [5]. This planner did not require the construction of an explicit description of the configuration space. Although this planner was much faster than previous approaches, it could get stuck at a local minimum of the potential function. One year later, different researchers independently devised the Probabilistic Roadmap Method (PRM), which was successfully able to deal with many motion planning problems [6], [7]. This success is due to its wide applicability and good performance: if a solution for a problem is easy, then it seems that the PRM can solve it very fast.

The probabilistic roadmap method consists of two phases: a construction and a query phase. In the construction

phase, a roadmap (graph) is built, approximating the motions that can be made in the environment. First, a free random sample (configuration) is created. Such a sample describes a particular placement of the moving object in the workspace. Then, a simple local planner is employed to connect the sample to some useful neighbors. A neighbor is useful if its distance to the new configuration is less than a predetermined constant. Samples and connections are added to the graph until the roadmap is dense enough. In the query phase, the start and goal samples are connected to the graph. The path is obtained by a Dijkstra's shortest path query. See e.g. [8] for a more extensive elaboration of the PRM method.

The (analysis of the) complexity of a motion planning problem is often expressed in terms of geometric complexity (of the obstacles and moving object) and the number of degrees of freedom (DOFs) of the moving object. This is reasonable for methods that are based on the geometry of obstacles such as visibility graphs, Voronoi diagrams and exact cell decompositions [9]. In practice, these methods fail when the geometric complexity is high or when there are many (> 4) DOFs or many primitives involved.

Complexity analysis is also employed for sampling based planners. Analyses for these planners use the coverage of the free configuration space ($\mathcal{C}_{\text{free}}$) with (hyper)spheres which results in exponential complexity bounds, see e.g. [10]. In practice though, the PRM can successfully handle this curse of dimensionality because it is *reachability* based, i.e. a sample can often be connected to other samples that are far away because they can be reached by the local planner. For example, if each sample 'sees' a large part of $\mathcal{C}_{\text{free}}$, then $\mathcal{C}_{\text{free}}$ will be covered and connected quickly. This does not follow from the standard analysis that only allows a sample to be connected to its adjacent neighbors. Another reason why the PRM method is fast is because its primitive operations are simple. Checking samples for collisions does not require an explicit representation of the configuration space (whose combinatorial complexity can be very high). When a path or a sample is checked for collisions, only the obstacles in the vicinity are involved. As a result, 'redundant' primitives on the other side of the scene do not affect the performance. These properties lead to a favorable performance that is proportional to some measure of difficulty for the problem to be solved.

In this paper, we will study properties of commonly used techniques in sampling based planning by performing a reachability analysis which emphasizes the notions of

*This research was supported by the Dutch Organization for Scientific Research (N.W.O.). This research was also supported by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-2001-39250 (MOVIE - Motion Planning in Virtual Environments).

coverage and maximal connectivity. These concepts are introduced in section II. In section III, we describe the experimental setup. In sections IV, V and VI we analyze various techniques, leading to a better understanding of these techniques. We observe that the main difficulty is not getting C_{free} covered, but getting the nodes connected, especially when the problem gets more complicated. We conclude in section VII that a hybrid sampling technique and a new potential field local planner we propose lead to a better performance of the PRM approach.

II. COVERAGE AND MAXIMAL CONNECTIVITY

The PRM was designed to be a multiple shot planner which enables fast querying. This goal can be reached by creating a graph $G = (V, E)$ that covers C_{free} and captures its connectivity. We define coverage and maximal connectivity as follows:

Definition 1 (coverage). G covers C_{free} when each configuration $c \in C_{\text{free}}$ can be connected using the local planner to at least one vertex $v \in V$.

Definition 2 (maximal connectivity). G is maximally connected when for all vertices $v', v'' \in V$, if there exists a path in C_{free} between v' and v'' , then there exists a path in G between v' and v'' .

Coverage ensures that every query (which consists of a start and goal configuration) can be directly connected to the graph, as is required to solve the problem. If there exists a path (in C_{free}) between the start and goal configuration, then maximal connectivity ensures that a path between them can be found in the graph. Note that the path in the graph and the path in C_{free} do not have to be in the same homotopic class.

The problem has been solved if both criteria have been satisfied. Also other criteria are of course imaginable, for example creating a graph that optimizes path quality (see e.g. [11]), but we will not consider them here. A number of authors have studied the use of PRM for solving single motion planning queries. For single shot problems coverage does not play an important role and our analysis is less relevant.

We use coverage and connectivity as an analysis tool to gain insight in sampling based methods. Our goal is to determine for the various techniques how long it takes before C_{free} has been covered and connected. Because this would be rather complex for a continuous (high dimensional) configuration space C , we discretize C (for problems that arise in 2D and 3D C -spaces): for each cell (whose dimensions are determined by the step size used by the local planner) in C we check whether the placement of the robot for that cell is free and store this information in an array. When a vertex v is added to V , its discretized reachability region is calculated by checking for each free cell c if there exists a local path between v and c . All free cells that can be connected by the local planner are labeled with a unique region number. If each free cell has been covered by at least one region, the coverage criterion

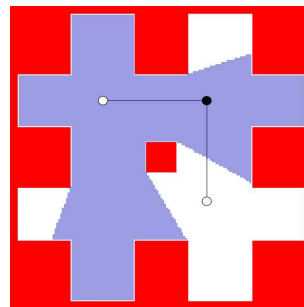
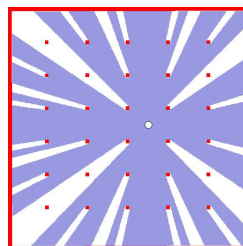


Fig. 1. The coverage and maximal connectivity criteria have been met: the reachability regions of the white vertices cover the complete free space and are connected via the black vertex



(a) A star-shaped 2D reachability region



(b) A 3D reachability region for a manipulator arm with three DOFs

Fig. 2. Complicated 2D and 3D reachability regions

has been met. The connectivity criterion is checked as follows: for each added vertex $v \in V$ we calculate the set of vertices $W \in V$ to which it can be connected through the grid of free cells. Then we add all combinations of v with W to a connectivity list. If there exists a path in G for each connection in the connectivity list then C_{free} is maximally connected. Please realize that these calculations are only done to compare planning techniques. They should of course never be used in the actual planners because they are too slow for this.

Fig. 1 shows an environment whose free space is covered by two (white) vertices and is connected via one extra (black) vertex. The reachability region for the upper left vertex has been drawn. Each configuration in this region can be connected with a straight-line local planner to the vertex, so a three-node graph suffices to solve this problem. The shape of a reachability region can be complicated. Fig. 2(a) shows a region for a 2D environment with many pins and Fig. 2(b) shows a 3D region for the manipulator arm of Fig. 6 with three DOFs.

III. EXPERIMENTAL SETUP

All techniques were integrated in a single motion planning system called SAMPLE (System for Advanced Motion PLanning Experiments), implemented in Visual C++ under Windows XP. SAMPLE automates conducting experiments, i.e. statistics are automatically generated and processed, decreasing the chance for errors. All experiments were run on a 2.66GHz Pentium 4 processor with 1 GB internal memory. We used Solid as basic collision checking

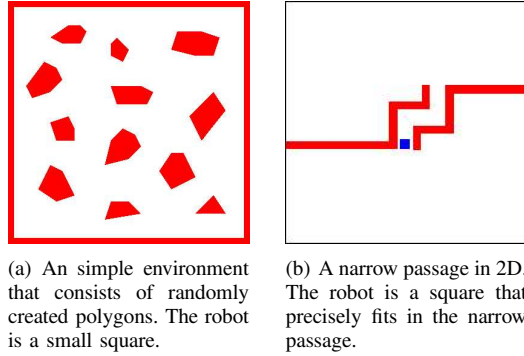


Fig. 3. Scenes used to test the sampling techniques

package [12].

The statistics were all averaged over 100 independent runs. When we performed the experiments, we noted that some had a very large variance which makes results based on averages less useful. In such cases we discarded the experiments and conducted them again. Nevertheless, it is important to recognize that some PRM choices can lead to high variances. Fortunately, these can be decreased by restarting the PRM once a while [13].

For each experiment, we ran the technique until both coverage and maximal connectivity had been achieved. We record the following statistical data that will be used below to compare the approaches. Let n be the number of regions $r_i \in R$ discovered so far. We record statistics regarding the number of regions on the moment that C_{free} was covered and the moment that C_{free} was maximally connected (after the problem has been solved). Furthermore, let $|r_i|$ be the number of cells in region i and $|C_{\text{free}}|$ the total number of free cells.

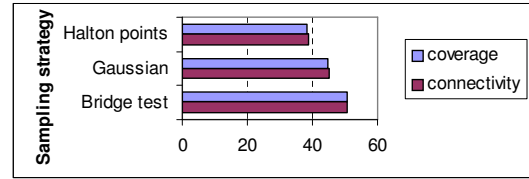
Definition 3 (number of regions). *Each $v \in V$ implies a new region. The number of regions is denoted by n .*

Definition 4 (average size of regions). *The average size of the regions is the total number of cells contained in all regions divided by the number of regions; this number is normalized by dividing it by the number of free cells: $\frac{1}{n} \sum_{i=1}^n |r_i| / |C_{\text{free}}|$.*

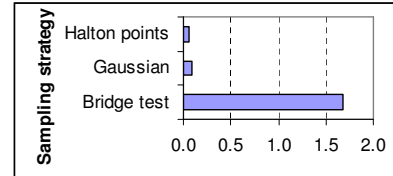
Also the construction time, number of samples, number of collision checks and number of local planner calls are recorded. Because we are not interested in the time spent on checking the two criteria, we did not include this amount of time in the construction time.

IV. SAMPLING

The PRM has been expressed as a sampling based motion planning method. In this section we will study the behavior of different sampling techniques. Three categories of sampling techniques have been proposed: uniform, non-uniform and hybrid techniques. For each category we choose a representative, for the uniform technique we choose 'halton', for non-uniform 'gaussian' and for hybrid we choose 'bridge test'. See e.g. [13] for an extensive elaboration on these techniques. We will compare their behavior

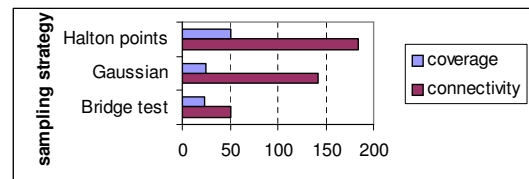


(a) number of regions

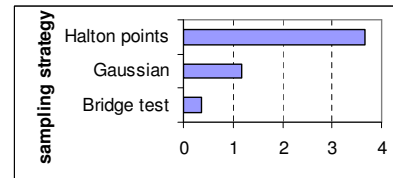


(b) running time (s)

Fig. 4. Statistics for the simple environment of Fig. 3(a)



(a) number of regions



(b) running time (s)

Fig. 5. Statistics for the narrow passage environment of Fig. 3(b)

by considering the experiments we performed on the two environments depicted in Fig. 3. These environments are representative for many different motion planning problems so we expect the observations to apply rather general. The first environment consists of 16 polygons through which a small robot must navigate. It should be easy to create a maximally connected roadmap for this environment, because all possible reachability regions can 'see' a large portion of the free space. The second environment has been designed to be more difficult, i.e. it contains a narrow passage through which a square robot has to move. The passage is surrounded by two large open spaces. The results of the experiment are visualized in Fig. 4 and Fig. 5.

The first category of sampling techniques comprises the uniform techniques, such as random, grid, cell-based and halton sampling. It is well known that these techniques can have difficulties with the narrow passage problem [13]. While halton performed well in the simple environment, it had troubles in finding its way through the narrow passage: its uniform distribution created too many samples in the two wide open areas and too few in the narrow passage.

Because some samples are needed to cover the space in the narrow passage and the chance is small to sample there, the total number of regions was higher for this method than for the other methods. Many researchers recognized that an opposite behavior is needed: few samples should be created in a large open space and more samples in a narrow passage, which led to the development of non-uniform sampling techniques.

The second category biases the sampling distribution. Some techniques try to tackle the narrow passage problem by adding more samples in difficult regions of the environment, i.e. they filter out samples that probably do not contribute to the coverage and maximal connectivity of the roadmap. Examples include gaussian and obstacle-based sampling. The gaussian technique needed fewer samples than halton in the narrow passage environment. The reason for this is that relatively more samples are concentrated in the difficult areas of the \mathcal{C} -space, which resulted in faster coverage. Also, the open space was fewer times covered (due to the Gaussian distribution of the samples). As this distribution generates fewer samples in open space and more near obstacles, it can though be difficult to connect them. This explains why it took four times as many regions to connect the space compared to covering it.

Recently, hybrid techniques (such as the bridge test) have been proposed which combine the strengths of the previous two categories. The bridge test concentrates samples in difficult areas but it also generates some samples in open areas. This results in a faster connectivity. We think that also other combinations of existing sampling strategies could be suited to serve as a hybrid technique.

By looking at the charts of Fig. 4 and Fig. 5, we can make an important observation. For the simple environment, the difference between the moment that $\mathcal{C}_{\text{free}}$ is covered and the moment that $\mathcal{C}_{\text{free}}$ is maximally connected is very small. In contrast, for the narrow passage environment this difference is much larger. It seems that covering $\mathcal{C}_{\text{free}}$ is not the problem, but getting $\mathcal{C}_{\text{free}}$ maximally connected is more difficult when the environment contains a narrow passage. To make this more clear, we considered two versions of an environment containing a narrow passage: the first variant was the 3D manipulator environment depicted in Fig. 6. The second variant was the same as the first one, except that we made the passages narrower by scaling the workspace in the y-direction, i.e. the workspace became 25% less high. Concerning the first variant, all sampling strategies needed about 150 regions to cover the space. Only a few more regions were needed to connect the space. While the second variant needed the same number of regions to cover the space, the number of regions to connect the space now became five times as large. Hence, connecting samples near or in narrow passages is more difficult. The important observation that can be made is that the narrow passage problems is not so much caused by coverage but by connecting the nodes. Rather than concentrating on more clever sampling, it might be beneficial to spend more effort on connecting

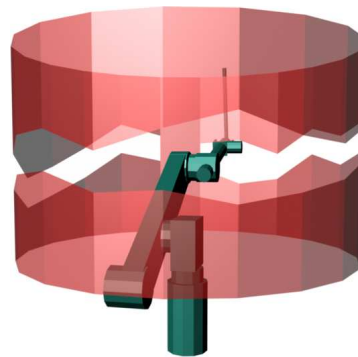


Fig. 6. Manipulator arm with three DOFs

nodes in difficult regions. Actually, already one of the first papers on PRM [7] did this by trying to connect difficult nodes in a second phase using some bouncing strategy. In section VI we will show how more powerful local planners can be used for this as well. The challenge is to apply such a connection strategy only when and where it is necessary.

Ideal sampling strategy

An ideal sampling strategy should create a small amount of samples that covers and connects $\mathcal{C}_{\text{free}}$. The smaller the number of regions, the less time is needed to connect those regions which is the most time consuming step in the PRM. However, some overlap is wanted because this simplifies creating connections between them. This goal can be achieved by creating a hybrid technique which filters out samples that do not contribute to extra coverage or maximal connectivity. The visibility sampling technique tries to achieve this by throwing away nodes [14]. The approach though is too strict and should probably be combined with other sampling techniques. We are currently investigating a different sampling strategy that uses workspace information as a guide to creating a small set of samples that covers and maximally connects $\mathcal{C}_{\text{free}}$.

V. NEIGHBOR SELECTING STRATEGY

The neighbor selecting strategy specifies for a particular sample how a set of neighbor samples is chosen to which it is connected. The goal of the strategy is to make the graph connected as fast as possible. A strategy usually selects neighbors based on (a combination of) the following criteria: maximal distance and maximal number of connections tried. We study the effects of different choices for these criteria.

Because a neighbor strategy selects neighbors to which a sample must be connected (with a local planner), a distance metric must be defined. In this paper we use $d = d_t + d_r$ where d_t denotes the translational distance of the origin of the moving object and d_r denotes the distance traveled by the point of the robot furthest from its origin, while performing rotation. It is recognized that the maximum connection distance should not be too small nor too large: although making long connections seems to be important for the PRM, it is in general not useful to

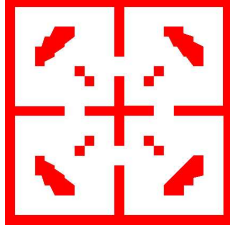


Fig. 7. Test environment for the neighbor selecting strategy

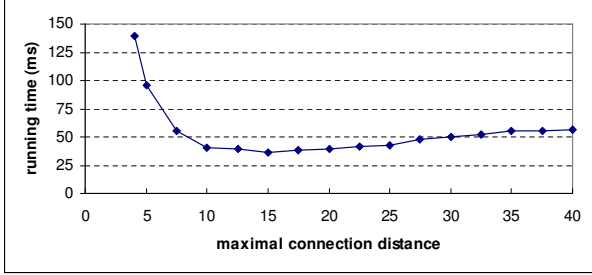


Fig. 8. Relation between the maximal connection distance and running time

make too long connections since the chance of success for such connections is small while the collision checks required for testing the local path are expensive. On the other side, a very small connection distance will always require an exponential number of samples. In the following experiment, depicted in Fig. 7, we vary the maximal connection distance from 5 to 40 (which is the diameter of the environment). It is clear that, as the maximum distance gets larger, the average size of the reachability regions increases and the number of regions needed to solve the problem decreases. So the number of samples required to solve the problem decreases while the time required per sample increases. The relation between the maximum distance and time is plotted in Fig. 8. It shows that there is some optimal tradeoff. Unfortunately, this value is dependent on the environment: if the environment consists of small rooms, a small connection distance is preferable and vice versa. When we make the connection distance very small, PRM starts looking like grid based techniques in which nodes are only connected to their direct neighbors. Fig. 8 shows that this considerably increases the running time. Indeed, the power of PRM is that it can make longer connections.

The second criterion is the maximal amount of connections tried in each step. Also for this criterion, there seems to be an (environment dependent) optimal value. Note that the number of connections does not influence the coverage, but has a clear influence on the connectivity. If the number of connections is too small, it might be hard to get the free space maximally connected because much luck is needed to select those few regions to which a connection is possible. If connections are tried with too many nodes, C_{free} will be maximally connected using less regions, but this might

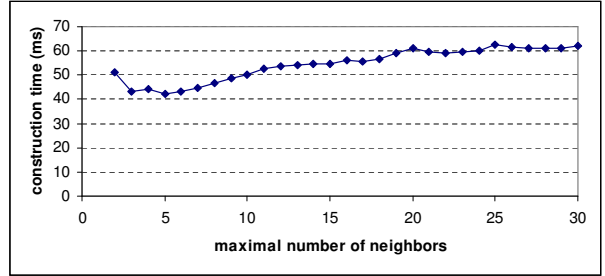
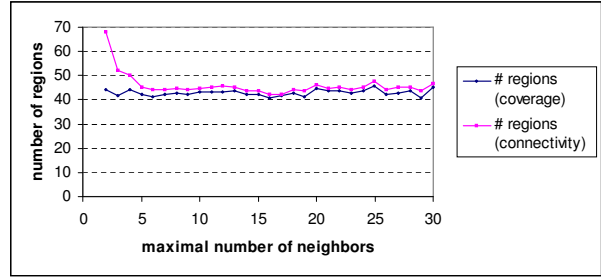


Fig. 9. Influence of the maximal number of connections on the number of regions and running time

negatively influence the running time since testing those connections is expensive. In our experiments, we varied the maximum number of connections. The results are shown in Fig. 9. Indeed, when only a few connections are tried, more regions are needed to get the roadmap maximally connected. Making more and more connections does not seem to be useful, since the number of regions needed to get C_{free} covered and maximally connected is at a certain point constant while the running time does increase.

In conclusion, the optimal parameters for the connection distance and maximum number of connections are dependent of the scene. We expect that sampling based planners can gain some performance if these parameters are made adaptive. For example, in the process of sampling the number of regions increases, so the expected number of regions within the same distance increases. Hence, a constant distance results in creating more (useless) connections which can be avoided by decreasing the connection distance over time.

VI. LOCAL PLANNERS

In the previous sections we showed that it is sometimes difficult to connect certain regions while the coverage criterion has already been met. If we were able to create a local planner that is more powerful than the straight-line local planner (SLLP), then we could decrease the gap between the moment of coverage and maximal connectivity, improving the total running time. Although this new planner might be more expensive, a careful tradeoff between the power and speed of the planner should lead to a better performance of the PRM.

To be successful, this planner should preferably satisfy the following two criteria. First, it must cover at least the same volume as the SLLP does, i.e. it must subsume each reachability region that is created using the SLLP. If the

regions are larger we expect that the space is covered faster. More importantly, because of the larger expected overlap between the regions, they will sooner become maximally connected. Second, the planner should be fast enough to be useful in practice. This can be achieved by letting the planner behave as a SLLP if the connections can be made in a straight line; if the straight-line connection results in a collision, then a more clever approach should be employed. These criteria are satisfied by the simple potential field local planner (PFLP) we will describe below.

In general, a potential field method calculates distances between the robot and obstacles to define a force vector on the robot [9]. These operations make a PFLP expensive in comparison to the simple SLLP. To mitigate this effect, we use a modified version of the potential field planner used in [7].

It is implemented as follows. The planner tries to make small steps on the straight line towards the goal, as does the SLLP. This assures that making such connection will be equally expensive as the SLLP. When the robot walks into an obstacle, the planner checks a step from the last collision-free configuration in a number of directions on the hemisphere oriented towards the goal. The most promising step is considered first. A local minimum is easily detected when all possible steps fail.

While this planner is more powerful, it will be more expensive than the SLLP in terms of consumed time. A second drawback is that a new parameter is introduced that has to be optimized, i.e. the number of directions on the unit sphere has to be chosen. In 2D we choose 8 directions and in 3D we choose 26 directions on the unit sphere and select only those that bring the robot closer to the goal. It is a tradeoff between the accuracy and speed: the higher this number, the larger the reachability region, but the slower the planner is. The number of directions we choose seems to work reasonably, but it is in essence arbitrary.

Fig. 10 shows two different environments for which the reachability region for a particular node, using the SLLP and PFLP is drawn. The first environment consists of 40 polygons through which a small square robot must navigate. The first picture shows a region in this environment that can be reached by a SLLP. While reasonably long connections can be made by this planner, the reachability region of the PFLP significantly extends the area to which connections can be made. This is shown in the second picture. The second environment consists of a narrow passage through which the same robot must maneuver. Besides the advantage of covering larger regions, the PFLP, in contrary to the SLLP, is able to find its way through the narrow passage. This allows connections to be made from one side of the passage to the other one.

To test this approach further, we created two extra 3D environments, see Fig. 11. The Hole environment has two open spaces separated by a wall with a narrow hole in it. The second environment (Corridor) consists of one long corridor with four hairpins. In all experiments the robot is a small cube that can only translate. We expect that

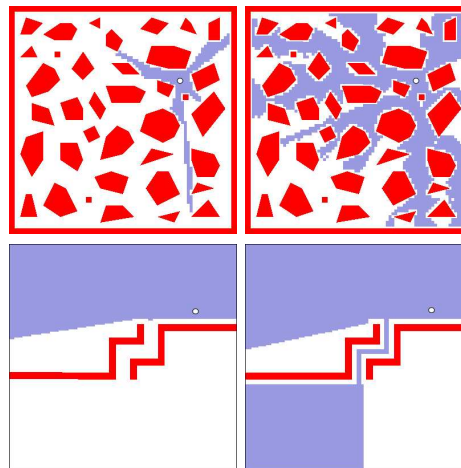


Fig. 10. Reachability regions for straight-line (left column) and potential field local planner (right column).

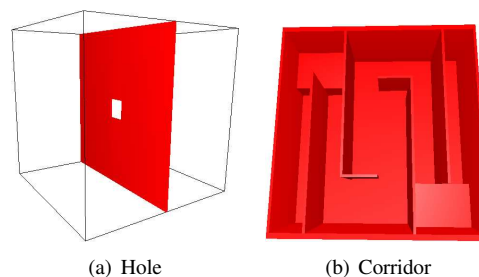


Fig. 11. Two extra environments used for testing the local planners

the PFLP outperforms the SLLP in the two environments depicted in Fig. 10 and the Hole environment, because the reachability regions will be larger than those created with the SLLP. The PFLP will also be able to easily connect the two open spaces in the Hole environment. In the Corridor environment though, the PFLP may only have an advantage in the hairpins; in other situations, much ineffectual work might be done before it is concluded that no connections exist. This is expected to have a negative effect on the running time.

The results are shown in Table I, II III and IV. It shows the number of regions and the average size of the regions, after the moment of coverage (between brackets) and after maximal connectivity. Also, the running times (in ms) using the two local planners are stated.

Three observations can be made from these results. First, for all four environments, the results show, as expected, that the number of regions needed to cover the space are lower for the PFLP, which is caused by the regions being larger. Second, the space was much faster maximally connected. For example, the Clutter environment contained few local minima for the PFLP which resulted in large regions: the average size of the regions created with the PFLP was 58% of C_{free} , compared to only 6% for the SLLP. In the Hole environment, the PFLP had no problems connecting the two open spaces via the narrow passage, i.e. it needed only four regions on average. The Corridor environment was

TABLE I
STATISTICS FOR SLLP AND PFLP IN THE 2D CLUTTER ENVIRONMENT

	#regions	size region	time (ms)
SLLP	(173) 185	(0.06) 0.06	507
PFLP	(11) 11	(0.58) 0.58	45

TABLE II
STATISTICS FOR SLLP AND PFLP IN THE 2D NARROW PASSAGE ENVIRONMENT

	#regions	size region	time (ms)
SLLP	(51) 184	(0.41) 0.42	3678
PFLP	(5) 6	(0.54) 0.55	39

also faster connected (against our expectations), which is caused by the larger regions whose volume is on average 33% of the free space (compared to 26% for the SLLP). Third, the difference between the moment of coverage and moment of maximal connectivity was much smaller for the PFLP than for the SLLP. This is also caused by the fact that the regions have a larger size. Because the space is much faster connected, the PFLP turns out to be an efficient local planner.

Ideally, a local planner should be simple in an 'easy' part of the \mathcal{C} -space and more advanced in more 'difficult' parts. We have seen that a potential field local planner combines those requirements: easy connections (i.e. straight-line connections) are made at the expense of a marginal overhead, while difficult connections (i.e. connections that avoid obstacles) can actually be made. Experiments showed that sampling based methods can benefit from more powerful local planners such as the potential field local planner. Because the local planner is slower, the time improvement is in general less dramatic than the improvement in number of regions required. We believe though that one could improve the time even further. We are currently investigating techniques that can identify difficult regions in the space, which can be used by employing the most appropriate local planner. This should lead to a better performance of the PRM.

VII. CONCLUSION

While classical complexity analysis ends up with exponential complexity bounds, sampling based planners can successfully handle this curse of dimensionality in practice because they are reachability based. In this paper we presented a reachability analysis for these planners. This led to the insight that not coverage is the main problem but getting the nodes connected, especially when the problems get more complicated, i.e. a narrow passage is present. The narrow passage problem can be tackled by incorporating a hybrid sampling strategy that aims at concentrating samples in difficult areas, but it should also generate some samples in large open areas. Another strategy to get $\mathcal{C}_{\text{free}}$ faster maximally connected is to employ a more powerful local planner. We presented a potential field local planner that creates larger reachability regions which eases making con-

TABLE III
STATISTICS FOR SLLP AND PFLP IN THE HOLE ENVIRONMENT

	#regions	size region	time (ms)
SLLP	(5) 40	(0.51) 0.51	106
PFLP	(4) 4	(0.57) 0.57	14

TABLE IV
STATISTICS FOR SLLP AND PFLP IN THE CORRIDOR ENVIRONMENT

	#regions	size region	time (ms)
SLLP	(19) 86	(0.22) 0.26	264
PFLP	(10) 21	(0.34) 0.33	199

nections. Also this planner is better able to find the entry of a narrow passage, decreasing the number of regions needed to get the nodes connected. Experiments showed that this approach leads to a better performance of sampling based methods.

We are currently investigating how to best extend the potential field local planner to higher dimensions and how to use information of the workspace as a guide to more clever sampling and neighbor selecting strategies. Preliminary results show that considerable performance gains can be achieved.

ACKNOWLEDGMENT

The authors would like to thank Dennis Nieuwenhuisen for developing the Callisto collision and visualization toolkit and Marin Hekman for implementing the potential field local planner.

REFERENCES

- [1] J. Reif, "Complexity of the mover's problem and generalizations," in *IEEE Symp. on Foundations of Computer Science*, 1979, pp. 421–427.
- [2] J. Schwartz and M. Sharir, *On the piano movers' problem II*. Ablex Publishing Corporation, Norwood, New Jersey, 1987.
- [3] M.B.-Or, D. Kozen, and J. Reif, "The complexity of elementary algebra and geometry," *Computer and Systems Sciences*, vol. 32, pp. 251–264, 1986.
- [4] J. Canny, *The Complexity of Robot Motion Planning*. MIT Press, 1988.
- [5] J. Barraquand, "Automatic motion planning for complex articulated bodies," Paris Research Laboratory, Tech. Rep., 1991.
- [6] L. Kavraki, P. Švestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [7] M. Overmars, "A random approach to motion planning," Utrecht University, Tech. Rep. RUU-CS-92-32, 1992.
- [8] R. Geraerts and M. Overmars, "A comparative study of probabilistic roadmap planners," in *Workshop on the Alg. Foundations of Robotics*, 2002, pp. 43–57.
- [9] J.-C. Latombe, *Robot Motion Planning*. Kluwer, 1991.
- [10] L. Kavraki, M. Kolountzakis, and J. Latombe, "Analysis of probabilistic roadmaps for path planning," in *IEEE Internat. Conf. Robot. Autom.*, vol. 4, 1996, pp. 3020–3025.
- [11] R. Geraerts and M. Overmars, "Clearance based path optimization for motion planning," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 2386–2392.
- [12] G. van den Bergen, *Collision Detection in Interactive 3D Environments*. Morgan Kaufmann, 2003.
- [13] R. Geraerts and M. Overmars, "Sampling techniques for probabilistic roadmap planners," in *Conference on Intelligent Autonomous Systems (IAS-8)*, 2004, pp. 600–609.
- [14] C. Nissoux, T. Siméon, and J.-P. Laumond, "Visibility based probabilistic roadmaps," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 1999, pp. 1316–1321.