CHAPTER
THREE

Reachability-based analysis

While theoretically, the complexity of the motion planning problem is exponential in the number of degrees of freedom, sampling-based planners can successfully handle this curse of dimensionality in practice. The success of these planners for problems with many degrees of freedom can be explained by the fact that no explicit representation of the free configuration space ($\mathcal{C}_{\text{free}}$) is required. The main operation of these planners is checking configurations for collisions with obstacles in the environment, which can be performed efficiently by the current generation of collision checkers. The second reason for their success is that problems which are not pathological have favorable reachability properties. That is, the free configuration space of a reasonable problem can often be captured with few nodes and each node can reach a large portion of $\mathcal{C}_{\text{free}}$ using a local planner. Therefore, a PRM usually finds a solution quickly, even if the geometric complexity is high.

In the previous chapter, we compared and analyzed techniques based on solving one particular query. In contrast, we now inspect the techniques based on solving every possible query. We say that a motion planning problem is solved if the following two criteria are satisfied. First, each free configuration in $\mathcal{C}_{\text{free}}$ can be reached by at least one node in the graph (*coverage*). Second, for each two nodes in the graph: if these nodes share the same connected component in $\mathcal{C}_{\text{free}}$, then there exists a path between them in the graph (*maximal connectivity*). Our experiments show that covering $\mathcal{C}_{\text{free}}$ is not the main difficulty, but getting the nodes connected, especially when the environments get more complicated, e.g. 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. The strategy must also generate some samples in ample free spaces. Another strategy to get $\mathcal{C}_{\text{free}}$ faster connected is to use a more powerful local planner. We present a potential field local planner that creates larger reachability regions which eases making connections. 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. Our experiments show that this approach leads to a better performance of sampling-based methods.

3.1 Introduction

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. In practice, these methods fail when the geometric complexity is high or when there are many (> 3) DOFs or many primitives involved.

Complexity analysis is also employed for sampling-based planners such as the PRM. 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. [81]). Yet, in practice, 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 can reach a large part of $\mathcal{C}_{\text{free}}$ by using a local planner, 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 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 environment 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 chapter, we will study properties of commonly used techniques in sampling-based planning by performing a reachability analysis which emphasizes the notions of *coverage* and *maximal connectivity*. These concepts are introduced in Section 3.2. In Section 3.3, we describe the experimental setup. In the following three sections, we analyze neighbor selection, sampling and local planning techniques, resulting in a better understanding of these techniques.

We observe that the main difficulty is not getting $\mathcal{C}_{\text{free}}$ covered, but getting the nodes connected, especially when the problem gets more complicated. We conclude in Section 3.7 that a hybrid sampling technique and a newly proposed potential field local planner lead to a better performance of the PRM.

3.2 Coverage and maximal connectivity

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

Definition 3.1 (coverage)**.** *G covers $\mathcal{C}_{\text{free}}$ when each configuration $c \in \mathcal{C}_{\text{free}}$ can be connected using the local planner to at least one node $v \in V$.*

Definition 3.2 (maximal connectivity)**.** *G is maximally connected when for all nodes $v', v'' \in V$, if there exists a path in $\mathcal{C}_{\text{free}}$ 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 connected directly to the graph, as is required to solve one problem. If there exists a path (in $\mathcal{C}_{\text{free}}$) 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 $\mathcal{C}_{\text{free}}$ do not have to be in the same homotopic class.

If both criteria are satisfied, then a path can always be found for every query. Additional criteria are of course imaginable, for example creating a graph that optimizes path quality (see Part II), but we will not consider them here. Several authors have studied the use of PRM for solving single motion planning queries. For single shot techniques, 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 various techniques how long it takes before $\mathcal{C}_{\text{free}}$ has been covered and connected. Because this would be rather complex for a continuous (high-dimensional) configuration space $\mathcal{C}$, we discretize $\mathcal{C}$ (for problems that arise in 2D and 3D $\mathcal{C}$-spaces): for each cell (whose dimensions are determined by the step size used by the local planner) in $\mathcal{C}$, we check whether the placement of the robot for that cell is free and store this information in an array. When a node v is added to V, its discretized *reachability* region is calculated by checking for each free cell c in the array whether

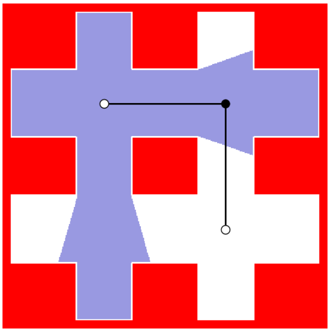
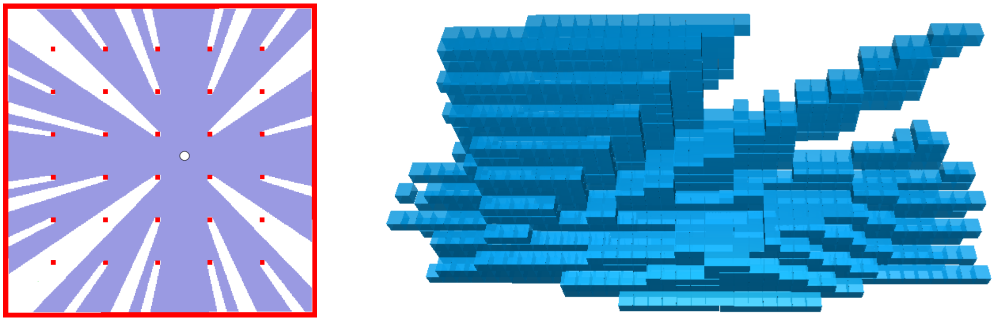


Figure 3.1 The coverage and maximal connectivity criteria have been met. The reachability regions of the white nodes cover the complete free space and are connected via the black node.

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 has been met.[1] The connectivity criterion is verified as follows: for each added node $v \in V$ we calculate the set of nodes $W \subseteq V$ 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 graph G for each connection in the connectivity list then $\mathcal{C}_{\text{free}}$ is maximally connected. Please realize that these calculations are only done to compare planning techniques. They are not part of the actual motion planning algorithm and, hence, are not taken into account for the running times.

As an example, Figure 3.1 shows an environment whose free space is covered by two (white) nodes and is connected via one extra (black) node. Hence, the three-node graph suffices to solve this problem. The reachability region for the upper left node has been drawn. Each configuration in this region can be connected with a straight-line local planner to the node. The shape of a reachability region can be complicated. Figure 3.2(a) shows a region for a 2D environment with many small obstacles and Figure 3.2(b) shows a 3D region for the manipulator arm depicted in Figure 3.3(f) with three rotational DOFs.

[1] We will provide a more efficient way to compute a reachability region in Section 6.4.1.



(a) A 2D reachability region (b) A 3D reachability region for the robot of Figure 3.3(f)

Figure 3.2 Complicated 2D and 3D reachability regions.

3.3 Experimental setup

For the experiments we use our SAMPLE system. As the techniques involve random choices, we report statistics gathered from 100 independent runs for each experiment. See Section 1.2 for more information on our general experimental setup.

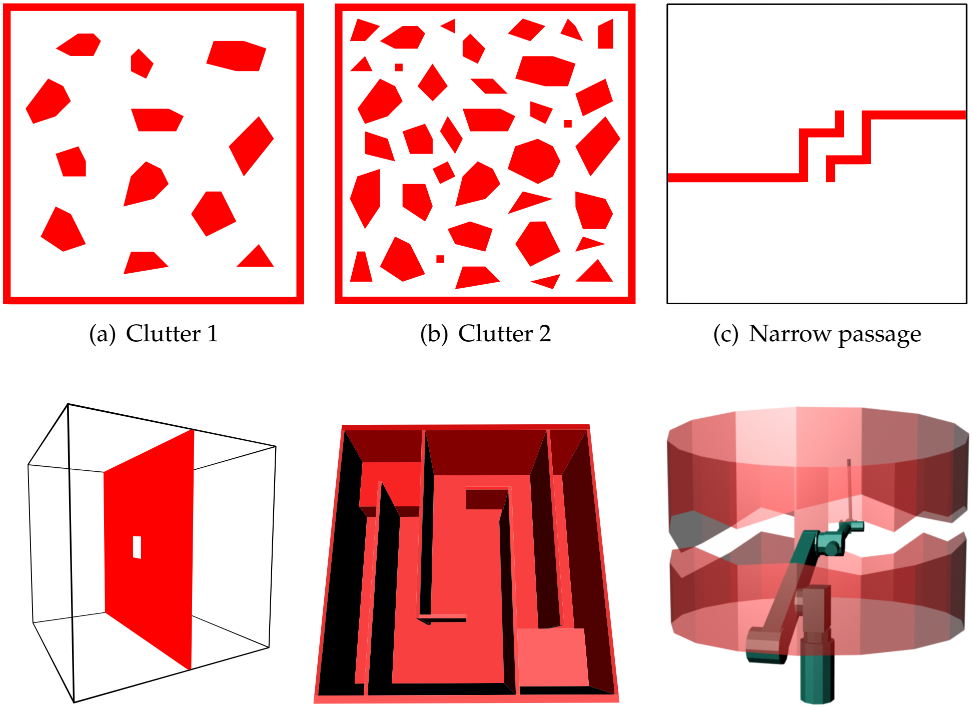
We used the six environments depicted in Figure 3.3. Their bounding boxes are stated in Table 3.1. The environments have the following properties:

Clutter 1 The 2D cluttered 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 each sample covers a large portion of the free space. The robot is a square with two translational degrees of freedom (DOFs).

Clutter 2 We added 24 polygons to the first environment to reduce the average size of the regions. Again, the robot is a translating square.

Narrow passage This environment has been designed to be more difficult. It contains a narrow passage through which a square has to move. The passage is surrounded by two large open spaces. The robot is a translating square.

Hole The Hole environment has two large open spaces separated by a wall with a narrow hole in it. The robot is a small cube that can only translate. Each sample will cover a large portion of $\mathcal{C}_{\text{free}}$.



(a) Clutter 1 (b) Clutter 2 (c) Narrow passage

(d) Hole (e) Corridor (f) Manipulator

Figure 3.3 The six test environments.

Corridor A small translating cube has to move through a 3D winding corridor consisting of four hairpins. The walls of the corridor will limit the size of the reachability regions.

Manipulator This 3D environment features a robot arm with three rotational DOFs which operates in a constrained workspace. The $\mathcal{C}$-space has a long passage.

We discretized the $\mathcal{C}$-space of each environment.[2] The level of discretization can be found in Table 3.2. Consider for example the Clutter 1 environment: The ranges of the translational DOFs of the robot are $[0 : 40] \times [0 : 40]$. The step size used by the local planner is 0.5. Hence, the $\mathcal{C}$-space is discretized with 80×80 cells.

[2]Note that we used other environments than in Chapter 2. Because our computer has a limited amount of memory, we only used robots having at most three DOFs.

	Dimensions of the bounding boxes	
	environment	**robot**
Clutter 1	40×40	0.5×0.5
Clutter 2	40×40	0.5×0.5
Narrow passage	40×40	0.5×0.5
Hole	$40 \times 40 \times 40$	$2 \times 2 \times 2$
Corridor	$40 \times 8 \times 40$	$2 \times 2 \times 2$
Manipulator	variable	variable

Table 3.1 Information on the workspaces of the environments.

	DOF range	**step size**	**number of cells**
Clutter 1	40×40	0.5	80×80
Clutter 2	40×40	0.5	80×80
Narrow passage	40×40	0.5	80×80
Hole	$40 \times 40 \times 40$	2.0	$20 \times 20 \times 20$
Corridor	$40 \times 8 \times 40$	2.0	$20 \times 4 \times 20$
Manipulator	$6.1 \times 0.6 \times 0.7$	0.05	$122 \times 12 \times 14$

Table 3.2 Information on the configuration spaces of the environments.

	Sampling strategy	
	Gaussian	**Bridge**
Clutter 1	4.0	4.8
Clutter 2	4.0	4.8
Narrow passage	1.2	2.4
Hole	4.0	5.6
Corridor	4.0	5.6
Manipulator	1.2	4.0

Table 3.3 The optimal values of σ used in Gaussian and Bridge sampling.

We used the metric from Section 1.2.3. We set the weights for the translational DOFs to 1 and set the weights for the rotational DOFs to 6. The optimal parameters for the the sampling strategies are listed in Table 3.3 and the optimal parameters for neighbor selection strategy are listed in Table 3.4.

For each experiment, we ran the PRM until both coverage and maximal connectivity had been achieved and recorded the following statistical data.

Definition 3.3 (number of regions)**.** *Each node $v \in V$ in the graph implies a new region. The number of regions is denoted by n.*

Definition 3.4 (average size of the regions)**.** *Let n be the number of regions r_i discovered so far. Furthermore, let $|r_i|$ be the number of cells in region i and $|\mathcal{C}_{\text{free}}|$ the total number of free cells. Then the* average size of the regions *equals: $\frac{1}{n}\sum_{i=1}^{n}|r_i|/|\mathcal{C}_{\text{free}}|$.*

We recorded the number of regions and the average size of the regions at two moments: the moment that $\mathcal{C}_{\text{free}}$ was covered and at the moment that $\mathcal{C}_{\text{free}}$ was maximally connected (after $\mathcal{C}_{\text{free}}$ was covered). We also recorded the running time after both criteria had been satisfied. We give an indication of the dispersion of the running times by a *box plot*. Each plot consists of a box, one large and two small horizontal lines and a vertical line. The box represents the middle 50% of the data, the large horizontal line represents the average, the small lines represent the average $\pm$ the standard deviation and the vertical line represents the minimum and maximum value.

3.4 Neighbor selection strategy

The neighbor selection 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: the *maximum connection distance*, the *maximum number of connections* tried, and the *node adding strategy*, see Section 2.5. We chose the *nearest-k* node adding strategy as this method performed reasonably well on different environments. We used the optimal sampling strategy for each environment (see below). That is, we used Bridge sampling for the Narrow passage environment. For the other ones, we used Halton* sampling. We study the effects of different choices for the first two criteria on the six test environments.

3.4.1 Maximum connection distance

We concluded in Section 2.5 that the maximum connection distance should not be too small nor too large. A very small connection distance will always require an exponential number of samples which increases the running times a lot. The PRM works best if reasonably long connections can be made. In general, it is not useful to try 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.

In each of the following experiments, we varied the maximum connection distance from a small value (close to zero) to a large value. The maximum number of connections was set to 75 (see below). Figure 3.5 shows the results for the 2D environments and Figure 3.6 shows the results for the 3D environments.

When we make the connection distance very small, the PRM starts looking like grid-based techniques in which samples are only connected to their direct neighbors. The figures show that this considerably increases the (average and variance of the) running time. Furthermore, there is a large difference in the moment of coverage and the moment of maximal connectivity. This shows that a small connection distance complicates making connections.

It is clear, as the maximum distance gets larger, that the average size of the reachability regions will increase (up to some value), and hence, the number of samples needed to solve the problem will decrease. In other words, the number of samples required to solve the problem decreases when the time required per sample increases. The results show that there is some optimal trade-off which is dependent on the environment (and metric). If the size of a region (that corresponds to a particular sample) is small, then the sample can be connected to other samples to which the distance is small, and *vice versa*. In general, if the average reachability of the samples is low, then a small connection distance is preferable and *vice versa*. See Figure 3.4 which shows the average size of the regions for the six environments corresponding to the optimal neighbor selection parameters. We can make two observations. First, the larger the average size of the regions, the smaller the growth of running times when the maximum connection distance increases. Second, the larger the average size of the regions, the larger the value for the optimal maximum connection distance. This information can be used to estimate the (local) optimal maximum connection distance. For example, when a sample can be connected to other samples at a large distance, then the average size of its reachability region is large. Hence, using a large maximum connection distance is a good choice for choosing neighbors. In addition, few samples should be created near the sample.

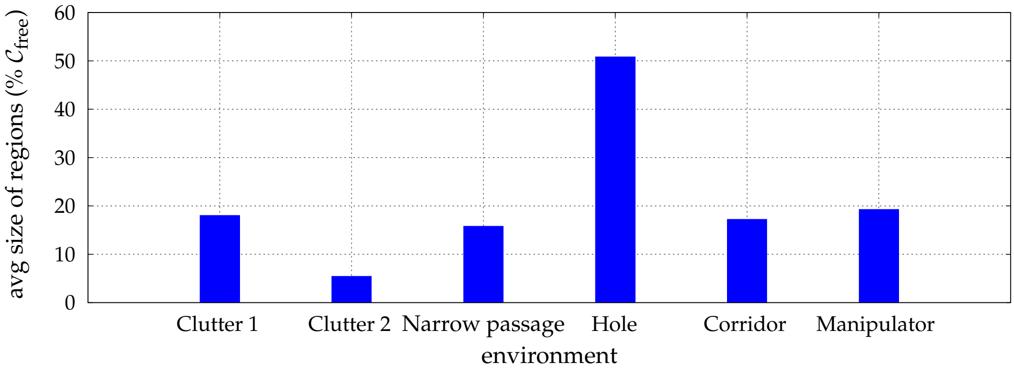


Figure 3.4 The average size of the regions corresponding to the optimal neighbor selection parameters.

The results again confirm that the PRM derives its strength from making long connections. While a very small connection distance has a dramatic negative impact on the running time, a large value has a moderate effect, especially when the average size of the reachability regions is large.

3.4.2 Maximum numbers of connections

The second analyzed criterion for selecting neighbors is the *maximal number of connections* attempted to connect a node. We concluded in Section 2.5.2 that this value should be large. In this section we will show the relation between the maximal number of connections and the coverage and connectivity.

The number of attempted 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 the chance is small that those few samples are selected to which a connection is possible. If connections are attempted with (too) many nodes, $\mathcal{C}_{\text{free}}$ will become maximally connected using less regions. Nevertheless, this might negatively influence the running time since testing those connections is expensive.

In our experiments, we varied the maximum number of connections. Figure 3.7 shows the results for the 2D environments and Figure 3.8 shows the results for the 3D environments. 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 because the number of

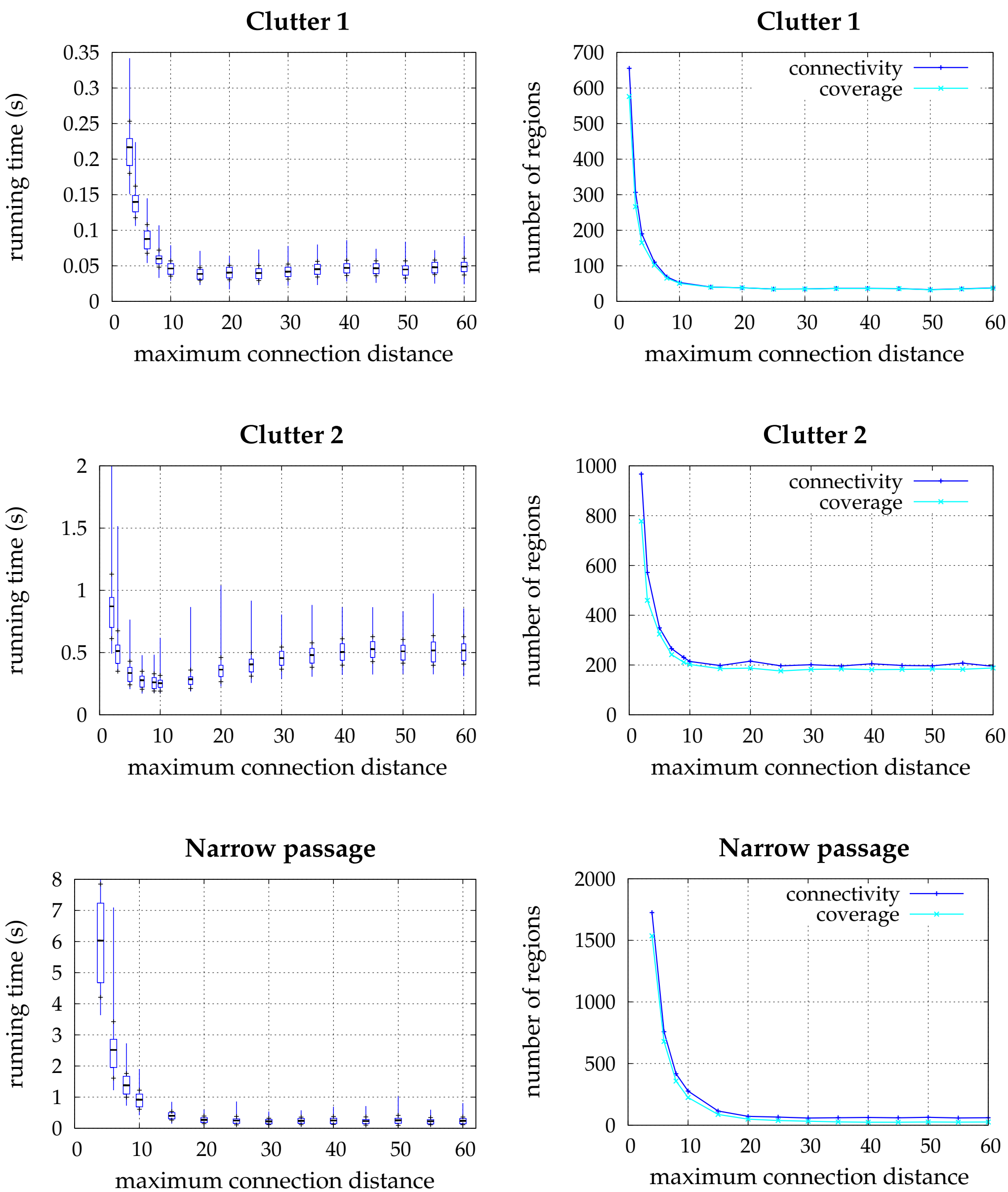


Figure 3.5 Influence of the *maximum connection distance* on the running time and number of regions required to get the free space covered and maximally connected in the 2D environments.

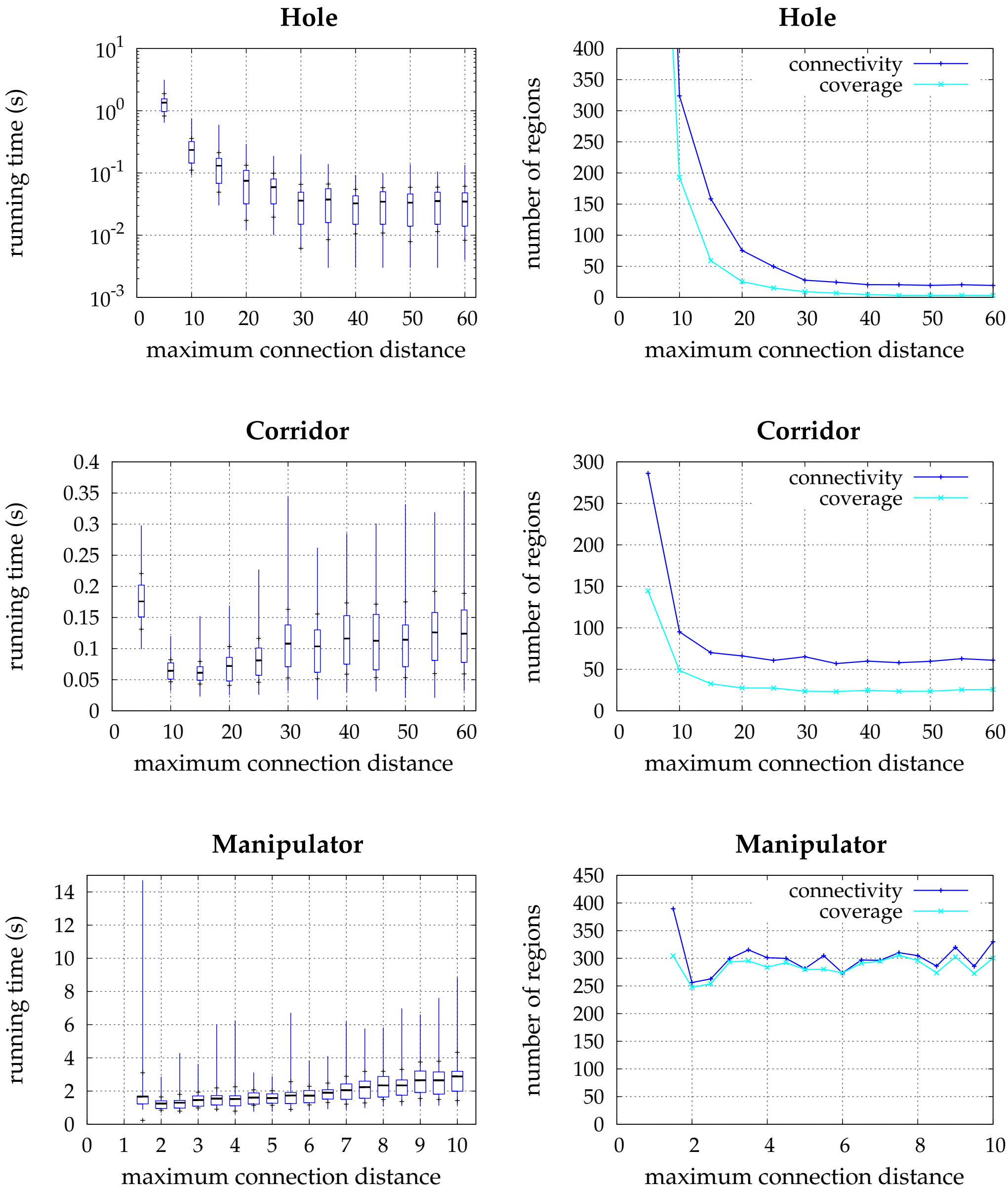


Figure 3.6 Influence of the *maximum connection distance* on the running time and number of regions required to get the free space covered and maximally connected in the 3D environments.

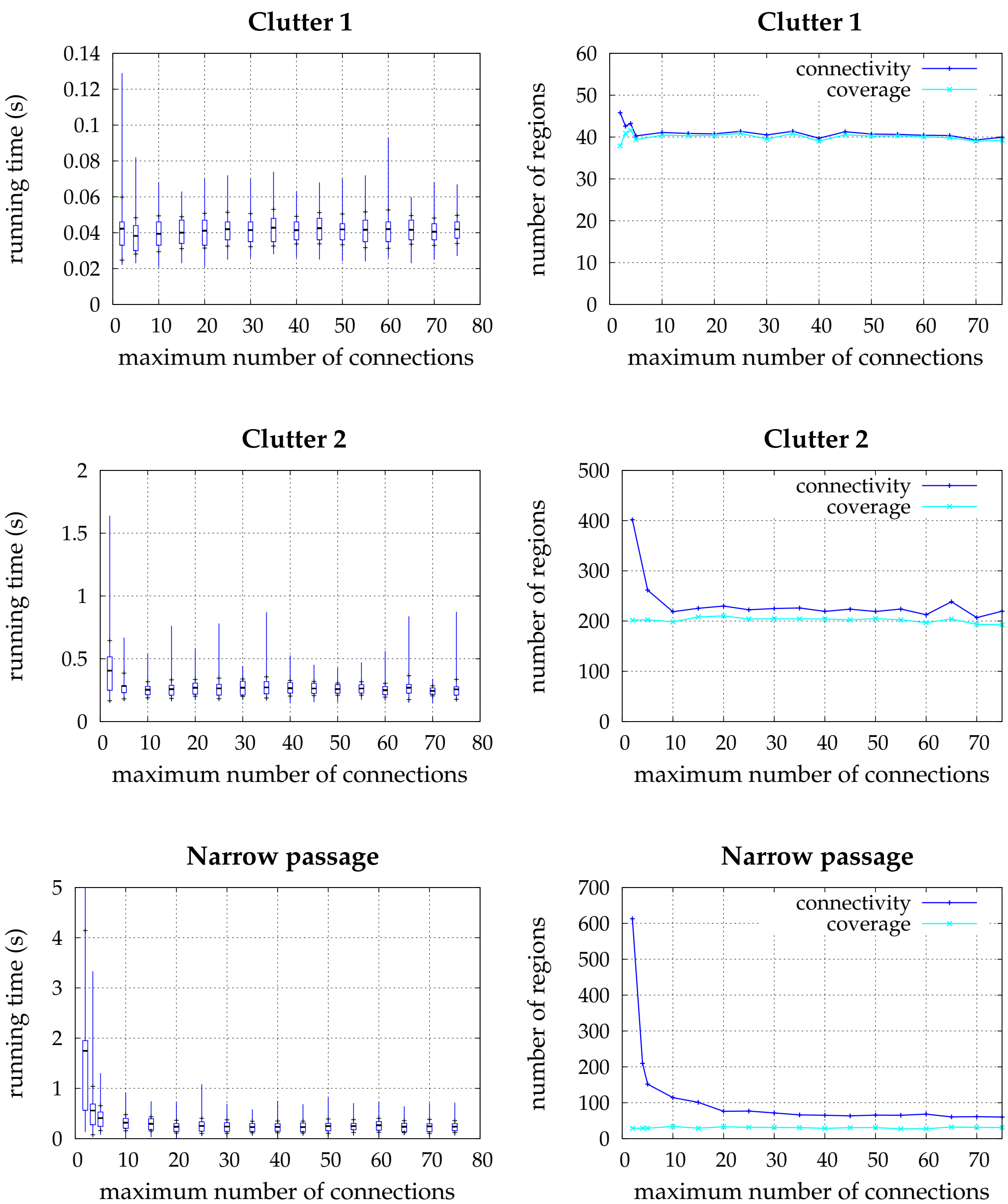


Figure 3.7 Influence of the *maximum number of connections* on the running time and number of regions in the 2D environments.

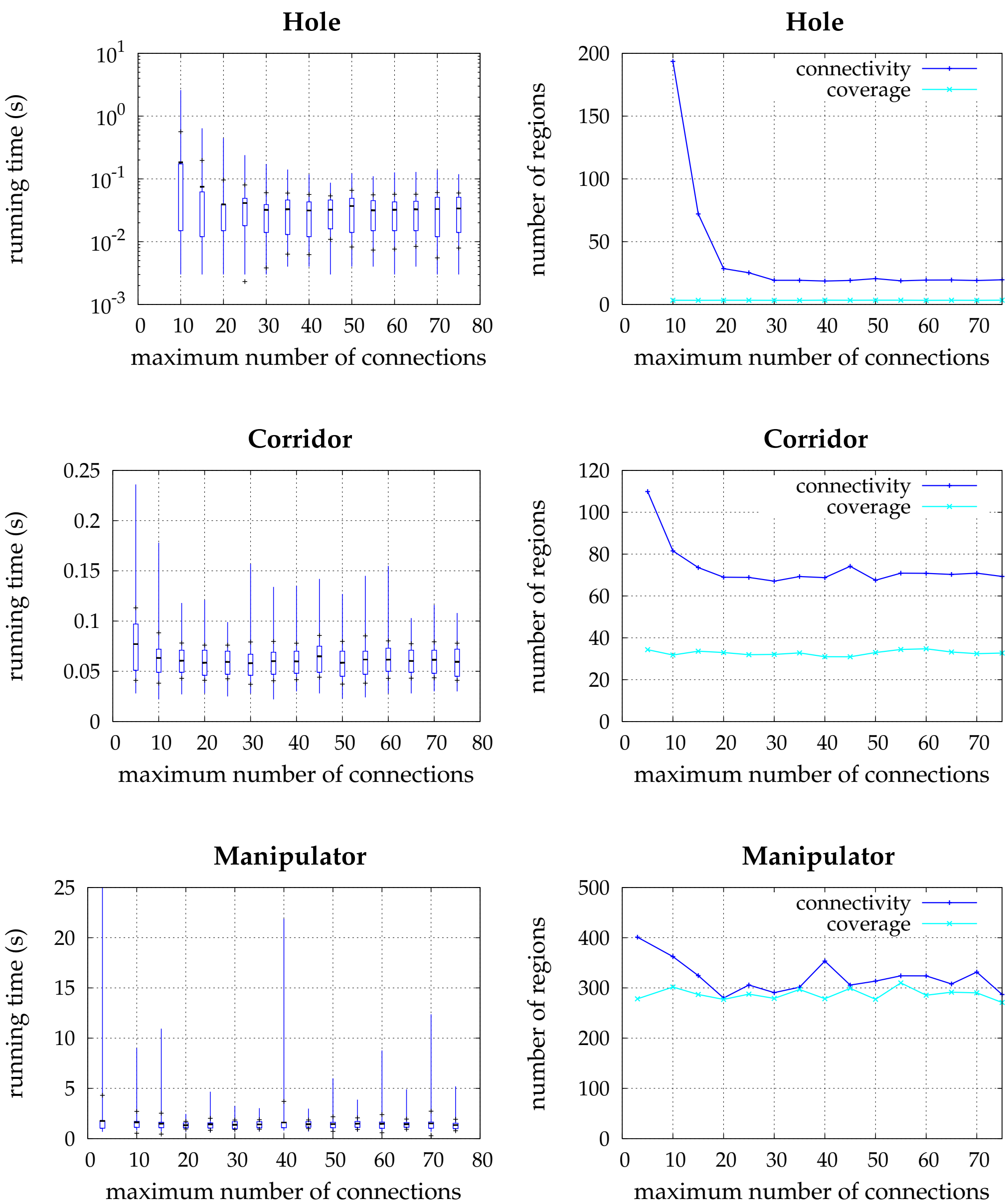


Figure 3.8 Influence of the *maximum number of connections* on the running time and number of regions in the 3D environments.

	Neighbor selection parameters	
	max. connection distance	**max. number of connections**
Clutter 1	15	75
Clutter 2	10	75
Narrow passage	30	75
Hole	40	75
Corridor	15	75
Manipulator	2	75

Table 3.4 The optimal values used in the neighbor selection strategy.

regions needed to get $\mathcal{C}_{\text{free}}$ covered and maximally connected remains constant at a certain point while the running time does increase. However, as a large maximum number of connections (e.g. 75) does not affect the running time much, we recommend to use such a high value. This can be explained as follows. First, we observe that setting the maximum number of connections larger than the expected number of regions (that satisfies the coverage and connectivity criteria) has no effect on the running time. Consider for example the Clutter 1 environment. As it uses a random sampling strategy, the expected number of samples to which a particular sample can be connected is the connection area, divided by the area of the free space, multiplied by the average number of regions that corresponds to satisfying both criteria: the expected number of samples equals $\pi * 15^2/40^2 * 40 = 17.7$. This means, if the samples are uniformly random spread in the space and a maximum connection distance of 15 is used, that the expected number of samples to which a connection is tried (per sample) is 17.7. Setting the maximum connection distance higher than this value will have little effect on the running time. Setting the parameter slightly smaller than this value will also not have much effect on the running time as the chances are high that the discarded samples do already belong to the same connected component.

3.5 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. They can be classified into three categories: uniform, non-uniform and hybrid techniques.

The first category 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, see Section 2.6. The second category tackles the narrow passage problem by biasing the sampling distribution. That is, more samples are added in 'difficult' regions of the environment. A region is difficult if the size of the region (which corresponds to a particular sample) is small compared to the total free space. The number of samples that are generated within these regions can be increased by filtering out samples that probably do not contribute to the coverage and maximal connectivity of the roadmap. Examples include Gaussian and obstacle-based sampling. The third category combines the strengths of the previous two categories. The bridge test for example concentrates samples in difficult areas but it also generates some samples in open areas. Several combinations of existing sampling strategies are suited to serve as a hybrid technique, see the paper of Hsu *et al.* [73] for an elaboration.

Experiments

For each category we choose a representative method. For the uniform technique we choose *Halton**, for non-uniform *Gaussian* and for hybrid we choose *Bridge test*. See Section 2.6 for more information on these techniques. We will compare their behavior by considering the experiments we performed on the six environments. These environments are representative for many different motion planning problems so we expect the observations to apply rather general.

Figure 3.9 shows the results for the 2D environments and Figure 3.9 shows the results for the 3D environments.

Halton* sampling resulted in relatively low running times for all environments, except for the Narrow passage environment. This is consistent with the results from the previous chapter. Halton's uniform distribution created too many samples in the two wide open areas and too few in the narrow passage. Since the chance is small of obtaining a set of samples that covers the space in the narrow passage , the total number of regions required for coverage and maximal connectivity was higher for this method than for the other two methods.

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. In addition, the ample free space was covered fewer times (due to

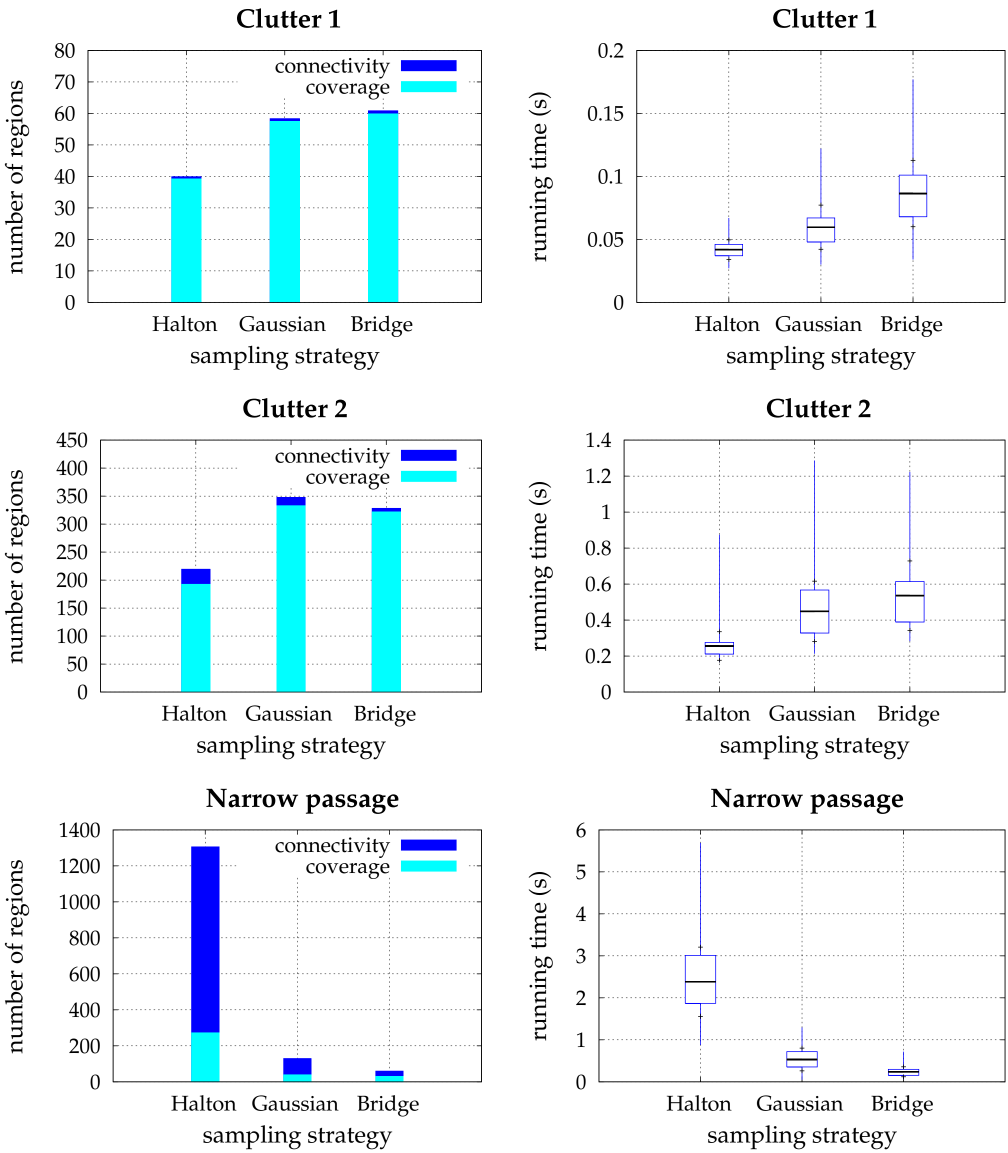


Figure 3.9 Sampling statistics for the 2D environments. The charts in the left column show the average number of regions required to cover and connect the free space. In the Narrow passage environment for example, Halton sampling required 275 regions to cover the free space and 1300 regions to connect the free space. The right column shows box plots corresponding to the time needed to satisfy both criteria.

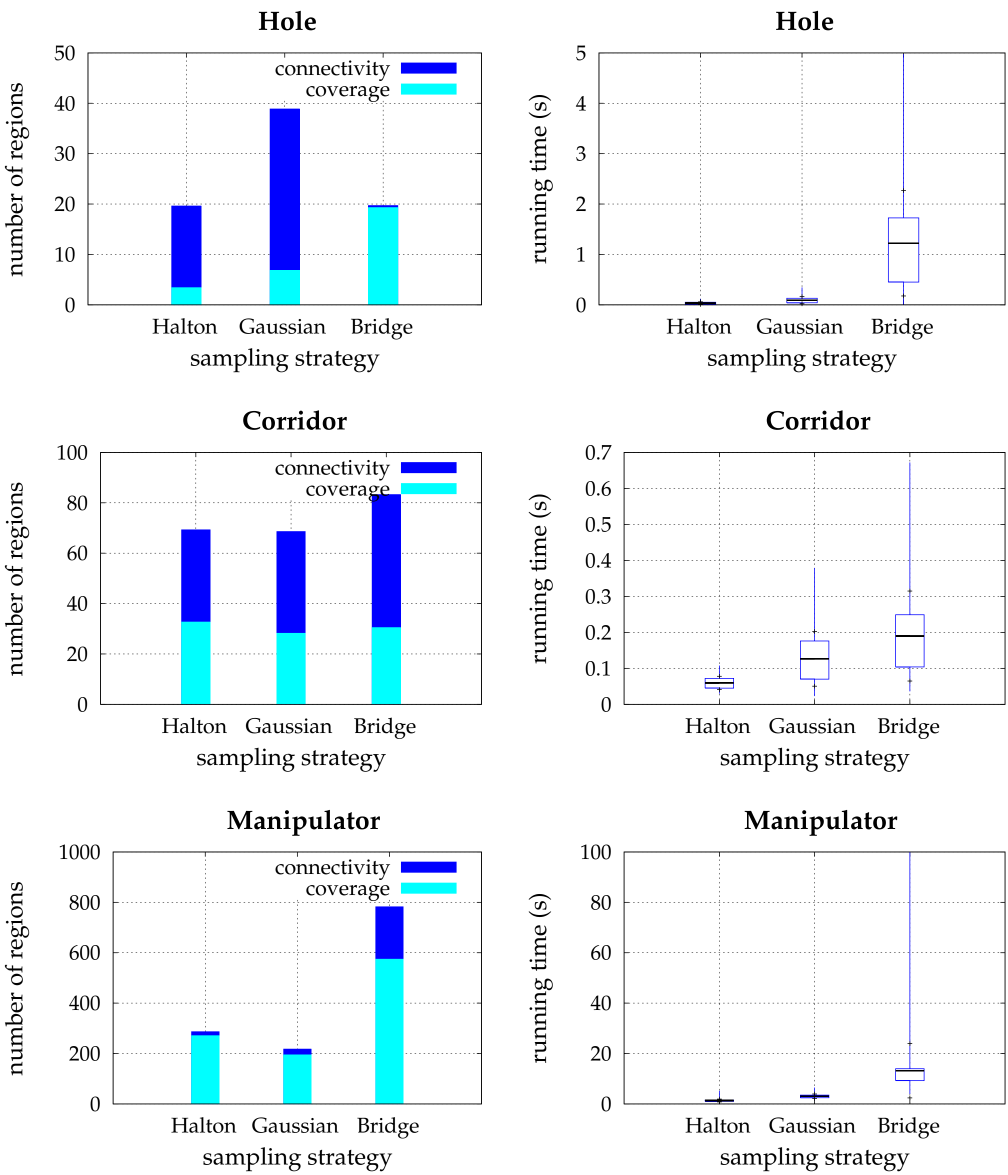


Figure 3.10 Sampling statistics for the 3D environments.

the Gaussian distribution of the samples). As this distribution generates fewer samples in ample free space and more near obstacles, it can be difficult to connect them. This explains why connecting $\mathcal{C}_{\text{free}}$ involved more than three times as many regions compared to covering $\mathcal{C}_{\text{free}}$.

The Bridge technique has been designed to combine the strengths of the previous two categories by concentrating samples in difficult areas while some samples are also generated in open areas. This resulted in the lowest average running time and the lowest number of regions needed to get $\mathcal{C}_{\text{free}}$ connected in the Narrow passage environment. However, the technique performed moderately on the other environments. The charts show that it had difficulties in getting $\mathcal{C}_{\text{free}}$ covered. This can be explained by looking at the properties of these environments. As they have no narrow passages, a technique that is designed to create samples in narrow passages will spend time uselessly. In the Hole environment, many Bridge samples were created near the walls while most of them did not increase the coverage of $\mathcal{C}_{\text{free}}$. The same argument holds for the Manipulator environment.

By looking at the charts of Figure 3.9 and Figure 3.10, we can make an important observation. For the Clutter 1 and Clutter 2 environments, the difference between the moment that $\mathcal{C}_{\text{free}}$ was covered and the moment that $\mathcal{C}_{\text{free}}$ was maximally connected is very small. In contrast, for the Narrow passage environment this difference was much larger. Hence, 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 clarify this, we considered two versions of the Manipulator environment. The first variant is the one depicted in Figure 3.3(f). The second variant is 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. In the first variant, the Halton sampling strategy needed about 271 samples to cover the space. Only 6% more samples were needed to connect the space. While the second variant needed the same number of samples to cover the space, connecting the space required five times as many samples. Hence, connecting samples near or in narrow passages is more difficult. The main observation that can be made is that the narrow passage problem is not so much caused by coverage but by connecting the nodes. Rather than concentrating on more clever sampling, it may be beneficial to spend more effort on connecting nodes in difficult regions. Actually, already one of the first papers on PRM did this by trying to connect difficult nodes in a second phase using a bouncing strategy [124]. In Section 3.6, 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 few samples that covers and connects $\mathcal{C}_{\text{free}}$. The smaller the number of samples, the less time is needed to connect those samples which is the most time-consuming step in the PRM. However, some overlap between the regions that belong to the samples is required because this simplifies creating connections between them. This 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 [121]. Section 6.5 will show that this technique has difficulties satisfying the coverage and maximal connectivity criteria. The approach is too strict and should probably be combined with other sampling techniques.

In Chapter 6, we propose a new technique that creates small graphs satisfying both criteria.

3.6 Local planners

In the previous sections we showed that it can be 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 time-consuming, a careful trade-off 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 [99]. 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 [124].

It is implemented as follows. The planner tries to make small steps on the straight line toward the goal, as does the SLLP.[3] This assures that the region reached by the PFLP subsumes the region reached by the SLLP. When the robot walks into an obstacle, the planner checks a step from the last collision-free configuration in several directions on the hemisphere oriented toward the goal. The most promising step is considered first. A local minimum is easily detected when all possible steps fail in which case the local planner stops and reports failure.

Although 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.

Figure 3.11 shows the Clutter 2 and Narrow passage environments for each of which a reachability region is drawn. The left pictures show the area that can be reached by the SLLP from a particular sample. The right pictures correspond to the PFLP. While reasonably long connections can be made by the SLLP, the reachability region of the PFLP significantly extends the area to which connections can be made. Besides the advantage of covering larger regions, the PFLP, in contrast 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.

We expect that the PFLP outperforms the SLLP in all environments except the Corridor environment, because the reachability regions will be larger than those created with the SLLP. The PFLP will be able to easily connect the two ample free spaces in the Narrow passage and Hole environments. By contrast, in the Corridor environment, the PFLP may only have an advantage in the hairpins; much ineffectual work might be done elsewhere before it is concluded that no connections exist. This is expected to have a negative effect on the running time.

[3]However, the SLLP uses binary collision checking while the PFLP uses incremental collision checking, see Section 2.4.

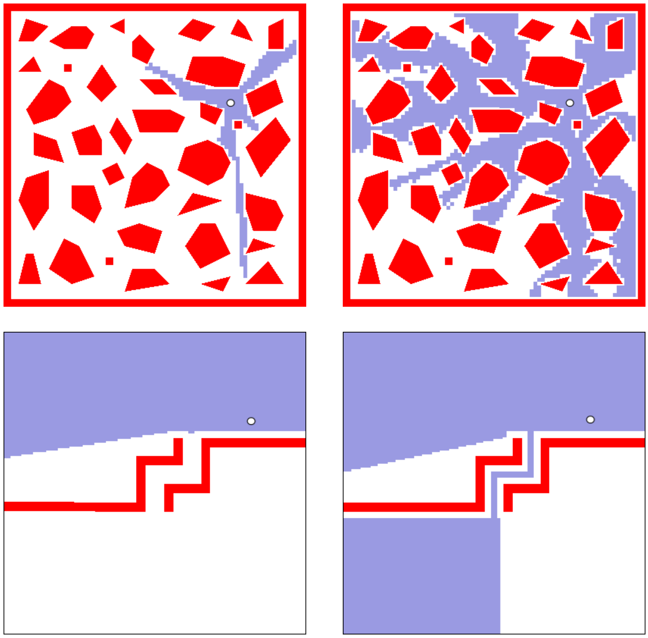


Figure 3.11 Reachability regions for straight-line (left column) and potential field local planner (right column) in the Clutter 2 environment (top row) as well as the Narrow passage environment (bottom row).

Experiments

We used the optimal sampling strategy in all experiments.[4] That is, we used Halton* for each environment except the Narrow passage environment where we used Bridge sampling. As neighbor selection strategy we used nearest-k, where k was set to 75.

In the first experiment we study the effect of using the SLLP or PFLP on the average size of the regions. We set the maximum connection distance to infinity to reveal the full potential of the planners. Figure 3.12 shows the results. In all cases, the PFLP covered larger regions than the SLLP. For the Clutter 1 environment, the PFLP created regions that were on average 86% of $\mathcal{C}_{\text{free}}$ compared to 27% for the SLLP. In the Clutter 2 environment, the regions of the PFLP were even ten times as large as the regions of the SLLP. Even in the Corridor environment, the differences were large.

[4]Unfortunately, we cannot conduct experiments with the Manipulator environment as our implementation of the PFLP currently does not support rotational DOFs.

Next, we conduct experiments to find out whether using a PFLP (which creates larger regions, but takes more time per region) outperforms a SLLP (which creates smaller regions, but takes less time per region). To make a fair comparison possible, we use for both planners the optimal choices. The SLLP uses the neighbor selection parameters from Table 3.4. The maximum connection distances for the PFLP are set to $\{\infty, \infty, 30, \infty, 20\}$ for the different environments. The PFLP generally uses larger (optimal) maximum connection distances as the regions will be larger.

Figure 3.13 shows the average number of regions and Figure 3.14 shows the running times for the two local planners. The results show, as expected, that the number of regions needed to cover the space are considerably lower for the PFLP, which is caused by the larger average size of the regions. In addition, the maximal connectivity criterion was satisfied by considerably less regions. The (absolute) difference between the number of nodes to achieve coverage and number of nodes to achieve maximal connectivity was much smaller for the PFLP than for the SLLP. The PFLP clearly outperformed the SLLP in all environments, except in the Corridor environment. But also for this environment, the difference of the average running time is small. Thus, 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. The 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 in general is less dramatic than the improvement in number of regions required. However, we believe that one could improve the time even further.

In future work, one could investigate techniques that can identify difficult regions in the space. This information can be used to select the most appropriate local planner which should lead to an even better performance of the PRM.

3.7 Discussion

While theoretically, the complexity of the motion planning problem is exponential in the number of degrees of freedom, sampling-based planners can successfully handle this curse of dimensionality in practice because they are

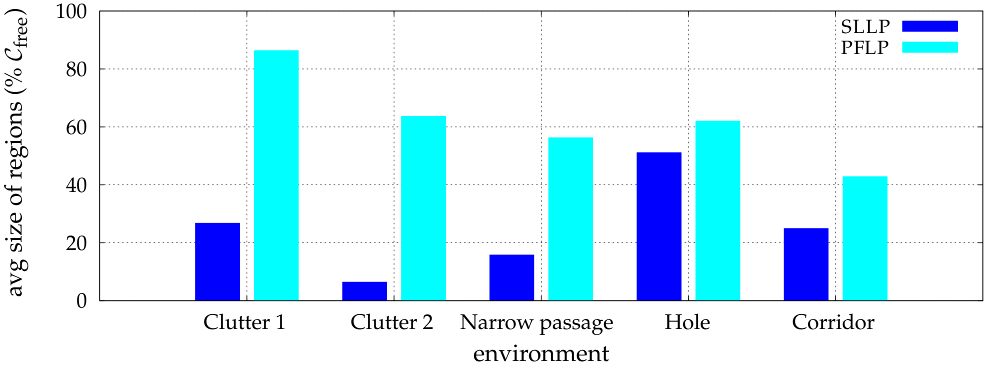


Figure 3.12 The average size of the regions corresponding to the two local planners. The maximum connection distance has been to set to infinity.

reachability-based. 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. The strategy must also generate some samples in large open areas. Another strategy to get the free configuration space faster connected is to use a more powerful local planner. We presented a potential field local planner that creates larger reachability regions and accordingly eases making connections. This planner is also 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.

In Chapter 6, we will use a reachability analysis of $\mathcal{C}_{\text{free}}$ to obtain a more clever sampling and connecting strategy. This results in very small roadmaps that satisfy the coverage as well as the maximal connectivity criteria.

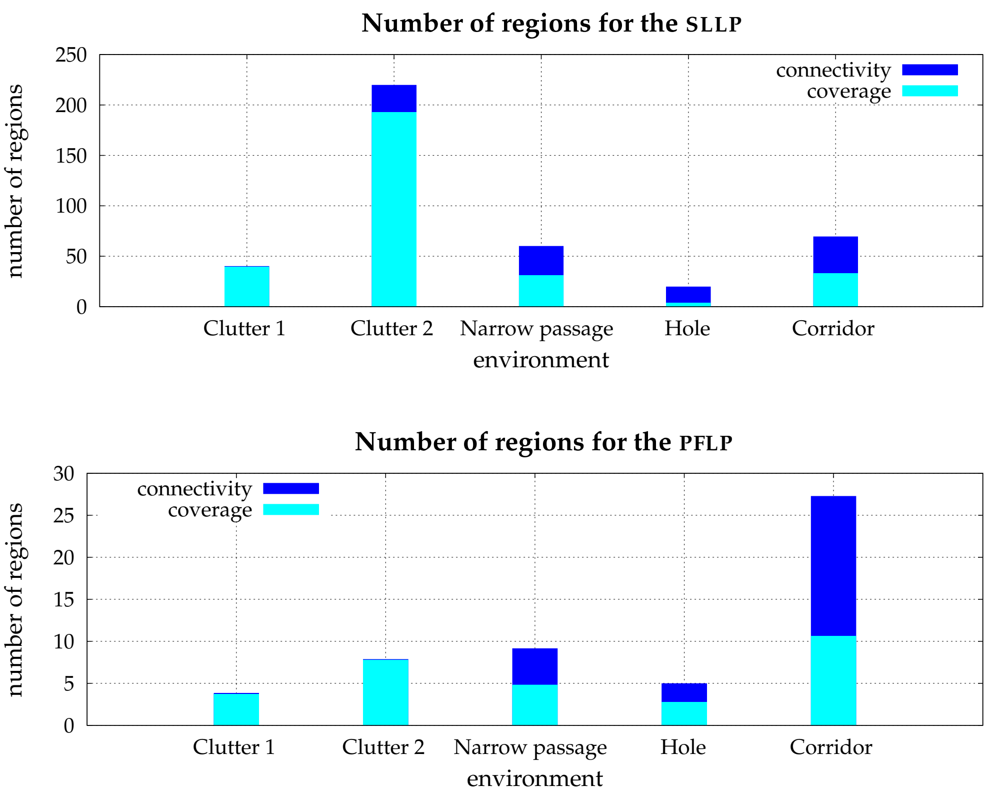


Figure 3.13 Influence of the two local planners on the number of regions. Note the different scales.

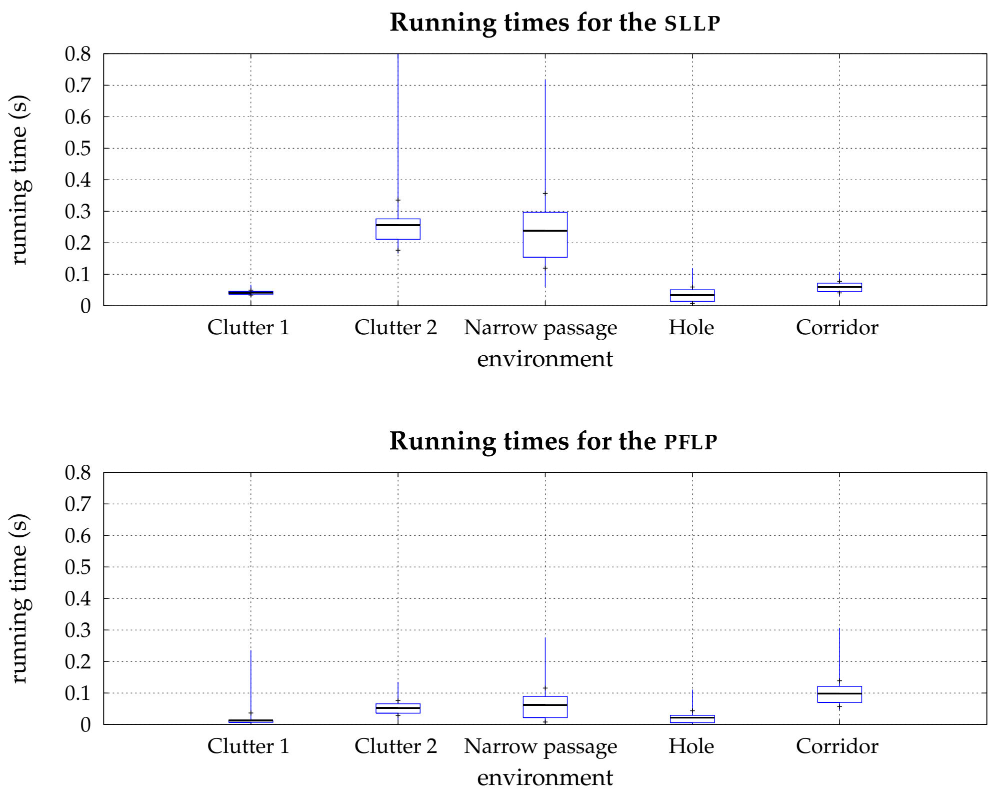


Figure 3.14 Influence of the two local planners on the running time.