Robotics: Science and Systems 2008
Zurich, CH, June 25-28, 2008

Structural Improvement Filtering Strategy for PRM

Roger Pearce[†], Marco Morales[‡], Nancy M. Amato[†]
[†]Parasol Laboratory, Department of Computer Science
Texas A&M University, College Station, Texas, 77843-3112, USA
[‡]Department of Digital Systems, Instituto Tecnológico Autónomo de México
Rio Hondo 1, Tizapán, México D.F., 01080, México
{rpearce,amato}@cs.tamu.edu
marco.morales@itam.mx

***Abstract*—**

Sampling based motion planning methods have been highly successful in solving many high degree of freedom motion planning problems arising in diverse application domains such as traditional robotics, computer-aided design, and computational biology and chemistry. Recent work in metrics for sampling based planners provide tools to analyze the model building process at three levels of detail: sample level, region level, and global level. These tools are useful for comparing the evolution of sampling methods, and have shown promise to improve the process altogether [15], [17], [24].

Here, we introduce a filtering strategy for the Probabilistic Roadmap Methods (PRM) with the aim to improve roadmap construction performance by selecting only the samples that are likely to produce *roadmap structure improvement*. By measuring a new sample's maximum potential structural improvement with respect to the current roadmap, we can choose to only accept samples that have an adequate potential for improvement. We show how this approach can improve the standard PRM framework in a variety of motion planning situations using popular sampling techniques.

I. Introduction

The general *motion planning* problem consists of finding a valid path for an object from a start configuration to a goal configuration. In the traditional application of robotics, a valid path is defined as a collision-free path. Many of the techniques originally designed for robotics have been extended to other applications such as the study of protein folding in Biology and Chemistry [3], [5], [21]–[23], virtual prototyping in manufacturing and mechanical design [4], [8], and the simulation of characters for animation and games [13], [14].

Unfortunately, exact or complete motion planners are intractable for most practical problems because the complexity grows exponentially with the problem's dimensionality [20]. This led researchers to explore sampling based methods and create incomplete approximate solutions. One popular sampling-based method is the Probabilistic Roadmap Method (PRM) [11] which randomly builds a *roadmap* representation of the planning space. Many heuristics have been added to the PRM framework [1], [6], [9], with the overall goal to increase the distribution of samples in regions of the space that model highly constrained robot motions. As a result, there are many planners to choose from and it is not always clear how to choose among them.

Dynamically adapting the planning strategy to features discovered in each problem [7], [10], [16], [18], [24] has been successful in addressing the shortcomings of a particular sampling strategy. However these methods need metrics that gather relevant information about the planning process in order to make effective decisions to adapt the planning strategy.

Another area of research focuses on creating minimal roadmaps. VisPRM [19] is a strategy that achieves minimal roadmaps by accepting only the necessary samples; samples are accepted only if they improve the structure of the roadmap, and this criterion is tested in brute-force manner.

In our previous work, we have developed online metrics [15], [17] to monitor the sampling process to estimate the performance of the model building process at three levels of detail: at the sample level, how "good" and efficient are the samples created? At a small region level, how well are the small locally-homogeneous regions in the large high-dimensional space being explored? At a global level, how well is the "global view" of the high-dimensional space being modeled? Unfortunately, these questions cannot be directly answered because there is not a perfect solution to compare with. These questions are answered indirectly by measuring the relative performance change over small time intervals for each of the three levels of detail. The overall goal of this work is to use the online performance metrics in creating adaptive sampling algorithms to solve more complex problems.

In this paper we develop a metric and filtering strategy to estimate the potential *roadmap structural improvement* of a new sample. This estimate of improvement can be used with a threshold to bias the sampling process towards new samples that have a higher potential for structural improvement. We show that this addition to the PRM framework can increase the speed of roadmap convergence but may come at the price of roadmap quality.

The remainder of this paper is organized as follows: Section II introduces previous work in filtering strategies for PRM; Section III covers how we define and calculate *Roadmap*

This research supported in part by NSF Grants EIA-0103742, ACR-0081510, ACR-0113971, CCR-0113974, ACI-0326350, CRI-0551685, by the DOE, IBM, and Intel. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Pearce supported in part by a Dept. of Education Graduate Fellowship (GAANN).

Structure Improvement; Section IV describes the experimental analysis of this method, and compares the speed of convergence and roadmap quality of the different strategies; Section V discusses our final conclusions.

II. PRM Filtering Strategies

The basic PRM framework [11] builds a C-Space *roadmap* model in two main steps: node generation and node connection. Node generation consists of randomly sampling configurations, testing them, and keeping the valid ones as roadmap vertices. For the valid samples kept in the roadmap, a neighborhood of potential edge candidate samples in the roadmap is gathered using some simple heuristic (e.g. k-closest neighbors). Edges between the sample and its neighborhood are tested with a local planner, and the valid transitions are kept as roadmap edges. The resulting roadmap can be queried as many times as needed.

To improve the performance of roadmap construction, many filtering techniques have been proposed. The general aim of this filtering is to reduce sampling and local planning in areas that are easily mapped (oversampling) while biasing the exploration to the difficult regions of C-Space. At the node level, sometimes called node generation, filtering is done to bias sampling to difficult regions of the C-Space; Gauss-PRM [6], Bridge-Test [9], and OBPRM [1] are examples of node-level filtering. Similarly, edge pairs can be filtered in different ways from k-closest or neighborhood-radius to more advanced methods like VisPRM [12]. Table I briefly summarizes the PRM filtering techniques mentioned in this work.

TABLE I

Roadmap-Based Planners Studied

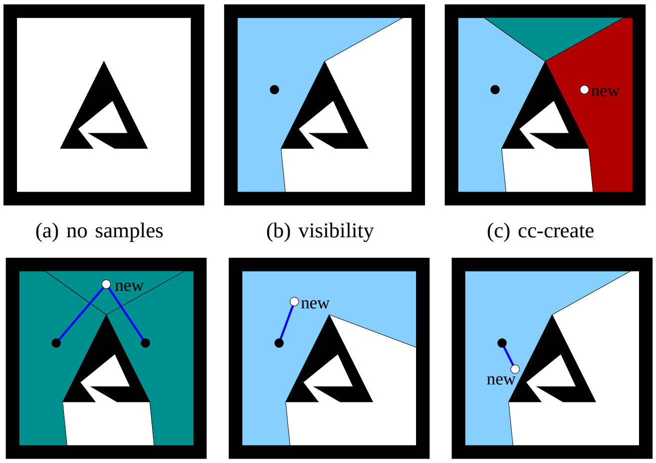
Method	**Node-level Filtering**	Edge-level Filtering
Basic-PRM [11]	only basic CD filter	k-closest neighborhood
Gauss-PRM [6]	after CD check	k-closest neighborhood
Bridge-Test [9]	after CD check	k-closest neighborhood
OBPRM [1]	only basic CD filter	k-closest neighborhood
VisPRM [12]	after all edge checks	all nodes not in current connected component
Structural Improvement Filtering	based on roadmap structure, before edge checks	based on roadmap structure, before edge checks

Most of the filtering strategies shown in table I filter based on information gathered during collision detection tests. For example, in Gauss-PRM and Bridge-Test many samples may be discarded based on the local region of C-Space explored during collision detection tests (whether in open free-space, or deep inside obstacle-space); the only samples that remain are on the boundaries of C-Obstacles. In VisPRM many samples (along with its fully computed edges) are discarded if the sample and its edges fail to either merge two connected components or to create a new one.

In this work we are proposing a new filtering strategy based only on the current roadmap and an estimate of the structural improvement. In this way filtering is done before expensive edge local planning.

III. Roadmap Structure Improvement

In our previous work, we monitored the evolution of the PRM process and classify each new sample. These classifications have been used to compare the quality of samples generated by different sampling strategies. As a quick summary, each new sample X is classified as follows: (See [15], [17] for more details.)



(a) no samples (b) visibility (c) cc-create

(d) cc-merge (e) cc-expand (f) cc-oversample

Fig. 1. Classification of new nodes when modeling the C-Space of a point robot moving in the plane shown in (a). (b) The first sample in the model with its visibility region. (c) A new sample lying outside the visibility region of any other sample creates another component with its own visibility region. (d) A new sample lying in the overlap of the visibility region of two components allows to merge them. (e) A new sample lying inside the visibility region of one component expanding its visibility: cc-expand. (f) A new sample lying inside the visibility region of one component without changing its visibility: cc-oversample.

- cc-create — A new component CC with X as its only node is created as seen in Figure 1(c). The coverage of the roadmap increases by the coverage of X and the connectivity and topology improve due to the new component.
- cc-merge — X merges two or more existing components in the roadmap as seen in Figure 1(d). The coverage, connectivity, and topology improve due to the new pathways found.
- cc-expand — X expands an existing component in the roadmap as seen in Figure 1(e). The coverage and topology improve due to the new pathways found.
- cc-oversample — X fails to expand the coverage of a component in the roadmap as seen in Figure 1(f). The coverage and connectivity remain constant.

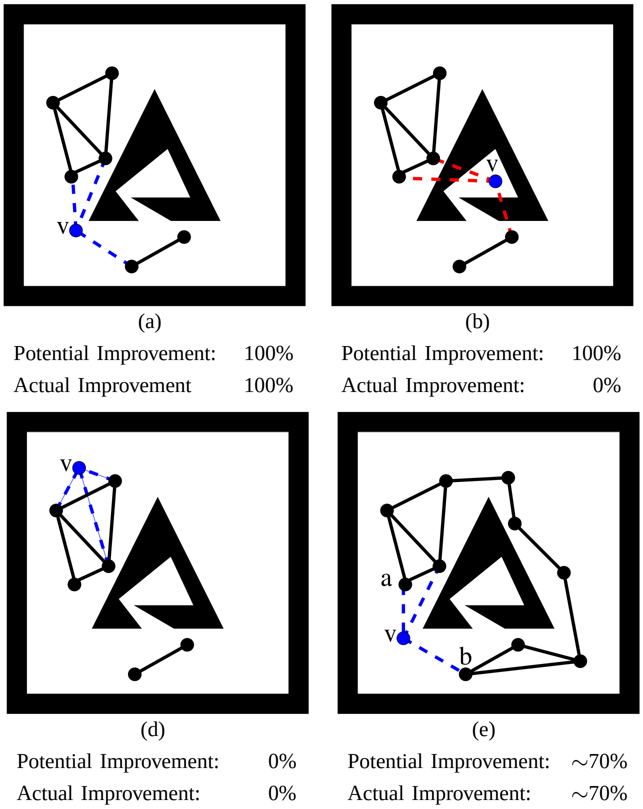
VisPRM [19] is an aggressive strategy to eliminate *cc-expand* and *cc-oversample* nodes. It bypasses the neighborhood information for a brute-force all-pairs method to classify the sample type. In this paper we will make comparisons to this work as a case where only *cc-create* or *cc-merge* nodes are accepted.

The remainder of this section discusses how we define structural improvement, estimate the potential structural improvement of a sample, and design acceptance polices for the PRM framework.

A. Defining Structural Improvement

In this work we filter the sampling process and bias the samples towards areas that improve the structure of the roadmap. We define structural improvement as:

- The addition of pathways previously nonexistent: *cc-merge*
- Finding shorter pathways between existing nodes: some *cc-expand*



(a)
Potential Improvement: 100%
Actual Improvement 100%

(b)
Potential Improvement: 100%
Actual Improvement: 0%

(d)
Potential Improvement: 0%
Actual Improvement: 0%

(e)
Potential Improvement: ∼70%
Actual Improvement: ∼70%

Fig. 2. Cases of Potential Structural Improvement vs. Actual Structural Improvement. Solid edges represent existing roadmap pathways, dashed edges represent potential neighbors of the new sample v.

When the addition of a new sample and its corresponding neighbor edges reduces the distance between any two neighbors there is structural improvement. For each new sample we estimate its maximum potential to create such a structural improvement before the edge pairs are actually checked to improve the quality of roadmap nodes and reduce costs.

Estimating the potential before edge pairs are tested is what differentiates this work from previous work [15], [17] or VISPRM [12]. Previously, expensive local planners for the edge pairs were needed to make reliable classifications or to calculate the improvement of a new sample.

B. Estimating Sample Potential Improvement

Following the PRM framework of sample selection X and neighborhood identification $\{N_1, ..., N_k\}$, we use the existing roadmap model to evaluate the potential structural improvement of a sample X.

Figure 3 illustrates how the potential improvement of a new sample X is computed. First, the existing pairwise pathways between all neighbors $\{N_1, ..., N_k\}$ are evaluated. If it is found that some neighbors belong to different connected components (CC), then X is a potential *cc-merge* node and its potential structural improvement is set to 100%. Each existing graph pathway between N_i and N_j must be evaluated, and a *Single Source Shortest Path* (SSSP) algorithm can be used to obtain the pathway $P_{i,j}$. The weight of this pathway can then be compared to the new potential pathway through X: $P'_{i,j} = N_i \rightarrow X \rightarrow N_j$. The potential improvement of X is the maximum percentage improvement of all $P'_{i,j}$ over the existing $P_{i,j}$. Algorithm 1 details how the maximum potential improvement is calculated.

Algorithm 1 Calculation of potential *structural improvement*

```
Input: The new sample $X$, the existing roadmap $R$, and the sample's neighborhood $\{N_1, ..., N_k\}$
Output: $max\_imp$ – Maximum potential improvement, as a percentage
$max\_imp = 0$
if $\{N_1, ..., N_k\}$ are not in the same $CC$ of $R$ then
    $max\_imp = 100\%$
    RETURN
end if
for every $N_i$ in $\{N_1, ..., N_k\}$ do
    Find SSSP($R$, $N_i$, $\{N_{i+1}, ..., N_k\}$)
    for every $N_j$ in $\{N_{i+1}, ..., N_k\}$ do
        $P_{i,j}$ = SSSP from $N_i$ to $N_j$ through $R$
        $P'_{i,j}$ = distance from $N_i \rightarrow X \rightarrow N_j$
        $improvement$ = % improvement of $P'_{i,j}$ over $P_{i,j}$
        $max\_imp$ = max( $max\_imp$, $improvement$ )
    end for
end for
```

C. Sample Acceptance Policy

Based on a new sample's potential improvement we can make informed decisions about the fate of the sample. By creating sample improvement thresholds we can effectively filter the samples and accept the samples that yield a desired potential improvement.

At one extreme, an improvement threshold of 0% will accept any sample which offers any amount of improvement. This does not accept all samples, because cases arise where all pathways through the sample offer longer pathways than currently exist in the roadmap.

On the other extreme, an improvement threshold of 100% will only accept samples which are potential *cc-merge* nodes. This aggressive threshold comes with quality considerations discussed in Section IV-B.

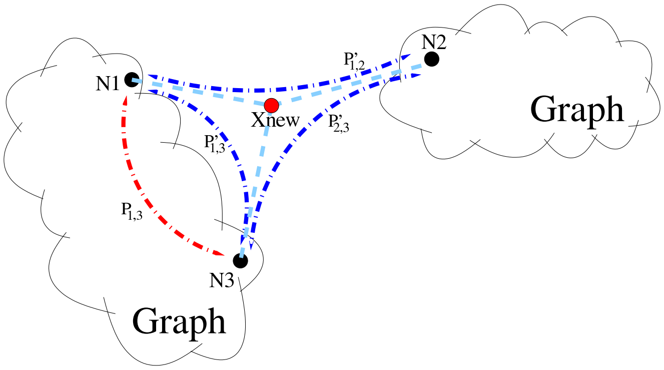


Fig. 3. Illustrates calculating the potential *structural improvement* of sample X. $N1$, $N2$, $N3$ are neighbors of X. $P_{i,j}$ represents existing roadmap paths between N_i and N_j. $P'_{i,j}$ represents potential new pathways between N_i and N_j through X.

It is important to note that delaying the *sample acceptance policy* for an initial time period is necessary. During the initial phase of roadmap building the model is in a "Quick Learning" stage [15], [17]. Here, estimations and classifications of samples are not accurate due to the primitive knowledge in the model. It is not until the majority of the planning space is covered (but not necessarily connected) that the sample estimations and classifications become accurate. In this work the initial window is set to 20 samples; we plan to automate this in future work.

IV. EXPERIMENTAL ANALYSIS

In this study we show the effect of both *Structural Improvement Filtering* in PRM and VISPRM as two different *sample acceptance policies*. Our experimental setup, described in detail below, shows the effect of aggressive policies that accept few samples and how it leads to cheaper roadmap construction at the cost of roadmap quality.

A. Experimental Setup

1) Motion Planning Environments: Throughout this paper we study instances of the motion planning problem with different valid and invalid densities, and with a mixture of open spaces and narrow passages. The instances discussed are:

- The *rigid-walls* problem, Fig. 4 (left), has a 6-DOF rigid-body box robot that should pass through the small openings (slightly larger than the robot) in the walls that divide the environment into five chambers from one side to the other side. Three of the chambers are cluttered with small cube-shaped obstacles. This problem has a C-Space that is similar to its workspace, with four narrow passages and open and cluttered spaces in between.
- The *rigid-maze* problem, Fig. 4 (center), has a 6-DOF rigid-body robot that should pass through a series of tunnels with some dead-ends from the top to the bottom. This problem is interesting because its C-Space resembles the workspace with two clear free areas, the tunnels form a long and narrow passage with dead ends, and the obstacle occupies the majority of the planning space.
- The *rigid-hook* problem, Fig. 4 (right), has a 6-DOF rigid-body hook robot that should pass through the narrow openings in the two walls that divide the environment into three chambers from one side to the other side of the environment. This is a difficult problem that requires simultaneous translational and rotational motions.

2) Node Generation Strategies: We study the *Node Generation* strategies described in Table II.

TABLE II
ROADMAP-BASED PLANNERS STUDIED

Planner	Sampling Strategy
BASIC-PRM [11]	Uniform, keeping all valid configurations
OBPRM [1]	Generate invalid samples and push them away to get valid samples around obstacles
GAUSS-PRM [6]	Uniform sampling, find valid samples within distance d from invalid samples. Valid samples have a Gaussian distribution around obstacles
BRIDGE-TEST [9]	Uniform sampling, find pairs of invalid samples separated a distance d and keep valid samples between pairs.

3) Neighborhood Selection and Local Planner Strategies: We use simple heuristics for neighborhood selection and local planning. We implemented these strategies as described in [24]: every new sample in the roadmap attempts to connect to the 10-closest nodes already in the roadmap by using the straight-line and rotate-at-s local planners [2].

4) Sample Acceptance Policies: Throughout the experiments in this paper, we will make comparisons between four *sample acceptance policies* (Table III) in the PRM framework.

TABLE III
STYLES OF PRM STUDIED

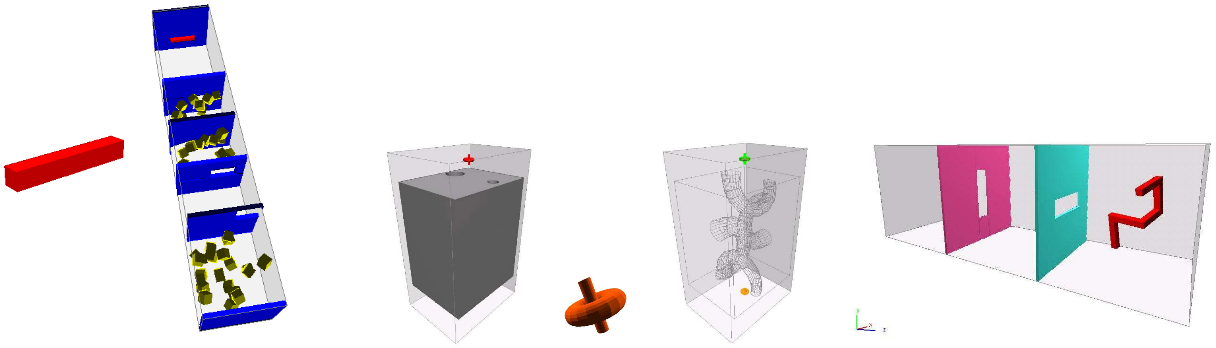
PRM Style	Description
Pure [11]	The original PRM strategy, this will be our standard or baseline comparison.
Imp=50%	This will accept any sample with at least 50% potential improvement.
Imp=100%	This will accept any sample with 100% potential improvement.
VISPRM [12]	Visibility PRM is another style of PRM which aims to create a minimal roadmap.

5) Methods of Comparison: We compare the different acceptance policies in two ways. In a table we average the statistics gathered over 20 random iterations of each set of parameters. Table IV describes the statistics we gathered and averaged for comparison.

We also show the effects of filtering on a single run by plotting the evolution of the diameter of the largest connected component. For these plots we fix the random seed across the different parameter tests to ensure that the different *acceptance policies* are working with the same stream of random samples.

B. Quality Compromises

The decision to discard samples with low potential for improvement comes at the cost of reduced roadmap quality. To



(Walls) (Maze) (Hook)

Fig. 4. (*Walls*) 6-DOFcubic robot, four short passages; (*Maze*) solid view, 6-DOF robot, and wire view; (*Hook*) 6-DOFhook robot, two medium passages.

TABLE V

Average statistics gathered while solving the *Maze* environment

Sampler	Measure	Sample Acceptance Policy			
		Pure	Imp = 50%	Imp = 100%	VisPRM [12]
Basic-PRM [11]	% Samples Accepted	100% †	2.92%	1.60%	0.28% †
	CD-Calls	1,224,784 †	96,522	57,642	1,010,809 †
	Largest CC Dia	328.2 †	327.2	338.5	434.3 †
	Time Struct Imp (sec)	none †	40.32	10.60	none †
	Total Time (sec)	1,207.3 †	154.9	38.1	239.9 †
Bridge-Test [9]	% Samples Accepted	100% †	54.79%	49.87%	19.30% †
	CD-Calls	873,011 †	622,088	622,447	1,100,653
	Largest CC Dia	280.4 †	287.8	245.7	509.2
	Time Struct Imp (sec)	none †	0.19	0.18	none
	Time (sec)	256.3 †	180.2	179.8	317.2
Gauss-PRM [6]	% Samples Accepted	100% †	12.50%	7.64%	1.66% †
	CD-Calls	308,871 †	93,168	82,232	328,388
	Largest CC Dia	321.7 †	325.5	326.5	436.7
	Time Struct Imp (sec)	none †	2.86	2.08	none
	Time (sec)	99.1 †	46.5	40.1	103.9
OBPRM [1]	% Samples Accepted	100% †	82.9%	9.63%	4.41% †
	CD-Calls	720,818 †	619,470	456,050	957,076
	Largest CC Dia	332.9 †	346.8	409.6	917.3
	Time Struct Imp (sec)	none †	10.23	2.69	none
	Time (sec)	219.6 †	214.7	138.5	265.8

† — represents implementations as defined in [1], [6], [9], [11], [12].

TABLE IV

Statistics averaged for comparison

Statistic	**Description**
% Samples Accepted	The ratio of samples accepted by the policy to the total generated.
CD-Calls	The total number of Collision Detections preformed.
Largest CC Dia	The diameter of the largest connected component which represents the vast majority of the connectable roadmap. The distance is measured in the number of resolution ticks.
Time Struct Imp (sec)	The time spent calculating the *estimated structural improvement* described in this paper.
Time (sec)	The total time spent in PRM: Sampling, Local Planning, and Struct Imp. Additional time generating statistics (e.g. Largest CC Dia) is not included.

compare the quality between the different *sample acceptance policies* in PRM, we examine how the largest connected component's diameter evolves over the number of valid-samples evaluated (as shown in figure 5). The *Pure* PRM sample acceptance policy selects every attempt; this serves as our quality baseline. *Pure* PRM provides the best roadmap quality here because the filtering process inevitably removes nodes and edges that would have refined existing paths in the roadmap.

Measuring the largest component's diameter is not an exact measure of roadmap quality, but can be used as an approximation [24]. The diameter is the longest shortest-path in the roadmap, and the quality we consider refers to how efficiently the robot can move between the two extreme points in the roadmap.

In Tables V, VI, and VII, we can evaluate the average Largest CC Diameter. Here we see that the sample acceptance policies *Imp=50%* and *Imp=100%* lead to roadmaps where the diameter is only slightly larger than that of *Pure*, while VisPRM's more aggressive sample acceptance policy significantly reduces the roadmap quality compared with other policies; this can be seen in the significantly larger component diameters.

TABLE VI

Average statistics gathered while solving the *Hook* environment

	Sample Acceptance Policy				
Sampler	Measure	Pure	Imp = 50%	Imp = 100%	VisPRM [12]
Basic-PRM [11]	% Samples Accepted	n/a* †	n/a*	4.72%	0.47% †
	CD-Calls	n/a* †	n/a*	200,635	4,552,916 †
	Largest CC Dia	n/a* †	n/a*	189.0	301.6 †
	Time Struct Imp (sec)	none †	n/a*	846.1	none †
	Total Time (sec)	n/a* †	n/a*	1,937.4	333.4 †
Bridge-Test [9]	% Samples Accepted	100% †	81.8%	71.6%	11.5% †
	CD-Calls	2,438,478 †	2,406,267	1,859,074	2,633,966
	Largest CC Dia	145.9 †	136.4	145.5	249.5
	Time Struct Imp (sec)	none †	1.15	0.84	none
	Time (sec)	343.0 †	341.7	269.3	346.5
Gauss-PRM [6]	% Samples Accepted	100% †	63.9%	43.6%	5.32% †
	CD-Calls	341,562 †	130,773	121,713	1,018,312
	Largest CC Dia	171.8 †	182.4	181.1	364.4
	Time Struct Imp (sec)	none †	14.27	11.43	none
	Time (sec)	317.5 †	208.9	196.4	211.4
OBPRM [1]	% Samples Accepted	100% †	99.1%	83.5%	17.6% †
	CD-Calls	122,466 †	109,750	107,483	330,648
	Largest CC Dia	177.5 †	185.4	180.2	259.4
	Time Struct Imp (sec)	none †	0.65	0.92	none
	Time (sec)	9.66 †	9.39	9.21	20.95

† — represents implementations as defined in [1], [6], [9], [11], [12].
n/a* — the planner was unable to consistently solve the problem.

TABLE VII

Average statistics gathered while solving the *Walls* environment

	Sample Acceptance Policy				
Sampler	Measure	Pure	Imp = 50%	Imp = 100%	VisPRM [12]
Basic-PRM [11]	% Samples Accepted	100% †	34.4%	7.76%	4.03% †
	CD-Calls	813,877 †	83,573	48,669	599,239 †
	Largest CC Dia	476.3 †	527.7	518.2	936.3 †
	Time Struct Imp (sec)	none †	8.53	5.60	none †
	Total Time (sec)	207.6 †	27.59	12.97	40.58 †
Bridge-Test [9]	% Samples Accepted	100% †	89.6%	44.4%	13.7% †
	CD-Calls	395,514 †	324,548	303,785	599,438
	Largest CC Dia	535.7 †	582.2	610.7	1,002.7
	Time Struct Imp (sec)	none †	2.58	3.52	none
	Time (sec)	33.19 †	29.74	27.39	42.23
Gauss-PRM [6]	% Samples Accepted	100% †	88.5%	19.6%	10.8% †
	CD-Calls	358,524 †	202,494	99,039	401,057
	Largest CC Dia	486.4 †	516.5	578.7	1,034.9
	Time Struct Imp (sec)	none †	8.49	7.90	none
	Time (sec)	51.12 †	44.56	21.01	28.27
OBPRM [1]	% Samples Accepted	100% †	95.2%	38.1%	11.4% †
	CD-Calls	399,078 †	181,127	111,142	572,487
	Largest CC Dia	540.2 †	590.0	649.4	1,121.7
	Time Struct Imp (sec)	none †	3.92	3.96	none
	Time (sec)	69.83 †	25.05	14.79	40.07

† — represents implementations as defined in [1], [6], [9], [11], [12].

C. Speed of Improvement

To compare the *speed of improvement* between different *sample acceptance policies* in PRM, we examine how the largest connected component's diameter evolves over the number of Collision Detections (shown in Figure 6). Both a rise and a fall in the component's diameter signify an improvement: a rise corresponds to an addition of connected C-Space, and a fall corresponds to improvements in shorter pathways. We also show when the point at which the query for the environment is solved, although this is a poor metric for comparisons [15], [17], [24]. Additionally, tables of average statistics gathered when solving the queries are shown for: *Maze* in Table V, *Hook* in Table VI, and *Walls* in Table VII.

All sample acceptance policies for a given planner provide some level of improvement over *Pure*; the amount of improvement depends on the type of planner. A combination of node-level filtering and then *structural improvement* filtering can also be advantageous. For the *Maze* environment, Basic-PRM with *Imp=100%* preformed best with Gauss-PRM and *Imp=100%* at a close second. In the *Hook* environment,

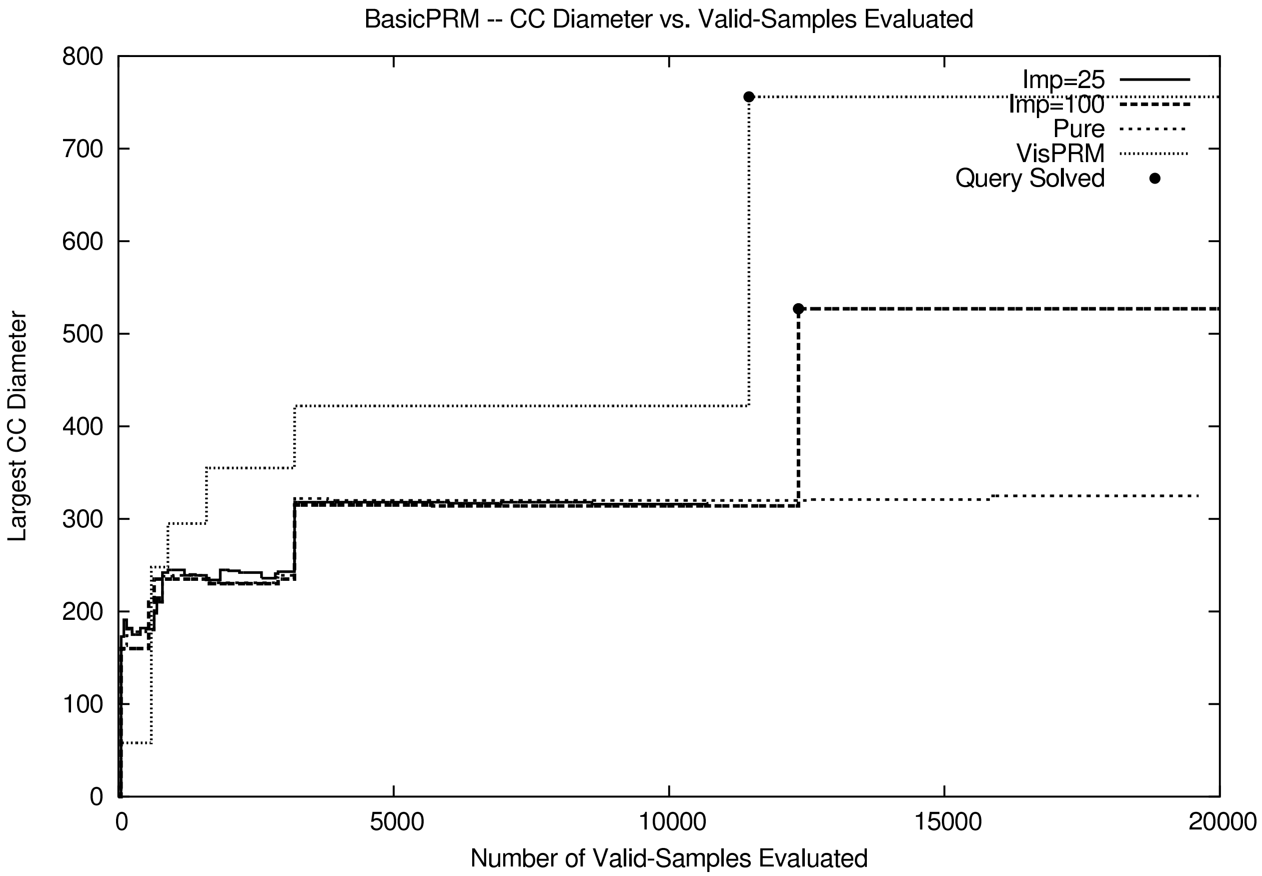


Fig. 5. Maximum roadmap diameter over number of attempts in the *Maze* environment with BASIC-PRM.

OBPRM with *Imp=100%* outperformed the other samplers by a large margin. In the *Walls* environment, BASIC-PRM and OBPRM with *Imp=100%* preformed well.

All of the tests shown here show the power of intelligent sample acceptance policies. In most cases, an intelligent policy combined with node-level filtering can lead to significant improvements. The trade-off in roadmap quality can be significant for an aggressive acceptance policy. Fortunately *Structural Improvement Filtering* allows the policy to be tuned.

V. CONCLUSIONS

In this paper we introduced an addition to the PRM framework that effectively filters samples and edges thereby increasing the percentage of samples which improve the roadmap structure. We have created a metric that can estimate the maximum potential roadmap structural improvement for each new sample and we have used this metric to improve PRM roadmap construction.

We have shown how varying the *sample acceptance policy* can affect both the speed of roadmap improvement and the overall quality of the roadmap. In particular, we have shown that, for all samplers tested, good samples are effectively identified without a drastic sacrifice in roadmap quality. The largest benefit was observed when using uniform samples produced by BASIC-PRM, making it very competitive with the other strategies.

In the future, we plan to explore adaptive ways to control the *sample acceptance policy*. In addition to filtering, we are exploring strategies to search the roadmap and locate areas of potential improvement to guide the sampling process. We also plan to estimate improvement of other criteria, such as path clearance, when filtering samples and edges.

REFERENCES

[1] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective*, pages 155–168, Natick, MA, 1998. A.K. Peters. Proc. Third Workshop on Algorithmic Foundations of Robotics (WAFR), Houston, TX, 1998.

[2] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo. Choosing good distance metrics and local planners for probabilistic roadmap methods. *IEEE Trans. Robot. Automat.*, 16(4):442–447, August 2000.

[3] N. M. Amato and G. Song. Using motion planning to study protein folding pathways. *J. Comput. Biol.*, 9(2):149–168, 2002. Special issue of Int. Conf. Comput. Molecular Biology (RECOMB) 2001.

[4] O. B. Bayazit, G. Song, and N. M. Amato. Enhancing randomized motion planners: Exploring with haptic hints. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 529–536, 2000.

[5] O. B. Bayazit, G. Song, and N. M. Amato. Ligand binding with OBPRM and haptic user input: Enhancing automatic motion planning with virtual touch. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 954–959, 2001. This work was also presented as a poster at *RECOMB 2001*.

[6] V. Boor, M. H. Overmars, and A. F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, volume 2, pages 1018–1023, May 1999.

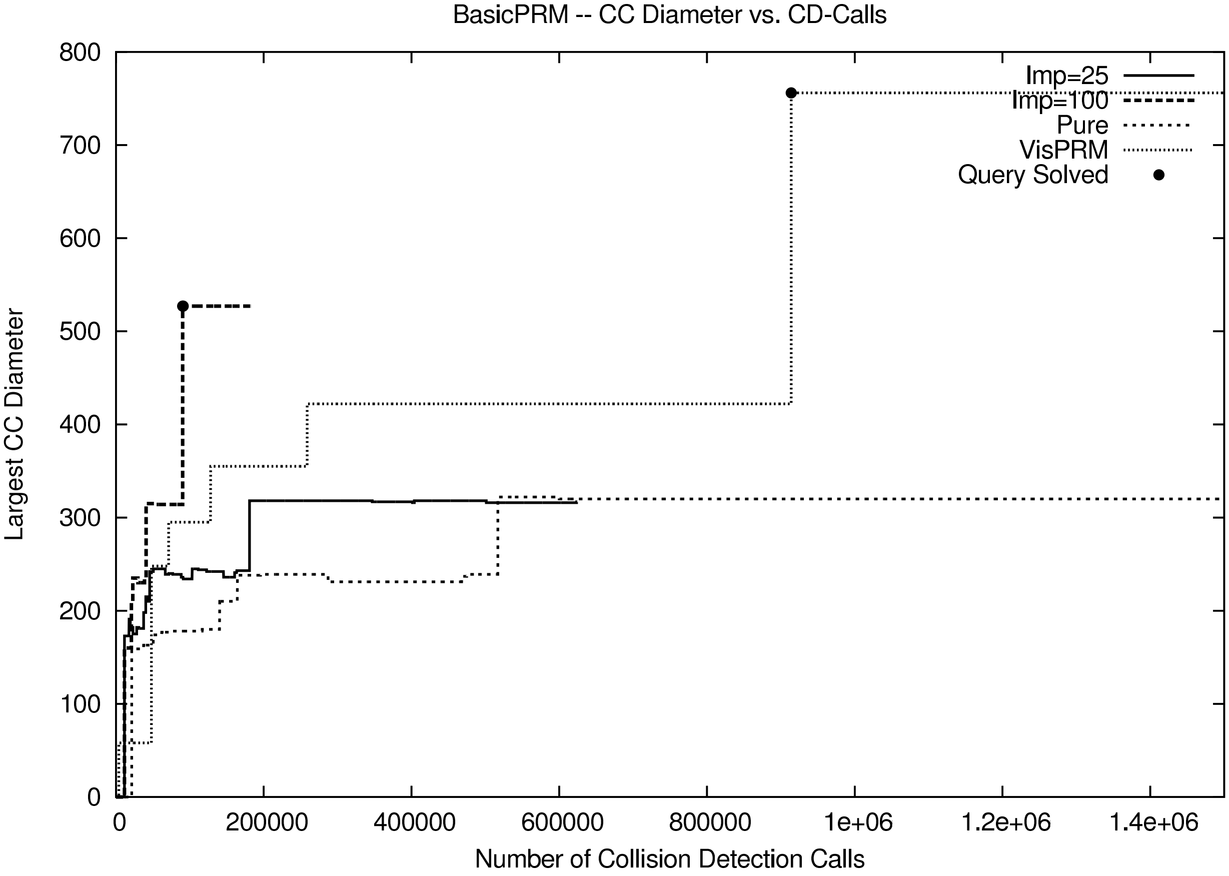


Fig. 6. Maximum roadmap diameter over number of Collision Detections in the *Maze* environment with BASIC-PRM.

[7] Brendan Burns and Oliver Brock. Sampling-based motion planning using predictive models. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2005.

[8] H. Chang and T. Y. Li. Assembly maintainability study with motion planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1012–1019, 1995.

[9] D. Hsu, T. Jiang, J.H. Reif, and Z. Sun. Bridge test for sampling narrow passages with probabilistic roadmap planners. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4420–4426, 2003.

[10] D. Hsu, G. Sánchez-Ante, and Z. Sun. Hybrid PRM sampling with a cost-sensitive adaptive strategy. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3885–3891, 2005.

[11] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, August 1996.

[12] J.-P. Laumond and T. Siméon. Notes on visibility roadmaps and path planning. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2000.

[13] Jyh-Ming Lien, O. B. Bayazit, R.-T. Sowell, S. Rodriguez, and N. M. Amato. Shepherding behaviors. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 4159–4164, April 2004.

[14] Jyh-Ming Lien, Samuel Rodriguez, Jean-Philippe Malric, and Nancy M. Amato. Shepherding behaviors with multiple shepherds. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3413–3418, April 2005.

[15] Marco Morales, Roger Pearce, and Nancy M. Amato. Analysis of the evolution of C-Space models built through incremental exploration. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1029–1034, April 2007.

[16] Marco Morales, Lydia Tapia, Roger Pearce, Samuel Rodriguez, and Nancy M. Amato. A machine learning approach for feature-sensitive motion planning. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 361–376, Utrecht/Zeist, The Netherlands, July 2004.

[17] Marco A. Morales A., Roger Pearce, and Nancy M. Amato. Metrics for analyzing the evolution of C-Space models. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1268–1273, May 2006.

[18] Marco A. Morales A., Lydia Tapia, Roger Pearce, Samuel Rodriguez, and Nancy M. Amato. C-space subdivision and integration in feature-sensitive motion planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3114–3119, April 2005.

[19] C. Nissoux, T. Simeon, and J.-P. Laumond. Visibility based probabilistic roadmaps. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)*, pages 1316–1321, 1999.

[20] J. H. Reif. Complexity of the mover's problem and generalizations. In *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, pages 421–427, San Juan, Puerto Rico, October 1979.

[21] A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to flexible ligand binding. In *7th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, pages 252–261, 1999.

[22] X. Tang, B. Kirkpatrick, S. Thomas, G. Song, and N. M. Amato. Using motion planning to study RNA folding kinetics. In *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, pages 252–261, 2004.

[23] Shawna Thomas, Xinyu Tang, Lydia Tapia, and Nancy M. Amato. Simulating protein motions with rigidity analysis. In *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, pages 394–409, 2006.

[24] Dawen Xie, Marco Morales, Roger Pearce, Shawna Thomas, Jyh-Ming Lien, and Nancy M. Amato. Incremental map generation (IMG). In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, July 2006.