Toward Optimal Configuration Space Sampling

Brendan Burns   Oliver Brock
Laboratory for Perceptual Robotics
University of Massachusetts Amherst

Abstract— **Sampling-based motion planning discovers the implicit connectivity of a configuration space by selecting and connecting sets of configurations. The structure of every configuration space dictates a number of optimal sets of samples whose selection by a sampling-based planner results in a complete roadmap of the space. Though it is generally computationally impractical to develop complete knowledge of configuration space, each individual sample provides information about the configuration space. We propose a new utility-guided sampling strategy that accumulates this information into an approximate model of the configuration space. The model is an approximation of both the state (obstructed or free) of individual configurations and the connectivity of the configuration space. Our proposed sampler uses the approximate configuration space model to select samples that are maximally relevant to the planning task. The relevance of a sample is measured by its expected utility to the further coverage of the configuration space roadmap. The utility metric blends information from both configuration space state and connectivity. The planner incorporates the information obtained from each sample into its approximation and uses these improved models for subsequent sampling. Experimental results with an implementation of this approach to motion planning indicate that it is capable of significantly reducing the runtime required to construct a complete roadmap for configuration spaces with arbitrary degrees of freedom.**

I. Introduction

The efficiency of multi-query, sampling-based motion planning results from its implicit representation of configuration space connectivity. Computing the exact boundary of configuration space obstacles is avoided by using a single sample to capture a large region of configuration space. The performance of a sampling-based motion planner derives from the set of samples it requires to construct a roadmap that represents the complete connectivity of the configuration space. With complete knowledge of configuration space structure it is possible to identify an optimal set of samples whose union of captured regions covers the entire free configuration space. In most cases, the size of such a minimal set is quite small. A sampling-based planner that selects one of these optimal sets constructs a complete roadmap of configuration space with minimal computation. For a simple two dimensional world, this set is illustrated in Figure 1.

Unfortunately, complete knowledge of configuration space structure is generally unavailable to a motion planner and computing this structure is tantamount to solving the original motion-planning problem. Despite the absence of a priori complete knowledge of configuration space structure, we will see that an approximation of this structure can be constructed from a planner's past experience and that this approximation can guide the selection of samples toward the optimal set.
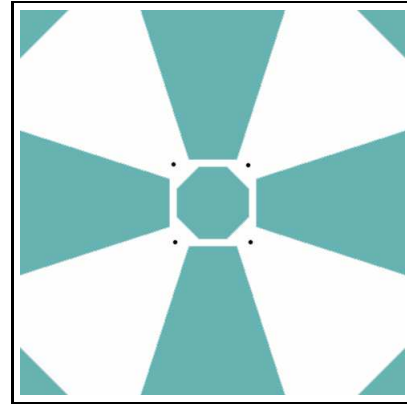

Fig. 1.   A simple configuration space for a point robot and a minimal set of samples which completely represent it

Currently, sampling-based planners use one of two fundamentally different approaches to select configurations. The first, used by probabilistic roadmap (PRM) techniques, uses uniform random sampling [16]. Every sample is generated uniformly at random, independent of previous samples. The second, used by rapidly-expanding random trees (RRT) planners, places samples by wavefront expansion using a Voronoi-bias [17] which progresses the wavefront more quickly toward large unexplored regions.

Each of these strategies is designed around its particular goal. Uniform sampling, designed for multi-query planning, attempts to explore all of configuration space. In contrast, wave front expansion is aimed at single-query planning. It makes the heuristic assumption that regions near the initial and final configuration are more likely to contain a solution path. We propose an alternative to both of these fixed strategies: an adaptive approach that selects configurations by actively examining their relevance to the current state of the motion planner.

Our approach to sampling begins with the insight that each observed configuration provides information about configuration space structure. Neither uniform nor wavefront sampling make use of observed information to select configurations. Because they ignore this information, sampling-based motion planners using such strategies expend computational resources on regions that are either already well represented by previous samples, known to be obstructed from past experience, or less relevant to configuration space connectivity than optimal samples. Using information from past experience to guide sampling significantly improves the relevance of the samples

selected.

Results from active learning [8], [9], [21] show that using past experience to intelligently select future samples can significantly improve the performance of approximate modeling tasks. These results inspire us to apply similar approaches to sampling-based motion planning. In the past we have directly applied active learning to sampling based motion planning [7], however active learning is not a perfect fit to sampling-based motion planning.

The goals of sampling-based motion planning differ from those of machine learning. In the context of configuration spaces, active learning selects samples in an attempt to maximally understand the decision boundary between obstructed and free space. This focus is different from sampling-based motion planning which only wants to maximally understand configuration space connectivity. A detailed understanding of the boundary between obstructed and free space is only important to a sampling-based motion planner in so much as it aids understanding of configuration space connectivity. Further detail is wasted computation. Because of this distinction, traditional methods of active learning are inappropriate for motion planning. To intelligently use past experience to guide future sampling a new method for selecting configurations is necessary. We call this new method *utility-guided sampling*.

This sampler is not interested in optimizing model accuracy, but rather usefulness to motion planning. We identify two types of structure in configuration space that our sampler can approximate from past experience and exploit to select useful samples.

Because there is regularity in the distribution of obstructed and free configuration space an approximate predictive model can be constructed from past experience using statistical modeling techniques. A model, M, built from past experience can predict $P(q = i|M)$, the probability that an unobserved configuration q has the state i.

The planner's current understanding of configuration space connectivity provides a second type of structure. Samples that expand the planner's understanding of this connectivity are obviously more useful to the planner than those that simply confirm what is already known. This structure is used to compute the utility function $U(q = i, R)$ of some configuration q to a roadmap R. Utility-guided sampling brings these two forms of structure together in the formulation of expected utility [3], [25].:

$$U_{exp}(q|M) = \sum_{i \in \{\text{obs,free}\}} P(q = i|M) \cdot U(q = i, R)$$

and selects configurations that maximize this value.

Details of modeling configuration space and calculating the utility of a configuration are given in Section III. Experimental results (Section IV) comparing the utility-guided motion planner to existing sampling-based motion planners indicate that it is capable of significant improvements in planner runtime.

II. RELATED WORK

A. Motion Planning

There have been many extensions to the uniformly random sampling strategy used by the initial PRM algorithm [16]. Generally, these extensions attempt to improve performance by reducing the number of samples required to construct a complete configuration space roadmap.

The Gaussian sampling strategy [4] and the bridge test [11] select configurations that are thought to be close to obstacles or inside narrow passages, respectively. Other heuristic sampling strategies modify obstructed configurations to discover nearby free configurations. These heuristic samplers use obstacle surface properties [1] or dilating and contracting obstacles [12] to modify colliding samples into free ones. All of these strategies are based on uniform random sampling and require additional computational effort to filter configurations to find those thought heuristically to be valuable. Despite this extra computation, only a subset of configurations selected by the heuristic are truly relevant to roadmap construction.

Visibility-based PRM planners [24] label configurations that act as "guards." Such configurations capture a region of configuration space containing every configuration that has a straight line path to the guard. Only configurations that are not in a captured region, or connect two guards are inserted into the roadmap. The roadmaps that are constructed are significantly smaller, but the visibility region is quite expensive to compute.

Two approaches calculate and use the medial axis to minimize the probability that a configuration is obstructed [10], [26]. A significant challenge to these approaches is the difficulty of finding configurations near the medial axis for articulated robots.

In all of these approaches, samples are chosen based on local, fixed, criteria chosen a priori and designed to heuristically estimate the relevance of a configuration. In contrast, the utility-guided sampling strategy directly estimates a configuration's relevance and selects configurations that maximize this value. Further the estimate of relevance is adapted as the planner's state changes.

By focusing on a single path from start to goal configurations, the rapidly-expanding random tree family [17] of single-query motion planners can heuristically bias sampling toward regions of configuration space nearby to these start and goal configurations. The wavefront expansion away from both of these locations insures a focus on finding that particular path. While this approach uses information from the start and goal configurations and the likely path between them to influence its sampling, it does not use information obtained as the planner operates. The practical result of this is repeated attempts to moves through regions that have previously been found to be obstructed. The RRT method has been extended [27] to adjust the sampling distribution used for exploration in an effort to limit attempted exploration of obstructed paths. This approach does not make use of information obtained from sampling but rather heuristically limits the radius within which to allow

connection for points near obstacles.

Several other guided sampling strategies use information obtained from previous experience to guide their behavior. The entropy-guided [5] approach to sampling adapts sampling to find configurations that offer maximal information gain, however, the approach does not calculate maximal *expected* information gain and thus does not make full use of information from previous experience and can exhibit pathological behavior. The model-guided [7] sampling strategy chooses configurations that maximize the decrease in variance of an approximate model of configurations space. While these configurations are relevant to building a model with maximum accuracy they are not necessarily relevant to successful motion planning. Morales et al. [20] propose a method that splits configuration space into regions and selects different planners depending on the features of the regions. The selection is guided by a decision tree that is learned offline. While the selection of planners is guided by observation of configuration space, the methods themselves are not adaptive. Hsu et al. [13] propose sampling using an adaptive mixture of sampling strategies. The mixture of strategies is adapted based on the past success of each strategy. Finally, Jaillet et al. [14] proposes an adaptive approach to RRT that limits the connectivity domain of points in the roadmap where previous connection attempts have failed.

B. Active Learning

Our belief that information from past samples can guide current selections to improve planner performance is supported and inspired by results from the field of active learning. The term "active learning" first appeared in Cohn, Atlas and Lander [8] and encompasses a variety of techniques in machine learning that use the state of their current classification model to select training examples expected to maximize the resultant improvement in the model. Cohn, Gharamani and Jordan [9] showed that a planner which selects configurations that maximize the reduction in variance of the resulting model construct optimally accurate classifiers for the number of samples examined. Further experimental evidence [21] shows that models built using active learning guided by model variance significantly outperform models constructed by uniform random sampling in most domains where different examples carry varying information (as is the case in configuration spaces).

C. Expected Utility

The formalization of expected utility was originally proposed by Danuel Bernoulli [3]. This theory was popularized, in slightly different form, by von Neumann and Morgenstern [25], it is their approach we present here. Expected utility presents a formal approach to specifying and evaluating an agent's preferences regarding actions with non-deterministic outcomes. In utility theory, these actions are referred to as *lotteries*.

A lottery consists of a set of outcomes X. There is a distribution over the lottery, which provides a probability $P(x)$ for each of the outcomes $x \in X$. This type of lottery is termed a *simple* lottery. A *compound* is formed from a probabilistic combination of a set of lotteries.

Given a set of lotteries, an agent has a preference function which provides an ordering of lotteries in terms of the agent's desire to participate in the lottery. This function is called the *utility* function. The *expected utility* of a lottery l is given by the expected value summation over the utility of the individual outcomes:

$$U_{exp}(l) = \sum_x P(x)U(x)$$

This expected value can then be used to choose an individual's preferred lottery.

The role of the utility function is to establish and maintain a preference ordering on the set of outcomes such that any outcome can be judged as preferential to some other outcome. Further the utility function must satisfy several axioms regarding the preference ordering of a mixture of outcomes. [15] gives a thorough explanation of Bernoulli-von Neumann-Morgenstern utility theory.

III. UTILITY GUIDED SAMPLING

To represent all possible paths, a motion planning algorithm must develop a global understanding of the connectivity of a robot's free configuration space. To be efficient, the planner must minimize the exploration required to develop such an understanding.

It is clear that if a planner perfectly understands the structure of a configuration space it can sample optimally. It is also clear that this perfect understanding is generally unavailable to the planner. It is desirable then, to use knowledge of configuration space structure without requiring perfect a priori knowledge. The key observation behind utility-guided sampling is that an approximate model of configuration space structure can be incrementally constructed from past experience and that this approximation can guide future exploration.

Structure in configuration space derives from two sources. First, there is regularity in the distribution of obstructed and free configuration space. Knowing that a configuration's neighbors are obstructed implies a probability that it is obstructed as well. The same is true of a configuration with neighboring free configurations. This regularity means we can learn a predictive approximate model of the configuration space from samples. Second, the planner's current roadmap, which partially represents the connectivity of configuration space, provides additional structure. The state of the roadmap suggests regions were further exploration is useful and shows what is already well understood. By examining its current approximation of this configuration space structure, the sampler can select samples with maximal expected benefit. The formulation of expected utility is used to estimate this expected benefit because it merges both elements of configuration space structure. Recall that the expected utility of a configuration space sample combines the probability of an outcome with the utility of that outcome. The approximate predictive model of configuration space is used to estimate $P(q = i|M)$, the

probability that a configuration is obstructed or free. The planner's approximation of configuration space connectivity provides $U(q = i, R)$ the utility of a particular sample. In the following section we will explore each of these terms in more detail and develop a concrete implementation of a utility-guided sampling strategy.

A. Modeling configuration space

Configuration space can be viewed as a binary classification, $C(q) = 0, 1$, which takes some configuration q and returns whether or not that configuration is obstructed. Because of the topological properties of configuration space, if some q is obstructed, it is more likely that its neighbors are also obstructed. The same is true of free configurations. Given a collection of sampled configurations that have been labeled with their state, we can use learning methods from machine learning [19] to construct an approximation of the function C which we call $\tilde{C}$. This approximation function returns a number between zero and one, estimating the likelihood that a particular configuration is obstructed or free. This approximation function is the approximate model of configuration space.

There are numerous methods for constructing an approximate model of the configuration space function C. In other work [6], [7] we have explored the use of mixture of Gaussian models and locally weighted regression [2]. In this work, we use a simpler k-nearest neighbor model.

Given a collection of configuration space samples Q, which have been labeled with their state, a query configuration q, which has not been observed, and $N(q, k)$, the function that provides the k-nearest neighbors in Q, we calculate $\tilde{C}$ as follows:

$$\tilde{C}(q) = \sum_{i}^{N(q,k)} C(q_i)/k$$

Note that although we don't have a complete definition of C, it is defined for the configurations in the set Q, that we have already observed.

We take the output of $\tilde{C}$ to be the probability that a configuration is free.

$$P(q = \text{free}|M) = \tilde{C}(q)$$

$$P(q = \text{obs}|M) = 1 - \tilde{C}(q)$$

Nearest-neighbor learners have known problems as the dimensionality of the problem expands. To counteract this we use a distance metric first applied to motion planning by Leven [18] that measures the maximum workspace distance between a set of reference points located along the robot. This metric achieves significantly higher model accuracy, especially as dimensionality increases. Because of its simplicity, the nearest-neighbor model offers better computational efficiency than more complex models. This efficiency is an attractive property for motion planning.

B. Utility Functions

We have seen how an approximate model, M, of configuration space can be used to estimate the probability that a configuration has a particular state ($P(q = i|M)$). It remains to define a the utility function for a configuration $U(q = i, R)$.

The choice of utility function is critical to the performance of the sampling strategy. It must adequately characterize the relevance of a configuration to successfully guide sampling.

To develop a concrete implementation of a utility-guided motion planner, we choose to use roadmap information gain as our measure of utility. Information gain for roadmap motion planning is a component of entropy-guided sampling [5] which uses information theory [22], [23] to formalize the contribution that any particular sample makes to the task of motion planning.

Information gain represents the change in the entropy of a system as a result of gaining knowledge related to the system. Given some system S, some new knowledge K, the entropy of the system prior to observing K ($H(S)$), and the entropy of the system after observing K, ($H(S|K)$). The information gain resulting from K is:

$$IG(S|K) = H(S) - H(S|K)$$

For motion planning, the system is the roadmap R and the new information is the observation of some unobstructed configuration q. We use the information gain provided by the configuration as our utility function.

In order to do this in practice, we must define a distribution which has minimal entropy when the roadmap is fully connected. This distribution is described as follows:

At any time, a configuration space roadmap consists of a number of disconnected components. Each of these components has a visibility region containing all free configurations with an unobstructed straight-line path to a node in the component. For our purposes we restrict these regions to be a strictly disjoint set by assigning any configuration in the visibility region of multiple components to the visibility region of the nearest component. Given this set of configuration space regions, we use the probability distribution that a particular free space sample drawn uniformly at random will lie in a particular component's visibility region. It is easy to see that this distribution has the desired characteristics for defining roadmap entropy. When the roadmap is fully connected, there is only a single component and the entropy of the distribution is zero. When there are a large number of different connected components, the entropy is large.

This distribution is used to calculate roadmap entropy. Roadmap information gain is calculated by examining the changes in the roadmap that the addition of a particular sample might produce. The utility of a particular sample is set equal to the roadmap information gain of that sample. Since an obstructed configuration can't modify the state of the roadmap it offers no information and thus no utility.
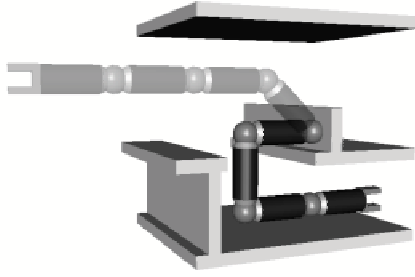
(a) The fixed arm workspace with 12-DOF robot
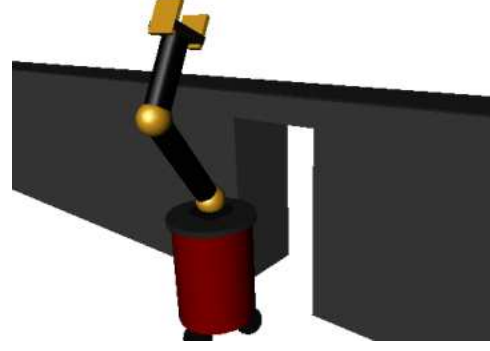
(b) The mobile workspace with 6-DOF robot

Fig. 2. The two experimental workspaces

C. Sampling algorithm

Given the predictive model and utility function, the expected utility of some configuration q is:

$$U_{exp}(q|M) = \sum_{i \in \text{obs,free}} P(q = i|M) \cdot U(q = i, R)$$

$$U_{exp}(q|M) = P(q = \text{free}|M) \cdot U(q = \text{free}, R)$$

$$U_{exp}(q|M) = \tilde{C}(q) \cdot IG(M|q)$$

In practice, the computation of utility for each potential sample in the configuration space is computationally impractical. However, an analysis [5] of roadmap information gain shows that utility can be heuristically maximized by sampling configurations likely to connect two large disjoint components. Thus samples on the border region between two components are likely to provide maximal information gain. This border region is difficult to compute exactly but can be approximated using hyper bounding-boxes around each connected component [5] or by selecting configurations from the area surrounding the midpoint of the line connecting a random node from each component. We use this second approach for our experiments with utility-guided sampling, but we have not observed significant differences between the two approximations. Other heuristic approximations such as trees of bounding boxes or oriented hyper-ellipses or clustering algorithms are also possible and bear future exploration. Since we equate information gain and utility, heuristically maximizing information gain, maximizes expected utility as well. Pseudo-code for the utility-guided sampling algorithm is shown in Figure 3.

IV. EXPERIMENTS

To validate our new sampling strategy we ran a set of experiments in two workspaces with robots of varying degrees of freedom. The first experimental workspace was a fixed articulated arm situated in a constrained environment. Each joint in the robot has three degrees of freedom. We ran experiments with robots with either three or four links, resulting in nine and twelve degrees of freedom respectively. The workspace with the twelve degree of freedom arm is shown in Figure 2a.

UtilityGuidedSampling(M : Model, Roadmap : R) : q
    q := nil
    do k **times**
        q' = EntropyGuidedSample(R)
        if $(P(q' = \text{free}|M) > P(q = \text{free}|M))$
            $q = q'$
    return q

EntropyGuidedSample(R : Roadmap) : q
    do
        C_1 := random component in R
        C_2 := random component in R, $C_1 \neq C_2$
    while (distance(C_1, C_2) > Threshold)
    q_1 := random node in C_1
    q_2 := random node in C_2
    q_n := midpoint(q_1, q_2)
    for each d in D
        $q_r[d]$:= $q_n[d] \pm$ UniformRandom(τ)
    return q_r

Fig. 3. The utility-guided sampling algorithm

The second workspace contains a mobile robot in a world separated by a wall containing a doorway. The mobile base is holonomic, providing two degrees of freedom. On top of the mobile base there is a two link arm. One experiment had one degree of freedom at each joint, the other each joint had two degrees of freedom. Thus, the robots had four and six degrees of freedom respectively. The workspace with the six degree of freedom robot is shown in Figure 2b.

We compare our algorithm against uniform sampling, bridge sampling, entropy-guided sampling and model-based sampling. The identical roadmap construction algorithm was used for all experiments, only the sampling strategy was changed. Both workspaces are split in two by a narrow passage. In the first workspace, this passage consists of folding the arm underneath the table. In the second the narrow passage leads through

the doorway. For all of the experiments the planner was run until a roadmap which lead through this narrow passage was found. The algorithm and strategies were implemented in the Java programming language and all experiments ran on a 3Ghz Pentium 4 running the Linux operating system.

The utility-guided sampling strategy incurs some overhead both in sampling to construct a model of configuration space and in evaluating the expected utility in order to select configurations. This overhead is included in all of the reported runtimes. In order to examine the influence of this overhead, we ran additional experiments in which we profiled the behavior of the various implementations using a sampling profiler. In the runtime graphs, each runtime is broken down into four categories:

Collision checking
> The examination of individual configurations to determine if they are obstructed or free,

Edge Checking
> The examination of a series of an interpolated series of connections between two configurations to determine if a straight line path is possible,

Guided Sampling
> The calculation and selection of configurations guided by information from previous experience (note that this only pertains to the entropy-guided, model-based and utility-guided motion planners),

Roadmap Construction
> All other activities pertaining to constructing a roadmap (e.g. finding neighbors, inserting vertices/edges, etc).

It is instructive to note that although a significant portion of the runtime of the guided sampling strategies is consumed by selecting configurations. The configurations chosen are more relevant to the motion-planning process. The resulting computational savings in edge checking and roadmap construction outweighs all overhead from selecting samples.

For each robot in each workspace we ran ten experiments. The average runtimes for these experiments are shown in Figure 4. In the graphs we also show profiling information which shows the percentage of time the algorithm spends performing particular pieces of the roadmap construction process.

In the graphs it is also interesting to compare the performance of utility-guided sampling with entropy-guided [5] and model-based [7] sampling. Entropy-guided sampling uses roadmap structure, while model-based sampling uses an approximation of the state of the configuration space. Each of these approaches does well in one environment and poorly in the other. This shows that their performance is dependent on environmental features. Model-based sampling does poorly in the mobile robot environment because it is uniformly interested in the wall, not just in the opening that is crucial for the solution of the problem. On the other hand entropy-guided sampling does poorly in the fixed arm world where its greedy attempts to connect disjoint configuration space components generally end in failure. By blending together both sources of configuration space information the utility-guided

sampling strategy is able to achieve strong improvements in both environments. An important consideration for motion-planning is sensitivity to degenerate behavior in particular types of configuration space. The fact that utility-guided planning avoids the degenerate behavior exhibited by its component pieces (entropy-guided and model-based sampling) shows that it is more robust and reduces the probability that the planner will exhibit degenerate behavior.

We also ran a series of experiments where we interrupted the roadmap construction process at timed intervals and tested the coverage of the roadmap which had been constructed so far. To test these partial roadmaps we selected twenty paths at random and recorded the fraction of paths that were successful. For each algorithm and experimental domain we ran ten of these series. The average of these ten runs is shown in Figure 5. These graphs demonstrate the rate at which each algorithm provides coverage of the entire configuration space. In all experiments, utility-guided sampling outperforms other methods.

The results of these experiments clearly demonstrate that the utility-guided sampling strategy improves the performance of the PRM algorithm. In all cases the utility-guided strategy reduces the average runtime by at least a factor of two. In several worlds the improvement is even more dramatic. The graphs illustrating coverage as a function of time also show that the utility-guided sampling strategy produces a planner which achieves greater coverage of the space of possible paths more rapidly than other sampling strategies.

V. CONCLUSIONS

In the preceding we have proposed a novel approach to multi-query motion planning which uses information from its previous experience to guide sampling to more relevant configurations. Every exploration of configuration space provides information to a motion planner. To be maximally efficient, a motion planner must exploit information from past experience to select maximally beneficial configurations to sample next.

Our proposed approach begins by constructing an approximate model of configuration space. This model captures and maintains information from each configuration and allows the prediction of the state of unobserved configurations. In conjunction with a utility function which measures the relevance of a configuration the model enables a sampling strategy which selects configurations that have maximal expected utility to the motion planner.

To demonstrate the effectiveness of this approach we implemented a utility-guided motion planner using a nearest-neighbor approximate model and roadmap information gain [5] as the utility function. Experiments run with two robots with varying degrees of freedom indicate that the utility-guided approach to motion planning results in faster runtimes compared to existing state of the art approaches. The experiments also show that the utility-guided approach is less sensitive to workspace features that hinder the performance of other sampling strategies. These experimental results indicate that the proposed utility-guided sampling is both a more
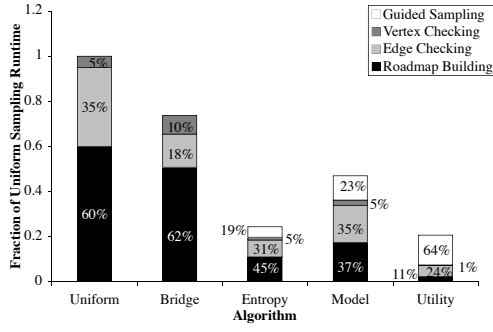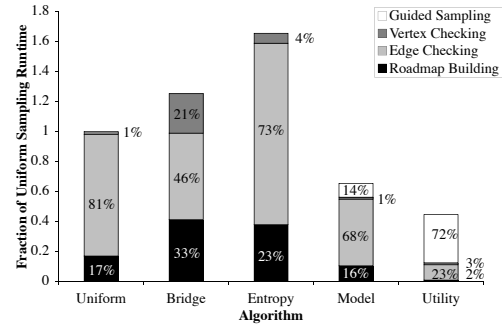
(a) 4-DOF mobile manipulator
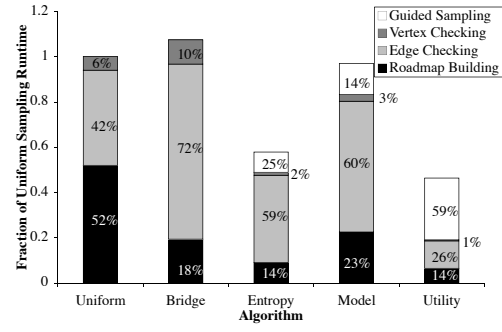
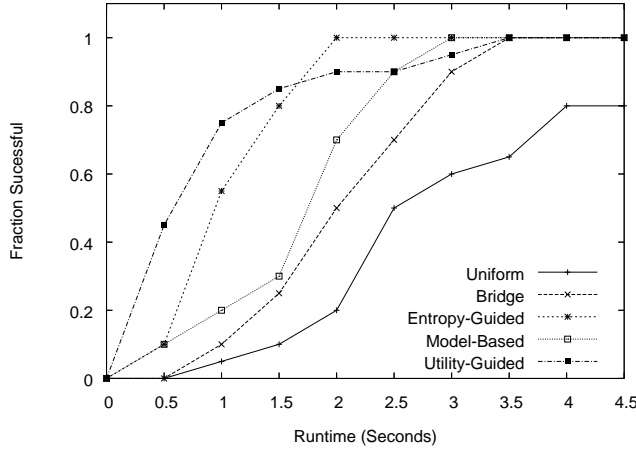(b) 6-DOF mobile manipulator
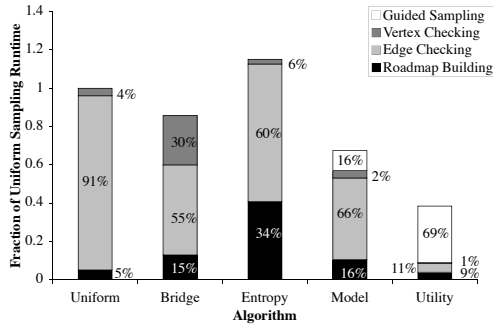
(c) 9-DOF arm

(d) 12-DOF arm

Fig. 4.   Runtimes for various sampling strategies as a percentage of the runtime using the uniform sampling strategy

efficient and more robust sampling strategy than existing approaches.

REFERENCES

[1] N. Amato, O. B. Bayazit, L. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective*. AK Peters, 1998.

[2] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.

[3] D. Bernoulli. Exposition of a new theory of risk. *Econometrica (translated and reprinted)*, 1954. originally in Comentarii Academiae Scientiarum Imperialis Petropolitanae, 1738.

[4] V. Boor, M. Overmars, and F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Proceedings of the International Conference on Robotics and Automation*, 1999.

[5] B. Burns and O. Brock. Information theoretic construction of probabilistic roadmaps. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 650–655, Las Vegas, USA, 2003.

[6] B. Burns and O. Brock. Entropy-guided single-query motion planning. In *Proceedings of the International Conference on Robotics and Automation*, 2005.

[7] B. Burns and O. Brock. Model-based motion planning. In *Proceedings of the International Conference on Robotics and Automation*, 2005.

[8] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[9] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical methods. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[10] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in PRM planners. In *Proceedings of the International Conference on Robotics and Automation*, volume 2, pages 1408–1413, San Francisco, USA, 2000.

[11] D. Hsu, T. Jiang, J. Reif, and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Proceedings of the International Conference on Robotics and Automation*, 2003.

[12] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, pages 141–154. A K Peters, 1998.

[13] D. Hsu, G. Snchez-Ante, and Z. Sun. Hybrid prm sampling with a cost-sensitive adaptive strategy. In *Proceedings of the International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[14] L. Jaillet, A. Yershova, S. LaValle, and T. Simeon. Adaptive tuning of
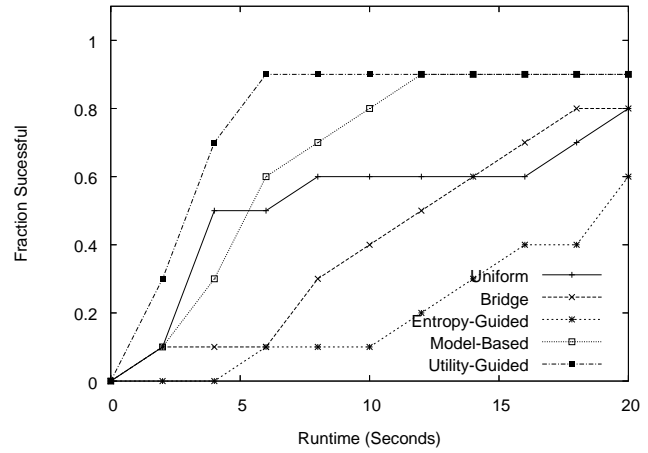
(a) 4-DOF mobile manipulator
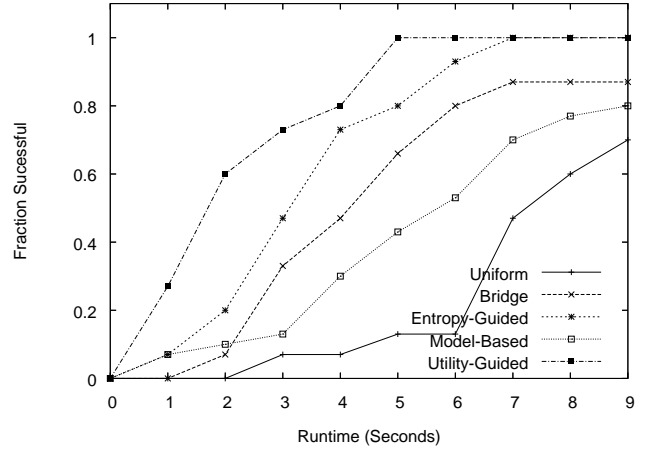
(b) 6-DOF mobile manipulator
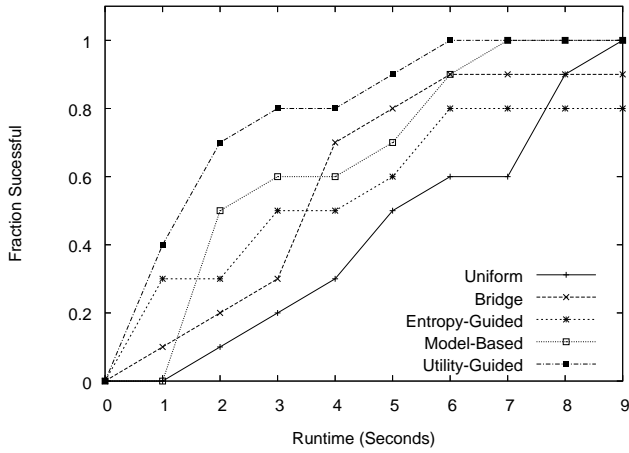
(c) 9-DOF arm

(d) 12-DOF arm

Fig. 5.   Fraction of configuration space covered as a function of runtime for various sampling strategies and workspaces

the sampling domain for dynamic-domain RRTs. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2005. To Appear.

[15] N. E. Jensen. Introduction to bernoullian utility theory. *Swedish Journal of Economics*, pages 163–183, 1967.

[16] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[17] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, pages 293–308, 2000.

[18] P. Leven and S. Hutchinson. Toward real-time path planning in changing environments. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, 2000.

[19] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[20] M. Morales, L. Tapia, R. Pearce, S. Rodriguez, and N. Amato. A machine learning approach for feature-sensitive motion planning. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, 2004.

[21] M. Saar-Tsechansky and F. Provost. Active sampling for class probability estimation and ranking. *Machine Learning*, 54(2):153–178, 2004.

[22] C. E. Shannon. A mathemtatical theory of communication. *Bell System Technical Journal*, 27:379–423, July 1948.

[23] C. E. Shannon. A mathemtatical theory of communication. *Bell System Technical Journal*, 27:623–656, October 1948.

[24] T. Simeon, J.-P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 1999.

[25] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1953.

[26] Y. Yang and O. Brock. Adapting the sampling distribution in prm planners based on an approximated medial axis. In *Proceedings of the International Conference on Robotics and Automation*, 2004.

[27] A. Yershova, L. Jaillet, T. Simeon, and S. M. LaValle. Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain. In *Proceedings of the International Conference on Robotics and Automation*, 2005.