Toward Optimal Configuration Space Sampling

name
lab
affiliation

***Abstract*— Efficient motion planning is obtained by focusing computation on relevant regions of configuration space. In the following we propose a new approach to multi-query sampling-based motion planning, which exploits information obtained from earlier exploration and its current state to guide exploration. This approach attempts to minimize the selection of samples to those required to completely capture configuration space connectivity. Our planner constructs an approximate model of configuration space that is used in conjunction with a utility function to select configurations with maximal expected importance given the planner's current state. The resulting utility-guided planner is online and adaptive. Its behavior adjusts to the developing state of the motion planner and its understanding of the configuration space. Experimental comparisons with existing planners show that this utility-guided approach significantly decreases the runtime required for motion planning.**

I. Introduction

Efficient motion planning requires the application of computational resources to areas of configuration space that are most beneficial. In the following we present a novel sampling strategy which locates and samples in these beneficial regions.

The success of multi-query, sampling-based motion planning results from its implicit representation of configuration space. Computing the exact boundary of configuration space obstacles is avoided by using a single sample to capture a large region of configuration space. Were complete knowledge of a configuration space available, it would be possible to design an optimal sampling strategy. Such a strategy would select a minimal set of samples whose associated captured configuration space, when unioned, comprises all free configuration space. In most cases, the size of this minimal set is quite small. For a simple two dimensional world, this set is illustrated in Figure 1.

Unfortunately, complete knowledge of configuration space is almost always unavailable to a motion-planner. Regardless, the efficiency of any sampling-based motion planner derives from its ability to refine its selection of configurations toward this minimal set. In the following we present a novel sampling strategy aimed at reducing the number of samples required by a motion planner toward this optimal number.

Currently, sampling-based planners use one of two fundamentally different approaches to choose configurations. The first, used by probabilistic roadmap (PRM) techniques, is based on uniform random sampling [14]. 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 [15] which progresses the wavefront more quickly toward large unexplored regions.

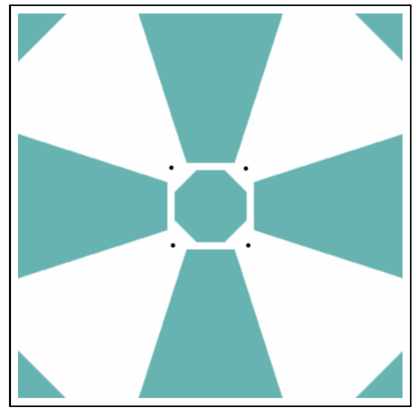


Fig. 1. A simple configuration space for a point robot and a minimal set of samples which completely represent it

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. Our approach attempts to find a middle ground performing both exploratory and path-directed sampling.

Every configuration examined provides information to the motion planner about both the state of configuration space and the progress of a successful path. However, neither of the two previously mentioned sampling strategies make use of observed information to select configurations. Because they ignore this information, current sampling-based motion planners expend computational resources on regions that are either already well represented by previous samples or known to be obstructed from past experience. Only by remembering the information obtained through earlier sampling and using this information to guide current choices can sampling move toward the optimal strategy derived from complete knowledge of the configuration space.

In this paper, we present a novel view of sampling in sampling-based motion planning based on the assumption that using information to guide sampling improves the relevance of each sample. This increased relevance reduces the number of configurations required by the motion planner to construct a complete representation of configuration space. This assumption is supported by results from the active learning literature [8], [9], [17] regarding learning a binary classification, for example the obstructed or free state of a configuration space. The results show that selecting samples using information from previous samples significantly reduces the number of samples required to produce equivalent accuracy when compared to uniform random sampling. Reducing the number of samples required by the motion planner results in finding a complete representation more quickly.

However, motion planning is unlike machine learning. We are less interested in optimizing model accuracy, but rather optimizing the utility of the model for motion planning. Thus, we base this sampling strategy on the formal notion of expected utility [3], [21]. For some configuration q, and an approximate model M and a roadmap R, the expected utility, $U_{exp}(q)$, is:

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

Where $P(a = i|M)$ is the probability of q having the state i given some model M and $U(q = i, R)$ is the utility of the configuration q given the planner's current state. The utility-guided sampling strategy selects configurations that maximize this value. This formalization exploits information from previous samples in two ways. First the probability of a configuration q having a particular state i is estimated using an approximate model of configuration space built from previous experience. Secondly, the utility of a configuration is derived from the current state of the roadmap constructed from earlier unobstructed samples. Details of modeling configuration space and calculating the utility of a configuration are given in Sections III-A and IV respectively. Experimental results (Section V) 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 [14]. 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 [20] 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], [22]. 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 [15] 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. This can be seen as sampling to maximize utility, but not *expected utility*. 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 [23] 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.

Two earlier 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.

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 [17] 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 [21], 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 an 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 [13] gives a thorough explanation of Bernoulli-von Neumann-Morgenstern utility theory.

III. Using Models to guide sampling

Every motion planner constructs a model of configuration space. We propose constructing a model that can be used to guide sampling. Each exploration of configuration space provides information about that space to the planner. To maximize its efficiency, a planner must use information obtained from previous experience to guide current sampling. We combine information obtained from sampling into an approximate model of configuration space and using this integrated information to guide the selection of configurations. We are motivated by results from the active learning literature which demonstrate that models built using guided sampling exhibit improved performance relative to models built from random samples.

A. Modeling configuration space

Configuration space can be viewed as a binary classifier, $C(q) = 0, 1$, 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 classification methods from machine learning [16] to construct an approximation of the function C which we call 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 samples of configuration space 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 C' as follows:

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

Note that although we don't we 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 C' to be the probability that a configuration is free.

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

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

While nearest-neighbor problems have known problems as the dimensionality of the problem expands, we have empirically observed them to have reasonable accuracy in our experimental environments (Figure 2). 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. Relationship to Active Learning

In the context of motion planning, active learning is an effective solution to the "narrow-passage" problem [12]. Active learning directs sampling toward regions of high variance. Because narrow passages consist of a close arrangement of free and obstructed configurations they are naturally regions of high variance. Active learning in the context of motion planning has been shown to select configurations that improve planner performance [7].

Unfortunately, although regions, such as narrow passages, that are relevant to motion planning have high variance, the converse is not necessarily true. Different regions of configuration space with high variance have different relevance to motion planning. All boundary regions between free and obstructed space have higher variance than uniform regions, and cul de sacs are indistinguishable from narrow passages in terms of their variance. Additionally, the relevance of a region is also defined by its relationship to the global configuration space connectivity. A narrow passage in configuration space has equally high variance regardless of the shape of the remainder of configuration space, but it only has high relevance to motion planning if it is the only connectivity between two configuration space regions. If there is a wider passage connecting the same regions, the narrower passage is significantly less relevant.

Active learning was designed to construct models with maximal accuracy, but for the task of motion planning, we are actually interested in a different type of model, a model with maximal utility. Such a model maximizes our ability to motion plan successfully in the configuration space. In following section we described this utility-guided approach to motion planning.

IV. Utility-guided Sampling

To insure that we select configurations that are relevant to motion planning, rather than the construction of the approximate model of configuration space, we use the notion of expected utility to guide our selection of configurations. Previously we have defined the expected utility of choosing a particular configuration as:

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

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. We can conceive of many ways to measure this relevance. Configurations that lie in unexplored regions near to existing roadmap components, for example. Or configurations that are maximally distance from existing components in unexplored regions of configuration space. Further, a function which measures the relevance of a configuration to roadmap construction might be combined with additional measures of utility to bias paths through certain types of configurations. Configurations with attractive dynamic characteristics, or end-effector locations might have higher utility in this combined function. Alternatively, the utility function could be used to create a single-query sampling strategy by defining the utility of a configuration in reference to a particular path.

A. Example Utility function

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 developed for entropy-guided sampling [5]. This analysis uses information theory [18], [19] 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:

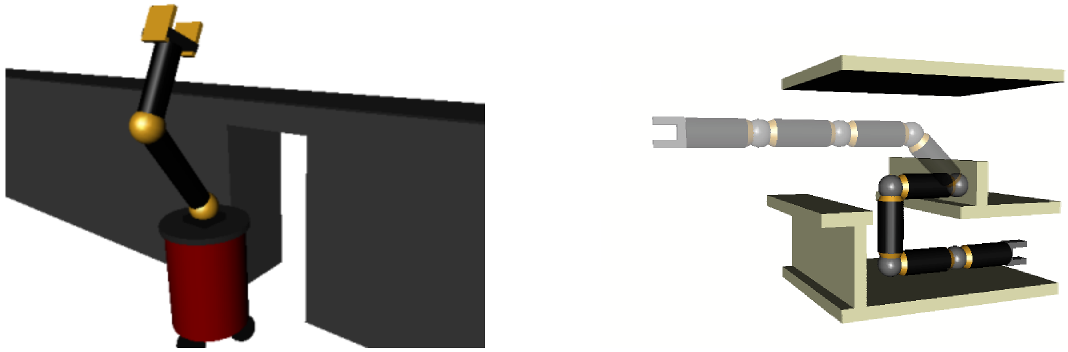
$$IG(S,K) = H(S) - H(S|K)$$

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

$$U(q,R) = IG(R,q)$$

In order to do this in practice, we must define a distribution which has minimal entropy when the roadmap is fully connected. As a roadmap is constructed, it consists of a number of disconnected components. Each of these components have a region of configuration space which they "cover." Any configuration within this covered region has a straight-line path to a node in the component and this path is shorter than any straight-line path to another component. If we examine the distribution representing the probability that a free configuration chosen at random will land in a particular component's area, we see that this distribution has the desired characteristics. 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. Given this distribution, we evaluate samples based upon the information gain they provide. Note that an obstructed configuration has no possibility of extending or connecting components in the roadmap, so the information gain and therefore the utility of an obstructed configuration is zero.

The remaining complication is the estimation of the area covered by a connected component. It is possible to calculate this area explicitly, but doing so is tantamount to complete configuration space exploration and is thus computationally intractable. Instead, a bounding box is maintained around each component. These bounding boxes are at best a coarse approximation and future research is directed at developing a better one. Configurations lying directly between two bounding boxes as well as configurations that are in the intersection of bounding boxes are estimated to offer greater information gain. Greater details of this approach can be found in [5].



(a) The mobile workspace with 6-DOF robot

(b) The fixed arm workspace with 12-DOF robot

Fig. 2. The two experimental workspaces

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

Fig. 3. The utility-guided sampling algorithm

B. Sampling algorithm

Given a method for estimating the state of some unobserved configuration and evaluating its utility to the motion planner, we can calculate the expected utility of some configuration q:

$$\begin{aligned}
U_{exp}(q|M) &= \sum_{i \in \text{obs,free}} P(q=i|M) \times U(q=i,R) \\
U_{exp}(q|M) &= P(q=\text{obs}|M) \times U(q=obs,R) + \\
&\quad\; P(q=\text{free}|M) \times U(q=free,R) \\
U_{exp}(q|M) &= P(q=\text{free}|M) \times U(q=free,R) \\
U_{exp}(q|M) &= C'(q) \times IG(M|q)
\end{aligned}$$

Although it is possible to evaluate this function in closed form over the entire configuration space, it is computationally intractable to do so. Instead, we use entropy-guided sampling to suggest a number of configurations that have high utility. Of these configurations, we choose the one most likely to be unobstructed. This approximation assumes that all samples chosen by entropy guided sampling offer equal information gain. This is obviously not the case, but this approach offers an efficient approximation for the actual expected utility function. The pseudo-code for this sampling algorithm is shown in Figure 3.

V. Experiments

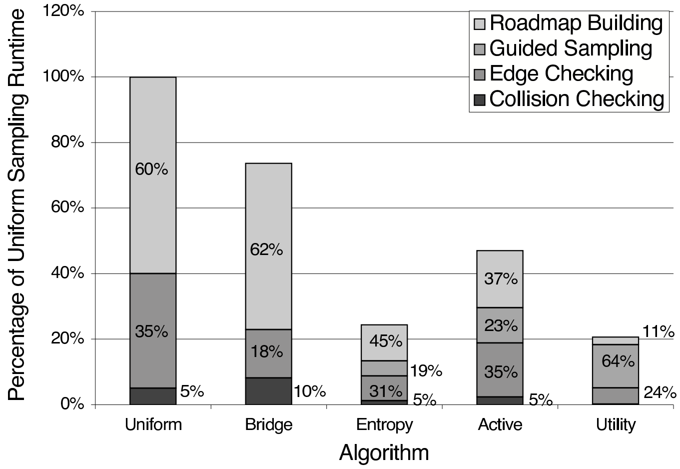
To validate our new sampling strategy we ran a set of experiments in two workspaces with robots of varying degrees of freedom. The first workspace consists of a fixed articulated arm situated in a constrained environment. Each joint in the robot has three degrees of freedom, we ran experiments with robots 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 2b.

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 a two link arm was placed. In one experiment there was one degree of freedom at each joint, in 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 2a.

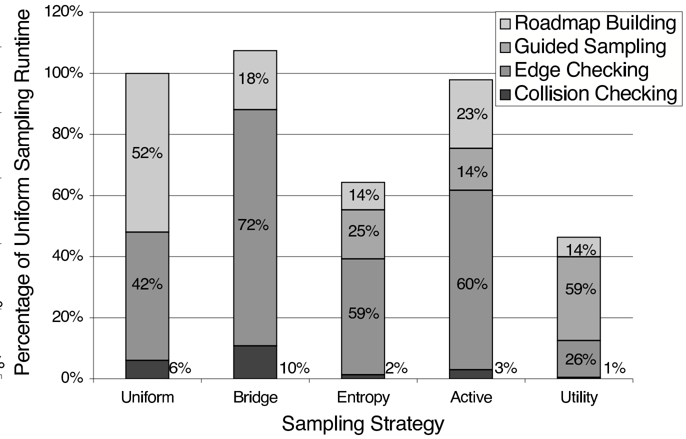
We compare our algorithm against uniform sampling, bridge sampling, entropy-guided sampling and model-based sampling. The identical roadmap construction algorithm was used for all experiments, only the sampling strategy was changed. In both cases, the workspace is split in two by a narrow passage. In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

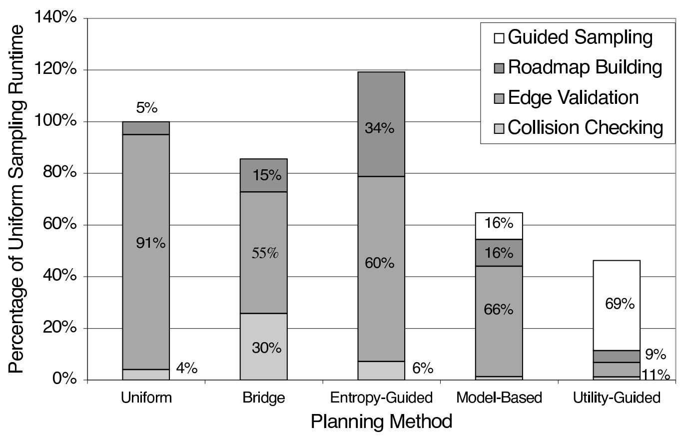
- **Collision checking**: The examination of individual configurations to determine if they are obstructed or free.
- **Edge Checking**: The examination of a series of an interpolated series of connections between two configurations



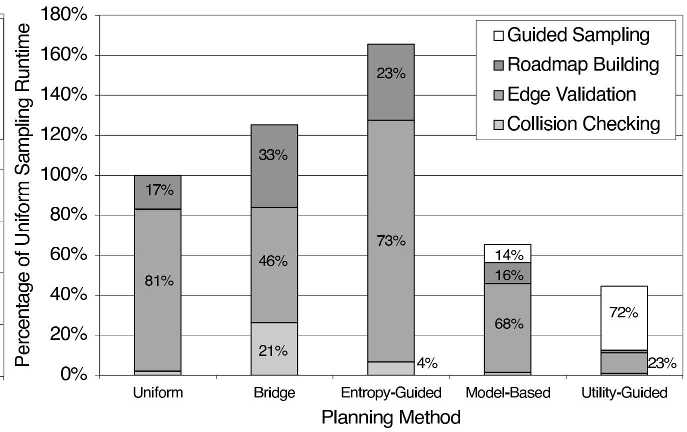
(a) 4-DOF mobile manipulator



(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

to determine if a straight line path is possible.

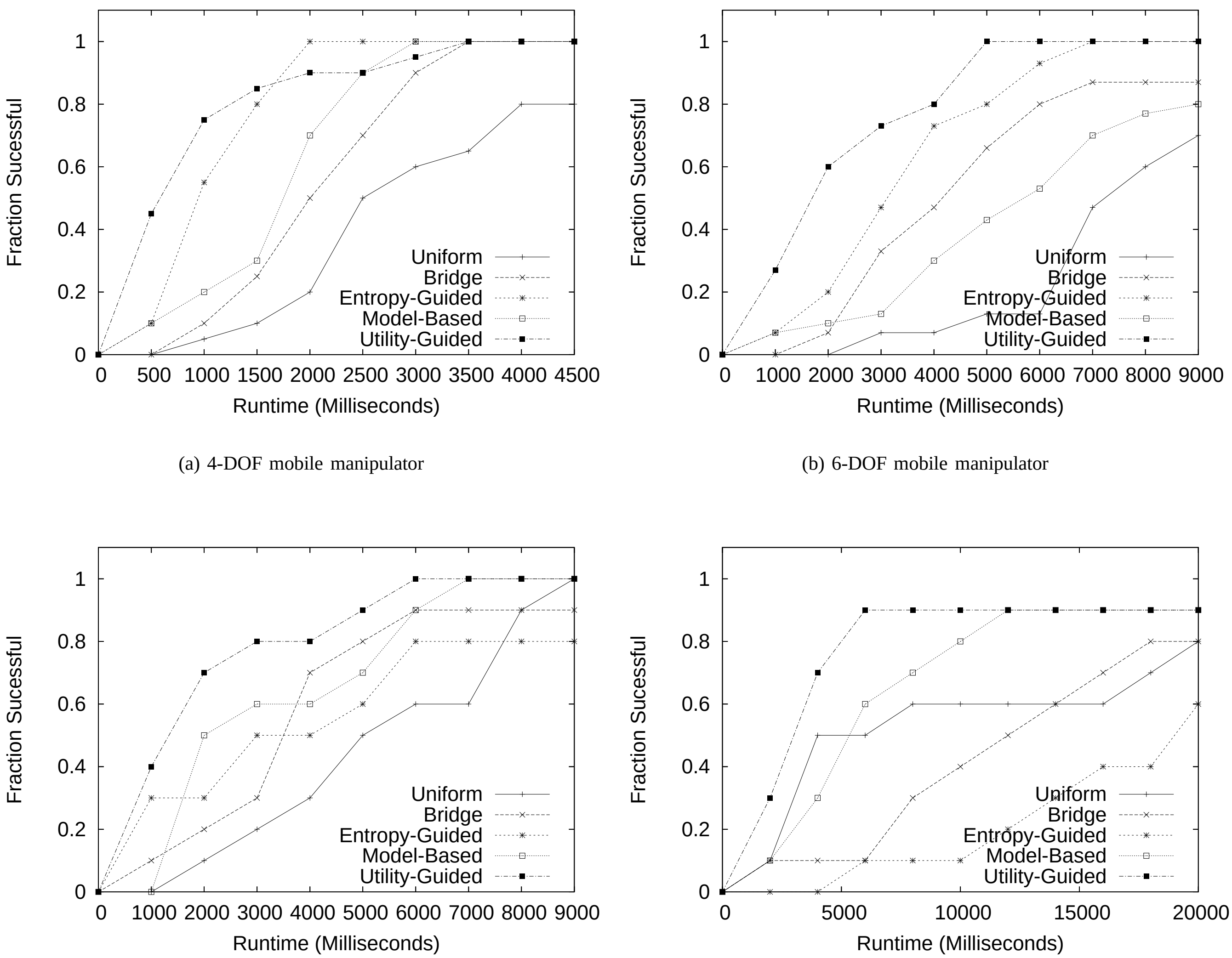
- **Guided Sampling**: The calculation and selection of configurations guided by information from previous experience (note that this only pertains to the entropy-guided, model-based and utility-guided motion planners).
- **Roadmap Construction**: All other activities pertaining to constructing a roadmap (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. These average runtime for these experiments are given in graphically 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.

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 are instructive because they demonstrate the rate at which each algorithm provides coverage of the entire configuration space.

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



(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

show that the utility-guided sampling strategy results in a planner which achieves greater coverage more rapidly than other sampling strategies.

It is also instructive to note that both the entropy-guided and model-based sampling strategies are better suited to one workspace or the other. The entropy-guided sampling strategy does extremely well in the mobile-robot environment, while doing quite poorly in the articulated arm workspace. We believe that this is because the bounding boxes used as an approximation of the coverage of each connected component are a better approximations in the mobile robot's configuration space than in the articulated arm. Although model-based sampling demonstrates improvement over uniform sampling in all environments, its improvements are significantly less in the mobile robot environment. In that environment, focusing the planner on connecting the disconnected components on either side of the wall significantly improves performance. The utility-guided sampling strategy exploits the strengths of both strategies while avoiding their weakness. The integration of multiple sources of information focuses the motion planner on configurations which are truly relevant and leads to improved performance.

VI. 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 all available information in order to proactively choose configurations with maximal expected benefits.

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 it enables a sampling strategy which selects configurations that have maximal expected utility or importance 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 information gain [5] as the utility function. Experiments run with two robots with varying degrees of freedom show that the utility-guided approach to motion planning results in faster runtimes compared to existing state of the art approaches.

References

[1] N. Amato, B. Bayazid, 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, 2003.

[6] B. Burns and O. Brock. Entropy-guided single-query motion planning. In *To Appear International Conference on Robotics and Automation (ICRA)*, 2005.

[7] B. Burns and O. Brock. Model-based motion planning. In *To Appear International Conference on Robotics and Automation (ICRA)*, 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] N. E. Jensen. Introduction to bernoullian utility theory. *Swedish Journal of Economics*, pages 163–183, 1967.

[14] 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.

[15] 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.

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

[17] M. Saar-Tsechansky and F. Provost. Active sampling for class probability esitmation and ranking. *Machine Learning*, 2004.

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

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

[20] 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.

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

[22] 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.

[23] 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. In Submission.