2005

Sampling-Based Motion Planning Using Predictive Models

Brendan Burns
University of Massachusetts - Amherst

Oliver Brock
University of Massachusetts - Amherst

Sampling-Based Motion Planning Using Predictive Models

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

Abstract— **Robotic motion planning requires configuration space exploration. In high-dimensional configuration spaces, a complete exploration is computationally intractable. Practical motion planning algorithms for such high-dimensional spaces must expend computational resources in proportion to the local complexity of configuration space regions. We propose a novel motion planning approach that addresses this problem by building an incremental, approximate model of configuration space. The information contained in this model is used to direct computational resources to difficult regions, effectively addressing the narrow passage problem by adapting the sampling density to the complexity of that region. In addition, the expressiveness of the model permits predictive edge validations, which are performed based on the information contained in the model rather then by invoking a collision checker. Experimental results show that the exploitation of the information obtained through sampling and represented in a predictive model results in a significant decrease in the computational cost of motion planning.**

I. Introduction

Complete motion planning requires understanding a robot's configuration space. The acquisition of such an understanding is computationally challenging. The general motion planning problem has been shown to be PSPACE complete [18]. Sampling-based motion planning methods [12], [14] address this difficulty by constructing a connectivity graph which implicitly approximates the structure of the configuration space, thus minimizing exploration. However, even sampling-based methods can require an exponential number of uniformly placed samples to build a representation of configuration space [20].

Uniform sampling, which forms the basis of almost all multi-query sampling-based motion planning approaches, makes the implicit assumption that configuration space is uniformly complex. As a consequence, a motion planner using uniform sampling must expend on every region of configuration space, the amount of computation required by the most complex region. This shortcoming of uniform sampling is generally referred to as the "narrow passage problem" [9]. A great deal of research in sampling-based motion planning is concerned with the resolution of this problem [1], [4], [8], [10]. The approaches proposed in the literature so far use heuristics to filter configurations after examination in the configuration space but prior to insertion into the roadmap.

In this paper we introduce a novel motion planning technique, called model-based motion planning. This technique is motivated by the insight that an efficient, practical motion planner has to exploit the structure inherent in a particular problem instance to avoid the computational complexity of the general motion planning problem. This new approach incrementally constructs and refines an approximate statistical model of the entire configuration space. The model indicates the areas of configuration space which are complex and the areas which are simple. This information is used to bias sampling toward difficult regions. As areas of configuration space become understood, i.e., the model represents them accurately, they receive no further sampling. Consequently, the proposed planner expends computational resources in proportion to the local complexity of configuration space regions, effectively addressing the narrow passage problem.

The approximate model built by model-based motion planners is also capable of making predictions about unexplored parts of the configuration space. The proposed model-based motion planner takes advantage of these predictions to build *predictive roadmaps*. Predictive roadmaps avoid unnecessary invocations of a collision checker for edge validation when the model can predict the state of the edge with high confidence. Exploiting predictions from the model greatly reduces the computational cost of roadmap construction.

Experimental results show that the proposed model-based method is outperforms existing sampling-based motion planners for motion planning problems in relatively high-dimensional configuration spaces.

II. Related Work

A. Motion Planning

All global motion planners require a model of configuration space [13]. Due to the computational complexity of modeling configuration space obstacles exactly, however, the most efficient techniques for motion planning construct approximate models by sampling configuration space. The probabilistic roadmap (PRM) method [12] constructs a sample based roadmap of configuration space. Because uniform sampling is used to construct the roadmap, PRM planners must sample the entire configuration space with the density required by the most complex area of configuration space. This presents a serious computational challenge when the configuration space contains narrow passages [9]. Much of the research in sampling-based motion planning over the last decade has focused on the design of sampling strategies to address the narrow passage problem.

Some approaches attempt to find points near obstacles, noting that they are more likely to be in narrow passages. These methods use heuristics based on obstacle surface properties [1] or shrinking and growing obstacles [9] to modify colliding samples into free ones. This process can be computationally expensive and, even for points near obstacles, the minority of them are actually in narrow passages.

The computational cost of building large roadmaps with numerous edges leads to sampling strategies that attempt to minimize unnecessary samples. The Gaussian sampling strategy [4] and the bridge test [8] ensure that most configurations in the roadmap are close to obstacles or lie inside a narrow passages, respectively. Visibility-based PRM planners [19] minimize the number of samples inserted into the roadmap by ensuring that randomly generated configurations are not redundant with milestones already present in the roadmap. In a different approach, Fuzzy PRM planners [17] reduce the number of necessary edge validations by estimating their probability of being free. The probability of an edge is estimated by a calculation based upon the length of the edge. The Fuzzy PRM graph bears a passing resemblance to the predictive roadmap, but the Fuzzy PRM approach eventually validates every edge and its approximations of edge probabilities are not based on observations of the configuration space.

The methods discussed so far are classified as multi-query approaches. Single-query methods avoid exploration of the entire configuration space by searching for a single path. Lazy PRM planners [3] bias exploration toward regions close the the initial and final configurations. Rapidly exploring random trees (RRTs) [14] use simulated diffusion to build a connectivity trees in configuration space connecting start and goal configurations. The approach of diffusion is also taken by expansive spaces [10], an approach quite similar to RRTs.

The entropy-guided motion planning approach [5] maintains an approximate representation of connected components of the roadmap to influence the selection of configurations lying between these components. Entropy-guided motion planners demonstrate that exploiting the information represented in expressive models can significantly reduce the computational requirements of motion planning.

More recently, machine learning techniques have been employed in the context of motion planning. In particular, learning techniques have been used to determine which motion planner is applicable to a particular region of configuration space [16] and to adapt the mixture of a number of sampling strategies during roadmap construction [11].

B. *Machine Learning*

The task of building an approximate model of configuration space can be viewed as a classification task, in which points in configuration space are classified as free or obstructed. The machine learning literature provides numerous algorithms for classification [15]. In the context of this paper we limit the discussion to our chosen technique of locally weighted regression [2]. Locally weighted regression is an efficient approach to modeling arbitrary functions based on samples. The approach has a number of attractive features for approximating configuration space. Models based on locally weighted regression are locally adaptive to the structure of the underlying function, their training cost is constant, regardless of the number of training examples, and an efficiently computable closed form derivation of an active learning strategy [7] exists.

Active learning considers the task of learning from examples when the learner can select the data from which it learns. Active learning selects examples which maximize the accuracy of the resulting learner. Modeling configuration space is an active learning situation since any configuration can be examined by the collision checker. We use the derivation of an active learner for locally weighted regression presented by Cohn et al. [7]. Further details concerning locally weighted regression and active learning are given in Section III-A.

III. Model-Based Motion Planning

To build a computationally efficient sampling-based motion planner we propose the use of an approximate model of the entire configuration space. This approximate model is constructed incrementally, as a solution to the motion planning problem is computed. The sampling strategy associated with such a model uses information from the model to adapt sampling densities in proportion to an area's complexity. Complex regions, which by definition are more difficult for the model to understand, will be sampled densely, while areas which are easy to represent are sampled sparsely.

The approximate model also provides predictions about unexplored regions of configuration space. Predictions require significantly less computation than the invocation of a the collision checker. By exploiting the predictive capabilities of the model to reduce the required number of invocations of the collision checker, a significant reduction in computational cost of motion planning can be achieved.

A. *Underlying Representation*

Machine learning traditionally concerns itself with the approximation of a function $f(x) \rightarrow y$ where x is the (possibly multi-dimensional) input, and y is the (possibly multi-dimensional) classification or prediction. For configuration space approximation the input x to the function is a point in configuration space. The output y of the approximating function is a continuous value in the range $[-1, 1]$. Training configurations are labeled with -1 if the configuration is obstructed and 1 if the configuration is free.

Locally weighted regression [2] (LWR) provides a computationally efficient sample-based model. Given a query point, locally weighted regression fits a surface to nearby training points to make a prediction. A distance weighting function calculates the influence that a particular training point has on the model's prediction of the query point. We have chosen to use a Gaussian function for distance weighting:

$$w(x,x') = e^{-k(x-x')\dot{(}x-x')}$$

The smoothing parameter k adjusts the spread of the Gaussian. A larger spread incorporates information from more distant configurations. Following Cohn et al. [7], we fit a Gaussian distribution to the region surrounding our query point. The parameters of the Gaussian distribution fit to the local region are derived below, where x is the query point whose classification we are interested in and x_i,y_i are members of the sets of input and output training data X, Y.

The means of the Gaussian are calculated as follows:

$$\mu_x = \frac{\sum_i (w(x,x_i)x_i)}{\sum_i w(x,x_i)} \qquad \mu_y = \frac{\sum_i (w(x,x_i)y_i)}{\sum_i w(x,x_i)}$$

The variances of the Gaussian are given by:

$$\sigma_x^2 = \frac{\sum_i (w(x,x_i)(x_i-\mu_x)^2)}{\sum_i w(x,x_i)}$$

$$\sigma_y^2 = \frac{\sum_i (w(x,x_i)(y_i-\mu_y)^2)}{\sum_i w(x,x_i)}$$

The covariance is:

$$\sigma_{xy} = \frac{\sum_i (w(x,x_i)(x_i-\mu_x)(y_i-\mu_y))}{\sum_i w(x,x_i)}$$

The conditional variance is calculated:

$$\sigma_{y|x}^2 = \sigma_y^2 \frac{\sigma_{xy}^2}{\sigma_x^2}$$

Using this parameterization, we can then calculate the expected value of the output $\hat{y}$. This value is the prediction for our query point x:

$$\hat{y} = \mu_y + \frac{\sigma_{xy}}{\sigma_x^2}(x-\mu_x)$$

The variance of this prediction ($\sigma_{\hat{y}}^2$) is:

$$\begin{aligned}\sigma_{\hat{y}}^2 \quad = \quad & \frac{\sigma_{y|x}^2}{(\sum_i w(x,x_i))^2}\left(\sum_i w(x,x_i)^2 + \right. \\ & \left. \frac{(x-\mu_x)^2}{\sigma_x^2}\sum_i w(x,x_i)^2 \frac{(x_i-\mu_x)^2}{\sigma_x^2}\right)\end{aligned}$$

So far we have shown how to use a collection of samples to construct an approximate model of configuration space. We now discuss how this model can be constructed incrementally by placing samples that maximally improve the model.

B. Sampling Strategy

A sampling-based motion planner has the ability to query any location in configuration space by invoking a collision checker. A sampling-based roadmap motion planner selects configurations to incorporate into a roadmap. The method of selection is the planner's *sampling strategy*. To be maximally efficient, the sampling strategy should carefully select samples to accelerate the computation of a solution path. The field of active learning examines similar situations in which a machine learning algorithm can select arbitrary training data. Active learning methods have been devised to accelerate learning through the selection of appropriate training data. Our active learning-based sampling strategy will select configurations that maximize the improvement of the model. For our purposes, maximal improvement in the model corresponds to minimizing the model's variance. Cohn et al. [7] provide an efficient method for determining the expected variance of the model. High model variance corresponds to unreliable predictions by the model, whereas in low variance means the model is reliable. By biasing sampling toward regions of which decrease the variance of the locally weighted regression model, sampling is directed to complex regions of configuration space [6].

The combination of locally weighted regression and active learning provide a sampling strategy for model-based motion planning. Whenever a configuration space sample is required by the motion planner, the sampling strategy examines the state the approximate model of configuration space. A configuration that minimizes the expected variance of the model is selected. The configuration is sampled and the resulting information is added to the approximate model of the configuration space. We call this sampling strategy *active sampling*.

An experimental validation of the proposed model based on locally weighted regression and the corresponding active learning-based sampling technique has shown that this type of model is capable of representing configuration space information efficiently and accurately [6].

C. Predictive Edge Validations

Edge validation is one of the most expensive phases of constructing a traditional roadmap [8] (cf. Figure IV). The approximate model presented above can be used to predict whether an edge is free or obstructed. A motion planner can use this information to determine if the computationally expensive examination of the edge using a collision checker is warranted. By avoiding unnecessary edge validations, the computational cost of motion planning can be reduced significantly.

To predict if an edge is free or obstructed using locally weighted regression it is necessary to calculate a new weight function, w' which measures contribution of a sample in the model to the prediction of an edge. This is determined by the distance to the nearest point on the edge. For some edge e, and a training point x_i, the function is given by:

$$w'(e, x_i) = e^{-k(\text{NearestPoint}(e,x_i)-x_i)^2}$$

The function w' is substituted for w in the equations for the parameters of the Gaussian detailed earlier in Section III-A. In order to calculate the prediction ($\hat{y}$), the distance between a point and the mean of the regressed Gaussian is needed. Again, we substitute the point on the edge nearest to the mean of the Gaussian ($\text{NearestPoint}(e, \mu_x) - \mu_x$)) and calculate the prediction of $\hat{y}$ as follows:

$$\hat{y} = \mu_y + \frac{\sigma_{xy}}{\sigma_x^2}(\text{NearestPoint}(e, \mu_x) - \mu_x).$$

Now that locally weighted regression can give predictions for a line, we might simply use a single prediction for an edge. However, this may result in false predictions for edges that are split between free and obstructed space. Because prediction averages over the entire edge, the resulting prediction is uncertain. If the prediction is uncertain, the edge is divided in half, and each half is recursively tested. If either half is predicted obstructed, the entire edge is predicted obstructed, otherwise the entire edge is predicted free. If the prediction is certain, the original prediction is returned.

D. A Model-Based Motion Planner

We now describe a model-based motion planner. The resulting motion planner differs from traditional sampling-based approaches in the following aspects: Before roadmap construction begins, an initial approximate model is constructed from a small number of configurations selected uniformly at random. These initial configurations are not added to the roadmap. Next, the algorithm improves the model using the model-based sampling strategy (Section III-B). Free samples as well as samples in collision are incorporated into the approximate model. This is an important distinction from the traditional PRM algorithm, which discards information from obstructed configurations. If the chosen configuration is free it is also incorporated into a *predictive roadmap*.

The predictive roadmap is another important distinction between our new planner and traditional PRM methods. For traditional PRM, more than three-quarters of the time is spent validating edges. This computational cost is often unnecessary. Many checked edges are actually redundant. They could be removed from the final roadmap without affecting the completeness of the roadmap. The model used in the proposed approach to motion planning can provide predictions about the state of unobserved edges (Section III-C). This enables an alternative to the expensive construction of a traditional roadmap: a predictive roadmap whose edges have been validated using the model rather than a collision checker. Since paths found in the predictive roadmap have only been predictively validated, a solution

```
PREDICTIVEMODELBASEDMOTIONPLAN
        (INIT, ITERATION, START, END) : PATH
    do init times
        Select a random configuration $x$
        Add $x$ to the initial data set $D$
    Construct a model $M$ from $D$
    do iteration times
        Use active sampling to determine configuration $x$
        Add $x$ to model $M$
        if $x$ is free according to model $M$
            Add $x$ to the roadmap $R$
            foreach $x_i$ in the set of its $n$ neighbors
                if path between $x$ and $x_i$ is free in $M$
                    Connect $x$ to $x_i$
    return EXTRACTPATH(START, END, $R$)

EXTRACTPATH(START, END, ROADMAP) : PATH
    do
        $p$ := DykstraPathPlan(start, end, roadmap)
        for each edge $e(x_i, x_j)$ in $p$
            if $e$ is in collision
                resample between $x_i$ and $x_j$
                    and add configurations to roadmap
            else
                mark $e$ validated
    while $p$ is not a valid path
    return $p$
```

Fig. 1. A predictive model-based motion planner

path obtained from the predictive roadmap still has to be verified for collisions. We refer to this process as path extraction.

During the process of path extraction, edges predicted as free by the model can be found to be obstructed when validated with a collision checker. In such a case the edge is removed from the predictive roadmap and repair is attempted. To repair an edge, configurations in the vicinity of the invalid edge are sampled and attempts are made to reconnect the endpoints of the invalid edge through these now configurations. The process is similar to those used by others [3], [17], [9]. If the repair process fails, roadmap construction is resumed until a new candidate path between start and goal is found. This process repeats until a path is found (see Figure 1).

The predictive roadmap algorithm marks a middle ground between multi-query motion planners and single-query motion planners. The predictive roadmap contains general but incomplete information relevant to the construction of any possible path. The verification and repair of edges in the roadmap as a result of a particular planning query focuses computation on those areas most relevant to the solution of that query. The predictive model-based motion planning approach is ideally suited for dynamic environments. In the absence of a query, computational resources are directed toward acquiring an approximate model of the entire configuration space. Since single-query approaches do not begin with this approximate information, this additional information provided to the predictive model-based motion planner is expected to outperform existing single-query methods.

The predictive model-based motion planner distinguishes itself from existing motion planners in several important aspects. Model-based motion planners use a more expressive underlying model than a roadmap to represent configuration space information. The information contained in this model is exploited to adapt the local sampling density to the complexity of the configuration space region and to make predictions about unexplored regions.

IV. Experimental Validation

The performance of the predictive model-based planner introduced in Section III-D is compared with traditional PRM [12] and a motion planner based on the hybrid bridge test [8]. To differentiate between the effect of the sampling strategy and the predictive roadmap, the model-based planner was tested with and without predictive roadmaps. In Figure IV, the model-based planner including active sampling but no predictive roadmap is labeled “Active”; the planner described in Section III-D is labeled “Predictive”.

We perform path planning experiments with two simulated arms with either nine or twelve DOF. The twelve degree of freedom version of the arm with its initial and final configuration are shown in Figure 2. Note that the final configuration of the robot is inside the most confined region of the workspace. We make the assumption that the corresponding configuration space region exhibits very high complexity. If the model-based planner can successfully plan for the experimental scenario, it is likely to have explored the entire configuration space and consequently will be able to answer subsequent queries in constant time. By choosing the experimental scenario in this manner, the difference between the multi-query approach to motion planning and the hybrid multi-query/single-query nature of predictive model-based motion planning is minimized. The experimental results given here therefore represent a fair comparison.

In our experimental evaluation the four algorithms run until a successful path between the start and goal positions is found. In the case of the predictive roadmap, the time to verify and repair (if necessary) the candidate path in the predictive roadmap was included in the overall time. This time is labeled “Path Extraction” in Figure IV, for the predictive model-based planner, the time labeled “Path Validation” corresponds to predictive validation in the model. The times given represent the average performance over ten runs of each algorithm.

From the results in Figure IV, it can be seen that active sampling is an improved sampling technique over bridge

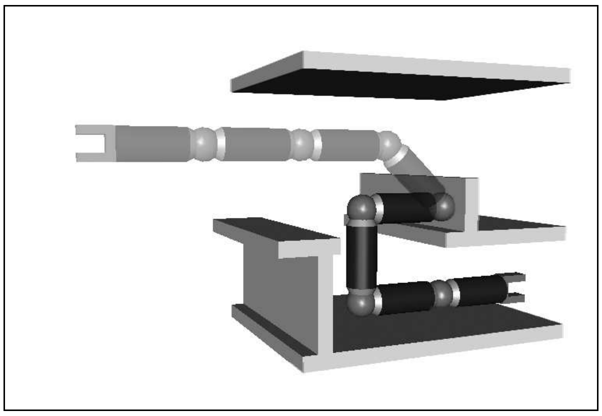


Fig. 2. The initial (transparent) and fi nal (solid) confi guration of a twelve degree of freedom arm in the experimental environment.

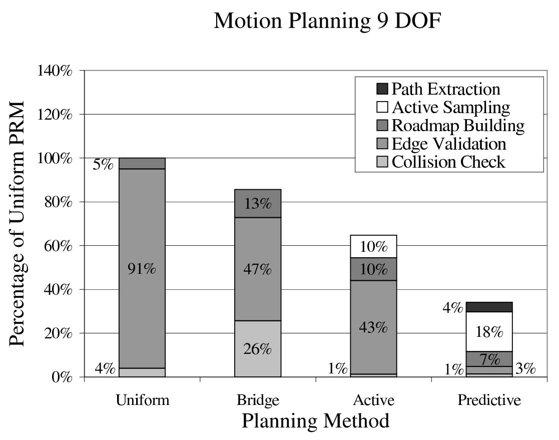
and uniform sampling. Further, the amount of improvement (around a 40% decrease in time) appears constant as the degree of freedom increases, suggesting that the performance may degrade gracefully for motion planning in very high-dimensional configuration spaces. Adding the predictive roadmap results in nearly a three times speed-up. Since the predictive roadmap does its edge checking in the model, the amount of time it spends checking edges is significantly less than traditional roadmap approaches. The predictive roadmap pays a price for the potential inaccuracy of its roadmap, using a third of its computational time to perform path verification and repair. This cost is offset by savings from predictive edge checking.

Lastly, it should be noted that the performance of PRM with hybrid bridge sampling degrades because it falsely identifies many configurations as important and enters them into the roadmap. Since model-based motion planning is inherently adaptive to each environment, such problems do not arise.

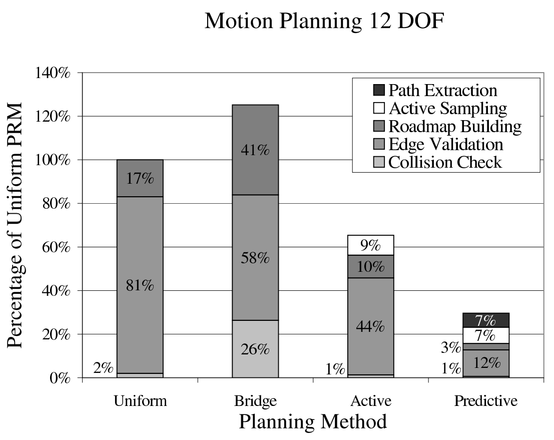
V. Conclusion

We have proposed a novel sampling-based motion planning approach. Its main distinguishing feature is the underlying representation of configuration space information used for path planning. We refer to this representation as a model of configuration space. The model provides an approximate picture of configuration space that is used to direct further exploration.

Model-based motion planning adapts the sampling density to the local complexity of configuration space regions. As a result, computational resources are expended in proportion to the local difficulty of a configuration space region. This effectively addresses the narrow passage problem which has been the focus of much research over the past decade. The adaptation of the sampling density is achieved by incrementally sampling uncertain regions in the model. Every additional sample ensures maximum expected improvement for the model.



(a) Motion planning with nine degrees of freedom



(b) Motion planning with twelve degrees of freedom

Fig. 3. Time to find a successful path for a nine and twelve degree of freedom arm as a percentage of PRM with uniform sampling and the percentage of the time taken by each algorithmic component of sampling-based motion planning.

Model-based motion planning is capable of avoiding expensive configuration space exploration. The model can be efficiently queried for predictions about the state of an edge based on proximal observed samples. This generalization across samples significantly increases the information gained from each examination of configuration space and results in greater efficiency. These predictions give rise to the predictive roadmap, a middle ground between single and multi-query approaches which minimizes unnecessary (and expensive) edge checks.

Model-based motion planning represents a middle ground between multi- and single-query approaches. It rapidly builds an approximate representation of the entire configuration space which is refined using information provided by a specific query.

Experiments demonstrate the effectiveness of the proposed predictive model-based approach to path planning, relative to other sampling-based motion planning techniques.

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] R. Bohlin and L. E. Kavraki. Path planning using lazy PRM. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 521–528, San Francisco, USA, 2000.

[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. Model-based motion planning. Technical Report TR 04-32, Computer Science Department University of Massachusetts, 2004.

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

[8] 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.

[9] 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.

[10] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. In *Proceedings of the International Conference on Robotics and Automation*, volume 3, pages 2719–2726, 1997.

[11] D. Hsu and Z. Sun. Adaptive hybrid sampling for probabilistic roadmap planning. Technical Report TRA5/04, National University of Singapore, 2004.

[12] 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.

[13] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[14] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Iowa State University, 1998.

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

[16] 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.

[17] C. L. Nielsen and L. E. Kavraki. A two level fuzzy PRM for manipulation planning. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1716–1722, Takamatsu, Japan, 2000.

[18] J. H. Reif. Complexity of the mover's problem and generalizations. In *Proceedings of the Symposium on Foundations of Computer Science*, pages 421–427, 1979.

[19] T. Siméon, J.-P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *Journal of Advanced Robotics*, 14(6):477–494, 2000.

[20] A. G. Sukharev. Optimal strategies of the search for an extremum. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 11(4):119–137, 1971. Translated from Russian, *Zh. Vychisl. Mt. i Mat. Fiz.*, 11(4):910-924.