Single-Query Entropy-Guided Path Planning

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

Technical Report 04-76

September 2004

Abstract

Motion planning for robots with many degrees of freedom requires the exploration of an exponentially large configuration space. Single-query motion planners restrict exploration to regions of configuration space determined to be relevant to a particular planning query. The heuristics employed by existing single-query planners to estimate the relevance of a region, however, remain unchanged throughout the planning process. An incorrect estimate by the heuristic for a configuration space region will only be corrected by explicit exploration. As a result, unnecessary exploration is performed. In this paper we propose an alternative approach. We observe that every incremental sample improves the planner's understanding of configuration space. This improved understanding can be exploited to inform the single-query heuristic of a motion planner. We formalize the improvement in understanding by employing the notion of entropy from information theory and derive a principled method of configuration space exploration in the single-query setting. Experiments show that the proposed single-query entropy-guided motion planner outperforms existing single-query techniques.

1 Introduction

The general motion planning problem has been shown to be PSPACE-complete [5, 13]. In spite of this computational complexity, sampling-based motion planners are able to solve many practical problems in high-dimensional configuration spaces [9]. But even these planners cannot avoid the exploration of a configuration space of exponential size. During exploration, a model of free configuration space is constructed. In sampling-based motion planners this model is a graph, called roadmap, that captures free space connectivity. Once such a roadmap has been computed for the entire configuration space, any planning query can be answered efficiently.

To answer only a single motion planning query, the complete exploration of configuration space is not necessary. It suffices to build a model of free configuration space in regions relevant to the given query. This provides the motivation for single-query motion planners. These planners construct a partial model of free configuration space by biasing exploration toward regions determined to be relevant to the query. A number of such single-query planners have been proposed in the literature. They differ in their method of estimating the relevance of a particular region of configuration space (see Section 2).

A common feature of all heuristics for single-query motion planners presented in the literature is that they ignore the information obtained during the planning process. The estimated relevance of a region provided by these heuristics is only altered by its explicit examination. This is because existing heuristics estimate relevance of regions assuming they are free. In existing planners, this assumption can only be disproven by explicitly invoking a collision checker. In reality, the information captured by nearby samples taken during the motion planning process provides an indication of whether a particular region is free or not. This information currently is ignored by existing planners.

We present a single-query motion planner that estimates the utility of configuration space regions based on their relevance to a particular planning query *and* the probability that they are collision free. Both factors are determined based on information obtained incrementally throughout the planning process. Extending our previous work on entropy-guided motion planning [4], we exploit the notion of entropy reduction to derive an information theoretic heuristic for single-query motion planning. We first define a probability distribution over paths in the configuration space. This distribution is designed to have high entropy when little information about the single-query is available and minimum entropy when a solution has been determined. Configuration space exploration is guided by the heuristic of maximum entropy reduction, or equivalently, maximum information gain. This heuristic uses all available information to guide the planner's exploration. Each exploration step results in maximal progress toward discovering a solution path, given the available information at the time.

Empirical evidence shows that the proposed single-query entropy-guided motion planner outperforms other approaches to sampling-based single-query motion planning.

2 Related Work

The first probabilistic approach to single-query path planning was the LazyPRM algorithm [2]. LazyPRM initially samples the configuration space without performing collision checks. Samples are assumed to be free and are connected to their nearest neighbors by edges without verifying their validity. LazyPRM searches the resulting roadmap using the A* algorithm, which biases search toward the region of space surrounding a straight line path between the start and goal state. This heuristic remains unchanged, irrespective of acquired information indicating, for example, that a region is obstructed. When a candidate path in the roadmap is found, it is validated by testing all nodes and then all edges. If obstructed nodes or edges are found, they are removed from the graph and A* path search begins again. A multi-grid variant of LazyPRM [1] discretizes the range of motion for each degree of freedom to simplify the configuration space. The granularity of the discretization is adapted until the motion planner can find a path.

FuzzyPRM [12], unlike LazyPRM, does not delay the examination of nodes in its graph, but does delay evaluation of edges. FuzzyPRM maintains an estimated probability value for each edge based on a distribution over the unchecked portion of the path. For a particular edge segment, this estimate exclusively depends on the length of the edge. Consequently, is remains constant throughout the planning process and is not updated based on information obtained by sampling. Candidate paths through configuration space are found using Dijkstra's algorithm. When a path is found, it is verified by examining edges in the order from longest (judged by FuzzyPRM to be least likely to be free of collision) to shortest.

Similar to the multi-grid extensions of LazyPRM is the single-query quasi-random grid approach (LazyQRM) [8]. Quasi-random grids establish a lattice of configurations spanning the configuration space. Since the structure of the grid is implicitly defined, the costly pre-sampling and graph construction required by LazyPRM and FuzzyPRM can be avoided. The A* algorithm, using Euclidean distance to the goal as its heuristic, is used for search within the grid. This heuristic is identical to the one employed by LazyPRM.

An adaptive approach to single-query path planning is presented in [15]. It features a meta-planner that incrementally plans between start and goal. At each planning attempt, the "best" algorithm, based upon the number of obstructed configurations and the algorithm's previous performance, is selected. When planning fails, any progress made by the planner is grafted onto the start and goal tree. This single-query motion planning techniques effectively combines a variety of planners with the goal of exploiting their strengths and weaknesses in configuration space regions with specific characteristics.

Rapidly-growing random trees (RRTs) [7] quickly explore the area between start and goal configurations by diffusing random trees of short edges through configuration space. The employed heuristic is termed the Voronoi bias [10]; it guides sampling toward unexplored regions of configuration space. Again, this planner does not take into account information obtained during the planning process to alter this heuristic, i.e., the heuristic does not differentiate between different open regions based on whether access to them is blocked or not. Single-query planning with expansive spaces [6] is similar to RRT planners; they also use diffusion from start and goal configurations to find a solution to the planning problem.

Others have suggested the use of information theory for motion planning. Yu and Gupta [16] use reduction in entropy to guide the visual exploration of workspace for an eye-in-hand system. The notion of entropy is used to determine placements for the camera that are expected to provide maximum information about the workspace. By iterating the process of placing the camera based on this criteria, visual exploration can proceed efficiently. In our previous work, the notions of entropy has been used to successfully guide sampling in the construction of multi-query probabilistic roadmaps [3].

3 Single-Query Entropy-Guided Motion Planning

We propose a novel single-query motion planning approach based on an information theoretic framework. The configuration space exploration performed by this planner is influenced by *a)* the relevance of a configuration space region for finding a solution to the given planning problem, and *b)* the probability that this region is free of collision and thus can be part of a potential solution.

The relevance of a region is estimated based on whether or not a potential solution path traverses it. A potential solution path is a path that, given the current information about the configuration space, is not known to be obstructed. The path is found based on the current representation of configuration space. Since this representation is updated with each sample, all information available at a particular point in time is taken into account. This is an important distinction to previous single-query motion planners.

In addition to the binary criterion for potential solution paths (a path can either be shown to be invalid, or is assumed to be valid), we propose to estimate the probability of a particular solution path to be valid based on the information available in the configuration space representation. This information can be used to guide configuration space exploration toward regions most likely to contain a solution to the planning problem.

These two criteria are used in an information theoretic exploration scheme based on entropy reduction (or information gain). The derivation of information gain in this setting will be given in Section 3.1. In Section 3.3 we show how information gain can be used to guide exploration of configuration space in a concrete implementation.

3.1 Information Gain in Single-Query Motion Planning

In this section we discuss the information theoretic background for the single-query motion planning method presented here. Information gain [14] is a formal representation of the reduction in uncertainty that results from some additional knowledge. It was originally proposed to formally model information transfer through electronic signals. In the case of sampling-based motion planning, additional knowledge is the observation that a configuration is obstructed or free. In prior work, we defined information gain for multi-query motion planning [3]. For single-query motion planning we must define expected information gain for the task of discovering a particular path.

Entropy is the measure of uncertainty of a probability distribution P over a domain D:

$$H(D) = -\sum_{d \in D} P(d) \log P(d)$$

Information gain is the reduction in the entropy of a distribution as a result of obtaining some information i:

$$IG(D|i) = H(D) - H(D|i)$$

A distribution that has low entropy when a path between start and goal has been found and high entropy otherwise allows information gain to be used to direct exploration. At each step, the motion planner operates to maximize information gain (and thus minimize entropy) of this distribution. Because of the design of the distribution, this results in maximal progress toward a solution to the specified path query.

For single-query motion planning we use a distribution over a set of possible paths A. Each member of this set $a \in A$ represents a path connecting the start and goal configurations. The probability assigned to each path a in this distribution is the probability that it will be the successful path returned by the motion planner. This probability is the combination of the probability that the path is free ($P_f(a)$) and the probability that this path will be examined by the motion planner prior to any other path which is free ($P_s(a)$). Since these probabilities are independent, the joint probability that the path is free and examined prior to any other free path is given by $P_p(a) = P_s(a) \cdot P_f(a)$. The probability, $P_s(a)$ is difficult to calculate exactly but it is proportional to the length of the path since the planner uses A* which searches for shortest paths.

$P_f(a)$ can be calculated as the product of the probability that it is constituent vertices $V(a)$ and edges $E(a)$ are free:

$$P_f(a) = \left(\prod_{v \in V(a)} P_f(v) \right) \left(\prod_{e \in E(a)} P_f(e) \right)$$

Edge and vertex probabilities are either the result of direct observation in the collision checker or estimated by the approximate model (Section 3.2). The entropy of this distribution is given by:

$$H(D) = -\sum_{a \in A} P_p(a) \log P_p(a)$$

Every exploration of configuration space results in obtaining of some new information i which pertains to the feasibility of the path a.

For each path a in A, there are two possible outcomes of learning i: a may be more likely to be free, or a may now be known to be obstructed. In each case, the information gain is given by the difference between the prior and current entropy. Most of the probabilities for the paths of the distribution will remain the same, only those paths that contain a vertex or edge related to i will be affected. Let A' be this set of all paths in A that contain paths affected by i .

First, consider the case where i results from an observation that something is free. In this case, the probability of each path that i pertains to increases slightly:

$$IG(D|i) = H(D) - H(D|i)$$

$$= -\sum_{a \in A'} P_p(a) \log P_p(a) - -\sum_{a \in A'} P_p(a|i) \log P_p(a|i)$$

When i results from an obstructed observation path $a \in A'$ that i pertains to, the probability of the path becomes zero. The information gained is:

$$\begin{aligned} & IG(D|i) = H(D) - H(D|i) \\ &= -\sum_{a \in A'} P_p(a) \log P_p(a) - -\sum_{a \in A'} P_p(a|i) \log P_p(a|i) \\ &= -\sum_{a \in A'} P_p(a) \log P_p(a) \end{aligned}$$

Expected information gain is given by:

$$\begin{aligned} < IG(D|i) > \;=\; & -\sum_{a \in A'} P_p(a) \log P_p(a) + \\ & P(i = \text{free}) \sum_{a \in A'} P_p(a|i) \log P_p(a|i) \end{aligned}$$

Observing that $\log P_p(a|i) \leq 0$ and that $P_p(a|i) \geq 0$ for any path a, we can see that for information pertaining to a set of paths A', the information gain from discovering an obstruction is greater than or equal to information gain for observing free space. Intuitively this can be seen by noting that observing a configuration is obstructed immediately eliminates the entire path, while observing a configuration is free only increases the probability the path is free.

In the following section we use this information theoretic analysis of single-query motion planning to develop a single-query entropy-guided motion planner.

3.2 Modeling Configuration Space

The derived formulation of information gain for single-query motion planning requires an estimate of the probability that previously unexplored configurations and edges are collision-free. We propose to use memory-based models from the machine learning literature [11] to provide such an estimate. Memory-based models are based on a collection of samples, much like the traditional roadmap in PRM planners. The particular model used in this paper estimates the state of an unobserved configuration by examining the set of nearby neighbors in the model. The majority state of the nearby neighbors determines the prediction about the unobserved query configuration is made. We have shown elsewhere [4], that memory-based models can build successful approximations of configuration space.

In addition to estimating the state of unexplored configuration space regions, memory-based models have a number of appealing characteristics for our purposes. First, adding data to the model takes constant time regardless of the size of the model. Second, querying the model is linear in the number of configurations used to construct it. Third and maybe most importantly, the model allows the incorporation of positive and negative samples. In traditional roadmaps, colliding samples are discarded, although they provide useful information about the state of the configuration space.

3.3 A Single-Query Entropy-Guided Motion Planner

We now present a single-query entropy-guided motion planner based on the formal definition of information gain presented in Section 3.1 and the configuration space model introduced in Section 3.2. To render this implementation practical, we have to discretize the set of all possible paths considered in the original derivation of information gain. This is accomplished using a roadmap, as in traditional sampling-based motion planners. The probability of a particular sample is free can be approximated using a memory-based model in addition to this roadmap. Since both the roadmap and the memory-based model are based on samples, no additional configuration space exploration is required for maintaining the memory-based model.

At the initial stage of motion planning, the single-query entropy-guided motion planner chooses a set of samples from which an initial roadmap and memory-based model of configuration space are constructed. A fraction of the samples are chosen uniformly at random, while the majority are chosen from the bounding box surrounding the start and goal configurations (reflecting the heuristic used in LazyPRM). To build the model, all sampled configurations are inspected by the collision checker to determine if they are free or obstructed. The initial roadmap is constructed from configurations which are found to be free, but without verifying the connecting edges. This is one important difference between the entropy-guided approach and LazyPRM [2]. The latter constructs an initial roadmap without examining any configurations. Another important distinction is that while LazyPRM samples quite densely and uses short edges to connect configurations, we sample sparsely and connect using longer edges.

Once the initial roadmap has been constructed, A* is used to find a candidate path between start and goal configurations. The choice of A* to search the roadmap and the particular cost metric used for edges, are derived from the information theoretic considerations in Section 3.1.

In contrast to previous uses of A* for path planning [1, 2, 8] that use edge length for edge cost, the cost used by entropy-guided planning is the product of edge length and the probability the edge is obstructed: $mboxCost(e) = k \cdot \text{Length}(e) \cdot P_f(e)$. The probability that an edge is obstructed is estimated using the memory-based model. This cost is designed to favor edges that are likely to be free ($P_f(e)$), while simultaneously maximizing exploration ($\text{Length}(e)$). The constant k is used to balance this trade-off.

This cost function in combination with the A* algorithm represent a practical way of maximizing $P_s(a)$ and $P_f(a)$, which in turn maximizes the information gain. $P_s(a)$ is maximized because short paths are favored by the A* search, and $P_f(a)$ is maximized because paths which are likely to be free are favored. Note that the estimate of $P_f(a)$ provided by the memory-based model is updated as configuration space exploration proceeds and observes the state of additional configurations. In addition, as edges are invalidated, they are removed from the roadmap. Both these factors allow the heuristic used to guide exploration to effectively incorporate the information obtained about the configuration space during the process of motion planning.

Once a candidate path is found, the algorithm begins by examining each of its vertices. The expected information gain from examining a vertex is greater than for examining an edge since the set of paths A' affected by gaining information about a vertex is greater than the set of paths affected by gaining information about an edge. Because observing an obstructed vertex provides more information than observing a free vertex (see Section 3.1), the vertices are examined in order according to their probability of obstruction. If any vertex is obstructed, examination of the candidate path stops and search for a new candidate path resumes.

Once all vertices in the candidate path are verified, the edges of the candidate path are examined. Again the edges are examined in order by their probability of obstruction. If an edge is found to be obstructed it is removed and search for a new candidate path resumes. If all edges are found to be free, the path is the solution.

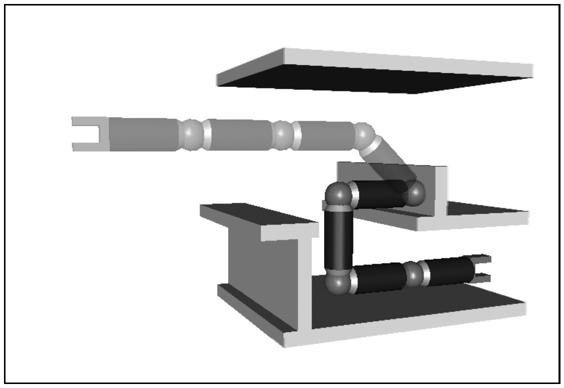


Figure 1: The initial (transparent) and final (solid) configuration of a twelve degree of freedom arm in the experimental environment.

If a candidate path between start and goal cannot be found in the roadmap, it is necessary to enhance the roadmap to introduce new candidate paths. The resampling used in our planner is similar to that of LazyPRM. We resample configurations that are near to obstructed edges that connect valid configurations in the roadmap. Unlike LazyPRM, we filter the configurations that we resample through the approximate model of configuration space. If a resampled configuration is likely to be obstructed, we do not attempt to add it into the roadmap. The planner learns from experience and avoids resampling the same invalid regions of configuration space. Once resampling is performed, the search for a candidate path continues using the augmented roadmap.

4 Experiments

To validate the entropy-guided approach to single-query motion planning we perform experiments with an implementation of the single-query entropy-guided planner described in section 3.3). The performance of the proposed planner is compared to the performance of traditional LazyPRM [1] and a single-query quasi-random planner (LazyQRM) [8]. Initial parameters for these two algorithms are set based upon descriptions in the respective papers.

To compare the algorithms we measure the number individual calls to the collision checker, the number of calls to validate an edge and the total overall time to find a path. For the entropy-guided planner, the number of collision checks used to construct the initial roadmap and model is included in the total number of collision checks.

Experiments were run for an arm with six, nine, and twelve degrees of freedom. The twelve degree of freedom arm is shown in Figure 1. The six (nine, twelve) degree-of-freedom arm consist of three (three, four) links connected by joints with two (three, three) degrees of freedom. The workspace for all of the arms is the same and is shown in Figure 1. Each algorithm runs ten times with ten different path queries. Each path query consisted of a random starting location in the vicinity of the straight configuration shown in Figure 1 and a goal configuration with the end effector inside the constrained location in workspace (also pictured in Figure 1).

The results of the experiments are shown in Figure 2. It can be seen that the single-query entropy-guided planner outperforms the other two planners. The LazyQRM planner fails to complete for either the nine or twelve degree-of-freedom robot. It consumes all available memory and exits on a Pentium 4, 3.2Ghz

Algorithm	Success	Collision Checks	Edge Checks	Runtime
Entropy Guided	12%	1385.6	26.1	3.7
LazyPRM	50%	1575.5	19.25	5.0
LazyQRM	0%	N/A	N/A	N/A

Table 1: Percentage of successful motion plans for the 12-DOF robot

with 1 gigabyte of RAM. This is indicative of the fact that LazyQRM's grid grows exponentially in the dimensionality of the configuration space.

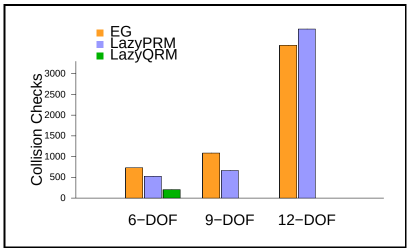
It is important to note that in the twelve degree of freedom world neither the LazyPRM nor the entropy-guided approach could reliably find a path. The LazyPRM planner successfully found a path 50% of the time and the entropy-guided planner found a path 75% of the time. These results are summarized in Table 1. When LazyPRM is successful, it is because it has selected beneficial placements for its initial roadmap. Entropy-Guided motion planning is less reliant on receiving a good initial roadmap and can find solutions more often. The data given for LazyPRM is the averaged over all successful planning attempts. The execution time given for entropy-guided planning is the average time of the same number of experiments as for the LazyPRM; the slowest experiments were discarded, as they solve motion planning problems that LazyPRM was unable to solve.

For the purposes of practical motion planning we apply a time cut-off to each algorithm. Any path planning attempt that lasted longer than thirty seconds is halted and path planning restarts from the beginning. The collision and edge checks as well as the runtime are all accumulated until a successful motion plan can be determined. The graphs in Figure 2 indicate that the entropy-guided approach leads to better runtime for all three problems. The greater number of collision checks in six and nine degrees of freedom are from the checks used to build the initial roadmap and model. This constant cost becomes insignificant for motion planning in higher dimensions, as seen with the twelve degree of freedom arm. It is important to note that individual collision checks require an order of magnitude less computation than edge checks, so the slight difference in the number of collision checks has much less of an effect on runtime than the number of edge checks. Additionally, single-query entropy-guided motion planning is biased toward checking edges which are likely to be obstructed, while LazyPRM is biased toward edges likely to be free. Obstructed edges are generally less computationally difficult to check than free edges, resulting in further performance gains.

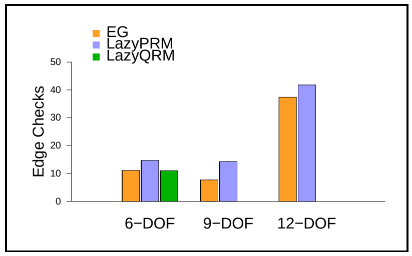
5 Conclusions

We present a novel single-query motion planner based on an information theoretic framework. Information theory provides a principled way of guiding configuration space exploration to maximize progress toward the computation of a solution to the given planning problem. This is accomplished by designing a practical planner capable of sampling those regions of configuration space that provide maximum expected information gain—or maximum expected entropy reduction—at each step of the planning process. Empirical results show that the proposed single-query entropy-guided motion planner outperforms other single-query motion planners presented in the literature.

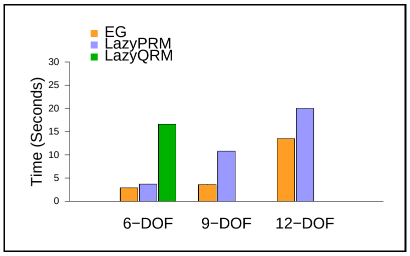
The performance improvements realized by the proposed planner can be attributed to two factors. First, at every point during the planning process, the information obtained by previously placed samples is used to guide the process of future exploration. In contrast, existing single-query methods rely on a heuristic that does not take into account this information. Second, the proposed method reasons about the probability



(a) Average collision checks



(b) Average edge checks



(c) Average runtimes

Figure 2: Experimental results for motion planning in 6, 9 and 12 degree configuration spaces

of potential solution paths based on all available information about configuration space. Previous planners estimate this probability without considering all available information. These factors are captured in an information theoretic framework that allows the proposed planning method to make maximum progress toward finding a solution, given the available information about configuration space.

We have also demonstrated the use of incrementally constructed, memory-based models in motion planning. These models augment the prevalent roadmaps and provide estimates of the state of configurations that have not been observed, based upon known nearby samples. They are also capable of using information from colliding samples, which are discarded by other sampling-based motion planning techniques. In the proposed planner these models are used to estimate the probability of potential solution paths to be free of collision.

References

[1] R. Bohlin. Path planning in practive: Lazy evaluation on a multi-resolution grid. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 49–54, Maui, USA, 2001.

[2] 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.

[3] 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.

[4] B. Burns and O. Brock. Model-based motion planning. Technical Report TR 04-32, Computer Science Department, University of Massachusetts Amherst, 2004.

[5] J. E. Hopcroft and G. Wilfong. Reducing multiple object motion planning to graph searching. Technical Report TR 84-616, Cornell University, 1984.

[6] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *International Journal of Computational Geometry and Applications*, 9(4):495–512, 1999.

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

[8] S. M. LaValle and M. S. Branicky. On the relationship between classical grid search and probabilistic roadmaps. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, Nice, France, 2002.

[9] S. R. Lindemann and S. M. LaValle. Current issues in sampling-based motion planning. In *Proceedings of the International Symposium of Robotics Research*, 2003.

[10] S. R. Lindemann and S. M. LaValle. Incrementally reducing dispersion by increasing Voronoi bias in RRTs. In *Proceedings of the International Conference on Robotics and Automation*, New Orleans, USA, April 2004.

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

[12] 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.

[13] 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.

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

[15] D. Vallejo, I. Remmler, and N. M. Amato. An adaptive framework for single shot motion planning: A self-tuning system for rigid and articulated robots. In *Proceedings of the International Conference on Robotics and Automation*, pages 21–26, Seoul, Korea, 2001.

[16] Y. Yu and K. Gupta. An information theoretic approach to viewpoint planning for motion planning of eye-in-hand systems. In *Proceedings of the International Symposium on Industrial Robotics*, 2000.