Toward Optimal Sampling In the Space of Paths

Colin J. Green and Alonzo Kelly

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
cjgreen@ri.cmu.edu, alonzo@ri.cmu.edu

Summary. While spatial sampling of points has already received much attention, the motion planning problem can also be viewed as a process which samples the function space of paths. We define a search space to be a set of candidate paths and consider the problem of designing a search space which is most likely to produce a solution given a probabilistic representation of all possible environments. We introduce the concept of relative completeness which is the prior probability, before the environment is specified, of producing a solution path in a bounded amount of computation. We show how this probability is related to the mutual separation of the set of paths searched. The problem of producing an optimal set can be related to the maximum k-facility dispersion problem which is known to be NP-hard. We propose a greedy algorithm for producing a good set of paths and demonstrate that it produces results with both low dispersion and high prior probability of success.

1 Introduction

The discretization and sampling of states and actions has a significant impact on motion planner performance[12]. Motion planning has been concerned with optimality of sequences of primatives from the very beginning[5] and this topic is still of interest today[7].

Independent of optimality, the question of whether an existing solution will be found is typically answered in terms of resolution completeness for deterministically sampled approaches such as [2] and probabilistic completeness for randomly sampled approaches such as [9][11]. These guarantees describe a search process which incrementally discovers, in progressively finer detail, the topology of a prespecified environment in an unbounded amount of time.

In a sensor-based real-time replanning context, the environment is partially unknown and it can only be ascertained by moving through it. It is often desireable to move continuously, so motion planners must find a solution in limited time in order to be responsive to continuously updated sensor data. In this limited time context, it may be impossible to find a solution, even if one exists, for reasons of insufficient computing resources. For continuous motion, the lack of a solution may imply an inevitable collision [6] [10] so safety can be a valid concern.

Therefore, an equally relevant completeness question is the *likelihood* that an existing solution will be found with the limited computation time available. That question can be framed in terms of an unknown environment whose properties are known only probabilistically. We define the *relative completeness* of a set of paths as the probability of finding a solution to a motion planning query in a fixed period of time (or equivalent amount of computation). Maximizing this probability can be viewed as a search space design problem and it is the main concern of this paper.

Many planning algorithms choose to discretize actions (controls, inputs) in order to reduce computation while inherently respecting differential motion constraints. An appropriate choice of primitive motions can lead to benefits with regard to density of the reachable set[3], robustness[8], and optimality[13]. The question of which actions to use is therefore a related search space design question because inputs determine the path followed.

In Sect. 2 we address the theoretical relationship between relative completeness of a set of paths and their mutual separation. Sect. 3 describes an algorithm which generates a set of paths whose mutual separation is favorable. Finally Sect. 4 presents the results of simulations which verify the above relationship and show that it has an impact on performance in motion planning problems.

2 Obstacle Correlation and Path Dispersion

The design of many motion planning algorithms depends on choices for how to discretize states and actions. The process of making these choices is a search space design activity that can be addressed somewhat independently of the process used to conduct the search. We will show that all search spaces are not created equal. In particular, this section shows that the finite size of obstacle regions and/or the spatial correlation of obstacles in space has implications to optimal search space design.

2.1 Spatial Correlation of Obstacles

While analog cost fields are sometimes used to represent the environment, we will adopt the popular abstraction of partitioning it into two subsets: obstacles and free space. Consider a region of the plane containing obstacles. Let $o(x, y)$, defined over $\mathbb{R}^2$, denote the joint probability density that the point (x, y) is inside an obstacle. Hence $o(x, y)dxdy$ is the probability that a differential region at (x, y) is contained in an obstacle.

Consider a subregion $R \subseteq \mathbb{R}^2$ and let $C(R)$ denote the event that some part of R contains some part of an obstacle, in which case we say it is *in collision.* Let $P[C(R)]$ denote the probability of the event $C(R)$. By the axioms of probability, regardless of the magnitude or the spatial correlation characteristics of $P[C(R)]$, adding more points to R cannot decrease the probability of collision:

$$P[C(R \cup dR)] \geq P[C(R)] \tag{1}$$

Consider next two nonempty differential regions dR_1 and dR_2 (Fig. 1a) located arbitrarily in $\mathbb{R}^2$. Let them be so small that $o(x, y)$ may be considered to be uniform everywhere inside and in a small surrounding neighborhood. The distance between them is defined as the distance between their centers. Consider the question of how obstacles in one region may affect those in the other. By the definition of conditional probability:

$$P[C(dR_1)|C(dR_2)] = \frac{P[C(dR_1) \wedge C(dR_2)]}{P[C(dR_2)]} \tag{2}$$

Suppose knowledge of dR_2 being in collision implies a greater probability that dR_1 is also in collision:

$$P[C(dR_1)|C(dR_2)] > P[C(dR_1)] \tag{3}$$

Under these conditions, we say that the two events are *positively correlated* because the occurrence of one raises the probability of the other. Assuming that $P[C(dR_2)] \neq 0$, we have $P[C(dR_2)] > 0$. Substituting from Equation (2) and rearranging leads to:

$$P[C(dR_1) \wedge C(dR_2)] > P[C(dR_1)]P[C(dR_2)] \tag{4}$$

Hence, positive correlation raises the probability that *both* regions are in collision relative to what it would be otherwise. There are two related situations where such positive correlation may occur: overlap of the regions, and spatial correlation of $o(x, y)$ as outlined below.

If the regions overlap, (Fig. 1b) let the quantity $\alpha(dR_1, dR_2)$ denote the area of their region of overlap. Assume for the moment that $o(x, y)$ exhibits

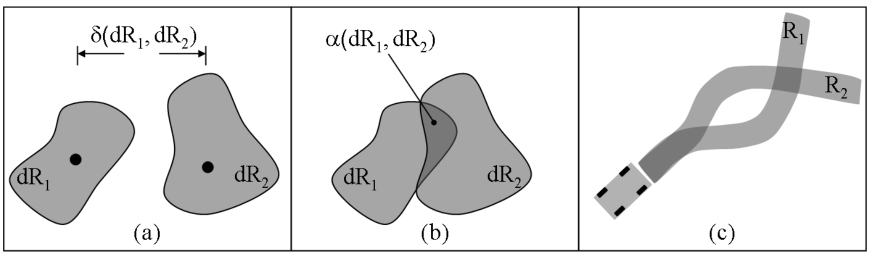


Fig. 1. Regions which may contain obstacles a) two separated differential regions b) two overlapping differential regions c) two finite regions (swept volumes) that overlap.

no spatial correlation and consider variations that allow the regions to deform while preserving their areas so that the area of overlap changes. An obstacle point in the region of overlap is in both regions. By Equation (1), for such variations, we can conclude that the probability that both regions contain an obstacle is monotone in α:

$$\frac{d}{d\alpha} P[C(dR_1) \wedge C(dR_2)] > 0$$

Likewise, spatial correlation of $o(x, y)$ may occur, for example, when obstacles tend to occur in contiguous regions. Consider allowing the regions above to move slightly while retaining their shapes. Let $\delta(dR_1, dR_2)$ denote the distance between them. This distance can be spanned by all obstacles whose width along the line between them exceeds the gap. For convex obstacles, and any distribution of obstacle sizes, or more generally for a spatial correlation which decreases monotonically with distance, there are fewer obstacles able to do this as δ increases. We conclude that the probability that both regions contain an obstacle point is monotone in δ:

$$\frac{d}{d\delta} P[C(dR_1) \wedge C(dR_2)] \leq 0$$

For two finite regions R_1 and R_2, $P[C(R_1)|C(R_2)]$ depends on the influence of every element of R_2 on every element of R_1. For finite sets, the notion of distance between the two sets is more difficult to define, particularly if it must satisfy the triangle inequality. Note however, that if two compact separated regions are further separated along the line between their centroids, the individual distances between every subregion of R_1 and every subregion of R_2 all increase, so we would still expect the probability of an obstacle spanning both regions to decrease with their separation.

2.2 Probability Of Solving A Query

The above analysis is useful for comparing different search spaces in motion planning. Let a search space be defined simply as a set of paths, whether encoded explicitly or generated during a search process that sequences states or actions. As a trivial motivating case, suppose a motion planner were able to search only two paths in the available time. Consider the swept volumes that would be occupied by a vehicle if it traversed each of these two paths and let R_1 and R_2 represent these volumes (Fig. 1c). Based on the above analysis, it would be a poor choice if these swept volumes overlapped unnecessarily or were even unnecessarily close to each other. If one were in collision, the other would also be highly likely to be in collision. Furthermore, if one were not in collision, the other is likely to be a redundant solution whose presence is probabilistically of little value from the perspective of completeness – because only one safe path is required.

The *relative completeness* of a set of paths is the prior probability that at least one of them will not intersect an obstacle. If the size of the set is related to some bounded amount of computation, this quantity provides the prior probability of finding a solution to a motion planning problem within the specified computational bound.

Relative completeness is a prior probability because it is computed over the set of all possible environments encoded in the joint obstacle density $o(x, y)$. It is not conditioned on the specification of a particular planner but is rather a property of a set of paths (i.e. a search space) independent of how it is represented or searched.

It was argued above that the probability that one of a pair of paths is not in collision increases with path separation. This implies that the relative completeness of a set of n paths is related to how they all are mutually separated (from each other in some holistic sense). In order to formalize this notion, we turn now to the concept of dispersion of a set of points.

2.3 Path Dispersion

The dispersion[14] of a finite set P of points is given by

$$d(P; X) = \sup_{x \in X} \min_{p \in P} d(x, p)$$

for the metric space (X, d) consisting of a set of points in X and an associated metric $d(x, y)$ which defines the distance between all pairs of points x and y. In our case X will be a finite set of paths and P will be a search space constituting some subset of X. Intuitively, dispersion computes the size of the largest sphere centered at a point in X which does not contain a point in P. This concept will be used to quantify the degree of mutual separation of a set of candidate paths. Note that, counterintuitively, the definition is such that lower dispersion implies a more disperse set as the word “disperse” is used in everyday communication.

Our intuitive notion of path separation suggests that the area between two paths [$AreaBetween(R_1, R_2)$] behaves roughly correctly. As it decreases toward zero, the swept volumes of two paths approach each other [$\delta(R_1, R_2)$ decreases] and then begin to overlap [$\alpha(R_1, R_2)$ increases] until they overlap completely and their separation becomes zero.

It is not clear that the conditional probability $P[C(R_1)|C(R_2)]$ is necessarily monotone in this distance for finite regions of arbitrary shape. Nor is it clear that it satisfies the triangle inequality required of a metric space. In any case, we will use this notion of distance as a computational expedient in the remainder of this paper and note that our results rest more fundamentally on probabilistic concepts which require no concept of distance between finite regions. Given this definition for the distance between two paths, and the concept of dispersion, we are now in a position to measure the mutual separation of a set of paths and to choose those sets which are most separated.

3 Search Space Selection Algorithm

This section presents an off-line algorithm for generating a mutually separated set of paths. In order to conveniently exhibit the relationship between dispersion and mutual separation, the approach used is to find a good subset P of an explicit and much larger set of paths X.

3.1 Path Generation

For this work we consider two different classes of search spaces, useful for two different classes of motion planners. All paths in a *goal directed* search space (identified later by subscript GD) start at the same start location and terminate at the same goal region. All paths in a *depth limited* search space (identified later by subscript DL) start at the same start and radiate arbitrarily outward from there.

A goal directed search space is suitable for analysing the standard motion planning problem where it is possible to search all the way to the goal in the available time. A limited depth search space is suitable for analysing the case when this is not so. The latter case occurs in sensor based obstacle avoidance when the goal is relatively far from the present position or when the combination of maps and sensor data cannot be trusted beyond some maximum distance. Under these circumstances, the paths which radiate away from the direction to the goal are provided in case this direction is blocked by an obstacle.

The set of paths X is produced by elaborating a reachability tree for a simple vehicle model incorporating limits on curvature and curvature rate. Any set of discrete inputs could be used in combination with any vehicle model but a large number of Dubins car-like motions[5] were chosen for simplicity. At each level in the tree a hard left, hard right, and a set of lower curvature turns in between are used to expand the leaves of the tree into successor states. For X_{GD}, paths which achieve a goal position (within a threshold) are included in the set, and all other paths are pruned. For X_{DL}, the tree is used as is. Example sets are shown in Fig. 2.

3.2 Separation Based Greedy Algorithm

The problem of finding the subset P of maximum dispersion is the "maximum k-facility dispersion" problem, an optimization problem which is known to be NP hard[1]. Given a set of facilities X, the problem is to find a subset P with $|P| = n$ which maximizes the minimum distance between any two facilities.

Given the difficulty of finding a true optimum, we settle on a simple algorithm, which attempts to greedily minimize dispersion. It is used to generate a set $\widehat{P^*}$, an approximation of some optimal set of paths P^*. $\widehat{P^*}$ is seeded with either the minimum length path to the goal (for X_{GD}), or the zero curvature path (for X_{DL}). Next, paths from X are added to $\widehat{P^*}$ one by one such

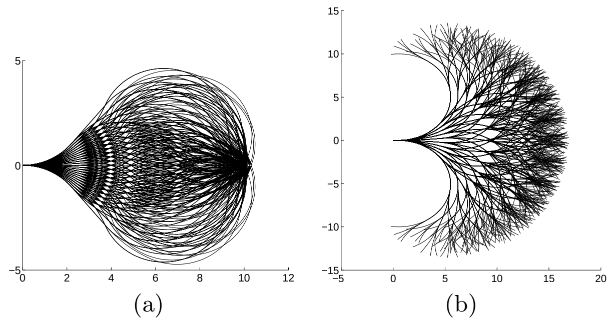


Fig. 2. Example sets of paths. a) X_{GD}, contains 667 paths. b) X_{DL}, contains 640 paths.

that each has maximal distance between it and its closest neighbor already contained in $\widehat{P^*}$ or $\arg\max_{x \in X} \min_{p \in \widehat{P^*}} d(x, p)$. This has the effect of filling the largest hole in function space at each step.

It is important to distinguish the off-line search over function spaces, presented above, from the on-line search of a set of paths that is conducted by a motion planner. In order to enforce the condition of limited computation during the on-line search, the total cumulative length of all paths in $\widehat{P^*}$ is fixed. Paths are added to $\widehat{P^*}$ until the set accumulates the specified total path length.

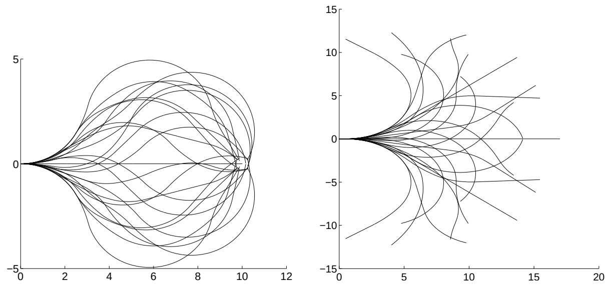
Later results will require a capacity to compute billions of distances between two paths. For efficiency, the area between two paths is approximated as the sum of the distances between pairs of points at equal path length along each path times the distance between subsequent points. 5–10 point pairs are used in this computation. Since this is an approximation of area its units are in m^2 and dispersion will refer to the area between two paths.

The incremental nature of this algorithm provides the additional benefit that the sequence of paths in a set of size n encodes all sets of size $\leq n$. Specifically, for a set of size n, $\forall k \leq n$, the set of size k which this algorithm would generate is simply the first k paths in the set. This property is very useful in the context of varying computing time budgets which might result, for example, from varying vehicle speeds.

For the X_{GD} and X_{DL} shown in Fig. 2, the resulting $\widehat{P^*_{GD}}$ is shown in Fig. 3a, and the resulting $\widehat{P^*_{DL}}$ is shown in Fig. 3b.

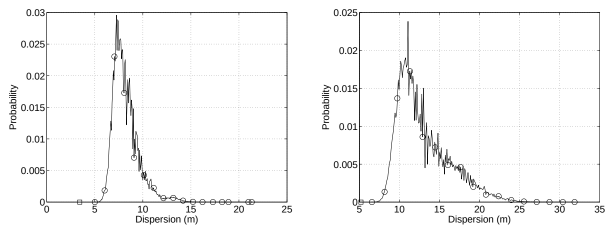
3.3 Search Space Generation

This section serves the dual purpose of describing aspects of our experimental setup as well as demonstrating the quality of the $\widehat{P^*}$ produced by the above algorithm. Around 400,000 P's were generated by randomly selecting paths from each of X_{GD} and X_{DL} and the dispersion was computed for each



(a) (b)

Fig. 3. The sets a) $\widehat{P_{GD}^*}$ and b) $\widehat{P_{DL}^*}$ which approximate the optimal set P^*. Both sets contain 25 paths.



(a) (b)

Fig. 4. A plot of the probability distribution over dispersions computed from a)400,000 random P_{GD} and b)400,000 P_{DL} sets of paths. The circles represents the uniformly sampled dispersion that will be used in the simulations and the square (bottom, far left in each graph) represents the dispersion of $\widehat{P^*}$.

set. Each set P contains a constant total path length and around 100 paths. Figure 4 shows the probability distribution over dispersion for the randomly generated P's, as well as the dispersion for $\widehat{P^*}$. As shown, $\widehat{P_{GD}^*}$ has a 30% lower dispersion than the best of the randomly generated P_{GD}'s and $\widehat{P_{DL}^*}$ has a 20% lower dispersion than the best of the randomly generated P_{DL}'s. Again, because of the definition of dispersion, lower dispersion implies a more disperse set.

In order to generate trajectory sets with higher dispersion, P_{DL}s were generated by using a "hole protection" algorithm. This algorithm works by selecting a trajectory from the set X, and then removing all trajectories from X with are within a distance d of that selected trajectory. The greedy algorithm described earlier is then used to generate the set P from this modified X guaranteeing that the dispersion will be at least d. This hole is selected in a way to ensure the straight trajectory is never removed from X because removing it would lead to an unusally ineffective search space.

4 Results

This section empirically verifies the relationship between relative completeness and path dispersion and it shows how it impacts performance in two mobile robot motion planning applications.

4.1 Relating Path Dispersion and Relative Completeness

The first set of tests are concerned with the relative completeness of a random sample of P_{GD} search spaces and the $\widehat{P^*_{GD}}$ generated above. For each of 17 values of dispersion, 7 sets P_{GD} are selected for testing at or very near that dispersion. Each is tested in 1,250,000 randomly generated obstacle fields to determine its relative completeness. Obstacle fields consist of a number of circular obstacles of random size (drawn from a Normal distribution with a lower bound) which are placed in a simulated environment at random positions (drawn from a uniform distribution). For any set P_{GD} placed in this environment, if at least one path is obstacle free, the set is considered to have succeeded in that particular obstacle field. Results are shown in Fig. 5. Error bounds provide the spread from minimum to maximum probability of success for the 7 sets tested at each dispersion. $\widehat{P^*_{GD}}$ has no error bounds since no other P_{GD}'s were found with dispersion close to it.

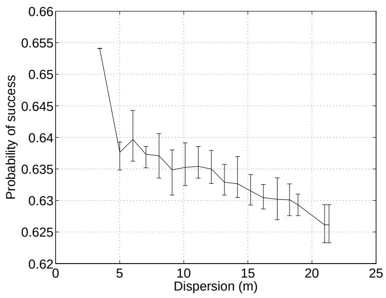


Fig. 5. The results from the simulations of $\widehat{P^*_{GD}}$ and various P_{GD}'s in random worlds. The error bars in the graph represent the min and max values observed for 7 different P's with very similar dispersions. The data point furthest to the left represents the simulation of $\widehat{P^*_{GD}}$

The point of this experiment was to examine the relationship between dispersion and relative completeness, and the results do follow the expected trend of decreasing dispersion leading to higher relative completeness. It is also interesting that $\widehat{P^*_{GD}}$ (the lowest dispersion point in the figure) does not follow the preceding trend. Instead it performs better than expected based on its dispersion. This likely indicates that our dispersion metric is not completely capturing the behavior of the underlying probabilities.

4.2 Relating Path Dispersion and Obstacle Avoidance Competence

The second set of simulations are concerned with the P_{DL}s and $\widehat{P^*_{DL}}$ depth limited search spaces. In this simulation there is a goal that is 1km away from the vehicle start location. The world is random and populated with circular obstacles which have a randomly generated size, giving the world the feel of a forest or boulder field. Obstacle positions are drawn from a uniform distribution and obstacle size is drawn from a bounded uniform distribution. There is a grid of points which represents the portion of the world known to the vehicle which is updated in each planning cycle with all obstacles within a radius of the vehicle.

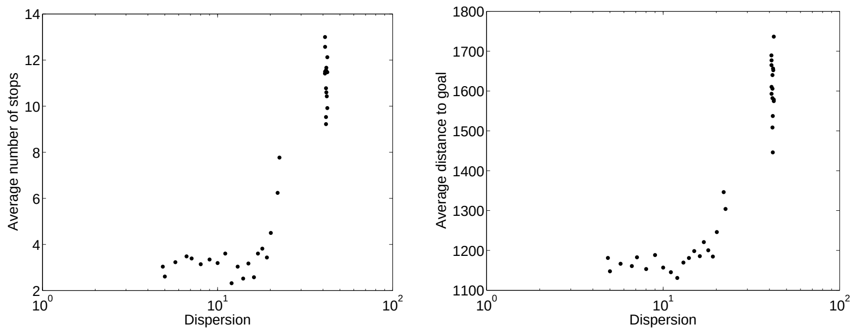
During each planning cycle, the vehicle follows the path which was sensed as obstacle free for the longest length in the last planning cycle. That selected path is followed for a distance corresponding to how far the vehicle would travel during the next planning cycle. In the case of a tie, the grid based distance to goal is computed from the end of each path, based on the portion of the world which is known to the vehicle, and the path with the lowest grid based distance to goal is selected. The paths are each of length 17 m. The method of evaluating a constant set of paths has been applied to the obstacle avoidance problem at least as far back as[4].

For simplicity, the vehicle cannot adjust its speed during a test. Instead, if the vehicle is in a position where it cannot avoid an obstacle at its fixed speed then the vehicle heading is changed to point along the grid based path to goal to simulate a stop and turn correction. These tests were performed for $\widehat{P^*_{DL}}$ and the P_{DL}'s. Approximately 100 runs were performed for each set (using different random worlds) and the average distance to goal was computed for each set. The results from this investigation are shown in Fig. 6. The results indicate that the final distance to goal followed an upward trend as the dispersion of the search space increases. Path dispersion relates to distance travelled because when the true optimal solution is near the largest hole in path space the best approximation available is relatively far away.

5 Conclusions and Future Work

Motion planner performance depends on both the quality of the search space and the quality of the algorithm which searches it. While it is intuitively clear that searching a large number of nearly identical paths is a poor approach to motion planning, it is not so clear how we should characterize an optimal search space. This paper has taken some steps toward defining these characteristics when search time is limited.

One practical motivation for this work is the case of an outdoor mobile robot which is trying to move as quickly as possible toward some distant goal. Our own field experiments for high speed outdoor vehicles on our PerceptOR and UPI programs have measured average speed of travel over very difficult



(a) (b)

Fig. 6. The results from a dynamic vehicle simulation with limited perception range for selected P_{DL}'s and $\widehat{P^*_{DL}}$. The data point furthest to the left represents the simulation of $\widehat{P^*_{DL}}$. a) Shows the average number of stops durring a run and b) shows the total distance to goal.

terrain as the most basic performance metric among many others. The time required to stop, reverse, and avoid an obstacle that was not avoided more gracefully had a substantial effect on our performance scores. In many cases, it was clear that an elegant solution might have been found if a more complete planner could be fit into the available computing resources. This observation was the original motivation for the work. All other things being equal, a planner less likely to stop would clearly be preferable in a competitive situation.

Although our simulation results are based on very simple planners this is because we are interested here in establishing the connection between relative completeness and mutual separation. During an online elaboration of a search space, most planners will focus the search based on what is observed during the search. The relative completeness of such planners can be improved by off-line optimization of the connectivity of the search space based on prior obstacle probabilities (to minimize correlation).

The trend toward the use of primitives in motion planning leads naturally to the question of which ones should be used and this work provides a degree of guidance in answering this question. We are in the process of extending this work to produce optimal symmetric state lattices for use in Mars rover motion planning.

Optimality of the solution is not considered in this paper although extensions to rank search spaces based on the expected optimality of a solution relative to continuum search are equally interesting. In this work we have identified an important characteristic in search space design. The online application of this idea to new and existing motion planning algorithms is likely to be an interesting research area.

6 Acknowledgments

This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) under contract “Unmanned Ground Combat Vehicle – PerceptOR Integration” (contract number MDA972-01-9-0005). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

1. G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.
2. J. Barraquand and J-C. Latombe. Robot motion planning: a distributed representation approach. *Int. J. Rob. Res.*, 10(6):628–649, 1991.
3. A. Bicchi, A. Marigo, and B. Piccoli. On the reachability of quantized control systems. *IEEE Trans. on Automatic Control*, 4(47):546–563, April 2002.
4. M. Daily, J. Harris, D. Keirsey, K. Olin, D. Payton, K. Reiser, J. Rosenblatt, D. Tseng, and V. Wong. Autonomous cross-country navigation with the ALV. In *Proceedings of the IEEE International Conference on Robotics and Automation, 1988*, volume 2, pages 718–726, 1988.
5. L. E. Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Amer. J. Math.*, 79:497–516, 1957.
6. Thierry Fraichard and Hajime Asama. Inevitable collision states - a step towards safer robots? *Advanced Robotics*, 18(10):1001–1024, 2004.
7. E. Frazzoli, M. A. Dahleh, and E. Feron. A maneuver-based hybrid control architecture for autonomous vehicle motion planning,. In *Software Enabled Control: Information Technology for Dynamical Systems*. 2003.
8. P. Jacobs and J. Canny. Robust motion planning for mobile robots. In *Proc. of the 1990 IEEE International Conference on Robotics and Automation*, pages 2–7, 1990.
9. Lydia Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. Technical Report CS-TR-94-1519, 1994.
10. S. LaValle and J. Ku. Randomized kinodynamic planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, 1999. To appear.
11. S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, May 2001.
12. S. R. Lindemann and S. M. LaValle. Current issues in sampling-based motion planning. In *Robotics Research: The Eleventh International Symposium*, pages 36–54. Springer-Verlag, Berlin, 2005.
13. S. Morgan and M. Branicky. Sampling-based planning for discrete spaces. In *Proc. IEEE/RSJ Intl. Conf. Intelligent Robots and Systems*, 2004.
14. H. Niederreiter. *Random number generation and quasi-Monte Carlo methods.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.