Sampling-Based Motion Planning: A Survey
Planificación de Movimientos Basada en Muestreo: Un Compendio

Abraham Sánchez López, René Zapata* and Maria A. Osorio Lama
Computer Science Department, Autonomous University of Puebla
14 Sur and San Claudio, CP 72570, Puebla, Pue. México
* LIRMM – UMR55606 CNRS, 161, rue Ada 34392 Montpellier Cedex 5, France
asanchez, asorio@cs.buap,mx, zapata@lirmm.fr

Article received on April 13, 2008; accepted on June 20, 2008

Abstract. Sampling-based motion approaches, like Probabilistic Roadmap Methods or those based on Rapidly-exploring Random Trees are giving good results in robot motion planning problems with many degrees of freedom. Following these approaches, several strategies have been proposed for biasing the sampling towards the most promising regions, thus improving the efficiency and allowing to cope with difficult motion planning problems.
The success of these planners in solving challenging problems can be explained by the fact that no explicit representation of the free configuration space is required. This paper reviews some of the most influential proposals and ideas, providing indications on their practical and theoretical implications. The contributions made by Mexican researchers in this field are also presented.
Keywords: Motion planning, probabilistic roadmaps, sampling-based motion planning, path planning, algorithms.

Resumen. Los enfoques de planificación de movimientos basados en muestreo, como los métodos de Roadmap Probabilista o aquellos basados en los Árboles Aleatorios de Exploración Rápida están dando buenos resultados en la planificación de movimientos de robots con muchos grados de libertad. Con estos enfoques, se han propuesto varias estrategias para predisponer el muestreo hacia las regiones más prometedoras, mejorando con esto la eficiencia y permitiendo la solución de problemas difíciles de planificación de movimientos. El éxito de estos planificadores en la solución de problemas desafiantes se puede explicar por el hecho de que no se requiere ninguna representación explícita del espacio de configuraciones libre.
Este artículo repasa algunas de las propuestas e ideas más influyentes en el área, proporcionando indicaciones de sus implicaciones teóricas y prácticas. También se presentan las contribuciones realizadas por los investigadores Mexicanos en este campo.
Palabras claves: Planificación de movimientos, roadmaps probabilistas, planificación de movimientos basada en muestreo, planificación de trayectorias, algoritmos.

1 Introduction

The existing industrial robot programming systems still have very limited motion planning capabilities. Motion planning is one of the components for the necessary autonomy of the robots in real contexts and it is also a fundamental issue in robot simulation software. We will restrict ourselves to motion planning for two and tree dimensional rigid bodies moving in static and known virtual environments.

The configuration of a robot is normally described by a number of variables. For mobile robots these ones are the position and orientation of the robot, i.e., 3 variables in the plane. For robot arms, these variables are the positions of the different joints of the robot. A motion for a robot can be considered as a path in the configuration space.

A typical motion planning problem asks for the computing of collision-free motion between two given placements of a given robot in an environment populated with obstacles. The problem is typically solved in *the configuration space C*, in which each placement (or configuration) of the robot is mapped as a point [Latombe, 1991]. *The free configuration space F* is the subset of the configuration space at which the robot does not intersect any obstacle. Figure 1 illustrates the basic motion planning problem. It shows the path of a point among polygonal obstacles in a two-dimensional workspace.

The robot can move from an initial to a goal configuration without intersecting an obstacle if the two configurations lie in the same connected component of *F*. Planning a collision-free path thus reduces connectivity and other topological questions in *F*.

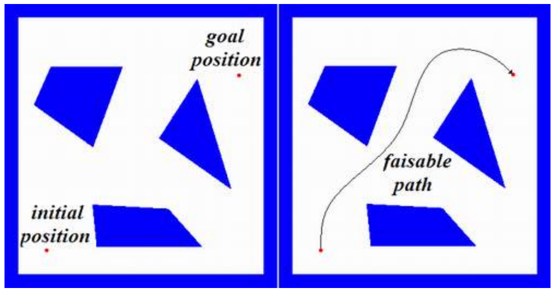


Fig. 1. The basic motion planning problem

When all degrees of freedom can be changed independently, e.g., like in a fully actuated arm, we talk about *holonomic motion planning*. In this case, the existence of a collision-free path is characterized by the existence of a connected component in *F*. Motion planning consists in building *F*, and in finding a path in its connected components. When the degrees of freedom of a robot system are not independent, e.g., a car that cannot rotate around its axis without also changing its position, we talk about nonholomic motion planning, In this case, any path in *F* does not necessarily correspond to a feasible one. Nonholonomic motion planning is much more difficult than holonomic motion planning and it is a fundamental issue for most types of mobile robots.

The dimension of *C* depends on the degrees of freedom (dof) of the robot, which can be quite high. Figure 2 shows the configuration space of a planar two-revolute joint manipulator arm. The configuration space of this robot is $S^1 \times S^1$, i.e., a torus in a three-dimensional Euclidean space. An exact computing of a high-dimensional configuration space is impractical [Barraquand and Latombe, 1991]. Therefore, techniques to compute an approximate representation of *F* are needed. Much of the difficulty in approximating *F* consists of understanding the topology and its simplifications.

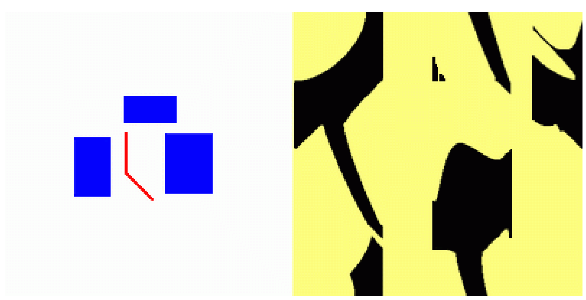


Fig. 2. A planar two-revolute joint manipulator arm

The history of motion planning is quite recent. The first works appeared in the late 60's and the active algorithmic development started in the 80's with the notion of configuration space [Lozano, 1983]. During these two decades, a very large number of techniques have been proposed. Latombe's book [Latombe, 1991] provides an excellent overview of the progress on motion planning until the early 90's. A couple of books [Laumond, 1998] and [Gupta and Polbil, 1998] that collect recent articles, provide a good overview and the essential references of the related works of the last decade. In the last few years two books that collect posterior theoretical advances and modern algorithms have been published [Choset et al, 2005; LaValle, 2006].

In 1979, Reif showed that path planning for a polyhedral robot among a finite set of polyhedral obstacles was PSPACE-hard [Reif, 1979]. Four years later, Schwartz and Sharir proposed a complete general-purpose path planning algorithm based on an algebraic decomposition of the configuration space of any fixed dimension *d*. When the space of collision-free placements is a set defined by *n* polynomial constraint of maximal degree *m*. a path can be computed by an algorithm whose time complexity is doubly exponential in *d* and polynomial in both *n* (geometrical complexity) and *m* (algebraic complexity) [Schwartz and Sharir, 1983]. Few years later, Reif improved this algorithm to a single exponential time algorithm. Canny found a PSPACE algorithm for the general motion planning problem and showed that it was PSPACE-complete [Canny, 1988], showing that exact planners have little chance of solving complicated problems. If the density of the obstacles in the environment is low, lower complexity bound exist [Van der Stappen et al, 1998]. Efficient heuristics have been proposed in the last years to tackle a large diversity of motion planning problems.

Several approaches were subsequently proposed aiming to overcome complexity and implementation inconveniences of exact methods. For this, some continuous quantities in the problem definition, such as object dimensions or configurations parameters, are discretized. Algorithms based on an approximated cell decomposition of the *F* are *resolution complete* (They are complete for a given discretization size.) [Latombe, 1991]. Other approaches use a grid to quantize *C* and perform a search process over this grid (see [Hwang and Ahuja, 1992] for an excellent overview in this topic). Efficient heuristic algorithms have been developed based on *the potential field approach* [Latombe, 1991; Hwang and Ahuja, 1992] to carry out this search. Such algorithms are able to perform very fast, but get easily trapped at local minima. The design of potential function is a critical point and difficult for high-dimensional spaces. These approaches using cells or grids are applicable in practice to mobile systems involving only a few variables. The number of cells or grids points becomes enormous for attaining acceptable resolution in high dimension. There are a large number of exact and non-heuristic algorithms with provable worst-case complexities for various versions of the motion planning problem, but they are not usually suitable for practical implementations [Latombe, 1991]. For systems with more than a few dofs, the exact algorithms become very inefficient in practice. These results suggest that approximate or heuristic methods must be used if practical implementations are needed.

In recent years, a number of motion planning algorithms have been successfully used to solve challenging motion planning problems. Examples include the randomized path planner (RPP) [Barraquand and Latombe, 1991], Ariadne's clew [Bessière et al, 1993], probabilistic roadmap planners (PRM) [Kavraki et al, 1996], and rapidly-exploring random trees (RRT) [LaValle and Kuffner, 2000]. Each of these methods can be seen as belonging to a field we call *sampling-based motion planning* [LaValle and Branicky, 2002]. These practical planners satisfy a weaker form of completeness, i.e., they use randomization to treat the high dimensionality of *C*. The term *probabilistically complete* was introduced to characterize these sampling-based algorithms, able to find a solution if sufficient running time is given. Motion planning has applications in other areas, such manufacturing design [Chang and Li, 1995], computer animation [Pettre et al, 2003], computational biology [Song and Amato, 2001], logistic and operations in large industrial installations [Siméon et al, 2001], CAD [Van Geem et al, 1998], etc. It is remarkable that in these applications PRM has also been used.

Computer science began as a discipline about 1965. Having established itself as a science, it is now blossoming and reaching out into application areas including graphics, animation, electronic prototyping, simulation, and robotics. Computer science is destined to play a major role in numerous application areas. The rapid development of robotics and the resulting need for computer scientists to be better trained in traditional mathematics, necessitate changes in computer science curricula.

The solution of problems in robotics is an excellent opportunity for the computer science scientists to contribute in this attractive area, and it is also an opportunity to offer efficient algorithmic solutions to solve difficult problems. This survey tries to show the importance that has computer science in the solution of difficult problems of the knowledge, as robotics.

The paper is organized as follows. Section II introduces the formal statement of the sampling-based approaches. Algorithms based on probabilistic roadmaps are also discussed in this section. The basic formulation, as well as the improvements are illustrated and discussed. Section III details the most important ingredients for the design of PRM-based motion planning algorithms and also provides the fundamental theoretical results concerning probabilistic convergence. The importance of the related issues to the choice of a specific algorithm is also mentioned. Section IV discusses the most important contributions made by Mexican researchers in the field of sampling-based motion planning. Finally, conclusions are detailed in Section V.

2 Sampling-based approaches

The aim of this section is not to make an exhaustive survey of all the available techniques, but a simple overview which gives an idea on the last tendencies in motion planning. We present a classification (partially inspired by [Lindemann and LaValle, 2005]) of some of the approaches which have been particularly well accepted in the robotics community. The main ideas of each approach are briefly described and we point to papers which provide full explanations.

Sampling-based algorithms developed in the last sixteen years, have demonstrated their efficacy for solving motion planning problems in high-dimensional spaces. They capture the connectivity of the collision-free regions of the *F* without requiring to explicitly compute this subset. Sampling-based approaches can be grouped in two main families: those using sampling techniques for constructing a roadmap in *F* [Kavraki et al, 1996] and those using sampling within incremental search methods for exploring *F* looking for a particular path [Barraquand and Latombe, 1991; Bessière et al, 1993; LaValle and Kuffner, 2000]. The choice mainly depends on the application. Roadmap methods (PRM) are more suitable when several motion planning queries involving the same mechanical system (One can use this name to denote a manipulator arm, a mobile robot or a free-flying object.) moving in a static environment must be solved. Computing time is spent in a pre-processing phase and then planning queries can be solved in real-time. They are called *multiple-query* methods, although some roadmap-based algorithms have been developed to efficiently solve particular queries, *single-query* methods [Bohlin and Kavraki, 2000], [Sánchez, 2003]. Single-query methods are in general faster since they need not preprocessing. However, as they focus on solving a particular problem, the processed information is less appropriate for later use.

2.1 The rise of the roadmap methods

In the 1980s, constructing a representation of configuration space-obstacles C_{obs}, either completely or in part, was the predominant approach to motion planning (see for example [Faverjon, 1989]). References to many combinatorial planners and a few early sampling-based ones can be found in Hwang and Ahuja's survey [Hwang and Ahuja, 1992]. Glimpses of sampling-based ideas can be seen even in Donald's work [Donald, 1987]; he placed a six-dimensional lattice over the configuration space and attempted to find a connected sequence of lattice points. Lacking the collision detection abstraction, however, he relied on equations representing the C_{obs} boundaries during the search process. Greater movement toward sampling-based motion planning began in the late 1980s. Algorithms in this direction typically centered around advances in efficient calculation of distance between polyhedra [Gilbert et al, 1988] (chapter VI in [Laumond, 1998] provides an excellent overview to the subject).

One early planner that strongly reflects classical grid search techniques was presented by Kondo [Kondo, 1991]. The planner is based on the observation that even if a fine grid is placed over the *C*, it may be possible to find a solution without visiting large portions of the grid. Kondo's planner uses multiple heuristics (i.e., different assignments of the heuristic weight constants), and adaptively selects between them based on an estimate of their effectiveness. Hence, the effectiveness of the planner strongly depends on the quality of the heuristic functions, and on the planner's ability to choose the appropriate one to apply. If either of these are poor, then performance will degrade greatly.

Another early sampling-based motion planner is the SANDROS planner of Chen and Hwang [Chen and Hwang, 1992], which was developed for manipulator arms. This planner searches in a multi-resolution manner over a non-uniform grid (i.e., the resolution on the coordinate axes may differ). The axes are given at different resolutions because for manipulator arms, links near the base have the greatest impact on end effector position. The algorithm uses the GJK algorithm for collision detection. It also uses the distance information to place links of the arm at maximal distance from the workspace obstacles.

A planner introduced by Glavina [Glavina, 90] contains early PRM ingredients. The planner first attempts to connect the initial and goal queries using a straight and slide local planner (a method which does not allow backtracking but is more powerful than the straight-line local planner). If this fails, which is usually the case, then a new configuration is chosen as a subgoal, and attempts to connect the subgoal to the initial and goal configurations using the same local planner. If this fails, new subgoals are added and attempts are made to connect them with previously existing subgoals, as well as the initial and goal configurations. Edges between subgoals are checked for collisions at a pre-defined subsampling resolution. Glavina also identifies the well-known "narrow corridor problem"' and uses connected component analysis to speed up his planner. However, he uses a primitive collision detection method which prevents him from applying his algorithm to challenging, high-degrees of freedom problems; also, the straight and slide local planner becomes expensive in high dimensions.

An interesting work that provides important ideas for the development of PRM methods was presented in [Overmars, 1992]. The technique combines simple potential fields with a roadmap method, constructing a random network of possible motions. The method turns out to work very fast in a planar situation for many different types of scenes. It is simple, hence, easy to implement. One of the nice properties of the method is that it uses a learning approach (i.e., gradually it learns the possible motions of the robot among the obstacles).

2.2 Incremental search methods

Nodes and edges can also be organized in a tree rather than in a graph. By using this approach efficient planners have been designed. They are well suited for addressing single-query motion planning problems, and while growing a tree, it is possible to use the motion equations of the robot, thus obtaining paths complying with kinodynamic constraints.

Incremental search methods applied to motion planning explore the collision-free regions of the *C* trying to find a feasible path between two given configurations, q_s and q_g. The exploration is biased to solve this particular planning query and not to obtain information about the whole space. Most of the algorithms construct trees whose nodes are configurations computed during exploration. The search can be performed in unidirectional or bidirectional directions (see Figure 3). The unidirectional strategy constructs a single tree from one of the two given configurations until the other configuration is reached. The bidirectional strategy constructs one tree from q_s and another from q_g. The solution is found when the two trees meet at a point. Choosing an unidirectional or a bidirectional search mainly depends on the characteristics of problem to be solved. We present the most important approaches.

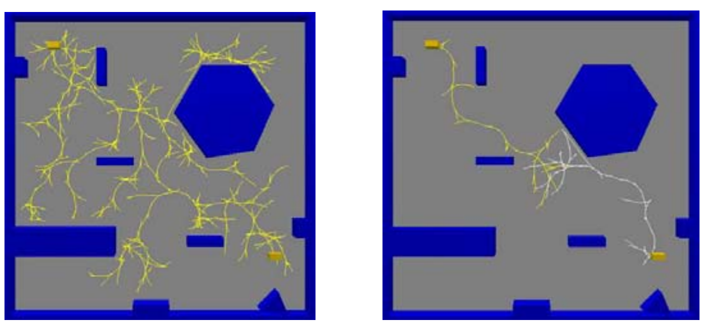


Fig. 3. Bidirectional search vs. unidirectional search

Randomized Path Planner (RPP)

Barraquand and Latombe introduced one of the first randomized motion planners [Barraquand and Latombe, 1991]. This planner combines a gradient descent of the potential with a random walk procedure to scape spurious local minima. RPP is one the most important motion planning algorithms (it solved problems with many dof's, typically many more than other planners at the time were capable of handling). The RPP approach is probabilistically complete and has provided very good results. However, it is now well known that this planner is hindered by narrow passages problems.

ISSN 1405-5546

Ariadne's Clew Algorithm (ACA)

Ariadne's clew approach grows a search tree that is biased to explore as much new regions as possible in each iteration. There are two modes, SEARCH and EXPLORE, which alternate over successive iterations. SEARCH is used to place point subgoals or "landmarks"' evenly in *C* and EXPLORE distributes the landmarks at the end of a Manhattan path from the start configuration [Bessière et al, 1993]. The draw attributed to this approach is that the optimization process carried out by EXPLORE is costly and may require some parameter tuning.

Expansive-Space Tree (EST)

The EST planner presented in [Hsu et al, 1999] and [Hsu, 2000] shares some ideas with PRM approaches, it tries to sample only the portion of *C* that is relevant for a particular planning query, avoiding the cost of precomputing a roadmap for the whole free-space. The algorithm iteratively executes two steps: *expansion* and *connection*, in a similar way than ACA. This is a bidirectional planner (i.e., it constructs two trees), although a unidirectional version is also implemented.

Each node q in the tree has an associated weight w, which is defined to be the number of nodes inside $N_r(q)$, the ball of radius r centered at q. At each iteration, it picks a node to extend; the probability that a give node q will be selected is $1/w(q)$, in which w is the weight function. Then N points are sampled from $N_r(q)$ for the selected node q, and the weight function value for each is calculated. Each new point q' is retained with probability $1/w(q')$ and the planner attempts to connect each retained point to the node q. The goal of the weight function is to avoid oversampling in regions already explored and to rather bias the expansion towards unexplored areas of the *C*. In this respect both RRT and the expansive planner aim to the same goal, the only difference being in the technique used to identify poorly explored zones.

Single query, Bidirectional Lazy collision checking (SBL)

SBL planner is another approach issued from the probabilistic roadmap framework for single planning queries [Sánchez and Latombe, 2002a]. In this case, a roadmap is not built trying to cover the whole *C*. The idea is to exploit the delayed collision checking by combining it with an adaptive sampling technique similar to the one used by EST. The planner searches F by building a roadmap made of two trees of nodes, T_s and T_g. The root of T_s is the start configuration q_s and the root of T_g is the goal configuration q_g (i.e., it uses bidirectional search). Every new node generated during the planning is installed in either one of the two trees as the child of an already existing node. The link between the two nodes is the straight-line segment joining them in *C*. This segment will be tested for collision only when it becomes necessary to perform this test to prove that a candidate path is collision-free (lazy collision checking). This algorithm has been applied to problems involving several manipulators arms operating in the same workspace [Sánchez and Latombe, 2002].

Rapidly-exploring Random Tree (RRT)

The RRT approach, introduced in [LaValle and Kuffner, 2000], has become the most popular single-query motion planner in the last years. RRTs are a class of PRM algorithms that can be used both for systems involving kinodynamic constraints or not. RRT-based algorithms where first developed for non-holonomic and kinodynamic planning problems, where the space to be explored is the state-space (i.e., a generalization of the *C* that involves time). However, tailored algorithms for problems without differential constraints (i.e., which can be formulated in *C*) have also been developed based on the RRT approach.

RRT-based algorithms combine a construction phase with a connection phase. For building a tree, a configuration q is randomly sampled and the nearest node in the tree (given a distance metric in *C*) is expanded toward q. We can obtain different alternatives for the RRT-based planners [LaValle and Kuffner, 2000](see for example Figure 3). The recommended choice depends on several factors, such as whether differential constraints exist, the type of collision detection algorithm, or the efficiency of the nearest neighbour computations.

2.3 The Probabilistic Roadmap

The Probabilistic Roadmap Method (PRM) is one of the leading motion planning techniques. It was developed simultaneously at Stanford and Utrecht [Kavraki et al, 1996]. The key idea of the PRM is to randomly distribute a set of nodes in the robot's configuration space and then connect these nodes using a simple local planner, to form a graph (or a tree) known as a *roadmap*. An important property of a roadmap is that it provides a good approximation of the connectivity of the *F*. If the roadmap is successful capturing this connectivity, motion planning may be reduced to a graph search. Because of the random sampling, the obtained paths would have to be smoothed. The algorithm for constructing the roadmap is shown in pseudo code in the Figure 4.

```
BUILD_PRM()
1 n ← 0
2 while n < N
3    q ← SamplingStrategy.generateConfiguration()
4    if q ∈ F
5      U ← NeighborStrategy.findNeighbors(q)
6      V ← V ∪ q
7      for each u ∈ U
8          if LocalPlaner.verifyPath(q, u) = true
9              E ← E ∪ (q, u)
10     n ← n+1
```

Fig. 4. The basic algorithm

The two time-consuming steps in this algorithm are Line 3, where a free sample is generated, and Line 8 where the algorithm tests whether the local planner can find a path between the new sample and a configuration in the graph. Figure 5 shows the three phases of the approach.

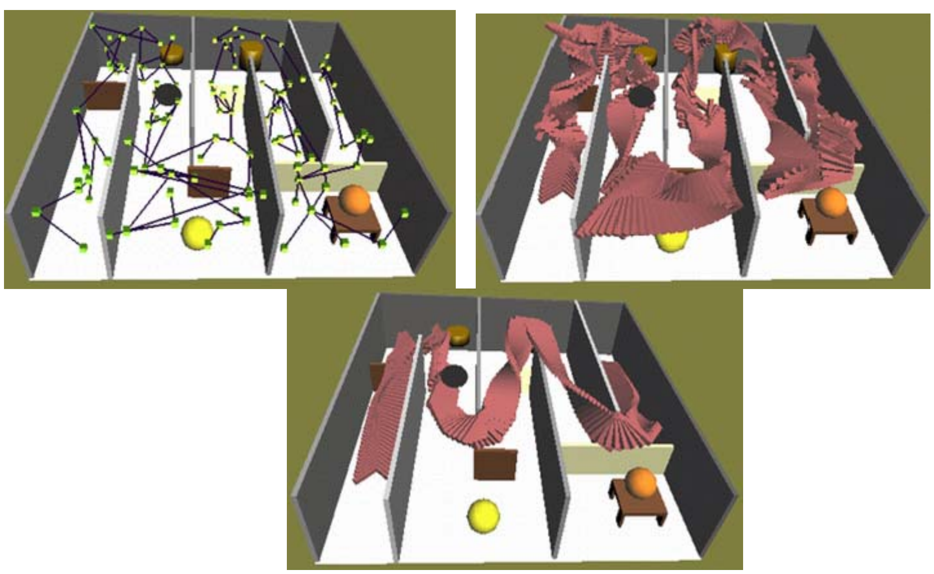


Fig. 5. The three phases of PRM: construction, query and smoothing

The default sampling approach samples the free space in a random way. Narrow passages create significant difficulty for PRM planners. To obtain enough random samples in such narrow passages one would need way too many samples in total.

In recent years, the most popular paradigm for sampling-based motion planning has been the PRM. This success is mainly due to its great efficiency, reliable performance, conceptual simplicity and applicability to many different types of problems. Indeed, algorithms based on these approaches have been used to solve more complex instances of the motion planning problem than the classic mover's problem. In Robotics, they have been extended to: nonholonomic planning [Svestka and Overmars, 1997], [Laumond, 1998], [Sánchez et al, 2003], kinodynamic planning [Hsu, 2000], sensor-based motion planning [Yu and Gupta, 2000], [Lanzoni et al, 2003], motion planning under closure constraints [Cortes et al, 2002] and manipulation planning [Siméon et al, 2002]. They have also been adopted to solve problems in other areas, as for example: maintenance problems in industrial logistics [Siméon et al, 2001], [Van Geem et al, 1998], animation of characters in computer graphics [Pettre et al, 2003], [Sánchez et al, 2005] or computational Biology [Song and Amato, 2001].

The design of PRM-based motion planning algorithms requires some ingredients we describe in the next section. Several options are available for some of these basic elements. The behavior of the algorithm is associated with the selection of these ingredients. Some choices are more appropriate than others depending on particularities of the problem to be solved. A work published in [Geraerts and Overmars, 2002] provides a comparative study of variants of the classic PRM algorithm obtained by selecting different options for ingredients. This study yields very useful conclusions for the design of PRM planners. However, the determination of this analysis must be relativized since studied examples only involve free-flying robots (i.e., the dimension of C is six).

3 Ingredients for sampling-based planners

The basic PRM approach leaves many details to be filled in, like how to sample the space, what local planner to use and how to select promising pairs. Over the past ten years researchers have investigated these aspects and developed many improvements over the basic scheme (see [Amato et al, 1998; Hsu et al, 1998; Boor et al,1999; Nissoux et al, 1999; Wilmarth et al, 1999; Bohlin and Kavraki, 2000; LaValle and Branicky, 2002; Sánchez, 2003; Brekis et al, 2003]). This has led to many variants of the method, each with its own merits. It is difficult to compare the different techniques because they were tested on different types of environments, using different underlying libraries, implemented by different people on different machines. As far as we know, a study has not been made for general mechanical systems.

3.1 Node adding strategies

We consider the following techniques:

Nearest-k. The strategy tries to connect the new configuration to the nearest-k nodes in the graph that lie close enough. The rationale is that nearby nodes result in short connections that can be efficiently checked for collisions.
Component. In this case, it tries to connect the new configuration to the nearest node in each connected component that lies close enough. It is preferable to connect to multiple connected components.
Component-k. This strategy tries to connect the new configuration to at most k nodes in each connected component. When the number of components is small, one can prefer to spend some extra time on trying to make connections. Otherwise the time required for adding the node will become the dominant factor.
Visibility. The method is based on visibility sampling technique described in [Nissoux et al, 1999]. The strategy only connects configurations to useful nodes, i.e., when a new node can not be connected to other nodes it forms a new connected component and is labeled useful. If it connects two or more components it is also labelled useful. If it can be connected to just one component it is not labeled useful. The number of useful nodes remains small, making it possible to try connections to all of them. The connection distance and number of connections are not restricted.
Visibility-k. The method is again based on the visibility approach. Now, only the nearest k nodes that lie close enough will be considered for the usefulness test.

RRT-based algorithms select a node in the tree to be expanded toward a sampled configuration. The choice of the nearest neighbor seems the most reasonable strategy. Due to this strategy, the behavior of the planner strongly depends on the distance metric.

3.2 Local planners

The local planner is usually a simple, incomplete but fast heuristic. The local planner is made fast by only considering local information about the configuration and/or workspace to find a path, hence the name “local planner”'. Because local planners only use local information they are bound to be incomplete, contrary to the global planners which search the global configuration space, and usually have some kind of completeness property. Local planners can be either deterministic or nondeterministic. Deterministic local planners always produce the same result when applied to the same problem, whereas a nondeterministic local planner might produce different results.

The local planner computes a feasible path in C between any two configurations considering some intrinsic motion constraints of the mechanical system (it considers constraints imposed by joints and non-holonomic constraints but ignores collision avoidance). In the PRM literature, the local method is often called *the steering method* [Laumond and Siméon, 2000]. This choice may affect the combinatorial complexity of the algorithm. For mobile and free-flying systems that are not affected by differential constraints, devising steering methods is in general an easy task.

Straight-line. The path computed by this steering method is always the straight-line, or linear interpolation bewteen q_s and q_g in C.
Manhattan. A one dimensional Manhattan path is a path where the joints move one at the time from their start position to their goal position.
Rotate-at-s. It produces a path, where first a subset of the dofs moves a fraction s on the straight-line bewteen q_s and q_g, then the remaining dofs are moved from their start to their goal positions, and then the first dofs are moved from their intermediate positions to q_g [Amato et al, 2000].
A*-like. This method uses a simple potential function to guide its search from q_s to q_g [Amato et al, 2000].

Next figure illustrates different paths obtained by the different steering methods mentioned above in a 2 dof configuration space.

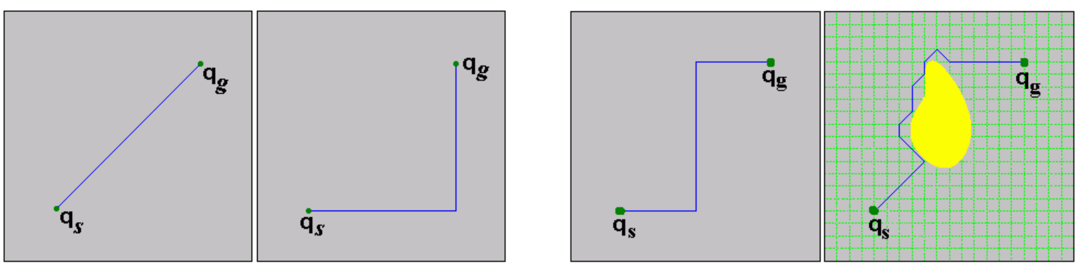


Fig. 6. The different steering methods for mobile and free-flying systems without differential constraints

In most cases, a simple straight-line segment in C is an admissible connection. However, under differential constraints the design of adequate steering methods is a difficult problem [Svestka and Overmars, 1997; Laumond, 1998; Laumond and Siméon, 2000]. Indeed, techniques are only available for some classes of systems depending on controllability issues. For car-like robots one can use RTR paths as steering method [Svestka and Overmars, 1997], another alternative is to use a steering method that constructs the shortest path connecting the two configurations; Dubins paths for robots that move always forward [Dubins, 1957] or Reeds & Shepp paths for robots moving both forward and backward [Reeds and Shepp, 1990] (see Figure 7).

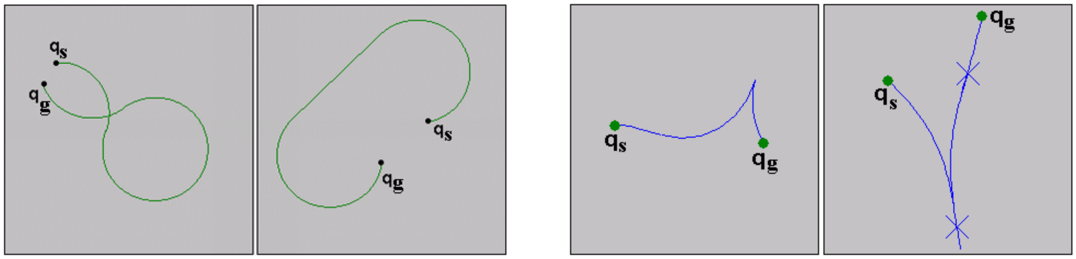


Fig. 7. Steering methods for car-like robots: Dubins and Reeds & Shepp paths. At the right of each curve, one can see the continuous-curvature for both shortest paths

Most motion planners for car-like robots compute Reeds & Shepp paths or Dubins paths made up of line segments connected with circular arcs. Such paths have a discontinuous curvature that makes them difficult to track (curvature is related to the orientation of the front wheels). To solve this problem, we can devise planners that computes paths with continuous-curvature and upper bounded curvature derivative (curvature derivative is related to the steering velocity). Continuous-Curvature curves allow the robot to traverse the path without having to stop and they also allows the robot to more accurately track the path. In the Figure 7, one can see these differences.

3.3 Sampling strategies

These strategies are used to generate new samples during the roadmap construction phase. The choice of sampling strategy depends on the motion planning problem. In some cases, it is advantageous to spend more time on the generation of each sample, if the resulting samples are placed more efficiently in the *C*. The first papers on the PRM used uniform random sampling of the *C* to select the nodes that are added to the graph. In recent years, other uniform sampling approaches have been suggested to remedy certain disadvantages of the random behavior.

Random. A sample is created by choosing random values for all degress of freedom of the mechanical system.
Grid. One can choose samples on a grid. Because the grid resolution is unknown in advance, one can start with a coarse grid and refine it in the process, halving the cell size. Grid points on the same level of the hierarchy are added in random order.
Low-discrepancy sequences. In [LaValle and Branicky, 2002] and [Sánchez, 2003] it has been suggested to use low-discrepancy point sets as samples. Low-discrepancy point sets have been used in discrepancy theory to obtain a coverage of a region that is better than using a grid [Branicky et al, 2001]. LaValle et al., only used two low-discrepancy sequences, Halton and Hammersley. The work in [Sánchez et al, 2003] proposed other low-discrepancy sequences, Sobol, Faure, Niederreiter and generalized Halton.
Randomized low-discrepancy sequences. In this variant of Halton, the randomized Halton sequence is obtained by randomizing the starting point of the sequence [Sánchez et al, 2007].
Cell-based. In this approach, one can take random configurations within cells of decreasing size in the workspace. The first sample is generated randomly in the whole space. Next the workspace is split in 2^3 equally sized cells. In a random order, a configuration in each cell is generated. Next each cell is split into sub-cells and it is repeated for each sub-cell. This should lead to a better distribution of the samples over the *C* compared to random sampling.

Rather than uniform sampling, it has been suggested to add more samples in difficult regions of the workspace. We present a number of these techniques.

Gaussian. This sampling strategy is intended to add more samples near obstacles. The idea is to take two random samples, where the distance σ between the samples is chosen according to a Gaussian distribution. Only if one of the samples lies in F and the other lies in C_{obst} the method add the fre sample. This leads to a favourable sample distribution [Boor et al, 1999]. In some especial cases, this sampling strategy can be extended to reduce the number of wasted samples by paying a higher computational cost.

Obstacle-based. This technique has a similar goal [Amato et al, 1998]. The method picks a uniform random sample. If it lies in *F*, the sample is added to the graph. Otherwise, the algorithm pick a random direction and move the sample in that direction with increasing steps until it becomes free and add the free sample. One possible variation of the previous technique is when the sample is discarded if it initially lies in *F*. This will avoid many samples in large open regions.

Bridge test. The bridge test is a hybrid technique that aims at better coverage of the *F* [Hsu et al, 2003]. The idea is to take two random samples, where the distance σ between the samples is chosen according to a Gaussian distribution. Only if both samples lie in the forbidden configuration space (A configuration that describes the placement of the robot in the workspace is called forbidden, if it causes the system to touch or intersect with the obstacles in the workspace) and the point in the middle of them lies in *F* the free sample is added. To also get points in open space, every sixth sample is chosen random.

Medial axis. This technique generates samples near the medial axis of the free space [Wilmarth et al, 1999]. All samples represent mechanical system placements having at least two equidistant nearest points on the obstacles resulting in a large clearance from obstacles. The method is relatively expensive to compute since expensive closest pair calculations are involved.

3.4 Distance metrics

Distance metrics are used to determine the adjacency of configurations. The selection of a good distance metric is critical for the performance of PRM-based algorithms since this information is normally used to decide which connections or expansions must be tested. The ideal metric should consider the existence of motion constraints (i.e., joint limits, obstacles, differential constraints, etc). Designing such a metric remains obviously a very difficult issue. The distance metric is a symmetric function $dist \in C \times C \rightarrow R^+$, we mention the most common distance metrics.

Euclidean distance. The Euclidean distance is the length of the straight-line in *C* between the start and goal configurations.

Manhattan distance. The Manhattan distance is the length of the one-dimensional Manhattan path connecting the two configurations [Bessière et al, 1993].

Weighted distance. The Euclidean distance metric and the Manhattan distance treats all dimensions of *C* equal. This is usually too rough an estimate of the actual change of success of the local planner. If for instance, we are working with an articulated robot, then the robot is likely to sweep far more volume per unit movement, if we move joints near the robot based, then if we move joint near the tool. The problem can be solved by applying weights to the individual dimensions in the above metrics. The weighting can also be used to give preference to movement in certain directions of *C*.

Under differential constraints, the design of an appropriate metric in *C* is quite more complex [Laumond, 1998]. For car-like robots, the distance from a robot configuration to an obstacle is the length of the shortest feasible path bringing the robot in contact with the obstacles. This kind of robot is a nonholonomic system. This means that any path in the configuration space is not necessarily feasible. As a consequence, the length of the shortest feasible paths induces a special distance, the so-called nonholonomic distance, that is not an Euclidean distance (see Chapter 1 in [Laumond, 1998] for an excellent overview).

3.5 Ingredients for incremental search planners

Rapidly-exploring Random Trees are a class of robot motion planning algorithms that can be used both for systems involving kinodynamic constraints or not. In addition to the check and distance routines used in the PRM framework, the RRT algorithm assumes the availability of the following elements:

- a set of *U* of inputs to be applied to the system
- an incremental simulator, i.e., a procedure that given a state $x(t) \in X$ (the state space) and an input $u \in U$, produces the state $x(t + \Delta t)$, provided that the input *u* has been applied over the given time interval.

If we include system's equations into the incremental simulator, the planner is able to directly produce paths satisfying the kinodynamic constraints. If the system does not have differential constraints, the planner is required to produce a path for holonomic robots, no incremental simulation takes place and simple interpolation is performed as every motion is allowed.

The ingredients for the expansive space planner are the same that have appeared in previous subsections. When we consider kinodynamic constraints, the planner build a single tree in the space-time [Hsu, 2000]. Along the same lines of the RRT approach, it is assumed the availability of an incremental simulator and a set of inputs U_l. In this context, the inputs U_l are piecewise constant functions with at most l pieces. RRT-based algorithms perform collision checking while trying to expand a node of the tree. This expansion process can also benefit from efficient dynamic checkers with adaptive resolution.

3.6 Properties

The analysis of sampling-based planners is a difficult problem. Main subjects of analysis involve completeness (i.e., the ability to find a solution whenever one exists), coverage and connectivity properties (i.e., the amount of the search-space encoded in the roadmap structure) and computational complexity.

Concerning the PRM approach, proofs for probabilistic completeness are given in [Kavraki, 1995; Hsu et al, 1998]. In [Svestka, 1997], detailed demonstrations of probabilistic completeness are extended to more general kinds of mobile systems, affected by nonholonomic constraints (see also the Chapter I in [Laumond, 1998]).

The notion of $\in$-goodness was introduced in [Barraquand et al, 1997] for the analysis of coverage properties. F is $\in$-good if the volume of the visibility set (The set of samples attainable by the local planner) of any sample in F is greater than some fixed percentage $(1 - \in)$ of the total volume of F. Barraquand et al., proved that if F is $\in$-good, the probability that a roadmap computed by the PRM algorithm does not cover this subset decreases exponentially with the number of nodes.

The notion of *expansiveness* introduced in [Hsu et al, 1999] deals with connectivity. The authors also explained the notion of narrow passages in F and the difficulty to go through them. They showed that for an expansive F, the probability for a (basic) roadmap not to capture the connectivity of F decreases exponentially with the number of nodes. The dependence of the $\in$-goodness, expansiveness and clearance notions on the dimension of C is also discussed in this paper. The *clearance* of a path is an important factor. In [Hsu et al, 1998; Barraquand et al, 1997] a bound on the number of nodes required to capture the existence of a path of given clearance is provided, this bound depends on the length of the path. In [Svestka, 1997] the dependence on the length is replaced by the dependence on the number of visibility sets required to cover the path. Again, the probability to fail decreases exponentially with the number of nodes.

All these results are based on parameters characterizing the geometry of F which are a priori unknown. An approximated knowledge of such parameters could be very useful for the setting of algorithms, but it seems that a reliable estimation would take at least as much time as building the roadmap itself. A recent publication [Ladd and Kavraki, 2002] unifies some previous works and makes reformulation looking for more intuitive parameters and opening a novel framework for the analysis of PRM-based algorithms applicable to more general problems.

The theoretical development and analysis of RRT algorithms has been, up to date, entirely fulfilled by LaValle’s group. RRT planners are probabilistically complete, and the tree nodes converge to the sampling distribution [LaValle and Kuffner, 2000].

3.7 Some considerations

Given a motion planning problem, the choice of the planning algorithm to use is driven by different factors. The first aspect to consider is whether the problem involves kinodynamic constraints or not. If this is the case, the choice is for one of the two algorithms: RRT or EST. Up to now no analytic comparison is available, and also no fair experimental comparisons have been performed. On the other hand both algorithms proved to be suitable for being used in real world applications, they address the same class of problems, and they require the same components. Certain authors report that expansive planners are more difficult to tune because of the higher number of parameters. It is however somehow difficult to give general indications on the one which could better fit the needs, or could be easier to implement. If the problem to be solved involves just kinematics constraints, then all the proposed algorithms can be used. In the single-query scenario, tree based algorithms are in general much faster. It has however to be pointed out that speed comes to the price of path quality, since these planners stop as soon as a path is found. PRM-based planners, instead, can produce a set of paths, and then the most favorable one is returned. In a situation where many successive queries have to be solved, also the use of the basic PRM algorithm appears appropriate. If the operating environment exhibits a *C* with narrow passages, then one of the outlined refined PRM algorithms is the choice. It is nevertheless evident that in general no algorithm is better, but rather the environment influences the performance. Another important issue is the sampling and resampling strategy. The vast majority of the proposed planners propose uniform sampling over either *C* or a suitable subset. This leads to easy implementation, but has the outlined drawbacks.

4 Mexican contributions

In this section we present the contributions made by Mexican researchers in the last fifteen years. The main development focus are distributed in United States (Stanford, Texas A & M, Rice universities,…) and Europe (France, Germany, Holland, Spain). It is difficult to mention with much detail each one of the works proposed. We summarized briefly the most important contributions. The research in Mexico in this field, are located in the cities of Mexico city, Guadalajara, Puebla, Guanajuato among others. Even if there are common conferences in the topic, there is no common research between the groups. Contributions include the Ariadne's clew algorithm, an adaptive framework for single-shot motion planning, SBL algorithm, deterministic sampling and other important contributions proposed the last five years.

4.1 The Ariadne's clew algorithm

As we have seen in the section I, the computation of the *C* is a very time-consuming task. The main idea behind the Ariadne's clew algorithm is to avoid this computation. In order to do this, the algorithm searches directly for a feasible path in the trajectory space. The *C* is never explicitly computed. In this kind of space, motion planning may be seen as an optimization problem and solved as such by an algorithm called SEARCH. It is possible to build an approximation of free space by another algorithm called EXPLORE that is also posed as an optimization problem. The Ariadne's clew algorithm is the result of the interleaved execution of SEARCH and EXPLORE procedures [Mazer et al, 1998]. The algorithm is general in the sense that it may be used for numerous and very different applications in robotics. Basically, the main thing that needs to be changed in the algorithm is the distance *d* used in the evaluation functions of the two optimization problems.

4.2 An adaptive framework for single-shot motion planning

Vallejo et al., [Vallejo et al, 2000] proposed an adaptive framework for single-shot motion planning (i.e., a single-query planner). The author's strategy consists of adaptively select a planner whose strengths matches the current situation, and then switches on-line to a different planner. Experimental results show that this strategy solves queries that none of the planners could solve on their own; and is suitable for crowded environments in which the robot's free-configuration space has narrow corridors such as in maintainability studies in complex 3D CAD models. The author's also presented an interesting comparative evaluation of different distance metrics and local planners within the context of probabilistic roadmap methods for motion planning [Amato et al, 2000]. The study concentrates on cluttered 3D workspaces, their results include recommendations for selecting appropriate combinations of distance metrics and local planners for use in motion planning methods, particularly probabilistic roadmap methods.

4.3 SBL algorithm

The planner presented by Sánchez and Latombe [Sánchez and Latombe, 2002a] uses a lazy collision-checking strategy (it postpones collision tests along connections in the roadmap until they are absolutely needed) with single-query bidirectional sampling techniques (it does not pre-compute a roadmap, but uses the two input query configurations to explore as little space as possible and searches the robot's free space by concurrently building a roadmap made of two tree rooted at the query configurations). Combining these techniques, the planner can solve motion planning problems of practical interest (i.e., with realistic complexity) in times ranging from fractions of a second to a few seconds. There are two well established approaches to multi-robot motion planning: centralized and decoupled. An interesting study [Sánchez and Latombe, 2002] compared the decoupled and the centralized approaches with the SBL (Single query, Bi-directional, Lazy in collision checking) planner. The study concluded that decoupled planners are very unreliable in practice due to their inherent incomplete nature. Hence they suggest that a centralized approach should usually be the preferred choice (especially in cases requiring tight robot coordination), even though they are slower than that a decoupled planner when the latter succeeds.

4.4 Deterministic sampling

Deterministic sampling techniques have been developed by numerous mathematicians over the past century. Monte Carlo methods are described simply as numerical methods based on random sampling. It is therefore a method with a strong statistical and probabilistic flavour. Quasi-Monte Carlo methods are deterministic versions of the classical Monte Carlo methods, in the sense that the random samples are replaced by well-chosen deterministic points.

One can implement a deterministic roadmap (DRM), by using the *d*-dimensional low-discrepancy sequence instead of points generated from a pseudo-random number generator. The operation of the DRM is identical to PRM (i.e., we can only replace random points in Step 3 of the roadmap generation algorithm above by specialized low-discrepancy points [Sánchez, 2003; Sánchez et al, 2003]), this is an improvement over the classic PRM. A 2D example for which N=1000 is shown in Figure 8. Note the characteristic clumping of the randomly chosen nodes and the relatively large areas of free space that contain no samples. The figure also confirms the well-known fact that narrow passages in configuration space are notoriously more difficult to find at random.

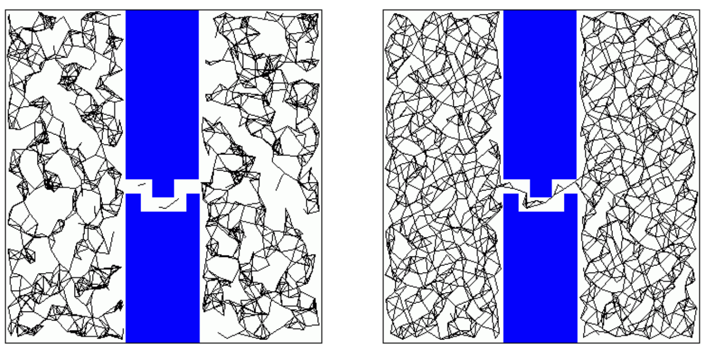


Fig. 8. Random sampling vs. deterministic sampling. Each uses 1000 samples and the same connection radius

In [Sánchez, 2003], the author presented an empirical study about the uniformity of the low-discrepancy sequences used to generate samples in a deterministic way. This study shows that the dimension is a crucial factor that affects the performance of the deterministic generators. These experiments consolidate two ideas: 1) With a larger base, a low-discrepancy sequence can present certain pathologies?, and 2) does the minimal size of a low-discrepancy sample have a better equidistribution properties than a pseudo-random sequence that grows exponentially with the dimension?. The main rationale for using deterministic sources is that they reduce the discrepancy or dispersion of the samples. However, the computational cost of achieving a fixed discrepancy or dispersion grows exponentially with the dimension of C [Sánchez, 2003]. The samples generated by a deterministic source are distributed evenly and regularly over $[0, 1]^{\dim(C)}$. In PRM planning, N (N is the number of samples) is relatively small and the dimension of C could be large (greater than six). This observation leads to large discrepancy and dispersion, even when a deterministic source is used. Hence, the advantage that deterministic sources can possibly achieve over pseudo-random sources necessarily fades away as dimension of C increases.

4.5 Other contributions

In this subsection the most recent works are presented, which is related to the sampling-based motion planning field, although we know that many important contributions exist in robotics, we tried to summarize these new proposals without discrediting other important works.

A Lazy PRM approach for non-holonomic motion planning was presented in [Sánchez et al, 2002]. The algorithm is similar to the work presented by Bohlin and Kavraki [Bohlin and Kavraki, 2000] in the sense that the aim is to find the shortest path in a roadmap generated by randomly distributed configurations (in a later work the authors showed that the use of deterministic sampling improved remarkably past results obtained with random sampling [Sánchez et al, 2003]).

Esteves et al. [Esteves et al, 2004] proposed a decoupled approach to solve the geometric and dynamic aspects of the motion planning problem sequentially for two animated virtual characters manipulating an object cooperatively. The decomposition enables the character's system to plan a collision-free path for a reduced version, then to animate locomotion and grasping behaviors independently, and finally to automatically tune the animation to avoid residual collisions [Esteves et al, 2006]. Esteves also proposed a strategy for the generation of automatic collision-free motions for humanoid robots. The goal is to provide a method for planning humanoid dynamic motions based on her three-stage approach [Esteves, 2007].

The development of good metrics for sampling-based planners that, although essential for improving and expanding the applications of motion planning, have not been sufficiently developed to address key aspects of the process. First, in order to compare planners we need metrics to measure their ability to model different types of problems. Second, in order to stop planning at the right moment we need metrics to monitor sampling progress over time. Third, in order to spatially adjust sampling, we need metrics to analyze the different areas of the planning space found. Moreover, metrics are also needed to analyze the types of motions represented in a model, Morales et al., proposed a set of metrics that measure the amount of improvement achieved in a model due to each sample added by the planner [Morales et al, 2006; Morales et al, 2007].

Most of the studies made to compare the performance of the PRM planners has taken place with multiple-query planners. In [Sánchez et al, 2007], the authors proposed a novel study to compare random and deterministic sampling in the context of single-query strategies. They used two planners, the expansive planner proposed by Hsu [Hsu et al, 1999; Hsu, 2000] and the SBL planner proposed by Sánchez and Latombe [Sánchez and Latombe, 2002a]. Nevertheless, Geraerts and Overmars have made an interesting study with free-flying objects in a three-dimensional space [Geraerts and Overmars, 2002], but only in the multiple-query approach. Such objects have six dofs. The authors used the most simple local method that consists of a straight-line motion in configuration space. Another interesting study was made in [Sánchez et al, 2003], in this study they proposed the use of single-query and multiple-query approaches and different local methods.

The results presented in the single-query study show that many claims on efficiency of certain sampling approaches could not be verified. There was little difference between the various uniform sampling methods. One thing that is clear from this study is that a careful choice of techniques is important. Also, it is not necessarily true that a combination of good techniques and parameter choices results in optimal running times. In conclusion, the authors emphasize that the advantage of the deterministic sampling on the pseudo-random sampling is only observable in low dimension problems.

The work proposed by [Arechavaleta et al, 2006] is related with the proposal of a differential system which accurately describes the geometry of human locomotor trajectories of humans walking on the ground level, in absence of obstacles. Their approach emphasizes the close relationship between the shape of the locomotor paths in goal-directed movements and the simplified kinematic model of a wheeled mobile robot. This work shows that human locomotion can be approximated by the motion of a nonholonomic system.

5 Conclusions

During the last four decades motion planning has emerged as a crucial and productive research area in Robotics. Several extensions of the basic problem have been studied, where, for instance, obstacles are moving, kinematic and dynamic constraints limit robot motions, optimized trajectories must be computed, or multiple robots have to be coordinated. Like the basic problem, some of these problems now have solutions that can be used in practice, but many still require more research.

Sampling-based planners can successfully handle a large diversity of problems. The success of these planners in solving problems with many degrees of freedom and many obstacles can be explained by the fact that no explicit representation of the F is required. The main operation of these planners is the ability of checking placements where the robot collide with obstacles in the environment; this procedure can be efficiently performed by the current generation of collision checkers. Information on the configuration space is acquired by generating samples and edges to connect them, which are stored in a suitable data structure. Following this paradigm, many different algorithmic techniques have been proposed, and some of them are now widely accepted as part of the standard literature in the field.

To speed up PRM-based planners, one promising direction is to design better sampling strategies (and perhaps connection strategies as well) by exploiting the partial knowledge acquired during roadmap construction and using this knowledge to adjust the sampling measure on-line, to make it more effective. The sampling-based planners tend to perform poorly when crucial configurations lie in and around narrow regions of the configuration space; it has been identified as the narrow passage problem. The narrow passage problem can be tackled by incorporating a hybrid or adaptive sampling strategy that concentrates samples in difficult areas, and/or generates samples in large open areas. The use of the uniform sampling strategy is not a good choice for environments involving narrow passages. Another tactic is to employ a more powerful local planner.

Motion planning applications will grow considerably. CAD-based robot programming systems will include planners to compute quasi-optimal motions of robots and to optimize robot layouts. CAD systems will include planners to verify that products or building facilities can be easily manufactured or built, and serviced. In video games, character motions will be (re-)computed on the fly to be adapted to user inputs, allowing new forms of games. More generally, video games, movies, animated WebPages and interfaces will converge, with motion planning and motion capture, used jointly to produce rich interactive animation of virtual characters. Motion planners will help surgeons to plan for minimally-invasive operations. Software packages to assist biochemists in the discovery of new drugs will include motion planners to compute plausible motions of ligands binding against proteins. Simultaneously, new motion planning problems will be investigated. For instance, reconfigurable robots made up of thousands of modules will require planners to compute their reconfiguration motions. Robots equipped with sensors to collect information will need planners to decide which are the most efficient motions to obtain pertinent information. Minimizing surgical invasiveness will require planners that think about the deformations of soft tissues caused by the motions of surgical tools. It is almost certain that the basic motion planning problem which has been the focus of the research in motion planning for more than two decades will soon loose this status, No other problem; however seems basic enough to play the same role in the future.

We presented the most influential algorithms developed in the last years in the field of sampling-based motion planning. This paper illustrated the most common sampling based algorithmic techniques, namely graph based and tree based. The field is however continuously growing, and more and more refinements are being proposed, in the way that an exhaustive enumeration of the many possible variations is doomed to early obsolescence. Now, the user who needs to implement a motion planning strategy should have concrete indications about its strength and limitations, and will not find many difficulties in adapting them to its specific needs. The future indicates that these strategies will be more and more sophisticated and specialized.

References

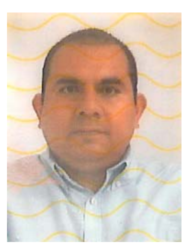
1. **Amato N. M., O. Bayazit, L. K. Dale, C. Jones** and **D. Vallejo,** "OBPRM: An obstacle-based PRM for 3D workspaces". *Proc. Workshop on the Algorithmic Foundations of Robotics*, 1998, 155-168.

2. **Amato N. M., O. Bayazit, L. K. Dale, C. Jones** and **D. Vallejo,** "Choosing good distance metrics and local planners for probabilistic roadmap methods". *IEEE Transactions on Robotics and Automation*, 16(4):442-447 (2000).

3. **Arechavaleta G., J. P. Laumond, H. Hicheur** and **A. Berthoz,** "The nonholonomic nature of human locomotion: A modeling study", *International Conference on Biomedical Robotics and Biomechatronics*, 2006.
4. **Barraquand J.** and **J. C. Latombe,** "Robot motion planning: A distributed representation approach", *The International Journal of Robotics Research*, 10(6):628-649 (1991).
5. **Barraquand J**., **L. E. Kavraki**, **J. C. Latombe**, **T. Yen Li**, **R. Motwani** and **P. Raghavan**, "A random sampling scheme for Path planning", *The International Journal of Robotics Research*, 16(6):759-774 (1997).
6. **Bessière P**., **J. M. Ahuactzin**, **El-Ghazali Talbi** and **E. Mazer**, "The Ariadne's clew algorithm: Global planning with local methods", *IEEE Int. Conf. on Intelligent Robots and Systems*, 1993.
7. **Bohlin R**. and **L. Kavraki**, "Path planning using lazy PRM", *IEEE Int. Conf. on Robotics and Automation*, 2000, 521-528.
8. **Boor V**., **M. H. Overmars** and **A. Frank van der Stappen**, "The gaussian sampling strategy for probabilistic roadmap planners", *IEEE Int. Conf. on Robotics and Automation*, 1999, 1018-1023.
9. **Brekis K. E**., **B. Y. Chen**, **A. M. Ladd**, **E. Plakue** and **L. Kavraki**, "Multiple query probabilistic roadmap planners using single query primitives", *IEEE Int. Conf. on Intelligent Robots and Systems*, 2003.
10. **Branicky M. S**., **S. M. LaValle**, **K. Olson** and **L. Yang**, "Quasi-randomized path planning", *IEEE Int. Conf. on Robotics and Automation*, 2001, 1481-1487.
11. **Canny J.**, *The complexity of robot motion planning*, MIT Press, 1988.
12. **Chang H**. and **T. Y. Li**, "Assembly maintainability study with motion planning", *IEEE Int. Conf. on Robotics and Automation*, 1995, 1012-1019.
13. **Chen P. C**. and **Y. K. Hwang**, "Sandros: A motion planner with performance proportional to task difficulty", *IEEE Int. Conf. on Robotics and Automation*, 1992, 2346-2353.
14. **Choset H**., **K. M. Lynch**, **S. Hutchinson**, **G. Kantor**, **W. Burgard**, **L. E. Kavraki** and **S. Thrun**, *Principles of robot motion: Theory, algorithms and implementations*, MIT Press, Boston, 2005.
15. **Cortes J**., **T. Siméon** and **J. P. Laumond**, "A random loop generator for planning the motions of closed kinematic chains using PRM methods", *IEEE Int. Conf. on Robotics and Automation*, 2002, 2141-2146.
16. **Donald B. R.**, "A search algorithm for motion planning with six degrees of freedom", *Artificial Intelligence*, 31:295-353 (1987).
17. **Dubins L. E.**, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents", *American Journal of Mathematics*, 79:497-516 (1957).
18. **Esteves C.**, **G. Arechavaleta** and **J. P. Laumond,** "Planning cooperative motions for animated characters", *International Symposium on Robotics and Automation (ISRA)*, 2004.
19. **Esteves C.,** *Motion planning: From digital actors to humanoid robots*, Phd Thesis, INPT, 2007.
20. **Esteves C.**, **G. Arechavaleta**, **J. Pettre** and **J. P. Laumond**, "Animation planning for virtual characters cooperation", *ACM Transactions on Graphics*, 25(2):319-339 (2006).
21. **Faverjon B**., "Hierarchical object models for efficient anti-collision algorithms", *IEEE Int. Conf. on Robotics and Automation*, 1989, 333-340.
22. **Glavina B.**, "Solving findpath by combination of goal-directed and randomized search", *IEEE Int. Conf. On Robotics and Automation*, 1990, 1718-1723.
23. **Geraerts R**. and **M. H. Overmars**, "A comparative study of probabilistic roadmap planners", *Proc. of the Workshop on the Algorithmic Foundations of Robotics*, 2002, 43-57.
24. **Gilbert E. G**., **D. W. Johnson** and **S. S. Keerthy**, "A fast procedure for computing the distance between complex objects in three-dimensional space, *IEEE Journal of Robotics and Automation*, 4(2):193-203 (1988).
25. **Gupta K.** and **A. Del Polbi** (Editors), *Practical motion planning in robotics: Current approaches and future directions*, John Wiley & Sons, 1998.
26. **Hwang Y. K**. and **N. Ahuja**, "Gross motion planning: A survey", *ACM Computing Surveys*, 24(3):219-291 (1992).
27. **Hsu D**., **L. Kavraki**, **J. C. Latombe**, **R. Motwani** and **S. Sorkin**, "On finding narrow passages with probabilistic roadmap planner", *Proc. of the Workshop on the Algorithmic Foundations on Robotics*, 1998, 141-154.

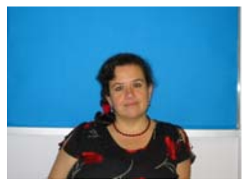
28. **Hsu D**., **J. C. Latombe** and **R. Motwani**, "Path Planning in expansive configuration spaces", *International Journal Computational Geometry & Applications*, 9(4-5):495-512 (1999).
29. **Hsu D.**, *Randomized single-query motion planning in expansive spaces*, PhD thesis, Stanford University, 2000.
30. **Hsu D**., **T. Jiang**, **J. Reif** and **Z. Sun**, "The bridge test for sampling narrow passages with probabilistic roadmap planners", *IEEE Int. Conf. on Robotics and Automation*, 2003, 4420-4426.
31. **Kavraki L. E.**, *Random networks in configuration space for fast path planning*, Phd Thesis, Stanford University, 1995.
32. **Kavraki L**., **P. Svestka**, **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):566-579 (1996).
33. **Kondo K**., "Motion planning with six degrees of freedom by multistrategic bidirectional heuristic fre-space enumeration", *IEEE Transactions on Robotics and Automation*, 7(3):267-277 (1991).
34. **Ladd A**. and **L. Kavraki**, "Generalizing the analysis of PRM", *IEEE Int. Conf. on Robotics and Automation*, 2002, 2120-2125.
35. **Lanzoni C**., **A. Sánchez L**. and **R. Zapata**, "Sensor-based motion planning for car-like mobile robots in unknown environments", *IEEE Int. Conf. on Robotics and Automation*, 2003, 4258-4263.
36. **Latombe J. C.**, *Robot motion planning*, Kluwer Academic Publishers, 1991.
37. **Laumond J. P**. (Editor), *Robot motion planning and control*, Springer-Verlag, 1998.
38. **Laumond J. P.** and **T. Siméon**, "Notes on visibility roadmaps and path planning", *Proc. of the Workshop on the Algorithmic Foundations of Robotics*, 2000.
39. **LaValle S. M**. and **J. J. Kuffner**, "Rapidly-exploring random trees: Progress and prospects", *Proc. of the Workshop on the Algorithmic Foundations on Robotics*, 2000.
40. **LaValle S. M.** and **M. S. Branicky**, "On the relationship between classical grid search and probabilistic roadmaps", *Proc. of the Workshop on the Algorithmic Foundations of Robotics*", 2002.
41. **LaValle S. M**., *Planning algorithms*, Cambridge University Press, 2006.
42. **Lindemann S.** and **S. M. LaValle**, "Current issues in sampling-based motion planning", *Robotics Research: The Eleventh International Symposium*, 2005, 36-54.
43. **Lozano-Pérez T**., "Spatial planning: A configuration space approach", *IEEE Transactions on Computers*, 32(2):108-120 (1983).
44. **Mazer E**., **J. M. Ahuactzin** and **P. Bessière**, "The ariadne's clew algorithm", *Journal of Artificial Intelligence Research*, 9:295-316 (1998).
45. **Morales M.**, **R. Pearce** and **N. M. Amato**, "Metrics for analyzing the evolution of C-Space models", *IEEE Int. Conf. on Robotics and Automation*, 2006, 1268-1273.
46. **Morales M**., **R. Pearce** and **N. M. Amato**, "Analysis of the evolution of C-Space models built through incremental exploration", *IEEE Int. Conf. on Robotics and Automation*, 2007, 1029-1034.
47. **Nissoux C**., **T. Siméon** and **J. P. Laumond**, "Visibility based probabilistic roadmaps", *IEEE Int. Conf. on Intelligent Robots and Systems*, 1999, 1316-1321.
48. **Overmars M. H.**, "A random approach to motion planning", *Technical Report RUU-CS-92-32*, Utrecht University, 1992.
49. **Pettre J**., **J. P. Laumond** and **T. Siméon**, "A 2-stages locomotion planner for digital actors", *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003, 258-264.
50. **Reeds J. A**. and **R. A. Shepp**, "Optimal paths for a car that goes forward and backwards, *Pacific Journal of Mathematics*, 145(2):367-393 (1990).
51. **Reif J. H.**, "Complexity of the mover's problem and generalizations", *Proc. IEEE Symp. on Foundations of Computer Science*, 1979, 421-427.
52. **Sánchez A. G**. and **J. C. Latombe**, "Using a PRM planner to compare centralized and decoupled planning for multi-robot systems, *IEEE Int. Conf. on Robotics and Automation*, 2002, 2112-2119.
53. **Sánchez A. G.** and **J. C. Latombe**, "On delaying colllision checking in PRM planning: Application to multi-robot coordination, *The International Journal of Robotics Research*, 21(1):5-26 (2002).
54. **Sánchez L. A**., **R. Zapata** and **A. Arenas B.**, "Motion planning for car-like robots using a lazy probabilistic roadmap method", *MICAI02, LNAI 2313*, 2002, 1-10.

55. **Sánchez L. A.**, **R. Zapata** and **C. Lanzoni**, "On the use of low-discrepancy sequences in non-holonomic motion planning", *IEEE Int. Conf. on Robotics and Automation*, 2003, 3764-3769.
56. **Sánchez L. A.**, *Contribution à la planification de mouvement en robotique: Approches probabilistes et approches déterministes (In French*), PhD thesis, Université Montpellier II, 2003.
57. **Sánchez L. A.**, **J. Sánchez T.** and **R. Zapata**, "Planning motions for animated characters", *Journal Research in Computing Science*, 17:265-274 (2005).
58. **Sánchez L. A**., **R. Juarez G**. and **M. A. Osorio L**., "On the performance of deterministic sampling in probabilistic roadmap planning", *MICAI07, LNAI 4827*, 2007, 1089-1098.
59. **Schwartz J. T.** and **M. Sharir**, "On the piano mover's problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers", *Communications on Pure and Applied Mathematics*, 36:345-398 (1983).
60. **Simeón T**., **J. P. Laumond**, **C. Van Geem** and **J. Cortes**, "Computer aided motion: Move3D within MOLOG", *IEEE Int. Conf. on Robotics and Automation*, 2001, 1494-1499.
61. **Siméon T.**, **J. Cortes**, **A. Sahbani** and **J. P. Laumond**, "A manipulation planner for pick and place operations under continuous grasps and placements", *IEEE Int. Conf. on Robotics and Automation*, 2002, 2022-2027.
62. **Song G**. and **N. M. Amato**, "Using motion planning to study protein folding pathways", *Proc. Int. Conf. Comput. Molecular Biology (RECOMB)*, 2001, 287-296.
63. **Svestka P.** and **M. H. Overmars**, "Probabilistic path planning", *Technical Report UU-CS-1995-22*, Utrecht University, 1995.
64. **Svestka P**. and **M. H. Overmars**, "Motion planning for car-like robots using a probabilistic learning approach", *The International Journal or Robotics Research*, 16(2):119-143 (1997).
65. Svestka P., *Robot motion planning using probabilistic roadmaps*, Phd Thesis, Utrecht University, 1997.
66. **Vallejo D**., **C. Jones** and **N. Amato**, "An adaptive framework for 'single shot' motion planning, *IEEE Int. Conf. on Intelligent Robots and Systems*, 2000, 1722-1727.
67. **Van der Stappen A. F**., **M. H. Overmars**, **M. de Berg** and **J. Vleugels**, "Motion planning in environments with low obstacle density", *Discrete & Computational Geometry*, 20:561-587 (1998).
68. **Van Geem C.**, **T. Siméon** and **J. P. Laumond**, "Mobility analysis for feasibility studies in CAD models of industrial environments", *IEEE Int. Conf. on Robotics and Automation*, 1998.
69. **Wilmarth S**., **N. M. Amato** and **P. R. Stiller**, "MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space", *IEEE Int. Conf. on Robotics and Automation*, 1999, 1024-1031.
70. **Yu Y.** and **K. Gupta**, "Sensor-based probabilistic roadmaps: Experiments with and eye-in-hand system, *Journal of Advanced Robotics*, 2000.



Abraham Sánchez López *received his M.S. degree in Computer Science from the Universidad de las Américas Puebla (UDLA) in 1996 and the PhD degree in Computer Science and Robotics from Université Montpellier II, Montpellier, France in 2003. He is currently a researcher at Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, and a member of the National Researchers System (SNI). His research interests include robot motion planning, mobile robots, computer animation and mobile computing.*

René Zapata *received his PhD degree in Robotics and the Habilitation degree from the University Montpellier II. He teaches linear and non-linear control and mathematics tools at the University Montpellier II (UM2) and more specifically at the Polytech Institute of Montpellier (Polytech). His research interests are focused on robot control, reactive behaviors of mobile robots, robot localization, path planning and vision-based control. He participates in NEtworked RObots project (NERO) regarding the design of fast reactive control algorithms.*



Maria Auxilio Osorio Lama *is a Chemical Engineer from the Universidad de las Américas, with a master degree in Systems from the Universidad Iberoamericana and master and doctorate degrees in Operations Research from the Universidad Nacional Autónoma de México. Her lines of research include the use of logic and heuristic optimization and engineering and financial applications in Operations Research. She belongs to the National System of Researchers with Level I, has published over 25 articles in first-level international journals. She has been guest editor in four volumes (2003, 2008) of Annals of Operations Research.*