RMP3D: A multipurpose Platform for Motion Planning

Josué Sánchez, Abraham Sánchez, Miguel Rodriguez, Xochitl Hernández and Gerardo García

Facultad de Ciencias de la Computación, BUAP
14 Sur esq. San Claudio, CP 72570
Puebla, Pue., México
asanchez@cs.buap.mx

Abstract. This paper reports on RMP3D, a multipurpose software platform dedicated to motion planning. RMP3D makes it possible to set up motion planning experiments very quickly. The generality comes from a dedicated software architecture allowing a rapid design of motion planners. Comprehension of concepts and algorithms involved in the robotics field can be improved through the use of an interactive visualization tool. We present an interactive tool for visualizing and editing motion planning environments, problem instances, and their solutions. RMP3D is specialized for model-based randomized planners such as Probabilistic Roadmap (PRM) methods. The paper focuses on recent results obtained in robot motion planning, graphics animation, and behavioral animation.

1 Introduction

Robot motion planning refers to the ability of a robot to automatically plan its own motions to avoid collision with the physical objects in its environment. Such a capability is crucial, since a robot accomplishes tasks by physical motion in the real world. This capability would be a major step toward the goal of creating autonomous robots. This observation has motivated much research in robot motion planning. The approaches to robot motion planning can be roughly divided into two categories: the classical motion planning or model-based motion planning, and sensor-based planning. The first approach, assumes that the robot system has an explicit representation of the robot's environment. On the other hand, in the second approach, the environment is unknown and the robot is guided directly from the sensory input without constructing internal representation for the environment. This work considers solely the model-based motion planning problem.

Motion planning problem is typically solved in the configuration space ($\mathcal{C}$), in which each placement of the robot is mapped as a point. The free configuration space, $\mathcal{F}$, is the subset of $\mathcal{C}$ at which the robot does not intersect any obstacle. The dimension of $\mathcal{C}$ depends on the degrees of freedom (dof) of the robot, which can be high. An exact computing of a high-dimensional configuration space is impractical. For this reason, a large family of model-based planners has been

© A. Gelbukh, S. Torres, I. López (Eds.)
Advances in Artificial Intelligence
Research in Computing Science 20, 2006, pp. 167-178

developed [1], [2], [3], [4]. These algorithms have been successfully used to solve challenging problems.

The most popular paradigm for model-based motion planning is the Probabilistic Roadmap Method (PRM) [1]. The key idea of the PRM is to randomly distribute a set of nodes in $\mathcal{C}$ 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 $\mathcal{F}$. If the roadmap is successfully capturing this connectivity, motion planning may be reduced to a graph search.

Motion planning has application in many other areas, such as assembly planning, design for manufacturing, virtual prototyping, computer animation, medical surgery simulation, computational biology, etc. As stressed by Latombe [5], non-robotics applications (e.g., graphics animation, surgical planning and computational biology) are growing in importance and are likely to shape future motion planning research at least as much as robotics.

In Section II, we present the software architecture of RMP3D. The following sections give practical results obtained in problems arising in fields as diverse as robot motion planning, planning motions for animated characters and behavioral animation. Finally, Section VI presents the conclusions and future work.

2 RMP3D Architecture

RMP3D is composed of diverse modules associated with functionalities such as the modeling of the mechanical systems (geometric modeling, steering methods), geometrical tools (collision detection, distance calculation), motion planning, and a graphic interface that allows to define the problems, call the algorithms, and to display the produced results. Fig. 1 shows the structure of the motion planning software RMP3D.

- the modeling module enables the user to describe mechanical systems and environments.
- the geometric tools for the collision detection algorithms.
- the steering methods allows to compute local paths satisfying the kinematic constraints of the mechanical systems.
- the planning algorithms module contains many procedures based on randomized techniques such as PRM, Lazy PRM, Gaussian PRM.

The following steering methods are actually integrated within RMP3D:

- **Linear** computes a straight line segment between two configurations, this method works for any holonomic system like a manipulator arm.
- **Nonholonomic** computes smooth paths for both car models, Reeds & Shepp [6] and Dubins [7] or articulated mobile robots.

Other methods could be easily integrated into this library. They can also be combined to design more complex steering methods for mechanical systems subjected to different motion constraints.

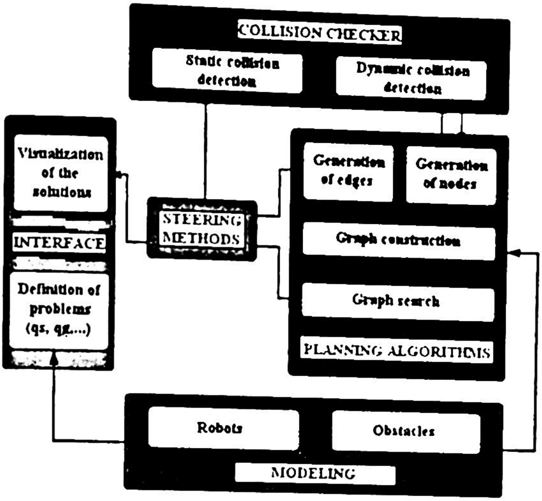

Fig. 1. Architecture of RMP3D

The planning module integrates three of the randomized techniques proposed recently. These techniques are based on the probabilistic roadmap methods that first construct a roadmap connecting collision-free configurations picked at random, and then use this roadmap to answer multiple or single queries. Additionally to the planning methods, we added some generators of determinist samples like Halton, Hammersley, Faure, Sobol, and Sukharev grid.

- Basic-PRM is based on the basic PRM scheme [1]. The key idea of this scheme is to randomly distribute a set of nodes in C and then connect these nodes using a simple local planner (or a steering method), to form a graph (or a tree) known as a roadmap.
- Gaussian-PRM is meant to add more samples near obstacles [8]. The idea is to take two random samples, where the distance between the sampless is chosen according to a Gaussian distribution. Only if one of the samples lies in the $\mathcal{F}$ and the other lies in C_{obst} do we add the free sample. It has been shown that this leads to a favorable sample distribution.
- Lazy-PRM [9], the idea is not to test whether the paths are collision free unless they are really needed. The goal of this variant is to minimize the number of collision checks. The rational behind this is that for most paths we only need to consider a small part of the graph before a solution is found.

The collision checker integrated is PQP[1] for determining whether a given path is collision-free or not (this is performed by multiple calls to the interference detection algorithm). PQP is a library for performing three types of proximity queries on a pair of geometric models composed of triangles.

[1] A collision detection package from University of North Carolina at Chapel Hill

All techniques were integrated in a single motion planning system called RMP3D, implemented in Builder C++ under Windows XP. RMP3D automates conducting experiments, i.e. statistics are automatically generated and processed, decreasing the chance for errors. All experiments were run on a 2.4 GHz Pentium 4 processor with 512 MB internal memory.

3 Robot Motion Planning

In this section, we discuss the field of model-based motion planning. In contrast to methods that construct boundary representations of configuration space obstacles, model-based methods[2] use only information from a collision detector as they search the configuration space. The simplicity of this approach, along with increases in computation power and development of efficient collision detection algorithms, has resulted in the introduction of a number of powerful motion planning algorithms, capable of solving challenging problems with many degrees of freedom (dofs).

All of the recent methods rely on some method for generating samples over the configuration space. Typically, the samples are taken at random from a statistically uniform distribution; however, this method of sampling is not as uniform as some deterministic methods [10]. Several weakness of random sampling were shown in the context of the PRM in [11], [12], [13].

Despite the success of PRM planners, narrow passages in a robot's configuration space create significant difficulty for PRM planners. A narrow passage is a small region whose removal changes the connectivity of the free space. There are several sophisticated sampling strategies that can alleviate this difficulty, but a satisfactory answer remains elusive. Indeed, many of them only solved partially the motion planning problem, e.g., the case of free-flying robots. We claim that deterministic sampling is suitable to capture the connectivity of configuration spaces with narrow passages [10]. Figures 2 and 3 show the stages of the PRM approach.

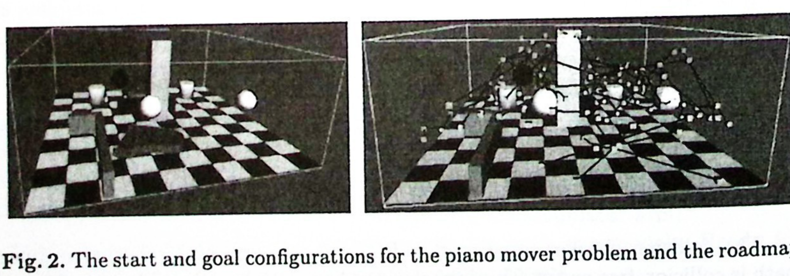

Fig. 2. The start and goal configurations for the piano mover problem and the roadmap graph

[2] These methods are called sampling-based methods

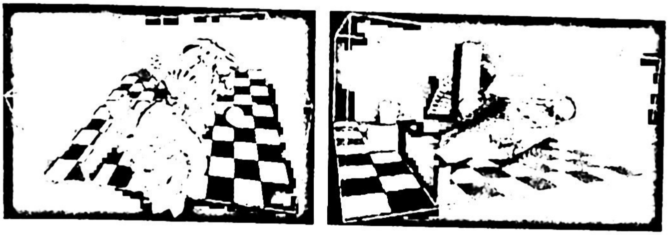

Fig. 3. An un-smoothed path for the piano problem and the path obtained by smoothing

Figures 4 and 5 show some resolute examples, the examples correspond to free-flying objets in a three-dimensional workspace, the objects have six dofs, three translational degrees and three rotational degrees.

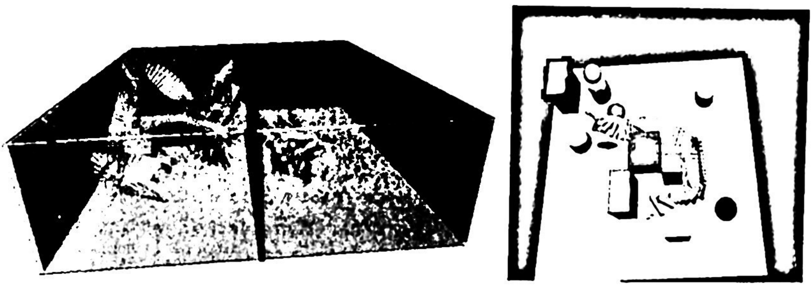

Fig. 4. Left, a famous test scene, in which an elbow-shaped robot passes through a small opening. Right, the same elbow-shaped robot in other environment

The PRM approach has been applied to many different types of motion planning problems involving different kinds of robots. Unfortunately, the different improvements suggested are difficult to compare. Each author used his or her own implementation of PRM. Also the effectiveness of one technique sometimes depends on choices made for other parts of the method.

With this tool, we provide a comparative study of a number of PRM techniques, all implemented in a single system and run on the same test scenes and on the same computer. In particular, we can compare basic sampling techniques, steering methods and node adding techniques.

4 Planning Motions for Animated Characters

The design of autonomous characters capable of planning their own motions continues to be a challenge for computer animation. We present a novel method

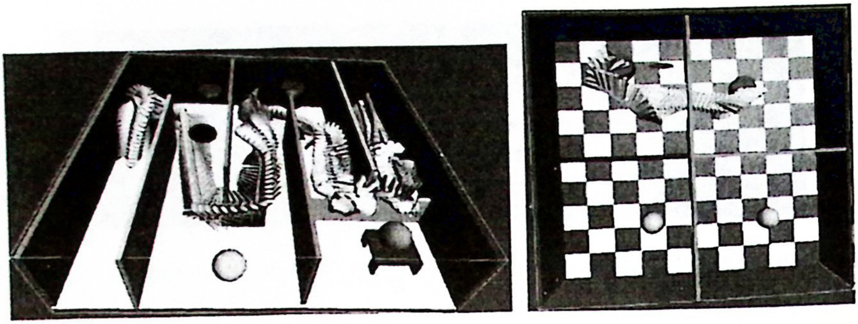

Fig. 5. Two scenes with narrow passages. Left, a spaceship in a labyrinth. Right, an object with complicated geometry in four roms with lots of space and with narrow doors between them

for animating human characters. Our method is integrated in a probabilistic roadmap method scheme. The navigation of the human character in this environment is modeled by a composition of Bezier curves. The controllers are based on motion data capture techniques.

Given start and goal positions in a virtual environment, our objective is to find a sequence of motions of a human character to move from the start and to the goal. Conventional motion planning techniques in robotics typically generate very efficient mechanical movements rather that lifelike natural-looking motions desired in computer animation applications. On the other hand, motion editing techniques in computer graphics are not equipped with a high-level planning capability to yield a desired motion. To rapidly plan convincing motions of the human-like character with high-level directives.

Although the motion planning and following concept generally applies to many types of characters and motions, we will concentrate on generating walking or running motions for a human-like character. We would like the character's motion to be smooth and continuous, natural-looking, and follow the computed path as closely as possible. For more details on this technique, you can review the work in [14]. Our procedure consists of the following three steps: roadmap construction, roadmap search, and motion generation. Figure 6 shows a high-level description of the proposed approach.

A result of complete trajectory (composition of several local paths) is presented on figure 7. One can notice several specificities in this result: the model strictly respects the initial and final configurations required. The structure of the character is modeled in two levels. Pelvis and legs are used for the locomotion, all the 18 dofs are said to be active dofs. The 34 other ones are said to be reactive dofs, they deal with the control of the arms and the spine. The pelvis is the root of five kinematics chains modelling respectively the arms, the legs and the spine.

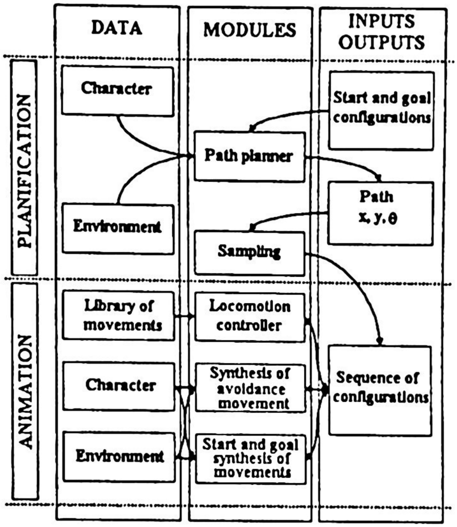

Fig. 6. High-level description of our approach

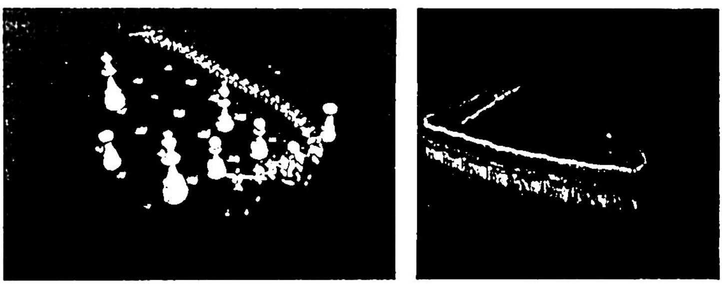

Fig. 7. Walking through the chessboard and the labyrinth

Figure 8 illustrates the result of the warping module, the goal of the warping module is to locally modify the animation of the upper bodies of the character (arms and spine) when collision occur in the animation produced by the module locomotion-controller.

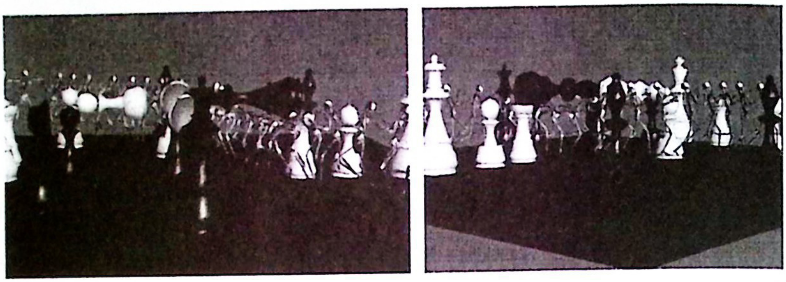

Fig. 8. Two snapshots to show the effectiveness of the warping module

The approach satisfies some computer graphics criteria such as realistic results, collision-free motion in cluttered environments and low response time. The approach has been implemented in our RMP3D architecture and successfully demonstrated on several examples. The combination of randomized motion planning techniques and motion capture editing techniques offer promising results.

We have presented a solution for digital actors locomotion problems. The solution insists on the modularity of the approach. Each component can be modified or replaced with ease. Our locomotion planner is still to be enhanced: we want to introduce the ability for the digital actor to change its locomotion behavior by crouching, crawling, etc.

Motion capture system have been widely used during the last few years for creating new animations of virtual humans. These approaches assume that realistic motions can be obtained by editing and tuning library of motion data. The computational cost is low ($O(n)$) because each articulation is computed separately. Although several convincing animations have been designed using these methods, the parameter control is not simple: a number of trial and error iterations is often needed before obtaining the desired result.

5 Behavioral Animation

Behavioral animation is a type of procedural animation, which is a type of computer animation. In behavioral animation an autonomous character determines its own actions, at least to a certain extent. This gives the character some ability to improvise, and frees the animator from the need to specify each detail of every character's motion. These improvisational characters are, in the words of Ann

Marion: "puppets that pull their own strings". An early example of behavioral animation was the 1987 boids model of bird flocking [15]. While in some limited sense autonomous characters have a mind, their simplistic behavioral controllers are more closely related to the field of artificial life than to artificial intelligence.

Like particle systems, behavioral animation is a vague term which refers to a number of techniques. Also like particle systems, behavioral animation is used to control the motion of many objects automatically. The primary difference is in the objects being animated. Instead of simply procedurally controlling the position of tiny primitives, motion is generated for actors with orientations, current state, and interactions. Behavioral animation has been used to animate flocks, schools, herds, and crowds. All of these require interaction between a large number of characters with relatively simple, rule-based motion. Fabric can also be simulated using behavioral techniques.

While many methods to simulate flocking behaviors have been proposed, these techniques only provide simplistic navigation and planning capabilities, because each flock member's behavior depends only its local environment [16], [17].

In this section, we show that complex group behavior can be generated using a roadmap providing global environment information. The roadmap contains topological information and adaptive edge weights that enables the flock to achieve behaviors that cannot modeled with local information alone. We propose new techniques for different group behaviors: homing, goal searching, and traversing narrow areas. We extended ideas from cognitive modeling to create behavior rules in individual flock members and in the roadmap. These embedded behaviors enable the creatures to modify their actions based on their current location and state.

These behaviors exploit global knowledge of the environment and utilize information gathered by all flock members which is communicated by allowing individual flock members to dynamically update the shared roadmap to reflect (un)desirable routes or regions.

Homing behavior consists of two sub-models, one representing the individual behavior of flock members and the other influencing the global behavior of the flock. Once a path is found (with our RMP3D tool), individual flock members follow the path. The path is discretized to subgoals based on and individual flock member's sensor range. Each member keeps track of subgoals and as soon as a subgoal comes within the sensory range the next subgoal becomes the steering direction for the global goal. With other interacting forces from neighboring flock members and obstacles, steering toward the subgoal has the lowest priority, so individual members still move together while moving toward the goal. This results in a flocking toward the goal and avoids getting trapped in local minima.

Goal searching is a type of exploring behavior. We achieve this behavior using a roadmap graph with adaptive edge weights. Each individual member behaves independently from its flock mates and uses the roadmap to wander around. They follow roadmap edges and there are no predefined paths. If they reach a roadmap node with several roadmap edges, they probabilistically choose a roadmap edge to follow based on weight of the edge. The edge weights represent any preferences for the current task, i.e., searching for and reaching the goal.

A naive solution to achieve narrow passage traversal by the flock is to use the homing behavior and to select two nodes as goals, first a node in front of the entrance to the passage and then a node outside the exit from the passage. One drawback is that flock members may bunch up and conflict with each other as they try to move through the passage. A strategy that may avoid the congestion problems of the naive idea is the follow-the-leader strategy. We first assemble the flock in front of the narrow passage, and then select the closest flock member to the entrance to the narrow passage as the leader. The remaining flock members are arranged into a queue that follows the leader.

Figure 10 show some results obtained with our approach. The behavior rules embedded in our roadmaps enable the flocks to modify their actions based on their current location and state. Our simulation results for the three types of behaviors studied show that the performance of the rule-based roadmap behaviors is very close to ideal behaviors that have complete knowledge.

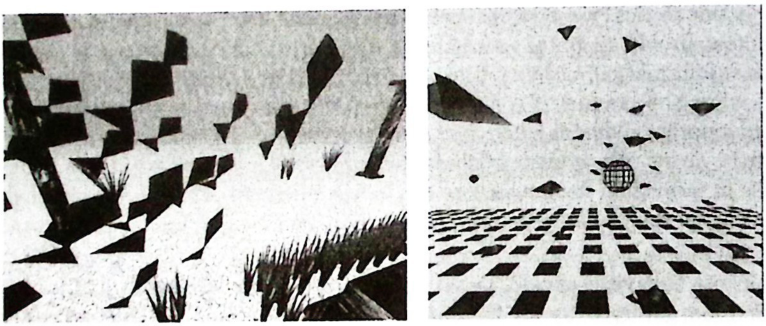

Fig. 9. Complex behaviors, such goal searching can be improved using global information

6 Conclusions and Future Work

Motion planning has been an active research field in robotics for more than 20 years. Within the 80's, roboticians addressed the motion planning problem by devising a variety of heuristics and approximate methods. While complete and deterministic algorithms for motion planning are very time-consuming as the dimension of the configuration space increases, it is now possible to address complicated problems in high dimension thanks to alternative methods (for instance PRM methods) that relax the completeness constraint for the benefit of practical efficiency and probabilistic completeness.

This paper presented the software platform RMP3D developed at the University of Puebla (Facultad de Ciencias de la Computación) for generic multipurpose

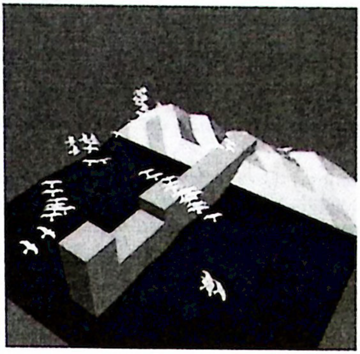

Fig. 10. Complex behaviors, such Homing can be improved using global information

applications. A large number of variant PRM planners can be constructed by combining the algorithmic components presented in our software platform.

RMP3D is a multipurpose tool for visualizing and editing motion planning environments, problem instances, and their solutions. RMP3D offers a self-explanatory graphical user interface that provides functionality that can help us understand robotics and motion planning concepts. We are convinced that researchers and students can take advantage of visualization tools such as RMP3D which in addition to visualization, allows interaction with the robots workspace.

The aim of the projet RMP3D is to develop a general planning software for providing systems with motion planning facilities. The examples shown in the paper illustrate the kind of problems that can be solved today by the algorithms integrated within RMP3D. It remains that additional work still needs to be done for improving the efficacy of the planning techniques. Another challenging issue is to develop more sophisticated planning algorithms (i.e., multiple robots, manipulation planning).

Motion planning applications are emerging today in various domains such as computer graphics, drugs design, medicine...

Finally, our paper does not introduce new algorithms, nor new analysis. It should be viewed as an experience feedback in developing motion planning technology within the context of a well focused application field. The advantages of our platform are many with respect to other proposals, but the most important thing is the level of the modelling (environments and robots), since this is possible by using Inivis AC3D.

References

1. Kavraki L. E., Švetska P., Latombe J. C., and Overmars M. H. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, (1996) 566-580

2. Švestka P. and Overmars M. H. "Motion planning for car-like using a probabilistic learning approach", *The International Journal of Robotics Research*, Vol. 16, No. 2, (1997) 119-143
3. Siméon T., Laumond J. P., and Nissoux C. "Visibility-based probabilistic roadmaps for motion planning", *Journal of Advanced Robotics*, Vol. 14, No. 6, (2000) 477-494
4. Sánchez G., and Latombe J. C. "On delaying colllision checking in PRM planning: Application to multi-robot coordination", *The International Journal of Robotics Research*, Vol. 21, No. 1, (2002) 5-26
5. Latombe J. C. "Motion planning: A journey of robots, molecules, digital actors, and other artifacts", *The International Journal of Robotics Research*, Vol. 18, No. 11, (1999) 1119-1128
6. Reeds J. A., and Shepp R. A. "Optimal paths for a car that goes both forward and backwards", *Pacific Journal of Mathematics*, Vol. 145, No. 2, (1990) 367-393
7. 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, Vol. 79, (1957) 497-516
8. Boor V., Overmars M. H., and Van der Stappen A. F. "The Gaussian sampling strategy for probabilistic roadmap planners", *Proc. of the IEEE Robotics and Automation Conference*, (1999) 10181023
9. Bohlin R. and Kavraki L. E. "Path planning using lazy PRM", *Proc. of the IEEE Robotics and Automation Conference*, (2000) 521-528
10. Sánchez L. A. "Contribution à la planification de mouvement en robotique: Approches probabilistes et approches déterministes", *PhD thesis, Université Montpellier II* (in French) (2003)
11. Branicky M. S., LaValle S. M., Olson K., and Yang L. "Quasi-randomized path planning", *Proc. of the IEEE Robotics and Automation Conference*, (2001) 1481-1487
12. Geraerts R. and Overmars M. H: "A comparative study of probabilistic roadmap planners", *Proc. of Workshop on the Algorithmic Foundations of Robotics* (2002)
13. Sánchez A., Zapata R., and Lanzoni C. "On the use of low-discrepancy sequences in nonholonomic motion planning", *Proc. of the IEEE Robotics and Automation Conference*, (2003) 3764-3769
14. Sánchez T. J. "Animación de personajes virtuales utilizando técnicas probabilisticas", *BS thesis, FCC - BUAP*, (in Spanish) (2005)
15. Reynolds C. W. "Flocks, herds and schools: A distributed behavioral model", *Computer Graphics*, Vol. 21, No. 4, (1987) 25-34
16. Ward C. R., Gobet F., and Kendall G. "Evolving collective behavior in an artificial ecology", *Artificial Life*, (2001) 191209
17. Funge J., Tu X., and Terzopoulos D. "Cognitive modeling: Knowledge, reasoning and planning for intelligent characters", *Computer Graphics*, (1999) 29-38