



HAL
open science

Towards a software development kit for motion synthesis in virtual worlds

Thierry Simeon, Jean-Paul Laumond, Florent Lamiroux

► To cite this version:

Thierry Simeon, Jean-Paul Laumond, Florent Lamiroux. Towards a software development kit for motion synthesis in virtual worlds. 7th International Conference on Virtual Systems and Multi-Media (VSMM), Oct 2001, Berkeley, United States. hal-04295924

HAL Id: hal-04295924

<https://laas.hal.science/hal-04295924v1>

Submitted on 20 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a software development kit for motion synthesis in virtual worlds *

T. Siméon, J-P. Laumond, F. Lamiroux
LAAS-CNRS
7, avenue du Colonel-Roche
31077 Toulouse Cedex - France
{nic,jpl,lamiroux}@laas.fr

Abstract

This paper reports on Move3D, a software platform dedicated to collision-free path planning in virtual worlds. The goal of Move3D is to address generic multipurpose applications with the same software, ranging from logistics in industrial installation to character animation, via game design. The algorithms are based on probabilistic approaches and take advantage of the progress of the computer performance. The generality comes from a dedicated software architecture allowing a rapid design of path planners.

1: The piano mover problem

The collision-free path planning problems covers several facets of what the research community calls the *piano mover problem*. Deciding whether a body can reach a position from a starting point while avoiding the environment obstacles, computing an admissible path, this is the standard piano mover problem.

Figure 1 shows an exemple modeled from an existing puzzle. At the beginning (figure on the left) both yellow and blue parts are intricate. On the right picture they are not. The central picture shows the volume swept by the yellow part when escaping its initial position. This problem is known to be rather difficult. Algorithms are today capable to solve the problem automatically.



Figure 1. How to disassemble yellow and blue parts? This is a path planning problem.

*This paper is built upon previous work presented at [33]

Consider now the mechanical systems displayed in Figure 2 together with examples of collision-free paths. Modeling a mechanical system in the context of path planning addresses two issues.

The first one deals with the placement constraints. We should identify the configuration space CS of the system, i.e. a minimal set of parameters locating the system in its workspace. This is an easy task for the robot arm, the mobile robot and the rolling bridge, all of them being open kinematic chains. The system constituted by the two mobile robots manipulating an dumbbell is a closed kinematic chain. The placement parameters of both robots are linked by equations modeling the grasping of the dumbbell and giving rise to constraints. For this special example it is possible to select five independent parameters defining the configuration space properly, the other placement parameters being deduced by explicit equations involving only the five independent parameters. For more general closed loop chains, characterizing CS properly is not an easy task.

Second, we should consider the kinematic constraints. There is no special constraint for the robot arm, any configuration parameter being a degree of freedom directly controlled by a motor independently from any other one. Any path in CS is admissible. The same property holds for the coordinated path of the two mobile robots. The control of the rolling bridge may impose special constraints, like moving a degree of freedom at once. The motion of the mobile manipulator is submitted to the constraint of rolling without sliding. This is a constraint that affects the range of admissible paths, but not the dimension of the reachable configuration space.

Path planning then consists in computing admissible paths in the configuration space of the considered system.

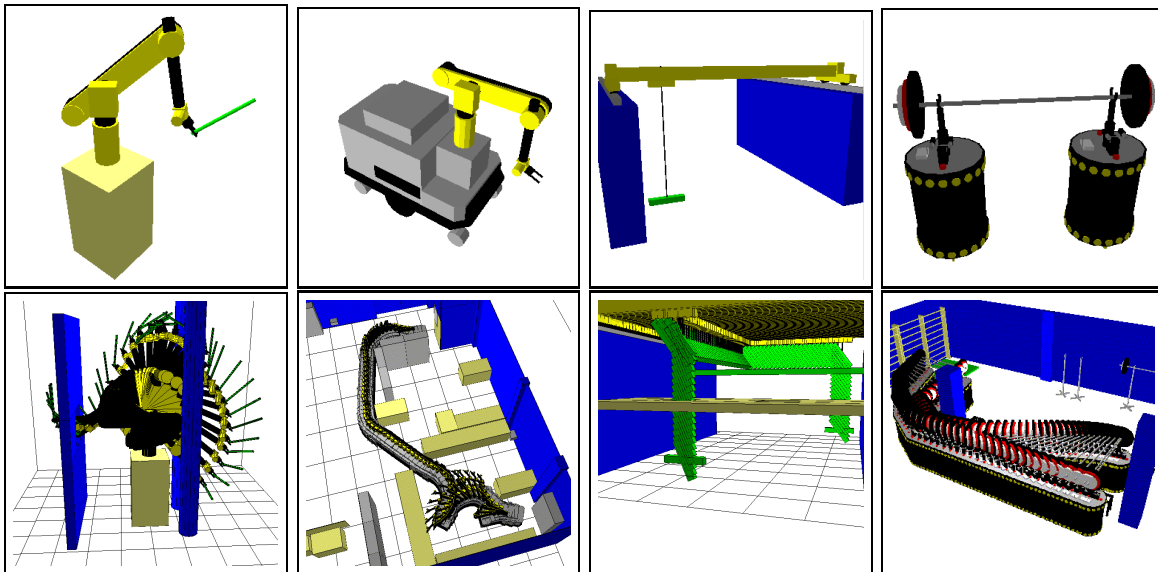


Figure 2. Examples of mechanical systems and collision-free paths.

This paper presents the software platform Move3D currently developed at LAAS for generic multipurpose applications. From an algorithmic view point, Move3D is based on probabilistic roadmap algorithms briefly reviewed in Section 2. Then we present the soft-

ware architecture of Move3D implementing the various ingredients required by the roadmap algorithms: collision-checkers, steering methods, roadmaps builders (Section 3). The following sections give practical results obtained in problems arising respectively in logistics for industrial installations, in graphics animation or game design.

2: Algorithms

Path planning is an active research fields in robotics for more than 20 years [26]. Algorithms become more and more mature. Moreover, they take advantage of the increasing progresses of the computer power: with the same algorithms, it is today possible to solve problems which were out of scope five years ago. The range of the potential applications of path planning algorithms becomes broader.

The probabilistic roadmap algorithms first introduced in [19, 30, 20] and now investigated by numerous researchers [3, 4, 5, 8, 9, 16, 17, 21, 22, 32] is a general scheme to devise path planners. A roadmap is a graph that tends to capture the connectivity of the collision-free configuration space. The nodes are collision-free configurations while the edges indicate the existence of an admissible collision-free path between two configurations. Roadmaps are computed in a probabilistic manner by selecting the nodes of the graph randomly. Implementing such approaches only requires two basic geometric procedures: a steering method to compute admissible paths for a given mechanical system and a collision-checker which is used both to select the nodes of the roadmap and to check whether an admissible path is collision-free or not.

The main ingredients to devise a path planner are then:

- the steering methods: they account for the kinematic constraints of the considered mechanical system (e.g., nonholonomic constraints) or the kinematic constraints imposed by the application (e.g., moving a degree of freedom at once). The inputs of a steering method are two configurations of the mechanical systems, the output is an admissible path linking them and computed in the absence of obstacle.
- the roadmap builder: this is the algorithm used to compute a roadmap. Two collision-free configurations being randomly chosen, the algorithm checks whether the admissible path computed with the selected steering method is collision-free or not.
- basic geometric tools: they are mainly used to check whether a given configuration is collision-free or not (interference detection) and whether a given path is collision-free or not (collision checking).

Once a roadmap has been computed, a query phase is used to solved specific problems. A start and goal configurations being given, the query phase first checks if the configurations can be added to the roadmap with collision-free admissible paths. Then, the procedure operates a graph search algorithm to find a global solution. If the graph search succeeds then we obtain a first solution path. This path is then optimized: the goal is to shorten it avoiding the unusefull detours induced by the random search.

The following section presents an architecture implementing such principles.

3: Move3D architecture

Figure 3 shows the global architecture of the motion planning software Move3D. Move3D is composed of the following set of modules:

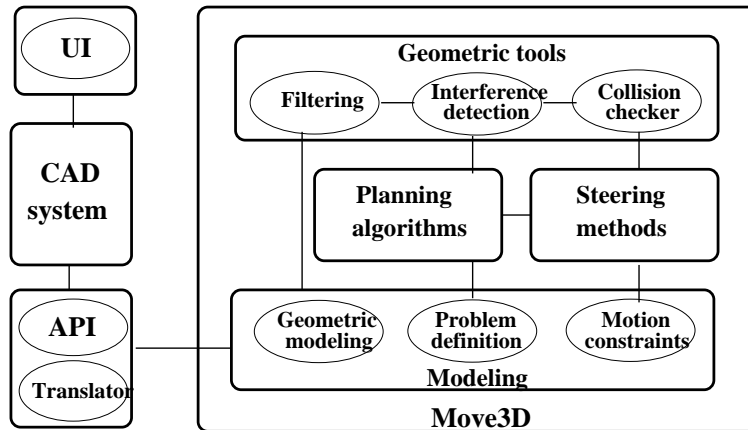


Figure 3. Architecture of Move3d

- mechanical systems and environment modeling, definition of the motion planning problem. The input of this module is provided through the API used to connect Move3D to the external CAD system and to transform the geometric data of the CAD system into the geometric data structures of Move3D
- geometric tools for filtering the geometric database and initializing the interference detection algorithms only with the part relevant to the specified planning problem.
- the library of steering methods allowing to compute local paths admissible with respect to the own motion constraints of the mechanical systems involved by the planning problem.
- the planner module contains several algorithms based onto randomized techniques (see next section) for computing collision-free paths.
- finally, Move3D has its own Developer Interface module (not displayed onto the Figure) allowing to call the algorithms and to visualize the solutions independently of any external CAD system.

3.1: Steering methods

The following methods illustrated by Figure 4 are today integrated within Move3D:

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 carts [31] or articulated mobile robots [25].

Manhattan accounts for the constraints to move one degree of freedom at once.

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

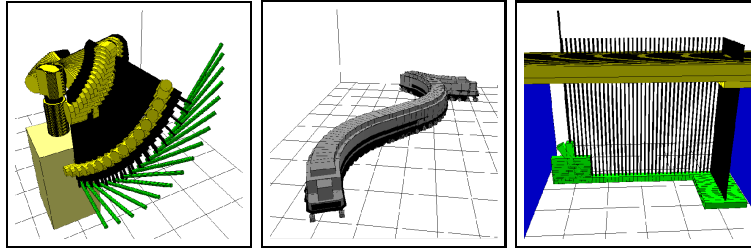


Figure 4. Examples of elementary steering methods

constraints (eg. for a mobile manipulator, the mobile platform and the manipulator are respectively controlled by the **Nonholonomic** and the **Linear** methods).

Some mechanical systems also require to consider passive joints that can not be directly controlled, but have to follow the motion of other leading joints. In presence of such motion constraints, the steering method only acts onto the leading joints, while the passive ones are computed by dedicated “follow” functions that express the coupling relations.

3.2: Motion planners

This module currently integrates four of the randomized planning techniques proposed in the literature. Three of them share the underlying concept of the Probabilistic Roadmap Methods that first construct a roadmap connecting collision-free configurations picked at random, and then use this roadmap to quickly answer multiple queries.

Basic-PRM is based onto the basic PRM scheme [20]. The search space is uniformly sampled at random. All collision-free samples are added to the roadmap and checked for connections with all connected components. The planner offers the choice between several strategies for selecting the promising nodes inside the components, and allows the tuning of several parameters (adjacency neighborhood, size of the roadmap...).

Visib-PRM computes visibility roadmaps [32] that are bipartite graphs defined with two types of nodes: the guards and connectors. Collision-free samples are kept as a new guard node when they cannot be connected to the current roadmap, or as a new connector if they improve the connectivity of the roadmap. An interest of this algorithm is the small size of the computed roadmaps. Another one is the possibility to control the quality of the roadmap in term of coverage.

Gaussian-PRM [9] uses a non-uniform sampling strategy in order to create a higher density of nodes near the boundary of the free-space. The gaussian sampler generates pairs of configurations separated by a random distance. It only retains a collision-free configuration of the pair when the other one lies in the collision space. This gaussian sampling strategy can be used when computing basic or visibility roadmaps.

The last planner is based onto the **RRT-Connect** algorithm [24] that was designed to answer single-query problems without requiring the preprocessing of the roadmap. The algorithm develops simultaneously two random trees rooted at the initial and goal configurations, and that explore the space while advancing toward each other.

3.3: Geometric algorithms

A motion planning task is often localized on a small part of the environment. The role of the **Filter** is to pre-process the geometric data basis to extract the only bodies that may interact when searching a collision-free path. The filter either uses inputs given by the operator (e.g., to constrain the path to lie in a user-defined area) or automatically computes the workspace spanned by the bodies to be moved. Consider for instance the problem illustrated in Figure 5. The filter automatically removes the subset of the obstacles that cannot interact with the crane. It also determines that most of the remaining obstacles have to be checked for potential collisions with the freight, but determines that no obstacle can collide with the first vertical body of the crane and very few with the second horizontal one.

It remains that the filtered model may still include a huge number of geometric primitives. It is well known that the most time consuming operation in motion planning are interference detection and collision-checking along a given path.

The algorithms [13] developed within Move3D for the **Interference-detector** combine techniques proposed in [27] while allowing to process non convex polyhedra together with other volumic primitives (eg. spheres, tubes, torus...) as in [7]. Computation is performed in two stages: selection of possibly colliding pairs based onto simple bounding volumes that approximate the geometry of the objects, and a precise interference detection limited to the pairs selected at the first stage. A hierarchical structure based onto OBB-trees is constructed on top of the convex primitives (or facets), instead of the triangular decomposition required by [14]. This allows to reduce the size of the data structures when facing with large CAD models where most of the objects (eg. piping) are modeled by a collection of simple primitives. Also, CAD systems often group into a same object several components sharing the same semantic but localized at different places within the scene (eg. one single object may represent the piping part of the installation). Such components are therefore grouped according to their workspace occupancy to improve the efficacy of the hierarchical models.

The **Collision-checker** [13] developed for determining whether a given path is collision-free or not, is currently performed by multiple calls to the interference detection algorithms, using a dichotomic sampling of the path and a non uniform step computed from the distance to the obstacles.

4: Application 1: Operation in industrial installation

Today CAD systems are widely used in manufacturing and more generally to help the operation of complex systems and complex tasks. They are supported by powerful dedicated software including 3D visualization possibilities, geometric tools and friendly interfaces. In the framework of logistics of industrial installations, CAD systems provide tools for the manipulation and storage of plant layout and design information.

The industrial installation of the first example is a stabilizer (subset of a plant in chemical industry) modeled by 300.000 facets. The rotating crane (around 1000 facets) has to place a tank inside the metallic structure, starting from an initial position where the tank is horizontally placed onto the ground. Here the steering method produces **Manhattan** paths for the crane, while accounting for the constrained motion of the tank which slides onto

the ground before it reaches a vertical position.

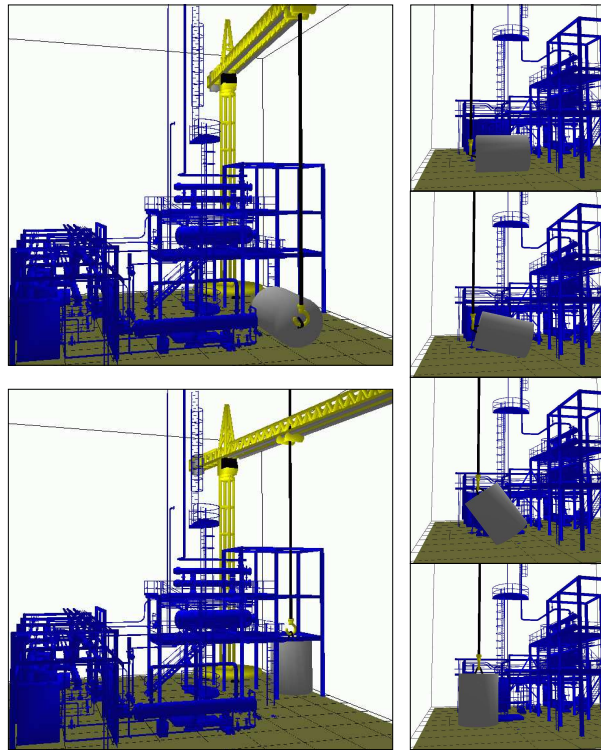


Figure 5. Stabilizer environment with a rotating crane (model provided by Cadcentre)

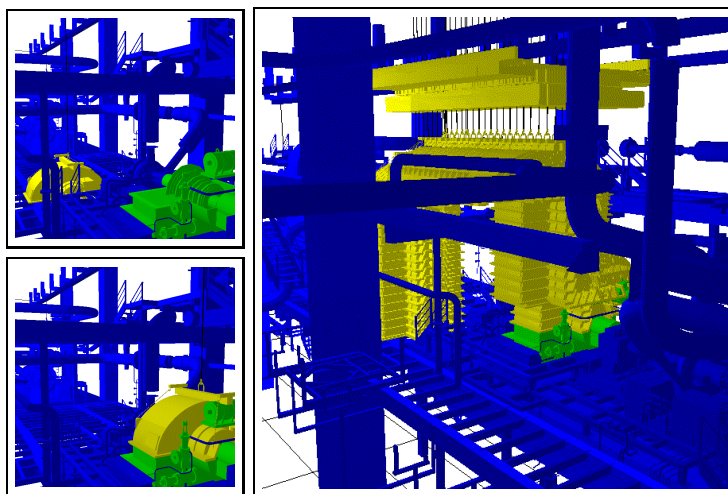
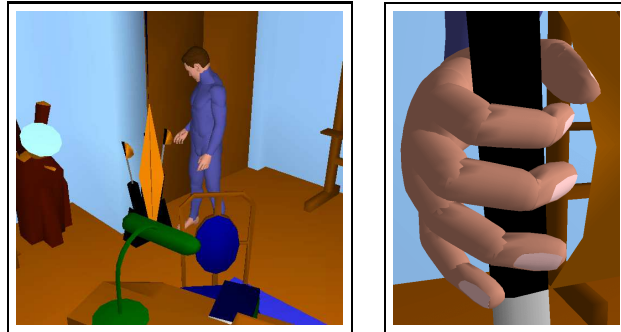


Figure 6. Travelling crane in the engine room of a nuclear plant (model provided by EDF)

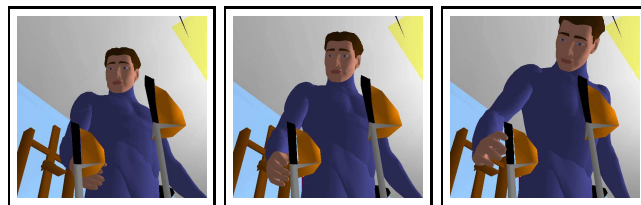
The second example shows a partial view of the engine room of a nuclear plant. Here, the purpose of the maintenance operation is to repair the water turbopump composed of three elements which have to be moved to a lower floor with a traveling crane. The path displayed onto Figure 6 corresponds to the motion of the traveling crane computed to grasp the first

element of the pump. For this scene (same geometric complexity than the stabilizer), the construction of the roadmap took less than ten minutes allowing queries to be processed in a couple of seconds.

5: Application 2: Animating characters



Initial configuration and detail of the goal



Intermediate configurations computed by Move3D

Figure 7. Grasping a sword (environment provided by Dramaera)

Providing realistic animation of human characters in graphics takes a lot of time. The automated synthesis of fine motions (e.g., grasping an object) may help the operator. In the example of Figure 7 the goal configuration is not explicitly specified by the operator (think to the huge number of parameters to tune to fully specify the position of all the fingers). The operator just gives a preshaping position roughly defined around the grip of the sword. Then Move3D automatically computes a collision-free path, including the automated computation of the final configuration (see [12] for details).

6: Application 3: Motion compression in game design

3D motion engines integrating collision-free capabilities may result in less repetitive motions of characters allowing to design more sophisticated games and also to speed-up the game design process. The example of Figure 8 illustrates some motion compression capabilities of the planning algorithms developed in Move3D: all the admissible motions of the spacecraft in the environment made of 60000 polygons can be stored into a 20 Kbytes roadmap data structure, allowing to compute in real time collision-free motions of the spacecraft to any position within the environment.



Figure 8. Spacecraft in a game environment (model provided by Kalisto).

7: Conclusion

The aim of the work on the software platform Move3D is to develop a general planning software for providing systems with motion planning facilities. The challenge for application fields such as logistics or graphics is to face real size problems involving a large class of mechanical systems. The examples shown in the paper illustrate the kind of problems that can be solved today. They are representative of the current state of the art offered by the software.

Acknowledgment

The practical results have been obtained from the generic motion planning software platform Move3D developed in collaboration with C. Nissoux. We thank EDF and Cadcentre for providing models of industrial scenes. We also thank T. Gaudin for having provided the example of Figure 7.

References

- [1] J.M. Ahuactzin, K.K. Gupta and E. Mazer. Manipulation planning for redundant robots: a practical approach. *Practical Motion Planning in Robotics*, K.K. Gupta and A.P. Del Pobil (Eds), J. Wiley, 1998.
- [2] R. Alami, J.P. Laumond and T. Siméon. A geometrical approach to planning manipulation tasks. In *Robotic Research: the fifth international symposium*, Tokyo (Japan), 1989.
- [3] N. Amato, O Bayazit, L. Dale, C. Jones and D. Vallejo. OBPRM: an obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective (WAFR98)*, P. Agarwal and all (Eds), AK Peters 1998.
- [4] N. Amato, O Bayazit, L. Dale, C. Jones and D. Vallejo. Choosing good distance metrics and local planners for probabilistic roadmap methods. *IEEE International Conference on Robotics and Automation*, Leuven (Belgium), 1998.
- [5] J. Barraquand, L. Kavraki, J.C. Latombe, T.Y. Li, R. Motvani and P. Raghavan. A random sampling scheme for path planning. In *Robotics Research*, G. Giralt and G. Hirzinger (Eds), Springer, 1996.
- [6] J. Barraquand and J.C. Latombe. Robot motion planning: a distributed representation approach. In *International Journal of Robotics Research*, vol 10 n 6, 1991.

- [7] G. van den Bergen. A fast and robust GJK implementation for collision detection of convex objects. In *Journal of Graphics Tools*, vol 4 n 2, 1999.
- [8] P. Bessiere, J. Ahuactzin, T. El-Ghazali and E. Mazer. The “Ariane’s clew” algorithm: Global planning with local methods. In *IEEE International Conference on Robots and Systems*, 1993.
- [9] V. Boor, M. Overmars, A. Frank van der Stappen. The Gaussian Sampling Strategy for Probabilistic Roadmap Planners. In *IEEE International Conference on Robotics and Automation*, Detroit, MI, 1999.
- [10] R. Davies, M. Overmars, V. Boor, A. Kamphuis, C. Van Geem, and T. Siméon. API: Requirements and architecture for motion planning interaction with PDMS. MOLOG Report A.1.1, 1999.
- [11] W. Du, A. Drouin, L. Masquelier. User Interface Definition. MOLOG Report A.2.1, 1999.
- [12] T. Gaudin. Synthèse de mouvement pour un personnage animé. Master Thesis, LAAS-CNRS, Toulouse, November 2000 (in french).
- [13] C. van Geem, T. Siméon. A Practical Collision Detector for Path Planning in Factory Models. *LAAS report 011073*, March 2001.
- [14] S. Gotschalk, M.C. Lin and D. Manocha. OBB-Tree: A hierarchical structure for rapid interference detection. In *ACM Siggraph ’96*, 1996.
- [15] D. Halperin, L. Kavraki and J.C. Latombe. Robotics. In *Handbook of Discrete and Computational Geometry*, J.E. Goodman and J. O’Rourke (Eds), CRC Press, 1997.
- [16] D. Hsu, J.C. Latombe, R. Motwani. Path planning in expansive configuration spaces. In *IEEE International Conference on Robotics and Automation*, Albuquerque (USA), 1997.
- [17] D. Hsu, L. Kavraki, J.C. Latombe, R. Motwani and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Robotics: The Algorithmic Perspective (WAFR98)*, P. Agarwal et al (Eds), AK Peters, 1998.
- [18] A. K. Jones. CAD geometry algorithms in a large industrial enterprise. In *Algorithms for Robotic Motion and Manipulation (WAFR96)*, J.P. Laumond and M. Overmars (Eds), AK Peters, 1997.
- [19] L. Kavraki and J.C. Latombe. Randomized preprocessing of configuration space for fast path planning. In *IEEE International Conference on Robotics and Automation*, San Diego (USA), 1994.
- [20] L. Kavraki, P. Švestka, J.-C. Latombe and M.H. Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces, In *IEEE Transactions on Robotics and Automation vol 12, n4, 1996*.
- [21] L. Kavraki and J.C. Latombe. Probabilistic roadmaps for robot path planning. In *Practical Motion Planning in Robotics*, K. Gupta and A. del Pobil (Eds), Wiley Press, 1998.
- [22] L. Kavraki, M. Kolountzakis and J.-C. Latombe. Analysis of Probabilistic Roadmaps for Path Planning, In *IEEE Transactions on Robotics and Automation vol 14, n1, 1998*.
- [23] Y. Koga, K. Kondo, J. Kuffner and J.C. Latombe. Planning motion with intentions. In *ACM SIGGRAPH*, Orlando (USA), 1994.
- [24] J. Kuffner and S.M. La Valle. RRT-Connect: An efficient Approach to Single-Query Path Planning. In *IEEE International Conference on Robotics and Automation*, San Francisco (USA), 2000.
- [25] F. Lamiroux and J.P. Laumond. Flatness and small-time controllability of multi-body mobile robots: application to motion planning. In *European Control Conference*, Brussels, 1997. Extended version to appear in *Transaction of Automatic Control*.
- [26] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [27] M. Lin, D. Manocha, J. Cohen and S. Gottschalk. Collision detection: Algorithms and applications. In *Algorithms for Robotic Motion and Manipulation (WAFR96)*, J.P. Laumond and M. Overmars (Eds), AK Peters 1997.
- [28] J.M. McCarthy. *Geometric Design of Linkages (preliminary version)* Springer Mathematics, 2000.
- [29] Ch. Nielsen, L. Kavraki. A two-level fuzzy PRM for manipulation planning. In *IEEE International Conference on Intelligent Robots and systems*, Japan, 2000.
- [30] M. Overmars and P. Svestka. A Probabilistic learning approach to motion planning. In *Algorithmic Foundations of Robotics (WAFR94)*, K. Goldberg et al (Eds), AK Peters, 1995.
- [31] J. A. Reeds and R. A. Shepp. Optimal paths for a car that goes both forward and backwards. *Pacific Journal of Mathematics*, 145 (2), pp. 367–393, 1990.
- [32] T. Siméon, J.P. Laumond and C. Nissoux. Visibility based probabilistic roadmaps for motion planning. In *Advanced Robotics*, vol 14 n 2, 2000. A short version appeared in *IEEE IROS 1999*, Korea.
- [33] T. Siméon, J.P. Laumond, F. Lamiroux. Move3D: a generic platform for path planning. *IEEE Int. Symposium on Assembly and Task Planning*, Japan, 2001.