Move3D: a generic platform for path planning

T. Siméon, J-P. Laumond, F. Lamiraux

LAAS-CNRS
7, avenue du Colonel-Roche
31077 Toulouse Cedex - France
{nic,jpl,lamiraux}@laas.fr

Abstract

This paper reports on Move3D, a software platform dedicated to collision-free path planning. The algorithms are based on probabilistic approaches and take advandtage of the progress of computer performance. The generality comes from a dedicated software architecture allowing a rapid design of path planners. The paper focuses on recent results obtained in logistics for industrial installations, in graphics animation and in mobile robotics as well.

1 Introduction

Path planning has been an active research field in robotics for more than 20 years [26]. Algorithms become more and more mature. Moreover, they take advantage of the increasing progresses of computer speed: 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.

This paper presents the software plateform Move3D currently developed at LAAS for generic multipurpose applications. From an algorithmic point of view, Move3D is based on probabilistic roadmap algorithms briefly reviewed in Section 2. In Section 3, we present the software architecture of Move3D implementing the various ingredients required by roadmap algorithms: collision-checkers, steering methods, roadmaps builders. The following sections give practical results obtained in problems arising in fields as diverse as logistics for industrial installations, graphics animation and mobile robotics.

2 Probabilistic approaches to path planning

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 detection method that detects both interferences between the robot and the obstacles for a given configuration and collisions of an admissible path with the obstacles.

The main components of a roadmap based 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 one degree of freedom at once). The inputs of a steering method are two configurations of the mechanical system, the output is an admissible path linking them without taking into account the obstacles.

- **basic geometric tools**: they are of two types:

  - *interference detection*: checks that in a given configuration, the robot and the obstacles have empty intersection,

  - *collision checking*: checks that along a given path, no configuration is in collision with the obstacles.

- **the roadmap builder**: this is the algorithm used to compute a roadmap. Two collision-free configurations being randomly picked, the roadmap builder checks whether the admissible

path computed between these configurations using the selected steering method is collision-free. If so, an edge is added between these configurations.

Once a roadmap has been computed, a specific path planning problem between a start and goal configurations can be solved as follows. The planner tries to connect the initial and goal configurations to the roadmap, using the specified steering method. Then, a graph search procedure looks for a path in the roadmap. If the graph search succeeds, a solution path has been found. This first solution path is then optimized to remove useless detours induced by the random search.

In the following section we present the architecture of our roadmap based planning software Move3D and we give details about the different components presented above.

3 Move3D architecture

Figure 1 shows the global architecture of the motion planning software Move3D, composed of the following modules:

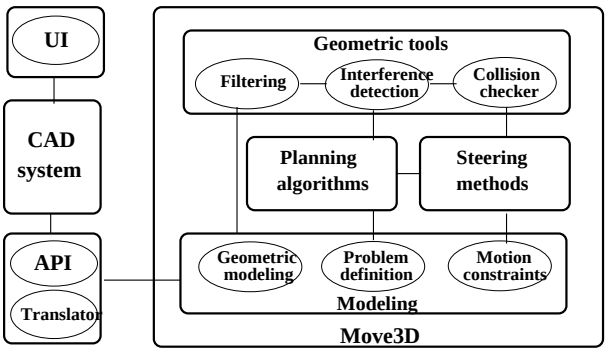


Figure 1: Architecture of Move3d

- *the Modeling module* enables the user to describe mechanical systems and environments. The motion planning problem to solve is also described in this module. 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 Move3D geometric data structures
- *the geometric tools* for filtering the geometric database and initializing the interference detection algorithms only with the part relevant to the specified planning problem.
- *the steering method* library allowing to compute local paths satisfying the kinematic constraints of the mechanical systems involved by the planning problem.
- *the planning algorithms* module contains the roadmap builder, the graph search algorithms as well as other planning procedures based on randomized techniques (see next section).
- finally, Move3D has its own *Developer Interface module* (not displayed in the Figure) allowing to call the algorithms and to visualize the solutions independently of any external CAD system.

3.1 Steering methods

The following steering methods illustrated by Figure 2 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 (trailer systems) [25] as well.
- `Manhattan` accounts for the constraint of moving one degree of freedom at once.

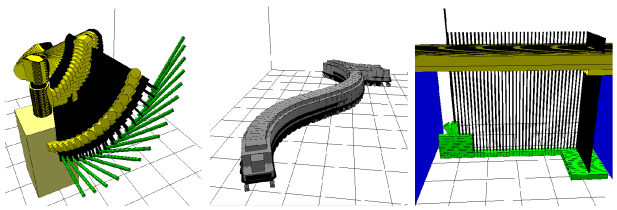


Figure 2: Examples of elementary steering methods

Other methods can 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 (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. For instance the front wheels of a car have to turn together. In the 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 Planning algorithms

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 on the basic PRM scheme [20]. The configuration space is uniformly sampled at random. All collision-free samples are added to the roadmap and checked for connection 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: guards and connectors. A collision-free sample is kept as a new guard node if it cannot be connected to the current roadmap, or as a new connector if it improves the connectivity of the roadmap (i.e. it can be connected to two different connected components). An advantage 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 gaussian 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 on 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 tools: data filtering and collision-checking

A motion planning task is often localized on a small part of the environment. The role of `Filtering` is to pre-process the geometric data basis to determine which bodies may interact between each other when searching a collision-free path and thus to reduce the number of tests within collision checking and interference detection. `Filtering` 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 a problem of path planning for a mobile robot as illustrated in Figure 6. In this case, the filter automatically removes all the parts of the environment that cannot interact with the robot (outside the slice defined by two horizontal planes). For the example shown in Figure 3, the filter retains most of the obstacles 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 operations 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 on 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 on 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 efficiency 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 algorithm, using a dichotomic sampling of the path and a non uniform steps 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 following examples come from the European project MOLOG dedicated to the integration of path planning technology to the CAD software used in logistics of industrial installation (see `http://www.laas.fr/~molog` and [33]).

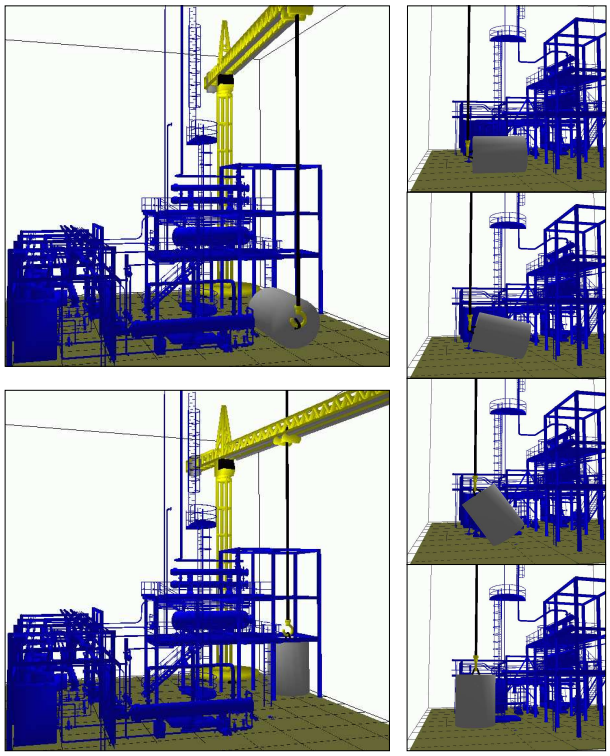


Figure 3: Stabilizer environment with a rotating crane (model provided by Cadcentre)

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 on the ground. Here the steering method produces `Manhattan` paths for the crane, while accounting for the constrained motion of the tank which slides on the ground before it reaches a vertical position.

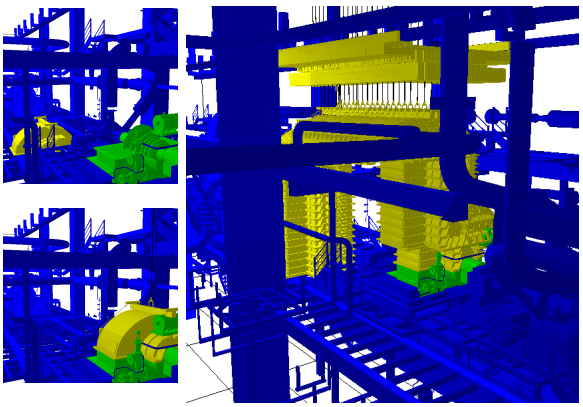


Figure 4: Travelling crane in the engine room of a nuclear plant (model provided by EDF)

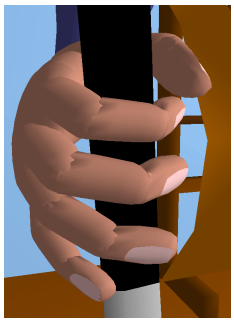
The second example shows a partial view of the engine room of a nuclear plant. Here, the purpose of the maintenance operation is to repair the water turbopump composed of three elements which have to be moved to a lower floor with a traveling crane. The path displayed in Figure 4 corresponds to the motion of the traveling crane computed to grasp the first element of the pump. For this scene (same geometric complexity as 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

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 5 the goal configuration is not explicitely specified by the operator (think to the huge number of parameters to tune to fully specify the poisition 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: Mobile robotics

Move3D is also used in the robotics projects conducted at LAAS. Figure 6 shows the mobile robot Hilare pulling a trailer. This system is nonholonomic: it has two degree of freedom (i.e., the two driving wheels) while the dimension of the configuration space is four (3 parameters to locate Hilare and the angle of the trailer). A special steering method has been devised for this system [25]. Figure 6 shows a path computed



Initial configuration and detail of the goal

Intermediate configurations computed by Move3D

Figure 5: Grasping a sword (environment provided by Dramaera)

Hilare with its trailer

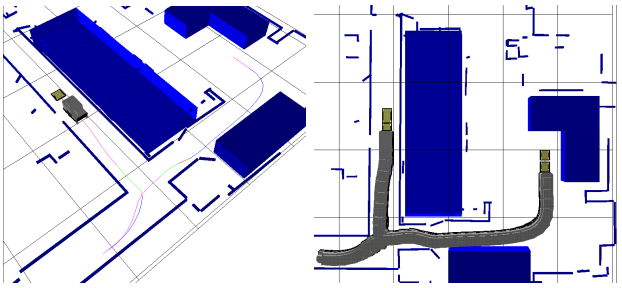


Figure 6: A path computed by Move3D together with the swept volume. The geometric model of the environment has been automatically computed by Hilare from a laser range-finder.

within Move3D in an environment modeling the experimental room of LAAS.

7 Conclusion

The aim of the project 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 in path planning.

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 exemple of Figure 5.

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 sheme for path planning. In *Robotics Research*, G. Giralt and G. Hirzinger (Eds), Springer, 1996.

[6] J. Barraquand and JC. 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. *In preparation.*

[14] S. Gootschalk, 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 Dicrete and Computational Geometry*, J.E. Goodman and J. O'Rourke (Eds), CRC Press, 1997.

[16] D. Hsu, JC. 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, JC. 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 entreprise. In *Algorithms for Robotic Motion and Manipulation (WAFR96)*, J.P. Laumond and M. Overmars (Eds), AK Peters, 1997.

[19] L. Kavraki and JC 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 JC. 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. Lamiraux and JP. 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)*, JP. 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, C. VanGeem and J. Cortes. Computer aided motion: Move3D within Molog. submitted to IEEE Int. Conf. on Robotics and Automation, September 2000.