16

Incremental Motion Planning With Las Vegas Algorithms

Jouandeau Nicolas, Touati Youcef and Ali Cherif Arab
University Paris8
France

1. Introduction

Las Vegas algorithm is a powerful paradigm for a class of decision problems that has at least a theoretical exponential resolving time. Motion planning problems are one of those and are out to be solved only by high computational systems due to such a complexity (Schwartz & Sharir, 1983). As Las Vegas algorithms have a randomized way to meet problem solutions (Latombe 1991), the complexity is reduced to polynomial runtime. In this chapter, we present a new single shot random algorithm for motion planning problems. This algorithm named RSRT for Rapidly-exploring Sorted Random Tree is based on inherent relation analysis between Rapidly-exploring Random Tree components, named RRT components (LaValle, 2004). RRT is an improvement of previous probabilistic motion planning algorithms to address problems that involve wide configuration spaces. As the main goal of the discipline is to develop practical and efficient solvers that automatically produce motion, RRT methods successfully reduce the complexity in exploring the space partially and producing non-deterministic solutions close to optimal ones. In the classical RRT algorithm, space is explored by repeating successively three phases: generation of a random configuration in the whole space (including free and non-free space); selection of a nearest configuration; and generation of a new configuration obtained by numerical integration over a fixed time step. Then the motion planning process is discretized into steps from the initial configuration to other configurations in the space. In such a way, RRT algorithms are the motion planners last generation that generally addresses a large set of motion planning problems. Mobile, geometrical or functional constraints, input methods and collision detection are unspecified. As it is possible to measure solutions provided by RRT, RSRT or other improvements in spaces with arbitrary dimension, experiments are realized on a wide set of path planning problems involving various mobiles in static and dynamic environments. We experiment the RSRT and other RRT algorithms using various configurations spaces to produce a massive experiment analysis: from free flying to constraint mobiles, from single to articulated mobiles, from wide to narrow spaces, from simple to complex distance metric evaluations, from special to randomly generated spaces. These experiments show practical performances of each improvement, and results reflect their classical behavior on each type of motion planning problems.

2. RRT Sampling Based-planning

2.1 Principle

In its original formulation (LaValle, 1998), RRT method is described as a tree G = (V,E) , where V is the set of vertices and E the set of edges in the research space. From an initial configuration q_{init}, the objective is to generate a sequence of commands, leading a mobile M, to explore all the configurations space C. The RRT method can solve this problem by searching solution which spans a tree, where the configuration q_{init}, describes the root node. One can note that nodes and arcs represent respectively eligible configurations of M and commands which are applied to move between the configurations. RRT method is a random incremental search of configurations which permits a uniform exploration of the space. The RRT implementation consists on a three phases: generate a configuration q_{rand}, select a configuration q_{prox} inside the current tree, and integrate a new configuration q_{new} from q_{prox} towards q_{rand}.

During the first phase, a random function is implemented to select an element of a configurations space. The second phase consists of choosing q_{prox} of G, which is the nearest element of q_{rand}. This phase is based on a metric ρ. Finally, a new configuration q_{new} from q_{prox} towards q_{rand} is generated and the objective is to implement a control which leads to bring q_{prox} closer to q_{rand}. The new configuration q_{new} is generated by integrating from q_{prox}, during a predefined time interval.

2.2 Graph construction of RRT method

Firstly, the RRT method is developed to solve planning problem in mobile robotic. In the original algorithm, the possible constraints associated to M are not mentioned. During the formulation of G, changes to be made for adding new constraints are minors, and the precision depends mainly on the chosen local planning method. The graph elementary construction in RRT method is described according to algorithm ALG. 1.

```
consRrt (q_init , k , Δt , C )
    init (q_init , G )                                          (1)
    for i in 1 to k                                             (2)
        q_rand = randConfig ( C )
        q_prox = nearestConfig (q_rand , G )                    (3)
        q_new = newConfig (q_prox , q_rand , Δt )               (4)
        addConfig (q_new , G )
        addEdge (q_prox , q_new , G )
    return G

nearestConfig (q_rand , G )
    d = inf
    foreach q in G
        if ρ ( q , q_rand ) < d                                 (5)
            q_prox = q
            d = ρ ( q , q_rand )
    return q_prox
```

ALG. 1. Original RRT algorithm formulation

We remark that the algorithm implements three functions. The first one, randConfig, ensures a uniform partition of random samples in C, and guaranties uniform exploration (Yershova & LaValle, 2004 and Lindemann et al., 2004). Function nearestConfig selects the nearest configuration q_{rand} of G. This relation proximity is defined by a distance metric ρ, as it is illustrated in (ALG. 1. (5)). In the case of probabilistic methods PRM and RRT, the nearest neighbour search with arbitrary dimension can be optimised (Yershova & LaValle, 2007 and Yershova & LaValle, 2002). Reducing of the search time of a nearest neighbour permits to use a complex distance metric. A new configuration q_{new} can be defined by newConfig from q_{prox} towards q_{rand}. Knowing that M is subject to holonomic constraints, a control inputs can be applied to move from q_{prox} towards q_{rand} with displacements amplitudes Δt. Functions addConfig and addEdge add respectively q_{new} to the list of nodes of G and arcs between q_{prox} and q_{new} .

2.3 Cardinality and layer

For each new configuration q_{new} in the generation phase, RRT method adds a configuration by propagating q_{prox} of G. In this case, no restriction on q_{new} is imposed according to configurations set G. So, q_{new} can be similar to q_{exist}, which can make possible to span a graph with or without cycle. For example, let's define Card as a cardinal of set, thus, if Card (V) = Card (E) + 1, then we can conclude that the graph is non-cyclic. To avoid stacking of identical movements, each nodes q_{prox} can't be extended towards q_{rand} for creating q_{new}, if it doesn't already have a similar descendent.

If q_{prox} is extended towards q_{rand}, a new arc between q_{prox} and q_{new} is inserted in E.

If Card (V) ≤ Card (E), we can conclude that the graph contains at least one cycle. Thus, is q_{new} deleted and a new arc is inserted in E between q_{prox} and q_{exist}.

Creating cycles leads to decrease an expansion number of G in unexplored zones. However, it permits to list possible solutions in the case of halt. Knowing that this scenario is more topologic than geometric, RRT method is better without cycle [LAV98]. Fig 1 shows the expansion of G respectively after 100, 500 and 1500 samples. Random samples have been uniformly spread in the square. q_{init} is initially in the center of the square. The mobile is a simple point (without geometric shape) with holonomic constraints.

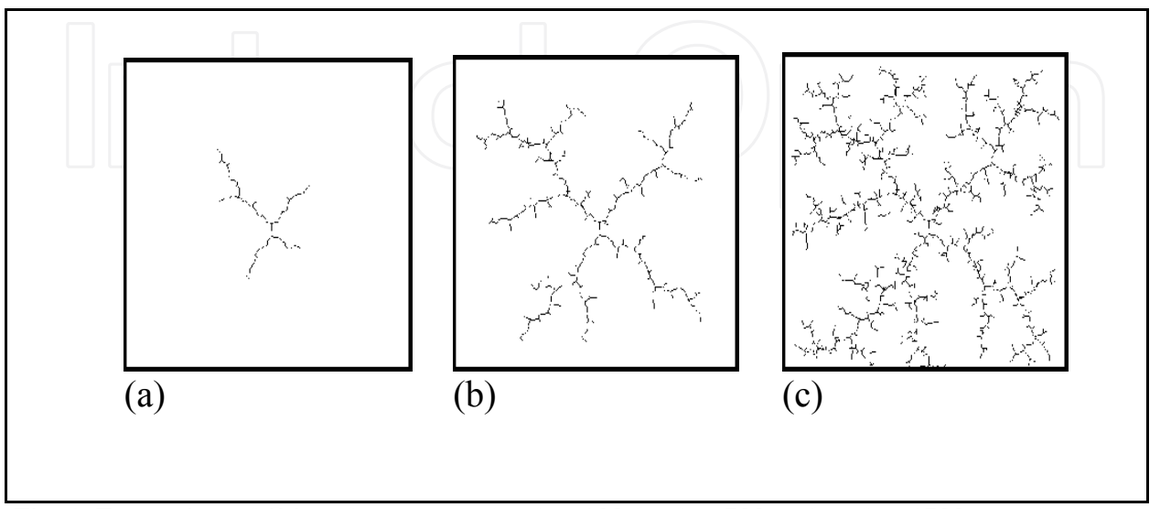


Fig. 1. Expansion of G in a free square (a after 100, b after 500 and c after 1500 samples)

2.4 Natural expansion

The random distributions of samples which performs expansions, directs naturally the growth of G towards the wider regions of space. This can be verified by constructing Voronoï diagram which associates, for each new node of C, one Voronoï cell. For each iteration of RRT method, the localization probability of the next random sample is more important towards the largest cells of Voronoï diagram, which is defined by a previous random samples set.

Let's C_k be a distribution of k random samples in the configurations space C. the distribution C_k converges in term of probability to C under condition of the uniformity of a random samples partition in C (LaValle & Kuffner, 2000).

Knowing that Delaunay triangulation is a dual of Voronoï diagram, an example of a graph expansion associated to RRT method is presented in Fig. 2. Graphs presented in (a), (b) and (c) illustrate respectively the results of 25, 275 and 775 expansions including those of Delaunay triangulations illustrated in (a'), (b') and (c'). The space is two dimensional squares without obstacles. For each iteration, adding a new item leads to construct a new triangulation. In Fig. 2, initial configuration is represented by a circle in the center of the space.

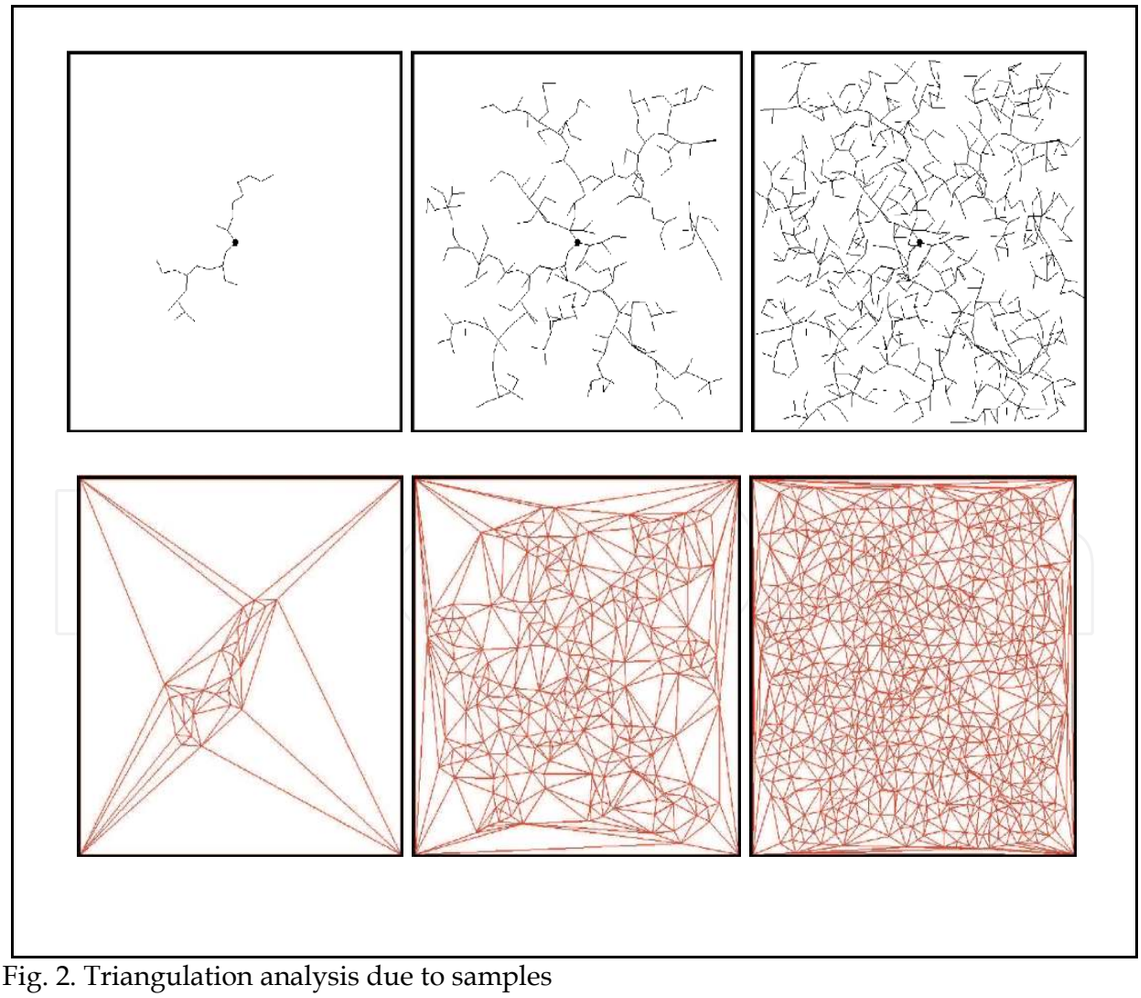


Fig. 2. Triangulation analysis due to samples

The evolution of new configurations of G along iterations is illustrated in Fig 3. X-axis represents the configuration number contained in the graph and Y-axis represents the percentage of the entire surface S. The surface graph represents the average, minimal, and maximal surface variations. In this case, the average surface is the average triangles surfaces. The standard deviation graph represents the average, minimal and maximal standard deviations. The initial configuration divides the space into four triangles with 0.25 in term of surface and zero in standard deviation. The average area of triangles decreases linearly according to the number of configurations.

In Figures 2, 3 and 4, positions in (a), (b) and (c) are placed around area average and standard deviations curves. Maximum and minimum variations can increase or decrease according to their relative positioning to the decreasing average value. Due to the logarithmic scale, position of minimal variations vis-à-vis average values shows the almost-equality between average value and minimum value. On the other hand, position of maximal variations shows triangles much larger than the average value before a density threshold (8.15 times larger before 353 configurations). From 353 configurations, the ratio between the higher triangle and the average value progresses in stair-steps. Two stair-steps p0 and p1 are placed on average and standard deviations curves as it's illustrated in Fig 3. This ratio tends to be stabilized around 2 from p1. The initial configuration position has no influence on statistics relative to its expansion.

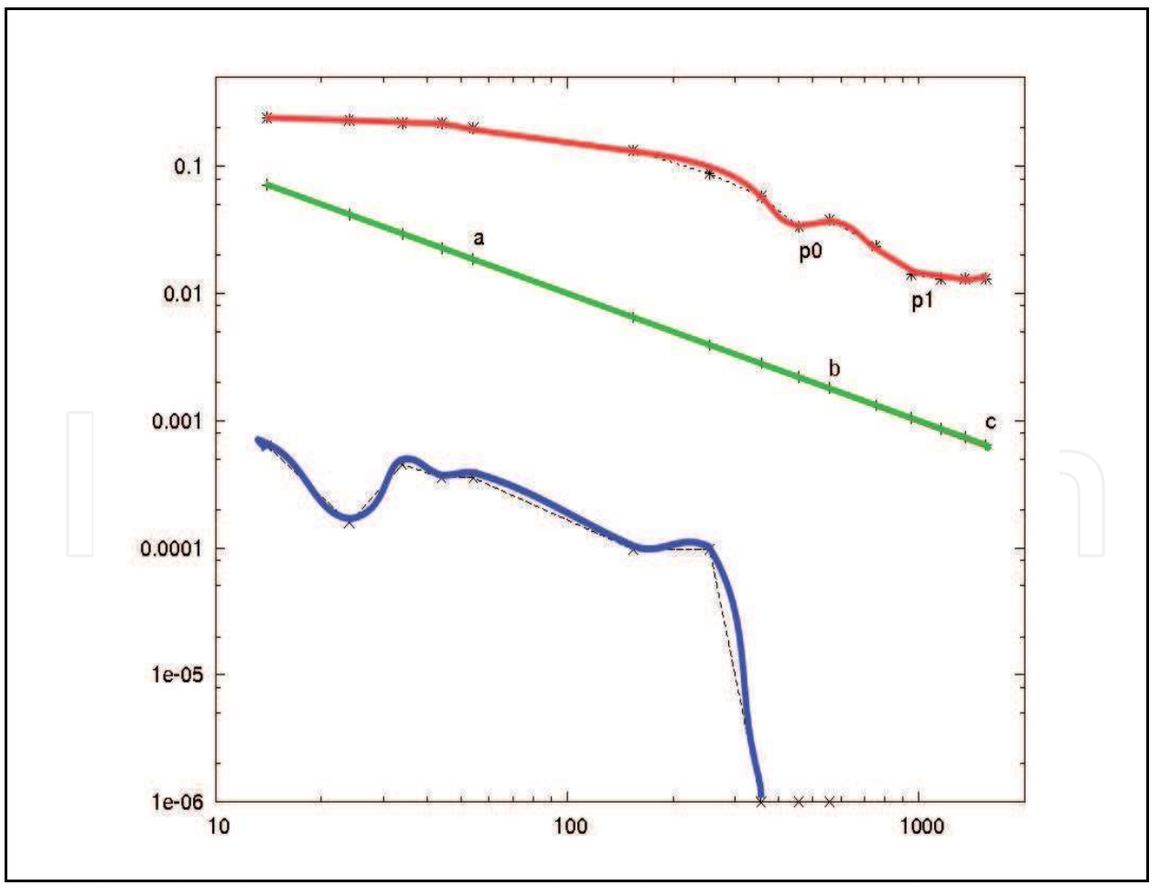


Fig. 3. Evolution of average, min and max of triangles areas during sampling

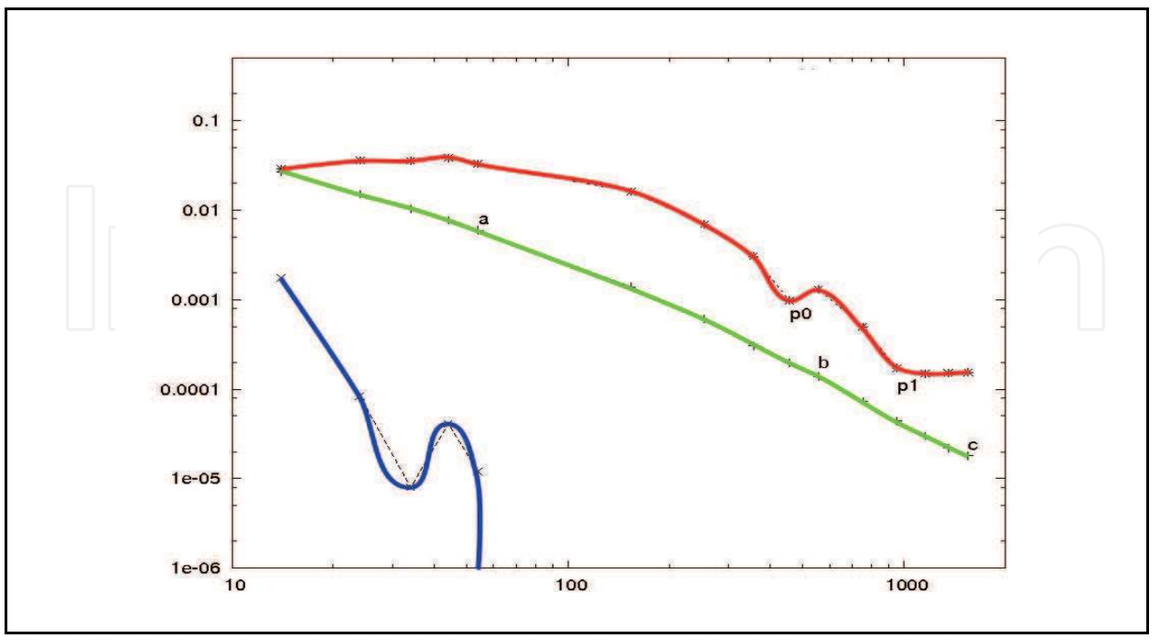


Fig. 4. Evolution of standard deviation, min and max of deviation areas during sampling

2.5 End condition

A query of a mobile trajectory planning can be formulated according to a pair of configuration-objective, which is instantiated on q_{obj} or on a set of configurations C_{obj}. Restricting the search to a single configuration-objective can penalize the mobiles which are subjected to dynamics or non-holonomics constraints. To improve the convergence towards the objective, RRT resolutions implement a configuration q_{obj}, whose components are not fixed. Thus, the planning problem consists to find a path connecting q_{init} to an element of C_{obj}. From q_{init}, graph G seeks to achieve a configuration q_{obj}. This can be done by a successive adding of new configuration q_{new} in the tree G. Variable k defines the number of iterations required to solve the problem. In the case of k is not sufficient it is possible to continue conducting research on new k iterations from the previously generated tree. The construction of G is achieved when $q_{obj} \cap G=\varnothing$.

3. Related Works

In the previous section, C is presented without obstacle in an arbitrary space dimension. At each iteration, a local planner is used to connect each couples (q_{new} , q_{obj}) in C. The distance between two configurations in T is defined by the time-step Δt. The local planner is composed by temporal and geometrical integration constraints. The resulting solution accuracy is mainly due to the chosen local planner. k defines the maximum depth of the search. If no solution is found after k iterations, the search can be restarted with the previous T without re-executing the init function. This principle can be enhanced with a bidirectional search, shortened Bi-RRT (LaValle & Kuffner, 1999). Its principle is based on the simultaneous construction of two trees (called T_{init} and T_{obj} that grows respectively from q_{init}

and q_{obj}. The two trees are developped towards each other while no connection is established between them. This bidirectional search is justified because the meeting configuration of the two trees is nearly the half-course of the initial configuration space. Therefore, the resulting resolution time complexity is reduced (Russell & Norvig, 2003).
RRT-Connect is a variation of Bi-RRT that consequently increase the Bi-RRT convergence towards a solution (Kuffner & LaValle, 2000) thanks to the enhancement of the two trees convergence. This has been settled :

- to ensure a fast resolution for "simple" problems (in a space without obstacle, the RRT growth should be faster (ALG.2. (1)) than in a space with many obstacles)
- to maintain the probabilistic convergence property. Using heuristics modify the probability convergence towards the goal and also should modify its evolving distribution. Modifying the random sampling can create local minima that could slow down the algorithm convergence

```
connectRrt (q , Δt , T )
    r = ADVANCED
    while r equals ADVANCED                                    (1)
        r = expandT ( q , Δt , T )
    return r
```

ALG. 2. Connecting a configuration q to T with RRT-Connect.

As it makes RRT less incremental, RRT-Connect is more adapted for non-differential constraints (Cheng, 2001). It iteratively realize expansion by replacing a single iteration (ALG. 1. (2)) with connectT function which corresponds to a succession of successful single iterations (ALG. 2. (1)). An expansion towards a configuration q becomes either an extension or a connection.

```
connectBiRrt (q_init , q_obj , k, Δt , C )
    init ( q_init, T_a )
    init ( q_obj, T_b )
    for i in 1 to k
        qrand = randConfig ( C )
        r = expandRrt (q_rand , Δt , T_a )
        if r not equals TRAPPED
            if r equals REACHED
                q_co = q_rand
            else
                q_co = q_new
            if connectRrt (q_co , T_a , T_b )
                Return solution
        swap (T_a , T_b )
    return TRAPPED
```

ALG. 3. Expanding two graphs with RRTConnect

According that two trees are constructed by Bi-RRT, growth is realized inside two trees named T_a and T_b and a successfull connection of q_{new} towards q_{rand} in T_a, implies many other extensions (as many as the free space admits new free configurations, i.e. q_{new} in C_{free}) of q_{prox} found in T_b towards q_{new}. This new configuration q_{new} becomes a convergence configuration named q_{co} (ALG. 3).
To improve the construction of T to an adequate progression of G in Cfree, previous works propose :

- to deviate from its initial distribution the random sampling Bi-RRT and RRT-Connect. Other Variations of RRT-Connect are called RRT-ExtCon, RRT-ConCon and RRT-ExtExt; they modify the construction strategy of one of the two trees. The priorities of extension and connection are balanced with new values according to previous extensions (LaValle, 1998)
- to adapt q_{prox} selection to a collision probability (Cheng & LaValle, 2001)
- to restrict q_{prox} selection in an accessibility vicinity of the previous q_{prox} in the variation called RC-RRT (Cheng & LaValle, 2002)
- to bias sampling towards free spaces (Lindemann & LaValle, 2004)
- to parallelize growing operations for n distinct graphs in the variation OR parallel Bi-RRT and to share G with a parallel q_{new} sampling in the variation embarrassingly parallel Bi-RRT (Carpin & Pagello, 2002)
- to focus the sampling of special parts of C to control the RRT growth (Cortès & Siméon, 2004 and Lindemann & LaValle, 2003 and Yershova et al. 2005)

By adding the collision detection in the configuration space, the selection of nearest neighbor q_{prox} is garanted by a collision detector. The collision detection is expensive in computing time, the distance metric evaluation ρ is subordinate to the collision detector.

```
expandRrt(q , Δt , T )
    q_prox = closestConfig ( q, T )
    dmin = rho (q_prox , q )                                    (1)
    success = FALSE
    foreach u in U
        q_tmp = integrate ( q , u , Δt )
        if isCollisionFree (q_tmp , q_prox , M , C)
            d = ro (q_tmp , q_rand )
            if d < dmin
                q_new = q_tmp
                success = TRUE
    if success equals TRUE
        insert (q_prox , q_new , T )
        if q_new equals q
            return REACHED
        return ADVANCED
    return TRAPPED
```

ALG. 4 Expanding according to a collision detector

As U defines the set of admissible orders available to the mobile M, the size of U mainly defines the computation times needed to generate, validate and select the closest configuration with as the best expansion configuration. For each expansion, the function expandRrt (ALG. 3.) returns three possible values: REACHED if the configuration q_{new} is connected to T, ADVANCED if q is only an extension of q_{new} which is not connected to T, and TRAPPED if q cannot accept any successor configuration q_{new}.
The construction of T corresponds to the repetition of such a sequence. The collision detection discriminates the two possible results of each sequence :

- the insertion of q_{new} in T (i.e. without obstacle along the path between q_{prox} and q_{new})
- the rejection of each q_{prox} successors (i.e. due to the presence of at least one obstacle along each successors path rooted at q_{prox})

The rejection of q_{new} induces an expansion probability related to its vicinity (and then also to q_{prox} vicinity); the more the configuration q_{prox} is close to obstacles, the more its expansion probability is weak. It reminds one of fundamentals RRT paradigm: free spaces are made of configurations that admit various number of available successors; good configurations admit many successors and bad configurations admit only few ones. Therefore, the more good configurations are inserted in T, the better the RRT expansion will be. The problem is that we do not previously know which good and bad configurations are needed during the RRT construction, because the solution of the considered problem is not yet known. This problem is also underlined by the parallel variation (Carpin & Pagello, 2002) called OR Bi-RRT (i.e. to define the depth of a search in a specific vicinity). For a path planning problem p with a solution s available after n integrations starting from q_{init}, the question is to maximize the probability of finding a solution; According to the concept of ``rational action'', the response of P3 class to adapt a on-line search can be solved by the definition of a formula that defines the cost of the search in terms of ``local effects'' and ``propagations'' (Russell, 2002). These problems find a way in the tuning of the behaviour algorithm like CVP did (Cheng, 2001).

3.2 Tunning the RRT algorithm according to relations between components

In the case of a space made of a single narrow passage, the use of bad configurations (which successors generally collide) is necessary to resolve such problem. The weak probability of such configurations extension is one of the weakness of the RRT method (Jaillet L. et al. 2005).
To bypass this weakness, we propose to reduce research from the closest element (ALG. 4) to the first element of C_{free}. This is realized by reversing the relation between collision detection and distance metric; the solution of each iteration is validated by subordinating collision tests to the distance metric; the first success call to the collision detector validates a solution. This inversion induces :

- a reduction of the number of calls to the collision detector proportionally to the nature and the dimension of U. Its goal is to connect the collision detector and the derivative function that produce each q_{prox} successor
- an equiprobability expansion of each node independently of their relationship with obstacles

The T construction (we called RSRT) is now based on the following sequence:

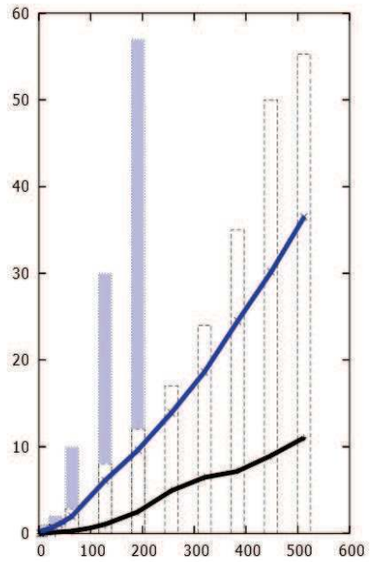
- the generation of a random configuration q_{rand} in C
- the selection of q_{prox} the nearest configuration to q_{rand} in T
- the generation of each successors of q_{prox}. Each successor is associated with its distance metric from q_{rand}. It produces a couple called s stored in S
- the sort of s elements by distance
- the selection of the first collision-free element of S and breaking the loop as soon as this first element is discovered

4. Results

Fig. 6. and 7. present two types of environment that have been chosen to test algorithms. In these environments, obstacles are placed. For each type, we have generated series of environments that gradually contains more obstacles. This is one element of these series that we call a problem. For each problem, we generate 10 different instances, to realise statistics on solutions provide (Fig. 5). The number of obstacles is defined by the sequence 2, 4, 8 … 512 and also until the resulting computing time is less than 60 sec. We have fixed this limit to see what could be possible in an embedded system. The two types of environment correspond to a simple mobile robot and a small arm with 6-DOF. We used the Proximity Query Package (PQP) library to test collisions and the Open Inventor library to visualize solutions. For each mobile in each environment, we have applied a uniform inputs set dispatched over translation and rotation.
Considering generic systems, we have apply different mover's model:

- that consider the trajectory as a list of position
- that consider the trajectory as a list of position with a velocity for each DOF

Each set of instances are associated with different distances metrics (Euclidian, scaled Euclidian and Manhattan distances).



Black and blue curves show respectively results for moving free-flyer and 6-DOF arm with RSRT. Boxes show respectively results for classical RRT. Until 296 obstacles, classical RRT is not able to provide solution for 6-DOF arm. Resolving time of tuned RRT (we called RSRT) is 4 times faster for hard problems and faster for easier problems. Classical RRT seems to be more dependent on the input set dimension. Tuned RRT computing time seems also to be independent of the distance metric used. However Manhattan metric is the most efficient for 6-DOF arm in any case.

Fig. 5. Computing resolving times while gradually increasing environment complexity

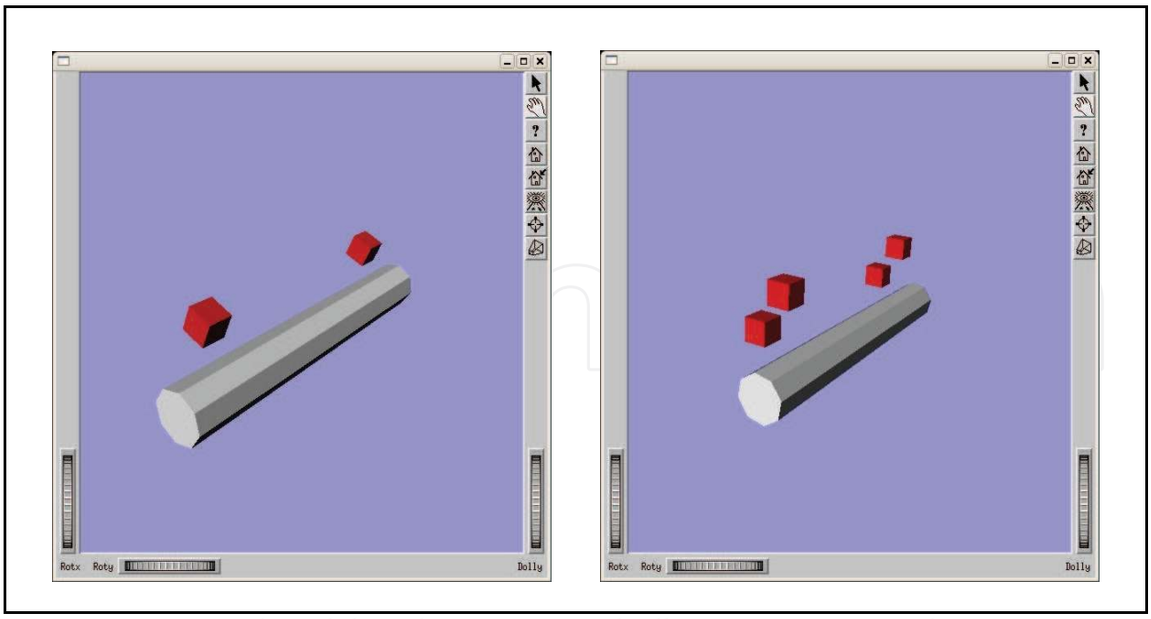


Fig. 6. Moving simple mobile and increasing gradually environment complexity

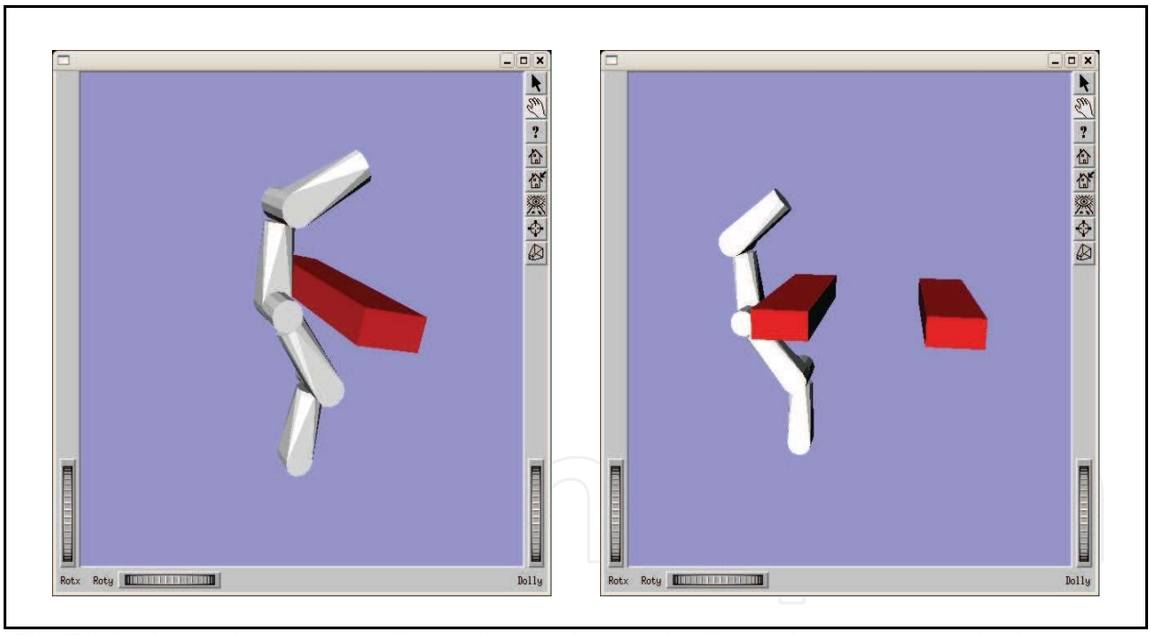


Fig. 7. Moving articulated mobile and increasing gradually environment complexity

7. Conclusion

We have described a way of tuning RRT algorithm, to solve more efficiently hard problems. RSRT algorithm accelerates consequently the required computing time. The result have been tested on a wide set of problems that have an appropriate size to be embedded. This approach allows RRT to deal with motion planning strategies based on statistical analysis.

8. References

Carpin, S. & Pagello, E. (2002). On Parallel RRTs for Multi-robot Systems, *8th Conf. of the Italian Association for Artificial Intelligence* (AI*IA)

Cheng, P. & LaValle, S.(2002). Resolution Complete Rapidly-Exploring Random Trees, *Int. Conf. on Robotics and Automation* (ICRA)

Cheng, P. (2001) Reducing rrt metric sensitivity for motion planning with differential constraints, *Master's thesis*, Iowa State University

Cheng, P. & LaValle, S. (2001). Reducing Metric Sensitivity in Randomized Trajectory Design, *Int. Conf. on Intelligent Robots and Systems* (IROS)

Cortès, J. & Siméon, T. (2004). Sampling-based motion planning under kinematic loop-closure constraints, *Workshop on the Algorithmic Foundations of Robotics* (WAFR)

Jaillet L. et al. (2005). Adaptive Tuning of the Sampling Domain for Dynamic-Domain RRTs, *IEEE International Conference on Intelligent Robots and Systems* (IROS)

Kuffner, J. & LaValle, S. (2000). RRT-Connect: An efficient approach to single-query path planning, *Int. Conf. on Robotics and Automation* (ICRA)

Latombe, J. (1991). Robot Motion Planning (4th edition), *Kluwer Academic*

LaValle, S. (2004). Planning Algorithms, *[on-line book]* http://msl.cs.uiuc.edu/planning/

LaValle, S. & Kuffner, J. (2000). Rapidly-exploring random trees: Progress and prospects, *Workshop on the Algorithmic Foundations of* Robotics (WAFR)

LaValle, S. & Kuffner, J. (1999). Randomized kinodynamic planning, *Int. Conf. on Robotics and Automation* (ICRA)

LaValle, S. (1998). Rapidly-exploring random trees: A new tool for path planning, *Technical Report* 98-11, Dept. of Computer Science, Iowa State University

Lindemann, S. & LaValle, S. (2004). Incrementally reducing dispersion by increasing Voronoi bias in RRTs, *Int. Conf. on Robotics and Automation* (ICRA)

Lindemann, S. et al. (2004). Incremental Grid Sampling Strategies in Robotics, *Int.Workshop on the Algorithmic Foundations of Robotics* (WAFR)

Lindemann, S.R. & LaValle, S.M. (2003). Current issues in sampling-based motion planning, *Int. Symp. on Robotics Research* (ISRR)

Lozano-Pérez, T. (1983). Spatial Planning: A Configuration Space Approach, *Trans. on Computers*

Russell, S. & Norvig, P. (2003). Artificial Intelligence, A Modern Approach (2nd edition), *Prentice Hall*

Russell, S. (2002). Rationality and Intelligence, *Press O.U.*, ed.: Common sense, reasoning, and rationality

Schwartz, J. & Sharir, M. (1983). On the piano movers problem:I, II, III, IV, V, *Technical report*, New York University, Courant Institute, Department of Computer Sciences

Yershova, A. & LaValle, S. (2007). Improving Motion Planning Algorithms by Efficient Nearest Neighbor Searching, *IEEE Transactions on Robotic*s 23(1):151-157

Yershova, A. et al. (2005). Dynamic-domain rrts: Efficient exploration by controlling the sampling domain, *Int. Conf. on Robotics and Automation* (ICRA)

Yershova, A. & LaValle, S. (2004). Deterministic sampling methods for spheres and SO(3), *Int. Conf. on Robotics and Automation* (ICRA)

Yershova, A. & LaValle, S. (2002). Efficient Nearest Neighbor Searching for Motion Planning, *Int. Conf. on Robotics and Automation* (ICRA)

New Developments in Robotics Automation and Control

Edited by Aleksandar Lazinica

ISBN 978-953-7619-20-6
Hard cover, 450 pages
Publisher InTech
Published online 01, October, 2008
Published in print edition October, 2008

This book represents the contributions of the top researchers in the field of robotics, automation and control and will serve as a valuable tool for professionals in these interdisciplinary fields. It consists of 25 chapter that introduce both basic research and advanced developments covering the topics such as kinematics, dynamic analysis, accuracy, optimization design, modelling , simulation and control. Without a doubt, the book covers a great deal of recent research, and as such it works as a valuable source for researchers interested in the involved subjects.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jouandeau Nicolas, Touati Youcef and Ali Cherif Arab (2008). Incremental Motion Planning With Las Vegas Algorithms, New Developments in Robotics Automation and Control, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-20-6, InTech, Available from: http://www.intechopen.com/books/new_developments_in_robotics_automation_and_control/incremental_motion_planning_with_las_vegas_algorithms

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen