Journal of Scientific & Industrial Research
Vol. 79, June 2020, pp. 513-516

Optimal Path Planning using RRT for Dynamic Obstacles

N Kalpitha[1,3]* and S Murali[2,3]

[1]Department of Information Science and Engineering, Jyothy Institute of Technology, Bangalore, Karnataka, India

[2]Department of Computer Science and Engineering, Maharaja Institute of Technology, Mysore, Karnataka, India

[3]Visvesvaraya Technological University, Belagavi, Karnataka, India

Received 13 March 2020; revised 22 April 2020; 12 May 2020

Rapidly Exploring Random Tree is a technique that utilizes samples as constraints for arranging the paths from given source to reach multiple goals. A sample is randomly selected from the configuration space. Each data point is represented as a step that helps in travelling of a robot from one point to another. By initiating the search process, the pursuit calculation endeavors to achieve the objective of the proposed work, further the position assisted regions are investigated. However, when compared with A* calculation, the proposed method has a more reasonable good performance. The Simulation results demonstrate for reaching the desired goal starting from a source to destination in dynamic environment.

Keywords: RRT, A*, RRG

Introduction

Mobile robot is a programmed machine equipped for various applications. Mobiles have capability to move around and without settling in one physical location. It is generally utilized in different fields, example, military, industrial, agricultural, and numerous different applications. A robot possesses autonomy and is equipped for exploring an uncontrolled domain without the need of physical or electro-mechanical direction gadgets. Recently, versatile administration robots used in indoor regions like houses, workplaces, hospitals are presented broadly. It is important to design a collision free trajectory limiting expense in terms of time, energy and distance.

The most critical way is to discover an ideal path without hurdles that can contribute for motion planning. Path planning algorithm with collision avoidance technique[1] is receiving considerable attention over the past twenty years and many algorithms play a role in development. Some of the techniques involves in detecting the obstacles, performing quantitative measurement with dimensions of the obstacles. Basically, these algorithms are comprised of two categories namely offline trajectory of path[2] requires complete information of the obstacles and the environment. Hence, this enables the system to generate complete traversing direction from initial to goal point even before the robot starts its motion and online motion planning[2] which works when the environment and obstacles information are completely unknown. Dave Ferguson et al. have proposed Delayed D*[3], a re-planning technique. It is a grid-based approach that holds exactly the inconsistent states. Shortest path is determined by computing smallest values obtained. Joshi & Shinde[4] have proposed the hybrid form of methodology for computing desirable solutions. Obstacle detection system detects obstacles by combining variance values and threshold values. The portable robots must need to be designed with a capacity to explore in powerful conditions to execute a given undertaking task by avoiding from obstacles.[5]

*Author for Correspondence
E-mail: kalpitha.n@jyothyit.ac.in

Methodology and Results

Path planning in dynamic environment for mobile robots is proposed as shown in Fig. 1. Designing a way for carrying out movement that allows in reaching from source to their corresponding target is presented. In the proposed system, multiple goals are considered. When a robot is set to move from source to destination it may have to deal with dynamic obstacles as given in Table 1.

The major issue in path planning is to overcome dynamic obstacles and reach the target goal in a given time. The dynamic obstacles problem can be solved using RRG algorithm as shown in Table 2. The robot has to efficiently plan a path towards the target goal. Dynamic planning method is introduced with a specific end goal to stay away from arbitraries when it is recognized. A fully autonomous robot requires a robust and efficient path planning and a re-planning algorithm to traverse in a dynamic environment. The path is biased towards goal which leads to less exploration of the map that generates a feasible path in less time.

Rapidly-exploring Random Tree (RRT)

The rule of probability comes up with new possible ways by seeking through areas quickly.

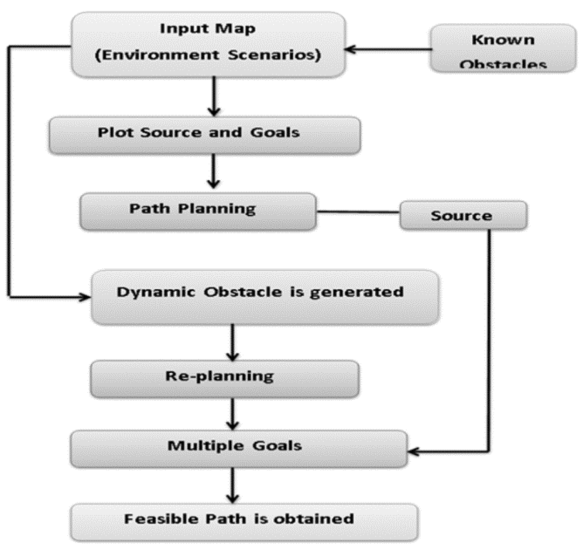


Fig. 1 — Flowchart of proposed RRT

This methodology works with high efficiency for issues with large dimensions, nonlinear, dynamics or differential constraints. It's additionally sensible in distinct or continuous areas. Pre-computing for the configuration space isn't needed during processing. Then it is fully grown by connecting random samples in free space. An edge is replaced in the tree if there is any chance of collision with obstacles.

Basic RRG

The system intends to maintain a graphical map instead of tree. One or more nodes in the graph are connected to each other. The ideal properties of the graph are:

1 Interconnection of graph consisting disconnected sub-graphs is possible in order to build a complete structure.
2 The least distance between two concurrent nodes denoted by r results with a greater cost computing system for slightly increasing quality. The separation element is termed as step size approximating value of r.
3 All nodes are represented with a location pointing to sequential hierarchy by expanding.
4 The association formed between nodes is said to be collision free if they are interconnected at a distance less than step size. A single node grouping together with many goals of similar color signifies connectivity redundantly. Hence,

Table 1 — Experimental Results before generation of dynamic obstacle

	Path Length 1	Processing Time(secs)	Path Length 2	Processing Time(secs)	Path Length 3	Processing Time(secs)
Map 1	759	4.7750	746	5.5954	729	5.0637
Map 2	401	0.9817	420	2.6707	457	1.6374
Map 3	327	0.8317	628	1.6713	570	2.0759
Map 4	115	0.2211	280	0.4811	746	2.4028
Map 5	66	0.1604	284	0.2717	519	0.5324
Map 6	651	0.1824	321	0.6612	737	4.2408
Map 7	169	0.2812	338	0.5366	230	0.3760
Map 8	103	0.3546	375	0.7767	825	1.8195

Table 2 — Experimental Results after generation of dynamic obstacle

	Path Length 1	Processing Time(secs)	Path Length 2	Processing Time(secs)	Path Length 3	Processing Time(secs)
Map 1	685	2.5979	766	5.1327	784	5.4029
Map 2	405	1.1825	429	1.8193	432	1.5052
Map 3	331	0.7367	586	3.2394	664	4.3957
Map 4	114	0.2188	259	0.4595	755	2.8565
Map 5	63	0.1056	318	0.3005	585	1.1546
Map 6	95	0.1802	305	0.5651	750	5.98
Map 7	188	0.3609	422	9.5571	256	0.8949
Map 8	96	0.2209	376	1.8879	610	2.0076

expanding a child from parent always resultantly connects each other. The colored sub-graphs could be differently represented via links.

5. Marking the area and constructing cover for nodes out of the configuration space already been explored is made easier. Newly covered plots may not be easily produced. Therefore, the algorithm should terminate.

Firstly, an arrangement of initiator nodes separated from one another could be the main parts of block diagram shown in Fig. 2. Coloring takes place for every plot uniquely. Searching activity arbitrarily begins a way from the start point might be an expensive investigation. Henceforth, non- region attractor might be indicated for each point. The exploration, beginning from indicated point, proceeds all together towards continuity in orderly moves. Following sequence from source and objectives to its respective positions is difficult decision. Distance of r from actual point or less than that can be repetitively taken as one. The global attractor considers its corresponding goals for source and vice versa.

Planning by Expansion

The graph repeats by continually including node parameters, in which the recently visited point must be associated with edges. The Iteration is done for every vertex of visited point. A color indicates a sub-graph which could conceivably be associated with each other relying on the nearness of bridging edges. Sample is produced for each c, nearest position to node is found. The vertex 'V' may stretched out towards 'S' to create an indistinguishable shading from V (color(N) = c). Then N is added if there is no other plot lies in RRG sufficiently close to N. Arbitrarily pick a random sample out of the whole configuration space C.

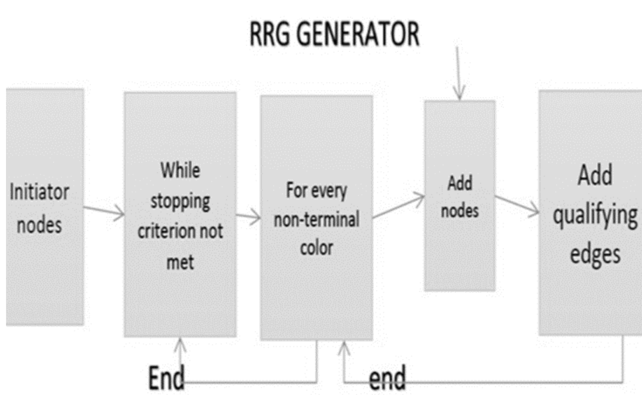


Fig. 2 — RRG Framework

Post Processing

The proposed technique continues assured descriptions for achieving particular edge estimations of development some specific color. The trajectory is computed by searching elements in graph. For the following scenarios, the best diagram seek system is A* calculation with Euclidean separation from acceptable target heuristic capacity. The final trajectory (t) is obtained by post processing method consisting small local optimization until spreads either converge with other shading variables or breaking points of the arrangements. Monitoring each and every color fails in adding nodes, if a particular threshold is crossed by terminating process due to the expansion of for producing path.

Proposed RRG method

The issue is to obtain an impact free direction for a mobile robot, ranging from a given source to multiple goals, wherever the journey completes in time T. The path is intended for a mechanism that it should not run into any obstacle at any time. The sample is drawn from the configuration area at random way which is denoted by C. In addition, the way might value more highly to maintain some minimal separation from obstacles that are obstructing the specified path from start node to goals. The algorithmic rule perpetually increments in order to utilize an anticipating path.

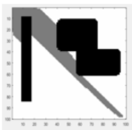
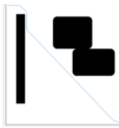
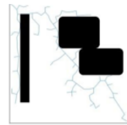
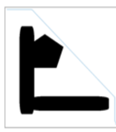
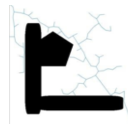
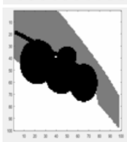
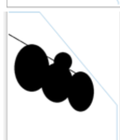
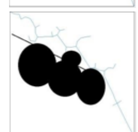
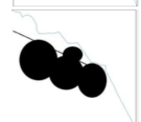
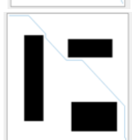
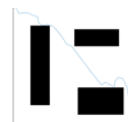
The proposed system clarifies the rationale utilized for way arranging from source to goal. Build a virtual map of the environment considering known obstacles. This guide will characterize the limit of the field zone to be secured and the direction required for the way arranging thinking about the given static obstacles. A route field for this calculation is considered in two measurements. The X arranges increase from left to right and Y facilitates increase through.

Source and destination nodes are represented in the map. Planning is done for navigation that starts from source node and reaches desired goals if there are no unknown or newly-discovered obstacles along the path as shown in Fig. 3. Re- planning is done when dynamic obstacle obstructs the desired path from source to goal by generating an alternate feasible pathway.

Discussions and Conclusion

A* algorithm is a well-known method in motion planning problems which can find the optimum path between two point in a finite time. In contrast, the

Table 3 — Comparisons of RRT and A*

Sl. No.	Graphs	Path	Path Length	Processing Time(secs)	Tree	Path	Path Length	Processing Time(secs)
1			414	0.484			414	0.484
2			640	1.249			640	1.249
3			712	1.074			712	1.074
4			753	1.390			753	1.390

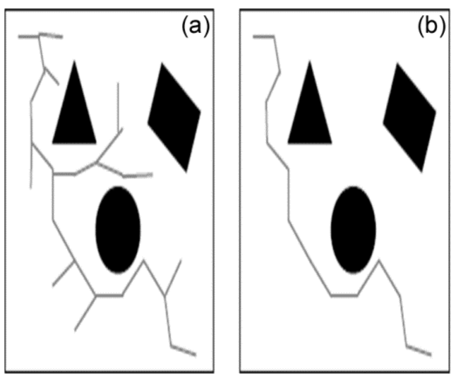


Fig. 3 — RRT Tree (a) Roadmap (b) Path computed

RRT family algorithm, by random sampling from the environment, converges to a collision-free path. By simulating these algorithms in complex environments by using MATLAB, it is concluded that RRT algorithms are significantly faster and produce feasible path than A*. The choice of algorithm is a major factor that contributes in producing high quality results and ensuing in little computation time. Improved RRG adds quality in mobile robots to achieve multiple goals planning problems. Promote from an explanatory perspective, RRG apparently is superior to A* as illustrated in Table 3. The proposed methodology might be made suited for monitoring and surveillance purposes in various application domains of robotics. It is also utilized to rapidly enhance the touring plan in a dynamic condition.

Acknowledgments

The authors wish to thank the Director, Centre for Incubation, Innovation, Research and Consultancy (CIIRC) for his inputs during the research investigations. The authors wish to thank Principal Jyothy Institute of Technology, Bengaluru and Principal, Maharaja Institute of Technology, Mysore India for providing research laboratory facilities.

References

1 Siegwart R, Nourbakhsh I R & Scaramuzza D, *Introduction to Autonomous Mobile Robots*, *A Bradford Book* (The MIT Press) 2004, 365–419.

2 Raja P & Pugazhenthi S, Optimal path planning of mobile robots: A review, *Int J Phys Sci*, **7(9)** (2012) 1314–1320.

3 Ferguson D & Stentz A, The Delayed D* Algorithm for Efficient Path Replanning, in *IEEE Int Conf on Rob and Auto* (Barcelona, Spain), 18–22 April 2005.

4 Joshi H N & Shinde J P, An Image Based Path Planning And Motion Planning for Autonomous Robot, *Int J Comput Sci Inf Technol*, **5(4)** (2014) 4844–4847.

5 Elayaraja D & Ramabalan S, Investigation in Autonomous Line Follower Robot, J *Sci Ind Res*, **76(04)** (2017) 212–216.