A New RT Shortest Path Planning Algorithm For Mobile Robot Navigation In Known Terrain

Ravi teja Tirumalapudi, Rajay Vedaraj.I.S

Abstract: Shortest path planning is the basic and fundamental topic in the navigation system of mobile robotics, but this is the main research field of mobile robotics. Previously so many path planning algorithms are proposed. This paper proposed a new methodology for mobile robot path planning in known terrain with rotation and transformation obstacles, inspired by the bug algorithm. Using the Rotation and Transformation algorithm find the optimal shortest path in known terrain. It works by moving a mobile robot in path detect any obstacle whether it moves rotation or transformation, a mobile robot will assume the new shortest path avoid that obstacle. This path is the best path compared by standard bug2 algorithm consider it has a avoid different shapes obstacles to reach goal point. This algorithm is tested in a mobile robot having a laser sensor, shown and included simulation results.

Keywords: Bug1 algorithm, Bug2 algorithm, Graph search, Mobile robotics, Obstacle avoidance, Path planning algorithms, Path Search.

1. INTRODUCTION:

Path planning is the main challenging in robot navigation problems. From last few decades researchers are working in this area. There are immense changes in this field. Many methods are implemented using for path planning. Han-ye Zhang [1] explored so many methods such as Graph based search algorithm (Visibility Graph, Voronoi Graph, Tangent Graph) [2]-[4], Free Space Approach[5], Cell Decomposition Approach[6], Topological Method[7], Probabilistic Roadmap Method[8], Path search algorithm (Dijkstra algorithm, A* algorithm, D* algorithm)[9]-[15], Artificial Intelligence Algorithm (ANN, GA, ACO, PSO, SA)[16]-[20]. The main criteria of the path planning are path length, time and weigh of the path. Lumelsky and Stepanov et al [21] implemented simple Bug algorithm also called as common sense algorithm using hit point and leave points in a path. Unlikely this algorithm faced so many challenges. Later again they introduced Bug1 algorithm [22] in 1986 based on bug algorithm with solved faced challenges. In this bug1 algorithm works by search for obstacle entire boarder of the terrain and parallel search nearest path to the target, after avoid hit point next moves the nearest hit point and which path is closest to target. Then it will leave hit point. Main drawback of this Bug1 algorithm is unnecessarily searching all paths and borders of the terrain, it takes much time. So again they introduced Bug2 algorithm [23] in 1986 based on Bug1 algorithm. Bug2 algorithm works by drawn lines between staring point to goal points along with boarder. This line is called as M-Lines. The robot will follows the drawn line up to hit another m-line, then change the M-line and moves to target position. This paper explored the RT shortest path planning algorithm with considering the avoid rotation [24] and transformation [25],[26] moving obstacles also find the shortest path between start point to goal position.

- *Raviteja is currently pursuing Doctoral degree program in mechanical engineering in VIT, India, PH-+917416799380. E-mail: tirumala.raviteja@vit.ac.in*
- *Rajay Vedaraj I.S is Associate Professor in mechanical engineering in VIT, India, PH-+919486526615 E-mail: rajay@vit.ac.in*

This algorithm simulated in matlab and tested different known terrains. Part II explain RT Shortest path planning algorithm in known terrain. Part III showed simulation results. Part IV discussions on results compare with Bug2 algorithm.

2. RT SHORTEST PATH PLANNING ALGORITHM:

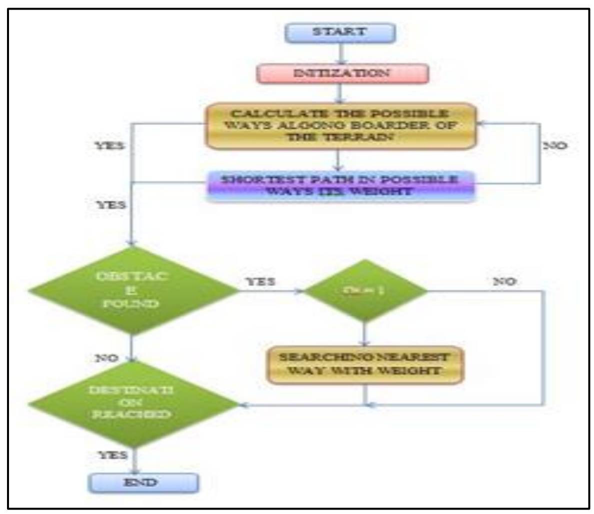

Fig 1. RT Shortest Path Planning Flow Chart

This algorithm mainly focussed on find shortest and avoid different shapes of obstacles. Its works mobile robot scans the full terrain along boarder which is possible ways to reach target position. Let as assume S is start position and G is Goal point. Weight of the path is represented as Wp. Which is d is the distance of distance and Sd is the shortest distance or best path of the starting position to goal position.

$$Sd \leq l + 1/2 \sum_{i=1}^{N} Oi$$

The above equation is modified for the solve shortest distance of the given terrain.

$$\mathrm{Wr} = \frac{\mathrm{W}}{\mathrm{UiWoi}} - \mathrm{Woi}$$

From the Wr equation solve the working space upper bound to the free space lower bound. That is called as required workspace.

3. PROBLEM

of this paper is find shortest path in known terrain having rotation and transformation obstacles avoiding. Main two tasks of this problem is avoiding obstacles and find path between starting goal to end point using this algorithm.
We build a mobile robot environment model code in matlab. For finding the distance between 2 coordinate points used Euclidean distance between two points.

$$\text{distance} = \sqrt{(\text{x1} - \text{y1})2 + (\text{x2} - \text{y2})2}$$

Input:
Snode: Start Node
Gnode: Goal Node
O: Obstacles
Length: length between the two points
Wi: Wight between the points

Output:
Shortest path between the Start node to goal node.

/*Function dist = distance(x1, y1, x2, y2)
/*dist = sqrt ((x1-y1) ^2+(x2-y20^2

Snode → X1 : nearest node
Calculate the weight
X1, X2, and X3….Xn → Gnode

1. Start from the source; initial path length and weight considering is zero.
2. If find any obstacle; change the all possible traveling path.
3. Else
4. Destination reached.

For finding the possible ways between the start node to goal node.
/*from start node to goal node distance = dist (xTarget, yTarget,x,y)
weights calculated between two points.
/*Weight of the between two nodes = dist+distance (x_node,y_node,x,y)

4. SIMULATION RESULT:

RT Path planning algorithm solved in matlab based Bug2 algorithm [27],[28], but changes that algorithm added shortest path program. We tested this algorithm in different terrains shown in Table 1. In terrain 2 algorithm works not hit any obstacles and weight value is 2.8 sec only. In terrain 1 it hit obstacle and weight value also 4.6 sec only.

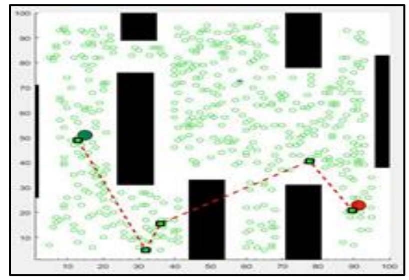

Fig 3. *Terrain 1 path between Start to goal*

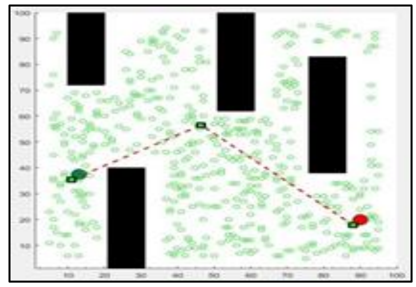

Fig 4. *Terrain 2 path between Start to goal*

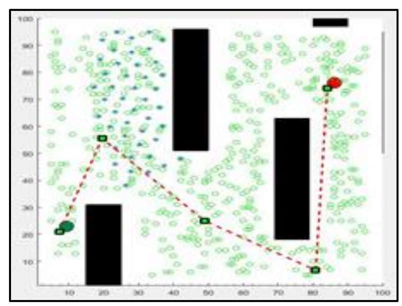

Fig 5. *Terrain 3 path between Start to goal*

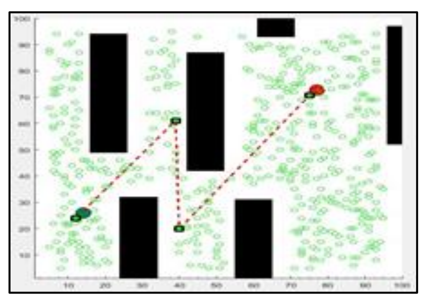

Fig 6. *Terrain 4 path between Start to goal*

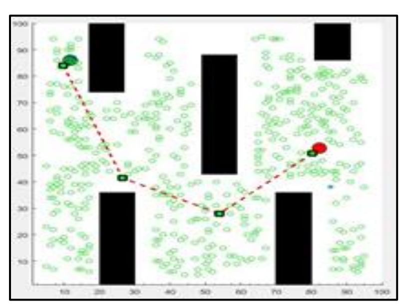

Fig 7. *Terrain 6 path between Start to goal*

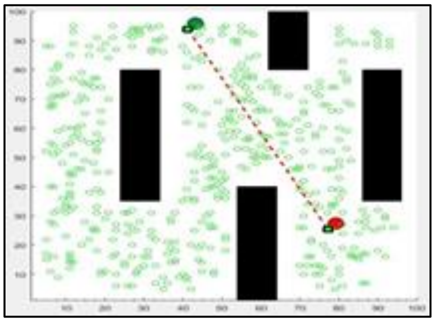

Fig 8. *Terrain 7 path between Start to goal*

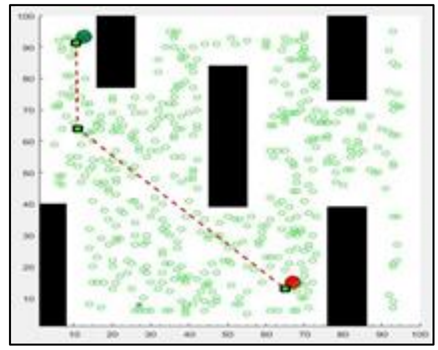

Fig 9*. Terrain 8 path between Start to goal.*

Table 1. *Simulation results along path length and weight*

Environment	Grid Size	Obstacle Status	Path Length	Weight	Environment	Grid Size	Obstacle Status	Path Length	Weight
Terrain 1	100*100	Free	45.2	4.6	Terrain 37	100*100	Free	52.8	3.2
Terrain 2	100*100	Free	40	2.8	Terrain 38	100*100	Free	45	3
Terrain 3	100*100	Free	47	4.9	Terrain 39	100*100	Free	42	3.4
Terrain 4	100*100	Hit	51	5.2	Terrain 40	100*100	Free	47.5	4.8
Terrain 5	100*100	Free	46	4.7	Terrain 41	100*100	Free	56.8	5
Terrain 6	100*100	Free	49	5.2	Terrain 42	100*100	Free	54	4.5
Terrain 7	100*100	Free	55	3.4	Terrain 43	100*100	Free	52.8	4.8
Terrain 8	100*100	Free	57	6.1	Terrain 44	100*100	Free	54	4.2
Terrain 9	100*100	Free	42	3.1	Terrain 45	100*100	Free	48	3.8
Terrain 10	100*100	Free	56	5	Terrain 46	100*100	Free	40	2.9
Terrain 11	100*100	Free	41	3.1	Terrain 47	100*100	Free	42	3.1
Terrain 12	100*100	Free	40	2.9	Terrain 48	100*100	Free	43	3.2
Terrain 13	100*100	Free	40.8	2.9	Terrain 49	100*100	Free	47	3.8
Terrain 14	100*100	Free	42.5	4	Terrain 50	100*100	Free	46	3.7
Terrain 15	100*100	Free	54	4.5	Terrain 51	100*100	Hit	52	4.2
Terrain 16	100*100	Free	43.2	3.1	Terrain 52	100*100	Free	40	4.1
Terrain 17	100*100	Free	41	2.9	Terrain 53	100*100	Free	41	3
Terrain 18	100*100	Free	56	4.8	Terrain 54	100*100	Free	44	4.4
Terrain 19	100*100	Free	53	4.2	Terrain 55	100*100	Free	45	4.6
Terrain 20	100*100	Free	59	3.5	Terrain 56	100*100	Free	46	4.5
Terrain 21	100*100	Free	45	3.2	Terrain 57	100*100	Free	49	4.6
Terrain 22	100*100	Hit	50.5	4	Terrain 58	100*100	Free	45	4.4
Terrain 23	100*100	Free	42	3.4	Terrain 59	100*100	Free	58	5
Terrain 24	100*100	Free	54.6	3	Terrain 60	100*100	Free	54	4.9
Terrain 25	100*100	Free	46	3.1	Terrain 61	100*100	Free	54	4.9
Terrain 26	100*100	Free	44	3.6	Terrain 62	100*100	Free	42	3.3
Terrain 27	100*100	Free	47	4.4	Terrain 63	100*100	Free	48	4.6
Terrain 28	100*100	Hit	51.2	3	Terrain 64	100*100	Free	54	4.8
Terrain 29	100*100	Free	48	5.3	Terrain 65	100*100	Hit	51.6	4.4
Terrain 30	100*100	Free	53	3.2	Terrain 66	100*100	Free	41.8	3
Terrain 31	100*100	Free	58	6.3	Terrain 67	100*100	Free	42	3.2
Terrain 32	100*100	Free	48.5	5	Terrain 68	100*100	Free	54.1	4.6
Terrain 33	100*100	Free	49	5.1	Terrain 69	100*100	Free	55	4.7
Terrain 34	100*100	Free	48.1	5	Terrain 70	100*100	Free	53	4.4
Terrain 35	100*100	Free	43	3.6	Terrain 71	100*100	Free	46	4.2
Terrain 36	100*100	Free	44	3.6	Terrain 72	100*100	Free	43.5	3.8

5. CONCLUSIONS AND FUTURE WORK:

This algorithm is suited for known terrain for finding shortest path compared with previous available algorithms. For extension of the test in unknown environments also work. Suppose its works in autonomous vehicles so that find the shortest path between start position to goal position. Main challenge of this algorithm is consider the weight of the path along distance also according characteristics and considerations.

6 . REFERENCES

[1] Zhang Han Ye, Lin Wei Ming and Chen Ai Xia," Path planning for the mobile robot: A review", Symmetry 10(10), 2018.

[2] You, Yangwei, Caixia Cai, and Yan Wu. "3D Visibility Graph based Motion Planning and Control." (2019).

[3] Ellips Masehia, M. R. Amin-Naseri, " A voronoi diagram-visibility graph-potential field compound algorithm for robot path planning", journal of field robotics 2014, 21(6), pp 275-300.

[4] Wang, Haiyan, and Zhiyu Zhou. "A heuristic elastic particle swarm optimization algorithm for robot path planning." Information 10.3 (2019): 99.

[5] Hwang, Yong Koo, and Narendra Ahuja. "A potential field approach to path planning." IEEE Transactions on Robotics and Automation 8.1 (1992): 23-32.

[6] Lingelbach, Frank. "Path planning using probabilistic cell decomposition." IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004. Vol. 1. IEEE, 2004.

[7] Bhattacharya, Subhrajit, Maxim Likhachev, and Vijay Kumar. "Topological constraints in search-based robot path planning." Autonomous Robots 33.3 (2012): 273-290.

[8] Siméon, Thierry, J-P. Laumond, and Carole Nissoux. "Visibility-based probabilistic roadmaps for motion planning." Advanced Robotics 14.6 (2000): 477-493.

[9] Edsger W. Dijkstra, "A Note on Two Problems in Connection with Graphs" Numerische Mathematik. 23 (3), pp 269–271, 1959.

[10] Duchoň, František, et al. "Path planning with modified a star algorithm for a mobile robot." Procedia Engineering 96 (2014): 59-69.

[11] Reher, Jenna, and Wenlong Ma. "Dynamical Path Planning For Humanoid Robot Atlas Via D* Lite." (2019).

[12] Stentz, Anthony (1994), "Optimal and efficient path planning for unknown and dynamic Environments ", CMU-RI-TR-93-20, The Robotics institute, Carnegie Mellon University, Aug 1993.

[13] Xu, Feiyi, et al. "A new global best guided artificial bee colony algorithm with application in robot path planning." Applied Soft Computing (2020): 106037.

[14] Dakulović, Marija, and Ivan Petrović. "Two-way D∗ algorithm for path planning and replanning." Robotics and autonomous systems 59.5 (2011): 329-342.

[15] Ferguson, Dave, Maxim Likhachev, and Anthony Stentz. "A guide to heuristic-based path planning." Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS). 2005.

[16] Mohanty, Prases Kumar, and Dayal R. Parhi. "Path planning strategy for mobile robot navigation using MANFIS controller." Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013. Springer, Cham, 2014.

[17] Tu, Jianping, and Simon X. Yang. "Genetic algorithm based path planning for a mobile robot." 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422). Vol. 1. IEEE, 2003.

[18] Garcia, MA Porta, et al. "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation." Applied Soft Computing 9.3 (2009): 1102-1110.

[19] Qin, Yuan-Qing, et al. "Path planning for mobile robot using the particle swarm optimization with mutation operator." Proceedings of 2004 international conference on machine learning and cybernetics (IEEE Cat. No. 04EX826). Vol. 4. IEEE, 2004.

[20] Park, Min Gyu, Jae Hyun Jeon, and Min Cheol Lee. "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing." ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570). Vol. 3. IEEE, 2001.

[21] Wu, Qing, et al. "Real-Time Dynamic Path Planning of Mobile Robots: A Novel Hybrid Heuristic Optimization Algorithm." Sensors 20.1 (2020): 188.

[22] Lumelsky, Vladimir J., and Alexander A. Stepanov. "Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape." Autonomous robot vehicles. Springer, New York, NY, 1990. 363-390.

[23] Lumelsky, V., and Alexander Stepanov. "Dynamic path planning for a mobile automaton with limited information on the environment." IEEE transactions on Automatic control 31.11 (1986): 1058-1063.

[24] Fujimura, Kikuo, and Hanan Samet. "A hierarchical strategy for path planning among moving obstacles (mobile robot)." IEEE transactions on robotics and Automation 5.1 (1989): 61-69.

[25] Zhu, Qiuming. "Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation." IEEE Transactions on Robotics and Automation 7.3 (1991): 390-397.

[26] Zelinsky, Alexander, et al. "Planning paths of complete coverage of an unstructured environment by a mobile robot." Proceedings of international conference on advanced robotics. Vol. 13. 1993.

[27] Yang, Shuai, et al. "A New Intelligent Trajectory Planning Algorithm Based on Bug2 Algorithm: Bridge Algorithm." 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2019.

[28] Le, Than. "Real-Time Search-based Planning in Structure Environments." (2020).