Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-1 (2016) 178–182

Terrain Based D* Algorithm for Path Planning

C. Saranya[1], Manju Unnikrishnan[2], S. Akbar Ali[3], D. S. Sheela[4] and Dr. V.R. Lalithambika[5]

[1]Scientist/Engineer, SCG, ISRO Satellite Centre, India
[2]Deputy Division Head, CGFSD, Vikram Sarabhai Space Centre, India
[3]Division Head, CGFSD, Vikram Sarabhai Space Centre, India
[4]Group Director, Control and Guidance Design Group, Vikram Sarabhai Space Centre, India
[5]Deputy Director, Control Guidance and Simulation Entity, Vikram Sarabhai Space Centre, India

Abstract - Path planning algorithms provide autonomy in mobile robots to reach targets even in unknown environments. Path planning using grid based techniques such as A and D*, are cost function based, which is primarily a function of the distance to be travelled to reach the target. Robots targeted for outdoor environments should consider the terrain features also during path planning. In this paper, a modified approach of D* path planning algorithm is proposed. In addition to distance to be travelled, terrain slope estimate is also used in cost function computation to plan the path. The algorithm was simulated and tested with different terrain slopes. Results with different test scenarios are also brought out.*

Index Terms — Terrain based Path planning, D* algorithm

I. Introduction

Mobile robots are supplemented with Path Planning algorithms to enable them to reach target autonomously. Several techniques are available in literature for path planning. Steven M.LaValle(2006) gives an overview on various approaches to Path Planning. Grid based techniques such as A* and D* algorithms are classical methods of path planning. These algorithms reach the target in shortest path, provided path is available. Takayuki Goto et al (2003), briefs on the Heuristics of A* algorithm. GopiKrishnan et al (2011) gives a comparison of different path planning techniques. Y. Koren et al (1991) & Jorge Nieto et al (2010) deal with other approaches to Path Planning.

Grid based algorithms plan the path based on cost functions. Cost functions are based on distance to be travelled to reach the target point. For even-terrain environments distance to be travelled will be the deciding factor for path planning. However for outdoor environments especially un-even terrains, nature of terrain will also play a major role. The effort spent by the rover while climbing up or down an un-even region has to be accounted in addition to the linear distance to be travelled.

Also additional terrain features can be included to make the path planning more efficient. Genya Ishigami et al (2011) & Xie Yuan et al (2010) discuss on approaches to terrain based path planning. For instance nature of the terrain can also be used to determine the regions which are traversable by the rover. Rocky regions can be discarded as obstacles. Marshy and slippery terrains are also to be avoided to prevent wheel slippage.

Terrain Ruggedness index (TRI) gives the amount of elevation difference between the adjacent cells. Based on the range of TRI, the level of ruggedness can be determined. Based on the level of ruggedness, level or nearly level regions can be identified for path planning.

In this paper a modified version of D* algorithm is presented where cost function computation also involves terrain based factors. In addition to the distance to be travelled, a factor based on 3D slope of the terrain is included in the cost function computation.

The algorithm was simulated in C and tested for different terrain slopes. Following is the categorization of the Paper: Section II describes the modified D* algorithm for terrain based path planning. Section III describes the Simulation while Section IV discusses the results. And Section V gives the conclusion.

II. Modified D* Algorithm For Terrain Based Path Planning

D* algorithm is a grid based approach. Ref 8 & Ref 9 give the overview of D* algorithm and its applications. The area of exploration is split into n x n grid. Each cell in the grid has a state to denote the presence or absence of obstacles. The basic cost function of D* is based on distance to be travelled. As the region to be explored is split into squares, the cost involved in moving along and across the cell is the same. Moving to the diagonal cell requires additional cost. Let ‘p’ be the cost of moving horizontally or vertically in the cell. Then cost function is given in equation 1

$$\text{Cost}(x,y)_{hor|ver} = p * \text{units}$$

$$\text{Cost}(x,y)_{diag} = (\sqrt{2} * p) * \text{units} \qquad (1)$$

The above cost function is modified to include a factor based on the terrain slope. The slope of terrain and its azimuth is computed from 3D points.

Consider two points A(x1,y1,z1) and B(x2,y2,z2). The slope between the points A and B is computed as per equation 2. The azimuth (in degrees) or the orientation angle measured clockwise from North is given in equation 3

$$\text{Slope} = (z2-z1) / \sqrt{((x2-x1)^2 + (y2-y1)^2)} \qquad (2)$$

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.03.049

$$\text{Azimuth} = (180 / \prod)\ \tan^{-1}((x2-x1)/(y2-y1)) \quad (3)$$

Cost function of D* algorithm is modified as per equation 4 to accommodate a term based on the terrain slope. The factor in the equation can be tuned based on different terrains.

$$\text{Cost\_Function} = \text{Cost}(x,y) + \text{abs(slope)} * \text{Fact} \quad (4)$$

Ascending a slope is given a higher factor compared to descending. For moving across a level terrain factor is made zero. The cost of moving vertically or horizontally in a cell of size 40 cms is taken as 10 units. The cost of moving diagonally is assumed to be 14 units. Factor for ascending is assumed to be 20 while for descending it is taken as 10.

If ascending

Fact = 20

Else if descending

Fact = 10

Else

Fact = 0

End If

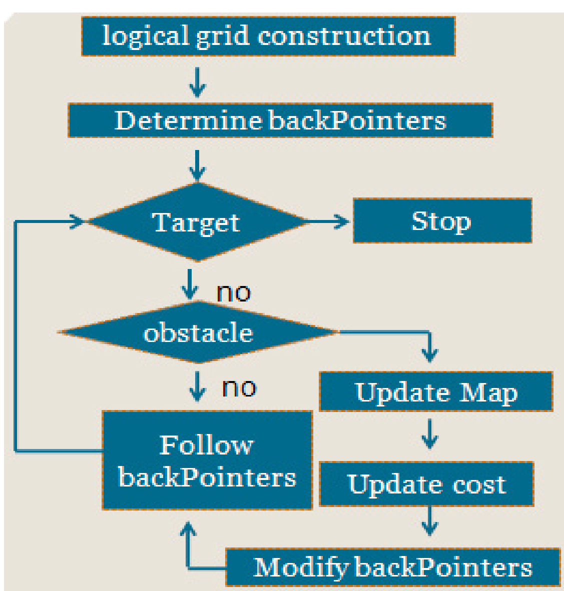

Figure 1: D* algorithm

Based on the above equations, cost functions are computed for all the cells. Rover moves through the cells which have the least cost functions. The flowchart of the algorithm is given in Figure 1.

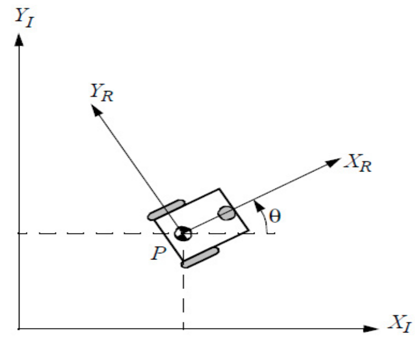

Figure 2: States of Rover

III. SIMULATION

Modified D* algorithm for Terrain based path planning was implemented in C.

An ideal rover was assumed. Cost functions computed based on the terrain information of the adjacent cells is used to identify the cell with the minimum cost function. Rover then moves to the chosen cell.

During experimentation each cell was assumed to be of 40 x 40 cms. The cell size could be decided based on the dimensions of the actual rover. It could be planned in such a way that rover will be confined within a cell at any point of time.

The location of the rover can be described with three states(x, y, θ) at any instant as in Figure 2. The orthogonal rotational matrix to map the rover's motion from global reference frame to its local reference frame is given in equation 5.

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Since rover can move to only one of its adjacent eight cells, it has eight possible orientations as in Figure 3.

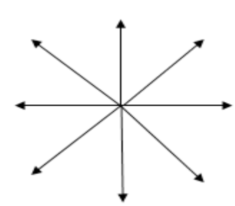

Figure 3: Eight possible orientations of rover as it moves to adjacent cell

The position and orientation of the rover is controlled by commanding the servos of the wheels. When a rover, which does not have the capability to move laterally, has to move to a cell to its left or right, it first orients to the required direction and then moves forward.

The current position and orientation of rover is estimated by accumulating the wheel encoder measurements. For one rotation of the wheel the distance traversed by the rover is given in equation 6

$$\text{Dist} = 2*\text{pi}*r \qquad (6)$$

where, r is the radius of the wheel.

Let p denote the encoder value corresponding to one full rotation i.e encoder reading required for moving by distance, Dist as in equation 6. Then encoder measurement, p1, for moving the rover by a distance of Dist1 units, would be as in equation 7

$$p1 = p* \text{Dist1} / \text{Dist} \qquad (7)$$

Based on equation 7 the reference encoder value is generated for each movement.

IV. RESULTS

Modified D* algorithm was implemented in C and tested in ideal rover condition. Two typical cases are elaborated below. Test case 1 is a 4x4 grid where each cell occupies 40cmx40cm as in Figure 4.

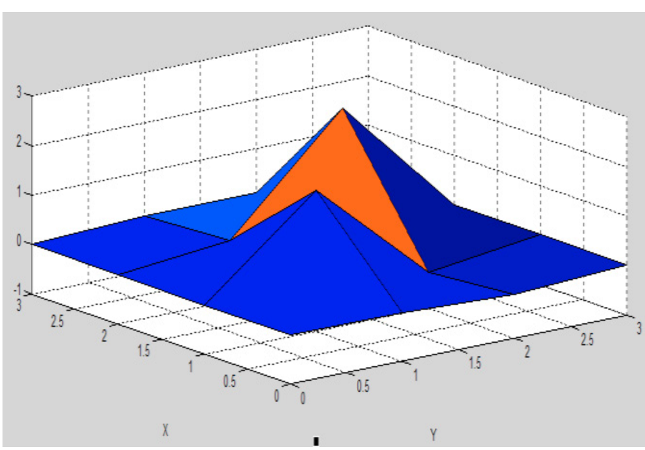

Figure 4: Terrain considered – Test case 1

X/Y	0	1	2	3
0	0.0001	-0.0299	-0.1100	-0.0000
1	-0.0088	1.8559	-0.2729	0.0130
2	-0.0137	0.2289	2.4338	0.0125
3	0.0000	0.1099	0.1107	0.0000

Table 1: Height of cells

Table 1 shows the height of the each cells, of the grid considered, along with X and Y co-ordinates. The Rover has to start at the cell 10 and reach the target 13.

If the cost of moving 4 cm is taken as 1 unit, then for moving along (vertically / horizontally) a cell of 40cms, cost involved is 10 units and the cost of moving diagonally is 14 units. Cost function computed based on the distance to be travelled is given in Table 2.

Path traced by the Rover using D* algorithm is given in Figure 5. Since slope of the terrain is not included in cost function computation path is not optimal.

X/Y	0	1	2	3
0	34	24	14	10
1	30	20	10	0
2	34	24	14	10
3	44	34	24	20

Table 2: Cost function computed based on D*

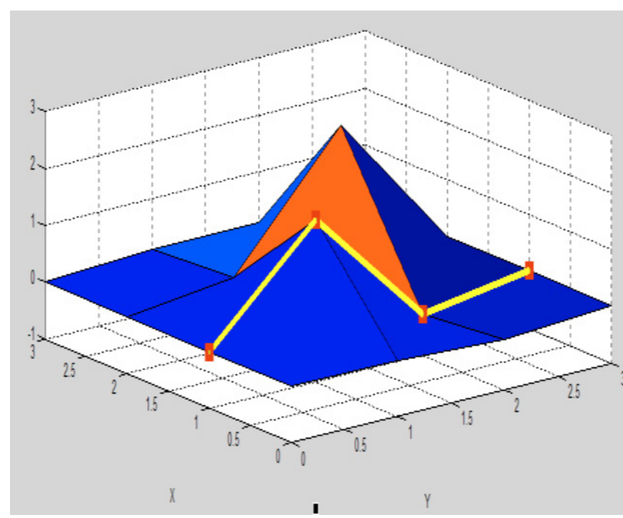

Figure 5: Path traced by Rover using D* algorithm

Cost function computed using the modified D* algorithm involving the slope of the terrain is given in Table 3.

Path traced by the rover using the modified D* algorithm is given in Figure 6. Though shortest path based on distance is through cells 10->11->12->13, since there is a peak in cell 11, rover reaches the targes through cells 10->01->12->13.

X/Y	0	1	2	3
0	55.6745	31.4363	15.7395	10.26
1	45.5855	47.006	15.718	0
2	55.6835	33.2663	31.1176	10.01
3	50.5034	34.7204	24.7044	20.26

Table 3: Cost function computed based on Modified D*

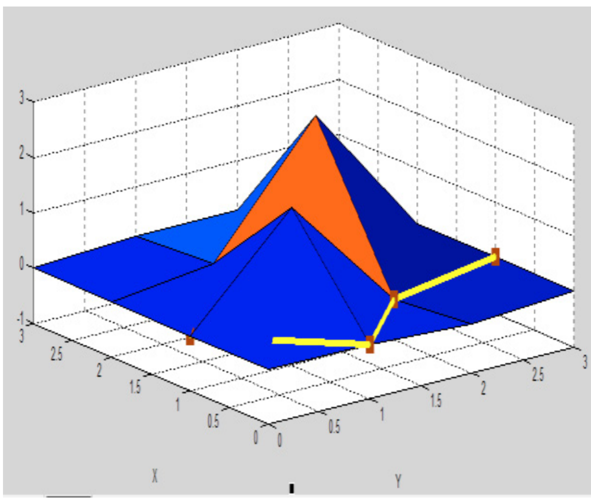

Figure 6: Path traced by Rover using Modified D* algorithm

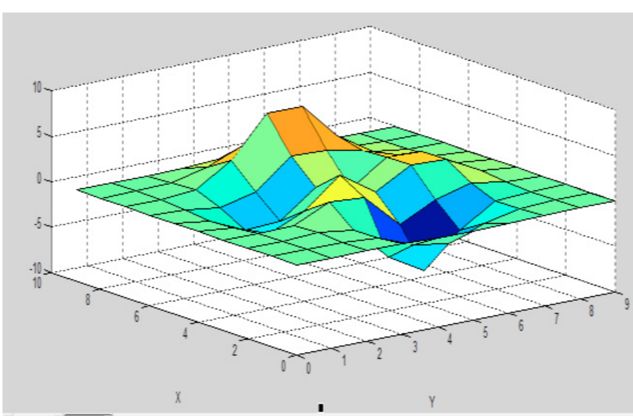

Figure 7: Terrain considered – Test case 2

Another test case with 10 x 10 grid as in Figure 7, is elaborated below. The height of each cell is given in Tables 4 & 5. The rover has to start from cell 93 and reach cell 09. The cost function computed by D* algorithm is given in Table 6 while the one computed using the modified D* algorithm is given in Tables 7 & 8

X/Y	0	1	2	3	4
0	0.0001	0.0013	0.0053	-0.0299	-0.1809
1	0.0005	0.0089	0.0259	-0.3673	-1.8670
2	0.0004	0.0214	0.1739	-0.3147	-4.0919
3	-0.0088	-0.0871	0.0364	1.8559	1.4995
4	-0.0308	-0.4313	-1.7334	-0.1148	3.0731
5	-0.0336	-0.4990	-2.3552	-2.1722	0.8856
6	-0.0137	-0.1967	-0.8083	0.2289	3.3983
7	-0.0014	-0.0017	0.3189	2.7414	7.1622
8	0.0002	0.0104	0.1733	1.0852	2.6741
9	0.0000	0.0012	0.0183	0.1099	0.2684

Table 4: Height of cells 00 to 94

X/Y	5	6	7	8	9
0	-0.2465	-0.1100	-0.0168	-0.0008	-0.0000
1	-2.4736	-1.0866	-0.1602	-0.0067	0.0000
2	-6.4101	-2.7589	-0.2779	0.0131	0.0020
3	-2.2171	-0.2729	0.8368	0.2016	0.0130
4	0.4444	2.6145	2.4410	0.4877	0.0301
5	-0.0531	2.6416	2.4064	0.4771	0.0294
6	3.1955	2.4338	1.2129	0.2108	0.0125
7	7.1361	3.1242	0.6633	0.0674	0.0030
8	2.6725	1.1119	0.1973	0.0152	0.0005
9	0.2683	0.1107	0.0190	0.0014	0.0000

Table 5: Height of cells 50 to 99

X/Y	0	1	2	3	4	5	6	7	8	9
0	90	80	70	60	50	40	30	20	10	0
1	94	84	74	64	54	44	34	24	14	10
2	98	88	78	68	58	48	38	28	24	20
3	102	92	82	72	62	52	42	38	34	30
4	106	96	86	76	66	56	52	48	44	40
5	110	100	90	80	70	66	62	58	54	50
6	114	104	94	84	80	76	72	68	64	60
7	118	108	98	94	90	86	82	78	74	70
8	122	112	108	104	100	96	92	88	84	80
9	126	122	118	114	110	106	102	98	94	90

Table 6: Cost function computed based on D*

The path traced by the rover according to D* algorithm is shown in Figure 8. Figure 9 gives the path traced by rover as per Modified D* algorithm. Rover travels more distance to reach the target using Modified D* algorithm, but it avoids peak and thus reaches target in an efficient way.

X/Y	0	1	2	3	4
0	97.552	87.528	77.448	67.096	55.586
1	101.539	91.4735	81.4906	72.2221	109.105
2	105.594	95.5542	90.049	82.7481	129.925
3	109.981	107.74	99.2307	100.062	86.4982
4	122.138	119.845	183.62	115.369	112.234
5	136.657	131.199	180.177	166.517	82.1573
6	148.631	144.222	190.161	105.444	110.629
7	159.603	209.865	120.081	133.919	147.663
8	223.878	138.444	132.993	139.499	137.46
9	152.591	149.427	151.763	139.931	126.761

Table 7: Cost function computed based on Modified D* - cells 00 to 94

X/Y	5	6	7	8	9
0	44.93	32.2	20.336	10.016	0
1	93.0388	49.4653	26.2703	14.0948	10
2	186.289	77.0214	31.9301	24.0926	20.02
3	94.8525	45.9655	43.9171	35.4314	30.13
4	65.0375	77.0013	65.2663	47.4866	40.301
5	84.9875	80.6848	75.9583	57.4618	50.315
6	98.6015	90.1521	76.6647	65.5977	60.653
7	137.402	107.056	82.7973	75.0412	70.843
8	127.444	99.9694	89.9597	84.9293	80.893
9	116.76	105.184	98.9561	94.8994	90.903

Table 8: Cost function computed based on Modified D* - cells 50 to 99

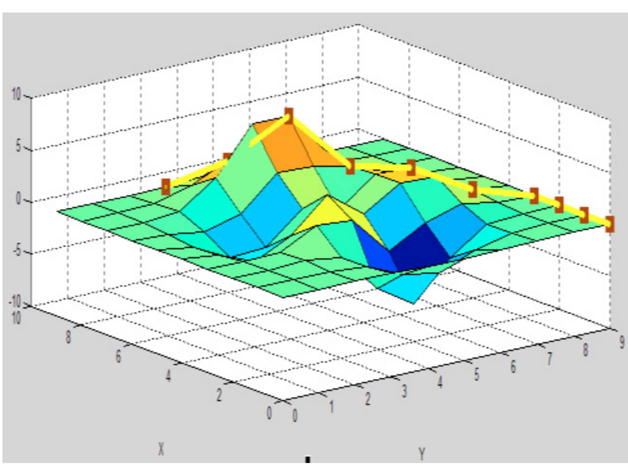

Figure 8: Path traced by Rover using D* algorithm

V. CONCLUSION

D* is an efficient path planning algorithm for autonomous navigators, even for unknown environments. Real world outdoor environments are usually un-even terrain. Path planning using modified D* algorithm involving the terrain slope for cost computation is successfully demonstrated in this paper. The algorithm was simulated in C and tested with various input conditions. Path planned with modified D* algorithm produces efficient results compared to D* algorithm.

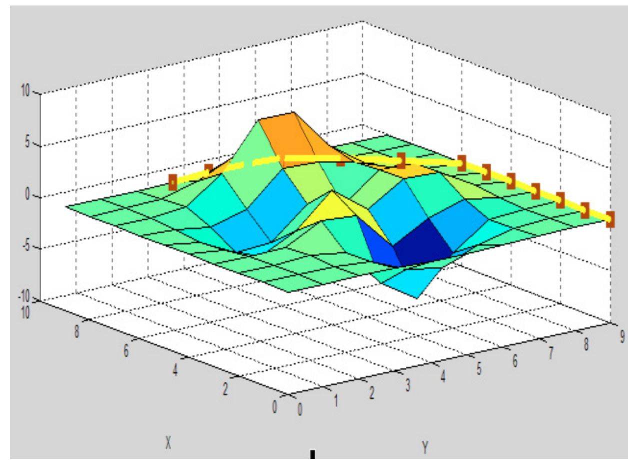

Figure 9: Path traced by Rover using Modified D* algorithm

REFERENCES

Steven M.LaValle (2006), *'Planning Algorithms'*, Cambridge University Press.

Takayuki Goto, Takeshi Kosaka, and Hiroshi Noborio (2003), "On the Heuristics of A* or A Algorithm in ITS and Robot Path-Planning", *Proceedings of the 2003 IEEWRSJ, InU. Conference on Intelligent Robots and Systems*

GopiKrishnan S, BVS Shravan, Harishit Gole, Pratik Barve and Ravikumar L (2011), "Path Planning Algorithms: A comparative study", *Space Transportation Systems,* Trivandrum, VSSC.

Y. Koren, Senior Member, IEEE and J. Borenstein, Member, IEEE (1991) , "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation", *Proceedings of the IEEE Conference on Robotics and Automation, Sacramento, California*, April 7-12, 1991, pp. 1398-1404.

Jorge Nieto, Emanuel Slawinski, Vicente Mut and Bernardo Wagner (2010), "Online Path Planning based on Rapidly-Exploring Random Trees", *Industrial Technology (ICIT), 2010 IEEE International Conference* , 14-17, March 2010.

Genya Ishigami, Keiji Nagatani, and Kazuya Yoshida, "Path Planning and Evaluation for Planetary Rovers Based on Dynamic Mobility Index" *IEEE/RSJ International Conference on Intelligent Robots and Systems September 25-30, 2011.* San Francisco, CA, USA

Xie Yuan, Zhou Jianliang and Wang Yong, "Dynamic Mission-level Path Planning for Lunar Rovers", International Conference on Control, Automation and Systems 2010 Oct. 27-30, 2010 in KINTEX, Gyeonggi-do, Korea

Anthony Stentz, "Optimal and Efficient Path Planning for partially known environment", IEEE International Conference on Robotics and Automation, May 1994.

Joseph Carsten and Arturo Rankin, Dave Ferguson, Anthony Stentz, "Global Planning on the Mars Exploration Rovers: Software Integration and Surface Testing", Journal of Field Robotics 26(4), 337–357, 2009