Survey and the Relative Issues on the Path Planning of Mobile Robot in Rough Terrain [1]

Ouanezar Aissa
Mechanical and Electrical department
Harbin Engineering University
Harbin City, 150001
issaouan@yahoo.fr

He Xu
Mechanical and Electrical department,
Harbin Engineering University
Harbin City, 150001
railway_dragon@163.com

Gang Zhao
Mechanical and Electrical department,
Harbin Engineering University
Harbin City, 150001
zhaogang@hrbeu.edu.cn

Abstract

This paper presents a survey about autonomous mobile robot path planning in rough terrain focusing on algorithms that produce an optimal path for a robot to navigate in rough terrain-, the modeling of terrain and performing the cost function used in path planning algorithms by integration of energy consumption.

Keywords: Path planning, rough terrain, cost function, triangulation, mobile robot.

1 Introduction

The problem of path planning of mobile robots consists of selecting the geometric path and vehicle model [6], so as to avoid obstacles and to minimize some cost function such as time or energy consumption. While extensive work focused on computing the geometric path [1, 2, 4, 5, 8, 7, 9, 10, 11, 12 and 13], little attention has been given to energy consumption. In this field we focus in this paper to study parameters influencing path planning of mobile robot in rough terrain basing in surveying and study of some works developed in the latest years [9], [22] and [23]. For this we at first need make a difference between path planning in rough terrain and planning in well-structured environments.

1. In rough terrain, Terrain data cannot be assumed to be perfectly known, due to errors in range sensing techniques and sensors miscalibration [9].

2. The planned path may not be accurately followed by the robot due the significant path-following and localization errors[9]

3. In rough terrain, the concept of an obstacle is not clearly defined, as it depends on an understanding of the terrain and the mobility characteristics of the robot [9].

4. Ground exploration systems will generally have limited computational resources to devote to path planning [9]

These five points make some complication for path planning. And in path planning in rough terrain we need to consider:

[1] This work was supported by the National Science Foundation of China under Grant 60775060, the Foundation Research Fund of Harbin Engineering University under Grant HEUFT07027, Natural Science Foundation of the Heilongjiang Province of China under Grant F200801, Specialized Research Fund for the Doctoral Program of Higher Education under Grant 200802171053.

- Path planning algorithms for mobile robots in rough terrain must explicitly consider the robot terrain interaction. And the rough terrain path planning algorithms must also consider real-world implementation issue.
- Rough terrain path planning algorithms must consider uncertainty inherent in terrain sensing systems such as rangefinders.

Planning algorithms must account for the fact that the mobile robot cannot precisely track a planned path due to the path-following errors.

Path planning of mobile robots in rough terrain makes us more opportunity for contact with nature that obliges us to have good knowledge with the middle through to reach our goal. Rough terrain path planning algorithms must consider real-world implementation issue. Planning algorithms must account for the fact that the mobile robot cannot precisely track a planned path due to the path-following errors.

The path planning in rough terrain in the general case has set targets to achieve:

1. Developing a 3-D model of rough terrain for planning the path of our mobile robot.
2. Find and plan a mobile robot path in a rough terrain from a given start location to a given goal location based on the 3-D terrain model developed and the important parameters used in cost function.
3. Find a method for the dynamic path planning and replanning in rough terrain.
4. To improve :

- Optimality: Requirements can be based on minimum distance traveled, time taken, vehicle safety,
- Completeness: Will it find a path if one exists?
- Robot Structure: Does the path planner provide a sufficient but not excessively detailed representation of the vehicle with respect to size and mobility constraints?

These three firsts targets are the parameters which affecting the path planning in rough terrain and the next figure resume the order and a combination between these parameters in path planning.

In next paragraph we will explain some steps to perform the terrain model, and in paragraph 3 we introduce a survey about the graph search algorithms used in path planning algorithms in rough terrain. And final paragraph is for study parameters used in path cost function developed by searchers in the latest years and extract a novel idea.

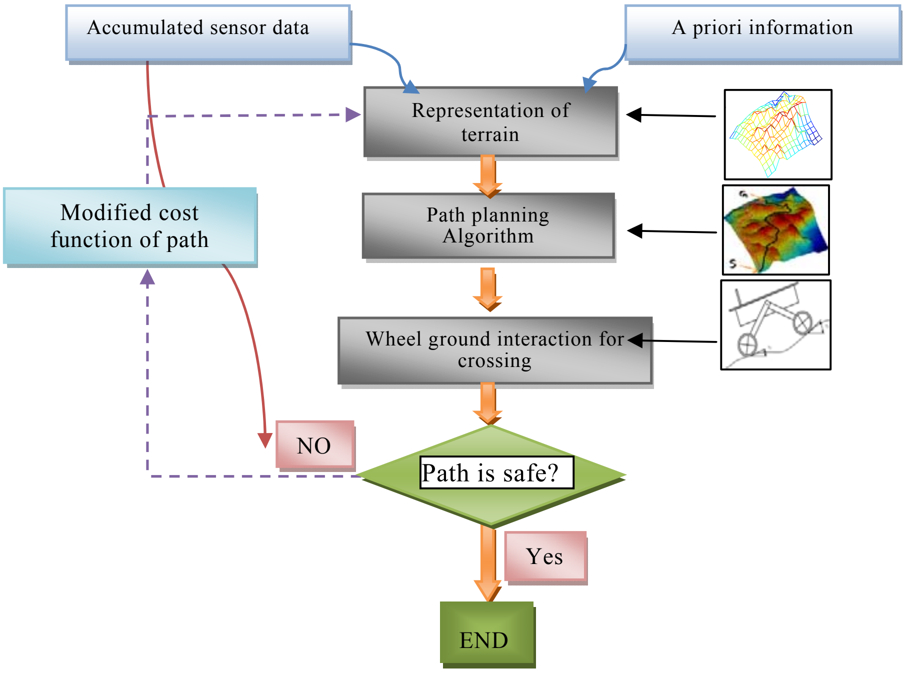

Accumulated sensor data
A priori information
Representation of terrain
Modified cost function of path
Path planning Algorithm
Wheel ground interaction for crossing
NO
Path is safe?
Yes
END

Figure 1: Process of developing of path planning algorithm in rough terrain

2 Modeling and preparing of terrain model

There is some research was initiated to design an algorithm that would guide a mobile robot to explore and map rough terrain. Two-dimensional polygonal maps which have been extensively used in planar environments ([24], [25] and [26]) are definitely not suitable to represent three-dimensional rough terrain. Some research ([27]; [28] and [29]) described 3D grid maps for representation of 3D outdoor environments. However grid maps will become intractable for rough terrain. Dupuis in [30] used the triangular mesh map to represent a Mars-like environment in a planetary exploration task. The triangular mesh map have efficacy in representing large-size environments, such as rough terrain. The triangular mesh map also has advantages over grid maps in its ability to generate a smoother path for navigation tasks. Every triangle has 3 edge neighbors and 9 vertex neighbors, so it has 12 cell neighbors in a non-boundary triangle cell. The triangle mesh map provides 12 moving direction choices for each location, which results in a much smoother path compared with the traditional grid map.

In this path planning issue, it is assumed that a terrain map where a mobile robot travels has been already given without any uncertainties. As shown in Fig. 2, the terrain map is represented as Digital Elevation Map (DEM) which is defined by a series of elevations along with a node *ni* in (*xi*, *yi*, *zi*). Subscript *i* mean a node number. It is generally known that such DEM-based terrain map can be easily measured and developed by using a Laser Range Finder (LRF) or stereo camera systems mounted on a rover in practical situation.

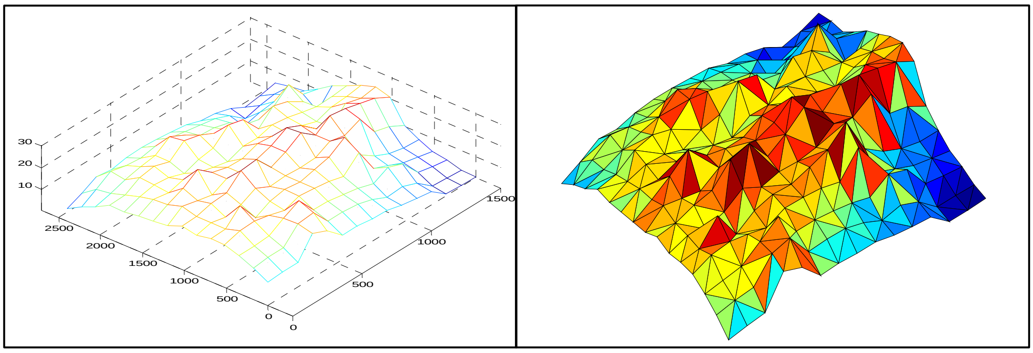


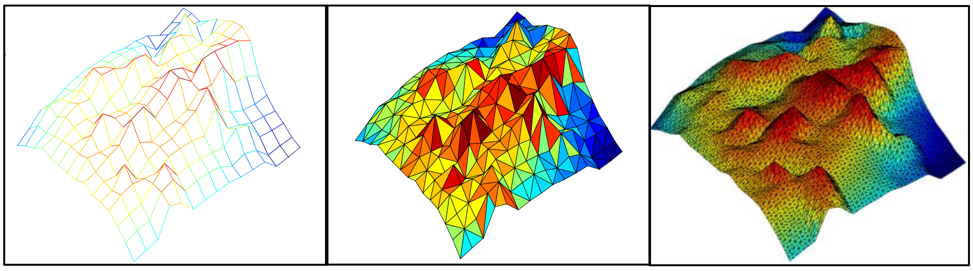
Figure 2: Terrain map example

2.1 Performing terrain model:

For performing the model of terrain her we follow these steps:

1. Using a database contain a coordinate points of our model to make the original figure of terrain. Please see figure.3 a).
2. Using of a Delaunay triangulation to obtain more details about terrain. Please see figure.3 b).

Using a Delaunay triangulation and approximation based in least-square function to obtain more developed and detailed terrain. Please see figure.3c).



a) b) c)

Figure 3: a) original model of terrain with first database. b) New model with Delaunay triangulation. c) Delaunay triangulation and new approximation function with least-square formula

2.2 Approximation based in least- square function:

Is to find a function that is as close to the original data points as possible.

We can write ***z*** in this form:

$$z = a_1 + a_2 x + a_3 y + a_4 x^2 y + a_5 x y^2 + \ldots + a_{(2p+2)} x^p y^p \tag{1}$$

The aim is to find a polynomial with a minimal error.

We can write error in this form:

$$E^2 = \sum_{i=1}^{n} \left[z_i - \left(a_1 + a_2 x + a_3 y + a_4 x^2 y + a_5 x y^2 + \ldots + a_{(2p+2)} x^p y^p \right) \right]^2 \tag{2}$$

The minimum of this error is found by setting the partial derivatives equal to 0.

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 & x_1^2 y_1 & x_1 y_1^2 & x_1^2 y_1^2 & \ldots & x_1^p y_1^p \\ 1 & x_2 & y_2 & x_2 y_2 & x_2^2 y_2 & x_2 y_2^2 & x_2^2 y_2^2 & \ldots & x_2^p y_2^p \\ . & . & . & . & . & . & . & \ldots & . \\ . & . & . & . & . & . & . & \ldots & . \\ . & . & . & . & . & . & . & \ldots & . \\ 1 & x_n & y_n & x_n y_n & x_n^2 y_n & x_n y_n^2 & x_n^2 y_n^2 & \ldots & x_n^p y_n^p \end{bmatrix} X \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_{(2p+2)} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ . \\ . \\ . \\ z_n \end{bmatrix} \quad (X*A=Z) \tag{3}$$

This equation (3) can be solved for the vector A using MATLAB.

3 Graph search algorithms most used in path planning

This part represents the graph search algorithms most used in path planning (figure 4).and in the table (1) there is strength and weakness of these algorithms.

Numerous algorithms have been developed over the last few years to create real time path planning system for autonomous robots. There are three things or activities that must be followed or carried out by an autonomous robotic system to enable the execution of the task of robot navigation. These activities are mapping and modeling the environment, path planning and driving systems. The selection of an appropriate algorithm in every stage of the path planning process is very important to ensure that the navigation process will run smoothly.

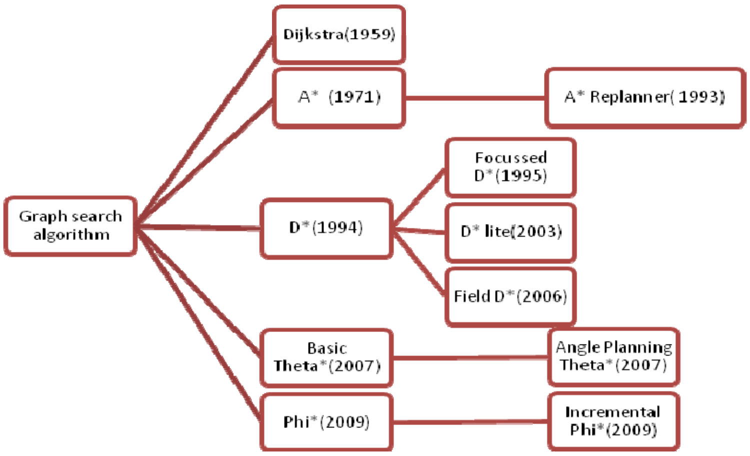


Figure 4: Graph search algorithms most used in path planning

A* Search [15]: A* (pronounced "A star") is a best-first graph search algorithm that finds the least-cost path from a given initial node to one goal node (out of one or more possible goals). It uses a distance-plus-cost heuristic function (usually denoted f(x)) to determine the order in which the search visits nodes in the tree. The distance-plus-cost heuristic is a sum of two functions: The path-cost function, which is the cost from the starting nod to the current node (usually denoted g(x)) and an admissible "heuristic estimate" of the distance to the goal (usually denoted h(x)). This method is analyzed in [15].

D* Search (1994) [18]: The name of the algorithm, D*, was chosen because it resembles A* [8], except that it is *dynamic* in the sense that arc cost parameters can change during the problem solving process. Provided that robot motion is properly coupled to the algorithm, D* generates optimal trajectories. But this algorithm spends a long time in path calculation and subsequent replanning Operations.

The focused D*(1995) [18]: an extension to D* that focuses the repairs to significantly reduce the total time required for the initial path calculation and subsequent replanning Operations. It is real-time path replanning algorithm. The algorithm computes an initial path from the goal state to the start state and then efficiently modifies this path during the traverse as arc costs change. The algorithm produces an optimal traverse, meaning that an optimal path to the goal is followed at every state in the traverse, assuming all known information at each step is correct. The focussed version of D* outperforms the basic version, and it offers the user the option of distributing the computational load amongst the on- and off-line portions of the operation, depending on the task requirements.

D* lite (2006) [21]: a novel replanning method that implements the same navigation strategy as D* but is algorithmically different. D* Lite is substantially shorter than D*, uses only one tie-breaking criterion when comparing priorities, which simplifies the maintenance of the priorities.

Field D*(2006) [16, 17]: an interpolation-based planning and replanning algorithm that alleviates this problem. This algorithm extends D* and D* Lite to use linear interpolation to efficiently produce low-cost paths that eliminate unnecessary turning. The paths are optimal given a linear interpolation assumption and very effective in practice.

Basic Theta*(2007) [19]: any angle path-planning algorithm that finds shorter paths than Field D* for our path-planning problems. The key idea behind Basic Theta*is the parent of a vertex must be a visible neighbor of that vertex.

Angle-Propagation Theta*[19]: (AP Theta*) performs the line of-sight checks in constant time per vertex expansion by calculating and maintaining angle ranges incrementally prior to expanding vertices.

Table 1: strength and weakness of algorithms used in path planning

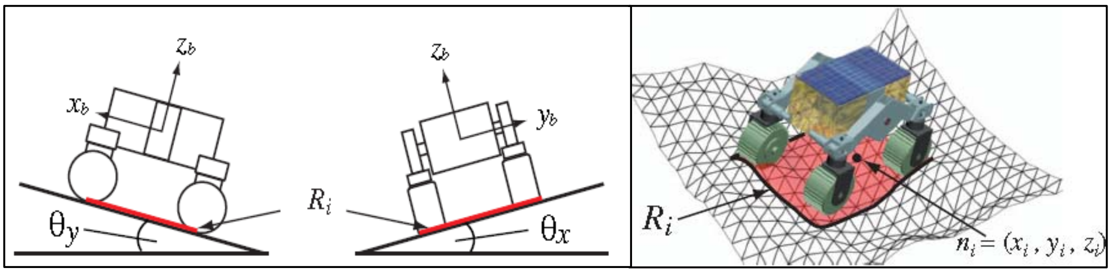
Work space	Type of Algorithm	Strength
Complex environment cluttered with obstacles	Artificial potential field Algorithm [12,13],	**Strength:** Help in robot navigation **Weakness:** Suffer in local minima problem especially in an environment cluttered with obstacles
Complex environment cluttered with obstacles	Cell decomposition +D*[18]	**Strength:** Dynamic path planning. **Weakness:** • Spends a long time in path calculation and subsequent replanning Operations. • Plan from the center of each grid cell and only allow transitions to the centers of adjacent grid cells
Complex environment cluttered with obstacles	Cell decomposition +focussed D*[18]	**Strength:** • Dynamic path planning. • More appropriate than D* in time planning and replanning **Weakness:** -Plan from the center of each grid cell and only allow transitions to the centers of adjacent grid cells
Complex environment cluttered with obstacles	Cell decomposition + Field D* [17]	**Strength:** • Dynamic path planning. • The most appropriate in time consumption, and optimum path.

Work space	Type of Algorithm	Strength
Complex environment cluttered with obstacles	Cell decomposition + D* Lite [21]	**Strength:** 1. Dynamic path planning and more simple and fast calculation time 2. More appropriate than D* and focussed D* in time planning and replanning **Weakness:** • Plan from the center of each grid cell and only allow transitions to the centers of adjacent grid cells
Static environment cluttered with obstacles	Cell decomposition + Basic theta*[19]	**Strength:** • Any angle path-planning algorithm • Shortest path more than field D* **Weakness:** • Spend a long time in path planning calculation
Static environment cluttered with obstacles	Cell decomposition + AP theta*[19]	**Strength:** • Shortest path more than Theta* field D* • Give more information about our vertices **Weakness:** • Spend a long time in path planning calculation
Static environment cluttered with obstacles	Cell decomposition + Phi*[20]	**Strength:** Shortest path more than Theta*
Static environment cluttered with obstacles	Cell decomposition + Incremental Phi*[20]	**Strength:** • incremental version of Phi* and most optimal path algorithm • Faster than repeated single shot Basic Theta* or Phi* searches when cells become blocked.

After analyzing these algorithms basing in the case when the terrain model is given, and applying some condition combing with parameters of cost function base in next paragraph works we can propose for a new generation of path planning algorithms.

4 Parameters affecting calculation of cost function

This paragraph is a summary of some works developed in the latest years which related with parameters affecting calculation of cost function used in developing an optimum path basing in geometry of the terrain and the robot. In general case we can say there is three big research works was developed in this field [9], [22] and [23]. The next table gives an idea about which parameters was used in developing the cost function (please see table 2).



a) b)

Figure5: a) terrain inclination index, b) terrain roughness

Where:

1. Terrain roughness :(Figure5, b) at each grid (x, y), terrain roughness ϕ , B_i and $r'(x,y)$ is defined as the variance of elevation **z** of all points on a circular domain centered at (x, y) with a radius

R_i related to the size of the robot. And terrain roughness have many form used in cost function [9], [14], [22]and [23]

2. Terrain slope: Terrain slope **k** from the grid point p1(x1, y1,z1) to p2 (x2, y2,z2)

$$k_{12} = \frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \tag{4}$$

3. Path length :The travel distance from grid point $p_1(x_1, y_1, z_1)$ to p_2 (x_2, y_2, z_2)

$$D_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{5}$$

4. Terrain inclination index figure (5, a) :

$$\begin{cases} \theta_{x_i} = \bar{\theta}_x(R_i) \\ \theta_{y_i} = \bar{\theta}_y(R_i) \end{cases} \tag{6}$$

Table 2: Parameters affecting calculation of cost function

Cost function Parameters	Research work groups		
	K.Iagnemma 2003 [9]	Genya Ishigami, Keiji Nagatani, and Kazuya Yoshida 2007 [23]	Yi Guo Parker, L.E. Jung, D. Zhaoyang Dong 2004[22]
Terrain roughness	$r'(x,y) = \left(\frac{\sqrt{\mathrm{var}(z(\mathrm{R}))}}{d}\right)^{\alpha 1}$	$\phi(x,y) = \mathrm{var}\sqrt{z(\mathrm{R})}$	$B_i = \sqrt{\frac{1}{n}\sum_{R_i}(z(R_i) - \bar{z}(R_i))^2}$
Robot turning action *t*, Terrain slope k_{12}, Terrain inclination index $\theta_{x_i}, \theta_{y_i}$	$t(x,y) = \left(\frac{\sqrt{\mathrm{var}(z(\mathrm{R}))}}{d}\right)^{\alpha 2}$	$k_{12} = \frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$	$\begin{cases} \theta_{x_i} = \bar{\theta}_x(R_i) \\ \theta_{y_i} = \bar{\theta}_y(R_i) \end{cases}$
Path length	$l(x,y) = \frac{L}{d}$	$D_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$	$L_i = \lvert n_i - n_j \rvert$ $= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$
Cost function	$\phi = k_1 r' + k_2 t + k_3 l + p$	$f_{pp} = \alpha_1 \sum P(\phi, k) + \alpha_2 \sum D$	$C(p) = \sum_{i=p}(W_B N_B B_i +$ $+ W_L N_L L_i + W_{\theta_x} N_{\theta_x} \theta_{\theta_x} + W_{\theta_y} N_{\theta_y} \theta_{\theta_y})$

The calculation of the cost function is based in geometry of robot and the geometry of terrain; so for this we it proposed in [31] a new Calculation based in Integration of energy consumption parameter in cost function. This parameter is presented:

In a graph: Vertex is a place to visit and Edges represent energy consumption. Path planning determines the order to visit these vertices (Route).

The energy consumed by the robot is generated as:

$$E_t = W * \sum_{i=1}^{n} [\square z_i + d_{ni} * \mu_i] \tag{7}$$

Where

E_t =the energy requirement (N m),

W: the weight of the robot (N),

$\square z_i$ Slope height of the ith segment of a piecewise path (m),

d_{ni} Horizontal distance of the ith segment of a piecewise path (m),

$\mu_i = \frac{d_{ni}}{l_i} * B + 0.04$, rolling resistance coefficient (dimensionless),

n = the number of the segments of the path,
B = constant (dimensionless), related with the robot weight, soil hardness, and the tire size
l_i Is the Euclidean distance of the i_{th} segment of a piecewise path (m).

5 Conclusion and trends

This paper has surveyed autonomous mobile robot path planning in rough terrain focusing on the modeling of terrain, algorithms -that produce an optimal path for a robot to navigate in rough terrain-, and performing the cost function used in path planning algorithms by integration of energy consumption. The selection between algorithms based on such criteria as speed of operation, ability to find the shortest path, and applicability to the scale of obstacles and complexity of the environment. Performance is defined at first as a selection of the world representation. Secondly the graph search algorithm needs to suit the representations and the application. Finely the selection of parameters of cost function influencing optimization of path planning algorithms is the key of optimization. New method developed and give the shortest path more than Field D*, is theta* and phi* but they have a problem with time spend in calculation of optimal path. These two methods give a new generation of path planning in rough terrain if we combine with the previews works developed in [9], [22] and [23]. The integration of energy consumption in cost function makes more efficiency of path planning algorithm.

Acknowledgments

The insightful comments and suggestions from the anonymous reviewers are gratefully acknowledged. And the stuffs in our laboratory are also acknowledged.

References

[1] Farbod Fahimi,” Autonomous Robots, Modeling, Path Planning, and Control”, Springer Science + Business Media, LLC 2009

[2] Steven M. LaValle, ” Planning Algorithms”, University of Illinois, 2006

[3] Bruno Siciliano, Oussama Khatib (Eds.), “Springer Handbook of Robotics”, Springer-Verlag Berlin Heidelberg 2008

[4] Yang k. hwang, “Gross motion planning”, ACM computing surveys. Vol 24.No.3. septembre 1992

[5] D. J. Spero,” A Review of Outdoor Robotics Research”, MECSE-17-2004

[6] He Xu, Dawei Tan, Zhenyu Zhang,” A Survey on the Prototypes of Mobile Robot in Rough Terrain”, HEU, Mechanical and Electrical department, Harbin City, 150001

[7] J. Borenstein, H.R. Everett, L. Feng, and D.Wehe,” Mobile Robot Positioning Sensors and Techniques”, Invited paper for the Journal of Robotic Systems.Vol. 14 No. 4, pp. 231 – 249.

[8] Raja Chatila, Simon Lacroix, Michel Devy, Thierry Simeon, ” Autonomous Outdoor Mobile Robot Navigation: The EDEN Project”, Anibal T. de Almeida and Oussama Khatib (Eds) Springer-Verlag London Limited 1998

[9] k.iagnemma S. Dubovsky,” Mobil robot in rough terrain”, Springer verlag berlin Heidelberg 2004

[10] George Kantor, Wolfram Burgard, Lydia Kavraki, and Sebastian Thrun :”Principle of robot motion : theory algorithms and implementation A Bradford Book”, The MIT Press Cambridge, Massachusetts London, England 2005

[11] Roland Siegwart and Illah R. Nourbakhsh,” Introduction to Autonomous Mobile Robots”, the MIT Press Cambridge, Massachusetts London, England 2004

[12] Sadao Kawamura Mikhail Svinin,” Advances in Robot Control”, Springer-Verlag Berlin Heidelberg 2006

[13] A. Bensoussan M.J. Grimble P. Kokotovic.H. Kwakernaak J.L. Massey Y.Z. Tsypkin,” Robot Motion Planning and Control”, Editor Dr J.-P. Laumond 1998

[14] A. Ettlin and H. Bleuler,” Rough-terrain robot motion planning based on obstacleness”, Proc. Int. Conf. on Control, Automation, Robotics and Vision, pp. 1–6, 2006.

[15] N. Sariff 1 and N. Buniyamin 2,” An Overview of Autonomous Mobile Robot Path Planning Algorithms”, 4th Student Conference on Research and Development (Scored 2006), June 2006

[16] J.Carsten and A.Rankin, D. Ferguson and A.Stentz,” Global Path Planning on Board the Mars Exploration Rovers”, IEEEAC paper # 1125 2007

[17] Dave Ferguson and Anthony Stentz,” Field D*: An Interpolation-based Path Planner and Replanner”, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 2006

[18] Anthony Stentz,” Optimal and Efficient Path Planning for Partially-Known Environments”, the Robotics Institute; Carnegie Mellon University; Pittsburgh, 1994

[19] Alex Nash and Kenny Daniel and Sven Koenig,” Theta*: Any-Angle Path Planning on Grids”, Computer Science Department, University of Southern California, Los Angeles, California 90089-0781, USA 2007

[20] Alex Nash and Sven Koenig,” Incremental Phi*: Incremental Any-Angle Path Planning on Grids”, Computer Science Department, University of Southern California, Los Angeles, California 90089-0781, USA 2009

[21] Sven Koenig, Maxim Likhachev,” D* Lite”, College of Computing, Georgia Institute of Technology

Atlanta, GA 30312-0280 2003

[22] Yi Guo , Lynne E. Parker , David Jung and Zhaoyang Dong, “Performance-Based Rough Terrain Navigation for Nonholonomic Mobile Robots”, IEEE/RSJ International Conference on Intelligent Robots and Systems 2004.

[23] Genya Ishigami, Keiji Nagatani, and Kazuya Yoshida,” Path Planning for Planetary Exploration Rovers and Its Evaluation based on Wheel Slip Dynamics”, 2007 IEEE International Conference on

Robotics and Automation, Roma, Italy, 10-14 April 2007

[24] Bourgault, F., A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. 2002. Information Based Adaptive Robotic Exploration. In IEEE/RSJ International Conference on Intelligent Robots and Systems 1: 540- 545. Lausanne, Switzerland. September 30 –October 5.

[25] Stachniss, C. and W. Burgard. 2003. Exploring Unknown Environments with Mobile Robots using Coverage Maps. In Proceedings of the International Joint Conference on Artificial Intelligence(IJCAI) 1127-1134. Acapulco, Mexico. August 9-15.

[26] Taylor, C. and D. Kriegman. 1998. Vision-based motion planning and exploration algorithms for mobile robots. IEEE Trans. On Robotics and Automation 14(3):147–427.

[27] Moorehead, S., R. Simmons, and W.L. Whittaker. 2001. Autonomous Exploration Using Multiple Sources of Information. In Proceedings of the IEEE International Conference on Robotics and Automation 3: 3098- 3103. May 21-26.

[28] Sujan,V. A., and S. Dubowsky. 2005. Efficient Information-based Visual Robotic Mapping in Unstructured Environments. The International Journal of Robotics Research 24 (4): 275-293.

[29] Thrun, S., S. Thayer. W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and W. Whittaker. 2005. Autonomous Exploration and Mapping of Abandoned Mines. IEEE Robotics and Automation Magazine 11(4): 79-91.

[30] Dupuis, E., P. Allard, J. Bakambu, T. Lamarche, and W.H. Zhu. 2004. towards autonomous long-range navigation. In 8th ESA Workshop on Advanced Technologies for Robotics and Automation 'ASTRA 2004', ESTEC. Noordwijk. The Netherlands. November 2-4.

[31]Lifang Liu, Trever G. Crowe, Joseph N. Bakambu, ” Efficient Exploration Algorithms for Rough Terrain Modeling Using Triangular Mesh Maps”, in Robotics, Automation and Mechatronics, 2008 IEEE, 21-24 Sept. 2008, pages: 1206-1211