Research article

Path planning for mobile robot in a 2.5-dimensional grid-based map

Jiajun Gu and Qixin Cao
Research Institute of Robotics, Shanghai Jiaotong University, Shanghai, China

Abstract

Purpose – Two and one half-dimensional (2.5D) grid maps are useful for navigation in outdoor environment or on non-flat surface. However, little attention has been given to how to find an optimal path in a 2.5D grid map. The purpose of this paper is to develop a path-planning method in a 2.5D grid map, which aims to provide an efficient solution to robot path planning no matter whether the robot is equipped with the prior knowledge of the environment.

Design/methodology/approach – A 2.5D grid representation is proposed to model non-flat surface for mobile robots. According to the graph extracted from the 2.5D grid map, an improved searching approach derived from A* algorithm is presented for the shortest path planning. With reasonable assumption, the approach is improved for the path planning in unknown environment.

Findings – It is confirmed by experiments that the proposed planning approach provide a solution to the problem of optimal path planning in 2.5 grid maps. Furthermore, the experiment results demonstrate that our 2.5D D* method leads to more efficient dynamic path planning for navigation in unknown environment.

Originality/value – This paper proposes a path-planning approach in a 2.5D grid map which is used to represent a non-flat surface. The approach is capable of efficient navigation no matter whether the global environmental information is available at the beginning of exploration.

Keywords Robots, Motion, Navigation

Paper type Research paper

1. Introduction

An important task for mobile robots is autonomous navigation, where the robot travels between a starting point and a target point without the need for human intervention (Latombe, 1991). A path between the starting and target points that avoids collisions with obstacles is said to be feasible (Ashlock *et al.*, 2006). Thus, robot navigation methods need to solve the path-planning problem, which is to generate a feasible path and optimize this path with respect to certain criteria. There are many studies on robot path planning using various approaches. Latombe (1991) classified robot planning methods appeared in the literature into three main categories: roadmap (Kavraki *et al.*, 1995) or skeleton methods (Takahashi and Schilling, 1989), cell decomposition methods (Kambhampati and Davis, 1986; Hou and Zheng, 1994) and potential field methods (Chuang and Ahuja, 1998; Valavanis *et al.*, 2000). The skeleton methods are mostly used for two-dimensional (2D) path planning and are limited to simple configuration spaces. The methods based on potential fields are fast but the moving object can be trapped in local minima (Keymeulen and Decuyper, 1996). In the cell decomposition approaches, the configuration space is decomposed into a set of uniform or non-uniform cells, for which they are also called as grid-based approaches. Sophisticated A* and D* algorithm family provides promising solution to the shortest path search in a grid-based map. In the past decades, intelligent algorithms such as fuzzy logic, neural network and genetic algorithm have been employed for path planning in grid-based map. Ji *et al.* (1999) adopted fuzzy logic to realize 3D local path planning for an AUV. Yang and Meng (2001) use neural network approaches to generate collision-free path in a non-stationary environment. For its good search performance in large and complex space, genetic algorithm (Randria *et al.*, 2007) is also used for path generation in grid-based maps.

However, most research works assume that the robots move on flat surface which can be simply denoted by a 2D grid-based map. For navigation tasks on outdoor or non-flat surface, Seraji (1999) and Ye (2007) evolved a new 2D grid map with traversability indices. Taking account of terrain characters such as slope and roughness, the planner generates an optimal path with minimum energy consummation or shortest distance. Carsten *et al.* (2006) extend the conventional 2D map to a three-dimensional (3D) grid representation and proposes 3D field D* algorithm to generate the shortest path for submarine and aerial robots. Zhang *et al.* (2008) employ genetic algorithm to solve path-planning problem in 3D grid environment. However, full 3D models usually have high computational demands that prevent them from being directly applicable in large-scale environments (Triebel *et al.*, 2006). To overcome this problem, Fong *et al.* (2003) proposed a two and one

The current issue and full text archive of this journal is available at **www.emeraldinsight.com/0143-991X.htm**

Industrial Robot: An International Journal
38/3 (2011) 315–321

[DOI 10.1108/01439911111122815]

half-dimensional (2 1/2D or 2.5D) grid map and implemented path search based on TRULLA algorithm. Gutmann *et al.* (2005) adopted 2.5D grid map for environmental modeling and navigation in indoor environment containing stairs for biped robots. Also, we presented our study on 2.5D grid map which is capable of rapid traversability assessment for mobile robots traversing on rough terrain (Gu *et al.*, 2008).

In the original works introduced above, we can see that little attention has been received for the shortest path-planning problem in 2.5D grid maps. This paper is focused on finding a feasible path which is also as short as possible within the 2.5D grid set which consists of all passable grids in the map. Traversability assessment is employed before path planning to determine whether a 2.5D grid is passable. The method is introduced in our previous work (Gu *et al.*, 2008) and will not be presented in this paper. After identifying the passable grids, we propose 2.5D A* algorithm derived from sophisticated A* for path planning in known maps. In such cases, robot is equipped with full knowledge of the map before carrying planning. Thereafter, heuristic functions and criteria for replanning are improved for efficient dynamic planning in partially known or unknown maps.

The remainder of the paper is organized as follows. In Section 2, we will brief the 2.5D grid representation which is first proposed in our previous work (Gu *et al.*, 2008). The improved planner for path searching in known 2.5D grid map is introduced in Section 3. For dynamic path planning in 2.5D grid maps without prior knowledge of the environment, cost functions and replanning criteria are described in Section 4. In Section 5, simulations and experiments are studied to examine the performance of the proposed algorithms in different maps. Finally, a conclusion is presented in Section 6.

2. 2.5D grid-based map

Generally, when navigating outdoors or on non-flat surface autonomously, robot uses diverse sensors (e.g. laser range finder (LRF) and stereo vision sensors) to acquire the 3D profile of environment. The key idea underlying the 2.5D grid maps is to store the height information of the surface in a 2D grid map. The result from either stereo vision or LRF is a set of discrete 3D points, each of which can be denoted as $\boldsymbol{p} = [x \quad y \quad z]^T$ (where z is the height value). According to the x- and y-value, we can locate the corresponding grid for every point p. For certain grid s_{ij}, the 3D points located in the same grid make up of point set $P_{ij} = \{p | x_{(i,j)} \le x < x_{(i+1,j)}, y_{(i,j)} \le y < y_{(i,j+1)}\}$, where $x_{(i,j)}$, $y_{(i,j)}$, $x_{(i+1,j)}$ and $y_{(i,j+1)}$ define the horizontal range of s_{ij}. Let n denote the number of points in P_{ij}. The average height h_{ij} of P_{ij} can be computed as:

$$h_{ij} = \frac{1}{n}\sum_{i=1}^{n} z_i, \quad \text{for } p \in P_{ij}. \tag{1}$$

With the average heights of adjacent grids, we can construct a 2.5D grid map representing the environmental surface as shown in Figure 1(a).

For easy implementation of path-planning approaches, a graph representing the 2.5D grid map can be extracted as shown in Figure 1(b). Each grid can be represented by a node (black circle in Figure 1(b)) which resides at the left-bottom corner of the grid. The nodes are connected with edges (dashed line in Figure 1(b)) for adjacent grid patches. Each node carries a height value which indicates the average height within the grid. The edge between adjacent nodes records the cost value that the robot takes when traveling from one to another. In our study, we use overall path length to evaluate search result. Therefore, the cost $c(s_i, s_j)$ is represented by the Euclidean distance between the two adjacent grids.

3. 2.5D A* algorithm

For 2D grid maps, it has been demonstrated that the planning result of A* accords with that of Dijkstra's algorithm. By knowing each grid occupancy before planning, both A* and Dijkstra's algorithm can generate shortest path between specified starting and target grids. However, the A* algorithm visits much less grids than Dijkstra's by adopting a heuristic function which helps to guide the search towards the target. In a 2.5D grid map, the traveling cost between two adjacent grids is first calculated from the grids' height. Thereafter, Dijkstra's algorithm can get the shortest path between two specified grids. Similar with the planning process in a 2D grid map, the Dijkstra's is slow and low efficient. In order to achieve efficient shortest path planning in a 2.5D grid map, an appropriate heuristic function should be developed first.

For each grid s, the function $g(s)$ indicates the minimum cost to move from the starting grid, while the heuristic function $h(s)$ is an estimated cost of moving from s to the target grid. In our study, $g(s)$ is the sum of edge costs from the starting grid:

$$g(s) = \sum_{i=0}^{n-1} c(s_i, s_{i+1}) \tag{2}$$

where $\{s_0, s_1, \ldots, s_{n-1}, s_n\}$ represents the grid sequence that have been traversed before arriving current grid s. $h(s)$ is the Euclidean distance between s and the target grid. Each grid propagated during the search is assigned a cost function $f(s)$ which is given by:

$$f(s) = g(s) + h(s) \tag{3}$$

The search maintains a priority queue ordered by the cost function $f(s)$. It pops a grid with minimum $f(s)$ in the queue each time as the next grid to go until the search reaches the target grid.

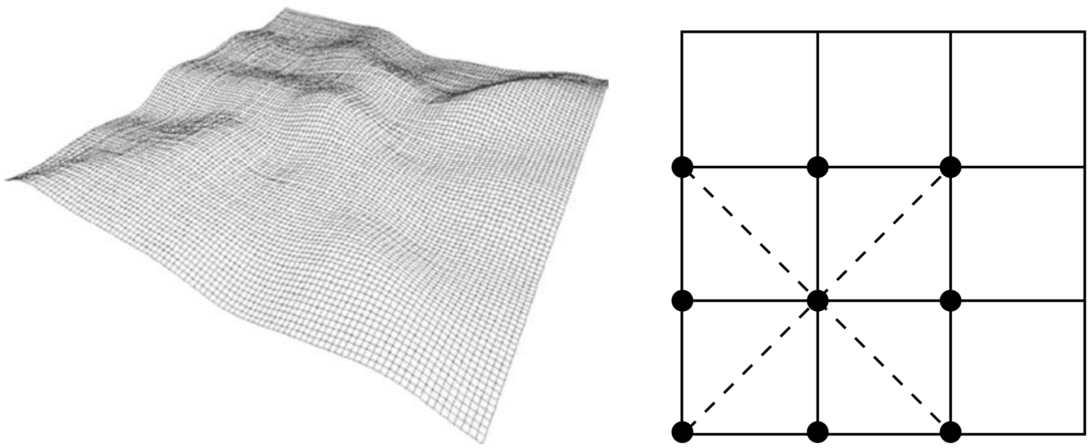
We testify our proposed improved A* algorithm (named as 2.5D A*) and compare its result with that from Dijkstra's. The comparison in Figure 2 shows that the result of our 2.5D A* also accords with that of Dijkstra's. And with heuristic functions, 2.5D A* plans more efficiently by expanding less grids.

4. 2.5D D* algorithm

When traversing in unknown environment, the robot is not able to learn the optimal path from its initial planning because of the lack of the full knowledge of environment. The D* (Stentz, 1995) planning method treats unknown areas as free space at first. When moving along the path, robot observes the environment and updates corresponding grids' status in the 2D map. If the renewed status of a grid conflicts with its assumption, re-planning is employed to generate a new and feasible path between current grid and specified target grid. The process will continue until the robot gets to the target grid or finds out the target grid unreachable. Therefore,

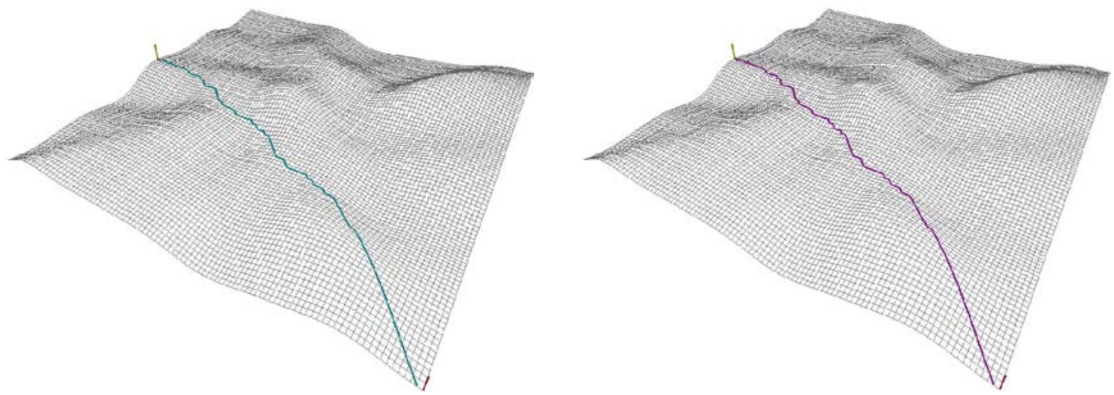
Industrial Robot: An International Journal
Volume 38 · Number 3 · 2011 · 315–321

Figure 1 (a) 2.5D grid representation and (b) graph extracted from 2.5D grid



(a) **(b)**

Figure 2 Path planning results



(a) (b)

Notes: (a) 2.5D A* algorithm; (b) Dijkstra's algorithm

in unknown environment, robot's track between specified starting and target grids consists of path-planning results derived at different positions along the track. In this case, path planning is aimed at generating shortest robot track during reasonable planning times. For a 2.5D grid map, a planning method originated from D* algorithm is proposed, named as 2.5D D* algorithm. We will introduce the cost functions of 2.5D D* algorithm, as well as the criteria for replanning.

A. Cost functions

In a 2.5D grid map, each grid contains a height value instead of a status (obstacle or free space). The cost functions in aforementioned 2.5D A* algorithm, $g(s)$ and $h(s)$, are not available without prior knowledge of the environment before planning. During robot's movement in unknown environment, the grid height is first updated by deployed sensor. Impassable grids identified by traversability assessment (Gu *et al.*, 2008) will not be involved in path planning. In other words, 2.5D D* algorithm will only propagate passable grids or those that have not been observed yet as path candidate. According to whether the grid is observed, the edge cost between two adjacent grids is estimated as follows:

$$c(s_i,s_j)=\begin{cases}\sqrt{(x_i-x_j)^2+(y_i-y_j)^2+(h_i-h_j)^2}, & \textit{if } s_i,s_j \textit{ known}\\ \sqrt{(x_i-x_j)^2+(y_i-y_j)^2}, & \textit{otherwise}\end{cases} \tag{4}$$

Still, $g(s)$ is denoted by the sum of edge costs from starting grid s_s. In the following content, diverse heuristic functions, $h(s)$, are built according to different assumptions of the target grid s_t. The results are studied in the following sections to distinguish the appropriate planning scheme in unknown 2.5D grid maps.

Assumption 1

The heuristic function is constructed WITHOUT grid heights. In this case, the heuristic function of grid s is estimated as:

$$h(s)=\sqrt{(x-x_t)^2+(y-y_t)^2} \tag{5}$$

This estimate is similar with that used for planning in 2D grid map. It provides quick planning speed by using the horizontal distance instead and also helps to guide the search towards the target grid. Obviously, path planning based on such heuristic functions cannot generate shortest trajectories because it neglects the grid heights during planning. Even given the global knowledge of the environment, the robot is barely able to shorten its trajectory.

Assumption 2

The height of the target grid is available at the beginning and the heuristic function is constructed WITH grid heights. In some practical robot applications, the surroundings around specified target position will be perceived before navigation. In such cases the height of target grid is learnt before the first planning, which makes it possible to build a heuristic function similar with that in the aforementioned 2.5D A*. However, the function $h(s)$ depends on whether the height of grid s is available. If s has been observed by robot already, $h(s)$ is represented by the Euclidean distance between s and target grid s_t. On the contrary, the height of s is denoted by the height of starting grid h_s. As a conclusion, the $h(s)$ is estimated as:

$$h(s) = \begin{cases} \sqrt{(x-x_t)^2 + (y-y_t) + (h-h_t)}, & \text{if } s \text{ known} \\ \sqrt{(x-x_t)^2 + (y-y_t) + (h_s-h_t)}, & \text{otherwise} \end{cases} \tag{6}$$

Assumption 3

The height of the target grid is NOT available at the beginning and the heuristic function is constructed WITH grid heights. During most navigation tasks, the robot is not always equipped with target information before the navigation starts. In planning in unknown 2D grid maps, the planner cannot tell whether the target grid is obstacle or free space until it goes close enough to see. Nevertheless, it assumes the target grid as free space. For the 2.5D D* algorithm, the Euclidean distance between s and s_t is employed to indicate the heuristic function, $h(s)$. The height of target grid, in this case, is presumed as zero until it is observed by the robot. Besides, the height of s is denoted by h_s until s is observed. Therefore, the heuristic function is built as:

$$h(s) = \begin{cases} \sqrt{(x-x_t)^2+(y-y_t)^2+(h-h_t)^2}, & \text{if } s, s_t \text{ known} \\ \sqrt{(x-x_t)^2+(y-y_t)^2+(h_s-h_t)^2}, & \text{if } s \text{ unknown}, s_t \text{ known} \\ \sqrt{(x-x_t)^2+(y-y_t)^2+h^2}, & \text{if } s \text{ known}, s_t \text{ unknown} \\ \sqrt{(x-x_t)^2+(y-y_t)^2+h_s^2}, & \text{otherwise} \end{cases} \tag{7}$$

By adopting the heuristic function according to assumption 2 or 3, the robot takes account of grid height aiming at generating a trajectory as short as possible. The performance of 2.5D D* algorithms based on the two heuristic functions are introduced in Section 5.

B. Re-planning criterions

When traversing in a 2D grid map, the robot employs re-plan when the grid in its way, which is assumed as free space, turns out to be obstacle. The D* Lite algorithm (Koenig and Likhachev, 2002) invokes re-plan when the cost of grid is inconsistent with its prediction. During the navigation in unknown environment, however, the cost of grid in 2.5D grid map will frequently conflicts with its prediction if we use Koenig and Likhachev (2002) re-planning criterion. In order to enhance the efficiency of re-plan, we adopt local minima as one of the re-planning criterions. During exploration, robot analyzes its neighboring eight grids and finds the grid s with lowest $f(s)$ cost. If the grid with minimum $f(s)$ cost accords with the next grid to go provided by the planner, the robot will continue moving along the current path. Otherwise, the planner will determine whether a re-plan is needed.

However, it is widely accepted that a robot cannot be efficiently guided by local minima. The robot may employ redundant re-plans and, in some cases, even be trapped in some region. The second re-planning criterion is that the height of the grid that disagrees with the first criterion is out of planner's knowledge before the path is given. For example, when the planner indicates that grid s is the next grid to go and the height of s has been perceived before the planning, the robot will go to s no matter whether s has the minimum $f(s)$ cost among the eight neighboring grid of robot's current position. Otherwise, a re-plan is employed if $f(s)$ is not minimum and a new path is generated between the current and target grid.

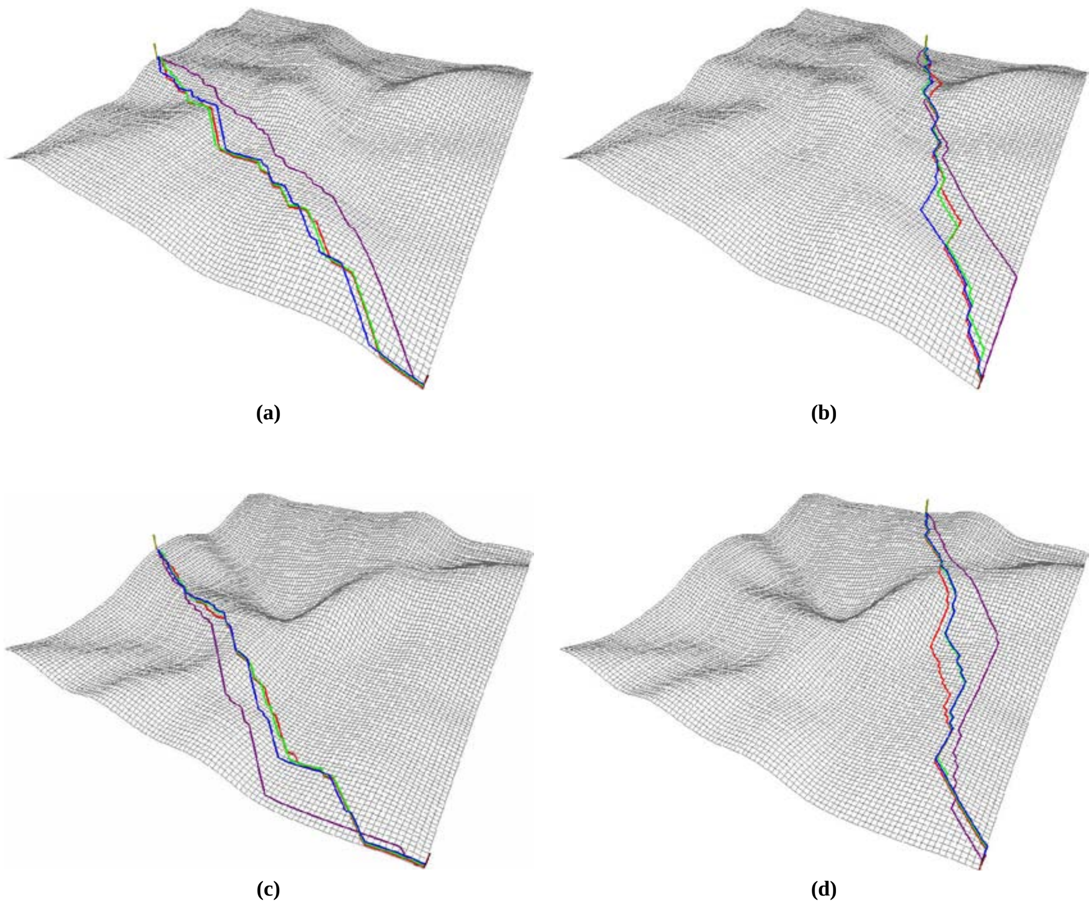
5. Experiments and results

In this section, we present some experimental results. Our proposed 2.5D D* algorithm is testified in a 2.5D grid map which consists of 100×100 grids. Each grid denotes the height within an area of 10×10 cm. The grid height is stochastically generated by 1,000 iterations of Fault Formation (Polack, 2002). Since the robot is not equipped with prior knowledge of the environment at the beginning, it has to perceive the environmental information during its operation. First, we implement our aforementioned 2.5D D* algorithm according to different assumptions and run the simulated navigation in pre-generated 2.5D grid maps. According to simulation results, cost function with shortest trajectory is employed to carry out an experiment on dynamic path planning.

Three different 2.5D D* algorithms derived from the three assumptions, respectively, are employed separately to drive the robot from a same starting grid to a specified target grid. Four experiments are done for each 2.5D D* algorithm in two different 2.5D grid maps (T1 and T2). In each grid map, additionally, the robot moves towards two different target grids (G1 and G2) in two separated experiments. The trajectory generated by each planner in four experiments is shown in Figure 3, with the red, green and blue line representing the result of 2.5D D* algorithm according to assumptions 1, 2 and 3, respectively. Besides, the purple line in Figure 3 indicates the shortest path between the starting and target grids in the map. However, different from the other three tracks, the shortest path is generated by Dijkstra's or 2.5D A* algorithm with the assumption that all grids' height are known before planning. It is used to gauge the difference between trajectories produced by dynamic planning and optimum planning result in a map.

As illustrated in Table I, the robot will traverse a longer distance if the path planner does not take account of the grid heights during planning. In addition, assumptions 2 and 3 output close trajectories. However, assumption 2 requires the height of target grid be given before initial planning while assumption 3 can work in a totally unknown map. For the

Figure 3 Trajectories produced by 2.5D D* path planners



Notes: (a) Move towards G1 in T1 (depicted as T1/G1); (b) move towards G2 in T1; (c) move towards G1 in T2; (d) move towards G2 in T2

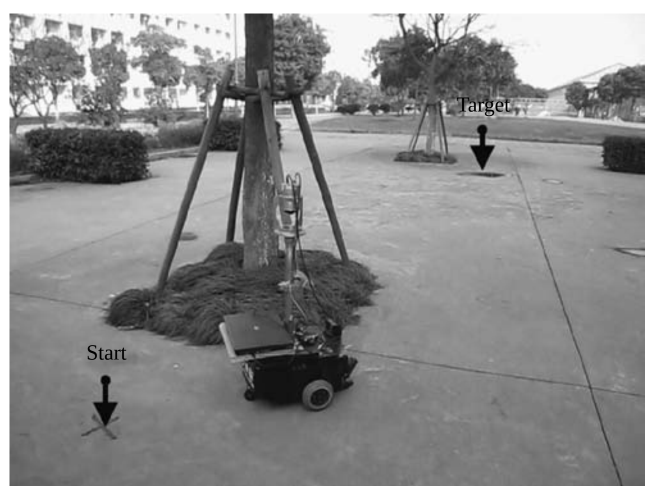
Table I Trajectory length from starting grid to target grid

	T1/G1 (cm)	T1/G2 (cm)	T2/G1 (cm)	T2/G2 (cm)
Dijkstra's	2,005.3	1,988.5	2,634.8	2,666.3
Assumption 1	2,663.9	2,517.3	3,092.9	3,133.0
Assumption 2	2,593.7	2,402.6	3,062.6	2,991.1
Assumption 3	2,551.9	2,439.2	2,937.4	2,990.3

sake of autonomy, 2.5D D* algorithm according to assumption 3 provides a generic and feasible solution to problem of path planning in unknown environment.

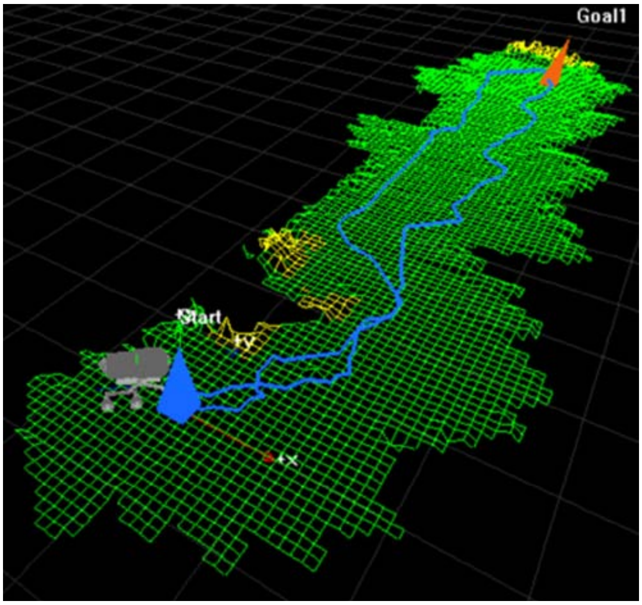
In order to validate the performance of the 2.5D D* algorithm, we carried out an autonomous navigation experiment. The 2.5D D* algorithm is developed according to assumption 3. The experiment setup is shown in Figure 4. A differential-drive robot is employed as mobile platform. A commercial stereo vision solution (Bumblebee2) is adopted to reconstruct environment in front of robot during its execution. The robot has no knowledge of the environment before the initial planning. It has to update the 2.5D grid map for the environment through deployed sensor in the process of navigation. The 2.5D D* algorithm helps to improve the efficiency of navigation by guiding the robot towards the specified target. The robot will move back to the start position after it arrives at the target. The results of the experiment are shown in Figure 5.

Figure 4 An experiment on dynamic path planning



As shown in Figure 5, most regions (black region in the figure) in the environment remain still unknown after the experiment.

Figure 5 Results of the experiment on dynamic path planning



By adopting 2.5D D* algorithm, only grids necessary for determining the path are perceived and updated in the 2.5D grid map. The blue curve in Figure 5 is the trajectory of robot after returning the start position. If we neglect the wriggle caused by robot execution, the robot trajectory is close to the shortest between start and target position, which demonstrates the advantage of 2.5D D* algorithm.

6. Summary

In this paper, we extend the sophisticated A* algorithm and employ the improved searching approaches for robot path planning in proposed 2.5D grid maps. For the navigation in known 2.5D maps, 2.5D A* algorithm is capable of generating the shortest path between specified starting and target grids. Meanwhile, it involves much less grids in path propagation which greatly reduces time consumed for path planning. Therefore, 2.5D A* algorithm provides an efficient solution to path planning in known environment. For robot navigation with no prior knowledge, we present 2.5D D* algorithm and compare the planning results from three variations of heuristic cost function. It is demonstrated by simulation and experiment results that the 2.5D D* algorithm is capable of efficient navigation in a 2.5D grid map even lacking exact target information. Guided by 2.5D D* algorithm, robot can generate a collision-free path with reasonable length. In our study, however, we notice that the path generated by our 2.5D D* approach is suboptimal and not smooth enough for execution because of the heading limitation in an eight-connected grid representation. In the future, we will continue to improve our planning approach and work on the integration of terrain traversability and shortest path search for more efficient and safe exploration.

References

Ashlock, D.A., Manikas, T.W. and Ashenayi, K. (2006), "Evolving a diverse collection of robot path planning problems", *Proceedings of the 2006 IEEE Congress on Evolutionary Computation, Vancouver, Canada, July*, pp. 1837-44.

Carsten, J., Ferguson, D. and Stentz, A. (2006), "3D field D*: improved path planning and replanning in three dimensions", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China*, pp. 3381-6.

Chuang, J. and Ahuja, N. (1998), "An analytically tractable potential field model of free space and its application in obstacle avoidance", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 28 No. 5, pp. 729-36.

Fong, E.H.L., Adams, W., Crabbe, F.L. and Schultz, A.C. (2003), "Representing a 3-D environment with a 2 1/2-D map structure", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, October*, pp. 2986-91.

Gutmann, J.S., Fukuchi, M. and Fujita, M. (2005), "A floor and obstacle height map for 3D navigation of humanoid robot", *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain*, pp. 1066-71.

Gu, J.J., Cao, Q.X. and Huang, Y. (2008), "Rapid traversibility assessment in 2.5D grid-based map on rough terrain", *International Journal of Advanced Robotic Systems*, Vol. 5 No. 4, pp. 389-94.

Hou, E. and Zheng, D. (1994), "Mobile robot path planning based on hierarching hexagonal decomposition and artificial potential fields", *Journal of Robotic Systems*, Vol. 11 No. 7, pp. 605-14.

Ji, J.J., Li, B.Q. and Cui, W.Y. (1999), "Three-dimensional local path planner for AUV", *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, Edmonton, Canada*.

Kambhampati, S. and Davis, L.S. (1986), "Multi-resolution path planning for mobile robots", *IEEE Journal of Robotics and Automation*, Vol. RA-2 No. 3, pp. 135-45.

Kavraki, L., Latombe, J.C., Motwani, R. and Raghavan, P. (1995), "Randomized query processing in robot path planning", *Proceedings of the 27th Annual ACM Symposium on Theory of Computing, Las Vegas, NV, USA*, pp. 353-62.

Keymeulen, D. and Decuyper, J. (1996), "Self-organizing system for the motion planning of mobile robots", *Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA*, pp. 3369-74.

Koenig, S. and Likhachev, M. (2002), "D* Lite", *Proceedings of the AAAI Conference of Artificial Intelligence (AAAI), Edmonton, Canada*, pp. 476-83.

Latombe, J.C. (1991), *Robot Motion Planning*, Kluwer Academic, Boston, MA.

Polack, T. (2002), *Focus on 3D Terrain Programming (Game Development)*, Course Technology, PTR, Boston, MA.

Randria, I., Khelifa, M.M.B., Bouchouicha, M. and Abellard, P. (2007), "A comparative study of six basic approaches for path planning towards an autonomous navigation", *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei, Taiwan*.

Seraji, H. (1999), "Traversability index: a new concept for planetary rovers", *Proceedings of 1999 IEEE International Conference on Robotics and Automation, Detroit, Michigan, May*, Vol. 3, pp. 2006-13.

Stentz, A. (1995), "The focussed D* algorithm for real-time replanning", *Proceedings of the International Joint Conference on Artificial Intelligence, Montréal, Canada, August*.

Takahashi, O. and Schilling, R.J. (1989), "Motion planning in a plane using generalized voronoi diagrams", *IEEE Transactions on Robotics and Automation*, Vol. 5 No. 2, pp. 143-50.

Triebel, R., Pfaff, P. and Burgard, W. (2006), "Multi-level surface maps for outdoor terrain mapping and loop closing", *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, October*, pp. 2276-82.

Valavanis, K.P., Hebert, T., Kolluru, R. and Tsourveloudis, N. (2000), "Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field", *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, Vol. 30 No. 2, pp. 187-96.

Yang, S.X. and Meng, M. (2001), "Neural network approaches to dynamic collision-free trajectory generation", *IEEE Transactions on Systems, Man and Cybernetic – Part B: Cybernetics*, Vol. 31 No. 3, pp. 302-18.

Ye, C. (2007), "Navigating a mobile robot by a traversability field histogram", *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, Vol. 37 No. 2, pp. 361-72.

Zhang, H., Liu, M.L., Liu, R. and Hu, T.L. (2008), "Path planning of robot in three-dimensional grid environment based on genetic algorithm", *Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China*, pp. 1010-4.

Further reading

Samet, H. (1988), "An overview of quadtrees, octrees, and related hierarchical data structures", NATO ASI Series, Vol. F40.

Corresponding author

Jiajun Gu can be contacted at: carl.gu@yahoo.com