Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Mobile Robot Path-planning Approach under Unknown Environments

Zi-Xing CAI *, Zhi-Qiang WEN **, Xiao-bing ZOU ***, Bai-fan CHEN****

School of Information Science & Engineering, Central South University, Changsha, China 410083
(e-mail: zxcai@mail.csu.edu.cn, ** zhqwen20001@163.com,*** csutzxb@263.net, ****chenbaifan@21cn.com)*

Abstract: An inverse D* algorithm is presented in this paper for path-planning under unknown environments. In this inverse D* algorithm, the local potential energy around the current position is created firstly by defining the robot distance, then the leave point is searched to be regarded as the local goal position satisfying requirement of the rolling optimization. The leave point is searched locally and iteratively until the robot reaches the goal finally. The experimental results show the validity of our algorithm.

1. INTRODUCTION

Under unknown environments, a robot does not know any information of environments before it enters into a region, but the environment information can be obtained by the sensors installed on robot. It is to say that the mobile robot can obtain the environment information and can transform this information into the knowledge of itself to instruct and optimize its behaviour. Only if the robot has possessed the ability of using the knowledge accumulated before, it will show its aptitude behaviour. Under unknown environments, the navigation of mobile robot includes the bottom-up cognizing process and the top-down process using the accumulated knowledge to instruct and optimize robot's behaviour (Cai Z, Zhou X, Li M, et al, 2000).

Under unknown environments, real-time robot control is in contradiction with large computational cost which is introduced by environment modelling and global path-planning. This contradiction must be compromised validly, or robot will lose its intelligent behaviour. In fact, Global map is unfeasible for autonomous robot navigation in large scale environments because the environment modelling always lags behind the environment changes. Regional planning is a deliberative behaviour exiting between global planning and behaviour control, and instructs robot move to a local goal according to the accumulated environment knowledge. For a small work region, the region planning is similar to global path-planning.

There were many researches fruit aiming at the mobile robot path-planning under the global or local environments (Fan H., 2003, Zou X., Cai Z., 2003). D* algorithm is applied in many robot systems(Koenig S, Likhachev M,2002, Stentz A.,1995) for it can adapt to complex environments. For example D* algorithm is applied in Bullwinkle the America Mars explorer (Singh S, et al, 2000), Unmanned Ground Vehicle System (Stentz A., Hebert M., 1995) and tactics mobile robot under city environments developed by CMU (Matthies L., Xiong Y., Hogg R, et al, 2000). D* (or D Lite) algorithm is proposed by Anthony Stentz, a professor of Carnegie Mellon university in 1993 (Stentz, A., 1993). It is also named dynamic A* algorithm. It makes A* algorithm can adapt to the dynamic obstacle environments. In D* algorithm, A* algorithm is used to create the potential field of goal distance in off-line way at the first time. After finding the dynamic obstacle in the process of searching path towards the direction along which the goal distance will reduce, D* algorithm will search a new path from current position to potential field position which is less than goal distance of the path exiting obstacles. The robot will avoid the dynamic obstacle along this new path. After the robot reaches the potential field area of goal distance that has been created before, it resumes searching path towards the gradient direction in which goal distance reduces. The disadvantage of D* algorithm is that it is only fit for known or partly known environments. When a robot meets the unknown obstacle, the created heuristic potential field distributions will change and the potential field distributions must resume to be computed. Under unknown environments, a robot usually only obtain the local environment information around itself and does not know the environment information around final goal. This will lead to a large discrimination between the potential field model created formerly and the actual model. In addition, D* algorithm is only valid in small area, but in large area, its computation costs is large and the memory space it need also is large (Stentz A., 1995).

Under unknown environments, a robot can only obtain the local environment information and do not know the environment information around final goal. In this case, the method to create the potential field of goal distance in off-line way does not work, in other words, D* algorithm is not fit for path-planning on mobile robots under unknown environments. So we propose an inverse D* algorithm for robot path-planning under unknown environments in this paper. In this inverse D* algorithm, the initial potential field of goal distance do not need to be computed, but the local potential fields around the current position of robot is created by defining the robot distance. Then the local goal position is searched for the minimum estimation satisfying the requirement of the rolling optimization. The algorithm in this paper is named inverse D* algorithm because the search direction of this algorithm is reverse against that in D* algorithm.

2. INVERSE D* ALGORITHM

In inverse D* algorithm, the local potential energy around the current position is computed by defining the robot distance and the leave point is searched to be regarded as the local goal position for satisfying the rolling optimization. The leave point is searched locally and iteratively until robot reaches the goal finally. The leave point is similar to the local goal point. The basic process of inverse D* algorithm is as follows.

1) At first, the grid map is created according to current all known environment, then the obstacle grid is expanded beforehand. The radius of expanded obstacle grid is set according to the robot's width and safety distance against obstacle.

2) Supposing E_{min} is the minimal potential energy of goal from the initial point $S(x_s, y_s)$ to goal point $G(x_g, y_g)$, the goal potential energy function of the free grid point i in coordinate (x_i, y_i) can be expressed as follows.

$$E(i) = \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2} \qquad (1)$$

In the process of rolling optimization, the grid point where the goal potential energy is less then $E_{min} - \varepsilon$ ($\varepsilon > 0$) is searched towards outer grid points regarded current position $R(x_r, y_r)$ as the centre of a circle. The result grid point is named as the leave point of current potential energy trap region denoted by $L_j(x_j, y_j)$, ε is constant and is set to ensure the goal potential energy $E(i)$ is less then the potential energy E_{min}. $L_j(x_j, y_j)$ is the middle goal which is close to final goal step by step. expression (1) is regarded as the value cost function, namely:

$$E(L_j) = \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2} < E_{\min} - \varepsilon \qquad (2)$$

3) In the process of finding $L_j(x_j, y_j)$, the value cost function of search priority is decided by A* algorithm. If the searched point of coordinate (x_i, y_i) is i, the value cost function is $f(i)=g(i)+h(i)$. Differing from A* algorithm, in the inverse D* algorithm $g(i)$ is the actual cost from $R(x_r, y_r)$ to searched point i, named robot distance. $h(i)$ is the estimated cost from searched point i to goal point computed by the Euclidean distance between point i and goal point. The Euclidean distance between point i and $G(x_g, y_g)$ also can be regarded as potential energy function, namely $h(i)= E(i)$.

4) In search area, $g(i)$is marked in grid map. After finding the $L_j(x_j, y_j)$ according to (2), this algorithm can remount to current position setting out from $L_j(x_j, y_j)$ along the direction that gradient of $g(i)$ reduces. In a word, the inverse process of backdating is the path from $R(x_r, y_r)$ to $L_j(x_j, y_j)$.

5) Setting out from $S(x_s, y_s)$, A* algorithm is uninterruptedly used to find $L_j(x_j, y_j)$ until find the final goal point. In this way, the rolling dynamic search is realized.

3 PERFORMANCE ANALYSES

The inverse D* algorithm is developed based on A* algorithm and D* algorithm. In this section, inverse D* algorithm, A* algorithm and D* algorithm is discussed by experimental comparisons. Fig1 and Fig2 respectively shows the experimental results of A* algorithm and D* algorithm on robot path-planning. Under the same environments, the experimental result of inverse D* algorithm is shown in Fig3. In Fig3, time L_1 is firstly found as the middle point after 9 steps from initial point, then in second time, L_2 is found as the second middle point. Thirdly the third point is found as the third middle point, and so on.

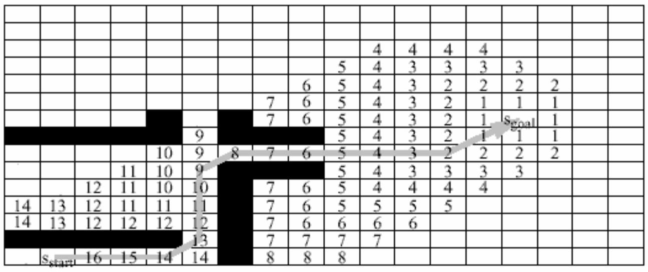

Fig1. Path-planning experimental results based on A* algorithm

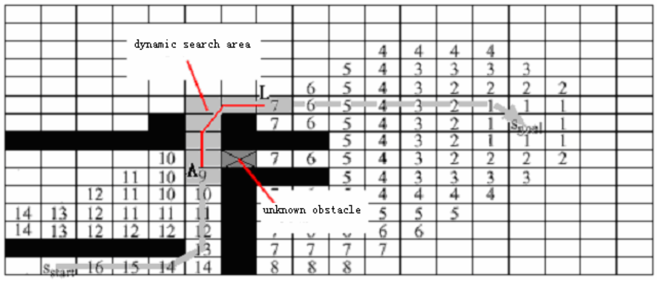

Fig2. Path-planning experimental results based on D* algorithm

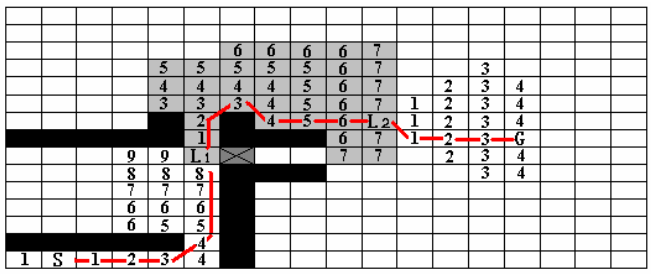

Fig3. Path-planning experimental results based on inverse D* algorithm

In many cases, the search area in A* algorithm is less then blind search methods such as Dijstra algorithm, so the computational cost reduce greatly. From Fig2 and Fig3, the total accumulative search area in inverse D* algorithm is close to search area of initial goal distance potential field in D* algorithm, moreover both computational cost are equivalent under known environments. But under unknown environments, for the environment information is updated in time by detecting the local environment information around current position in inverse D* algorithm, the real-time environment information is more credible than the information of goal area far away from current position. In addition, the probability to meet the uncertain obstacle in search process also is less.

Table1. time cost of D* algorithm(sec)

Method	State space 100×100		State space 350×350		State space 1000×1000	
	Off-line	On-line	Off-line	On-line	Off-line	On-line
Focussed D* (completely initial)	1.85	1.09	19.75	9.53	224.62	10.01
Focussed D* (Locally initial)	0.16	1.7	0.68	18.2	9.53	41.72
Basic D*	1.02	1.31	12.55	16.94	129.08	21.47

Table2. time cost of inverse D* algorithm(sec)

Steps	L=100	L=150	L=250	L=350	L=500	L=800
inverse D* algorithm	0.42	0.82	1.78	3.10	7.65	16.58

In inverse D* algorithm, the middle point is searched dynamically around robot. In our experiment, the data from Stentz (Stentz A., 1995) is used and the time costs of three kind of D* algorithm also from Stentz, which is used to initialize potential field in off-line way and in on-line way, are shown in table1. The time cost of inverse D* algorithm, which run from 100 step to 800 step, is shown in table2. Space complexity of inverse D* algorithm in the worst case is $O(L^2)$, where L is the length of search path. Compared with time cost of the 3 kind of D* algorithm in table1, in inverse D* algorithm, the initial global distance potential field need not be computed and can avoid the off-line computation cost and can improve the computation efficiency of robot under unknown environments and can reduce spending on memory in the process of finding new path under large scale environments.

4 EXPERIMENTS

4.1 Simulation experiment

Compared to D* algorithm, inverse D* algorithm is a rolling optimization algorithm. Without process of creating initial potential field, inverse D* algorithm can be applied to navigation of robot under larger scale environments. Meanwhile, this algorithm can be applied to robot path-planning combined with reactive obstacle avoidance (Zheng M., Cai Z., Yu J., 2006). Under simple environments, the inverse D* algorithm is applied to search the leave point against local potential energy trap. In heuristic search, initial search area is supposed to be the area where value cost function $f(i)$ satisfies $f(i)=g(i)+h(i)\leq a$, where a is the distance that is longer than the distance between robot and goal. In inverse D* algorithm, $a=1.5E_n$, where E_n is the distance between robot and goal. The maximal search area is an ellipse with long axes $1.5E_n$ and short axes $\sqrt{5}/2\,E_n$. If this algorithm does not find the middle goal in this region, it will expend the search area along the boundary of this ellipse until find the middle goal. Fig4 shows the simulation experiments of inverse D* algorithm. Robot moves from initial point (marked "Start") to goal point (marked "Goal") using the inverse D* algorithm. In our simulation experiments, robot move to T1 from start point, then after detecting the obstacle at position T1, robot use the inverse D* algorithm to find the path to L1. After reaching L1, robot use the inverse D* algorithm to find the path to L2 again, at last it reach final goal. This experiment shows robot navigation can be implemented by the inverse D* algorithm. In Fig4, isoclines of robot distance in each area that robot passes is marked.

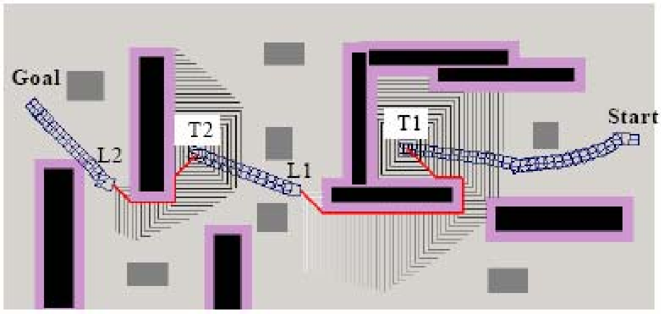

Fig4 simulation experiment based on inverse D* algorithm

4.2 experiment on robot platform

The presented inverse D* algorithm is used in our robot platform shown in Fig5(a). Robot hardware and experiment conditions relative to our experiments can be found in literature (Cai Z., Zou X., Wang L., et al, 2005) and mapping methods can be found in literature (Cai Z., Yu J., Duan Z., et al, 2006). Fig5(b)~(d) show the experimental results in our laboratory and corridor, Where Fig5(b) shows the initial state of robot and its environment robot detects. In initial state, robot does not know any information about the whole environments. Robot detects the local environment information and provides this information for its navigation. A goal is given to the robot in corridor (in Fig5, marked by "Goal"), which is far beyond the detection bound. In process of navigation, the robot uses inverse D* algorithm to find the path to goal point per 10ms, at the same time it moves forwards along this path. Fig5(c) shows the process that robot move from indoor to corridor where there are not dynamic obstacles. Fig5(d) shows the process that the robot return to the given goal where there exits some dynamic obstacles. In Fig5(c),(d), robot can reach the given goals automatically under two kind of environment.

The time cost of inverse D* algorithm is proportion to the search area which is influenced by distribution of obstacle in environment. After precise test in PIII800M, the time cost of inverse D* algorithm for 300 step is 2~3sec and satisfies the requirement of real-time path planning.

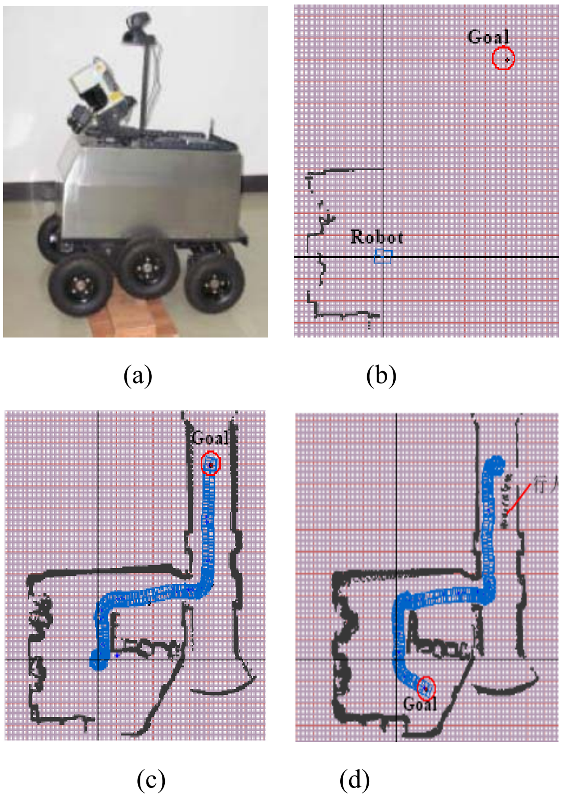

Fig5 results of experiment on inverse D* algorithm, where (a) shows the mobile robot platform, (b) shows the initial state of robot, (c) shows the navigation experimental result on static environments, (d) shows navigation experimental result on dynamic environments.

5 CONCLUSIONS

An inverse D* algorithm is presented in this paper. This algorithm has many advantages: Compared to D* algorithm, the global information is not required in inverse D* algorithm which adapt to navigation under the unknown environment. Robot can avoid the dynamic obstacles in time by combining inverse D* algorithm and reactive obstacle avoidance method (Zheng M., Cai Z., Yu J., 2006). But as the same of artificial potential field, robot be likely to fall in local trap for lacking of full information in our presented inverse D* algorithm. We hope our work on path-planning can contribute to development of intelligent robot technology.

6 ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant No.60234030 and Basic Research Project of China under Grant No.A1420060159.

REFERENCES

Cai Z, Zhou X, Li M, et al(2000). *Evolutionary Control Architecture for Autonomous Mobile Robot based on Function/Behavior Integration*, Robot, **Vol.22, No.3**, pp.169-175. in Chinese.

Fan H(2003), *The Potential-Field Based Collision-Free-Path-Planning Approach for Point Robot*, Bulletin of Science and Technology, **Vol.19, No.4**, pp. 285-287.in Chinese.

Zou X, Cai Z(2003). *Non-Smooth Environment Modeling and Global Path Planning for Mobile Robot*. Journal of Central South University of Technology (Science and Technology),**Vol.10, No3**, pp. 248-254.

Koenig S, Likhachev M (2002). *Improved Fast Replanning for Robot Navigation in Unknown Terrain*. In: ICRA2002, pp. 968-975, IEEE Press, Washington.

Stentz A(1995). The Focussed D* Algorithm for Real-Time Replanning. In: Proc. of the 14th Int. Joint Conf. on Artificial Intelligence, pp.1652-1659, Montreal, Canada.

Singh S, Simmons R, Smith T, et al(2000). *Recent Progress in Local and Global Travers ability for Planetary Rovers*. In: ICRA2000. pp.1194-1200, San Francisco.

Stentz A, Hebert M(1995). *A Complete Navigation System for Goal Acquisition in Unknown Environments*. Autonomous Robots, **Vol. 2, No.2**, pp.127-145

Matthies L, Xiong Y, Hogg R, et al(2000). A Portable, Autonomous, Urban Reconnaissance Robot. In: Proc. of the 6th Int. Conf. on Intelligent Autonomous Systems, pp.163-172,Venice.

Stentz, A(1993). *Optimal and Efficient Path Planning for Unknown and Dynamic Environments*, Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-93-20.

Zheng M. ,Cai Z., Yu J(2006), *An obstacle avoidance strategy of mobile robots under dynamic environments*, High Technology Letters, **vol.16, No.8,** pp.813-819. in Chinese.

Cai Z.,Zou X.,Wang L.,et al(2005). *Design of distributed control system for mobile robot*, Journal of Central South University of Technology (Science and Technology), **Vol.36, No.5**, pp. 727-732.

Cai Z. Yu J., Duan Z., ,et al(2006). *Design of the 3D perceptive system based on laser scanner for a mobile robot*, Int. Journal of Computer Science and Network Security, **Vol.6, No.3**, pp.20-29.