An Optimized Algorithm Based on Random Tree and Particle Swarm

Zhou Feng
Shandong Institute of Commerce and Technology, Shang Dong JiNan, 250103, China

Article Info

Article history:

Received Aug 11, 2013
Revised Feb 20, 2014
Accepted Mar 10, 2014

Keyword:

Particle swarm optimizer
Path planning
Rapidly-exploring random tree
Robot

ABSTRACT

A based on Rapidly-exploring Random Tree (RRT) and Particle Swarm Optimizer (PSO) for path planning of the robot is proposed. First the grid method is built to describe the working space of the mobile robot, then the Rapidly-exploring Random Tree algorithm is used to obtain the global navigation path, and the Particle Swarm Optimizer algorithm is adopted to get the better path.Computer experiment results demonstrate that this novel algorithm can plan an optimal path rapidly in a cluttered environment. The successful obstacle avoidance is achieved, and the model is robust and performs reliably.

Corresponding Author:

Zhou Feng,
Shandong Institute of Commerce and Technology, ShangDong JiNan, 250103, China
Email: zhoufengkey@163.com

1. INTRODUCTION

Robot path planning means in the working space, a path is found with the robot starting from a point, rounding barriers and arriving to the destination. Commonly, there are many paths for robot to accomplish the task, but in fact the best path is selected according to some guide line. These guide lines are: shortest path, least energy consuming or shortest time.So, the robot path planning is a constrained optimization problem.Robot path planning has been an active research area, and many methods have been developed to tackle this problem, such as rolling plan [1], RRT methods [2], neural networks approaches [3],GA methods[4] and ACO algorithm [5] so on. These works have made some innovative achievements.but the common shortage is that solving time is too long, efficiency is not high. Although In [6] the algorithm can achieve rapid optimization, but because a robot go dogleg path in free zone, the path may not be optimal. Grid method is one of the commonly used modeling modeling method, but it has a defect: a robot go dogleg path in free zone. The dogleg is suboptimal paths and robot arrival time will increase.

In this paper, A based on Rapidly-exploring Random Tree (RRT) and Particle Swarm Optimizer (PSO) for path planning of the robot is proposed. First the grid method is built to describe the working space of the mobile robot, then the Rapidly-exploring Random Tree algorithm is used to obtain the global navigation path, and the Particle Swarm Optimizer algorithm is adopted to get the better path. The effectiveness and efficiency of the proposed algorithm is demonstrated by simulation studies.

2. DESCRIPTION OF ENVIRONMENT

To ensure that the path is not too close to any of the obstacles, the dimension of the robot is represented by a point, and the boundaries of the obstacles are expanded according to the plus of the maximum distance occupied by robot's cross.Let *AS* be the finite walking area of *Rob* in a two-dimension plane and *AS* is a protruding polygon, in which there are finite numbers of static obstacles b_1, b_2, ……, b_n.

The task is to plan a much shorter path for the robot to walk from g_{begin} to g_{end} collision-freely. And $\sum_0$ is the right-angled coordinate system in AS whose origin is the left and upper corner of AS and its landscape orientation is X-axes, its portrait is Y-axes. The maximal values of x and y are x_{max} and y_{max} , respectively. Then x and y can be divided usingδas an unit, as a result, all grids can be formed one by one (fig.1). The numbers of each row and column are $N_x = x_{max} / R_a$ and $N_y = y_{max} / R_a$, respectively, (if AS is considered as the arbitrary shape, some grid-obstacles can be filled on the boundary of AS to make it square or rectangle), where $b_i\,(i = 1, 2, \cdots, n)$ holds one or more gratings, when it is not a full one, they are considered as the full one.

The mobile robot environment is represented by orderly numbered grids *C*={1,2,3,…， ,*M*}, each of which represents a location in the environment. *g*(*x* ,*y*) also represents a location in the environment,*x* is the row number,*y* is the columns number.

In the grid based robot environment, the number *i* and *g*(*x* ,*y*) denote grid and environment coordinate, respectively,thus

$$x_i=((i-1) \bmod Nx)+1, \quad y_i=(int)((i-1)/Nx)+1 \tag{1}$$

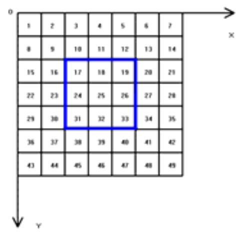

Figure 1. Relationship between grid coordinates and sequence number

3. RRT-PSO

Grid method is simple common modeling methods. But its drawback is that partition size is difficult to control, the grid size is smaller, obstacles are said to be more precise, but at the same time it will take up a lot of storage space; the grid size is too large, the planned path is not accurate. Therefore when planning the precise path, the grid method cannot be used.To resolve the shortage in this paper the Rapidly-exploring Random Tree algorithm is used to obtain the global navigation path,and the Particle Swarm Optimizer algorithm is adopted to adjustment the intersection point of each initial path and the boundary of the grid. Each dimension of each dimension represents a point of intersection. The connection which from low dimension to high dimension will constitute a new path, After several iterations the approximate optimal global path will be finded. The optimization task is to adjust the position of x^d to shorten the length of path and get the optimized (or acceptable) path in the planning space.The adjust process of x^d is shown in Figure 2. Any x^d can slide freely along the free link that it lies on. Path Coding Method new path formed by the slide of x^d on line x^d_{min}、x^d_{max} will not intersect with the obstacles. After processing every x^d, the new path nodes sequence forms a new collision free path for the robot.

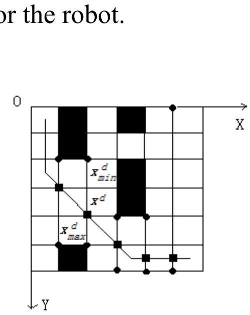

Figure 2. Path Coding Method

For conveniently describe, we make the following definition:
Definition 1. The distance between any two grid cells g_i and g_h (or corresponding points P_i and P_h) is the length of the line between the center points of the two grids and is denoted by d(gi , gh) or d(P_i , P_h). i,h∈C.

$$d(g_i,\ g_h) = \sqrt{(x_i - x_h)^2 + (y_i - y_h)^2} \quad (2)$$

Definition 2. The number of *i* which is free grid make up a set which was called the feasible region, Marked as F. F∪S=A., i⊆F。

Definition 3. For all nodes in the RRT corresponding grid serial number *i* consisting of a collection called RRT node set, Marked as *P*(*RRT*).
Definition 4. If g is a grid cell, the set $NEIB_i$ is called the neighborhood of g_i. The broad lines marked the neighborhood of g(4,4) in fig1.
Definition 5. Collection which is composed by the serial number that the robot is not searched called *Q(RRT)*

3.2 The Basic Idea and Steps of the Algorithm

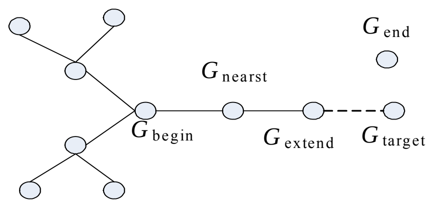

Figure 3. The RRT Construction

The rapidly-exploring random tree planner is an incremental search algorithm that provides benefits over conventional roadmap planners due to the inherent feasibility of the solutions generated.Initially, the G_{begin} is the only node of G_{end}. For each iteration, a random state G_{rand} is chosen, and G_{nearst} is selected as the nearest state to G_{rand} according to a metric function *p*. For G_{nearst} *a* best input G_{best} is chosen to generate a new state G_{extend} which is closest to G_{rand} of all states generated by applying one step control from G_{rand}.If G_{rand} satisfies the global constraints, G_{rand} will be added to $G_{end.}$

Step1: G_{begin} as the start point, G_{end} as the goal point, The grid serial number of G_{begin} as *RRT* root node, initializes relevant parameters.

Step2: Let G_{nearst}= G_{root} ,G_{neast} is the closest node in the *P(RRT)* to the G_{end}.

Step3: if G_{nearst}= G_{end}, then goto Step6;else goto Step4.

Step4: *p* which obeys uniform distribution is a random number ($p \in [0,1]$). If *p* is less than p_g, then let G_{targe} equals G_{end}. If *p* is greater than p_g, let G_{target} is blank grid which is randomly selected individuals from *Q(RRT)*. p_g is known as the constants.

Step5: Find a node Gn($Gn \subset P(RRT)$),and G_n is the closest to the $G_{target.}$ And then find a node G_{extend} ($Gextend \subset NEIB_{Gn}$, $Gextend \notin P(RRT)$)which is the closest to the G_n.If G_{extend} can be find ,it will be added to *P(RRT)*.If d($G_{extend,}G_{end}$) is less than d($G_{nearst,}G_{end}$), then let G_{nears} equals G_{extend}. Otherwise, the extension failed, goto Step3;

Step6: Return to the formation of the RRT, a path from G_{end} to G_{begin} can be made by traversing up the tree until the root node is reached.

Step7: Select the best navigation path $P=\{P_0P_1\ldots\ldots P_{u+1}\}$,P_0 is the start point, P_{u+1} is the goal point.

Step8: Calculated the number(Marked as *Pnum*) of intersection points the navigation path and the vertical boundary of the grid,that is each particle has *Pnum* dimension, Calculate the sliding range of the first d-dimensional from x^d_{min} to x^d_{max}.

Step9: Initialize particle *c* (randomly initialize each dimension of the particle's position $x_c(0)$ and velocity in the solution space $v_c(0)$), each particle's historic optimal position p_c is itself.

Initializes relevant parameters: ω_{max} 、 ω_{min} 、 c_1 、 c_2 P_{max} (the largest number of iterations) n (iteration counter).

Step10: Calculate each article's fitness value according to **Eq. *(3)*** and label the particle with the minimum fitness value as P_g;

$$F_c(x(t)) = d(G_{begin}, x_{c,1}) + \sum_{d=1}^{Pnum-1} d(x_{c,d+1}(t), x_{c,d}(t)) + d(G_{end}, x_{c,Pnum}(t)) \quad (3)$$

Step11: Update particle's velocity according to Eq. (4). if $v_{c,d} < -v_{\max}^{d}$, set $v_{c,d} = -v_{\max}^{d}$, if $v_{c,d} > v_{\max}^{d}$, set $v_{c,d} = v_{\max}^{d}$;r_1 and r_2 are random number between 0 and 1.

$$\begin{aligned} v_{c,d}(t+1) = \omega V_{c,d}(t) + c_1 r_1 [p_{c,d}(t) - x_{c,d}(t)] \\ + c_2 r_2 [p_{g,d}(t) - x_{c,d}(t)] \end{aligned} \quad (4)$$

Update particle's position according to Eq. (5), if $x_{c,d} < x_{\min}^{d}$,set $x_{c,d} = x_{\min}^{d}$. if $x_{c,d} > x_{\max}^{d}$, set $x_{c,d} = x_{\max}^{d}$;

$$x_{c,d}(t+1) = x_{c,d}(t) + v_{c,d}(t+1) \quad (5)$$

Step12: Calculate each article's fitness value ($F_c(x(t+1))$) according to Eq. *(3)*. Update p_c、Update p_g、 n=n+1, $\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times n / p_{\max}$.

Step13: if n<p_{max},turn to step 11 and iterate; if n=p_{max} turn to step 14.

Step14: find the optimal path.

4. SIMULATION RESULT

To investigate the effect of the algorithm proposed in this paper, many simulation experiments were conducted. As will be shown in the next sections, the results were quite satisfactory and compare favorably to results using other algorithms. A Pentium desktop computer with Intel Dual Core CPU, 2.33GHz, and 2GB memory was used. The software is written in MS Visual Studio 6.0 C. In order to results were comparable, setting the same experimental environment

Under the same conditions and with the environment of Figure 4. We compare our results with the results in [7]. The fine line which length is 32.14 stands for the path of ACO. The bold line which length is 30.12 stands for the path of the algorithm in [7]. The red line which length is 29.57 stands for the path of our algorithm. The grid is used as length unit in this drawing.

In complex environments the robot navigation is very successful using our algorithm. We compare our results with the results in [6]. The black line which length is 32.13 stands for the path of the algorithm in [6]. The red line which length is 31.25 stands for the path of our algorithm.

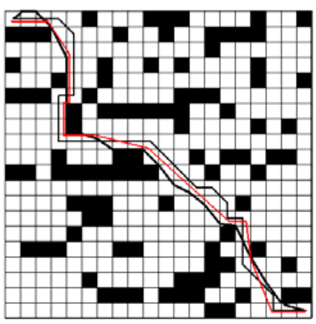

Figure 4. The path in a simple environment

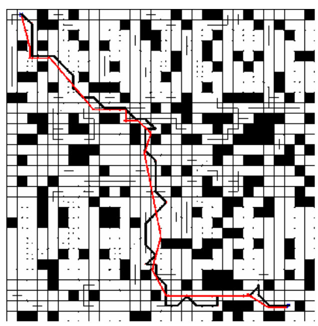

Figure 5. The path in a complex environmen

In order to further validate the algorithm, We compared our algorithm with others in the 30 * 30 environment of reference [8]. In the table 1 are listed the average length of path which obtained in five tests.

Table 1. Performance Contrast

environment	A*	GA	ACO－grid	RRT-grid[7]	reference [9]	our algorithm
30*30 [8]	N/A	85.491	65.0	61.2	45.0	44.5

5. CONCLUSION

In this paper, the authors propose a novel method to solve the global path-planning problem for the mobile robot. First the grid method is built to describe the working space of the mobile robot, then the Rapidly-exploring Random Tree algorithm is used to obtain the global navigation path, and the Particle Swarm Optimizer algorithm is adopted to get the better path. Experiment results show the validity and practicability of the method. By this method, it is easy to build a model and meet the real-time demand for mobile robot's navigation with the simple algorithm, which results in a certain practical value.

REFERENCES

[1] Yanrong Hu and Simon X. Yang. *A Knowledge Based Genetic A1gorithm for Path of a Mobile Robot*. Proceedings of the 2004 IEEE international Conferenceon Robotics & Automation New Orleans. 200: 4350-4355.
[2] GUO Tongying, QU Daokui, DONG Zaili. *Research of Path Planning for Polishing Robot Based on Improved Genetic Algorithm*. Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics: 334-338
[3] Qin Yuan Qing, Sun De Bao. *Path Planning For Mobile Robot Using The Particle Swarm Optimization With Mutation Operator*. Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004: 2473-2478.
[4] Sun Bo, Chen Weidong, Xi Yugeng. Particle Swarm Optimization Based Global Path Planning for Mobile Robots. *Control and Decision*. 2005; 20(9): 1052-1060 (in Chinese)
[5] Zhu Qing Bao. Ant Algorithm for Navigation of Multi-Robot Movement in Unknown Environment. *Journal of Software*. 2006; 17(9): 1890-1898 (in Chinese).
[6] Guo Haitao, Zhu Qingbao, Xu Shoujiang. Rapid-exploring random tree algorithm for path planning of robot based on grid method. *Journal of Nanjing Normal University: Engineering and Technology Edition*. 2007; 7(2): 58-61. (in Chinese)
[7] GUO Haitao, ZHU Qingbao, SI Yingtao. Novel Path Planning for Robots Syncretizing Ant Algorithm and Genetic Algorithm. *Journal of Chinese Computer Systems*. 2008; 29(10): 1838-1841. (in Chinese)
[8] Zhang Meiyu, Huang Han, Hao Zhifeng. Motion Planning Of Autonomous Mobile Robot Based On Ant Colony Algorithm. *Computer Engineering and Applications*. 2005; 41(9): 34-37 (in Chinese)
[9] Cai Wenbin, Zhu Qingbao. Rolling Path Planning of Robot Based on Rapidly Exploring Random Tree in an Unknown Environment. *Journal of Nanjing Normal University: Engineering and Technology Edition*. 2009; 6(2): 79-83. (in Chinese)