Application of Dijkstra algorithm in robot path-planning

Huijuan Wang
North China Institute of Aerospace Engineering
Langfang, China
e-mail: wanghuijuan1982@sohu.com

Yuan Yu
Langfang Health School
Langfang, China
e-mail:visbas@sohu.com

Quanbo Yuan
North China Institute of Aerospace Engineering
Langfang, China
e-mail: yuanquanbo@126.com

Abstract—**In this paper, the robot is a maze robot. Dijkstra algorithm is used in the robot path planning. The shortest path is selected in the process of barrier. Simulation results prove the model valid; it can effectively solve the maze robot path planning.**

Keywords-Robot; path-planning; Dijkstra algorithm;

I. INTRODUCTION

Robot path planning is an important area of research, it is an important point between artificial intelligence and robotics. In order to complete the task in a mobile security way, mobile robot path planning is an important indicator of intelligence. According to certain evaluation criteria, path planning is a collision-free path that mobile robot will find and reach the goal of a state from the initial obstacles environment,[1]. Based on the situation of environment, path planning can be divided into global path planning and local path planning. Between them, the global path planning environment focus on the information the situation as known, and local path planning mainly refers to the unknown environmental information on the overall situation.

It is an important problem that global path planning for robot path planning in static environment. It has a wide range of applications. Based on the results of previous studies in this part, robot path planning applications makes an optimal path to walk the maze by independence, and it is proved an effective method with simulation experiments.

II. PROBLEM DESCRIPTION

A. Environment Model

In case, the robot's working space is a two-dimensional structure of space, we use the same size space by two-dimensional grid, it is a standard that the grid size of the obstacles which the robot can take a free walk in a grid .If the robot can walk without any obstacles, the grid is a free grid, and the grid is a barrier grid. In order to sense environmental information through the sensor, to establish an environmental model map, the robot is simplified as a point., obstacles is simplified as a straight line, the thickness of line is no more than a grid of line, to ensure the barrier requirements. To simplify the problem, the standard map is not used complex map, but with the following map to make a better description of algorithm.

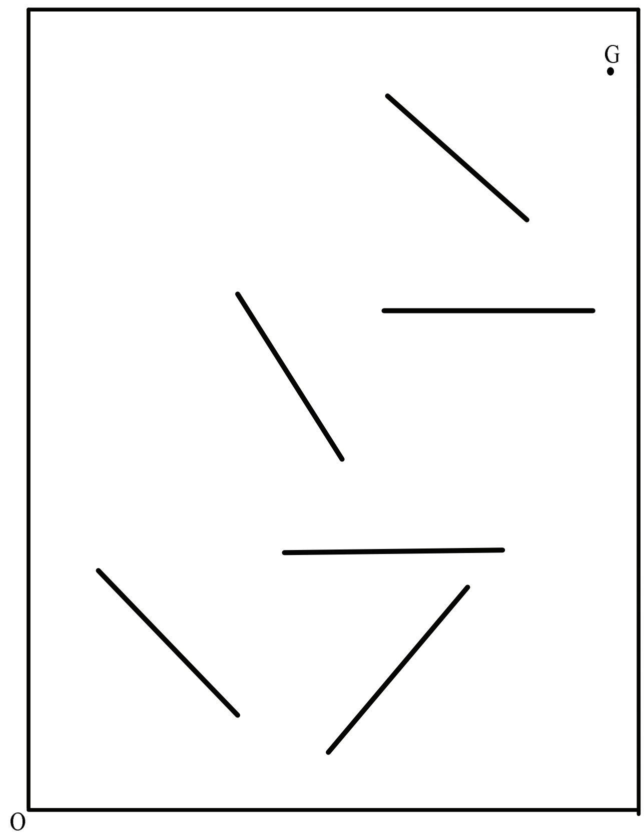

Figure 1. The map of a robot

In order to describe robot motion in the method of Cartesian space Left corner of the lower map is as the origin of coordinates, the horizontal right direction is x-axis positive direction, the straight up direction is Y-axis positive direction, each axis grid interval corresponds to a unit of length. Any one grid can be identified with Cartesian (x, y) uniquely. As is shown in Figure 2.

This paper is supported by Langfang Scientific & Technical Research Foundation under Grant No.2010011004.

978-1-4244-9439-2/11/$26.00 ©2011 IEEE

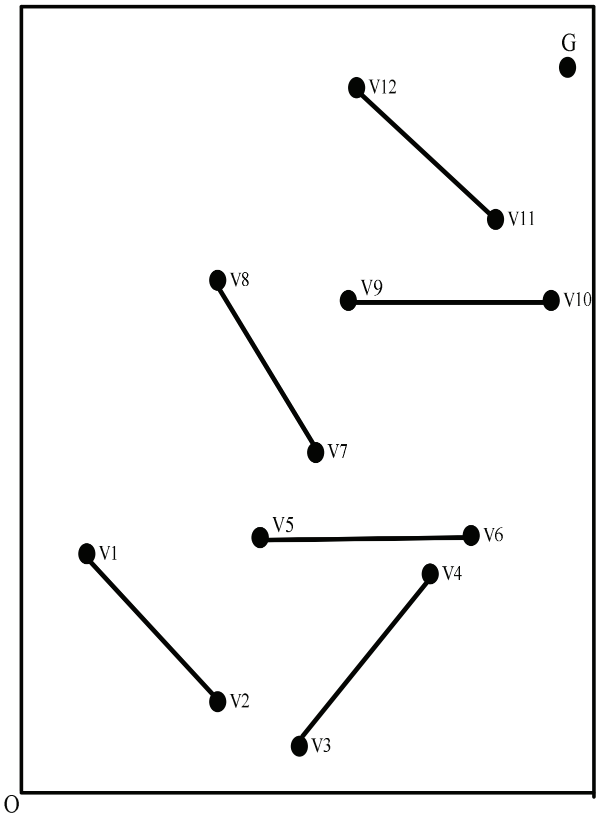

Figure 2. The workspace of a robot

B. Determining the current obstacles

The robot will bypass the obstacles in above model environment as described above, the robot must determine whether to bypass certain obstacles at first. An intersection method is presented, the steps are as followings:

- Drawing a line from the point to the target, the line is *l*;
- Determining whether there are intersections between *l* and obstacles;
- If so, identifying the obstacles that intersected with *l* first, the obstacle is that the current robot will bypass. Connecting the point with the two endpoints (v_i,v_{i+1})of the obstacle (v_i and v_{i+1} are points that the robot path planning may go through);
- If *l* has no intersection with the obstacles, the robot will directly draw a straight line from the target point and judgment is end;
- For v_i and v_{i+1}, they have respectively above four operations.

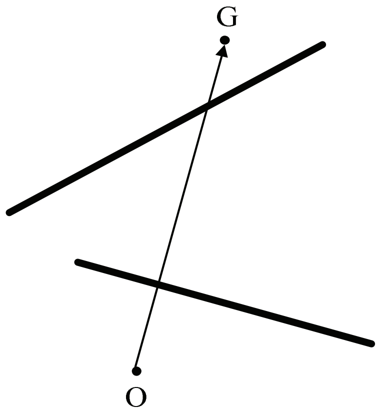

Figure 3. Intersect method

III. DIJKSTRA ALGORITHM

A. Dijkstra algorithm

Dijkstra's algorithm is described as follow: firstly, the introduction of an auxiliary vector D, each of its component D [i] that the length of the shortest path is found in the current starting point from v to each end of v_i. Its initial state is: if there are arcs from v to v_i, then D [i] is the weight of arc; otherwise make D [i] is ∞. Clearly, the length is:

$$D[j]= \underset{i}{Min} \{ D[i] \mid v_i \in V\}$$

This path is a shortest path from v, This path is (v, v_j).

Based on the above analysis, the algorithm can be described as follows:

- Suppose the adjacency matrix arcs with weights to represent directed graph, arcs [i] [j] represents the weight of arc <v_i, v_j>. If < v_i, v_j > does not exist, then it sets arcs [i] [j] will be ∞. S is found the shortest path collection of the end of point from v, its initial state is empty. Then the initial value from the v to map remaining v_i, may be the shortest possible is:

$$D[i]=arcs[\ Locate\ Vex(G,v)[i]\ ] \quad v_i \in V$$

- Select v_j, making

$$D[j]= \underset{i}{Min} \{ D[i] \mid v_i \in V-S\}$$

v_j is the current obtained end point is the shortest path from the v .Order

$$S=S \cup \{j\}$$

- Changes the shortest path length from v to a vertex v_k. If

$$D[j]+ arcs[j][k]< D[k]$$

Changes D[k]:

$$D[k]=D[j]+arcs[j][k]$$

- Repeat two or more times. It is an increasing sequence by the length of the shortest path from the v to the remaining vertexes.

B. Using the Dijkstra algorithm for robot path planning

In the last section of the built environment model, determining the need by using the intersection method to pass the obstacles, drawing the length of the barrier between the end points on the graph to get multiple graphs of vertex with weighted, Figure 4 (the picture shows part of the matrix that v_0 to v_3) is as shown. Adjacency matrix with weight by directed graph is:

$$\begin{vmatrix} \infty & 6.3 & 6.5 & \infty \\ \infty & \infty & \infty & 7.9 \\ \infty & \infty & \infty & 2.7 \\ \infty & \infty & \infty & \infty \end{vmatrix}$$

Figure 4. the matrix of digraph

Dijkstra algorithm is implemented in this matrix, it is obtained the shortest path from the starting point of to the target.

IV. SIMULATION RESULTS

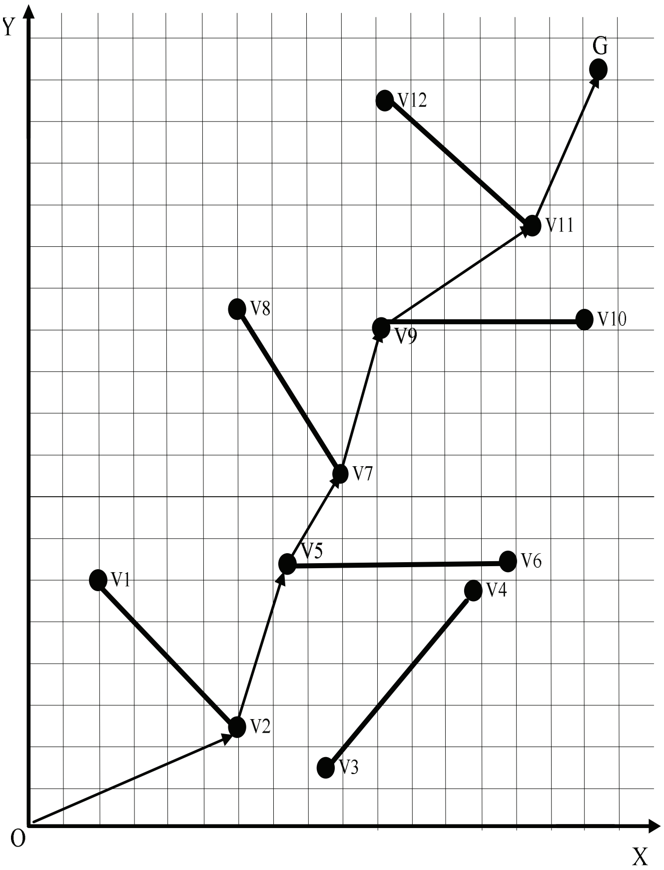

Figure 5. Simulation results of the path planning for Figure 2

To verify the feasibility of the method described, in this paper there is a simulation experiment, as shown in Figure 4. The workspace for path planning simulation results is shown in Figure 5. O is the starting point, and G is the target point. The coarse straight lines are obstacles, and the fine lines is obtained path with the simulation of the final path.

Finally, from the simulation results, the proposed planning algorithm can be effectively to solve the robot path planning problem which is correct and effective.

V. SUMMARY

This paper presents a solving the shortest path planning methods of maze robot path planning, and algorithms Simulation results indicate that the application of the proposed path planning algorithm , Dijkstra algorithm, can be correct and effective in mobile robot path planning.

REFERENCES

[1] 《Introduction to Robotics》Jiang Xinsong

[2] 《Data Structure》Yan Weimin, Wu Weimin, 187-190

[3] WANGC , SOHYC, WANGH ,et al .A hierarchical enetic algorithm for path planning ina static environme nt with obstacles[A].I EEE Canadian Conference on Electrical and Computer Engineering [C].Canada :IE EE, 20 02.(3):1652 一 1657.

[4] GOLDBERG D E .Genetic Algorithm in Search, OptimAssion and Machine Learning[M].New York: Assion 一 Wesley Publishing Company, 1989.

[5] 《Robot control system design and simulation matlab》 Liu Jinkun

[6] NOBORU N,HIDEO T. Path planning of agricultural mobile robot by neural network and genetic algorithm [J]. Computers and Electronics in Agriculture,1997,(18):187—204.

[7] WOONGGIE H, SEUNGMIN B, TAEYONG K. Genetic algorithm based path planning and dynamic obstacle avoidance of mobile robots[A]. IEEE International Conference on Computational Cybernetics and Simulation[C].Orlando:IEEE,1997.(3):2747—27