A Path Planning Algorithm of Mobile Robot in Known 3D Environment

Jiahai Liang*
College of Mathematics and Computer Science, Qinzhou University, Qinzhou, 535000

Abstract

For path planning problem of mobile robot in a known three-dimensional (3D) environment, proposed a path planning algorithm. Firstly, rasterize the known three-dimensional environment. Established the running costs estimate model for the grids from starting point to the target. Using the model to estimate the running costs for each grid. Then used A* search algorithm to calculate the grid path from starting to target with lowest running costs. By detectives of mobile robot, we planned an optimum path under the constraints of grids using hill climbing algorithm. Simulated results show that the algorithm can reduce the uphill or downhill times and running costs improve security, avoid the problem of local minimum. It is simple and robustness.

© 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.

Keywords: mobile robots; known three-dimensional environment; path planning; algorithm

1. Introduction

The path planning of mobile robot is to find the path from start point to the target in an environment with obstacles. So that robot can bypass all the obstacles safety with shortest processing and lowest cost. Path planning is divided into global path planning in known environment and local path planning in unknown environment[3][4]. It also can be divided into 3D and 2D by characteristics of the space[3][4]. For two-dimensional space, the main plan methods include Roadmap, Cell decomposition, Artificial potential field, Swarm algorithms, etc[5][6][7]. The problems of two-dimensional path planning were solved perfectly using these methods. For three-dimensional space, paper[1] gives a path planning algorithm based on energy function of neural network architecture for mobile robot. But this method is mainly for the regular obstacles not for the irregular. Paper [7] proposed a method based on traversability of obstacle

* Corresponding author. Tel.: +0-086-0777-2807161.
E-mail address: qzuljh@163.com.
classification for path planning, but the lack of global planning. This paper proposed a path planning algorithm in the three-dimensional environment to planning path both of global and local. Avoided the problem of local minimum, simply and robustly.

2.1 Path planning theory in Known three-dimensional environment

2.1.1 Question put forward and the path planning algorithm theory

Mobile robot, similar to lunar rover, usually has ability of detection and location. It can sense its own position and detect the forward terrain with certain angle and distance. Generally, the running environment is natural and known. Performance of mobile robot is variation with different environment. So there are different running costs in the three kinds: uphill, downhill and flat running. Here, the running costs including energy consumption, security risk and other integrated factors. It has practical significance to select a path with lowest costs.

Since of its limited detectivity, mobile robot cannot detect directly from starting point to the target. Due to lack of global constraint information, only hill climbing algorithm can be used for planning local optimum path. Though the local optimum path is not global. So, how to convert the known topographic information to global constraint information come to the key in our study. First, we divided the running area into M * N grids (or cells). Secondly, estimated the running costs of each grid by using costs of mobile robots in each terrain and weighted distances from the grid to target. A * algorithm is used to build a grid path from starting gird to the target. Finally, we planned a near-optimal path under the constraints of grids by using the hill climbing algorithm and detectivity of mobile robot. With global optimization of grid path and local optimization of hill climbing algorithm, ensuring the reasonably of the path planning.

2.2 Terrain rasterization and design of running costs estimate model

In theory, reality natural environment can be considered as innumerable terrain control points. It needs to be digitized in order to search out a global path by using A * algorithm. Terrain rasterization is the process of real natural environment digitized. That is, divide the physical space into M * N logic grids (cells). Each grid (cell) is the abstraction of a region of the physical environment. So that infinite physical space can be converted to finite logical units.

In this paper the focus of the study is the global path planning in known 3D environment. That is to plan a path between connected grids from the starting point to the target. To plan such a grid path, required a combination of distance, elevation, terrain fluctuations and other factors. According to these composite factors, we can estimate the running costs when planning global path. The running costs estimate model of mobile robots as follows:

(1) Mean deviation of grid $\overline{h}$: $\overline{h}$ shows the average vertical distance between starting grid to target. Whose estimate model as follow:

\[
\overline{h} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} h_{ij} - h_{t}}{M \times N}
\]

(1)

Where, $M, N$: grid size, $h_{ij}$: the elevation of point $(i, j)$, $h_{t}$: the elevation of target.
(2) The mean-square deviation of elevation is defined as $\overline{S}$, means the fluctuation degree of the grid:

$$\overline{S} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (h_{ij} - \overline{H})^2}{M \times N}$$  \hspace{1cm} (2)

(3) The average distance between grid and target is as follow:

$$\overline{L} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sqrt{(x_{ij} - x_{target})^2 + (y_{ij} - y_{target})^2}}{M \times N}$$  \hspace{1cm} (3)

(4) The total running costs pass a grid is defined as:

$$F = \alpha \times \overline{H} + \beta \times \overline{S} - \delta \times \overline{L}$$  \hspace{1cm} (4)

In this study, the 3D environment is randomly generated by computer. Which size is 1200 *600 length units. Shown as Figure 1. The color depth represents the low and high of elevation. The environment is divided into 12 * 6 grids. Each grid is 100 *100 units in size. Evaluate running costs of each grid using the above estimate model. Parameters are selected as: $\alpha = 6, \beta = 1.5, \delta = 1.0$. Target coordinates: (1120,100). Figure 1 shows the estimate results (The figure in brackets is the serial number of the grid, from top to bottom, left to right).

Fig.1 Rasterization of running environment and estimate results of running costs

3. Path planning

Path planning is divided into two steps. First, to plan grid path. Then, to plan specific path under the constraints of grids. This is a process from rough to detail.

3.1. Planning of grid path

Grid path is the grids which the mobile robot must pass through from the starting point to target. It’s a rough path. It reflects the general direction of the final path. Grid path planning is to find a grid path from the starting point to the target point whose running costs is the lowest. Set the starting grid as root, create a search tree from starting grid to target grid. Shown as Figure 2.
Size of the tree depends on the number of grids. The denser the grid to divide and the larger the search tree be. In the search tree, each node corresponds to a grid. The nodal variable is the estimated running costs from the grid to the target. Using A* search algorithm, to find a path with lowest running costs from the root node.

Grid path of the estimated results as shown in Figure 1 is planned. The final grid path is: 61 → 50 → 51 → 52 → 41 → 30 → 19 → 20 → 21 → 22 → 23.

3.2. Final path planning under the grid path

Rough grid path must be refined to get final path. By detectivity of mobile robot, we plan a optimum path under the constraints of grid using hill climbing algorithm. The basic principle is to find a direction with lowest running costs lead to the next target, within the detection range of the mobile robot. Then mobile robot goes ahead for a step in the direction, until reached the target point. The path the mobile robot traversed is the final path. The detection range of the mobile robot shown as Figure 3,

where, \( i \) is the serial number of detection direction, \( K \) is the amount of detected direction, \( j \) is the serial number of detection distance, \( L \) is the total length of detection range.

The steps to plan path under the constraints of grid as follows:

Step1: Mobile robot search for a certain direction in front \(( K \) directions, each direction in the same interval), within the limits of a certain distance \(( L \) ). The elevation data for each detection are matrix representation as shown in equation (5). Distance between the current points and next center points shown in equation (6)

\[
H=(h_{ij})_{k\times L} = \begin{bmatrix}
h_{i1}, h_{i2}, ..., h_{iL} \\
h_{b1}, h_{b2}, ..., h_{bL} \\
... \\
h_{ki}, h_{k2}, ..., h_{kL}
\end{bmatrix}
\]  

(5)

Where, \( h_{ij} \) : elevation of detection point \((i, j)\).

\[
D=(d_{ij})_{k\times L} = \begin{bmatrix}
d_{i1}, d_{i2}, ..., d_{iL} \\
d_{b1}, d_{b2}, ..., d_{bL} \\
... \\
d_{ki}, d_{k2}, ..., d_{kL}
\end{bmatrix}
\]  

(6)

Where, \( d_{ij} \) : distance between current points to the target.
Step2: Calculate the elevation difference between two adjacent sampling points for each direction, as shown in equation (7). The direction with lowest running costs can be worked out by equation (9).

\[ DH = (dh_k)_{k=1}^{K} = (h_i - h_{i-1})_{k=1}^{K} \quad (7) \]

Where, \( dh_k \) is the elevation difference between the point \((i, j)\) and the point \((i, j-1)\).

\[ Ev = (e_i) = \sum_j dh_j \times \gamma + \sum_j dh_j \times \eta + \sum_j dh_j \times \lambda_k \times \kappa \quad (8) \]

Where, \( e_i \) is the running costs assessed value of the detect direction \( i \), \( \gamma, \eta, \lambda, \kappa \) are the running costs evaluation criteria of the downhill, flat, uphill and distance.

\[ _{k}Min_i = Min(e_i) \quad (9) \]

Where, \( _{k}Min_i \) is the direction with lowest running costs.

Step3: Mobile robot goes ahead in the calculated direction. Stop when reaches target point, else goes to Step1.

Step4: If the grid is the last, then use the current point as final target to calculate running costs of each direction.

4. Simulation and result analysis

In this paper, we designed a simulation program by using Visual C# 2005. First, we generate 3D terrain by random. Secondly, we divided the 3D terrain with grids and evaluate the running costs. Finally, we planned the grid path and the specific path.

Results of simulation are evaluated by using comparison method. Three simulations have been done with different parameters, same starting points and target points. Other parameters as follows:

Test1: No planning, the path direct from the starting point to the target point;

Test2: Using 6 × 12 grids, terrain cost parameters: \( \lambda = 6, \gamma = 1.5, \eta = 0.5, k = 0.5 \);

Test3: Using 12 × 24 grids, terrain cost parameters: \( \lambda = 6, \gamma = 1.5, \eta = 0.5, k = 0.5 \).

![Fig.5 Comparing of paths of the mobile robot in three experiments](image-url)
Table 1 The final results and costs of running

<table>
<thead>
<tr>
<th>scheme</th>
<th>Distance</th>
<th>Uphill steps</th>
<th>Downhill steps</th>
<th>total cost</th>
<th>Optimization rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test1</td>
<td>1238.2</td>
<td>61</td>
<td>114</td>
<td>1156.1</td>
<td>0.0%</td>
</tr>
<tr>
<td>Test2</td>
<td>1276.7</td>
<td>55</td>
<td>108</td>
<td>1130.3</td>
<td>2.2%</td>
</tr>
<tr>
<td>Test3</td>
<td>1264.7</td>
<td>51</td>
<td>104</td>
<td>1094.4</td>
<td>5.4%</td>
</tr>
</tbody>
</table>

Figure 5 shows the paths planned using our algorithm have the same direction. They can bypass the uneven area as possible and finally arrived at the target point. Figure 5 and Table 1 shown that path of Test1 is shortest, but it has the most uphill, downhill and running costs. Results of Test2 and Test3 shown that the path of Test3 is shorter than Test2 because of using higher density grids. It also has fewer uphill and downhill, and lower running costs.

In conclusion, our algorithm can reduce uphill, downhill and running costs. We can improve density of the grids to increase path optimization rate. But the improvement increases the complexity. So we should consider the specific conditions in application.

5. Conclusions

By estimating running costs in known 3D environment and establishing grid path we realized global path planning. Also avoid the problem of local minimum effectively. Under the constraints of grids we plan a local optimizations optimum path with detectivity of mobile robot. This made path further optimal and materialize possible. Simulation results show the effectiveness of the proposed algorithm.

References


