Journal of Southeast University (English Edition) Vol. 24, No. 2, pp. 188 − 191 June 2008 ISSN 1003—7985

Robot coverage algorithm under rectangular decomposition environment

Zhang Chibin Yan Xiaolong

(School of Mechanical Engineering, Southeast University, Nanjing 211189, China)

Abstract: The environment modeling algorithm named rectangular decomposition, which is composed of cellular nodes and interleaving networks, is proposed. The principle of environment modeling is to divide the environment into individual square sub-areas. Each sub-area is orientated by the central point of the sub-areas called a node. The rectangular map based on the square map can enlarge the square area side size to increase the coverage efficiency in the case of there being an adjacent obstacle. Based on this algorithm, a new coverage algorithm, which includes global path planning and local path planning, is introduced. In the global path planning, uncovered subspaces are found by using a special rule. A one-dimensional array P, which is used to obtain the searching priority of node in every direction, is defined as the search rule. The array P includes the condition of coverage towards the adjacent cells, the condition of connectivity and the priorities defined by the user in all eight directions. In the local path planning, every sub-area is covered by using template models according to the shape of the environment. The simulation experiments show that the coverage algorithm is simple, efficient and adapted for complex two-dimensional environments.

Key words: path planning; complete coverage algorithm; rectangular decomposition

Complete coverage path planning is based on optimization of some characters, to find a continuous path from the first point to the end in a special area[1-2]. This path should cover everywhere that can be reached. Complete coverage path planning as a special path planning problem, attracts many people to study it. At present, there are many robot appliances based on these theories, such as cleaning robots, autonomic croppers, underwater robots and so on[3-4]. Presently, research on these algorithms is on the exploration stage. For example, the efficiency of path planning is too low, and some appliances can only carry out simple random searches to implement coverage[5-7]. In this paper, an algorithm based on rectangle decomposition for complete coverage is presented.

1 Environment Modeling

The principle of environment modeling is to divide the environment into individual sub-areas. Each area is presented by a special point called a node. The line through the nodes stands for the connectible relationships between each cellular node. The nodes and the lines construct the whole interactive network.

The first step in rectangular environment modeling is to establish square models. It is the base for rectangular modeling. The following are the definitions of the square environment modeling:

- Node: The point serves as the central point for each cellular area that is divided from the whole environment in the same scale.
- Connectible or disconnectible paths between nodes: If there is a line that can connect two nodes without passing any obstacle, the connection between the two nodes is defined as a connectable path; otherwise, the connection is a disconnectible path.
- Virtual and real node: If the area is covered by obstacles, the node in this area is defined as a virtual node; otherwise, it is defined as a real node.

Square environment modeling divides the environment into square areas in the same scale. Every central point of a square area is defined as a node. The information of each square map includes three aspects of elemental information: the square size, the node and the relationships among the adjacent nodes. The information on a square size is the side length of each square cell. We define the size of a square side as a unit length. The information on a node contains the attribute of virtual or real, the relative coordinates and the absolute coordinates. In Fig. 1, node 2, node 5 and node 11 are regarded as virtual nodes, and the other nodes are regarded as real nodes. Fig. 2 shows that the robot moves along with the border of the environment led by the manipulator. The geomagnetic field sensor corrects the orientation of the robot and the pulse coder gives absolute coordinates. From the nodes around the unknown environment, we obtain the initial information of other nodes. Integrating all the nodes' information, we can establish an electro-map of the environment and update some data. The electro-map is made up of a vector network. The network represents a real physical area and contains data which can be used to describe the attributes of the coverage area, such as the number of cells, the area, covered or not, the size of area and the adjacent areas, connectible or not.

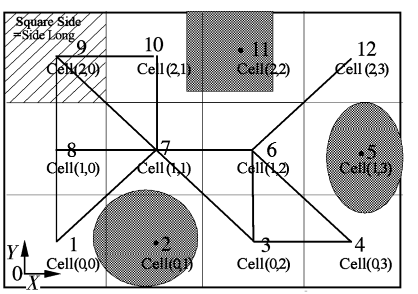

Fig. 1 Square environment modeling

Received 2007-09-17.

Biography: Zhang Chibin (1968—), male, doctor, associate professor, chibinchang@ yahoo. com. cn.

Foundation items: The National Natural Science Foundation of China (No. 50475076), the National High Technology Research and Development Program of China (863 Program) (No. 2006AA04Z234).

Citation: Zhang Chibin, Yan Xiaolong. Robot coverage algorithm under rectangular decomposition environment [J]. Journal of Southeast University (English Edition), 2008, 24(2): 188 − 191.

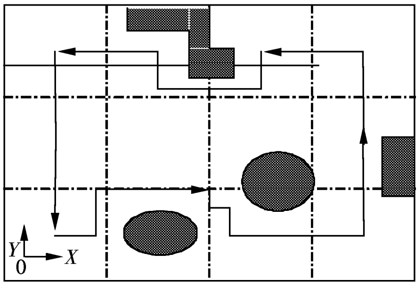

Fig. 2 Side border learning

The rectangular map based on the square map can enlarge square area side size to increase the coverage efficiency in the case of there being an adjacent obstacle. In a certain real node square area, we should judge its relationship with surrounding cells first. To test whether it can connect to the four adjacent nodes on the left, right, upward and/or downward. If the adjacent nodes cannot connect to the real node, the square border in that direction must offset a distance. In Fig. 3, node 7 cannot connect to node 2. So, we choose the maximal values of three sensors in the downward direction and compare them with the sides of the square. If the maximal value is greater than a square side 1 unit, one and half a square side length will be taken as the distance between node 7 and the new rectangle downward border. In Fig. 4, we can obtain the rectangle model map of the environment by transforming all the square areas that contain real nodes in the whole environment.

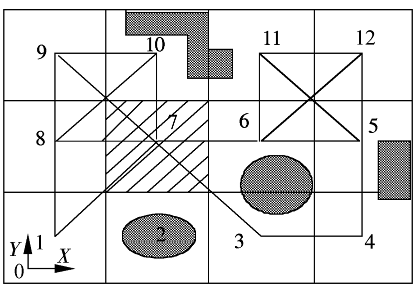

Fig. 3 Square environment modeling

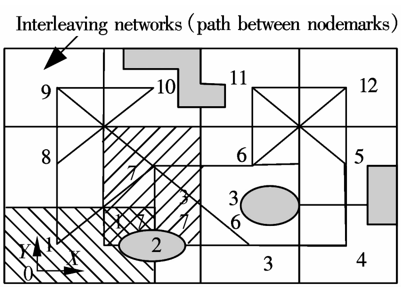

Fig. 4 Rectangle environment

2 Robot Coverage Algorithm

In fact, the rectangle environmental modeling algorithm is one of the cellular decomposition algorithms. The entire environment is divided into several cells by using cellular decomposition. Different from other cellular decompositions, the rectangle environmental modeling algorithm establishes the information on nodes and interleaving networks, as well as every rectangle area corresponding to the only node. To search for nodes on the interleaving network, until every node is found, the whole environment is covered. Therefore, complete coverage task is transformed as searching for a path to cover all the nodes on the interleaving network of rectangle environment modeling.

Based on rectangle decomposition, we propose a new coverage algorithm, which includes global path planning and local path planning (see Fig. 5). The global path planning acts on the interleaving network of rectangle environment modeling and completes the coverage of all the nodes by the rule that has been confirmed beforehand. Moreover, in the local path planning a right coverage established template is chosen by studying the surrounding environment, and the robot completes the coverage in the local area. The global path planning is put into effect between nodes, but the local path planning is carried out in a rectangle area. Each local path planning starts from the node and ends at the node; that is, local coverage starts from the node of the local area and ends at the node of the area after it has covered the area. The global path planning and the local path planning act in turns and finally fulfill the coverage of the whole environment.

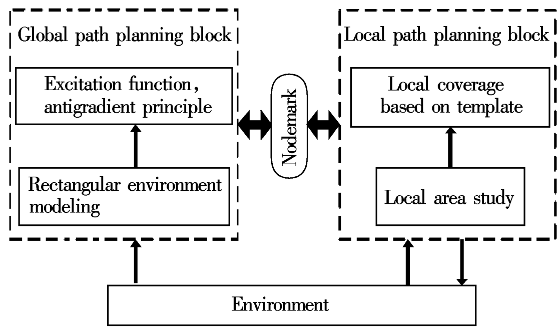

Fig. 5 Complete coverage path planning

To search for the next node, first we should search for it in adjacent rectangle areas, which are not covered in eight directions, then search for it in the local areas that are not covered. The target of the searching rule is to look for the next target in the surrounding nodes. If there is no target node in the surrounding environment, this is a "dead node" problem, and we will discuss it in the next section.

From the former section, we know that the distance between adjacent nodes is related to the space location of nodes. The space location of nodes can be obtained from the electron map of environment modeling. So, in fact, it is only necessary for the searching rule to confirm the direction relationship between the next node and the current node. The next adjacent target node must meet three conditions: 1) The node should be near the current node; 2) The area that the node lies on is not covered; 3) The node can connect to the current node. The nodes that meet the three conditions are more than one. So, how to pick out the appropriate node is the problem for the searching rule to resolve. Therefore, we prescribe eight direction priorities and each direction has a different priority. In Fig. 6, we can see that the starting node is node 1 and the trend of the whole coverage is from downward to upward. In the course of the coverage, it is better to look for a node along the axes of X and Y. In this way, most of the appearances of "missing areas" and "dead node areas" can be avoided. Furthermore, the priority of each direction is different. Based on the reasons above, a numerical value of the coverage direction priorities is confirmed(see Fig. 7). Except for the direction's priorities, the searching rule is related to whether the adjacent areas have been covered and there is connectivity between areas. So the coverage complexion of nearby areas, the connectivity between nearby areas and the direction's priorities are three factors of the searching rule.

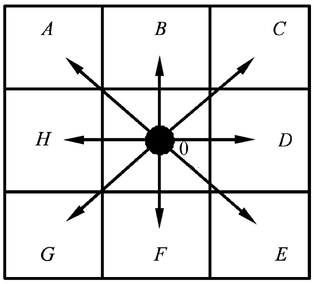

Fig. 6 Moving direction between nodes

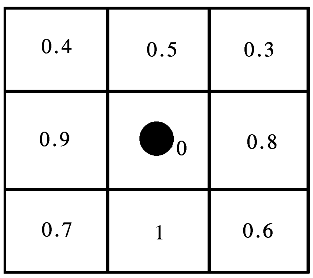

Fig. 7 Priorities of coverage directions

Therefore, we define a one-dimensional array $P[8]$, which is used to obtain the searching priorities of nodes in every direction. $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7$ represent the priority that the mobile robots move towards the regions of A, B, C, D, E, F, H (see Fig. 6). The one-dimensional array $C[8]$ represents the condition of coverage towards the adjacent regions of A, B, C, D, E, F, G, H; value 0 represents "coverage" and value 1 represents "non-coverage". The one-dimensional array $T[8]$ represents the conditions of connectivity in A, B, C, D, E, F, G, H directions, value 1 represents "connectible" and value 0 represents "disconnectable". The one-dimensional array $L[8]$ $(0.3 \leqslant L[i] \leqslant 1,\ i = 0, 1, \ldots, 7)$ represents priorities. Direction F(see Fig. 7) has the top priority; that is to say, $L[5] = 1$. Direction priorities:

$$P[i] = C[i] \times T[i] \times L[i] \qquad i = 0, 1, \ldots, 7 \qquad (1)$$

According to Eq. (1), we can obtain the searching priorities of the eight directions, and we can also obtain the maximal value of the priorities and the directions with maximal coverage priority.

$$P_{\max} = \max(P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7) \qquad (2)$$

$$D_{\mathrm{p}} = i \qquad P_i = P_{\max};\ 0 \leqslant i \leqslant 7 \qquad (3)$$

According to Eq. (3), we can obtain the direction with the maximal priority. Along with this direction, we can calculate the distance, and then move this distance to find the next target node. Obviously, the searching rule is setting the direction with the maximal priority as the searching direction. Continuous searching for the uncovered nodes with this searching rule, we will eventually cover all the nodes.

We will probably encounter a "dead node" when searching for the next target node. The encountered "dead node" means that all the nearby nodes around the current node have been covered, but still there are nodes that have not been covered in the environment. In order to achieve complete coverage, we have to find a path to the nearest "dead node". The anti-gradient principle is a kind of path planning algorithm, which is utilized to obtain the shortest path[7-8]. According to this algorithm, we can divide the environment into several grids, and calculate the nearest distance from every grid to the target node. The existing target node is marked 0 in the grid, while other grids are initialized with greater values. The arithmetic starts from the target node and covers all the cells adjoining it, repeating this approach continuously. If the state $P_i + 1$ which is near state P_i is located in the obstacle, its value is initialized as ∞; otherwise, $V(P_i + 1) = \min(V(P_i) + 1, V(P_i + 1))$. During the course of complete coverage, we may often encounter a "dead node". The solution is: First, we manage to find out the nearest uncovered area according to the electron map; then we calculate the distance from the current area to all the uncovered areas(special distances), and then we find out the nearest uncovered area as the next target area. Finally, the robot discovers the path to the nearest uncovered area with the anti-gradient principle.

The algorithm defines three kinds of basic coverage templates(spiral motion, horizontal reciprocating motion, vertical reciprocating motion) in local coverage path planning. The coverage template choice depends on the shape of the environment, and the shape of the environment can be obtained by environmental study. The spiral motion template is usually utilized in the situation that there is no obstacle in the surrounding rectangle area, and the latter two templates are utilized in the situation that there is an obstacle nearby.

3 Simulation Experiments

According to the algorithm of complete coverage path planning, we have done many simulation experiments. Figs. 8(a) to (f) show the results of local rectangle area coverage. Fig. 9 shows complete coverage path planning and lines represent complete path planning.

Coverage ratios and repetitious ratios are different on different maps. Lots of simulation results indicate that the algorithm of coverage achieves more than 90% of coverage ratio, some even higher than 95%, and the repetitious ratio is below 10%. As a result, for the most of the maps, the algorithm is highly efficient, practical and adaptive.

4 Conclusion

The complete coverage process is carried out by the robot we developed. The algorithm we proposed is highly real time and has less information to deal with. The system has few missing areas after coverage searching. But sometimes, it is hard to avoid recovered areas nearby the obstacle. However, it is a great improvement on the previous algorithms. Optimization of the rectangle modeling for complex environments is the next subject to be studied.

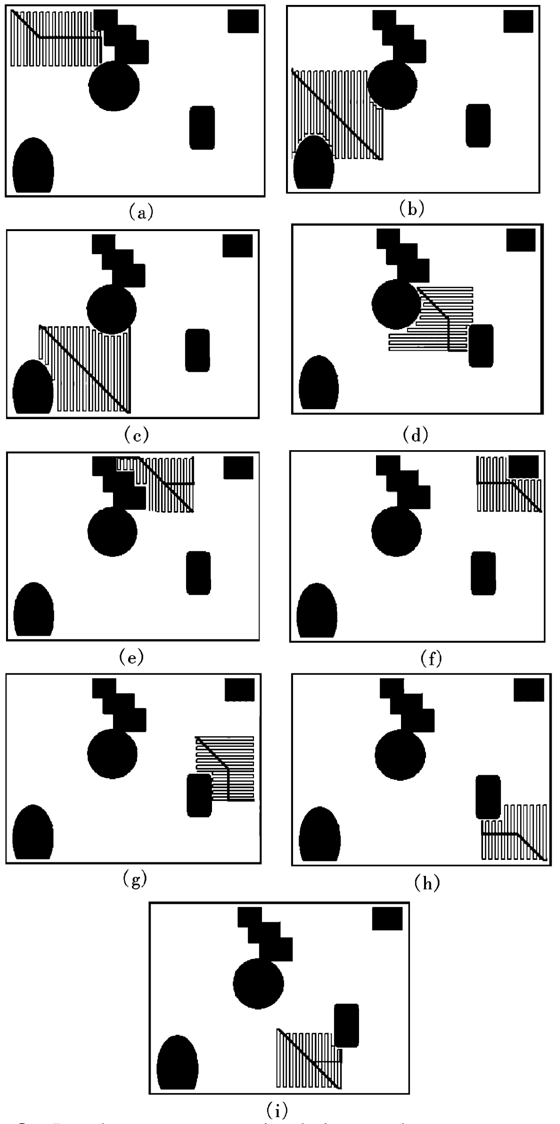

Fig. 8 Local area coverage simulation results. (a) 1 region coverage; (b) 2 region coverage; (c) 3 region coverage; (d) 4 region coverage; (e) 5 region coverage; (f) 6 region coverage; (g) 7 region coverage; (h) 8 region coverage; (i) 9 region coverage

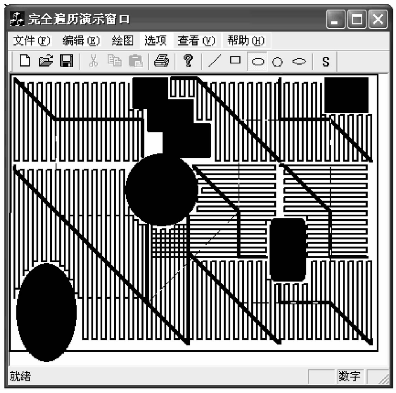

Fig. 9 Complete coverage path planning

References

[1] Zheng Xiaoming, Koenig Sven. Robot coverage of terrain with non-uniform traversability [C]//*Proceedings of the* 2007 *IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Diego, 2007: 3757 − 3764.

[2] Hazon N, Kaminka G A. Redundancy efficiency and robustness in multi-robot coverage[C]//*Proceedings of the* 2005 *IEEE International Conference on Robotics and Automation*. Barcelona, Spain, 2005: 735 − 741.

[3] Correll N, Martinoli A. Robust distributed coverage using a swarm of miniature robots [C]//2007 *IEEE International Conference on Robotics and Automation*. Roma, Italy, 2007: 379 − 384.

[4] Kondo M, Kimura K. Collision avoidance using a free space enumeration method based on grid expansion[J]. *Advanced Robotics*, 1989, **3**(3): 159 − 175.

[5] Trahanias P. Visual recognition of workspace landmarks for topological navigation [J]. *Autonomous Robotics*, 1999, **7** (2): 143 − 158.

[6] Alberto Elfes. Sonar-based real-world mapping and navigation[J]. *IEEE Journal of Robotics and Automation*, 1987, **RA-3**(3): 249 − 265.

[7] Wang Meng, Liu James N K. Online path searching for autonomous robot navigation[C]//2004 *IEEE Conference on Robotics, Automation and Mechatronics*. Singapore, 2004: 746 − 751.

矩形环境分解机器人遍历算法

张赤斌　　颜肖龙

(东南大学机械工程学院,南京 211189)

摘要:提出了一种矩形分解的环境建模方法. 该方法模型用环境分解出的节点及节点之间的连接关系网络来表示. 环境建模首先将环境分解为正方形区域,区域中心点代表该区域位置. 在邻近区域有障碍物情况下,扩大正方形边长形成矩形区域,有利于提高局部遍历效率. 在这种环境建模思想下,提出了一种包含局部和全局 2 层结构路径规划下的机器人遍历方法. 在全局路径规划中子区域之间的探索采用了基于知识规则的方法,定义了决定局部子区域向周围区域移动的一维优先权数组,该数组综合表征了本区域与相邻区域的遍历情况、连通关系和用户定义优先权级别. 局部子区域根据环境形状采用模版匹配法实现遍历. 仿真实验结果说明,该方法简单、高效,并适合解决复杂二维环境遍历问题.

关键词:路径规划;完全遍历算法;矩形分解

中图分类号:TP27