Proceedings of the 34th Chinese Control Conference
July 28-30, 2015, Hangzhou, China

A Complete Coverage Path Planning Algorithm for Mobile Robot Based on FSM and Rolling Window Approach in Unknown Environment

LI Caihong[1], WANG Fengying[1], SONG Yong[1,2], LIANG Zhenying[1], WANG Zhiqiang[1]

1. School of Computer Science and Technology, Shandong University of Technology, Shandong 255049
E-mail: lich@sdut.edu.cn

2. School of Control Science and Engineering, Shandong University, Shandong 264209
E-mail: songyong@sdu.edu.cn

Abstract: This paper presents an algorithm of complete coverage path planning (CCPP) for mobile robot in unknown environment based on the Finite State Machine (FSM) approach and rolling windows approach. The rolling window approach is used to detect the local environments. The robot only uses on-board sensors to acquire a limited range knowledge of the surroundings and construct the rolling windows. Then it translates the local unknown environment information to the known range. The proposed algorithm further abstracts the known environment as a union of robot-sized cells. In other words, range in rolling window is constructed to a cell map, or a grid map. Based on the grid map, the designed FSM method which has five states and three strategies is used to organize the CCPP task in the rolling windows. Strategies of the FSM method assume the greedy means to own the local optimal performance for the whole path planning task. To verify the designed method performance, we compare the simulation result with that of the method which uses random search planning. Simulation results show that the mobile robot can cover the entire workspace with low repetition rate and high work efficiency.

Key Words: Mobile robot, complete coverage path planning (CCPP), Finite State Machine (FSM), rolling window, grid map

1 Introduction

Complete coverage path planning (CCPP) of mobile robot is a special type of trajectory generation in 2-dimentinal (2D) environments. It requires the robot to pass through the whole area of the workspace and optimize some performance cost, e.g., the shortest path and the minimum repeat coverage [1]. Many robot appliances require the CCPP, such as cleaning robots, vacuum robots, painter robots, lawn mowers, land mine detectors and so on [2].

Most existing CCPP approaches for coverage path planning assume that the environment is static and known. In known environments, the planning problem can be solved by off-line global planning. But to be useful in the real world, the environmental information for path planning is not complete or even unknown. The robot can only detect local sensory information of the environment. CCPP is required to move safely in unknown environments.

There have been many studies on CCPP in unknown environments using various techniques, such as cell decomposition [3], template based on models [4], neural network approaches [5], fuzzy logic models [6], genetic algorithm [7], sensor based coverage approaches [8], etc. Most existed algorithms use grid map representation based on approximate cellular decomposition, where the size of each cell is the same as the robot. David Michel et al.[9] used a grid map to represent the workspace and assigned each cell a value for movement cost. Tae-Kyeong Lee et al. [10] presented an on-line CCPP algorithm for mobile robot by adopting a high-resolution grid map representation of the environment based on approximate cellular decomposition.

*This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61473179; Shandong Province Natural Science Fund of China under Grant ZR2014FM007 and ZR2013012FM; Shandong Province Science and Technology Development Projects under Grant 2013GGX10116; A Project of Shandong Province Higher Educational Science and Technology Program under Grant J13LN27.

Young-Ho Choi et al. [11] introduced a sensor-based online CCPP algorithm guaranteeing a complete coverage of unstructured planar environments by a mobile robot. It can be largely classified as an approximate cellular decomposition approach. In this representation, choosing the robot's next position in the map is very simple. And it can also total out the CCPP coverage rate conveniently by counting the grid numbers. In this paper, we also use the grid map based on the rolling window to represent the robot working space and evaluate the coverage performance.

Meanwhile, the rolling windows method is a powerful tool for the planning problems in unknown environments. It executes local planning repeatedly and makes full use of the newest local environmental information detected. It can solve the planning problem in an unknown environment with high efficiency but low calculation burden [12]. Here we also use it to construct the local grid map.

The Finite State Machine (FSM) approach can be used for intelligent movement in unknown environment. FSM means can generally be achieved by mapping high-level tasks (e.g. path planning) to state machines. Building such a system is not trivial and requires careful design of states and state transitions. S. Vechet and V. Ondrousek [13] presented the method for motion planning of autonomous mobile robot in highly populated environment. M.Foukarakis et al. [14] proposed combination of a well-known FSM and a custom decision-making tool for implementing adaptive robot behaviors. In this paper we use FSM to achieve the CCPP task and owning the better coverage performance of the local shortest path by designing a greedy strategy.

Some of the early approaches for CCPP task relied on random approaches [15]. The robot does not plan searching path, but rather select direction at random until it encounters an obstacle for retrieval. A random search does not guarantee complete coverage, but the robots using randomized search strategies can be built for less cost than robots using methods that require precise positioning [1]. So the random approach is still applicable in real robots. In this paper, the random method is used to compare our proposed approach performance.

This paper presents a novel CCPP method based on the FSM and rolling window. The FSM is organized based on the local grid map and the rolling window is constructed on-line with contact sensors. The mobile robot performs the planning method in the unknown environment and covers the whole workspace in the series rolling windows. The sequence of analysis and movement is performed continuously until the accessible space has been explored to meet the requirements. Comparison studies and simulation results illustrate the effectiveness and efficiency of the proposed method.

This paper is organized as follows. In Section 2, combining the rolling window method, the unknown environment is modeled as a known local grid map and the various property cells are defined. Section 3 presents the FSM algorithm to achieve the CCPP task based on the constructed grid map. The three designed strategies in FSM are described in Section 4. Two evaluation indexes are devised to estimate the coverage path planning effect of the designed algorithm in Section 5. Section 6 offers the verification of the proposed CCPP with the simulation controller. And the comparison of the proposed method with the common random CCPP approach has also been studied by simulations. Conclusions are drawn in Section 7.

2 Environment Modeling

Consider a mobile robot in a two-dimensional unknown workspace of finite size. Workspace is convex and is arbitrarily cluttered with finite number of static convex obstacles. Robot has no priori knowledge of the workspace. At any instant, robot can only scan a local circular region by its on-board sensors around itself, whose radius is equal to *r*. The circular region is called rolling window, here labeled as S_{sense} . Robot can move omni-directionally in the workspace with a step length ε ($0<\varepsilon\le r$). The scanning angle is supposed as 360 degree and when the robot takes a step, S_{sense} will be refreshed. So the adjacent S_{sense} will be lapped over in some degree.

The path planning is in a rolling style, and in each step, the FSM is used. Environment modeling is only carried in current S_{sense} , so it can reduce calculation and it is effective. With the rolling window moving forward, robot obtains newer environmental information. Update the environmental information in the current S_{sense} . The rolling window rolls forward a step.

The range of S_{sense} is divided into cells of equal size to that of the step length ε . So the robot constructs a grid map based on the current S_{sense} . Fig.1 shows the model of the rolling window S_{sense} . The center of S_{sense} is the robot current position, labeled as P_{robot} . And the obstacles position is labeled as $P_{obstacle(i)}$ $(i=1,...,n)$. So the task of the robot is to cover each free grid in the rolling window S_{sense} as quickly as possible.

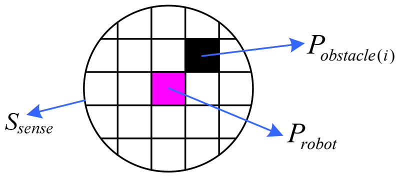


Fig.1: Constructed grid map in the rolling window S_{sense}

In order to design FSM method for the CCPP task, each grid has been assigned four properties, which are obstacle property $Flag(x,y)$, coverage property $Vivited(x,y)$, associate property $\mathrm{Re}\,lated(x,y)$, and the optional property $Enabled(x,y)$ respectively. And each grid position is identified by its center coordinate (x,y) . Meaning and value of each property are defined as follows.

1) Obstacle property $Flag(x,y)$

$$Flag(x,y)=\begin{cases}1, & free\\ 0, & occupied\end{cases} \tag{1}$$

The property means whether the grid has been occupied by obstacles or not in the work space.

2) Coverage property $Vivited(x,y)$

$$Visited(x,y)=\begin{cases}0, & uncovered\\ 1, & covered\ only\ once\\ \ge 2, & coverage\ times\end{cases} \tag{2}$$

The property represents the coverage times of a free grid by the robot during the CCPP procedure.

3) Associate property $\mathrm{Re}\,lated(x,y)$

$\mathrm{Re}\,lated(x,y)$ value is equal to the numbers of the free grids adjacent to a certain cell. The property means how much free grids are around a cell.

4) Optional property $Enabled(x,y)$

$$Enabled(x,y)=\begin{cases}0, & outside\ the\ S_{sense}\\ 1, & in\ the\ S_{sense}\end{cases} \tag{3}$$

The property shows whether a free grid is in the rolling window S_{sense} or not.

3 FSM Algorithm

In the constructed grid map S_{sense} , FSM algorithm is designed to achieve the CCPP task and is expected to have a better performance to completely cover each free grid in S_{sense} as much as possible based on the greedy thought. Then the robot executes FSM repeatedly and makes full use of the newest local environmental information. At each step of planning, the mobile robot covers the detected area based on the locally detected information by FSM method within the current rolling window S_{sense} . Then it moves a step along the local path. Environment modeling is only carried in the current rolling window, and so it can reduce calculation and it is effective. Here the complete coverage procedure stops when the total coverage rate achieves 98%.

The designed FSM algorithm covers five states and three strategies. In any instant robot executes one strategy based on the current state. The three strategies transform according to the changing of the five states. A free grid in S_{sense} is selected as the local target grid of the robot when it has not been visited and is nearest to the robot. So the local CCPP can achieve a better coverage performance within S_{sense}. The target grid is also the start point of the next planning.

The five states and three strategies of FSM algorithm are defined as follows.

S_1: initial state

S_2: state of existing uncovered free grids adjacent to P_{robot}

S_3: state of existing no uncovered free grids adjacent to P_{robot}

S_4: state of the target grid has been found

S_5: the end state

F_1: strategy of searching the grid owning the maximum property value of associate $\mathrm{Re}\,lated(x, y)$

F_2: strategy of searching the local target grid

F_3: strategy of two-point searching

Fig.2 shows the designed FSM flow chart for CCPP task.

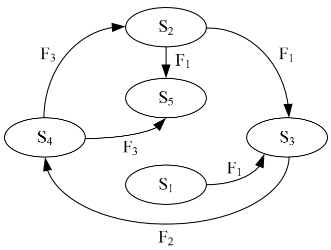


Fig.2: FSM flow chart for CCPP task

When the robot is in initial state S_1, executes strategy F_1 until all the adjacent free grids has been visited. And the robot state converts from S_1 to S_3. Then the robot executes F_2. When the target grid has been found, the robot performs F_3. Finally the robot moves to the local target grid and carries out F_1. Then repeat this process until the coverage rate achieves. This is the organization procedure of FSM algorithm also.

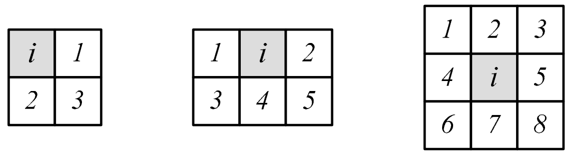
4 Strategies Designing

4.1 *F1* Strategy Designing

The robot executes F_1 strategy when in S_2 state. Task of F_1 strategy is to search the adjacent free grid which has the maximum associate $\mathrm{Re}\,lated(x, y)$ and covers it. The robot always covers the free grid which has the maximum associate $\mathrm{Re}\,lated(x, y)$ preferentially based on the greedy thought. With the rolling window S_{sense} moving forward, robot obtains newer environmental information when it moves a step and detects whether there exists any adjacent uncover free grids. If exists, robot executes F_1 strategy, otherwise robot jumps to S_3 state.

In F_1 strategy, the values of all related grids need to satisfy the following conditions: $Flag(x, y) = 1$, $\mathrm{Re}\,lated(x, y)$ having the maximum value, and $Enabled(x, y) = 1$.

In F_1 strategy, computation of the maximum associate $\mathrm{Re}\,lated(x, y)$ is the key problem. According to the position of a certain grid i in the local grid map, the local grid map can be modeled as three sizes, namely four-grid size, six-grid size and nine-grid size. All arbitrary grid regions are in line with the three models size by transformation. So we only study the three sizes. Fig.3 shows the three models.



(a) four-size (b) six-size (c) nine-size

Fig.3: Three models of the local grid map

An array structure is assumed to store the associate grids information. Suppose the array name is $A(i)$, which i is the grid serial number.

When grid i is the free grid, its associate $\mathrm{Re}\,lated(x, y)$ value are expressed as follows according to the models respectively.

a) four-size model

$$A(i).\mathrm{Re}\,lated = \sum_{i=1}^{3} A(i).Flag(x, y) \tag{4}$$

b) six-size model

$$A(i).\mathrm{Re}\,lated = \sum_{i=1}^{5} A(i).Flag(x, y) \tag{5}$$

c) nine-size model

$$A(i).\mathrm{Re}\,lated = \sum_{i=1}^{8} A(i).Flag(x, y) \tag{6}$$

Here when the adjacent grids to the grid i are obstacle grids, the responding $Flag(x, y)$ value of the grids are equal to zero, namely $Flag(x, y) = 0$. So there is no influence to the $\mathrm{Re}\,lated(x, y)$ property of grid i. The value of $A(i).\mathrm{Re}\,lated(x, y)$ can reflect the true number of the associate grids to grid i.

When grid i is an obstacle grid,

$$A(i).\mathrm{Re}\,lated(x, y) = 0 \tag{7}$$

If grid i is the robot grid, search free grid j owning the maximum associate $\mathrm{Re}\,lated(x, y)$ according to the following and above formulas.

a) four-size model

$$A(i).\max \mathrm{Re}\,lated = \max\{A(i).\mathrm{Re}\,lated \mid i = 1 \sim 3\} \tag{8}$$

b) six-size model

$$A(i).\max \mathrm{Re}\,lated = \max\{A(i).\mathrm{Re}\,lated \mid i = 1 \sim 5\} \tag{9}$$

c) nine-size model

$$A(i).\max \mathrm{Re}\,lated = \max\{A(i).\mathrm{Re}\,lated \mid i = 1 \sim 8\} \tag{10}$$

Compute $A(i).\max \mathrm{Re}\,lated$ by comparing all the associate $\mathrm{Re}\,lated(x, y)$. And record the result grid j of maximum associate $\mathrm{Re}\,lated(x, y)$. Then the robot moves to grid j and updates the environmental information.

4.2 *F2* Strategy Designing

The robot needs to find a target grid k which is the nearest grid to the robot current position and has not been covered in the current rolling window S_{sense} to leave the already complete covered area and go to the new uncovered region. Then the robot changes to state S_4 when the grid k has been found.

In F_2 strategy, the values of all related grids need to satisfy the following condition: $Flag(x,y)=1$, $Visited(x,y)=0$, $Enabled(x,y)=1$. The distance between robot and the detected grid should have the minimum value.

The center coordinate (x,y) of a grid is used to mark its location. Then the distance $Dist(i)$ between the selected grid $P_i(x_i,y_i)$ ($i=1,\ldots,m$) and the robot position $P_{robot}(x_r,y_r)$ is:

$$Dist(i)=\sqrt{(x_i-x_r)^2+(y_i-y_r)^2} \quad (i=1,\ldots,m) \tag{11}$$

And the minimum distance is:

$$Dist=\min(Dist(i)) \quad (i=1,\ldots,m) \tag{12}$$

Then the serial number of the grid which has the minimum distance is labeled as k, and the target grid k has been found.

4.3 *F3* Strategy Designing

The robot moves to the target grid k which has been found. During the moving procedure, the whole complete coverage path planning stops when the coverage rate satisfies the need. Otherwise the FSM procedure transits to state S_2.

F_3 strategy is also designed based on the greedy thought. It's thought is that the robot moves to the target grid always in a straight line. During the moving procedure, the robot changes its direction to avoid the front obstacles while it always head to the target grid. Fig.4 shows the procedure.

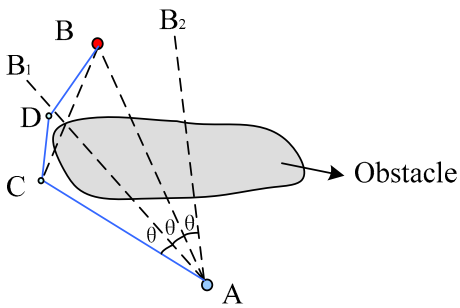


Fig.4: Searching procedure of F_3 strategy

In fig.4, A is the start point, and B is the target grid. There is an obstacle existing on the segment AB lining point A to B. Angle θ is the turning angle when the method need to change searching direction. The robot first turns left at an angle θ and gets a segment AB_1. Judge whether there are obstacles on the segment. If exits then turn right at an angle θ and gets another segment AB_2. Repeat the judgements until there are no obstacles on the segments. In fig.4, point C and D are the local target points during the searching procedure. The procedure repeats until the robot reaches to the target B.

5 Evaluation Index

We need to design some evaluation indexes to test the CCPP effect of the designed FSM method in the workspace. The coverage rate J_{hc} and overlapping coverage rate J_{cc} are used here. Suppose symbol S expresses the whole working area, S_Ω expresses the free area, S_{hc} means the covered area, and S_{cc} means the overlapping area. Then J_{hc} and J_{cc} can be computed as follows.

$$J_{hc}=(S_{hc}/S_\Omega)\times 100\% \tag{13}$$

$$J_{cc}=(S_{cc}/S_\Omega)\times 100\% \tag{14}$$

We hope J_{hc} the higher, the better, and J_{cc} the lower, the better.

6 Simulation

Verification of the proposed CCPP is conducted with the simulation controller. The simulation platform can perform the following functions, such as the running environment setting up, the size of the rolling window S_{sense} demonstration of the algorithm effect, color label for different property grids and the computation result of the evaluation index. In the grid map, the obstacles can be set up randomly. In fig.5, the black grids express the obstacles, and

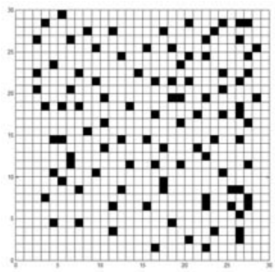


Fig.5: Running environment setting

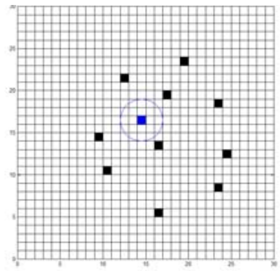


Fig.6: The rolling window S_{sense}

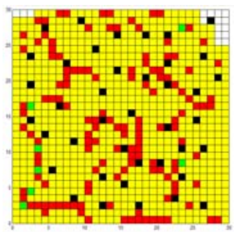


Fig.7: Simulation result of the FSM algorithm

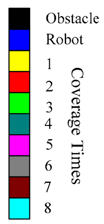


Fig.8: Colored grids label

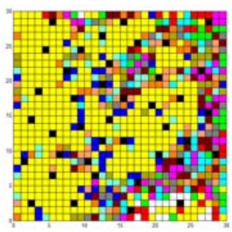


Fig.9: Simulation result of the random algorithm

the white grids express the free grids. Then select a free grid arbitrarily as the robot original position in the grid map and construct a rolling window S_{sense} around it. In fig.6, the blue cell expresses the robot location, and the blue circle expresses the rolling window S_{sense} . Here suppose the coverage rate is about 98%. Fig.7 shows the simulation result of the designed FSM algorithm and CCPP for one case running environment setting. The meaning of the different colored cells is described in fig.8. The simulation platform can also total out the overlap rate. For example, as fig.7 setting, when the coverage rate is 98%, the overlap rate is 11.9767%。The comparison of the proposed method with the common random CCPP approach has been studied by simulations also. From the colored map in fig.9, we can see the overlap rate is very high than the proposed algorithm when the robot achieve 98% coverage rate. The overlap rate is about 40%. So the designed FSM algorithm has a better behavior than the common random methods.

7 Conclusion

In this paper, the CCPP task of mobile robot in globally unknown environments is considered. Taking the advantage of FSM and the rolling windows, the present method is able to perform the real-time path planning of the mobile robot effectively in unknown environment. The performance of the present method is better than that of the common used random approach by simulation comparison. The simulations also show that the present approach has a high complete coverage rate and a lower overlap rate, and it is computationally simple and effective.

References

[1] G.Xiaoqin, Coverage rolling path planning of unknown environments with dynamic heuristic searching, *2009 WRI World Congress on Computer Science and Information Engineering*, 2009: 261-265.

[2] C.Young-Ho, L.Tae-Kyeong, B.Sang-Hoon, etc. Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform, *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009:5788-5793.

[3] D.Batsaikhan, A.Janchiv, S.G.Lee. Sensor-based incremental boustrophedon decomposition for coverage path planning of a mobile robot, *12th International Conference on Intelligent Autonomous Systems*, 2012:621-628.

[4] T.Chunying, L.Yu, F.Shenshen, etc. Complete coverage of known space-rectangular decomposition, *Jixie Gongcheng Xuebao/Chinese Journal of Mechanical Engineering*, 40(10): 56-61, 2004.

[5] L.Chaomin, Y.X.Simon. A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments, *IEEE Transactions on Neural Networks*, 19(7): 1279-1298, 2008.

[6] X.Xiao, F.Yong-Chun, H.Feng, etc. Autonomous search technology for mobile robots under unknown environments, *Jiqiren/Robot*, 29(3): 224-229, 2007.

[7] K.Muzaffer, A.Mete, O.Metin, etc. A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time, *Journal of Intelligent Manufacturing*, 23(4):1035-1045, 2012.

[8] S.Aydin, K.Gokhan, P.Osman, etc. Energy constrained multi-robot sensor-based coverage path planning using capacitated arc routing approach, *Robotics and Autonomous Systems*, 58(5): 529-538, 2010.

[9] M.David, M.Kenneth, New path planning scheme for complete coverage of mapped areas by single and multiple robots, *2012 IEEE International Conference on Mechatronics and Automation*, 2012:1233-1240.

[10] L.Tae-Kyeong, B.Sang-Hoon, O.Se-Young, etc. Complete coverage algorithm based on linked smooth spiral paths for mobile robots, *11th International Conference on Control, Automation, Robotics and Vision*, 2010: 609-614.

[11] C.Young-Ho, P.Kyungbuk, L.Tae-Kyeong, and etc. Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform, *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009: 5788-5793.

[12] Q.Xuena, S.Jiatao, Zh.Xuejun and etc. A complete coverage path planning method for mobile robot in uncertain environments, *2006 Proceedings of the World Congress on Intelligent Control and Automation*, 2006: 8892-8896.

[13] S. Vechet, V. Ondrousek. Motion planning of autonomous mobile robot in highly populated dynamic environment, *Mechatronics*, 2012:453-461.

[14] M.Foukarakis, A.Leonidis, M.Antona and etc. Combining finite state machine and decision-making tools for adaptable robot behavior, *Universal Access in Human-Computer Interaction. Aging and Assistive Environments, Lecture Notes in Computer Science*, 8515: 625-635, 2014.

[15] D. Gage. Randomized search strategies with imperfect sensor, *Proc of SPIE Mobile* Robots Ⅷ, 270-279, 1993.