An Artificial Intelligent algorithm for an autonomous Cleaning Robot

현웅근
Woong Keun Hyun
호남대 전자공학과
*Dept. of Electronics Engr., Honam Univ. KwangJu, KOREA
wkhyun@honam.honam.ac.kr

ABSTRACT

An intelligent path planning algorithm for an autonomous cleaning robot is presented. This algorithm recognizes obstacle on the architectural CAD draft and generates subgoals as tracking points which executes the area filling task based on heuristic approach. A sweeping path is planned by sequentially connecting the tracking points in such a way that (1) the connected line segments should be crossed, (2) the total tracking points should be as short as possible, (3) the tracking line should not pass through the obstacle. Feasibility of the developed techniques has been demonstrated on real architectural CAD draft.

1. Introduction

Public services such as smearing or cleaning of building, platform or hall will dominate in a growing affluent society. To meet the demands of high quality service at reasonable cost, competitive service companies have an increasing interest in applying innovative technologies ensuring economical benefits. Consequently there is a need for automation of such service tasks. In the past mobile robotics research has focused mainly on transportation and manipulation tasks, while service robot in smearing or cleaning task still seem to be open field of investigation. A few researcher have presented aspects of path planning and navigation for cleaning robot[1,2,3]. The development of specialized service robots was brought about by the increasing demand to economize on several kinds of services. A representative application fields are floor-cleaning or smearing of building. Most of today's smearing machines are guided by human operators along extended and fixed motion patterns in areas often greater than some hundreds of square meters. Due to increasing machine maintenance and personal costs this cleaning task should be performed in future by an autonomous robot. Many researcher have been devoted to the path planning algorithm[4,5,6]. Among them, some researcher developed semi-autonomous sweeping robots and applied it to the floor-cleaning tasks. However these robots may meet dead-lock station in some environment with obstacles owing to local collision avoidance approach based on sensors. Hofner proposed a heuristic path planning algorithm using potential field function which is bounded in local planning approach. This method also can not solve the dead-locking problem[7].

This paper presents an planning guidance techniques for smearing path planning for free navigating service robots. To identify a complete smearing task, we distinguished the following operation. (1) interfacing architectural CAD S/W, (2) off-line obstacle map building, (3) scanning the whole workspace for subgoals of sweeping line, (4) tracking sequence of the subgoals, and (5) obstacle avoiding.

Section 2 outlines the operation of the proposed smearing path planner. Section 3 presents a discussion of heuristic subgoal generating technique, and in section 4, a sweeping line generation method by sequential connection of the subgoals is described.

2. An Area filling Path planning for a Cleaning robot on the Architectural CAD draft

Since a cleaning robot works on 2-dimensional space, the objects and robot are modeled in 2-dimensional space. It is assumed that (1) the mobile robot moves in omni-direction, and (2) the shape of the robot and obstacle in workspace are polygon or circular type. A flow chart of smearing robot planner is shown in Fig. 1. In Auto CAD interface module programmed by Auto Lisp, workspace and working robot type are selected. The obstacles of DXF file format are recognized in next procedure using AUTO CAD data library. For obstacle avoidance, obstacle map is built by scanning the whole workspace in the obstacle map building procedure. The map is constructed in 3-dimensional space and quantized as a equal size cell. A cell in the grid-based configuration space is identified as FREE, FULL cell.

Subgoal points generation procedure scans workspace with an adjustable width and records the contacting point

The work reported in this paper was supported by KOSEF under grant number 981-0910-045-2.

between robot and obstacle as a subgoals for sweeping path. The next procedure, subgoal point scanning procedure generates a sequential sweeping path in such a way that (1) the path connecting the subgoals should not be crossed each other and the length of the sweeping path is as short as possible. If the path goes through the obstacle, the obstacle avoidance procedure generates a collision free path for the path segments. Finally, the data is converted to Auto CAD data format for display in Auto CAD tool and robot language command is generated for the sweeping motion.

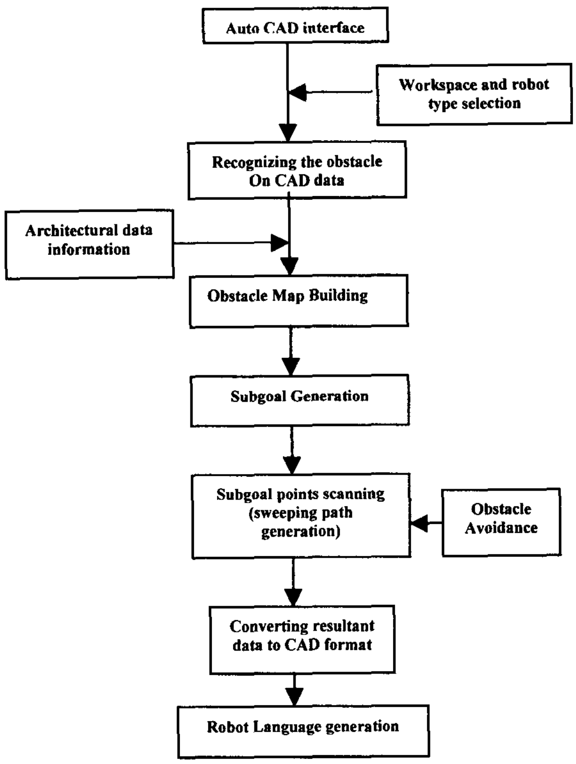

Fig. 1 A structure of the proposed algorithm

3. Subgoal points generation procedure

Subgoal points generation procedure scans the whole workspace with adjustable width and records the contacted position between robot and obstacle while scanning. In this processing, the separation of each scanning line is set to be half of the width of the robot. The separation can be varied and the smaller width makes the more overlap rezone of the scanning line. While the robot scans the selected workspace, Subgoal points generation procedure records the contact position and status between robot and wall of the workspace, and robot and obstacles. These positions are utilized as subgoals for sweeping path. The data structure of the Subgoal points is as follows;

```
Struct Subgoals {
        int   posion[DIMENSION];
        char status;
        char tracked_flag;
        int   obstacleNo;
} *SubgoalPoints;
```

Where position[DIMENSION] is the contact position between robot and wall or obstacle, and the status represents contact type which are LEFT_WALL, RIGHT _WALL, LEFT_OBSTACLE, and RIGHT_OBSTACLE. Each means contacting state between left side of robot and wall, right side of robot and wall, left side of robot and obstacle, and right side of robot and obstacle, respectively. In Subgoal points generation procedure, the robot moves form left top to right bottom in workspace and the orientation of the robot is fixed as 0 degree. The heuristic algorithm is as follows;

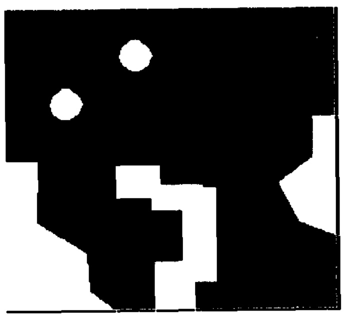

Fig. 2 Exemplary workspace

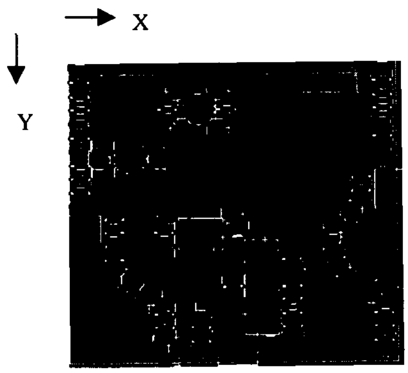

Fig. 3 Generated Subgoal points

For a given list ***Si*** which is i-th data entry of Subgoal points list, and robot initial position and orientation (x,y,q) = (0,0,0), Δx and Δy which are scanning widths along x-axis and y-axis, respectively.

Step 1: Set x, y, and i to 0, respectively. And find initial robot position in workspace by increasing x position with Δx and y position with Δy, respectively, while orientation of the robot is fixed as 0 degree. And record its position and status as LEFT_WALL to S_i. Mark tracked_flag of S_i as UNTRACKED. $i \leftarrow i+1$.

Step 2: Move the robot from current position along x-axis while y position is fixed. If the robot contacts with the wall, record its position and status as RIGHT_OBSTACLE or RIGHT_WALL according to contact type between robot and objects and obstacle number to S_i. Mark tracked_flag of S_i as UNTRACKED. $i \leftarrow i+1$.

Step 3: If the robot meets right bottom of the workspace, go to Step 5. Move the robot as Δy along y-axis while x position is fixed. And then move the robot toward left wall along x-axis. If the robot collides with obstacle, record its position and record status as LEFT_OBSTACLE and obstacle number. And move the robot toward left wall. When the robot meets the wall, record the position and status as LEFT_WALL to S_i. Mark tracked_flag of S_i as UNTRACKED. $i \leftarrow i+1$.

Step 4: Move the robot as Δy along y-axis and go to Step1.

Step 5: Record i as maximum number of Sobgoal points as i_max.

4. Subgoal points scanning procedure

This procedure links the recorded Subgoal points continuously under the condition that a Subgoal point should not be linked more than once and the linked path is never closed as possible.

Subgoal points scanning procedure finds Subgoal points to be continuously linked according to the status of the current Subgoal point. The scaning strategy consists of following 4 algorithms according to the collision status. The initial position of the robot is first entry in the list of the Subgoal points, and its status of the collision is LEFT_WALL since subgoal points are generated by scanning from left top to right bottom in the workspace.

CASE 1: When the status of the current Subgoal point is LEFT_WALL or LEFT_OBSTACLE.

Step 1: Search the next connecting SPs (Subgoal Point) of which status is RIGHT_WALL or RIGHT_OBSTACLE.

Step 2: Find minimum distance SP from current SP in Step 1. If its status is TRACKED and the path from current SG to the next SP does not collide with obstacle go to Step 6, otherwise go to Step 3.

Step 3: $y \leftarrow y - \Delta y$. If $y < 0$, then go to Step 5

Step 4: Search SP of which is RIGHT_WALL or RIGHT_OBSTACLE

Step 5: $y \leftarrow y + \Delta y$. If $y > y\_max$, then go to Step 7

Step 6: Find next SP to be linked which is minimum distance one between current SP and the next SP of untracked SP in Subgoal points data list. If the path from current SP to found SP collide with any obstacle, then Call collision free path finding procedure.

Step 7: If the number of the linked SP is the same as i_max, then go to END, otherwise go to link the SP and mark the tracked_flag of the linked SP as TRACKED. Go to the next Step in CASE1 or CASE 2 according to the status of the linked SP.

CASE 2: When the status of the current Subgoal point is RIGHT_WALL or RIGHT_OBSTACLE.

The process is the same as the CASE 1 except that in Step 1 and Step 4 RIGHT_WALL and RIGHT_OBSTACLE are replaced with LEFT_WALL and RIGHT_OBSTACLE, respectively.

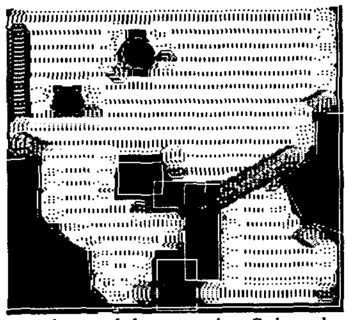

Fig. 4 Sweeping path by scanning Subgoal points (without collision avoidance)

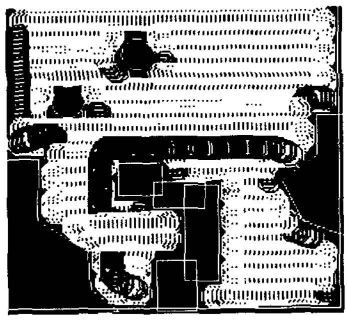

Fig. 5 Final resultant sweeping path (with collision avoidance)

5. Collision free path finding procedure

In Step 6, Subgoal point scanning procedure find a minimum distance SP from current SP in whole UNTRACKED SP of the subgoal point data list. If the path segment from the current SP to the found SP goes through obstacle, collision free planning procedure in [8] is applied.

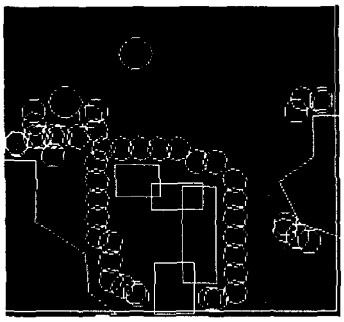

Fig. 6 Collision Free path

6. Simulation result

The proposed algorithm has been tested with cleaning robot of octagon type. In test-bed example shown in Fig. 2, there are 7 obstacles in workspace which is made by GUI interface. The workspace is 20m × 20m square meters. It is assumed that the robot can move omni-direction. The sequential of the proposed procedures are shown in Fig.3 through Fig. 6. Generated subgoal points for sweeping path are shown in Fig. 3. Fig. 4 shows that the sweeping path without applying a collision avoidance procedure. We can see that there were 7 collisions. For each collision the proposed collision algorithm was applied. Fig.6 represents that proposed collision free path generation procedure found collision free path successfully. A final sweeping path without collision is shown in Fig. 5. The planning time for this example is about 4 seconds.

Our planner uses AUTO CAD S/W as a architectural CAD tool. The interface S/W was programmed by Auto Lisp. Fig. 7 shows a real Architectural CAD draft. And Auto CAD interface module programmed by Auto LISP is shown in Fig. 8. We can setup parameters of working robot and capture some rezone as a workspace for smearing or cleaning job in this module. A final resultant sweeping path workspace on architectural CAD draft is represented in Fig.9. This figure represents that our planner can generate a successful sweeping path for real architectural CAD draft.

Fig. 7 Exemplary Architectural CAD draft

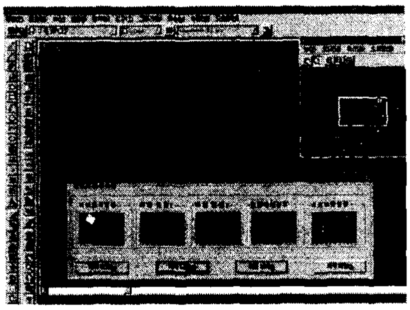

Fig. 8 Auto CAD interface menu of the proposed planner

Fig. 9 A resultant sweeping path for smearing robot on architectural CAD draft

7. Concluding remarks

A heuristic sweeping path planning algorithm for smearing robot was represented. The proposed algorithm consists of Auto CAD S/W interface module, Subgoal points generation module and subgoal point scanning module for sweeping path. And if there is a collision in path, proposed collision free path algorithm is applied.

The proposed planner can fill successfully whole workspace without deadlocking and collision even though any obstacles are in workspace. The planning times for motion paths with maximum coverage are in order of seconds. Our future work is to develop a planning algorithm working in environment with unknown or uncertain obstacles.

[Reference]

[1] Sherman. Y.T.Lang and B.Y. Chee, "Coordination of Behaviours for mobile robot floor cleaning," *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems,* Vol.2, p1236-1241., 1998

[2] Burhanpurkar. V.P," Real world application of a low cost high performance sensor system for autonomous mobile robot," *Proc. Intl. Conf. on IROS*, Munich, Vol.3, p1840-1846, 1994

[3] F.Yasutomi, D.Takao, M.Yamada and K.Tsukamoto," Cleaning robot control," *Proc. of IEEE Intl. Conf. on Robotics and automation,* Vol.3, p1839-1841, 1988

[4] T.Fukuda, S.Ito, N.Ota and Y.Abe," Navigation system based on ceiling landmark recording for autonomous mobile robot," *proc. of Intl. Conf. on Industrial Electronics, Control and Instrumentation,* vol. 3 , p1466-1471, 1993

[5] W.Li, " Perception-action behavior control of a mobile robot in uncertain environments using fuzzy logic," *proc. of IEEE/RSJ/GI Intl. Conf. on IROS,* Vol.1 , p439-461, 1994

[6] D. Kurabayashi, J.Ota, T.Ari and E.Yoshida, "Cooperative sweeping by multiple mobile robot ," *proc. of IEEE conf. on robotics and automation,* Vol.2, p1744-1749, 1996

[7] C.Hofner and G.Schnidit, "Path planning and guidance techniques for an autonomous mobile cleaning robot," *proc. of IEEE Conf. on IROS,* Vol.1, p610-617, 1994

[8] W.K.hyun and I.H.Suh, "A heuristic Collision-free path planning algorithm for robots," *IEEE IROS,* Vol.2, p837-845, 1995