

Path Planning Using Modified Gradient Method for Small-Sized Robots

Seo-Yeon Hwang
School of Mechanical Eng.
Korea University
Seoul, Korea
etoile02@korea.ac.kr

Jae-Bok Song
School of Mechanical Eng.
Korea University
Seoul, Korea
jbsong@korea.ac.kr

Abstract – This paper deals with the path planning method for small-sized robots in an indoor environment where the activity is limited. In mobile robotics, the gradient method has been the most popular path planning method. The purpose of the gradient method is to generate an optimal path which has no collision with obstacles. However, in case of the small-sized robots, the robot might pass through the limited areas under the tables and chairs. The conventional gradient method assigned the costs around the obstacles in order to calculate the intrinsic cost to make collision-free paths. This method provides an optimal path for the robots in wide areas. On the other hand, the robot in the narrow areas cannot figure out an optimal path because the high intrinsic costs hinder from generating gradient paths. This paper proposes a novel scheme how the robots figure out a path in narrow areas. It assigns the intrinsic costs simultaneously during the adjacency cost assignment step. Then the robot can figure out a safe path in narrow areas. The proposed method reduces the computational time significantly by assigning the costs to the minimum area. The intrinsic cost is also used to control the robot's velocity. The results show that the proposed method successfully works in various environments.

Keywords – Mobile robot, gradient method, path planning.

1. Introduction

Simultaneous localization and mapping (SLAM) has been one of the most important issues in mobile robotics. In recent years, despite the computational burden, researchers have applied the SLAM algorithm to small-sized robots using the microprocessors (e.g., ARM). Several small-sized robots, such as cleaning robots, adopted the SLAM algorithm and worked well in the indoor environments. In the past, the cleaning robots moved randomly in the environment without considering its position. If the cleaning robots can localize themselves, they can conduct coverage path planning to cover the entire area in the shortest time.

In mobile robotics, several coverage path planning methods using cellular decomposition were proposed and successful implementation was demonstrated in [1]. These methods can be executed successfully when the pose estimation is accurate. During the process of coverage path planning, the optimal path planning and path tracking

functions are necessary to move from the current position to the next area to be covered. It is difficult for a cleaning robot to cover the areas such as under the tables, desks, and chairs, because they have many narrow passages as shown in Fig. 1. A grid map built during navigation is usually used to generate a path, but it has a limitation to represent the environment accurately due to the limit of the cell size. As an example, if the width of the passage is represented smaller than its actual width due to the cell size, the robot cannot generate a path.



Fig. 1. Narrow areas under tables, desks, and chairs.

The gradient method [2] is one of the most popular path planning methods in mobile robotics. This method uses a navigation function to generate a gradient field in which the optimal path to the goal is computed at every point in the map. The intrinsic cost is assigned around the obstacles to generate a smooth and safe path. However, this method cannot generate a path in the environment consisting of only narrow areas because the high intrinsic cost makes it difficult to find the gradient paths. This situation frequently occurs when the robot size is as small as the cell size of a grid map.

This paper proposes the path planning method which can generate a path in both narrow and wide areas. While assigning the adjacency costs, the intrinsic cost is considered simultaneously to generate a path through the narrow areas. The whole process ends when there are no active cells which can create low-cost paths. The proposed method generates collision-free paths in the wide areas, and generates the safest path in the narrow areas. The small-sized robots can pass through the narrow areas even if many collisions occur, since they are basically equipped with contact sensors such as bumper sensors. After path generation, the intrinsic cost is also considered to determine the velocity of the robot. The small-sized robots cannot adopt velocity control algorithms, such as DWA, due to its short-range sensors and limited computational resources.

The rest of this paper is organized as follows. In section 2, the problems of the conventional gradient method in the narrow areas are introduced. Section 3 presents the details of the modified gradient method for the small-sized robots. Section 4 describes the motion control of a robot during the path tracking process using the pre-assigned intrinsic cost in section 3. The simulation results are presented in section 5 and conclusions are drawn in section 6.

2. Conventional Gradient Method

The gradient method [2] is used to generate an optimal path in real-time. It is based on the grid map which was previously built using range sensors or contact sensors. The intrinsic cost is assigned to the neighborhood of an obstacle to avoid it, whereas the adjacency cost is assigned to the entire grid map to generate a shortest path. Then the optimal path which has the minimum cost is generated using the gradient vectors among the grid cells.

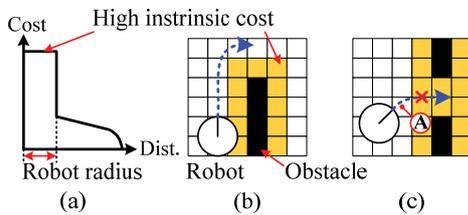


Fig. 2. Assignment of intrinsic cost and a problem in narrow area.

For a large-sized robot (e.g., a guide robot), 1 cell difference in the width of a passage does not affect the path planning based on the gradient method since the passage (e.g., a door) is large enough to pass through. However, for a small-sized robot (e.g., a cleaning robot), 1 cell difference in the width of the passage (e.g., the width between chair legs) often affects whether the robot can pass through the passage or not. Since the small-sized robots usually has limited computational resources, they usually adopt relatively larger grids (e.g., larger than 10 x 10 cm²) than their size (e.g., smaller than 30 cm in diameter) to reduce the computational burden. Therefore, it is not enough to represent the environment in detail for the small-sized robots when using the gradient method. To prevent collision with the obstacles, a high (infinite) intrinsic cost is assigned nearby the obstacles as shown in Fig. 2. From the intrinsic cost model shown in Fig. 2(a), a high intrinsic cost is assigned for the obstacle within a distance greater than the robot radius. In case of the small-sized robots, a high intrinsic cost should be assigned at least one cell from the obstacle to avoid the collision as shown in Fig. 2(b). However, some problems can occur in the path planning process based on the gradient method. The method cannot generate a path due to high intrinsic cost in the narrow area as shown in Fig. 2(c). Since the gradient method generates a path which has a minimum cost, it cannot continue to generate the next path at point A.

3. Modified Gradient Method

To cope with the problems mentioned in section 2, this paper proposes a path planning method which is also appropriate for the small-sized robots equipped with low-cost microprocessors. A simplified intrinsic cost model with a pre-calculated lookup table is used to speed up the computation. After assigning the intrinsic cost, the adjacency cost is assigned by a LPN algorithm [2], and the intrinsic cost is considered simultaneously during the assignment. When the adjacency cost is assigned from the goal point to the starting point, the algorithm starts to assign the adjacency costs to the remaining cells which can generate better paths. This step enables the algorithm to assign the adjacency cost only for the minimum area of the map, while the conventional gradient method assigns the adjacency cost to the entire map.

3.1 Cost Assignment

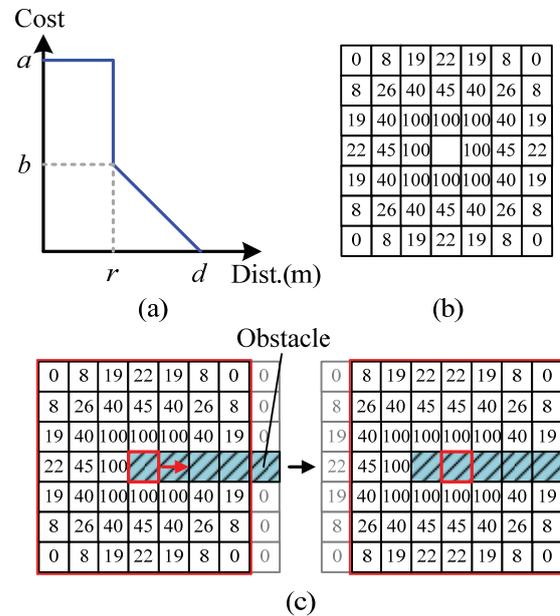


Fig. 3. Intrinsic cost model and its lookup table.

The simplified intrinsic cost model is represented in Fig. 3(a). To increase the computational speed, the lookup table of the intrinsic cost is pre-calculated before the path planning process. An example of the lookup table is shown in Fig. 3(b) with the values of $a=100$, $b=50$, $r=0.18$, and $d=0.40$. A data set from the lookup table are applied around the obstacles as shown in Fig. 3(c). Initially all empty cells in the grid map are assigned with a value of 0. The cost for each cell is updated only when the new value from the data set is larger than the previous one. This process lasts until the intrinsic cost is assigned to the entire grid map.

After assigning the intrinsic cost, the adjacency cost is assigned from the goal point to the starting point. At first, the algorithm starts to assign 1 to the cell which corresponds to the goal point and activates the cell. Then, based on the LPN algorithm, the algorithm adds 14 for the

diagonal neighbors and 10 for the other neighbors from the current active cell. For example, suppose that the active cell has a value of 1. Then, add 14 for the diagonal neighbor, and also add the intrinsic cost of the diagonal neighbor if it is not zero. The process of assigning the adjacency cost is shown in Fig. 4. ‘S’ and ‘G’ denote the starting and the point, respectively. Starting from the goal point, the proposed algorithm successfully assigns the adjacency cost progressively to the starting point.

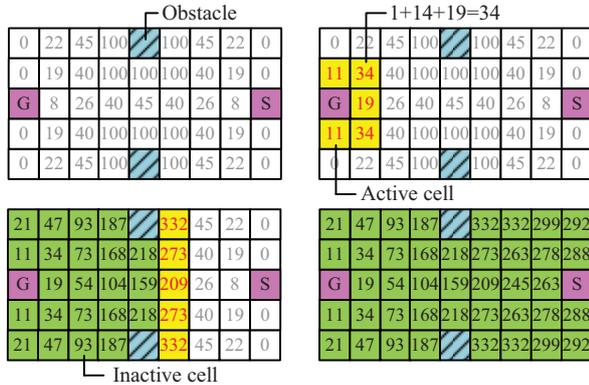


Fig. 4. Assignment of adjacency cost in narrow area.

3.2 Efficient Cost Assignment

In the conventional gradient method, the whole process of assigning the adjacency cost ends when there are no more active cells in the entire map. Thus, the computational time for path planning is proportional to the map size, but not the distance between the starting point and the goal point. Several methods have been proposed to reduce the computational burden. The first method is to complete the cost assignment and path generation when the adjacency cost reaches the starting point. This method always generates the shortest path even if there are many safer paths nearby. The second method, which is used in the proposed method, is to assign the adjacency costs to the remaining active cells which have the potential to generate a safe path. The collisions can frequently occur in the narrow area, which decreases the robot speed, so the collision-free paths which are adjacent to the narrow path are more efficient to the robot.

When the adjacency costs reach from the goal point to the starting point as shown in Fig. 5, the proposed method searches for the remaining active cells which can generate a safer path. At first, the distance between each active cell and the starting point is measured as follows:

$$d_1 = \min\{a, b\}, \quad d_2 = \max\{a, b\} \quad (1)$$

$$d = 14d_1 + 10(d_2 - d_1) \quad (2)$$

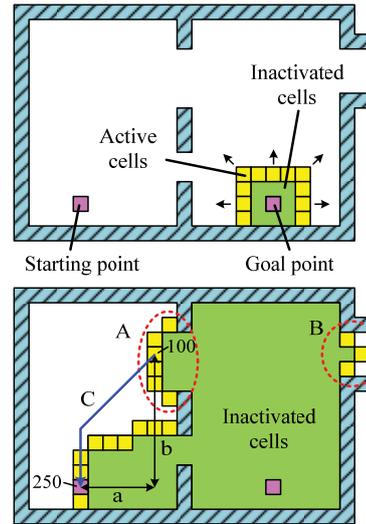


Fig. 5. Assignment of adjacency costs in minimum area.

where d_1 and d_2 are the smaller and larger value between a and b , and d is the resulting distance based on the LPN algorithm. The resulting distance in Eq. (2) represents a path to assign the minimum cost from the active cell to the starting point, as marked by ‘C’ in Fig. 5. If $d +$ (adjacency cost of the active cell) is smaller than the adjacency cost assigned to the starting point, then the active cell continues the assignment; otherwise, the cell is inactivated since it has no potential to generate a better path. In the example shown in Fig. 5, $a = 5$, $b = 9$, and $d = 110$ in Eq. (1) and (2). The adjacency cost of the corresponding cell is 100, and the minimum cost which can be assigned to the starting point is ‘210’ (= ‘110’ + ‘100’). Since the value is smaller than the pre-assigned value ‘250’ at the starting point, the active cells in ‘A’ continue to assign the cost. The active cells in ‘B’ stop the assignment and are inactivated due to the distance from the starting point. The process continues until there are no more active cells in the map. This method reduces the computational burden significantly.

3.3 Path Generation

The proposed method considers only the adjacency costs to generate a path, because the adjacency cost already includes the information of the intrinsic cost. At the starting point, the algorithm selects a cell which has the minimum adjacency cost among the 8 neighboring cells. The cell is registered to a path list, and its neighboring cells are considered in the next step. When the goal point is registered in the list, the process ends. The whole process of path generation is represented in Fig. 6. A high cost is marked in a light color and a low cost in a black color. The intrinsic cost is considered simultaneously to generate a path through the narrow areas while assigning the adjacency cost as shown in Fig. 6(c). The proposed method generates collision-free path in the wide areas, and also generates a path in the narrow areas as shown in Fig. 6(a).

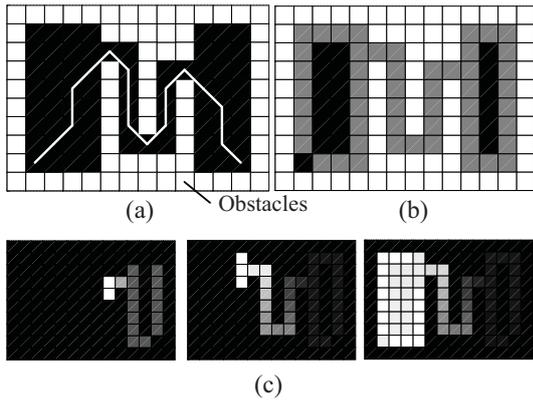


Fig. 6. Modified gradient method: (a) obstacles and a resulting path, (b) intrinsic cost, and (c) assignment of adjacency cost.

4. Motion Control

Due to the short-range sensors which are usually used for a small-sized robot, it is difficult to control the robot speed before it approaches the obstacle. Using the fact that the intrinsic cost is assigned around the obstacles, the robot can control its velocity according to the size of the passage. The robot velocity is determined by

$$v(i) = v_{\max} - k \cdot C_{\text{int}}(i) \quad (3)$$

where $v(i)$ is the robot velocity at the i -th point in the path list, v_{\max} is the maximum velocity, k is the scale factor, and $C_{\text{int}}(i)$ is the intrinsic cost at the i -th point in the list. An example is shown in Fig. 7.

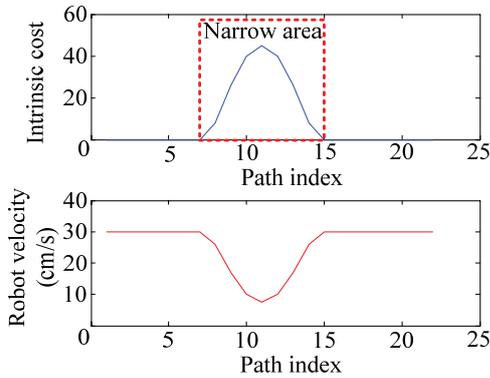


Fig. 7. Robot velocity in a narrow area.

For path tracking, a tracking control method proposed by Kanayama [3] was adopted. Arrival boundaries are used to track the path smoothly as shown in Fig. 8. Each corner point is used as a waypoint, and the robot sets the next waypoint when it reaches the current arrival boundary. As a result, the robot follows its path with a smooth movement.

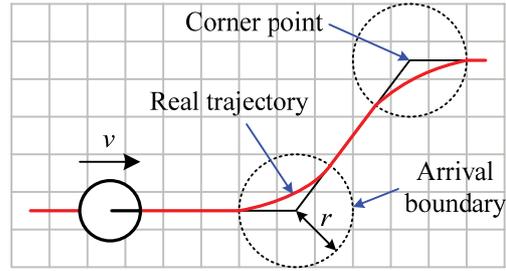


Fig. 8. Smooth path tracking.

5. Experimental Results

The results shown in Fig. 9 show that the proposed method successfully works in various environments. A high cost is marked in a light color and a low cost in a dark color for the intrinsic and adjacency costs. The proposed method generates no-collision paths in the wide areas, and generated the safest path in the narrow areas. When the wide areas exist near the narrow area, the proposed method selects the wide area.

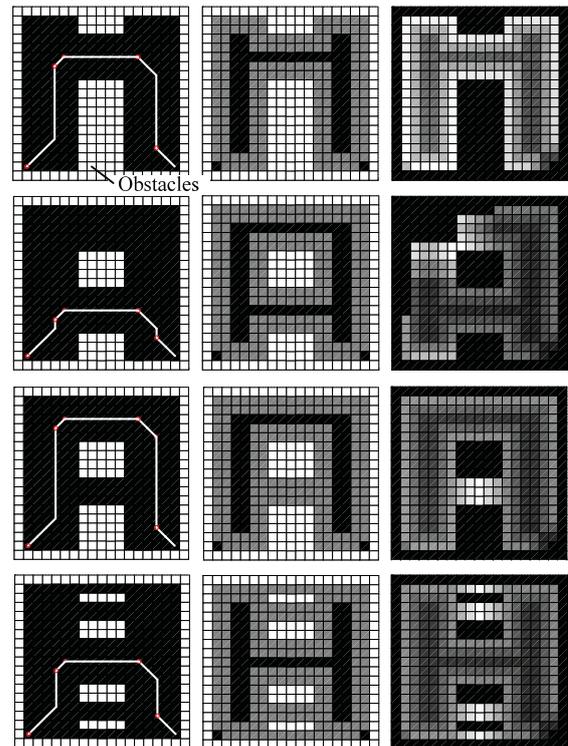


Fig. 9. Simulation results: obstacles and a resultant path (left), intrinsic costs (middle), and adjacency costs (right).

6. Conclusions

The proposed method generates an optimal path in both wide and narrow areas. It is useful for the small-sized robots which need to pass through the narrow areas such as under the tables, chairs, and near the corners. The experiments in various environments show the robustness of the proposed method. The method can work with the

low-cost processors in real-time, and can provide reliable performance in various situations.

Acknowledgement

This paper was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Knowledge Economy of Korea.

References

- [1] H. Choset, "Coverage of known spaces: the Boustrophedon cellular decomposition," *Autonomous Robots*, Vol. 9, No. 3, pp. 247-253, 2000.
- [2] K. Konolige, "A Gradient Method for Realtime Robot Control," *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vol. 1, pp. 639-646, 2000.
- [3] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, "A Stable Tracking Control Method for a Non-Holonomic Mobile Robot," *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1236-1241, 1991.