

Path planning for automatic guided vehicle with multiple target points in dynamic environment

Chang Kyu Kim¹, Huy Hung Nguyen¹, Dae Hwan Kim¹, Hak Kyeong Kim¹, Sang Bong Kim^{1*}

¹Department of Mechanical Design Engineering, Pukyong National University, Busan, 48549, Republic of Korea

Abstract. In path planning field, Automatic guided vehicle (AGV) has to move from an initial point towards a target point with capability to avoid obstacles. There are A*, D* and D* lite path planning algorithms in the path planning algorithm. This paper proposes a modified D* lite path planning algorithm using the most efficient D* lite among these algorithms. The modified D* lite path planning algorithm is to improve these D* lite path planning algorithm's weaknesses such as traversing across obstacles sharp corners, or traversing between two obstacles. To do this task, the followings are done. First, a work space is divided into square cells. Second, cost of each edge connecting current node to neighbor nodes is calculated. Third, the shortest paths from the initial point to all multiple target points are computed and the shortest paths from any target point to remaining target points including the goal point are computed by using Hamilton path. Fourth, a cost-minimal path is re-calculated as soon as the laser sensor detects an obstacle and make an updated list of target points. Finally, the validity of the proposed modified D* lite path planning algorithm is verified through simulation and experimental results.

1 Introduction

Automatic guided vehicles (AGVs) are used in many industrial fields for goods delivery, and robot cleaning, health care, etc. In particular, when looking for features in the entire map space, set multiple target points. The path planning of the AGV focuses on how much it costs to reach from the starting point to the target point in the workspace, and how to move to the shortest distance with avoiding obstacles on the route.

M. Likhachev and S. Koenig proposed Lifelong Planning A*(LPA*) algorithm [1]. LPA* is to find shortest paths from a given start point to a given goal point repeatedly. But, while the edge costs of a graph change, data of a point are added or deleted. S.M. Lavalley proposed the basics of path planning [2]. M.S. Alam and M.U. Rafique et al. proposed optimal path planning algorithm using particle swarm optimization technique [3]. The optimal waypoints computed by the planner lead the robot to the goal position effectively. D* lite algorithm among A*, D*, and D* lite path planning algorithms [1-3] is the most efficient in its path planning. Y.D. Setiawan et al. also proposed D* lite path

* Corresponding author : kimsb@pknu.ac.kr

planning algorithm that was widely used for AGV in static and dynamic environments, and it could replan from the current point when there was an obstacle blocking path [4]. The error of the path replanning algorithm is very small. However, there are weaknesses such as passing over the sharp edges of obstacles or passing between two obstacles.

In this paper, a modified D* lite path planning algorithm is proposed to improve the weakness of the D* lite path planning algorithm. The proposed modified D* lite algorithm is widely used for AGV in static and dynamic environments and it can replan from the current point when there is an obstacle blocking path. The validity of the proposed modified D* lite algorithm is verified through simulation and experimental results.

2 Path planning algorithm

There are A*, D*, and D* lite algorithms in the path planning algorithm. The D* lite path planning algorithm among them is the most efficient in path planning. However, the D* lite path planning algorithm has some weaknesses such as traversing across obstacles sharp corners, or traversing between two obstacles.

In this paper, a modified on D* lite path planning algorithm is proposed to improve these D* lite path planning algorithm's weaknesses and generates an optimal path from a starting point to a goal point through multiple target points with avoiding obstacles. The proposed path planning algorithm also controls an AGV with the laser sensor NAV-200 for detecting obstacles to move from the initial point to a goal point through multiple target points optimally. To do the abovementioned tasks, the proposed path planning algorithm is performed as follows:

Firstly, divide a work space into square cells. Each cell center is assigned as a node. The cell size is greater a bit than AGV size

Secondly, calculate cost of each edge connecting current node to neighbor nodes. It is a combination of the traversal cost of the two cells which it connects and length of the edge is defined as Fig. 1. The costs in horizontal and vertical directions are 1, the cost in diagonal direction is $\sqrt{2} \approx 1.414$ and if AGV moving in a direction detects obstacles, the cost is set ∞ as shown in Fig. 1.b.

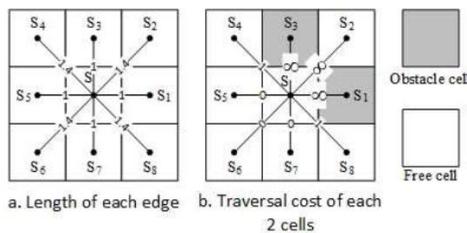


Fig. 1. Cost of each edge connecting current node to neighbor nodes.

Thirdly, compute the shortest paths from the initial point to all target points and the shortest paths from any target point to remaining target points including a goal point and make a Hamilton path with arranged target points based on a nearest neighbor principle. The Hamilton path is defined as follows. In graph theory, the Hamiltonian path is a single path in undirected or directed graph that visits each vertex exactly once. A graph that contains a Hamilton path is called a traceable graph. A graph is Hamilton-connected if for every pair of vertices there is a Hamilton path between the two vertices.

The following Fig. 2 shows the Hamiltonian cycle passing through all the vertices of a graph. The Hamiltonian graph must be a regular polygon, a complete graph of size 3 or

greater, and a circulating graph of size 3 or greater. If the graph is discontinuous or not circulating, it is not a Hamiltonian graph. Also, having a vertex with a graph size of 2 or less is not a Hamiltonian graph either.

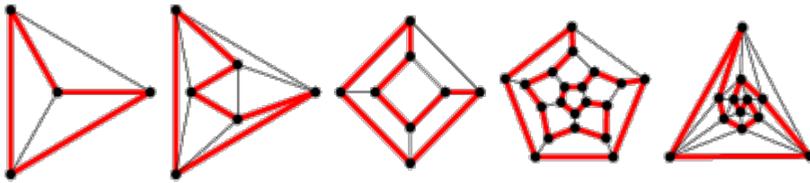


Fig. 2. Hamiltonian cycle.

Fourthly, re-calculate a cost-minimal path as soon as the laser sensor detects an obstacle and make an updated list of target points.

Finally, show simulation results to verify the effectiveness of the proposed path planning algorithm.

The followings show the proposed D* lite algorithms.

2.1 Comparison of D* lite algorithm and proposed D* lite path planning algorithm

In Fig. 3a, the D* lite path planning algorithm can perform incremental retrieval of information in a dynamic environment. Also, the number of lines of code is small and easy to understand. However, Sometimes, the D* lite path planning algorithm has some weaknesses such as traversing across obstacles sharp corners, or traversing between two obstacles. So, a suitable and optimal path planning algorithms should be chosen depending on the requirement of the system. To overcome these disadvantages, the obstacles could be avoided by setting multiple target points.

Therefore, Fig. 3b shows the proposed D* lite path planning algorithm code with four target points. By adding four target points to the existing D* lite path planning algorithm, it does not pass through sharp edges or between obstacles.

Fig. 4a shows the path planning of the D* lite algorithm as a node. As shown in the figure, the D* lite algorithm moves from a start point to a goal point in the shortest distance avoiding obstacles. However, if there exists a path in cell (4, G), the path hits the corner of the obstacle. This is very fatal to the movement of the AGV and can cause damage to the AGV body. Therefore, some measures are needed to prevent this path.

Fig. 4b shows the proposed modified D* lite path planning algorithm in each node. Set the target points which are cell (6,F), cell (3,E) in the middle of moving AGV from a start point to a goal point so that the position of the obstacle can be confirmed clearly. It no longer passes over the edge of the obstacle.

<pre> key(s) 01. return[$\min\{g(s), rh(s) + h(s_{start}, s), \min\{g(s), rh(s)\}\}$]; UpdateState(s) 02. if s was not visited before 03. $g(s) = \infty$; 04. if ($s = s_{goal}$) $rh(s) = \min_{v \in Succ(s)}\{c(s, v) + g(v)\}$; 05. if ($s \in OPEN$) remove s from OPEN; 06. if ($g(s) \neq rh(s)$) insert s into OPEN with key(s); ComputeShortestPath() 07. while($\min_{s \in OPEN}(key(s)) < key(s_{start})$ OR $rh(s_{start}) \neq g(s_{start})$) 08. remove state s with the minimum key from OPEN; 09. if ($g(s) > rh(s)$) 10. $g(s) = rh(s)$; 11. for all $s' \in Pred(s)$ UpdateState(s'); 12. else 13. $g(s) = \infty$; 14. for all $s' \in Pred(s) \cup \{s\}$ UpdateState(s'); Main() 15. $g(s_{start}) = rh(s_{start}) = \infty$; $g(s_{goal}) = \infty$; 16. $rh(s_{goal}) = 0$; $OPEN = \emptyset$; 17. insert s_{goal} into OPEN with key(s_{goal}); 18. forever 19. ComputeShortestPath(); 20. Wait for changes in edge costs; 21. for all directed edges (u, v) with changed edge costs 22. Update the edge costs $c(u, v)$; 23. UpdateState(u); </pre>	<pre> 12. obstacle = []; if Node_start = 13 % Assuming that an obstacle is detected at the node 13 (target point 1) obstacle = [17, 24, 27, 28]; % Node position of actual obstacle k_m = k_m + heuristic(Node, Node_last, Node_start); Node_last = Node_start; if Node_start = 32 % Assuming that an obstacle is detected at the node 32 (target point 2) obstacle = [34, 43, 44, 45, 46, 47]; % Node position of actual obstacle k_m = k_m + heuristic(Node, Node_last, Node_start); Node_last = Node_start; if Node_start = 62 % Assuming that an obstacle is detected at the node 62 (target point 3) obstacle = [76, 77]; % Node position of actual obstacle k_m = k_m + heuristic(Node, Node_last, Node_start); Node_last = Node_start; if Node_start = 80 % Assuming that an obstacle is detected at the node 80 (target point 4) obstacle = [79]; % Node position of actual obstacle k_m = k_m + heuristic(Node, Node_last, Node_start); Node_last = Node_start; </pre>
---	---

(a)

(b)

Fig. 3. Comparison of D* lite and proposed D* lite path planning algorithm; (a) D* lite algorithm; (b) Proposed D* lite path planning algorithm code.

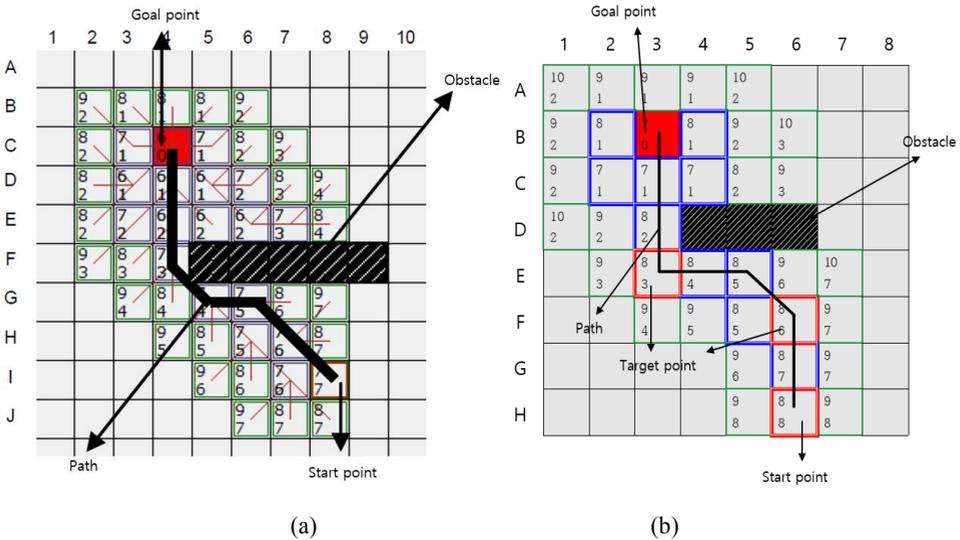


Fig. 4. Proposed D* lite path planning algorithm in each node; (a) D* lite algorithm path planning algorithm; (b) Proposed D* lite path planning algorithm.

3 Simulation and experimental results

Using the proposed D* lite path planning algorithm, a scenario of Fig. 5 below is set up to confirm the effect of setting four target points, avoiding obstacles, and choosing the shortest path. The test conditions of the proposed D* lite path planning algorithm are as follows : Space size is 3.5m x 7m, cell size is 0.5m x 0.5m, as shown in Fig. 5, a starting point is (7,1) and a goal point is (1,14).

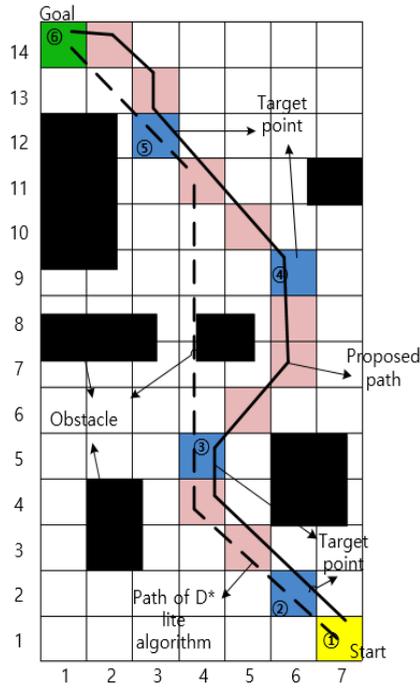


Fig. 5. Path planning in the scenario.

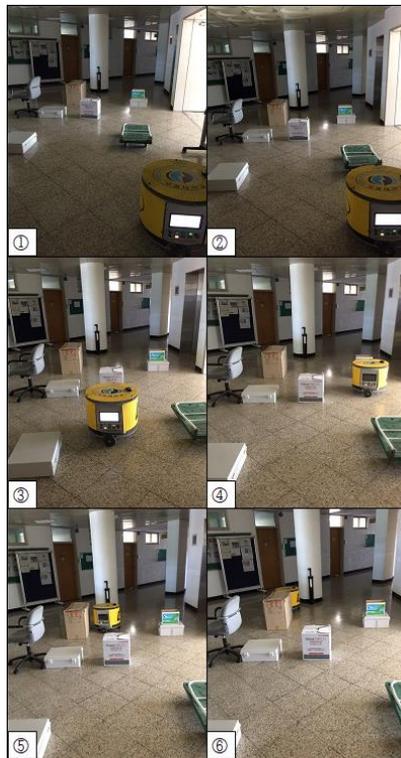


Fig. 6. Experimental results of the proposed modified path planning algorithm.

In the scenario of Fig. 5, the AGV moves from cell (7,1) with start point ① to cell (6,2) with a first target point ②. It detects an obstacle; cell (6,4) and (7,4). AGV keeps going a second target point ③ with avoiding obstacle. In this case, the D* lite path planning algorithm selects the path indicated by the dashed line in Fig. 5. This algorithm is unsafe. However, the proposed D* lite path planning algorithm moves safely to cell (6,9) of a third target point ④ with avoiding obstacles. When the AGV arrives at the cell (6,9), it detects the obstacles of the cells (7,11) and (6,11) and moves to cell (3,12) of a fourth target point ⑤. After the AGV moves to the position of the cell (3,12), it detects the cell (2,12) of the obstacle and reaches the goal point ⑥ with a safe route. As the results, the proposed D* lite path planning algorithm generates a continuous line in Fig. 5. Fig. 6 shows pictures showing the experiment results of the proposed modified path planning algorithm.

4 Conclusion

In this paper, a modified path planning algorithm based on D* lite algorithm was proposed to find the safest path for AGV to move from a starting point to a goal point through predefined multiple target points in dynamic environment. The proposed path planning algorithm was verified by simulation and experiment. The conclusion was described as follows: (i) a modified D* lite path planning algorithm was proposed to improve these D* lite path planning algorithm's weaknesses such as traversing across obstacle's sharp corners (a) and traversing between two obstacles (b). An optimal path from a starting point to a goal point through multiple targets points with avoiding obstacles was proposed. (ii) Simulation and experimental results could be used to identify the surrounding obstacles more clearly by placing multiple targets. So, the AGV was prevented from traversing between two obstacles, and target points was arranged optimally so that the AGV movement from the start point to the final target point was the minimum distance path. Although the D* lite path planning algorithm could give the shortest path, the AGV could collide with obstacles in real environment. Therefore, the proposed path might not be shortest but it is guaranteed to be safe in the proposed algorithm.

This work (Grants No. C0456375) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2017.

References

1. M. Likhachev, S. Koenig, *Lifelong planning A* and dynamic A* lite: The proofs. Technical report* (Atlanta, 2001)
2. S.M. Lavelle, *Planning algorithm* (Cambridge university press, 2006)
3. M.S. Alam, M.U. Rafique, *In Control, Automation and Robotics (ICCAR)*, 5 (2015)
4. Y.D. Setiawan, P.S. Pratama, J.W. Kim, D.H. Kim, Y.S. Jeong, S.B. Kim, *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (2014)