한국정밀공학회지 제 16 권 제 4 호 (1999 년 4 월)
Journal of the Korean Society of Precision Engineering, Vol. 16, No. 4, April 1999.

부분적 미지 환경에서의 이동로보트 경로계획

정학영*, 김기용*

Path Planning for a Mobile Robot in a Partially Unknown Environment

Hakyoung Chung* and Ki Yong Kim*

ABSTRACT

환경지도를 갖고 있는 이동로봇은 정확한 경로계획에 의하여 주행하게 된다. 그러나 주행 중 예상하지 못한 장애물을 만나는 경우 새로운 경로정보가 요구된다. 본 논문에서는 부분적인 환경정보를 갖고 있는 이동로봇의 경로계획기법을 제시한다. 경로계획은 전체경로계획과 지역경로계획으로 구분되면 전체환경을 노드와 아크로 표시한 네트워크 모델을 이용하여 수행된다. 경로계획시간과 메모리부담을 개선하기 위하여 네트워크 분할기법을 이용한 경로계획기법을 제안하였으며 지역경로계획에서는 정보가 변경된 부 네트워크에 대하여 경로계획을 수행하여 계산시간을 적게 소요하며 새로운 경로를 계산한다. 제안한 기법을 자동화 공장에서 주행하는 이동로봇에 적용하였으며 시뮬레이션과 실험을 통하여 제안한 기법의 성능을 보였다.

Key Words: Mobile robot(이동 로봇), Global path planning(전체경로계획), Local path planning(지역경로계획), Network decomposition(네트워크분할).

1. INTRODUCTION

An autonomous mobile robot in FMS should be able to navigate without human aids, performing fetch and carry tasks. Therefore planning a collision-free path to each station is one of the important requirements for the robot to perform its tasks. Ideally, a robot should move along optimal paths in a working area without colliding with obstacles and should not be trapped because of a peculiar shape of obstacles. In order to optimally navigate it and avoid being trapped, global path planning is necessary.

Several approaches to global path planning proposed in the past can be classified into following criteria: artificial potential method[1], configuration-space method[2], distance function method[3,4], grid search method[5,6], intelligent heuristic method[7] and network/graph method[8-11]. In an artificial potential field method, it is assumed that repulsive potential fields around the obstacles force away the robot and an attractive potential field around the goal attracts the robot. A path is determined by tracing the attractive potential field. In a configuration-space method, a robot is supposed to be a single point in a configuration space and path planning is to move the robot without colliding with nearby obstacles. Distance method expresses obstacle avoidance in terms of the distances between potentially colliding parts. Path planning problems is formulated as problems in optimal control. In a grid search method, path planning is implemented by connecting the grids that are not occupied by obstacles. In intelligent heuristic method, fuzzy logic, generic algorithm, and neural network are adopted to solve the obstacle avoidance path. Path planning using

* 서울산업대학교 제어계측공학과

network/graph method is to find a chain of arcs where each two subsequent arcs in the path share a common node, and the path is determined as the shortest path between two nodes. Many algorithms have been developed to solve single-origin shortest path and all-pairs shortest path problems. Dijkstra's algorithm is a typical single-origin shortest path algorithm that finds a shortest path by using a tree-structure. Many variations have been performed to reduce the amount of calculation and memory resources. A* algorithm is one of the most popular heuristic algorithms, which uses the best-first search[8]. Bi-directional search algorithm computes suitable paths by using forward search from an origin and backward search from a destination[9]. All paths between all pairs of nodes can be computed by the dynamic programming method[10,11]. We use the network/graph method in order to compute all paths between all stations of FMS and the network can be obtained from a grid-type world model [12].

A concept of decomposition in which a network is divided into several subnetworks has been proposed[10-15]. The path computations are performed for individual subnetworks, and the solutions are recombined to obtain the paths of the original network. The mobile robot moves following the global paths. However, the robot may meet unexpected obstacles while it moves along the path. Then a revised path, local path has to be provided to the robot. We propose local path planning for real-time computation. In local path planning, a new path is obtained by integrating global path data and path data computed in the subnetwork where the unknown obstacles are detected. Our goal is to generate global optimal paths in a network that represents its working area and to compute a local path in real time, which guarantees that the robot reaches a destination following the revised path.

The proposed concept has two advantages; 1) global optimal paths between all pairs of nodes in a network can be calculated much faster 2) local path planning is performed by using the information of its working area. That is: local path calculations need to be done within one or two sectors instead of the whole world model.

In Section 2, the proposed method for global path planning is presented and local path planning algorithm is presented in Section 3. In Section 4, some simulation results are shown and some experimental results with our two-wheel driven indoor robot in our FMS is presented in Section 5.

2. GLOBAL PATH PLANNING

Global path planning is for a robot to efficiently navigate following an optimal path, while it is not to be caught by traps. Calculating optimal paths is the most important step in global path planning. We propose a global shortest path algorithm using decomposition technique.

General shortest path algorithms provide optimal paths between any pair of nodes. But these algorithms impose a heavy computational burden. Some researchers have studied on network decomposition method in order to reduce the computational burden[13-15]. The algorithms require much less memory and offer a faster solution time than those without decomposition. Also, the resulting solution is identical to the one obtained when the problem is solved without decomposition, assuming the solution is unique. We decompose the network into several subnetworks in order to reduce the computational burden and perform local path planning in real time. The proposed global path algorithm is described in the sequel.

A network G is defined as a pair $G = (N, A)$ where N represents a node set and A an arc set. A cost of an arc is associated with an elapsed time for the robot to move from a node to another node. A path is a chain of arcs where each two subsequent arcs in the path share a common node and global path in the proposed algorithm is defined as the shortest time path between two nodes. To formally describe the problem, it is convenient to define the cost, minimum cost, and successor matrices associated with a given network. The cost matrix $C = [c_{ij}]$ is the matrix whose ij th element c_{ij} equals the arc cost connecting the node n_i to the node n_j. The minimum cost matrix $D = [d_{ij}]$ is the matrix whose ij element d_{ij} represents the total cost of the shortest path from n_i to the node n_j. The successor matrix $S = [s_{ij}]$ is the matrix whose ij th element s_{ij} is the number of the node which is adjacent to n_i in the shortest path from n_i to n_j. Using these representations, the shortest paths problem is to calculate D and S from given C. Upon calculation of D and S, for any pair of nodes, the minimum cost of the shortest path is obtained from D and the path is obtained from S.

A subnetwork $G_i = (N_i, A_i)$ is defined as a network formed by a subset of nodes $N_i \subset N$ together with the set of all arcs of A that join any two nodes of N_i. In the network $G = (N, A)$ a set of nodes $N_c (N_c \subset N)$ can be found whose removal, along with all the incident arcs from G, generates completely separated subnetworks and it is called a cut-set.

The algorithm decomposes a network into a number of smaller subnetworks with a minimum number of cut-nodes and a size limit for each subnetwork. Two theorems of [11] are necessary for guarantee the optimality of the path planning using network decomposition.

Theorem 1 of [11] describes a sufficient condition that warrants the shortest paths solution of a subnetwork calculated independently of other subnetworks, and Theorem 2 of [11] gives a methodology to calculate the shortest paths between nodes of different subnetworks provided that the shortest paths of each subnetwork are known. The theorems are as follows and the proofs are omitted for simplicity.

Theorem 1: Given $G = (N, A)$ and a subnetwork $G_i = (N_i, A_i)$ of G, define $\overline{G_i} = (\overline{N_i}, \overline{A_i})$ such that $N_i \cup \overline{N_i} = N$ and $A_i \cup \overline{A_i} = A$, and $N_i \cap \overline{N_i} = N_x$, i.e., N_x is a cut-set between G_i and $\overline{G_i}$. $\overline{G_i}$ represents the complement of G_i only overlapped by the cut-set. The shortest paths between any two nodes of the shortest paths are in G_i, whether all intermediate nodes of the shortest paths are in G_i or not, can be obtained by considering only the subnetwork G_i if conditional shortest paths between all pairs of N_x in $\overline{G_i}$ are known.

Theorem 2: Let $G_k = (N_k, A_k)$ and $G_t = (N_t, A_t)$ be subnetworks of $G = (N, A)$, joined by a cut-set N_x, i.e., $N_k \cap N_t = N_x$. Then for $n_k \in N_k$, $n_t \in N_t$, and n_k, $n_t \notin N_x$, the following holds,

$$d_{kt} = \min_{n_x \in N_x} \{d_{kx} + d_{xt}\}$$

and

$$d_{tx} = \min_{n_x \in N_x} \{d_{tx} + d_{xk}\}$$

where d_{kt} is the minimum distance from n_k to n_t.

Based on the two theorems, the algorithm calculating the global paths by optimal decomposition is outlined here; and a detailed description of the algorithm can be found in [10,11]. It consists of the following steps.

Global path planning

1. Define a cut node set to decompose the network into small subnetworks, and obtain the decomposed cost and initial successor matrices.
2. Compute the shortest paths between all pairs of nodes in the cut-sets, considering all possible paths in the network.
3. Compute the shortest paths for each subnetwork by Rosenthal' algorithm[16].
4. Compute the shortest path between each pair of nodes that are in different subnetworks.

In the decomposed network, any path from a node in a subnetwork to a node in another subnetwork must include at least one intermediate node in the cut-set. This does not exclude the possibility that the shortest path from a node in a subnetwork to another node in the same subnetwork passes through nodes in any number of other subnetworks, as long as there is at least one cut-node on a path between two nodes in different subnetworks.

3. LOCAL PATH PLANNING

The mobile robot moves following the paths determined by global path planning. However, the robot may meet unexpected obstacles while it moves along the global path. Then a revised path, local path should be provided to the robot in real time and it should be guaranteed that the robot reaches a destination along the path. In the proposed local path planning, a new path is obtained by integrating global path data and path data computed in the subnetwork where the unknown obstacles are detected. For the local path planning, two lemmas and a theorem are first presented. Lemma 1 describes a sufficient condition that warrants the shortest paths in the subnetwork computed last in calculating the shortest paths between all pairs of nodes in the cut-sets. Lemma 2 describes a condition that the results of global path planning can be employed in local path computation. Theorem 1 shows a methodology to calculate a local path between a robot and a destination of a different subnetwork,

Lemma 1: Computing the shortest paths between all pairs of nodes in the cut-sets, the shortest paths between all pairs of nodes in the subnetwork calculated last are obtained.

Proof of Lemma 1: The shortest paths between all pairs of the cut-set nodes are obtained by repeated shortest paths computations, beginning with subnetwork 1 and continuing through the last subnetwork m. The computations may be done in any order by the shortest path algorithm, e.g., Rosenthal's algorithm [16]. The first computation for subnetwork 1 results in incomplete shortest paths of the cut-nodes since it is obtained by considering only subnetwork 1. The second computation for subnetwork 2 still results in incomplete shortest paths of the cut-nodes since it is obtained by considering only subnetwork 1 and subnetwork 2. This procedure is continued. At the last subnetwork m, complete shortest paths of the cut-set nodes are obtained by considering subnetworks 1 through $m-1$ and the results actually becomes the shortest paths between all pairs of nodes in subnetwork m by theorem 1 of [11].

Lemma 2: Given $G=(N,A)$ and a subnetwork $G_k=(N_k,A_k)$ of G, where some costs of arcs of G_k are changed large, define $\overline{G_k}=(\overline{N_k},\overline{A_k})$ such that $N_k \cup \overline{N_k}=N$ and $A_k \cup \overline{A_k}=A$, and $N_k \cap \overline{N_k}=N_x$, i.e., N_x is a cut-set. For $n_t \in \overline{N_k}$, $n_x \in N_x$ if a path from n_x to n_t does not contain $n_k \in N_k$, d_{xt} is equal to $\hat{d}_{xt}$. Otherwise, d_{xt} is not less than $\hat{d}_{xt}$, where d_{xt} is the minimum distance from n_x to n_t and ^ represents that the distance is determined without considering the changes.

Proof of Lemma 2 : Unless the path from n_x to n_t contains $n_k \in N_k$, the changes do not have influence on the path. Therefore $\hat{d}_{xt}$ is the minimum distance of the path. Since the proof in the case that the path contains $n_k \in N_k$ is self-evident, we omit the details.

Theorem 1: Let $G_k=(N_k,A_k)$ and $G_t=(N_t,A_t)$ be subnetworks of $G=(N,A)$, joined by a cut-set N_x, i.e., $N_k \cap N_t = N_x$. Also, assume that some costs of the arcs of G_k are changed larger. Then for $n_k \in N_k$, $n_t \in N_t$, $n_k, n_t \notin N_x$, and for the path that is determined by the following equation, if the path from n_x to n_t does not contain the nodes in G_k, then d_{kt} is the minimum distance from n_k to n_t.

$$d_{kt} = \min_{n_x \in N_x} \{d_{kx} + \hat{d}_{xt}\}$$

where d_{kt} is the minimum distance from n_k to n_t and ^ represents that the distance is determined without considering the changes.

Proof of Theorem 1: A path from a node of G_k to another node of G_t must pass through at least a node in N_x because of the definition of cut-set. Also, for $n_{x_i}, n_{x_j} \in N_x$, where the path from n_{x_i} to n_t does not contain the nodes in G_k and the path from n_{x_j} to n_t contains the nodes in G_k, if $d_{kx_i}+\hat{d}_{x_i t} < d_{kx_j}+\hat{d}_{x_j t}$, then

$$d_{kx_i}+\hat{d}_{x_i t} < d_{kx_j}+d_{x_j t}$$

according to Lemma 2. In computing the shortest path from n_k to n_t, if a minimum is taken over all nodes in N_x and the path from n_x to n_t does not contain the nodes in G_k, then d_{kt} is the minimum.

Based on the two lemmas and one theorem, the algorithm calculating the local path is now given.

Local path planning: This procedure is executed when the robot encounters an unexpected obstacle.

1. Define a new node n_k that is located at the robot position. Assume that n_k is located in G_k. n_t is a

destination node, and n_x is a node in the cut-set.

2. Calculate the costs of the arcs of G_k considering the obstacle.
3. Compute the shortest paths between all node pairs in the cut-set considering all possible paths in the network. This may be done in any order, but the computation for G_k must be performed at the end.
4. If the destination node is located in G_k, the shortest path to the destination is obtained from the results of Step 3 by Lemma 1. Otherwise, continue the following steps.
5. Compute the following equation. If the path from n_x to n_t in the path of d_{kt} does not contain the nodes in G_k, d_{kt} is the minimum distance by Theorem 1. Otherwise, continue the following steps.

$$d_{kt} = \min_{n_x \in N_x} \{ d_{kx} + \hat{d}_{xt} \}$$

where d_{kt} is the minimum distance from n_k to n_t, d_{kt} is the result from Step 3, and $\hat{d}_{xt}$ represents the minimum distance predetermined by global path planning

6. Compute the shortest path according to Step 3 and 4 in global path planning.

4. SIMULATION

The mobile robot moves following the global paths. However, the robot may meet unexpected obstacles while it moves along the path. Then a revised path, local path has to be provided to the robot. In global path planning, new optimal path is computed by considering the whole network. But in the proposed local path planning, the optimal path can be calculated by considering the subnetwork where the obstacle is located. The simulations have been performed for comparing the computation time. Three square-type grid networks are used in our simulations, each of which has 49, 529, and 1089 nodes and has 84, 1012, and 2112 arcs, respectively. Each grid network is decomposed into four same sized subnetworks. Fig 1 shows the 49-node network. A junction of two lines represents a node. A node set on bold lines represents a cut node set since the removal of

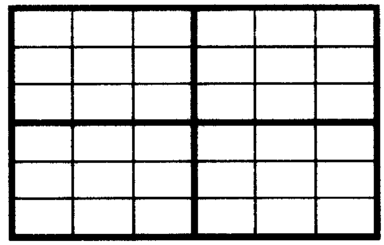


Fig. 1 A 49-node network divided into four subnetworks

the cut nodes, along with all the incident arcs from the network, generates completely separated the subnetworks. The algorithms are implemented in C++ and executed on a 120MHz Pentium PC. Three cases are employed in the simulation. Firstly, no network decomposition is performed (Case A). Secondly, network decomposition is performed (Case B). Finally, the network decomposition and the proposed local path planning is performed (Case C). In Case C, it has been assumed that arc costs in a subnetwork were changed.

Table 1 shows the computation time. Case C proves the proposed algorithm is more efficient than other two cases. The simulation results show that our proposed algorithm reduces the amount of computational burden. Also, the proposed algorithm guarantees the mathematically optimal solution. Fig. 2 shows a network represents

Table 1. Computation Times (seconds)

Order	Case A	Case B	Case C
49	0.037	0.041	0.027
529	35.15	11.04	7.18
1089	298.52	78.65	51.12

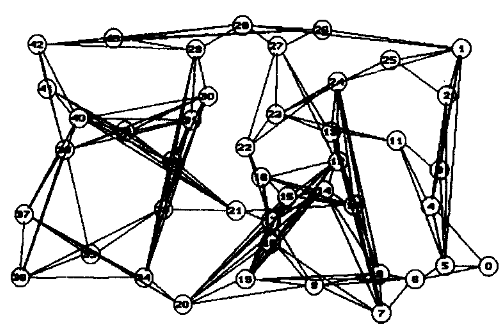
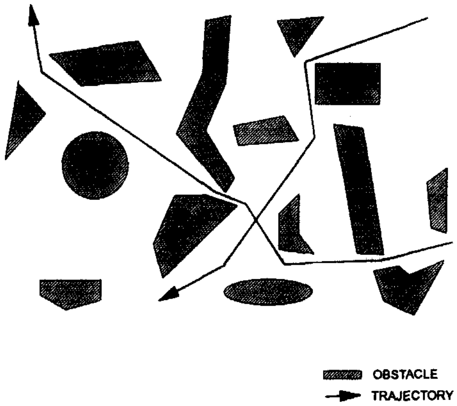


Fig. 2 A network model.



(a) Global path planning

(b) Local path planning

Fig. 3 An example of path planning.

the environment of Fig. 3. Fig 3 (a) shows paths of global path planning between two node sets and Fig. 3 (b) shows paths of local path planning. The revised path are computed when the mobile robot meets unknown obstacles.

5. EXPERIMENTS

The proposed path planning has been implemented on our mobile robot (see Fig. 4), whose role is to transport materials in FMS. The mobile robot is a two wheeled, four castered indoor robot. The size of the robot

Fig. 4 The mobile robot.

is $780(W) \times 780(L) \times 1150(H)mm$. The robot has a maximum linear speed of 1 m/sec and a maximum angular velocity of 128°/sec. 24 sonar sensors are sensors are installed for world modeling. In order to overcome sensor inaccuracy, we proposed a multi-sensor system in determining the values of certainty grid [12]. Upon the calculation of the grid type world model, the network for path planning is obtained from the world model. We designed a fuzzy controller to implement path following of a physical mobile robot [17]. The basic idea of the controller is to pass through the nodes on the optimal path with curve, minimizing a deviation from the path.

Fig. 5 and Fig. 6 show experimental results. The grid in black in Fig. 6 represents an unknown obstacle. The

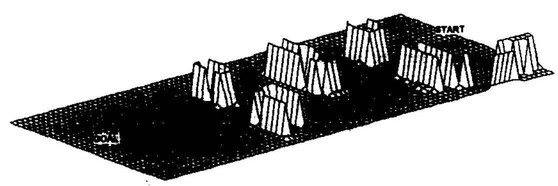


Fig. 5 An example of global path planning

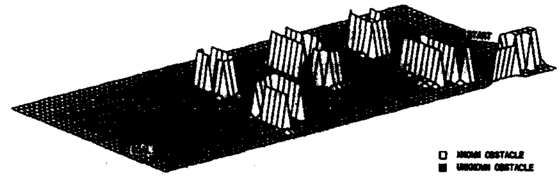


Fig. 6 An example of local path planning

robot determines a shortest path from a start position to a goal position by global path planning. The mobile robot navigates following the global path until the robot meets the unknown obstacle. The local path is determined in real time by computing the shortest path of a subnetwork and combining the path with the stored global path.

Therefore when the robot encounters an unknown obstacle, it is able to continue to navigate to a destination without halting.

6. CONCLUSION

This paper presents a new methodology of optimal path planning for an autonomous mobile robot, which performs task-oriented navigation in FMS. The optimal path algorithm divides a working area into small size sectors so as to reduce the computational burden. Global paths which is the shortest path in a network composed of nodes and arcs are computed by off-line calculation before the robot starts performing its tasks. Local path planning which is necessary when the robot encounters unknown obstacles was presented. Local path calculations are done within one or two subnetworks instead of the whole network by the concept of the global path planning. Therefore local paths are computed fast. Our experiments have confirmed that a physical robot optimally navigates to its destination in partially unknown environment by employing the proposed algorithm.

ACKNOWLEDGEMENTS

This paper was supported by the research fund of Seoul National University of Technology.

REFERENCES

1. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Proc. 1985 IEEE Conf. of Robotics and Automation*, pp. 500-505, 1985.
2. R. A. Brooks, " Solving the find-path problem by good representation of free space," *Trans. on System, Man and Cybernatics*, vol. SMC-13, no. 3, pp. 290-297, 1983.
3. E. G. Gilbert, and D. W. Johnson, "Distance functions and their application to robot path planning in the presence of obstacle," *IEEE Trans. on Robotics and Automation*, vol. RA-1, no. 1, pp. 21-30, 1985.
4. 전홍주, 이범희, "거리 함수를 이용한 로봇의 시변 장애물 회피 동작계획," 전기학회논문지, 41권, 12호, pp 1429-1439, 1992.
5. A. Zelinsky, "A mobile robot exploration algorithm," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 6, pp. 707-717, 1992.
6. Bogdan Kreczmer, "Path planner for a car-like robot applying multi-layer grid map of the environment," 3rd *IFAC Symposium on Intelligent Autonomous Vehicles, pp 324-329*, 1998.
7. 김경훈, 조형석, "최적 경유점 선택방법을 이용한 이동로봇의 반응적 주행," Proc. Of the 12th KACC, pp227-230, 1997.
8. P. E. Hart, N. J. Nillson, and B. Rafael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Sys. Sci. and Cyb. SSC-4*, pp.100-107, 1968.
9. H. Ikeda, M. Y. Hsu, H. Imai, "A Fast Algorithm for Finding Better Routes by AI Search Techniques, " *Vehicle Navigation & Information Systems Conference Proceedings*, pp.291-296, 1994.
10. 정학영, 이장규, "네트워크 분할기법을 이용한 자율주행운반체의 경로계획 (II)-최적경로 계산," 전기학회논문지, 제 45권, 7호, pp.1045-1049, 1996.
11. J. G. Lee, W. G. Vogt, and M. H. Mickle, "Calculation of the shortest paths by optimal decomposition," *IEEE Trans. System, Man Cybernetics* SMC-12(3):410-415, 1982.
12. 정학영, 이장규, "네트워크 분할기법을 이용한 자율주행운반체의 경로계획 (I)-환경모델 구성," 전기학회논문지, 제 44권, 12호, pp.1675-1679, 1995.
13. J. G. Lee, W. G. Vogt, and M. H. Mickle, "Optimal decomposition of large-scale network," *IEEE Trans. Systems Man Cybernetics* SMC-9(7):369-375, 1979.
14. T. C. Hu, A decomposition algorithm for shortest paths in a network. *Oper. Res.*, vol. 16, no. 1, pp. 91-

102, 1968.
15. G. Mills, " A decomposition algorithm for shortest-route problem," Oper. Res., vol. 14, pp.279-291, 1966.
16. A. Rosenthal, "On finding shortest paths in non-negative networks," *Discrete Math.* 10:159-162, 1974.
17. H. Chung J. G. Lee, "Optimal path planning and navigation for an autonomous mobile robot," *Fifth IFSA World Congress*, 1258-1261, 1993.