

논문 2006-43SC-3-5

분포 밀도를 이용한 이동 로봇의 최단 경로 설정

(Minimum Path Planning for Mobile Robot using Distribution Density)

곽 재 혁*, 임 준 흥*

(Jaehyuk Kwak and Joonhong Lim)

요 약

이동로봇의 경로 설정과 장애물 회피에 관한 문제는 이동로봇을 제어하기 위한 기본적인 문제로서 지금까지 많은 연구가 진행되어 오고 있다. 다양한 센서들로부터 얻어진 정보는 로봇이 진행하기 위한 경로에 대한 결정과 장애물에 대한 정보를 제공해준다. 제한 조건이 존재하는 경우에서 그 상황에 알맞은 방법들이 대부분이며, 다른 상황을 해결하기에는 어려움이 있다. 또한, 최적화된 경로에 대한 좋은 결과를 만들기 위해서는 복잡한 계산 과정을 통해서 시간 지연이 발생하게 된다. 이와 같은 처리 방법은 움직이는 장애물의 회피와 같이 빠른 처리를 필요로 하는 상황에서 단점으로 작용한다. 본 논문에서는 장애물과 이동 가능한 경로들의 영역을 작은 영역들이 분포하는 공간으로 변형하고 이와 같이 구성된 공간에서 밀도와 연결 상태를 통해 이동 로봇이 진행 할 수 있는 최단 경로를 결정한다. 제안된 방법을 통해 작은 영역들은 자신의 번호를 가지게 되며, 주변 영역들의 개수를 통해 이동 경로를 결정하게 된다.

Abstract

Many researches on path planning and obstacle avoidance for the fundamentals of mobile robot have been done. Informations from various sensors can find obstacles and make path. In spite of many solutions of finding optimal path, each can be applied to only a constrained condition. This means that it is difficult to find a universal algorithm. A optimal path with a complicated computation generates a time delay which cannot avoid moving obstacles. In this paper, we propose the algorithm of path planning and obstacle avoidance for mobile robot. We call the proposed method Random Access Sequence(RAS) method. In the proposed method, a small region is set first and numbers are assigned to its neighbors, then the path is selected using these numbers. It has an advantage of fast planning and simple operation. This means that new path selection may be possible within short time and that helps a robot to avoid obstacle in any direction. When a robot meets moving obstacles, it avoids obstacles in a random direction. RAS method using obstacle information from variable sensors is useful to get minimum path length to goal.

Keywords : 경로설정, 영역확장, 이동 로봇, 임의순서

I. 서 론

최근 로봇에 대한 연구가 활발해 지면서 이족 보행과 새로운 형태의 이동 로봇에 대한 연구가 활발히 진행되고 있다. 이동 로봇의 경로 설정과 장애물 회피는 기본적인 문제이며 해결해야 하는 중요한 요소이다. 지금까지 많은 논문들에서 이러한 문제를 해결하기 위한 방법들이 제시되고 있다. 시작점과 도착점사이의 최단

경로를 빠른 시간 안에 찾는 방법은 주변 환경이나 이동로봇의 크기, 또는 장애물의 형태 등에 의해서 다양한 방법들이 있다. 즉, 하나의 실험에 최적화된 결과는 다른 환경에서는 최적화된 결과를 보장할 수 없게 된다. 이동 로봇이 작업자의 도움이 없이 스스로 주변 상황에 대한 인식과 위치의 파악을 통해 주어진 작업을 수행하기 위해서 다양한 종류의 센서와 경로 설정에 대한 방법을 수행한다^[1-4]. 초음파, 적외선, 레이저, 카메라와 같은 센서들을 통해서 장애물을 파악하고, 엔코더를 통해서 위치를 알 수 있으며, 더욱 정확한 위치를 찾기 위해 표식과 같은 방법을 사용한다. 이동 로봇은 제한적인 조건에서의 작업뿐 아니라 주변 상황이 정해지지

* 정희원, 한양대학교 전자공학과
(School of Electrical Engineering and Computer Science, Hanyang University)
접수일자: 2005년10월27일, 수정완료일 2006년3월30일

많은 동적 환경에서도 작업을 수행하여야 한다. 이를 위해서 이동 로봇의 제어는 여러 가지 작업을 독립적으로 수행할 수 있는 계층적인 구조로 되어 있다^[2]. 경로 설정의 과정에서 다양한 방법들이 연구되고 있으며, 최적화된 경로를 찾기 위해 복잡한 계산을 필요로 하고 있다^[4]. 경로 설정에 관한 문제는 주변에 대한 환경 정보에 따라서 구분할 수 있으며, 많은 이론들이 환경에 대한 정보를 사용하여 최적화된 경로를 찾는 방법에 대해서 언급하고 있다. 주변 환경에 대해서 정보가 없는 곳에서는 지도에 대한 정보를 사용하여 경로를 찾는 방법이 가능하다^{[5][8]}, 이 방법에서는 오래 시간을 통한 전처리 과정을 통해 얻어진 결과를 사용하여 재 경로 설정 시 빠른 시간을 보장하지만, 환경이 바뀌면 다시 전처리 시간에 많은 노력을 해야 한다. 이동 경로에 대한 다양한 연구는 최단 경로 길이를 찾기 위해 수행된다. Skeleton, cell decomposition과 potential field 방법들이 경로설정에 관한 연구 주제로 연구되어 왔다^[6].

경로를 설정하는 알고리즘은 매우 다양하지만, 그래프 이론을 기반으로 하는 A* 방법과 Dijkstra's 알고리즘을 이용한 방법들이 빠른 경로 설정과 최적화된 경로 설정을 위해 사용되고 있다. 이 두 가지 방법은 시작점과 도착점까지의 이동 방향과 가중치를 조절함으로써 최단 경로를 찾을 수 있다. Open List, Closed List로 표현되는 A* 알고리즘은 탐색방법(heuristic)을 사용하여 G(N), H(N)의 두 가지 값을 이용하여 경로를 찾아낸다. 여기서 G(N)은 방향을 찾는 데 이용되는 실제 값으로서 거리 값이나, 고저차등을 사용하며, H(N)은 하나의 영역에서 도착점까지의 이동 거리 값을 나타내며 경험적인 방법으로 찾아낸다. 따라서, 모든 영역에 대해서 주변영역들의 가중치 값(F)을 구하고 최소 값을 갖는 영역을 찾아가는 방법으로 시작점과 도착점사이의 경로를 찾아낸다. 이 과정에서 장애물과 같이 이동이 불가능한 영역은 제외시킨다. A* 알고리즘에 대해서 간단하게 순서를 정리하면 다음과 같다.

- a. 시작점을 중심으로 주변 영역에 대해서 F를 구한다.
- b. 리스트에 대한 이동을 한다.
- c. 가장 낮은 F값을 갖는 영역을 다음 단계의 시작점으로 한다.
- d. 도착점이 나올 때까지 a-c의 단계를 반복한다.

Dijkstra's 알고리즘은 노드와 선분으로 표현된 그래프에 거리 값을 사용하여 경로를 찾는 방법이다. 시작점으로부터 연결되어 있는 모든 선분에 대해서 거리 값과 가중치 값을 더하고 이 값이 변하지 않을 때까지 주변 노드에 대해서 계산을 수행한다. 이와 같은 방법으로 모든 노드에 대해서 값이 주어지고 이 값을 통해서 경로를 설정하게 된다. 일반적으로 H(N)값을 사용하지 않는 이 방법은 모든 방향에 대해서 도착점까지 도달할 때까지 계산을 수행하여야 한다. 따라서 A* 방법에 비해 더 많은 계산 시간이 소요된다. 그러나, A* 알고리즘의 경우 넓은 영역에 대해서 도착점의 거리가 먼 경우 가장 가까운 영역을 찾는데 거의 무한에 가까운 계산을 해야 하는 경우가 발생한다. Dijkstra's 알고리즘은 하나의 시작점으로부터 무조건 확장해가는 형식이기 때문에 이와 같은 문제는 발생하지 않는다.

본 논문에서 제안한 임의접근순서확장법(Random Access Sequence, RAS)은 경험적인 방법을 사용하지 않기 때문에 경로 설정을 더 빠른 시간 안에 해결할 수 있다. RAS의 밀도값은 이동 로봇의 크기에 대해서 실제로 주행 가능한 이동 경로를 찾게 해준다. 다른 알고리즘과의 경로 비교와 다양한 환경에서의 실험을 통해 성능을 평가한다.

II. 임의접근순서확장법

임의접근순서확장법(RAS)은 이동로봇이 이동할 수 있는 영역과 장애물, 벽 그리고 지형장애물등과 같이 이동이 불가능한 영역으로 분할한 뒤 목적지까지의 경로를 찾아내는 방법이다^[9]. I장에서 보이고 있는 A* 알고리즘의 과정에서 ad에 해당하는 과정만을 통해서 주어진 알고리즘에 의해 모든 영역에 번호가 부여되며, 이 과정이 끝나면 그 결과를 사용하여 도착점으로부터 시작점까지의 경로를 추정해 간다.

2.1 임의접근순서확장법의 개념

공간을 표현할 수 있는 형태는 매우 다양하게 존재한다. 점, 선, 그리고 다각형 형태가 그것이며, 그림 1에서는 정사각형을 사용하여 빈 공간이 없고 이웃한 공간이 최소의 수를 가진다. 그림 1에서 나타내고 있는 것은 임의접근순서확장법에 대한 기본 개념도이며, 장애물을 제외한 로봇의 이동 가능한 영역을 작은 영역(region)들로 분할해 놓은 것이다. 그림 1은 한 번의

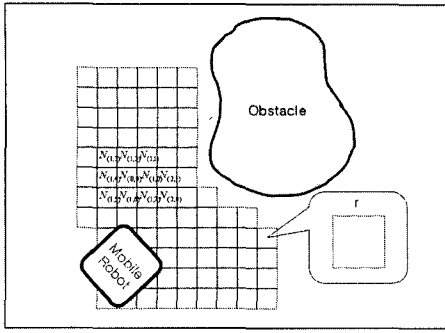


그림 1. 기본 개념도
Fig. 1. Basic scheme.

길이가 r 인 정사각형 형태의 영역 분할 형태이다.

RAS는 그림 1과 같이 분포되어 있는 영역들과 이웃한 영역들이 가지고 있는 번호들의 관계를 통해서 경로를 찾을 수 있다. 임의접근순서확장법에 의해서 구해지는 결과는

$$F = (k, p, N, d, n) \tag{1}$$

과 같다. 식(1)에서 k 는 각 영역들에 붙여진 번호(ID)를 나타내며 $p = (x, y)$ 로서 중심점의 좌표 값이다. N 은 하나의 영역에 연결되어 있는 주변영역들의 개수이다. 이 값은 영역을 나누는 형태에 의해서 변경되며 동일한 크기를 사용할 경우와 영역의 크기를 가변적으로 한 경우에 따라 달라진다.

$n = (n_0, n_1, \dots, n_{N-1})$ 은 중심점과 연결되어 있는 영역들의 번호를 나타내고 있다. d 는 밀도값으로서 다양한 목적으로 이용될 수 있다. 밀도 값 d 는

$$d = \sum_{i=0}^n \sum_{j=0}^n F_{i,j}(N) \tag{2}$$

과 같이 정의한다. $F_{i,j}(N)$ 는 그림 1에서 나타내고 있듯이 계층적으로 되어 있는 접근영역의 개수를 나타낸다.

F 를 구하는 방법은 시작점으로부터 임의의 방향으로 회전시켜 각 영역에 번호를 부여할 수 있으며, 이때 시작점 또한 임의로 설정할 수 있다. 그러나 경로 설정은 목적지로부터 시작점까지의 최단 경로수가 구해지기 때문에 이동로봇의 초기 위치를 사용하여 시작점을 설정한다.

RAS를 사용하여 공간은 그림 2와 같이 각각의 번호로 표현된 식별자(ID)를 갖는 영역으로 결정된다. 여기

60	51	50	49	48	47	46	45
53	52	31	30	29	28	27	26
54	32	■	14	13	12	11	25
55	33	15	4	3	2	10	24
56	34	16	5	8	1	9	23
57	35	17	6	7	8	■	22
58	36	18	■	19	20	21	44
59	37	38	39	40	41	42	43

그림 2. 식별자를 이용한 공간 표현
Fig. 2. Expression space using ID.

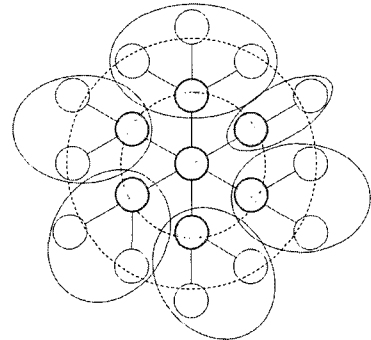


그림 3. 영역들 간의 관계 표현
Fig. 3. The relationships among regions.

서 0은 시작점을 나타내며 실선으로 표현된 원은 시작점으로부터 단계적으로 확장되는 것을 나타낸다. 그림 2에서 검은색으로 표현된 식별자가 없는 공간은 바로 장애물이 위치한 공간이다. 그림 2와 같이 나누어진 공간에서 경로를 찾는 방법을 설명하기 위해 그림 2는 다음의 그림 3과 같이 바뀌서 설명할 수 있다.

그림 2에서 실선으로 표현되었던 단계별 구분은 그림 3에서는 점선으로 나타내어진다. 식(1)의 F 에서 n 으로 표현된 하나의 영역에 연결된 영역들은 그림 3에서 실선으로 표현한 것이다. 즉 시작이 되는 위치를 어두운 영역으로 표현하고 조건에 맞는 영역들을 찾아서 연결을 해 놓은 것이다. 첫 단계에서 발견되어진 영역들은 원점과 연결된 영역들이며 이러한 영역들은 그림 2에서 알 수 있듯이 서로의 중심점이 최단 거리를 갖는 관계를 갖는다. 다음 단계인 실선으로 표현된 영역에서 마찬가지로 시작점이 되는 어두운 영역들과 주변의 영역들은 역시 최단 거리의 관계를 갖는다. 이와 같이 각각의 영역들은 전단계의 시작점과 최단 거리의 관계를

유지하면서 영역을 확장해 나가게 된다. 즉 $F_k(p)$ 를 k 번째 영역의 중심점이라고 할 때, 모든 영역은 $\|F_k(p) - F_{k-1}(p)\| = d_c$ 인 관계를 갖는다. 이동 경로는 바로 목표점으로 지정된 영역으로부터 전단계의 시작점을 찾아가는 방법을 사용하여 $F(k)$ 를 역으로 추종해 간다. 만약 식 F 에 대해서

$$F_0 = (0, p_0, 6, d, (1, 2, 3, 4, 5, 6)) \quad (3)$$

$$F_1 = (1, p_1, 3, d, (7, 8, 9)) \quad (4)$$

와 같이 구해진다고 가정하면, 그림 2에서와 같이 (7,8,9)의 영역이 갖는 시작점은 1이 된다. 또한 영역 1이 갖는 시작점은 이동 로봇의 시작점인 0이 된다. 따라서, 원점으로부터(7,8,9)의 영역으로 가는 경로는 영역 1을 통해서 가는 경로가 된다.

2.2 경로 설정 변수

임의접근순서확장법에서 경로 설정을 결정하는 가장 중요한 변수는 로봇의 크기(S_R), 영역의 크기(S_p)와 밀도 값(d)이다. 임의접근순서확장법에 의해서 얻어진 경로는 최단 경로수를 갖는 하나의 경로만 구해진다. 이 경우 임의접근순서확장법에 의해서 이동 경로가 구해지더라도 이동 로봇이 가지는 크기에 의해 벽면이나 장애물의 사이와 같은 곳은 지나갈 수 없는 경우가 발생한다. 이와 같은 문제는 영역 크기 r 을 이동 로봇이 가지는 넓이보다 충분히 큰 값을 사용하여 해결 할 수 있으나 이와 같은 큰 값의 적용은 세밀한 제어를 어렵게 하며 실제 최단 거리보다 더 큰 이동 경로를 가지게 된다. 그러나 r 값을 작게 설정하면 영역수의 증가로 인해 경로 설정의 시간 지연이 더 커지게 된다.

밀도값은 식(2)에서와 같이 하나의 영역 주변에 있는 장애물을 제외한 번호가 부여된 영역의 개수로서 표현된다. 그림 4는 밀도값의 중요성을 표현하기 위해서 맨 하단 방법^[10] 과 같이 정사각형 형태로 장애물과 이동 가능한 영역을 표현한 것이다. 그림4 에서 표시된 시작점으로부터 목적지로 가기 위해서 이동로봇은 최단 경로인 장애물 1과 2의 사이를 결정하게 된다. 그러나, 이동로봇은 이 경로를 따라서 이동할 수 없다. 그 이유는 장애물 사이의 간격이 이동로봇의 크기보다 작아서 이동 로봇이 이동할 수 없기 때문이다. 이러한 문제는 임의접근순서확장법의 초기단계인 순서부여과정에서

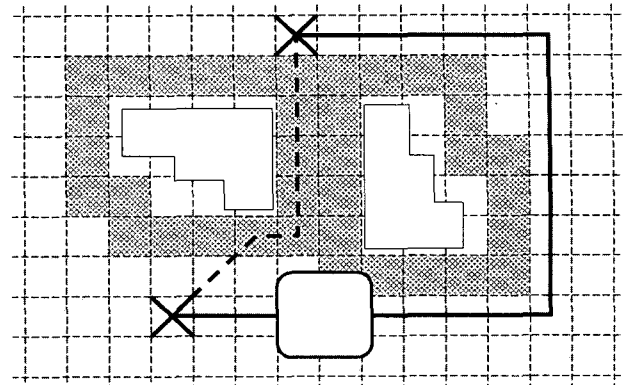


그림 4. 밀도값을 이용한 경로 수정
Fig. 4. Path planning using density.

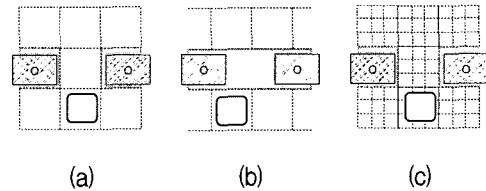


그림 5. 장애물에 대한 영역의 위치
Fig. 5. positions of regions for obstacles.

밀도값을 설정함으로써 해결할 수 있다. 그림 4에서 회색으로 표시된 영역들은 모두 밀도값이 8인 위치를 나타내며 회색 점으로 표현된 영역은 밀도값이 8보다 작은 영역이고, 흰색 영역은 장애물을 나타내고 있다. 즉, 밀도값의 제한치를 두어 조건을 만족하는 경우에만 번호를 부여한다. 이 과정을 통해서 이동 경로는 로봇이 진행 가능한 위치를 찾을 수 있으며 영역의 중심위치를 따라서 이동하게 되는 로봇은 이를 통해 장애물이나 벽면에 부딪치게 되는 상황을 피할 수 있다.

밀도값을 사용해야 되는 또 다른 이유는 그림 5(b)와 같이 영역이 놓이는 위치가 항상 장애물을 피해서 생성되지 않기 때문이다.

그림 5의 (a)와 같이 영역의 크기가 큰 경우 상황에 따라 장애물 사이로 경로 설정이 가능한 영역이 위치하게 될 가능성도 있지만, 실험을 통해 그림 5(b)와 같이 주변 장애물들에 의해서 장애물로 인식되는 경우가 더 많이 발생한다는 사실을 알 수 있다. 이와 같은 사실은 동적 환경에서의 임의접근순서확장법을 사용할 경우 경로수의 증가를 가져와 결과적으로 더 많은 이동 경로 거리를 가지게 된다. 따라서, 그림 5(c)와 같이 작은 영역으로 분할하고 밀도값을 사용함으로써 로봇이 이동 가능한 영역을 더 넓힐 수 있다.

앞에서 설명한 바와 같이 임의접근순서방법은 목적

지를 찾기 위해서 경험적으로 영역을 찾아가지 않는다. 따라서, A*나 Dijkstra's의 알고리즘에 비해서 더 빠른 경로설정 시간을 갖는다. 임의접근순서확장법이 갖는 특징은 시작점과 목적지까지 도달 경로가 존재한다면 반드시 경로를 찾을 수 있으며, 이 때 결정되는 경로는 반드시 하나이며 최단 경로수를 갖는다. 영역에 대한 전처리 단계가 필요 없기 때문에 빠른 경로 설정이 가능하다. 또한 밀도값을 조절함으로써 이동로봇이 진행할 수 있는 공간을 확보하고 실제 환경에서의 적용을 가능하게 해준다.

III. 임의접근순서확장법의 구현(Implementation)

3.1 순서부여과정

평면을 표현할 수 있는 다각형들은 다양하게 존재한다. 그림 6은 임의접근순서확장법에서 공간을 표현할 수 있는 영역들의 형태에 대해서 나타내고 있다.

그림 6의 (a,b,c)는 계산 속도를 향상시키기 위해서 행렬형태로의 변형이 가능하며 그림 6(d)의 경우 행렬 형태가 아니더라도 구현이 가능한 형태이다. 또한 방향의 선택이 여러 가지가 존재하기 때문에 좀 더 세밀한 경로 설정이 가능하다. 그러나, 탐색 방향의 증가로 인해 빠른 경로 설정은 어렵다. 본 논문에서는 행렬과 놓인 형태가 일치하고 모든 공간을 표현할 수 있는 그림 6(b)의 정사각형 형태를 사용하여 장애물 영역과 이동 가능한 영역을 정의하였다. 이와 같은 공간에서는 다음과 같이 매우 단순한 식을 사용하여 행렬의 위치를 결정할 수 있다.

$$f_{m,n} : \begin{bmatrix} m \\ n \end{bmatrix} = \left(\frac{1}{r} \right) P \quad (5)$$

식(5)에서 $P = (x, y)$ 로서 이동이 가능한 영역을 나타내며 r 은 영역의 크기를 나타낸다. 그림 2와 같이 분할된 영역들에 번호를 부여하고 식 (1)의 F 를 구하

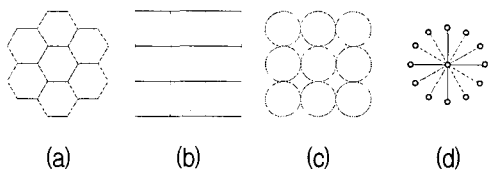


그림 6. 영역 형태
Fig. 6. Various shapes of regions.

기 위해서 시작점으로부터 식별자를 구하면,

1st step

$$\{(k_0, p_0, N_0, d_0, (n_0, n_1, \dots, n_{N_0}))\}$$

$$n_0 = k_1$$

2nd step

$$\left\{ \begin{array}{l} (k_1, p_1, N_1, d_1, (n_0, n_1, \dots, n_{N_1})) \\ \vdots \\ (k_{h_1}, p_{h_1}, N_{h_1}, d_{h_1}, (n_0, n_1, \dots, n_{N_{h_1}})) \end{array} \right\}$$

3rd step

$$\left\{ \begin{array}{l} (k_{h_1+1}, p_{h_1+1}, N_{h_1+1}, d_{h_1+1}, (n_0, n_1, \dots, n_{N_{h_1+1}})) \\ \vdots \\ (k_{h_2}, p_{h_2}, N_{h_2}, d_{h_2}, (n_0, n_1, \dots, n_{N_{h_2}})) \end{array} \right\}$$

⋮

nth step

$$\left\{ \begin{array}{l} (k_{h_{m-2}}, p_{h_{m-2}}, N_{h_{m-2}}, d_{h_{m-2}}, (n_0, n_1, \dots, n_{N_{h_{m-2}}})) \\ \vdots \\ (k_{h_{m-1}}, p_{h_{m-1}}, N_{h_{m-1}}, d_{h_{m-1}}, (n_0, n_1, \dots, n_{N_{h_{m-1}}})) \end{array} \right\}$$

(6)

과 같은 순서로 진행된다. 식(6)에서 각 단계의 마지막 번호를 나타내는 h_j 는

$$h_j = \sum N_{(j-1)th \ step} \quad (7)$$

과 같다. 식(6)의 첫 번째 단계는 이동 로봇의 시작점으로부터 주변의 영역에 대한 탐색과정이다. 이 과정에서 장애물이 없는 영역은 번호가 부여되며 장애물이 놓여져 있는 영역은 제외된다. 두 번째 단계에서는 첫 번째 단계에서 부여된 가장 낮은 번호를 갖는 영역에서부터 주변에 대한 탐색이 시작된다. 마찬가지로 장애물이 없는 영역에 대해서만 번호가 부여되며 번호는 전단계의 마지막 영역의 번호가 갖는 번호에 이어서 시작된다. 이때 장애물이 있는 영역 뿐 만 아니라 이미 번호가 부여된 영역에 대해서도 새로운 번호를 부여하지 않는다. 이러한 과정을 통해서 작업 공간의 모든 영역에 대해서 번호를 부여하는 과정이 진행된다. 식(6)으로부터 모든 영역에 식별자를 생성하고 모든 영역에 대해서 F 를 구하게 되면

$$T_M = \begin{bmatrix} f_{0,0} & f_{0,1} & \dots & f_{m,0} \\ f_{0,1} & f_{1,1} & \dots & f_{m,1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{0,n} & f_{1,n} & \dots & f_{m,n} \end{bmatrix} \quad (8)$$

$$f_{m,n} = (s, F)$$

$$s = \begin{cases} 0 & : F = \emptyset \\ 1 & \end{cases}$$

과 같이 전체 공간을 나타내는 T_M 을 구하게 된다. 식(8)에서 f 는 그림 1에서 표현된 각 영역들을 표현하고 있다. s 는 (0,1)의 값을 가지며 0일 경우 이 영역은 번호가 부여되지 않은 장애물과 같은 영역을 표현하며 이때에 F 는 아무런 값을 가지지 않는다.

3.2 경로설정방법

식(8)의 T_M 으로 표현된 공간에서 경로를 찾는 방법은 다음 관계에 의해서 설명 되어질 수 있다.

$$F_0(n) = (1, 2, 3, 4, 5, 6) \quad (9)$$

식 (9)가 의미하는 것은 영역 (1,2,3,4,5,6)들의 시작점이 바로 0이라는 것이다.

마찬가지로 $F_k(n) = (n_0, n_1, \dots, n_{N-1})$ 이 나타내는 것은 영역 $(n_0, n_1, \dots, n_{N-1})$ 의 시작점은 k 라는 것이다. 식(6)과 식(7)로부터

$$n_0 = \{ \sum (F_0(N), \dots, F_{k-1}(N)) \} + 1 \quad (10)$$

의 관계를 갖는다. 식 (10)을 이용하여

$$n_0 < \sum_{i=0}^k F_i(N) \leq n_N \quad (11)$$

인 것을 알 수 있다. 이와 같은 방법을 사용하여 이동 로봇의 경로는 위에서 구한 F 에서 k 와 N 만을 사용한 $F'(k, N)$ 를 통해 결정한다. $F'(k, N)$ 를 이용하여 경로를 결정하는 방법은 다음과 같다.

표 1에서 K 가 나타내는 것은 영역의 번호를 나타내며, N_i 는 앞에서 구한 각 영역이 갖는 인접한 영역들의 개수인 N 을 나타낸다. 이를 통해 얻어진 (R_0, R_1, \dots, R_n) 은 영역이 가지고 있는 번호의 조합으로 이루어지며 각각의 위치를 이용하여 원하는 목적지로 이동할 수 있다. 제안한 방법을 통해 얻어진 경

표 1. 경로 설정 알고리즘

Table 1. Algorithm for path planning.

1:	while ($R_{n+1} \neq \text{origin}$)
2:	if ($R_n \leq \sum_0^K N_i$)
3:	$R_{n+1} = K,$
4:	end
5:	end

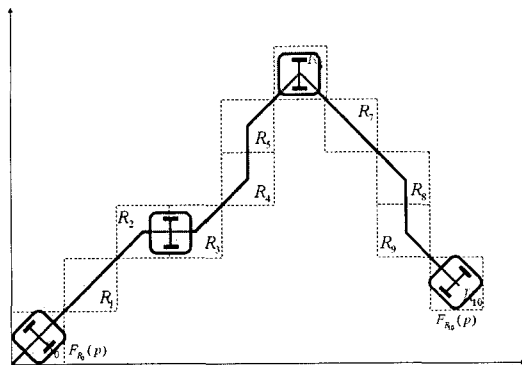


그림 7. 이동 로봇의 이동 경로

Fig. 7. A mobile robot path planning.

로는 시작점과 목적지까지의 최단 경로 수만을 선택한다.

RAS를 통해 정의된 영역에서 결과적으로 이동 로봇이 진행해야 될 전체의 경로를 결정한다. 아래의 그림7은 시작점 p_0 에서부터 도착점 p_{10} 으로 가는 경로에 대해서 표현하고 있다. 그림 7.에서 알 수 있듯이 결정된 경로에 해당하는 영역의 중심점을 따라서 이동하게 된다.

IV. 실험 및 결과

RAS의 성능평가를 위해서 실제 이동 로봇을 사용하여 좁은 영역에서 경로주행을 하였으며, 모의실험을 통해서 다른 알고리즘과의 비교 및 복잡한 환경에서의 경로 설정 시간을 측정하였다. 그림 8은 실험에 사용된 이동 로봇이며 크기는 11cm x 10cm x 7cm이며 무선 통신을 위해 블루투스를 사용하였다.

그림 9는 RAS를 이용한 경로설정법의 실험 환경을 보여주고 있다. 검은색으로 표현된 장애물과 이동로봇의 초기 위치와 크기를 검출하고 임의로 지정한 목적지를 표현하고 있다. 카메라와 프레임 그래버를 통해 들

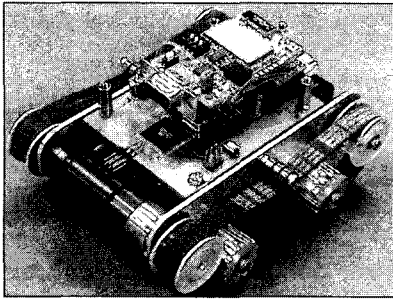


그림 8. 이동 로봇
Fig. 8. Mobile Robot.

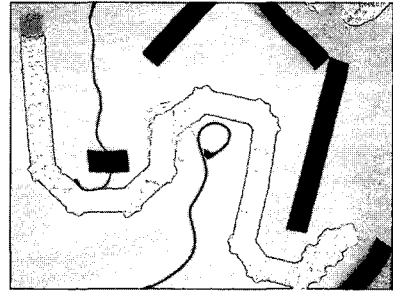


그림 11. 로봇의 실제 이동 경로
Fig. 11. The moving path of Mobile Robot.

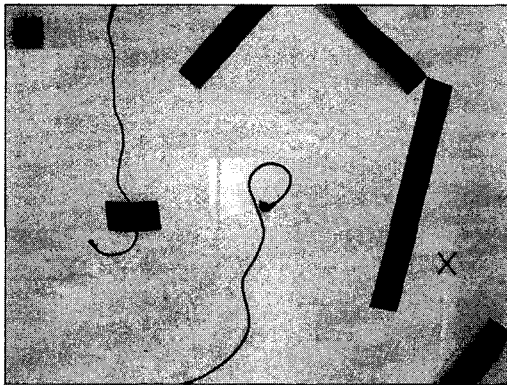
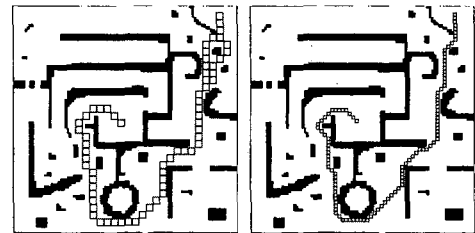


그림 9. 실험환경
Fig. 9. Experimental Environment.



(a) 영역크기 16 (b) 영역크기 8

그림 12. 경로설정 방법 비교
Fig. 12. Comparison with other algorithms.

표 3. 경로 설정 시간
Table 3. Elapsed time for path planning.

알고리즘	영역크기	경로 수	소요시간(s)
A*	16	57	0.00634
	8	106	0.07968
Dijkstra's	16	57	0.02025
	8	106	0.64825
RAS	16	57	0.00378
	8	106	0.00417

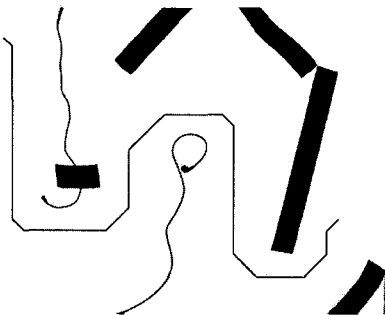


그림 10. 모의실험에서의 경로 설정
Fig. 10. Simulation of Path Planning.

어은 영상은 (640pixels x 480pixels)의 크기를 가지며 실제 지도의 크기는 (2.4m x 1.6m)이다. 그림 10은 실제 지도와 같은 지도로 모의실험을 한 결과이다. 실제 실험에서 검출된 이동로봇의 크기는 32pixels이며, 따라서, 영역의 크기는 18로 결정하고 밀도값을 8로 설정하였다. 그림 11은 이동 로봇의 실제 이동 경로를 연속적으로 표현한 것이다.

다양한 실험 환경을 구현하기 위해서 모의실험을 하였다. 로봇의 크기와 영역의 크기를 다양하게 변경하여 경로 설정의 시간과 측정하였으며, P4 2.8GHz, 512M의 성능을 갖는 윈도우 기반에서 수행되었다.

그림 12는 A*,Dijkstra's 알고리즘과 본 논문에서 제안하고 있는 RAS를 같은 환경에서 목적지까지의 경로를 나타내고 있다. 지도의 크기는 (512,512)이며 영역의 크기는 8,16의 두 가지로 경로를 결정하였다. 그림 12(b)의 경우 세 가지의 방법을 통해 구해진 경로가 일치하는 것을 볼 수 있다. 그림12(a)의 경우 경로 수는 같으나 전체적인 이동거리에서 차이가 발생한다. RAS의 밀도값은 다른 알고리즘과의 비교를 위해 0으로 설정하였다. 다음의 표3은 경로설정에 소요된 시간을 나타내고 있다.

그림 13은 경로 설정에 대해서 영역의 크기와 밀도값의 관계를 설명하기 위해 사용된 지도이다. 다양한 거리 값을 가지는 장애물 지역을 두어 임의접근순서확장법의 영역의 크기와 밀도값에 의해 경로가 변형되는

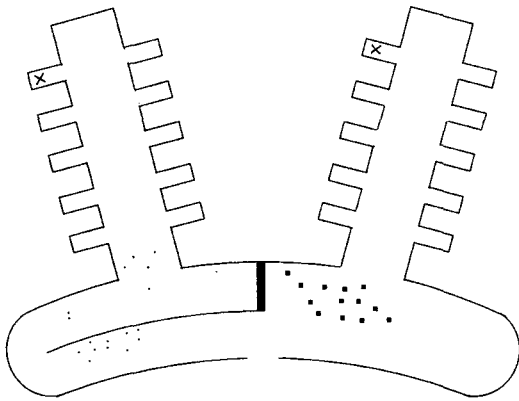


그림 13. 실제 환경의 지도
Fig. 13. The map in real world.

표 4. 실험 변수
Table 4. Variables of Experiments.

순서	영역 크기	탐색방향	밀도값	결과
1	2	default	0	elapsed time : 0.050 region num : 161228 route num : 735
2			8	elapsed time : 0.038 region num : 121642 route num : 749
3		optimal	0	elapsed time : 0.051 region num : 161228 route num : 735
4			8	elapsed time : 0.038 region num : 121642 route num : 749
5	8	default	0	elapsed time : 0.004 region num : 9475 route num : 192
6			8	elapsed time : 0.003 region num : 7560 route num : 226
7		optimal	0	elapsed time : 0.003 region num : 7485 route num : 192
8			8	elapsed time : 0.003 region num : 6637 route num : 6637

것을 보인다. 시작점과 도착점은 그림에서 표시된 (100,250)과 (700,250)이며 지도의 크기는 (1000 x 1000)이다. 표 4는 영역의 크기와 밀도 값 탐색 방향에 대해서 실험에 사용된 변수들을 정의하고 있으며, 벽면에 대해서 안정적인 경로를 설정할 수 있게 밀도값은 8로 고정하였다. 또한, 로봇의 크기는 8로 모두 같은 값을 가진다.

그림 13에서 보이고 있는 지도에서 아랫부분에 점으로 표시된 영역은 장애물 영역으로서 서로의 거리가 다른 점들로 이루어져 있다. 이를 통해서 본 논문에서 제시한 RAS의 밀도값을 조절하여 이동 할 수 있는 경로가 바뀌는 것을 볼 수 있으며, 이 값을 통해 이동 로봇

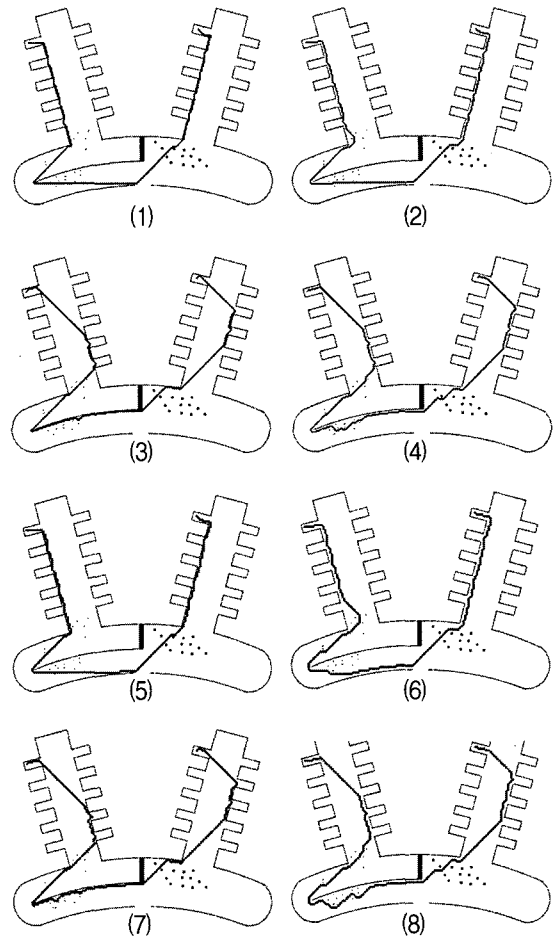


그림 14. 표4의 결과
Fig. 14. The Results of table 4.

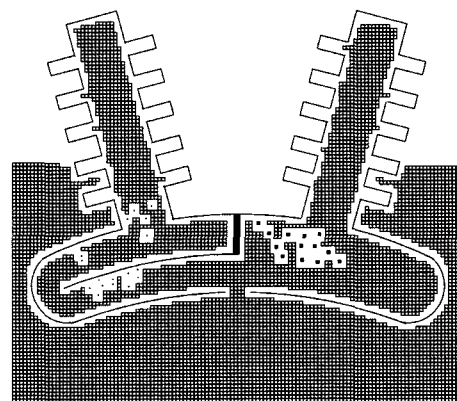


그림 15. 표4(8)의 RAS 결과
Fig. 15. RAS result when 8 is used for density.

의 크기와 벽면과 같이 장애물을 피한 최적 경로의 설정을 볼 수 있다.

그림 14의 영역크기가 2인 (1-4)경우에는 이 결과를

실제로 적용하기는 불가능하다. 그 이유는 로봇의 크기가 8이기 때문에 로봇은 벽과 부딪쳐 이동을 할 수 없다. 또한, 밀도값이 0인 경우에도 영역크기가 로봇의 크기보다 작은 경우에는 이동 로봇이 진행할 수 없다. 그 이유는 경로의 중심점을 따라서 이동하는 로봇은 영역의 크기가 작은 경우 벽면과 충돌하여 진행할 수 없기 때문이다. 위의 결과에 대해서 밀도 값이 8인 경우에 완성된 RAS지도는 그림 15에서 보이고 있다.

V. 결론 및 추후과제

본 논문에서 제안한 임의접근순서확장법(RAS)은 빠른 경로 설정 시간을 통해 다양한 환경에서 시작점으로부터 도착점까지 향하는 이동 로봇의 경로를 매우 적은 경로수와 이동거리로서 도달 할 수 있게 해준다. 다른 알고리즘과의 비교를 통해 그 성능을 입증하였으며, 다른 알고리즘도 역시 영역의 개수가 작은 경우에는 우수한 성능을 보여주었다. 그러나 영역의 개수가 증가할수록 다른 알고리즘들의 수행시간은 매우 빠르게 증가되었다. 따라서 넓은 영역에서는 시작점이나 도착점이 변경될 경우 경로를 설정하기 위해 매우 많은 시간이 필요하다. 이러한 사실은 이동하고 있는 로봇에게 있어서 매우 큰 단점으로 작용한다. 하나의 영역의 크기와 밀도값의 적절한 조화는 이동 로봇의 작업형태에 따라 조절할 수 있으며, 상황에 맞는 영역 크기와 밀도값의 선택이 중요하다. 영역 값을 매우 크게 하는 경우 넓은 지역에 대해서도 원활한 경로 설정이 가능하지만 영역 크기를 줄일 경우 방대한 양의 영역의 증가로 인해 경로 설정에 있어서 지연이 발생한다.

본 논문에서는 기본적인 방법만을 사용하였으나 이동하는 장애물이 존재하거나 영역에 대한 정보가 없는 동적환경, 입체적인 3차원에서도 확장 적용하는 연구 및 다양한 영역의 형태를 사용하는 방법에 관한 연구가 이루어져야 할 것이다.

참 고 문 헌

- [1] 이동하, 이영진, 김무진, "유전 알고리즘을 이용한 이동 로봇의 최적경로 계획", Proceedings of the 14th KACC, pp. 385-388, 1999. 10
- [2] 임경수, 국태용, "초음파 센서를 이용한 이동 로봇의 경로제어", Proceedings of the 14th KACC, pp. 358-361, 1999. 10

- [3] de Sousa, E. M. Hemeryl, "Adaptive control for mobile robot using wavelet networks", System, Man and Cybernetics IEEE transaction on, vol. 32, pp. 493-504, 2002. 8
- [4] E. Zalama, J. Gomez, J.R. Peran, " Adaptive behavior navigation of a mobile robot", System, Man and Cybernetics IEEE transaction on, vol. 32, pp. 160- 169, 2002. 1
- [5] Alex Yahja, Sanjiv Singh and Anthony Stentz, "Recent Results in Path Planning for Mobile Robots Operating in Vast Outdoor Environments", Proc. of Symposium on Image, Speech, Signal Processing and Robotics, 1998. 9
- [6] Jooyoung Hwang, Junsong Kim, Sangseok Lim, Kyuho Park, "A fast path planning by path graph optimization", IEEE Transactions on Systems, Man and Cybernetics, vol.33, pp. 121-129, 2003.1
- [7] Podsedkowski, L., "Path planner for nonholonomic mobile robot with fast replanning procedure", Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, pp. 3588-3593, 1998. 5
- [8] Anthony Stentz and Martial Hebert, "A Complete Navigation System for Goal Acquisition in Unknown Environment", Proceedings of the IRS 95', vol. 1, pp. 425-432, 1995.
- [9] 광재혁, 임준홍, "이동 시스템 제어를 위한 블루투스 네트워크", 제어자동화시스템학회 논문지, Vol. 10, no. 11, pp. 1052-1057, 2004.
- [10] <http://www.policyalmanac.org/games/aStarTutorial.htm>

저 자 소 개



곽 재 혁(정회원)
 1997년 한양대학교
 전자공학과 학사졸업
 2000년 한양대학교 대학원
 전자공학과 석사졸업
 2001년~현재 한양대학교 대학원
 박사과정 재학 중

<주관심분야 : 이동 로봇의 경로 제어 및 무선 네트워크>



임 준 흥(정회원)
 1979년 서울대학교
 전자공학과 학사
 1981년 한국과학기술원
 전기 및 전자공학 석사
 1986년 Univ. of Iowa
 전자공학 공학박사

1986년~1989년 한국항공대학 항공전자공학과
 조교수

1989년~1996년 한양대학교 전자공학과 부교수
 1996년~현재 한양대학교 전자컴퓨터공학부 교수
 <주관심분야 : 로보틱스, 시스템공학>