

논문 2006-43SC-6-1

동적 환경에서의 실시간 경로 설정 방법

(Real-time path replanning in dynamic environments)

곽 재 혁*, 임 준 홍**

(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 얻어지는 정보는 로봇이 진행하기 위한 경로에 대한 결정과 장애물에 대한 정보를 제공해준다. 대부분의 연구들은 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 결정하는 방법을 제안한다. 제안된 방법에서 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 recently. Informations from various sensors can be used to find obstacles and plan feasible path. In spite of many solutions of finding optimal path, each can be applied in only a constrained condition. This means that it is difficult to find universally good algorithm. An optimal path with a complicated computation generates a time delay which cannot avoid moving obstacles. In this paper, we propose an 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 and cumulative numbers. It has an advantage of fast planning time and completeness of path if one exists.. This means that new path selection may be possible within short time and that helps a robot to avoid obstacle in dynamic environments. Using the information of the start and destination position, the RAS can be performed for collision-free navigation by reforming feasible paths repeatedly in dynamic environments.

Keywords : 경로설정, 영역확장, 이동 로봇, 동적환경

I. 서 론

경로 설정에 관한 연구는 이동 로봇의 응용과 VLSI 디자인, 위성을 이용한 위치 탐색등 다양한 분야에서 핵심적인 문제로서 연구되고 있다. 경로 탐색의 과정은 공간의 효율적인 분할과 이 결과에서 최종 경로를 찾아내는 과정으로 구성된다. 공간의 분할을 위해서 skeleton, cell decomposition, potential field와 같은 방법이 많이 사용되고 있다^[1]. Skeleton을 사용한 접근방

식은 로봇의 움직임이 가능한 자유공간을 최소한의 개수로 표현된 일차원의 직선들로 연결된 네트워크를 구성한다. Voronoi diagram, subgoal network 등의 알고리즘에서 Skeleton의 방법을 사용하고 있다.

그러나 초기 상태의 그래프를 만들기 위해서는 실제 로봇의 센서가 매우 높은 정밀도를 가져야 하며 계산 과정에서도 많은 시간을 필요로 한다.^[2-6] Cell decomposition 은 자유공간을 작은 영역으로 분할한뒤 각 영역과 연결되어 있는 영역들과의 관계를 주어진 정보에 의해서 설정해 나간다. 따라서, 영역의 크기가 작을수록 공간에 대한 표현은 자세히 얻을 수 있지만 복잡한 관계에 의해 수행 시간이 증가되며 장애물과

* 학생회원, ** 정회원, 한양대학교 전자컴퓨터공학부
(School of Electrical Engineering and Computer
Science, Hanyang University)

접수일자: 2006년8월18일, 수정완료일 2006년10월30일

영역의 위치 관계에 의해서도 경로의 특성이 변한다. 이러한 cell decomposition은 2차원의 다각형으로 공간을 표현하는 quadtree에서도 사용된다.^[7] 높은 정밀도를 갖는 공간에서는 최단 거리를 구할 수 있는 가능성도 있지만, 같은 크기의 작은 영역들로 나누어진 공간에서는 최단거리를 보장하지는 못한다.

앞에서 설명한 경로 설정의 방법은 자유공간에 대한 정보, 이동 가능한 영역과 장애물에 대한 정보를 모두 사용하여 공간을 표현하는 과정에 대부분의 시간을 사용한다. 특정한 조건이 부여된 상황에서는 매우 좋은 결과를 얻을 수 있지만, 시작점과 도착점의 정보만을 사용하여 이동 중에 나타나는 장애물들에 대해서 새로운 경로를 생성해야 되는 동적환경이나 전체장애물의 지도를 모르고 국부적인 장애물(local obstacle)의 정보만을 아는 미지의 환경에서는 효과적이지 않다. 이러한 동적환경에서는 센서를 갖춘 이동 로봇이 실시간으로 얻게되는 주변 정보를 사용하여 경로의 재설정을 통해 작업을 수행한다. 또한, 이동하고 있는 장애물에 의해서도 이미 결정된 경로를 수정해야 할 필요가 생기며 로봇의 위치에 대한 오차를 통해서도 경로는 변경될 수 있다. 동적환경에서 이동 로봇의 경로가 갖는 특징은 최적화보다는 작업을 완전히 수행하는 것이 중요하다. 그래프이론을 기반으로 하는 A*,D*는 단순한 형태의 영역으로 분할된 공간에서 경로 설정 시간이 매우 빠르며, 최단 경로수를 가진다. 이러한 특성으로 인해 동적환경에서는 A*와 D* 알고리즘이 많이 사용되고 있다.^[8-10] 그러나, 목적지까지의 경로를 결정하는 방법이 경험적이기 때문에 경로 생성이 실패를 하는 경우도 발생한다.

본 논문에서는 동적환경이나 미지의 환경에서 적용할 수 있는 최적화된 경로 설정 알고리즘을 제안한다. 제안된 알고리즘에서는 시작점과 목적지까지의 정보만을 사용하여 이동중에 센서로부터 얻어지는 지형이나 이동 물체에 대한 정보를 사용하여 연속적인 경로 설정을 통해 이동 로봇이 장애물과의 충돌 회피가 가능한 경로를 따라 이동하게 한다. 또한 환경 변수의 설정을 통해 로봇의 크기에 대해서 영역의 크기와 밀도값을 조절하여 최적화된 경로가 생성되는 것을 보인다.

II. 임의순서확장법을 이용한 경로 설정

본 논문에서 제안하고 있는 알고리즘은 임의순서확장법(Random Access Sequence,RAS)이며^[11,12], 주변 영

역에 대한 탐색 방향에 제한을 갖지 않는 특징을 가지고 있다. 일반적으로 동적환경에 적용되고 있는 A*알고리즘은 Open List, Closed List로 표현되며 경험적(heuristic)인 방법을 사용하여 각각의 노드 N에 대해서 $G(N)$, $H(N)$ 의 두 가지 값을 이용하여 경로를 찾아낸다. 여기서 $G(N)$ 은 방향을 찾는데 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.

- 주변 영역에 대해서 F를 구한다.
- 리스트에 대한 이동을 한다.
- 가장 낮은 F값을 갖는 영역을 다음 단계의 시작점으로 한다.
- 도착점이 나올 때까지 a-c의 단계를 반복한다.
- closed list를 사용하여 경로를 생성한다.

이 과정에서 d의 과정을 나오기 위한 조건을 만족하지 못하면 경로 생성에 실패하게 된다. 이와 비교하여 RAS의 경로 설정 과정은

- 시작점으로부터 도착점까지 확장을 통해 순서를 부여한다.
- 누적값을 이용하여 경로를 생성한다.

이며, 만약 경로가 존재한다면 반드시 경로를 생성할 수 있다. 앞에서 비교한 순서에서 알 수 있듯이 RAS는 간단한 과정을 통해서 빠른 경로 설정 시간을 가진다.

1. 임의순서확장법의 개념

공간을 표현할 수 있는 형태는 매우 다양하게 존재한다. 점, 선, 그리고 다각형 형태가 그것이며, 그림 1에서는 정사각형 영역을 사용하여 장애물과 이동 가능한 영역을 표현한 것이다.

RAS는 그림1(a)와 같이 분포되어 있는 영역들과 이웃한 영역들이 가지고 있는 고유번호들의 관계를 통해서 경로를 찾을 수 있다. 그림1(a)에서 원으로 표현된 RAS의 계층적인 확장단계(step)는 그림1(b)와 같이 표

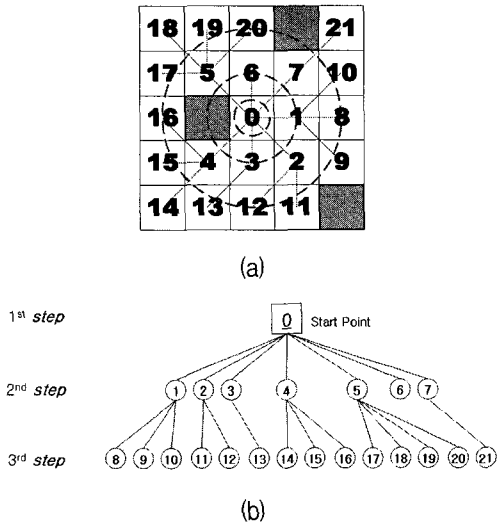


그림 1. RAS의 기본 개념도
Fig. 1. Basic scheme of RAS.

현된다. 전체 영역들은 하나의 집합 C 로 표현되며 시작점으로부터 도착점까지의 유한한 개수를 가진다. k_c 는 영역 c 에 부여되는 고유 번호이며 $c \in C$ 이다. N_{k_c} 와 P_{k_c} 는 각각 영역 c 가 갖는 인접한 영역들의 개수와 영역의 중심 위치이다. 분할된 영역의 형태에 의해 N_c 는 최대 8의 값을 갖게 된다. $Neigh(c)$ 는 주변에 있는 영역들이 가지는 고유 번호들의 집합이다. 순서 부여 과정을 위해 정의되는 $Start(c)$ 는 현재 단계에서 다음 단계를 시작하기 위해서 사용되는 시작점의 고유번호이며 $Start(c) = \min(Neigh(c))$ 이다. 정의에서 사용된 영역 c 는 고유번호 k_c 가 이전단계(step)가 시작되는 영역을 나타낸다.

2. 순서부여과정

순서부여 과정은 시작점의 고유번호를 '0'으로 해서 주변 영역에 대한 탐색 과정을 시작한다. 이때 장애물로 인식된 영역은 순서부여과정에서 제외시키고 그림 1에서 표현한 각단계별로 순서를 부여한다. 따라서, 시작점으로부터 외부로 확장해 나가는 형태로 과정이 진행되며, 2.1절에서 정의한 표현방법을 사용하여 순서부여 과정은

1st step

$$k_c = 0 (\star t \text{ point})$$

$$N_c = n_0$$

$$\neq igh(c) = \{1, \dots, n_0\} \text{ at origin}$$

2nd step

$$k_c = \star t(c) = 1$$

$$N_c = n_1$$

$$\neq igh(c) = \{n_0 + 1, \dots, n_0 + n_1\}$$

⋮ (1)

qth step (q > 1인 경우)

$$k_c = \star t(c)$$

$$N_{k_c} = m_{k_c}$$

$$\neq igh(c) = \left\{ \sum_{i=0}^{k_c-1} N_i + 1, \dots, \sum_{i=0}^{k_c} N_i \right\}$$

$$k_c = n_0$$

$$N_{k_c} = n_{n_0}$$

$$\neq igh(c) = \left\{ \sum_{i=0}^{k_c-1} N_i + 1, \dots, \sum_{i=0}^{k_c} N_i \right\}$$

⋮

$$k_c = \sum_{i=0}^{\star t(c)-1} N_i$$

$$N_{k_c} = m_{k_c}$$

$$\neq igh(c) = \left\{ \sum_{i=0}^{k_c-1} N_i + 1, \dots, \sum_{i=0}^{k_c} N_i \right\}$$

과 같이 표현할 수 있다. 그림 2를 사용하여 순서 부여 과정을 설명하면 시작점(S)를 '0'으로 설정하고 오른쪽을 시작으로 시계방향으로 '1'과 '2'의 순서를 부여한다. 그 다음 단계는 '1'로부터 주변영역에서 '3'을 찾는다. 이 과정을 통해 도착점(G)에 '11'을 찾고 과정을 완료한다.

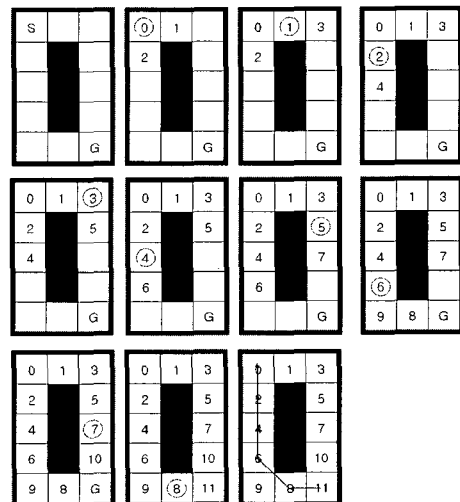


그림 2. 순서부여과정 예제
Fig. 2. Example of Numbering Process.

3. 경로 설정

도착점까지의 경로는 고유번호 k_c 와 인접한 영역의 개수인 N_{k_c} 만을 사용한다. 만약 $Neigh(c) =$

{1,2,3,4,5,6}과 같고 영역 c 의 고유번호가 $k_c = 0$ 과 같다면 $Neigh(c)$ 의 고유번호와 일치하는 영역들의 시작점은 원점인 '0'인 된다. 즉, 하나의 영역 c 에 대해서 $k_c = q$ 라면 $Neigh(c)$ 의 시작점은 q 가 된다. 따라서 식(1)을 사용하여 각 영역의 고유번호는 다음과 같은 표현된다.

$$\min(Neigh(c')) (at k_c = q) = \sum_{i=0}^{q-1} N_i + 1 \quad (2)$$

식(2)를 사용하여 영역들의 고유번호의 관계는 다음과 같이 나타낼 수 있다.

$$\min(Neigh(c')) < \sum_{i=0}^q N_i \leq \max(Neigh(c')) \quad (3)$$

at $k_c = q$

식(3)의 관계를 이용하여 경로를 설정하는 과정을 표 1에서 보이고 있다.

표 1.에서 알 수 있듯이 경로의 탐색은 도착점으로부터 시작점을 찾아가는 역방향이다. 생성된 경로 R_n 은 고유번호들이 나열된 형태이며 $k_c = R_n$ 를 만족하는 영역 c 의 중심점 P_{k_c} 를 사용하여 경로를 생성한다.

표 1. 경로 설정 알고리즘
Table 1. Algorithm for path planning.

```

1: while ( $R_{n+1} \neq \text{origin}$ )
2:   if ( $R_n \leq \sum_{i=0}^K N_i$ )
3:      $R_{n+1} = K$ ,
4:      $n = n + 1$ 
5:   end
6: end
    
```

4. 환경 설정 변수

임의순서확장법에서 경로 설정을 결정하는 가장 중요한 변수는 로봇의 크기(S_0), 영역의 크기(S_c)와 밀도값(d_c)이다. 임의순서 확장법에 의해서 얻어진 경로는 최단 경로수를 갖는 하나의 경로만 구해진다. 이 경우 임의순서확장법에 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(4)와 같이 정의되고 각 영역의 주변에 연결된 장애물을 제외한 이동 가능한 영역의 개수이다. 인접한 영역과 다음 단계들의 영역에 대해서 그 값을 포함한다.

$$d_c = \begin{cases} N_{k_c} & \text{if } step = 0 \\ \sum_{i=0}^{N_k} N_{k_c'} & \text{for } k_c' \in Neigh(c) \quad \text{else} \end{cases} \quad (4)$$

그림 4는 밀도값의 적용 여부에 따른 이동 로봇의 경로의 차이점에 대해서 나타내고 있다. 맨하탄방법과 같이 정사각형 형태로 장애물과 이동가능한 영역을 표현한 공간에서 시작점과 도착점의 경로는 밀도값을 적용하지 않은 경우 점선으로 표현된다. 이와 같은 경우에 이동 로봇의 크기가 영역의 크기보다 크기 때문에 그림에서처럼 충돌을 하거나 장애물들의 사이를 통과해 갈 수 없다. 밀도값은 실제 환경에서의 적용을 효율적으로 하기 위한 변수이며 그림 4의 장애물 주변의 어두운 부분은 밀도값 8을 만족하지 못한다. 따라서 밀도값을 적용하여 생성된 경로는 실선으로 표현되며, 도착점까지의 작업을 수행 할 수 있다.

장애물에 대해서 독립적인 경로생성도 영역의 크기와 밀도값을 조절함으로써 가능하다. 그림 5와 같이 영역의 크기가 이동 로봇의 크기보다 큰 경우 (a)와 같이 이상적으로 장애물이 놓이는 경우보다는 장애물로 인식

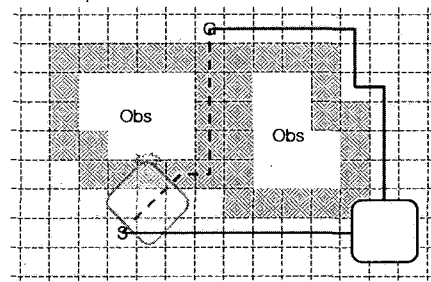


그림 4. 밀도값을 이용한 경로 수정
Fig. 4. Path planning using density.

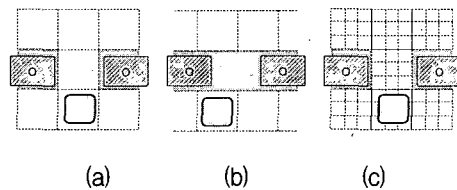


그림 5 장애물에 대한 영역의 위치
Fig. 5. region position for obstacles.

되는 (b)의 경우가 많이 발생한다. 이와 같은 경우 전체 경로의 길이를 불필요하게 증가시키는 문제를 발생시키기 때문에, (c)와 같이 영역의 크기를 줄이고 밀도 값을 사용하여 최단 경로를 찾을 수 있는 가능성을 높여 줄 수 있다.

3. 동적 환경에서의 경로 설정

임의순서확장법은 주변 정보가 없는 동적환경에서도 적용될 수 있다. 그 이유는 경로 설정에 소요되는 시간이 짧기 때문에 이동 로봇의 이동 시간에 영향을 주지 않고, 시작점으로부터 도착점까지의 경로가 존재한다면 반드시 하나의 경로만이 생성되기 때문이다. 시작점으로부터 이동로봇은 주변에 있는 벽이나 장애물들을 인식하고 경로의 지속적인 재설정을 통해 도착점을 찾아낸다. 전체 시스템은 그림 6과 같이 구성되며 센서로부터 얻게되는 거리 정보 S 는 로봇 제어와 임의순서확장법을 수행하는 프로세서로 동시에 보내지게 된다. 이동로봇이 진행하게될 경로는 RAS로부터 구해진 경로 (R_0, R_1, \dots, R_n) 를 따라서 이동하게 되며, 로봇 제어부로 보내진 거리 정보는 로봇이 다음 경로까지 가는 도

중에 나타난 장애물을 피하기 위한 정보로 사용된다. 구해진 경로 R_0 는 이동 로봇의 현재 위치와 동일하다.

그림 7은 동적 환경에서의 장애물 회피 및 경로 설정의 과정을 순차적으로 나타낸 것이다. 먼저 시작점에서 주변 장애물에 대한 정보를 얻고 도착점까지의 경로를 RAS를 통해서 구한다. 구해진 경로 R_n 중에서 시작점 R_0 에서 다음 위치인 R_1 을 향해 이동한다. 이동중에는 지속적인 장애물 정보에 대한 획득을 통해 움직이는 장애물에 대한 회피를 가능하게 한다. R_1 에 도착하면 목

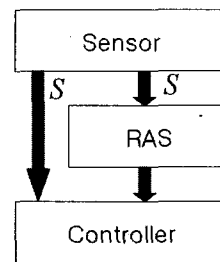


그림 6. 동적 환경에서의 이동 로봇 제어 시스템
Fig. 6. A control system for mobile robot in the Dynamic environment.

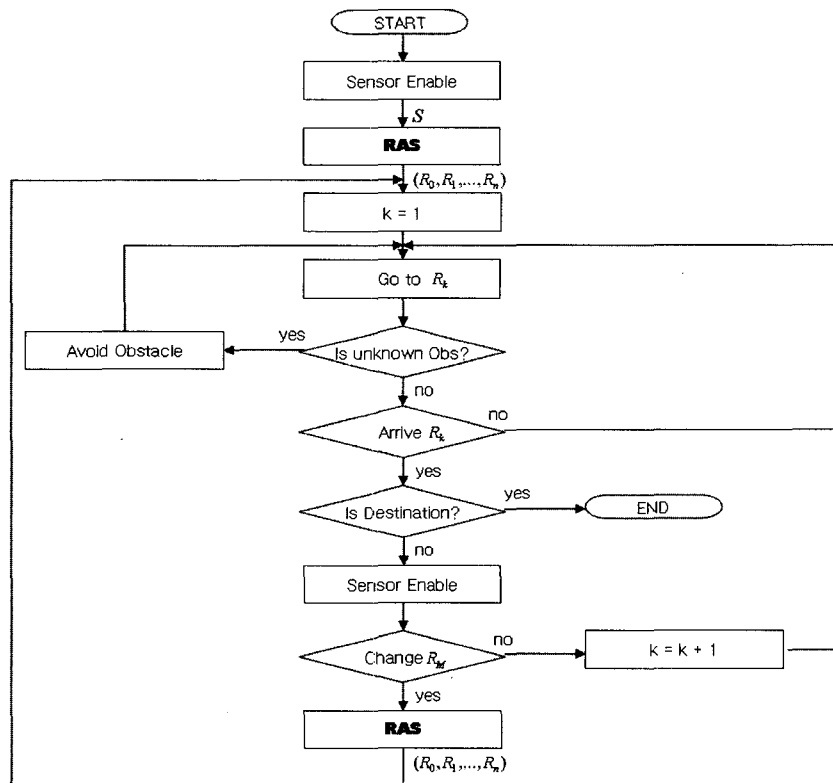


그림 7. 동적 환경에서의 경로 설정 및 장애물 회피 과정
Fig. 7. process of path planning and avoid obstacles in dynamic environment.

적지의 여부를 판단하고 목적지가 아니면 주변영역에 대한 장애물 정보를 갱신한다. 새로운 장애물이 존재한다면, RAS를 통해 새로운 경로를 생성하고 현재의 위치인 R_1 이 새로운 경로의 R_0 가 된다. 이와 같은 과정을 반복하면서 동적환경에서 목적지까지의 경로에 대한 이동 로봇의 주행을 한다.

IV. 실험 및 결과

동적환경에서 RAS의 성능평가를 위해서 실제 이동 로봇을 사용하여 경로주행을 하였으며, 모의실험을 통해서 복잡한 환경에서의 경로 설정 시간과 동적환경에 대한 이동 거리를 측정하였다. 실제환경에서의 실험은 상단에 위치한 CCD카메라를 사용하여 로봇의 위치와 장애물을 인식하고 RAS를 통해 생성된 경로를 통해서 주변 환경에 대한 정보가 있는 경우와 동적환경에 대해서 실험하였다. 두가지의 실험에서 사용된 센서는 모두 가상으로 구현되었으며, 레이저 센서와 같이 전방의 영역에 대해서 1도씩의 간격으로 최대 100픽셀의 거리에 대해서 장애물을 인식하도록 설정하였다. 카메라를 통해 인식된 지도는 (640×480)픽셀의 해상도를 가지며 실제 크기는 2.4m × 1.6m이다.

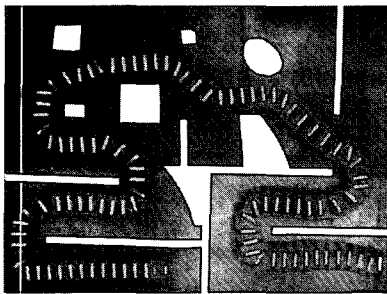


그림 8. 경로 설정 후의 이동 로봇의 궤적
Fig. 8. A trajectory of mobile robot after complete path planning.

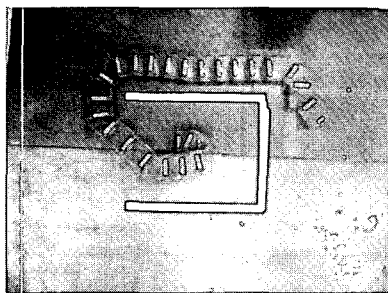


그림 9. 동적 환경에서의 경로설정('C'자 형태)
Fig. 9. A path planning in unknown environment(shape of 'C').

실제 환경에서의 실험은 인식된 로봇의 크기가 32이며, 이동로봇의 크기를 고려하여 영역의 크기는 22로 설정하고 벽면과의 충돌을 피하기 위해 밀도값은 8로 하였다. 그림 8은 장애물에 대한 정보를 사용하여 시작점으로부터 도착점까지의 경로 생성 후 이동한 로봇의 궤적을 나타내고 있다.

그림 9와 10은 시작점과 도착점의 정보만을 사용하여 가상의 센서를 사용하여 경로를 찾아가고 있는 모습을 나타내고 있다.

그림 9의 경우 장애물 안에서 도착점을 향해 놓여진 로봇은 하나의 영역 크기만큼 이동 후 장애물을 인식하고 다시 돌아서 나오는 과정을 보여주고 있다. 그림 10에서도 마찬가지로 도착점인 아래쪽을 향해 놓여진 이

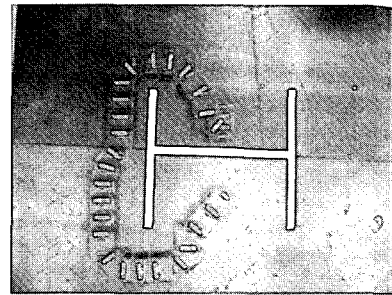


그림 10. 동적 환경에서의 경로설정('H'자 형태)
Fig. 10. A path planning in unknown environment(shape of 'H').

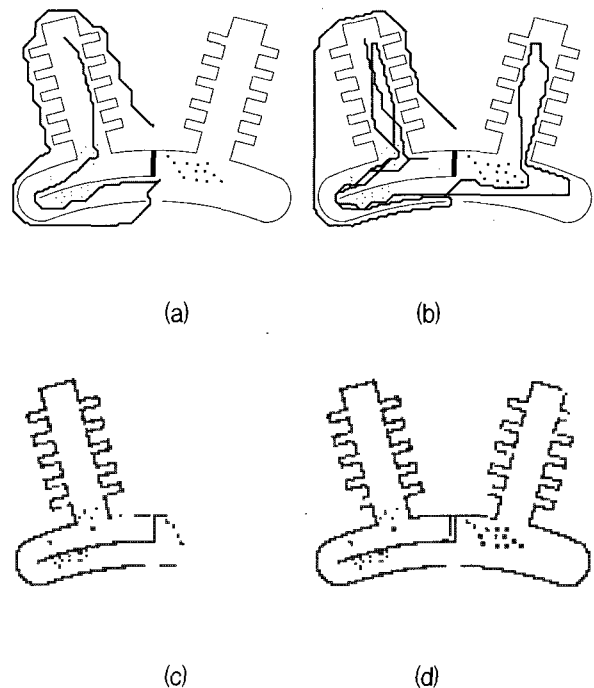


그림 11. 동적환경에서의 모의실험 결과
Fig. 11. Simulation results in the Dynamic Environment.

표 2. 모의실험 변수 및 이동 거리

Table 2. A variables of simulation and moving length.

	그림 11(a)	그림 11(b)
지도 크기	1000×1000	1000×1000
시작점	500,500	200,200
도착점	200,200	500,500
영역크기	8	8
밀도값	8	8
이동거리	3208.8	7270.0

동로봇은 장애물인식전까지 전진 후 다시 돌아오는 행동을 한다. 좌측면에서의 이동 경로가 직선을 나타내지 못하는 이유는 그림 5(b)와 같은 이유로서 실제 장애물의 각도가 기울어져 있기 때문에 좌측면의 아래쪽은 장애물로 인식되어 영역의 크기만큼 이동된 경로가 생성된 결과이다.

그림 11의 (a)와 (b)는 모의실험에서 장애물에 대한 정보가 없는 상태에서의 결과를 나타내고 있으며 (c)와 (d)는 이동 중에 인식한 장애물들을 나타낸 것이다. 표 2에서는 RAS의 환경 변수들과 이동 경로를 보여준다.

그림 11의 결과에서 보듯이 시작점이 다른 두 가지의 경우에 이동거리가 차이가 나는 것을 볼 수 있다. 그림 11(b)의 경우 시작과 동시에 도착점을 향하는 방향으로 이동을 시작하기 때문에 시작점의 윗부분에 있는 장애물을 인식하지 못하였다. 따라서, 로봇은 내부의 모든 영역을 중복하여 검색을 하기 때문에 이동 거리가 매우 크다.

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. 단순한 계산 과정은 영역의 크기나 장애물의 복잡도와 상관없이 목적지까지의 경로를 생성할 수 있다. 이러한 특성은 주변 환경에 대한 정보가 없는 동적환경에서도 적용가능하다는 것을 보여주고 있으며, 환경 설정을 위한 영역의 크기나 밀도값의 적용을 통해서 실제 환경에서도 좋은 결과를 나타내고 있다. 일반적으로 생성된 경로와는 다르게 RAS는 실제 환경에서 이동 로봇이 충돌을 회피하고 진행할 수 있는 경로를 생성할 수 있게 해준다. 또한 국소해(local minima)와 같은 문제도 발생하지 않는다.

본 논문에서는 가상으로 구현된 센서를 사용하여 절대적인 정확도로 장애물을 인식하였다. 그러나 실제의 센서는 외부 요인이나 성능에 의해 부정확성을 가지게 된다. 이러한 문제점을 해결하기 위한 알고리즘의 적용과 함께 광범위한 영역에서의 이동 로봇의 주행에 대한 연구가 이루어져야 할 것이다.

참고 문헌

- [1] Y. K. Hwang and N. Ahuja, "Gross Motion planning - A survey", ACM Comput. Surv., vol. 24, pp.219-291, 1992.
- [2] 이동하, 이영진, 김무진, "유전 알고리즘을 이용한 이동 로봇의 최적경로 계획", Proceedings of the 14th KACC, pp. 385-388, 1999. 10
- [3] 임경수, 국태용, "초음파 센서를 이용한 이동 로봇의 경로제어", Proceedings of the 14th KACC, pp. 358-361, 1999. 10
- [4] Sven Koenig, "Fast Replanning for Navigation in Unknown Terrain", IEEE TRANSACTIONS ON ROBOTICS, vol. 21, no. 3, pp 354-363, 2005.
- [5] 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
- [6] 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
- [7] 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
- [8] 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
- [9] 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
- [10] 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.
- [11] 광재혁, 임준홍, "이동 시스템 제어를 위한 블루투스 네트워크", 제어자동화시스템학회 논문지, Vol. 10, no. 11, pp. 1052-1057, 2004.

[12] 곽재혁, 임준홍, “분포 밀도를 이용한 이동 로봇의
 최단 경로 설정”, 대한전자공학회지, vol. 43, no. 3,
 pp.31-40, May 2006.

저 자 소 개



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

<주관심분야 : 이동 로봇의 경로 제어 및 무선 네트워킹>



임 준 홍(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>