

A Sweeping Algorithm for an Autonomous Mobile Robot under the Unknown Environment

朴 柱 勇* 李 起 東**
(Ju-Yong Park · Ki-Dong Lee)

Abstract - There has been an ever increasing interest in mobile robot for home services. However, issues currently being investigated for path planning of the mobile robot is concentrated to solving the problem of finding the optimal path from the initial location to the final location under the given performance index. In this study, we newly present a sweeping algorithm for autonomous mobile robot to cover the whole closed area under the unknown environment. And we verify the validity of the formalized algorithm by computer simulation with the changing environment conditions. In addition to this, we analyse the effect of real system implementation of the proposed algorithm to a experimental miniature mobile robot(Khepera).

Key Words : mobile robot, path planning, sweeping algorithm, computer simulation

1. 서 론

오늘날 로봇은 산업계의 여러 분야에서 많이 이용되고 있다. 그러나 향후에는 산업계뿐만 아니라 군사용 및 의료용과 가정용 등과 같이 다양하게 로봇이 이용될 것이다. 그 중에서도 특히 서비스용의 로봇에 대한 수요가 폭발적으로 증가하리라고 예상된다. 이러한 로봇을 동작시키기 위하여 필수적인 여러 가지 알고리즘 중에 상당히 중요한 부분을 차지하는 것이 바로 로봇의 경로계획이다. 이러한 경로계획은 현재 주변의 환경에 대하여 충돌회피를 하면서 최적의 경로를 찾는 문제로 귀착되고 있다. 물론 주변환경에 대한 정보를 미리 가지고 경로계획을 하는 경우와 미지의 환경에 대하여 경로계획을 하는 두 가지로 나눌 수 있으며, 장애물의 종류에 따라 시변(time-varying)장애물과 시불변(time-invariant)장애물의 두 가지 경우에 대하여 각각 적용되는 알고리즘들이 연구되고 있다. 일반 산업용 로봇의 경우에는 움직이고 있는 로봇의 몸체 자체가 시변 장애물로 인식되기 때문에 현실적으로 고기능의 로봇(5축 이상의 다축 로봇)에 대한 충돌회피 알고리즘은 실 시스템에 적용하는 것이 상당히 힘들다.

일반적인 매니플레이터의 경로계획 방법은 Lozano-Perez가 주장한 형상공간법(configuration-space method 또는 C-space method)과 Khatib가 주장한 인위적인 위치장 기법(artificial potential-field method)등으로 대표될 수 있으나 이들의 실제 구현은 상당히 힘든 반면 이동로봇의 경우는 보다 용이하여 실용적 구현이 가능하다[1-3]. 그러나 고정형 로봇의 경우와 마찬가지로 현재까지의 주된 연구 관점은 환경에 대한 정보를 미리 가지고 초기 위치에서 목표로 설정된 최종 위치까지 충돌을 회피하면서 주어진 성능평가 지표에 따른 최적의 경로를 찾아내는 문제에 모아지고 있다[4-8]. 이와 같은 조건하에서의 이동로

봇 경로계획 방법들은 Chung 등이 분류해 놓은 자료를 참조하면 다음과 같다[9].

(1) 형상공간법(configuration-space method)

장애물을 운반체의 모양에 따라 확장하고 운반체를 하나의 점으로 가정하여 문제를 단순화시킨 후 경로를 구하는 방법으로 운반체가 장애물에 점유되지 않은 자유공간에 있도록 하는 방법이다[1]. 그러나 운반체가 다각형 형태이며 주행 중에 회전 운동을 병행하는 경우 장애물과 충돌 할 수 있으며 실시간으로 적용하기 어렵다[10].

(2)거리함수법(distance function method)

운반체와 장애물과의 충돌회피를 거리함수로 표시한 후 이를 상태변수와 입력의 제한 조건을 갖는 최적제어 문제로 해결하는 방법이다[11]. 그러나 거리함수를 구하기 위하여 장애물의 윤곽과 정점의 위치를 정확히 알고 있어야 한다.

(3) 격자탐색법(grid search method)

운반체의 작업공간을 이차원으로 가정하고 격자형 환경모델(grid-type world model)을 이용하여 충돌회피 경로를 구하는 방법으로 장애물에 점유되지 않은 격자를 연결하여 경로를 구하는 방법이다[12]. 그러나 격자 탐색 방법은 두 지점간의 경로만을 구할 수 있으며 다른 지점간의 경로를 구하기 위하여 환경모델을 다시 설계해야 한다.

(4) 네트워크법(network method)

운반체의 작업공간을 노드와 아크로 구성된 네트워크로 표현 한 후 아크들의 연결에 의하여 경로를 구하는 방법이다. 경로계획은 사용하는 방법에 따라 두 노드간의 경로, 한 노드와 다른 다수 노드와의 경로 또는 모든 노드간의 경로가 구해진다.[9]

위의 연구 결과에서 보듯이 로봇이 여러 가지 형태의 작업 공간에서 자율적으로 작업을 하기 위해서는 필수적으로 경로계획이 필요하다. 따라서 로봇의 경로계획에 대한 연구는 지금까지 상당히 많은 연구자들에 의하여 많은 연구 결과가 있으며 현재도 진행 중이다. 그러나 고정 로봇의 경우뿐 아니라 보다 구현이 용이한 이동 로봇의 경로계획에 대한 대부분의 연구는

* 準 會 員 : 嶺南大 工大 컴퓨터工學科 碩士
 ** 定 會 員 : 嶺南大 工大 컴퓨터工學科 助教授 · 工博
 接受日字 : 1998年 4月 11日
 最終完了 : 1998年 11月 18日

초기위치에서 목표위치까지의 최적경로를 구하는 문제에 관한 것이며 그 중에서도 계획 이전에 미리 환경에 대한 정보를 인지하고 있는 경우에 대한 연구이다.

본 연구에서 새로이 제안하고자 하는 자율주행 이동로봇의 경로계획에 대한 연구는 주변 환경에 대한 정보를 미리 인지하고 있지 않은 상황에서 센서 정보를 이용하여 주변 환경 정보를 받아들이면서 닫혀진(closed) 이차원 전체 평면을 빠짐없이 청소(sweep)하는 알고리즘을 제시하고자 한다. 이와 같은 분야에서의 연구는 거의 전무한 상태이며, 본 연구에서 최초로 정형화된 알고리즘을 제안하고자 한다. 제시된 알고리즘에 대하여 컴퓨터 시뮬레이션을 수행하여 그 타당성을 입증하며, 이러한 분석 결과를 이용하여 실제로 실험실용 이동로봇(Khepera)에 본 알고리즘을 이식하여 그 실용성을 테스트한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 지금까지 개발된 몇 가지의 청소용 로봇시스템에 대하여 알아본 후, 새로 제안된 알고리즘을 3장에서 기술한다. 4장에서는 제안된 알고리즘에 대한 소프트웨어적인 시뮬레이션뿐만 아니라 상용의 실험용 로봇인 Khepera를 이용하여 하드웨어적인 시뮬레이션을 실시하여 그 유용성을 검증한다. 마지막으로 본 연구의 결론 및 문제점, 그리고 향후 연구과제에 대하여 5장에서 기술한다.

2. 청소용 자율주행 로봇

지금까지 몇몇 청소 로봇을 위한 주행 알고리즘이 개발되어 왔다[13-16]. 이러한 알고리즘은 주어지는 작업영역 지도를 보고 로봇이 청소 작업을 하기 위해서 이동해야할 경로를 먼저 계획한 다음 이 경로에 따라 로봇이 움직이면서 청소 작업을 수행하였다. 그래서 불 특정한 미지의 환경에서 작업을 수행할 경우에는 유연성 있는 실행이 어렵다.

Maryland 대학에서 개발된 시스템은 간단한 구조의 본체에 각각 8개의 접촉센서와 적외선센서를 장착하고, 입력 환경을 여러 개의 차선(lane)으로 분류하여 로봇을 구동시킨다. Microcomputing사에서 개발한 청소 로봇은 기본 본체 위에 부가적인 팔을 부착하여 사용한다[13]. 운행 알고리즘은 우선 외곽벽을 따라 이동하면서 작업영역 범위를 확인하여 윤곽 지도를 생성한다. 그리고 로봇의 현재 위치에서 동서남북 4방향을 확인하여 그 중 가장 청소가 이루어지지 않은 방향으로 이동하면서 외곽벽을 만날 때까지 이동하며 청소 작업을 수행한다. 벽을 만나게 되면 다시 4방향을 확인하여 앞의 과정을 반복하며 청소 작업을 수행한다. 그림 1은 장애물이 존재하지 않는 작업영역에 이러한 Microcomputing사의 청소 알고리즘을 적용한 예이다. 그림 1을 살펴보면 중복청소가 이루어지는 영역이 많고 완벽한 청소가 이루어지지 못함을 알 수 있다. 이 알고리즘은 작업영역 내에 장애물이 없는 경우 주어진 작업영역의 약 90%를 청소한다.

동일본철도 회사에서는 닦는 형태(scrubber)와 쓰는 형태(sweeper)의 두 가지 청소 로봇을 개발하였다[16]. 쓰는 형태의 로봇은 자화 테이프 위로 진행되는 방법 또는 교시-반복 방법으로 주행할 수 있으며, 닦는 형태의 로봇은 미리 정해진 영역을 청소하는 모드와 build-in map 모드의 두 가지 모드를 제공한다.

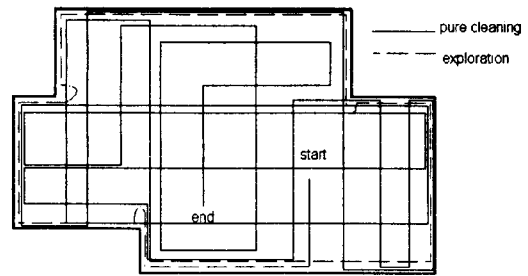


그림 1 Microcomputing사의 청소로봇 주행경로
Fig. 1 Cleaning path of the Microcomputing robot

3. 새로운 청소 알고리즘

2장에서 기술된 바와 같이 청소 알고리즘에 대한 연구는 매우 빈약하다. 또한, 기존의 알고리즘으로는 미지의 환경 하에서 효율적으로 전체 영역을 청소하기 힘들다. 따라서 본 연구에서는 불 특정한 미지의 환경에 장애물이 있는 작업영역에서도 완벽한 청소 작업을 수행할 수 있도록 하는 알고리즘을 제안한다. 그리고 작업을 수행하면서 장애물의 위치나 청소 작업이 이루어진 위치 정보 지도를 생성하기 때문에 중복된 청소 작업을 피할 수 있다. 제안된 청소 알고리즘은 평면 청소 알고리즘(plane sweeping algorithm)과 두 위치 점간의 이동 알고리즘(point-to-point transfer algorithm)과 마무리 작업 알고리즘(corner-work algorithm)으로 나눌 수 있다. 즉, 청소 작업을 시작하면 평면 청소 알고리즘과 두 위치 점간의 이동 알고리즘을 이용하여 평면 청소 작업을 수행한 후 장애물 주위나 외곽벽 주위에 청소가 이루어지지 않은 부분을 청소하기 위해서 마무리 작업 알고리즘을 적용하면 청소 작업이 끝나게 된다. 이러한 청소 알고리즘은 그림 2와 같은 순서도로 나타낼 수 있다.

각 알고리즘의 세부적인 내용은 다음과 같다.

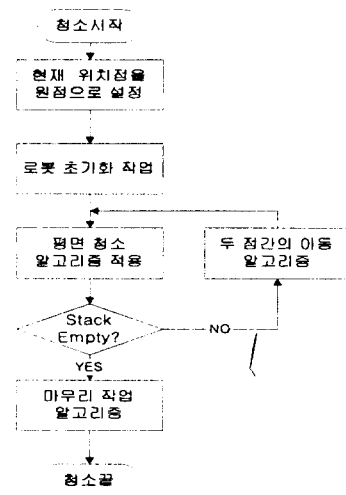


그림 2. 청소 알고리즘 순서도
Fig. 2 Flowchart of the sweeping algorithm

3.1 평면 청소 알고리즘(Plane Sweeping Algorithm: PSA)

PSA는 청소가 이루어지지 않고 지나가는 영역 없이 청소 로봇이 이동할 수 있도록 미지의 작업영역에서 로봇이 이동할 경로를 자동으로 찾아 준다. PSA의 기본적인 개념은 청소 작업을 수행하는 동안 청소가 이루어지지 않은 채 지나갈 가능성이 있는 장소의 위치 값을 스택에 저장해 둔 후, 추후 스택에 들어 있는 위치 값을 청소 작업을 수행하면서 생성한 청소 지도에서 확인하여 그때까지 청소가 이루어지지 않은 경우에는 그 위치로 이동하여 평면 청소 알고리즘을 계속 수행하도록 한다. 여기서 청소 지도는 청소 작업을 수행하면서 장애물의 위치 값이나 청소가 이루어진 곳의 위치를 Bitmap형태로 저장해 두어 이 값을 이용하여 스택에서 가져온 위치가 청소가 이루어졌는지 아닌지를 쉽게 알 수 있다. 세부적인 PSA처리 방법은 다음과 같다.

- 단계1) 로봇이 장애물이나 벽을 만나기 전까지 직진하며 청소 작업을 수행한다.
- 단계2) 이동하는 선분 양옆으로 가상의 선분을 생성한다.
- 단계3) 이 가상의 선분에 청소가 이루어진 곳이 있는지 없는지를 확인한다. 만약 두 선분 중 청소가 이루어지지 않은 곳이 있다면 그 선분의 시작점을 스택에 저장해 둔다.

단계4) 더 이상 청소할 곳이 없는 경우 단계1)로 이동한다.

그림 3을 보면 평면 청소 알고리즘의 이해를 도울 수 있을 것이다. 사용자는 우선 청소 로봇을 임의의 위치에 두고 작업을 시작한다. 그러면 청소 로봇은 작업 환경에 대한 아무런 정보 없이 전진하면서 청소 작업을 시작한다. 이러한 청소 작업 중 장애물이나 벽을 만나게 되면, 청소할 폭 만큼 오른쪽 또는 왼쪽으로 번갈아 가며 이동하고 로봇 이동 방향을 반대로 하여 청소 작업을 계속 수행한다. 청소 작업을 수행할 수 있는 공간이 더이상 없을 때까지 이 과정을 반복하여 수행한다. 이러한 처리가 이루어지면, 그림 4와 같은 결과를 얻을 수 있다. 이 과정에서 그림 5 a)와 같이 청소가 이루어지지 않은 양쪽 선분의 시작점들(A, B)이 스택에 저장된다. 그림 5 a)와 같이 첫 번째 작업이 끝나면 스택으로부터 위치값 B를 얻어내어 이 점에서 청소 작업을 재 수행한다. 이 과정의 결과는 그림 5 b)와 같이 되고, 스택에서 위치값 A를 가져와 다시 청소 작업을 재 수행하면 그림 5 c)와 같은 결과를 얻게 된다. 이 경우는 다시 위치값 C가 스택에 저장된다. 위치값 C를 스택에서 가져와 청소 작업을 수행하면, 그림 5 d)와 같이 청소 작업이 이루어진다. 이 작업이 끝난 후 스택을 확인하면, 더 이상 스택에 저장된 위치 값이 없다. 그러면 평면 청소 작업을 끝낸다. 이와 같이 평면 청소 알고리즘의 모든 처리 과정이 끝나면, 그림 5 d)와 같이 모든 영역에서 청소가 이루어진다.

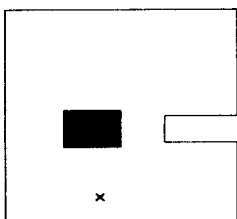


그림 3 간단한 작업 환경
Fig. 3 Simple environment

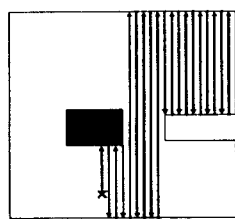


그림 4 일단계 청소 작업
Fig. 4 First step of sweeping

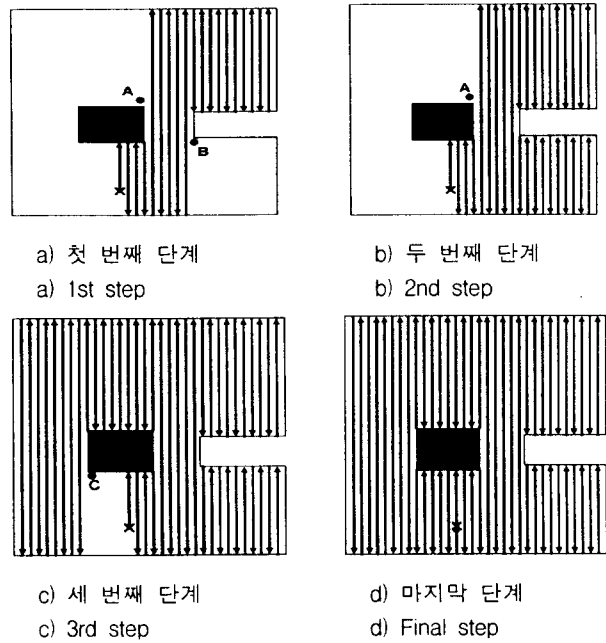


그림 5 PSA를 이용한 청소작업 과정

Fig. 5 The procedure of the PSA

본 연구에서는 평면 청소 작업 수행 중 청소가 이루어지지 않고 지나친 영역의 시작점을 저장하기 위해서 큐(queue)를 사용하지 않고 스택(stack)을 사용하였다. 그 이유는 스택의 후입력 선출력(LIFO:Last-In First-Out) 특성 때문이다. 큐의 경우에는 선입력 선출력(FIFO:First-In First-Out)의 특성을 가지기 때문에 다음 청소 작업 시작을 위해 스택에서 가져온 시작점이 큐에서 가져온 시작점보다 이전 청소 작업이 끝난 위치에서 가까운 위치 값을 얻을 가능성이 높다. 이렇게 하면 다음 청소를 수행할 위치를 현재 위치에서 가까운 것을 선택함으로써 청소 로봇이 이동하는 전체 이동 거리를 줄일 수 있기 때문에, 큐보다는 스택을 사용하여 위치값을 관리한다

3.2 두 점간의 이동 알고리즘

앞의 절과 같이 로봇은 장애물을 피하면서 현재 위치 점에서 다음 청소 작업이 이루어질 곳으로 이동할 수 있어야 한다. 그래서 청소 알고리즘을 수행하기 위해서는 두 점간의 이동 알고리즘이 필요하다. 본 논문에서는 이러한 알고리즘을 새롭게 개발하지 않고 이전에 개발되어 있는 두 점간의 이동 알고리즘을 사용한다. 이동 로봇을 위한 두 점간의 이동 알고리즘은 많이 연구되었으나, 이동지도를 이용한 방법(roadmap approach)과 셀 분할법(cell decomposition)과 자기장 이용한 방법(potential field)등이 많이 사용된다[17].

이동지도를 이용한 방법은 로봇의 이동 가능한 영역을 일차원 곡선으로 연결하여 이동 경로 지도를 생성하는 방법이다. 이 방법에서 중요한 것은 이동 경로 지도를 생성하는 것이다. 이 이동 경로 지도를 생성하는 방법에는 투시도 그래프(visibility graphs)를 이용한 방법, Voronoi 도표법(voronoi

diagram)[18], 자유 경로망 방법(free way nets)과 실루엣 방법(silhouette)이 있다. 셀 분할법은 로봇의 이동 가능한 작업영역을 셀이라는 영역으로 나누어 두 셀 간의 이동 경로를 두 셀 사이에 존재하는 셀들의 연결로 나타내는 방법이다. 자기장을 이용한 방법이란 로봇을 하나의 자기력을 가진 물체로 보고 장애물과는 척력이 작용하고 목표점과는 인력이 작용한다고 가정하여 로봇과 작업 환경과의 힘의 관계에 따라 로봇이 이동 경로를 선택하는 방법이다. 이러한 여러 가지 경로 결정 방법 중에서 본 논문에는 셀 분할법 중 근사적 셀 분할법(approximate cell decomposition)을 사용한다[19]. 이 방법은 작업영역을 사용자가 미리 정해 놓은 모양의 셀들로 나뉘 이것들을 연결하여 경로를 찾는 방법이다. 쿼드트리(quadtrees)를 이용하여 셀 분할시에 소요되는 계산량을 줄일 수 있고, 무한한 시간이 주어졌을 때 반드시 두 점간의 이동 경로를 찾을 수 있다는 특징을 가진다. 이러한 경로 보장성 즉, 반드시 두 점간의 이동 경로를 찾을 수 있다는 성질은 청소 알고리즘에서 반드시 필요한 조건이다. 이러한 계산량이 적고, 경로를 보장한다는 특성 때문에 본 연구에서는 근사적 셀 분할법을 두 위치 점간의 이동 알고리즘으로 선택한다.

3.3 마무리 작업 알고리즘(Corner Work Algorithm: CWA)

PSA와 두 점간의 이동 알고리즘을 이용하여 청소 작업을 수행한 후에도 벽이나 장애물 주위에 다소 청소가 이루어지지 못한 곳이 남아 있다. 마무리 작업 알고리즘은 이러한 벽이나 장애물 주위의 청소가 이루어지지 못한 영역을 청소하도록 도와주는 알고리즘이다. CWA는 두 가지 지도를 이용하여 처리가 이루어진다. 하나는 PSA에서 생성한 청소 지도(sweep map)이고 다른 하나는 청소 지도를 통해 찾아 낼 수 있는 장애물 위치만 나타내는 장애물 지도(obstacle map)이다. 이 장애물 지도를 사용함으로써 CWA는 쉽게 장애물에 대한 정보를 얻을 수 있다. CWA는 다음과 같은 단계로 수행된다.

- 단계1) 청소지도를 오른쪽 위에서 왼쪽 아래로 청소가 이루어지지 않은 위치가 나올 때까지 검색한다.
- 단계2) 청소가 이루어지지 못한 곳의 위치 값을 스택에 넣고 이점과 연결된 폐곡선을 장애물 지도에 기록하고 장애물 지도에는 이점들을 청소된 곳으로 표시한다.
- 단계3) 단계1)과 단계2)를 전체 청소 지도를 검색할 때까지 반복 수행한다.
- 단계4) 스택으로부터 위치 값을 받아서 장애물 주위로 이동한 후 장애물의 외곽벽을 따라 가며 청소를 수행한다.
- 단계5) 스택에 더 이상의 값이 없을 때까지 단계4)을 반복 수행한다.

4. 모의 실험 및 성능 분석

제안된 알고리즘의 성능을 알아보기 위해서 소프트웨어 시뮬레이션 및 하드웨어 시뮬레이션을 실시하여 제안된 알고리즘의 유용성을 검증하고, 나타난 문제점들을 제시한다.

4.1 소프트웨어 시뮬레이션

그림 6과 같은 시뮬레이션 프로그램을 작성하였다. 작업환경은 bitmap형태로 시스템에 주어진다. 작업환경 그림에서 검은색으로 나타나는 부분은 장애물이나 외곽벽을 나타낸다. 그리고 청소 작업의 시작 부분은 마우스를 이용하여 임의로 위치를 지정할 수 있다. 그림 6은 청소 알고리즘에 따라 청소 작업을 시뮬레이션할 간단한 작업영역이다. 그리고 그림 7에서 그림 10은 단계별 청소 작업의 시뮬레이션 결과를 나타내고 있다. 이들 그림에서 회색 영역은 평면 청소가 이루어진 장소이고 장애물이나 외곽벽 주위에 있는 좀더 어두운 회색 영역은 마무리 청소 작업이 이루어진 영역을 표시한다. 그리고 작업영역 중간 중간에 있는 선분은 다음 평면 청소를 위해서 로봇이 이동한 경로를 나타낸다. 일련의 청소 작업을 수행한 결과, 그림 10과 같이 전체 영역에 걸쳐 깨끗한 청소가 이루어짐을 시뮬레이션을 통해 알 수 있다. 그림 9는 마무리 작업을 수행하기 이전까지 청소 알고리즘에 따라 청소가 이루어진 영역을 나타낸다. 그림 9를 보면 장애물의 주위나 외곽벽 주위에 청소가 이루어지지 않은 채 남아 있는 곳을 찾을 수 있다. 이런 상태의 작업 환경에 마무리 작업 알고리즘에 따라 청소 작업이 이루어지면 그림 10과 같이 장애물 주위나 외곽벽 주위에 청소가 이루어져 완벽하게 청소 작업을 마감하게 된다.

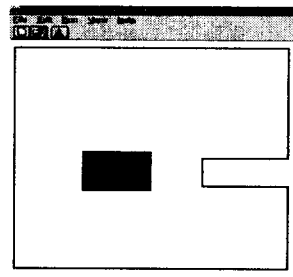


그림 6 간단한 작업영역
Fig. 6 Simple environment.

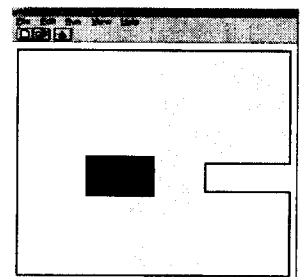
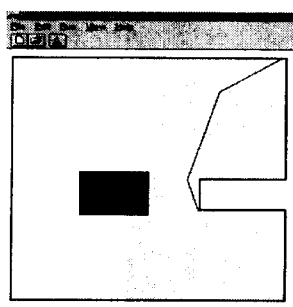
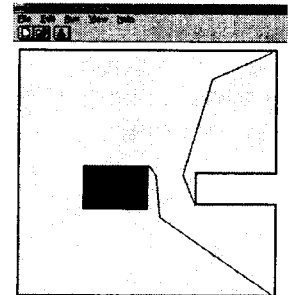


그림 7 첫 단계 평면 청소
Fig. 7 1st step of sweeping.



a) 두 번째 단계 평면 청소
a) 2nd step of sweeping.



b) 세 번째 단계 평면 청소
b) 3rd step of sweeping.

그림 8 스택에서 pop된 위치부터 다시 평면 청소
Fig. 8 The procedure of the PSA and ptp transfer algorithm.

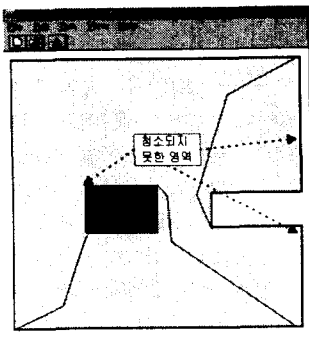


그림 9 PSA 적용 결과
Fig. 9 The results of PSA.

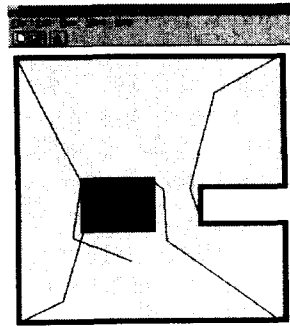


그림 10 마무리 작업 결과
Fig. 10 The results of CWA.

4.1.1 보다 실제적인 예

그러나 실제 청소 작업이 이루어질 작업 환경은 그림 6과 같이 단순하지 않은 않다. 제안된 알고리즘의 보다 실질적인 성능을 알아보기 위해서 좀더 복잡하고 실제적인 작업 환경에서 모의 실험을 실시한다. 그림 11은 일반적인 아파트 환경을 2차원 평면상에 나타낸 것이고 그림 12는 미로를 이루는 작업 환경이다. 이러한 작업영역에서 본 연구에서 제안된 청소 알고리즘을 적용해 본 결과, 그림 13과 그림 14와 같이 작업 환경의 복잡도나 장애물의 모양과 개수에 상관없이 완벽한 청소 작업이 이루어짐을 알 수 있다.

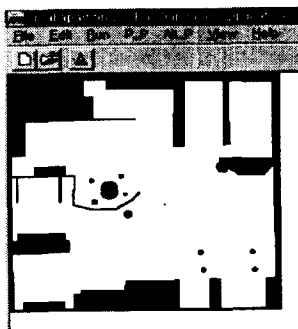


그림 11 복잡한 환경 I
Fig. 11 Complex environment I

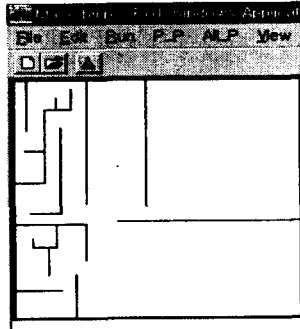


그림 12 복잡한 환경 II
Fig. 12 Complex environment II

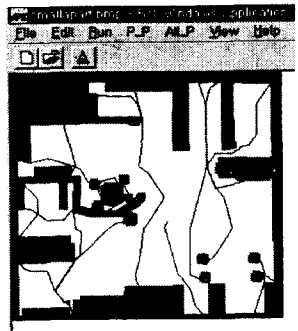


그림 13 그림 11의 결과
Fig. 13 Results for Fig.11

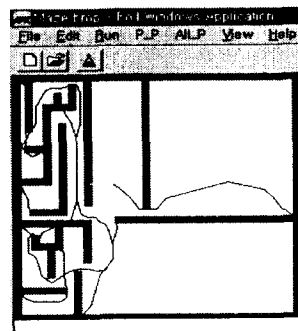


그림 14 그림 12의 결과
Fig. 14 Results for Fig.12

4.1.2 초기 시작 위치에 따른 성능 분석

청소 작업이 시작되는 위치가 청소 알고리즘에 미치는 영향을 알아보기 위해서 그림 12와 같은 미로환경 하에서 청소작업을 수행할 수 있는 모든 점을 시작점으로 하여 시작점부터 청소작업이 끝날 때까지 청소 작업을 수행하면서 이동한 거리를 계산하였다. 그 결과, 그림 15와 같은 시작점과 이동 거리의 관계를 그레이레벨(gray level)로 나타낸 그림을 얻을 수 있다. 해당 위치에서 흰색으로 표시된 부분일수록 짧은 거리이며 검은색에 가까울수록 이동거리가 긴 시작점이라는 것을 의미한다. 그림 15를 살펴보면 이동거리가 수직선상으로 주기를 가지고 반복됨을 알 수 있다. 로봇과 장애물사이의 간격이 로봇폭의 정수배와 가까울수록 이동거리는 짧게 나타난다. 그러나 최대 이동거리(20843 base unit)와 최소 이동거리(20193 base unit) 사이의 차이는 전체 이동 거리에서 볼 때 무시할 수 있을 만큼(3%내외) 작다. 따라서 본 연구에서 제안된 청소 알고리즘은 청소 작업이 시작되는 위치와는 거의 상관없이(3% 정도) 같은 성능을 나타낸다고 할 수 있다.

4.1.3 초기 시작 방향에 따른 성능 분석

로봇이 놓여지는 방향에 따라 어떤 성능을 나타내는지 알아보기 위해서 그림 12와 같은 작업 환경에서 시작점을 고정시켜 두고 로봇이 놓여지는 각도를 변화시키면서 이동 거리를 측정하였다. 그 결과 그림 16과 같이 놓여지는 방향과 이동 거리의 관계를 얻을 수 있다.

그림 16에서 알 수 있듯이 방향의 변화에 따라 다소의 이동 거리의 차이를 나타낸다. 이러한 차이는 청소 로봇이 단순한 사각형 모양을 하고 있다고 가정하기 때문에 장애물이나 외곽벽과 청소 로봇이 부딪칠 때 경우 그림 17과 같이 청소 로봇과 장애물이 이루는 각도에 따라서 청소가 이루어지지 않고 지나가는 영역의 차이가 발생하여 마무리 작업시 중복 청소되는 영역의 차이가 발생한다. 즉, 중복 청소되는 영역이 작을수록 이동거리도 작아지게 된다. 그림 17에서 보듯이 로봇과 벽이 이루는 각도가 45도일 때 청소가 이루어지지 않은 영역이 가장 많아 중복작업이 적게 발생한다. 그러나 이들 간의 차이도 전체 이동거리의 3% 이하이다.

이러한 모의 실험의 결과, 제안된 청소 알고리즘은 청소 작업의 시작점과 놓여지는 방향에 관계없이 동일한 성능(3% 이내에서)을 나타냄을 알 수 있다.

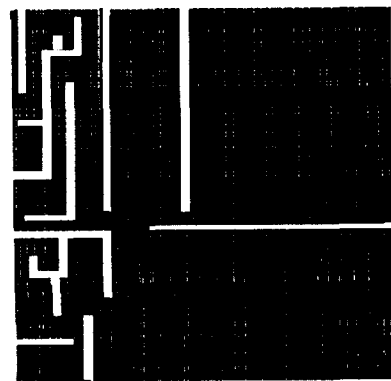


그림 15 시작점에 따른 이동거리
Fig. 15 The moving distance for each initial position.

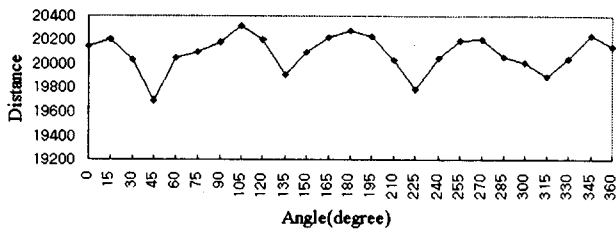


그림 16 놓여지는 방향과 이동거리와의 관계
Fig. 16 The relations between initial orientation and distance.

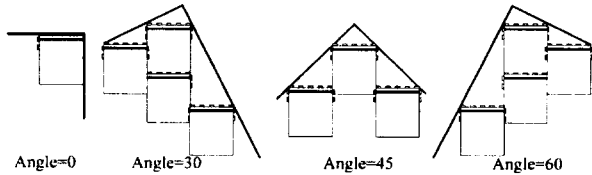


그림 17 벽과 로봇이 이루는 각에 따른 미청소 영역
Fig. 17 Unswept area according to angle between robot and the wall.

4.2 하드웨어 시뮬레이션

소프트웨어 시뮬레이션 뿐만 아니라, 실용성을 위하여 실험실용 이동로봇(Khepera)에 본 알고리즘을 이식하여 그 성능을 테스트한다. 본 연구에서 사용한 Khepera 로봇은 K.Team S.A. 회사의 제품이며 특히 실험실용으로 많이 사용되는 로봇이다. 크기는 지름이 약 5cm 정도의 원통형 로봇이며 8개의 적외선 센서를 장착하고 있다. 그 외관은 그림 18과 같다. Khepera 로봇을 사용한 하드웨어 시뮬레이션 환경은 아래 그림 19와 같다. 이 로봇을 이용하여 알고리즘을 구현하는 방법은 크게 두 가지로 나눌 수 있는데 PC에서 프로그램을 작성하여 다운로드하여 구동시키는 방법과 PC와 메시지 통신을 하면서 구동하는 방법이 있다. Khepera 시스템은 LabVIEW와 용이하게 인터페이스 할 수 있는 다양한 기능을 제공하고 있으므로 본 연구 계획에서도 LabVIEW를 이용하여 개발 환경을 구축하고 실제로 로봇을 구동시켰다. LabVIEW는 National Instrument사에서 제공하는 통합 시스템 환경 구축용 소프트웨어이다.

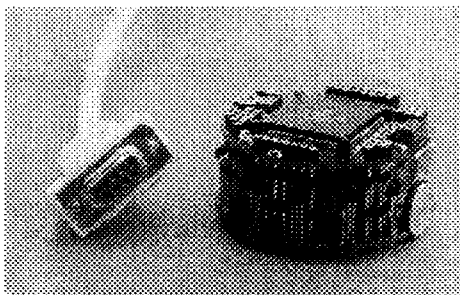


그림 18 Khepera 로봇
Fig. 18 Khepera robot

그림 19와 같은 환경 하에서 제시된 알고리즘을 구현해 본 결과 실제 환경 하에서도 전체 영역을 청소하는 것을 볼 수 있었다. 그러나 예상대로 문제점이 노출되었는데, 즉 주행 시에

주행오차(미끄럼 또는 제작 오차 또는 바퀴의 마찰력 차이 등에 의한)가 발생한다는 것이다. 따라서, 제시된 알고리즘이 이론적인 것으로 끝나지 않고 실용성을 갖추려면 주행오차에 대한 보정 및 그 영향 분석이 필수적이다. 아무리 정밀하게 제작된 이동로봇이라 할지라도 주행오차는 반드시 생긴다. 따라서 이러한 오차를 주기적으로 보정해 주지 않으면 오차가 누적되어 이동로봇으로 하여금 오동작을 유발하게 할 가능성이 높아진다. 예를 들어 다음 그림 20과 같이 현재 위치에서 화살표 방향으로 주행(방향)오차가 생겼을 경우 이후 주행오차가 생기지 않는다고 가정한다 하더라도, 이동로봇이 인식하는 환경과 실제환경과는 차이가 난다는 것을 알 수 있다.

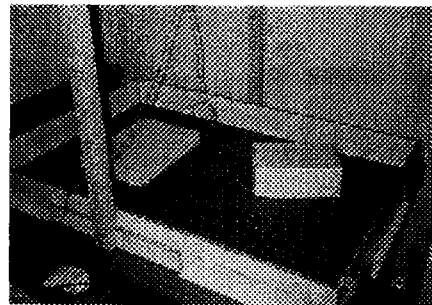
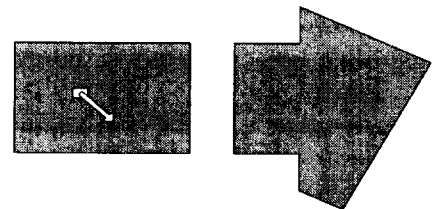


그림 19 하드웨어 시뮬레이션 환경
Fig. 19 Experimental environment.



a) 실제 환경 a) real environment
b) 이동 로봇이 인식한 환경 b) Build-in-map of the robot.

그림 20 주행 오차에 의한 영향
Fig. 20 The effect of moving error(orientation error).

5. 결론 및 향후과제

본 연구에서는 자율주행 이동 로봇이 주변의 환경에 대한 정보를 미리 인지하고 있지 않은 상황에서 센서 정보를 이용하여 주변 환경 정보를 받아들이면서 닫혀진(closed) 이차원 전체 평면을 빠짐없이 청소(sweep)하는, 정형화된 새로운 형태의 알고리즘을 제안하였다. 알고리즘은 평면 청소 알고리즘, 두 점간 이동 알고리즘, 마무리 작업 알고리즘의 세부분으로 구성되어 있다. 제시된 알고리즘의 유용성을 검증하기 위하여 시뮬레이션을 수행하는데, 먼저 소프트웨어 시뮬레이션을 실시하여 로봇이 정상적으로 동작한다는 가정 하에서 제안된 알고리즘을 구현해 보았고, 여기에 덧붙여 실험실용 이동로봇인 Khepera를 이용하여 보다 실제적인 하드웨어 시뮬레이션을 실시하였다. 시뮬레이션 결과를 볼 때, 제시된 알고리즘이 정확히 동작한다는 것을 알 수 있으며, 또한 정량적인 분석도 실시하였다. 그런데, 실제 구현 테스트를 통하여 발생한 문제들 중에 제일 중요

한 것은 주행오차 문제이다. 이동로봇의 주행에서는 항상 주행 오차가 생긴다. 따라서, 실용성을 고려한다면 이와 같은 주행 오차를 보정해 주는 방법에 대한 추가 연구가 필요하며, 만일 주기적으로 오차보정이 가능하다고 할 때 주행 오차에 따라 기존의 알고리즘이 얼마나 유용한지를 컴퓨터 시뮬레이션을 통하여 분석해 볼 수 있다. 또한, 향후 실용화를 위해서는 진공청소 기능이 부착된 자율주행 이동 로봇의 제작이 뒤따라야 한다. 특히 한국형의 가정에서는 물걸레 청소까지 가능한 기계적인 구조가 첨가되는 것이 보다 바람직하며 비용면이나 부피면에서도 적절한 수준을 유지해야하는 것도 실용화의 기본적인 단계이다.

감사의 글

본 연구는 '97년도 한국과학재단 핵심전문연구과제 (971-0920-138-1)의 지원으로 연구되었음.

참 고 문 헌

[1] T.Lozano-Perez, "Spatial planning : A configuration space approach," *IEEE Trans. Comput.*, vol. C-32, pp. 108-120, 1983 .

[2] T.Lozano-Perez, "A simple motion-planning algorithm for general robot manipulators," *IEEE Trans. Robotics and Automat.*, vol. RA-3, no. 3, pp. 224-238, June 1987.

[3] O.Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robotics Res.*, vol. 5, no. 1, pp. 90-98, 1986.

[4] J.Guldner and V.I.Utkin, "Sliding mode control for gradient tracking and robot navigation using artificial potential fields," *IEEE Trans. Robotics and Automat.*, vol. 11, no. 2, pp. 247-254, April 1995.

[5] K.Kant and S.W.Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 72-89, 1986.

[6] B.H.Lee and C.S.G.Lee, "Collision-free motion planning of two robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 17, no. 1, pp. 21-32, Jan./Feb. 1987.

[7] E.G.Gilbert and D.W.Johnson, "Distance functions and their application to robot path planning in the presence of obstacles," *IEEE Trans. Robotics and Automat.*, vol. RA-1, no. 1, pp. 21-30, March 1985.

[8] E.G.Gilbert and S.M.Hong, "A new algorithm for detecting the collision of moving objects," in *Proceedings of the 1989 IEEE Int. Conf. on Robotics and Automat.*, pp. 8-14, 1989.

[9] 정학영, 이장규, "네트워크 분할 기법을 이용한 이동로봇의 경로계획II-최적경로계산", 대한전기학회논문지, 제45권, 제7호, pp.1045-1049, 1996.

[10] O.Takahashi and R.J.Schilling, "Motion Planning in a plane using generalized Voronoi diagrams," *IEEE*

Trans. Robotics and Automat., vol. 5, no. 2, pp.143-150, 1989.

[11] E.G.Gilbert and C.P.Foo, "Computing the distance function between general convex objects in three dimensional space," *IEEE Trans. Robotics and Automat.*, vol. RA-6, no. 1, pp. 53-61, 1990.

[12] A Zelinsky, "A mobile robot exploration algorithm," *IEEE Trans. Robotics and Automat.*, vol. 8, no. 6, pp. 707-717, 1992.

[13] Iwan R. Ulrich, Francesco Mondada, J. D. Nicoud, Autonomouse vacuum cleaner, LAMI.

[14] E. Prassler, E. stroulia, M. Strobel, Office waste cleanup: an application for service robots, *Proceeding of the 1997 IEEE, International conference on Robotics and Automation*, pp. 1863-1868.

[15] Schofield, M., Cleaning robots, *Service Robots international Journal*, vol. 1,pp. 11-16.

[16] Hiroshi yaguch, Robot introduction to cleaning work in the East Japan Railway Company, *Advanced Robotics*, pp.403-414,1996.

[17] A. Zelinsky, "A mobile robot exploration algorithm", *IEEE transaction Robotics and Automation*, Vol. 8, 1992.

[18] O. Takahashi and R. J. Schilling., "Motion Planning in a Plane using general voronoi diagrams", *IEEE transaction Robotics and Automation*, 1989.

[19] Latombe. "Robot Motion Planning". Kluwer Academic publishers.

저 자 소 개



박 주 용 (朴 柱 勇)

1973년 4월 9일생. 1996년 영남대 공대 컴퓨터공학과 졸업, 1998년 동 대학원 컴퓨터공학과 졸업(석사)



이 기 동 (李 起 東)

1962년 10월 5일생. 1985년 서울대 공대 제어계측공학과 졸업, 1994년 동 대학원 제어계측공학과 졸업(공학박). 1995년 ~ 현재 영남대 공대 컴퓨터공학과 조교수