

사람과 로봇의 상호작용을 통한 청소 로봇 알고리즘

(Cleaning Robot Algorithm through Human-Robot Interaction)

김 승 용 [†] 김 태 형 ^{**}

(Seung-Yong Kim) (Tae-Hyung Kim)

요 약 청소 로봇은 지도 작성 및 위치 인식을 기준으로 청소 방법을 랜덤(random)방식과 매핑(mapping)방식으로 분류 할 수 있다. 랜덤방식은 지도를 작성하지 않아 가격경쟁력이 있지만 효율이 떨어진다. 반면, 매핑방식은 지도를 작성하므로 청소 효율이 높지만 상대적인 가격경쟁력이 떨어진다. 그러므로 랜덤방식과 매핑방식의 문제점들을 보완하기 위해 본 논문은 고가의 센서 정보를 사용하지 않고 사람이 청소 로봇에게 청소 공간에 대한 정보를 알려주며, 이 정보를 이용하여 기존의 청소 로봇보다 효율적이고 저렴한 청소 로봇을 제안한다. 또한 기존의 청소 로봇과 본 논문에서 제안한 청소 로봇과의 성능 비교를 통해서 본 논문에서 제안한 방식의 효율성을 보인다.

키워드 : 청소 로봇, 랜덤방식 청소, 매핑방식 청소, 사람과 로봇간의 상호 작용

Abstract We present a cleaning robot algorithm that can be implemented on low-cost robot architecture while the cleaning performance far exceeds the conventional random style cleaning through human-robot interaction. We clarify the advantages and disadvantages of the two notable cleaning robot styles: the random and the mapping styles, and show the possibility how we can achieve the performance of the complicated mapping style under the random style-like robot architecture using the idea of human-aided cleaning algorithm. Experimental results are presented to show the performance.

Key words : Cleaning robots, Random style cleaning, Mapping style cleaning, Human-robot interaction

1. 서 론

청소 로봇이 작업 공간을 자율적으로 이동하면서 완벽하게 청소하기 위한 많은 연구들이 있었다. 그러한 기존 연구들은 로봇의 위치 인식 여부를 기준으로 랜덤(random)방식 청소와 매핑(mapping)방식 청소로 분류

할 수 있다.

랜덤방식은 환경에 대한 지도 작성을 하지 않고, 충돌 감지 센서만을 의존하여 로봇이 벽이나 장애물과 충돌할 경우 이동경로를 변경하면서 청소를 하기 때문에 고가의 센서와 고성능의 처리장치를 필요로 하지 않는다[1,2].

랜덤방식의 청소 로봇이 고가의 센서 장치를 사용하지 않아 가격적인 이득이 있을지라도 로봇이 랜덤한 움직임으로 청소를 진행하기 때문에 청소 공간의 많은 중복과 청소한 공간, 청소할 공간의 연속성이 결여된다는 문제가 있다. 반면, 매핑방식은 환경에 대한 지도 작성과 위치 추정을 하고 청소를 진행하기 때문에 랜덤방식의 문제점을 보완하여 좀더 효율적인 청소가 가능하다. 하지만 고가의 장비나 비전(vision) 센서를 사용하기 때문에 가격경쟁력에서 뒤쳐진다. 그로 인해 상대적인 가격대 성능비의 많은 차이로 인하여 랜덤방식이 주로 사용되고 있다[3,4].

일반적으로 청소 작업이 일정한 공간 내에서 오랜 기간 반복해서 행하여지기 때문에 일정한 방향성 없이 청

* 이 논문은 2007 한국컴퓨터종합학술대회에서 '사람과 로봇의 상호작용을 통한 청소 로봇 알고리즘'의 제목으로 발표된 논문을 확장한 것임

[†] 학생회원 : 한양대학교 컴퓨터공학 연구원
kimsy@cse.hanyang.ac.kr

^{**} 종신회원 : 한양대학교 컴퓨터공학 교수
tkim@cse.hanyang.ac.kr

논문접수 : 2007년 9월 27일

심사완료 : 2008년 2월 27일

Copyright©2008 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제35권 제5호(2008.5)

소하는 것과 작업 공간의 정보를 얻기 위해 고가의 장비를 사용하는 것은 비효율적이다. 따라서 본 논문에서는 효율적인 청소를 위한 지도 작성시 요구되는 고가의 센서나 비전 센서를 사용하지 않고 청소 로봇에게 사람이 청소 공간에 대한 정보를 주고, 주어진 정보를 이용하여 기존의 청소 로봇보다 효율적이면서 저가의 청소 로봇을 제안하고자 한다.

2. 관련연구

2.1 청소 로봇의 주행 시스템

청소 로봇의 주행은 랜덤방식과 매핑방식 두가지로 분류할 수 있다[3].

그림 1의 Sensor module은 범퍼, 초음파, 적외선 센서와 같은 여러 종류의 센서 정보들을 관리하고 Driving processor module은 입력 받은 센서 값을 바탕으로 청소 로봇의 주행 알고리즘을 수행한다. 또한 Control module은 Driving processor module으로부터 받은 제어 명령을 통해 액추에이터(actuator)를 구동한다.

랜덤방식의 청소 로봇은 주어진 환경에 대한 지도 작성을 하지 않고, 랜덤하게 청소를 진행하기 때문에 중복 청소 공간과 청소 되지 않은 공간이 발생한다. 이러한 문제는 에너지 한계성을 가진 청소 로봇의 특징을 미루어 볼 때, 완벽한 청소를 보장하지 못한다. 하지만, 재청소를 위한 로봇의 충전이 필요 없는 좁은 청소 공간에서는 어느 정도 효과적인 청소를 행할 수 있다.

그림 2의 Sensor module와 Driving processor module, Control module 부분은 그림 1과 같지만 Driving processor module이 3가지의 모듈들과 상호 작용을 한다. 먼저 Maps는 입력 받은 센서 값을 바탕으로 지도를 작성하고 저장한다. 그리고 Localizers는 저장된 지도와 입력 받는 센서 값을 이용하여 위치 추정을 한다. 또한 Path planners는 저장된 지도와 로봇의 위치 추정을 통하여 주행 경로를 계획한다.

매핑방식은 로봇 스스로가 주어진 환경에 대한 지도 작성을 하여 로봇이 청소한 영역을 알고 있기 때문에



그림 1 랜덤방식의 주행 시스템

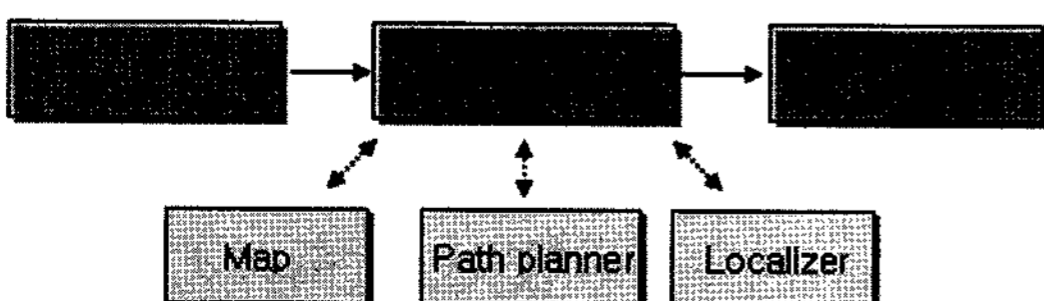


그림 2 매핑방식의 주행 시스템

청소 공간의 중복을 줄일 수 있다. 하지만, 로봇 스스로가 지도를 작성하고 로봇의 위치를 보정하기 위해서는 랜덤방식에서 사용하였던 적외선, 초음파, 범퍼 센서 이외에도 비전 센서와 이를 처리하기 위한 고성능의 처리장치를 사용한다.

2.2 하드웨어 요구사항(H/W requirement)

청소 로봇은 각각의 주행 방법에 따라 필요로 하는 센서들이 다르다. 그 이유는 지도 작성과 위치인식을 위해 추가적으로 필요한 센서들이 있기 때문이다. 아래의 그림 3은 각 방식에 따라 필요한 센서들을 나열한 것이다[2,3,5].

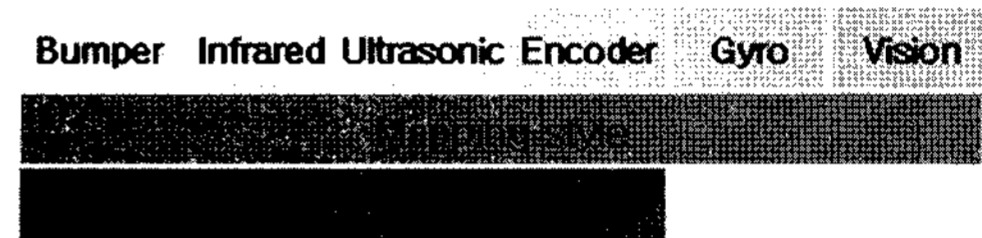


그림 3 청소 로봇에 사용되는 센서 구성

모든 청소 로봇이 위와 같은 하드웨어 구성을 따르는 것은 아니지만 랜덤방식과 매핑방식 로봇에서 일반적으로 요구되는 하드웨어 구성이라 할 수 있다. iRobot사의 Roomba, LG전자의 Roboking, Electrolux사의 Trilobite등이 랜덤방식의 청소 로봇이라 할 수 있고, 삼성전자의 Crubo, 한울로보틱스의 Ottoro가 매핑방식의 청소 로봇이라 할 수 있다. 랜덤방식은 장애물 회피를 위해 그림 3과 같은 센서를 필요로 하지만 매핑방식은 지도 작성 과 위치 인식을 위해 비전 센서를 추가로 필요로 한다. 본 논문에서 청소 로봇은 장애물과의 거리를 측정할 수 있는 레이저(laser) 센서나 초음파(ultrasonic) 센서, 각도를 측정하는 자이로(gyro) 센서를 필요로 하며 이동거리를 측정하기 위해 엔코더(encoder)를 사용한다. 그리고 낭떠러지나 계단과 같은 장애물을 회피하기 위해 적외선(infrared) 센서가 추가적으로 구성될 수 있다.

2.3 경로계획(Path Planning)

경로계획이란 로봇이 목적지까지 최소한의 거리로 이동하게 하는 것이다. 이러한 경로계획의 방법은 전역 경로 계획(global path planning)와 국부 경로 계획(local path planning) 두 부분으로 나눌 수 있다[6,7], 전역 경로 계획이란 전반적인 환경 정보를 가지고 로봇의 이동 경로를 계획하는 것이고, 국부 경로 계획은 초음파나 영상, 레이저 센서들로부터 입력되는 데이터에 의존하여 바로 앞에 존재하는 장애물의 크기와 위치 등을 구별하여 경로를 계획하는 것이다.

로봇이 청소 공간을 완벽하게 청소하기 위해서는 전역 경로 계획과 국부 경로 계획이 서로 상호 보완적인 관계를 가져야 한다. 예를 들어, 로봇이 청소 중에 계단

주위에 있다면 적외선 센서로 계단을 감지하여 떨어지는 것을 피하면서 능동적으로 경로를 재설정해야 한다. 또한, 로봇이 배터리 소모등의 이유로 청소를 끝낸 후에 청소하지 못한 공간이 남아 있을 경우나 방을 옮겨서 청소할 때 이동하기 위한 경로계획과 위치 인식이 필요하다.

2.4 지도 작성(Map Building)

청소 공간의 연속적이고 중복 없는 청소 작업을 위해 로봇의 지도 작성은 중요하다. 작성된 지도 안에서 로봇의 경로 및 위치를 나타내는 방법은 크게 두가지로 나눌 수 있는데 지도 정보를 처음부터 입력하여 주행 중에 업데이트(update)하는 방법과 지도 정보가 전혀 없는 상태에서 주행을 통해 지도를 작성하는 방법이 있다[3].

만약 로봇이 자율적으로 지도 작성을 한다면, 전체 주어진 환경을 이동하면서 정보를 수집하게 된다. 이때 만일, 거리계만으로도 정확한 위치 추정이 가능하다면, 데드 레컨(dead reckon) 값과 저장된 센서 값만으로도 정확한 지도 작성이 가능하다. 하지만 바퀴의 미끄러짐이나 청소 공간의 바닥 상태에 따라 데드 레컨 값의 오차 누적을 피할 수 없으므로, 위치 추정을 통한 로봇의 위치 오차를 해소해야 한다[6,8].

앞서 2.1절에서 언급한 것과 같이 비전 센서를 사용하여 지도를 작성하고 위치추정을 할 수 있다. 이때 지도를 작성하기 위해서는 입력 받는 영상 속에 특징점들이 필요하다. 특징점들을 추출하는 방법은 바코드 반사기, 초음파 표식, 시각 패턴 등과 이용하는 인공적인 랜드마크(landmark)와 환경에 특별한 장치를 하지 않고 자연의 것을 이용하는 자연적인 랜드마크가 있다. 입력되는 영상들로부터 특징점들을 추출하여 데이터베이스에 저장하고 이를 바탕으로 지도를 작성한다. 그리고 새로 입력되는 영상의 특징점과 저장된 특징점을 비교하여 위치를 추정하게 된다[3].

2.5 위치 추정(Localization)

모바일 로봇이 주행 시에 미끄러짐이나 바퀴의 마찰력, 혹은 바닥의 상태 등에 따라 주행 오차가 발생된다. 따라서 이러한 오차를 주기적으로 보정해 주지 않으면 오차가 누적되어 로봇으로 하여금 오작동을 유발하게 된다. 그러한 오차로 인해 실제 환경과 로봇이 인식하는 환경이 달라질 수 있다[8]. 그렇기 때문에 모바일 로봇의 주행에 있어서 로봇 자신의 위치를 정확히 추정하는 위치 추정은 로봇의 성공적인 자율 주행에 있어 가장 중요한 문제 중 하나이다.

위치 추정 문제를 해결하기 위한 방법으로 데드 레커닝(Dead-reckoning)방법[8]과 칼만필터(Kalman Filter)를 이용한 방법[9], 마코프 위치추정(Markov localization)[10], 몬테카를로 위치추정(Monte Carlo localiza-

tion)[11,12] 방법 등이 있다.

3. 제안내용

기존의 청소 로봇의 구동방식은 랜덤방식과 매핑방식 2가지가 존재한다. 랜덤방식은 지도 작성을 하지 않고 청소를 진행하기 때문에 다양한 map에 쉽게 적응할 수 있고 가격이 저렴하지만 완벽한 청소를 기대하기 어렵다. 반면, 매핑방식은 청소 공간을 인식하여 청소를 실행하기 때문에 청소 효율을 높일 수 있다. 하지만 지도 작성 및 자기 위치 추정을 위한 고가의 장비나 비전 센서를 필요로 하기 때문에 가격이 비싸진다는 단점이 있다.

본 논문에서는 매핑방식의 청소 로봇이 고성능의 장치를 필요로 하기에 청소 로봇의 가격이 비싸다는 단점과 랜덤방식이 가진 비효율적인 청소 로봇의 작업을 극복하기 위해서 청소할 공간의 정보를 사람이 청소 로봇에게 알려주는 방식을 제안한다. 이는 사람이 청소 로봇에게 해줄 수 있는 일과 청소 로봇이 잘할 수 있는 일을 분리하는 것으로, 인간과 로봇간의 상호 보완적인 작업을 통하여 매핑방식의 청소로봇 보다 간단하면서 랜덤방식 보다 효율적인 청소 로봇을 제안한다.

제안하는 청소 로봇의 구현을 위해 청소 로봇의 구조 및 사람이 청소 로봇에게 정보를 전달하는 방법에 대해 정의한다.

3.1 청소 로봇의 구조

청소 로봇에 적용되는 기본 명령어들과 청소 로봇의 움직임을 정의하는 Region filling 알고리즘, 구현을 위해 필수적인 하드웨어 구성을 정의한다.

• 기본 명령어 세트

제안하는 청소 로봇의 청소 수행을 위해 표 1과 같은 6가지의 기본 명령어를 사용한다. 기본 명령어들은 장애물과의 충돌 감지를 통해 해당 명령어를 수행하며 장애물은 청소 환경에서 벽이나 가구 등 센서를 통해 감지할 수 있는 것 이외에 사용자가 로봇에게 입력한 가상 외곽정보도 포함된다.

Forward 명령어는 장애물과 충돌할 때까지 수행되고 RT(right turn) 명령어는 로봇의 왼쪽과 정면, LT(Left turn) 명령어는 로봇의 오른쪽과 정면에 장애물이 있을

표 1 기본 명령어 세트

Instruction	Description
Forward	직선 운동
RT	오른쪽 90도 회전
LT	왼쪽 90도 회전
BRT	Back trace right turn
BLT	Back trace left turn
Goto	다음 청소 공간의 시작 point로 이동

간을 설정할 수 있다.

- 각도 측정 센서

청소 로봇이 다음 청소 공간으로 이동할 때와 기본 명령들을 수행하기 위해서는 자이로나 마그네틱 콤파스(magnetic compass)와 같은 각도 측정 센서가 필요하다. 왜냐 하면 로봇이 장애물과 충돌이 일어났을 시 현재 진행 방향 및 각도를 바탕으로 우측 회전과 좌측 회전을 구분하여 수행하기 때문이다.

3.2 정보 전달 방법

사용자가 청소 로봇에게 제공할 수 있는 정보들과 실제 청소 공간에 대한 정보를 청소 로봇에게 제공하는 방법 그리고 사용자가 청소 로봇과 상호 연동하기 위해 사용되는 User Interface에 대해 정의한다.

• 제공 가능한 정보

3.1절의 Region filling 알고리즘은 사용자가 로봇에게 제공하는 공간 정보를 바탕으로 이루어진다. 사용자가 로봇에게 제공할 수 있는 정보는 청소 시작위치, 분할된 청소 공간의 상대적 좌표 정보, 다음 청소공간까지 이동 경로이다. 이러한 정보를 통하여 하나의 청소 공간을 여러 개의 공간으로 나누어서 청소 할 수 있고 원하는 공간만을 여러 번 청소 할 수도 있다. 이것은 더러운 공간을 집중적으로 청소하는 것이 가능하며 청소 공간을 사용자가 원하는 방식으로 재분배 할 수 있음을 의미한다. 또한 로봇이 필요한 정보들을 사용자로부터 제공받지 않더라도 청소 수행이 가능하다. 예를 들어 그림 5의 로봇 초기 위치에서 사용자가 로봇에게 어떠한 정보도 제공하지 않는다면 로봇은 하나의 청소 공간이 정의된 것으로 간주하여 그림 5와 같은 기본 움직임만을 수행할 것이다. 하지만 이러한 청소 수행은 사용자 측면에서 간편하게 로봇을 설정할 수 있다는 장점이 있지만 청소되지 않고 지나치는 부분이 발생되어 전체적인 청소 수행 능력이 떨어진다.

• 청소 로봇에게 정보를 제공하는 방법

Region filling 알고리즘을 사용하여 완벽한 청소를 진행하기 위해서는 청소 공간에 대한 정보입력이 필요하다. 청소 공간 정보는 하나의 청소 공간을 여러 개의 공간으로 분할하여 입력할 수 있으며 분할된 청소 공간에 대한 이동 경로 입력으로 청소 공간에 대한 청소 재분배가 가능하다.

- 좌표값 설정

사용자가 하나의 청소 공간을 여러 개의 청소 공간으로 분할하였을 때, 분할된 공간의 면적을 의미하는 상대적 좌표정보와 Goto 명령어를 수행할 수 있는 이동 경로의 좌표값을 필요로 한다. 좌표값들은 로봇의 이동거리를 이용하여 표현될 수 있다. 로봇의 이동거리는 바퀴의 회전수를 측정하는 encoder만을 사용한다

면 바퀴의 미끄럼이나 표면의 상태에 따라 오차가 발생할 수 있기 때문에 레이저 센서나 초음파 센서와 같은 정밀한 거리 측정 센서를 사용해야 한다. 이를 통하여 장애물이나 벽과의 거리를 지속적으로 떨어진 거리를 측정하여 장애물의 유·무를 판단하고 로봇의 이동거리를 정확하게 측정할 수 있다.

- 측정된 정보 입력 및 청소 수행

만약 그림 6과 같은 청소 공간이 주어졌다면 분할된 청소 공간의 좌표값을 아래 표 4와 같이 입력할 수 있다. 전체 면적이 6m × 3m인 청소 공간을 그림 6의 화살표 순서에 따라 로봇을 이동시키며 좌표값을 입력하고 입력된 좌표값들을 통해 청소 공간을 분할할 수 있게 된다. 예를 들어 화살표 ①과 ②의 입력을 통해 Rectangle 1의 청소 공간을 지정할 수 있다.

표 4의 값들은 end point를 제외하고 distance와 angle로 이루어져 있다. 그 이유는 로봇의 이동거리만을 측정해 작업공간을 나누었기 때문에 이동방향을 나타내는 angle값을 통해 전체적인 동작의 방향성을 나타낼 수 있다.

분할된 공간에 대한 좌표는 width를 나타내는 X축 값과 height를 나타내는 Y축 값을 기준으로 정해진다. 예를 들어 표 4와 같이 Rectangle1의 경우는 width가 2m이고 height가 3m인 청소 공간을 설정함으로써 (0,0), (2,0), (3,0), (2,3)의 공간이 설정되고 마지막 좌표 값인 (2,3)은 end point로 지정되어 분할된 공간의 작업 완료 시점을 설정하게 된다. 또한 현재 지정한 청소 공간에서 지정해야 할 청소 공간이 더 존재한다면 Goto 좌표를 입력하여 다음 청소 공간까지의 시작 위치 및 시작 방향을 지정할 수 있다. 예를 들어 Rectangle1의 공간 설정 후 Rectangle2의 설정을 원한다면 Rectan-

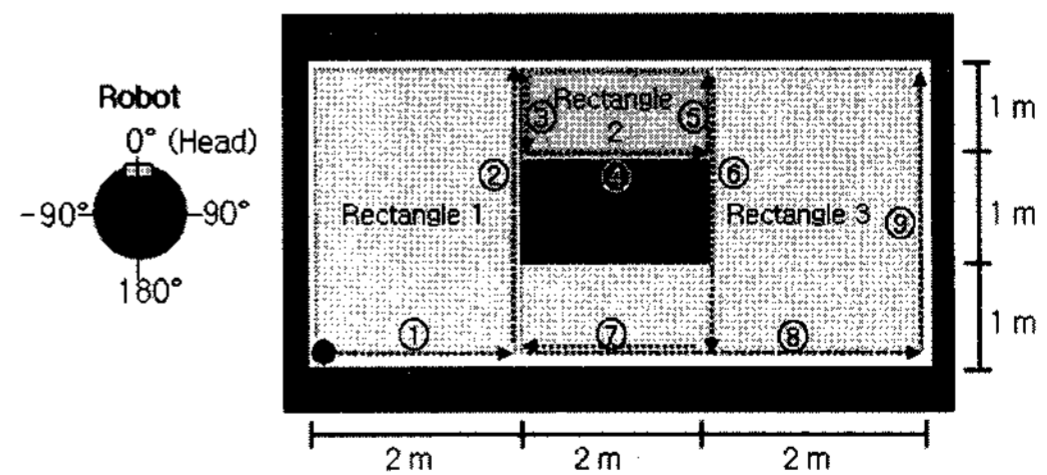


그림 6 분할된 청소 공간의 좌표입력

표 4 분할된 공간에 대한 좌표값(distance, angle)

Coordinate Region	X	Y	End point	Goto
Rectangle1	(2,90°)	(3,-90°)	(2,3)	(1,180°)
Rectangle2	(2,-90°)	(1,-90°)	(2,1)	(3,180°), (2,90°)
Rectangle3	(4,180°)	(3,-90°)	(4,3)	Nothing

gle1의 end point 기점으로 Rectangle2의 시작점을 Goto 좌표값으로 설정한다.

위와 같은 방식으로 사용자는 청소 공간에 대한 정보를 로봇에게 입력하게 되며 로봇은 이러한 정보와 3.1절의 청소 수행 알고리즘을 바탕으로 청소를 진행하게 된다. 로봇의 전체 작업완료는 end point에 도달하였을 때 Goto 좌표값을 통하여 결정하게 되는데 표 4에서 Rectangle1, Rectangle2의 경우와 같이 Goto 좌표값이 존재한다면 저장된 거리와 방향을 통해 다음 공간으로 이동할 것이지만 Rectangle3과 같이 Goto 좌표값이 저장되어 있지 않다면 모든 청소가 종료된다.

• User Interface

사용자는 작업 공간의 구조 및 사용자 편의에 따라 청소 로봇의 알고리즘을 선택할 수 있다. 아래의 그림7은 청소 로봇의 메뉴구성과 청소 로봇 제어를 위한 remote controller를 나타낸다. 제어를 위해 remote controller를 사용한 이유는 사람이 정보를 입력하기 위해 직접 로봇을 끌고 다니는 것은 비현실적이며 PDA와 같은 장비를 통해 임의의 좌표값을 전송하는 것은 실제 측정된 값이 아닌 데이터므로 원하지 않는 실행결과가 발생 될 수도 있기 때문이다. 그러므로 remote controller를 이용하여 실제적인 환경에 적용시키면서 로봇이 직접 이동거리 및 각도를 측정하는 것이 제안하는 방법에 가장 부합된다.

Random 메뉴는 청소 로봇에 랜덤방식 알고리즘을 적용한 것으로 좁은 청소 공간일 경우 효과적으로 사용될 수 있으며 사용자로부터 어떠한 입력을 요구하지 않기 때문에 간단하게 사용될 수 있다. 반면, Region 메뉴와 Full region 메뉴는 모두 region filling 알고리즘을 청소 로봇에 적용한 것이지만 사용자 입력 수준에 따라 다르게 선택될 수 있다.

Region 메뉴는 사용자가 로봇에게 아무런 정보를 입력하지 않은 것으로 현재 위치에서 3.1절의 그림 5와 같이 기본적인 움직임만을 수행할 것이며 청소 공간 안에 장애물이 없고 직사각형 모형의 청소 공간에서 효과적으로 사용될 수 있다. 반면, Full region 메뉴는 region filling 알고리즘에 사용될 3가지 정보를 모두 입력할 수

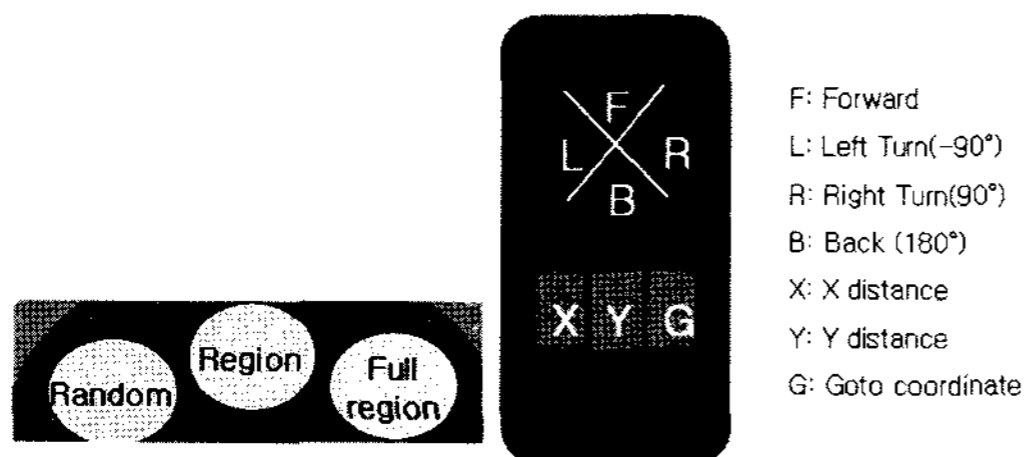


그림 7 청소 로봇과 remote controller의 User Interface

있는 것으로 복잡한 청소 환경에 알고리즘을 적용시킬 수 있다. 하지만 Full region 메뉴를 사용하기 위해서는 정보를 입력하기 위한 별도의 작업이 필요하다.

그림 7에서 오른쪽 그림이 Full region 메뉴와 연동되는 remote controller의 모습이다. F 버튼을 통해 로봇의 이동거리를 측정하며 L, R, B 버튼을 통해 로봇의 turn의 각도를 조절한다. 단, turn의 각도는 -90°, 90°, 180°로 제한하였는데 그 이유는 청소 공간 지정에 앞의 3개의 각도로 충분하며 다양한 각도는 복잡성 증가로 인해 오히려 사용자의 편의를 떨어트릴 수 있기 때문이다. 또한 X, Y, G 버튼을 통해 X축과 Y축의 이동거리, 다음 청소 공간으로의 이동 경로를 저장할 수 있다.

4. 실험 및 결과

본 논문에서 제안한 방식의 효율성을 증명하기 위해서 4가지의 시뮬레이션 Type을 정의하고 map별로 각 Type에 대한 성능을 측정한다. 성능 평가는 시간에 따른 청소 진행률과 일정한 작업을 달성하기 위해 필요한 시간을 기준으로 한다.

4.1 실험 환경

본 논문에서 제안된 방법을 적용한 청소 로봇과 기존 연구된 청소 로봇의 성능 비교를 위해, USC(University of Southern California)에서 오픈 소스로 제공하는 Player/Stage project의 Player/Stage simulator를 사용하였고 시뮬레이터의 환경은 Fedora 3.0 + kernel 2.6.9-1.667에서 진행하였다. 성능 비교를 위해 player/stage를 선택한 가장 큰 이유는 레이저 센서, 적외선 센서, 범퍼 센서 등에 대한 다양한 가상 디바이스(virtual device)가 지원되며 이러한 디바이스들은 많은 실험을 통해 검증되었기 때문이다[13,14].

• Map

각 map에서 흰색 공간이 청소할 영역을 나타낸 것이며 4가지의 map 모두 동일한 청소 면적을 가진다. 청소해야 하는 면적은 (a)를 기준으로 8.2(m) × 4.5(m)이다.

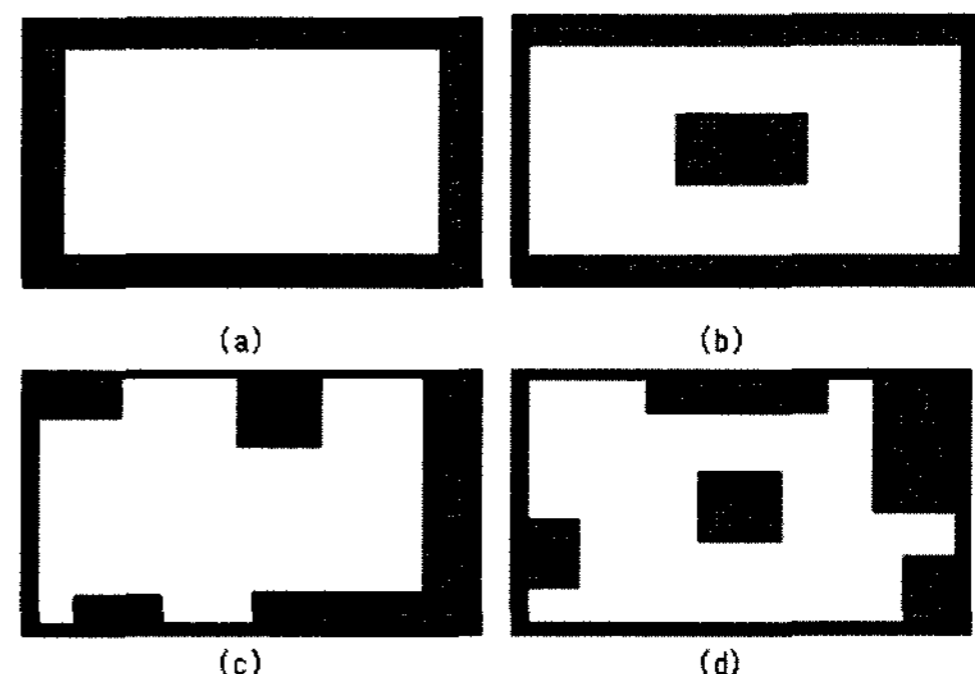


그림 8 시뮬레이션에 사용되는 map

• 로봇 specification

표 5 로봇 specification

크기	0.32 (m) × 0.32 (m)
속도	0.3 m/sec

• 시뮬레이션 type

표 6 시뮬레이션 type 정의

Type	설명
Type0	Roomba algorithm을 적용한 랜덤방식
Type1	Roomba algorithm + 가상 벽(virtual wall)
Type2	Region filling algorithm (시작 위치만 제공)
Type3	Region filling algorithm (시작 위치, 청소 공간 정보, 이동 경로)

Type0는 roomba의 알고리즘 적용한 랜덤방식이고 Type1은 Type0의 랜덤방식에 가상 벽을 적용하여 임의로 청소 공간을 나누는 것이다. 또한 Type2와 Type3는 region filing 알고리즘을 적용한 것으로 사용자가 로봇에게 제공하는 정보의 양과 종류에 따라 분류하였다.

4.2 실험 및 결과 분석

• 시뮬레이션 실행 화면

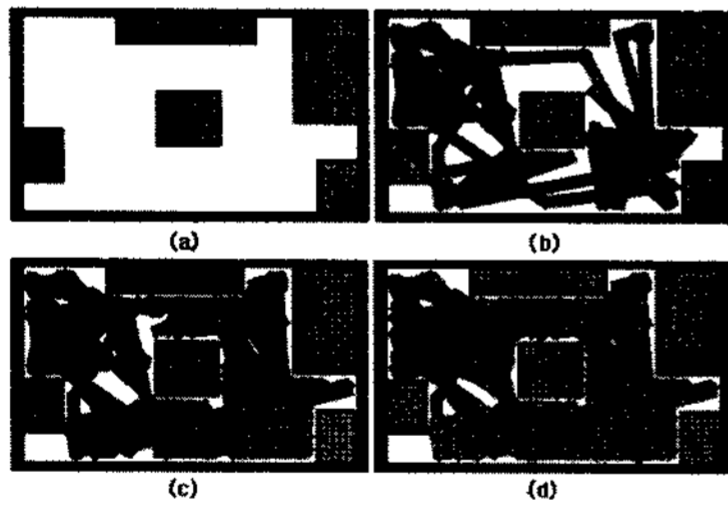


그림 9 Map 4에 대한 Type0의 실행 화면 : 10분 간격

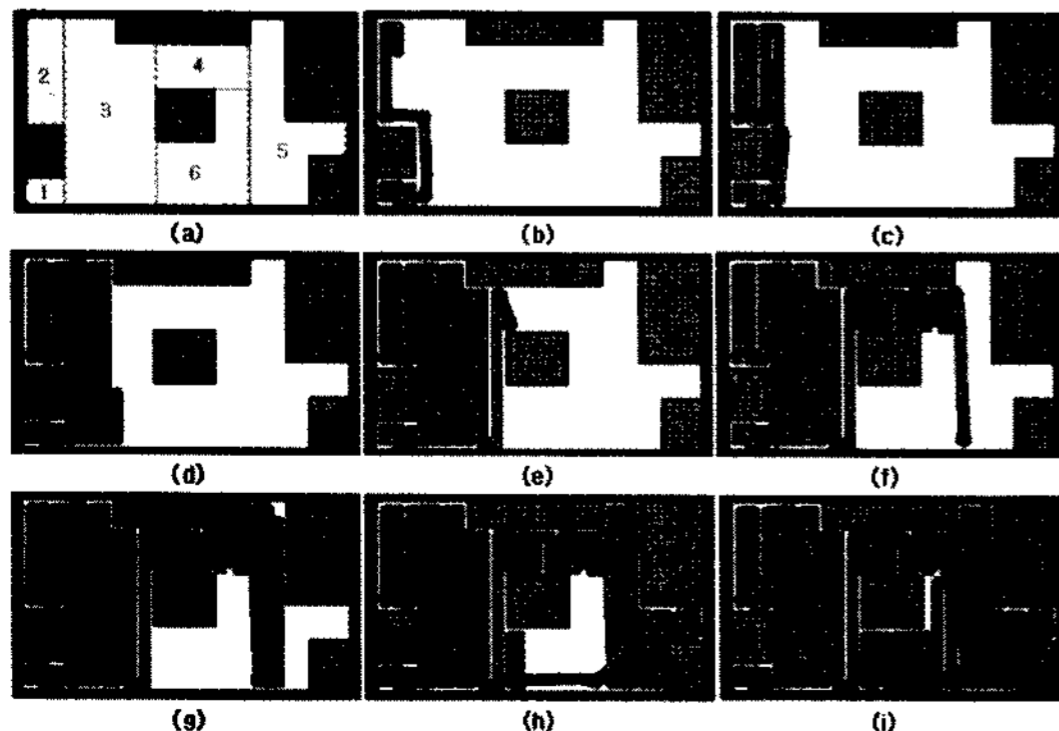


그림 10 Map 4에 대한 Type3의 실행 화면 : 1분 간격

그림 9는 roomba의 알고리즘을 적용한 Type0의 실행 화면이며 그림 10은 region filling 알고리즘을 적용한 Type3의 실행 화면이다. 또한 그림 10의 Type3는 (a)와 같이 사용자가 임의로 나누는 6개의 청소 공간 정보와 이동 경로를 입력하였다.

• 시간에 따른 청소 진행률

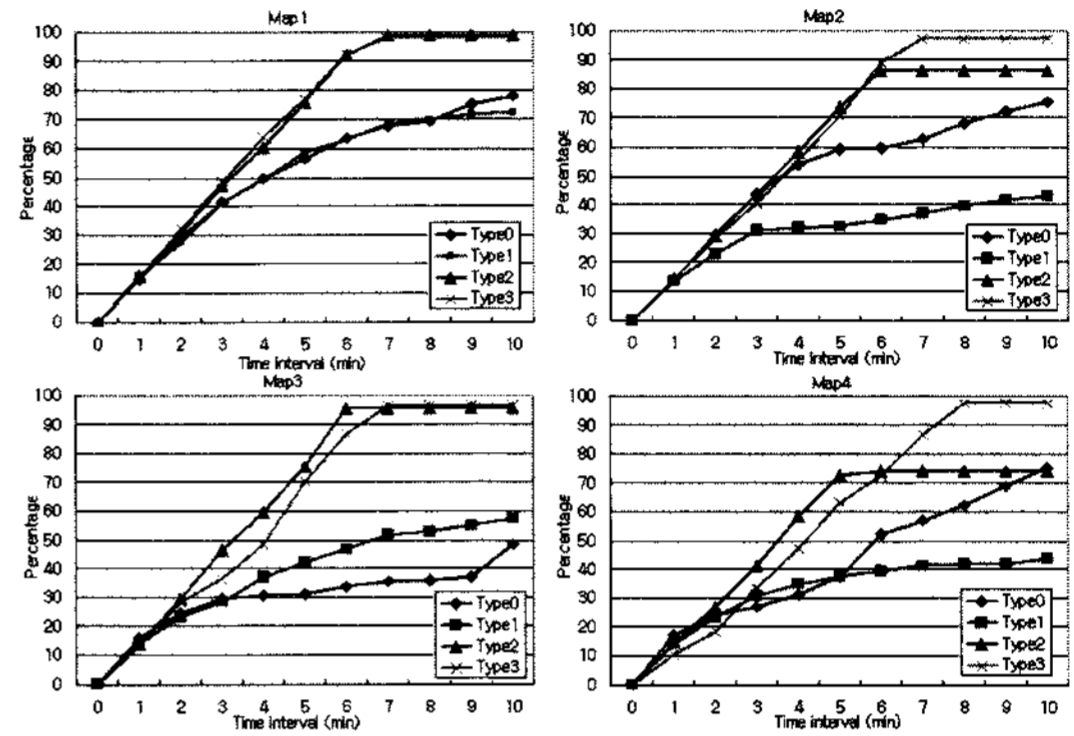


그림 11 Map 1, 2, 3, 4에 대한 청소 진행률

시간 간격을 1~10분까지만 성능비교를 하였다. 그 이유는 Type2, Type3의 청소 진행이 8분 안에 모두 종료되기 때문이다. 또한 Type0와 Type1이 95% 이상의 청소 작업을 수행하기 위해서는 대략 60~70분 이상의 시간이 필요하기 때문에 청소 진행률은 10분까지만 성능평가를 실시하였다.

• 일정 면적을 청소하기 위해 소요된 시간

표 7 Map 1, 2, 3, 4에 대한 일정면적을 청소하기 위해 소요된 시간(min)

map 1	Complete coverage (%)				map 2	Complete coverage (%)			
	30	60	90	95		30	60	90	95
Type0	2	6	20	82	Type0	3	6	21	48
Type1	3	6	18	31	Type1	3	33	42	70
Type2	2	4	6	7	Type2	2	5	don't	don't
Type3	2	4	6	7	Type3	2	5	7	7

map 3	Complete coverage (%)				map 4	Complete coverage (%)			
	30	60	90	95		30	60	90	95
Type0	3	12	55	93	Type0	4	8	33	90
Type1	4	21	24	29	Type1	3	36	52	58
Type2	2	4	6	6	Type2	2	4	don't	don't
Type3	3	5	7	7	Type3	3	5	7	8

don't 표기는 정해진 비율까지 청소를 진행하지 못하는 것이다. map 2에서 Type2는 86.4%에서 청소를 완료하고 map 4에서 Type2는 73.82%에서 청소를 완료한다.

• 결과 분석

Roomba의 알고리즘은 초기 전체 면적의 50%정도가량은 region filling 알고리즘과 비교적 비슷한 시간에

수행되었지만 90% 이상의 작업을 수행하기 위해서는 소요 시간이 비하급수적으로 증가하였다. 이것은 작업을 진행 할수록 중복된 작업의 경우가 높아지기 때문이다.

표 7에서 map 4의 Type3 경우는 지도 공간을 6개로 나누어 로봇에게 알려주어 실행하였고 Type2의 경우는 단순히 시작 포인트만을 알려주고 실행 시켜 보았다. 그 결과 Type2는 작업 달성률이 최대 73.82%에서 작업을 완료되었지만 Type3는 97.32%까지 진행되었다. 이를 통해 로봇에게 청소 공간의 정보제공이 성능에 어떠한 영향을 미치는지 알아볼 수 있었다. 또한 표 7에서 map 2의 성능평가 표를 통하여 Type1이 공간을 가상 벽을 통해 분할하였을 때 잘못된 분할은 분할하지 않았을 때 보다 더 성능이 떨어짐을 확인하였습니다. 이것은 사용자가 지도 공간을 비효율적으로 나누었을 때 그렇지 않은 경우 보다 더 효율이 떨어진다는 것을 알 수 있었다.

5. 결론 및 향후 과제

Roomba와 같은 랜덤방식의 청소 로봇은 다양한 map에 적응하여 진행할 수 있는 장점이 있다. 하지만 이것은 복잡한 환경일수록 소요 시간은 기하급수적으로 증가함을 실험을 통해 알 수 있었다. 예를 들어 표 7에서 Type0이 map 4의 청소 면적을 90%에서 95%으로 5%의 면적을 더 처리하기 위해서는 57분의 시간이 필요하였다. 이것은 엄청난 중복수행이 존재 하였음을 나타낸다.

Region filling 알고리즘은 사용자가 아무런 정보를 주지 않으면 수행시간이 짧아지고 사용자 편의성에서 간편해 진다. 하지만 완벽한 청소를 기대하기 힘들다. 표 7의 실험 결과와 같이 사용자가 보다 완벽한 청소를 원한다면 로봇에게 효율적으로 정보를 제공해야 한다. 결론적으로 랜덤방식은 map이 복잡 할수록 많은 중복 작업으로 인해 청소 효율이 떨어지며 region filling 알고리즘은 사용자가 로봇에게 올바른 정보를 제공하였을 때 성능이 좋아짐을 실험을 통해 알아보았다. 하지만 사용자가 불필요하거나 효율적이지 못한 정보를 제공하였을 때에는 오히려 성능이 떨어짐을 알 수 있었다. 다시 말해, 사용자가 로봇에게 필요한 정보를 제공할 수 있다면 고성능의 장비나 센서 없이도 매핑방식의 청소 로봇과 유사한 성능을 보일 수 있다.

본 논문에서는 청소 공간 안에 하나의 로봇이 존재하여 작업을 진행하였지만 향후 여러 개의 로봇이 한 청소 공간에 작업을 진행하는 cooperation 방식에 대해 연구를 확장할 계획이다.

참 고 문 헌

[1] Joseph L. Jones, "Robots at the Tipping Point: The Road to the iRobot Roomba," IEEE Robotics

- & Automation Magazine, Volume:13, pp.76-78, 2006.
- [2] Jodi Forlizzi, Carl DiSalvo, "Service Robots in the Domestic Environment: A Study of the Roomba Vacuum in the Home," Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction HRI, '06, pp. 258-265, 2006.
- [3] ALBERTO ELFES, "Sonar-Based Real-World Mapping and Navigation," IEEE JOURNAL of robotics and Automation, Volume:3, No:3, 1987.
- [4] Jose Neira, Juan D. Tardos, Joachim Horn, and Gunther Schmidt, "Fusing Range and Intensity Images for Mobile Robot Localization," IEEE transaction on robotics and automation, Volume:15, No:1, 1999.
- [5] Sewan Kim, "Autonomous Cleaning Robot: Robo-king System Integration and Overview," Proceedings of the 2004 IEEE International Conference on Robotics & Automation New Orleans, Volume:5, pp. 4437-4441, 2004.
- [6] Joon Seop Oh; Yoon Ho Choi; Jin Bae Park; Zheng, Y.F. "Complete coverage navigation of cleaning robots using triangular-cell-based map," Industrial Electronics, IEEE Transactions on, Volume 51, Issue 3, June 2004 Page(s):718-726
- [7] Yong K. Hwang, Narendra Ahuja, "Gross Motion Planning-A Survey," ACM Computing Surveys, Volume:24, No:3, Sep. 1992.
- [8] Ching-Chih Tsai, "A localization system of a mobile robot by fusing dead-reckoning and ultrasonic measurements," Instrumentation and Measurement, IEEE Transactions on, Volume:47, pp. 1399-1404, 1998.
- [9] F.Lu, E.Milios, "Globally consistent range scan alignment for environment mapping," Autonomous Robots, Volume:43, pp. 333-349, 1997.
- [10] D.Fox, W.Burgard, S.Thrun, "Active markov localization for mobile robots," Robotics and Autonomous Systems, Volume:25, pp. 195-207, 1998.
- [11] S.Thrun, D.Fox, W.Burgard, F.Dellart, "Robust monte carlo localization for mobile robots," Proc. of National Conf. on Artificial Intelligence, Volume: 128, 2001.
- [12] F.Dellart, D.Fox, W.Burgard, S.Thrun, "Monte carlo localization for mobile robots," Proc. IEEE Int. Conf. Robotics and Automation, pp.1322-1328, 1999.
- [13] Brian P.Gerkey, Richard T.Vaughan, Andrew Howard, "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," In Proceeding of the International Conference on Advanced Robotics (ICAR 2003), pp. 317-323, 2003.
- [14] Toby H.J. Collett, Bruce A. MacDonald, and Brian P. Gerkey, "Player 2.0: Toward a Practical Robot Programming Framework," In Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005), Sydney, Australia, December 2005.



김 승 용

2005년 가톨릭대학교 컴퓨터공학과(학사). 2006년~현재 한양대학교 컴퓨터공학과(석사과정). 관심분야는 청소로봇, 협동로봇, 로봇 주행알고리즘, 인간로봇 상호 작용



김 태 형

1986년 서울대학교 컴퓨터공학과(학사)
1988년 서울대학교 컴퓨터공학과(석사)
1990년 현대전자산업(주) 응용소프트웨어 개발팀. 1996년 Univ. of Maryland, College Park, Computer Science Department (Ph.D.). 1999년 현대정보기술(주) 정보기술연구소 책임연구원. 현재 한양대학교 전자컴퓨터공학부 부교수. 관심분야는 Middleware System, Embedded System Software, Mobile Computing, Software Architecture