

# 인간과 상호작용을 통한 청소 로봇 알고리즘

김승용<sup>o</sup> 김태형

한양대학교 컴퓨터 공학과

[kimsy@cse.hanyang.ac.kr](mailto:kimsy@cse.hanyang.ac.kr) [tkim@cse.hanyang.ac.kr](mailto:tkim@cse.hanyang.ac.kr)

## Human-Aided Robot Cleaning Algorithm

Seung-Yong Kim<sup>o</sup> Tae-Hyung Kim

Dept. of Computer Science & Engineering, Hanyang University

### 1. 서론

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 분류할 수 있다.

랜덤방식은 환경에 대한 지도 작성을 하지 않고, 충돌 감지센서만을 의존하여 로봇이 벽이나 장애물과 충돌할 경우 이동경로를 변경하면서 청소를 하기 때문에 고가의 센서와 고성능의 처리장치를 필요로 하지 않는다. 하지만 고가의 센서 장치를 사용하지 않아 가격적인 이득이 있을지라도 랜덤한 움직임으로 청소를 진행하기 때문에 청소 실행의 많은 중복과 연속성이 결여된다는 문제가 있다. 반면, 매핑방식은 환경에 대한 지도 작성과 위치 추정을 하고 청소를 진행하기 때문에 랜덤방식의 문제점을 보완하여 좀 더 효율적인 청소가 가능하다. 하지만 비전(vision)센서와 같은 고가의 장비를 사용하기 때문에 가격경쟁력에서 뒤쳐진다. 그로인해 상대적인 가격대 성능비의 많은 차이로 인하여 랜덤방식이 주로 사용되고 있다.

### 2. 관련연구

청소 로봇의 주행은 랜덤방식과 매핑방식 두가지로 분류할 수 있다. 랜덤방식의 주행 시스템은 범퍼, 초음파, 적외선 센서의 정보를 관리하는 Sensor module과 입력받은 센서 값을 바탕으로 청소 알고리즘을 수행하는 Driving processor module, Driving processor module로부터 받은 제어 명령을 통해 actuator를 구동하는 Control module로 구성된다. 반면, 매핑방식의 주행 시스템은 완벽한 청소 알고리즘을 수행하기 위해 입력 받은 센서 값을 바탕으로 지도를 작성하는 Map module과 저장된 지도와 입력받은 센서 값을 이용하여 위치 추정하는 Localizer, 주행 경로를 계획하는 Path planner들이 랜덤방식의 주행 시스템에 추가적으로 구성된다.

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

위치 추정은 로봇이 주행 시에 미끄럼이나 바퀴의 마찰력, 바닥의 상태 등에 따라 주행 오차가 발생된다. 따라서 이러한 오차를 주기적으로 보정해 주지 않으면 오차가 누적되어 로봇으로 하여금 오작동을 유발하게 된다.

경로 계획은 로봇이 목적지까지 최소한의 거리로 이동하게 하는 것이다. 경로계획의 방법은 전반적인 환경 정보를 가지고 경로를 계획하는 전역 경로 계획과 초음파나 비전, 레이저 센서들로부터 입력되는 정보에 의존하여 바로 앞에 존재하는 장애물의 크기와 위치 등을 구별하여 경로를 계획하는 국부 경로 계획으로 나눌 수 있다.

### 3. 제안내용

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. 인간과 로봇간의 상호 보완적인 작업을 통하여 매핑방식의 청소로봇 보다 간단하면서 랜덤방식 보다 효율적인 청소 로봇을 제안한다.

제안하는 청소 로봇은 6가지 기본 명령어 세트와 region filling 알고리즘으로 구성된다. 기본 명령어 세트는 Forward, RT, LT, BRT, BLT, Goto으로 이루어진 6가지 명령어들이며 region filling 알고리즘을 수행하기 위해 사용된다. Forward는 장애물과 충돌할 때까지 수행되고 RT(Right Turn)은 로봇의 왼쪽과 정면, LT(Left Turn)은 로봇의 오른쪽과 정면에 장애물이 있을 시 수행된다. 또한 RT와 LT는 이동방향에 따라 정면충돌만 있을 시에도 수행된다. 그리고 BRT(Back Trace Right Turn)은 LT할 방향에 장애물이 있다면 장애물을 피할 때까지 되돌아간 후 RT를 수행한다. 또한 BLT(Back Trace Left Turn)은 BRT와 반대로 RT할 방향에 장애물이 있다면 장애물을 피할 때까지 되

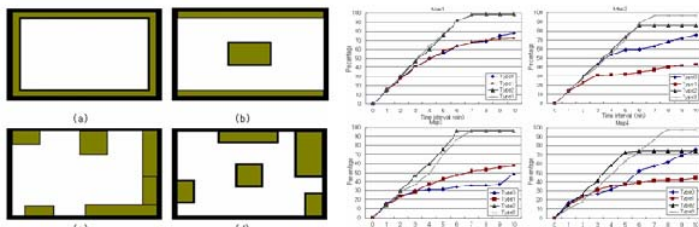
돌아간 후 LT를 수행한다. 마지막으로 Goto는 청소 공간을 사용자가 여러 개로 지정하였을 때 현재 청소 공간의 청소가 완료된 후 다음 청소 공간으로 이동할 때 수행된다.

region filling 알고리즘은 전체적인 청소 진행 방향이 왼쪽에서 오른쪽으로 진행되며 로봇이 직선으로 이동하다 사용자가 입력한 청소영역이나 경계, 장애물을 만났을 시 회전하는 것을 기본 움직임으로 한다. 또한 사용자가 입력한 청소영역 안에 장애물이 발견되었을 때에는 장애물에 반응하여 회전을 수행한다. region filling 알고리즘은 사용자가 로봇에게 제공하는 공간정보를 바탕으로 이루어진다. 사용자가 로봇에게 제공할 수 있는 정보는 청소 시작위치, 분할된 청소 공간의 상대적 좌표정보, 다음 청소공간까지 이동 경로이다. 이러한 정보를 통하여 하나의 청소 공간을 여러 개의 공간으로 나누어서 청소 할 수 있고 원하는 공간만을 여러 번 청소하는 것이 가능하다. 또한 로봇이 필요한 정보들을 사용자로부터 제공받지 않더라도 하나의 공간으로 간주하여 기본 움직임만을 수행한다. 하지만 이러한 청소 수행은 사용자 측면에서 간편하게 로봇을 설정할 수 있다는 장점이 있지만 청소되지 않고 지나치는 부분이 발생되어 전체적인 청소 수행 능력이 떨어진다.

앞서 설명한 청소 로봇을 구성하기 위해서는 거리측정 센서와 엔코더, 각도측정 센서가 필요하다. region filling 알고리즘은 초음파 센서나 레이저 센서와 같이 거리측정 센서를 기반으로 수행된다. 이러한 센서를 사용하여 장애물과의 충돌을 판단한다. 예를 들어 장애물과의 거리가 0.3m미만일 경우 충돌을 감지하고 회전을 수행한다. 또한 제안된 청소 로봇은 거리측정 센서와 엔코더를 통하여 로봇의 이동거리를 측정한다. 측정된 이동거리는 사용자가 청소 공간을 지정할 때 사용된다. 예를 들어 상대적인 좌표값을 {0,0},{0,3},{2,3},{2,0}으로 지정한다면 2(m)x3(m)의 청소 공간을 설정할 수 있다. 마지막으로 청소 로봇이 다음 청소 공간으로 이동 할 때와 기본 명령들을 수행하기 위해서는 자이로나 마그네틱 콤파스와 같은 각도측정 센서가 필요하다. 왜냐하면 로봇이 장애물과 충돌이 일어났을 시 현재 진행 방향 및 각도를 바탕으로 우측회전과 좌측회전을 구분하여 수행하기 때문이다.

4. 실험 및 결론

USC(University of Southern California)에서 오픈소스로 제공하는 Player/Stage 시뮬레이터를 사용하였고 시뮬레이터 환경은 Fedora 3.0 + kernel 2.6.9-1.667에서 진행하였다. 본 논문에서 제안한 방식의 효율성을 증명하기 위해서 Type0부터 Type3까지의 4가지 시뮬레이션 type을 정의하였고 map별로 각 type에 대한 성능을 측정하였다. Type0는 roomba의 알고리즘을 적용한 랜덤방식이고 Type1은 Type0의 랜덤방식에 가상벽을 적용하여 임의로 청소 공간을 나눈 것이다. 또한 Type2와 Type3는 region filling 알고리즘을 적용한 것으로 사용자가 로봇에게 제공하는 정보의 양에 따라 분류한 것이다. Type2는 시작 위치만을 입력한 것이고 Type3는 시작위치, 청소 공간 정보, 이동 경로를 입력한 것이다.



<그림 1. Map>

<그림 2. 청소 진행률>

| 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     |

<표 1. 일정면적을 청소하기 위해 소요된 시간>

각 map에서 흰색 공간이 청소할 영역을 나타낸 것이며 4가지 map 모두 8.2(m)x4.5(m)의 동일한 청소 면적을 가진다. 또한 실험에 사용될 가상 로봇의 크기는 0.32(m)x0.32(m)이고 속도는 0.3m/sec이다. 랜덤방식의 알고리즘은 초기 전체 면적의 50%정도는 region filling 알고리즘과 비교적 비슷한 시간에 수행되었지만 90%이상의 작업을 수행하기 위해서는 소요시간이 기하급수적으로 증가하였다. 이것은 작업을 진행 할수록 중복된 작업의 경우가 높아지기 때문이다. <표 1>에서 map4의 Type3 경우는 지도 공간을 6개로 나누어 로봇에게 알려주어 실행하였고 Type2의 경우는 단순히 시작 포인트만 알려주고 실행 시켜 보았다. 그 결과 Type2는 작업 달성률이 최대 73.82%에서 작업이 완료되었지만 Type3는 97.32%까지 진행되었다. 이를 통해 로봇에게 청소공간의 정보제공이 성능에 어떠한 영향을 미치는지 알아볼 수 있었다. 또한 <표 1>에서 map2의 성능평가 표를 통하여 Type1이 공간을 가상벽을 통해 분할하였을 때 잘못된 분할은 분할하지 않았을 때보다 더 성능이 떨어짐을 확인하였다. 이것은 사용자가 지도 공간을 비효율적으로 나누었을 때 그렇지 않은 경우보다 더 효율이 떨어진다는 것을 알 수 있었다. 결론적으로 랜덤방식은 map이 복잡 할수록 많은 중복작업으로 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 맵핑방식의 청소 로봇과 유사한 성능을 보일 수 있다.