

# 동적환경에서 무선 AP를 이용한 모바일 로봇의 목표 탐색 알고리즘

조정우\*, 배기민\*\*, 원일용\*\*\*

\*서울호서전문학교 사이버해킹보안과

e-mail:whwjddn369@naver.com

## Target Object Search Algorithm for Mobile Robot Using Wireless AP in Dynamic Environment

Jung-woo Jo\*, Gi-min Bae\*\*,Il-Young Weon\*\*\*

\*Dept. of Cyber Hacking Security, Seoul Hoseo

### 요 약

로봇 주행 기술은 전통적인 로봇요소 기술 외에도 여러 기술로 대상 응용서비스에 따라 IT 기술과 적극적인 융합을 통해 다양한 주행방법과 주행성능이 향상되고 있다. 본 논문에서는 대표적인 실내 모바일 로봇인 로봇 청소기를 대상으로 기존의 방법인 적외선과 카메라 방법이 아닌 보통 가정에도 쉽게 존재하는 AP를 이용해 목표를 설정하여 포섭구조 이론을 기반으로 동적인 환경에서도 충전 스테이션 까지 자율 주행이 가능한 로봇 알고리즘을 설계하였다. 그 결과 동적인 환경을 설정하여 로봇이 AP를 찾아가는 것을 확인하였고 주행 경로와 경과 시간을 표로 도출하여 다른 경우를 예측할 수 있게 하였다. 향후 행동 기반 로봇과 다양한 센서를 이용하여 로봇의 위치와 목표점 사이의 최단거리 경로를 구하여 주행하는 것이 목표이다.

### 1. 서론

과거의 로봇 주행은 환경 정보와 위치 정보를 기반으로 현재 위치로부터 목적지까지 경로를 생성하고 제어하는 기술 체계를 의미한다. 주행 기술은 이미 로봇청소기, 군용로봇, 무인주행 자동차, 농업용 무인 트랙터 등 개인용 서비스 로봇으로부터 전문서비스 로봇까지 다양한 응용제품의 형태로 구현되고 있다. 현재는 전통적인 로봇요소 기술 외에도 RTLS, WiFi, GIS, 센서 네트워크 등 대상 응용서비스에 따라 IT 기술과 적극적인 융합을 통해 향상된 주행 성능과 다양한 주행방법이 연구되고 있다.[1]

실내 모바일 로봇의 대표적인 예로 로봇 청소기를 들 수 있다. 과거에는 단순히 청소의 기능만 있었다면 현재는 자동충전 방식까지 활발하게 연구되고 있다. 보통 로봇 청소기의 자동충전 방식은 충전기에서 적외선 빛을 쏘고 청소로봇을 적외선 빛의 세기에 따라 충전 스테이션으로 이동하는 IR-LED light 유도방식과 전방 카메라를 이용한 Visual servoing방식이 적용된다. 그러나 IR-LED light 유도 방식은 충전 스테이션과 거리가 멀거나 장애물에 의해 적외선 빛을 인식하지 못하는 문제점이 있다. Visual servoing방식은 영상처리를 통해 IR-LED light유도방식 보다는 정확한 도킹 성능을 보여주지만 하드웨어의 가격이 비싸다는 단점이 있다.[2]

본 연구는 기존에 적외선 유도 방식을 개선하기 위해 가정에 많이 보급되어 있는 AP를 이용한 목표 탐색을 추가하여 거리가 멀거나 장애물이 존재해도 목표점에 도달

할 수 있다.

본 논문의 구성은 2장 관련 연구에서 행동 기반 내용을 소개하고 3장에서는 실험에서 사용되는 알고리즘을 설명한다. 그리고 4장과 5장에서는 실험결과와 결론으로 본 논문의 끝을 맺는다.

### 2. 관련 연구

#### 2.1 포섭 구조

포섭 구조란 각각의 기능을 모듈화 하여 수직적 계층을 가지는 구조이다. 이러한 구조의 계층은 자신의 계층의 역할만 수행하며 기능의 중요도에 따라 상위 계층과 하위 계층으로 구분된다. 상위 계층은 하위 계층의 제어가 가능하며 이러한 구조는 기존의 연구와는 다르게 동적인 환경에서 즉각적으로 반응할 수 있다.[3]

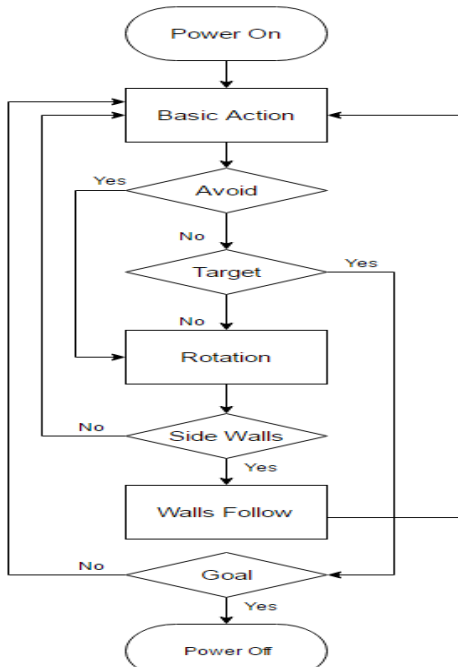
#### 2.2 행동 기반 시스템

행동 기반 시스템은 각각의 행동 집합들이 서로 상호작용을 통해 목표를 얻는다. 시스템 제작 시 의도한 결과를 얻을 수 있지만 상호작용을 통한 새로운 목표를 얻을 수 있는 장점이 있다. 또한 각 행동들은 저장 공간을 가지고 있어 반응형 시스템과 달리 기존의 저장 값과 현재 받은 값을 비교해 학습시킬 수 있다[4]

### 3. 제안 알고리즘

실내에서 로봇이 충전 스테이션에 가기 위해 AP 탐색 알고리즘과 기존 연구와는 다르게 동적환경에서 반응하기 위해 3계층으로 구성된 포섭 구조를 사용하였다.

(그림 1)은 제안 알고리즘을 순서도로 표현하였다. 로봇이 전원이 들어오면 로봇이 Goal 조건에 성립할 때까지 반복된 동작을 수행하게 되고 성립하면 반복을 끝내게 된다. Basic Action은 로봇의 기존 동작으로 기본 주행을 한다. 주행을 하는 사이에 장애물이 탐지되면 로봇은 회전하여 장애물을 회피하지만 로봇의 측면에 벽이 존재한다면 Wall Follow 동작을 하게 된다. 장애물이 탐지되지 않으면 Target을 하는데 AP의 값이 커지면 목표에 점점 가까워져 Goal 조건으로 도달했는지 확인하고 아니면 기존의 동작을 유지하고 그렇지 않다면 AP값이 점점 작아져 Rotation으로 로봇 기존의 방향을 바꾸어 동작한다.



(그림 1) 제안 알고리즘 순서도

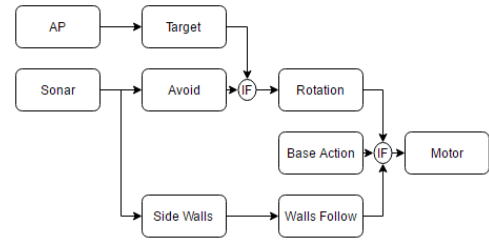
#### 3.1 AP 탐색

AP 탐색 알고리즘은 행동기반 로봇이 목표점이 될 즉 충전 스테이션인 AP까지의 거리를 탐색하여 목표점까지 주행하게 만드는 알고리즘이다.

AP의 신호 세기를 행동기반 로봇에서 측정하여 AP신호 세기가 약해지면 방향을 바꾸어 신호가 강해지는 방향으로 행동기반 로봇이 움직일 수 있게 한다.

#### 3.2 장애물 대응

기존의 연구에서 문제점이 되었던 동적인 환경에서 즉각적인 반응을 하기 위해 본 연구에서는 청소 로봇에 행동 기반 알고리즘을 적용하였다.



(그림 2) 포섭 구조

(그림 2)는 수직적 계층을 가진 포섭구조 기반의 행동 기반 알고리즘으로 표현한 구조이다.

가장 하위 계층인 Base Action은 단순히 로봇에게 기본적인 동작을 지시한다. 그리고 Base Action 보다 상위 계층인 Avoid는 로봇이 기본 동작을 하는 도중 Sonar를 통해 센서 값을 얻어와 로봇의 앞에 장애물이 있다면 센서로 측정을 하여 왼쪽 혹은 오른쪽으로 Rotation을 해서 장애물을 피하게 동작을 지시한다.

Target은 가장 상위 계층으로 아래 모든 하위 계층 보다 우선순위에 있다. 따라서 로봇은 AP를 통해 얻은 측정값을 계속해서 Target과 비교 연산해서 값이 작아지면 로봇의 기본 동작을 통해 가는 방향을 바꾸어 측정값이 커지는 방향으로 바꾸게 지시하여 AP 즉 목표점에 도달하게 한다.

```

If 1 == state and atarget < btarget
  print "target"
  turn(dir)
Else
  go()
    
```

(그림 3) 목표 탐색 계층

(그림 3)은 목표 탐색 계층의 의사코드로 현재 상태 값인 state 값이 목표 탐색 모드일 경우와 동시에 atarget 값인 기존의 AP 세기와 btarget 값인 이후의 AP 세기를 비교하여 기존의 값보다 이후의 AP 세기 값이 커진다면 목표에 멀어지는 것으로 보아서 로봇의 방향을 바꾸고 조건에 만족하지 않는다면 로봇은 그전의 상태인 앞으로 움직이는 행동을 하게 된다.

회피는 목표 탐색의 바로 하위 계층으로 로봇이 동적인 환경에서 즉각적으로 반응하기 위해 앞에 장애물을 탐지하고 장애물의 유무에 따라 로봇의 방향을 정한다.

```

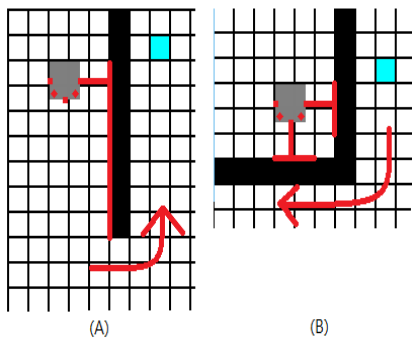
If True == collision(distance)
  print "avoid"
  back()
  turn(dir)
Else
  state *= -1
  go()
    
```

(그림 4) 회피 계층

(그림 4)는 회피 계층의 의사코드로 로봇이 벽에 닿기 전에 초음파 센서로 거리를 측정하여 일정 거리 내에 벽이 있다면 collision함수로 벽을 인식해서 로봇이 뒤로 간 다음 회전을 하게 된다. 만약 로봇이 벽에 충돌하지 않는다면 상태 값에 -1을 곱해서 목표 탐색 모드로 변경할 수 있게 한다. 그 다음에 로봇을 앞으로 움직이게 한다.

### 3.3 벽 대응

본 논문에서는 행동 기반 로봇이 목표를 탐색하여 자율 주행하는 과정에서 예외적인 상황이 발생하게 된다. 예외의 상황이 발생할 경우는 보통 크게 두 가지로 나뉘는데 그림 (A)의 상황은 목표점과 로봇 사이에 벽이 길게 있으면 로봇이 빠져나가지 못하는 경우이고 그림 (B)의 상황은 마찬가지로 로봇과 목표 사이에 벽이 있는데 그 벽의 모양이 'J'와 같을 때 벽을 따라 돌아서 가야 하는데 그러지 못한 경우가 존재한다. 따라서 목표점을 향해 주행하려면 벽을 돌아서 목표점에 가야하기 때문에 기존의 방식에서는 예외가 발생하게 된다.



(그림 5) 벽 대응 상태

(그림 5)에서 말하는 예외 상황들을 해결하기 위해 본 논문에서는 벽 대응 알고리즘을 소개한다.

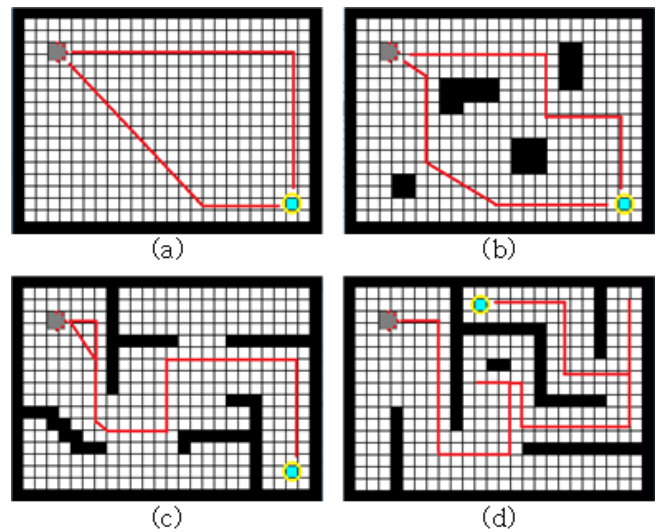
```
void function wallsfollow
print "walls follow"
While True
If distance-10 > sensor(distance)
go()
If True == collision(distance)
turn(dir)
Else
go()
turn(dir)
check = False
break
```

(그림 6) 벽 대응 의사코드

벽 대응 상태가 된다면 (그림 6)에서의 의사코드처럼 반복문이 실행되는데 측면 센서로 거리를 측정하여 벽을 인식하면 벽을 그대로 따라 계속 직진 한 뒤 측면에 벽이 없다면 앞으로 한 번 더 진행한 후 방향을 돌려 빠져나가게 된다. 이 동작으로 (A)의 상황을 빠져나가 목표점에 도달한다. (B)의 상황과 같은 경우는 측면의 벽을 인식하고 직진하는 중간에 앞에 장애물이 존재한다면 처음에 돌린 방향대로 한 번 더 돌려서 그대로 벽을 타고 나머지는 (A)의 경우와 같이 벽을 빠져나오게 된다. 벽 대응 상태는 스테드를 사용하기 때문에 check 변수로 상태를 주어서 False일 경우는 스테드가 동작하고 있을 경우이고 True일 경우는 동작 안하고 있을 경우로 동시에 실행이 되어 문제가 생기지 않게 설정을 한다.

### 4. 실험 및 결과

본 논문에서는 초음파 센서와 AP장비를 갖고 목표점인 충전 스테이션에 도달하는 것을 목적으로 진행하고 있다. 따라서 알고리즘을 테스트하기 위해 현실과 비슷한 환경을 만들어 주기 위해 python 언어를 사용하여 알고리즘을 테스트하기 위한 가상공간을 제작하였다. pygame을 이용하여 맵을 그렸다.



(그림 7) 시뮬레이터 결과

(그림 7)은 본 논문의 제안 알고리즘을 증명하여 로봇의 주행을 테스트하기 위한 시뮬레이터를 직접 제작한 이미지이다. 검정색으로 된 상자는 장애물을 표시하고 로봇이 갈 수 없게 하였다. 왼쪽 위의 상자는 로봇을 표시한 것이며 5개의 점들은 각각의 초음파센서를 표시한 것이고 오른쪽 표시는 로봇이 도착해야할 목표점을 표시한다. 그리고 로봇이 주행한 경로를 좀 더 자세하게 확인하고 연구하기 위해 로봇이 움직인 경로를 선으로 표시하였다.

(그림 7)의 (a)는 아무런 장애물 없이 로봇과 목표점만 두고 실험을 한 경우이다. 아무런 장애물이 없기 때문에 바로 목표점을 향해 주행을 하거나 직진을 하다 벽을 인

식 한 뒤 바로 목표점을 향해 주행하는 것을 볼 수 있다.

(b)는 로봇이 목표점에 도달하는 과정 사이사이에 장애물들이 있는 경우로 갑작스러운 장애물의 존재를 인식하고 피해가 동적인 환경에서 목적지까지 회피 상태와 목표를 추적하는 상태를 번갈아가며 장애물을 회피해 도달하는 것을 볼 수 있다.

(c)는 벽이 많아져서 좀 더 복잡한 공간이 형성된 상태이다. <표 1>의 결과를 보면 앞선 (a),(b) 실험보다 오래 걸리는 것을 알 수 있다. 그 이유는 로봇이 목표점에 가는 도중 샛길로 빠져 좋지 않은 상태가 반복되면 제안 시간 내에 목표점에 도달하지 못하기 때문이다.

마지막으로 (d)의 상황이 있는데 이 상황은 벽 대응 상태가 적용되어야만 해결할 수 있는 상황이다. <표 1>을 보면 (d)의 상황은 다른 상황 보다 시간이 더 오래 걸리는 것을 볼 수 있다. 다른 상황보다 오래 걸리는 것은 벽 대응 상태를 통해 벽을 따라 주행을 하다보면 원하는 방향으로 한 번에 가지 못하는 상황이 발생하기 때문에 로봇이 길을 돌아가는 경우가 많아져 평소보다 훨씬 오래 걸리거나 제시간에 도달하지 못한다.

<표 1> 상황별 주행시간

	(a)	(b)	(c)	(d)
1회	7초	12초	30초	50초
2회	7초	14초	35초	56초
3회	8초	13초	28초	70초
4회	8초	12초	50초	45초
5회	7초	15초	27초	40초
평균	7.4초	13.2초	34초	52.2초

<표 1>은 시뮬레이터 기반으로 5회를 기준으로 각각의 상황을 테스트해 걸린 경과 시간과 평균 시간을 계산해서 기록하였다. 따라서 결과표를 보면 각각의 상황별 시간이 얼마나 걸리는지를 대략적으로 한눈에 볼 수 있어 다른 경우도 예측을 할 수 있다.

<표 1>의 (a)와 (b)의 경우와 같은 상황일 때는 크게 장애물과 벽들이 복잡하게 되어 있지 않기 때문에 목표점에 도달하는데 큰 문제가 없다. 때문에 각각의 걸린 시간과 평균 시간을 보면 알 수 있듯이 크게 차이는 나지 않는다.

(a),(b)와 다르게 (c),(d)와 같이 장애물이 많이 존재할 때는 평소보다 로봇이 목표점에 도달하는 것이 오래 걸린다. 그리고 많은 장애물들 때문에 원하는 방향으로 로봇이 주행하지 않고 목표점과 멀어져 가장 빨리 도착할 때와 가장 늦게 도착할 때의 차이가 매우 큰 것을 볼 수 있다.

## 5. 결론 및 향후 과제

기존의 로봇은 동적인 환경에서 반응하지 못하고 정해진 규칙에 따라 움직이는데, 본 연구는 기존의 로봇과 다르게 동적인 환경에서도 반응할 수 있도록 포섭구조와 행동기반 시스템을 사용하였다. 시뮬레이터를 통한 실험으로 동적인 환경에서 반응하여 환경에 따라 주행방법이 바뀌면서 목표점까지 정상적으로 찾아가는 것을 확인하였다.

향후 행동 기반 자동 충전 로봇의 위치와 목표점의 위치를 기준으로 최단거리 경로를 찾아 주행해야하며, 초음파 센서에만 의존하여 장애물을 탐지 및 회피하는 것이 아니라 다양한 센서를 이용하여 빠르고 정확한 주행을 가능하도록 연구가 지속적으로 이루어져야 한다.

## 참고문헌

- [1] W.P. Yu, S.L. Choi, J.Y. Lee, S.H. Park "Robot Navigation Technology and Its Standardization Trends" ETRI Electronics and Telecommunications Trends Vol.26 No.6 p.3 Dec 2011
- [2] Korean Intellectual Property Office "List of source · standard patent of robot technology Collection(1)" p.20 August 2013
- [3] Rodney A Brooks "A Robust Layered Control System For A Mobile Robot" MIT Press. pp.1-8 (1985)
- [4] Rodney A Brooks "A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network" February, 1989