

방향 보정을 통한 행동기반 로봇의 목표 탐색

민병준 성중곤 원일용
호서전문학교 사이버해킹보안과
e-mail : okminkr@gmail.com

Target Object Search Algorithm for Behavior-based Robot Using Direction Refinement

Byeong-Jun Min, Joong-Gon Sung, Il-Young Won
Dept. of Cyber Hacking Security, Ho-seo Vocational-School

요 약

제한된 환경에서 로봇이 동적 장애물들에 대해 능동적으로 대처하며 목표한 지점까지 도달하기 위한 알고리즘을 제안한다. 로봇은 행동기반 시스템으로 만들어져 주변 장애물들을 자율적으로 회피한다. ex-agent 는 공중에서 주변 환경들을 modeling 한 뒤 cell-map 을 만들어 A*알고리즘을 통해 이동 경로를 설정한다. 이동 경로와 로봇의 진행방향을 비교하여 회전 방향을 조언해준다. 로봇은 ex-agent 로부터 받은 조언과 센서값들을 조율하여 장애물들을 능동적으로 회피하며 목표 위치를 찾아갈 수 있다. 실험은 시뮬레이터를 통해 이루어졌으며 장애물들에 대해 원만한 회피율을 보였다.

1. 서론

전통적인 로봇의 목표탐색 알고리즘[1][2]은 비전처리를 중심으로 주변 환경을 modeling 하여 map 을 만든 뒤 경로계획 및 이동 거리를 계산하여 로봇을 움직인다. 이러한 방식은 모델링 사이의 동적 장애물들을 회피하지 못한다. 이로 인해 지속해서 경로계획을 수정해주어야 하며 또 그 수가 많을수록 복잡도와 헛수 또한 증가한다.

이러한 문제를 해결하는 방법으로 행동기반제어가 있다. 센서들의 융합을 통해 주변 장애물들에 대해 즉각 반응하며 로봇이 목표를 완수하는 데 필요한 행동들을 구분하고 이를 결합하는 포섭구조를 사용한다. 이러한 방식으로 만들어진 로봇으로는 Rodney-boorks 교수의 Genghis[3]가 있다.

본 논문에서는 전통적인 로봇의 목표탐색 알고리즘과 행동기반 제어를 융합하여 효율적이면서도 간단하게 목표를 찾아가는 알고리즘을 제안한다. 로봇은 자율적으로 주변 장애물들을 회피하며 이동하게 되고 ex-agent 는 공중에서 비전처리를 통해 조언자 역할을 한다. 로봇은 자신이 처한 상황과 ex-agent 로부터 전달받은 조언을 조율하여 목표를 향해 나아간다. 두 에이전트 간의 커뮤니케이션을 통해 정확한 이동거리 계산 없이 빠르게 목표를 탐색할 수 있다.

본 논문의 구성은 2 장에서 관련된 연구들을 보게 되며 3 장에서는 로봇/ex agent 의 목표탐색 알고리즘을 제안하고 4 장에서는 시뮬레이터를 통해 알고리즘의 성능을 측정하며 5 장에서 결론 및 향후 과제를 다룬다.

2. 관련연구

2.1 포섭구조

포섭구조[4]란 로봇의 행동들을 수직적으로 분할한 뒤 개별행동에 따라 모듈화를 시키는 것을 말한다. 각 모듈은 자신의 기능을 완수하는 것에만 관심이 있다. 모듈의 출력이 다른 모듈의 입력으로 들어갈 수 있다. 모듈들은 중요도에 따라서 수직적으로 배치된다. 상위 모듈은 하위 모듈의 입출력을 억제 가능하며 하위에 있는 모듈들을 수렴하게 된다.

2.2 행동기반 시스템

행동기반 시스템[5][6]은 센서와 동작의 단단한 결합을 유지하며 행동으로 정의하게 된다. 전통적인 세계 모델에 의존하여서 광범위하게 계산하거나 추론하여 행동하지 않는다. 행동기반 시스템에서 계산은 동적으로 응답하고 빠르게 변화하는 환경에서의 작업에 적당한 방식이다.

2.3 A* 알고리즘

A*알고리즘[7]은 출발점에서 목표 지점까지 최단 경로를 찾는 휴리스틱한 그래프/트리 알고리즘이다. 개별 노드들에 대해 평가함수를 정의하고 최단 경로를 찾는다. 출발 노드로부터 n 노드 까지의 경로 가중치와 n 노드에서 목표 노드까지 거리 추정 가중치의 합으로 개별 노드들을 평가한다. 가치 판단의 근거에 따라 거리 추정 가중치를 매기는 방법이 달라질 수 있다.

3. 제안

3.1 Ex-Agent

ex-gent 는 전지적 시점의 영상 데이터를 받아서 cell-Map 을 만든다. cell 의 크기는 실 세계와 k 비율로 대응한다. 모델링 된 map 으로 A*알고리즘을 통해 목표 경로를 계획한다.

그림 3.1.1 는 경로가 계획된 뒤 로봇에게 advice 를 하는 algorithm 의 의사 코드다. point 는 목표에 다가가기 위한 중간 목표 지점이다. point 의 위치는 휴리스틱하게 선정한다. point 에 도달할 때마다 새롭게 갱신하며 현재 point 위로 동적 장애물이 나타날 경우도 새롭게 갱신한다. 로봇이 목표에 도달할 경우 로봇은 제자리에서 멈추게 한다.

```

procedure run()
  while True do
    set pos to rPos

    if isGoal then
      set rState to stop

    if checkPoint then
      set point to nextPoint

    set point is checkOnWall

    set rState to advice
  end while
end procedure
    
```

(그림 3.1.1) Ex-Agent 알고리즘

3.2 Robot-Agent

로봇 에이전트의 기본 동작			
Forward	Turn-left	Turn-right	Backward

robot-agent 는 주변 장애물들과의 거리를 측정하기 위해 sonar 모듈을 장착하고 있다. 초음파 센서는 잡음에 민감하며 정확도가 떨어지기 때문에 n 개 이상의 초음파 센서를 융합해서 사용한다. 다수의 초음파 센서들의 검출 값을 통해서 회전방향을 결정하고 로봇의 이동 방향을 회전시킨다.

목표에 도달하기 위한 로봇의 행동을 adjust, avoid, straight 3 가지 행동으로 구분한다. 각 행동은 하드웨어와 함께 모듈화되어서 로봇에 장착되며 우선순위에 따라 계층화 되어있다. 모듈들의 출력은 arbitrator 로 들어가 로봇의 움직임을 변화시킨다.

그림 3.2.1 은 로봇의 내부 algorithm 의 의사 코드다. 로봇은 장착된 모듈들의 act 를 모두 fire on 시킨 뒤 출력을 arbitrator 에 전달한다.

```

procedure run()
  while True do
    set d to distances
    fire behaviors

    set action to arbitrate

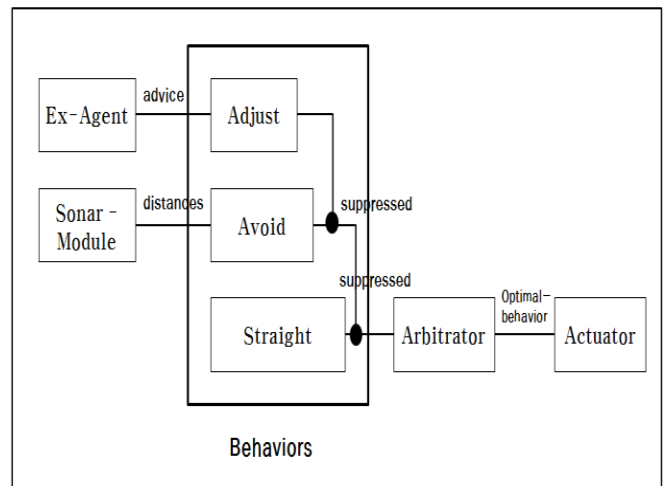
    run actuate
  end while
end procedure
    
```

(그림 3.2.1) 로봇 알고리즘

3.3 행동기반 로봇의 목표점 탐색

straight, avoid, adjust 모듈들은 계층화되어 있으며 가장 낮은 계층을 0-layer 라 한다. 0-layer 는 항상 fire on 상태로 avoid 와 adjust 모듈의 다른 출력이 없을 때 로봇을 항상 앞으로 나아가게 한다. 1-layer 는 주변 장애물들과의 회피를 담당한다. 전면부의 충돌방지 거리는 f 값으로 측면부의 충돌방지 거리는 s 값으로 설정한다. sonar 모듈로부터 거리 값을 전달받아 적절한 회피 방향을 선정하고 로봇의 진행 방향을 변경시킨다. 2-layer 는 비동기적으로 ex-agent 의 조언을 기다린다. 조언이 들어올 경우 하위 layer 들의 출력을 suppressed 한 뒤 출력을 arbitrator 로 전달한다. arbitrator 에서는 현재 로봇의 행동과 들어온 입력을 조율하여서 optimal-behavior 를 선출하여 actuator 에 전달한다.

그림 3.3.1 은 로봇의 포섭구조와 구성도를 보여준다. ex-agent 로부터 로봇의 2-layer 에 advice 가 전달되는 모습과 Sonar 모듈로부터 1-layer 에 distances 가 전달되는 모습을 나타낸다. 상위 모듈들의 출력이 하위 모듈의 출력을 suppressed 하는 모습을 나타낸다. arbitrator 를 통해 선정된 optimal-behavior 가 actuator 에 들어가게 되어 로봇의 행동을 변화시킨다.



(그림 3.3.1) 로봇의 포섭구조

4. 실험

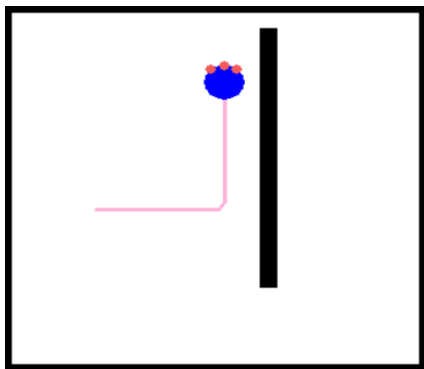
본 논문에서는 실험환경에 제한을 두었다. 공중에서 실험 환경을 촬영 가능 하여 전지적 시점의 데이터를 ex-agent 가 처리 가능해야 한다. 로봇은 행동기반 제어 방식으로 구성되어 자율적으로 주변 장애물들을 회피 가능하여야 한다. 두 agent 는 서로 통신이 가능하여야 한다.

실험은 python 으로 직접 개발한 시뮬레이터를 통해 실험하였으며 k 값은 30cm 로 1cell 당 30cm 로 실험하였다. 그림 4.1 은 실험을 위해 만들어진 가상 실험공간을 모델링 한 Cell-map 이다.



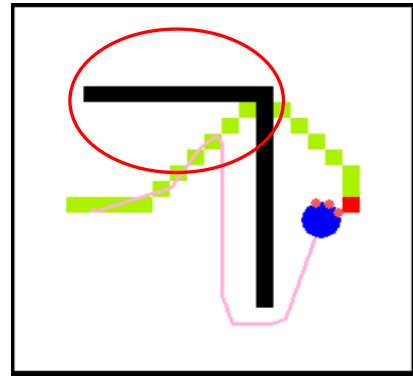
(그림 4.1) Cell-map

그림 4.2 는 1-layer 까지 장착한 로봇의 회피를 보여주는 사진이다. 로봇의 전면부에 초음파 모듈이 장착되어 있으며 40 도 간격으로 총 3 개의 초음파 센서를 부착하고 있다. 초음파 모듈의 전면부에 25cm 측면부는 15cm 의 충돌방지 거리를 두었다.



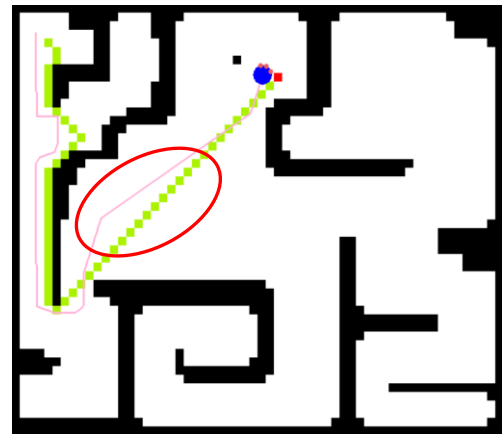
(그림 4.2) 로봇의 회피

그림 4.3 은 2-layer 까지 장착한 로봇이 장애물을 회피하면서 목표에 도달하는 모습을 보여준다. 실선은 로봇이 실제 이동한 경로이며 격자 모양의 선은 ex-agent 가 계획한 경로이다. 최초 로봇이 계획한 경로 위로 추가적인 장애물이 나타났을 때 유연하게 경로를 바꾸어 목표에 도달하였다.



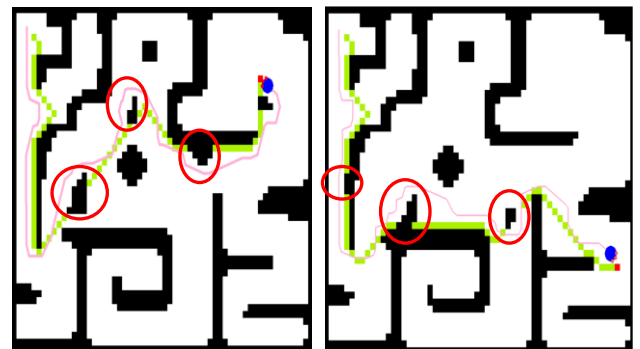
(그림 4.3) 2-layer 장착한 로봇의 움직임

그림 4.4 2-layer 까지 장착한 로봇이 방향 보정을 통해 이동한다는 것을 보여준다. 계획한 경로로 고정적으로 따라가는 것이 아니라 방향 보정을 통해서 로봇은 나아간다.



(그림 4.4) 방향보정을 통한 이동

그림 4.5 는 좀 더 먼 거리에서 ex-agent 가 계획한 경로 위로 추가적인 다수의 장애물이 나타난 경우 로봇의 회피를 보여주는 그림이다. 계획 이후 나타난 동적 장애물들을 유연하게 회피하면서 목적지에 도달하였다.



(그림 4.5) 동적 장애물 회피

5. 결론

본 논문에서는 제한된 환경에서 두 에이전트의 커뮤니케이션을 통해 동적 장애물들에 대해 능동적으로 대처하며 목표한 지점까지 도달하기 위한 알고리즘을 제안했다. 자율적으로 회피하는 로봇과 ex-agent의 커뮤니케이션만으로 지속해서 전체 경로계획을 수정할 필요가 없이 목표에 도달할 수 있다. 또한, 정확히 계산된 이동 거리로 움직이지 않기 때문에 초기 계획경로와 간단한 방향 보정만을 통해서 빠르게 목적지에 도달 가능하다.

실험 환경에서 작은 크기의 동적 장애물들에 대해서는 능동적으로 대처하며 목표지점까지 도달하였다. 하지만 경로 중간이 크게 막힌 경우는 높은 실험률을 보이지 않았다. 따라서 좌수법 등을 도입해 알고리즘을 개선한다면 더 높은 실험률을 기대할 수 있다.

참고문헌

- [1] - Hans P. Moravec "The Stanford Cart and the CMU Rover"
- [2] - Nils J. Nilsson "Shakey the Robot" SRI AI Center Technical Note 323, April 1984
- [3] - Rodney. A. Brooks "A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network"
- [4] - Rodney. A. Brooks "A Robust Layered Control System for a Mobile Robot"
- [5] - R.C Arkin "Behavior-Based Robotics"
- [6] - Jonathan H. Connell "A Behavior-based Arm Controller"
- [7] - Steve Rabin. "AI Game Programming Wisdom"