군집로봇 간의 충돌 회피를 위한 시뮬레이션

조창권, 우 균
부산대학교 컴퓨터공학과
e-mail:ckcho@pusan.ac.kr

Swarm Robot Simulation for Collision Avoidance

Chang-Kwon Cho, Gyun Woo
Dept of Computer Science & Engineering, Pusan National University

요 약

군집로봇은 어떤 작업을 수행하기 위해 여러 로봇이 함께 움직이게 된다. 이렇게 여러 로봇이 함께 작업을 수행하다 보면 로봇 간의 충돌을 피해야 할 경우가 발생한다. 각 로봇은 주어진 환경을 인식하고 모델링하며 로봇들 간에 충돌을 피하여 주어진 임무를 효과적으로 이행할 수 있게 하는 것이 이 연구의 핵심이다. 로봇 간 충돌을 효과적으로 해결하기 위해 충돌 전 다양한 위치와 다양한 방법으로 충돌을 피할 수 있도록 했다. 따라서 이 논문에서는 카메라 센서와 위치정보 통신을 이용한 방법을 통해 군집로봇간의 충돌 회피에 대해 연구하였다.

1. 서론

최근 로봇 분야의 발전으로 로봇을 그룹화 하는 모델이 많은 주목을 받고 있다. 대표적으로 사회적 협동체제 기반으로 하는 집단 지능의 최적화 모델인 Boid 모델, Floys 모델이 있다[8]. 이러한 모델들은 몇 가지 특징을 가지고 있다. 첫째로 각각의 로봇은 주위환경을 잘 인식하며 독립적으로 행동한다. 둘째 주어진 작업을 수행하기 위해 다른 로봇과 협동적으로 행동한다. 이런 행동의 기반으로 자연계 생물을 응용한 한 대규모 집단 모델의 연구가 진행 되고 있다.

Boid 모델과 Floys 모델은 개방된 공간을 가정한 대규모 집단 모델로, 장애물을 만날 경우에 문제점이 발생한다. 장애물을 만나게 되면, 두 그룹으로 나뉘어 졌다가 장애물을 회피 한 후 반대쪽에서 다시 그룹을 형성한다. 이렇게 많은 수의 로봇이 동시에 동일한 지역으로 이동했을 때 로봇간 충돌이 발생한다. 이때 충돌을 회피 하기위해 많은 자원과 시간적인 낭비가 발생한다. 이런 문제는 특히 장애물이 많이 곳에서 발생한다. 둘째 환경의 변화가 크게 나타났을 때 로봇 스스로 작업을 수행하기 위한 행동을 결정하는 것이 어렵다.

이 논문에서는 특정 환경에서 주어진 일을 효과적으로 처리하기위해 충돌을 방지하는 방법을 2차원 공간에서 제시하였다. 로봇 간 충돌 회피를 제어하기위해 두 가지 방법을 수립하고 실험하였다. 첫 번째는 카메라의 센서를 통해 다른 군집로봇을 인식하고 로봇 간의 거리를 유지하여 충돌을 회피하는 방법과 두 번째는 실시간으로 각 군집로봇의 위치 정보를 통해 거리를 확보하여 충돌을 회피하는 방법이다[2,4].

2. 군집로봇의 제어시스템 환경조건

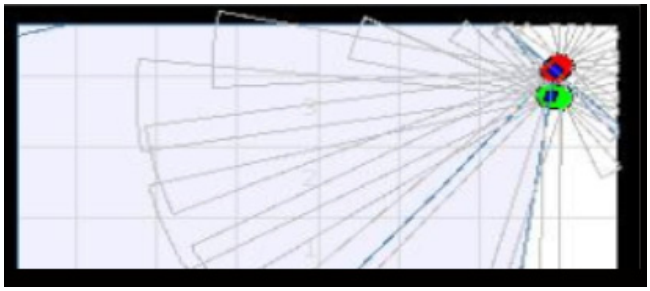
2차원 맵에서 군집 로봇을 제어하는 알고리즘을 실험했다[1,3,6,7]. 2차원 맵에서 군집로봇 제어시스템의 환경조건과 제약 사항은 다음과 같다.

- 로봇은 모두 동일한 모델이다.
- 작업 공간이 평평하고 벽과 장애물 이외 아무런 방해요소를 포함되지 않는다.
- 참조 로봇은 궤도를 따라 같은 속도를 유지한다.
- 각 지정 로봇 특유의 번호에 의해 움직인다.
- 각 로봇의 센서를 통해 필요한 정보를 추출한다.

로봇의 다양한 환경에서 다른 로봇들과 협동을 통해 주어진 일을 안정적으로 수행할 수 있도록 위와 같은 가정 하에 실험을 했다.

3. 군집 로봇 간의 충돌 회피

로봇의 작업 공간에서 원하는 목표작업을 수행하기 위해서 이동이 필요하다. 그러나 그림 1에 보이는 것처럼 로봇이 서로의 목표지점으로 이동할 때 로봇 간 마주치는 경우가 발생한다. 이런 경우 서로가 장애물로 판단하고 장애물을 피하기 위해 회전 하면서 충돌이 발생한다.

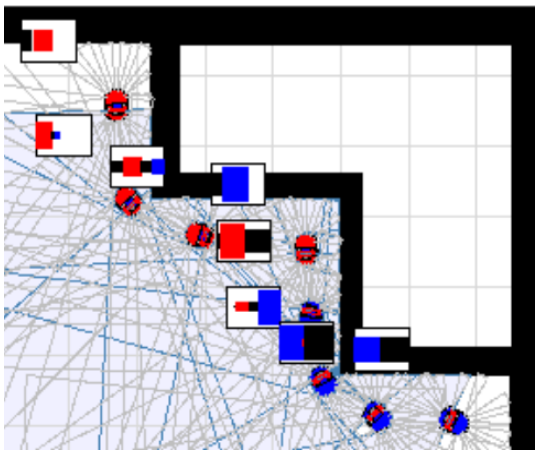


(그림 1) 교착 상태 발생

이 논문에서는 이 같은 문제점을 방지하고 안전적인 임무를 수행할 수 있도록 다음과 같은 방법을 제시한다.

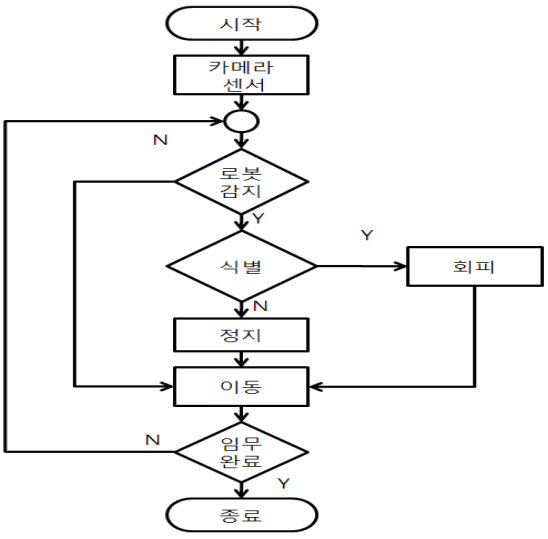
3.1 카메라 센서

이 장에서는 카메라 센서를 이용한 방법을 제안한다. 그림 2와 같이 카메라 센서를 이용하여 로봇의 주위환경을 인식하고 다른 군집로봇의 위치와 거리를 파악할 수 있다.



(그림 2) 로봇의 카메라 센서

카메라 센서를 사용하기 위해서는 먼저 로봇의 카메라 센서를 초기화 시키고 카메라가 주변사물의 색을 인식할 수 있도록 한다. 그림 2와 같이 카메라를 통해 인식한 색의 값(크기)을 저장하고 그 값을 통해 로봇 간 일정 거리를 유지한다. 카메라 센서를 이용한 알고리즘의 다이어그램은 다음 그림 3과 같다.

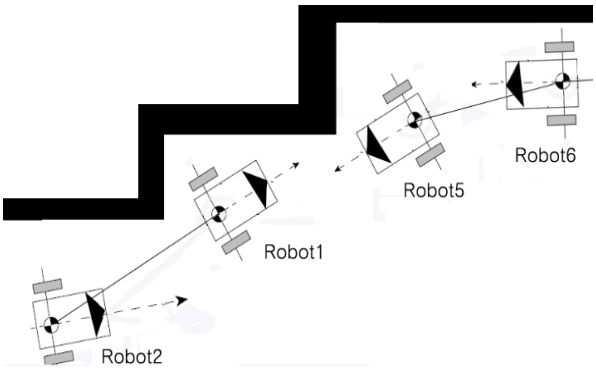


(그림 3) 카메라 센서 알고리즘 다이어그램

군집로봇의 모든 로봇은 카메라센서를 이용하며 다른 로봇들의 위치를 파악한다. 다른 군집로봇이 교착지점 안으로 들어왔을 때 카메라를 통해 감지하며 한쪽 군집로봇에게 우선순위를 부여 먼저 지나갈 수 있도록 한다. 다른 군집로봇은 우선순위를 부여 받은 군집 로봇이 지날 갈 때까지 정지한 후 출발한다.

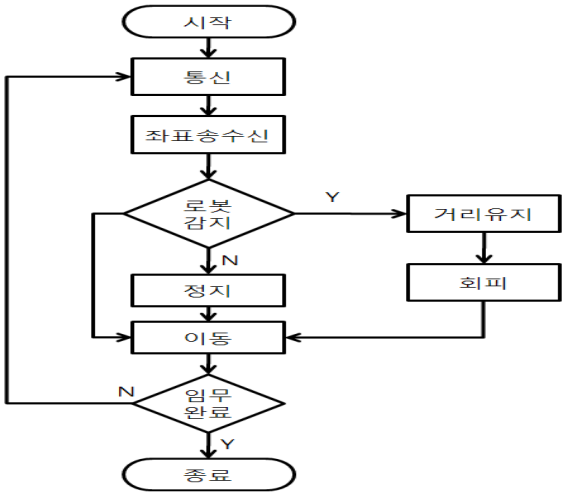
3.2 위치정보 통신

이 장에서는 그림 4와 같이 각 로봇의 P2P통신을 이용하는 방법을 제안한다. 실시간으로 위치정보를 공유하여 각 로봇의 거리를 유지하면서 혼잡 상황에서의 충돌을 회피한다.



(그림 4) 위치기반을 이용한 통신

통신 할당 크기는 4×256 크기로 지정 하고 각 로봇의 위치정보는 이차원(X , Y좌표)으로 저장한다. 저장된 좌표 값은 각 로봇이 공유한다. 각 좌표 값을 계산하여 로봇 간, 그룹 간 거리를 유지하고 충돌을 피할 수 있게 한다. 위치정보를 이용한 통신의 다이어그램은 그림 5와 같다.



(그림 5) 위치정보를 이용한 통신 다이어그램

군집로봇의 목표작업이 수행 되었을 때부터 통신을 통해 위치정보를 공유하며 로봇 간 일정 거리를 유지한다. 그리고 군집로봇 간 서로 통신을 통해 위험 감지한다. 교착 지점에 들어왔을 때 한쪽 군집로봇에게 우선순위를 부여하여 거리를 유지하면서 충돌을 피해 지나간다. 다른 군집로봇은 정지하고 기다렸다가 출발한다.

4. 구현

4.1 시뮬레이션 환경 구축

본 연구는 리눅스 환경에서 구동되는 Player/Stage를 위해 윈도우용 가상머신인 VM웨어를 바탕으로 실험을 진행하였다. 로봇관련 명령어를 처리하기 위해 Python 언어를 사용하였다[5]. 또한 로봇의 작업 환경 시뮬레이션을 위한 이동 경로와 장애물은 직접 이미지 파일로 만들어 주위 환경을 설정하였다.

4.2 시뮬레이션

본 연구의 시뮬레이션을 위해 2개의 그룹으로 나누고 각 그룹에는 4대의 로봇으로 구성했다. 각 로봇에는 Blobfinder, Sonar, Laser, Position 4가지 센서를 연결하였다. 그리고 이 시뮬레이션에 적용된 환경을 사각형태의 벽과 장애물로 이루어져 있다.

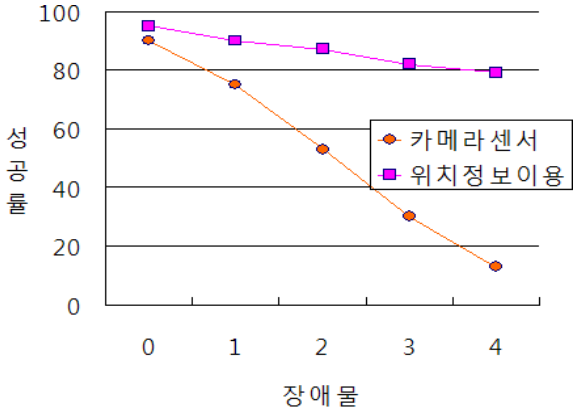


그림 6 시뮬레이션 성공률

그림 6에는 카메라 센서와 위치정보를 이용하여 군집로봇 간 충돌 회피 시뮬레이션의 성공률을 나타낸다. 군집로봇 간 접촉이 일어나는 지점에 장애물을 추가하여 충돌 회피를 테스트 하였다. 장애물은 0개부터 4개 까지 만들어 각 30번의 실험을 반복하였다.

카메라 센서를 이용한 방법에서는 카메라에 보이지 않는 사각지역에서 다른 로봇그룹이 나타났을 경우 충돌이 발생한다. 이런 사각지역이 많이 나타나는 굴곡이나 장애물이 늘어날수록 성공률도 크게 하락하였다. 하지만 위치정보를 이용한 방법에서는 로봇이 보지 못하는 사각지역까지도 다른 로봇그룹의 위치정보를 공유할 수 있기 때문에 굴곡이 증가하여도 성공률은 크게 낮아지지 않을 수 있었다.

5. 결론

이 논문에서는 군집 로봇 간 충돌회피를 위한 카메라 센서를 통한 방법과 위치정보 통한 방법을 제안하였다. 그리고 장애물이 있는 환경에서 각 방법을 시뮬레이션하고 비교, 평가 하였다.

두 방법을 비교하여 로봇의 교착상태를 감지하고 안정적으로 회피방법을 강구할 수 있었다. 특히 위치정보를 이용한 방법을 응용한다면 실시간으로 위험 상황전달을 통해 사전에 위험으로부터 대비할 수 있을 것이다. 하지만 모든 교착 상태를 해결할 수 있는 것은 아니다. 본 연구에서 제시한 방법은 일정 공간을 가져야 하기 때문에 작은 공간에 많은 수의 로봇이 함께 이동한다면 로봇이 길을 잃어버리는 일이 발생하였다. 이를 보완한 알고리즘과 여러 가지 변수들을 제시하여 연구를 진행하여야 할 것이다.

참고문헌

[1] 임준원, 이상훈, 서일홍, 김종복, 김경진, “플레이어 스테이지 로봇 시뮬레이터의 로보이드 컴포넌트화에 관한 고찰”, 한국지능로봇 종합학술대회, pp. 281-286, 2009.

[2] Rami Yared, Xavier D´efago, Julien Iguchi-Cartigny , Matthias Wiesmann, “Collision Prevention Platform for a Dynamic Group of Asynchronous Cooperative Mobile Robots”, Journal of Networks, Vol.2, No.4, PP28-39, 2007.

[3] 김영덕, 김진욱, 강원석, 안진웅, “멀티로봇 환경에서 트래픽량을 고려한 효율적인 이동로봇 경로계획 기법”, 대한전자공학회 정보 및 제어 심포지움, PP363-365, 2009.

[4] Asaf Shiloni, Noa Agmon and Gal A. Kaminka , “Of Robot Ants and Elephants”, Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, pp81-88, 2009.

[5] Douglas Blank, Deepak Kumar, Bryn Mawr College, “Pyro: A Python-based Versatile Programming Environment for Teaching Robotics”, ACM. Journal of Educational Resources in Computing, Vol.3, PP1-15, 2003.

[6] Leandro Soriano Marcolino and Luiz Chaimowicz, “Traffic Control for a Swarm of Robots: Avoiding Group Conflicts”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp1949-1954, 2009.

[7] Leandro Soriano Marcolino and Luiz Chaimowicz ,“Traffic Control for a Swarm of Robots: Avoiding Target Congestion”, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp1955-1961, 2009.

[8] Makoto Itoh, Leon O. Chua, “Boids Control of Chaos”, International Journal of Bifurcation and chaos, Vol. 17, No. 2, PP427-444, 2007.