

# PBUG: 모바일 로봇 쌍을 위한 버그 알고리즘

조창권\*, 우균\*

\*부산대학교 컴퓨터공학과

e-mail: ckcho@pusan.ac.kr,

woogyun@pusan.ac.kr

## PBUG: Bug Algorithms for a Pair of Mobile Robots

Chang-Kwon Cho\*, Gyun Woo\*

\*Dept of Computer Science and Engineering, Pusan National University

### 요 약

이 논문은 한 대의 모바일 로봇의 모션 계획 알고리즘인 Bug1과 Bug2를 개선한 알고리즘을 제안한다. 장애물이 있는 환경에서 목표지점까지 도달하기 위한 경로 계획 알고리즘으로 Bug1과 Bug2가 제안되었지만, 이 두 알고리즘은 모두 장애물 형태에 따라 탐사 시간이 오래 걸릴 수 있다는 단점이 있다. Bug2 알고리즘은 Bug1 알고리즘을 개선한 형태로 제안되었지만 심지어 극적인 경우에는 무한 루프에 빠진다는 단점이 있다. 이 논문에서는 이러한 단점을 해결하기 위해 한 쌍의 모바일 로봇을 이용한 병렬 탐색 PBug1, PBug2 알고리즘을 제안한다. 제안된 PBug1과 PBug2 알고리즘은 각각 Bug1과 Bug2의 속도를 보장하며 일반적으로 빠른 탐사시간을 보인다. 특히 PBug2 알고리즘은 Bug2와 달리 무한루프에 빠지는 경우가 없다. 제안된 알고리즘의 성능을 평가하기 위해 PBug1, PBug2 알고리즘을 구현하여 Bug1, Bug2 알고리즘과 비교하였다. 실험결과 PBug1 알고리즘은 Bug1 알고리즘보다 탐사 시간이 64.9%로 감소하였고 PBug2 알고리즘은 Bug1 알고리즘과 비슷한 탐사 시간을 보였다.

### 1. 서론

최근 로봇 분야의 발전으로 로봇을 그룹화하는 모델이 주목받고 있다. 대표적으로 사회적 협동체제 기반으로 하는 집단 지능의 최적화 모델인 Boid 모델과, Flocks 모델이 있다[1, 2]. 이런 행동의 기반으로 자연계 생물을 응용한 대규모 집단 모델의 연구가 진행되고 있다.

하지만 이동 로봇의 이동 중 예측할 수 없는 임의의 장애물이 있는 2D 환경에서 흩어진 로봇의 최초 위치에서 다른 로봇과 융합하여 군집을 형성하는 알고리즘에 대한 연구도 필요하다. 그래서 본 연구는 최초 위치에서 목표 위치를 찾아 이동하는 Bug 알고리즘을 이용하여 군집화 형성을 위한 기초단계를 연구해 보고자 한다.

기존의 버그 알고리즘인 Bug1과 Bug2는 목표 지점에 도달하는 것을 목적을 두고 있다[3]. 하지만 기존 Bug 알고리즘은 시작지점에서 목표지점까지 장애물 형태에 따라 탐사 시간이 오래 걸리거나 장애물을 모두 탐사할 수 없는 경우로 있다는 단점이 있다.

본 논문에서는 Bug1과 Bug2의 단점을 해결하기 위한 PBug1과 PBug2 알고리즘을 제안한다. 동적으로 변하는 상황에서 한 쌍의 모바일 로봇을 이용하여 병렬 탐색함으로써, 두 대의 로봇이 서로의 위치를 알려주는 신호를 방출하여 무사히 목적지에 대한 탐사를 효과적으로 수행할 수 있는 PBug1과 PBug2 알고리즘을 제시하고자 한다. 본 논문에서 연구한 알고리즘을 비교하기 위해 간단한 실험을 진행하였다. 먼저 Bug1과 Bug2 알고리즘을 실험하고 제

안한 한 쌍의 모바일 로봇을 이용한 병렬 탐색 PBug1, PBug2 알고리즘을 실험하였다.

### 2. Bug 알고리즘

버그 알고리즘이란 마치 벌레가 벽을 따라 움직이는 것 같다고 하여 명명된 버그 알고리즘이다[3]. 버그 알고리즘은 간단한 경로 계획에 따라 실험된다. 예측할 수 없는 장애물에 직면하면 장애물의 외곽을 따라 이동하며 최종 목표지점으로 향한다. 버그 알고리즘을 구현하기 위한 4가지 가정이 적용된다. 첫째, 맵 상의 시작지점과 목표지점의 좌표를 인지할 수 있어야 한다. 두 번째, 맵 상의 장애물의 범위를 미리 알 수 없다. 셋째, 센서를 사용하여 범위 안에서 장애물을 감지할 수 있다. 네 번째, 맵 상에서 로봇은 자기 위치를 알 수 있다. 즉, 자기 조직화를 이룰 수 있는 기본정보를 획득할 수 있다고 가정한다[4].

#### 2.1 Bug1 알고리즘

Bug1 알고리즘은 먼저 로봇의 시작위치와 목표 위치를 알 수 있다[5]. 로봇이 장애물을 감지 안 되는 경우 목표 지점으로 바로 움직인다. 감지할 경우 목표지점에서의 동작이 다시 가능하게 될 때까지 로봇은 장애물의 주변을 조사한다. 장애물을 만나면 시계 반대 방향으로 돈다(기본값). 그리고 장애물을 도는 동안 정 방향과 목표지점의 거리를 계산하여 떠나는 지점(leave point)을 결정한다. 로봇은 떠나는 지점으로 이동 후 그 다음 단계의 직선 경로를

따라 목표지점으로 이동한다. 최종 위치 지점까지 다시 장애물을 만나면 반복 작업을 계속 수행한다.

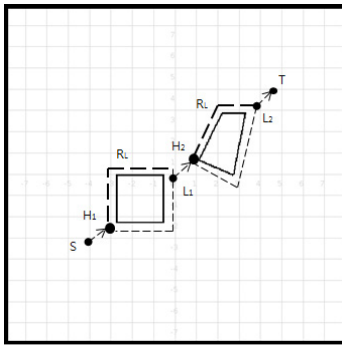


그림 1 Bug1 알고리즘

### 2.2 Bug2 알고리즘

Bug2 알고리즘도 Bug1 알고리즘과 같이 먼저 로봇의 시작위치와 목표위치를 알 수 있다[4]. 그리고 출발 지점과 목표 지점을 잇는 직선(slope)을 지속적으로 따라 이동한다. 로봇이 장애물에 의해 막히지 않는 한 목표지점으로 동작을 유지하면서 이동한다. 장애물을 만나게 되면 초기의 직선을 다시 찾을 때까지 센서를 이용하여 시계 반대 방향으로 장애물의 가장자리를 돌며 이동한다.

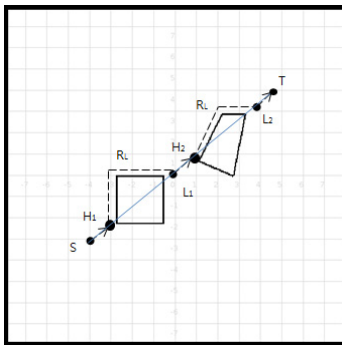


그림 2 Bug2 알고리즘

이러한 특성으로 인해 대부분의 경우 Bug2 알고리즘이 Bug1 알고리즘보다 더 짧은 경로를 탐색한다. Bug2 이동 길이는 Bug1 보다 짧다는 것은 그림 2에서 알 수 있다. 그러나 그림 3과 같은 맵의 경우에는 Bug2 로봇의 이동 거리가 Bug1 보다 길어진다.

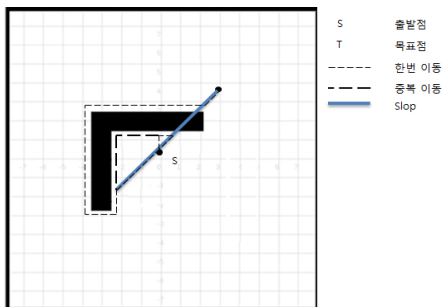


그림 3 Bug2 알고리즘의 단점

### 2.4 PBug2 알고리즘

PSBug1 알고리즘은 Bug1 알고리즘에 로봇 두 대를 이용한 알고리즘이다. 두 대의 로봇이 각각 다른 방향으로 장애물 주위를 돈다. 두 대 모두 장애물을 도는 동안 목표 지점의 거리 계산하여 떠나는 지점을 결정한다. 그리고 떠나는 지점을 로봇 두 대 정보를 공유한다. 로봇 두대가 만나는 지점(meeting point)에서 장애물을 모두 탐사하였기 때문에 떠나는 지점으로 이동한다. 이후 목표지점까지 모든 작업을 반복하여 수행한다. 그림 4에서 PPSBug1 알고리즘을 확인할 수 있다.

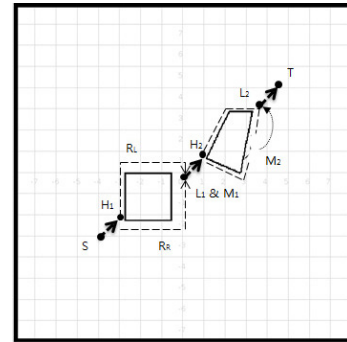


그림 4 PPSBug1 알고리즘

### 2.4 PPSBug2 알고리즘

PSBug2 알고리즘은 PPSBug1 알고리즘과 같이 Bug2 알고리즘에 로봇 두 대를 이용한 알고리즘이다. 로봇 두 대가 각각 다른 방향으로 장애물 주위를 돈다. 두 대의 로봇이 장애물을 도는 과정에서 시작 지점과 목표 지점의 라인을 만나게 되면 로봇이 떠난다. 이후 과정은 목표 지점을 만날 때까지 계속 반복작업을 수행한다. 그림 5에서 PPSBug2 알고리즘 수행 예를 확인할 수 있다.

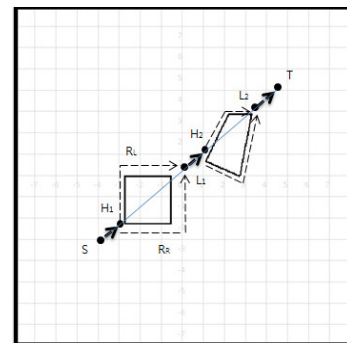


그림 5 PPSBug2 알고리즘

## 3. 구현 환경

본 연구는 리눅스 환경에서 구동되는 Player/Stage 시뮬레이션을 이용하여 실험을 진행하였다[5]. 로봇관련 명령어를 처리하기 위해 Python 언어를 사용하였다. 또한, 로봇의 작업 환경 시뮬레이션을 위한 이동 경로의 장애물은 직접 이미지 파일로 만들어 주위환경을 설정하였다.

### 3.1 시뮬레이션 환경

시뮬레이션 환경에서 모바일 로봇은 자기 위치를 두고, 그 위치와 지시데이터를 업데이트하기 위해 이동 로봇이 이동 중에 위치를 측정할 수 있도록 하였다. 특히, Bug2와 PBug2 알고리즘에서 직선경로를 지나치지 않도록 모바일 로봇의 위치를 자주 측정하는 것이 중요하다.

이 연구에서는 사용된 맵은 길이 16mm, 폭 16mm로 구성된 2차원 맵에 두 개의 장애물을 구성하였다. 그림 6에서 확인할 수 있다. 그리고 Bug 알고리즘의 효율을 확인하기 위해 시작지점과 목표지점을 설정하였다. 각 로봇은 최초 시작지점에서 목표지점까지 이동하면 된다.

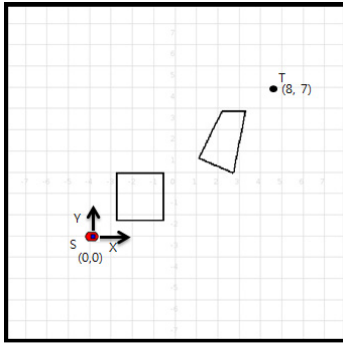
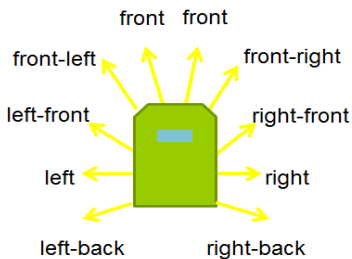


그림 6 Pyro 시뮬레이션 환경

### 3.2 시뮬레이션

본 연구에 사용되는 버그 알고리즘은 범위센서가 필요하다. 장애물 감지와 벽면주행은 파이오니어 로봇의 초음파센서를 사용하여 수행하였다. 파이오니어 로봇의 초음파센서 및 이름은 그림 7에서 센서를 확인할 수 있다.

장애물 감지하는 전방 센서는 front, front-right, front-left, left-front, right-front를 사용하였다. 방향을 바꾸거나 벽면주행에 필요한 측면센서는 left, right, left-back, right-back을 사용하였다. 센서의 개수는 최대 16방향까지 탑재할 수 있지만 본 연구에서는 10개의 방향센서만 사용하였다.



(그림 7) 초음파 센서 배치

## 4. 실험 및 결과

본 연구에서는 Bug 알고리즘을 이용하여 로봇의 능력을 비교하는 실험을 하였다. 각각 로봇 1대, 2대를 이용해 Start 지점에서 탐사를 시작하여 Target 지점까지 이동한 시간과 거리를 측정하였다. 로봇들은 10개의 range 센서를 이용해 왼쪽 벽과 오른쪽 벽을 따라 이동한다. 먼저 Bug1과 Bug2 알고리즘을 실험하고 PBug 알고리즘을 구현하여 비교하였다.

Bug1 알고리즘을 적용한 로봇은 테스트된 버그 알고리즘 중 가장 느렸다. Bug1 알고리즘은 장애물 전체를 탐색한 후 장애물에서부터 다음 장애물까지 가장 짧은 거리를 계산하여 이동하기 때문이다. 실제 로봇이 이동한 거리는 38.75이며 그림 8에서 확인할 수 있다.

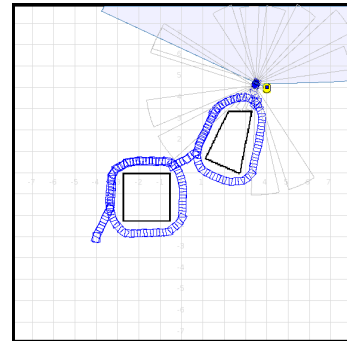


그림 8 Bug1 알고리즘

Bug2 알고리즘은 초기 Slope을 설정한다. Slope이란 Start 위치에서 Target 위치까지 일직선을 그었을 때를 나타낸다. 앞서 3장에서 언급했듯이 로봇이 Slope을 만나게 되면 장애물을 만날 때까지 Slope을 따라 이동한다. 이 연구에서 측정된 거리는 14.27이며 그림 9에서 확인할 수 있다.

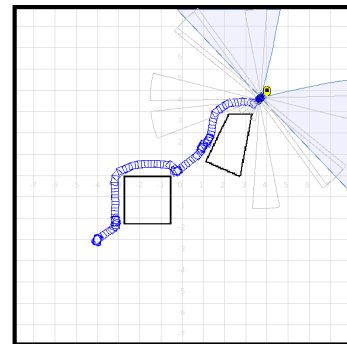


그림 9 Bug2 알고리즘

PBug 알고리즘은 초기 2대의 로봇이 충돌을 방지하기 위해 첫 번째 로봇 위치에서 X= -1, Y= -1 정도에 위치하도록 하였다. PBug 알고리즘은 두 대의 로봇이 만나거나 장애물을 모두 탐색했을 때까지 장애물을 돌다가 다음 장애물에 가장 가까운 위치로 이동하게 하였다. 이 실험에서는 Bug1 알고리즘보다 이동 거리가 짧아졌는데 그 결과는 그림 10에서 확인할 수 있다.

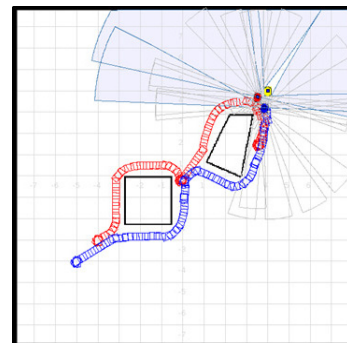


그림 10 PBug1 알고리즘

PBug2 알고리즘은 Bug2 알고리즘에서 사용한 직선 개념을 그대로 사용하였다. 두 대의 로봇은 각각 다른 방향으로 이동하게 하였으며 먼저 직선을 만난 로봇이 직선을 따라 이동하게 하였다. 만약 한 대의 로봇이 비슷한 시간에 직선을 만났을 때 먼저 만난 로봇이 첫 번째 장애물을 떠날 때까지 기다리다가 장애물을 떠난 후에 출발하도록 하였다.

이 연구에서 사용한 PBug2 알고리즘의 총 이동 거리는 14.27mm으로 Bug2 알고리즘과 비슷한 이동 거리 유지했다. 그림 11에서 확인할 수 있다.

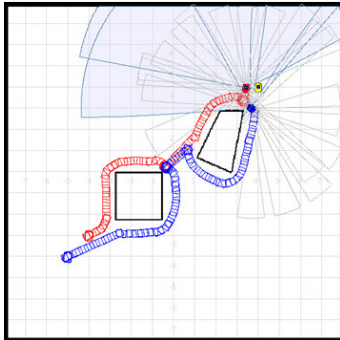


그림 11 PBug2 알고리즘

본 실험을 통해 4가지 실험 결과 값은 표 1에서 알 수 있다. 결과적으로 Bug1 알고리즘에 따른 이동 경로가 가장 긴 것으로 나타났다. 표 1을 보면, 시작 위치에서 목표 지점까지의 직선거리는 10.65mm이며 Bug1, PBug1, Bug2 그리고 PBug2의 최종 목표지점까지 이동거리는 각각 38.75mm, 20.05m, 14.27mm, 14.27mm이다. Bug1과 PBug1의 이동거리의 차이는 18.70mm로 나타났으며 Bug2와 PBug2의 이동거리는 허용 오차 내에서 같은 것으로 나타났다.

Bug1이 목표지점까지 이동시간은 32.5초이며 PBug1의 이동시간은 21.1초로 상대적으로 35.1% 정도 시간을 단축시켰다. Bug2의 경우 PBug2와 비교했을 때 비슷한 시간을 보였다. 또한, PBug2는 장애물을 모두 탐사할 수 있어서 목표지점까지 이동할 수 있다는 것을 보증하였으며 그림 3의 경우가 발생하더라도 시간상으로 단축할 수 있다.

<표 1> Bug 알고리즘 따른

	경로 길이 (mm)	시간 (sec)	탐사 시간 PBug/Bug (%)	장애물을 모두 탐사하는 지 여부
직선거리	10.65	-	-	-
Bug1	38.75	32.5	-	O
PBug1	20.05	21.1	64.9	O
Bug2	14.27	16.4	-	X
PBug2	14.27	16.6	101.2	O

## 5. 연구 결과 및 결론

본 연구는 Bug 알고리즘 중 모션 계획 알고리즘인 Bug1과 Bug2 알고리즘을 개선한 PBug1, PBug2 알고리즘 제안하고 비교해 보았다. 알고리즘을 비교하기 위해 같은 환경을 구성하였다. 먼저 Bug1 알고리즘은 4개의 알고리즘 중 가장 긴 이동 거리를 나타냈다. Bug2는 Bug1 알고리즘 보다 나은 14.27mm가 나왔으며 PBug1 알고리즘은 20.05mm Bug1 알고리즘은 18.7mm로 상대 비율 64.9% 더 작게 나왔다. 마지막으로 PBug2 알고리즘은 Bug2 알고리즘과 같은 이동거리를 유지하였지만, 장애물을 모두 탐사할 수 있었다. 본 연구를 통해 흩어져 있는 이동 로봇을 군집 형태로 형상화하기 위한 발판으로 Bug 알고리즘을 테스트해보았다.

아직 군집화 형태로 구현하기에는 본 연구에서 제시되지 않은 알고리즘을 구현해보는 것이 필요하다. 그리고 또한, 본 연구를 통해 알게 된 Bug 알고리즘을 이동하는 여러 로봇을 적용하여 군집화 형태로 구성해보는 것 또한 필요할 것이다. 마지막으로 PBug1 알고리즘을 응용한다면 추후 로봇이 더 추가되었을 때 3번째 로봇에 최단 경로 정보를 얻을 수 있을 것이다.

### 참고문헌

- [1] Craig Reynolds, "Boids: Background and Update," <http://www.red3d.com/cwr/boids/>, 1986
- [2] Wenfeng and Weiming Shen Li, "Swarm behavior control of mobile multi-robots with wireless sensor networks," *Journal of Network and Computer Applications*, 34(4):1398 - 1407, 2011
- [3] Vladimir J. Lumelsky, Stepanov A., "Dynamic path planning for a mobile automaton with limited information on the environment," *IEEE Transactions on Automatic Control*, 31(11): 1058-1063, 1986.
- [4] James Ng and Thomas Brunl, "Performance Comparison of Bug Navigation Algorithms," *Journal of Intelligent & Robotic Systems*, 50(1): 73-84, 2007
- [5] Yufka A, Parlaktuna O. "Performance comparison of bug algorithm for mobile robots," 5th international advanced technologies symposium, Karabuk, Turkey, 2009.
- [6] Douglas Blank, Deepak Kumar, Lisa Meeden, and Holly Yanco, "The Pyro toolkit for AI and robotics," *AI Magazine*, 27(1): 30-50, 2006