

다중이동로봇의 장애물 회피 및 경로계획 알고리즘에 관한 연구

박 경 진*, 이 기 성, 이 중 수
홍익대학교 전기제어공학과
전화 : (02) 320-1493

A Study on the Obstacle Avoidance and Path Planning Algorithm of Multiple Mobile Robot

Kyung Jin Park*, Ki Sung Lee, Jong Soo Lee
School of Electrical and Control Engineering Hongik University
E-mail : pkjni@nownuri.net

Abstract

In this paper, we design an optimal path for multiple mobile robots. For this purpose, we propose a new method of path planning for multiple mobile robots in dynamic environment. First, every mobile robot searches a global path using a distance transform algorithm. Then we put subgoals at crooked path points and optimize them. And finally to obtain an optimal on-line local path, every mobile robot searches a new path with static and dynamic obstacle avoidance.

성하여 주어진 목표점까지 최단거리로 이동시키는 경로계획에 관하여 연구하였다. 본 논문의 경로계획 과정은 셀 분해(Cell Decomposition) 과정을 거친 후 거리 변환법(Distance Transform)을 적용하는 단계로 구현하였고, 경로곡선상에 중간 경유점(Subgoal)을 두어 거리 변환으로 구한 최적의 경로를 따라 주행하도록 하였다. 또한, 중간 경유점을 최적화하여 이동 로봇의 주행시 방향 전환 횟수를 감소시켜 최단 거리 및 시간으로 주행할 수 있도록 하였다. 본 논문에서는 제안한 알고리즘의 타당성을 검토하기 위하여 여러대의 로봇과 고정된 장애물들이 있는 환경하에서 모의실험 하였다.

I. 서론

현재 산업현장에서 응용되는 대부분의 로봇방식은 매니플레이터방식으로, 작업공간의 한계성이 있었다. 이러한 단점을 보완하기 위하여 자율주행로봇방식이 필요하게 되었다. 최근에는 한 대의 로봇으로 해결하기 어려운 문제를 여러 이동로봇으로 해결하는 방식이 대두되고 있으며, Service Robot의 비중의 증대로 동적환경하에서의 경로계획에 대한 필요성이 커져가고 있다. 이를 위해, 본 논문에서는 위와 같은 점에 주안하여, 이동로봇을 고정된, 움직이는(이동로봇) 장애물이 있는 환경하에서 장애물 회피시마다 최단경로를 생

II. 셀 분해(Cell Decomposition) 과정

경로계획을 하는 과정에서, 활동영역을 묘사하기 위한 과정으로 본 논문에서는 셀 분해법을 적용하였다. 셀 분해법은 활동영역을 셀 단위로 처리하기 때문에 활동영역에 대한 데이터가 줄어들고 알고리즘이 간단하며 반복적으로 구현할 수 있다는 장점이 있다. 본 논문에서는, 셀 분해 과정을 다음과 같은 방법으로 하였다.

- 한 셀을 로봇이 통과할 수 있는 크기로 나누었다.
- 로봇의 중심은 항상 셀의 중심에 위치한다.

- 한 셀의 크기를 로봇 크기의 125%로 정하였다.
(로봇은 한 셀의 80%를 점유)

또한, 셀(cell) 점유에 있어서는 다음과 같이 분류하였다.

$$Cell(row, column) = \begin{cases} 0: \text{이동가능 cell} \\ 1: \text{이동불가능 cell} \end{cases}$$

III. 거리 변환(Distance Transform)

셀 분해 과정을 거친 후 이동로봇의 전체 map에 대한 최적경로를 찾기 위한 탐색 알고리즘으로는 깊이 우선 탐색(Depth First Search), A*, 거리 변환(DT) 알고리즘 등이 있다. 이중 거리 변환은 숫자를 확산시키며 최단거리 경로를 찾는 알고리즘으로 다른 탐색 알고리즘에 비해 최적경로를 보장하고, 프로그램으로 구현하기가 간단하다는 장점을 가지고 있다. 본 논문에서, 거리 변환(DT)을 이용한 경로 계획하는데 있어서의 전체 조건은 다음과 같다.

첫째, 실제 장애물의 크기를 로봇의 반지름(r) 보다 크게 잡아 확장된 장애물로 보았다.

(단, 로봇의 중심은 장애물 셀에 이동 불가)

둘째, 로봇의 궤적을 점으로 축소하여 표현하였다.

셋째, 8 connected 방법을 적용하였다.

일반화된 거리 변환(DT) 알고리즘은 다음과 같다.

- step1. Start(i,j), Blocked(i,j), Goal(i,j) 배열 정의
- step2. Cell(i,j) 배열의 초기화
- step3. Cell(i,j) 배열안의 값 계산
 - 단계1. J1->Jmax, I1->Imax (정방향 탐색)
 - 단계2. Jmax->J1, Imax->I1 (역방향 탐색)
- step4. Cell 값 변화 유무 판별
- step5. i) 변화 有 - goto step3.
ii) 변화 無 - 주변 8개의 Cell값 중 가장 작은 값 선택
- step6. Goal

IV. 중간경유점(Subgoal) 설정 및 최적화

중간 경유점(subgoal)은 최적화된 경로를 따라주행하는데 있어서, 꼭 거쳐가야 할 임시 목표점이라고 할 수 있다. 본 논문에서는, 여러 이동 로봇의 주행시 경로 계획하는데 있어서 off-line에서 거리 변환(DT)을 이용하여 구한 최적 경로를 따라 주행하기 위해 경로 곡선상의 구부러지는 경로(crooked path) 지점에 중간 경유점을 두었으며, 이동 로봇의 주행시 방향 전환 횟

수를 감소시켜 최단 거리 및 시간으로 주행하기 위하여 설정된 중간 경유점을 최적화하였다. 중간 경유점 설정은 위에서 설명한 바와 같이, crooked path 지점에 두었으며 설정된 중간 경유점을 최적화시키는 방법은 다음과 같다.

- 중간 경유점(subgoal) 최적화 알고리즘

- step1. subgoal $G_{s_1}(=S), G_{s_2}, G_{s_3}, \dots, G_{s_n}(=G)$,
i=1로 설정
- step2. i) 임의의 세 subgoal $G_{s_i}, G_{s_{i+1}}, G_{s_{i+2}}$ 에서,
 G_{s_i} 에서 $G_{s_{i+2}}$ 로의 visibility test 수행
(최단거리 궤적상의 장애물 유무 검사)
ii) Goal에 도착한 경우
단계1. subgoal 기록
($G_{s_1}(=S), G_{s_2}, G_{s_3}, \dots, G_{s_n}(=G)$)
단계2. 이전의 subgoal 기록과 비교 후,
변화유무 판별
 - i) 변화 有
:이전의 subgoal 기록 삭제 및
새로운 subgoal 기록 후 goto step2.
 - ii) 변화 無
:subgoal 최적화 종료
- step3. i) test 성공시(경로 가능),
: subgoal $G_{s_{i+1}}$ 삭제 및 $G_{s_{i+1}} \leftarrow G_{s_{i+2}}$
($k=i, i+1, \dots, n-2$)
ii) test 실패시(경로 불가),
: $i \leftarrow i+1$ 후 goto step2.
- step4. goto step2.
step2.에 실시한 장애물 유무 검사 방법은 다음과 같다.
 - visibility test 알고리즘
 - step1. 로봇의 현재 위치를 위치좌표 (x, y)로 표현
 - step2. 로봇의 위치좌표 (x, y)를 cell 좌표 (row, column)로 표현
 - step3. 해당하는 cell 좌표 (row, column)의 cell 값 ('0' or '1')으로 장애물 유무 판별

위와 같은 방법으로 구한 중간 경유점(subgoal)을 아래 그림에 나타내었다. 그림1.은 거리 변환(DT)에 의한 경로 곡선이며, 그림2.는 visibility test(장애물 유무 검사) 과정의 가상 경로(virtual path)를 나타낸 것이다. 끝으로 최적화된 중간 경유점을 이용하여 off-line 상태에서의 최적에 가까운 경로 곡선을 구한 것이 그림3.이다.

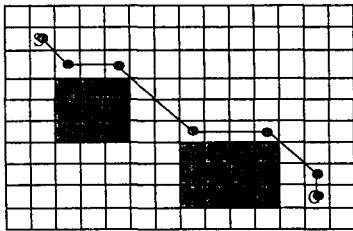


그림 1. 거리변환(DT)에 의한 경로계획

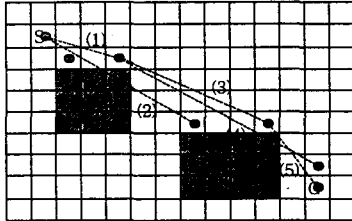


그림 2. visibility test 과정(virtual path 형성)

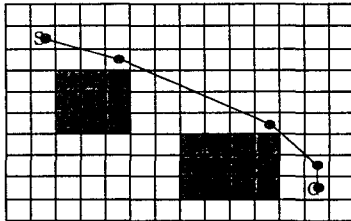


그림 3. 최적화된 subgoal에 의한 경로계획

V. 충돌회피 알고리즘

본 논문에서, 각 이동 로봇은 off-line 상태에서 거리 변환(DT)으로 구한 경로를 바탕으로 주행하도록 하였으며, on-line 상태에서도 로봇의 위치는 셀(cell)의 행(row) 또는 열(column)의 중심에 위치하도록 하였다. 또한, 이동 경로 계획은 셀(cell) 단위로 이루어지며, 각 이동 로봇은 현 위치 좌표(x,y)에서 다음 셀의 위치 좌표(x',y')에서의 충돌 여부를 검사하게 된다. 이동 로봇간의 충돌회피를 위한 경로 계획 방법으로는 로봇의 진행 방향 순서를 정한 후, 로봇간의 우선 순위(priority)를 두어 상호간의 원활한 충돌회피가 이루어지도록 하였다. 그리고, 충돌 회피한 후 각 이동 로봇은 중간 경유점 선정 작업을 거친 후에 중간 경유점으로의 최단 거리 경로 계획을 한다.

(1) 이동로봇의 주행방향 순서

본 논문에서는, 그림4와 같이 이동 로봇의 주행 방향 결정에 있어서 8-방향을 고려하였다. 또한, 가능한 한 2 or 2'와 같은 대각선 방향이나 현 위치에 대기하

도록 하여 주행시 방향회전을 줄이도록 하였다. 그리고, 2 or 2'의 경우에 주행 방향 결정에 있어서는 각 위치에서 중간 경유점(subgoal)까지의 visibility test (장애물 유무 검사)를 거친 후, 중간 경유점과 가까운 방향으로 이동한 후 주행하도록 하였다. 그리고 모든 방향으로의 경로가 막힌 경우를 고려하여 5 방향을 두어 경로 계획 하였다.

- 이동로봇의 주행 순서 (1방향으로 주행 상태)

- step1. 1 방향으로 주행
- step2. 1 방향으로 진행 불가능인 경우, 2 or 2' 방향으로 주행 (visibility test 후, subgoal과 가까운 방향으로 이동)
- step3. 2 or 2' 방향으로 진행 불가능인 경우, 현 위치 (3)에 대기
- step4. 2회 이상 대기시에는, 4 or 4' 방향으로 주행 (visibility test 후, subgoal과 가까운 방향으로 이동)
- step5. 4 or 4' 방향으로 진행 불가능인 경우, 5 방향으로 주행 (5 방향으로도 진행 불가능인 경우, 대기)

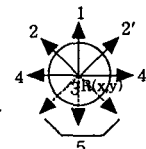


그림 4. 이동로봇의 주행 순서

(2) 위치계산 방법 및 중간경유점 선정

이동 로봇간의 회피 후 중간 경유점(subgoal) 까지 경로 계획하는데 있어서 중간 경유점까지의 위치계산 방법을 살펴보면 다음과 같다.

- 중간 경유점의 위치가 $S(x,y)$ 인 경우, 로봇의 현재위치 $R(x,y)$ 와 중간경유점 $S(x,y)$ 와의 (row,column)을 고려하여,

- 1) row > column : row cell만큼 위치계산후 주행
- 2) column > row : column cell만큼 위치계산후 주행

또한, 이동 로봇간의 충돌회피 후 중간 경유점으로의 경로 계획에 있어서 중간 경유점 선정 방법은 다음과 같다.

- 중간 경유점(Subgoal) 선정 방법

- 1) 중간 경유점으로의 위치 계산 방법이 row cell에 따른 경우
: 이동 로봇의 현 위치좌표 $R(row,column)$ 를 기준으로 하여 R 의 column 좌표와 Goal 및 subgoal

(row,column)의 column 좌표를 비교한다.

비교 후, visibility test를 거친 후 중간 경유점을 선정한다.

2) 중간 경유점으로의 위치 계산 방법이 column cell 에 따른 경우

: 이동 로봇의 현 위치좌표 $R(row, column)$ 를 기준으로 하여 R 의 row 좌표와 Goal 및 subgoal (row,column)의 row 좌표를 비교한다.

비교 후, visibility test를 거친 후 중간 경유점을 선정한다.

(3) 이동로봇간의 충돌회피

본 논문에서는, 이동 로봇간의 충돌회피 후 중간 경유점(subgoal) 까지 경로 계획하는데 있어서 각 로봇의 중심 위치 좌표간의 거리측정에 의한 방법으로 경로를 구하였다. 또한, 이동로봇의 현(t_0) 위치와 다음(t_1) 위치의 중간 위치좌표에서도 거리측정을 하여 안전한 주행을 할 수 있도록 하였다. on-line 상태에서의 visibility test(장애물 유무 검사)를 통한 충돌회피 알고리즘은 그림5와 같다. 로봇과 중간 경유점의 위치 좌표는 아래와 같다.

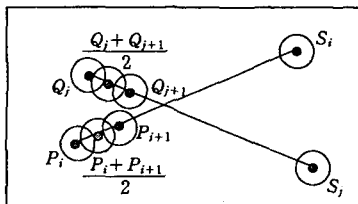


그림 5. 이동로봇의 이동경로 계획

R_i 의 현(t_0) 위치좌표 $P_i(x, y)$

다음(t_1) 위치좌표 $P_{i+1}(x', y')$

R_j 의 현(t_0) 위치좌표 $Q_j(x, y)$

다음(t_1) 위치좌표 $Q_{j+1}(x', y')$

R_i 경로상의 중간위치좌표 $\frac{P_i(x, y) + P_{i+1}(x', y')}{2}$

R_j 경로상의 중간위치좌표 $\frac{Q_j(x, y) + Q_{j+1}(x', y')}{2}$

R_i 의 subgoal좌표 $S_i(x, y)$, R_j 의 subgoal좌표 $S_j(x, y)$

- 충돌회피 알고리즘 (R_i 에서의 거리 값 계산)

step1. 로봇의 현 위치 $P_i(x, y)$ 에서 subgoal $S_i(x, y)$ 까지 최단거리로 이동

step2. (i) 로봇(R_i)은 다음 cell로 이동하기 전에,

$$|P_{i+1}(x', y') - Q_{j+1}(x', y')| \text{ 거리 구함}$$

(ii) 또한, 중간위치좌표에서도 거리 구함

$$\left| \frac{P_i(x, y) + P_{i+1}(x', y')}{2} - \frac{Q_j(x, y) + Q_{j+1}(x', y')}{2} \right|$$

step3. (i) 에서 구한 거리 값과 로봇지름($\neq 2.5r$)과의 크기 비교

(ii) 에서 구한 거리 값도 같은 방법으로 로봇 지름($\neq 2.5r$)과 크기 비교

만일, (i) or (ii)에서 구한 거리값 \leq 로봇지름($\neq 2.5r$) 충돌위험이 있으므로 priority가 낮은 로봇은 회피 or 대기한 후 goto step1.

step4. 로봇(R_j)은 진행 방향으로 한 cell 주행한 후, goto step2.

VI. 결론 및 추후 연구

본 논문에서는 이동 로봇의 경로 계획을 기존의 정적인 환경에서 동적인 환경으로 확대하여 연구하였다. 또한, 제안한 알고리즘으로 단일 로봇이 아닌 여러 이동로봇의 경로 계획에도 적용 가능함을 알 수 있었고, 또한, 기존의 거리변환(DT) 경로 곡선상에 중간 경유점(subgoal)을 두어 목표점까지 최단 거리, 시간으로 경로 계획함을 보였다. 앞으로의 향후 과제로는 실제 이동로봇에의 적용과 다양한 환경에서의 주행 실험시 발생 가능한 문제점들을 해결하는 방향으로 연구가 이루어져야겠다.

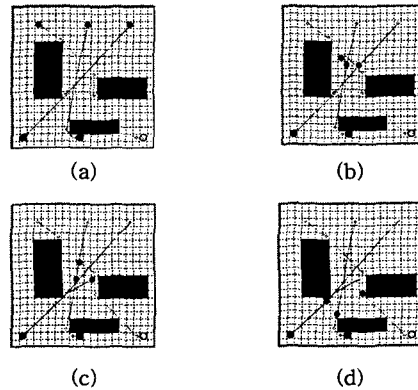


그림 6. 여러대의 이동로봇의 경로계획

참고 문헌(Reference)

[1]남윤석, "이동장애물 동작의 모델링 및 이동로봇의 동적운행에의 적용" 서울대 박사학위논문, 1997
 [2]김상문, "거리변환과 VFH를 이용한 이동로봇의 장애물 회피" 홍익대학교 석사학위논문, 1997
 [3]김도원, "다중이동로봇을 위한 온라인 경로계획", 한국과학기술원 석사학위논문, 1997
 [4]Philip John Mckerrow, "Introduction to Robotics", Addison Wesley Publishing, 1993
 [5]Jean-Claude Latombe, "Robot Motion Planning", Kluwer Academic Publishers
 [6]T. Arai and J. Jota, "Motion Plannong of Multiple Mobile Robots", Proc. IEEE International Conference on Intelligent Robots and Automation, pp.1761-1768, 1992
 [7]R.A. Jarvis, "Distance Transform Based Path Planning for Robot Navigation", Recent Trends in Mobile Robots, pp.3-31, 1993