

자율이동로봇을 위한 효율적 경로 계획 방법

윤희상, 박태형
충북대학교 제어계측공학과 및 CBITRC

Effective Path-Planning for Autonomous Mobile Robots

Hee-Sang Yoon, Jin-Oh You, Tae-Hyoung Park
Dept. of Control and Instrumentation Eng., CBITRC, Chungbuk National University

Abstract - 자율이동로봇을 위한 새로운 경로 계획 방법을 제안한다. 실시간으로 장애물을 피하고, 목표 지점까지의 최단 경로를 생성하여 유용성을 극대화시키기 위하여 방법을 다룬다. 본 논문에서는 효율적인 경로 계획 방법으로 초기 경로를 생성하고, 생성된 경로를 개선하는 방법을 제안한다. 초기 경로는 그래프 기반 방법인 골격선 그래프와 탐색방법으로 디스트라(Dijkstra) 알고리즘을 사용한다. 초기 경로에 대해 동적 프로그래밍 알고리즘을 이용하여 최단거리에 가깝게 경로를 개선한다. 시뮬레이션을 통해 제안하는 방법의 성능을 검증한다.

경유점 v_k 와 v_{k+1} 을 연결하는 직선을 간선 (v_k, v_{k+1}) 이라 정의하면, 식 (3)과 같은 장애물의 회피 조건이 만족되어야 한다.

$$(v_k, v_{k+1}) \notin O_i, \forall i \in \{1, \dots, M\}, k=1, \dots, M-1 \quad (3)$$

실제 많은 이동 경로가 존재하며, 최단거리를 찾는 것이 경로 계획문제이다. 간선 (v_k, v_{k+1}) 의 길이가 $d(v_k, v_{k+1})$ 라 하면, 이동경로의 총 길이는 식 (4)와 같다.

$$\sum_{k=1}^{M-1} d(v_k, v_{k+1}) \quad (4)$$

1. 서 론

자동화 공장에서 물류의 이동을 담당하는 무인운반차 및 청소로봇, 안내 로봇 등 다양한 로봇에 대해 자율이동기능이 포함 되어있다. 이런 자율이동 로봇의 주행성능을 높이기 위해서, 현재 지점에서 목표 지점까지 최단 시간에 이동할 수 있도록 경로를 생성하는 경로계획 기능이 중요시 된다. 그림 1은 자율이동로봇이 현재 지점에서 목표 지점까지 많은 경로에서 최단 경로를 찾는 경로계획 문제를 나타낸다[1].

자율이동로봇의 경로계획에 관한 연구는 크게 그래프 기반의 방법과 비 그래프 기반의 방법으로 구분할 수 있다. 그래프 기반의 방법은 그래프 지도를 구성하고, 그래프 탐색을 통하여 경로를 생성하는 방법이다. 비 그래프 기반의 방법은 그래프 지도를 사용하지 않고 경로를 생성하는 방법이다. 최단 경로를 생성하는 측면에서는 그래프 기반의 방법이 보다 효율적인 방법으로 알려져 있다.

그래프 기반의 방법은 가시도 그래프 기반의 방법과 골격선 그래프기반으로 구분될 수 있으며 그림 2에 나타내었다. 골격선 그래프 기반의 방법은 세션화 알고리즘 및 보르노이 알고리즘 등을 적용하여 그래프를 생성할 수 있으며, 경로의 탐색은 디스트라 알고리즘 및 A*와 같은 방법을 사용할 수 있다.

본 논문은 경로를 빠르고 최단 경로에 근사하게 생성하는 실용적 경로계획 방법을 제안한다. 비교적 빠르게 경로를 생성하는 골격선 그래프 기반의 방법에, 경로를 개선하는 단계를 추가시켜 기존의 정적인 경로계획 방법이 아니라, 그래프 구성과 경로탐색, 개선을 계속 변화시키는 동적계획 방법이다. 제안하는 방법은 경로 생성 단계와 경로 개선 단계로 구성되며, 각 단계는 탐색 과정을 포함한다. 경로 생성 단계는 세션화 알고리즘으로 골격선 그래프를 구성하고, 디스트라 알고리즘을 적용하여 경로를 탐색하여 초기 경로를 생성한다. 경로 개선 단계에서는 동적 프로그래밍 (dynamic programming, DP) 알고리즘을 적용하여, 그래프 및 경로를 동시에 개선시킨다.

2. 경로 계획 문제

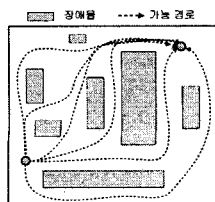
이차원 평면에서 현재 로봇의 위치를 $s \in R^2$ 라 정의하고 목표점을 $g \in R^2$ 라 정의한다. 시작점과 목표점사이에 존재하는 N 개의 장애물은 다음과 같다.

$$O_i = \{(x, y) \mid l_i \leq x \leq r_i, b_i \leq y \leq t_i\}, i=1, \dots, N \quad (1)$$

여기서 l_i, r_i 는 각각 장애물의 좌한 및 우한값이며, b_i 및 t_i 는 장애물의 하한 및 상한 값이다.

이동로봇은 점의 이동으로 표현하고, 시작점과 목표점 사이의 경유점 $v_k \in R^2$ ($k=1, \dots, M$) 들의 순열 집합은 이동경로 p 로 표현될 수 있다.

$$p = \langle v_1, v_2, \dots, v_M \rangle, (v_1 = s, v_M = g) \quad (2)$$



〈그림 1〉 경로계획 문제

3. 경로 계획 방법

본 논문은 자율이동로봇의 경로 계획 문제에 국지적 최적화 기법을 적용하여, 전체 단계를 초기 경로를 생성하는 경로 생성 단계와 경로를 개선하는 경로 개선 단계로 구분하여 진행된다. 경로 생성 단계에서는 골격선 기법을 이용 하여 그래프를 구하고 이를 탐색하여 초기경로를 구한다. 경로 개선 단계는 동적 프로그래밍 기법을 적용하며, 초기경로를 반복적으로 개선하여 이동거리를 단축시킨다.

3.1 경로 생성 단계

3.1.1 그래프 구성

시작점, 목표점 및 장애물이 주어진 이차원 공간에서, 다음의 정점의 집합 Γ 과 간선의 집합 Δ 로 구성되는 그래프 (Γ, Δ) 를 구성하는 단계이다.

$$\Gamma = \{n_1, n_2, \dots, n_r\} \quad (5)$$

$$\Delta = \{(n_i, n_j) \mid i \neq j, n_i, n_j \in \Gamma, (n_i, n_j) \notin O_k, \forall k \in \{1, \dots, M\}\} \quad (6)$$

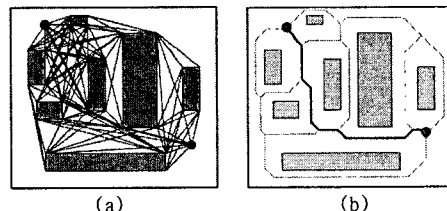
(6) 은 모든 간선과 장애물은 서로 간섭하지 않아야 함을 의미한다.

자율이동로봇의 경로계획에 필요한 골격선 그래프를 생성하기 위한 여러 가지 방법들이 발표되었다. 세션화 알고리즘[2] 은 영상처리 과정의 하나인 골격화에 사용되는 방법으로서, 경로계획 문제의 그래프를 구하는 데 효과적으로 적용될 수 있다. 세션화 알고리즘을 적용하여 그래프를 구성하는 과정은 그림 3에 나타내었다.

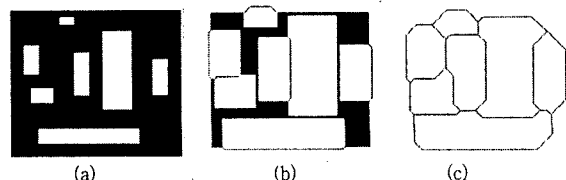
3.1.2 경로 탐색

구성된 그래프에 대하여, 시작점에서 목표점에 이르는 최단 경로를 찾는 단계이다. 간선 길이의 총합이 최소화 정점들의 순서 조합을 찾는 문제로서, 최단경로문제에 해당한다. 디스트라 알고리즘 [3] 은 최단경로 문제의 해를 구하는 대표적인 알고리즘으로서, 구현이 용이하고 최적의 해를 구할 수 있다.

3.2 경로 개선 단계



〈그림 2〉 경로계획 방법. (a) 가시도 그래프 기반 (b) 골격선 그래프기반



〈그림 3〉 그래프 구성 과정 예. (a) 최초 비트맵 (b) 골격화 과정 (반복횟수=10) (c) 최종 비트맵.

경로 개선 단계는 초기 경로를 구성하는 경유점들의 위치를 변경하여, 이동거리의 총합을 단축시키는 단계이다. 본 논문은 골격선 그래프 상에 생성된 경로를 개선시키기 위하여 격자지도 (grid map) 를 도입한다. 초기 경로 $p = \langle v_1, \dots, v_M \rangle$ 를 구성하는 경유점 중 시작점 v_1 및 목표점 v_M 을 제외한 나머지 경유점 v_2, \dots, v_{M-1} 에 대하여, 각 경유점을 중심으로 한 평면에 $W \times W$ 개의 정방형 격자를 그림 4 와 같이 구성한다. 이 때 경유점 $v_k (k=2, \dots, M-1)$ 를 중심으로 한 격자점들을 $v_{k,1}, \dots, v_{k,W}$ 이라 표시한다. 경유점이 격자지도 상의 다른 격자점으로 변경되는 경우, 전체 이동거리가 단축될 수 있다.

격자지도에서 최적의 격자점을 선택하는 문제는 동적 프로그래밍 (dynamic programming, DP) 알고리즘을 적용한다. DP 는 최적성의 원리 [4] 에 의하여 최적성이 입증된 최적화 기법으로서, 광범위한 범주의 최적화 문제에 널리 활용되고 있다 [5]. DP 의 적용을 위해서는 스테이지 (stage) 와 스테이트 (state) 로 구성된 탐색평면이 필요하며, 그림 5 은 그림 4 의 격자그래프로부터 구성된 탐색평면을 보여준다. M 개의 경유점 v_1, \dots, v_M 을 스테이지로 하고, 각 경유점 v_k 에 대한 W^2 개의 격자점 $v_{k,1}, \dots, v_{k,W^2}$ 을 스테이트로 하는 탐색평면이다. 시작점에서 격자점 $v_{k,\alpha}$ 까지의 최소 누적 거리를 $d_{k,\alpha}^*$ 표시하고, 격자점 $v_{k,\alpha}$ 와 $v_{k+1,\beta}$ 사이의 거리를 $d_{(k,\alpha)(k+1,\beta)}$ 라 표시한다.

경로 개선 알고리즘은 다음과 같다.

S1. 스테이지 $k=1$ 에 대하여 최소누적거리 $d_{1,1}^* = 0$ 으로 설정한다.

S2. 스테이지를 $k=2, \dots, M-1$ 로 증가시키면서, 각 격자점 $v_{k,\alpha}$ 에 대한 최소누적거리 $d_{k,\alpha}^*$ 를 다음과 같이 계산하여 저장한다.

$$d_{k,\alpha}^* = \min (d_{k-1,\beta}^* + d_{(k-1,\beta)(k,\alpha)}) \quad (7)$$

$d_{k,\alpha}^*$ 가 생성되는 β 값을 지시자 $\pi_{k,\alpha}$ 로 저장한다.

S3. 최종 스테이지 $k=M$ 에 대하여, 시작점부터 목표점까지의 최소누적거리 $d_{M,1}^*$ 를 (7) 와 같이 구하고 지시자 $\pi_{M,1}$ 을 저장한다.

S4. 스테이지를 $k=M, \dots, 2$ 로 감소시키면서 지시자를 역 추적하며, 각 스테이지의 최적 격자점 v_k^* 을 찾는다. 개선 경로 p' 는 각 스테이지의 최적 격자점으로 다음과 같이 구성된다.

$$p' = \langle v_1, v_2^*, \dots, v_{M-1}^*, v_M \rangle \quad (8)$$

S5. 개선 경로의 경유점을 중심으로 다시 격자지도를 구성하여, 일정값 이상의 경로 개선이 없을 때 까지 S1-S4 의 과정을 반복한다.

그림 6 은 위의 알고리즘을 적용하여 경로가 개선되는 과정의 예를 보여준다. 반복횟수가 증가함에 따라 경로가 개선되어 총 이동거리가 단축됨을 알 수 있다.

4. 시뮬레이션 결과

제안한 경로계획 방법의 성능을 평가하기 위한 시뮬레이션을 수행하였다. 알고리즘은 IBM-PC Pentium IV 급 (2.8GHz) MS-Windows XP 상에서 Microsoft Visual C++ 을 사용하여 프로그램으로 구현되었다.

13,000 (mm) x 10,000 (mm) 크기의 평면에 다양한 크기의 장애물 5 - 230 개를 배치하였으며, 비트맵 데이터의 총 화소 수는 $B_x \times B_y = 640 \times 480$ 로 설정하였다. 경로 개선 단계에 사용되는 격자 지도의 단위격자의 크기는 100 (mm) x 100 (mm) 로 설정하였으며, 격자 수 $W \times W$ 는 3 x 3 로 설정하였다. 제안된 방법의 성능을 비교 평가하기 위하여 골격선 그래프와 가시도 그래프를 구현하여 비교하였다.

표 1 은 각 방법으로 생성된 경로의 총 이동거리를 비교하여 보여 준다. 장애물의 수가 다른 다양한 경로 계획문제에 적용하였을 경우 제안 방법이 골격선 방법에 비해 평균 13 % 정도의 경로 단축 효과가 있음을 확인하였다. 가시도 방법은 경로 단축 효과가 매우 뛰어난 방법으로 알려져 있으며, 제안 방법은 가시도 방법에 대비 평균 1.4 % 정도의 총 이동거리만 증가하였다. 즉 제안한 방법이 가시화 방법과 근사한 수준의 경로 단축 효과를 갖고 있음을 알 수 있다.

표 2 는 각 방법으로 경로를 생성하는 데 소요되는 계산시간을 비교하여 보여 준다. 골격선 방법에 비하여 가시도 방법의 계산시간이 장애물의 수에 의거 매우 급격하게 증가함을 볼 수 있다. 이는 장애물 각 꼭지점을 연결하는 간선의 수가 장애물의 수에 따라 기하급수적으로 증가하기 때문이다. 제안 방법의 계산시간은 골격선 방법의 계산시간과 거의 차이가 없음을 볼 수 있다. 이는 골격선 방법에 추가된 경로 개선 단계의 DP 알고리즘이 전

체 계산시간에 별 영향을 주지 않음을 의미한다.

결과적으로 본 시뮬레이션 결과는 제안된 방법이 경로 단축을 측면에서는 가시도 방법에 근사하고, 소요 계산시간 측면에서는 골격선 방법에 근사한 방법임을 입증한다.

5. 결 론

자율이동로봇의 효율적인 경로 계획방법을 제안하였다. 기존 방법은 그래프를 생성 후 경로를 탐색하는 정적 계획 방법이었다. 그러나 제안 방법은 그래프를 생성하여 초기 경로를 탐색 후 이를 개선하는 과정이 포함된 동적 계획 방법이다. 초기 경로 탐색은 골격화 기법과 딥스트라 알고리즘을 이용하여 그래프 생성 및 초기경로를 생성하였으며, 경로 개선은 동적 프로그래밍을 이용하였다. 기존 방법에 비해 짧은 이동거리의 경로를 생성함에 있어서 상대적으로 적은 계산시간이 소요된 것을 확인하였다.

제안한 방법은 복잡한 작업환경과 실시간으로 최단 경로를 생성하는 이동로봇의 경로계획 시스템에 효과적으로 적용될 수 있으리라 기대된다.

[참 고 문 헌]

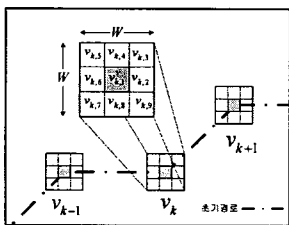
- [1] J. Latombe, Robot Motion Planning, Kluwer academic publishers, 1996.
- [2] E. R. Davies, Machine Vision, 3rd Edition, Morgan Kaufmann publishers, 2004.
- [3] T. H. Cormen, C. E. Leiserson, et al, Introduction to Algorithms, 2nd Ed., The MIT Press, 2001.
- [4] Z. Michalewicz, D.B.Fogel, How to Solve It: Modern Heuristics, Springer-Verlag, 2000.
- [5] 이시웅, 최재각, 남재열, "동적 프로그래밍에 기반한 윤곽선 근사화를 위한 점진 선택 방법," 전자공학회논문지, 제41권, 제3호, pp. 121-127, 2004.
- [6] S. Simhon, G. Dudek, "A Global topological map formed by local metric maps," IEEE/RSJ Int. Conf. On Intelligent Robot and Systems, pp. 1708-1724, 1998.

<표 1> 총 이동 거리 비교.

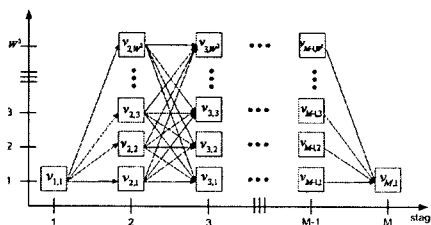
| 장애물 개수 | 총 이동거리 [mm] | | 개선율 [%] | |
|--------|-------------|--------|-----------|-----------|
| | 가시도 방법 | 골격선 방법 | 가시도 방법 대비 | 골격선 방법 대비 |
| 5 | 611 | 767 | -5.07 | 16.30 |
| 10 | 912 | 977 | 2.14 | 8.90 |
| 20 | 1024 | 1236 | -1.27 | 16.10 |
| 35 | 732 | 963 | -4.78 | 20.35 |
| 55 | 1058 | 1199 | 0.19 | 11.93 |
| 80 | 915 | 1114 | -2.62 | 15.71 |
| 110 | 1062 | 1209 | 0 | 12.16 |
| 145 | 900 | 1032 | -1.33 | 11.63 |
| 185 | 1106 | 1239 | -0.36 | 10.41 |
| 230 | 1155 | 1294 | -0.87 | 9.97 |
| 평균 | | | -1.40 | 13.35 |

<표 2> 계산 시간 비교.

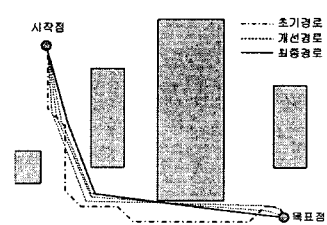
| 장애물 개수 | 계산 시간 [sec] | | |
|--------|-------------|-------|------|
| | 가시도방법 | 골격선방법 | 제안방법 |
| 5 | 0.56 | 0.52 | 0.56 |
| 10 | 1.08 | 0.75 | 0.80 |
| 20 | 8.75 | 1.05 | 1.11 |
| 35 | 22.70 | 1.44 | 1.50 |
| 55 | 46.52 | 2.13 | 2.20 |
| 80 | 89.19 | 2.75 | 2.85 |
| 110 | 150.97 | 3.78 | 3.86 |
| 145 | 275.56 | 4.83 | 4.92 |
| 185 | 418.58 | 5.94 | 6.00 |
| 230 | 594.56 | 7.59 | 7.69 |



<그림 4> 경로 개선을 위한 격자지도.



<그림 5> 동적 프로그래밍을 위한 탐색평면.



<그림 6> 경로 개선 예.