점층적 격자지도를 이용한 안전한 경로 계획1)

최성록, 이재영, 이유철, 박승환, 유원필
한국전자통신연구원 로봇연구부
e-mail: sunglok@etri.re.kr

Generating Safe Path with a Gradation Grid Map

Sunglok Choi, Jae Yeong Lee, Yu-Cheol Lee, Seung-Hwan Park, Wonpil Yu
Robot Research Department, ETRI

요 약

A* 알고리즘을 이용한 경로 계획은 이동 로봇이나 게임 유닛이 주어진 환경에서 장애물과 충돌 없이 이동 가능한 경로를 찾는데 많이 사용된다. 그러나 A* 알고리즘은 출발 지점에서 목적 지점까지의 최단 거리를 갖는 경로를 찾는다. 두 지점 사이에 장애물이 있는 경우, 최단 경로는 보통 두 지점 사이의 장애물의 모서리를 스치는 경로이다. 따라서 로봇이나 게임 유닛이 최단 경로로 이동할 때, 장애물의 모서리에 부딪힐 수 있는 위험성이 커진다. 일반적인 격자 지도의 한 셀은 장애물이 있는 경우의 무한대의 비용과 장애물이 없는 경우의 비용을 갖는다. 본 논문에서는 장애물에 가까울수록 장애물이 없는 경우보다 높은 비용을 갖는 점층적 격자 지도(gradation grid map)를 이용한다. 점층적 격자 지도를 통해 A* 알고리즘은 장애물 근처의 높은 비용을 갖는 셀을 되도록 통과하지 않는 경로, 즉 보다 안전한 경로를 생성한다.

1. 서론

경로 계획은 이동 로봇이나 게임 유닛이 주어진 환경에서 벽 등과 같은 장애물과의 충돌 없는 이동 가능한 경로를 찾는데 사용된다.

경로 계획을 위해 주어진 환경은 보통 그래프로 표현된다. 주어진 환경 내의 장애물들의 모서리를 연결한 가시선 그래프(visibility graph), 보로노이 다이어그램(Voronoi diagram)을 통해 생성된 도로 지도, RRT(Rapidly-exploring Random Trees)에 의해 생성된 트리 지도, 그리고 환경 내 장애물 존재 유무를 이산화하여 표현한 격자 지도가 그 예이다. 그 중 격자 지도는 장애물이 많은 복잡한 환경을 표현하기 쉽고, 또 장애물 정보가 바뀌는 경우 수정하기 쉽기 때문에 환경을 표현하는 방법으로 널리 사용된다. 격자 지도의 장애물이 없는 한 셀은 이웃하는 장애물의 없는 다른 셀들과 연결된 그래프로 볼 수 있다.

그래프로 표현된 환경에서의 경로 계획은 출발 노드에서 도착 노드까지 이동하기 위해 방문하여야 할 노드들을 계산하는 문제, 즉 그래프 탐색 문제가 된다. 그래프 탐색 문제는 초창기 전산학 분야에서 많은 해법이 제시되었는데, A* 알고리즘과 같은 휴리스틱 탐색(heuristic search) 기법은 빠른 시간에 최소 비용의 해를 보장하기 때문에 많이 사용된다. 그러나 실제 주어진 환경은 지형을 가진 3차원 환경이고, 이동하는 또는 지도에 없는 장애물이 있다. 특히 실제 환경에서 구동되는 이동 로봇의 경우 (위치 인식센서의 오차로 인해) 그래프로 표현된 지도에서의 정확한 위치를 얻기 힘들다. 따라서 최단 거리 이동의 경우 그림 2(좌)와 3(좌)와 같이 장애물의 모서리를 스치게 되는데, 실제 한경에서는 가능한 장애물과 부딪힐 위험이 적은 경로를 통해 이동하는 것이 안전하다.

본 논문에서는 그림 1(우)와 같이 점층적 격자 지도를 이용한 A* 알고리즘에서 안전한 경로를 생성하는 방법에 대해 논의한다. 점층적 격자 지도는 장애물에 가까울수록 이동할 때 보다 높은 비용을 갖는 격자 지도이다. 따라서 A*를 이용한 최소 비용 노드 탐색 시 가능한 장애물에 가까운 경로가 생성되지 않는다. Kanehara et al.의 연구[1] 또한 점층적 격자 지도를 통해 안전한 경로를 생성하는데, 이 방법은 일반 격자 지도에서 경로를 생성하고, 점층적 격자 지도는 별도의 후처리(post-processing)에서 사용된다. 그러나 본 논문의 별도의 후처리 과정이 아니라 경로 생성 시 점층적 격자 지도를 사용한다.

2. 점층적 격자지도를 이용한 경로 계획

2.1. A* 알고리즘

A* 알고리즘은 시작 노드 s에서 목표 노드 t까지 최소 비용으로 이동 가능한 노드를 찾기 위해 이동 가능한 노드들 중 가장 최소 비용을 보장할 가능성이 높은 노드를 먼저 탐색한다. 이러한 탐색의 지표로

$$f(n) = c(s,n) + h(n,t) \tag{1}$$

와 같은 평가 함수를 사용한다. 비용 함수 $c(s,n)$은 시작 노드 s에서 평가되는 노드 n까지 이동에 필요한 최소 비용이다. 비용 함수는 일반적으로

1) 본 연구는 지식경제부(MKE)와 한국산업기술평가관리원(KEIT)의 연구 과제 (Hybrid u-Robot Service System Technology Development for Ubiquitous City, 2008-S-031-01)의 지원을 받았습니다.

(그림 1) 실험에 사용된 격자 지도 (좌: 실험에 사용된 환경, 중: 일반 격자 지도, 우: 점층적 격자 지도)

$$c(s,p) = c(s,p) + c(p,n) \tag{2}$$

와 같이 부모 노드 p에서 자식 노드 n으로 이동시 필요한 비용 $c(p,n)$을 누적하여 계산된다. 휴리스틱 함수 $h(n,t)$은 노드 n에서 목표 노드 t까지 이동에 예상되는 비용이다. 잘 알려진 휴리스틱 함수는

$$h(n,t) = \sqrt{(n_x - t_x)^2 + (n_y - t_y)^2} \tag{3}$$

와 같은 유클리디안 거리(Euclidean distance)이다. A* 알고리즘은 큐(queue)에 쌓인 탐색할 노드들 중 최소의 평가 함수를 가지는 노드를 우선으로 탐색한다. 격자 지도에서 A* 알고리즘을 통해 경로를 생성하기 위해서는 로봇이나 게임 유닛의 반지름만큼 그림 1(중)과 같이 장애물을 팽창시킨다. 장애물의 팽창을 통해 로봇이나 게임 유닛을 점, 즉 격자 지도 내의 셀 하나로 여겨지게 되고, 복잡한 장애물 충돌 검사 없이 그래프 탐색을 수행할 수 있게 된다. 그림 1(중)의 농도는 해당 노드로 이동하기 위해 필요한 비용 $c(p,n)$의 배율을 나타낸다 (회색: 1, 검은색: ∞).

2.2. 점층적 격자 지도의 생성

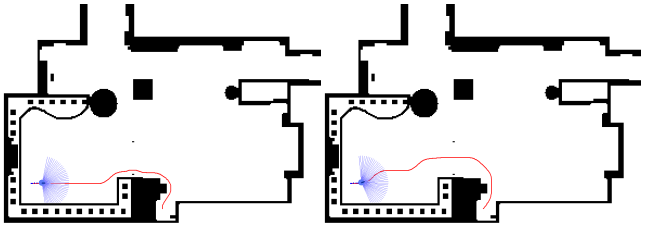
점층적 격자 지도는 그림 1(우)와 같이 장애물에 가까운 경우, 열린 공간에 비해 큰 비용 $c(p,n)$을 갖고, 이를 통해 A* 알고리즘이 가능한 장애물에서 떨어진 경로를 생성하게 한다. 점층적 격자 지도는 그림 1(중)의 격자 지도에서 영상 처리에 사용되는 형태적 연산(morphological operation)의 팽창(dilation)과 유사한 방법으로 생성된다.

```
costWall = ∞ (cost on obstacles)
costFill = the given starting cost
FOR t FROM 1 TO the given thickness (unit: cell)
  costFill = costFill - the given delta
  FOR EACH cell IN the given grid map
    IF GetCost(cell) ≥ costWall THEN CONTINUE
    IF any neighbor of cell has costWall THEN
      SetCost(cell, costFill)
  END FOR EACH
  costWall = costFill
END FOR
```

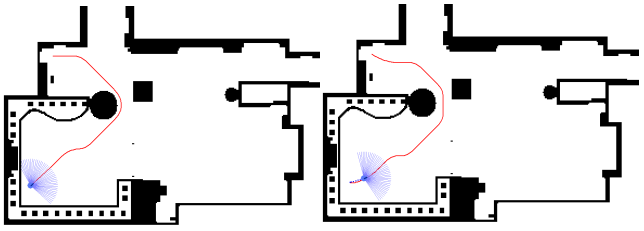
GetCost 함수와 SetCost 함수는 주어진 노드로 이동하기 위해 필요한 비용 $c(p,n)$의 배율을 반환하고 설정하는 함수이다.

3. 실험 및 논의

일반 격자 지도와 점층적 격자 지도에서 동일한 A* 알고리즘을 수행하였다. 그림 2와 그림 3은 생성된 경로의 예이다. 일반 격자 지도에서 생성된 경로의 경우 장애물에 근접한 최단 거리를 생성하였는데, 점층적 격자 지도는 장애물에서 어느 정도 거리를 둔 경로가 생성되었다. 점층적 격자 지도의 점층 두께(thickness), 시작 비용 배율(starting cost)과 감소량(delta)을 바꿈으로서 장애물에 대해 떨어지는 정도를 조절 가능하다.



(그림 2) 경로 계획 결과 #1 (좌: 일반, 우: 점층적)



(그림 3) 경로 계획 결과 #2 (좌: 일반, 우: 점층적)

참고문헌

[1] Kanehara et al., "Path Shortening and Smoothing of Grid-based Path Planning with Consideration of Obstacles", Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2007

[2] Patel, A., "Amit's A* Page", http://theory.stanford.edu/~amitp/GameProgramming/