

셀-분할 방식에 의한 운동계획 알고리즘의 확장

안진영^o 유견아
덕성여자대학교

{jyahn^o, kyeonah}@namhae.duksung.ac.kr

Expansion of Motion Planning Algorithms by Cell-Decomposition

Jinyoung Ahn^o Kyeonah Yu

Dept. of computer science, Duksung Women's University

요 약

사다리꼴-분할 방식은 로봇 경로 계획 알고리즘 중 완전 셀-분할 방식중의 하나로서 장애물과 떨어진 경로를 제공하므로 안정성을 제공하는 방식이다. 사다리꼴-분할 방식은 다각형 환경으로 이루어진 형상공간에서 정의되며 자유공간을 볼록 다각형으로 이루어진 셀(cell)로 나누어 로봇 운동을 계획하는데, 원과 같은 비다각형 장애물이 존재하는 경우에 대해서는 이 성질을 만족하는 분할 방법이 알려져 있지 않다. 본 논문에서는 기존의 다각형 환경에서 정의된 사다리꼴-분할 방식을 분할의 완전성을 잃지 않고 원의 호를 포함하는 환경으로 확장하는 알고리즘을 소개하고 구현한다.

고 원의 호를 처리하기 위해 어떻게 수정되는지 설명한다.

1. 서 론

셀 분할 방식은 로봇 운동 계획분야에서 가장 광범위하게 연구되는 분야중의 하나로서 로봇의 자유공간을 경로가 쉽게 계획될 수 있는 단순 영역(셀)들로 나누어 전체 운동 경로를 계획하는 방식이다.[1] 나누어진 셀을 기반으로 이웃한 셀들을 연결하는 연결 그래프(connectivity graph)를 생성한 후 최적경로를 탐색하는데, 셀을 나누는 방식에 따라 근사(approximate) 셀 분할 방식과 완전(exact) 셀 분할 방식으로 나눌 수 있다. 완전 셀 분할 방식은 분할한 셀들의 합집합이 정확히 전체 자유공간이 되는 조건을 만족해야 하는 방식으로 셀들의 합집합이 전체 자유공간의 부분집합이 되는 근사 방식에 비해 조건이 엄격하다고 할 수 있다. 이러한 방식의 대표적인 예로 사다리꼴-분할(trapezoidal decomposition) 방식이 있다. 사다리꼴-분할 알고리즘은 로봇 운동계획을 위한 대부분의 알고리즘들처럼 형상공간(configuration space, C-공간)을 기반으로 하며 C-공간에서의 자유공간을 볼록 다각형으로 이루어진 셀(cell)로 나누어 로봇 운동을 계획한다. 이 방법의 전제 조건은 작업 환경에 있는 C-공간 장애물들이 모두 다각형이어야 하는 것으로 비다각형 C-공간 장애물이 있는 경우, 기존의 방식으로 셀을 나눈다면 셀이 볼록 도형이어야 하는 조건을 만족시킬 수 없다. 본 논문에서는 기존의 다각형 환경에 원의 호를 포함한 일반화 다각형 환경으로 이 알고리즘을 확장하고자 한다. 논문[2]에서 설명한 원의 성질과 장애물로 부터 일정한 거리만큼 떨어진 경로를 보장한다는 성질을 이용하면 일반화 다각형 장애물 환경을 셀로 나누고, 이 셀을 기반으로 다각형 환경에서와 같은 연결 그래프를 생성할 수 있다. 확장된 알고리즘은 기존의 사다리꼴 분할 방식의 완전성(exactness)을 유지함을 보여준다.

2. 사다리꼴-분할 방식의 수정

이 장에서는 기존의 사다리꼴-분할 방식에 대해 살펴보

2.1 사다리꼴-분할 방식

사다리꼴-분할 방식은 C-공간에서의 자유공간(C_{free}) 공간을 셀로 나누어 셀을 노드로 하고 인접한 셀을 연결한 선분을 링크로 하는 무방향(non-directed) 그래프를 정의한다. 사다리꼴-분할 방식으로 생성되는 그래프는 장애물과 어느 정도 거리를 갖는 안정성 있는 경로를 제공해 준다.

사다리꼴-분할 방식을 이용해 그래프를 생성하기 위해서는 두 단계를 거쳐야 한다. 첫 번째 단계에서는 y축에 평행하는 임의의 선분 L(sweep-line)을 x축으로 스위핑하면서 C_{free} 를 수직으로 분할할 셀들을 찾아낸다. 장애물의 꼭지점을 x 좌표값으로 정렬해 놓고 L이 x와 만나면 이 점을 기준으로 장애물과 교차하지 않는 L에 평행한 선분을 그어 C_{free} 상의 셀을 나눈다. 이렇게 나누어진 셀은 삼각형 또는 사다리꼴 형태, 즉 볼록 다각형 형태로 정의된다. 이때 임의의 두 셀의 내부는 교차하지 않아야 하고 C-공간상의 모든 셀의 합집합은 전체 C-공간의 C_{free} 이어야 한다. 두 번째 단계에서 노드와 링크를 정의하여 그래프를 생성한다. 그래프의 노드는 셀로 정의되며 링크는 인접한 셀의 노드들을 연결한 선분이 된다. 임의의 두 셀이 수직선분을 공유한다면 이는 인접한 셀이다. 생성된 그래프를 기반으로 경로를 탐색하기 위해 여러 탐색 알고리즘들이 알려져 있는데 그 중, A* 알고리즘은 과소추정된 휴리스틱(underestimated heuristic)을 이용하면 그래프의 최단 경로를 찾는 것을 보장한다는 장점이 있다. 특히 휴리스틱 값으로 유클리드 거리를 취하면 A* 알고리즘의 시간복잡도가 $O(n \log n)$ 까지 개선될 수 있으므로 많은 경우에 A* 알고리즘을 이용한다. 사다리꼴-분할 방식에 의한 그래프에서 탐색 알고리즘을 통한 최단 경로는 실제 거리상의 최단

거리는 아니지만 장애물로부터 떨어진 점을 노드로 정하기 때문에 로봇이 움직이는 경로로서 안정성을 제공한다.

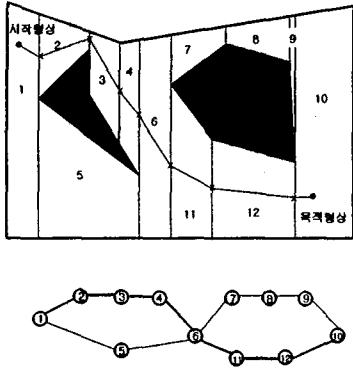


그림 1. 사다리꼴-분할 방식의 예

2.2 사다리꼴-분할 방식의 확장

사다리꼴-분할방식은 C_{free} 공간을 임의의 셀로 나누기 위해 y축에 평행한 수직선(sweep-line)을 x 방향으로 스위핑한다. 이때, 꼭지점들을 x 좌표 값으로 정렬하는데 원의 호를 포함하여 정렬하기 위해서는 먼저 원을 단순 증가 또는 감소하는 성질을 갖도록 분할한다. 이를 위해 논문 [2]에서 소개한 바와 같이 L에 평행, 수직하는 방향으로 각각 2개씩, 4개의 가상점을 찾아 가상점을 포함한 꼭지점들을 기준으로 셀을 나눈다.

그러나 이러한 가상점을 포함하여 나누더라도 분할된 셀은 연결 그래프를 생성하기 위한 셀의 성질에 어긋난다. 즉, 셀이 삼각형이나 사다리꼴이 아닌 오목 도형의 경우가 발생한다. 볼록 다각형으로 이루어진 셀이 아닌 경우 그림 2(a)와 같이 C-공간의 경계선(boundary)의 형태에 따라 경로가 반드시 C_{free} 상에 있음을 보장하지 못하는 경우가 발생한다. 즉, 사다리꼴 분할 방식에서는 경로를 셀의 경계선의 1/2점을 연결하여 생성하므로, 비록 원의 호가 단순 증가 혹은 감소하는 성질을 가지고 있다 해도 생성된 경로가 장애물과 겹치는 경우가 발생하는 것이다.

이를 해결하기 위해 셀이 오목 도형인 경우, 그림 2(b)에서와 같이 C_{free} 상에 존재하지 않는 선분을 생성한 두 수직선으로부터 현재 처리되고 있는 셀의 호에 접선을 구하여 두 접선이 만나는 교점을 기준으로 셀을 나누어 처리한다. 추가된 노드(그림 2(c))는 호의 접선을 기준으로 나누어졌기 때문에 오목한 부분에 대해 장애물과 떨어진 안정성있는 경로의 제공을 보장하고 셀은 두 개의 셀로 나뉜다.

이때 셀을 이루고 있는 호와 마주보는 선분과 호 사이의 거리가 아주 작아지면 교점이 장애물의 내부에 생기는 경우(그림 2(d))가 생기는데, 이러한 경우 호에 대한 두 점선의 접점을 기준으로 셀을 세 부분으로 나누어 처리한다. 이때 나누어진 셀의 양 끝 셀(그림 2(e).a)은 기존의 수직선의 1/2점으로부터의 생긴 접선을 기준으로 나누어졌기 때문에 1/2점을 연결한 선분이 호와 교차하지 않을 것이 보장된다. 즉, 장애물에서 떨어진 안정된 경로를 보장한다. 그러나 가운데 셀은 재귀적으로 오목 셀 처리 과정을 거쳐야 한다. 이러한 과정이 반복되면 셀의 폭이 점점 감소하며, 선분과 호 사이에 자유공간이 존재하는 한 1/2점을 연결한 선이 호와 교차하지 않는 순간이 생긴다.

림 2(e).a)은 기존의 수직선의 1/2점으로부터의 생긴 접선을 기준으로 나누어졌기 때문에 1/2점을 연결한 선분이 호와 교차하지 않을 것이 보장된다. 즉, 장애물에서 떨어진 안정된 경로를 보장한다. 그러나 가운데 셀은 재귀적으로 오목 셀 처리 과정을 거쳐야 한다. 이러한 과정이 반복되면 셀의 폭이 점점 감소하며, 선분과 호 사이에 자유공간이 존재하는 한 1/2점을 연결한 선이 호와 교차하지 않는 순간이 생긴다.

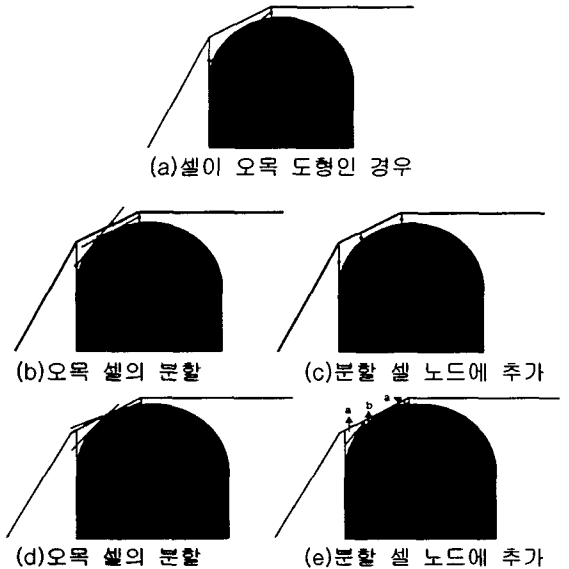


그림 2. 오목 셀 처리 방법

이렇게 확장된 일반화 사다리꼴-방식 알고리즘으로 경로를 계획한 예는 그림 3과 같다.

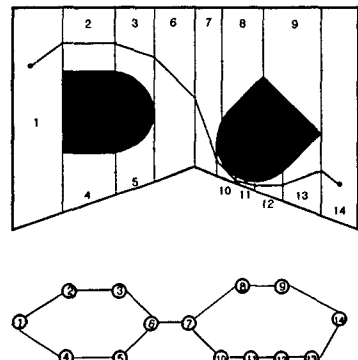


그림 3. 사다리꼴-분할 방식의 확장

3. 구현 및 결과

수정된 알고리즘은 Pentium III 플랫폼에서 Visual C++를 이용하여 구현하였다.

3.1 알고리즘의 구현

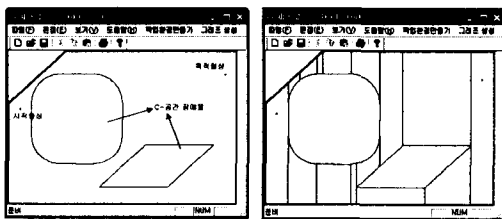
사다리꼴-분할 알고리즘을 이용하여 일반화다각형 환경으로 확장된 그래프를 구현하기 위해서는 두 경우에 대한 모서리 처리가 필요하다.

① 장애물의 꼭지점 : 모든 장애물을 포함하는 C-공간상의 테두리를 포함하는 장애물의 꼭지점의 y 값에 ϵ_y 값을 더한 점과 뺀 점이 그 장애물의 외부에 존재하는 경우 [3]에, 꼭지점으로부터 외부 방향으로 수직선을 그어 셀을 나눈다. 만약 두 점 모두 외부에 존재하면 양방향으로 수직선이 생긴다. 이때 셀을 나누기 위한 수직선은 다른 장애물과 교차하지 않아야 하므로 다른 장애물과의 교차여부를 판별하여 그 교차점이 수직선의 끝점이 된다.

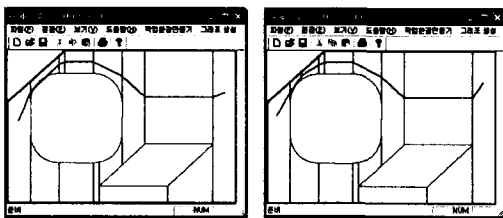
② 호 : 호 위의 가상점과 꼭지점에서는 ①과 동일한 방법으로 셀을 나누어준다. 이러한 방법으로 생성된 셀 상에 경로가 C_{free} 상에 존재하는지의 여부는 일반화 다각형을 모서리로 표현한 자료구조와 2차 방정식의 판별식을 이용하여 판별할 수 있다. 이때 C_{free} 상에 존재하지 않는 경우, 셀을 이루는 양끝의 두 수직선으로부터 처리 되고 있는 셀 내부로 호에 대한 점선을 구하여 두 점선의 교점을 기준으로 두개의 셀로 나누어 노드를 추가한다. 이때 교점이 장애물의 내부에 생기면 호에 대한 두 점선의 접점을 기준으로 하나의 셀을 세 개의 셀로 나누고, 나누어진 셀 중 가운데 위치한 셀을 ②의 방법을 기준으로 재귀적(recursive)으로 처리한다.

①, ②에 의해 노드와 인접한 셀에 대한 링크가 정의 되면 시작형상을 포함하는 셀을 시작점으로, 목표형상을 포함하는 셀을 끝점으로 하는 그래프를 생성한 후 유클리드 거리를 휴리스틱 값으로 갖는 A* 알고리즘을 이용해 최단경로를 탐색한다.

3.2 실행 결과



(a) C-공간 (b) 셀



(c) 오목 셀 처리 전 경로 (d) 오목 셀 처리 후 경로
그림 4. 실행 결과

그림 4는 일반화 다각형 장애물이 있는 C-공간 환경에서 사다리꼴-분할 방식을 이용해 그래프를 생성하고 A*알고리즘을 적용하여 최적의 경로를 찾은 예를 보여준다. 그림 4(a)는 C-공간에서 C-공간 장애물과 점으로 환산된 로봇의 시작형상과 목적 형상을 보여준다. 이러한 공간을 기준으로 그림 4(b)는 C_{free} 공간을 셀로 분할하는 것을 보여주는데 이때 분할된 셀이 오목 도형인 경우 셀을 두 부분 또는 세 부분이 상으로 나누어 처리하는데, 이때 두 부분으로 나누어 처리하였다. 그림 4(c)는 오목 셀을 처리하지 않고 사다리꼴-분할 방식을 적용한 예로서 그래프 탐색에 의한 경로가 장애물과 겹치는 경우를 보여주며 그림 4(d)는 오목 셀을 분할 처리 한 후의 탐색 경로인데 (c)에서 나타났던 문제가 해결되었음을 보여준다.

4. 결론

본 논문에서는 로봇의 운동계획 알고리즘 중 대표적인 사다리꼴-분할 알고리즘을 일반화 다각형 환경으로 확장하고 구현하였다. 이는 일반화 다각형으로 이루어진 환경에서 움직이는 로봇이 안정성있는 최적경로를 찾아 움직일 수 있게 한다. 좀 더 다양한 환경에서 로봇이 경로계획을 할 수 있도록 하는데 한 걸음 나아간 것이라고 할 수 있다.

호를 스윙핑 선인 L에 대해 수직과 평행을 기준으로 가상점을 첨가하고 이 가상점과 호의 꼭지점으로 이루어진 셀을 경우에 따라 임의의 두개 또는 세개 이상의 셀로 나누어 처리하여 알고리즘이 완전 셀 분할 성질을 유지하면서 일반화다각형 환경으로 확장할 수 있었다. 비록 오목 셀을 생성하여 일반화 다각형으로의 확장이 힘들어 보였으나 기존의 방식에 대한 정의와 성질을 기반으로 오목 셀에서도 볼록 셀에서와 같은 결과를 유도할 수 있었다. 이는 일반화 다각형으로 확장에 대한 사다리꼴-분할 방식에 국한된 한정적 제안으로 앞으로는 임의의 방식에 적용할 수 있고 일반화된 성질로 확장될 수 있는 원리를 위해 일반화 다각형이나 원에 대한 더 많은 연구가 필요하고 활성화되어야 한다 [4].

[참고문헌]

[1] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, pp.153-199, 1991
 [2] 유건아 안진영, 일반화 다각형을 위한 plane-sweep 알고리즘의 구현, *한국정보과학회 봄 학술발표논문집*, pp.691-693, 2002
 [3] J. O'Rourke, "Computational Geometry in C", Cambridge Press, 1998
 [4] J.C. Latombe. Motion planning: A journey of robots, molecules, digital actors, and other artifacts. *International Journal of Robotics Research*, 18(11):pp.1119--1128, 1999.