Proceedings of the 12th KACC, October 1997

일반적인 다면체 사이의 최소거리 계산을 위한 효율적인 알고리즘의 계산

Development of an efficient algorithm for the minimum distance calculation between general polyhedra

°임준근*, 오재윤*, 김창회**, 김기호**, 김승호**

* 전북대학교 기계공학부(Tel: +82-652-70-2377;Fax:+82-652-70-2388;E-mail:ohcy@moak.chonbuk.ac.kr)

** 원자력연구소 로봇시스템개발팀(Tel:+82-42-868-2930;Fax:+82-42-868-8833;E-mail:chkim3@nanum.kaeri.re.kr)

Abstracts This paper developes an efficient algorithm for the minimum distance calculation between general polyhedra(convex and/or concave). The polyhedron approximates an object using flat polygons which composed of more than three vertices. The algorithm developed in this paper basically computes minimun distance betwen two convex polygons and finds a set of polygons which makes a global minimum distance. The advantage of the algorithm is that the global minimum distance can be computed in any cases. But the big disadvantage is that the minimum distance computing time is rapidly increased with the number of polygons which used to approximate an object. This paper developes a method to eliminate unnecessary sets of polygons, and an efficient algorithm to compute a minimum distance between two polygons in order to compensate the inherent disadvantage of the algorithm. It takes only a few times iteration to find minimum distance for most polygons. The correctness of the algorithm are visually tested with a line which connects two points making a global minimum distance of simple convex object(box) and concave object(pipe). The algorithm can find minimum distance between two convex objects made of about 200 polygons respectively less than a second computing time.

Keywords Polygon, Minimum distance, Normal vector, Convex and concave object

1. 서론

로봇을 이용하여 어떠한 작업을 수행하기 위해서는 on-line으로 혹은 off-line으로 수행하고자 하는 작업을 위한 경로를 계획하고, 계획된 경로를 모의실험을 통해 확인하는 과정이 필요하다. On-line 혹은 off-line으로 로봇 작업 경로 모의실험을 실시하는데 로봇과 작업환경 사이의 상호관계를 이해하는 것은 매우 중요하다. 여기서 말하는 상호관계 중에는 로봇과 주위환경이 이루는 최소거리를 포함한다.

모의실험은 실제 로봇과 주변환경을 근사한 상황에서 이루어지므로 모의실험시에는 충돌을 하지 않으나 로봇과 주위환경과 사이에 거리가 너무 가까우면 실제 로봇 운용 중에는 충돌을 일으킬 수 있다. 따라서 최소거리가 너무 작지 않게 경로 계획을 하기 위해서는 로봇과 주위환경과 사이에 거리정보가 필요하나, 현재 로봇 경로 모의실험에 많이 사용되는 상업용 소프트웨어는 대부분 충돌여부에 대한 정보만 제공할 뿐 로봇이 주위환경과 충돌하지 않는 경우에는 둘 사이의 거리와 같은 정보는 제공하지 않는다.

컴퓨터 그래픽에서 널리 사용되는 3-D 기하학적 모델링(geometric modeling) 방법은 크게 세 가지로 분류 할 수 있는데[1-5], 일반적으로 로봇과 작업환경 등을 근사하는데 이 중 면(polygon)을 이용하여 물체를 근사하는 방법은 다른 방법들에 비해 상대적으로 구현하기가 용이하므로 많이 사용된다. 면에 의해 근사된 물체는 convex 물체와 concave 물체로 구분된다. Convex 물체는 임의의 직선과 항상 둘 이하의 교차점을 가지는 경우이고 concave한 물체는 임의의 직선과 셋 이상의 교차점을 가질 수 있는 경우로 정의 될 수 있다.

면에 의해 근사 되어진 convex 물체와 convex 물체 사이에서 최소거리를 찾는 것은 비교적 용이하고, 임의의 시작점으로부터 최소거리를 이루는 점으로의 접근이 보장되므로 많은 효율성을 강조한 알고리즘들이 개발되었다[6-10]. Bobrow[7]은 찾고자 하는 최소거리를 이루는 점과 아주 가까운 최초 시작점을 찾는 방법을 통하여 알고리즘의 효율성을 증가시키는 방안을 고안하였다. Hurteau and Stewart[8]은 가장 작은 반복횟수를 통해 최소거리를 이루는 점으로 찾아가는 방안을 제시하였다. Smith[9]는 그래픽 하드웨어를 이용하여 효율적으로 3차원 물체들 사이의 최소거리 계산 방법을 개발하였다. Convex 물체와 convex 물체 사이에서 최소거리는 적절하게 선택된 시작점을 이용하면 최소거리를 이루는 점으로의 접근이 보장되므로 최소거리를 찾는 것이 비교적 용이하다 할 수 있다. 그러나 두 물체중 하나라도 concave 물체이면 전역(global) 최소거리를 이루는 점으로 접근

을 보장하는 시작점을 구하는 것이 용이하지 않다. 또한 시작점이 적절하게 선택되었다 하더라도 두 물체 사이의 상대 위치에 따라 전역 최소거리를 이루는 점으로 접근을 보장하지 않는다. 그러므로 convex 물체를 포함하는 환경에서 효율적으로 전역 최소거리를 계산하는데 이용된 알고리즘을 concave 물체를 포함하는 환경에 적용할 수 없다. 따라서 concave 물체를 포함하는 환경에서 최소거리를 찾는 이해하기 쉬우며, 수학적으로 복잡한 이론을 가지지 않으며, 효율적인 알고리즘은 상대적으로 많이 보고되지 않았다.

본 논문에서는 면에 의해 근사된 convex 물체와 concave 물체를 포함하는 일반적인 작업환경에서 물체들 사이의 최소거리 정보를 제공할 수 있는 수학적으로 복잡한 이론을 가지지 않는 매우 효율적인 알고리즘을 개발한다.

2. 두 직선 사이의 최소거리 계산

3차원 공간에서 무한한 길이를 갖는 임의의 두 직선사이의 최소거리는 두 직선에 공통으로 수직인 직선사이의 거리이다[2]. 이 경우는 다음과 같은 두 식을 동시에 만족하는 두 점을 찾으면 이 두 점 사이의 거리가 두 직선사이의 최소거리가 된다.

$$(\vec{p} - \vec{q}) \cdot \overrightarrow{\text{직선 } 1} = 0$$

$$(\vec{p} - \vec{q}) \cdot \overrightarrow{\text{직선 } 2} = 0$$

그러나 유한한 길이를 갖는 두 직선 사이에서는 공통법선이 두 직선사이에서 존재할 수도 존재하지 않을 수도 있기 때문에 최소거리를 이루는 점을 찾기는 그리 쉽지가 않은 문제이다[2,11].

본 논문에서는 두 면사이의 최소거리 계산을 하는데 뿐 아니라 두 물체사이의 최소거리 계산에 유한한 길이를 갖는 두 직선사이의 최소거리를 이루는 점들을 이용한다. 본 논문에서는 참고문헌[11]에서 개발한 알고리즘을 이용하여 3차원 공간에서 유한한 크기를 가지는 두 직선사이의 최소거리를 이루는 점들을 찾는다.

3. 두 면 사이의 최소거리 계산

3차원 공간에서 정의되는 무한한 크기를 갖는 두 면 사이의 최소거리는 두 면에 공통으로 수직인 직선사이의 거리이다[2]. 이 경우는 다음과 같은 두 식을 동시에 만족하는 두 점을 찾으면 이 두 점 사이의 거리가 두 면사이의 최소거리가 된다.

$$(\vec{p} - \vec{q}) \cdot \overrightarrow{\text{면 } 1\text{의 법선벡터}} = 0$$

$$(\vec{p} - \vec{q}) \cdot \overrightarrow{\text{면 } 2\text{의 법선벡터}} = 0$$

그러나 유한한 크기를 갖는 두 면 사이에서는 공통법선이 두 면 사이에서 존재할 수도 존재하지 않을 수도 있기 때문에 최소거리를 이루는 점을 찾기는 그리 쉽지가 않은 문제이다[2,6]. 본 논문에서는 다음과 같은 알고리즘을 개발하여 유한한 크기를 갖는 두 면 사이에서 최소거리를 이루는 두 점을 찾는다.

Step 1. 각 면의 중심점을 참조점으로 한다.

Step 2. search direction의 계산

(1) 두 참조점이 모두 면 안에 있는 경우

(i) 각 면에서 벡터 $\vec{v}$ 계산

$$\vec{v} = \overrightarrow{object\ i\ \text{참조점}} - \overrightarrow{object\ j\ \text{참조점}}$$

* 상대면 에서는 $-\vec{v}$ 가 된다.

(ii) 각 면에서 search direction 계산

$$\vec{s} = \vec{v} - (\vec{v} \cdot \vec{n})\,\vec{n}$$

* $\vec{n}$: 면의 법선 벡터

(iii) 각 면에서, 현 참조점들을 지나는 search direction 벡터 $\vec{s}$ 방향의 직선들과 면의 모서리와의 교점을 구해서, 이 교점들의 연결선을 search direction 으로 한다.

(iv) 만약, 두 면에서 $\|\vec{s}\|^2 = 0$ 이면, Step 4로 간다.

(2) 한 참조점이 모서리에 있고 다른 참조점은 면 안에 있는 경우

(i) 참조점이 모서리에 놓이는 면에서는 이 모서리를 search direction으로 한다.

(ii) 상대면에서 search direction

a) (i)의 참조점이 놓이는 모서리의 양 끝점 중 상대면의 참조점과 가까운 점을 찾는다.

b) 이 점과 참조점이 놓이는 모서리를 상대면상에 투영한다.

c) 이 같이 상대면에 투영된 점을 지나는 투영된 모서리 방향의 직선과 상대면의 모서리들과 교점을 구하여, 이 점들을 연결하면 상대면의 search direction으로 한다.

(iii) 두 search direction 사이에서 최소거리를 계산하고, Step 4로 간다.

(3) 두 참조점이 모두 모서리에 있는 경우

(i) 참조점이 놓이는 각 모서리를 각 면에서 search direction으로 한다.

(ii) 두 search direction 사이에서 최소거리를 계산하고, Step 4로 간다.

Step 3. 두 search direction 사이에서 최소거리를 이루는 두 점을 계산해서, 이 두 점을 다음 반복 계산을 위한 참조점으로 하고, Step 2로 돌아간다.

Step 4. 위에서 구한 두 참조점 사이의 최소거리를 구하면, 이 거리가 3차원 공간상에서 현재 고려중인 두 면사이의 최소거리가 된다.

4. 두 물체 사이의 최소거리 계산

본 논문에 이용되는 모든 물체는 동일 직선 상에 놓이지 않는 세 개이상의 꼭지점을 반 시계방향으로 연결하여 정의되는 convex 면을 이용하여 근사된다. 세 개이상의 꼭지점이 동일 직선 상에 놓이지 않는다는 것은 각 면이 면적을 가지며, 그 면 표면으로부터 밖을 향하는 법선 벡터를 정의 할 수 있음을 의미한다. 그리고 실제 물체는 표면에 의해 둘러싸이므로 면들이 서

로 겹치지 않고, 면이 한 군데도 빠지지 않게 나타낸다.

본 논문에서 개발하는 3차원 공간에서 일반적인 최소거리 계산을 위한 알고리즘은 면과 면 사이의 국부 최소거리(local minimum)를 계산하여 이 들의 조합 중에서 전역 최소거리(global minimum)를 이루는 조합을 찾는 것이다.

면과 면 사이의 최소거리를 계산하는 알고리즘의 장점은 물체의 속성에 관계없이 모든 경우에 확실하게 물체와 물체 사이의 최소거리를 계산할 수 있다는 것이다. 어느 물체를 면을 이용해서 좀 더 정확하게 근사하고자 한다면, 좀 더 작고 좀 더 많은 면이 필요하다. 면 수가 많아지면 면 사이의 조합은 이의 자승에 비례하여 증가하므로 전역 최소거리 계산시간이 급격히 증가한다는 단점을 가지고 있다. 본 논문에서는 이와 같은 면과 면 사이의 최소거리를 계산하는 알고리즘의 단점을 최소화하기 위하여 불필요한 면과 면 사이의 최소거리 계산을 제거하는 방법을 고안하여 알고리즘의 효율성을 향상시키고자 하였다.

불필요한 면과 면 사이의 최소거리 계산을 제거하는 방법은 크게 다음과 같이 두 경우로 나누어 생각 할 수 있다.

(1) 면과 면 사이의 거리가 충분히 멀어서 최소거리 계산이 의미가 없는 경우 : 두 면 중심 사이의 거리가 각 면의 중심에서 가장 먼 꼭지점까지 거리를 합한 값에 프로그램 사용자가 설정하는 적절한 거리(tolerance)를 더한 거리 보다 작은 경우에만 두 면 사이의 최소거리 계산을 수행한다.

(2) 면과 면 사이의 관계로부터 전역 최소거리 발생 가능성이 없는 경우: 두 물체 사이의 전역 최소거리는 분명히 서로 마주보고 있는 면 사이에서 발생한다. 이를 수학적으로 기술하면, 두 면 법선 벡터 사이의 내적(dot product)이 음수(-)이고, 기준면의 중심점에서 상대면의 중심점을 향하는 벡터와 기준면 법선 벡터 사이의 내적이 양수(+)인 경우에만 두 물체사이의 최소거리가 발생될 수 있다.

예를 들어 Fig. 1에서 n_2면과 n_3면의 조합과 n_4면과 n_8면의 조합은 둘 다 법선 벡터 사이의 내적이 음(-)이다. 그러나 n_2면과 n_3면의 조합에서는 법선 벡터 $\vec{n_2}$와 벡터 $\vec{v_1}$의 내적이 양(+)이나, n_4면과 n_8면의 조합에서는 법선 벡터 $\vec{n_8}$와 벡터 $\vec{v_4}$와 내적이 음(-)이다. 그러므로 n_2면과 n_3면의 조합에서는 최소거리가 발생할 수 있으나, n_4면과 n_8면의 조합에서는 최소거리 발생 가능성이 없으므로 계산이 필요 없는 두 면의 조합이다.

위 (1)이나 (2)에 해당되지 않는 두 면의 조합에 대해서만 최소거리 계산을 수행하여 알고리즘의 효율성을 향상시켰으며, 다음은 본 논문에서 개발한 일반적인 두 물체 사이의 최소거리 계산을 위한 알고리즘이다.

Step 1. set i = 1 (for object 1)
j = 1 (for object 2)
global minimum distance = positive big number

Step 2. 현재 고려하고 있는 두 면 법선 벡터의 내적이
if 음수(-)이면,
Step 3 으로 가고,
else if 양수(+)이면,
Step 8 (ii) 로 간다.

Step 3. 각 면의 중심점을 참조점으로 한다.

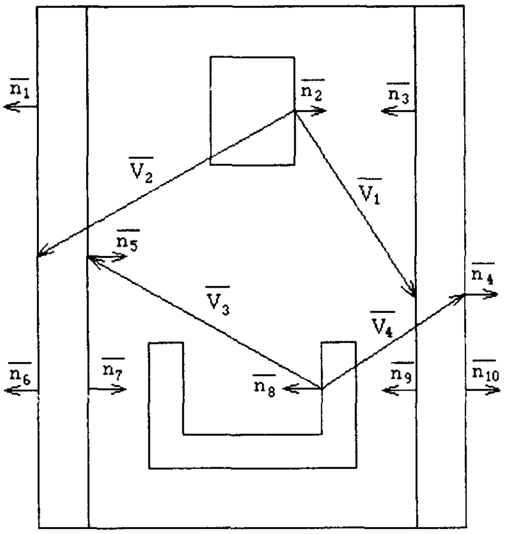

그림 1. 법선 벡터와 참고 벡터

Fig. 1 Normal vectors and reference vectors

Step 4 $\vec{v}$ = $\overrightarrow{object\ 1\ \text{참조점}}$ − $\overrightarrow{object\ 2\ \text{참조점}}$

Step 5. 각 면의 법선 벡터와 상대면으로의 $\vec{v}$의 내적이
if 양수(+) 이면,
Step 6 으로 가고,
else if 음수(-)이면,
Step 8 (ii) 로 간다.

Step 6. 두 면사이의 최소거리 계산 필요성 여부 검사
tmp_dist = (두 면 참조점 사이의 거리) - (두 면 참조점과 각 꼭지점 사이의 최대거리 합)
if (tmp_dist) < (tolerance) 이면,
Step 7 로 가고,
else if (tmp_dist) ≥ (tolerance) 이면,
Step 8 (ii) 로 간다.

Step 7. 두 면에서 최소거리 계산

Step 8.
(i) update global minimum distance
(ii) update 면 number i and/or j
(1) if (j < m) 이면 j = j + 1 로 갱신하고
Step 2 로 간다.
(2) else if (j = m) 이고
if (i < n) 이면 j = 1
i = i + 1 로 갱신하고
Step 2 로 간다.
else if (i = n) 이면
Step 10 으로 간다.

Step 9. 두 물체 사이에 상대 위치가 변하면 위의 Step 1에서 Step 8을 수행하여 현 두 물체의 상대위치에서 이루는 최소거리를 계산한다.

본 논문에서 개발한 알고리즘의 정확성과 효율성을 검증하기 위하여 몇 개의 convex 물체와 concave 물체를 만들어서 테스트하였다. 먼저 본 알고리즘의 정확성을 검증하기 위하여 (box (convex) vs. box)와 (box vs. pipe(concave))의 두 경우를 고려하였다. 각 경우 두 물체의 상대적 위치를 바꿔가면서 전역 최소거리를 이루는 두 점을 연결하여 시각적으로 정확성을 검증하였다. 그리고 면과 면 사이에서 최소거리는 대부분 대략 2-3회 정도 반복 안에 구해졌다. Convex 물체와 concave 물체의 경우에는 상대 위치에 따라 최소거리 계산을 필요로 하는 면과 면의 조합이 많아지는 이유로 인하여 다소 계산시간이 길어지나, convex 물체와 convex 물체의 경우에는 서로 마주보는 면과 면의 조합만이 고려됨으로 인하여 각각 대략 200개 면으로 구성된 두 convex 물체의 경우에도 1초안에 최소거리를 계산할 수 있는 효율성을 보였다.

5. 결 론

본 논문은 세 개이상의 꼭지점으로 구성되는 면(flat polygon)을 이용하여 나타낸 다면체(convex and/or concave polyhedron) 사이의 최소거리를 계산하기 위한 효율적인 알고리즘을 개발하였다.본 논문에서 개발하는 알고리즘은 기본적으로 면과 면사이의 최소거리를 계산하여 두 물체 사이에서 제일 작은 거리를 이루는 두 면의 조합을 찾는다. 모든 면과 면 조합에서 최소거리를 계산하는 알고리즘은 어떠한 경우에서도 확실하게 두 물체 사이의 최소거리를 계산할 수 있다는 장점이 있다. 그러나 물체를 좀더 정확하게 근사하기 위해 면의 수가 증가함에 따라 계산시간이 급격하게 증가한다는 단점이 있다. 본 논문에서는 이와 같은 단점을 최대한 보완하기 위하여 다면체 사이에서 최소거리 발생 가능성이 있는 두 면을 찾기 위한 방법을 고안하였고, 두 면 사이에서 최소거리 계산을 위한 효율적인 알고리즘을 개발하였다. 본 논문에서 개발되는 알고리즘의 정확성은 단순한 convex 물체와 concave 물체의 상대위치를 바꾸어 가면서 최소거리를 이루는 두 점을 연결하여 시각적으로 검증하였다. 그리고 Convex 물체와 concave 물체의 경우에는 상대 위치에 따라 최소거리 계산을 필요로 하는 면과 면의 조합이 많아지는 이유로 인하여 다소 계산시간이 길어지나, 각각 대략 200개 면으로 구성된 두 convex 물체 사이의 최소거리를 1초안에 계산하는 효율성을 보였다.

참고문헌

[1] J. D. Foley., A. Van Dam., S. K. Feiner, J. F. Hughes, and R. L. Phillips. "Introduction to Computer Graphics". Addison -Wesley Publishing Company, 1994.

[2] M. E. Mortenson, "Geometric modeling". John Wiley & sons, 1985.

[3] V..B. Anand., "Computer Graphics and Geometric Modeling for Engineers". John Wiley & sons, 1993.

[4] W. Rodriguez, "The Modeling of Design Ideas". McGraw -Hill, 1992.

[5] A. A. G. Requicha, "Representations for Rigid Solids: Theory, Methods, and Systems". Computing Surveys 12 (4) (December 1980): pp. 437-464.

[6] J. T. Schwartz, "Finding the minimum distance between two convex polygons". Information Processing Letters. Vol. 13, No. 4, pp. 7-9

[7] J. E. Bobrow, "A direct minimization approach for obtaining the distance between convex polyhedra". The International Journal of Robotics Research. Vol. 8, No. 3, (June 1989) pp. 65-76

[8] G. Hurteau and N. F. Stewart, "Distance calculation for imminent collision indication in a robot system simulation". Robotica Vol. 6 (1988): pp. 47-51

[9] R. C. Smith, "Fast robot collision detection using graphics hardware". IFAC Robot Control(Syroco '85). Barcelona. Spain. (1985) pp. 277-282

[10] M. C. Lin. " Efficient Collision Detection for Animation and Robotics". Ph.D. dissertation, U. of California at Berkeley, Berkeley, California, 1993.

[11] V. J. Lumelsky, "On fast computation of distance between line segments". Information Processing Letters 21 (1985) pp. 55-61