논문
한국정밀공학회지 제15권 제11호 (1998년 11월)
Journal of the Korean Society of Precision Engineering, Vol. 15, No. 11, November 1998.

삼차원 공간에서 두 다면체 사이의 최소거리 계산을 위한 효율적인 알고리즘의 개발

오 재 윤*, 김 기 호**

Development of an Efficient Algorithm for the Minimum Distance Calculation between two Polyhedra in Three-Dimensional Space

Chae-Youn Oh*, Ki-Ho Kim**

ABSTRACT

This paper develops an efficient algorithm for the minimum distance calculation between two general polyhedra(convex and/or concave) in three-dimensional space. The polyhedra approximate objects using flat polygons which composed of more than three vertices. The algorithm developed in this paper basically computes minimum distance between two polygons(one polygon per object) and finds a set of two 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 develops a method to eliminate sets of two polygons which have no possibility of minimum distance occurrence, and an efficient algorithm to compute a minimum distance between two polygons in order to compensate the inherent disadvantage of the algorithm. The correctness of the algorithm is verified not only comparing analytically calculated exact minimum distance with one calculated using the developed algorithm but also watching a line which connects two points making a global minimum distance of a convex object and/or a concave object. The algorithm efficiently finds minimum distance between two convex objects made of 224 polygons respectively with a computation time of about 0.1 second.

Key Words : Polygon(면), Global minimum distance(전역최소거리), Normal vector(법선벡터), Convex object(컨벡스한 물체), Concave object(컨케이브한 물채)

1. 서 론

로봇을 이용하여 어떠한 작업을 수행하기 위해서는 온라인으로 혹은 오프라인으로 수행하고자 하는 작업을 위한 경로를 계획하고, 계획된 경로를 모의실험을 통해 확인하는 과정이 필요하다. 온라인 혹은 오프라인으로 로봇

* 전북대학교 기계공학부
** 한국원자력연구소

작업 경로 계획과 모의실험을 실시하는데 로봇과 주위환경이 이루는 최소거리를 포함하는 로봇과 작업환경 사이의 상호관계를 이해하는 것은 매우 중요하다.

로봇과 주변환경이 모의실험시에는 충돌을 하지 않으나 ①실제 로봇과 주변환경을 그래픽 환경으로 근사함에 따른 오차, ②로봇 제어오차, ③센서오차, ④기계적인 오차, ⑤로봇의 동역학을 고려하지 않고 경로를 계획하는 경우 누락된 로봇의 동역학에 의한 오차 등에 의하여 로봇과 주위환경과 사이에 거리가 너무 가까우면 실제 로봇 운용 중에는 충돌을 일으킬 수 있다. 따라서 로봇과 주변환경 사이의 최소거리가 너무 작지 않게 경로 계획을 하기 위해서는 로봇과 주위환경과 사이에 거리정보가 필요하다. 현재 로봇 경로 모의실험에 많이 사용되는 일반적인 상업용 소프트웨어는 대부분 충돌여부에 대한 정보만 제공할 뿐 로봇이 주위환경과 충돌하지 않는 경우에는 둘 사이의 거리와 같은 정보는 제공하지 않는다.

컴퓨터 그래픽에서 널리 사용되는 삼차원 기하학적 모델링(geometric modeling) 방법은 크게 ①wire-frame 모델링, ②surface 모델링, ③solid 모델링의 세 가지로 분류 할 수 있다[1-5]. 위 세 가지 방법 중 surface 모델링 기법인 면(polygon)을 이용하여 물체를 근사하는 방법은 다른 방법들에 비해 상대적으로 구현하기가 용이할 뿐 아니라 물체에 관한 많은 정보를 제공하므로 일반적으로 로봇과 작업환경을 나타내는데 많이 사용된다. 면에 의해 근사된 물체는 임의의 직선과 항상 둘 이하의 교차점을 가지는 경우로 정의되는 컨벡스한 물체(convex object)와 임의의 직선과 셋 이상의 교차점을 가질 수 있는 경우로 정의되는 컨케이브한 물체(concave object)로 구분된다.

면(polygon)으로 근사한 컨벡스한 물체와 컨벡스한 물체 사이에서는 임의의 시작점으로부터 최소거리를 이루는 점으로의 접근이 항상 보장되므로(convex problem)[6] 두 물체 사이의 최소거리를 찾는 것은 비교적 용이하여 효율적인 알고리즘들이 많이 개발되었다[7-14]. Bobrow[8]는 찾고자 하는 최소거리를 이루는 점과 아주 가까운 최초 시작점을 찾는 방법을 통하여 알고리즘의 효율성을 증가시키는 방안을 고안하였다. Hurteau와 Stewart[9]는 가장 작은 반복횟수를 통해 최소거리를 이루는 점으로 찾아가는 방안을 제시하였다. Smith[10]는 그래픽 하드웨어를 이용하여 효율적으로 삼차원 물체들 사이의 최소거리 계산 방법을 개발하였다. Gilbert, Johnson과 Keerthi[13]는 최소거리 계산 시간이 다면체를 구성하는 꼭지점의 수에 선형적인 알고리즘을 개발하였으며, Cameron[14]는 이 알고리즘의 효율성을 향상시킨 알고리즘을 제시하였다.

그러나 두 물체 중 한 물체라도 컨케이브하면 임의의 시작점으로부터 전역(global) 최소거리를 이루는 점으로 접근을 보장하는 시작점을 구하는 것이 용이하지 않으므로 최소거리를 구하는 것이 상대적으로 용이하지 않다. 이 같이 물체가 컨케이브 함으로써 발생되는 문제점을 해결하는 방법으로 컨케이브한 물체를 단순한 컨벡스한 물체들의 조합으로 분해(convex decomposition)한[15-19] 후에 컨벡스한 물체 사이의 최소거리를 찾는 알고리즘을 이용하는 연구가 활발히 진행되고 있다. 컨케이브한 물체를 단순한 컨벡스한 물체들의 조합으로 분해하는 방법은 알고리즘을 이해하기가 용이하지 않고, 분해되는 컨벡스한 물체 크기에 따라서 비효율적일 수 있을 뿐 아니라[18], 경우에 따라서는 가장 단순한 컨벡스한 물체인 사면체(tetrahedra)로 분해가 어려울 수도 있다는[19] 문제점을 가지고 있다.

본 논문에서는 삼차원 그래픽 환경에서 컨벡스한 면(convex polygon)으로 근사된 컨벡스한 물체와 컨케이브한 물체를 포함하는 일반적인 물체들 사이의 최소거리 정보를 제공할 수 있는 이해하기 쉬우며, 수학적으로 복잡한 이론을 가지지 않는 효율적인 알고리즘을 개발한다.

2. 두 직선 사이의 최소거리

삼차원 공간에서 무한한 길이를 갖는 임의의 두 직선 사이의 최소거리는 Fig. 1에서 보여 주는 바와 같이 두 직선에 공통으로 수직인 직선 사이의 거리이다[2]. 이 경우는 다음과 같은 두 식을 동시에 만족하는 두 점을 찾으면 이 두 점 사이의 거리가 두 직선 사이의 최소거리가 된다.

$$(\overrightarrow{point\ 1} - \overrightarrow{point\ 2}) \cdot \overrightarrow{line\ 1} = 0 \qquad (1)$$

$$(\overrightarrow{point\ 1} - \overrightarrow{point\ 2}) \cdot \overrightarrow{line\ 2} = 0 \qquad (2)$$

그러나 유한한 길이를 갖는 두 직선 사이에서는 공통법선이 두 직선 사이에서 존재할 수도 있으나, 존재하지 않

을 수도 있기 때문에 최소거리를 이루는 점을 찾기는 그리 쉽지가 않은 문제이다[2,20].

본 논문에서 개발하고자 하는 두 물체 사이의 최소거리

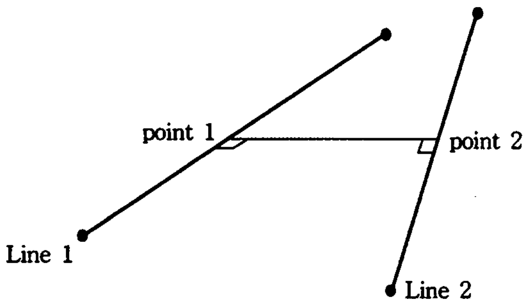

Fig. 1 Minimum distance between two straight lines

계산을 위한 알고리즘은 유한한 길이를 갖는 두 직선 사이의 최소거리 정보를 기본으로 필요로 한다. 그러므로 유한한 길이를 갖는 두 직선 사이의 최소거리를 계산하는 알고리즘의 효율성이 본 논문에서 개발하고자 하는 알고리즘의 효율성에 많은 영향을 미친다. 본 논문에서는 참고문헌[20]에서 개발한 알고리즘을 이용하여 삼차원 공간에서 유한한 크기를 가지는 두 직선사이의 최소거리를 이루는 점들을 찾는다.

3. 두 면 사이의 최소거리

삼차원 공간에서 정의되는 유한한 크기를 갖는 두 면 사이의 최소거리는 Fig. 2에서 보여 주는 바와 같이 두 면에 공통으로 수직인 직선 사이의 거리이다[2]. 이 경우는 다음과 같은 두 식을 동시에 만족하는 두 점을 찾으면 이 두 점 사이의 거리가 두 면 사이의 최소거리가 된다.

$$(\overrightarrow{point\ 1} - \overrightarrow{point\ 2}) \times \overline{normal\ vector\ of\ \ polygon\ 1} = 0 \quad (3)$$

$$(\overrightarrow{point\ 1} - \overrightarrow{point\ 2}) \times \overline{normal\ vector\ of\ \ polygon\ 2} = 0 \quad (4)$$

그러나 유한한 크기를 갖는 두 면 사이에서는 공통법선이 두 면 안에 존재할 수도 존재하지 않을 수도 있기 때문에 최소거리를 이루는 점을 찾기는 그리 쉽지가 않은 문제이다[2,7].

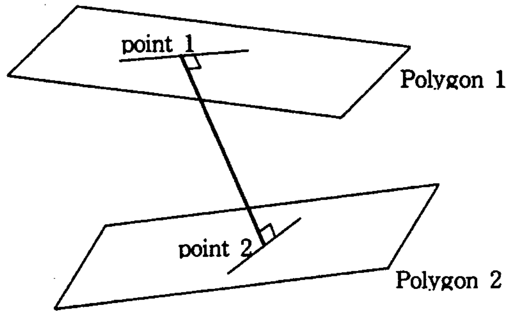

Fig. 2 Minimum distance between two polygons

본 논문에서는 유한한 크기를 갖는 평평하고 컨벡스한 면(flat and convex polygon) 사이에서 최소거리를 이루는 두 점을 찾기 위한 다음과 같은 알고리즘을 개발한다.

〈알고리즘 I : 유한한 크기를 갖는 두 면 사이의 최소거리〉

Step 1. 각 면의 중심점을 참조점으로 한다.

Step 2. 탐색선의 계산

(1) 두 참조점이 모두 면 안에 있는 경우

(i) 각 면에서 참조벡터 $\vec{v}$ 계산

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

* 상대면 에서는 $-\vec{v}$ 가 된다.

(ii) 각 면에서 탐색 방향 계산

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

* $\vec{n}$: 면의 법선벡터

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

(iv) 각 면에서, 현 참조점을 지나는 탐색 방향 벡터 $\vec{s}$ 방향의 직선들과 면의 모서리와의 교점을 구해서, 이 교점들의 연결선을 탐색선으로 한다.

(2) 한 참조점은 모서리 위에 있고 다른 참조점은 면 안에 있는 경우

(i) 참조점이 모서리 위에 놓이는 면에서는 참조점이 놓이는 모서리를 탐색선으로 한다.

(ii) 참조점이 면 안에 놓이는 경우의 탐색선

1) (i)의 탐색선 양 끝점 중 상대면의 참조점과 가까운 점을 찾는다.

2) 이 점과 (i)의 탐색선을 상대면에 투영한다.

3) 이 같이 상대면에 투영된 점을 지나는 투영된 탐색선 방향의 직선과 상대면의 모서리들과 교점을 구하여, 이 점들을 연결하여 상대면의 탐색선으로 한다.

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

(3) 두 참조점이 모두 모서리 위에 있는 경우

(i) 참조점이 놓이는 각 모서리를 각 면의 탐색선으로 한다.

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

Step 3. 유한 크기를 갖는 두 탐색선 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 거리가 현재 고려중인 두 면 사이의 최소거리가 된다.

위의 알고리즘은 거의 모든 경우에서 면과 면 사이의 최소거리를 대략 2-3회 반복 안에 구하는 효율성을 보여주었다.

4. 두 물체 사이의 최소거리

본 논문은 모든 물체를 동일 직선 상에 놓이지 않는 세 개 이상의 꼭지점을 반 시계방향으로 연결하여 정의되는 평평하고 컨벡스한 면(flat and convex polygon)을 이용하여 이 면들이 서로 겹치지 않고, 면이 한 군데도 빠지지 않게 근사한다고 가정한다. 여기서 세 개 이상의 꼭지점이 동일 직선 상에 놓이지 않는다는 것은 각 면이 면적을 가지며, 연속한 두 모서리 벡터의 외적을 이용하여 그 면으로부터 밖을 향하는 법선벡터를 정의 할 수 있음을 의미한다.

본 논문에서 개발하는 삼차원 공간에서 일반적인 다면체(convex and/or concave) 사이의 최소거리 계산을 위한 알고리즘은 면과 면 사이의 최소거리를 이루는 점을 찾아, 이 면들의 조합 중에서 가장 작은 최소거리(global minimum distance)를 이루는 조합을 찾는 것이다. 이와 같이 일반적인 다면체 사이의 최소거리를 구하는데 면과 면 사이의 최소거리를 이용하는 알고리즘의 장점은 물체의 속성(convex 혹은 concave)에 관계없이 모든 경우에 확실하게 물체와 물체 사이의 최소거리를 계산할 수 있다는 것이다. 그러나 이 방법은 어느 물체를 좀 더 정확하게 근사하기 위해 면 수를 늘이면, 면 사이의 최소거리 계산을 위한 면의 조합은 이의 자승에 비례하여 증가하므로 전역 최소거리 계산 시간이 급격히 증가한다는 단점을 가지고 있다. 본 논문에서는 이와 같은 단점을 최소화하기 위하여 다음과 같은 불필요한 면과 면 사이의 최소거리 계산을 제거하는 방법을 고안하여 알고리즘의 효율성을 향상시키고자 하였다.

최소거리 계산이 불필요한 면의 조합을 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. 즉, 각 면 중심점과 중심점 사이의 거리에서 각 면 중심점과 꼭지점 사이의 최대거리의 합을 뺀 값이 프로그램 사용자가 설정하는 적절한 거리보다 작은 경우에만 최소거리 계산을 수행한다.

(2) 면과 면 사이의 관계로부터 전역 최소거리 발생 가능성이 없는 경우: 두 물체 사이의 전역 최소거리는 서로 마주보고 있는 면 사이에서 발생한다. 이를 수학적으로 기술하면, 두 면 법선벡터(normal vector) 사이의 내적(dot product)이 음(-)이고, 두 면 중 한 면(기준면)의 중심점에서 상대면의 중심점을 향하는 벡터(참고벡터, reference vector)와 기준면 법선벡터 사이의 내적이 양(+)인 경우에만 두 물체 사이의 최소거리가 발생할 가능성이 있다.

Fig. 3은 파이프(concave object) 안에 박스(convex object)가 있는 경우를 보여주고 있다. 이 그림에서 (n_1면과 n_4면), (n_1면과 n_5면), (n_1면과 n_6면), (n_2면과 n_6면)의 네 개의 면 조합을 고려해보자. (n_1면과 n_5면), (n_2면과 n_6면) 조합은 각 면 사이의 법선벡터 내적이 양(+)이므로 최소거리 발생 가능성이 없는 조합이므로 고려 대상에서 제외된다. 나머지 (n_1면과 n_4면), (n_1면과 n_6면) 조합은 각 면 법선벡터 내적이 음(-)이므로 일단 최소거리가 발생할 가능성을 생각할 수 있다. n_1면의 법선벡터

와 참고벡터 $\bar{V}_2$의 내적은 음(-)이므로 (n_1면과 n_6면) 사이에서는 최소거리가 발생 가능성이 제거된다. 그러나 n_1면의 법선벡터와 참고벡터 $\bar{V}_1$의 내적은 양(+)이므로 (n_1면과 n_4면) 사이에서는 최소거리가 발생할 가능성이 있는 면의 조합이다. 따라서 위의 네 개의 면의 조합 중 (n_1면과 n_4면)사이에서만 최소거리가 계산된다.

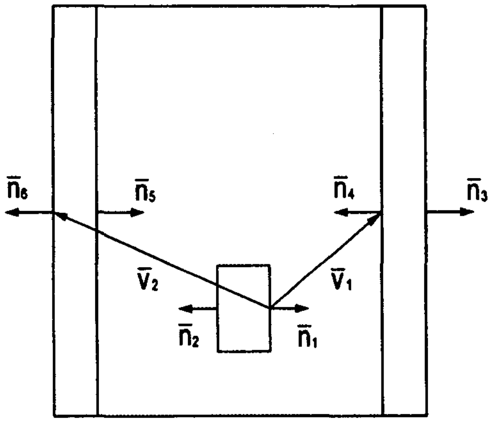

Fig. 3 Normal vectors and reference vectors

본 논문에서 개발되는 알고리즘의 효율성을 향상시키기 위하여 위 조건(1)과 조건(2)에 해당되지 않는 두 면의 조합에 대해서만 최소거리 계산을 수행한다. 다음은 본 논문에서 개발한 일반적인 두 물체 사이의 최소거리 계산을 위한 알고리즘이다.

〈알고리즘 II : 일반적인 두 물체 사이의 최소거리〉

Step 1. 초기 각 물체의 면 번호와 초기 전역 최소거리를 세팅한다.
set i = 1(for object 1)
j = 1(for object 2)
set global minimum distance
= positive big number

Step 2. 현재 고려하고 있는 두 면 법선벡터의 내적이
만약, 음수(-)이면, "Step 3" 으로 가고,
만약, 양수(+)이면, "Step 8 (ii)" 로 간다.

Step 3. 각 면의 중심점을 참조점으로 세팅한다.

Step 4. 참고벡터 계산

$$\vec{v} = \overrightarrow{object\ 1\ \text{참조점}} - \overrightarrow{object\ 2\ \text{참조점}}$$

Step 5. 각 면의 법선벡터 $\vec{n}$과 참고벡터 $\vec{v}$의 내적이
만약, 양수(+) 이면, "Step 6" 으로 가고,
만약, 음수(-) 이면, "Step 8 (ii)" 로 간다.

Step 6. 두 면 사이의 최소거리 계산 필요성 여부 검사
tmp_dist = (두 면 참조점 사이의 거리)
-(두 면 참조점과 각 꼭지점 사이의 최대거리합)
(i) 만약, (tmp_dist) 〈 (tolerance) 이면, "Step 7" 로 가고,
(ii) 만약, (tmp_dist) ≥= (tolerance) 이면, "Step 8 (ii)" 로 간다.

Step 7. 〈알고리즘 I〉을 이용하여 현 두 면 사이의 최소거리 계산

Step 8. (i) 전역 최소거리 갱신
(ii) 면 번호 i and/or j 갱신
(iii) 만약, (j = m) 이고, (i= n) 이면, "Step 9" 로 간다.

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

본 논문에서 개발한 알고리즘의 정확성과 효율성을 검증하기 위하여 몇 개의 컨벡스한 물체와 컨케이브한 물체를 만들어서 테스트하였다. Fig. 4는 컨케이브한 물체(파이프) 안에서 컨벡스한 물체(상자)가 움직이는 경우 최소거리를 계산하는 예를 보여 준다. 그리고 Fig. 5는 컨벡스한 물체(구)와 컨벡스한 물체(구) 사이에서 최소거리를 계산하는 예를 보여 준다. 두 그림에서 각 물체를 연결하는 직선은 두 물체 사이에서 전역 최소거리를 형성하는 두 점을 연결한 것이다.

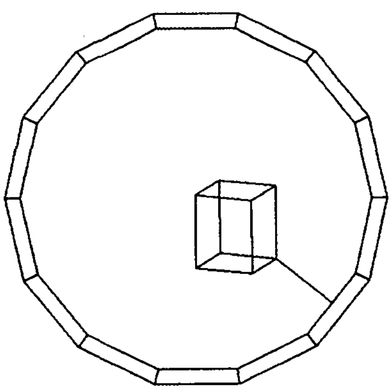

Fig. 4 An example of minimum distance calculation between concave object(pipe) versus convex object(box)

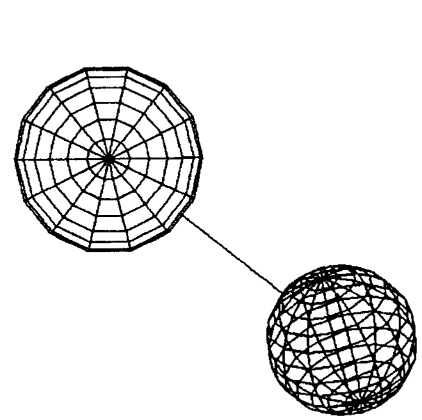

Fig. 5 An example of minimum distance calculation between convex object(sphere) versus convex object(sphere)

개발된 알고리즘의 정확성은 두 가지 방법을 통해 검증하였다. 한 방법은 위 그림들에서 보여주는 바와 같이 기하학적으로 단순한 컨벡스한 물체와 컨케이브한 물체를 만들어서 계산이 용이한 몇 가지 상대위치에서 수치적으로 계산한 정확한 값과 본 논문에서 개발한 알고리즘으로 계산한 값을 비교하는 것이다. 이때 정확한 값과의 차이가 물체를 다각형으로 근사함에 따른 오차 정도의 차이를 보여주므로 알고리즘이 정확함을 검증하였다. 다른 한 방법은 여러 조합의 두 물체의 상대적 위치를 바꿔가면서 전역 최소거리를 이루는 두 점을 Fig. 4와 Fig. 5에서 보여 주는 바와 같이 선으로 연결하여 개발된 알고리즘의 정확성을 시각적으로 검증하였다.

본 논문에서 개발한 알고리즘은 컨벡스한 물체와 컨케이브한 물체 사이의 경우는 상대 위치에 따라 최소거리 계산을 필요로 하는 면과 면의 조합이 많아지는 이유로 인하여 다소 계산시간이 길어지나, 컨벡스한 물체와 컨벡스한 물체의 경우에는 서로 마주보는 면의 조합만이 고려됨으로 인하여 각각 224개 면으로 구성된 두 컨벡스한 물체(구)의 경우에도 "실리콘 그래픽스 워크스테이션 Indigo 2"에서 0.1초안에 최소거리를 계산하는 효율성을 보였다.

5. 결 론

본 논문은 세 개이상의 꼭지점으로 구성되는 평평하고 컨벡스한 면을 이용하여 나타낸 다면체(convex and/or concave polyhedron) 사이의 최소거리를 계산하기 위한 효율적인 알고리즘을 개발하였다. 본 논문에서 개발하는 알고리즘은 면과 면 사이의 최소거리를 계산하여 두 물체 사이에서 제일 작은 거리를 이루는 두 면의 조합을 찾는 것을 기본으로 한다. 이 알고리즘은 어떠한 경우에서도 확실하게 두 물체 사이의 최소거리를 계산할 수 있다는 장점이 있다. 그러나 물체를 좀 더 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 없는 두 면의 조합을 제거하기 위한 방법을 고안하였고, 두 면 사이에서 최소거리 계산을 위한 효율적인 알고리즘을 개발하였다. 본 논문에서 개발되는 알고리즘의 정확성은 해석적으로 구한 정확한 값과 알고리즘을 이용해 구한 값을 비교하여 검증하였을 뿐 아니라, 두 물체의 상대위치를 바꾸어 가면서 최소거리를 이루는 두 점을 연결하여 시각적으로 검증하였다. 그리고 본 논문에서 개발되는 알고리즘은 컨벡스한 물체와 컨케이브한 물체의 경우에는 상대위치에 따라 최소거리 계산을 필요로 하는 면과 면의 조합이 많아지는 이유로 인하여 다소 계산시간이 길어지나, 각각 대략 224개 면으로 구성된 두 컨벡스한 물체 사이의 최소거리를 0.1초안에 계산하는 효율성을 보였다.

참 고 문 헌

1. Foley, J. D., Van Dam, A., Feiner, S. K., Hughes, J. F., and Phillips, R. L., "Introduction to Computer Graphics," Addison - Wesley Publishing Company, 1994.
2. Mortenson, M. E., "Geometric modeling," John Wiley & sons, 1985.
3. Anand, V..B., "Computer Graphics and Geometric Modeling for Engineers," John Wiley & sons, 1993.
4. Rodriguez, W., "The Modeling of Design Ideas," McGraw-Hill, 1992.
5. Requicha, A. A. G., "Representations for Rigid Solids: Theory, Methods, and Systems," Computing Surveys Vol.12, No. 4, pp. 437-464, 1980.
6. Turnbull, C. and Cameron S., "Computing distances between NURBS-defined convex objects," ICRA '98. Leuven, 1998.

7. Schwartz, J. T., "Finding the minimum distance between two convex polygons," Information Processing Letters. Vol. 13, No. 4,5, pp. 7-9, 1981.
8. Bobrow, J. E., "A direct minimization approach for obtaining the distance between convex polyhedra," The International Journal of Robotics Research. Vol. 8, No. 3, pp.65-76, 1989.
9. Hurteau, G. and Stewart N. F., "Distance calculation for imminent collision indication in a robot system simulation," Robotica Vol. 6, pp. 47-51, 1988.
10. Smith, R. C., "Fast robot collision detection using graphics hardware," IFAC Robot Control(Syroco '85). Barcelona. Spain, pp. 277-282, 1985.
11. Lin, M. C., "Efficient Collision Detection for Animation and Robotics," Ph.D. dissertation, University of California at Berkeley, Berkeley, California, 1993.
12. Cameron, S. A. and Culley, R. K., "Determining the minimum translational distance between two convex polyhedra," IEEE International Conference on Robotics & Automation. San Francisco, pp. 591-596, 1986.
13. Gilbert, E. G., Johnson, D. W. and Keerthi, S. S., "A fast procedure for computing the distance between complex objects in three-dimensional space," IEEE Journal of Robotics & Automation, Vol. 6, No. 2, pp. 193-203, 1988.
14. Cameron, S., "Enhancing GJK: computing minimum and penetration distances between convex polyhedra," IEEE International Conference on Robotics & Automation, Albuquerque, pp. 3112-3117, 1997.
15. Chazelle, B. and Palios, L., "Triangulating a Nonconvex Polytope," Discrete and Computational Geometry, Springer-Verlag, Vol. 5, pp. 505-526, 1990.
16. Bajaj, C. L. and Dey, T. K., "Convex Decomposition of Polyhedra and Robustness," SIAM Journal on Computing Vol. 21, No. 2, pp. 339-364, 1992.
17. Chazelle, B., "Convex Partitions of Polyhedra : A Lower Bound and Worst Case Optimal Algorithm," SIAM Journal on Computing Vol. 13, pp. 488-507, 1984.
18. Chazelle, B. and Palios, L., "Decomposing the Boundary of a Nonconvex Polyhedra," Algorithmica, Vol. 17, pp. 245-265, 1997.
19. Ruppert, J. and Seidel, R., "On the Difficulty of Tetrahedralizing 3-Dimensional Non-Convex Polyhedra," Fifth ACM Annual Symposium on Computational Geometry, pp. 380-392, 1989.
20. Lumelsky, V. J., "On fast computation of distance between line segments," Information Processing Letters 21, pp. 55-61, 1985.