

실시간 동적 3차원 환경에서의 효율적인 충돌탐지

알고리즘

이영호⁰, 김성범, 정승원, 한대만, 한상진, 구용완
수원대학교 컴퓨터학과

lovingkiss⁰@hanmail.net, doklib@mail.suwon.ac.kr powerbrains@neoidea.com, {han38, han37}@hanmail.net,

ywkoo@mail.suwon.ac.kr

Efficient Collision Detection Algorithm in Dynamic 3D Environment at Run-time

Yung-Ho Lee⁰, Seong-Beom Kim, Sung-Won Jeong, Dae-Man Han, Sang-Jin Han, Yong-Wan Koo
Dept. of Computer, The University of Suwon

요 약

본 논문에서는 실시간에 강체 운동을 하는 일반적인 모델사이의 효율적인 충돌검사 알고리즘을 제안한다. 기존의 경계볼륨 알고리즘에 계층적 구조를 적용하였다. 이는 볼록한 물체를 위한 브로노이 영역 기반의 충돌검사 알고리즘을 오목한 물체에도 적용할 수 있도록 확장한다. 추가적으로 빠르게 움직이는 물체에 대한 관통을 탐지하기 위해서 물체의 이동 경로에 대한 교차 검사를 진행한다. 구현된 알고리즘은 일반적인 응용에서 기대한 성능 향상을 얻을 수 있다.

1. 서 론

현재의 3차원 컴퓨터 그래픽 분야에서 많은 기술들이 개발되고 있다. 그 중에서 충돌 탐지 문제는 3차원 컴퓨터 애니메이션, 물리 기반 모델, 로봇 공학 그리고 컴퓨터 시뮬레이션이나 가상 현실 등에서 중요하게 다루어지는 주제이다.

본 논문에서는 강체 역학에 의한 둘 또는 그 이상의 물체들 사이의 효율적인 충돌 탐지 알고리즘을 제안한다.

본 논문은 다음과 같이 구성되어 있다. 관련연구에서 충돌 탐지를 이해하는데 기초적인 지식들을 살펴본다. 그리고 볼록 다면체를 갖는 물체들 사이의 최단거리 부분을 일정 시간 내에 추적하는 알고리즘을 살펴보고, 그 것들을 일반적인 다면체 모델을 위한 간접 탐지로 확장하는 방법을 알아본다. 기존의 충돌 처리 알고리즘들을 사용한 계층 구조의 부분 분할 방법과 관통 탐지 방법을 제안하고, 그 구현과 성능에 대해서 설명한다. 끝으로 결과를 분석하고 앞으로의 연구 방향을 제시하고자 한다.

2. 관련연구

충돌 탐지의 문제를 이해하는데 필요한 기초적인 지식들을 수학적 방법을 사용하여 살펴본다.

2.1 충돌탐지 알고리즘

평면과 직선이 교차한다면, 다음 단계로 그 교점을 구한다. 가장 간편하게 직선을 표현하는 방법은 거리를 이용하는 것이다. 위치 P_0 에서 D 방향으로 거리 d 를 갖는 직선의 방정식은 다음과 같다.

$$P(d) = P_0 + dD$$

이 연산의 수식은 폴리곤과 폴리곤의 교차 검사를 위해서는 매 폴리곤 쌍마다 매우 비싼 연산인 아크코사인을 여섯 번씩이나 해야한다. 폴리곤 단위의 교차 검사 비효율적이므로 이를 개선하기 위해서 지역적 분할이나 경계 볼륨과 같은 기술을 사용하여 충돌할 가능성이 없는 폴리곤들을 지역적으로 제거한다.

2.2 최단거리 알고리즘

최단거리 점의 추적을 위한 간단하고 효율적인 계산 알고리즘을 살펴본다. 충돌할 가능성이 있는 두 볼록 다면체 사이의 유클리드 거리를 계산하기 위해서 그 최단거리 부분(점, 선, 면)을 찾고 유지한다. 다음 수식은 면의 법선 벡터와 동일 직선 상에 위치하는 속성을 이용하여 최단거리를 추적하는데 사용된다.

$$F_1(s, t, u, v, a_1, a_2) = F_{11}(s, t, u, v, a_1) - F_{12}(s, t, u, v, a_1) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$F_2(s, t, u, v, a_1, a_2) = F_{21}(s, t, u, v, a_2) - F_{22}(s, t, u, v, a_2) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

위에서 나타낸 수식은 여섯 개의 미지수를 갖는 여섯 개의 등식이다. 하지만, 각 면은 실제 면의 부분집합으로 정의되기 때문에 큰 의미를 갖지 못한다. 최단거리 부분들 사이의 경계 면은 다시 다음과 같이 표현될 수도 있다. 두 개의 수학적 면이 $f(x, y, z) = 0$ 와 $g(x, y, z) = 0$ 일 때, 최단거리 부분을 결정하는 문제는 다음 6개의 등식에 대한 근을 찾는 것으로 축소될 수 있다.

$$\begin{aligned} f(x_1, y_1, z_1) &= 0 \\ g(x_2, y_2, z_2) &= 0 \end{aligned}$$

$$\begin{pmatrix} f_x(x_1, y_1, z_1) \\ f_y(x_1, y_1, z_1) \\ f_z(x_1, y_1, z_1) \end{pmatrix} = a_1 \begin{pmatrix} g_x(x_1, y_1, z_1) \\ g_y(x_1, y_1, z_1) \\ g_z(x_1, y_1, z_1) \end{pmatrix} \quad (\text{수식 1})$$

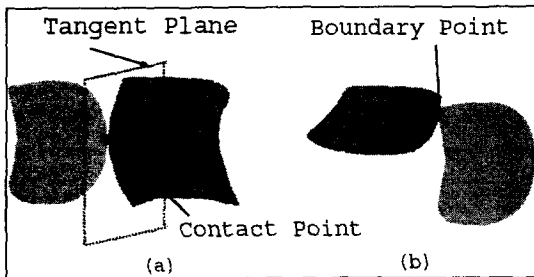
$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} - \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = a_2 \begin{pmatrix} g_x(x_2, y_2, z_2) \\ g_y(x_2, y_2, z_2) \\ g_z(x_2, y_2, z_2) \end{pmatrix}$$

(수식-1)에 대한 Bernstein 경계는 $n^2(n^2+3)(n-1)^2$ 이다. Canny와 Emiris는 부족한 수식을 섞어 사용하여 Bernstein 경계를 계산한다[7].

3. 충돌 탐지 알고리즘의 설계

3.1 접촉의 탐지

간접 탐지의 문제는 두 물체 사이에 어떤 접촉이 있는가를 결정하는 것이다. 시작할 때 두 물체가 겹쳐져 있지 않아야 한다. 물체들이 움직임에 따라, 물체들 사이에 세밀한 접촉이라도 있는가를 알아내는 것이 중요해진다. 면의 교차와 경계의 교차 두 가지 종류가 있다.



(그림 1) 두 베지어 면 사이의 면 교차와 경계 교차

면 교차 : 이것은 (그림 1-a)과 같이 두 면 사이의 접촉에 대한 것이다. 접점은 각 면의 안쪽에 존재하고 그 점의 법선은 동일 직선 상에 놓여진다. 이 경계는 다음과 같이 수식화 할 수 있다.

$$\begin{aligned} \mathbf{F}(s, t) &= \mathbf{G}(u, v) \\ (\mathbf{F}_s(s, t) \times \mathbf{F}_t(s, t)) \cdot \mathbf{G}_u(u, v) &= 0 \\ (\mathbf{F}_s(s, t) \times \mathbf{F}_t(s, t)) \cdot \mathbf{G}_v(u, v) &= 0 \end{aligned} \quad (\text{수식 2})$$

다른 가능한 경계 교차는 비슷한 방법으로 얻을 수 있다. 교점은 전역 방법을 사용하여 쉽게 계산할 수 있다. (그림 1-b)가 그 예이다.

3.2 관통의 탐지

대부분의 경우에 시간 제약적인 형식으로서 물체의 움직임을 표현하는 것은 어렵다. 하지만 결과적으로 물체의 위치와 기하학 정보만을 갖는 경우에도 충돌에 대한 정확한 시간을 계산하는 알고리즘은 가능하다. 다면체 모델과 곡면 모델에 대한 알고리즘에 기초하여 두

물체가 가까워지는지를 알 수 있다. 결국 그 주요 루틴은 짧은 시간에 충돌을 탐지할 수 있다. 하지만 첫 단계에서 두 물체가 충돌하고 있지 않다가 다음 단계에서 서로를 관통할 수 있다. 면의 접촉과 경계의 접촉에 근거한 접촉 분석은 두 물체의 면이나 경계가 접촉할 때만 만족한다. 두 물체가 관통되었을 때는 더 이상 만족하지 않는다. 이 절에서는 곡면 모델 사이의 관통에 대한 상태를 공식화하고 그것을 충돌 탐지에 사용하는 방법을 살펴본다.

(그림 1)에서 보는 바와 같이 두 물체 사이의 유한한 접촉 영역이 존재하면, 두 물체는 관통한 것이다. 따라서 관통은 두 면의 경계가 교차하는 것에 대한 1 차원 교차의 결과를 갖는다.

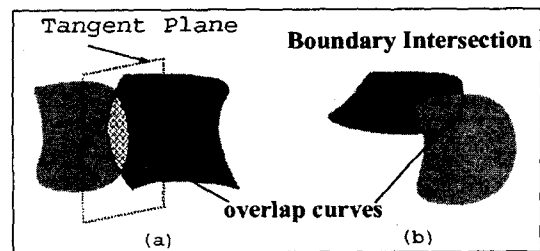
(그림 1-b)에서와 같은 경계 교차를 보자. $\mathbf{F}(s, t)$ 에 해당하는 경계 곡면을 따라 접촉이 존재할 때, 주어진 예에서 두 면은 관통한다. 결과적으로 다음 방정식의 공통 해를 계산하면

$$\begin{aligned} \mathbf{F}(s, t) &= \mathbf{G}(u, v) \end{aligned} \quad (\text{수식 3})$$

$\mathbf{G}(u, v)$ 의 경계 곡면에 해당하지 않거나 이 방정식의 한 개 이상의 확실한 해가 존재하는 영역에서 실제 해가 존재한다. 하지만, 경계 곡면 $\mathbf{F}(s, t)$ 을 따라 결정되는 작은 움직임은 해당 영역의 이들 방정식에 대한 해를 결과로 갖는다.

이제 (그림 2-a)와 같이 두 수학적 면 사이의 면 접촉을 고려해 보자. 역시 비슷한 방법이 면 접촉에 적용된다. 면 접촉을 탐지하기 위해서 다음과 같이 네 개의 미지수를 갖는 네 개의 방정식에 대한 해를 계산하는 방법을 사용한다.

$$\begin{aligned} f(x, y, z) &= 0 \\ g(x, y, z) &= 0 \\ \begin{pmatrix} f_x(x, y, z) \\ f_y(x, y, z) \\ f_z(x, y, z) \end{pmatrix} &= a \begin{pmatrix} g_x(x, y, z) \\ g_y(x, y, z) \\ g_z(x, y, z) \end{pmatrix} \end{aligned} \quad (\text{수식 4})$$



(그림 2) (a) 면 교차에 의한 관통 (b) 경계 교차에 의한 관통

4. 구현 및 성능평가

본 논문에서 기술하고 제안한 알고리즘을 구현하여, 그 성능을 실험해 보았다. 불록 하지 않은 모양의 공간에 이동하는 물체를 두고, 실시간에 이동시 발생하는 충돌에 따라 반응하는 프로그램을 목표로 했다.

시뮬레이션 환경은 프로세서로 Intel Pentium III 700MHz와 메모리로 SDRAM 512 MB를 사용하는 시스템이다. 시뮬레이션 프로그램은 Microsoft Visual C++ 6.0(SP2) 틀에서 Microsoft DirectX 6.0의 Direct3D IM 라이브러리를 사용하여 제작되었다. 경계 불륨으로는 상자를 사용하여 수작업으로 그 생성과 계층 구조를 구축하고, 물체의 이동 경로에 대한 교차 탐지로 관통을 탐지하도록 했다.

```
foreach(obj1 = object in space)
{
    foreach(obj2 = object in space)
    {
        if(obj1 == obj2) continue;
        CalculateClosestDistance(obj1, obj2);
        dist = Closest Distance between obj1 and obj2
        if(dist < delta)
            // delta is Minimum Closest Distance
            {
                DetectCollision(obj1, obj2);
            }
        else
            {
                DetectPenetraion(obj1, obj2);
            }
    }
}
```

시뮬레이션을 위한 의사 코드

오목한 물체를 위해서 계층적 구조를 사용한 알고리즘은 폴리곤 수의 증가에 크게 영향을 받지 않고 비슷한 평균치의 프레임 수를 보였다. 시뮬레이션에서 블록하거나 블록하지 않거나 상관없이 일반적인 모양을 갖는 물체들에 대해서 계층적 구조를 사용하지 않을 때와 사용했을 때를 비교했다. 각 실험에 사용한 공간은 총 폴리곤 500개, 1000개, 5000개를 사용하여 구성했다. 계층적 구조를 사용하지 않을 경우, 500개의 폴리곤에 대해서 85~90 프레임을 나타냈다. 1000개의 폴리곤에 대해서 75~80 프레임을 나타냈고, 5000개의 폴리곤에 대해서는 40~60프레임을 나타냈다. 프레임의 변동이 심한 것은 블록하지 않은 물체를 충돌 검사에 제거하지 못한 경우가 있기 때문이다. 반면 계층적 구조를 사용할 때에는 500개의 폴리곤에 대해서 평균 90 프레임 이상을 나타냈다. 1000개의 폴리곤에 대해서 약 80 프레임을 나타냈고, 5000개의 폴리곤에 대해서는 70 프레임을 나타냈다. 물체의 이동 경로에 대한 교차 탐지를 사용함으로써 사용하지 않을 때 탐지하지 못했던 빠르게 움직이는 물체에 대해서도 충돌 및 관통을 잘 탐지하였다.

5. 결론

본 논문에서는 동적 환경에서의 효과적인 충돌탐지 알고리즘에 대해서 기술했다. 그 알고리즘은 몇 개의 수학적 식으로 기술될 수 있는 경계들을 갖는 다면체 모델에 적용된다. 이것은 컴퓨터 그래픽과 기하학 모델에서 공통적으로 사용되는 Nurbs 모델을 포함한다. 결

과적인 알고리즘은 방정식의 해법에 대해서 지역적 및 전역적 방법을 같은 계층적 구조로 표현된다. 이 알고리즘의 구현은 실제로 적용될 수 있다.

본 논문에서 제시한 실시간 동적 환경에서의 효과적인 충돌 탐지 알고리즘으로 기존의 경계 불륨에 계층 구조를 사용하는 방법과 움직이는 물체의 변위 정보를 사용하여 그것들 사이의 관통을 탐지하는 방법을 제안했다. 제안한 알고리즘은 실제 구현의 성능 평가에서 실시간 환경에 응용하기에 적합하다. 충돌 탐지는 물리 기반 모델의 출발점으로써 계속적인 연구를 통해 관련 분야에 기여할 것을 기대한다.

6. 참고문헌

- [1] P.K. Agarwal and M. Sharir, Red-blue intersection detection algorithms, with applications to motion planning and collision detection, *SIAM J. Comput.*, 19:297-321, 1990.
- [2] P.K. Agarwal, M. van Kreveld, and M. Overmars, Intersection queries for curved objects, In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 41-50, 1991.
- [3] D. Baraff, Curved surfaces and coherence for non-penetrating rigid body simulation, *ACM Computer Graphics*, 24(4):19-28, 1990.
- [4] R. Barzel and A. Barr, A modeling system based on dynamic constraints, *ACM Computer Graphics*, 22(4):31-39, 1988.
- [5] D. N. Bernstein, The number of roots of a system of equations, *Funktsional'nyi Analiz i Ego Prilozheniya*, 9(3):1-4, Jul-Sep 1975.
- [6] J. L. Bentley and J. H. Friedman, Data structures for range searching, *Computing Surveys*, 11(4), December 1979.
- [7] J. Canny and I. Emiris, An efficient algorithm for the sparse mixed resultant, In *Proceedings of AAECC*, 1993.
- [8] J. F. Canny, Collision detection for moving polyhedra, *IEEE Trans. PAMI*, 8:pp. 200-209, 1986.
- [9] B. Chazelle, An optimal algorithm for intersecting three-dimensional convex polyhedra, In *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 586-591, 1989.
- [10] B. Chazelle and D. P. Dobkin, Intersection of convex objects in two and three dimensions, *J. ACM*, 34: 1-27, 1987.
- [11] D. P. Dobkin and H. Edelsbrunner, Space searching for intersecting objects, *J. Algorithms*, 8:348-361, 1987.