

복잡한 다수의 기하학적 물체들로 이루어진 가상 환경에서

단계적인 상세를 이용한 충돌 검출 시뮬레이션

이경명[○], 한창호

인하대학교 전자계산공학과

Collision Detection Simulation Using LOD In Virtual Environment of Geometrically Multi and Complex Bodies

Kyung Myung Lee[○], Chang Ho Han

Dept of Computer Science and Engineering, Inha University

요 약

큰 규모의 가상 환경에서 물체들은 사실적인 이미지효과와 함께 충돌 검출과 같은 다이내믹한 효과를 동반하여 더욱더 실제 물체와 같이 느껴지도록 해야 한다. 특히, 가상세계에서의 충돌 검출은 실시간으로 계산되어야 한다. 가상환경을 이루는 기하학적 데이터의 양이 엄청나게 증가하면서 충돌 검출은 하나의 병목으로서 극복되어야 할 문제로 여겨져 왔다. 본 논문에서는 많은 수의 복잡한, 3차원의 기하학적 모델들로 구성된 가상환경에서 단계적인 상세를 이용하여 가상 물체를 사실적으로 표현하면서도 어느 정도의 충돌 검출 병목을 해결하는 절충적인 방법을 제안하고 많은 양의 기하학적 데이터를 포함하는 토끼 모델을 사용하여 단계적인 상세를 이용한 가상환경을 구현하였다.

1. 서론

충돌 검출(collision detection)은 모델링된 물체들이 서로 파고드는지(penetrates)의 여부를 결정하는 것을 말한다. 충돌 검출은 컴퓨터 애니메이션(computer animation), 물리적이반 모델링(physically-based modeling), 기하학적 모델링(geometric modeling) 로봇틱스(robotics)에서 충족되어야 하는 기본적인 요소이다. 특히 가상현실 분야에서 적절한 충돌 응답(collision response)을 제공하는 충돌 검출은 더욱 사실적으로 보이게 한다. 큰 규모의 가상 환경(large-scale virtual environment)은 사실적으로 보이는 가상 물체(real and virtual object)들로 채워진 컴퓨터가 생성한 세계이다[CLMP95]. 그러한 세계에서 객체들은 서로 관통하지 않으며 밀리거나 닿지 않을 때 기대되어지는 대로 움직여야 한다. 그러한 작용들은 정확한 충돌검출을 요구한다. 가상 세계에서는 수 천, 수 만개의 물체들이 존재할 수 있다. 이러한 상황에서 모든 방울에 대한 충돌 가능성을 테스트하는 것은 상호작용 가능한 가상세계의 특성을 잃어버리게 한다[CLMP95]. [HILCGM97]에서는 많은 수의 복잡한 물체들이 움직이는 VRMI 환경하에서 정확한 충돌검출을 위한 가속화된 충돌 검출방법을 제안하였다. 그러나 위에서 제안한 충돌 검출방법은 물체의 복잡성과 물체의 개수라는 변수에 의존하여 한 프레임 당 처리시간에 영향을 끼치고 있다. 많은 컴퓨터 그래픽스 응용은 확실히 사실적인 수준을 유지하기 위해서 복잡하고 매우 상세히 표현되는 모델들을 요구한다[GH97]. 아주 복잡한 3차원의 모델은 점점 빠르게 일반화되고 있다. 예를 들면, 레이저 스캐닝 시스템은 색깔과 같은 추가 정보를 포함하여 몇 십만 개의 정점(vertex)들로 구성되는 기하학적 모델을 만들어낸다[CFRDDW96]. 그러나 또한, 3차원의

기하학적 모델의 사실적인 표현을 위해서 항상 상세한 표현수준이 요구되는 것만은 아니다. 원래의 복잡한 사실적인 3차원의 모델을 더 단순한 기하학적 표현으로 변환하는 것이 더 유용한 경우가 있다.

가상 세계를 바라보는 관찰자의 시점에서 볼 때, 시점에 가까이 있는 물체일수록 사실적인 표현이 요구된다. 그러나 시점에서 아주 먼 거리에 물체가 위치한다면 그 물체에 대하여 더 이상 상세한 표현을 고집할 필요는 없다. 즉 관찰자가 단지 그 물체의 기본적인 속성 또는 표현 특성을 인식할 수 있으면 될 것이다. 1976년 초반, Clark은 관찰자로부터 멀리 떨어져서 시작적인 중요성이 덜 요구되는 기하학적인 물체에 대하여 더 단순한 기하학적인 구조를 사용할 것을 제안하였다[C76]. 이렇게 기하학적인 모델의 구조를 단순화해 가는 것을 단계적인 상세(Levels of Detail LOD)이라한다. 본 논문에서는 많은 수의 복잡한 3차원의 기하학적인 모델들로 이루어진 가상환경에서 단계적인 상세를 이용하여 가상 물체를 사실적으로 표현하면서도 충돌 검출 병목을 해결할 수 있는 절충적인 방법을 제안하고 구현하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 많은 수의 복잡한, 3차원의 기하학적 모델들로 이루어진 가상환경에서의 충돌 검출과 모델의 단순화와 관련된 두 가지의 접근 방법, 그리고 단계적 상세를 생성하는데 이용되는 폴리곤의 단순화(polygonal simplification)에 대하여 기술한다. 3장에서는 LOD를 이용한 가상 환경 충돌검출 시스템을 제안하고 4장에서는 시스템 구현 및 결과를 분석한다. 5장에서는 결론을 맺고 향후 연구 방향을 기술한다.

2. 관련 연구

2.1 모델의 단순화

상호 작용이 가능한 렌더링(rendering)을 수행하기 위하여, 많은 양의 기하학적 데이터를 포함하는 모델을 그 데이터량이 충분히 적어질 때까지 모델을 단순화하는 두 가지의 접근 방법이 있다[GA96] 한 가지 접근 방법은 보여지는 것만을 모으는 방법(visibility culling)이다 이 방법은 모델에서 시각적으로 보이는 일부분을 결정하여 그 부분만을 렌더링하는 것이다 다른 접근 방법은 LOD 를 이용하는 것으로 특히, 워크스루(workthrough)환경에서 렌더링되어야 하는 엄청나게 복잡한 데이터의 양을 줄여 렌더링하는 것이다 첫번째의 방법은 많은 프리미티브(primitive)가 보이는 경우에 효율적이지 못하고 LOD 방법은 보통 다수의 LOD 를 수동(manual)으로 생성해야 하는 단점이 있다 본 논문에서 구현하는 시스템은 많은 양의 데이터가 시각적으로 보여지며, 충돌 검출의 상대적인 효율성때문에 LOD 방법을 이용하였다

2.2 폴리곤의 단순화

LOD 의 생성을 위해 이용되는 폴리곤 단순화의 목적은 물체 원형의 주요한 시각적 특성을 유지하는 더 단순화된 모델을 생성하기 위하여 원형 모델에서 삼각 폴리곤(triangle polygon)과 같은 프리미티브를 제거하는 것이다

LOD 생성을 위해서 사용되는 폴리곤 단순화에서 고려되어야 할 사항은 다음과 같다

1) 형태 보존(Shape Preservation)

평평한 폴리곤은 인접한 폴리곤의 노말(normal)을 검사함으로써 알 수 있다. 이러한 폴리곤은 합병될 수 있다 날카로운 모서리부분은 보존되어야 한다 인접한 면들의 노말 사이의 각을 비교하여 이러한 모서리를 찾아낸다 피라미드 모델의 꼭대기와 같은 포인트 모서리도 또한 보존되어야 한다

2) 근사 오차(Approximation Error)

단순화의 정도 깊으로 물체와의 거리가 사용될 수 있고 또한 직접적으로 넘지 말아야 할 어떠한 범위 값을 지정할 수도 있다

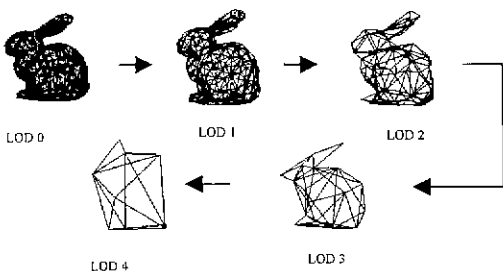
3) 토폴로지 보존(Topology Preservation)

폴리곤의 단순화에서 토폴로지를 보존하거나 단순화하는 것은 선택의 문제이다 토폴로지의 수정이 가능하도록 하는 것은 단순화의 더 많은 여지를 남겨둔다

4) 조절 가능한 단순화(Controllable Simplification)

한 물체의 단순화 정도가 그 물체의 부분마다 다르도록 단순화를 조절할 수 있도록 할 수도 있다

[그림 1]은 폴리곤의 단순화를 이용한 토끼 모델의 연속적인 LOD 를 와이어프레임(wireframe)형태로써 나타낸 것이다 또한 [그림 1]과 관련된 각 LOD 의 정점과 폴리곤 정보를 [표 1]에서 보인다



[그림 1] 연속적인 LOD 모델

[표 1] 각 LOD 에 대한 정점과 폴리곤 정보

LOD 모델	정점(Vertex)수	삼각형 면(Triangle Polygon)수
LOD 0	1040	2000
LOD 1	260	500
LOD 2	66	124
LOD 3	19	32
LOD 4	9	14

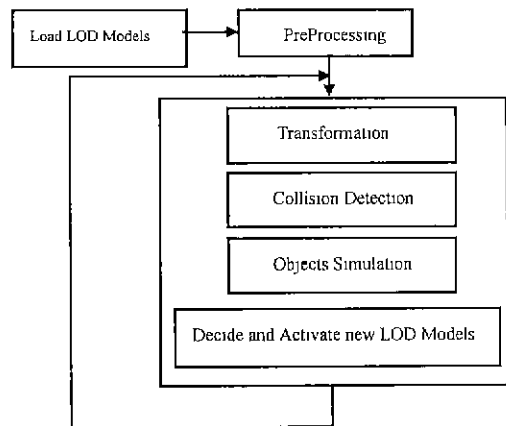
2.3 복잡한 3 차원의 기하학적인 모델들로 이루어진 가상환경에서의 충돌 검출

충돌 검출의 목적은 물체들 사이의 모든 기하학적인 접촉을 검사해 내는 것이다. 만약 물체들의 위치와 방향에 대한 정보를 미리 알고 있다면 시간의 함수로서 충돌 검출을 해결할 수 있다. 그러나 이러한 방법은 상호 작용하는 응용에서는 사용할 없다[CLMP95] 따라서 상호작용 가능한 가상 환경에서의 충돌 검출은 주요한 병목으로 간주된다 이러한 충돌 검출을 극복하기 위하여 제안된 것으로 V-COLLIDE 시스템[HLCGM97]이 있다

V-COLLIDE 시스템은 복잡한 3 차원의 단단한 기하학적인 폴리곤 모델들로 이루어진 상호작용 가능한 VRML 가상 환경에서 정확한 충돌 검출이 가능함을 보여주는 시스템이다 이 시스템은 기존의 I-COLLIDE[CLMP95]시스템의 기술과 OBB-Tree[GLM96]의 데이터 구조를 통합하였다 이 시스템은 모델 수(삼각형 폴리곤 수)의 증가량에 의존하여 한 프레임 당 충돌 검출 계산 시간이 증가함을 보여준다. 폴리곤 수가 엄청나게 증가하게 되면 그 시스템은 실시간으로 상호 작용 가능한 가상 환경의 특성을 잃어버리게 될 것이다

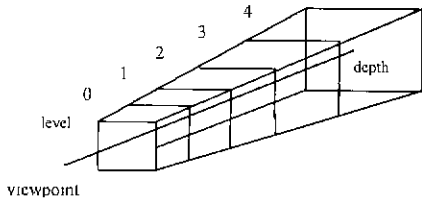
3. LOD 를 이용한 가상 환경 충돌 검출 시스템의 설계

본 논문은 전체 가상시스템의 수행능력에 영향을 미치는 폴리곤의 수를 LOD 를 이용하여 실시간으로 줄이는 효과를 가짐으로써 한 프레임 당 평균적인 충돌 검출 계산 시간을 줄여 전체 가상시스템의 수행능력을 향상시키고자 한다 본 논문에서 제안한 시스템의 전체적인 구성도는 [그림 2]와 같다



[그림 2] 시스템의 전체적인 구성도

각 물체에 대한 LOD 모델들은 메모리로 로딩된 후 전처리 과정을 거친다 이 과정에서 관찰자(view point)위치에서의 시각적인 면을 고려하여 적절하게 LOD 레벨 간격을 결정한다 [그림 3]은 LOD 레벨 수를 다섯으로 하여 가상 환경 볼륨을 적절히 다섯으로 분할한 것을 보여준다 숫자는 LOD 레벨을 말하며, LOD 레벨 0의 간격이 d 라던, LOD 레벨 1의 간격은 $2*d$, LOD 레벨 2의 간격은 $3*d$, LOD 레벨 3의 간격은 $4*d$, LOD 레벨 4의 간격은 $5*d$ 로 설정한다



[그림 3] LOD 레벨을 위한 가상 환경 볼륨의 분할

또한 초기좌표를 결정하고 초기좌표로 각 물체의 초기 LOD 레벨을 결정한다 결정된 LOD 레벨에 해당하는 LOD 모델을 충돌 검출 연산에 포함(LOD 모델의 활성화)시킨다 전처리 과정 이후, [그림 2]와 같이 좌표변환, 충돌검출, 물체의 시뮬레이션, 각각의 물체에 대한 새로운 LOD 모델 결정 및 활성화의 과정을 반복한다 새로운 LOD 모델과 현재의 LOD 모델의 교체는 간단하게 LOD 모델을 가리키는 포인터 값을 포함하는 배열의 인덱스를 교체하는 과정이다 따라서 LOD 모델의 실시간 교체는 시스템의 전체 수행 능력에 거의 영향을 미치지 않는다 따라서 LOD를 가상환경에 적절히 이용함으로써 가상환경의 사실적인 효과와 시스템의 수행 속도면에서 모두 이득을 얻게 된다 [그림 4]에서 새로운 LOD 레벨에 대한 LOD 모델의 교체를 보인다

```
RenderScene()
{
    for( i = 0, i < 물체의 개수; i++)
    {
        Transformation();
    }
    Collision_detection();

    for(int i = 0, i < 물체의 개수, i++)
    {
        Simulation ();
    }

    for( i = 0, i < 물체의 개수, i++)
    {
        new LOD_LEVEL 계산.
        if(new LOD_LEVEL != current LOD_LEVEL)
        {
            new LOD 모델의 비활성화,
            current LOD 모델 = 새로운 LOD 모델;
            current LOD 모델의 활성화.
        }
    }
}
}
```

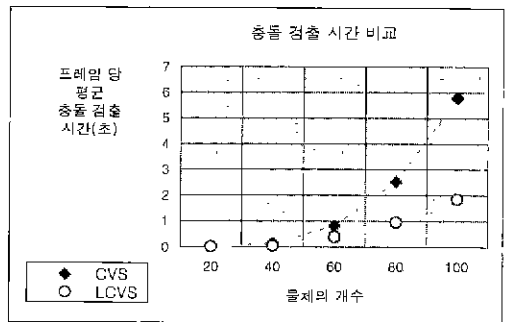
[그림 4] 새로운 LOD 레벨에 내린 LOD 모델의 교체

4 실험결과

본 논문은 메모리 64Mbytes, 166MHz Pentium PC 의 Windows 95

OS 환경에서 구현하였다 가상 환경을 시뮬레이션하기 위하여 OpenGL 1.1을 이용하였다 구현에서 사용된 LOD 모델은 [그림 1]과 같으며 각 LOD 모델의 구성 데이터는 [표 1]과 같은 구성을 가진다

[그림 1]의 토끼 모델의 원형의 출처는 Stanford Computer Graphics Laboratory 이다 충돌 검출 루틴만을 포함한 가상 환경 시스템(CVS)과 본 논문에서 제안한 시스템(LCVS)을 [그림 5]에서 비교한다 CVS의 경우, 폴리곤 2000 개로 구성된 모델을 사용하였고, 본 논문의 시스템의 경우, LOD 레벨 수를 다섯으로 하여 [그림 3]과 같이 구성하였다 100 프레임동안의 충돌 검출 계산시간을 산출하여 한 프레임 당 평균적인 충돌 검출 계산시간을 비교한다



[그림 5] 한 프레임 당 평균적인 충돌 검출 계산시간의 비교

5. 결론 및 향후 연구과제

물체의 정보, 특히 기하학적인 정보량이 엄청나게 많은 가상환경에서 적절한 LOD의 이용은 가상환경의 사실적인 효과를 보장하면서 시스템 효율을 높여준다. 폴리곤을 단순화하는 작업은 복잡한 수학적 계산으로 많은 처리시간이 요구되므로 폴리곤을 단순화한 모델들을 미리 생성한 후, 이를 시뮬레이션에 이용하였다 따라서 실시간으로 폴리곤의 단순화를 이용하기 위해서 그 처리시간을 줄이는 연구가 더 필요하다 시각적인 면을 고려해 볼 때, 관찰자의 시점에서 물체는 자연스럽게 LOD 변화과정을 거쳐야 한다 이러한 부분들에 대해서는 향후 연구과제로 남기도록 한다

6. 참고문헌

[C76] Clark, J. H. "Hierarchical geometric models for visible surface algorithms," *Communications of the ACM* 1976

[GH97] Michael Garland and Paul Heckbert, "Surface Simplification Using Quadric Error Metrics," *SIGGRAPH 97*, 1997.

[Lin94] Ming Chih Lin, "Efficient Collision Detection for Animation and Robotics," *Technical Report UCB/ERL-94-13*, 1994

[GA96] Daniel G. Aliaga, "Dynamic Simplification using Textures," *Technical Report UNC/CSD-TR96-007*, 1996

[GLM96] S. Gottschalk, M. Lin and D. Manocha, "OBB-Tree: A Hierarchical Structure for Rapid Interference Detection," *In Proc of ACM SIGGRAPH 96*, 1996

[CLMP95] J. Cohen, M. Lin, D. Manocha, and M. Ponamgi, "I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments," *In Proc of ACM Interactive 3D Graphics Conference*, 1995.

[CPRDDW96] A. Certain, J. Popovic, T. DeRose, T. Duchamp, D. Salesin and W. Stuetzle, "Interactive Multiresolution Surface Viewing," *In Proc of ACM SIGGRAPH 96*, 1996

[HLCGM97] T. Hudson, M. Lin, J. Cohen, S. Gottschalk and D. Manocha, "V-COLLIDE: Accelerated Collision Detection for VRML," *In Proc of VRML'97*, 1997