
LOD(Level-of-Detail)이용한 3D객체의 동적 계층의 충돌 검사 성능 향상

LOD(Level-of-Detail) using Dynamic-Hierarchies of collision detection efficiency improvement in 3D object



이춘호, Chun-Ho Yi*, 김태용, Tae-Yong Kim**,

요약 ~ 본 논문에서는 현재 3D 그래픽뿐만 아니라 게임에서 정확한 충돌감지(collision-detection)나 컬링(culling)등은 3D공간에서 이러한 표준객체를 중심으로 많은 연구가 이루어지고 있다. 3D그래픽 분야에서 H/W의 놀라운 발달과 3D게임을 즐기는 게이머들이 좀 더 사실적인 표현에 깊은 관심을 가지고 있다. 90년대 중반 이후로 많이 연구되어진 3D 게임 엔진과 알고리즘 중에서 표준 3D 객체의 다양한 충돌 알고리즘을 분석하고, 기존의 3D 객체의 단순한 Hierarchies 구조에서 탈피하여 3D공간상에서 LOD(Level-of-Detail) 알고리즘을 이용하여, 3D객체가 3D 공간상에서 충돌검사의 성능을 향상시켜서 3D 게임의 필수 요소인 3차원 공간상의 효율적인 렌더링과 사실적인 표현의 알고리즘을 제안하여 실시간을 중요시 하는 3D 게임에서 사실감과 효율성을 높일 수 있게 제안한다. ↓

↓
Abstract ~

In this paper introduce Standard 3D object(Bounding-Volume). In 3D game very efficient control algorithm Using collision detection which controls the convenient of a game based on Standard 3D object specially collision-detection. This algorithm is designed LOD(Level-of-Detail) using Dynamic-Hierarchies of collision detection efficiency improvement in 3D object ↓

핵심어: 3D Game Engine, Dynamic-Hierarchies weight, Collision-Detection, Level-of- Detail ↓



본 연구는 ITRC(Information Technology Research Center)와 서울시 산학연 협력사업의 지원으로 수행되었음.

*주저자 : 서울디지털대학교 멀티미디어학부 교수

**공동저자 : 중앙대학교 첨단영상대학원 교수



1. 서론

표준 3D 객체의 활용은 3D 엔진에서 렌더링 프로세스 신속한 컬링(culling)과 충돌검사(collision detection)를 결정하기 위한 경계범위로서 편리하게 사용되어 지고 있다.

이러한 표준3D객체의 종류는 크게 구체, 유향상자, 캡슐, 마름모, 실린더, 타원체 등으로 분화되어 지고, 또 각각의 3D 객체를 중심으로 세부적인 다양한 알고리즘을 나누어지고 있다.

현재 3D 그래픽뿐만 아니라 게임에서 정확한 충돌감지(collision-detection)나 컬링(culling)등은 3D공간에서 이러한 표준객체를 중심으로 많은 연구가 이루어지고 있다. 3D그래픽 분야에서 H/W의 놀라운 발달과 3D게임을 즐기는 게이머들이 좀 더 사실적인 표현에 깊은 관심을 가지고 있다.

3D 게임 개발에서 빠르고 사실적인 렌더링 기법이 필수적이다. 더욱이 이러한 신속한 렌더링 기법의 핵심은 충돌검사(collision-detection)와 컬링(culling)을 통해서 이루어진다.

기존의 3D객체의 충돌알고리즘에서 'LOD바탕으로 하여 3D객체의 동적계층의 충돌검사 성능 향상'의 기법을 제안하고, 기존의 제안되어지고 있는 3D 그래픽의 알고리즘 보다 좀 더 사실적인 충돌검사의 방법을 모색하여, 실시간을 중요시 하는 3D 게임에서 사실감과 효율성을 높일 수 있게 제안한다. ↙

2. Bounding-Volume 정의 및 분류

: 실시간 성능을 요구하는 모든 3D 엔진 알고리즘은 다음과 같은 특징이 있다.

① 공간적/ 시간적 응집도 이용

객체들은 한 프레임에서 다른 프레임까지 공간의 영역을 점유

② 사전계산

사전계산이나 오프라인 단계에는 상당한 메모리가 필요하다. 여기서 공간 분할과 경계 볼륨 등을 미리 계산에 포함시켜 메모리 효율적인 사용을 분석.

③ Collision detection system에 요구되는 특징

충돌계수는 collision handling의 일부로서 게임 상에서는 일반적으로 다음과 같은 절차를 갖는다.

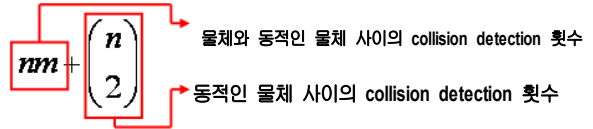
- Collision Detection(충돌탐지)
- Collision Determination(충돌결정)
- Collision Response(충돌 후 반응)

: 3D 모델들이 멀리 있을 때나 가까울 때나 상관없이 많은 수의 polygon들로 이루어진 모델들을 가지고 실시간 충돌 검출을 할 수 있다

Polygon soups(다각형 덩어리)를 처리함으로써 불룩하다거나 이웃 정보가 필요하다거나 하는 등의 제약 없이 일반적인 다각형 형태로 단순화하여 처리한다.

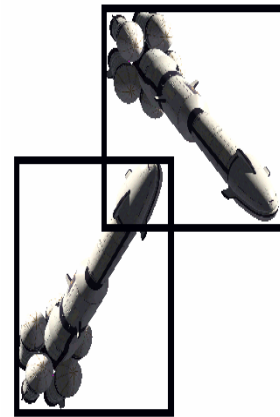
3D모델들은 rigid-body로 변환을 할 수 있다.

- 효율적이고, 모델과 가깝도록 bounding volume 생성하여 제공하여 크기가 작은 (모델 크기와 가까운) bounding volume은 collision algorithms의 속도를 향상시키고 Bounding volume 생성 속도도 향상되어야 한다. n개의 움직이는 물체, m개의 정적인 물체 사이에 충돌 검출을 따른다.



2.1 AABB(Axis Aligned Bounding Box)

: 좌표축과 정렬된 바운딩 박스



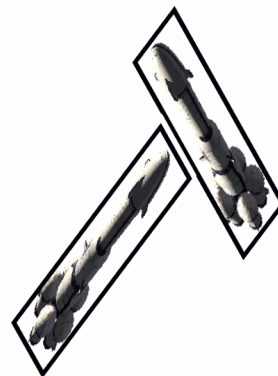
[그림 1 AABB 방식]

AABB방식은 정렬과 스위프(sweep)접근이라 불리는 정렬된 리스트를 통과하여 충돌을 처리한다.

3D 표준객체에서 작업진행이 다른 방법보다 상대적으로 간단함과 빠르게 객체간의 교차를 검사하는 특징을 갖고 있으나 객체간의 충돌 효율성이 다른 방법보다는 상대적으로 떨어진다.

2.2 OBB(Oriented Bounding Box)

: 오브젝트의 이동방향과 정렬한 바운딩 박스

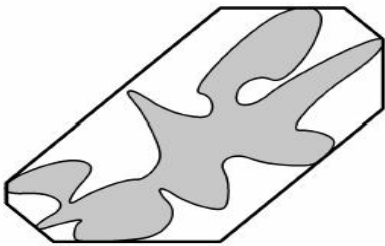


[그림 2 OBB 방식]

두개의 OBB간의 간섭에 대한 일반적인 생각은 각각의 오브젝트에서 모서리 → 면으로 충돌검사가 이루어진다. 즉, 2개의 유향상자의 충돌을 검사할 경우 12모서리 × 6면 × 2박스 = 144의 충돌 검사가 이루어져야 한다.

2.3 K-dop Tree

: K-dop은 AABB가 일반화된 것이다. 경계 볼륨은 실제로 계층구조의 형태에서 사용된다. 트리 구조는 조직화되는 경계 볼륨의 효율성은 충돌 결정을 해결하기 위해 얼마나 해당 트리(영역)에 신속히 접근하는 것이다. K-dop에 대한 AABB의 일반화를 위한 이론적 근거는 AABB의 접촉 효율을 증가시키고, 동시에 OBB에 대한 상대적인 장점을 얻는다는 것이다.

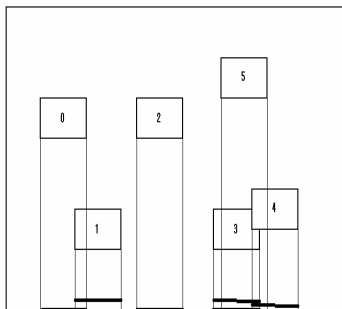


[그림 3 K-dop 방식]

K-dop은 AABB의 구석을 잘라내는 것에 의해 구축된다. K의 값은 경계 부피의 면에 의해 정의되는 반(half)공간들의 숫자이고 K의 값이 높아질수록 더 많은 빈공간이 AABB를 잘라낸다. K-dop의 문제는 K-dop이 일반화된 것이기 때문에 포함된 객체가 회전할 때 갱신되어야 한다는 것이다.

2. 4 축 정렬

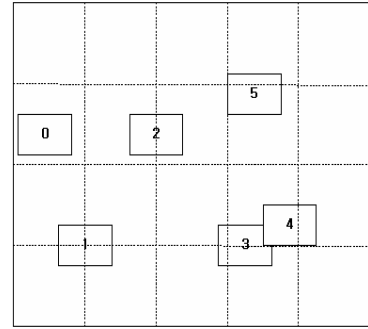
: 충돌 가능성이 있는 3D 객체를 확인 하고 다음 단계로 이를 구체적으로 확인



[그림 4 축정렬 방식]

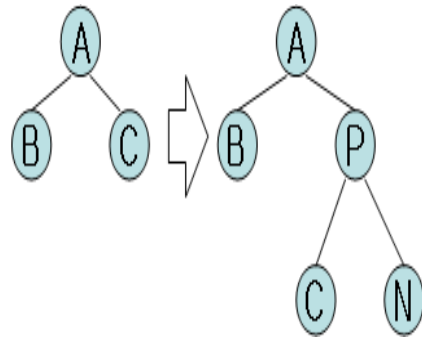
2. 4 그리드 방식

: 오브젝트를 동일한 간격의 그리드 배치 후 셀 내부의 충돌검사



[그림 5 그리드 방식]

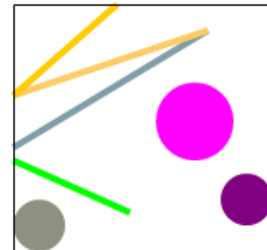
3. 동적 LOD(Level-of-Detail) 적용한 Hierarchy Building을 통한 충돌계수 방식

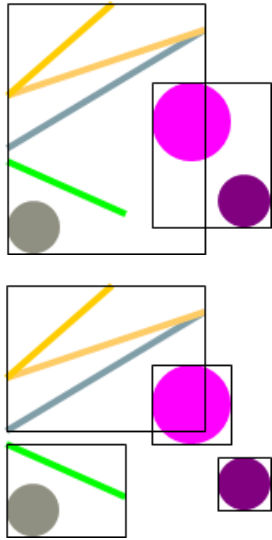


[그림 6 Hierarchy Building]

3.1 Hierarchy Model의 개요:

- 계층적 3D 모델은 각 노드에 간단한 볼륨형태를 가지고 있다.
- 각 모델의 서브 형태는 각각의 위치에서 노드를 중심으로 존재한다.
- 계층적 구조에서 각각의 서브객체를 중심으로 폴리곤이 존재한다





[그림 7 Hierarchy 모델의 적용]

3.2 Bounding Volume Hierarchy 연산

$$F = N_u \times C_u + N_{bv} \times C_{bv} + N_p \times C_p \quad (1)$$

F: 총 충돌 검사 함수

N_u : bounding volumes updated 횟수

C_u : bounding volume updated 메모리비용

N_{bv} : bounding volume pair overlap tests 횟수

C_{bv} : overlap test between 2 BVs 메모리비용

N_p : primitive pairs tested for interference 횟수

C_p : testing 2 primitives for interference 메모리비용

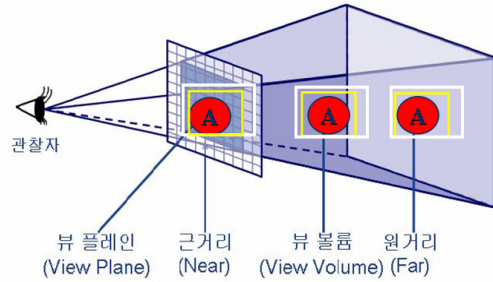
3.3 LOD(Level-of-Detail) 개요

: LOD(Level-of-Detail) 알고리즘은 사람의 시각이 가까운 것은 선명하게 보이고 먼 것은 희미하게 보이는 현상을 이용하여 1976년 James Clark이 화면에 등장하는 물체를 여러 단계의 해상도로 표현할 경우 얻을 수 있는 이점을 언급하면서 최초의 'LOD' 개념을 도입하였다.

1990년대부터 3D 그래픽 관련 된 H/W와 S/W의 비약적인 발전과 3D 관련된 컴퓨터 그래픽스 논문들이 발표 되면서 LOD 알고리즘은 효과적인 3D 그래픽 표현의 하나로써 시점에 안 보이는 것은 삭제(Culling)하고, 또 멀리 보이는 것은 표현하는 메쉬(Mesh)를 대폭 줄이고, 가까운 것은 그대로 표현하는 것을 특징으로 하는 알고리즘의 하나이다.

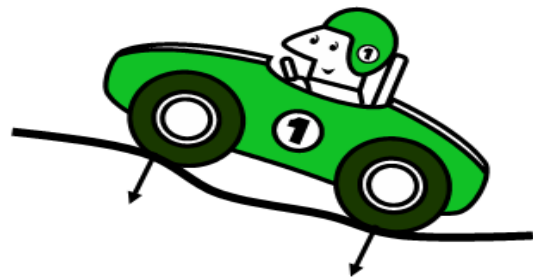
LOD 알고리즘의 일반적인 구성요소로는 시점에 대응하여 표현 하고자 하는 3D그래픽스 화면 즉, 같은 대상의 사물(Object)일지라도 3D공간상의 거리에 따라 가까이 있으면 자세히 표현 되고, 멀리 있으면 간소하게 표현되는 것 점과, 3D 공간상의 표현에서 시점의 이동에 따른 화면이 급격히 떨어지는 단점(Popping)현상을 줄일 수 있는 예러의 값을

최소화 하는 것이다.



[그림 8 Level-of-Detail]

3.3 움직이는 물체를 몇 개의 rays를 통한 근사화 : 모든 기본 요소들을 비교하여 collision detection을 하는 방법 대신 ray를 사용함.

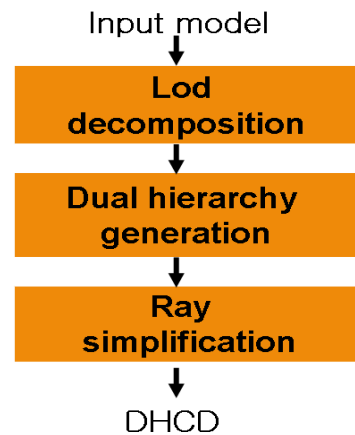


[그림 9 Ray 모델의 적용]

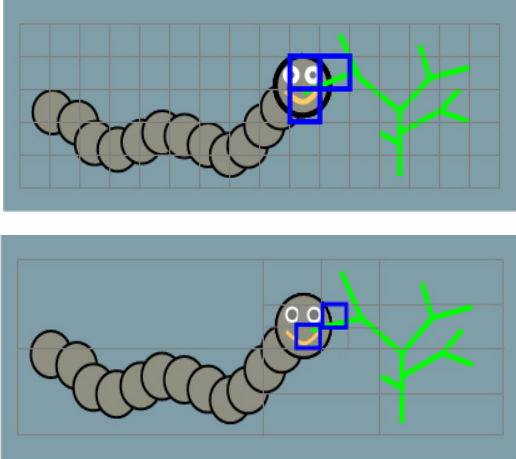
그림처럼 차의 경우는 네 개의 바퀴 intersection test

- 0 : 바닥과 일치
- Positive value : 바닥과 떨어져 있음
- Negative value : 바닥을 뚫고 들어간 경우

4. 동적 LOD(Level-of-Detail) 적용한 Hierarchy Building을 통한 충돌계수 방식 제안



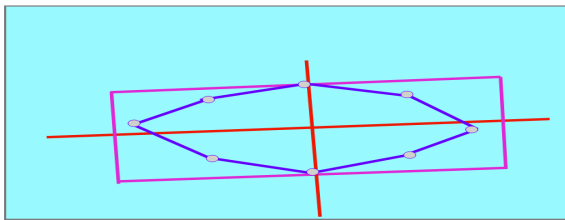
4.1 충돌계수에 Hierarchy Model의 적용



[그림 10 충돌계수에 Hierarchy 모델의 적용]

- 계층적 3D 모델은 각 노드에 간단한 볼륨형태를 을 가진다
- 각 모델의 서브 형태는 각각의 위치에서 노드를 중심 존재
- 계층 구조에서 각각의 서브객체를 중심으로 폴리곤이 존재

4.2 Ray를 통한 기준선의 충돌상자 사실성 향상



[그림 11 Ray를 통한 기준선을 충돌 모델의 적용]

a^i, b^i, c^i : 버텍스의 i '번째의 삼각형의 포인트

μ : 평균값

C : covariance matrix

$$\mu = \frac{1}{3n} \sum_{i=0}^n (a^i + b^i + c^i),$$

$$C_{jk} = \frac{1}{3n} \sum_{i=0}^n (\bar{a}_j^i \bar{a}_k^i + \bar{b}_j^i \bar{b}_k^i + \bar{c}_j^i \bar{c}_k^i), \quad 1 \leq j, k \leq 3 \quad (2)$$

삼각형 폴리곤이 'n'일 경우

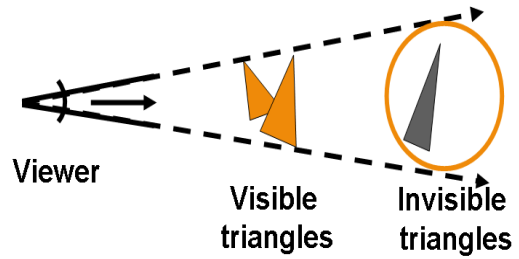
$$\bar{a}^i = a^i - \mu, \quad \bar{b}^i = b^i - \mu, \quad \bar{c}^i = c^i - \mu, \quad (3)$$

4. 3 Ray를 통한 기준선의 충돌상자에 Lod를 적용

: Lod의 알고리즘 특징을 적용하여 3D 공간의 거리에서 충돌 계산을 Hierarchy의 모델의 하위계층 까지 Ray를 이용하여 충돌검사의 효율성 높인다.

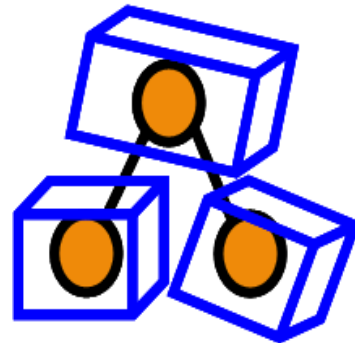
4. 4 Ray를 통한 기준선의 충돌상자에 Lod를 적용한 새로운 아이디어 제안

① Lod 이용한 3D 공간상에서의 객체의 계층 구조 판별

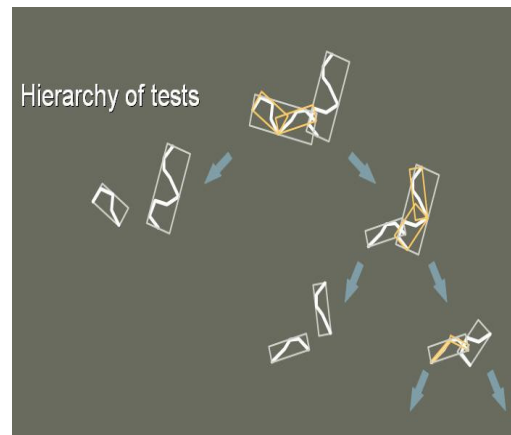


② 3D 모델을 복수 계층 구조로 분할

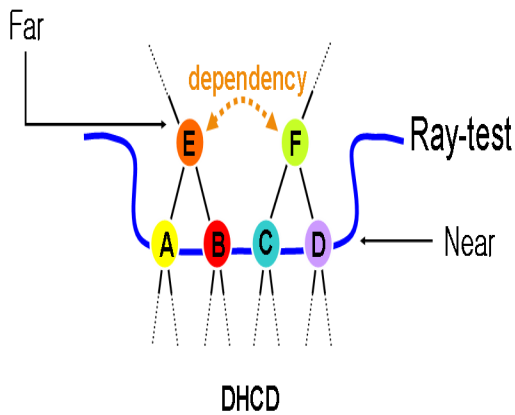
: LOD Hierarchy와 Bounding volume hierarchy로 분할



③ Ray를 계층적 충돌상자에 적용



④ 3D 공간상에서 알고리즘 적용



- 3D 공간상에서 두개의 독립된 계층적 구조의 개체
- 화면상에서 LOD를 이용한 충돌판별식
- 개체의 충돌여부 거리 'δ'
- DHCD적용으로 충돌의 사실성 증가
- : error bound ϵ
- $\text{dist}(tA0, tB0) < \delta + \epsilon$

5. 결론 과 향후 발전방향

현재 컴퓨터 그래픽 카드와 더불어 H/W의 놀라운 발달이 이루어졌지만, 아직도 3D게임에서 실시간 계산 능력에 많은 어려움이 있다. 3D 게임 엔진에서 3D 표준객체와 이에 따른 충돌계수 처리는 지금까지 소개되어진 3D엔진의 중요한 근간이 되고 있는 알고리즘이다.

본 논문에서는 현재 3D게임 엔진에서 3D표준 객체를 계층 구조로 나누어 LOD를 적용하여 충돌 탐지에 대한 사실성에 대한 방법을 연구하였다. 기존의 논문들은 2000년을 전후로 충돌계수에 대해서 특정분야에 제한적으로 적용하여 특화시키면서 발전하고 있다. 여기서는 LOD의 장점인 사실감을 유지하면서 계층적으로 분화된 3D객체를 RAY-TEST를 활용하여 3D 객체의 사실적인 충돌검사를 통하여 속도도 어느 정도 유지하면서 사실성을 높이는 방법을 모색하였다.

본 연구의 실험환경은 PIV-2.0, Geforce5200, 이미지 사이즈가 약1,000폴리곤인 비행기를 가지고 제안된 알고리즘 초기 실험하였다.

향후, 실험환경 구축에 중점을 두어 단순한 비행기 형태를 탈피하여 좀 더 다양한 3D 객체를 실험하여 새로운 알고리즘에 검증과 보완을 하여 좀 더 발전된 형태의 3D게임 제작에 이바지한다.

참고문헌

- [1]Peter Lindstrom, "Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization, IEEE Transactions on Visualization and Computer Graphics, 8(3), July-September 2002, pp. 239-254.
- [2]Hugues Hoppe, "Geometry clipmaps: Terrain rendering using nested regular grids," Siggraph 2004.
- [3]Jonathan Blow, "Terrain Rendering at High Levels of Detail" GDC 2000
- [4]Louis Castle, "Real-time Continuous LOD for PCs and Consoles," GDC 2000
- [5] A. Watt and F. Plocarpo, "3D Game: Real - Time Rendering and Software Technology" Addison-Wesley, 2001
- [6] Tim Schroeder "Collision Detection Using Ray Casting", Game Developer Magazine 2001
- [7] Mark DeLoura, Charles "Game Programming Gems", ed., River Media, 2000.
- [8] Young J. Kim, Gokul Varadhan, Ming C. Lin, Dinesh Manocha: Fast swept volume approximation of complex polyhedral models, Symposium on Solid Modeling and Applications 2003: 11-22
- [9] Kelly Ward, Ming C. Lin, Joohee Lee, Susan Fisher, Dean Macri: Modeling Hair Using Level-of-Detail Representations. CASA 2003: 41-47
- [10]Stephane Redon, Young J. Kim, Ming C. Lin, Dinesh Manocha, Jim Templeman: Interactive and Continuous Collision Detection for Avatars in Virtual Environments, VR 2004: 117-124
- [11] Sung-Eui Yoon, Brian Salomon, Ming C. Lin, Dinesh Manocha: Fast Collision Detection between Massive Models using Dynamic Simplification, Symposium on Geometry Processing 2004: 139-150