

# 변형된 절두체 컬링을 이용한 3차원 FPS 게임에서의 오브젝트 탐색 연구

최원태\* · 박창민\*

\*영산대학교 게임·컨텐츠학과

## A Study on the Object Search in 3D FPS Games Using Modified Frustum Culling

Won-Tae Choi\* · Chang-Min Park\*

\*Dept of Game · Contents, Young San University

E-mail : nowwind@msn.com, cmpark@ysu.ac.kr

### 요 약

최근, 3차원 슈팅 게임들은 빠른 화면 전환과 카메라 시야에 있는 오브젝트들을 상대로 게임을 한다. 특히 온라인 게임에서는 플레이어의 시야에 있지 않은 상대 오브젝트들의 위협성을 인식하는 것이 매우 중요하다. 본 논문에서는 변형된 절두체 컬링을 이용하여 3차원 FPS 게임에서의 오브젝트를 효율적으로 탐색하는 방법을 제시하였다. 플레이어가 감지하지 못하는 오브젝트를 위해 플레이어와 카메라 위치를 일치 시켰으며, 위협적인 오브젝트들의 개수를 파악하기 위해 플레이어와 오브젝트들의 거리를 이용하였다. 제안한 방법은 향후 3차원 FPS 게임의 발전에 중요한 역할을 할 것이다.

### ABSTRACT

A 3D game plays in fast screen conversion and the objects a partner with camera visual field. Shall recognize a threat of the opponent objects that there is not to a visual field of a player in on-line games. In this paper, we proposed a method to efficiently search for the object in 3D game as used modified frustum culling. For the objects which a player cannot perceive, we accord the player and camera position and used a distance of the player and objects for count of the threatening objects. The method that proposed will play an important role in development of a 3D FPS games.

### 키워드

절두체 컬링(frustum culling), FPS(First Person Shooting) 게임(Game), 온라인 게임(On-line Game)오브젝트 탐색(object search),

### 1. 서 론

최근 국내 온라인 게임 시장이 확대 되면서 많은 FPS(First Person Shooting) 게임이 출시되고 있다. 왜냐 하면 FPS 게임은 대부분 온라인에서 즐기는 대표적인 게임이기 때문이다. 대부분의 이러한 슈팅 게임들은 게임 내에 등장하는 오브젝트들을 local에 초점을 맞추어 빠른 화면 전환 및

카메라 시야에 들어오는 오브젝트만을 상대로 하여 전투를 벌이는 특징들을 포함하고 있다[1]. 따라서 플레이어의 관점에서는 카메라 시야에 들어 오지 않은 적 오브젝트들에 대한 위협성을 감지 하지 못하는 단점들이 있다.

특히 온라인 게임에서 이러한 문제점들은 게임의 흥미를 감소시키는 요인이 된다. 따라서 플레이어들이 게임을 플레이할 때 위협이 되는 오브

젝트들을 미리 인식할 수 있다면 FPS와 같은 1인칭 슈팅 게임에서 새로운 전략 전술 시스템의 출발점이 될 것이다.

한편, 실시간 3차원 게임 월드에서 수많은 폴리곤과 객체를 렌더링 하는 것은 게임을 플레이할 때 속도에 많은 부담을 안겨 준다. 따라서 3차원 그래픽을 구현하는 단계에서 실제로 카메라의 시야 범위에 포함되는 폴리곤들과 객체들만 렌더링 하고, 나머지 것들은 렌더링 하지 않는 절두체 커팅 기법[2]은 이러한 문제를 해결하는데 매우 유용하다.

본 논문에서는 변형된 절두체 커팅(frustum culling) 기법을 이용하여 온라인 슈팅 게임에서 플레이어의 시각에 나타나지 않는 오브젝트들을 탐색하여 전투의 위험성을 인식하게 하는 방법을 제안 한다.

## II. 기존의 절두체 커팅을 이용한 오브젝트 탐색

절두체는 모두 6개의 평면으로 이루어져 있다. 절두체 내부에 점이 포함되는지를 판단하기 위하여 6개의 평면 방정식에 점의 좌표를 대입하여 결과값이 양수(+)면 이 점은 절두체 내부에 포함되어 있는 것으로 판단하여 렌더링 한다. 따라서 절두체 내부에 포함된 점들만 플레이어가 위험성을 느껴 공격해야할 오브젝트로 인식하게 되고 보이지 않는 오브젝트에 대한 위험성을 판단하지 못하게 된다. 이러한 문제점들은 현재 출시된 대부분의 FPS 게임에서 나타나고 있다. 그림 1은 기존의 절두체 커팅 방법을 이용하여 렌더링 되는 오브젝트들을 표현하고 있다.

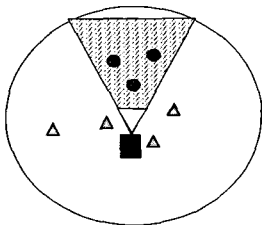


그림 1. 기존의 절두체 커팅을 이용하여 렌더링 되는 오브젝트

그림 1에서 검정색 사각형은 플레이어를 표현하고 있으며 빗금으로 나타낸 부분은 플레이어가 볼 수 있는 영역을 나타낸다. 이 영역 안에 렌더링 되어 있는 오브젝트는 플레이어가 위험성을 인식하게 되는 오브젝트이다. 따라서 빗금으로 이루어진 영역 밖에 위치한 삼각형 모양의 오브젝트들은 플레이어가 인식하지 못하기 때문에 빗금

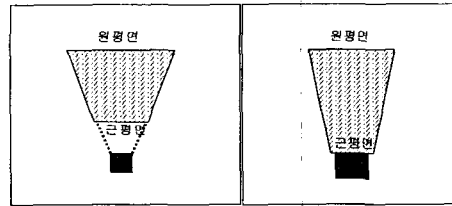
영역에 있는 오브젝트 보다 더욱 가까이 있음에도 불구하고 위협적인 요소로 판단하지 못한다.

## III. 변형된 절두체 커팅을 이용한 오브젝트 탐색

절두체 커팅을 이용한 폴리곤 및 오브젝트의 렌더링 기법은 게임을 플레이 하는데 속도를 증가 시키는 방법을 제공한다[3]. 특히 3D 슈팅 게임에서 속도는 게임의 흥미를 가장 시키는 중요한 역할을 한다. 그러나 게임 플레이 속도를 감소 시키지 않는 범위 내에서 플레이어가 보지 못하는 오브젝트들 까지 탐색하여 전술 및 전략에 이용할 수 있다면 FPS 1인칭 슈팅 게임의 중요한 요소로서 작용할 수 있을 것이다.

### 3.1 변형된 절두체 커팅

절두체 커팅은 실제로 카메라의 시야 범위에 포함되는 폴리곤이나 오브젝트들을 렌더링 한다. 그리고 카메라의 위치는 실제 플레이어가 게임을 플레이할 때 바라보는 시각보다 뒤에 있기 때문에 II 장에서 언급한 문제점들이 발생한다.



(a) 기존의 절두체 (b) 변형된 절두체

그림 2. 위에서 바라본 절두체

그림 2의 (a)에서 나타나고 있는 바와 같이 카메라의 위치와 플레이어의 위치가 일치하지 않다. 따라서 폴리곤이나 오브젝트가 렌더링 되는 영역은 근평면(near plane)을 포함한 원평면(far plane), 좌평면(left plane), 우평면(right plane), 상평면(top plane), 하평면(bottom plane)이다. 따라서 탐색되는 오브젝트들은 이러한 영역 안에서만 이루어진다. 결과적으로 플레이어의 후방평면(rear plane) 영역 밖에 등장하는 오브젝트들은 비록 플레이어를 공격할 수 있는 위험성이 있지만 렌더링 되지 않기 때문에 무시되는 경향이 있다.

이와는 반대로 그림 2의 (b)는 변형된 절두체를 나타내고 있다. 먼저 카메라의 위치와 플레이어의 시각 점이 일치하게 설정한다. 왜냐 하면 카메라의 위치정보보다 후방에 등장하는 오브젝트들을 탐색할 수 있게 하기 위함이다. 뿐만 아니라 카메라 후방에 등장하는 오브젝트들은 당연히 플레이어에게 위협감을 주는 요소가 되기 때문이다. 따라

서 일정 거리 안에서 플레이어에게 접근하는 모든 오브젝트들을 탐색할 수 있는 가능성이 높아진다.

이러한 방법은 FPS 게임에서 매우 유용한 전술로서 작용할 수 있으며 게임의 흥미를 더해주는 주요한 요소가 될 것이다. 뿐만 아니라 게임 인터페이스 제작 시 플레이어에게 오브젝트들의 위협성을 시각적으로 나타내게 하여 긴장감을 더욱 고조시킬 수 있는 그래픽 요소로 작용하게 된다.

특히 온라인 FPS 게임에서는 게임 인터페이스에 위협성의 강약에 따라 메뉴의 칼라를 변화시켜 오브젝트에 대한 상대 플레이어의 위협에 대처할 수 있는 기회를 제공한다.

### 3.2 오브젝트 탐색 알고리즘

본 논문에서 사용하는 변형된 절두체 켤링은 기존 절두체 켤링의 수식이나 계산 방법을 변형하는 것을 의미 하지 않는다. 기존의 절두체 켤링 방법을 이용하되 플레이어가 위치하는 곳에서부터 보이지 않는 오브젝트들 사이의 거리를 계산하여 일정 거리 안에 들어오는 모든 오브젝트들을 탐색하는 방법이다.

먼저 게임 플레이어에 해당하는 조작 가능한 하나의 오브젝트의 위치좌표를(0,0,0)로 설정하고 그 주변의 오브젝트를 계속 검색을 한다. 플레이어가 자신이 아닌 다른 오브젝트가 다가오게 되면 그 오브젝트의 위치 값이 자신과 어느 정도 떨어져 있는지 수식 1 및 수식 2를 이용하여 탐색 판단한다.

$$Dist_1 = \sqrt{(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2} \quad (1)$$

$$Dist_n = \sqrt{(x-x_n)^2 + (y-y_n)^2 + (z-z_n)^2} \quad (2)$$

수식 1에서 x, y, z는 플레이어가 위치해 있는 좌표를 나타내고 있으며  $x_1, y_1, z_1$  은 절두체 내부에 포함되어 있는 오브젝트의 위치 좌표값을 보여주고 있다. 그리고 수식 2의  $x_n, y_n, z_n$  은 절두체 범위내에 포함되지 않은 오브젝트들의 개수를 나타내고 있다. 이때 Dist 는 수식 3에서 보여 주고 있는 일정 범위의 영역 안에서만 계산된다.

if  $Dist_1 \geq Dist_n$  : 위협성은 상승

if  $Dist_1 < Dist_n$  : 플레이어는 절두체 안에 렌더링 되어 있는 오브젝트에 우선적으로 관심을 보이게 되어 전투에 전념.

그러나 이와 같은 시스템의 단점은 오브젝트들의 위치 값이기 때문에 게임도중에 계속 변하여 혼란을 줄 수 있다는 것이다. 따라서 오브젝트의 위치 값이 아닌 오브젝트의 수를 count하여 그 수를 화면에 표현 해주는 것이 혼란을 막는 방법

이 될 것이다. 이것은 화면에 보이는 절두체와는 전혀 별개로 나뉘며 시각불 형식이 아닌 그림 3에서와 같이 하나의 점을 중심으로 둠을 구성하여 그곳에 있는 모든 오브젝트 충돌을 감지해 내는 것이다.

그리고 게임 상에서 맵을 1, 2, 3, 4분면으로 나눌시 각 사분면 중앙으로부터 끝점까지의 반지름으로 하는 경계구를 이용하여 절두체 판정을 하는 것이다. 4분면의 절두체에 포함되지 않는 모든 지식 노드들을 판정에서 제외 시켜 지형을 빠르게 렌더링 시키며 제외된 다른 3개의 분면은 오브젝트 개수만을 체크 한다.

한편, 수식 1과 수식 2를 이용하여 플레이어 위치와 다른 오브젝트들과의 거리를 계산할 때 플레이어의 초기 위치값을 기준으로 계산하는 문제가 발생한다. 따라서 3.1절에서 언급한바와 마찬가지로 카메라 위치와 플레이어 위치를 일치시키기 위하여 초기 위치값을 -1로 세팅 한다. 이것은 플레이어 시야와 카메라 시야를 동일하게 하는 역할을 한다.

```

=====
length = pEnemy->CalcAI(&m_Player,
m_fElapsedTime);
a[i] = pEnemy->CalcAI(&m_Player,
m_fElapsedTime);
=====

```

여기서 a[i] 값은 행렬로 구현을 하여 상대 오브젝트의 범위 값을 각각 받아들이게 되고 a[i]값이 -1이면 오브젝트가 (0,0,0)에 위치한 상대 오브젝트를 따라가게 된다. 따라서 각각의 오브젝트들의 a[i]값을 체크하여 -1인 값들을 count하게 된다.

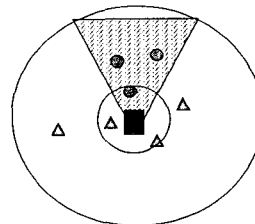


그림 3. 변형된 절두체 켤링을 이용한 오브젝트 탐색

그림 3에서 나타내고 있듯이 먼저, 플레이어가 위치한 곳에서 일정한 범위에 들어 있는 모든 오브젝트들에 대한 거리를 계산한다. 이때 플레이어가 주위에 있는 오브젝트들 중에서 절두체 내부에 위치한 오브젝트와의 거리보다 같거나 또는 더 가까운 거리에 위치한 오브젝트들을 검색하게 된

다. 그리고 검색된 오브젝트의 수량에 따라 플레이어가 느끼는 위협성은 변하게 된다. 뿐만 아니라 만약 절두체 안에 위치한 오브젝트의 거리 보다 더 가까운 곳에 위치한 오브젝트가 검색 되면 그 위협성의 수위는 최고에 달하게 된다.

#### IV. 실험 및 토의

본 논문에서 제안한 방법은 PC에서 Visual C++로 구현하여 실험 하였다. 오브젝트의 탐색을 위하여 플레이어 오브젝트를 제외한 상대 오브젝트의 개수를 모두 8개로 제한하였다. 왜냐 하면 실험을 위해 간단한 게임을 먼저 제작하는 과정에서 렌더링 되는 모든 폴리곤 및 오브젝트들의 생성 시간의 부담을 덜어주기 위함이다. 슈팅 게임뿐만 아니라 모든 3D 게임은 이러한 폴리곤 생성에 많은 문제점들이 있기 때문이다. 그리고 현재 완전한 게임을 제작하지 못한 관계로 게임의 인터페이스를 형성하지 못했다. 따라서 논문에서 제안한 알고리즘만을 이용하여 오브젝트 탐색에 대한 정확도만을 제시한다.

표 1은 본 논문에서 제안된 방법을 통하여 탐색되어진 상대 오브젝트의 탐색 정확도를 시간에 따라 나타낸다. 시간은 1분 간격으로 6분간 체크하여 정확성을 계산 하였다.

표 1. 오브젝트 탐색 정확도

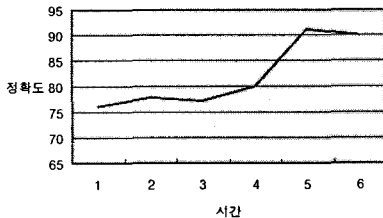


표 1에서 알 수 있듯이 처음 5분간은 게임진행에서 절두체 안에 오브젝트 및 폴리곤들을 생성하는 과정에서 상대 오브젝트의 탐색 시간이 지연되어 정확성이 다소 떨어지는 것을 알 수 있다. 그러나 점차 시간이 지날수록 상대 오브젝트들의 탐색 정확도가 높아지고 있음을 알 수 있다. 이것은 현재 FPS 게임에서 문제시 되고 있는 게임의 단순성을 해결할 수 있는 가능성을 제시하고 있다. 뿐만 아니라 전체적인 게임 인터페이스가 완성되면 정확도를 이용한 상대 오브젝트들의 위협성을 시각적으로 표현함으로써 게임의 긴박감과 스틸을 더하는 주요한 요소가 될 것이다.

#### V. 결론 및 향후 연구

본 논문에서는 변형된 절두체 쉐이딩을 이용하여

3D FPS 게임에서의 오브젝트를 탐색하는 방법을 제안 하였다. 제안된 방법을 게임 시뮬레이션에 적용한 결과, 상대 오브젝트들을 탐색하여 플레이어에게 위협성을 알려주는 역할을 하는데 어느 정도 만족할만한 결과를 얻을 수 있었다. 그러나 상대 오브젝트의 개수가 많아져서 계산 시간이 길어지는 경우와 플레이어들의 컴퓨터 사양에 따른 제약이 있을 수 있다. 따라서 향후 오브젝트들의 위치 정보 이외의 정보들을 이용하여 탐색 시간을 줄일 수 있는 연구가 진행될 것이다.

#### 참고문헌

- [1] Jin-Hyuk Hong, Sung-Bae Cho, "Evolution of emergent behaviors for shooting game characters in Robocode", Evolutionary Computation, CEC2004, Congress on Vol. 19-23, pp 19-23, 2004
- [2] 김용준, "3D 게임 프로그래밍", 한빛미디어, 2007
- [3] 주영호, "절두체 쉐이딩 기법과 Quad Tree, LOD 기법을 이용한 렌더링 속도 개선 방법", Technical Report REP05010, 그래픽스 응용 연구실, 부산대학교, 2005