

# 게임엔진을 이용한 3D 그래픽 렌더링 기술

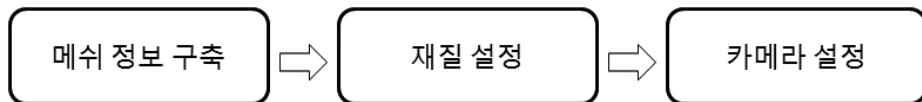
김성동 (계원예술대학교 디지털콘텐츠군)

## 1. 서론

컴퓨터 그래픽에서 렌더링이란 단어는 인간이 상상할 수 있는 객체나 장면을 표현하고 실제 세계에서 실현이 불가능한 것을 나타내기 위한 기술이다. 가상 객체 및 장면의 형상을 만드는 모델링 기술, 객체를 현실 세계에서 보이는 모습으로 보이게 하는 렌더링 기술, 정지 이미지들을 연속적으로 빠르게 재생함으로써 장면이 실제로 움직이는 것처럼 보이게 하는 애니메이션 기술을 적용하여 가상의 영상을 생성한다.

렌더링 기술은 3차원 CG 오브젝트들을 형상과 위치, 광원과 시점 정보에 의해 2차원 영상으로 변환하면서 모델링된 장면에 실재감을 부여하는 과정을 말한다. 렌더링에서의 영상 생성은 일반적으로 3단계를 거쳐 이루어 지는데, 첫 번째 단계는 3차원상의 점, 선, 면 등을 2차원 평면에 투영하는 단계이며, 두 번째 단계는 불투명 오브젝트의 면과 불투명 오브젝트 뒤에 있는 면을 차별화하여 앞에 보이는 면은 보이게 하고 앞의 물체에 가려지는 뒷면은 제거해야 한다.

세 번째 단계는 오브젝트의 보이는 면에 대해서 음영을 표시하게 된다. 3차원공간에서 빛과 환경요소를 고려해 물체를 사실적으로 표현한다는 뜻으로 주로 사용된다. 삼각형처럼 간단한 도형이라 할지라도 렌더링 과정을 통해 나타난다면 [그림 1]과 같이 과정을 거쳐야 한다[1,2,3].



[그림 1] 렌더링의 3단계 과정

렌더링 엔진은 2D나 3D 그래픽의 표현을 담당한다. 이 기술은 컴퓨터·비디오 게임에 있어 가장 주목을 받는 기술인만큼, 거의 모든 게임 엔진이 반드시 제공하는 기능이다. 지금과 같은 게임 엔진이 3D와 일인칭 슈팅 게임 시대부터 시작되었지만, 아마추어 개발자나 2D 그래픽에 대한 요구로 인해 2D 게임 엔진도 다수 등장했다[2].

1990년대 중후반에 폴리곤 객체를 표현하는 게임들이 등장한 이래로 렌더링 엔진은 많은 발전을 거쳤다[1,2]. 2007년의 게임 엔진은 하이 다이내믹 레인지 렌더링, 픽셀 셰이더, 정점 셰이더, 법선 맵핑, 피사계 심도, 레벨 오브 디테일, 광산란 시뮬레이션, 소프트 색도와 같은 기술이 지원되는 추세이다. 또한 오늘날의 게임 엔진은 3D 게임 세계를 객체 지향적으로 묘사하는 썬 그래프도 제공한다. 썬 그래프는 게임의 설계를 단순화시켜주고 거대한 가상 세계를 더욱 효율적으로 렌더링할 수 있게 해준다[1]. 어떤 엔진은 다른 필요한 기능을 모두 빼고



실시간 3D 그래픽스 렌더링 기능만을 제공하기도 한다. 이런 엔진은 게임 개발사가 나머지 기능을 스스로 작성하거나, 다른 미들웨어 구성 요소와 조합해야만 한다. 이런 형식의 엔진은 보통 ‘그래픽스 엔진’, ‘렌더링 엔진’으로 불린다. 그래픽스 엔진에는 렐로 포지, 오거, 크리스털 스페이스, 제네시스 3D, 유니티 등이 있다.

## 2. 지역조명 모델

지역 조명 모델은 빛의 작용을 계산함에 있어 사용자가 관심을 갖는 특정 표면과 이 표면을 직접 비추는 광원만을 고려하는 방식으로 광원, 표면, 시점간의 관계를 고려해서 난반사 및 정반사되는 빛의 세기를 계산하여 물체 표면의 색상을 구하는 모델로 많은 모델들이 개발되어 있으나 비교적 간단한 계산에 의해 사실에 근접한 효과를 낼 수 있는 Phong 모델과 Blinn 모델이 가장 널리 사용되고 있다.

### 가. Phong 모델

Phong 모델은 Phong이 개발한 가장 대표적이고 기본적인 지역조명 모델이다. 난반사의 계산은 Lambert의 코사인 법칙을 이용하고 정반사의 계산에는  $\cos\alpha$ 을 곱하는 방법을 이용하였다.  $\alpha$ 는 반사 벡터 R과 시점 벡터 V사이의 각도로서 이 방법을 사용하면 반사 벡터 R과 시점 벡터 V가 일치하는 지점을 최대값으로 해서 두 벡터 사이의 각도가 벌어질수록 정반사의 세기가 점점 줄어들게 된다. n은 사용자가 임의로 지정하는 숫자로서 하이라이트의 크기를 조절하는 역할을 한다[2,3]. Phong 모델은 물체 표면에서 일어나는 물리적인 현상을 기초로 한 것이 아니므로 실제 상황과 다소 차이가 날 수 밖에 없으나 상대적으로 계산량이 적고 구현이 간단하며 최종적으로 얻어지는 결과물이 상당히 유사하게 보이기 때문에 기본적인 조명모델로서 가장 널리 애용되고 있다.

### 나. Blinn 모델

Phong 모델보다 좀 더 사실적인 결과를 생성하기 위해 물리적으로 좀 더 올바른 조명 모델들을 개발하게 되는데 이와 같은 방법들이 바로 물리 기반 지역 조명 모델(physically based local illumination model)들이다. Blinn 모델은 최초로 개발된 대표적인 물리 기반 지역 조명 모델로서 정반사 특성을 계산함에 있어 Phong이 사용했던  $\cos\alpha$  대신에 Torrance-Sparrow 모델을 사용한다. Blinn 모델이 Phong 모델과 비교해서 갖게 되는 실질적인 차이점은 정반사를 계산할 때 조명의 입사각에 따라 정반사 특성이 바뀌느냐 안 바뀌느냐에 있다. 즉 Phong 모델에서는 입사각과 관계없이 항상 반사각방향으로 정반사율이 높게 나오지만 실제 세계에서는 빛이 비스듬하게 입사할 경우 반사각보다 더 비스듬한 방향으로 정반사율이 높게 나오게 되는 off specular peak 현상이 일어나는데 정오에 태양이 지면에서 거의 수직해 있을 때는 하이라이트가 별로 발생하지 않던 아스팔트 표면이 석양 무렵 태양 빛이 지면에 비스듬하게 입사할 때에는 훨씬 반짝여 보이는 것이 바로 이 현상

때문이다. [그림 2]에서와 같이 Blinn 모델은 off-specular peak 현상을 제대로 표현할 수 있으며, 조명이 뒤에서 비춰지는 경우에도 Phong 모델보다 사실적인 결과를 생성할 수 있다. Blinn 모델을 비롯한 대부분의 물리 기반 지역 조명 모델들은 올바른 하이라이트를 얻기 위해 정반사 특성의 계산에만 이와 같은 방법을 적용할 뿐 난반사 특성은 Phong에서 사용했던 Lambert의 코사인법칙을 그대로 사용하지만, 최근에는 Oren-Nayer 모델 같이 난반사 특성을 계산하기 위한 모델들도 개발되고 있다.

### 3. 전역조명 모델

지역 조명 모델만 가지고는 오브젝트간의 반사나 굴절, 그림자 등과 같이 다른 오브젝트에 의해 반사되거나 다른 오브젝트를 투과, 굴절해서 오브젝트 표면에 영향을 미치는 빛에 의한 결과들을 재현할 수 없다는 문제가 있다. 이러한 문제를 해결하기 위하여 개발된 전역 조명 모델은 렌더링 과정에서 오브젝트 상호간의 관계도 모두 고려하기 때문에 오브젝트들 간의 상호 반사(inter reflection), 굴절, 그림자 효과 등을 재현할 수 있어 훨씬 사실적인 이미지를 생성할 수 있으나 엄청난 렌더링 시간이 걸린다는 치명적인 단점을 가지고 있다. 최근에는 하드웨어 환경이 급속히 좋아지는데다 속도 개선을 위한 많은 방법들이 개발되어 활용이 점점 늘어나는 추세이다[1,2,3].

#### 가. Sub Surface Scattering

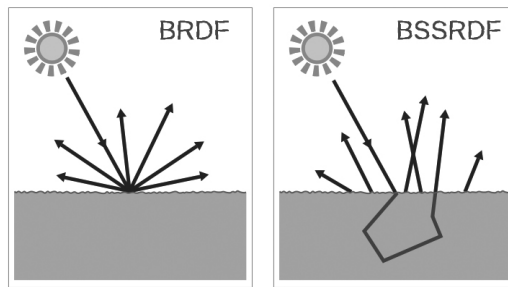
라디오시티(Radiosity)와 유사한 빛 처리 과정을 수행하나 오브젝트의 재질까지 함께 고려한 방식이다. 일반적인 라디오시티 알고리즘은 동일한 정반사와 난반사 수치를 갖고 있으면 어떤 물질이라도 같은 방식으로 계산되나, sub surface scattering은 물질의 반투명성과 정확한 각도의 빛 반사를 계산하는 것으로 BSSRDF에 기반한 반투명 재질에 대한 사실적인시뮬레이션 결과를 얻을 수 있다[2,3]. [그림 2]은 각각 BRDF(Bidirectional Reflectance Distribution Function)와 BSSRDF(Bidirectional Sub Surface Scattering Reflectance Distribution Function)를 이용한 얼굴 렌더링 결과를 나타내며 BRDF는 BSSRDF 모델에서 sub surface scattering을 고려하지 않은 간략화된 모델이다. Sub surface scattering은 현재 MentalRay, Brazil Renderer, V-Ray 등에서 기능을 제공하고 있다.



[그림 2] BRDF을 이용한 결과[왼쪽], BSSRDF을 이용한 결과[오른쪽][6]

## 나. BRDF

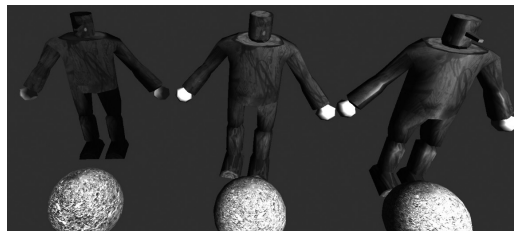
[그림 3]에서처럼 반사(Reflection)는 어떤 한 표면의 한 쪽 면으로 들어오는 빛이 같은 쪽 면으로 반사되어 나가는 현상을 말한다.(들어오는 면과 반대쪽 면으로 나가는 현상을 투과(Transmission)라 한다.) 또한, 투과, 흡수, 분광, 편광, 형광, 인광 효과 역시 표면의 재질과 빛이 상호작용하는 것을 모델링하는데 중요한 역할을 한다. 그러나 여기서는 이것들을 다루지 않고 반사를 만족시키는 일반적인 속성에 대해서만 다룬다. 사실 우리의 눈이 사물을 볼 수 있는 것은 대부분 반사된 빛에 의한 것이다. 또한 반사를 제외한 위의 각 효과들은 독립적으로 개발해서 전체 조명계산 알고리즘에 통합시킬 수 있다.



[그림 3] BRDF와 BSSRDF의 기본 Diagram[4]

## 다. BSSRDF

물체와 빛의 상호작용을 구현하는데 사용되는 알고리즘 중 일반적인 물체를 묘사하는데 가장 널리 사용되는 방법이 BRDF이다. 이 방법은 물체묘사에 있어서 빛과 상호작용하는 물체의 부분을 표면에 한 전하고 있다. 대부분의 재질이 불투명하고 치밀한 재질(금속)에 대하여는 이러한 접근방식이 유효하였으나 유기체(피부)를 포함한 반투명 혹은 투명한 일반적 재질의 경우, 실제로 물체와 상호작용은 표면 뿐 아니라 표면과 근방의 부피 내에서 이루어지게 된다. 그 결과 물체 표면 하에서의 빛의 상호작용을 포함하는 BSSRDF의 필요성이 제기되었다[7].



[그림 4] 3D 유니티 엔진을 이용한 렌더링: (a)Ambient 표현 (b)specular표현 (c) diffuse 표현

## 5. 결론

렌더링 기법 연구와 개발이 진행되고 있는 것이 사실이다. 렌더링 과정은 수작업으로 성능 개선이 거의 불가능하여 사용자들이 사용하는 렌더러 기능이나 렌더링 프로개발에 의해 렌더링 품질이 좌우되고 있다. 현재 널리 사용되고 있는 상용 렌더러들은 실시간과 고품질의 두 가지 목적을 한꺼번에 얻기 위한 노력을 계속하고 있으며 하드웨어 성능의 발전과 함께 렌더링 속도도 급속히 빨라지고 있다. 그러나 실제와 구별할 수 없을 수준의 사실적 영상을 생성하기 위해서는 아직도 상당한 렌더링 시간을 필요로 하고 있다. 본문에서 언급한 BSSRDF는 부가적으로 해당 물체의 재질을 더 고려한 radiosity의 결과지이다. BRDF는 빛이 물체 표면 한 점에 도달한 후 같은 지점에서 반사된다고 계산하는데 반해, BSSRDF는 다른 지점에서 반사하는 것으로 연산한다. 그러므로 BSSRDF로 BRDF보다 훨씬 사실적인 결과를 얻을 수 있다. 앞으로 게임엔진에서 사실적 묘사의 끝은 어디일까?

---

### 참고문헌

- [1] 위키백과 [http://ko.wikipedia.org/wiki/%EA%B2%8C%EC%9E%84\\_%EC%97%94%EC%A7%84](http://ko.wikipedia.org/wiki/%EA%B2%8C%EC%9E%84_%EC%97%94%EC%A7%84)
- [2] 전자통신동향분석 제20권 제6호 2005년 12월, 한국전자통신연구소
- [3] 이득우저, 유니티게임개발의 정석 에이콘 출판사
- [4] [http://en.wikipedia.org/wiki/Bidirectional\\_reflectance\\_distribution\\_function](http://en.wikipedia.org/wiki/Bidirectional_reflectance_distribution_function)
- [5] 김성훈역 게임프로그래밍 정석, 2012.4 한빛미디어
- [6] Henrik Wann Jensen, Steve Marschner, Marc Levoy and Pat Hanrahan, "A Practical Model for Subsurface Light Transport", Stanford University, SIGGRAPH 2001
- [7] 안우철, 고흥석, 이상욱, "안면 피부의 BSSRDF의 측정 및 응용" 한국컴퓨터그래픽스학회 2002

김성동 (sdkim@kaywon.ac.kr)

계원예술대학교 디지털콘텐츠교수