

멀티미디어 (Graphic)

2024.

목차

1. 컴퓨터 그래픽스의 개요
2. 2D 그래픽과 3D 그래픽
3. 조명과 그래픽스
4. 3D 모델 표현의 표현 방식
5. 3D 컴퓨터 그래픽스의 제작 과정

1.1 컴퓨터 그래픽스의 개요

■ 컴퓨터 그래픽스(CG, Computer Graphics)의 개념과 활용분야

- 개념 : 컴퓨터를 사용하여 제작된 이미지와 그들을 구현하기 위해 필요한 기술을 통합하여 부르는 말
- 컴퓨터로 만들어진 그래픽 이미지는 CMYK 컬러 모델과 RGB 컬러 모델 형식으로 표현됨
 - CMYK 컬러 모델 : 결과물을 인쇄 매체 형식으로 출력(카탈로그, 브로슈어, 신문, 서적, 그림 등)
 - RGB 컬러 모델 : 결과물을 영상 매체 형식으로 출력(웹디자인, 아이콘 디자인, 영상 합성 등)

영화 아바타 : <https://youtu.be/kihrFxwdMb4?si=eu8q3VAme9Fwfx35>



▲ 컴퓨터 그래픽스의 세계

1.1 컴퓨터 그래픽스의 개요

■ 컴퓨터 그래픽스(CG, Computer Graphics)의 개념과 활용분야

- 상업적인 용도
 - 영화, 방송, CF, 애니메이션 등과 같은 분야에 광범위하게 이용
 - 특히 시각적인 특수 효과를 내는 비주얼 이펙트(VFX, Visual Effects) 분야가 많은 발전을 함
- 창작 예술 분야
 - 회화, 디자인
- 기타 분야
 - 시뮬레이션(Simulation), CAD(Computer Aided Design), 의료, 과학, 군사, 교육



▲ 비주얼 이펙트

https://youtu.be/8a2WDAZ_TPc?si=4bGPz_Fbdmm1urVL

▶ 동영상 보기 : [시뮬레이션, 비행방송](#)

1.2 그래픽 이미지의 기본 요소

■ 컴퓨터 그래픽스의 이미지 표현 방식

- 벡터 그래픽(Vector Graphics) 방식과 래스터 그래픽(Raster Graphics) 방식을 사용
 - 벡터 그래픽 방식 : 점, 선분, 곡선, 다각형 등을 이용하여 복잡한 이미지를 표현
 - 래스터 그래픽 방식 : 이미지 표면을 형성하는 각각의 픽셀에 빛과 색상을 표현
- 두 가지 방식을 혼합하면 이미지를 보다 입체적으로 표현할 수 있음
- 컴퓨터의 디스플레이 장치는 벡터 방식의 이미지를 래스터 방식으로 변환하여 출력함
- 그래픽스 소프트웨어는 선분, 다각형, 문자열 등과 같은 기본 요소를 가지고 이미지를 생성함

▶ 동영상 보기 : [벡터 그래픽 방식과 래스터 그래픽 방식의 차이점](#)



(a) 다각형



(b) 그래픽 패턴

■ 그래픽 이미지를 구성하는 기본 요소

- 선분(Line Segment)

- 두 점을 이어주는 직선
- 다각형을 구성하거나 색상이나 패턴으로 채워질 영역의 경계를 정의하기 위해 사용

- 다각형(Polygon)

- 세 개 이상의 선분으로 둘러싸인 도형으로 하나의 면을 나타냄
- 다각형은 3차원의 입체적인 모양을 표현하기 위한 가장 작은 단위
- 다각형의 개수가 많을수록 실제 모양에 근접한 형태로 표현됨
- 다각형은 화면에 디스플레이(Display)하기가 용이함
- 그래픽스 소프트웨어의 성능은 1초에 디스플레이 되는 다각형의 개수로 측정됨

1.2 그래픽 이미지의 기본 요소

■ 그래픽 이미지를 구성하는 기본 요소

• 그래픽 문자열(Graphic String)

➤ 그래픽 문자열은 폰트(서체), 크기, 색상과 같은 매개변수를 조절하여 다양한 형태로 표현한 문자열

✓ 벡터 방식의 문자열 : 문자의 외곽선을 나타내기 위해 점, 선분, 곡선 등을 사용하는 방식

원하는 크기와 형태로 변환할 수 있음

✓ 래스터 문자열 : 문자를 표현하기 위해 사각형의 비트 블록에 0과 1의 조합으로 채워 넣는 방식

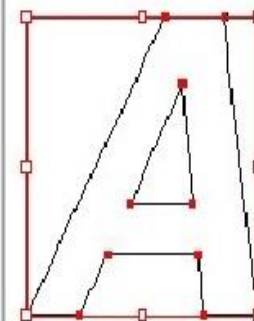
확대하면 문자의 외곽선이 계단 모양으로 깨짐

안티앨리어싱 기법을 사용하여 해결

Raster Graphic



Vector Graphic



▲ 벡터 문자열과 래스터 문자열의 비교

2.1 2D 그래픽

■ 2D 그래픽

- 2차원 평면에 구현된 이미지
- 래스터 그래픽스 방식을 사용하다가 벡터 그래픽스 방식으로 발전함
- 제작 소프트웨어로는 드로잉, 페인팅, 이미지 프로세싱, 2D 애니메이션 같은 프로그램을 사용
- 초기 2D 그래픽은 단순하고 기하학적인 추상물을 표현하거나 그림 합성, 변형 등에 사용됨
- 그래픽 처리 장치(GPU, Graphics Processing Unit)가 보편화되면서 보다 정교하고 사실적인 표현이 가능해짐
- 현재는 가정용 비디오 게임이나 영화의 특수영상 제작에도 사용됨



▲ 2D 그래픽과 3D 그래픽의 비교

■ 3D 그래픽

- 해당 모델의 기하학적 데이터를 (X,Y,Z)축의 공간 좌표를 이용하여 3차원으로 표현한 후 2차원 화면에 나타냄
- 대상의 빛, 농도, 원근, 재질 등 많은 속성의 상호 작용을 고려하여 2차원의 표면을 사실적으로 표현해야 함
- 최근 영화, 애니메이션, 광고와 같은 엔터테인먼트 분야와 시뮬레이션, 건축, 디자인 등의 설계에 다양하게 사용됨

■ 3D 그래픽 제작 순서

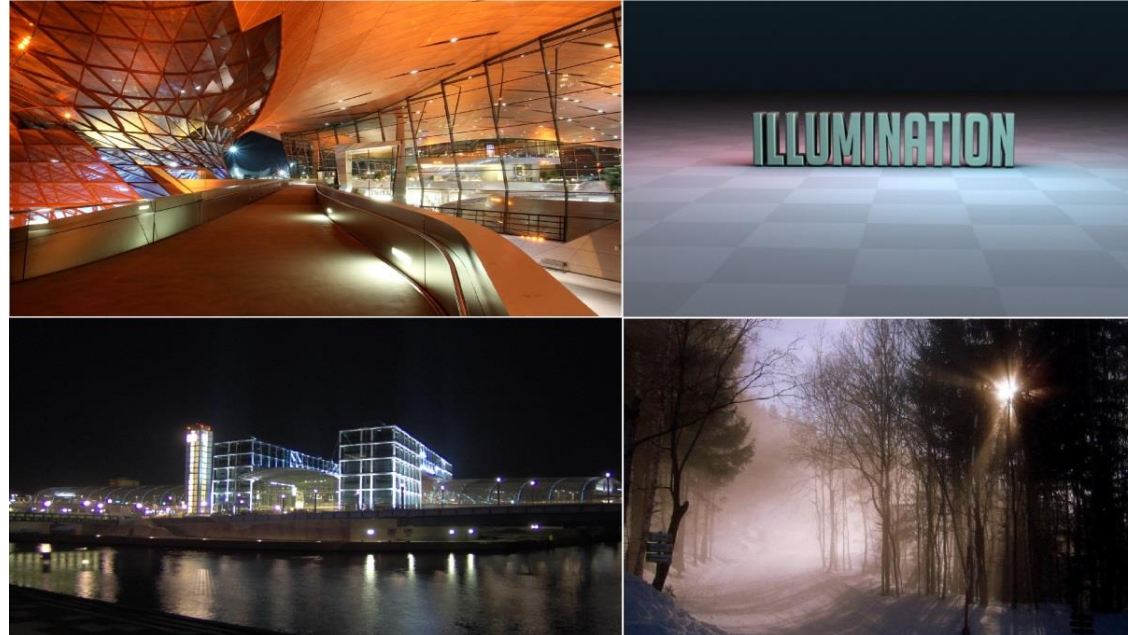
1. 각종 캐릭터, 사물, 배경 등의 대상을 3차원으로 형상화하는 모델링(Modeling) 작업을 함
2. 각각의 장면(또는 물체)에 조명을 비추어 음영처리(Shading)를 하고 질감 및 기타 효과를 넣음
3. 여러 명령으로 연산 처리를 하여 최종적인 그래픽 이미지를 출력하는 렌더링(Rendering) 과정을 거침

<https://youtu.be/cvcAjgMUPUA?si=xpcMxYgD7nabvICu>

3.1 조명과 그래픽스

■ 조명의 개요(Illumination)

- 전체적인 느낌과 분위기를 표현하기 위하여 **인위적으로 사용하는 광선**
- 대상의 **원근감과 입체감을 표현**하거나 등장인물이나 배경을 밝히기 위하여 사용
- 기본 조명은 **주광선(Key Light)**, **보조 광선(Fill Light)**, **후면 광선(Back Light)**으로 구성된 3점 조명임
- 조명이 추가로 필요할 경우 배경 광선(Background Light)을 사용하여 4점 조명을 구성함



▲ 컴퓨터 그래픽스에서 조명의 역할

3.1 조명과 그래픽스

■ 3점 조명

- 주광선 : 앞쪽에서 피사체를 비추기 위해 기본으로 사용하는 광선

피사체에 그림자를 만듦

주광선의 세기, 색깔, 각도에 따라 피사체의 형태, 표면 질감 등 물체의 전반적인 모습이 결정됨

실내의 주광선 : 전등, 카메라 플래시 가 되고 실외에서는 태양

- 보조 광선 : 물체를 측면에서 비추는 광선

주광선에 의해 생긴 그림자를 비추어 제거, 명암의 대조를 감소시켜 물체를 부드럽게 보임

- 후면 광선 : 주광선의 반대편에서 비춰지는 역광선을 의미함

화면의 평면화를 방지하여 피사체의 입체감을 더함

→ 조명이 추가로 필요할 경우 배경 광선(Background Light)을 사용하여 4점 조명을 구성함

▶ 동영상 보기 : [How To Set Up 3-Point Lighting for Film, Video and Photography](#)

▶ 동영상 보기 : [3점 조명\(3-Point Lighting\)](#)

▶ 동영상 보기 : [4점 조명\(4-Point Lighting\)](#)

3.2 조명의 종류

■ 부분 광원(Area Light)

- 장면을 강조하거나 효과를 극대화하기 위해 **일부분을 비추는 조명**, 부분 조명
- 스포트라이트가 대표적

■ 방향성 광원(Directional Light)

- 태양광처럼 **일정한 방향**으로 동일한 광선량을 투사하는 조명
- 그림자를 만들고 뚜렷한 입체감과 질감 표현에 유리

■ 전방향성 광원(Point Light)

- **모든 방향**으로 빛을 비추는 조명

■ 스포트라이트 광원(Spot Light)

- 스포트라이트라고 알려진 광원으로 **무대 조명**에 많이 사용
- 주인공을 강조하거나 현실감을 최대한 나타내기 위해 원추 모양의 빛을 비춤

■ 분위기 광원(Ambient Light)

- 방향과 거리가 없는 가상의 조명
- 주변에 위치한 여러 종류의 광원에 의한 빛의 반사와 산란 등으로 발생한 광원

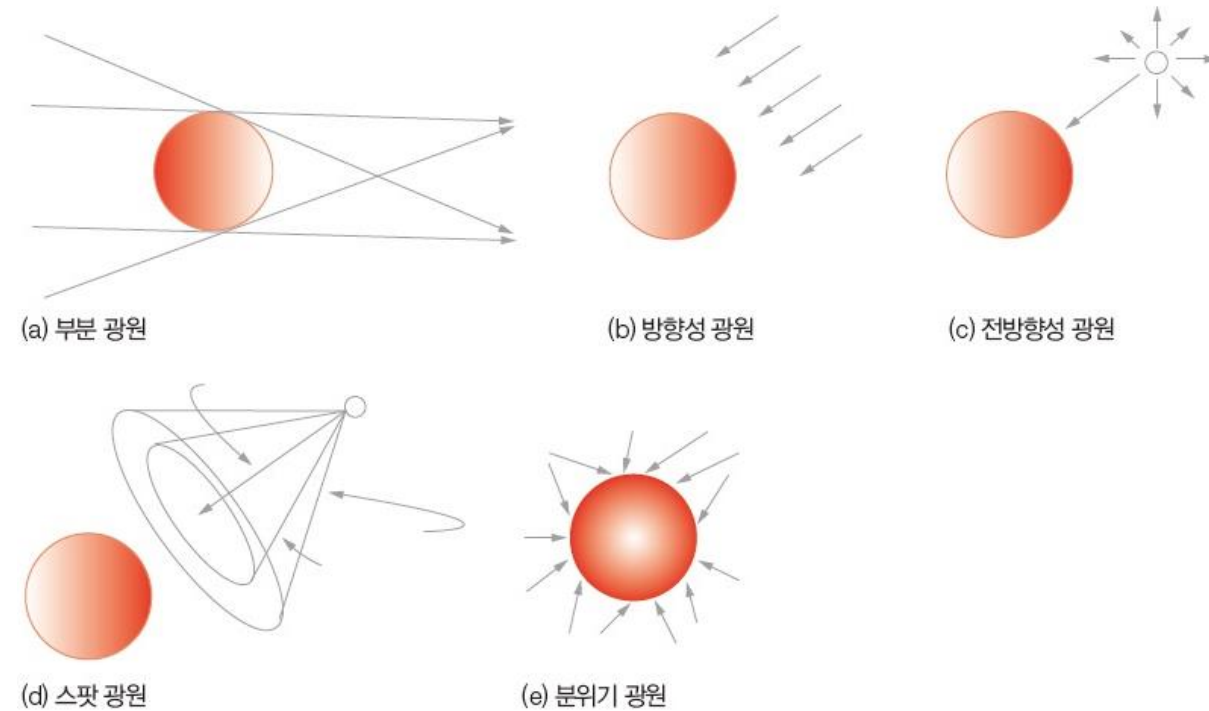


그림 6-8 조명의 종류와 특징

3.2 조명의 종류

■ 볼륨 광원(Volume Light)

- 공간 속의 물체 자체가 조명 역할을 하는 광원

■ 효과 광원(Effects Light)

- 레이저광과 안개 효과를 표현하는 포그(Fog) 조명 등이 있음



그림 6-9 볼륨 광원과 효과 광원 [04]

4.1 3D 모델 표면의 표현 방식

■ 3D 모델 표면의 표현 방식

- 폴리곤 방식
- 넵스 방식
- 패치 방식

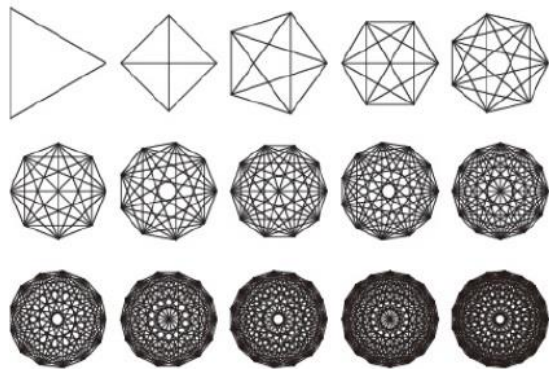
■ 폴리곤(Polygon) 방식

- 3D 객체(Object) 또는 모델을 구성하는 모든 표면을 삼각형 또는 그 이상의 다각형으로 구성하는 방식
- 수많은 폴리곤이 모여 3차원 형상을 표현
- 폴리곤의 크기를 감소시키고 개수를 늘리면 대상을 좀 더 세밀하고 현실에 가깝게 표현할 수 있음
- 계산 절차가 간단하고, 모델을 구성하는 표면의 개수를 원하는 만큼 조절할 수 있기 때문에 렌더링 속도가 빠름
- 초기 컴퓨터에서 글꼴을 제작하기 위해 사용
- 현재는 3차원 컴퓨터 그래픽의 기본 단위로 사용됨
- 삼각형 폴리곤을 기본으로 사용(다른 폴리곤에 비하여 계산이 쉽고 연산 처리 속도가 빠르기 때문)
- 폴리곤을 사용하는 대표적인 프로그램으로 3D Max가 있음

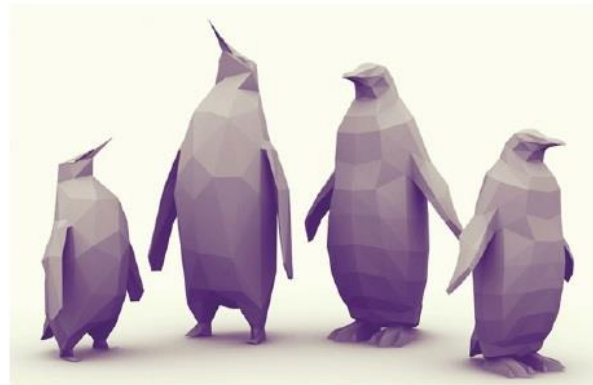
4.1 3D 모델 표면의 표현 방식

■폴리곤 방식의 특징

- 모델의 표면을 3차원 공간 좌표인 (X, Y, Z)로 표현
- 표면을 추가하거나 변형하기 쉬움
- 폴리곤의 변형 여부를 판단하는 논리 연산(Boolean)이 가능(합집합, 교집합, 차집합 등)
- 유선의 곡면은 폴리곤을 분할하여 다수의 작은 폴리곤으로 처리함
- 곡면을 확대하면 평면이 됨
- 폴리곤 방식으로 표현된 3차원 형상을 3D 객체라고 부름(이 객체는 점, 선분, 메시로 구성되어 있음)



(a) 다양한 형태의 폴리곤

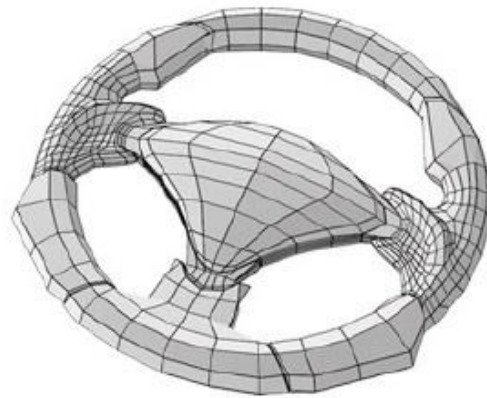


(b) 폴리곤을 사용하여 만든 3차원 객체

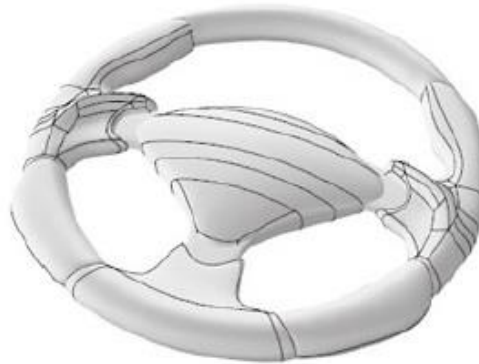
4.2 넵스 방식

■넵스(NURBS, Non-Uniform Rational B-Spline) 방식

- 폴리곤 형식의 단점인 테두리가 꺾여 표현되어 모델의 표면이 부드럽지 못한 것을 개선하기 위해 개발
- 적은 개수의 제어점으로 매끄러운 곡선을 표현하는 스플라인(B-Spline)을 사용함
- 3차원의 입체 곡면이나 덩어리까지 매우 정밀하고 정확하게 표현할 수 있음
- 표면을 확대해도 곡면이 그대로 유지됨
- 대표적인 프로그램으로 마야(Maya), 블렌더(Blender) 등이 있음
- 처리 속도가 폴리곤보다 느린 단점이 있지만 완벽한 유선형 객체를 표현할 수 있음



(a) 폴리곤 방식



(b) 넵스 방식

4.2 녁스 방식

■ 녁스 방식의 특징

- 곡면을 제어하는 조절점을 추가하면 가로 또는 세로줄의 표면에 모두 조절점이 2배로 증가
- 조절점과 조절점을 계산하여 연결하기 때문에 처리 시간이 많이 걸림
- 논리(Boolean) 연산과 테두리를 손질하는 트림(Trim) 연산이 가능
- 폴리곤 모델로 전환이 가능하기 때문에 게임 개발에서는 폴리곤 모델로 전환해서 사용함

■ 패치(Patch Modeling) 방식

- 폴리곤 방식과 녁스 방식의 장점을 결합한 모델링 방법
- 다양한 패치들을 연속적으로 표면에 연결하면서 결합시켜 3D 객체 또는 모형을 표현
- 패치는 3D 객체를 구성하는 작은 조각(Fragment)이며, 스플라인 곡면, 다수의 삼각형, 사각형 폴리곤들을 사용하여 구성

4.2 패치 방식

■ 패치 방식의 특징

- 표면의 곡선은 넙스 모델만큼 부드러운 유선형으로 표현됨
- 조절점 사이를 단순한 곡선으로 표현하고 단계값으로 곡면을 조절
- 표면을 나누면 가로 방향 또는 세로 방향의 모든 표면이 2배로 증가
- 논리(Boolean) 연산과 테두리를 손질하는 트림(Trim) 연산이 불가능
- 패치 모델은 폴리곤이나 넙스로 전환이 가능
- 렌더링 속도는 넙스 모델보다 빠르고 폴리곤 모델보다 느림
- 게임에서는 패치 모델을 폴리곤 모델로 변환하여 사용

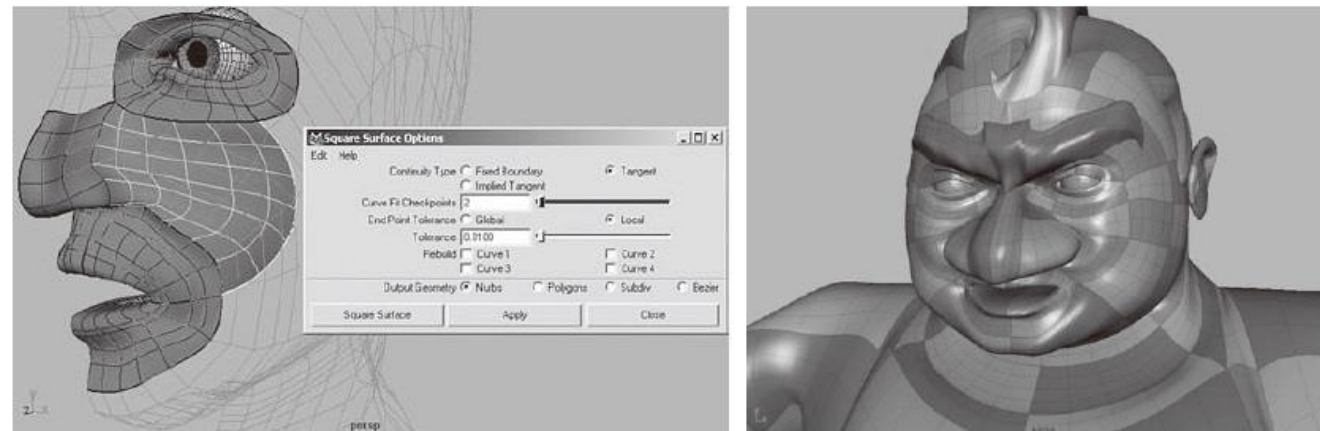


그림 6-12 패치 방식으로 표현한 모델의 표면 [07]

5.1 3D 컴퓨터 그래픽스의 제작 과정

■ 3차원 그래픽스의 제작 과정

- 모델링(Modeling) : 대상 모델의 기하학적인 형상을 만드는 과정
- 투영(Projection) : 3차원 형태의 그래픽 이미지를 2차원의 평면에 표현하는 과정
- 3D 매핑(3D Mapping) : 모델 표면의 색상, 재질, 패팅 등을 표현하는 과정
- 셰이딩(Shading) : 조명의 위치를 조정하여 모델 표면에 색상과 명암을 적용하는 과정
- 렌더링(Rendering) : 최종적인 그래픽 이미지를 출력하는 과정

→ 마지막 렌더링 과정에서 모델에 동작을 입력하면 애니메이션이 됨

▶ 동영상 보기 : [Modeling, Texturing & Rendering](#)



그림 6-13 컴퓨터 그래픽스의 제작 과정 [08]

5.2 모델링

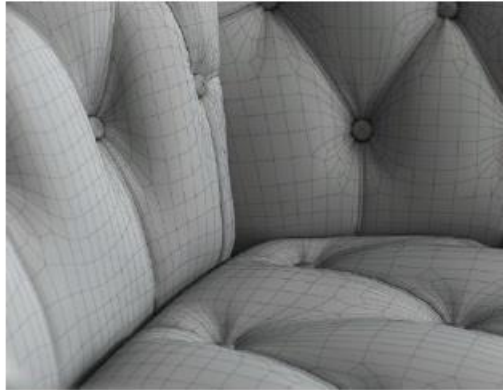
■ 모델링 방법

- 로프트 모델링(Loft Modeling)
 - 2차원 대상물에 높이 값을 입력하여 3차원 모델을 만드는 방법
 - 모델의 단면은 기본 도형을 사용하여 표현하고, 기본 도형에 직선 및 곡선 경로의 높이를 입력함
- 레이드 모델링(Lathe Modeling)
 - 도자기를 만드는 것처럼 회전축을 중심으로 회전시켜 3D 모델을 만드는 방법
- 프리미티브 모델링(Primitive Modeling)
 - 구슬, 직육면체, 원통, 원뿔 등과 같은 3차원의 기본 도형을 사용해서 모델링 하는 방식
 - 모델링 과정이 비교적 단순하여 일반적으로 많이 사용되는 방식
- 패치 모델링(Patch Modeling)
 - 사각형의 패치를 연속적으로 이어 붙여서 모델의 표면 형태를 만드는 방식
 - 가장 많은 제작 기간이 소요됨
- 스캔 모델링(Scan Modeling)
 - 3차원 스캐너를 사용하여 모델링하는 방법
 - 단순히 3차원 좌표를 측정하던 과거와 달리 현재는 모델의 질감(Texture)까지 측정할 수 있음

▶ 동영상 보기 : [모바일용 3D 스캐너](#)

5.2 모델링

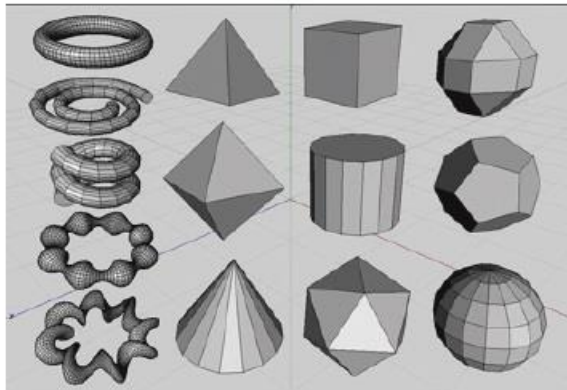
■ 모델링 방법



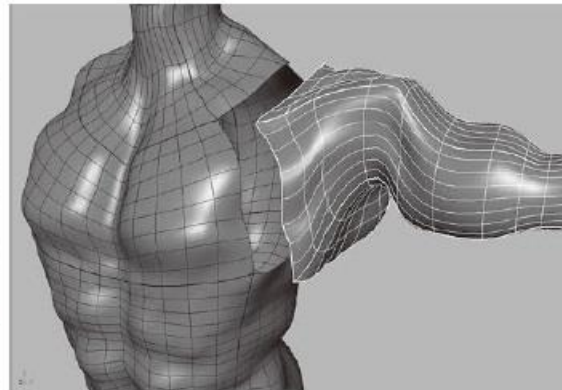
(a) 로프트 모델링



(b) 레이드 모델링



(c) 프리미티브 모델링



(d) 패치 모델링

- ▶ 동영상 보기 : [로프트 모델링\(Loft Modeling\)](#)
- ▶ 동영상 보기 : [레이드 모델링\(Lathe Modeling\)](#)
- ▶ 동영상 보기 : [프리미티브 모델링\(Primitive Modeling\)](#)
- ▶ 동영상 보기 : [Spline Patch Modeling in Modo - Pt. 4](#)
- ▶ 동영상 보기 : [Intro to Patch modeling in Max](#)
- ▶ 동영상 보기 : [스캔 모델링\(Scan Modeling\)](#)

■ 입체적인 모델의 형태를 표현하는 방법

- 와이어프레임 모델(Wire-Frame Model)

- 모델의 표면을 삼각형 또는 사각형으로 분할하고 점, 직선, 곡선 등으로 연결하여 뼈대를 만든 모델
- 복잡한 구성체라도 간단하게 표현할 수 있음
- 데이터 양이 적어 모델링 처리 과정도 빠름
- 보이지 않아야 할 모델의 뒷부분(Hidden Line)까지 보이는 단점이 있음

▶ 동영상 보기 : [와이어프레임 모델\(Wire-Frame Model\)](#)

- 서피스 모델(Surface Model)

- 와이어프레임 모델 형태에 면을 씌워 놓은 형태로 표면 모델이라고도 함
- 데이터의 양은 와이어프레임 모델보다 많지만 솔리드 모델보다는 적기 때문에 고속으로 렌더링 할 수 있음
- 게임에서는 모두 서피스 모델을 사용함

▶ 동영상 보기 : [서피스 모델\(Surface Model\)](#)

■ 입체적인 모델의 형태를 표현하는 방법

- 솔리드 모델(Solid Model)
 - 벽돌과 같은 3차원 도형을 사용해서 모델 내부를 채우는 방식
 - 덩어리 형태로 입체적인 모양을 구현하기 때문에 3D 객체의 질량, 부피, 강도 등과 같은 정보를 계산할 수 있음
 - 대량의 데이터로 인해 작업 속도가 느리기 때문에 게임에서는 사용하지 않음

▶ 동영상 보기 : [솔리드 모델\(Solid Model\)](#)

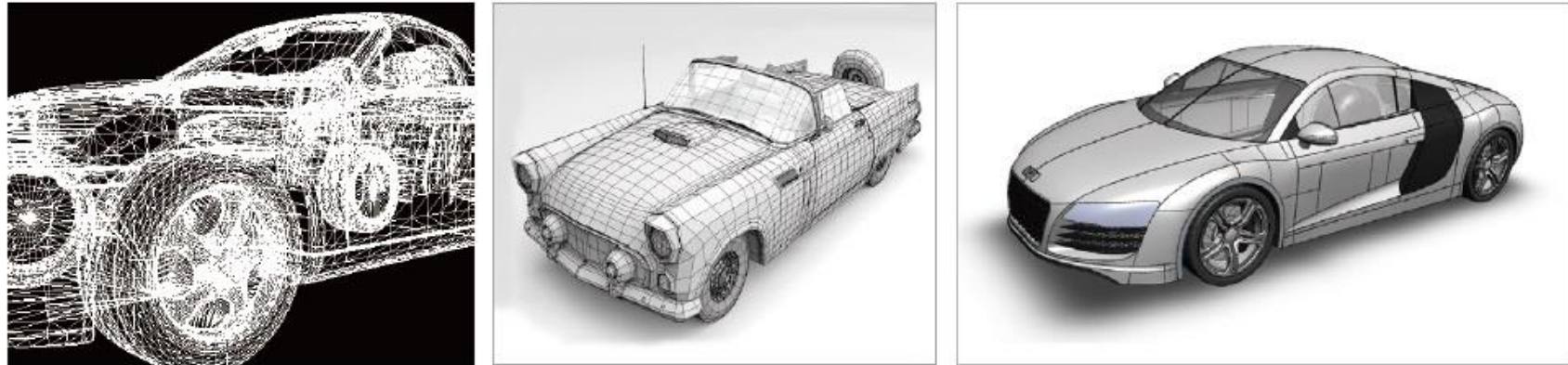


그림 6-15 와이어프레임 모델, 서피스 모델, 솔리드 모델 [10]

5.2 모델링

■ 입체적인 모델의 형태를 표현하는 방법

- 파라메트릭 모델(Parametric Model)
 - 곡선을 표현하는 데 유용한 방식
 - 모델 표면의 패치(Patch)에서 자유 곡선을 표현하기 위한 각 점들을 제어점으로 설정
 - 제어점에 치수 등과 같은 파라미터를 입력하여 점과 점을 연결하는 선분을 부드러운 곡선 형태로 표현함
- 프랙털 모델(Fractal Model)
 - 삼각형과 같은 기본 도형을 중심으로 기존 도형들을 계속 연결하여 더 복잡하고 섬세한 모델을 제작함
 - 기존 모델링 방식으로 표현하기 어려운 산이나 구름 같은 불규칙적인 대상물을 표현하는 데 유용한 기법



그림 6-16 파라메트릭 모델, 프랙털 모델 [11]

- ▶ 동영상 보기 : [파라메트릭 모델\(Parametric Model\)](#)
- ▶ 동영상 보기 : [Fractal of DLA model](#)
- ▶ 동영상 보기 : [Fractal Zoom Mandelbrot Corner](#)

5.3 투영과 은면 제거

■ 투영법(Projection Method)

- 평행 투영법
 - 투영의 중심을 무한한 원점으로 생각했을 때 투영 방법
 - 투영선은 모두 평행을 이루고 있음
 - 주로 설계도면에 사용됨
- 투시 투영법
 - 투영점에서의 거리에 따라 물체의 크기를 변화시키는 기법으로 원근 투영법이라고도 함
 - 주로 건축도면에 사용됨
 - 3D 객체를 좀 더 자연스럽게 표현할 수 있음

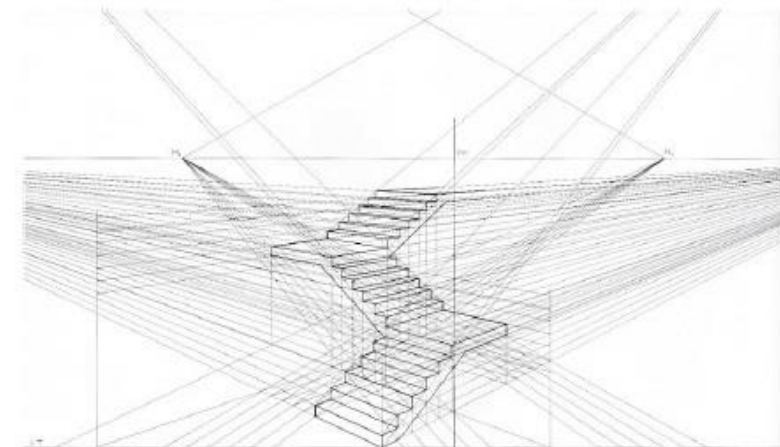


그림 6-17 3D 모형의 투영 과정 [12]

5.3 투영과 은면 제거

■ 은면 제거

- 객체의 표면은 보는 위치에 따라 보이지 않는 표면이 있는데, 이 표면을 은면(Hidden Surface)이라고 함
- 은면 제거 기법으로는 Z 정렬법, Z 버퍼법, 주사선(Scan Line)법, 광선 투사법 등이 있음
- 다각형 표면단위로 제거할 수도 있고 다각형의 표면을 구성하는 모든 픽셀 단위로 제거할 수도 있음
- 은면 제거 기법들은 각각 계산 시간과 렌더링 과정에서 화질의 차이가 발생함

▶ 동영상 보기 : [은면\(Hidden Surface\) 제거](#)

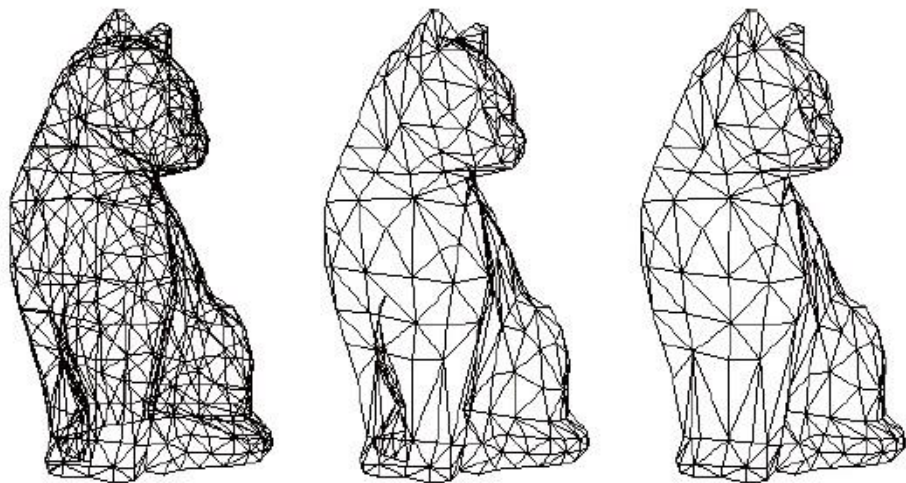


그림 6-18 3D 모형의 은면 제거 [13]

■ 3D 매핑

- 2차원 이미지를 굴곡이 있는 3차원의 입체적인 표면으로 옮겨 표현하는 작업
- 물체가 가지고 있는 고유의 재질, 표면의 색상, 패턴 등을 표현하는 과정
- 3차원 객체의 표면에 원하는 무늬, 색상, 질감, 또는 사진이나 그림 등의 2차원 이미지를 붙여 사실감을 향상시킴
- 텍스처 매핑, 범프 매핑, 디스플레이스먼트 매핑, 오퍼시티 매핑, 리플렉션 매핑, 라이트 매핑 등이 있음

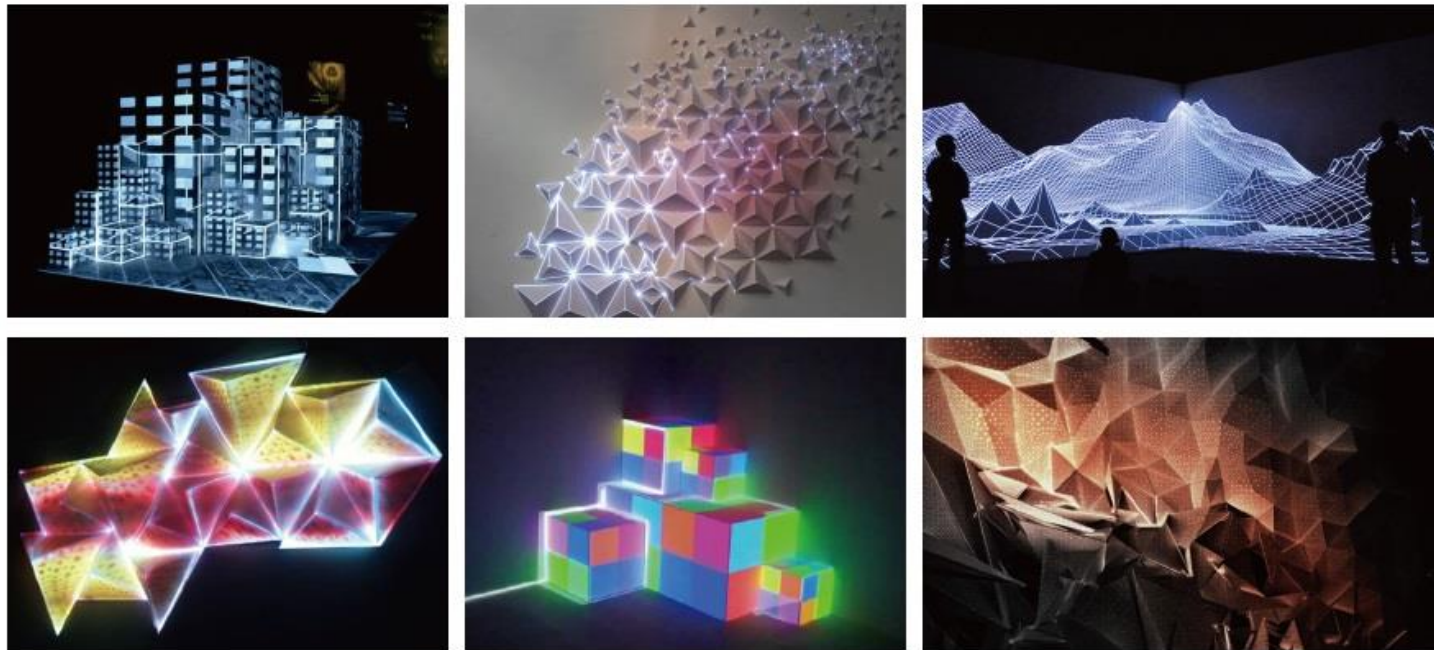


그림 6-19 다양한 3D 매핑 사례 [14]

5.4 3D 매핑

■ 텍스처 매핑(Texture Mapping)

- 3D 객체의 표면에 질감 데이터를 입히는 것으로 질감 매핑이라고도 함
- 2차원 이미지가 3차원 객체의 표면에 칠해진 것처럼 표현됨
- 물체 표면의 밝기, 빛의 반사, 굴절 등을 고려하여 질감을 표현함

▶ 동영상 보기 : [텍스처 매핑](#)

■ 범프 매핑(Bump Mapping)

- 표면이 불규칙하거나 굴곡이 존재할 때 표면을 구성하는 픽셀마다 수치를 변경하여 엠보싱 효과를 낸 것처럼 표현
- 조명으로 각 픽셀의 밝기와 음영을 조절할 수 있음
- 텍스처 매핑과 병용해서 이미지를 좀 더 현실적으로 표현할 수 있음

▶ 동영상 보기 : [범프 매핑\(Bump Mapping\)](#)



그림 6-20 텍스처 매핑과 범프 매핑 [15]

5.4 3D 매핑

■디스플레이스먼트 매핑(Displacement Mapping)

- 3D 객체의 표면을 뒤틀어 변형하는 기법으로 변위 매핑이라고도 함
- 표면의 픽셀들을 공간적으로 이동시켜 실제로 굴곡진 효과가 나타남
- 표면을 강하게 변화시키기 때문에 범프 매핑보다 훨씬 사실적임
- 렌더링 시간이 범프 매핑에 비해서 훨씬 많이 소요되는 단점이 있음

▶ 동영상 보기 : [디스플레이스먼트 매핑\(Displacement Mapping\)](#)

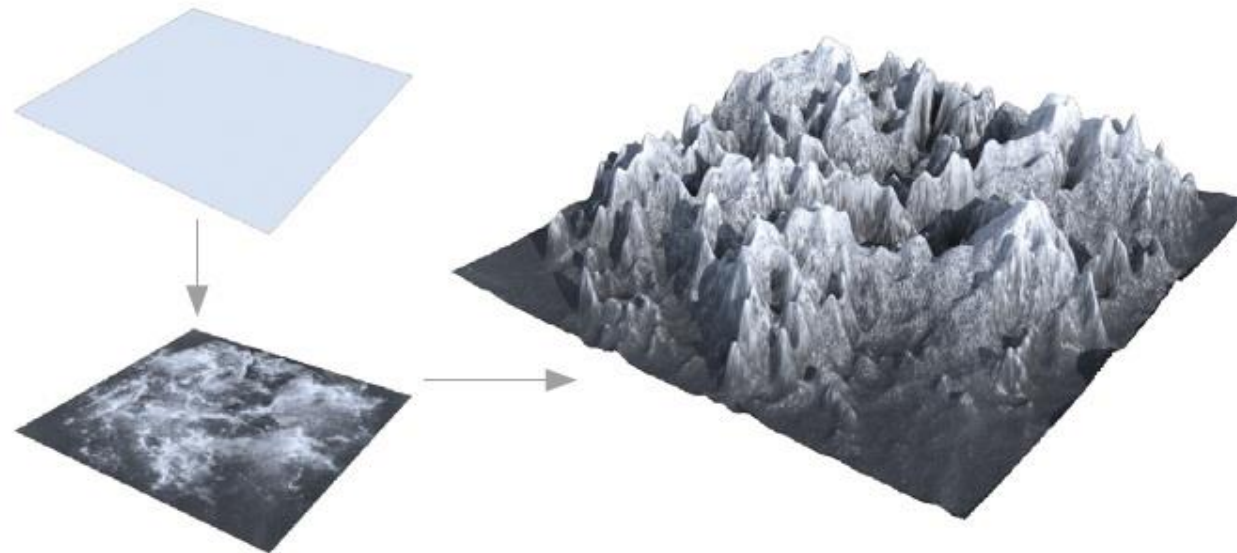


그림 6-21 디스플레이스먼트 매핑 과정

■ 오퍼시티 매핑(Opacity Mapping)

- 3D 객체의 표면을 불투명한 부분과 투명한 부분으로 구분하여 표현하는 기법, 불투명 매핑이라고도 함
- 매핑 이미지의 명도 단계에 따라 3D 객체의 투명도를 조절
- 흰색 부분은 투명하게 검은색 부분은 불투명하게 표현됨

▶ 동영상 보기 : [오퍼시티 매핑\(Opacity Mapping\)](#)



그림 6-23 오퍼시티 매핑에서 투명도 조절의 결과 [17]

5.4 3D 매핑

■리플렉션 매핑(Reflection Mapping)

- 3D 객체가 조명을 받아 반사체에 주위의 환경이 비쳐 보이는 것을 표현하는 기법, 반사 매핑이라고도 함
- 금속, 유리 등과 같은 투명체를 통해 보는 경우에 발생하는 굴절 현상을 표현
- 계산량이 적기 때문에 렌더링 시간이 적어 많이 사용됨

▶ 동영상 보기 : [리플렉션 매핑\(Reflection Mapping\)](#)



그림 6-24 리플렉션 매핑의 예 [18]

■라이트 매핑(Light Mapping)

- 조명을 사용하여 3D 객체의 표면을 표현하는 기법으로 투영 매핑(Projection Mapping)이라고도 함
- 이미지 표면에 라이트 맵(Light Map)을 적용하여 새로운 이미지를 만듦
- 조명을 현실적으로 표현하기 위해 정점 조명을 사용했으나, 요즘에는 픽셀당 조명을 처리하는 방식으로 대체

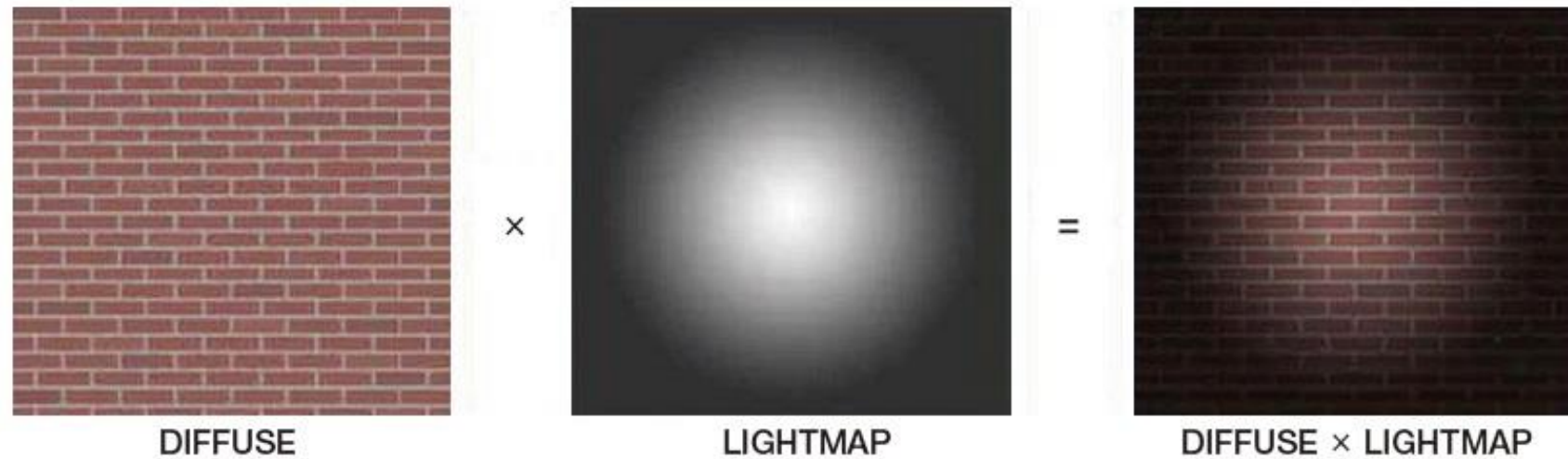


그림 6-25 라이트 매핑의 예 [19]

■라이트 매핑(Light Mapping)

➤정점 조명(Vertex Lighting)

- ✓3D 객체를 구성하는 모든 폴리곤의 색상 값을 빛이 미치는 영향을 고려하여 계산하는 방식
- ✓모든 계산은 실시간으로 처리되기 때문에 실시간 조명이 가능하며 조명 효과도 우수함
- ✓그림자는 정확하지 않다는 단점이 있음

➤픽셀 기반의 실시간 조명 처리

- ✓픽셀마다 조명이 미치는 영향에 근거하여 색상 값을 계산하는 방식
- ✓픽셀마다 계산하기 때문에 그래픽 엔진에 커다란 부하를 주어 실시간 게임에는 맞지 않음

➤픽셀 기반의 라이트 맵 조명 처리

- ✓실시간 동적 조명을 표현하기 위해 정점 조명과 결합하면 유리함
- ✓전처리 작업에서 비용이 많이 소모되는 조명 계산 과정이 처리되기 때문에 속도가 빠름

5.5 Shading

■ Shading

- 3D 객체의 표면에 대하여 광원으로부터의 거리, 각도, 색상, 명암 등을 계산하여 음영을 표현하는 기법
- 렌더링(Rendering) 과정의 일부
- 모든 픽셀에 대해 색상을 계산하기 때문에 사용하는 기법에 따라서 렌더링 시간과 품질의 차이가 발생함
- 와이어 프레임 Shading, 플랫 Shading, 고라운드 Shading, 평 Shading 등이 있음

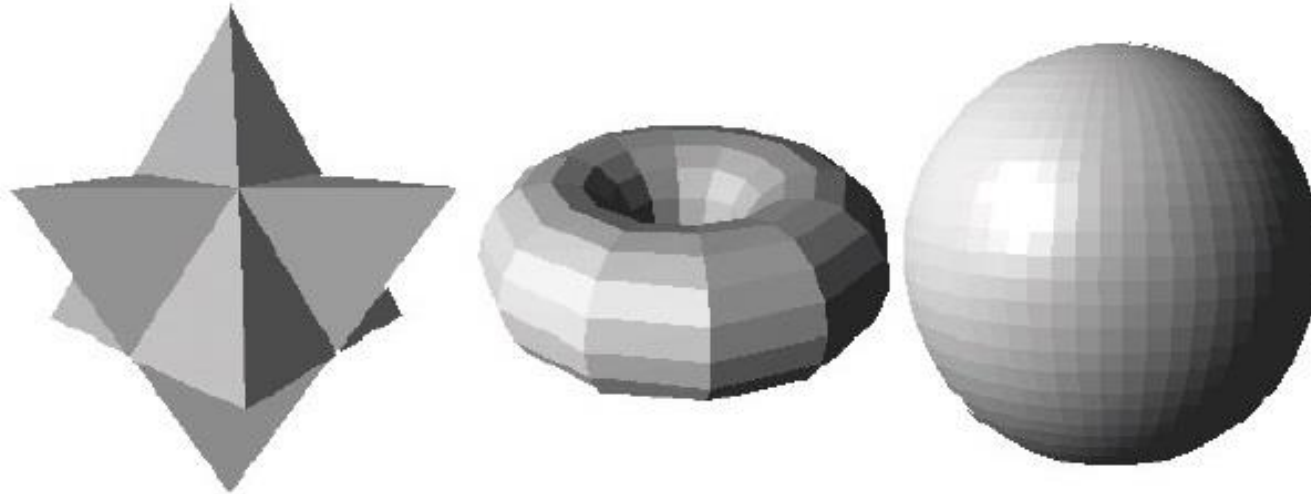


그림 6-27 셰이딩 기법에 의한 음영 표현 [21]

5.5 Shading

■ 와이어 프레임 셰이딩(Wire-Frame Shading)

- 철사 같은 수많은 선을 결합하여 입체적으로 만든 뼈대 형태의 3D 객체에 음영을 표현하는 기법
- 은선 또는 은면 제거 작업을 수행한 후 각각의 면에 적절한 색상과 음영을 표현
- 구조가 간단하기 때문에 물체의 표면에 색상과 음영을 표현하는 데 시간이 적게 걸림

■ 플랫 셰이딩(Flat Shading)

- 3D 객체를 구성하는 폴리곤들의 표면을 단순히 하나의 색상으로 채우는 방법
- 표면과 표면이 마주치는 모서리 경계 부분에 개별적인 색상을 지정하거나 급격한 명암차를 줘 음영 표현
- 폴리곤의 표면마다 단색을 지정하므로 렌더링 시간이 빠른 반면 단조롭고 질감이 거침

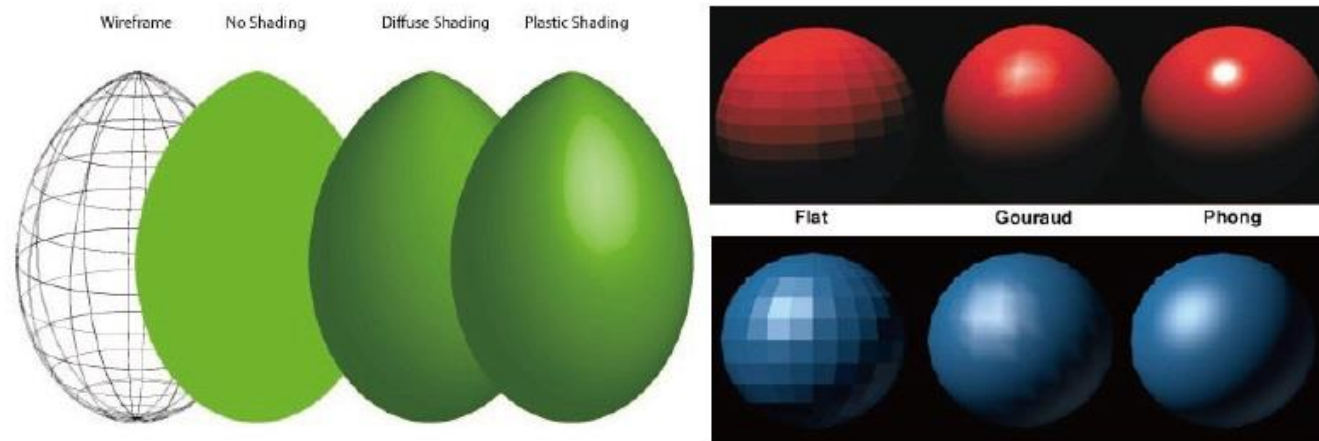


그림 6-28 와이어 프레임 셰이딩과 다양한 셰이딩 기법의 이미지 [22]

■고라우드 셰이딩(Gouraud Shading)

- 3D 객체의 표면에 연속적인 음영을 나타내기 위하여 보간법을 사용
- 모든 삼각형 폴리곤에 대하여 꼭짓점의 조명을 계산하고 삼각형 표면에 연속적인 조명을 표현
- 플랫 셰이딩보다 부드러운 명암을 표현할 수 있고 평 셰이딩보다는 처리 속도가 빠르며 음영 품질이 우수함

■평 셰이딩(Phong Shading)

- 모든 픽셀들이 가지고 있는 조명과 색상을 개별적으로 계산하여 실제와 가장 유사하게 표현하는 방법
- 부드러운 표면 표현이 가능하여 스무드 셰이딩(Smooth Shading)이라고도 함
- 가장 수준 높은 렌더링이 가능하지만 많은 시간이 소요됨

■ 렌더링(Rendering)

- 컴퓨터 그래픽스 또는 디지털 애니메이션 분야에서 3D 객체가 실제로 존재하는 것처럼 입체적으로 표현하는 과정
- 3차원으로 형상화한 객체에 매핑과 셰이딩을 모두 계산하여 최종적으로 완성된 그래픽 이미지를 출력
- 최종 이미지의 품질은 렌더링 알고리즘(Rendering Algorithm)에 의해 결정됨
- 렌더링 알고리즘은 많은 명령을 수행하고 많은 픽셀값들을 계산하기 때문에 처리 시간이 오래 걸림
- 현재 대부분의 그래픽 시스템들은 렌더링 알고리즘이 하드웨어로 구현되어 있기 때문에 손쉽게 사용
- 렌더링에는 와이어프레임 렌더링, 레이트레이싱 렌더링, 래디오시티 렌더링, 스캔라인 렌더링, 평 렌더링 등이 있음



▲ 3D 모형의 모델링과 렌더링의 출력 이미지

▶ 동영상 보기 : [렌더링\(Rendering\)](#)

5.6 렌더링

■ 와이어프레임 렌더링(Wireframe Rendering)

- 3D 객체의 뼈대에 해당하는 모서리만을 표현함
- 렌더링 방식 중에서 가장 단순하며 빠름
- 와이어프레임의 각 면에 대하여 조명으로부터의 거리, 각도, 색상, 명암 등을 복합적으로 계산하여 표현함

▶ 동영상 보기 : [와이어프레임 렌더링\(Wireframe Rendering\)](#)

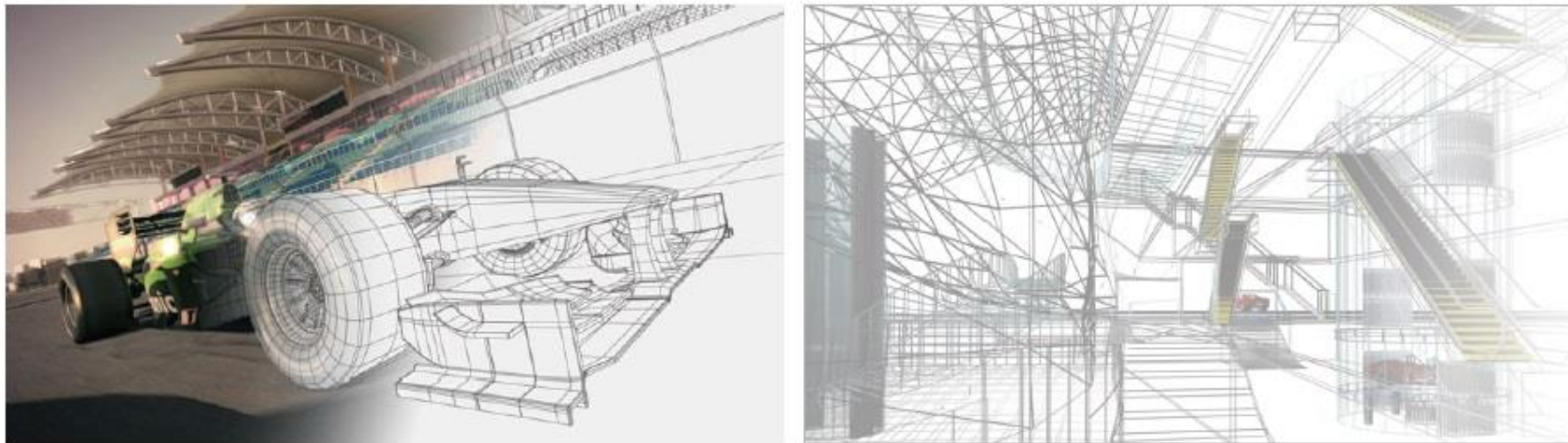


그림 6-30 와이어프레임 렌더링 [24]

5.6 렌더링

■레이트레이싱 렌더링(Raytracing Rendering)

- 광선 추적 방식
- 조명이 시작된 위치에서 도달할 때까지의 경로를 역추적하면서 조명에 의해 변화된 모든 픽셀의 색상을 표현
- 조명을 추적하는 과정에서 반복적으로 발생하는 조명의 반사와 굴절을 적용하여 이미지를 생성함
- 현실에 가까운 입체적인 표현이 가능한 방법

▶ 동영상 보기 : [레이트레이싱 렌더링\(Raytracing Rendering\)](#)

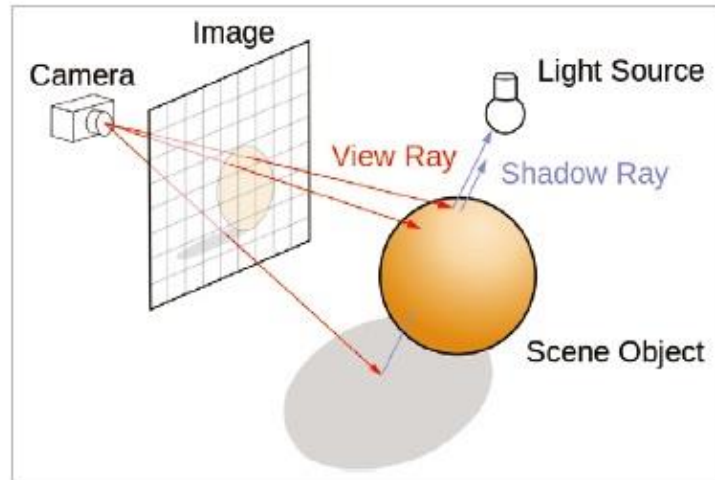


그림 6-31 레이트레이싱 렌더링의 개념과 출력 이미지 [25]

5.6 렌더링

■라디오시티 렌더링(Radiosity Rendering)

- 현실 세계처럼 조명이 난반사될 때 주변의 다른 물체들과의 관계까지 포괄적으로 고려하여 정밀하게 계산하는 방식
- 가장 많이 선호하는 방식
- 3D 객체의 모든 표면을 패치라고 불리는 작은 조각으로 분할하고 상호간의 반사 에너지를 주고 받음
- 약한 조명에 대해서도 계산이 가능
- 일반적인 렌더링보다 많은 시간이 소요됨

▶ 동영상 보기 : [라디오시티 렌더링\(Radiosity Rendering\)](#)



그림 6-32 라디오시티 렌더링의 출력 이미지 [26]

5.6 렌더링

■ 스캔라인 렌더링(Scanline Rendering)

- 한 줄의 스캔라인에서 다음 스캔라인으로 순차적으로 이동하면서 각 픽셀의 색상과 음영을 계산하는 방식
- 레이트레이싱 렌더링과 유사하게 결과물을 표현함
- 처리 속도가 빠르지만 안티앨리어싱처럼 경계 부분이 부드럽지 못하다는 단점이 있음

▶ 동영상 보기 : [스캔라인 렌더링\(Scanline Rendering\)](#)

■ 평 렌더링(Phong Rendering)

- 평 셰이딩 방식을 사용하는 방식
- 가장 실체에 가깝게 3D 객체를 렌더링하는 방식
- 픽셀이 갖는 색상과 음영을 개별적으로 계산함

