

# 그래픽스 강의노트 02 - 컴퓨터 그래픽스란

강영민

동명대학교

2015년 2학기

# 컴퓨터 그래픽스란

- 컴퓨터 그래픽스 = 컴퓨터를 이용한 그래픽스
  - 그래픽스의 어원은 그리기, 쓰기 등을 의미하는 그리스어 ' $\gamma\rho\alpha\varphi\eta$ (graphé)'
    - 어떠한 표면에 시각적인 표현을 드러내는 것
    - 컴퓨터 그래픽스는 컴퓨터를 활용하여 시각적으로 관찰할 수 있는 영상을 생성하는 것
- 그래픽스는 단순히 시각 정보 그 자체를 의미하는 것이 아니라, 이러한 시각 정보를 생성하는 것과 관련된 다양한 이론과 기술을 다루는 분야

# 3차원 컴퓨터 그래픽스의 의미

- 컴퓨터 그래픽스는 매우 다양한 의미를 가짐.
  - 컴퓨터를 이용한 영상 데이터의 표현과 조작
  - 영상을 생성하고 조작하기 위한 다양한 기술
  - 시각적 콘텐츠를 디지털 기술로 합성/조작하는 방법을 연구하는 전산학 분야
- 3차원 컴퓨터 그래픽스
  - 3차원 기하(geometry) 객체 표현을 사용하는 그래픽스 분야
  - 컴퓨터 게임이나 영화의 경우 결과 영상은 2차원 공간인 모니터나 스크린에 표현되지만, 이 영상을 얻기 위해 처리되는 데이터는 3차원 공간에서 정의되고 조작되므로 3차원 컴퓨터 그래픽스
  - 컴퓨터 그래픽스의 세부 분야는 모델링(modeling), 애니메이션(animation), 렌더링(rendering)

# 모델링 modeling

- 영상 생성에 사용되는 객체의 기하적 특성을 정의하는 일
  - 객체를 표현하기 위해 사용되는 정점의 수를 결정
  - 정점들의 위치와 연결성을 설정
  - 그려질 면을 구성하는 일

# 애니메이션 animation

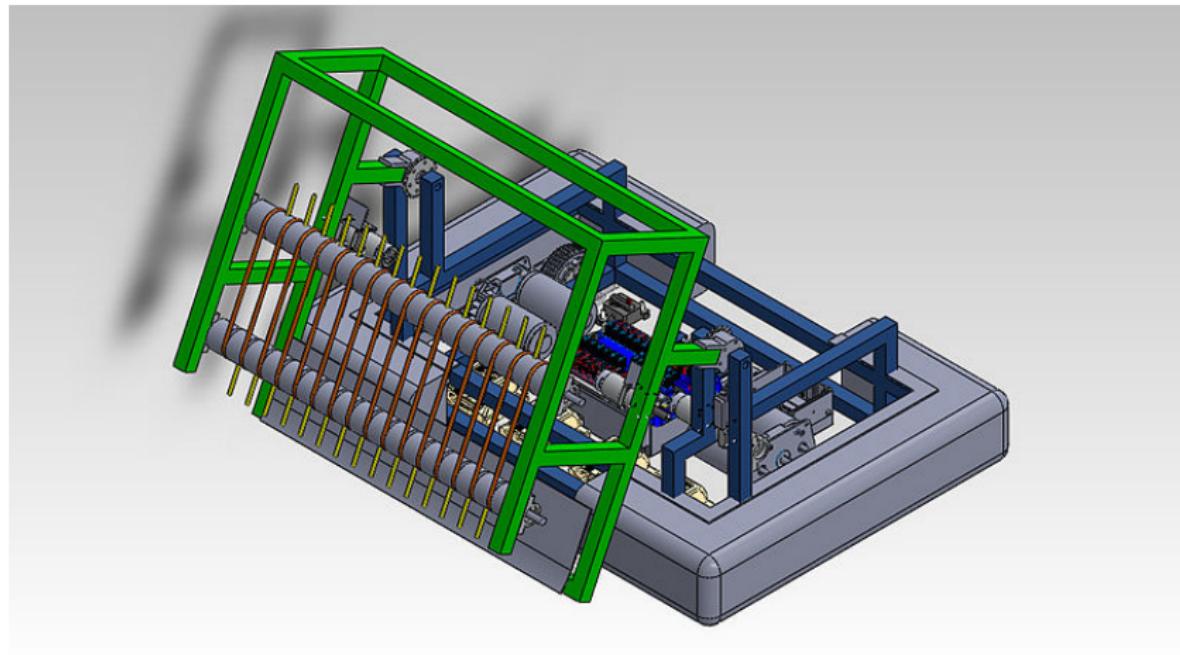
- 모델링 과정을 통해 결정된 기하객체에 대해 시간에 따른 변화를 설정하는 작업
- 일반적으로 이 변화의 대상은 기하객체의 위치
- 시간이 흐름에 따라 객체가 움직이는 모습을 생성
- 위치만이 애니메이션의 유일한 대상은 아니며, 시간에 따라 색상이 변하든가 물체에 적용된 텍스처(texture)가 변하는 등의 일도 모두 애니메이션

# 렌더링 rendering

- 모델링된 기하객체를 최종 관찰 표면에 그려내어 영상을 생성하는 분야
- 조명 모델을 이용하여 표면의 색상 및 밝기 등을 결정
- 결정된 색상을 2차원의 공간으로 투영하여 영상을 생성

# 그래픽스의 응용 - 캐드(CAD)

- 컴퓨터를 활용한 설계, 즉 ‘Computer-Aided Design’의 머릿글자
- 건축, 제조 등 다양한 분야의 설계에 적용
- 컴퓨터 기반 제조(Computer-Aided Manufacturing) 기술로 발전



# 그래픽스의 응용 - 가상 현실(virtual reality)

- 가상이라는 말은 ‘버추얼(virtual)’을 번역
- ‘버추얼’이라는 단어는 ‘사실과 거의 차이가 없는’ 것의 의미
- 가상현실을 가능하게 하는 기술적 요소는 입체화면, 3차원 입체 음향, 데이터 장갑 등의 입출력 장비, 컴퓨터 그래픽스 등
- 실재감을 제공하기 위해 인지과학, 전자공학, 기계공학 등의 다양한 분야의 연구결과들을 적용



# 그래픽스의 응용 - 실사 수준 고품질 가시화

- 가상 객체의 렌더링 결과를 실세계 물건과 구별이 어려워짐
- 실사 수준의 고품질 가시화 기술이 다양한 산업 분야에 적용
- 이미지를 얻는 데에 오랜 시간 소요 - 실시간 응용에 사용되지 않았음
- 하드웨어 성능과 그래픽스 알고리즘 발전 - 실시간 응용에 점차 적용



# 그래픽스의 응용 - 컴퓨터 아트(art)

컴퓨터 그래픽스 기술은 예술적 창작물을 만드는 분야에도 다양하게 활용되고 있다.



# 그래픽스의 응용 - 컴퓨터 애니메이션/영화

- 애니메이션/영화 산업에서 이제 컴퓨터 그래픽스는 가장 중요한 기술
- 영화는 그 특성상 상호작용성(interactivity)이 요구되지 않음
- 오랜 시간 동안 렌더링 작업을 수행하여 높은 수준의 영상 생성
- 오프라인 렌더링(offline rendering)



오프라인 렌더링



실시간 렌더링

# 그래픽스의 응용 - 컴퓨터 게임

- 컴퓨터 그래픽스가 가장 상업적으로 성공한 분야 중에 하나
- 상호작용성(interactivity)가 매우 중요
- 실시간 고속 렌더링 기술이 매우 중요 - GPU 활용
- 고품질의 영상을 빠른 시간에 그려내기 위한 다양한 기법을 발전



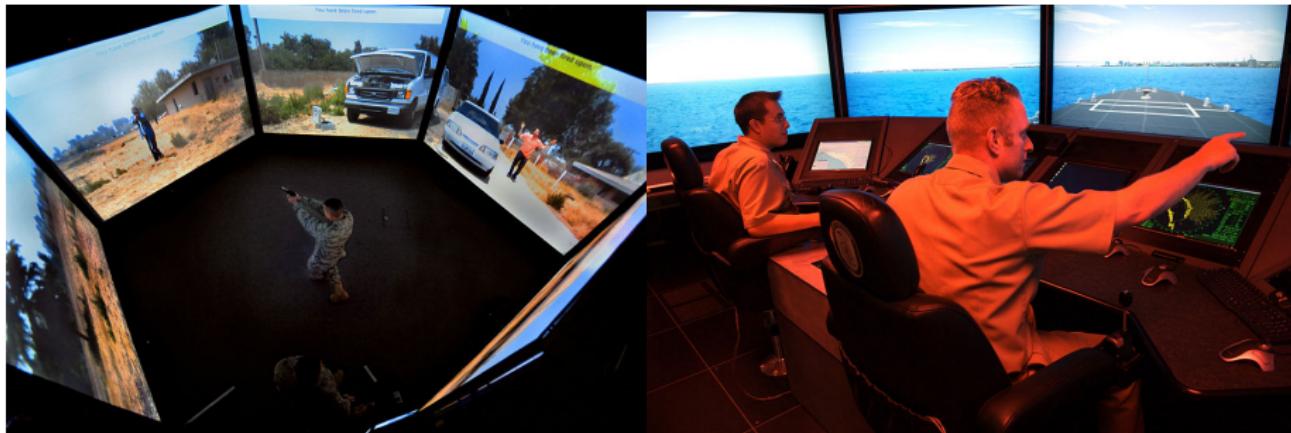
오프라인 렌더링



게임 그래픽스

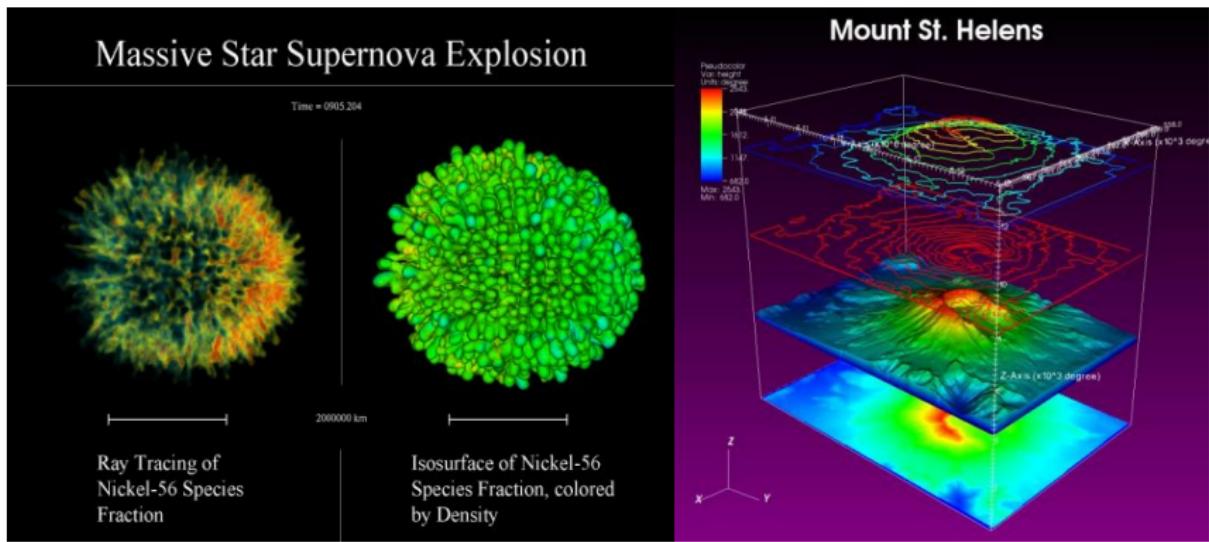
# 그래픽스의 응용 - 교육 및 훈련

- 강력한 가시화 기능은 교육과 훈련 분야에서 매우 효과적
- 실제 환경에서 훈련하기 힘든 전투나, 항공기 조종 등의 훈련
- 위험하거나 비용이 많이 드는 군사 분야 등에서 높은 활용도



# 그래픽스의 응용 - 정보 가시화

- 과학기술 분야에서 획득된 데이터를 이해하기 쉬운 형태로 표현
- 자연 현상을 관찰하여 수집된 데이터
- 계산을 통해 얻은 데이터
- 현상을 직관적으로 이해할 수 있게 하며, 패턴이나 추세 등을 파악



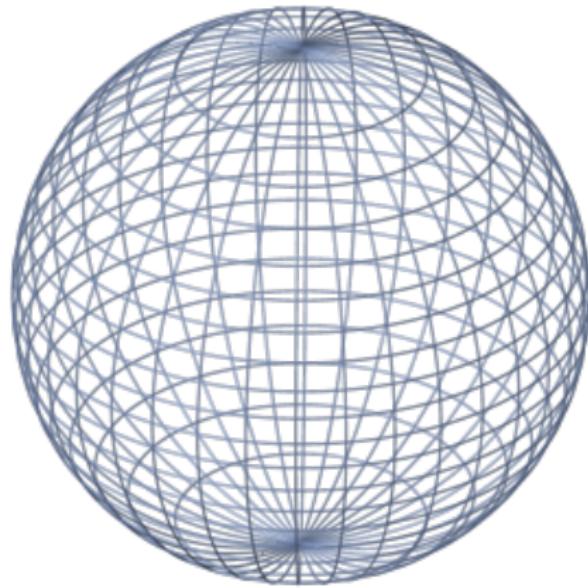
# 그래픽스의 역사 - 1950년대

- 컴퓨터 그래픽스가 시작된 시기
- 컴퓨팅 역사의 초기
- 래스터 장치 보다는 벡터 디스플레이 장치를 이용한 그래픽 표현
- 초창기 게임이라고 할 수 있는 ‘테니스 포 투(Tennis for Two)’



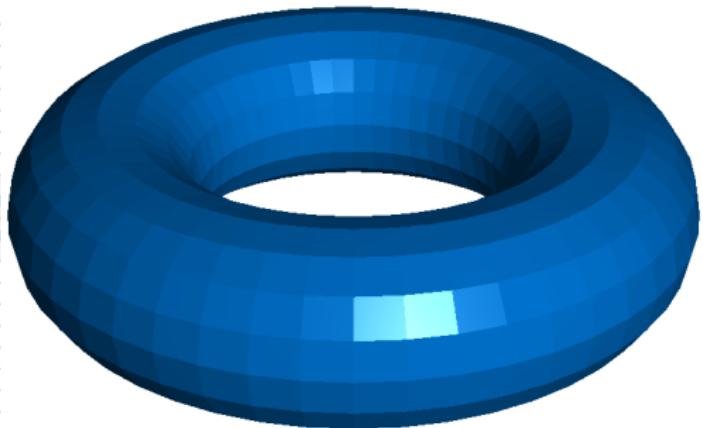
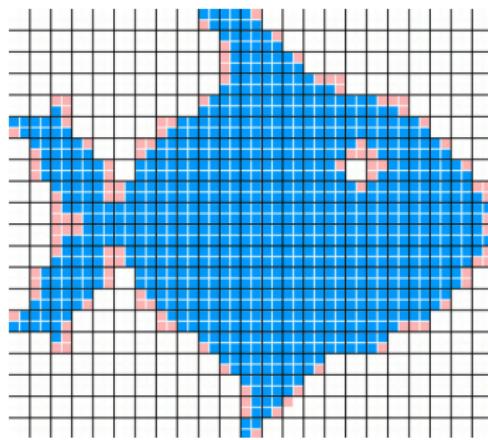
# 그래픽스의 역사 - 1960년대

- 연결구조만을 가시화하는 와이어 프레임(wireframe) 가시화 수준
- 다양한 그래픽 관련 기술과 산업이 출현
- HMD(head mounted display) 장치도 이 무렵에 개발



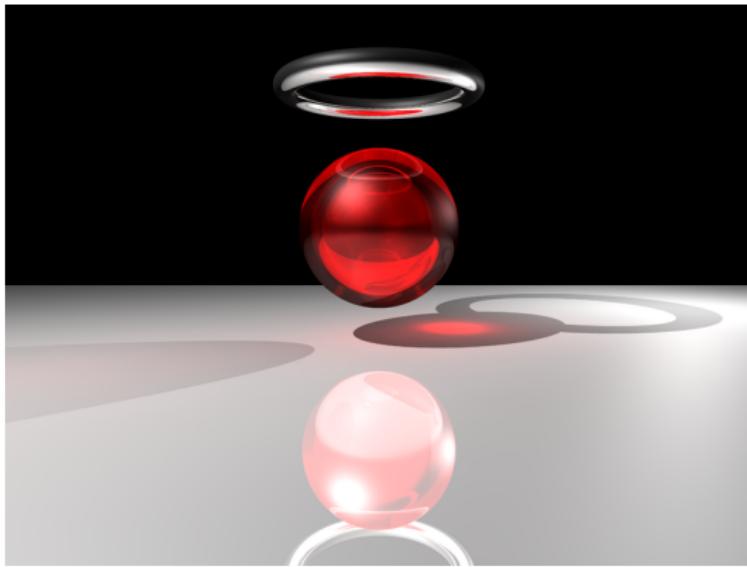
# 그래픽스의 역사 - 1970년대

- 래스터 그래픽스가 표준적인 그래픽스로 자리
- 래스터 그래픽스는 벡터 그래픽스와 달리 면을 칠하는 것이 쉬움
- 면의 음영을 결정하거나, 화소별 색상을 계산
- 고품질 이미지 생성이 가능



# 그래픽스의 역사 - 1980년대

- 그래픽스 관련 기술이 고도화
- 개인용 컴퓨터 보급으로 제한된 자원을 활용한 기술 연구 활발
- 고품질 이미지 생성 영역도 비약적인 발전
- 컴퓨터의 성능이 비약적으로 발전 - 구현이 어려웠던 광선 추적법(ray tracing) 등의 기술을 활용한 이미지 생성



# 그래픽스의 역사 - 1990년대

- 그래픽스 처리 전담 하드웨어가 개인용 컴퓨터 설치
- 실시간 그래픽스 분야가 획기적인 발전
- OpenGL API(application programming interface)가 완성
- 개인용 컴퓨터에서도 텍스처 매핑, 블렌딩(blending) 가능
- 다양한 종류의 버퍼(buffer) 활용 기술
- 애니메이션 작품들이 대단한 성공 - 관련 산업의 발전



# 그래픽스의 역사 - 2000년대 이후

- 극사실적 그래픽 표현 기술의 발전
- 실시간 그래픽스 분야에서는 NVIDIA, ATI의 경쟁 구도
- 게임이 시장의 중요한 동력
- GPU의 병렬처리 능력을 활용



# 3차원 공간의 표현과 영상 생성의 기본 개념

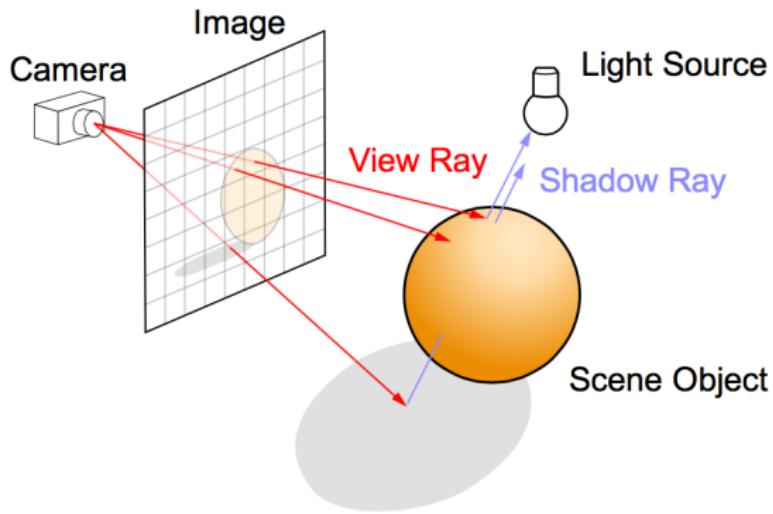
- 가상의 3차원 공간에 표현된 환경을 2차원 영상 이미지로 만들어 내는 일에는 기본적으로 다음과 같은 세 가지 물리적 요소가 고려되어야 함
  - 광원(Light)
  - 색(Color)
  - 인지(Perception)
- 합성 카메라 모델(synthetic camera model)
  - 광학 시스템에서 영상이 맷히는 방식을 흉내내어 가상의 카메라를 만드는 것
  - 객체, 관측자, 광원

# 광원(Light Source)

- 광원은 빛을 발생시키는 객체, 혹은 지점
- 빛이란 전자기파의 일부로 우리 눈이 반응하는 스펙트럼 영역
- 390nm 720nm 정도의 범위의 전자기파
- 파장인 긴 쪽은 붉은색 짧은 쪽은 자색
- 광원이 중요한 이유는 우리가 영상으로 담아내는 내용은 빛이 객체와 반응하여 관측자에게 도달한 결과이기 때문

# 광선 추적 기법(Ray Tracing Method)

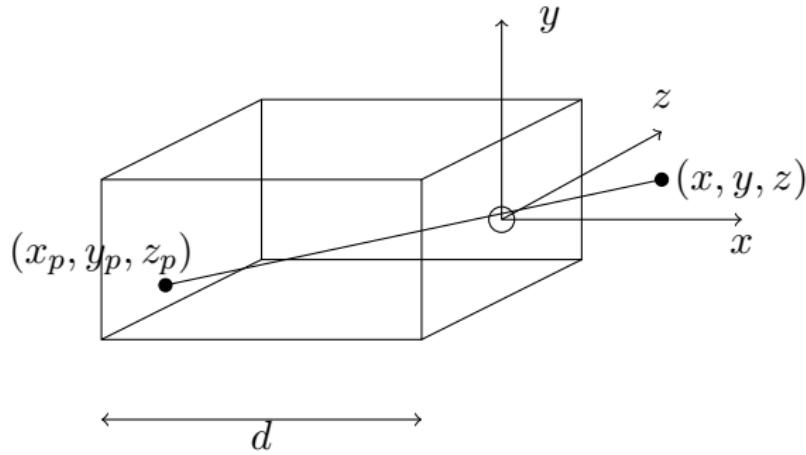
- 영상이 빛을 담아내는 가장 간단한 접근법은 광원에서 나온 빛을 추적하여 객체에 충돌하고 반사하는 과정을 계산한 뒤에 관측자의 영상 시스템에 도달하는 결과를 알아내는 것
- 이 방법의 단점은 빛과 객체의 상호작용을 완벽히 추적하는 것이 불가능하다는 점



# 바늘구멍 카메라(Pinhole Camera)

- 구멍을 통해 들어온 빛이 반대편 벽에 상을 맺게 함
- 상의 위치  $(x_p, y_p, z_p)$ 는 간단한 원근 투영법으로 계산 가능

$$\left( x_p = -\frac{xd}{z}, y_p = -\frac{yd}{z}, z_p = -d \right) \quad (1)$$



# 합성 카메라 모델

- 가상의 광학 시스템을 구성하여 가상 객체를 담아내는 것
- 구멍을 통과하기 전에 적절한 거리  $d$  만큼 앞에 구멍을 향해 진행하는 빛을 잡아낼 수 있는 평면
- 상이 거꾸로 맷히는 것과 달리 원래의 모양대로 상이 맷힘
- 상이 맷히는 곳은 다음과 같이 계산할 수 있다.

$$\left( x_p = \frac{xd}{z}, y_p = \frac{yd}{z}, z_p = d \right) \quad (2)$$

이러한 합성 카메라 모델은 객체와 관측자, 광원을 분리하여 다룰 수 있게 하며, 객체, 광원, 카메라의 속성을 설정하는 간단한 소프트웨어 API로 모델을 만들 수 있다. 또한 이 방식은 고속으로 처리가능한 하드웨어 구현이 용이하다는 장점도 가진다.

# 전역 조명과 지역 조명



(a) 지역조명



(b) 전역조명