컴퓨터 그래픽스
제10장 조명 모델

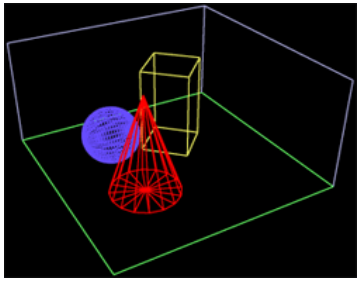
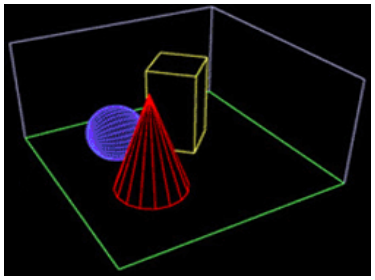
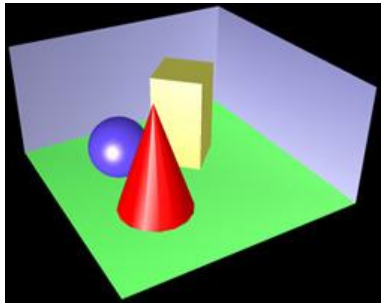
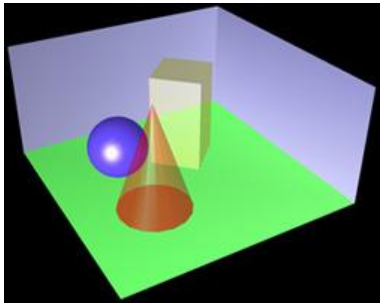
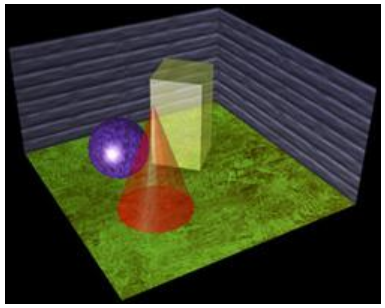
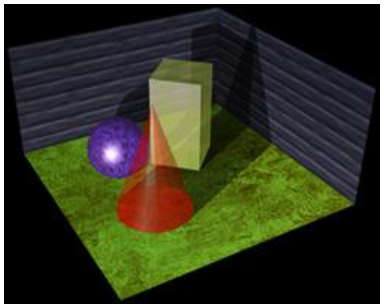
2017년 2학기

10장 학습 내용

- **3차원 렌더링**
- **조명 모델**
  - 산란 반사
  - 거울 반사
- **다각형 쉐이딩**
  - 균일 쉐이딩
  - Gouraud 쉐이딩
  - Phong 쉐이딩
  - 레이 트레이싱
- **텍스처 매핑**

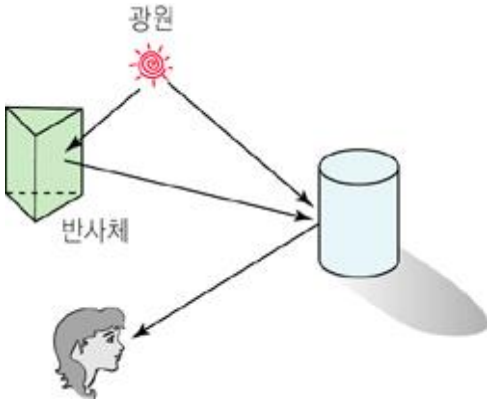
3차원 렌더링 과정

- **렌더링 (Rendering):**
  - 모델링된 3차원 객체가 실감나게 보이도록 처리하는 모든 과정
  - 은면 제거
    - 불필요한 렌더링을 피해 계산량을 줄일 수 있다.
  - 면의 쉐이딩
    - 객체의 다각형 면의 각 점의 위치에서 관찰자의 눈에 들어오는 빛의 반사량과 성분을 계산하는 과정
    - 입체감 있게 보이게 한다.
  - 투명한 물체의 표현
    - 빛의 투과량을 계산
  - 텍스처 매핑
    - 이미지나 사진 등 패턴을 을 표면에 입히는 과정
  - 그림자 생성
    - 빛에 의한 그림자를 생성한다.

조명 모델

- **조명 모델, 밝기 모델 (Lighting Model)**
  - 한 점에서의 색상과 명암: 그 점의 위치, 방향, 면의 재질에 따라 결정
  - 조명 모델: 한 점에서의 명암과 색상을 정하는 광학적 모델
  - 실세계에서 물체를 보려면
    - 광원(조명의 원천): 주변 조명, 점광원
    - 반사체: 빛을 반사하는 물체
  - 물체 표면에서는
    - 흡수 (Absorption)
    - 반사 (Reflection): 객체 표면이 성질의 재질에 따라 산란반사, 거울반사 현상
    - 굴절(Refraction) 도는 투과 (Transmission) 투명한 물체
  - 물체의 색은 광원, 물체, 관찰자 위치, 광원과 물체의 특성에 의해 결정된다.



조명 모델

- **조명의 종류**
  - 주변 조명, 배경 조명 (Ambient light, Background light)
    - 물체가 놓인 위치에 상관 없이 모든 물체에 균일하게 비추어지는 조명
  - 점 광원 (Point Light Source)
    - 위치와 방향을 가진 광원, 산란반사와 거울반사를 동시에 발생
    - 점광원에 노출되어 있는 객체는 더 많은 빛을 반사 → 어떤 면은 밝게 보이고 어떤 면은 어둡게 보인다.
    - 점광원 종류: 태양, 전구, 플래시 라이트, 형광등 등…
    - 빛이 반사될 표면과의 거리의 제곱에 비례하여 밝기 감쇠
    - 점광원에 의해 발생되는 반사
      - 산란반사 (Diffuse Reflection)
        - » 반사된 빛을 모든 방향으로 고르게 산란
        - » 관측자의 위치에 무관, 물체의 표면이 점광원을 향하고 있는 방향과 점광원까지의 거리에 의해서만 영향을 받는다.
      - 거울반사 (Specular Reflection)
        - » 한 방향으로 많은 빛을 반사
        - » 관찰자의 눈이 반사 방향과 일치하게 되면 빛이 많이 반사된 부분은 매우 밝게 보인다. (하이라이트)
      - 그림자 (Shadow)

조명 모델

- **표면의 성질**
  - 물체 표면의 반사양은 광원의 밝기와 위치, 물체 표면이 놓인 방향에 영향을 받는다.
  - 또한, 표면의 고유 색상과 재질, 광택의 정도(Shining or Dull) 에 따라서도 결정된다.
  - 그리고, 물체 표면의 투명도 및 빛의 반사도 같은 표면의 성질도 반사되는 빛에 영향을 준다.

  - 3차원 객체의 밝기나 색상 → 광원에 대한 정보, 객체 표면의 재질과 특성으로 결정
    - 광원에 대한 데이터:
      - 주변조명의 밝기, 산란반사와 거울반사를 만드는 점광원의 밝기
    - 객체 표면의 재질과 특성:
      - 주변조명과 산란반사의 반사계수, 거울반사를 일으키는 표면 물질의 특성 및 광택의 정도

산란 반사 (Diffuse Reflection)

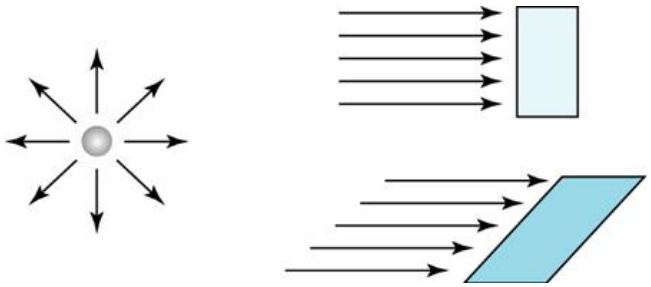
- **산란반사**
  - 주변조명에 의한 산란반사
    - 물체가 놓인 위치에 상관없이 모든 물체에 균일하게 비추어지는 조명
  - 점광원에 의한 산란반사
  - 총 산란 반사량 = (주변조명에 의한 산란반사량) + (점광원에 의한 산란반사량)
  - 산란반사의 특징
    - 광원에 의한 빛의 반사량이 관찰자의 위치에 관계없다.

산란 반사 (Diffuse Reflection)

- **주변 조명에 의한 산란 반사**
  - 물체가 놓인 위치에 상관없이 모든 물체에 균일하게 비추어지는 조명
    - $I = K_a I_a$, $0 < K_a < 1$
      - I_a: 주변 조명의 밝기
      - K_a: 주변조명 반사계수, 물체의 표면이 입사된 빛을 반사하는 정도, 표면을 이루는 고유 물질에 따라 다르다

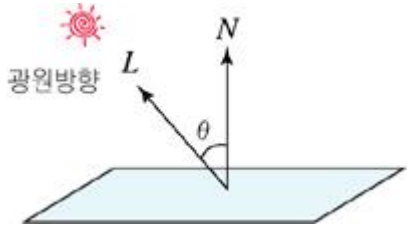
산란 반사 (Diffuse Reflection)

- **점광원에 의한 산란반사**
  - 표면에서 점광원까지의 거리와 빛의 투사각에 영향을 받는다



  - 물체의 표면이 광원을 향하여 정면으로 향하고 있을 때 가장 많은 빛을 받게 되고, 비스듬하게 놓인 경우에는 상대적으로 적은 양의 빛을 받게 된다.

  - 램버트의 코사인 법칙:
    - 표면이 받는 빛의 양은 cosθ에 비례한다.
      - $\cos\theta = N \bullet L$
        - » N: 표면의 법선벡터 (정규벡터)
        - » L: 광원의 방향벡터 (정규벡터)



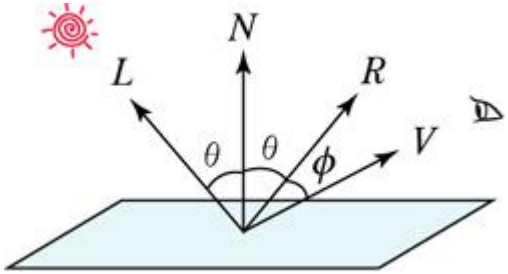
산란 반사 (Diffuse Reflection)

- 점광원에 의한 산란반사량: I
  - $\mathbf{I = \frac{K_d Ip}{d+d_0} \cos\theta = \frac{K_d Ip}{d+d_0} (N \cdot L)}$

    - I_p: 광원의 밝기
    - K_d: 표면의 산란반사 계수
    - d: 표면에서 광원까지의 거리
    - d_0: 일정값을 가지는 상수

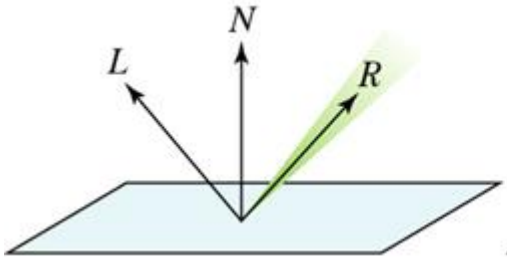
거울 반사 (Specular Reflection)

- **거울반사**
  - 점광원에 의해 발생하는 현상
  - 빛이 광택이 나는 표면에 입사될 때 관찰자가 빛의 입사각과 거의 같은 반사각 부근에 위치할 경우, 입사된 빛의 전부를 인식

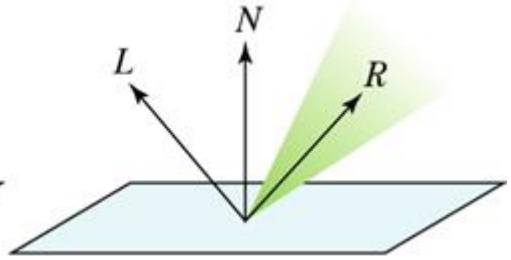


L: 빛의 입사 방향
R: 빛의 반사 방향
V: 관찰자 위치
Φ: V와 R 사이의 각도, Φ가 0도에 가까울수록 거울 반사량이 증가,

  - 반짝이는 표면일수록 거울반사가 일어날 수 있는 반사의 범위가 좁다



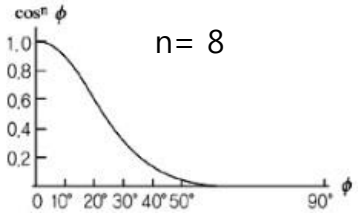
반짝거리는 표면의 반사 범위



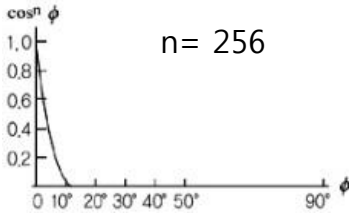
반짝거리지 않는 표면의 반사 범위

거울 반사 (Specular Reflection)

- **Phong의 거울반사**
  - $\mathbf{I = \frac{I_p}{d+d_0}\, w(\theta)\, \cos^n \Phi}$
    - I_p: 광원의 밝기
    - d: 광원에서 표면까지의 거리
    - w(θ): 표면 물질의 특성과 빛의 입사각 θ에 의해 결정하는 거울반사계수를 의미하는 함수
      - 반짝이는 물질일수록 거울반사 계수의 값이 크다
    - n: 표면의 광택 정도에 따라 정해지는 값
      - 광택이 많이 있는 표면: n의 값이 크다
      - 광택이 전혀 없는 표면: n의 값이 적다
      - $\cos^n \Phi$은 0과 1 사이의 값을 가지므로 n의 값이 클 경우 $\cos^n \Phi$의 값은 급격히 줄어들게 된다.



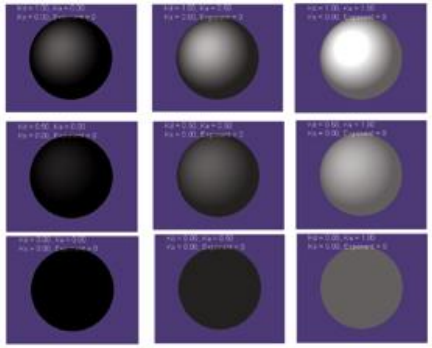
n= 8



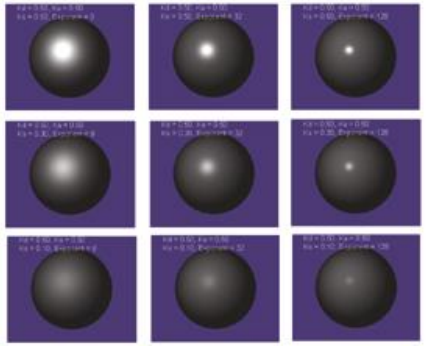
n= 256

거울 반사 (Specular Reflection)

- **Phong의 조명 모델**
  - 주변 조명과 점광원에 의한 산란반사와 거울 반사를 모두 합하면, 표면의 한 점에서의 총 밝기는
    - $I = K_a I_a + \frac{I_p}{d+d_0} [K_d (N \cdot L) + K_s (V \cdot R)^n]$
      - $W(\theta) = K_s$ (일정한 값)
      - 광원까지의 거리가 멀면 $\cos\theta = N \cdot L$은 표면의 위치에 상관없이 일정한 값을 가진다.
      - 관측자의 위치가 멀면 $\cos\Phi = V \cdot R$은 일정값을 가지게 된다.



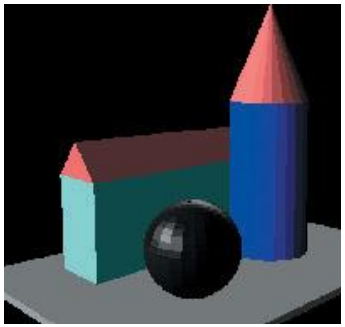
산란반사



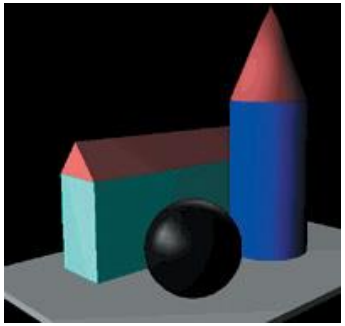
산란반사 + 거울반사

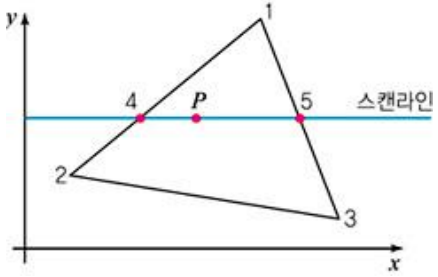
다각형 쉐이딩 기법

- **균일 쉐이딩 (Constant Shading, Flat Shading)**
  - 3차원 객체의 한 면을 일정한 색상이나 명암으로 표현
  - 객체 자체가 다면체인 경우는 대체적으로 비교적 양호함
  - 계산이 간단하지만 현실감이 떨어짐



- **Gouraud 쉐이딩**
  - 각 꼭지점의 밝기 값의 선형 보간
  - Flat shading에서의 intensity discontinuity를 제거함
  - 다각형의 정점에서의 명암/밝기 계산 → Linear interpolation
  - 어떤 경우 Highlight가 변칙적인 모양으로 나타나는 경우가 있다.
    - Mach Band 효과





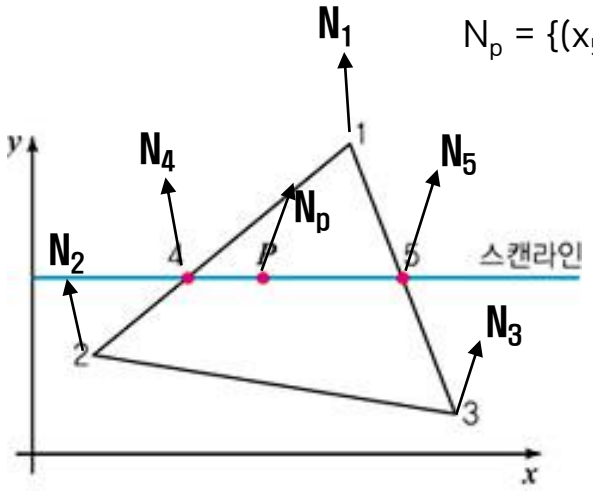
$I_4 = ((y_4-y_2)/(y_1-y_2))*I_1 + ((y_1-y_4)/(y_1-y_2))*I_2$

$I_p = ((x_5-x_p)/(x_5-x_4))*I_4 + ((x_p-x_4)/(x_5-x_4))*I_5$

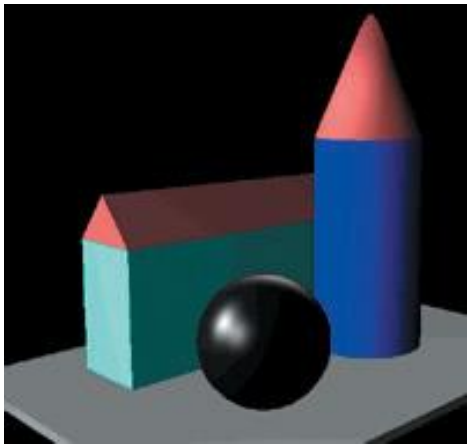
x: 각 점의 x 좌표, y: 각 점의 y 좌표

다각형 쉐이딩 기법

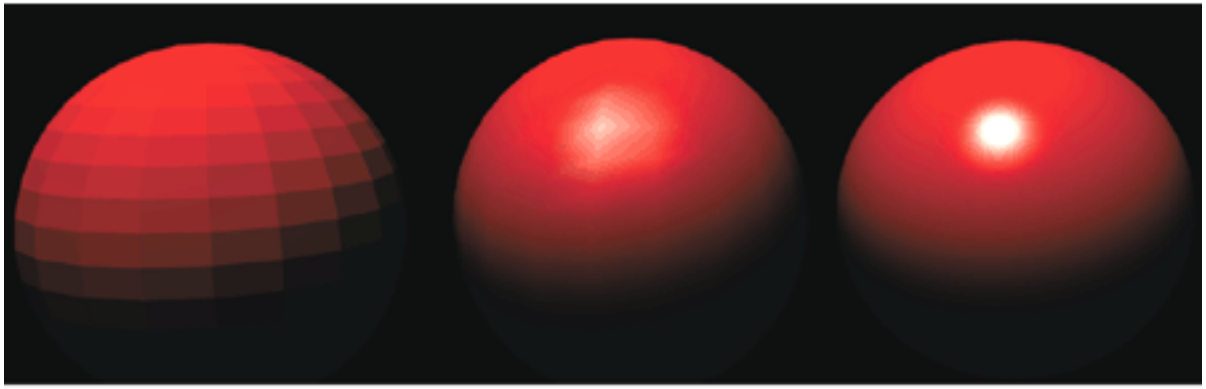
- **Phong Shading**
  - 법선 벡터의 선형 보간
  - 거울반사(Highlight)가 실감나게 보인다 (Mach Band 효과를 크게 줄임).
  - 계산시간이 오래 걸림(각 점에서 법선 벡터의 근사치를 이용하여 명암/밝기를 계산)



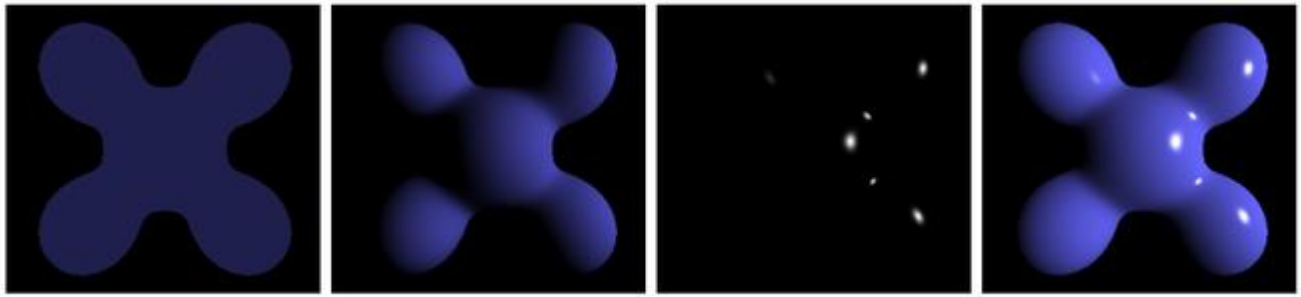
$$N_p = \{(x_5 - xp) / (x_5 - x_4)\}*N_4 + \{(x_p - x_4) / (x_5 - x_4)\} * N_5$$



쉐이딩 예제



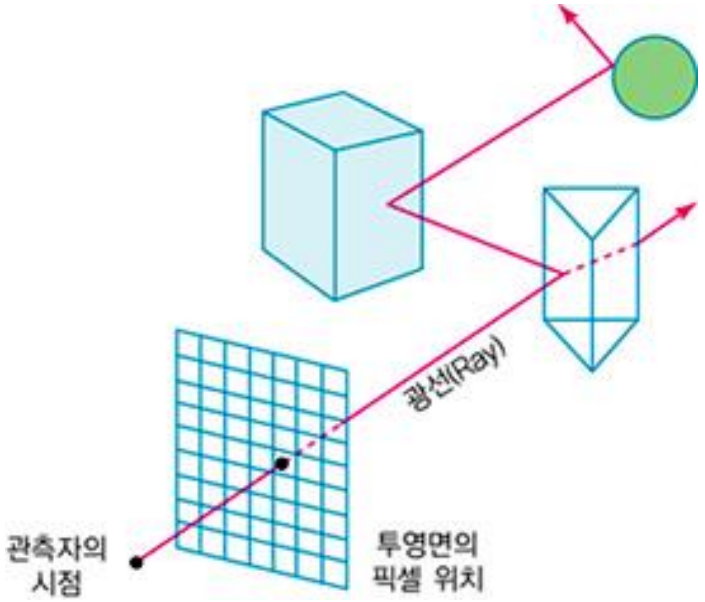
Flat Gouraud Phong

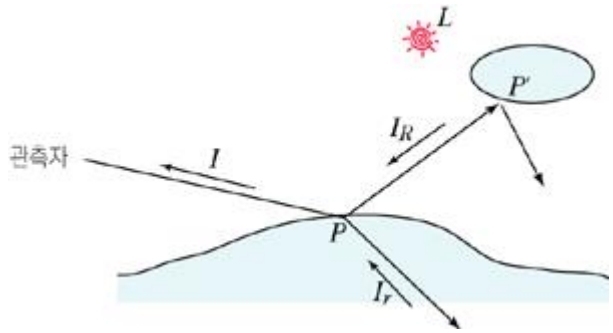


Ambient + Diffuse + Specular = Phong Reflection

광선 추적법

- **광선 추적법 (Ray Tracing)**
  - 광원에서 나온 빛이 물체에 투사되면 일부는 반사되고 일부는 굴절된다.
  - 반사된 빛은 다시 물체를 만나 반사, 굴절 가정을 거친다.
  - 이러한 과정을 통해 투영면의 각 픽셀 위치에 도달하는 모든 빛이 합해져서 최종적인 빛의 밝기를 결정한다.
    - 주변 조명, 산란반사, 거울반사, 반사광, 굴절광 등을 합해서 결정
    - $I = I_{amb} + I_{diff} + I_{spec} + I_{refl} + I_{refr}$





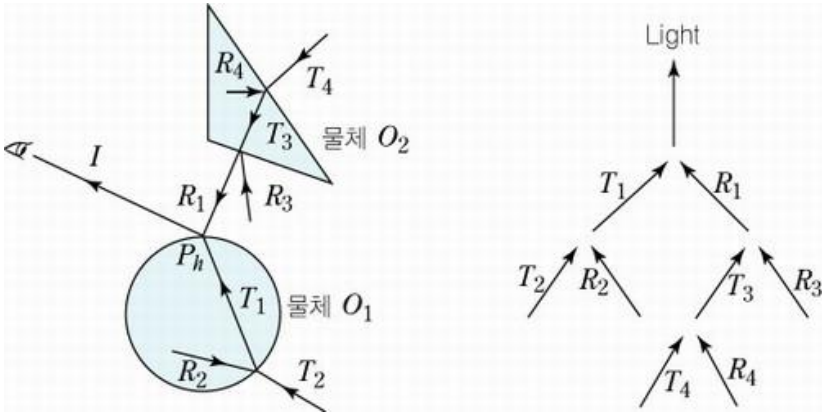
I_R, I_r: 2차 광선

광선 추적법

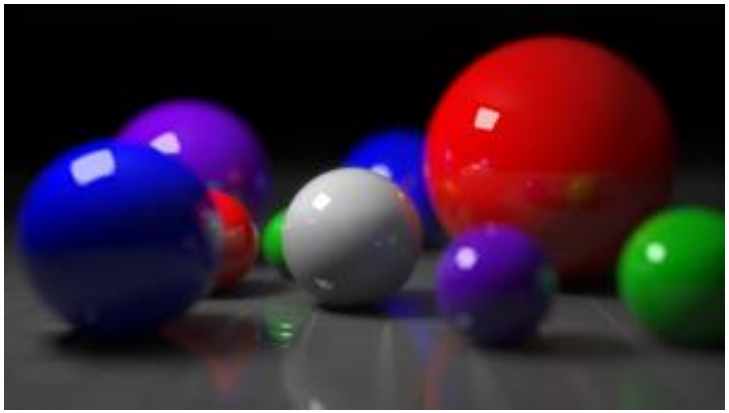
- 물체 O1상의 점 Ph에서 빛의 밝기 I는
  - 주변 조명, 점광원에 의한 산란반사 및 거울반사의 합
  - 반사광 R1과 굴절광 T1 에 의해 야기되는 빛의 성분
  - 빛 R1은 광선과 굴절광 T3에 의해 야기되는 빛의 합

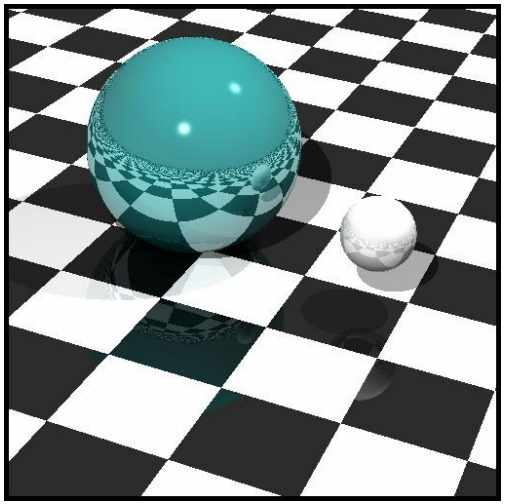
- 광선 추적 과정의 종료 조건:
  - 광선이 어떤 물체와도 만나지 않는 경우
  - 광선이 광원과 만나는 경우
  - 반사와 굴절의 회수가 지정회수보다 많아질 경우
  - 반사 및 굴절에 의한 빛의 세기가 임계값 이하가 될 경우

- 은면 제거와 그림자 생성 등을 동시에 처리
- 광선추적법은 매우 우수하고 현실감 있는 결과를 얻지만 많은 계산시간이 요구된다.



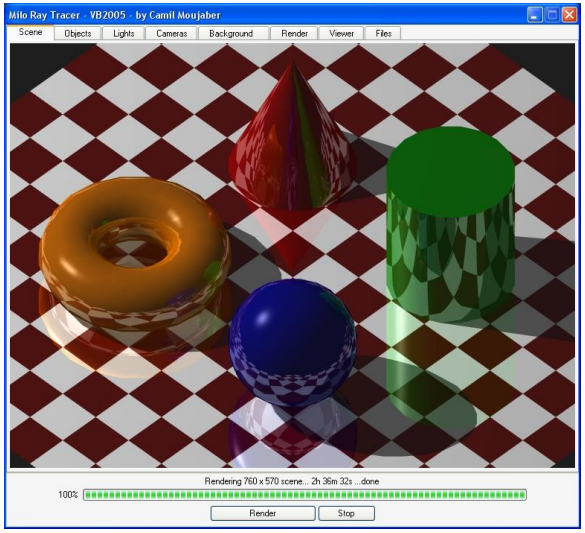
광선 추적법

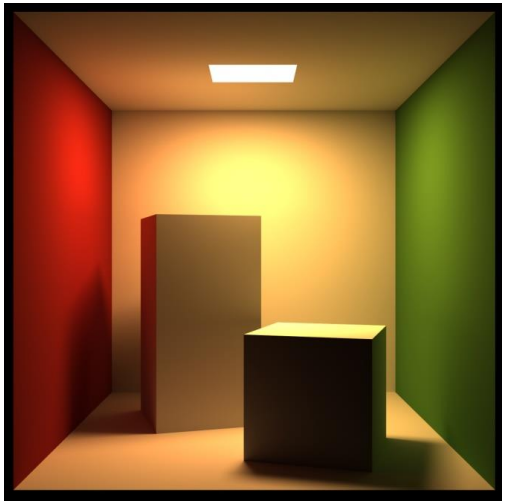


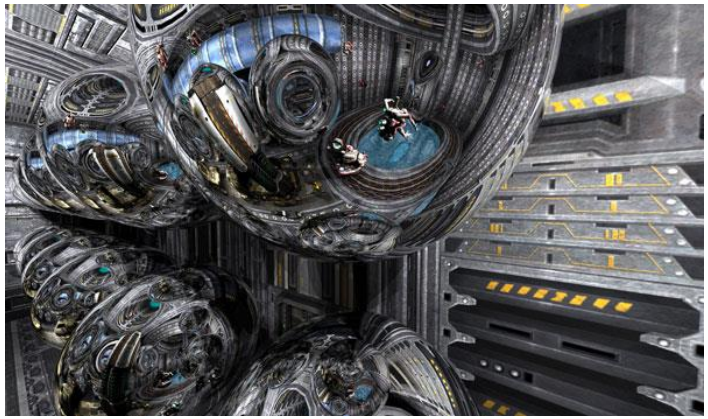


광선 추적법 (Ray Tracing)을 적용하여 생성된 렌더링 효과

광선 추적법


Milo Ray Tracer - VB2005 - by Camil Moujaber
Scene
Objects
Lights
Cameras
Background
Render
Viewer
Files
Rendering 760 x 570 scene... 2h 36m 32s ...done
100%
Render
Stop





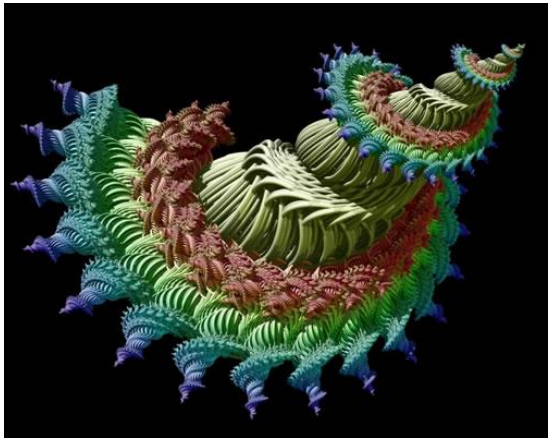

Rasterized
Ray traced

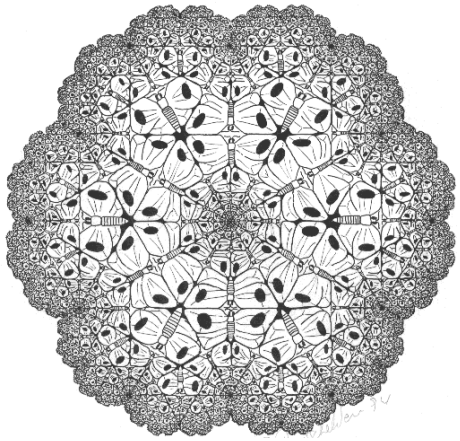
컴퓨터 그래픽스
제 11장 절차적 그래픽스 기법
제 12장 애니메이션

2017년 2학기

프랙탈

- **Fractal Model**
  - 1974년 Mandelbrot이 프랙탈 기하이론을 통해 산, 구름, 해안선, 나무 등의 자연물 등을 표현
  - 프랙탈: 불규칙해 보이는 자연물을 묘사하기에 좋은 수단
  - 비정수 차원(Fractional Dimensional) 이라는 용어에서 유래





프랙탈

- **프랙탈의 특성**
  - 비정수적 차원
    - 유클리드 기하학과는 다르게 비정수 차원으로 존재
  - 자기복제의 성질(Self-similarity Property)
    - 어떤 형태가 순환적으로 반복되어 전체 형상과 유사한 형태를 국부적인 부분에서도 발견
  - 무한대로 순환 반복(Infinite detail at every point)
    - 그림의 작은 부분을 확대하면 새로운 그림이 생성

- **프랙탈 생성 원리**
  - 임의적 반복 알고리즘(Random Iteration Algorithm)
    - 원칙 또는 규칙 몇 가지를 정한 후 무작위로 규칙을 반복 적용하면 프랙탈이 생성
    - 자기 순환적 복제

- **프랙탈은 자기 반복적인 성질을 가지고 있는 대상물 묘사에 적합**
  - 나무, 해안선 등 자연물 표시에 적절하다

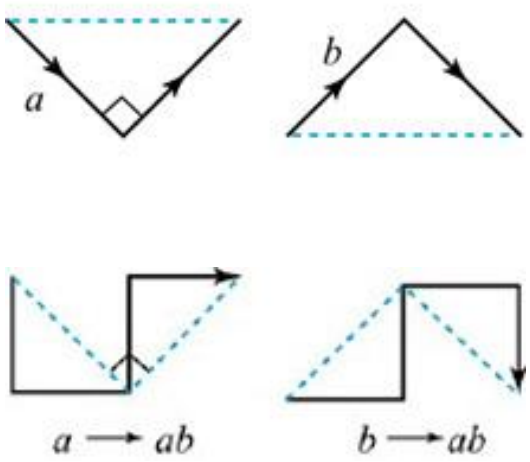
프랙탈의 생성 원리

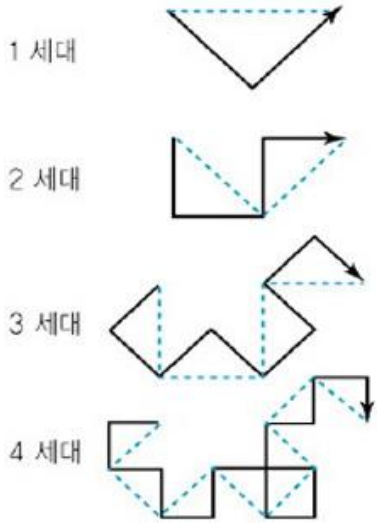
- **프랙탈 기하학은 간단한 절차(Fractal Generation Procedure)에 의해 정량적으로 표현**

  P1 = F(P0), P2 = F(P1), P3 = F(P2), …

  F : 변환 함수 (규칙적 혹은 임의 변이)

- **프랙탈의 예: Dragon curve**





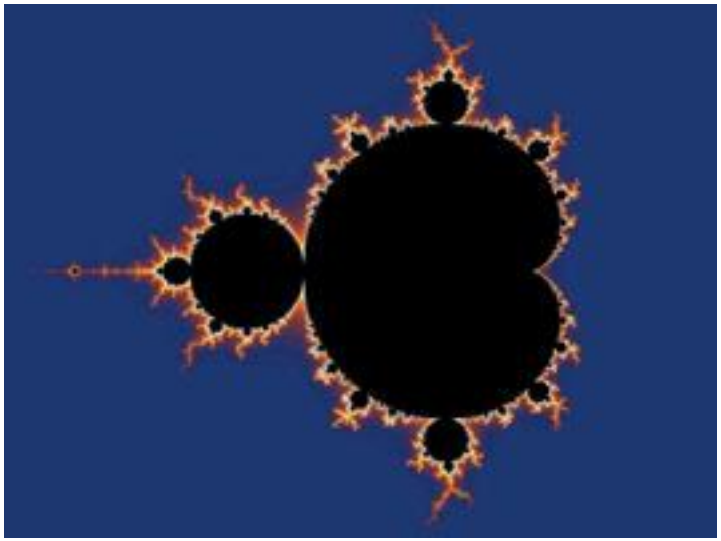
프랙탈의 생성 원리

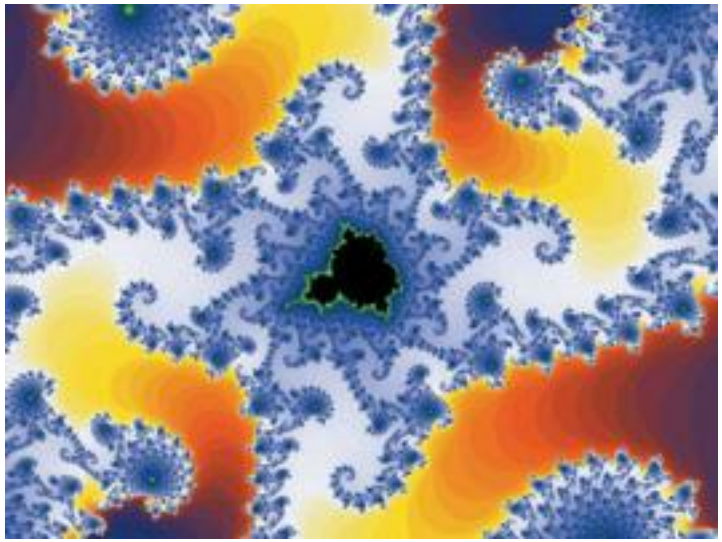
- **프랙탈 생성 기법**
  - IFS (Iterated Function System)
    - 자기복제적 성질을 가지는 함수나 기하변환을 통하여 초기 객체의 형태와 유사한 객체들을 반복적으로 복제해 나간다.
    - 대표 예) Barnsley 고사리, IFS 이용 애니메이션

  - 생성 문법 (Production Grammar) 이용 방법
    - 생성 규칙을 반복적으로 적용하여 새로운 모양을 만들어가는 방법
    - 대표 예) dragon curve
  - 중점변위법 (Midpoint Displacement Method)
    - 주어진 변의 중점을 정규분포로부터 생성된 난수만큼의 거리로 변위시켜 새로운 두 개의 변을 만들고 이러한 과정을 다시 변위시켜 새로운 변을 만들어가는 방법
    - 예) 산의 표면

프랙탈의 예

- **Mandelbrot 집합**
  - 가장 대표적인 프랙탈 기하학
  - 자기 순환적인 복제가 무한히 반복





프랙탈 기하학의 차원

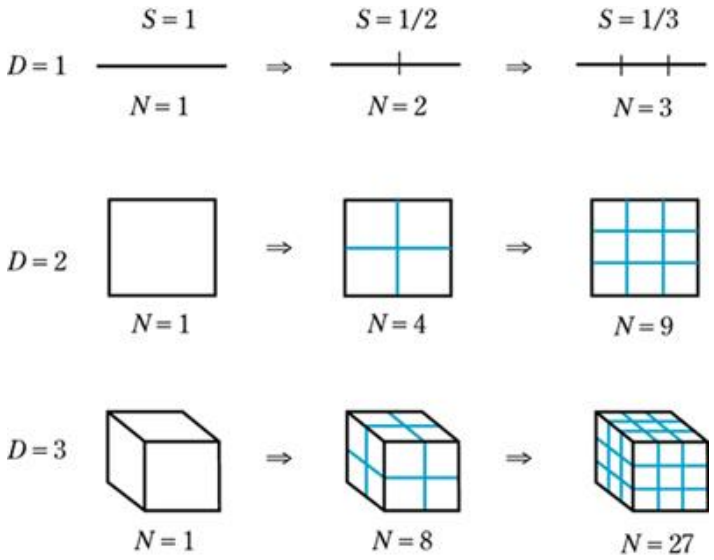
- **프랙탈 차원**

  S : 생성되는 객체의 크기(Scaling factor)
  N : 생성되는 객체의 수(Number of subparts)

  $NS^D = 1$

  $\rightarrow \quad D = \ln\frac{1}{N} / \ln S = \ln N / \ln\frac{1}{S}$

- **프랙탈의 차원을 유클리드 기하학에 적용**

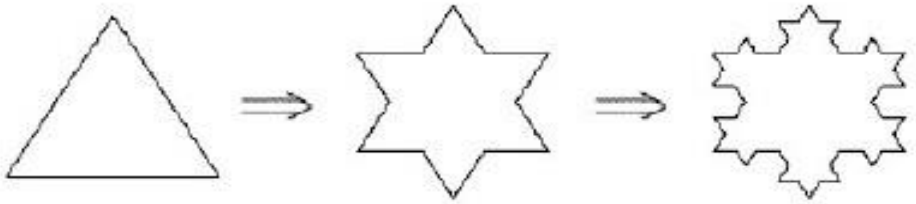


프랙탈 기하학의 차원

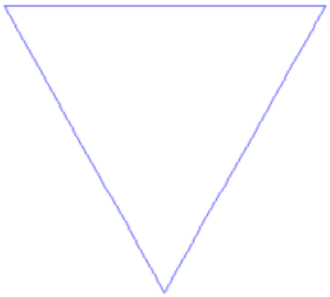
- **프랙탈 기하학의 차원의 예: Koch 의 Snowflake**

  D = ln4 / ln3 = 1.26...

  즉, 2차원은 아니지만 1차원도 아닌 비정수 차원



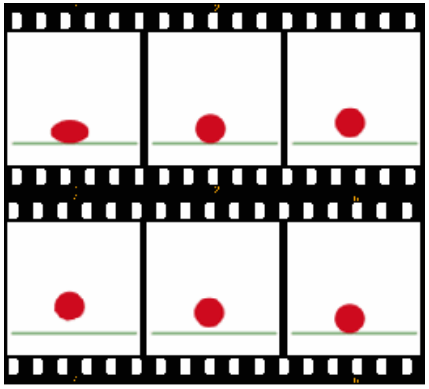
세그먼트 길이 = 1　　세그먼트 길이 = 1/3　　세그먼트 길이 = 1/9



전체 길이 = 1　　전체 길이 = 4/3　　전체 길이 = 16/9

애니메이션 (Animation)

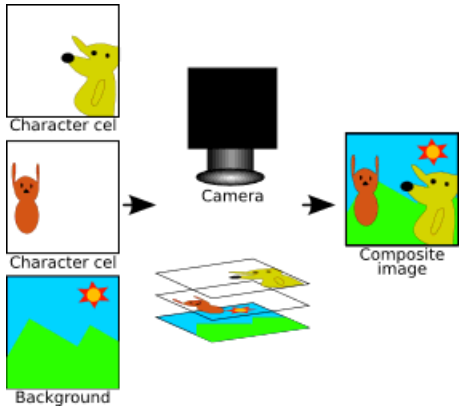
- **용어의 원래 의미: 생명이 없는 사물에 영혼이나 정신을 부여하는 행위**
  - 일련의 정지된 그림을 빠른 속도로 재생
  - 사람 눈의 생리적 특성인 잔상현상을 이용: 약 1/16초
  - 프레임(Frame) 사이의 간격이 잔상의 지속 시간을 초과하지 않아야 함
    - 플릭커(Flickering) 현상: 프레임 사이의 간격이 잔상 지속시간인 1/16초를 초과하면 화면이 끊어져 보인다.
  - 영화에서는 초당 24 프레임, NTSC 비디오 방식에서는 초당 30 프레임



애니메이션의 종류

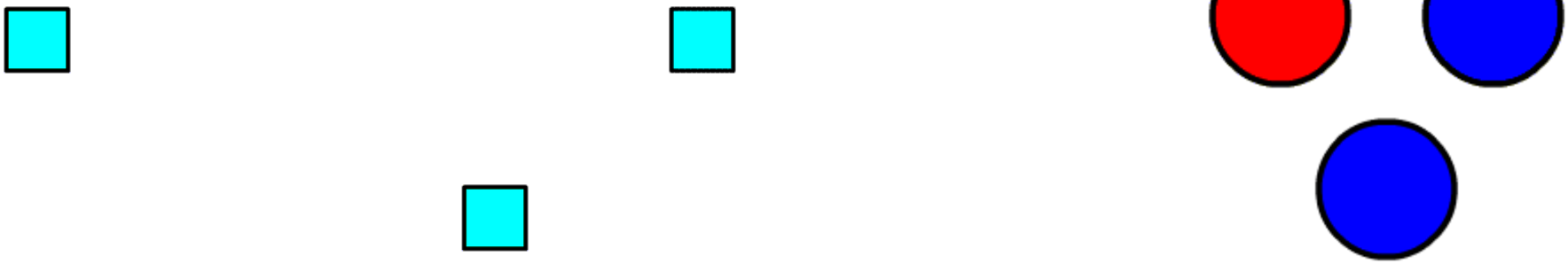
- **플립북 애니메이션(Flip-book Animation)**
  - 전통적인 애니메이션 기법으로 프레임(Frame-based) 애니메이션이라고도 함
  - 모든 프레임을 일일이 그려서 저장
  - 파일의 크기가 크기 때문에 인터넷과 같은 환경에서 데이터 전송 시간 많이 걸림
  - 예제)

- **셀 애니메이션(Cel Animation)**
  - 1910년 존 랜돌프가 개발
  - "CEL"이란 단어는 투명한 종이를 뜻하는 Celluloid를 의미
  - 2차원 애니메이션을 제작할 때 자주 사용하는 기법
    예) 디즈니 애니메이션 만화영화
  - 애니메이션을 만들기 위해서는
    하나의 배경 셀과 여러 장의 전경 셀이 필요
  - 1990년 이후 컴퓨터를 이용하여 제작



애니메이션의 종류

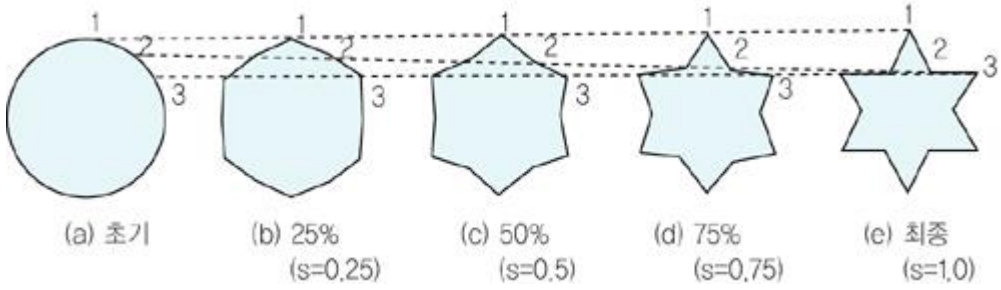
- **키프레임 애니메이션(Key-Frame Animation)**
  - 중요한 프레임(키프레임) → 그사이 장면(In-between Frames)들을 삽입
    - 트위닝(Tweening): 컴퓨터를 이용한 애니메이션에서도 이 개념을 그대로 적용
  - 애니메이터가 특정 프레임들을 키 프레임으로 지정하여 그리고,컴퓨터가 중간 프레임들을 보간법(Interpolation)으로 자동 생성
  - 선형 보간법(Linear Interpolation) 적용



시작 키프레임 끝 키프레임 최종 키프레임 색상변환 애니메이션

애니메이션의 종류

- **모핑 (Morphing)**
  - 모핑은 "변형"을 의미하는 "Metamorphosis" 에서 유래
  - 2개의 서로 다른 이미지나 3차원 모델 사이의 변화하는 과정을 표현
  - 모핑은 초기 형상과 최종 형상이 완전히 다른 물체인 경우에 적용
  - 두 프레임 간에 대응점들을 지정해 주고 중간 프레임들은 컴퓨터가 자동 생성
  - 2차원 그림에서는 물론 3차원 그림에도 적용
  - 모핑 예제



캐릭터 애니메이션

- **운동학(Forward Kinematics)**
  - 세그먼트(Segment)들이 계층구조에 의해 연결된 모델(Hierarchical model)에서 움직임의 시작점인 루트 관절로부터 각 관절의 각도와 위치를 누진적으로 계산
  - 기계적인 물체(예, 로봇)의 비교적 단순한 움직임이라면 객체 세그먼트의 각도와 위치를 일일이 수동적으로 조절 가능
  - 사람이나 동물의 관절 움직임을 애니메이션 하고자 할 때 운동학을 이용하면 대단히 번거로운 작업이 될 뿐만 아니라 직관적이지 않음

- **역운동학(Inverse Kinematics)**
  - 계층구조(Hierarchy) 제일 말단 부분의 객체(Effector)의 위치를 지정
  - 나머지 중간 관절들의 위치와 각도를 역으로 계산해 나가는 방식
  - 역운동학은 캐릭터 애니메이션에서 움직임을 효과적으로 표현