

# Illumination Models & Shading

---

COLLEGE OF COMPUTING

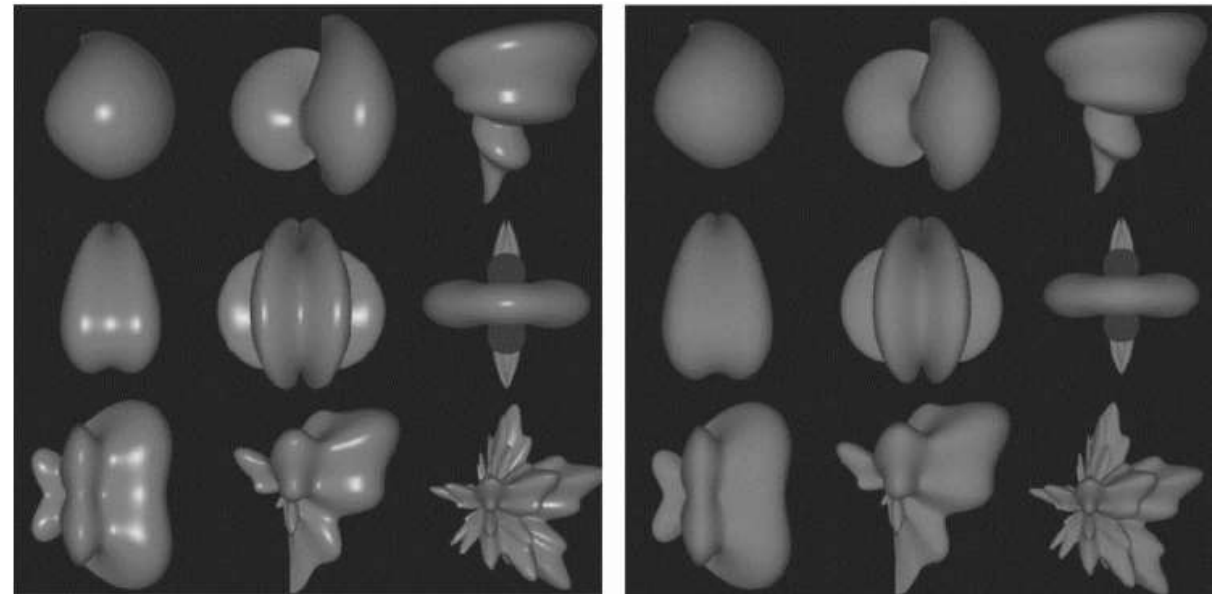
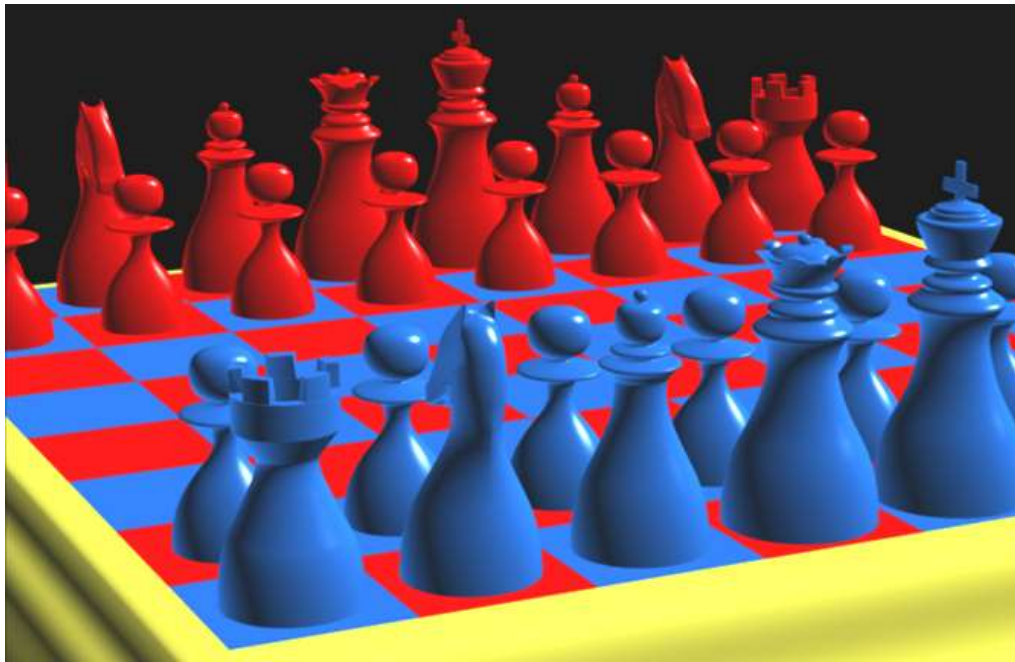
HANYANG ERICA CAMPUS

Q YOUN HONG (홍규연)

# Surface Shading



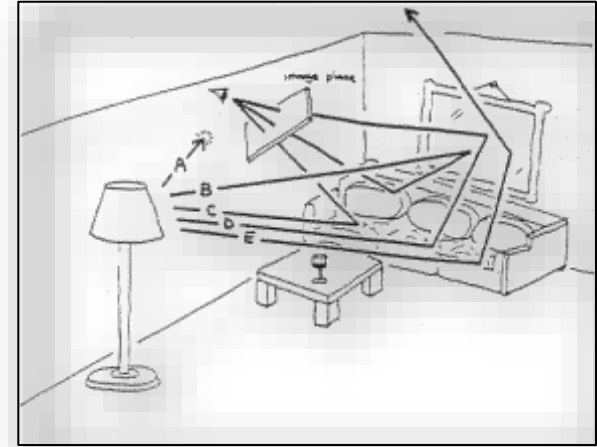
- 물체의 appearance를 결정: 물체의 표면의 색상, 질감, 재질 등을 표현



Glossy vs. Matte Objects

# Lighting vs. Shading

- Lighting: material들과 light source(광원)들의 상호작용
  - 물리 법칙이 연관되어 있음
- Shading: pixel의 색을 결정
  - 컴퓨터 그래픽스의 한 요소
- Shading은 대개 lighting에 의해서 결정되지만, lighting에 의해 다른 방식으로 가시화될 수도 있음
  - 예: Non Photorealistic Rendering(NPR)



ZBuffer(Scene)

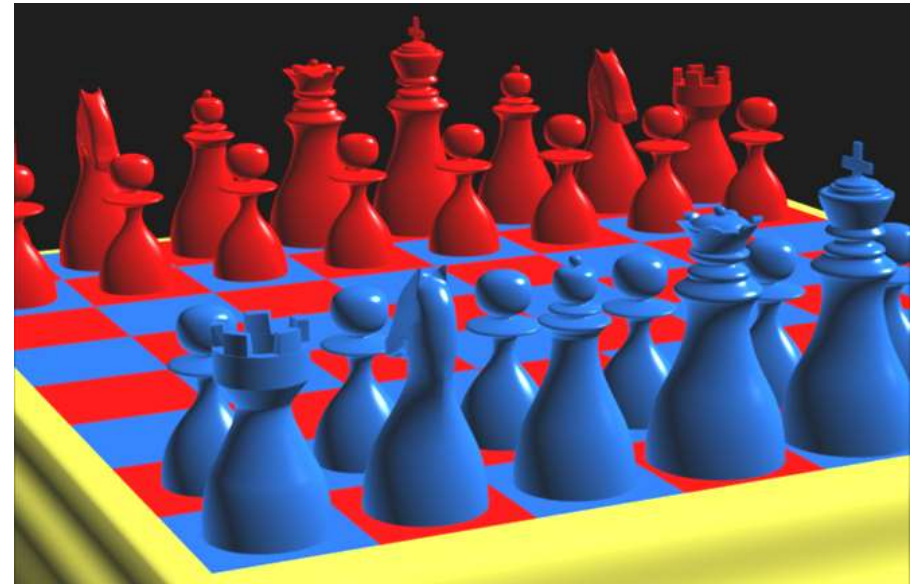
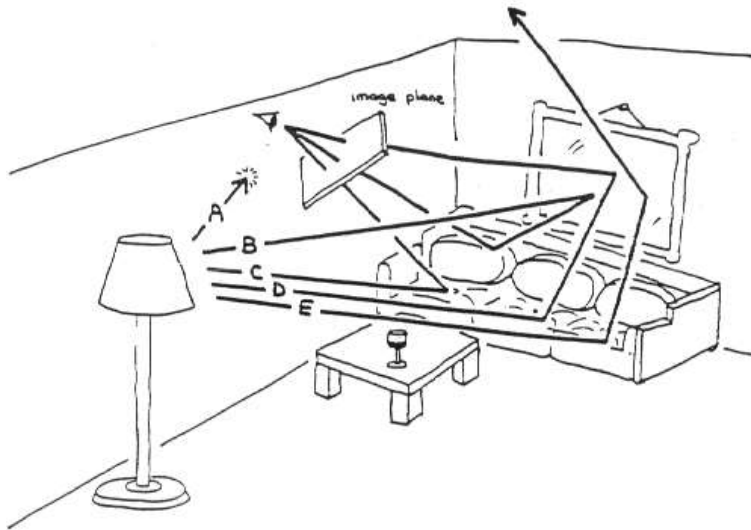
```
...  
    PutColor(x,y,Col(P));  
...  
end
```



# Shading Models



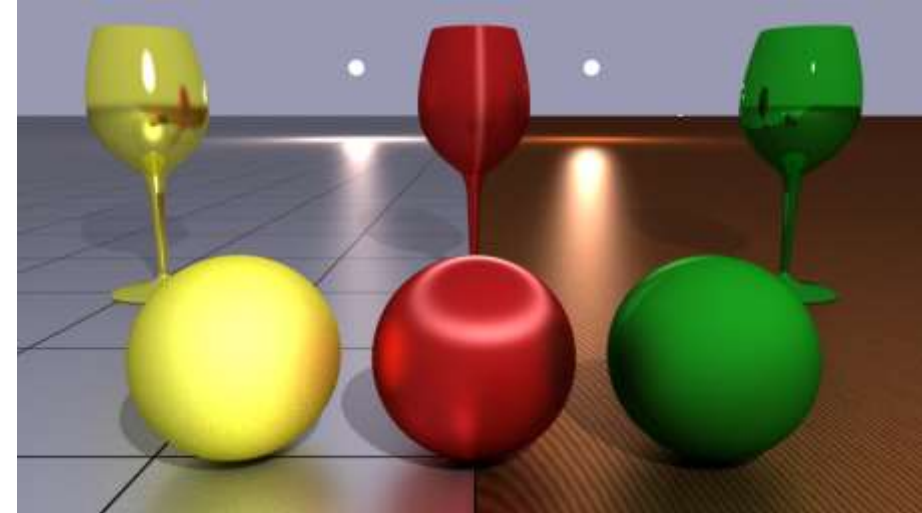
- 물체가 어떻게 보이는지, 물리적인 현상을 시뮬레이션
  - 정확한 illumination을 시뮬레이션하는 것은 복잡하고 고비용임
  - 물리 법칙에 기반한 근사(approximation)와 heuristics을 사용
  - Shading Model을 사용해도 좋은 가시화 결과를 보임



# Local vs. Global Illumination Models



- **Local model** – 각각의 물체는 빛과 직접적이고 지역적인 상호작용을 보임
- **Global model** – 다른 물체들 간에도 빛 에너지의 교환과 상호작용이 이루어짐

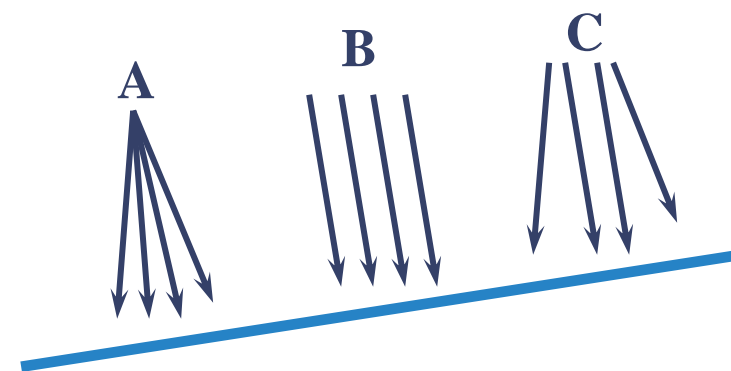




# Light Sources (광원)



- **Point source (A):** 모든 빛이 한 점에서 출발함
  - 광선이 평면에 다른 입사각(incidence angle)으로 닿음
- **Parallel source(B):** 모든 광선이 평행함
  - 광선이 평면에 같은 입사각으로 닿음
  - 무한대(infinity)에 위치한 점 광원(point source)으로 모델링 가능
  - Directional light source
- **Area source(C):** 빛이 공간상에 있는 특정 영역에서 출발
  - 광원의 특정 영역(finite area)에서 빛을 발광
  - Distributed source



# Shading Component



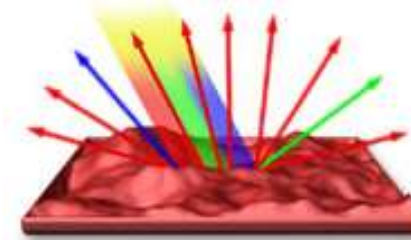
- **Specular reflection**

- On Metallic (smooth) surface
- 빛이 정의된 특정 각도로 반사됨



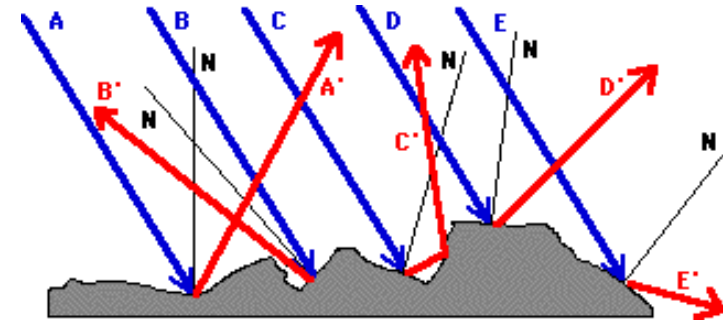
- **Diffuse reflection**

- On Plastic (rough) surface
- 빛이 모든 방향으로 반사됨



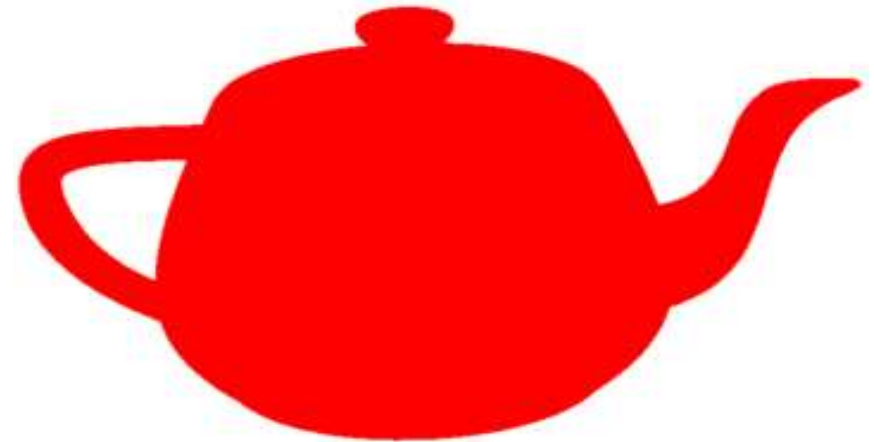
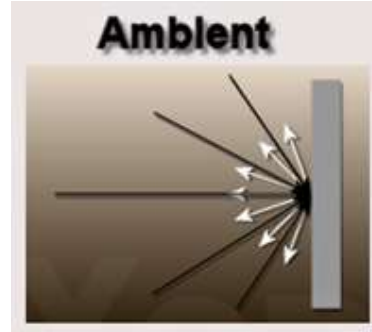
- **Ambient light**

- 빛이 여러 번 반사되어 모든 방향에서 같은 양으로 들어오는 것처럼 인식



# Ambient Light

- 환경 상에서 방향성이 없는 빛으로 가정
- 모든 장소에서 같은 빛으로 물체를 비추는 것처럼 보여짐
  - 마치 실루엣 (Silhouette)처럼 보여지게 됨
- Illumination equation:  $I = I_a k_a$ 
  - $I_a$  – ambient light intensity
  - $k_a$  – ambient light가 surface에서 반사되는 비율
  - RGB별로 따로 정의되어 물체의 색을 결정  
( $I = (I_{ar} \cdot k_{ar}, I_{ag} \cdot k_{ag}, I_{ab} \cdot k_{ab})$ )

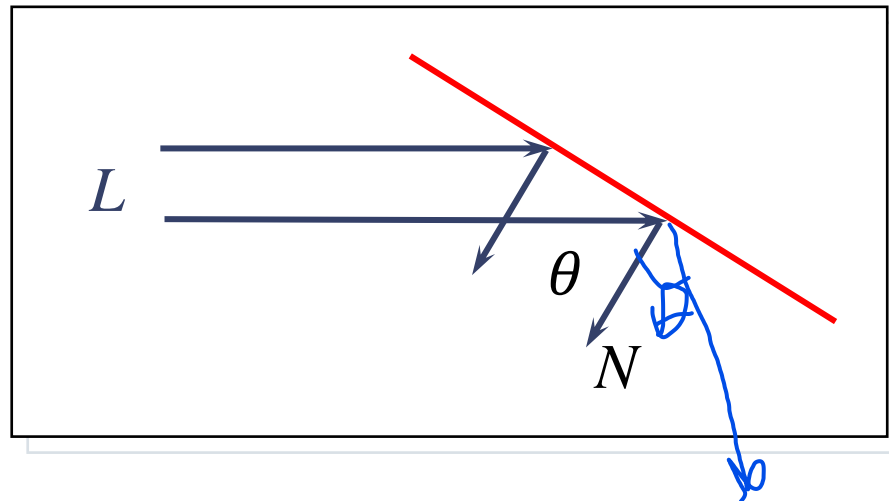
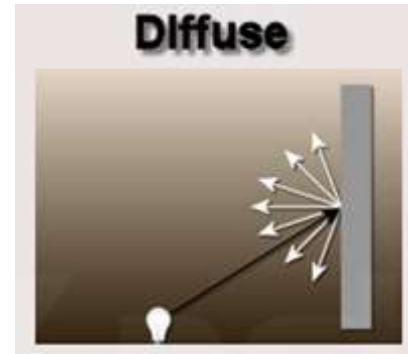




# Diffuse Light



- **matte plastic**과 같은 dull surface는 모든 방향에서 오는 빛을 같은 방식으로 반사
- Diffuse reflection 또는 Lambertian reflection
- 광원이 (normalized된)  $L$  방향에서 들어오고, surface의 법선 벡터가  $N$ 일 때, surface의 빛의 세기는  $\langle N, L \rangle$ 에 비례함



$\frac{1}{N \cdot L} \cos \theta$

# Diffuse Reflection



- Illumination equation becomes...

$$I = I_a k_a + I_p k_d \langle N, L \rangle = I_a k_a + I_p k_d \cos \theta$$

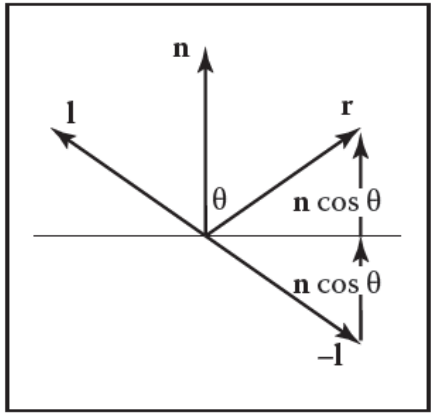
- $I_p$  – point light source의 세기(intensity)
- $k_d$  – surface의 diffuse reflection coefficient



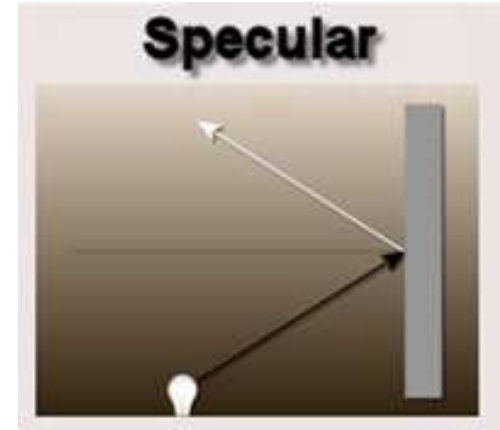
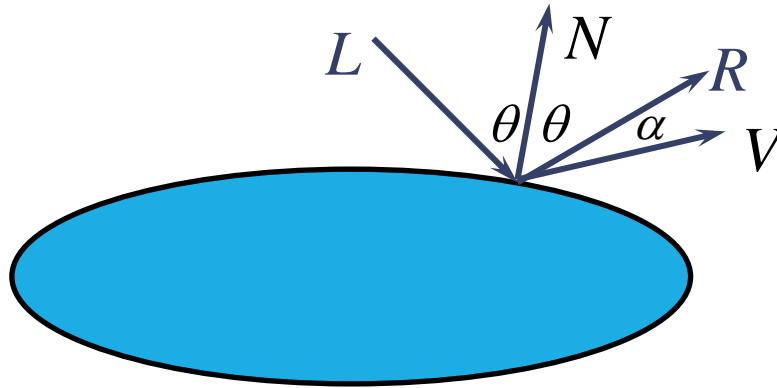
# Specular Reflection



- 빛나는 물체(예: **metallic**)는 빛을 surface의 법선 벡터  $N$ 에 의해서 결정되는 특정 방향  $R$ 로 빛을 반사함

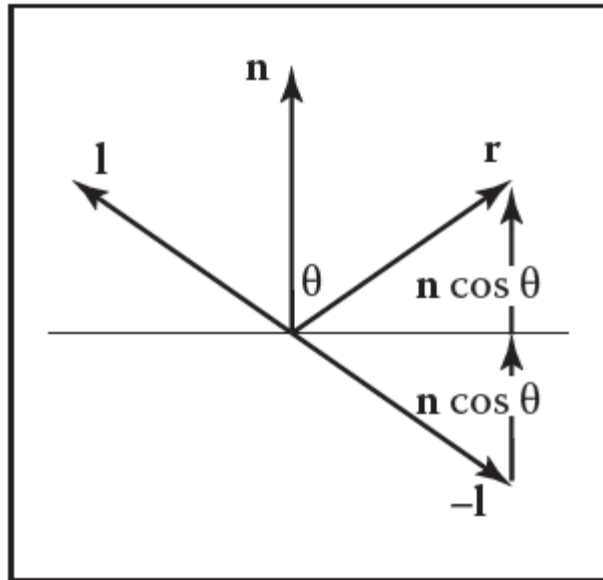


$$R = -L + 2\langle L, N \rangle N$$



- 완벽한 거울은  $L$  방향으로 들어온 빛을  $R$  방향으로 100% 반사함
- 대부분의 물체는 완벽한 거울이 아님 - **빛을  $R$  방향의 부근으로 반사**
- Phong Model – 빛의 경감(attenuation)을  $\cos^n \alpha$ 의 형태로 모델

# Specular Reflection



$$r = -l + 2\langle l, n \rangle n$$

# Specular Reflection



- Illumination equation becomes...

$$I = I_a k_a + I_p (k_d \langle N, L \rangle + k_s \langle R, V \rangle^n)$$

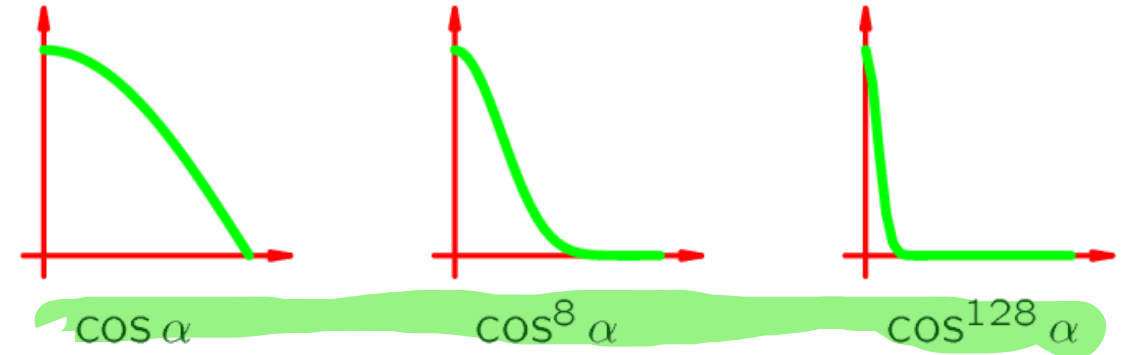
- $k_s$  – Specular reflection coefficient
- $n$  – Specular exponent



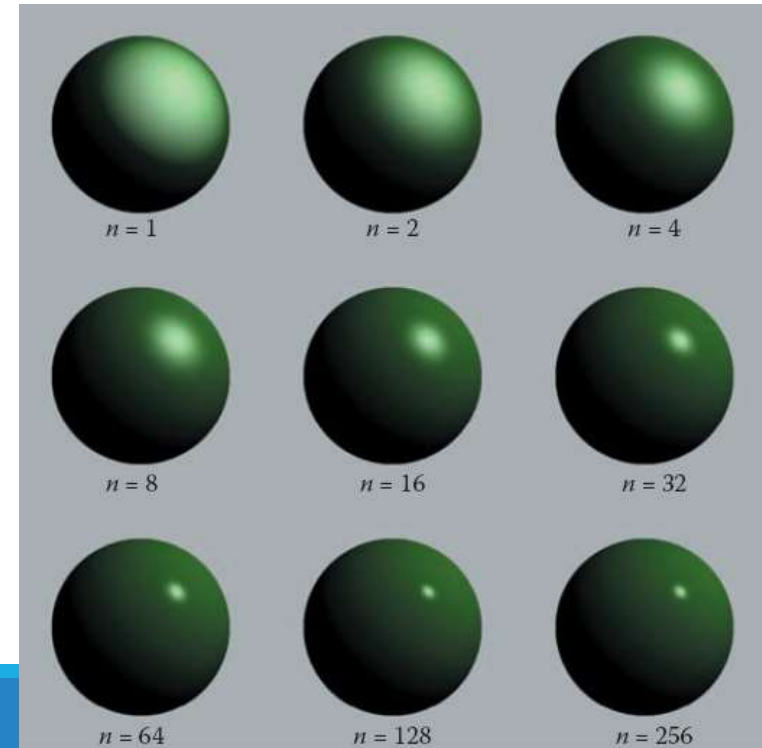
# Specular Reflection



- Exponent  $n$ 이 attention의 경감 정도 (decay factor)를 결정



- 물리적 기반은 없지만 관측을 결과로 보임

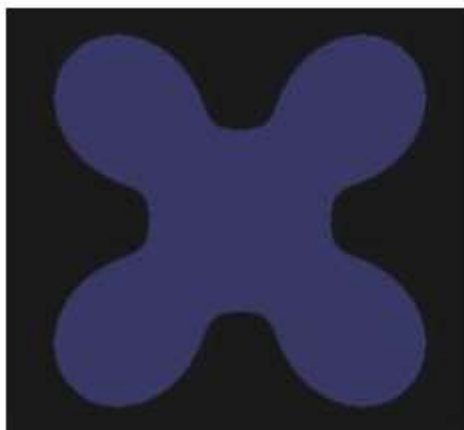
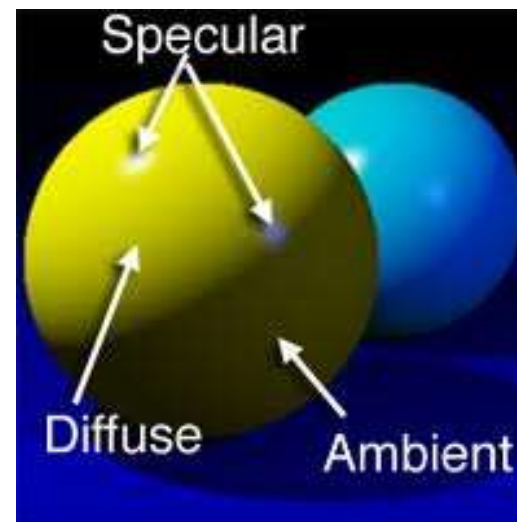




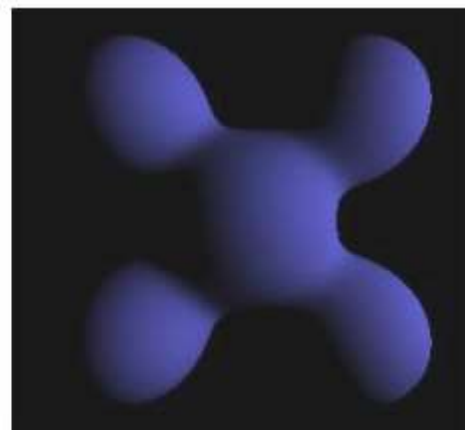
# Illumination Equation

- 전체 Illumination Equation

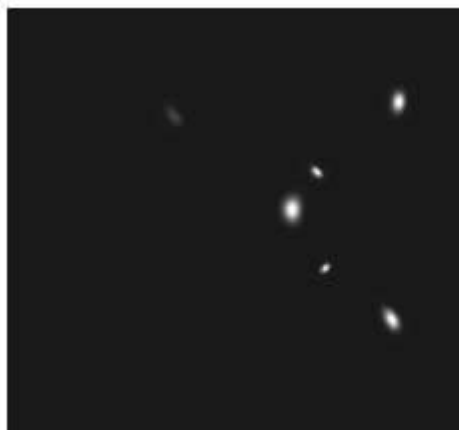
$$I = I_a k_a + I_p (k_d \langle N, L \rangle + k_s \langle R, V \rangle^n)$$



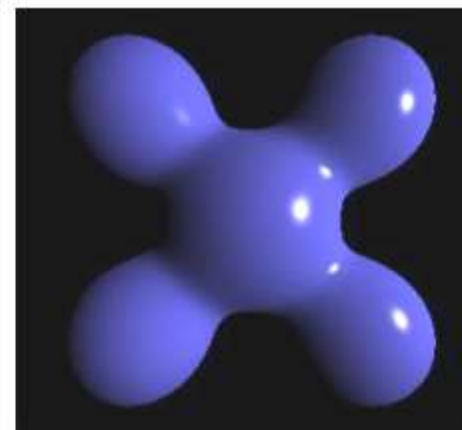
Ambient



Diffuse



Specular



= Phong Reflection

# Illumination Equation

- 여러 개의 light source(광원)이 존재할 때:

$$I = I_a k_a + \sum_p I_p (k_d \langle N, L \rangle + k_s \langle R, V \rangle^n)$$

- 모든 광원들의  $I_p$ 들이 더해짐
- Overflow에 대한 처리 필요

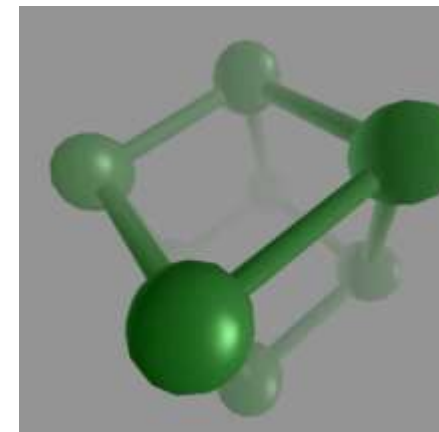
# Illumination Equation

Q) 왼쪽과 같은 atmospheric attenuation은 어떻게 구현?

- 다음의 illumination equation을 이용:

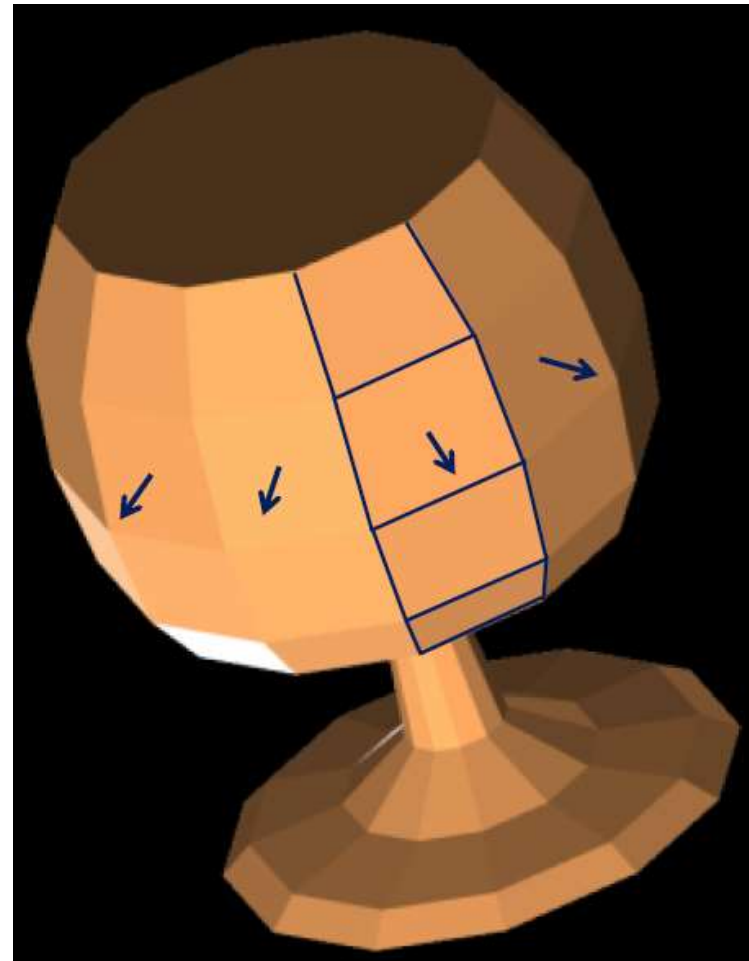
$$I = I_a k_a + \sum_p \frac{I_p}{d_p} (k_d \langle N, L_p \rangle + k_s \langle R_p, V \rangle^n)$$

- $d_p$  – surface와 light source 간의 거리 또는 surface와 viewer 사이의 거리 (heuristic atmosphere attenuation)



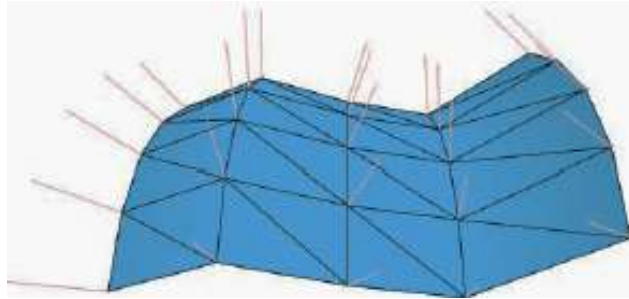
# Flat Shading

- piecewise linear polygonal model에 적용
  - polygon에 적용할 수 있는 단순한 surface lighting
  - illumination 값은 각각의 polygon의 법선 벡터에 의해 결정 됨
- ⇒ 각각의 polygon은 같은 intensity로 칠해짐
- 부드럽게 보이지 않음

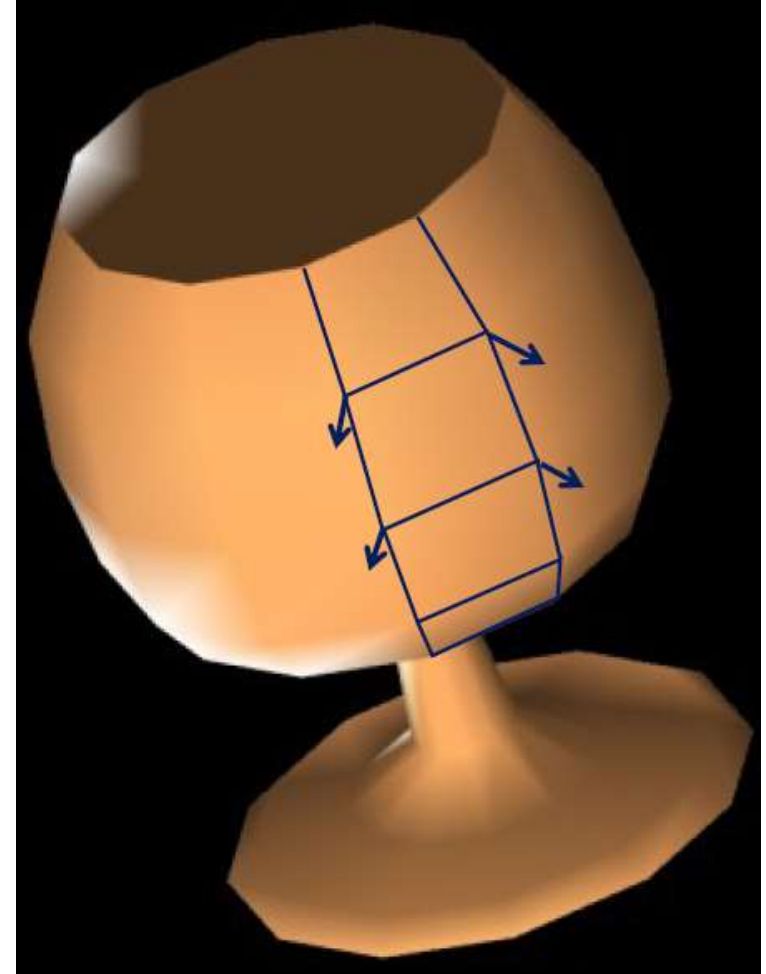


# Gouraud Shading

- 만일 polygon이 부드러운 곡면의 근사라면:
  - 물체의 각 vertex에 법선 벡터를 부여할 수 있음
  - 각 vertex의 법선 벡터를 원래 곡면에서 가져올 수 있음
  - 만일 원래 곡면이 존재하지 않으면 법선 벡터를 추정



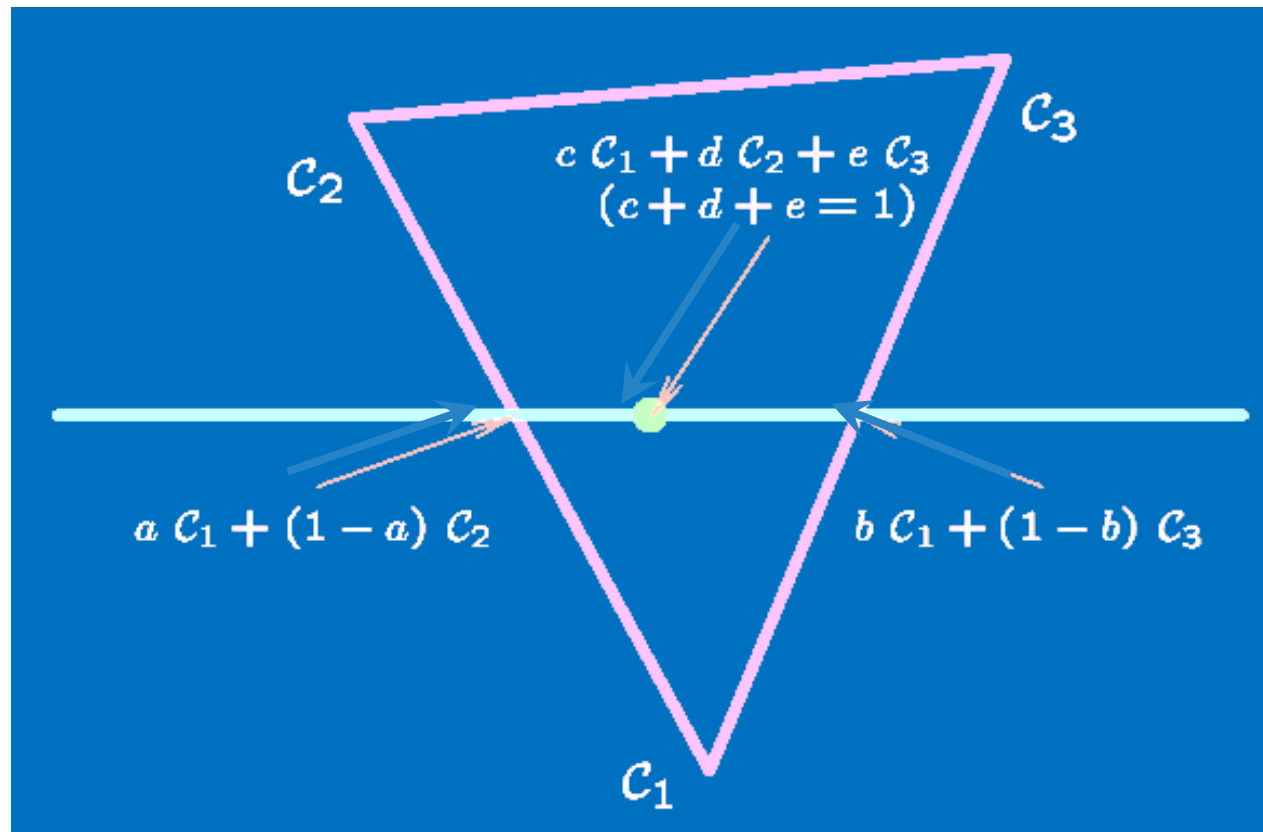
- vertex normal을 이용해서 illumination intensity를 각 vertex에서 계산



# Gouraud Shading



Polygon의 내부에 있는 pixel의 lighting intensity를 구하기 위해  
서 vertices들에서 계산한 **lighting intensity(RGB)**를 선형 보간함



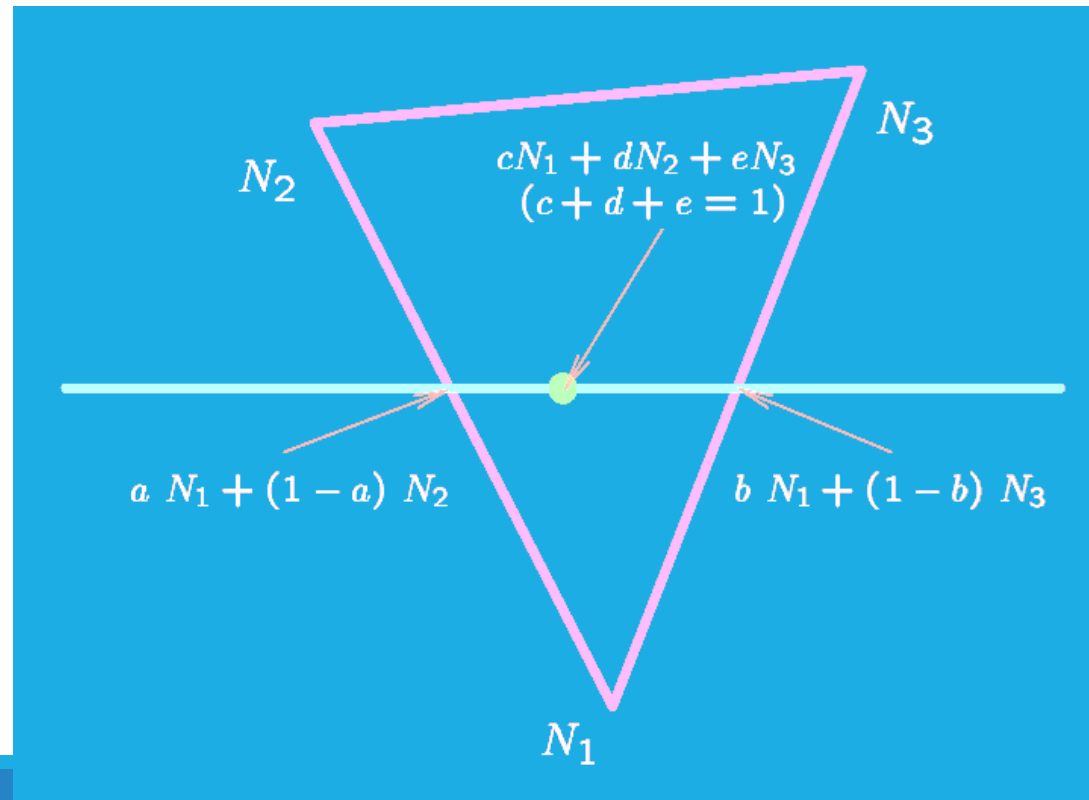
→ Barycentric  
coordinates



# Phong Shading



- illumination intensity들 대신에 법선 벡터들을 선형 보간함
- polygon의 내부에 있는 pixel들에 대해서 pixel에 해당하는 (보간된) normal을 이용해서 illumination equation을 계산함



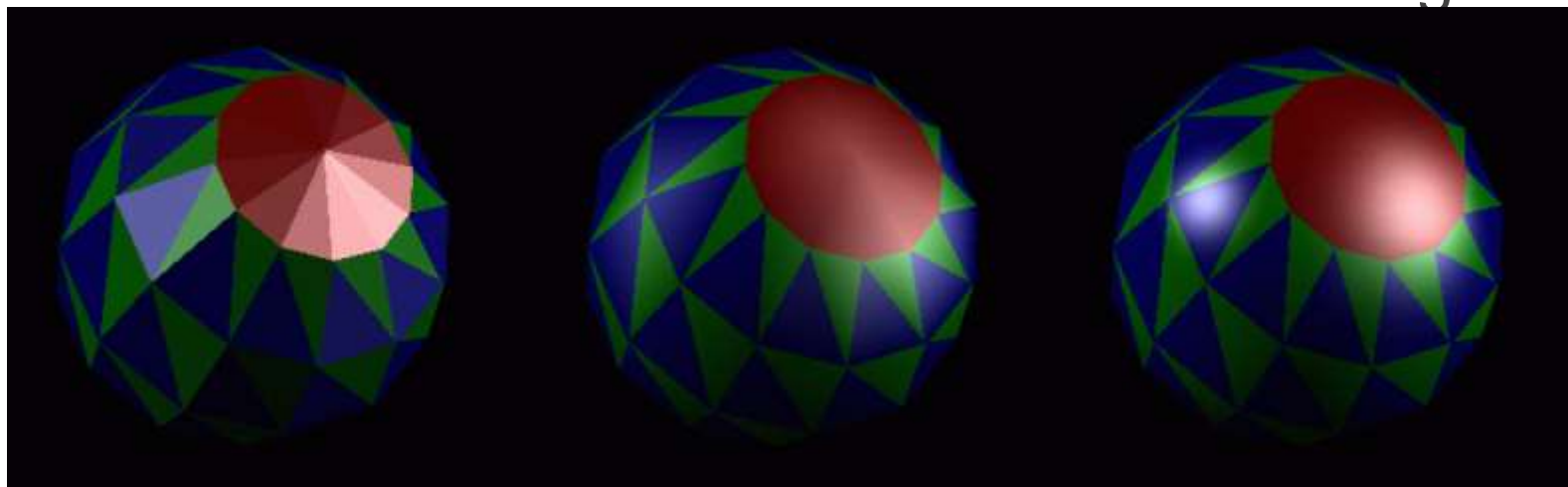
# More Examples



Flat

Gouraud

Phong





# Shading



- Phong shading은 Gouraud Shading에 비해서 비쌈
- specular highlight effect를 잘 구현할 수 있음
- Gouraud와 Phong Shading은 모두 image plane에서 이루어짐
- Gouraud와 Phong Shading은 view dependent
- Transformation dependent – Animation중에 작은 artifact 있을 수 있음