

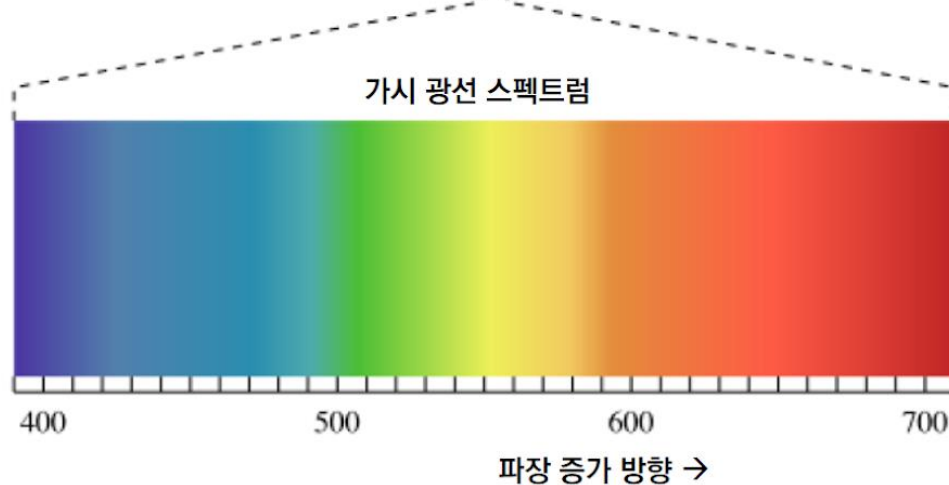
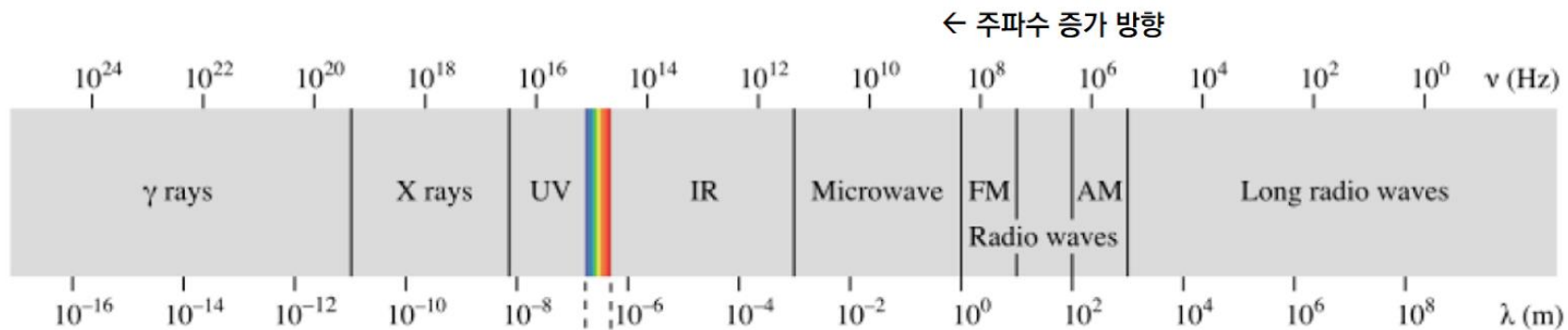
색과 조명

동명대학교 게임공학과

강영민

가시광선

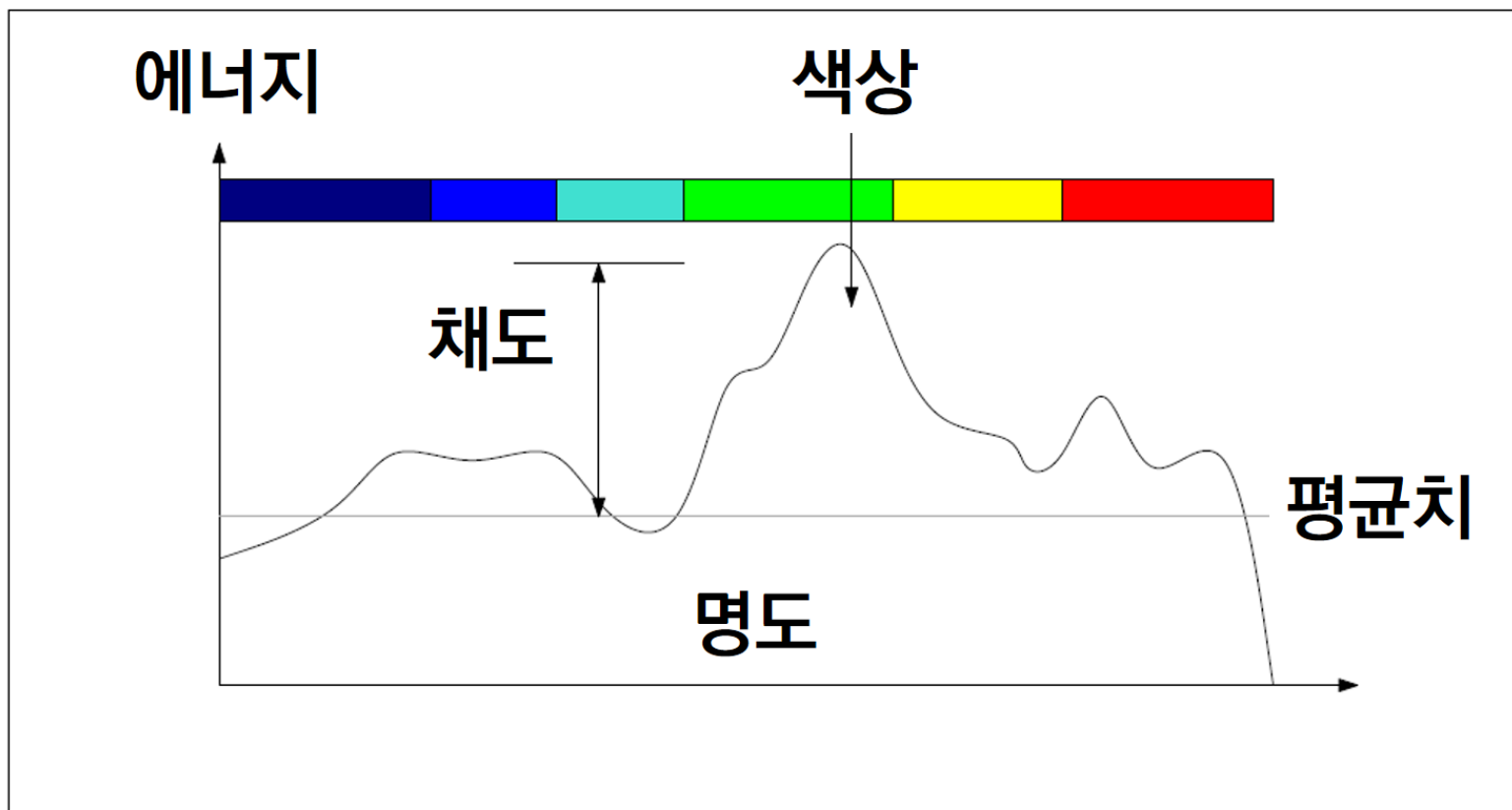
- 전자기파와 가시광선



390 nm에서 720 nm의 범위

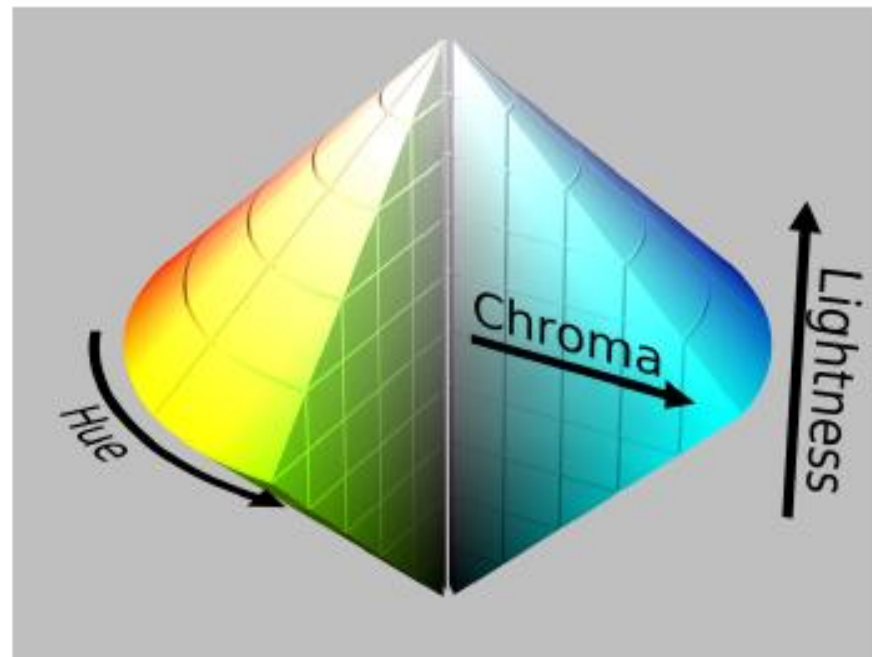
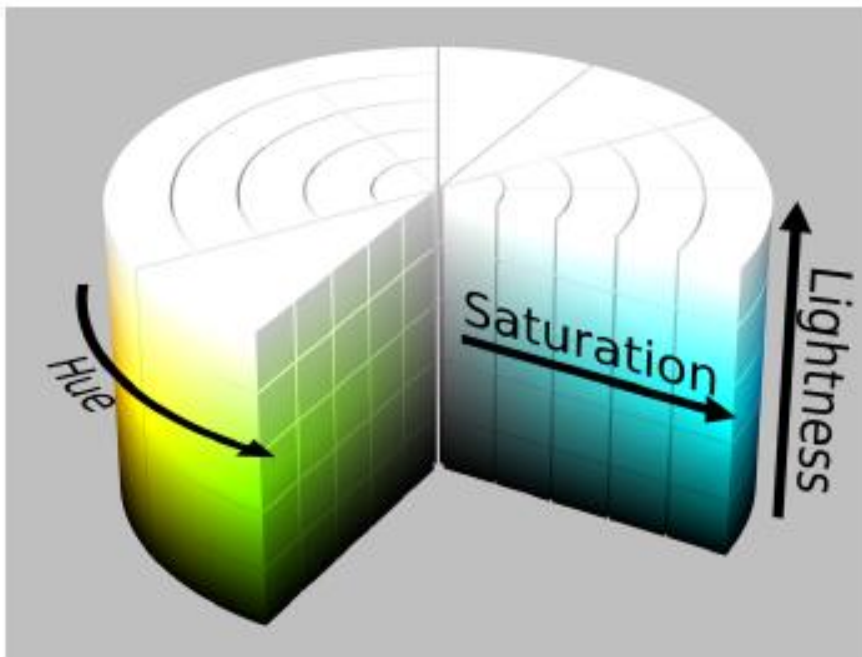
색에 대한 이해

- 스펙트럼 별 에너지와 색의 감각



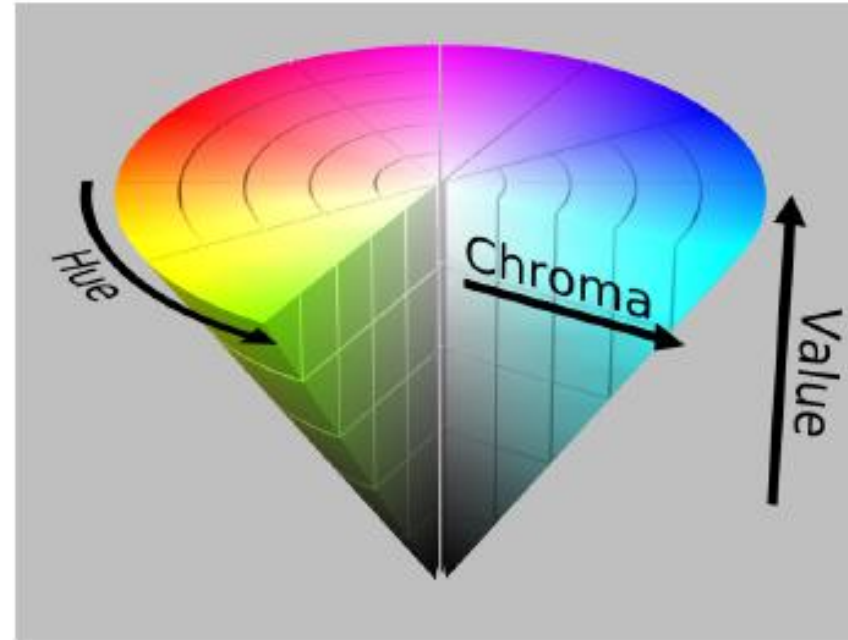
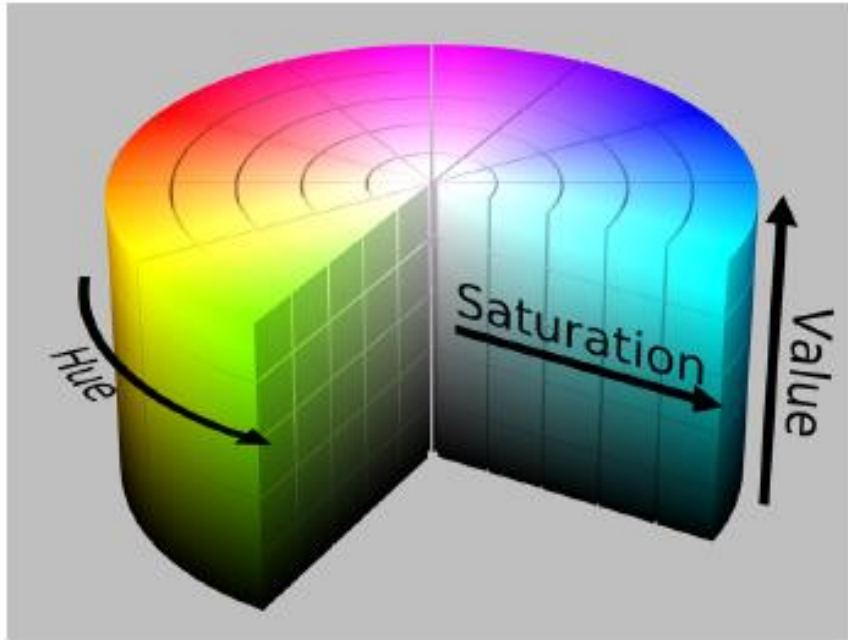
색모델

- HSL – Hue, Saturation, Lightness



색모델

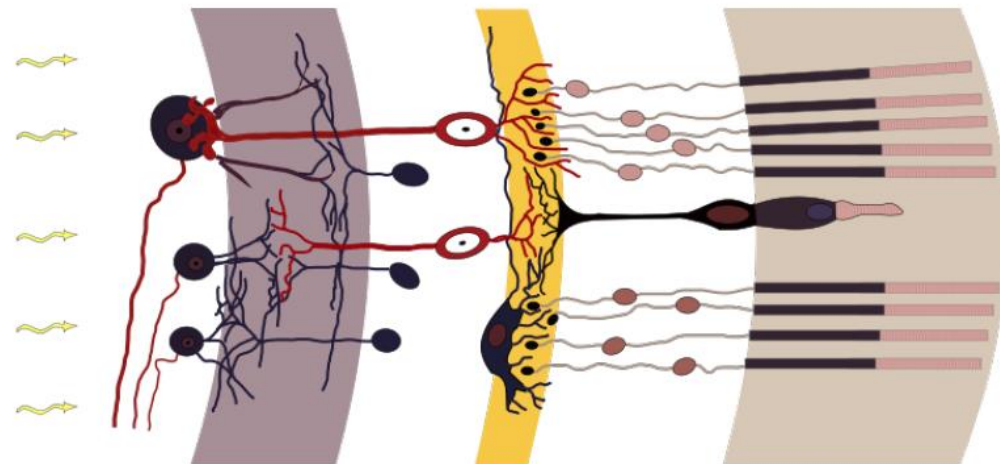
- HSV – Hue, Saturation, Value



HSL의 'L'과 HSV의 'V' 모두 가장 낮은 값일 때는 검정색을 표현한다는 점에서는 같다. 하지만 'L'이 가장 큰 값에 이르면 색이 흰색이 되는데 반해, 'V'가 최대치에 이르면 가장 채도가 높은 선명한 색이 된다.

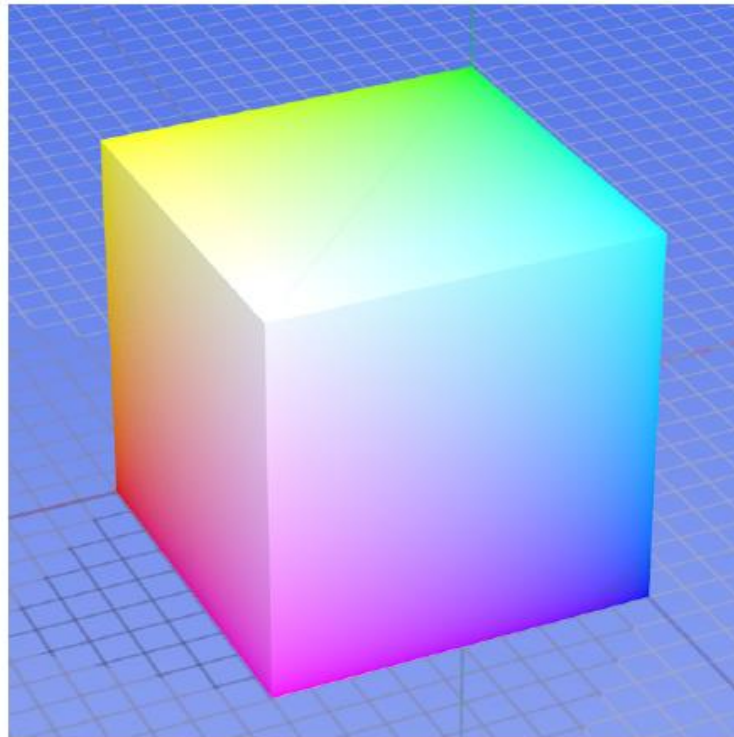
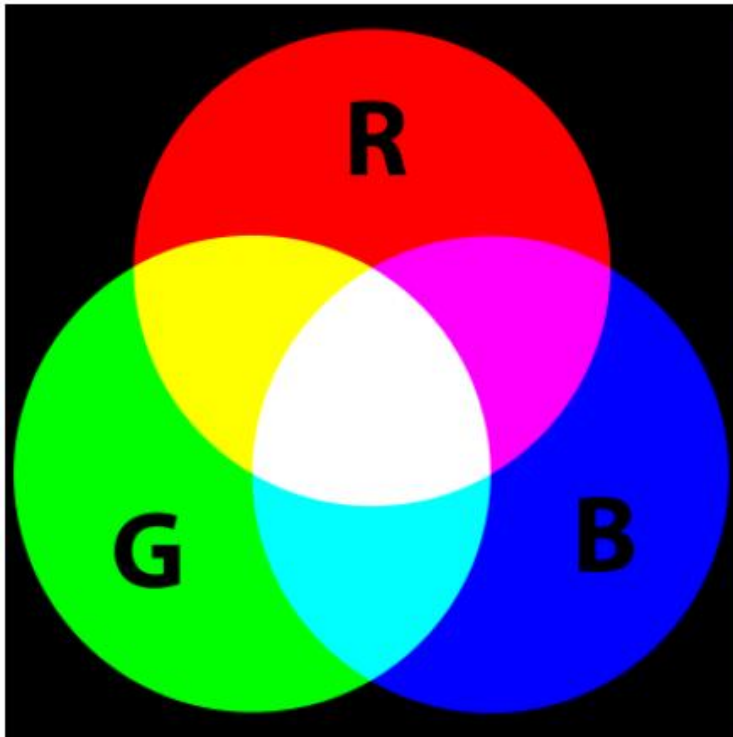
RGB 색모델과 삼중 자극 이론

- RGB 색 모델
 - 삼중자극(三重刺激, tristimulus) 이론에 근거하
 - 하나의 색을 세 종류의 색의 합성으로 보는 것
 - 요소가 되는 세 색은 빨강, 녹색, 파랑: R, G, B
 - 색을 인식하는 원추 세포가 이 세 가지 색상에 가장 민감하게 반응
- 눈
 - 각막(角膜, cornea)을 통해 들어온 빛
 - 렌즈의 역할을 하는 수정체^{lens}
 - 망막^{retina}에 상을 맺게 함
 - 망막의 세포
 - 명암을 인식하는 막대^{rod} 세포
 - 색상을 인지하는 원추^{cone} 세포
 - 원추 세포가 빨강, 녹색, 파랑에 민감하게 반응



RGB 색모델 - 가산혼합 모델

- RGB 모델은 그림과 같이 요소 색을 합쳤을 때 명도가 높아지는 가산혼합 모델
- 요소색: Red, Green, Blue → RGB (에너지가 합쳐지므로 밝은 색을 인지하게 됨)

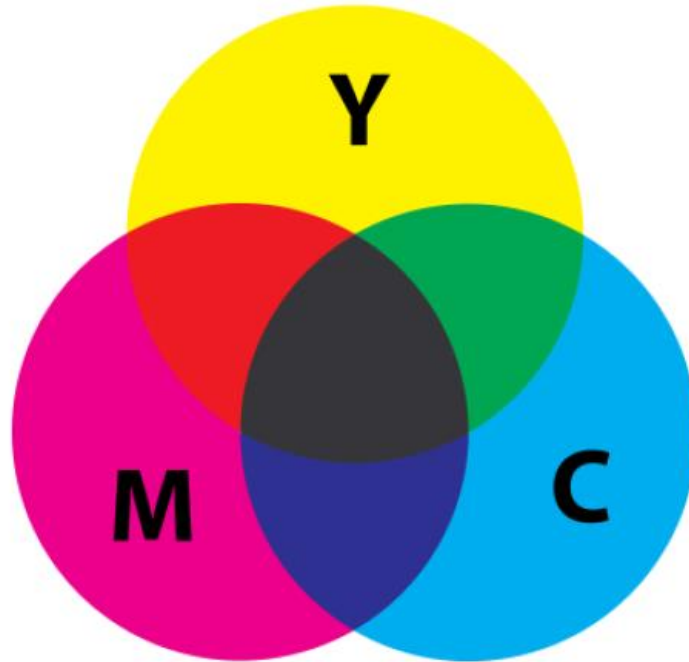


감산혼합 모델 - CMYK

CMYK 모델은 그림과 같이 요소 색을 합쳤을 때 명도가 낮아지는 감산혼합 모델

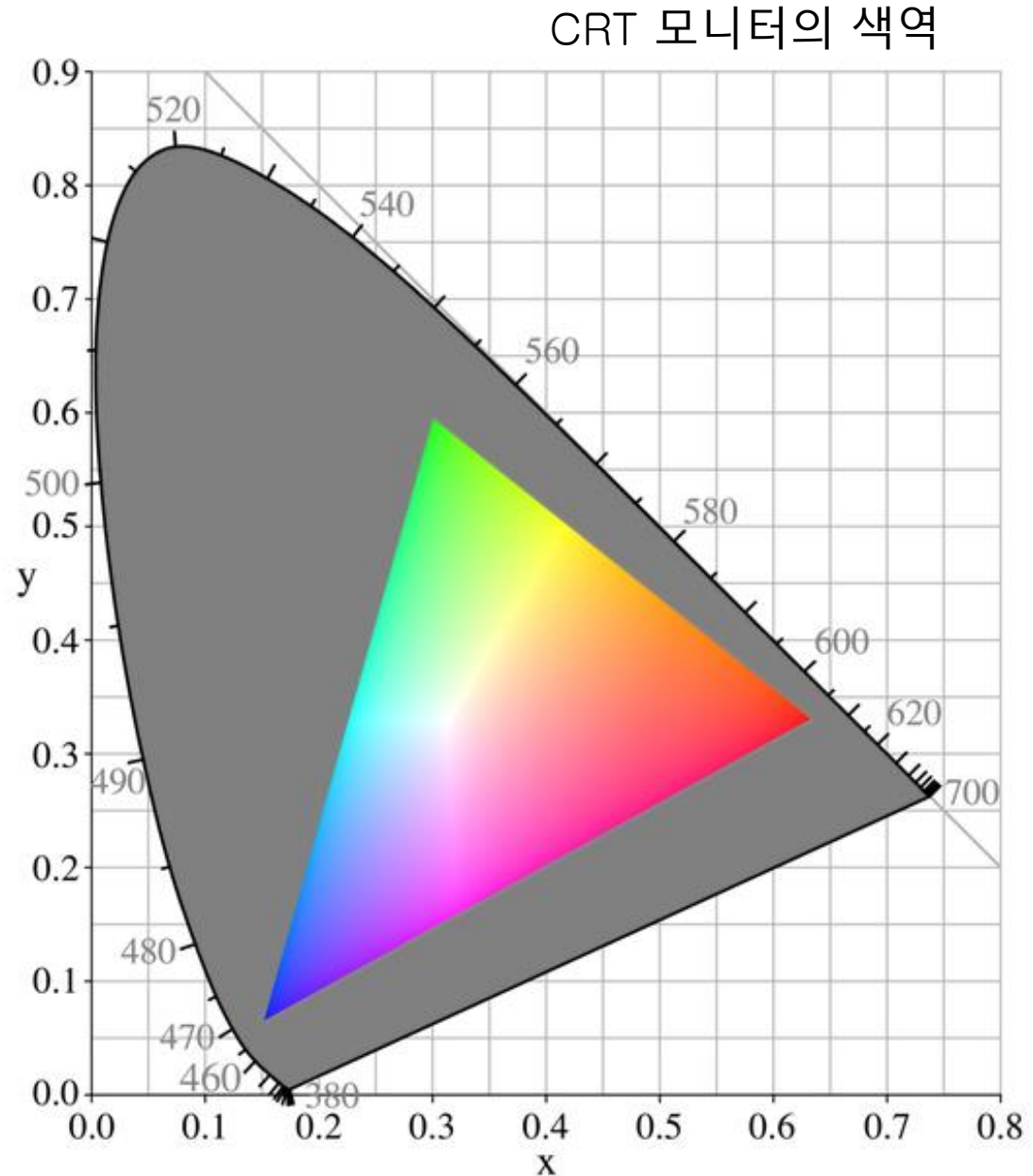
- 요소색: Cyan, Magenta, Yellow, Black → CMYK

- 잉크 모델
- 인쇄에서 사용됨

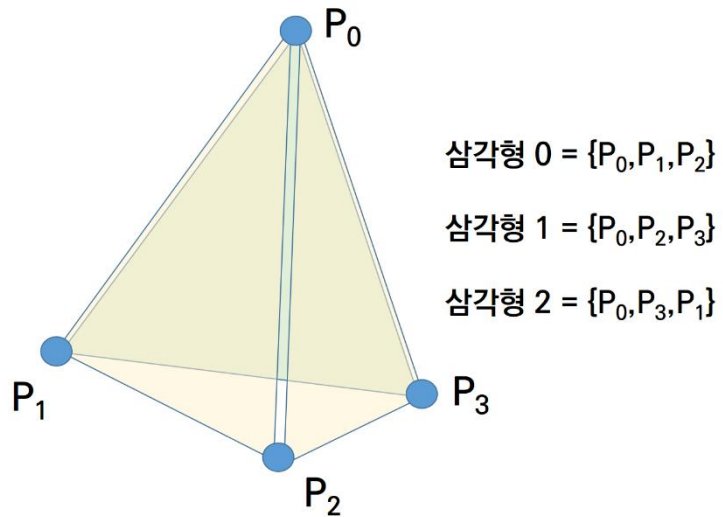


색역 – color gamut

색 모델에 따라서 표현할 수 있는 색의 범위가 제한된다. 이를 색의 범위, 혹은 색역^{color gamut}이라고 부른다. 색역의 표현은 국제조명위원회^{CIE}에서 정한 CIE 1931 색공간으로 표현하는 것이 일반적이다. 오른쪽 그림은 색공간 내에서 CRT 모니터가 갖는 색역을 보이고 있다.

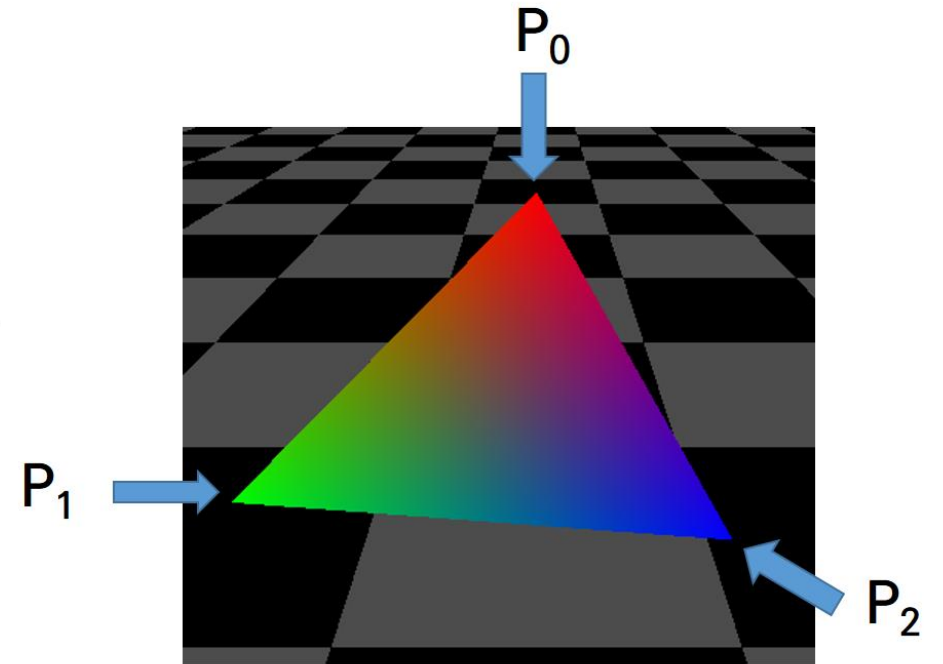


단순한 색상 지정

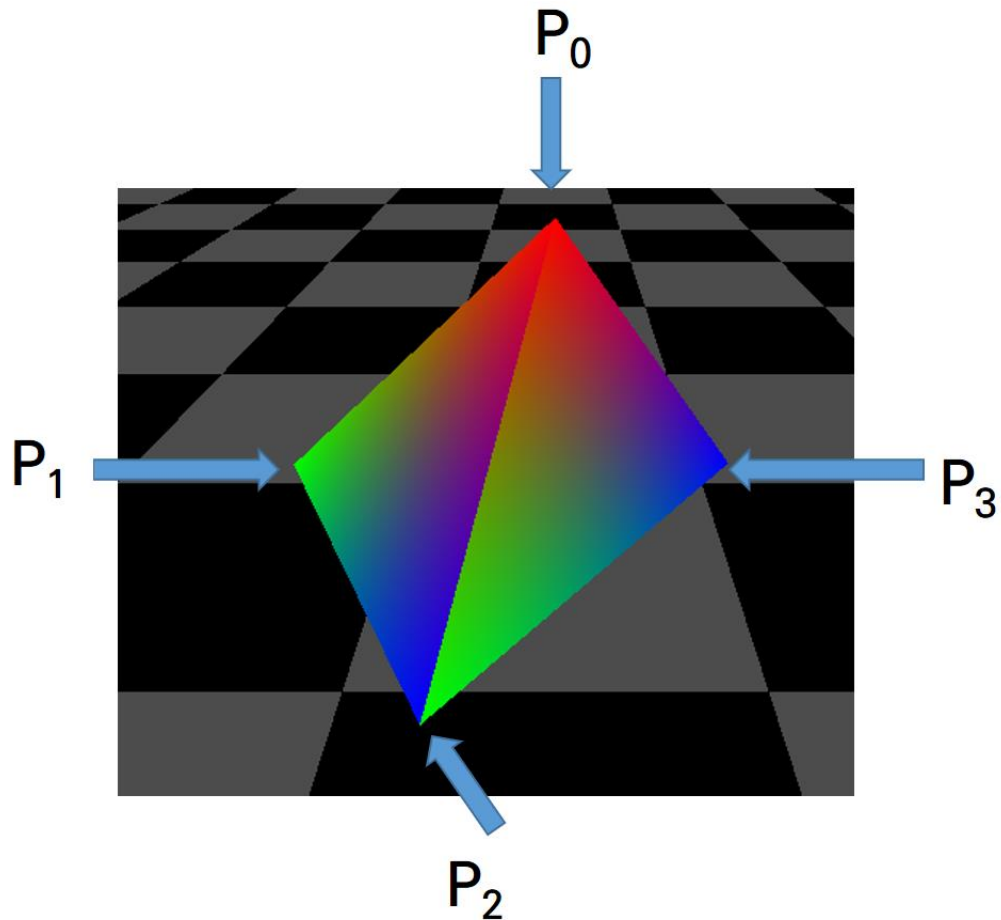


P₀ = [0, 5, 0]
P₁ = [-3, 1, 3]
P₂ = [3, 1, 3]
P₃ = [0, 1, -3]

```
glBegin(GL_TRIANGLES)  
glColor3f(1, 0, 0)  
glVertex3fv(P0)  
glColor3f(0, 1, 0)  
glVertex3fv(P1)  
glColor3f(0, 0, 1)  
glVertex3fv(P2)  
glEnd()
```



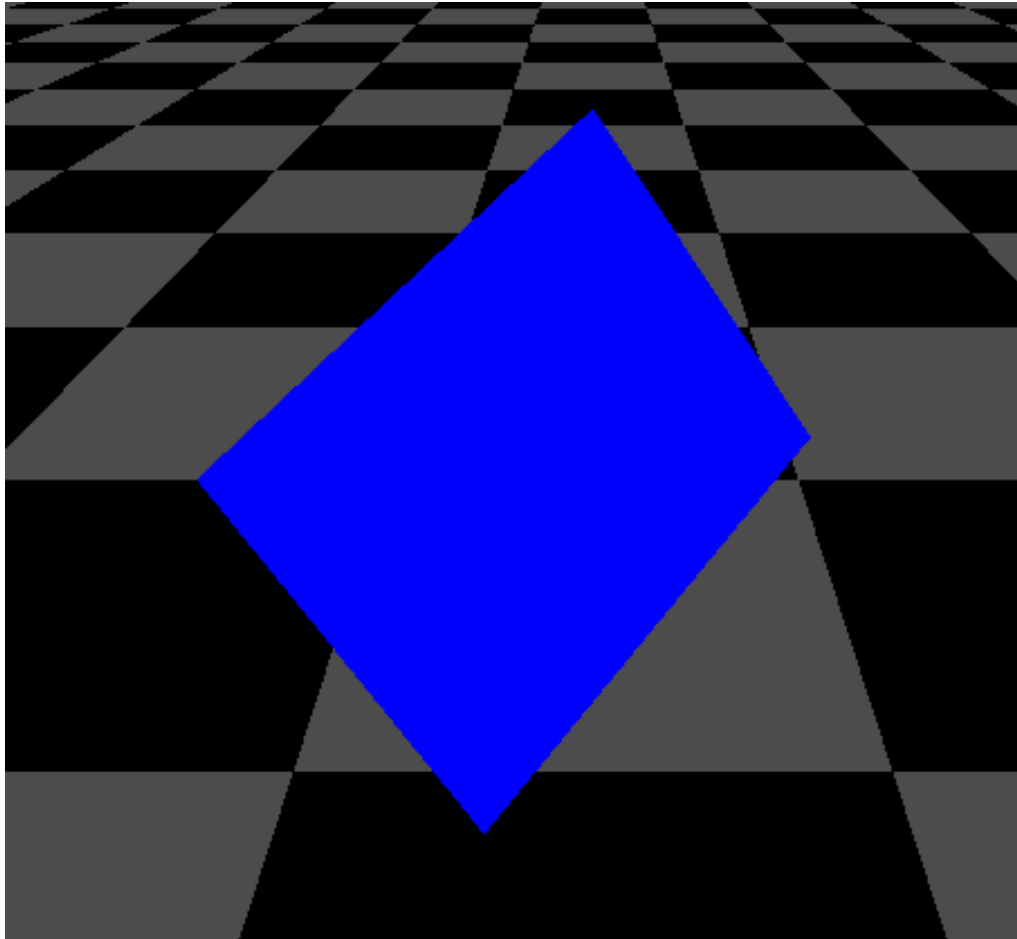
단순한 색상 지정



```
P0 = [0,5,0]  
P1 = [-3,1,3]  
P2 = [ 3,1,3]  
P3 = [ 0,1,-3]
```

```
glRotatef(self.angle, 0, 1, 0)  
self.angle += 1  
glBegin(GL_TRIANGLES)  
glColor3f(1, 0, 0)  
glVertex3fv(P0)  
glColor3f(0, 1, 0)  
glVertex3fv(P1)  
glColor3f(0, 0, 1)  
glVertex3fv(P2)  
  
glColor3f(1, 0, 0)  
glVertex3fv(P0)  
glColor3f(0, 1, 0)  
glVertex3fv(P2)  
glColor3f(0, 0, 1)  
glVertex3fv(P3)  
  
...  
glEnd()
```

단순한 색상 지정

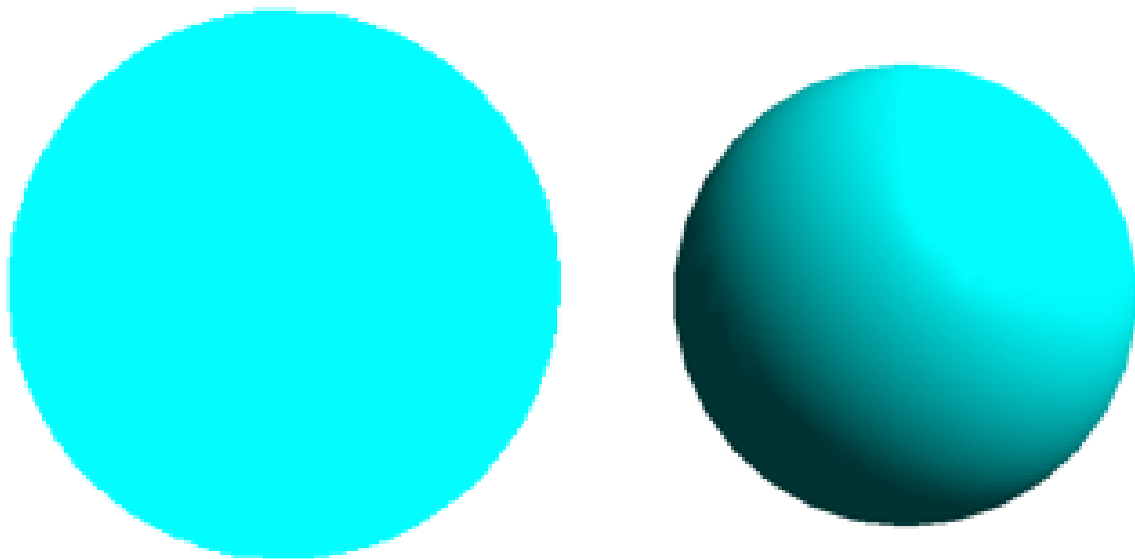


```
glShadeModel(GL_FLAT)
```

```
glBegin(GL_TRIANGLES)  
glColor3f(1, 0, 0)  
glVertex3fv(P0)  
glColor3f(0, 1, 0)  
glVertex3fv(P1)  
glColor3f(0, 0, 1)  
glVertex3fv(P2)
```

기본적으로
`glShadeModel(GL_SMOOTH)`
방식으로 동작

조명모델



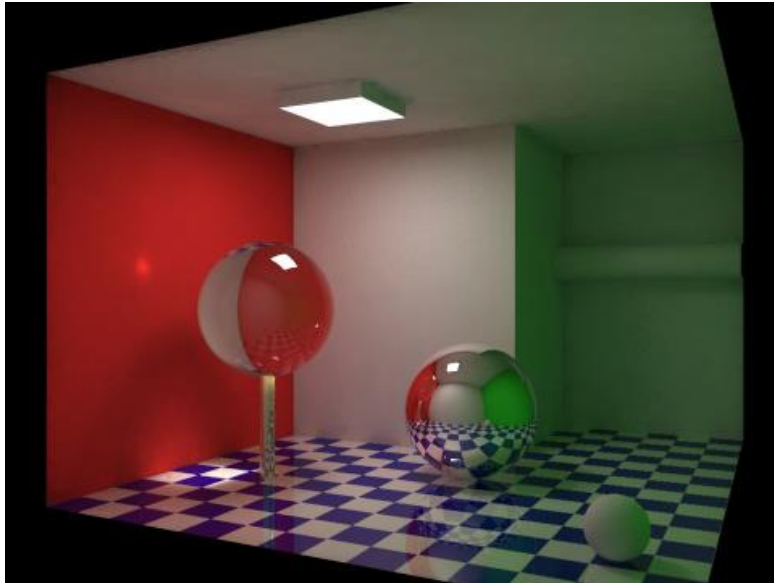
입체감이 없는 구와 입체감이 있는 구의 예

입체감은 무엇을 통해 얻을 수 있을까?

- 양안의 시차 (parallax)
- 그림자(shadow)
- 음영(shading)

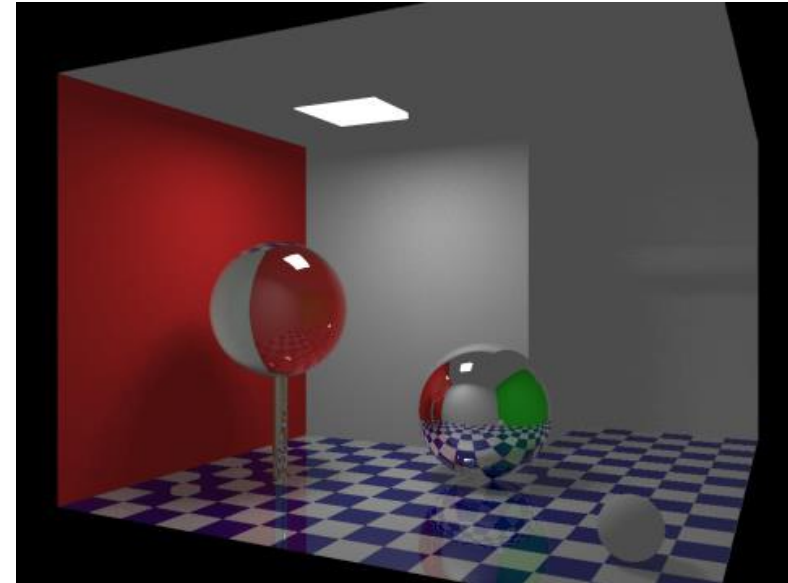
조명모델

전역조명(global illumination)



- 그림자(shadow)
- 음영(shading)

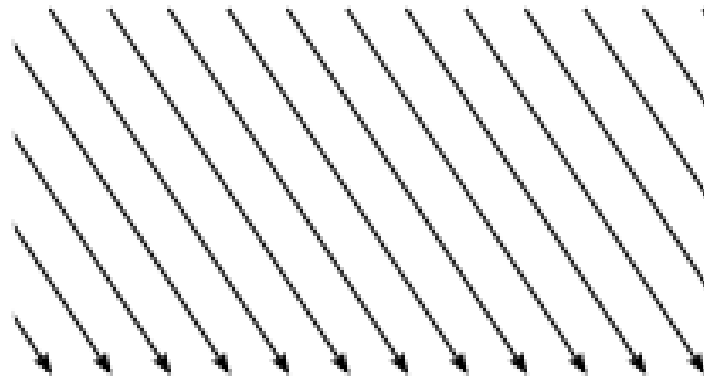
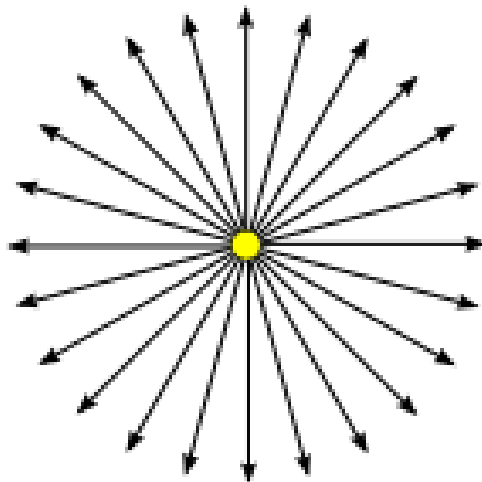
지역조명(local illumination)



- 음영(shading)
- * 그림자를 표현해도 제한적으로 표현

광원

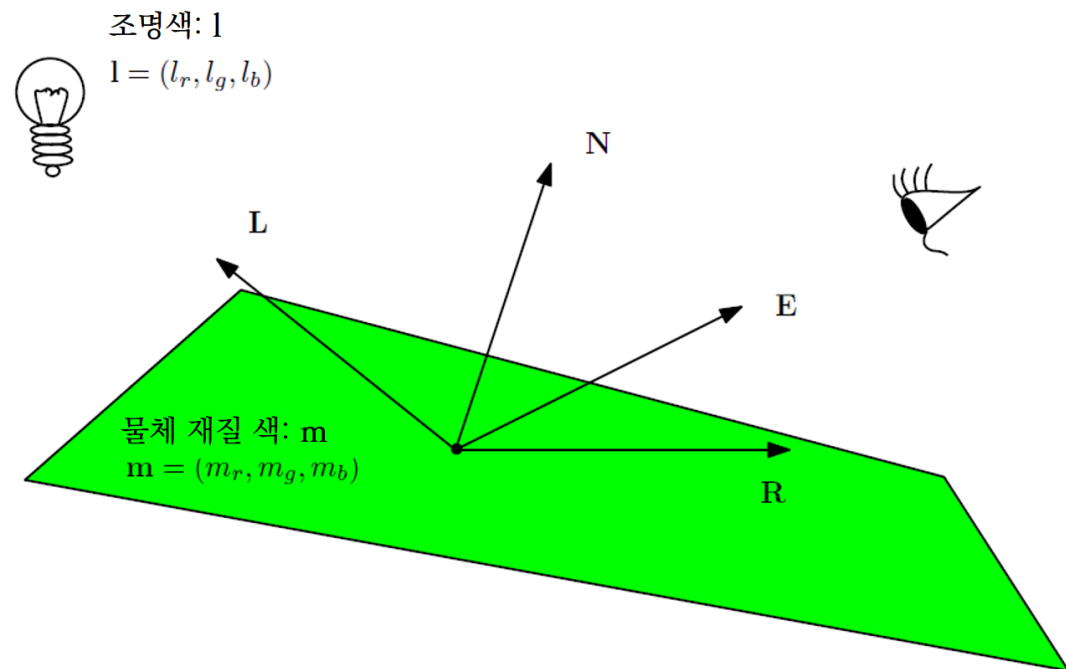
종류	특징
점 광원	광원의 위치와 색으로 결정. 전방향(omni-directional)으로 빛 진행
집중 광원	점광원과 동일하나 빛의 진행을 일부 입체각으로 빛을 제한
방향 광원	특정한 위치가 아니라 방향에 광원 존재
주변 광원	모든 곳에 동일하게 가해지는 빛



지역조명 - 뽕 모델

표면의 밝기 = 다음 밝기 요소들의 합

- 난반사 (diffuse reflection)
- 정반사 (specular reflection)
- 주변광 반사 (ambient reflection)



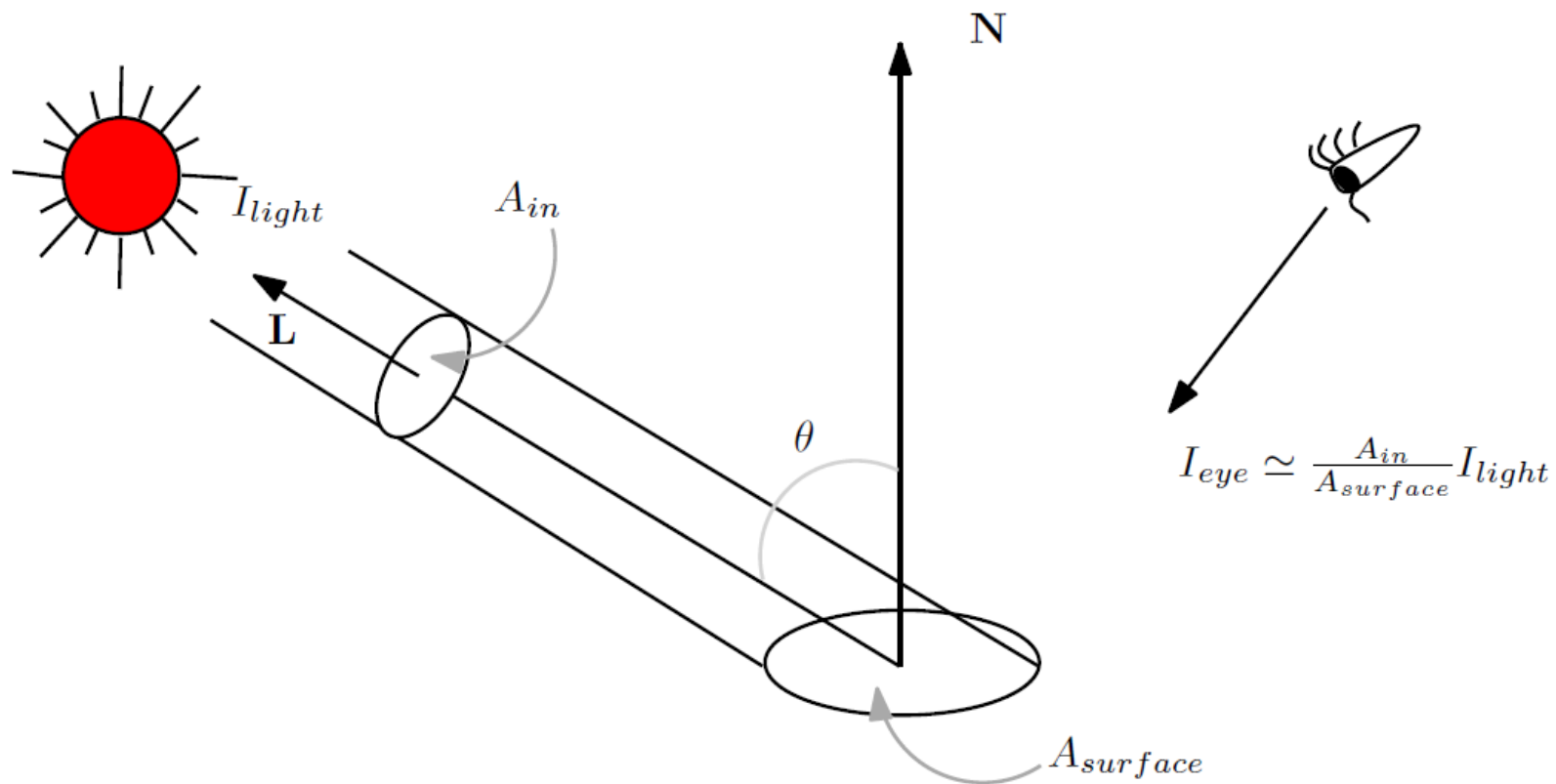
계산에 필요한 요소들

- 광원으로 향하는 벡터 \mathbf{L}
- 카메라(시점)을 향하는 벡터 \mathbf{E}
- 법선 벡터 \mathbf{N}
- 이상적인 반사 방향 \mathbf{R}

$$\kappa = \mathbf{l}_a \otimes \mathbf{m}_a + I_d \mathbf{l}_d \otimes \mathbf{m}_d + I_s \mathbf{l}_s \otimes \mathbf{m}_s$$

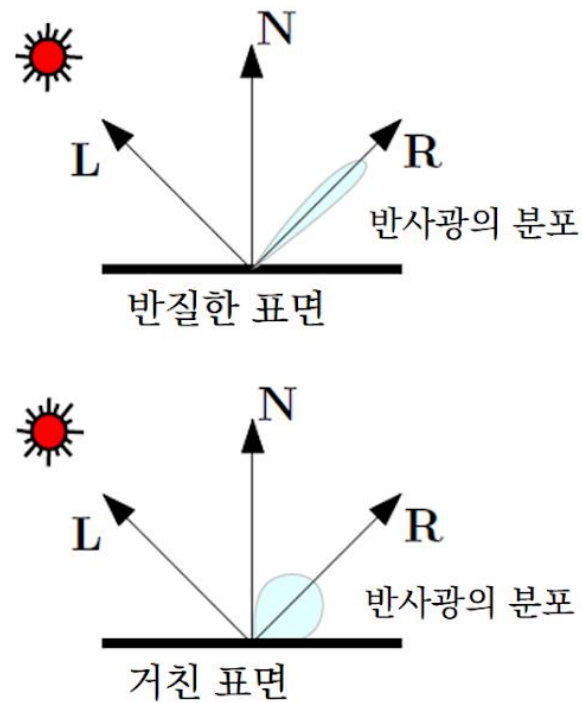
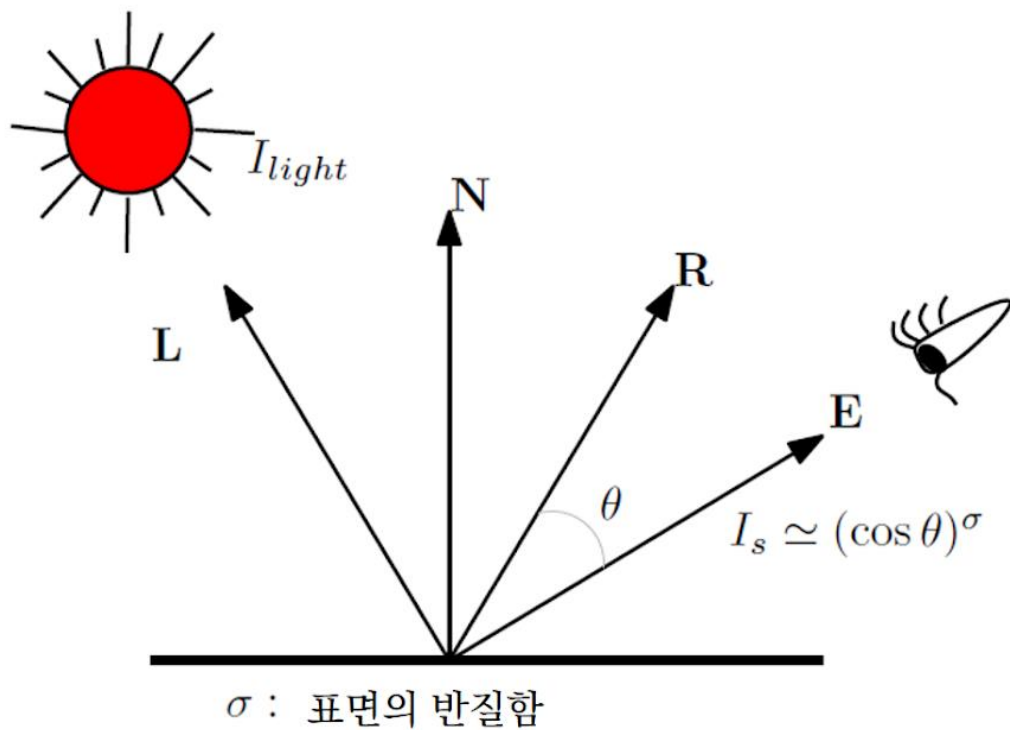
난반사의 계산

$$I_d = \cos \theta = \mathbf{L} \cdot \mathbf{N}$$



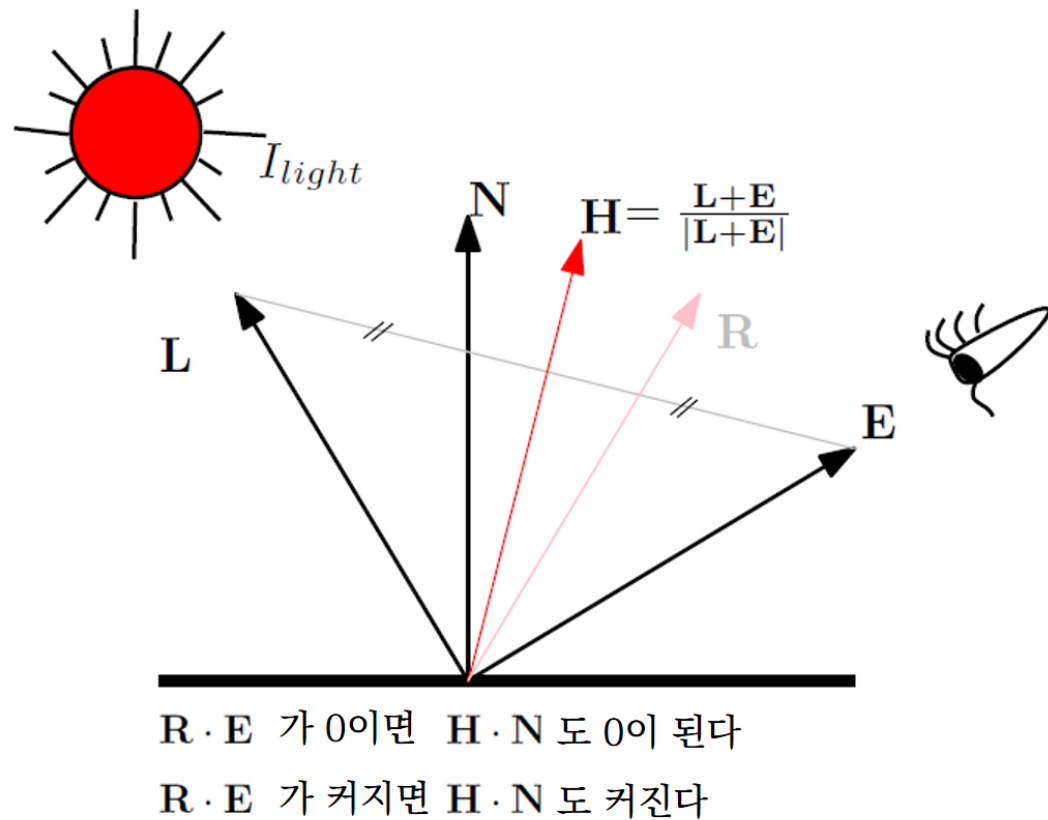
정반사의 계산

$$I_s = \cos\theta = (\mathbf{R} \cdot \mathbf{E})^\sigma$$



수정된 phong 모델 - 블린 모델

$$\mathbf{H} = \frac{\mathbf{L} + \mathbf{E}}{|\mathbf{L} + \mathbf{E}|}$$



$$\kappa = \mathbf{l}_a \otimes \mathbf{m}_a + (\mathbf{L} \cdot \mathbf{N}) \mathbf{l}_d \otimes \mathbf{m}_d + (\mathbf{H} \cdot \mathbf{N})^\sigma \mathbf{l}_s \otimes \mathbf{m}_s$$