

# 컴퓨터그래픽스 (CG)

## HW03\_Ray\_Tracing



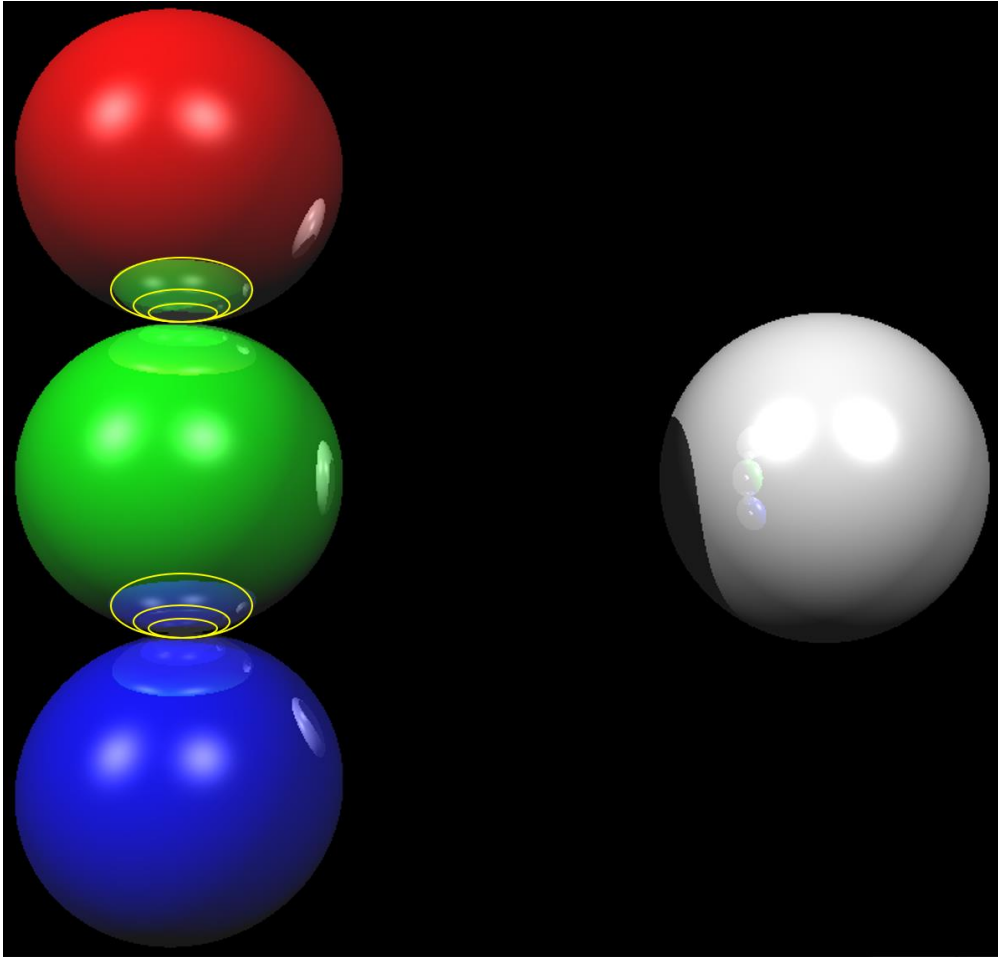
Self-scoring table

	1	2	3	4	Total
Score	1	1	1	1	4

2018707068 김경환

# KwangWoon University

## 1. Ray tracing of reflective (but not translucent) spheres (With a ray tracing depth 4):

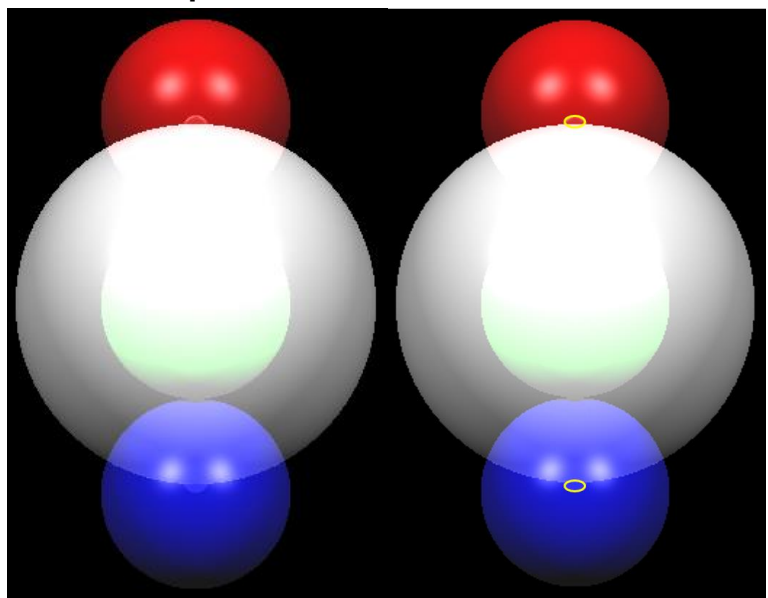
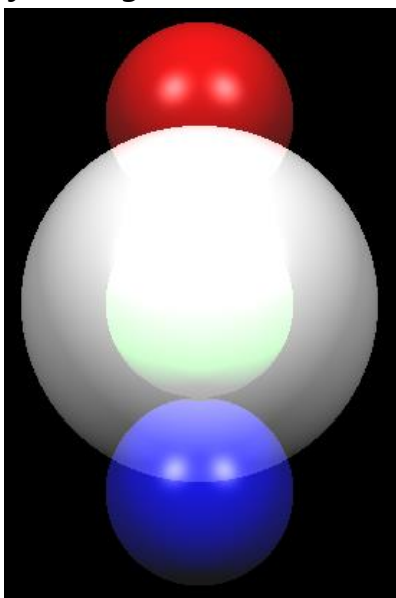


Ray tracing depth가 4이고, translucent 하지 않으며 reflective한 구들의 모습이다.

Depth가 4이기 때문에 recursive하게 ray casting 하여 노란색 동그라미로 표시한 반사된 상이 총 3개가 만들어진 것을 볼 수 있다.

또한, shadow ray를 통한 shadowed를 판단하기 때문에 흰색 구에 그림자가 구현된 것을 볼 수 있다. 마지막으로 구들끼리의 reflection이 구 표면에 표현된 것이므로 반사된 상이 구가 서로 마주보는 곳에 생긴 것을 볼 수 있다.

## 2. Ray tracing of translucent (but not reflective) spheres (Without refraction with $K_T = 0.95$ ):



왼쪽 사진은 Ray tracing depth가 3이고, reflection 하지 않으며 translucent한 구들의 모습이다.

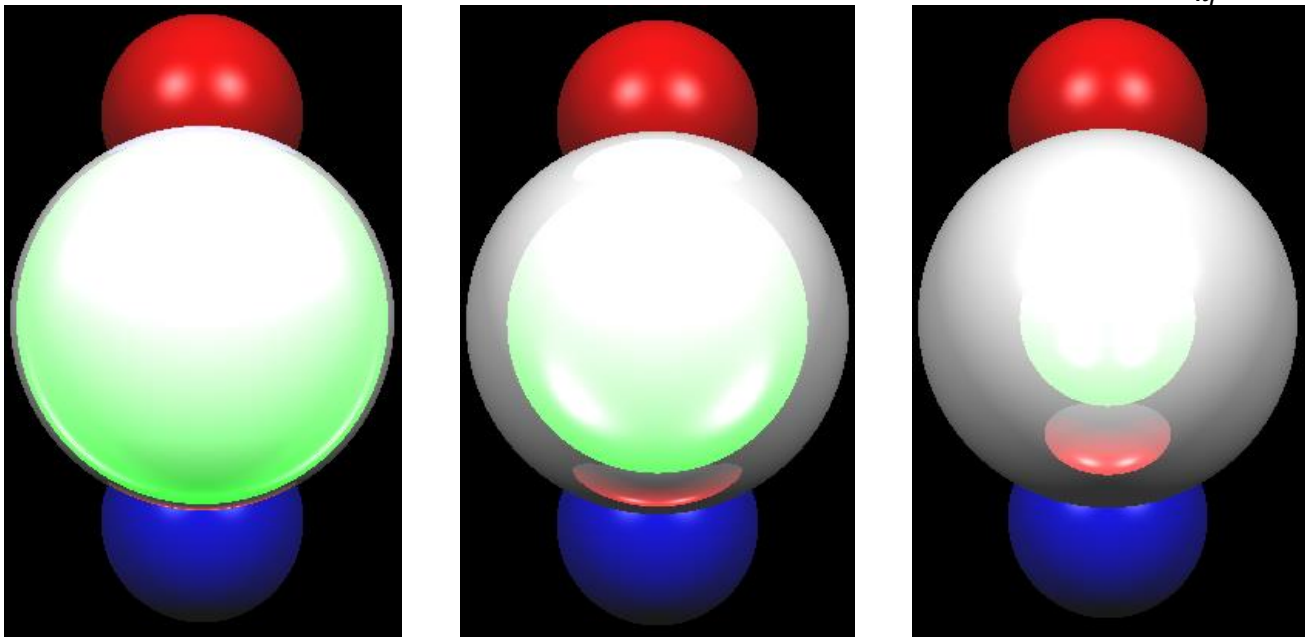
Depth를 3 이상으로 줬기 때문에 빛이 흰색 공을 통과하여 흰색 공 뒤에 겹쳐있는 세 구를 보여주는 것을 확인할 수 있다.

만약 Depth를 2로 줄 경우 흰색 구 내부에서 ray tracing이 끝나므로 겹쳐져 있는 구들의 모습을 볼 수가 없다. 약간의 밝아짐은 확인할 수 있다.

오른쪽 사진은 왼쪽 사진에서 reflection을 추가한 모습이다. Reflection이 작동하기 때문에 translucent 하지 않은 빨간색과 파란색의 구에는 흰색 구가 반사되어 상이 맺힌 것을 확인할 수 있다.

그리고 현재는 relative refractive index가 1이기 때문에 refraction이 일어난다기 보다 transmission된다.

### 3. Ray tracing of translucent (but not reflective) spheres (With a relative refractive index $\frac{n_i}{n_r} = 1 \rightarrow \frac{2}{3}$ ):



위 세 사진은 모두 Ray tracing depth가 3이고, reflection 하지 않으며 translucent한 구들의 모습이다.

다른 점은 왼쪽 사진은 흰색 구 외부의 refractive index가 2.7이고, 가운데 사진은 2.4, 오른쪽 사진은 2이다. 세 사진 모두 구 내부의 refractive index는 3으로 동일하다.

세 사진 모두 relative refractive index가 1이 아닌 다른 값을 갖기 때문에 transmission이 일어난다기 보다 refraction이 일어난다. 그렇기 때문에 흰색 구 외부의 refractive index가 2인 오른쪽 사진의 경우 빨간색과 파란색 구의 상이 반전되어 흰색 구의 아래쪽에 빨간색 구가 표시된 것을 볼 수 있다.

이러한 refraction은 Snell's law ( $n_r \sin \theta_r = n_i \sin \theta_i$ )를 기반으로  $l_t = \left( \frac{n_i}{n_r} \cos \theta_i - \cos \theta_r \right) n - \frac{n_i}{n_r} l$ 에 의해 계산된다.

해당 refraction의 구현 code로는 우선 relative refractive index가 안에서 밖으로, 밖에서 안으로에 따라 바뀌므로 이에 대한 정보를 얻어냈다.

이를 위해서는 Intensity()에서 E(except) parameter를 intersection을 제외하는 용도로 사용하는 것이 아닌 ray가 출발한 구의 index를 나타내는 것으로 사용하였다.

이는 기존 findIntersection()에서 if (I == E) continue;를 주석처리하여 용도에 맞게 설정할 수 있다.

따라서 이제 Intensity()의 E parameter는 ray가 출발한 구의 index를 나타내고, 출발 이후에 intersection이 일어난 구의 index는 Intensity() 내부의 실행 code로써 findIntersection()의 반환값인 iObject가 된다.

따라서 E (ray가 출발한 구)와 iObject (ray가 intersection된 구가 서로 같을 경우 내부이고, 서로 같지

않을 경우 외부가 되게 된다.

그리하여 해당 내부인지 외부인지에 대한 정보를 가지고 Intensity()에서 기존 reflection ray를 recursive 하게 만들었던 것처럼 refraction ray를 만든다.

이제 Refraction ray가 만들어지는 과정은 내부에서 일어나는 refraction인지 외부에서 일어나는 refraction인지에 따라 relative refractive index를 서로 역수 관계를 갖는다.

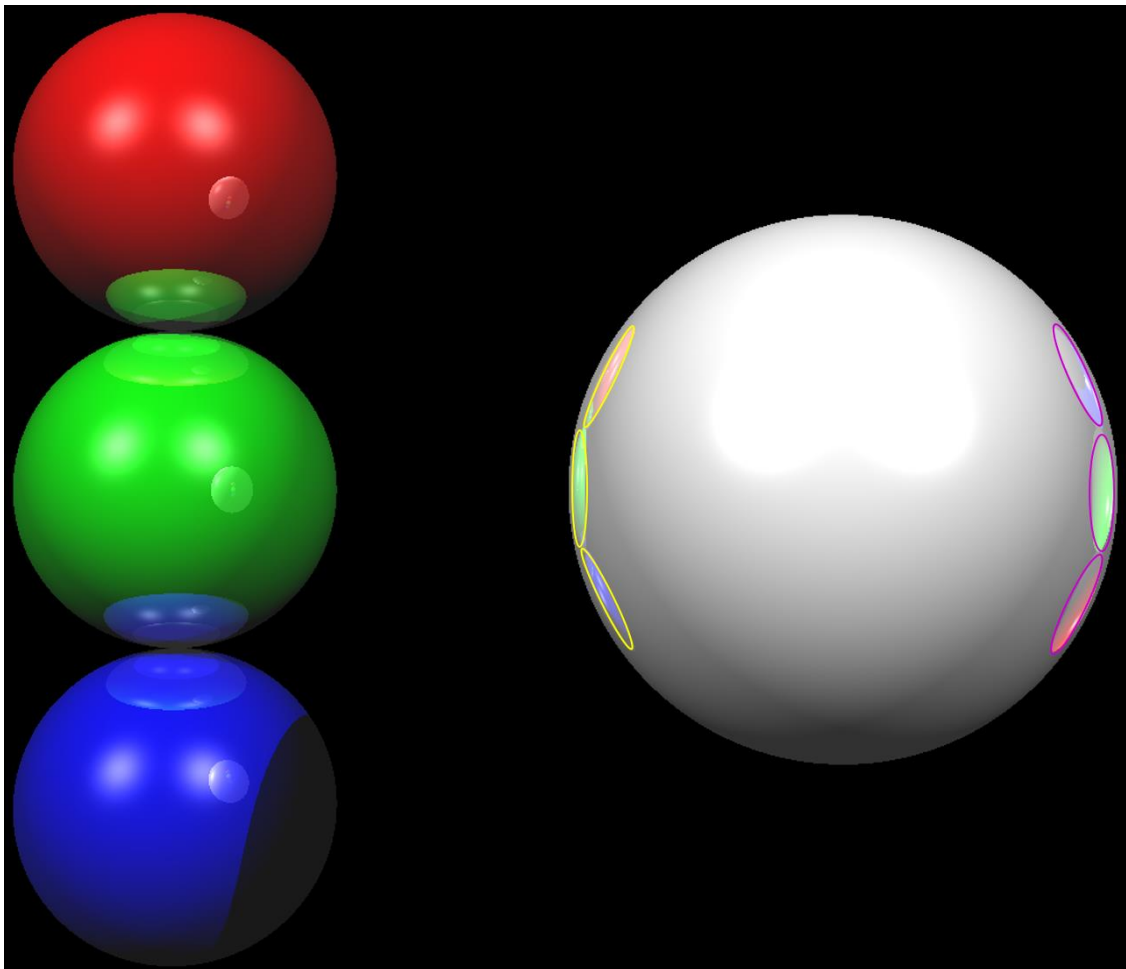
그렇게 relative refractive index를 구하고 두 vector 사이의 theta값을 구하는 computeRotation()를 사용하여 theta\_i (refraction되기 전의 theta)을 얻어낸다.

그 후 theta\_i와 Snell's law를 사용하여 theta\_r값을 구한다.

마지막으로 위에서 작성한  $l_t = \left(\frac{n_i}{n_r} \cos \theta_i - \cos \theta_r\right) n - \frac{n_i}{n_r} l$  공식에 기반하여 refraction ray를 얻어낸다.

최종적으로 refraction ray에 의한 빛의 intensity의 계산은 material의 transmission (m\_transmission) 값에 따라 다르게 계산되어야 함으로 refraction ray를 통해 얻은 빛의 intensity에 m\_transmission을 곱해 final intensity에 더해준다.

#### 4. Ray tracing of reflective and translucent spheres (No reflections inside spheres):



위 사진은 1개의 흰색 구는 reflective 및 translucent하고 나머지 3개의 빨강, 초록, 파랑 구는 reflective한 모습이다. 여기서 구 내부에서 reflection은 일어나지 않고 ray tracing depth는 3이다.

그리고 위 사진의 경우 1, 2, 3과는 다르게 reflection에 의한 상과 refraction에 의한 상이 모두 나타나 있는데, 이는 RGB의 구와 흰색 구가 마주보는 곳에 reflection에 의한 상이 노란색 동그라미로 표시되

어 있고 RGB의 구를 기준으로 흰색 구의 반대편에 refraction에 의한 상이 보라색으로 동그라미로 표시되어 있다.

노란색의 reflection 상의 경우 RGB의 순서대로 상이 맺힌 것을 볼 수 있고 refraction 상의 경우 BGR의 순서대로 상이 역전되어 맺힌 것을 볼 수 있다.

Refraction에 의한 상이 순서가 역전된 이유는 refractive index의 값이 흰색 구의 내부와 외부에서 다르고 translucent한 물체가 구이기 때문이다.