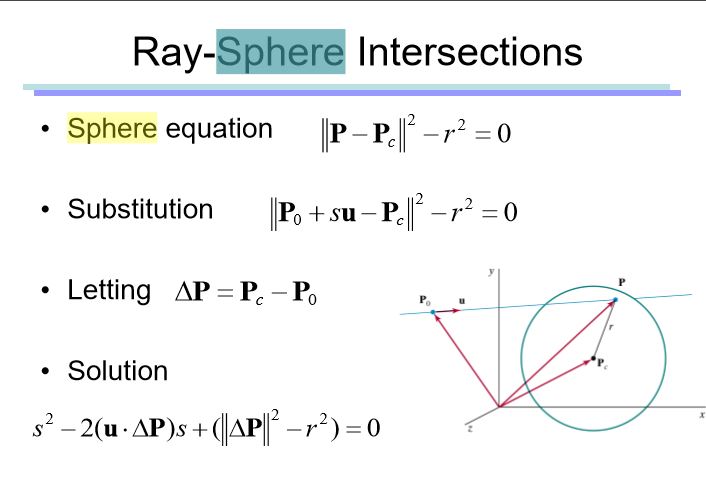
Computer Graphics HW5 Report

2015-11985 남중혁

<구현 목록>

1. Ray tracing sphere

* Sphere-Ray intersection equation을 이용하여 intersection을 구했습니다.
* 교재에서의 Sphere-Ray intersection 공식을 사용하였습니다.



1. Ray tracing polygons

* Polygon Class (child class of Object class) 는 이름은 Polygon이지만 삼각형을 그리도록 하였습니다.
* 모든 Polygon은 삼각형으로 쪼개서 그렸습니다.
* 제 Fish model(Swept Surface)는 삼각형으로 그려서 BSPTree로 정렬하였습니다.

1. Importing geometry files such as OBJ [5pt] X
2. Export image file [5pt]

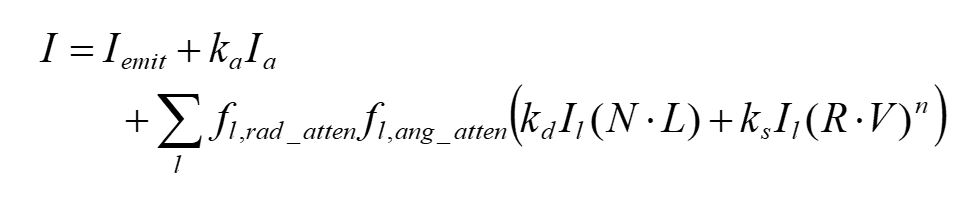
* 결과를 .bmp파일로 출력합니다.

1. Texture mapped spheres and polygons [10pt]

* Sphere : 구면 좌표계를 이용하였습니다. Ray를 쏘았을 때, Hit point를 얻어서, Sphere의 중심으로부터 Theta와 Phi를 구하여, 해당 값을 u, v ( 0 < u < 1, 0 < v <1 )에 매칭시켰습니다. 가로로 조금 늘어지는 현상을 보았는데, 더 고급진 interpolation 방법을 사용한다면.. 더 좋은 결과를 얻을 수 있지 않을까.. 싶습니다.
* Polygons : 다양한, 불균형한 모양의 Polygon들에 대해서는 어떻게 매핑을 할까 고민해봤는데,,,, 별로 좋은 결과는 얻지 못했고, 벽면과 바닥면, 사각형들에 대해서는 Texture Mapping을 깔끔하게 진행했고, 90도가 아닌 사각형에 대해서도 Mapping 되게 하였습니다.(모양은 조금 찌그러지겠지만.. )

1. Phong illumination

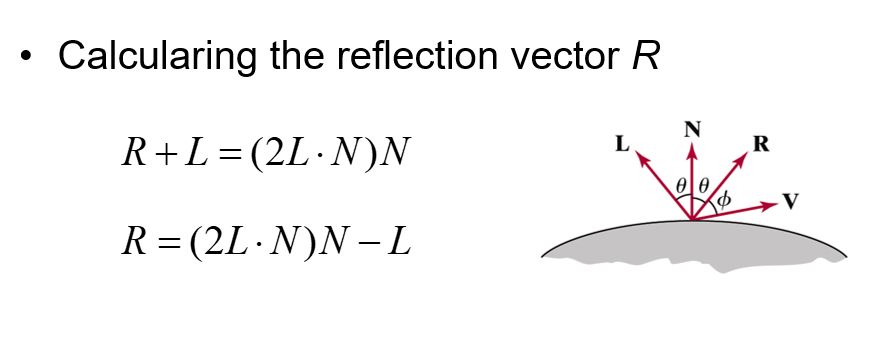
* Object는 ambient color, diffuse color, specular color, shininess 값을 가지고 있으며, 그 값에 따라 Phong illumination합니다.



수업 자료에 나온 이 식으로 Phong illumination을 진행했습니다.

1. Recursive Reflection

* 최대 Depth를 설정하였고, 그 한계까지 Trace()함수를 recursive하게 부를 수 있도록 구현하였습니다.



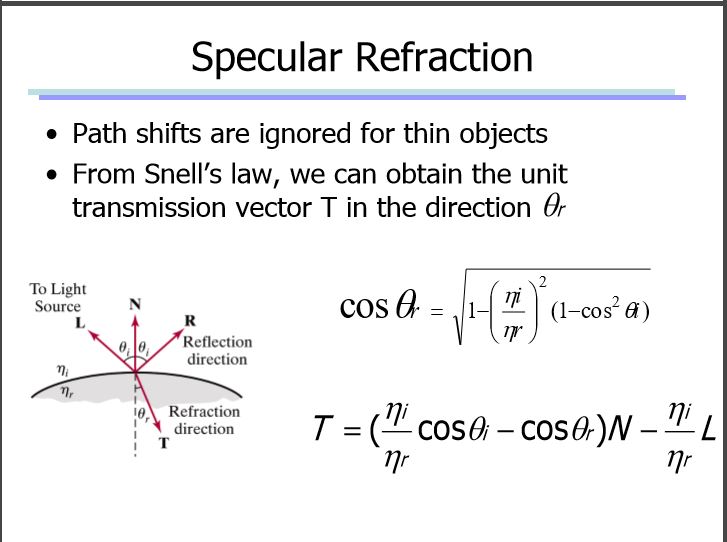
Reflection 또한 교재의 식을 사용하였습니다.

1. 추가구현
   1. Binary Space Partitioning

: 저번 시간에 그린 Fish Model ( Swept Surface Model)을 BSP-Tree를 이용하여 그렸습니다. Object가 가장 먼저 닿는 점을 얻어내고자 원래 방식은 min value를 찾아내며 매 trace마다 모든 삼각형을 접근하지만, BSP-Tree를 이용한다면, Tree를 중위 순회 하였을 때, (partitioning하여, eye를 기준으로 평면 뒤쪽은 제거, 남은 부분에서 partitioning하고… + 반복) O(n)의 시간 안에 가장 가까운 삼각형을 찾아 낼 수 있기 때문에 매우 효과적입니다.

* 1. Recursive Reflection

:



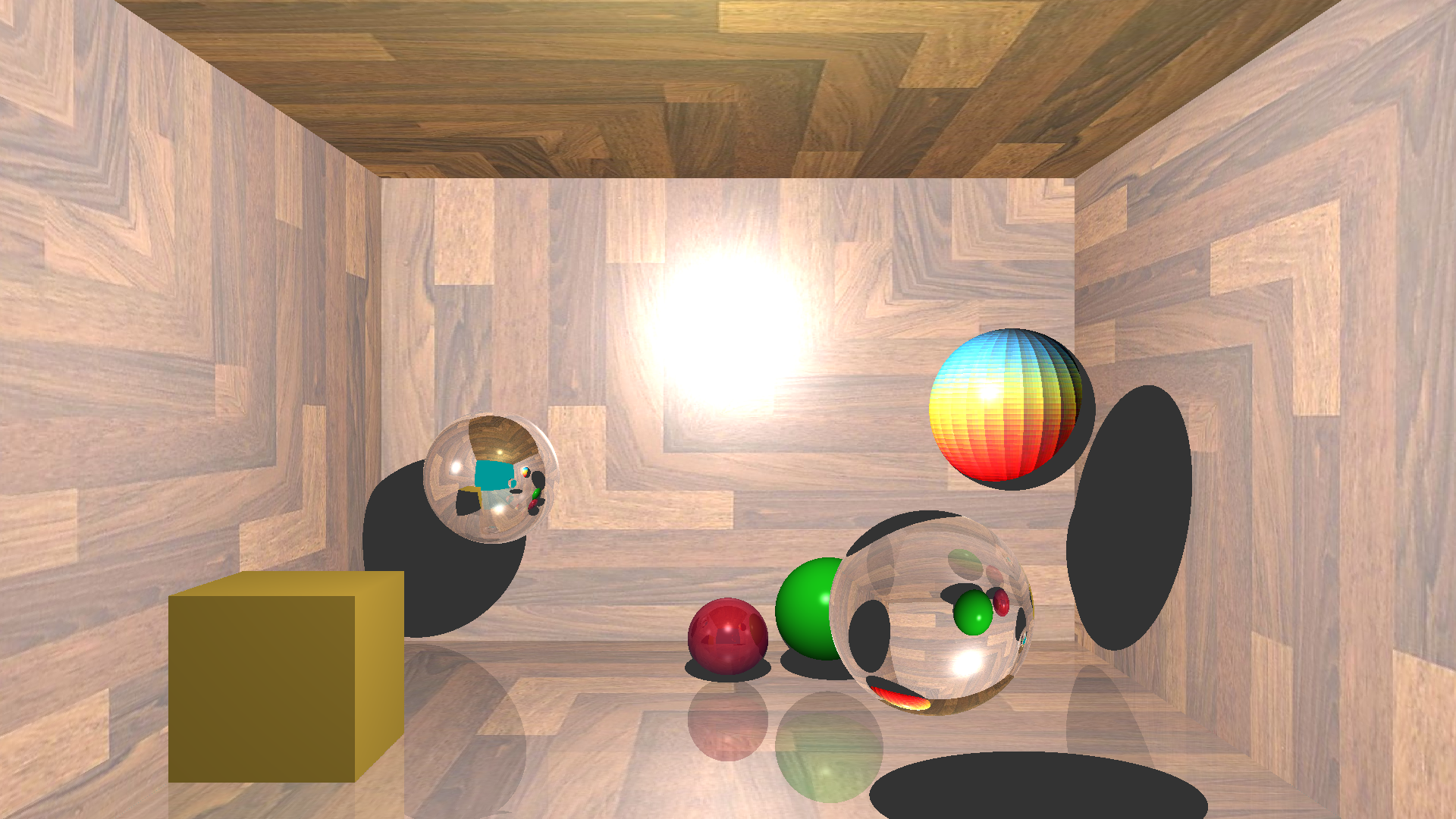
교재에 나온 공식을 사용하여, recursive refraction을 구현하였습니다. 따라서, 각 reflection ray과 refraction ray는 Binary Ray-Tracing Tree를 그리며 각 ray가 얻는 색깔 값을 contribution하여 scene을 그려냅니다.

1. 개발 환경
   1. 언어 : C++

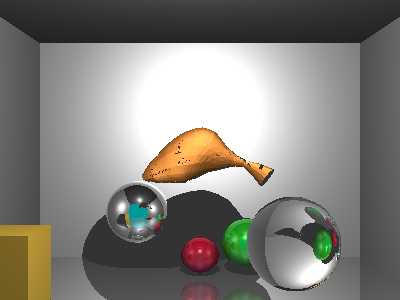
gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.4)

* 1. OS : Linux Ubuntu 16.04.2 LTS
  2. 컴파일 방법 : hw5 폴더 안에서 $make
  3. 실행 : $./main

1. 사진



* Ray tracing spheres
* Ray tracing polygons ( wall )
* Texture mapped spheres and polygons (오른쪽 공은 texture mapping 한 것입니다. 원본 그림에 명암이있어서… 아니게보일지도)
* Phong illumination
* Recursive reflection ( reflection ball, 바닥, 빨간공에도 약간의 반사도 ).
* Recursive refraction ( glass ball )



* BSP-Tree

(물고기 polygon 개수가 꽤 많아서,, rendering시간이 걸리는 바람에 저화질인데 고화질로 rendering해서 한번 더 보내보겠습니다..)