

컴퓨터그래픽스 (CG)

HW02_Lighting_a_Torus



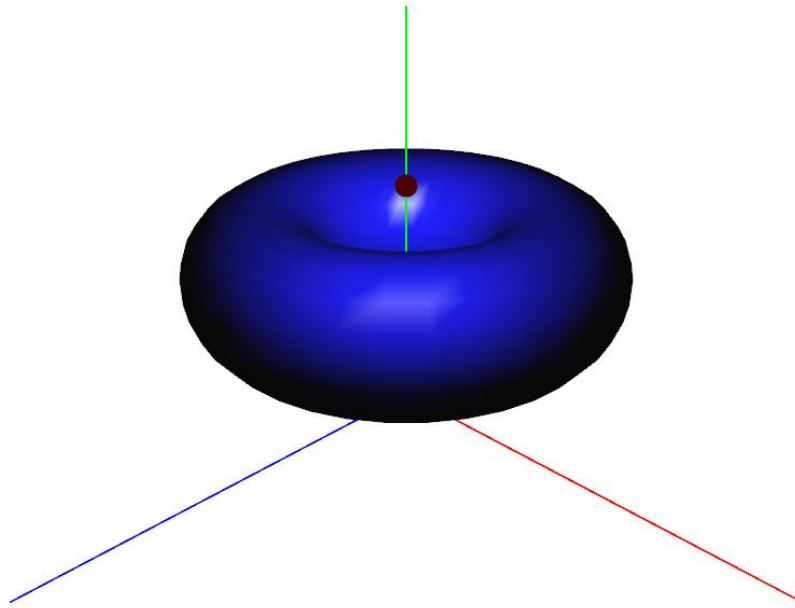
Self-scoring table

	1	2	3	4	5	Total
Score	1	1	1	1	1	5

2018707068 김경환

KwangWoon University

1. Draw a smooth-shaded torus with the vertex normal vectors,

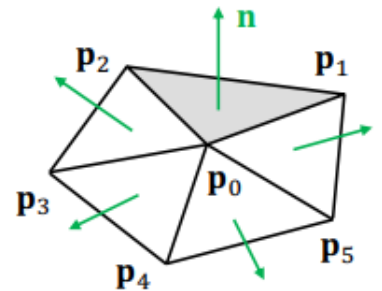


우선 torus를 그리는 방법에 대해서는 이전 HW1에서 설명했으므로 넘어가고 shading에 관련된 이야기를 하겠다.

Vertex normal vector를 사용하여 smooth shading을 하기 위해서는 먼저 face normal vector가 필요하다.

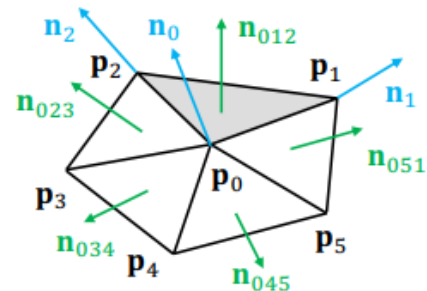
예를 들어 오른쪽 사진의 삼각형의 경우에는 $n = \frac{(p_1 - p_0) \times (p_2 - p_0)}{\|(p_1 - p_0) \times (p_2 - p_0)\|}$ 로 계산된다.

Torus의 경우에는 사각형이므로 점 4개 중 하나를 기준으로 나머지 셋 중 두 개로 뺀어나가는 vector의 cross product를 구하고 normalize하면 된다.



이제 해당 vertex를 공유하는 face normal vector의 평균으로 vertex normal vector를 계산한다. 식은 다음과 같다. ($n_k = \frac{\sum_{ij} n_{kij}}{\|\sum_{ij} n_{kij}\|}$)

그리하여 실제 code에서는 평균을 계산하기 위해 face normal vector를 계산하고 이를 해당 face를 이루는 4개의 vertex normal vector에 더해준다. 그렇게 되면 최종적으로 vertex normal vector에는 한 vertex를 공유하는 사각형 4개의 face normal vector가 더해지게 되고 이제 이를 normalize하면 vertex normal vector가 계산된다.



이제 이렇게 계산한 vertex normal vector를 이용해 vertex의 좌표를 지정하기전에 normal vector를 지정해줌으로써 smooth shading할 수 있다. (ex. `glNormal3f(n0); glVertex3f(p0);`)

2. Make a point light and draw its geometry with a sphere:

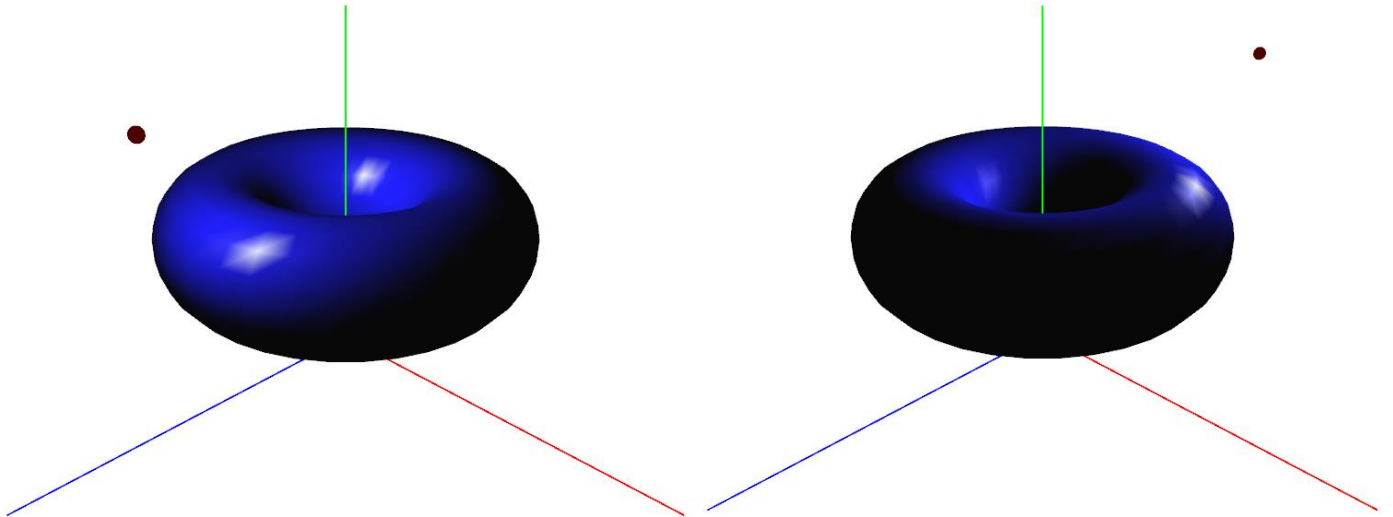
Point light를 생성하고 이를 구의 형상으로 그리기 위해서는 우선 light의 초기 시작 위치가 필요하다. 그리고 해당 light의 경우에는 회전도 가능해야함으로 rotation matrix가 곱해질 수 있는 homogenous coordinate로 선언해주어야 한다.

빛의 효과를 주기 위해 `glLightfv()`를 이용하여 빛의 3가지 속성인 ambient, diffuse, specular를 lightP의

위치에 설정해준다. 또한, 기본적으로 attenuation 효과를 상수 부분만 지정하였다.

마지막으로 다시 lightP를 이용하여 해당 위치로 translate하고 구가 잘 보이도록 빨간색의 색상과 material의 specular, shininess 속성을 부여하여 gluSphere()로 구의 형상을 그린다.

3. Rotate the light around the torus:



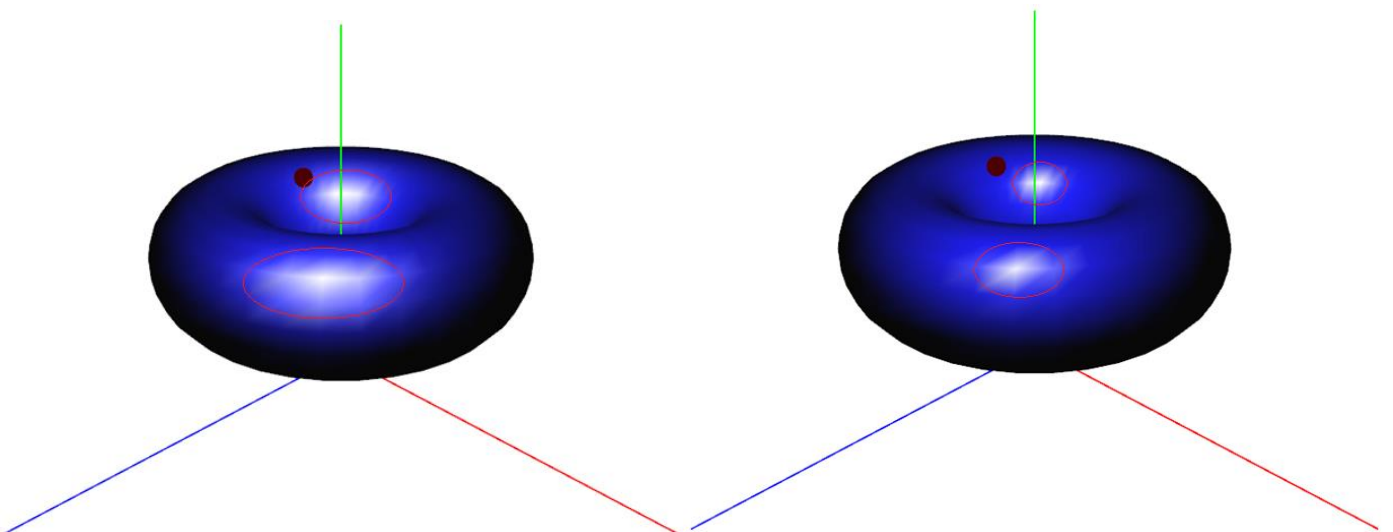
위 사진은 torus 주변을 light가 회전하는 모습이다.

이는 torus가 y축 (0, 0, 1)을 중심으로 만들어져 있으므로 light도 y축을 중심으로 회전하도록 하였다.

그리고 각도의 경우 glfwGetTime()을 이용하여 경과시간(delta)을 계산하고 이러한 경과시간이 1.0f/120 보다 커질 경우 즉 120fps 환경에서 1 frame이 지날 때마다 1°씩 증가하도록 하였다.

마지막으로 해당 rotation을 구현하기 위해서는 위에서 진행한 빛 효과를 부여하고 구의 형상을 그리기 이전에 빛의 위치를 변화시켜야 한다.

4. Fix the position of the light and change the shininess coefficient within the proper range using the arrow keys:



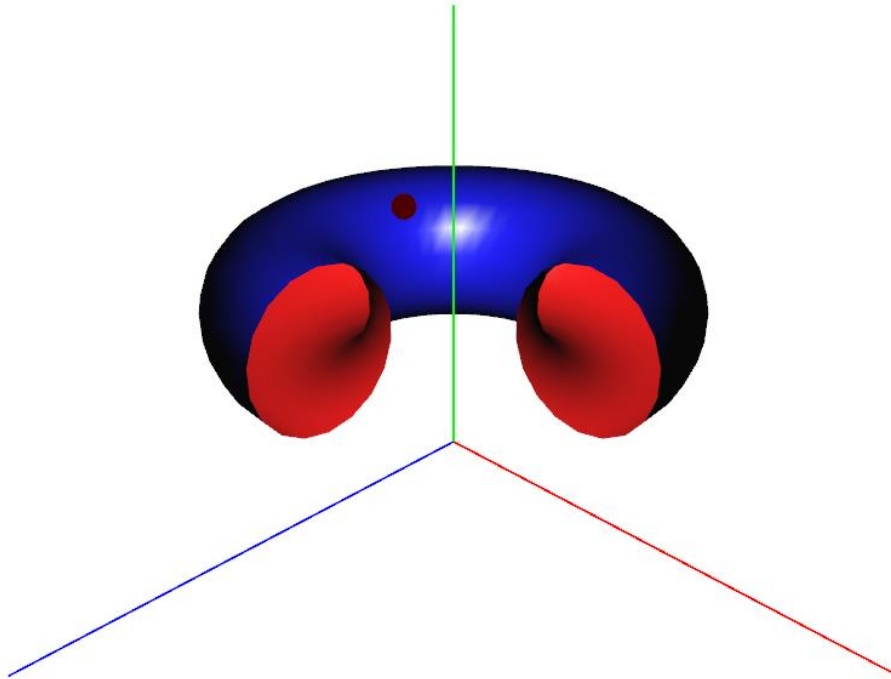
왼쪽은 shininess coefficient = 15를 준 모습이고, 오른쪽은 shininess coefficient = 50을 준 모습이다.

OpenGL에서 Shininess coefficient는 0에서 128사이의 값을 가질 수 있으므로 15의 값은 상대적으로 낮은 값이라고 할 수 있다. 그렇기 때문에 왼쪽처럼 오른쪽보다 shininess coefficient를 상대적으로 낮게

주면 재질이 덜 반짝이는 재질로써 빨간색으로 표시한 하이라이트가 상대적으로 넓게 퍼진 모습을 보인다.

위에서 설명한 것처럼 OpenGL에서 Shininess coefficient는 0에서 128사이의 값을 가질 수 있으므로 50의 값은 범위에 비하면 상대적으로 낮은 편에 속한다. 하지만, 왼쪽보다는 반짝이는 재질로써 하이라이트가 좁게 모여 반사광의 집중도가 많이 높아진 것을 볼 수 있다

5. Control the sweep angle to demonstrate the two-sided lighting:



위 사진은 Torus의 안쪽면을 빨강, 바깥면을 파랑으로 서로 다른 색상을 갖도록 lighting한 모습이다. 해당 내용을 구현하기 위해서는 우선 양쪽면을 그리기 위해서 `glDisable(GL_CULL_FACE)`을 사용하여 back face culling을 disable해야한다.

그리고 `glLightModeli(GL_LIGHT_MODEL_TWO_SIDE, GL_TRUE)`을 사용하여 two-sided lighting을 enable하도록 한다.

그리고나서 마지막으로 `glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_blue)`을 사용하여 앞면의 diffuse 색상을 blue로 설정하고, `glMaterialfv(GL_BACK, GL_DIFFUSE, mat_red)`을 사용하여 뒷면의 diffuse 색상을 red로 설정한다.

Sweep angle을 control하는 방법으로는 torus를 렌더링할 때 36 by 18의 2차원 배열에서 36에 해당하는 값을 `sweep_angle`이라는 변수로 두어 초기의 36값에서 방향키 입력에 따라 1씩 증감하도록 하였다. (증감하는 1의 값은 10°에 해당)

또한, 여기서 오류가 발생하지 않도록 0보다 작거나 36보다 커지지 않도록 조건문을 추가하였다.