

光线跟踪 3

● 简述

本次作业的主要内容是在前两次作业的基础之上实现 Phong 材质模型，阴影渲染，以及使得光线能够在材质上进行反射和折射。

● Phong 材质模型与光线反射

Phong 材质模型在实现的时候，我们使用的是以下方程（程序中的 shade 函数）：

$$diffuse = \overrightarrow{dirToLight} \cdot \overrightarrow{normal}$$

$$specular = -\overrightarrow{direction} \cdot \overrightarrow{reflection_{out}}$$

$$\overrightarrow{pixel} = (\overrightarrow{color} * diffuse + \overrightarrow{specular_{color}} * specular^{exponent}) * \overrightarrow{light}$$

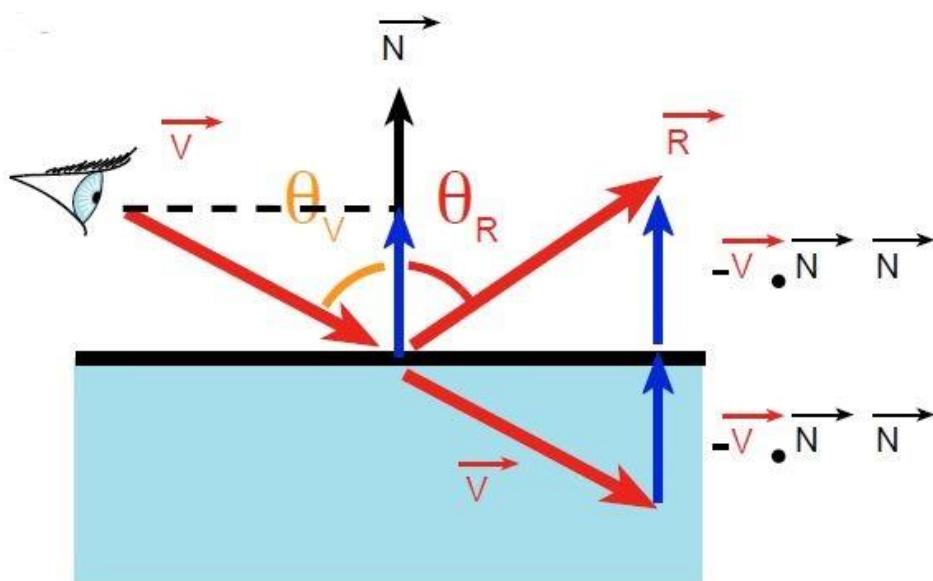
其中 $\overrightarrow{dirToLight}$ 从反射点到环境光源点的方向向量， $\overrightarrow{direction}$ 是当前追踪的主光线到反射点的方向向量， $exponent$ 是一个常量，由试验数据给出， \overrightarrow{light} 则是环境光源点， \overrightarrow{color} 是物体本身的颜色，上式中需要注意的是最后两个向量使用的乘法是对应分量相乘，其结果还是一个向量，这样得到的像素值 \overrightarrow{pixel} 只要在加上一个周围环境光 (ambient) 即可：

$$\overrightarrow{pixel} = \overrightarrow{pixel} + \overrightarrow{color} * \overrightarrow{ambient}$$

上式中童谣需要注意两个向量使用的乘法是对应分量相乘。这样我们就能够得到一个比较真实的材质模型（相比之前，虽然依然很假）。

由于程序框架中已经给我们计算过光线的衰减了，所以在这里我们使用的 \overrightarrow{light} 是计算过衰减之后的光线了。

注意到我们还有 $\overrightarrow{reflection_{out}}$ 没有计算，假设情景如下图所示：



如图我们可以很轻松的求出反射光线的方向：

$$\overrightarrow{reflection_{out}} = \overrightarrow{reflection_{in}} - 2 \left(\overrightarrow{reflection_{in}} \cdot \overrightarrow{normal} \right) * \overrightarrow{normal}$$

在按照上面的方法计算反射光线的时候，需要特别注意向量的方向。

● 阴影

在实现程序的时候，我们是先计算周围环境光对物体的影响，然后再分别计算其它的环境光源点对物体的影响，要实现阴影效果只需要再计算环境光源点的时候判断当前环境光源是否能够照射到当前物体，如果能则再利用该光源进行 shade 渲染，这样光线如果被挡住，当前渲染点就不会收到光源的影响。

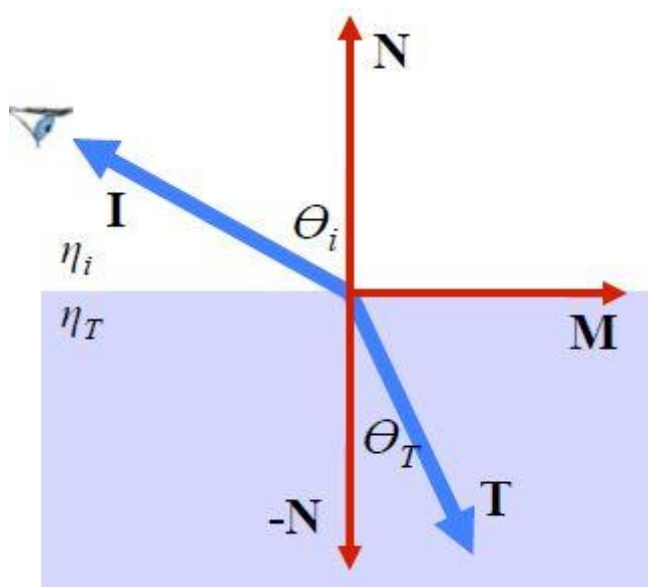
在程序实现的时候，我们采取的方法是从交点发出一条到光源发出点的光线，然后利用该光线对其它物体求交，如果有交点则代表光线被挡住了，这时不进行 shade 渲染。同时为了排除光线与射出点所在的物体相交（这个时候参数 t 为 0），我们在这个时候求交的时候设置 t_{min} 为一个比 0 稍大一点的数值：

$$\varepsilon = 0.0001$$

否则图片上会产生黑点（因为 float 精度问题，所以黑点是离散状）。

● 光线折射折射

首先假设光线折射的情形如下图所示：



已知了两个介质的折射率，入射光线的方向以及法向量，和入射角。需要注意的是这里的向量都是单位向量，并且为了运算的方便，这里我们取入射光线的反方向。如图，我们可以知道：

$$\vec{I} = \vec{N} \cos \theta_i - \vec{M} \sin \theta_i$$

变形，我们可以得到 \vec{M} ：

$$\vec{M} = \frac{\vec{N} \cos \theta_i - \vec{I}}{\sin \theta_i}$$

由于待求的 \vec{T} 也是单位向量，我们可以把它写成：

$$\vec{T} = -\vec{N}\cos\theta_T + \vec{M}\sin\theta_T$$

由光的折射定律：

$$\frac{\sin\theta_i}{\sin\theta_T} = \frac{\eta_i}{\eta_T} = \eta_r$$

将其带入上式中，消去未知项，可以得到：

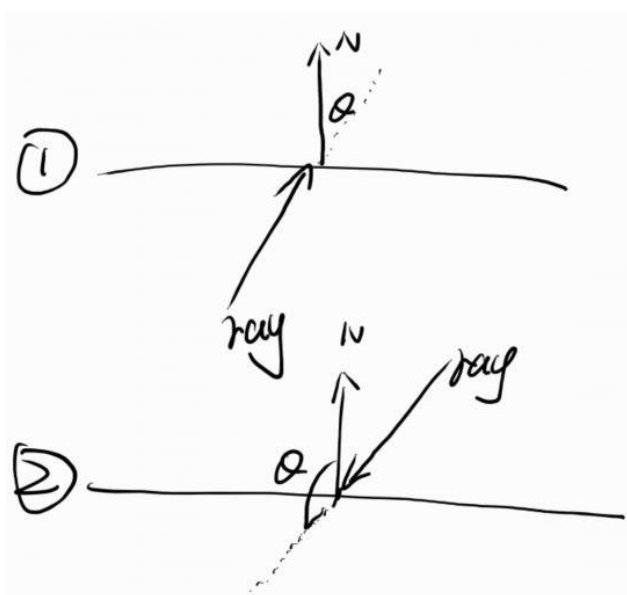
$$\vec{T} = \left[\eta_r(\vec{N} \cdot \vec{I}) - \sqrt{1 - \eta_r^2 (1 - (\vec{N} \cdot \vec{I})^2)} \right] \vec{N} - \eta_r \vec{I}$$

● 图像渲染

渲染过程的伪代码如下图所示：

```
trace ray
↑
Intersect all objects
color = ambient term
For every light
    cast shadow ray
    color += local shading term
If mirror
    color += colorrefl *
    └── trace reflected ray
If transparent
    color += colortrans *
    └── trace transmitted ray
```

在渲染图像的时候，需要注意的一点是，在折射的时候我们需要判断光线是从外部射向物体内部，还是从物体的内部射到物体的外部，通过光线和法向量的角度我们可以判断属于哪种情况，如下图所示：



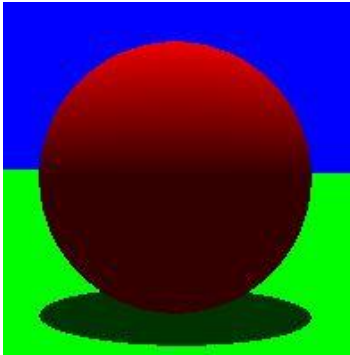
可以发现，当夹角小于 90° 的时候是从物体内部射出到物体外部，如果夹角大于 90° 则

代表是从物体外部射入物体内部。当光线从物体内部射出到物体外部时需要按如下方法处理法向量和折射率：

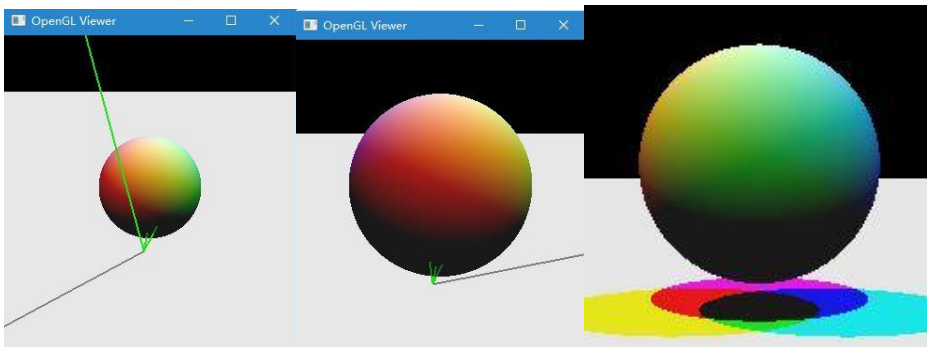
$$\overrightarrow{normal} = -\overrightarrow{normal}$$
$$\eta_r = \frac{1}{\eta_r}$$

● 实验结果

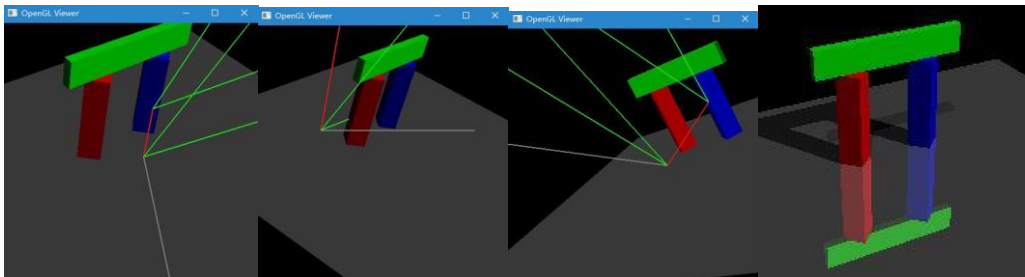
```
-input scene4_01_sphere_shadow.txt -size 200 200 -output output4_01.tga -shadows
```



```
-input scene4_02_colored_shadows.txt -size 200 200 -output output4_02.tga -shadows
```



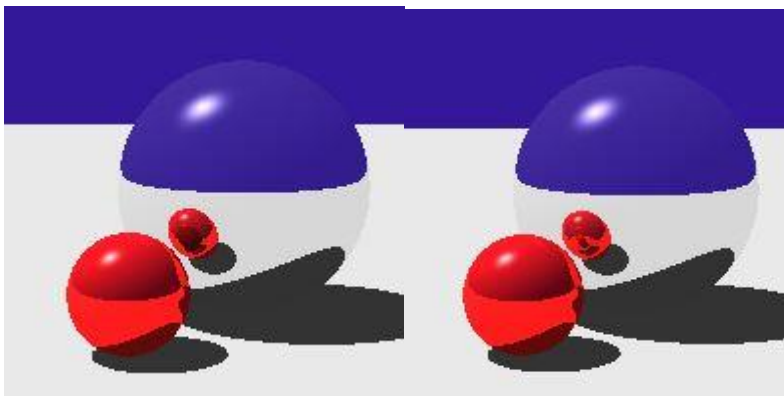
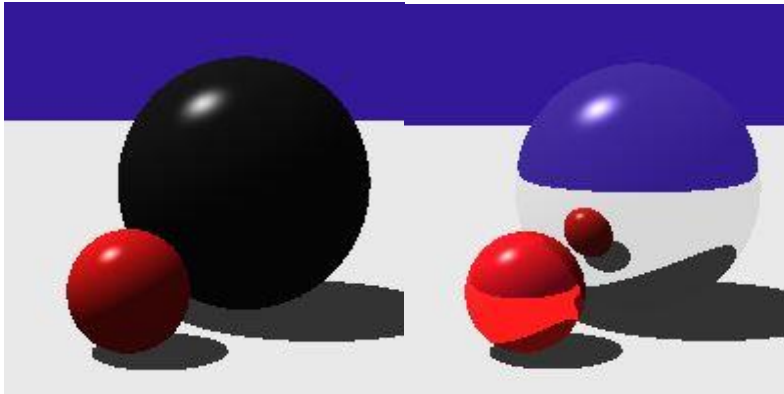
```
-input scene4_03_mirrored_floor.txt -size 200 200 -output output4_03.tga -shadows -bounces 1 -weight 0.01
```



```

-input scene4_04_reflective_sphere.txt -size 200 200 -output output4_04a.tga
-shadows -bounces 0 -weight 0.01
-input scene4_04_reflective_sphere.txt -size 200 200 -output output4_04b.tga
-shadows -bounces 1 -weight 0.01
-input scene4_04_reflective_sphere.txt -size 200 200 -output output4_04c.tga
-shadows -bounces 2 -weight 0.01
-input scene4_04_reflective_sphere.txt -size 200 200 -output output4_04d.tga
-shadows -bounces 3 -weight 0.01

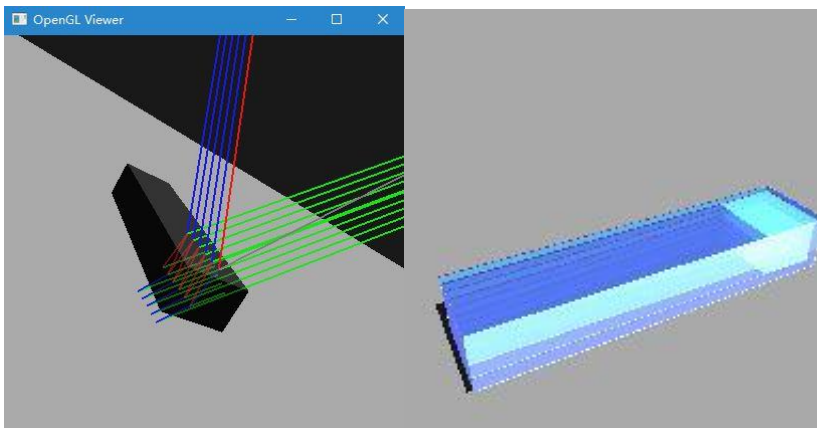
```



```

-input scene4_05_transparent_bar.txt -size 200 200 -output output4_05.tga -
shadows -bounces 10 -weight 0.01 -shade_back

```



```

-input scene4_06_transparentBars.txt -size 200 200 -output output4_06a.tga -
shadows -bounces 0 -weight 0.01 -shade_back
-input scene4_06_transparentBars.txt -size 200 200 -output output4_06b.tga -
shadows -bounces 1 -weight 0.01 -shade_back
-input scene4_06_transparentBars.txt -size 200 200 -output output4_06c.tga -
shadows -bounces 2 -weight 0.01 -shade_back
-input scene4_06_transparentBars.txt -size 200 200 -output output4_06d.tga -
shadows -bounces 3 -weight 0.01 -shade_back
-input scene4_06_transparentBars.txt -size 200 200 -output output4_06e.tga -
shadows -bounces 4 -weight 0.01 -shade_back
-input scene4_06_transparentBars.txt -size 200 200 -output output4_06f.tga -
shadows -bounces 5 -weight 0.01 -shade_back

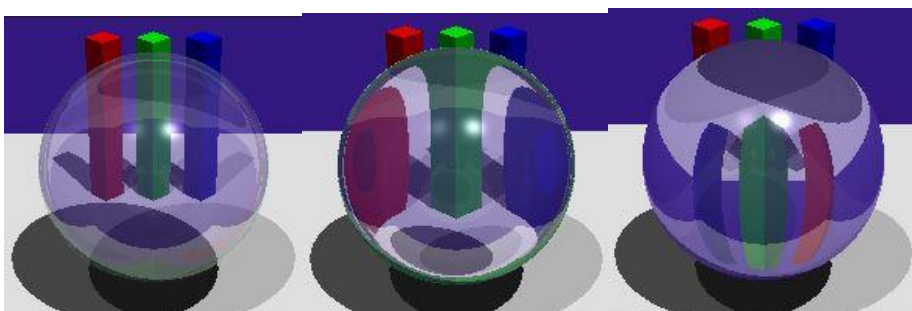
```



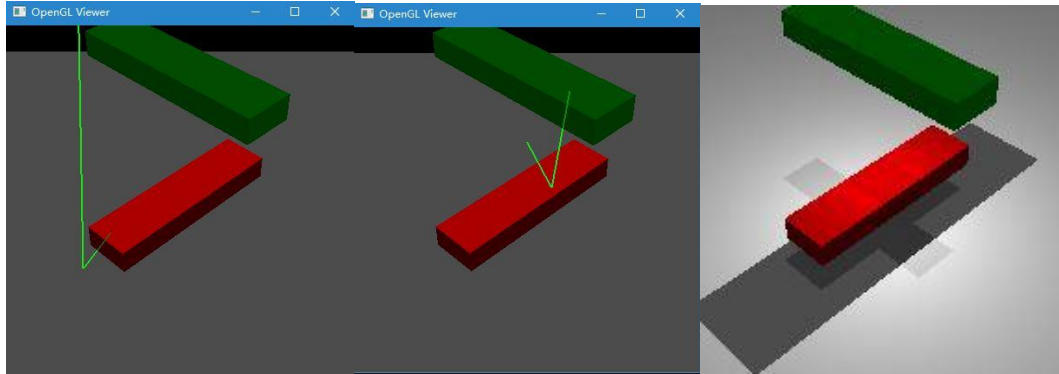
```

-input scene4_07_transparent_sphere_1.0.txt -size 200 200 -output
output4_07.tga -shadows -bounces 5 -weight 0.01 -shade_back
-input scene4_08_transparent_sphere_1.1.txt -size 200 200 -output
output4_08.tga -shadows -bounces 5 -weight 0.01 -shade_back
-input scene4_09_transparent_sphere_2.0.txt -size 200 200 -output
output4_09.tga -shadows -bounces 5 -weight 0.01 -shade_back

```



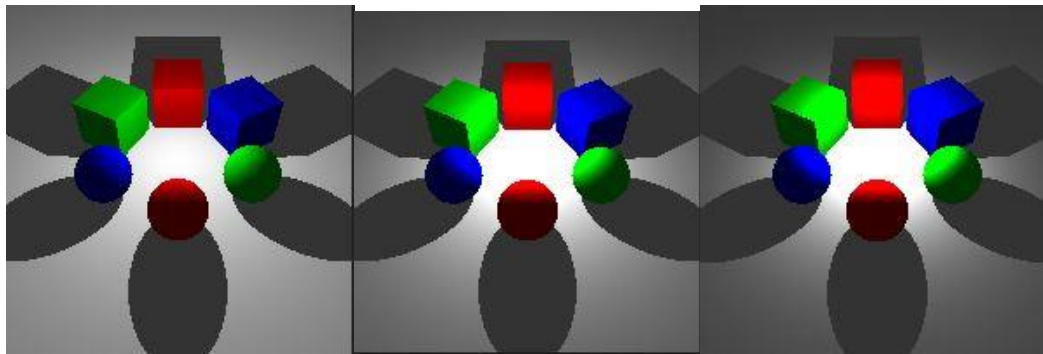

```
-input scene4_10_point_light_distance.txt -size 200 200 -output output4_10.tga  
-shadows
```



```
-input scene4_11_point_light_circle.txt -size 200 200 -output output4_11.tga  
-shadows
```

```
-input scene4_12_point_light_circle_d_attenuation.txt -size 200 200 -output  
output4_12.tga -shadows
```

```
-input scene4_13_point_light_circle_d2_attenuation.txt -size 200 200 -output  
output4_13.tga -shadows
```



```
-input scene4_14_faceted_gem.txt -size 200 200 -output output4_14a.tga -  
shadows -shade_back -bounces 0 -weight 0.01
```

```
-input scene4_14_faceted_gem.txt -size 200 200 -output output4_14b.tga -  
shadows -shade_back -bounces 1 -weight 0.01
```

```
-input scene4_14_faceted_gem.txt -size 200 200 -output output4_14c.tga -  
shadows -shade_back -bounces 2 -weight 0.01
```

```
-input scene4_14_faceted_gem.txt -size 200 200 -output output4_14d.tga -  
shadows -shade_back -bounces 3 -weight 0.01
```

```
-input scene4_14_faceted_gem.txt -size 200 200 -output output4_14e.tga -  
shadows -shade_back -bounces 4 -weight 0.01
```

```
-input scene4_14_faceted_gem.txt -size 200 200 -output output4_14f.tga -  
shadows -shade_back -bounces 5 -weight 0.01
```

