CS 465 Homework 2: Ray Tracing Bugs

out: Friday 1 September 2006
due: Friday 8 September 2006

What will be the effect on the image for each of the following errors in coding up a ray tracer?

1. You accidentally swap the viewpoint with the image plane point in the camera code.
2. You start your shadow rays exactly at the reflection point.
3. You forget to normalize the eye and light direction before computing Lambertian shading.
4. Rather than checking all the objects to find the closest intersection along a viewing ray, your intersection method always returns the first one that is found.
5. You accidentally use the view direction for the image plane normal, leaving the supplied image plane normal unused.
6. You neglect to check that shadow ray intersections are happening between the shading point and the light.
7. You forget to normalize the surface normals when you compute them for spheres.

Your answers can be fairly brief, and you should phrase them in terms of symptoms you would see when looking at the image. For instance:

> **Bug**: The outer loop of your ray tracer looks something like this:
>
> ```
> for $i$ = 1 to width
>     for $j$ = 1 to height
>         image[$j$, $i$] = computePixelValue($i$, $j$)
> ```
>
> **Symptom**: The image will be flipped sideways (that is, transposed).
>
> **Bug**: Your sphere intersection correctly finds the two roots, but returns the larger one rather than the smaller one.
>
> **Symptom**: You will see the inside of the back of the sphere, but shaded strangely (the exact shading depends on how your shading code handles normals that face away from the camera).

For simplicity, assume that the scene is closed—that is, you can't see out to the background from inside the scene.