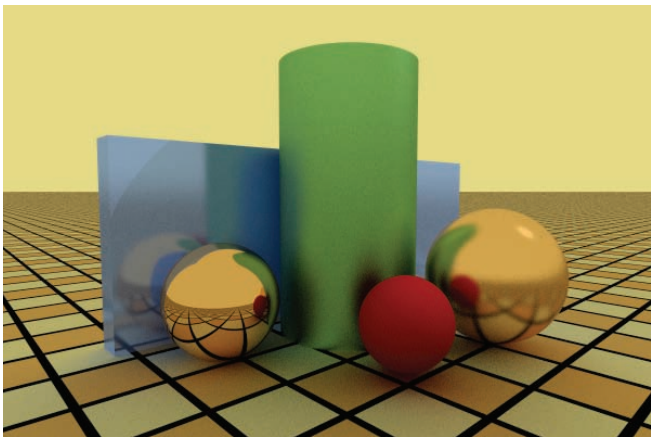
25 Glossy Reflection



Objectives

By the end of this chapter, you should:

- understand what glossy reflection is;
- understand how it can be modeled in ray tracing;
- have implemented glossy reflection in your ray tracer;
- have more tools to produce nice images.

In this chapter, you will learn how to render glossy reflections. Many real materials are imperfect reflectors, where the reflections are blurred to various degrees due to surface roughness that scatters the incident radiance. The use of glossy reflections will allow us to remove the main inconsistency discussed in Section 24.4 by treating specular highlights and reflections in a consistent manner. Figure 25.1 is a photograph of three glossy spheres.

25.1 Modeling

Consider a ray that hits a surface from the direction $\boldsymbol{\omega}_o$ and its associated direction of mirror reflection $\boldsymbol{r}$, as illustrated in Figure 25.2. We can simulate glossy reflection by choosing a random direction for the reflected ray instead of using

Figure 25.1. Glossy spheres. Photograph by Kevin Suffern.

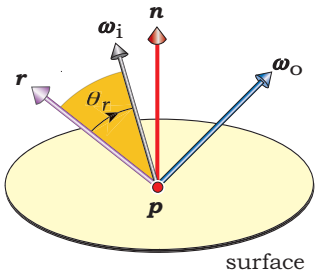


Figure 25.2. The reflected ray direction $\boldsymbol{\omega}_\text{i}$ makes an angle θ_r with the direction of mirror reflection $\boldsymbol{r}$.

the direction of mirror reflection, but how do we choose the direction? To make the simulation consistent with the specular-highlights model in Chapter 15, we should use the same cosine power formula. The density d in solid angle of the reflected rays around the mirror reflection direction $\boldsymbol{r}$ is then given by

$$d = (\cos\theta_r)^e, \tag{25.1}$$

where θ_r is the angle between $\boldsymbol{r}$ and $\boldsymbol{\omega}_\text{i}$ and e is the specular exponent.

A suitable BRDF for this purpose is

$$f_{r,s}(\boldsymbol{p}, \boldsymbol{\omega}_\text{i}, \boldsymbol{\omega}_\text{o}) = c\,k_\text{r}\boldsymbol{c}_\text{r}\cos(\theta_r)^e = c\,k_\text{r}\boldsymbol{c}_\text{r}(\boldsymbol{r} \bullet \boldsymbol{\omega}_\text{i})^e, \tag{25.2}$$

where c is a normalization constant. This constant is proportional to $e + 1$, so that $c \to \infty$ as $e \to \infty$. It expresses the fact that as e increases, the BRDF (25.2) becomes more concentrated around the direction of mirror reflection. In the limit $e = \infty$, the BRDF becomes a delta function.

Substituting Equation (25.2) into (24.1) gives the following expression for the reflected radiance in the $\boldsymbol{\omega}_\text{o}$ direction:

$$L_\text{indirect}(\boldsymbol{p}, \boldsymbol{\omega}_\text{O}) = ck_\text{r}\boldsymbol{c}_\text{r}\int_{2\pi^+} (\boldsymbol{\omega}_\text{i} \bullet \boldsymbol{r})^e L_\text{o}(\boldsymbol{r}_\text{c}(\boldsymbol{p}, \boldsymbol{\omega}_\text{i}), -\boldsymbol{\omega}_\text{i})\cos\theta_\text{i} d\omega_\text{i}. \tag{25.3}$$

Because there are no delta functions in this integrand, we have to estimate its value using Monte Carlo integration. The estimator is

$$\langle L_\text{r}(\boldsymbol{p}, \boldsymbol{\omega}_\text{o})\rangle = \frac{ck_\text{r}\boldsymbol{c}_\text{r}}{n_\text{s}}\sum_{i=1}^{n_\text{s}} \frac{(\boldsymbol{\omega}_{\text{i},j} \bullet \boldsymbol{r})^e L_\text{o}(\boldsymbol{r}_\text{c}(\boldsymbol{p}, \boldsymbol{\omega}_{\text{i},j}), -\boldsymbol{\omega}_{\text{i},j})\cos\theta_{\text{i},j}}{p(\boldsymbol{\omega}_{\text{i},j})}. \tag{25.4}$$

In this expression, the measure of the pdf is solid angle in the hemisphere at $\boldsymbol{p}$. Because the choice of pdf determines the directions of the rays shot into the hemisphere, to use the distribution (25.1), the pdf should be proportional to the BRDF; that is,

$$p(\boldsymbol{\omega}_{\mathrm{i},j}) \propto (\boldsymbol{r} \bullet \boldsymbol{\omega}_{\mathrm{i}})^{e}. \tag{25.5}$$

A problem with this approach is that it ignores the $\cos\theta_{\mathrm{i}}$ term in Equation (25.3). If we want the glossy reflection to be consistent with perfect specular reflection in the limit $e \to \infty$, we need to get rid of this term. One way would be to make $p(\boldsymbol{\omega}_{\mathrm{i},j})$ proportional to the product of the BRDF and $\cos\theta_{\mathrm{i}} = \boldsymbol{n} \bullet \boldsymbol{\omega}_{\mathrm{i}}$:

$$p(\boldsymbol{\omega}_{\mathrm{i},j}) \propto (\boldsymbol{r} \bullet \boldsymbol{\omega}_{\mathrm{i}})^{e} (\boldsymbol{n} \bullet \boldsymbol{\omega}_{\mathrm{i}}),$$

but then the ray directions should also be derived from this product. Unfortunately, that's difficult to do. In spite of not being able to sample this correctly, I'll still use it with

$$p(\boldsymbol{\omega}_{\mathrm{i},j}) \propto c(\boldsymbol{r} \bullet \boldsymbol{\omega}_{\mathrm{i}})^{e} (\boldsymbol{n} \bullet \boldsymbol{\omega}_{\mathrm{i}}), \tag{25.6}$$

but I'll keep the distribution (25.1) for the ray directions. With this pdf, the BRDF normalization constant c cancels. The inconsistency in the pdf (25.6) and the ray direction distribution approaches zero as $e \to \infty$.

With this choice of pdf, the estimator (25.4) simplifies to

$$\left\langle L_{\mathrm{r}}(\boldsymbol{p}, \boldsymbol{\omega}_{\mathrm{o}}) \right\rangle = \frac{k_{\mathrm{r}} \boldsymbol{c}_{\mathrm{r}}}{n_{\mathrm{s}}} \sum_{j=1}^{n_{\mathrm{s}}} L_{\mathrm{o}}(\boldsymbol{r}_{\mathrm{c}}(\boldsymbol{p}, \boldsymbol{\omega}_{\mathrm{i},j}), -\boldsymbol{\omega}_{\mathrm{i},j}), \tag{25.7}$$

where each randomly reflected ray is treated in the same way as perfect mirror reflection was treated in Chapter 24. The only source of noise in Equation (25.7) is variations in the radiance returned by the reflected rays. This model has k_{s}, $\boldsymbol{c}_{\mathrm{s}}$, and e as parameters and, as you will see, produces nice blurred reflections that approach perfect mirrors as $e \to \infty$. Physically based models of glossy reflection are much more complex (see the Further Reading section).

Figure 25.3 shows the plane through the hit point that's perpendicular to $\boldsymbol{r}$. I'll call this the *normal plane*. It's relevant to the modeling because the hemisphere over which the ray directions are distributed is oriented around $\boldsymbol{r}$, instead of $\boldsymbol{n}$ as in previous chapters. The black dots sketched on the hemisphere

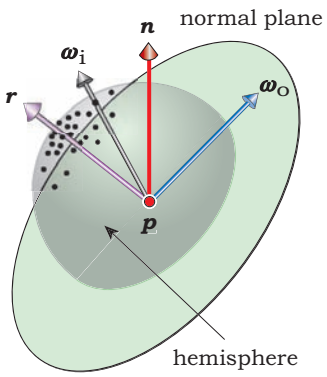


Figure 25.3. The normal plane at a hit point is perpendicular to the direction of mirror reflection.

show where reflected rays may penetrate the hemisphere. The ray directions will all be on the "$\boldsymbol{r}$" side of the normal plane.

In this context, Figure 15.4 shows how the exponent e controls the distribution of the samples over the hemisphere and concentrates them around $\boldsymbol{r}$ as e increases. We can therefore use e to control the amount of glossiness. Because materials become more glossy as the samples spread out over the hemisphere, the smaller e is, the more glossy the material will be. This continues down to $e = 1$, which is Lambertian reflection but oriented around $\boldsymbol{r}$, not $\boldsymbol{n}$.

We can express the reflected ray direction $\boldsymbol{\omega}_i$ in terms of an orthonormal basis ($\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$), where $\boldsymbol{w}$ is parallel to $\boldsymbol{r}$, and $\boldsymbol{u}$ and $\boldsymbol{v}$ are in the normal plane. The orientation of $\boldsymbol{u}$ and $\boldsymbol{v}$ around $\boldsymbol{r}$ doesn't matter because the distribution of samples on the hemisphere is rotationally symmetric about $\boldsymbol{r}$. Provided we have samples on a hemisphere with a cosine power distribution, the ray direction is given by the same expression as Equation (17.4):

$$\boldsymbol{\omega}_i = s_x\,\boldsymbol{u} + s_y\,\boldsymbol{v} + s_z\,\boldsymbol{w} \tag{25.8}$$

Finally, we have to deal with the fact that Equation (25.8) can generate rays that lie below the object's surface at the hit point. To see how this happens, consider an incoming ray that hits a surface with different incident angles, as in Figure 25.4. In Figure 25.4(a), the incoming ray direction $\boldsymbol{\omega}_o$ is normal to the surface and therefore coincident with $\boldsymbol{r}$ and $\boldsymbol{n}$. This is the only case where the hemisphere that's centered on $\boldsymbol{r}$ lies entirely above the surface. In

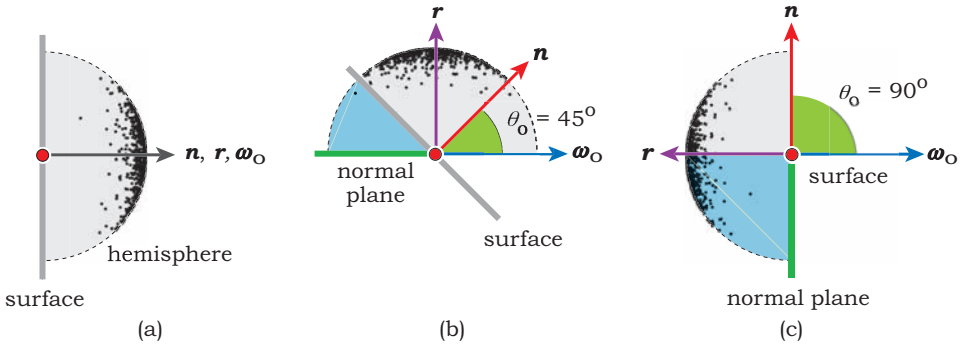


Figure 25.4. (a) At normal incidence, none of the hemisphere is below the surface; (b) part of the hemisphere is below the surface; (c) at grazing incidence, half of the hemisphere is below the surface.

Figure 25.4(b), the incident angle is 45°, and the cyan section of the hemisphere is below the surface[1]. In Figure 25.4(c), the incoming ray is tangent to the surface, and half the hemisphere is below the surface, a situation that occurs at object silhouettes as seen from the camera or ray origin. The *fraction* of reflected rays that lie below the surface depends on the angle of incidence and the value of e but is roughly 50% at grazing incidence for all values of e.

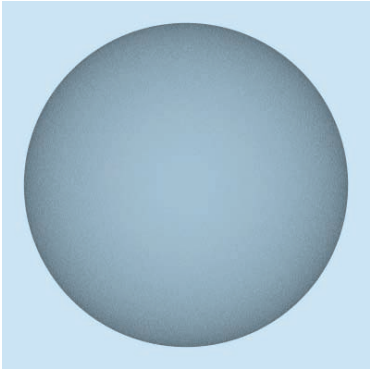
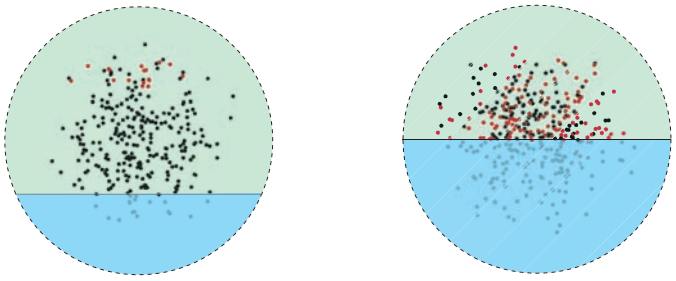


Figure 25.5. Incorrect rendering of a glossy sphere.

Figure 24.5(a) is a test scene that consists of a glossy sphere and a blue background color. This image was rendered with the estimator (25.7), $e = 1$, $k_r = 0.8$, $\boldsymbol{c}_r$ = white, 100 rays per pixel, and all reflected rays traced. The color of the sphere should be constant, but as you can see, it's not. The problem is that the rays under the surface hit the inside of the sphere, and since the sphere's material has no direct shading component, they return zero radiance. As a result, the sphere is only half as bright at the silhouette as it is in the center.

Fortunately, there's a simple solution. We just reflect any rays that are below the surface through $\boldsymbol{r}$, a process that puts them above the surface. Figure 25.6 illustrates how this works, where rays below the surface (the gray dots) are reflected to become the red dots. This introduces some bias in the distribution, where there are more rays than normal towards the periphery of the distribution, as in Figure 25.6(a). However, the bias decreases as the fraction of reflected rays increases, and it disappears at the silhouette, as in Figure 25.6(b).



(a) (b)

Figure 25.6. Rays below the surface (cyan segments of the hemisphere) are reflected through $\boldsymbol{r}$ (at the center of the disks) to the red dots, above the surface.

1. For $\theta_1 = 45°$, $\boldsymbol{\omega}_o$ is in the normal plane.

The test for the initial ray being below the tangent plane is $\boldsymbol{n} \bullet \boldsymbol{\omega}_i < 0$, and the expression for the reflected ray direction is

$$\boldsymbol{\omega}_i = -s_x \boldsymbol{u} - s_y \boldsymbol{v} + s_z \boldsymbol{w}.$$

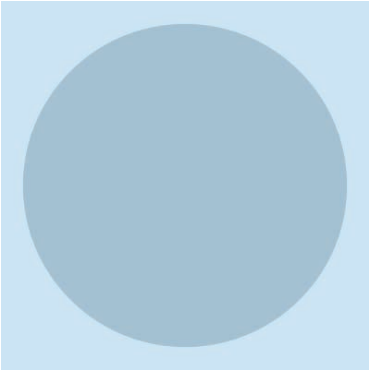


Figure 25.7. Correct rendering of a glossy sphere.

A bad feature of this technique is that as the fraction of the hemisphere below the surface increases, any stratification of the samples is progressively destroyed. Recall from Chapter 5 that one of the benefits of stratification is that samples generated with widely different random number pairs, for example, (0.001, 0.05) and (0.95, 0.99), are expected to be far apart on the unit square and therefore far apart on the hemisphere. By reflecting the rays through $\boldsymbol{r}$, these two samples can end up arbitrarily close to each other. We'll just have to live with this, because the alternative of just sampling the part of the hemisphere above the surface is difficult. Fortunately, the results are still good. Figure 25.7 shows the results of using this technique for the test sphere, which is now correctly rendered, although arguably not as interesting to look at.

25.2 Implementation

There's not much work required to implement glossy reflection, as we already have a glossy specular BRDF that was used in Chapter 15 for modeling specular highlights. The existing data members `ks` and `cs` can be used for k_r and $\boldsymbol{c}_r$, respectively, and `exp` can still be used for e. We have to add a sampler object and write the `sample_f` function. Listing 25.1 shows the function `Glossyspecular::set_samples`, which sets up multi-jittered samples. This function will be called by the material that implements glossy reflection (see Listing 25.3). You should

```
void
GlossySpecular::set_samples(const int num_samples, const float exp) {
     sampler_ptr = new MultiJittered(num_samples);
     sampler_ptr->map_samples_to_hemisphere(exp);
}
```

Listing 25.1. The function `Glossyspecular::set_samples`. Notice that this maps the samples to a hemisphere with the specified cosine power distribution.

```
RGBColor
GlossySpecular::sample_f(const ShadeRec&        sr,
        const Vector3D& wo,
        Vector3D& wi,
        float& pdf) const {

     float ndotwo = sr.normal * wo;
     Vector3D r = -wo + 2.0 * sr.normal * ndotwo;  // direction of mirror
                                                   // reflection

     Vector3D w = r;
     Vector3D u = Vector3D(0.00424, 1, 0.00764) ^ w;
     u.normalize();
     Vector3D v = u ^ w;

     Point3D sp = sampler_ptr->sample_hemisphere();
     wi = sp.x * u + sp.y * v + sp.z * w;       // reflected ray direction

     if (sr.normal * wi < 0.0)           // reflected ray is below surface
          wi = -sp.x * u - sp.y * v + sp.z * w;

     float phong_lobe = pow(r * wi, exp);
     pdf = phong_lobe * (sr.normal * wi);

     return (ks * cs * phong_lobe);
}
```

Listing 25.2. The function `Glossyspecular::sample_f`.

also have a `set_sampler` function that sets a user-specified sampler, as this will allow you to experiment with different sampling techniques.

The function `Glossyspecular::sample_f` in Listing 25.2 is the version that has the pdf as an argument because it has to return this. It also computes the direction $\boldsymbol{\omega}_i$ of the reflected ray according to Equation (25.8), computes the pdf according to Equation (25.6) but without the c, and returns the BRDF in Equation (25.2), again without the c.

I'll use the `AreaLight` tracer for glossy reflection because, again, it saves us writing a new tracer and allows us to render glossy reflections with all of the light sources.

We also need a `GlossyReflector` material that inherits from `Phong` and has its own `GlossySpecular` BRDF. This allows us to include direct illumination and to set the direct and indirect glossy reflection components independently, including having no direct specular reflection. Listing 25.3 shows

```
class GlossyReflector: public Phong {
     public:

          // constructors etc.

          void
          set_samples(const int num_samples, const float exp);

          void
          set_kr(const float k);

          void
          set_exponent(const float exp);

          virtual RGBColor
          area_light_shade(ShadeRec& sr);

     private:

          GlossySpecular* glossy_specular_brdf;
};

inline void
GlossyReflector::set_samples(const int num_samples, const float exp) {
     glossy_specular_brdf->set_samples(num_samples, exp);
}

inline void
GlossyReflector::set_kr(const float k) {
     glossy_specular_brdf->set_ks(k);
}

inline void
GlossyReflector::set_exponent(const float exp) {
     glossy_specular_brdf->set_exp(exp);
}
```

Listing 25.3. Sample code from the `GlossyReflector` class.

part of the class declaration for `GlossyReflector`, along with three access functions. Note that `set_samples` just calls the function `glossy_specular_brdf->set_samples` from Listing 25.1, and `set_kr` sets `glossy_specular_brdf::ks`. The set functions for c_r (not shown) similarly set `glossy_specular_brdf::cs`. The `set_exponent` function sets e in `glossy_specular_brdf` and is required

```
RGBColor
GlossyReflector::area_light_shade(ShadeRec& sr) {
     RGBColor L(Phong::area_light_shade(sr));      // direct illumination
     Vector3D wo(-sr.ray.d);
     Vector3D wi;
     float    pdf;
     RGBColor fr(glossy_specular_brdf->sample_f(sr, wo, wi, pdf));
     Ray reflected_ray(sr.hit_point, wi);

     L += fr * sr.w.tracer_ptr->trace_ray(reflected_ray, sr.depth + 1) *
     (sr.normal * wi) / pdf;

     return (L);
}
```

Listing 25.4. The function `GlossyReflector::area_light_shade`.

as a separate function from the inherited function `set_exp`, which sets e in `Phong:specular_brdf`.

Listing 25.4 shows the function `GlossyReflector::area_light_shade`, where the radiance is divided by the pdf.

25.3 Results

A good place to start is the scene in Figure 24.30(c) with a glossy reflector material on the sphere, as this will allow us to compare the results with mirror reflection. Listing 25.5 is a fragment from the build function that illustrates how to set up the glossy reflector material, which, in this case, has no direct-illumination component.

Figure 25.8 shows the results for six values of e that range from $e = 1.0$ in Figure 25.8(a) to $e = 100000.0$ in Figure 25.8(f). Figure 25.8(a), which is Lambertian reflection, shows no reflected image. There is some image visible in Figure 25.8(b) with $e = 10.0$. The images become progressively sharper in Figure 25.8(c)–(e). Figure 25.8(f) is visually identical to Figure 24.30(c). The more glossy the material, the more rays per pixel we need to reduce the noise to acceptable amounts. These images were rendered with multi-jittered sampling and 256 samples per pixel in Figure 25.8(a) and (b), 100 samples in Figure 25.8(c) and (d), and 25 samples in Figure 25.8(e) and (f). See Figure 25.13 for the consequences of using fewer samples per pixel and of using non-random sampling patterns.

```
int nun_samples = 100;
...

float exp = 100.0;
GlossyReflector* glossy_ptr = new GlossyReflector;
glossy_ptr->set_samples(num_samples, exp);
glossy_ptr->set_ka(0.0);
glossy_ptr->set_kd(0.0);
glossy_ptr->set_ks(0.0);
glossy_ptr->set_exp(exp);
glossy_ptr->set_cd(1.0, 1.0, 0.3);
glossy_ptr->set_kr(0.9);
glossy_ptr->set_exponent(exp);
glossy_ptr->set_cr(1.0, 1.0, 0.3); // lemon
```

Listing 25.4. Code fragment from the build function for Figure 25.8.

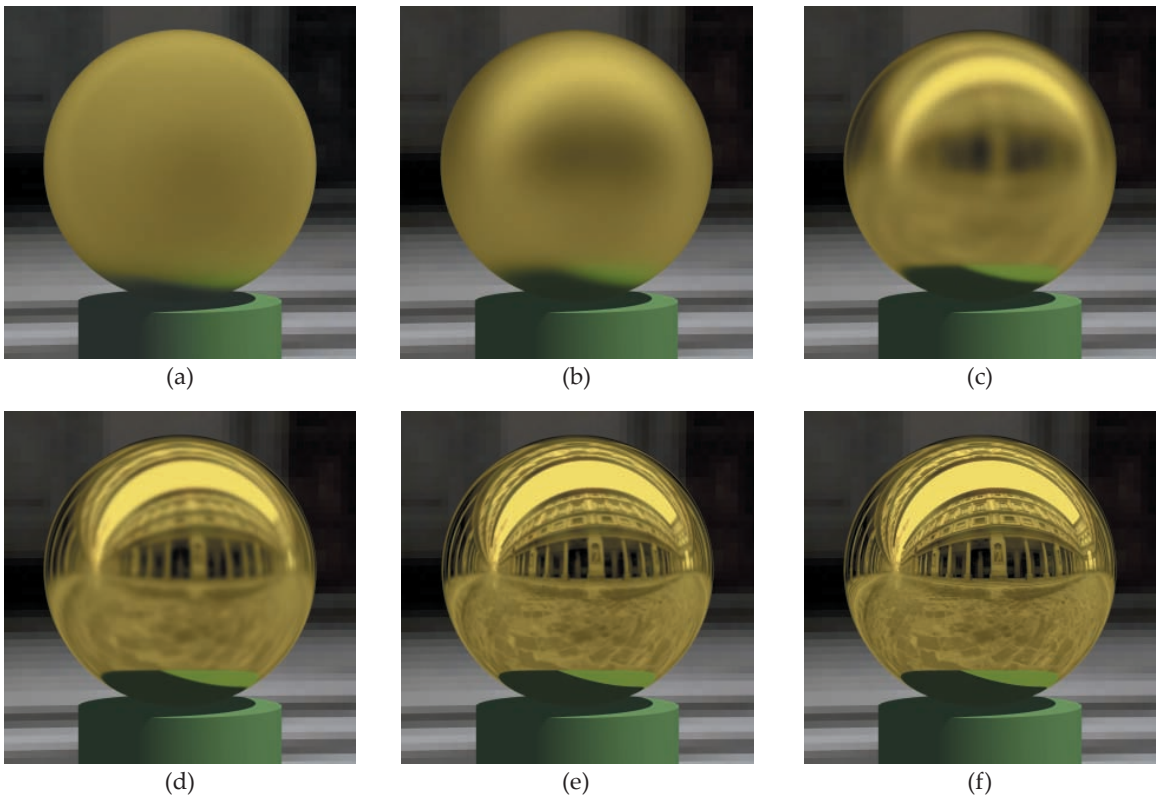


Figure 25.8. Glossy sphere surrounded by the Uffizi image and rendered with the following values of e: (a) 1.0; (b) 10.0; (c) 100.0; (d) 1000.0; (e) 10000.0; (f) 100000.0.

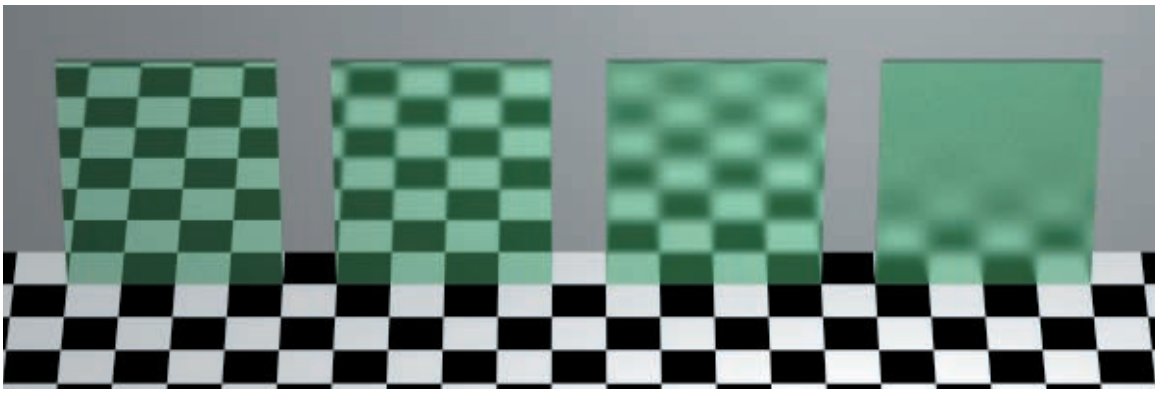
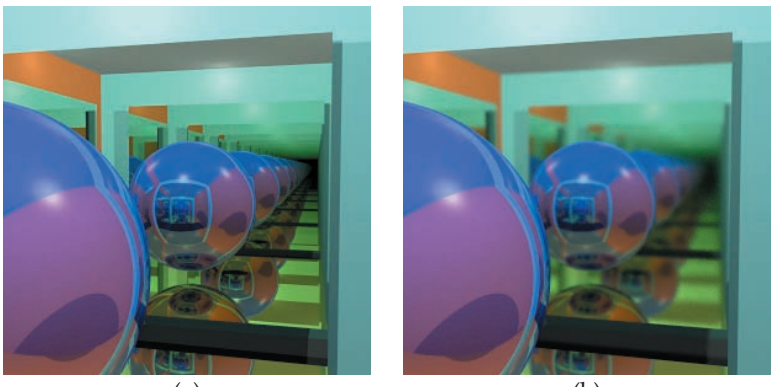


Figure 25.9. Thin blocks with glossy reflection and direct illumination. From left to right, e = 100000.0, 10000.0, 1000.0, 100.0. This was inspired by a grayscale figure in Westlund and Meyer (2001).

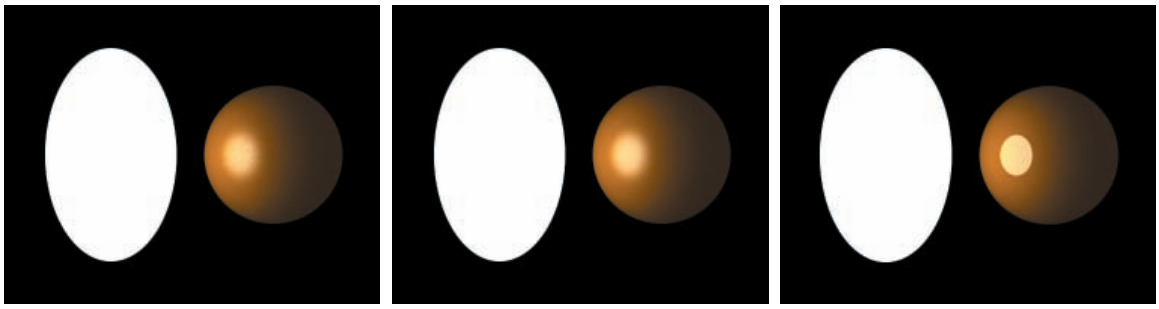
The glossy blocks in Figure 25.9 include ambient and diffuse shading. Notice how the reflections on all of the blocks are sharp where they touch the plane (on which they are sitting). This image was rendered with multi-jittered sampling and 256 samples per pixel because of the block on the right.

Figure 25.10(a) shows the hall of mirrors from Section 24.7.2 with the camera looking at the mirror on the back wall. This was rendered with `max_depth` = 19. Figure 25.10(b) shows the scene where the two mirrors on the front and back walls have a glossy reflector material with e = 25000.0. Notice how the



(a) (b)

Figure 25.10. Hall of mirrors rendered with perfect specular reflection (a) and with glossy reflection with e = 25000.0 (b).



(a) (b) (c)

Figure 25.11. Direct and indirect illumination with a disk light: (a) specular highlight with $e = 25$; (b) glossy reflection with $e = 25$; (c) simulation of mirror reflection with $e = 100000$. Images rendered with 100 multi-jittered samples per pixel.

reflections become more blurred with each bounce, in spite of the high value of e. Incidentally, this scene would not be renderable with a branching factor greater than one; for example, starting with one primary ray per pixel and a branching factor of 2, we would end up with 2^{20} rays! This would, of course, be silly to attempt, but as you'll see in Chapter 27, ray tracing transparent objects essentially results in a branching factor of 2.

Figure 25.11 shows how we can use the `GlossyReflector` material to simulate different optical effects. This figure shows a glossy sphere illuminated by a disk light. In Figure 25.11(a), the sphere is only shaded with direct illumination, where $k_s = 0.65$ and $e = 25$, and so the bright area is the specular highlight. In Figure 25.11(b), there's no direct specular reflection, but there is glossy reflection with $k_r = 0.65$ and $e = 25$ again. The bright area is now the reflection of the light. Notice that there's little visible difference between these two images, demonstrating that specular highlights are the smeared-out reflections of the lights. Finally, Figure 25.11(c) is the same as Figure 25.11(b) but rendered with $e = 100000.0$ to simulate a perfect mirror. We could, of course, get this same effect with a `Reflective` material on the sphere.

Further Reading

Hunter (1937) provided the original definitions and measurement of gloss. The technical definition of gloss involves a parameter $\lambda \in [0, 100]$, but there's no simple relation between λ and e. Wallace et al. (1987) first presented images of glossy reflection using a combination of radiosity and ray tracing. Westlund

and Meyer (2001) discusses the technical definition of gloss and presents some nice ray-traced images of glossy materials. Ashikhmin and Shirley (2001, 2002) discuss Monte Carlo techniques for ray tracing anisotropic glossy reflections. Shirley and Morley (2003) discusses glossy reflection with the pdf (25.6). Dutré et al. (2003) discusses a model for glossy reflection that is similar to the model used in this chapter and discuss various pdfs that could be used, along with their advantages and disadvantages. Pharr and Humphreys (2004) discusses Monte Carlo techniques for ray tracing glossy reflection based on the Blinn (1967) microfacet distribution and on Ashikhmin and Shirley's anisotropic model. These models are considerably more complex than the model presented here.

Questions

25.1. Figure 25.12 shows a glossy sphere with $e = 100$ and a checker plane. Can you explain the crescent-shaped reflections of the plane and background color on the sphere? These are visible near where the sphere cuts the horizon.

25.2. Figure 25.13(a) is the sphere in Figure 25.8(a) with $e = 1$ but rendered with one random sample per pixel. As expected, the sphere surface is completely covered with noise. Figure 25.13(b) is rendered with one uniform sample per pixel, and Figure 25.13(c) is rendered with one Hammersley sample per pixel. Can you explain the mirrored appearance of this Lambertian sphere in Figure 25.13(b) and (c)? What do these images tell you about

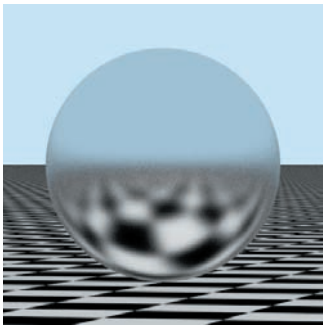


Figure 25.12. Glossy sphere and checker plane.

(a) (b) (c)

Figure 25.13. The Lambertian sphere in Figure 25.8(a) rendered with one sample per pixel and the following sampling patterns: (a) random; (b) uniform; (c) Hammersley.

the suitability of uniform and Hammersley sampling for simulating glossy reflection? See also Exercise 25.4.

Exercises

25.1. Implement glossy reflection as described in this chapter, and use the scene in Figure 25.8 to test your implementation.

25.2. As the scene in Figure 25.8 has a point light, add a specular highlight to the sphere with the same specular parameters as used for the glossy reflection.

25.3. Add ambient and diffuse illumination to the sphere in Figure 25.8.

25.4. Experiment with different sampling patterns in Figures 25.8 and 25.9 and different numbers of samples per pixel, starting with one sample.

25.5. Render the hall of mirrors in Figure 25.10 with different values of e, `max_depth`, numbers of samples, and sampling patterns.

25.6. Render the sphere and checker plane in Figure 25.12 with different values of e and numbers of samples.

25.7. Render some glossy objects illuminated with area lights. An example is the figure on the first page of this chapter, where the plane, large sphere, cylinder, and box are glossy and where the scene is illuminated by a point light and an environment light. Notice the color bleeding onto the plane, which has $e = 1.0$. You can render this scene without the texture on the plane.

25.8. In the figure on the first page of this chapter, the glossy objects have two specular highlights, one from the point light and the other from the environment light. Re-render this image with the specular highlight color set to red and compare the results with the existing image. Start with one sample per pixel.

25.9. Render some nice glossy-reflection images of your own.