Monte Carlo
(Helper Slides)

Cosine Sampling

$$p(\Psi) = \cos(\Psi, N_x)/\pi.$$

If we also assume that the BRDF f_r is diffuse at x, we obtain the following estimator:

$$\langle L_{indirect}(x \to \Theta) \rangle = \frac{\pi f_r}{N} \sum_{i=1}^{N} L_r(r(x, \Psi_i) \to -\Psi_i).$$

Direct and Indirect Illumination

$$L_r(x \to \Theta) = \int_{\Omega_x} L(x \leftarrow \Psi) f_r(x, \Theta \leftrightarrow \Psi) \cos(\Psi, N_x) d\omega_\Psi$$
$$= \int_{\Omega_x} L(r(x, \Psi) \to -\Psi) f_r(x, \Theta \leftrightarrow \Psi) \cos(\Psi, N_x) d\omega_\Psi .$$

Rewriting $L(r(x, \Psi) \to -\Psi)$ as a sum of self-emitted and reflected radiance at $r(x, \Psi)$ gives us

$$\begin{aligned} L_r(x \to \Theta) = &\int_{\Omega_x} L_e(r(x, \Psi) \to -\Psi) f_r(x, \Theta \leftrightarrow \Psi) \cos(\Psi, N_x) d\omega_\Psi \\ &+ \int_{\Omega_x} L_r(r(x, \Psi) \to -\Psi) f_r(x, \Theta \leftrightarrow \Psi) \cos(\Psi, N_x) d\omega_\Psi \\ = &L_{direct}(x \to \Theta) + L_{indirect}(x \to \Theta). \end{aligned} \tag{5.2}$$

Direct Illumination

$$L_{direct}(x \to \Theta) = \int_{A_{\mathrm{sources}}} L_e(y \to \overrightarrow{yx}) f_r(x, \Theta \leftrightarrow \overrightarrow{xy}) G(x, y) V(x, y) dA_y, \tag{5.3}$$

or by explicitly summing over all N_L light sources in the scene,

$$L_{direct}(x \to \Theta) = \sum_{k=1}^{N_L} \int_{A_k} L_e(y \to \overrightarrow{yx}) f_r(x, \Theta \leftrightarrow \overrightarrow{xy}) G(x, y) V(x, y) dA_y.$$

Area Light Sources

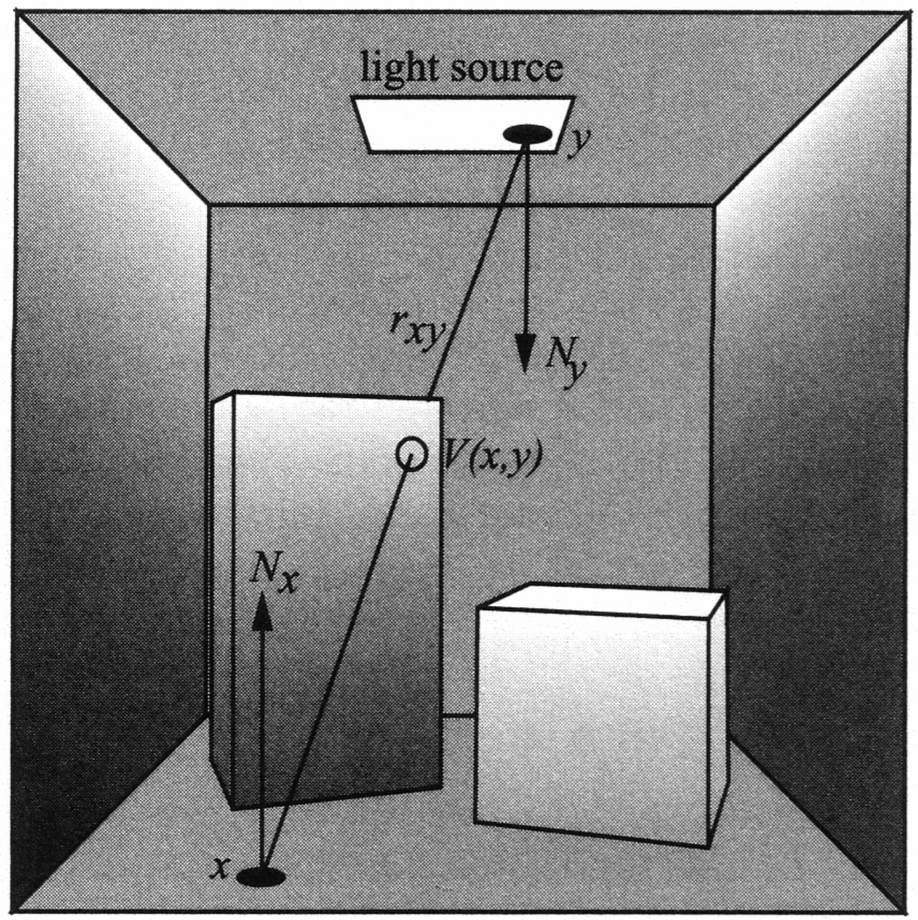

Figure 5.6. Area integration over light sources for direct illumination.

Area Light Sources

$$\langle L_{direct}(x \to \Theta)\rangle = \frac{1}{N_s}\sum_{i=1}^{N_s}\frac{L_e(y_i \to \overrightarrow{y_i x})f_r(x, \Theta \leftrightarrow \overrightarrow{x y_i})G(x, y_i)V(x, y_i)}{p(y_i)}.$$

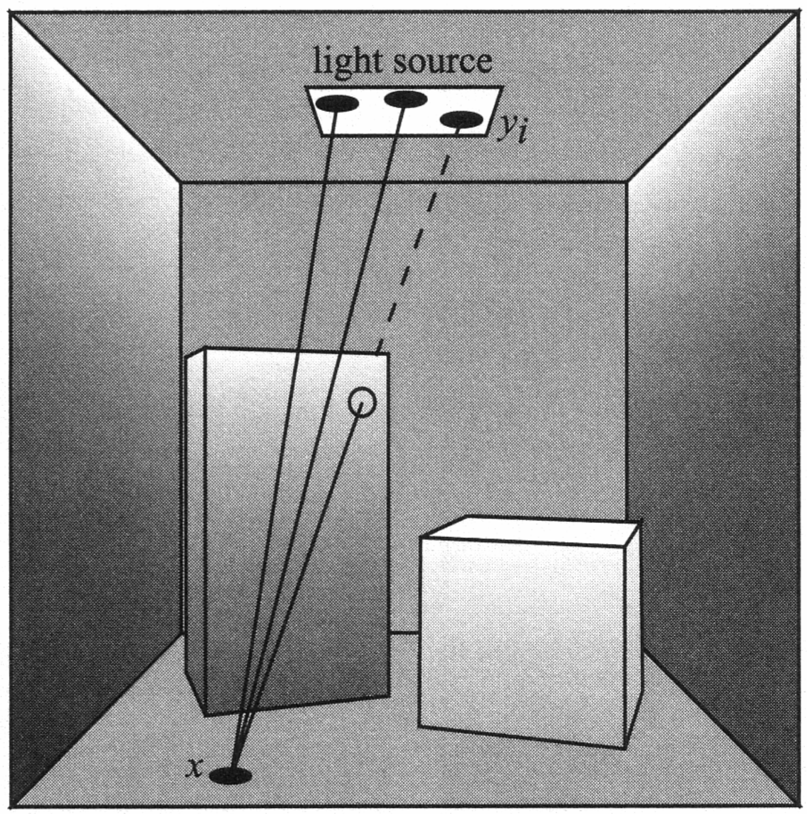

Figure 5.8. Uniform light-source sampling for direct illumination.

Area Light Sources

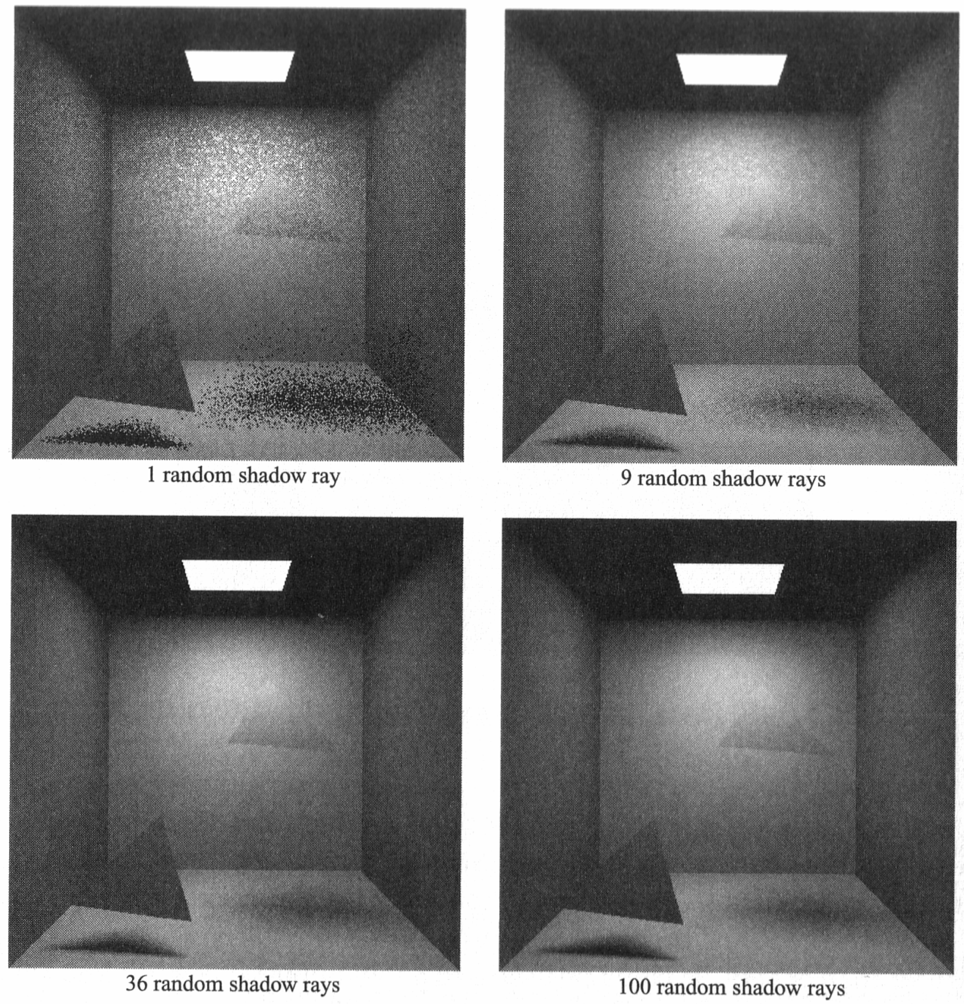

Figure 5.9. Uniform light source sampling. The images are generated with 1 viewing ray per pixel, and 1, 9, 36, and 100 shadow rays. The difference in quality between soft shadows and hard shadows is noticeable.

Multiple Light Sources

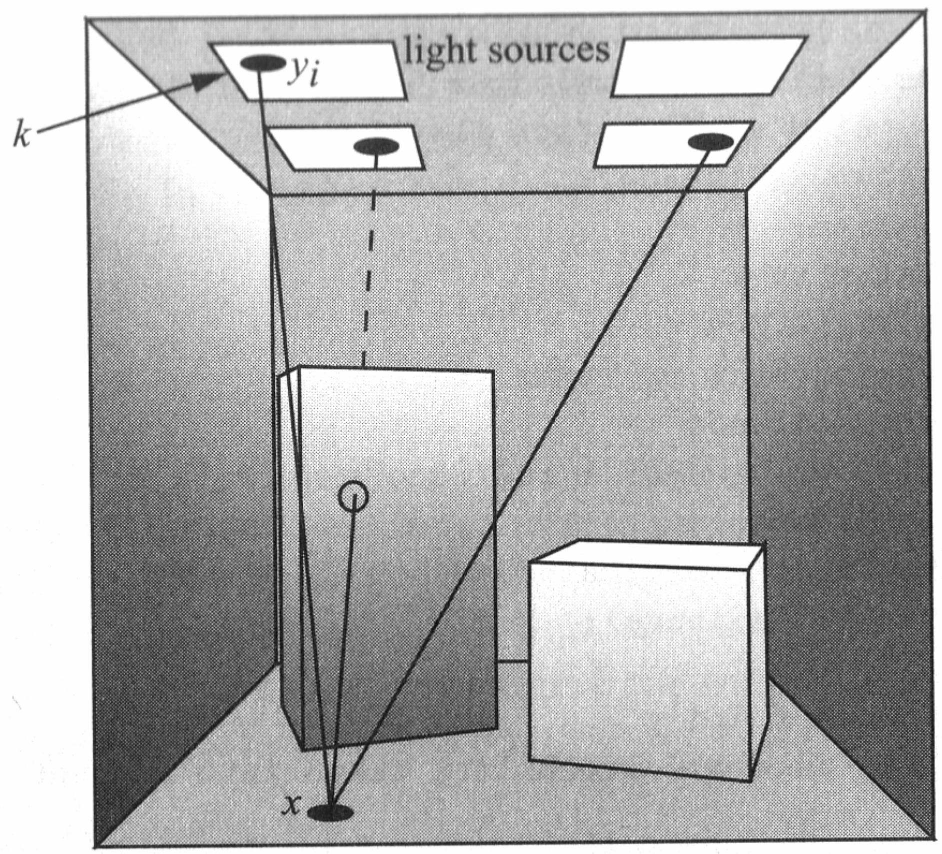

Figure 5.10. Sampling multiple light sources for direct illumination.

Other Light Source Samplings

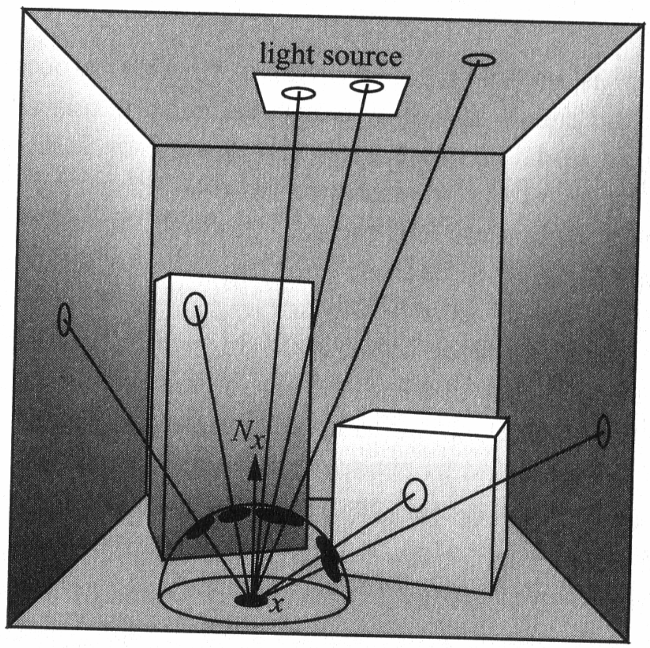

Figure 5.12. Shadow rays generated using hemisphere sampling.

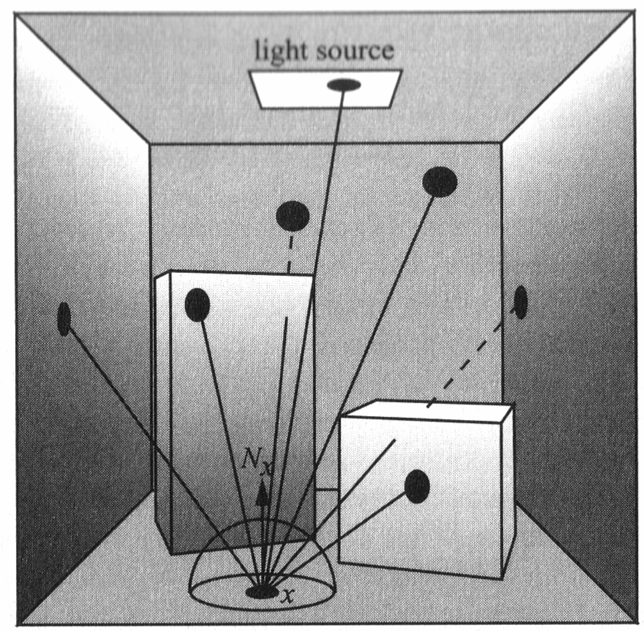

Figure 5.13. Shadow rays generated using global area sampling.

Computing Direct Illumination

```
// direct illumination from a single light source
// for a surface point x, direction theta
directIllumination (x, theta)
    estimatedRadiance = 0;
    for all shadow rays
        generate point y on light source;
        estimatedRadiance +=
            Le(y, yx) * BRDF * radianceTransfer(x,y)/pdf(y);
    estimatedRadiance = estimatedRadiance / #shadowRays;
    return(estimatedRadiance);

// transfer between x and y
// 2 cosines, distance and visibility taken into account
radianceTransfer(x,y)
    transfer = G(x,y)*V(x,y);
    return(transfer);
```

Figure 5.7. Computing direct illumination from a single light source.

Indirect Illumination

$$L_{indirect}(x \to \Theta) = \int_{\Omega_x} L_r(r(x, \Psi) \to -\Psi) f_r(x, \Theta \leftrightarrow \Psi) \cos(\Psi, N_x) d\omega_\Psi.$$

N random directions Ψ_i. This produces the following estimator:

$$\langle L_{indirect}(x \to \Theta) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L_r(r(x, \Psi_i) \to -\Psi_i) f_r(x, \Theta \leftrightarrow \Psi_i) \cos(\Psi_i, N_x)}{p(\Psi_i)}.$$

Indirect Illumination

$$L_{indirect}(x \to \Theta) = \int_{A_{\mathrm{scene}}} L_r(y \to \overrightarrow{yx}) f_r(x, \Theta \leftrightarrow \overrightarrow{xy}) G(x, y) V(x, y) dA_y.$$

The corresponding estimator when using a PDF $p(y)$:

$$\langle L_{indirect}(x \to \Theta) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L_r(y_i \to \overrightarrow{y_i x}) f_r(x, \Theta \leftrightarrow \overrightarrow{x y_i}) G(x, y_i) V(x, y_i)}{p(y_i)}.$$

Indirect Illumination

```
// indirect illumination
// for surface point x, direction theta
indirectIllumination (x, theta)
    estimatedRadiance = 0;
    if (no absorption)
        for all indirect paths
            sample direction psi on hemisphere;
            y = trace(x, psi);
            estimated radiance +=
                computeRadiance(y, -psi) * BRDF *
                cos(Nx, psi) / pdf(psi);
        estimatedRadiance = estimatedRadiance / #paths;
    return(estimatedRadiance/(1-absorption));

computeRadiance(x, dir)
    estimatedRadiance = Le(x, dir);
    estimatedRadiance += directIllumination(x, dir);
    estimatedRadiance += indirectIllumination(x, dir);
    return(estimatedRadiance);
```

Figure 5.14. Computing indirect illumination.

Full algorithm...

```
// global illumination algorithm
// stochastic ray tracing
computeImage(eye)
    for each pixel
        radiance = 0;
        for each sample          // Np viewing rays
            pick sample point p within support of h;
            construct ray at eye, direction p-eye;
            radiance = radiance + rad(ray)*h(p);
        radiance = radiance/#samples;

rad(ray)
    find closest intersection point x of ray with scene;
    computeRadiance(x, eye-x);

compute_radiance(x, dir)
    estimatedRadiance = Le(x, dir);
    estimatedRadiance += directIllumination(x, dir);
    estimatedRadiance += indirectIllumination(x, dir);
    return(estimatedRadiance);

directIllumination (x, theta)
    estimatedRadiance = 0;
    for all shadow rays          // Nd shadow rays
        select light source k;
        sample point y on light source k;
        estimated radiance +=
            Le * BRDF * radianceTransfer(x,y)/(pdf(k)pdf(y|k));
    estimatedRadiance = estimatedRadiance / #paths;
    return(estimatedRadiance);

indirectIllumination (x, theta)
    estimatedRadiance = 0;
    if (no absorption)           // Russian roulette
        for all indirect paths  // Ni indirect rays
            sample direction psi on hemisphere;
            y = trace(x, psi);
            estimatedRadiance +=
                compute_radiance(y, -psi) * BRDF *
                cos(Nx, psi)/pdf(psi);
        estimatedRadiance = estimatedRadiance / #paths;
    return(estimatedRadiance/(1-absorption));

radianceTransfer(x,y)
    transfer = G(x,y)*V(x,y);
    return(transfer);
```

Figure 5.16. Total global illumination algorithm.