Material Modeling 8

CS 184: Foundations of Computer Graphics

Key Concepts and Equations

The Reflection Equation

Calculates the outgoing radiance L_r from a point p by integrating incoming radiance L_i over the hemisphere H^2, modulated by the BRDF f_r and the cosine of the incident angle.

$$L_r(\mathrm{p}, \omega_r) = \int_{H^2} f_r(\mathrm{p}, \omega_i \to \omega_r)\, L_i(\mathrm{p}, \omega_i) \cos\theta_i \, \mathrm{d}\omega_i$$

Physically Plausible BRDF Properties

- **Positivity:** $f(\omega_i, \omega_o) \geq 0$
- **Helmholtz Reciprocity:** $f(\omega_i, \omega_o) = f(\omega_o, \omega_i)$
- **Energy Conservation:** $\int_{H^2} f(\omega_i, \omega_o) \cos\theta_o d\omega_o \leq 1$

Common BRDF Models

- **Lambertian (Diffuse):** $f_r = \frac{\rho}{\pi}$, where ρ is the albedo.
- **Perfect Mirror (Specular):** $f_r(\theta_i, \phi_i; \theta_o, \phi_o) = \frac{\delta(\cos\theta_i - \cos\theta_o)}{\cos\theta_i} \delta(\phi_i - \phi_o \pm \pi)$

Microfacet Theory (Cook-Torrance Model)

Models a surface as a collection of microscopic mirrors. The half-vector $\mathbf{h} = \frac{\mathbf{i}+\mathbf{o}}{||\mathbf{i}+\mathbf{o}||}$ is key.

$$f(\mathbf{i}, \mathbf{o}) = \frac{\mathbf{F}(\mathbf{i}, \mathbf{h})\mathbf{G}(\mathbf{i}, \mathbf{o}, \mathbf{h})\mathbf{D}(\mathbf{h})}{4(\mathbf{n} \cdot \mathbf{i})(\mathbf{n} \cdot \mathbf{o})}$$

- **F**: Fresnel term (reflectance vs. angle)
- **G**: Geometry term (shadowing/masking)
- **D**: Normal Distribution Function (microfacet orientation)

Schlick's Approx. for Fresnel: $F(\theta) \approx R_0 + (1 - R_0)(1 - \cos\theta)^5$ with $R_0 = \left(\frac{\eta_1 - \eta_2}{\eta_1 + \eta_2}\right)^2$

Geometric Optics

- **Snell's Law:** $\eta_i \sin(\theta_i) = \eta_t \sin(\theta_t)$
- **Total Internal Reflection (Critical Angle):** $\theta_c = \arcsin\left(\frac{\eta_t}{\eta_i}\right)$ for $\eta_i > \eta_t$

Monte Carlo Integration

The Monte Carlo estimator for an integral $I = \int_a^b f(x)dx$ using N samples drawn from a probability distribution $p(x)$ is:

$$I \approx \frac{1}{N} \sum_{k=1}^{N} \frac{f(x_k)}{p(x_k)}$$

For rendering, this is often used for the reflection equation, integrating over the hemisphere H^2. For cosine-weighted sampling, the PDF is $p(\omega) = \frac{\cos\theta}{\pi}$.

1 Optics for Ideal Surfaces

1. **Specular Reflection**

   A light ray with direction $\boldsymbol{\omega_i}$ is incident on a perfect mirror with normal $\boldsymbol{n}$. Derive the formula for the outgoing reflected direction $\boldsymbol{\omega_o}$.

   Solution: The reflection vector $\boldsymbol{\omega_o}$ is found by reflecting the incident vector $\boldsymbol{\omega_i}$ about the normal $\boldsymbol{n}$. The component of $\boldsymbol{\omega_i}$ parallel to the normal is $(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}$. The component perpendicular to the normal is $\boldsymbol{\omega_i} - (\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}$. To reflect, we flip the parallel component and keep the perpendicular component the same.

   $$\begin{aligned}\boldsymbol{\omega_o} &= -(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n} + (\boldsymbol{\omega_i} - (\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}) \\ &= \boldsymbol{\omega_i} - 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}\end{aligned}$$

   Note: The formula $\boldsymbol{\omega_o} = -\boldsymbol{\omega_i} + 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}$ is also correct and perhaps more common; it corresponds to a different geometric construction.

2. **Snell's Law and Total Internal Reflection**

   A ray of light passes from a medium with refractive index η_i to a medium with refractive index η_t. The incident angle is θ_i. Derive the critical angle θ_c at which total internal reflection occurs.

   Solution: Snell's Law states $\eta_i \sin(\theta_i) = \eta_t \sin(\theta_t)$. Total internal reflection occurs when light travels from a denser medium to a less dense medium ($\eta_i > \eta_t$) and the angle of refraction θ_t reaches $90°$. At this point, the incident angle is the critical angle θ_c.

   $$\eta_i \sin(\theta_c) = \eta_t \sin(90°) = \eta_t$$

   $$\theta_c = \arcsin\left(\frac{\eta_t}{\eta_i}\right)$$

   For any incident angle greater than θ_c, the ray will be perfectly reflected.

3. **Fresnel-Schlick Approximation**

   The Fresnel equations are complex. The Schlick approximation simplifies them for dielectrics:

   $$R(\theta) = R_0 + (1 - R_0)(1 - \cos\theta)^5 \quad \text{where} \quad R_0 = \left(\frac{\eta_1 - \eta_2}{\eta_1 + \eta_2}\right)^2$$

   Explain the physical meaning of R_0 and the purpose of the $(1 - \cos\theta)^5$ term.

   Solution: R_0 is the reflectance at normal incidence (looking straight at the surface). It represents the base reflectivity of the material. The $(1 - \cos\theta)^5$ term is a polynomial approximation of the angular dependence of reflectance. It smoothly interpolates the reflectance from its base value R_0 at normal incidence ($\theta = 0$) to 1.0 at grazing angles ($\theta = 90°$), capturing the characteristic increase in reflectivity at shallow viewing angles for dielectric materials.

4. **Multi-Layer Refraction and Reflection**

A ray of light passes through a sequence of three media with refractive indices $\eta_1 = 1.0$ (air), $\eta_2 = 1.5$ (glass), and $\eta_3 = 1.33$ (water). The glass layer has thickness $d = 10$ mm. The ray enters from air at an angle of $\theta_1 = 30°$ from the normal.

1. Calculate the angle of refraction θ_2 in the glass layer.
2. Calculate the angle of refraction θ_3 in the water layer.
3. Calculate the lateral displacement of the ray after passing through the glass layer.
4. Using the Fresnel-Schlick approximation, calculate the reflectance at each interface. Assume normal incidence reflectances of:
   - Air-to-glass: $R_0 = \left(\frac{1.0-1.5}{1.0+1.5}\right)^2 = 0.04$
   - Glass-to-water: $R_0 = \left(\frac{1.5-1.33}{1.5+1.33}\right)^2 = 0.0041$
5. If the incident light has intensity $I_0 = 100$, calculate the intensity of light that makes it through both layers, accounting for reflection losses at both interfaces but ignoring absorption.

Solution:

1. Using Snell's Law at the first interface:

$$\eta_1 \sin(\theta_1) = \eta_2 \sin(\theta_2)$$
$$1.0 \sin(30°) = 1.5 \sin(\theta_2)$$
$$\sin(\theta_2) = \frac{1.0 \sin(30°)}{1.5} = \frac{0.5}{1.5} = 0.333$$
$$\theta_2 = \arcsin(0.333) \approx 19.5°$$

2. Using Snell's Law at the second interface:

$$\eta_2 \sin(\theta_2) = \eta_3 \sin(\theta_3)$$
$$1.5 \sin(19.5°) = 1.33 \sin(\theta_3)$$
$$\sin(\theta_3) = \frac{1.5 \sin(19.5°)}{1.33} = \frac{1.5 \cdot 0.333}{1.33} = \frac{0.5}{1.33} = 0.376$$
$$\theta_3 = \arcsin(0.376) \approx 22.1°$$

3. The lateral displacement is:

$$\Delta x = d \cdot \tan(\theta_2) = 10 \text{ mm} \cdot \tan(19.5°) = 10 \text{ mm} \cdot 0.355 = 3.55 \text{ mm}$$

4. Reflectance at each interface:

   For air-to-glass interface:

$$R_1 = 0.04 + (1 - 0.04)(1 - \cos(30°))^5 = 0.04 + 0.96(1 - 0.866)^5 = 0.04 + 0.96 \cdot (0.134)^5$$
$$= 0.04 + 0.96 \cdot 0.0000499 \approx 0.04 + 0.000048 = 0.04005$$

   For glass-to-water interface:

$$R_2 = 0.0041 + (1 - 0.0041)(1 - \cos(19.5°))^5 = 0.0041 + 0.9959(1 - 0.943)^5$$
$$= 0.0041 + 0.9959 \cdot (0.057)^5 \approx 0.0041 + 0.9959 \cdot 0.00000175 = 0.0041 + 0.00000174 = 0.00410$$

5. The intensity after the first interface (entering glass) is:

$$I_1 = I_0 \cdot (1 - R_1) = 100 \cdot (1 - 0.04005) = 100 \cdot 0.95995 = 95.995$$

The intensity after the second interface (entering water) is:

$$I_2 = I_1 \cdot (1 - R_2) = 95.995 \cdot (1 - 0.00410) = 95.995 \cdot 0.9959 = 95.602$$

Therefore, approximately 95.6

5. **Symbolic Vector Manipulation: Perfect Specular Reflection**

Given the incident direction vector $\boldsymbol{\omega_i}$ and the surface normal $\boldsymbol{n}$, we can derive the reflected direction $\boldsymbol{\omega_o}$ using vector decomposition.

1. Starting with the relationship $\boldsymbol{\omega_o} + \boldsymbol{\omega_i} = 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}$, solve for $\boldsymbol{\omega_o}$ in terms of $\boldsymbol{\omega_i}$ and $\boldsymbol{n}$.
2. Now consider a ray with direction $\boldsymbol{\omega_i} = (0.6, 0.8, 0)$ incident on a surface with normal $\boldsymbol{n} = (0, 1, 0)$. Using your derived formula, express the reflected direction $\boldsymbol{\omega_o}$ symbolically without computing numerical values.
3. Prove that the angle of incidence equals the angle of reflection by showing that $\boldsymbol{\omega_i} \cdot \boldsymbol{n} = \boldsymbol{\omega_o} \cdot \boldsymbol{n}$.

Solution:

1. Starting with the relationship:

$$\boldsymbol{\omega_o} + \boldsymbol{\omega_i} = 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}$$

Solving for $\boldsymbol{\omega_o}$:

$$\boldsymbol{\omega_o} = 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n} - \boldsymbol{\omega_i}$$

This is the reflection formula: $\boldsymbol{\omega_o} = -\boldsymbol{\omega_i} + 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}$

2. For $\boldsymbol{\omega_i} = (0.6, 0.8, 0)$ and $\boldsymbol{n} = (0, 1, 0)$:

First, compute the dot product:

$$\boldsymbol{\omega_i} \cdot \boldsymbol{n} = 0.6 \cdot 0 + 0.8 \cdot 1 + 0 \cdot 0 = 0.8$$

Then apply the reflection formula:

$$\boldsymbol{\omega_o} = -\boldsymbol{\omega_i} + 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}$$

$$\boldsymbol{\omega_o} = -(0.6, 0.8, 0) + 2(0.8)(0, 1, 0)$$
$$\boldsymbol{\omega_o} = (-0.6, -0.8, 0) + (0, 1.6, 0)$$
$$\boldsymbol{\omega_o} = (-0.6, 0.8, 0)$$

3. To prove that the angle of incidence equals the angle of reflection, we need to show that $\boldsymbol{\omega_i} \cdot \boldsymbol{n} = \boldsymbol{\omega_o} \cdot \boldsymbol{n}$:

$$\boldsymbol{\omega_o} \cdot \boldsymbol{n} = [-\boldsymbol{\omega_i} + 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})\boldsymbol{n}] \cdot \boldsymbol{n}$$

$$= -(\boldsymbol{\omega_i} \cdot \boldsymbol{n}) + 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})(\boldsymbol{n} \cdot \boldsymbol{n})$$

Since $\boldsymbol{n}$ is a unit vector, $\boldsymbol{n} \cdot \boldsymbol{n} = 1$:

$$= -(\boldsymbol{\omega_i} \cdot \boldsymbol{n}) + 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})$$
$$= -(\boldsymbol{\omega_i} \cdot \boldsymbol{n}) + 2(\boldsymbol{\omega_i} \cdot \boldsymbol{n})$$
$$= \boldsymbol{\omega_i} \cdot \boldsymbol{n}$$

Therefore, $\boldsymbol{\omega_i} \cdot \boldsymbol{n} = \boldsymbol{\omega_o} \cdot \boldsymbol{n}$, which proves that the angle of incidence equals the angle of reflection.

2 BRDF

1. **The Reflection Equation**

   The reflection equation describes the equilibrium distribution of radiance at a surface point:

   $$L_r(\mathrm{p}, \omega_r) = \int_{H^2} f_r(\mathrm{p}, \omega_i \to \omega_r) L_i(\mathrm{p}, \omega_i) \cos\theta_i \, \mathrm{d}\omega_i$$

   Explain the role of each component and why this is an integral equation.

   Solution:

   - $L_r(\mathrm{p}, \omega_r)$: Outgoing radiance at point p in direction ω_r.
   - $f_r(\ldots)$: The BRDF, a 4D function defining how light scatters from a surface.
   - $L_i(\mathrm{p}, \omega_i)$: Incoming radiance at point p from direction ω_i.
   - $\cos\theta_i$: Projects the incident differential solid angle onto the surface, accounting for foreshortening.

   This is an integral equation because the outgoing radiance L_r depends on the incoming radiance L_i from all directions over the hemisphere, which itself is the outgoing radiance from other surfaces. This recursive relationship is the foundation of light transport algorithms.

2. **Physical Plausibility**

   What three properties must a BRDF satisfy to be physically plausible? Provide the mathematical condition for each.

   Solution:

   1. **Positivity:** $f_r(\omega_i \to \omega_o) \geq 0$
   2. **Helmholtz Reciprocity:** $f_r(\omega_i \to \omega_o) = f_r(\omega_o \to \omega_i)$
   3. **Energy Conservation:** $\forall\omega_i, \int_{H^2} f_r(\omega_i \to \omega_o) \cos\theta_o d\omega_o \leq 1$

3. **Lambertian BRDF**

   A Lambertian BRDF is given by $f_r = \frac{\rho}{\pi}$. Show that it conserves energy.

   Solution: To conserve energy, $\int_{H^2} f_r \cos\theta_o d\omega_o \leq 1$.

   $$\int_{H^2} \frac{\rho}{\pi} \cos\theta_o d\omega_o = \frac{\rho}{\pi} \int_{H^2} \cos\theta_o d\omega_o$$

   The integral of the projected solid angle over the hemisphere is π.

   $$\frac{\rho}{\pi} \cdot \pi = \rho$$

   Since albedo ρ is in the range $[0, 1]$, the condition is satisfied.

4. **Perfect Specular BRDF**

   The BRDF for a perfect mirror can be written using Dirac delta functions. Explain its form and how it collapses the reflection equation.

$$f_r(\theta_i, \phi_i; \theta_o, \phi_o) = \frac{\delta(\cos\theta_i - \cos\theta_o)}{\cos\theta_i}\delta(\phi_i - \phi_o \pm \pi)$$

Solution: The two Dirac delta functions, $\delta(\cdot)$, constrain the reflection to a single outgoing direction based on the incoming direction, enforcing the law of reflection. The first delta function ensures the angle of incidence equals the angle of reflection ($\theta_i = \theta_o$), and the second ensures the incoming and outgoing vectors are coplanar. When this BRDF is substituted into the reflection equation, the sifting property of the delta functions causes the integral to collapse, evaluating the incoming radiance L_i only from the perfect mirror direction. The $\cos\theta_i$ term in the denominator cancels the one in the reflection equation.

5. **BRDF Walkthrough: Diffuse Reflection**

   The Bidirectional Reflectance Distribution Function (BRDF) describes how light is reflected at a surface. For a diffuse (Lambertian) surface, the BRDF is constant: $f_r = \frac{\rho}{\pi}$, where ρ is the albedo (reflectivity) of the surface.

   1. Why is the BRDF for a Lambertian surface constant across all incoming and outgoing directions?
   2. Explain why the factor of $\frac{1}{\pi}$ appears in the formula.
   3. How does the rendering equation simplify for a Lambertian surface under uniform illumination?

Solution:

1. A Lambertian surface scatters light equally in all directions, regardless of the incident direction. This is because the microstructure of the surface is assumed to be perfectly rough at the microscopic level, causing incident light to be scattered uniformly across the hemisphere.
2. The factor of $\frac{1}{\pi}$ ensures energy conservation. For a surface with albedo ρ, the total amount of reflected light must be ρ times the incident light. The π comes from integrating $\cos\theta$ over the hemisphere: $\int_\Omega \cos\theta \, d\omega = \pi$.
3. Under uniform illumination where $L_i(\omega_i) = L_i$ is constant:

$$\begin{aligned} L_o(\omega_o) &= \int_\Omega f_r \, L_i(\omega_i) \, \cos\theta_i \, d\omega_i \\ &= f_r \, L_i \int_\Omega \cos\theta_i \, d\omega_i \\ &= f_r \, L_i \, \pi \\ &= \frac{\rho}{\pi} L_i \, \pi \\ &= \rho \, L_i \end{aligned}$$

   This means that the outgoing radiance is simply the incoming radiance scaled by the albedo ρ.

6. **BRDF Physical Plausibility**

A physically plausible BRDF must satisfy certain properties. Consider the following BRDF:

$$f_r(\omega_i, \omega_o) = \frac{k\,(\omega_i \cdot \omega_o)^n}{(\omega_i \cdot n)\,(\omega_o \cdot n)}$$

where k and n are constants, ω_i is the incoming direction, ω_o is the outgoing direction, and n is the surface normal.

1. Check if this BRDF satisfies positivity.
2. Check if this BRDF satisfies Helmholtz reciprocity (symmetry).
3. Find the constraints on k and n that would make this BRDF conserve energy.

Solution:

1. Positivity requires that $f_r(\omega_i, \omega_o) \geq 0$ for all valid ω_i and ω_o.

   For physically valid directions (above the surface), we have $(\omega_i \cdot n) \geq 0$ and $(\omega_o \cdot n) \geq 0$.

   If $k \geq 0$ and n is even, then $(\omega_i \cdot \omega_o)^n \geq 0$ for all directions.

   If n is odd, then $(\omega_i \cdot \omega_o)^n$ can be negative when $(\omega_i \cdot \omega_o) < 0$.

   Therefore, the BRDF satisfies positivity if $k \geq 0$ and n is even.

2. Helmholtz reciprocity requires that $f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i)$.

   Let's check:

   $$f_r(\omega_i, \omega_o) = \frac{k\,(\omega_i \cdot \omega_o)^n}{(\omega_i \cdot n)\,(\omega_o \cdot n)}$$

   Swapping ω_i and ω_o:

   $$f_r(\omega_o, \omega_i) = \frac{k\,(\omega_o \cdot \omega_i)^n}{(\omega_o \cdot n)\,(\omega_i \cdot n)}$$

   Since the dot product is commutative, $(\omega_i \cdot \omega_o) = (\omega_o \cdot \omega_i)$. Therefore:

   $$f_r(\omega_o, \omega_i) = \frac{k\,(\omega_i \cdot \omega_o)^n}{(\omega_o \cdot n)\,(\omega_i \cdot n)} = f_r(\omega_i, \omega_o)$$

   So the BRDF satisfies Helmholtz reciprocity.

3. Energy conservation requires that for any incoming direction ω_i, the total energy reflected in all directions cannot exceed the energy received:

   $$\int_\Omega f_r(\omega_i, \omega_o)\,(\omega_o \cdot n)\,d\omega_o \leq 1$$

   Substituting our BRDF:

   $$\int_\Omega \frac{k\,(\omega_i \cdot \omega_o)^n}{(\omega_i \cdot n)\,(\omega_o \cdot n)}\,(\omega_o \cdot n)\,d\omega_o = \frac{k}{(\omega_i \cdot n)} \int_\Omega (\omega_i \cdot \omega_o)^n\,d\omega_o$$

   This integral depends on the value of n. For energy conservation, we need:

   $$\frac{k}{(\omega_i \cdot n)} \int_\Omega (\omega_i \cdot \omega_o)^n\,d\omega_o \leq 1 \text{ for all } \omega_i$$

The worst case is when $\omega_i = n$ (normal incidence), which maximizes the integral. In this case:

$$k \int_{\Omega} (\cos\theta_o)^n \, d\omega_o \leq 1$$

The integral evaluates to $\frac{2\pi}{n+1}$ for even n. Therefore:

$$k \leq \frac{n+1}{2\pi}$$

This is the constraint on k for energy conservation, assuming n is even (which we need for positivity).

7. **Symbolic Derivation: Lambertian BRDF and Cosine-Weighted Sampling**

For a Lambertian surface with BRDF $f_r = \frac{\rho}{\pi}$, we can use cosine-weighted hemisphere sampling to efficiently compute the reflected radiance.

1. Given the rendering equation for a diffuse surface:

$$L_o(\omega_o) = \int_{\Omega} \frac{\rho}{\pi} L_i(\omega_i) \cos\theta_i \, d\omega_i$$

Show that when using cosine-weighted sampling with PDF $p(\omega_i) = \frac{\cos\theta_i}{\pi}$, the Monte Carlo estimator simplifies to:

$$L_o(\omega_o) \approx \frac{\rho}{N} \sum_{j=1}^{N} L_i(\omega_i^{(j)})$$

2. Derive the equations for generating a cosine-weighted sample on the hemisphere using the following transformation from uniform random variables $\xi_1, \xi_2 \in [0, 1]$:

$$\phi = 2\pi\xi_1$$
$$\theta = \arccos(\sqrt{1 - \xi_2})$$

Prove that this transformation produces a PDF of $p(\omega) = \frac{\cos\theta}{\pi}$.

3. Show that the Jacobian determinant for the transformation from spherical coordinates (θ, ϕ) to solid angle $d\omega$ is $\sin\theta$, and use this to verify that the PDF integrates to 1 over the hemisphere.

Solution:

1. Starting with the rendering equation for a diffuse surface:

$$L_o(\omega_o) = \int_{\Omega} \frac{\rho}{\pi} L_i(\omega_i) \cos\theta_i \, d\omega_i$$

The Monte Carlo estimator for this integral with importance sampling is:

$$L_o(\omega_o) \approx \frac{1}{N} \sum_{j=1}^{N} \frac{\frac{\rho}{\pi} L_i(\omega_i^{(j)}) \cos\theta_i^{(j)}}{p(\omega_i^{(j)})}$$

When using cosine-weighted sampling with PDF $p(\omega_i) = \frac{\cos\theta_i}{\pi}$, we get:

$$L_o(\omega_o) \approx \frac{1}{N}\sum_{j=1}^{N} \frac{\frac{\rho}{\pi} L_i(\omega_i^{(j)}) \cos\theta_i^{(j)}}{\frac{\cos\theta_i^{(j)}}{\pi}}$$

$$= \frac{1}{N}\sum_{j=1}^{N} \frac{\rho L_i(\omega_i^{(j)}) \cos\theta_i^{(j)} \pi}{\cos\theta_i^{(j)} \pi}$$

$$= \frac{\rho}{N}\sum_{j=1}^{N} L_i(\omega_i^{(j)})$$

This simplification is why cosine-weighted sampling is so effective for diffuse surfaces - the $\cos\theta$ terms cancel out, reducing variance in the Monte Carlo estimator.

2. To derive the equations for cosine-weighted sampling, we start with the PDF $p(\omega) = \frac{\cos\theta}{\pi}$ for $\theta \in [0, \frac{\pi}{2}]$ and $\phi \in [0, 2\pi]$.

   The CDF for ϕ is uniform: $P(\phi) = \frac{\phi}{2\pi}$

   For θ, the CDF is:

$$P(\theta) = \int_0^\theta \frac{\sin\theta' \cos\theta'}{\pi} d\theta' = \frac{1}{\pi}\int_0^\theta \sin\theta' \cos\theta' d\theta' = \frac{1}{\pi}\int_0^\theta \frac{1}{2}\sin(2\theta') d\theta' = \frac{1}{2\pi}[1-\cos(2\theta)] = \frac{1}{\pi}\sin^2\theta$$

   To generate samples, we invert the CDFs:

$$\phi = 2\pi\xi_1$$

$$\xi_2 = \frac{1}{\pi}\sin^2\theta$$

$$\sin^2\theta = \pi\xi_2$$

$$\sin\theta = \sqrt{\pi\xi_2}$$

$$\theta = \arcsin(\sqrt{\pi\xi_2})$$

   Since $\sin^2\theta = 1 - \cos^2\theta$, we can also write:

$$1 - \cos^2\theta = \pi\xi_2$$

$$\cos^2\theta = 1 - \pi\xi_2$$

$$\cos\theta = \sqrt{1 - \pi\xi_2}$$

   For $\xi_2 \in [0, 1]$ and scaling to ensure $\cos\theta \in [0, 1]$:

$$\cos\theta = \sqrt{1 - \xi_2}$$

$$\theta = \arccos(\sqrt{1 - \xi_2})$$

3. The differential solid angle in spherical coordinates is:

$$d\omega = \sin\theta \, d\theta \, d\phi$$

This means the Jacobian determinant for the transformation from (θ, ϕ) to solid angle is $\sin\theta$.

To verify that the PDF integrates to 1:

$$\begin{aligned}
\int_\Omega p(\omega)\, d\omega &= \int_0^{2\pi} \int_0^{\pi/2} \frac{\cos\theta}{\pi} \sin\theta \, d\theta \, d\phi \\
&= \frac{1}{\pi} \int_0^{2\pi} d\phi \int_0^{\pi/2} \cos\theta \sin\theta \, d\theta \\
&= \frac{2\pi}{\pi} \int_0^{\pi/2} \frac{1}{2} \sin(2\theta)\, d\theta \\
&= \left[-\frac{1}{2} \cos(2\theta) \right]_0^{\pi/2} \\
&= \frac{1}{2} - \frac{1}{2} \cdot (-1) = 1
\end{aligned}$$

Therefore, the PDF is properly normalized over the hemisphere.

3 Microfacet BRDFs

1. **Cook-Torrance BRDF Walkthrough**

   The Cook-Torrance microfacet BRDF is given by:

   $$f_r(\omega_i, \omega_o) = \frac{F(\omega_i, \mathbf{h}) \cdot G(\omega_i, \omega_o, \mathbf{h}) \cdot D(\mathbf{h})}{4 \cdot (\omega_i \cdot \mathbf{n}) \cdot (\omega_o \cdot \mathbf{n})}$$

   where $\mathbf{h} = \frac{\omega_i + \omega_o}{|\omega_i + \omega_o|}$ is the halfway vector.

   1. What does each term in this BRDF represent?
   2. Why is the denominator $4 \cdot (\omega_i \cdot \mathbf{n}) \cdot (\omega_o \cdot \mathbf{n})$?
   3. How does the roughness parameter affect the appearance of the surface?

   Solution:

   1. The terms in the Cook-Torrance BRDF represent:
      - $F(\omega_i, \mathbf{h})$: The Fresnel term, which describes the fraction of light reflected at the interface between two media. It depends on the incident angle and the material's index of refraction. The Schlick approximation is often used: $F(\omega_i, \mathbf{h}) = F_0 + (1 - F_0)(1 - (\omega_i \cdot \mathbf{h}))^5$, where F_0 is the reflectance at normal incidence.
      - $G(\omega_i, \omega_o, \mathbf{h})$: The geometry term (or shadowing-masking term), which accounts for the self-shadowing of microfacets. It represents the fraction of microfacets that are visible from both the light and the viewer.
      - $D(\mathbf{h})$: The normal distribution function (NDF), which describes the statistical distribution of microfacet normals. Common NDFs include Beckmann, GGX (Trowbridge-Reitz), and Blinn-Phong.
   2. The denominator $4 \cdot (\omega_i \cdot \mathbf{n}) \cdot (\omega_o \cdot \mathbf{n})$ comes from:
      - A factor of $(\omega_i \cdot \mathbf{n}) \cdot (\omega_o \cdot \mathbf{n})$ to normalize the BRDF.
      - A factor of 4 from the Jacobian of the transformation between the half-vector and the reflection vector.

      The full derivation involves relating the differential solid angle of the half-vector $d\omega_h$ to the differential solid angle of the reflection $d\omega_o$, which introduces the factor of $\frac{1}{4(\omega_i \cdot \mathbf{h})}$. Combined with other terms in the microfacet model, this simplifies to the denominator shown.
   3. The roughness parameter primarily affects the normal distribution function $D(\mathbf{h})$:
      - Low roughness values create a narrow distribution of microfacet normals, resulting in sharp, mirror-like reflections.
      - High roughness values create a wide distribution of microfacet normals, resulting in blurry, diffuse-like reflections.

      The roughness also affects the geometry term G, as rougher surfaces tend to have more self-shadowing and masking.

2. **Symbolic Derivation: Smith Geometry Term for GGX Distribution**

The Smith geometry term models the shadowing and masking effects in microfacet BRDFs. For the GGX (Trowbridge-Reitz) distribution, the Smith geometry term can be derived from the distribution function.

1. The GGX normal distribution function is given by:

$$D_{GGX}(\mathbf{h}) = \frac{\alpha^2}{\pi \cdot ((\mathbf{n} \cdot \mathbf{h})^2 \cdot (\alpha^2 - 1) + 1)^2}$$

   where α is the roughness parameter. The Smith geometry term for a single direction ω is:

$$G_1(\omega) = \frac{2}{1 + \sqrt{1 + \alpha^2 \cdot \tan^2 \theta_\omega}}$$

   where θ_ω is the angle between ω and the normal $\mathbf{n}$.

   Derive this expression for $G_1(\omega)$ starting from the GGX distribution function. Hint: The Smith G_1 term can be derived as:

$$G_1(\omega) = \frac{1}{\Lambda(\omega) + 1}$$

   where $\Lambda(\omega)$ is the average slope of visible microfacets in direction ω.

2. The full geometry term is often approximated as:

$$G(\omega_i, \omega_o, \mathbf{h}) = G_1(\omega_i) \cdot G_1(\omega_o)$$

   Explain why this is an approximation and not exact.

3. Rewrite $G_1(\omega)$ in terms of $(\mathbf{n} \cdot \omega)$ instead of $\tan \theta_\omega$.

Solution:

1. To derive the Smith G_1 term for the GGX distribution, we start with the definition of $\Lambda(\omega)$, which is the average slope of visible microfacets in direction ω.

   For the GGX distribution, $\Lambda(\omega)$ can be calculated as:

$$\Lambda(\omega) = \frac{-1 + \sqrt{1 + \alpha^2 \cdot \tan^2 \theta_\omega}}{2}$$

   This can be derived from the GGX distribution by integrating over all possible slopes that would cause shadowing in direction ω.

   Given that $G_1(\omega) = \frac{1}{\Lambda(\omega)+1}$, we substitute:

$$G_1(\omega) = \frac{1}{\frac{-1+\sqrt{1+\alpha^2 \cdot \tan^2 \theta_\omega}}{2} + 1}$$

   Simplifying:

$$G_1(\omega) = \frac{2}{-1 + \sqrt{1 + \alpha^2 \cdot \tan^2 \theta_\omega} + 2}$$

$$G_1(\omega) = \frac{2}{1 + \sqrt{1 + \alpha^2 \cdot \tan^2 \theta_\omega}}$$

2. The approximation $G(\omega_i, \omega_o, \mathbf{h}) = G_1(\omega_i) \cdot G_1(\omega_o)$ assumes that the shadowing and masking events are statistically independent, which is not strictly true in a real microfacet surface.

   In reality, the visibility of a microfacet from both the incident and outgoing directions is correlated. If a microfacet is shadowed from the incident direction, it's more likely to also be masked from the outgoing direction, especially when the directions are similar.

   The Smith model treats these as independent probabilities, which simplifies the mathematics but introduces some error. Despite this approximation, the Smith model works well in practice and is widely used due to its simplicity and reasonable accuracy.

3. To rewrite $G_1(\omega)$ in terms of $(\mathbf{n} \cdot \omega)$, we use the relationship:

$$\tan\theta_\omega = \frac{\sin\theta_\omega}{\cos\theta_\omega} = \frac{\sqrt{1-\cos^2\theta_\omega}}{\cos\theta_\omega} = \frac{\sqrt{1-(\mathbf{n}\cdot\omega)^2}}{(\mathbf{n}\cdot\omega)}$$

   Substituting into the G_1 formula:

$$G_1(\omega) = \frac{2}{1+\sqrt{1+\alpha^2 \cdot \frac{1-(\mathbf{n}\cdot\omega)^2}{(\mathbf{n}\cdot\omega)^2}}}$$

   Simplifying inside the square root:

$$\begin{aligned} G_1(\omega) &= \frac{2}{1+\sqrt{\frac{(\mathbf{n}\cdot\omega)^2+\alpha^2\cdot(1-(\mathbf{n}\cdot\omega)^2)}{(\mathbf{n}\cdot\omega)^2}}} \\ &= \frac{2}{1+\sqrt{\frac{(\mathbf{n}\cdot\omega)^2+\alpha^2-\alpha^2\cdot(\mathbf{n}\cdot\omega)^2}{(\mathbf{n}\cdot\omega)^2}}} \\ &= \frac{2}{1+\sqrt{\frac{(\mathbf{n}\cdot\omega)^2\cdot(1-\alpha^2)+\alpha^2}{(\mathbf{n}\cdot\omega)^2}}} \\ &= \frac{2}{1+\sqrt{\frac{\alpha^2}{(\mathbf{n}\cdot\omega)^2}+(1-\alpha^2)}} \end{aligned}$$

   Therefore:

$$G_1(\omega) = \frac{2}{1+\sqrt{1+\frac{\alpha^2\cdot(1-(\mathbf{n}\cdot\omega)^2)}{(\mathbf{n}\cdot\omega)^2}}} = \frac{2(\mathbf{n}\cdot\omega)}{(\mathbf{n}\cdot\omega)+\sqrt{(\mathbf{n}\cdot\omega)^2+\alpha^2(1-(\mathbf{n}\cdot\omega)^2)}}$$

3. **Walkthrough: Cook-Torrance BRDF Calculation**

   This problem will walk you through a full shading calculation for a single point on a surface using the Cook-Torrance microfacet model.

   Scenario:

   - A point on a surface has normal $\mathbf{n} = (0, 0, 1)$.
   - The viewer is in direction $\omega_o = (\frac{\sqrt{3}}{2}, 0, \frac{1}{2})$.

- A single point light provides radiance L_i from direction $\omega_i = (-\frac{\sqrt{3}}{2}, 0, \frac{1}{2})$.
- Assume for this setup, the material properties evaluate to $D = 1/\pi$, $G = 1$, and $F = F_0$.

Your task is to calculate the final outgoing radiance, $L_o(\omega_o)$.

1. Calculate the halfway vector, $\mathbf{h}$.
2. Calculate the dot products $(\mathbf{n} \cdot \omega_i)$ and $(\mathbf{n} \cdot \omega_o)$.
3. Assemble the full Cook-Torrance BRDF value, $f_r = \frac{D \cdot G \cdot F}{4(\mathbf{n} \cdot \omega_i)(\mathbf{n} \cdot \omega_o)}$.
4. Calculate the final outgoing radiance L_o using the reflection equation for a single point light: $L_o = f_r \cdot L_i \cdot (\mathbf{n} \cdot \omega_i)$.

Solution:

1. **Halfway Vector h:**

$$\mathbf{h} = \frac{\omega_i + \omega_o}{||\omega_i + \omega_o||} = \frac{(-\frac{\sqrt{3}}{2}, 0, \frac{1}{2}) + (\frac{\sqrt{3}}{2}, 0, \frac{1}{2})}{|| \dots ||} = \frac{(0, 0, 1)}{||(0, 0, 1)||} = (0, 0, 1)$$

The halfway vector is aligned with the normal, representing a perfect specular reflection condition.

2. **Dot Products:**

$$(\mathbf{n} \cdot \omega_i) = (0, 0, 1) \cdot (-\frac{\sqrt{3}}{2}, 0, \frac{1}{2}) = \frac{1}{2}$$

$$(\mathbf{n} \cdot \omega_o) = (0, 0, 1) \cdot (\frac{\sqrt{3}}{2}, 0, \frac{1}{2}) = \frac{1}{2}$$

3. **Assemble the BRDF f_r:**

$$f_r = \frac{D \cdot G \cdot F}{4(\mathbf{n} \cdot \omega_i)(\mathbf{n} \cdot \omega_o)} = \frac{(1/\pi) \cdot 1 \cdot F_0}{4 \cdot (1/2) \cdot (1/2)} = \frac{F_0/\pi}{1} = \frac{F_0}{\pi}$$

4. **Final Outgoing Radiance L_o:**

$$L_o = f_r \cdot L_i \cdot (\mathbf{n} \cdot \omega_i) = \frac{F_0}{\pi} \cdot L_i \cdot \frac{1}{2} = \frac{F_0 L_i}{2\pi}$$

4. **Microfacet BRDF Components Analysis**

Consider the Cook-Torrance microfacet BRDF model:

$$f(\mathbf{i}, \mathbf{o}) = \frac{\mathbf{F}(\mathbf{i}, \mathbf{h})\mathbf{G}(\mathbf{i}, \mathbf{o}, \mathbf{h})\mathbf{D}(\mathbf{h})}{4(\mathbf{n} \cdot \mathbf{i})(\mathbf{n} \cdot \mathbf{o})}$$

For a surface with normal $\mathbf{n} = (0, 0, 1)$, analyze how each component affects the BRDF as the viewing angle changes. Assume the light direction is fixed at $\mathbf{i} = (0, 0, 1)$ (directly overhead) and the viewing direction $\mathbf{o} = (\sin\theta, 0, \cos\theta)$ varies with θ.

1. Express the halfway vector $\mathbf{h}$ in terms of θ.
2. If we use the Schlick approximation for the Fresnel term with $F_0 = 0.04$ (typical for dielectrics), calculate $F(\mathbf{i}, \mathbf{h})$ for $\theta = 0$, $\theta = 60$, and $\theta = 85$.

3. For a microfacet distribution $D(\mathbf{h}) = \frac{1}{\pi\alpha^2 \cos^4\theta_h} e^{-\tan^2\theta_h/\alpha^2}$ with roughness $\alpha = 0.1$, calculate $D(\mathbf{h})$ for the same three viewing angles. Here, θ_h is the angle between $\mathbf{n}$ and $\mathbf{h}$.
4. Assuming $G = 1$ for simplicity, calculate the complete BRDF value for each of the three viewing angles.

Solution:

1. When $\mathbf{i} = (0, 0, 1)$ and $\mathbf{o} = (\sin\theta, 0, \cos\theta)$:

$$\mathbf{h} = \frac{\mathbf{i}+\mathbf{o}}{||\mathbf{i}+\mathbf{o}||} = \frac{(0,0,1)+(\sin\theta,0,\cos\theta)}{||(0,0,1)+(\sin\theta,0,\cos\theta)||} = \frac{(\sin\theta,0,1+\cos\theta)}{\sqrt{\sin^2\theta+(1+\cos\theta)^2}}$$

We can simplify this. First, note that $\sin^2\theta+(1+\cos\theta)^2 = \sin^2\theta+1+2\cos\theta+\cos^2\theta = 2+2\cos\theta$. So:

$$\mathbf{h} = \frac{(\sin\theta,0,1+\cos\theta)}{\sqrt{2+2\cos\theta}} = \frac{(\sin\theta,0,1+\cos\theta)}{2\cos(\theta/2)}$$

This gives us $\mathbf{h} = (\frac{\sin\theta}{2\cos(\theta/2)}, 0, \frac{1+\cos\theta}{2\cos(\theta/2)})$

2. For the Schlick approximation: $F(\mathbf{i},\mathbf{h}) = F_0 + (1-F_0)(1-(\mathbf{i}\cdot\mathbf{h}))^5$

First, we need $(\mathbf{i}\cdot\mathbf{h})$ for each angle:

For $\theta = 0$: $\mathbf{o} = (0,0,1)$, so $\mathbf{h} = (0,0,1)$, and $(\mathbf{i}\cdot\mathbf{h}) = 1$

$$F = 0.04 + (1-0.04)(1-1)^5 = 0.04 + 0 = 0.04$$

For $\theta = 60$: $\mathbf{o} = (0.866, 0, 0.5)$, $\mathbf{h} \approx (0.5, 0, 0.866)$, and $(\mathbf{i}\cdot\mathbf{h}) = 0.866$

$$F = 0.04 + (1-0.04)(1-0.866)^5 = 0.04 + 0.96\cdot(0.134)^5 \approx 0.04 + 0.00005 = 0.04005$$

For $\theta = 85$: $\mathbf{o} = (0.996, 0, 0.087)$, $\mathbf{h} \approx (0.707, 0, 0.707)$, and $(\mathbf{i}\cdot\mathbf{h}) = 0.707$

$$F = 0.04 + (1-0.04)(1-0.707)^5 = 0.04 + 0.96\cdot(0.293)^5 \approx 0.04 + 0.00076 = 0.04076$$

3. For the microfacet distribution:

For $\theta = 0$: $\mathbf{h} = (0,0,1)$, so $\theta_h = 0$, $\cos\theta_h = 1$, $\tan\theta_h = 0$

$$D = \frac{1}{\pi\cdot 0.1^2\cdot 1} e^{-0/0.1^2} = \frac{100}{\pi} \approx 31.83$$

For $\theta = 60$: $\mathbf{h} \approx (0.5, 0, 0.866)$, so $\theta_h \approx 30$, $\cos\theta_h = 0.866$, $\tan\theta_h = 0.577$

$$D = \frac{1}{\pi\cdot 0.1^2\cdot 0.866^4} e^{-0.577^2/0.1^2} \approx \frac{100}{\pi\cdot 0.563}\cdot e^{-33.3} \approx \frac{177.6}{\pi}\cdot 3.7\times 10^{-15} \approx 0$$

For $\theta = 85$: $\mathbf{h} \approx (0.707, 0, 0.707)$, so $\theta_h \approx 45$, $\cos\theta_h = 0.707$, $\tan\theta_h = 1$

$$D = \frac{1}{\pi\cdot 0.1^2\cdot 0.707^4} e^{-1/0.1^2} \approx \frac{100}{\pi\cdot 0.25}\cdot e^{-100} \approx \frac{400}{\pi}\cdot 3.7\times 10^{-44} \approx 0$$

4. For the complete BRDF:

   For $\theta = 0$: $(\mathbf{n} \cdot \mathbf{i}) = 1$, $(\mathbf{n} \cdot \mathbf{o}) = 1$

$$f = \frac{0.04 \cdot 1 \cdot 31.83}{4 \cdot 1 \cdot 1} = \frac{1.27}{4} = 0.318$$

   For $\theta = 60$: $(\mathbf{n} \cdot \mathbf{i}) = 1$, $(\mathbf{n} \cdot \mathbf{o}) = 0.5$

$$f = \frac{0.04005 \cdot 1 \cdot 0}{4 \cdot 1 \cdot 0.5} = 0$$

   For $\theta = 85$: $(\mathbf{n} \cdot \mathbf{i}) = 1$, $(\mathbf{n} \cdot \mathbf{o}) = 0.087$

$$f = \frac{0.04076 \cdot 1 \cdot 0}{4 \cdot 1 \cdot 0.087} = 0$$

   Note: The BRDF values at the larger angles are essentially zero because the microfacet distribution is extremely peaked with the small roughness value of $\alpha = 0.1$. This means that only microfacets very closely aligned with the halfway vector contribute to the reflection, which is characteristic of very smooth, mirror-like surfaces.

4 Isotropic and Anisotropic Materials

1. **Isotropy vs. Anisotropy**

   Anisotropic BRDFs are used for materials with a directional structure, such as brushed metal. They depend on tangent and bitangent vectors, $\mathbf{t}$ and $\mathbf{b}$, in addition to the normal $\mathbf{n}$.

   Consider an anisotropic microfacet model where the roughness depends on direction. Let α_x be the roughness along the tangent direction and α_y be the roughness along the bitangent direction. A common anisotropic NDF is:

   $$D(\mathbf{h}) = \frac{1}{\pi\alpha_x\alpha_y}\frac{1}{\left((\frac{(\mathbf{h}\cdot\mathbf{t})}{\alpha_x})^2 + (\frac{(\mathbf{h}\cdot\mathbf{b})}{\alpha_y})^2 + (\mathbf{h}\cdot\mathbf{n})^2\right)^2}$$

   If the halfway vector is $\mathbf{h} = (0.5, 0.5, \frac{\sqrt{2}}{2})$, the normal is $\mathbf{n} = (0, 0, 1)$, the tangent is $\mathbf{t} = (1, 0, 0)$, and the bitangent is $\mathbf{b} = (0, 1, 0)$, calculate the value of the NDF for:

   1. An isotropic material with $\alpha_x = 0.5, \alpha_y = 0.5$.
   2. An anisotropic material with $\alpha_x = 0.1, \alpha_y = 1.0$.

   Solution: First, calculate the dot products: $\mathbf{h}\cdot\mathbf{t} = 0.5$, $\mathbf{h}\cdot\mathbf{b} = 0.5$, $\mathbf{h}\cdot\mathbf{n} = \frac{\sqrt{2}}{2} \approx 0.707$.

   1. For the isotropic material:

   $$D = \frac{1}{\pi(0.5)^2}\frac{1}{\left((\frac{0.5}{0.5})^2 + (\frac{0.5}{0.5})^2 + (0.707)^2\right)^2} = \frac{1}{0.25\pi}\frac{1}{(1+1+0.5)^2} = \frac{4}{\pi}\frac{1}{2.5^2} \approx 0.204$$

   2. For the anisotropic material:

   $$D = \frac{1}{\pi(0.1)(1.0)}\frac{1}{\left((\frac{0.5}{0.1})^2 + (\frac{0.5}{1.0})^2 + (0.707)^2\right)^2} = \frac{1}{0.1\pi}\frac{1}{(5^2+0.5^2+0.5)^2} = \frac{10}{\pi}\frac{1}{(25+0.25+0.5)^2} \approx 0.004$$

   The NDF value is much lower for the anisotropic case because the halfway vector is not aligned with the low-roughness direction.

2. **The Phong BRDF**

   A classic (non-physically-based) Phong BRDF can be written as:

   $$f_r(\omega_i, \omega_o) = k_d\frac{\rho}{\pi} + k_s\frac{\alpha+2}{2\pi}(\mathbf{R}\cdot\omega_o)^\alpha$$

   where $\mathbf{R}$ is the perfect mirror reflection of ω_i. Show that this BRDF violates Helmholtz reciprocity.

   Solution: To check for Helmholtz reciprocity, we must check if $f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i)$. The diffuse term is reciprocal. For the specular term, we must check if $(\mathbf{R}(\omega_i)\cdot\omega_o)^\alpha = (\mathbf{R}(\omega_o)\cdot\omega_i)^\alpha$. In general, these are not equal. The Blinn-Phong model, which uses the half-vector, was developed in part to fix this issue, as the half-vector calculation is symmetric: $\mathbf{h}(\omega_i, \omega_o) = \mathbf{h}(\omega_o, \omega_i)$.

3. **Kajiya-Kay Hair Model**

The Kajiya-Kay model is a classic anisotropic BRDF used for rendering hair and other fibrous materials. Unlike surface-based BRDFs, it models light scattering from thin cylindrical fibers. The model is:

$$f_r(\omega_i, \omega_o) = k_d + k_s \frac{(\mathbf{t} \cdot \omega_i)(\mathbf{t} \cdot \omega_o) + \sin\theta_i \sin\theta_o \cos(\phi_i - \phi_o)}{\sin\theta_i \sin\theta_o} \sin^p(\theta_h)$$

where $\mathbf{t}$ is the tangent direction along the hair fiber, θ_i and θ_o are the angles between the incident/outgoing directions and the tangent, ϕ_i and ϕ_o are the azimuthal angles around the tangent, and θ_h is the half angle.

Consider a hair fiber with tangent direction $\mathbf{t} = (0, 0, 1)$ and the following configuration:

- Incident light direction $\omega_i = (0.5, 0, 0.866)$ (30° from the tangent)
- Viewing direction $\omega_o = (0, 0.5, 0.866)$ (30° from the tangent, 90° azimuthal difference)
- Model parameters: $k_d = 0.1$, $k_s = 0.9$, $p = 80$ (high specular exponent)

1. Calculate the angles θ_i, θ_o, ϕ_i, and ϕ_o.
2. Calculate the half angle θ_h.
3. Compute the value of the Kajiya-Kay BRDF for this configuration.
4. How would the BRDF value change if the viewing direction were at the perfect specular reflection angle? Calculate this value.

Solution:

1. First, let's calculate the angles:

   For $\omega_i = (0.5, 0, 0.866)$:

   $$\cos\theta_i = \mathbf{t} \cdot \omega_i = (0, 0, 1) \cdot (0.5, 0, 0.866) = 0.866$$

   $$\theta_i = \arccos(0.866) = 30$$
   $$\sin\theta_i = \sin(30) = 0.5$$

   Since ω_i is in the xz-plane with positive x, $\phi_i = 0$.

   For $\omega_o = (0, 0.5, 0.866)$:

   $$\cos\theta_o = \mathbf{t} \cdot \omega_o = (0, 0, 1) \cdot (0, 0.5, 0.866) = 0.866$$

   $$\theta_o = \arccos(0.866) = 30$$
   $$\sin\theta_o = \sin(30) = 0.5$$

   Since ω_o is in the yz-plane with positive y, $\phi_o = 90$.

2. To calculate the half angle θ_h, we need the projection of the half vector onto the tangent-normal plane.

   First, compute the half vector:

   $$\mathbf{h} = \frac{\omega_i + \omega_o}{||\omega_i + \omega_o||} = \frac{(0.5, 0.5, 1.732)}{||(0.5, 0.5, 1.732)||}$$

The magnitude of the half vector is:

$$||\omega_i + \omega_o|| = \sqrt{0.5^2 + 0.5^2 + 1.732^2} = \sqrt{0.25 + 0.25 + 3} = \sqrt{3.5} \approx 1.871$$

So the normalized half vector is:

$$\mathbf{h} = \frac{(0.5, 0.5, 1.732)}{1.871} \approx (0.267, 0.267, 0.926)$$

The half angle is:

$$\cos\theta_h = \mathbf{t} \cdot \mathbf{h} = (0, 0, 1) \cdot (0.267, 0.267, 0.926) = 0.926$$

$$\theta_h = \arccos(0.926) \approx 22.3$$

$$\sin\theta_h = \sin(22.3) \approx 0.38$$

3. Now we can compute the BRDF value:

$$f_r(\omega_i, \omega_o) = k_d + k_s \frac{(\mathbf{t} \cdot \omega_i)(\mathbf{t} \cdot \omega_o) + \sin\theta_i \sin\theta_o \cos(\phi_i - \phi_o)}{\sin\theta_i \sin\theta_o} \sin^p(\theta_h)$$

Substituting our values:

$$\begin{aligned} f_r &= 0.1 + 0.9 \frac{(0.866)(0.866) + (0.5)(0.5)\cos(0 - 90)}{\sin(30)\sin(30)} \sin^{80}(22.3) \\ &= 0.1 + 0.9 \frac{0.75 + (0.5)(0.5)(0)}{(0.5)(0.5)} (0.38)^{80} \\ &= 0.1 + 0.9 \frac{0.75}{0.25} (0.38)^{80} \\ &= 0.1 + 0.9 \cdot 3 \cdot (0.38)^{80} \end{aligned}$$

Since $(0.38)^{80}$ is an extremely small number (approximately 1.5×10^{-39}), the specular term is effectively zero:

$$f_r \approx 0.1 + 0.9 \cdot 3 \cdot 0 = 0.1$$

The BRDF value is dominated by the diffuse term because the viewing direction is far from the specular reflection direction.

4. For the perfect specular reflection, the outgoing direction would be in the same plane as the incident direction and tangent, with the same angle from the tangent but on the opposite side. This would be:

$$\omega_o^{specular} = (0.5, 0, 0.866) \text{ (same as } \omega_i\text{)}$$

In this case, $\phi_o = \phi_i = 0$, and the half vector would be exactly aligned with the tangent:

$$\mathbf{h} = \frac{(0.5, 0, 0.866) + (0.5, 0, 0.866)}{||(0.5, 0, 0.866) + (0.5, 0, 0.866)||} = \frac{(1, 0, 1.732)}{||1, 0, 1.732||} = \frac{(1, 0, 1.732)}{\sqrt{1^2 + 1.732^2}} = \frac{(1, 0, 1.732)}{2} = (0.5, 0, 0.866)$$

This gives us $\theta_h = 30$ and $\sin \theta_h = 0.5$.

Computing the BRDF:

$$\begin{aligned} f_r &= 0.1 + 0.9 \frac{(0.866)(0.866) + (0.5)(0.5)\cos(0-0)}{\sin(30)\sin(30)} \sin^{80}(30) \\ &= 0.1 + 0.9 \frac{0.75 + (0.5)(0.5)(1)}{(0.5)(0.5)} (0.5)^{80} \\ &= 0.1 + 0.9 \frac{0.75 + 0.25}{0.25} (0.5)^{80} \\ &= 0.1 + 0.9 \cdot 4 \cdot (0.5)^{80} \end{aligned}$$

Again, $(0.5)^{80}$ is extremely small (approximately 8.3×10^{-25}), so:

$$f_r \approx 0.1 + 0.9 \cdot 4 \cdot 0 = 0.1$$

Even at the perfect specular angle, the high exponent $p = 80$ makes the specular contribution negligible for this configuration. This demonstrates how the Kajiya-Kay model can create very sharp specular highlights that are only visible at very specific viewing angles.

Note: In practice, for hair rendering, lower exponents (e.g., $p = 10$ to $p = 40$) are often used to create more visible specular highlights.

5 Importance Sampling

1. **Monte Carlo Estimator** The Monte Carlo estimator for an integral $I = \int f(x)dx$ is $\frac{1}{N}\sum_{k=1}^{N}\frac{f(x_k)}{p(x_k)}$. Explain the intuition behind this formula. Why is it important to choose a PDF $p(x)$ that is similar to the function $f(x)$ being integrated?

   Solution: The estimator computes the average value of $\frac{f(x)}{p(x)}$. The division by the PDF $p(x)$ corrects for the fact that we are sampling from a non-uniform distribution. If we sample in a region where $p(x)$ is high, we divide by a larger number to down-weight the sample's contribution, and vice-versa. Choosing a PDF $p(x)$ that is proportional to $f(x)$ is the core idea of importance sampling. This strategy concentrates samples in the regions where the function's value is highest, which are the regions that contribute most to the integral. This dramatically reduces the variance of the estimator, allowing it to converge with far fewer samples compared to, for example, uniform sampling.

2. **Sampling the NDF** When importance sampling a microfacet BRDF, a common strategy is to sample the NDF. This produces a microfacet normal (or halfway vector) $\mathbf{h}$. However, the reflection equation integrates over incoming light directions ω_i. How do we convert the PDF for sampling $\mathbf{h}$ to a PDF for sampling ω_i?

   Solution: This requires a change of variables from the microfacet normal's solid angle $d\omega_h$ to the incoming light's solid angle $d\omega_i$. The relationship is given by the Jacobian of the transformation from ω_i to $\mathbf{h}$:

   $$p(\omega_i) = p(\mathbf{h})\left|\frac{d\omega_h}{d\omega_i}\right|$$

   For reflection, ω_i is determined by ω_o and $\mathbf{h}$. The Jacobian for this relationship is:

   $$\left|\frac{d\omega_h}{d\omega_i}\right| = \frac{1}{4(\omega_o \cdot \mathbf{h})}$$

   Therefore, if we sample the NDF such that $p(\mathbf{h}) = D(\mathbf{h})\cos\theta_h$, the corresponding PDF for the light direction is:

   $$p(\omega_i) = \frac{D(\mathbf{h})\cos\theta_h}{4(\omega_o \cdot \mathbf{h})}$$

3. **Coding: Path Tracing** Complete the C++ code to implement one recursive step of a path tracer.

```
Vector3D at_least_one_bounce_radiance(const Ray& r, const Intersection& isect) {
    // Given: r, isect, bvh
    // Task: Sample a new direction, trace a ray, and return the estimated radiance
        .
    const BSDF* bsdf = isect.bsdf;
    Vector3D w_out = isect.to_local(-r.d);

    Vector3D w_in;
    double pdf;
    Vector3D f = bsdf->sample_f(w_out, &w_in, &pdf);

    if (pdf == 0) return Vector3D();

    Vector3D w_in_world = isect.to_world(w_in);
```

```
    Ray bounce_ray(isect.hit_p + EPS_F * w_in_world, w_in_world);
    bounce_ray.depth = r.depth - 1;

    // --- YOUR CODE HERE ---


}
```

Solution:

```
    // --- YOUR CODE HERE ---
    if (bounce_ray.depth == 0) return Vector3D();

    Intersection next_isect;
    if (bvh->intersect(bounce_ray, &next_isect)) {
        Vector3D L_i = at_least_one_bounce_radiance(bounce_ray, next_isect);
        double cos_theta = std::abs(w_in.z);
        return (f * L_i * cos_theta) / pdf;
    } else {
        return Vector3D();
    }
    // --- END YOUR CODE ---
```

4. **C++ Implementation: Multiple Importance Sampling**

In path tracing, we often need to sample from multiple distributions to reduce variance. Multiple Importance Sampling (MIS) is a technique to combine samples from different strategies. Implement a C++ function that uses the balance heuristic for MIS to combine BRDF sampling and light sampling.

```
struct Scene {
    // Returns the direct illumination from a point light source
    Vector3f sampleDirectLight(const Intersection& isect, const Vector3f& wo) const
        ;

    // Returns a random point on a light source and its PDF
    std::pair<Vector3f, float> sampleLightPoint() const;

    // Returns the emission from a point in a given direction
    Vector3f emission(const Vector3f& point, const Vector3f& direction) const;

    // Tests if a ray intersects any object in the scene
    bool intersect(const Ray& ray, Intersection* isect) const;
};

struct BSDF {
    // Evaluates the BSDF for the given directions
    Vector3f evaluate(const Vector3f& wi, const Vector3f& wo) const;

    // Samples a direction according to the BSDF and returns the PDF
    std::pair<Vector3f, float> sample(const Vector3f& wo, const Point2f& u) const;

    // Returns the PDF of sampling the given direction
    float pdf(const Vector3f& wi, const Vector3f& wo) const;
};

// Compute direct illumination using multiple importance sampling
Vector3f computeDirectIllumination(
```

```
    const Scene& scene,
    const Intersection& isect,
    const Vector3f& wo,
    const BSDF& bsdf,
    const Point2f& u1,  // Random numbers for BSDF sampling
    const Point2f& u2   // Random numbers for light sampling
) {
    // TODO: Implement multiple importance sampling with the balance heuristic
    // 1. Sample from both the BSDF and the light
    // 2. Evaluate the contribution from each sample
    // 3. Combine using the balance heuristic

}
```

Solution:

```
Vector3f computeDirectIllumination(
    const Scene& scene,
    const Intersection& isect,
    const Vector3f& wo,
    const BSDF& bsdf,
    const Point2f& u1,  // Random numbers for BSDF sampling
    const Point2f& u2   // Random numbers for light sampling
) {
    Vector3f result(0.0f);

    // Part 1: Sample the BSDF
    auto [wi_bsdf, pdf_bsdf] = bsdf.sample(wo, u1);

    // Check if the sampled direction is valid
    if (pdf_bsdf > 0.0f) {
        // Create a ray in the sampled direction
        Ray bsdf_ray(isect.position, wi_bsdf);
        Intersection light_isect;

        // Check if the ray hits a light source
        if (scene.intersect(bsdf_ray, &light_isect) && light_isect.isEmissive) {
            // Evaluate the emission from the light
            Vector3f L_i = scene.emission(light_isect.position, -wi_bsdf);

            // Compute the PDF of sampling this point from the light sampling
                strategy
            float pdf_light = light_isect.lightPdf;  // Assume the intersection
                stores this

            // Evaluate the BSDF
            Vector3f f = bsdf.evaluate(wi_bsdf, wo);

            // Compute the MIS weight using the balance heuristic
            float mis_weight = pdf_bsdf / (pdf_bsdf + pdf_light);

            // Compute the contribution
            float cos_theta = std::abs(dot(wi_bsdf, isect.normal));
            result += mis_weight * f * L_i * cos_theta / pdf_bsdf;
        }
    }
```

```
    // Part 2: Sample the light
    auto [light_point, pdf_light] = scene.sampleLightPoint();

    // Compute the direction to the light
    Vector3f wi_light = normalize(light_point - isect.position);
    float dist_squared = lengthSquared(light_point - isect.position);

    // Check if the light is visible
    Ray light_ray(isect.position, wi_light);
    Intersection test_isect;
    bool visible = scene.intersect(light_ray, &test_isect) &&
                  (lengthSquared(test_isect.position - isect.position) -
                      dist_squared) < 1e-4f;

    if (visible && pdf_light > 0.0f) {
        // Evaluate the BSDF
        Vector3f f = bsdf.evaluate(wi_light, wo);

        // Compute the PDF of sampling this direction from the BSDF
        float pdf_bsdf = bsdf.pdf(wi_light, wo);

        // Compute the MIS weight using the balance heuristic
        float mis_weight = pdf_light / (pdf_bsdf + pdf_light);

        // Compute the contribution
        float cos_theta = std::abs(dot(wi_light, isect.normal));
        Vector3f L_i = scene.emission(light_point, -wi_light);

        // The geometric term includes distance attenuation and the cosine at
            the light
        float G = cos_theta / dist_squared;

        result += mis_weight * f * L_i * G / pdf_light;
    }

    return result;
}
```

This implementation uses multiple importance sampling with the balance heuristic to combine two sampling strategies:

1. BSDF sampling: We sample a direction according to the BSDF and check if it hits a light source.
2. Light sampling: We sample a point on a light source and check if it's visible from the intersection point.

The balance heuristic weights each sample by its PDF divided by the sum of PDFs from all strategies. This effectively gives more weight to samples that are more likely to be generated by their respective strategies, reducing variance in the final estimate.

The implementation handles edge cases like zero PDFs and visibility checks to ensure robustness. In a real renderer, you might also need to handle special cases like delta distributions (perfect mirrors, glass) where MIS might not be applicable.

6 Participating Media

1. **Volume Rendering Equation** How does the volume rendering equation differ from the surface rendering equation? What do the absorption and scattering coefficients (σ_a, σ_s) and the phase function (p) represent?

   Solution: The volume rendering equation accounts for the continuous interaction of light within a volume, rather than just at a surface. It includes terms for emission from the medium itself and the extinction of light due to absorption and out-scattering.

   - σ_a: The absorption coefficient, representing the rate at which light is absorbed by the medium per unit distance.
   - σ_s: The scattering coefficient, representing the rate at which light is scattered into other directions per unit distance. The extinction coefficient is $\sigma_t = \sigma_a + \sigma_s$.
   - $p(\omega, \omega')$: The phase function, which describes the probability that light traveling in direction ω is scattered into direction ω'. It is analogous to the BRDF for surfaces.

2. **Subsurface Scattering** What is subsurface scattering, and why can it not be modeled by a standard BRDF? What is a BSSRDF?

   Solution: Subsurface scattering is a phenomenon in translucent materials where light penetrates the surface, scatters multiple times within the material, and exits at a different point. A standard BRDF cannot model this because it assumes that light enters and exits at the same point.

   A **BSSRDF (Bidirectional Surface Scattering Reflectance Distribution Function)** is a more general 8D function that accounts for this spatial transport of light. It relates the outgoing radiance at one point to the incoming flux at another point on the surface.

3. **C++ Implementation: Volumetric Path Tracing**

   Implement a C++ function for delta tracking (also known as Woodcock tracking), a technique used to efficiently sample free-flight distances in heterogeneous participating media. The function should sample a distance to the next scattering event and return both the distance and whether absorption or scattering occurred.

```
struct VolumeProperties {
    // Returns the absorption coefficient at position p
    float sigma_a(const Vector3f& p) const;

    // Returns the scattering coefficient at position p
    float sigma_s(const Vector3f& p) const;

    // Returns the maximum extinction coefficient in the volume
    float sigma_t_max() const;

    // Returns the phase function value for the given directions
    float phase(const Vector3f& wi, const Vector3f& wo) const;

    // Samples a direction from the phase function given an incident direction
    Vector3f sample_phase(const Vector3f& wi, const Point2f& u) const;
};
```

```
// Sample a distance in the volume using delta tracking
// Returns:
// - distance to the next event
// - boolean indicating if it was a scattering event (true) or absorption (false)
std::pair<float, bool> sample_distance(
    const Ray& ray,
    const VolumeProperties& volume,
    float max_distance,
    const Point2f& u  // Random numbers in [0,1)^2
) {
    // TODO: Implement delta tracking
    // 1. Sample a tentative distance based on sigma_t_max
    // 2. Accept/reject based on the actual medium properties at that point
    // 3. Return the sampled distance and event type

}
```

Solution:

```
std::pair<float, bool> sample_distance(
    const Ray& ray,
    const VolumeProperties& volume,
    float max_distance,
    const Point2f& u  // Random numbers in [0,1)^2
) {
    // Maximum extinction coefficient in the volume
    const float sigma_t_max = volume.sigma_t_max();

    // Current distance along the ray
    float t = 0.0f;

    // Delta tracking algorithm
    while (true) {
        // Sample free-flight distance based on sigma_t_max
        float dt = -std::log(1.0f - u[0]) / sigma_t_max;
        t += dt;

        // Check if we've exceeded the maximum distance
        if (t > max_distance)
            return {max_distance, false};

        // Compute the current position along the ray
        Vector3f p = ray.origin + t * ray.direction;

        // Get the actual extinction coefficients at this position
        float sigma_a = volume.sigma_a(p);
        float sigma_s = volume.sigma_s(p);
        float sigma_t = sigma_a + sigma_s;

        // Probabilistically accept this interaction
        float acceptance_prob = sigma_t / sigma_t_max;
        if (u[1] < acceptance_prob) {
            // Determine if this is a scattering or absorption event
            bool is_scattering = (u[1] * sigma_t_max < sigma_s);
            return {t, is_scattering};
        }
```

```
        // If rejected, continue the random walk
    }
}
```

This implementation uses the delta tracking algorithm to efficiently sample distances in heterogeneous media:

1. We sample a tentative distance assuming the maximum extinction coefficient throughout the volume. 2. At each tentative interaction point, we probabilistically accept or reject the interaction based on the ratio of the actual extinction coefficient to the maximum. 3. If an interaction is accepted, we determine whether it's a scattering or absorption event based on the relative proportions of σ_s and σ_a.

The algorithm effectively samples from the correct distribution without having to perform expensive ray marching or numerical integration of the transmittance function.