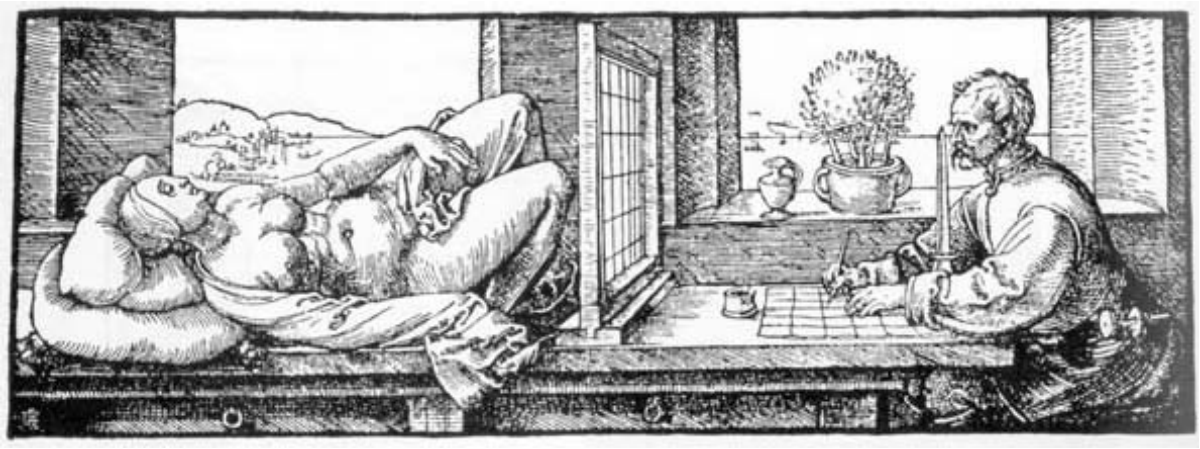
Global Illumination Compendium

September 29, 2003

Philip Dutré
phil@cs.kuleuven.ac.be
Computer Graphics, Department of Computer Science
Katholieke Universiteit Leuven

http://www.cs.kuleuven.ac.be/~phil/GI/



This collection of formulas and equations is supposed to be useful for anyone who is active in the field of global illumination in computer graphics. I started this document as a helpful tool to my own research, since I was growing tired of having to look up equations and formulas in various books and papers. As a consequence, many concepts which are 'trivial' to me are not in this ***Global Illumination Compendium***, unless someone specifically asked for them. Therefore, any further input and suggestions for more useful content are strongly appreciated.

If possible, adequate references are given to look up some of the equations in more detail, or to look up the derivations. Also, an attempt has been made to mention the paper in which a particular idea or equation has been described first. However, over time, many ideas have become 'common knowledge' or have been modified to such extent that they no longer resemble the original formulation. In these cases, references are not given. As a rule of thumb, I include a reference if it really points to useful extra knowledge about the concept being described.

In a document like this, there is always a fair chance of errors. Please report any errors, such that future versions have the correct equations and formulas.

Thanks to the following people for providing me with suggestions and spotting errors: **Neeharika Adabala, Martin Blais, Michael Chock, Chas Ehrlich, Piero Foscari, Neil Gatenby, Simon Green, Eric Haines, Paul Heckbert, Vladimir Koylazov, Vincent Ma, Ioannis (John) Pantazopoulos, Fabio Pellacini, Robert Porschka, Mahesh Ramasubramanian, Cyril Soler, Greg Ward, Steve Westin, Andrew Willmott**.

This document produced at **Program of Computer Graphics, Cornell University** (1999-2001) and **Department of Computer Science, Katholieke Universiteit Leuven** (2001-2003)

Table of Contents

I. General Mathematics

(1) Dirac-impulse (δ-function)

$$\delta(x) = 0 \quad \text{if} \quad x \neq 0$$

$$\int_D \delta(a-x) f(x)\,dx = f(a) \quad \text{if} \quad a \in D$$

Notation: $\delta(a-x) = \delta_a(x)$

(2) Kronecker δ

$$\delta_{ij} = 1 \quad \text{if} \quad i = j$$

$$\delta_{ij} = 0 \quad \text{if} \quad i \neq j$$

II. Probability

(3) Probability density function (pdf)

Constraints: $\int_D p(x)dx = 1 \qquad \forall x: \; p(x) \geq 0$

Probability that a random variable y belongs to interval $[a, b]$: $Prob[a \leq y \leq b] = \int_a^b p(x)dx$

(4) Probability distribution function (a.k.a cumulative distribution function or cdf)

$P(x)$ is the probability that a random variable y, generated using $p(y)$, has a value lower than or equal than x.

$$P(x) = \int_{-\infty}^{x} p(x)dx$$

$$0 = P(-\infty) \leq P(x) \leq P(\infty) = 1$$

(5) Expected value of a random variable *x* with pdf *p*(*x*)

$$E[x] = \int_{-\infty}^{\infty} xp(x)dx \qquad E[f(x)] = \int_{-\infty}^{\infty} f(x)p(x)dx$$

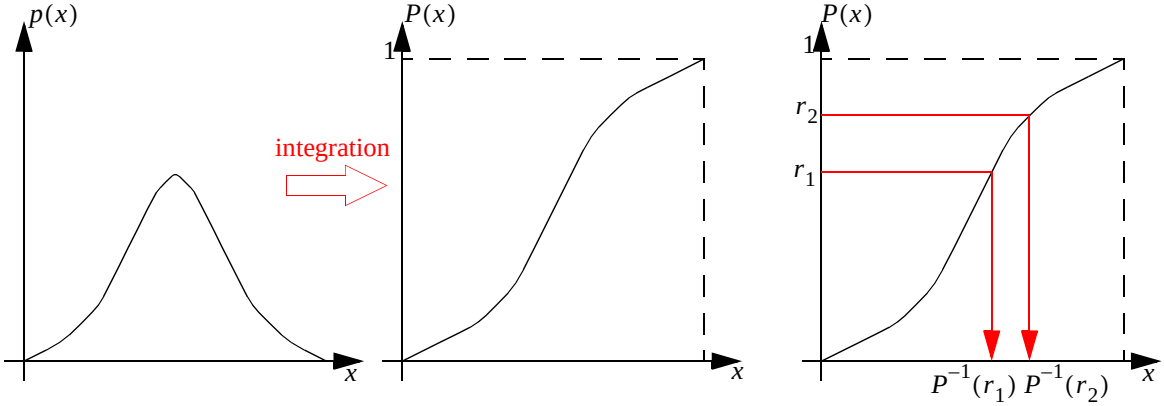
(6) Variance of a random variable *x* with pdf *p*(*x*)

$$\sigma^2[f(x)] = E[(f(x) - E[f(x)])^2]$$
$$\sigma^2[f(x)] = E[f(x)^2] - E[f(x)]^2$$

(7) Generate random variable with given density $p(x)$, using inverse cdf

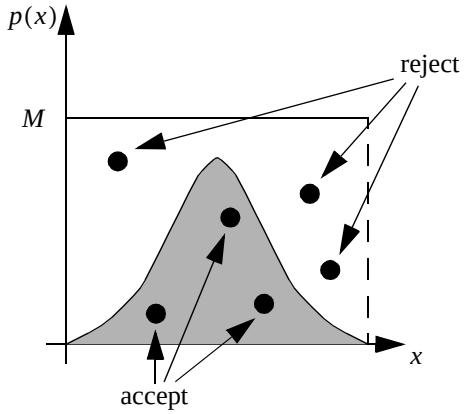
Generate uniform random number r, $0 \leq r \leq 1$ then compute $x = P^{-1}(r)$.

For multidimensional sampling: use marginal or conditional probability distributions, and apply the inverse for each variable seperately.



(8) Generate random variable with given density $p(x)$, using rejection sampling

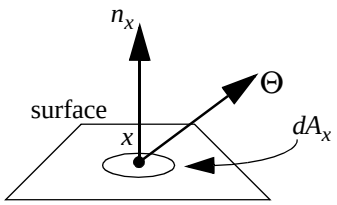
Find M such that $\forall x: \ p(x) \le M$; generate uniform random tuplet $(x', y') \in D \times [0, M]$. If $p(x') \ge y'$ accept sample x'; otherwise reject and try again.



More general:
construct a pdf $q(x)$ such that $\forall x: \ p(x) \le Mq(x)$.
Generate a random sample x' according to $q(x)$.
Generate a uniform random number r, $0 \le r \le 1$.
If $r \le p(x')/Mq(x')$ accept sample x', otherwise reject and try again.

III. Geometry

Notations:



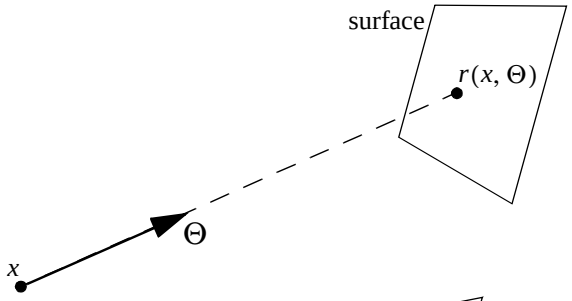
dA_x	Differential surface area at surface point x
n_x	Surface normal at point x
Θ	Direction vector: Θ (usually assumed to be normalized)
$\overline{xy}$	Direction vector from x to y: $\overline{xy} = y - x$
A	Union of all surfaces in the scene, also used for total surface area of all surfaces in the scene.
A_p	Collection of all surface patches in the scene

(9) Ray casting function

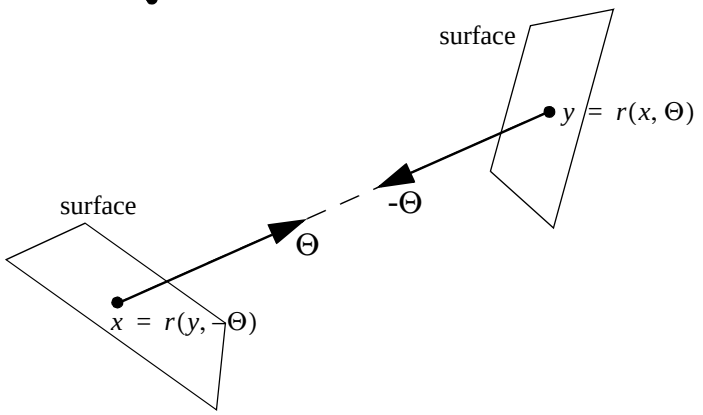
Closest surface point visible from (any) point x in direction Θ:

$$r(x, \Theta) = x + t_{inf} \cdot \Theta$$

$$t_{inf} = inf\{t > 0 : (x + t \cdot \Theta) \in A\}$$



$$\text{if } y = r(x, \Theta) \text{ and } x \in A$$
$$\Rightarrow x = r(y, -\Theta) \text{ or } x = r(r(x, \Theta), -\Theta)$$



Shorthand notation for $r(x, \Theta)$: x^{Θ}

(10) Visibility function

$$\forall x, y \in A: \; V(x, y) = \begin{cases} 1 & \text{if x and y are mutually visible} \\ 0 & \text{if x and y are not mutually visible} \end{cases}$$

The visibility function is often used in various formulations of the rendering equations using area integrals (e.g. 73), or in the equations describing form factors (e.g. 79).

(11) Member function

$$\forall x \in A, j \in A_p\colon\ M(x,j) = \begin{cases} 1 & \text{if x belongs to surface patch j} \\ 0 & \text{if x does not belong to surface patch j} \end{cases}$$

The member function is a handy shorthand notation when one wants to express whether a surface points belongs to a patch. E.g. computing form factors using Monte Carlo integration (87).

(12) Intersection of ray with object

A good overview of various ray-object intersection techniques can be found at:
`http://www.realtimerendering.com/int/`

A. GEOMETRIC TRANSFORMATIONS

Tranformation in 3D graphics are usually represented by 4x4 matrices. The 4th homogeneous coordinate is needed for translations and perspective transforms. A basic overview can be found in any introductory book on computer graphics.

A right-handed coordinate system is assumed.

(13) Translation

Translate a point x by translation vector d: $x' = T(d) \cdot x$

$$T(d) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T^{-1}(d) = \begin{bmatrix} 1 & 0 & 0 & -d_x \\ 0 & 1 & 0 & -d_y \\ 0 & 0 & 1 & -d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(14) Rotation

Rotate point x around the X-axis by angle θ: $x' = R_x(\theta) \cdot x$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R_x^{-1}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotate point x around the Y-axis by angle θ: $x' = R_y(\theta) \cdot x$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R_y^{-1}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotate point x around the Z-axis by angle θ: $x' = R_z(\theta) \cdot x$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R_z^{-1}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation such that basis vectors (x,y,z) become orthonormal vectors (u,v,w):

$$R(uvw) = \begin{bmatrix} u_x & v_x & w_x & 0 \\ u_y & v_y & w_y & 0 \\ u_z & v_z & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R^{-1}(uvw) = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(15) Coordinate transforms

Summary: To transform the coordinates of a point, expressed in coordinate system 1, to coordinates expressed in coordinate system 2; apply the transformation that transforms the axes of coordinate system 2 into the axes coordinate system 1.

$o - xyz$: Canonical coordinate system

$p - uvw$: Coordinate system with origin in p, and orthonormal axes uvw (expressed in OXYZ).

q_{o-xyz} : point q expressed in $o - xyz$

q_{p-uvw} : point q expressed in $p - uvw$

$$q_{p-uvw} = M(p-uvw)q_{o-xyz}$$

$$M(p-uvw) = R^{-1}(uvw)T^{-1}(p) = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 1 & -p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$q_{o-xyz} = M^{-1}(p-uvw)q_{p-uvw}$$

$$M^{-1}(p-uvw) = T(p)R(uvw) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_x & v_x & w_x & 0 \\ u_y & v_y & w_y & 0 \\ u_z & v_z & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u_x & v_x & w_x & p_x \\ u_y & v_y & w_y & p_y \\ u_z & v_z & w_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

B. TRIANGLES

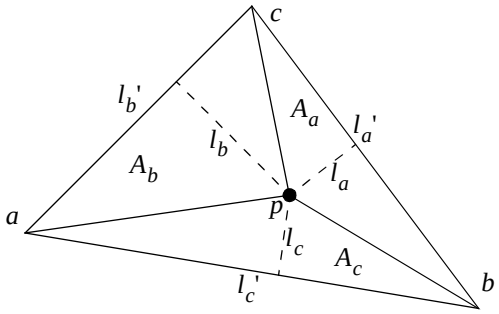
(16) Surface area of a triangle

If a, b, c are the vertices of a triangle:

$$area = \frac{\|a \times b + b \times c + c \times a\|}{2}$$

(17) Barycentric coordinates (a.k.a. trilinear coordinates or homogeneous coordinates)

$p(\alpha, \beta, \gamma) = \alpha a + \beta b + \gamma c \quad (\alpha + \beta + \gamma = 1)$



p is inside triangle if and only if $0 < \alpha < 1$, $0 < \beta < 1$, $0 < \gamma < 1$

$$\alpha = \frac{A_a}{A} = \frac{l_a l_a'}{l_a l_a' + l_b l_b' + l_c l_c'}$$

Also: $$\beta = \frac{A_b}{A} = \frac{l_b l_b'}{l_a l_a' + l_b l_b' + l_c l_c'}$$

$$\gamma = \frac{A_c}{A} = \frac{l_c l_c'}{l_a l_a' + l_b l_b' + l_c l_c'}$$

(A is total area of triangle)

(18) Generate random point in triangle with probability density $p(x) = 1/A$

r_1 and r_2 are random numbers, uniformly generated over the interval $[0, 1]$

$\alpha = 1 - \sqrt{r_1}$

$\beta = (1 - r_2)\sqrt{r_1}$

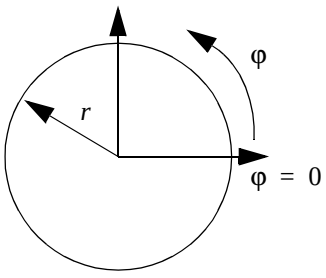
$\gamma = r_2\sqrt{r_1}$

with (α, β, γ) the barycentric coordinates (see 17) of the random point.

(method as described in R40)

C. DISKS

(19) Generate random point on unit disk with probability density $p(x) = 1/\pi$



In general, a disk is parametrized by:

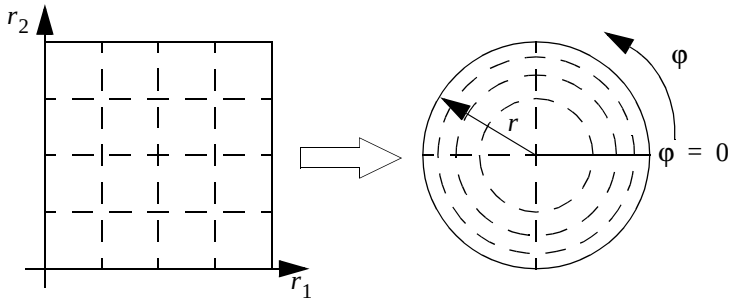
$\varphi \in [0, 2\pi]$

$r \in [0, 1]$

Random numbers $(r_1, r_2) \in [0, 1] \times [0, 1]$ need to be mapped to (φ, r) coordinates.

(19a) Polar map

This map preserves fractional area, but is not bicontinuous and has severe distortions.

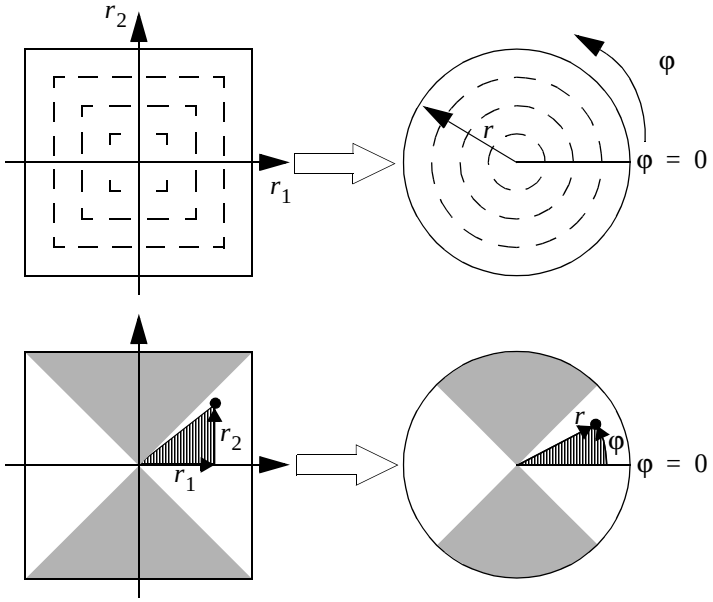


r_1 and r_2 are random numbers, both uniformly generated over the interval $[0, 1]$

$$r = \sqrt{r_1}$$
$$\varphi = 2\pi r_2$$

(19b) Concentric map

This map was first proposed by Shirley in the context of ray tracing and illumination computations (R31)[1]. This map preserves fractional area, is bicontinuous and has low distortion. See reference in footnote for an implementation.



r_1 and r_2 are random numbers, both uniformly generated over the interval $[-1, 1]$ (This can easily be transformed from random numbers in the interval $[0, 1]$).

1st triangular region ($r_1 > -r_2$ and $r_1 > r_2$):

$$r = r_1$$
$$\varphi = \frac{\pi}{4} \cdot \frac{r_2}{r_1}$$

1. Pictures in this section of the Compendium are based on the pictures in this book.

2nd triangular region ($r_1 < r_2$ and $r_1 > -r_2$):

$$r = r_2$$

$$\varphi = \frac{\pi}{4} \cdot \left(2 - \frac{r_1}{r_2}\right)$$

3rd triangular region ($r_1 < -r_2$ and $r_1 < r_2$):

$$r = -r_1$$

$$\varphi = \frac{\pi}{4} \cdot \left(4 + \frac{r_2}{r_1}\right)$$

4th triangular region ($r_1 > r_2$ and $r_1 < -r_2$):

$$r = -r_2$$

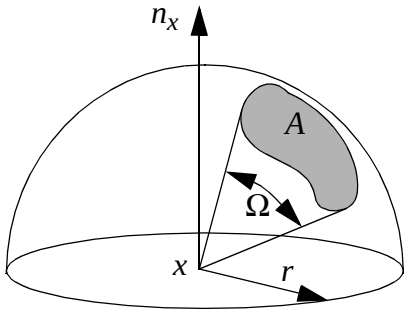
$$\varphi = \frac{\pi}{4} \cdot \left(6 - \frac{r_1}{r_2}\right)$$

IV. Hemispherical Geometry

A. GENERAL

(20) Finite Solid angle

$$\Omega = \frac{A}{r^2}$$

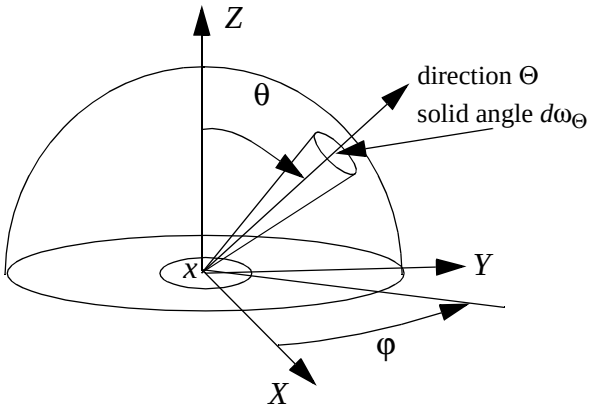


Solid angles are dimensionless and expressed in *steradians*.
The solid angle subtended by all space equals 4π.

(21) (Hemi-)Spherical coordinates: (φ, θ) parametrisation

direction $\Theta = (\varphi, \theta)$

$\varphi \in [0, 2\pi]$
$\theta \in [0, \pi/2]$



$$x = r\cos\varphi\sin\theta$$
$$y = r\sin\varphi\sin\theta$$
$$z = r\cos\theta$$

$$r = \sqrt{x^2 + y^2 + z^2}$$
$$\tan\varphi = y/x$$
$$\tan\theta = \frac{\sqrt{x^2 + y^2}}{z}$$

$$dxdydz = r^2\sin\theta dr d\theta d\varphi$$

Spherical coordinates: above formulas remain the same, except $\theta \in [0, \pi]$

(22) Differential solid angle for (φ, θ) parametrisation

$$d\omega_\Theta = \sin\theta d\theta d\varphi$$

Integral over the hemisphere: $\int_\Omega f(\Theta) d\omega_\Theta = \int_0^{2\pi}\int_0^{\pi/2} f(\varphi, \theta)\sin\theta d\theta d\varphi$

(23) (φ, c) parametrisation of hemisphere

Apply coordinate transform: $\begin{matrix}\varphi = \varphi \\ c = 1 - \cos\theta\end{matrix}$: $\int_\Omega f(\Theta) d\omega_\Theta = \int_0^{2\pi}\int_0^{\pi/2} f(\varphi, \theta)\sin\theta d\theta d\varphi = \int_0^{2\pi}\int_0^{1} f(\varphi, c) dc d\varphi$

Equally sized areas in (φ, c) space translate to equally sized solid angles.

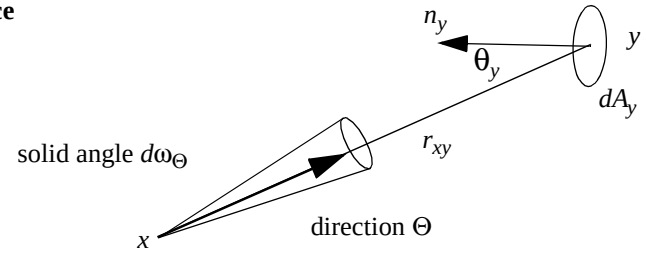
(24) (ξ_1, ξ_2) parametrisation of hemisphere

Apply coordinate transform: $\begin{aligned}\xi_1 &= \sin^2\theta \\ \xi_2 &= \frac{\varphi}{2\pi}\end{aligned} : \int_\Omega f(\Theta)\,d\omega_\Theta = \int_0^{2\pi}\int_0^{\pi/2} f(\theta,\varphi)\sin\theta\,d\theta\,d\varphi = \int_0^1\int_0^1 \frac{f(\xi_1,\xi_2)}{\sqrt{1-\xi_1^2}}\,d\xi_1\,d\xi_2$

Equally sized areas in (ξ_1, ξ_2) space translate to equally sized, cosine-weighted solid angles.

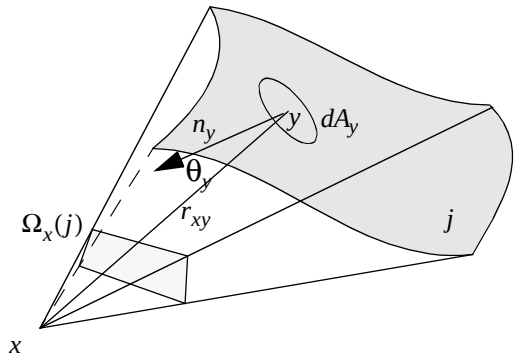
(25) Transformation between differential surface area and differential solid angle

$$d\omega_\Theta = \frac{\cos\theta_y\,dA_y}{r_{xy}^2}$$



(26) Solid angle subtended by a surface

$$\Omega_x(j) = \int_{A_j} \frac{\cos\theta_y}{r_{xy}^2}\,dA_y$$



(27) Visible solid angle subtended by a surface

Parts of the surface j might not be visible to x. Therefore, visibility between x and all surface points y on j needs to be included explicitly.

$$\Omega_x^{vis}(j) = \int_{A_j} \frac{\cos\theta_y}{r_{xy}^2} V(x,y)\,dA_y$$

(28) Solid angle subtended by a polygon

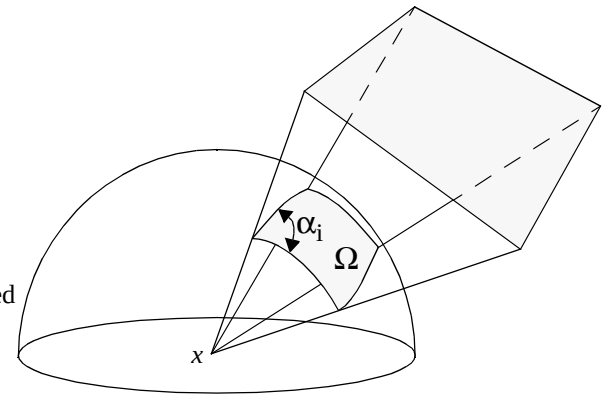
Girard's formula:

$$\Omega_x = \sum_i \alpha_i - (n-2)\pi$$

with:

n : number of vertices of the polygon

α_i : dihedral angles (angles between the planes formed by the edges of the polygon and the projection point x)



This formula is valid for convex and concave polygons.
An implementation is given in Graphics Gems (R27, R6).

(29) Tangent-sphere function

In the context of global illumination, one is often interested in the cosine of the angle between a direction Θ on the hemisphere and the normal vector n_x at a surface point x, but only if the direction is located at the same side of the surface of n_x. If Θ is 'below' the surface, the value is 0. Some authors (R34) introduce the 'tangent-sphere' function for this purpose:

$$T_{n_x}(\Theta) = \begin{cases} n_x \cdot \Theta & \text{if } n_x \cdot \Theta \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

In this document, the notation $\cos(\Theta, n_x)$ is often used.

(30) Useful integrals (cosine lobes) over the hemisphere (see also 33, 34, 35 and 36)

$$\int_{\Omega} d\omega_{\Theta} = \int_0^{2\pi} d\varphi \int_0^{\pi/2} \sin\theta d\theta = 2\pi$$

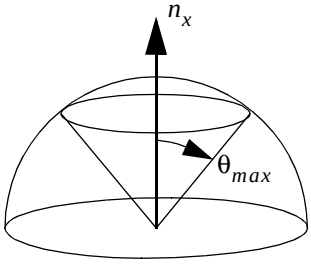
$$\int_{\Omega} \cos(\Theta, n_x) d\omega_{\Theta} = \int_0^{2\pi} d\varphi \int_0^{\pi/2} \cos\theta \sin\theta d\theta = \pi$$

$$\int_{\Omega} \cos^n(\Theta, n_x) d\omega_{\Theta} = \int_0^{2\pi} d\varphi \int_0^{\pi/2} \cos^n\theta \sin\theta d\theta = \frac{2\pi}{n+1}$$

Integrals of cosine lobes are useful because many BRDF models (e.g. Lafortune model, see 70) make use of these lobes, although usually not centered around the normal.

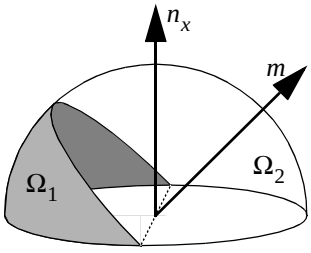
Limiting the integration area to the spherical cap bounded by $\theta \in [0, \theta_{max}]$

$$\int_{\Omega_m} \cos^n(\Theta, n_x) d\omega_{\Theta} = \int_0^{2\pi} d\varphi \int_0^{\theta_m} \cos^n\theta \sin\theta d\theta = \frac{2\pi}{n+1}(1 - \cos^{n+1}\theta_{max})$$



(31) Useful integrals over spherical digons

(31a) $\int_{\Omega_2} \cos^n(\Theta, m) \cos(\Theta, n_x) d\omega_{\Theta}$ (cosine lobe is non-zero on Ω_2 only)



A method for computing this integral was presented by J. Arvo, as part of a general method for computing double-axis moments on the hemisphere (R2). Pseudocode looks as follows:

```
F(nx, m, n)
     S = 0;
     d = m.nx;
     c = sqrt(1-d*d);
     T = if even(n) then π/2 else c;
     A = if even(n) then π/2 else π-acos(d);
     i = if even(n) then 0 else 1;
     while i <= n-2 do
```

```
            S = S+T;
            T = T*c*c*(i+1)/(i+2);
            i = i+2;
            endwhile
        return 2*(T + d*A + d*d*S)/(n+2)
    end
```

(31b) $\int_{\Omega_2} \cos^n(\Theta, m)d\omega_\Theta$ (cosine lobe is non-zero on Ω_2 only)

An expression can be derived from the same paper (although this form is not explicitly mentioned):

n even: $$\int_{\Omega_2} \cos^n(\Theta, m)d\omega_\Theta = \frac{2(\pi - \theta_m) + \cos\theta_m[\sin^{n-1}\theta_m F_{n-1} + \sin^{n-3}\theta_m F_{n-3} + \ldots + \sin\theta_m F_1]}{n+1}$$

n odd: $$\int_{\Omega_2} \cos^n(\Theta, m)d\omega_\Theta = \frac{\pi + \cos\theta_m[\sin^{n-1}\theta_m F_{n-1} + \sin^{n-3}\theta_m F_{n-3} + \ldots + F_0]}{n+1}$$

where: $$\theta_m = \text{acos}(m \cdot n_x)$$
$$F_n = \frac{n-1}{n}F_{n-2} \qquad F_0 = \pi \qquad F_1 = 2$$

(32) Dirac-impulse on hemisphere (see also 1)

$$\delta(\Theta_1 - \Theta) = \delta(\cos\theta_1 - \cos\theta)\delta(\varphi_1 - \varphi)$$
$$\Theta_1 = (\theta_1, \varphi_1)$$
$$\Theta = (\theta, \varphi)$$

such that:

$$\int_\Omega \delta(\Theta_1 - \Theta)f(\Theta)d\omega_\Theta = \int_0^{2\pi} d\varphi\delta(\varphi_1 - \varphi) \int_0^{\pi/2} d(\cos\theta)\delta(\cos\theta_1 - \cos\theta)f(\varphi, \theta)$$
$$= \int_0^{2\pi} d\varphi\delta(\varphi_1 - \varphi)f(\varphi, \theta_1) = f(\varphi_1, \theta_1) = f(\Theta_1)$$

B. GENERATING POINTS AND DIRECTIONS ON THE (HEMI)SPHERE

Generating random directions over the hemisphere is a fundamental operations in most Monte Carlo-based rendering algorithms. The rendering equation (see section IX) is usually expressed as an integral over the hemisphere, so sampling the hemispherical domain requires generating directions over the hemisphere.

r_1 and r_2 are random numbers, uniformly generated over the interval $[0, 1]$. Some of the formulas can be simplified by substituting a uniform random variable r with $1 - r$ or vice versa. Since both have the distribution, the resulting distribution of directions is not affected.

(33) Generate random point on sphere (c_x, c_y, c_z, R) with density $p(\Theta) = \frac{1}{4\pi R^2}$

$$\varphi = 2\pi r_1$$
$$\theta = \operatorname{acos}(1 - 2r_2)$$

$$x = c_x + 2R\cos(2\pi r_1)\sqrt{r_2(1-r_2)}$$
$$y = c_y + 2R\sin(2\pi r_1)\sqrt{r_2(1-r_2)}$$
$$z = c_z + R(1 - 2r_2)$$

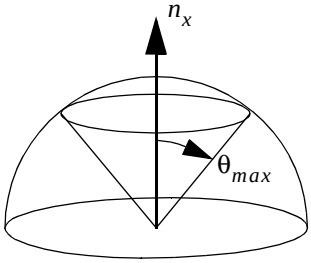
(34) Generate random direction on unit hemisphere proportional to solid angle

PDF: $p(\Theta) = \frac{1}{2\pi}$

$$\varphi = 2\pi r_1$$
$$\theta = \operatorname{acos}(r_2)$$

$$x = \cos(2\pi r_1)\sqrt{1 - r_2^2}$$
$$y = \sin(2\pi r_1)\sqrt{1 - r_2^2}$$
$$z = r_2$$

With $\theta \in [0, \theta_{max}]$ and $p(\Theta) = \frac{1}{2\pi(1 - \cos\theta_{max})}$:



$$\varphi = 2\pi r_1$$
$$\theta = \operatorname{acos}(1 - r_2(1 - \cos\theta_{max}))$$

$$x = \cos(2\pi r_1)\sqrt{1 - (1 - r_2(1 - \cos\theta_{max}))^2}$$
$$y = \sin(2\pi r_1)\sqrt{1 - (1 - r_2(1 - \cos\theta_{max}))^2}$$
$$z = 1 - r_2(1 - \cos\theta_{max})$$

Generating points uniformly on the disk (see 19), and then applying the following transformation, also produces a uniform distribution of points on the hemisphere:

Point on disk: (φ_d, r_d)

Resulting point on hemisphere:

$$\varphi = \varphi_d$$
$$\theta = \operatorname{acos}(1 - r_d^2)$$

or

$$x = \cos\varphi_d \cdot r_d\sqrt{2 - r_d^2}$$
$$y = \sin\varphi_d \cdot r_d\sqrt{2 - r_d^2}$$
$$z = 1 - r_d^2$$

(35) Generate random direction on unit hemisphere proportional to cosine-weighted solid angle

PDF: $p(\Theta) = \frac{\cos\theta}{\pi}$

$$\varphi = 2\pi r_1 \qquad \theta = \operatorname{acos}(\sqrt{r_2})$$

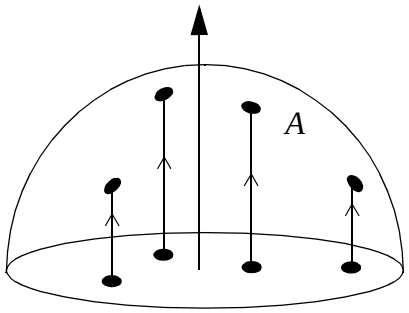
$$x = \cos(2\pi r_1)\sqrt{1-r_2} \qquad y = \sin(2\pi r_1)\sqrt{1-r_2} \qquad z = \sqrt{r_2}$$

With $\theta \in [0, \theta_{max}]$ and $p(\Theta) = \dfrac{\cos\theta}{\pi \sin^2\theta_{max}}$

$$\varphi = 2\pi r_1 \qquad \theta = \operatorname{acos}(\sqrt{1-r_2\sin^2\theta_{max}})$$

$$x = \cos(2\pi r_1)\sin\theta_{max}\sqrt{r_2} \qquad y = \sin(2\pi r_1)\sin\theta_{max}\sqrt{r_2} \qquad z = \sqrt{1-r_2\sin^2\theta_{max}}$$

Generating points uniformly on the disk (see 19), and then projecting them on the hemisphere, also gives a cosine-weighted distribution of points on the hemisphere.



Point on disk: (φ_d, r_d)

Resulting point on hemisphere: $\varphi = \varphi_d \quad \theta = \operatorname{asin}(r_d)$ or $x = r_d\cos\varphi_d \quad y = r_d\sin\varphi_d \quad z = \sqrt{1-r_d^2}$

(36) Generate random direction on unit hemisphere proportional to cosine lobe around normal

PDF: $p(\Theta) = \dfrac{n+1}{2\pi}\cos^n\theta$

$$\varphi = 2\pi r_1 \qquad \theta = \operatorname{acos}\left(r_2^{\frac{1}{n+1}}\right)$$

$$x = \cos(2\pi r_1)\sqrt{1-r_2^{\frac{2}{n+1}}} \qquad y = \sin(2\pi r_1)\sqrt{1-r_2^{\frac{2}{n+1}}} \qquad z = r_2^{\frac{1}{n+1}}$$

With $\theta \in [0, \theta_{max}]$ and $p(\Theta) = \dfrac{(n+1)\cos^n\theta}{2\pi(1-\cos^{n+1}\theta_{max})}$

$$\varphi = 2\pi r_1 \qquad \theta = \operatorname{acos}\left((1-r_2(1-\cos^{n+1}\theta_{max}))^{\frac{1}{n+1}}\right)$$

$$x = \cos(2\pi r_1)\sqrt{1-(1-r_2(1-\cos^{n+1}\theta_{max}))^{\frac{2}{n+1}}} \qquad y = \sin(2\pi r_1)\sqrt{1-(1-r_2(1-\cos^{n+1}\theta_{max}))^{\frac{2}{n+1}}} \qquad z = (1-r_2(1-\cos^{n+1}\theta_{max}))^{\frac{1}{n+1}}$$

$n = 0$ produces (34); $n = 1$ produces (35).

(37) Generate uniform random direction on a spherical triangle

See the paper published by J. Arvo in SIGGRAPH 95 for a complete formula and algorithm (R1).

(38) Generate random direction on spherical digon; density proportional to $\cos^n\alpha$; α angle from off-normal axis

1. Generate direction on unit hemisphere proportional to $\cos^n\theta$ using (36).
2. Transform direction by transforming normal to off-normal axis.
3. If transformed direction has angle greater then $\pi/2$ with normal, reject direction and try again.
4. Compute correct pdf-value by normalizing $\cos^n\alpha$ using (31b).

V. Monte Carlo Integration[1,2]

(39) General Properties of Monte Carlo estimators

Let $F(N)$ be a stochastic estimator for quantity Q, using N samples.

Error: $F(N) - Q$

Bias: $\beta(F(N)) = E[F(N) - Q]$. If bias is zero for all N, then estimator is unbiased.

Estimator is consistent if: $Prob\left\{ \lim_{N \to \infty} F(N) = Q \right\} = 1$. A sufficient condition for an estimator to be unbiased is: $\lim_{N \to \infty} \beta(F(N)) = \lim_{N \to \infty} \sigma^2[F(N)] = 0$

(40) Basic MC integration

$$I = \int_D f(x)dx$$

$$D = [\alpha_1 \ldots \beta_1] \times [\alpha_2 \ldots \beta_2] \times \ldots \times [\alpha_d \ldots \beta_d] \qquad (\alpha_i, \beta_i \in \Re)$$

Generate a sequence of samples $(x_1, x_2, x_3, ..., x_N)$ using a uniform pdf $p(x) = \left(\prod_{i=1}^{d} (\beta_i - \alpha_i) \right)^{-1}$

$$\langle I \rangle = \left(\frac{1}{N} \sum_{i=1}^{N} f(x_i) \right) \cdot \left(\prod_{i=1}^{d} (\beta_i - \alpha_i) \right) \qquad E[\langle I \rangle] = \int_D f(x)dx \text{ (unbiased estimator)}$$

$$\sigma^2[\langle I \rangle] = \frac{1}{N \cdot \prod_{i=1}^{d} (\beta_i - \alpha_i)} \int_D (f(x) - I)^2 dx = \frac{1}{N \cdot \prod_{i=1}^{d} (\beta_i - \alpha_i)} \left(\int_D f(x)^2 dx - I^2 \right)$$

(41) MC integration using importance sampling

Generate a sequence of samples $(x_1, x_2, x_3, ..., x_N)$ using pdf $p(x)$

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \qquad E[\langle I \rangle] = \int_D f(x)dx$$

$$\sigma^2[\langle I \rangle] = \frac{1}{N} \int_D \left(\frac{f(x)}{p(x)} - I \right)^2 p(x)dx = \frac{1}{N} \left(\int_D \frac{f(x)^2}{p(x)} dx - I^2 \right)$$

1. For an introduction on Monte Carlo integration: (R18, R37, R21, R14)
2. An excellent text covering Monte Carlo methods for global Illumination is the Ph.D. thesis of Eric Veach (R44).

Optimal pdf for importance sampling, giving minimum variance:

$$p(x) = \frac{|f(x)|}{\int_D f(x)dx}$$

For the estimator to be unbiased, $p(x)$ must be non-zero wherever $f(x)$ is non-zero.

(42) MC integration using stratified sampling

$$\int_0^1 f(x)dx = \int_0^{\alpha_1} f(x)dx + \int_{\alpha_1}^{\alpha_2} f(x)dx + ... + \int_{\alpha_{m-2}}^{\alpha_{m-1}} f(x)dx + \int_{\alpha_{m-1}}^{1} f(x)dx$$

If each stratum j receives n_j samples, distributed uniformly within each stratum:

$$\sigma^2 = \sum_{j=1}^{m} \frac{(\alpha_j - \alpha_{j-1})}{n_j} \int_{\alpha_{j-1}}^{\alpha_j} f(x)^2 dx - \sum_{j=1}^{m} \frac{1}{n_j} \left\{ \int_{\alpha_{j-1}}^{\alpha_j} f(x)dx \right\}^2$$

If all strata have equal size, and each stratum contains one sample:

$$\sigma^2 = \frac{1}{N} \int_0^1 f(x)^2 dx - \sum_{j=1}^{N} \left\{ \int_{\alpha_{j-1}}^{\alpha_j} f(x)dx \right\}^2$$

(43) Combined estimators

$$\langle I \rangle = \sum_{j=1}^{n} \frac{1}{N_j} \sum_{i=1}^{N_j} w_j(x_{i,j}) \frac{f(x_{i,j})}{p_j(x_{i,j})}$$

with $x_{i,j}$ the i^{th} sample, from a total of N_j, taken from pdf $p_j(x)$.

If $\sum_{j=1}^{n} w_j(x) = 1$ for all x, then $\langle I \rangle$ is unbiased.
(see R43)

(44) Combined estimators: balance heuristic

Balance Heuristic: $w_j(x) = \frac{c_j p_j(x)}{\sum_j c_j p_j(x)}$ with $N_j = c_j N$, with N total number of samples.

If $\langle I \rangle_{bh}$ is obtained by the balance heuristic, and $\langle I \rangle$ by any other set of $w_j(x)$, then:

$$\sigma^2[\langle I \rangle_{bh}] \le \sigma^2[\langle I \rangle] + \left(\frac{1}{min_j N_j} - \frac{1}{\sum_j N_j} \right) \int_D f(x)dx$$

One sample model ($N = 1$): if p_j is cosen with probability c_j, then the balance heuristic gives the lowest variance.

(45) Efficiency of a Monte Carlo estimator

$\varepsilon = \frac{1}{T\sigma^2}$ where T is the time to take 1 sample, and σ^2 is the variance for 1 sample.

(see R21, pp. 91-92)

(46) Quasi-random sequences

Radical inverse function: $i = \sum_{j=0}^{\infty} a_j b^j \Rightarrow \Phi_b(i) = \sum_{j=0}^{\infty} a_j b^{-j-1}$

a. Van der Corput sequence

$x_i = \Phi_b(i)$

b. Halton sequence (s dimensions)

$x_i = (\Phi_{b_1}(i), \Phi_{b_2}(i), \Phi_{b_3}(i), \Phi_{b_4}(i), \ldots, \Phi_{b_s}(i))$ with $b_1, b_2, b_3, \ldots, b_s$ relative primes.

c. Hammersley sequence (s dimensions, length N)

$x_i = \left(\frac{i}{N}, \Phi_{b_1}(i), \Phi_{b_2}(i), \Phi_{b_3}(i), \Phi_{b_4}(i), \ldots, \Phi_{b_{s-1}}(i)\right)$ with $b_1, b_2, b_3, \ldots, b_{s-1}$ relative primes.

VI. Radiometry & Photometry

- A general overview of units and measurements can be found at the "**How many?**" website (`http://www.unc.edu/~rowlett/units/index.html`).
- Another overview of light measurements and units is compiled in the **Light Measurement Hanbook**: `http://www.intl-light.com/handbook/registered.html`

(47) Radiometric and Photometric units

Radiometry			Photometry	
Joule	Radiant Energy Q	→	Luminous Energy	Talbot
Watt	Radiant Power	→	Luminous Power	Lumen (lm)
Watt/m^2	Radiosity (Radiant exitance) Irradiance	→	Luminosity Illuminance	Lux (lm/m^2)
Watt/sr	Radiant Intensity	→	Luminous Intensity	Candela[a] (cd or lm/sr)
Watt/m^2sr	Radiance	→	Luminance	Nit (cd/m^2 or lm/m^2sr)

a. From the "How Many" website: Candela: the fundamental SI unit for measuring the intensity of light. Candela is the Latin word for "candle." The unit has a long and complicated history. Originally, it represented the intensity of an actual candle, assumed to be burning whale tallow at a specified rate in grains per hour. Later this definition was replaced with a definition in terms of the light produced by the filament of an incandescent light bulb. Still later a standard was adopted which defined the candela as the intensity of 1/600 000 square meter of a "black body" (a perfect radiator of energy) at the temperature of freezing platinum (2042 K) and a pressure of 1 atmosphere. This definition has also been discarded, and the candela is now defined to be the luminous intensity of a light source producing single-frequency light at a frequency of 540 terahertz (THz) with a power of 1/683 watt per steradian, or 18.3988 milliwatts over a complete sphere centered at the light source. The frequency of 540 THz corresponds to a wave length of approximately 555.17 nanometers (nm). Light of this frequency has a yellow-green color and roughly the same visual brightness as ordinary daylight. In addition, normal human eyes are more sensitive to this wavelength than to any other. In order to produce 1 candela of single-frequency light of wavelength l, a lamp would have to radiate 1/(683V(l)) watts per steradian, where V(l) is the relative sensitivity of the eye at wavelength l. Values of V(l) are defined by the International Commission on Illumination (CIE).

(48) Flux: radiant energy flowing through a surface per unit time (Watt = Joule/sec)

$$\Phi = P = \frac{dQ}{dt}$$

(49) Irradiance: incident flux per unit surface area (Watt/m^2)

$$E = \frac{d\Phi}{dA}$$

Radiant exitance (a.k.a Radiosity): departing flux per unit surface area (Watt/m^2)

$$M = B = \frac{d\Phi}{dA}$$

(50) Radiant Intensity: flux per solid angle (Watt/sr)

$$I = \frac{d\Phi}{d\omega}$$

(51) Radiance: flux per solid angle per unit projected area (Watt/m^{2}sr)

$$L = \frac{dE}{d\omega} = \frac{d^2\Phi}{d\omega dA^{\perp}} = \frac{d^2\Phi}{d\omega dA\cos\theta}$$

(51a) **Notations**:

$L(x \to \Theta)$: radiance leaving point x in direction Θ
$L(x \leftarrow \Theta)$: radiance arriving at point x from direction Θ
$L(x \to y)$: radiance leaving point x, arriving at point y
$L(x \leftarrow y)$: radiance arriving at point x, coming from point y

(51b) Wavelength Dependency:

$$L(x \to \Theta) = \int_{380nm}^{780nm} L(x \to \Theta, \lambda) d\lambda$$

Wavelength dependency is usually assumed in radiometric eqautions.

(51c) Invariant along straight lines:

Radiance remains invariant along a straight path in vacuum:

$L(x \to y) = L(y \leftarrow x)$ if x and y are mutually visible
$L(x \to \Theta) = L(r(x, \Theta) \leftarrow -\Theta)$

Proof: consider power exchange $d^2\Phi$ between 2 differential surfaces dA_x and dA_y. Power $d^2\Phi$ arriving at dA_y from dA_x must equal power leaving dA_x in the direction of dA_y (full proof see R13 p.24).

(51d) Integration: specify integration domain if specific values are needed

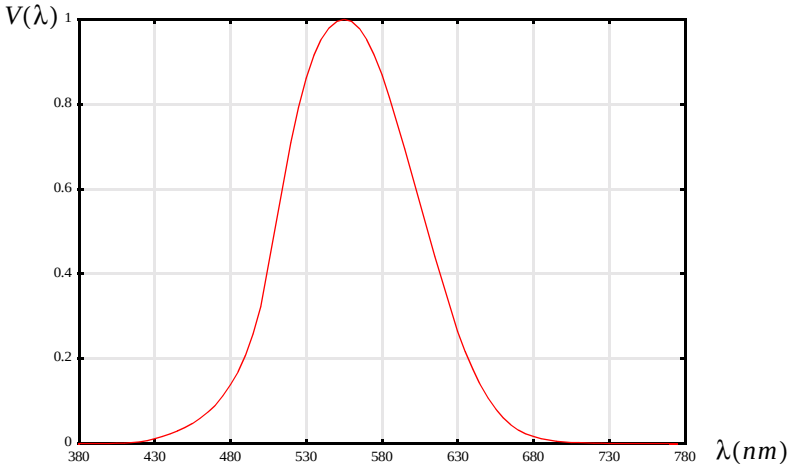
$$\Phi = \iint L(x \to \Theta)\cos\theta d\omega_{\Theta} dA_x = \int E(x) dA_x = \int I(\Theta) d\omega_{\Theta}$$
$$E(x) = \int L(x \leftarrow \Theta)\cos\theta d\omega_{\Theta}$$
$$B(x) = \int L(x \to \Theta)\cos\theta d\omega_{\Theta}$$
$$I(\Theta) = \int L(x \to \Theta)\cos\theta dA_x$$

For a diffuse emitter ($L(x \to \Theta) = L$) with surface area A and hemispherical solid angle (2π sr):

$$\Phi = LA\pi$$
$$E(x) = L\pi$$
$$B(x) = L\pi$$
$$I(\Theta) = LA\cos\theta$$

(52) Radiometric quantities → Photometric quantities

Integrate radiometric quantity $R(\lambda)$ weighted by the *spectral luminous efficiency curve* $V(\lambda)$. This curve is the same as the color matching function $\bar{y}(\lambda)$ in the CIE XYZ color system (see 116)



$$P = K_m \int_{380nm}^{770nm} V(\lambda)R(\lambda)d\lambda \qquad K_m = 680 \text{lumen/watt}$$

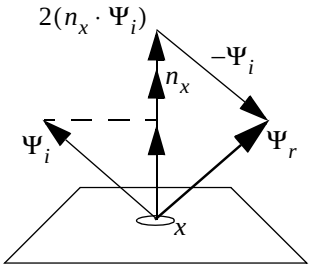
Tabular data for the spectral luminous efficiency curve can be found in R16, p. 1170.

VII. Optics

(53) Reflection at perfect mirror (incoming, outgoing direction, surface normal in same plane)

$\theta_r = \theta_i$ (incident angle = reflected angle)

Vector computation (incident direction Ψ_i, normal n_x, reflection vector Ψ_r): $\Psi_r = 2(n_x \cdot \Psi_i) - \Psi_i$

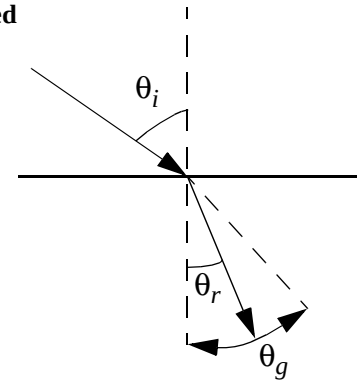


(54) Refraction at transition from vacuum to material (incoming, refracted direction, surface normal in same plane)

$$\frac{\sin\theta_i}{\sin\theta_r} = n \qquad \sin\theta_g = \frac{1}{n}$$

$$0 < \theta_i < \frac{\pi}{2} \qquad 0 < \theta_r < \theta_g$$

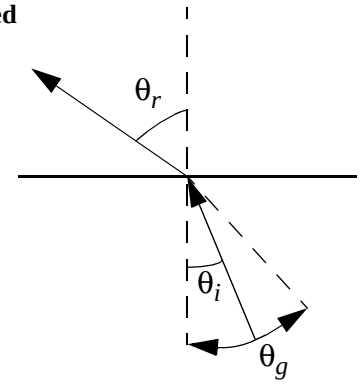
for small angles: $\frac{\theta_i}{\theta_r} = n$ (Kepler's formula)



(55) Refraction at transition from material to vacuum (incoming, refracted direction, surface normal in same plane)

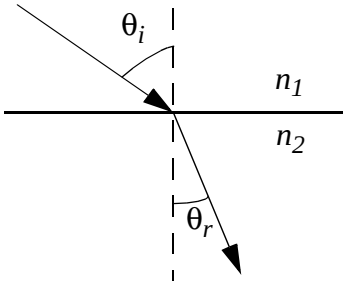
if $0 < \theta_i < \theta_g$ $\qquad \frac{\sin\theta_i}{\sin\theta_r} = \frac{1}{n}$

for small angles: $\frac{\theta_i}{\theta_r} = \frac{1}{n}$

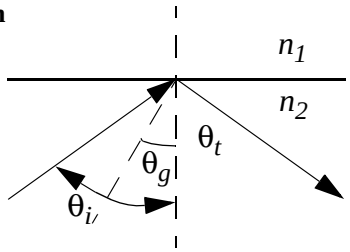


(56) Refraction at transition from material 1 to material 2 (incoming, refracted direction, surface normal in same plane)

$$\frac{\sin\theta_i}{\sin\theta_r} = \frac{n_2}{n_1} = n_{1\to 2}$$



(57) Total internal refraction (incoming, refracted direction, surface normal in same plane)



if $\theta_g < \theta_i < \frac{\pi}{2}$ $\quad \theta_t = \theta_i$

(58) Fresnel Reflection - Conductors

Material 1 to material 2; relative index of refraction $\frac{n_2}{n_1} = \eta + j\kappa$; θ is angle of incidence. The Fresnel coefficients express the directional-hemispherical spectral specular reflectivity.

$$F_s = \frac{a^2 + b^2 - 2a\cos\theta + \cos^2\theta}{a^2 + b^2 + 2a\cos\theta + \cos^2\theta}$$

$$F_p = F_s \frac{a^2 + b^2 - 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}{a^2 + b^2 + 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}$$

where a and b are given by:

$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} + (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} - (\eta^2 - \kappa^2 - \sin^2\theta)$$

Polarized light: reflectance $F_r = sF_s + pF_p$ with $s + p = 1$

Unpolarized light: $F_r = \frac{F_s + F_p}{2}$

F_s and F_p are also written as $F_\perp$ and $F_{//}$ in various textbooks[1].

(59) Fresnel Reflection - Dielectrics

Material 1 to material 2; relative index of refraction $\frac{n_2}{n_1} = \eta$; θ_i is angle of incidence, θ_r is angle of refraction (given by 56). The Fresnel coefficients express the directional-hemispherical spectral specular reflectivity.

$$F_s = \left(\frac{\cos\theta_r / \cos\theta_i - 1/\eta}{\cos\theta_r / \cos\theta_i + 1/\eta}\right)^2 = \frac{\sin^2(\theta_i - \theta_r)}{\sin^2(\theta_i + \theta_r)}$$

$$F_p = \left(\frac{\cos\theta_i / \cos\theta_r - 1/\eta}{\cos\theta_i / \cos\theta_r + 1/\eta}\right)^2 = \frac{\tan^2(\theta_i - \theta_r)}{\tan^2(\theta_i + \theta_r)}$$

(F_s and F_p are also written as $F_\perp$ and $F_{//}$ in various textbooks)

Unpolarized light: $F_r = \frac{F_s + F_p}{2} = \frac{1}{2} \cdot \frac{\sin^2(\theta_i - \theta_r)}{\sin^2(\theta_i + \theta_r)} \cdot \left(1 + \frac{\cos^2(\theta_i + \theta_r)}{\cos^2(\theta_i - \theta_r)}\right)$

1. s stands for '*senkrecht*', German for *perpendicular.*

If $\theta_i = 0$, and thus $\theta_r = 0 : F_r = F_s = F_p = \left(\frac{\eta - 1}{\eta + 1}\right)^2$ (also sometimes written as F_0)

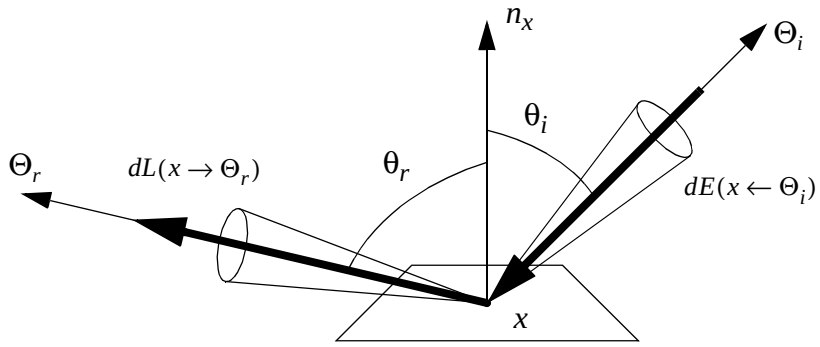
(60) Index of Refraction Data Values

A website listing all sort of possible data is at `http://www.luxpop.com/`

VIII. Bidirectional Reflectance Distribution Functions (BRDFs)

A. General properties

(61) BRDF: $f_r(x, \Theta_i \rightarrow \Theta_r) = \dfrac{dL(x \rightarrow \Theta_r)}{dE(x \leftarrow \Theta_i)} = \dfrac{dL(x \rightarrow \Theta_r)}{L(x \leftarrow \Theta_i)\cos\theta_i d\omega_{\Theta_i}}$



BRDF is dimensionless but is expressed as 1/sr

BTDF = Bidirectional Transmittance Function: similar as BRDF but defined for transmittance
BSDF = combination of 2 BRDFs and 2 BTDFs (one pair for each side of the surface)

(62) BRDF Reciprocity

$$f_r(x, \Theta_i \rightarrow \Theta_r) = f_r(x, \Theta_r \rightarrow \Theta_i) = f_r(x, \Theta_i \leftrightarrow \Theta_r)$$

(63) BRDF Energy conservation

$$\forall\Theta: \int_{\Omega_x} f_r(x, \Theta \leftrightarrow \Psi)\cos(n_x, \Psi)d\omega_\Psi \leq 1$$

(64) Biconical Reflectance

$$\rho(\Theta_{in} \rightarrow \Theta_{out}) = \frac{\int_{\Omega_{out}}\int_{\Omega_{in}} f_r(\Theta_{in} \leftrightarrow \Theta_{out})\cos\theta_{out}\cos\theta_{in}d\omega_{\Theta_{in}}d\omega_{\Theta_{out}}}{\int_{\Omega_{in}} \cos\theta_{in}d\omega_{\Theta_{in}}}$$

Ω_{in} and Ω_{out} can be a single direction, a solid angle, or the hemisphere. Combine the words ‘directional’, ‘conical’ and ‘hemispherical’ to obtain the right adjective for the reflectance. E.g. Biconical reflectance; directional-hemispherical reflectance etc. (definitions from R7, p. 32).

(65) Lambertian Diffuse Reflection

$$f_r(x, \Theta_i \leftrightarrow \Theta_r) = f_{r,d} = \text{constant}$$

Outgoing radiance due to incoming radiance field:

$$L(x \rightarrow \Theta) = f_{r,d}\int_{\Omega_{in}} L(x \leftarrow \theta_{in})\cos\theta_{in}d\omega_{\Theta_{in}} = f_{r,d}E$$

Bihemispherical reflectance: $\rho_d = f_{r,d}\pi$ and $\rho_d = \frac{B}{E}$

(B is the hemispherical radiosity and E is the hemispherical irradiance, see VI)

B. BRDF MODELS

General notations for BRDF models:

Θ_r Outgoing direction of light.

Θ_i Incoming direction of light

Θ_s Perfect specular direction w.r.t. Θ_i

Θ_h Halfway vector between Θ_r and Θ_i

n_x Normal vector at surface point x where BRDF is evaluated

Spherical coordinates for each direction are using the same index. E.g. for the outgoing direction (φ_r, θ_r)

Cartesian coordinates are also using the same index. E.g. for the outgoing direction (x_r, y_r, z_r). Normally, these will assumed to be normalized.

In general, negative values for cosine-lobes are ignored and set to zero.

(66) Modified Phong-BRDF[1]

$$f_r(\Theta_i \leftrightarrow \Theta_r) = k_d + k_s \cos^n(\Theta_r, \Theta_s)$$

or also:

$$f_r(\Theta_i \leftrightarrow \Theta_r) = \frac{\rho_d}{\pi} + \frac{\rho_s(n+2)}{2\pi}\cos^n(\Theta_r, \Theta_s) \text{ with } \rho_d = \pi k_d \text{ and } \rho_s = \frac{2\pi}{n+2}k_s$$

Energy conservation: $\rho_d + \rho_s \leq 1$. ρ_d and ρ_s are the maximum energy ($\Theta_i = n_x$) reflected through the diffuse part and specular part respectively. In other words:

$$\rho_d = \int_{\Omega_x} k_d \cos(n_x, \Psi) d\omega_\Psi \qquad \rho_s = \int_{\Omega_x} k_s \cos^{n+1}(n_x, \Psi) d\omega_\Psi \text{ (to compute, see 30)}$$

The energy reflected through the specular lobe, given a direction Θ_i:

$$\rho_s' = \int_{\Omega_x} k_s \cos^n(\Theta_r, \Theta_s)\cos(n_x, \Psi) d\omega_\Psi \qquad \rho_s' \leq \rho_s \text{ (to compute, see 31)}$$

Proportional sampling: There are various ways to sample according to the modified Phong-BRDF, using a combination of 34, 35, 36 and 38. In a global illumination context, one is often interested in computing an integral of the following type (see 72), using Monte Carlo integration:

$$I(\Theta) = \int_{\Omega_x} L(\Psi) f_r(\Psi \leftrightarrow \Theta)\cos(n_x, \Psi) d\omega_\Psi$$

1. The modified Phong-BRDF is very similar to the Phong shading function and was introduced as a BRDF in R26. See also R23.

Since the modified Phong-BRDF is often used in global illumination algorithms, there are various strategies for sampling proportional to the modified Phong-BRDF.

a. The integral can be split in two parts, compute each part independently using PDFs $p_1(\Psi)$ and $p_2(\Psi)$.

$$I = \int_{\Omega_x} L(\Psi) k_d \cos(n_x, \Psi) d\omega_\Psi + \int_{\Omega_x} L(\Psi) k_s \cos^n(\Psi, \Theta_s) \cos(n_x, \Psi) d\omega_\Psi = I_1 + I_2$$

$$\langle I \rangle = \langle I_1 \rangle + \langle I_2 \rangle$$

$$= \frac{1}{N_1} \sum_{i=1}^{N_1} \frac{L(\Psi_i) k_d \cos(n_x, \Psi_i)}{p_1(\Psi_i)} + \frac{1}{N_2} \sum_{i=1}^{N_2} \frac{L(\Psi_i) k_s \cos^n(\Psi_i, \Theta_s) \cos(n_x, \Psi_i)}{p_2(\Psi_i)}$$

Some interesting choices for $p_1(\Psi)$:

$p_1(\Psi) = \frac{1}{2\pi}$ (see 34): $\langle I_1 \rangle = \frac{2\rho_d}{N_1} \sum_{i=1}^{N_1} L(\Psi_i) \cos(n_x, \Psi_i)$

$p_1(\Psi) = \frac{\cos(n_x, \Psi)}{\pi}$ (see 35): $\langle I_1 \rangle = \frac{\rho_d}{N_1} \sum_{i=1}^{N_1} L(\Psi_i)$

Some interesting choices for $p_2(\Psi)$:

Uniform over hemisphere (see 34): $p_2(\Psi) = \frac{1}{2\pi}$; $\langle I_2 \rangle = \frac{2\pi k_s}{N_2} \sum_{i=1}^{N_2} L(\Psi_i) \cos^n(\Psi_i, \Theta_s) \cos(n_x, \Psi_i)$

Cosine -weighted (see 35): $p_2(\Psi) = \frac{\cos(n_x, \Psi)}{\pi}$; $\langle I_2 \rangle = \frac{\pi k_s}{N_2} \sum_{i=1}^{N_2} L(\Psi_i) \cos^n(\Psi_i, \Theta_s)$

Proportional to specular lobe of BRDF (see 38): $p_2(\Psi) = \frac{\cos^n(\Psi, \Theta_s)}{\int_{\Omega_x} \cos^n(\Psi, \Theta_s) d\omega_\Psi}$;

$$\langle I_2 \rangle = \frac{k_s}{N_2} \left(\int_{\Omega_x} \cos^n(\Psi, \Theta_s) d\omega_\Psi \right) \sum_{i=1}^{N_2} L(\Psi_i) \cos(n_x, \Psi_i)$$

b. Integral is split in two parts, use discrete probabilistic selection of what term to evaluate.

Select random event based on discrete PDF (q_1, q_2, q_3) with $q_1 + q_2 + q_3 = 1$. Then generate Ψ_i using either $p_1(\Psi)$ or $p_2(\Psi)$.

if event 1: $eval(\Psi_i) = \frac{L(\Psi_i) k_d \cos(n_x, \Psi_i)}{q_1 p_1(\Psi_i)}$

if event 2: $eval(\Psi_i) = \frac{L(\Psi_i) k_s \cos^n(\Psi_i, \Theta_s) \cos(n_x, \Psi_i)}{q_2 p_2(\Psi_i)}$

if event 3: $eval(\Psi_i) = 0$ (If $L(\Psi)$ contains recursive evaluations, this can be used to stop recursion).

Total estimator: $\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} eval(\Psi_i)$

Interesting choices for $p_1(\Psi)$, $p_2(\Psi)$: see above.

Interesting choices for (q_1, q_2, q_3):

An obvious choice is to pick $q_1 = \rho_d$, $q_2 = \rho_s'$, $q_3 = 1 - \rho_d - \rho_s'$ (proportional to energy reflected in different modes). However, this requires the evaluation of ρ_s' for every value of Θ. Another choice is to use the following values: $q_1 = \rho_d$, $q_2 = \rho_s$, $q_3 = 1 - \rho_d - \rho_s$. There will be some more samples drawn in the specular lobe relative to the reflected energy in the lobe, but this can be countered by NOT resampling any directions that are located in the part of the lobe below the surface. These samples then give a contribution = 0.

Total PDF is combination of discrete and continuous sampling.
Generate samples as above, then the total PDF of all samples generated is:
$p(\Psi) = q_1 p_1(\Psi) + q_2 p_2(\Psi)$ $(q_1 + q_2 = 1)$

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{L(\Psi_i)\left(\frac{\rho_d}{\pi} + \frac{\rho_s(n+2)}{2\pi}\cos^n(\Psi_i, \Theta_s)\right)\cos(n_x, \Psi_i)}{q_1 p_1(\Psi_i) + q_2 p_2(\Psi_i)}$$

Interesting choices:

$p_1(\Psi) = \frac{1}{2\pi}$;

$p_2(\Psi) = \frac{n+1}{2\pi}\cos^n(\Psi, \Theta_s)$ (no resampling of directions under surface, so full lobe)

$$q_1 = \frac{2\rho_d}{2\rho_d + \frac{(n+2)}{(n+1)}\rho_s}; \; q_2 = \frac{\frac{(n+2)}{(n+1)}\rho_s}{2\rho_d + \frac{(n+2)}{(n+1)}\rho_s}$$

then the above estimator is written as: $\langle I \rangle = \frac{2\rho_d + \frac{(n+2)}{(n+1)}\rho_s}{N} \sum_{i=1}^{N} L(\Psi_i)\cos(n_x, \Psi_i)$

(directions under the surface in the specular lobe evaluate to 0).

(67) Modified Phong-BRDF - Blinn Variant

(see notations page 32)

$f_r(\Theta_i \leftrightarrow \Theta_r) = k_d + k_s \cos^n \theta_h$

(68) Cook-Torrance-BRDF

(see notations page 32)

$$f_r(\Theta_i \leftrightarrow \Theta_r) = \frac{F(\beta)}{\pi} \cdot \frac{D(\theta_h)G}{\cos\theta_i \cos\theta_r}$$

β : angle between Θ_i and Θ_h (equal to angle between Θ_r and Θ_h)

$F(\beta)$: Fresnel Reflectance (see 58)

$D(\theta_h)$: Facet distribution

G : Geometric masking term

$$D(\theta) = \frac{1}{m^2 \cos^4\theta} e^{-\left[\frac{\tan\theta}{m}\right]^2} \quad (m \text{ is material-parameter})$$

$$G = min\left\{1, \frac{2\cos\theta_h \cos\theta_i}{\cos\beta}, \frac{2\cos\theta_h \cos\theta_r}{\cos\beta}\right\}$$

The Cook-Torrance model is described in more detail in R8.

(69) Ward-BRDF

(see notations page 32)

a. Isotropic Gaussian Model: $f_r(\Theta_i \leftrightarrow \Theta_r) = \frac{\rho_d}{\pi} + \rho_s \cdot \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \cdot \frac{\exp(-\tan^2\theta_h / \alpha^2)}{4\pi\alpha^2}$

α is standard deviation (RMS) of the surface slope.

Energy conservation: $\rho_d + \rho_s \le 1$

Normalization $1/4\pi\alpha^2$ is accurate as long α is not much greater than 0.2 (surface is then mostly diffuse).

Proportional sampling of specular part (is already normalized as a pdf):

$$\theta_h = \text{atan}(\alpha\sqrt{-\log(r_1)})$$
$$\varphi = 2\pi r_2$$

b. Anisotropic Elliptical Gaussian Model:

$$f_r(\Theta_i \leftrightarrow \Theta_r) = \frac{\rho_d}{\pi} + \rho_s \cdot \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \cdot \frac{\exp(-\tan^2\theta_h(\cos^2\varphi_h/\alpha_x^2 + \sin^2\varphi_h/\alpha_y^2))}{4\pi\alpha_x\alpha_y}$$

α_x and α_y are are the standard deviation of the surface slope in the x and y direction at the tangent plane.

Energy conservation: $\rho_d + \rho_s \le 1$ and $\alpha_x^2 \ll 1$ and $\alpha_y^2 \ll 1$

Proportional sampling of specular part (is already normalized as a pdf):

$$\theta_h = \text{atan}\left(\sqrt{\frac{-\log(r_1)}{\cos^2\varphi/\alpha_x^2 + \sin^2\varphi/\alpha_y^2}}\right)$$
$$\varphi = \text{atan}\left(\frac{\alpha_x}{\alpha_y}\tan(2\pi r_2)\right)$$

See R46 for a more complete description of the Ward BRDF.

(70) Lafortune-BRDF

$$f_r(\Theta_i \leftrightarrow \Theta_r) = \sum_k (C_{x,k} x_i x_r + C_{y,k} y_i y_r + C_{z,k} z_i z_r)^{n_k}$$

which can also be written as:

$$f_r(\Theta_i \leftrightarrow \Theta_r) = \sum_k C_k(\Theta_i)(\Theta_a \cdot \Theta_r)^{n_k}$$

where:

$$\Theta_a = \frac{(C_{x,k} x_i, C_{y,k} y_i, C_{z,k} z_i)}{\sqrt{C_{x,k}^2 x_i^2 + C_{y,k}^2 y_i^2 + C_{z,k}^2 z_i^2}}$$

$$C_k(\Theta_i) = (C_{x,k}^2 x_i^2 + C_{y,k}^2 y_i^2 + C_{z,k}^2 z_i^2)^{\frac{n_k}{2}}$$

Therefore, the BRDF is a combination of cosine-lobes, each centered around a different axis Θ_a

Proportional sampling:

1. Integrate different cosine-lobes over spherical digons using (31b)
2. Select cosine-lobe proportional to the above computed values.
3. Generate direction according to this cosine-lobe (38)

See R25 for a complete description.

C. BRDF MEASUREMENTS

(71) Cornell Measurements

The Program of Computer Graphics, Cornell University, has a number of measurements online:

`http://www.graphics.cornell.edu/online/measurements/`

IX. Rendering Equation and Global Illumination Formulations

A. Radiance Transport Formulations

All formulations in this section take radiance as the main transport quantity. For the sake of clarity, the wavelength dependency of radiance values is implicitly assumed in all equations.

(72) Rendering Equation (Radiance), integration over incoming hemisphere

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} L(x \leftarrow \Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi) d\omega_\Psi$$

Direct illumination only: $L(x \to \Theta) = \int_{\Omega_x} L_e(r(x, \Psi) \to -\Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi) d\omega_\Psi$

(73) Rendering Equation (Radiance), integration over all surfaces in the scene

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_A L(z \to \overline{zx}) f_r(x, \Theta \leftrightarrow \overline{zx}) G(x, z) V(x, z) dA_z$$

$$G(x, z) = \frac{\cos(n_x, \Theta)\cos(n_z, -\Theta)}{\|x - z\|^2}$$

(many authors include $V(x, z)$ in the definition of $G(x, z)$, but in this document they are kept separate for clarity).

Direct illumination only: see 74.

(74) Direct Illumination Equation (Radiance), integration over all light sources

(computing direct illumination using various sampling techniques for shadow rays: page 50)

L_i: all light sources $(i = 1 \ldots N_L)$ in the scene.

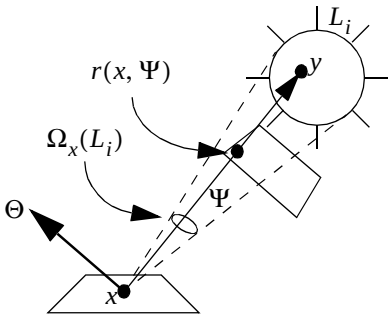
A_{L_i}: area of light source L_i.

$\Omega_x(L_i)$: solid angle subtended by light source L_i w.r.t. x.

$\Omega_x^{vis}(L_i)$: visible solid angle subtended by light source L_i w.r.t. x.

y: intersection of ray (x, Ψ) and light source L_i

N_L: number of light sources in the scene



(74a) Integration over the area of all light sources:

$$L(x \to \Theta) = L_e(x \to \Theta) + \sum_{i=1}^{N_L} \int_{A_{L_i}} L_e(y \to \overline{yx}) f_r(x, \Theta \leftrightarrow \overline{xy}) G(x, y) V(x, y) dA_y$$

(74b) Integration over solid angles subtended by light sources:

$$L(x \to \Theta) = L_e(x \to \Theta) + \sum_{i=1}^{N_L} \int_{\Omega_x(L_i)} L_e(y \to -\Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi) V(x, y) d\omega_\Psi$$

(74c) Integration over visible solid angles subtended by light sources:

(in this case $y \equiv r(x, \Psi)$)

$$L(x \to \Theta) = L_e(x \to \Theta) + \sum_{i=1}^{N_L} \int_{\Omega_x^{vis}(L_i)} L_e(y \to -\Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi) d\omega_\Psi$$

(75) Continuous Radiosity Equation: diffuse reflection, diffuse light sources, integration over surfaces

If all surfaces are diffuse reflectors and light sources are diffuse emitters, radiance values are independent of direction and can be expressed by the hemispherical radiometric quantities:

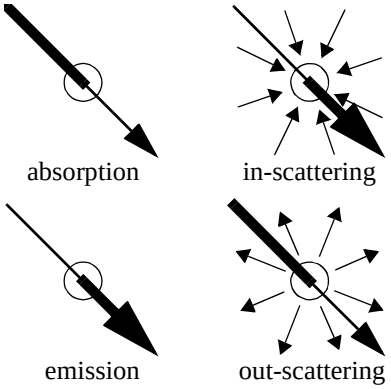
$$B(x) = \pi L(x)$$
$$B_e(x) = \pi L_e(x)$$
$$\rho(x) = \pi f_r(x)$$

$$B(x) = B_e(x) + \rho(x) \int_A \frac{B(z) G(x, z) V(x, z)}{\pi} dA_z$$

(76) Participating medium

See also descriptions in (R7, p. 325) and (R33, p. 174).
There are 4 phenomena that affect the radiance along a path in a medium:

absorption: the medium absorbs light
emission: the medium emits light
in-scattering: light scatters into the direction of travel
out-scattering: light scatters away from the direction of travel



Change in radiance along a path:

$$\frac{dL(s)}{ds} = -\kappa_t L(s) + \kappa_a L_e(s) + \kappa_s \int_\Omega L_i(\Theta) f(\Theta) d\omega_\Theta$$

$-\kappa_t L(s)$: absorption + out-scattering
$\kappa_a L_e(s)$: emission
$\kappa_s \int_\Omega L_i(\Theta) f(\Theta) d\omega_\Theta$: in-scattering

with:

$\kappa_a(s)$: absorption coefficient (dimension: *1/meter*), fraction by which radiance is reduced per unit length due to absorption in the medium.
$\kappa_s(s)$: scattering coefficient (dimension: *1/meter*), fraction by which radiance is reduced per unit length due to out-scattering.
$\kappa_t(s) = \kappa_a(s) + \kappa_s(s)$: extinction coefficient.

$f(\Theta)$: phase function, describing fraction of radiance arriving from direction Θ that is in-scattered along the path. If the medium is isotropic: $f_{\text{isotropic}} = \frac{1}{4\pi}$

Solving the above differential equation:

$$L(s) = L(0)\tau(0, s) + \int_0^s \tau(s', s)J(s')\kappa_t(s')ds'$$

with:

$\tau(s_1, s_2) = \exp\left(-\int_{s_1}^{s_2} \kappa_t(s)ds\right)$: transmittance function. If the medium is isotropic, $\kappa_t(s)$ is constant and thus

$\tau(s_1, s_2) = e^{-\kappa_t|s_1 - s_2|}$

$J(s)$: source function, describing in-scattering and emission.

$$J(s) = (1 - R(s))L_e(s) + R(s)\int_\Omega L_i(\Theta)f(\Theta)d\omega_\Theta$$

$R(s) = \frac{\kappa_s(s)}{\kappa_t(s)} = \frac{\kappa_s(s)}{\kappa_a(s) + \kappa_s(s)}$: scattering albedo of the medium

(77) Participating medium, no scattering

Equations in 76 apply, with $\kappa_s(s) = 0$ and $\kappa_a(s) = \kappa_t(s)$

$$L(s) = L(0)\tau(0, s) + \int_0^s L_e(s')\tau(s', s)\kappa_t(s')ds'$$

and if $\kappa_t(s)$ and $L_e(s)$ constant:

$L(s) = L(0)e^{-\kappa_t s} + L_e(1 - e^{-\kappa_t s})$ (can be used as a simple fog-model)

B. DUAL TRANSPORT FORMULATION

Light transport can also be formulated by using the adjoint equations. The adjoint transport quantity is called *importance* or *potential function* by various authors, and an often used notation is W. An intuitive way of thinking about the potential function is to consider it an incident function in combination with radiance as an exitant function.

(78) Relationship between Flux, Radiance, Potential

See R12 for a more complete description.

Consider a set S of surface points and associated directions $S = A_s \times \Omega_x \subset A \times \Omega$. The exitant flux for S can be written as:

$$\Phi(S) = \int_{A}\int_{\Omega} L(x \rightarrow \Theta) W_e(x \leftarrow \Theta) \cos(\Theta, n_x) d\omega_\Theta dA_x$$

with $W_e(x \leftarrow \Theta)$ the initial potential, defined as:

$$W_e(x \leftarrow \Theta) = \begin{cases} 1 & \text{if } (x, \Theta) \in S \\ 0 & \text{if } (x, \Theta) \notin S \end{cases}$$

The above integral can also be formally defined as an inner product: $\Phi(S) = \langle L, W_e \rangle$.

Using the dual formulation, $\Phi(S)$ can also be written as an integral over all light sources, or formally:

$$\Phi(S) = \int_{A}\int_{\Omega} L_e(x \rightarrow \Theta) W(x \leftarrow \Theta) \cos(\Theta, n_x) d\omega_\Theta dA_x = \langle L_e, W \rangle$$

with

$$W(x \leftarrow \Theta) = W_e(x \leftarrow \Theta) + \int_{\Omega_{r(x,\Theta)}} W(r(x, \Theta) \leftarrow \Psi) f_r(r(x, \Theta), \Psi \leftrightarrow \Theta) \cos(n_{r(x,\Theta)}, \Psi) d\omega_\Psi$$

$W(x \leftarrow \Theta)$ is dimensionless and has the same transport properties as radiance (invariant along straight lines).

X. Form Factors

Form Factor: the fraction of uniform diffuse radiant energy leaving one surface that is incident upon a second surface.

Form Factor Algebra: mathematical relations between form factors

Notation: The 'sender' surface is written as the first index, the 'receiver' surface as the second. An arrow indicates the 'energy flow'. This notation is consistent with other notations in literature, where the arrow usually is not used.

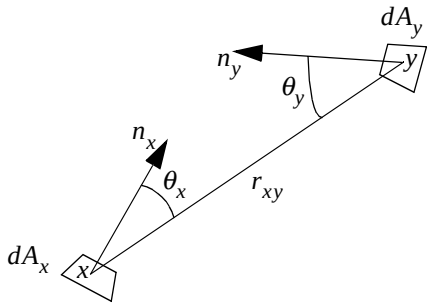
Form Factors are usually treated very extensively in books dealing with thermal radiation heat transfer (R32).

A good on-line resource for analytical avaluations of form factors: "*A Catalog of Radiation Heat Transfer Configuration Factors*" by John R. Howell: `http://www.me.utexas.edu/~howell/`

Global illumination context: computing form factors is one of the main elements of radiosity algorithms. Radiosity algorithms based on some variant of particle tracing often implicitly compute form factors without storing them. Direct illumination computations in ray tracing using shadow-rays also often implicitly computes a point-to-area form factor.

A. General Expressions and Properties

(79) Differential element to differential element Form Factor



$$dF_{dA_x \to dA_y} = \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} dA_y = \frac{G(x,y)}{\pi} dA_y$$

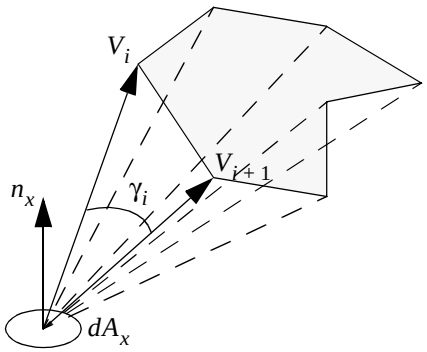
With visibility term:

$$dF_{dA_x \to dA_y} = \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} V(x,y)\, dA_y = \frac{G(x,y)V(x,y)}{\pi} dA_y$$

(80) Differential element to element Form Factor

$$F_{dA_x \to j} = \int_{A_j} \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} V(x,y)\, dA_y = \int_{A_j} \frac{G(x,y)V(x,y)}{\pi} dA_y$$

(81) Differential element to polygon Form Factor; full visibility



$$F_{dA_x \to j} = \frac{1}{2\pi} \sum_{i=1}^{E} n_x \cdot \Gamma_i \quad (E = \text{number of edges or vertices of the polygon})$$

where $\Gamma_i = V_i \otimes V_{i+1}$ and magnitude of $\Gamma_i = \gamma_i$.

See also R3.

(82) Element to element Form Factor

$$F_{i\to j} = \frac{1}{A_i}\int_{A_iA_j}\int \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} V(x,y)\,dA_y\,dA_x = \frac{1}{A_i}\int_{A_iA_j}\int \frac{G(x,y)}{\pi} V(x,y)\,dA_y\,dA_x$$

$\Omega_x(j)$ is subtended solid angle covering surface j as seen from x (see 26 and 27; also 9):

$$F_{i\to j} = \frac{1}{A_i}\int_{A_i}\int_{\Omega_x(j)} \frac{\cos\theta_x}{\pi} M(x^{\Theta}, j)\,d\omega_{\Theta}\,dA_x$$

$$F_{i\to j} = \frac{1}{A_i}\int_{A_i}\int_{\Omega_x^{vis}(j)} \frac{\cos\theta_x}{\pi} d\omega_{\Theta}\,dA_x$$

(83) Element to element Form Factor; full visibility; Stoke's Theorem

$$F_{i\to j} = \frac{1}{2\pi A_i}\oint_{C_i}\oint_{C_j} (\ln r\,dx_i\,dx_j + \ln r\,dy_i\,dy_j + \ln r\,dz_i\,dz_j)$$

An analytic solution for this integral, given any two polygons, was derived by Schröder and Hanrahan (R29).

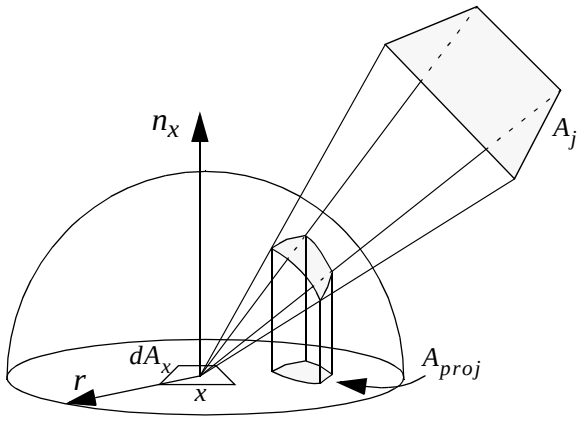
(84) Form Factor Algebra

a. $A_i F_{i\to j} = A_j F_{j\to i}$

b. In a closed environment: $\sum_j F_{i\to j} = 1$

(85) Nusselt's Analog (projection on a disk)

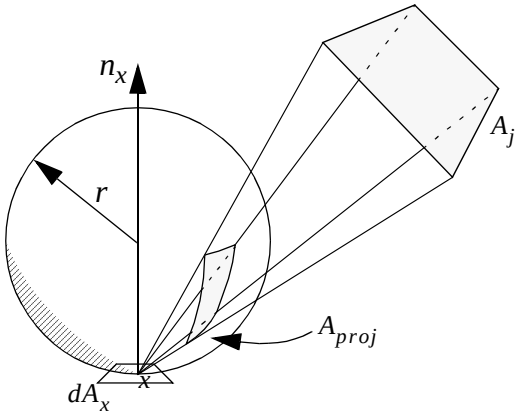
$$F_{dA_x\to j} = \frac{A_{proj}}{\pi r^2}$$



(86) Projection on a sphere

$$F_{dA_x \to j} = \frac{A_{proj}}{4\pi r^2}$$

See also R39.



B. COMPUTING FORM FACTORS USING MONTE CARLO INTEGRATION[1]

(87) Uniform area sampling on both surfaces

MC integration applied to (82 - 1st equation):

$$p(x,y) = \frac{1}{A_i \cdot A_j}$$

$$\langle F_{i \to j} \rangle = \frac{1}{N} \sum_{k=1}^{N} \frac{A_j G(x_k, y_k) V(x_k, y_k)}{\pi}$$

$$\sigma^2[\langle F_{i \to j} \rangle] = \frac{1}{N}\left(\frac{A_j}{A_i} \int_{A_i} \int_{A_j} \frac{G^2(x,y) V^2(x,y)}{\pi^2} dA_y dA_x - F_{i \to j}^2 \right)$$

(88) Uniform area sampling + uniform solid angle sampling

MC integration applied to (82 - 2nd equation)

$$p(x,\Theta) = \frac{1}{A_i \cdot \Omega_x(j)}$$

$$\langle F_{i \to j} \rangle = \frac{1}{N} \sum_{k=1}^{N} \frac{\cos\theta_{x_k} \Omega_{x_k}(j) M(x_k^{\Theta_k}, j)}{\pi}$$

$$\sigma^2[\langle F_{i \to j} \rangle] = \frac{1}{N}\left(\int_{A_i} \int_{\Omega_x(j)} \frac{\cos^2\theta_x \Omega_x(j) M^2(x^{\Theta}, j)}{A_i \pi^2} d\omega_{\Theta} dA_x - F_{i \to j}^2 \right)$$

By using visibility culling or rejection sampling, sampling and integration can be limited to $\Omega_x^{vis}(j)$.

1. Extensive coverage of computing form factors using MC integration can be found in the Ph.D. thesis of Ph. Bekaert (R5). This section is partly based on that text.

(89) Uniform area sampling + cosine-weighted solid angle sampling

MC integration applied to (82 - 2nd equation)

$$p(x,\Theta) = \frac{1}{A_i} \cdot \frac{\cos\theta_x}{\int\limits_{\Omega_x(j)} \cos\theta_x d\omega_\Theta}$$

$$\langle F_{i\to j}\rangle = \frac{1}{N}\sum_{k=1}^{N} \frac{M(x_k^{\Theta_k}, j)\int\limits_{\Omega_x(j)} \cos\theta_x d\omega_\Theta}{\pi}$$

$$\sigma^2[\langle F_{i\to j}\rangle] = \frac{1}{N}\left(\int\limits_{A_i}\int\limits_{\Omega_x(j)} \frac{\cos\theta_x M^2(x^\Theta, j)\int\limits_{\Omega_x(j)} \cos\theta_x d\omega_\Theta}{A_i\pi^2} d\omega_\Theta dA_x - F_{i\to j}^2\right)$$

By using visibility culling or rejection sampling, sampling and integration can be limited to $\Omega_x^{vis}(j)$.

(90) Uniform area sampling + cosine-weighted hemisphere sampling

Directions are sampled over the entire hemisphere Ω_x and do not need to point to surface element j. Just count the number of rays arriving at j to get an estimate for $F_{i\to j}$.

$$p(x,\Theta) = \frac{1}{A_i} \cdot \frac{\cos\theta_x}{\pi}$$

$$\langle F_{i\to j}\rangle = \frac{1}{N}\sum_{k=1}^{N} M(x_k^{\Theta_k}, j)$$

$$\sigma^2[\langle F_{i\to j}\rangle] = \frac{1}{N}F_{i\to j}(1 - F_{i\to j})$$

This sampling procedure can be used to compute all $F_{i\to j}$ simultaneously for a given i using the same set of rays (x_k, Θ_k):

```
for all patches j: F_i→j = 0;
for k = 1,...,N do
     generate a uniformly distributed random point x on patch i
     generate a cosine distributed direction Θ
     j = patch hit by the ray (x, Θ)
     F_i→j += 1/N
```

(91) Global Lines

Global lines methods compute all form factors in the scene without explicitly sampling points on the surface patches[1]. Various techniques can be used to compute the global lines:

a. Construct a bounding sphere for the scene. Generate each global line by connecting 2 uniform random points on the surface of the sphere.

1. An overview of global lines in the context of radiosity can be found in the Ph.D. Thesis of Mateu Sbert (R28):

b. Generate a random direction Θ on the unit sphere. Project the scene on a plane perpendicular to Θ. Generate a uniformly distributed random surface point x in the projection. (x, Θ) is a global line.

All global lines are intersected with all surfaces in the scene, resulting in several spans on each line. The probability that a line intersects a surface is proportional to the area of that surface.

$F_{i \to j}$ is estimated by N_{ij} / N_i, with N_{ij} the number of spans connecting surface patches i and j, and N_i the number of spans starting from surface patch i.

XI. Radiosity System & Algorithms

The term ‘Radiosity’ is used in various ways in photorealistic image synthesis: as the radiometric quantity describing incident energy integrated over the hemisphere; but also to indicate a specific class of algorithms that compute an energy distribution in the scene. Although there is no formal definition of what a ‘radiosity algorithm’ actually is, it is usually assumed that one means a finite-element solution, computing diffuse light interaction only.

Most radiosity literature uses Galerkin Radiosity, a projection method to transform the continuous radiosity equations (75) into a set of linear equations. Unless otherwise indicated, Galerkin radiosity is used in this section.

Notations:

B_i : constant radiosity of patch i (see 49, 51) - expressed in Watt/m^2

B_{ei} : self-emitted constant radiosity of patch i - expressed in Watt/m^2

ρ_i : diffuse bihemispherical reflectance of patch i (see 65) - dimensionless

$F_{i \to j}$: form factor between patch i and j (see 82) - dimensionless

P_i : power of patch i - expressed in Watt $P_i = A_i B_i$

A_i : surface area of patch i - expressed in m^2

(92) System of radiosity equations, constant basis functions

$$\forall \text{patches i:} \qquad B_i = B_{ei} + \rho_i \sum_j F_{i \to j} B_j$$

$$\begin{bmatrix} 1-\rho_1 F_{1\to1} & -\rho_1 F_{1\to2} & \cdots & -\rho_1 F_{1\to n} \\ -\rho_2 F_{2\to1} & 1-\rho_2 F_{2\to2} & \cdots & -\rho_2 F_{2\to n} \\ \cdots & \cdots & \cdots & \cdots \\ -\rho_n F_{n\to1} & -\rho_n F_{n\to2} & \cdots & 1-\rho_n F_{n\to n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdots \\ B_n \end{bmatrix} = \begin{bmatrix} B_{e1} \\ B_{e2} \\ \cdots \\ B_{en} \end{bmatrix}$$

(93) System of power equations, constant basis functions

$$\forall \text{patches i:} \qquad P_i = P_{ei} + \rho_i \sum_j F_{j \to i} P_j$$

$$\begin{bmatrix} 1-\rho_1 F_{1\to1} & -\rho_1 F_{2\to1} & \cdots & -\rho_1 F_{n\to1} \\ -\rho_2 F_{1\to2} & 1-\rho_2 F_{2\to2} & \cdots & -\rho_2 F_{n\to2} \\ \cdots & \cdots & \cdots & \cdots \\ -\rho_n F_{1\to n} & -\rho_n F_{2\to n} & \cdots & 1-\rho_n F_{n\to n} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \cdots \\ P_n \end{bmatrix} = \begin{bmatrix} P_{e1} \\ P_{e2} \\ \cdots \\ P_{en} \end{bmatrix}$$

(94) Discretizing the continuous radiosity equation[1]

Continuous radiosity equation (see 75): $B(x) = B_e(x) + \rho(x) \int_A \frac{B(z) G(x,z) V(x,z)}{\pi} dA_z$

1. Based on chapter 3 of R7.

Approximation of $B(x)$: $B(x) \approx \hat{B}(x) = \sum_{i=1}^{N} B_i N_i(x)$

Residual (substitute $\hat{B}(x)$ in continuous equation): $r(x) = \hat{B}(x) - B_e(x) - \rho(x)\int_A \frac{\hat{B}(z)G(x,z)V(x,z)}{\pi} dA_z$

General approach: pick N weighting functions $W_i(x)$. The norm of residual: $|r(x)| = \sum_{i=1}^{N} |\langle r(x), W_i(x)\rangle|$.

Each of the terms $\langle r(x), W_i(x)\rangle = \int_A r(x)W_i(x)dA_x$ can now be set to be zero, resulting in a linear equation for each $W_i(x)$.

(94a) Point Collocation

$W_i(x) = \delta(x - x_i)$ (residual is zero for selected number of points x_i)

then: $\forall i$: $\sum_{j=1}^{N} K_{ij}B_j - B_e(x_i) = 0$ with $K_{ij} = N_j(x_i) - \frac{\rho(x_i)}{\pi}\int_A N_j(z)G(x_i, z)V(x_i, z)dA_z$

(94b) Galerkin

$W_i(x) = N_i(x)$

then: $\forall i$: $\sum_{j=1}^{N} K_{ij}B_j - \int_A B_e(x)N_i(x)dA_x = 0$

with $K_{ij} = \int_A N_i(x)N_j(x)dA_x - \int\int_{AA} \rho(x)N_i(x)N_j(z)\frac{G(x,z)V(x,z)}{\pi} dA_z dA_x$

Choosing $N_i(x) = \begin{cases} 1 & \text{if } x \text{ belongs to surface element } i \\ 0 & \text{otherwise} \end{cases}$ results in the classic radiosity formulation (92 or 93).

(95) Basic Relaxation Algorithm[1]

Most radiosity algorithms are relaxation methods, that solve the radiosity system through a series of iterations. This item describes the basic principle of all relaxation algorithms.

Linear system to solve: $Ax = e$ or $\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} e_1 \\ \dots \\ e_n \end{bmatrix}$

Subsequent iterations: $x^{(0)}, x^{(1)}, x^{(2)}, \dots$

1. A very good overview of different relaxation techniques for solving the radiosity system can be found in the Ph.D. thesis of Philippe Bekaert (R5).

Correction vectors $\Delta x^{(k)}$: $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$

Residu vectors $r^{(k)}$ (measure for error): $r^{(k)} = e - Ax^{(k)}$, and therefore $r^{(k+1)} = r^{(k)} - A\Delta x^{(k)}$

Basic relaxation algorithm:

```
Choose initial x(0);
r(0) = b - A.x(0);
for k = 0, 1, ... until convergence do
     compute Δx(k) (based on x(k), r(k), other information)
     x(k+1) = x(k) + Δx(k);
     r(k+1) = r(k) - A.Δx(k);
```

(96) Gauss-Seidel iteration

Each component of the solution is updated in turn: $x_s \leftarrow \left(e_s - \sum_{j \neq s} a_{sj} x_j \right) / a_{ss}$

Correction vector: $\Delta x_i \leftarrow \delta_{is} r_s / a_{ss}$ (all components 0 except s).

Residu vector: $r_i \leftarrow r_i - a_{is} r_s / a_{ss}$ (component s equals 0).

Applied to (93) ($a_{ss} = 1$ if $F_{s \to s} = 1$):

$$P_s \leftarrow P_{es} + \rho_s \sum_{j \neq s} F_{j \to s} P_j$$

Algorithm:

```
for all patches j: Pj = Pej;
cycle through patches until convergence (selected patch = s)
     Ps = Pes;
     for all patches j
          Ps += ρs.Fj→s.Pj;
```

Incremental Gauss-Seidel: compute residu vector explicitly and increment solution vector instead of replacing previous solution.

XII. Radiosity Extensions

(97) Clustering - Equivalent extinction coefficient

When clustering objects, an equivalent isotropic extinction coefficient can be computed for the cluster (R35), in an analogy with a participating medium (76).

$$\kappa_t = \frac{A}{4V}$$

where A is the total surface area of objects in the cluster and V is the volume of the cluster.

Directional extinction (R34):

$$\kappa_t(\Theta) = \frac{A^{\perp}(\Theta)}{V}$$

where $A^{\perp}(\Theta)$ is the total projected area in direction Θ.

(98) Final Gathering

A **final gathering step** is used to make a high-quality rendering of a radiosity solution, using a per-pixel radiance computation. It usually involves computing a radiance value for the visible surface point through each pixel, using the radiosity values stored on the surface patches.

Suppose $\bar{L}$ is the pre-computed (diffuse) radiosity solution, then the radiance for surface point x in direction Θ can be approximated as (see section IX for basic rendering equation):

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} \bar{L}(x \leftarrow \Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi) d\omega_{\Psi} \quad \text{(hemisphere integration)}$$

or:

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{A} \bar{L}(z \rightarrow \overline{zx}) f_r(x, \Theta \leftrightarrow \overline{zx}) G(x, z) V(x, z) dA_z \quad \text{(surface integration)}$$

The integral in either of the two above formulations can be approximated by an appropriate sampling technique, thereby ‘gathering’ radiance from the other surfaces. Many different variants are possible, depending on how much information was stored with the radiosity solution.

Variants: Uniform hemisphere sampling; cosine-weighted hemisphere sampling; uniform area sampling; area sampling proportional to solid angle; area sampling proportional to cosine weighted angle; area sampling proportional to $\bar{L}$; area sampling sampling proportional to stored (linked) form factors. See (R41, R4) for an overview.

XIII. Pixel-driven Path Tracing Algorithms

Many rendering algorithms use (random) paths in object space to compute illumination values. If these paths are directly used to compute the illumination of the pixels, one has a specific class of algorithms, of which classic ray tracing is the best-known. Other algorithms include stochastic ray tracing; light tracing (also called backward ray tracing or particle tracing); and bidirectional ray tracing.

See also: Ray-object intersections: chapter on Geometry (chapter III)

A. DIRECT ILLUMINATION USING SHADOW-RAYS

Most algorithms that compute direct illumination using shadow rays apply a specific numerical integration scheme directly derived from the equations described in 74.

When there are multiple light sources in the scene, the complete sampling procedure can usually be divided in two parts: a discrete selection of one of the light sources, followed by the generation of a shadow-ray for the selected light source. Various choices for each of these sampling procedures produce different estimators. The obvious alternative is to split the direct illumination integral in seperate integrals, one for each light source, and compute the illumination due to each light source seperately.

A good general overview for direct lighting calculations using Monte Carlo techniques can be found in R30.

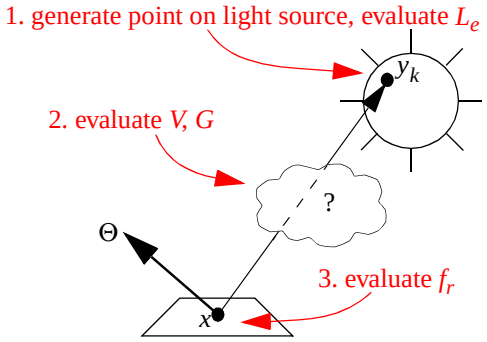
(99) Single light source, uniform sampling of light source area

Apply Monte Carlo integration to integral 74a:

$$p(y) = 1/A_L$$

$$\langle L(x \rightarrow \Theta) \rangle = \frac{A_L}{N} \sum_{k=1}^{N} L_e(y_k \rightarrow \overline{y_k x}) f_r(x, \overline{x y_k} \leftrightarrow \Theta) G(x, y_k) V(x, y_k)$$

Graphical representation:



(100) Single light source, uniform sampling of solid angle subtended by light source

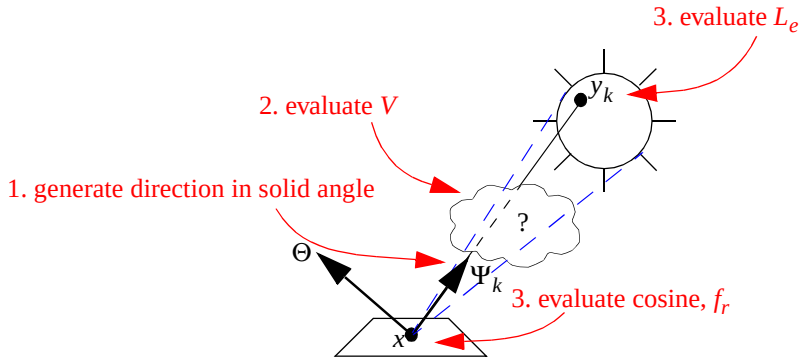
Apply Monte Carlo integration to integral 74b:

$$p(\Psi) = 1/\Omega_x(L)$$

$$\langle L(x \rightarrow \Theta) \rangle = \frac{\Omega_x(L)}{N} \sum_{k=1}^{N} L_e(y_k \rightarrow -\Psi_k) f_r(x, \Psi_k \leftrightarrow \Theta) \cos(n_x, \Psi_k) V(x, y_k)$$

with y_k = intersection of ray (x, Ψ_k) and light source L

Graphical representation



Solid angle sampling of the light source usually yields lower variance as opposed to area sampling (99), due to the abscence of a cosine and inverse distance factor in the estimator.

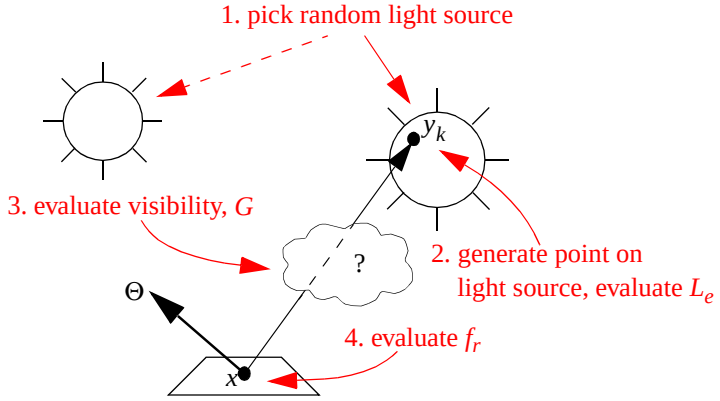
(101) Multiple light sources, uniform random source selection, uniform sampling of light source area

Apply 2 sampling steps for Monte Carlo evaluation of integral 74a:

1. Select a light source i using a discrete pdf $p_L(i) = 1/N_L$
2. Select a surface point y on light source i using a uniform pdf $p(y) = 1/A_{L_i}$

$$\langle L(x \rightarrow \Theta) \rangle = \frac{N_L}{N} \sum_{k=1}^{N} A_{L_k} L_e(y_k \rightarrow \overline{y_k x}) f_r(x, \overline{x y_k} \leftrightarrow \Theta) G(x, y_k) V(x, y_k)$$

Graphical representation:



(102) Multiple light sources, uniform random source selection, uniform sampling of light source solid angle

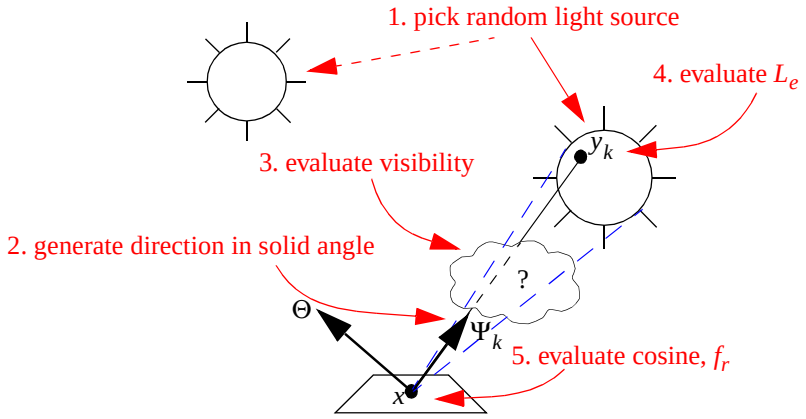
Apply 2 sampling steps Monte Carlo evaluation of integral 74b:

1. Select a light source i using a discrete pdf $p_L(i) = 1/N_L$
2. Select a random direction Ψ within solid angle subtended by light source i using pdf $p(\Psi) = 1/\Omega_x(L)$

$$\langle L(x \rightarrow \Theta) \rangle = \frac{N_L}{N} \sum_{k=1}^{N} \Omega_x(L_i) L_e(y_k \rightarrow -\Psi_k) f_r(x, \Psi_k \leftrightarrow \Theta) \cos(n_x, \Psi_k) V(x, y_k)$$

with y_k = intersection of ray (x, Ψ_k) and light source L_i

Graphical representation



Solid angle sampling of the light source usually yields lower variance as opposed to area sampling (101), due to the absence of a cosine and inverse distance factor in the estimator.

(103) Multiple light sources, non-uniform random source selection, non-uniform sampling of light source area

Apply 2 sampling steps during the Monte Carlo evaluation of 74a:
1. Select a light source *i* using a discrete pdf $p_L(i)$
2. Select a surface point y on light source *i* using a pdf $p(y)$

$$\langle L(x \rightarrow \Theta) \rangle = \frac{1}{N} \sum_{k=1}^{N} \frac{L_e(y_k \rightarrow \overline{y_k x}) f_r(x, \overline{x y_k} \leftrightarrow \Theta) G(x, y_k) V(x, y_k)}{p_L(i) p(y_k)}$$

Some interesting choices:

$p_L(i)$ can be chosen proportional to the power emitted by each light source (bright light sources get sampled more often). E.g. for diffuse light sources: $\Phi_i = L_{ei} A_i \pi$ and thus $p_L(i) = \Phi_i / \Phi_{tot}$ where Φ_{tot} is the total emitted power of all light sources.

In this case: $\langle L(x \rightarrow \Theta) \rangle = \frac{\Phi_{tot}}{\pi N} \sum_{k=1}^{N} \frac{f_r(x, \overline{x y_k} \leftrightarrow \Theta) G(x, y_k) V(x, y_k)}{A_i p(y_k)}$

and if $p(y) = 1/A_i$: $\langle L(x \rightarrow \Theta) \rangle = \frac{\Phi_{tot}}{\pi N} \sum_{k=1}^{N} f_r(x, \overline{x y_k} \leftrightarrow \Theta) G(x, y_k) V(x, y_k)$

(104) Various strategies for computing direct illumination due to multiple light sources

Various schemes for speeding up the computation of direct illumination due to multiple light sources have been proposed by several authors. These usually involve a smart choice for the pdf $p_L(i)$ in 103, taking into account light source brightness, visibility, etc.

G. Ward (R45) proposes to rank all light sources according to their potential illumination contribution to the point to be shaded (that is, without taking into account visibility). Then, the illumination of the light source ranked first is computed first (e.g. using 99), followed by the light source ranked second, and so on; until the potential contribution of the remaining light sources falls below a certain threshold. The illumination due to the remaining light sources is then computed using e.g. 101 or 103. Alternatively, one can estimate the contribution of the remaining light sources by multiplying their potential contribution by the average hit ratio of shadow rays

send to these light sources so far (e.g. when computing previous points). The accuracy goal can be relaxed by a user-specified parameter.

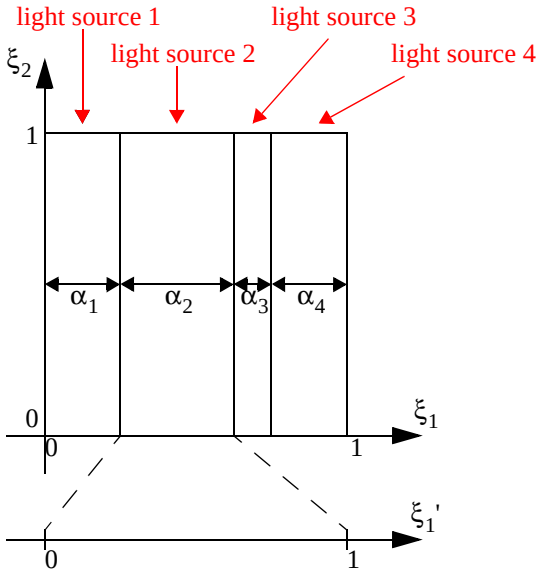
In (R30), an approach is presented where the scene is divided in octree cells. Each cell maintains a list of 'important' light sources for the points in that cell, based on intensity of the light source, distance, and reflectivity. When selecting a light source, the important light sources are sampled proportional to their potential contribution (not taking into account visibility), and all remaining light sources are selected with equal probability. The number of samples used for the important and unimportant light sources can vary. In (R48), an extension using the visibility function to classify light sources is presented.

The same authors also describe the following technique: instead of using two different sampling steps (pick the light source first, then sample within the light source), a single procedure can be used that only uses 2 random numbers. Suppose we want to pick light source i with probability α_i. Generate ξ_1 uniform over $[0, 1]$. Choose light source i if:

$$\sum_{j=1}^{i-1} \alpha_j < \xi_1 < \sum_{j=1}^{i} \alpha_j$$

To sample the light source, use the pair (ξ_1', ξ_2) with $\xi_1' = \left(\xi_1 - \sum_{j=1}^{i-1} \alpha_j\right) / \alpha_i$.

Graphically, this means all light sources are mapped into the $[0, 1] \times [0, 1]$ domain:



Advantage: one less random variable, so stratified sampling becomes more effective; less variance.

(105) Multiple light sources, uniform sampling of hemisphere

The main difference with the previous methods is that shadow rays are not targeted towards light sources, but are generated over the entire hemisphere. Apply Monte Carlo evaluation to direct illumination equation of 72.

$p(\Psi) = 1/2\pi$ (uniform sampling of hemisphere, see 34)

$$\langle L(x \rightarrow \Theta) \rangle = \frac{2\pi}{N} \sum_{k=1}^{N} L_e(r(x, \Psi_k) \rightarrow -\Psi_k) f_r(x, \Psi_k \leftrightarrow \Theta) \cos(n_x, \Psi_k)$$

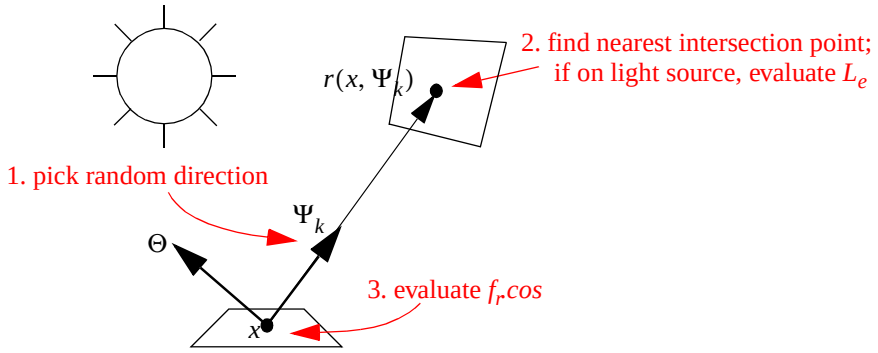
(106) Multiple light sources, non-uniform sampling of hemisphere

The main difference with the previous methods is that shadow rays are not targeted towards light sources, but are generated over the entire hemisphere. Apply Monte Carlo evaluation to direct illumination equation of 72.

Choose a pdf for generating random directions: $p(\Psi)$

$$\langle L(x \rightarrow \Theta) \rangle = \frac{1}{N} \sum_{k=1}^{N} \frac{L_e(r(x, \Psi_k) \rightarrow -\Psi_k) f_r(x, \Psi_k \leftrightarrow \Theta) \cos(n_x, \Psi_k)}{p(\Psi_k)}$$

Graphical representation:



Some interesting choices: uniform sampling, cosine-sampling, brdf sampling, brdf.cosine sampling.

(106a) Multiple light sources, uniform sampling of hemisphere: see 105

(106b) Multiple light sources, cosine-sampling of hemisphere

$p(\Psi) = \cos(n_x, \Psi)/\pi$ (cosine sampling of hemisphere, see 35)

$$\langle L(x \rightarrow \Theta) \rangle = \frac{\pi}{N} \sum_{k=1}^{N} L_e(r(x, \Psi_k) \rightarrow -\Psi_k) f_r(x, \Psi_k \leftrightarrow \Theta)$$

(106c) Multiple light sources, BRDF-sampling of hemisphere

$$p(\Psi) = \frac{f_r(x, \Psi \leftrightarrow \Theta)}{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) d\omega_\Psi}$$

$$\langle L(x \rightarrow \Theta) \rangle = \frac{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) d\omega_\Psi}{N} \sum_{k=1}^{N} L_e(r(x, \Psi_k) \rightarrow -\Psi_k) \cos(n_x, \Psi_k)$$

For a general BRDF, the appropriate sampling function might be very difficult to construct. Rejection sampling can be used (see 8), but might yield very high variance, especially when the BRDF has a narrow peak.

(106d) Multiple light sources, BRDF.cosine-sampling of hemisphere

$$p(\Psi) = \frac{f_r(x, \Psi \leftrightarrow \Theta)\cos(n_x, \Psi)}{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta)\cos(n_x, \Psi)d\omega_\Psi}$$

$$\langle L(x \rightarrow \Theta)\rangle = \frac{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta)\cos(n_x, \Psi)d\omega_\Psi}{N} \sum_{k=1}^{N} L_e(r(x, \Psi_k) \rightarrow -\Psi_k)$$

For a general BRDF, the appropriate sampling function might be very difficult to construct. Rejection sampling can be used (see 8), but might yield very high variance, especially when the BRDF has a narrow peak.

B. RAY TRACING

The term ‘ray tracing’ has been used for different algorithms over the years. In the context of global illumination, ‘ray tracing’ usually implies ‘stochastic ray tracing’, where one can compute a full solution to the rendering equation. The traditional ray tracing algorithm (R47) is then often referred to as ‘classic ray tracing’ or ‘Whitted-style ray tracing’.

Stochastic ray tracing computes the radiance value for a given surface point in a given direction. These points are usually points visible through a pixel, and by averaging such radiance values using an appropriate filter, one can compute the illumination value of a pixel in the image.

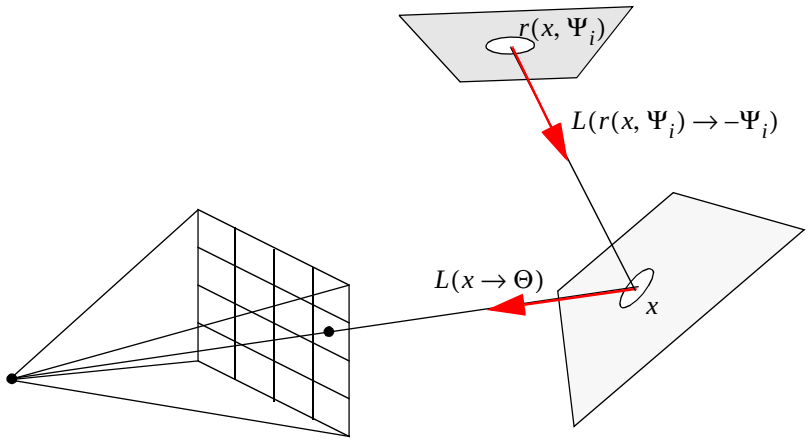
(107) Stochastic ray tracing - general idea

Rendering equation (72): $L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} L(x \leftarrow \Psi) f_r(x, \Psi \leftrightarrow \Theta)\cos(n_x, \Psi)d\omega_\Psi$

Monte Carlo integration using pdf $p(\Psi)$, and invariance property of radiance $(L(x \leftarrow \Psi_i) = L(r(x, \Psi_i) \rightarrow -\Psi_i))$:

$$\langle L(x \rightarrow \Theta)\rangle = L_e(x \rightarrow \Theta) + \frac{1}{N}\sum_{i=1}^{N} \frac{L(r(x, \Psi_i) \rightarrow -\Psi_i) f_r(x, \Psi_i \leftrightarrow \Theta)\cos(n_x, \Psi_i)}{p(\Psi_i)}$$

$L(r(x, \Psi_i) \rightarrow -\Psi_i)$ is evaluated recursively using the same procedure.



(108) Next event estimation (split in direct and indirect term)

Rewrite rendering equation (72,74) as integral over area of all light sources and hemispherical integral of all indirect light (not directly from light sources).

$$L(x \to \Theta) = L_e(x \to \Theta) + L_r(x \to \Theta)$$

$$\begin{aligned} L_r(x \to \Theta) &= \int_{\Omega_x} L(x \leftarrow \Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi) d\omega_\Psi \\ &= \int_{A_L} L_e(y \to \overline{yx}) f_r(x, \Theta \leftrightarrow \overline{xy}) G(x, y) V(x, y) dA_y + \int_{\Omega_x} L_i(x \leftarrow \Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi) d\omega_\Psi \\ &= L_{direct}(x \to \Theta) + L_{indirect}(x \to \Theta) \end{aligned}$$

and $L_i(x \leftarrow \Psi) = L_r(r(x, \Psi_i) \to -\Psi_i)$.

$L_{direct}(x \to \Theta)$ is evaluated using any of the techniques for direct illumination (99 - 106). $L_{indirect}(x \to \Theta)$ is evaluated with Monte Carlo integration:

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

$L_r(r(x, \Psi_i) \to -\Psi_i)$ is evaluated recursively.

Some interesting choices for $p(\Psi)$: uniform sampling, cosine-sampling, brdf sampling, brdf.cosine sampling.

(108a) Uniform sampling of hemisphere

$$\begin{aligned} p(\Psi) &= \frac{1}{2\pi} \\ \langle L_{indirect}(x \to \Theta) \rangle &= \frac{2\pi}{N} \sum_{i=1}^{N} L_r(r(x, \Psi_i) \to -\Psi_i) f_r(x, \Psi_i \leftrightarrow \Theta) \cos(n_x, \Psi_i) \end{aligned}$$

(108b) Cosine-sampling of hemisphere

$p(\Psi) = \cos(n_x, \Psi)/\pi$ (cosine sampling of hemisphere, see 35)

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

(108c) BRDF-sampling of hemisphere

$$p(\Psi) = \frac{f_r(x, \Psi \leftrightarrow \Theta)}{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) d\omega_\Psi}$$

$$\langle L_{indirect}(x \rightarrow \Theta) \rangle = \frac{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) d\omega_\Psi}{N} \sum_{i=1}^{N} L_r(r(x, \Psi_i) \rightarrow -\Psi_i) \cos(n_x, \Psi_i)$$

For a general BRDF, the appropriate sampling function might be very difficult to construct. Rejection sampling can be used (see 8), but might yield very high variance, especially when the BRDF has a narrow peak.

(108d) Multiple light sources, BRDF.cosine-sampling of hemisphere

$$p(\Psi) = \frac{f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi)}{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi) d\omega_\Psi}$$

$$\langle L_{indirect}(x \rightarrow \Theta) \rangle = \frac{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) \cos(n_x, \Psi) d\omega_\Psi}{N} \sum_{i=1}^{N} L_r(r(x, \Psi_i) \rightarrow -\Psi_i)$$

For a general BRDF, the appropriate sampling function might be very difficult to construct. Rejection sampling can be used (see 8), but might yield very high variance, especially when the BRDF has a narrow peak.

(109) End of recursion - Russian Roulette

Russian Roulette is a technique that can be used to ends recursive evaluation, but still keeps the final result unbiased. For any (recursive) expression A to be evaluated, use the following procedure:

Pick value $\alpha \in [0, 1]$, and generate uniform random number $r \in [0, 1]$:

if $r \leq \alpha$: $\langle A \rangle = \frac{A}{\alpha}$

if $r > \alpha$: $\langle A \rangle = 0$

Expected value: $E[\langle A \rangle] = \alpha \cdot \frac{A}{\alpha} + (1 - \alpha) \cdot 0 = A$

Global illumination context: when recursively tracing rays, Russian Roulette can be used to stop the recursion. Any value for α can be used, but a popular choice is the directional-hemispherical reflectance (64) of the surface point where the reflection is being considered.

Another variant of looking at RR is by rescaling the function to be integrated, such that part of the integration domain yields 0, and thereby effectively stopping any further evaluations.
E.g. computation of the one-dimensional integral

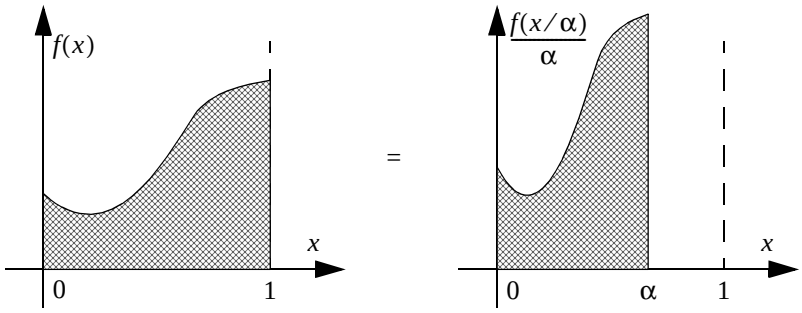
$$I = \int_0^1 f(x) dx = \int_0^\alpha \frac{1}{\alpha} f\left(\frac{x}{\alpha}\right) dx$$

Using pdf $p(x)$:

if $x_i \le \alpha$: $\langle I \rangle = \dfrac{f(x_i/\alpha)}{\alpha \cdot p(x_i)}$

if $x_i > \alpha$: $\langle I \rangle = 0$

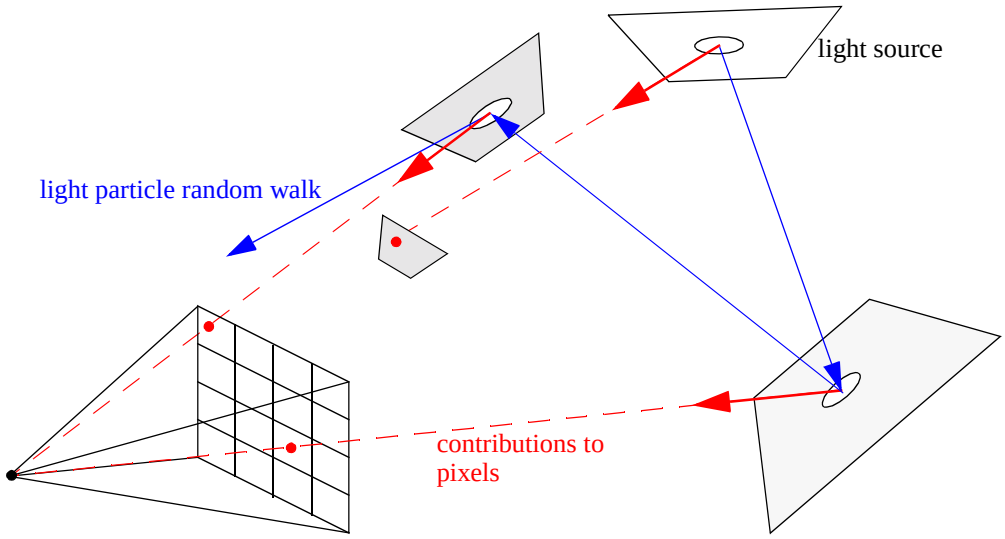
Graphical representation:



C. Light Tracing

Light tracing is the dual algorithm of ray tracing. Instead of shooting rays from the eye through a pixel, and working towards the light source, light tracing shoots rays from the light sources, and works towards the pixels.

References: R9, R10, R11, R12
Principle:



Variants:

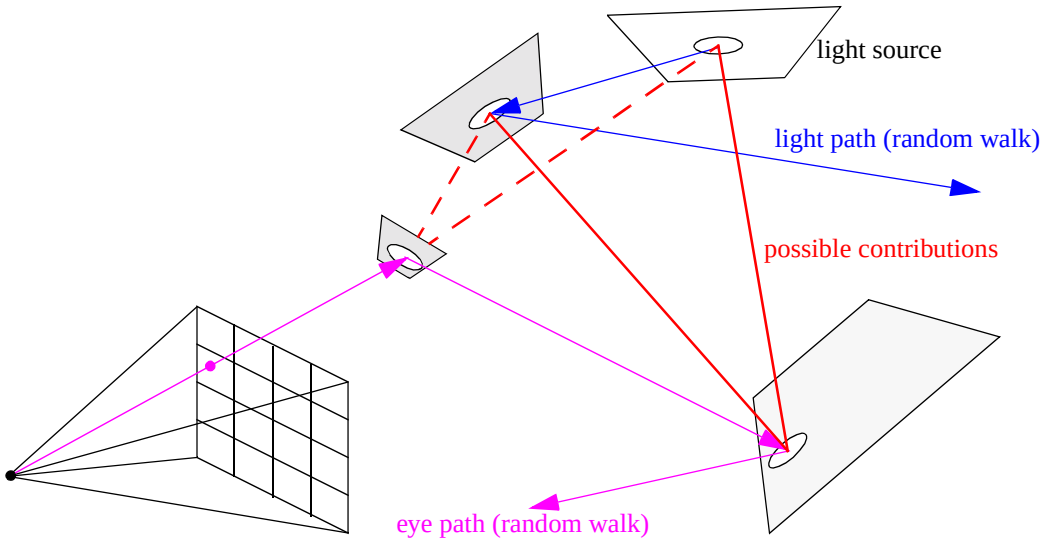
a. If the eye is a surface, one could just let the light particles bounce around the scene till they hit the eye surface. A contribution to the relevant pixel is then recorded. This would be the dual of stochastic ray tracing without shadow rays.

b. All hit points of all light particles can be stored in the scene. When the camera moves, the hit points can be reprojected to new camera view. This requires a re-evaluation of the BRDF at the reprojected hit points. This could be combined with any visibility determination scheme used in reprojection algorithms, or even using the original geometry.

D. Bidirectional Tracing

Bidirectional ray tracing constructs paths from the light sources and the eye simultaneously, and employs a meet-in-the-middle strategy for the paths generated. Ray tracing and light tracing can be considered to be special cases of bidirectional tracing.

References: R22, R24, R42
Principle:



If the light paths are of length 0 (only a random point on the light source is generated), bidirectional tracing behaves as stochastic ray tracing. If the eye path is of length 0, bidirectional tracing becomes identical to light tracing.

XIV. Multipass Algorithms

Multipass algorithms denote a class of algorithms that combine several global illumination passes into a single solution. Different passes are selected to compute subsets of all light paths. Clever combinations of different passes have been the subject of many global illumination papers. Often, one or more passes are used to store partial illumination information in the scene, which is then read out by a subsequent pass.

To classify the different types of lightpaths, regular expressions can be used based on Heckbert's notation (R19). See also (R38).

D, D_r, D_t : diffuse reflection, transmission

G, G_r, G_t : glossy reflection, transmission

S, S_r, S_t : specular reflection, transmission

X : $(D|S|G)$

E : the eye

L : a light source

* : zero or more occurences

+ : one or more occurences

| : "or"

A. PHOTON MAPPING

Photon mapping traces light particles or photons through the scenes, and stores all possible hits of the random walks. During a subsequent stochastic ray tracing pass, information from the stored photons is used to compute several lighting illumination components.

References: R20

XV. Test Scenes for Global Illumination

(110) Mother of all test scenes

Data for the original Cornell Box can be found at:

`http://www.graphics.cornell.edu/online/box/`

(111) Analytic Solution - General Rendering Equation

If $\forall(x, \Theta)$: $L_e(x \rightarrow \Theta) + \int_{\Omega_x} d\omega_\Psi f_r(x, \Psi \leftrightarrow \Theta)\cos(n_x, \Psi) = 1$ and if the environment is closed, then:

$L(x \rightarrow \Theta) \equiv 1$ is the solution for $L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} d\omega_\Psi L(x \leftarrow \Psi) f_r(x, \Psi \leftrightarrow \Theta)\cos(n_x, \Psi)$ (72).

For diffuse surfaces: $\forall(x, \Theta)$: $L_e(x \rightarrow \Theta) + \rho_d = 1$

In other words, every surface point emits exactly the amount of energy that is absorbed at the point.

(112) Analytic Solution - Radiosity System

if $\forall$patches i: $P_{ei} + \rho_i = 1$ and if the environment is closed, then:

$\forall$ i: $P_i = 1$ is the solution for the linear system $P_i = P_{ei} + \rho_i \sum_j F_{j \rightarrow i} P_j$ (93).

(113) Testing global illumination algorithms

Brian Smits and Henrik Jensen have proposed a number of simple and elegant scenes to test light transport modes (R36), available at the following URL: `http://www.cs.utah.edu/~bes/papers/scenes/`.

A number of scenes, designed to test performance and overall image appearance, made available by Greg Ward Larson and Peter Shirley are available from the RADIANCE website: `http://radsite.lbl.gov/mgf/`.

Andrew Willmott and Paul Heckbert have designed parametrized test scenes for diffuse global illumination (radiosity), so that algorithms' sensitivity to various parameters, such as reflectance or scene complexity, can be tested. They also described a scheme for making comparisons that normalizes for differences in error and time. All information can be found at the following URL: `http://www.cs.cmu.edu/~radiosity/`.

(114) Testing ray tracing performance

The classic Standard Procedural Database (R17), designed by Eric Haines to test the performance of classic ray tracers, is available from: `http://www.acm.org/tog/resources/SPD/overview.html`.

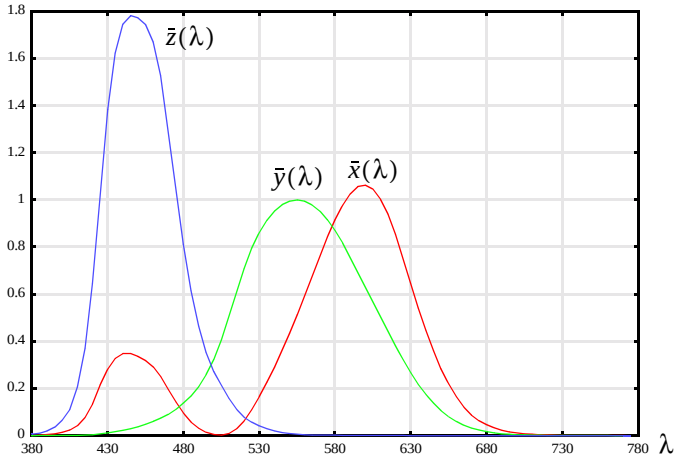
(115) Testing animated ray tracing

Jonas Lext, Ulf Assarsson, and Tomas Möller have designed a number of scenes to test animated ray tracing. `http://www.ce.chalmers.se/BART/`.

XVI. Color & Display[1]

(116) Spectrum $C(\lambda)$ to CIE XYZ

Color matching functions $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ [2]



$$X = \int_{380nm}^{780nm} C(\lambda)\bar{x}(\lambda)d\lambda \qquad Y = \int_{380nm}^{780nm} C(\lambda)\bar{y}(\lambda)d\lambda \qquad Z = \int_{380nm}^{780nm} C(\lambda)\bar{z}(\lambda)d\lambda$$

$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z} \qquad z = \frac{Z}{X+Y+Z}$$

(117) xyY to XYZ

$$X = Y\frac{x}{y} \qquad Y = Y \qquad Z = Y\frac{(1-x-y)}{y}$$

(118) CIE XYZ to Spectrum $C(\lambda)$

Choose 3 linearly independent functions $F_1(\lambda)$, $F_2(\lambda)$ and $F_3(\lambda)$.

$C(\lambda)$ can be writtten as $aF_1(\lambda) + bF_2(\lambda) + cF_3(\lambda)$ with a, b, c determined by:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \int F_1(\lambda)\bar{x}(\lambda)d\lambda & \int F_2(\lambda)\bar{x}(\lambda)d\lambda & \int F_3(\lambda)\bar{x}(\lambda)d\lambda \\ \int F_1(\lambda)\bar{y}(\lambda)d\lambda & \int F_2(\lambda)\bar{y}(\lambda)d\lambda & \int F_3(\lambda)\bar{y}(\lambda)d\lambda \\ \int F_1(\lambda)\bar{z}(\lambda)d\lambda & \int F_2(\lambda)\bar{z}(\lambda)d\lambda & \int F_3(\lambda)\bar{z}(\lambda)d\lambda \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

See also R15.

1. A good source for color data: `http://cvision.ucsd.edu/basicindex.htm`
2. Tabular data for the color matching curves can be found in R16, p. 1170.

E.g. 1: Dirac-impulse functions at wavelengths λ_u, λ_v and λ_w.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \bar{x}(\lambda_u) & \bar{x}(\lambda_v) & \bar{x}(\lambda_w) \\ \bar{y}(\lambda_u) & \bar{y}(\lambda_v) & \bar{y}(\lambda_w) \\ \bar{z}(\lambda_u) & \bar{z}(\lambda_v) & \bar{z}(\lambda_w) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

E.g. 2: First three Fourier basis functions defined on interval $[\lambda_{min}, \lambda_{max}] = [380, 780]$

$$F_1(\lambda) = 1.0$$

$$F_2(\lambda) = \frac{1}{2}\left(1 + \sin\left(2\pi\frac{\lambda - 380}{400}\right)\right)$$

$$F_3(\lambda) = \frac{1}{2}\left(1 + \cos\left(2\pi\frac{\lambda - 380}{400}\right)\right)$$

(119) CIE XYZ to/from NTSC standard RGB based on standard CIE phospors and D6500 white point.

Scaling is such that RGB=[1,1,1] produces the chromaticity values of the white point with Y=1.0 (R16, p.104).

	x	*y*
r	0.670	0.330
g	0.210	0.710
b	0.140	0.080
w	0.313	0.329

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.5893 & 0.1789 & 0.1831 \\ 0.2904 & 0.6051 & 0.1045 \\ 0.0000 & 0.0684 & 1.0202 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.967 & -0.548 & -0.297 \\ -0.955 & 1.938 & -0.027 \\ 0.064 & -0.130 & 0.982 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

(120) CIE XYZ to/from $L^*u^*v^*$ color space

Defined w.r.t. white point (X_n, Y_n, Z_n), usually scaled such that $Y_n = 100$.

$$L^* = \begin{cases} Y/Y_n \geq 0.008856 & 116(Y/Y_n)^{1/3} - 16 \\ Y/Y_n \leq 0.008856 & 903.3(Y/Y_n) \end{cases}$$

$$u^* = 13L^*(u' - u'_n)$$

$$v^* = 13L^*(v' - v'_n)$$

where

$$u' = \frac{4X}{X + 15Y + 3Z}$$

$$v' = \frac{9Y}{X + 15Y + 3Z}$$

$$u'_n = \frac{4X_n}{X_n + 15Y_n + 3Z_n}$$

$$v'_n = \frac{9Y_n}{X_n + 15Y_n + 3Z_n}$$

$$X = -\left(\frac{A}{R} + 3Z\right) \qquad Q = \frac{u^*}{13L^*} + u'_n$$

$$Y = Y_n\left(\frac{L^* + 16}{116}\right)^3 \quad \text{where} \quad R = \frac{v^*}{13L^*} + v'_n$$

$$Z = \frac{(Q-4)A - 15QRY}{12R} \qquad A = 3Y(5R-3)$$

(121) CIE XYZ to/from $L^*a^*b^*$ color space

Defined w.r.t. white point (X_n, Y_n, Z_n), usually scaled such that $Y_n = 100$.

$$L^* = \begin{cases} Y/Y_n \geq 0.008856 & 116(Y/Y_n)^{1/3} - 16 \\ Y/Y_n \leq 0.008856 & 903.3(Y/Y_n) \end{cases}$$

$$a^* = 500L^*(f(X/X_n) - f(Y/Y_n))$$

$$b^* = 200L^*(f(Y/Y_n) - f(Z/Z_n))$$

where $f(r) = \begin{cases} r \geq 0.008856 & r^{1/3} \\ r \leq 0.008856 & 7.787r + 16/116 \end{cases}$

$$X = X_n\left(\left(\frac{Y}{Y_n}\right)^{1/3} + \frac{a^*}{500L^*}\right)^3$$

$$Y = Y_n\left(\frac{L^* + 16}{116}\right)^3$$

$$Z = Z_n\left(\left(\frac{Y}{Y_n}\right)^{1/3} + \frac{b^*}{200L^*}\right)^3$$

Index

Q

R

S

T

V

References

Abbreviations:

SIGGRAPH xx: Proceedings of SIGGRAPH conference held in 19xx
EGWR xx: Proceedings of the Eurographics Workshop on Rendering held in 19xx
EG xx: Proceedings of the Eurographics conference held in 19xx

[1] Arvo J.; *Stratified Sampling of Spherical Triangles*; SIGGRAPH 95

[2] Arvo J.; *Applications of Irradiance Tensors to the Simulation of Non-Lambertian Phenomena*; SIGGRAPH 95

[3] Baum D., Rushmeier H., Winget J.; *Improving Radiosity Solutions Through the Use of Analytically Determined Form-Factors*; SIGGRAPH 89

[4] Bekaert Ph., Dutré Ph., Willems Y.; *Final Radiosity Gather Step using a Monte Carlo Technique with Optimal Importance Sampling*; Report CW275, January 1996, Dept. of Computer Science, Katholieke Universiteit Leuven.

[5] Bekaert Ph.; *Hierarchical and Stochastic Algorithms for Radiosity*; Ph.D. Thesis, Katholieke Universiteit Leuven, 1999 http://www.cs.kuleuven.ac.be/cwis/research/graphics/CGRG.PUBLICATIONS/PHBPHD/

[6] Carvalho, P., Cavalcanti, P.; *Point in Polyhedron Testing Using Spherical Polygons*; Graphics Gems V, p. 42-49

[7] Cohen M., Wallace J.; *Radiosity and Realistic Image Synthesis*; Academic Press

[8] Cook R.L., Torrance K.E.; *A reflectance model for computer graphics*; Computer Graphics, Vol. 15, No. 3, 1981, pp. 307-316 (SIGGRAPH 81)

[9] Dutré Ph., Lafortune E., Willems Y.; *Monte Carlo Light Tracing with Direct Computation of Pixel Intensities*; Proceedings of CompuGraphics, pp. 128-137, Alvor, Portugal, December 1993

[10] Dutré Ph., Willems Y.; *Importance-driven Light Tracing*; EGWR 94

[11] Dutré Ph., Willems Y.; *Potential-driven Monte Carlo Particle Tracing for Diffuse Environments with Adaptive Probability Functions*; EGWR 95

[12] Dutré Ph.; *Mathematical Frameworks and Monte Carlo Algorithms for Global Illumination in Computer Graphics*; Ph.D. Thesis, Katholieke Universiteit Leuven, 1996.
http://www.cs.kuleuven.ac.be/~graphics/CGRG.PUBLICATIONS/PHDPHD/

[13] Dutré Ph., Bekaert Ph., Bala K.; *Advanced Global Illumination*; A K Peters 2003
http://www.advancedglobalillumination.com/

[14] Fishman G.S.; *Monte Carlo: Concepts, Algorithms and Applications*; Springer-Verlag, 1996

[15] Glassner A.; *How to Derive a Spectrum from an RGB Triplet*; IEEE CG&A, 9(4):95-99, July 1989

[16] Glassner A.; *Principles of Digital Image Synthesis*; Morgan Kaufmann

[17] Haines, E. *A Proposal for Standard Graphics Environments*; IEEE Computer Graphics and Applications, vol. 7, no. 11, November 1987, pp. 3-5

[18] Hammersley J.M., Handscomb D.C.; *Monte Carlo Methods*; Chapman and Hall, London, 1964

[19] Heckbert P.; *Adaptive radiosity textures for bidirectional ray tracing*, SIGGRAPH 90.

[20] Jensen H.W.; *Realistic Image Synthesis using Photon Mapping*; AK Peters, 2001

[21] Kalos M.H., Whitlock P.A.; *Monte Carlo Methods*; Wiley-Interscience, New York, 1986

[22] Lafortune E.P., Willems Y.D.; *Bi-directional Path Tracing*; Proceedings of CompuGraphics, pp. 145-153, Alvor, Portugal, December 1993

[23] Lafortune E.P., Willems Y.D.; *Using the Modified Phong Reflectance Model for Physically Based Rendering*; Report CW 197, November 1994, Department of Computer Science, Katholieke Universiteit Leuven.

[24] Lafortune E.; *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*; Ph.D. Thesis, Katholieke Universiteit Leuven, 1996.

[25] Lafortune E., Foo S., Torrance K., Greenberg D.; *Non-Linear Approximation of Reflectance Functions*; SIGGRAPH 97

[26] Lewis R.; *Making Shaders More Physically Plausible*; Computer Graphics Forum, 13 (2), 1994 (also in EGWR 93)

[27] Miller, Robert D.; *Computing the Area of a Spherical Polygon*; Graphics Gems IV, p. 132-137

[28] Sbert M.*; The use of global random directions to compute radiosity - Global Monte Carlo techniques*. Universitat Politècnica de Catalunya, Barcelona, Spain, 1997.
http://ima.udg.es/~mateu/

[29] Schröder P., Hanrahan P.; *On the Form Factor between Two Polygons*; SIGGRAPH 93

[30] Shirley P., Wang C., Zimmerman K.; *Monte Carlo Techniques for Direct Lighting Calculations*; ACM Transactions on Graphics, Vol. 15,1, January 1996.

[31] Shirley P.; *Realistic Ray Tracing*; A.K.Peters 2000

[32] Siegel R., Howell J.R.; *Thermal Radiation Heat Transfer*; Hemisphere Publishing Corporation

[33] Sillion F. and Puech C.; *Radiosity & Global Illumination*; Morgan-Kaufmann

[34] Sillion F., Drettakis G., Soler C.; *A Clustering Algorithm for Radiance Calculation in General Environments*; EGWR 95

[35] Sillion F.; *A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters*; IEEE Transactions on Vizualizations and Computer Graphics, 1(3), September 1995 (also in EGWR 94)

[36] Smits, B. and Jensen, H. W.; *Global Illumination Test Scenes*; Tech. Rep. UUCS-00-013, Computer Science Department, University of Utah, June 2000

[37] Spanier J., Gelbard E.M.; *Monte Carlo Principles and Neutron Transport Problems*; Addison-Wesley, 1969

[38] Suykens F., Willems Y.; *Weighted Multipass Methods for Global Illumination*; EG 99.

[39] Tobler R., Neumann L., Sbert M., Purgathofer W.; *A New Form Factor Analogy and its Application to Stochastic Global Illumination Algorithms*, EGWR 98

[40] Turk G.*; Generating Random Points in Triangles*; Graphics Gems I, p.24

[41] Ureña C., Torres J.; *Improved Irradiance Computation by Importance Sampling*; EGWR 97

[42] Veach E., Guibas L.; *Bidirectional Estimators for Light Transport;* EGWR 95

[43] Veach E., L. Guibas L.; *Optimally Combining Sampling Techniques for Monte Carlo Rendering*; SIGGRAPH 95

[44] Veach E.; *Robust Monte Carlo Methods for Light Transport Simulation*; Ph.D. Dissertation, Stanford University, December 1997.

[45] Ward G.; *Adaptive Shadow Testing for Ray Tracing*; Photorealistic Rendering in Computer Graphics. P. Brunet and F.W. Jansen, eds. Springer-Verlag (EGWR 91)

[46] Ward G.; *Measuring and Modeling Anisotropic Reflection*; SIGGRAPH 92

[47] Whitted T.; *An improved illumination model for shaded display*; Communications of the ACM, 23(6):343-349, June 1980

[48] Zimmerman K., Shirley P.; *A Two-pass Solution to the Rendering Equation with a Source Visibility Preprocess*; EGWR 95