Light Propagation Volumes - Annotations

Andreas Kirsch

July 17, 2010

1 Notation

$$\overline{f(x)} := max\{f(x), 0\}$$

$$\textit{flux } \Phi \textit{ of a light (energy / time)}$$

$$\textit{radiant intensity } I := \frac{d\Phi}{d\omega}, \textit{ with direction } \omega$$

$$\Phi = \int_{S^2} I\left(\omega\right)\, d\omega$$

$$\textit{radiance } L := \frac{d^2\Phi}{dAd\omega}, \textit{ with area } A$$

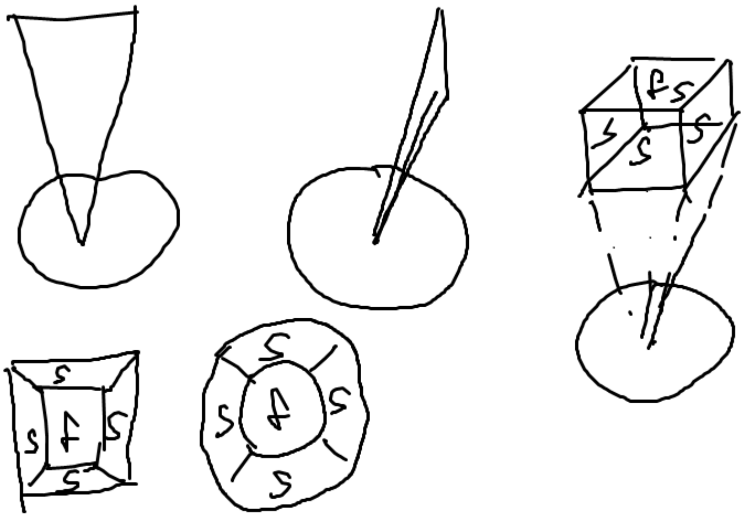
See Real-Time Rendering or Physically Based Rendering for more details. The notation tries to stick to Real-Time Rendering.

2 Subtended Solid Angle Calculations

It is necessary to calculate the solid angle of certain shapes as seen from the unit sphere. This solid angle is called the subtended solid angle (of the shape).

There are multiple ways to calculate the solid angle of a simple shape. Crytek's LPV paper needs the subtended solid angle of squares with different orientations relative to the origin of the unit sphere:

- back face of the neighboring cell during propagation
- side faces of the neighboring cell during propagation (due to symmetry the solid angles of such side faces are equal)



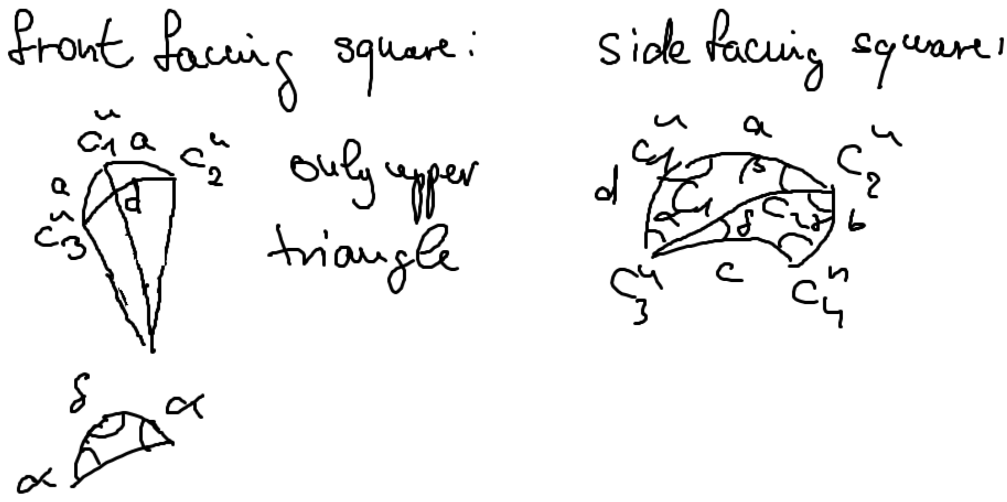
To calculate these subtended solid angles, I've used the following two methods for each type:

1. triangulate the square, project the corners onto the unit sphere and use spherical trigonometry to calculate the area of triangles.

2. use the following physical model: we assume that a point light is at the origin with flux $4\,\pi$ (that is the radiant intensity is 1), then we measure the amount of light (ie light flux) on the shape using physically correct integration. Since the radiant intensity is 1 and constant, the light flux on the shape is equal to the solid angle because of energy conservation:

$$solid\ angle = \int_{solid\ angle} 1\, d\omega = \int_{solid\ angle} intensity(\omega)\, d\omega = \int_{shape\ surface} radiance(x)\, dx$$

2.1 Solid Angle Using Spherical Trigonometry



The next few pages contain Maple worksheets to calculate the solid angle (first for the front faces, then the side faces). h is assumed to be cell width (it doesn't matter though due to direction normalizazion).

Loading LinearAlgebra

$c1 := Normalize\left(\left\langle \frac{h}{2}, \frac{h}{2}, \frac{3\,h}{2} \right\rangle, Euclidean\right)$ assuming $h > 0$

$$\begin{bmatrix} \frac{1}{11}\sqrt{11} \\ \frac{1}{11}\sqrt{11} \\ \frac{3}{11}\sqrt{11} \end{bmatrix} \tag{1}$$

$c2 := Normalize\left(\left\langle -\frac{h}{2}, \frac{h}{2}, \frac{3\,h}{2} \right\rangle, Euclidean\right)$ assuming $h > 0$

$$\begin{bmatrix} -\frac{1}{11}\sqrt{11} \\ \frac{1}{11}\sqrt{11} \\ \frac{3}{11}\sqrt{11} \end{bmatrix} \tag{2}$$

$c3 := Normalize\left(\left\langle \frac{h}{2}, -\frac{h}{2}, \frac{3\,h}{2} \right\rangle, Euclidean\right)$ assuming $h > 0$

$$\begin{bmatrix} \frac{1}{11}\sqrt{11} \\ -\frac{1}{11}\sqrt{11} \\ \frac{3}{11}\sqrt{11} \end{bmatrix} \tag{3}$$

$c4 := Normalize\left(\left\langle -\frac{h}{2}, -\frac{h}{2}, \frac{3\,h}{2} \right\rangle, Euclidean\right)$ assuming $h > 0$

$$\begin{bmatrix} -\frac{1}{11}\sqrt{11} \\ -\frac{1}{11}\sqrt{11} \\ \frac{3}{11}\sqrt{11} \end{bmatrix} \tag{4}$$

$a := c1.c2$

$$\frac{9}{11} \tag{5}$$

$a, c2.c4, c4.c3, c1.c3$

$$\frac{9}{11}, \frac{9}{11}, \frac{9}{11}, \frac{9}{11} \tag{6}$$

$d := c3.c2$

$$\frac{7}{11} \tag{7}$$

$d, c1.c4$

$$\frac{7}{11}, \frac{7}{11} \tag{8}$$

$d = a \cdot a + \sin(\arccos(a)) \cdot \sin(\arccos(a)) \cdot \cos(\text{delta})$

$$\frac{7}{11} = \frac{81}{121} + \frac{40}{121} \cos(\delta) \tag{9}$$

$\xrightarrow{\text{isolate for delta}}$

$$\delta = \pi - \arccos\left(\frac{1}{10}\right) \tag{10}$$

$\xrightarrow{\text{assign}}$

$a = d \cdot a + \sin(\arccos(d)) \cdot \sin(\arccos(a)) \cdot \cos(\text{alpha})$

$$\frac{9}{11} = \frac{63}{121} + \frac{12}{121} \sqrt{2} \sqrt{10} \cos(\alpha) \tag{11}$$

$\xrightarrow{\text{isolate for alpha}}$

$$\alpha = \arccos\left(\frac{3}{20} \sqrt{2} \sqrt{10}\right) \tag{12}$$

$\xrightarrow{\text{assign}}$

$FontFaceArea := 2 \cdot (2 \cdot \text{alpha} + \text{delta} - \text{Pi})$

$$4 \arccos\left(\frac{3}{20} \sqrt{2} \sqrt{10}\right) - 2 \arccos\left(\frac{1}{10}\right) \tag{13}$$

$\xrightarrow{\text{at 10 digits}}$

$$0.400669685 \tag{14}$$

Loading LinearAlgebra

$c1 := Normalize\left(\left\langle \frac{h}{2}, \frac{h}{2}, \frac{3}{2}h \right\rangle, Euclidean\right)$ assuming $h > 0$

$$\begin{bmatrix} \frac{1}{11}\sqrt{11} \\ \frac{1}{11}\sqrt{11} \\ \frac{3}{11}\sqrt{11} \end{bmatrix} \tag{1}$$

$c2 := Normalize\left(\left\langle \frac{h}{2}, -\frac{h}{2}, \frac{3}{2}h \right\rangle, Euclidean\right)$ assuming $h > 0$

$$\begin{bmatrix} \frac{1}{11}\sqrt{11} \\ -\frac{1}{11}\sqrt{11} \\ \frac{3}{11}\sqrt{11} \end{bmatrix} \tag{2}$$

$c3 := Normalize\left(\left\langle \frac{h}{2}, \frac{h}{2}, \frac{h}{2} \right\rangle, Euclidean\right)$ assuming $h > 0$

$$\begin{bmatrix} \frac{1}{3}\sqrt{3} \\ \frac{1}{3}\sqrt{3} \\ \frac{1}{3}\sqrt{3} \end{bmatrix} \tag{3}$$

$c4 := Normalize\left(\left\langle \frac{h}{2}, -\frac{h}{2}, \frac{h}{2} \right\rangle, Euclidean\right)$ assuming $h > 0$

$$\begin{bmatrix} \frac{1}{3}\sqrt{3} \\ -\frac{1}{3}\sqrt{3} \\ \frac{1}{3}\sqrt{3} \end{bmatrix} \tag{4}$$

$a := c1.c2$

$$\frac{9}{11} \tag{5}$$

$b := c2.c4$

$$\frac{5}{33}\sqrt{11}\sqrt{3} \tag{6}$$

$c := c3.c4$

$$\frac{1}{3} \tag{7}$$

$d := c3.c1$

$$\frac{5}{33}\sqrt{11}\sqrt{3} \tag{8}$$

$e := c3.c2$

$$\frac{1}{11}\sqrt{11}\sqrt{3} \tag{9}$$

$e = a \cdot d + \sin(\arccos(a)) \cdot \sin(\arccos(d)) \cdot \cos(C1)$

$$\frac{1}{11}\sqrt{11}\sqrt{3} = \frac{15}{121}\sqrt{11}\sqrt{3} + \frac{4}{363}\sqrt{10}\sqrt{66}\cos(C1) \tag{10}$$

$\xrightarrow{\text{isolate for C1}}$

$$C1 = \pi - \arccos\left(\frac{1}{220}\sqrt{11}\sqrt{3}\sqrt{10}\sqrt{66}\right) \tag{11}$$

$\xrightarrow{\text{assign}}$

$e = b \cdot c + \sin(\arccos(b)) \cdot \sin(\arccos(c)) \cdot \cos(C2)$

$$\frac{1}{11}\sqrt{11}\sqrt{3} = \frac{5}{99}\sqrt{11}\sqrt{3} + \frac{4}{99}\sqrt{66}\sqrt{2}\cos(C2) \tag{12}$$

$\xrightarrow{\text{isolate for C2}}$

$$C2 = \arccos\left(\frac{1}{132}\sqrt{11}\sqrt{3}\sqrt{66}\sqrt{2}\right) \tag{13}$$

$\xrightarrow{\text{assign}}$

$a = d \cdot e + \sin(\arccos(d)) \cdot \sin(\arccos(e)) \cdot \cos(\text{alpha})$

$$\frac{9}{11} = \frac{5}{11} + \frac{4}{363}\sqrt{66}\sqrt{22}\cos(\alpha) \tag{14}$$

$\xrightarrow{\text{isolate for alpha}}$

$$\alpha = \arccos\left(\frac{1}{44}\sqrt{66}\sqrt{22}\right) \tag{15}$$

$\xrightarrow{\text{assign}}$

$d = e \cdot a + \sin(\arccos(e)) \cdot \sin(\arccos(a)) \cdot \cos(\text{beta})$

$$\frac{5}{33}\sqrt{11}\sqrt{3} = \frac{9}{121}\sqrt{11}\sqrt{3} + \frac{4}{121}\sqrt{22}\sqrt{10}\cos(\beta) \tag{16}$$

$\xrightarrow{\text{isolate for beta}}$

$$\beta = \arccos\left(\frac{7}{660}\sqrt{11}\sqrt{3}\sqrt{22}\sqrt{10}\right) \tag{17}$$

$\xrightarrow{\text{assign}}$

$c = e \cdot b + \sin(\arccos(e)) \cdot \sin(\arccos(b)) \cdot \cos(gamma1)$

$$\frac{1}{3} = \frac{5}{11} + \frac{4}{363}\sqrt{22}\sqrt{66}\cos(\gamma 1) \tag{18}$$

$\xrightarrow{\text{isolate for gamma1}}$

$$\gamma 1 = \pi - \arccos\left(\frac{1}{132}\sqrt{66}\sqrt{22}\right) \quad \textbf{(19)}$$

$\xrightarrow{\text{assign}}$

$b = e \cdot c + \sin(\arccos(e)) \cdot \sin(\arccos(c)) \cdot \cos(\text{delta})$

$$\frac{5}{33}\sqrt{11}\sqrt{3} = \frac{1}{33}\sqrt{11}\sqrt{3} + \frac{4}{33}\sqrt{22}\sqrt{2}\cos(\delta) \quad \textbf{(20)}$$

$\xrightarrow{\text{isolate for delta}}$

$$\delta = \arccos\left(\frac{1}{44}\sqrt{11}\sqrt{3}\sqrt{22}\sqrt{2}\right) \xrightarrow{\text{assign}}$$

$SideFaceArea := C1 + \text{alpha} + \text{beta} - \text{Pi} + gamma1 + \text{delta} + C2 - \text{Pi}$

$$-\arccos\left(\frac{1}{220}\sqrt{11}\sqrt{3}\sqrt{10}\sqrt{66}\right) + \arccos\left(\frac{1}{44}\sqrt{66}\sqrt{22}\right) + \arccos\left(\frac{7}{660}\sqrt{11}\sqrt{3}\sqrt{22}\sqrt{10}\right) - \arccos\left(\frac{1}{132}\sqrt{66}\sqrt{22}\right) + \arccos\left(\frac{1}{44}\sqrt{11}\sqrt{3}\sqrt{22}\sqrt{2}\right) + \arccos\left(\frac{1}{132}\sqrt{11}\sqrt{3}\sqrt{66}\sqrt{2}\right) \quad \textbf{(21)}$$

$\xrightarrow{\text{at 20 digits}}$

$$0.42343135443673907668 \quad \textbf{(22)}$$

$\xrightarrow{\text{at 5 digits}}$

$$0.42347 \quad \textbf{(23)}$$

2.2 Solid Angle Using Integration

To calculate the front facing square's solid angle (ie the solid angle of the back face of the neighboring cell), assume that the square lies parallel to the xy-plane with its center at $(0, 0, 3)$ (with side length $= 2$). We integrate the radiance of its surface points $p = (x, y, z)$, that is $z = 3$ constant and x and y from -1 to 1.

$$\begin{aligned}
FrontFacingSolidAngle &= \int_{-1}^{1}\int_{-1}^{1} L(p)\,dp = \int_{-1}^{1}\int_{-1}^{1} \cos \angle p, z\text{-}axis\,\frac{1}{r^2}\,dp \\
&= \int_{-1}^{1}\int_{-1}^{1} \frac{\langle (0,0,1)\,, p\rangle}{\|p\|}\,\frac{1}{\|p\|^2}\,dx\,dy \\
&= \int_{-1}^{1}\int_{-1}^{1} 3\,\left(x^2 + y^2 + 9\right)^{-3/2}\,dx\,dy \\
&= 4\,\arctan\left(1/33\,\sqrt{11}\right) = 0.4006696844\ldots
\end{aligned}$$

To calculate the side facing square's solid angle (ie the solid angle of the back face of the neighboring cell), assume that the square lies parallel to the yz-plane with its center at $(1, 2, 0)$ (with side length $= 2$). We integrate the radiance of its surface points $p = (x, y, z)$, that is $x = 1$ constant, y from 1 to 3 and z from -1 to 1.

$$\begin{aligned}
SideFacingSolidAngle &= \int_{-1}^{1}\int_{1}^{3} \frac{\langle (1,0,0)\,, p\rangle}{\|p\|}\,\frac{1}{\|p\|^2}\,dydz \\
&= \int_{-1}^{1}\int_{1}^{3} \left(1 + y^2 + z^2\right)^{-3/2}\,dy\,dz \\
&= -1/3\,\pi + 2\,\arctan\left(3/11\,\sqrt{11}\right) = 0.423431354\ldots
\end{aligned}$$

3 Reflective Shadow Maps

The reflective shadow map papers only states that you store the "flux emitted through every pixel" in the flux render target. This means that you don't store the total light flux Φ_L or use the total light flux to calculate the outgoing flux but instead, you base it on the flux through the pixel: *outgoing flux* $\Phi_o =$ *surface area* $A \cdot$ *radiant exitance* M

$$E = \int_{\Omega} L_i(\omega_i) \cos\theta_i\,d\omega_i$$

$$M = \int_{\Omega} L_o(\omega_o) \cos\theta_o\,d\omega_o$$

$$L_o = \frac{c_{diff}}{\pi}\,E\,\overline{\cos\theta_i}$$

with $E = \frac{I}{r^2} = \frac{1}{r^2}\frac{\Phi_L}{4\pi}$. Then:

$$\begin{aligned}
\Phi_o &= A \cdot \int_\Omega L_o(\omega_o)\,\cos\theta_o\,d\omega_o && (1)\\
&= A\,L_o(\omega_o)\int_\Omega \cos\theta_o\,d\omega_o && (2)\\
&= A\,L_o(\omega_o)\,\pi && (3)\\
&= A\,\frac{c_{diff}}{\pi}\,\frac{\Phi_L}{r^2\,4\pi}\,\overline{\cos\theta_i}\,\pi && (4)\\
&= \frac{c_{diff}}{\pi}\,\frac{A}{r^2}\,\frac{\Phi_L}{4\pi}\,\overline{\cos\theta_i}\,\pi && (5)\\
&= c_{diff}\,\rho\,\frac{\Phi_L}{4\pi}\,\overline{\cos\theta_i} && (6)\\
&= c_{diff}\,\frac{\rho}{4\pi}\,\Phi_L\,\overline{\cos\theta_i} && (7)
\end{aligned}$$

with ρ being the subtended solid angle of a surfel. If we assume a field of view of 90 degrees with aspect ratio 1, the solid angle of a single pixel can be approximated with:

$$\frac{4\pi}{6}\,\frac{1}{width \times height}$$

4 Light Injection

4.1 Intensity Formula Correction

Crytek's LPV paper says that, if $I_p(\omega)$ is the radiant intensity of an VPL and n_p is its normal and Φ_p the reflected flux, then:

$$I_p(\omega) = \Phi_p \overline{\langle n_p, \omega\rangle}$$

However:

$$\Phi_p = \int_{S^2} I_p(\omega) d\omega$$

If we expand the equation above:

$$\Phi_p = \int_{S^2} \Phi_p \overline{\langle n_p, \omega\rangle} d\omega = \Phi_p \int_\Omega \langle n_p, \omega\rangle\, d\omega = \Phi_p \cdot \pi \neq \Phi_p$$

I suggest the following correction factor:

$$I_p(\omega) = \frac{\Phi_p}{\pi} \overline{\langle n_p, \omega\rangle}$$

That is, the original equation is divided by π as normalization factor.

4.2 Spherical Harmonics Analytical Basis Functions

The analytical presentation of the first four base functions is simple:

$$S_0(x, y, z) = \frac{1}{2\sqrt{\pi}} \tag{8}$$

$$S_1(x, y, z) = -\frac{\sqrt{3}}{2\sqrt{\pi}} y \tag{9}$$

$$S_2(x, y, z) = \frac{\sqrt{3}}{2\sqrt{\pi}} z \tag{10}$$

$$S_3(x, y, z) = -\frac{\sqrt{3}}{2\sqrt{\pi}} x \tag{11}$$

To evaluate lighting with SH for some direction v, you first determine the coefficients/weights of the SH basis functions and then sum them up.

$$L = \sum_i s_i \, S_i(v)$$

See Stupid SH Tricks and Spherical Harmonic Lighting: The Gritty Details for more information.

4.3 Spherical Harmonics Low-Order Rotation

Let's assume we know the coefficients $s_0^z, s_1^z, \ldots$ of the clamped cosine lobe around the z axis, then we can determine the lighting in direction v for the cosine lobe around the normal n by transforming it into the space where the normal coincides with the z axis (ie rotate n onto the z axis):

$$L = \sum_i s_i^z \, S_i(R_{n\to z} \, v)$$

where $R_{n\to z}$ is a rotation matrix that rotates n onto z.

Before expanding this further, let's first take a look at the coefficients of the clamped cosine lobe:

$$s_0^z = \frac{\sqrt{\pi}}{2} = \int_\Omega \cos(\phi) S_0(\omega) \, d\omega = \frac{1}{2\sqrt{\pi}} \int_\Omega \cos(\phi) \, d\omega \tag{12}$$

$$s_1^z = 0 = \int_\Omega \cos(\phi) S_1(\omega) \, d\omega \tag{13}$$

$$s_2^z = \sqrt{\frac{\pi}{3}} = \int_\Omega \cos(\phi) S_2(\omega) \, d\omega \tag{14}$$

$$s_3^z = 0 = \int_\Omega \cos(\phi) S_3(\omega) \, d\omega \tag{15}$$

The y and x directions are 0 because the cosine lobe is centered isotropically around the z axis.

So let's look at the expanded version of this formula, if r_1^T, r_2^T, r_3^T are the row vectors of the matrix $R_{n\to z}$, that is: $v = \left(\begin{smallmatrix} x \\ y \\ z \end{smallmatrix}\right)$ and $R_{n\to z} = \begin{pmatrix} r_1^T \\ r_2^T \\ r_3^T \end{pmatrix}$. Then:

$$L = \sum_i s_i^z \, S_i \left(R_{n\to z} \, v\right) = \sum_i s_i^z \, S_i \left(\begin{pmatrix} r_1^T \, v \\ r_2^T \, v \\ r_3^T \, v \end{pmatrix} \right) \tag{16}$$

$$L = s_0^z \, c_0 \tag{17}$$

$$+ \, s_1^z \, (-c_1) \, r_2^T \, v \tag{18}$$

$$+ \, s_2^z \, c_1 \, r_3^T \, v \tag{19}$$

$$+ \, s_3^z \, (-c_1) \, r_1^T \, v \tag{20}$$

with $c_0 := \frac{1}{2\sqrt{\pi}}$ and $c_1 := \frac{\sqrt{3}}{2\sqrt{\pi}}$. Since $s_1^z = 0$ and $s_3^z = 0$, we can simplify the equation to

$$L = s_0^z \, c_0 + s_2^z \, c_1 \, r_3^T \, v \tag{21}$$

$$= s_0^z \, c_0 + s_2^z \, c_1 \, r_{31} \, x + s_2^z \, c_1 \, r_{32} \, y + s_2^z \, c_1 \, r_{33} \, z \tag{22}$$

$$= s_0^z \, c_0 + s_2^z \, c_1 \, r_{32} - \; y + s_2^z \, c_1 \, r_{33} \, z + s_2^z \, c_1 \, r_{31} \, x \tag{23}$$

$$= s_0^z \, c_0 - s_2^z \, (-c_1) \, r_{32} - \; y + s_2^z \, c_1 \, r_{33} \, z + s_2^z \, (-c_1) \, r_{31} \, x \tag{24}$$

$$= s_0^z \, S_0 \, (v) - s_2^z \, r_{32} \, S_1 \, (v) + s_2^z \, r_{33} \, S_2 \, (v) - s_2^z \, r_{31} \, S_3 \, (v) \tag{25}$$

Now the question is: what is the third row of $R_{n\to z}$? If we look at the inverse matrix instead: $R_{z\to n}$, we can immediately see that its third column has to be n, because

$$R_{z\to n} \left(\begin{smallmatrix} 0 \\ 0 \\ 1 \end{smallmatrix}\right) = n$$

by construction. Since rotations are orthogonal matrices, the inverse is equal to the transposed, so we can deduce that the third row of $R_{n\to z}$ is the same as the third column of $R_{z\to n}$, that is: n. Thus with $n = \left(\begin{smallmatrix} n_x \\ n_y \\ n_z \end{smallmatrix}\right)$ we get:

$$L = s_0^z \, S_0 \, (v) - s_2^z \, n_y \, S_1 \, (v) + s_2^z \, n_z \, S_2 \, (v) - s_2^z \, n_x \, S_3 \, (v)$$

So the SH coefficients of the clamped cosine lobe along n are:

$$s_0^n = s_0^z = \frac{\sqrt{\pi}}{2} \tag{26}$$

$$s_1^n = -s_2^z \, n_y = -\sqrt{\frac{\pi}{3}} \, n_y \tag{27}$$

$$s_2^n = s_2^z \, n_z = \sqrt{\frac{\pi}{3}} \, n_z \tag{28}$$

$$s_1^n = -s_2^z \, n_x = -\sqrt{\frac{\pi}{3}} \, n_x \tag{29}$$

As you can see the coefficients are linear in n, which makes their evaluation quite simple.

Note: "My" values are equal to the ones from Crytek's LPV paper if you premultiply them with the SH constants c_0 and c_1.

5 Geometry Injection

Little reminder: projected area is increasing with squared distance and not linearly (in case anyone forgets, which can lead to stupid bugs).

```
output.surfelArea = 4.0 * posWorld.w * posWorld.w / RSMsize.x / RSMsize.y;
```

That is the real world area of the surface texel in the RSM is 4 / RSMsize.x / RSMsize.y (ie texel size in image space), scaled with the squared distance.

5.1 Double-Sided Injection Problem

Crytek's LPV paper only uses one-sided occlusion. It becomes obvious why, when you look at how to implement double-sided occlusion injection. For a double-sided surface element, you'd inject the surface area twice with flipped normals. Since only 4 SH coefficients are used (which depend linearly on the normal direction except for the ambient zeroth coefficient), this results in the cancellation of all but the ambient term.

Another idea is assuming a certain thickness of the material and using different area ratios. However since the backside of a surface element would be farther away that way, its area ratio would actually be bigger than the front-side's ratio, which would result in a face that is occluding more "in the wrong direction".

6 Propagation

6.1 Main Directions

The main direction of a front face is simply the normal of the face itself. The main direction of a side face is also the direction towards the center of the side face. Eg for the left side face (parallel to the yz-plane with $x = cellsize/2$) we have $center = \begin{pmatrix} cellsize/2 \\ cellsize \\ 0 \end{pmatrix}$, so the normalized direction is $\begin{pmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \\ 0 \end{pmatrix}$.

6.2 Intensity Propagation Correction

Crytek's LPV paper uses the following formula for computing the flux to a neighbouring face:

$$\Phi_f = \int_\Omega I(\omega)V(\omega)\,d\omega$$

and approximates it using the solid angle $\Delta\omega_f = \int_\Omega V(\omega)\, d\omega$ and the central direction ω_c as

$$\Phi_f = \frac{\Delta\omega_f}{4\pi} \cdot I(\omega_c)$$

If we insert the definition of $\Delta\omega_f$ back into the formula we get:

$$\Phi_f = \frac{1}{4\pi} I(\omega_c) \int_\Omega V(\omega)\, d\omega = \frac{1}{4\pi} \int_\Omega I(\omega_c) V(\omega)\, d\omega$$

. If we assume that $I(\omega_c)$ is constant, we immediately see that the division by 4π does not make sense here.

6.3 Reprojection Formula Correction

Crytek's LPV paper uses the formulas:

$$\Phi_f = \int_\Omega \Phi_l \overline{\langle n_l, \omega \rangle}\, d\omega$$

and

$$\Phi_l = \Phi_f / \pi$$

The integrand is exactly the intensity that had to be corrected in 4.1. So now we get:

$$\Phi_f = \int_\Omega \frac{1}{\pi} \Phi_l \overline{\langle n_l, \omega \rangle}\, d\omega = \frac{1}{\pi} \Phi_l \int_\Omega \overline{\langle n_l, \omega \rangle}\, d\omega = \frac{1}{\pi} \Phi_l\, \pi = \Phi_l$$

This also makes more sense because now energy conservation is obeyed during reprojection: now the whole flux that arrives at the face of the neighbouring cell is reprojected into this cell and nothing is lost. However, because of the intensity correction, the net value stays the same. This is just a correction to make the intermediate values physically more correct.

7 Additional Implementation Details

Injecting the VPL into the LPV is done using SV_RenderTargetArrayIndex to select the volume slice in the light propagation volume. For this every texel of the RSM is rendered as one point primitive. A geometry shader is used to set SV_RenderTargetArrayIndex depending on the z value of the point's position.