

1 Ray tracing

1.1 Introduction

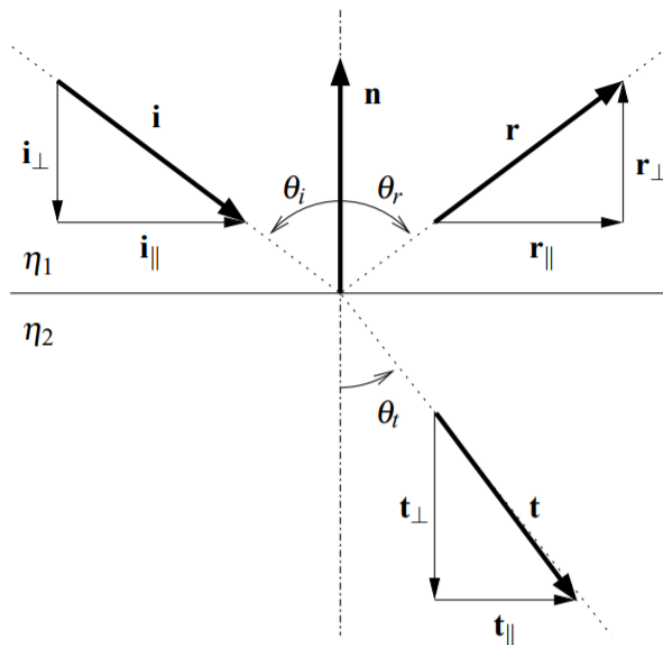


Figure 1: Scheme

In the figure we have a interface between two materials with different refractive indices n_1 and n_2 . The direction vector of the incident ray is \mathbf{i} and we assume this vector is normalized. The normalized direction vectors of the reflected and transmitted rays are \mathbf{r} and \mathbf{t} and will be calculated. The normal vector \mathbf{n} , orthogonal to the interface and pointing towards the first material n_1 .

$$|\mathbf{i}| = |\mathbf{r}| = |\mathbf{t}| = |\mathbf{n}| \quad (1)$$

The direction vectors of the rays can be split in components orthogonal and parallel to the interface. Taking the generic vector \mathbf{v} , we have

$$\mathbf{v}_\perp = \frac{\mathbf{v} \cdot \mathbf{n}}{|\mathbf{n}|^2} \mathbf{n} = (\mathbf{v} \cdot \mathbf{n}) \mathbf{n} \quad (2)$$

$$\mathbf{v}_\parallel = \mathbf{v} - \mathbf{v}_\perp \quad (3)$$

$$|\mathbf{v}|^2 = |\mathbf{v}_\parallel|^2 + |\mathbf{v}_\perp|^2 \quad (4)$$

$$\cos \theta_v = \frac{|\mathbf{v}_\perp|}{|\mathbf{v}|} = |\mathbf{v}_\perp| \quad (5)$$

$$\sin \theta_v = \frac{|\mathbf{v}_\parallel|}{|\mathbf{v}|} = |\mathbf{v}_\parallel| \quad (6)$$

1.2 Reflection

According to the law of reflection $\theta_r = \theta_i$, and 5 6, thus

$$|\mathbf{r}_\perp| = \cos \theta_r = \cos \theta_i = |\mathbf{i}_\perp| \quad (7)$$

$$|\mathbf{r}_\parallel| = \sin \theta_r = \sin \theta_i = |\mathbf{i}_\parallel| \quad (8)$$

$$\mathbf{r} = \mathbf{r}_\parallel + \mathbf{r}_\perp = \mathbf{i}_\parallel - \mathbf{i}_\perp = \mathbf{i} - 2\mathbf{i}_\perp = \mathbf{i} - 2(\mathbf{i} \cdot \mathbf{n})\mathbf{n} \quad (9)$$

And we can find that the reflection vector \mathbf{r} is also nomarlized.

1.3 Refraction

According to Snell's Law for the refraction case,

$$n_1 \sin \theta_i = n_2 \sin \theta_t \quad (10)$$

We leave out the total internal reflection case first here. With equations 5 6, we have

$$n_1 |\mathbf{i}_\parallel| = n_2 |\mathbf{t}_\parallel| \quad (11)$$

$$\mathbf{t}_\parallel = \frac{n_1}{n_2} \mathbf{i}_\parallel = \frac{n_1}{n_2} (\mathbf{i} + \cos \theta_i \mathbf{n}) \quad (12)$$

$$\mathbf{t}_\perp = -\sqrt{1 - |\mathbf{t}_\parallel|^2} \mathbf{n} \quad (13)$$

$$\mathbf{t} = \mathbf{t}_\parallel + \mathbf{t}_\perp = \frac{n_1}{n_2} \mathbf{i} + \left(\frac{n_1}{n_2} \cos \theta_i - \sqrt{1 - |\mathbf{t}_\parallel|^2} \right) \mathbf{n} = \frac{n_1}{n_2} \mathbf{i} + \left(\frac{n_1}{n_2} \cos \theta_i - \sqrt{1 - \sin^2 \theta_t} \right) \mathbf{n} \quad (14)$$

With 10, the refraction vector \mathbf{t} can be conducted.

```
int Refract(double thisn, double nextn, double *l, double *m,
double *n, double ln, double mn, double nn)
{
double nr, cosi, cosi2, rad, cosr, gamma;
if (thisn != nextn)
{
nr = thisn / nextn;
cosi = fabs((*l) * ln + (*m) * mn + (*n) * nn);
cosi2 = cosi * cosi;
if (cosi2 > 1) cosi2 = 1;
rad = 1 - ((1 - cosi2) * (nr * nr));
if (rad < 0) return(-1);
cosr = sqrt(rad);
gamma = nr * cosi - cosr;
(*l) = (nr * (*l)) + (gamma * ln);
(*m) = (nr * (*m)) + (gamma * mn);
(*n) = (nr * (*n)) + (gamma * nn);
}
return 0;
}
```

Figure 2: Refraction code in ZEMAX dll file

1.4 Reference

https://graphics.stanford.edu/courses/cs148-10-summer/docs/2006--degreve--reflection_refraction.pdf

2 Real ray tracing

Here the term real ray tracing refers to the 'case 5' part of the ZEMAX DLL file.

The real ray trace portion of the DLL, type code 5, allows definition of the surface transmittance. The surface transmittance must be a number between 0.0 and 1.0, and indicates the relative fraction of intensity that the ray transmits through the surface. In this context, transmit means "continues on", so for reflective surfaces, the transmitted portion is that which normally reflects to the next surface.

In standard dll file, real ray tracing calculates the intersection point (x, y, z) of the sag surface and the ray with the direction vector (l, m, n) , and the normal vector (ln, mn, nn) at this point.

2.1 Intersection

With the ray starting point $\mathbf{O} = (x_0, y_0, z_0)$ and the normalized ray direction vector $\mathbf{D} = (l, m, n)$, we can express the ray $\mathbf{P} = (x, y, z)$ in parametric form

$$\mathbf{P} = \mathbf{O} + t\mathbf{D} \quad (15)$$

$$\begin{cases} x = x_0 + tl \\ y = y_0 + tm \\ z = z_0 + tn \end{cases} \quad (16)$$

We can first transform the surface sag equation into another form:

$$z = \frac{c\rho^2}{1 + \sqrt{1 - (1 + K)c^2\rho^2}} \quad (17)$$

$$\rho^2 + (1 + K)z^2 - \frac{2z}{c} = 0 \quad (18)$$

$$\text{where } \rho^2 = x^2 + y^2 \quad (19)$$

To find the intersection between this surface and an arbitrary ray, substitute the ray equation in the surface equation:

$$(x_0 + tl)^2 + (y_0 + tm)^2 + (1 + K)(z_0 + tn)^2 = \frac{2(z_0 + tn)}{c} \quad (20)$$

$$(l^2 + m^2 + n^2(1 + K))t^2 + 2\left(x_0l + y_0m + (1 + K)z_0n - \frac{n}{c}\right)t + (x_0^2 + y_0^2 + z_0^2(1 + K) - \frac{2z_0}{c}) = 0 \quad (21)$$

Because the ray direction vector is normalized, so we have $l^2 + m^2 + n^2 = 1$. And if we assume the ray start point satisfies $z_0 = 0$, the equation can be simplified as

$$(1 + n^2K)t^2 + 2\left(x_0l + y_0m - \frac{n}{c}\right)t + (x_0^2 + y_0^2) = 0 \quad (22)$$

$$\text{choose } \begin{cases} a_1 = a = (1 + n^2K) \\ b_1 = -\frac{b}{2} = \frac{n}{c} - x_0l - y_0m \\ c_1 = c = x_0^2 + y_0^2 \end{cases} \quad (23)$$

Here we choose a_1 , b_1 and c_1 which are consistent with that in ZEMAX DLL file 'case 5' part. Therefore the solution is

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{b_1 \pm \sqrt{b_1^2 - a_1c_1}}{a_1} = \frac{c_1}{b_1 \mp \sqrt{b_1^2 - a_1c_1}} \quad (24)$$

We then can calculate the intersection using equation 16

2.2 Normal vector

First, we will derive the normal vector form of an arbitrary point on an arbitrary surface. We consider the surface $U(x, y, z) = k$, it is said to be smooth if the gradient $\nabla U = \langle U_x, U_y, U_z \rangle$ is continuous and non-zero at any point on the surface. Equivalently, we often write:

$$\nabla U = U_x \mathbf{e}_x + U_y \mathbf{e}_y + U_z \mathbf{e}_z \quad (25)$$

$$\text{where } \mathbf{e}_x = \langle 1, 0, 0 \rangle, \mathbf{e}_y = \langle 0, 1, 0 \rangle, \mathbf{e}_z = \langle 0, 0, 1 \rangle \quad (26)$$

Suppose that $\mathbf{r}(t) = \langle x(t), y(t), z(t) \rangle$ lies on a smooth surface $U(x, y, z) = k$. Applying the derivative with respect to t to both sides of the equation of the surface yields

$$\frac{dU}{dt} = \frac{d}{dt} k = 0 \quad (27)$$

$$\nabla U \cdot \mathbf{v} = 0 \quad (28)$$

$$\text{where } \mathbf{v} = \langle x'(t), y'(t), z'(t) \rangle \quad (29)$$

\mathbf{v} is tangent to the surface because it is tangent to a curve on the surface, which implies that ∇U is orthogonal to each tangent vector \mathbf{v} at a given point on the surface. We thus say that the gradient ∇U is normal to the surface $U(x, y, z) = k$ at each point on the surface. To get the normalized normal vector, we can simply divide the gradient ∇U with its length.

Have known the way to calculate the normal vector, we come back to our case, where we want to calculate the normal vector at the intersection point that we have already got in the former part.

$$U(x, y, z) = \rho^2 + (1 + K)z^2 - \frac{2z}{c} = 0 \quad (30)$$

$$\frac{dU}{d\rho} = 2\rho \quad (31)$$

$$\Rightarrow \frac{dU}{dx} = 2x, \frac{dU}{dy} = 2y \quad (32)$$

$$\frac{dU}{dz} = 2(1 + K)z - \frac{2}{c} \quad (33)$$

$$|U| = \sqrt{\left(\frac{dU}{dx}\right)^2 + \left(\frac{dU}{dy}\right)^2 + \left(\frac{dU}{dz}\right)^2} \quad (34)$$

$$= \sqrt{4\rho^2 + 4\left((1 + K)z - \frac{1}{c}\right)^2} \quad (35)$$

$$= 2\sqrt{-(1 + K)z^2 + \frac{2z}{c} + \left((1 + K)z - \frac{1}{c}\right)^2} \quad (36)$$

$$= 2\sqrt{K(1 + K)z^2 - \frac{2k}{c}z + \frac{1}{c^2}} \quad (37)$$

$$= \frac{2}{c}\sqrt{K(1 + K)c^2z^2 - 2kcz + 1} \quad (38)$$

And we can get the normalized normal vector by dividing the $\nabla U = \langle U_x, U_y, U_z \rangle$ with its length $|U|$, which is consistent with the code in DLL file.

```

case 5:
/* ZEMAX wants a real ray trace to this surface */
if (FD->cv == 0.0)
{
    UD->ln = 0.0;
    UD->mn = 0.0;
    UD->nn = -1.0;
    if (Refract(FD->n1, FD->n2, &UD->l, &UD->m, &UD->n, UD->ln, UD->mn, UD->nn)) return(-FD->surf);
    return(0);
}
/* okay, not a plane. */
a = (UD->n) * (UD->n) * FD->k + 1;
b = ((UD->n)/FD->cv) - (UD->x) * (UD->l) - (UD->y) * (UD->m);
c = (UD->x) * (UD->x) + (UD->y) * (UD->y);
rad = b * b - a * c;
if (rad < 0) return(FD->surf); /* ray missed this surface */
if (FD->cv > 0) t = c / (b + sqrt(rad));
else t = c / (b - sqrt(rad));
(UD->x) = (UD->l) * t + (UD->x);
(UD->y) = (UD->m) * t + (UD->y);
(UD->z) = (UD->n) * t + (UD->z);
UD->path = t;
zc = (UD->z) * FD->cv;
rad = zc * FD->k * (zc * (FD->k + 1) - 2) + 1;
casp = FD->cv / sqrt(rad);
UD->ln = (UD->x) * casp;
UD->mn = (UD->y) * casp;
UD->nn = ((UD->z) - ((1/FD->cv) - (UD->z) * FD->k)) * casp;
if (Refract(FD->n1, FD->n2, &UD->l, &UD->m, &UD->n, UD->ln, UD->mn, UD->nn)) return(-FD->surf);
break;

```

Figure 3: Real ray tracing code in ZEMAX dll file 'case 5'

2.3 Reference

ZEMAX manual-Chapter11(SurfaceType)-UserDefined

<https://www.cl.cam.ac.uk/teaching/1999/AGraphHCI/SMAG/node2.html>

<https://www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes/ray-sphere-intersection>

<http://math.etsu.edu/multicalc/prealpha/chap3/chap3-6/printversion.pdf>