

Equations for Ray Tracing Implementation

by Ilja Rukin

1 Physical based rendering

The light intensity (radiance) L_o emitted from point \vec{x} in direction $\vec{\omega}_o$ can be calculated with the rendering equation

$$L_o(\vec{x}, \vec{\omega}_o) = L_e(\vec{x}, \vec{\omega}_o) + \int_{\Omega} f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_o) L_i(\vec{x}, \vec{\omega}_i) (\vec{n} \cdot \vec{\omega}_i) d\omega_i \quad (1)$$

where $L_e(\vec{x}, \vec{\omega}_o)$ is the light emitted from a light source in \vec{x} . The Integral sums all incoming light rays $L_i(\vec{x}, \vec{\omega}_i)$ over a hemisphere Ω weighted with the face orientation $\cos(\theta) = (\vec{n} \cdot \vec{\omega}_i)$ and the contribution of the light-ray by the BRDF $f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}_o)$.

2 space partitioning with by kd-Trees with help of surface area heuristic (SAH)

In this section I explain, how kd-Trees for fast Ray Traversal were implemented.

Optimal depth of three according to Greg Humphreys $H = 8 + 1, 3 \log(\#triangles)$.

3 Rays

3.1 Ray

A ray is described by his origin and direction. In this implementation the direction vector is normalized.

$$\vec{r}(t) = \vec{o} + t \cdot \vec{d} \quad (2)$$

3.2 Rays Fisheye (Perspective)

This function generates Rays with the same origin and equally distributed across the two angles ϕ, θ in spherical Coordinates. The angles are bounded within fov_horizontal (left to right), float fov_vertical (top to bottom). example:

```
Vec3f origin = Vec3f(-1, -2, 1);
float fov_horizontal[3] = { 90, 30, -30 };
float fov_vertical[3] = { 70, 110, 150 };
```

The ray origin is given whereas direction is calculated by:

$$\vec{d} = \begin{pmatrix} \sin(\theta) \cos(\phi) \\ \sin(\theta) \sin(\phi) \\ \cos(\theta) \end{pmatrix} \quad (3)$$

3.3 Rays Parallel (Orthographic)

This function generates Rays with the same direction and linearly shifted origin across range_horizontal (left to right), range_vertical (top to bottom). As example:

```
Vec3f origin = Vec3f(0, -1, 0);
float rotation_horizontal = 110;
float rotation_vertical = 110;
float range_horizontal[3] = { -2, 0, 2 };
float range_vertical[3] = { -2, 0, 2 };
```

The direction is computed from ϕ, θ as previously. This results in a viewing direction along a the z-axis in a rotated Coordinate System. The x- and y- axis are rotated together with the z-axis to be the supporting vectors for shifting of the ray origin. They are calculated this way:

$$\vec{vec1} = \begin{pmatrix} \cos(\theta) \cos(\phi) \\ \cos(\theta) \sin(\phi) \\ -\sin(\theta) \end{pmatrix} \quad (4)$$

$$\vec{vec2} = \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix} \quad (5)$$

Than the supplied oriding vector is shifted by these calculated vectors in the range of range_horizontal, range_vertical.

3.4 Rays form a thin lens

The lens profile is given by $f(x, y) = R_1 - \sqrt{R_1^2 - x_n^2 - y_n^2} - l_1$ for the front face and $f(x, y) = \sqrt{R_2^2 - x_n^2 - y_n^2} - R_2 + l_2$ for the rear face, where R_i are the diameters of the lens halves and l_i their thickness. x_n, y_n are the coordinates from the lens center.

A ray intersects the front face of the lens at:

$$\vec{o} + t \cdot \vec{d} = \begin{pmatrix} x \\ y \\ R_1 - \sqrt{R_1^2 - x_n^2 - y_n^2} - l_1 \end{pmatrix} \quad (6)$$

$$\Rightarrow (o_3 + td_3 + l_1 - R_1) = \sqrt{R_1^2 - (o_1 + td_1)^2 - (o_2 + td_2)^2} \quad (7)$$

$$\Leftrightarrow t^2 + t \cdot \underbrace{\frac{2(\vec{d}\vec{o} + d_3(l_1 - R_1))}{\vec{d}^2}}_p + \underbrace{\frac{\vec{o}^2 - R_1^2 + (l_1 - R_1)^2 + 2o_3(l_1 - R_1)}{\vec{d}^2}}_q = 0 \quad (8)$$

$$t = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q} \quad (9)$$

$$= -\frac{\vec{d}\vec{o} + d_3(l_1 - R_1)}{\vec{d}^2} \pm \sqrt{\frac{(\vec{d}\vec{o} + d_3(l_1 - R_1))^2}{\vec{d}^4} - \frac{\vec{o}^2 - R_1^2 + (l_1 - R_1)^2 + 2o_3(l_1 - R_1)}{\vec{d}^2}} \quad (10)$$

The hit-point becomes the new ray origin and the ray direction is determined according to the law of refraction. The same goes for the exit of the ray at the rear side of the lens. Only in this instance the Hit point has to be calculated with a positive sign before the square root.

Lens normal:

$$\vec{n} = \begin{pmatrix} -\partial f / \partial x \\ -\partial f / \partial y \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{x}{\sqrt{R^2 - x^2 - y^2}} \\ \frac{y}{\sqrt{R^2 - x^2 - y^2}} \\ 1 \end{pmatrix} \quad (11)$$

$$(12)$$

3.5 lens array

A planar lens array can be described by small lenses shifted by translation vectors \vec{t}_h and \vec{t}_v .

In the small angle approximation suppose one ray can only pass through the lens, which is directly in front of it. This approximation is valid, since it resembles the viewing angle limit and at greater angles higher order images will appear. To calculate the resulting ray direction the offset of the ray from the center of the lens directly in front of it has to be determined. This can be done by dividing the x-position of the pixel by the x-component of \vec{t}_h and afterwards dividing the y-position by \vec{t}_v y-component.

4 Intersections

Triangles

Triangles are described by their 3 vertices $\vec{e}, \vec{f}, \vec{g}$ in barycentric coordinates:

$$\vec{p}_t(\alpha, \beta) = \vec{e} + \alpha(\vec{f} - \vec{e}) + \beta(\vec{g} - \vec{e}) \quad (13)$$

Note: The same equation can be used to linearly interpolate normals (for phong shading) or texture coordinates (for texture mapping) at the three vertices of the triangle.

We set the point on a triangle $\vec{p}_t(\alpha, \beta)$ equal to a point of a ray $\vec{p}_r(t)$ to get the equation for the intersection. The principle is described by Möller & Trumbore [1].

$$\vec{p}_r(t) = \vec{p}_t(\alpha, \beta) \quad (14)$$

$$\vec{o} + t \cdot \vec{d} = \vec{e} + \alpha(\vec{f} - \vec{e}) + \beta(\vec{g} - \vec{e}) \quad (15)$$

$$\Leftrightarrow \vec{o} - \vec{e} = \underbrace{\begin{pmatrix} \vec{d} & \vec{f} - \vec{e} & \vec{g} - \vec{e} \end{pmatrix}}_A \begin{pmatrix} t \\ \alpha \\ \beta \end{pmatrix} \quad (16)$$

The solution for the parameters is:

$$\begin{pmatrix} t \\ \alpha \\ \beta \end{pmatrix} = A^{-1}(\vec{o} - \vec{e}) \quad (17)$$

$$= \frac{1}{\det(A)} \begin{pmatrix} + \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & - \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & + \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ - \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & + \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & - \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} \\ + \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} & - \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & + \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix}^T \begin{pmatrix} o_1 - e_1 \\ o_2 - e_2 \\ o_3 - e_3 \end{pmatrix} \quad (18)$$

$$= e \begin{pmatrix} a_{22}a_{33} - a_{23}a_{32} & a_{13}a_{32} - a_{12}a_{33} & a_{12}a_{23} - a_{13}a_{22} \\ a_{23}a_{31} - a_{21}a_{33} & a_{11}a_{33} - a_{13}a_{31} & a_{13}a_{21} - a_{11}a_{23} \\ a_{21}a_{32} - a_{22}a_{31} & a_{12}a_{31} - a_{11}a_{32} & a_{11}a_{22} - a_{12}a_{21} \end{pmatrix} m a^T \begin{pmatrix} o_1 - e_1 \\ o_2 - e_2 \\ o_3 - e_3 \end{pmatrix} \quad (19)$$

$$= e \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \quad (20)$$

with

$$e = \frac{1}{a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{12}a_{21}a_{33} - a_{11}a_{23}a_{32}} \quad (21)$$

$$= \frac{1}{a_{11}(a_{22}a_{33} - a_{23}a_{32}) + a_{12}(a_{23}a_{31} - a_{21}a_{33}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})} \quad (22)$$

$$= \frac{1}{a_{11}B_{11} + a_{12}B_{21} + a_{13}B_{31}} \quad (23)$$

The parameter t is equal to the distance to the triangle plane (since the ray direction vector \vec{d} is normalized). If $\alpha + \beta \leq 1$, then the intersection lies within the triangle and a hit occurred. The parameters α, β can be used with the texture positions to calculate the texture at the hit-point or together with supplied normal vectors for smooth shading.

The normal vector is:

$$\vec{n}_t = (\vec{f} - \vec{e}) \times (\vec{g} - \vec{e}) \quad (24)$$

$$= \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \end{pmatrix} \times \begin{pmatrix} a_{13} \\ a_{23} \\ a_{33} \end{pmatrix} \quad (25)$$

$$= \begin{pmatrix} a_{22}a_{33} - a_{32}a_{23} \\ a_{32}a_{13} - a_{12}a_{33} \\ a_{12}a_{23} - a_{22}a_{13} \end{pmatrix} \quad (26)$$

$$= \begin{pmatrix} B_{11} \\ B_{12} \\ B_{13} \end{pmatrix} \quad (27)$$

Spheres

A sphere is described by its center \vec{c} and radius r . A ray \vec{r} hits a sphere, if the following equation is fulfilled:

$$(\vec{r} - \vec{c})^2 = R^2 \quad (28)$$

Inserting the formula for the ray gives the intersection point

$$(\vec{o} + t \cdot \vec{d} - \vec{c})^2 = R^2 \quad (29)$$

$$t^2 \cdot \underbrace{+t \cdot \frac{2\vec{d}(\vec{o} - \vec{c})}{\vec{d}^2}}_p + \underbrace{\frac{(\vec{o} - \vec{c})^2 - R^2}{\vec{d}^2}}_q = 0 \quad (30)$$

$$\Rightarrow t = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q} \quad (31)$$

$$= -\frac{\vec{d}(\vec{o} - \vec{c})}{\vec{d}^2} \pm \sqrt{\frac{(\vec{d}(\vec{o} - \vec{c}))^2}{\vec{d}^4} - \frac{(\vec{o} - \vec{c})^2 - R^2}{\vec{d}^2}} \quad (32)$$

For the implementation test if square root is ≥ 0 and pick the smallest t as the hitpoint.

The normal vector is:

$$\vec{n}_s = \vec{p}_s - \vec{c} \quad (33)$$

do not forget to normalize it.

5 Interaction

5.1 Reflection

The direction of the reflection ray can be calculated from the incoming ray \vec{r} and the face normal \vec{n} as follows

$$\vec{dr} = 2\vec{r}y + \vec{r} \quad (34)$$

$$= 2 \cos(\theta_1) \vec{n} + \vec{r} \quad (35)$$

$$= 2(-\vec{r} \cdot \vec{n}) \vec{n} + \vec{r} \quad (36)$$

with

$$\vec{r}y = \vec{n} \cos(\theta_1) = (-\vec{r} \cdot \vec{n}) \vec{n} \quad (37)$$

$$\vec{rx} = \vec{n} \sin(\theta_1) = \vec{dr} - \vec{r}y = (-\vec{r} \cdot \vec{n}) \vec{n} + \vec{r} \quad (38)$$

$$(39)$$

$$\vec{t}y = \vec{n} \cos(\theta_2) \quad (48)$$

$$= \vec{n} \sqrt{1 - \sin(\theta_2)^2} \quad (49)$$

$$= \vec{n} \sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 \sin(\theta_1)^2} \quad (50)$$

$$= \vec{n} \sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 (1 - \cos(\theta_1)^2)} \quad (51)$$

$$= \vec{n} \sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 (1 - (\vec{r}\vec{n})^2)} \quad (52)$$

$$= \vec{n} \sqrt{1 - \vec{t}x^2} \quad (53)$$

So the transmitted ray is equal to:

$$\vec{d}t = \vec{t}x + \vec{t}y \quad (54)$$

$$= \vec{t}x + \vec{n} \sqrt{1 - \vec{t}x^2} \quad (55)$$

5.3 Diffuse Reflection

In case of perfect diffuse reflection an incident light ray gets randomly reflected. We calculate the direction of the reflected ray with two random numbers u_1, u_2 :

$$\vec{d}r = \begin{pmatrix} \sqrt{1 - u_2^2} \cos(2\pi u_1) \\ \sqrt{1 - u_2^2} \sin(2\pi u_1) \\ u_2 \end{pmatrix} \quad (56)$$

where $\sqrt{1 - u_2^2}$ and u_2^2 form a circle and $2\pi u_1$ sample the angle ϕ and produce a random vector on a half-sphere.

5.4 Amount of reflections and transmission

The amount of reflektion and transmission is described by fresnel laws. To avoid the long computation and averaging over the two polarizations we can use Schlicks approximation.

$$R_0 = \left(\frac{\eta_1 - \eta_2}{\eta_1 + \eta_2} \right)^2 \quad (57)$$

$$Re = R_0 + (1 - R_0)(1 - \cos(\theta_1)^2) \quad (58)$$

$$Tr = 1 - Re \quad (59)$$

The factor R_0 is equal to the minimum reflectivity of the medium, i.e. the reflectivity at direct incidence $\theta = 0$ of light.

Literatur

- [1] Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. Journal of graphics tools, 2(1):21–28, 1997.
- [2] Bram De Greve. Reflections and refractions in ray tracing.