Vector Algebra for Ray Tracing

Dr. Thomas W. Sederberg

14 February 2005

1 Specifying the Camera Position

Ray tracing is a computer graphics rendering method that is capable of providing a very high degree of realism, including mathematically precise shadows and transparency.

The two geometric elements of ray tracing are a definition of the scene and a description of thw camera location and orientation.

The scene might include such elements as spheres, polygons, and surface patches, all given by their defining mathematical equations. For example, a sphere is defined by specifying its center and radius. A triangle is defined by specifying Cartesian coordinates for its vertices, plus possibly normal vectors at those vertices (for shading calculation).

The camera (or eye location) is specified by giving its Cartesian coordinates, a look-at point, and an up-vector. In addition, a field of view is specified in terms of a angle or a window width.

In Figure 1, the camera is the point **E** (for eye), the look-at point is **A**, the up vector is labelled **Up**, and the width and heighth of the window are H and V (for horizontal and vertical dimension).

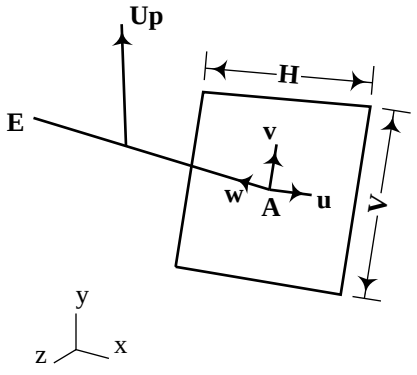


Figure 1: Eye coordinate system

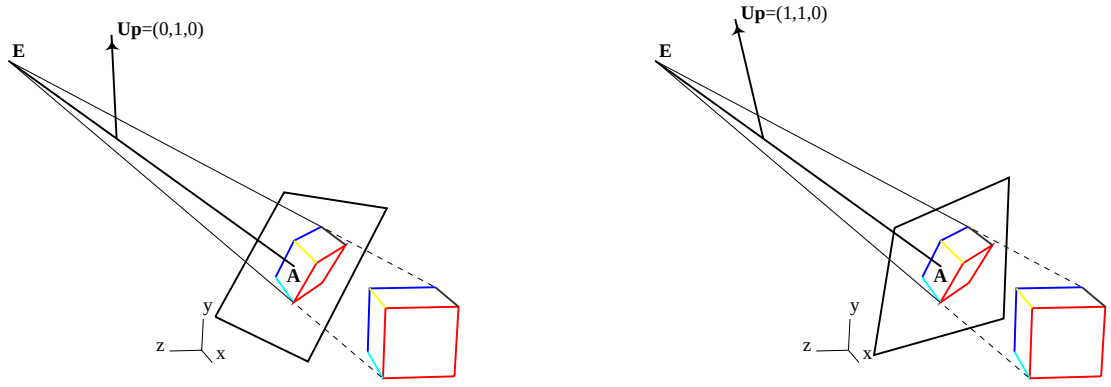
The eye (or camera) coordinate system is defined using three mutually perpendicular unit vectors **u**, **v**, and **w**. Since the only data that is given are the points **E** and **A** and the vector **Up**, the unit vectors **u**, **v**, and **w** must be computed, as follows:

$$\mathbf{w} = \frac{\mathbf{E} - \mathbf{A}}{|\mathbf{E} - \mathbf{A}|}$$

$$\mathbf{u} = \frac{\mathbf{Up} \times (\mathbf{E} - \mathbf{A})}{|\mathbf{Up} \times (\mathbf{E} - \mathbf{A}|}$$

$$\mathbf{v} = \mathbf{w} \times \mathbf{u}$$

The *projection plane* is perpendicular to line **E**—**A**. Figure 2 shows a cube projected onto the view plane. The up vector allows you to rotate the camera about the line **E**—**A**. The projection of the up vector onto the view plane in parallel to **v**. Notice that it is not required that the up vector itself is parallel to **v**; this is a convenience for the use. Typically, the up vector is just chosen to be one of the Cartesian coordinate unit vectors $(1, 0, 0)$, $(0, 1, 0)$, or $(0, 0, 1)$. The most common up vector is $(0, 1, 0$ because that is how we normally view the world (unless we are lying down).



a. **UP** $= (0, 1, 0)$ b. **UP** $= (1, 1, 0)$

Figure 2: Projection Plane and Up Vectors

Figure 3 illustrates the volume known as the viewing frustrum

2 Computing the Rays

Ray tracing is sometimes thought of as "screen-door viewing." Each hole in the screen corresponds to a pixel. What color you see in each hole is the color of the corresponding pixel. Of course, in real life, each screen hole does not generally contain a single color, so we go with the color at the center of the hole, or do some color averaging (which is called anti-aliasing).

To write a ray tracer, we must pass a ray through each hole in the screen and determine what the ray hits first. This section explains how to compute the equation of a ray through each pixel.

Referring to Figure 4, imagine that our viewing plane is actually a screen with x_{res} columns and y_{res} rows of holes. The hole in the lower-left corner of the screen is $(i, j) =$

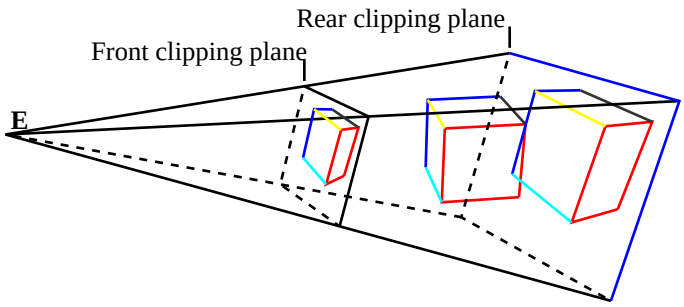


Figure 3: Viewing Frustrum

$(0,0)$. The width of the screen is L_u and the heighth of the screen is L_v. Then the (x,y,z) coordinates of the center of hole (i,j) are given by:

$$\mathbf{P}_{ij} = \mathbf{A} + c_u\mathbf{u} + c_v\mathbf{v} \tag{1}$$

where

$$c_u = \left(\frac{2i+1}{2x_{res}} - \frac{1}{2}\right) L_u; \qquad c_v = \left(\frac{2j+1}{2y_{res}} - \frac{1}{2}\right) L_v. \tag{2}$$

Then the ray through hole (i,j) is simply

$$\mathbf{E} + \frac{\mathbf{P}_{ij} - \mathbf{E}}{|\mathbf{P}_{ij} - \mathbf{E}|} t \tag{3}$$

2.1 OpenGL Compatability

The equations we just derived are equivalent to the view frustum defined in OpenGL with the following constraints:

1. left $= -L_u$.
2. bottom $= -L_v$.
3. right $= L_u$.
4. top $= L_v$.
5. $z_{near} = |\mathbf{E} - \mathbf{A}|$.

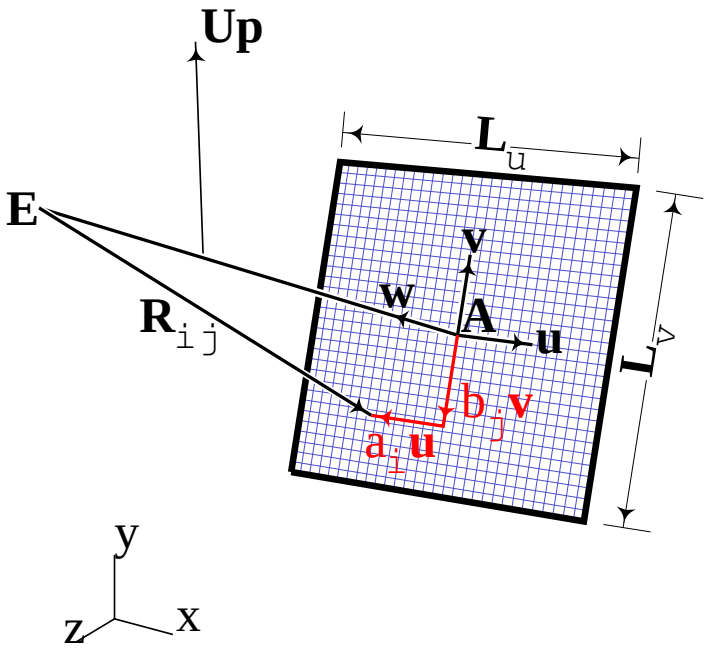


Figure 4: Computing the ray through pixel i, j

3 Equation of a plane containing a triangle

Given a triangle with vertices $\mathbf{P}_1 = (x_1, y_1, z_1)$, $\mathbf{P}_2 = (x_2, y_2, z_2)$, and $\mathbf{P}_3 = (x_3, y_3, z_3)$, the normal vector to this triangle is

$$\mathbf{n} = (a, b, c) = (\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_1).$$

The equation of the plane the the triangle lies in is:

$$ax + by + cz + d = 0$$

where a, b, c are the components of the normal vector. The value of d can be solved for by substituting any of the triangle vertices into the plane equation:

$$d = -(ax_1 + by_1 + cz_1) = -(ax_2 + by_2 + cz_2) = -(ax_3 + by_3 + cz_3).$$

4 Intersecting a ray and a triangle

To compute the point at which a ray intersects a triangle, first find the equation of the plane that contains the triangle: $ax + by + cz + d = 0$. Then substitute the equation of the ray into the equation for the plane:

$$a(x_0 + x_1 t) + b(y_0 + y_1 t) + c(z_0 + z_1 t) + d = 0$$

or

$$(ax_1 + by_1 + cz_1)t + (ax_0 + by_0 + cz_0 + d) = 0$$

from which

$$t = -\frac{ax_0 + by_0 + cz_0 + d}{ax_1 + by_1 + cz_1}.$$

Substitute this value for t into the ray equation to compute the Cartesian coordinates of the point at which the ray intersects the plane containing the triangle. Of course, that intersection point might be outside of the triangle. To determine if the point is inside or outside of the triangle, compute the barycentric coordinates of the point.

5 Barycentric Coordinates

Barycentric coordinates are sometimes called "triangle coordinates" because they are defined with respect to the three vertices triangle. We will denote the barycentric coordinates as (s, t, u). Figure 5 illustrates the barycentric coordinates for a point $\mathbf{P}$ inside of triangle $\mathbf{P}_s\mathbf{P}_t\mathbf{P}_u$. Refering to the triangles in that figure, s is the area of triangle $\mathbf{P}\mathbf{P}_t\mathbf{P}_u$ divided

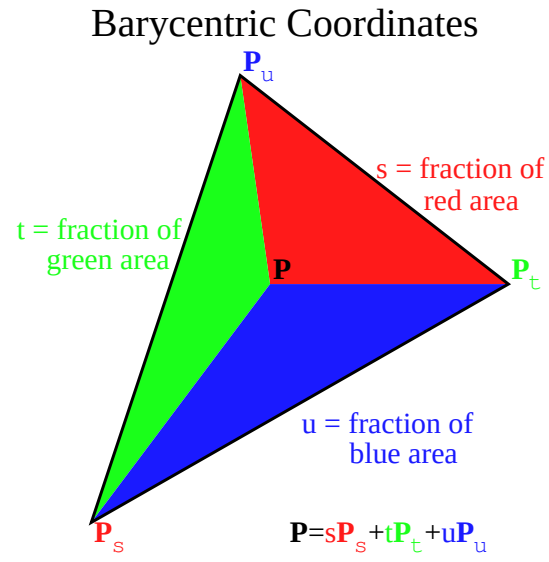


Figure 5: Barycentric Coordinates

by the area of triangle $\mathbf{P}_s\mathbf{P}_t\mathbf{P}_u$, and likewise for u and v. Expressed using vector algebra,

$$s = \frac{(\mathbf{P}_t - \mathbf{P}) \times (\mathbf{P}_u - \mathbf{P})}{(\mathbf{P}_t - \mathbf{P_s}) \times (\mathbf{P}_u - \mathbf{P_s})}; \quad t = \frac{(\mathbf{P}_u - \mathbf{P}) \times (\mathbf{P}_s - \mathbf{P})}{(\mathbf{P}_t - \mathbf{P_s}) \times (\mathbf{P}_u - \mathbf{P_s})}; \quad u = \frac{(\mathbf{P}_s - \mathbf{P}) \times (\mathbf{P}_t - \mathbf{P})}{(\mathbf{P}_t - \mathbf{P_s}) \times (\mathbf{P}_u - \mathbf{P_s})}. \tag{4}$$

Note the following properties of barycentric coordinates:
1. $s + t + u \equiv 1$.
2. $s, t, u \geq 0$ if and only if $\mathbf{P}$ is inside of the triangle.
3. $\mathbf{P} = s\mathbf{P}_s + t\mathbf{P}_t + u\mathbf{P}_u$.
4. If you do an orthographic projection of the triangle along with any point $\mathbf{P}$ inside the triangle, the barycentric coordinates of the projection are the same as the barycentric coordinates of the original triangle. Thus, if you simply set $z = 0$ for the triangle vertices and the point, you can compute barycentric coordinates in 2D. (Of course, depending on the triangle, it might be smarter to project to the $x = 0$ or $y = 0$ plane.)

6 Computing the intersection between a ray and a sphere

The equation of a sphere is

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 - r^2 = 0 \tag{5}$$

where (x_c, y_c, z_c) is the center of the sphere and r is the radius of the sphere. The equation of a ray is:

$$\mathbf{R}(t) = (x(t), y(t), z(t)) = (x_0, y_0, z_0) + (x_1, y_1, z_1)t = (x_0 + x_1t, y_0 + y_1t, z_0 + z_1t). \tag{6}$$

The ray intersects the sphere in two points. Those two points might be complex, real, or two identical points (in the case of a tangency). Those two points can be computed as follows. First, substitute the equation for the ray into the equation for the sphere.

$$(x(t) - x_c)^2 + (y(t) - y_c)^2 + (z(t) - z_c)^2 - r^2 = 0.$$

or

$$((x_0 + x_1t - x_c)^2 + (y_0 + y_1t - y_c)^2 + (z_0 + z_1t - z_c)^2 - r^2 = 0.$$

We can regroup these terms and write the equation as a polynomial in t:

$(x_1^2{+}y_1^2{+}z_1^2)t^2{+}2[(x_0{-}x_c)x_1{+}(y_0{-}y_c)y_1{+}(z_0{-}z_c)z_1]t{+}[(x_0{-}x_c)^2{+}(y_0{-}y_c)^2{+}(z_0{-}z_c)^2{-}r^2] = 0$

or $at^2 + bt + c = 0$ where

$a = (x_1^2{+}y_1^2{+}z_1^2)$; $b = 2[(x_0{-}x_c)x_1{+}(y_0{-}y_c)y_1{+}(z_0{-}z_c)z_1]$; $c = [(x_0{-}x_c)^2{+}(y_0{-}y_c)^2{+}(z_0{-}z_c)^2{-}r^2]$.

We next solve for the two roots of this polynomial using the quadratic formula:

$$t_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad t_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Now, t_1 and t_2 are the two values of t at which the ray hits the sphere. If $b^2 - 4ac < 0$, the ray does not hit the sphere (at least, in any real points). If $b^2 - 4ac = 0$, then the ray is tangent to the sphere. If $t_1 < 0$, then the intersection point $\mathbf{R}(t_1)$ is behind ray origin (x_0, y_0, z_0).

Once t_1 and t_2 are known, the Cartesian coordinates of the intersection points are simply $\mathbf{R}(t_1)$ and $\mathbf{R}(t_2)$.