# Normal Vector Transformation

Robert Joan-Arinyo
Computer Science Department
Universitat Politècnica de Catalunya
Barcelona, Catalonia

When lighting is enabled in OpenGL, the normal vectors are used to determine how much light is received at the specified vertex or surface. This lighting processing is performed at eye coordinate space, therefore, normal vectors in object coordinates must be also transformed to eye coordinates.

However, normal vectors are transformed in different way as vertices do. We cannot simply multiply the view matrix by the normal. Consider a normal vector (1,0,0) at vertex (0,0,0). If the transformation matrix is simply translating two units up along the Y-axis, then the vertex coordinates will be (0,2,0), as shown in Figure 1. But, the normal should remain the same, (1,0,0) not (1,2,0).

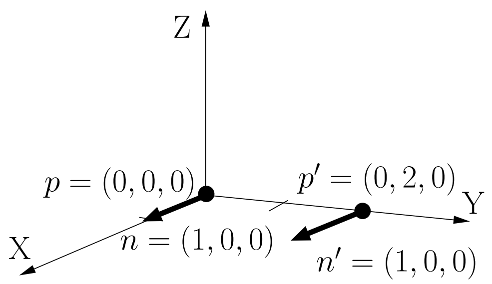


Figure 1: Translating a point $p$ and the associated normal $n$.

Let us see how normal vectors are transformed to eye space. Recall that in 3D, the equation of a plane $\Pi$ is given by the set of points $\vec{v}$ such that

$$< \vec{n}\ \vec{v} >= 0$$

where $\vec{n}$ is the normal to the plane $\Pi$. Refer to Figure 2.

Consider the plane and vertices embedded in a homogeneous space. The normal to the plane would be expressed as $\vec{n} = (n_x, n_y, n_z, n_w)$ and a point

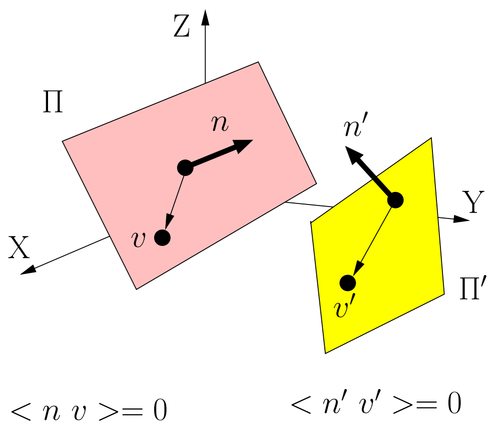


Figure 2: Plane and normal in object coordinates, $< nv >$, and in eye coordinates, $< n'v' >$.

on the plane as $\vec{v} = (x, y, z, w)$. Then the dot product leads us to write the plane equation as

$$n_x x + n_y y + n_z z + n_w w = 0$$

We can write this equation in a general matrix notation as

$$(n_x, n_y, n_z, n_w) \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = 0$$

Now assume that $M$ is the model view transformation matrix. Given that $M^{-1}M$ is the identity matrix, we have that

$$(n_x, n_y, n_z, n_w) M^{-1} M \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = 0$$

holds. Notice that

$$M \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = 0$$

transforms an arbitrari vertex $(x, y, z, w)$ from object coordinates to the eye coordinates, say $(x_{eye}, y_{eye}, z_{eye}, w_{eye})$. Therefore, we have

$$(n_x, n_y, n_z, n_w) M^{-1} (x_{eye}, y_{eye}, z_{eye}, w_{eye}) = 0$$

But this is the equation of a plane, say $\Pi'$, the normal of which is

$$(n_x, n_y, n_z, n_w)M^{-1}$$

Since $\Pi'$ is the plane $\Pi$ transformed by $M$ and the normal to $\Pi$ must remain normal to $\Pi'$, the expression above is the transformation that must be applied to the normal of $\Pi$ to be expressed in the eye coordinates. Thus

$$\begin{pmatrix} nx_{eye} \\ ny_{eye} \\ nz_{eye} \\ nw_{eye} \end{pmatrix} = (nx_{obj}, ny_{obj}, nz_{obj}, nw_{obj})M^{-1}$$

Or, equivalently, converting pre-multiplication to post-multiplication form we have

$$\begin{pmatrix} nx_{eye} \\ ny_{eye} \\ nz_{eye} \\ nw_{eye} \end{pmatrix} = (M^{-1})^T \begin{pmatrix} nx_{obj} \\ ny_{obj} \\ nz_{obj} \\ nw_{obj} \end{pmatrix}$$

# 1 Credits


1. Song Ho Ahn, *OpenGL Normal Vector Transformation*, `http://www.songho.ca/opengl/gl_normaltransform.html`, Visited on November 16, 2015.

2. OpenGL Programming Guide, Appendix F, *Homogeneous Coordinates and Transformation Matrices*, `http://www.glprogramming.com/red/appendixf.html`, Visited on November 16, 2015.