Transforming Normals

Transforming Points

Geometry, or more specifically the points associated with vertices can be transformed using scale, rotate, translate and other operations.

In computer graphics *homogeneous coordinates* and 4×4 matrices are used to represent and implement transformations.

The *model view* matrix is used to combine *modelling* and *viewing* transformations, and to combine individual transformations using matrix multiplication.

So points are transformed as follows

$$\mathbf{P}^{'} = \mathbf{M} \times \mathbf{P}$$

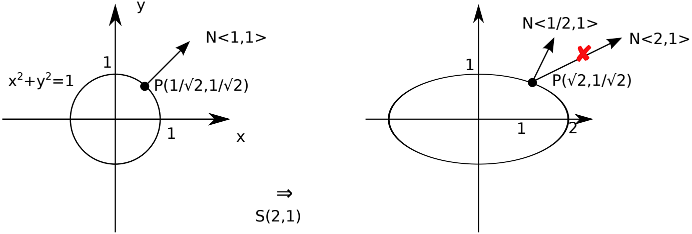
Transforming Points

which in full is

$$\begin{bmatrix} x^{'} \\ y^{'} \\ z^{'} \\ w^{'} \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Transforming Normals

Normal (and tangent) vectors must be transformed using the *normal matrix* instead of the model view matrix, otherwise they are wrong.



Transforming Normals

The normal matrix is the *transpose* of the *inverse* of the model view matrix

$$\mathbf{NM} = (\mathbf{M}^{-1})^T$$

This is a relatively expensive operation. An optimisation which works in most cases calculates the transpose of the inverse of just the upper 3×3 submatrix of the model view matrix.

There are further optimisations which can be made given certain assumptions.