

# Homogeneous Coordinates



**Oregon State**  
University  
**Mike Bailey**

mjb@cs.oregonstate.edu



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/)



**Oregon State**  
University  
Computer Graphics

## Homogeneous Coordinates: Adding a 4<sup>th</sup> Value to an XYZ Triple

We usually think of a 3D point as being represented by a triple: (x,y,z).

Using homogeneous coordinates, we add a 4<sup>th</sup> number: (x,y,z,w)

A graphics system, by convention, performs transformations and clipping using (x,y,z,w) and then divides x, y, and z by w before it uses them.

$$X = \frac{x}{w}, Y = \frac{y}{w}, Z = \frac{z}{w}$$

Thus (1,2,3,1) , (2,4,6,2) , (-1,-2,-3,-1) all represent the same 3D point.

## This Seems Awkward – Why Do It?

3

One reason is that it allows for perspective division within the matrix mechanism. The OpenGL call `glFrustum( left, right, bottom, top, near, far )` creates this matrix:

$$\begin{Bmatrix} x' \\ y' \\ z' \\ w' \end{Bmatrix} = \begin{bmatrix} \frac{2 \cdot \text{near}}{\text{right} - \text{left}} & 0 & \frac{\text{right} + \text{left}}{\text{right} - \text{left}} & 0 \\ 0 & \frac{2 \cdot \text{near}}{\text{top} - \text{bottom}} & \frac{\text{top} + \text{bottom}}{\text{top} - \text{bottom}} & 0 \\ 0 & 0 & \frac{-(\text{far} + \text{near})}{\text{far} - \text{near}} & \frac{-2 \cdot \text{far} \cdot \text{near}}{\text{far} - \text{near}} \\ 0 & 0 & \text{--1} & 0 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix}$$

This gives  $w' = -z$ , which is the necessary divisor for perspective.

## Another Reason is to be able to Represent Points at Infinity

4

This is useful to be able specify a parallel light source by placing the light source location at infinity.

The point  $(1,2,3,1)$  represents the 3D point  $(1,2,3)$

The point  $(1,2,3,.5)$  represents the 3D point  $(2,4,6)$

The point  $(1,2,3,.01)$  represents the point  $(100,200,300)$

So,  $(1,2,3,0)$  represents a point at infinity, but along the ray from the origin through  $(1,2,3)$

Points-at-infinity are used for parallel light sources and some shadow algorithms



## However, When Using Homogeneous Coordinates, You Sometimes Just Need to be able to get a Vector Between Two Points

To get a vector between two homogeneous points, we subtract them:

$$\begin{aligned} (x_b, y_b, z_b, w_b) - (x_a, y_a, z_a, w_a) &= \frac{(x_b, y_b, z_b)}{w_b} - \frac{(x_a, y_a, z_a)}{w_a} \\ &= \frac{(w_a x_b, w_a y_b, w_a z_b) - (w_b x_a, w_b y_a, w_b z_a)}{w_a w_b} \end{aligned}$$

Fortunately, most of the time that we do this, we only want a unit vector in that direction, not the full vector. So, we can ignore the denominator, and just say:

$$\hat{v} = \text{normalize}(w_a x_b - w_b x_a, w_a y_b - w_b y_a, w_a z_b - w_b z_a);$$

```
vec3
VectorBetween( vec4 a, vec4 b )
{
    return normalize( vec3( a.w*b.x - b.w*a.x , a.w*b.y - b.w*a.y , a.w*b.z - b.w*a.z ) );
}
```