OpenGL rendering pipeline & projection matrix (Perspective / orthographic)

Vertex positions $\vec{u}$ $\Rightarrow$ $\boxed{M}$ $\Rightarrow$ $\boxed{V}$ $\Rightarrow$ $\boxed{P}$ $\Rightarrow$ $P \cdot V \cdot M \cdot \vec{u}$ $=$ $\langle x, y, z, w \rangle$

4-vector $u = \langle u_1, u_2, u_3, u_4 \rangle$

M - model matrix (4x4)
(M - model objects in 3-space)

V - view matrix (4x4)
(V - place object of ~~viewer~~ viewer)

Projection matrix (4x4)
P - handle "field-of-view"

A single modelview matrix VM is used instead.

"Perspective Division"

$$\left\langle \frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right\rangle$$

$-1 \le \frac{x}{w} \le 1$
left-to-right on screen

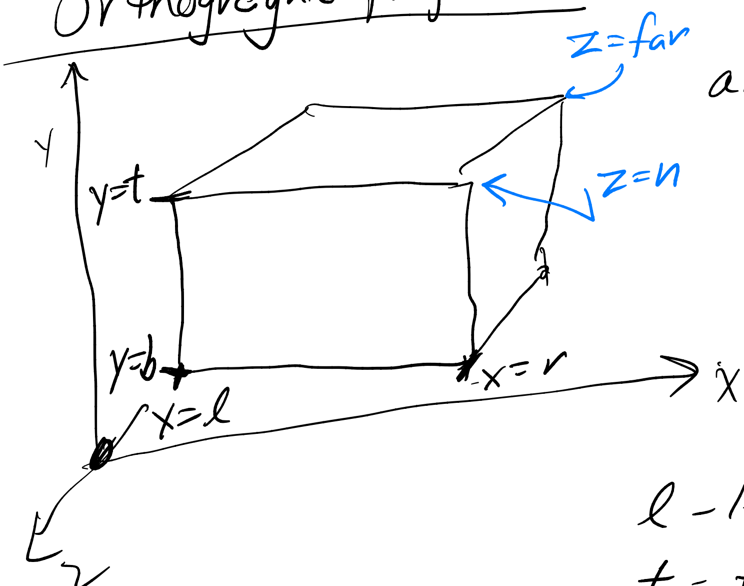
$-1 \le \frac{y}{w} \le 1$
bottom-to-top on screen

$-1 \le \frac{z}{w} \le 1$
depth value
-1 - closest visible
+1 - farthest visible

Items that are too close or too far away are clipped/culled by near clipping or the far clipping plane

Vertex shader has to compute $PVM\vec{u}$ and thus (x, y, z, w).

Orthographic projection: (No perspective)



axis-aligned visible (cube) region in essense, projected towards x, y plane & transformed to be in the 2x2x2 cube $[-1,1]^2$

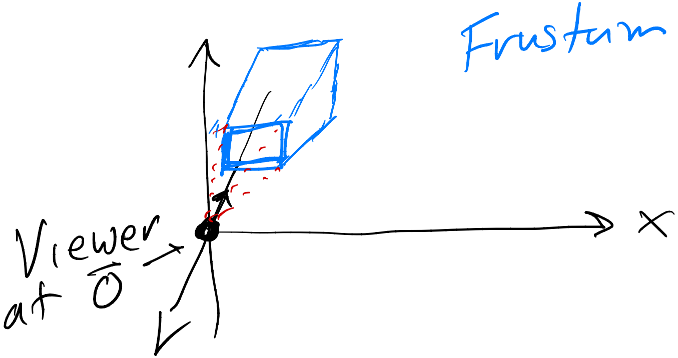
l - left, r - right (bounds x values)
t - top, b - bottom (" y ")
n - near, f - far (" z ")

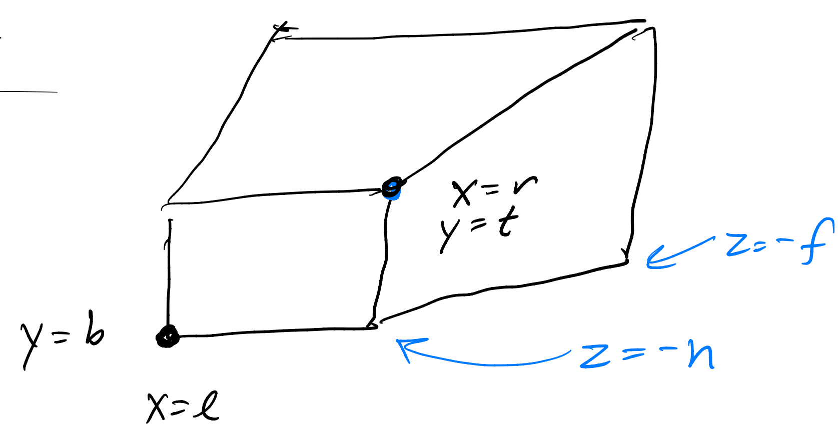
Matrix
$$\begin{pmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{n-f} & \frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= -\frac{n+f}{n-f}$$

P. Set_glOrtho(l, r, t, b, n, f);

Perspective Transformations





P. Set_glFrustum(l, r, b, t, n, f)

P. Set_gluPerspective(θ , aspectRatio , n, f)

centered on z-axis

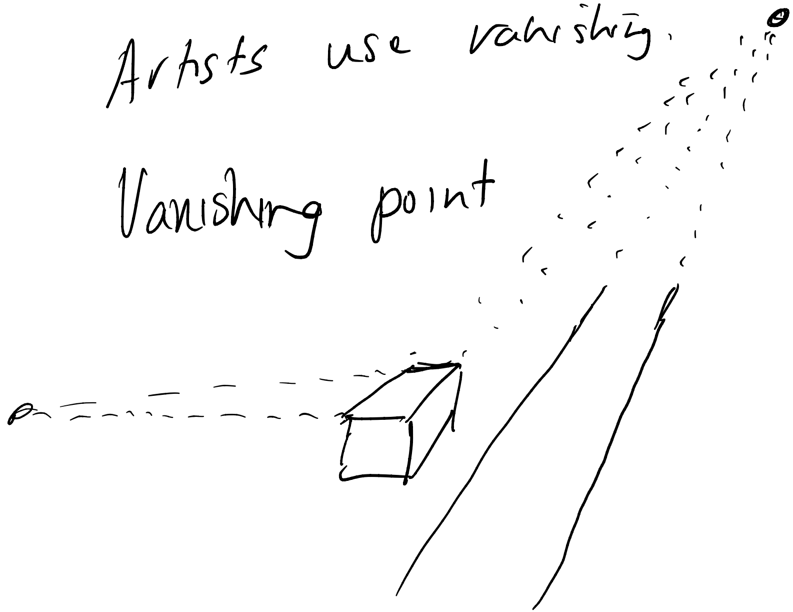
angle between the top & bottom planes

ratio of width to height

Perspective works how?

Artists use vanishing.

Vanishing point



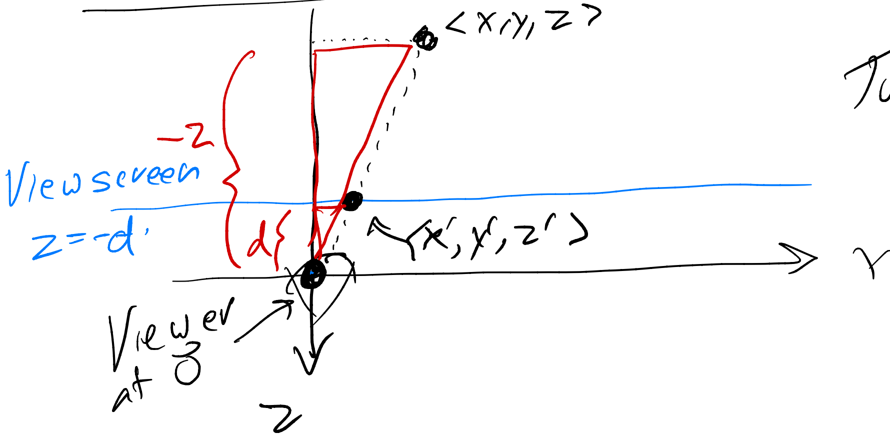
Vanishing point

(Points at infinity)

Road

In computer graphics, we don't do this — just calculate mathematically.

Formulas for perspective:



Top view

$\langle x,y,z\rangle$ is projected towards the viewer onto the $z=-d$ plane to the point $\langle x',y',z'\rangle$

By similar triangles

$$\frac{x}{-z} = \frac{x'}{d} \quad \text{So} \quad x' = \frac{d\cdot x}{-z}$$

Likewise $y' = \dfrac{d\cdot y}{z}$

And $z' = -d$ (of course)

Let's express this as a matrix! (4x4)

$$x' = \frac{d \cdot x}{-z} \qquad y' = \frac{d \cdot y}{-z} \qquad z' = -d$$

$$\langle x, y, z \rangle \longmapsto \langle dx/(-z), dy/(-z), -d \rangle$$ (Not affine!)

In homogeneous coordinates

$$\langle x, y, z, 1 \rangle \mapsto \langle dx/(-z), dy/(-z), -d, 1 \rangle$$

or

$$\langle x, y, z, 1 \rangle \mapsto \langle dx, dy, dz, -z \rangle$$

$$\begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} dx \\ dy \\ dz \\ -z \end{pmatrix}$$

not 0001
on bottom row.

Problem : Lost the depth (distance) information,

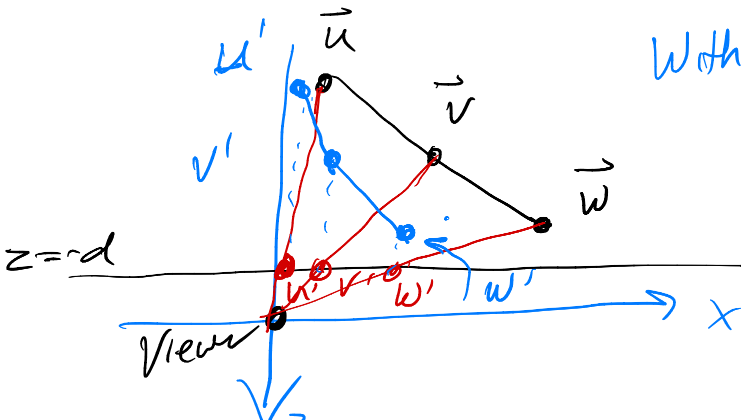
"Obvious idea that doesn't work well", is to use

$z' = z$. (Instead of $z' = -d$)

$$\langle x, y, z, 1 \rangle \mapsto \langle -\frac{dx}{z}, -\frac{dy}{z}, z, 1 \rangle$$

$$\text{"} \quad \mapsto \langle dx, dy, -z^2, -z \rangle$$

quadratic; nonlinear (non affine)



With $z' = z$

New u', v', w' - not on a straight line

- Mess up interpolation or averaging in the shaders.

Instead Let $z' = \text{pseudodist}(z) = A + B/z$

If $z_1 > z_2$ (z_1 is closer to the viewer)

$\text{pseudodist}(z_1) < \text{pseudodist}(z_2)$

provided $B > 0$.

(For mathematical convenience)

We want: $\text{pseudodist}(-n) = -1 = A + B/(-n)$

$\text{pseudodist}(-f) = 1 = A + B/(-f)$

Solve for A, B

$$A = \frac{f+n}{f-n} \qquad B = \frac{2fn}{f-n}$$

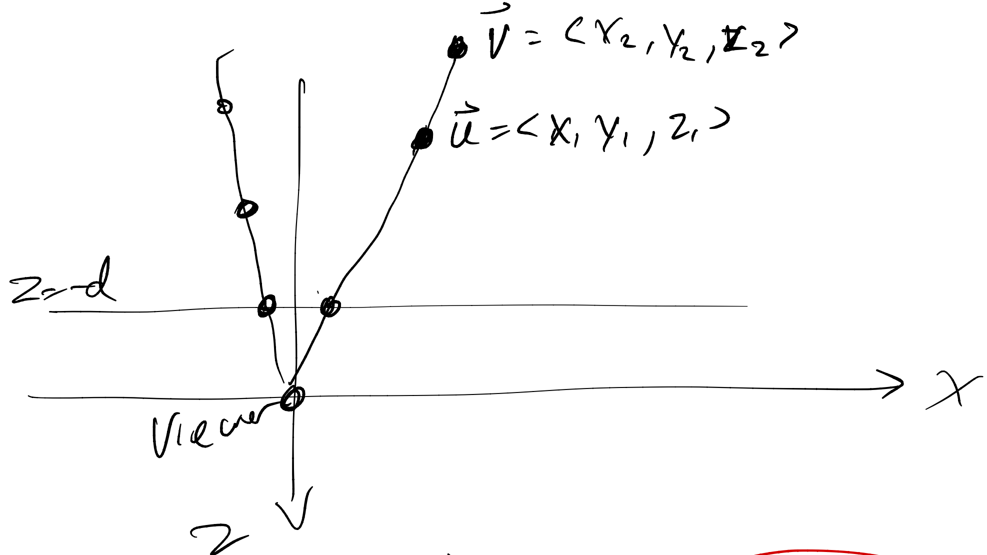
Let $\langle x, y, z, 1\rangle \longmapsto \langle -dx/z, -dy/z, A + B/z, 1\rangle$

<u>or</u> $\langle x, y, z, 1\rangle \longmapsto \langle dx, dy, -Az-B, -z\rangle$

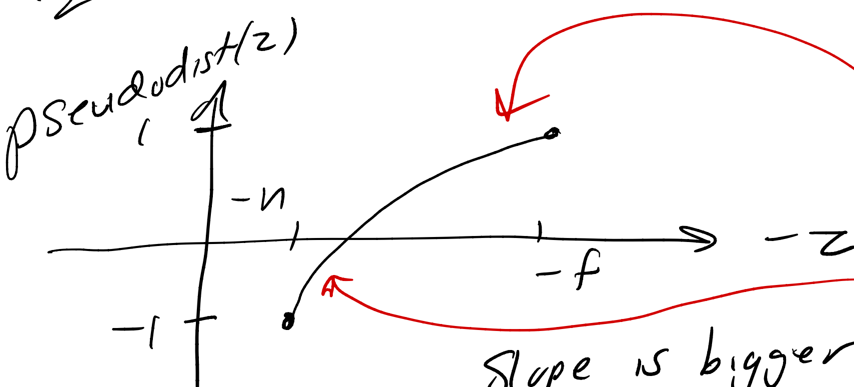
Use matrix

$$\begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & -A & -B \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} dx \\ dy \\ -Az-B \\ -z \end{pmatrix}$$

Set glFrustum
$$\begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$



$$\text{pseudodist}(z) = A + B/z$$



concave down

Slope is bigger here for near objects, than here for distant objects

So better distance resolution for near objects