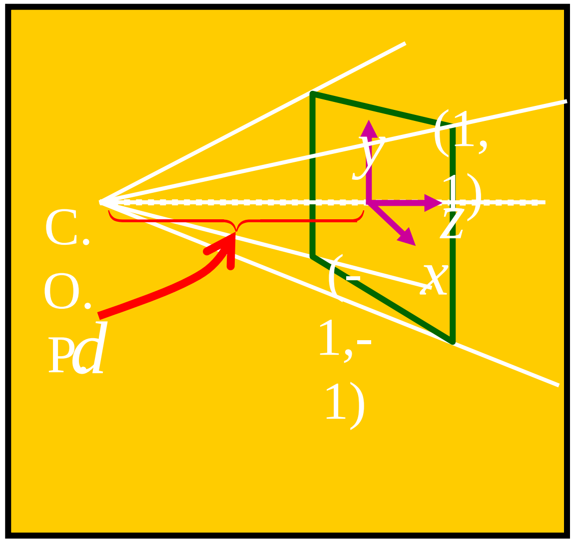
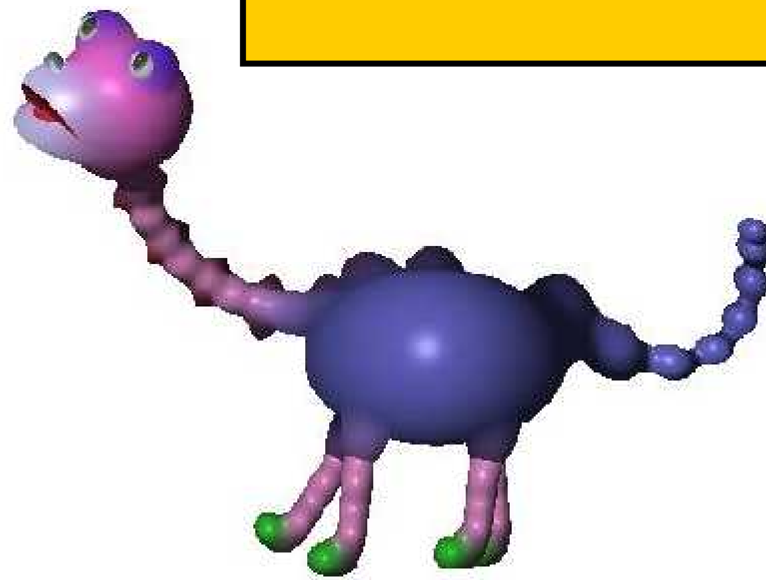
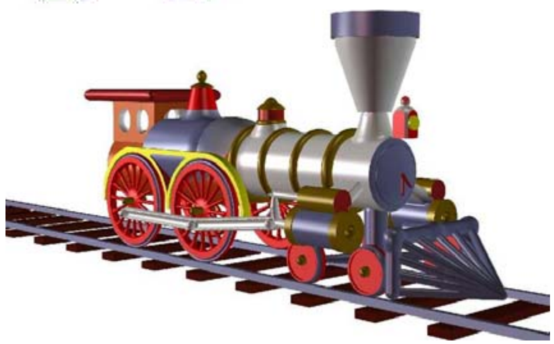


CSC 305
Perspective Viewing

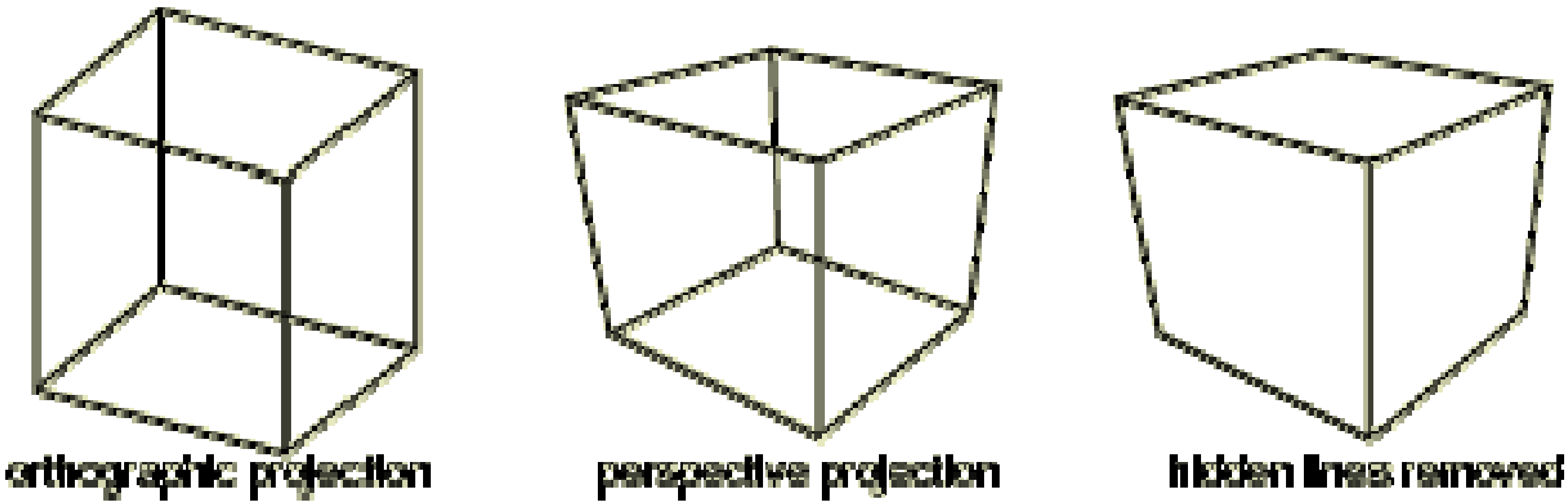
by Brian Wyvill



The University of Victoria Graphics Group

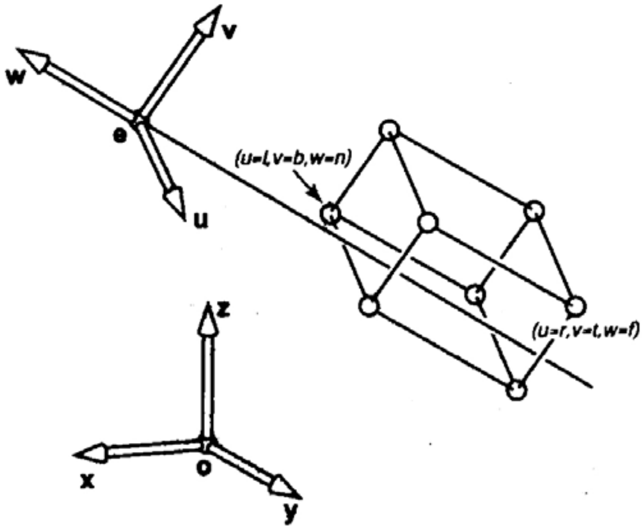


Perspective Viewing Transformation



- Tools for creating and manipulating a “camera” that produces pictures of a 3D scene
- Viewing transformations and projections
- Perform culling or back-face elimination
- The graphics pipeline

Viewing



View volume coordinates: origin o and xyz axes.

need to convert these to origin e and uvw axes.

We can use:

$$\mathbf{M_v} = \begin{bmatrix} x_u & y_u & z_u & 0 \\ x_v & y_v & z_v & 0 \\ x_w & y_w & z_w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_e \\ 0 & 1 & 0 & -y_e \\ 0 & 0 & 1 & -z_e \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Why is Mv computed this way?

$$
\mathbf{M}_{\mathbf{vrot}} = \begin{bmatrix} x_u & y_u & z_u & 0 \\ x_v & y_v & z_v & 0 \\ x_w & y_w & z_w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Orthogonal Matrix Properties –
upper 3x3 rotates row vectors into the major axes
applies to any rotation matrix or rotation translation (+normalization).

$$
\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta \\ -\sin \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}
$$

Special Orthogonal Matrices

1. For the upper 2x2 matrix, the row vectors are unit

2. $(\cos\theta \ -\sin\theta)$ and $(\sin\theta \ \cos\theta)$ are perpendicular to each other: $(\cos\theta \ -\sin\theta).(\sin\theta \quad \cos\theta) = \cos\theta\sin\theta - \sin\theta\cos\theta = 0$

3. take the determinant: $\cos\theta\cos\theta + \sin\theta\sin\theta = 1$

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Properties 1 and 2 are also true of the vectors formed by the columns: $(\cos\theta \ \sin\theta)$ and $(-\sin\theta \ \cos\theta)$
This defines a special orthogonal matrix

A transformation matrix which has such an upper 2x2 is called orthogonal i.e. transformations preserve angles and lengths. Matrices comprised of rotation and translation are orthogonal e.g. square when rotated stays a square

$$\begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Similarly in 3D the upper left 3x3 matrix comprise mutually perpendicular unit vectors and the submatrix has a determinant of unity.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \qquad \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \qquad \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

More Orthogonal Matrix Properties

Multiply one of the row vectors (transposed) of the upper left 3x3 rotation matrix by the rotation matrix rotates the vector onto one of the principal axes.

Z rotation

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta \\ -\sin\theta \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\theta+\sin\theta\sin\theta \\ \sin\theta\cos\theta-\cos\theta\sin\theta \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Also: the middle row rotates into the y axis
and the bottom row rotates into the z axis

similarly

X rotation

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ \cos\theta \\ -\sin\theta \end{bmatrix} = \begin{bmatrix} 0 \\ \cos\theta\cos\theta+\sin\theta\sin\theta \\ \sin\theta\cos\theta-\cos\theta\sin\theta \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Y rotation

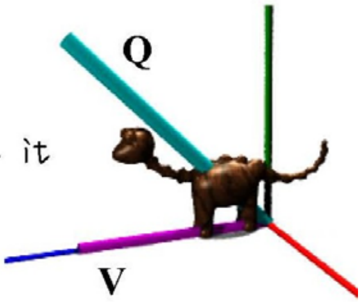
$$\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} -\sin\theta \\ 0 \\ \cos\theta \end{bmatrix} = \begin{bmatrix} -\sin\theta\cos\theta+\cos\theta\sin\theta \\ 0 \\ \cos\theta\cos\theta+\sin\theta\sin\theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Using Orthogonal Matrix Properties

The example required $R_z.R_x.R_y.T$ let R =

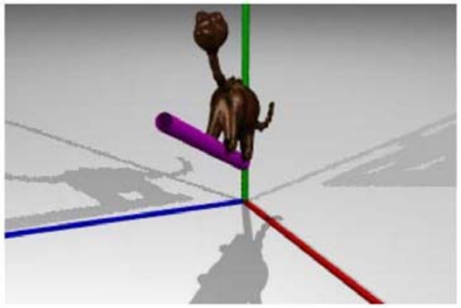
$$\begin{bmatrix} r_{1x} & r_{2x} & r_{3x} & 0 \\ r_{1y} & r_{2y} & r_{3y} & 0 \\ r_{1z} & r_{2z} & r_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The upper left 3x3 matrix handles the rotation
the bottom row rotates a vector into the z-axis.
Consider the magenta vector in the example,call it **V**



The matrix will rotate $(r_{1z}\ r_{2z}\ r_{3z})^T$ into the z-axis but it also rotated **V** into the z-axis so

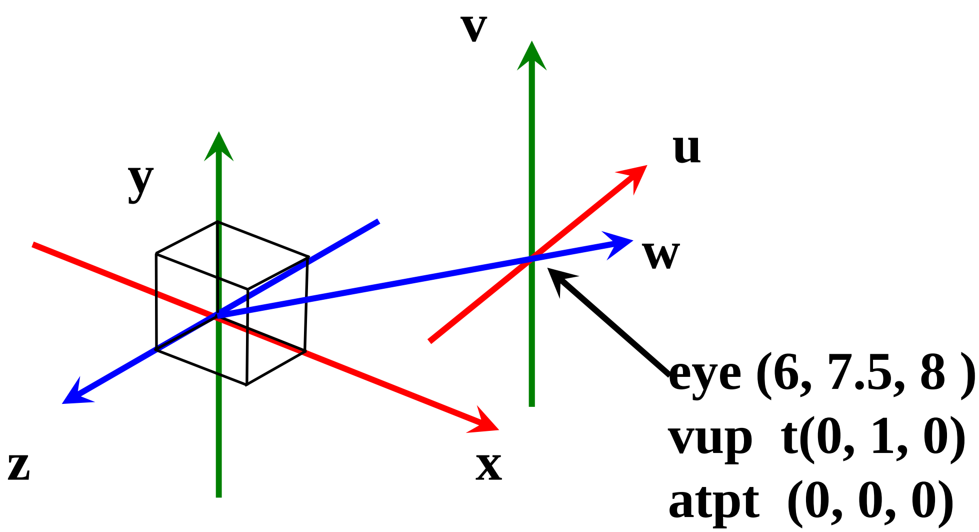
$\dfrac{\mathbf{V}}{||\mathbf{V}||}$ is also $(r_{1z}\ r_{2z}\ r_{3z})^T$



Similarly $(r_{1x}\ r_{2x}\ r_{3x})^T$ will be rotated into the x-axis. Given a vector **Q** in the plane of the dinosaur that is also rotated into the zy plane with **V** then

$(r_{1x}\ r_{2x}\ r_{3x})^T = \dfrac{\mathbf{V}\mathrm{x}\mathbf{Q}}{||\mathbf{V}\mathrm{x}\mathbf{Q}||}$ finally $(r_{1y}\ r_{2y}\ r_{3y})^T = \dfrac{\mathbf{V}\mathrm{x}\mathbf{V}\mathrm{x}\mathbf{Q}}{||\mathbf{V}\mathrm{x}\mathbf{V}\mathrm{x}\mathbf{Q}||}$

Example



gaze **g**= (0,0,0) – (6,7.5,8) sqrt(6*6+7.5*7.5+8*8) = 12.5

w = **-g**/||**g**|| = (6/12.5 , 7.5/12.5, 8/12.5)=(0.48, 0.6, 0.64)

u=(**t** x **w**)/||**t** x **W**||

v=(**w** x **u**)

Calculate U-vector

w= 1/12.5 (6, 7.5, 8) **t** =(0, 1, 0)

$$\mathbf{u} = (\mathbf{t} \times \mathbf{w}) / \|\mathbf{t} \times \mathbf{w}\| = (1/12.5)\det \begin{vmatrix} i & j & k \\ 0 & 1 & 0 \\ 6 & 7.5 & 8 \end{vmatrix}$$

$$\mathbf{t} \times \mathbf{w} = 1/12.5\ (8,\ -0.0,\ -6.0) = (0.6,\ 0.0,\ -0.48)$$

$$\frac{\mathbf{t} \times \mathbf{w}}{\|\mathbf{t} \times \mathbf{w}\|} = \frac{1/12.5\ (8,\ -0.0,\ -6.0)}{1/12.5 * \mathrm{sqrt}(64+36)} = (0.8,\ 0.0,\ -0.6)$$

Calculate V-vector

v = w x **u**

w = 1/12.5 (6, 7.5, 8)

u = (0.8, 0.0, -0.6)

$$\mathbf{(1/12.5 * 1/12.5)\ det} \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \mathbf{6} & \mathbf{7.5} & \mathbf{8} \\ \mathbf{10} & \mathbf{0} & \mathbf{-7.5} \end{vmatrix}$$

v = 1/(12.5 *12.5)(-56.25, 35, -75)

v = 1/12.5(-4.5, 10, -6)

v = (-0.36, 0.8, -0.48)

Calculate M_{vrot}

$$M_{vrot} = \begin{bmatrix} 0.8 & 0.0 & -0.6 & -6 \\ -0.36 & 0.8 & -0.48 & -75 \\ 0.48 & 0.6 & 0.64 & -8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

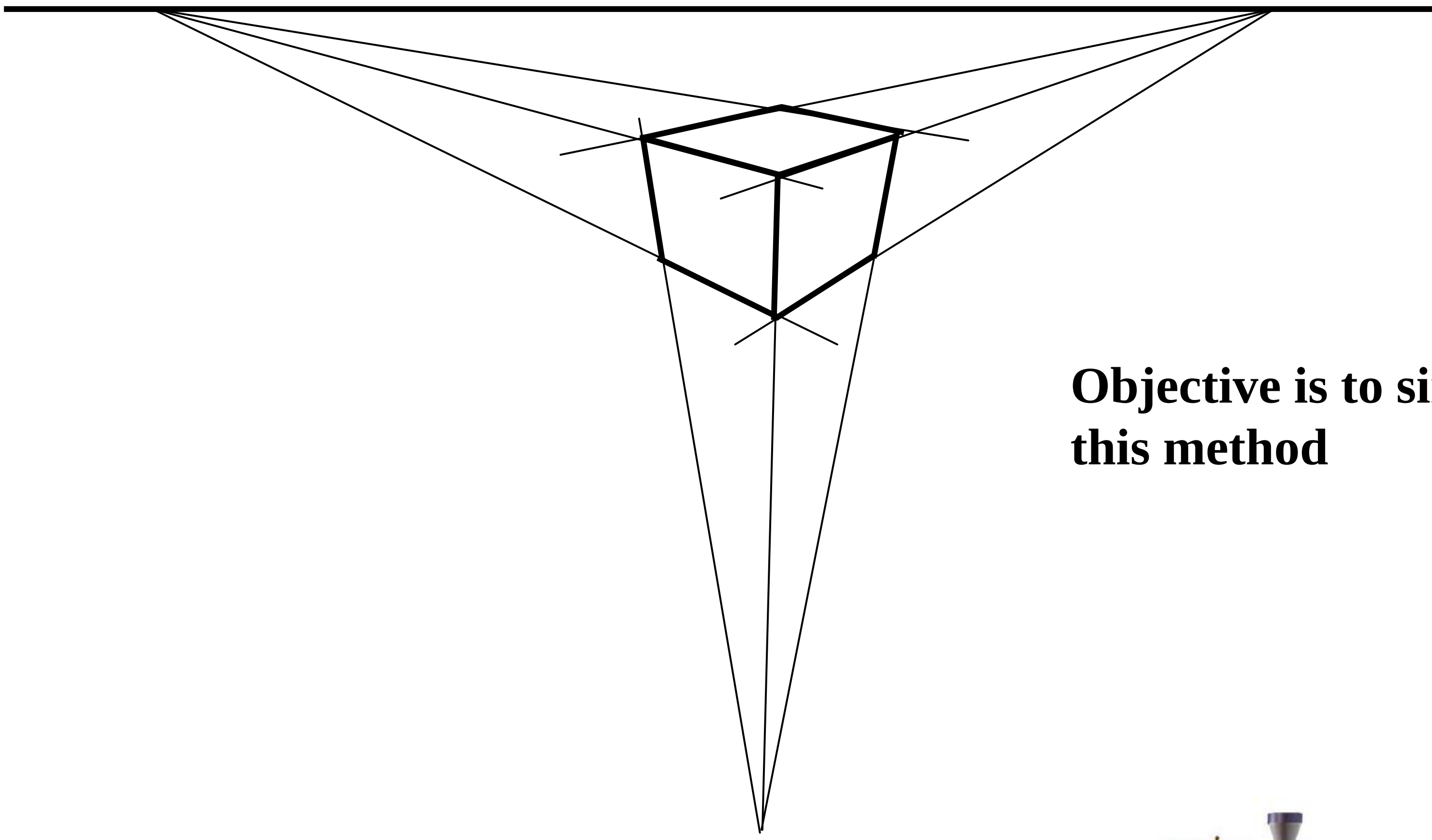
for eye at (6, 7.5, 8)

check the matrix:

$$\begin{bmatrix} 0.8 & 0.0 & -0.6 \\ -0.36 & 0.8 & -0.48 \\ 0.48 & 0.6 & 0.64 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.0 \\ -0.6 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

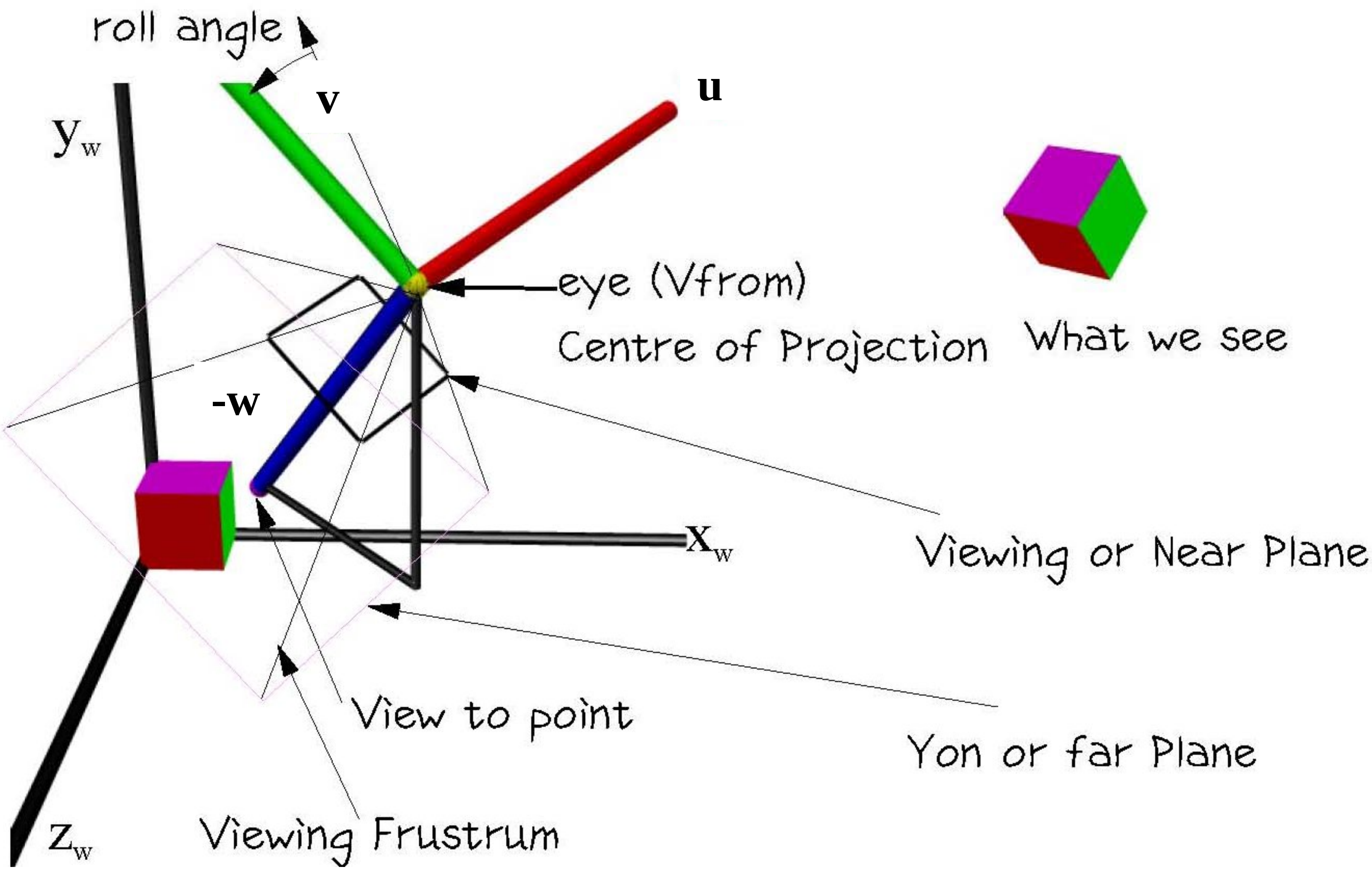
similarly
2nd row into y-axis
3rd row into z-axis

3 Point Perspective

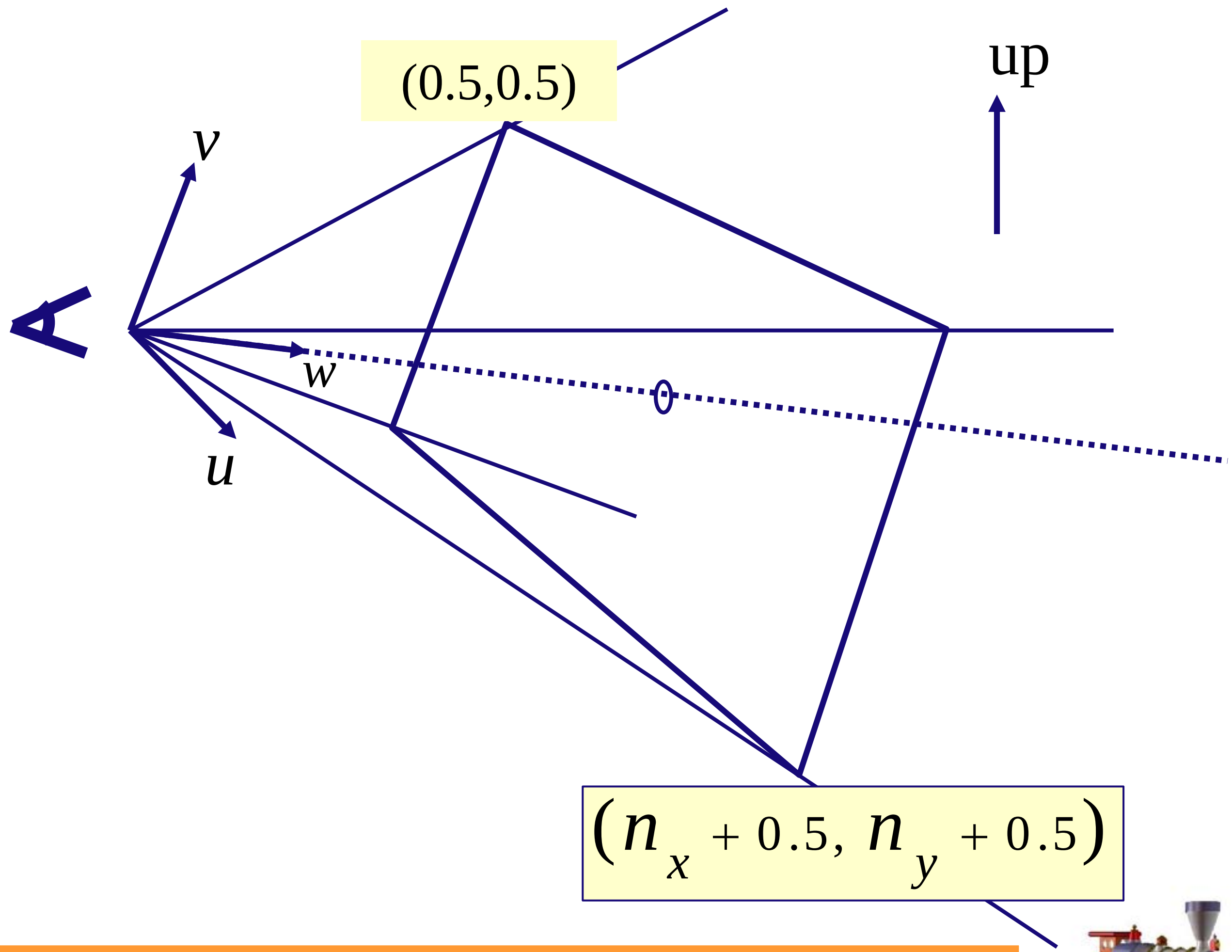


Objective is to simulate this method

The Eye System



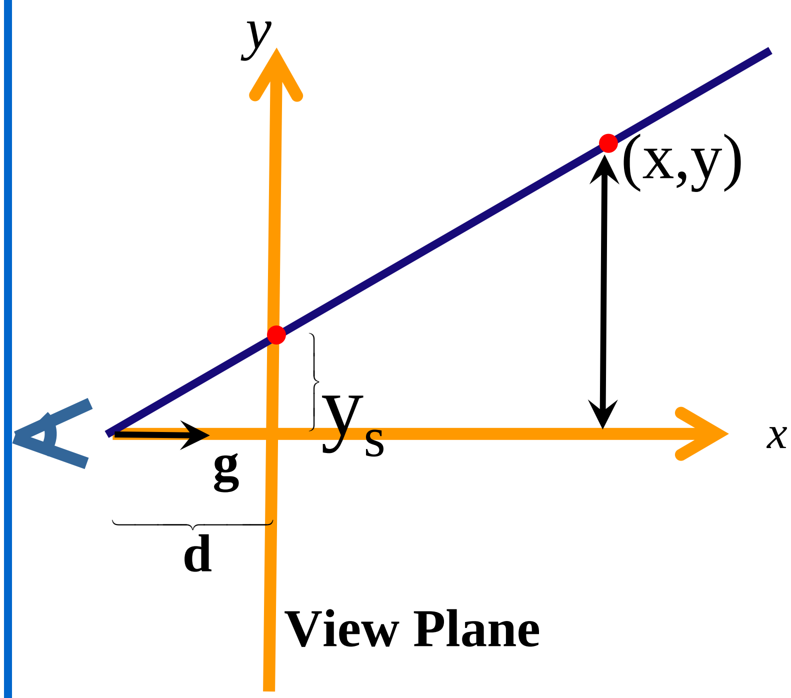
Geometry for the View Volume



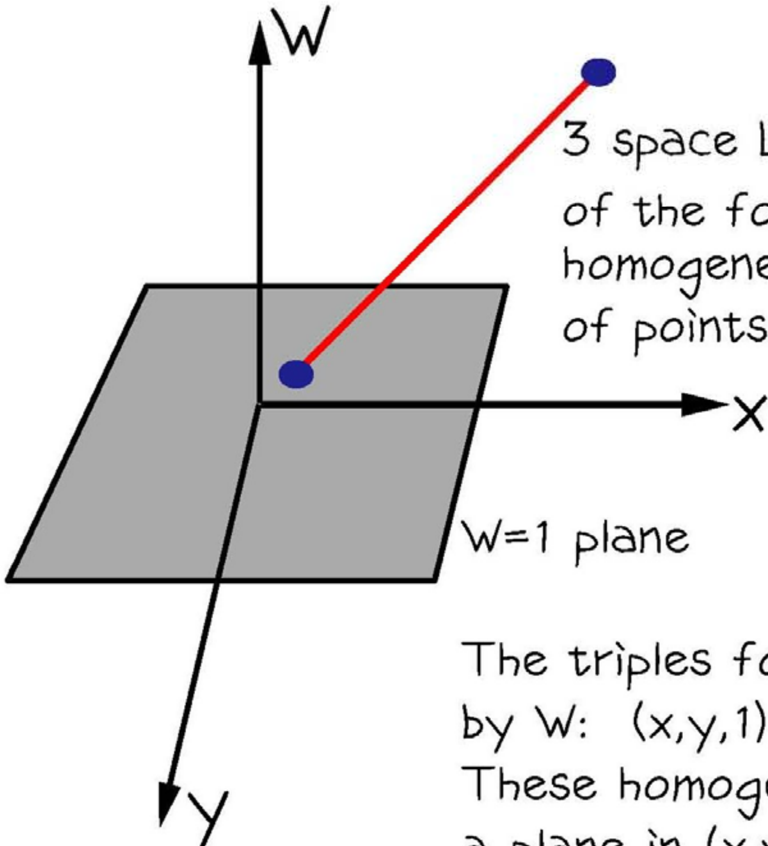
"True" Perspective in 2D

size of an object proportional to 1/z

$y_s = (d/z)\ y$



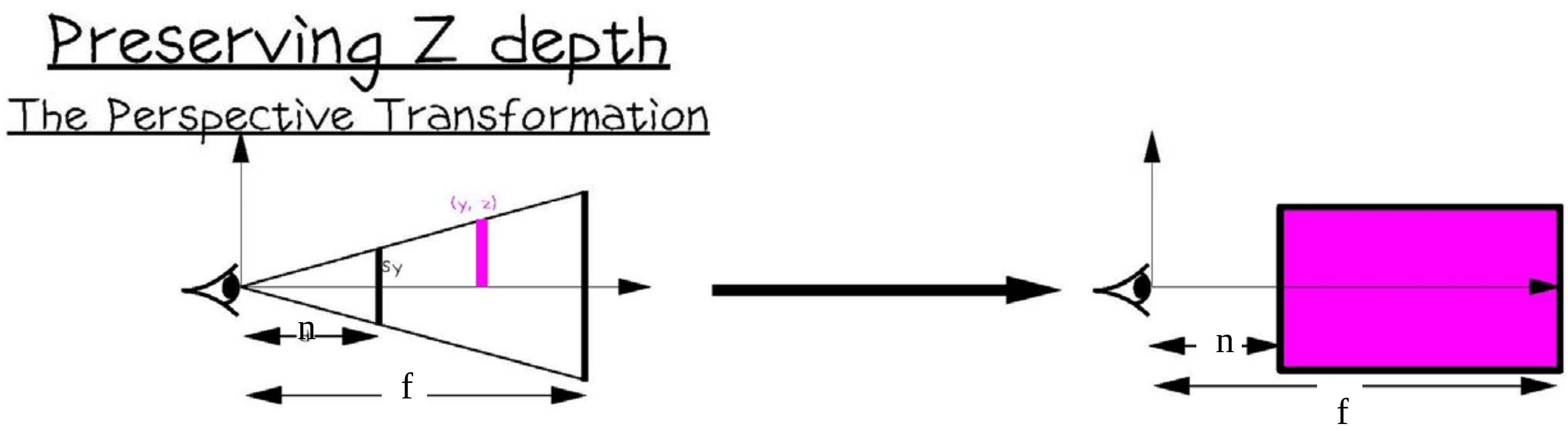
Homogeneous Representation of 2D transforms



3 space Line representing all the triples of the form (tx, ty, tW) t!=0 each 2D homogeneous point represents a line of points in 3 space.

The triples found by dividing by W: (x,y,1) represent points in 2 space. These homogenized points form a plane in (x,y,W) space where W=1

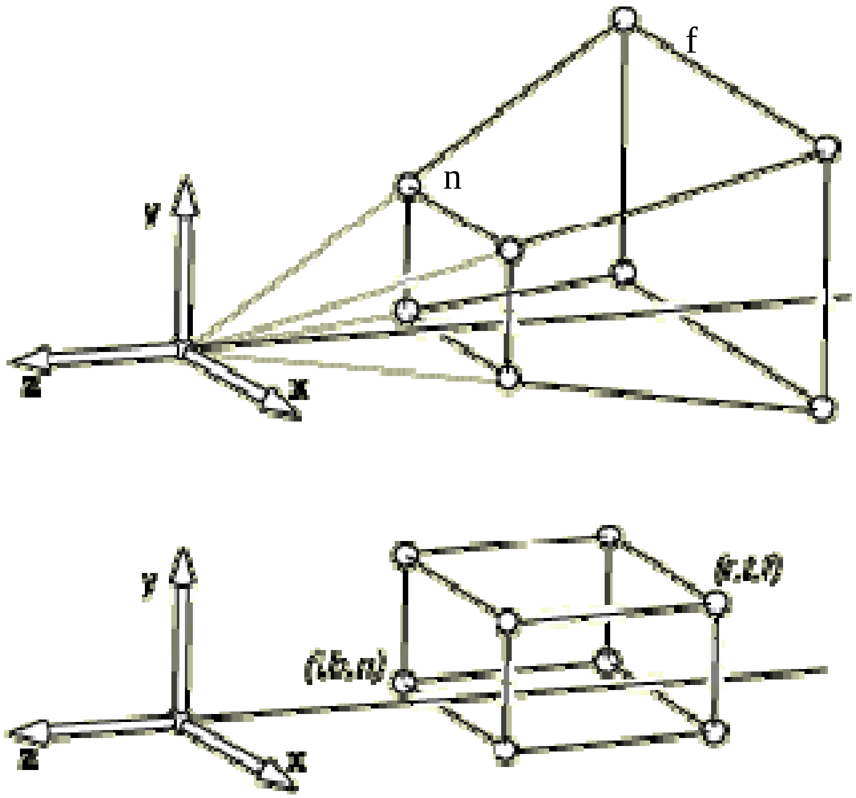
Perspective View Transformation



The view frustrum is an inconvenient shape.

Clipping
Z depth calculation
Hidden surface calculation

Perspective View Transformation



We want to find a matrix that

1. does not change points on z=n plane

2. maps large rectangle at z=f to small rectangle at z=n

3. achieve division by z by using homogeneous to cartesian conversion i.e. *homogenize*

Perspective Matrix

$$\mathbf{M_p} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{n+f}{n} & -f \\ 0 & 0 & \frac{1}{n} & 0 \end{bmatrix}$$

for homogenous point (x,y,z,1) = (x/h, y/h, z/h, h)

$$\mathbf{M_p} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z\frac{n+f}{n} - f \end{bmatrix} \xrightarrow{\textit{homogenize}} \begin{bmatrix} nx/z \\ ny/z \\ n + f - \frac{fn}{z} \\ 1 \end{bmatrix}$$

Perspective Matrix

for homogenous point (x,y,z,1) = (x/h, y/h, z/h, h)
thus we can multiply any transformation matrix by a constant.

$\mathbf{M}(h\mathbf{p}) = (h\mathbf{M})\mathbf{p} = \mathbf{M}\mathbf{p}$ so we can multiply perspective matrix by n:

$$\mathbf{M_p} = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Inverse Perspective Matrix

$$\mathbf{M_p} = \begin{bmatrix} 1/n & 0 & 0 & 0 \\ 0 & 1/n & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1/fn & \frac{n+f}{fn} \end{bmatrix}$$

e.g. for picking

Tidy this up by multiplying by nf

$$\mathbf{M^{-1}_p} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & fn \\ 0 & 0 & -1 & n+f \end{bmatrix}$$

Perspective Projection Algorithm

We can now use the mechanism we used before for orthographic projection

Again Arbitrary View Point

compute M_v
compute M_0
compute M_p

$M = M_0 M_v M_p$

for each line segment in 3D (a_i, b_i) do {
    $p = Ma_i$
    $q = Mb_i$
    drawline($\frac{x_p}{h}, \frac{y_p}{h}, \frac{x_q}{h}, \frac{y_q}{h}$)
}