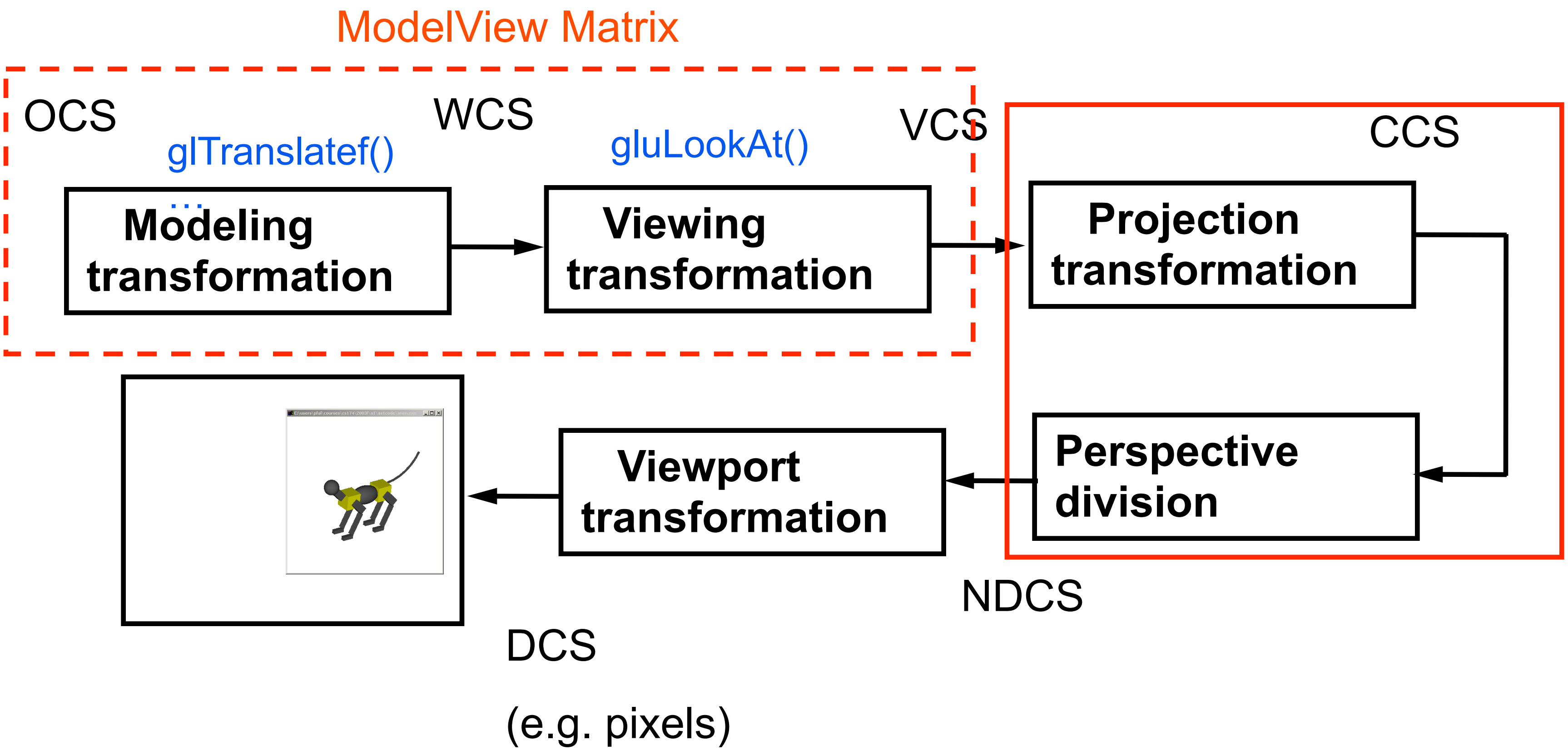
Transformations in the pipeline



Background (reminder)

Line (in 2D)

- Explicit

$$y = \frac{dy}{dx}(x - x_0) + y_0$$

- Implicit

$$F(x,y) = (x - x_0)dy - (y - y_0)dx$$

$$\begin{array}{llll} \text{if} & F(x,y) = 0 & \text{then} & (x,y) \text{ is on line} \\ & F(x,y) > 0 & & (x,y) \text{ is below line} \\ & F(x,y) < 0 & & (x,y) \text{ is above line} \end{array}$$

- Parametric

$$x(t) = x_0 + t(x_1 - x_0)$$
$$y(t) = y_0 + t(y_1 - y_0)$$
$$t \in [0,1]$$

$$P(t) = P_0 + t(P_1 - P_0), \text{ or}$$
$$P(t) = (1 - t)P_0 + tP_1$$

Background (reminder)

Plane equations

Implicit

$F(x, y, z) = Ax + By + Cz + D = \mathbf{N} \bullet P + D$

Points on Plane $F(x, y, z) = 0$

Parametric

$Plane(s,t) = P_0 + s(P_1 - P_0) + t(P_2 - P_0)$

P_0, P_1, P_2 not colinear

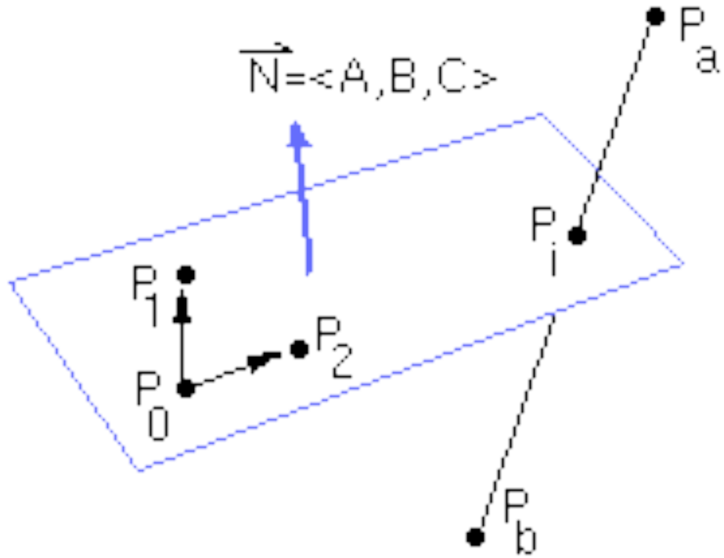
or

$Plane(s,t) = (1 - s - t)P_0 + sP_1 + tP_2$

$Plane(s,t) = P_0 + sV_1 + tV_2$ where V_1, V_2 basis vectors

Explicit

$z = -(A/C)x - (B/C)y - D/C, \; C \neq 0$

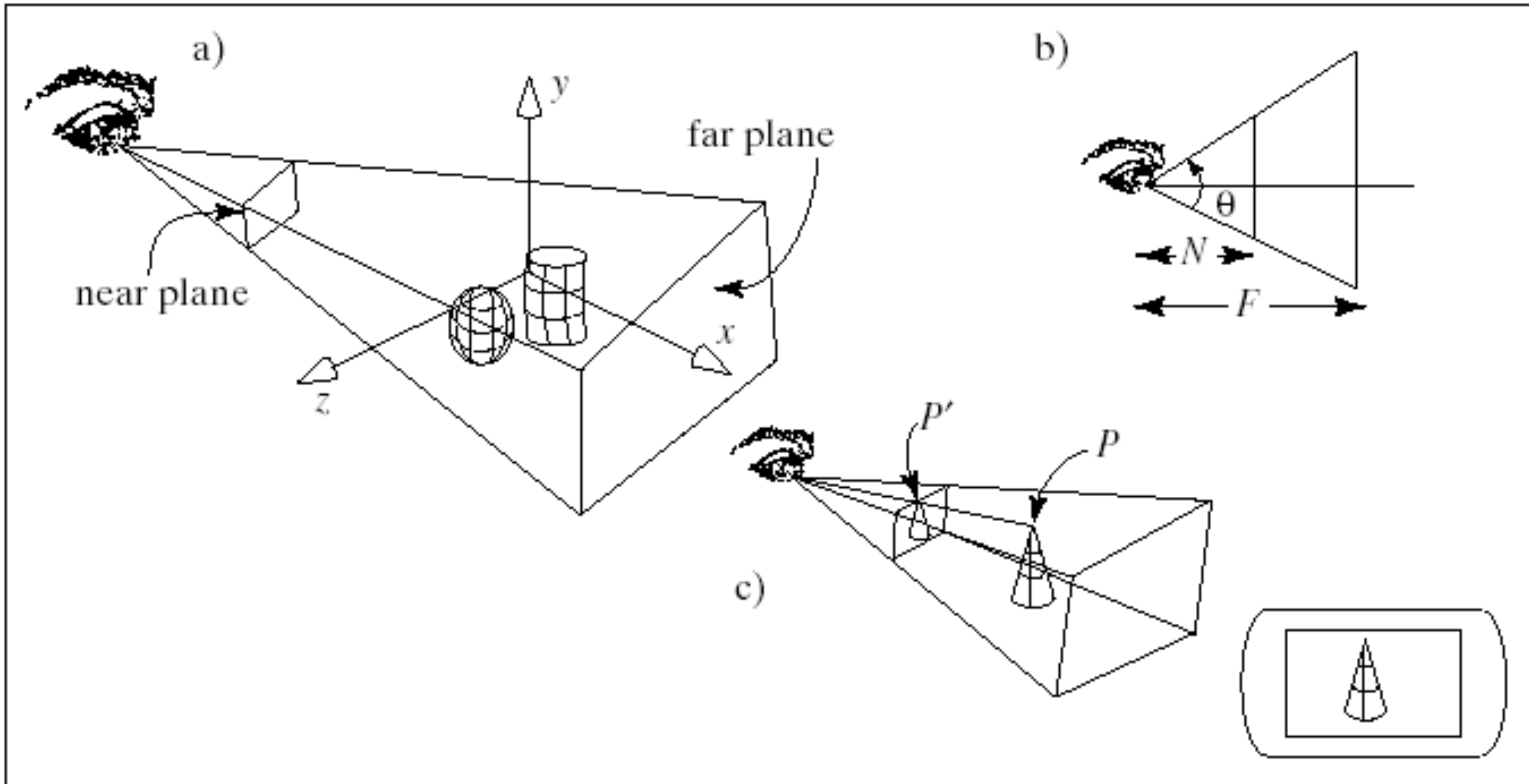


Reminder: Homogeneous Coordinates

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \underset{\times w}{\rightarrow} \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix}$$

$$\begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} \underset{/w}{\rightarrow} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Projection transformations



Introduction to Projection Transformations

[Hill: 371-378, 398-404. Foley & van Dam: p. 229-242]

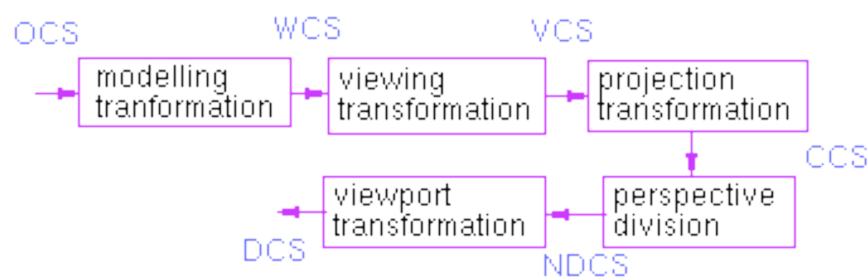
Mapping: f : $R^n \rightarrow R^m$

Projection: n > m

Planar Projection: Projection on a plane.

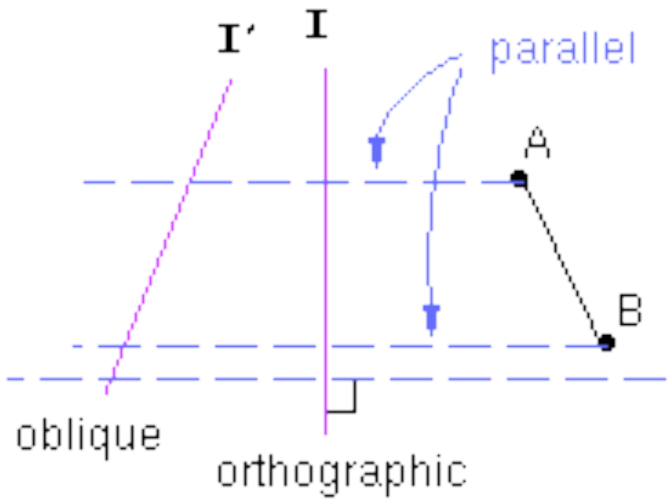
$R^3 \rightarrow R^2$ or

$R^4 \rightarrow R^3$ homogenous coordinates.

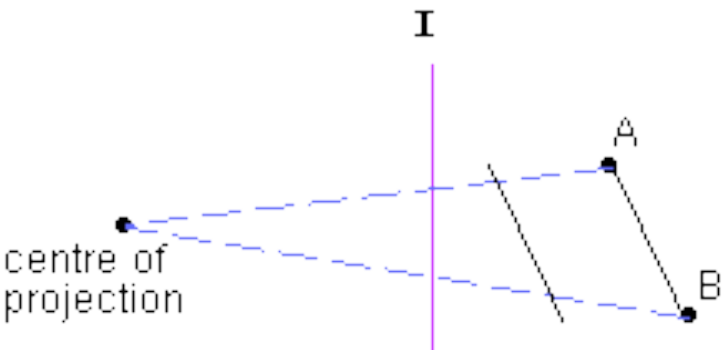


Basic projections

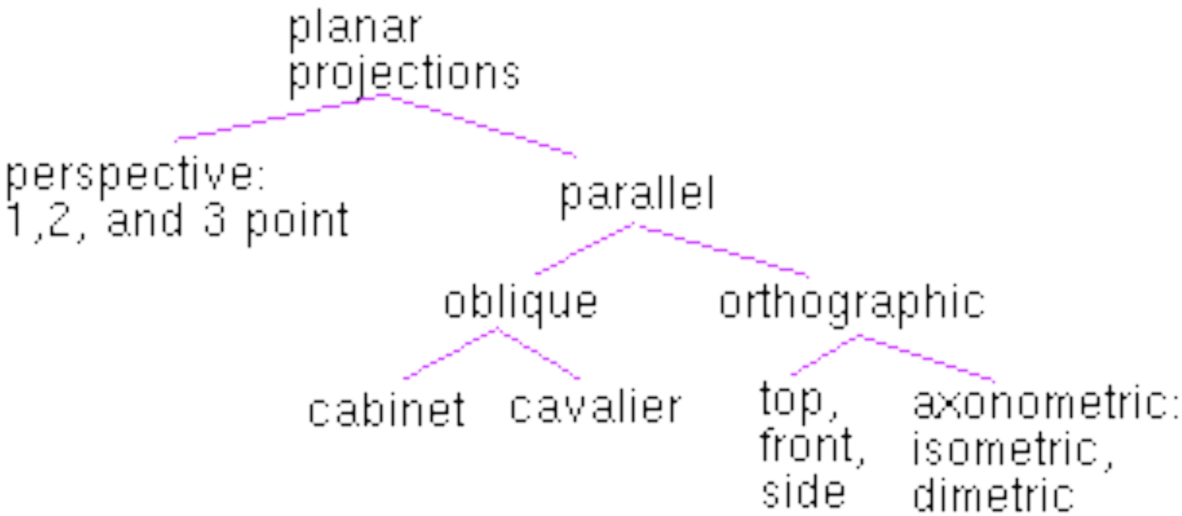
Parallel



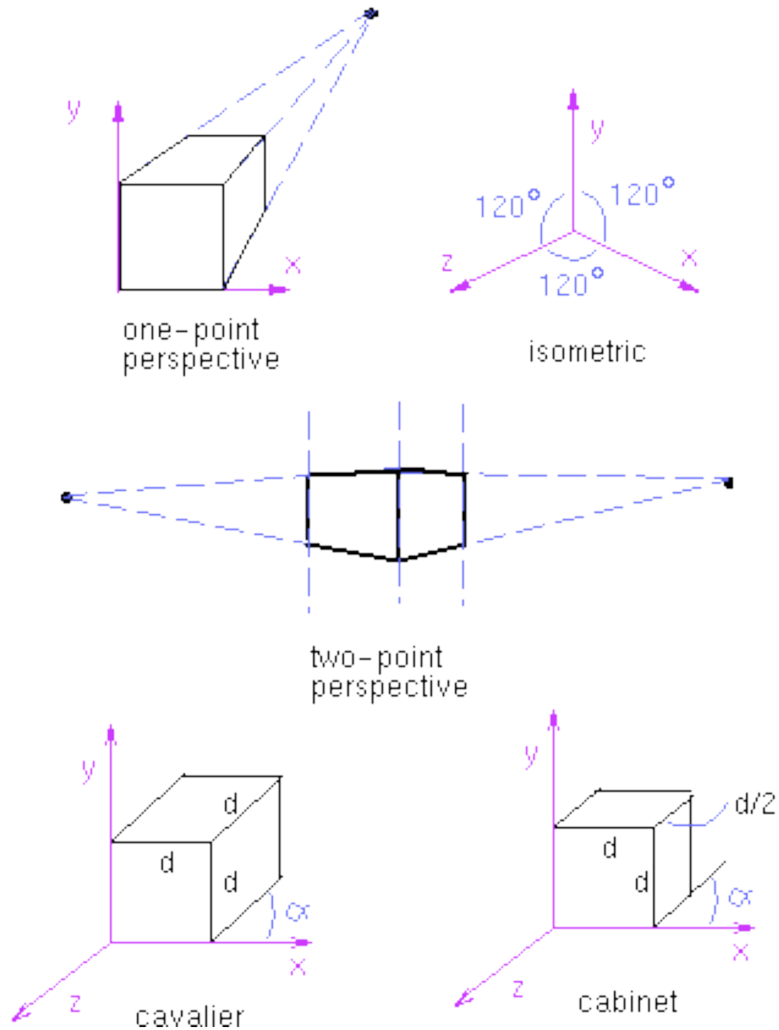
Perspective



Taxonomy


planar projections
perspective: 1,2, and 3 point
parallel
oblique
orthographic
cabinet
cavalier
top, front, side
axonometric: isometric, dimetric

Examples


y
x
one-point
perspective
y
120°
120°
z
x
120°
isometric
two-point
perspective
y
d
d
d
α
x
z
cavalier
y
d/2
d
d
α
x
z
cabinet

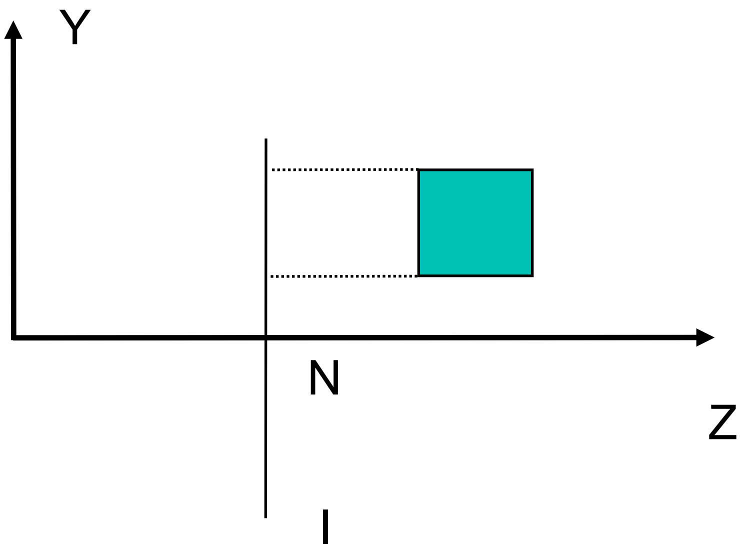
A basic orthographic projection

x' = x

y' = y

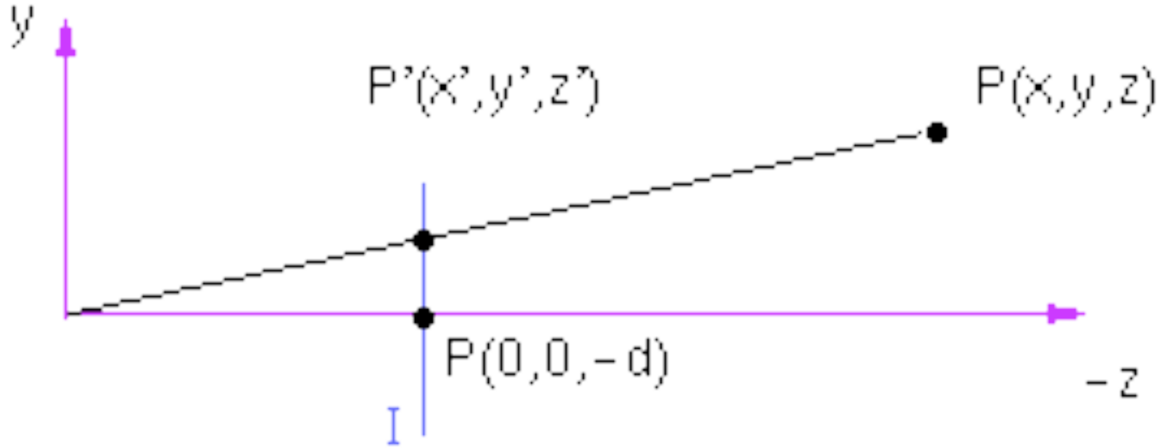
z' = N

Matrix Form



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & N \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ N \\ 1 \end{bmatrix}$$

A basic perspective projection



Similar triangles

x'/d = x/(-z) -> x' = x d/(-z)

y'/d = y/(-z) => y' = y d/(-z)

z' = -d

In matrix form

Matrix form of

x' = x d/(-z)

y' = y d/(-z)

z' = -d

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ -z/d \end{bmatrix}$$

$$\begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} xd \\ yd \\ zd \\ -z \end{bmatrix}$$

Moving from 4D to 3D

$$\begin{bmatrix} x \\ y \\ z \\ -z/d \end{bmatrix} \xrightarrow{h=-z/d} \begin{bmatrix} x/h \\ y/h \\ z/h \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} xd/(-z) \\ yd/(-z) \\ -d \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Projections in OpenGL

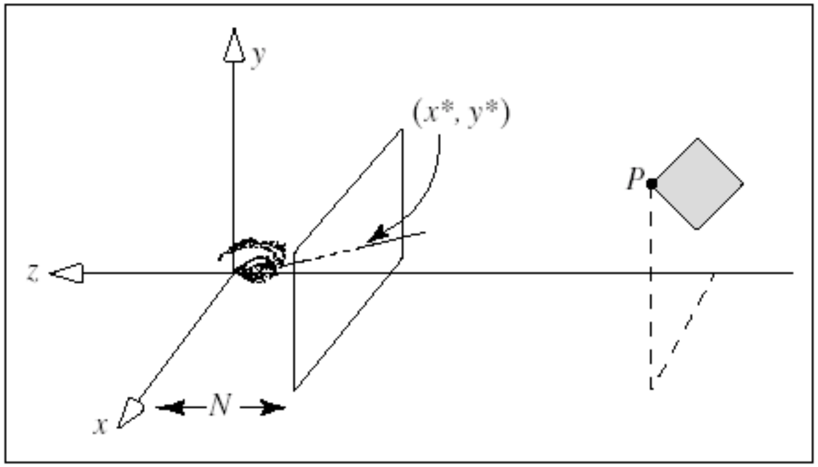
Camera coordinate system

Image plane = near plane

Camera at (0,0,0)

Looking at –z

Image plane at z = -N

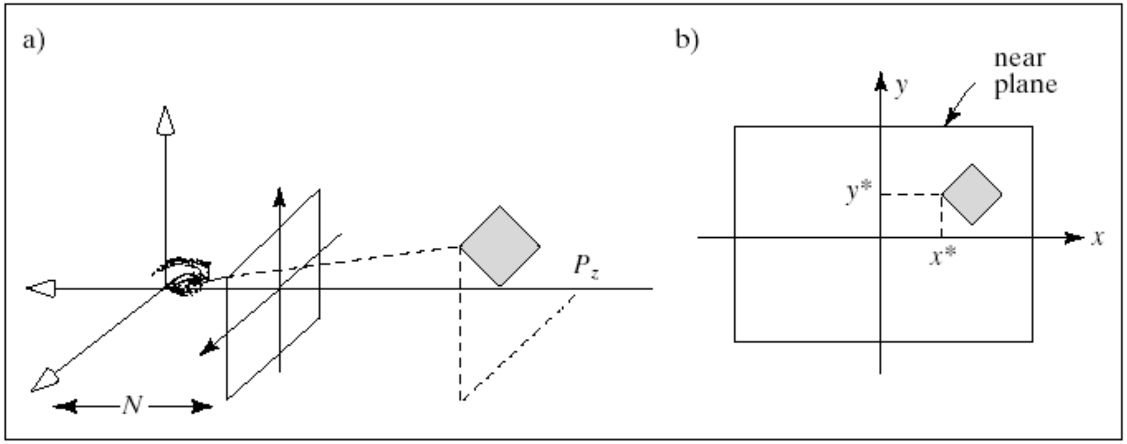


Perspective projection of a point

In eye coordinates P =[Px,Py,Pz,1]^T

x/Px = N/(-Pz) => x = NPx/(-Pz)

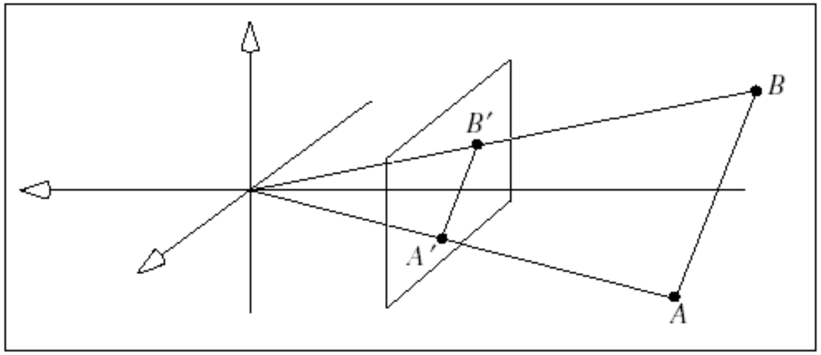
y/Px = N/(-Pz) => y = NPy/(-Pz)



Observations

- Perspective foreshortening
- Denominator becomes undefined for z = 0
- If P is behind the eye Pz changes sign
- Near plane just scales the picture
- Straight line -> straight line

$$x' = -N\frac{P_x}{P_z}$$
$$y' = -N\frac{P_y}{P_z}$$
$$z' = -N$$



Perspective projection of a line

$$L(t) = \mathbf{A} + \vec{\mathbf{c}}t = \begin{bmatrix} A_x \\ A_y \\ A_z \\ 1 \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \\ c_z \\ 0 \end{bmatrix} t$$

$$\widetilde{L}(t) = \mathbf{M}L(t) = \mathbf{M}(\mathbf{A} + \vec{\mathbf{c}}t) = \mathbf{M} \begin{bmatrix} A_x + c_x t \\ A_y + c_y t \\ A_z + c_z t \\ 1 \end{bmatrix} = \begin{bmatrix} N(A_x + c_x t) \\ N(A_y + c_y t) \\ N(A_z + c_z t) \\ -(A_z + c_z t) \end{bmatrix}$$

Perspective Division, drop fourth coordinate →

$$L'(t) = \begin{bmatrix} -N(A_x + c_x t)/(A_z + c_z t) \\ -N(A_y + c_y t)/(A_z + c_z t) \\ -N \end{bmatrix}$$

Is it a line?

$$\text{Original}: L(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} A_x + c_x t \\ A_y + c_y t \\ A_z + c_z t \end{bmatrix}$$

$$\text{Projected}: L'(t) = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -Nx/z \\ -Ny/z \\ -N \end{bmatrix} = \begin{bmatrix} -N(A_x + c_x t)/(A_z + c_z t) \\ -N(A_y + c_y t)/(A_z + c_z t) \\ -N \end{bmatrix}$$

$$x' = -N(A_x + c_x t)/(A_z + c_z t) \Rightarrow x'(A_z + c_z t) = -N(A_x + c_x t) \Rightarrow$$

$$x' A_z + x' c_z t = -NA_x - Nc_x t \Rightarrow \begin{cases} x' A_z + NA_x = -(x' c_z + Nc_x)t \\ \text{and similarly for y} \\ y' A_z + NA_y = -(y' c_z + Nc_y)t \end{cases}$$

Cont'd next slide

Is it a line? (cont'd)

$$\left.\begin{aligned} x' A_z + NA_x &= -(x'c_z + Nc_x)t \\ y' A_z + NA_y &= -(y'c_z + Nc_y)t \end{aligned}\right| \Rightarrow \left.\begin{aligned} x' A_z + NA_x &= -(x'c_z + Nc_x)t \\ -(y'c_z + Nc_y)t &= y' A_z + NA_y \end{aligned}\right| \Rightarrow$$

$$(x' A_z + NA_x)(y'c_z + Nc_y) = (x'c_z + Nc_x)(y' A_z + NA_y) \Rightarrow$$

$$x' A_z y'c_z + x' A_z Nc_y + NA_x y'c_z + N^2 A_x c_y = x'c_z y' A_z + x'c_z NA_y + Nc_x y' A_z + N^2 A_y c_x \Rightarrow$$

$$(A_z Nc_y - c_z NA_y)x' + (NA_x c_z + Nc_x A_z)y' + N^2(A_x c_y + A_y c_x) = 0 \Rightarrow$$

$\Rightarrow$ $ax' + by' + c = 0$ which is the equation of a line.

So is there a difference?

$$\text{Original}: L(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} A_x + c_x t \\ A_y + c_y t \\ A_z + c_z t \end{bmatrix}$$

$$\text{Projected}: L'(t) = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -Nx/z \\ -Ny/z \\ -N \end{bmatrix} = \begin{bmatrix} -N(A_x + c_x t)/(A_z + c_z t) \\ -N(A_y + c_y t)/(A_z + c_z t) \\ -N \end{bmatrix}$$

So is there a difference?

The speed of the lines if cz is not 0

$$\text{Original}: L(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} A_x + c_x t \\ A_y + c_y t \\ A_z + c_z t \end{bmatrix} \Rightarrow \boxed{\frac{\partial L(t)}{\partial t} = \vec{\mathrm{c}}}$$

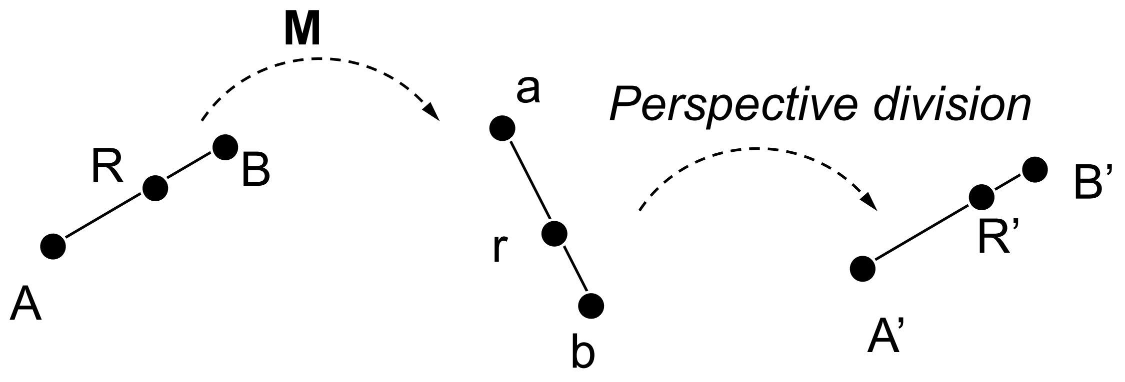
$$\text{Projected}: L'(t) = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -Nx/z \\ -Ny/z \\ -N \end{bmatrix} = \begin{bmatrix} -N(A_x + c_x t)/(A_z + c_z t) \\ -N(A_y + c_y t)/(A_z + c_z t) \\ -N \end{bmatrix} \Rightarrow$$

$$\frac{\partial x'}{\partial t} = -N \frac{\partial}{\partial t}((A_x + c_x t)/(A_z + c_z t)) = -N \frac{c_x(A_z + c_z t) - (A_x + c_x t)c_z}{(A_z + c_z t)^2} = -N \frac{c_x A_z - A_x c_z}{(A_z + c_z t)^2} \Rightarrow$$

$$\boxed{\frac{\partial L'(t)}{\partial t} = \frac{-N}{(A_z + c_z t)^2} \begin{bmatrix} c_x A_z - A_x c_z \\ c_y A_z - A_y c_z \end{bmatrix}}$$

Inbetween points

How do points on lines transform?



Viewing system: $R(g) = (1-g)A + gB$

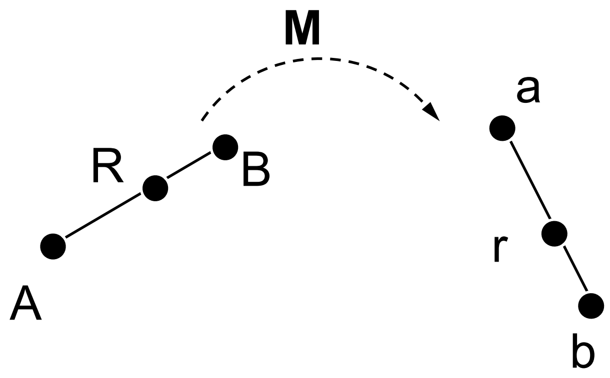
Projected (4D) : $r = MR$

Projected cartesian: $R'(f) = (1-f)A' + fB'$

What is the relationship between g and f?

First step

Viewing to homogeneous space (4D)



$$R = (1 - g)A + gB$$

$$r = MR = M[(1 - g)A + gB] = (1 - g)MA + gMB \Rightarrow$$

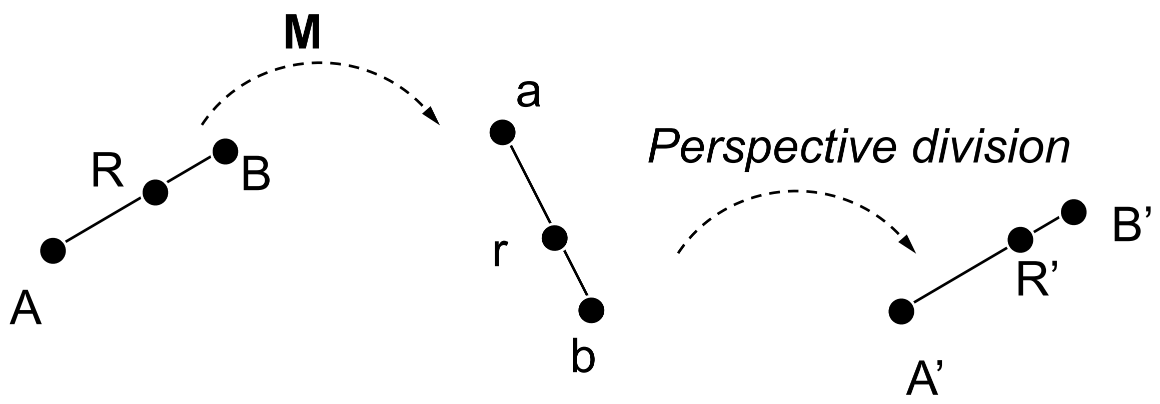
$$r = (1 - g)a + gb$$

$$a = MA = (a_1, a_2, a_3, a_4)$$

$$b = MB = (b_1, b_2, b_3, b_4)$$

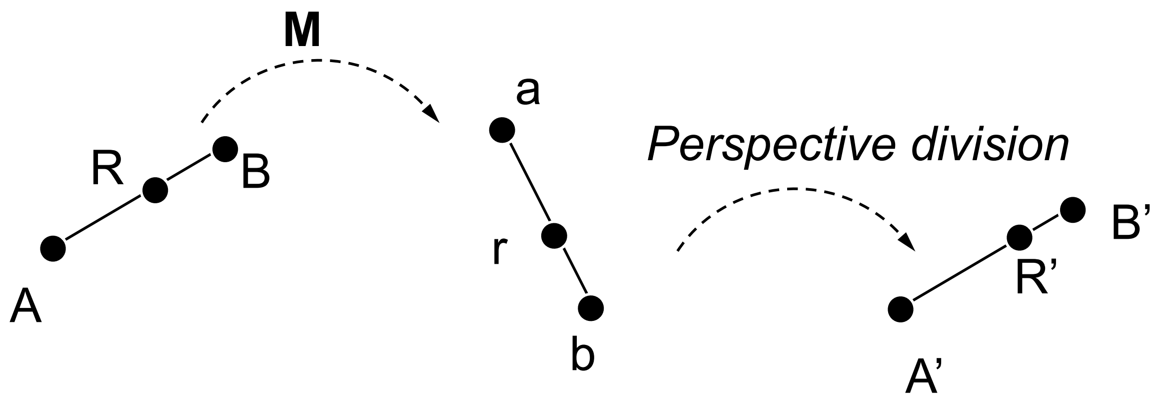
Second step

Perspective division



$$\left\{ \begin{array}{l} r = (1-g)a + gb \\ a = (a_1, a_2, a_3, a_4) \\ b = (b_1, b_2, b_3, b_4) \end{array} \right\} \Rightarrow \boxed{R'_1 = \frac{r_1}{r_4} = \frac{(1-g)a_1 + gb_1}{(1-g)a_4 + gb_4}}$$

Putting all together



$$R'_1 = \frac{(1-g)a_1 + gb_1}{(1-g)a_4 + gb_4} = \frac{lerp(a_1, b_1, g)}{lerp(a_4, b_4, g)}$$

$$At \;\; the \;\; same \;\; time:$$

$$R' = (1-f)A' + fB' \Rightarrow R'_1 = (1-f)A'_1 + fB'_1$$

$$R'_1 = (1-f)\frac{a_1}{a_4} + f\frac{b_1}{b_4} = lerp(\frac{a_1}{a_4}, \frac{b_1}{b_4}, f)$$

Relation between the fractions

$$\left.\begin{aligned} R'_1(g) &= \frac{lerp(a_1, b_1, g)}{lerp(a_4, b_4, g)} \\ R'_1(f) &= lerp\left(\frac{a_1}{a_4}, \frac{b_1}{b_4}, f\right) \end{aligned}\right\} \Rightarrow g = \frac{f}{lerp(\frac{b_4}{a_4}, 1, f)}$$

substituting this in $R(g) = (1-g)A + gB$ yields

$$R_1 = \frac{lerp(\frac{A_1}{a_4}, \frac{B_1}{b_4}, f)}{lerp(\frac{1}{a_4}, \frac{1}{b_4}, f)}$$

THAT MEANS: For a given f in **screen space** and A,B in **viewing space** we can find the corresponding R (or g) in **viewing space** using the above formula.

“A” can be texture coordinates, position, color, normal etc.

Effect of perspective projection on lines [Hill 375]

Equations

$$\text{Original}: L(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} A_x + c_x t \\ A_y + c_y t \\ A_z + c_z t \end{bmatrix}$$

$$\text{Projected}: L'(t) = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -Nx/z \\ -Ny/z \\ -N \end{bmatrix} = \begin{bmatrix} -N(A_x + c_x t)/(A_z + c_z t) \\ -N(A_y + c_y t)/(A_z + c_z t) \\ -N \end{bmatrix}$$

What happens to parallel lines?

Effect of perspective projection on lines [Hill 375]

Parallel lines

$$\text{Original}: L(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} A_x + c_x t \\ A_y + c_y t \\ A_z + c_z t \end{bmatrix}$$

$$\text{Projected}: L'(t) = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -Nx/z \\ -Ny/z \\ -N \end{bmatrix} = \begin{bmatrix} -N(A_x + c_x t)/(A_z + c_z t) \\ -N(A_y + c_y t)/(A_z + c_z t) \\ -N \end{bmatrix}$$

If parallel to view plane then:

$$c_z = 0 \rightarrow L'(t) = -\frac{N}{A_z}(A_x + c_x t, A_y + c_y t)$$

$$\text{slope} \quad = \frac{c_y}{c_x}$$

Effect of perspective projection on lines [Hill 375]

Parallel lines

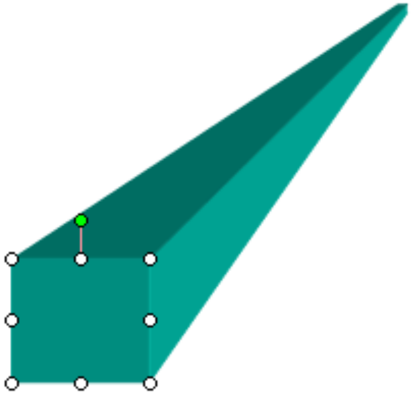
$$\text{Original}: L(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} A_x + c_x t \\ A_y + c_y t \\ A_z + c_z t \end{bmatrix}$$

$$\text{Projected}: L'(t) = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -Nx/z \\ -Ny/z \\ -N \end{bmatrix} = \begin{bmatrix} -N(A_x + c_x t)/(A_z + c_z t) \\ -N(A_y + c_y t)/(A_z + c_z t) \\ -N \end{bmatrix}$$

If not parallel to view plane then:

$$c_z \neq 0 \rightarrow \lim_{t\rightarrow\infty} L'(t) = -\frac{N}{c_z}(c_x, c_y)$$

Vanishing point!



Summary

Forshortening

Non-linear

Lines go to lines

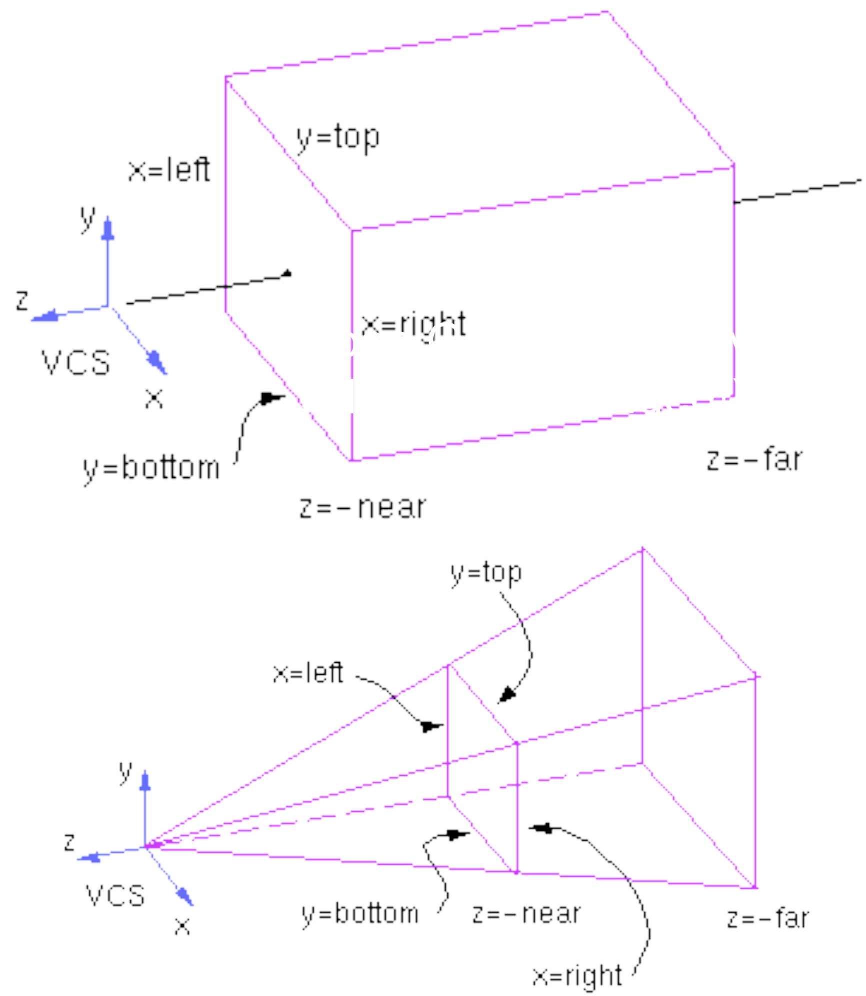
Parallel lines either intersect or remain parallel

Inbetweeneness

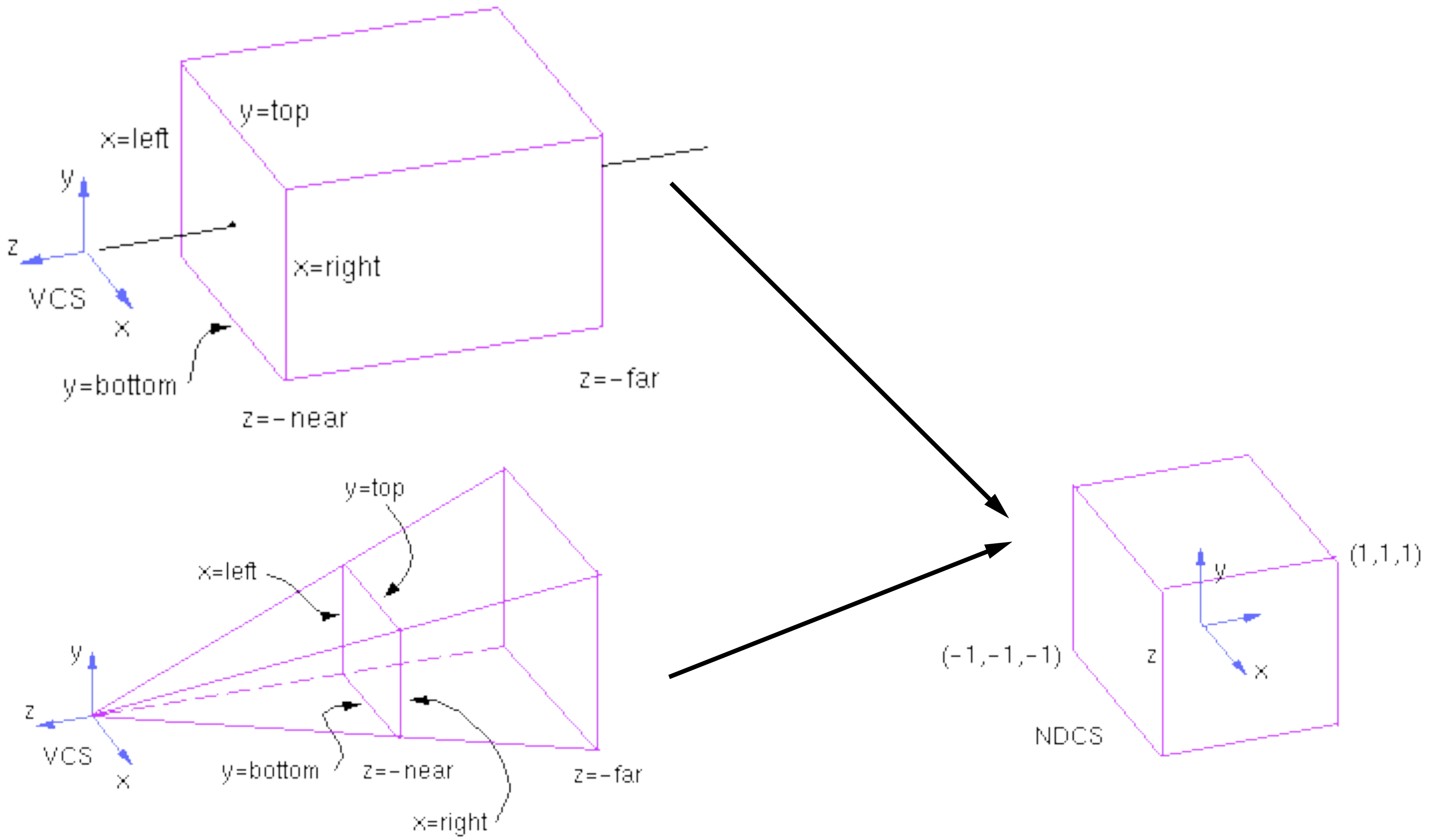
Projections in the Graphics Pipeline

View volumes

- Our pipeline supports two projections:
  - *Orthographic*
  - *Perspective*
- This stage also defines the view window
- What is visible with each projection?
  - *a cube*
  - *or a pyramid*



View volumes

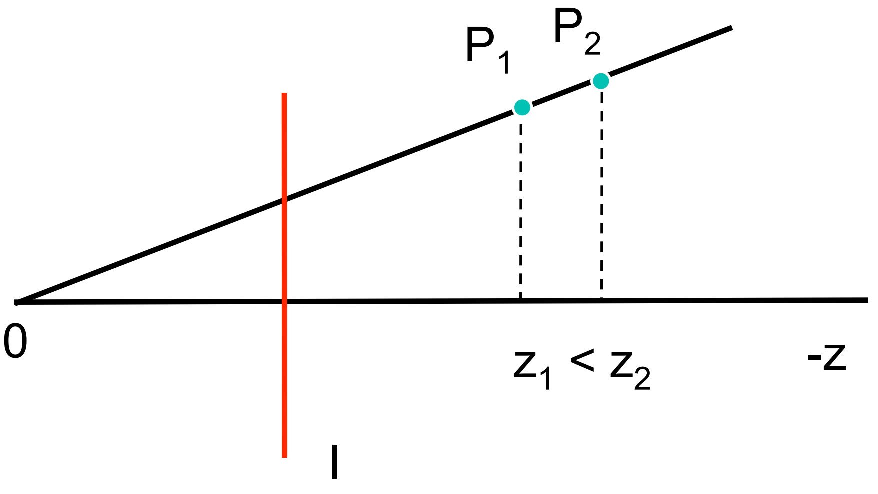


Transformation vs Projection

We want to keep z

Why?

- Pseudodepth



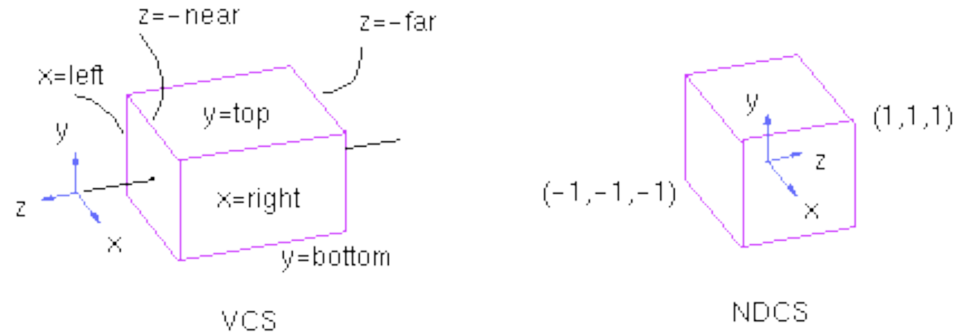
Derivation of the orthographic transformation

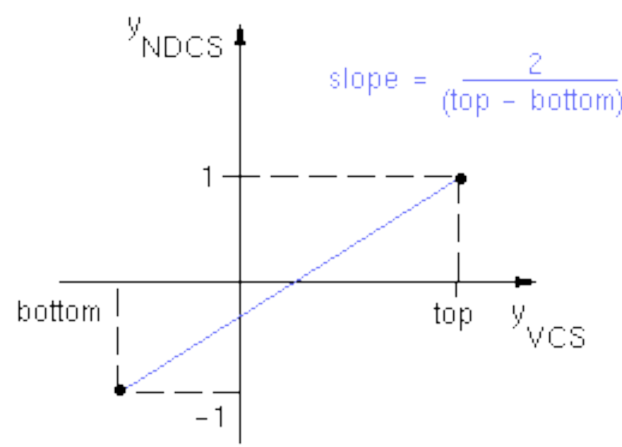
Map each axis separately:

- Scaling and translation

Let's look at y:

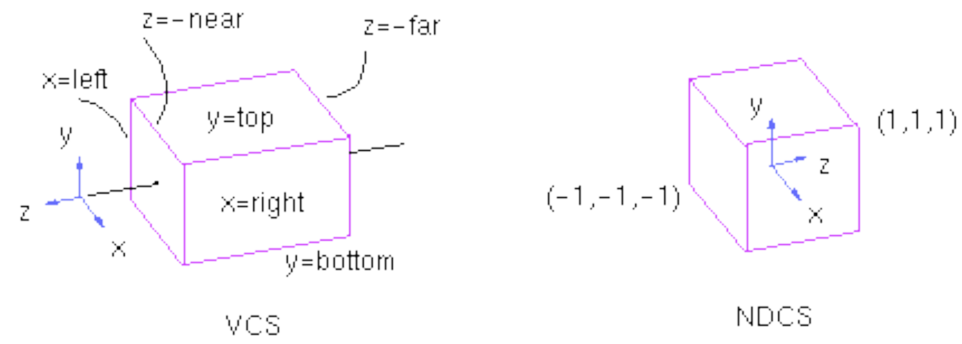
- y' = ay+b such that
  bottom → -1
  top → 1





Derivation of the orthographic transformation

Scaling and Translation



$$y_{VCS} \rightarrow y$$

$$y_{NDCS} \rightarrow y'$$

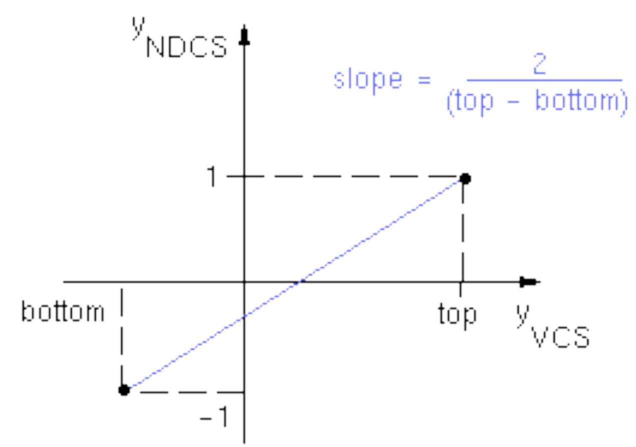
$$(y_b, y'_b) = (bottom, -1) \quad and$$

$$(y_t, y'_t) = (top, 1)$$

$$Line \quad equation \quad \frac{y' - y'_b}{y - y_b} = \frac{y'_t - y'_b}{y_t - y_b}$$

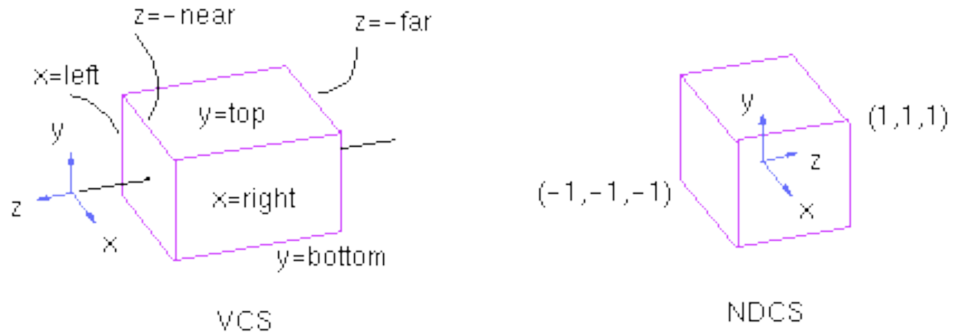
$$\frac{y' - (-1)}{y - bottom} = \frac{1 - (-1)}{top - bottom} \rightarrow$$

$$y' = \frac{2}{top - bottom} y - \frac{top + bottom}{top - bottom}$$



All three coordinates

Scaling and Translation

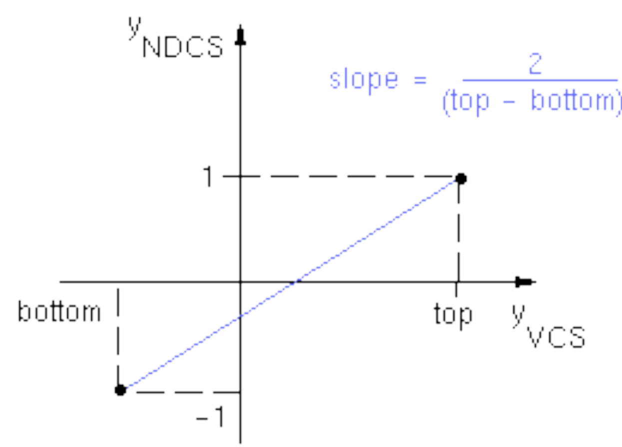


$Similarly,$

$$x' = \frac{2}{right - left}x - \frac{right + left}{right - left}$$

$$y' = \frac{2}{top - bottom}y - \frac{top + bottom}{top - bottom}$$

$$z' = \frac{-2}{far - near}z - \frac{far + near}{far - near}$$



Matrix form

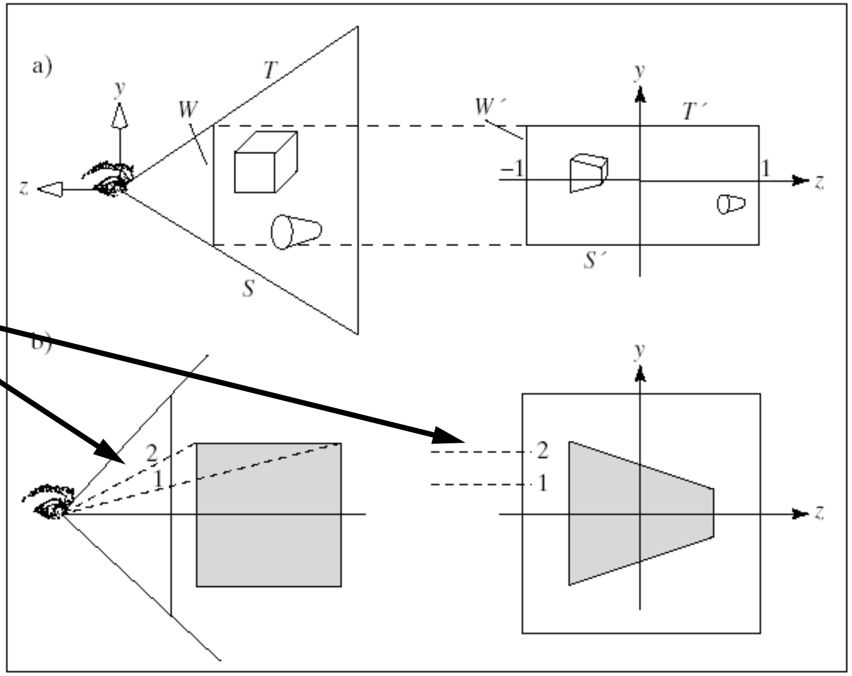
$$P' = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} P$$

Perspective transformation

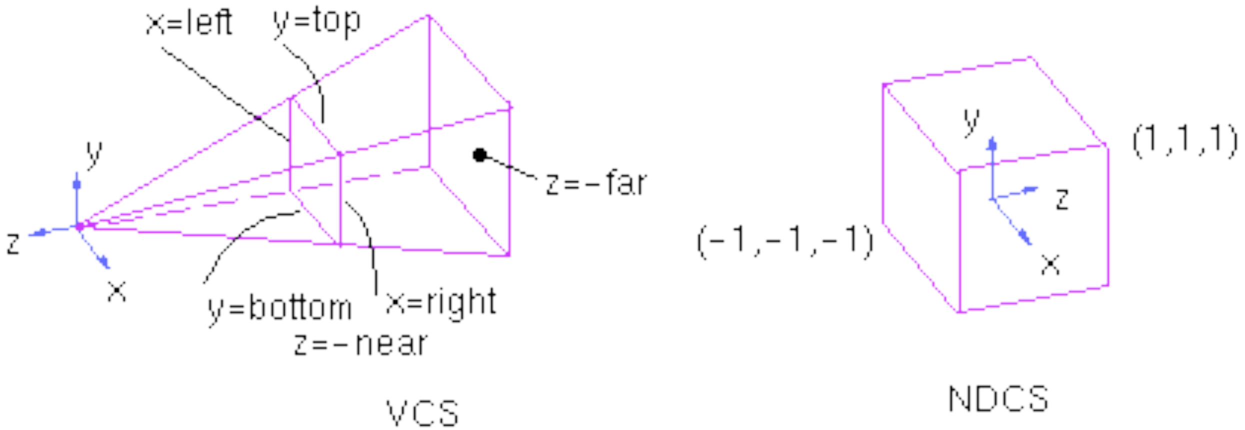
Warps the view volume and the objects in it.

- Eye becomes a point at infiinity, and the projection rays become parallel lines (orthographic projection)

- We also want to keep z



Derivation of the perspective transformation

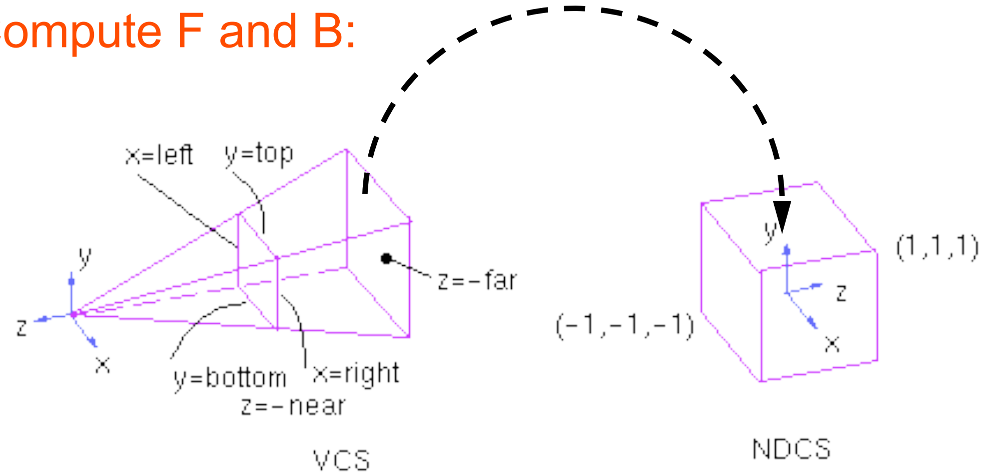


It is basically a mapping of planes

Normalized view volume is a left handed system!

Deriving the Matrix

Compute F and B:



$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{h} \end{bmatrix} = \begin{bmatrix} E & 0 & A & 0 \\ 0 & F & B & 0 \\ 0 & 0 & C & D \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Top plane:

$$Before\ \ projection\ \ y = -zt/n$$

$$After\ \ projection\ \ and\ \ division\ \ \tilde{y}/\tilde{h} = 1$$

From the matrix multiplication:

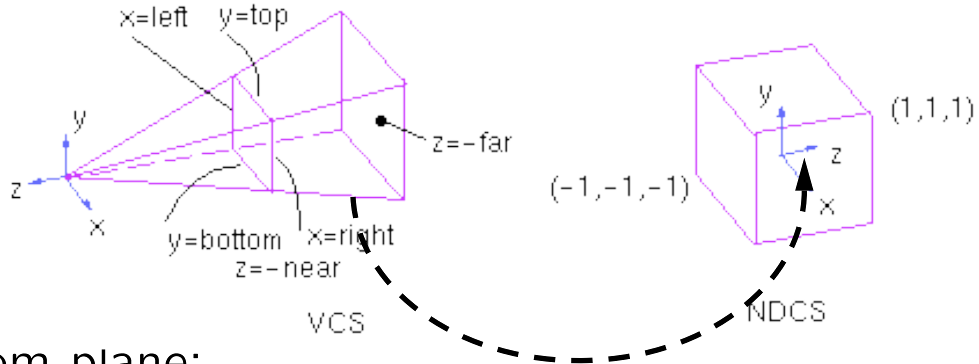
$$\tilde{y} = Fy + Bz \rightarrow (perspective\ \ division)$$

$$\tilde{y}/\tilde{h} = (Fy + bz)/\tilde{h} \rightarrow 1 = (Fy + Bz)/(-z) \rightarrow$$

$$Ft/n - B = 1 \quad (1)$$

Forming the second equation

From bottom plane



$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{h} \end{bmatrix} = \begin{bmatrix} E & 0 & A & 0 \\ 0 & F & B & 0 \\ 0 & 0 & C & D \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Bottom plane:

$$Before\ \ projection\ \ y = -zb/n$$

$$After\ \ projection\ \ and\ \ division\ \ \tilde{y}/\tilde{h} = -1$$

From the matrix multiplication:

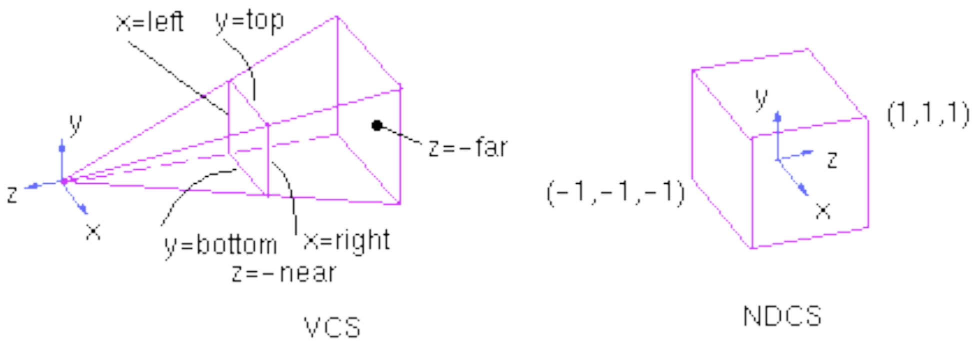
$$\tilde{y} = Fy + Bz \rightarrow (perspective\ \ division)$$

$$\tilde{y}/\tilde{h} = (Fy + bz)/\tilde{h} \rightarrow -1 = (Fy + Bz)/(-z) \rightarrow$$

$$Fb/n - B = -1 \quad (2)$$

Solving the 2x2 system

Compute F and B:

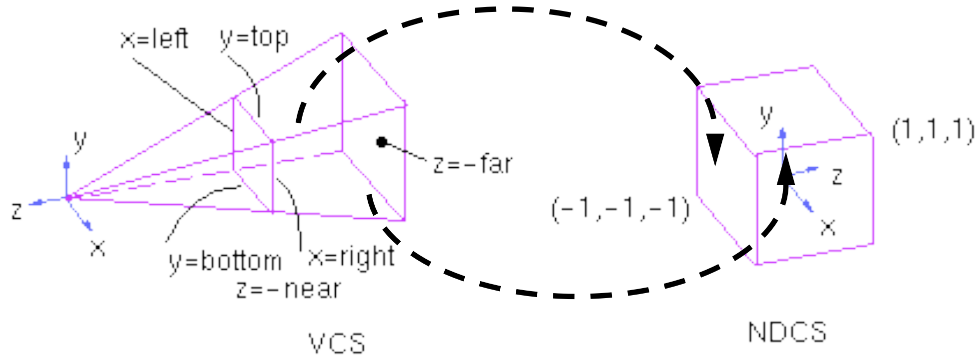


$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{h} \end{bmatrix} = \begin{bmatrix} E & 0 & A & 0 \\ 0 & F & B & 0 \\ 0 & 0 & C & D \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\left\{ \begin{array}{c} \frac{t}{n}F - B = 1 \\ \frac{b}{n}F - B = -1 \end{array} \right\} \Rightarrow \begin{array}{c} F = \frac{2n}{t-b} \\ B = \frac{t+b}{t-b} \end{array}$$

Similarly for x

Compute E and A:



$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{h} \end{bmatrix} = \begin{bmatrix} E & 0 & A & 0 \\ 0 & F & B & 0 \\ 0 & 0 & C & D \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\left\{ \begin{array}{c} \frac{r}{n}E - A = 1 \\ \frac{l}{n}E - A = -1 \end{array} \right\} \Rightarrow \begin{array}{c} E = \frac{2n}{r-l} \\ A = \frac{r+l}{r-l} \end{array}$$

Similarly z

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{h} \end{bmatrix} = \begin{bmatrix} E & 0 & A & 0 \\ 0 & F & B & 0 \\ 0 & 0 & C & D \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Compute C and D

$$\tilde{z} = Cz + D \quad (1)$$

$$\tilde{h} = -z \quad (2)$$

$$\textbf{Near plane:}$$

$$z = -n \rightarrow \frac{\tilde{z}}{\tilde{h}} = \frac{\tilde{z}}{n} = -1 \quad (3)$$

$$(1),(2),(3) \rightarrow C(-n)/n + D/n = -1 \quad (4)$$

$$\textbf{Similarly for far plane:}$$

$$z = f \rightarrow \frac{\tilde{z}}{\tilde{h}} = 1 \quad (5)$$

$$(1),(2),(5) \rightarrow C(-f)/f + D/f = 1 \quad (6)$$

$$\textbf{From}(4),(6) \quad C = -(f+n)(f-n), \quad D = -2fn/(f-n)$$

Putting everything together

$$\begin{bmatrix} x' \\ y' \\ z' \\ h' \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Putting everything together

$$\begin{bmatrix} x' \\ y' \\ z' \\ h' \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Same as orthographic except the scaling and the position of the x,y translation parameters

Orthographic projection in OpenGL

glMatrixMode(GL_PROJECTION) ;

glLoadIdentity() ;

Followed by one of:

glOrtho(left,right,bottom,top,near,far) ;

near plane at z = -near

far plane at z = -far

gluOrtho2D(left,right,bottom,top) ;

assumes near = 0 far = 1

Perspective projection in OpenGL

glMatrixMode(GL_PROJECTION) ;

glLoadIdentity() ;

Followed by one of:

glFrustrum(left,right,bottom,top,near,far) ;

near plane at z = -near

far plane at z = -far

gluPerspective(fovy, aspect,bottom,top) ;

fov measured in degrees and center at 0

Put the code in init or reshape callback.

Matrix pre-multiplies Modelview matrix

*M = Mproj*CM*

Important

Projection parameters are given in CAMERA Coordinate system (Viewing).

So if camera is at z = 50 and you give near = 10 where is the near plane with respect to the world?

Important

Projection parameters are given in CAMERA Coordinate system (Viewing).

So if the camera is at z = 50 and you give near = 10 where is the near plane with respect to the world?

Transformed by inverse(Mvcs)

Nonlinearity of perspective transformation

Tracks:

Left: x= -1, y = -1

Right: x = 1, y = -1

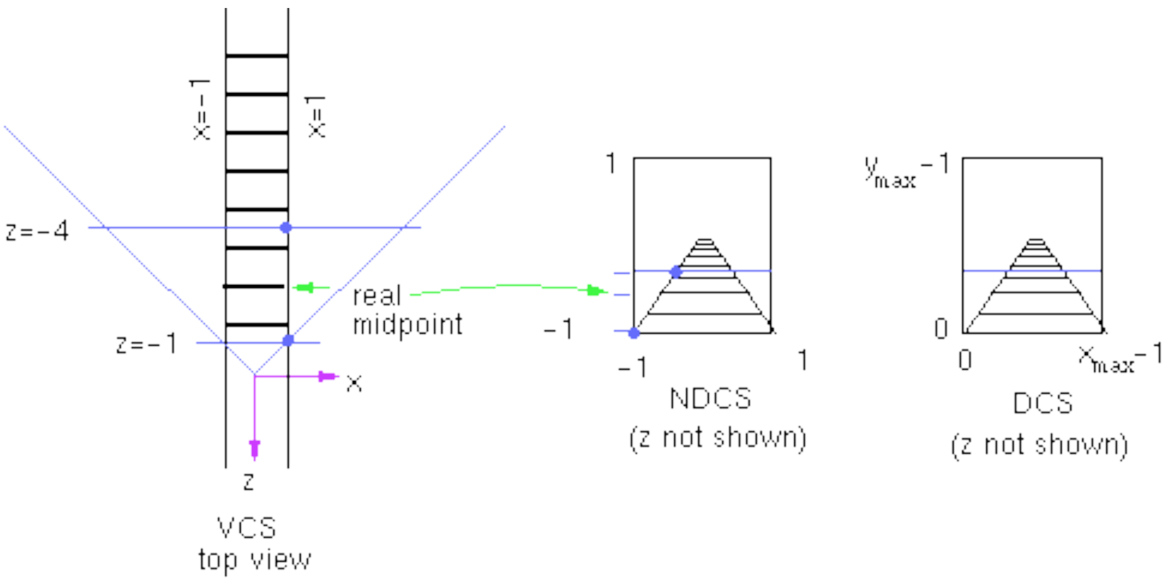
Z = -inf, inf

View volume:

Left = -1, right = 1

Bot = -1, top = 1

Near = 1, far = 4

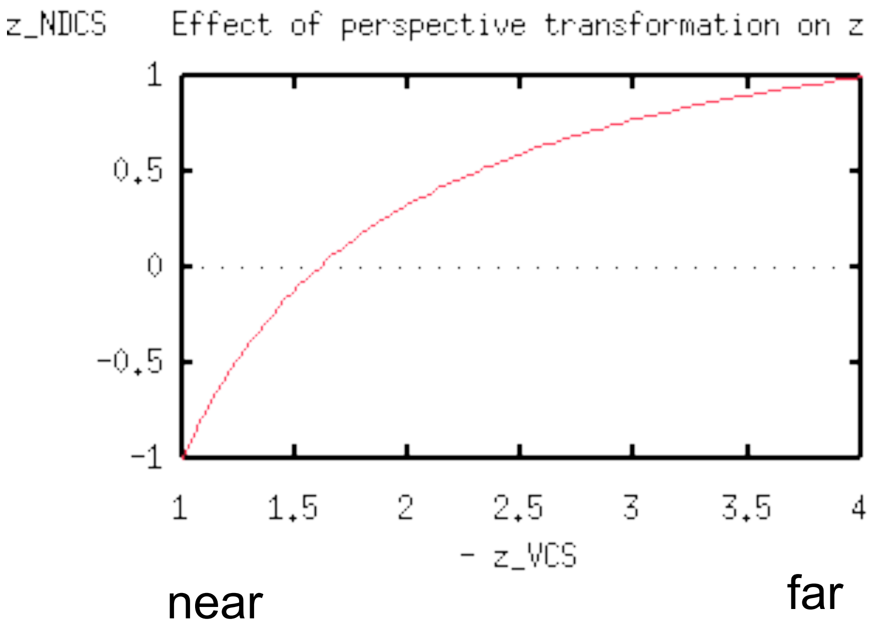


Comparison of cs's

VCS	*CCS*	*NDCS*
X	X' = EX + AZ	X''=X'/W'
Y	Y' = FY + BZ	Y'' = Y'/W'
Z	Z' = CZ + D	Z'' = Z'/W'
Point	Point'	Point''
1	X'= X = 1	X'' = -1/Z
-1	Y' = Y = -1	Y'' = 1 /Z
Z	Z' = -5Z/3 – 8/3	Z'' = 5/3 +8/(3Z)
	W' = -Z	

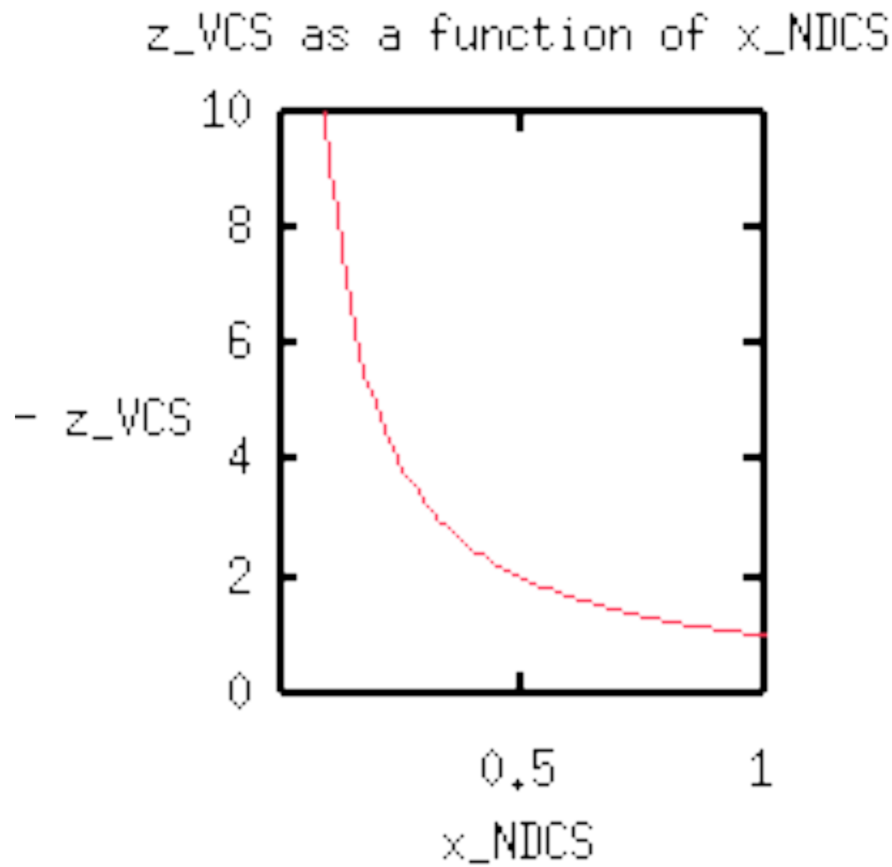
Z in NDCS vs –Z in VCS

Z" = 5/3 +8/(3Z)



Other comparisons

$X'' = -1/Z \rightarrow Z = -1/X''$

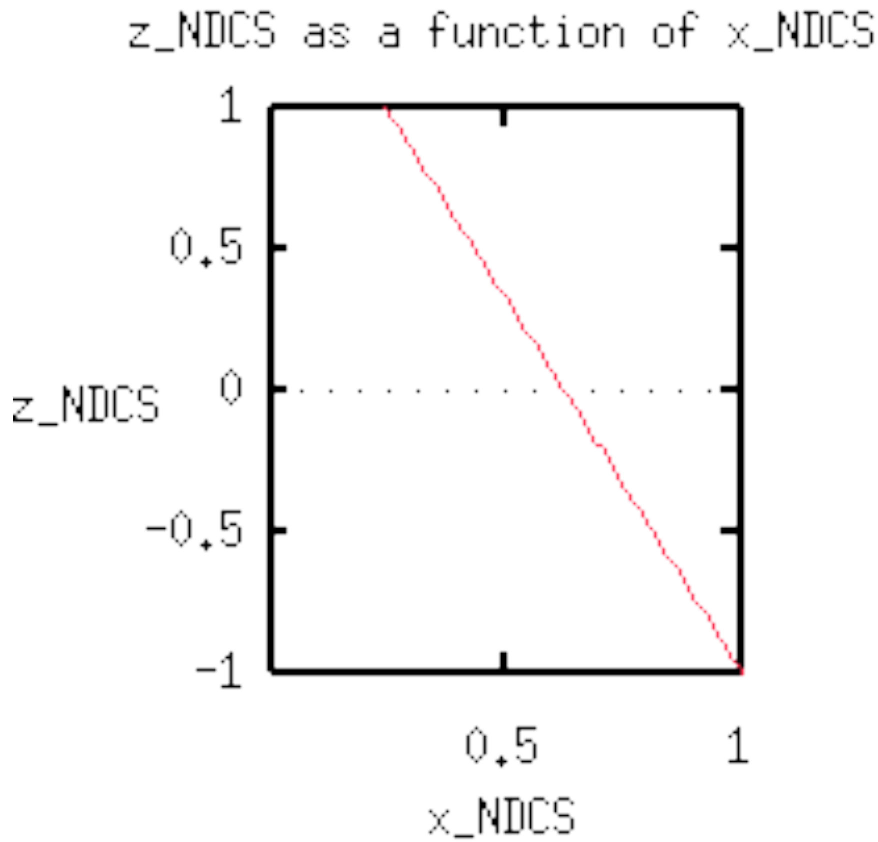


Other comparisons

X" = -1/Z

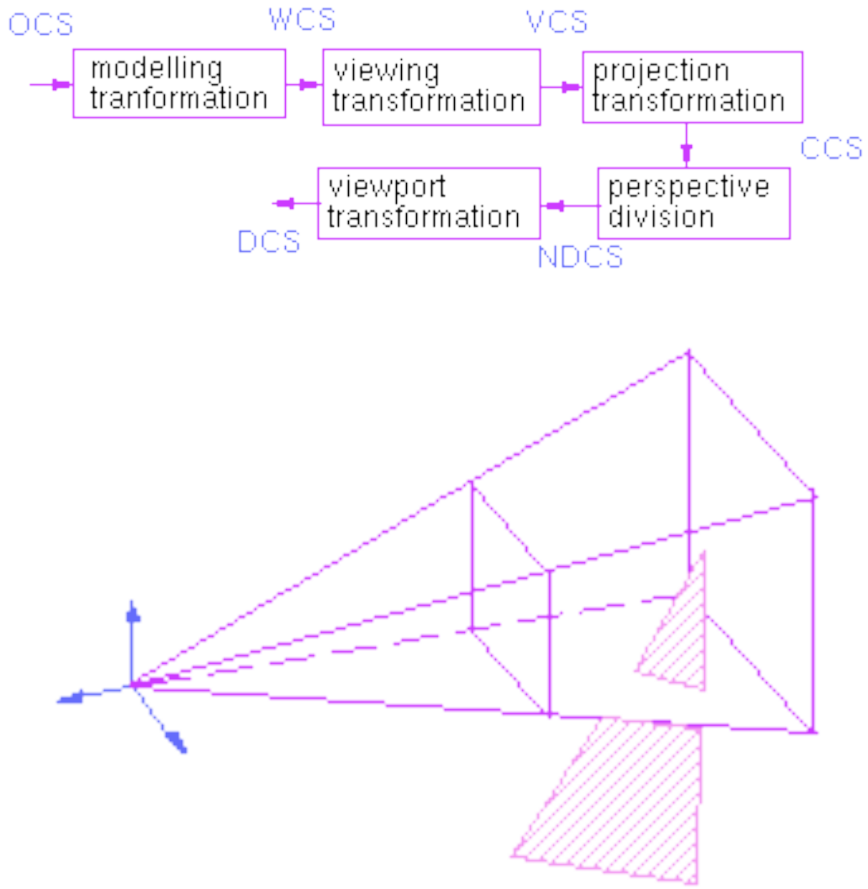
Z" = 5/3 +8/(3Z)

→ *Z" = 5/3-(8/3)X"*



3D Clipping

Keep what is visible



Background (reminder)

Plane equations

Implicit

$F(x,y,z) = Ax + By + Cz + D = \mathbf{N} \bullet P + D$

Points on Plane $F(x,y,z) = 0$

Parametric

$Plane(s,t) = P_0 + s(P_1 - P_0) + t(P_2 - P_0)$

P_0, P_1, P_2 not colinear

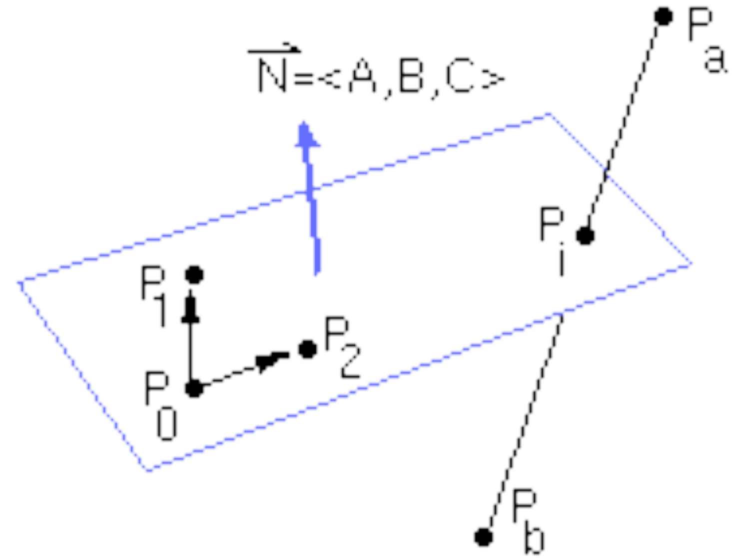
or

$Plane(s,t) = (1 - s - t)P_0 + sP_1 + tP_2$

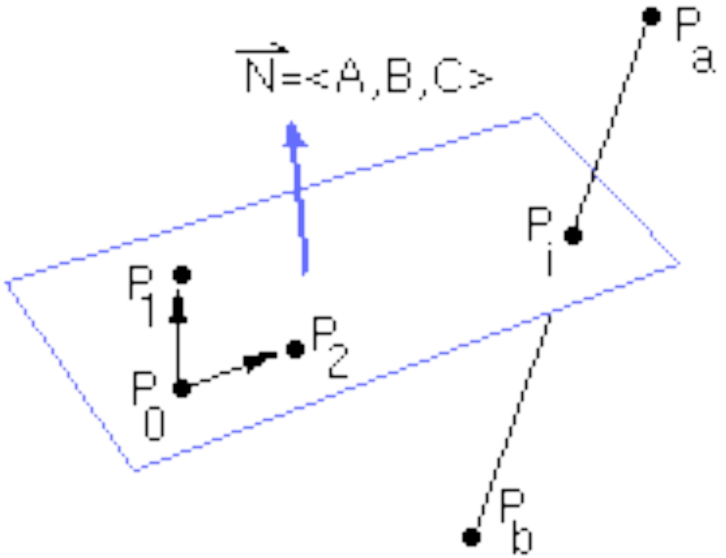
$Plane(s,t) = P_0 + sV_1 + tV_2$ where V_1, V_2 basis vectors

Explicit

$z = -(A/C)x - (B/C)y - D/C,\ C \neq 0$



Intersection of line and plane



$$\vec{N} \bullet (P_a + t\,(P_b - P_a)) + D = 0$$

$$t = \frac{-D - \vec{N} \bullet P_a}{\vec{N} \bullet P_b - \vec{N} \bullet P_a} = \frac{-F(P_a)}{F(P_b) - F(P_a)}$$

Orthographic view volume

Planes

Normals pointing inside

left: x - left = 0

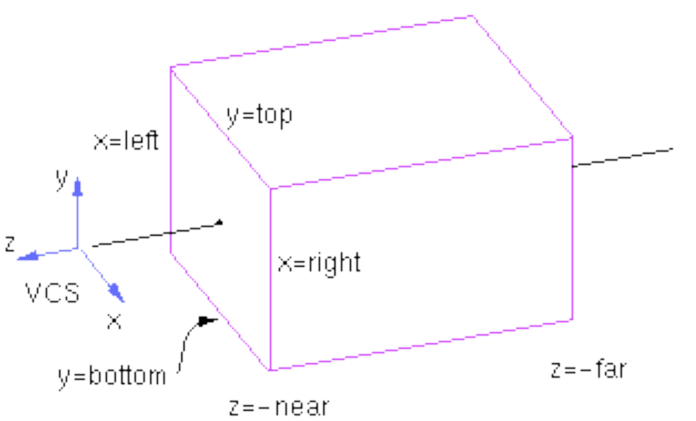
right: -x + right = 0

bottom: y - bottom = 0

top: -y + top = 0

front: -z - near = 0

back: z + far = 0



Perspective View volume

Planes

Normals pointing inside

left: x + left*z/near = 0

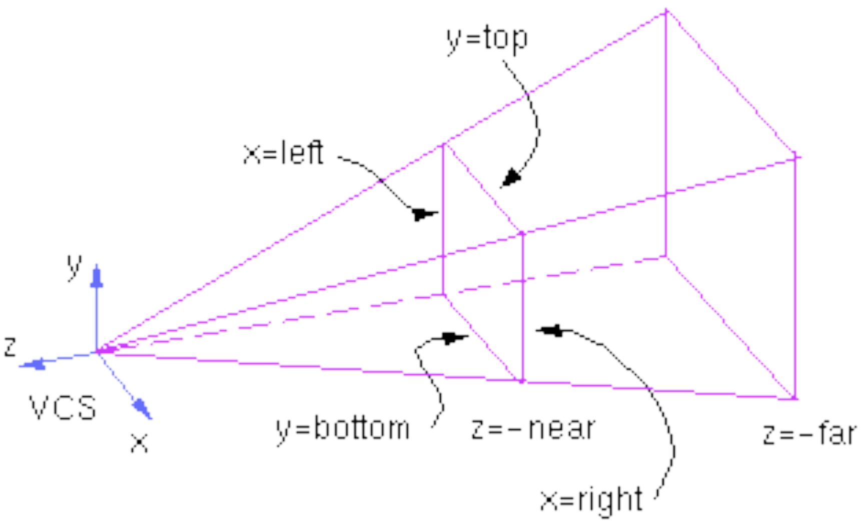
right: -x - right*z/near = 0

top: -y - top*z/near = 0

bottom: y + bottom*z/near = 0

front: -z - near = 0

back: z + far = 0



Clipping in NDCS

Normalized view volume

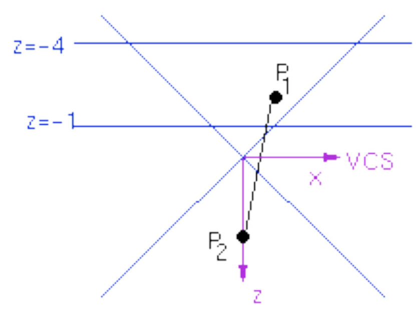
- Constant planes
- Lines in VCS lines NDCS

Problem

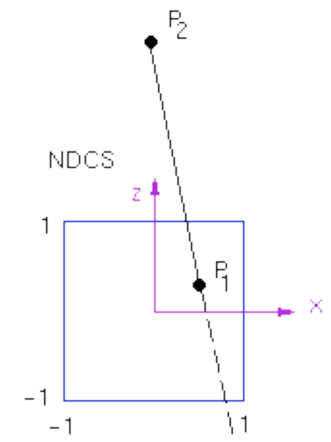
- Z coordinate loses its sign

	P_1	P_2
VCS	(1, 0, −2)	(0,0,2)
CCS	(1, 0, 2/3, 2)	(0,0,−6,−2)
NDCS	(1/2, 0, 1/3)	(0, 0, 3)

VCS



NDCS



Clipping in CCS

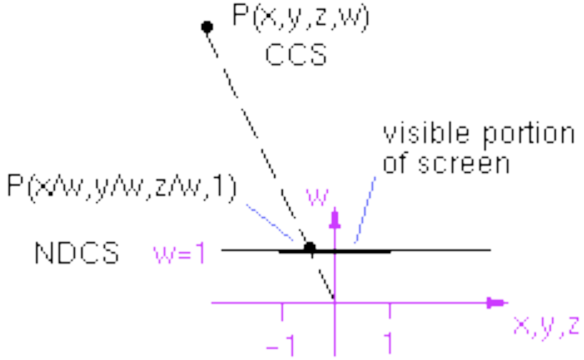
We'll define the clipping region in CCS by first looking at the clipping region in NDCS:

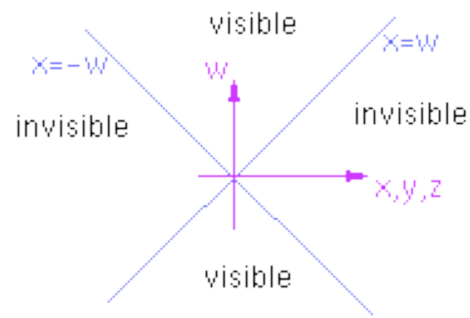
$-1 <= x/w <= 1$

This means that in CCS, we have:

$-w <= x <= w$

Similarly for y,z

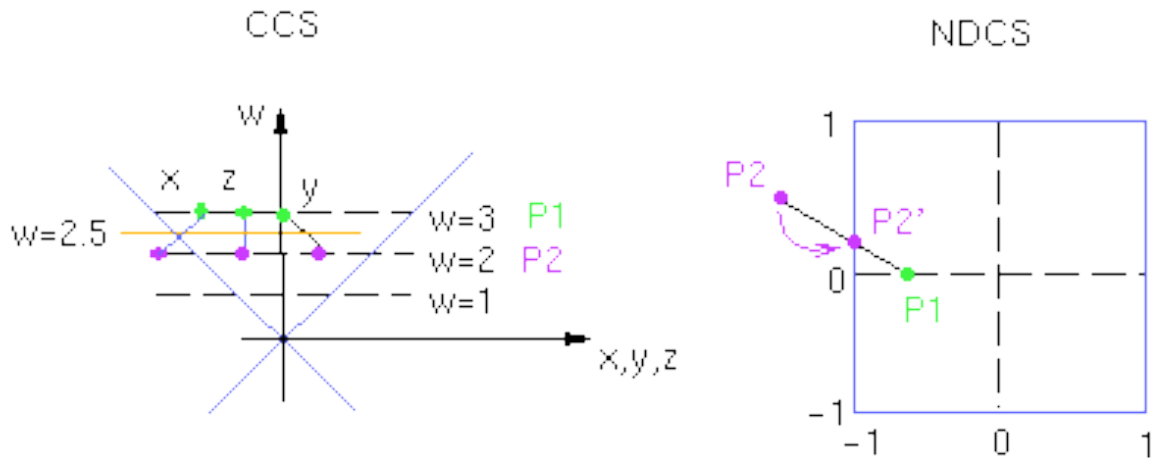




Example

The perspective transformation creates

W = -z



unclipped CCS
P1(-2,0,-1,3)
P2(-3,1,-1,2)

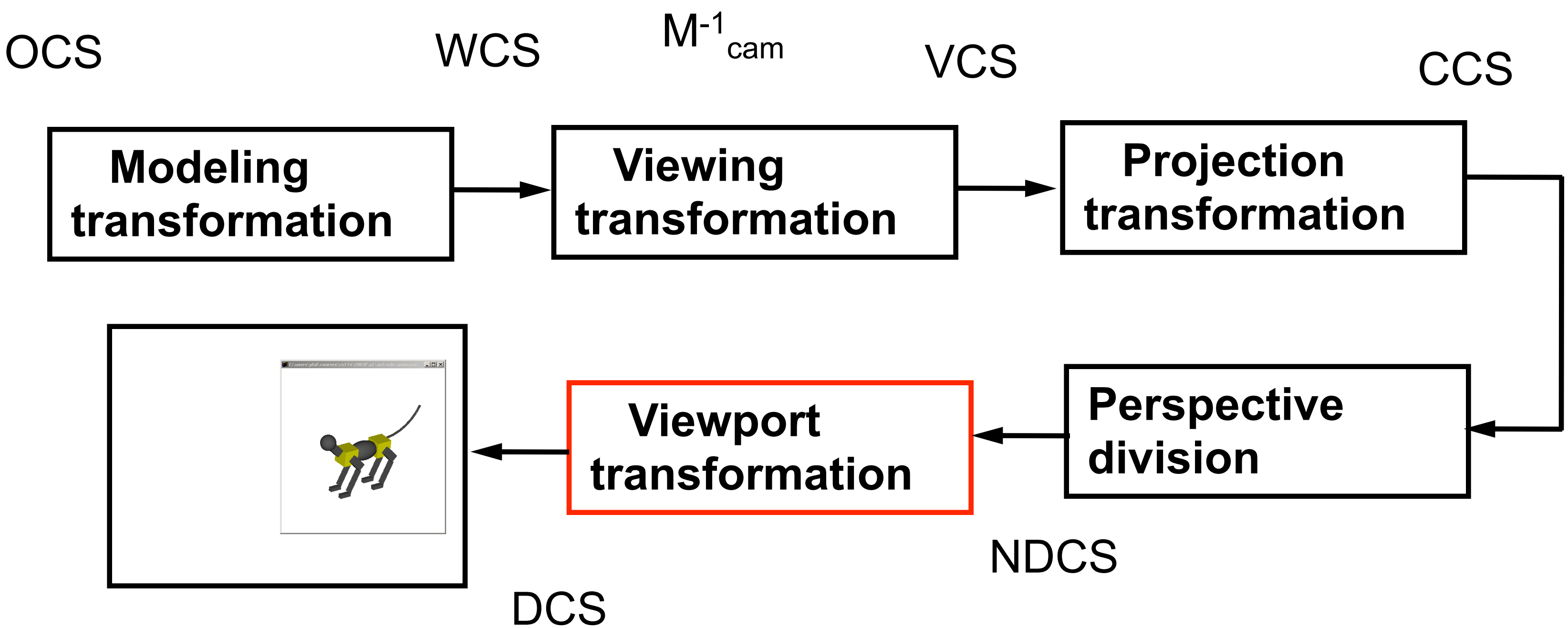
clipped CCS
P1(-2,0,-1,3)
P2(-2.5,0.5,-1,2.5)

unclipped NDCS
P1(-0.67,0,-0.33)
P2(-1.5,0.5,-0.5)

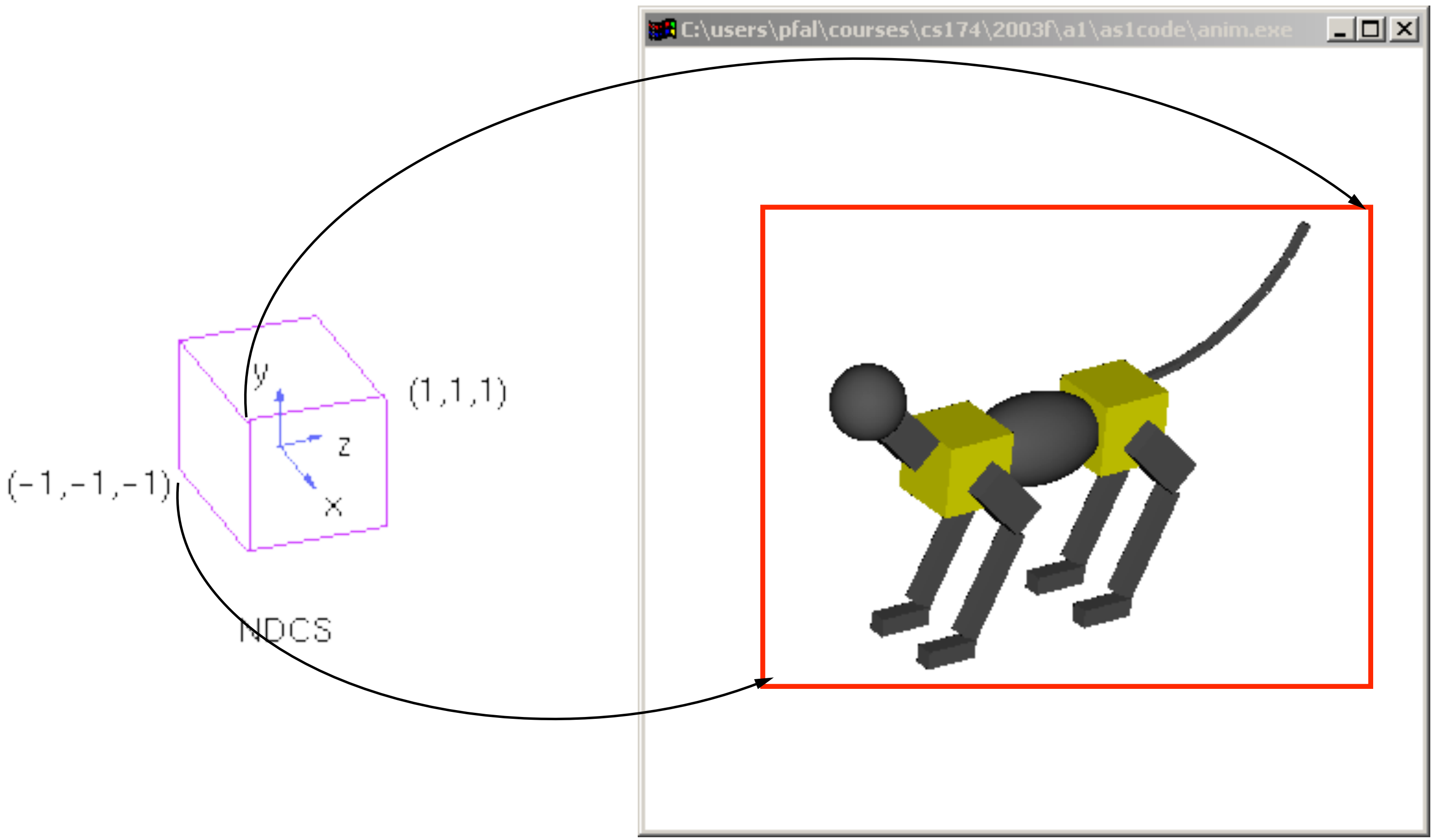
clipped NDCS
P1(-0.67,0,-0.33)
P2(-1,0.2,-0.4)

Typo: they should have different z

Viewport transformation



Viewport



C:\users\pfal\courses\cs174\2003f\a1\as1code\anim.exe
y
z
x
(1,1,1)
(-1,-1,-1)
NDCS

Viewport matrix

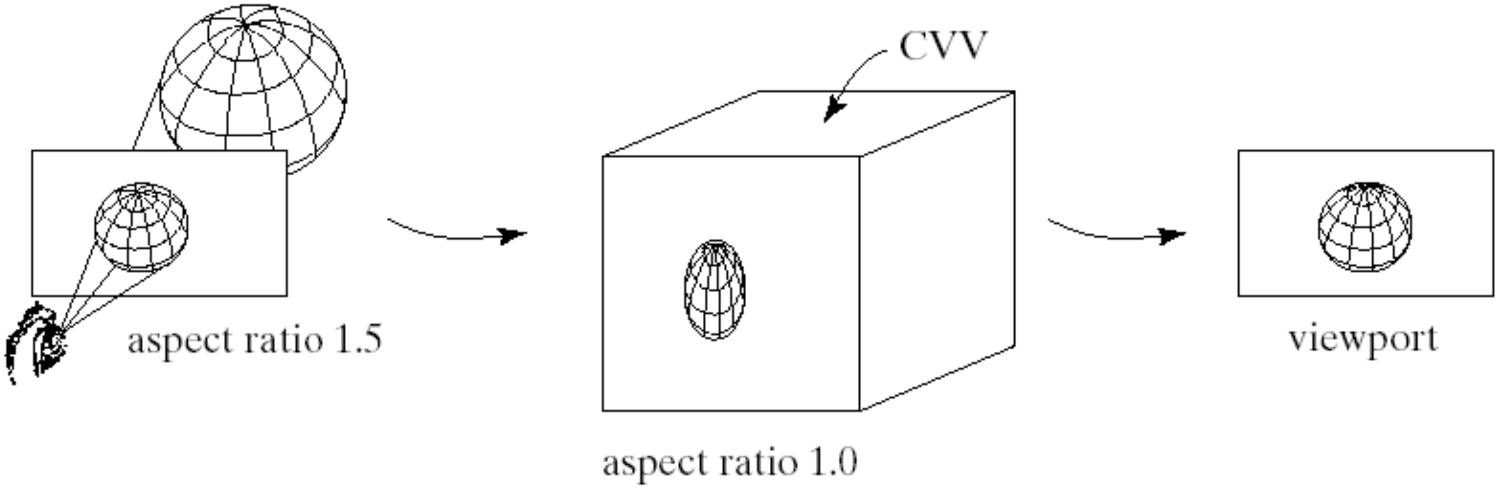
Scales the x,y to the dimensions of the viewport

Scales z to be in [0,1]

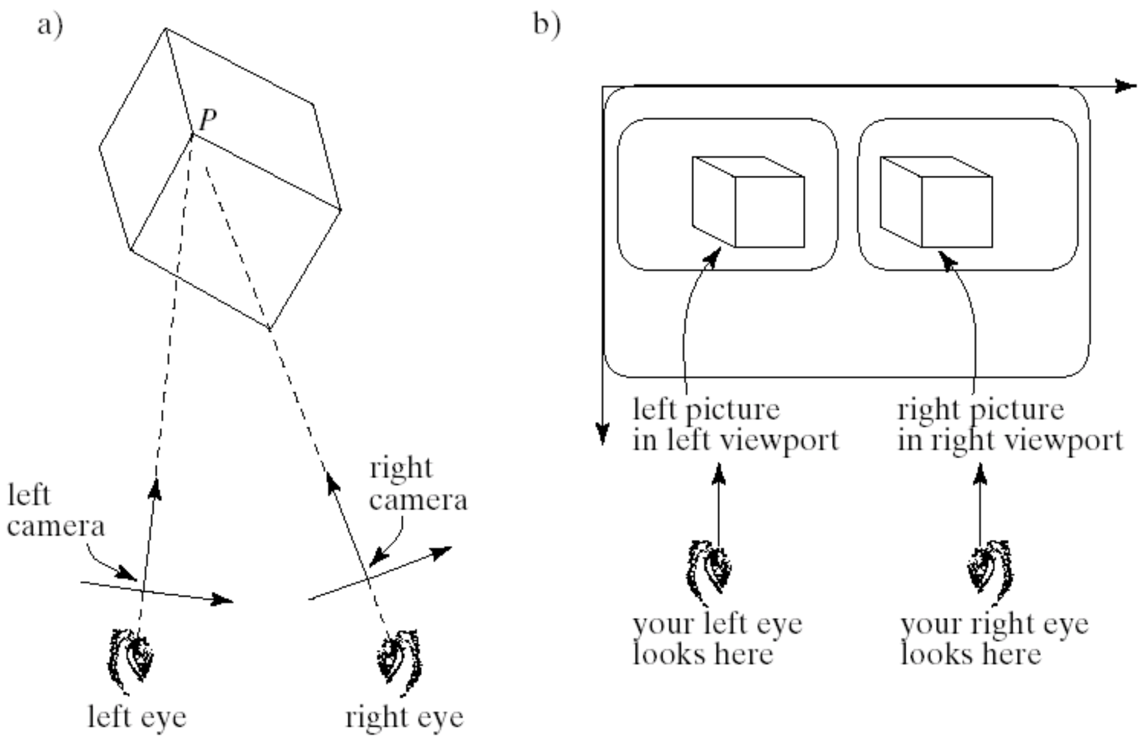
Matrix form left as an exercise.

Why viewports?

Undo the distortion of the projection transformation



Stereo views



Viewport in OpenGL

glViewport(GLint x, GLint y, GLsizei width, GLsizei height) ;

x,y: lower left corner of viewport rectangle in pixels

width, height: width and height of viewport.

Put the code in reshape callback.

Transformations in the pipeline