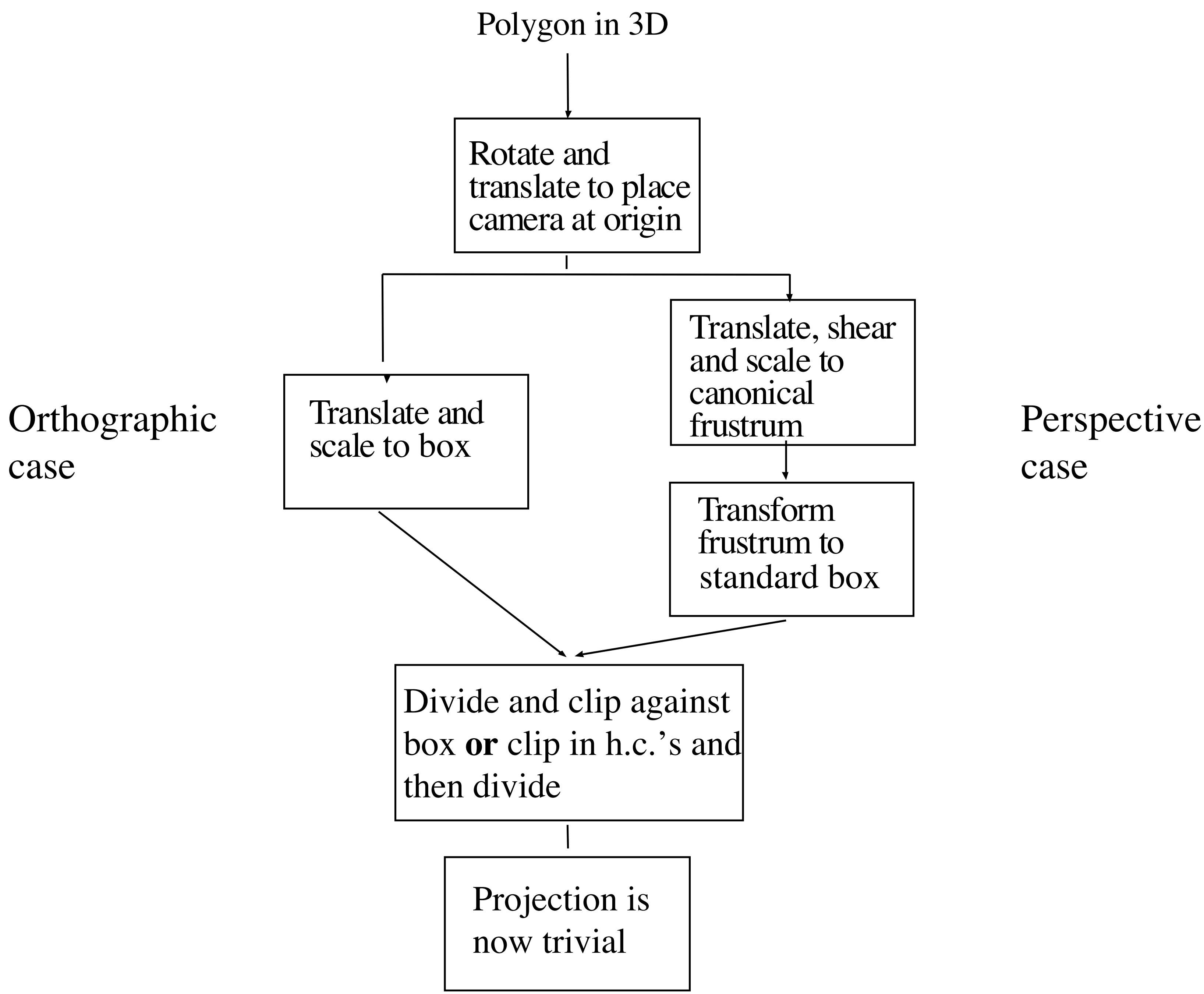
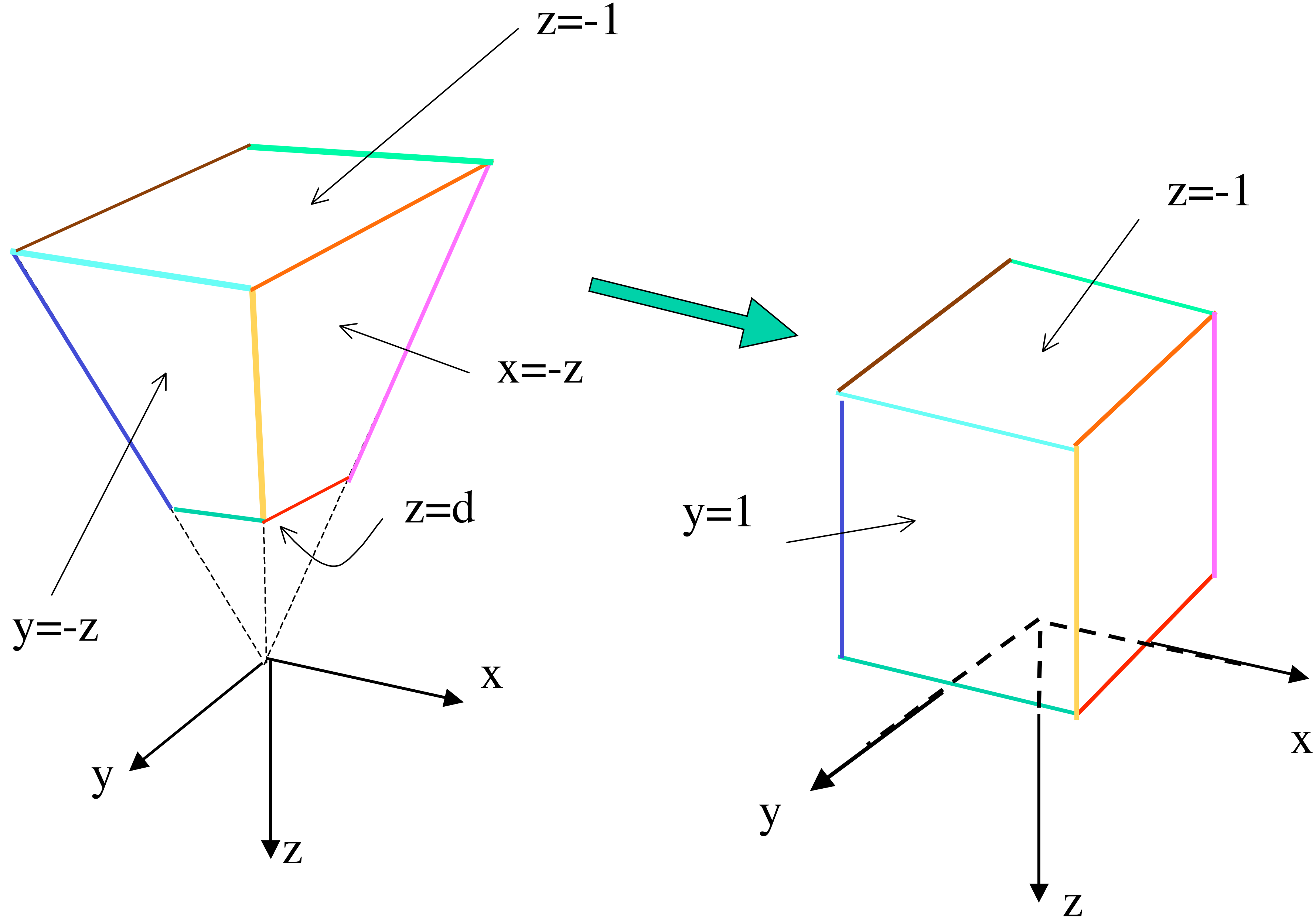
Plan B: Clipping in homogenous coords

- For any camera, can turn the view frustrum into a regular parallelepiped (box). We will use the box bounded by $x = \pm 1$, $y = \pm 1$, $z = -1$, and $z = 0$.
- Advantages
  - Simplified clipping in homogenous coordinates
  - Extends to cases where we use homogenous coordinates to represent additional information (and W could be negative).
  - Can simplify visibility algorithms.
- Approach: clever use of homogenous coordinates

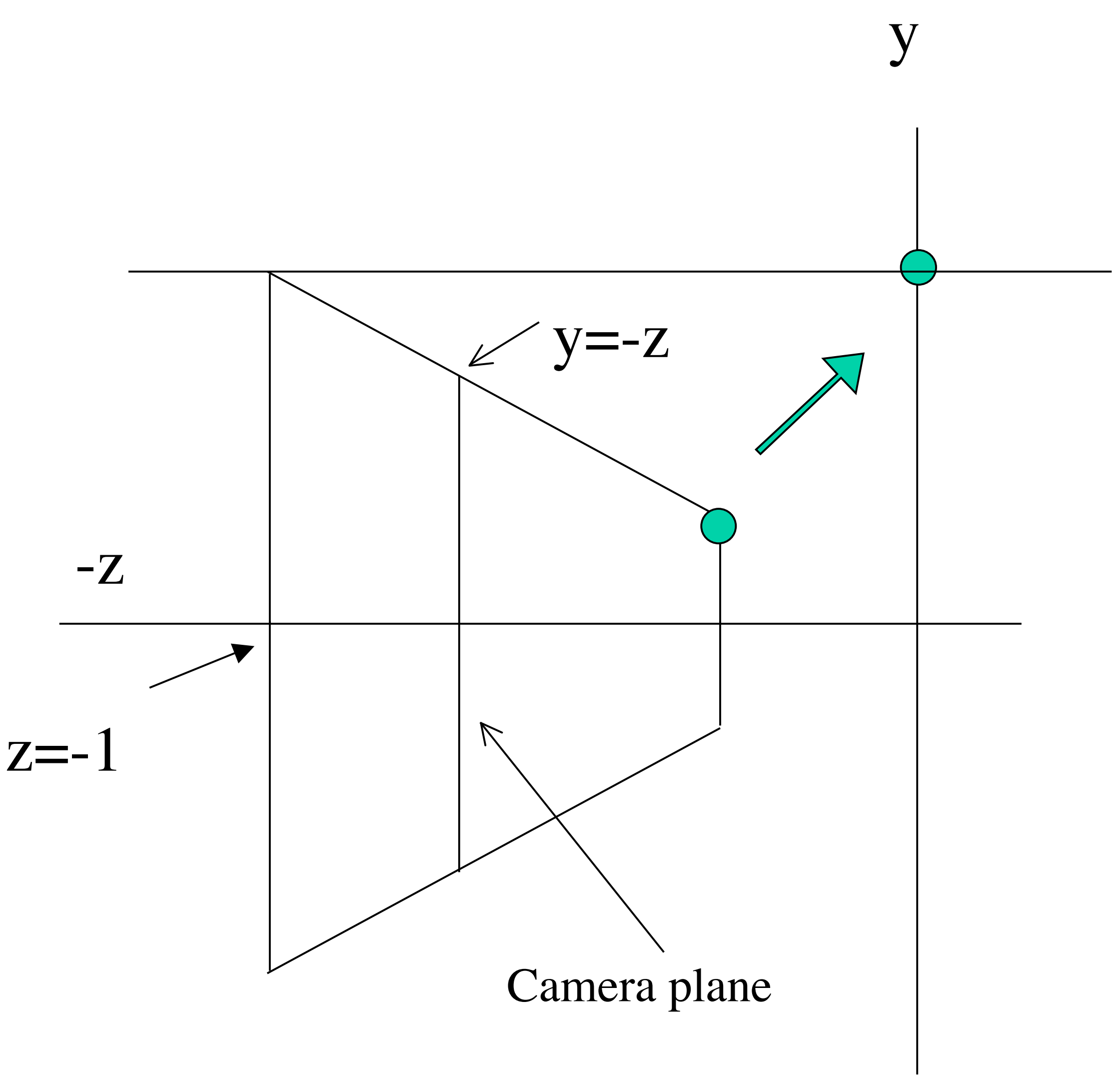
(Will be needed for assignment three)


Polygon in 3D
Rotate and translate to place camera at origin
Orthographic case
Translate and scale to box
Translate, shear and scale to canonical frustrum
Perspective case
Transform frustrum to standard box
Divide and clip against box **or** clip in h.c.'s and then divide
Projection is now trivial

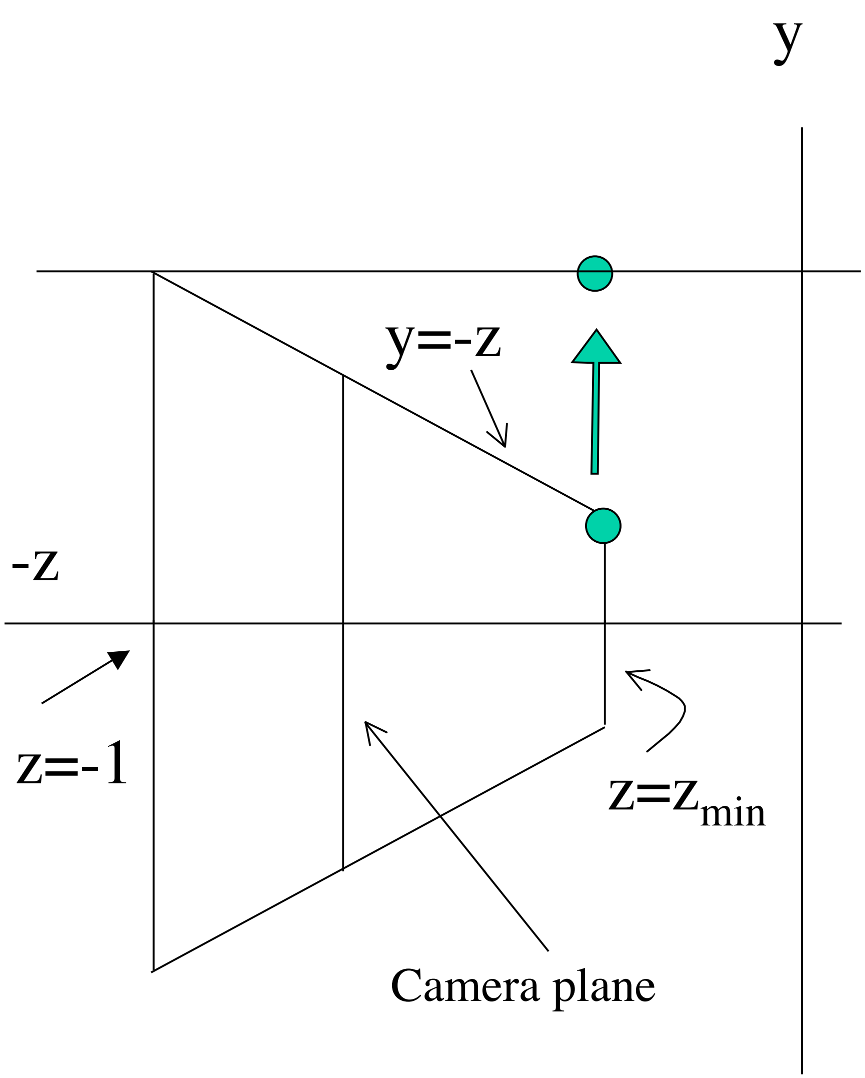
Transforming canonical frustum to box



Transforming canonical frustum to box



Transforming canonical frustum to box



On top, y-->1, so scaling is (1/y)
Recall that y=-x there.

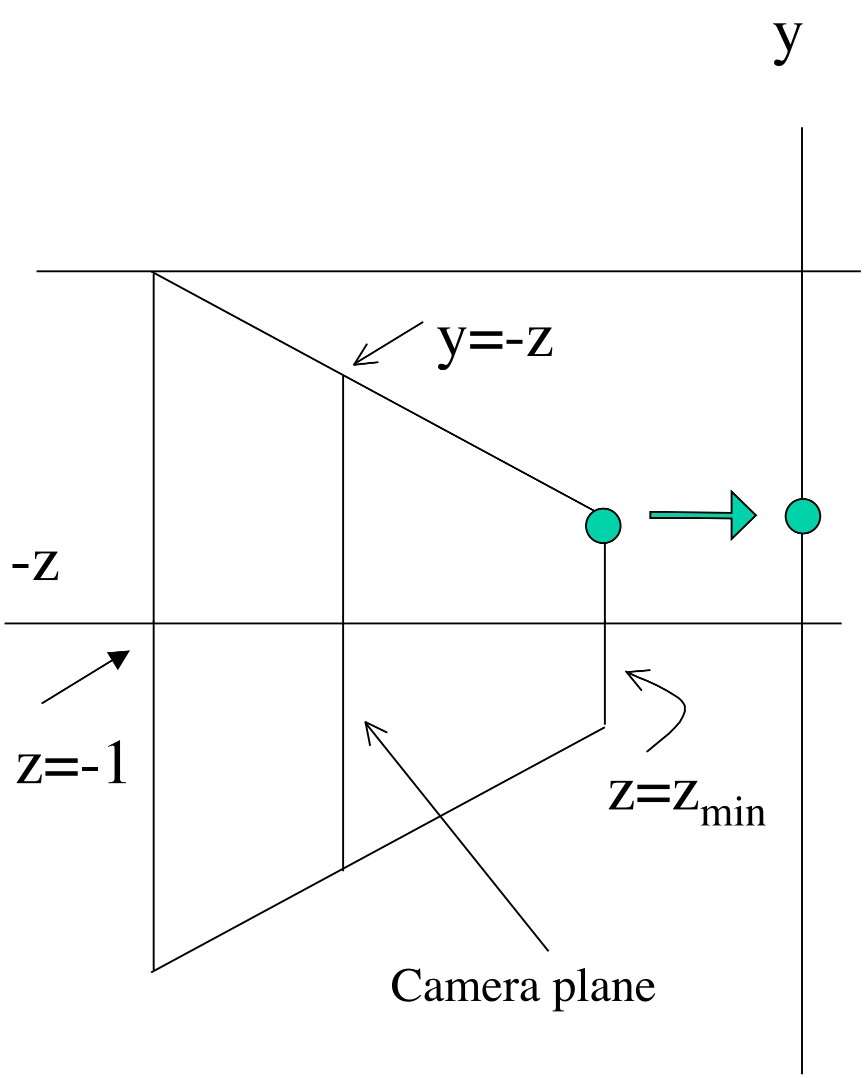
On bottom, y-->-1 so scaling is (-1/y). Recall that y=x there.

So scaling is y'= y/(-z)

Similarly, x' = x/(-z)

Transformation is **non-linear**, but in h.c., can use w=-z.

Transforming canonical frustum to box



For z, we shift near plane to origin. But now box is too small. Specifically it has z dimension $(1 + z_{min})$ (recall z_{min} is negative)

So we have an extra scale factor $1 / (1 + z_{min})$ and thus $z' = (z - z_{min}) / (1 + z_{min})$

But we want x and y to work nicely in h.c., we use

$$z' = ((z - z_{min}) / (1 + z_{min}))/(-z)$$

(Depths **also** transform **non-linearly**)

In h.c.,

$x \Rightarrow x$

$y \Rightarrow y$

$z \Rightarrow (z - z_{min}) / (1 + z_{min})$

$1 \Rightarrow -z$

So, the matrix is ?

In h.c.,

$x \Rightarrow x$

$y \Rightarrow y$

$z \Rightarrow (z - z_{min}) / (1 + z_{min})$

$1 \Rightarrow -z$

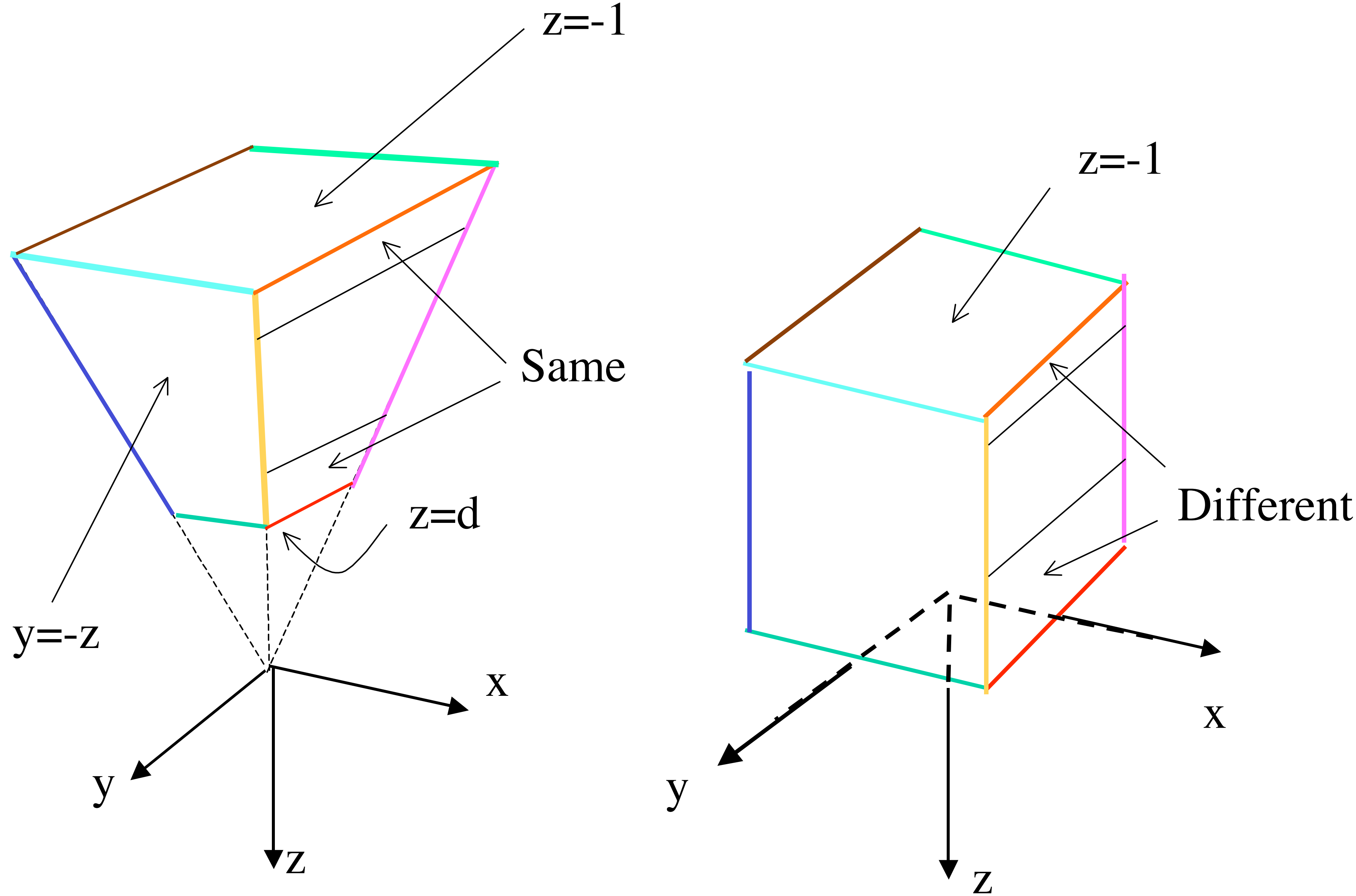
So, the matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{1}{1+z_{min}} & \dfrac{-z_{min}}{1+z_{min}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Mapping to standard parallel projection view volume (additional comments)

- The mapping from $[z_{min}, -1]$ to $[0,-1]$ is non-linear. (Of course, there exists a linear mapping, but not if we want everything else to work out nicely in h.c.).
- So a change in depth of $\triangle$ D at the near plane maps to a larger depth difference in screen coordinates than the same $\triangle$ D at the far plane.
- But order is preserved (important!); the function is monotonic (proof?).
- And lines are still lines (proof?) and planes are still planes (important!).

Transforming canonical frustum to box



Clipping in homogeneous coord.’s

- We have a cube, but its representation is in h.c., so we must divide
- Clipping requires a test for inside or out; in 2D, we used x>xmax, x<xmin, etc.
- In 3D, for clipping against the cube, we could use x>1, x<-1, etc.
- But to do this, we would have to convert to cartesian coords by division.
- Dividing before clipping is inefficient if many points are excluded, so we clip in h.c.’s.

Clipping in homogeneous coord.'s

- Write h.c.'s in caps, ordinary coords in lowercase.
- Consider case x>1, x<-1
- Rearrange clipping inequalities:

$$\left(\frac{X}{W}\right) > 1$$

$$\left(\frac{X}{W}\right) < -1$$

becomes

$$X > W, \quad X < -W, \quad W > 0$$

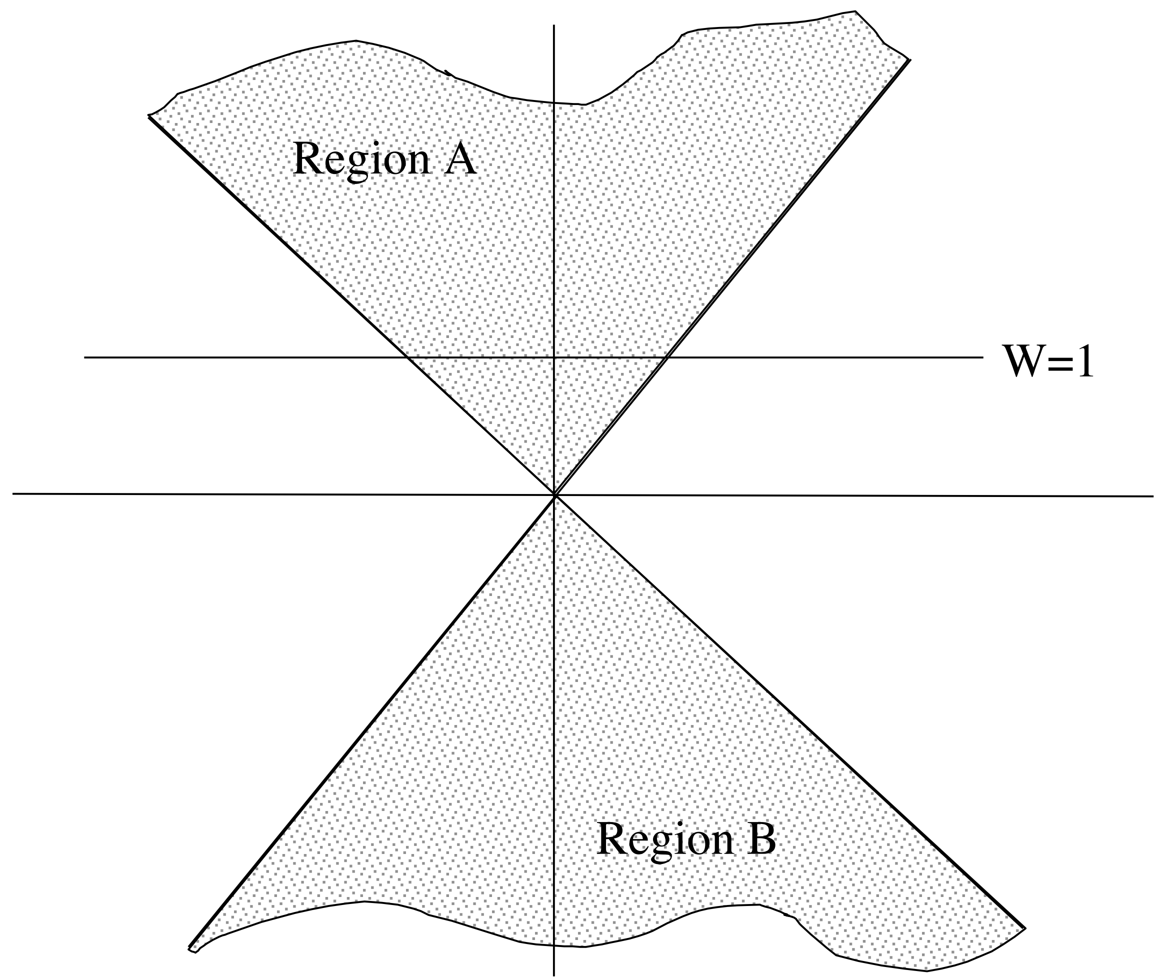
AND

$$X < W, \quad X > -W, \quad W < 0$$

(So far W is positive, but negatives occur if we further overload the use of h.c.'s)

Clipping in homogeneous coord.'s

The clipping volume in cross section



Clipping in homogeneous coord.'s

- If we know that W is positive (the case so far!), simply clip against region A
- If we are using the h.c. for additional deferred division, then W can be negative.
- If W is negative, then we use region B. The clipping can be done by negating the point, and clipping against A, due to the nature of A and B.
- Case where object has both positive and negative W is a little more complex.
- Notice that the actual clipping computations are not that different from the case in Plan A---no free lunch!