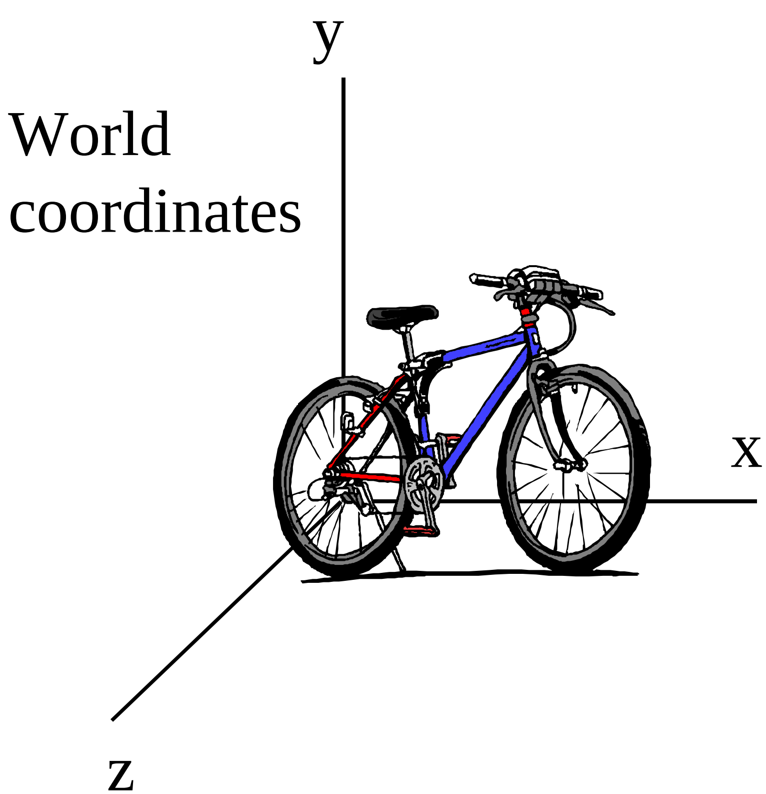
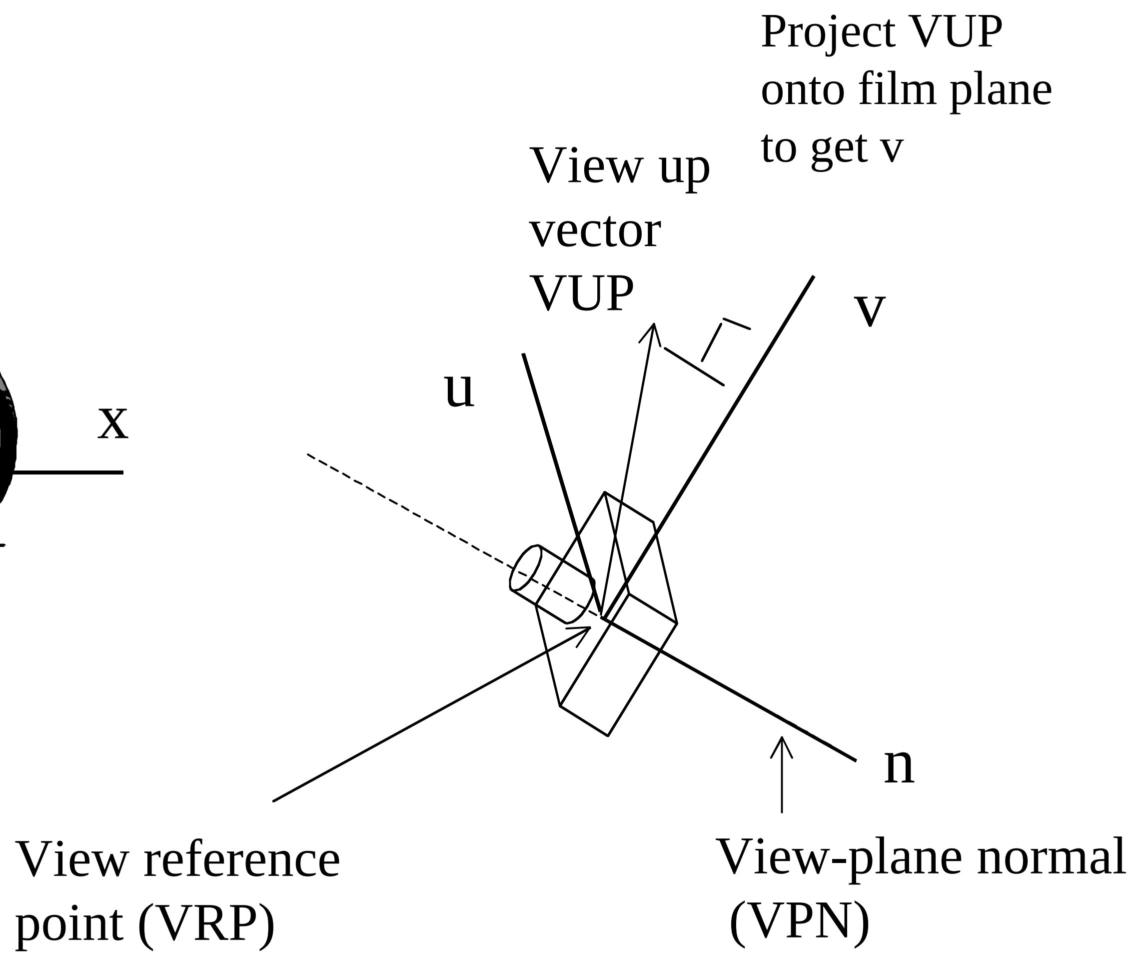
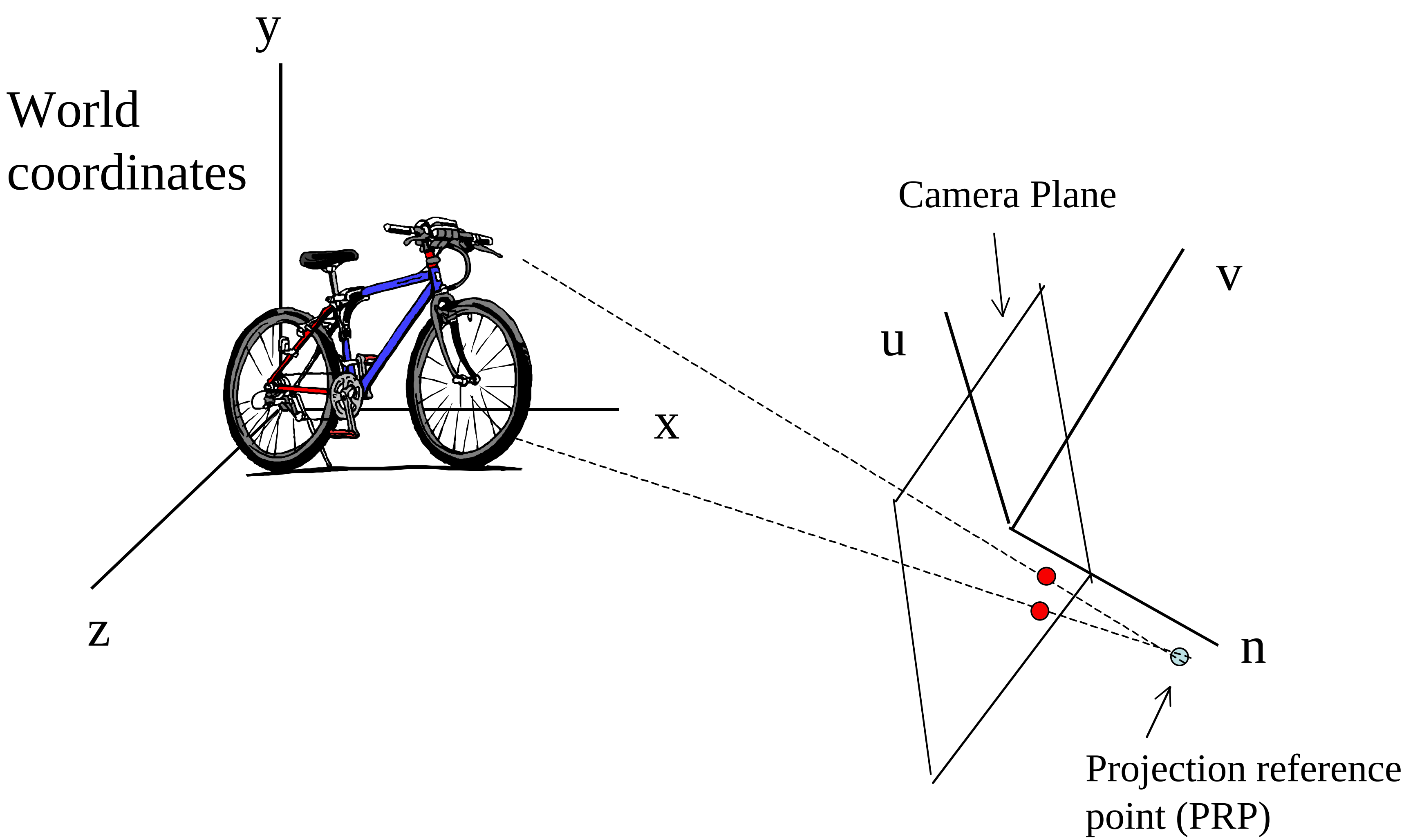
Specifying a camera

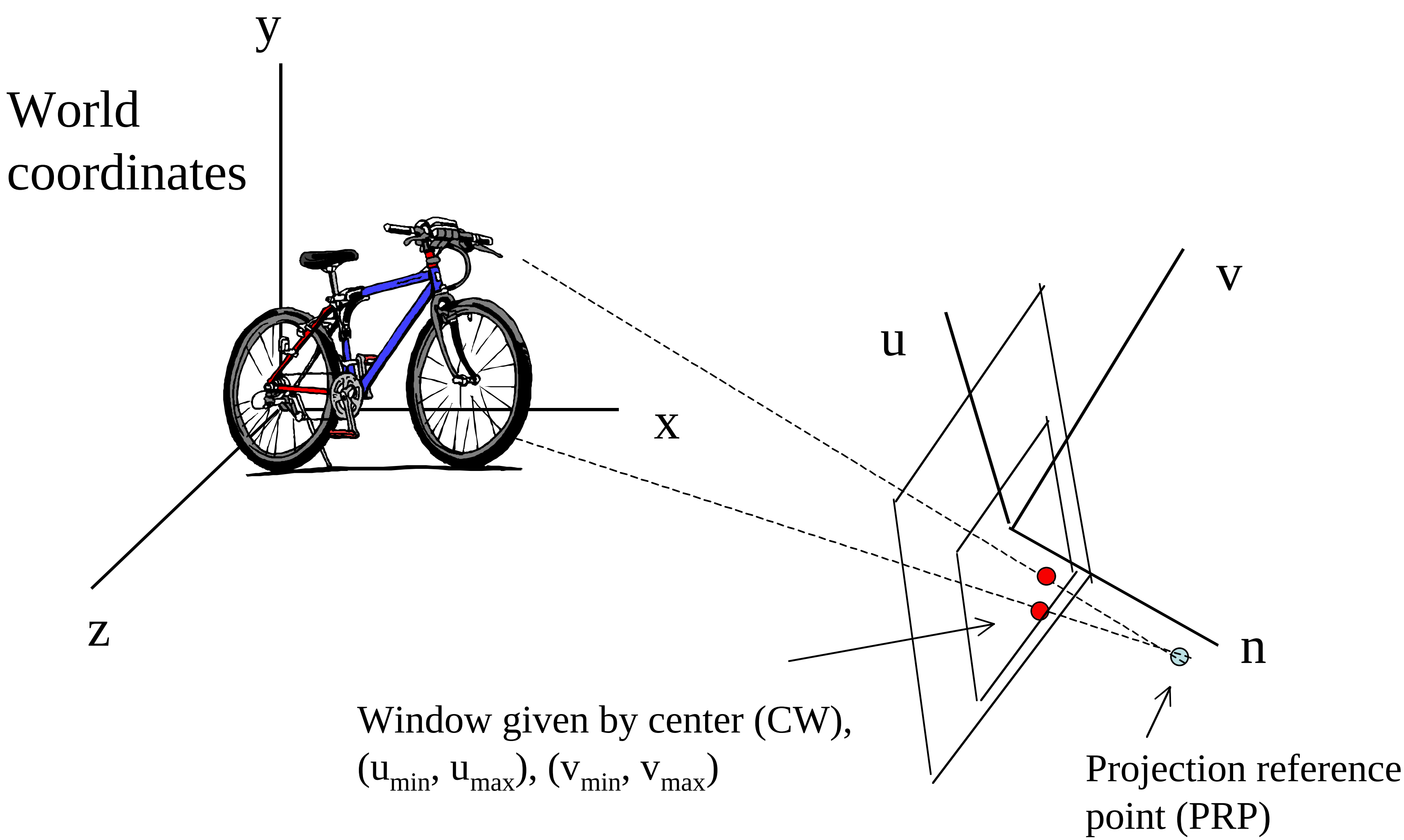




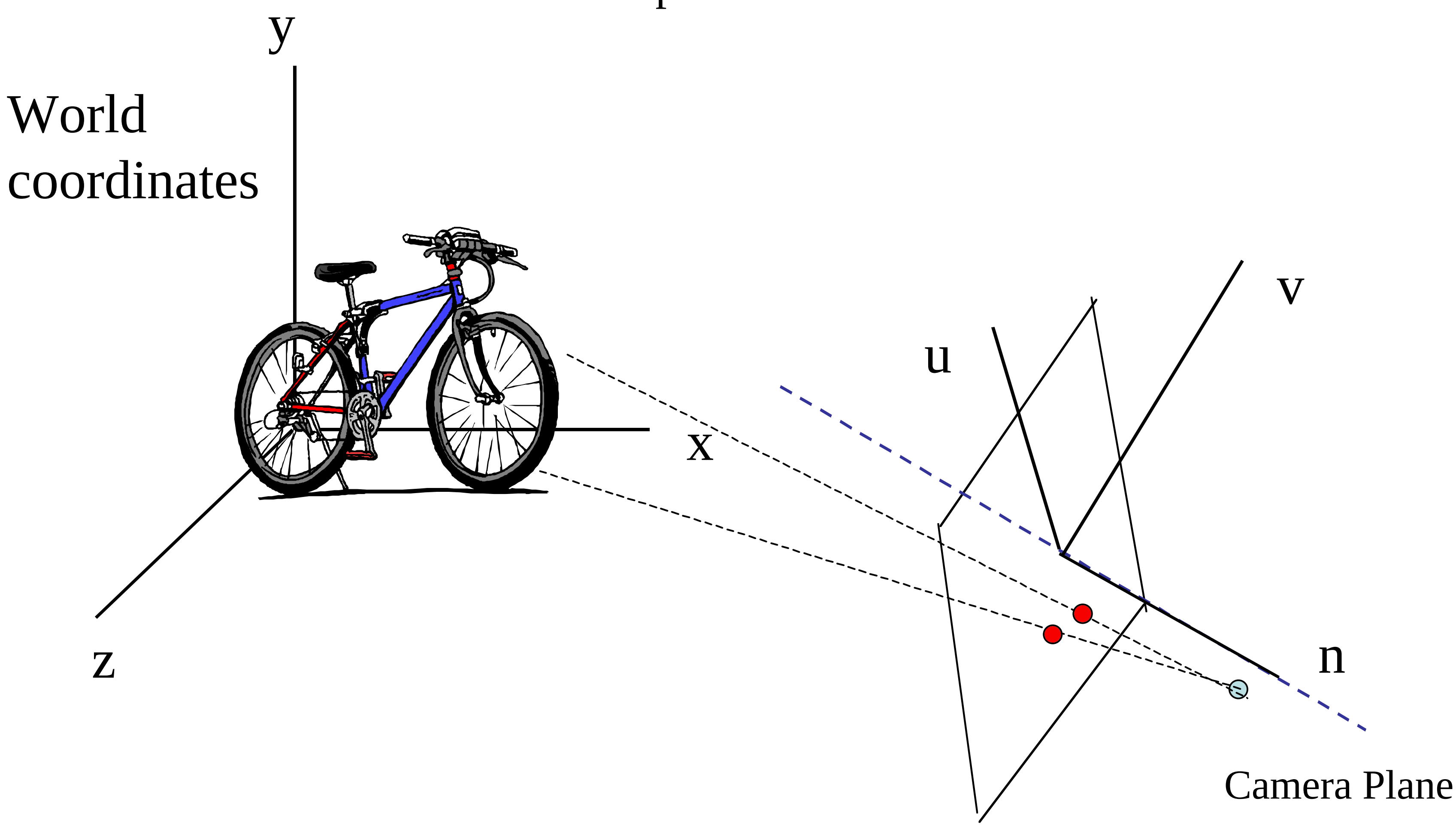
Specifying a camera



Specifying a camera

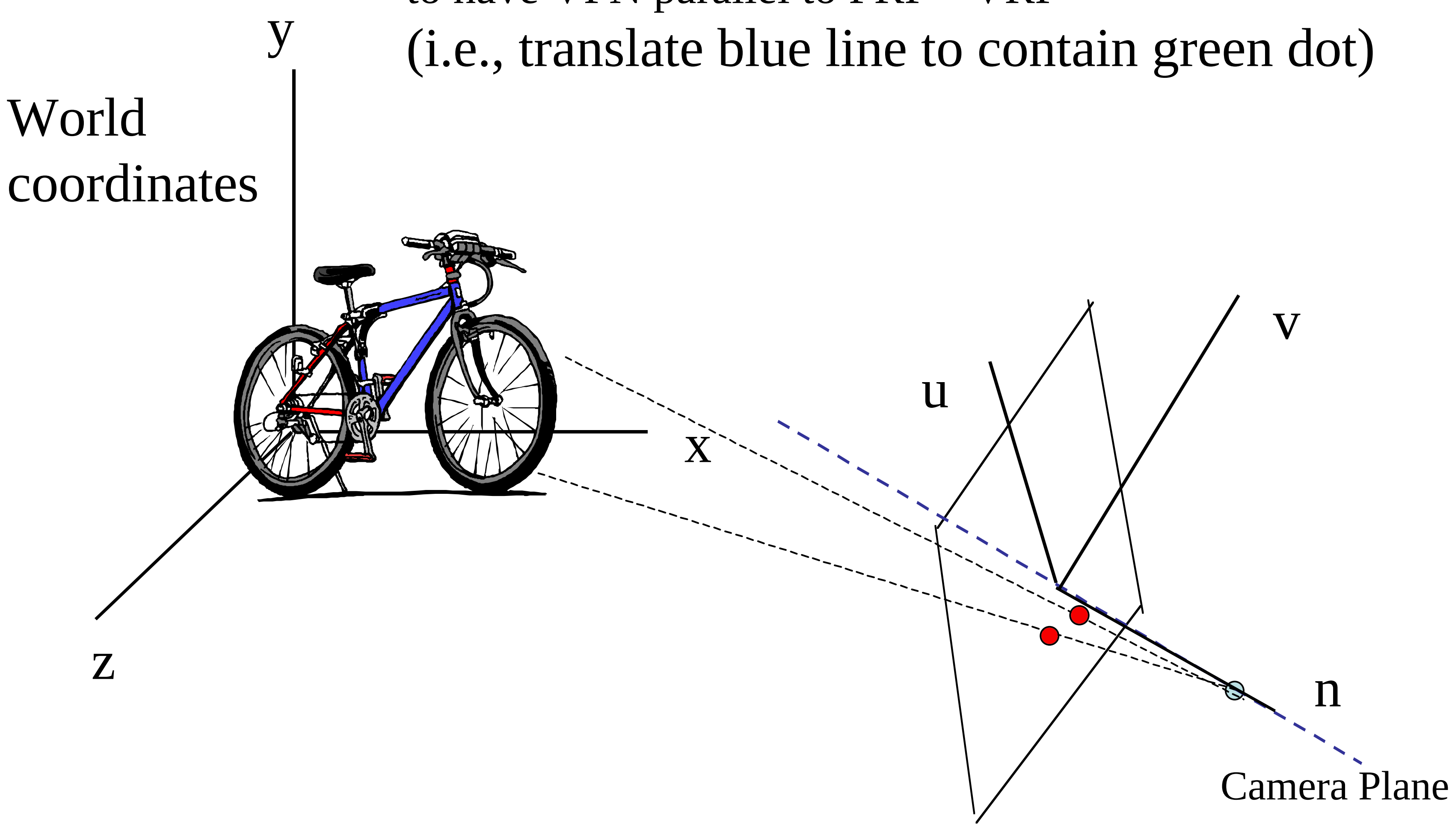


If PRP, VUP, and VPN are fixed in world coordinates, but VRP moves on the film plane, this is the same as a translation in the image plane - because this is usually absorbed by the 2D window-viewport transform, it is usual to have VPN parallel to PRP -- VRP



If PRP, VUP, and VPN are fixed in world coordinates, but VRP moves on the film plane, this is the same as a translation in the image plane - because this is usually absorbed by the 2D window-viewport transform, it is usual to have VPN parallel to PRP --VRP

(i.e., translate blue line to contain green dot)

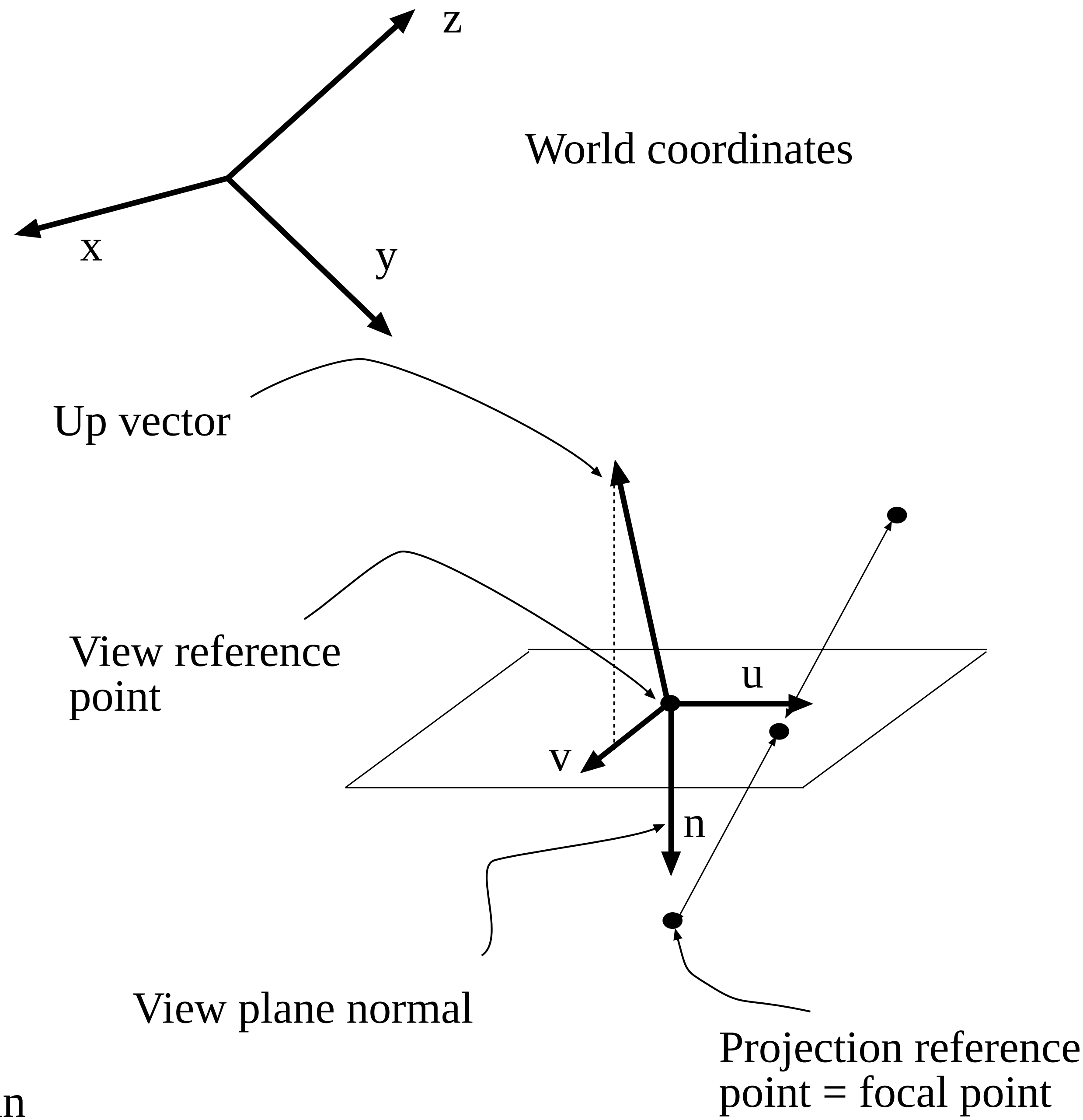


View reference point and view plane normal specify film plane.

Up vector gives an “up” direction in the film plane. vector v is projection of up vector into film plane ($= \mathbf{n} \times \mathrm{VUP} \times \mathbf{n}$).

u is chosen so that (u, v, n) is a right handed coordinate system; i.e. it is possible to rotate (x->u, y->v, z->n) (and we’ll do this shortly).

VRP, VPN, VUP must be in world coords; PRP could be in world coords or in camera coords

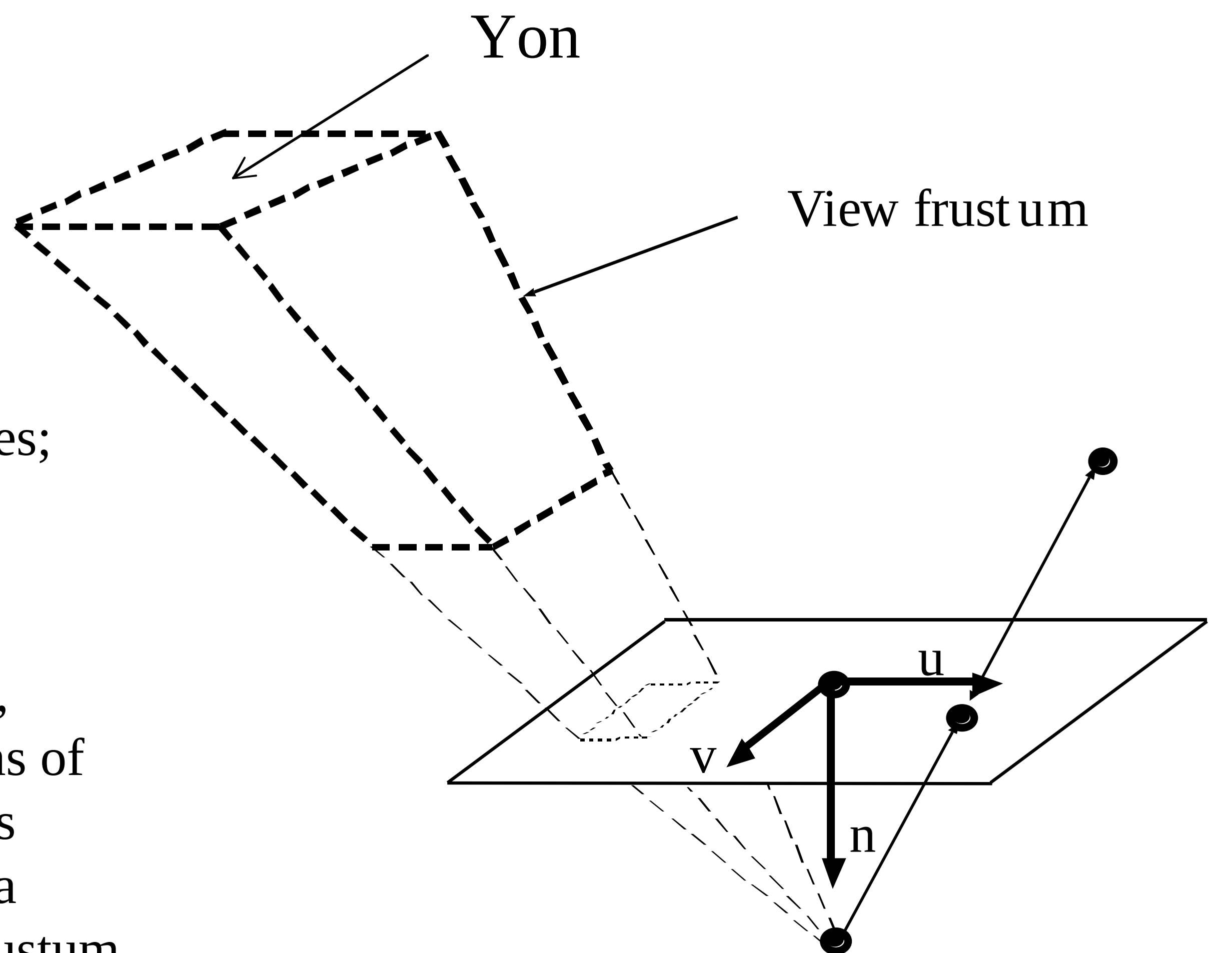


U, V can be used to specify a window in the film plane; only this section of film ends up on the screen.

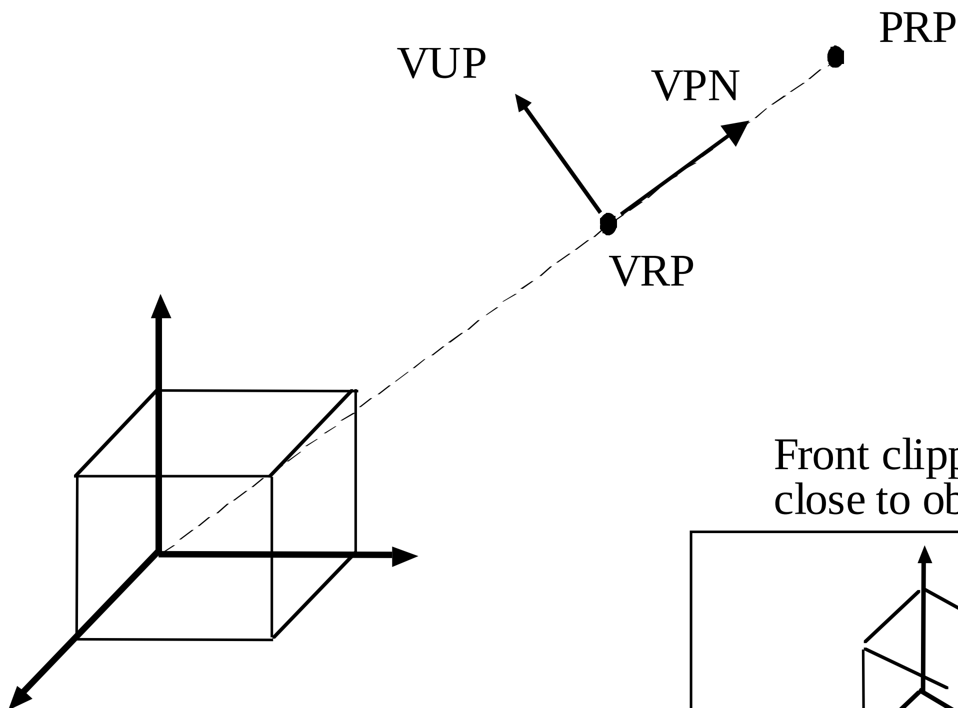
This window defines four planes; points outside these planes are not rendered.

Hither and yon clipping planes, which are always given in terms of camera coordinates, and always parallel to the film plane, give a volume - known as the view frustum.

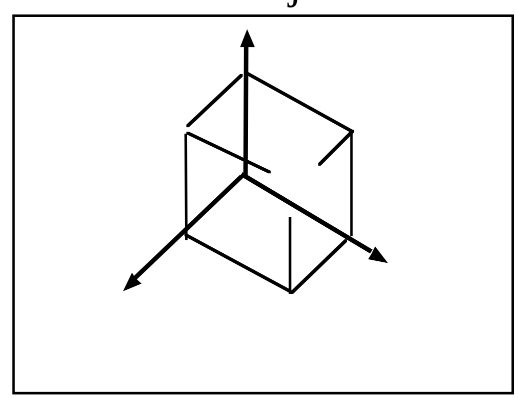
Orthographic case: - view frustum is cuboid (i.e. all angles right angles, but edges not necessarily of equal length).



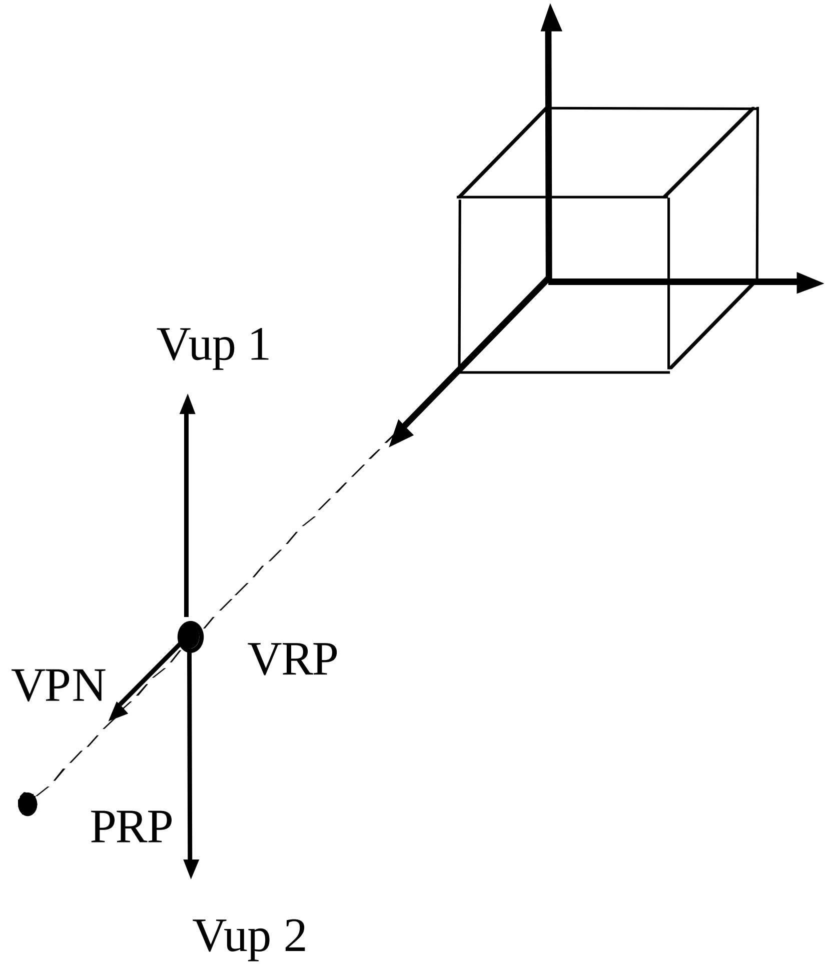
Hither and yon (front and back) clipping planes example--
hither too close - cuts off corner

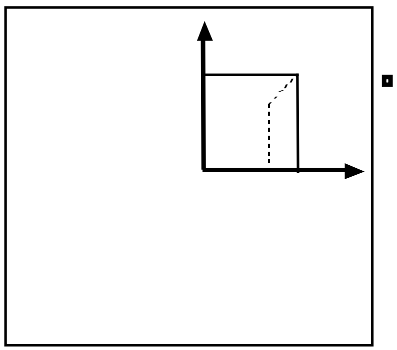


Front clipping plane
close to object

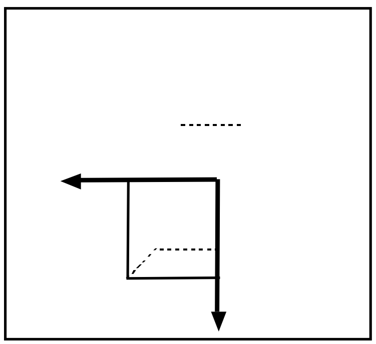


Changing the up vector is equivalent to a rotation in the film plane - think of rotating a camera in your hand.

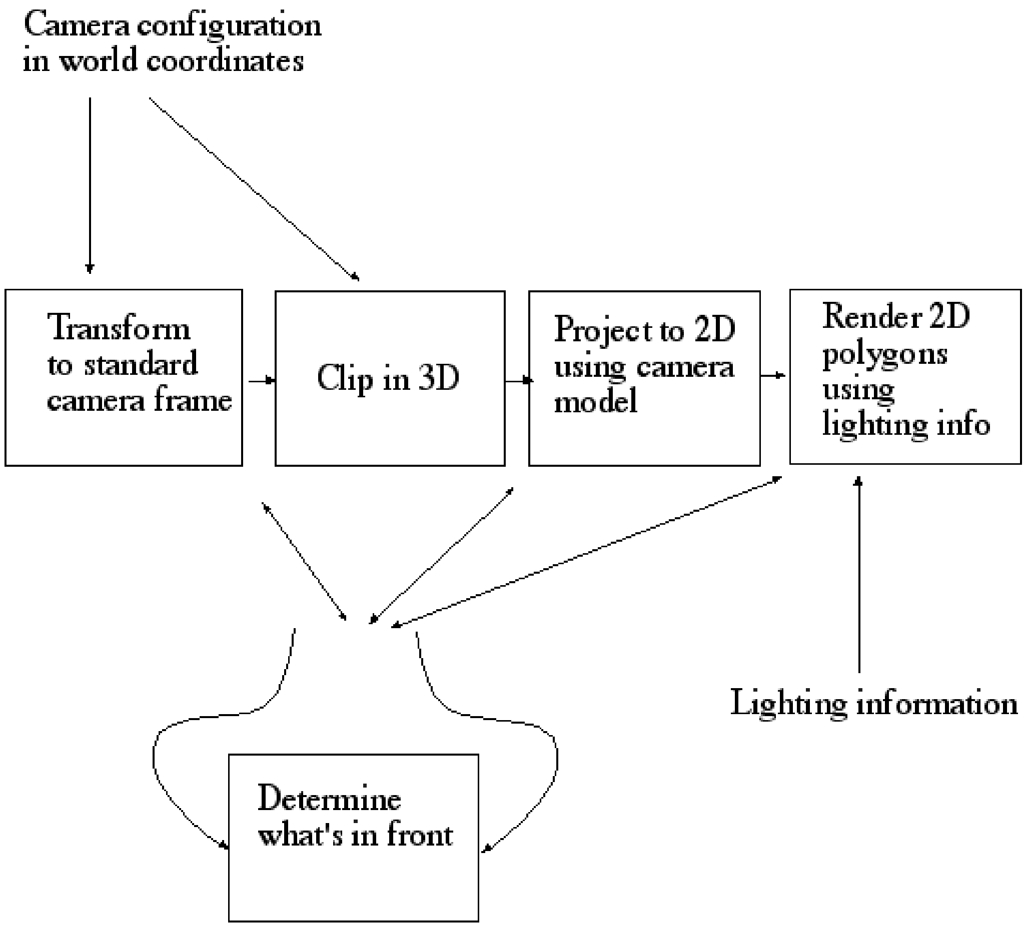


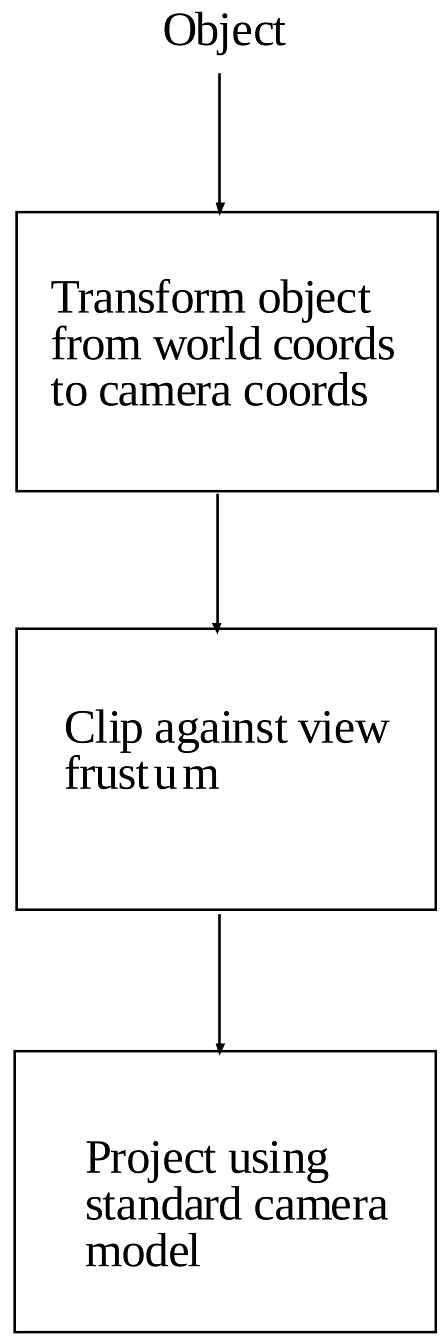


Vup 1

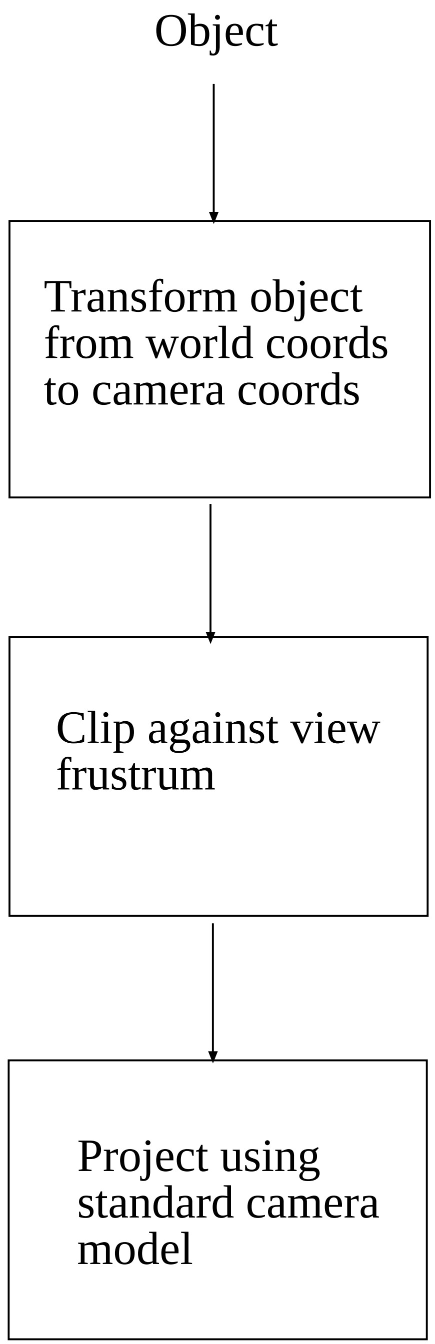


Vup 2

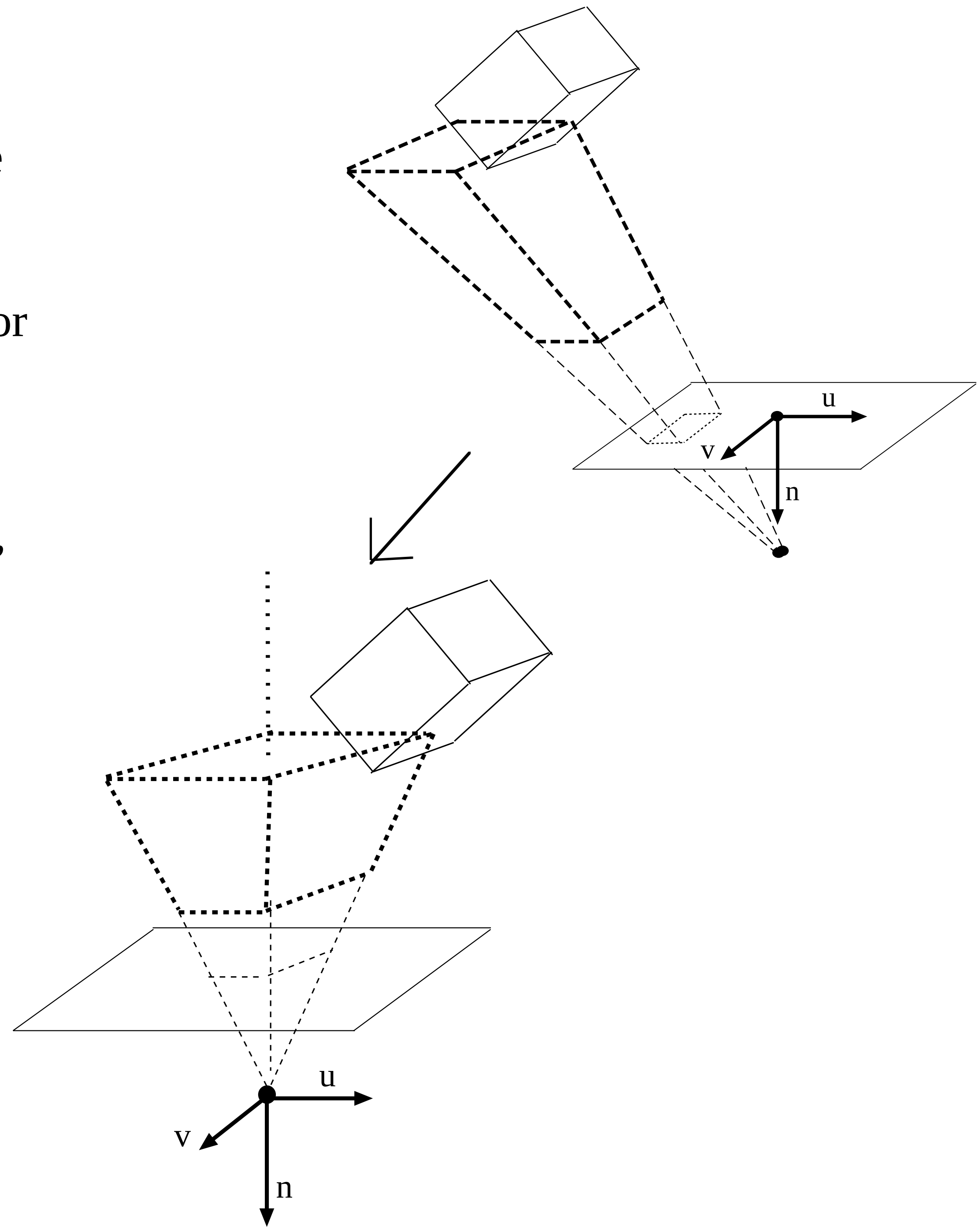

Camera configuration
in world coordinates
Transform
to standard
camera frame
Clip in 3D
Project to 2D
using camera
model
Render 2D
polygons
using
lighting info
Determine
what's in front
Lighting information


Object
Transform object
from world coords
to camera coords
Clip against view
frustum
Project using
standard camera
model

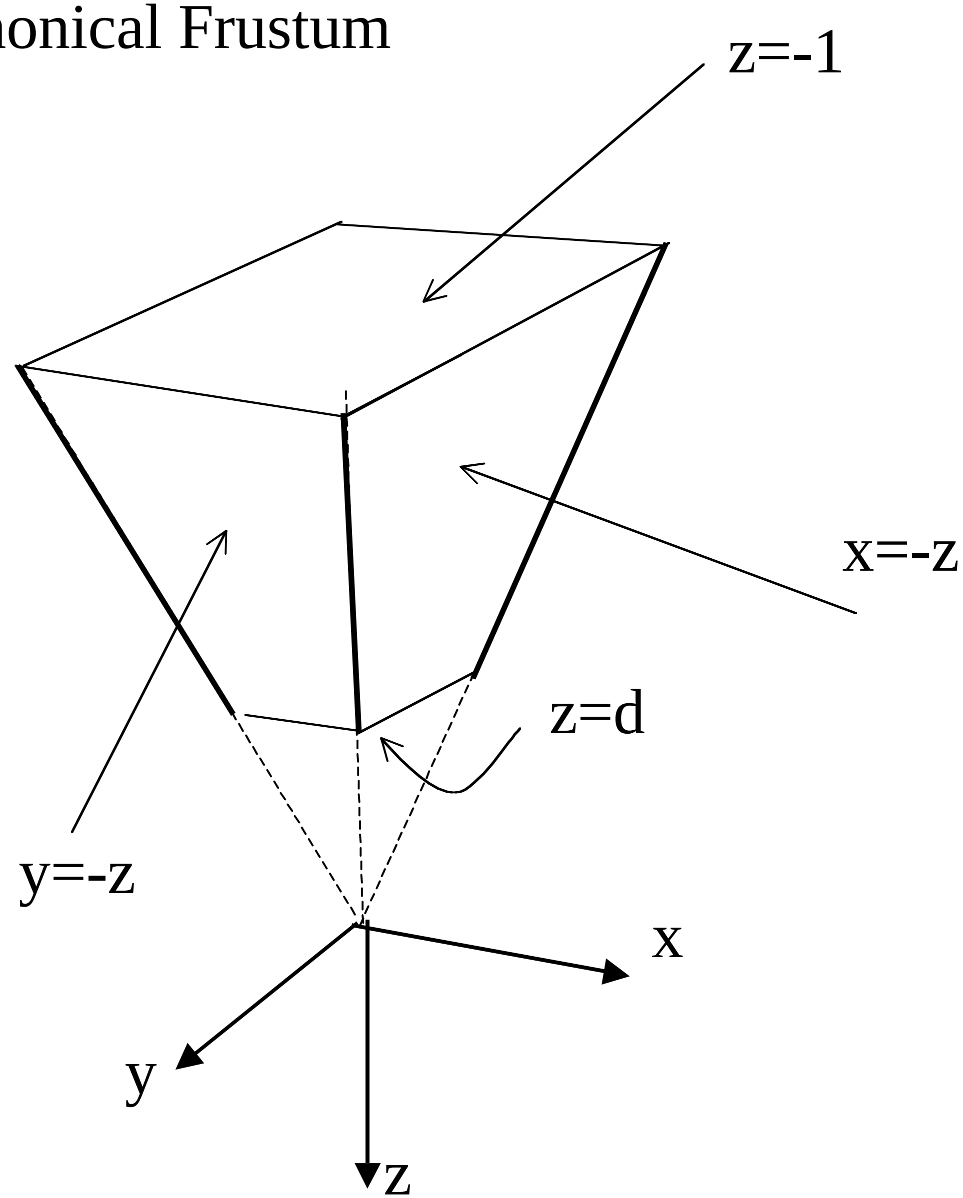
- Advantages of clipping against view frustrum:
  - Don’t project objects that aren’t drawn
  - cf. clip against hither/yon, project, clip against window in film plane
  - hence slightly less work.
- Advantage of clipping in camera frame (rather than in world frame):
  - Better supports transform to standard view frustrum, where clipping is easiest.
- Advantage of transforming to camera frame:
  - Easiest to compute the effects of the camera in this frame.

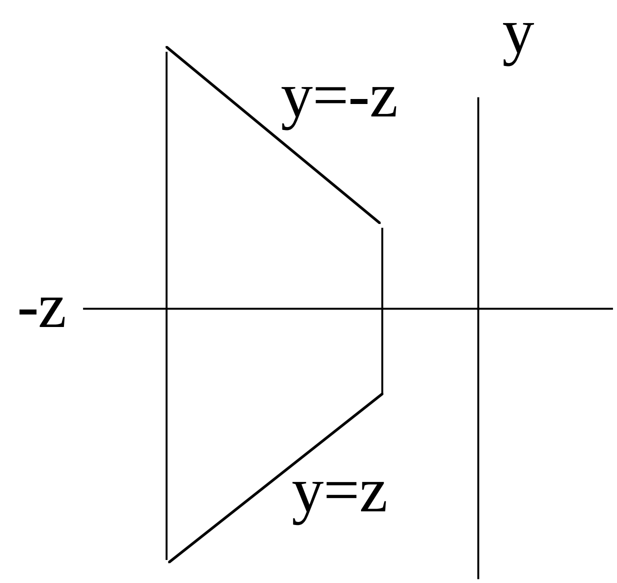


- If we clip against the frustum blindly, clipping is hard - this is because planes bounding the frustum have a complex form
- Thus, to test in/out, must test the sign of a x + b y + c z +d for some a, b, c, d - much worse than a simple compare.
- Solution: transform view frustrum into a canonical form, where clip planes have easy form - e.g. z=x, z=-x, z=y, z=-y, z=-1, z=d



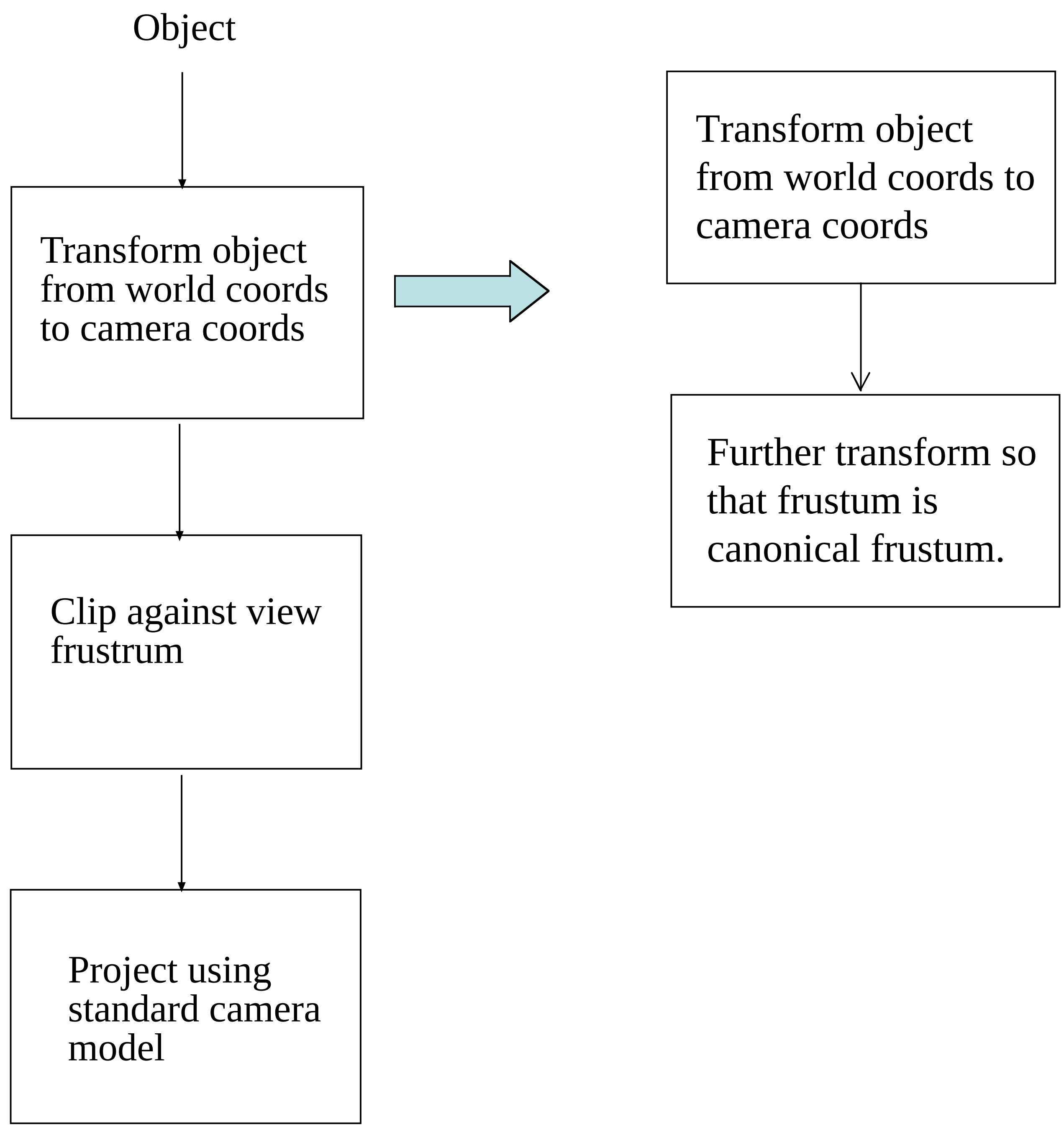
Canonical Frustum





If image plane transforms to z=m then in new frame, projection is easy:

(x, y, z) -> (m x /z, m y/z)


Object
Transform object from world coords to camera coords
Clip against view frustrum
Project using standard camera model
Transform object from world coords to camera coords
Further transform so that frustum is canonical frustum.

Transform object
from world coords to
camera coords

Step 1. Translate VRP to world origin. Call this T_1. T_1 maps world points (note opposite transformations for object and coordinate frame).

Transform object from world coords to camera coords

Step 2. Rotate camera coordinate frame so that u is x, v is y, and n is z. The matrix is ?

Transform object
from world coords to
camera coords

Step 2. Rotate camera coordinate frame so that u is x, v is y, and n is z. The matrix is:

$$\begin{vmatrix} & u^T & & 0 \\ & v^T & & 0 \\ & n^T & & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

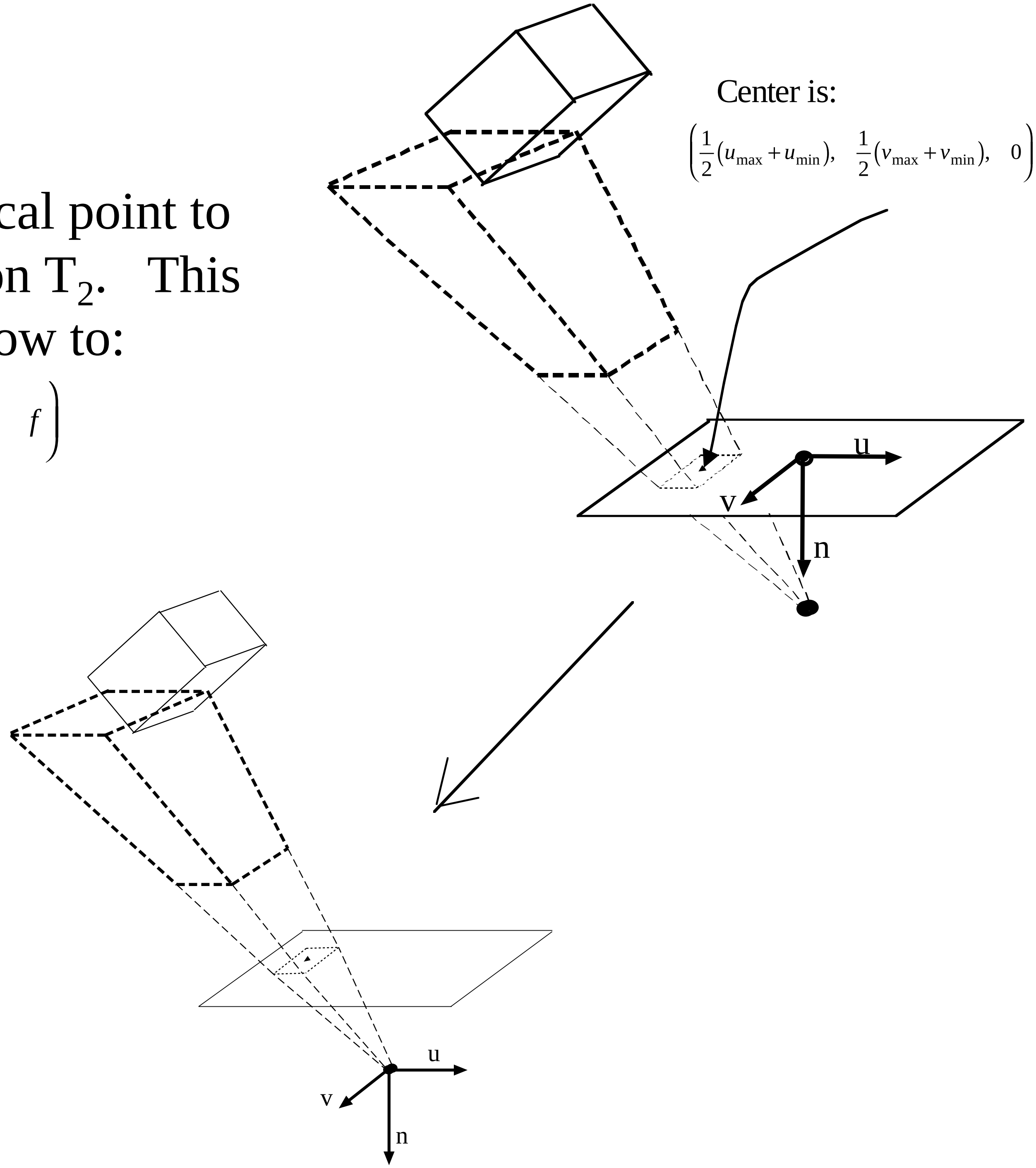
(why?)

Further transform so that frustum is canonical frustum.

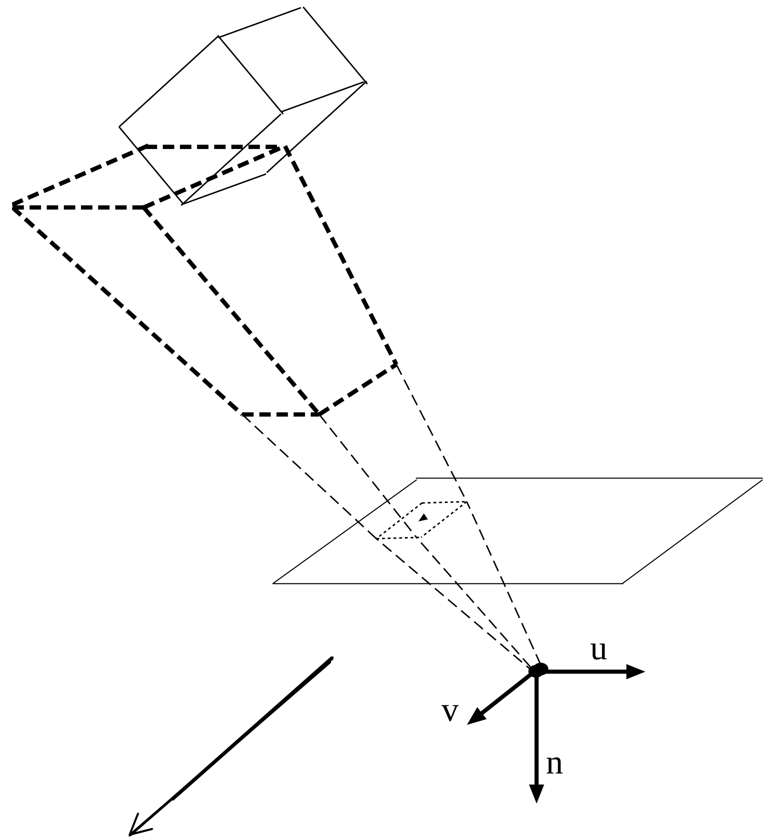
1. Translate focal point to origin
2. Shear so that central axis of frustum lies along the n axis
3. Scale x, y so that faces of frustum lie on planes
4. Isotropic scale so that back clipping plane lies at z=-1

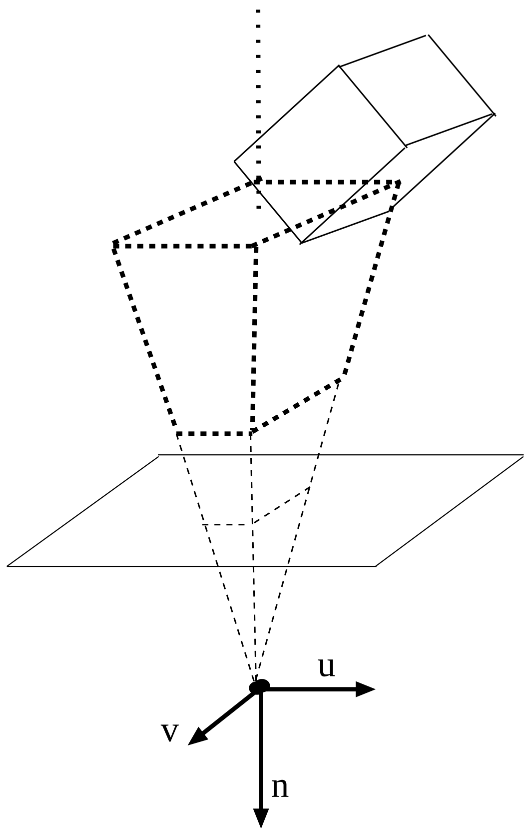
Step 1: Translate focal point to origin; call translation T_2. This takes center of window to:

$$\left(\frac{1}{2}(u_{max}+u_{min}),\quad \frac{1}{2}(v_{max}+v_{min}),\quad f\right)$$



Step 2: Shear this volume so that the central axis lies on the n-axis. This is a shear, because rectangles on planes n=constant must stay rectangles. Call this shear S_1





Shear S_1 takes previous window midpoint $\left(\frac{1}{2}(u_{\max}+u_{\min}),\ \ \frac{1}{2}(v_{\max}+v_{\min}),\ \ f\right)$ to (0, 0, f) - this means that matrix is

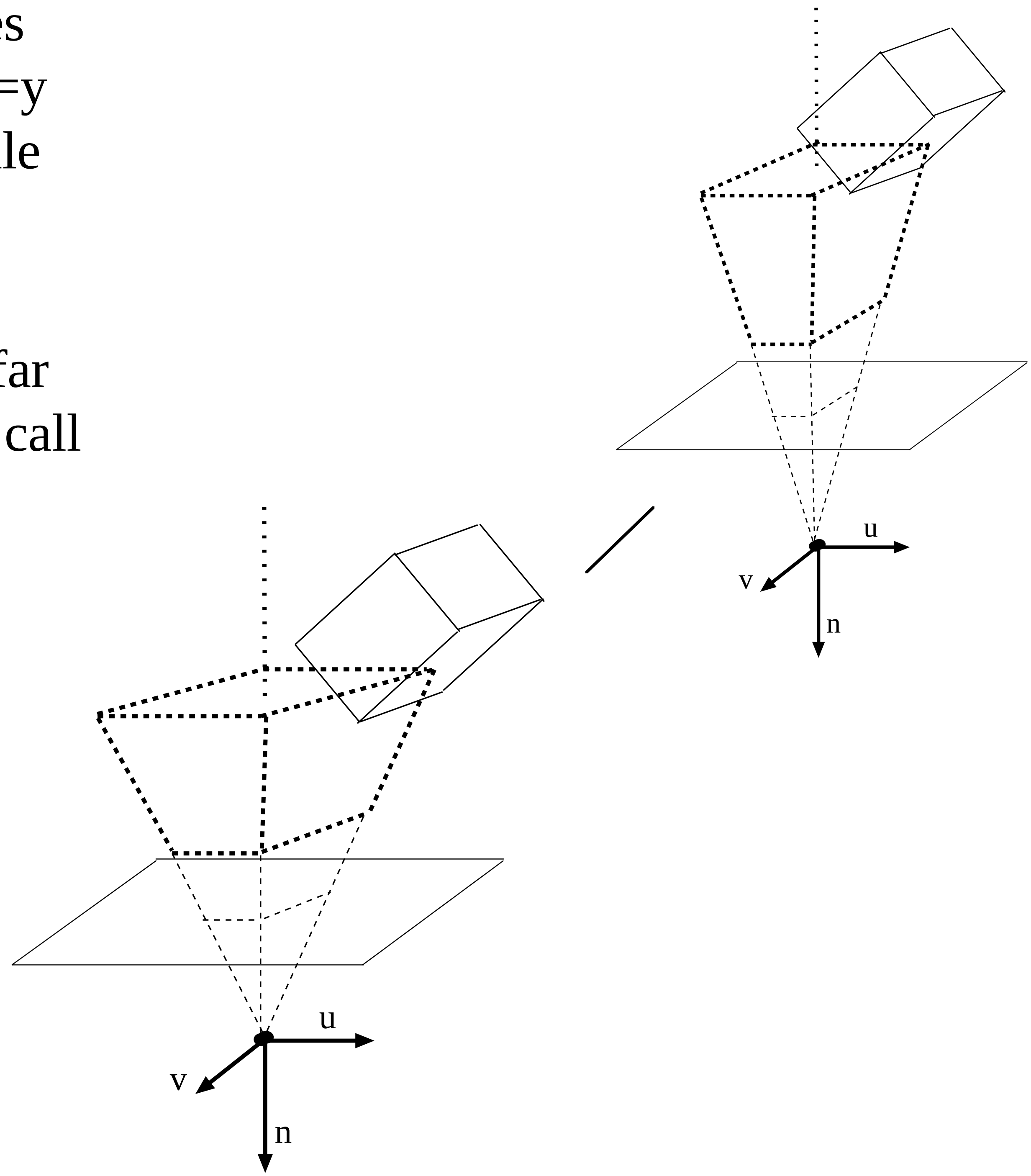
?

Shear S_1 takes previous window midpoint $\left(\frac{1}{2}(u_{\max}+u_{\min}),\ \ \frac{1}{2}(v_{\max}+v_{\min}),\ \ f\right)$ to (0, 0, f) - this means that matrix is:

$$\begin{pmatrix} 1 & 0 & -\frac{(u_{\min}+u_{\max})}{2f} & 0 \\ 0 & 1 & -\frac{(v_{\min}+v_{\max})}{2f} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

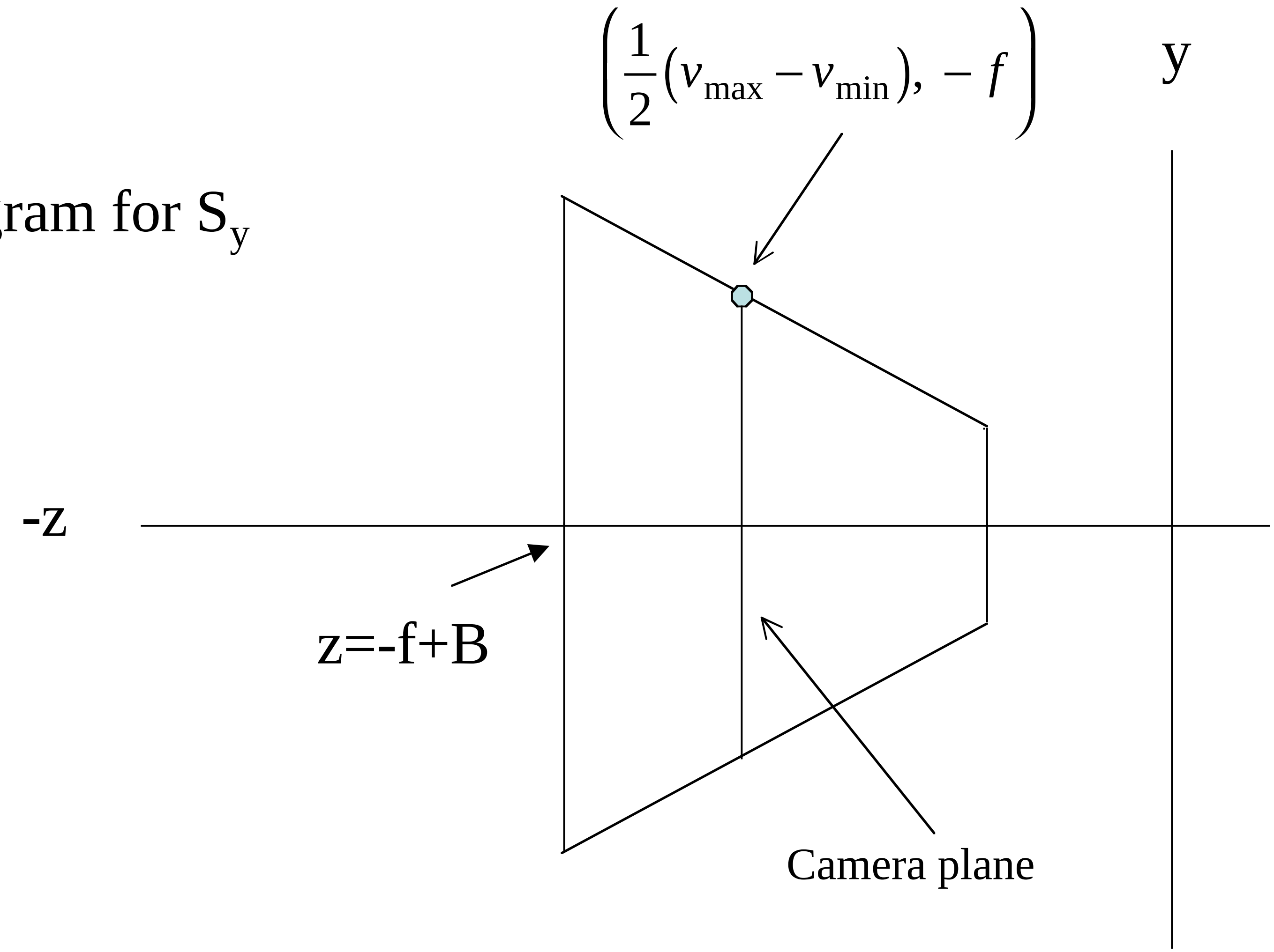
4. Scale x, y so that planes are on z=x, z=-x and z=y and z=-y. Call this scale Sc_1

5. Isotropic scale so that far clipping plane is z=-1; call this scale Sc_2



4. Scale x, y so that planes are on $z=x$, $z=-x$ and $z=y$ and $z=-y$. Call this scale Sc_1

Diagram for S_y



4. Scale x, y so that planes are on z=x, z=-x and z=y and z=-y. Call this scale Sc_1

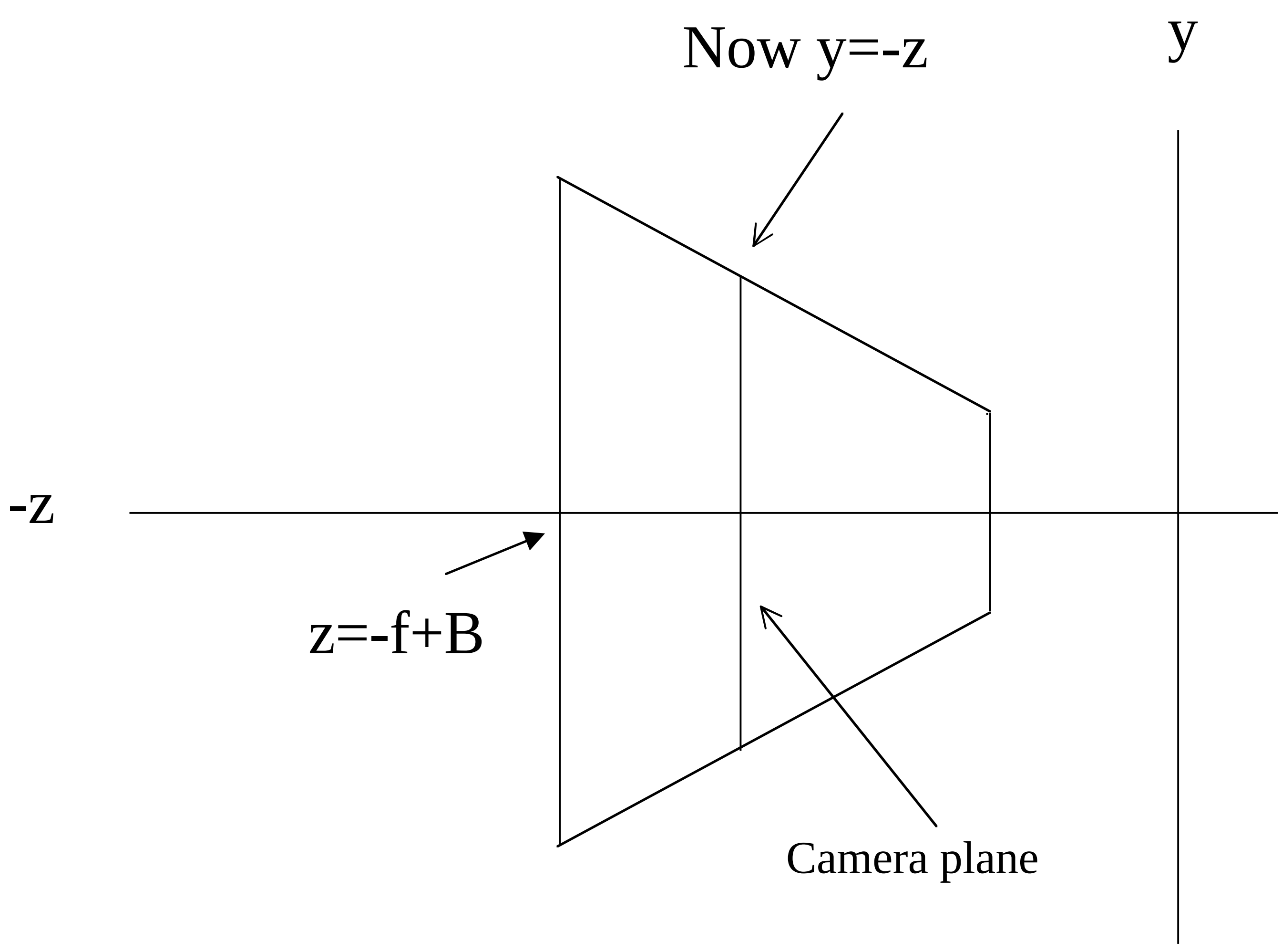
$$\left(\frac{1}{2}(v_{\max} - v_{\min}),\ -f\right) \Rightarrow \quad y=-z$$

$$k\frac{1}{2}(v_{\max} - v_{\min}) = f$$

$$k = \frac{2f}{(v_{\max} - v_{\min})}$$

(k is y scale factor)

5. Now isotropic scale so that far clipping plane is z=-1; call this scale Sc_2



5. Now isotropic scale so that far clipping plane is $z=-1$; call this scale Sc_2

Currently, at far clipping plane, $z=-f+B$

Want a factor k so that $k(-f+B)=-1$

So, $k = -1/(-f+B)$

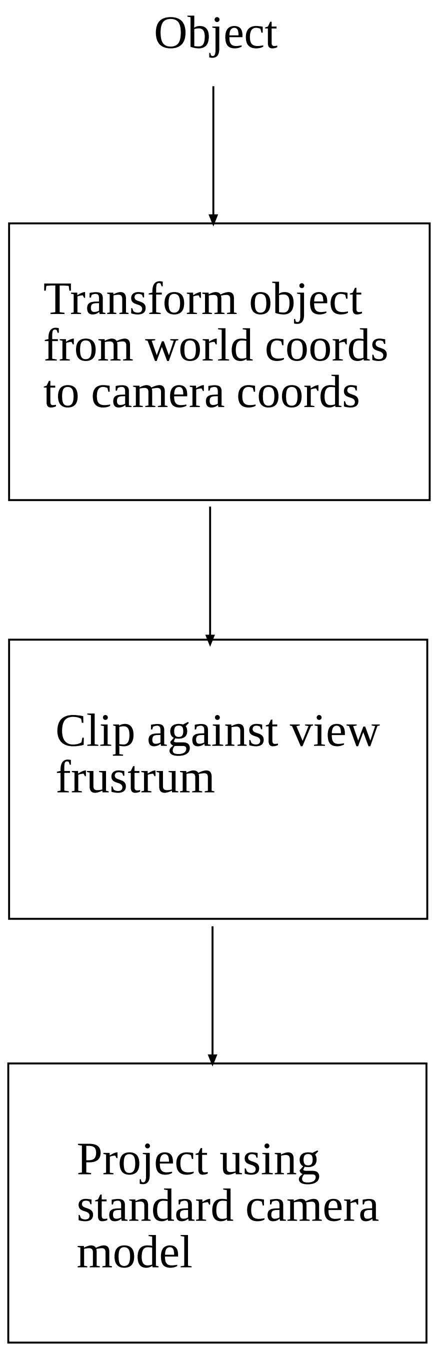
(Note that B is negative, and k is positive)

3D Viewing Pipeline

(Point in canonical camera coordinates)

$$Sc_2 Sc_1 S_1 T_2 R_1 T_1$$

(Point in world coordinates)



Plan A: Clip against canonical frustum (relatively easy--we chose the canonical frustum so that it would be easy)

Plan B: Be even more clever. Further transform to cube and clip in homogenous coordinates.

Clipping against the canonical frustum

2D algorithms are easily extended. For example, for Cohen Sutherland we use the following 6 out codes:

$y>-z \quad y<z \quad x>-z \quad x<z \quad z<-1 \quad z>z_{min}$

($z_{min} = (f-F)/(B-f)$)

Intersection of lines with the planes is simpler than the general case

More efficient algorithms are available.