Computing \((u,v,n)\) in world coordinates

\[
\begin{align*}
\mathbf{u} & \parallel \mathbf{VUP} \times \mathbf{n} \\
\mathbf{u} & = \frac{\mathbf{VUP} \times \mathbf{n}}{\|\mathbf{VUP} \times \mathbf{n}\|} \\
\mathbf{v} & = \mathbf{n} \times \mathbf{u}
\end{align*}
\]

- VRP, VPN, VUP must be in world coords;
- PRP (focal point) could be in world coords, but more commonly, camera coords (which are the same scale as world coords);
- We will use camera coords, and further assume that PRP = (0,0,f).

Specifying a camera

Object in world coordinates (after modeling transforms)

- Transform object from world coordinates to standard camera coordinates
- Clip against canonical view frustum
- Project using standard camera model

Transform object from world coords to camera coords

Further transform so that frustum is canonical frustum.

Window given by \((u_{\text{min}}, u_{\text{max}}), (v_{\text{min}}, v_{\text{max}})\), denote center by \(CW\)

Project reference point (PRP)—assume \(k\) is on \(\mathbf{n}\)

All in H&B chapter 7 "3D viewing"
\( u \) and \( v \) can be used to specify a window in the image plane; only this section of image plane 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).

We will first map world coordinates to the camera coordinates (top figure)

However, if clipping against the frustum is difficult because planes bounding the frustum have a complex form

**Solution:** further transform frustum to a canonical form where clip planes are an easy form to deal with:

\[
\begin{align*}
z &= x, \\
z &= -x \\
z &= y, \\
z &= -y \\
z &= -1, \\
z &= zd
\end{align*}
\]

Transform object from world coords to camera coords

Step 1. Translate the camera at VRP to the world origin. Call this \( T_1 \).

Translation vector is simply negative VRP.

(We are changing the coordinate system of the world, which is the same thing mathematically as moving the camera. We want object world coordinates to change so that the camera location (VRP) becomes the origin).
Transform object from world coords to camera coords

Step 2. Rotate camera coordinate frame (in w.c.) so that so that \( u \) is \( x \), \( v \) is \( y \), and \( n \) is \( z \). The matrix is ?

(We are changing the coordinate system of the world, which is the same thing mathematically as moving the camera. We want object world coordinates to change so that the camera axis becomes the standard axis—e.g., \( u \) becomes \((1,0,0,0)\), \( v \) becomes \((0,1,0,0)\) and \( n \) becomes \((0,0,1,0)\)).

\[
\begin{pmatrix}
\mathbf{u}^T \\
\mathbf{v}^T \\
\mathbf{n}^T \\
0 0 0 1
\end{pmatrix}
\]

In the current coords (world shifted so that VPR is at origin): \( u \) maps into the X-axis unit vector \((1,0,0,0)\) which is what we want.

(Similarly, \( v \rightarrow Y-axis \) unit vector, \( n \rightarrow Z-axis \) unit vector)

Transform object from world coords to camera coords

Object in world coordinates (after modeling transforms)

Transform object from world coordinates to standard camera coordinates

Clip against canonical view frustum

Project using standard camera model

Further transform so that frustum is canonical frustum.

Transform object from world coords to camera coords

Step 2. Rotate camera coordinate frame (in w.c.) so that so that \( u \) is \( x \), \( v \) is \( y \), and \( n \) is \( z \). The matrix is:

\[
\begin{pmatrix}
\mathbf{u}^T & 0 \\
\mathbf{v}^T & 0 \\
\mathbf{n}^T & 0 \\
0 0 0 1
\end{pmatrix}
\]

(why?)
Mapping the view frustum to the canonical view frustum

Further transform so that frustum is canonical frustum.

Since we are now in camera coordinates, we will often refer to them as (x,y,z) not (u,v,n).

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

Step 1: Translate focal point (PRP) to origin; call this translation T. Since we have PRP in camera coordinates (where we now are), the translation vector is simply negative PRP. In particular, in the very common case where PRP is (0,0,f), the translation vector is (0,0,-f)

Window center is:
\[
\left(\frac{u_{\text{max}} + u_{\text{min}}}{2}, \frac{v_{\text{max}} + v_{\text{min}}}{2}, \frac{f}{2}\right)
\]

Window center is now:
\[
\left(\frac{u_{\text{max}} + u_{\text{min}}}{2}, \frac{v_{\text{max}} + v_{\text{min}}}{2}, -f\right)
\]

Step 1 is relatively straightforward, but notice that the location of the clipping planes also gets shifted.

So, before we had the back clipping plane at B (which is negative). Now it is at: B-f.
Step 2: Shear this volume so that the central axis lies on the z-axis. This is a shear, because rectangles on planes z=constant must stay rectangles. Call this shear $S_1$.

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

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

Note that the size of a rectangle in the image plane does not change.

3. Scale x, y so that planes are on $z=x$, $z=-x$ and $z=y$ and $z=-y$. Call this scale $S_{c1}$.

4. Isotropic scale so that far clipping plane is $z=-1$; call this scale $S_{c2}$.

Diagram for $S_y$ (ignoring x coordinate)
4. Scale x, y so that planes are on z=x, z=x and y=z and z=-y. Call this scale Sc

\[
\left(\frac{1}{2}(v_{\text{max}} - v_{\text{min}}), -f\right) \rightarrow (f, -f)
\]
(because y=z)

\[
k_y \frac{1}{2}(v_{\text{max}} - v_{\text{min}}) = f
\]

\[
k_y = \frac{2f}{v_{\text{max}} - v_{\text{min}}}
\]  
\(k_y\) is y scale factor

\[
\begin{bmatrix}
\frac{2f}{v_{\text{max}} - v_{\text{min}}} & 0 & 0 & 0 \\
\frac{u_{\text{max}} - u_{\text{min}}}{v_{\text{max}} - v_{\text{min}}} & 2f & 0 & 0 \\
0 & 0 & \frac{v_{\text{max}} - v_{\text{min}}}{v_{\text{max}} - v_{\text{min}}} & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

5. Now isotropic scale so that far clipping plane is z=-1; call this scale Sc2

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

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

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

(Note that B is negative, and k is positive)
\[ \mathbf{S}_{c_2} = \begin{bmatrix} \frac{1}{f-B} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f-B} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

Note that the focal length, \( f \), also gets transformed (needed for the perspective transformation coming up).

It is:

\[ f' = \frac{f}{f - B} \]

3D Viewing Pipeline

Point in canonical camera coordinates \[ S_{c_2} S_{c_1} S_{1} T_{2} R_{1} T_{1} \] Point in world coordinates

Then clip (more later), and then project