A SPATIAL RELATIONSHIP

A.1 The Camera Model

The camera model can be defined by three points.

- the camera point, *cam*
- the camera vector origin, *cvo*
- the left vector origin, *lvo*

These points define two vectors, the camera vector and left frame vector, which describe the camera pose in world space. To initialize the camera model we choose values that place the camera vector centered between the left and right head glasses markers and place the *lvo* to match the scene cameras known field of view.

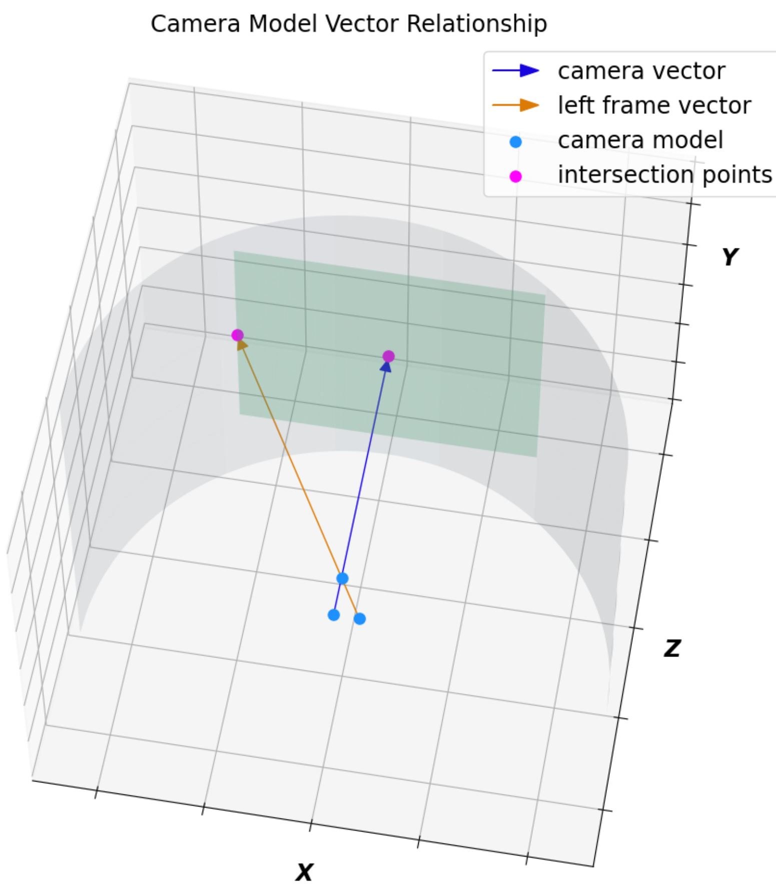

Fig. 1. Description of the camera model in world space, not to scale. The gray cylindrical surface is the screen onto which virtual reality objects are projected. The green rectangular plane represents the frame as captured by the scene camera. The three cyan points are the camera model which describe the camera position and orientation in world space and define the two vectors which intersect(magenta) with the scene frame.

A.2 Definition of a Ray

A ray is a vector that extends infinitely in one direction. Any point, $P(t)$, on the ray is defined by an origin point, o, and a direction vector, $\vec{d}$, multiplied by a scalar, t.

$$P(t) = o + t * \vec{d}, \quad t \geq 0 \tag{1}$$

or

$$P(t) = \begin{cases} x_o + t * x_d, \\ y_o + t * y_d, & t \geq 0 \\ z_o + t * z_d, \end{cases} \tag{2}$$

A.3 Intersection of the Camera Vector and Screen

The intersection of the camera vector and the screen define the frame center capture by the scene camera and is calculated by intersecting the ray with a cylinder. The radius, r, of our screen is 2.49 meters. This is represented as a cylinder whose axis is parallel to the y-axis and centered in world space at $(x, z) = (0, 0)$.
A point is on the cylinder surface when:

$$x^2 + z^2 = r^2 = 2.49^2 \tag{3}$$

Using the definition of a ray and our screen (a cylinder) we can calculate the intersection of the camera ray.

$$(x_o + tx_d)^2 + (z_o + tz_d)^2 = r^2 \tag{4}$$

$$x_o^2 + 2tx_ox_d + t^2x_d^2 + z_o^2 + 2tz_oz_d + t^2z_d^2 = r^2 \tag{5}$$

$$(x_d^2 + z_d^2)t^2 + 2(x_ox_d + z_oz_d)t + (x_o^2 + z_o^2 - r^2) = 0 \tag{6}$$

Now we can solve for t using the quadratic formula and the correct solution is $t >= 0$. where:

$$a = (x_d^2 + z_d^2) \tag{7}$$

$$b = 2(x_ox_d + z_oz_d) \tag{8}$$

$$c = (x_o^2 + z_o^2 - r^2) \tag{9}$$

and

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, t \geq 0 \tag{10}$$

A.3.1 Further Details. For any vector originating from inside the volume of our cylinder this returns two solutions. The positive solution representing the intersection with the cylinder in front on them and the negative behind them, which is disregarded.

Edge cases:

(1) Participant's camera vector is parallel to the y-axis, then the quadratic formula will return no real solutions, this is highly unlikely.
(2) Participant's camera vector exists outside of the cylinder (will never occur)
- If the camera vector intersects the cylinder, the first intersection is the $min(t_{soln_a}, t_{soln_b})$ and the second intersection the $max(t_{soln_a}, t_{soln_b})$.
- If the camera vector does **not** intersect the cylinder, the quadratic formula returns no real solutions.

A.4 Left Frame Center

Similarly we can use the camera model to calculate the left frame center. We define the ray starting at *lvo* and passing through *cam*. This ray points to the the left frame center, lfc, or the location of the center left pixel in world space. To calculate the lfc point we find the intersection of this ray and the frame plane defined by the camera vector, $\vec{n}$, and the previously calculated frame center. The ray defined by the points of the camera model:

$$P(t) = lvo + t(cam - lvo) \tag{11}$$

This ray intersects the plane when:

$$\vec{n} \cdot (P(t) - fc) = 0 \tag{12}$$

Solve for t

$$t = \frac{\vec{n} \cdot (fc - lvo)}{\vec{n} \cdot (cam - lvo)} \tag{13}$$

A.5 Horizontal Left Frame Center

Next we calculate the horizontal left frame center point, $hlfc$, where the left frame center would exist if the participant has not rolled their head. This requires the frame center point, fc, as calculated in section A.3. To calculate this point we perform a cross product between the camera vector, $\vec{a}$, and the a vector normal to the xz-plane, $\vec{b} = [0, -1, 0]$. The normalized resultant vector points in the direction of the horizontal left frame center. Multiplying the unit vector by the distance between the true left frame center and the frame center, o, and adding it the to frame center returns the horizontal left frame center.

$$hlfc = fc + \|\vec{a} \times \vec{b}\| * o \tag{14}$$

A.6 Frame Rotation

To calculate the frame rotation we create an isosceles triangle between the frame center, horizontal left frame center, and left frame center points, fc, $hlfc$, and lfc. First, we find the distance between the horizontal left frame center and left frame center points describing the base of the triangle, d. And the distance between the frame center and left frame center describing the leg, o. We then bisect the angle and calculate the angle of rotation θ:

$$sin(\tfrac{\theta}{2}) = \frac{d}{2o} \tag{15}$$

$$\theta = 2\arcsin(\frac{d}{2o}) \tag{16}$$

A.6.1 Further Notes.

To verify the direction of the angle we calculate the cross product between vectors frame center to horizontal left frame center and horizontal left frame center to left frame center. The direction of the cross product will either point in the same direction of the camera vector, representing a clockwise rotation, or the opposite direction, a clockwise rotation.

A.7 Scale Pixel Location to World Space

We transform and scale the pixel value to represent the (u, v) as real distance from the frame center ($u = 479.5$px, $v = 359.5$px). This method relies on the knowing the the field of view of the camera 60° horizontal and 46° vertical. Using the ratio between field of view and the horizontal distance from the frame center to the left frame center, d_h, we calculate the size of the frame in the vertical direction, d_v.

$$d_v = \frac{46}{60} * d_h \tag{17}$$

$$u_{real} = ((u - 479.5) * d_h)/479.5 \tag{18}$$
$$v_{real} = ((359.5 - v) * d_v)/359.5 \tag{19}$$

A.8 Apply Frame Rotation to Pixel

To rotate the pixel value, in real distance, to the true location in world space we calculate the rotation matrix with the angle, θ, calculated in section A.6. To calculate this rotation we let the axis of rotation be the unit camera vector, $\vec{u}$, the normalized vector pointing from the camera to frame center. This rotation is then applied to each pixel value $[u_{real}, v_{real}, 0]$.

$$R = (\cos\theta)\, I + (\sin\theta)\, [\mathbf{u}]_\times + (1 - \cos\theta)\, (\mathbf{u} \otimes \mathbf{u}) \tag{20}$$

where I is the identity matrix, $[\mathbf{u}]_\times$ is the cross product matrix, and $\mathbf{u} \otimes \mathbf{u}$ is the outer product

$$\mathbf{u} \otimes \mathbf{u} = \mathbf{u}\mathbf{u}^\mathsf{T} = \begin{bmatrix} u_x^2 & u_x u_y & u_x u_z \\ u_x u_y & u_y^2 & u_y u_z \\ u_x u_z & u_y u_z & u_z^2 \end{bmatrix}, \qquad [\mathbf{u}]_\times = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \tag{21}$$

The resultant point describes the pixel location in world space.