Camtrans Algorithm

1. Axis Aligned Rotation
   1.1)
   $$\text{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sin\theta & 0 \\ 0 & sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Arbitrary Rotation
   2.1) We can perform translation to shift the arbitrary point h to origin before performing the rotation, then translate back to point h after performing the rotation.
   The equation above will be changed as following for arbitrary point h = (h_x, h_y, h_z):
   $p' = T_2 \cdot M_1^{-1} \cdot M_2^{-1} \cdot M_3 \cdot M_2 \cdot M_1 \cdot T_1 \cdot p$
   where
   $$T_1 = \begin{bmatrix} 1 & 0 & 0 & -h_x \\ 0 & 1 & 0 & -h_y \\ 0 & 0 & 1 & -h_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ (Translation from h to origin)}$$
   $$T_2 = \begin{bmatrix} 1 & 0 & 0 & h_x \\ 0 & 1 & 0 & h_y \\ 0 & 0 & 1 & h_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ (Translation from origin to h)}$$

3. Camera Transformation
   3.1) Assuming camera is positioned at $P_0 = (x, y, z, 1)$, for $p' = M_1 \cdot M_2 \cdot M_3 \cdot M_4 \cdot p$:
   M_1 transforms perspective view volume into parallel one, preserving relative depth - "Unhinging transformation." (maps perspective to parallel)
   $$M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1+c} & \frac{-c}{1+c} \\ 0 & 0 & -1 & 0 \end{bmatrix} \text{ for near plane near, far plane far, } c = -near/far$$
   M_2 scales the view volume, so that bounds for x, y is at -1 and 1, and far clipping plane is at z = -1.
   $$M_2 = \begin{bmatrix} 1/fartan(\theta_w/2) & 0 & 0 & 0 \\ 0 & 1/fartan(\theta_h/2) & 0 & 0 \\ 0 & 0 & 1/far & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ for height angle } \theta_h \text{, width angle } \theta_w$$
   M_3 rotates to align the orthonormal unit vectors of camera coordinate system with x,y,z axes of world coordinate system.
   $$M_3 = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ for look vector look, up vector up, } w = \frac{-look}{\|\mathbf{look}\|} v = \frac{up-(up \cdot w)w}{\|\mathbf{up}-(\mathbf{up} \cdot \mathbf{w})\mathbf{w}\|}, u = v \times w$$
   M_4 translates the center of near plane to origin.
   $$M_4 = \begin{bmatrix} 1 & 0 & 0 & -P_{n_x} \\ 0 & 1 & 0 & -P_{n_y} \\ 0 & 0 & 1 & -P_{n_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ for } P_n = (P_{n_x}, P_{n_y}, P_{n_z} = P_0 - near * w)$$

   3.2) Unit vectors of camera coordinate system is defined in the world coordinate system, so we may use u, v, w vectors to describe the new position of the eye position E'.
   1. One unit left: E' = E - u
   2. One unit up: E' = E + v
   3. One unit forward: E' = E - w

3.3) I computed u,v,w in terms of u_0, v_0, w_0 by multiplying matrix M on $[u_0, v_0, w_0, 1]$.

1. Roll = rotation of camera around w, counterclockwise

   M = $\begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ for clockwise rotation

   $u = u_0 cos\theta + v_0 sin\theta$

   $v = -u_0 sin\theta + v_0 cos\theta$

   $w = w_0$

2. Pitch = rotation of camera around u, counterclockwise

   M = $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sin\theta & 0 \\ 0 & sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ for clockwise rotation

   $u = u_0$

   $v = v_0 cos\theta + w_0 sin\theta$

   $w = -v_0 sin\theta + w_0 cos\theta$

3. Yaw = rotation of camera around v, clockwise

   M = $\begin{bmatrix} cos\theta & 0 & sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ for clockwise rotation.

   $u = u_0 cos\theta + w_0 sin\theta$

   $v = v_0$

   $w = -u_0 sin\theta + w_0 cos\theta$