Virtual Reality
Assignment 5

Le Anh Phuong
Dang Thi Hoang Yen

16/12/18

Exercise 5.1

a)

Apply rotation to navigation node.
Let's create navigation2 node as a child of intersection node, such that:

$$n2.WT = n.WT$$

$$=> i.T * n2.T = n.T$$

$$=> n2.T = inv(i.T) * n.T$$

Apply rotation to navigation2 node, new navigation2 local transform is:

$$n2.T = R * n2.T = R * inv(i.T) * n.T$$

New transform matrix of navigation node is:

$$n.T = i.T * n2.T$$

$$=> n.T = i.T * R * inv(i.T) * n.T$$

b)

Transform matrix of pointer in intersection coordinate system is:

$$A = inv(i.T) * n.T * p.T$$

A consists of rotation and translation.
Distance from pointer to intersection would be calculated based on the translation matrix of A.

$$A.translate = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Z = distance = sqrt(x^2 + y^2 + z^2)$$

Let's attach intersection2 node as a child of intersection node, which represents new coordinate system and pointer2 node as a child of intersection2 node, which represents pointer in this new coordinate system. We need to calculate transform matrix of intersection2 node.

$$M = i2.T$$

$$p2.WT = p.WT$$

$$=> i.T * i2.T * p2.T = n.T * p.T$$

As requirement, local transform of pointer2 represents -z-translation of the pointer to the intersection.

$$=> P = p2.T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -Z \end{bmatrix}$$

Putting it together,

$$i.T * M * P = n.T * p.T$$

$$=> M = inv(i.T) * n.T * p.T * inv(P)$$

c)

delta is the transform of current frame pointer in relation to last frame pointer.

d)

Model transform of wheel1_geom

$$w1^W = cycle\_trans.T * wheel1\_trans.T * wheel1\_geom$$

View transform of wheel1_geom

$$w1^S = inv(s.T) * inv(n.T) * w1^W$$

Model View transform of wheel1_geom

$$w1^S = inv(s.T) * inv(n.T) * cycle\_trans.T * wheel1\_trans.T * wheel1\_geom$$

Exercise 5.2

Transformations matrix for front screen

$$s1.T = trans(0, 0, -2) * scale(4, 2, 1)$$

Transformations matrix for left screen

$$s2.T = trans(0, 0, -2) * scale(4, 2, 1) * rot(90, y)$$

Transformations matrix for right screen

$$s3.T = trans(0, 0, -2) * scale(4, 2, 1) * rot(-90, y)$$

Exercise 5.3

As stated in slide 15 in stereoscopic viewing lecture, maximal positive disparity is equal to eye distance.

Depth resolution, depth levels behind the screen is number of pixels corresponding to eye distance.

$$depth\ level = screen\ resolution/screen\ width * eye\ distance$$
$$=> depth\ level = 2560/400 * 6.5$$
$$=> depth\ level = 41$$

Exercise 5.4

Refer to slide 18 in viewing setup lecture, we can summeraize the parameters given as followed:

$$d = 2m$$
$$o_x = 0.7m$$
$$o_y = 1.75m$$
$$w = 4m$$
$$h = 2m$$
$$z_{near} = 0.05m$$
$$z_{far} = 0.05m$$

We calculate boundaries of viewing window in the z_{near} plane as followed:

$$\frac{l}{o_x - w/2} = \frac{z_{near}}{d}$$
$$=> \frac{l}{0.7 - 2} = \frac{0.05}{2}$$
$$=> l = -0.0325$$
$$\frac{r}{o_x + w/2} = \frac{z_{near}}{d}$$
$$=> \frac{r}{0.7 + 2} = \frac{0.05}{2}$$
$$=> r = 0.0675$$

Similarly, we calculate boundaries of viewing window in the z_{far} plane as followed:

$$\frac{l}{-o_x - w/2} = \frac{z_{far}}{d}$$

$$=> \frac{l}{-0.7-2} = \frac{100}{2}$$
$$=> l = -135$$
$$\frac{r}{-o_x + w/2} = \frac{z_{far}}{d}$$
$$=> \frac{r}{-0.7+2} = \frac{100}{2}$$
$$=> r = 65$$