Experiment: 6

Student Name: Shaurya Gulati
UID: 18BCS6092
Branch: AIML
Lab Group: A
Semester: 8th
Date of Performance: March 31
Subject Name: AR Lab
Subject Code: CSF- 484

1. Aim/Overview of the practical:

Introduction to viewing transforms; Canonical view of the human eye; Viewport transforms

2. Task to be done:

Discuss the viewing transforms, canonical view of the human eye and viewport transforms in detail. In Addition, implement as well as write down the steps to perform the viewport transforms.

3. Theory:

Viewing Transformations:

The view of the transformation is the coordinates of the point of forming an image with appropriate coordinates on the display device.

World Coordinate System (WCS) is a right-handed Cartesian coordinate system that defines captured images. WCS finite area is called a window. The corresponding coordinate system on the display device in which the image of the image is displayed is referred to as a physical coordinate system. The illustration of the window to the destruction of the display name called the viewport is called display translation.

The normalized device coordinates (NDC) are display areas of virtual display devices corresponding to a single field.

View Transformation:

A 3D scene can be viewed from any position in the 3D space. A "synthetic" camera positioned and oriented in 3D space can be used to describe the viewing, and the part of the image or scene to be viewed. It has the following three principal ingredients:

1. View Plane
2. View Coordinate System
3. An eye defined within this system

Canonical View:

Regular views are images of the most representative objects that come to mind first when assigning a name for the most accurate and fastest recognition performance.

Viewport Transformations:

The final step in the vertex transformation process is to map the point from the normalized view volume to the final position of the viewport on the screen. Converts the view volume to the shape and position of the viewport. A normalized view volume is a cube centred on the origin surrounded by points (-1, -1, -1) and (1, 1, 1). U and V coordinates of each point are converted to x and y screen coordinates (pixel coordinates), scaling the n coordinates (pseudo depth) from 0 to 1 (scaling the front layer to 0 and the rear layer (Scales to 1)). This depth value is called z.

x, y screen coordinates are used to draw the pixels in the refresh buffer. The z value is stored in the depth buffer for hidden surface determination.

The viewport transformation is represented by a 4x4 matrix and it encapsulates 2 stages:

1. scaling to the shape of the viewport
2. transformation to the position of the viewport

Steps to do viewport transformations:

The scaling matrix required is:

$$\hat{\mathbf{S}} = \begin{pmatrix} \frac{v_r - v_l}{2} & 0 & 0 & 0 \\ 0 & \frac{v_t - v_b}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

We need to translate the scaled normalised view volume to the position of the viewport. This is a combination of two translations; first, translate to the origin (add (vr−vl)/2 to u and add (vt−vb)/2 to v. Second move from the origin to the position of the viewport (add vl and vb to u and v). The n-coordinate (now in the range {−½.···½.}) needs to be shifted by +½.

So, the translation matrix becomes:

$$\hat{\mathbf{T}} = \begin{pmatrix} 1 & 0 & 0 & \frac{v_r + v_l}{2} \\ 0 & 1 & 0 & \frac{v_t + v_b}{2} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Combining the above two transformations, we get the Viewport transformation matrix.

$$\hat{\mathbf{V}}_{\mathbf{p}} = \hat{\mathbf{T}}\hat{\mathbf{S}} = \begin{pmatrix} \frac{v_r - v_l}{2} & 0 & 0 & \frac{v_r + v_l}{2} \\ 0 & \frac{v_t - v_b}{2} & 0 & \frac{v_t + v_b}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$