UNIT III: Three dimensional Concept- Three-Dimensional object representations – polygon surfaces – polygon tables- plane equations - Three-Dimensional geometric transformations – translation – rotation – scaling – other transformations.

TEXT BOOK

1. Donald Hearn and Pauline Baker, "Computer Graphics", Prentice Hall of India, 2001.

Prepared by **Dr.P.Sumathi**

THREE DIMENSIONAL CONCEPTS

Polygon Surfaces

Objects are represented as a collection of surfaces. 3D object representation is divided into two categories.

- **Boundary Representations (**B-reps): It describes a 3D object as a set of surfaces that separates the object interior from the environment.
- **Space–partitioning Representations**: It is used to describe interior properties, by partitioning the spatial region containing an object into a set of small, non-overlapping, contiguous solids (usually cubes).

The most commonly used boundary representation for a 3D graphics object is a set of surface polygons that enclose the object interior. Many graphics systems use this method. Set of polygons are stored for object description. This simplifies and speeds up the surface rendering and display of object since all surfaces can be described with linear equations.

The polygon surfaces are common in design and solid-modeling applications, since their wireframe display can be done quickly to give general indication of surface structure. Then realistic scenes are produced by interpolating shading patterns across polygon surface to illuminate.

Polygon Tables

We know that a polygon surface is defined by a set of vertices. As information for each polygon is input, the data are placed into tables that are used in later processing, display and manipulation of objects in the scene.

Polygon data tables can be organized into two groups: **Geometric tables and Attribute tables**.

- **Geometric data** tables contain vertex coordinates and parameters to identify the spatial orientation of the polygon surfaces.
- **Attribute information** for an object includes parameters specifying the degree of transparency of the object and its surface reflexivity and texture characteristics.

A suitable organization for storing geometric data is to create three lists, **a vertex table, an edge table and a polygon table.**

- Coordinate values for each vertex is stored in the vertex table.

- The edge table contains pointers back to the vertex table to identify the vertices for each polygon edge.
- The polygon table contains pointers back to the edge table to identify the edges for each polygon.

Plane Equations

The equation for plane surface can be expressed in the form

$$Ax + By + Cz + D = 0$$

Where (x, y, z) is any point on the plane, and the coefficients A, B, C, and D are constants describing the spatial properties of the plane. We can obtain the values of A, B, C, and D by solving a set of three plane equations using the coordinate values for three non collinear points in the plane. Let us assume that three vertices of the plane are (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3).

Let us solve the following simultaneous equations for ratios A/D, B/D, and C/D. You get the values of A, B, C, and D.

For any point x,y,z with parameters A, B, C, and D, we can say that –

- $Ax + By + Cz + D \neq 0$ means the point is not on the plane.
- $Ax + By + Cz + D < 0$ means the point is inside the surface.
- $Ax + By + Cz + D > 0$ means the point is outside the surface.

Three Dimensional Transformations

TRANSLATION

In a three-dimensional homogeneous coordinate representation, a point is translated (Figure given**)** from position P = ***(x,* y, *z)*** to position P' = ***(x',* y',** z') with the matrix operation.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

or

$$\mathbf{P'} = \mathbf{T} \cdot \mathbf{P}$$

Parameters t_x, t_y, *and* t_z, specifying translation distances for the coordinate directions x, y, and z, are assigned any real values. The matrix representation in Eq. 1 is equivalent to the three equations

$$x' = x + t_x, \qquad y' = y + t_y, \qquad z' = z + t_z$$

An object is translated in three dimensions by transforming each of the defining points of the object. For an object represented as a set of polygon surfaces, we translate each vertex of each surface and redraw the polygon facets in the new position.

We obtain the inverse of the translation matrix by negating the translation distances t_x , t_y & t_z. This produces a translation in the opposite direction, and the product of a translation matrix and its inverse produces the identity matrix.

ROTATION

To generate a rotation transformation for an object, we must designate an axis of rotation (about which the object is to be rotated) and the amount of angular rotation. Unlike two-dimensional applications, where all transformations are carried out in the xy plane, a three-dimensional rotation can be specified around any line in space. The easiest rotation axes to handle are those that are parallel to the coordinate axes. Also, we can use combinations of coordinate axis rotations.

By convention, positive rotation angles produce counterclockwise rotations about a coordinate axis, if we are looking along the positive half of the axis toward the coordinate origin.

(i) The two-dimensional z-axis rotation equations are easily extended to three dimensions:

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$
$$z' = z$$

Parameter θ specifies the rotation angle. In homogeneous coordinate form, the three-dimensional z-axis rotation equations are expressed as

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

(ii) We get the equations for x-axis rotations as

$$y' = y\cos\theta - z\sin\theta$$
$$z' = y\sin\theta + z\cos\theta$$
$$x' = x$$

which can be written in the homogeneous coordinate form

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

(iii) We get the equations for y-axis rotations as

$$z' = z\cos\theta - x\sin\theta$$
$$x' = z\sin\theta + x\cos\theta$$
$$y' = y$$

The matrix representation for y-axis rotation is

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

or

$$\mathbf{P}' = \mathbf{R}_y(\theta) \cdot \mathbf{P}$$

SCALING

The matrix expression tor the scaling transformation of a position P = (x, y, z) relative to the coordinate origin can be written as

$$P' = S \cdot P$$

Any positive numeric values can be assigned to the scaling factors s_x and s_y and s_z. Values less than 1 reduce the size of objects, values greater than 1 produce an enlargement. Specifying a value of 1 for s_x and s_y and s_z, leaves the size of objects unchanged.

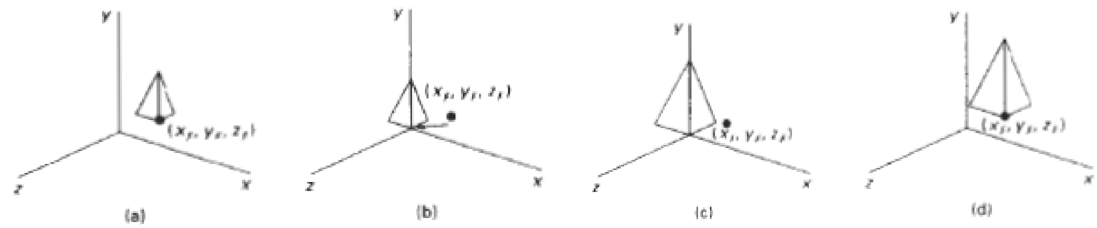
$$\mathbf{x' = x \cdot s_x}$$
$$\mathbf{y' = y \cdot s_y}$$
$$\mathbf{z' = z} \cdot s_z$$

Scaling with respect to a selected fixed position (x_f, y_f, z_f) can be represented with the following transformation sequence:

1. Translate the fixed point to the origin.
2. Scale the object relative to the coordinate origin
3. Translate the fixed point back to its original position.

This sequence of transformations is demonstrated in Fig. . The matrix representation for an arbitrary fixed-point scaling can then be expressed as the concatenation of these translate-scale-translate transformations as

$$\mathbf{T}(x_f, y_f, z_f) \cdot \mathbf{S}(s_x, s_y, s_z) \cdot \mathbf{T}(-x_f, -y_f, -z_f) = \begin{bmatrix} s_x & 0 & 0 & (1-s_x)x_f \\ 0 & s_y & 0 & (1-s_y)y_f \\ 0 & 0 & s_z & (1-s_z)z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



We form the inverse scaling matrix for the equations by replacing the scaling parameters s_x and s_y and s_z, with their reciprocals. The inverse matrix generates an opposite scaling transformation, so the concatenation of any scaling matrix and its inverse produces the identity matrix.
