Lecture 2: Projection and Transformation

3-Dimensional Objects Bounded by Planar Surfaces (Facets)

A planar facet is defined by an ordered set of 3D vertices, lying on one plane, which form a closed polygon, (straight lines are drawn from each vertex to the following one with the last vertex connected to the first). The data describing a facet are of two types. First, there is the numerical data which is a list of 3D points, (3*N numbers for N points), and secondly, there is the topological data which describes which points are connected to form edges of the facet.

Projections of Wire-Frame Models

Since our display device is only 2D, we have to define a transformation from the 3D space to the 2D surface of the display device. This transformation is called a *projection*. In general, projections transform an n-dimensional vector space into an m-dimensional vector space where m<n. Projection of a 3D object onto a 2D surface is done by selecting first the projection surface and then defining projectors or lines which are passed through each vertex of the object. The arrangement is shown in Diagram 2.1. The projected vertices are placed where the projectors intersect the projection surface. The most common (and simplest) projections used for viewing 3D scenes use planes for the projection surface and straight lines for projectors. These are called planar geometric projections.

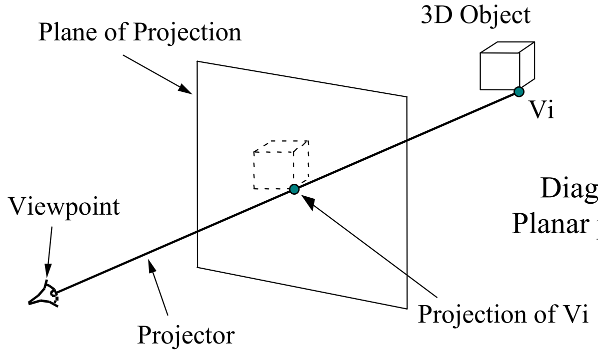


Diagram 2.1
Planar projection

The simplest form of viewing an object is by drawing all its projected edges. This is called a wire-frame representation, since the object could be modelled in three dimensions using wires for the edges of the object. Note that for such viewing the topological information for the facets is not required.

There are two common classes of planar geometric projections. Parallel projections use parallel projectors, perspective projections use projectors which pass through one single point called the viewpoint. In order to minimise confusion in dealing with a general projection problem, we can standardise the plane of projection by making it always parallel to the z=0 plane, (the plane which contains the x and y axis). This does not limit the generality of our discussion because if the projection plane of the actual scene is not parallel to the z=0 plane then we can use coordinate transformations in 3D and make the projection plane parallel to the z=0 plane. We shall restrict the viewed objects to be in the positive half space (z>0), therefore the projectors starting at the vertices will always run in the negative z direction.

Parallel Projections

If the direction of a projector is given by vector $\mathbf{d}=[d_x, d_y, d_z]$, and it passes through the vertex $\mathbf{V}=[V_x, V_y, V_z]$ it may be expressed by the parametric line equation:

$$\mathbf{P} = \mathbf{V} + \mu\mathbf{d}$$

In *orthographic* projection the projectors are perpendicular to the projection plane, which we define as z=0. In this case the projectors are in the direction of the z axis and:

$$\mathbf{d} = [0,0,-1]$$

and so $P_x = V_x$

and $P_y = V_y$

which means that the x and y co-ordinates of the projected vertex is equal to the x and y co-ordinates of the vertex itself and no calculations are necessary. A cube drawn in orthographic projection is shown in Diagram 2.2.

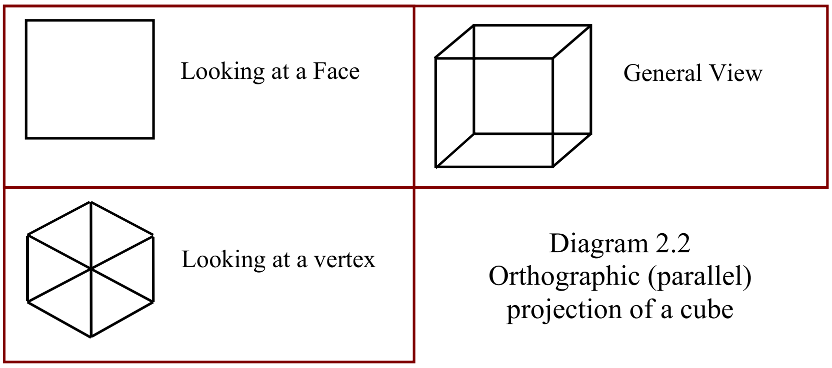


Diagram 2.2
Orthographic (parallel) projection of a cube

If the projectors are not perpendicular to the plane of projection then the projection is called *oblique*. The projected vertex intersects the z=0 plane where the z component of the **P** vector is equal to zero, therefore:

$$P_z = 0 = V_z + \mu\, d_z$$

so $\mu = -V_z / d_z$

and we can use this value of μ to compute:

$$P_x = V_x + \mu\, d_x = V_x - d_x V_z / d_z$$

and $P_y = V_y + \mu\, d_y = V_y - d_y V_z / d_z$

These projections are similar to the orthographic projection with one or other of the dimensions scaled. They are not often used in practice.

Perspective Projections

In perspective projection, all the rays pass through one point in space, the centre of projection, which we will designate with the capital letter C, as shown in Diagram 2.5. If the centre of projection is behind the plane of projection then the orientation of the image is the same as the image. By contrast, in a pin hole camera it is inverted. To calculate perspective projections we adopt a canonical form in which the centre of projection is at the origin, and the projection plane is placed at a constant z value, z=f. The projection of a 3D point onto the z=f plane is calculated as follows. If the centre of projection is at the origin, and we are projecting the point **V** then the projector has equation:

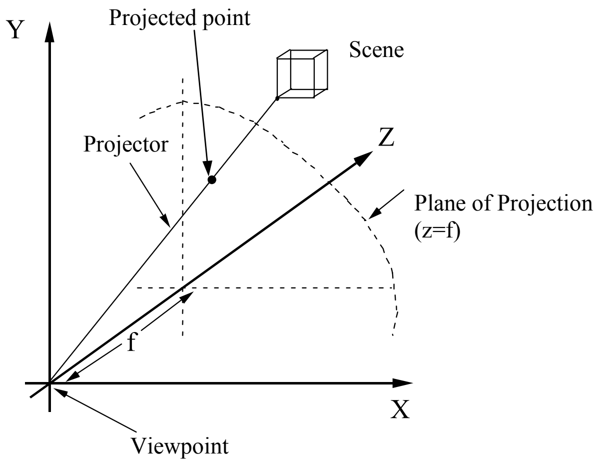


Diagram 2.3: Canonical form for Perspective projection

$$P = \mu \mathbf{V}$$

Since the projection plane has equation z=f, it follows that:

$$f = \mu\, V_z$$

If we write $\mu_p = f / V_z$ for the intersection point on the plane of projection then:
thus

$$P_x = \mu_p V_x = f^* V_x / V_z$$

and $P_y = \mu_p V_y = f^* V_y / V_z$

The factor μ_p is called the foreshortening factor, because the further away an object is, the larger V_z and the smaller is its image. The perspective projection of a cube is shown in Diagram 2.4.

The introduction of canonical forms for perspective and orthographic projection simplifies their computation. However it means that we must be able to transform a scene, which could be defined in any 3D coordinate system, such that the view direction is along the z axis and (for perspective projection) the viewpoint is at the origin. In general we would like to change the coordinates of every point in the scene, such that some chosen viewpoint **C** = [Cx,Cy,Cz] is the origin and some view direction **d** = [dx,dy,dz] is the Z axis. Frequently, we may want to transform the points of a graphical scene for other purposes such as generation of special effects in pictures, like rotating objects. Transformations of this kind are achieved by multiplying every point of the scene by a transformation matrix. Unfortunately however, we cannot perform a general translation using normal Cartesian coordinates, and for that reason we now introduce a system called *homogeneous coordinates*. Three dimensional points expressed in homogeneous form have a fourth ordinate:

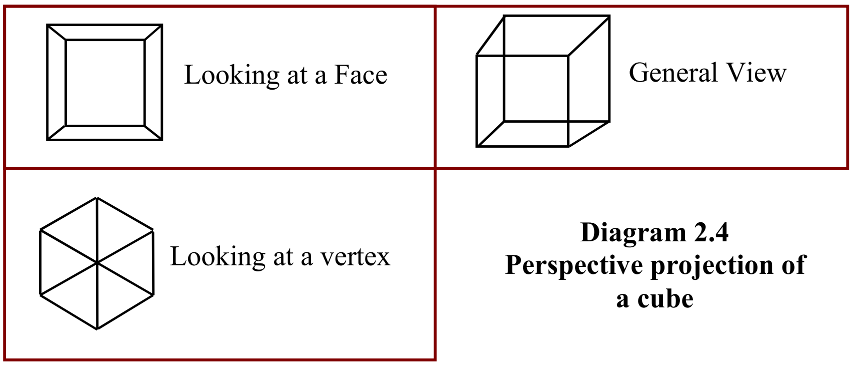


Diagram 2.4
Perspective projection of a cube

$$\mathbf{P} = [p_x, p_y, p_z, s]$$

The fourth ordinate is a scale factor, and conversion to Cartesian form is achieved by dividing it into the other ordinates, so

$[p_x, p_y, p_z, s]$ has Cartesian coordinate equivalent $[p_x/s, p_y/s, p_z/s]$.

In most cases, s will be 1. The point of introducing homogenous coordinates is to allow us to

$$[\,x, y, z, 1] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ tx & ty & tz & 1 \end{pmatrix} = [x+tx, y+ty, z+tz, 1\,]$$

translate the points of a scene by using matrix multiplication.

The matrix for scaling a graphical scene is also easily expressed in homogenous form:

$$[x, y, z, 1] \begin{pmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = [sx*x, sy*y, sz*z, 1]$$

Notice that these two transformations are not commutative, and it is essential that they are carried out in the correct order. Diagram 2.5 illustrates the problem for a simple picture.

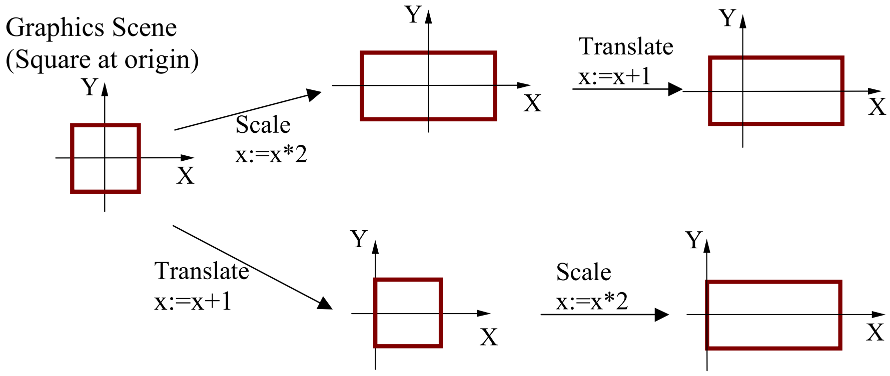


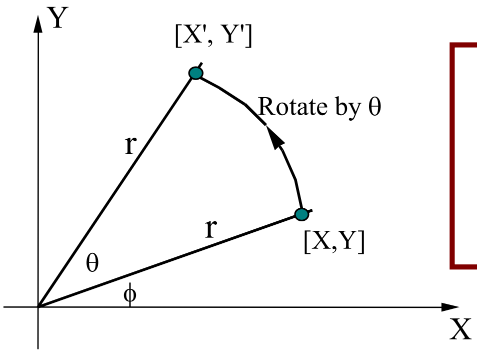
Diagram 2.5: The order in which transformations are applied is significant

Rotation has to be treated differently since we need to specify an axis. The matrices for rotation about the three Cartesian axes are:

$$\boldsymbol{R}x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & Cos(\theta) & Sin(\theta) & 0 \\ 0 & -Sin(\theta) & Cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \boldsymbol{R}y = \begin{pmatrix} Cos(\theta) & 0 & -Sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ Sin(\theta) & 0 & Cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\boldsymbol{R}z = \begin{pmatrix} Cos(\theta) & Sin(\theta) & 0 & 0 \\ -Sin(\theta) & Cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Some care is required with the signs. The above formulation obeys the conventions of a left hand axis system. That is, if the positive y-axis is taken as vertical, and the positive x-axis horizontal to the right, the positive z-axis is into the page. In these cases, rotation is in a clockwise direction when viewed from the positive side of the axis, or vice versa, anti-clockwise when viewed from the negative side of the axis. The derivation of the ***Rz*** matrix is shown in Diagram 2.6 the others may be proved similarly.



$$\begin{aligned} [X,Y] &= [r\,Cos\phi,\ r\,Sin\phi] \\ [X',Y'] &= [\ r\,Cos(\theta+\phi)\ ,\ r\,Sin(\theta+\phi)\ ] \\ &= [\ r\,Cos\phi\,Cos\theta - rSin\phi\,Sin\theta,\ \ rSin\phi Cos\theta +\ rCos\phi Sin\theta\ ] \\ &= [\ X\,Cos\theta - Y\,Sin\theta,\ YCos\theta + XSin\theta] \\ &= [\ X\ \ Y\ ] \begin{pmatrix} Cos\theta & Sin\theta \\ -Sin\theta & Cos\theta \end{pmatrix} \end{aligned}$$

Diagram 2.6: Derivation of the Rotation Matrix

Inversions of these matrices can be computed easily, without recourse to Gaussean elimination, by considering the meaning of each transformation. For scaling, we substitute $1/s_x$ for s_x, $1/\ s_y$ for s_y and $1/\ s_z$ for s_z to invert the scaling. For translation we substitute $-t_x$ for t_x, $-\ t_y$ for t_y and $-\ t_z$ for t_z. For the rotation matrices we note that:

$$Cos(-\theta) = Cos(\theta) \text{ and } Sin(-\theta) = -Sin(\theta)$$

Hence to invert the matrix we simply change the sign of the Sin terms.