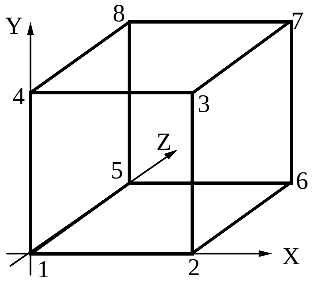
Lecture 2: Worlds in 2D and 3D

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. For the cube shown in Diagram 2.1 we need 24 real numbers for the numerical data, 24 integers to store the line topology, and 24 integers to store the face topology. We will also need to maintain the number of lines and faces in the figure, and the number of edges per face. All this could be done using static structures (arrays), alternatively, if we start with abstract data types that express the structure of three dimensional objects, we may define the following data types (Diagram 2.2):



NUMERICAL DATA	TOPOLOGICAL DATA	
Points	Lines	Faces
1. [0,0,0] 2. [2,0,0] 3. [2,2,0] 4. [0,2,0] &c	1. 1>>2 2. 1>>4 3. 1>>5 4. 3>>4 5. 3>>2 &c	1,2,4,5 1,3,7,8 &c

Diagram 2.1: Representing 3D objects

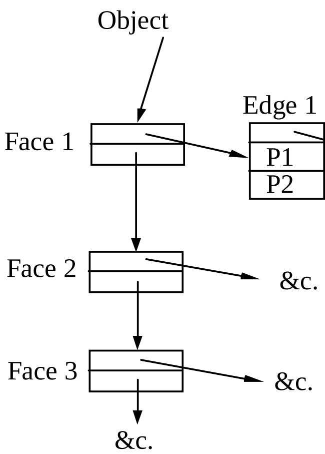


Diagram 2.2
Possible Organisation for Graphical Data

```
ordinate=x,y,z;
Point = array[ordinate] of real;
Edge =      record
                start, finish: Point;
            end ;
Edgelist =  record
                thisedge: Edge ;
                nextedge: ^Edgelist;
            end ;
Facetlist = record
                thisfacet: ^Edgelist;
                nextfacet:^Facetlist;
            end ;
Object=^Facetlist;
```

It should be noted that redundancy exists in this case, since edges which belong to two facets are duplicated, and vertices which belong to three edges appear three times. However, when a large number of objects are processed, redundancy of data may help the speed. Later on in this lecture we will see other examples where this is true.

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

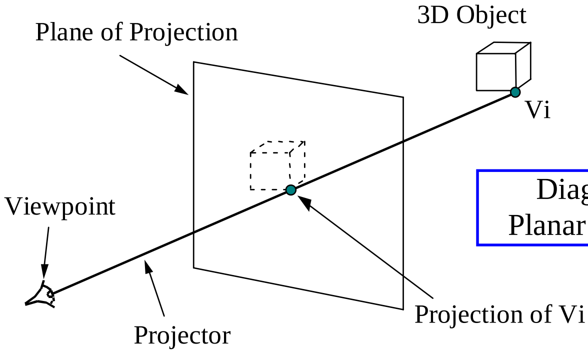


Diagram 2.3
Planar projection

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 2.3. 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.

Depending on how the projected 2D lines are computed they may become curved. For example, if we compute the image of a straight line focussed by a lens. However, all the projections that we will consider in detail will produce straight lines or points for straight edges in 3D. The simplest form of viewing such 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 classes of the most common planar geometric projections. Parallel projections use parallel projectors, perspective projections use projectors which pass through one single point. Parallel projectors are defined by the direction of projectors, while perspective projectors are defined by the centre of projection. In order to minimise our confusion in dealing with a general projection problem, we can visualise the plane of projection more easily 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 rotation transformations in 3D and make the projection plane parallel to the z=0 plane. A simple translation in the z direction now can place either the projection plane or the centre of projection at the origin. We shall restrict the viewed objects to be in the positive half space, therefore the projectors starting at the vertices will always run in the negative z direction.

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

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

The simplest case is when the projectors are perpendicular to the projection plane, (called *orthographic* projection). 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 2.4.

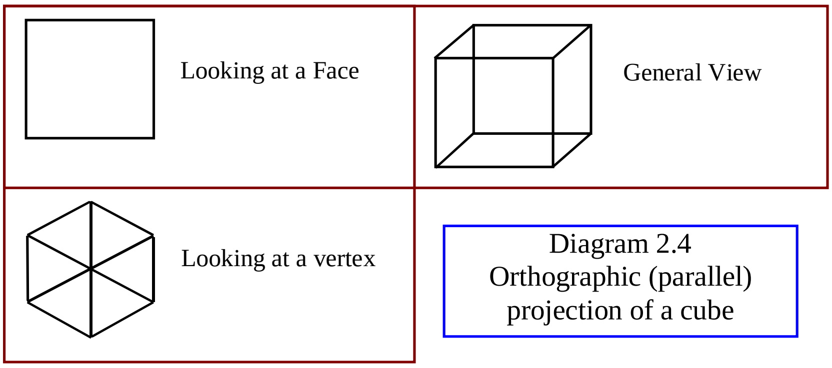


Diagram 2.4
Orthographic (parallel) projection of a cube

If the projectors are not perpendicular to the projection plane then the projection is called *oblique*. and 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 produced image is the same as the image. To calculate perspective projections it easier to place the centre of projection at the origin, in which case 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:

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

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

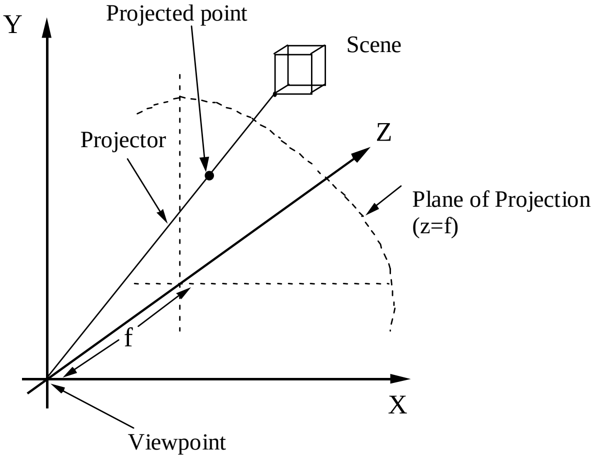


Diagram 2.5: Canonical form for Perspective projection (Catholic, not Copernican)

The perspective projection may also be calculated without recourse to vector methods. Diagram 2.6 shows a picture of the plane through the y-z axis looking at a particular projector.

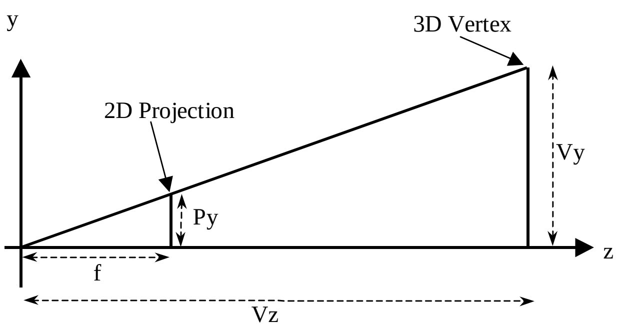


Diagram 2.6: Perspective projection by similar triangles

by similar triangles we see that:

$$Py/f = Vy/Vz$$

Thus $Py = f\ Vy/Vz$

By considering a plane through the x axis and the projector we can similarly obtain the result

$$Py = f\ Vy/Vz$$

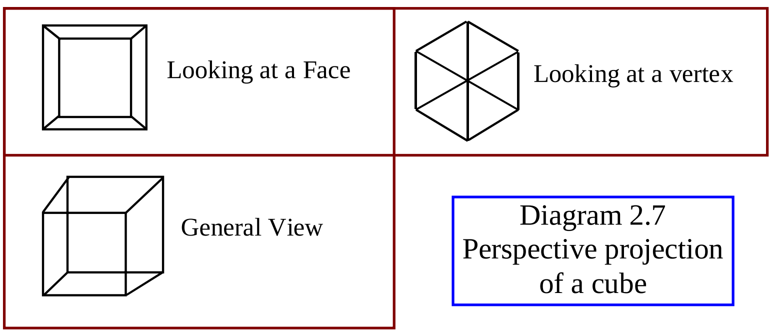


Diagram 2.7 Perspective projection of a cube

The factor μ_p is called the foreshortening factor, because the further an object is, the larger V_z and the smaller is its image. The perspective projection of a cube is shown in Diagram 2.7.

One of the interesting properties of perspective projection is that lines that are parallel in three dimensional space are not necessarily parallel in the 2D projection. In some cases they will meet in the image at points called *vanishing points*. Architects who use free hand and not a computer to draw perspective images use the vanishing point technique. This technique uses the fact that projected images of parallel lines which are not parallel to the projection surface all pass through one image point, and images of parallel lines which are parallel to the projection surface remain parallel. A vanishing point may be interpreted as the perspective projection of a point at infinity (since this is the point where the mathematicians tell us that parallel lines meet). We can calculate the image vanishing point $\mathbf{I}=[I_x, I_y]$ by the equation of lines which are all parallel to a 3D direction vector, $\mathbf{d}=[d_x, d_y, d_z]$. Assuming an arbitrary point $\mathbf{V_0}$, the vector equation of these parallel lines is:

$$\mathbf{L} = \mathbf{V_0} + \mu\mathbf{d}$$

Since the point $\mathbf{L}=[L_x, L_y, L_z]$ is on one of these lines and the x and y co-ordinates of the same point projected onto the z=0 plane are:

$$P_x = f * L_x / L_z$$
$$P_y = f * L_y / L_z,$$

substituting we get:

$$P_x = f(V_{0x} + \mu d_x)/(V_{0z} + \mu d_z)$$
$$P_y = f(V_{0y} + \mu d_y)/(V_{0z} + \mu d_z)$$

We may find the point at infinity by letting μ approach infinity in which case V_{0x}, V_{0y} and V_{0z} disappear from the equations and we get:

$$\mu \dashrightarrow \text{infinity}$$
$$P_x = I_x = f(d_x / d_z)$$
and $$P_y = I_y = f(d_y / d_z)$$

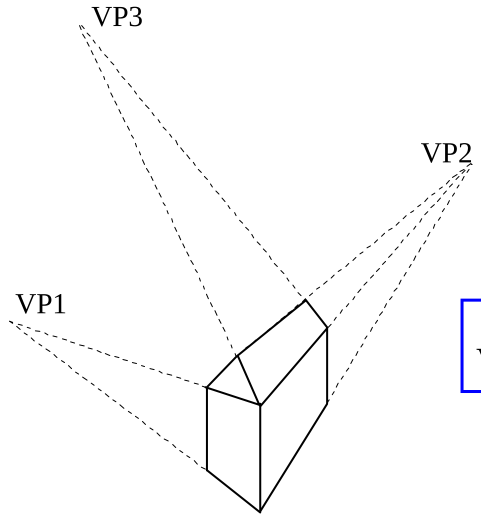


Diagram 2.8
Vanishing Points

Since the parameters V_{0x}, V_{0y} and V_{0z} select between different 3D lines, the above equations indicate that all these parallel lines pass through one point in the 2D space with vanishing point [Ix,Iy] which is a function only of the 3D direction of the lines. The equations also show that lines which are parallel to the projection plane are projected as parallel lines in which case their vanishing point is at infinity. All these calculations assumed that the centre of projection is located at the origin and the projection plane is at z=f. The vanishing point technique is illustrated in Diagram 2.8.