Lecture 4: More Transforms and Homogenous Coordinates

In the previous lecture we discussed three transformations, translation, scaling and rotation, and noted that these are all affine, in other words they preserve parallelism and linearity. We also saw that orthographic projection and perspective projection could be defined in terms of matix transformations, and combined with the other transforms in a consistent system, though the projections were not invertable. Perspective projection is clearly not affine since it does not preserve parallelism. We will add to our list of transformations two further ones which are not so common, but could be used for special effects.

The first is reflection, which in its simplest form could be considered equivalent to a negative scaling. We can define three matrices for the reflections in the planes x=0,y=0 and z=0 respectively. These are trivially:

$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

And we can derive a matrix for reflection in a general plane **n•p** = k using the same method that we used for rotation about an arbitrary line, and for the flying sequences. In other words, we do a translation of the points such that the plane of reflection goes through the origin, and then two rotations so that the normal vector is lined up with one of the coordinate axes. The reflection is then done with one of the above matrices and the inverse transformations applied to return the objects to the correct place in the 3D space. It will be seen that reflection is affine.

The second transformation is shear. Shears are most easily understood if applied as a deformation to one axis at a time. For example, we can apply shear to x only using the matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ b & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

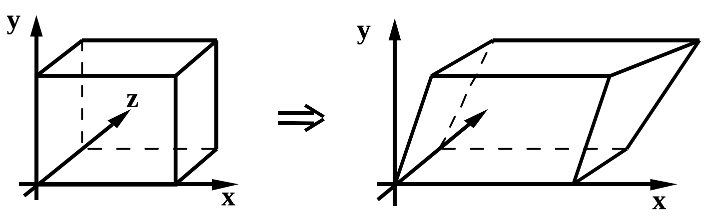

Diagram 4.1 The shear transformation

This will have the effect of distorting a cube as shown in Diagram 4.1. Any points on the x axis remain unchanged, and if b=0, the base of the cube, sitting on the x-z plane, remains unchanged while the top face is translated in the x direction by a distance a*y. In this simple case the transformation is affine. Similarly, if a=0 we shear the cube in the z direction, and the transformation is still affine, however, if both a and b are non-zero, then we get a more complex distortion in which the parallel lines are destroyed. The effect of shear is dependent on the distance from the origin.

Homogenous coordinates

We now take a second look at homogeneous coordinates, and their relation to vectors. In the previous lecture we described the fourth ordinate as a scale factor, and ensured that, with the exception of the projection transformation, the last ordinate was always normalised to 1. As an alternative, we can consider the fourth ordinate as indicating a type as follows. Informally we acknowledge that a normal Cartesian coordinate is a special form of vector, which we call a position vector, and usually denote using capital letters. Hence, we can say that a normalised homogenous coordinate is the same as a position vector.

By contrast, if we consider the case where the last ordinate is zero - [x,y,z,0] - we find that we cannot normalise this coordinate in the usual way because of the divide by zero, so clearly a homogenous coordinate of this form cannot be directly associated with a point in Cartesian space. However, it still contains information in the relative sizes of x y and z, and hence we can consider it to be a direction vector. Thus homogenous coordinates fall into two classes, those with the final ordinate non-zero, which can be normalised into position vectors, and those with zero in the final ordinate which are direction vectors, and which also have direction magnitude.

Consider now how vector addition works with these definitions. If we add two direction vectors, we add the ordinates as before and we obtain a direction vector. ie:

[xi,yi,zi,0] + [xj,yj,zj,0] = [xi+xj, yi+yj, zi+zj,0]
This is the normal vector addition rule which operates independently of Cartesian space. However, if we add a direction vector to a position vector we obtain a position vector or point:
[Xi,Yi,Zi,1] + [xj,yj,zj,0] = [Xi+xj,Yi+yj,Zi+zj,1]
This is a nice result, because it ties in with our definition of a straight line in cartesian space being defined by a one point and a direction as shown in Diagram 4.2.

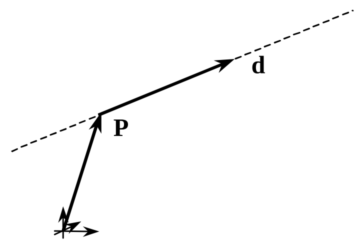

Diagram 4.2 Adding a direction vector to a position vector

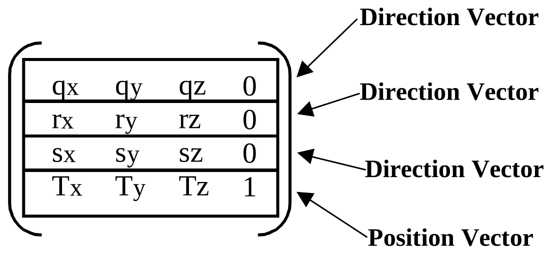

Diagram 4.3 The composition of an affine transformation matrix

Now, consider a general affine transformation matrix. We ignore the possibility of doing perspective projection or shear, so that the last column will always be $[0,0,0,1]^T$, and the matrix will be of the form shown in diagram 4.3, with the rows viewed as three direction vectors and a position vector. We ask what the individual rows mean, and to see this we consider the effect of the transformation in simple cases. For example take the unit vectors along the Cartesian axes eg:

$$[1,0,0,0] \begin{pmatrix} qx & qy & qz & 0 \\ rx & ry & rz & 0 \\ sx & sy & sz & 0 \\ Tx & Ty & Tz & 1 \end{pmatrix} = [qx, qy, qz, 0]$$

In other words the direction vector of the top row represents the direction in which the x axis points after transformation, and similarly we find that ***j***= [0,1,0,0] will be transformed to direction [rx,ry,rz,0] and ***k*** = [0,0,1,0] will be transformed to [sx,sy,sz,0]. Similarly, we can see the effect of the bottom row by considering the transformation of the origin which has homogeneous coordinate [0,0,0,1]. This will be transformed to [Tx,Ty,Tz,1]. Notice also that the zero in the last ordinate ensures that direction vectors will not be affected by the translation, whereas all position vectors will be moved by the same factor. Notice also that if we do not shear the object the three vectors **q** **r** and **s** will remain orthogonal so that **q•r** = **r•s** = **q•s** = 0.

Unfortunately however, this analysis does not help us to determine the transformation matrix. In general it would be more natural to assume that we know the vectors ***u***,***v***, and ***w*** which we would like to transform into the Cartesian axes ***i***,***j***,***k***. This is the case when for example we are transforming a scene before viewing it in the normal position for computing a projection, that is with the viewpoint at the origin and the viewing direction along the z axis. In this case we need to use the notion of the dot product as a projection onto a line. This is most readily seen in two dimensions as indicated in Diagram 4.4. By dropping perpendiculars from the point **P** to the line defined by vector ***u*** and vector ***v*** we see that the distances from the origin are respectively **P•*u*** and **P•*v*.** Now, suppose that we wish to rotate the scene so that the new x and y axes were the u and v vectors, then the x ordinate would be defined by **P•*u*** and the y by **P•*v***.

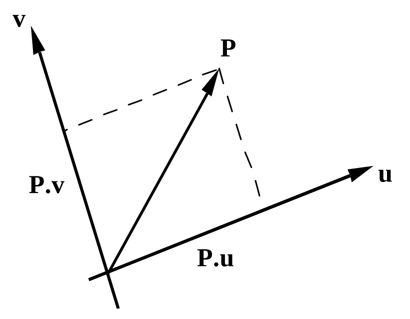

Diagram 4.4
The dot product as a projection

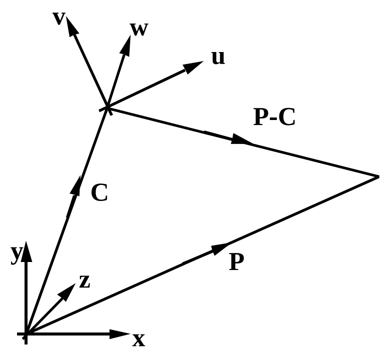

Diagram 4.5
Change of axes using the dot product

The generalisation of the result is shown in Diagram 4.5 where the transformation of P into the {***u***,***v***,***w***} axis system, translated by vector **C**, is given by:

P'x = (**P-C**)•***u***
P'y = (**P-C**)•***v***
P'z = (**P-C**)•***w***

Expressing this as a transformation matrix we get:

$$\begin{pmatrix} ux & vx & wx & 0 \\ uy & vy & wy & 0 \\ uz & vz & wz & 0 \\ -\mathbf{C}\bullet\mathbf{u} & -\mathbf{C}\bullet\mathbf{v} & -\mathbf{C}\bullet\mathbf{w} & 1 \end{pmatrix}$$

You could verify this by checking that the transformation we developed for a flying system does indeed have the direction vector [dx,dy,dz] as its third column.