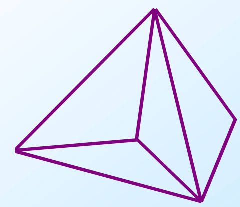
Interactive Computer Graphics

Lecture 2:

Three Dimensional objects, Projection and Transformations

Planar Polyhedra

These are three dimensional objects whose faces are all *planar polygons* often called *facets*.



Representing Planar Polygons

In order to represent planar polygons in the computer we will require a mixture of numerical and topological data.

Numerical Data

Actual 3D coordinates of vertices, etc.

Topological Data

Details of what is connected to what

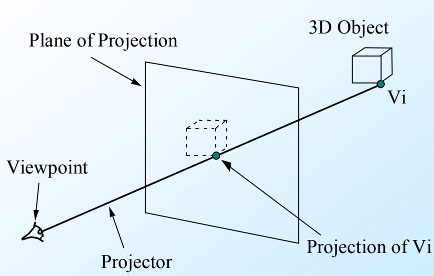
Projections of Wire Frame Models

Wire frame models simply include points and lines.

In order to draw a 3D wire frame model we must first convert the points to a 2D representation. Then we can use simple drawing primitives to draw them.

The conversion from 3D into 2D is a form of projection.

Planar Projection



Non Linear Projections

In general it is possible to project onto any surface:

Sphere

Cone

etc

or to use curved projectors, for example to produce lens effects.

However we will only consider planar linear projections.

Normal Orthographic Projection

This is the simplest form of projection, and effective in many cases.

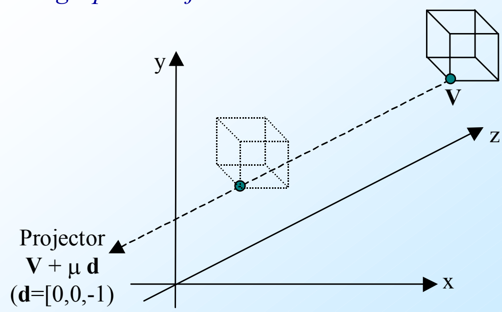
The viewpoint is at $z = -\infty$

The plane of projection is z=0

so

All projectors have direction d = [0,0,-1]

Orthographic Projection onto z=0



Calculating an Orthographic Projection

Projector Equation:

P = **V** + μ **d**

Substitute **d** = [0,0,-1]

Yields cartesian form

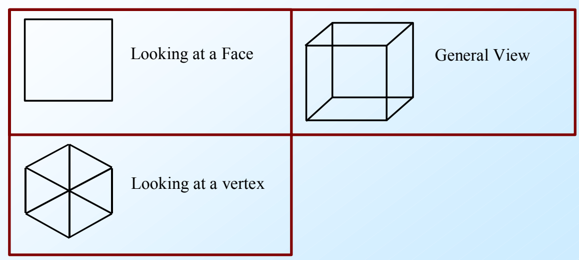
Px = Vx + 0   Py = Vy + 0   Pz = Vz - μ

The projection plane is z=0 so the projected coordinate is

[Vx,Vy,0]

ie we simply take the 3D x and y components of the vertex

Orthographic Projection of a Cube



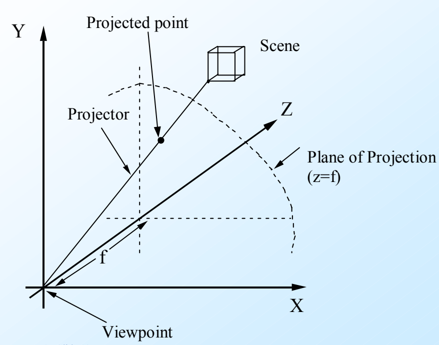
Perspective Projection

Orthographic projection is fine in cases where we are not worried about depth (ie most objects are at the same distance from the viewer).

However for close work (particularly computer games) it will not do.

Instead we use perspective projection

Canonical Form for Perspective Projection



Calculating Perspective Projection

Projector Equation:

P = μ **V** (all projectors go through the origin)

At the projected point Pz=f

$\mu_p = P_z/V_z = f/V_z$

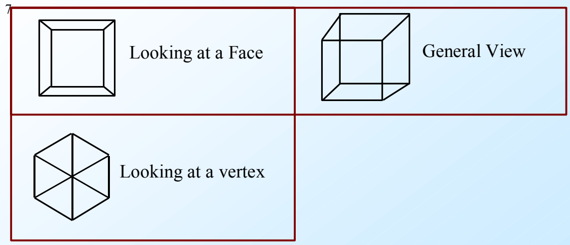
$P_x = \mu_p V_x$ and $P_y = \mu_p V_y$

Thus

$P_x = f V_x/V_z$ and $P_y = f V_y/V_z$

The constant μ_p is sometimes called the fore-shortenting factor

Perspective Projection of a Cube



Problem Break

Given that the viewpoint is at the origin, and the viewing plane is at z=5: What point on the viewplane corresponds to the 3D vertex {10,10,10} in

a. Perspective projection

b. Orthographic projection

Perspective $x' = f\,x/z = 5$ and $y' = f\,y/z = 5$

Orthographic $x' = 10$ and $y' = 10$

The Need for Transformations

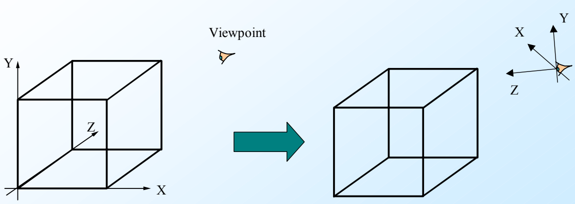
Graphics scenes are defined in one co-ordinate system

We want to be able to draw a graphics scene from any angle

To draw a graphics scene we need the viewpoint to be the origin and the z axis to be the direction of view.

Hence we need to be able to transform the coordinates of a graphics scene.

Transformation of viewpoint



Coordinate System for definition

Coordinate System for viewing

Matrix transformations of points

To transform points we use matrix multiplications.

For example to make an object at the origin twice as big we could use:

$$[x',y',z'] = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

yields

$x' = 2x \quad y' = 2y \quad z' = 2z$

Translation by Matrix multiplication

Many of our transformations will require translation of the points.

For example if we want to move all the points two units along the x axis we would require:

$x' = x + 2$
$y' = y$
$z' = z$

But how can we do this with a matrix?

Honogenous Coordinates

The answer is to use homogenous coordinates

$$[x', y', z', 1] = [x, y, z, 1] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \end{pmatrix}$$

General Homogenous Coordinates

In most cases the last ordinate will be 1, but in general we can consider it a scale factor.

Thus:

[x, y, z, s] is equivalent to [x/s, y/s, z/s]

Homogenous Cartesian

Affine Transformations

Affine transformations are those that preserve parallel lines.

Most transformations we require are affine, the most important being:

- Scaling
- Translating
- Rotating

Other more complex transforms will be built from these three.

Translation

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

Inverting a translation

Since we know what transformation matrices do, we can write down their inversions directly

For example:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ tx & ty & tz & 1 \end{pmatrix} \text{ has inversion } \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -tx & -ty & -tz & 1 \end{pmatrix}$$

Scaling

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

Inverting scaling

$$\begin{pmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ has inversion } \begin{pmatrix} 1/sx & 0 & 0 & 0 \\ 0 & 1/sy & 0 & 0 \\ 0 & 0 & 1/sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Combining transformations

Suppose we want to make an object at the origin twice as big and then move it to a point [5, 5, 20].

The transformation is a scaling followed by a translation:

$$[x',y',z',1] = [x, y, z, 1] \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 5 & 5 & 20 & 1 \end{pmatrix}$$

Combined transformations

We multiply out the transformation matrices first, then transform the points

$$[x',y',z',1] = [x, y, z, 1] \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 5 & 5 & 20 & 1 \end{pmatrix}$$

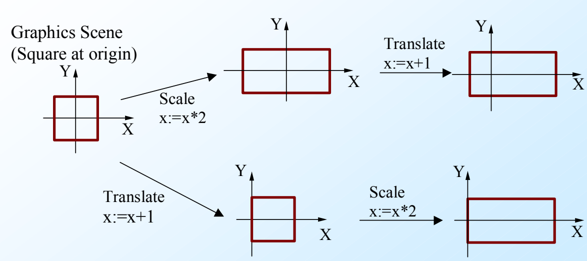
Transformations are not commutative

The order in which transformations are applied matters:

In general

T* * *S is not the same as ***S* * *T***

The order of transformations is significant



Rotation

To define a rotation we need an axis.

The simplest rotations are about the Cartesian axes

eg

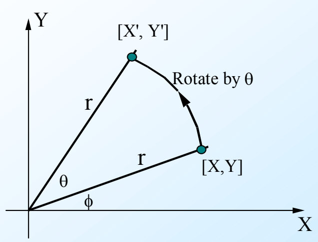
Rx - Rotate about the X axis
Ry - Rotate about the Y axis
Rz - Rotate about the Z axis

Rotation Matrices

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

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

Deriving Rz



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

Signs of Rotations

Rotations have a direction.

The following rule applies to the matrix formulations given in the notes:

Rotation is clockwise when viewed from the positive side of the axis

Inverting Rotation

Inverting a rotation by an angle θ is equivalent to rotating through an angle of -θ, now

$\mathrm{Cos}(-\theta) = \mathrm{Cos}(\theta)$

and

$\mathrm{Sin}(-\theta) = -\mathrm{Sin}(\theta)$

Inverting Rz

$$\begin{pmatrix} \mathrm{Cos}(\theta) & \mathrm{Sin}(\theta) & 0 & 0 \\ -\mathrm{Sin}(\theta) & \mathrm{Cos}(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ has inversion } \begin{pmatrix} \mathrm{Cos}(\theta) & -\mathrm{Sin}(\theta) & 0 & 0 \\ \mathrm{Sin}(\theta) & \mathrm{Cos}(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$