

# 3D Graphics

## Graphics – Lecture 9

Lecture 9

1

## Introduction

- We have mastered most of the techniques required for 2d graphics.
- Could now write applications as diverse as mapping systems, video games and CASE tools.
- Ready to take a surprisingly small quantum leap into 3-dimensional graphics.
- real topic of the course.
- 2d applications are fine, but there is something almost magical about staring ‘through’ a computer screen into a cyberspace and seeing worlds and objects that don’t really exist.

Lecture 9

2

## Introduction

- Modelling 3d with data structures
  - Expanding 2d points, lines, transforms etc.
- Drawing 3d on a 2d screen
  - projections

Lecture 9

3

## Introduction

- Where are we going
  - (show VRML demo)
  - Where would you like to go today?
  - (show Wire-frame aeroplane demo)

Lecture 9

4

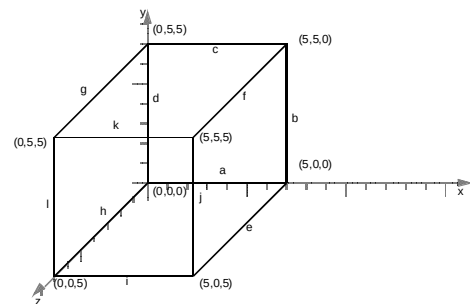
## 3D coordinate system

- In 2d we had height and breadth - but all shapes were “flat”
- Need to add a third coordinate to represent depth
- By convention we use 3 mutually perpendicular axes, ( any 3 non co-planar axes would do).
- Any point in 3d space can thus be represented by just those three numbers x,y & z

Lecture 9

5

## 3D coordinate system



Lecture 9

6

## 2D -> 3D

- representation of a cube using our new coordinate system.
- Each line of the cube (a-l) can still be described by its endpoints just as in 2d, but
- each point now has 3 coordinates.
- good news is that we can still use our drawing, shape, line and point classes.
- bad news is that they are all going to require some (hopefully small) alterations to work in 3d.

Lecture 9

7

## 3D Points

- Still using homogenous coordinates

- Need 4 coords

$$(x, y, z, 1) \quad \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Lecture 9

8

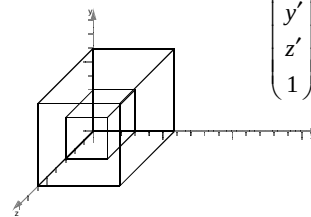
## 3D Transformations

- Since we now have an additional coordinate to cope with, our transformation matrices must also expand to cope.
- each transformation matrix will require an extra row and an extra column to become 4x4 matrices.

Lecture 9

9

## Scaling

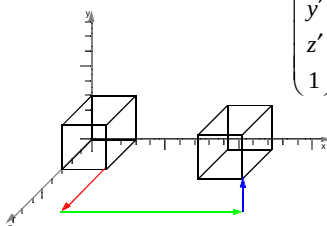


$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Lecture 9

10

## Translation



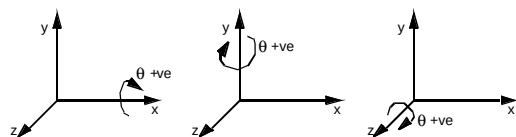
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Lecture 9

11

## Rotation (about axes)

- In 2 dimensions, rotation were defined to take place about some stationary point - the centre of rotation. In 3d, all rotation take place about one of the axes.

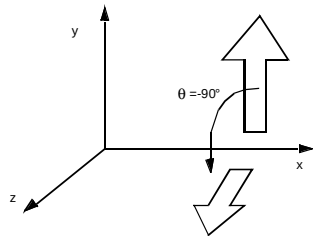


Lecture 9

12

## Rotation (about axes)

-90 about x axis



Lecture 9

13

## Rotation

- z-axis

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

- y-axis

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

- z-axis

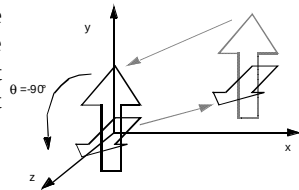
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Lecture 9

14

## Local Rotation

- translate the object to the origin, perform the rotation and translate it back to where it started from



Lecture 9

15

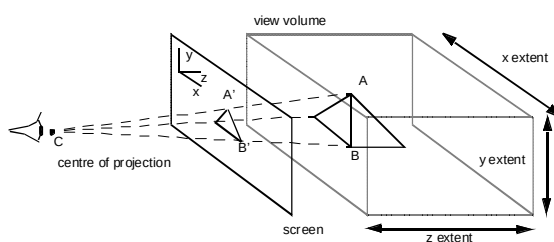
## Projection

- Representation of a 3d object on a 2d screen
- Originally developed by architects and draughtsmen
- Several different projects exist
- Compare the unit cube as viewed in the different projections

Lecture 9

16

## Terms and definitions

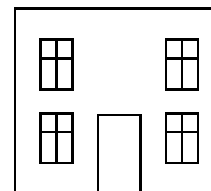


Lecture 9

17

## Parallel Projection

$$\begin{pmatrix} xp \\ yp \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



Lecture 9

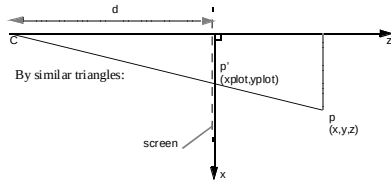
18

## Perspective Projection

$$\frac{x}{z+d} = \frac{xp}{d}$$

$$xp = \frac{x}{1 + \frac{z}{d}}$$

$$yp = \frac{y}{1 + \frac{z}{d}}$$

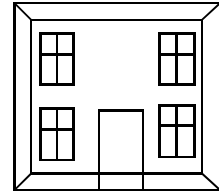


Lecture 9

19

## Perspective Projection

$$\begin{pmatrix} x \\ y \\ 0 \\ 1 + \frac{z}{d} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$



Lecture 9

20

## Perspective Projection

want

$$= \begin{pmatrix} \frac{x}{1 + \frac{z}{d}} \\ \frac{y}{1 + \frac{z}{d}} \\ 0 \\ 1 \end{pmatrix}$$

So....  
Divide by  $1 + \frac{z}{d}$

Lecture 9

21

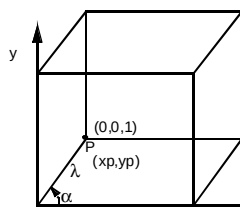
## Oblique Parallel Projections

- A parallel projection is one where all the lines that are parallel in 3d space remain parallel when projected.
- An oblique parallel projection is one where the object in question is viewed "from the side" - unlike the plain parallel projection.
- The z axis is drawn at some angle ( $\alpha$ ) to the x axis
- z coordinates are multiplied by some factor ( $\lambda$ )

Lecture 9

22

## Oblique Parallel Projections



Consider the point P:

P can be represented in 3D space - (0,0,1)

P can be represented in 2D (on screen coords) - (xp,yp)

P can be represented in 2D via polar coordinates - ( $\alpha, \lambda$ )

Lecture 9

23

## Oblique Parallel Projections

- $xp = \lambda \cos \alpha$
- $yp = \lambda \sin \alpha$
- This is valid only for (0,0,1)
- Generally multiply by z and allow for x and y
- $xp = x + z \cdot \lambda \cdot \cos \alpha$
- $yp = y + z \cdot \lambda \cdot \sin \alpha$

Lecture 9

24

## Oblique Parallel Projection as a matrix

$$\begin{pmatrix} xp \\ yp \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \lambda \cos \alpha & 0 \\ 0 & 1 & \lambda \sin \alpha & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

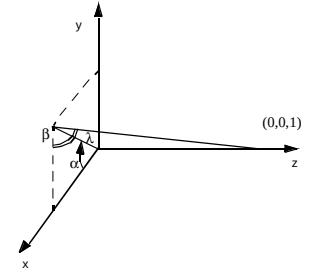
Lecture 9

25

## Oblique Parallel Projection

A number of standard oblique parallel projections are used in engineering drawing. They can all be specified in terms of  $(\alpha, \beta)$  or  $(\alpha, \lambda)$  since clearly (!)

$$\tan \beta = 1/\lambda$$



Lecture 9

26

## Oblique Parallel Projection

|               |                |                       |                    |
|---------------|----------------|-----------------------|--------------------|
| $\lambda=1$   | $\beta = 45$   | Cavalier projection   | $\alpha = 0 - 360$ |
| $\lambda=0.5$ | $\beta = 63.4$ | Cabinet projection    | $\alpha = 0 - 360$ |
| $\lambda=0$   | $\beta = 90$   | Orthogonal projection | $\alpha = 0 - 360$ |

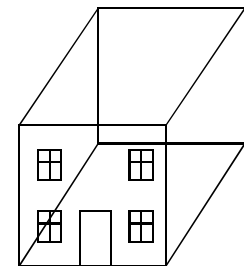
Lecture 9

27

## Cavalier Projection

The length of a line on the screen is equal to its length in the model.

This causes a distortion by over emphasising the z-axis

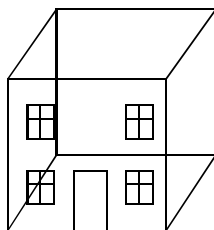


Lecture 9

28

## Cabinet Projection

The foreshortening of the z axis is increased to provide a more "realistic" view.

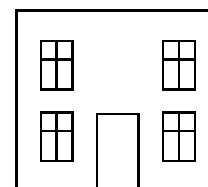


Lecture 9

29

## Orthogonal Projection

The orthogonal projection is just the simple parallel projection

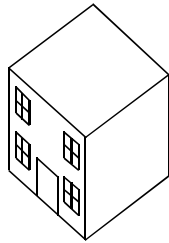


Lecture 9

30

## Isometric Projection

$$\begin{pmatrix} \sin 60^\circ & 0 & -\cos 60^\circ & 0 \\ 0 & 1 & 0 & 0 \\ -\sin 60^\circ & 0 & -\cos 60^\circ & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



All 3 axes are equally foreshortened allowing measurements along the axes to be made with the same scale.

Lecture 9

31

## Summary

Lecture 9

32

## Next time....

- Data structures for 3D
  - how to describe the world to a computer

Lecture 9

33