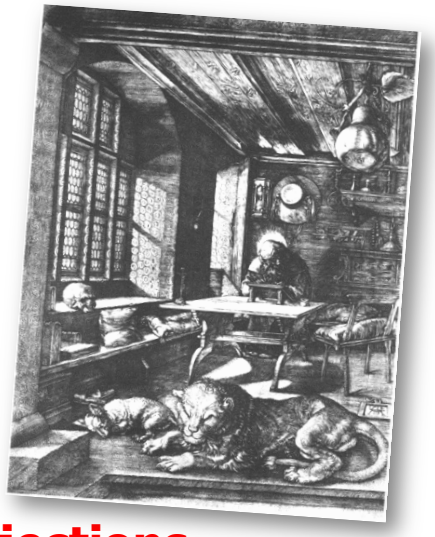
MANCHESTER 1824

COMP27112

Computer Graphics and Image Processing



6: Viewing 2: Projections

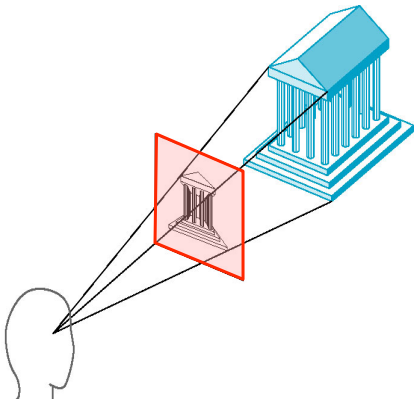
Toby.Howard@manchester.ac.uk

1

MANCHESTER 1824

Introduction

- In part 2 of our study of Viewing, we'll look at
  - The theory of geometrical planar projections
  - Classes of projections
    - Perspective
    - Parallel
  - Basic mathematics of projections
  - Matrix representations of projections



Temple figures courtesy: Prof. Ed Angel, University of New Mexico

2

MANCHESTER 1824

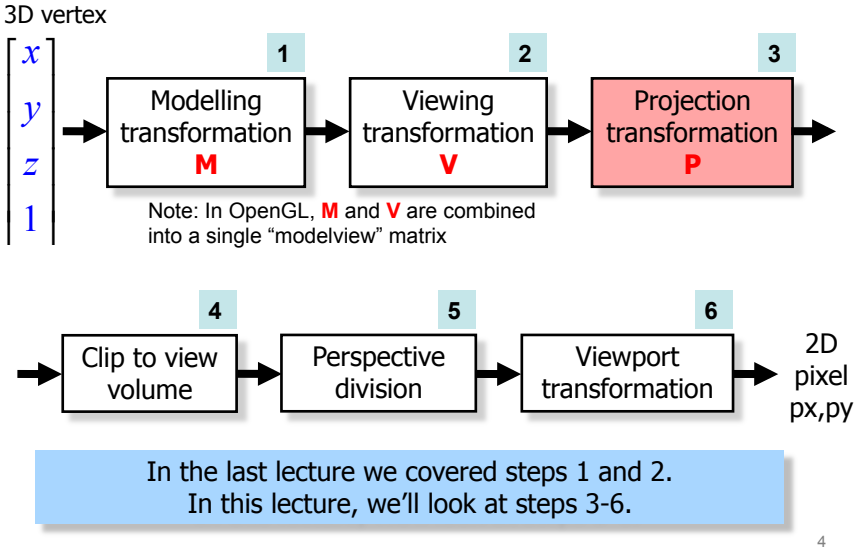
3D Viewing: the camera analogy

	Real World	Computer Graphics
1	Arrange the scene into the desired composition	Set Modelling Transformation
2	Point the camera at the scene	Set Viewing Transformation
3	Choose the camera lens, or adjust the zoom	Set Projection Transformation
4	Determine the size and shape of the final photograph	Set Viewport Transformation

3

MANCHESTER 1824

The 3D Viewing Pipeline



In the last lecture we covered steps 1 and 2.
In this lecture, we'll look at steps 3-6.

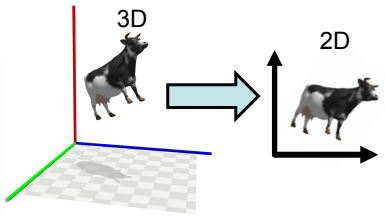
4

MANCHESTER 1824

Plamar geometric projections

- How do we specify the mapping from 3D coordinates to 2D coordinates on the screen?
- We clearly have to somehow convert all the 3D coordinates into suitable 2D coordinates.
- This conversion process is called **projection**.
- There are two classes of planar geometric projection:
  - Parallel projection
  - Perspective projection

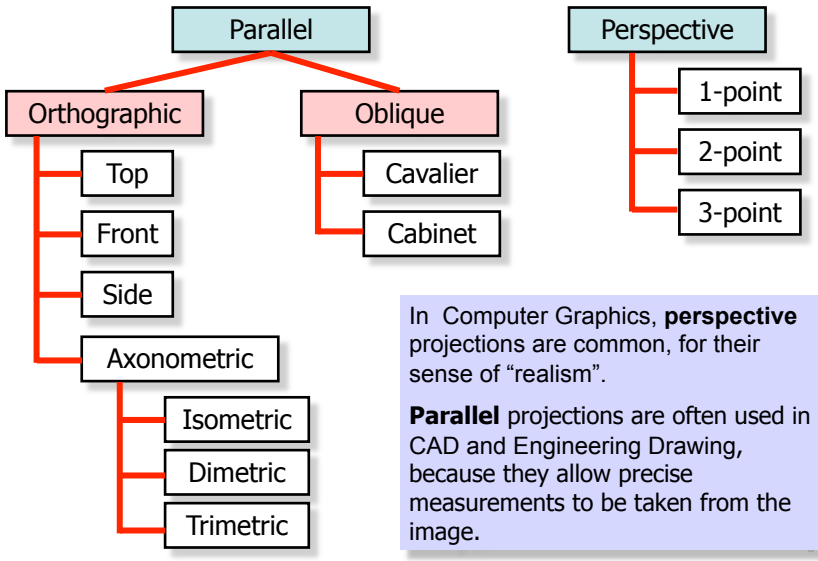
Note: We're considering only **planar geometric projections**: the kind that project lines to lines. Other kinds of projections change lines into curves and vice versa (used in cartography, for example), which we will not study.



5

MANCHESTER 1824

Planar Geometric Projections



In Computer Graphics, **perspective** projections are common, for their sense of "realism".

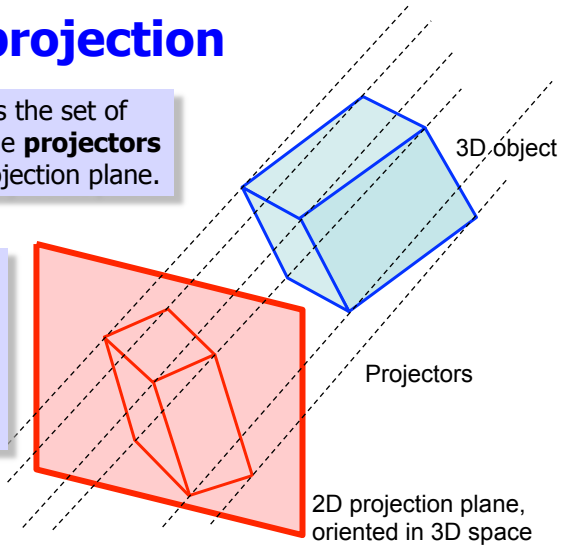
Parallel projections are often used in CAD and Engineering Drawing, because they allow precise measurements to be taken from the image.

MANCHESTER 1824

Parallel projection

The **projection** is the set of points at which the **projectors intersect** the projection plane.

Parallel edges remain parallel in the projection. Angles between edges may be **distorted**.

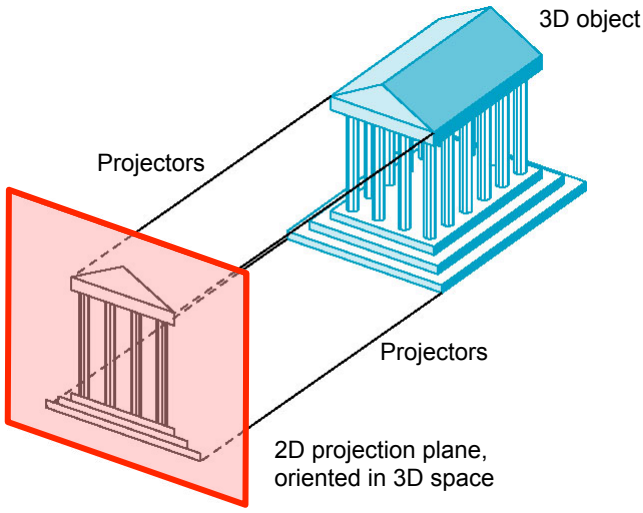


Centre of projection, or "eye point", is **at infinity**, so projectors are **parallel**

7

MANCHESTER 1824

Parallel projection



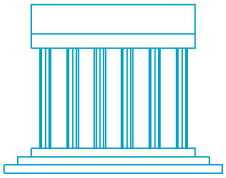
8

MANCHESTER 1824

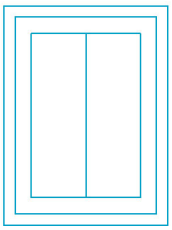
Parallel projection: orthographic

- In orthographic parallel projection, projectors are **perpendicular** to the projection plane
- The projection plane is **parallel** to one of the (X,Y,Z) axes of the 3D object being viewed
- The projected image shows only 2 of the 3 axes
- There is no distortion of lengths or angles, so well-suited for engineering drawing and CAD

Front



Side

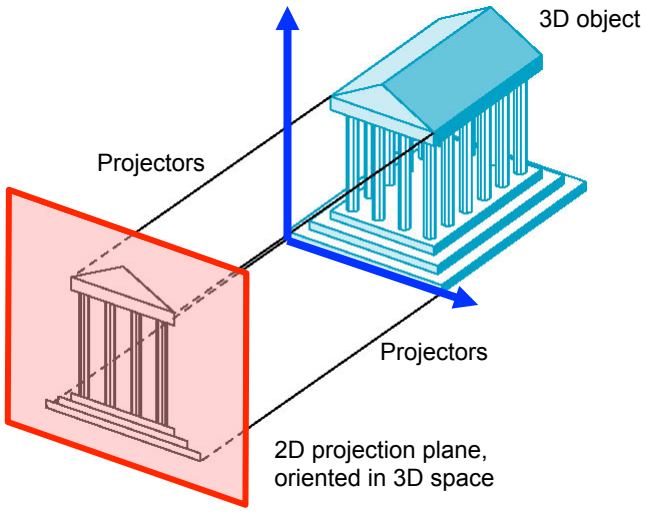


Top

9

MANCHESTER 1824

Parallel projection

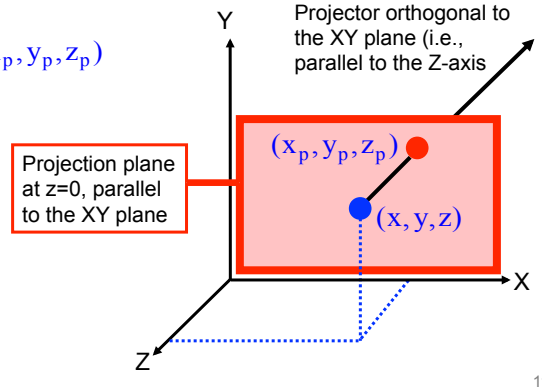


10

MANCHESTER 1824

Computing orthographic projection

- Suppose we have a projection plane located at **z=0**, and the plane is parallel to the XY plane
- A 3D point (x, y, z) will project to (x_p, y_p, z_p) on the projection plane
- We want to find (x_p, y_p, z_p)
- By definition:
  - $x_p = x$
  - $y_p = y$
  - $z_p = 0$



11

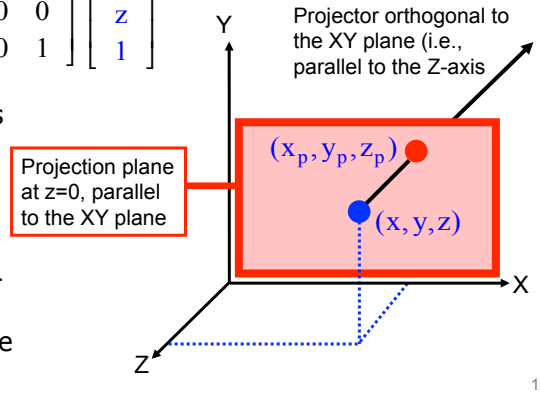
MANCHESTER 1824

Matrix for orthographic projection

- We can express the projection as the matrix

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- This is the same as scaling by (1,1,0)
- Notice this matrix is singular
- We can derive similar matrices for other positions of the projection plane

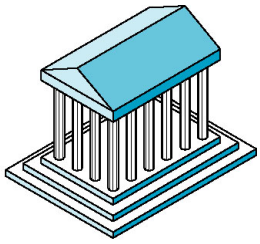


12

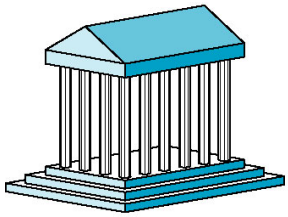
MANCHESTER 1824

Parallel projection: axonometric

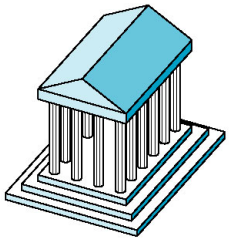
- In axonometric parallel projection, projectors are **perpendicular** to the projection plane
- The projection plane can have **any orientation** relative to the object being viewed
- Now we can see the three axes. There is distortion of lengths or angles, but measurements can still be made



Isometric: projection plane is symmetrical to (X,Y,Z)



Dimetric: projection plane is symmetrical to 2 of (X,Y,Z)



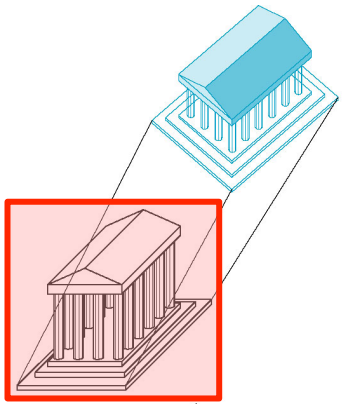
Trimetric: projection plane placed arbitrarily

13

MANCHESTER 1824

Parallel projection: oblique

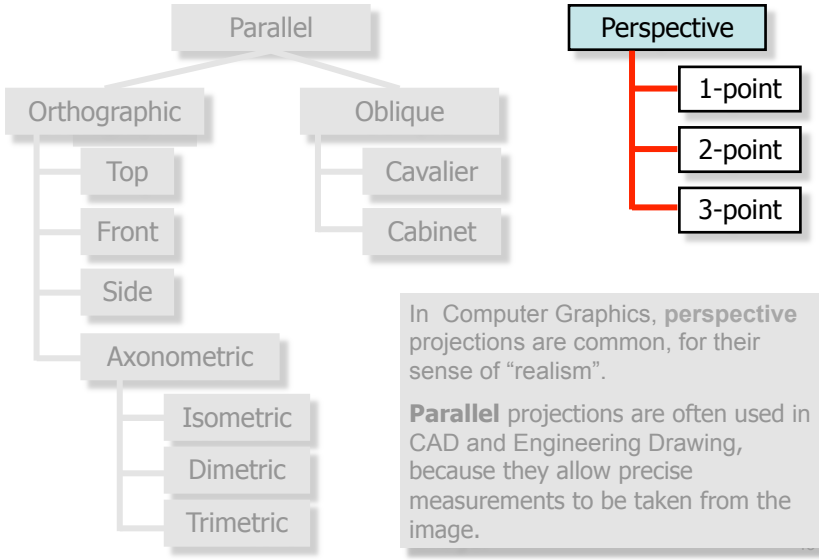
- This is the most general case of parallel projection
- Projectors can make **any angle** with the projection plane
- The projection plane can have **any orientation** relative to the object being viewed
- We see the three axes. There is distortion of lengths or angles, but measurements can still be made
- We can of course derive matrices to perform **all the parallel projections** we have seen *(but we will not go into the details in this course)*



14

MANCHESTER 1824

Planar Geometric Projections



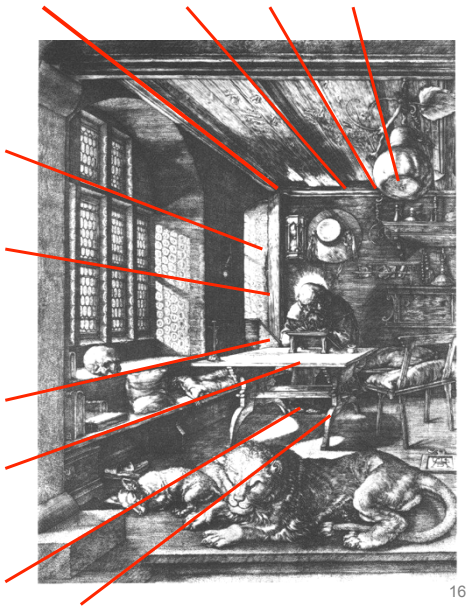
In Computer Graphics, **perspective** projections are common, for their sense of "realism".

Parallel projections are often used in CAD and Engineering Drawing, because they allow precise measurements to be taken from the image.

MANCHESTER 1824

Perspective

- During the Renaissance, artists began to understand and use perspective
- This is an engraving by Albrecht Dürer, "St Jerome dans sa Cellule" (1514)

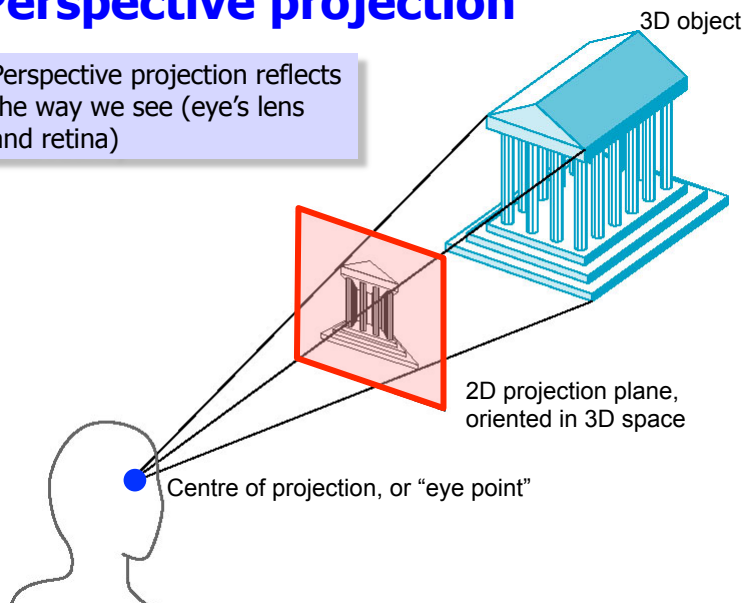


16

MANCHESTER 1824

Perspective projection

Perspective projection reflects the way we see (eye's lens and retina)

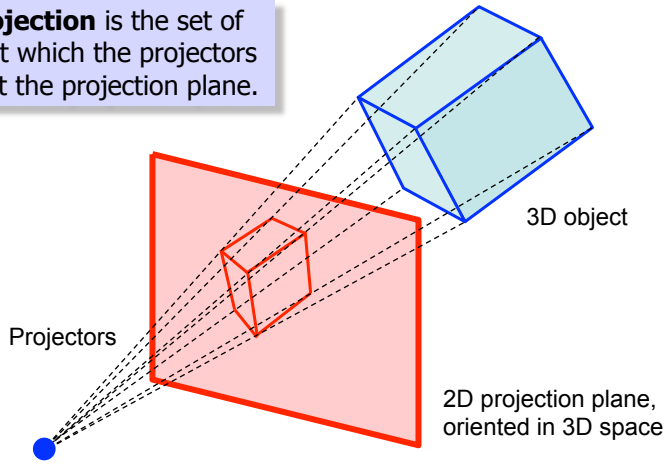


17

MANCHESTER 1824

Perspective projection

The **projection** is the set of points at which the projectors intersect the projection plane.



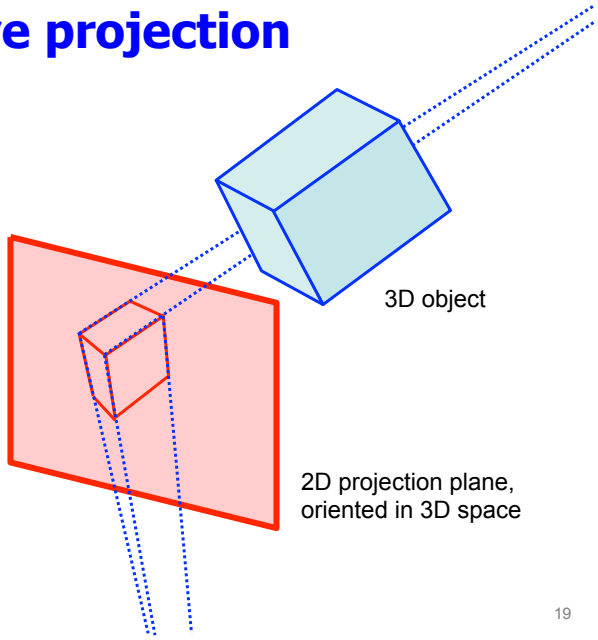
Centre of projection, or "eye point", is **a point**, so projectors **converge**

18

MANCHESTER 1824

Perspective projection

Objects further away from the COP become smaller. Edges that were parallel may converge. Angles between edges may be **distorted**.

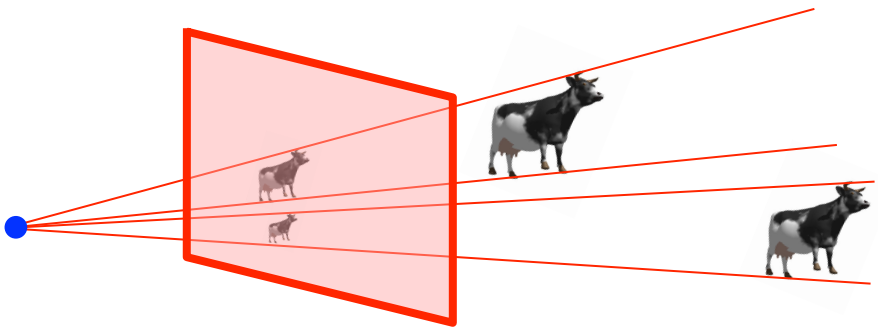


19

MANCHESTER 1824

Small or far away?

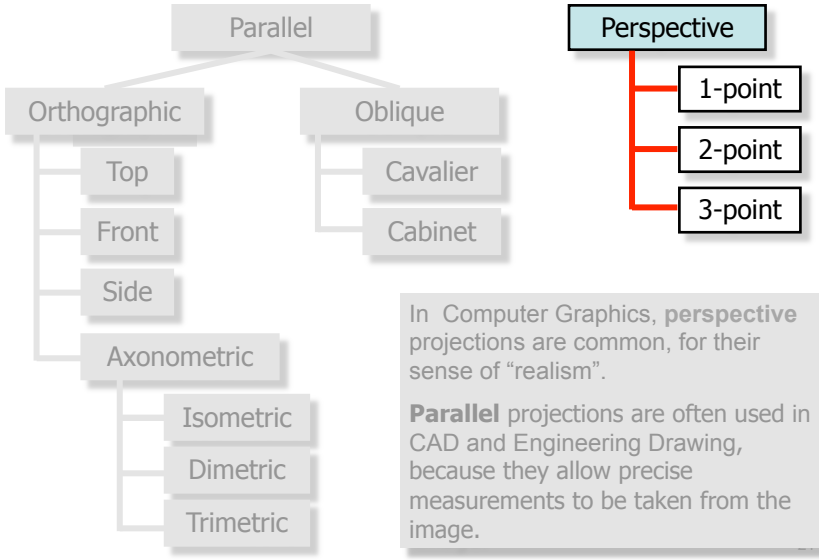
- With perspective projection, distant objects appear smaller in the projected image



20

MANCHESTER 1824

Planar Geometric Projections



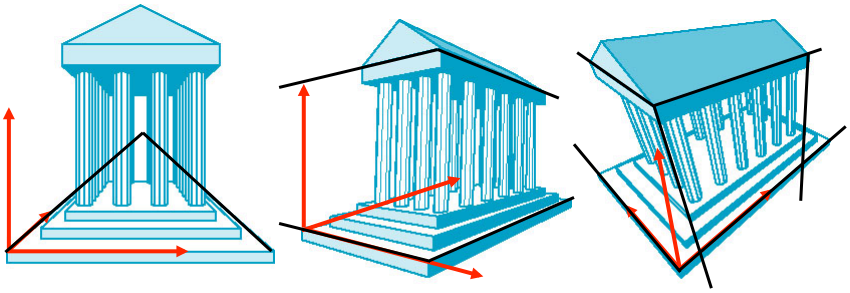
In Computer Graphics, **perspective** projections are common, for their sense of "realism".

Parallel projections are often used in CAD and Engineering Drawing, because they allow precise measurements to be taken from the image.

MANCHESTER 1824

Classical perspective projections

- How many of the (x,y,z) axes of the model are parallel to the projection plane determines how many **vanishing points** are seen in the projected image
  - 2 of (x,y,z) are parallel to the projection plane: 1-point perspective
  - 1 of (x,y,z) is parallel to the projection plane: 2-point perspective
  - 0 of (x,y,z) is parallel to the projection plane: 3-point perspective



1-point perspective | 2-point perspective | 3-point perspective

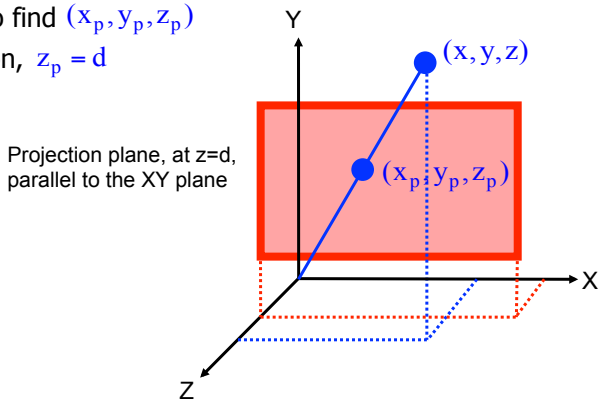
22

"False Perspective" by William Hogarth, 1753

MANCHESTER 1824

Computing perspective

- Suppose we have a projection plane located at z=d, and the plane is parallel to the XY plane
- A point (x, y, z) will project to (x_p, y_p, z_p) on the projection plane
- We want to find (x_p, y_p, z_p)
- By definition, $z_p = d$



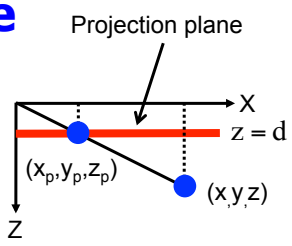
24

MANCHESTER 1824

Computing perspective

- We need to find x_p and y_p
- Looking at the top view, looking down the y-axis, by similar triangles, we have:

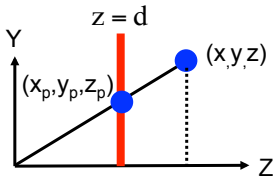
$$\frac{x_p}{d} = \frac{x}{z} \rightarrow x_p = \frac{dx}{z} \rightarrow x_p = \frac{x}{z/d}$$



Top view (looking down Y axis)

- Similarly, in a side view, with our eye at the origin and looking along the x-axis, we have:

$$y_p = \frac{y}{z/d}$$



Side view (looking down the x-axis)

25

MANCHESTER 1824

Perspective projection matrix

- We can express this transformation as a 4x4 matrix:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- If we multiply a point by this matrix, we get the transformed point

$$\begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix}$$

- Recall however that we must always end up with a 3D point which has **w = 1**, so we need to divide through all elements by w:
- This is called **perspective division**

$$\begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ \frac{z}{z/d} \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \\ 1 \end{bmatrix}$$

26

MANCHESTER 1824

Perspective matrices: summary

- We derived the matrix to perform a 1-point projection where the projection plane is parallel to the XY plane.

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d_z & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

We've renamed **d** to $\mathbf{d_z}$ just to remind us that it's a z-coordinate

- We can also derive (details omitted) a matrix which expresses the most general case, **3-point projection**, where the projection plane is not parallel to any of the XY, XZ or YZ planes:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1/d_x & 1/d_y & 1/d_z & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

The projection plane intersects the X,Y and Z axes at ($\mathbf{d_x}$ $\mathbf{d_y}$ $\mathbf{d_z}$)

27

MANCHESTER 1824

Projections: summary

- Parallel orthographic projection onto z=0 plane

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- 1-point perspective projection onto XY plane

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Then divide by w' to give 3D point

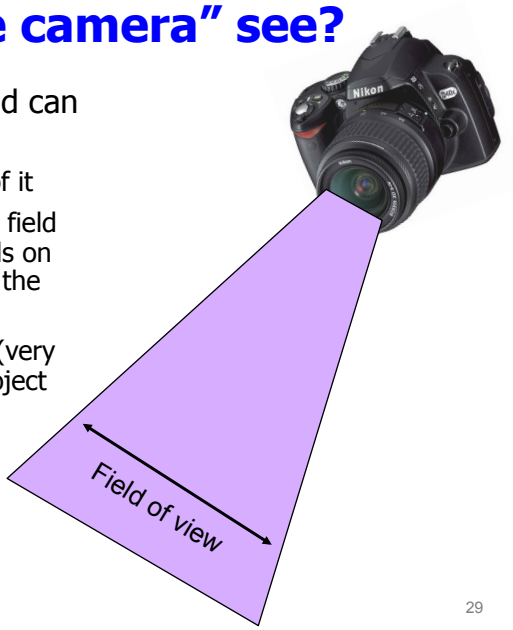
$$\begin{bmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \\ 1 \end{bmatrix}$$

28

MANCHESTER 1824

What can "the camera" see?

- What part of the world can a "camera" see?
  - Only objects in front of it
  - Only objects within its field of view, which depends on its lens (about 90° for the human eye)
  - Only a finite distance (very distant objects will project too small to be seen)



29

MANCHESTER 1824

Viewing volumes and clipping

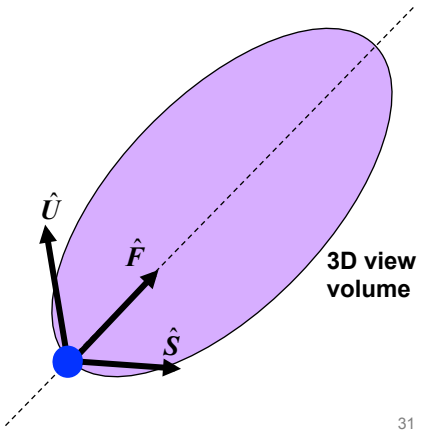
- So, having specified the position and orientation of the camera (see 3D Viewing (1))…
- …we now describe the field of view of the "camera" by defining a 3D "view volume"
- Only those parts of 3D objects which lie inside the view volume are displayed – all objects are **clipped** against the view volume
- We'll look at clipping shortly, but first we'll see how the view volumes are defined.

30

MANCHESTER 1824

Defining the view volume

- In **Viewing (1)** we established a coordinate system for the "camera" – with axes $\hat{S}\,\hat{U}\,\hat{F}$
- We define a 3D **view volume** which is "attached" to the camera
- As we would expect, the shapes of the volumes for parallel and perspective projection are different.

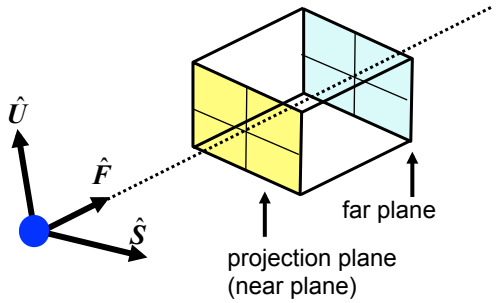


31

MANCHESTER 1824

View volume for parallel projection

- A view volume is a 3D shape, defined by six planes
- For parallel orthographic projection, it's a **cuboid**
- The cuboid is defined by a near plane (the projection plane) and a far plane, and top/bottom, left/right planes
- The near and far planes are orthogonal to the camera's $\hat{F}$ axis

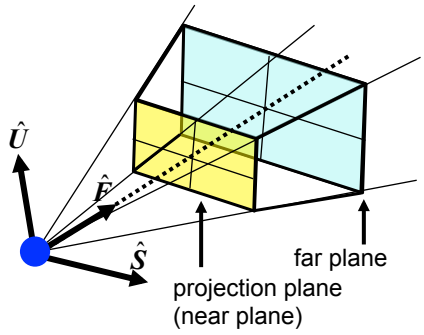


32

MANCHESTER 1824

View volume for perspective projection

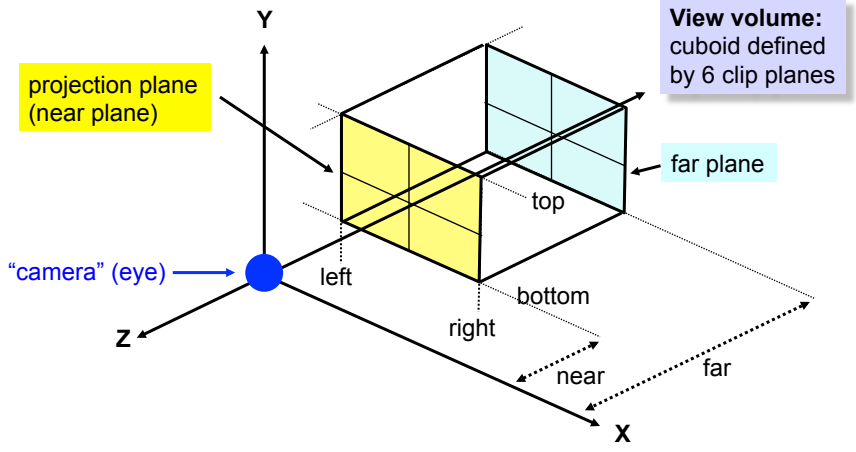
- For perspective projection, this view volume is a **frustum**, a truncated pyramid
- The frustum is defined by a near plane (the projection plane) and a far plane
- The near and far planes are orthogonal to the camera's $\hat{F}$ axis



33

MANCHESTER 1824

Orthographic projection in OpenGL



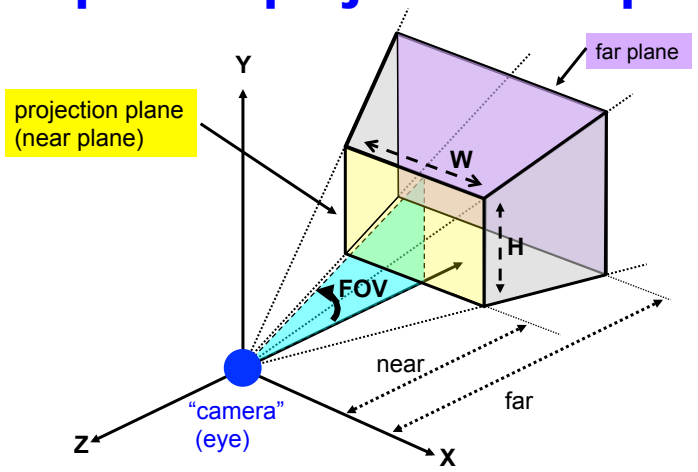

```
glOrtho (GLdouble left, GLdouble right,
         GLdouble top, GLdouble bottom,
         GLdouble near, GLdouble far);
```

OpenGL

34

MANCHESTER 1824

Perspective projection in OpenGL




```
gluPerspective (GLdouble fovy,
                GLdouble aspect,
                GLdouble near, GLdouble far);
```

OpenGL

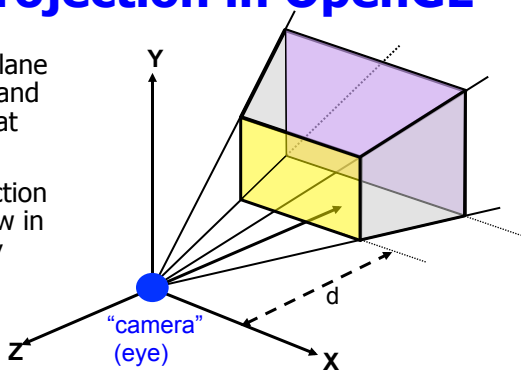
35

MANCHESTER 1824

Perspective projection in OpenGL

- Suppose the projection plane (near plane) is at z= -d, and the eye (or "camera") is at (0,0,0)
- Then the (1-point) projection transformation, as we saw in **Viewing (2)** is given by

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$



- Note: this projection transformation is **independent** of the far plane – which is only for clipping

36

MANCHESTER 1824

Perspective: a problem

- A projection matrix transforms a 3D point to a 3D point with W <> 1
- For example, for a 1-point perspective projection:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ w_p \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

In this case $w_p = z/d$

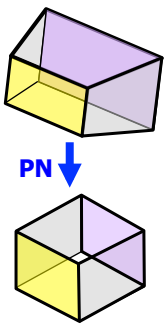
- So we need to divide through by w, to get (x_p, y_p, z_p)
- z_p = d, which means we have **lost** any idea of the original 3D object's z-coordinates – they've all been set to d.
- This is unhelpful, because we want to **keep** the z depth information, to do hidden-surface removal.

37

MANCHESTER 1824

Solution: Projection normalization

- We would like to have a perspective transformation which preserves depth.
- Suppose we have a particular perspective transformation expressed by frustum **F** and matrix **M**
- It can be shown that we can derive a new transformation matrix **PN**, based on **M**, that **distorts F into a cube**:
- Transforming our model by **PN**, and then taking an orthographic projection…
- …produces **exactly the same result** as performing our original perspective transformation **M**, with one difference: the z depth values are preserved
- This new technique is called **projection normalization**, and OpenGL creates **PN** for us automatically.

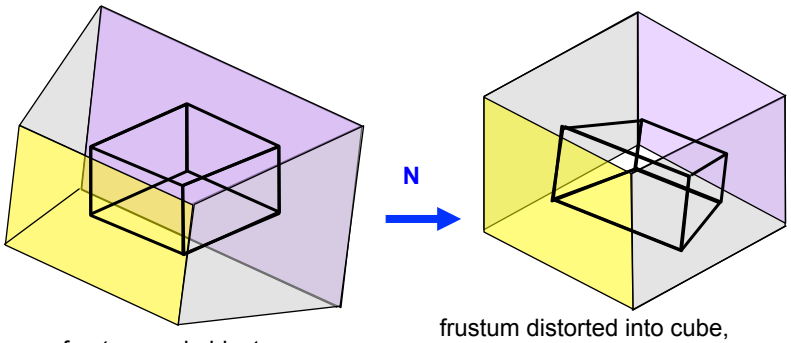


38

MANCHESTER 1824

Solution: Projection normalization

- **PN** distorts the perspective frustum to a unit cube
- An object inside the frustum is also distorted
- The distortion of the object is such that if we then apply an **orthographic projection**, we get the same view as if we had applied a perspective projection to the original undistorted object



frustum and object

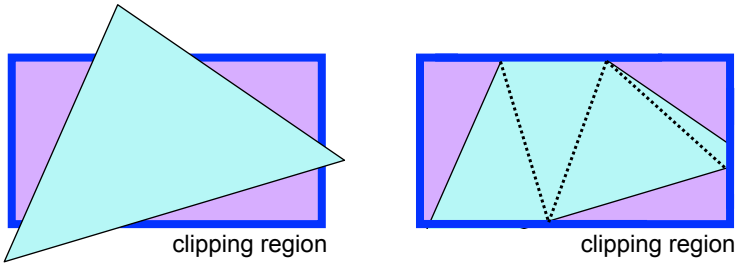
frustum distorted into cube, and object distorted

39

MANCHESTER 1824

The clipping operation

- Clipping takes place in the cube produced by projection normalization (easier than clipping against a frustum)
- Clipping is easiest to visualise in 2D
- Here we see polygon being clipped against a rectangular clipping region



clipping region

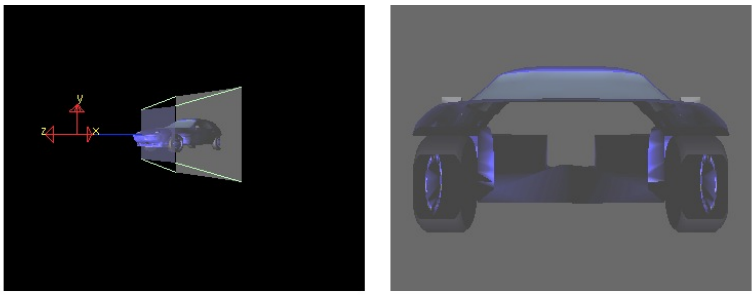
clipping region

- There are established algorithms for clipping in 2D and 3D

40

MANCHESTER 1824

Viewing volume and clipping



Command manipulation window

```
                fovy  aspect  zNear  zFar
gluPerspective( 36.0  , 1.00  , 1.3   , 2.6   );
    gluLookAt( 0.00   , 0.00  , 2.00  ,   <- eye
               0.00   , 0.00  , 0.00  ,   <- center
               0.00   , 1.00  , 0.00  );  <- up
```

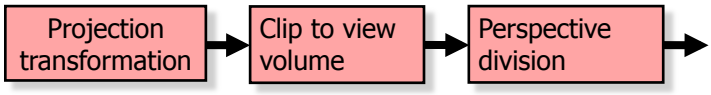
Click on the arguments and move the mouse to modify values.

1

MANCHESTER 1824

Perspective division

- The clipping operation returns a set of (x,y,z,w) vertices defining polygons which are inside the view volume
- OpenGL now performs the **perspective division by w** to convert these (x,y,z,w) values to (x,y,z) 3D points



42

MANCHESTER 1824

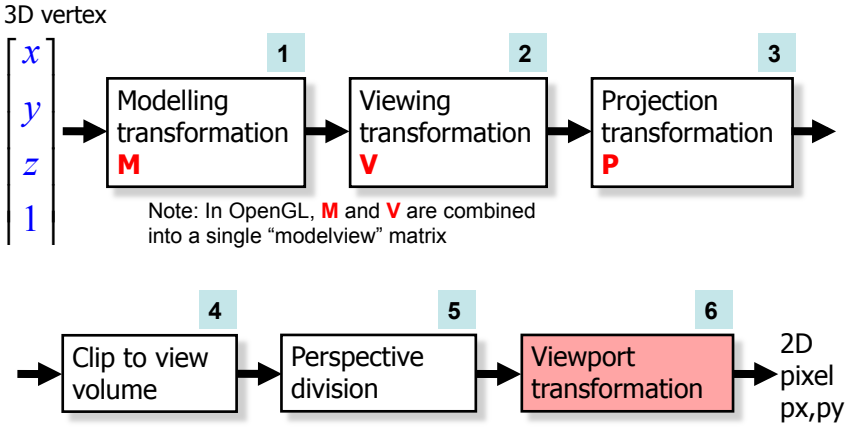
3D Viewing: the camera analogy

	Real World	Computer Graphics
1	Arrange the scene into the desired composition	Set Modelling Transformation
2	Point the camera at the scene	Set Viewing Transformation
3	Choose the camera lens, or adjust the zoom	Set Projection Transformation
4	Determine the size and shape of the final photograph	Set Viewport Transformation

43

MANCHESTER 1824

The 3D Viewing Pipeline

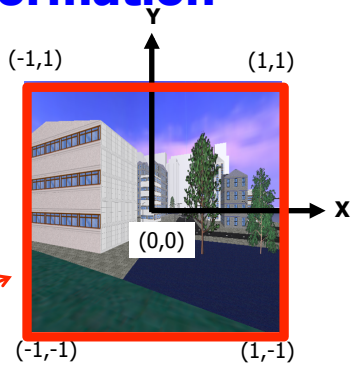


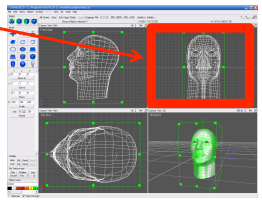
44

MANCHESTER 1824

The Viewport Transformation

- After perspective division, we now have 3D coordinates
- OpenGL has scaled these into the range [-1,+1] in X, Y and Z
- In the final step, using only the X and Y values, OpenGL computes a transformation from [-1,+1] to the display screen
  - Either the whole **window**
  - Or a part of it called a **viewport**
- **But we retain Z**, to remove hidden surfaces





45

MANCHESTER 1824

Summary of 3D viewing

- We can now summarise the steps of the viewing process

1. The modelling transformation arranges objects in our 3D world
2. The viewing transformation transforms the world to give the same view as if it were being photographed by a camera
3. The projection transformation performs a parallel/perspective projection within limits (the clip planes)
4. Those parts of the 3D world outside the clip planes are discarded
5. If it's a perspective view, the perspective division "flattens" the image
6. The viewport transformation maps the final image to a position in part of the display screen window.

46