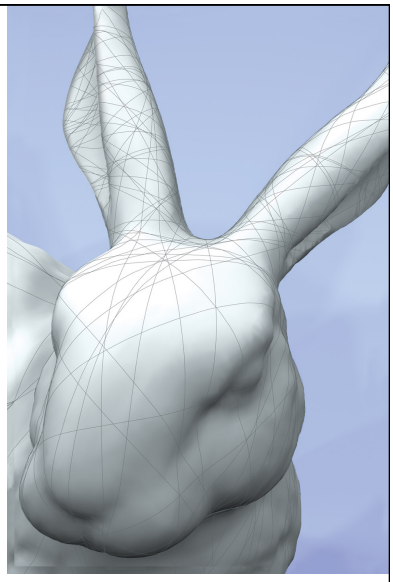
MANCHESTER 1824

COMP27112

Computer Graphics and Image Processing



1: Introduction to the course

1

MANCHESTER 1824

About the course

Toby Howard
(course leader)

toby.howard@manchester.ac.uk

Tim Morris

tim.morris@manchester.ac.uk

Lectures: Monday 14:00, Thursday 14:00

in Kilburn 1.1 podcasted

2

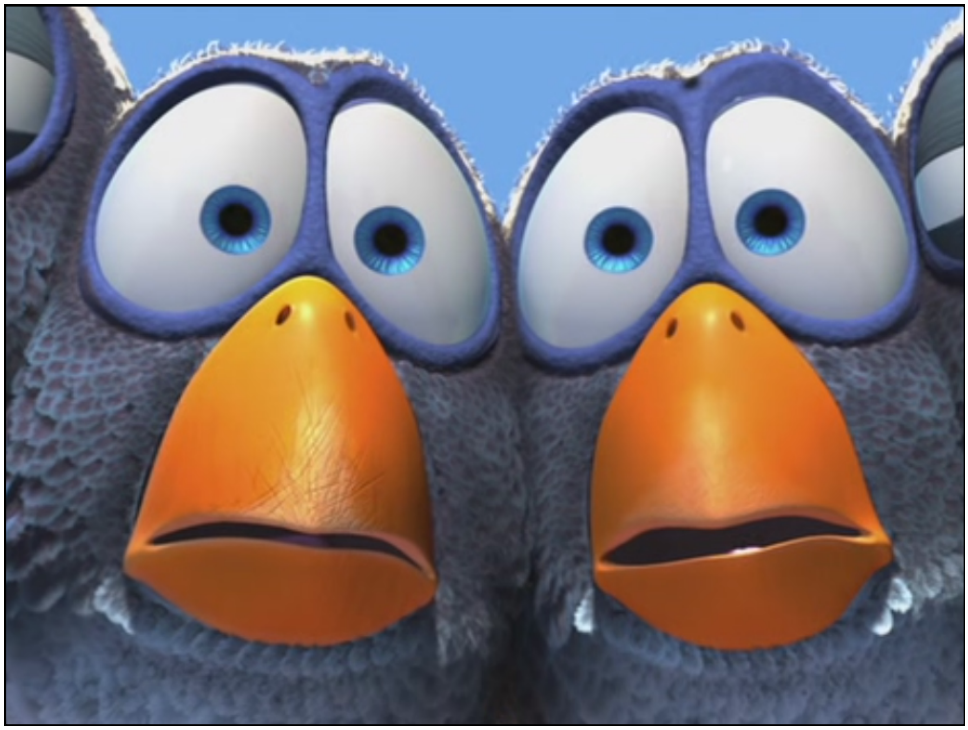
MANCHESTER 1824

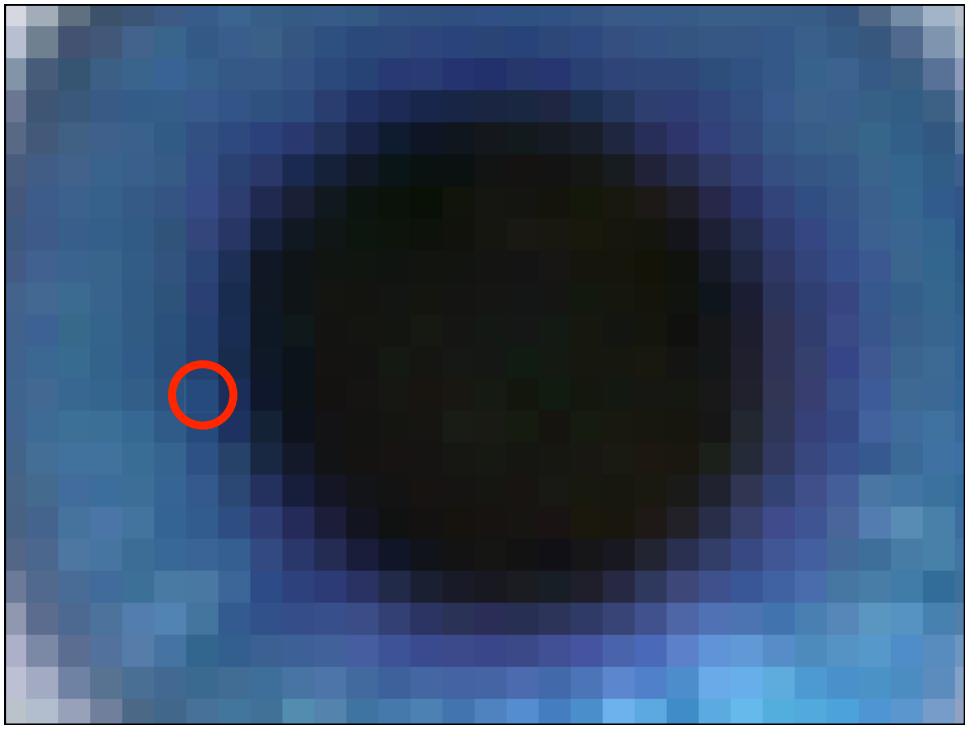
Aims and objectives

- **Aims**
  - to introduce the **theory and practice** of interactive 3D computer graphics, and image processing
- **Objectives**

  To understand:
  - the principles of interactive computer graphics (OpenGL)
  - the mathematics of 3D transformations and viewing
  - the principles of the rendering pipeline
  - the principles of image processing

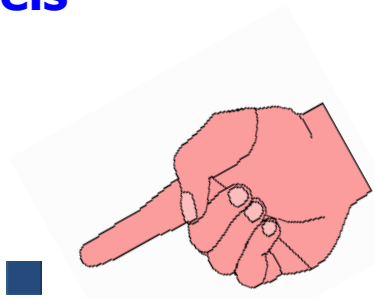
3





MANCHESTER 1824

Pixels



- On an 23” screen at 1280 x 1024 …
- … a pixel is about 0.2mm, or 0.01”

6

MANCHESTER 1824

Pixels in a 90 minute movie

- a movie frame is 4K x 4K pixels
- 24 frames/sec
- movie is 90 min

- Total: about 1,000,000,000,000 pixels
- 10^{12} pixels

7

MANCHESTER 1824

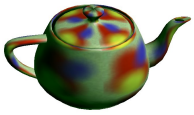
Course activities

- Lectures & demonstrations
- Labs (4 exercises) in **Week A**
  - TAs, attendance, submit by deadline (no extensions)
- Coursework (5 x 1-hour exercises) in **Week B**
  - Personal study time, no TAs, submit by deadline (no extensions)
- 2-hour online exam: 20 MCQs + essay-type Qs

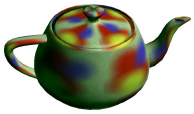
8

MANCHESTER 1824

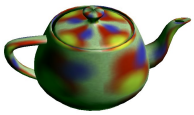
Assessment structure



Labs: 20%



Coursework: 5%



Exam: 75%

9

MANCHESTER 1824

Labs

- one lab every 2 weeks
- begin week of 13 February (Week 3A)

- 4 different lab exercises
  - we give you skeleton **C** code
  - deliverables are marked
  - code is **`submitted`**

- labs use **Linux**, **C, OpenGL**

10

MANCHESTER 1824

Lab schedule (Week A)

Lab G: Thu 0900-1100 in LF31

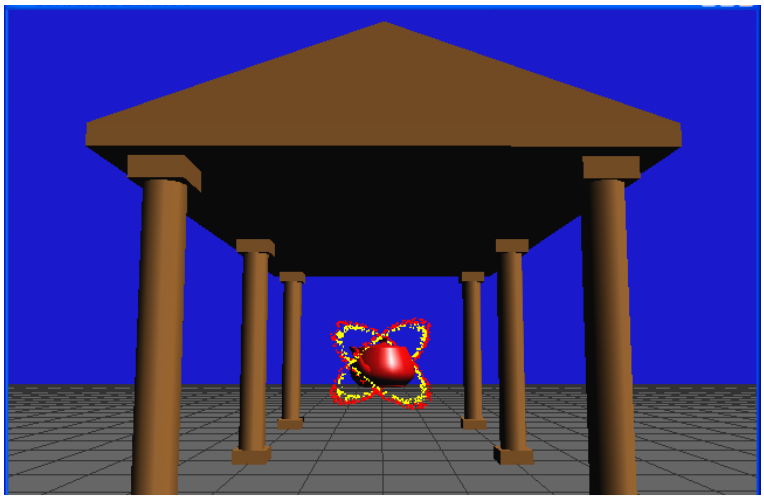
Lab H: Tue 1100-1300 in G23

Lab I: Fri 1100-1300 in LF31

Full timetable on course Blackboard page

MANCHESTER 1824

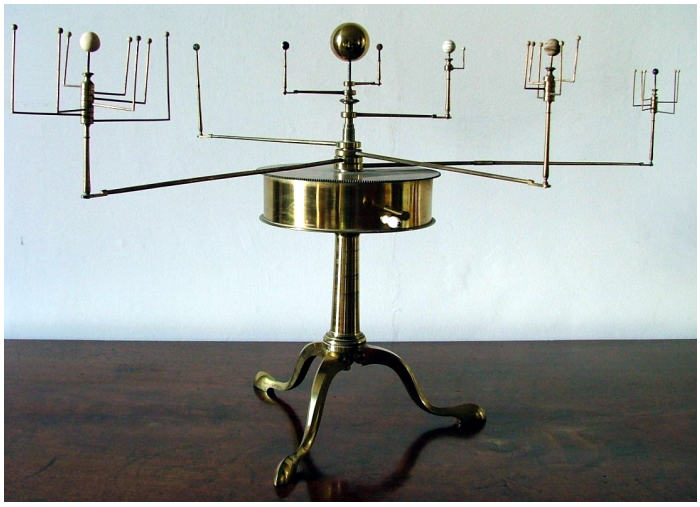
Lab 1 (1 session)



- 3D graphics: transformations and viewing

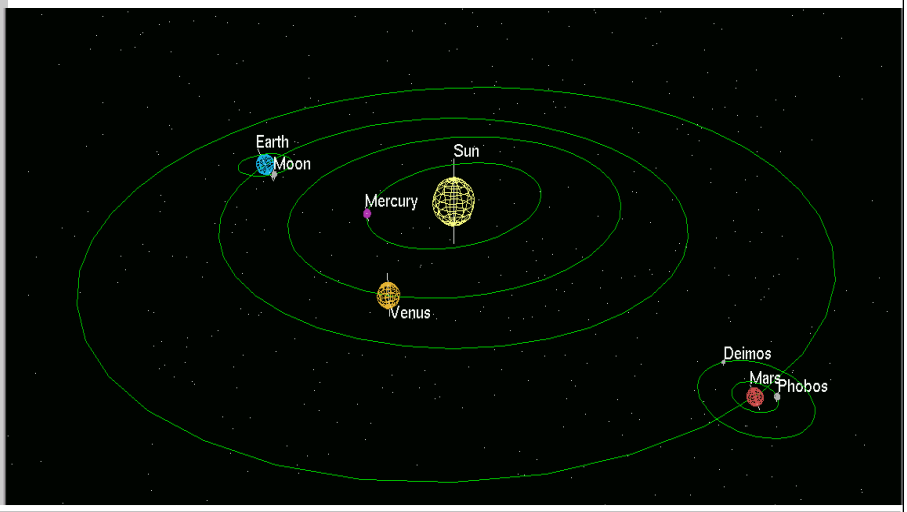
MANCHESTER 1824

Lab2: build an orrery



MANCHESTER 1824

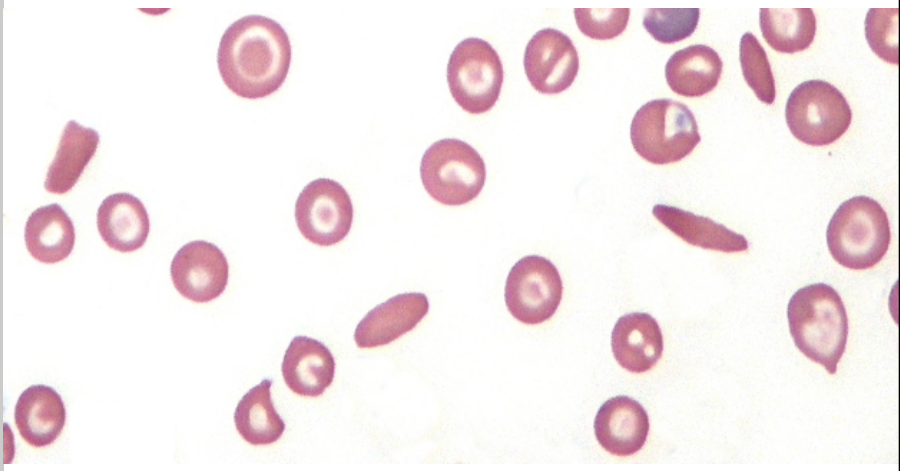
Lab 2 (2 sessions)



- Animated graphics with hierarchical transformations

MANCHESTER 1824

Lab 3 (1 session)

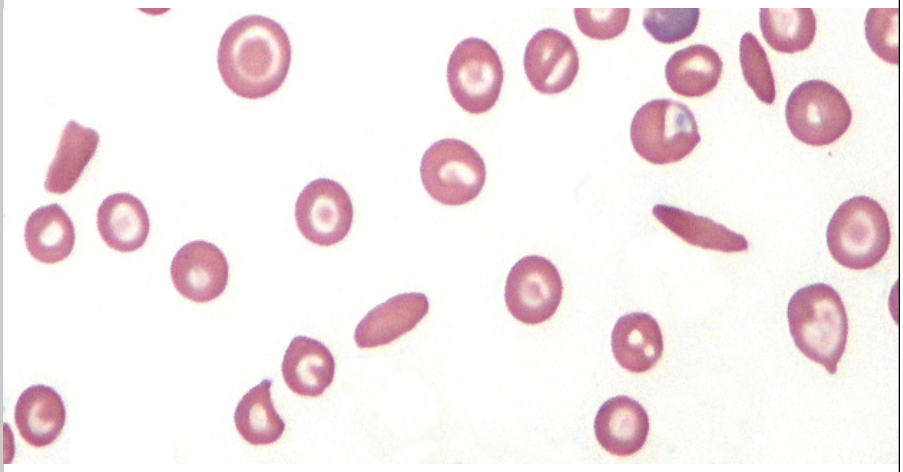


- Pixel processing

15

MANCHESTER 1824

Lab 4 (1 session)



- Connected component labelling

16

MANCHESTER 1824

Coursework

- Coursework Assignments due every 2 weeks (alternating with labs)
- Begin week of 6 February (Week 2B)
- 5 different Assignments
- hands-on experiments, small bit of coding

- `submit` results, marks for all 5 Assignments **add up to one Lab exercise**

17

MANCHESTER 1824

Coursework (week B)

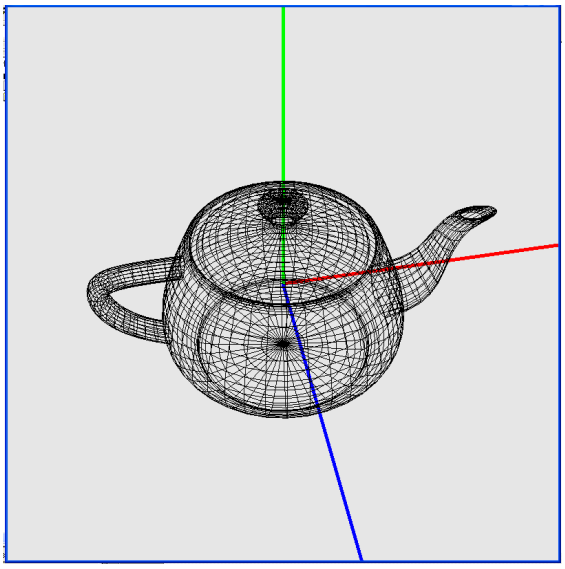
	Recommended start-time (but up to you)	Submission Deadline 17:00 (no extensions)
1	Mon 6 Feb	Fri 17 Feb
2	Mon 20 Feb	Fri 3 Mar
3	Mon 6 Mar	Fri 17 Mar
4	Mon 20 Mar	Fri 31 Mar
5	Mon 24 Apr	Fri 5 May

18

MANCHESTER 1824

Coursework 1

- Intro to OpenGL

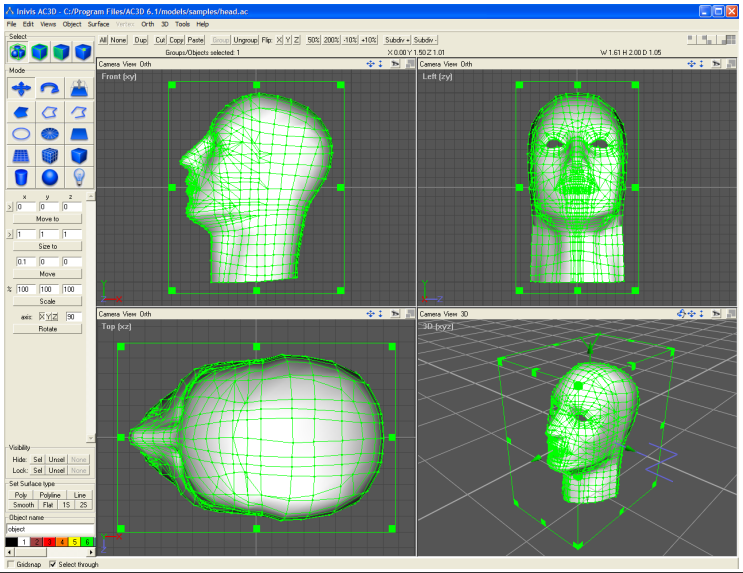


19

MANCHESTER 1824

Coursework 2

- 3D modelling

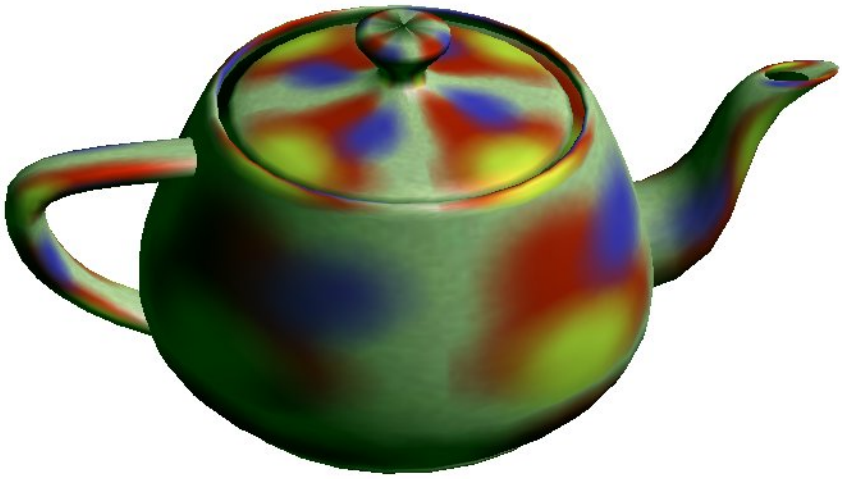


20

MANCHESTER 1824

Coursework 3

- Lighting and texture

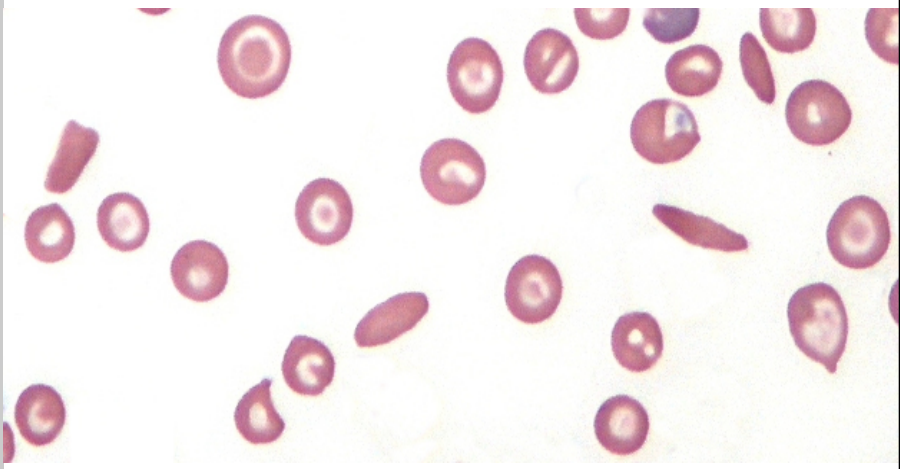


21

MANCHESTER 1824

Coursework 4

- Read/modify/save images

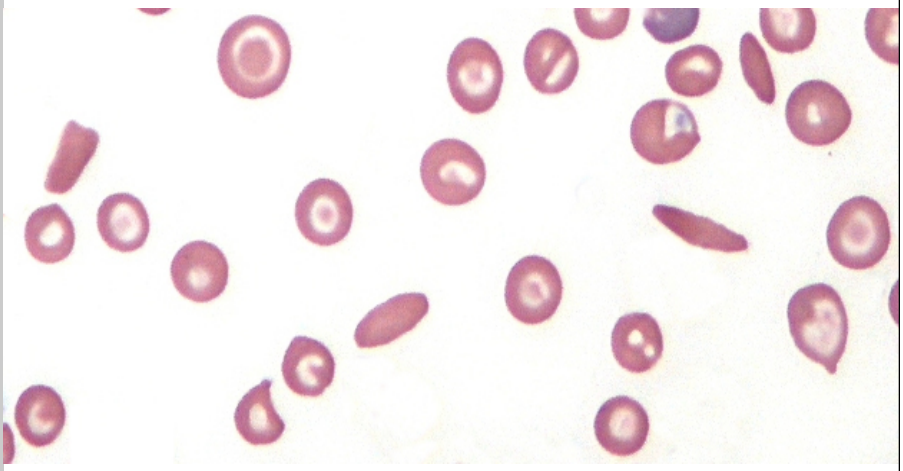


22

MANCHESTER 1824

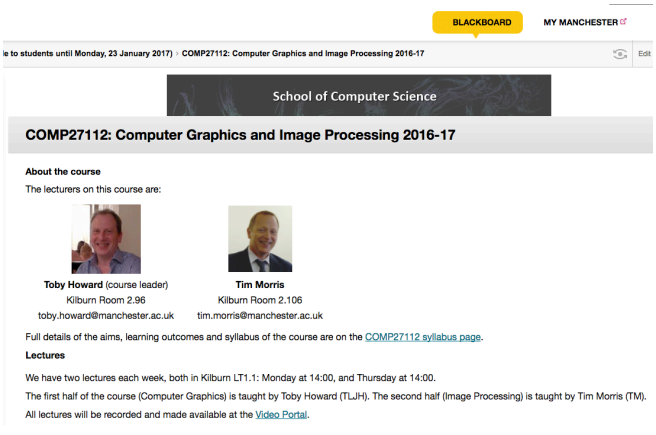
Coursework 5

- Region-based processing



MANCHESTER 1824

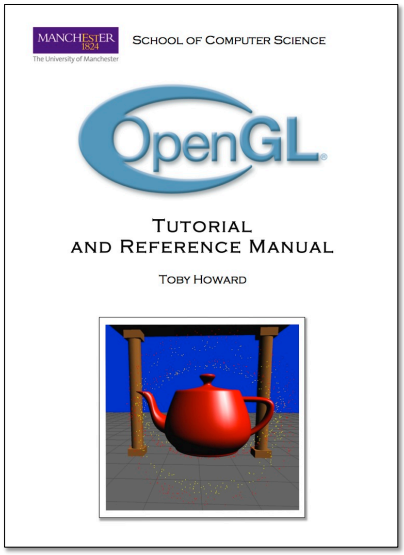
Essential resources



Blackboard

MANCHESTER 1824

Essential doc: OpenGL



- **OpenGL tutorial and reference manual**
- Collect from the Student Resource Centre after this lecture

25

MANCHESTER 1824

Essential documentation

OpenGL support

This page offers some support for using OpenGL in the School of Computer Science at The University of Manchester (so contains some site-specific information). The page is maintained by Toby Howard. Please let me know if you have any comments or suggestions about what would be useful here.

Running OpenGL on various platforms

- **GNU/Linux:**
  - If you have just one source file, use **/opt/common/bin/cogl** on the School network. This is a Perl script designed for the School's Linux configuration.
  - Also for one source file, Karl Sutt has kindly contributed a more portable shell-script version of cogl called ucogl.
  - For more complicated projects, use a Makefile. Here's one to get you started.
- **Mac:** First install Xcode (free from here). Then, compile using:

```
gcc -o teapot teapot.c -framework OPENGL -framework GLUT
```

Note: on the Mac your GLUT include needs to be: #include <GLUT/glut.h>. It's good

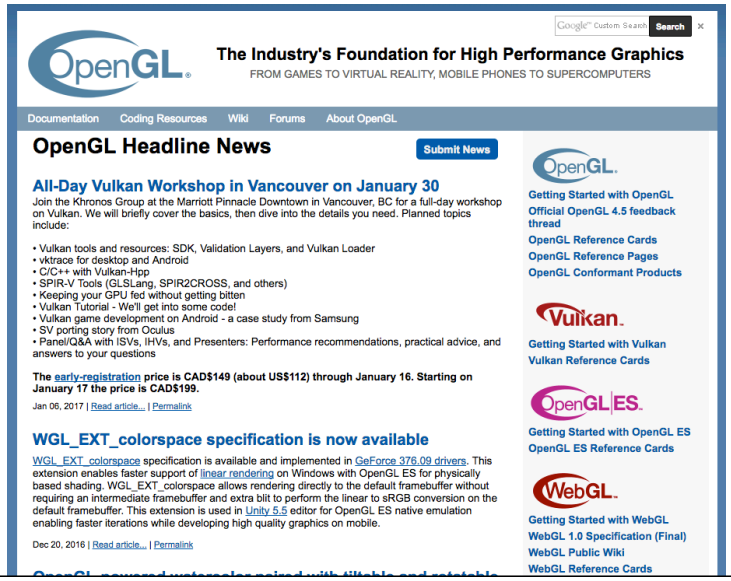
http://studentnet.cs.manchester.ac.uk/ugt/COMP27112/OpenGL/

- **Windows:** Here are instructions for Visual Studio and Visual C++. Also, this project zip may be

26

MANCHESTER 1824

Essential doc: OpenGL home



27

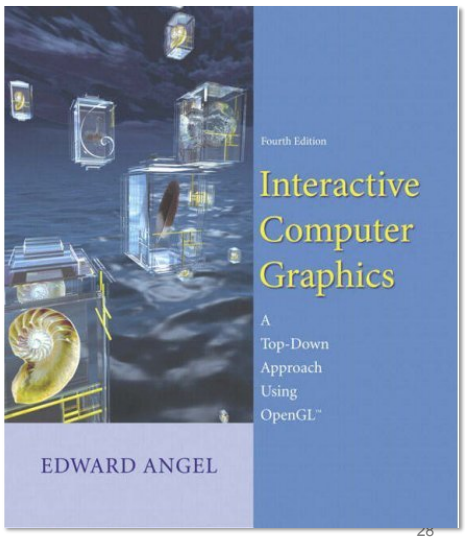
MANCHESTER 1824

Course books (not essential)

Interactive Computer Graphics: A top-down approach with OpenGL

Edward Angel

Addison-Wesley
ISBN 0-201-38597-7



28

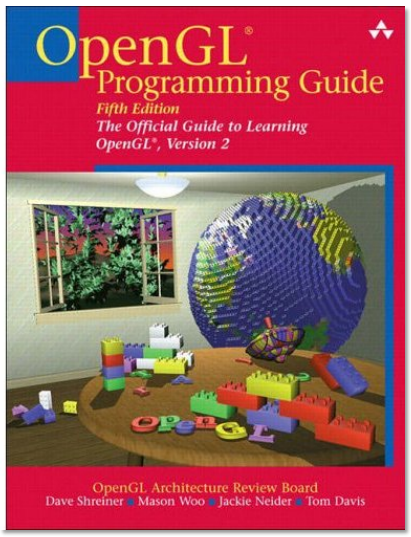
MANCHESTER 1824

Course books (not essential)

OpenGL Programming Guide (5th-8th Editions)

Mason Woo, Jackie Neider, Tom Davis et al

Addison-Wesley
ISBN 0321481003



29

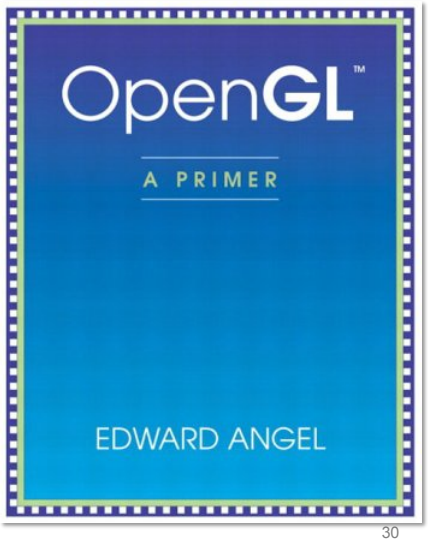
MANCHESTER 1824

Course books (not essential)

OpenGL: A Primer

Edward Angel

Addison-Wesley
ISBN 0201741865



30

MANCHESTER 1824

Context

Why are computer graphics and image processing important?

31

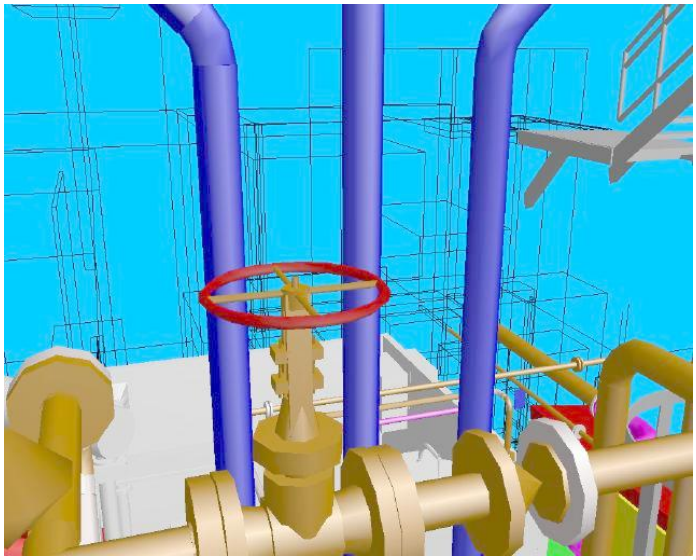
MANCHESTER 1824

Computer graphics applications

32

MANCHESTER 1824

Engineering



33

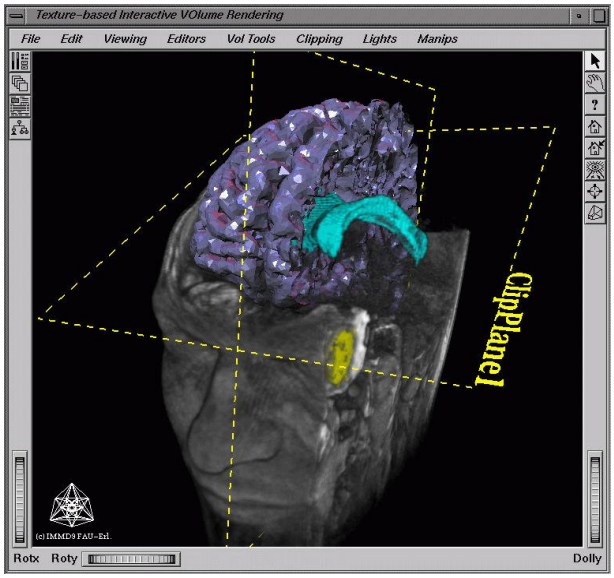
MANCHESTER 1824

Visualization

34

MANCHESTER 1824

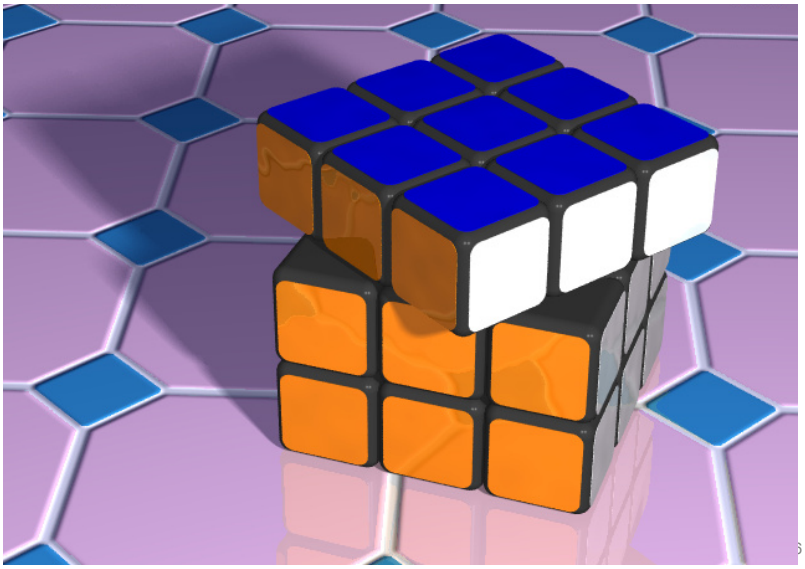
Visualization



35

MANCHESTER 1824

Simulation



MANCHESTER 1824

VR guy

Interactive virtual environments

$800

HTC Vive

$700

PlayStation VR

$600

Oculus Rift

$100

Google Daydream

$70

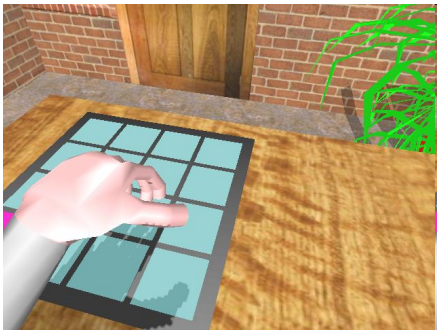
Samsung Gear

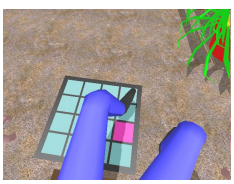
$15

Google Cardboard

MANCHESTER 1824

Virtual Environments for Therapy

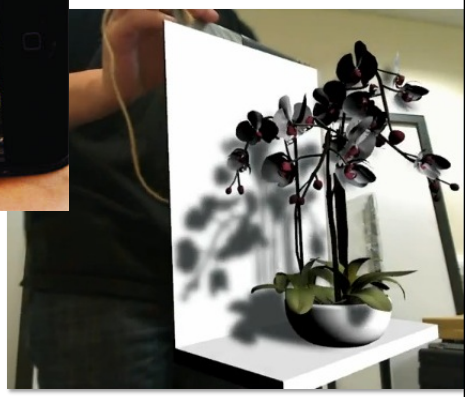




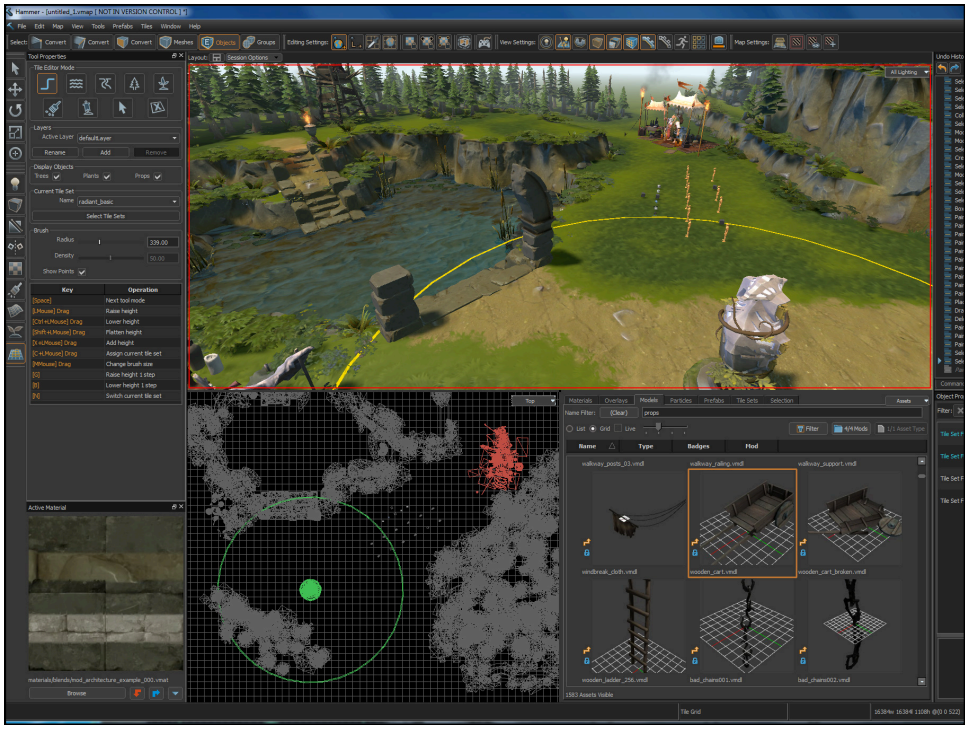
38

MANCHESTER 1824

Augmented reality



39



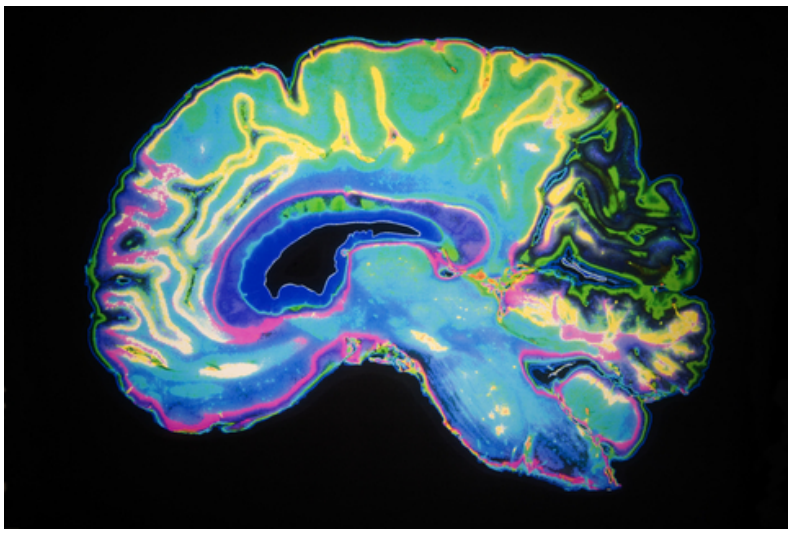
MANCHESTER 1824

Image processing applications

42

MANCHESTER 1824

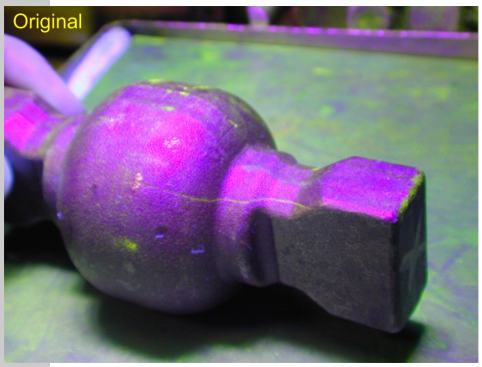
Medical

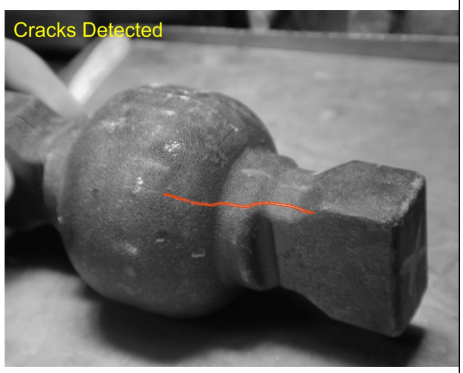


43

MANCHESTER 1824

Engineering

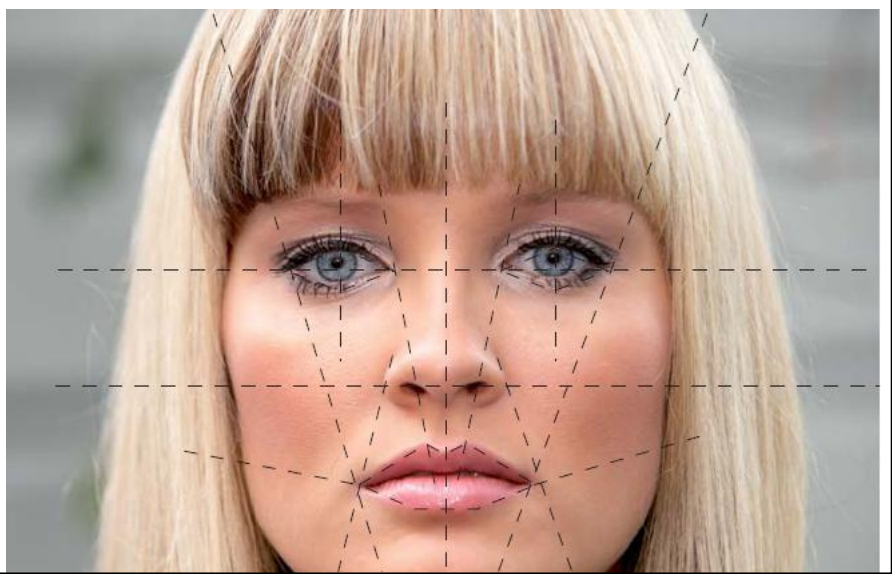




44

MANCHESTER 1824

Face recognition



MANCHESTER 1824

Photography/video/media

MANCHESTER 1824

COMP27112 syllabus

- The course is in two parts

1. Interactive Computer Graphics (Toby) – **"synthetic"**

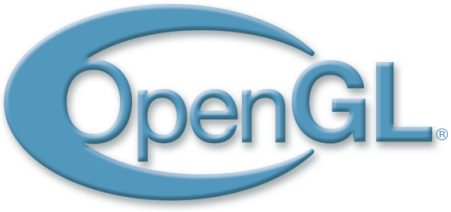
2. Image Processing (Tim) – **"analytic"**
   - image transformations
   - filters, convolution, finding edges
   - noise reduction, smoothing

47

MANCHESTER 1824

OpenGL

- History
- Basic concepts
- Program structure
- Demos

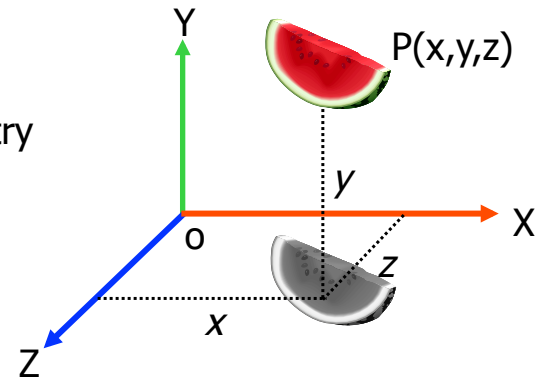


48

MANCHESTER 1824

3D coordinate systems

- 3D geometry
- 3D trigonometry
- Homogeneous matrices



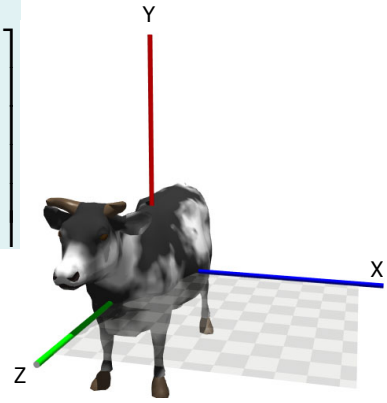
49

MANCHESTER 1824

A scaling matrix

$$P' = T \cdot P$$

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



50

MANCHESTER 1824

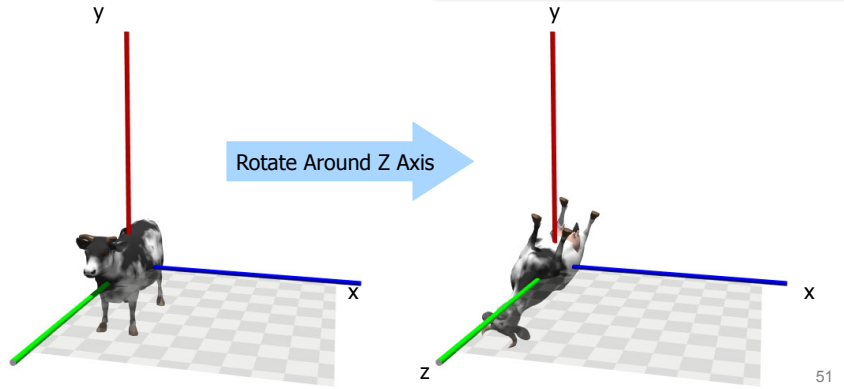
Rotation

Fundamental rotations are applied about the 3 major axes, e.g. a rotation angle θ about the Z axis

$$x' = x \,.\, \cos\theta \; - y \,.\, \sin\theta$$

$$y' = x \,.\, \sin\theta \; + y \,.\, \cos\theta$$

$$z' = z$$

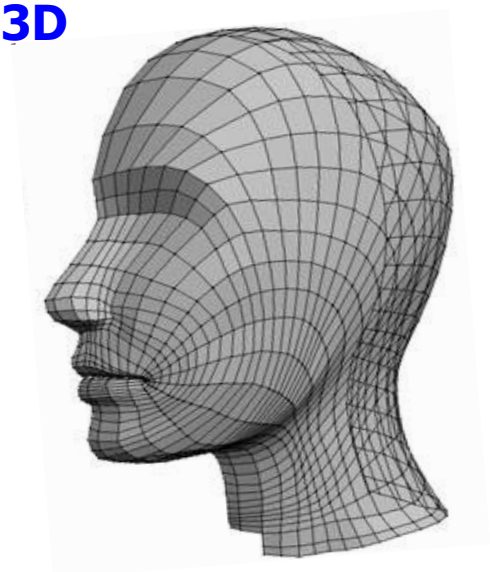


51

MANCHESTER 1824

Modelling in 3D

- Representing geometry
- Polygons, meshes, surfaces
- Managing complexity
- Model formats



52

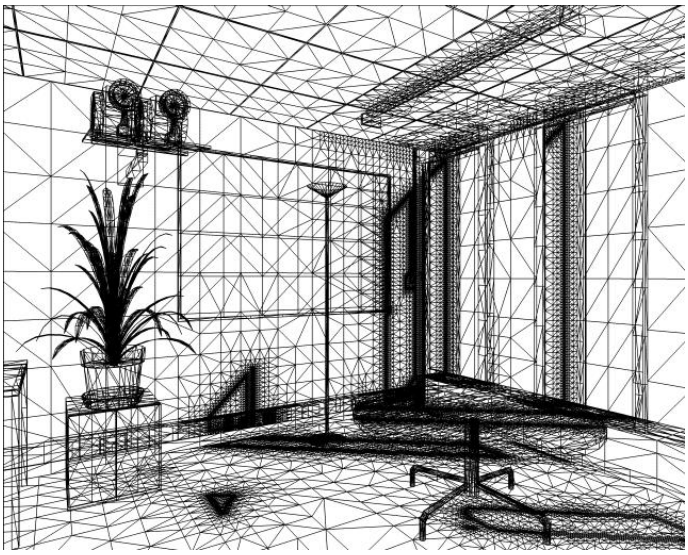
MANCHESTER 1824

Polygon meshes

53

MANCHESTER 1824

Polygon meshes

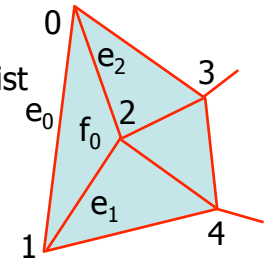


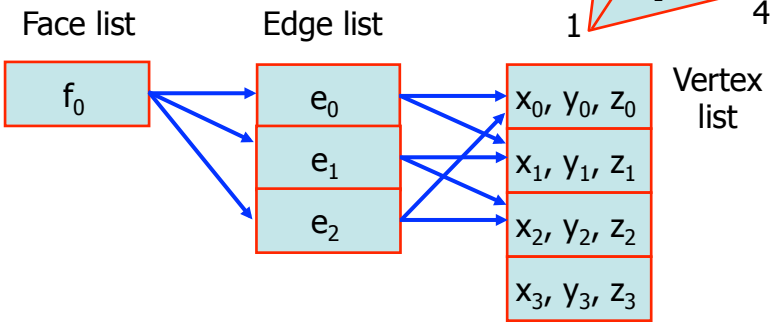
54

MANCHESTER 1824

Mesh data structure

- **vertex list**
- **edge list**, indexing into the **vertex** list
- **face list**, indexing into the **edge** list

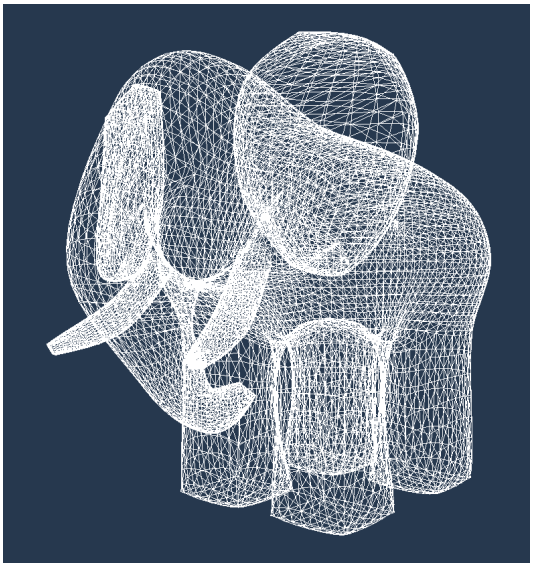




55

MANCHESTER 1824

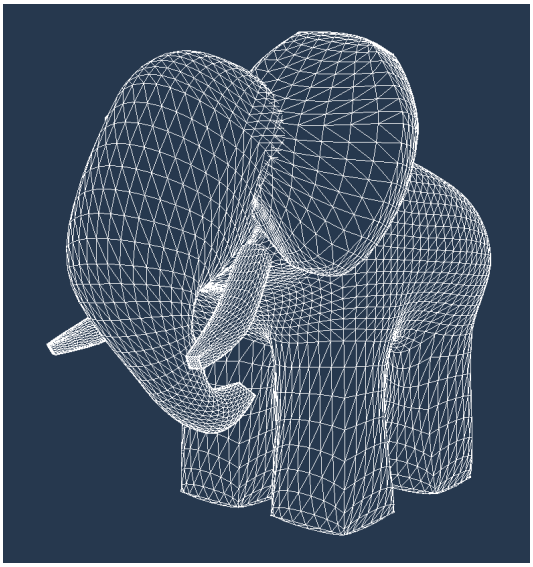
Hidden-surface removal



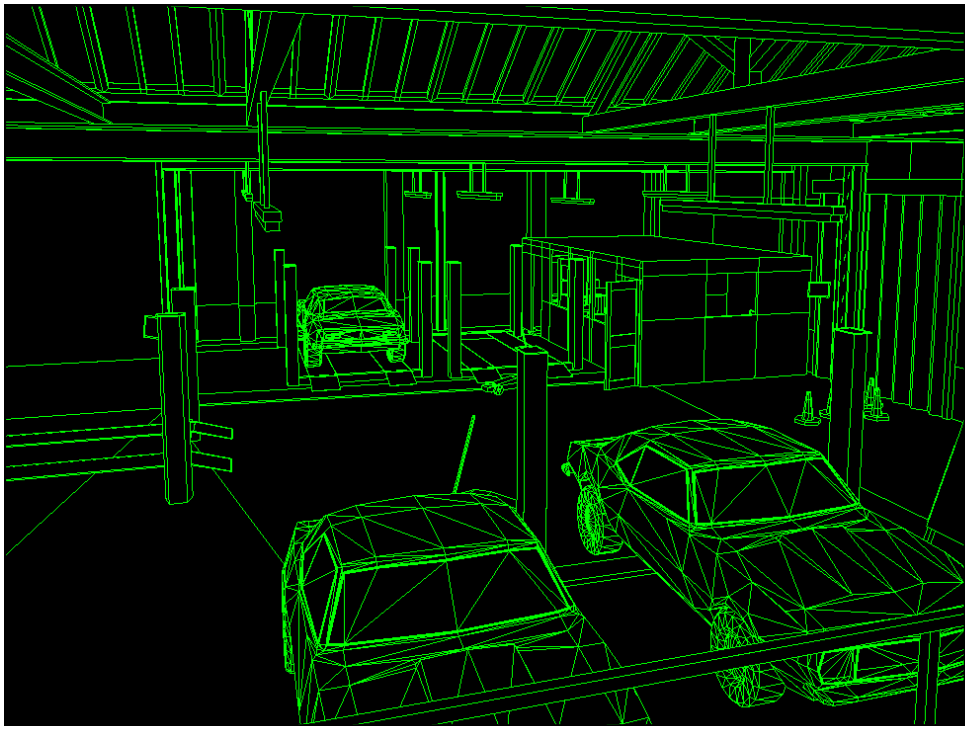
56

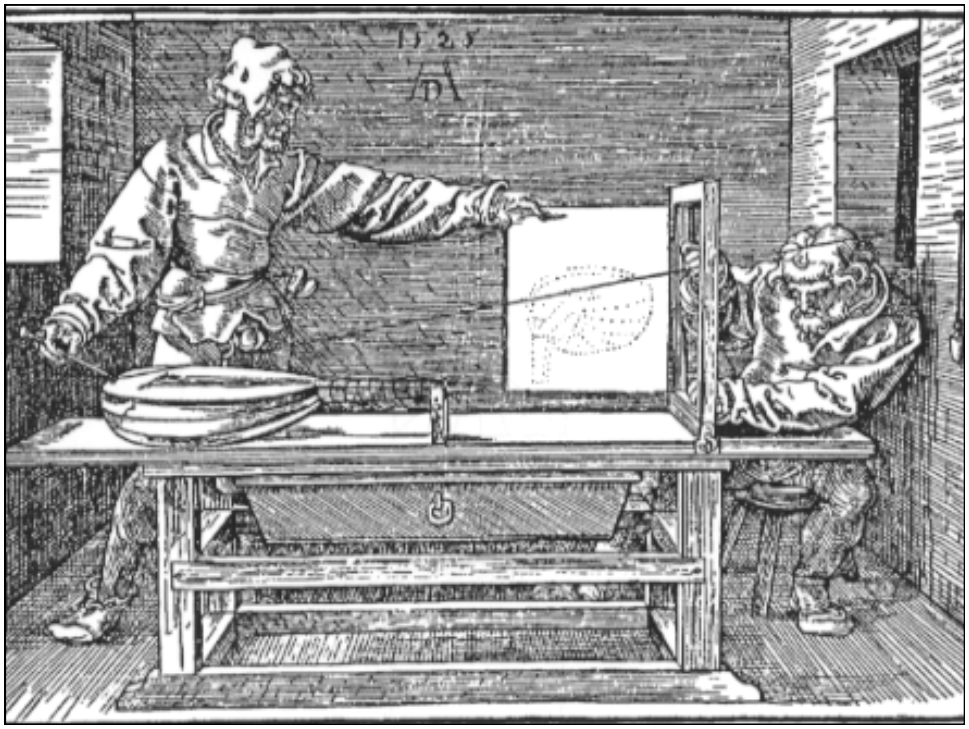
MANCHESTER 1824

Hidden-surface removal



57





MANCHESTER 1824

The 3D Viewing Pipeline

3D vertex

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

1 Modelling transformation **M** → 2 Viewing transformation **V** → 3 Projection transformation **P** →

Note: In OpenGL, **M** and **V** are combined into a single "modelview" matrix (see later)

→ 4 Clip to view volume → 5 Perspective division → 6 Viewport transformation → 2D pixel px,py

In this lecture we'll cover steps 1 and 2, and look at steps 3 to 6 in the next lecture.

60

MANCHESTER 1824

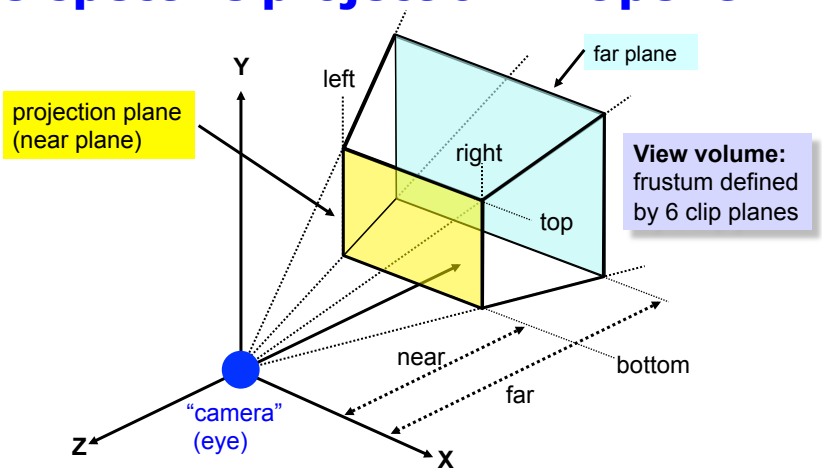
The Camera Analogy

Step 1: Set up your tripod and point your camera at the scene

61

MANCHESTER 1824

Perspective projection in OpenGL



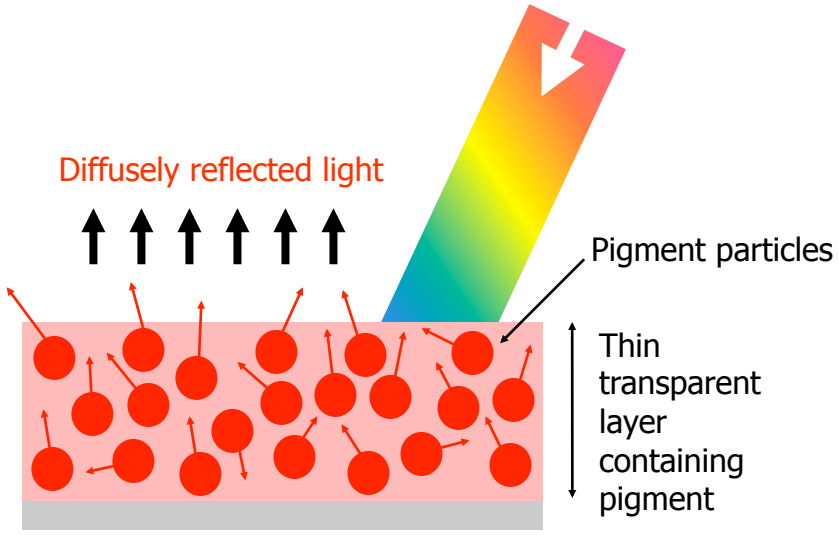

```
glFrustum (GLdouble left,   GLdouble right,
           GLdouble bottom, GLdouble top,
           GLdouble near,   GLdouble far);
```

OpenGL

62

MANCHESTER 1824

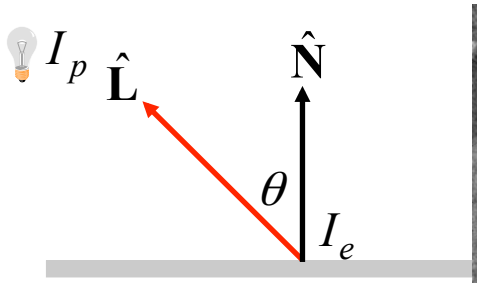
Diffuse light/surface interaction



63

MANCHESTER 1824

Diffuse reflection: Lambert's Law

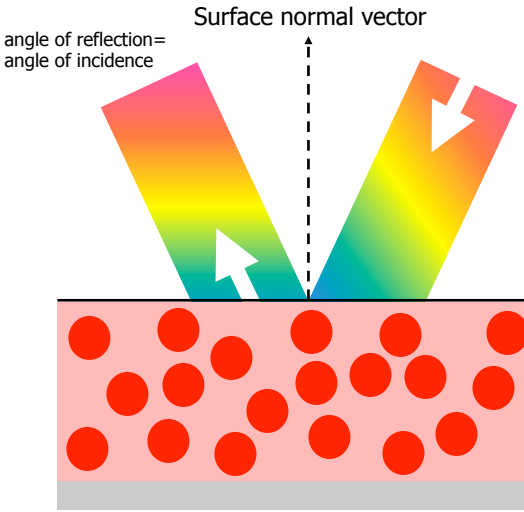


- Light source of intensity I_p
- Effective intensity received is I_e
- Lambert's Law: $I_e = I_p \cos\theta$
  - Johann Heinrich Lambert (1728-1777)

64

MANCHESTER 1824

Specular light/surface interaction



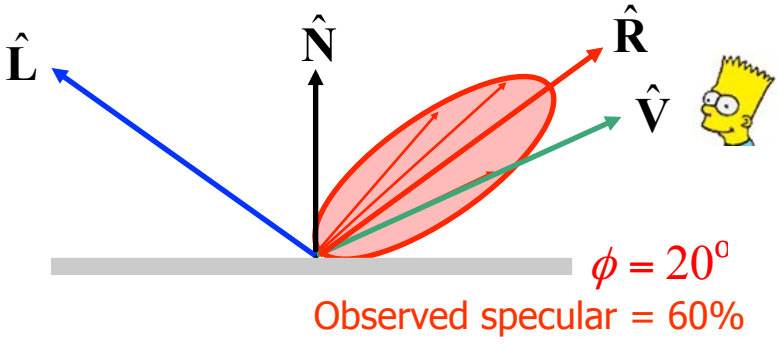
When reflection occurs at the air/surface interface, light is **specularly** reflected, with no change of colour

The pigment particles are not involved in the reflection

65

MANCHESTER 1824

Modelling specular reflection



Observed specular = 60%

- As $\hat{\mathbf{V}}$ diverges from $\hat{\mathbf{R}}$ by angle ϕ, viewer sees less specular reflection

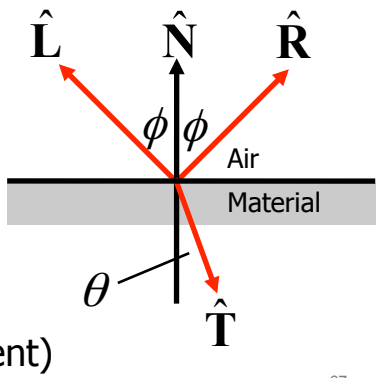
66

MANCHESTER 1824

Incident angle and wavelength

- This complex variation is expressed by the Fresnel equation (Augustin-Jean Fresnel, 1788-1827)

$$F = \frac{1}{2}\left[\frac{\sin^2(\phi-\theta)}{\sin^2(\phi+\theta)} + \frac{\tan^2(\phi-\theta)}{\tan^2(\phi+\theta)}\right]$$



- F is the fraction of light reflected
- $\sin\theta = \sin\phi / \mu$
- μ is the refractive index of the material (λ dependent)

67

MANCHESTER 1824

Shading and interpolation

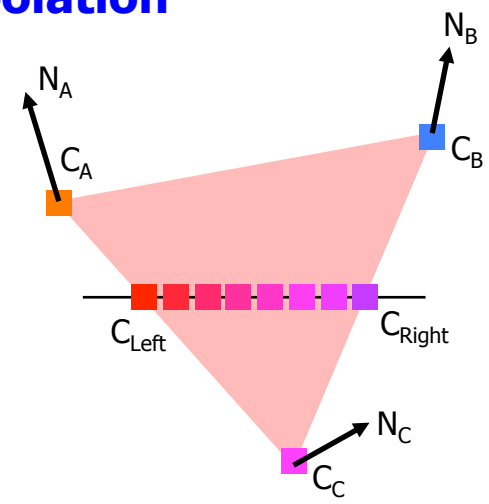
68

MANCHESTER 1824

Gouraud interpolation

- compute average vertex normals at A, B and C
- compute pixel colours $\mathbf{C_A}$, $\mathbf{C_B}$, $\mathbf{C_C}$
- for each scanline {
  - average colour $\mathbf{C_{Left}}$ from $\mathbf{C_A}$ and $\mathbf{C_C}$
  - average colour $\mathbf{C_{Right}}$ from $\mathbf{C_B}$ and $\mathbf{C_C}$
  - average between $\mathbf{C_{Left}}$ and $\mathbf{C_{Right}}$ along the scanline

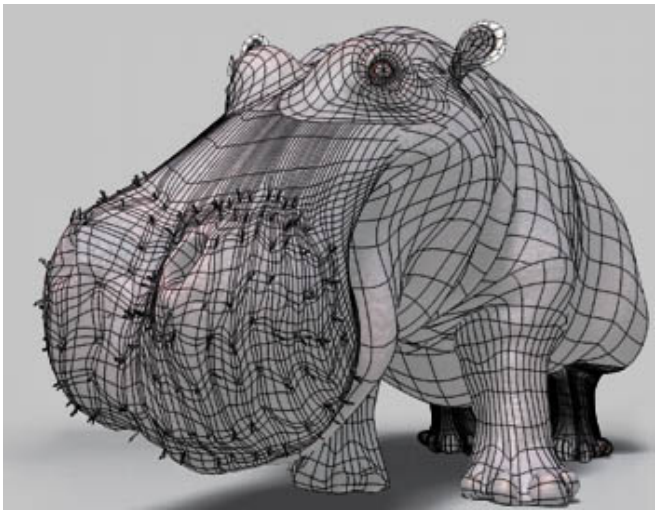
  }



69

MANCHESTER 1824

Texture mapping

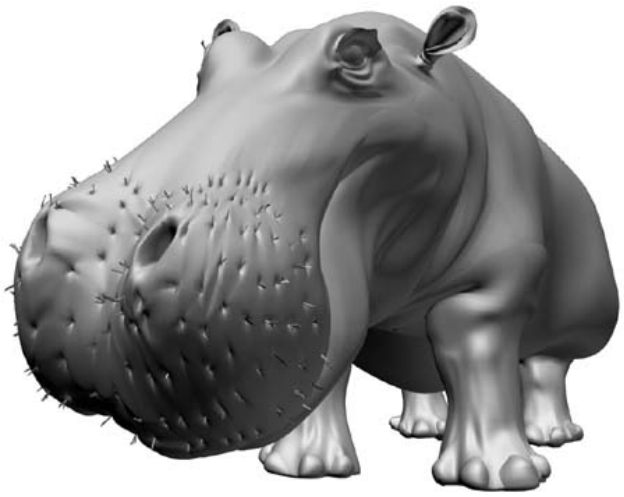


Jeremy Birn

70

MANCHESTER 1824

Texture mapping



Jeremy Birn

71

MANCHESTER 1824

Texture mapping

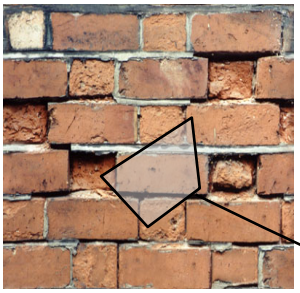
Jeremy Birn

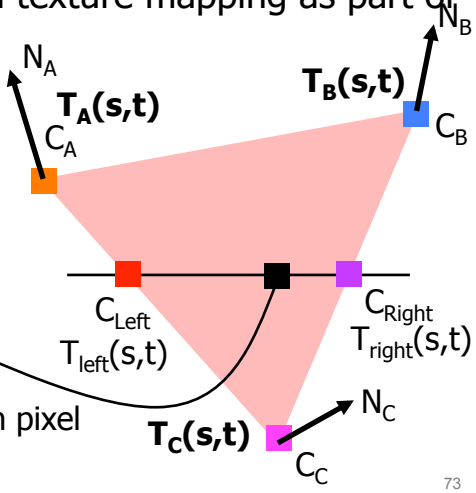
72

MANCHESTER 1824

Performing texture mapping

- We want to peform texture mapping as part of rasterisation





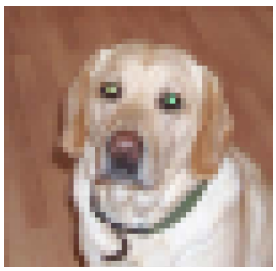
Sampling the texture at each pixel

73

MANCHESTER 1824

Texture filtering: comparison

Unfiltered texture (1 to 1 pixel correspondence)



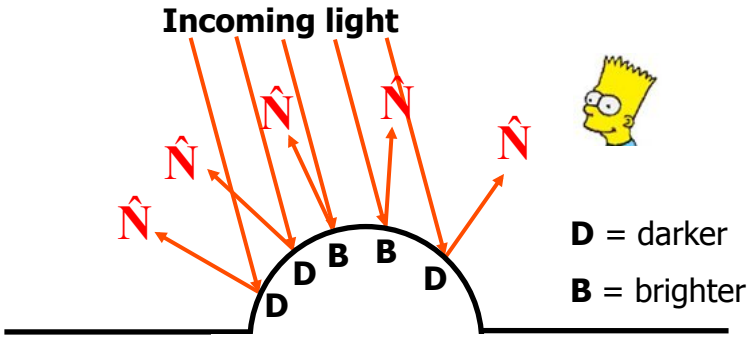
Simple filter

Bilinear interpolation filter

74

MANCHESTER 1824

Bump mapping

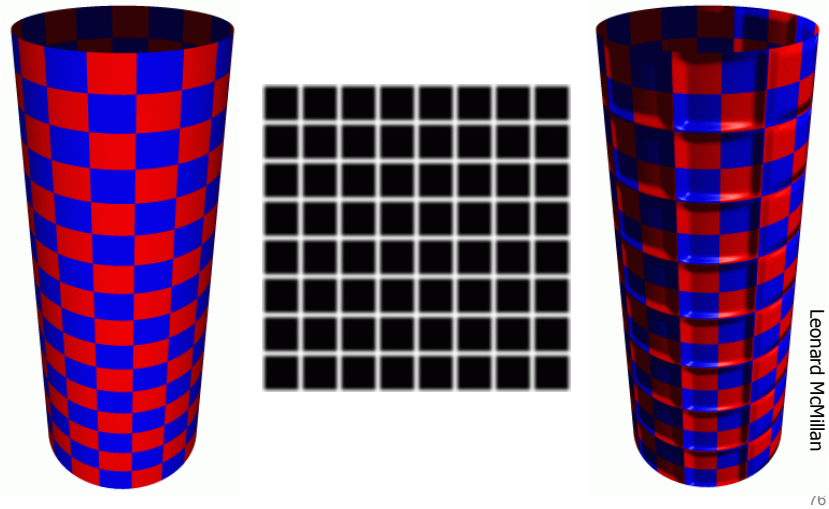


- Why bumpy surfaces look bumpy
  - The **surface normals** change across the bumps, so the top of the bumps appear brighter than the sides

75

MANCHESTER 1824

Bump mapping example

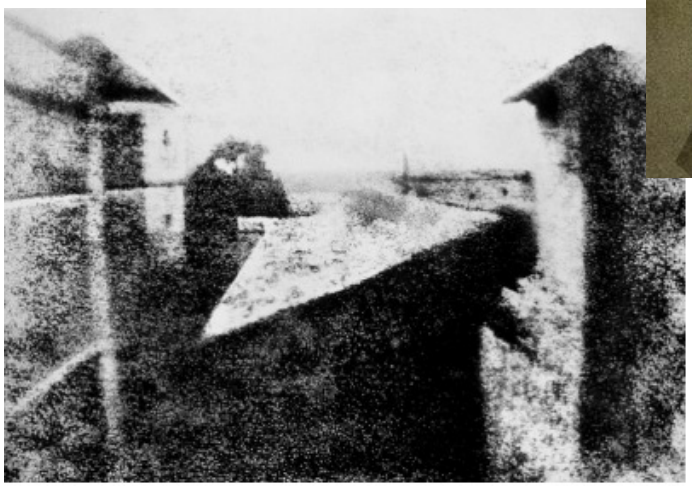


Leonard McMillan

76

MANCHESTER 1824

Image processing

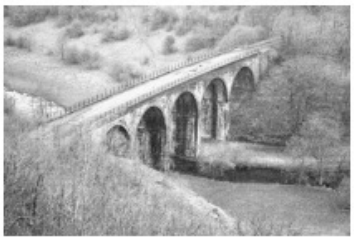


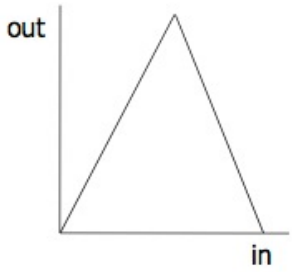
Joseph Nicéphore Niépce, 1826 (bitumen on pewter)

77

MANCHESTER 1824

Point processing





78

MANCHESTER 1824

Region processing

79

MANCHESTER 1824

Image sharpening - before

80

MANCHESTER 1824

Image sharpening - after

81

MANCHESTER 1824

Image smoothing - before

82

MANCHESTER 1824

Image smoothing - after

83

MANCHESTER 1824

Edge detection - before

84

MANCHESTER 1824

Edge detection - after

85

MANCHESTER 1824

Region detection - before

86

MANCHESTER 1824

Region detection - after

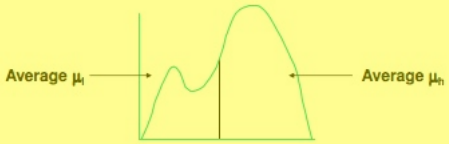
87

MANCHESTER 1824

FAQ: how much Maths is there?

$$I_R = k_{aR} I_{aR} + \frac{I_{pR}}{d'}\left[k_{dR}(\hat{\mathbf{N}}\cdot\hat{\mathbf{L}}) + k_s(\hat{\mathbf{R}}\cdot\hat{\mathbf{V}})^n\right]$$

Find a threshold θ such that



$$\theta = \frac{\mu_l + \mu_h}{2} \Rightarrow T = \theta$$

(Start at θ = 0 and work upwards.)

$$M_{\alpha\beta} = \sum_{image} (x-\bar{x})^{\alpha}(y-\bar{y})^{\beta} f(x,y)$$

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

88

MANCHESTER 1824

What to do next

1. Read the course webpage
2. Collect your OpenGL manual from SSO
3. Go through the Tutorial
4. Start Coursework 1