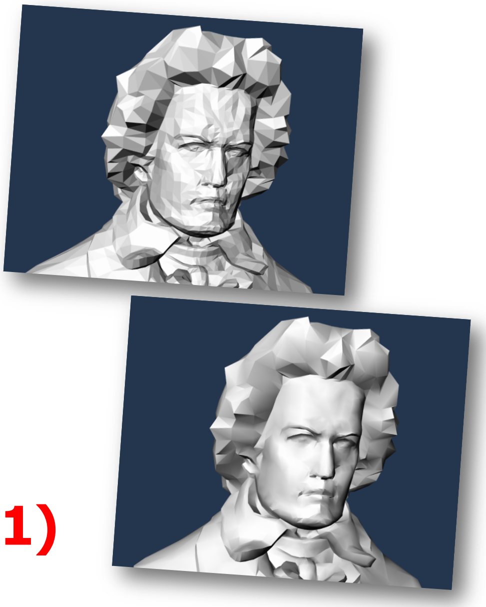
COMP27112

Computer
Graphics
and
Image Processing

7: Rendering (1)



Toby.Howard@manchester.ac.uk

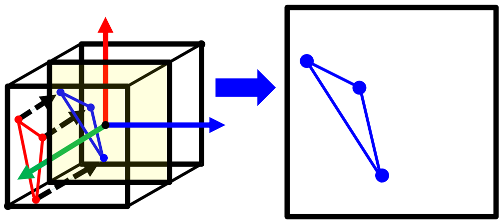
Introduction

- In the next 3 lectures we will look at
  - Recap: how we got to here, we're now "ready to render"
  - Local and global illumination
  - Developing a local illumination model
  - How to apply the local model to meshes
  - How to implement textures and surface detail

2

Recap: ready to render

- Vertices have passed through the viewing pipeline (Lectures 5 & 6) and we have a projection (including z) which we now rasterise (Lecture 4)



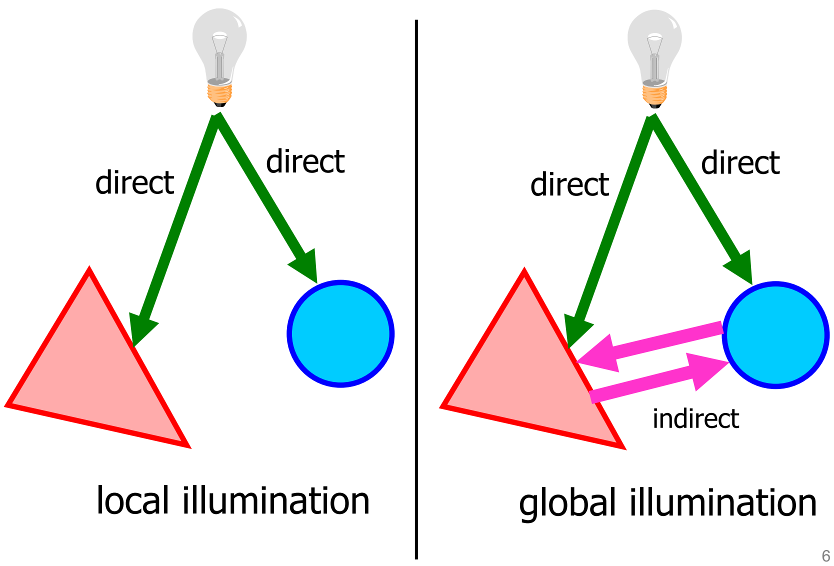
- During rasterisation we:
  - choose the best set of pixels
  - apply the z-buffer to remove hidden parts
  - apply an illumination model & shade smoothly

3

Raw geometry

Rendered geometry

Local and global illumination



Local and global models

- We can model light-matter interaction in two ways
    - **Locally**: we treat each object in a scene separately from any other object
    - **Globally**: we treat all objects together, and model the interactions between objects
- Shortly we'll develop a simple **local** illumination model
- we'll look at **global** models in COMP37111 next year

7

Approximation (yes, that again!)

- The interaction of light and matter is an extremely complex process
- In computer graphics we **model** this process
- We **approximate** it
    - the standard local model is a simple approximation
    - adding per-material algorithms (usually called shaders) gives better results
    - the global model is better than the local model
    - there is huge scope for creativity

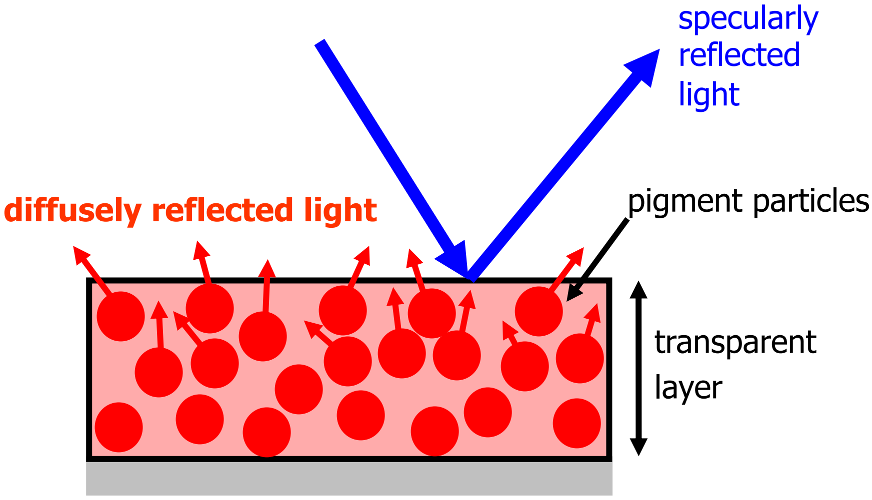
8

Local illumination: elements

- We'll develop a model step-by-step:
  - start with light intensity only (no colour)
  - ambient illumination
  - diffuse reflection
  - a positional light source
  - specular reflection
  - coloured lights and surfaces

9

Light/surface interaction



10

MANCHESTER 1824

Diffuse and specular reflection

- **Diffuse** reflection is absorption and uniform re-radiation
  - some wavelengths are absorbed, some are reflected
  - a green object looks green because it only reflects green
- **Specular** reflection is reflection at the air/surface interface
  - to a first approximation, the colour of the specular reflection is the same as the light source
  - (it isn't really, but we'll look at that again later)

11

MANCHESTER 1824

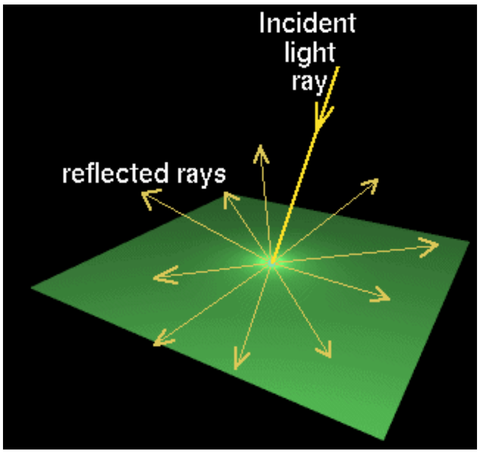
Reflectivity

- There are three broad classes of reflection:
  - diffuse reflection
  - perfect specular reflection
  - imperfect specular reflection

12

MANCHESTER 1824

Diffuse reflection



Ken Brodlie

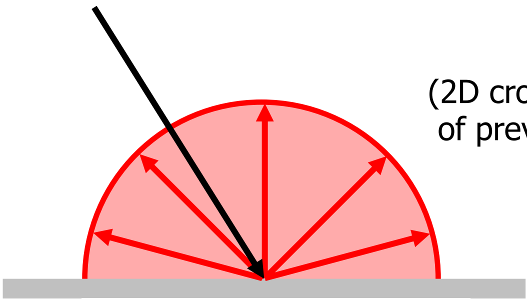
- Incident rays of light are reflected in all directions from the surface

13

MANCHESTER 1824

Perfect diffuse reflection

- A perfect diffuse surface reflects an incoming ray across all angles



(2D cross-section of previous slide)

- The surface looks dull, or matte

14

MANCHESTER
1824

Some diffuse reflectors

Sponge

Brick

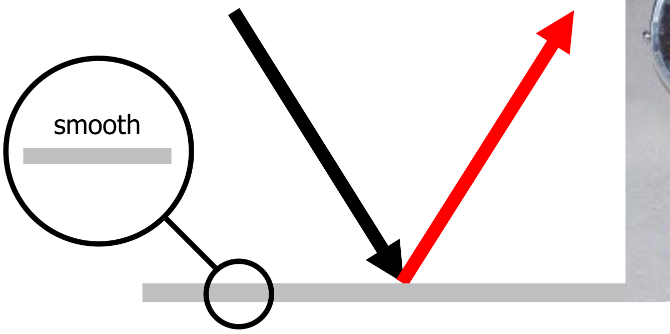
Carpet

Felt

15

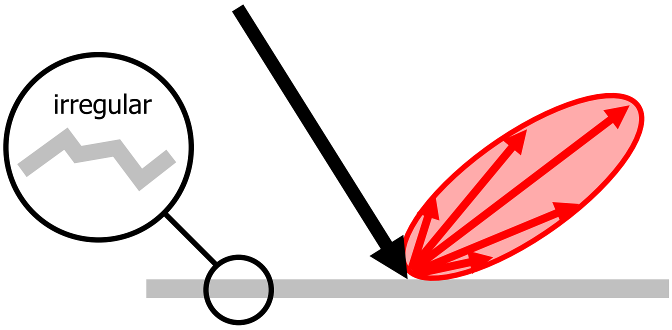
Perfect specular reflection

- A perfect specular surface reflects an incoming ray like a perfect mirror



Imperfect specular reflection

- A imperfect specular surface reflects an incoming ray across a small range of angles



- The surface looks shiny, with highlights

Some imperfect specular reflectors

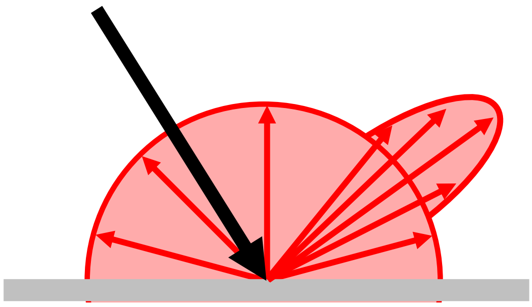
Lacquer-coated aluminium

Glazed ceramic

Stainless steel

Diffuse/specular surfaces

- Most surfaces exhibit a **mixture** of diffuse and specular reflection



- We can model these effects with varying degrees of realism

MANCHESTER 1824

Developing a local model

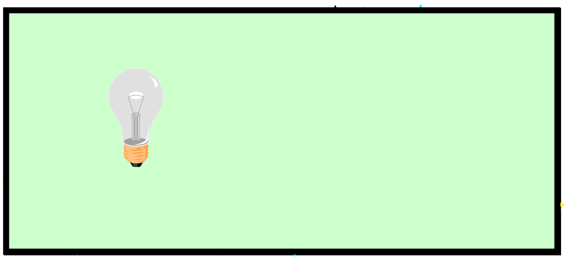
- We'll start with **diffuse** reflection
- and different **types** of illumination:
  1. ambient illumination
  2. point illumination source at infinity (directional illumination)
  3. point illumination source in the scene

21

MANCHESTER 1824

Ambient illumination

- Consider an environment containing a light source
- Multiple reflections will give a general level of illumination in the scene



22

Ambient illumination

- If monochrome intensity of ambient light is I_a
- Amount of ambient light diffusely reflected from a surface is

$$I_{\text{ambient}} = k_a I_a$$

- Where k_a is the **ambient reflection coefficient** of the surface and $0 \le k_a \le 1$

23

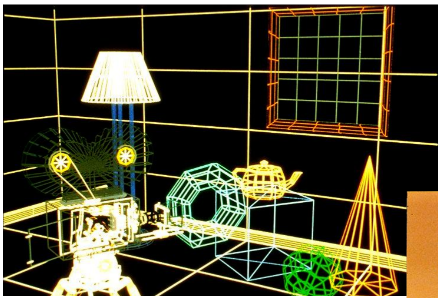
Local illumination model v1

- We now have the first term in the model we're developing
- I = ambient
- $\boxed{I = k_a I_a}$

24

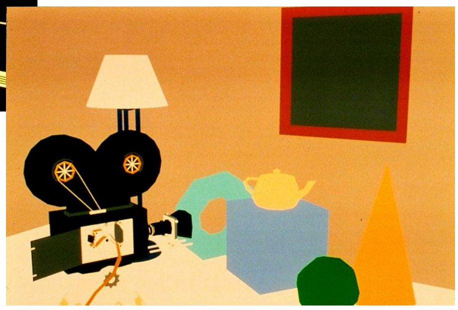
MANCHESTER 1824

Wireframe versus ambient



Wireframe rendering

Rendering with ambient illumination

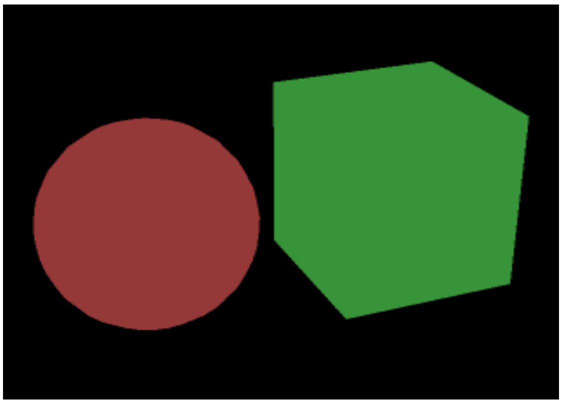


25

MANCHESTER 1824

Effect of ambient illumination

- Each object is uniformly illuminated
- We've lost all the 3D information



26

Effect of varying k_a

We keep I_a is constant and vary k_a

$k_a = 0.1$ → $k_a = 1.0$

27

True ambient lighting

In fact, $I = k_a I_a$ is a **HUGE** simplification of ambient illumination, which is not constant in a scene

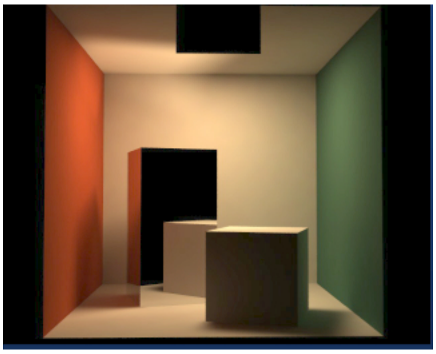
The scene below is rendered with a global model called **radiosity**, a far better approximation of ambient illumination

28

MANCHESTER 1824

True ambient lighting

Real scene (photograph)



Scene modelled and rendered with accurate global illumination model

Cornell University

They look the same, but the next slide shows how they differ.

29

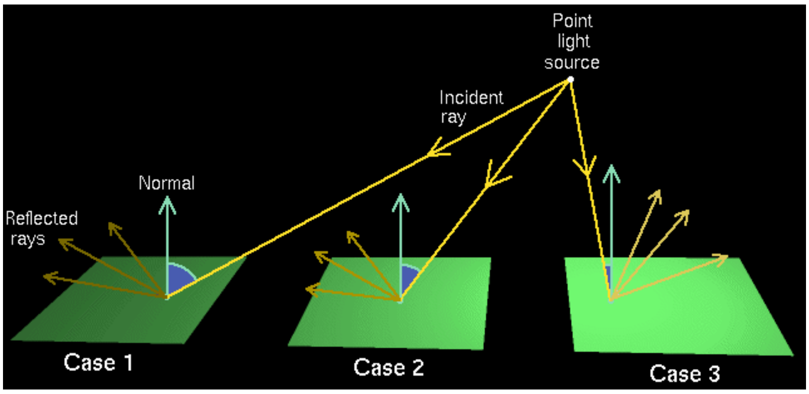
MANCHESTER 1824

True ambient lighting

Subtraction, pixel by pixel, of photograph and rendered image, showing differences

30

Directional lighting

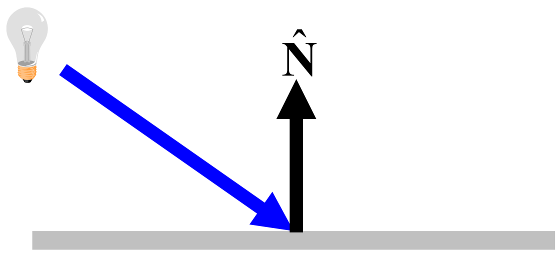


- We need to model the effects of
  - different angles of incidence
  - different distances from the light source

31

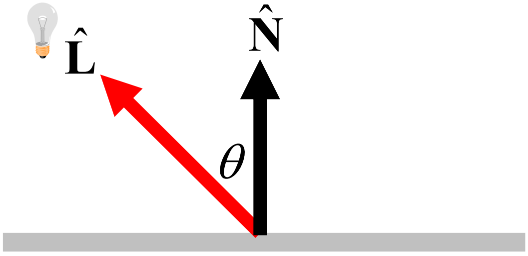
Directional lighting

- Consider light from a point source at infinity – so we're only concerned with the direction



32

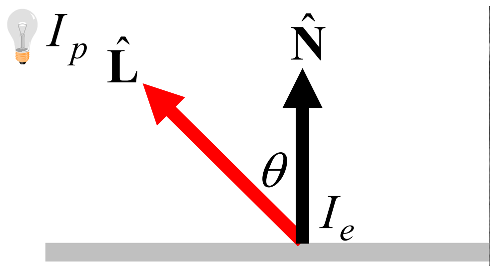
Describing surface orientation



- $\hat{\mathbf{N}}$ is surface normal
- $\hat{\mathbf{L}}$ is direction of light source
- θ is angle of incidence
- All vectors are normalised

33

Diffuse reflection: Lambert's Law



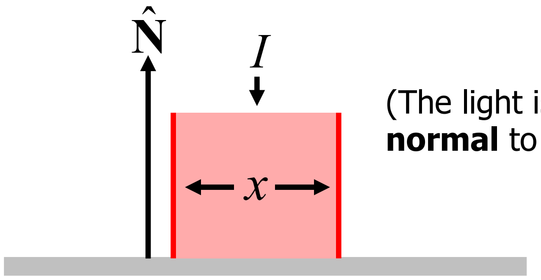
Johann Lambert (1728-1777)

- Light source of intensity I_p
- Effective intensity I_e received is
- $I_e = I_p \cos\theta$

34

Lambert's Law derived (1)

- Consider light of intensity I and cross-sectional width x falling on a surface:



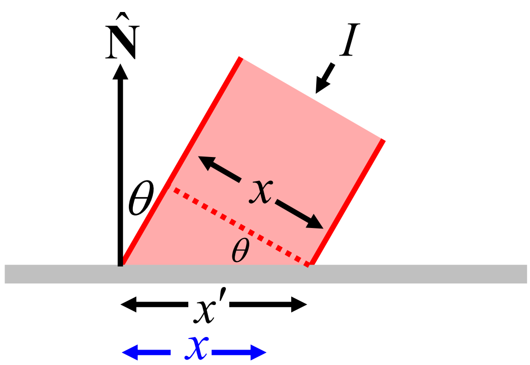
(The light is falling **normal** to the surface)

- So width x on surface receives all of intensity I

35

Lambert's Law derived (2)

- Now consider the light inclined at θ
- $x' = x / \cos\theta$, so $x = x' \cos\theta$
- So width x receives intensity $I \cos\theta$



36

MANCHESTER 1824

Diffuse reflectivity

- We describe the diffuse reflectivity of a surface by assigning it a value k_d
- k_d is the **diffuse reflection coefficient** of the surface, $0 \le k_d \le 1$
- Amount of diffusely reflected light is
  - $I_{\text{diffuse}} = I_p k_d \cos\theta$

    $I_{\text{diffuse}} = I_p k_d (\hat{\mathbf{N}} \cdot \hat{\mathbf{L}})$

37

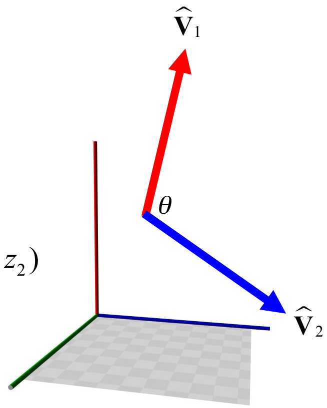
MANCHESTER 1824

The Dot Product

- is the scalar product of the individual components

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} \bullet \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = (x_1 \times x_2) + (y_1 \times y_2) + (z_1 \times z_2)$$

- For normalized vectors, their dot product is the cosine of the angle between them
- Essential for rendering



$$\cos\theta = \hat{\mathbf{V}}_1 \bullet \hat{\mathbf{V}}_2$$

38

MANCHESTER 1824

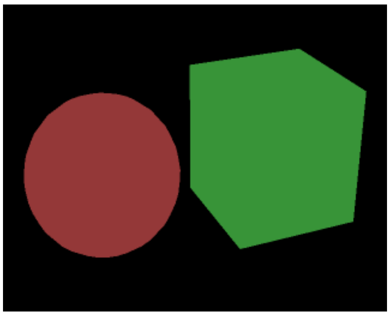
Local illumination model v2

- I = ambient + diffuse
- $$\boxed{I = k_a I_a + I_p k_d (\hat{\mathbf{N}} \cdot \hat{\mathbf{L}})}$$

39

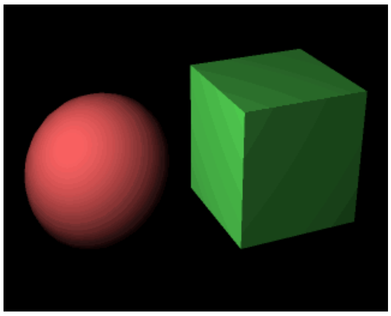
MANCHESTER 1824

Ambient versus point source



Rendering with
ambient illumination

Rendering with
a point light source
and diffuse reflection



Ken Brodlie

40

Ambient versus point source

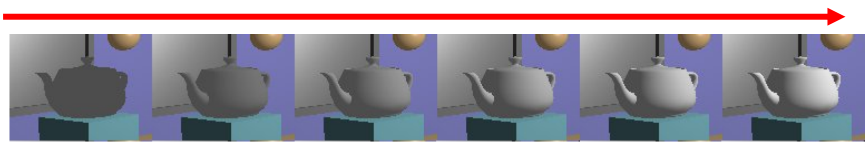
Rendering with ambient illumination

Rendering with a point light source and diffuse reflection

41

Effect of k_d

- Varying k_d from 0 to 1



42

Light source distance

- Now we'll consider the light source distance
- Physically, light intensity falls off with the square of distance travelled
- After travelling d, original intensity I_p is now I_e
- $I_e = \dfrac{I_p}{4\pi d^2}$

43

Light source distance

- While this is physically correct, it doesn't usually work well for computer graphics
- Why? Approximation!
- We only have a limited number of pixel intensities, and often the d^2 term changes too rapidly, so instead we use:

$$I_e = \frac{I_p}{k_c + k_l d + k_q d^2}$$

OpenGL provides functions to set these terms

We tune k_c, k_l, k_q for the best results

44

Local illumination model v3

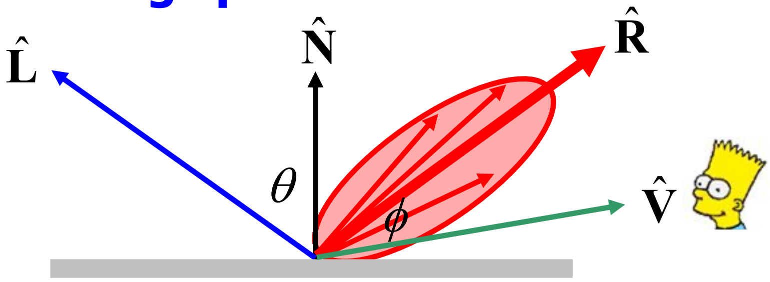
- I = ambient + distance (diffuse)
- $$\boxed{I = k_a I_a + \frac{I_p}{d'} k_d (\hat{\mathbf{N}} \cdot \hat{\mathbf{L}})}$$
- Where $d' = k_c + k_l d + k_q d^2$

45

Modelling specular reflection

46

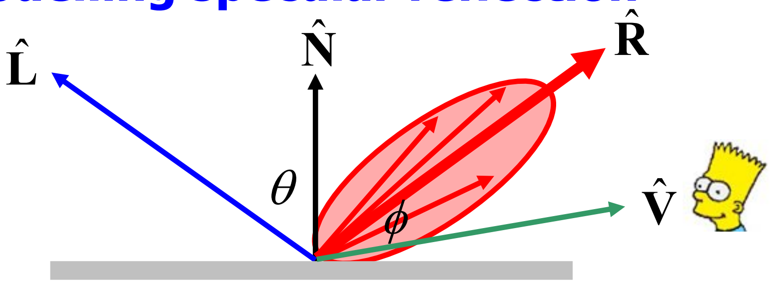
Modelling specular reflection



- $\hat{\mathbf{R}}$ is a vector giving the direction of maximum specular reflection.
- $\hat{\mathbf{R}}$ makes an angle of θ with $\hat{\mathbf{N}}$, as does $\hat{\mathbf{L}}$
  - (for a perfect mirror, angle of incidence == angle of reflection)
- $\hat{\mathbf{V}}$ is a vector pointing to the observer's position

47

Modelling specular reflection

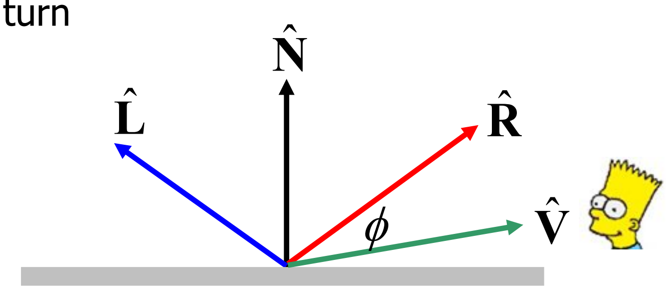


- Experiment shows specular reflection varies with:
  - Angle ϕ between $\hat{\mathbf{R}}$ and $\hat{\mathbf{V}}$
  - Incident angle θ and light wavelength λ
  - So we seek a function $I_{\text{specular}} = S(\phi, \theta, \lambda)$

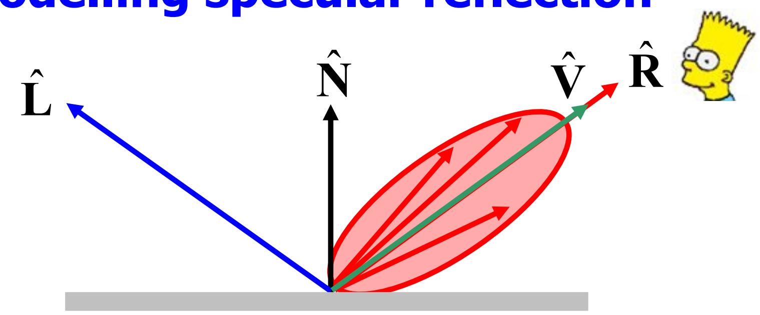
48

Modelling specular reflection

- What is this function $S(\phi, \theta, \lambda)$?
- We'll look at the effects of ϕ, θ, λ in turn
- First, ϕ



Modelling specular reflection

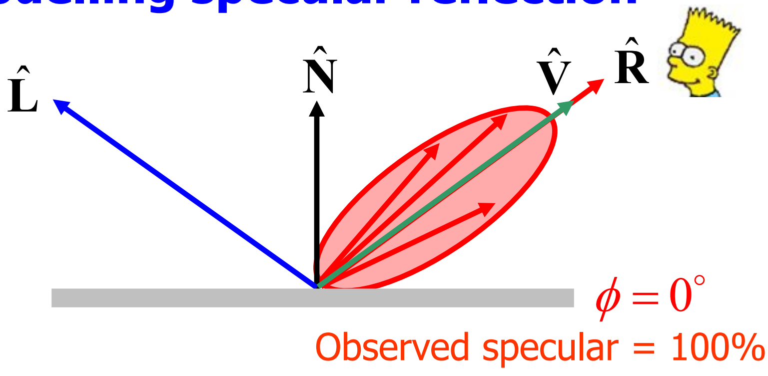


Observed specular = 100%

- Specular reflection spread out, around $\hat{\mathbf{R}}$
- Viewer sees maximum specular reflection when $\hat{\mathbf{V}}$ coincides with $\hat{\mathbf{R}}$

MANCHESTER 1824

Modelling specular reflection

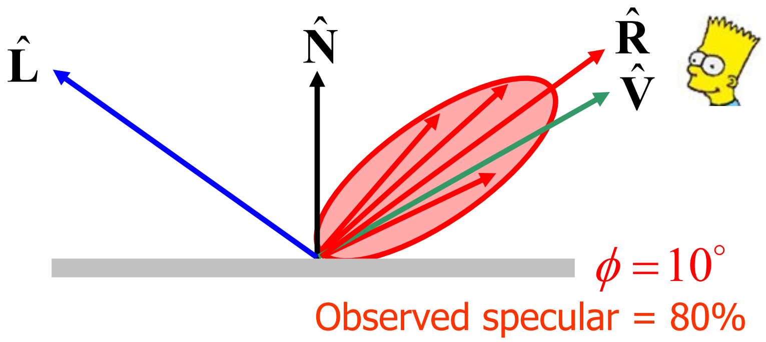


- As $\hat{\mathbf{V}}$ diverges from $\hat{\mathbf{R}}$ by angle ϕ, viewer sees less specular reflection

51

MANCHESTER 1824

Modelling specular reflection

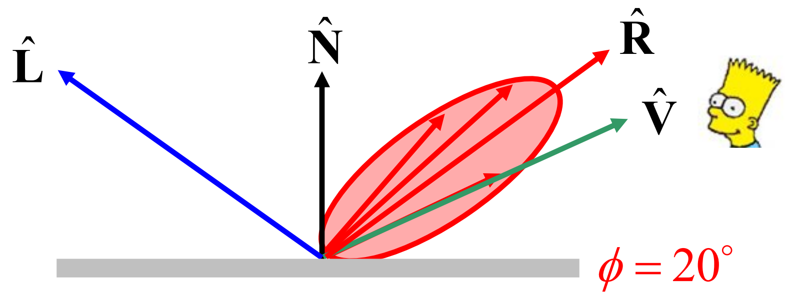


- As $\hat{\mathbf{V}}$ diverges from $\hat{\mathbf{R}}$ by angle ϕ, viewer sees less specular reflection

52

MANCHESTER 1824

Modelling specular reflection



$\phi = 20^\circ$

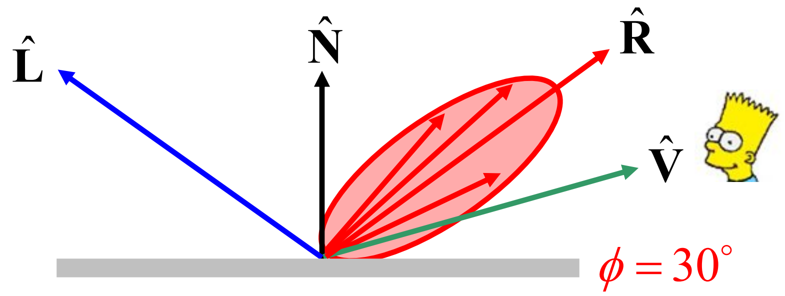
Observed specular = 60%

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

53

MANCHESTER 1824

Modelling specular reflection



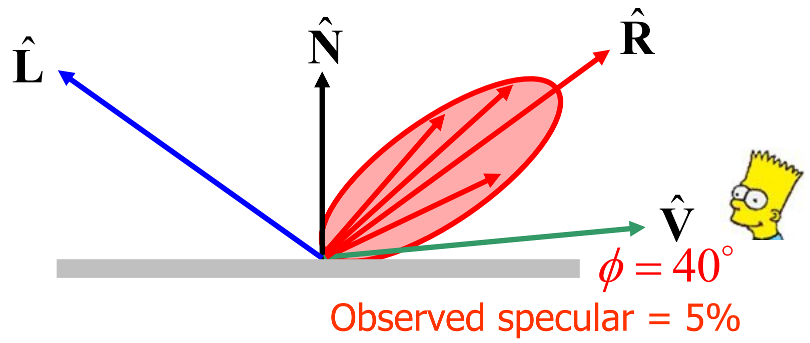
$\phi = 30^\circ$

Observed specular = 30%

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

54

Modelling specular reflection



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

55

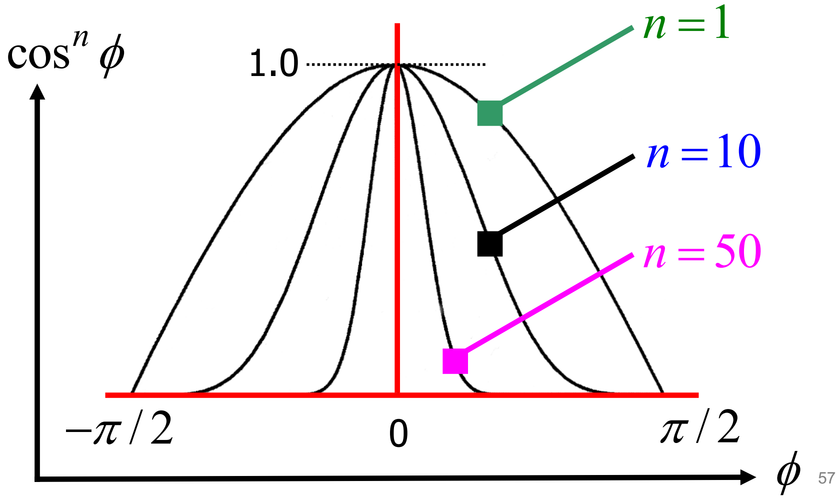
Modelling specular reflection

- Variation of observed specular = $F(\phi)$
- But what is the function F ?
- Bui-Tuong Phong (1942-1975) proposed using the function $\cos^n \phi$ to approximate F
  - it gives us a convenient way to think about things
  - but it has no basis in reality!
  - In fact, F is rather complex, as we will see shortly

56

Phong's specular function

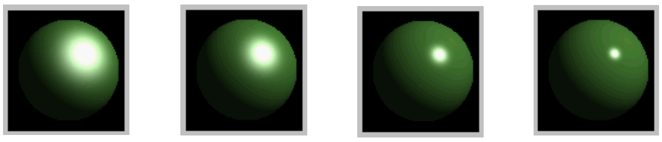
- Graph of $\cos^n \phi$ against ϕ



57

Phong's specular function

- So we now have
  - $I_{\text{specular}} = I_p \cos^n \phi$
- Which we can rewrite using vectors as
  - $I_{\text{specular}} = I_p (\hat{\mathbf{R}} \cdot \hat{\mathbf{V}})^n$
- Normally we use $1 \leq n \leq 200$



$n = 10$ $n = 20$ $n = 80$ $n = 200$

Richard Lobb

58

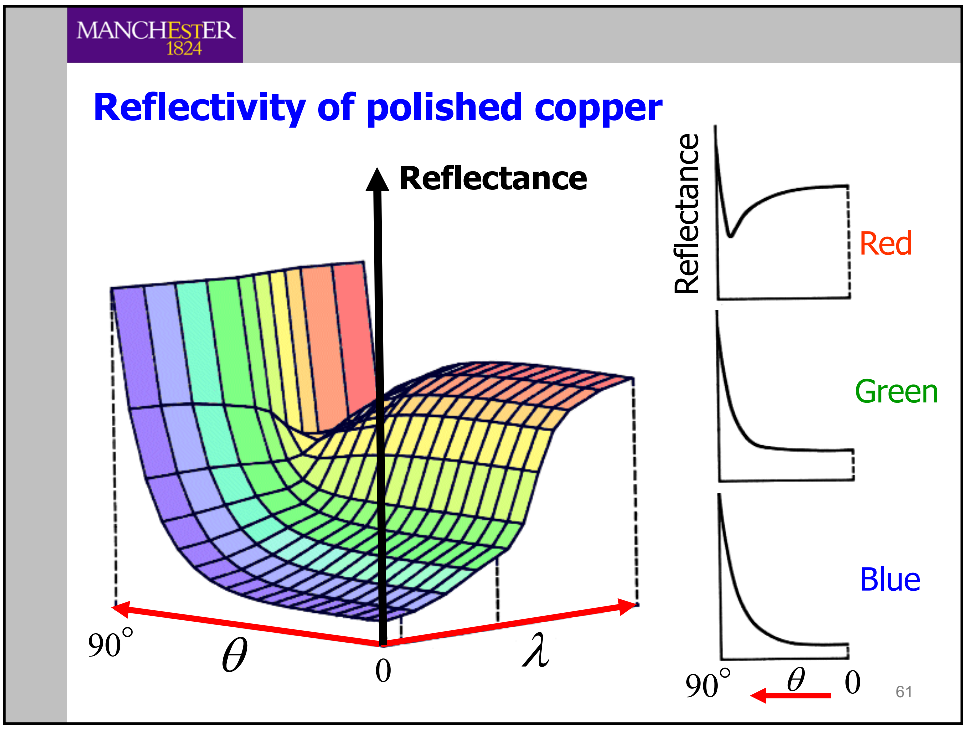
Phong's legacy

- *Computer Generated Pictures*, Bui Tuong Phong University of Utah, 1975
- http://www.cs.northwestern.edu/~ago820/cs395/Papers/Phong_1975.pdf
- "We do not expect to be able to display the object exactly as it would appear in reality, with texture, overcast shadows, etc. We hope only to display an image that approximates the real object closely enough to provide a certain degree of realism."

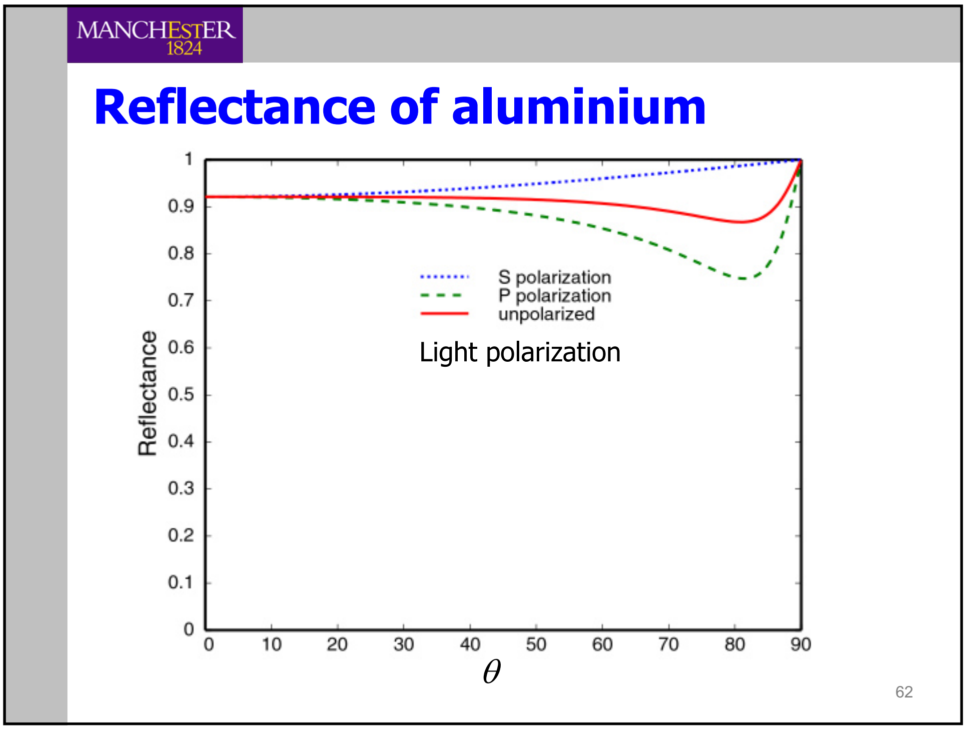
Incident angle and wavelength

- Recall $I_{\text{specular}} = S(\phi, \theta, \lambda)$
- We've accounted for ϕ
- Now we look at the effects of θ and λ

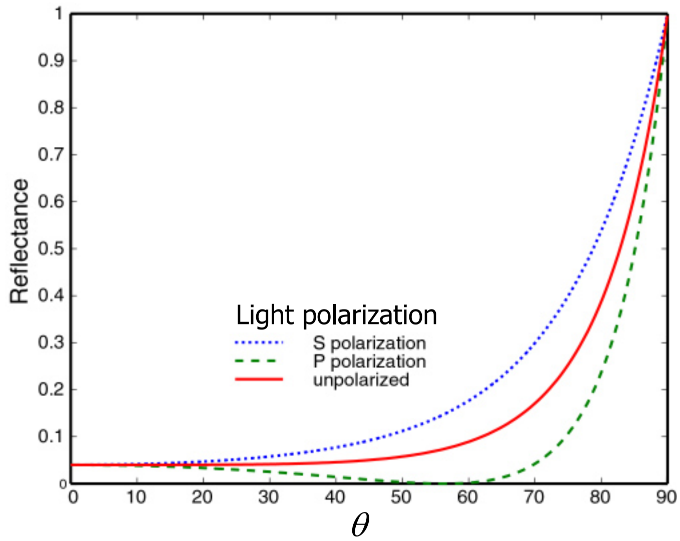
Reflectivity of polished copper



Reflectance of aluminium



Reflectance of a non-conductor



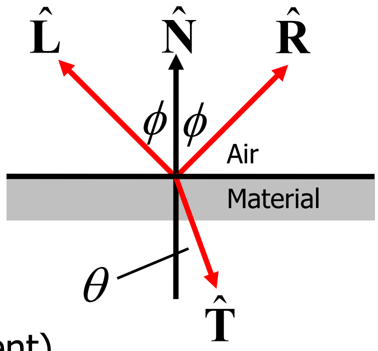
Incident angle and wavelength

- This complex variation is expressed by the Fresnel equation (Augustin-Jean Fresnel, 1788-1827)
  - A founder of the wave theory of light
  - Developed theory of diffraction of light
  - Obtained circularly polarised light
  - Developed the use of compound lenses instead of mirrors for lighthouses

Incident angle and wavelength

- The Fresnel equation

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



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

65

Incident angle and wavelength

- In practice, we often replace F with a single constant k_s!
- k_s is the **specular reflection coefficient** of the surface, $0 \le k_s \le 1$
- $I_{\text{specular}} = I_p k_s (\mathbf{R} \cdot \mathbf{V})^n$
- We sacrifice accuracy for efficiency, and the effect has a plastic look

66

Local illumination model v4

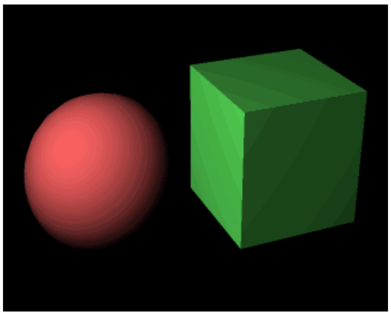
- I = ambient + distance (diffuse + specular)

$$I = k_a I_a + \frac{I_p}{d'}\left[k_d(\hat{\mathbf{N}} \cdot \hat{\mathbf{L}}) + k_s(\hat{\mathbf{R}} \cdot \hat{\mathbf{V}})^n\right]$$

- Where $d' = k_c + k_l d + k_q d^2$

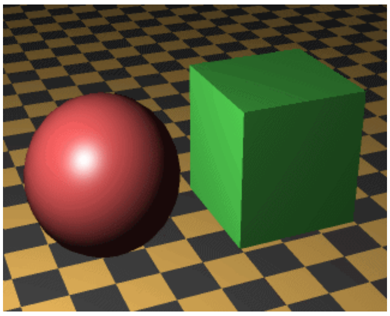
67

Diffuse and specular rendering



Rendering with a point light source and diffuse reflection

Rendering with a point light source and diffuse and specular reflection

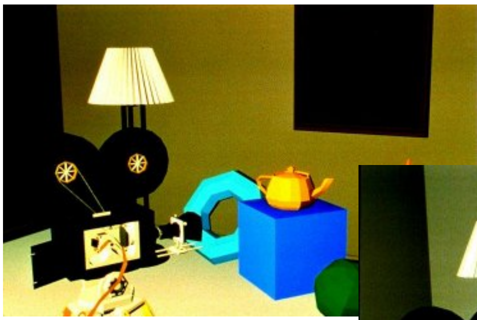


Ken Brodlie

68

MANCHESTER 1824

Diffuse and specular rendering



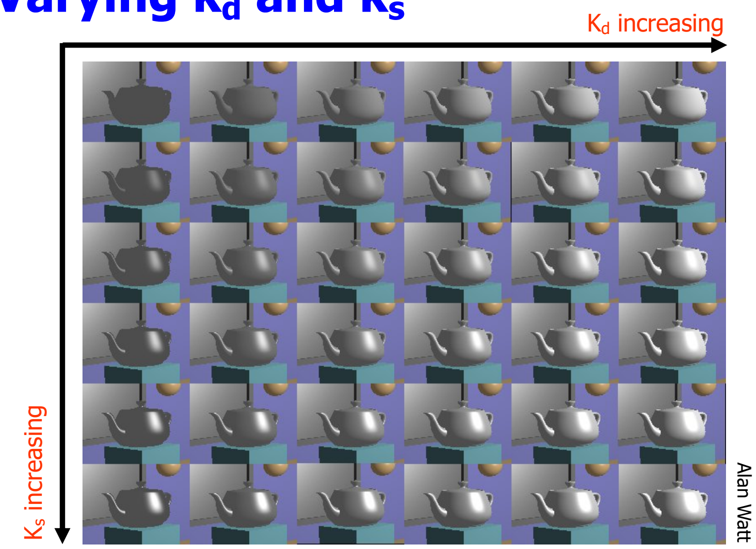
Rendering with a point light source and diffuse reflection

Rendering with a point light source and diffuse and specular reflection

69

MANCHESTER 1824

Varying k_d and k_s



70

Varying k_s and exponent n

Incorporating colour

- So far, we' ve only considered light **intensity**, not colour
- It' s easy to incorporate – we express light colour as a triple of RGB intensities:
  - I_{pR}, I_{pG}, I_{pB}
- And correspondingly we express surface colour using
  - k_{aR}, k_{aG}, k_{aB}
  - k_{dR}, k_{dG}, k_{dB}

MANCHESTER 1824

Local illumination model v5

- We now have a separate expression for each colour component. For example, for red:

- I_R = ambient$_{Red}$ +

  distance (diffuse$_{Red}$ + specular)

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

73

MANCHESTER 1824

Multiple lights

- Finally, what if we have more than one light?

- Easy, compute illumination separately for each and sum. For M lights:

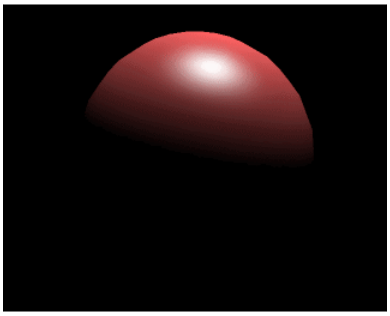
- $$I = ambient + \sum_{i=1}^{M} (\text{diffuse}_i + \text{specular}_i)$$

74

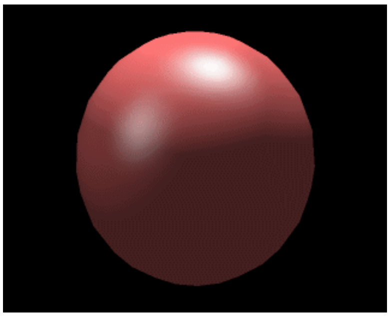
MANCHESTER 1824

Multiple lights

One light



Two lights

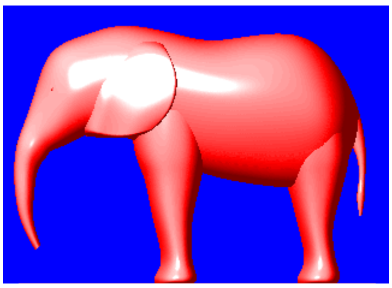


Ken Brodlie

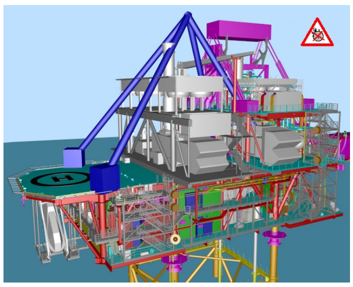
75

MANCHESTER 1824

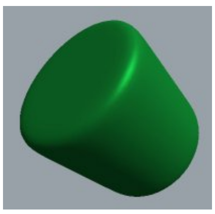
Examples of the model in use



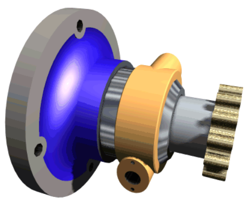
Lex Lennings



AIG



James Sinnott



Alan Watt

nVIDIA

76

MANCHESTER
1824

The standard model

- The local illumination model we've developed is the standard model in use today
- Implemented in OpenGL
- Implemented in hardware on GPUs

77