3D Graphics Programming Tools

Lighting

Queen Mary
University of London

Lighting versus “Colouring”

Today’s agenda

- Why do we need to calculate lighting?
- Definitions
- Ambient and directional light sources
- Diffuse and specular reflection
- Shading models

Light stage

Queen Mary
University of London

Solving the lighting problem

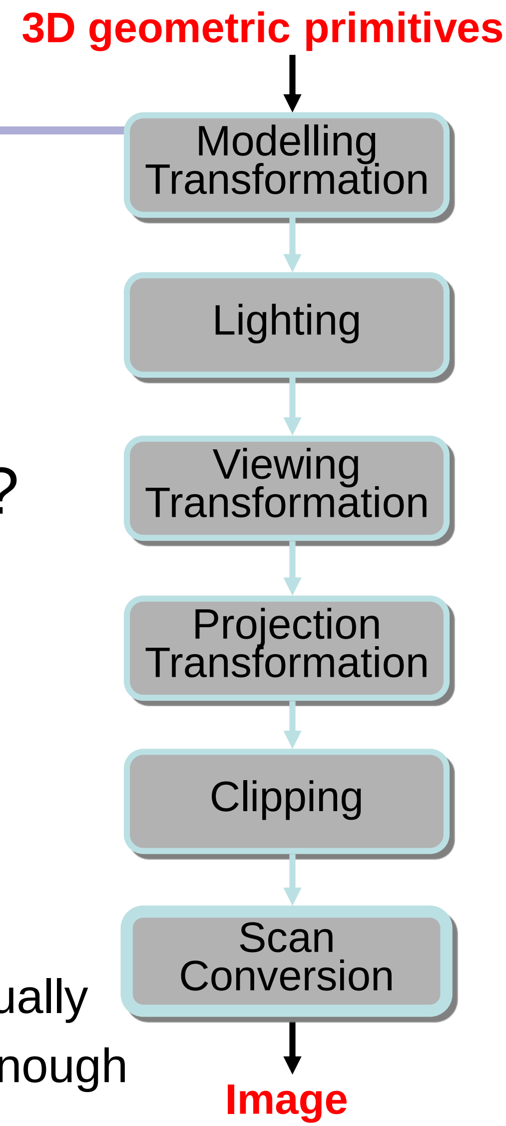
- Where are we?
  - We somewhat understand the perception of light (colour)
  - We know how to represent and generate colour using computers
- We now need to understand the interaction of light and objects

Lighting

- Later, we will learn how to rasterise
  – i.e., given a 3-D triangle and a 3-D viewpoint, we know which pixels represent the triangle
- But … what colour should those pixels be?

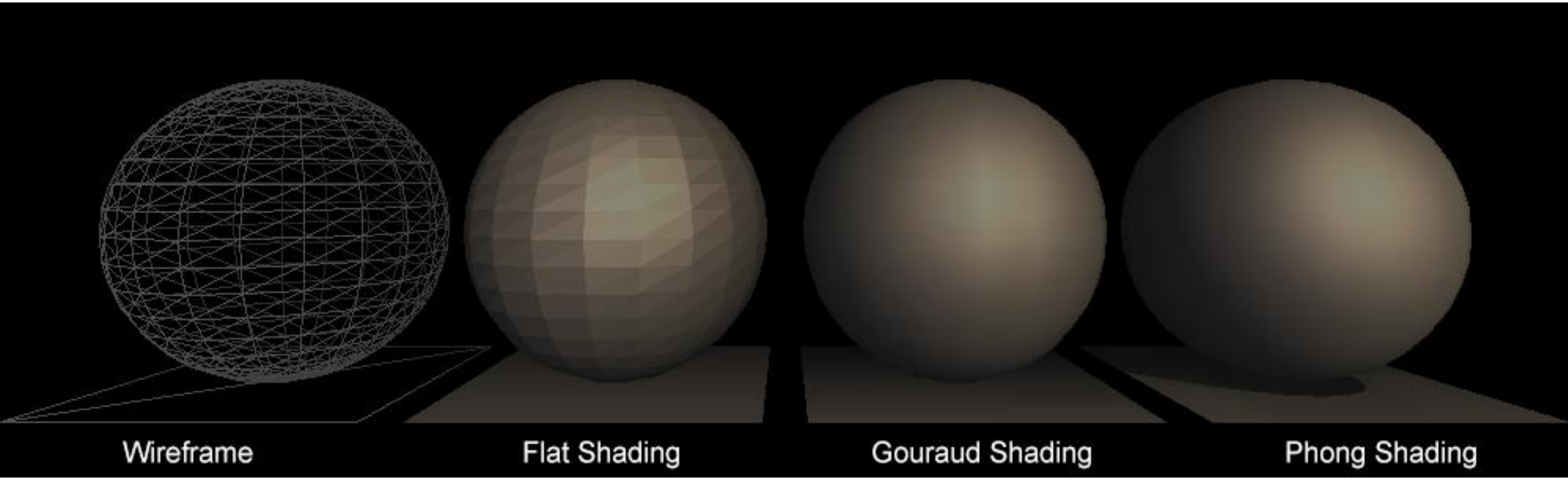
- To create a realistic image

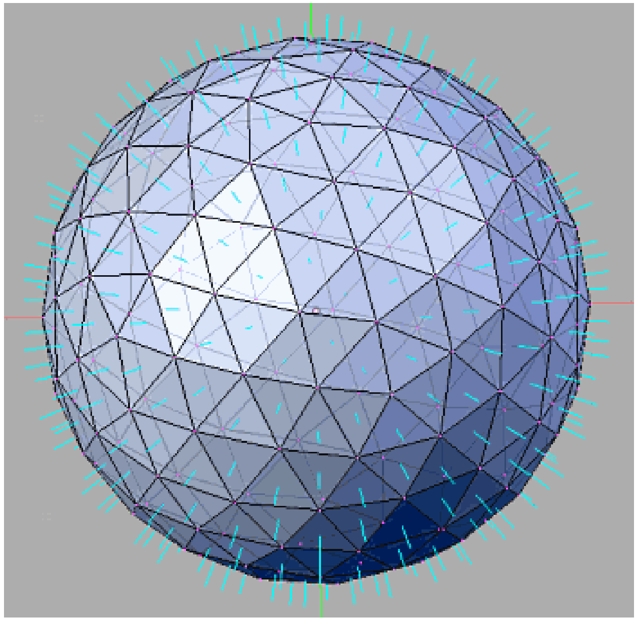
  → need to simulate the lighting
  of the **surfaces** in the scene
  - Fundamentally → simulation of *physics* and *optics*
  - In reality → we use a lot of approximations (perceptually based hacks to do this simulation fast enough



Queen Mary University of London

Modelling and Lighting





Queen Mary
University of London

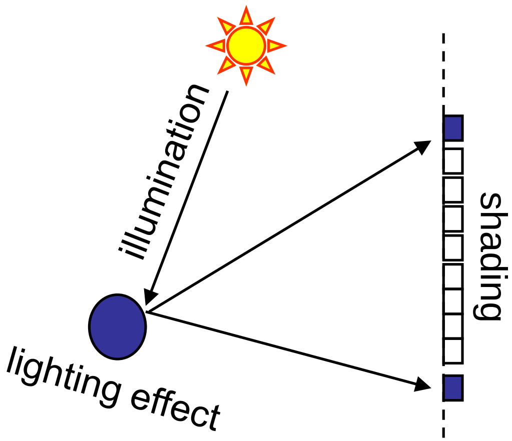
Today's agenda

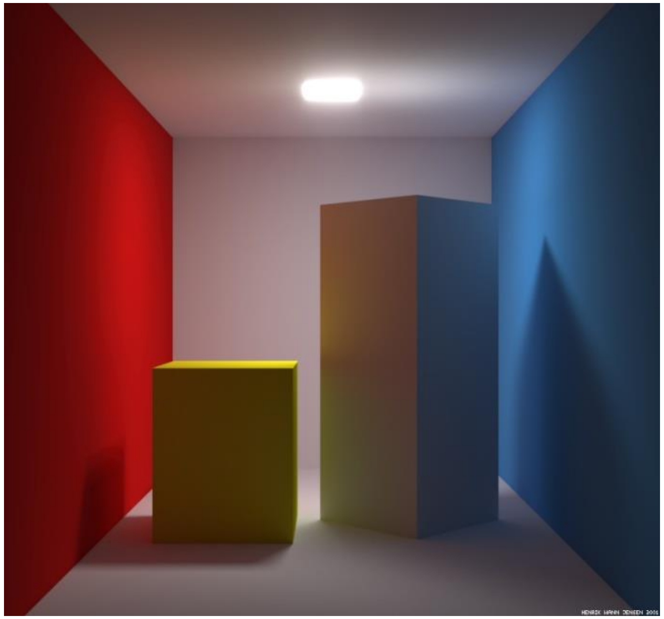
- Why do we need to calculate lighting?
- Definitions
- Ambient and directional light sources
- Diffuse and specular reflection
- Shading models

Queen Mary
University of London

Definitions

- Illumination
  - the transport of energy from light sources to surfaces & points
    - Note: includes *direct* and *indirect illumination*
- Lighting
  - the process of computing the luminous intensity (i.e. outgoing light) at a particular 3-D point
- Shading
  - the process of assigning colours to pixels





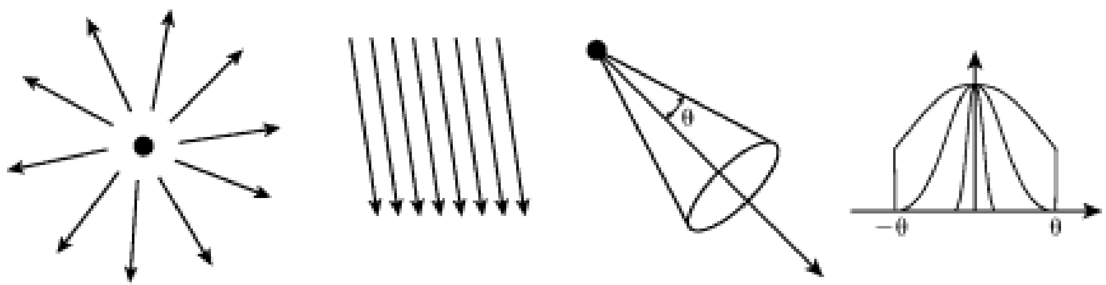
Definitions

- Illumination models → two categories:
  - Physically based
    - models based on the actual physics of light interacting with matter
  - Empirical
    - simple formulations that approximate observed phenomenon

- Interactive graphics
  - for simplicity → mostly use empirical models
  - increasingly → realistic graphics are using physically based models

Two components of lighting

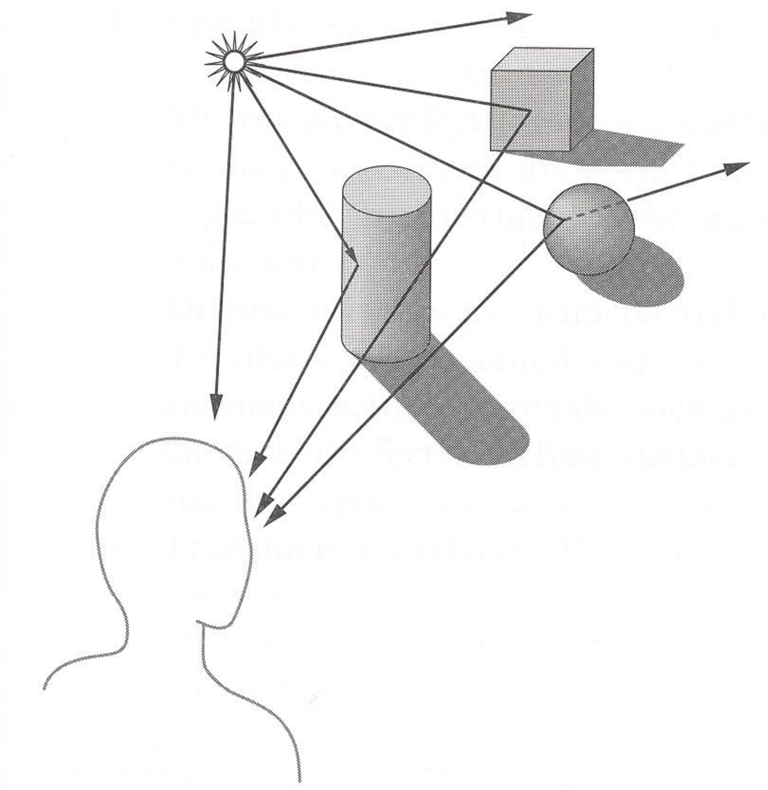
- Light sources (or emitters)
  - spectrum of emittance (*colour of the light*)
  - geometric attributes (*position*, *direction*, *shape*)
  - directional attenuation



- Surface properties
  - reflectance spectrum (*colour of the surface*)
  - subsurface reflectance
  - geometric attributes (*position*, *orientation*, *micro-structure*)

Light Models

- Light models: include 3 aspects of information:
  - Emission at light sources
  - Scattering at surfaces
  - Reception at the camera



- Desirable features …
  - Concise
  - Efficient to compute
  - “Accurate” (look real)

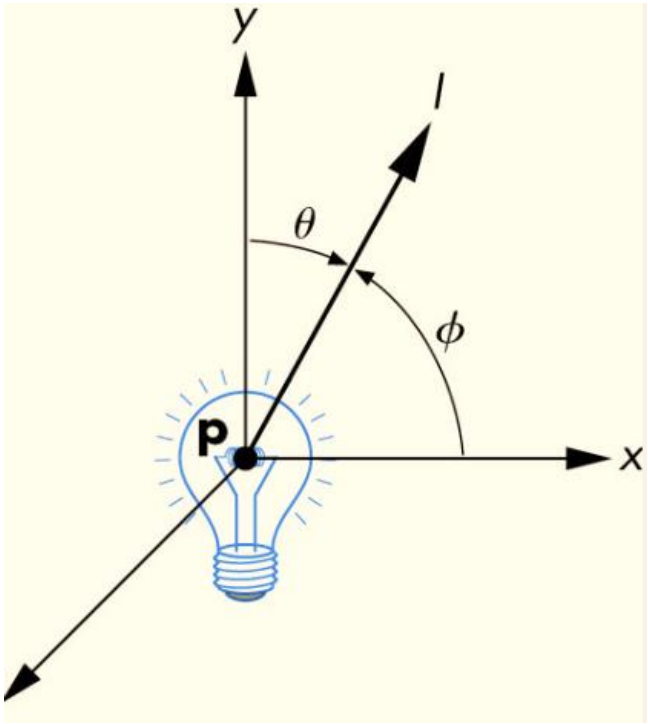
Today's agenda

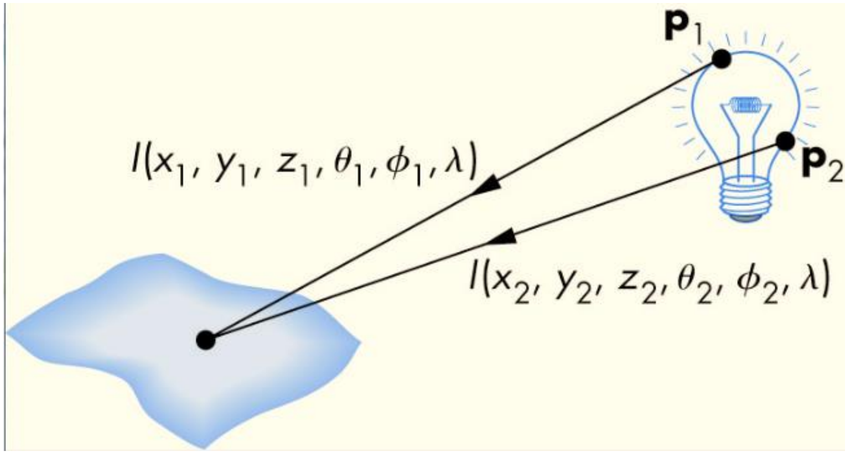
- Why do we need to calculate lighting?
- Definitions
- Ambient and directional light sources
- Diffuse and specular reflection
- Shading models

Queen Mary
University of London

Modelling light sources

- I ($x,y,z,\theta,\phi,\lambda$) ...
  - describes the intensity of energy (I)
  - leaving a light source from location(x,y,z)
  - with direction (θ,ϕ)
  - at wavelength λ

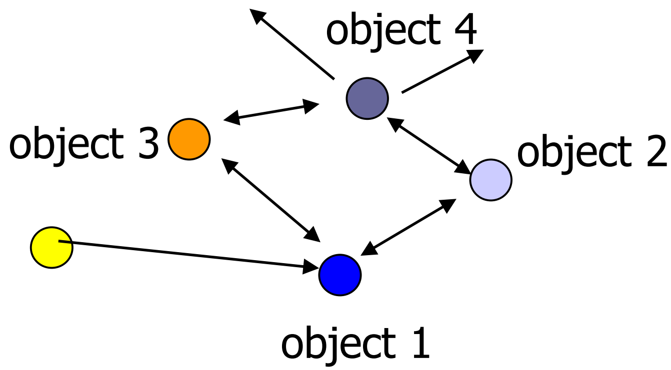




Queen Mary
University of London

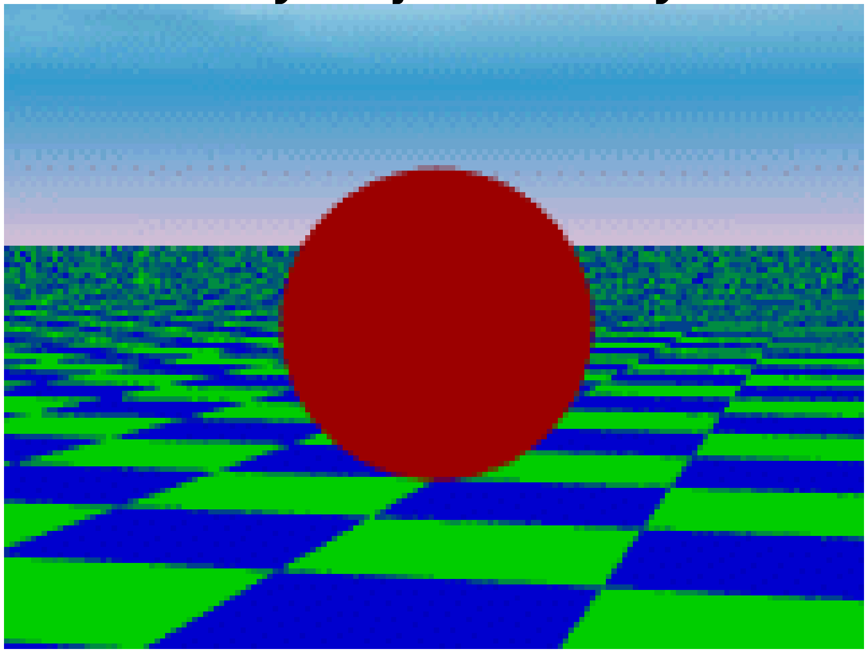
Ambient light sources

- Simulate indirect illumination from emitters, bouncing off intermediate surfaces (interaction from objects, environmental)
  - Objects not directly lit are typically still visible
    - e.g., the ceiling of a room, undersides of desks
  - Too expensive to calculate (in real time), so we use a hack called an ambient light source
    - No spatial or directional characteristics
    - Illuminates all surfaces equally
    - Amount reflected depends on surface properties



Ambient light model

- A scene lit only with an ambient light source
- The same for every object, everywhere and in any direction



Light position not important

Viewer position not important

Surface angle not important

$$I \cong k_a \cdot I_a$$

$$I_a = L(P_0) = L \quad \text{(constant)}$$

I_a: incident ambient light intensity

k_a: ambient reflection coefficient

I: resulting intensity

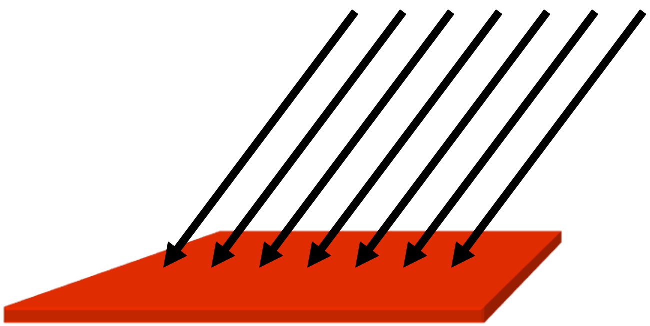
Queen Mary University of London

Distant light sources

- For a distant light source we make simplifying assumptions
  - direction is constant for all surfaces in the scene
  - all rays of light from the source are parallel
    - As if the source were infinitely far away from the surfaces in the scene
    - A good approximation to sunlight

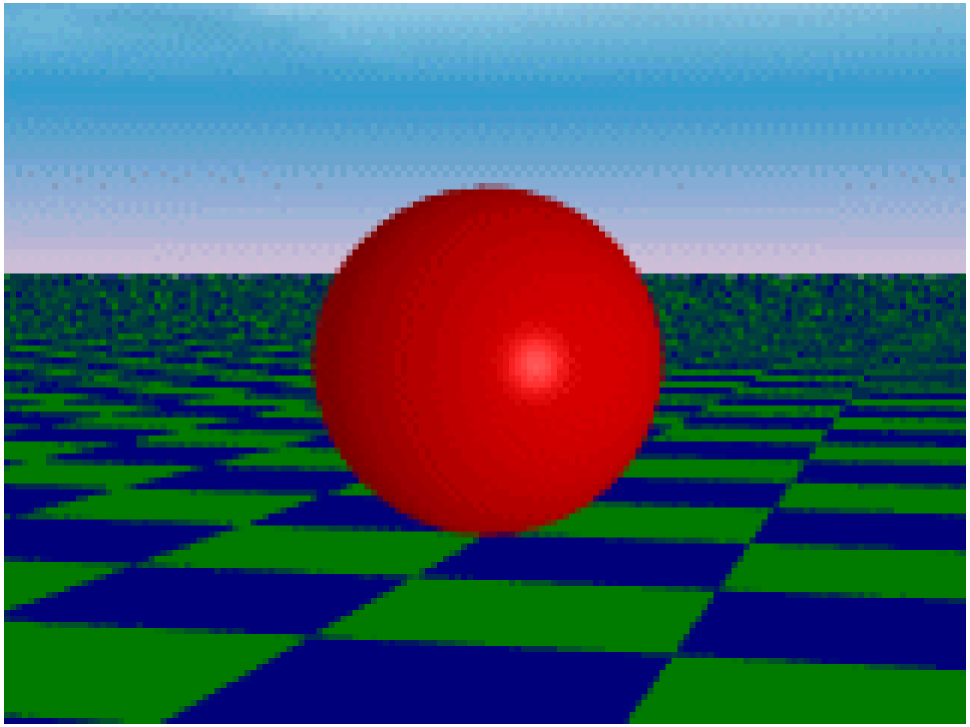
- The direction from the light source to a surface is important in lighting the surface

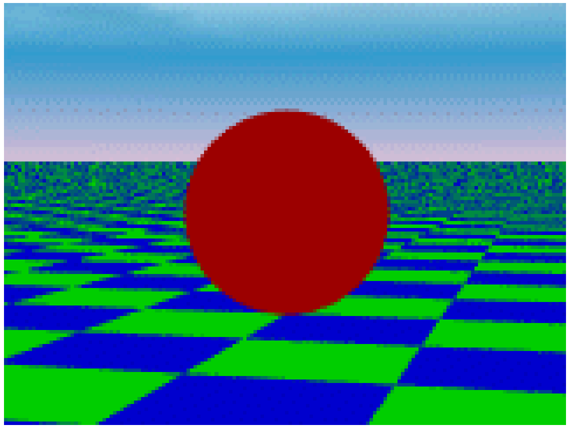
$$\vec{L}(P, P_0) = L \cdot \vec{l}$$



Distant light sources

- The same scene lit with a distant and an ambient light source





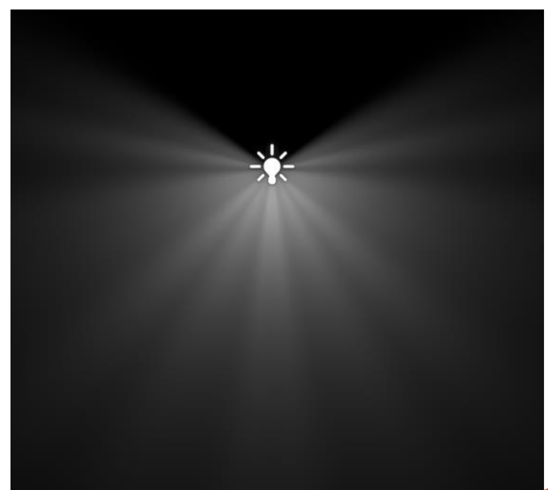
Point light sources

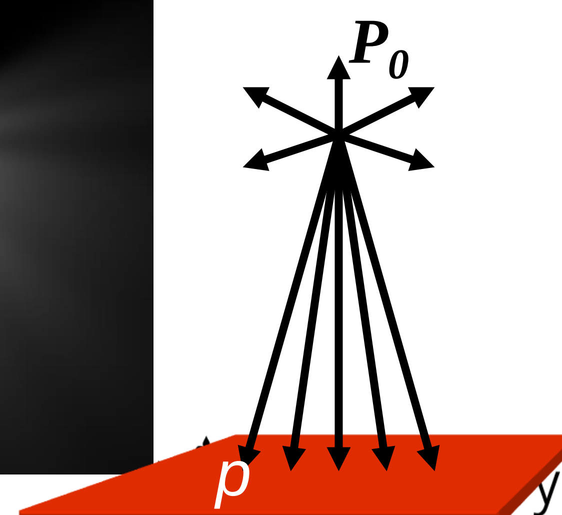
- Point light source
  - emits light equally in all directions from a single point
  - the direction to the light from a point on a surface differs for different points → need to calculate a normalised vector to the light source for every point we light

$$\vec{l} = \frac{P - P_0}{|P - P_0|}$$

  - The intensity of illumination received from a source located at P0 at a point P is proportional to the inverse square of the distance from the source

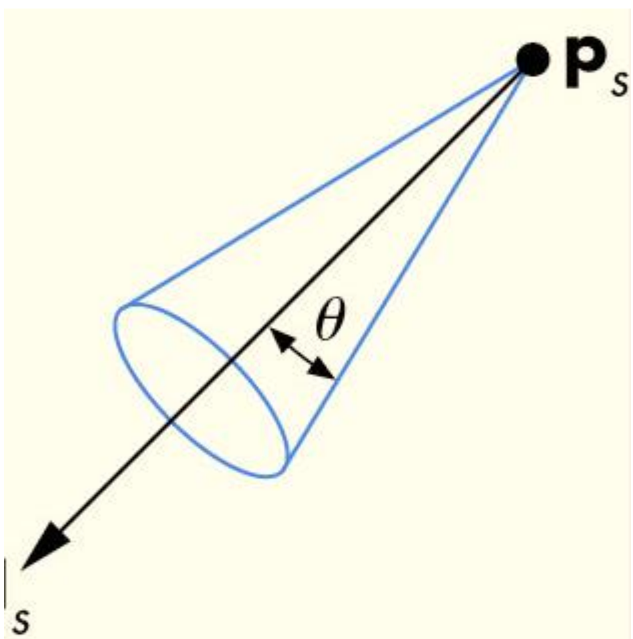
$$\vec{L}(P, P_0) = \frac{L(P_0) \cdot \vec{l}}{|P - P_0|^2}$$

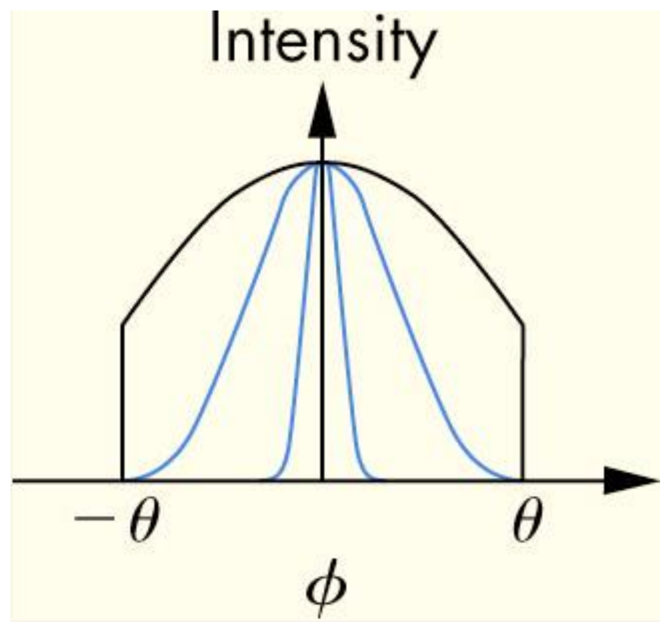




Spotlights

- Spotlights
  - Characterised by a narrow range of angles through which light is emitted.
  - Realistic spotlights are characterised by the distribution of light within the cone (usually with most of the light concentrated in the center of the cone).

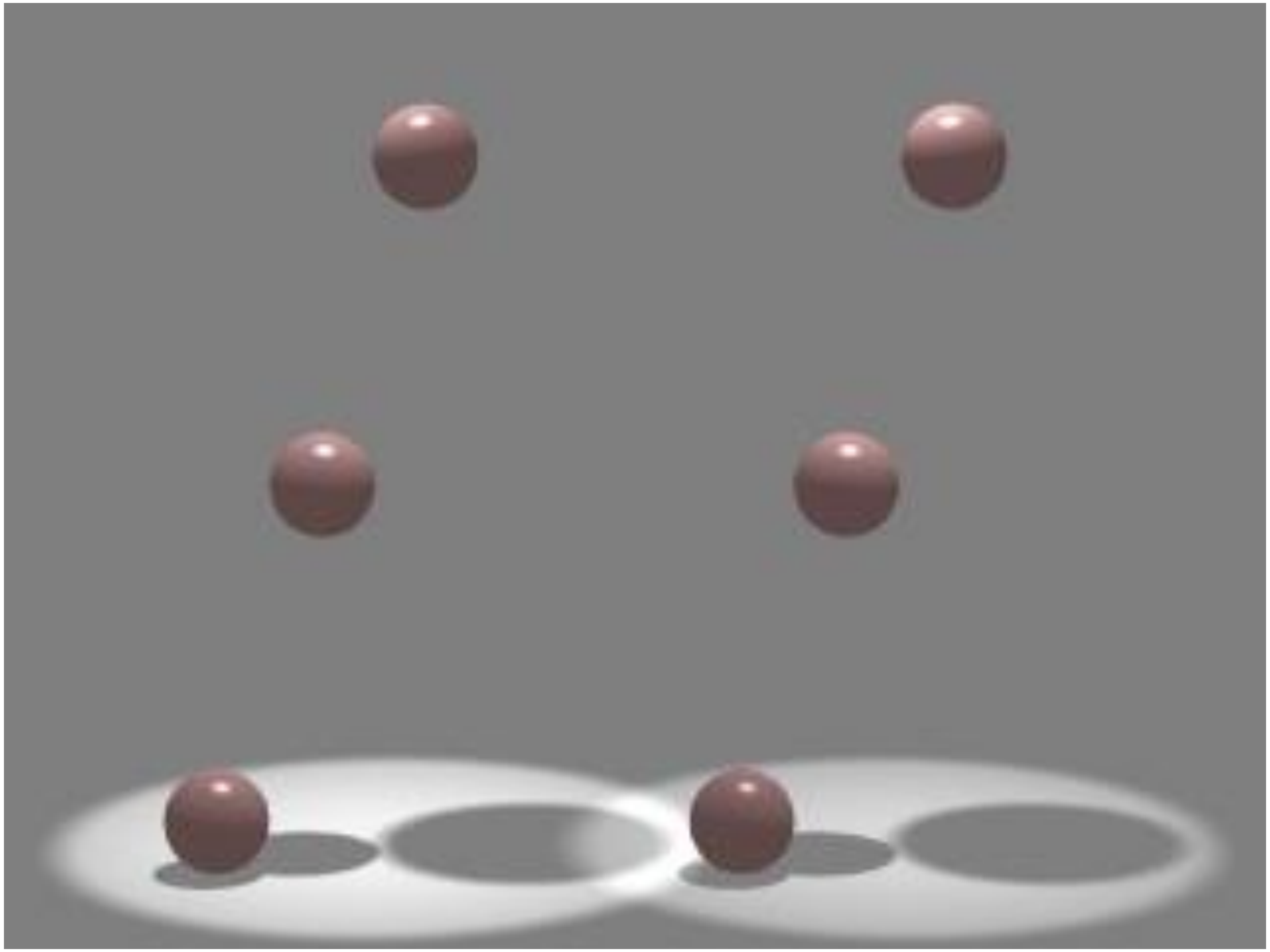




$$\vec{L}(\cos\theta) = \vec{L}\left(s \cdot \frac{P - P_s}{|P - P_s|}\right)$$

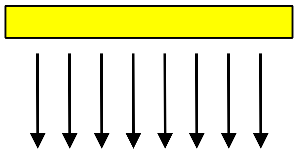
Queen Mary
University of London

2 Spotlights

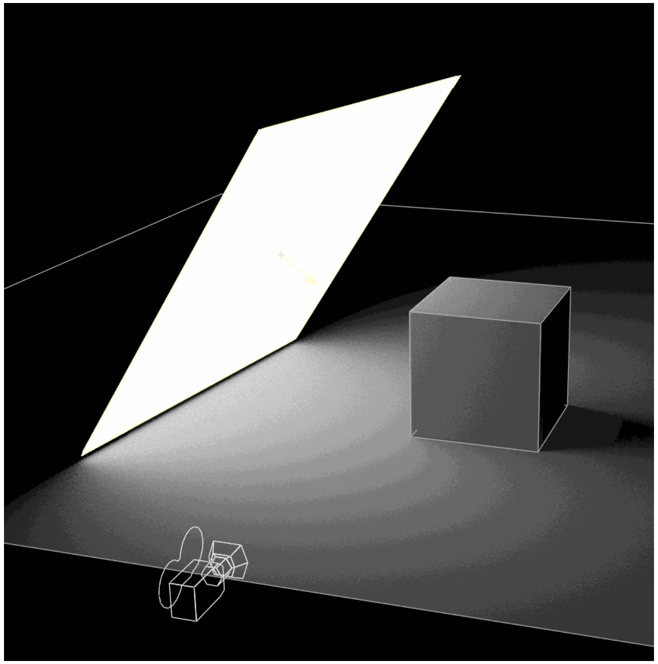


Other light sources

- Area light sources
  - 2-D emissive surface (usually a disc or polygon)
  - example: fluorescent light panels
  - capable of generating soft shadows



Area light



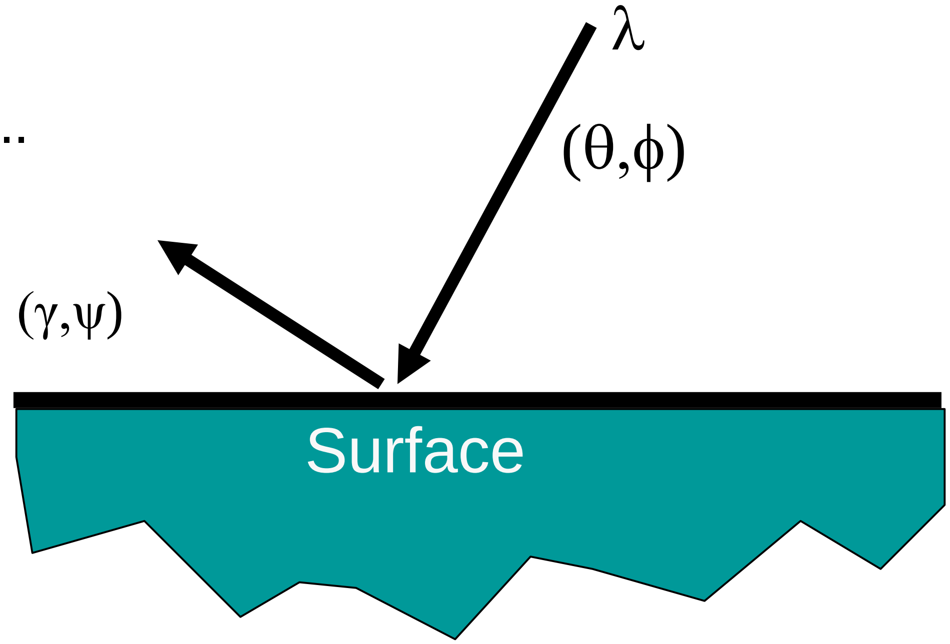
Queen Mary
University of London

Today's agenda

- Why do we need to calculate lighting?
- Definitions
- Ambient and directional light sources
- Diffuse and specular reflection
- Shading models

Modelling surface reflectance

- Incident light: $R_s(\theta,\phi,\gamma,\psi,\lambda)$
  - arriving from direction (θ,ϕ), ...
  - leaving in direction (γ,ψ), …
  - with wavelength λ



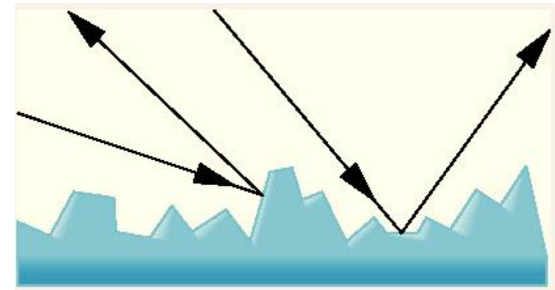
- Surface reflectance, ideally
  - measure radiant energy for "all" combinations of incident angles
    - too much storage
    - difficult in practice

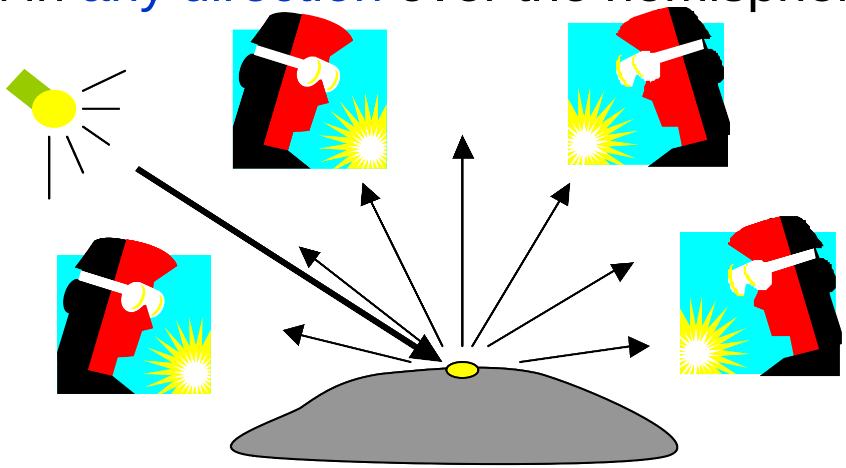
Bi-directional reflection distribution (BRDF)

Queen Mary University of London

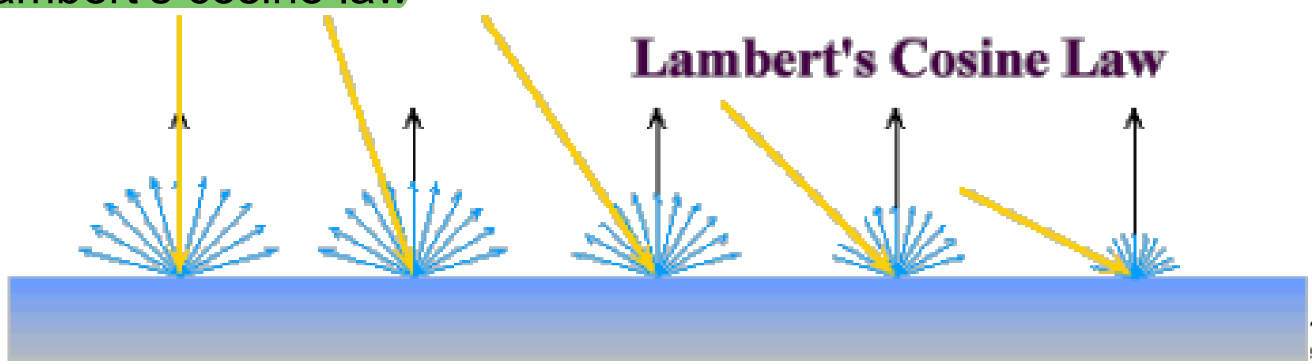
The physics of reflection

- Ideal diffuse reflection
  - is a very rough surface at the microscopic level
  - an incoming ray of light is equally likely to be reflected in any direction over the hemisphere:



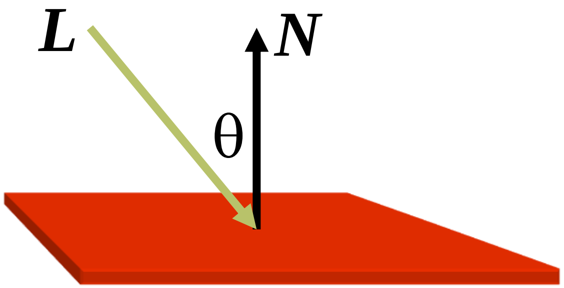


  - The amount of light reflected depends on angle of incident light
    - Lambert’s cosine law



Computing diffuse reflection

- The angle between the surface normal and the incoming light is the *angle of incidence*:



$$I_{diffuse} \cong I_l \cdot k_d \cdot \cos\theta, \quad 0 \le \theta \le \pi/2$$

angle between light vector and surface normal

diffusive reflection constant

incident light intensity

resulting intensity

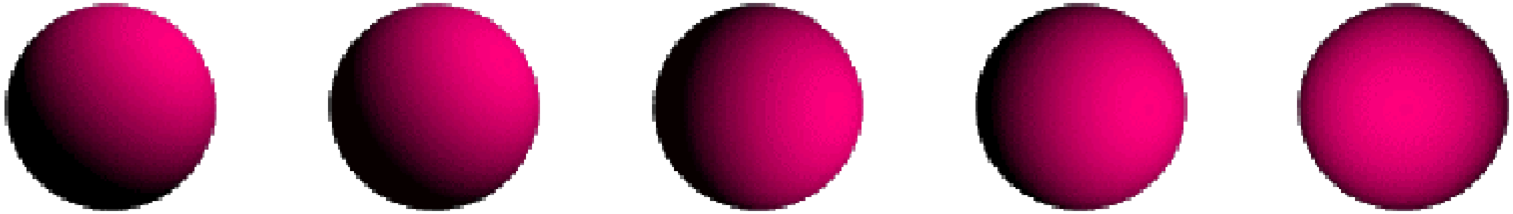
In practice we use vector arithmetic:

- $I_{diffuse} = I_l \, k_d \, (\boldsymbol{N} \bullet \boldsymbol{L})$

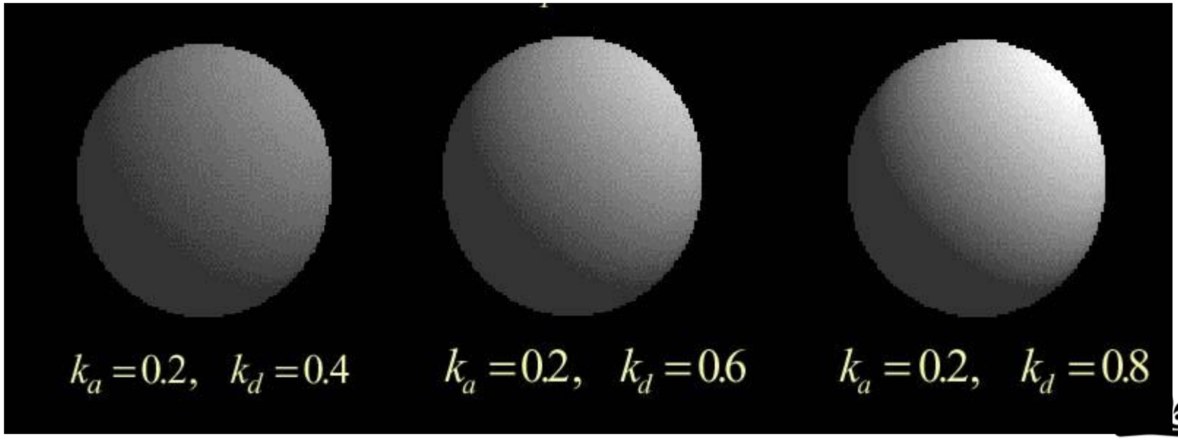
Queen Mary University of London

Diffuse lighting examples

- A Lambertian sphere seen at several different lighting angles (same reflection coefficient):

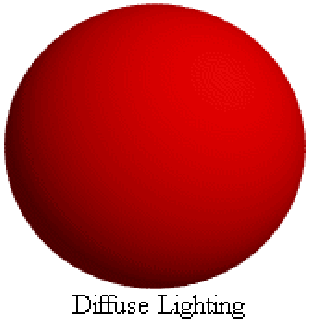


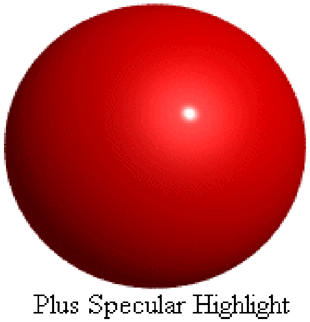
- Lambertian sphere of different coefficient (same lighting angles):



Specular reflection

- Shiny surfaces exhibit specular reflection
  - Polished metal
  - Glossy car finish





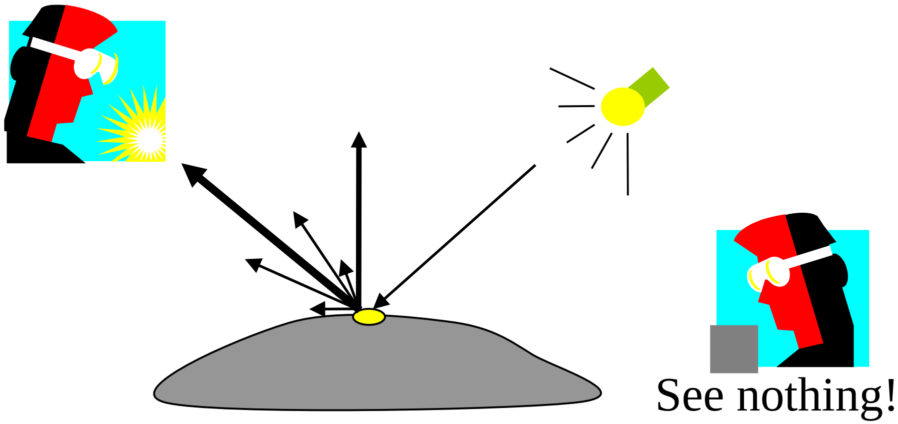
- A light shining on a specular surface causes a bright spot
  (*specular highlight*)
- Where these highlights appear is a function of the viewer's position
  → specular reflectance is view dependent

Queen Mary
University of London

The physics of reflection

- Specular reflecting surface
  - is very smooth at the microscopic level
  - rays of light → likely to bounce off the micro-geometry in a mirror-like fashion
  - The smoother the surface, the closer it becomes to a perfect mirror

  - Reflection is strongest near mirror angle



The optics of reflection

– The incoming ray and reflected ray lie in a plane with the surface normal

– The angle that the reflected ray forms with the surface normal equals the angle formed by the incoming ray and the surface normal:

$$I \cong I_l k_s \cos^n \alpha$$

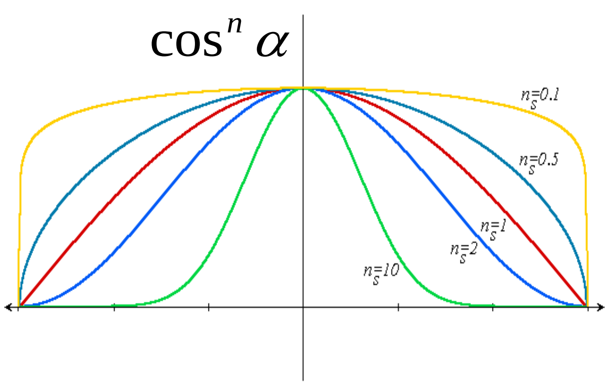
angle between reflection vector and view direction

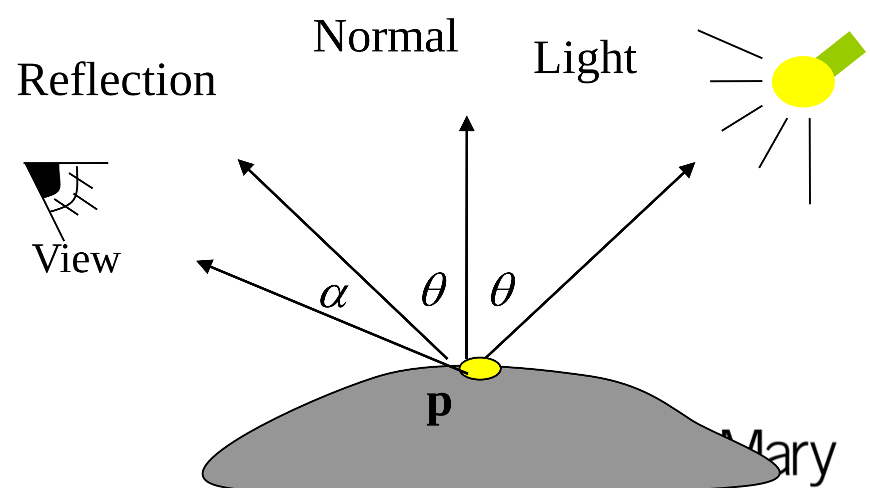
specular reflection exponent (n>=1)

specular reflection constant (0<=k_s<=1)

incident light intensity

resulting intensity



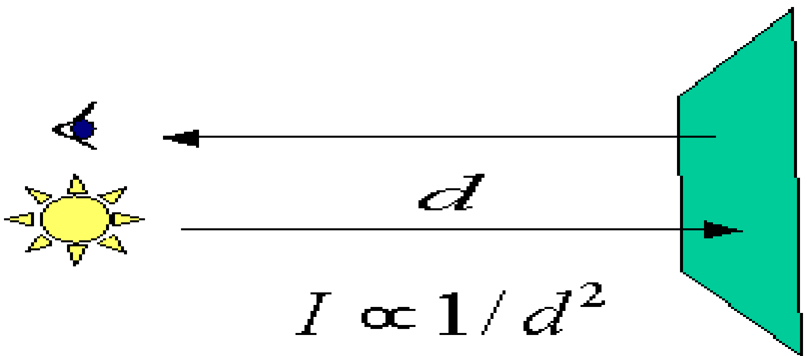


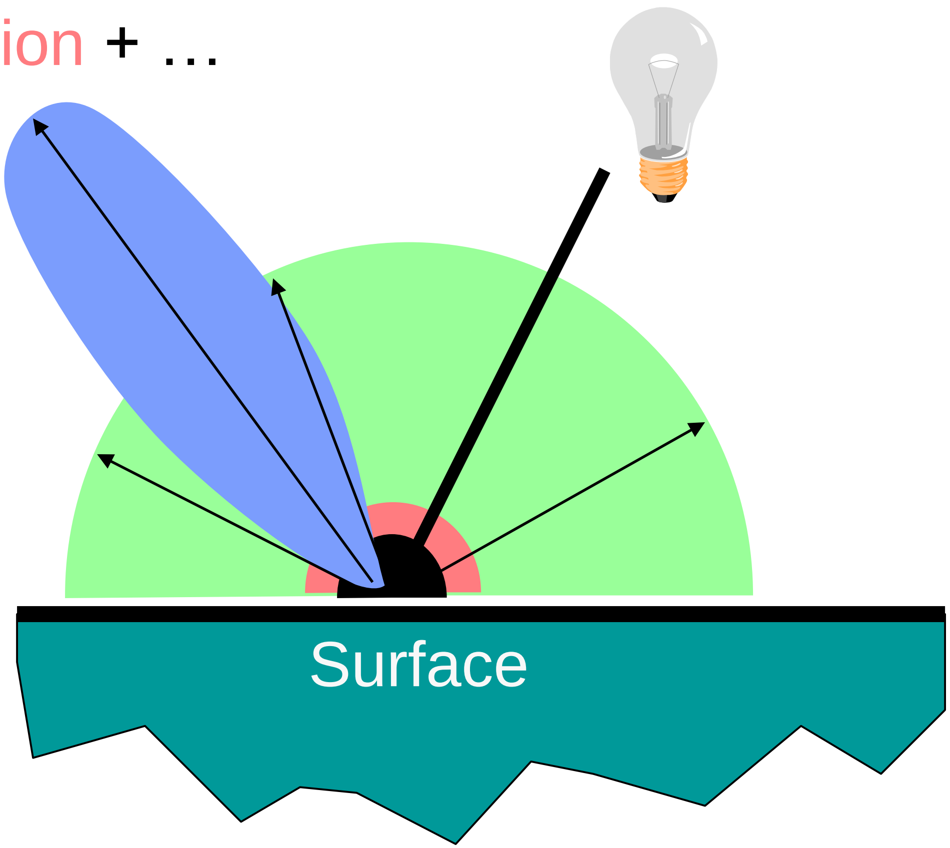
Combining everything

- Simple analytic model:

  ambient + diffuse reflection + specular reflection + emission + …

  with distance attenuation





Combining everything

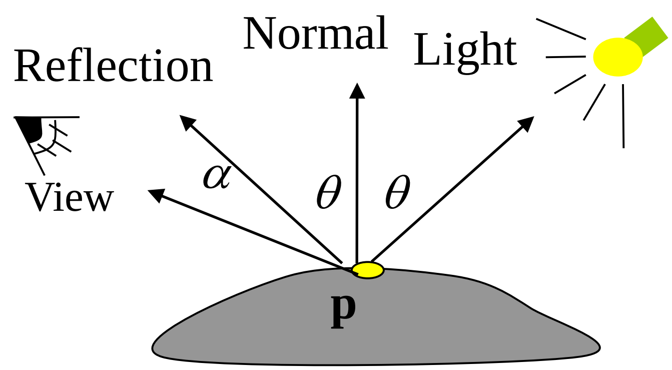
$$I = I_a + f_{att}(I_d + I_s) = I_a \cdot k_a + f_{att} I_l (k_d \cdot \cos\theta + k_s \cdot \cos^n \alpha)$$

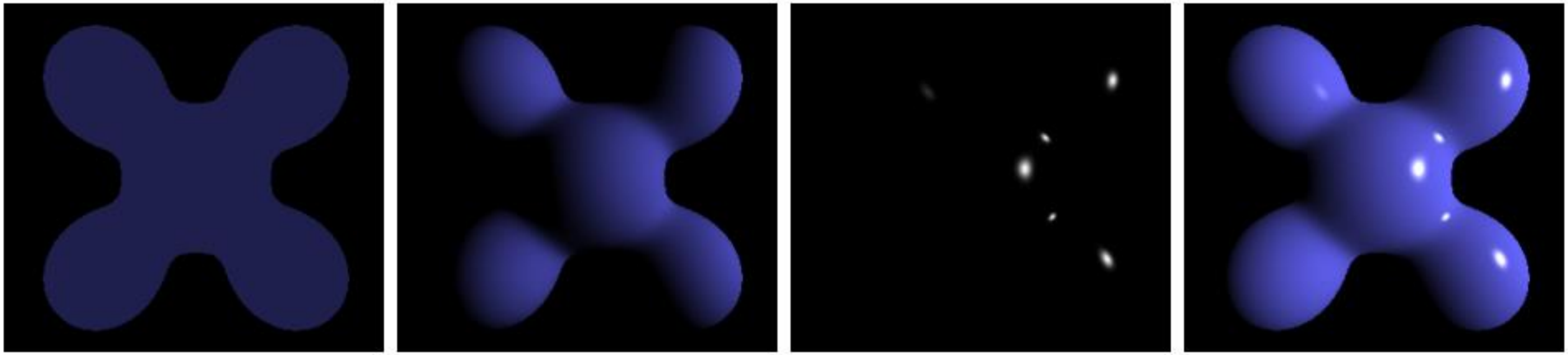
specular reflection

diffusive reflection

attenuation factor

ambient refection





Ambient + Diffuse + Specular Combined illumination

Multiple Colour Light Sources

$$I_\lambda = I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} (k_d \cdot \cos\theta_j + k_s \cdot \cos^n \alpha_j)$$

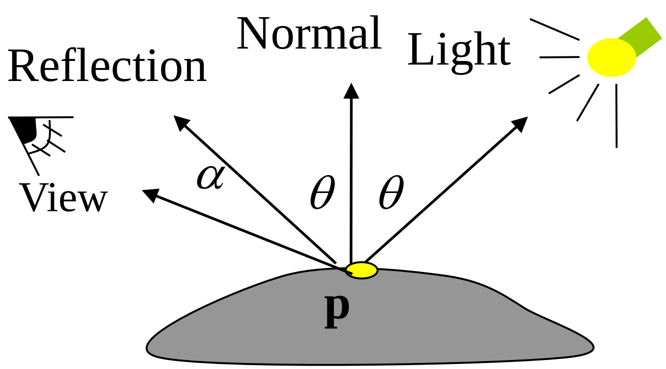
$$= I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} \left(k_d \cdot (\boldsymbol{N} \cdot \boldsymbol{L}_j) + k_s \cdot (\boldsymbol{R}_j \cdot \boldsymbol{V})^n \right)$$

view vector

reflection direction

lighting vector

normal of surface



- The ambient term contributes only once
- Repeat the diffuse and specular calculations for each light source
- Add the components from all light sources

Multiple Colour Light Sources

$$I_\lambda = I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} (k_d \cdot \cos\theta_j + k_s \cdot \cos^n \alpha_j)$$

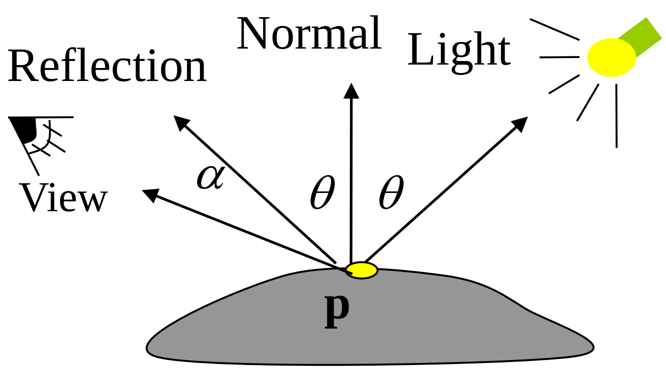
$$= I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} \left(k_d \cdot (\boldsymbol{N} \cdot \boldsymbol{L}_j) + k_s \cdot (\boldsymbol{R}_j \cdot \boldsymbol{V})^n \right)$$

view vector

reflection direction

lighting vector

normal of surface



Householder reflection:

$$\vec{\boldsymbol{R}} = \boldsymbol{H} \cdot \vec{\boldsymbol{L}} \quad where \quad \boldsymbol{H} = \boldsymbol{I} - 2\vec{\boldsymbol{N}} \cdot \vec{\boldsymbol{N}}^T$$

Direct illumination: summary

- Model for
  - determining the brightness (radiance) of a ray rooted at a point on a surface and oriented towards the camera
    - Ambient light contribution term
    - Diffuse light contribution term
    - Specular light contribution term

- Influencing factors
  - Light position
  - Sample point position
  - Camera position
  - Surface angle with respect to light vector
  - Surface angle with respect to camera vector

Direct illumination: changes/unchanged

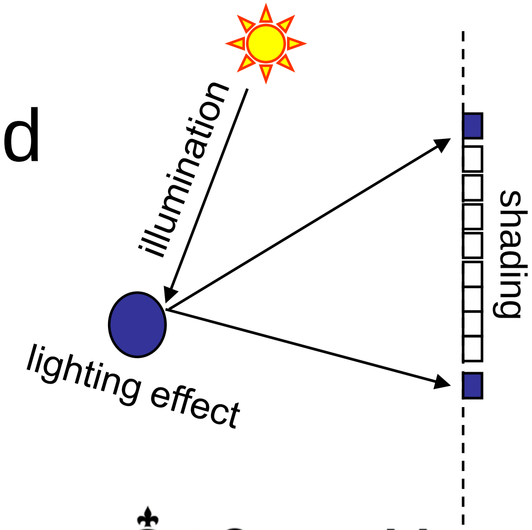
- Camera moves from one position to another
  - Angle between light and surface unchanged
  - Angle between camera and surface changes
- A tracking camera follows object as it moves in scene
  - Angle between light and surface changes
  - Angle between camera and surface unchanged
- An object moves from one position to another
  - Both angles change

Today's agenda

- Why do we need to calculate lighting?
- Definitions
- Ambient and directional light sources
- Diffuse and specular reflection
- Shading models

Shading

- Surfaces are often approximated by <u>polygons</u>, so we use various polygon shading techniques to make these approximations look more like the surfaces as we see in the real world.
- Shading model determines the shade of an object point or pixel by applying the illumination and lighting models
- The models are loosely physical, and emphasize:
  - Empirical success
  - Efficiency

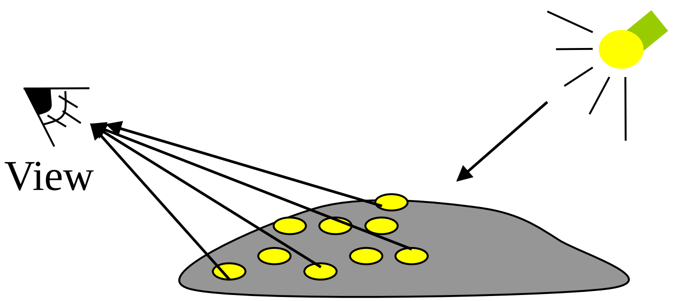


Shading models

Ideally, the renderer should apply lighting models at every visible point on each surface, but this approach requires too much computation.

Alternatively,

- Apply lighting models only to visible surfaces
- Apply lighting models at a subset of points.
- Interpolate the intensity at other points.



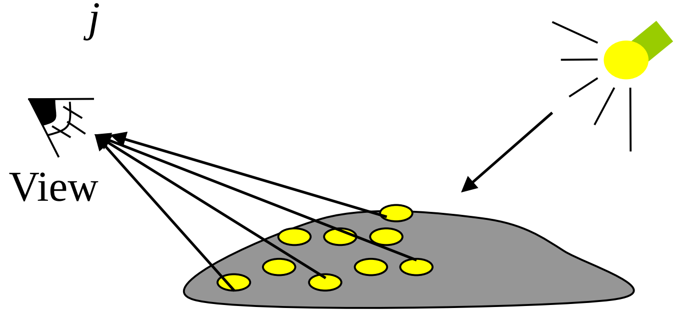
Shading models

Interpolation:

- Flat (Constant) Shading (**Nearest Neighbour Interpolation** of the illumination)
- Gouraud shading (Linear Interpolation of the **illumination**)
- Phong shading (Linear Interpolation of the **normal vectors**)

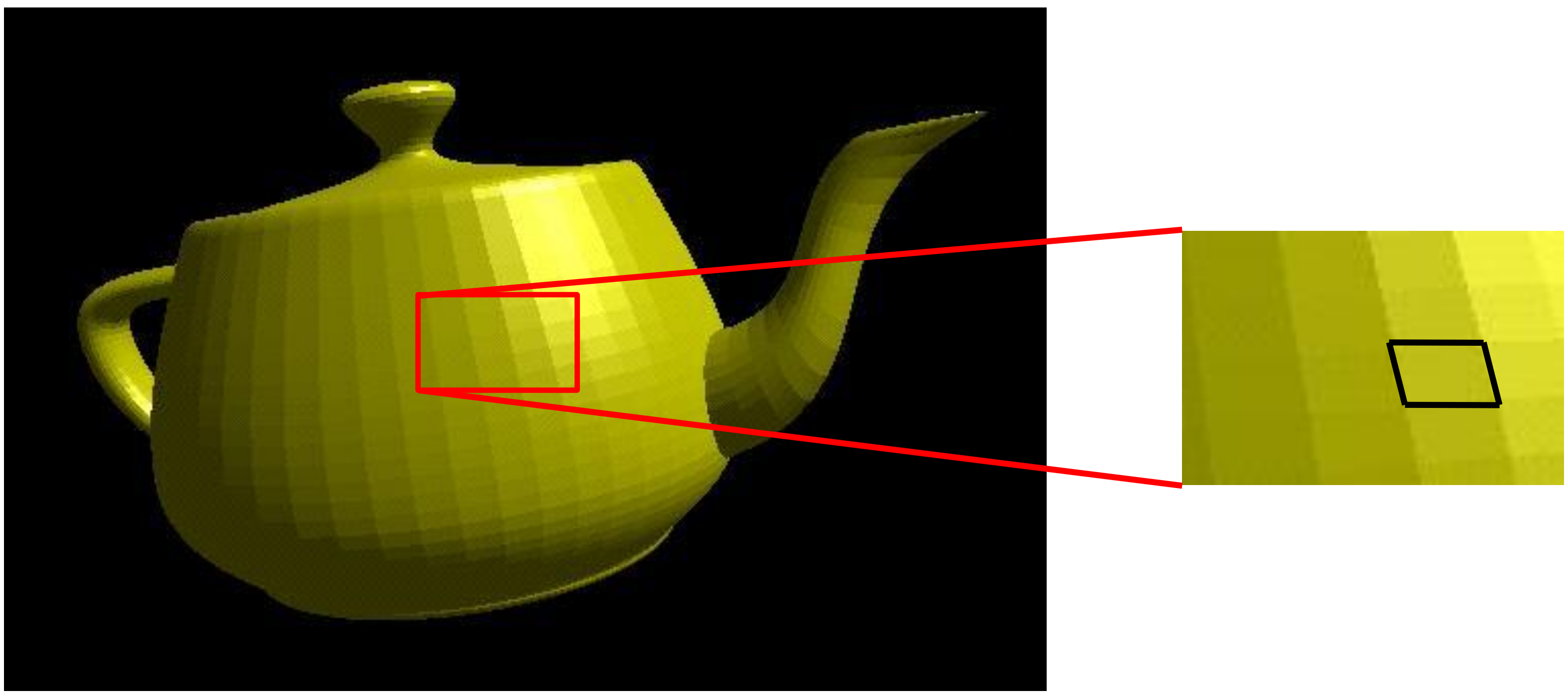
$$I_\lambda = I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} (k_d \cdot \cos\theta_j + k_s \cdot \cos^n \alpha_j)$$

$$= I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} \left(k_d \cdot (\vec{\boldsymbol{N}} \cdot \vec{\boldsymbol{L}}_j) + k_s \cdot (\vec{\boldsymbol{R}}_j \cdot \vec{\boldsymbol{V}})^n \right)$$



Flat (Constant) Shading

- Compute lighting once at one point per polygon and assign the color to the whole polygon.

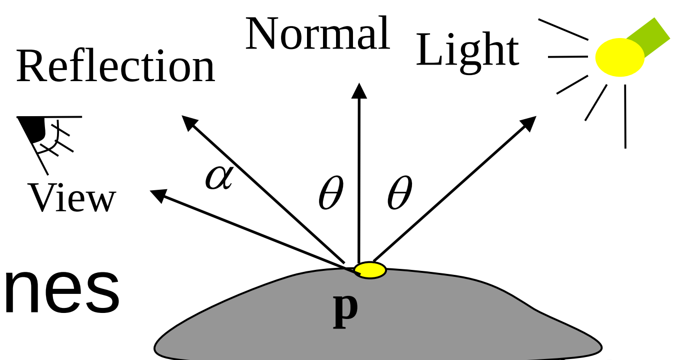


About Flat Shading

- Intensity interpolation: use constant
- Flat shading is good if:
  - The surface is faceted, not curved: ***N*** is constant.
  - Light source is very far away: ***L*** and ***R*** are constant
  - The viewer is very far away: ***V*** is constant

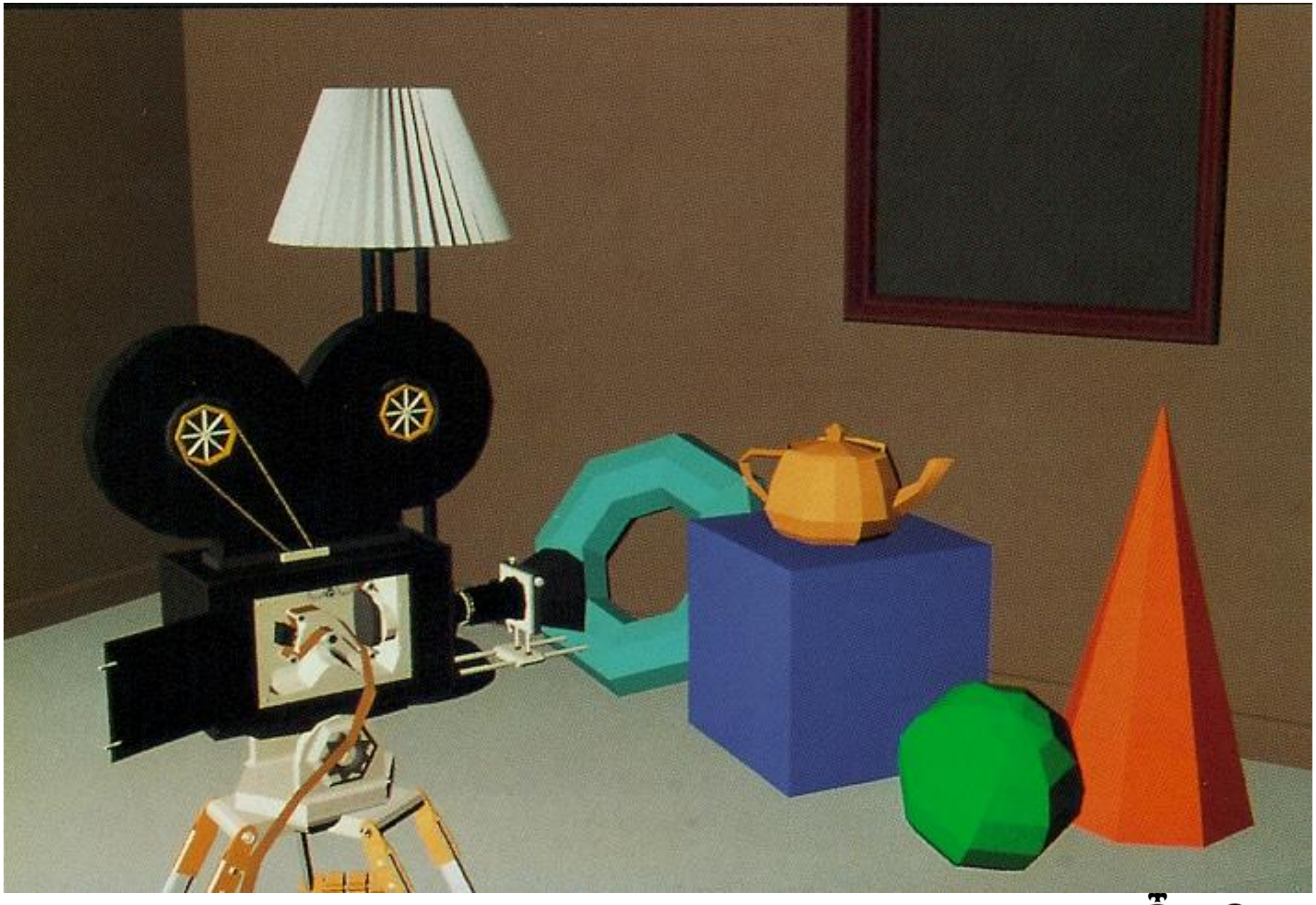
$$I_\lambda = I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} (k_d \cdot \cos\theta_j + k_s \cdot \cos^n \alpha_j)$$

$$= I_{a\lambda} \cdot k_a + \sum_j f_{att_j} I_{l\lambda_j} \left(k_d \cdot (\vec{\boldsymbol{N}} \cdot \vec{\boldsymbol{L}}_j) + k_s \cdot (\vec{\boldsymbol{R}}_j \cdot \vec{\boldsymbol{V}})^n \right)$$



- Good for coarse preview of scenes

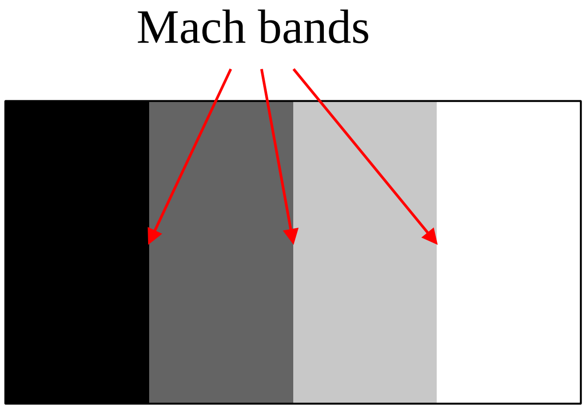
Flat Shading Example

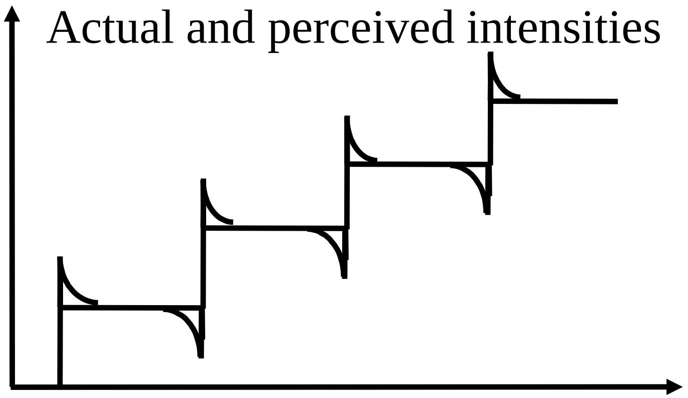


Queen Mary
University of London

Problem: Mach Band Effect

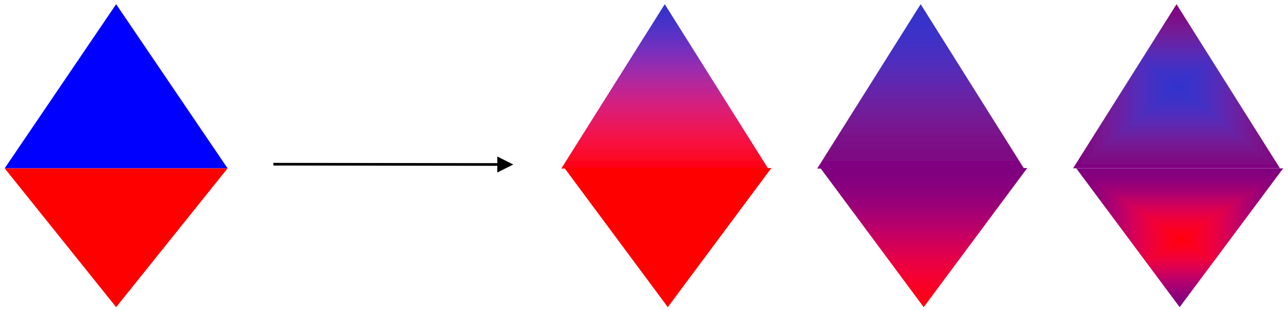
- Flat shading suffers from a perceptual artifacts – "Mach band effect"
- Mach band effect – for human eyes, imaginary dark and light lines appear at the face boundaries
- Intensity change is exaggerated : Dark facet looks darker and lighter looks even more lighter





Solution: Smooth Shading

- Fix the Mach band effect – remove edge discontinuity
- Compute at more points on each face

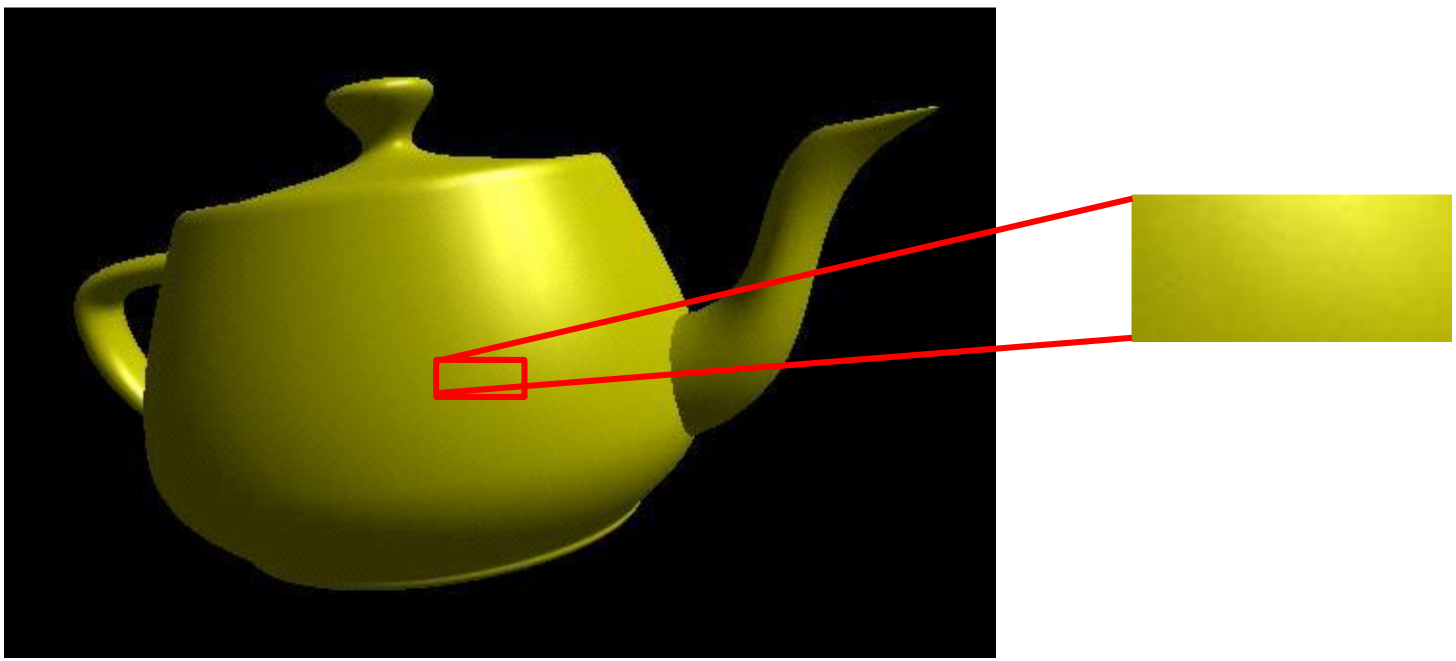


Flat shading Smooth shading

- Two popular methods: Gouraud, Phong

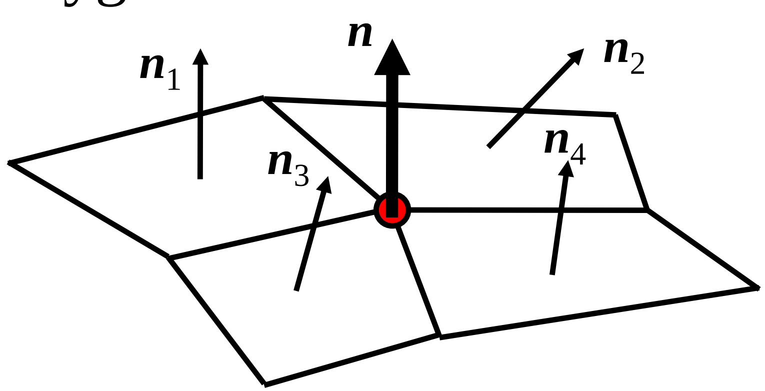
Gouraud Shading Idea

- Also called ***intensity interpolation shading***
- Lighting is calculated for each of the polygon vertices, colours are interpolated for interior pixels



About Gouraud Shading

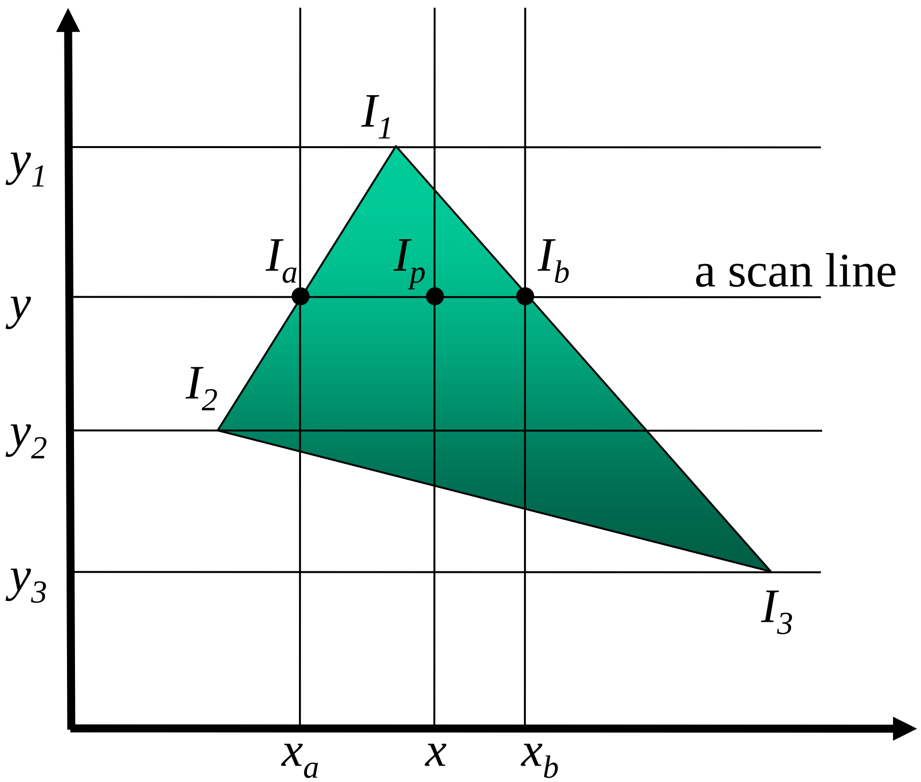
- Intensity interpolation: linear across a face
- Lighting calculation is only for vertices. Normals are only needed for each vertex of polygons.
- The normal at each vertex is the average of the normals of its adjacent faces.
- Fast, supported by most of the graphics accelerator
- Valid for small polygons



$$\boldsymbol{n} = (\boldsymbol{n}_1 + \boldsymbol{n}_2 + \boldsymbol{n}_3 + \boldsymbol{n}_4) / 4$$

Gouraud Shading Interpolation

- Use bilinear interpolation to find intensity I_p at a point along polygon edges and scan lines.

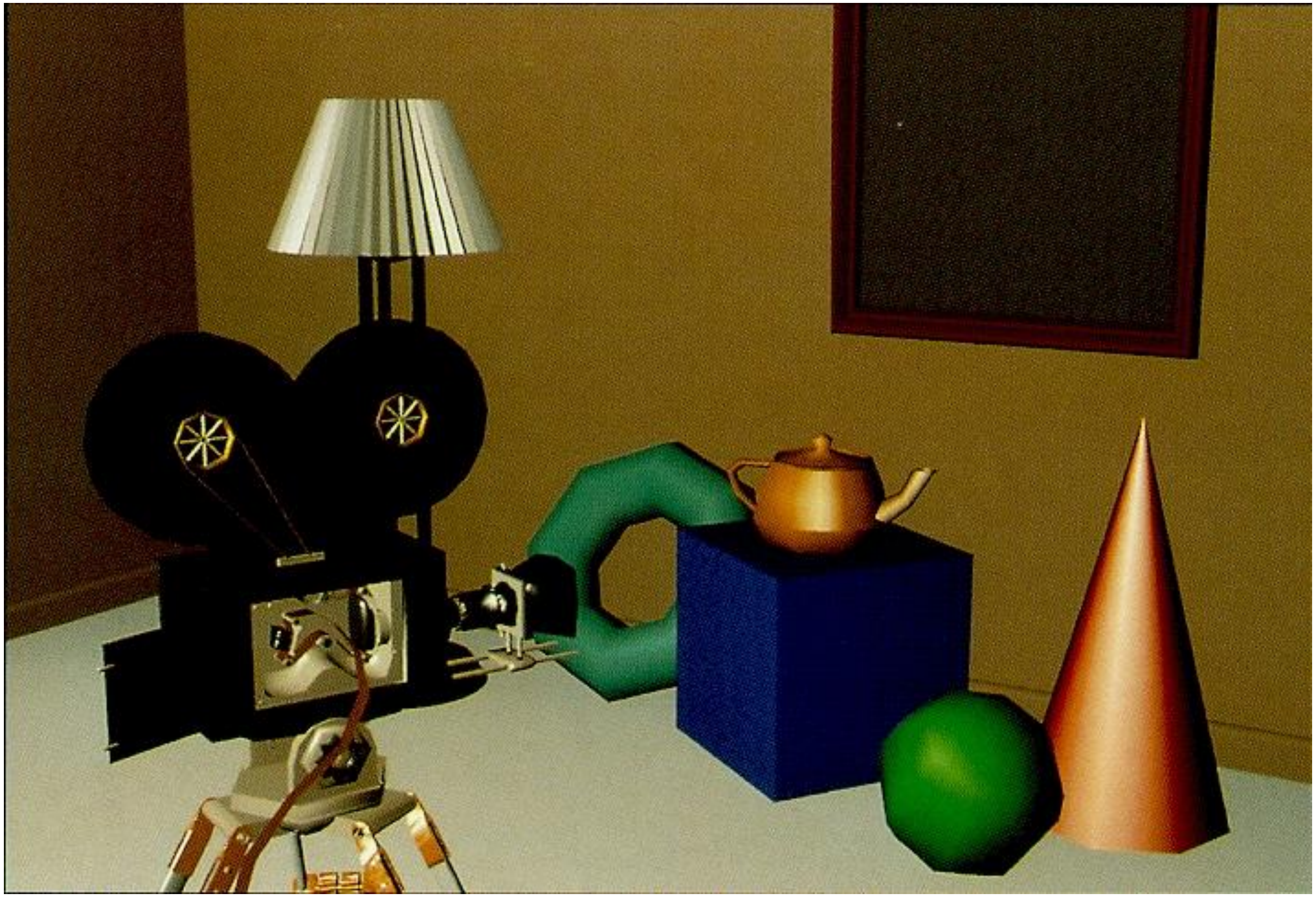


$$I_a = I_1 + (I_2 - I_1)\frac{y - y_1}{y_2 - y_1}$$

$$I_b = I_1 + (I_3 - I_1)\frac{y - y_1}{y_3 - y_1}$$

$$I_p = I_a + (I_b - I_a)\frac{x - x_a}{x_b - x_a}$$

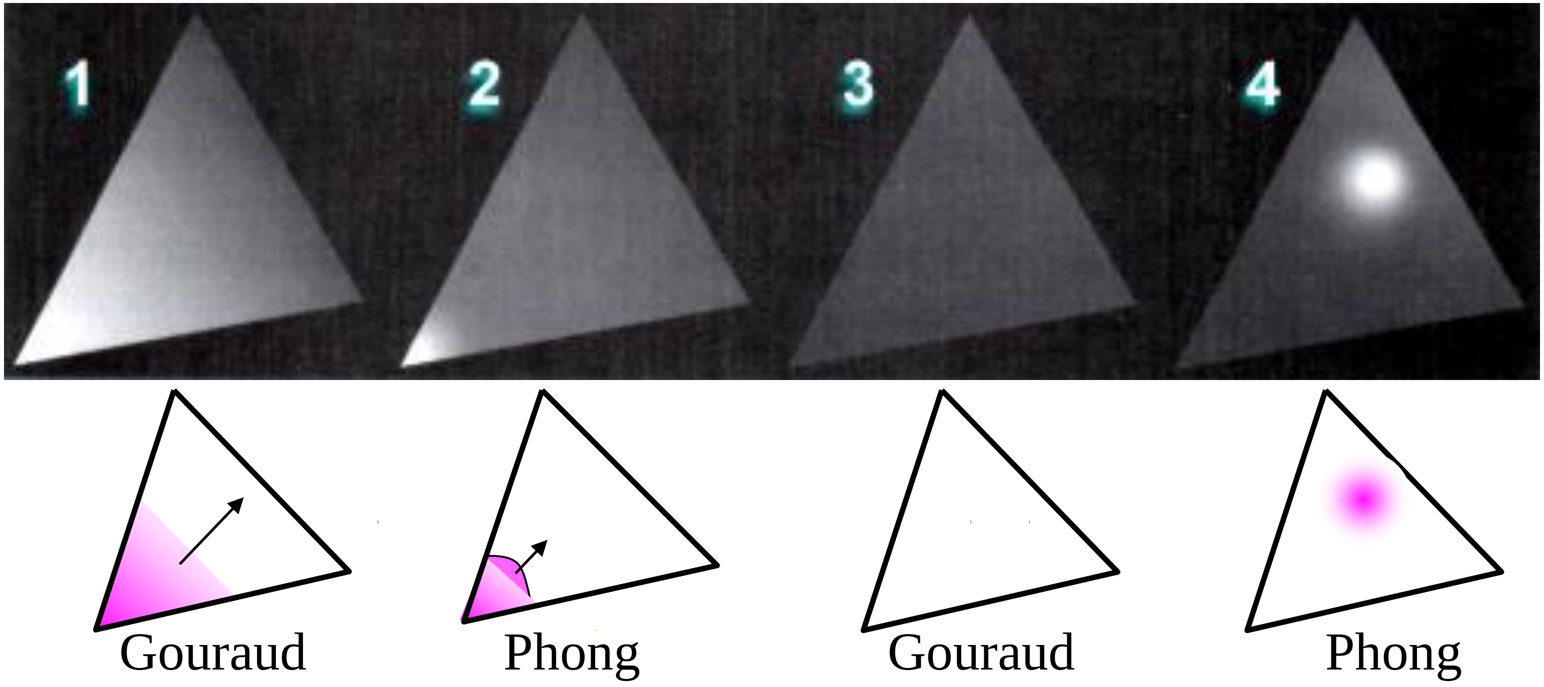
Gouraud Shading Example



Queen Mary
University of London

Gouraud Shading Problems

- Reduces Mach bands (but not entirely).
- May miss interior specular highlights if it's not at vertices.
- May spread specular highlights along edges.
- Some repetitive 3D patterns can be missed completely.



What happened?

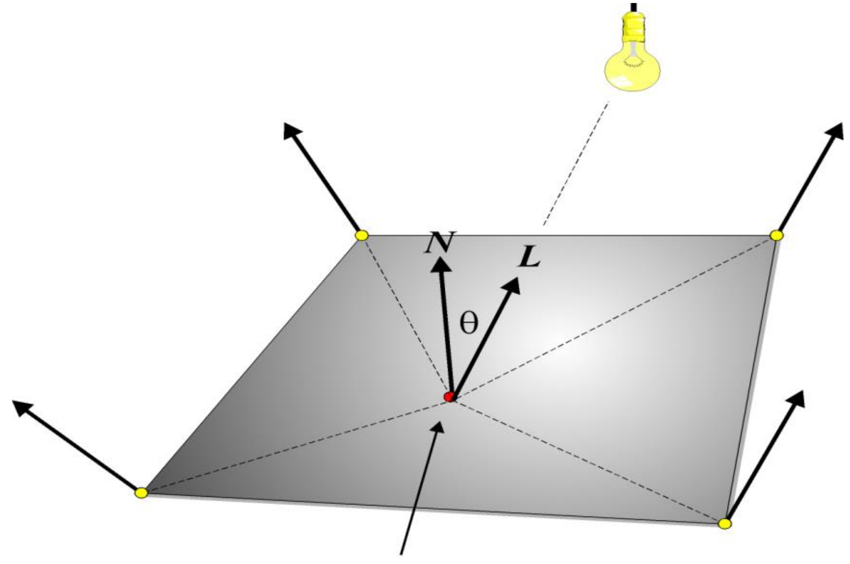
- Gouraud shading method:
  1. Compute the normal at each vertex
  2. Compute the intensity at the vertices with a lighting model with the normals
  3. Interpolate the intensity across a face with those at vertices.
- Gouraud method applies the lighting model at each vertex, then interpolates pixel **intensities**
  - This requires a normal **only at the vertices**

It didn’t use the **normals** in faces!

- We need to apply the lighting model at all pixels
  - This requires a normal (***N***) at each pixel, we need all the not-provided normals
  - ***N*** at each point can be interpolated from ***N*** at vertices

Phong Shading

- Also called ***normal-vector interpolation shading***
- Involves the following steps
  - A normal at a vertex is computed as the average of the normals of all the adjacent faces
  - Normals at each interior pixel are calculated by linear interpolation of the normals at the vertices

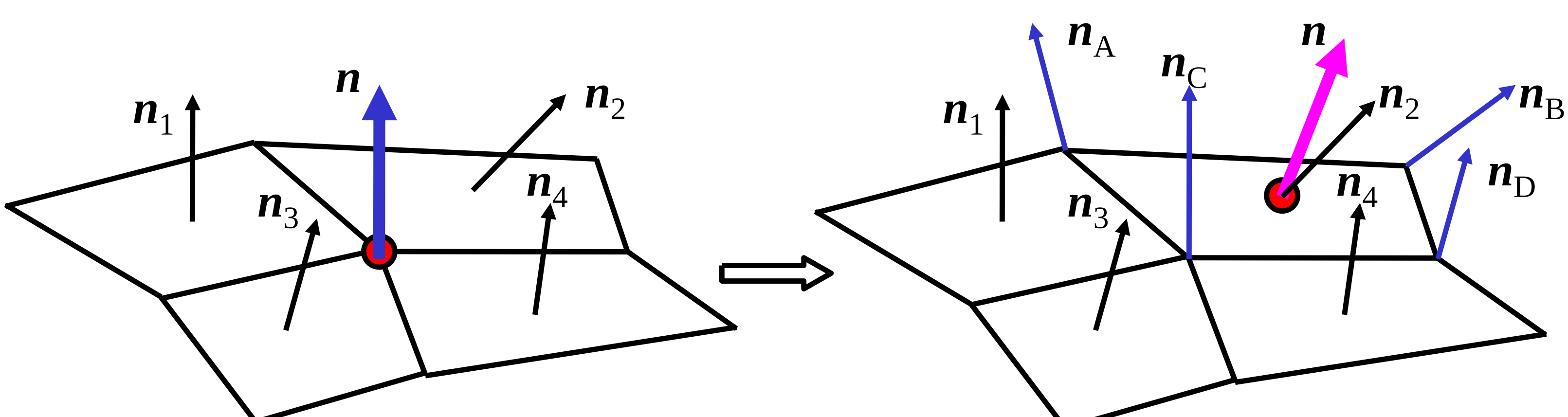


Interpolated normal at the interior point

Queen Mary University of London

About Phong Shading

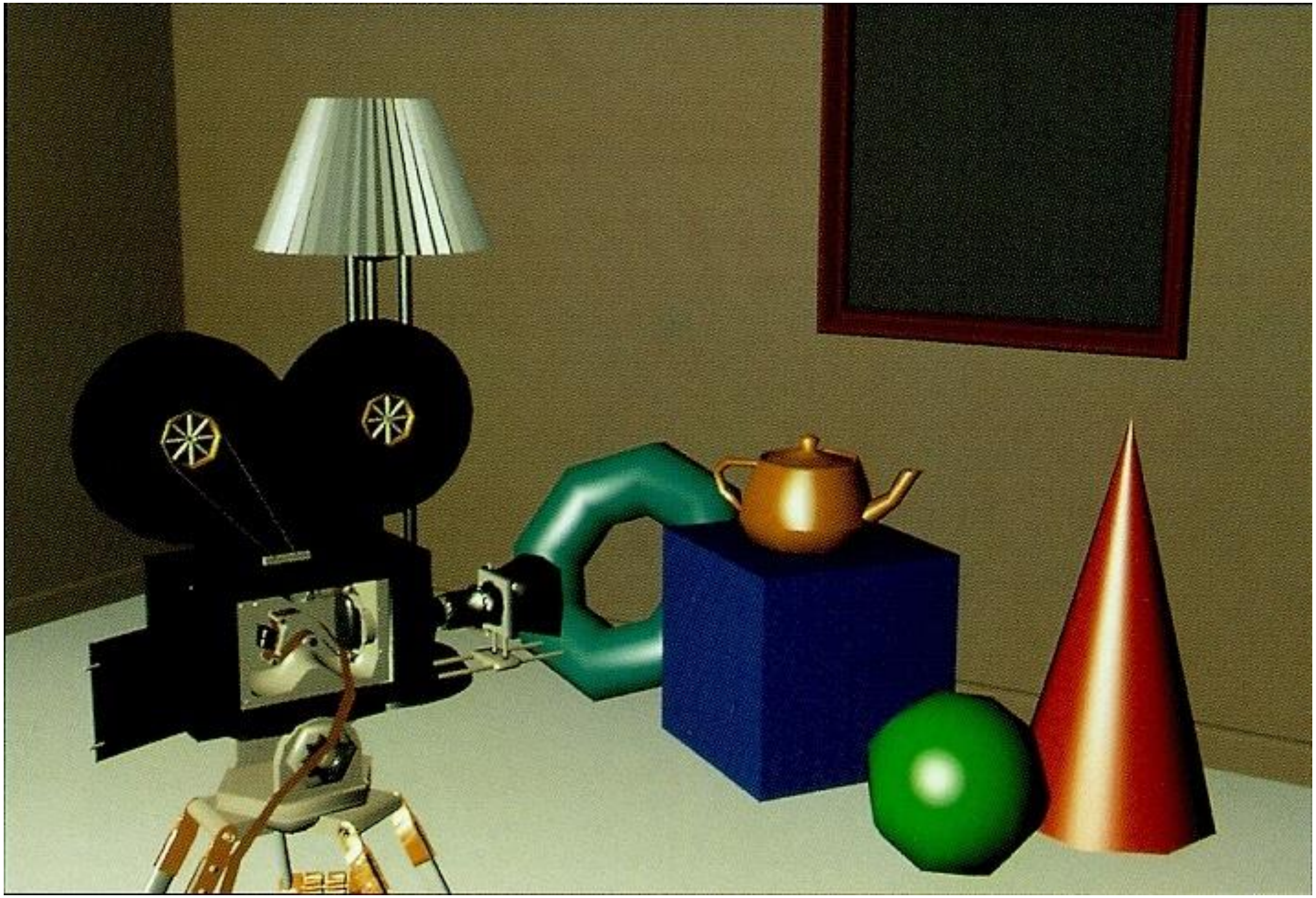
- Normal interpolation: bilinear across a face
- Interpolation is done after or before perspective transform
- Interpolation of normals is done like intensity interpolation in Gouraud shading
- More accurate, but slow. Not widely supported by hardware
- Good for smooth surfaces



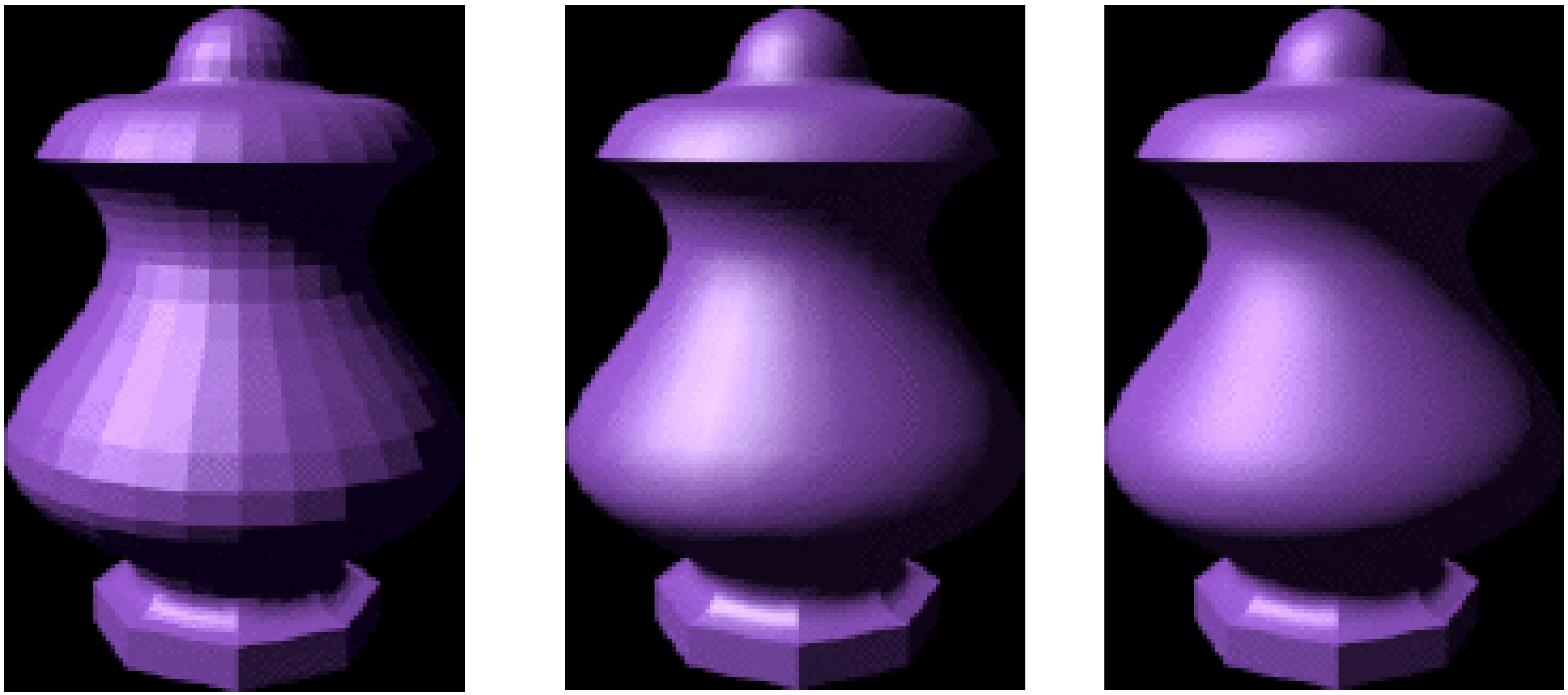
$\boldsymbol{n} = (\boldsymbol{n}_1 + \boldsymbol{n}_2 + \boldsymbol{n}_3 + \boldsymbol{n}_4) / 4$

$\boldsymbol{n} = a\boldsymbol{n}_A + b\boldsymbol{n}_B + c\boldsymbol{n}_C + d\boldsymbol{n}_D$

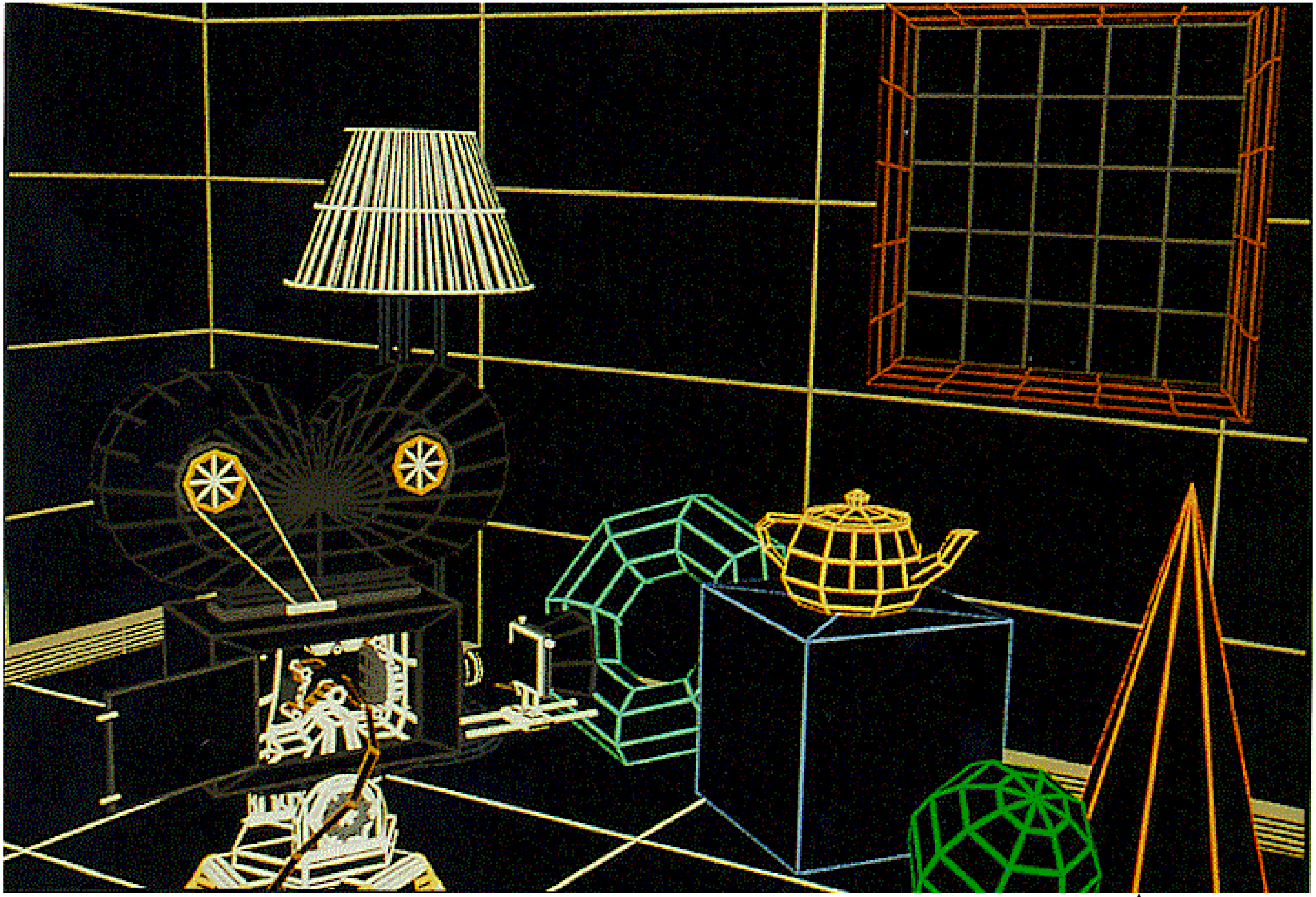
Phong Shading Example



Flat, Gouraud and Phong Shading

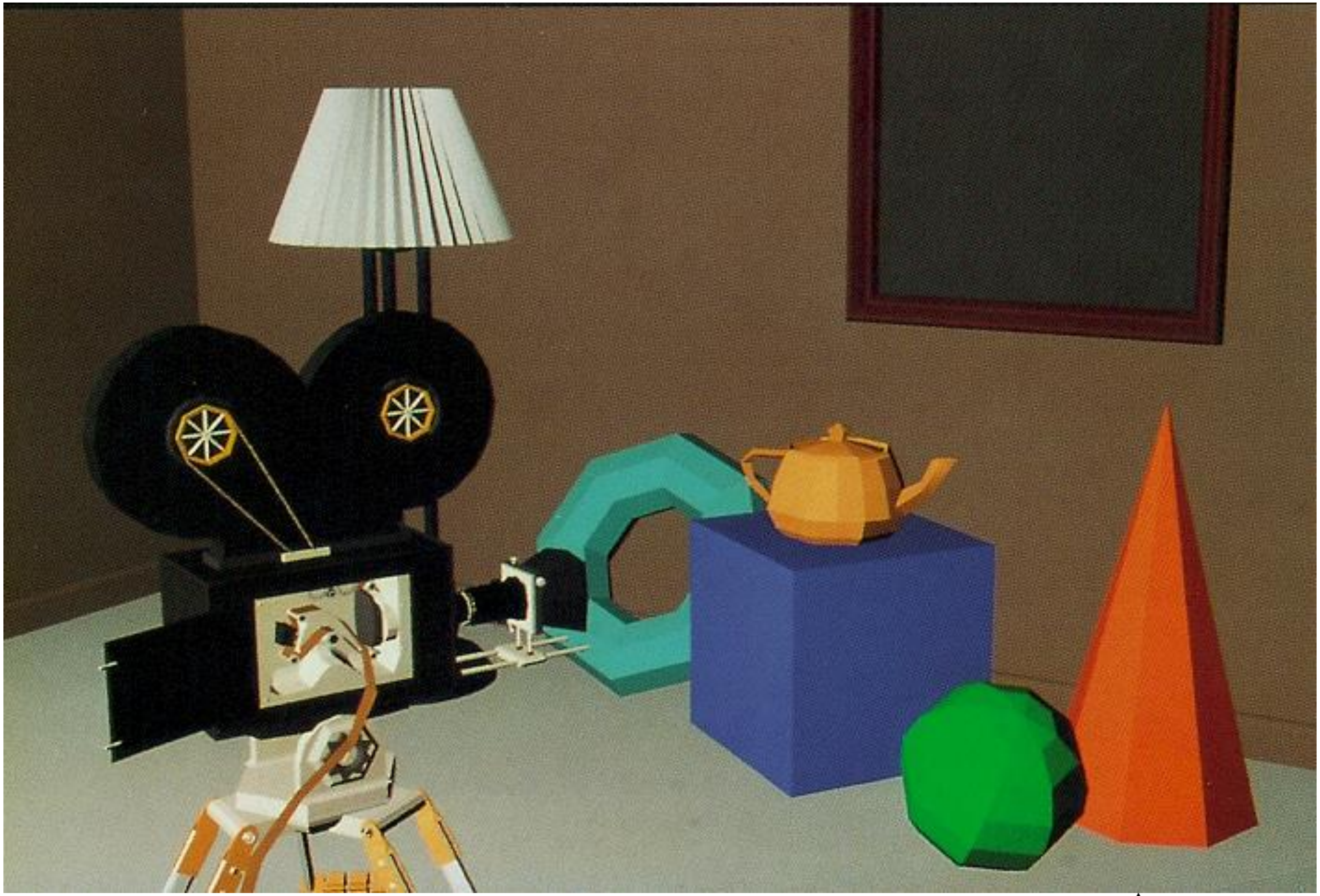


Coloured lines



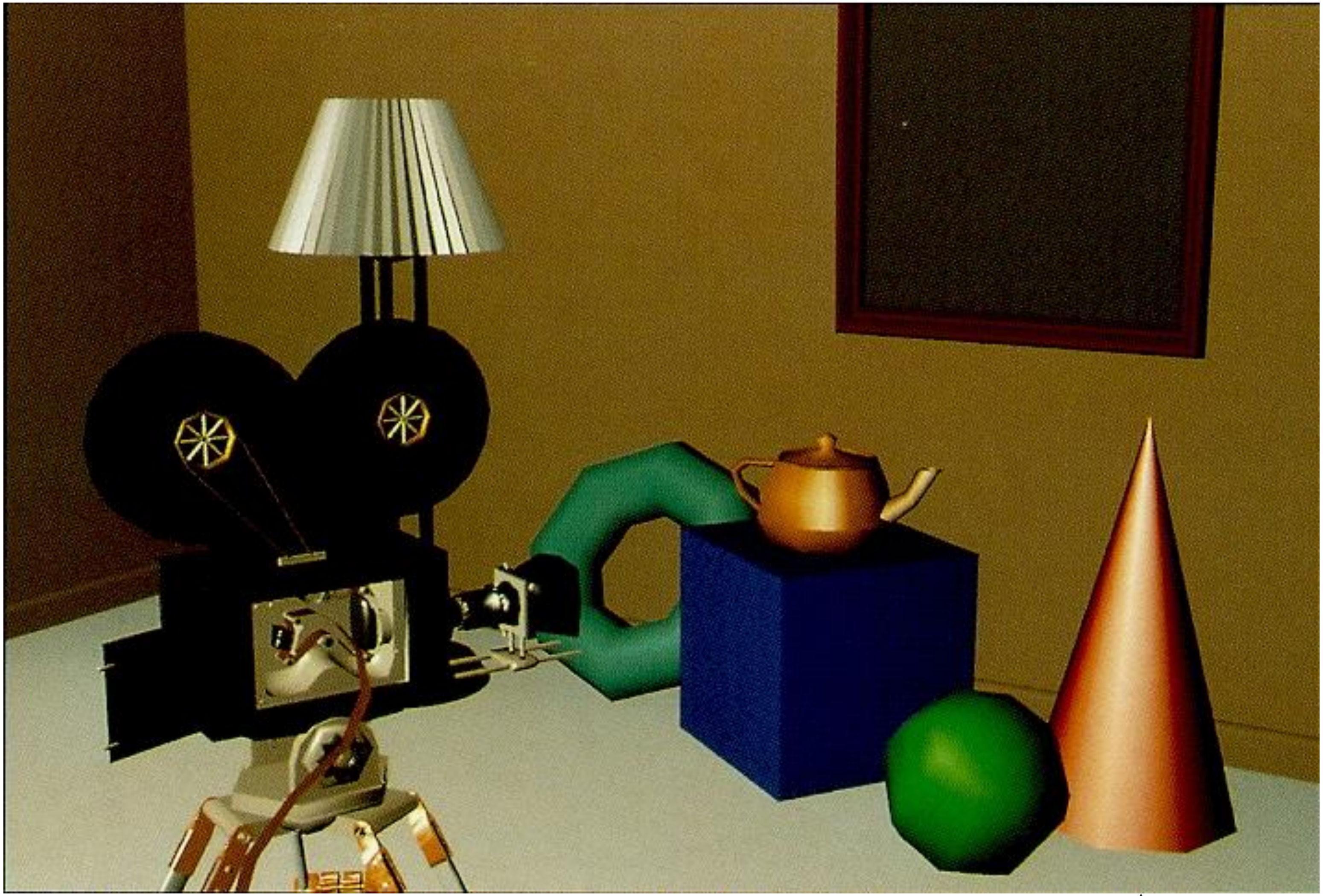
Ambient lighting

Flat shading



Queen Mary
University of London

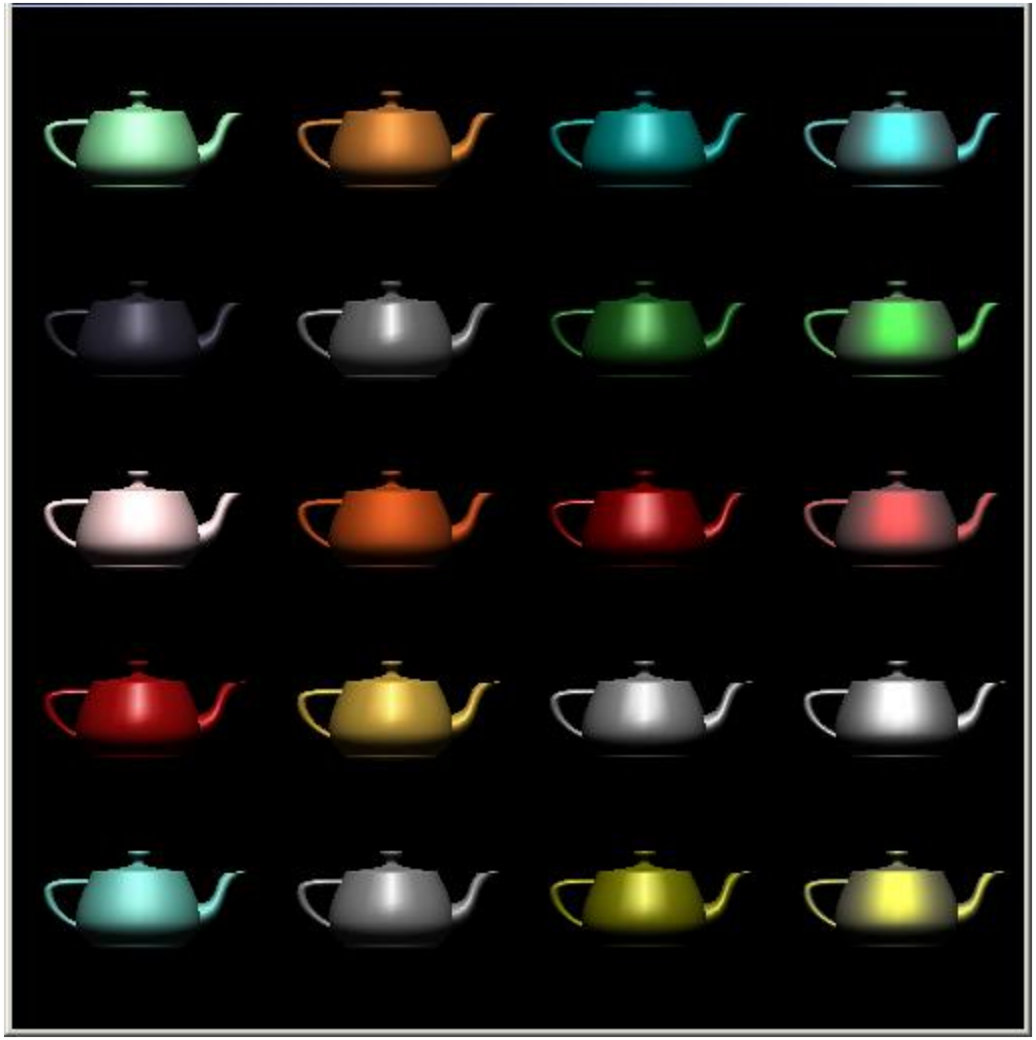
Gouraud shading



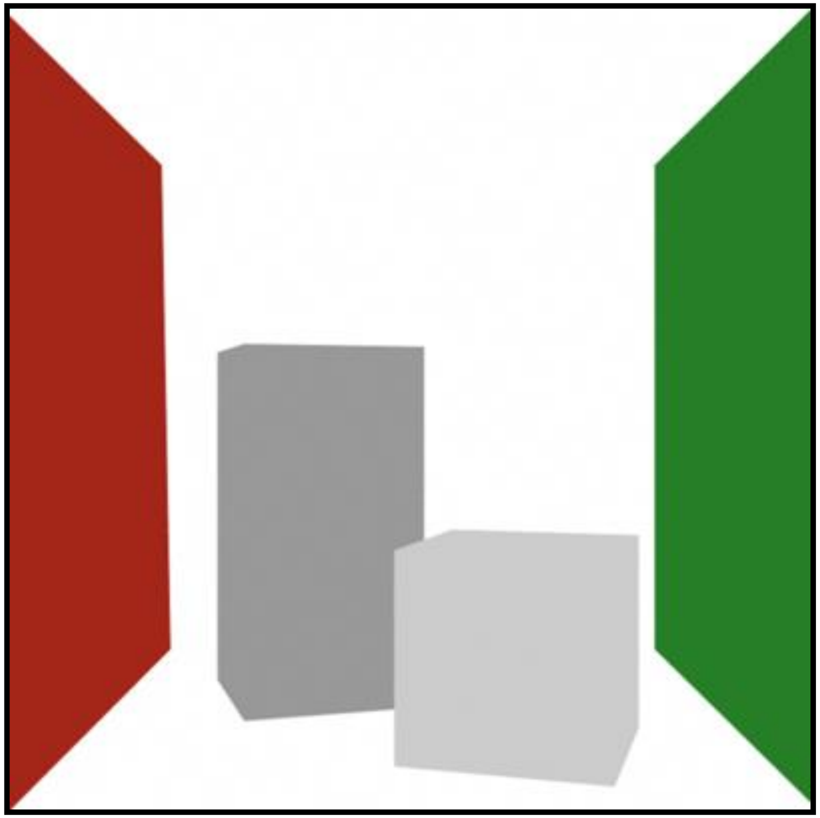
Queen Mary
University of London

Phong shading

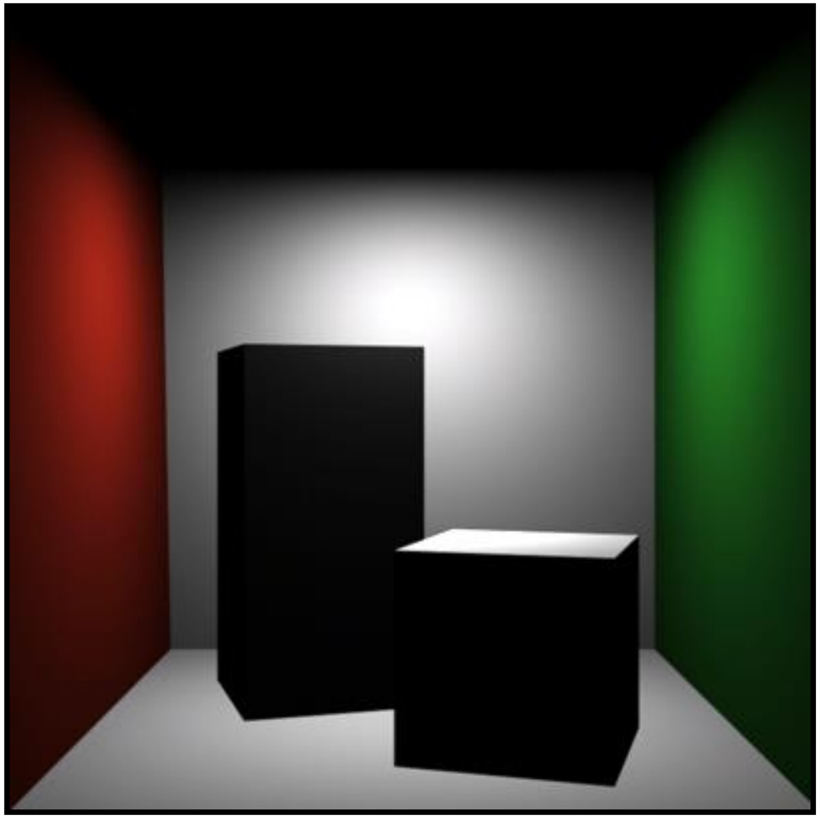
Teapot Examples



Lighting example: diffuse reflection



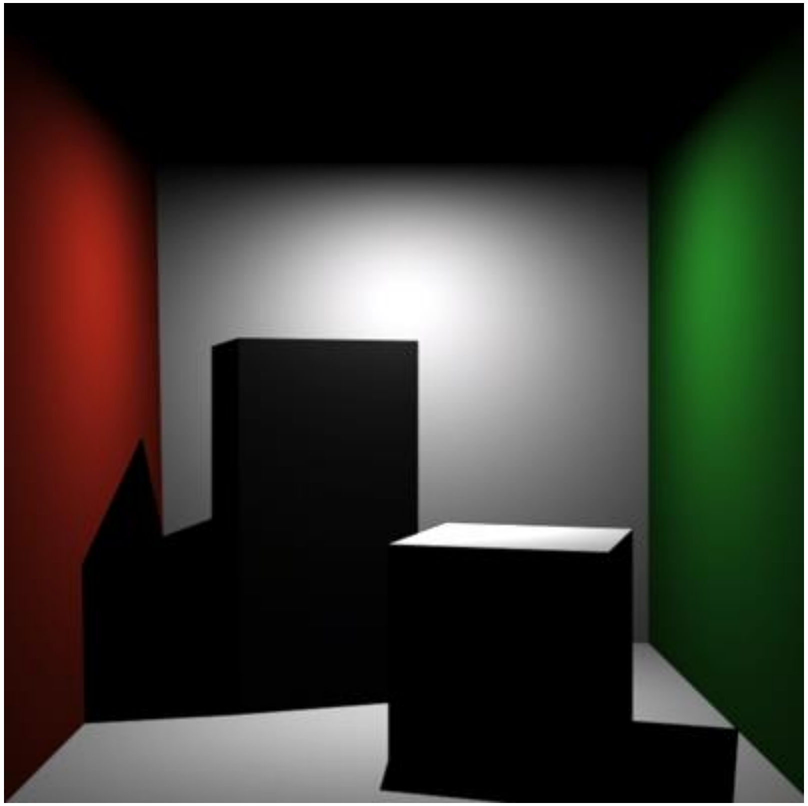
Surface Color



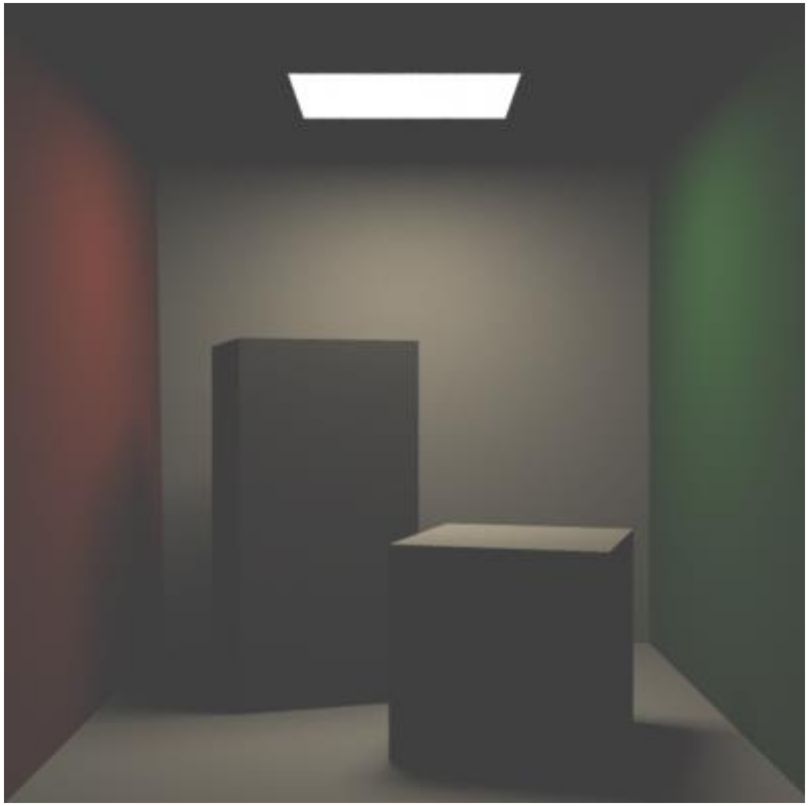
Diffuse Shading

Queen Mary University of London

Lighting example: soft shadows



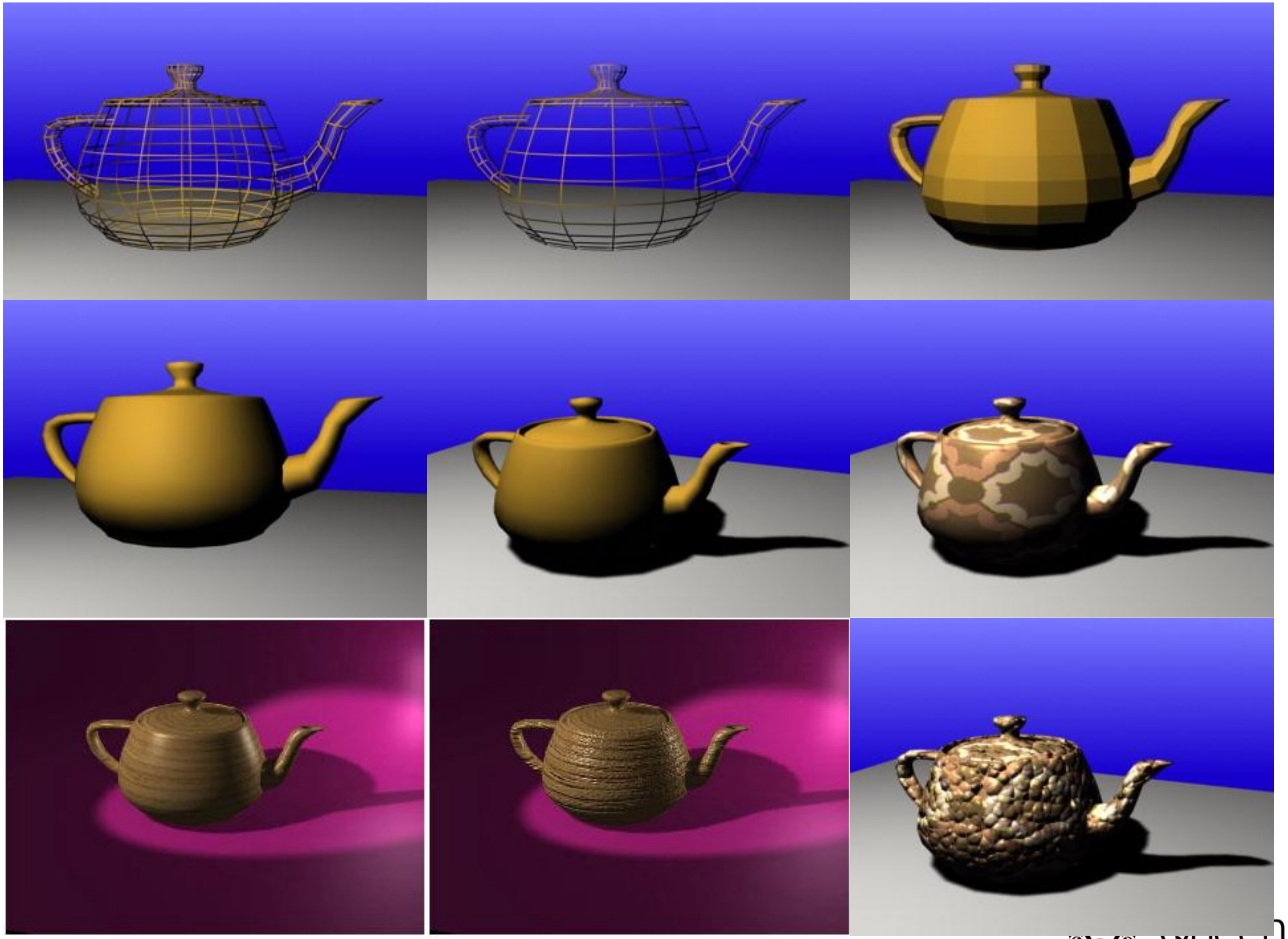
Hard Shadows
Point Light Source



Soft Shadows
Area Light Source

With more surface details

The whole story



Queen Mary
University of London