Interactive Computer Graphics: Lecture 6

Illumination and Shading

The Graphics Pipeline

Modelling Transformations

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

Projection (to Screen Space)

Scan Conversion (Rasterization)

Visibility / Display
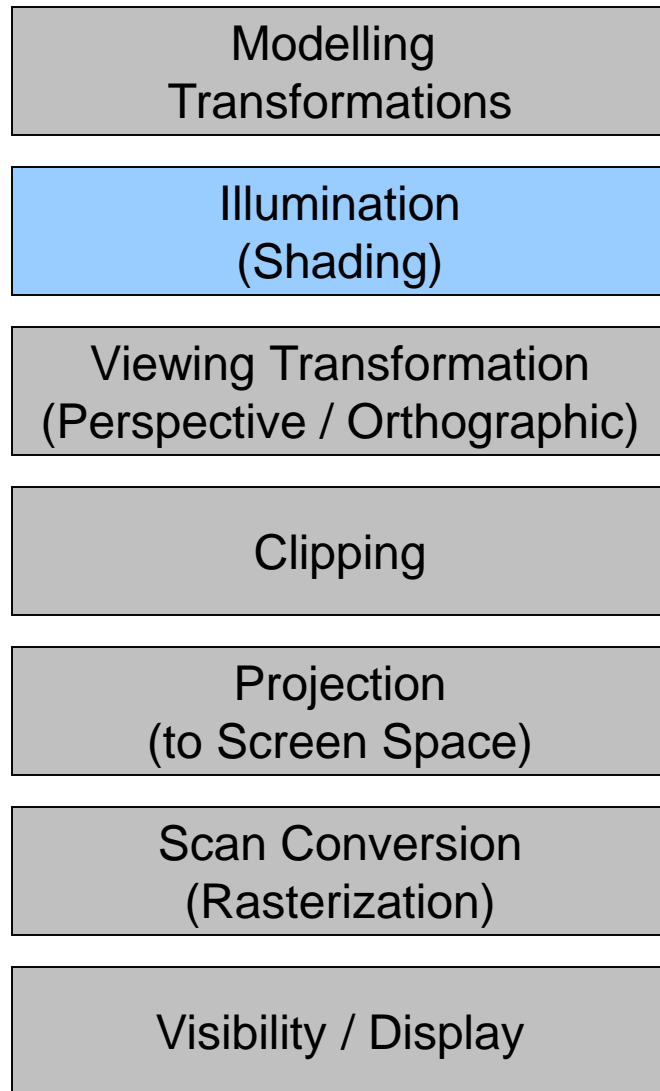
- Vertices are lit (shaded) according to material properties, surface properties and light sources
- Uses a local lighting model

Graphics Lecture 4: Slide 2
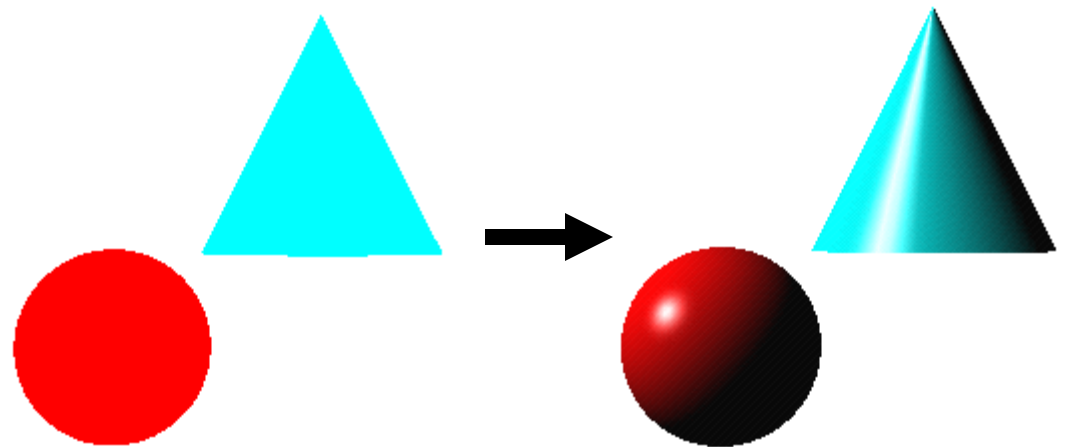
The Physics of shading

- If we look at a point on an object we perceive a colour and a shading intensity that depends on the various characteristics of the object and the light sources that illuminate it.

- For the time being we will consider only the brightness at each point. We will extend the treatment to colour in the next lecture.

The Physics of shading

- Object properties:
  - Looking at a point on an object we see the reflection of the light that falls on it. This reflection is governed by:
    1. The position of the object relative to the light sources
    2. The surface normal vector
    3. The albedo of the surface (ability to adsorb light energy) or reflectivity of the surface

- Light source properties:
  - The important properties of the light source are:
    1. Intensity of the emitted light
    2. Distance to the point on the surface

Radiometry

- Energy of a photon

$$e_\lambda = \frac{hc}{\lambda} \qquad h \approx 6.63 \times 10^{-34} J \times s \quad c \approx 3 \times 10^8 m/s$$

- Radiant Energy of n photons

$$Q = \sum_{i=1}^{n} \frac{hc}{\lambda_i}$$

- Radiation flux (electromagnetic flux, radiant flux)
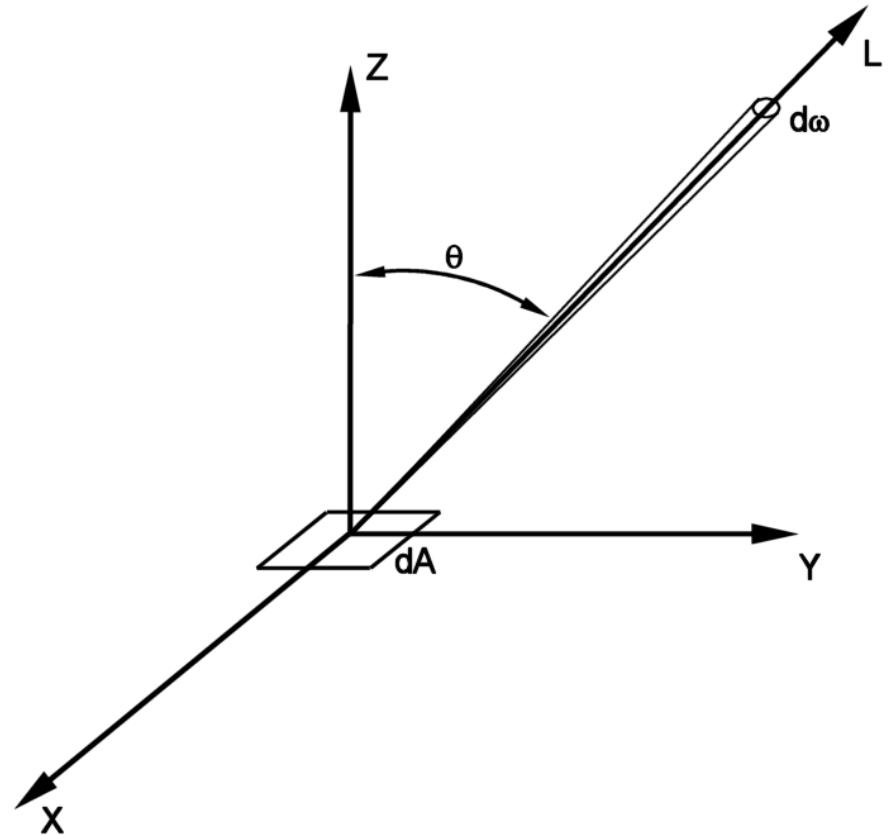 - Units: Watts

$$\Phi = \frac{dQ}{dt}$$

Radiometry

- **Radiance** – radiant flux per unit solid angle per unit projected area
    - Number of photons
    1. per time at a small area
    2. in a particular direction

$$L(w) = \frac{d^2\mathsf{F}}{\cos q \; dA \; dw}$$

$$Units : \frac{Watt}{meter^2 \; steradian}$$

Radiometry

- **Irradiance** – differential flux falling onto differential area

$$E = \frac{d\mathsf{F}}{dA} \qquad\qquad Units : \frac{Watt}{meter^2}$$

- Irradiance can be seen as a density of the incident flux falling onto a surface.

- It can be also obtained by integrating the radiance over the solid angle.

Reflection & Reflectance

- Reflection - the process by which electromagnetic flux incident on a surface leaves the surface without a change in frequency.

- Reflectance – a fraction of the incident flux that is reflected

- We do not consider:
    - absorption, transmission, fluorescence
    - diffraction

Reflectance

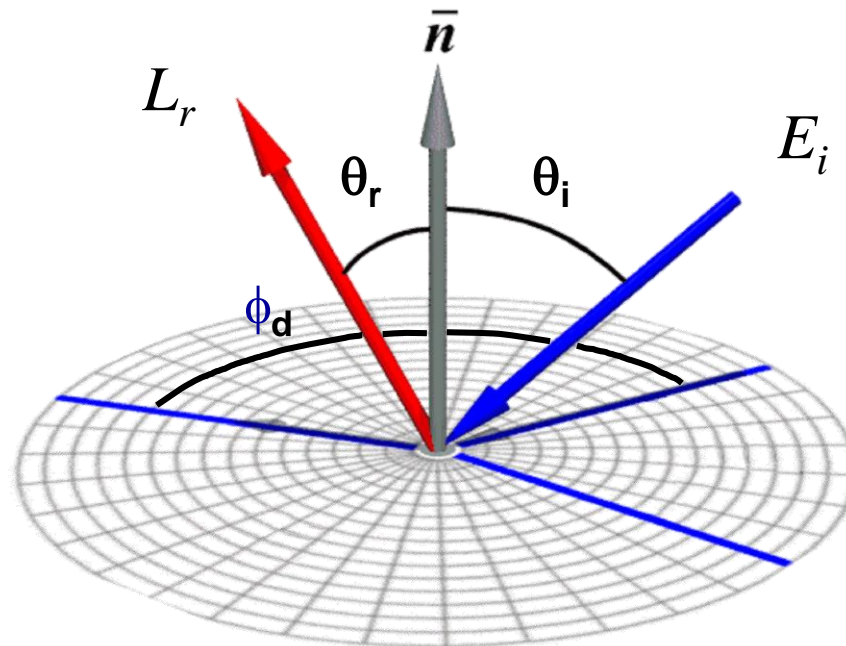- Bidirectional Reflectance Distribution Function (BRDF)

$$f_r(q_i, f_i, q_r, f_r) = \frac{dL_r(q_r, f_r)}{dE_i(q_i, f_i)}$$

$$Units: \frac{1}{steradian}$$
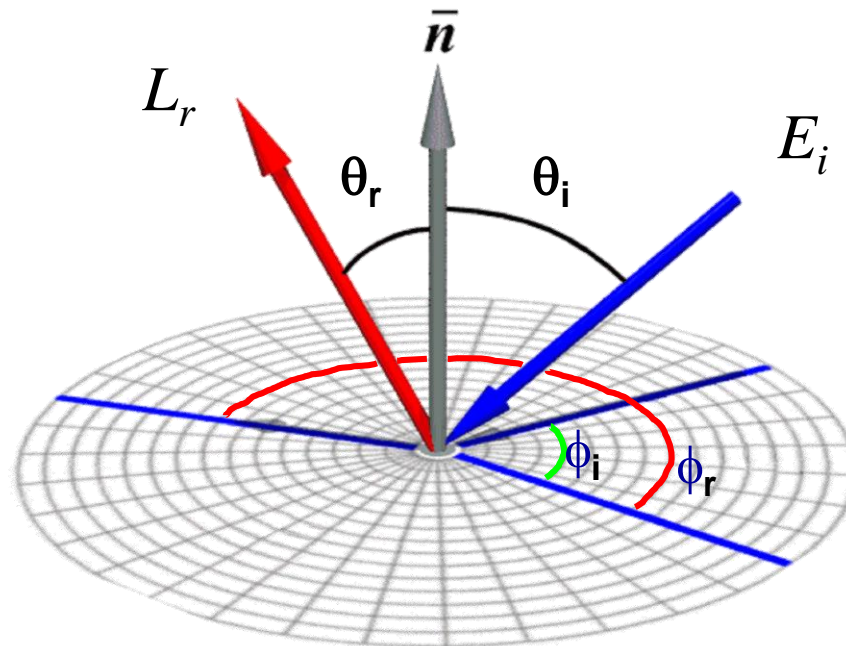
Isotropic BRDFs

- Rotation along surface normal does not change reflectance

$$f_r(q_i, q_r, f_r - f_i) = f_r(q_i, q_r, f_d) = \frac{dL_r(q_r, f_d)}{dE_i(q_i, f_d)}$$

Anisotropic BRDFs

- Surfaces with strongly oriented microgeometry elements
- Examples:
  - brushed metals,
  - hair, fur, cloth, velvet
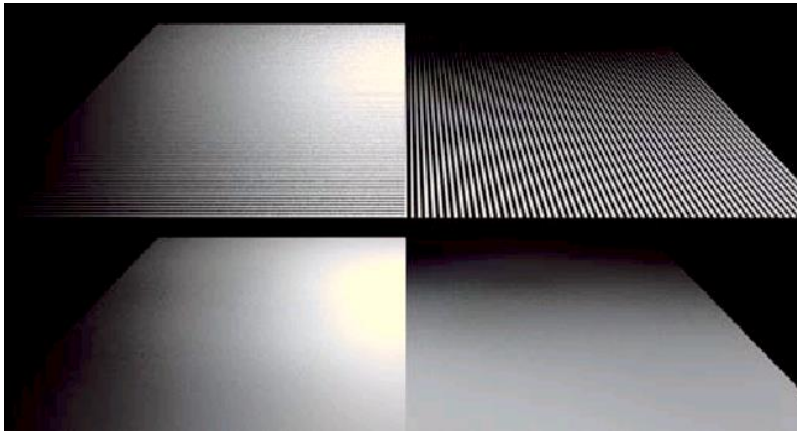
Source: Westin et al. 92

Properties of BRDFs

- Non-negativity

$$f_r(q_i, f_i, q_r, f_r) \geq 0$$

- Energy Conservation

$$\int_W f_r(q_i, f_i, q_r, f_r) dm(q_r, f_r) \leq 1 \quad \text{for all } (q_i, f_i)$$

- Reciprocity

$$f_r(q_i, f_i, q_r, f_r) = f_r(q_r, f_r, q_i, f_i)$$

Reflectance

- Bidirectional scattering-surface distribution Function (BSSRDF)

Jensen et al. SIGGRAPH 2001
A Practical Model for Subsurface Light Transport


Surface
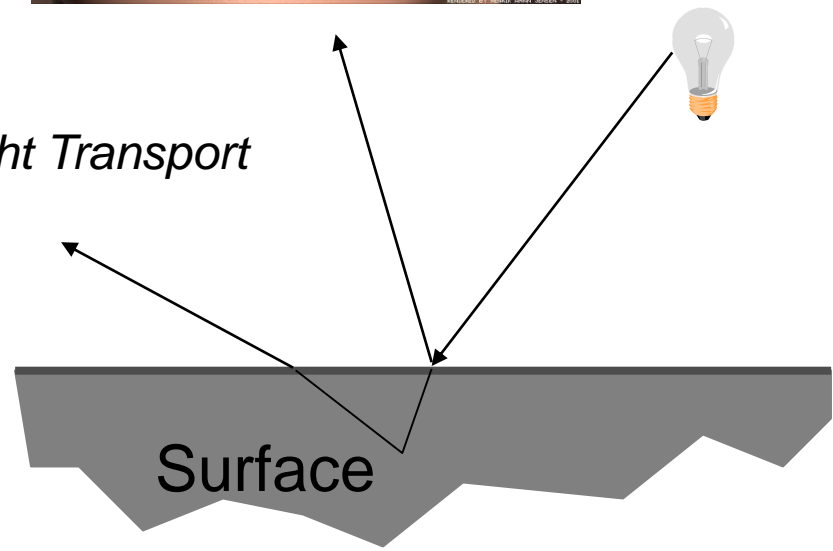
How to compute reflected radiance?

- Continuous version

$$L_r(W_r) = \int_W f_r(W_i, W_r) dE_i(W_i) =$$

$$= \int_W f_r(W_i, W_r)\, dL_i(W_i)\cos(W_i \cdot n)\, dW_i \qquad W = (q, f)$$

- Discrete version – n point light sources

$$L_r(W_r) = \sum_{j=1}^{n} f_r(W_{ij}, W_r) E_j =$$

$$= \sum_{j=1}^{n} f_r(W_{ij}, W_r) \cos q_j \frac{F_{sj}}{4p\, d_j^{\,2}}$$

Ideal Diffuse Reflectance

- Assume surface reflects equally in all directions.
- An ideal diffuse surface is, at the microscopic level, a very rough surface.
  - Example: chalk, clay, some paints

Surface
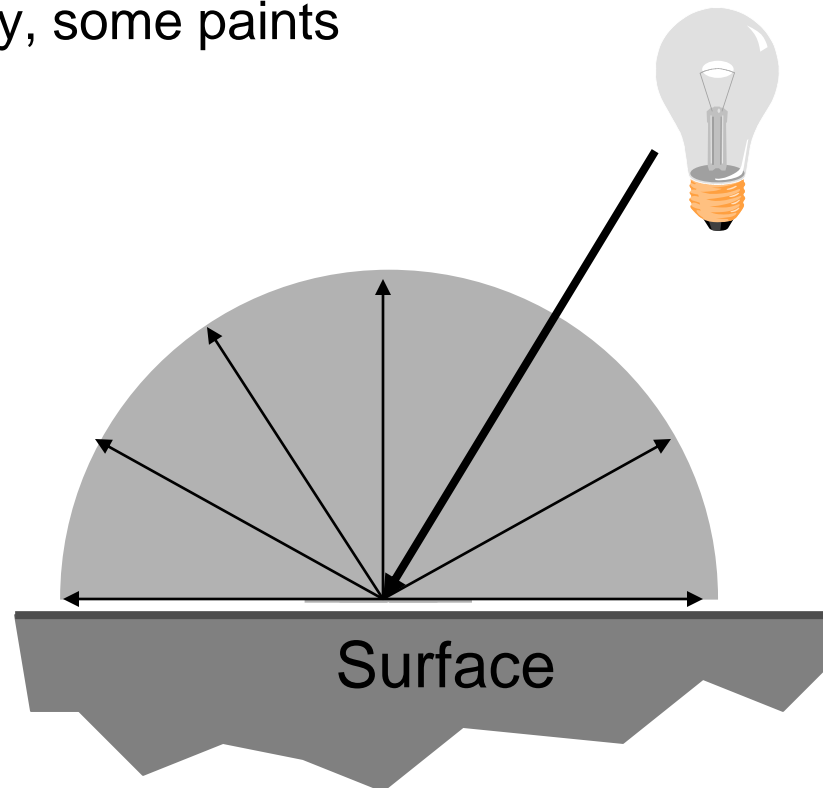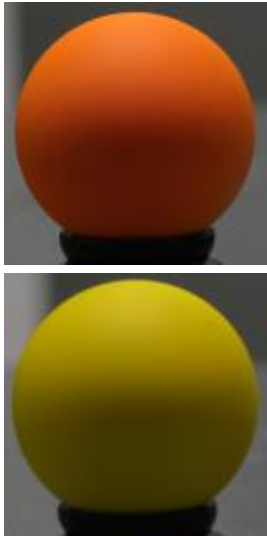
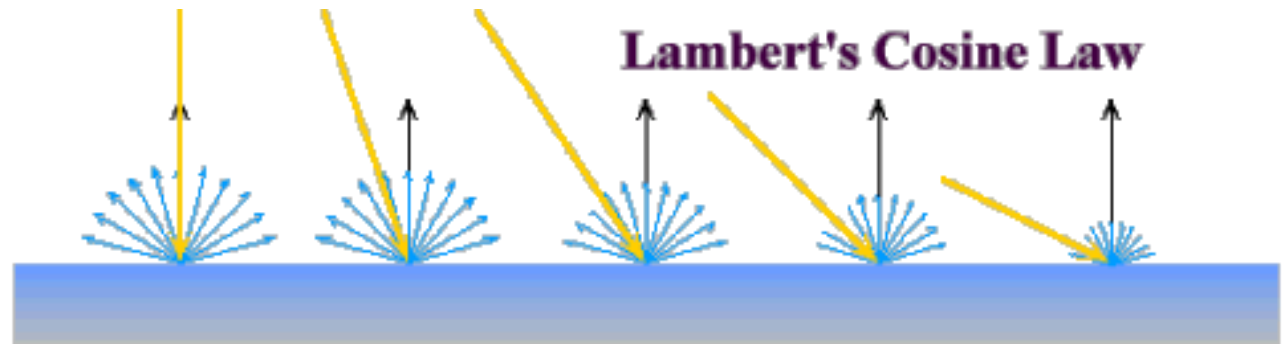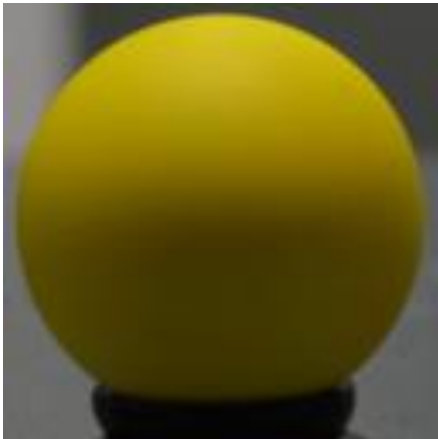# *Ideal Diffuse Reflectance*

- BRDF value is constant

$$L_r(\omega_r) = \int_\Omega f_r(\omega_i, \omega_r) dE_i(\omega_i) =$$

$$= f_r \int_W dE_i(W_i) =$$

$$= f_r E_i$$

$$dB = dA \cos q_i$$
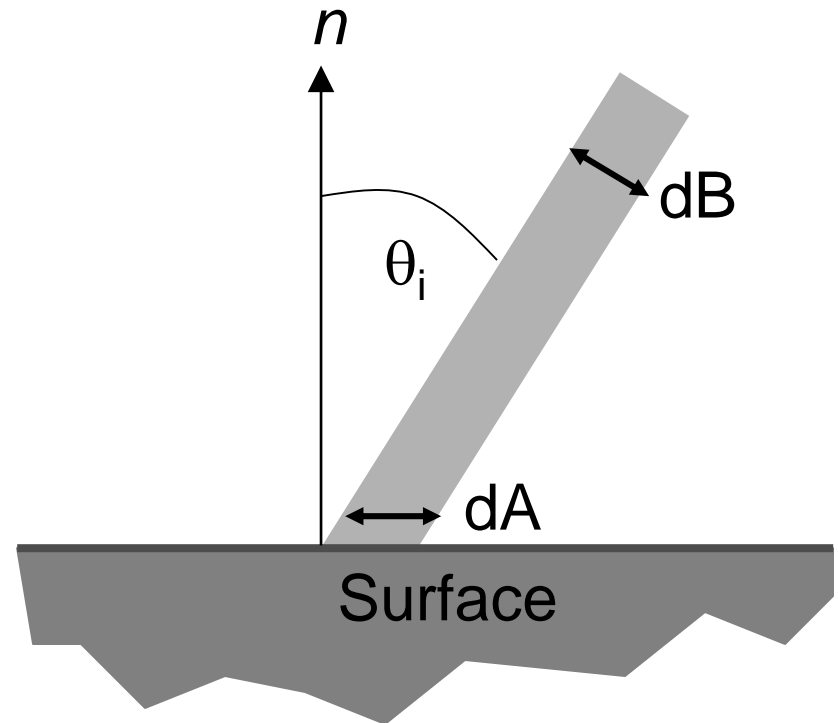
Ideal Diffuse Reflectance

- Ideal diffuse reflectors reflect light according to <u>Lambert's cosine law</u>.

Lambert's Cosine Law

Ideal Diffuse Reflectance

- Single Point Light Source
  - k_d: The diffuse reflection coefficient.
  - n: Surface normal.
  - l: Light direction.

$$L(\omega_r) = k_d (\mathbf{n} \cdot \mathbf{l}) \frac{\Phi_s}{4\pi \, d^2}$$

n

θ

l

Surface
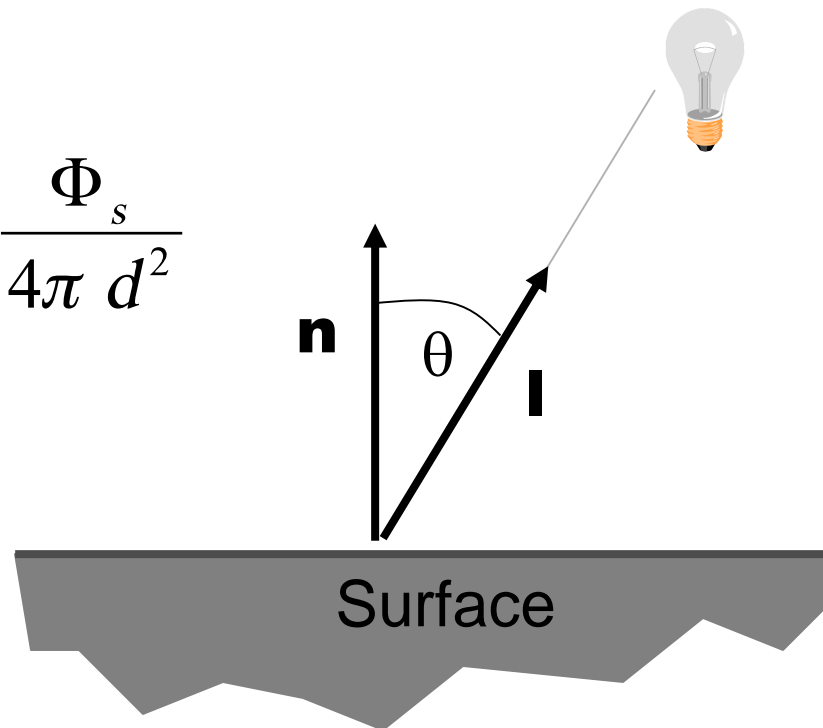
Ideal Diffuse Reflectance – More Details

- If $\mathbf{n}$ and $\mathbf{l}$ are facing away from each other, $\mathbf{n} \bullet \mathbf{l}$ becomes negative.

- Using $\max((\mathbf{n} \bullet \mathbf{l}), 0)$ makes sure that the result is zero.
  - From now on, we mean $\max()$ when we write •.

Do not forget to normalize your vectors
for the dot product!

Ideal Specular Reflectance

- Reflection is only at mirror angle.
  - View dependent
  - Microscopic surface elements are usually oriented in the same direction as the surface itself.
  - Examples: mirrors, highly polished metals.
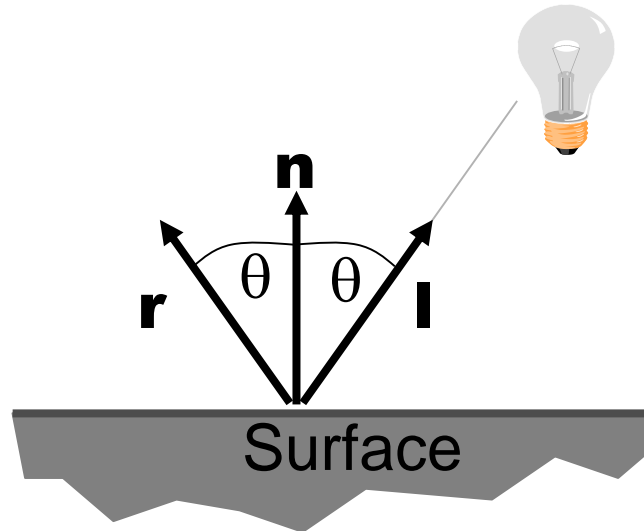
Ideal Specular Reflectance

- Special case of Snell's Law
  - The incoming ray, the surface normal, and the reflected ray all lie in a common plane.
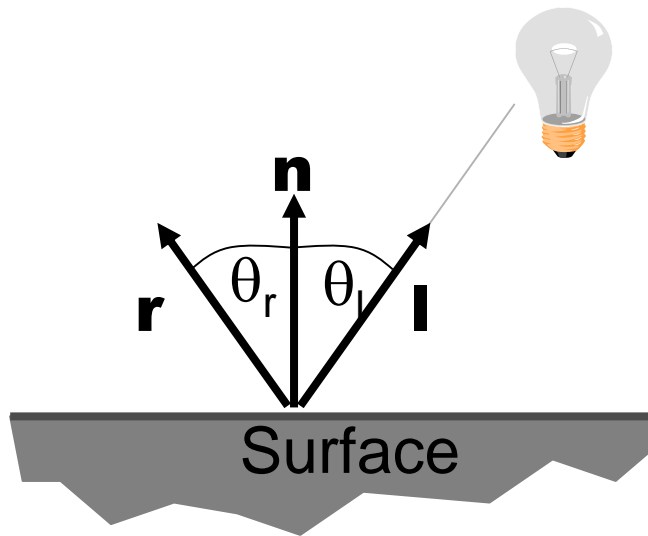
$$n_l \sin q_l = n_r \sin q_r$$

$$n_l = n_r$$

$$q_l = q_r$$

Non-ideal Reflectors

- Snell's law applies only to ideal mirror reflectors.
- Real materials tend to deviate significantly from ideal mirror reflectors.
- They are not ideal diffuse surfaces either …
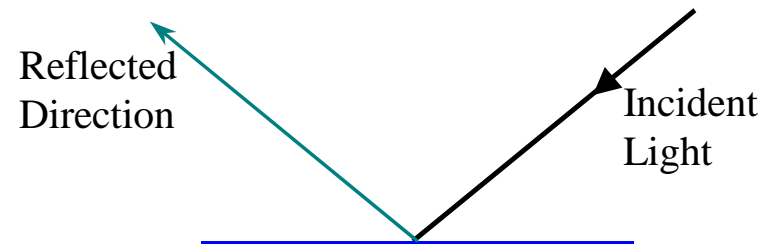
Non-ideal Reflectors

- Simple Empirical Model:
  - We expect most of the reflected light to travel in the direction of the ideal ray.
  - However, because of microscopic surface variations we might expect some of the light to be reflected just slightly offset from the ideal reflected ray.
  - As we move farther and farther, in the angular sense, from the reflected ray we expect to see less light reflected.
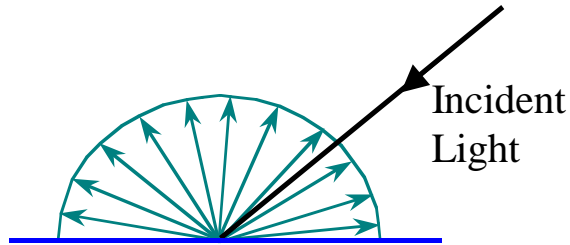
Non-ideal Reflectors: Surface Characteristics
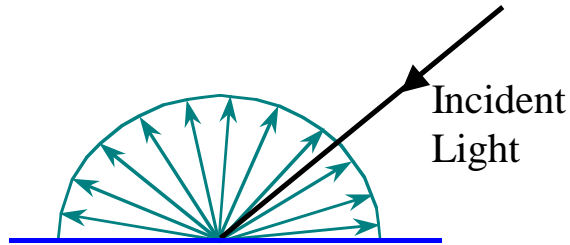
Incident
Light

Perfectly Matt surface
The reflected intensity is the same in all directions

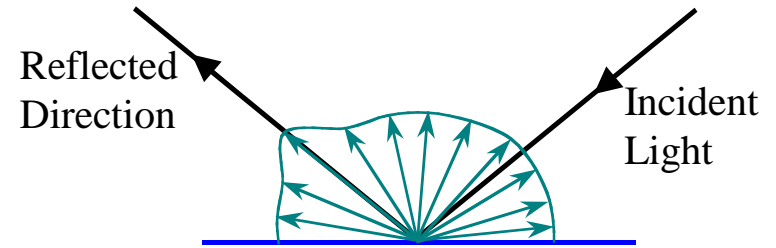Reflected
Direction

Incident
Light

Perfect Mirror
All light re-admitted in the reflected direction
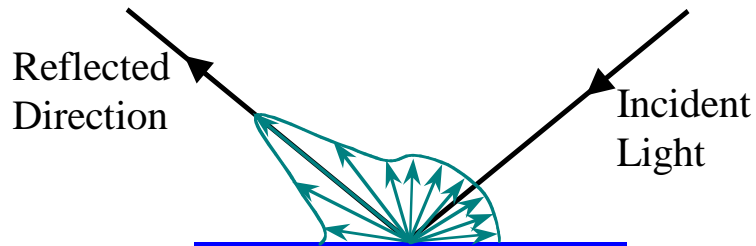
Non-ideal Reflectors: Surface Characteristics

Incident Light

Perfectly Matt surface
The reflected intensity is the same in all directions

Reflected Direction

Incident Light

Slightly specular (shiny) surface
Slightly higher intensity in the reflected direction

Reflected Direction

Incident Light

Highly specular (shiny) surface
High intensity in the reflected direction

Reflected Direction

Incident Light

Perfect Mirror
All light re-admitted in the reflected direction

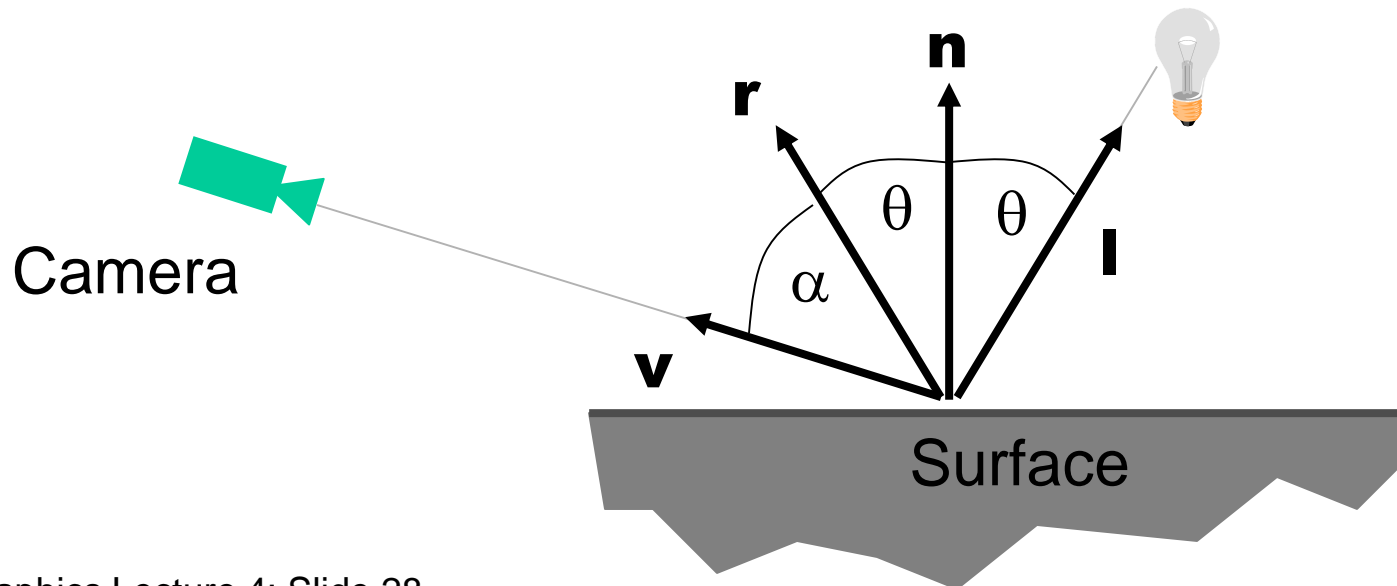Graphics Lecture 4: Slide 26
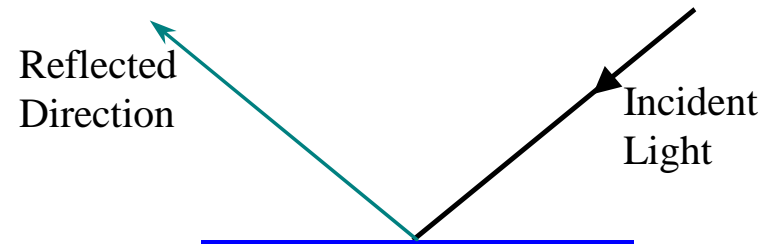
The Phong Model

- How much light is reflected?
  - Depends on the angle between the ideal reflection direction and the viewer direction α.
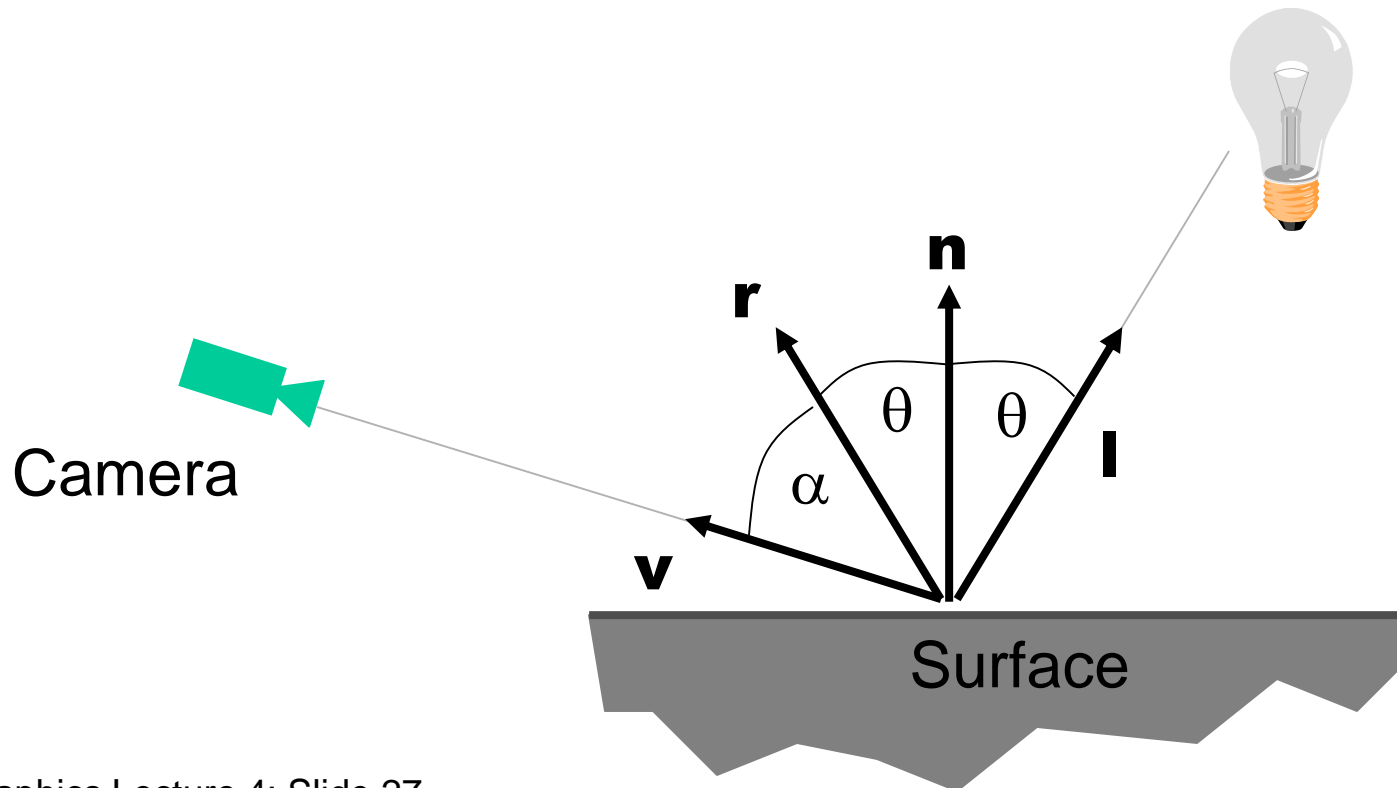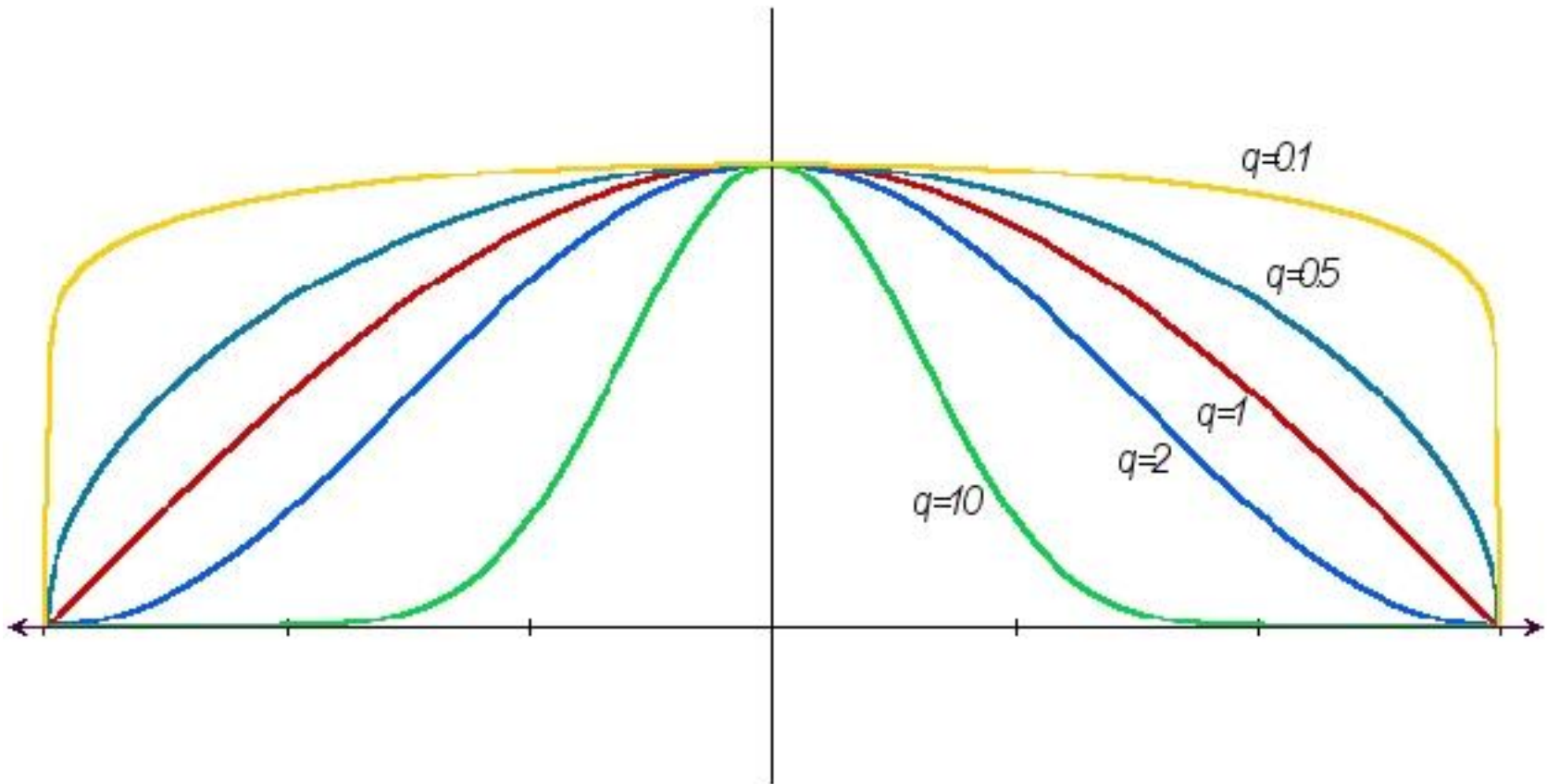
The Phong Model

- Parameters
  - k_s: specular reflection coefficient
  - q : specular reflection exponent

$$L(\omega_r) = k_s (\cos\alpha)^q \frac{\Phi_s}{4\pi\, d^2} = k_s (\mathbf{v}\cdot\mathbf{r})^q \frac{\Phi_s}{4\pi\, d^2}$$

Camera

n

r

θ θ

l

α

v

Surface

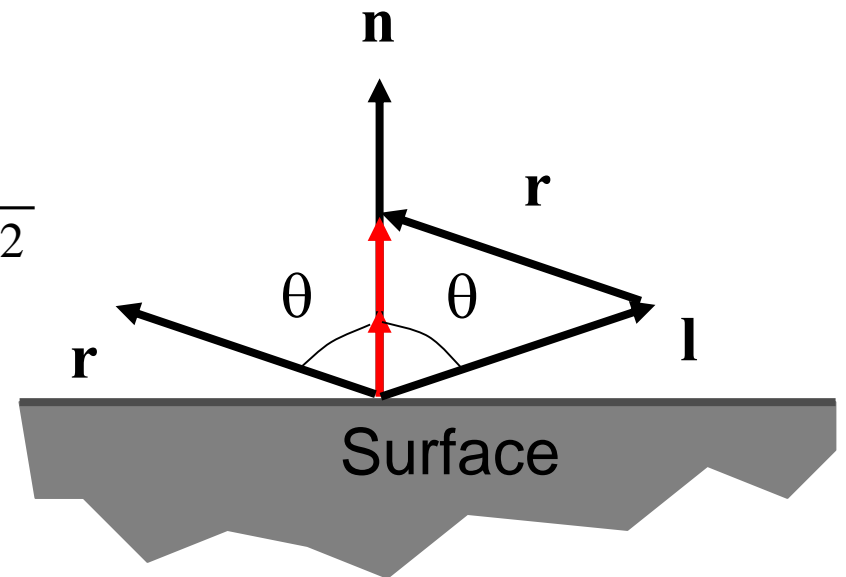# *The Phong Model*

- Effect of the q coefficient

The Phong Model

$$\mathbf{r} + \mathbf{l} = 2 \quad \cos q \, \mathbf{n}$$

$$\mathbf{r} = 2(\mathbf{n} \times \mathbf{l})\mathbf{n} - \mathbf{l}$$

$$L(W_r) = k_s (\mathbf{v} \times \mathbf{r})^q \, \frac{\mathsf{F}_s}{4 \rho \, d^2} =$$

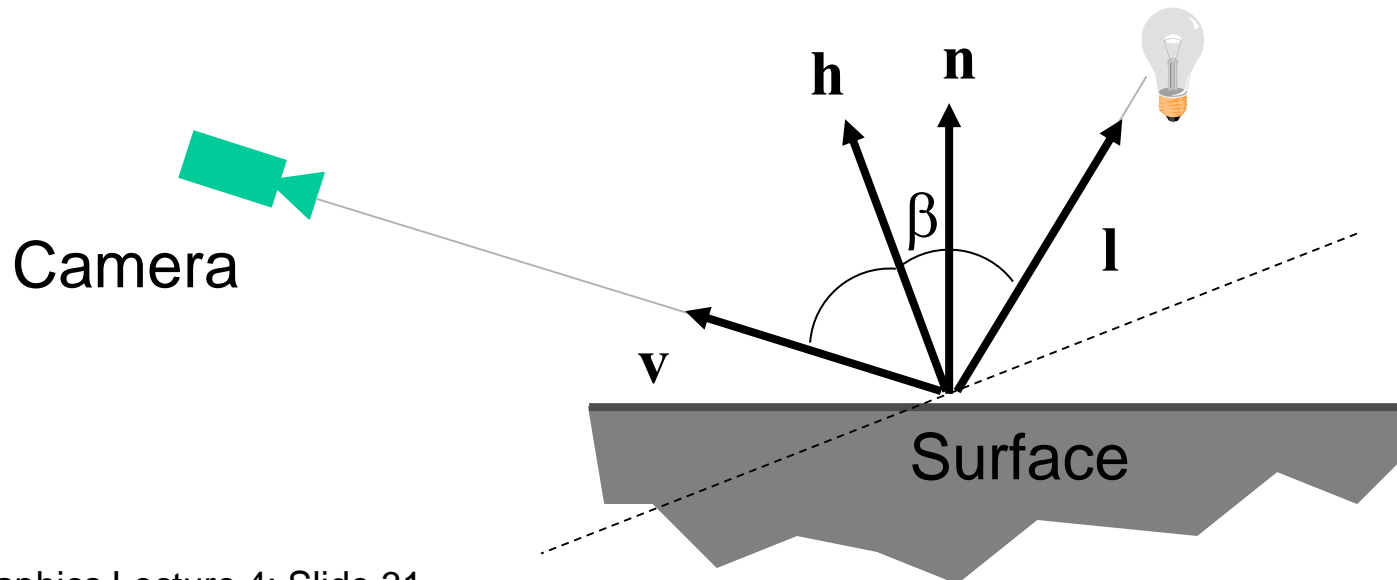$$= k_s (\mathbf{v} \times (2(\mathbf{n} \times \mathbf{l})\mathbf{n} - \mathbf{l}))^q \, \frac{\mathsf{F}_s}{4 \rho \, d^2}$$

Blinn-Phong Variation
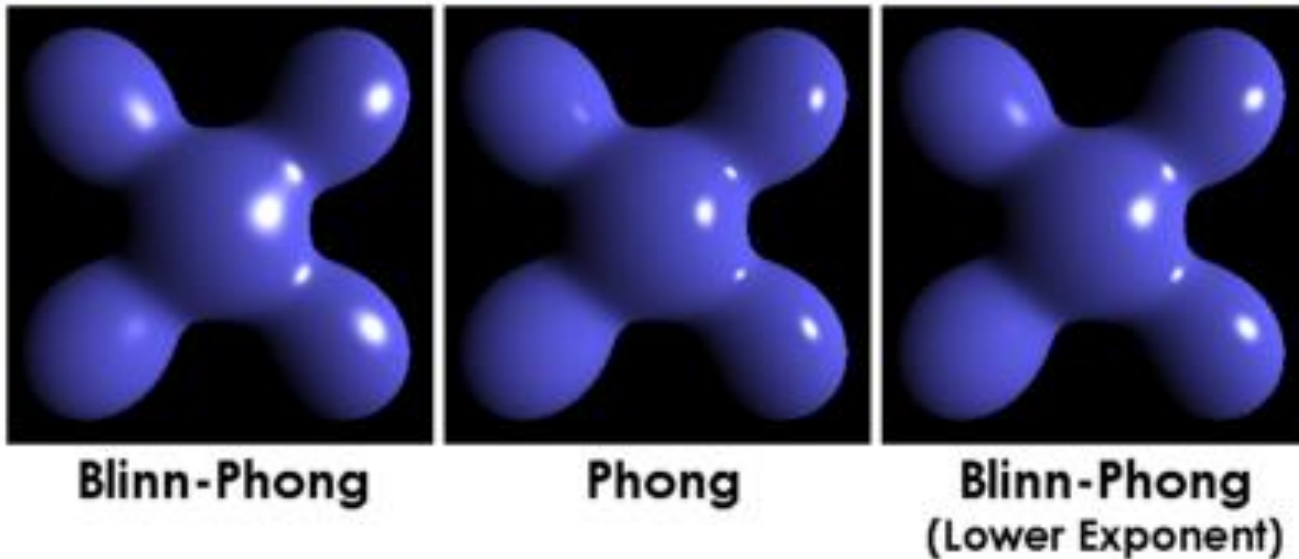
- Uses the halfway vector **h** between **l** and **v**.

$$\mathbf{h} = \frac{\mathbf{l} + \mathbf{v}}{\|\mathbf{l} + \mathbf{v}\|}$$

$$L(W_r) = k_s (\cos b)^q \frac{F_s}{4 p d^2} = k_s (\mathbf{n} \times \mathbf{h})^q \frac{F_s}{4 p d^2}$$

h **n**

β

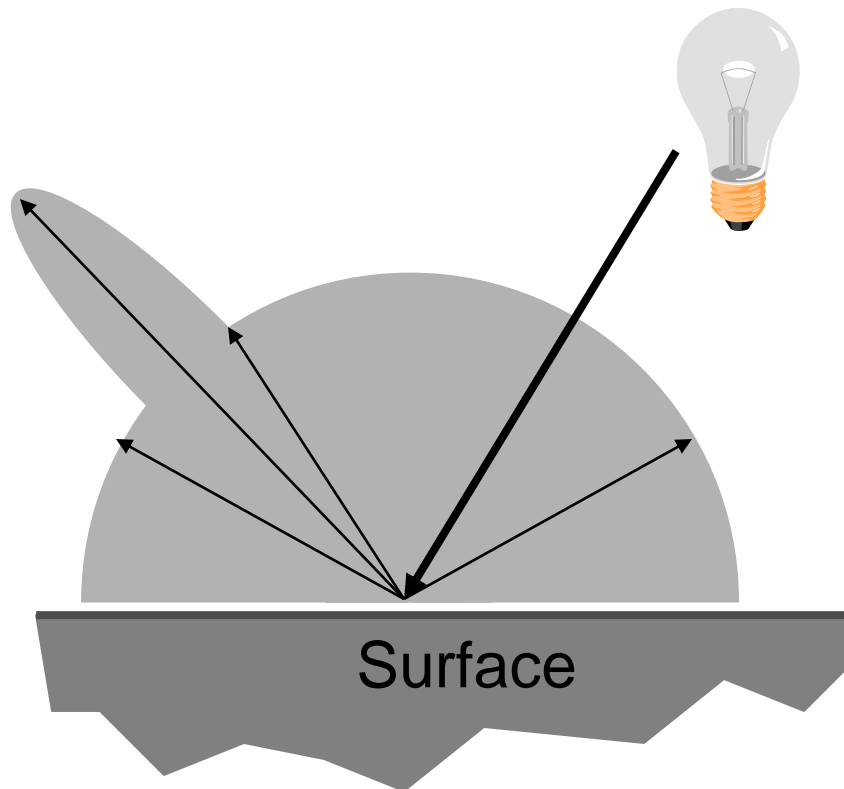Camera

l

v

Surface

Phong vs Blinn-Phong

- The following spheres illustrate specular reflections as the direction of the light source and the coefficient of shininess is varied.
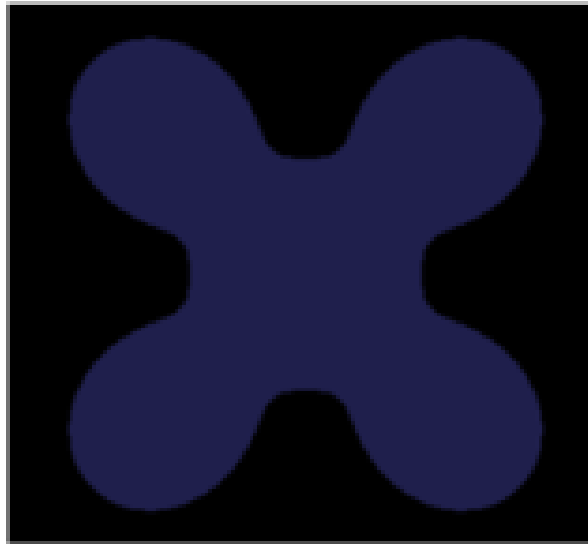

Blinn-Phong   Phong   Blinn-Phong (Lower Exponent)

The Phong Model

- Sum of three components:
    - diffuse reflection + specular reflection + ambient.

Surface

Ambient Illumination

- Represents the reflection of all indirect illumination.
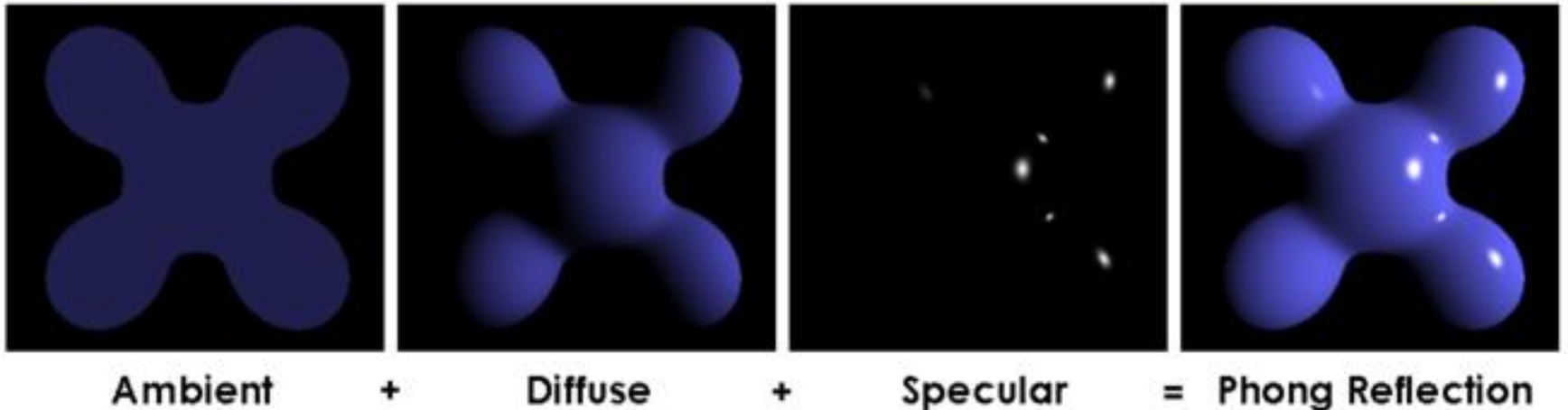- This is a hack!
- Avoids the complexity of global illumination.

$$L(W_r) = k_a$$

Putting it all together
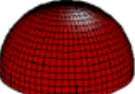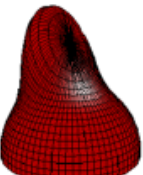
- Phong Illumination Model

$$L(W_r) = k_a + \left(k_d(\mathbf{n} \times \mathbf{l}) + k_s(\mathbf{v} \times \mathbf{r})^q \right) \frac{\mathsf{F}_s}{4\rho d^2}$$

Ambient   +   Diffuse   +   Specular   =   Phong Reflection

Putting it all together

- Phong Illumination Model

$$L(W_r) = k_a + \left(k_d (\mathbf{n} \times \mathbf{l}) + k_s (\mathbf{v} \times \mathbf{r})^q \right) \frac{F_s}{4\rho d^2}$$

Inverse Square Law

- It is well known that light falls off according to an inverse square law. Thus, if we have light sources close to our polygons we should model this effect.
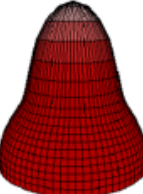
$$L(W_r) = k_a + \left(k_d(\mathbf{n} \times \mathbf{l}) + k_s(\mathbf{v} \times \mathbf{r})^q \right) \frac{F_s}{4 p \, d^2}$$

where d is the distance from the light source to the object

Heuristic Law

- Although physically correct the inverse square law does not produce the best results.

- Instead the following is often used:

$$L(W_r) = k_a + \Big(k_d\,(\mathbf{n}\times\mathbf{l}) + k_s(\mathbf{v}\times\mathbf{r})^q \Big)\frac{\mathsf{F}_s}{4p\,(d+s)}$$

  where s is an heuristic constant.

- One might be tempted to think that light intensity falls off with the distance to the viewpoint, but it doesn't!

- Why not?

Questions?

Using Shading

- There are three levels at which shading can be applied in polygon based systems:

  Flat Shading

  Gouraud Shading

  Phong Shading

  Interpolation Shading

- They provide increasing realism at higher computational cost

Using Shading

- Flat Shading:
  - Each polygon is shaded uniformly over its surface.
  - The shade is computed by taking a point in the centre and the surface normal vector. (Equivalent to a light source at infinity)
  - Usually only diffuse and ambient components are used.

- Interpolation Shading:
  - A more accurate way to render a shaded polygon is to compute an independent shade value at each point.
  - This is done quickly by interpolation:
  - 1. Compute a shade value at each vertex
  - 2. Interpolate to find the shade value at the boundary
  - 3. Interpolate to find the shade values in the middle

Calculating the shades at the edges

$$L = (d_1 L_3 + d_2 L_1) / (d_1 + d_2)$$
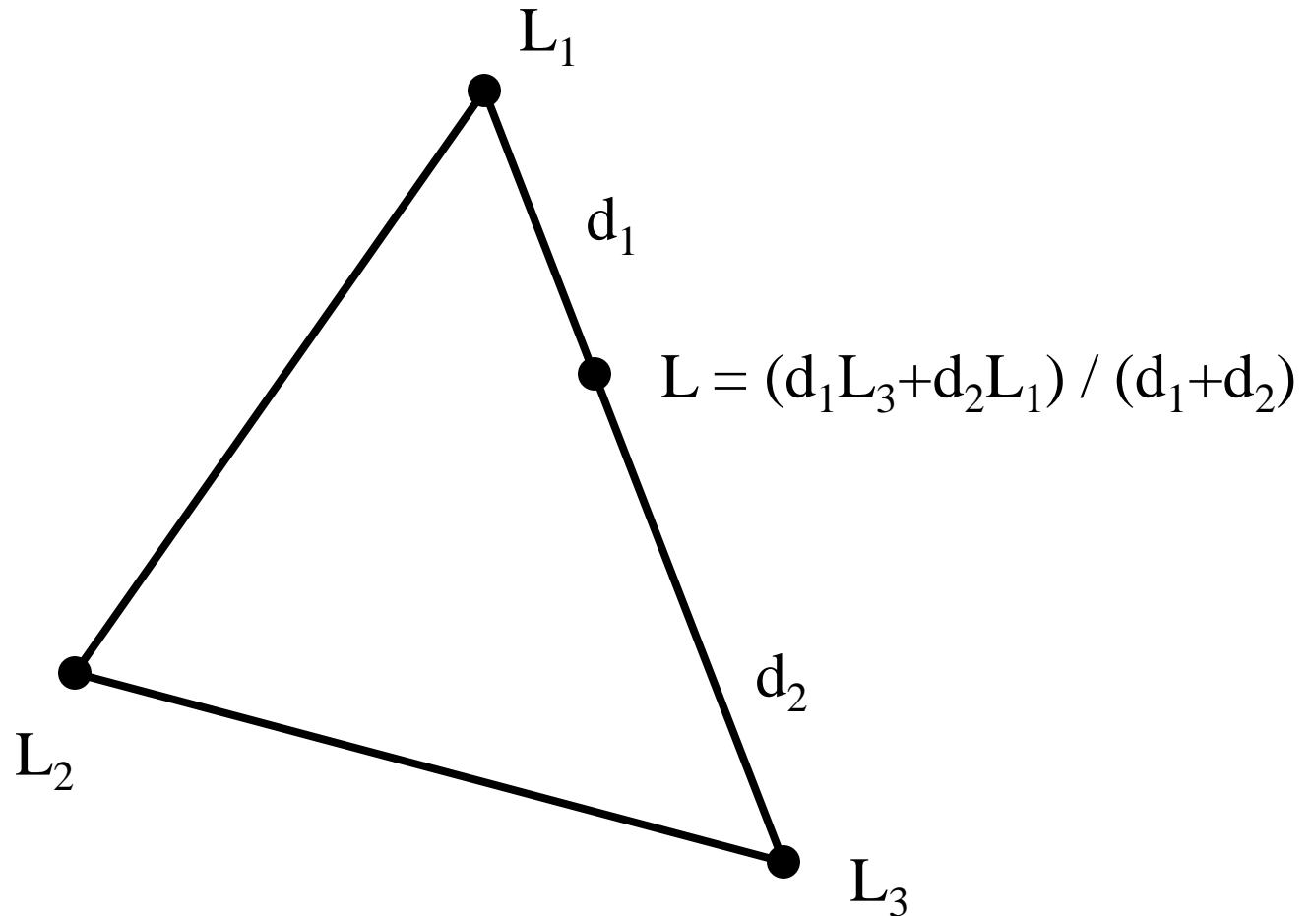
Calculating the internal shades

$$L = (d_1 L_B + d_2 L_A) / (d_1 + d_2)$$

L_1

L_A

L_B

d_1

d_2

L_2

L_3
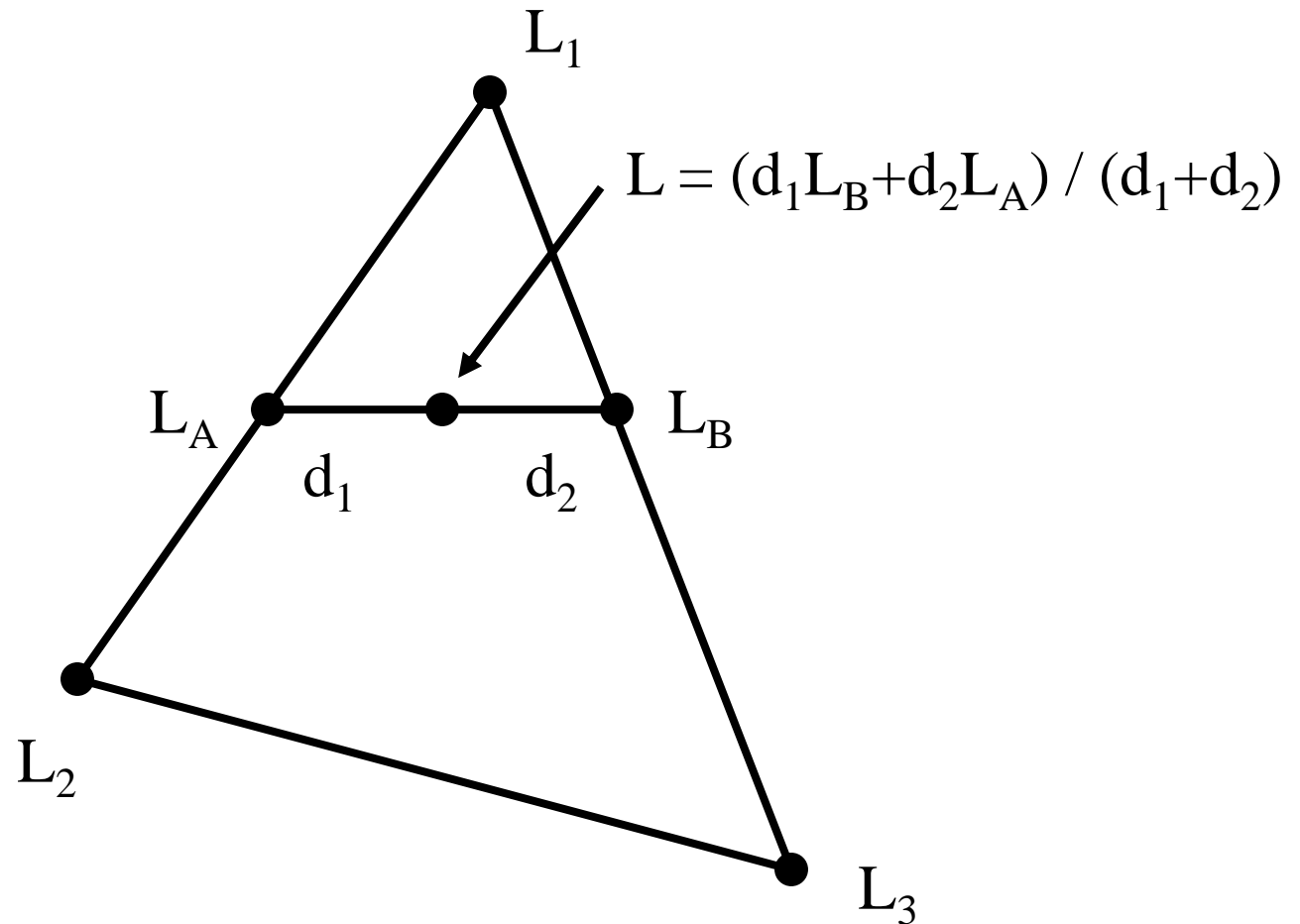
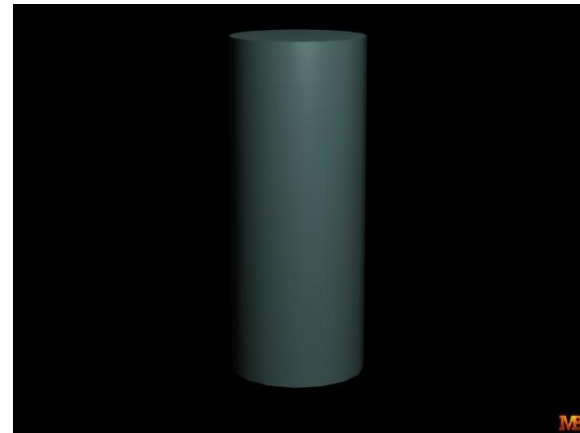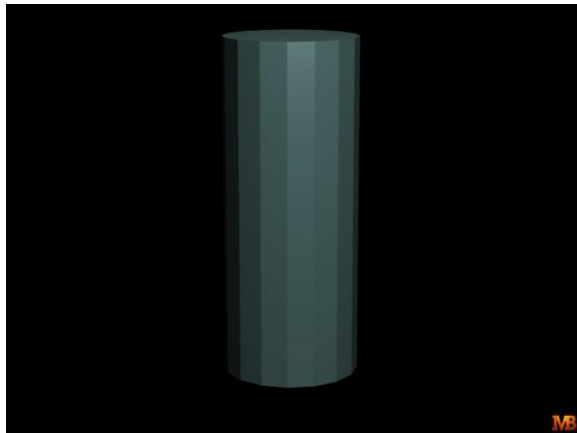# *Interpolating over polygons*

- In addition to interpolating shades over polygons, we can interpolate them over groups of polygons to create the impression of a smooth surface.

- The idea is to create at each vertex an averaged intensity from all the polygons that meet at that vertex.

Computing an average normal vector at a vertex

$$\mathbf{n_{ave}} = (\mathbf{\textit{n}_1} + \mathbf{\textit{n}_2} + \mathbf{\textit{n}_3} + \mathbf{\textit{n}_4})/4$$

$\textit{n}_1$

$\textit{n}_4$

$\mathbf{n_{ave}}$

$\textit{n}_2$

Gouraud Shading
(Object Space)

$\textit{n}_3$
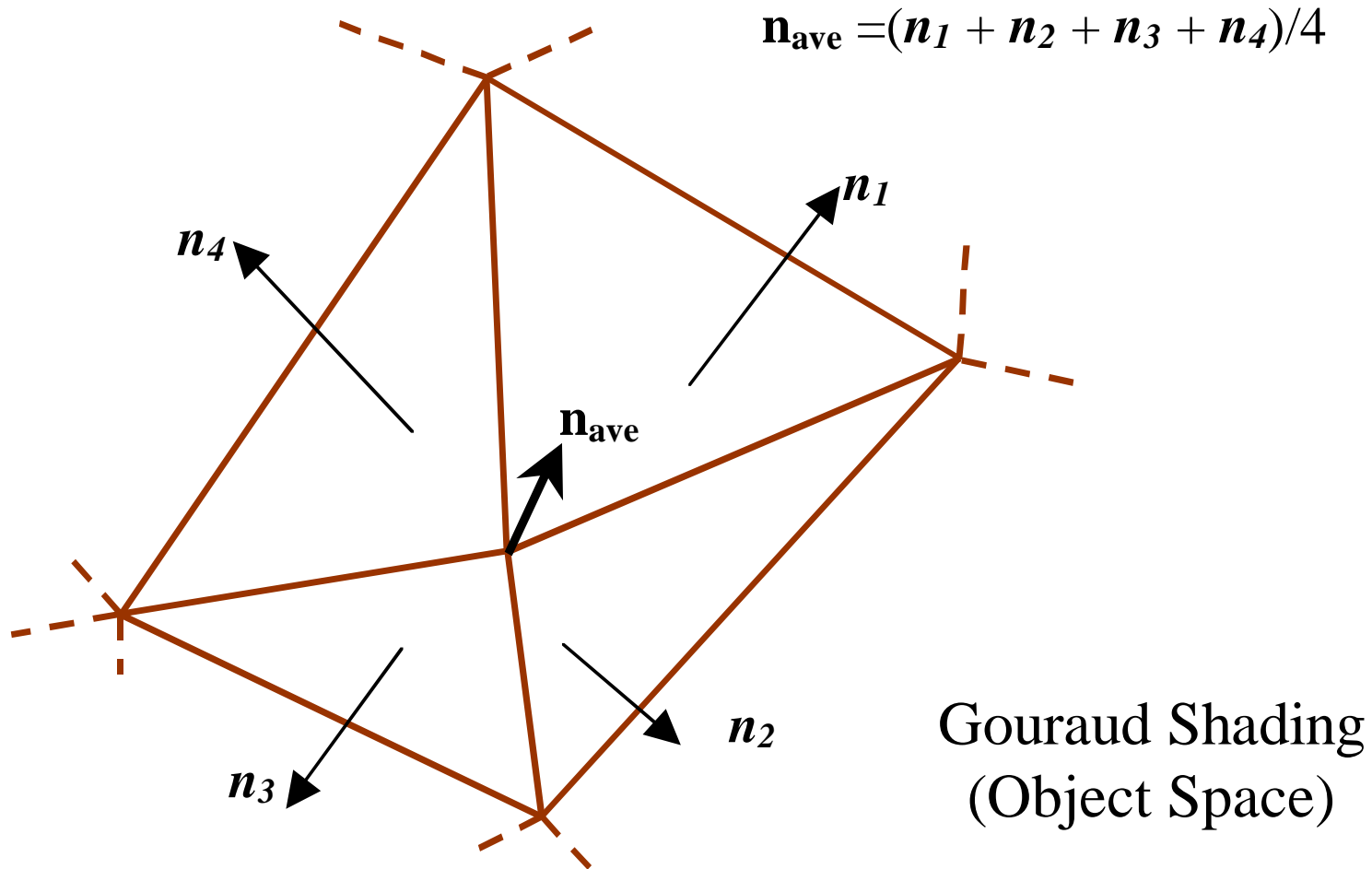
Smooth Shading

- Need to have per-vertex normals

- Gouraud Shading

  - Interpolate color across triangles

  - Fast, supported by most of the graphics accelerator cards

  - Can't model specular components accurately, since we do not have the normal vector at each point on a polygon.

- Phong Shading

  - Interpolate normals across triangles

  - More accurate modelling of specular compoents, but slower.

Interpolation of the 3D normals

- We may express any point for this facet in parametric form:

$$\mathbf{P} = \mathbf{V_1} + \mu_1(\mathbf{V_2} - \mathbf{V_1}) + \mu_2(\mathbf{V_3} - \mathbf{V_1})$$

- The average normal vector at the same point may be calculated as the vector $\mathbf{a}$:

$$\mathbf{a} = \mathbf{n_1} + \mu_1(\mathbf{n_2} - \mathbf{n_1}) + \mu_2(\mathbf{n_3} - \mathbf{n_1})$$

- and then

$$\mathbf{n_{average}} = \mathbf{a} \, / \, | \, \mathbf{a} \, |$$

2D or 3D

- The interpolation calculations may be done in either 2D or 3D

- For specular reflections the calculation of the reflected vector and viewpoint vector must be done in 3D.