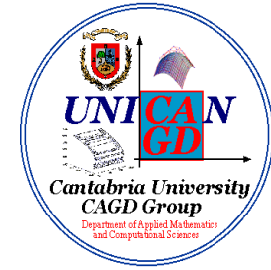




*Department of Applied Mathematics  
and Computational Sciences*  
*University of Cantabria*  
**UC-CAGD Group**



**COMPUTER-AIDED GEOMETRIC DESIGN  
AND COMPUTER GRAPHICS:  
ILLUMINATION MODELS**

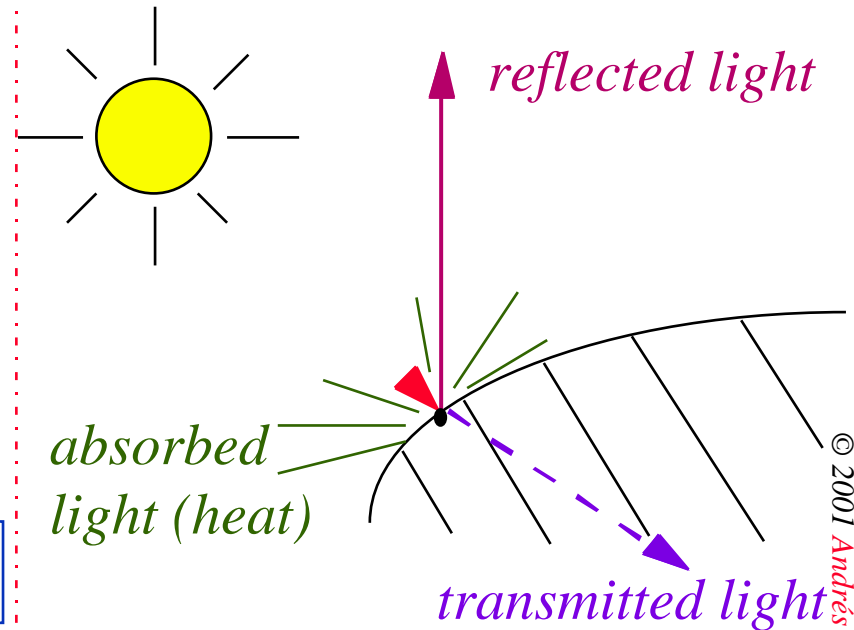
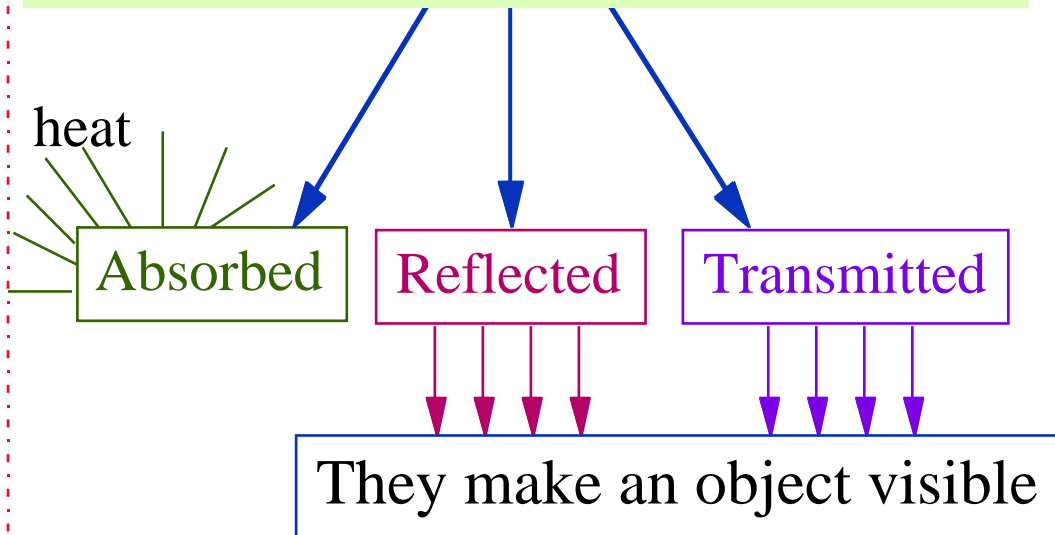
**Andrés Iglesias**

**e-mail: [iglesias@unican.es](mailto:iglesias@unican.es)**

**Web pages: <http://personales.unican.es/iglesias>  
<http://etsiso2.macc.unican.es/~cagd>**

# Illumination Models

Light energy falling on a surface can be:



The amount of energy absorbed, transmitted or deflected depends on the *wavelength* (w.l.) of the light.

If

then, the object

- all the incident light energy is absorbed.....is invisible
- nearly all the incident light energy is absorbed..... appears black
- only a small fraction is absorbed .....appears white
- the incident light energy is nearly equally reduced for all w.l. ....appears gray
- the incident light energy is selectively reduced for all w.l. ....appears colored

# Illumination Models

Rogers, D.F. *Procedural Elements for Computer Graphics*, McGraw-Hill, 2nd. Edition, 1998

The character of the light reflected or transmitted...

depends on:

- Composition of the light source
- Direction of the light source
- Geometry of the light source
- The surface orientation
- The surface properties of the object

and can be:

**Diffuse:** light that has penetrated below the surface of the object, been absorbed and then reemitted

Scattered equally in all directions

Observer's position is unimportant

**Especlar:** light does not penetrate below the surface

It is not scattered

Character of light reamins unchanged

Light is independent of surface's color

# *Illumination Models*

Each point may have several sources of illumination:

## **direct illumination**

light arrives straight from the light sources

## **indirect illumination**

light arrives after interacting with the rest of the scene



According to how they handle these sources, algorithms can be grouped into:

## **global illumination algorithms**

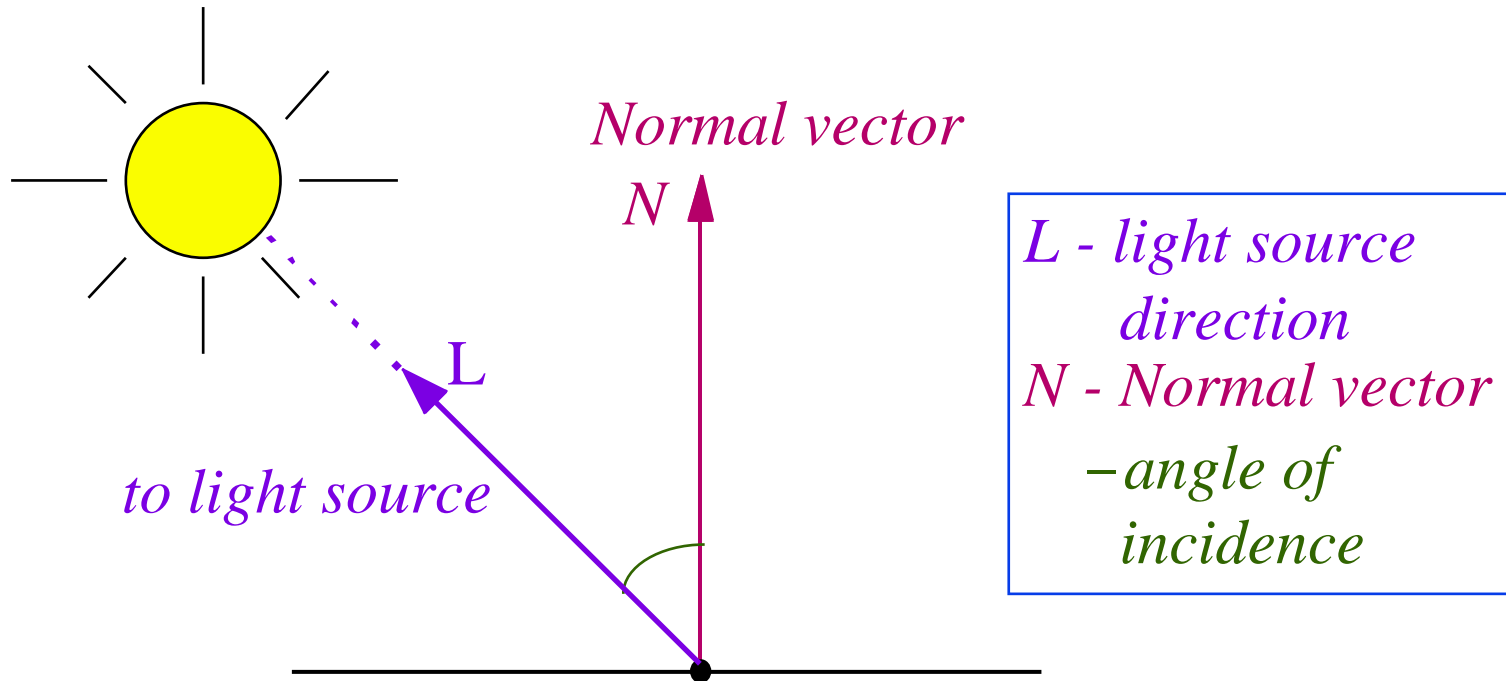
Both kinds of sources are considered

## **local illumination algorithms**

Only direct lights are taken into account

# *Illumination Models*

Some useful definitions:



- **N** is the **surface normal**
- **L** is the **direction to light source**
- Vectors **N** and **L** are *unit* vectors
- is the **angle of incidence**

# Illumination Models

## Illumination model 1: Ambient light

### Ambient light

- Uniform from all directions
- $K$  measures reflectivity of surface for diffuse light (values in the range: 0-1)

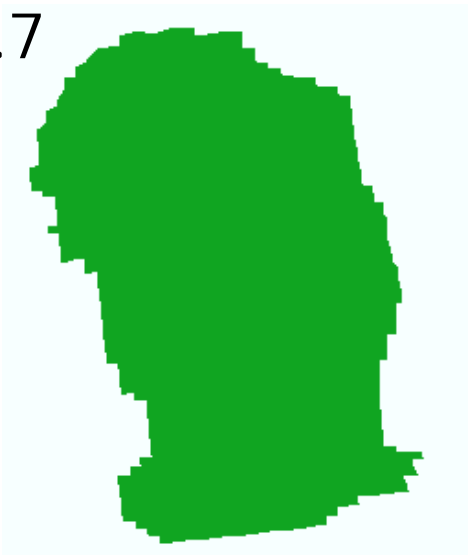
$$I = K I$$

Intensity of ambient light

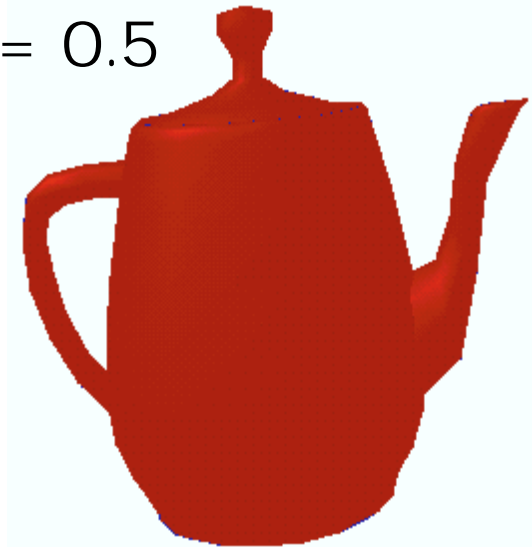
Ambient reflection coefficient

**Problem:** an object is illuminated uniformly

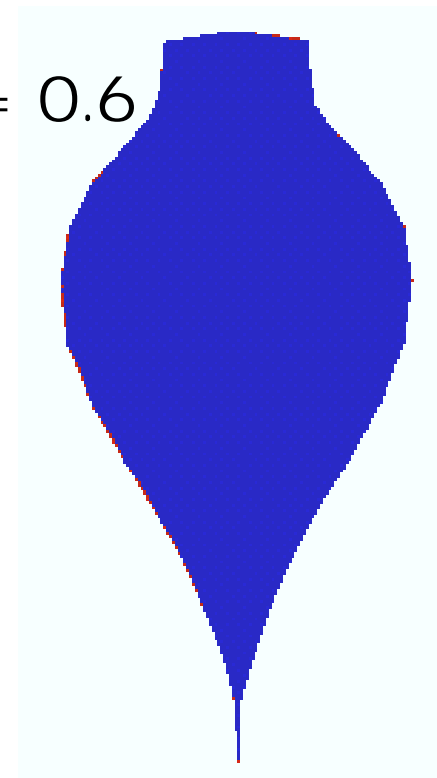
$K = 0.7$



$K = 0.5$



$K = 0.6$



# Illumination Models

## Illumination model 2: *Ambient* + *diffuse light*

### Lambert's Cosine Law

incident intensity from a point light source

$$I_d(\lambda) = I_l(\lambda) K_d(\lambda) \cos(\theta)$$

↑ wavelenght      ↑ diffuse reflection function (0 ≤ K<sub>d</sub>(λ) ≤ 1)

↓  
intensity of reflected diffuse light

Therefore, the Lambertian illumination model becomes:

$$I(\lambda) = \underbrace{I_l(\lambda) K_d(\lambda) \cos(\theta)}_{\text{diffuse light}} + \underbrace{K(\lambda) I_a(\lambda)}_{\text{ambient light}}$$

# Illumination Models

## Illumination model 2: Ambient + diffuse light

In practice, dependence on the wavelength is usually omitted:

$$I = I_l \underbrace{K_d \cos(\theta)}_{\text{diffuse light}} + \underbrace{K}_0 \underbrace{I}_{\frac{1}{2}}$$

$K + K_d < 1$

Since **N** and **L** are unit vectors, it holds that:  $\cos(\theta) = \mathbf{N} \cdot \mathbf{L}$

↑  
*dot product*

$$I = I_l \underbrace{K_d (\mathbf{N} \cdot \mathbf{L})}_{\text{diffuse light}} + \underbrace{K}_0 \underbrace{I}_{\frac{1}{2}}$$

$K + K_d < 1$



# Illumination Models

## Illumination model 2: *Ambient* + *diffuse* light

Surfaces with a simple Lambertian diffuse reflection appear to have a *dull matte* surface:

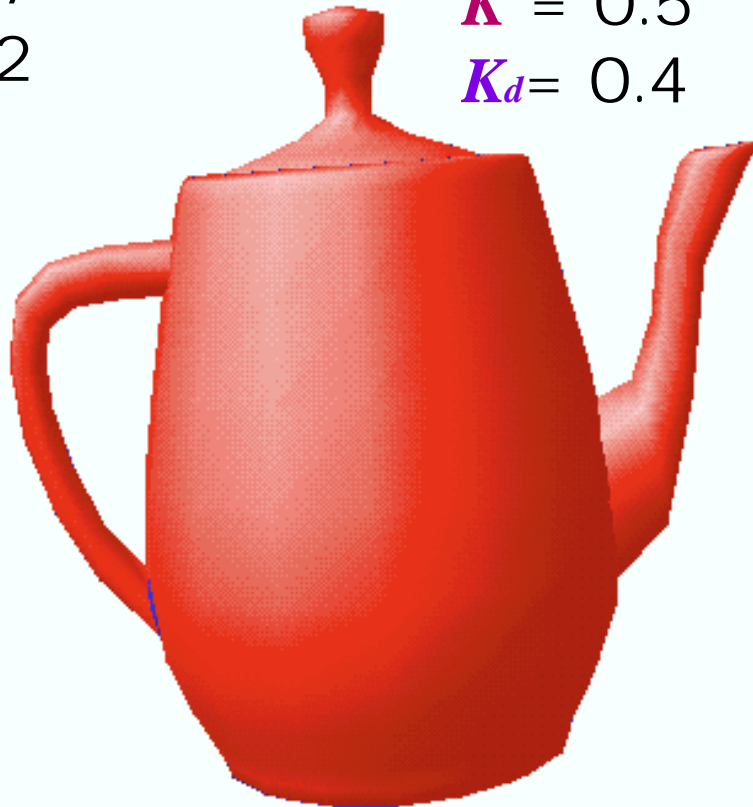
$$K = 0.7$$

$$K_d = 0.2$$



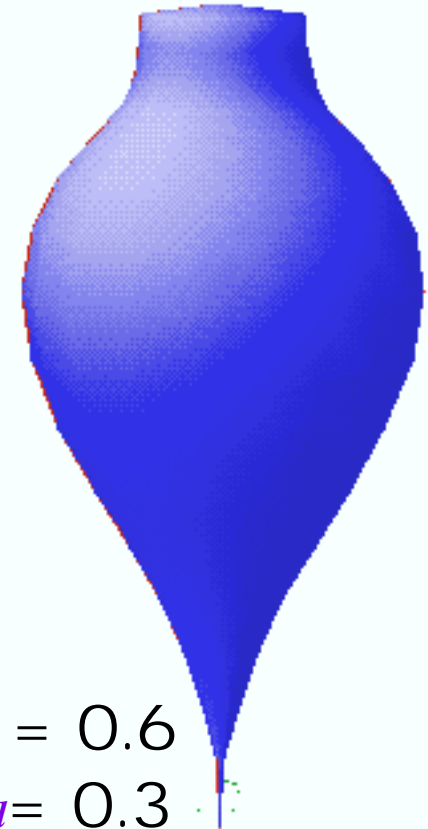
$$K = 0.5$$

$$K_d = 0.4$$



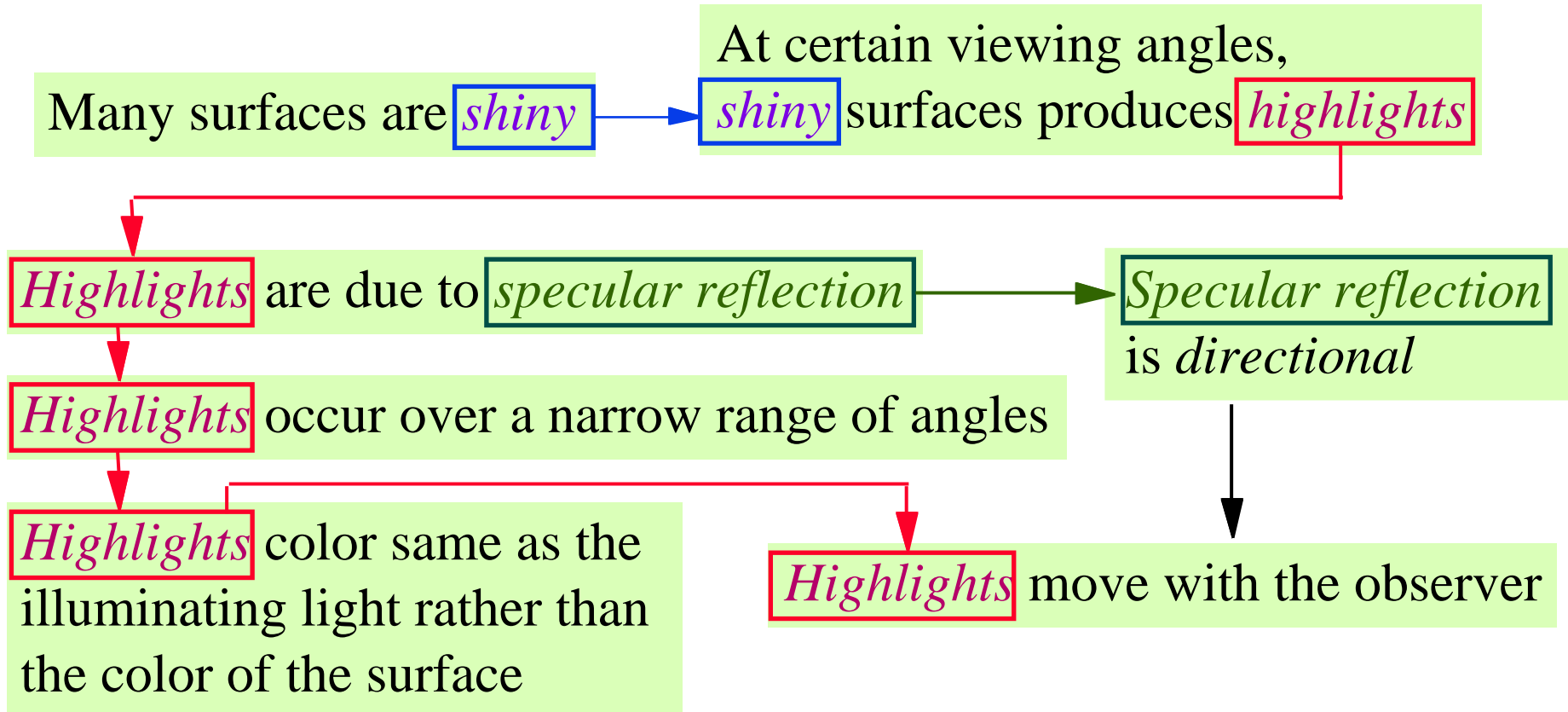
$$K = 0.6$$

$$K_d = 0.3$$



# *Illumination Models*

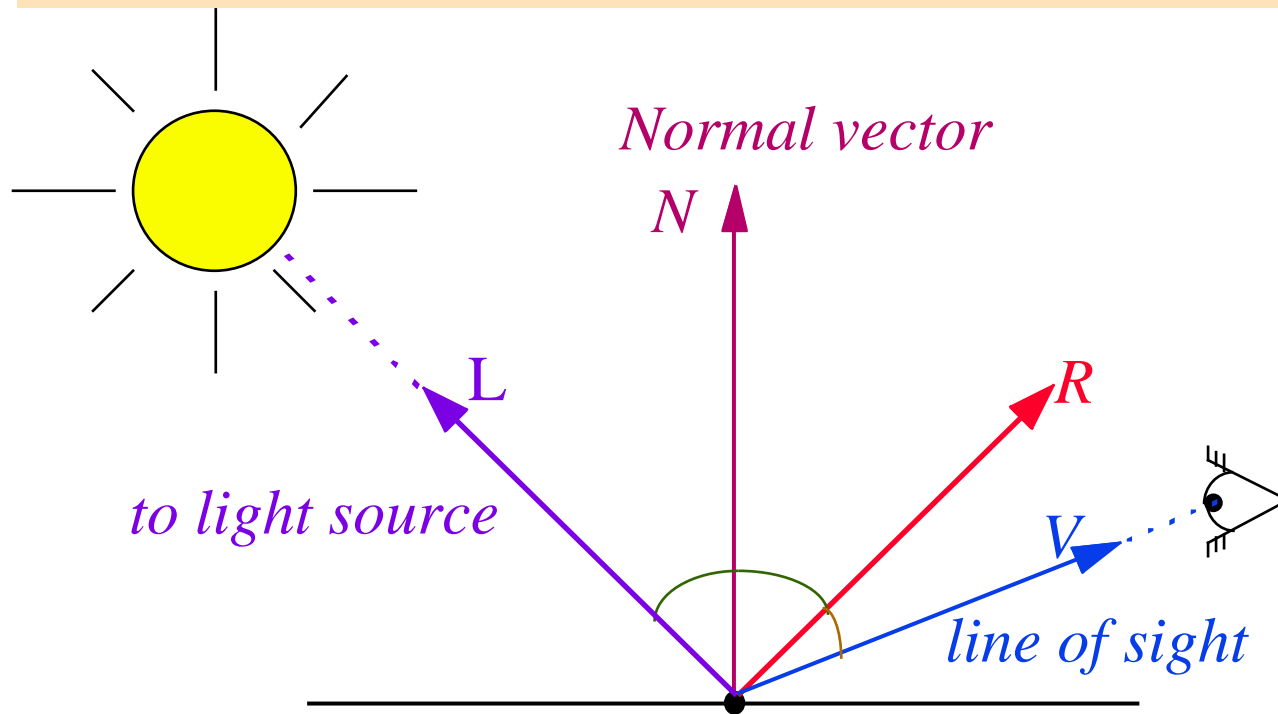
*Illumination model 3: Ambient + diffuse + specular light*



- For a **perfect reflecting** surface (a **mirror**) the angle of reflection is equal to the angle of incidence
- For **smooth** surfaces, the spatial distribution of specular light is narrow.
- For **rough** surfaces, it is spread out.

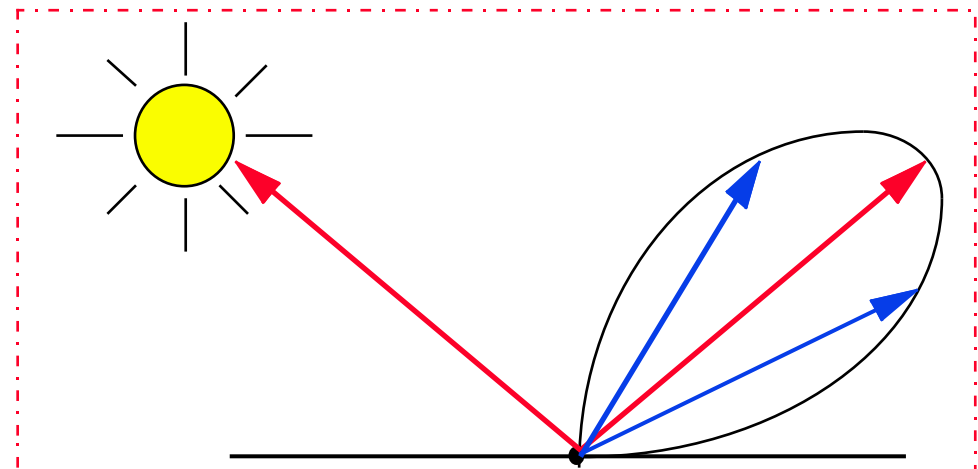
# Illumination Models

Illumination model 3: *Ambient* + *diffuse* + *specular light*



$L$  - light source direction  
 $N$  - Normal vector  
- angle of incidence  
 $V$  - line of sight  
 $R$  - direction of ideal specular reflection  
- angle between  $R$  and  $V$

If  $\theta = 0$ , we have a **perfect reflecting** surface (a **mirror**). An observer located here sees any specularly reflected light. **Otherwise**, we have a spatial distribution like:



# Illumination Models

Illumination model 3: *Ambient* + *diffuse* + *specular light*

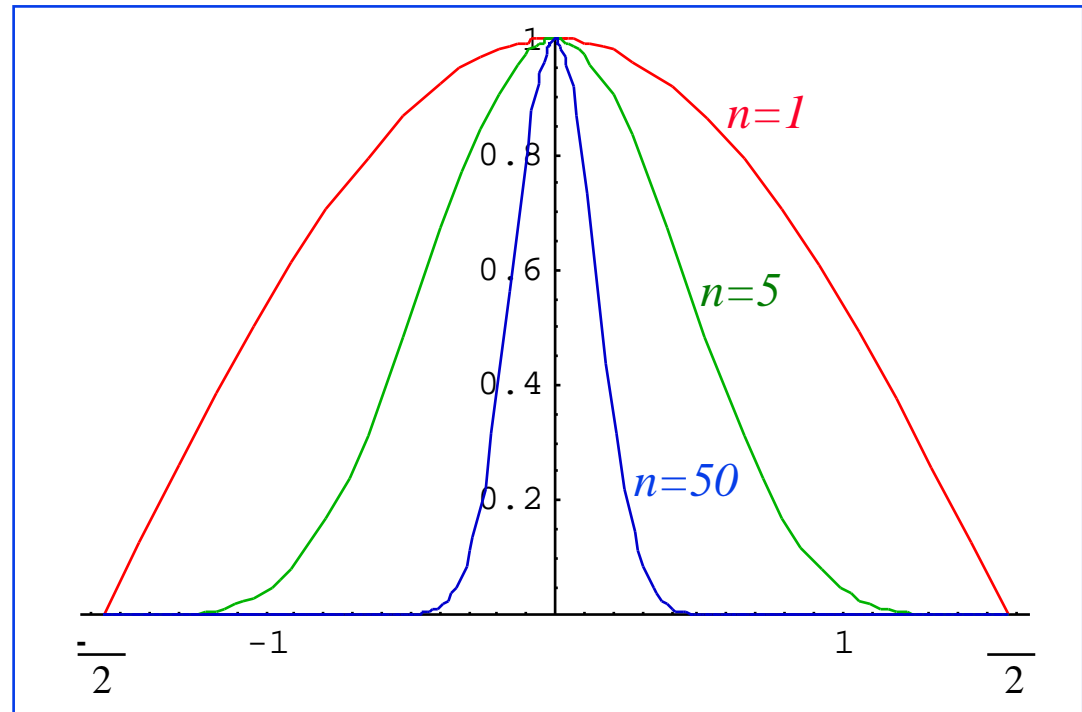
## Phong Model

Because of the complex physical characteristics of the specular light, an *empirical model* based on taking the function:

$$f(\theta) = \cos^n(\theta)$$

where  $n$  depends on surface properties. For:

- a perfect reflector,  $n = \infty$
- very poor reflector  $n = 1$
- in practice use  $1 \leq n \leq 200$



In general, we use:

**Larger** values of  $n$  for **metals** and other **shiny** surfaces

**Small** values of  $n$  for **nonmetallic** surfaces (e.g., paper)

# Illumination Models

*Illumination model 3: Ambient + diffuse + specular light*

Phong's empirical model controls the size of the specular highlight

incident intensity from a point light source

wavelength

reflectance curve

$$I_s(\lambda) = I_l(\lambda) w(i, \lambda) \cos^n(\theta)$$

intensity of reflected specular light

$w(i, \lambda)$ : ratio of the specularly reflected light to the incident light, as a function of the incidence angle,  $i$ , and the wavelength

Combining this term with model 2:

*Total Intensity light*

$$I(\lambda) = K(\lambda) I_l(\lambda) + I_l(\lambda) K_d(\lambda) \cos(\theta) + I_l(\lambda) w(i, \lambda) \cos^n(\theta)$$

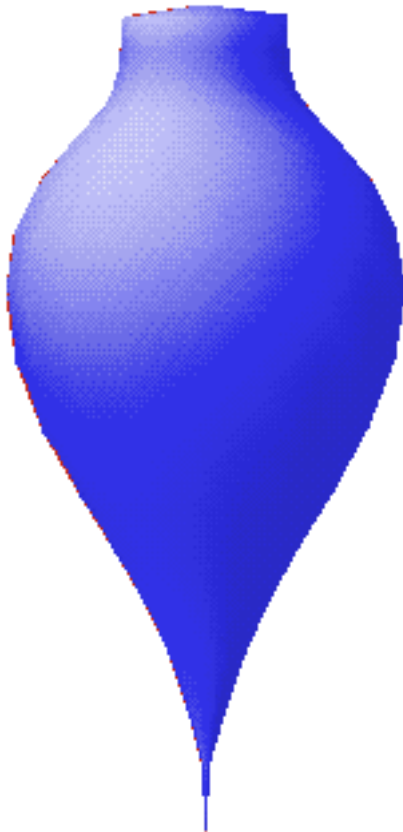
*Ambient light + Diffuse light + Specular light*

# Illumination Models

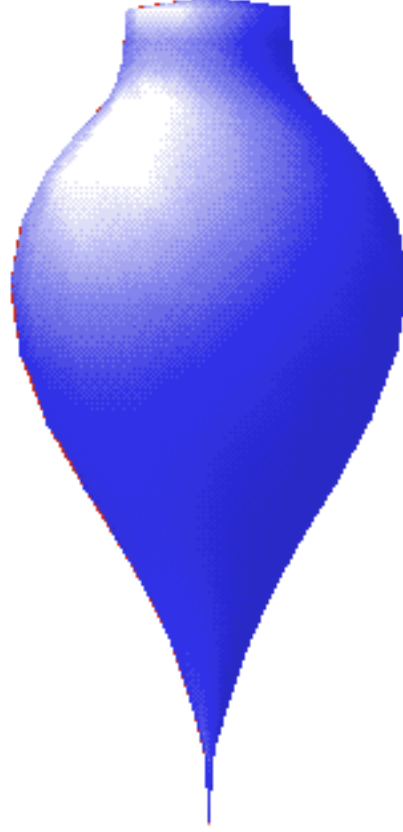
Illumination model 3: *Ambient* + *diffuse* + *specular light*

In practice, dependence on the wavelength is usually omitted. In addition,  $w(\mathbf{i}, \mathbf{n})$  is a very complex function, so it is replaced by an aesthetically or experimentally determined constant  $k_s$

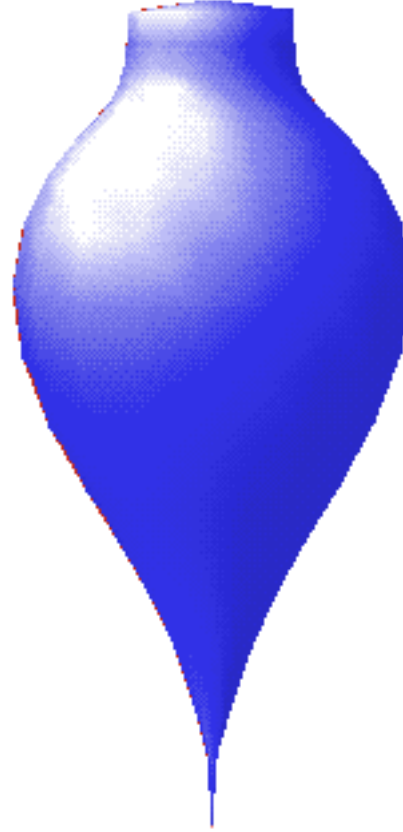
$K_s = 0.0$



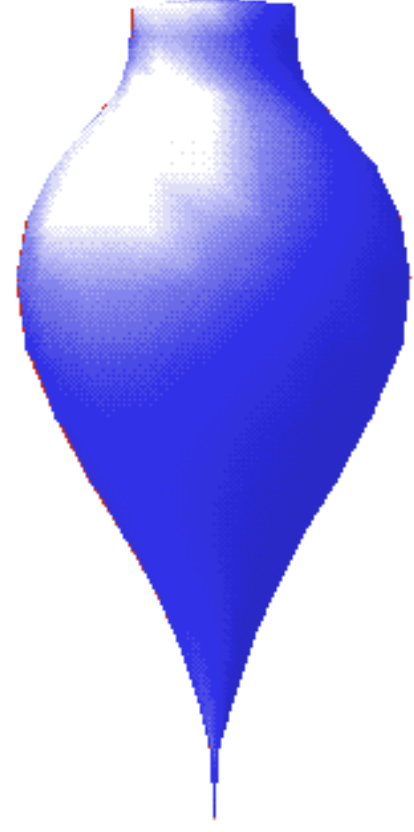
$K_s = 0.2$



$K_s = 0.4$



$K_s = 0.6$



# Illumination Models

Illumination model 3: *Ambient* + *diffuse* + *specular light*

$$I = K_a I_a + I_l K_d \cos(\theta) + I_l K_s \cos^n(\theta)$$

$K_a$  Ambient reflection

$K_d$  Diffuse reflection

$K_s$  Specular reflection

$$K = 0.6$$

$$K_d = 0.3$$

$$K_s = 0.2$$

$$K = 0.5$$

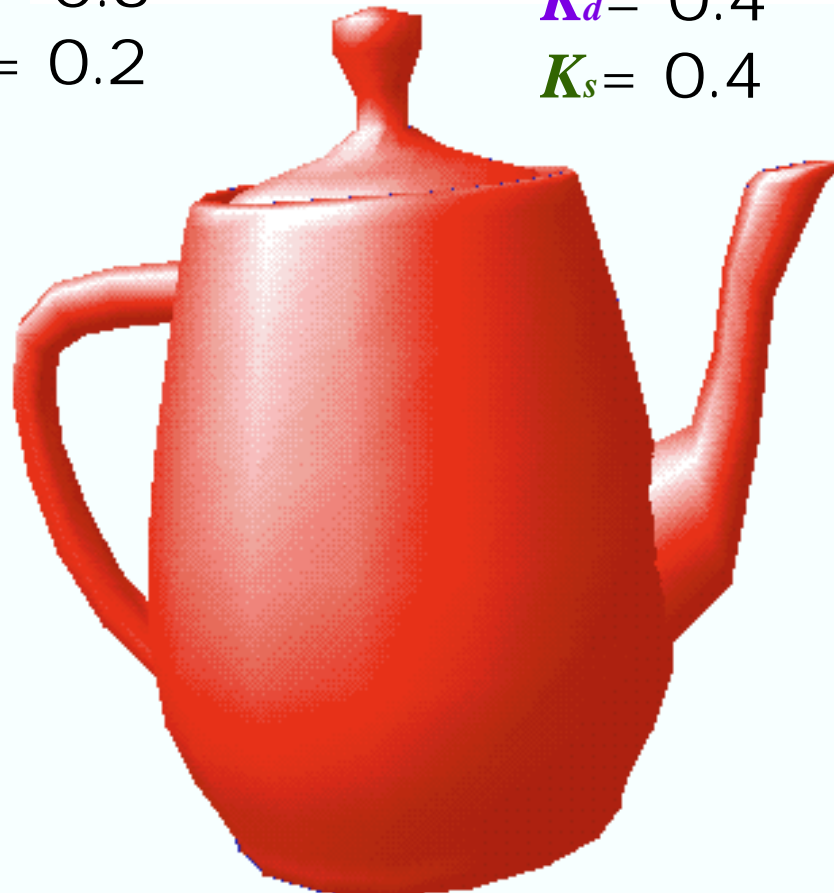
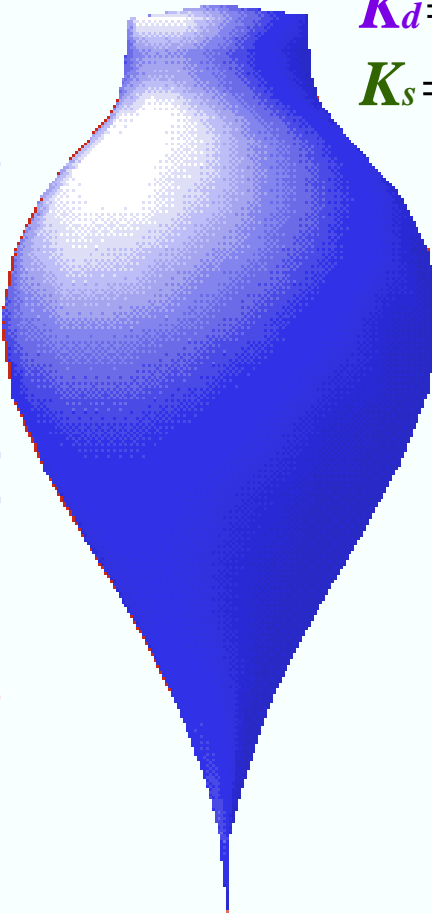
$$K_d = 0.4$$

$$K_s = 0.4$$

$$K = 0.7$$

$$K_d = 0.2$$

$$K_s = 0.8$$



# Illumination Models

Illumination model 3: *Ambient* + *diffuse* + *specular light*

Noting that:                      the model becomes:

$$\begin{aligned} \cos(\theta) &= \mathbf{N} \cdot \mathbf{L} \\ \cos(\phi) &= \mathbf{R} \cdot \mathbf{V} \end{aligned}$$

$$I = K I_0 + I_l K_d (\mathbf{N} \cdot \mathbf{L}) + I_l K_s (\mathbf{R} \cdot \mathbf{V})^n$$

However, two objects *at different distances* but with *the same orientation* to the light source exhibit *the same intensity*.

The intensity of light decreases inversely as the square of the distance.

*Objects farther away appear dimmer !!!!!*

$$I = \frac{I_{\text{source}}}{d^2}$$

**d** = source distance

In practice, this model produces unrealistic variations in intensity.

*Experimental model:*

$0 < p < 2$

$$I = \frac{I_{\text{source}}}{d^p + K}$$

**K** is an arbitrary constant used to prevent infinite intensity when **d=0**



# Illumination Models

Illumination model 4: *Ambient* + *diffuse* + *specular* + *attenuation light*

Phong, B. T. *Illumination for Computer Generated Images*, Communications of the ACM, Vol. 18, pp. 311-317, 1975.

*Phong Model:*

$$I = K I + \frac{I_l}{d^p + K} (K_d \cos(\ ) + K_s \cos^n(\ ))$$

$I = \text{Ambient} + \text{attenuation} ( \text{diffuse} + \text{specular} )$

$$I = K I + \frac{I_l}{d^p + K} (K_d (N.L) + K_s (R.V)^n)$$

Individual shading functions are used for each of the three primary colors

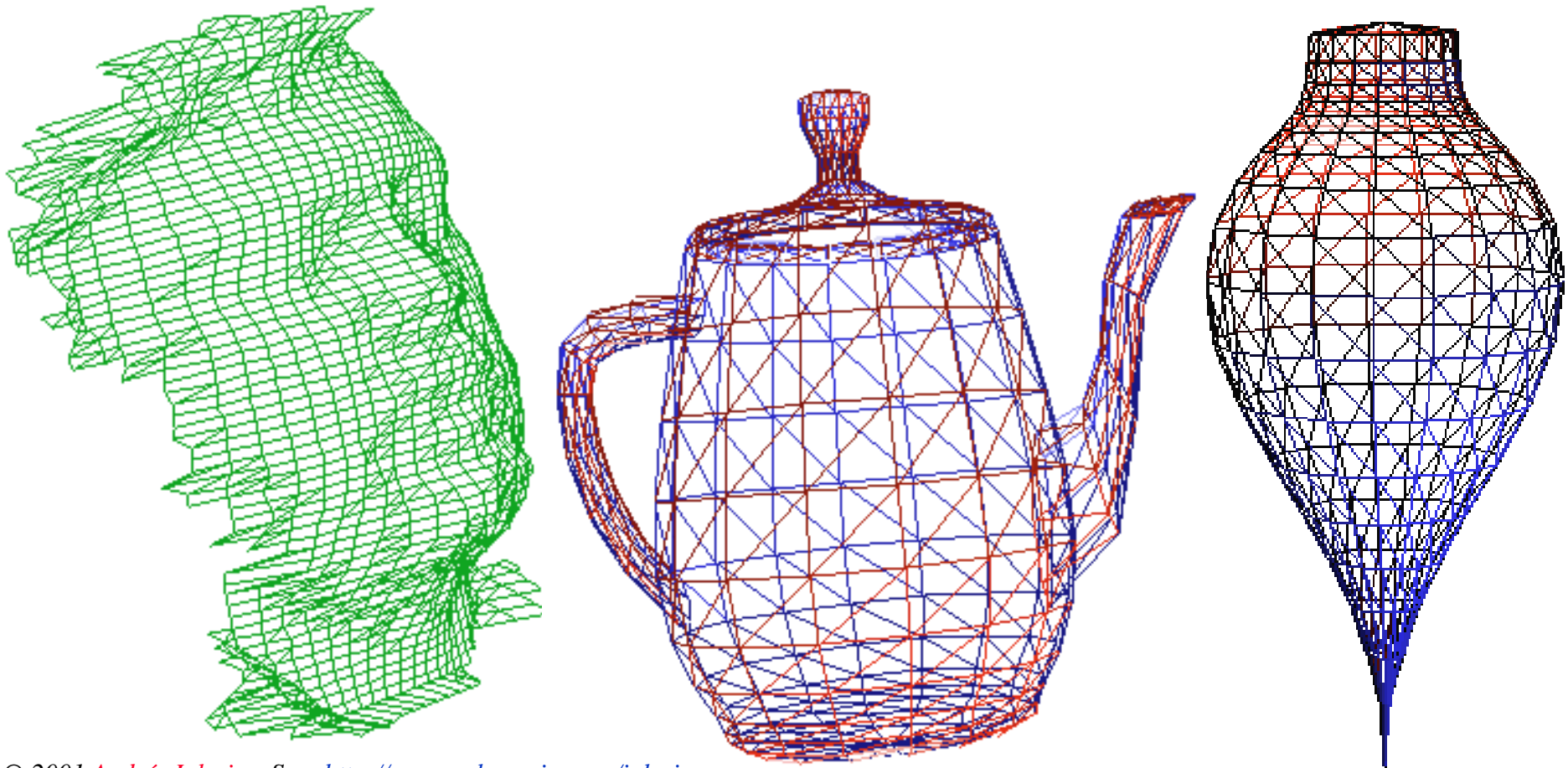
Multiple light sources: their effects are to be *linearly added*.

# *Illumination Models*

## *Polygon shading*

Until now, we compute the intensity light at a single point on a surface

*But, many objects are given by meshes of polygons !!!!!*



# Illumination Models

## Polygon shading

How to compute the intensity across the polygon ?

1. Compute the shade at all points:  
~~Unnecessary and impractical~~

2. Compute the shade at the centre and use this to represent the whole polygon: **Flat shading**

3. Compute shade at key points and interpolate for the rest:  
**Gouraud** and **Phong shading**

**Flat**



**Gouraud**



**Phong**



# *Illumination Models*

## *Polygon shading: Flat shading*

Is the simplest method and the most computationally efficient.

**But** it is also:

1. *Not realistic*: it exhibits polygon structure.

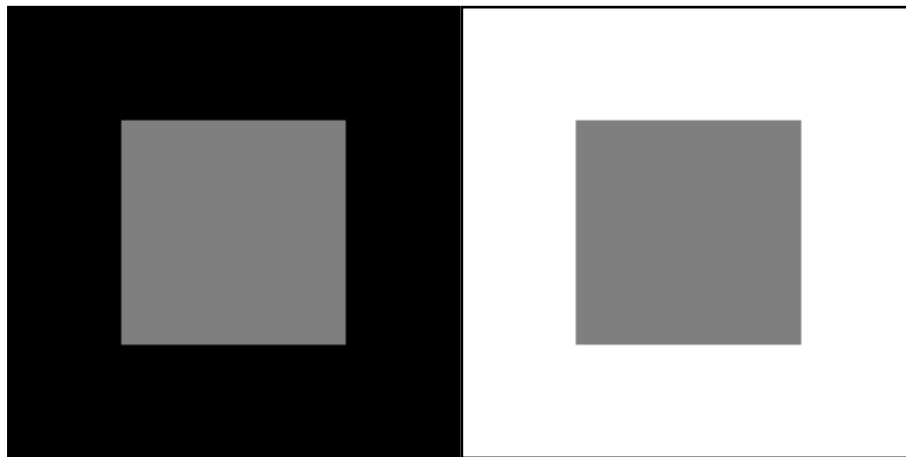
### *Faceted Appearance*



# Illumination Models

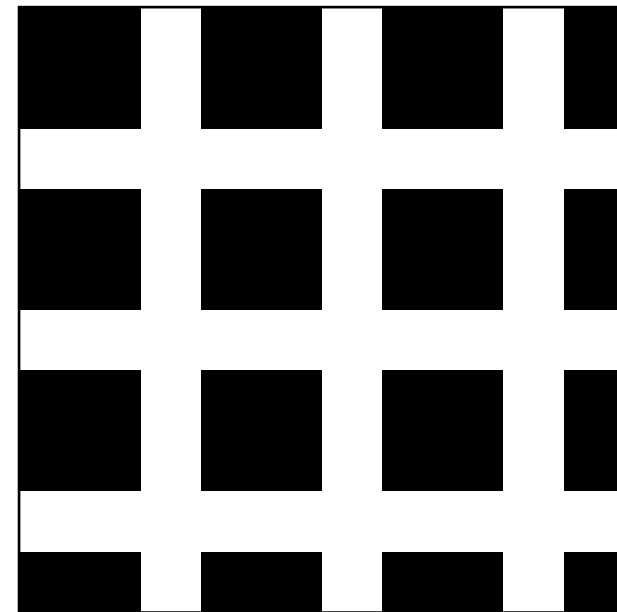
## Polygon shading: Flat shading

2. *Simultaneous contrast*: an area of constant brightness surrounded by a dark area is perceived to be brighter than the same area surrounded by a light area



$I_{\text{background}} = 1$	$I_{\text{background}} = 0$
$I_{\text{center}} = 0.5$	$I_{\text{center}} = 0.5$

3. *Mach banding*: brightness perceived by the eye tends to overshoot at the boundaries of regions of constant intensity



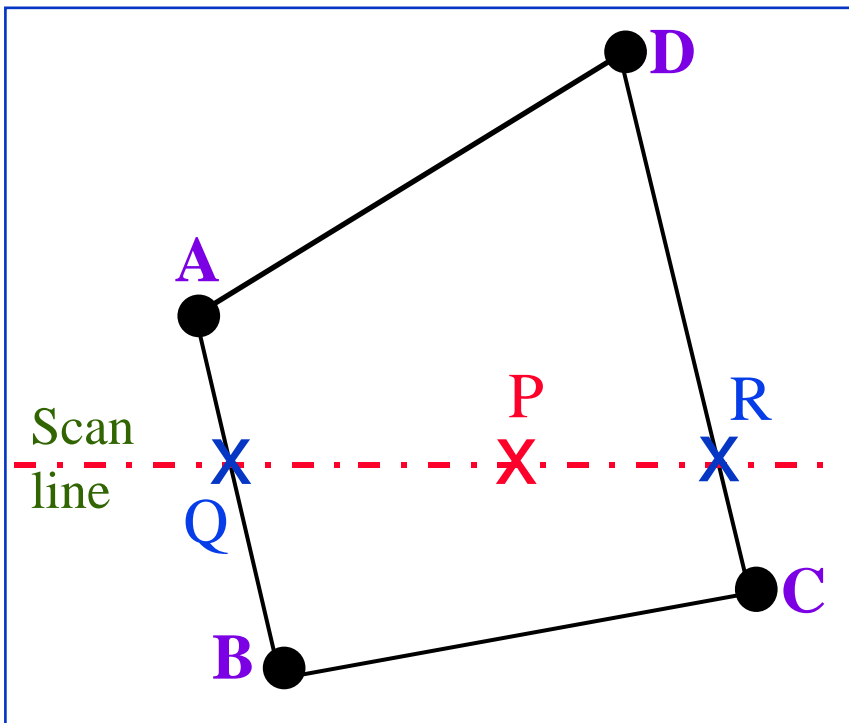
Abrupt changes in the shading of two adjacent polygons are perceived to be even greater

# Illumination Models

## Polygon shading: Gouraud shading

Gouraud, H. *Computer Display of Curved Surfaces*, IEEE Transactions on Comput., C-20, pp. 623-628, 1971.

Given a polygon and a scan-line, the problem is to determine the intensity at an interior point, such as **P**



1. First compute the intensity values at each polygon vertex

Output:  $I_A, I_B, I_C, I_D$

2. Next compute the intensity at points **Q** and **R** using linear interpolation

Output:  $I_Q, I_R$

3. Finally, linearly interpolate between  $I_Q$  and  $I_R$  to get intensity at point **P**

Output:  $I_P$

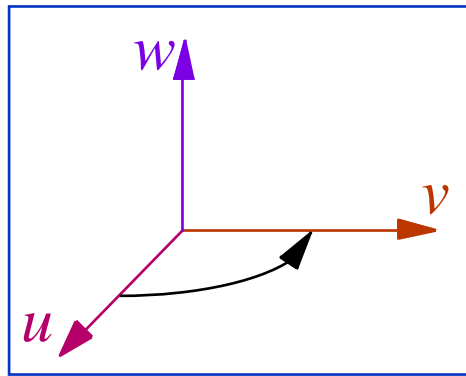
# Illumination Models

## Polygon shading: Gouraud shading

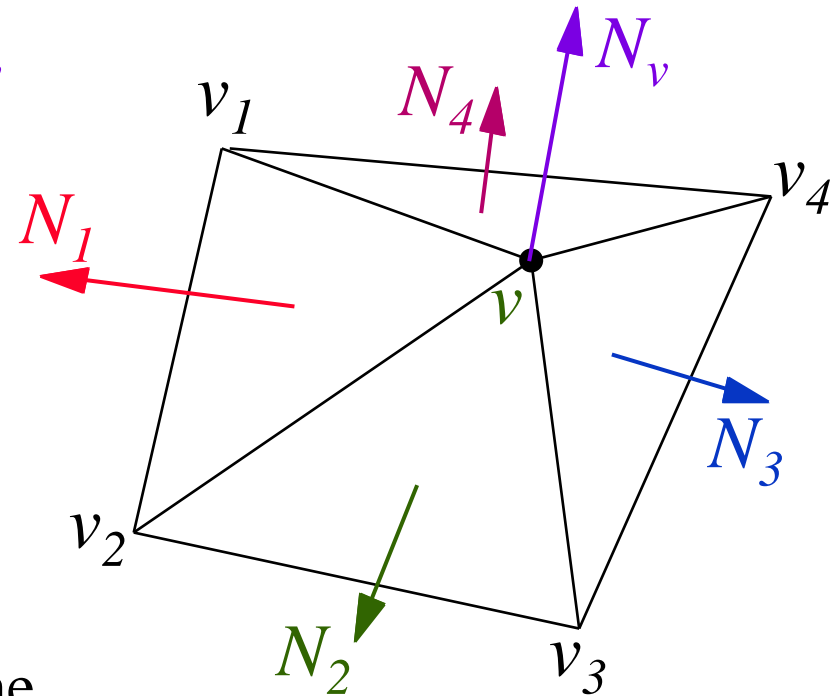
1. First compute the intensity values at each polygon vertex

We need the vertex normal vector  $N_v$

### Cross product



$$w = u \times v = \begin{vmatrix} i & j & k \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix}$$



averaging the cross products  $\times$  of all the edges that terminate at the vertex

$$N_v = \underbrace{v.v_1 \times v.v_2}_{N_1} + \underbrace{v.v_2 \times v.v_3}_{N_2} + \underbrace{v.v_3 \times v.v_4}_{N_3} + \underbrace{v.v_4 \times v.v_1}_{N_4}$$

# Illumination Models

## Polygon shading: Gouraud shading

2. Next compute the intensity at points **Q** and **R** using linear interpolation

Intensity at a point **Q**:

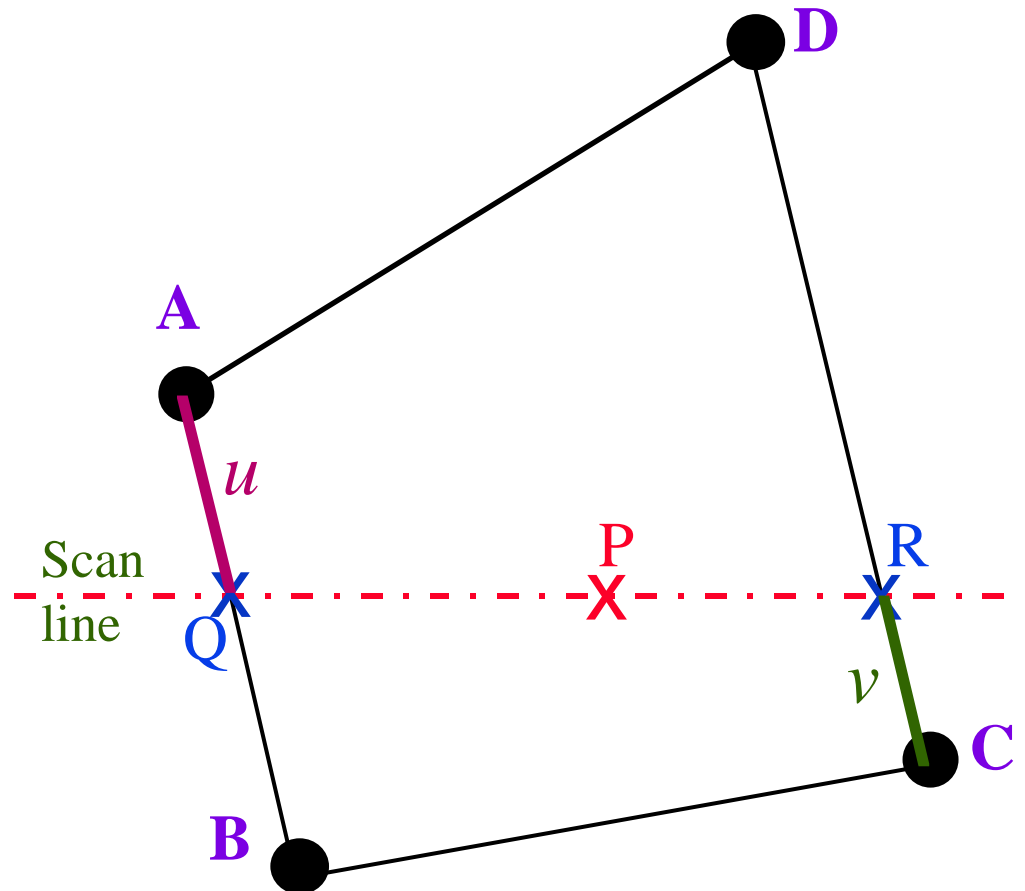
$$I_Q = u I_A + (1-u) I_B$$

where  $u = \frac{AQ}{AB}$

Intensity at a point **R**:

$$I_R = v I_C + (1-v) I_D$$

where  $v = \frac{CR}{CD}$





# Illumination Models

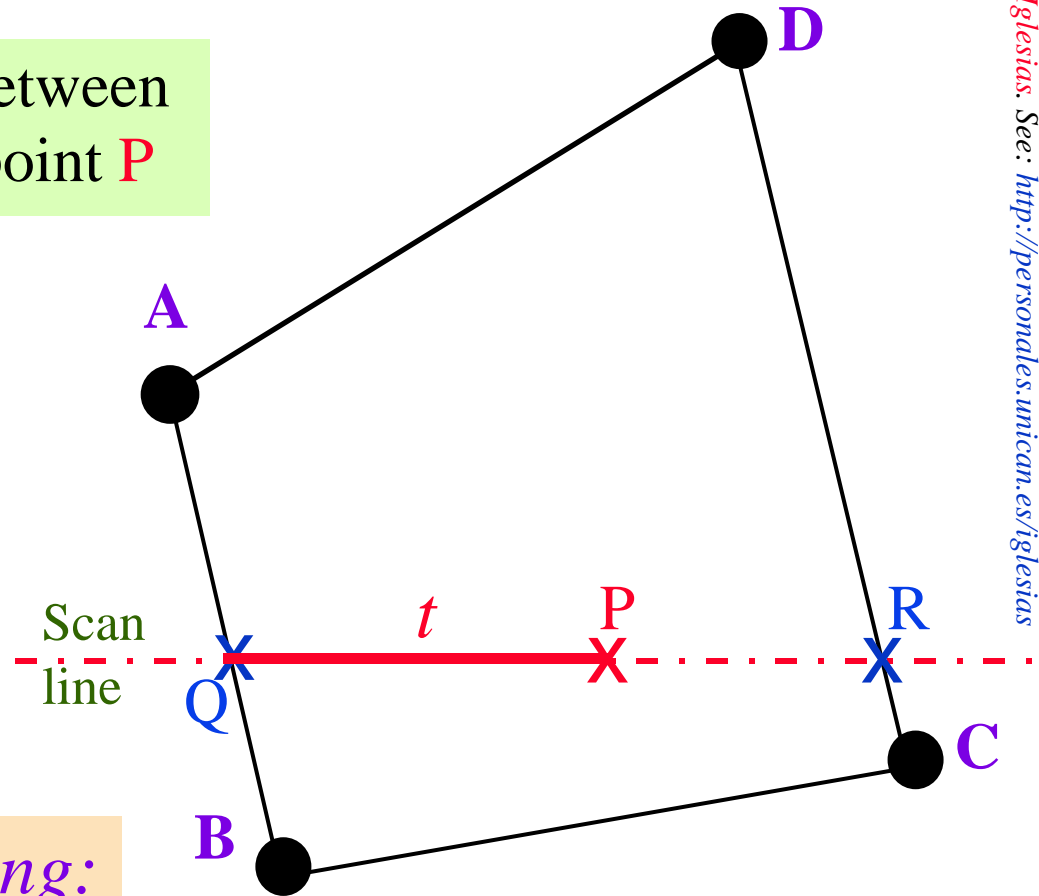
## Polygon shading: Gouraud shading

3. Finally, linearly interpolate between  $I_Q$  and  $I_R$  to get intensity at the point  $P$

Intensity at a point  $P$ :

$$I_P = t I_Q + (1-t) I_R$$

where  $t = \frac{QP}{QR}$



### Problems of Gouraud shading:

- This method yields only to continuity of intensity, but not continuity of changes of intensity => *Mach banding*
- Silhouette edges are still **polygonal**

# Illumination Models

## Polygon shading: Phong shading

It is similar to Gouraud shading, except we linearly interpolate the *surface normal vector* across the polygon

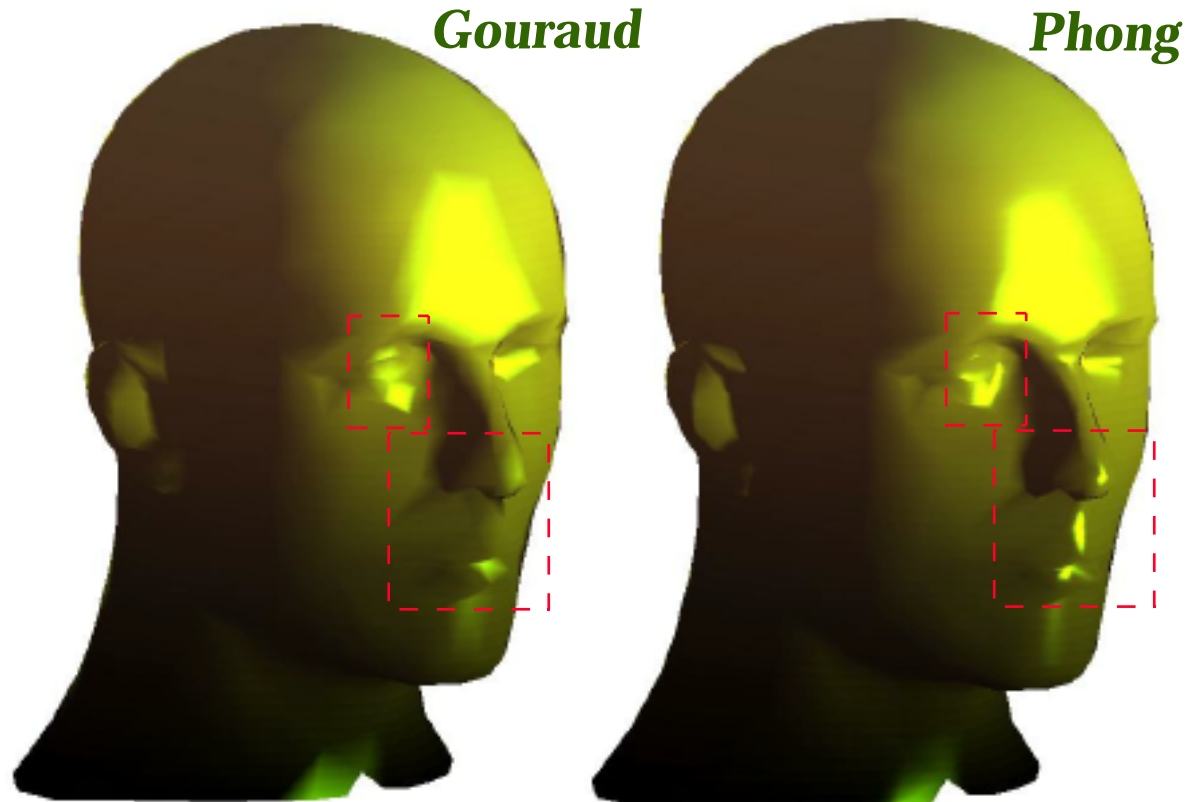
Determine:

$N_A, N_B, N_C, N_D$

$$N_Q = u N_A + (1-u) N_B$$
$$N_R = v N_C + (1-v) N_D$$

$$N_P = t N_Q + (1-t) N_R$$

Gouraud shading loses specular reflection if the highlight lies inside a single polygon



It is computationally more expensive.