Illumination and Shading

Illumination Models

- An **illumination model** is used to compute the intensity (colour) of light that we should see at a given point on the surface of an object.
- A **shading method** uses the results from applying the illumination model to compute the intensity of light for all pixels in the image corresponding to objects' surfaces.
- Shading methods can apply the illumination model to every visible surface point or by interpolating intensities across the surfaces from a small set of illumination-model calculations.
  - rasterisers (such as OpenGL) often use interpolation
  - ray tracing algorithms compute illumination at each pixel position (and at many more locations)

Illumination Models

- Photorealism requires:
  - Representing objects in an accurate way
  - good physical description of lighting effects in a scene
- Lighting effects are described with models that handle the interaction of light with object surfaces.
- As light enters our eyes, perception processes take place to determine what we actually "see".
- Physical illumination models take into account several factors: type (material) of object, relative position to light sources and other objects and light source conditions set up for the scene.
- Materials: opaque vs. transparent? dull vs. shiny? surface texture pattern?
- Light sources: shape? colour? position?

Illumination Models

Given:

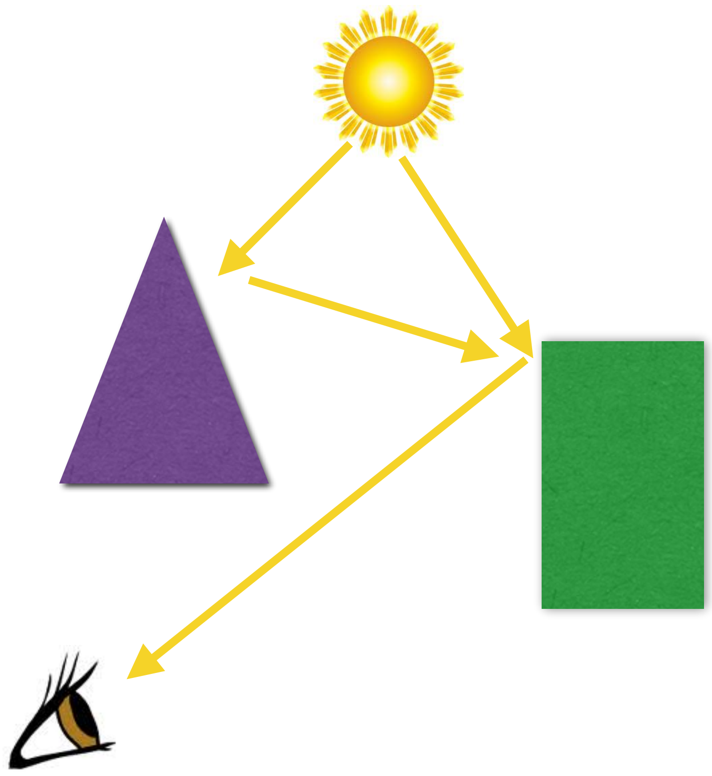
- optical properties of surfaces
- relative positions of surfaces in a scene
- colours and positions of the light sources
- a view specification

an illumination model will compute the intensity (colour) projected from a particular surface point in a specified viewing direction.

In CG, illumination models are often loosely derived from physical laws and often use empirical approximations to reduce computational complexity.

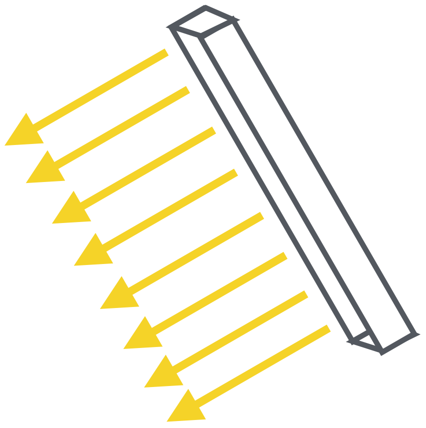
Light Sources

- The light that gets reflected by an object is the sum of contributions from **light sources** (light-emitting sources) and other reflecting surfaces (light-reflecting sources)
- types of sources:
  - point source
  - distributed light source
  - spotlight



Point light source

area of the light source is relatively small compared to object sizes

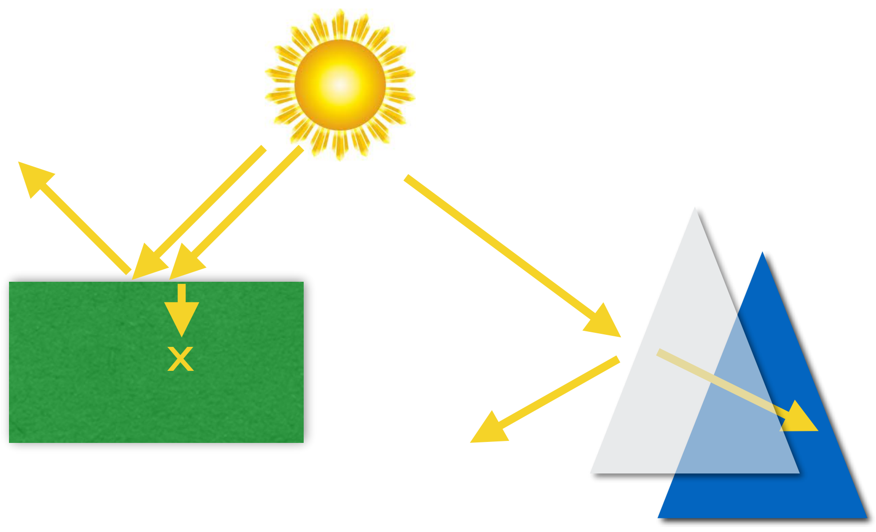


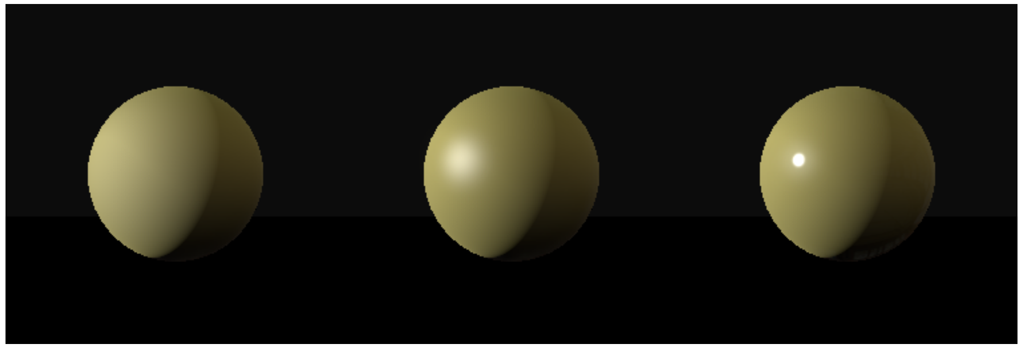
Distributed light source

area of the light source is not small compared to object sizes

Materials and Reflection

- Light incident on an opaque surface is partially reflected and partially absorbed
- For transparent materials, light is not only partially reflected but also partially transmitted through the material
- Shiny materials reflect more light while dull ones absorb more of the incoming light
- Rough or grainy surfaces tend to scatter the reflected light in all directions (**diffuse reflection**) and the surface appears the same from all viewing directions
- Highlights (bright spots) are more pronounced in shiny surfaces (**specular reflection**)





rough shiny

Ambient Light

- A surface not exposed directly to any light source will still be visible if nearby objects are illuminated
- A simple approximation is to consider that there is some amount of background or ambient light, constant through the whole scene and reaching every point
- It can be modelled by using a parameter I_a (ambient light level) affecting all surfaces
- However, the reflected light from each surface will depend on its optical parameters.

Basic Illumination Models

Ambient Light (Achromatic case)

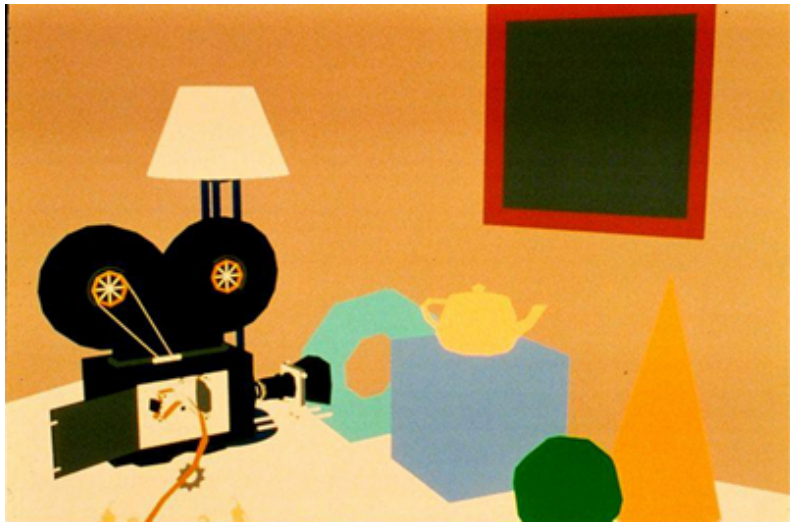
- The fraction of diffuse incident light that is reflected can be modelled by a surface parameter K_a in [0,1]
- K_a is an empirical convenience and it does not correspond to any physical material property
- The amount of light reflected can be modelled by:

$$I = K_a I_a$$

Observed intensity (I)

Fraction of light that gets reflected (K_a)

Intensity of light arriving on surface (I_a)

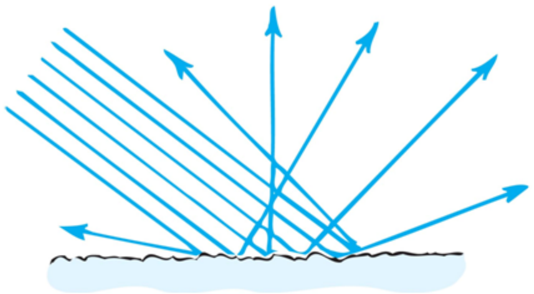


From Computer Graphics Principles and Practice, 2nd ed.

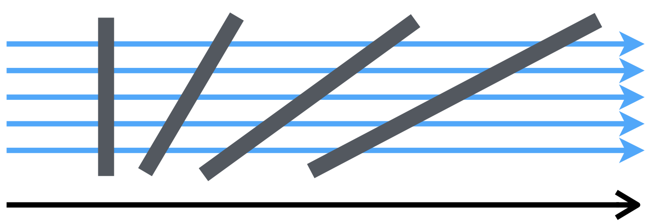
Basic Illumination Models

Diffuse Reflection

- Diffuse reflection does not depend on the view direction because light is equally reflected in all directions



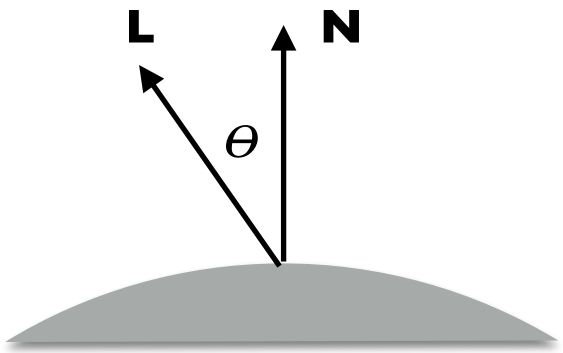
- The amount of light arriving per unit area depends on the angle of incidence



increasing area ⇒ less light per unit area

- Once again, the fraction of incident light that that is diffusely reflected can be modelled by a surface parameter K_d in [0,1]

- Brightness depends only on the angle θ between the vector pointing towards the light source **L** and the normal vector **N**

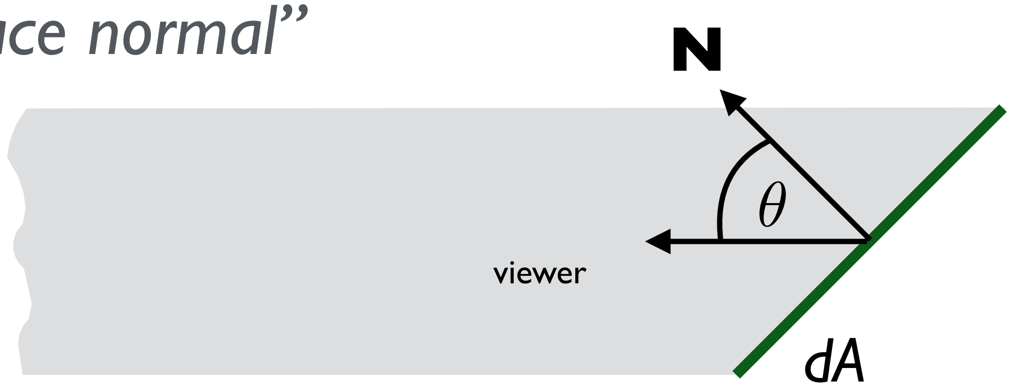


From Computer Graphics Principles and Practice, 2nd ed.

Diffuse Reflection (Physical explanation)

- Surfaces with this behaviour are called *Lambertian* reflectors, obeying Lambert's cosine law:

"the amount of light reflected from a unit differential area dA towards the viewer is directly proportional to the cosine of the angle between the direction of the viewer and the surface normal"



- Since the amount of surface area seen is inversely proportional to the cosine of that angle, the two factors cancel out.
- As the viewing angle increases, the viewer sees more surface area but the amount of light reflected at that angle per unit area is proportionally less.

From Computer Graphics Principles and Practice, 2nd ed.

Basic Illumination Models

Diffuse Reflection

- The diffuse illumination equation is thus:

$$I = I_p K_d \cos\theta$$

point light source intensity

material's diffuse reflection coefficient in [0,1]

angle between **N** and **L** in [0°,90°]

- If both **N** and **L** are normalised then we can write:

$$I = I_p K_d (\mathbf{N} \cdot \mathbf{L})$$

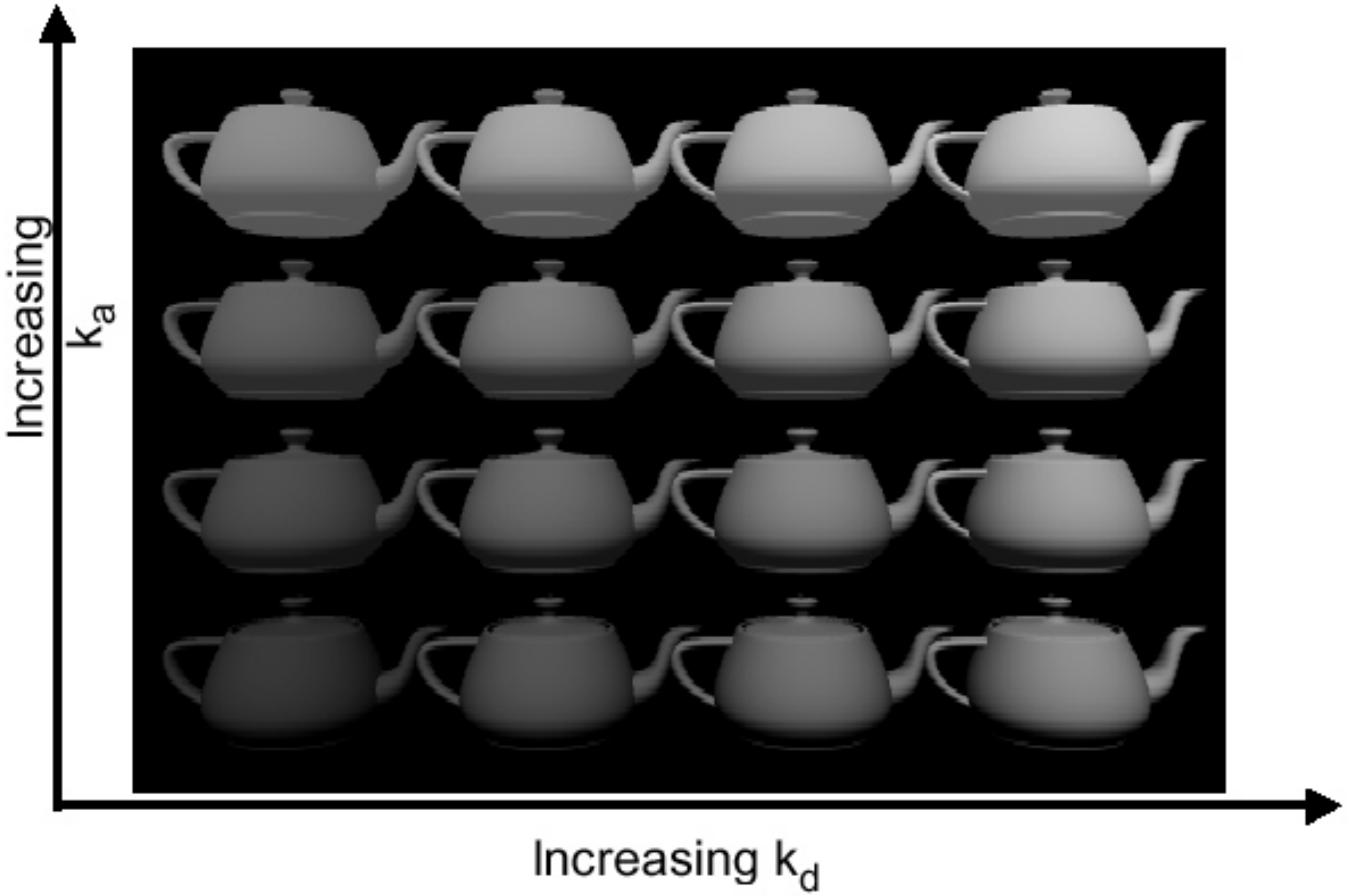
Note: If light is far away from the object, **L** is essentially constant through the whole surface and the light source becomes a directional light source

Note: In the pipeline, Illumination has to be performed in WC or in eye coordinates otherwise **N** may no longer represent a surface normal!

Ambient + Diffuse Reflection

- Combining both terms yields:

$$I = I_a K_a + I_p K_d (\mathbf{N} \cdot \mathbf{L})$$



Coloured lights and surfaces

- Coloured lights and surfaces can be treated by sampling light at different wavelengths
- We can write an equation identical to the monochromatic light case for each wavelength λ to be considered:

$$I_\lambda = I_{a,\lambda} K_a O_{d,\lambda} + I_{p,\lambda} K_d O_{d,\lambda} (\mathbf{N} \cdot \mathbf{L})$$

- Instead of covering the whole spectrum of visible light (too many wavelengths), we usually sample for red, green and blue light.
- $\mathbf{O}_d$* is a vector that represents the colour of the material (one component per each wavelength), while K_d (scalar) represents the fraction of incident light that is reflected

* sometimes $K_a\mathbf{O}_d$ and $K_d\mathbf{O}_d$ are replaced by $\mathbf{K}_a$ and $\mathbf{K}_d$ *(both vectors)*

Light source attenuation

- Energy from a light source reaching some surface falls off with the inverse of the square of the distance d_L from the surface to the light source
- With the introduction of a f_{att} factor we could write:

$$I_\lambda = I_{a,\lambda} K_a O_{d,\lambda} + f_{att} I_{p,\lambda} K_d O_{d,\lambda} (\mathbf{N} \cdot \mathbf{L})$$

- A natural choice for f_{att} would be:

$$f_{att} = \frac{1}{{d_L}^2}$$

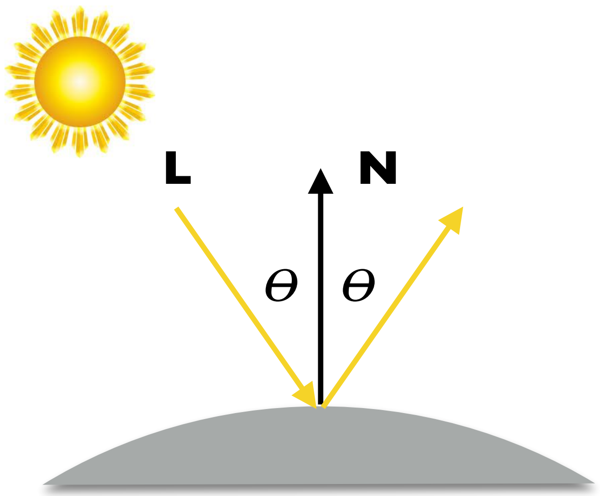
- But in practice it doesn't work well. Usually lights are not point light sources and illumination is more complex than our simple models. A better approximation is:

$$f_{att} = min(1, \frac{1}{c_1 + c_2 d_L + c_3 d_L^2})$$

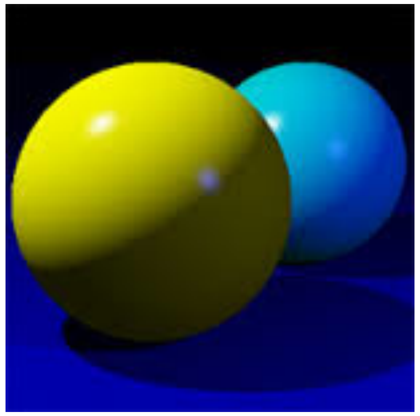
Basic Illumination Models

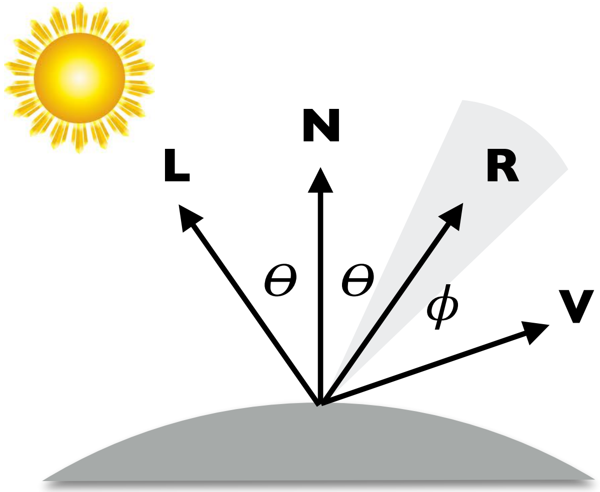
Specular Reflection

- Perfect reflectors (mirrors) reflect light according to the reflection law
- specular reflection is the result of total or near total reflection of the incident light in a concentrated region, creating highlights
- Non-ideal reflectors exhibit specular reflections over a finite range of viewing directions near **R**



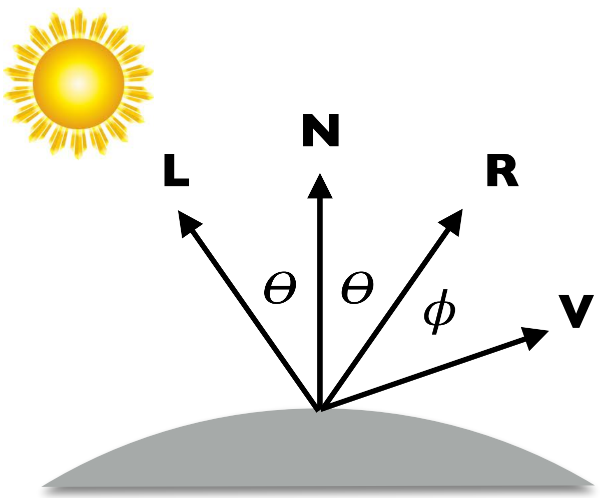
perfect reflector





Phong illumination model

- Developed by Phong Bui-Tuong in 1975
- Maximum specular reflection is obtained when **V**=**R** (ϕ=0) and falls rapidly as ϕ increases.
- falloff is modelled by a term $\cos^n\phi$



Specular Term

$$I_{p,\lambda} W(\theta) \cos^n \phi$$

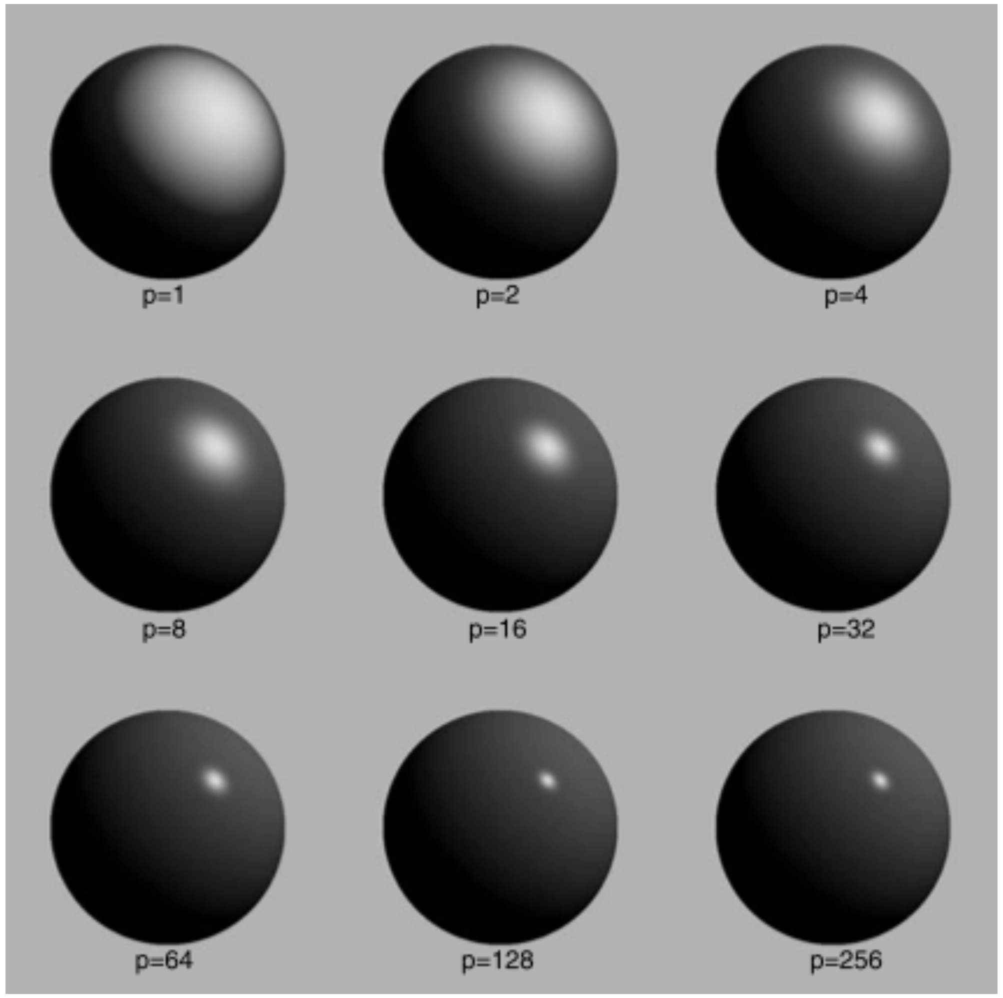
For real objects, specular coefficient is a function of θ. In practice it is usually approximated by a constant value K_s

Basic Illumination Models

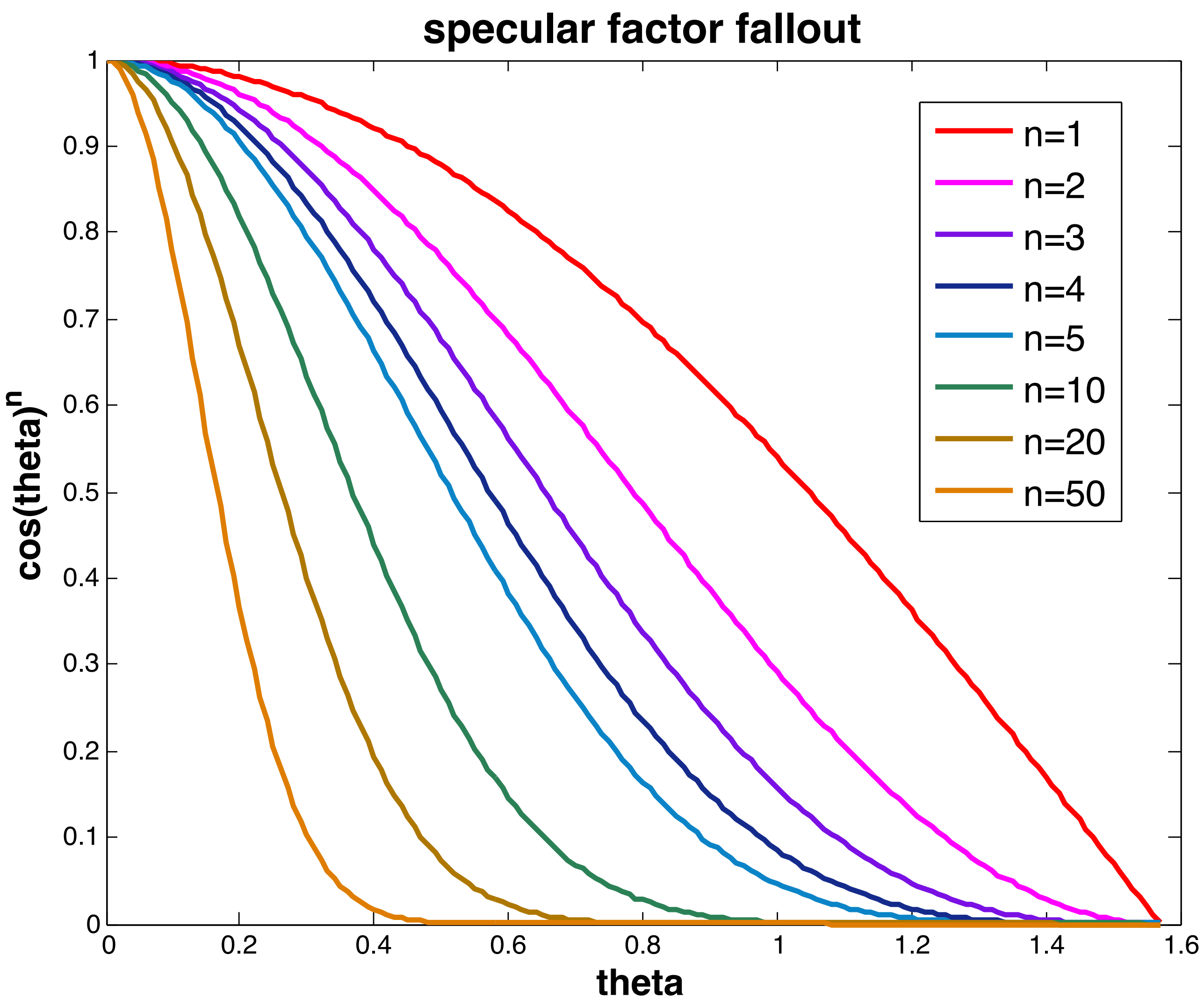
Specular Term

$$I_{p,\lambda} W(\theta) \cos^n \phi$$

varying the value of n (p in this example)



***from "Fundamentals of Computer Graphics, 3rd Ed."**



Phong illumination model

If **R**, **L** and **N** are normalised vectors, and $W(\Theta)$ is set to a constant K_s, then we write:

$$I_\lambda = I_{a,\lambda} K_a O_{d,\lambda} + f_{att} I_{p,\lambda} [K_d O_{d,\lambda} (\mathbf{N} \cdot \mathbf{L}) + K_s (\mathbf{R} \cdot \mathbf{V})^n]$$

Note that specular highlights will have the same colour as the light source illuminating the object. This is good for most objects but not for all…

Phong illumination model

We can accommodate more complex specular reflections by introducing a specular colour $O_{s,\lambda}$:

$$I_\lambda = I_{a,\lambda} K_a O_{d,\lambda} + f_{att} I_{p,\lambda} [K_d O_{d,\lambda} (\mathbf{N} \cdot \mathbf{L}) + K_s O_{s,\lambda} (\mathbf{R} \cdot \mathbf{V})^n]$$

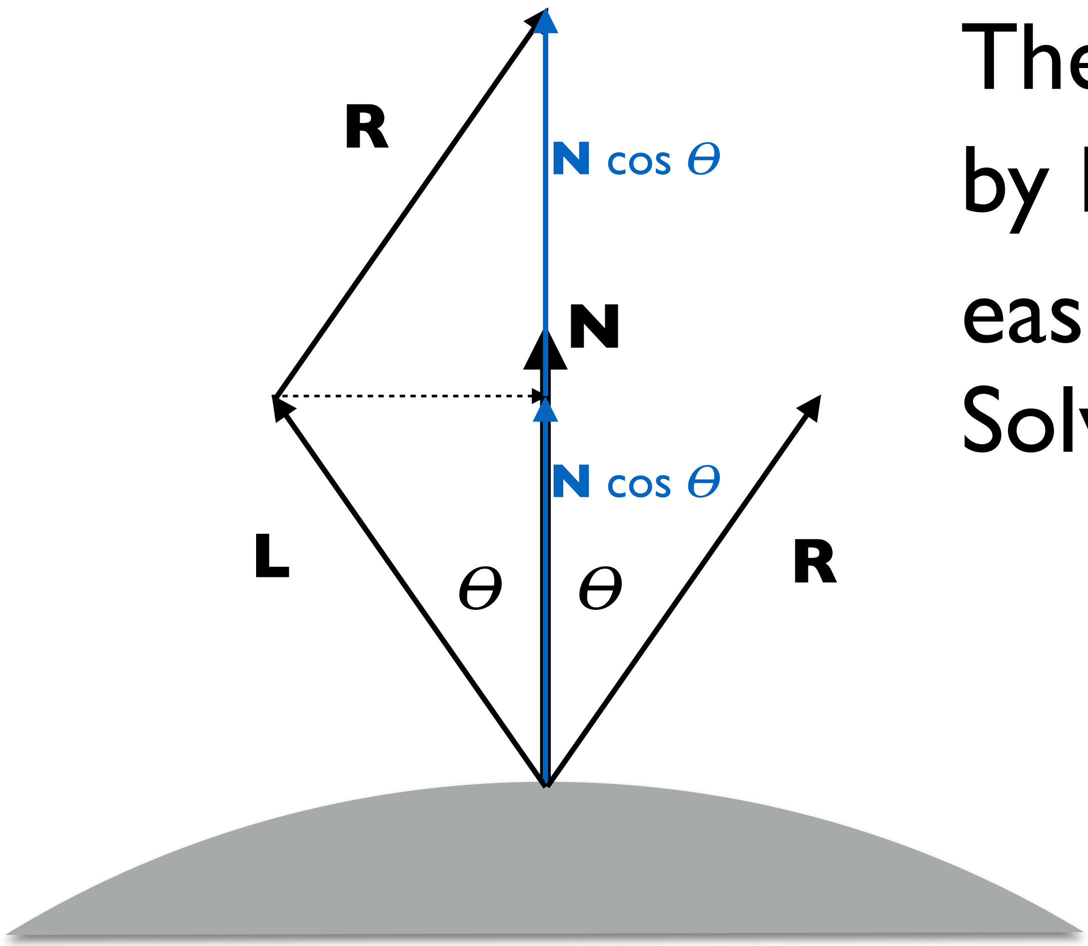
The above can be rewritten using vectors with each component assigned to a different wavelength as:

$$\mathbf{I} = \mathbf{I}_a K_a \mathbf{O}_d + f_{att} \mathbf{I}_p [K_d \mathbf{O}_d (\mathbf{N} \cdot \mathbf{L}) + K_s \mathbf{O}_s (\mathbf{R} \cdot \mathbf{V})^n]$$

Normally, we only use λ=R,G,B, thus $\mathbf{O}_d=(O_{dr},O_{dg},O_{db})$, for instance.

Practical considerations:

- To compute **R** we need to mirror **L** about **N**:



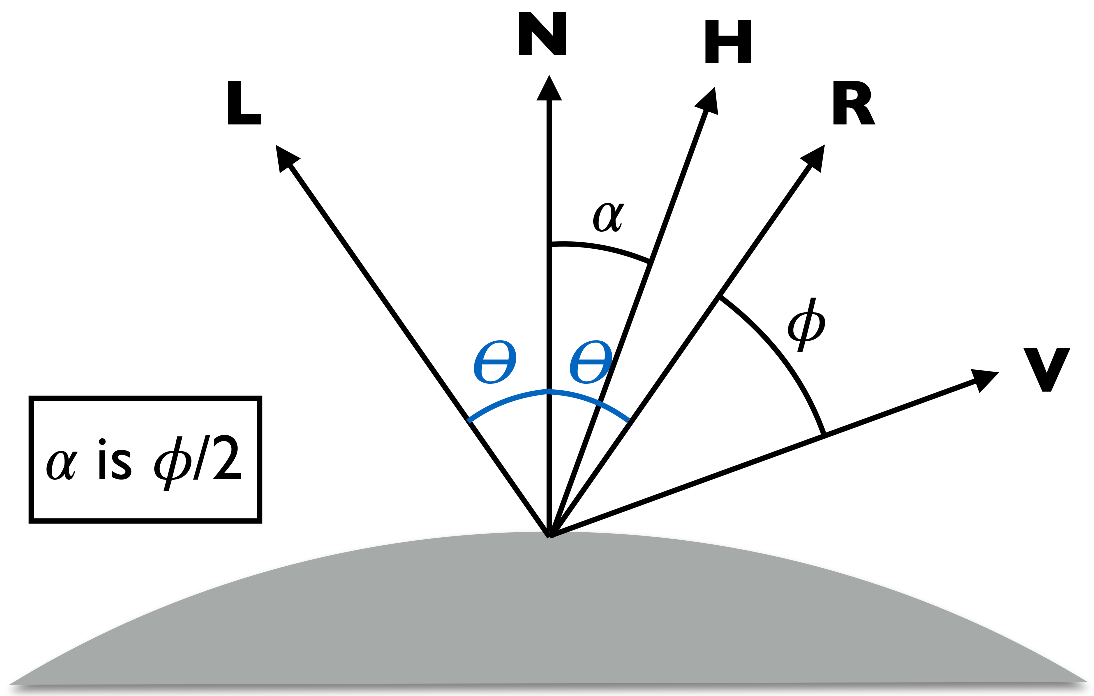
The projection of **L** onto **N** is given by **N** cos θ, or **N** (**N** · **L**). We can easily see that **L** + **R** = 2 **N** (**N** · **L**). Solving for **R** we get:

$$\mathbf{R} = 2\,\mathbf{N}\,(\mathbf{N} \cdot \mathbf{L}) - \mathbf{L}$$

The halfway vector formulation

- An approximation to Phong's specular component is to use the halfway vector **H**

$$\mathbf{H} = \frac{\mathbf{L} + \mathbf{V}}{\|\mathbf{L} + \mathbf{V}\|}$$



Note: The halfway formulation assumes that L, N, H, R and V are all in the same plane

Replacing the empirical cos ϕ $(\mathbf{R} \cdot \mathbf{V})$ with the empirical cos α $(\mathbf{N} \cdot \mathbf{H})$.

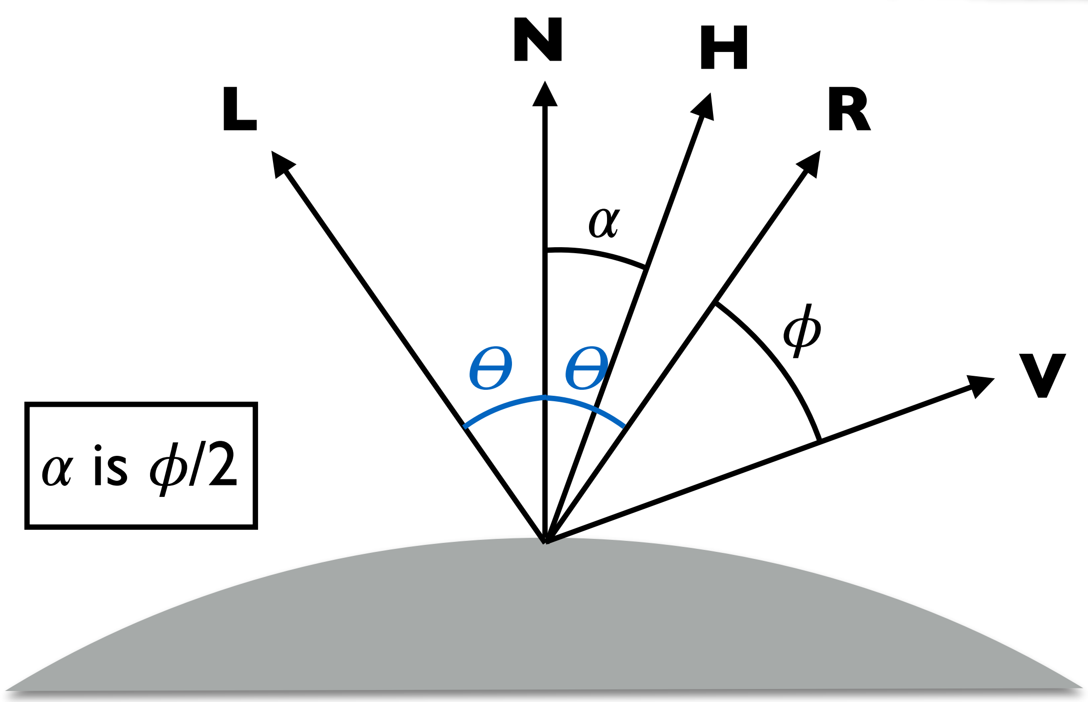
For lights and viewers far away, it is less expensive because **H** is constant over the surface, avoiding to compute **R**. For curved surfaces **N** is still varying over the surface.

The halfway vector formulation

$$\mathbf{I} = \mathbf{I}_a K_a \mathbf{O}_d + f_{att}\mathbf{I}_p\left[k_d \mathbf{O}_d (\mathbf{N}\cdot\mathbf{L}) + K_s \mathbf{O}_s (\mathbf{N}\cdot\mathbf{H})^n\right]$$

diffuse color

specular color



Sometimes the reflection coefficients are premultiplied by the respective colours:

$$\begin{cases} \mathbf{K}_a = K_a \mathbf{O}_d \\ \mathbf{K}_d = K_d \mathbf{O}_d \\ \mathbf{K}_s = K_s \mathbf{O}_s \end{cases}$$

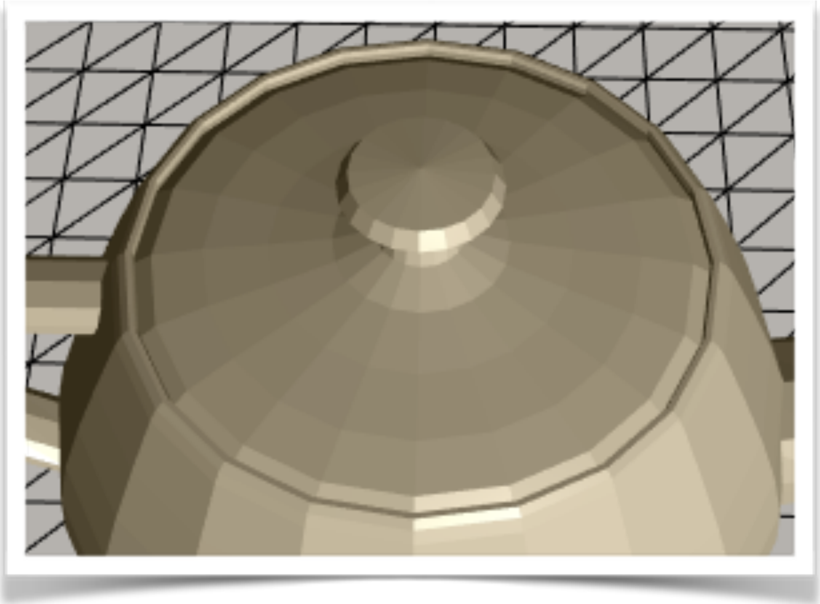
Multiple light sources

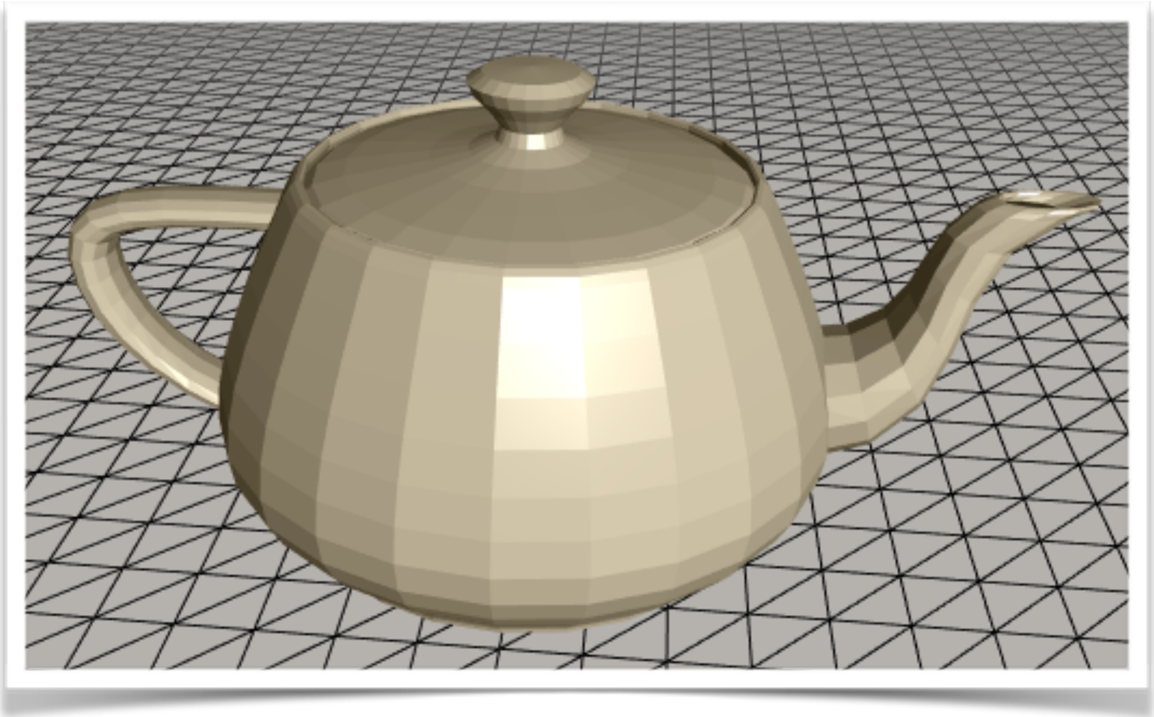
$$\mathbf{I} = \mathbf{I}_a K_a \mathbf{O}_d + \sum_{l=1}^{L} f_{att,l} \mathbf{I}_{p,l} [k_d \mathbf{O}_d (\mathbf{N} \cdot \mathbf{L}) + K_s \mathbf{O}_s (\mathbf{N} \cdot \mathbf{H})^n]$$

- For the diffuse and specular terms, we sum up all the contributions from each light source
- Each term in the above equation may saturate (>1.0). This can be handled in different ways:
  - clamp each term to [0,1]
  - compute the value for all the pixels in the image and adjust at the end by dividing by the maximum intensity value found.

Constant Shading

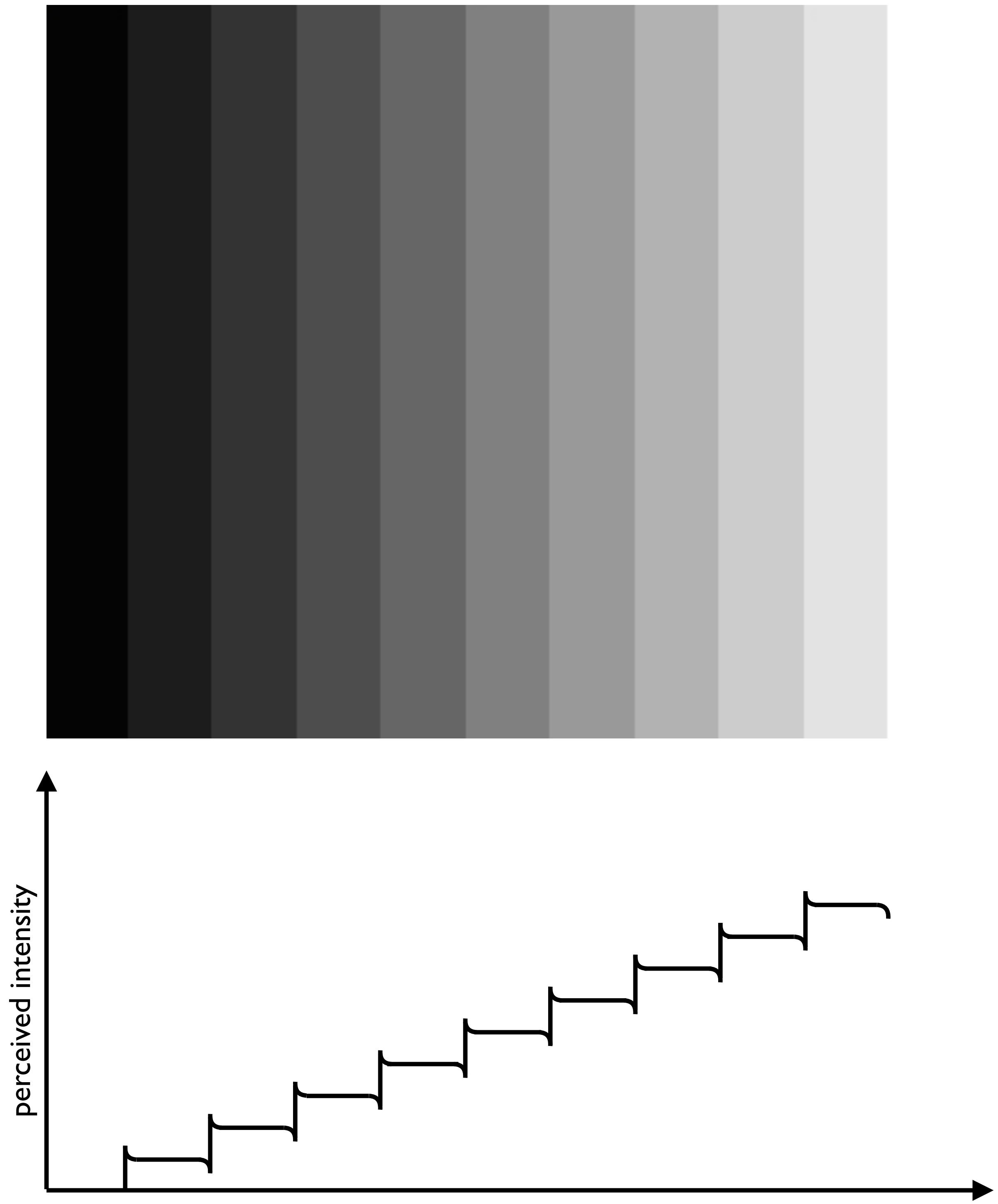
- Assumptions:
  - light sources are at an infinite distance (**L** is constant)
  - **N** is constant along the surface (surfaces are planar)
  - Viewer is at infinity (View vector **V** is constant)
- Illumination is performed once per face and the same colour is assigned to every pixel on that face
- If any of the assumptions is not valid, we need to choose a point to compute the colour (one of the vertices? centre of the polygon?)
- Mach banding at polygon edges!





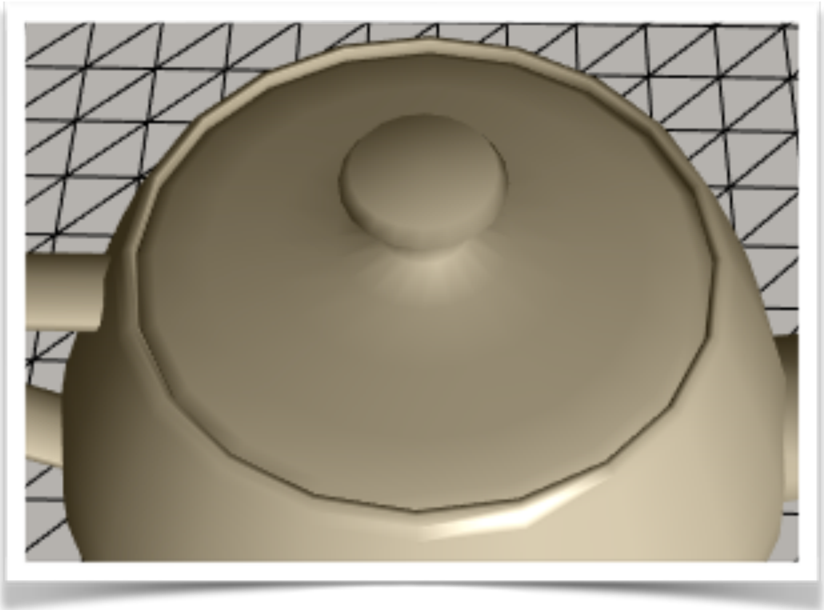
Mach Bands

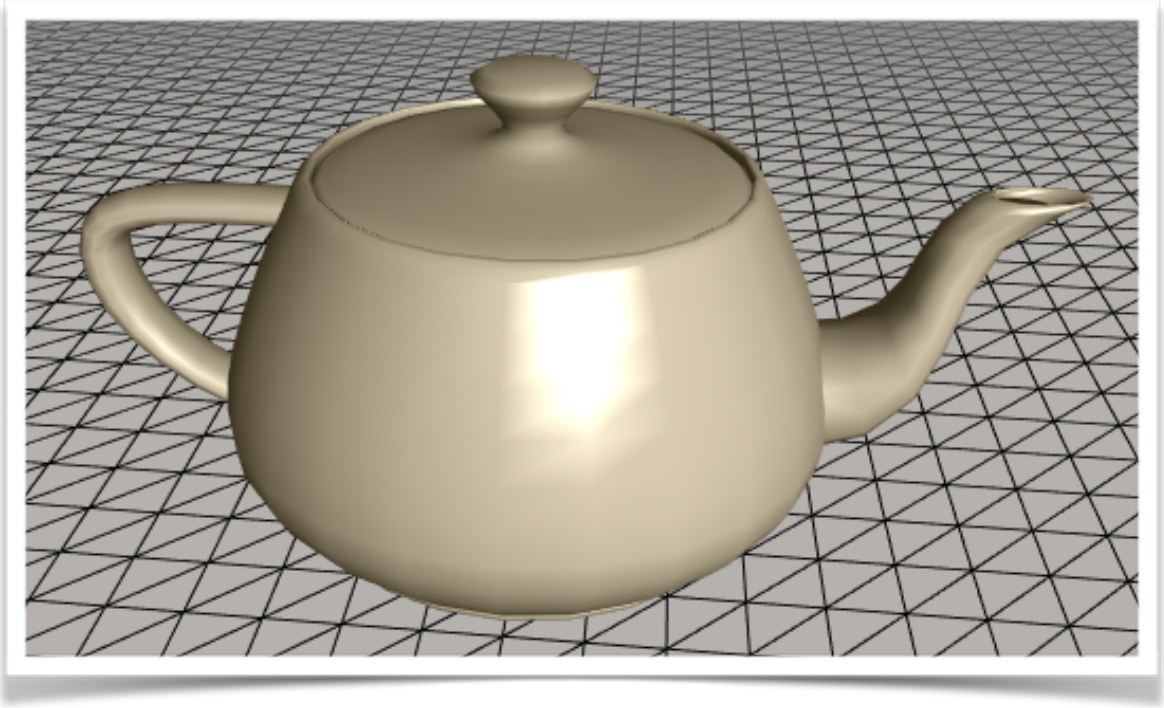
- Constant shading generates Mach banding
- intensities near darker colours appear lighter
- intensities near lighter colours appear darker
- Phenomena is created by lateral inhibition of photo sensors in the eye
- The surface appears to be a set of curved slices



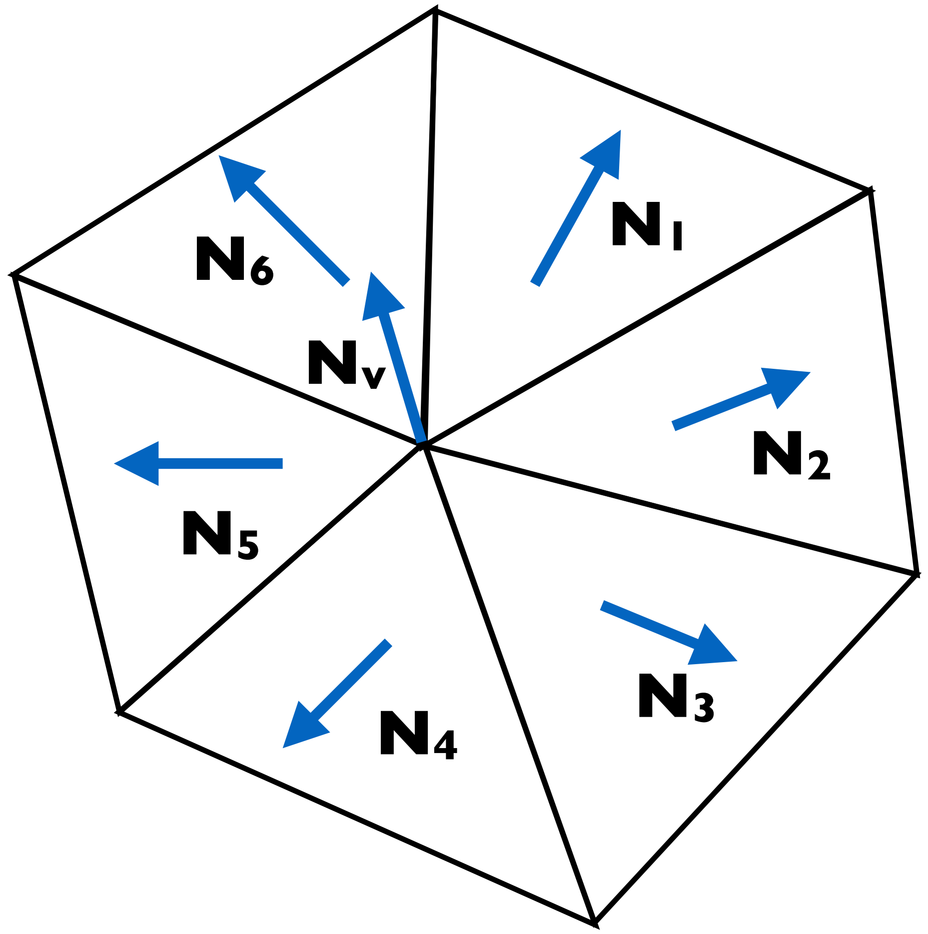
Gouraud Shading

- Gouraud shading eliminates discontinuities in colour by interpolation of the colour (or intensity) inside the polygon
- Normals at the vertices need to be known or possible to compute or estimate
- Evaluate the illumination model at each vertex and interpolate the colour inside the face
- Mach banding, although reduced, is still present (discontinuities on the rate of change of intensity)





Computing Vertex Normals

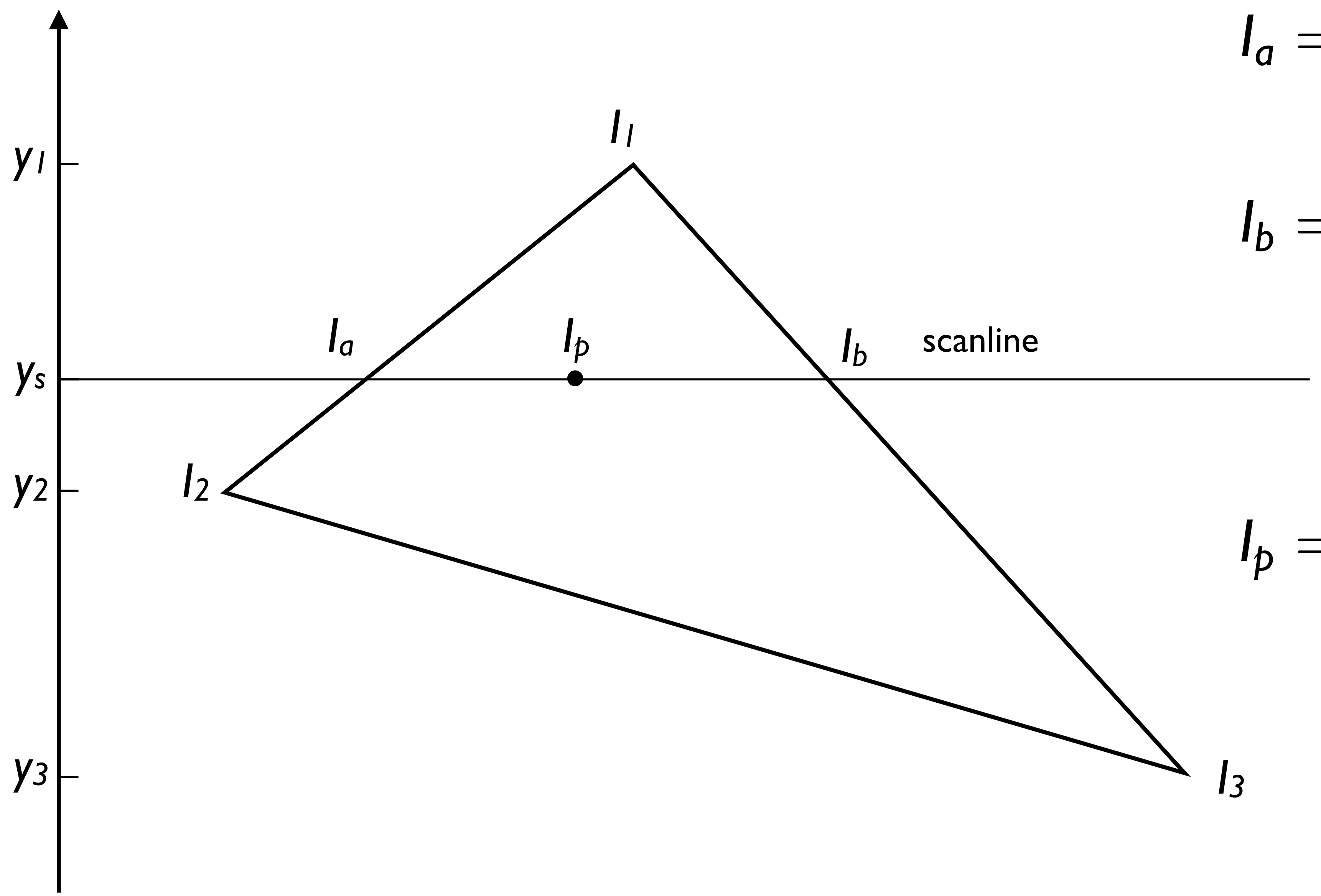


- If the normals at the vertices are not provided, they can be estimated as an average
- Sum all the normals (normalised beforehand) of the surrounding faces and normalise the result

$$\mathbf{N_v} = \frac{\sum_{i=1}^{N} \mathbf{N}_i}{\| \sum_{i=1}^{N} \mathbf{N}_i \|}$$

- If an edge is meant to be visible, normals have to be averaged separately on each side of the edge

Interpolating Colours/Intensities



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

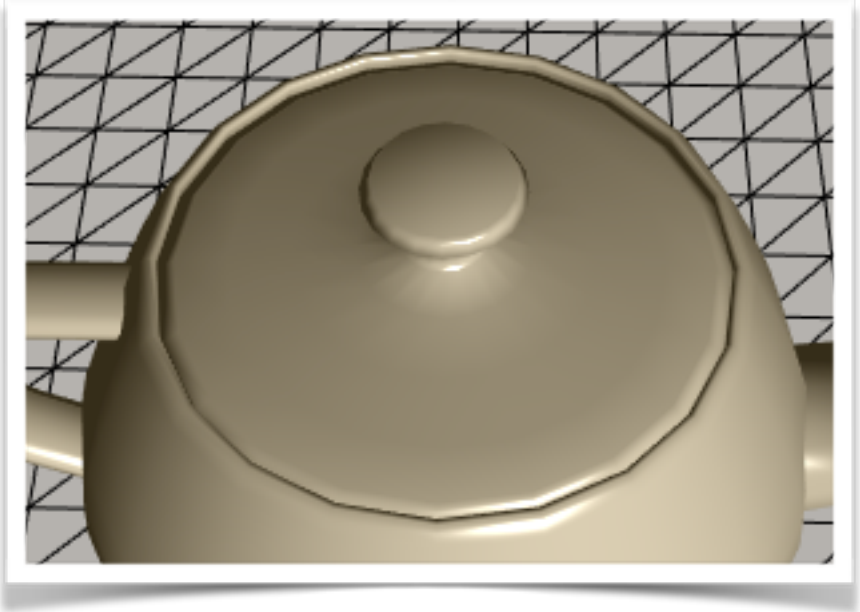
$$I_b = I_3 + (I_1 - I_3)\frac{y_s - y_3}{y_1 - y_3}$$

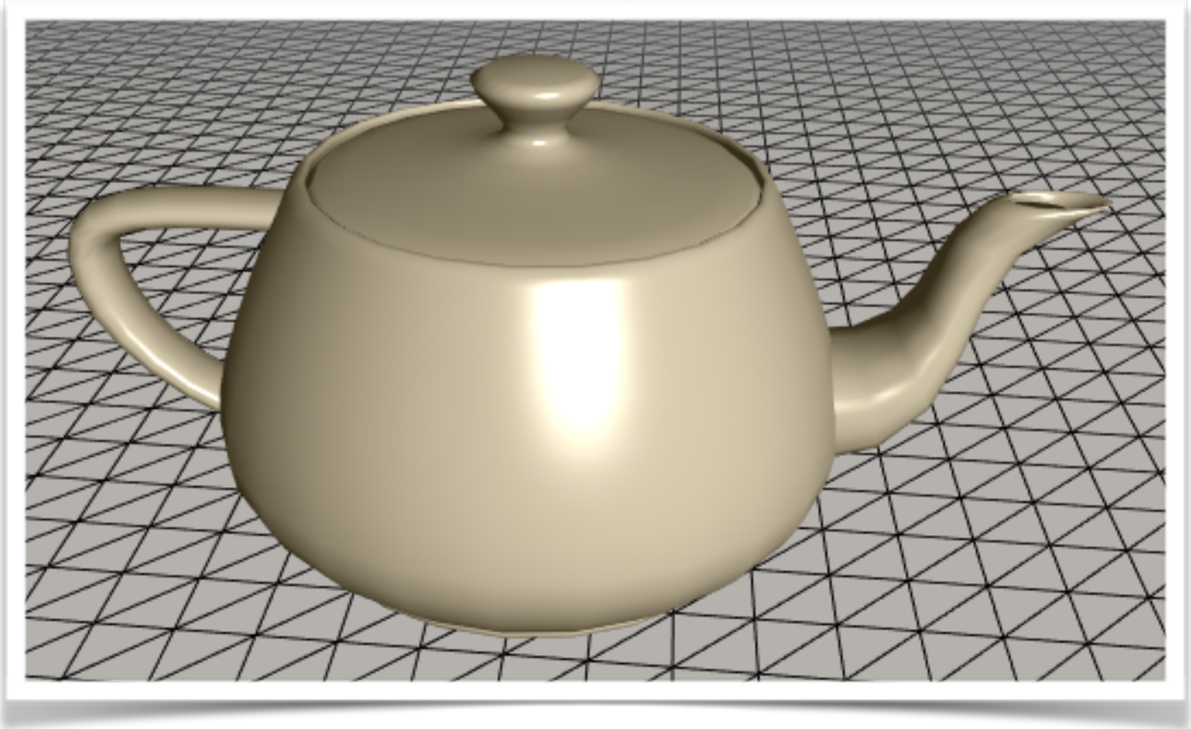
$$I_p = I_a + (I_b - I_a)\frac{x_p - x_a}{x_b - x_a}$$

Easily performed in hardware: similar to z coordinate interpolation for z-buffer HLHSR

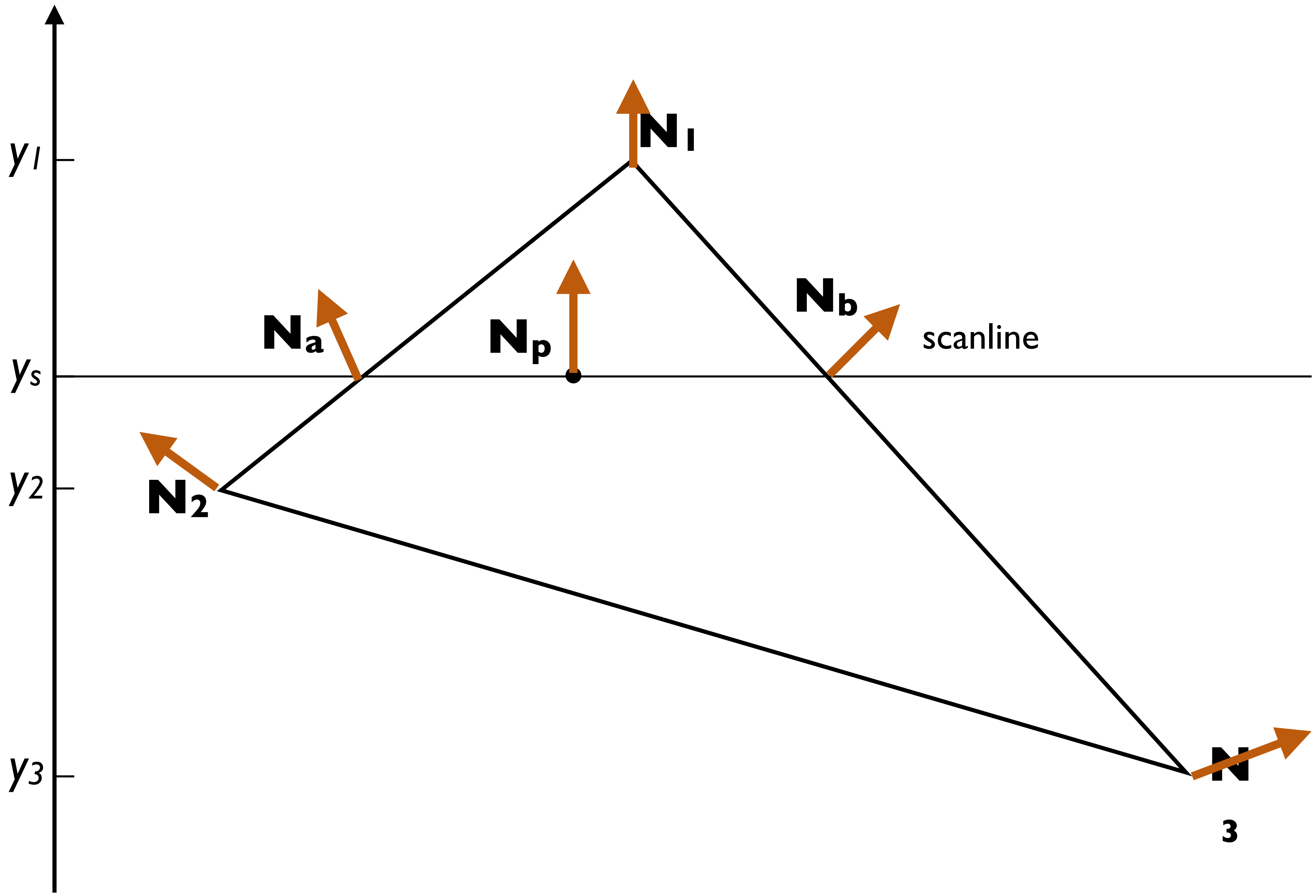
Phong Shading

- Assumptions:
  - The polygonal mesh is an approximation to the intended curved surface
  - Normals at the vertices are either defined or estimated from representative normals around the vertex
- Interpolate the surface normal inside each face instead of colour
- Evaluate the illumination model for each pixel using the interpolated normal





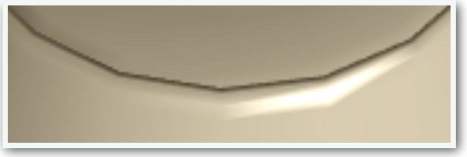
Interpolating Normals



- Interpolation can be done on a per component basis
- After interpolation, normal vector needs to be normalised
- Normal vector needs to be in WC or Eye coordinates to perform illumination

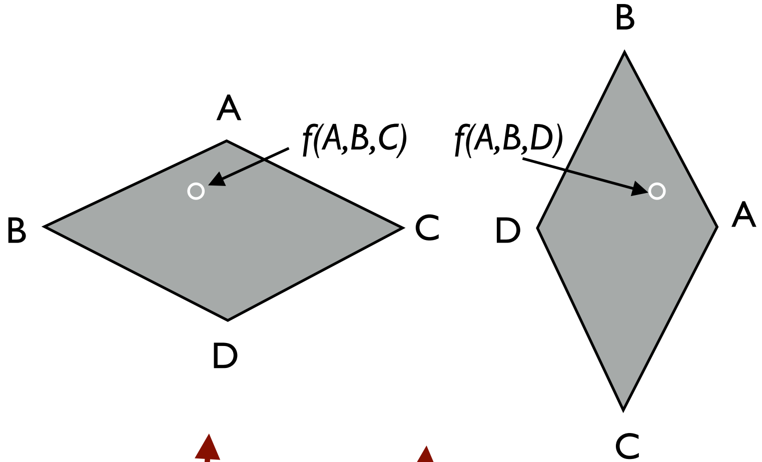
Limitations of Interpolated Shading

- Polygonal silhouette - The silhouette edge of the mesh is still clearly polygonal, contrasting with the curved appearance of its interior

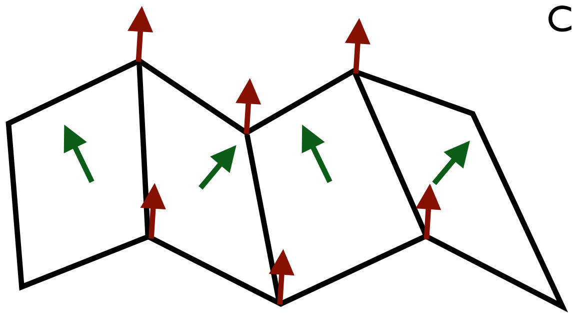


- Perspective distortion - Since interpolation is performed after perspective transformation, interpolation along consecutive scanlines near the observer should provide smaller increments than those farther away

- Orientation dependence - for general polygons, the interpolated value of a point inside it is dependent on the orientation of the polygon. This doesn't happen in the case of triangles



- Unrepresentative vertex normals - Computed vertex normals may not adequately represent the surface's geometry



Phong versus Gouraud (shiny materials)

specular reflection on vertex bleeds through the surface interior with Gouraud shading

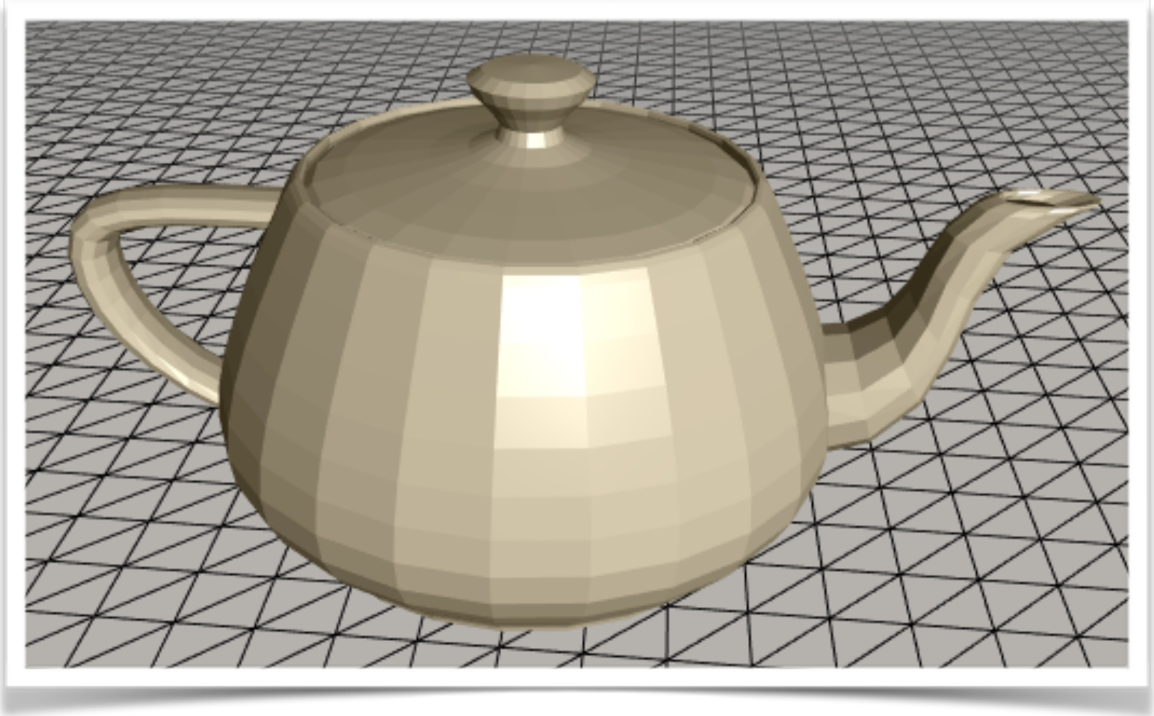
Highlight totally missed inside the polygon

Highlight correctly captured with Phong shading

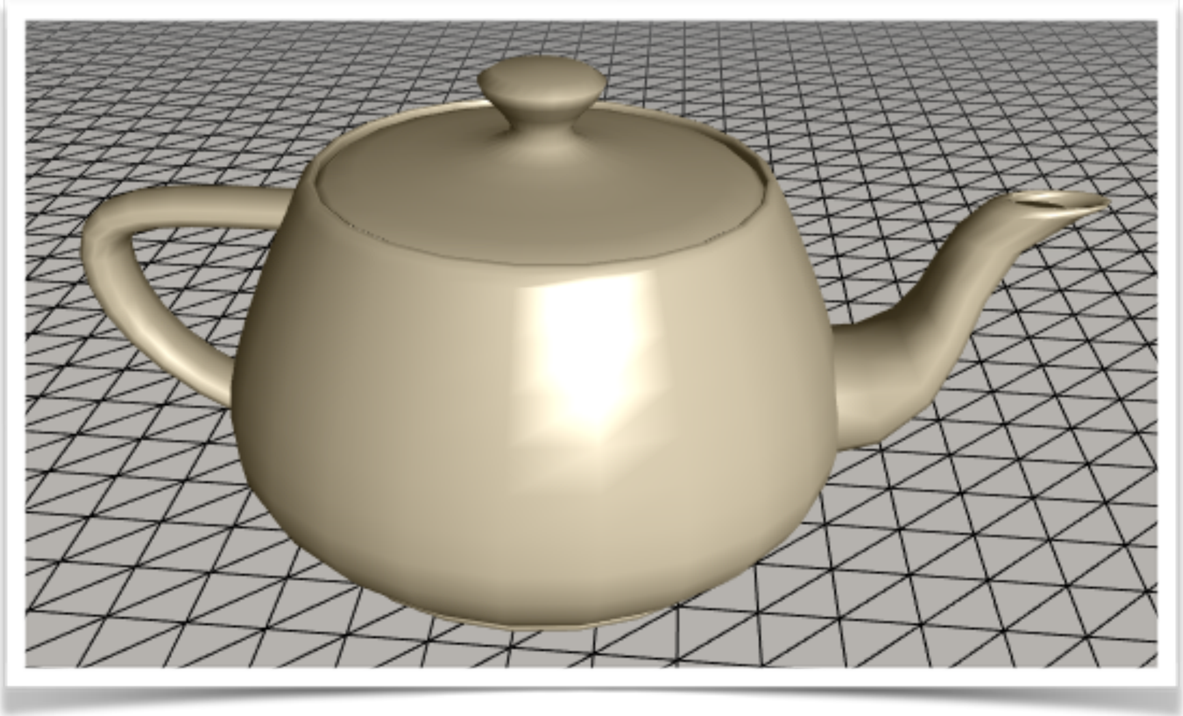
* fig. 16.21 from CGP&P 2nd ed.

specular reflection on vertex quickly falls off with Phong shading

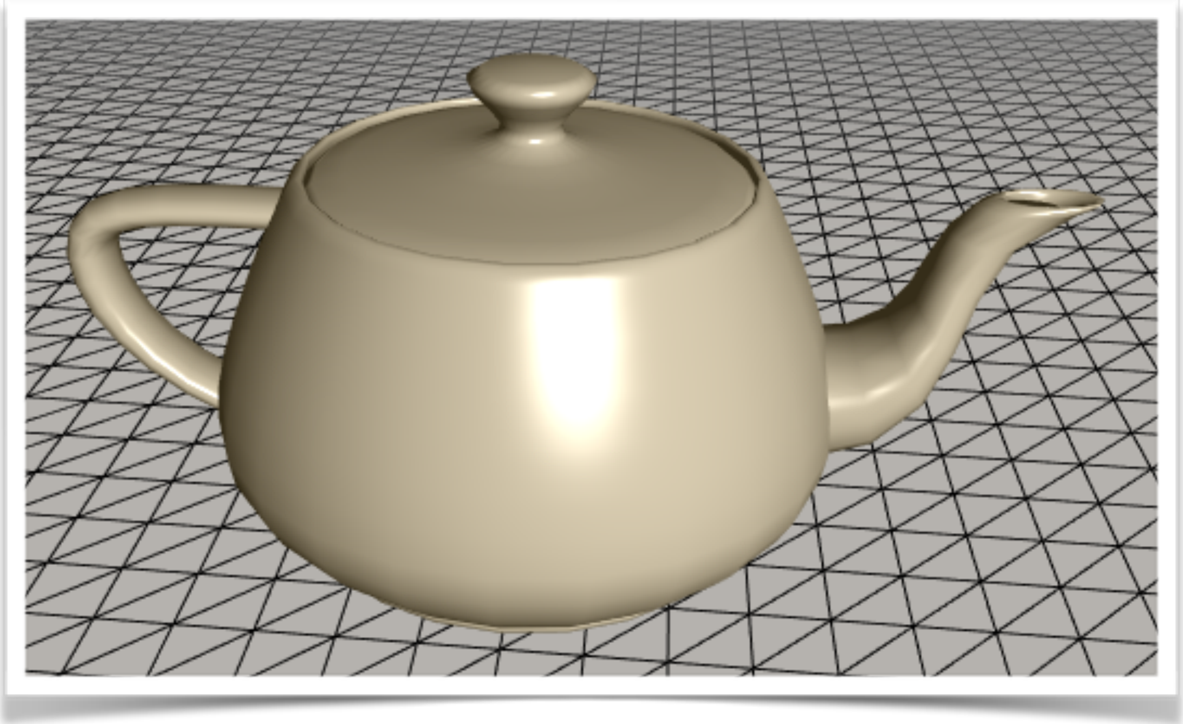
Shading Methods



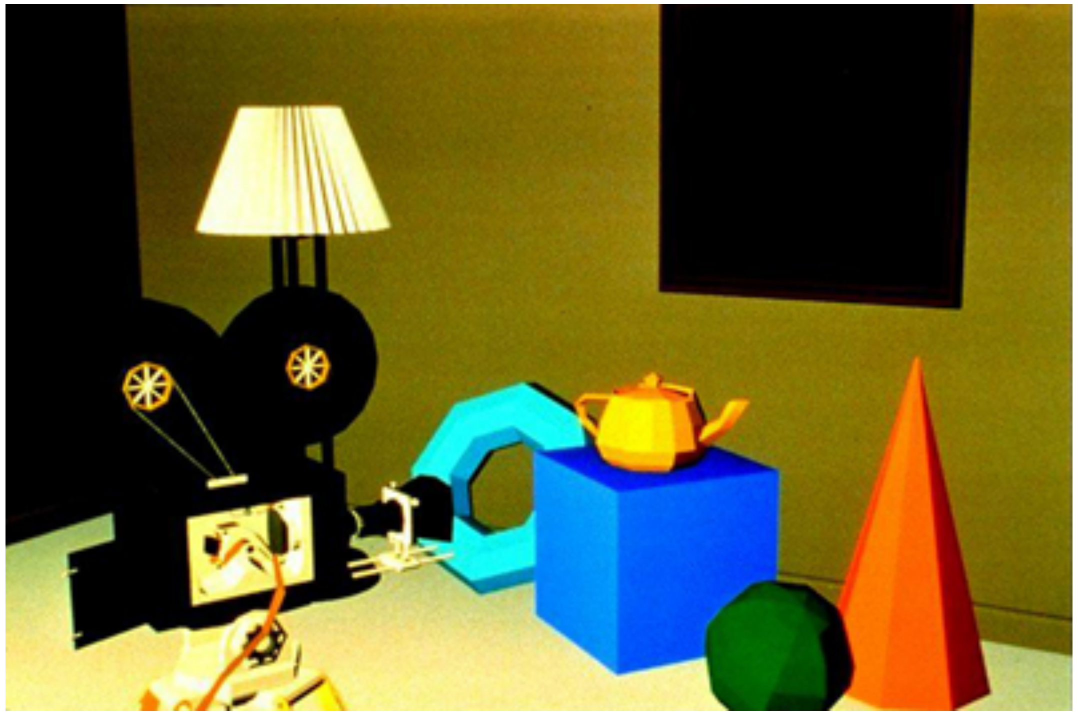
flat shading



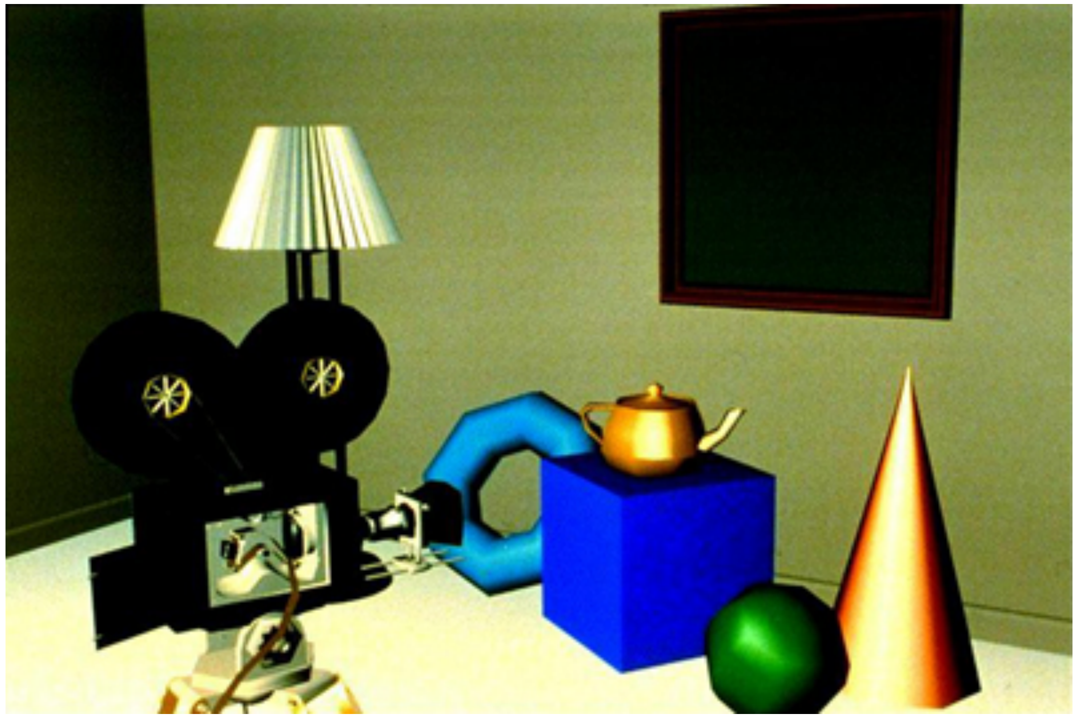
Gouraud shading



Phong shading



flat shading



Gouraud shading

Phong shading