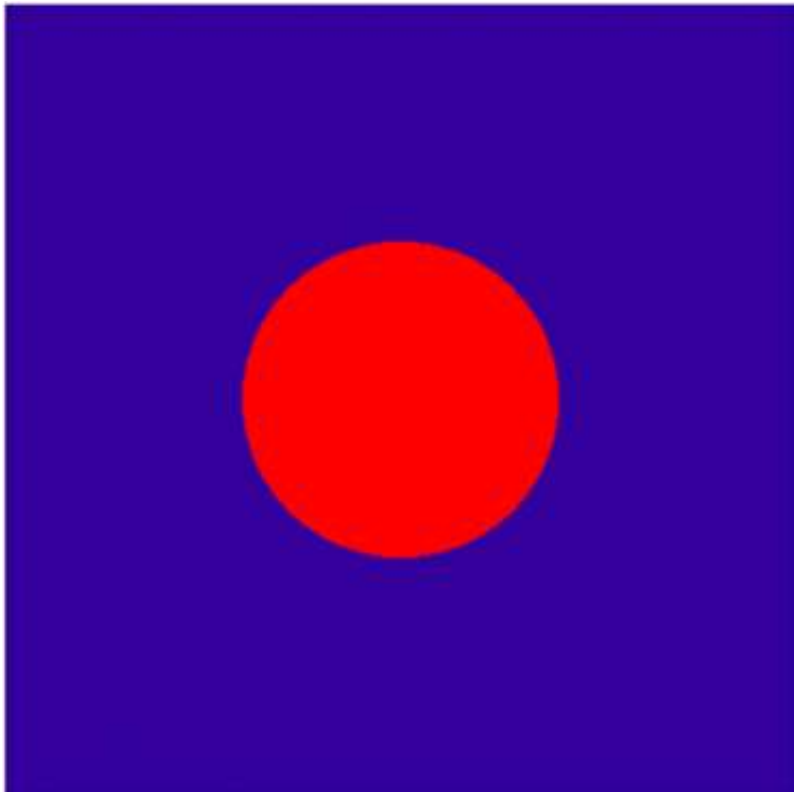
Illumination and Surface Rendering

Shmuel Wimer

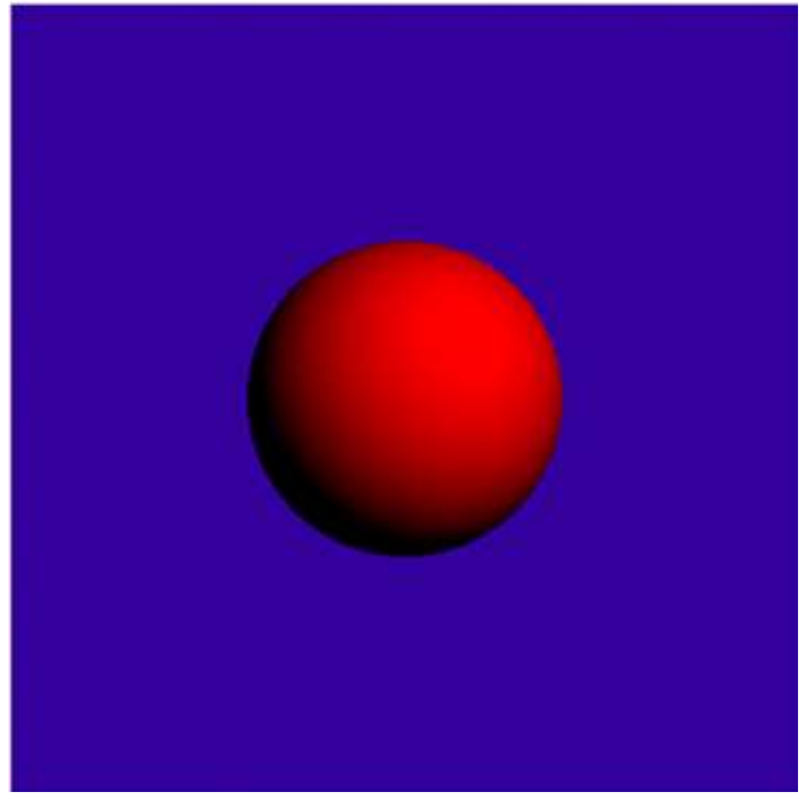
Bar Ilan Univ., Eng. Faculty

Technion, EE Faculty

Why Lighting?

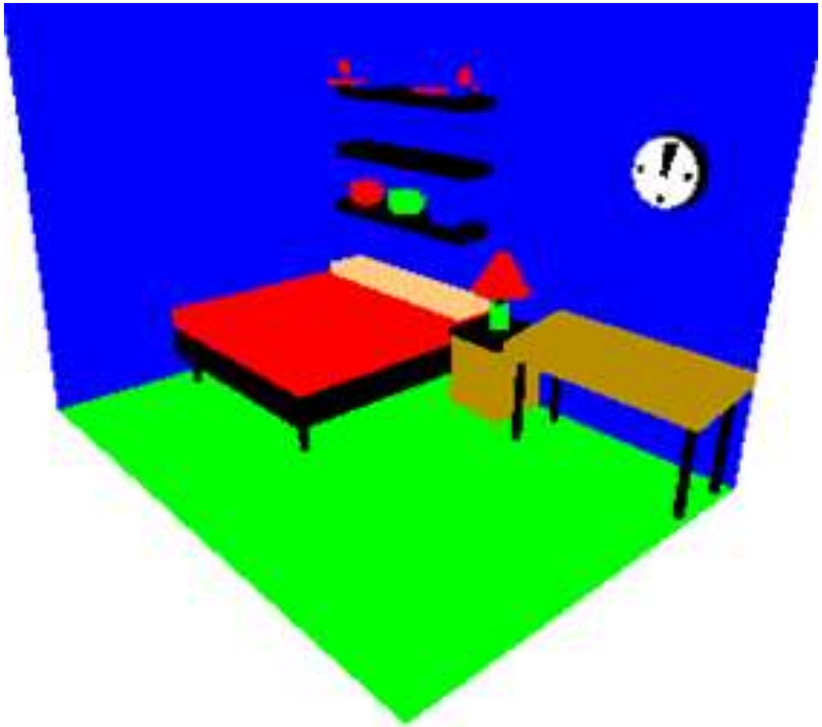


3D without lighting

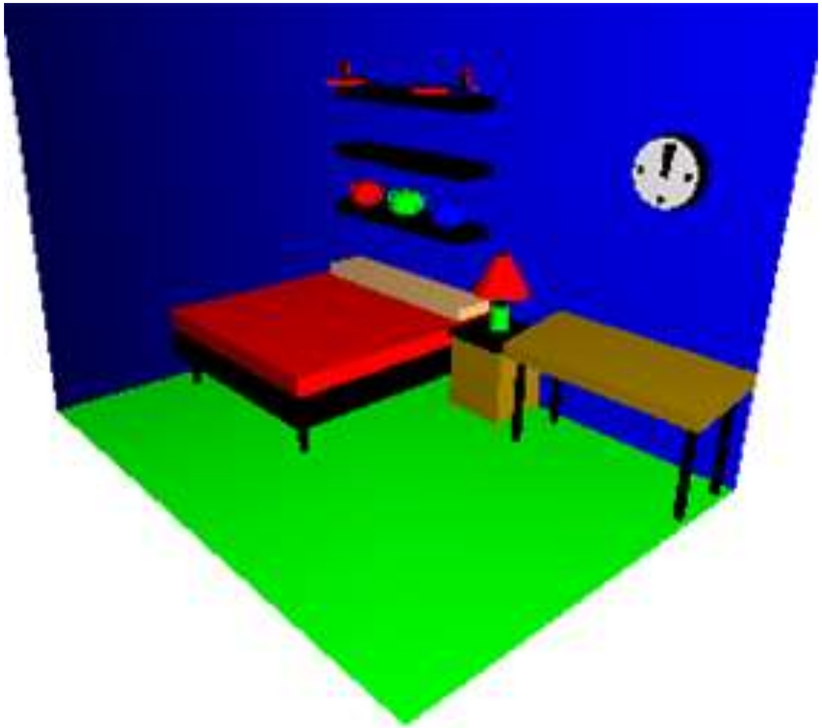


3D with lighting

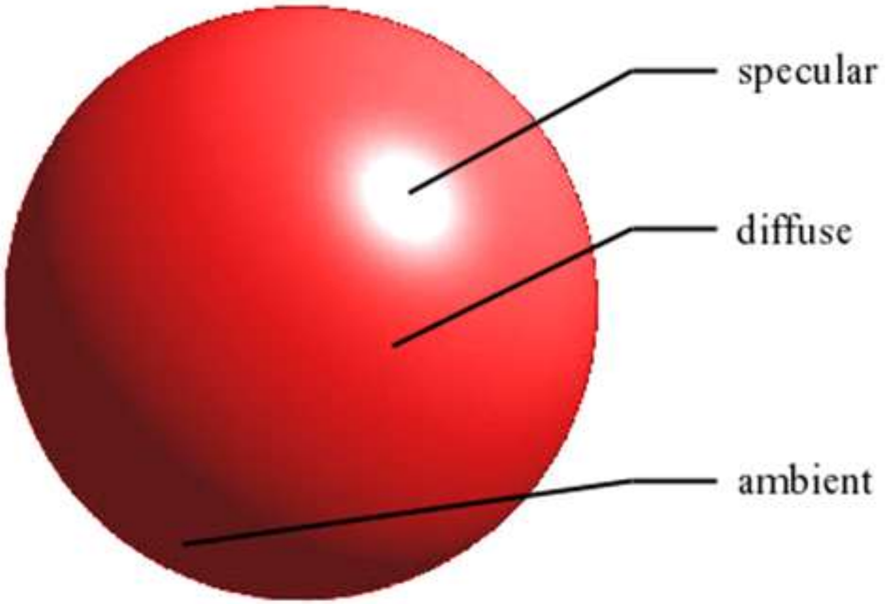
3D without lighting



3D with lighting



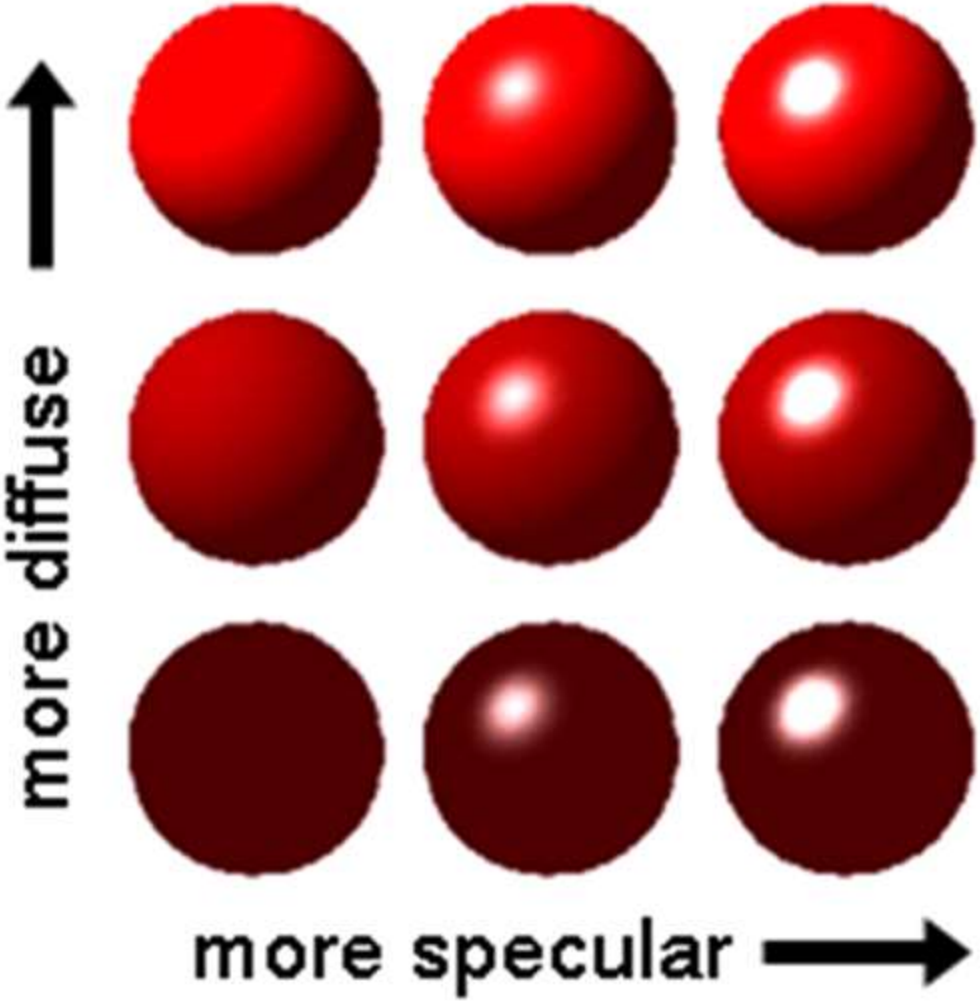
Components of Reflections



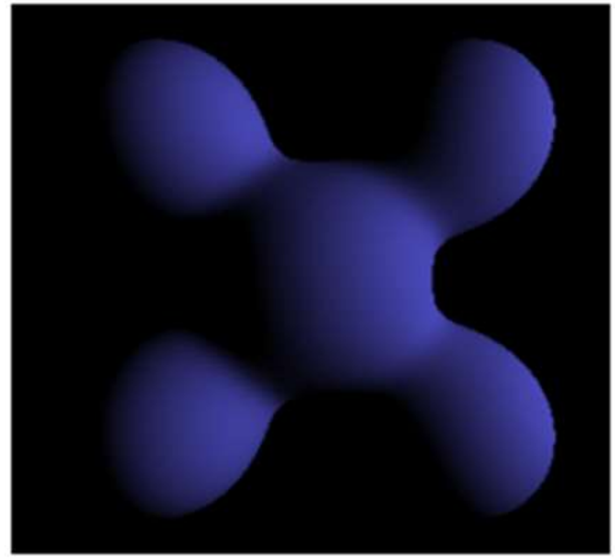
Ambient – surface exposed to **indirect light** reflected from nearby objects.

Diffuse – reflection from **incident** light with equal intensity in all directions. Depends on surface properties.

Specular – near **total** of the incident light around reflection angle.

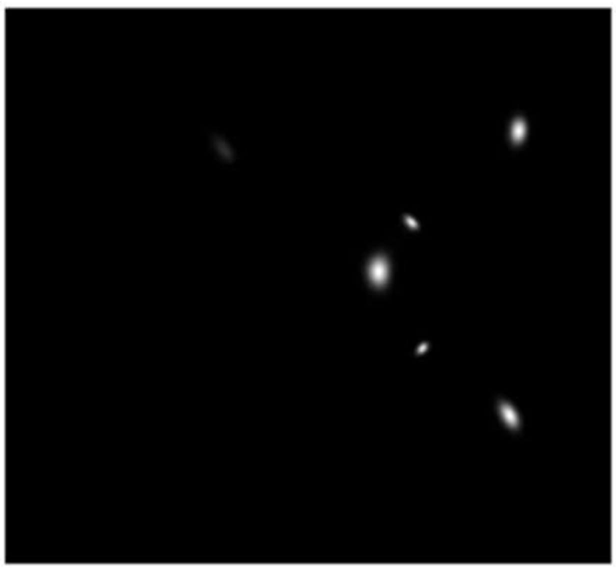

more diffuse
more specular

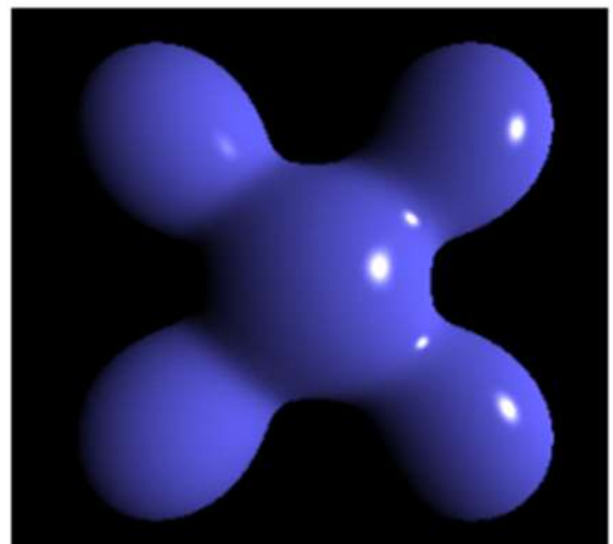
ambient



diffuse

specular

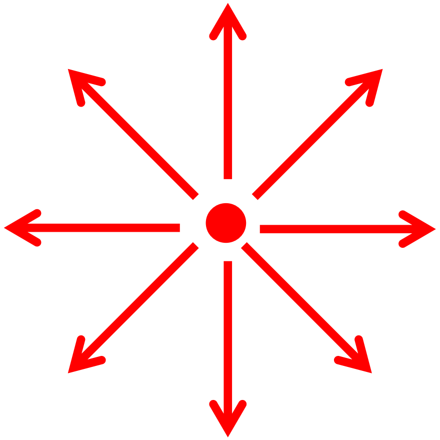




final

Light Sources

Radial intensity attenuation is $1/d^2$

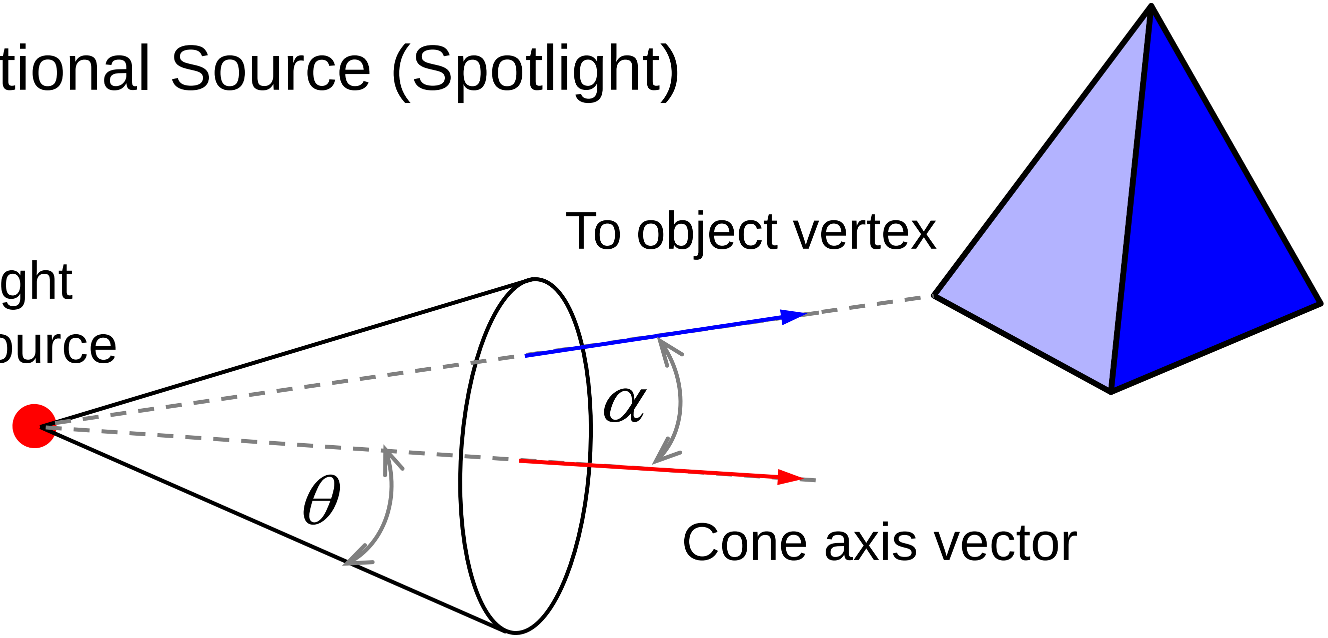


This is a problem in practice. $1/d^2$ produces too much intensity variations for objects very close to source. Realistic sources are not infinitesimal small. The remedy is in adding a linear term.

Another problem occurs when the source is at infinity. Light rays are nearly parallel but intensity decays quadratic and we still wish to display the object

$$f_{\substack{\text{radial}\\ \text{attenuation}}}(d) = \begin{cases} 1.0, & \text{source at infinity} \\ \dfrac{1}{a_0 + a_1 d + a_2 d^2}, & \text{local source} \end{cases} .$$

Directional Source (Spotlight)



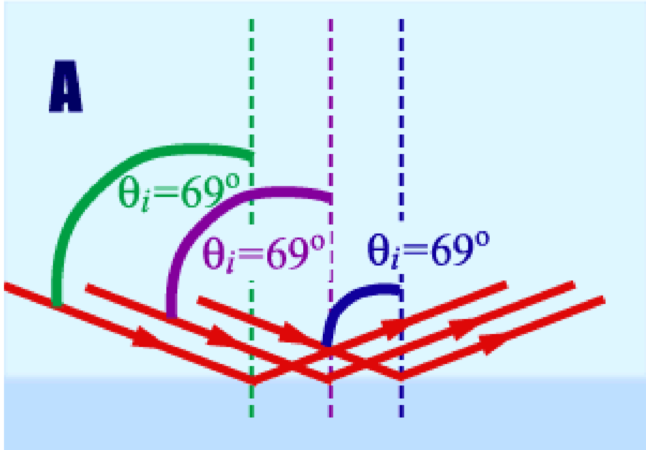
Points outside the cone are not illuminated (stay in dark).

Angular attenuation is:

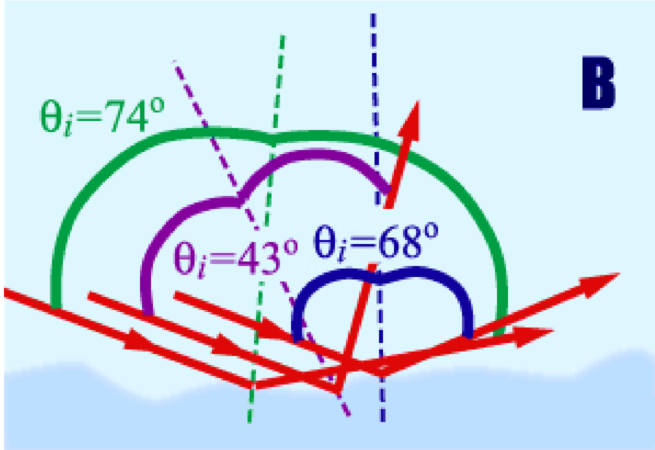
$$f_{\substack{\text{angular}\\ \text{attenuation}}}(\alpha)=\begin{cases} 1.0, & \text{source not a spotlight}\\ 0.0, & \mathbf{V}_{\text{obj}}\cdot\mathbf{V}_{\text{light}}=\cos\alpha<\cos\theta \\ \left(\mathbf{V}_{\text{obj}}\cdot\mathbf{V}_{\text{light}}\right)^{a}, & \text{otherwise}\end{cases}$$

Illumination Models

Surface lighting models compute the outcome of the interactions between incident radiant energy and the material composition of the object.



When the surface is smooth reflections are similar at all point. This is called ***total reflection***.



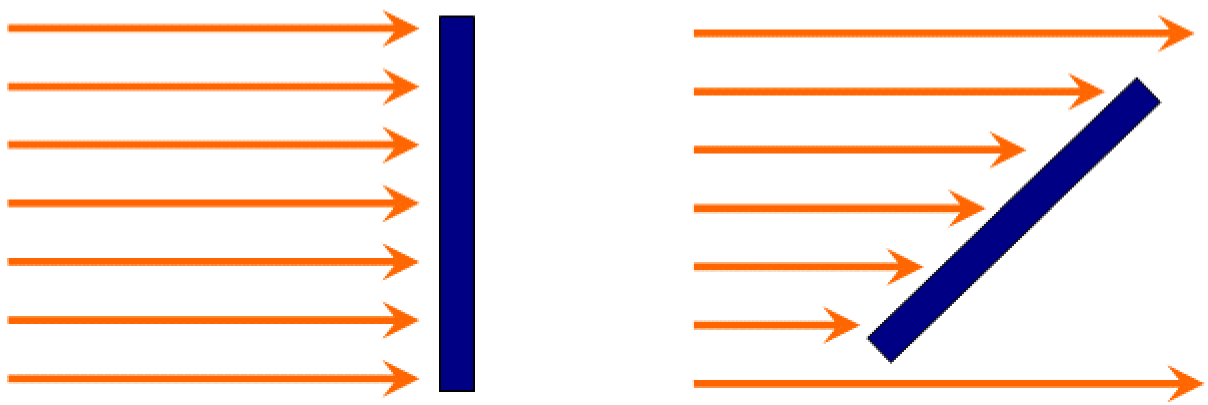
If the surface is rough. Reflected light is scattered in all directions. This is called ***diffuse reflection***.

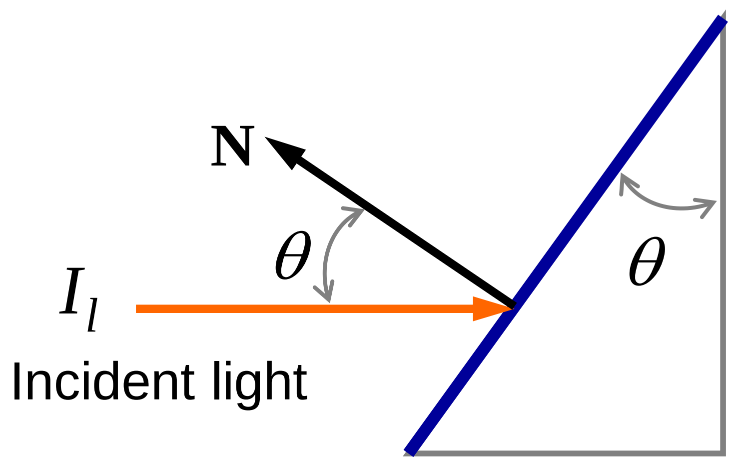
A basic illumination is obtained by an ***ambient light*** which determines the brightness of the scene. It is uniform for all objects, independent of the viewing point and direction. The amount of the reflected light (***diffuse reflection***) depends however on the surface optical properties.

If the ambient light of the scene is I_a and k_a is surfce ambient reflection parameter, the contribution to surface reflection at any point is $I_{\text{ambient}} = k_a I_a$.

Ambient light alone is not enough to render a scene and light sources are added.

The amount of incident light depends on the orientation of the surface relative to the light source direction.





For a light source I_l and surface reflection parameter k_d (independent of ambient relection parameter k_a), the diffused reflection is $I_{l,\text{diffuse}} = k_d I_l \cos\theta$.

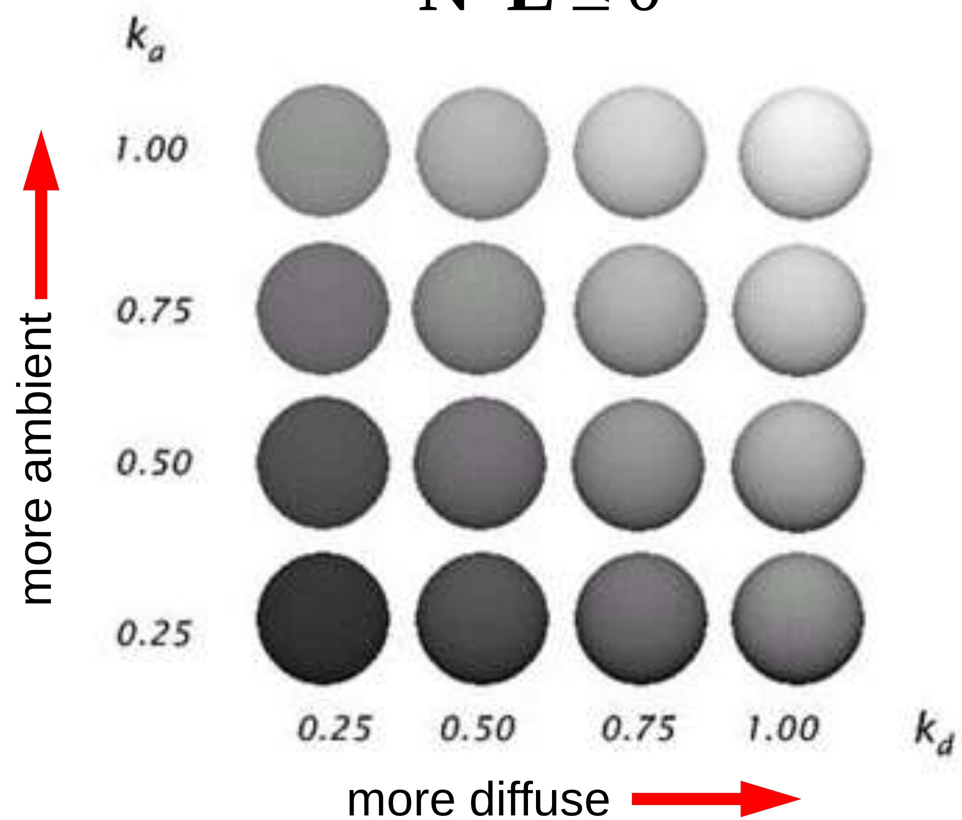
The unit direction vector $\mathbf{L}$ of a nearly point light source I_l incident with a surface is given by

$\mathbf{L} = \dfrac{\mathbf{P}_{\text{source}} - \mathbf{P}_{\text{surface}}}{|\mathbf{P}_{\text{source}} - \mathbf{P}_{\text{surface}}|}$. The difused reflection is

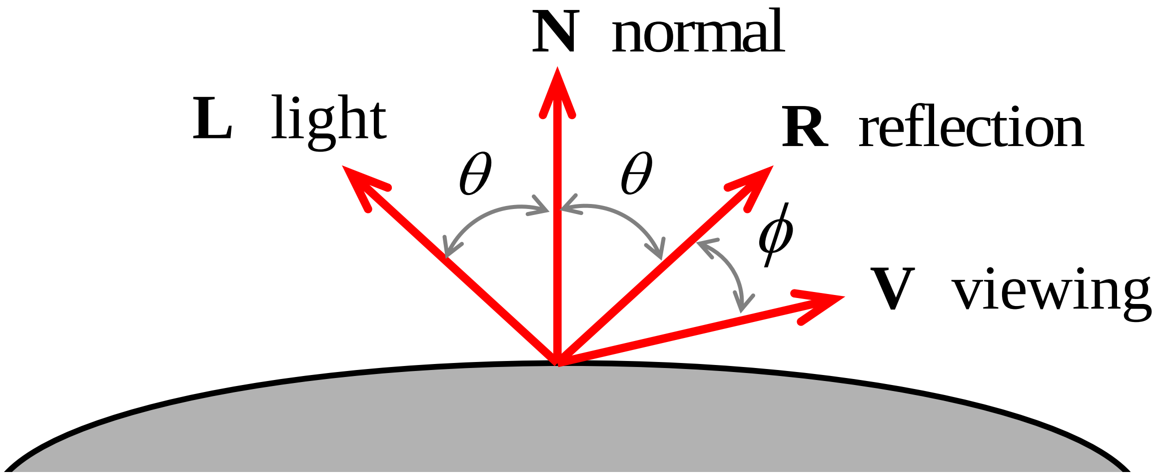
$$I_{l,\text{diffuse}} = \begin{cases} k_d I_l \left(\mathbf{N}\cdot\mathbf{L}\right), & \mathbf{N}\cdot\mathbf{L} > 0 \\ 0.0, & \mathbf{N}\cdot\mathbf{L} \le 0 \end{cases}.$$

Combining ambient and a single point source, the total diffuse reflection is

$$I_{\text{diffuse}} = \begin{cases} k_a I_a + k_d I_l \left(\mathbf{N} \cdot \mathbf{L}\right), & \mathbf{N} \cdot \mathbf{L} > 0 \\ k_a I_a, & \mathbf{N} \cdot \mathbf{L} \le 0 \end{cases}.$$

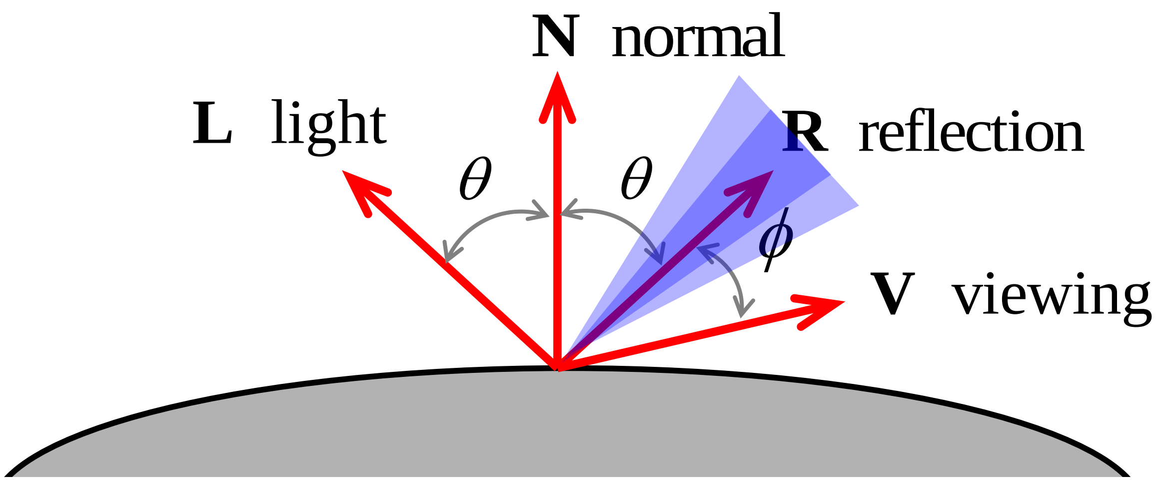


Specular Reflection and the Phong Model



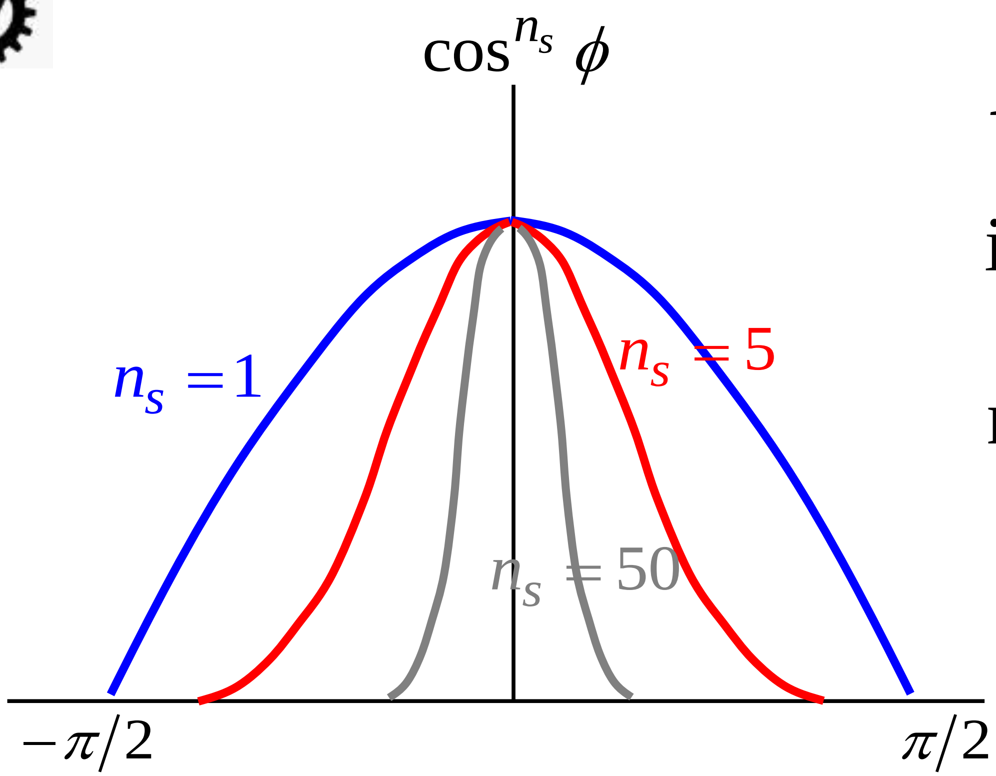
The bright spot (***specular reflection***) shown on a shiny surfaces is the outcome of total reflection of the incident light in a concentrated region around the ***specular-reflection angle***.

Specular Reflection and the Phong Model



A shiny surface has a narrow specular reflection range.

A dull (rough) surface has a wide specular reflection range.

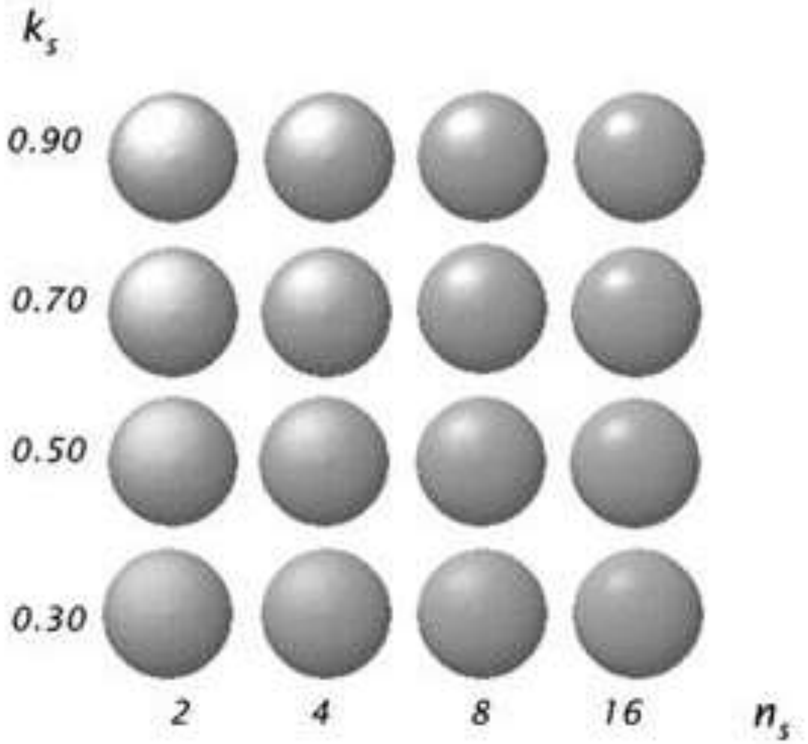


Phong model sets the intensity of specular reflection to $\cos^{n_s}\phi$.

$$I_{l,\text{specular}} = W(\theta) I_l \cos^{n_s}\phi.$$

$0 \le W(\theta) \le 1$ is called ***specular - reflelection coeficient***.

If light direction ***L*** and viewing direction ***V*** are on the same side of the normal ***N***, or if ***L*** is behind the surface, specular effects do not exist.



For most opaque materials specular-reflection coefficient is nearly constant k_s.

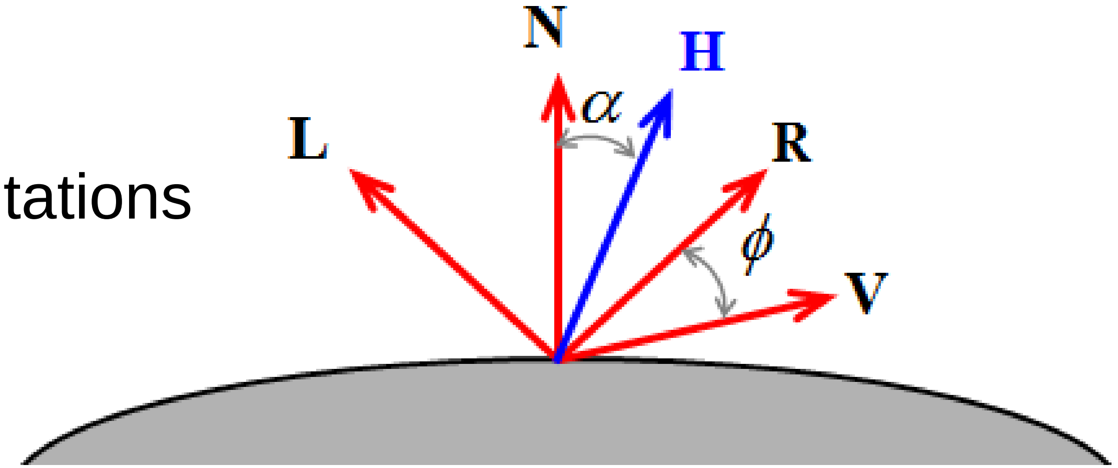
$$I_{l,\text{specular}} = \begin{cases} k_s I_l (\mathbf{V}\cdot\mathbf{R})^{n_s}, & \mathbf{V}\cdot\mathbf{R}>0 \text{ and } \mathbf{N}\cdot\mathbf{L}>0 \\ 0.0, & \text{otherwise} \end{cases}.$$

$\mathbf{R}$ can be calculated from $\mathbf{L}$ and $\mathbf{N}$, $\mathbf{R} = (2\mathbf{N}\cdot\mathbf{L})\mathbf{N} - \mathbf{L}$.

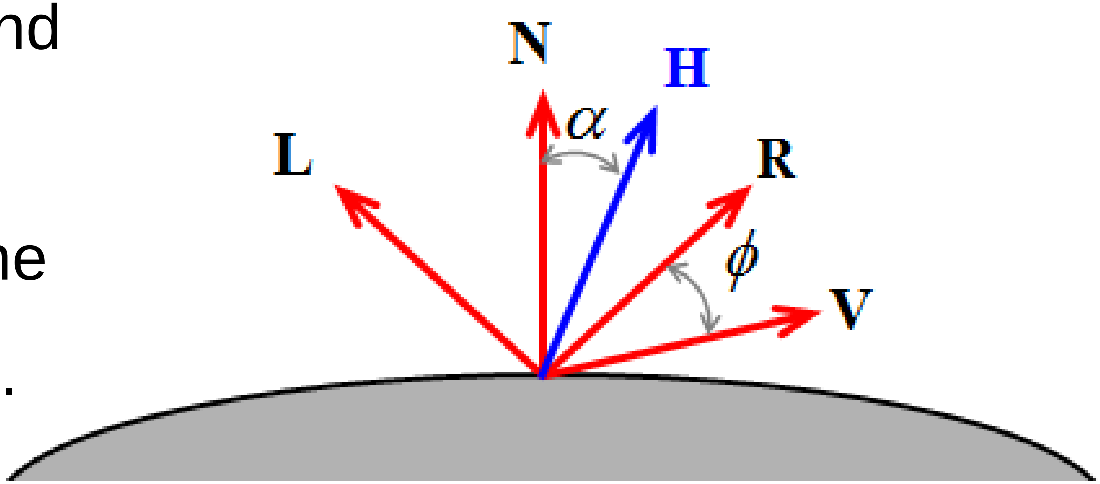
The normal ***N*** may vary at each point. To avoid ***N*** computations, angle *Φ* is replaced by an angle *α* defined by a halfway vector ***H*** between ***L*** and ***V***.

Efficient computations

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



If the light source and the viewer are relatively far from the object *α* is constant.



H is the direction yielding maximum specular reflection in the viewing direction ***V*** if the surface normal ***N*** would coincide with ***H***. If ***V*** is coplanar with ***R*** and ***L*** (and hence with ***N*** too) $\alpha = \Phi/2$.

Combining Everything Together

For a single point source:

$$I = I_{\text{diffuse}} + I_{\text{specular}} = k_a I_a + k_d I_l \left(\mathbf{N}\cdot\mathbf{L}_l\right) + k_s I_l \left(\mathbf{V}\cdot\mathbf{R}\right)^{n_s}$$

For multiple point sources:

$$I = k_a I_a + \sum_{l=1}^{n} I_l \left[k_d \left(\mathbf{N}\cdot\mathbf{L}_l\right) + k_s \left(\mathbf{V}\cdot\mathbf{R}\right)^{n_s} \right]$$

Accounting radial and angular attenuations:

$$I = k_a I_a + \sum_{l=1}^{n} I_l \, f_{\substack{l,\text{radial}\\ \text{attenuation}}} \, f_{\substack{l,\text{angular}\\ \text{attenuation}}} \left[k_d \left(\mathbf{N}\cdot\mathbf{L}_l\right) + k_s \left(\mathbf{V}\cdot\mathbf{R}\right)^{n_s} \right]$$

How Colors Get In?

Each light source has red, green and blue (RGB) components.

$$I_l = \left(I_{lR}, I_{lG}, I_{lB}\right)$$

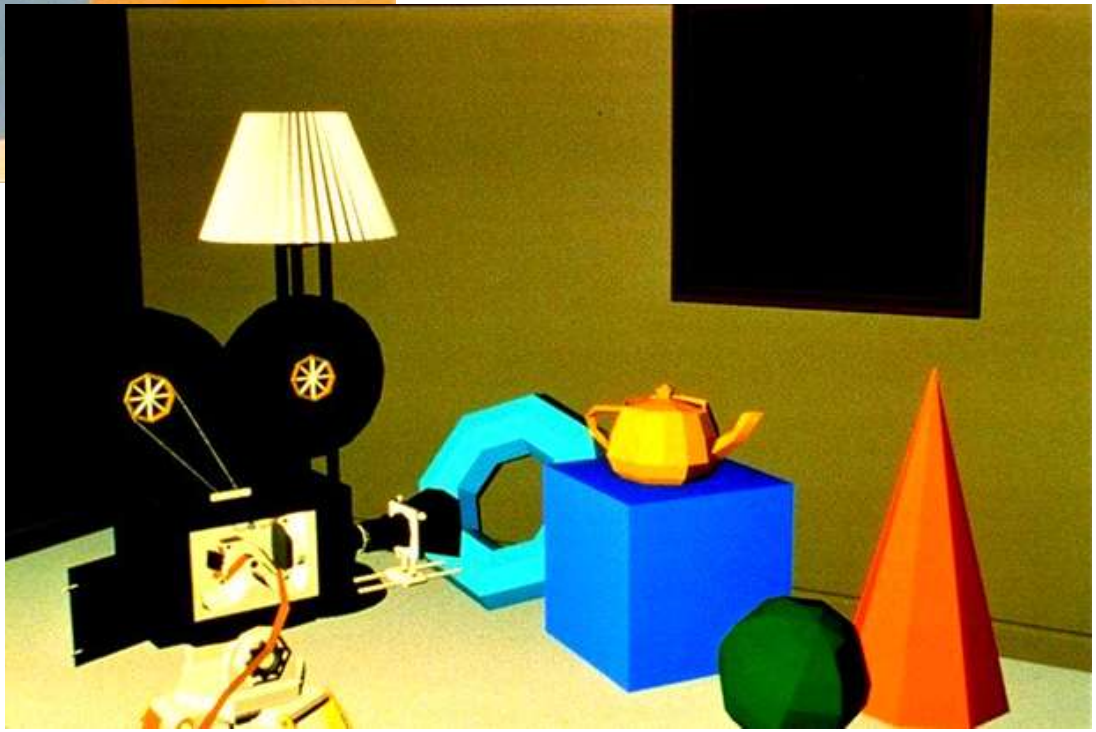
Similarly, the ambient, diffuse and specular coefficients of each surface has red, green and blue (RGB) components.

$$k_a = \left(k_{aR}, k_{aG}, k_{aB}\right) \; k_d = \left(k_{dR}, k_{dG}, k_{dB}\right) \; k_s = \left(k_{sR}, k_{sG}, k_{sB}\right)$$

Each components of the color is then calculated separately. All final components are then combined to set the final color (RGB) of a point.

Rendering Reminder

3D drawing without rendering results 2D image.



3D drawing with rendering obtains depth and perspective impression.

Polygon Rendering Methods

Illumination model can be applied at every projected pixel. Very expensive.

Intensity can be calculated at few locations of the surface and then approximated at other locations

Usually only surface approximation by polygons and scan-line rendering are supported. Color intensities are calculated at vertices and then interpolated for the rest points.

The simplest is to use the color of polygon vertex or centroid for the rest points. This is called ***flat surface rendering***.

Flat surface rendering provides accurate display if **<u>all</u>** the following assumptions are satisfied:

- Displayed object is polyhedron.
- All light sources are sufficiently far from object, so **N·L** attenuation is constant over entire surface.
- Viewing position is sufficiently far so **V·R** is constant over all surface.

Problem: abrupt color change across edges.

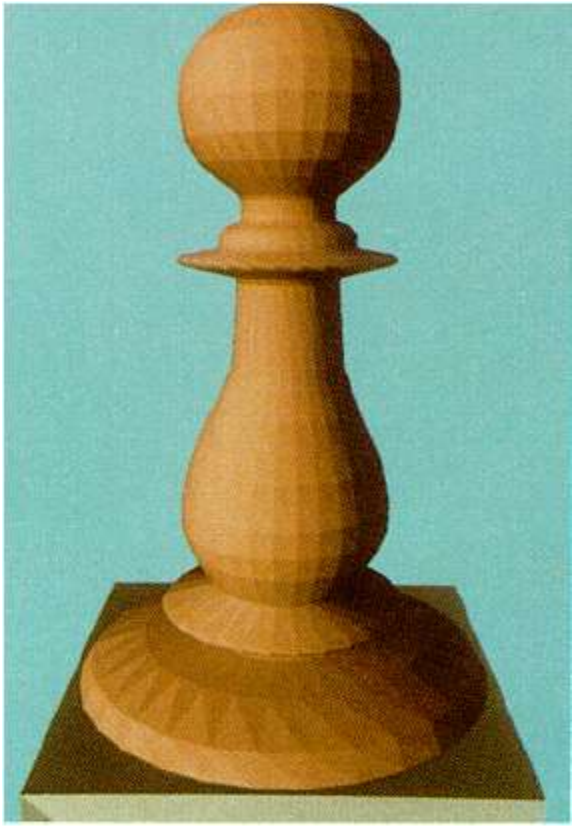
Gouraud Surface Rendering

Gouraud surface rendering, called also ***intensity-interpolation surface rendering***, provides smooth color transitions across edges of polygon surfaces. It eliminates the intensity discontinuities occurring in flat rendering.

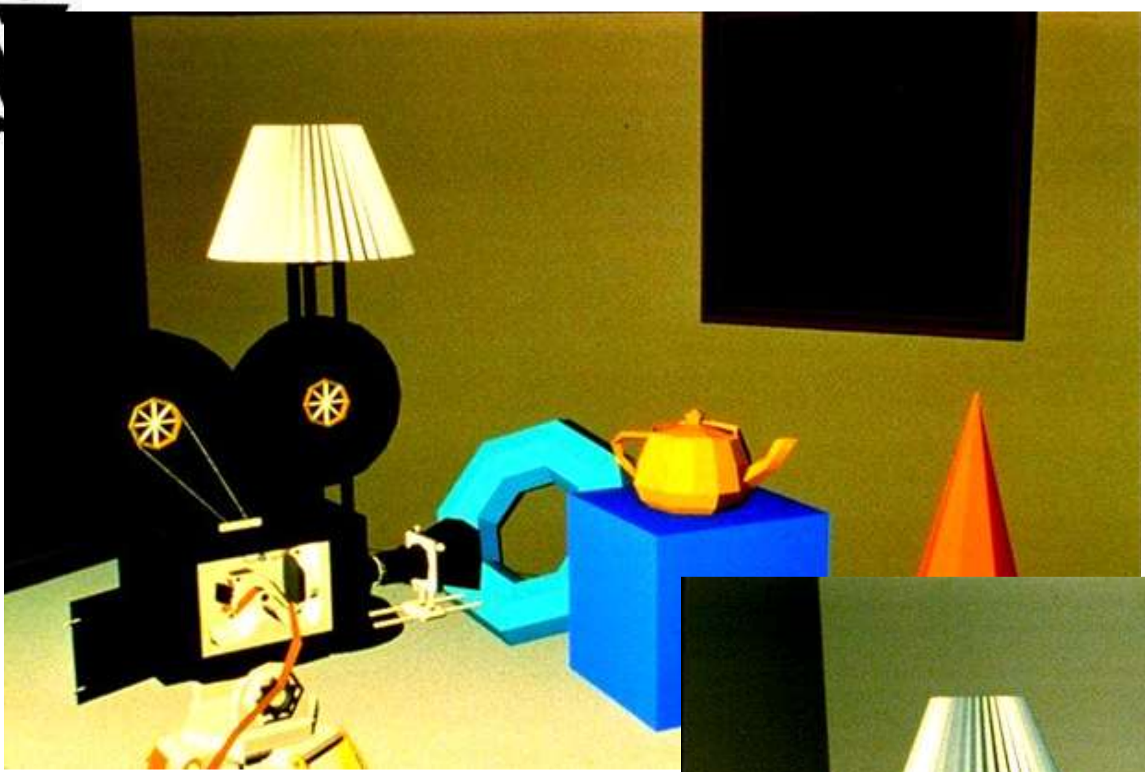
- Determine the average unit vector at each polygon vertex.
- Apply illumination model at each vertex to obtain color intensities at that position.
- Linearly interpolate intensities over projected areas.

Flat rendering

Gouraud rendering

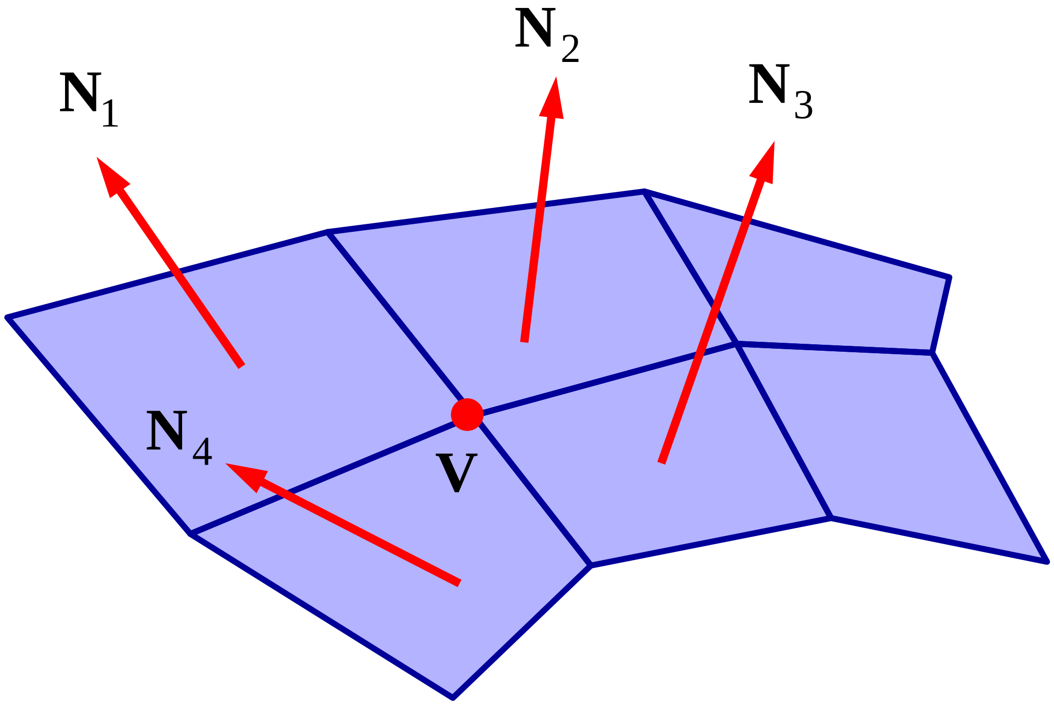

Flat rendering


Gouraud rendering



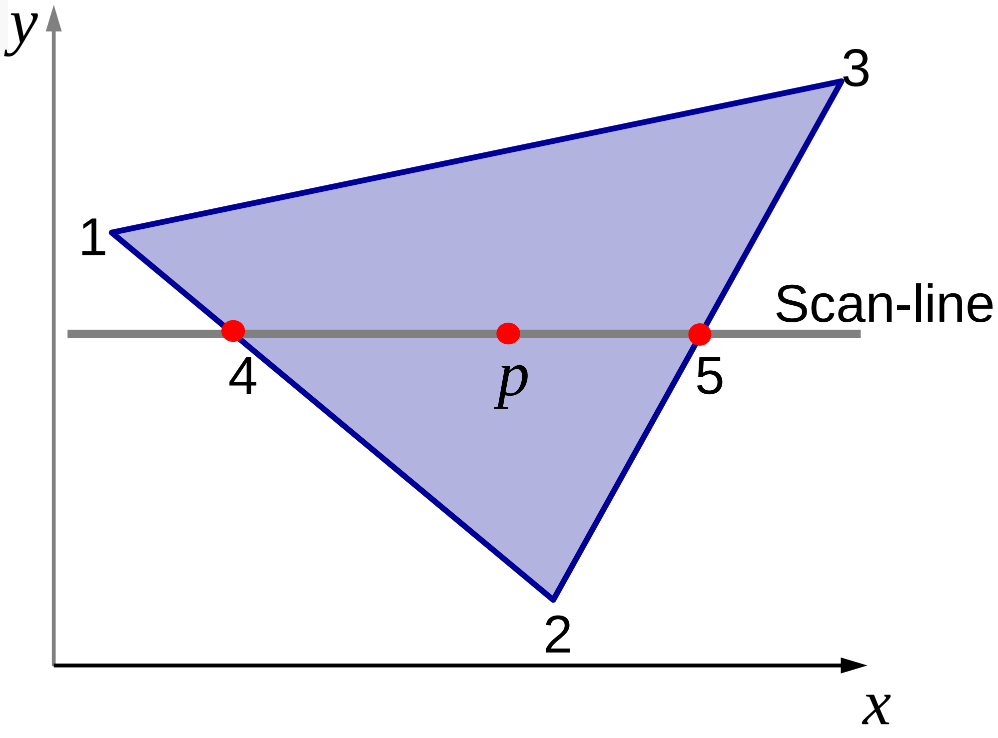
Flat rendering

Gouraud rendering



The normal vector at the vertex **V** is the average of the polygon normal vectors sharing that point.

$$\mathbf{N}_{\mathbf{V}} = \sum_{k=1}^{n} \mathbf{N}_k \Big/ \left| \sum_{k=1}^{n} \mathbf{N}_k \right|$$



Ends of a scan-line are interpolated from vertices.

$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

$$I_5 = \frac{y_5 - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_5}{y_3 - y_2} I_2$$

Internal point is interpolated from end points.

$$I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$

Avoid most divisions since progression of *y* is to *y-1* and progression of *x* is to *x+1*. Same methods as used in basic drawing are applicable.

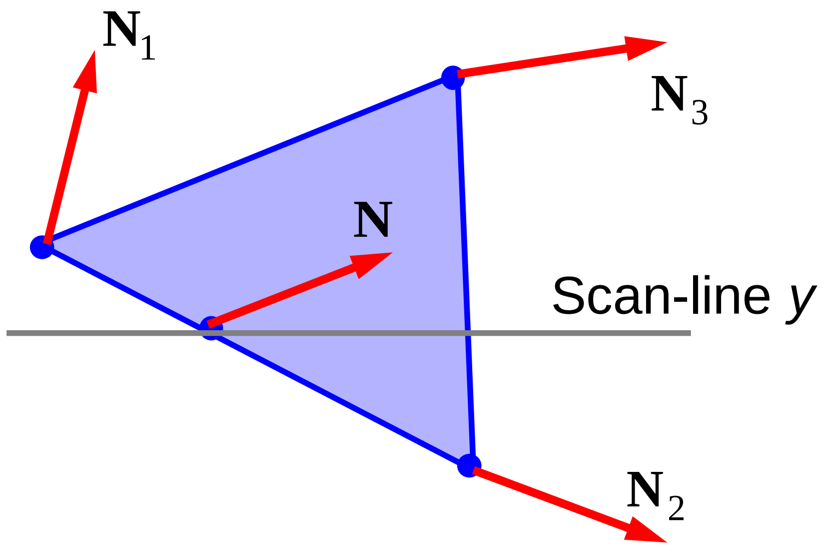
Phong Surface Rendering

Gouraud rendering may miss specular reflections and also create anomalies of bright or dark bands in image.

Gouraud rendering

Phong rendering

Instead of interpolating intensities, the average normal vectors at vertices are interpolated. Interpolation is done similarly as for intensities, taking advantage that *y* is progressing to *y-1* and *x* to *x+1*.



$$\mathbf{N} = \frac{y - y_2}{y_1 - y_2}\mathbf{N}_1 + \frac{y_1 - y}{y_1 - y_2}\mathbf{N}_2$$

The computation penalty is that the illumination model is now computed for every point of surface.

Fast Phong Surface Rendering

It significantly reduces the amount of computations at each point by approximating some of the illumination components (G. Bishop and D.M. Weimer, 1986).

Recall that rendering is done for projected polygons. Consider an object described by triangles. We interpolate the normal at (x, y) as $\mathbf{N}(x, y) = \mathbf{A}x + \mathbf{B}y + \mathbf{C}$, where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are determined from the equalities

$$\mathbf{N}(x_k, y_k) = \mathbf{A}x_k + \mathbf{B}y_k + \mathbf{C}, \; k = 1, 2, 3.$$

Omitting reflectivity and attenuation, light-source diffuse reflection is

$$I_{\text{diffuse}}(x,y)=\frac{\mathbf{L}\cdot\mathbf{N}}{|\mathbf{L}||\mathbf{N}|}=\frac{(\mathbf{L}\cdot\mathbf{A})x+(\mathbf{L}\cdot\mathbf{B})y+\mathbf{L}\cdot\mathbf{C}}{|\mathbf{L}||\mathbf{A}x+\mathbf{B}y+\mathbf{C}|}.$$

The following expression is obtained by few manipulations:

$$I_{\text{diffuse}}(x,y)=\frac{ax+by+c}{\left[dx^2+exy+fy^2+gx+hy+i\right]^{1/2}},$$

where $a=\dfrac{\mathbf{L}\cdot\mathbf{A}}{|\mathbf{L}|},\ b=\dfrac{\mathbf{L}\cdot\mathbf{B}}{|\mathbf{L}|},\ c=\dfrac{\mathbf{L}\cdot\mathbf{C}}{|\mathbf{L}|},\ d=\mathbf{A}\cdot\mathbf{A}$

$e=2\mathbf{A}\cdot\mathbf{B},\ f=\mathbf{B}\cdot\mathbf{B},\ g=2\mathbf{A}\cdot\mathbf{C},\ h=2\mathbf{B}\cdot\mathbf{C},\ i=\mathbf{C}\cdot\mathbf{C}.$

$I_{\text{diffuse}}\left(x,y\right)$ can be approximated by Taylor expansion

$$f\left(x_0+\xi,y_0+\eta\right)=f\left(x_0,x_0\right)+\left(\xi\frac{\delta}{\delta x}+\eta\frac{\delta}{\delta y}\right)f\left(x_0,y_0\right)$$

$$+\cdots+\frac{1}{n!}\left(\xi\frac{\delta}{\delta x}+\eta\frac{\delta}{\delta y}\right)^n f\left(x_0,y_0\right)+\cdots$$

Considering the first five terms in taylor expension

of $I_{\text{diffuse}}(x, y)$ yields

$$I_{\text{diffuse}}(x, y) = T_5 x^2 + T_4 xy + T_3 y^2 + T_2 x + T_1 y + T_0.$$

The various $T_i,\ 1 \le i \le 5,$ are functions of $a, b, \ldots, i$

calculated before, they are fixed for the entire

triangle and can be stored in registers.

Since the progression of scan line is $x \rightarrow x+1$ and $y \rightarrow y+1,\ I_{\text{diffuse}}(x, y)$ is fully calculated for the first point and then only two additions per pixel suffice.

The method can be extended to account sepecular reflections terms $(\mathbf{N} \cdot \mathbf{H})^{n_s}$.