CSC209: Computer Graphics

Unit 8 – Illumination Models and Surface Rendering Techniques

SUSAN SUNUWAR

Introduction

- Realistic displays of a scene are obtained by perspective projections and applying natural lighting effects to the visible surfaces of object.
- For realistic displaying of 3d scene it is necessary to calculate appropriate color or intensity for that scene.
- The realism of a raster scan image of a 3D scene depends upon the successful stimulation of shading effects.
- Once visible surface has been identified by hidden surface algorithm, a shading model is used to compute the intensities and color to display for the surface.

Illumination model or a lighting model or shading model

- It is a model for the interaction of light with a surface.
- It is the model for calculating light intensity at a **single surface point**. Sometimes also referred to as a shading model.
- An illumination model is used to calculate the intensity of the light that is reflected at a **given point on a surface**.

Surface Rendering/Surface Shading

- Rendering/Shading is the process of creating a high-quality realistic images or pictures.
- In 3-D graphic design, rendering is the process of add shading (how the color and brightness of a surface varies with lighting), color in order to create life-like images on a screen.
- Surface rendering is the process of calculating intensity values for **all pixel positions for the various surfaces in a scene**.
- A rendering method uses intensity calculations from the illumination model to **determine the light intensity at all pixels in the image**.

Surface Rendering/Surface Shading

- It is the process of applying illumination model to obtain the pixel intensities for all the projected surface positions in a scene.
- Surface rendering can be performed by applying the illumination model to every visible surface point, or the rendering can be accomplished by interpolating intensities across the surface.

Light Sources

- Object that radiates energy are called light sources, such as sun, lamp, bulb, fluorescent tube etc.
  - Point Light Source
  - Distributed Light Source

Point Light Source

- The rays emitted from a point light radially diverge from the source. Approximation for sources that are small compared to the size of objects in the scene. Radiates equal intensity in all directions. For example, sun.

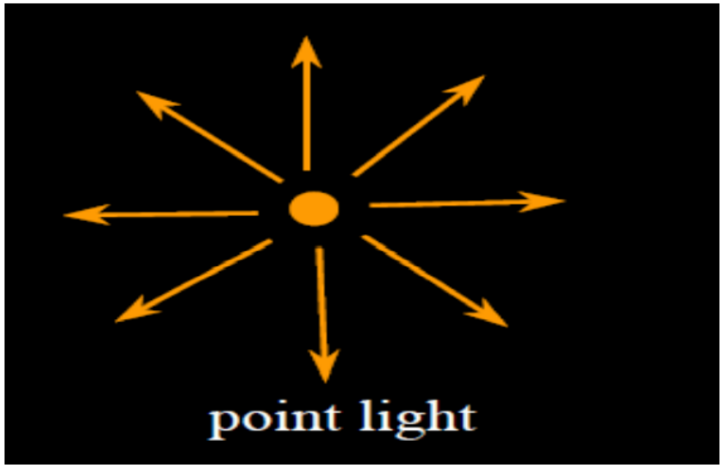


Fig: ***Diverging ray paths from a point light Source.***

Distributed Light Source

- A nearby source, such as the long fluorescent light. All the rays from a directional/distributed light source have the same direction, and no point of origin. All light rays are parallel.

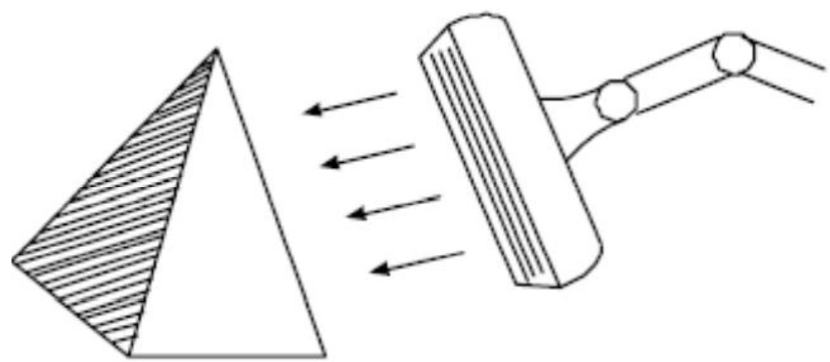
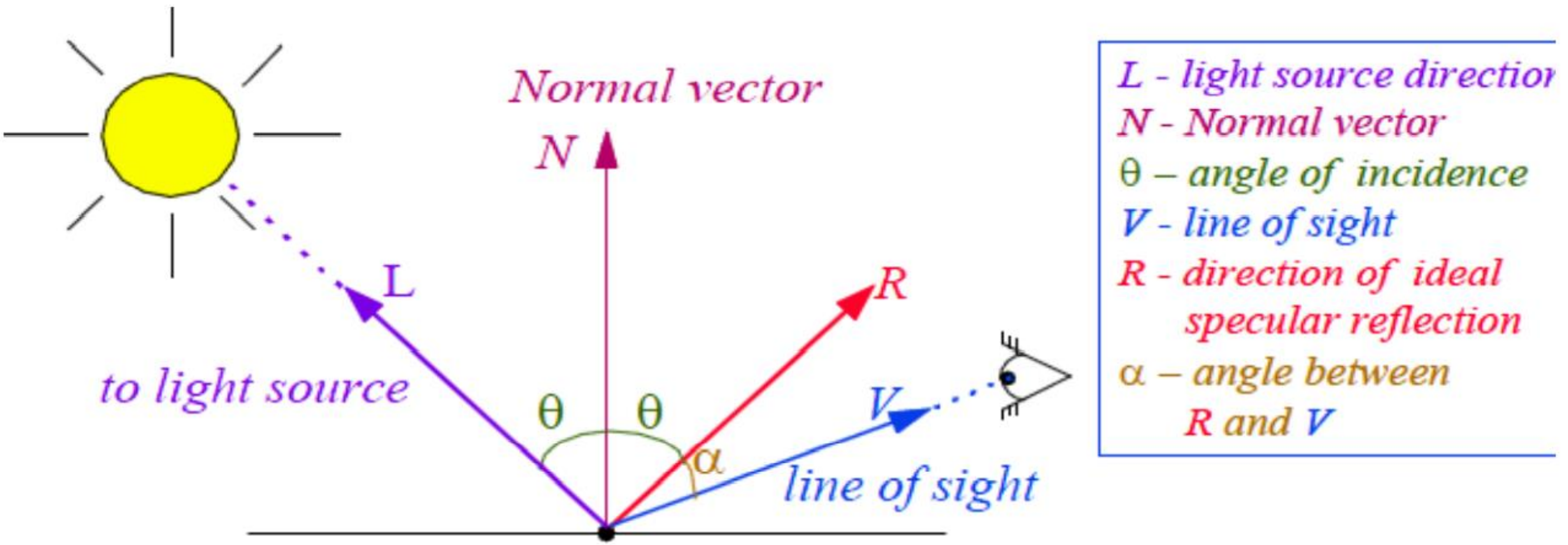


Fig: ***An object illuminated with a distributed light source.***

Interaction of Light Source with Surfaces

- When light is incident on opaque surface part of it is reflected and part of it is absorbed.
- The amount of incident light reflected by a surface depends on the type of material. **Shining material reflects more incident light and dull surface absorbs more of the incident light**.
- For transparent surfaces, some of the incident light will be reflected and some will be transmitted through the material.


Normal vector
N
L
R
to light source
θ
θ
α
V
line of sight
L - light source directio
N - Normal vector
θ – angle of incidence
V - line of sight
R - direction of ideal
specular reflection
α – angle between
R and V

Illumination Models

- Illumination models are used to calculate light intensities that we should see at a **given point on the surface of an object**.
- Intensity calculations are based on the optical properties of surfaces such as:
  - Reflectivity, opaque/transparent/translucent, shiny/dull, the background lighting conditions and the light source specifications.
- Some illumination models:
  - Ambient Light
  - Diffuse reflection
  - Specular reflection and Phong model

Ambient Light

- Surface that is not exposed directly to a light source still will be visible if nearby objects are illuminated. This light is called ambient light.
- This is a simple way to model the combination of light reflections from various surfaces to produce a uniform illumination called the ambient light, or background light.
- The amount of ambient light incident on each object is a **constant for all surfaces and over all directions**.
- If a surface is exposed only to ambient light, then the reflected intensity at any point on the surface is: $\boldsymbol{I = K_a I_a}$ Where $\boldsymbol{I_a}$ is the intensity of the ambient light, and $\boldsymbol{K_a}$ is the ambient reflection coefficient, $\boldsymbol{0 \leq K_a \leq 1}$.

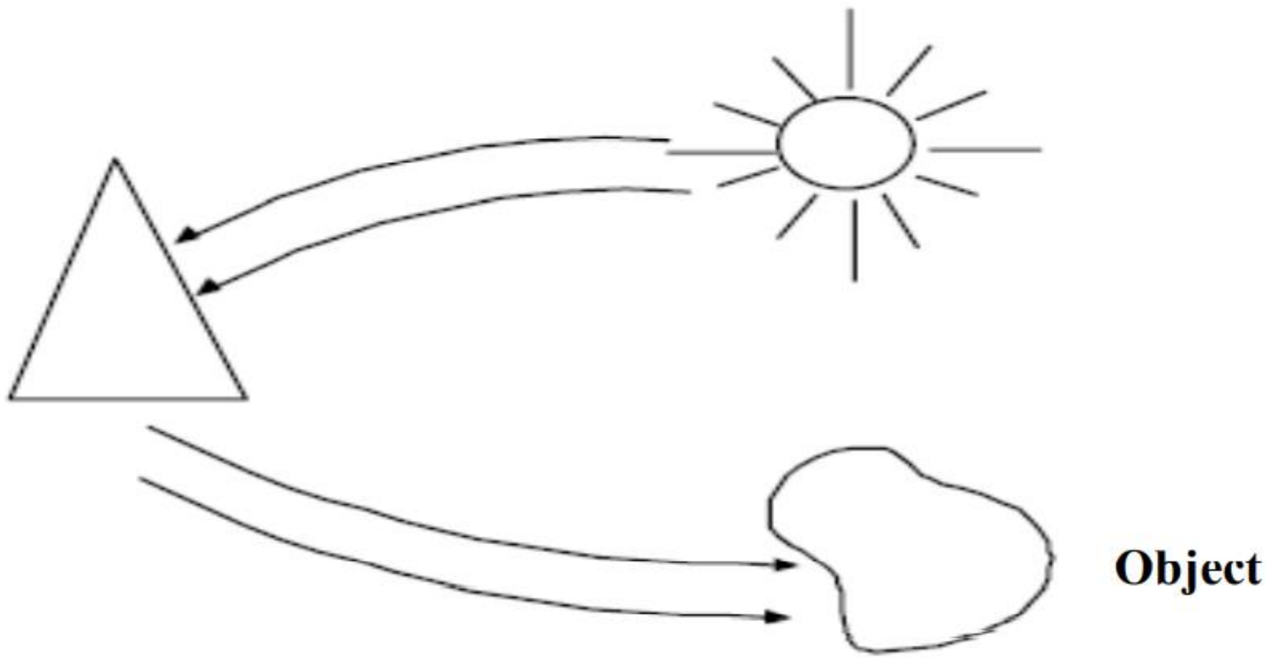


Fig: Object illuminated due to ambient light.

Diffuse reflection

- It is a reflection due to even scattering of light by uniform, rough surfaces.
- Rough surface tends to scatter the reflected light in all direction. The scattered light is called diffuse reflection. So, surface appears **equally bright from all viewing directions**, i.e., **reflected light is independent of viewing position**.
- Diffuse reflections are constant over each surface in a scene, independent of the viewing direction. Surfaces appear equally bright from all viewing angles since they reflect light with equal intensity in all directions.
- Color of an object is determined by the color of the diffuse reflection of the incident light. If any object surface is red, then there is a diffuse reflection for red component of light and all other components are absorbed by the surface.

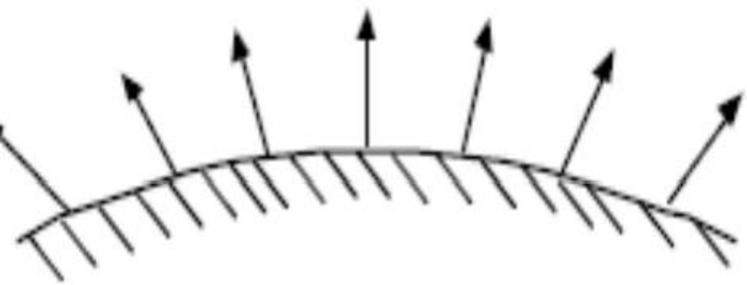


Fig: ***Diffuse reflections from a surface.***

Diffuse reflection

- The reflected Intensity I of any point on surface can be computed by **Lambert's Cosine Law**:

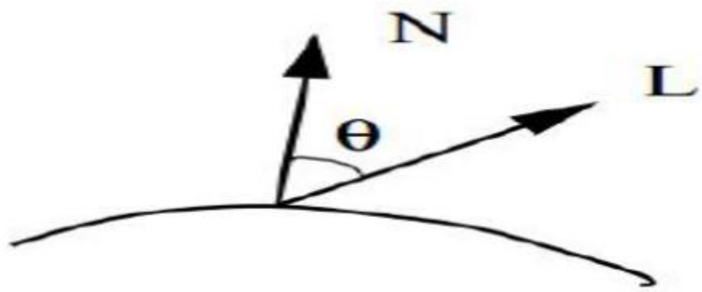


Fig: ***Angle of incidence θ between the unit light source direction vector L and the unit surface normal N***

$$I_{ldiff} = K_d I_l \cos\theta$$
$$I_{ldiff} = K_d I_l \,(N.L)$$
$$\because |N| = |L| = 1$$

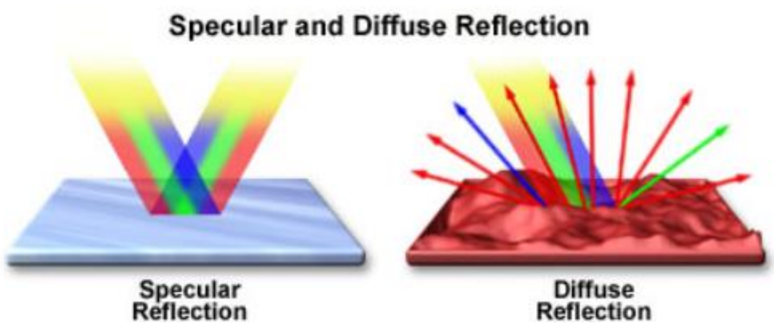
where N is the **unit** normal vector to a surface, L is the **unit** direction to the point light source from a position on the surface, I_l is the intensity of the point light source and K_d is the diffuse reflection coefficient

- Total diffuse reflection (I_{diff}) = Diffuse reflection due to ambient light + Diffuse reflection due to light source

$$I_{diff} = I_{ambdiff} + I_{ldiff}$$
$$I_{diff} = K_a I_a + K_d I_l (N.L)$$

Specular and Diffuse Reflection

Most things that we see (people, cars, houses, animals, trees, etc.) do not themselves emit visible light but reflect incident natural sunlight and artificial light. For instance, an apple appears a shiny red color because it has a relatively smooth surface that reflects red light and absorbs other non-red (such as green, blue, and yellow) wavelengths of light. The reflection of light can be roughly categorized into two types of reflection: **specular reflection** is defined as light reflected from a smooth surface at a definite angle, and **diffuse reflection**, which is produced by rough surfaces that tend to reflect light in all directions (as illustrated in Figure). There are far more occurrences of diffuse reflection than specular reflection in our everyday environment.



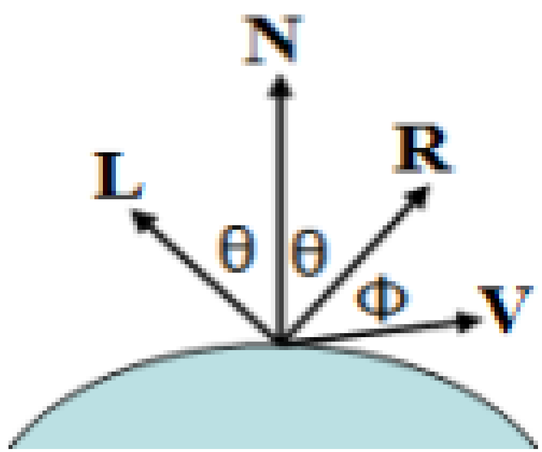
https://www.olympus-lifescience.com/en/microscope-resource/primer/java/reflection/specular/

Specular reflection and Phong model

- When we look at an illuminated shiny surface, such as polished metal, an apple etc., a highlight, or bright spot, at certain viewing directions is seen. This phenomenon, called specular reflection, is the result of **total, or near total, reflection of the incident light in a concentrated region around the "specular reflection angle"**.
- For an ideal reflector (perfect mirror), the angle of incident is equal to the angle of reflection or i.e., V and R coincide $(\emptyset = 0)$

Specular reflection and Phong model

- N - unit normal surface vector
- R - unit vector in the direction of ideal specular reflection
- L - unit vector directed towards point light source
- V - unit vector pointing to viewer from the surface position
- Ø - viewing angle relative to specular reflection direction R



Phong Model

- The empirical formula for calculating the specular reflection is given by Phong model.
- This is an empirical model, which is not based on physics, but physical observation. It sets the intensity of the specular reflection proportional to $\boldsymbol{cos^{\eta_s}\varphi}$, where $\boldsymbol{\eta_s}$ is specular reflection parameter and is determined by the type of surface. For very shiny surface $\boldsymbol{\eta_s}$ is set 100 and for dull surface $\boldsymbol{\eta_s}$ is set 1.
- The intensity of specular reflection can be modeled using specular reflection coefficient $W(\theta)$.

$$\boldsymbol{I_{spec} = W(\theta) I_l cos^{\eta_s}\varphi}$$

Where $\boldsymbol{I_l}$ is the intensity of the light source, θ angle of incidence. General variation of $\boldsymbol{W(\theta)}$ is over the range $\mathbf{0}^{\circ} - \mathbf{90}^{\circ}$. At $\boldsymbol{\theta = 90^{\circ}, W(\theta) = 1}$.

Phong Model

$$I_{spec} = W(\theta) I_l cos^{\eta_s} \varphi$$

For many opaque material surfaces, specular reflections are nearly constant for all incident angles. So, in such case, we can replace W(∅) with a constant coefficient(K_s), and the value lies between $\mathbf{0}$ & $\mathbf{1}$, for each surface:

$$I_{spec} = K_s I_l cos^{\eta_s} \varphi \ldots [1]$$

Since, V & R are unit vectors so, $|V| = |R| = 1$

$$V.R \ = \ |V|.|R| cos(\varphi)$$

$$V.R \ = cos(\varphi)$$

So, we can write equation [1] as:

$$I_{spec} = K_s I_l (R.V)^{\eta_s}$$

$$[R = 2(N.L)N - L]$$

Phong Model

So, for a single point light source, we can model the combined & specular reflections from a point on an illuminated surface as :

$$\boldsymbol{I = I_{diff} + I_{spec}}$$

$$\boldsymbol{I = K_a I_a \ + \ K_d I_l (N.L) + K_s I_l (R.V)^{\eta_s}}$$

Intensity Attenuation

- As radiant energy from a point light source travels through space, its amplitude is attenuated by the factor $\frac{1}{d^2}$, where d is the distance that the light has traveled.
- This means that a **surface close to the light source (small d) receives higher incident intensity** from the source than a distant surface (large d).
- Therefore, to **produce realistic lighting effects, illumination model should take intensity attenuation into account**. Otherwise, we are likely to illuminate all surfaces with same intensity.
- For a point light source attenuation factor is $\frac{1}{d^2}$.
- And for a distributed light source attenuation factor is given by inverse quadratic attenuation function, $f(d) = \frac{1}{a_0 + a_1 d + a_2 d^2}$

Color Consideration

- Most graphics displays of realistic scenes are in color. But the illumination model we have described so far only considers monochromatic lighting effects.
- To **incorporate color, we need to write the intensity equation as a function of the color properties of the light sources and object surfaces**. Diffuse reflection coefficient vector for RGB component (k_{dR}, k_{dG}, k_{dB})**.**
- For an RGB description, **each color in a scene is expressed in terms of red, green, and blue components. We then specify the RGB components of light-source intensities and surface colors, and the illumination model calculates the RGB components of the reflected light**.

Color Consideration

- If we want an object to have a blue surface, we select a nonzero value in the range from 0 to 1 for the blue reflectivity component, $\boldsymbol{k_{dB}}$, while the red and green reflectivity components are set to zero $(k_{dR} = k_{dG} = 0)$ ***[any nonzero red or green components in the incident light are absorbed, and only the blue component is reflected.]***

Intensity of blue color I_B,

$$I_B = K_{aB} I_{aB} + \sum_{i=1}^{n} f(d_i) I_{LBi} [k_{dB}(N.L_i) + k_{sR}(N.H_i)^{\eta_s}]$$

- Surfaces are typically illuminated with white light sources, and in general we can set surface color so that the reflected light has nonzero values for all three RGB components.
- Calculated intensity levels for each color component can be used to adjust the corresponding electron gun in an RGB monitor.

Color Consideration

- In his original specular-reflection model, Phong set parameter k_s to a constant value independent of surface color. This produces specular reflections that are same color as the incident light (usually white), which gives the surface plastic appearance.
- For a non-plastic material, the color of the specular reflection is a function of surface properties and may be different from the color of the incident light and the color of the diffuse reflections.

Transparency and Shadows

- A **transparent** surface, in general, produces both reflected and transmitted light.
- The relative contribution of the transmitted light depends on the degree of transparency of the surface and whether any light sources or illuminated surfaces are behind the transparent surface.
- Both diffuse and specular transmission can take place at the surfaces of a transparent object.
- Diffuse effects are important for partially transparent surfaces such as frosted glass.
- When light is incident upon a transparent surface, part of it is reflected and part is refracted.

θ_i = angle of incidence
$\eta_i = index\ of\ refraction\ of\ incident\ material$
$\theta_r = angle\ of\ refraction$
$\eta_r = index\ of\ refraction$

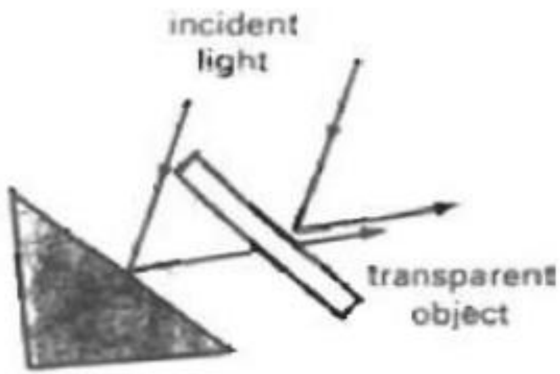


Fig: Light emission from a transparent surface is in general a combination of reflected and transmitted light.

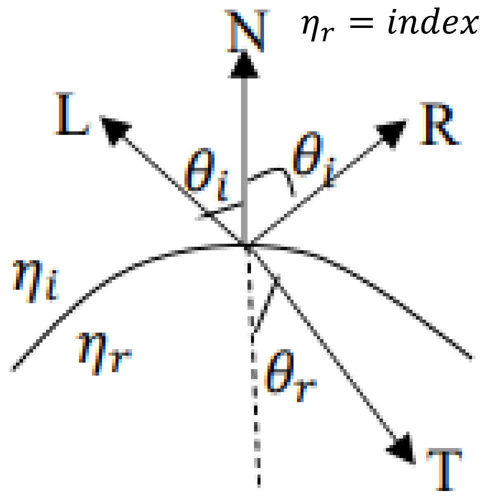


Fig: Reflection direction R and refraction direction T for a ray of light incident upon a surface with index of refraction η_r

According to Snell's law,

$$Sin\ \theta_r = \frac{\eta_i}{\eta_r} sin\theta_i$$

Transparency and Shadows

- Hidden surface method can be used to locate area where light sources produce **shadows**. By applying a hidden surface method with a light source at the view position, we can determine which surface sections cannot be seen from the light source.
- Shadow depends on two factors: **distance from the light source**, and the **angle of light incidence on object**.
- Shadow areas are usually displayed with ambient-light intensity.

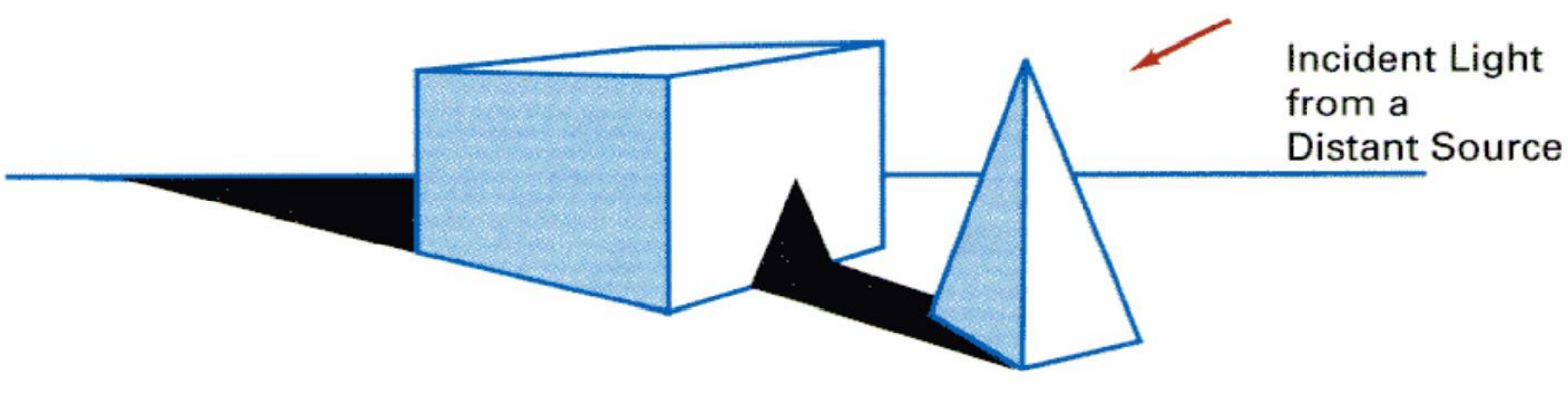


Fig: Shadow

Polygon/Surface Rendering

- Surface-rendering procedures are also called surface-shading methods. It is the process of applying illumination model to **obtain the pixel intensities for all the projected surface positions in a scene**.
- Each surface can be rendered using:
  - **Rendering entire surface with a single intensity**: very fast but does not produce realistic surfaces.
  - **Interpolation Scheme**: intensity values are interpolated to render the surfaces. Widely used approach, produces more realistic object surfaces than first method. Still suffers from Mach Band Effect.
  - **By applying the illumination model to every visible surface point**: best option, widely used approach, produces best quality surfaces, but requires large computations, so comparatively slow.

Polygon/Surface Rendering

- Three widely used approaches:
  - Constant Intensity shading Method (Flat Shading)
  - Gouraud Shading method (Intensity Interpolation)
  - Phong Shading Method (Normal Vector Interpolation).

Constant Intensity shading Method (Flat Shading)

- The fast and simplest model for shading/rendering a polygon is constant intensity shading also called faceted Shading or flat shading.
- In this approach, the illumination model is applied only once for each polygon to determine a single intensity value.
- The entire polygon is then displayed with the single intensity value.
- It does not provide realistic displaying.

Constant Intensity shading Method (Flat Shading)

- It provides an accurate rendering for an object if all the following assumptions are valid:
  - Polygon surface must be one face of a polyhedron and is **not a section of a curved-surface**.
  - The **light source is sufficiently far from the surface** so that $\vec{N}.\vec{L}$ is constant across the polygon face.
  - The **viewing position is sufficiently far from the surface** so that $\vec{V}.\vec{R}$ is constant over the surface.

Algorithm

- Divide polygon surface into polygon meshes.
- Determine surface unit normal vectors for each polygon.
- Calculate intensity value for a point for each surface (usually at the center).
- Apply this intensity value to all the points of that surface.

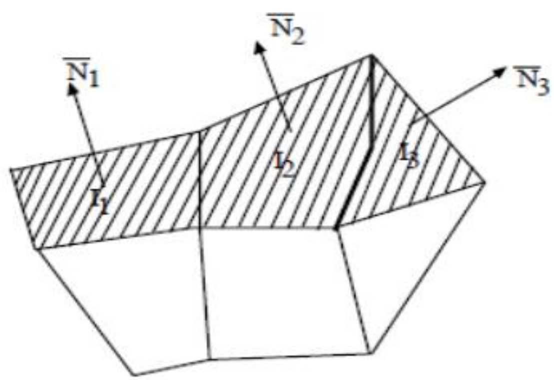


Fig: Flat Shading

Gouraud Shading

- It is an intensity interpolating shading or color interpolating shading introduced by Henri Gouraud.
- The polygon surface is displayed by **linearly interpolating intensity values across the surface**.
- Idea is to **calculate intensity values at polygon vertices**. Then, **linearly interpolate these intensities** across polygon surfaces of an object.

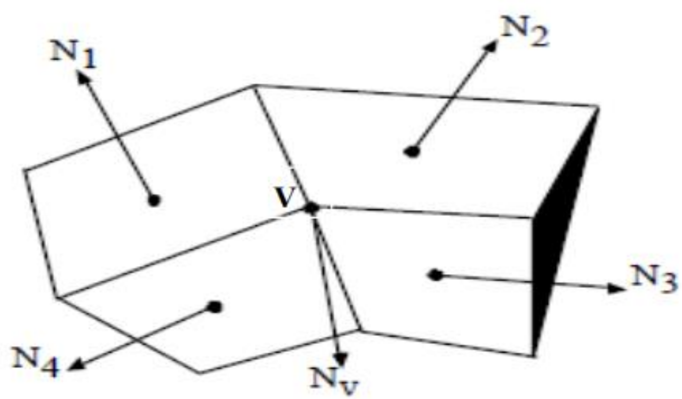


Fig: Gouraud Shading

Algorithm

1. Determine the average unit normal vector at each polygon vertex.
   - At each polygon vertex, we obtain a normal vector by averaging the surface normal of all polygons sharing that vertex.
   - Therefore, average unit normal vector at vertex V, is given by

$$\boldsymbol{Nv} = \frac{\sum_{k=1}^{n} \boldsymbol{N_k}}{|\sum_{k=1}^{n} \boldsymbol{N_k}|},$$ where n is the number of adjoining surfaces.

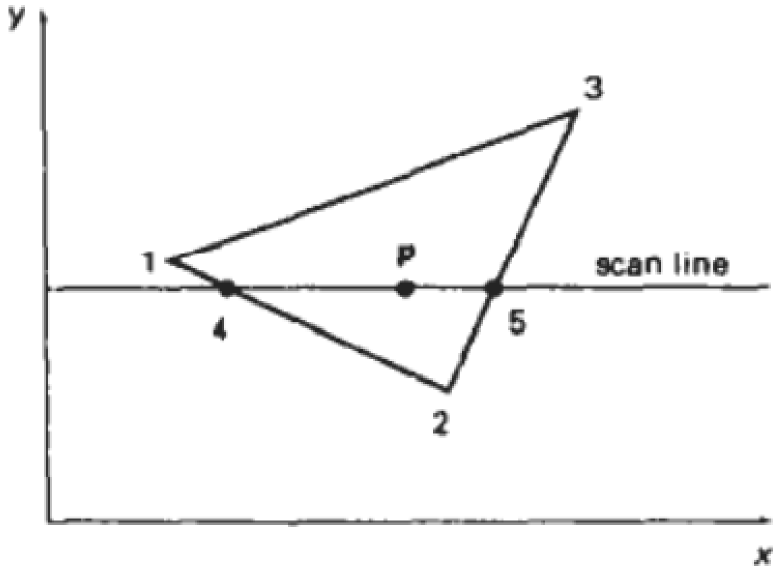
For above figure:

$$\boldsymbol{Nv} = \frac{\sum_{k=1}^{4} \boldsymbol{N_k}}{|\sum_{k=1}^{4} \boldsymbol{N_k}|} = \frac{\boldsymbol{N_1} + \boldsymbol{N_2} + \boldsymbol{N_3} + \boldsymbol{N_4}}{|\boldsymbol{N_1} + \boldsymbol{N_2} + \boldsymbol{N_3} + \boldsymbol{N_4}|}$$

Algorithm

2. Apply an illumination model to each vertex to calculate the vertex intensity.

3. **Linearly interpolate the vertex intensities** over the surface of the polygon.



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

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

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

Advantages

- It provides more realistic graphics than constant intensity shading.
- It eliminates intensity discontinuities that occur in flat shading.

Disadvantages

- It can cause bright or dark intensity streaks to appear on the surface called Mach banding.
- Involves additional computation.

Phong Shading

- Best known shading algorithm, developed by Phong Bui Tuong, is called Phong shading or normal vector interpolation shading.
- A **more accurate** method for rendering a polygon surface.
- Idea here is to **interpolate normal vectors instead of the light intensity** and then apply the illumination model to each surface point.
- Phong shading calculates the average unit normal vector at each of the polygon vertices and then interpolates the vertex normal over the surface of the polygon.

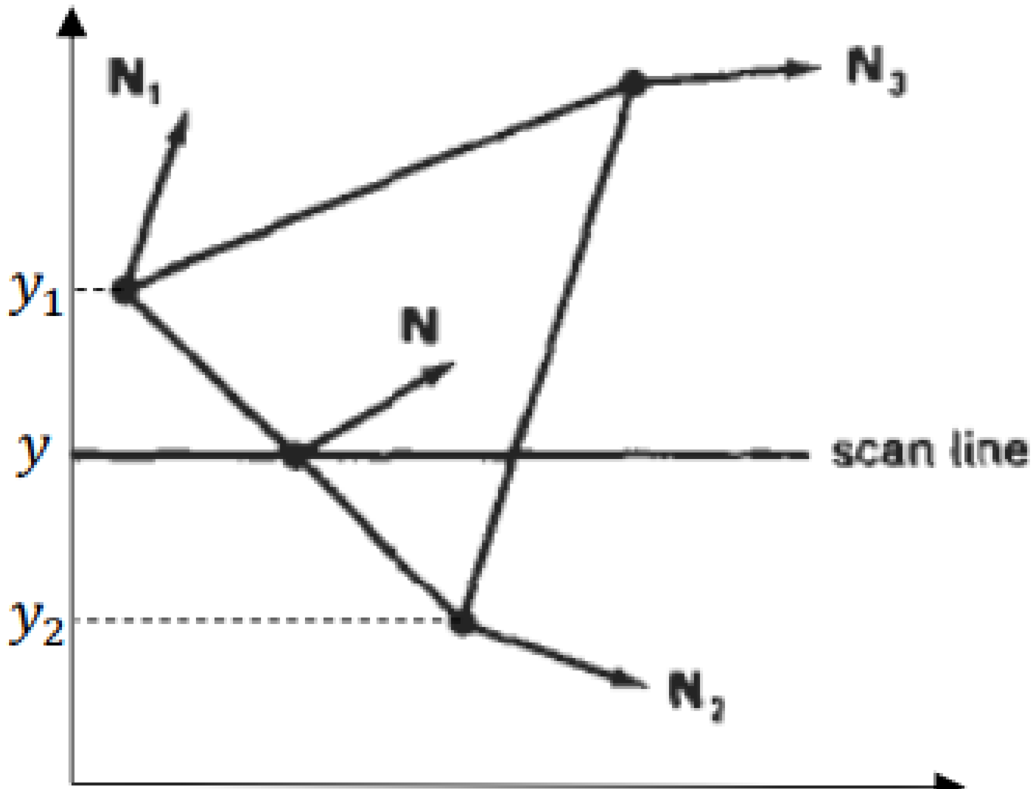
Algorithm

1. Determine the average unit normal vector at each polygon vertex.

   $\overrightarrow{\boldsymbol{N_{avg}}} = \frac{\sum_{i=1}^{n} \overrightarrow{\boldsymbol{N_i}}}{|\sum_{i=1}^{n} \overrightarrow{\boldsymbol{N_i}}|}$, where n is the number of adjoining surfaces.

2. Linearly **interpolate the vertex normal** over the projected area of the polygon.

3. Apply illumination model at position along each scan line to calculate pixel intensities using interpolated normal vectors.



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

Advantages

- It displays more realistic highlights on a surface.
- It greatly reduces the Mach band effect.
- It gives more accurate results.

Disadvantages

- It requires more calculation and greatly increases the cost of shading steeply.

Fast Phong Shading

- Fast phong shading approximates the intensity calculations using a Taylor-series expansion and triangular surface patches.
- Surface normal at any point (x, y) over a triangle as

$$N = Ax + By + C$$

- A, B, C are determined from the three vertex equations:

$$N_k = Ax_k + By_k + C, \qquad k = 1,2,3$$

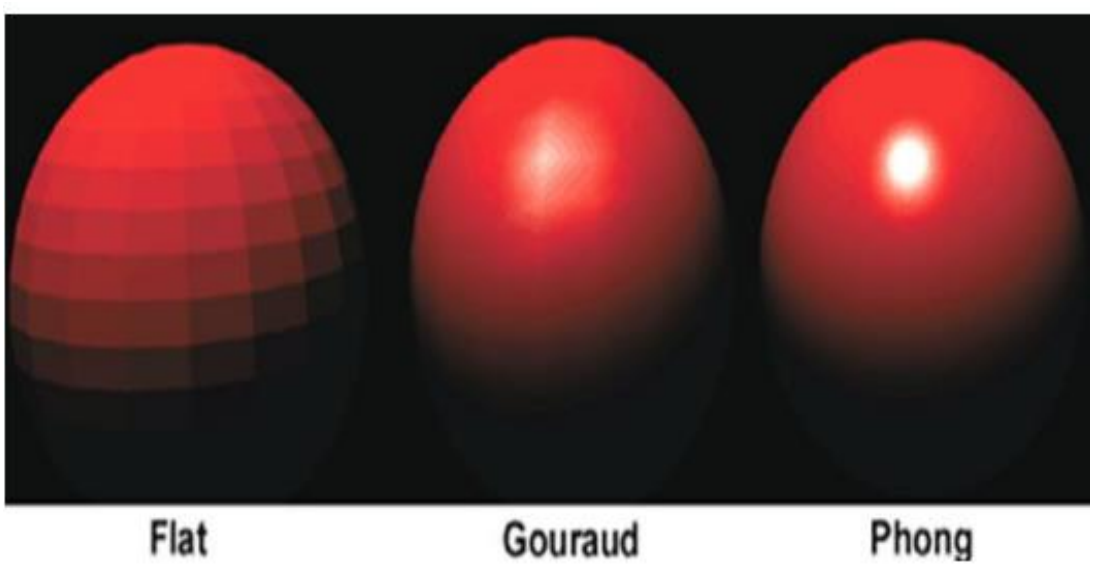
with (x_k, y_k) denoting a vertex position.

- Omitting the reflectivity and attenuation parameters,

$$I_{diff}(x,y) = \frac{L.N}{|L|.|N|} = \frac{L.(Ax + By + C)}{|L|.|Ax + By + C|}$$

Fast Phong Shading

- FPS is two times slower than Gouraud shading
- Normal Phong shading is 6 to 7 times slower than Gouraud
- FPS can be extended to include specular reflections



Flat shading	Gouraud shading	Phong shading
Computes illumination once per polygon and apply it to whole polygon.	Computes illumination at vertices and interpolates.	Applies illumination at every point of polygon surface.
Creates discontinuous in color.	Interpolates colors along edges and scan line.	Interpolates normal instead of colors.
Problems of Mach bands.	Handles Mach bands problem found in flat shading.	Removes Mach bands completely.
Low cost.	Not so expensive.	More expensive than gouraud shading.
Requires very less processing and is fast in time.	Required moderate processing time.	Requires complex processing and is slower but is more efficient as compared to other shading methods.
Lighting equation used once per polygon.	Lighting equation used at each vertex.	Lighting equation used at each pixel.

Offline or pre-computed 3D rendering

- Offline rendering allows you to have very high-quality graphics, which can be close to reality.
- It does not offer any interactivity because, as the name suggests, everything is calculated beforehand by a computer. This technology is mainly used by the cinematographic or real estate industry to present products and replace real photoshoots for example.
- It is therefore a **rendering dedicated to visualization**.

Real-time 3D rendering

- Real-time rendering incorporates the notion of **interactivity** because the scene, composed of a multitude of images, is calculated instantaneously just before being displayed on a screen. This gives the impression that the scenes unfold in real time with each user interaction.
- However, although the rendering quality is still decent, it will **not be as good as in offline rendering**.
- This technology is particularly used in video games. As a result, the user can interact with the 3D scene using various devices such as mouse, keyboard, joystick, tablet, etc. Overall, this technology can be found in most devices: from iPhones, computers and VR headsets, to video game consoles.

Offline vs Real-time 3D rendering

- But today, graphics cards and render engines have improved their software and hardware performances in order to push further and further towards photorealism. The gap between real time and offline is therefore narrowing more and more.
- Moreover, this gap is shrinking even more with the emergence of a technological breakthrough called **ray-tracing (RTX)**.
- Ray tracing is a graphic rendering technique that simulates the path of light rays and their interaction with various materials, resulting in a very realistic final rendering.