Unit 8: Illumination Model and Surface Rendering Technique

Prepared by Sumesh Gajmer

Illumination Model

Illumination model, also known as Shading model or Lighting model, is used to calculate the intensity of light that is reflected at a given point on surface.

There are three factors on which lighting effect depends on:

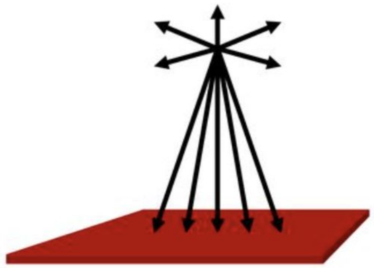
1. Light Source :

- Light source is the light emitting source.
- There are three types of light sources.

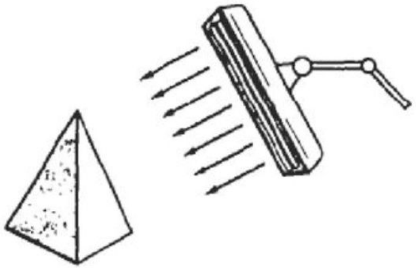
1. Point Sources – The source that emit rays in all directions (A bulb in a room).

2. Parallel Sources – Can be considered as a point source which is far from the surface (The sun).

3. Distributed Sources – Rays originate from a finite area (A tubelight).



Point Source



Distributed Source

2. Surface : When light falls on a surface part of it is reflected and part of it is absorbed. Now the surface structure decides the amount of reflection and absorption of light. The position of the surface and positions of all the nearby surfaces also determine the lighting effect.

3. Observer : The observer’s position and sensor spectrum sensitivities also affect the lighting effect

Ambient Light

Ambient light is indirect light that comes from all directions.

It does not originate from a specific source but is rather scattered light in the environment.

It is used in shading models to ensure objects are not completely dark, even if they are not directly lit by a light source.

Independent of : Viewer and object position

Dependent of : A constant factor (in each of the R,G,B channel)

Properties of Ambient Light

1. **Constant Illumination** – The same amount of ambient light reaches all objects regardless of their position or orientation.

2. **No Spatial or Directional Properties** – Ambient light is uniformly distributed and does not come from a specific direction like a spotlight or sun.

3. **Constant Reflection** – The intensity of ambient light reflected from a surface remains the same, no matter the viewing angle.

4. **Dependent on Surface Properties** – The final reflected light depends on the material of the object; brighter surfaces reflect more ambient light, while darker surfaces absorb more.

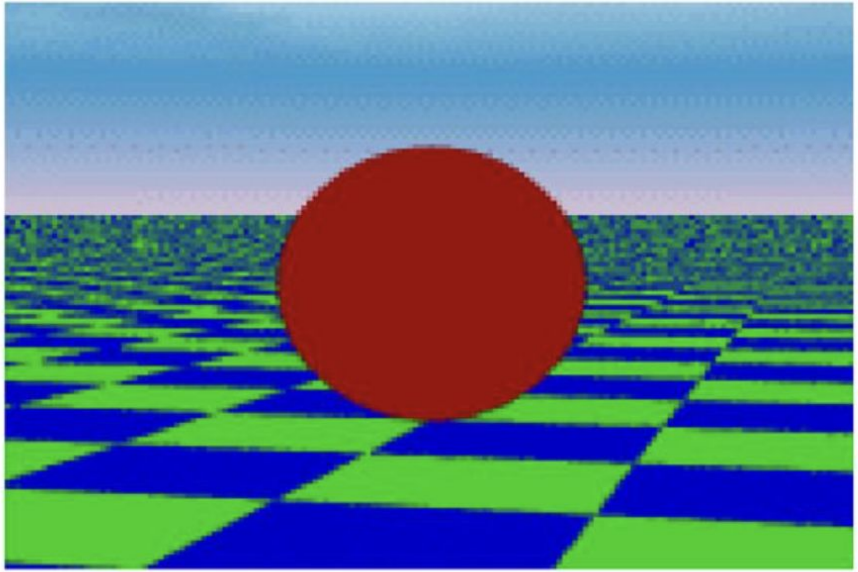
The reflected intensity I_{amb} of any point on the surface is:

$$I_{amb} = K_a \, I_a$$

Where as, K_a is the surface ambient reflectivity, varies from 0 to 1

I_a is the ambient light intensity.

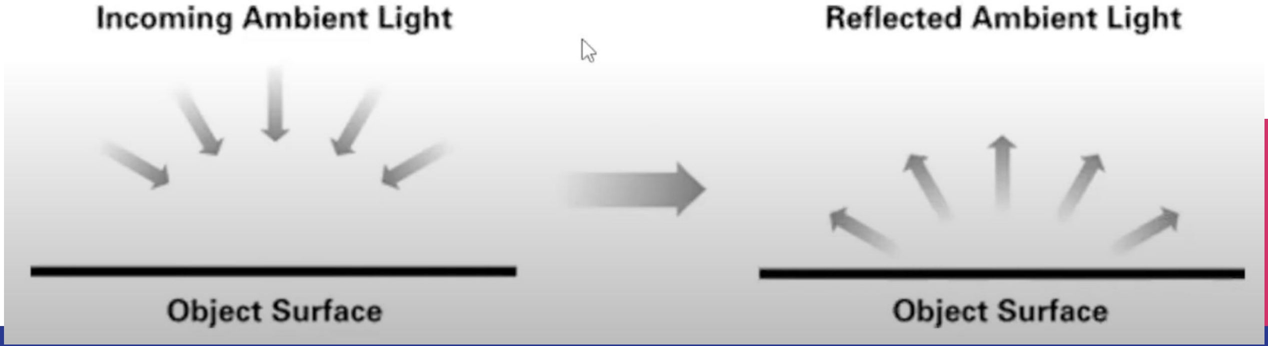
Example: Assume you are standing on a road, facing a building with glass exterior and sun rays are falling on that building reflecting back from it and the falling on the object under observation. This would be Ambient Illumination.



Light Position
Not Important

Viewer Position
Not Important

Surface Angle
Not Important

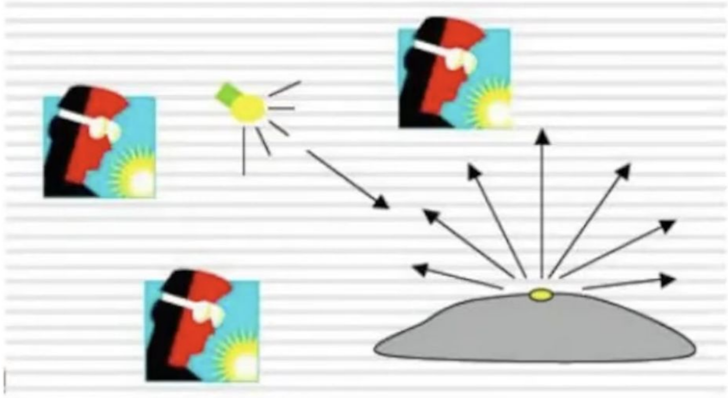


Diffuse Reflection:(Lambertian Reflection)

When some intensity of light falls on the object surface and that surface reflect the light in all direction in equal amount then the resulting reflection is called diffuse reflection.

It is a illumination that a surface receives from a light source that reflects equally in all direction.

- **Independent of:** Viewer position
- **Dependent of:** Light position and Surface position



The reflected intensity I_{diff} of a point on the surface is:

$$I_{diff} = K_d \, I_p \, (N.L)$$

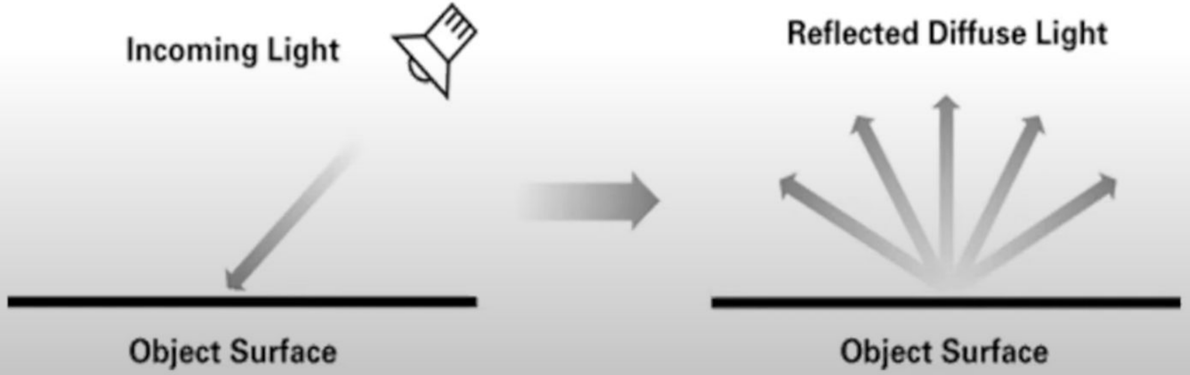
where as,

K_d is surface diffuse reflectivity, varies from 0 to 1

I_p is the point light intensity

N is surface normal

L is light direction

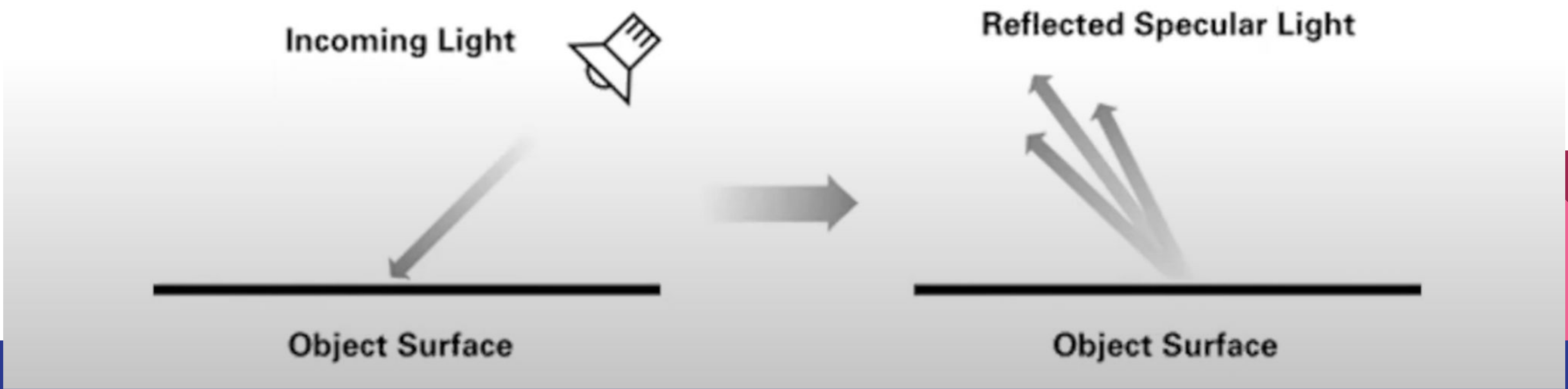


Specular Reflection:

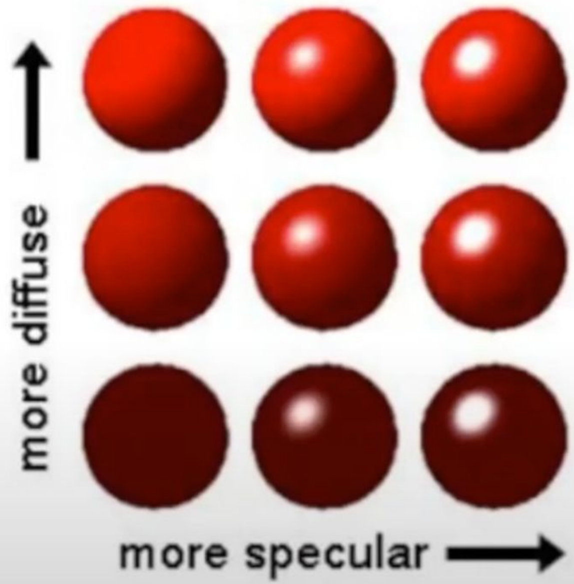
When we look at an illuminated shiny surface, such as polished metal we see a highlight, or bright spot, at certain viewing directions. This phenomenon is called **specular reflection.**

In short, it is a light reflection from a shiny surface(metal,mirror etc).

Dependent on : Light source position, viewer's position



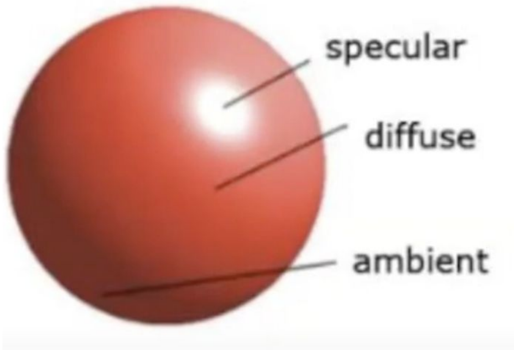
Putting diffuse and specular together in a basic illumination model

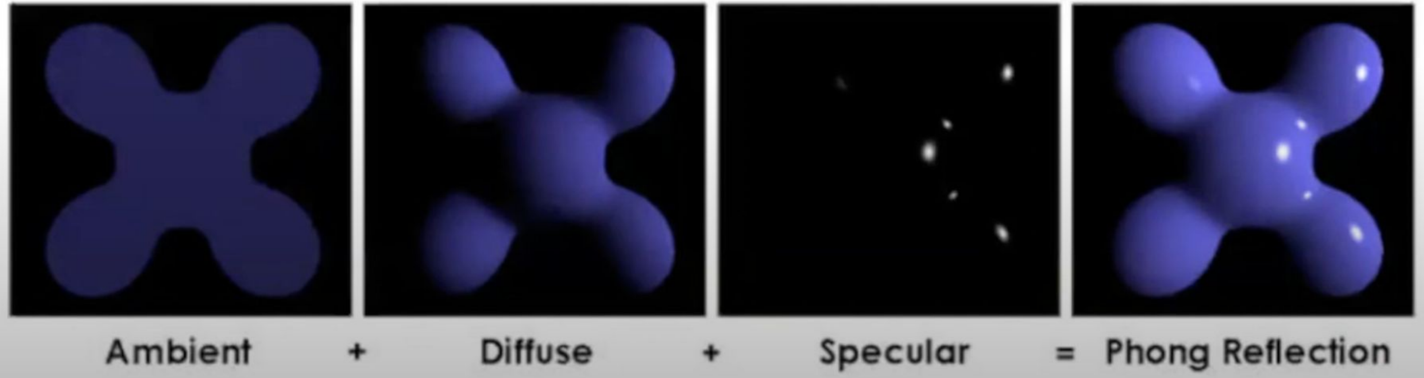


Phong Model:

In this model, we think of the interaction between light and a surface as having three distinct components:

1. Ambient
2. Diffuse
3. Specular





$$I_{spec} = W(\theta)\, I_l \cos^n(\Phi)$$

where, $W(\theta)$: K_s

L : direction of light source

N : normal to the surface

R : direction of reflected ray

V : direction of observer

Θ : Angle between L and R

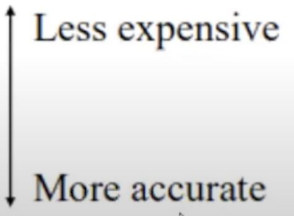
Φ : angle between R and V

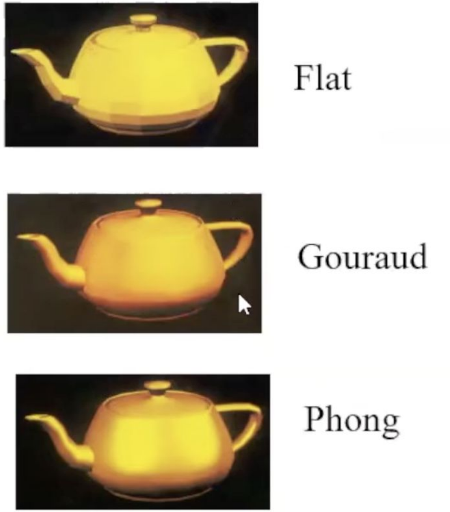
Polygon (Surface) Rendering Technique

Polygon Rendering means giving proper intensity at each point in a graphical object to make it look like real world object.

Different Rendering methods are

- Constant Intensity Shading
- Gouraud Shading
- Phong Shading



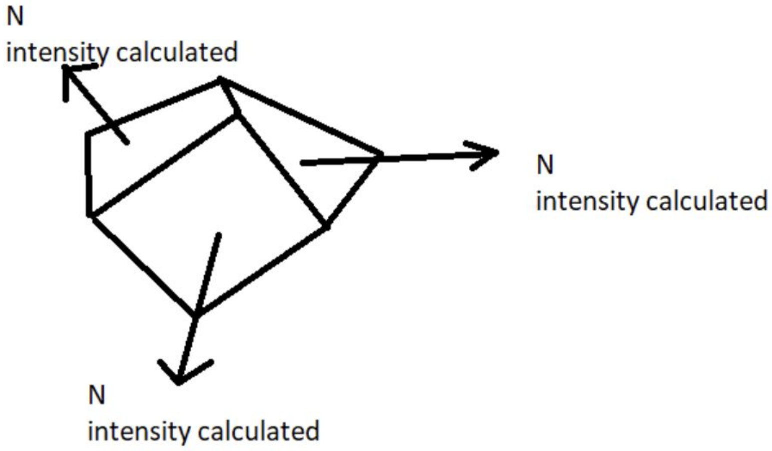


Constant Intensity Shading

It is also called flat shading.

It is fast and simple method.

This approach applies an illumination model once to determine a single intensity value that is then used to shade entire polygon, and holding the value across the polygon to reconstruct the polygon's shade.



Gouraud Shading

Gouraud shading is an **interpolation technique** used to produce smooth shading across polygons.

Named after **Henri Gouraud**, this method calculates vertex colors and then interpolates them across the polygon's surface.

Gouraud shading produces visually appealing results and is computationally less expensive than other shading models like Phong shading.

Gouraud shading may not capture specular highlights and subtle lighting variations as effectively as more advanced models.

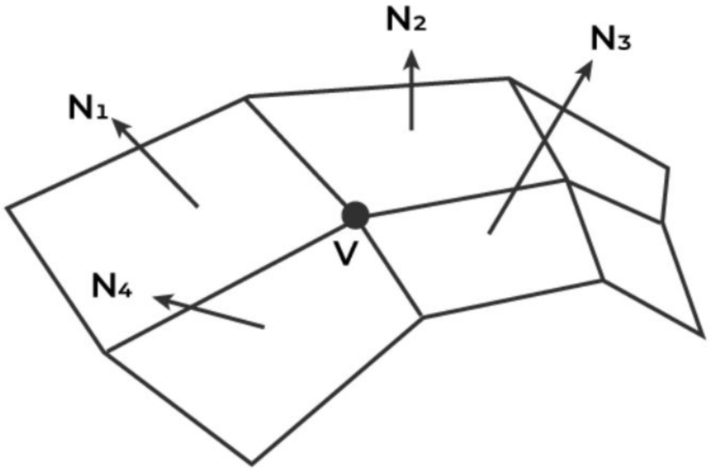
Color intensity at each vertex is computed based on the lighting model (e.g., Phong reflection model) and material properties.

Each polygon surface is rendered with Gouraud Shading by performing the following calculations:

1. Determining the average unit normal vector at each polygon vertex.
2. Apply an illumination model to each vertex to determine the vertex intensity.
3. Linear interpolate the vertex intensities over the surface of the polygon.

At each polygon vertex, we obtain a normal vector by averaging the surface normals of all polygons staring that vertex as shown in fig:

Once we have the vertex normals, we can determine the intensity at the vertices from a lighting model.



The Normal Vector at Vertex V is Calculated as the Average of the Surface Normal for each Polygon Sharing that Vertex

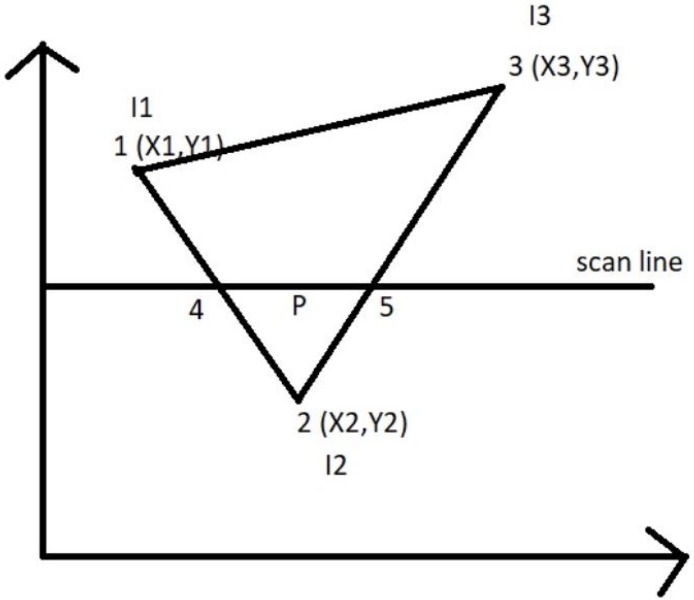
$$\mathbf{N} = \frac{\sum_{k=1}^{N} \mathbf{N}_k}{\left|\sum_{k=1}^{N} \mathbf{N}_k\right|}$$

$$\mathbf{N} = \frac{\mathbf{N}_1 + \mathbf{N}_2 + \mathbf{N}_3 + \mathbf{N}_4}{|\mathbf{N}_1 + \mathbf{N}_2 + \mathbf{N}_3 + \mathbf{N}_4|}$$

Following figures demonstrate the next step: Interpolating intensities along the polygon edges. For each scan line, the intensities at the intersection of the scan line with a polygon edge are linearly interpolated from the intensities at the edge endpoints.

For example: In fig, the polygon edge with endpoint vertices at position 1 and 2 is intersected by the scanline at point 4. A fast method for obtaining the intensities at point 4 is to interpolate between intensities I_1 and I_2 using only the vertical displacement of the scan line.

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



Similarly, the intensity at the right intersection of this scan line (point 5) is interpolated from the intensity values at vertices 2 and 3.

Once these bounding intensities are established for a scan line, an interior point (such as point P in the previous fig) is interpolated from the bounding intensities at point 4 and 5 as

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

Phong Shading

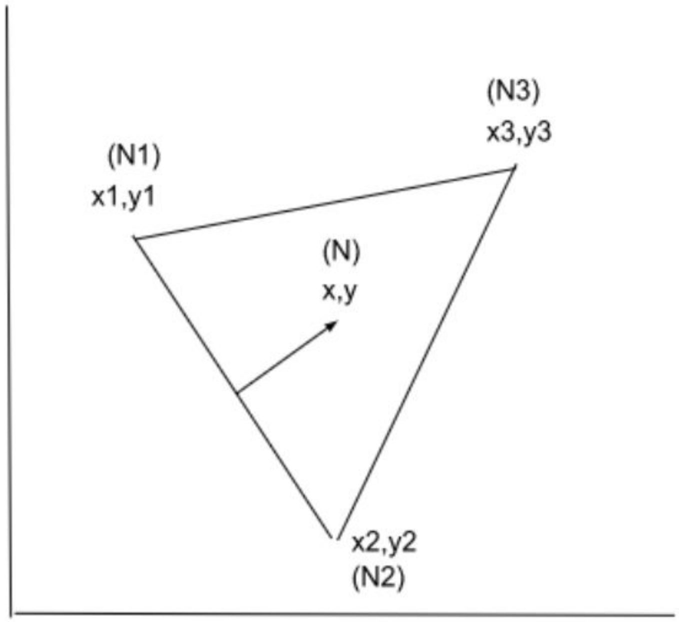
A more accurate method for rendering a polygon surface is to interpolate the normal vector and then apply the illumination model to each surface point.

This method developed by **Phong Bui Tuong** is called Phong Shading or normal vector Interpolation Shading. It displays more realistic highlights on a surface and greatly reduces the Match-band effect.

A polygon surface is rendered using Phong shading by carrying out the following steps:

- Determine the average unit normal vector at each polygon vertex.
- Linearly & interpolate the vertex normals over the surface of the polygon.
- Apply an illumination model along each scan line to calculate projected pixel intensities for the surface points.

Interpolation of the surface normal along a polynomial edge between two vertices as shown in fig:



$$\mathbf{N} = \frac{y - y_2}{y_1 - y_2}\,\mathbf{N_1} + \frac{y_1 - y}{y_1 - y_2}\,\mathbf{N_2}$$

Intensity Attenuation

Intensity Attenuation refers to the gradual decrease in the intensity (brightness) of light as it travels from a light source.

In computer graphics and OpenGL, attenuation is used to simulate how real-world light behaves over distance, making scenes look more realistic.

Why Does Light Attenuate?

In real life, light intensity decreases because of:

- **Distance:** The farther the light travels, the weaker it becomes.
- **Absorption & Scattering:** Light gets absorbed or scattered by particles in the environment

In **OpenGL and Phong lighting models**, intensity attenuation is usually calculated as:

$$I = \frac{I_0}{K_c + K_l d + K_q d^2}$$

where:

- I = Final intensity at a point
- I_0 = Initial intensity of the light source
- d = Distance from the light source to the point
- K_c = **Constant attenuation** (used to prevent complete darkness at small distances)
- K_l = **Linear attenuation** (controls how light fades linearly)
- K_q = **Quadratic attenuation** (simulates real-world light falloff)