Illumination Models & Surface-Rendering Methods

Illumination Models & Surface-Rendering Methods

- ***Illumination model*** or a ***lighting model*** is the model for calculating light intensity at a single surface point.
- ***Rendering*** is the process of generating an image from a 2D or 3D model by means of **computer** programs
- ***Surface rendering*** is a procedure for applying a lighting model to obtain pixel intensities for all the projected surface positions in a scene.

- ***Lighting*** **-** the process of computing the luminous intensity reflected from a specified 3-D point
- ***Shading*** **-** the process of assigning a colors to a pixels

Surface rendering

- 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 from a small set of illumination-model calculations.
- Scan-line algorithms use interpolation schemes.
- Ray tracing algorithms invoke the illumination model at each pixel position.
- Surface-rendering procedures are termed ***surface-shading methods***.

Illumination models

Given the parameters:

- the optical properties of surfaces (opaque/transparent, shiny/dull, surface-texture);
- the relative positions of the surfaces in a scene;
- the color and positions of the light sources;
- the position and orientation of the viewing plane.

Illumination models calculate the intensity projected from a particular surface point in a specified viewing direction.

Light Sources

- When we view an opaque (nontransparent) nonluminous object, we see reflected light from the surfaces of the object.
- The total reflected light is the sum of the contributions from ***light sources*** and other reflecting surfaces in the scene.
- Light sources = ***light-emitting sources***.
- Reflecting surfaces = ***light-reflecting sources***.

Light Sources

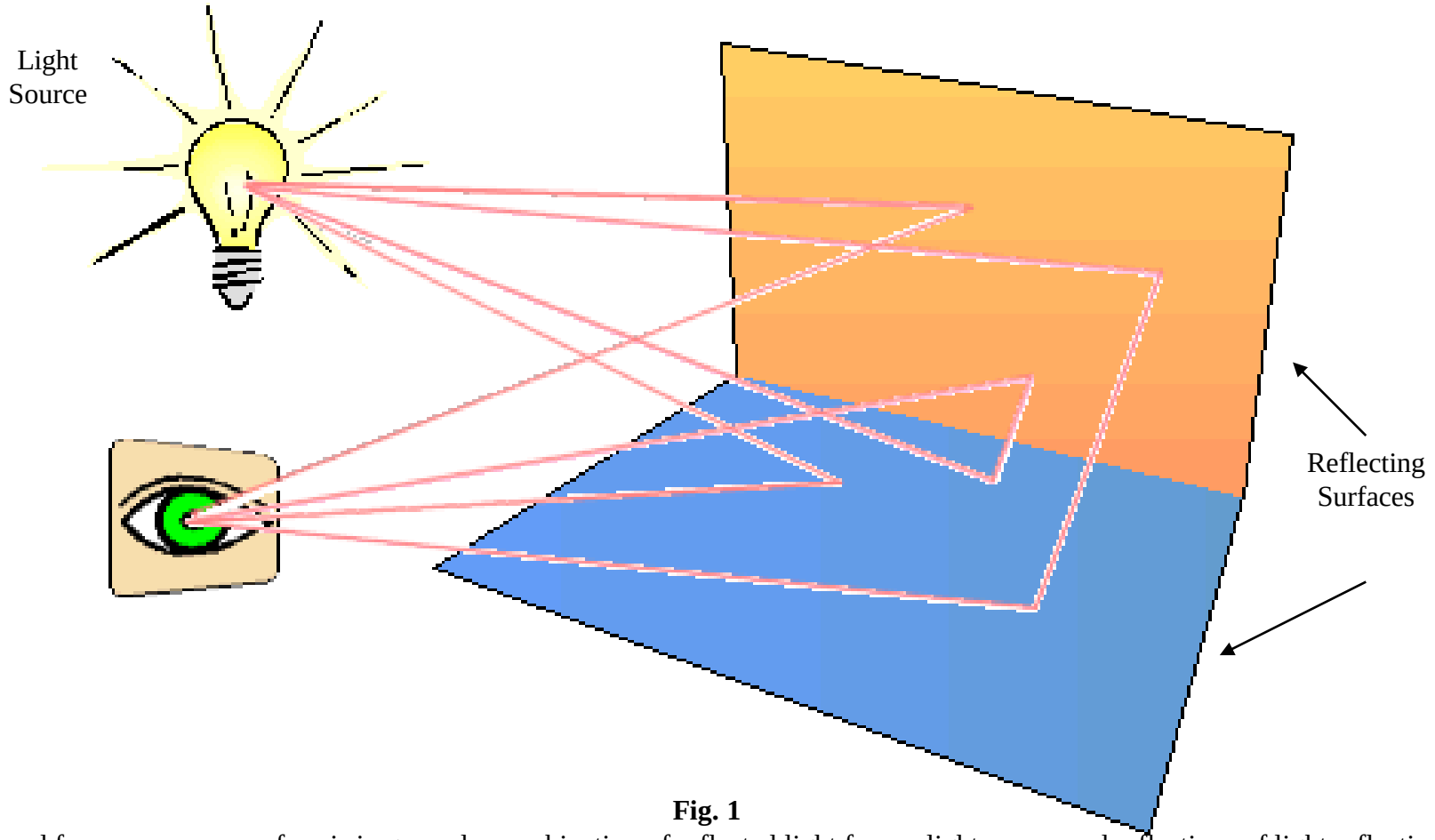


Fig. 1
Light viewed from an opaque surface is in general a combination of reflected light from a light source and reflections of light reflections from other surfaces.

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.
- A point light source is a fair approximation to a local light source such as a light bulb.
- The direction of the light to each point on a surface changes when a point light source is used.

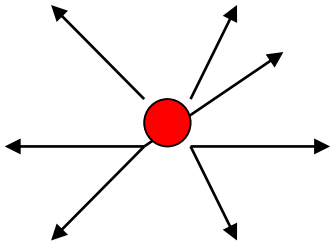


Fig. 2
Diverging ray paths from a point light source.

Distributed Light Source

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

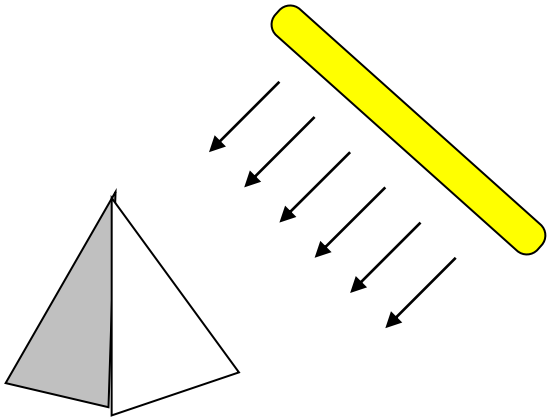


Fig. 3
An object illuminated with a distributed light source.

- It is as if the light source was infinitely far away from the surface that it is illuminating(bright).
- Sunlight is an example of an infinite light source.

Materials

- When light is incident on an opaque surface, part of it is reflected and part is absorbed.
- Shiny materials reflect more of the incident light, and dull surface absorb more of the incident light.
- For an illuminated transparent surface, some of the incident light will be reflected and some will be transmitted through the material.

Diffuse reflection

- Grainy surfaces scatter the reflected light in all directions. This scattered light is called ***diffuse reflection***.
- Diffuse reflection occurs from dull, matte surfaces, like latex paint, or chalk. These diffuse reflectors reradiate light equally in all directions. Picture a rough surface with lots of tiny micro facets.
- Diffuse reflections are constant over each surface in the scene, independent of the viewing direction.

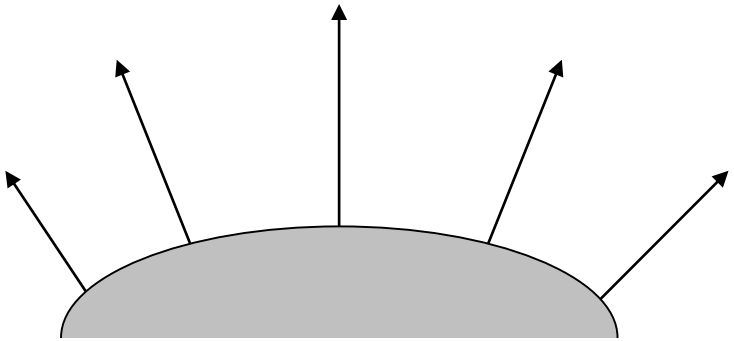


Fig. 4
Diffuse reflection from a surface.

Specular reflection

- Light sources create highlights, bright spots, called ***specular reflection***. More pronounced on shiny surfaces than on dull.

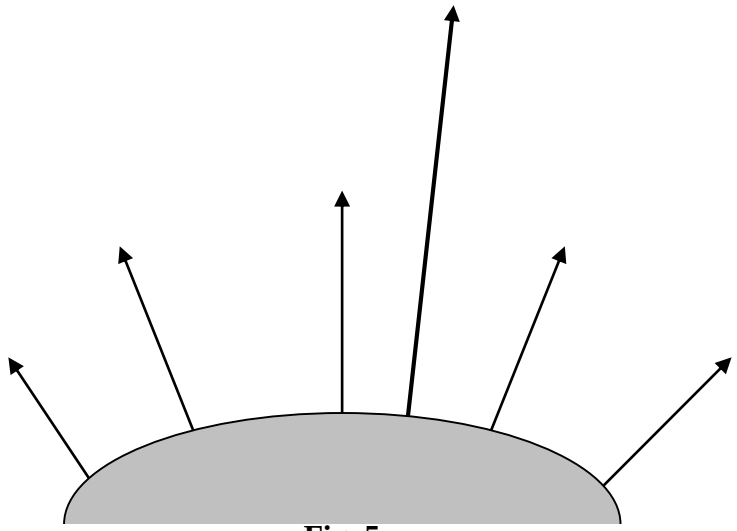
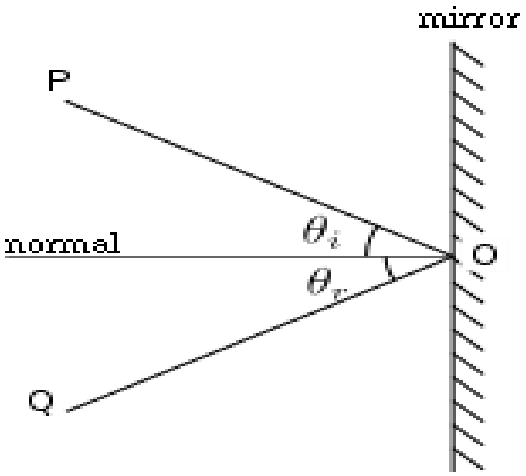
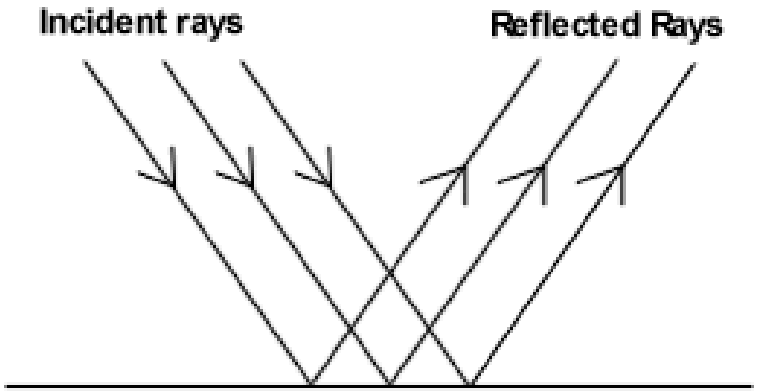


Fig. 5
Specular reflection superimposed on diffuse reflection vectors.

- **Specular reflection**
- is the mirror-like reflection of light (or of other kinds of wave) from a surface, in which light from a single incoming direction (a ray) is reflected into a single outgoing direction. Such behavior is described by the **law of reflection**, which states that the direction of incoming light (the incident ray), and the direction of outgoing light reflected (the reflected ray) make the same angle with respect to the surface normal, thus *the angle of incidence equals the angle of reflection* and that the incident, normal, and reflected directions are coplanar.

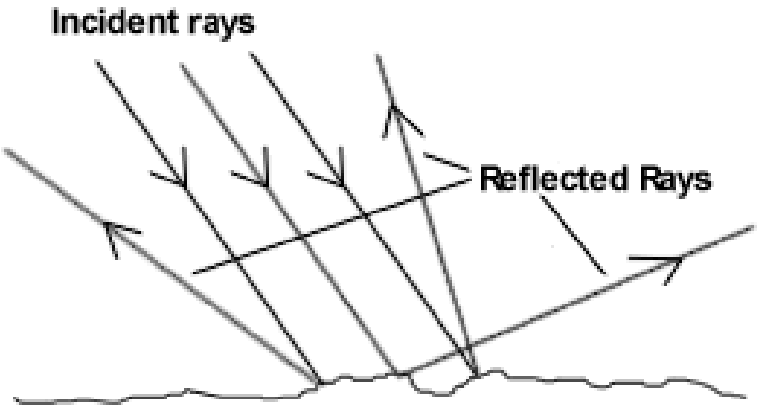


Regular Reflection



Eg. plane mirror or any other surface that produces a reflected image.

Diffuse Reflection



This is like any surface that we can see but does not reflect an image

Lighting calculations are based on:

- Optical properties of surfaces, such as glossy, matte, opaque, and transparent. This controls the amount of reflection and absorption of incident light.
- The background lighting conditions.
- The light-source specifications. All light sources are considered to be point sources, specified with a coordinate position and intensity value (color).

Ambient Light

- The lights are coming from different sources.

- Ambient light is the light that exists in a scene. Also referred to as “natural light” or “existing light,” ambient light can be the found light inside a home, a restaurant or concert hall, or a bright, sunny day, a deep foggy day, a city at night… in other words, any kind of pre-existing light. This is found light, not additional light that you, the photographer, might choose to add, such as flash

Ambient Light

- Even though an object in a scene is not directly lit it will still be visible. This is because light is reflected from nearby objects.
- Ambient light has no spatial or directional characteristics.

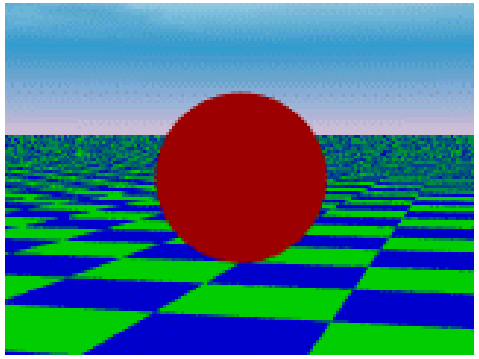


Fig. 6
Ambient light shading.

- The amount of ambient light incident on each object is a constant for all surfaces and over all directions.
- The amount of ambient light that is reflected by an object is independent of the objects position or orientation and depends only on the optical properties of the surface.

- In photography and cinematography, **available light** or **ambient light** refers to any source of light that is not explicitly supplied by the photographer for the purpose of taking photos.
- The term usually refers to sources of light that are already *available* naturally (e.g. the sun, moon, lightning) or artificial light already being used (e.g. to light a room).It generally excludes flashes, although arguably flash lighting provided by other photographers shooting simultaneously in the same space could be considered available light.
- Light sources that affect the scene and are included in the actual frame are called **practical light sources**, or simply **practicals**

- Use of available light is an important factor in candid photography in order not to disturb the subjects.
- The use of available light may pose a challenge for a photographer.
- The brightness and direction of the light is often not adjustable, except perhaps for indoor lighting. This will limit the selection of shutter speeds, and may require the use of shades or reflectors to manipulate the light.
- It can also influence the time, location, and even orientation of the photo shoot to obtain the desired lighting conditions. Available light can often also produce a color cast with color photography.

Ambient Light

- The level of ambient light in a scene is a parameter I_a, and each surface illuminated with this constant value.
- Illumination equation for ambient light is

$$I = k_a I_a$$

where

I is the resulting intensity

I_a is the incident ambient light intensity

k_a is the object's basic intensity, ***ambient-reflection coefficient***.

Ambient Light - Example

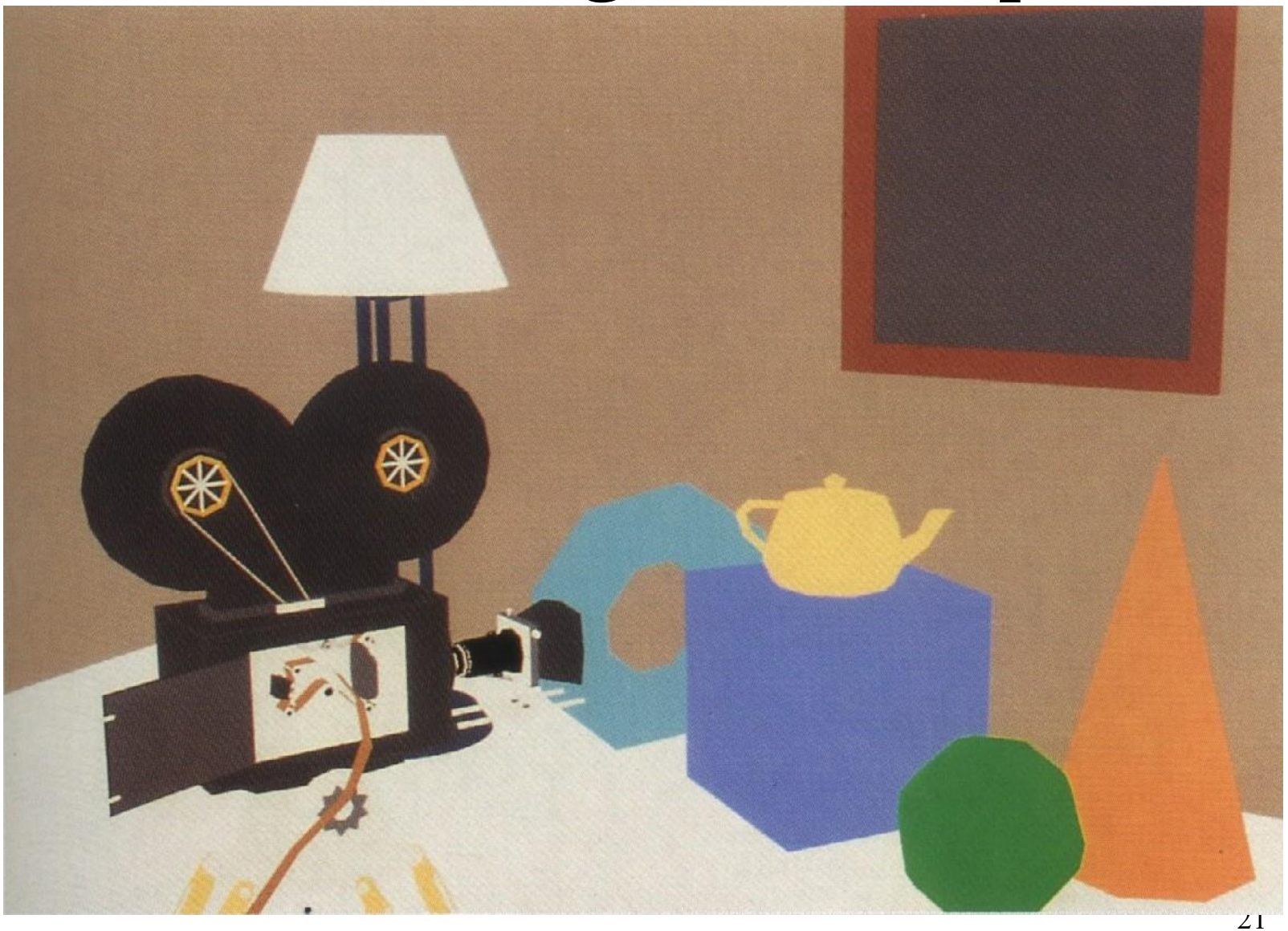
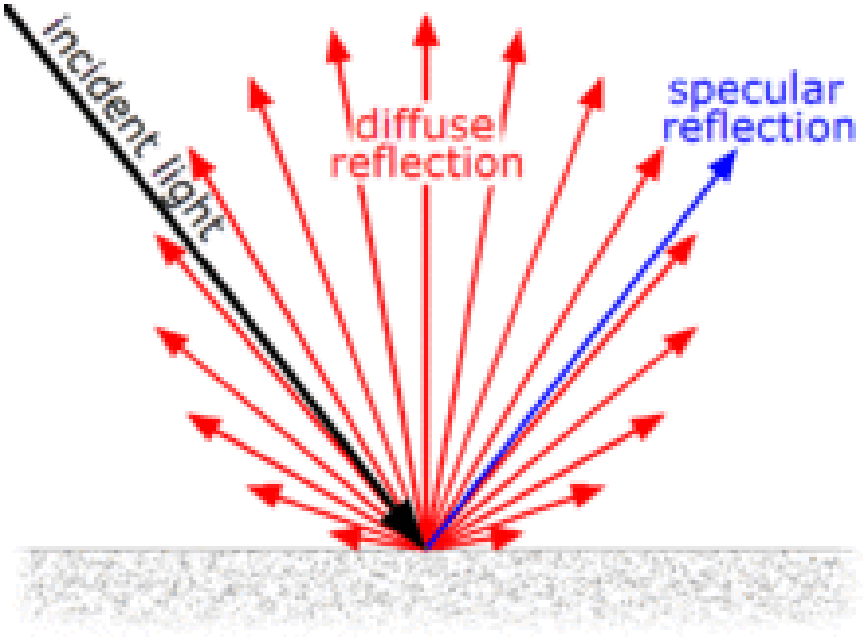


Fig. 7
An ambient illumination only.

Diffuse reflection

- **Diffuse reflection** is the reflection of light from a surface such that an incident ray is reflected at many angles rather than at just one angle as in the case of specular reflection. An illuminated *ideal* diffuse reflecting surface will have equal luminance from all directions which lie in the half-space adjacent to the surface



Diffuse Reflection

- Diffuse reflections are constant over each surface in a scene, independent of the viewing direction.
- The amount of the incident light that is diffusely reflected can be set for each surface with parameter k_d, the ***diffuse-reflection coefficient***, or ***diffuse reflectivity***.

  $0 \leq k_d \leq 1$;

  k_d near 1 – highly reflective surface;

  k_d near 0 – surface that absorbs most of the incident light;

  k_d is a function of surface color;

Diffuse Reflection

Even though there is equal light scattering in all direction from a surface, the brightness of the surface does depend on the orientation of the surface relative to the light source:

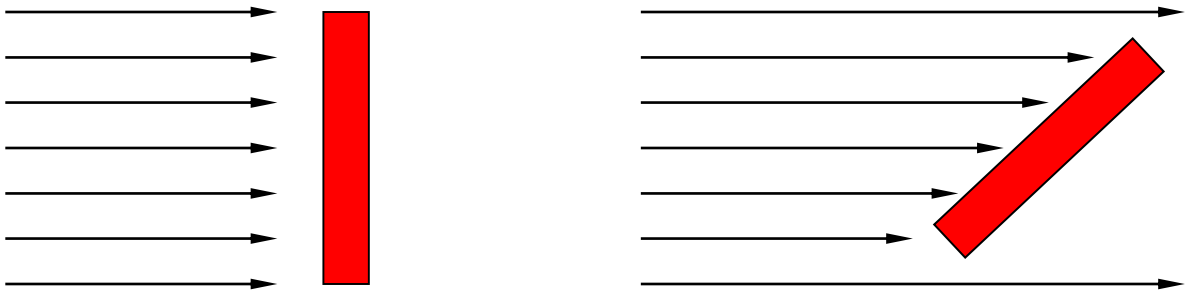


Fig. 8
A surface perpendicular to the direction of the incident light (a) is more illuminated than an equal-sized surface at an oblique angle (b) to the incoming light direction.

Diffuse Reflection

- As the angle between the surface normal and the incoming light direction increases, les of the incident light falls on the surface.
- We denote the ***angle of incidence*** between the incoming light direction and the surface normal as θ. Thus, the amount of illumination depends on $\cos\theta$. If the incoming light from the source is perpendicular to the surface at a particular point, that point is fully illuminated.

Diffuse Reflection

If I_l is the intensity of the point Light source, then the diffuse reflection equation for a point on the surface can be written as

$$I_{l,diff} = k_d I_l \cos\theta$$

or

$$I_{l,diff} = k_d I_l (\mathrm{N}\cdot\mathrm{L})$$

where

N is the unit normal vector to a surface and **L** is the unit direction vector to the point light source from a position on the surface.

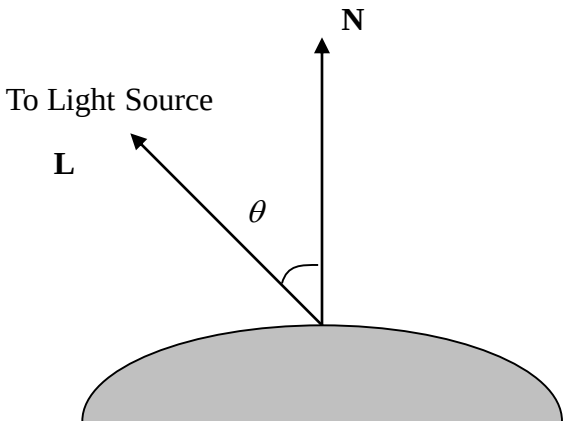


Fig. 9
Angle of incidence θ between the unit light-source direction vector **L** and the unit surface normal **N**.

Diffuse Reflection

Figure 10 illustrates the illumination with diffuse reflection, using various values of parameter k_d between 0 and1.

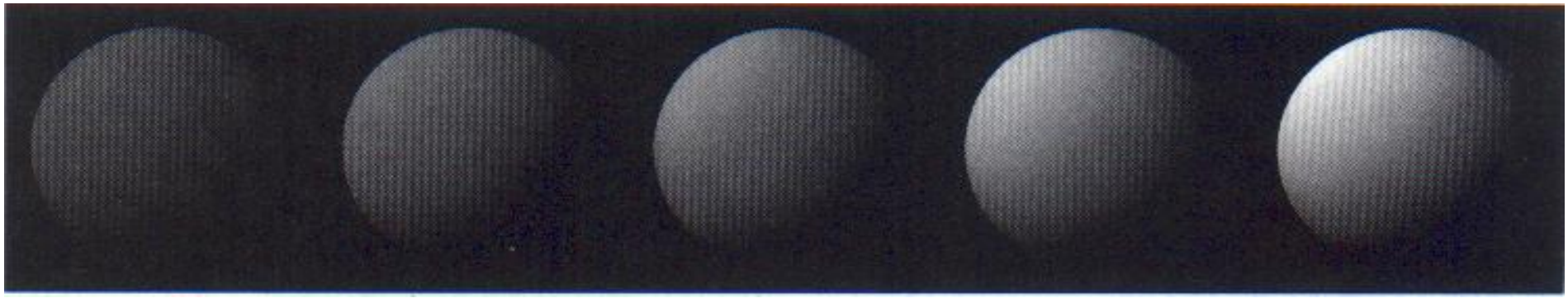


Fig. 10
Series of pictures of sphere illuminated by diffuse reflection model only using different k_d values (0.4, 0.55, 0.7, 0.85,1.0).

Diffuse Reflection

We can combine the ambient and point-source intensity calculations to obtain an expression for the total diffuse reflection.

$$I_{diff} = k_a I_a + k_d I_l(\mathbf{N}\cdot\mathbf{L})$$

where both k_a and k_d depend on surface material properties and are assigned values in the range from 0 to 1.

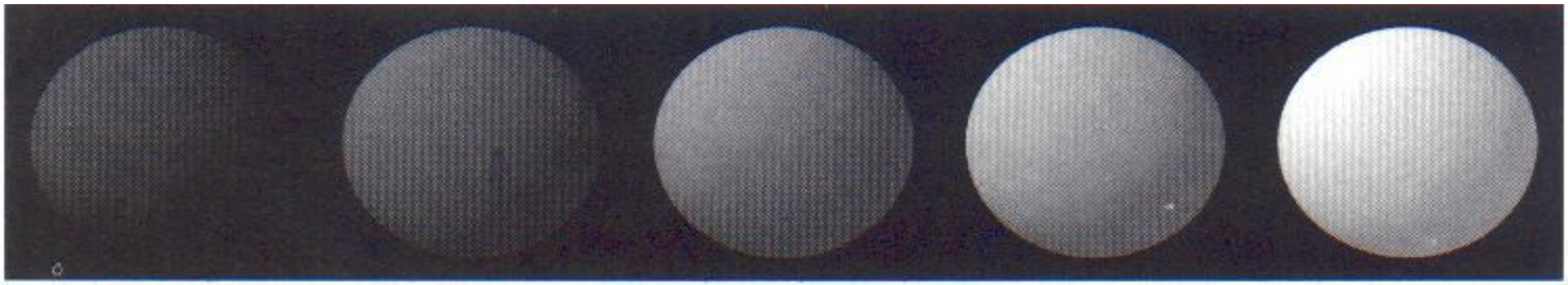


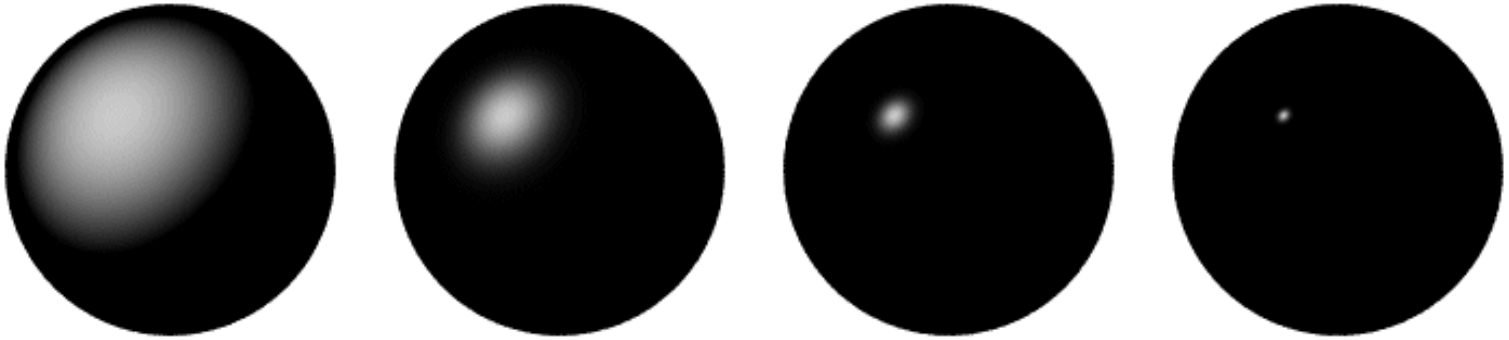
Fig. 11
Series of pictures of sphere illuminated by ambient and diffuse reflection model.
$I_a = I_l = 1.0$, $k_d = 0.4$ and k_a values (0.0, 0.15, 0.30, 0.45, 0.60).

Diffuse Reflection - Example

Fig. 12
Individually shaded polygons with diffuse reflection.

Specular Reflection and the Phong Model

- ***Specular reflection*** is the result of total, or near total, reflection of the incident light in a concentrated region around the ***specular-reflection angle.***
- Shiny surfaces have a narrow specular-reflection range.
- Dull surfaces have a wider reflection range.



Specular Reflection

Figure 13 shows the specular reflection direction at a point on the illuminated surface. In this figure,

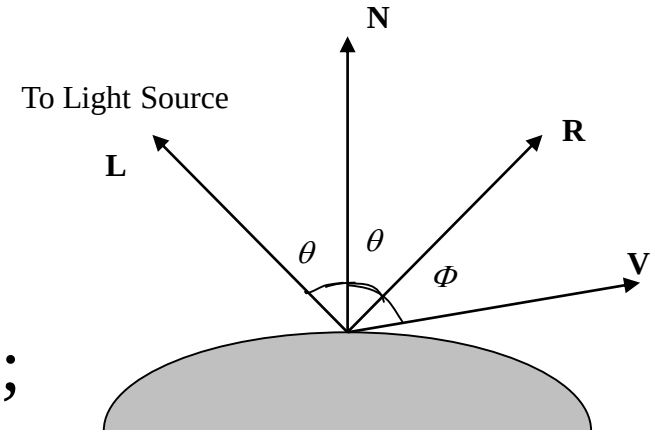


Fig. 13
Modeling specular reflection.

- **R** represents the unit vector in the direction of specular reflection;
- **L** – unit vector directed toward the point light source;
- **V** – unit vector pointing to the viewer from the surface position;
- Angle Φ is the viewing angle relative to the specular-reflection direction **R**.

Specular Reflection - Example

Fig. 18
Phong shading polygons with specular reflection.

Combine Diffuse & Specular Reflections

For a single point light source, we can model the combined diffuse and specular reflections from a point on an illuminated surface as

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

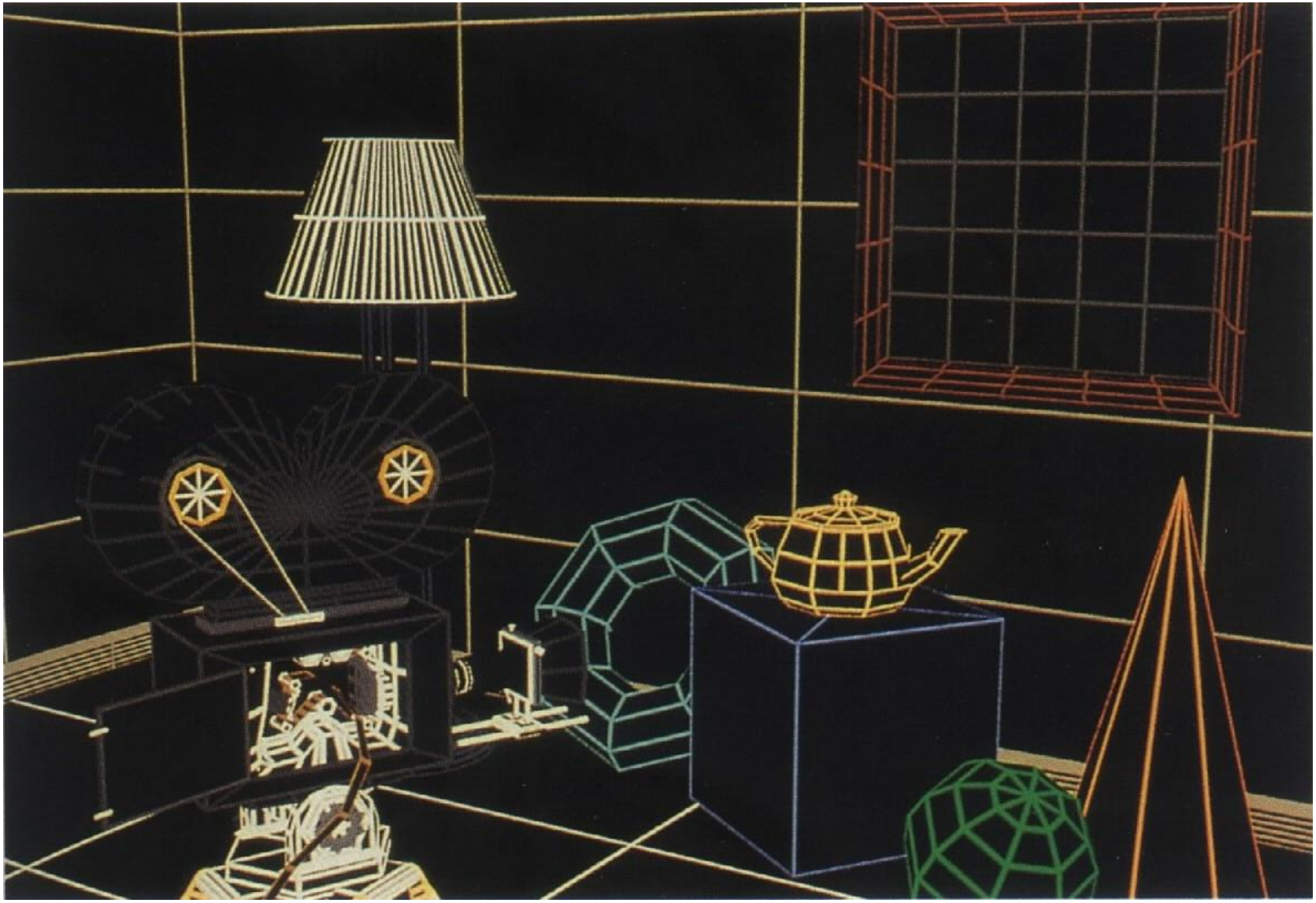
$$= k_a I_a + k_d I_l(\mathbf{N}\cdot\mathbf{L}) + k_s I_l(\mathbf{N}\cdot\mathbf{H})^{ns}$$

Combine Diffuse & Specular Reflections with Multiple Light Sources

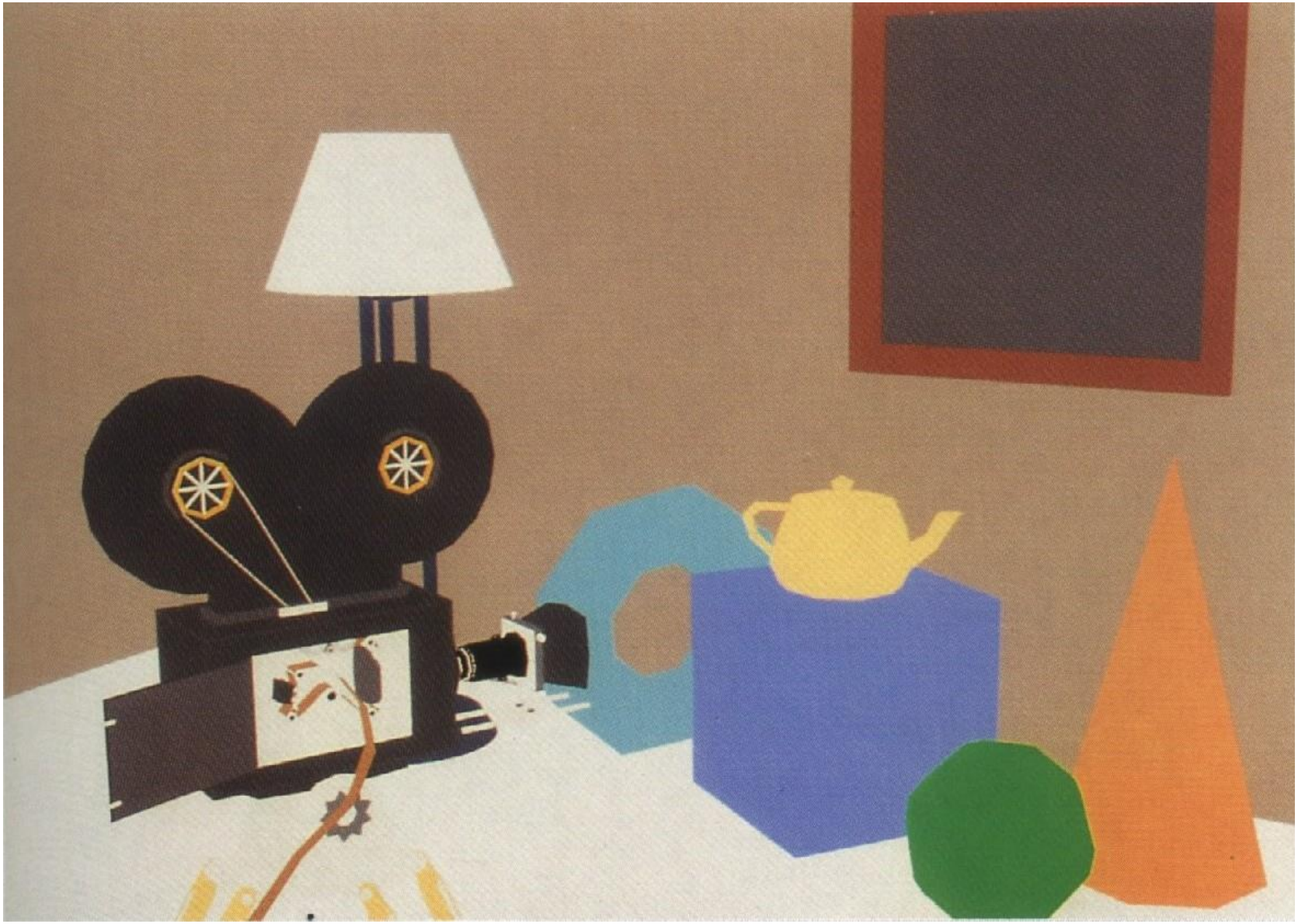
If we place more than one point source in a scene, we obtain the light reflection at any surface point by summering the contributions from the individual sources:

$$I = k_a I_a + \Sigma^{n}_{i=1} I_{li} [k_d (\mathbf{N} \cdot \mathbf{L_i}) + k_s (\mathbf{N} \cdot \mathbf{H_i})^{ns}]$$

Visible-line determination



Visible-surface determination with ambient illumination only



Individually shaded polygons with diffuse reflection

Gouraud shaded polygons with diffuse reflection

Gouraud shaded polygons with specular reflection

Phong shaded polygons with specular reflection

Curved surfaces with specular reflection

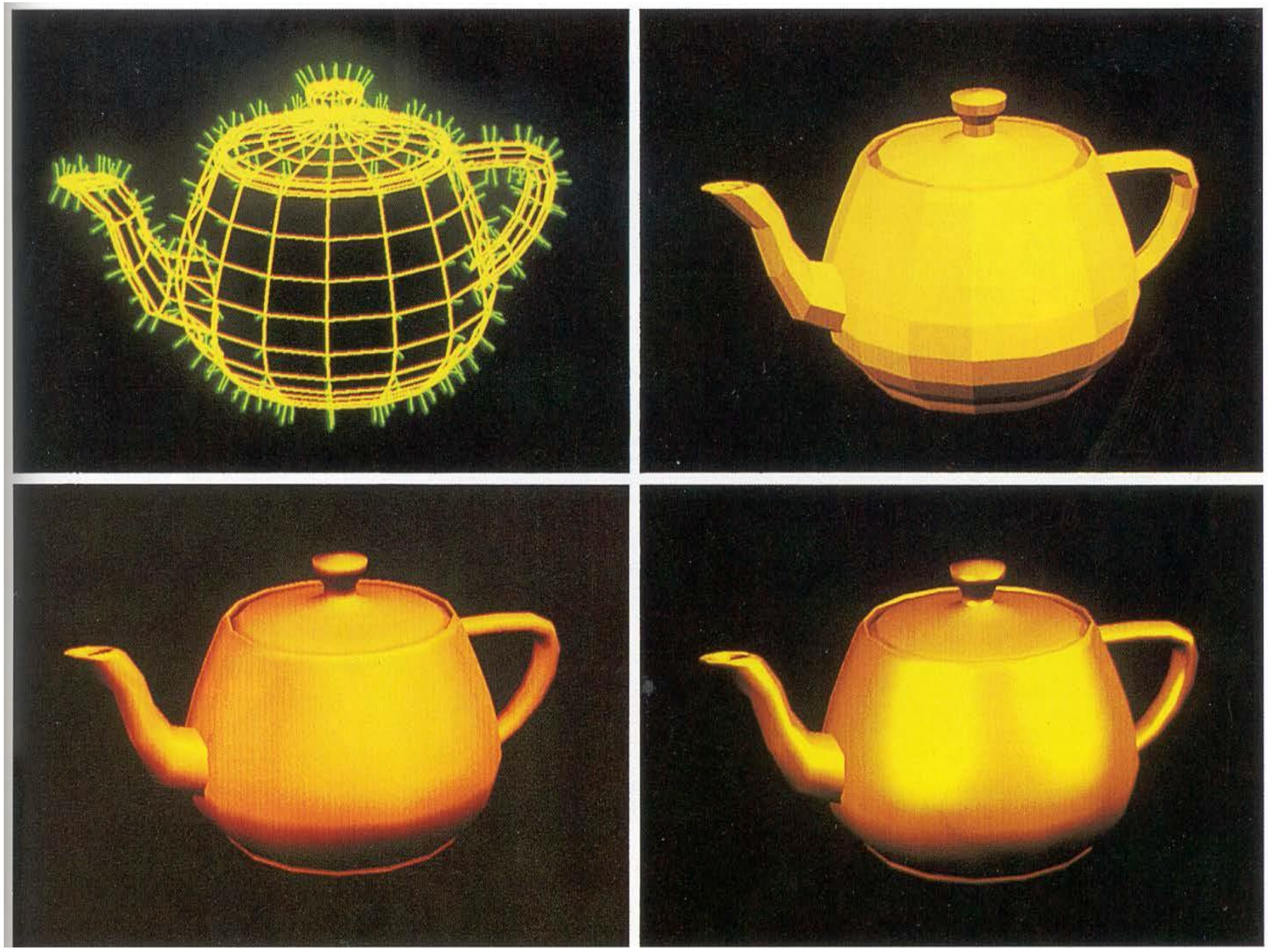
Polygon rendering methods

- Here we consider the application of an illumination model to the rendering of standard graphics objects those formed with polygon surfaces.
- Polygon Shading Algorithms
- Wireframe
- Flat
- Gouraud
- phong

Wireframe

- A **wire-frame model** is a visual presentation of a 3-dimensional (3D) or physical object used in 3D computer graphics. It is created by specifying each edge of the physical object where two mathematically continuous smooth surfaces meet, or by connecting an object's constituent vertices using straight lines or curves.
- The object is projected into screen space by drawing lines at the location of each edge. The term wire frame comes from designers using metal wire to represent the three-dimensional shape of solid objects. 3D wire frame allows to construct and manipulate solids and solid surfaces. The 3D solid modeling technique efficiently draws higher quality representations of solids than the conventional line drawing.

- Using a wire-frame model allows visualization of the underlying design structure of a 3D model.
- Traditional two-dimensional views and drawings can be created by appropriate rotation of the object and selection of hidden line removal via cutting planes.
- Since wire-frame renderings are relatively simple and fast to calculate, they are often used in cases where a high screen frame rate is needed.
- This allows the designer to quickly review solids or rotate the object to new desired views without long delays associated with more realistic rendering.



Shading

- **“ Shading is referred as the *implementation of illumination model at the pixel points or* polygon surfaces of the graphics objects.”**

Flat Shading

- A fast and simple method for rendering an object with polygon surfaces is constant intensity shading, also called flat shading.
- In this method, a single intensity is calculated for each polygon All points over the surface of the polygon are then displayed with the same intensity value.
- Constant shading can be useful for quickly displaying the general appearance of a curved surface, as in Fig

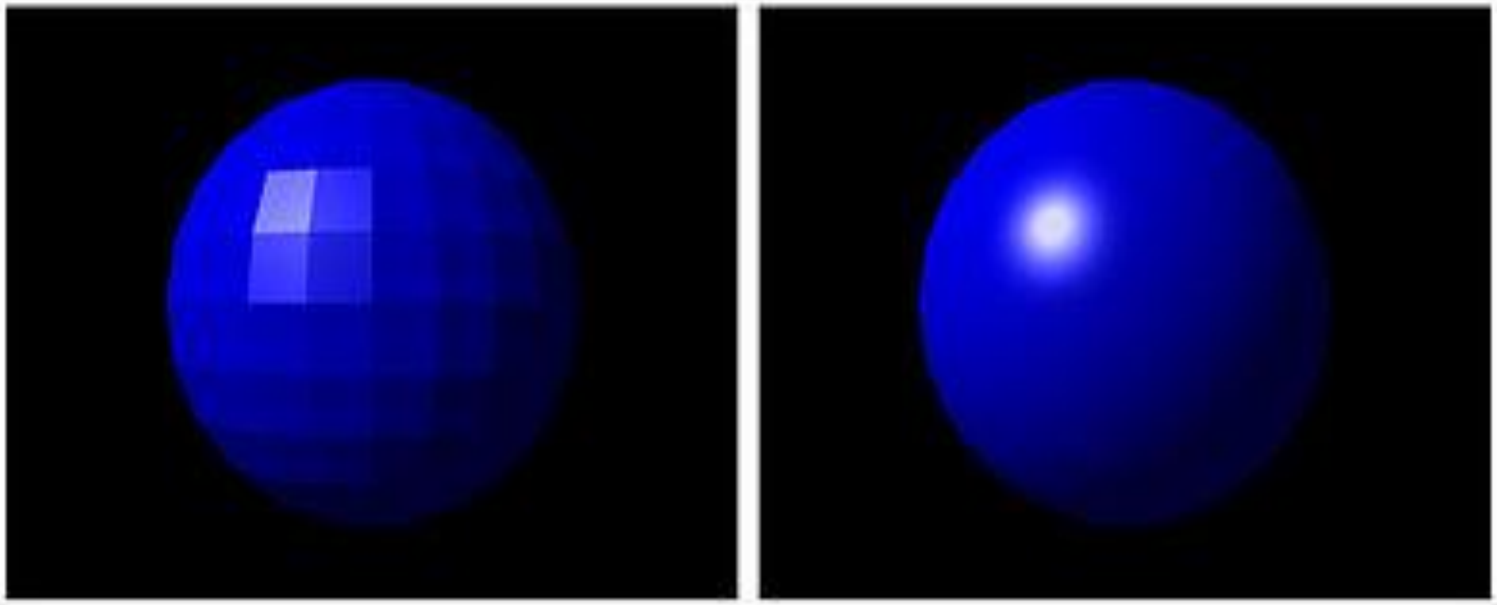
- Fig(1). A polygon mesh approximation of an object (a) is rendered with flat shading (b) and With Gouraud shading *(c)*.

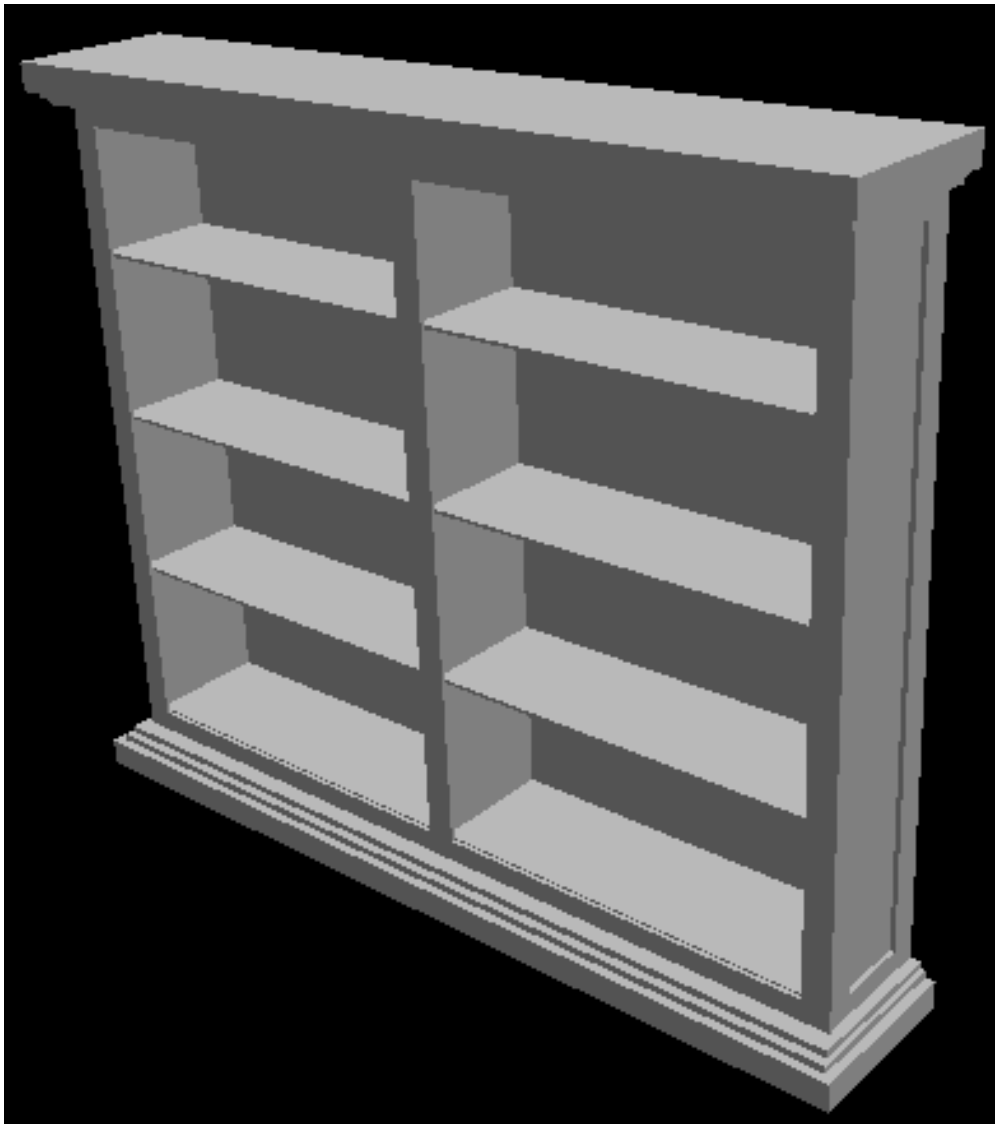
- In general, flat shading of polygon facets provides an accurate rendering for an object if all of the following assumptions are valid: -

- 1. The object is a polyhedron and is not an approximation of an object with a curved surface.
- 2. All light sources illuminating the objects are sufficiently far from the surface so that N . L and the attenuation function are constant over the surface(where N is the unit normal to a surface and L is the unit direction vector to the point light source from a position on the surface).

- 3. The viewing position is sufficiently far from the surface so that V .R is constant over the surface.(where V is the unit vector pointing to the viewer from the surface position and R represent unit vector in the direction of ideal specular reflection).

- : If there happens to be a large specular component at the representative vertex, that brightness is drawn uniformly over the entire face. If a specular highlight doesn't fall on the representative point, it is missed entirely. Consequently, the specular reflection component is usually not included in flat shading computation.

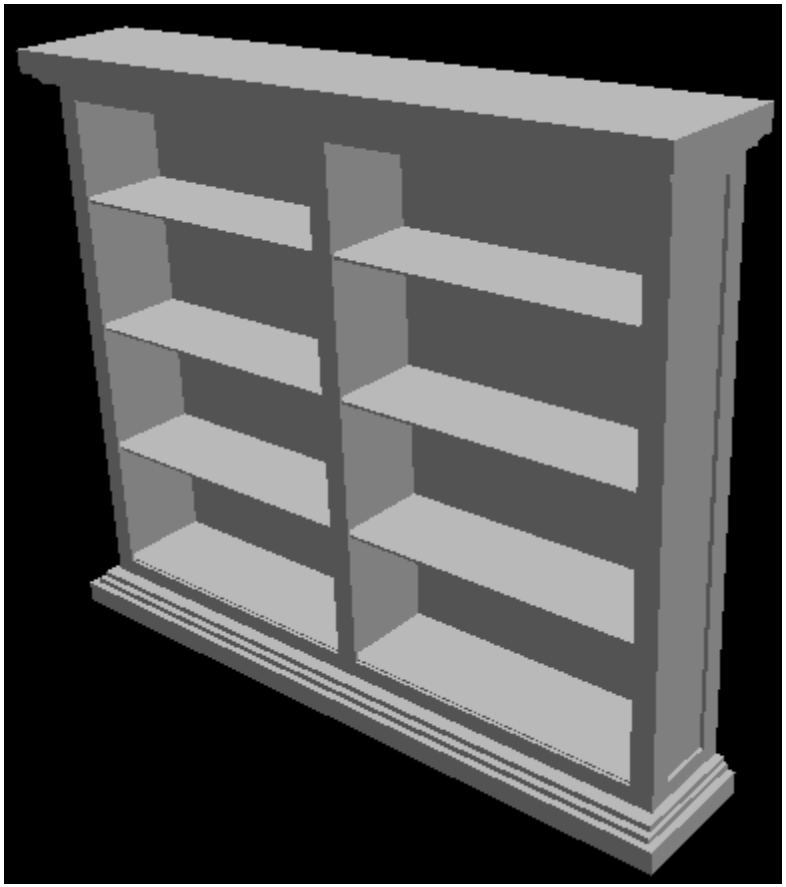


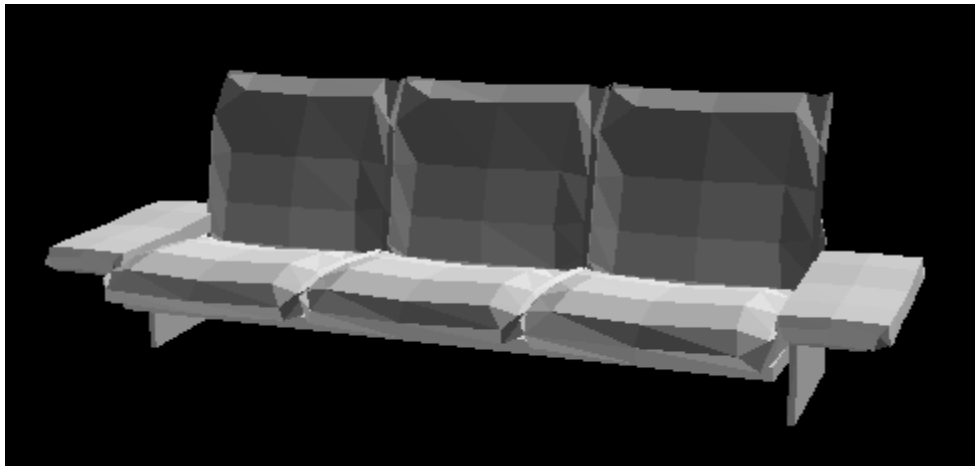


- Objects look like they are composed of polygons

  OK for polyhedral objects

  Not so good for ones with smooth surfaces





Smooth Shading

- In contrast to flat shading with smooth shading the color changes from pixel to pixel. It assumes that the surfaces are curved and uses interpolation techniques to calculate the values of pixels between the vertices of the polygons.
- Types of smooth shading include:
- phong Shading
- Gouraud Shading

Gouraud Shading

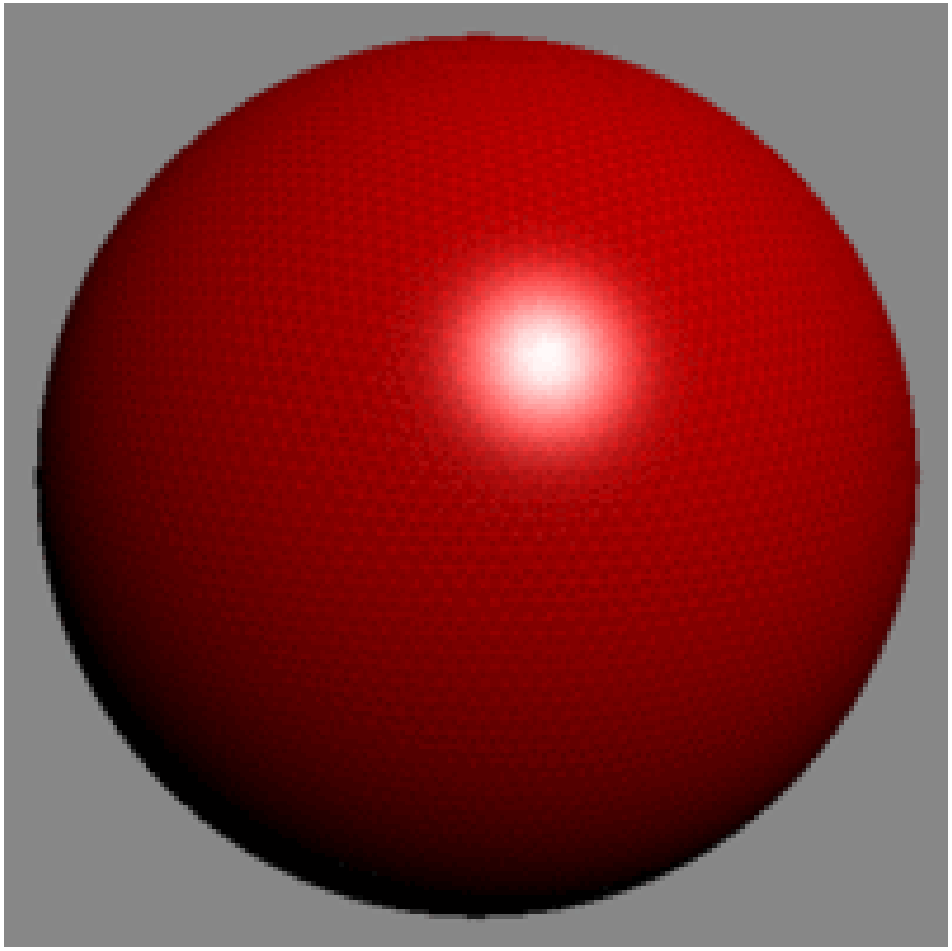
- This intensity-interpolation scheme, developed by Gouraud and generally referred to as Gouraud shading, renders a polygon surface by linearly interpolating intensity values across the surface.
- Intensity values for each polygon are matched with the values of adjacent polygons along the common edges, thus eliminating the intensity discontinuities that can occur in flat shading.
- Each polygon surface is rendered with Gouraud shading by performing the following calculations:

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

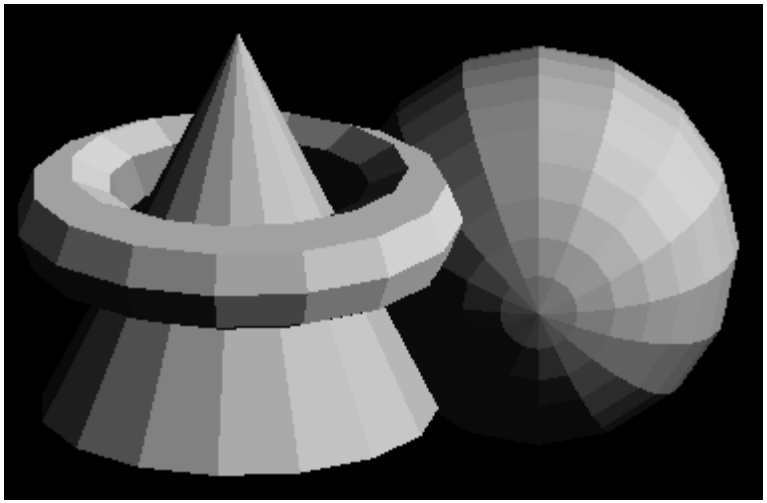
  2. Apply an illumination model to each vertex to calculate the vertex intensity.

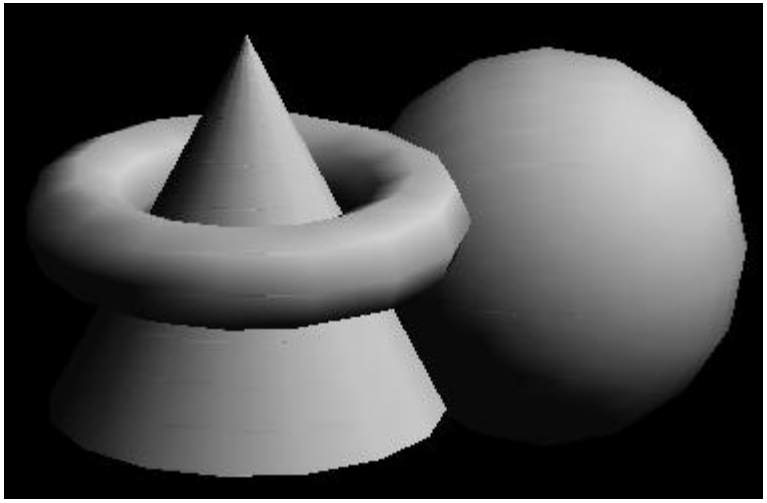
  3. Linearly interpolate the vertex intensities over the surface of the polygon.

- **Advantages**
- Polygons, more complex than triangles, can also have different colors specified for each vertex. In these instances, the underlying logic for shading can become more intricate.
- **Problems**
- Even the smoothness introduced by Gouraud shading may not prevent the appearance of the shading differences between adjacent polygons.
- Gouraud shading is more CPU intensive and can become a problem when rendering real time environments with many polygons.
- T-Junctions with adjoining polygons can sometimes result in visual anomalies. In general, T-Junctions should be avoided.



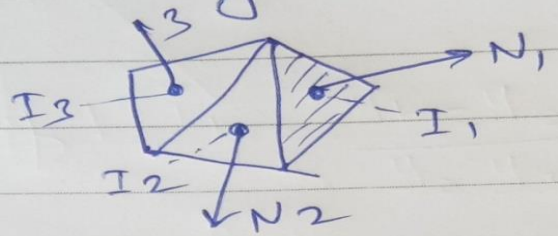
Polygon Smooth Shading





Gouraud shading :- (smooth shading)

1) Flat shading - (constant intensity shading)

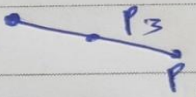


Draw unit normal vector for each face of given object - (i.e. N_1, N_2, N_3) and calculated intensity of anyone point for given surfaces (i.e. I_1, I_2 and I_3). shade the surface of the object by calculated Intensity value.

→ problem → discountinity.

2) Gaurand shading :-

* It uses Intensity Interpolation method.
* Interpolation means to calculate intensity of middle point.

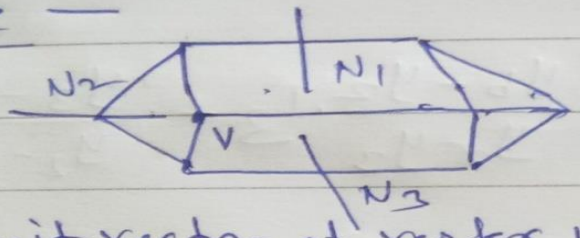
eg. 

3 steps to calculate Intensity using Interpolation

* Determine avg. unit normal vector at each vertex of polygon.
* Apply Illumination model to calculate Intensity of vertex.

* After calculating Intensity Apply Interpolation method to calculate Intensity of middle point

example —



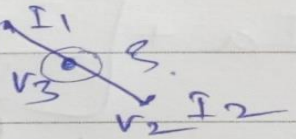
(Normal unit vector at vertex V)

1) $N_v = \frac{N_1 + N_2 + N_3}{|N_1 + N_2 + N_3|}$

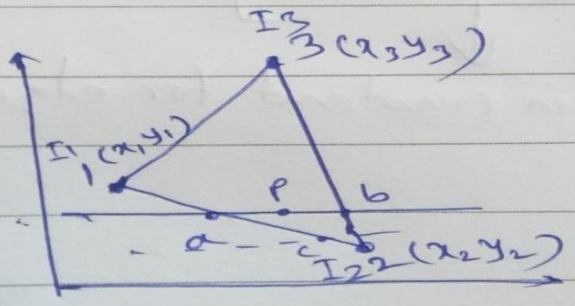
$= \frac{\sum_{i=1}^{n} N_i}{\left|\sum_{i=1}^{n} N_i\right|}$

n = no. of surfaces sharing the vertex V.

2) Apply illumination model to calculate intensity of each vertex. ∀.



3)



we have to calculate intensities of a, P, & b. by applying Intensity Interpolation method.

Intensity of a $I_a = \frac{y_a - y_2}{y_1 - y_2} \cdot I_1 + \frac{y_1 - y_a}{y_1 - y_2} \cdot I_2$

Intensity of b $I_b = \frac{y_b - y_2}{y_2 - y_3} \cdot I_3 + \frac{y_3 - y_b}{y_2 - y_3} \cdot I_2$

Intensity of p $I_p = \frac{x_b - x_p}{x_b - x_a} \cdot I_a + \frac{x_p - x_a}{x_b - x_a} \cdot I_b$

---

to Reduce the computations and to minimize the calculations to calculate intensity of another pixel. We can use the Intensity values of previous pixels.

example.

$$I_c = I_a + \left[\frac{I_2 - I_1}{y_1 - y_2}\right]$$

↓

this is constant for edge 12

Phong shading

- A more accurate method for rendering a polygon surfacevis to interpolate normal vectors, and then apply the illumination model to each surface point. This method, developed by Phong Bu Tuong, is called Phong shading, or normal vector interpolation shading.
- It displays more realistic highlights on a surface and greatly reduces the Mach-band effect. A polygon surface is rendered using Phong shading by carrying out the following steps:

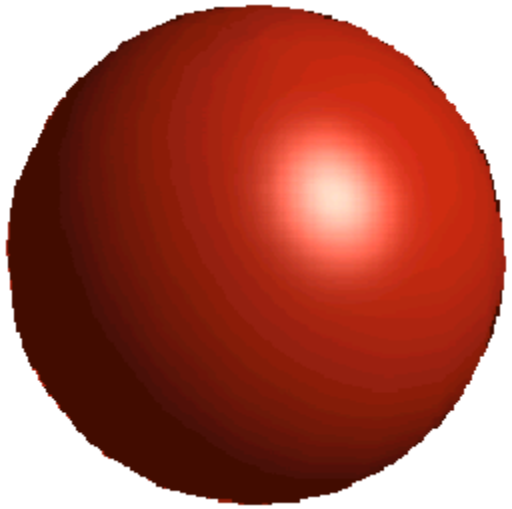
  1. Determine the average unit normal vector at each polygon vertex.

  2. Linearly & interpolate the vertex normals over the surface of the polygon.

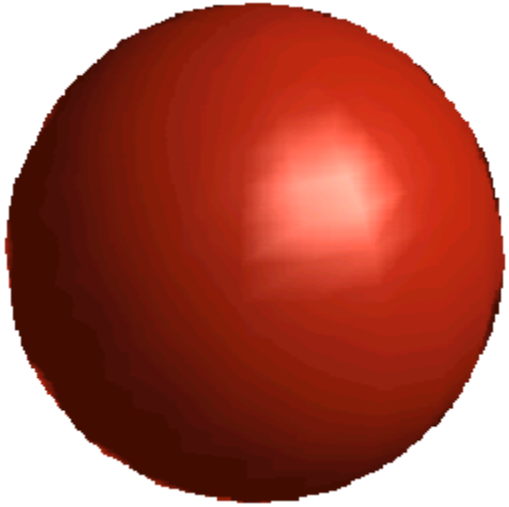
  3. Apply an illumination model along each scan line to calculate projected pixel intensities for the surface points.

Phong Shading

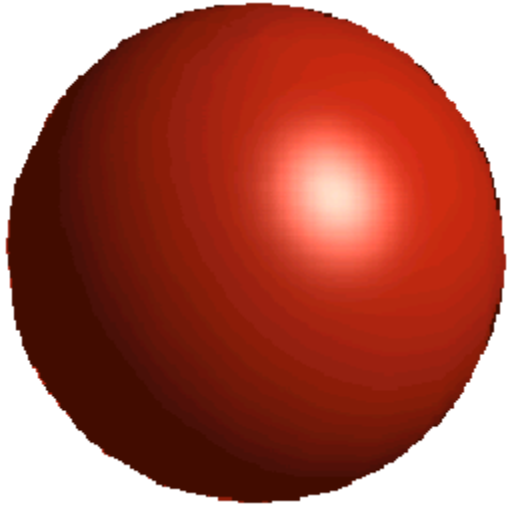
- A more accurate method for rendering a polygon surface is to interpolate normal vectors, and then apply the illumination model to each surface point.



Flat

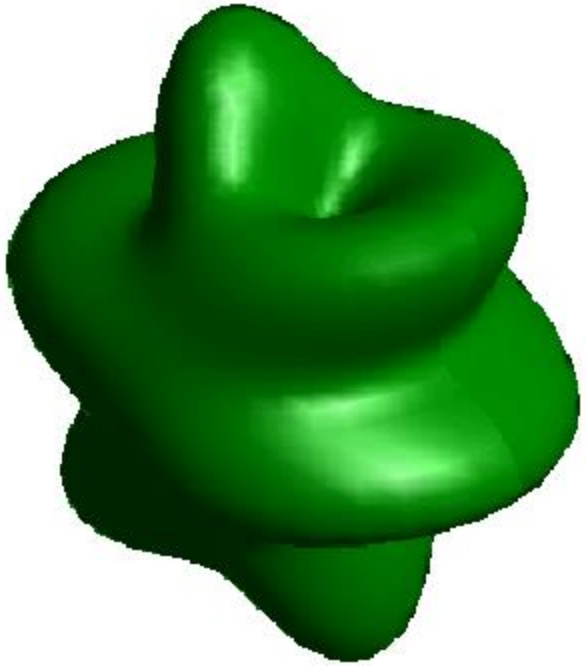


Gouraud

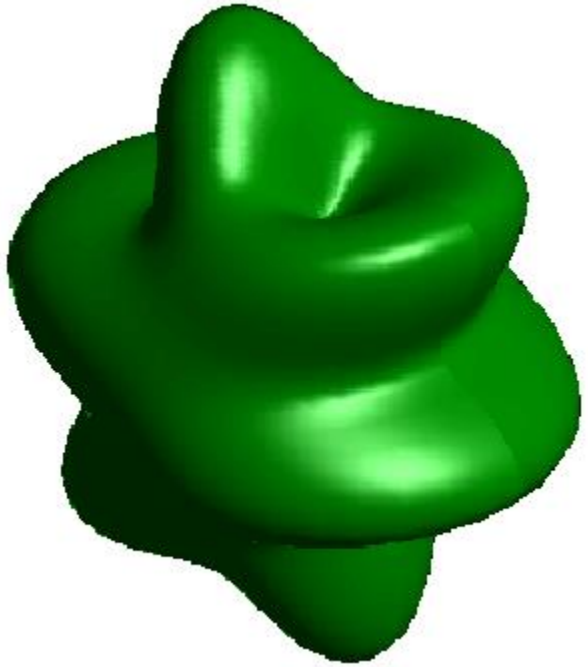


Phong

Flat shading



Gouraud shading



Phong shading

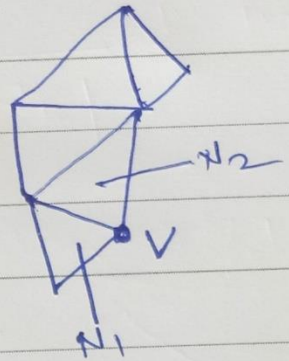
Phong shading :- (Normal vector Interpolation shading)

3 steps to calculate phong shading Intensity

- Determine avg. unit normal vector at each vertex of polygon.
- Linearly Interpolate the vertex normals over the surface of polygon
- Apply illumination model to calculate intensity of vertex.

example :-

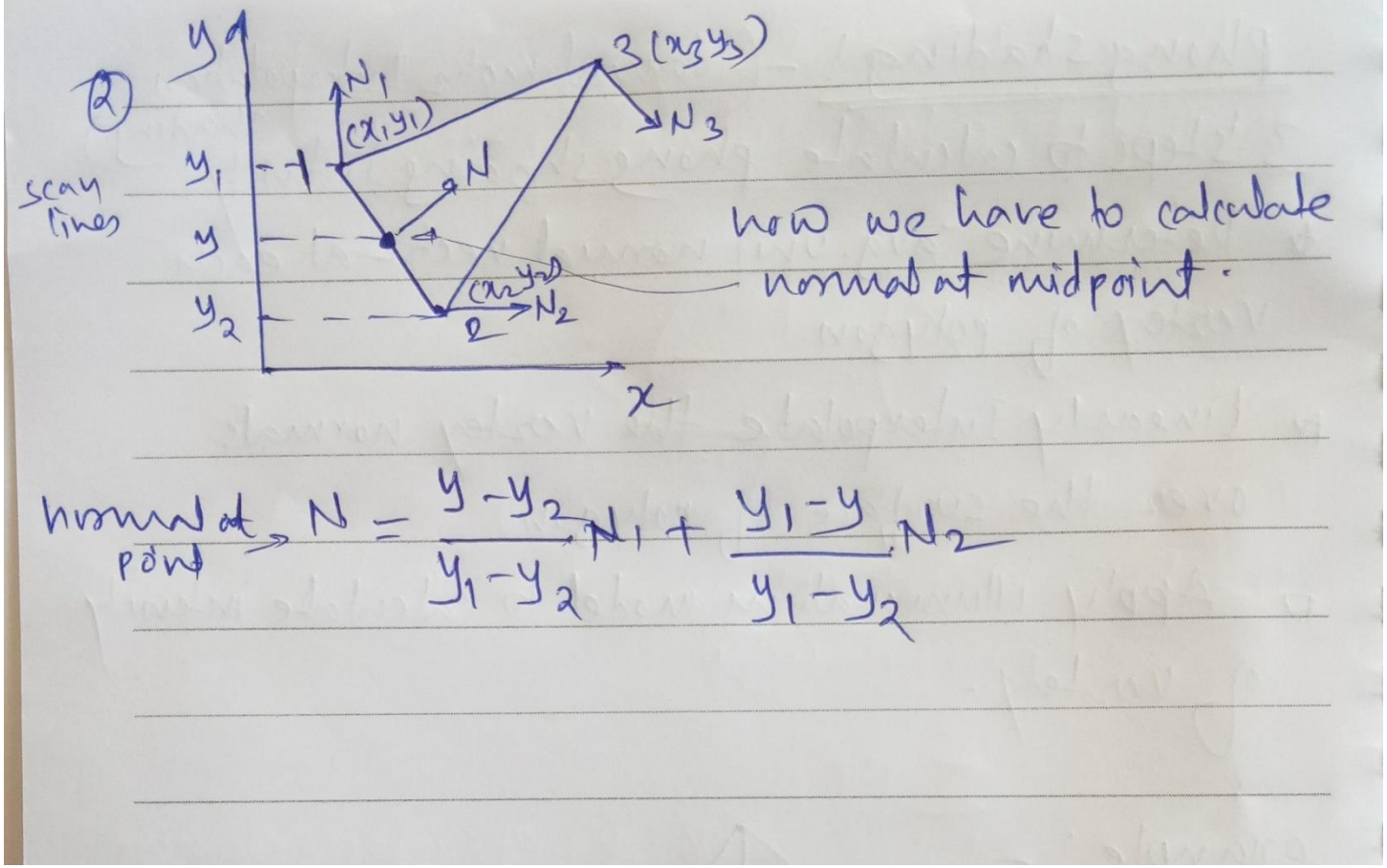
①



$$N_V = \frac{N_1 + N_2}{|N_1 + N_2|}$$

$$N_V = \frac{\sum_{i=1}^{n} N_i}{\left| \sum_{i=1}^{n} N_i \right|}$$

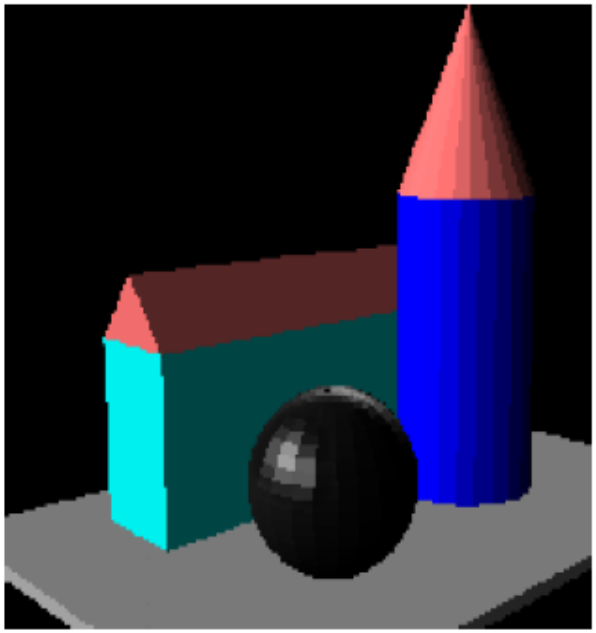
n = no. of surfaces sharing vertex.



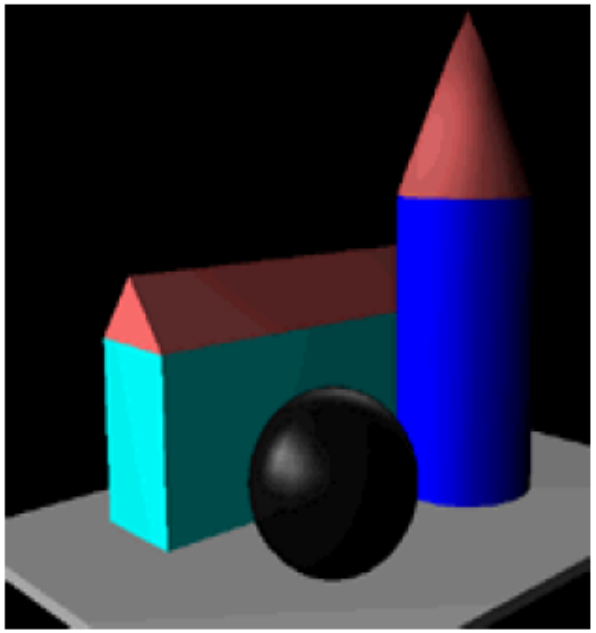
normal at point $\rightarrow N = \frac{y - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y}{y_1 - y_2} N_2$

Comparision

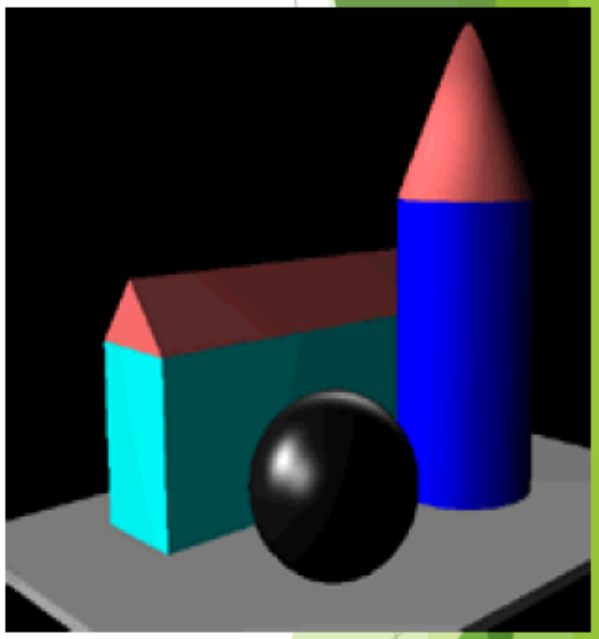
- Flat shading
  - The simplest shading method
- Difference of two shading models
- Phong shading is more accurate way of shading a polygon since the illumination model is applied to every point
- More computationally intensive than the Gouraud
- Illumination model is applied more often
- Interpolated normals need to be normalized



a) Flat shading



b) Gouraud shading

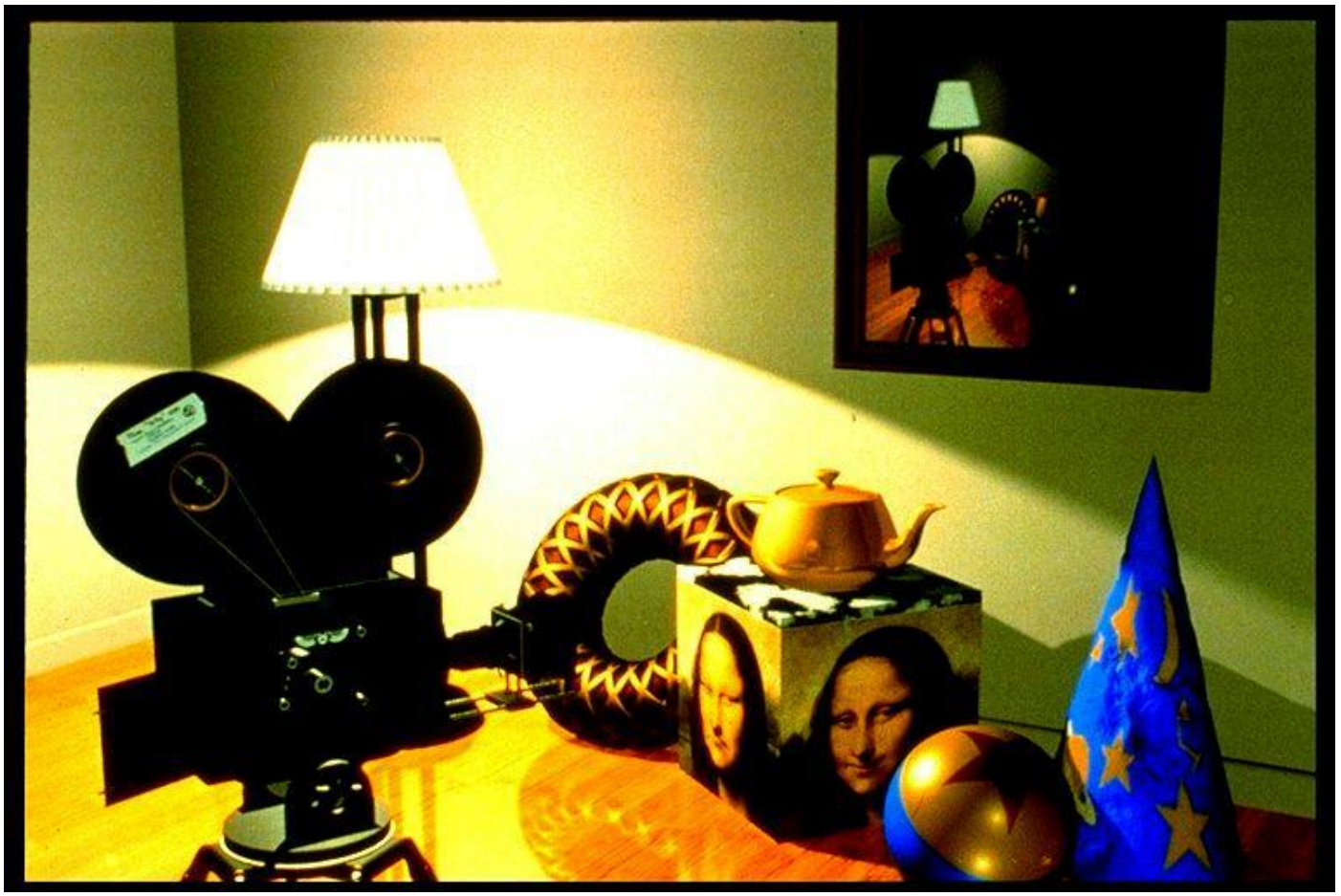


c) Phong shading

Ray Tracing!

- In **computer graphics**, **ray tracing** is a technique for generating an image by **tracing** the path of light through pixels in an image plane and simulating the effects of its encounters with virtual objects.

- Better method, can show these realistic effects.



Ray Tracing Method

- In computer graphics, **ray tracing** is a technique for generating an image by tracing the path of light through pixels in an image plane and simulating the effects of its encounters with virtual objects. The technique is capable of producing a very high degree of visual realism, usually higher than that of typical scan line rendering methods.
- Ray tracing is a technique for rendering three-dimensional graphics with very complex light interactions. This means you can create pictures full of mirrors, transparent surfaces, and shadows, with stunning results.

basic terminology

- When creating any sort of computer graphics, you must have a list of objects that you want your software to render. These objects are part of a **scene** or **world.**
- so when we talk about ``looking at the world," we're not posing a philosophical challenge, but just referring to the ray tracer drawing the objects from a given viewpoint.
- In graphics, this viewpoint is called the **eye** or **camera**. Following this camera analogy, just like a camera needs film onto which the scene is projected and recorded, in graphics we have a **view window** on which we draw the scene.
- .

- The difference is that while in cameras the film is placed *behind* the aperture or focal point, in graphics the view window is in *front* of the focal point.
- So the color of each point on real film is caused by a light ray (actually, a group of rays) that passes through the aperture and hits the film, while in computer graphics each pixel of the final image is caused by a simulated light ray that hits the view window on its path towards the eye. The results, however, are the same.

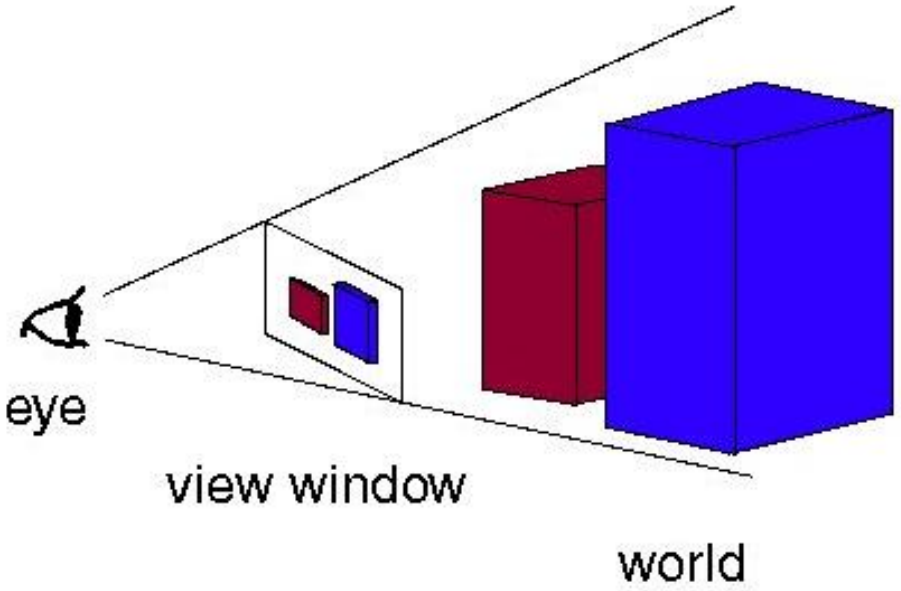
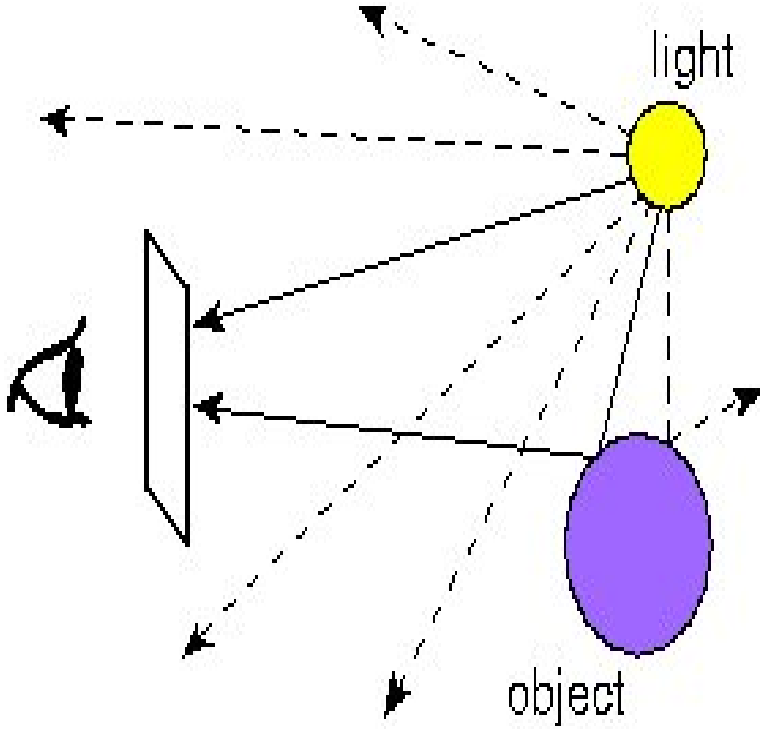


Figure 2. The eye, view window, and world.

- our goal is find the color of each point on the view window. We subdivide the view window into small squares, where each square corresponds to one pixel in the final image.
- If you want to create an image at the resolution of 640x400, you would break up the view window into a grid of 640 squares across and 400 square down. The real problem, then, is assigning a color to each square. This is what ray tracing does.

How it works

- Ray tracing is so named because it tries to simulate the path that light rays take as they bounce around within the world - they are *traced* through the scene.
- The objective is to determine the color of each light ray that strikes the view window before reaching the eye. A light ray can best be thought of as a single photon.
- The name ``ray tracing'' is a bit misleading because the natural assumption would be that rays are traced starting at their point of origin, the light source, and towards their destination, the eye .



. Tracing rays from the light source to the eye. Lots of rays are wasted because they never reach the eye.

- This would be an accurate way to do it, but unfortunately it tends to be very difficult due to the sheer numbers involved.
- Consider tracing one ray in this manner through a scene with one light and one object, such as a table. We begin at the light bulb, but first need to decide how many rays to shoot out from the bulb.
- Then for each ray we have to decide in what direction it is going. There is really an infinity of directions in which it can travel - how do we know which to choose?
- Let's say we've answered these questions and are now tracing a number of photons.
- Some will reach the eye directly, others will bounce around some and then reach the eye, and many, many more will probably never hit the eye at all. For all the rays that never reach the eye, the effort tracing them was wasted.

- In order to save ourselves this wasted effort, we trace only those rays that are *guaranteed* to hit the view window and reach the eye.
- After all, any given ray can bounce around the room many times before reaching the eye. However, if we look at the problem *backwards*, we see that it has a very simple solution.
- Instead of tracing the rays starting at the light source, we trace them backwards, starting at the eye.
- Consider any point on the view window whose color we're trying to determine. Its color is given by the color of the light ray that passes through that point on the view window and reaches the eye.
- We can just as well follow the ray backwards by starting at the eye and passing through the point on its way out into the scene.

- The two rays will be identical, except for their direction:
- if the original ray came directly *from* the light source, then the backwards ray will go directly *to* the light source;
- if the original bounced off a table first, the backwards ray will also bounce off the table.
- You can see this by looking at Figure 3 again and just reversing the directions of the arrows. So the backwards method does the same thing as the original method, except it doesn't waste any effort on rays that never reach the eye.
- how ray tracing works in computer graphics. For each pixel on the view window, we define a ray that extends from the eye to that point. We follow this ray out into the scene and as it bounces off of different objects. The final color of the ray (and therefore of the corresponding pixel) is given by the colors of the objects hit by the ray as it travels through the scene.

Just as in the light-source-to-eye method it might take a very large number of bounces before the ray ever hits the eye, in backwards method it might take many bounces before the ray every hits the light.
Since we need to establish some limit on the number of bounces to follow the ray on, we make the following approximation: every time a ray hits an object, we follow a single new ray from the point of intersection *directly towards* the light source

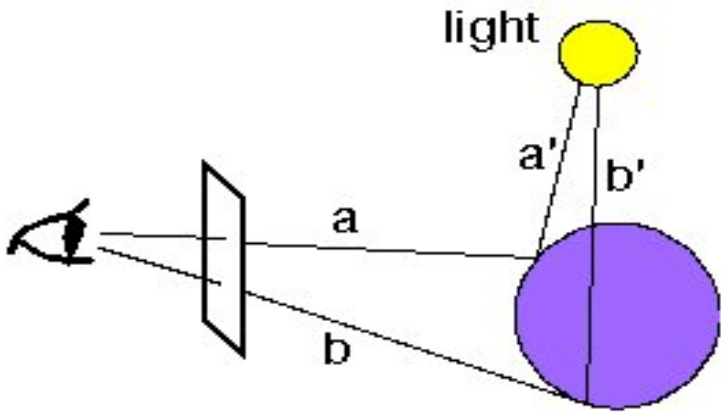
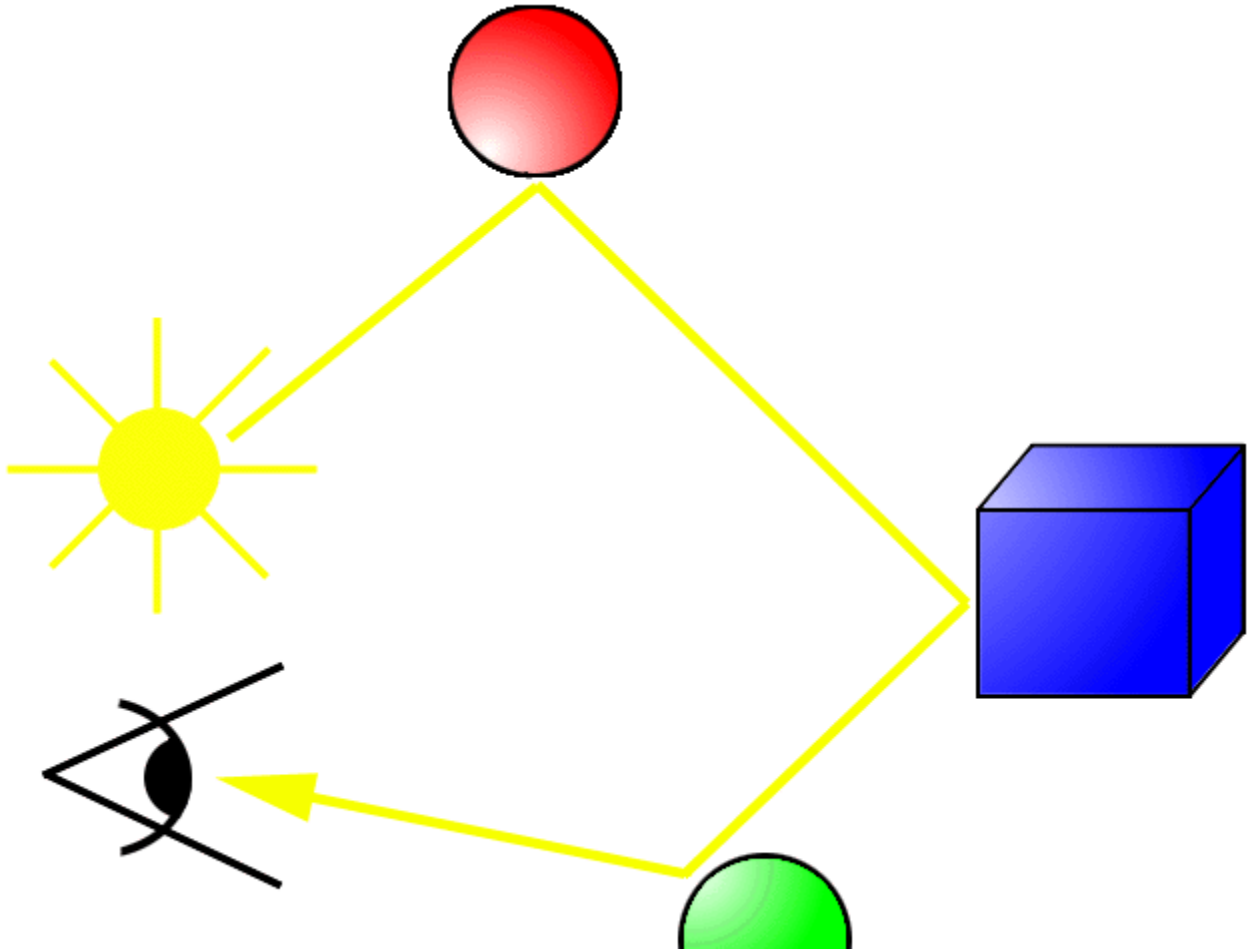


Figure 4. We trace a new ray from each ray-object intersection directly towards the light source

Types of Ray Tracing

- Forward Ray Tracing
  - Forward ray tracing follows the light particles (photons) from the light source to the object.
  - Although forward ray tracing can most accurately determine the coloring of each object, it is highly inefficient.
  - This is because many rays from the light source never come through the view plane and into the eye.
  - Tracking every light ray from the light source down means that many rays will go to waste because they never contribute to the final image as seen from the eye.
  - Forward ray tracing is also known as light ray tracing and photon tracing.



- Backward Ray Tracing
  - To make ray tracing more efficient, the method of backward ray tracing is introduced.
  - In backward ray tracing, an eye ray is created at the eye; it passes through the view plane and on into the world.
  - The first object the eye ray hits is the object that will be visible from that point of the view plane.
  - After the ray tracer allows that light ray to bounce around, it figures out the exact coloring and shading of that point in the view plane and displays it on the corresponding pixel on the computer monitor screen.

- Hybrid Ray Tracing:-
  - Since both forward ray tracing and backward ray tracing have their drawbacks, recent research tries to develop hybrid solutions that will compromise speed and accuracy.
  - In these hybrid solutions, only certain levels of forward ray tracing is performed.
  - The algorithm records the data, then goes on to perform backward ray tracing.
  - The final coloring of the scene takes both the backward ray tracing and the forward ray tracing calculations into account.


Ray Tracing

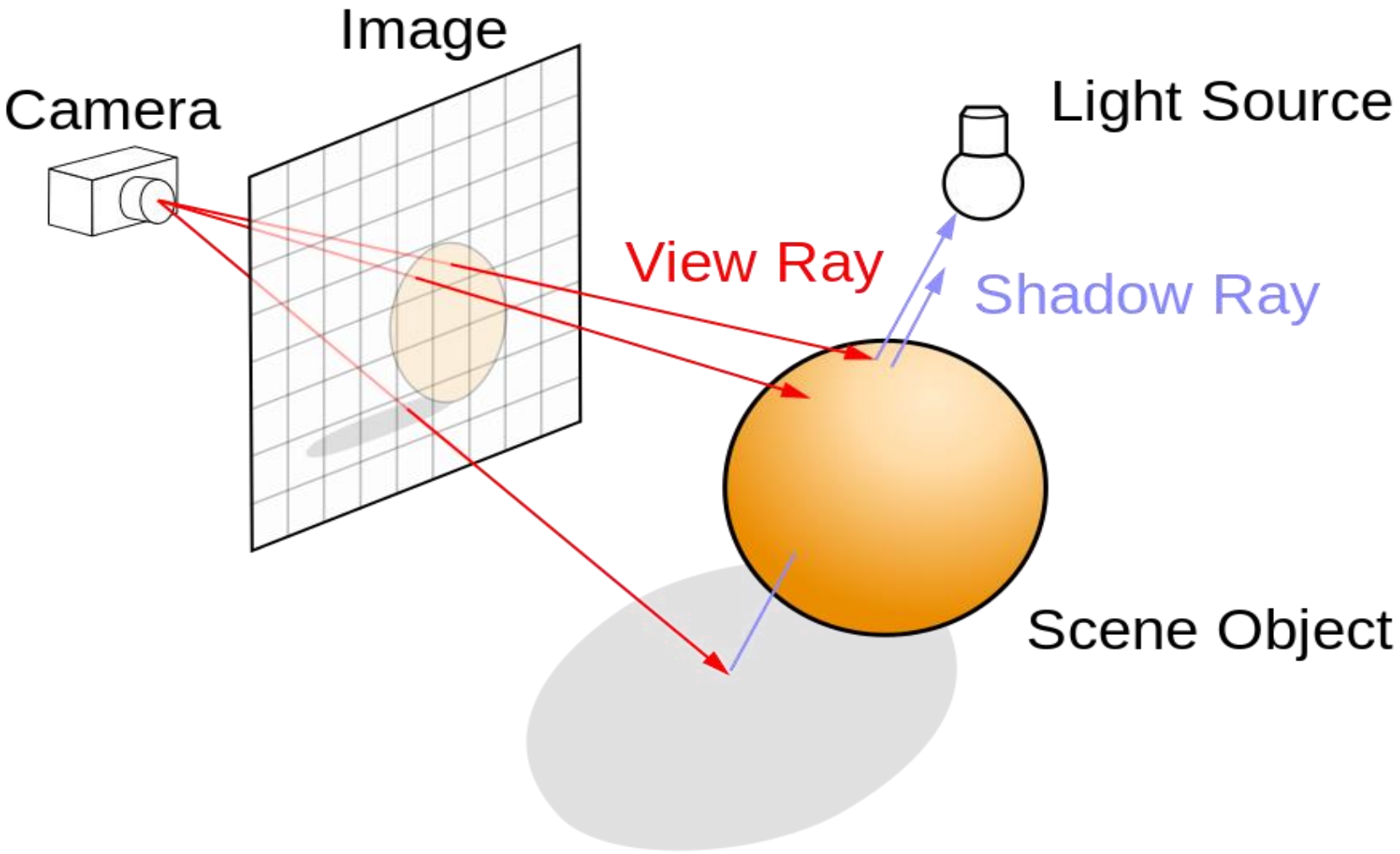


Image
Camera
Light Source
View Ray
Shadow Ray
Scene Object