Illumination

Illumination

- Illumination, an observable property and effect of light, may also refer to:

  - Illumination (lighting), the use of light sources.
  - Illumination (image), the use of light and shadow in art.
  - Global illumination,algorithms used in computer graphics to add realistic lighting to 3D scenes

Illumination (Lighting) model

- Expresses the factors determining a surface's color at a given point (defines the nature of the light emanating from a source – geometry of light distribution, etc.)

- Local Illumination
  - Defines single light and single surface interaction
- Global Illumination
  - Defines interchange of lights between all surfaces

Illumination Model Parameters

- Lighting effects are described with models that consider the interaction of light sources with object surfaces
- The factors determining the lighting effects are:
  - The light source parameters:
    - Positions
    - Electromagnetic Spectrum
    - Shape
  - The surface parameters
    - Position
    - Reflectance properties
    - Position of nearby surfaces
  - The eye (camera) parameters
    - Position
    - Sensor spectrum sensitivities

Local Illumination

Most such adhoc illumination models have three components

$$I = \text{ambient} + \text{diffuse} + \text{specular}$$

Ambient reflection

- Independent of object position and viewer position
- Constant
- Exists in most environments
  – some light hits surface from all directions
  –a way of approximating contributions from indirect lighting

Local Illumination

- A surface that is not exposed directly to a light source will be still visible if the nearby objects are illuminated
- Ambient illumination has no spatial or directional characteristics
- The amount of ambient light incident on each object is a constant for all surfaces over all directions

- The amount of light reflected is independent of objects position and orientation. It depends on surface properties

Diffuse (Lambertian) reflection

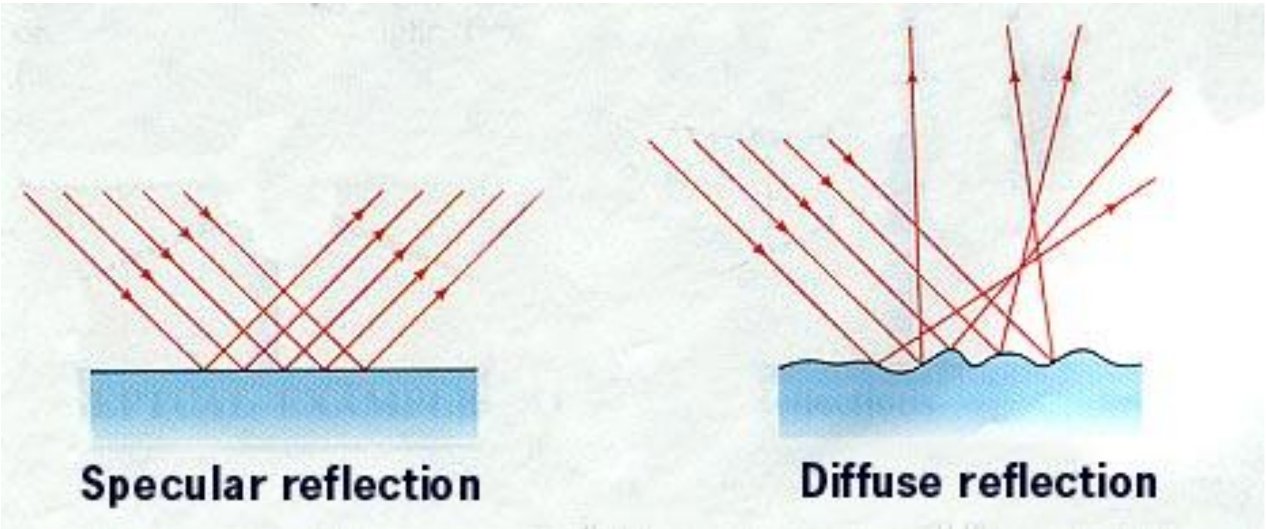
- Typical of dull matte surfaces
- Independent of viewer position
- Dependent of light source position (in this case a point source again a non-physical abstraction)

Specular reflection

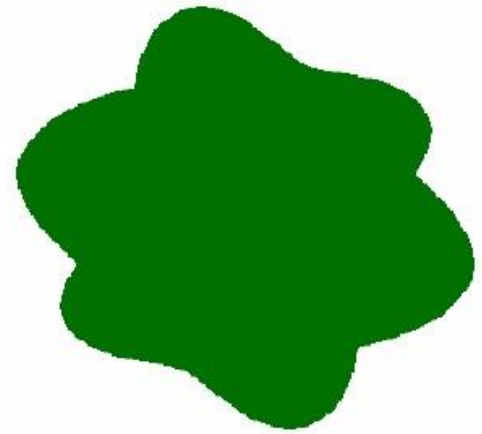
- Typical of bright and shiny surfaces
- Color depends on material and how it scatters light energy
- In plastics it is color of point source and in metal its the color of me tal
- In other materials will combine color of light source and color of m aterial
- Dependent on light source position <u>and</u> viewer position

Diffuse reflection vs Specular reflection

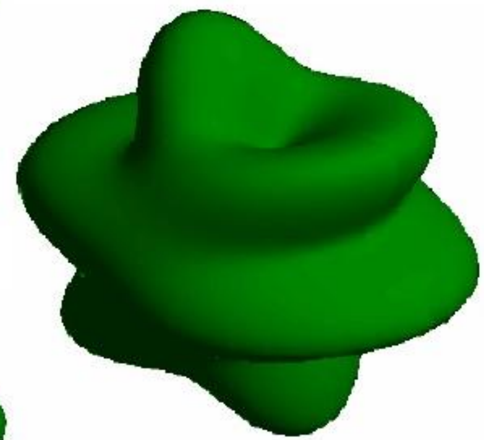
Specular Reflection	Diffuse Reflection
1. Reflection from a polished surface is called reegular reflection	1. Reflection from a rough surface is called diffuse reflection
2. Parallel rays remain after reflection	2. Parallel rays do not remain parallel after reflection



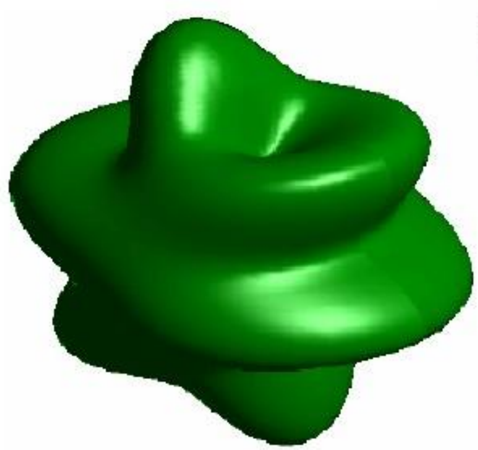
Example



Ambient Illumination



Ambient + Diffuse



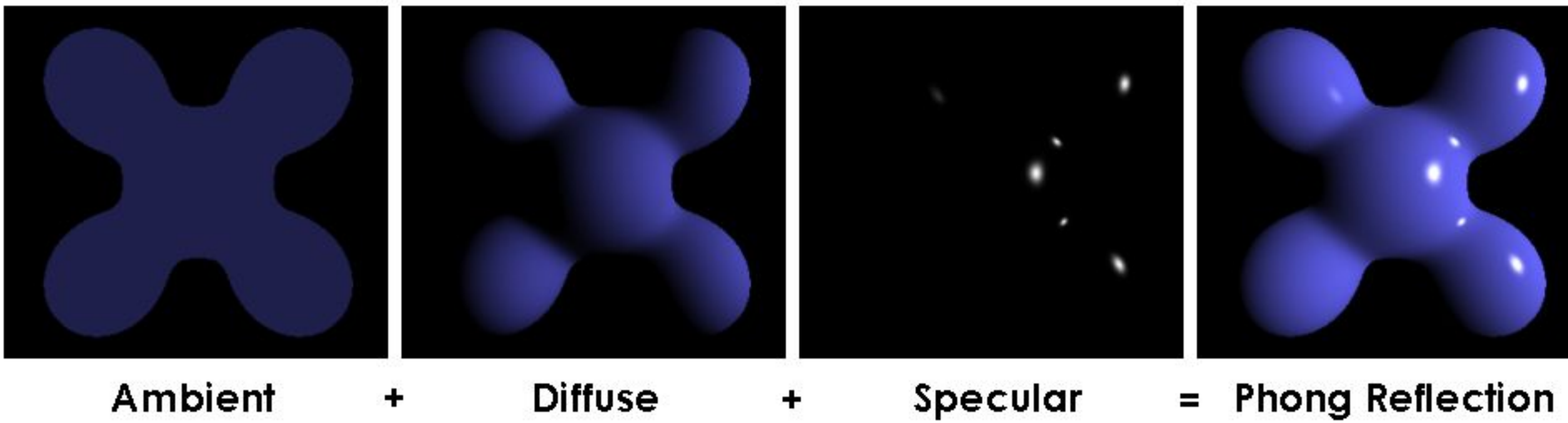
Ambient + Diffuse + Specular

Phong reflection model

- The Phong reflection model (also called Phong illumination or Phong lighting) is an empirical model of the local illumination of points on a surface.
- It describes the way a surface reflects light as a combination of the diffuse reflection of rough surfaces with the specular reflection of shiny surfaces.
- The model also includes an ambient term to account for the small amount of light that is scattered about the entire scene.

Phong Reflection…

- Visual illustration of the Phong equation:
- Here the light is white, the ambient and diffuse colors are both blue, and the specular color is white, reflecting a small part of the light hitting the surface, but only in very narrow highlights.
- The intensity of the diffuse component varies with the direction of the surface, and the ambient component is uniform (independent of direction).


Ambient + Diffuse + Specular = Phong Reflection

Locally illuminated test scene.

Ambient term only

Locally illuminated test scene.

Phong Model.

Ambient and Diffuse terms only

Locally illuminated test scene.

Phong Model.

Ambient, diffuse and Specular terms.

Global Illumination

- Extends the Local Illumination Model to include:
  - Reflection (one object in another)
  - Refraction (Snell’s Law)
  - Transparency (better model)
  - Shadows (at point, check each light source)
  - Antialiasing (usually means supersampling)

Global illumination.

- Two methods :
  - View dependent methods.
    - Calculate the view from the camera with global illumination.
    - Recursive ray-tracing.

  - View independent methods.
    - Solve lighting for the entire scene.
    - Radiosity solution.

Global illumination

- Global illumination or indirect illumination is a general name for a group of algorithms used in 3D computer graphics that are meant to add more realistic lighting to 3D scenes.

- Such algorithms take into account not only the light which comes directly from a light source (direct illumination), but also subsequent cases in which light rays from the same source are reflected by other surfaces in the scene, whether reflective or not (indirect illumination).

Global illumination…

- Theoretically reflections, refractions, and shadows are all examples of global illumination.
- Images rendered using global illumination algorithms often appear more photorealistic than images rendered using only direct illumination algorithms.
- However, such images are computationally more expensive and consequently much slower to generate.
- Radiosity, ray tracing, beam tracing, cone tracing, path tracing, Metropolis light transport, ambient occlusion, photon mapping, and image based lighting are examples of algorithms used in global illumination, some of which may be used together to yield results that are not fast, but accurate.

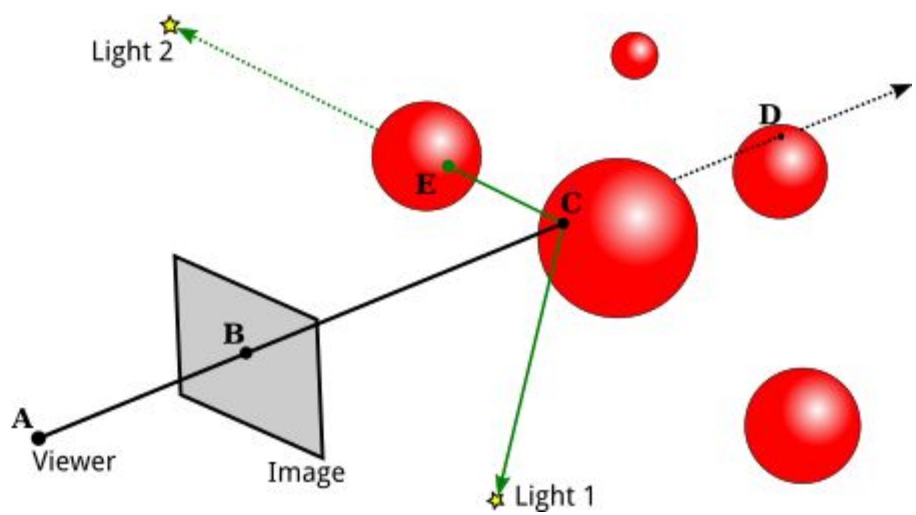
Ray Tracing

- Ray tracing is probably the best known technique for higher quality graphics.
- The idea behind it is not complicated: To find out what you see when you look in a given direction
- consider a ray of light that arrives at your location from that direction, and follow that light ray backwards to see where it came from.
- Or, as it is usually phrased, cast a ray from your location in a given direction, and see what it hits. That's what you see when you look in that direction.
- The operation of determining what is hit by a ray is called ray casting. It is fundamental to ray tracing and to other advanced graphics techniques.

Ray Casting

- A Raycaster takes an initial point and a direction, given as a vector.
- The point and vector determine a ray, that is, a half-infinite line that extends from a starting point, in some direction, to infinity.
- The Raycaster can find all the intersections of the ray with a given set of objects in a scene, sorted by order of distance from the rays's starting point.
- We are interested in the first intersection, the one that is closest to the starting point.

- In this illustration, the scene contains several red spheres and two point lights. A ray is cast from the viewpoint (A) through a point (B) in the image.
- The ray intersects two of the spheres, but we are only interested in the intersection point (C) that is closest to the viewpoint. That's the point that is visible at B in the image.

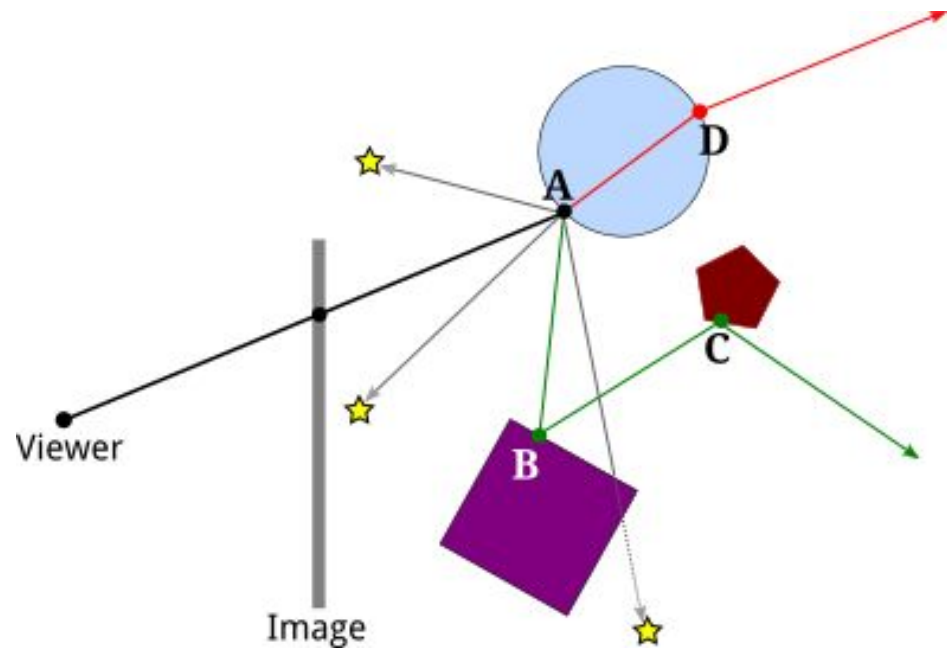


Recursive Ray Tracing

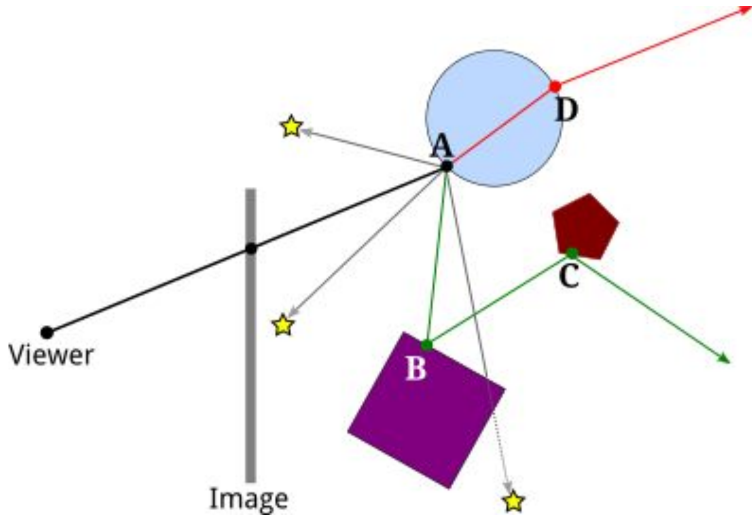
- Consider specular reflection.
- specular reflection can make an object look shiny in the sense that specular highlights can be seen where the object reflects light from a light source towards the viewer.
- But in reality, an object that has a mirror-like surface doesn't just reflect light sources; it also reflects other objects.

Recursive Ray Tracing

- If we are trying to compute a color for a point, A, on a mirror-like surface, we need to consider the contribution to that color from mirror-like reflection of other objects.
- To do that, we can cast a "reflected ray" from A. The direction of the reflected ray is determined by the normal vector to the surface at A and by the direction from A to the viewer.
- This illustration shows a 2D version. Think of it as a cross-section of the situation in 3D



- Here, the reflected ray from point A hits the purple square at point B, and the viewer will see a reflection of point B at A. (Remember that in ray tracing, we follow the path of light rays backwards from the viewer, to find out where they came from.)
- To find out what the reflection of B looks like, we need to know the color of the ray that arrives at A from B.
- But finding a color for B is the same sort of problem as finding a color for A, and we should solve it in the same way: by applying the ray-tracing algorithm to B!
- That is, we use the material properties of the surface at B, we cast shadow rays from B towards light sources to determine how B is illuminated, and—if the purple square has a mirror-like surface—we cast a reflected ray from B to find out what it reflects.



Recursive Ray Tracing

- In the illustration, the reflected ray from B hits a pentagon at point C, so the square reflects an image of the pentagon, and the disk reflects an image of the square, including its reflection of the pentagon.
- Ray-tracing can handle multiple mirror-like reflections between objects in a scene!

- Because applying the ray-tracing algorithm at one point can involve applying the same algorithm at additional points, ray tracing is a recursive algorithm.
- To distinguish this from simple ray casting, ray tracing is often referred to as **"Recursive Ray Tracing."**
- Ray tracing can be extended in a similar way to handle transparency or, more properly, translucency through considering the refraction property.