Lighting

Lecturer: Carol O'Sullivan

Credits: Some slides from Rachel McDonnell


NISSAN
xanavi
BRIDGESTONE


Ballet
Swan Lake
I ♥ MOMMY

Introduction

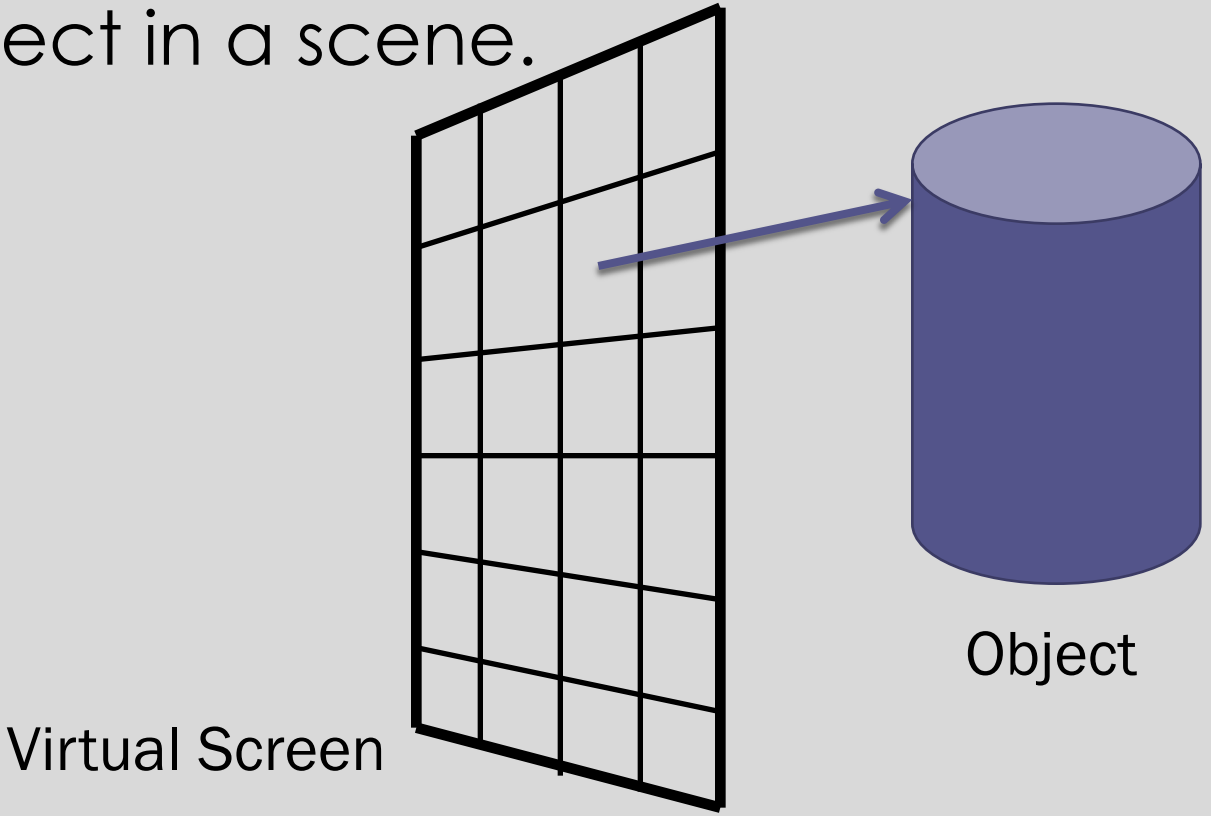
- Realistic image synthesis
  - Photorealism vs. physically-based realism
  - Film and visual effects, architecture, ergonomic design of buildings and offices, computer games, lighting engineering, predictive simulations, flight and car simulators, advertising
- Non-photorealistic rendering
  - Suited for an artistic, technical, or educational approach
  - Pen-and-ink drawings, cartoon-style drawings, technical illustrations, watercolour painting etc.

Overview

- Rendering algorithms (local, global, view dependent, view independent)
- Surface reflectance
- BRDF – Bidirectional Reflectance Distribution Function
- BRDF approximations
- Reflectance equation, radiance equation

Rendering

- Rendering is fundamentally concerned with determining the *most appropriate colour* (i.e. RGB tuple) to assign to a pixel associated with an object in a scene.



Question

- What factors determine the colour of an object at a specific point?

Answer

- What factors determine the colour of an object at a specific point?
  - ***geometry of the object at that point (normal direction)***
  - ***position, geometry and colour of the light sources (luminaires)***
  - ***position and visual response of the viewer***
  - ***surface reflectance properties of the object at that point***
  - ***scattering by any participating media (e.g. smoke, rising hot air)***

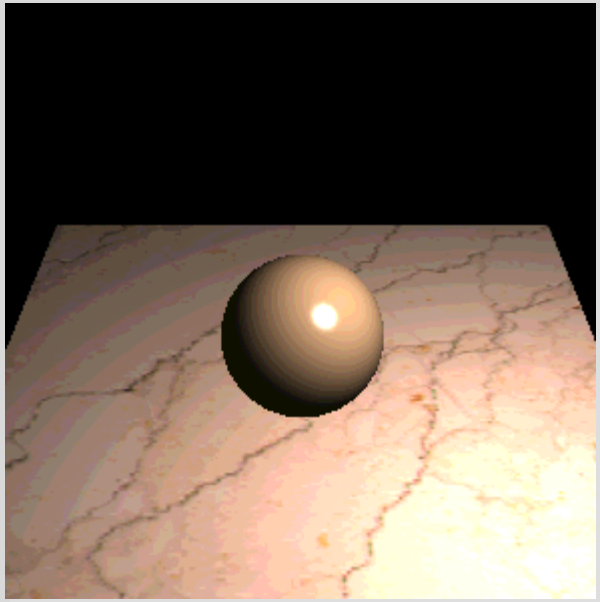
Rendering

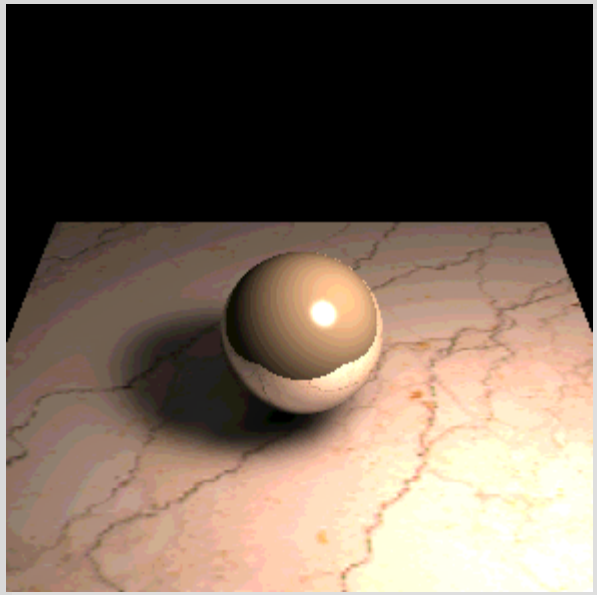
- The colour of an object at a point depends on:
  - geometry of the object at that point (*normal* direction)
  - position, geometry and colour of the light sources (*luminaires*)
  - position and visual response of the viewer
  - *surface reflectance properties* of the object at that point
  - scattering by any *participating media* (e.g. smoke, rising hot air)

Rendering Algorithms

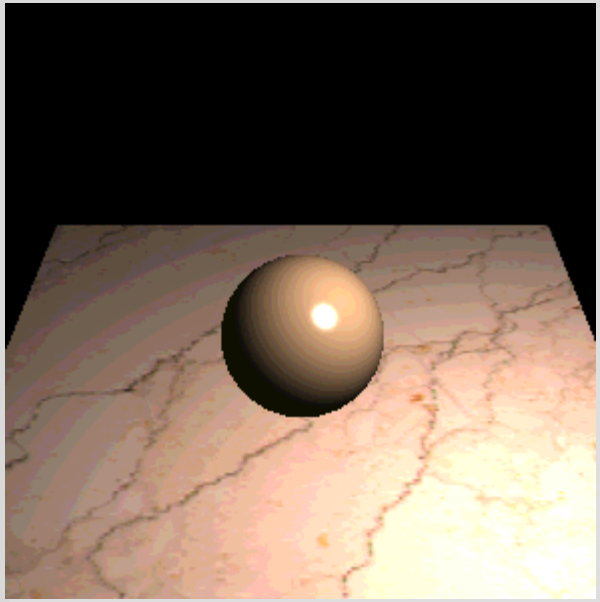
- Rendering algorithms differ in the assumptions made regarding lighting and reflectance in the scene and in the solution space:
  - **local illumination** algorithms: consider lighting only from the light sources and ignore the effects of other objects in the scene (i.e. reflection off other objects or shadowing)
  - **global illumination** algorithms: account for all modes of *light transport*
  - **view dependent** solutions: determine an image by solving the illumination that arrives through the viewport only.
  - **view independent** solutions: determine the lighting distribution in an entire scene regardless of viewing position. Views are then taken after lighting simulation by sampling the full solution to determine the view through the viewport.

Which Lighting Model?



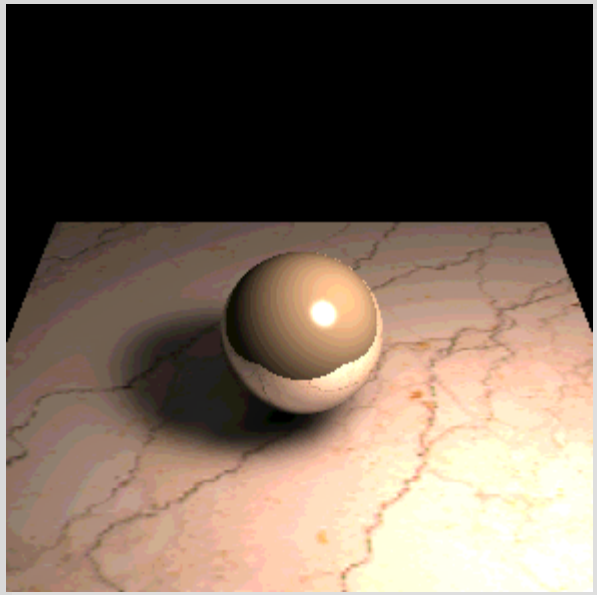


Local vs. Global Illumination



Local

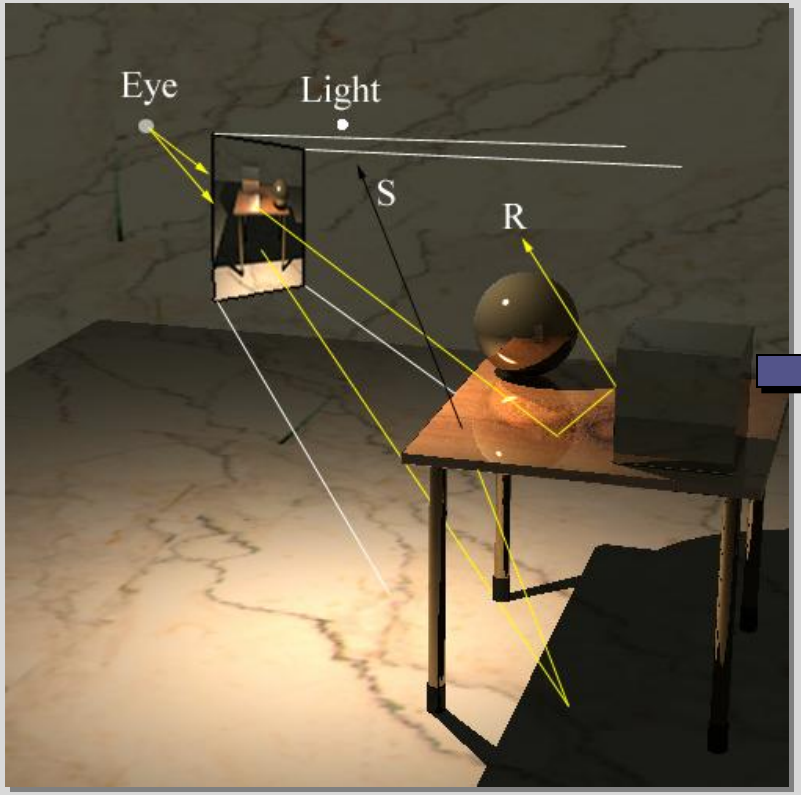
Illumination depends on local object & light sources only



Global

Illumination at a point can depend on any other point in the scene

View Dependent Solution (Ray Traced)



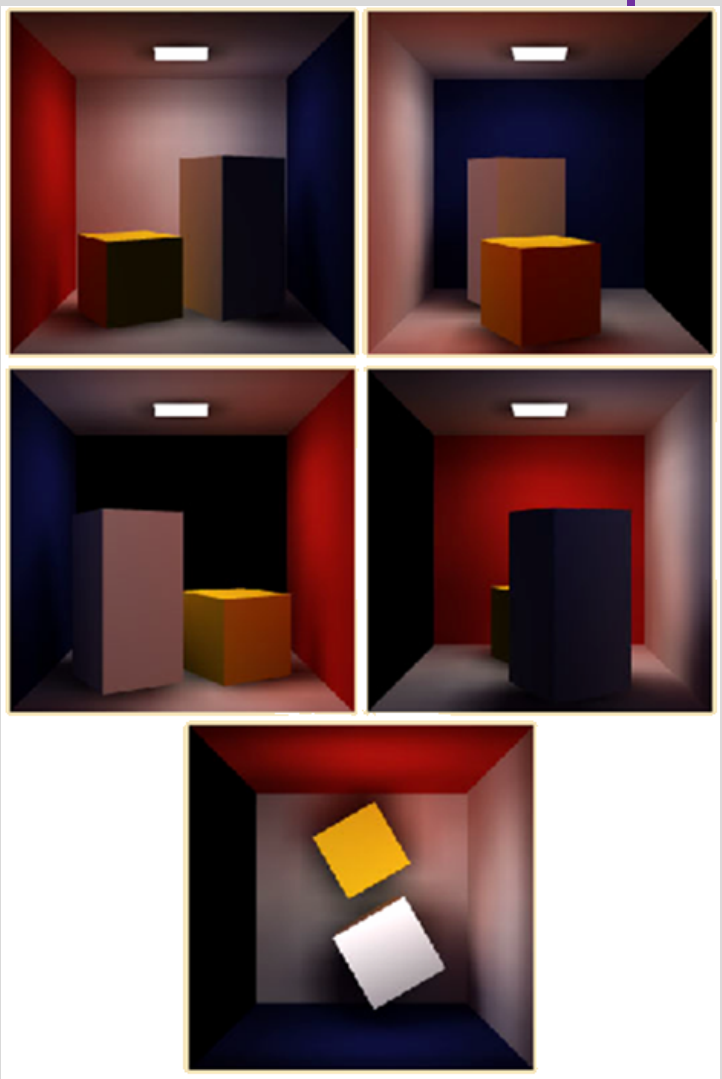
Scene Geometry

Solution determined only for directions through pixels in the viewport

View Dependent

- Advantages
  - Only the relevant parts of the scene are taken into consideration
  - Can work with any kind of geometry
  - Can produce very high-quality results
  - In some methods, view-dependent portions of the solution can be cached as well (glossy reflections, refractions etc).
  - Require less memory than a view-independent solution.
- Disadvantages
  - Requires updating for different camera positions; still, in some implementations portions of the solution may be re-used.

View Independent (Radiosity)



A single solution for the light distribution in the entire scene is determined in advance.

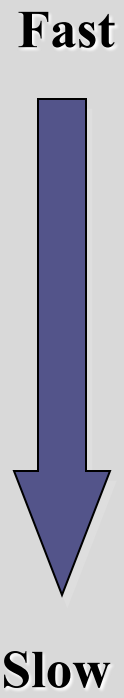
Then we can take different snapshots of the solution from different viewpoints by sampling the complete solution for specific positions and directions.

View Independent

- Advantages
  - Solution needs to be computed only once.
- Disadvantages
  - All of the scene geometry must be considered, even though some of it may never be visible.
  - The type of geometry in the scene is usually restricted to trianglular or quadrangular meshes (no procedural or infinite geometry allowed).
  - Detailed solutions require lots of memory.
  - Only the diffuse portion of the solution can be cached; view-dependent portions (glossy reflections) must still be computed.

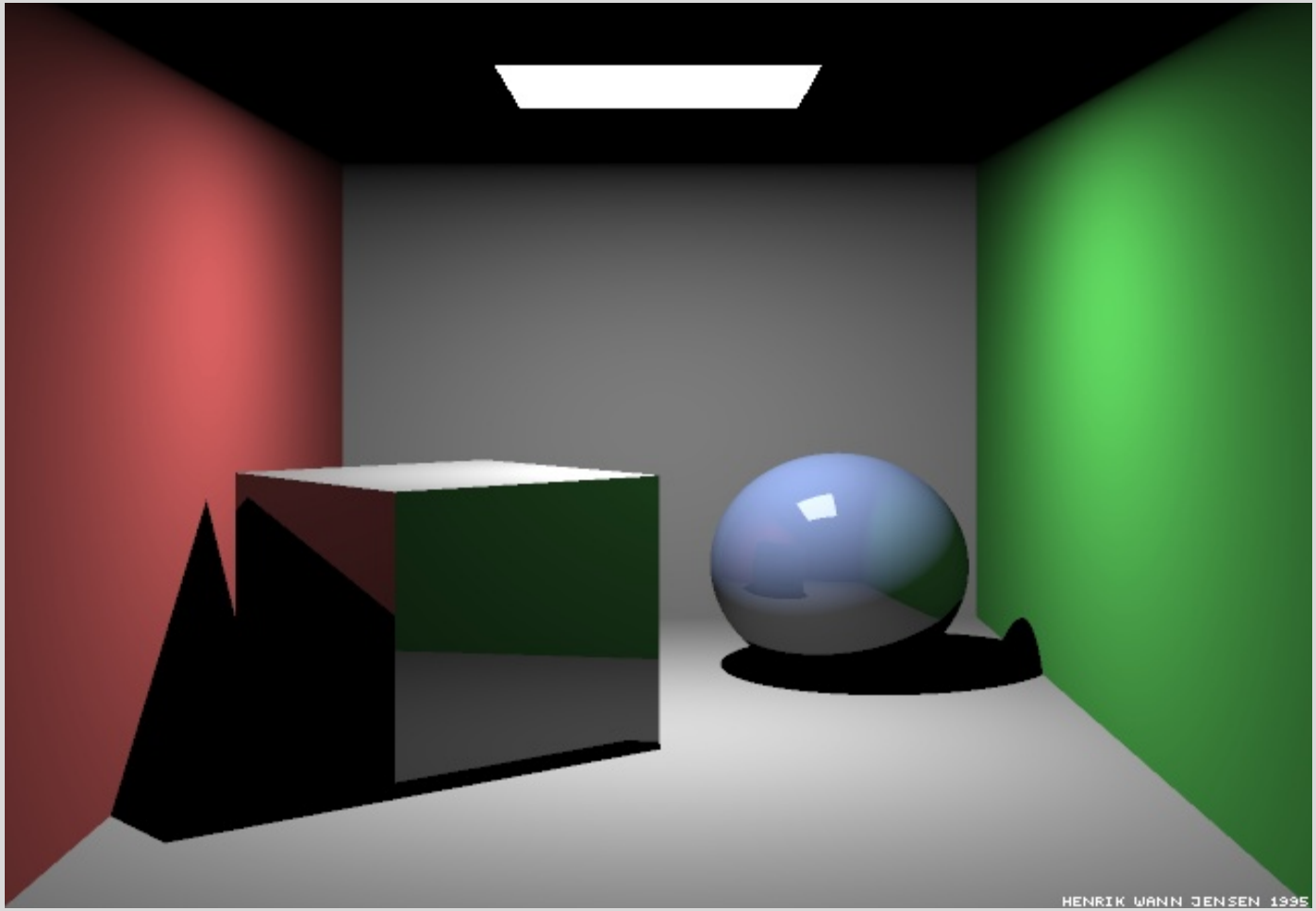
Global Illumination Algorithms

- Different algorithms solve the illumination problem with making different assumptions to vary the speed/accuracy tradeoff.
  - ***Z-Buffer Algorithms***:
    - can compute approximate shadows and reflection from planar surfaces
  - ***Ray Tracing Algorithms***:
    - determine exact shadows, reflections and refractive effects (transparency) assuming point light sources (no volume).
  - ***Radiosity Algorithms***:
    - computes approximate solutions assuming no *shiny* surfaces, but light sources can be arbitrarily large and all surfaces polygonal.
  - ***Path Tracing Algorithms***:
    - employing an expensive Monte-Carlo solution to handle arbitrary geometries, reflectance and lighting.

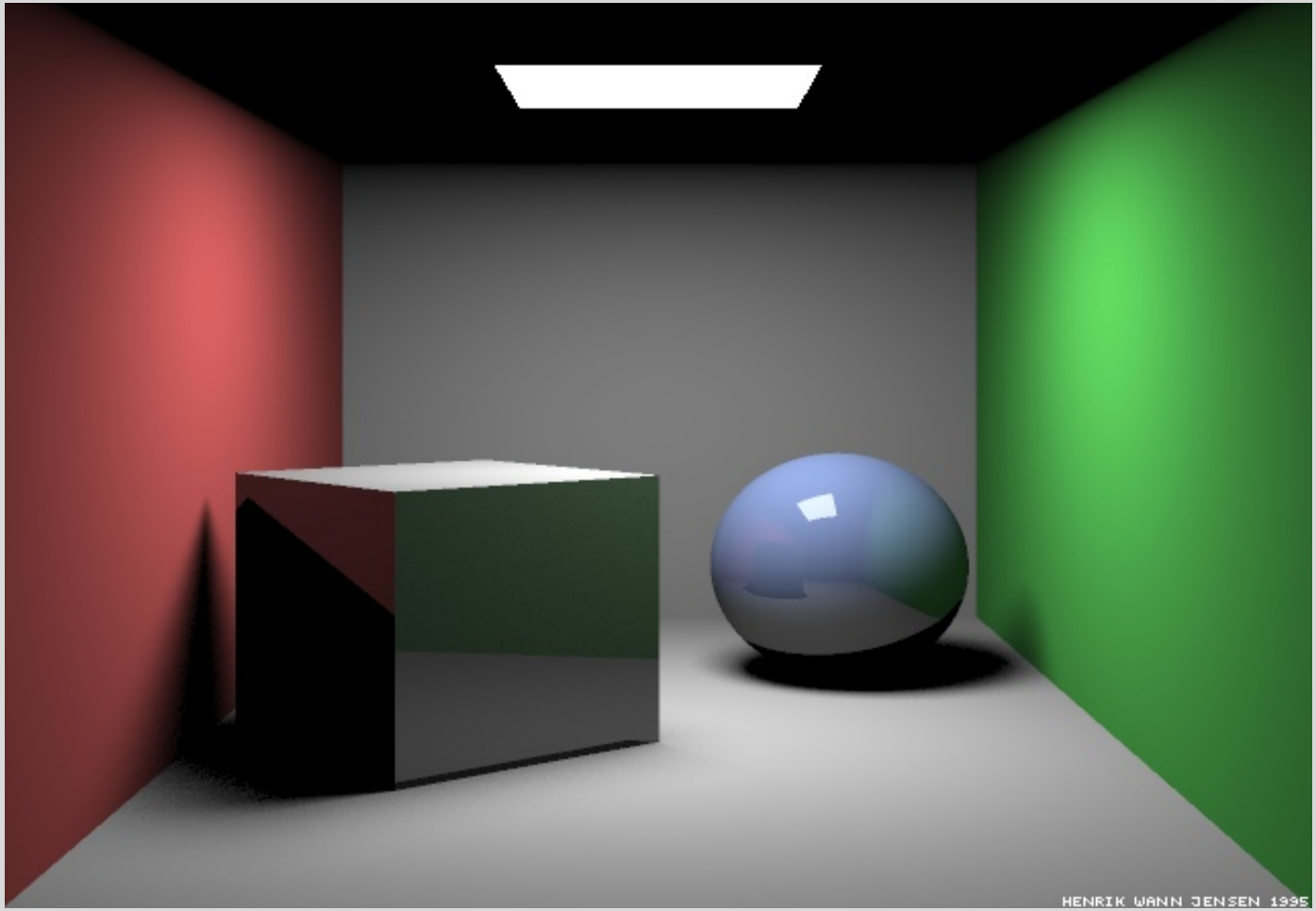


Z-buffer methods only

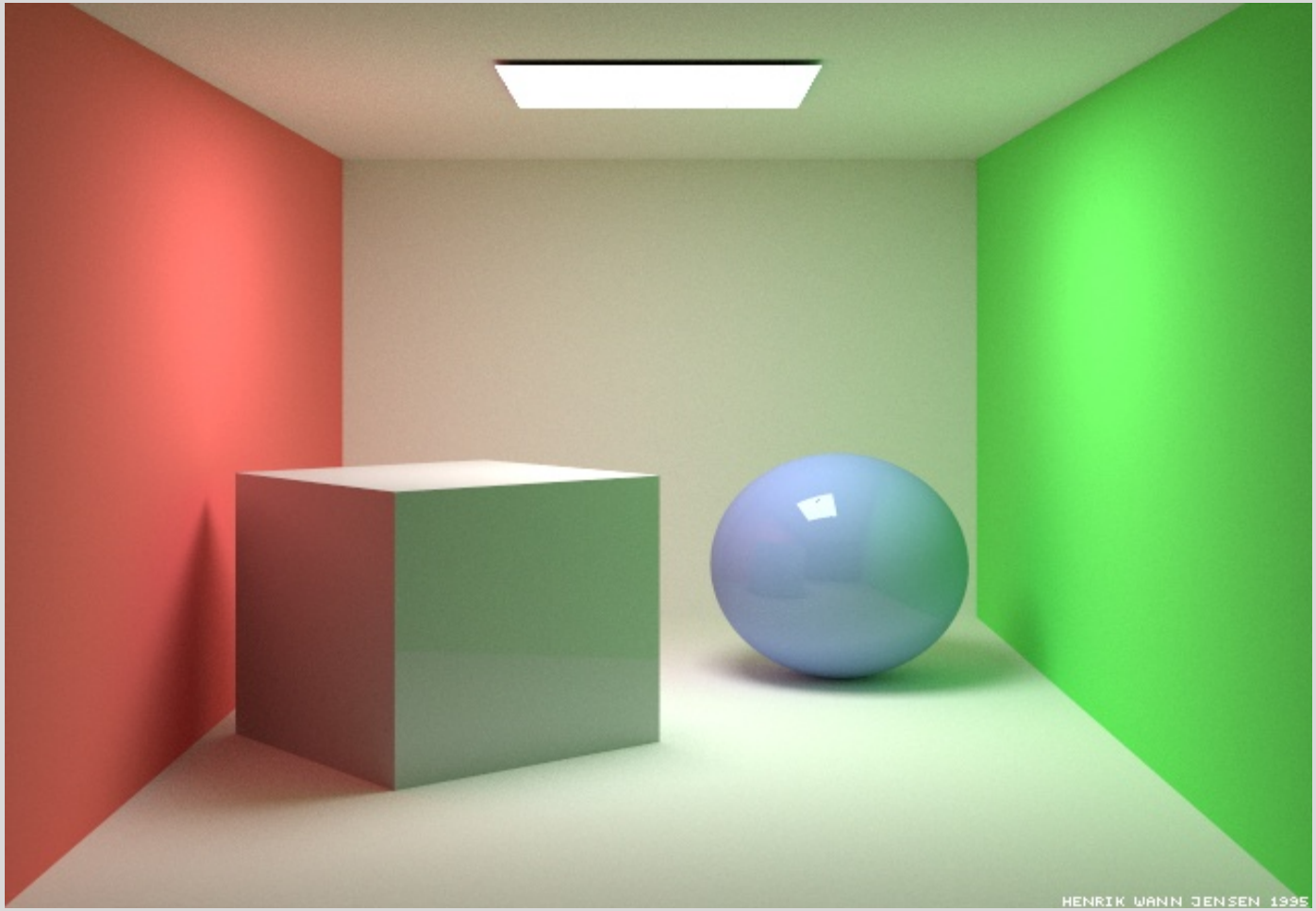
Crystal Glass Rendering using Path Tracing



Ray traced image



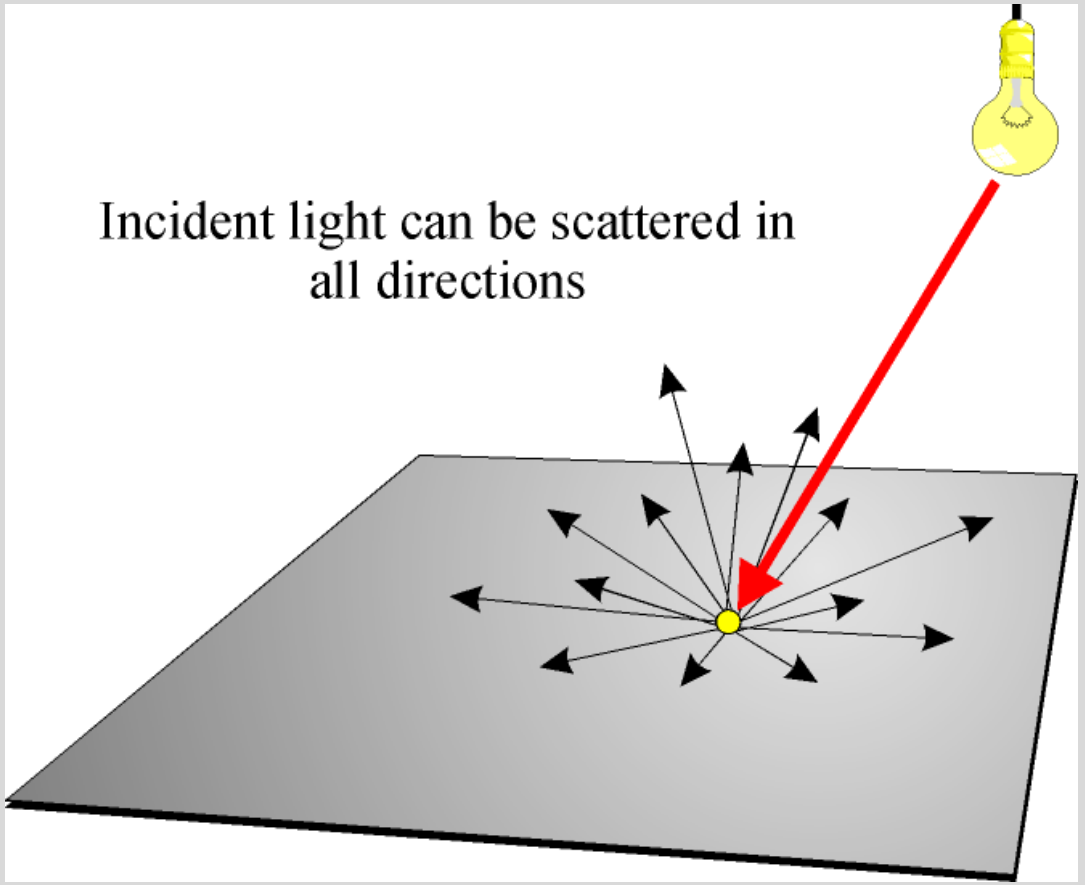
Stochastic ray traced image

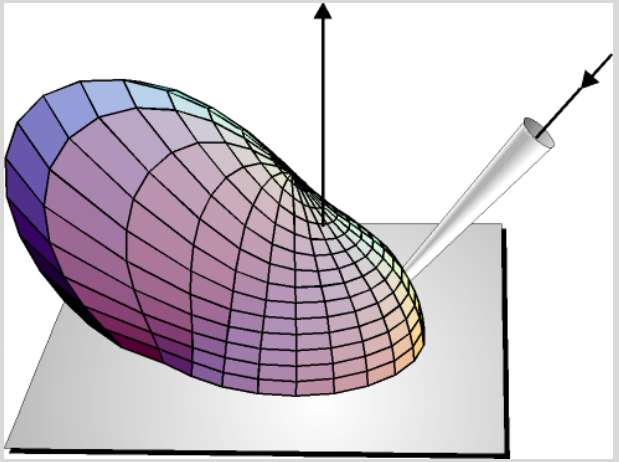


Path traced image

Surface Reflectance

- Much of current realistic image synthesis research is devoted to the modeling of surfaces
  - in particular the solving of ***light reflection*** off arbitrarily complex surface geometries.
- A surface ***scatters*** light that is incident on it.
  - This scattering can theoretically distribute the scattered light in any direction from the scattering point.
- Most algorithms make **assumptions** regarding the directions through which the light is scattered = ***scattering distribution***
- View dependent algorithms must determine the point in the scene which is visible through each pixel and then *determine the light that is scattered from here towards the pixel*.

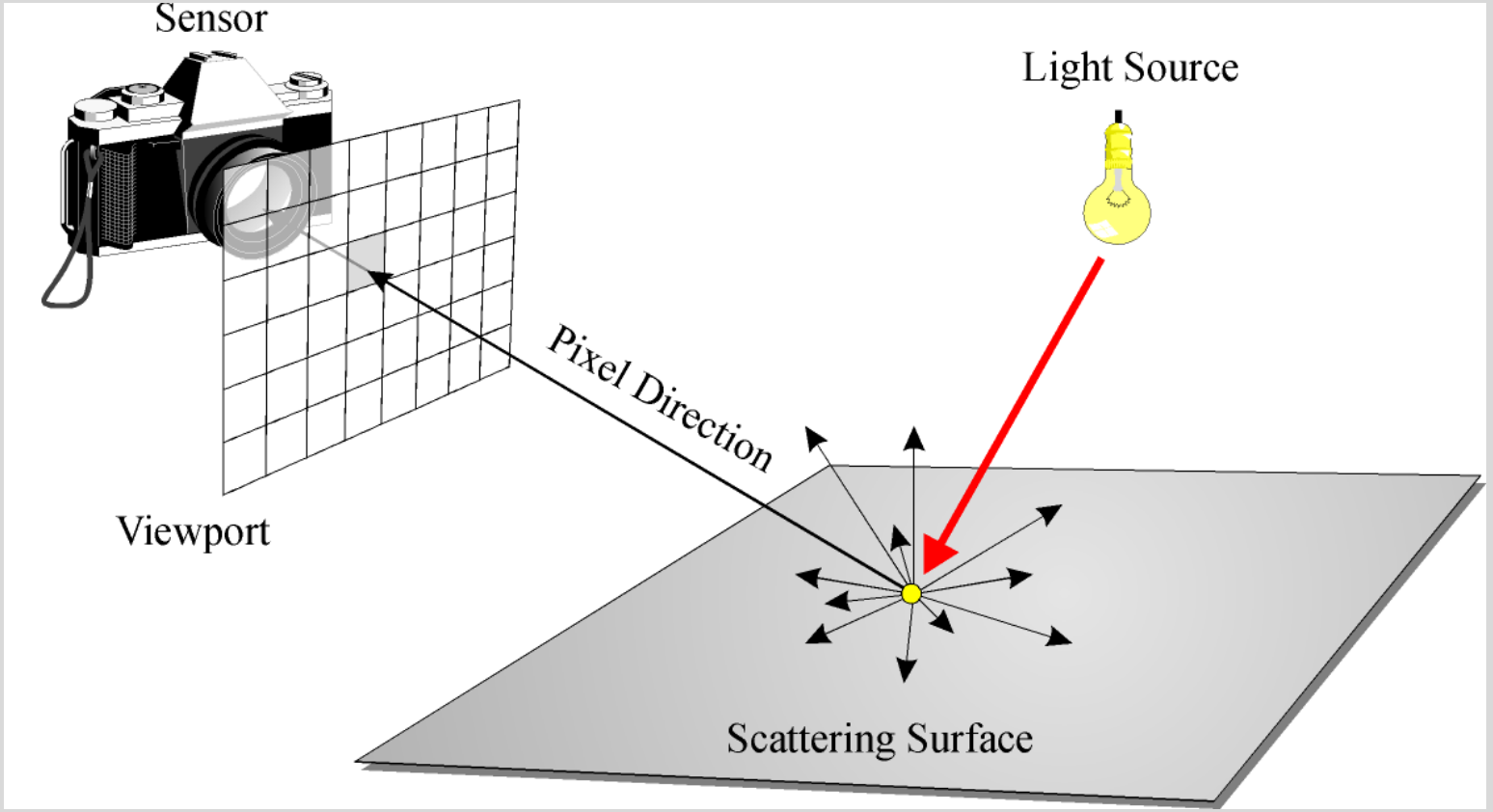




BRDF

- Energy is scattered from a surface in a distribution that depends on the surface's **microscopic geometry**.
- This distribution can be described as a function (strictly positive) that records the reflected energy per direction

= BRDF: Bidirectional Reflectance Distribution Function



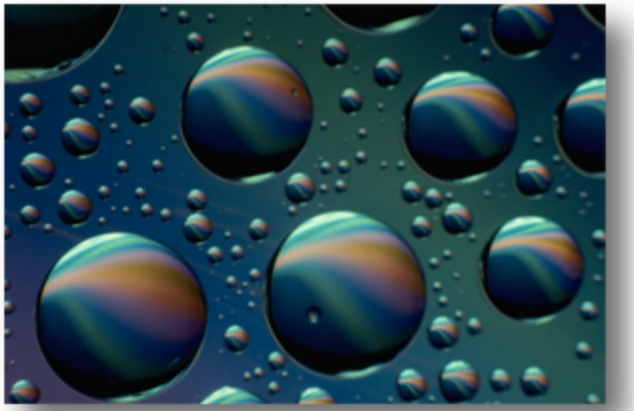
- An image is formed when light energy is scattered by surfaces in the scene towards the viewport
  - (or emitted directly towards the viewport)

Multi-layered Surfaces

Coloured Glass

Human Skin

Stone

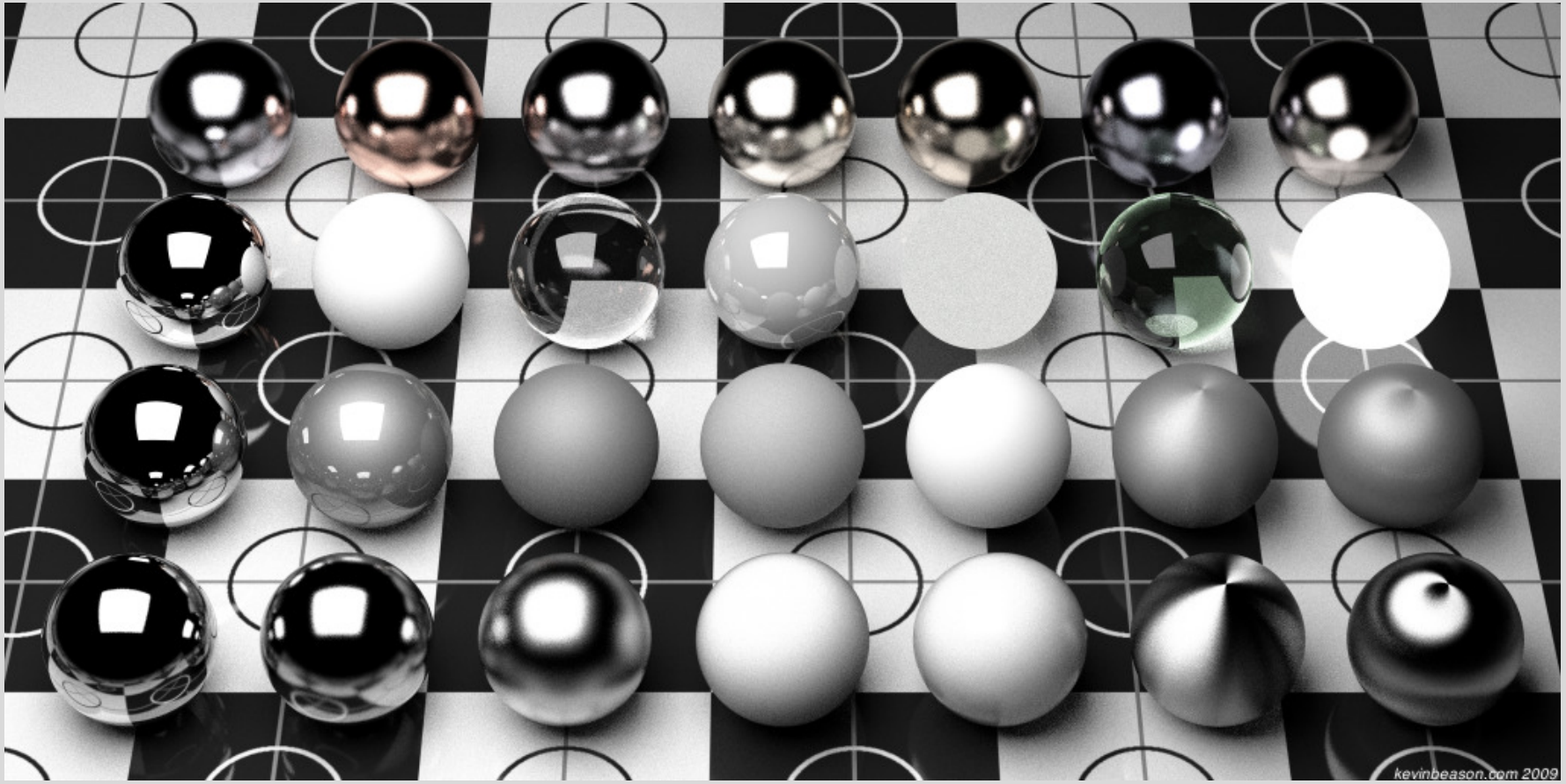


Thin Film

Tarnished Metal

Complex Surface Scattering Examples

Different BRDFs

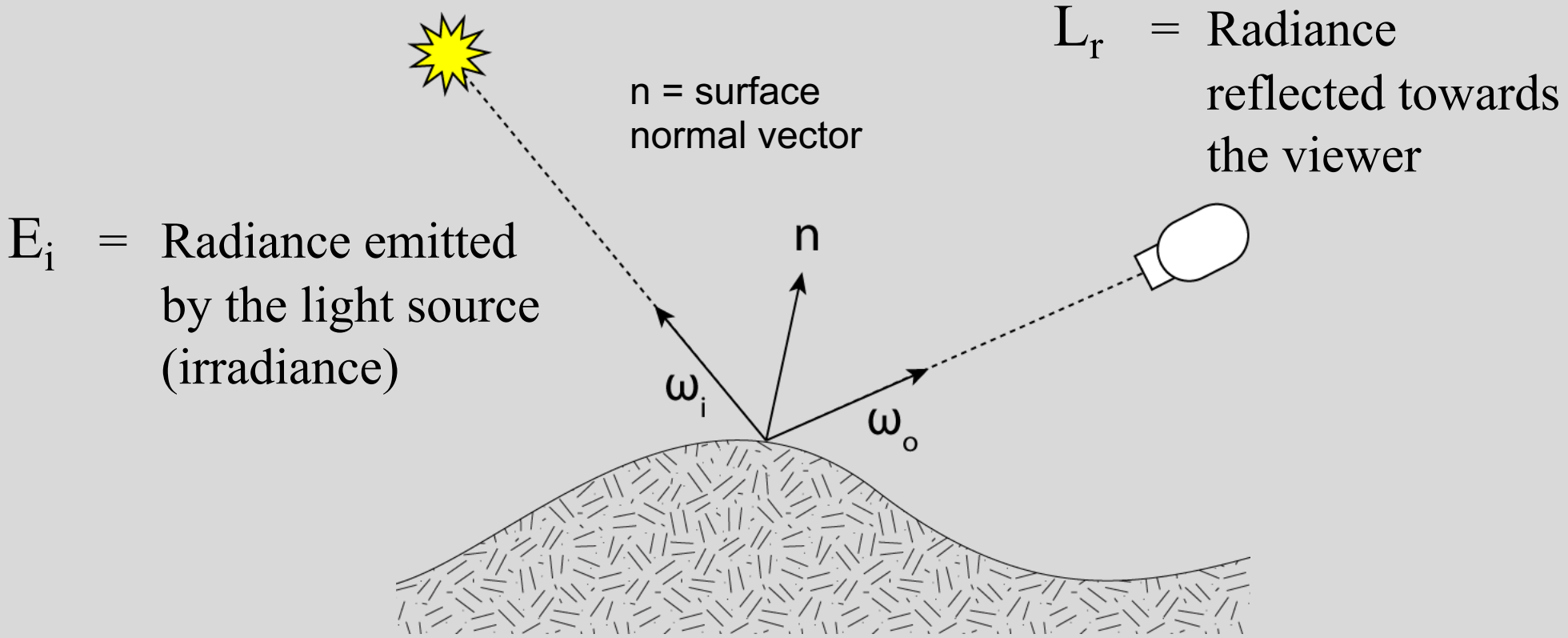


Subsurface scattering, NVIDIA

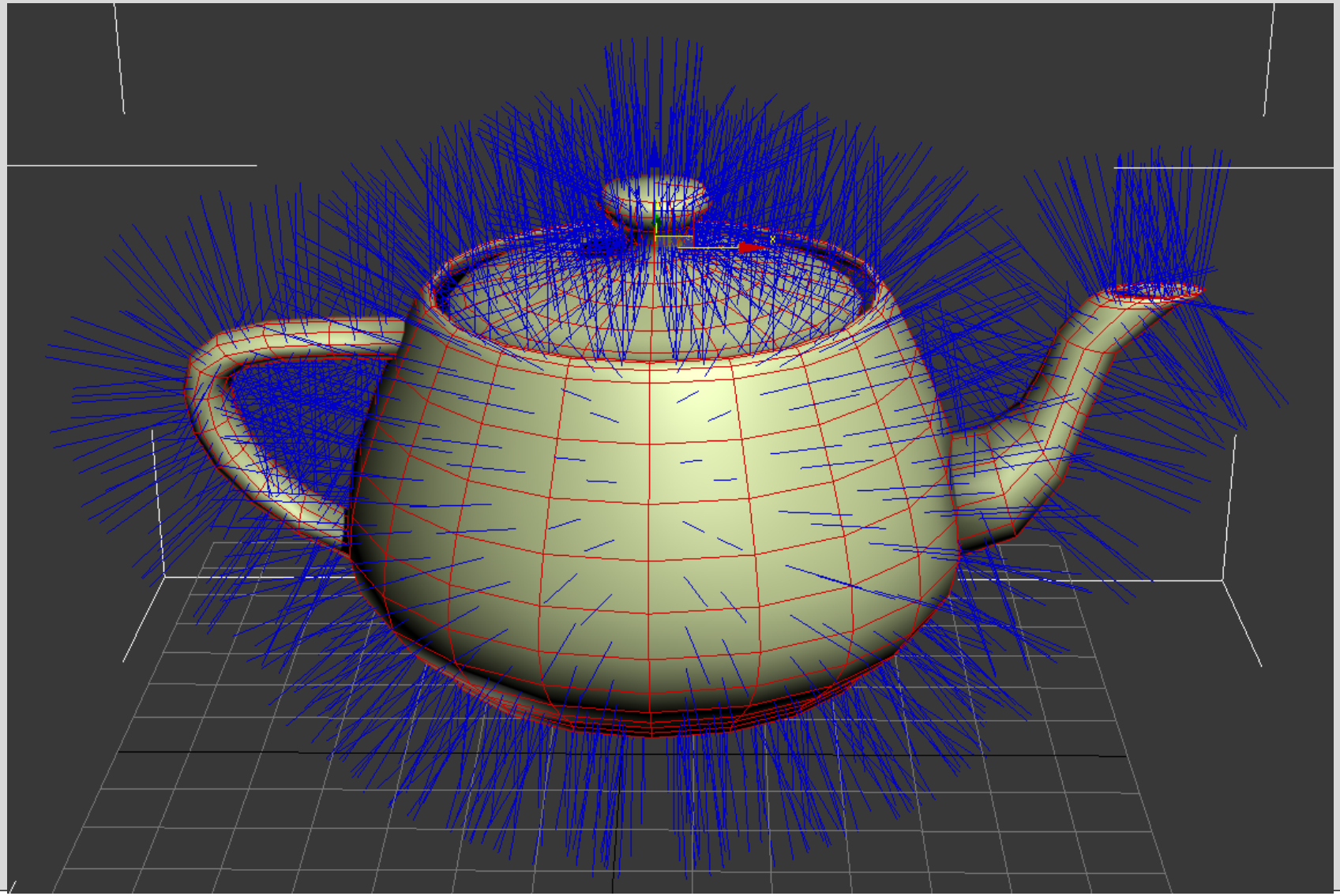
- this occurs when light enters a translucent object and exits at a different point.

Bi-directional Reflectance Distribution Function (BRDF)

- Describes reflected radiance L given incident radiance (irradiance) E.

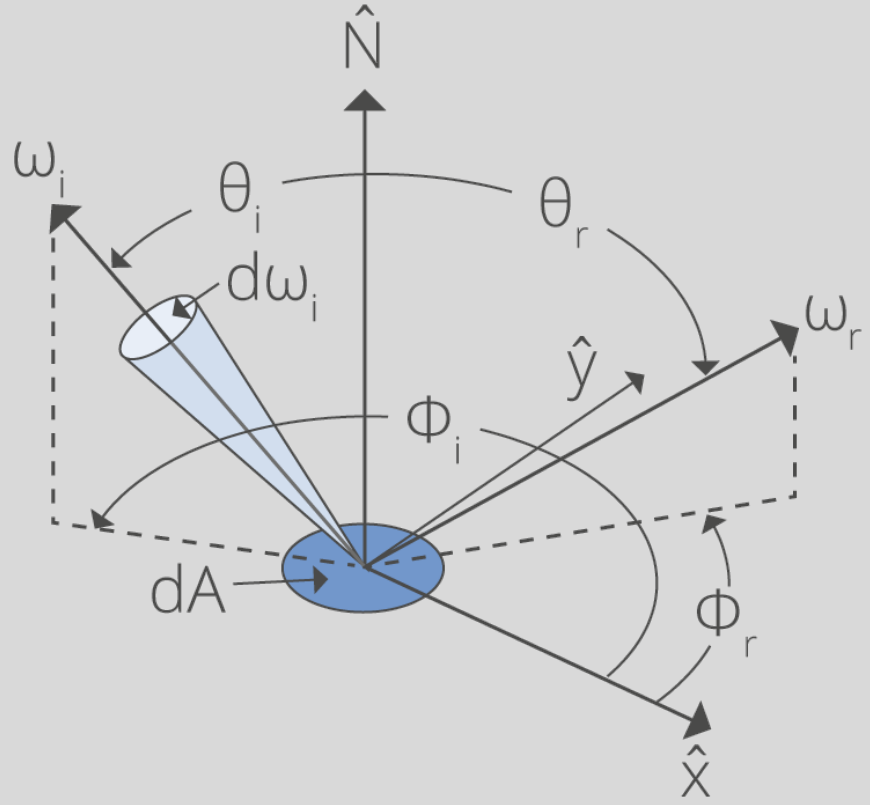


Polygon Normals



Bi-directional Reflectance Distribution Function (BRDF)

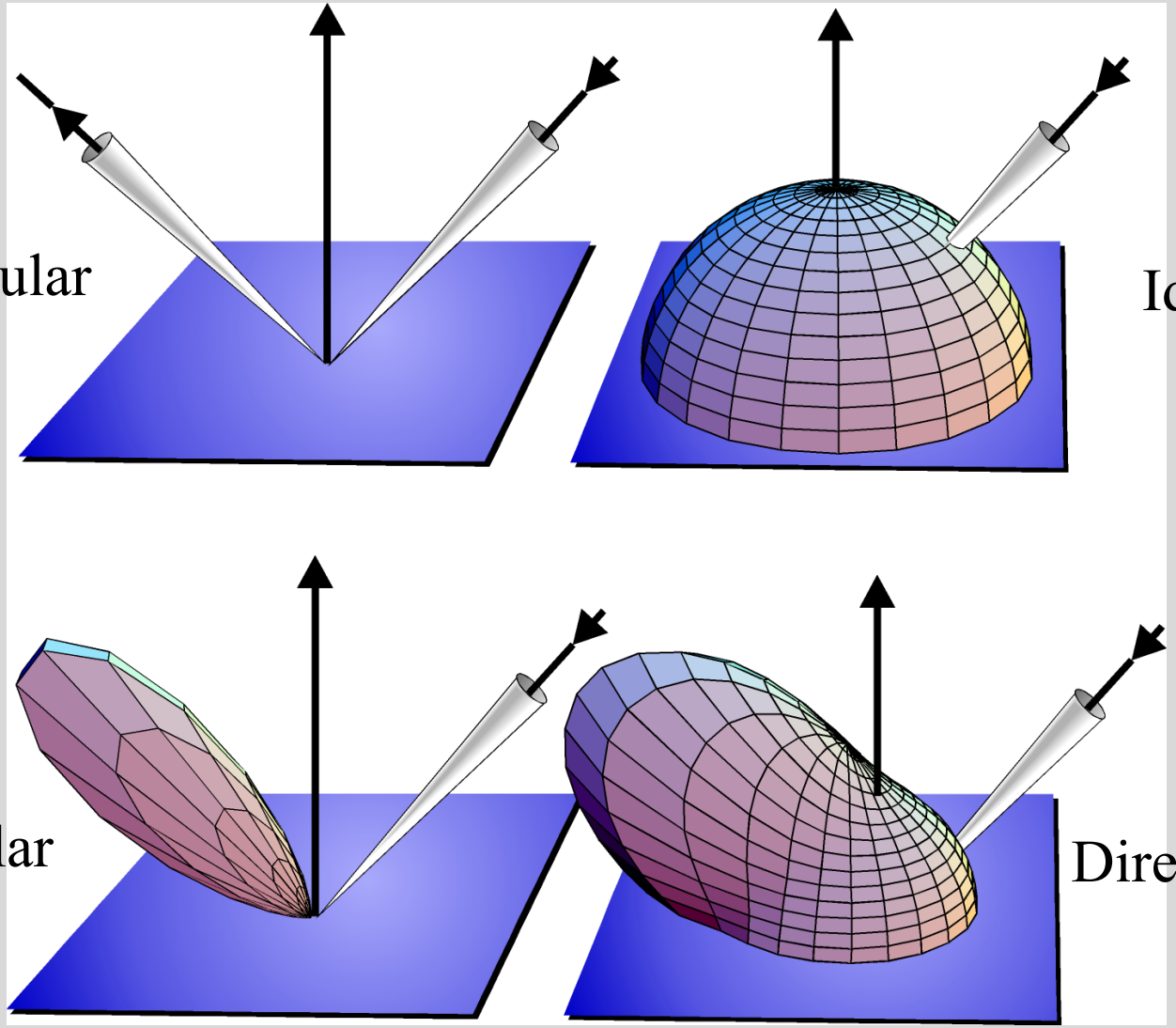
- Theoretically, a high dimension function:
  - position = x
  - incoming direction = $\omega_i = (\theta_i, \phi_i)$
  - reflected direction = $\omega_r = (\theta_r, \phi_r)$
  - wavelength = λ

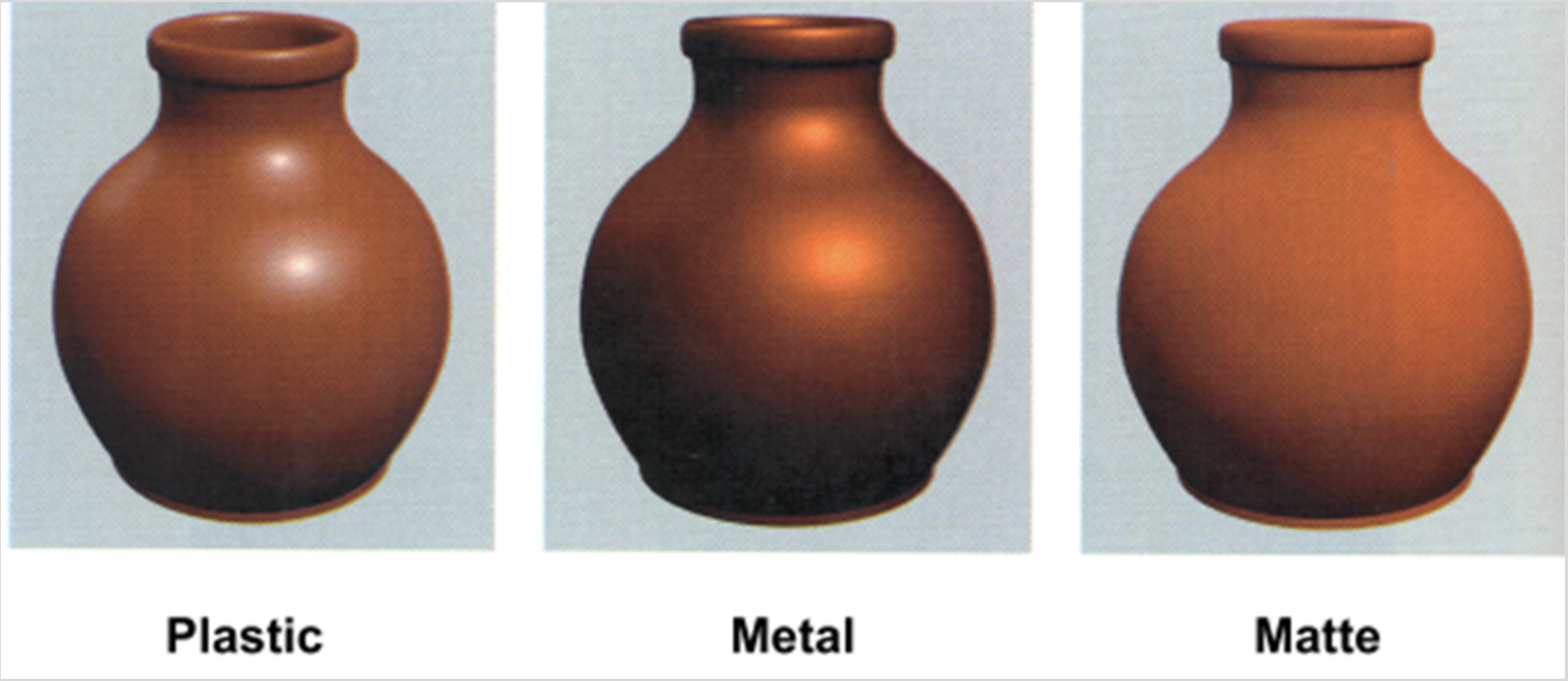


$$f_r(x, \omega_i, \omega_r) = \frac{dL_r(x, \omega_r)}{E_i(x, \omega_i)\cos\theta_i d\omega_i}$$

L_r = Reflected Radiance
E_i = Irradiance

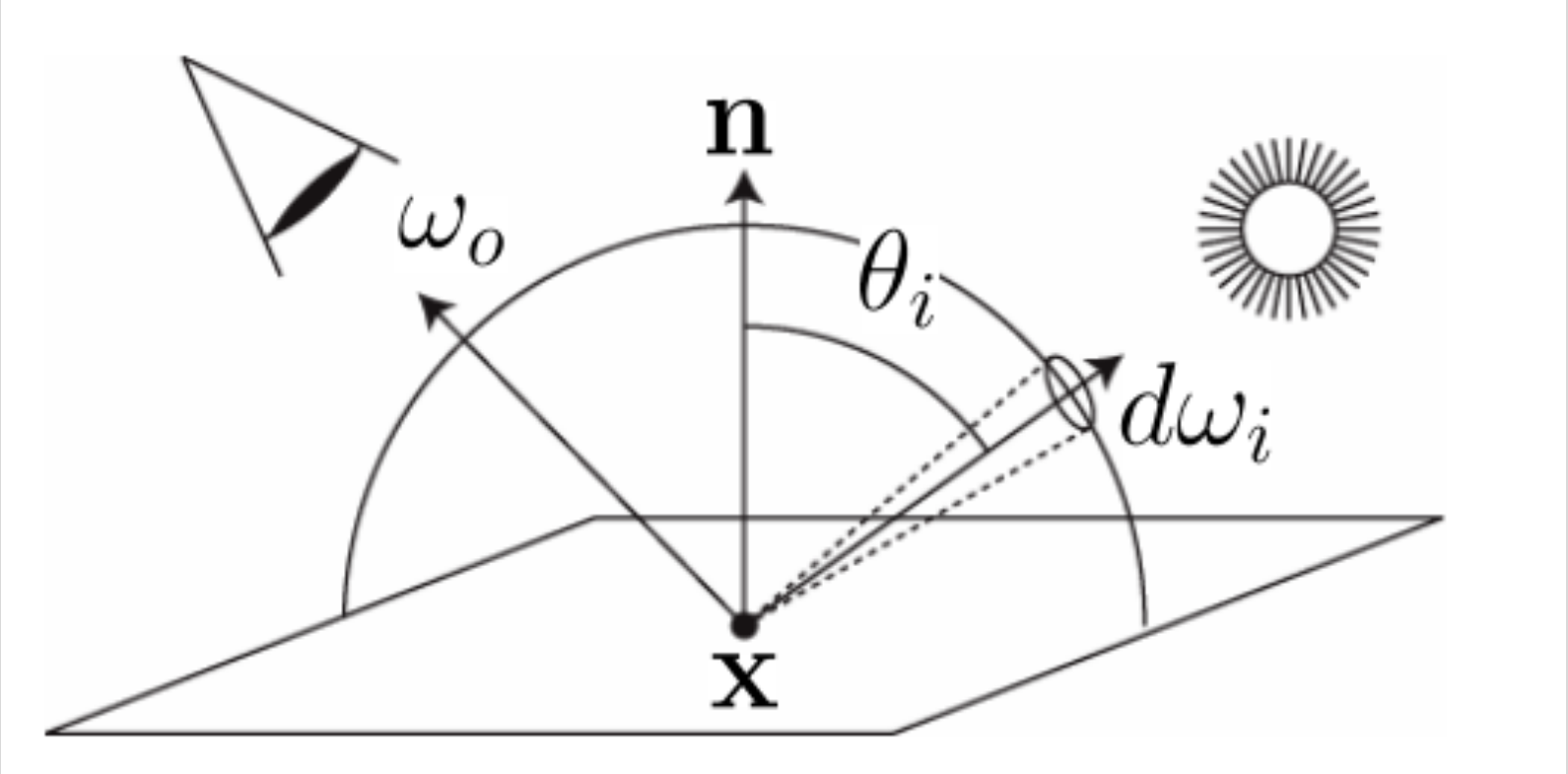
BRDF Approximations




Plastic
Metal
Matte

Reflectance Equation

- The *reflectance equation* relates reflected radiance to incoming radiance that is scattered according to the surface's BRDF:



Reflectance Equation

- The *reflectance equation* relates reflected radiance to incoming radiance that is scattered according to the surface's BRDF:

$$\underbrace{L_r(x,\omega_r)}_{\text{Reflected radiance}} = \int_{\Omega} f_r(x,\omega_i,\omega_r) L_i(x,\omega_i) \cos\theta_i \, d\omega_i$$

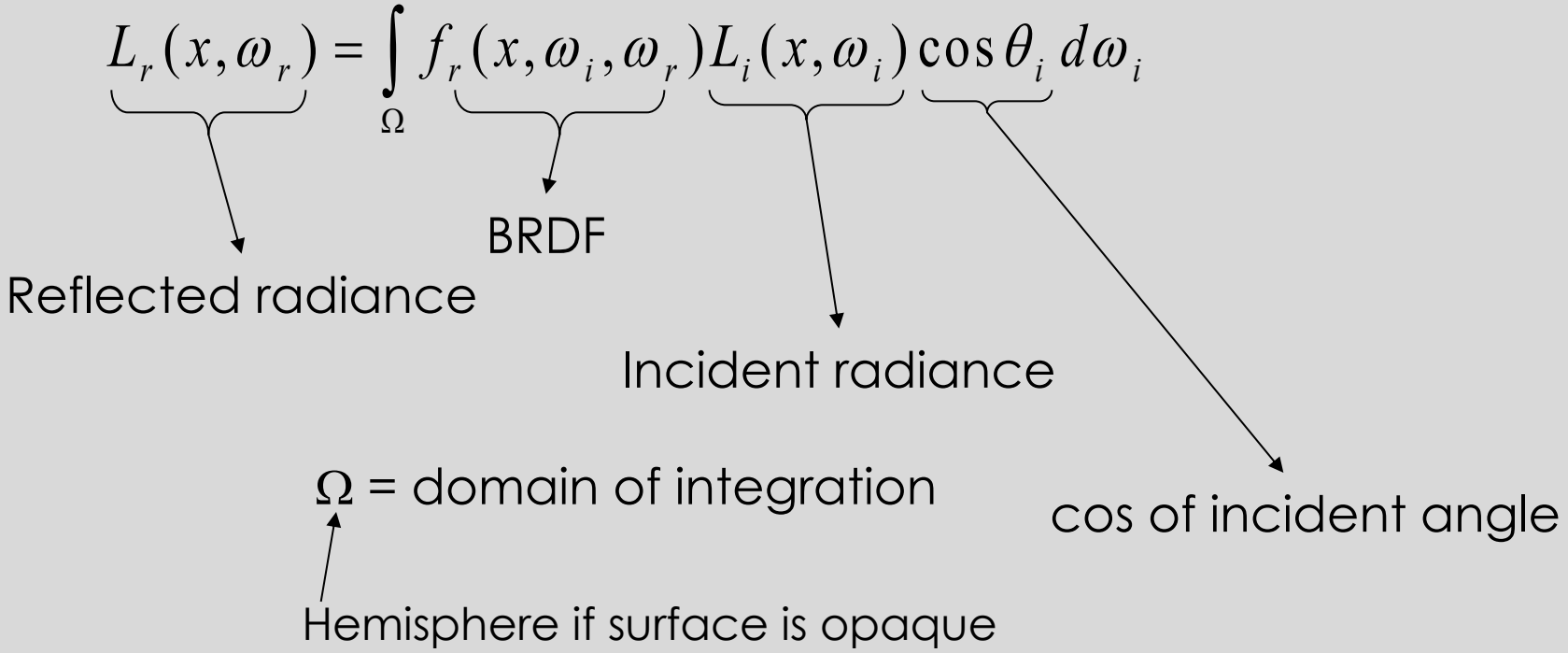
Reflectance Equation

- The *reflectance equation* relates reflected radiance to incoming radiance that is scattered according to the surface's BRDF:

$$\underbrace{L_r(x,\omega_r)}_{\text{Reflected radiance}} = \int_{\Omega} \underbrace{f_r(x,\omega_i,\omega_r)}_{\text{BRDF}} L_i(x,\omega_i)\cos\theta_i \, d\omega_i$$

Reflectance Equation

- The *reflectance equation* relates reflected radiance to incoming radiance that is scattered according to the surface's BRDF:



Radiance Equation

- The radiance equation includes a *self-emitted term* to account for light sources.
- This is the most important equation in rendering theory.
- We solve for $L_r(x, \omega_r)$ at each visible point in the scene.

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_{\Omega} f_r(x,\omega_i,\omega_r) L_i(x,\omega_i) \cos\theta_i \, d\omega_i$$

Self emitted radiance
(non zero for light sources)

Linear Fredholm Integral of the 2nd kind