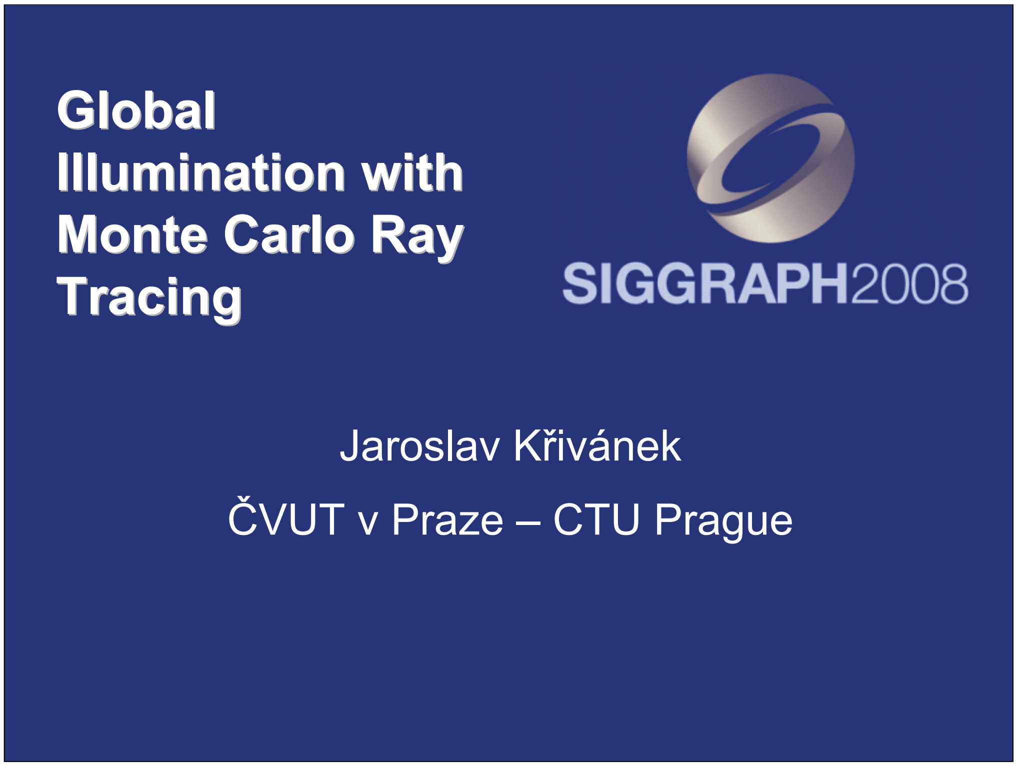
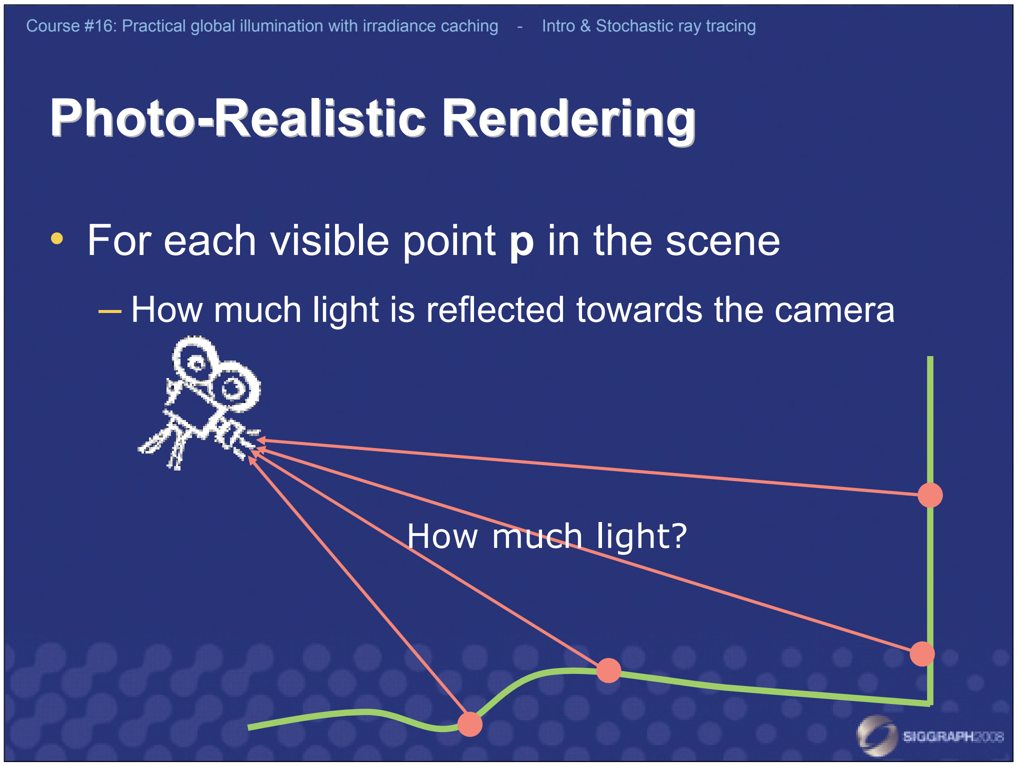


Global Illumination with Monte Carlo Ray Tracing

SIGGRAPH2008

Jaroslav Křivánek

ČVUT v Praze – CTU Prague



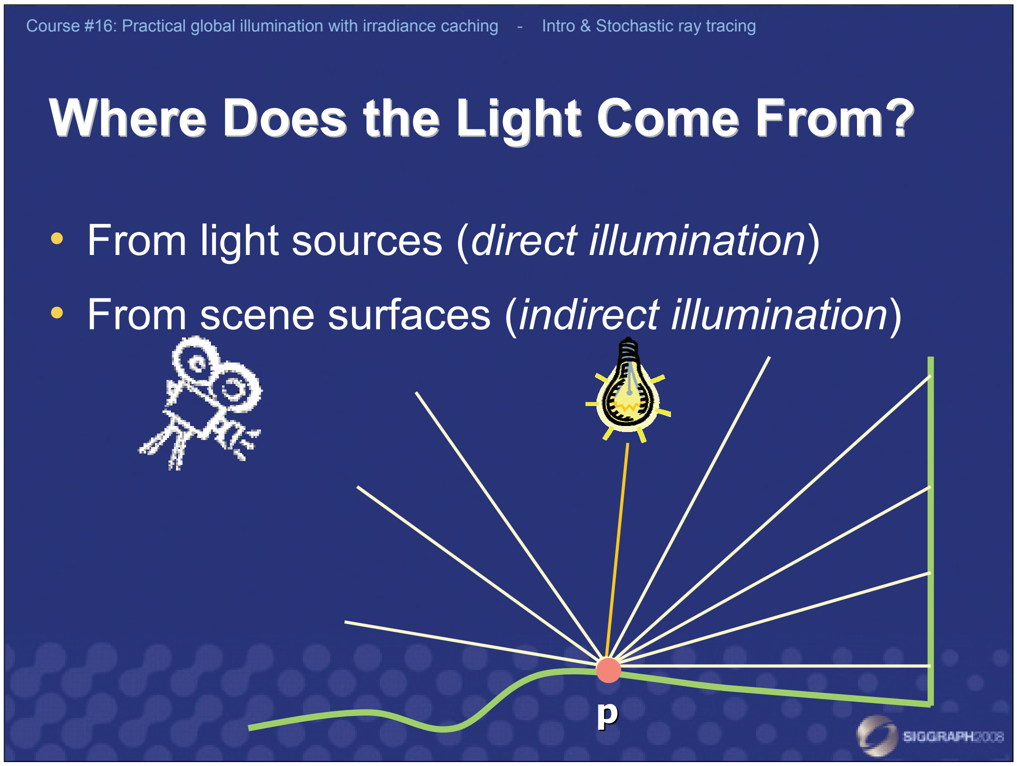
Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Photo-Realistic Rendering

- For each visible point **p** in the scene
  - How much light is reflected towards the camera

How much light?
SIGGRAPH2008

Given the description of a scene, the goal of photo-realistic rendering is to compute the amount of light reflected from visible scene surfaces that arrives to the virtual camera through image pixels. This light determines the color of image pixels.

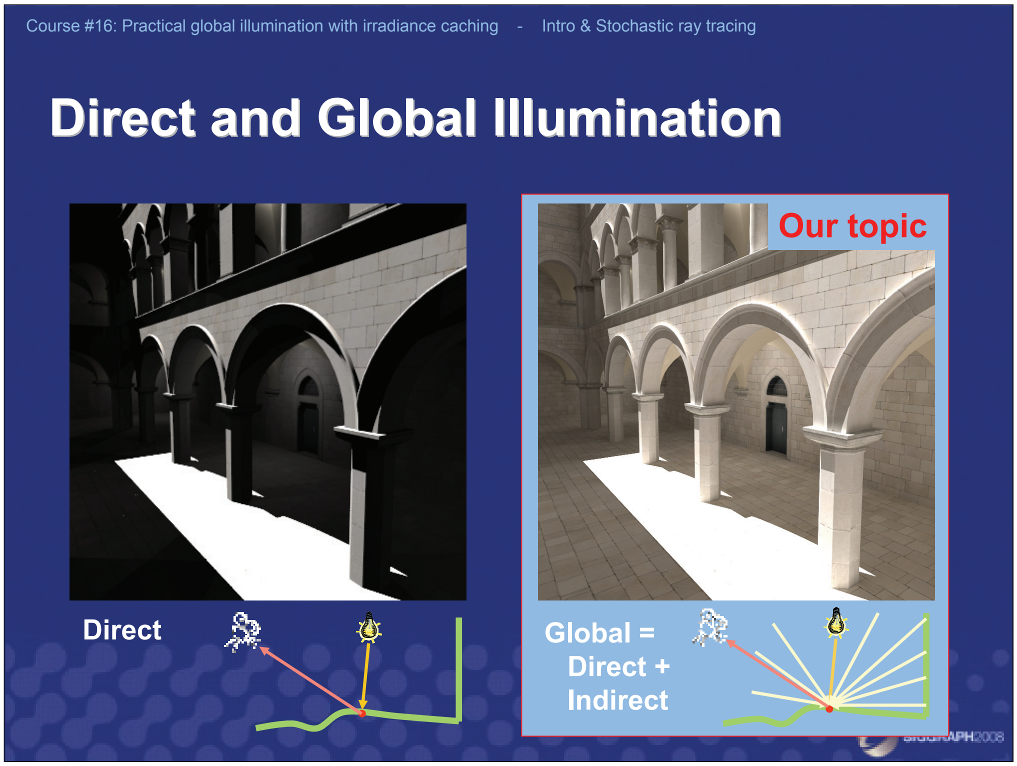


Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Where Does the Light Come From?

- From light sources (*direct illumination*)
- From scene surfaces (*indirect illumination*)

Focusing on one such point, where does the light come from? Surely, it comes directly from the light sources placed in the scene – this is called the *direct illumination*. But light also comes after being reflected from other scene surfaces. This is the *indirect illumination*.



Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

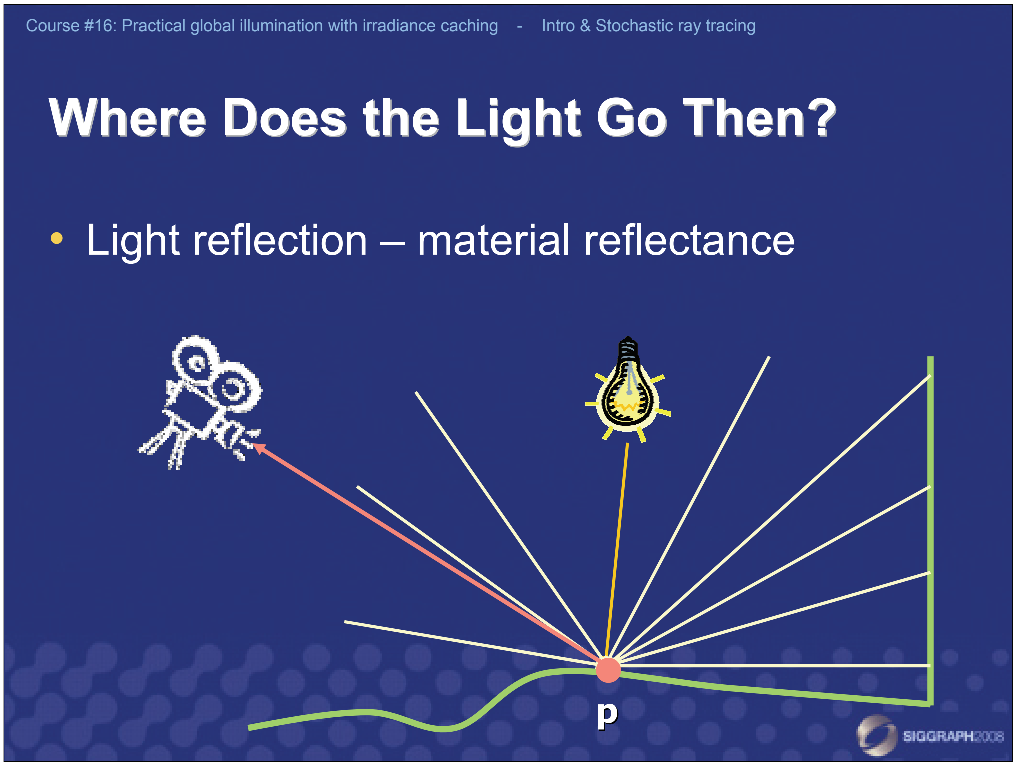
Direct and Global Illumination

Our topic

Direct

Global =
Direct +
Indirect

On the left is an image generated by taking into account only direct illumination. Shadows are completely black because, obviously, there is no direct illumination in the shadowed areas. On the right is the result of global illumination computation where light is allowed to bounce around in the scene before it is finally reflected towards the camera.

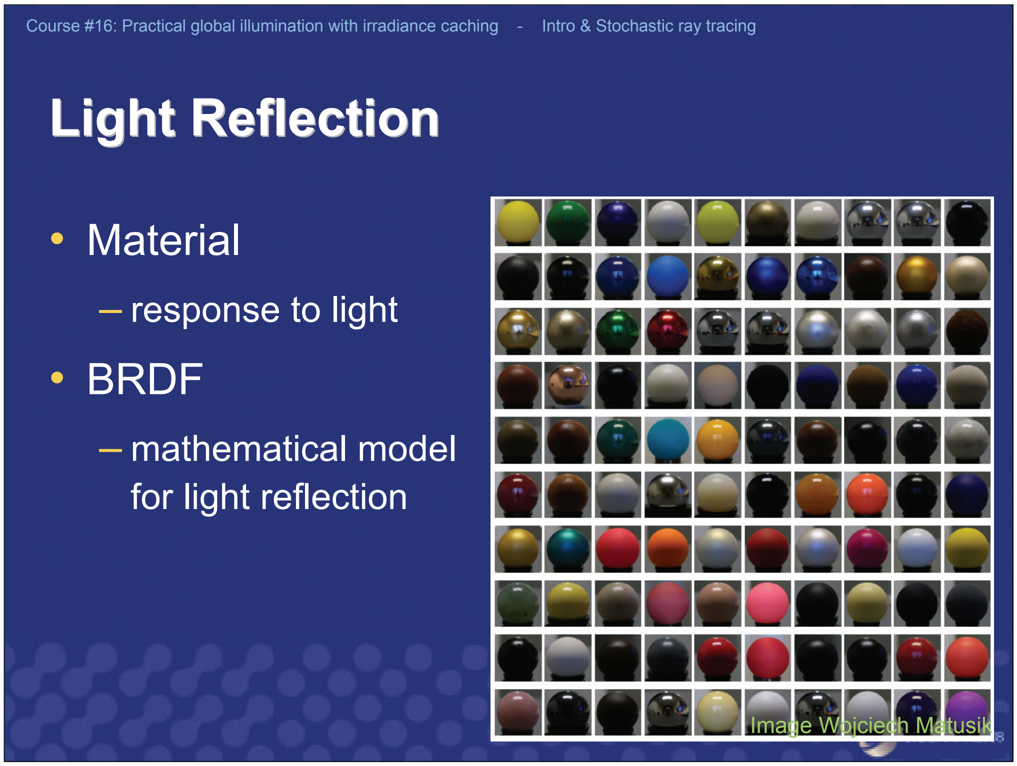


Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Where Does the Light Go Then?

- Light reflection – material reflectance

So far, we know where the light comes from at **p**. But we want to compute how much of it is reflected towards the camera. This is determined by the surface material reflectance properties.



Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

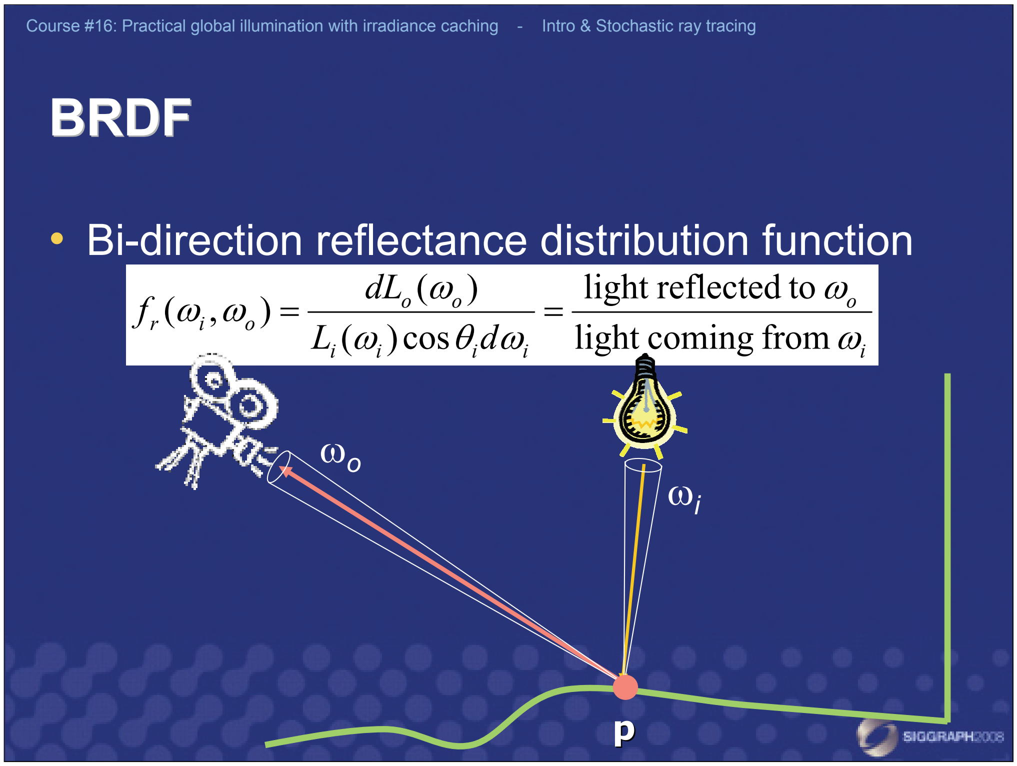
Light Reflection

- Material
  - response to light
- BRDF
  - mathematical model for light reflection

Image Wojciech Matusik

From the point of view of light simulation, the material can be looked upon as the surface's response to incoming light. The spheres on the right are all illuminated by the same light – their varying appearance is only determined by their different material properties.

In physically-based image synthesis (rendering), the most common way to mathematically describe material reflectance characteristics is the BRDF – bidirectional reflectance distribution function.



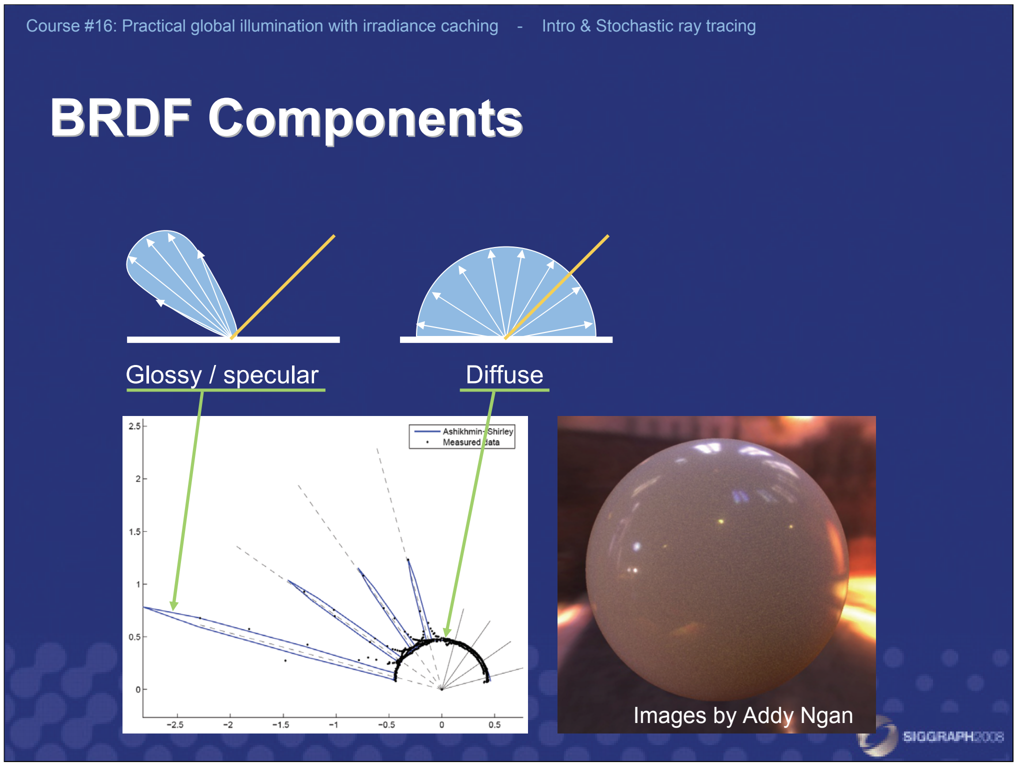
Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

BRDF

- Bi-direction reflectance distribution function

$$f_r(\omega_i,\omega_o)=\frac{dL_o(\omega_o)}{L_i(\omega_i)\cos\theta_i d\omega_i}=\frac{\text{light reflected to }\omega_o}{\text{light coming from }\omega_i}$$

The BRDF at a single point has two parameters: the incoming and the outgoing light directions, ω_i and ω_o, respectively. For our purposes, it is sufficient to say that the BRDF value is the ratio of the light reflected in the outgoing direction to the light coming from the incoming direction. In other words, the BRDF tells us how much of the light energy coming from ω_i gets reflected towards ω_o. In radiometry, the quantity associated with the amount of light reaching a point **p** along a direction w is the **radiance** $L(\mathbf{p},w)$.



Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

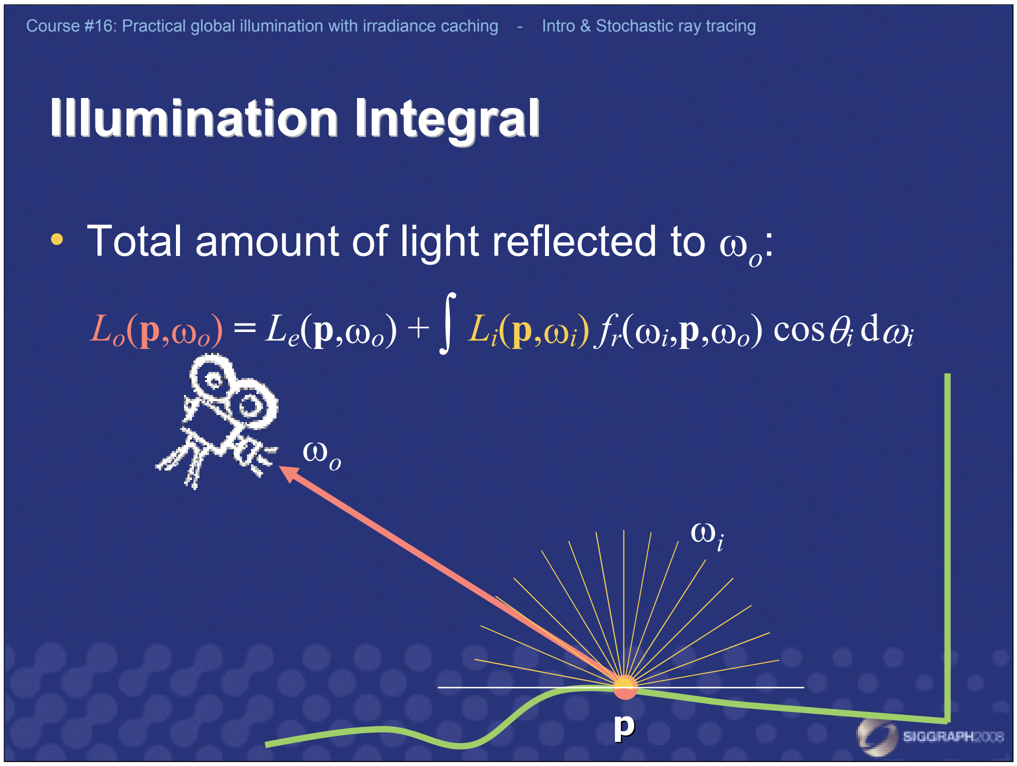
BRDF Components

Glossy / specular

Diffuse

Images by Addy Ngan

For practical purposes, we express a general BRDF as a sum of several terms, or components. On this slide you see a plot of the BRDF of an acrylic white paint (from the supplemental material for [Ngan 2002]). The BRDF is plotted for four different outgoing directions. For each fixed outgoing direction, the BRDF becomes a function of only the incoming direction – this is the BRDF lobe. In our example, there is a prominent diffuse component and clearly visible specular component on top of that. The diffuse BRDF component is a constant function, independent of the incoming and outgoing direction. It corresponds to the ‘color’ of an object. The specular BRDF component has significant values only in a narrow cone, often around the direction of perfect mirror reflection. The specular component adds the highlight on the sphere.



Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

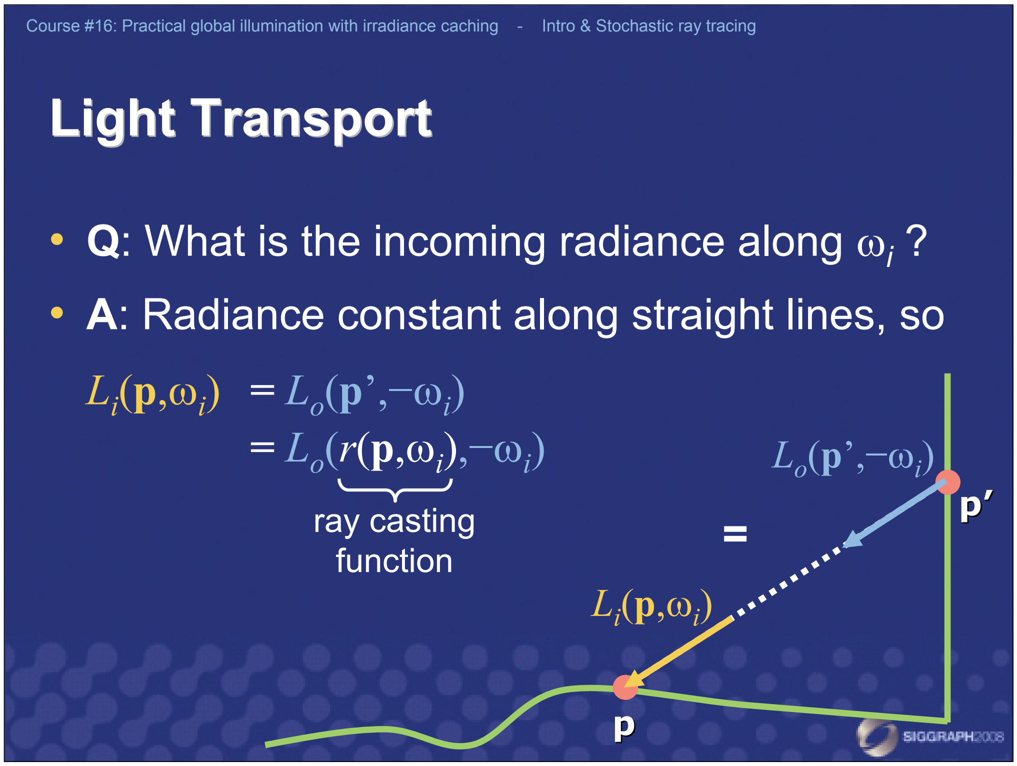
Illumination Integral

- Total amount of light reflected to ω_o:

$$L_o(\mathbf{p},\omega_o) = L_e(\mathbf{p},\omega_o) + \int L_i(\mathbf{p},\omega_i)\, f_r(\omega_i,\mathbf{p},\omega_o)\, \cos\theta_i \, \mathrm{d}\omega_i$$

ω_o
ω_i
p
SIGGRAPH2008

Let us go back to image rendering. We want to compute the amount of light going from **p** towards the camera. Assuming we know how much light is coming to **p** from all possible directions (the incoming hemisphere), the reflected light can be computed by evaluating the illumination integral: For each incoming direction, we multiply the radiance from that direction by the BRDF and the cosine term. To get the total amount of reflected light, we sum (integrate) these contributions over all incoming directions. Then we add the self emitted light at **p** (in case **p** is on the light source) and we have the total amount of light going from **p** towards the camera.



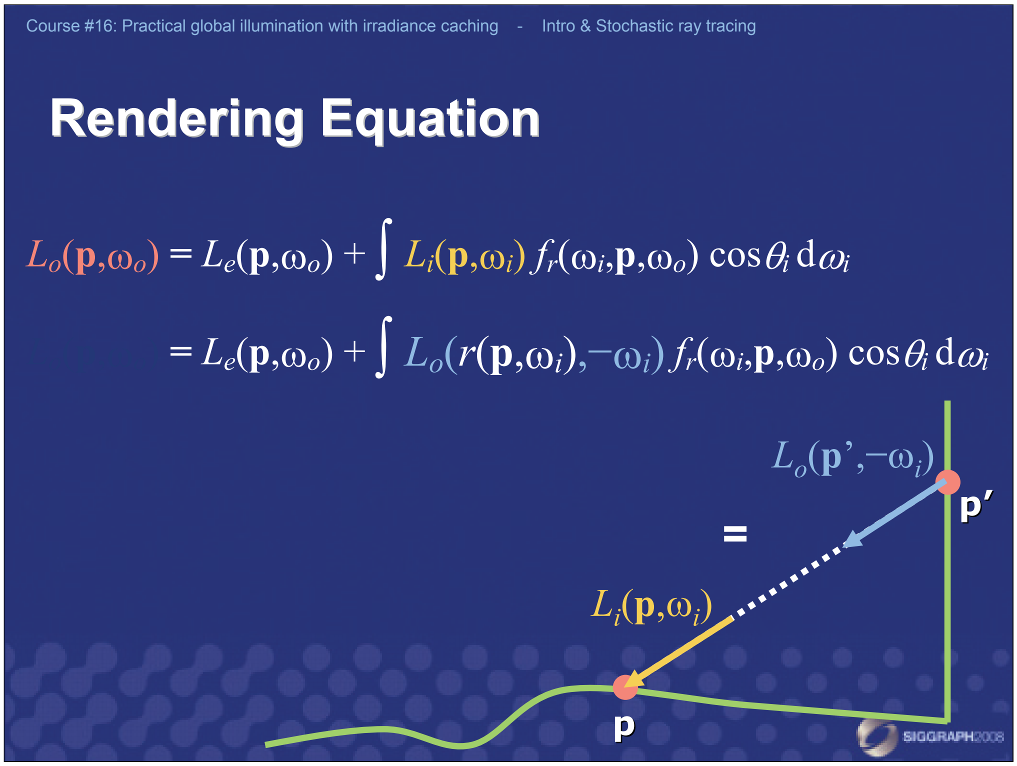
Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Light Transport

- **Q**: What is the incoming radiance along ω_i ?
- **A**: Radiance constant along straight lines, so

$$\begin{aligned} L_i(\mathbf{p},\omega_i) &= L_o(\mathbf{p}',-\omega_i) \\ &= L_o(\underbrace{r(\mathbf{p},\omega_i)}_{\text{ray casting function}},-\omega_i) \end{aligned}$$

We made an important assumption in computing the illumination integral. We assumed to know how much light is coming to **p** from each direction. But how do we find this out? Taking advantage of the fact that *radiance does not change along straight lines*, we see that the incoming radiance from direction w_i is equal to the outgoing radiance at a point **p**' visible from **p** along w_i.



Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

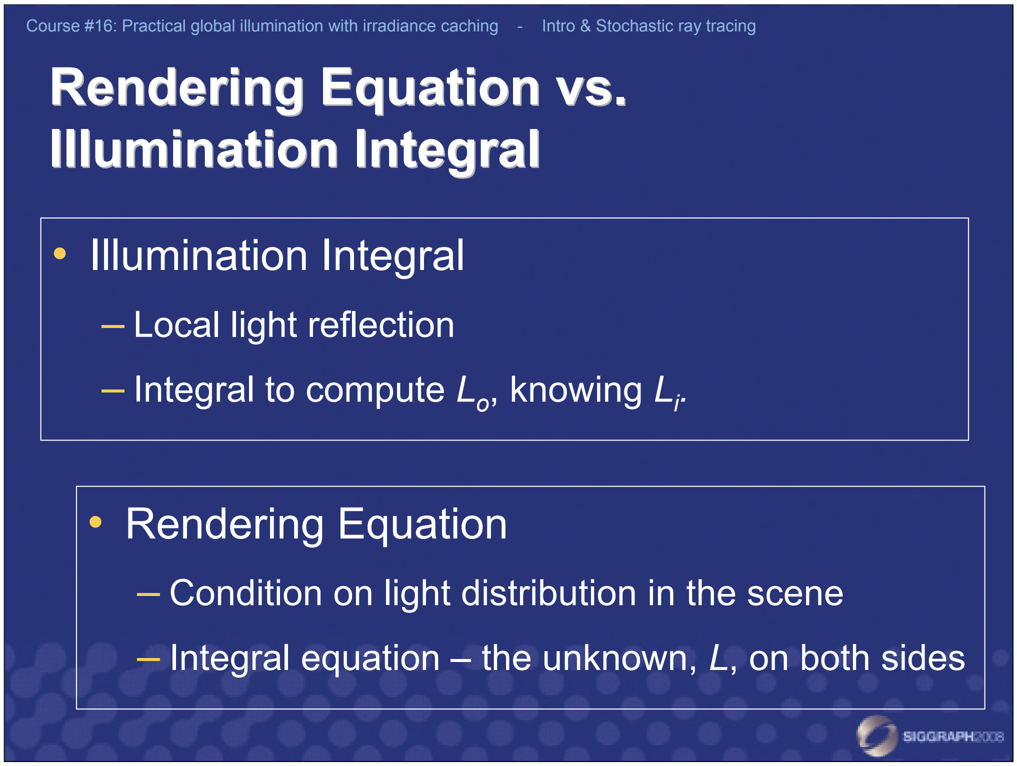
Rendering Equation

$$L_o(\mathbf{p},\omega_o) = L_e(\mathbf{p},\omega_o) + \int L_i(\mathbf{p},\omega_i)\, f_r(\omega_i,\mathbf{p},\omega_o) \cos\theta_i \, d\omega_i$$

$$= L_e(\mathbf{p},\omega_o) + \int L_o(r(\mathbf{p},\omega_i),-\omega_i)\, f_r(\omega_i,\mathbf{p},\omega_o) \cos\theta_i \, d\omega_i$$

$L_o(\mathbf{p}',-\omega_i)$
p′
=
$L_i(\mathbf{p},\omega_i)$
p

Plugging our knowledge of light transport to the illumination integral yields *the rendering equation* [Kajiya 1986].

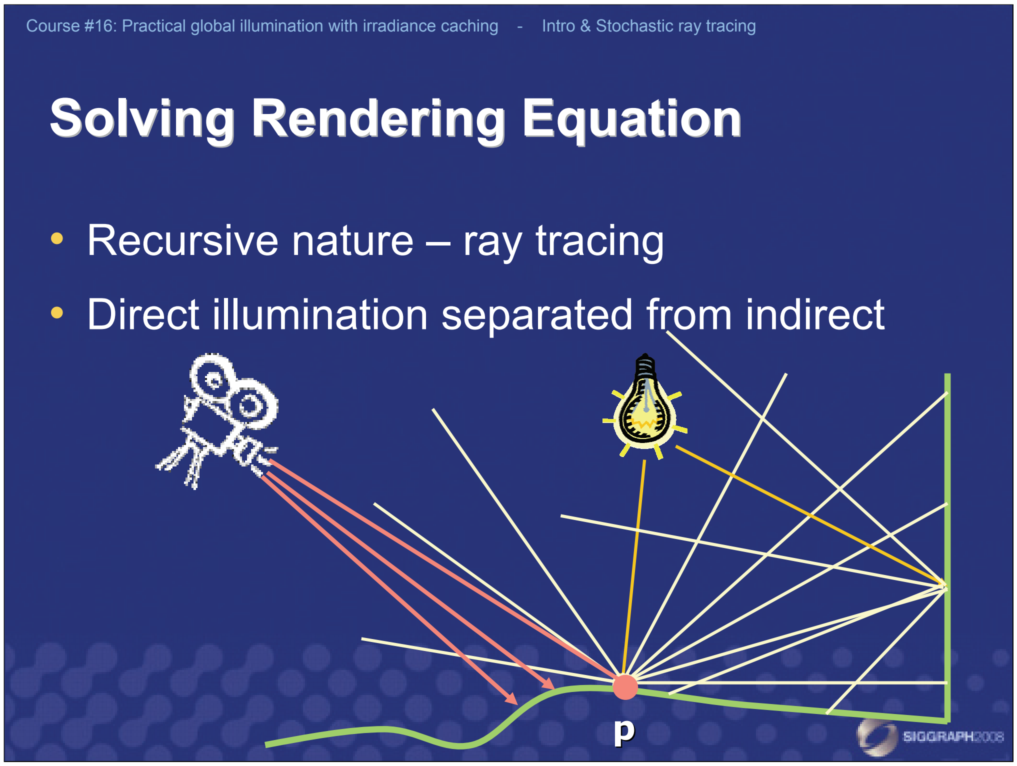


Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Rendering Equation vs. Illumination Integral

- Illumination Integral
  - Local light reflection
  - Integral to compute L_o, knowing L_i.
- Rendering Equation
  - Condition on light distribution in the scene
  - Integral equation – the unknown, L, on both sides

Unlike the illumination integral, which is simply a formula that computes reflected light from incident light and therefore has local character, the rendering equation has a more profound meaning. It is truly an ‘equation’ – the unknown quantity, radiance L, – appears on both sides. The rendering equation expresses a condition on equilibrium light distribution in the whole scene.



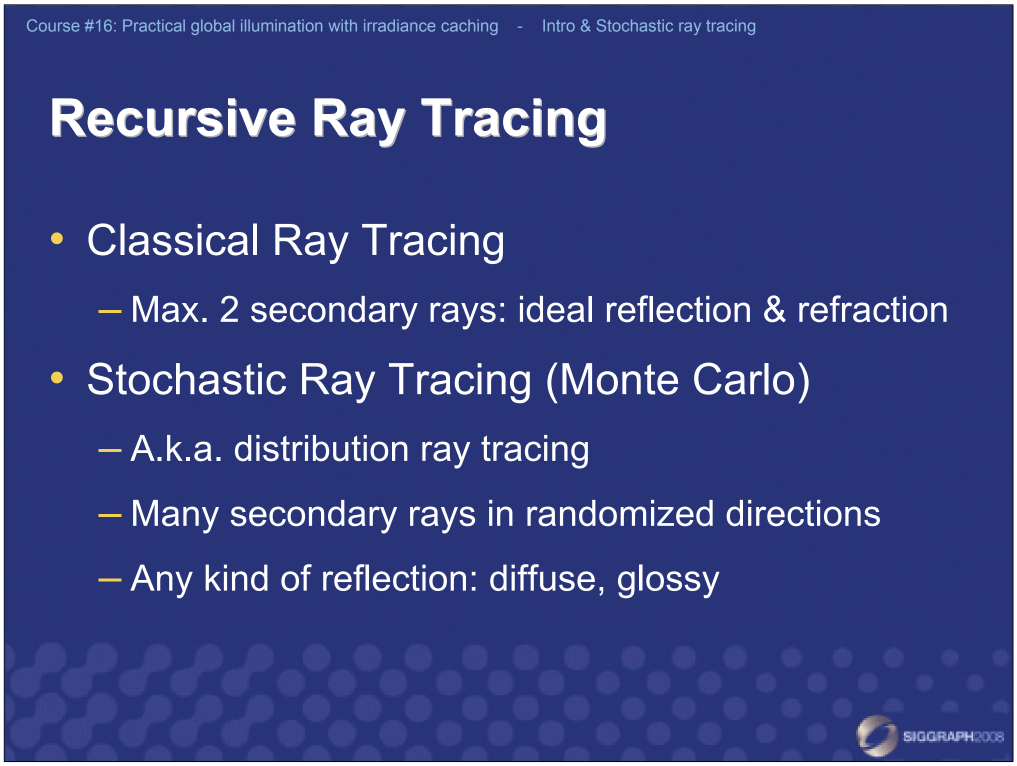
Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Solving Rendering Equation

- Recursive nature – ray tracing
- Direct illumination separated from indirect

Although the rendering equation is at the heart of all theoretical analyses of global illumination, our solution strategy – recursive ray tracing – is much more reminiscent of the illumination integral. At each point of the scene, we want to estimate the outgoing light. In order to do this, we want to numerically evaluate the illumination integral. So we shoot secondary rays, which are ‘samples’ of the incoming radiance. These samples are then multiplied by the BRDF and averaged to numerically estimate the outgoing radiance.

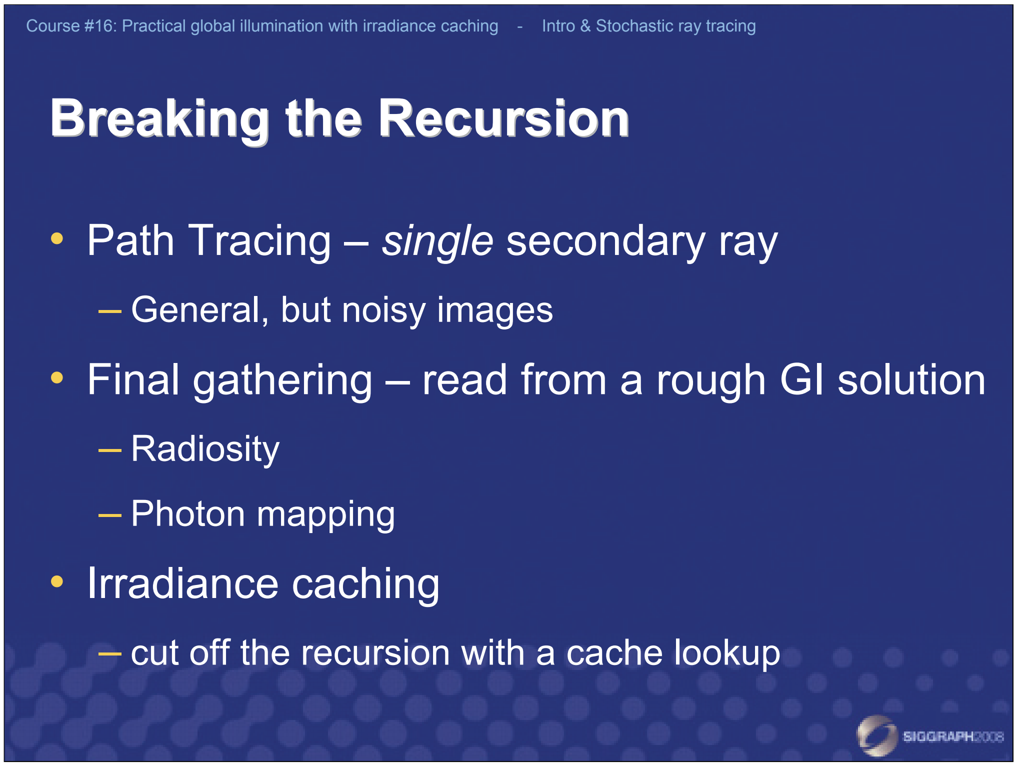
For each of these secondary rays, we need to evaluated the outgoing radiance at a point where they intersect the scene. So we use the same procedure recursively again and again. The recursion can be limited since each light reflection takes away some of the transported light energy. So after couple of reflections, the contribution of light to the image becomes insignificant.



Recursive Ray Tracing

- Classical Ray Tracing
  - Max. 2 secondary rays: ideal reflection & refraction
- Stochastic Ray Tracing (Monte Carlo)
  - A.k.a. distribution ray tracing
  - Many secondary rays in randomized directions
  - Any kind of reflection: diffuse, glossy

In the classical ray tracing, we shoot at most two secondary rays – in the direction of ideal specular reflection and refraction. That means we disregard indirect illumination on anything but ideal specular reflectors or refractors (such as water, glass etc.) If we want to add indirect light on surfaces of arbitrary reflective characteristics (glossy, diffuse), we need to send many rays to estimate the illumination integral – this is most often referred to as *distribution ray tracing*.



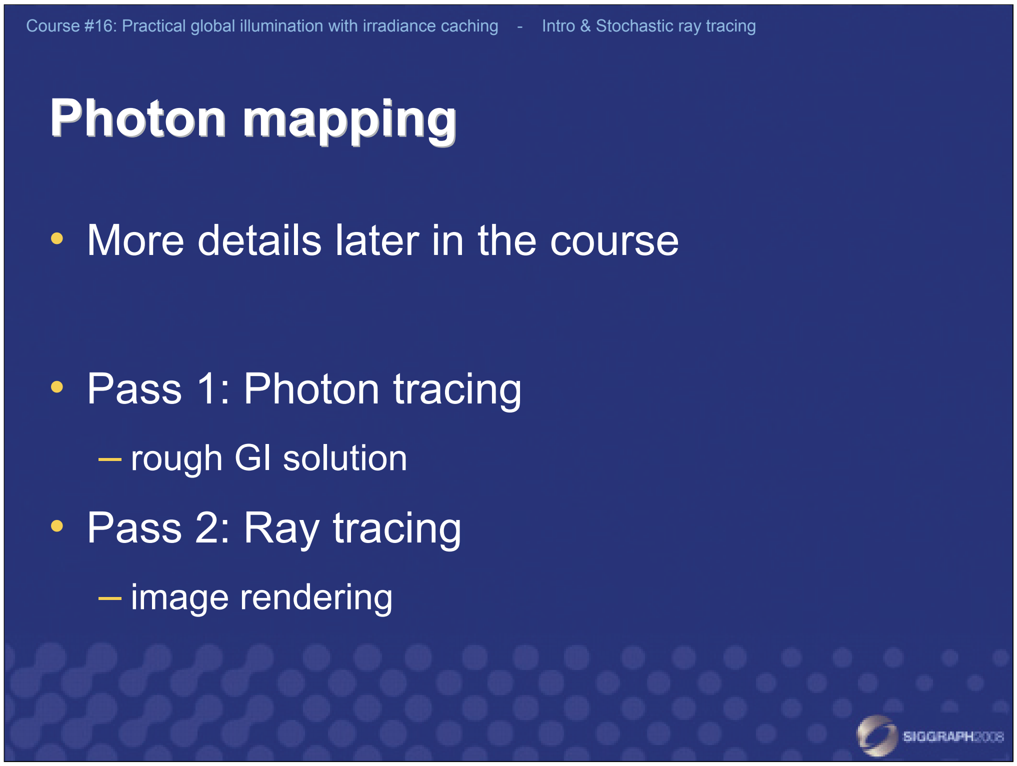
Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Breaking the Recursion

- Path Tracing – *single* secondary ray
  - General, but noisy images
- Final gathering – read from a rough GI solution
  - Radiosity
  - Photon mapping
- Irradiance caching
  - cut off the recursion with a cache lookup

One serious problem of distribution ray tracing is the explosion of the number of rays due to the recursion, where each ray is split into many rays upon each reflection. One way to combat this problem is path tracing where only one single ray is generated at each reflection. The rays then form linear ‘paths’ instead of the ray tree in distribution ray tracing. Path tracing is the original strategy that Jim Kajiya [1986] proposed for solving the rendering equation. It is very general and simple to implement. However, thousands of paths have to be traced through each image pixel in order to compute images without visible noise, which is time-consuming.

Another way to break the recursion is to compute a rough global illumination solution in a pre-process and store it in the scene. Then, as we trace the camera rays, we can read the rough GI solution instead of continuing the recursion. In practice, the most common way of doing this is to use photon mapping.

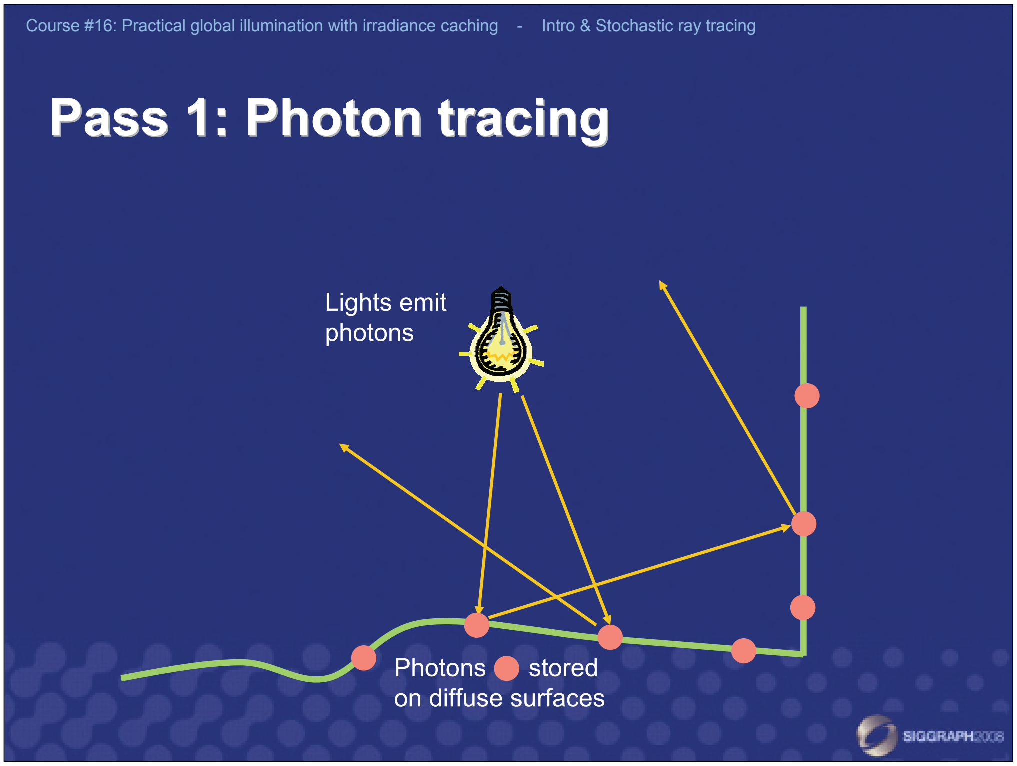


Photon mapping

- More details later in the course
- Pass 1: Photon tracing
  - rough GI solution
- Pass 2: Ray tracing
  - image rendering

SIGGRAPH2008

Photon mapping works in two passes. In the first pass – photon tracing – a rough GI solution is created in form of ‘photons’ distributed in the scene. The second pass is distribution ray tracing as described previously, where recursion is limited by reading the rough GI solution from the photon map.

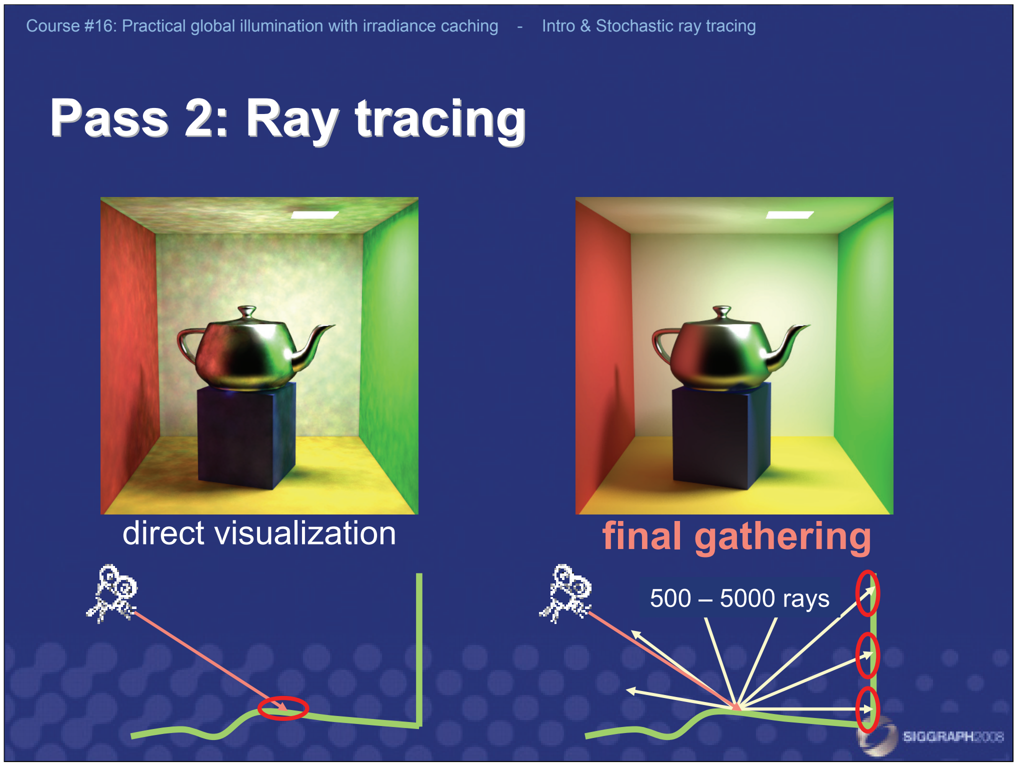


Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Pass 1: Photon tracing

Lights emit
photons

Photons ● stored
on diffuse surfaces



Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

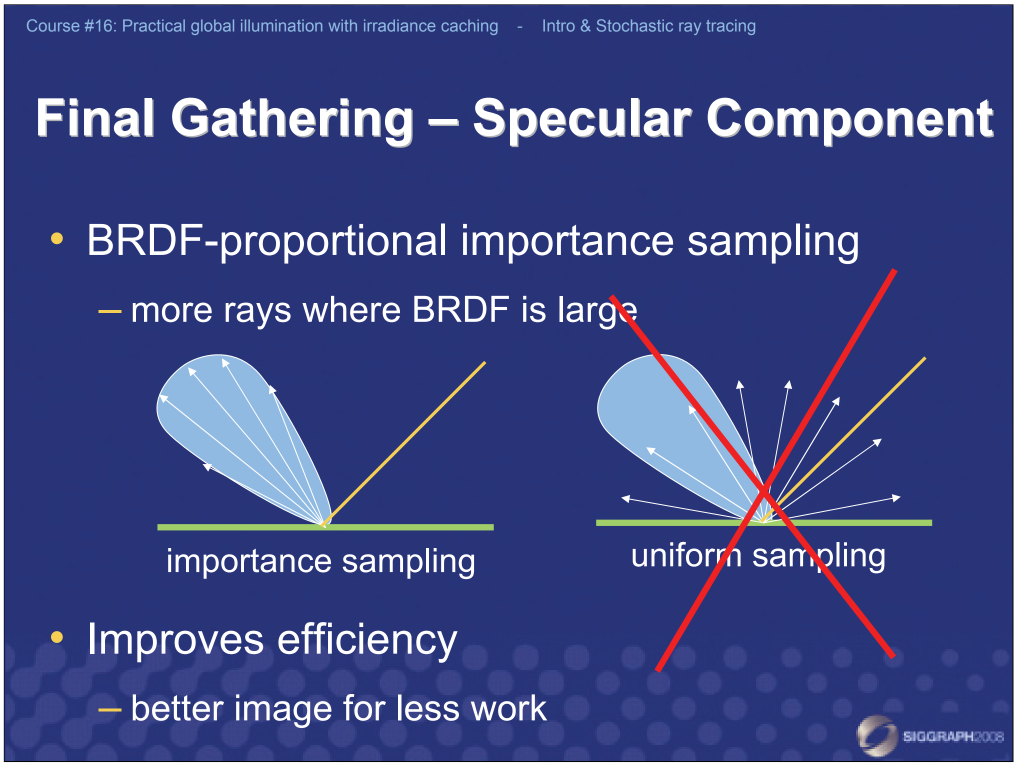
Pass 2: Ray tracing

direct visualization

final gathering

In ray tracing with photon maps, one possibility is to read the rough GI solution from the photon map immediately as a primary ray hits a surface. However, this gives a ‘splotchy’ artifacts in the images, precisely because the GI solution in the photon map is very rough.

A better way is to perform one level of distribution ray tracing and read the photon map after the first reflection. Shooting many secondary rays averages out any artifacts and the result is a clean image with global illumination. This technique is called *final gathering*. Final gathering as described so far is *slow*: for each of the thousands or millions primary rays, hundreds or thousands gather rays have to be shot.



Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Final Gathering – Specular Component

- BRDF-proportional importance sampling
  - more rays where BRDF is large

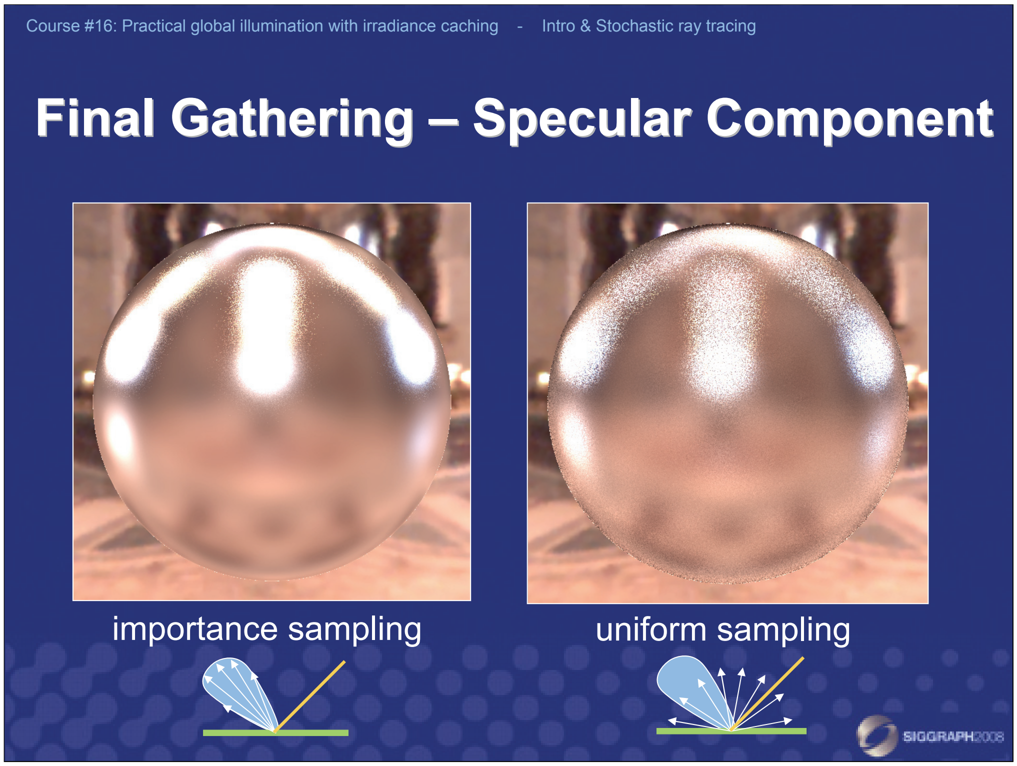
importance sampling

~~uniform sampling~~

- Improves efficiency
  - better image for less work

On specular surfaces, an extremely powerful way to reduce the cost of distribution ray tracing – and final gathering – is *importance sampling*. Since the BRDF lobe corresponding to the (known) outgoing direction has significant values only in a narrow cone, there is no use wasting time in sampling the whole hemisphere uniformly. It is much better to concentrate the rays within that cone.

Importance sampling improves efficiency – we get the same quality results with much fewer rays (or much better quality with the same number of rays). The more specular surface, the more effective importance sampling.

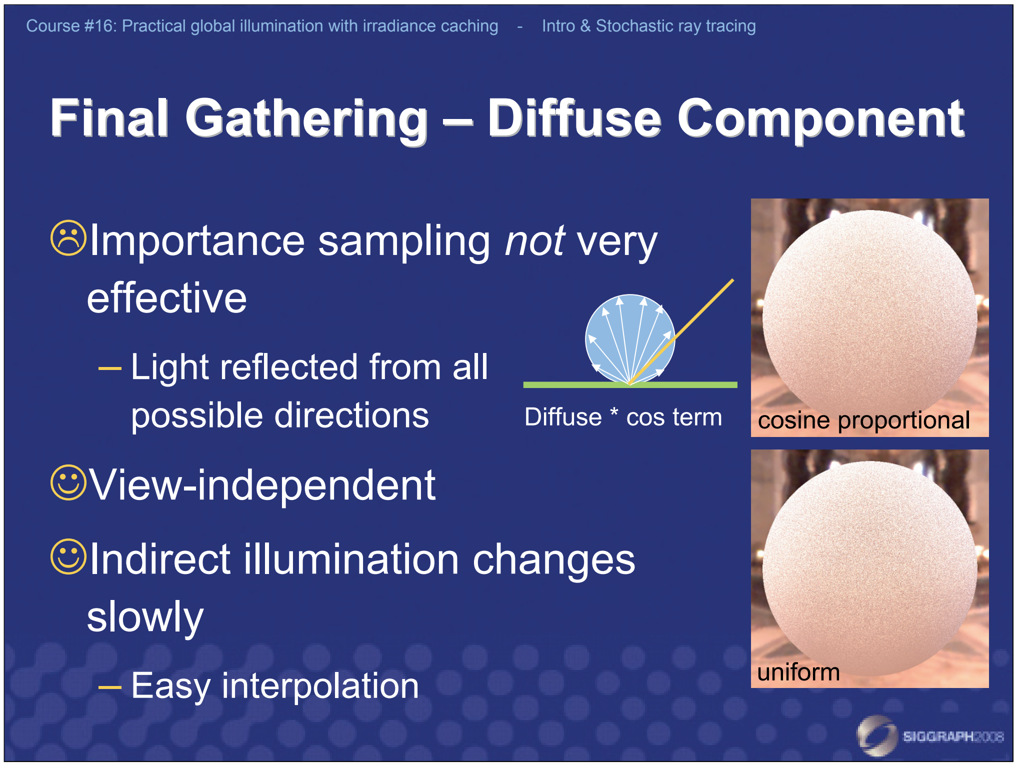


Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Final Gathering – Specular Component

importance sampling

uniform sampling



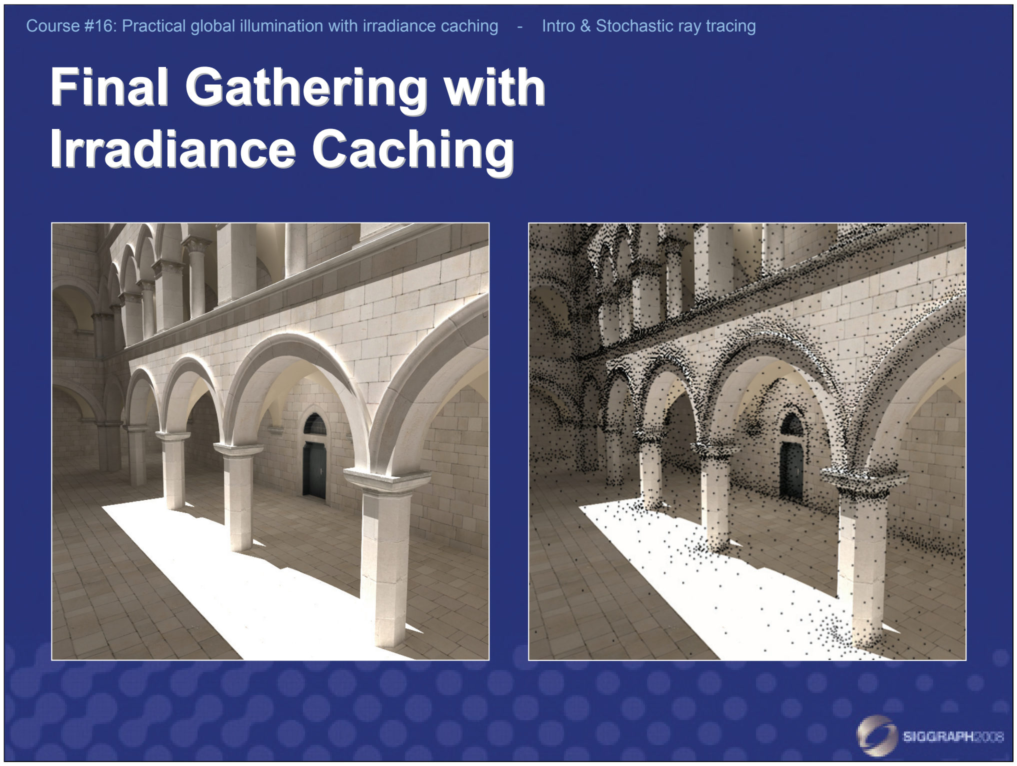
Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Final Gathering – Diffuse Component

- ☹Importance sampling *not* very effective
  - Light reflected from all possible directions
- ☺View-independent
- ☺Indirect illumination changes slowly
  - Easy interpolation

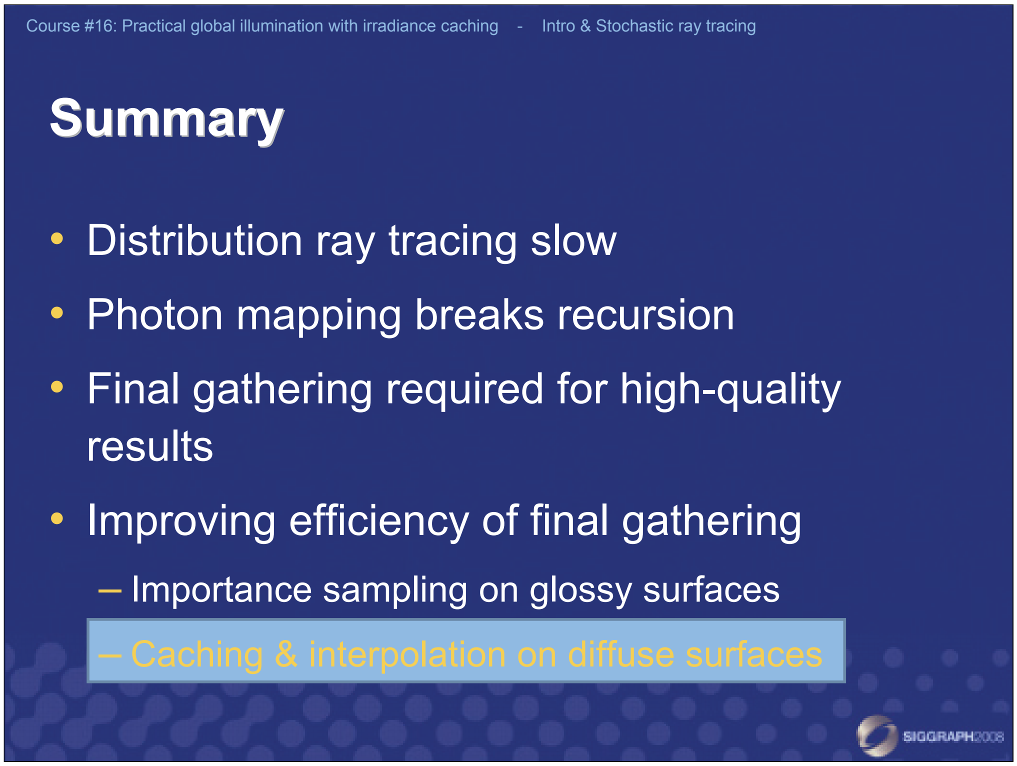
Diffuse * cos term
cosine proportional
uniform

Unfortunately, importance sampling isn't very effective on diffuse surfaces, since light is reflected nearly equally from all incoming directions. Two good news help us make distribution ray tracing on diffuse surfaces efficient: 1) indirect illumination is view-independent and 2) it varies slowly over surfaces. This makes it relatively easy to compute indirect illumination sparsely in the scene and interpolate.



Final Gathering with Irradiance Caching

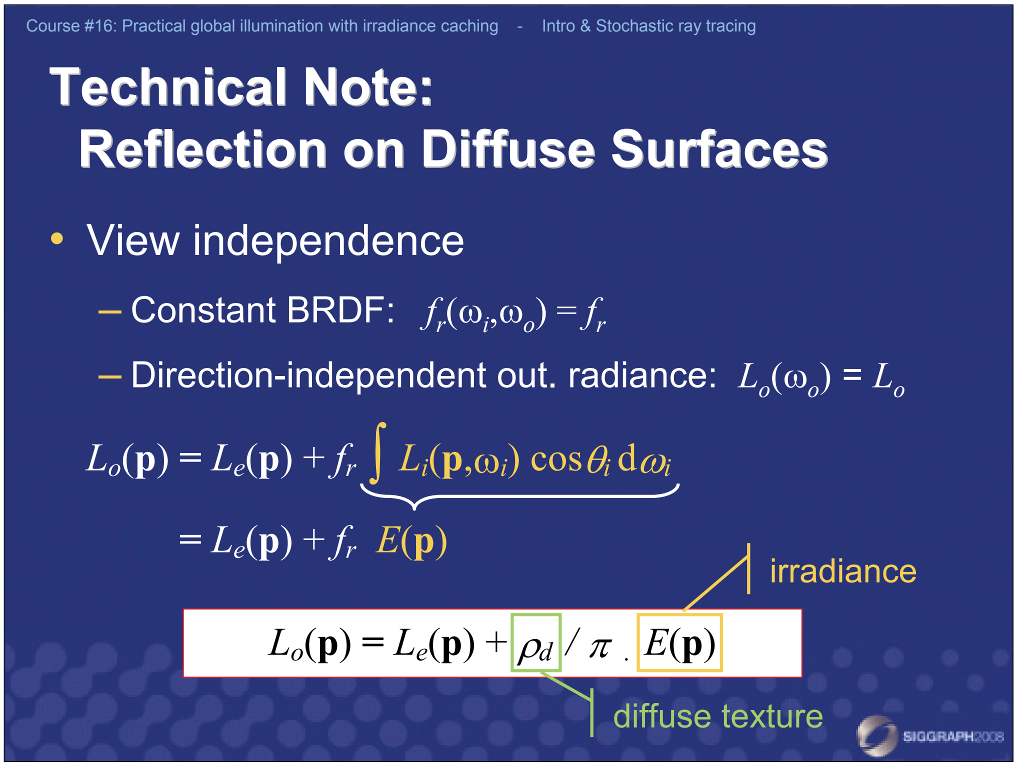
Shown on the left are the positions at which the costly final gathering computation was performed. The computed indirect illumination values were then used for interpolation for the remaining positions.



Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Summary

- Distribution ray tracing slow
- Photon mapping breaks recursion
- Final gathering required for high-quality results
- Improving efficiency of final gathering
  - Importance sampling on glossy surfaces
  - Caching & interpolation on diffuse surfaces



Course #16: Practical global illumination with irradiance caching - Intro & Stochastic ray tracing

Technical Note: Reflection on Diffuse Surfaces

- View independence
  - Constant BRDF: $f_r(\omega_i,\omega_o) = f_r$
  - Direction-independent out. radiance: $L_o(\omega_o) = L_o$

$$L_o(\mathbf{p}) = L_e(\mathbf{p}) + f_r \underbrace{\int L_i(\mathbf{p},\omega_i)\cos\theta_i \, \mathrm{d}\omega_i}$$

$$= L_e(\mathbf{p}) + f_r \; E(\mathbf{p})$$

irradiance

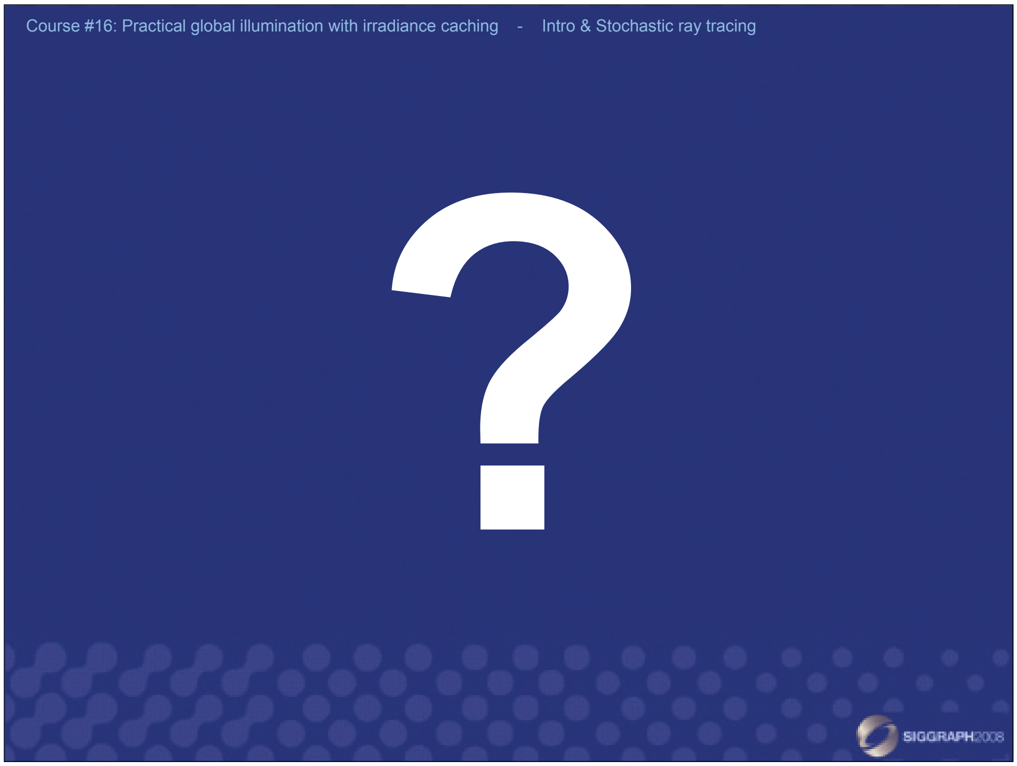
$$\boxed{L_o(\mathbf{p}) = L_e(\mathbf{p}) + \boxed{\rho_d} / \pi \, . \, \boxed{E(\mathbf{p})}}$$

diffuse texture

The view independence of diffuse materials means that the outgoing radiance $L_o(\omega_o)$ is independent of the direction, ω_o. This implies that the BRDF is constant in both the incoming and the outgoing directions and, in the illumination integral, it can be taken out of the integration. What remains is the integral of the incoming radiance multiplied by the cosine term – this is the irradiance *E*. Irradiance expresses the area density of light energy and is measured in Watts per square meter.

The BRDF of the diffuse surface is uniquely determined by one color triple – the reflectance ρ_d. Reflectance of a real surface must be between zero and one so that reflection conserves energy. This is why the radiometric quantity that corresponds to the diffuse texture is the reflectance.

Beware! Reflectance (=texture value) must be divided by π to get the BRDF value.



?