Introduction to Ray Tracing

RAJESH SHARMA
Walt Disney Animation Studios

Thank you to ACM SIGGRAPH!

Pol Jeremias-Vila : SIGGRAPH 2021 Chair

Tomasz Bednarz: Frontiers Program Chair

Alex Bryant: Student Volunteers Chair

Tim Hendrickson: Digital Marketing Manager

Student Volunteers:
Rogelio, Trinity, Aurora, Emily, Hunter & Kendra

SIGGRAPH 2021

3

Ray Tracing

Rajesh Sharma

Course Outline

-✔- Intro, Model, Sampling
-✔- Rays, Intersections
-✔- Scene, Recursion
-- Materials, BRDF
-- Importance Sampling, Lights
-- Systems View: Integrators, Accelerators
-- Wrap up, Learn more

Today

- Guest: Brent Burley
- Recap, Q&A, HW
- Reflection
- Materials

Housekeeping

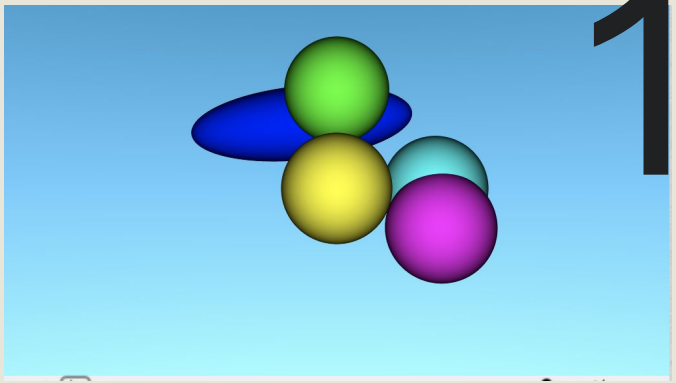
- Link to today's slides and shaderToys:
  - Log in to your google drive
  - Google drive folder: https://bit.ly/3viTHez
  - Code: https://www.shadertoy.com/user/xarmalarma
- Use the chat to ask questions, help others
- After the lecture: @xarmalarma, #siggraph2021

Brent Burley

Brent Burley is a Principal Engineer at Walt Disney Animation. During his 25 year tenure at the company, Brent has made many groundbreaking contributions to computer graphics including: Ptex, SeExpr, and DisneyBRDF. Brent is also the originator of the Hyperion renderer.

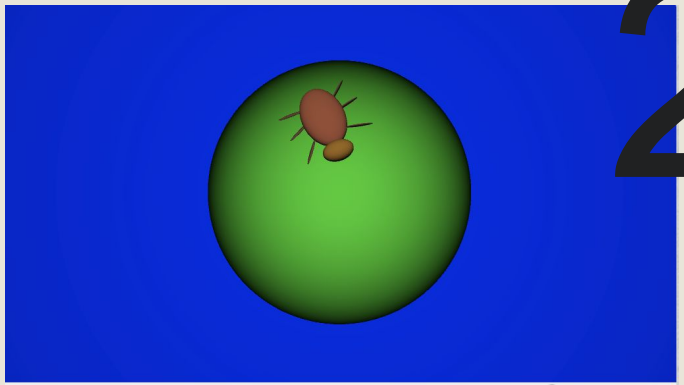
Choose Winners

1



https://www.shadertoy.com/view/ftlGzB

2



https://www.shadertoy.com/view/7lfGWS

3

https://www.shadertoy.com/view/7lf3zs

4

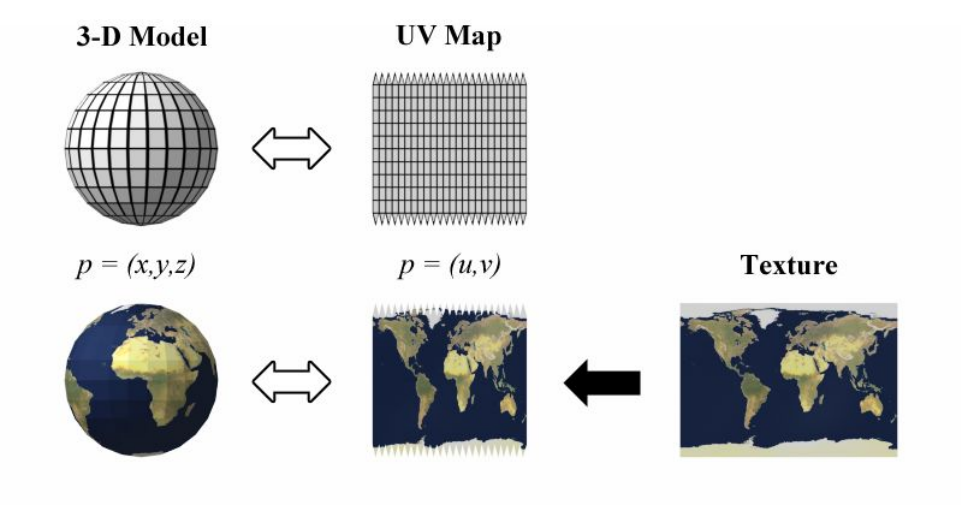
https://www.shadertoy.com/view/Ntl3WS

Texture Mapping

So far we are just setting a color for the sphere.

We can modulate the color by any signal:

- Wrap a video or an image texture or even sound!



$$u = 0.5 + \frac{\arctan2(d_x, d_z)}{2\pi},$$
$$v = 0.5 - \frac{\arcsin(d_y)}{\pi}.$$

Source: https://en.wikipedia.org/wiki/UV_mapping

Simple Shading: Light anywhere

Lambert's Cosine Law

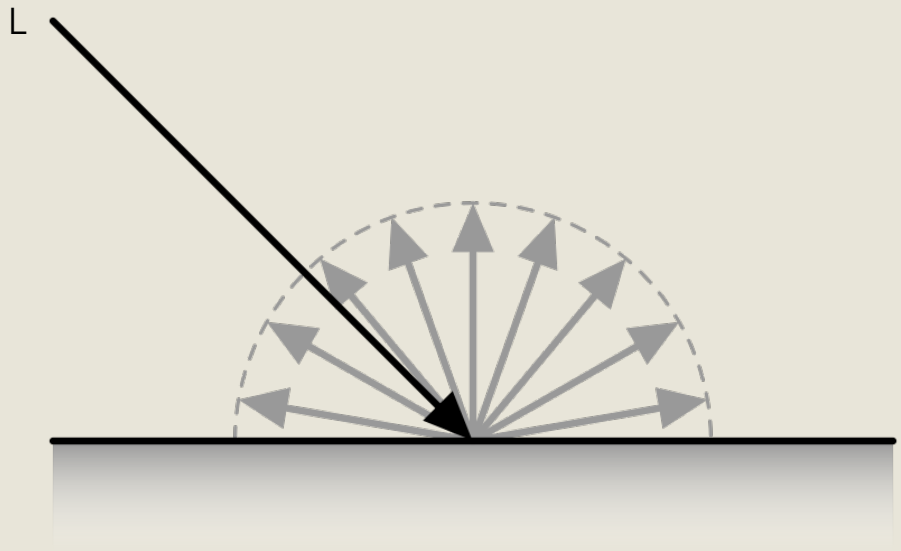
- Diffuse, Lambertian
- View Independent

Smaller the angle, more the intensity:

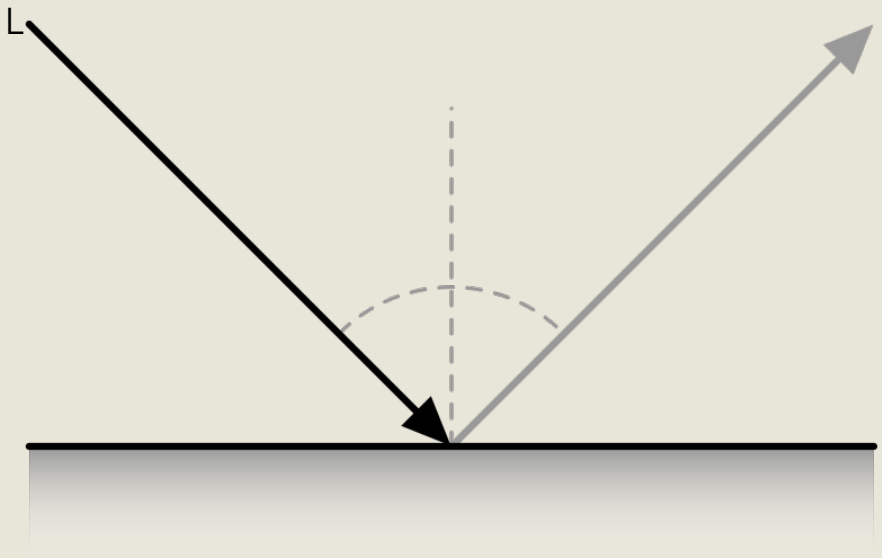
`dot(L^, N^) = cos(angle)`

Materials

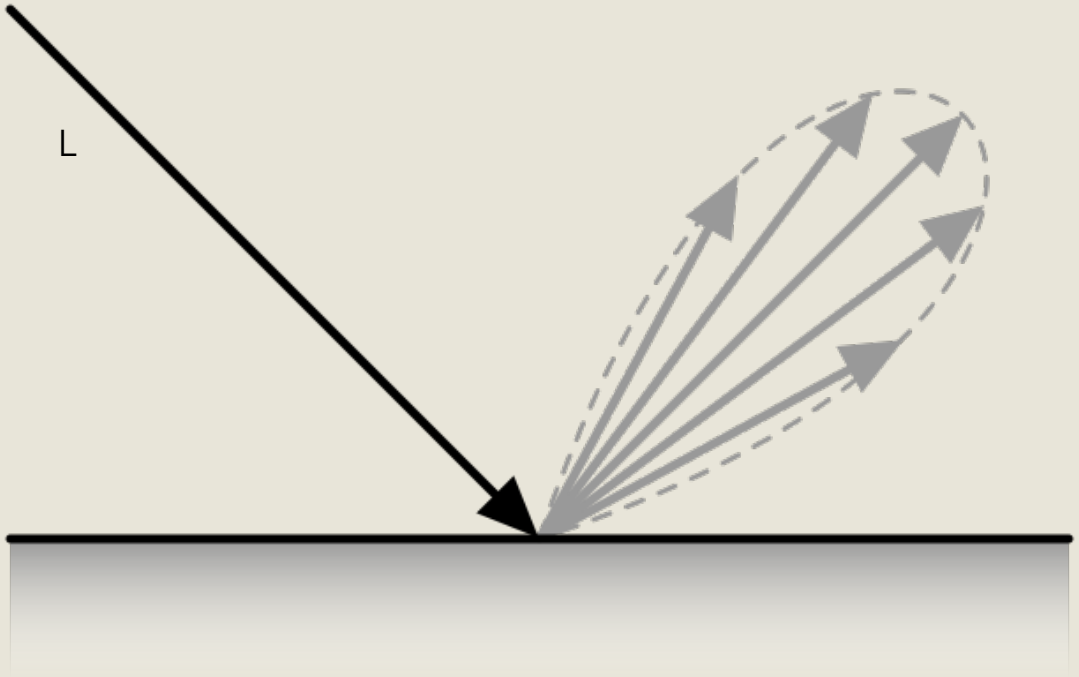
- So far our material is just a color and possibly a texture
- We have diffuse (Lambertian) surfaces



Materials - Mirror

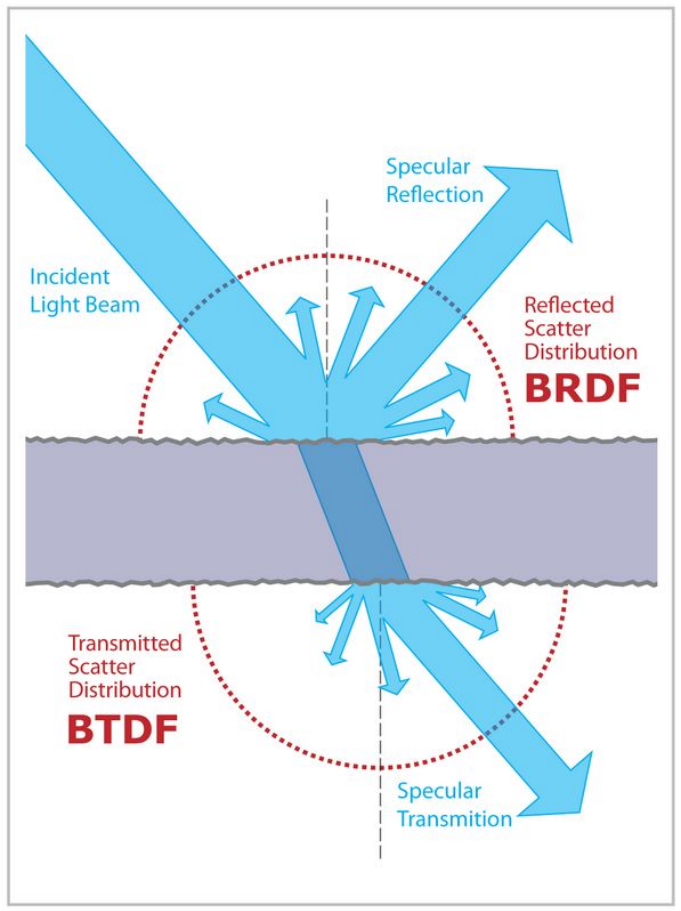


Materials - Glossy



Materials - Things can get complicated

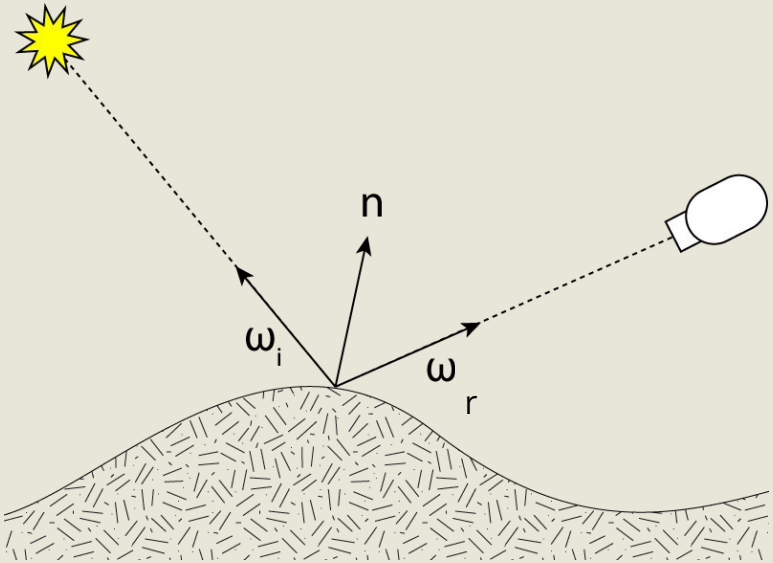
L



Mirror - Specular
Metal - Glossy
Skin - Subsurface
Glass - Transparent
Plaster, Paper - Diffuse

Materials - Simplify

$$f_r(\omega_i, \omega_r) = \frac{\mathrm{d}\, L_r(\omega_r)}{\mathrm{d}\, E_i(\omega_i)} = \frac{\mathrm{d}\, L_r(\omega_r)}{L_i(\omega_i) \cos\theta_i \; \mathrm{d}\, \omega_i}$$

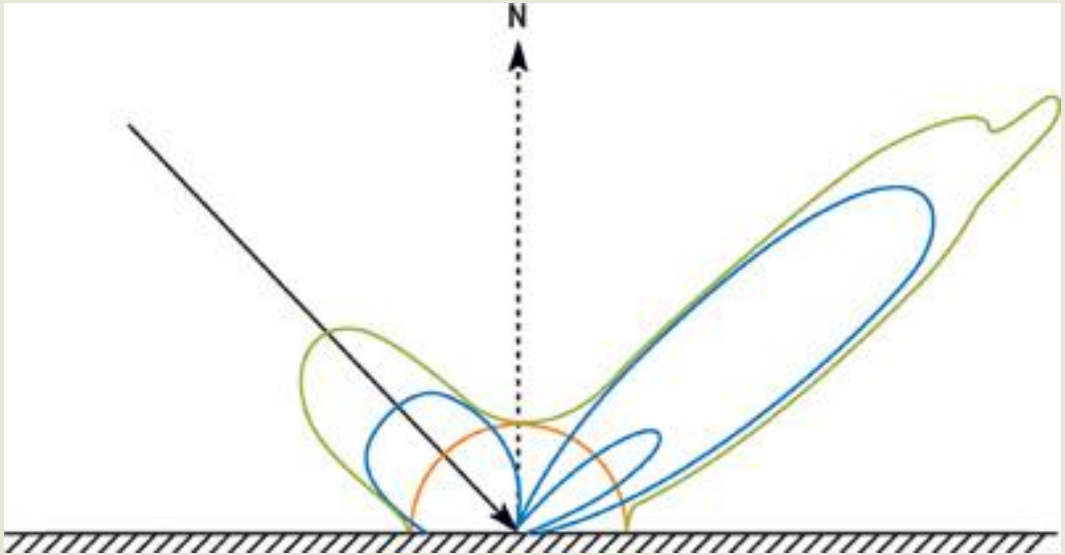


For Diffuse of Lambertian: wr doesn't matter

So, the BRDF in that case is simply the reciprocal of the dot product of normal and incident direction.

Materials - BRDFs for different materials

$$f_r(\omega_i, \omega_r) = \frac{\mathrm{d}\, L_r(\omega_r)}{\mathrm{d}\, E_i(\omega_i)} = \frac{\mathrm{d}\, L_r(\omega_r)}{L_i(\omega_i) \cos\theta_i \;\mathrm{d}\,\omega_i}$$

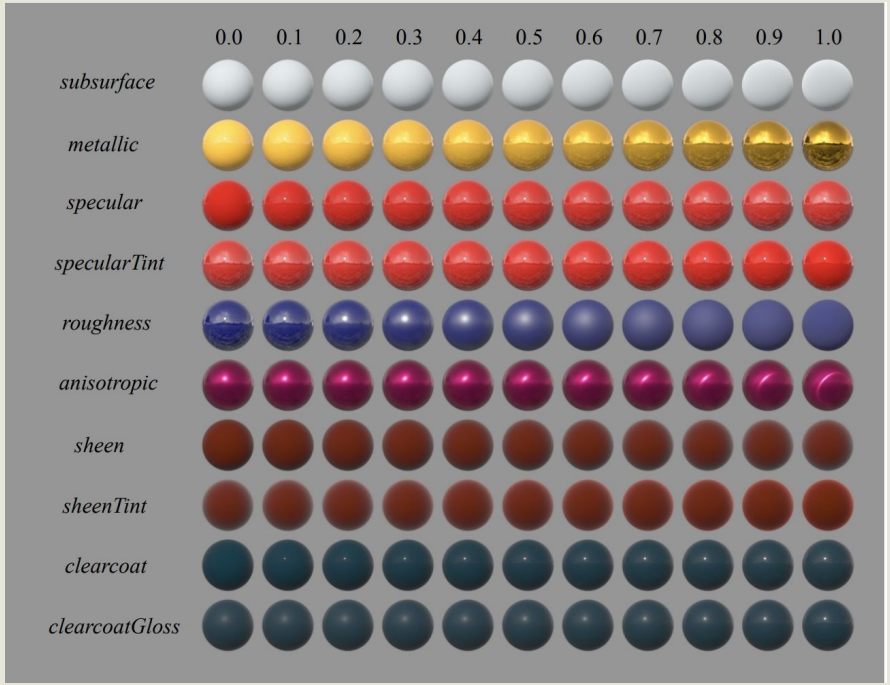


Multiple lobes

https://developer.nvidia.com/sites/all/modules/custom/gpugems/books/GPUGems/elementLinks/fig18-02.jpg

Materials - BRDFs for different materials

$$f_r(\omega_i, \omega_r) = \frac{\mathrm{d}\, L_r(\omega_r)}{\mathrm{d}\, E_i(\omega_i)} = \frac{\mathrm{d}\, L_r(\omega_r)}{L_i(\omega_i) \cos\theta_i \, \mathrm{d}\, \omega_i}$$



https://www.disneyanimation.com/publications/physically-based-shading-at-disney/

Hands-on

- ★ Log in to your google drive
- ★ Make a shortcut to: https://bit.ly/3viTHez
- ★ Create an account on shadertoy.com
- ★ Fork a copy of:
  - ○ https://www.shadertoy.com/view/7ts3WN

Next Class

- Complex Materials
- Unifying everything
- Homework:
  - Implement other BRDFs
  - Create interesting scenes, animations
  - @xarmalarma, #siggraph2021

QUESTIONS?

- Chat
- #xarmalarma