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: Karl Li
- Recap, Q&A, HW
- Simple Shading → Light
- Textures
- Multiple Objects
- Complexity

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

Karl Li

Karl is a senior software engineer at Walt Disney Animation Studios working on Disney's Hyperion Renderer. As part of the Hyperion team, he has worked on every Disney Animation theatrical release starting from Zootopia.

Karl's areas of interest cover all things production rendering, including topics such as advanced light transport, renderer architectures, raytracing on interesting hardware, volume rendering, surface appearance, and more.

He maintains a popular blog on rendering and technology at:

https://blog.yiningkarlli.com/

Last time: Ray-Sphere Intersection

We get a quadratic equation of the form:

$ax^2 + bx + c = 0$

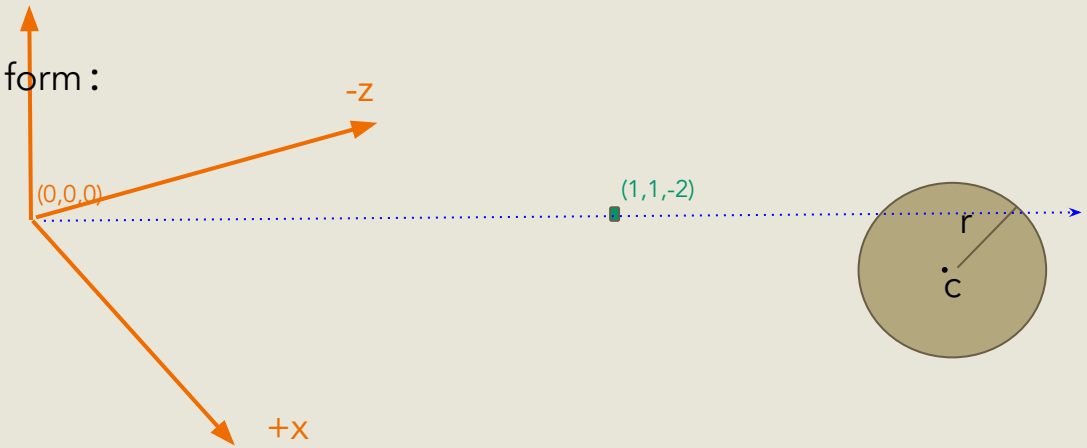
Where,

$a = 1,$

$b = 2*D*(o-c),$

$c = \|o-c\| * r^2$

Discriminant: disc = b^2 - 4a*c



When discriminant is greater than 0, we have two solutions (ray intersects the sphere at two points):
-b + sqrt(disc) and -b - sqrt(disc)

When discriminant is 0, there is only one solution (ray grazes the sphere)
When discriminant is negative, there is no solution (ray misses the sphere)

Solve for t, use the smaller t if it is positive. For negative t, the intersection is 'behind' the camera.

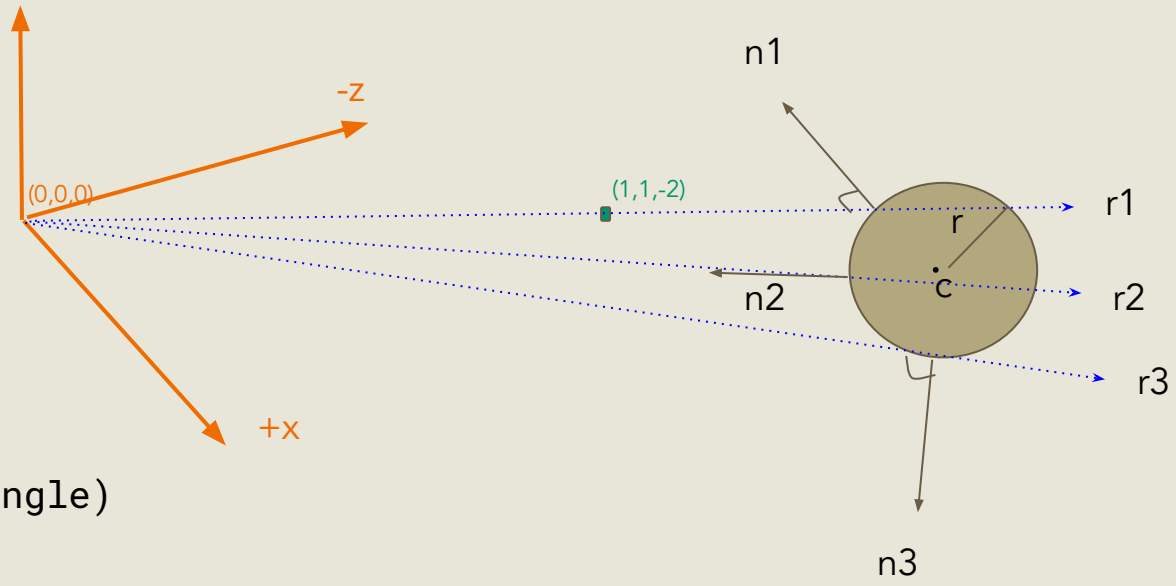
Simple Shading: Light at eye

Three rays:

r1, r2, r3.

Normals at intersection:

n1, n2, n3



Smaller the angle, more the light:

`dot(r, n)` proportional to `cos(angle)`

Simple Shading: Light anywhere

Lambert's Cosine Law

- Diffuse, Lambertian
- View Independent

Smaller the angle, more the intensity:

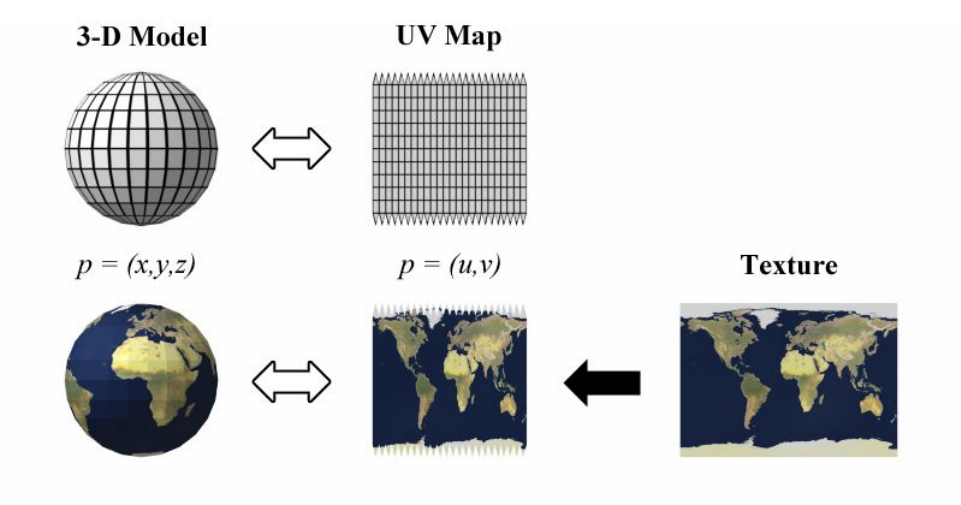
`dot(L^, N^) = cos(angle)`

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

Multiple Objects

- Keep track of nearest object and intersection

Other kinds of objects

- Intersection, hit point
- Normal at intersection

Ray-Triangle

- Möller-Trumbore

Ray-Quadric (Sphere, Ellipsoid, Cylinder, Cone, Hyperboloid)

$$ax^2 + 2bxy + 2cxz + 2dxw + ey^2 + 2fyz + 2gyw + hz^2 + 2izw + jw^2 = 0$$

$$\mathbf{x}^t Q \mathbf{x} = 0,$$

where $\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$, $\mathbf{x}^t = \begin{bmatrix} x & y & z & w \end{bmatrix}$, and $Q = \begin{bmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{bmatrix}$

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

A note on complexity of Ray-tracing

- ★ Width and Height of screen (w, h)
- ★ Number of samples per pixel (spp)
- ★ Number of objects in the scene (N)
- ★ Texture lookup per object (t)
- ★ Color lookup (shading) per object (s)
- ★ I/O for scene (R_s), textures (R_t)

Next Class

- Materials
- Reflection
- Homework:
  - Extend intersection to Quadrics
  - Create interesting scenes, animations
  - @xarmalarma, #siggraph2021

QUESTIONS?

- Chat
- #xarmalarma