Tutorial 8
CS3241 Computer Graphics (AY22/23)

November 1, 2022

Wong Pei Xian

e0389023@u.nus.edu

Question 1

pollev.com/peixian

Given a plane, whose equation is $2x + 4y + 4z - 6 = 0$, and a ray, whose origin is $[2, 0, -5]^T$ and its direction is $[1, 2, 3]^T$:

1. Calculate the location where the ray intersects the plane.
2. Compute the normalized surface normal vector at the intersection point.

Point of intersection

Equation of ray:

$$[2, 0, -5]^T + k[1, 2, 3]^T = [2 + k, 2k, -5 + 3k]^T$$

$$\begin{aligned} 2(2 + k) + 4(2k) + 4(-5 + 3k) - 6 &= 0 \\ 22k &= 22 \\ k &= 1 \end{aligned}$$

Normalized surface normal vector

Equation of plane:

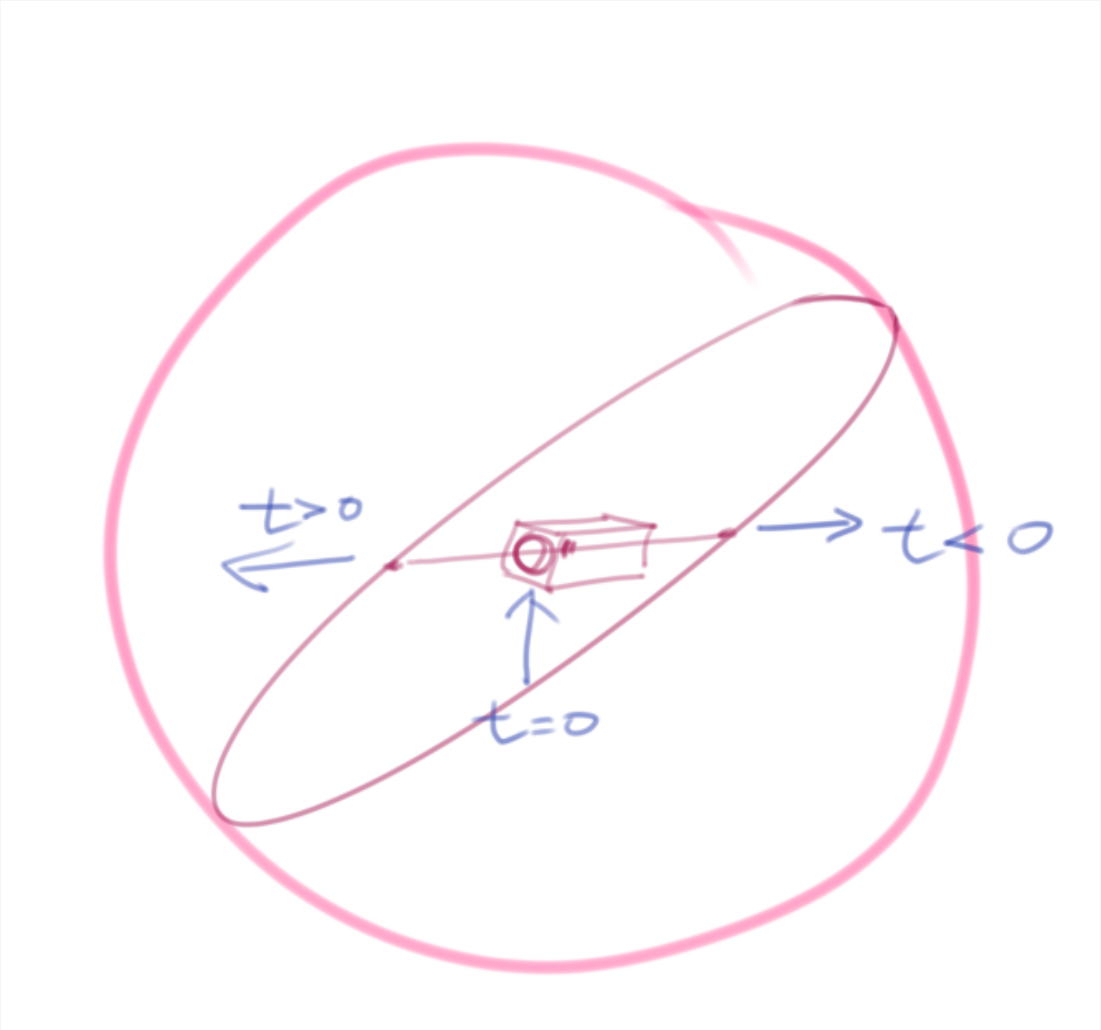
$$2x + 4y + 4z - 6 = 0 \Rightarrow n \cdot (r - r_0) = 0$$

$$n = \text{norm} \begin{pmatrix} 2 \\ 4 \\ 4 \end{pmatrix} = \begin{bmatrix} 2 \\ 4 \\ 4 \end{bmatrix} \times \frac{1}{\sqrt{2^2 + 4^2 + 4^2}} = \begin{bmatrix} 1/3 \\ 2/3 \\ 2/3 \end{bmatrix}$$

Question 2

In Whitted Ray Tracing, when a ray $P(t) = O + tD$ is being intersected with an opaque sphere, we often consider only the intersection at the smaller t value. In what situation should we consider the intersection at the larger t value?

Possible answer: Camera is inside sphere



Question 3

pollev.com/peixian

Suppose there is an enclosed scene, where all surfaces are **opaque** and have materials that have both diffuse and specular components. There are **two point light sources** in the scene.

Assuming we want to render a **200x100 pixels** image of the scene using Whitted Ray Tracing, with **three levels of recursion**, what would be the total number of rays that have to be shot?

Approach

For each pixel ($\times 20000$ pixels) per recursion:

- 2 shadow rays
- Reflected view ray (if recursion level < 3)
- NO refracted view ray ((all surfaces are opaque!))

Without the initial view ray, and we have

$$\begin{aligned} P(0) &= 2 \\ P(r) &= 3 + P(r-1) \\ P(3) &= 3+3+3+2 = 11 \end{aligned}$$

Hence the total number of rays per pixel: $11 + 1 = \mathbf{12}$.

Question 4

Explain why Whitted Ray Tracing cannot produce color bleeding effects (diffuse-to-diffuse reflection).

Local illumination vs Global illumination

Local illumination only accumulates light from **direct light sources**.
We only shoot shadow rays towards the light sources.

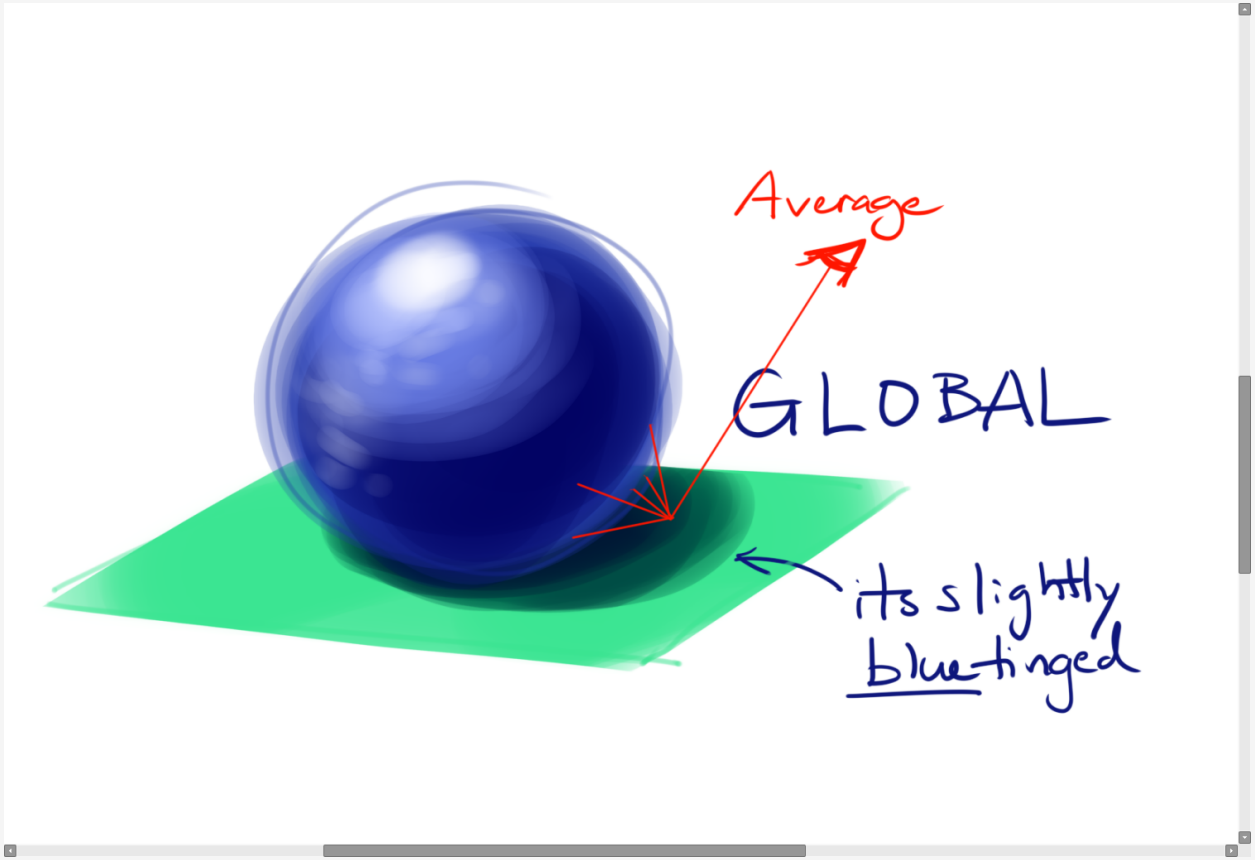
Global illumination accumulates light via **secondary light rays** shot in **all directions** to collect light from diffuse surfaces.

Local illumination

Illustration with whitted ray tracing (green surface is diffuse):

Global illumination

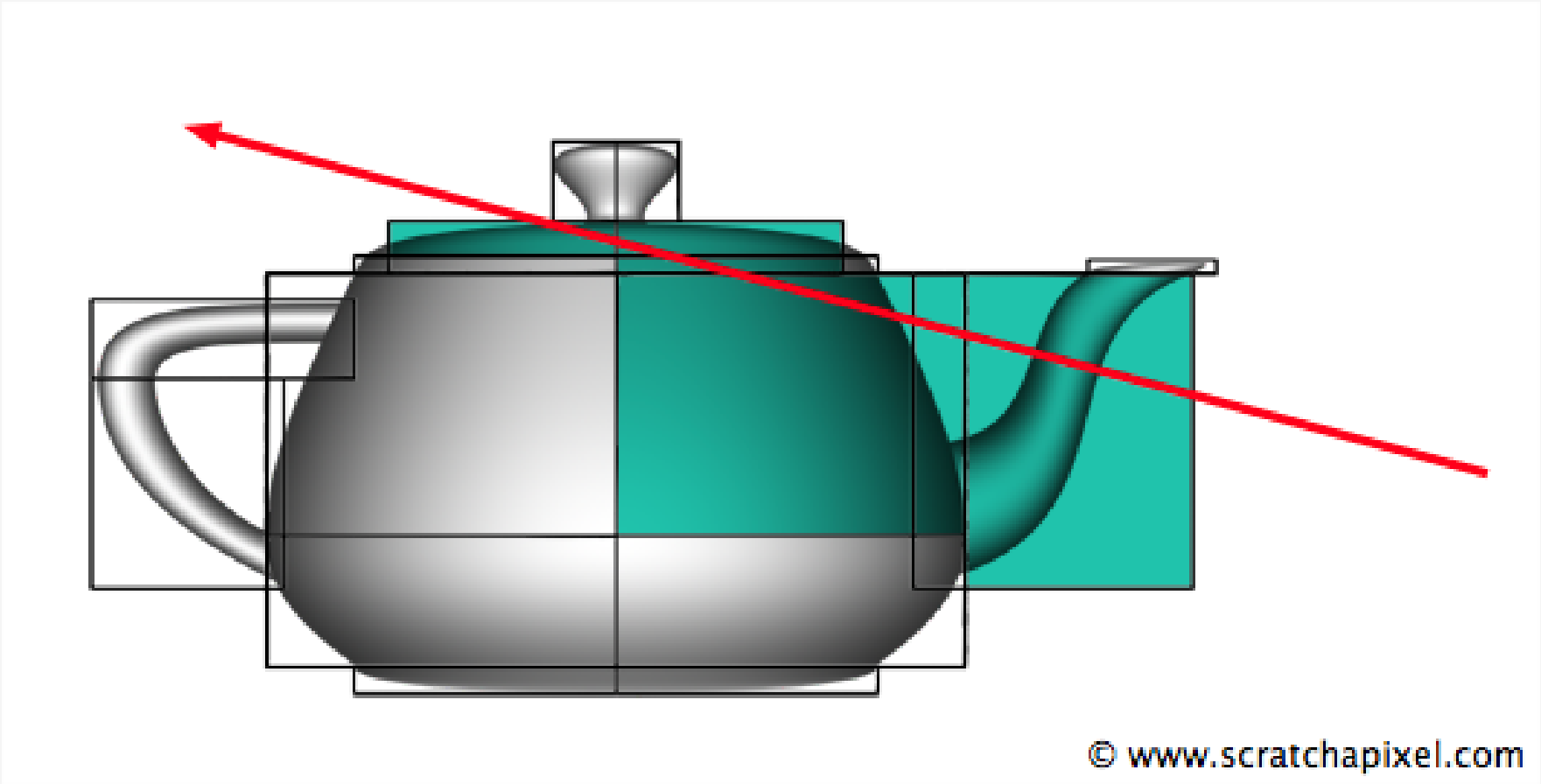
Illustration with global illumination (e.g. path tracing, green surface still diffuse):



Question 5

Explain how the use of bounding volumes can accelerate ray tracing.

Fast elimination



What's the most expensive part of raytracing?

Computing intersections.
e.g. How do you determine at which point does a ray intersect a parametric surface?

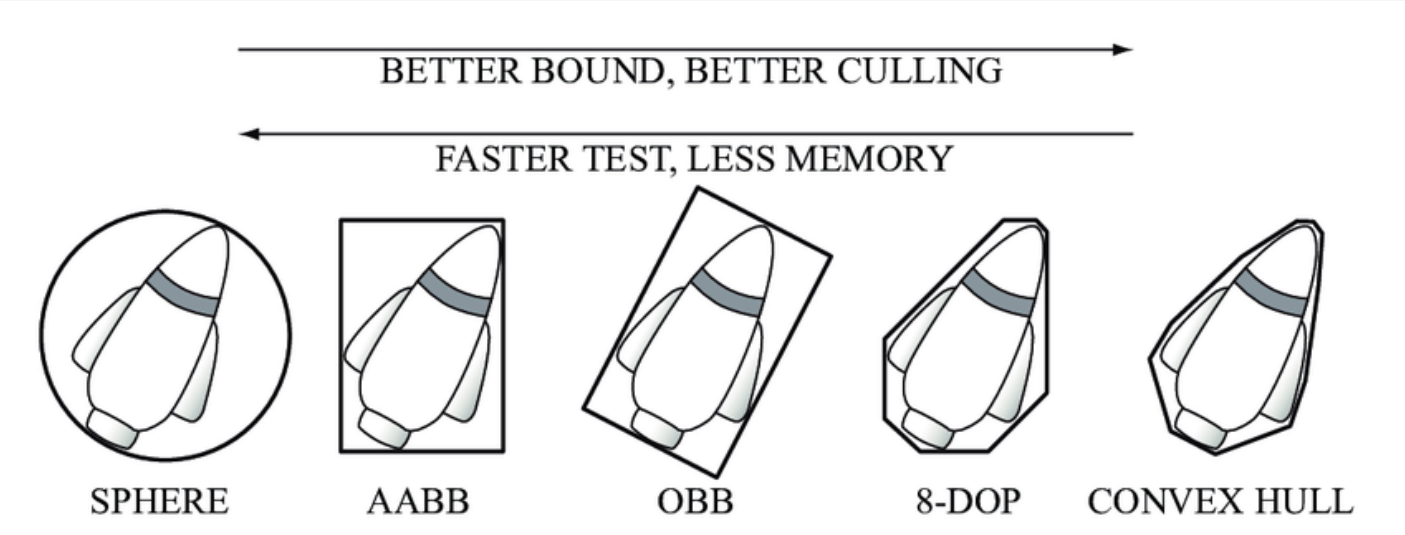
By carving up our object into **bounding volumes** which the part of the object is contained within, we can **compute intersections with the bounding volume** instead.

If miss: reject.
If intersect: do actual intersection computation with surface.

Question 6

Give two criteria for choosing a bounding volume shape for ray tracing acceleration. What 3D shape(s) fulfill these criteria?

Some types


BETTER BOUND, BETTER CULLING
FASTER TEST, LESS MEMORY
SPHERE
AABB
OBB
8-DOP
CONVEX HULL

Criteria

1. Must enclose each object as tightly as possible
2. Efficient to compute intersection with ray.

When is AABB bad?

e.g. compare AABB with OBB.

> May not be a tight bounding volume for object shapes that are elongated and oriented diagonally.

Attendance taking

Thanks! Get the slides here after the tutorial.

https://trxe.github.io/cs3241-notes