CS482: Ray Tracing

Sung-Eui Yoon (윤성의)

Course URL: http://sglab.kaist.ac.kr/~sungeui/ICG/

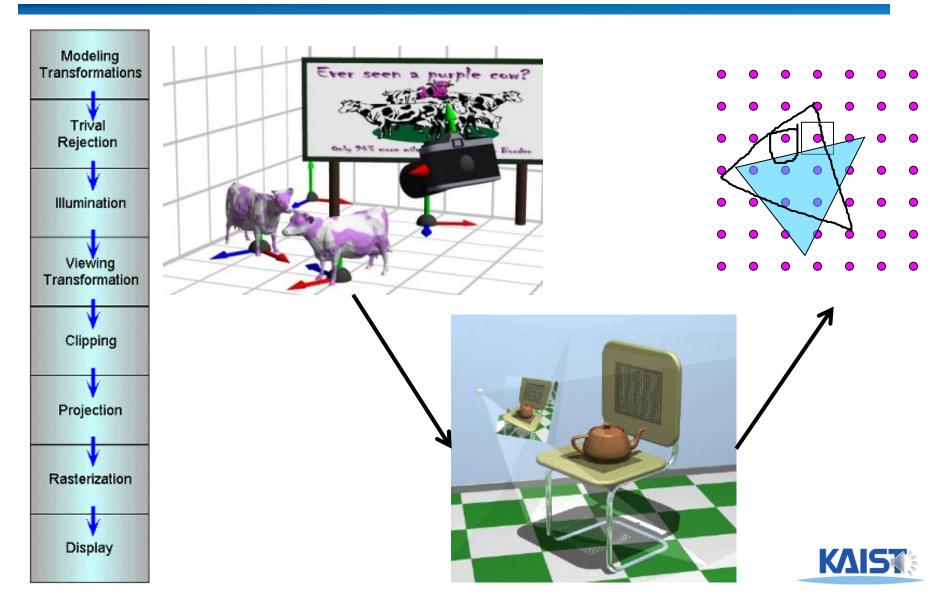


Class Objectives (Ch. 10)

- Understand a basic ray tracing
- Know its acceleration data structure and how to use it
- Rendering book https://sgvr.kaist.ac.kr/~sungeui/render/



The Classic Rendering Pipeline



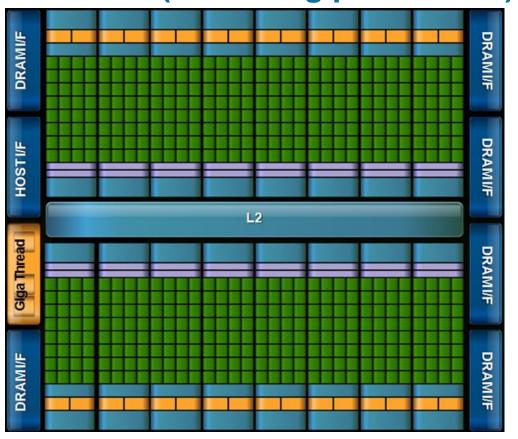
Why we are using rasterization?

- Efficiency
- Reasonably quality



Fermi GPU Architecture

16 SM (streaming processors)



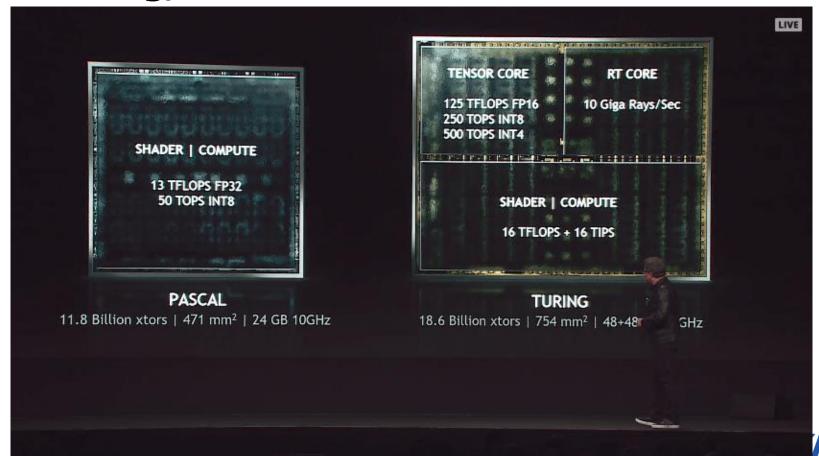
512 CUDA cores

Memory interfaces



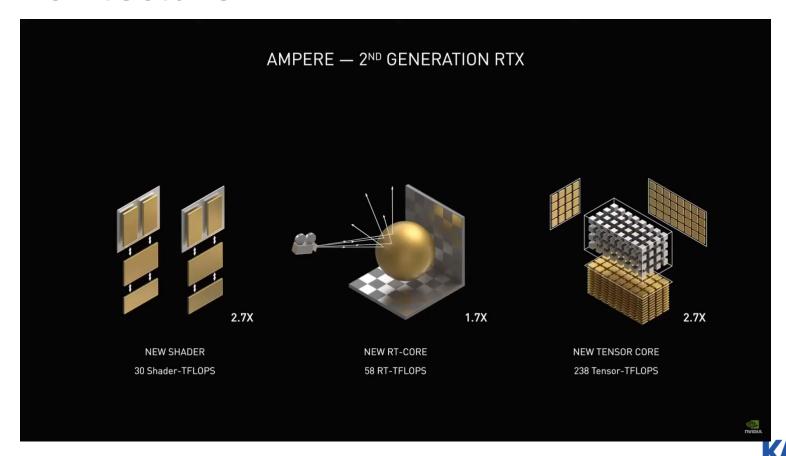
Turing Architecture, 2018

 Aims to combine shade, compute, ray tracing, and AI



Ampere Architecture, 2020

 More cores, faster computation than Turing Architecture



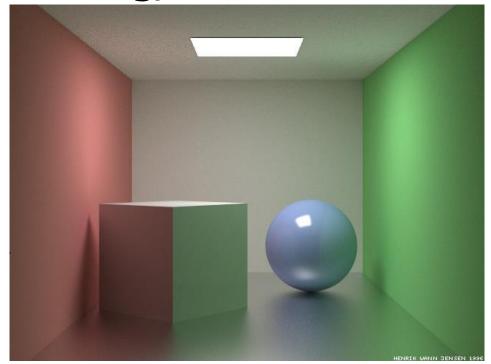
Where Rasterization Is

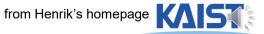


From Battlefield: Bad Company, EA Digital Illusions CE AB

But what about other visual cues?

- Lighting
 - Shadows
 - Shading: glossy, transparency
- Color bleeding, etc

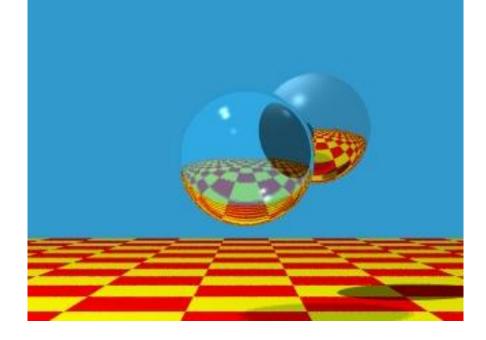




Recursive Ray Casting

 Gained popularity in when Turner Whitted (1980) recognized that recursive ray casting could be used for global illumination

effects





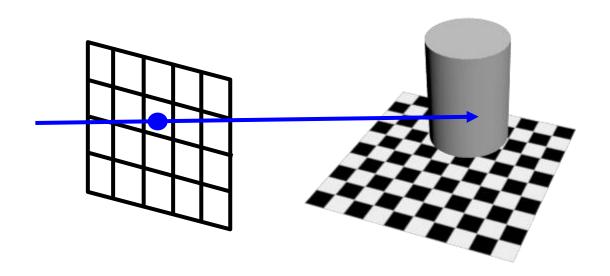
Ray Casting and Ray Tracing

- Trace rays from eye into scene
 - Backward ray tracing
- Ray casting used to compute visibility at the eye
- Perform ray tracing for arbitrary rays needed for shading
 - Reflections
 - Refraction and transparency
 - Shadows



Basic Algorithms

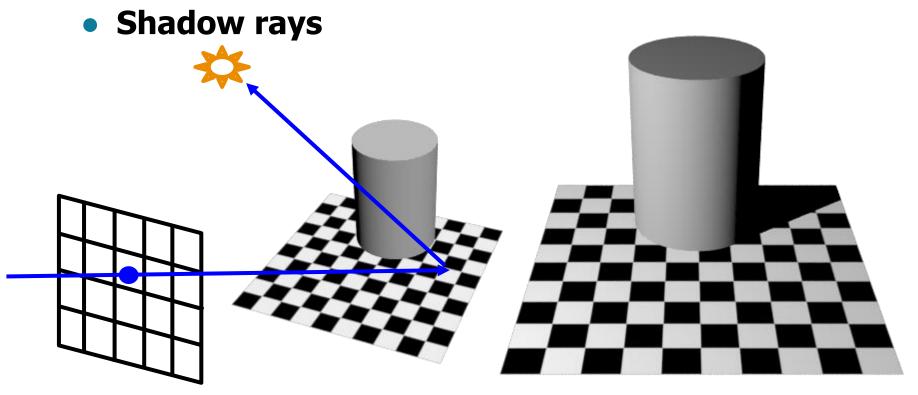
 Rays are cast from the eye point through each pixel in the image





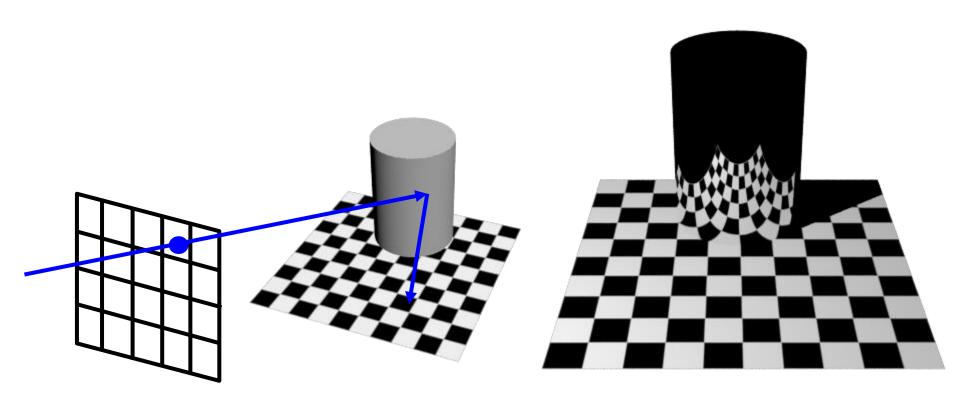
Shadows

 Cast ray from the intersection point to each light source



Reflections

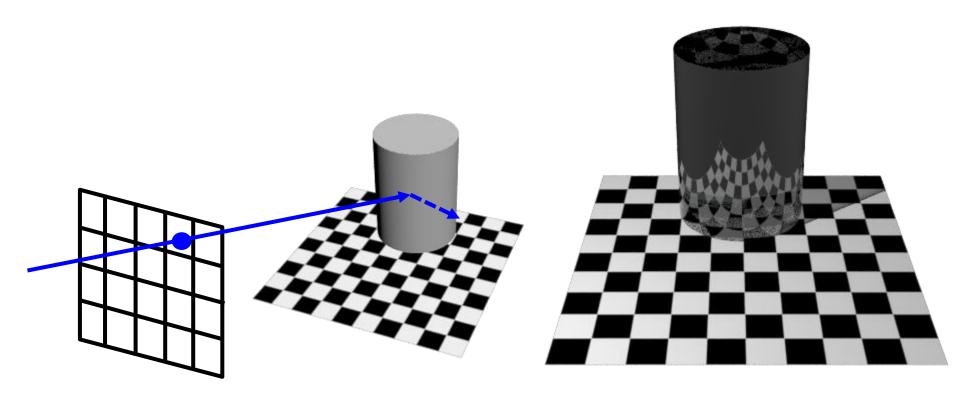
 If object specular, cast secondary reflected rays





Refractions

 If object transparent, cast secondary refracted rays





An Improved Illumination Model [Whitted 80]

Phong illumination model

$$I_r = \sum_{i=1}^{\text{numLights}} (k_a^j l_a^j + k_d^j l_d^j (\hat{N} \bullet \hat{L}_j) + k_s^j l_s^j (\hat{V} \bullet \hat{R})^{n_s})$$

Whitted model

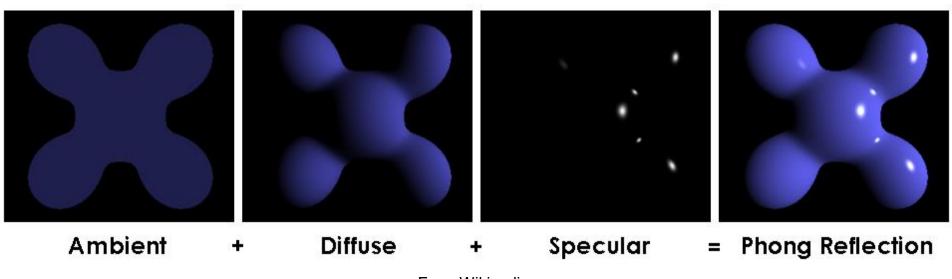
$$I_r = \sum_{i=1}^{\text{numLights}} (k_a^j l_a^j + k_d^j l_d^j (\hat{N} \bullet \hat{L}_j)) + k_s S + k_t T$$

- S and T are intensity of light from reflection and transmission rays
- Ks and Kt are specular and transmission coefficient



OpenGL's Illumination Model

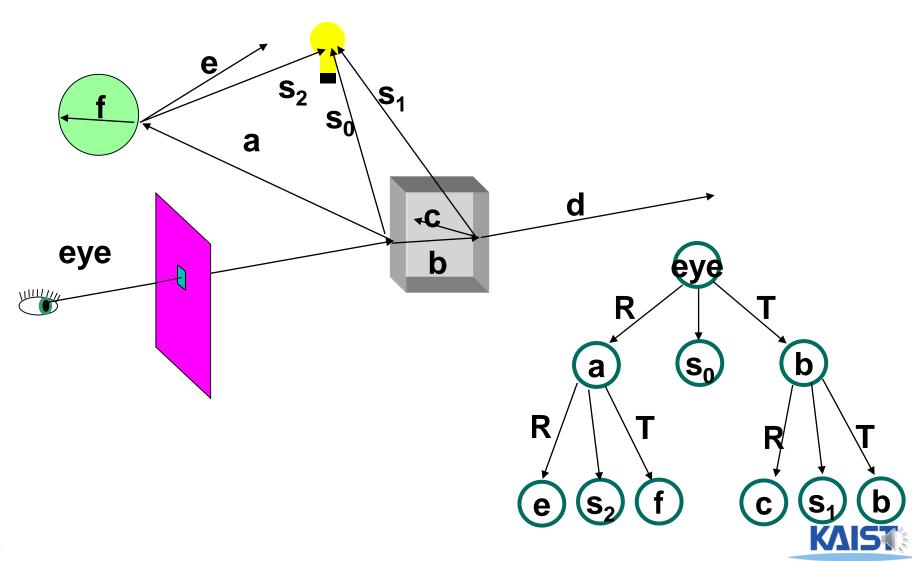
$$I_r = \sum_{j=1}^{\text{numLights}} (k_a^j l_a^j + k_d^j l_d^j \max((\hat{N} \cdot \hat{L}_j), 0) + k_s^j l_s^j \max((\hat{V} \cdot \hat{R})^n), 0)$$



From Wikipedia



Ray Tree



Acceleration Methods for Ray Tracing

- Rendering time for a ray tracer depends on the number of ray intersection tests per pixel
 - The number of pixels X the number of primitives in the scene

_ 🗆 X

- Early efforts focused on accelerating the rayobject intersection tests
 - Ray-triangle intersection tests
- More advanced methods required to make ray tracing practical
 - Bounding volume hierarchies
 - Spatial subdivision (e.g., kd-trees)

Bounding Volumes

- Enclose complex objects within a simple-tointersect objects
 - If the ray does not intersect the simple object then its contents can be ignored
 - The likelihood that it will strike the object depends on how tightly the volume surrounds the object.
- Spheres are simple, but not tight
- Axis-aligned bounding boxes often better
 - Can use nested or hierarchical bounding volumes









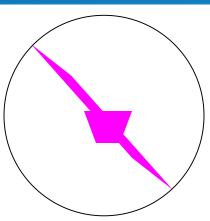
Bounding Volumes

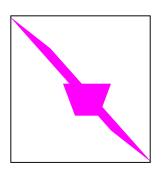
Sphere [Whitted80]

- Cheap to compute
- Cheap test
- Potentially very bad fit

Axis-Aligned Bounding Box

- Very cheap to compute
- Cheap test
- Tighter than sphere

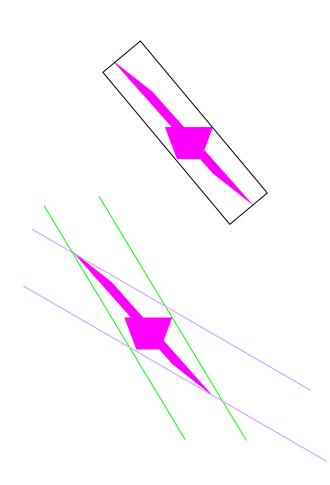






Bounding Volumes

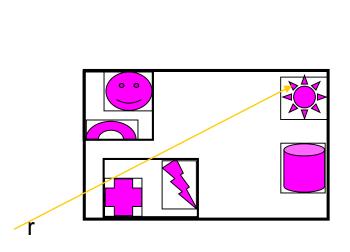
- Oriented Bounding Box
 - Fairly cheap to compute
 - Fairly Cheap test
 - Generally fairly tight
- Slabs / K-dops
 - More expensive to compute
 - Fairly cheap test
 - Can be tighter than OBB

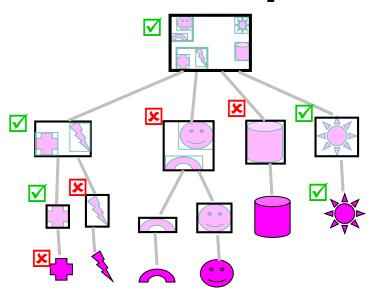




Bounding Volume Hierarchy (BVH)

- Organize bounding volumes as a tree
 - Choose a partitioning plane and distribute triangles into left and right nodes
- Each ray starts with the scene BV and traverses down through the hierarchy







Test-Of-Time 2006 Award

High-Performance Graphics 2015

Los Angeles, August 7-9, 2015

Home

Full Program

CFP

Registration

Accommodations

Venue

Submissions

Organization



RT-DEFORM: Interactive Ray Tracing of Dynamic Scenes using BVHs

Christian Lauterbach, Sung-eui Yoon, David Tuft, Dinesh Manocha

IEEE Interactive Ray Tracing, 2006



We present an efficient approach for interactive up making that formalls or strained marks. Unlike many of the most upformally or strained and the marks of the most uplacement of the marks of the marks of the marks of the homestical (NVIII) instead of laborate in the interactions about the marks of spikes or incompanies the UNIV. We also describe a marked to spike or incompanies the UNIV. We also describe a marked to spike or incompanies the UNIV. We also describe a marked to spike or incompanies the UNIV. We also districtly as the laboration of the Indiana, we show that the copy of the UNIV. The spike laborate which has been to allow one of the UNIV. The spike laborate which has been to make applied it different consists using it is necessive at these times and consists of time of themselves to stems to suggest that

cond on a desknop PC including secondary rays.

R Categories: L3.7 [Computer Graphics]: Throughties and Bealton—Raymacing;

1 INTRODUCTION

Bay tracing is a classic problem in computer graphics and has be maded in the literature for more than three decades. Most of "o-mail chifts.am.ads 'o-mail tonguait filal.gov accidents on structure, explosing toy colmerce, and pushed sign Most counts interactively travelage inflaments to before an a conference of the structure (T.P. 1) It is practice, before on explosing conference of the structure (T.P. 1) It is practice, before on explosing the conference of the structure (T.P. 1) It is practice, before on the final conference of the structure (T.P. 1) It is practice, before on the first man deadourtages of before to the light connections the first man deadourtages of before to the light connections the structure of the structure (T.P. 1). All the light travelages of the structure of the structure (T.P. 1) It is a structure of the structure of structure

Motor results: In this paper, we present a simple and officions of appeals for intensively conjugated to intensive conjugate on a public flower to exclusive the conjugate or update flower to intensive conjugate or applied flower to intensive conjugate or applied flower to intensive conjugated flower to





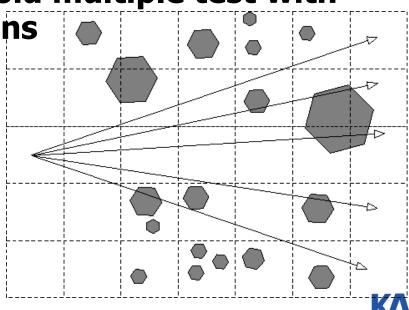
Spatial Subdivision

Idea: Divide space in to subregions

- Place objects within a subregion into a list
- Only traverse the lists of subregions that the ray passes through

 "Mailboxing" used to avoid multiple test with objects in multiple regions

- Many types
 - Regular grid
 - Octree
 - BSP tree
 - kd-tree



Classic Ray Tracing

- Gathering approach
 - From lights, reflected, and refracted directions
- Pros of ray tracing
 - Simple and improved realism over the rendering pipeline



Cons:

- Simple light model, material, and light propagation
- Not a complete solution
- Hard to accelerate with special-purpose H/W



History

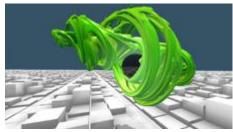
- Problems with classic ray tracing
 - Not realistic
 - View-dependent
- Radiosity (1984)
 - Global illumination in diffuse scenes
- Monte Carlo ray tracing (1986)
 - Global illumination for any environment



Interactive Ray Tracing Kernels

- OptiX, Nvidia
 - Utilize GPU computing architectures and CUDA







- Embree, Intel
 - Utilize CPUs (multi-threaded and SIMD)









PA₁

- Get to know OptiX or Embree
 - Download, and compile either one of those two methods
 - Or just use precompiled ones
 - Try out a few scenes
 - Upload images of those scenes in KLMS
- Deadline
 - Check the KLMS
- Note
 - Easy one, but start early







Homework

- Go over the next lecture slides before the class
- Watch 2 paper (or videos) and submit your summaries every Mon. class
 - Just one paragraph for each summary

Example:

Title: XXX XXXX XXXX

Abstract: this video is about accelerating the performance of ray tracing. To achieve its goal, they design a new technique for reordering rays, since by doing so, they can improve the ray coherence and thus improve the overall performance.

Class Objectives were:

- Understand a basic ray tracing
- Know its acceleration data structure and how to use it



Next Time

Radiosity

