

ECE 595 / CS 491 / CS 591
**Real-Time Rendering &
Graphics Hardware**

Pradeep Sen
Advanced Graphics Lab

Class 24
April 23, 2007

Announcements

- Hope that you have started the final project!
- Won't be in for office hours tomorrow
- Next week, Joe Kniss was supposed to give lectures but he has a family emergency. He won't be here so class is cancelled (work on your project)
- GFX Café this week: Ken Perlin

Last time

- GPGPU

Today

- Real time ray tracing
- Course evaluations

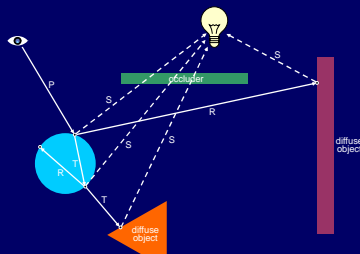
What is ray tracing?

- Rendering algorithm developed by Turner Whitted
- T. Whitted, "An Improved Illumination Model for Shaded Display." Communications of the ACM 1980

Basic idea

- Send "rays" out from the eye (a ray is specified by a origin and a direction)
- One or more rays are sent out per pixel
- Intersect the ray with the scene geometry
- If ray intersects, then compute shaded value at that surface (send shadow ray to test shadow)
- If surface is reflective, compute reflection ray and repeat
- If surface is refractive, compute refractive ray and repeat

Basic idea



Ray tracing algorithm

```
Color castRay(Ray r) {  
    if (intersectRayWithGeometry(r, &i) { // i is the first intersection  
        shadeSurface(r, i);  
  
        if (r->depth < MAX_DEPTH) {  
  
            // send reflection ray  
            if (i->primitive->reflective) {  
                r_refl = computeReflectionRay(r, i);  
                refl_color = castRay(r_refl);  
            }  
  
            // send refraction ray  
            if (i->primitive->refractive) {  
                r_refr = computeRefractionRay(r, i);  
                refr_color = castRay(r_refr);  
            }  
  
            return (surf_color + K_refl * refl_color + K_refra * refr_color);  
        }  
        return Color(0,0,0);  
    }  
}
```

Ray tracing algorithm

- Innately recursive (need a termination condition)
- "Embarassingly parallel"

Sample images



source: Whitted (1980)



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Newer images



image courtesy Henrik Wann Jensen



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Newer images



image courtesy Henrik Wann Jensen



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Newer images



image courtesy Henrik Wann Jensen



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Ray tracing in feature film

- Most high-end CG uses scanline renderers (ray tracing is only used to fill in certain effects)
- However, some films have been made using primarily ray tracing



Ice Age 2: The Meltdown - Blue Sky Studios (2006)



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Advantages

- Reflection and refraction are trivial (how do you do them in scanline rendering?)
- You can do things like soft shadows, depth of field, glossy surfaces by casting many shadow rays



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Observations

- As is, the algorithm is $O(n)$ where n is the number of triangles in the scene
- Can do better with acceleration structures $O(\log n)$



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Ray tracing acceleration techniques

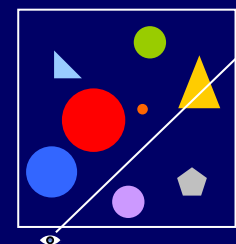
- Cast fewer rays
 - Adaptive ray tracing
 - Early ray termination
- More efficient intersection tests
 - Uniform grids
 - Spatial hierarchies (bsp, octree, k-d trees)
 - Hierarchical bounding volumes



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

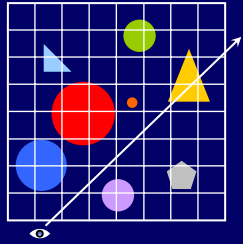
Uniform grids



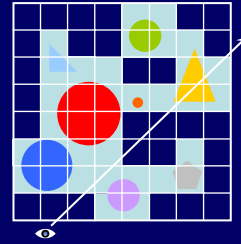
Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

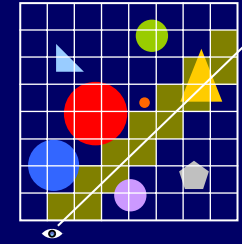
Uniform grids



Uniform grids



Uniform grids



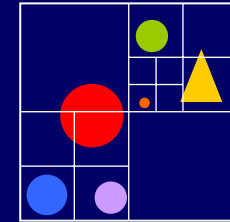
Worst-case for uniform grid

- Teapot in the stadium

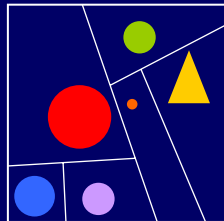
Spatial hierarchies

- Oct tree
- BSP tree
- k-d tree

Oct Tree



Binary space partition



Acceleration structures

- Acceleration structures are necessary if we want fast ray tracing
- If we want a real-time ray tracing implementation, we will need to be able to handle the acceleration structures

Approaches for real-time ray tracing

- Use fast CPU or cluster CPU implementations (Susallek et al.)
- Implement ray tracing on the commodity graphics hardware (Purcell et al.)
- Build custom hardware for ray tracing (Woop et al.)

Fast CPU/cluster implementations

- Examples include Saarland University's OpenRT implementation (Wald et al.), Utah's Star-Ray architecture, etc



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Saarland ray tracer

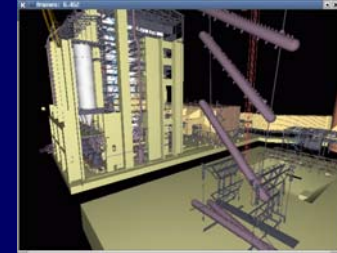
- Utilizes coherence by bundling rays and tracing, intersecting, and shading them in SIMD fashion
- 50 million triangles/sec
- Frame rates about 4fps (without reflections), 1 fps (with reflections)



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Examples



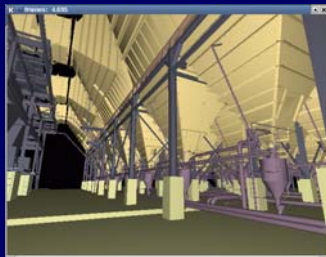
source: Wald et al. (2001)



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Examples



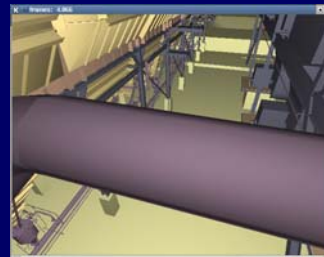
source: Wald et al. (2001)



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Examples



source: Wald et al. (2001)



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Ray tracing on the GPU

- The GPU is intended to do scanline rendering
- Has the new programmability allow it to do things it wasn't meant to do like ray tracing?
- Yes!



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Publication

- Purcell et al. "Ray Tracing on Programmable Graphics Hardware" SIGGRAPH 2002



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Basic idea

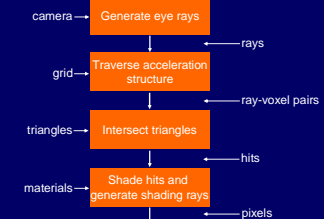
- Ray tracing can be expressed in the stream programming model
- Since the GPU can be thought of as a stream processor it is possible to map ray tracing to that architecture



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Streaming ray tracing



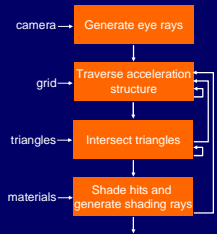
slide courtesy Tim Purcell



Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 24 - April 23, 2007

Streaming ray tracing

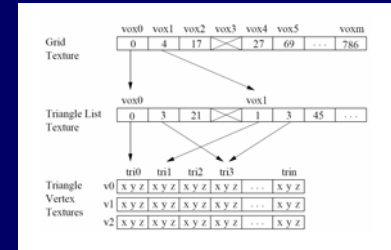


slide courtesy Tim Purcell

Representing data on the GPU

- The grid and the triangles are represented on the GPU using textures

Representing data on the GPU

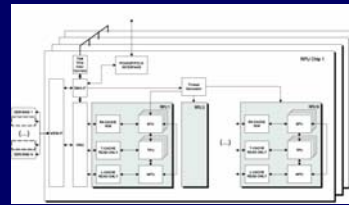


source: Purcell et al. 2002

Custom hardware for ray tracing

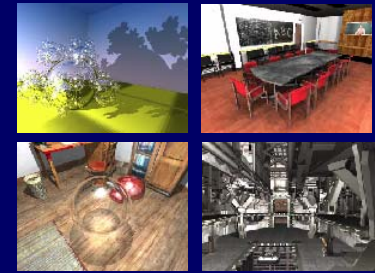
- Woop et al. "RPU: A Programmable Ray Processing Unit for Realtime Ray Tracing" SIGGRAPH 2005

RPU architecture



source: Woop et al. 2005

Results



source: Woop et al. 2005

Upcoming conferences

- EG Symposium on Interactive Ray Tracing, Ulm Germany
- Submission deadline: June 14

Reading

- Wald et al., "STAR: Real-time ray tracing and its use in interactive global illumination," Eurographics 2003