Real-Time Graphics Architecture

Kurt Akeley

Pat Hanrahan

http://www.graphics.stanford.edu/courses/cs448a-01-fall

Ray Tracing

with Tim Purcell

Topics

Why ray tracing?

Interactive ray tracing on multicomputers

Ray tracing hardware

The cost of ray tracing

SHARP architecture

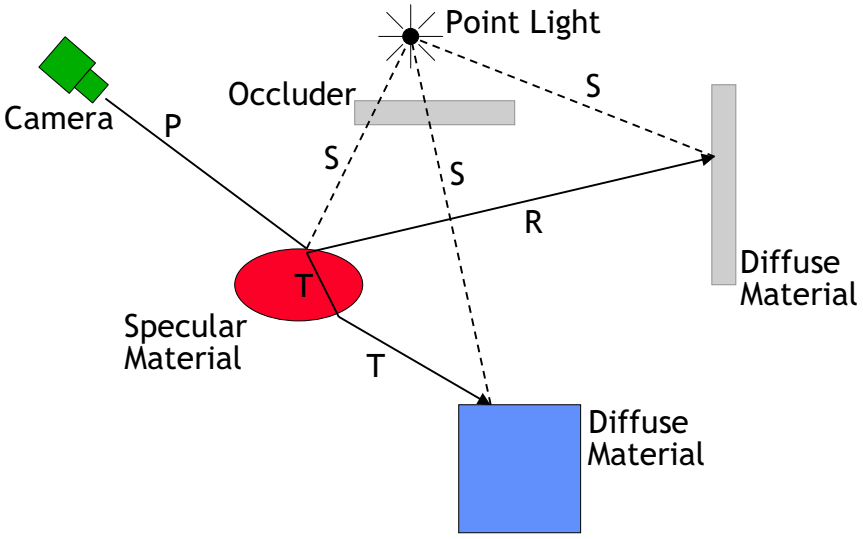
Trends

Readings

Required

1. I. Wald, P. Slusallek, State-of-the-art in interactive ray tracing, Eurographics 2001

Ray Tracing Algorithm



Photorealism

Direct light simulation vs. approximations

- True shadows vs. shadow maps
- True reflections vs. environment maps
- Interreflections

Henrik Wann Jensen

Marcos Fajardo

Target Implementation Spectrum

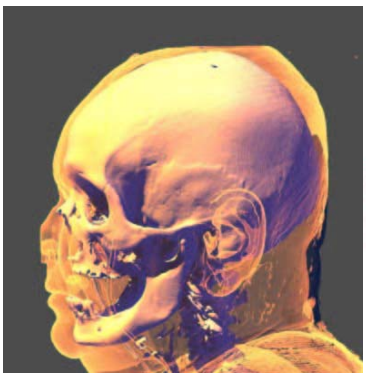
Specialization

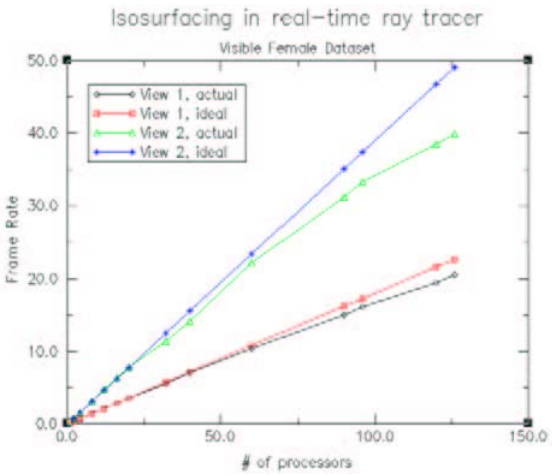
Custom hardware	ART AR350
Programmable graphics processor	NVIDIA/ATI
Stream processor	Imagine
MP on a single chip	Smart Memories
General Purpose CPU with specialized instructions	PC w/ SSE [Saarbrücken RTRT]
General purpose CPU	R10K Utah IRT

Interactive Ray Tracing

Embarrassingly parallel

Global database





Utah IRT on SGI Origin

Interactive Ray Tracing

Optimized data structures for cache

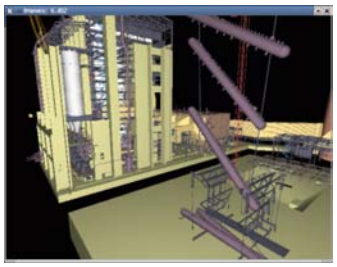
- Cluster data structures by access
- Pad and block (32B)
- Prefetch triangle data

Optimized code for IA w/ SSE

- Computation

  Min 78 22 3.5

  Max 148 41 3.7



Univ of Saarbrucken IRT

- Traces 4 rays at a time through BSP tree

Range of speeds 200 KR/s to 1.5 MR/s (800 Mhz PIII)

Advantages: Output Complexity

Lazy evaluation – need not touch every triangle

Logarithmic search for ray intersection

Scene	Tris	Oct.	Onyx	PC	RTRT
MGF office	40k	>24	> 36	12.7	1.8
MGF conf.	256k	>5	> 10	5.4	1.6
MGF theater	680k	0.4	6-12	1.5	1.1
Library	907k	1.5	4	1.6	1.1
Soda Floor	2.5m	0.5	1.5	0.6	1.5
Soda Hall	8m	OOM	OOM	OOM	0.8

Table 3: *OpenGL rendering performance in frames per second with SGI Performer on three different graphics hardware platforms compared with our software ray tracer at a resolution of* 512^2 *pixels on a dual processor PC. The ray tracer uses only a single processor, while SGI Performer actually uses all available CPUs.*

Advantages: Selective Sampling

- Adaptive sampling
- Frameless rendering (just-in-time rendering)

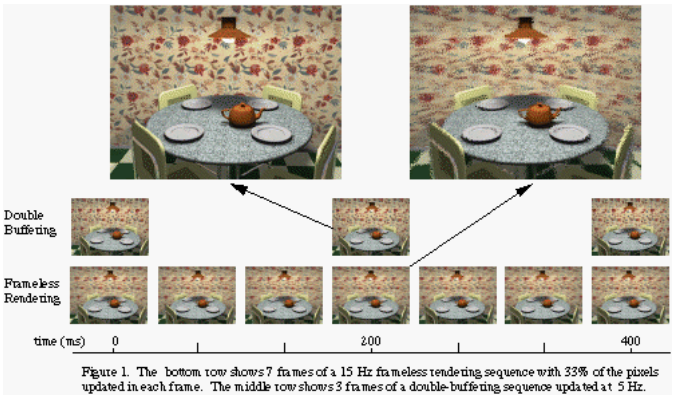
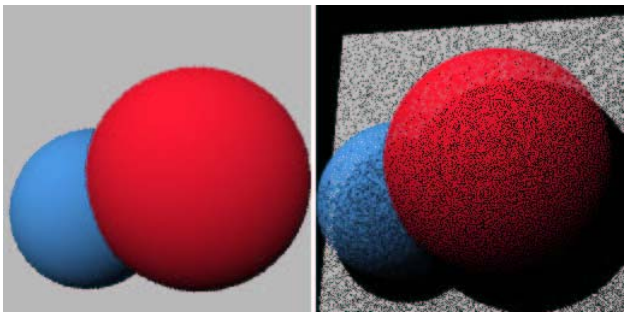


Figure 1. The bottom row shows 7 frames of a 15 Hz frameless rendering sequence with 33% of the pixels updated in each frame. The middle row shows 3 frames of a double-buffering sequence updated at 5 Hz.

From Bishop et al., Frameless Rendering, SIGGRAPH 94

Advantages: Selective Sampling

- Adaptive sampling
- Frameless rendering (just-in-time rendering)
- RenderCache
  1. Reproject
  2. Ray trace in holes



From Walter et al., RenderCache, Eurographics 97

Advantages: Selective Sampling

- Adaptive sampling
- Frameless rendering (just-in-time rendering)
- RenderCache
- Holodeck
- Foveal rendering
- Rendering under pressure

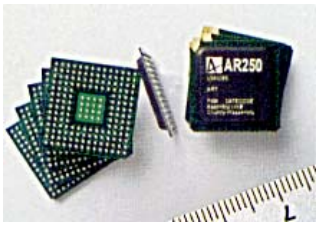
AR250 (newer AR350)

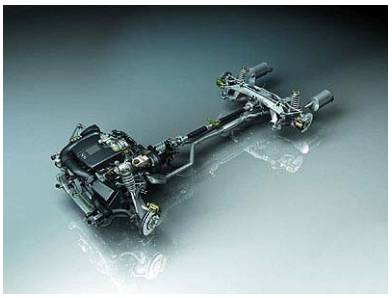
.35 um, 106 mm^2 die

650K gates

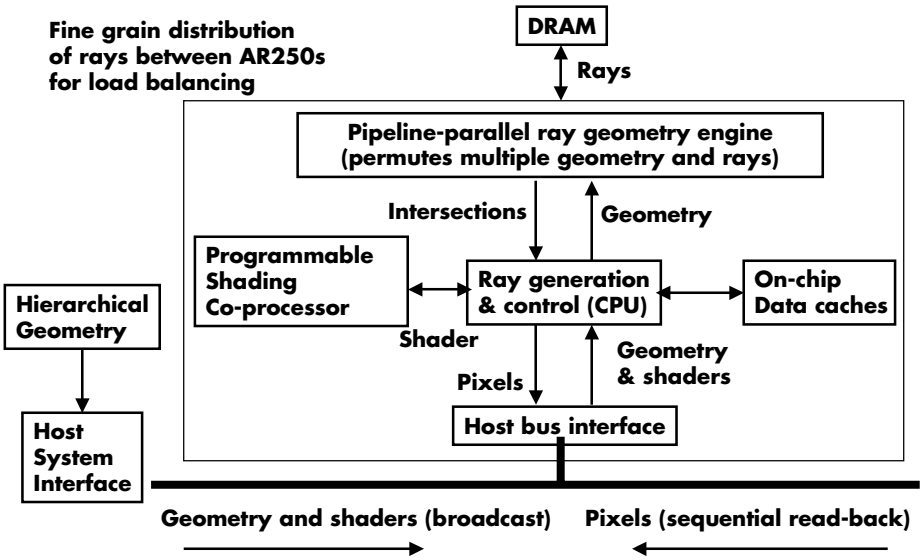
32 single stage IEEE FPUs

50 Mhz

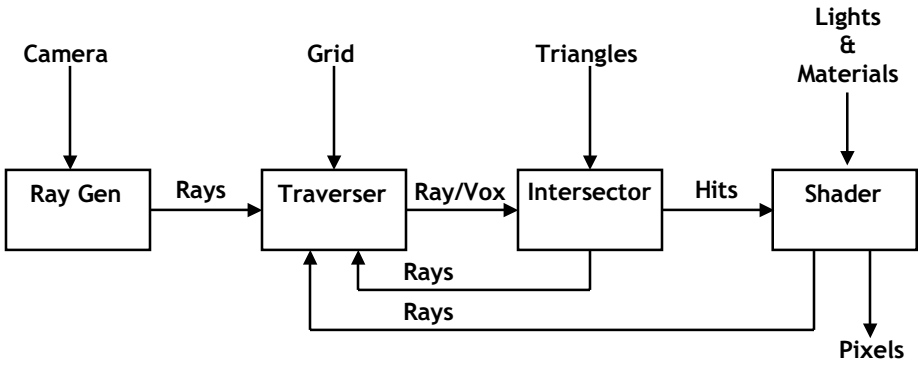




ART AR250



SHARP Architecture



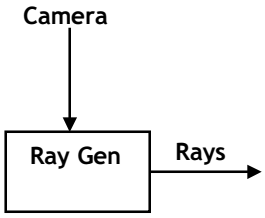
Recirculating streams

Eye Ray Generator

Computation

- 10+, 13*, 5/, 3 SPC
- 100 MIPS R10K
- 13 VP1.0

```
Ray {
    vec3f O;
    vec3f D;
    float t;
    vec2f xy;
    vec3f weight;
    byte  type;
}  // 49 bytes
```



Traverser

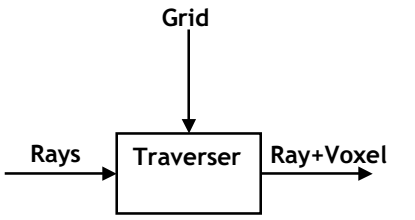
Acceleration data struction

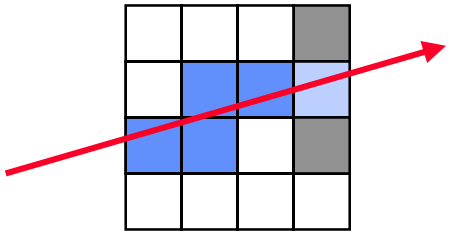
- Occupancy bitmap
  - No triangle data
- Ray-voxel grid traversal
  - 3D-DDA (line drawing)

Computation

Setup 15+, 6*, 9/, 30 CMP

Traverse 4+, 3*, 8 CMP





Grid Acceleration Structure

Reasonable: No clear 'best' accel. structure anyway

Optimizations

- Mailboxing (Doesn't parallelize well, don't use)
- Blocking (64bits = 4x4x4)
- Hierarchies (Two-level blocked grid)

Inserting triangles into grid

- Amortize cost for static geometry (display lists)
- Costly for time-varying geometry
- Basically 3D rasterization

Intersector

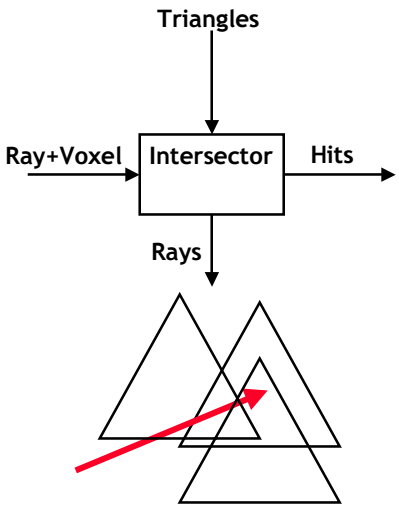
Ray-Triangle Intersection

- Möller-Trumbore algorithm
- Basically point inside triangle
- Rasterization!

Computation

23+, 27*, 1/, 14 CMP

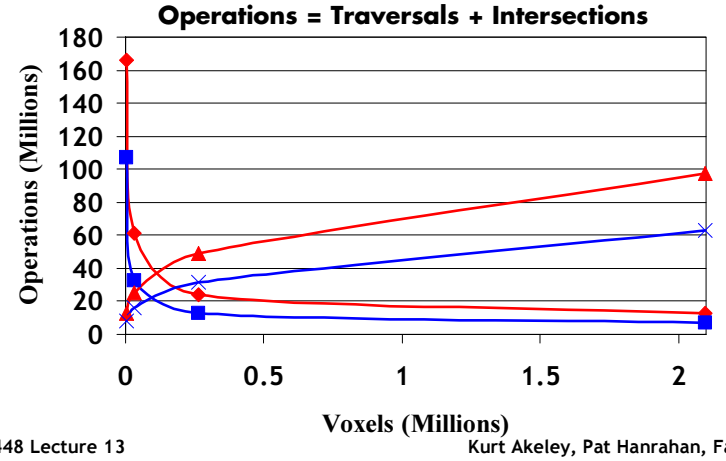
```
Triangle {
    vec3f P[3];
}  // 36 bytes
```



Total Operations

Bunny
69,415 Triangles

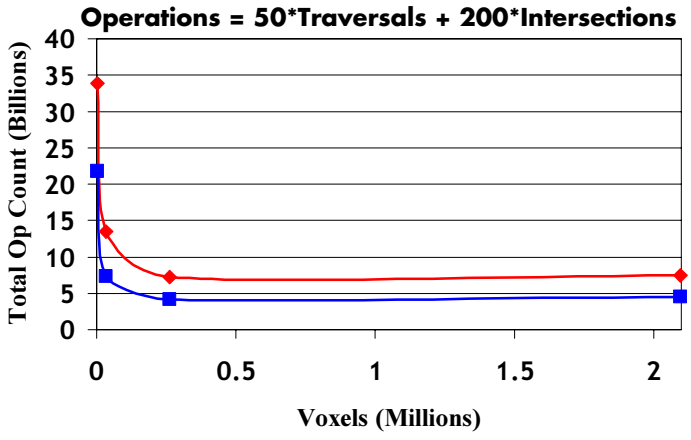
Quake 3
35,468 Triangles



Total Instructions

Bunny
69,415 Triangles

Quake 3
35,468 Triangles



Shader

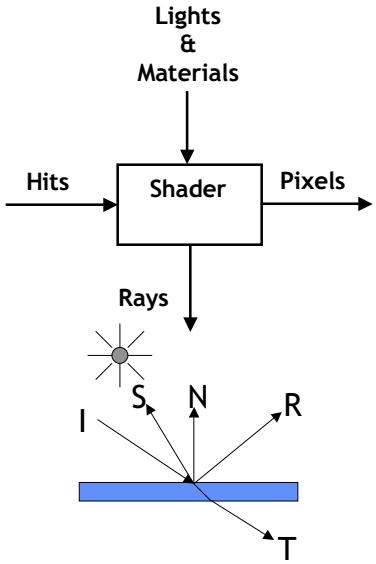
Diffuse shading

19+, 21*

```
Hit {
    vec3f O, D;
    vec2f xy;
    vec3f weight;
    float t;
    byte  type;
    long  tri;
    vec2f uv;
}  // 99 bytes
```

Shadow and secondary rays generated here

Weighted rays



Cost Model

C = R T + S

With acceleration

C = RV + RT + S

l = average number of ray-voxel traversals

mean free path of the ray

a = average number of triangles per voxel

t = l a (average number of ray-triangle intersections)

C = R (l Crv + t Crt + Cs) = R [l * (Crv + a Crt) + Cs]

l analogous to depth complexity

Cost Model

Ray-Triangle intersections = r T = t R

r = average number of rays tested per triangle

t = average number of triangles tested per ray

When r < 1, ray tracing wins

- Two effects
  - Depth complexity
  - Small triangles (might still cost something)

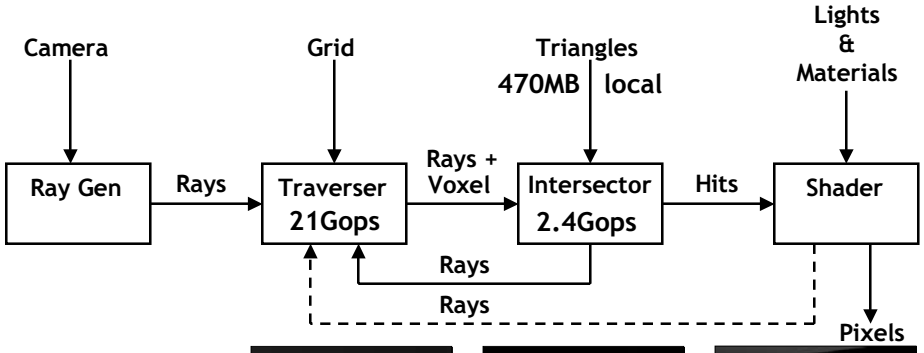
Cost Model

Deferred Shading

R l (Crt + Cs/l) = I d (Cfv + Cs)

When Cs large, ray tracing wins

Architecture Measurements



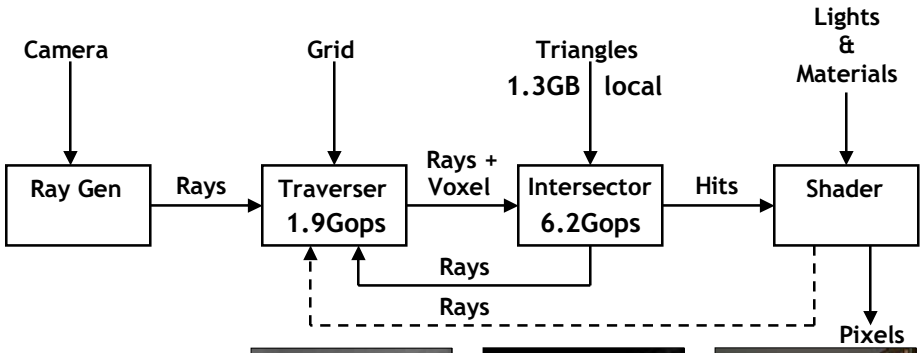
282K Triangles

Eye Rays Only

1 frame

1024x1024 pixels

Architecture Measurements



1.5M Triangles

Eye Rays Only

1 frame

1024x1024 pixels

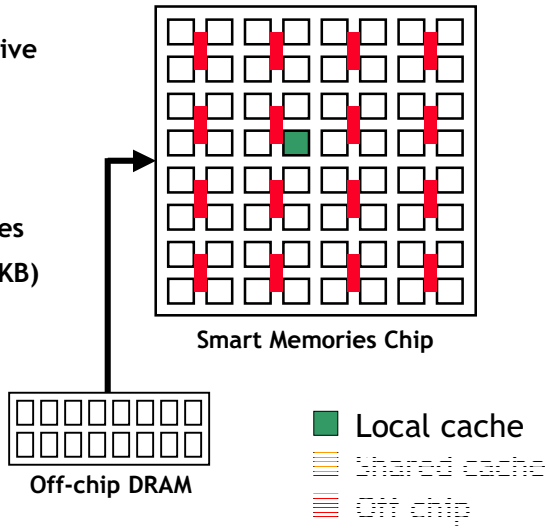
Caching on Smart Memories

Cache Policy

- 4-way set associative
- LRU replacement

System Configuration

- 32 intersection tiles
- 1024 triangle (44 KB) cache size each
- Shared caches



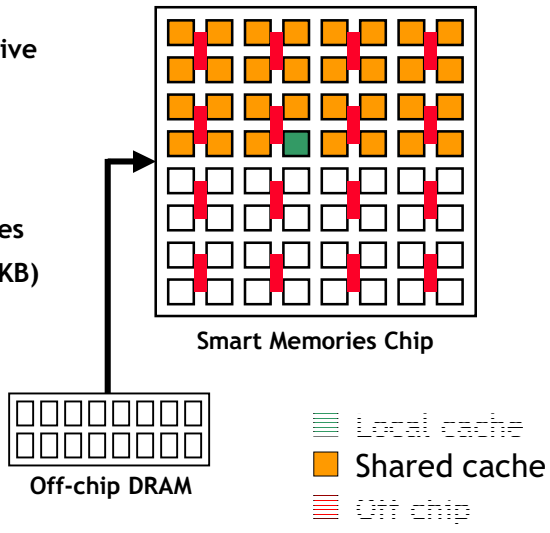
Caching on Smart Memories

Cache Policy

- 4-way set associative
- LRU replacement

System Configuration

- 32 intersection tiles
- 1024 triangle (44 KB) cache size each
- Shared caches



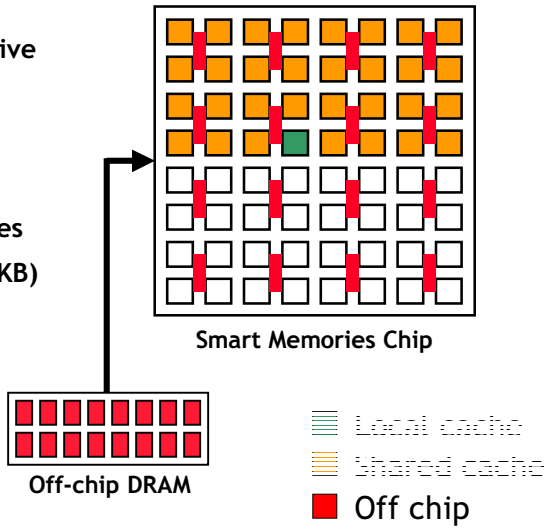
Caching on Smart Memories

Cache Policy

- 4-way set associative
- LRU replacement

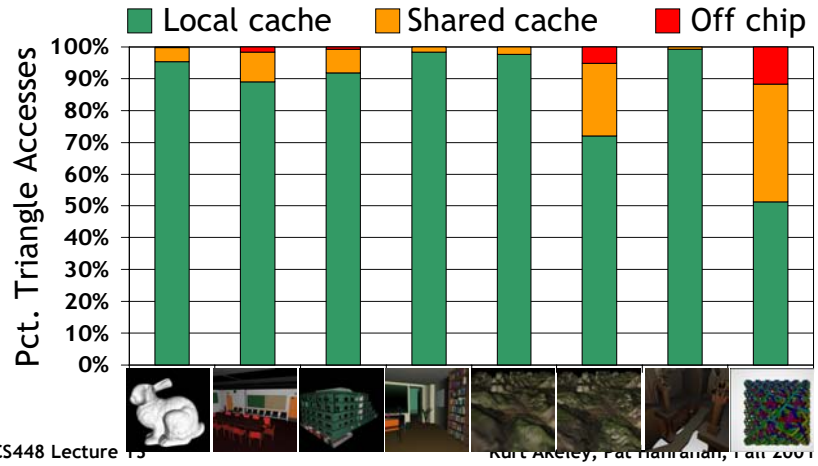
System Configuration

- 32 intersection tiles
- 1024 triangle (44 KB) cache size each
- Shared caches

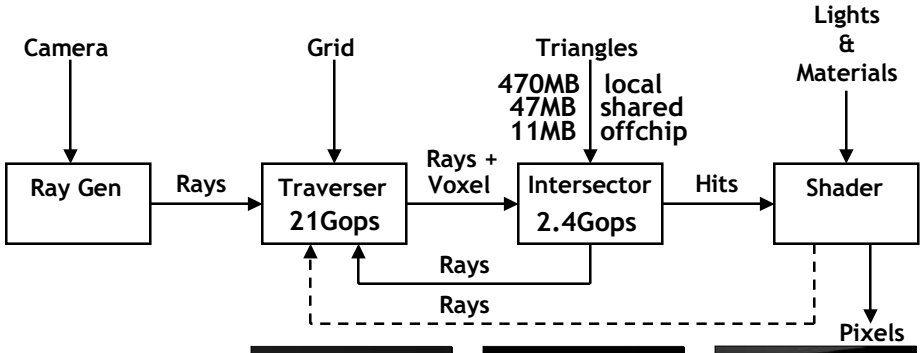


Small and Simple Caching Suffices

Over 93% of all triangle references are on chip -- stall time < 10% of execution time



Architecture Measurements



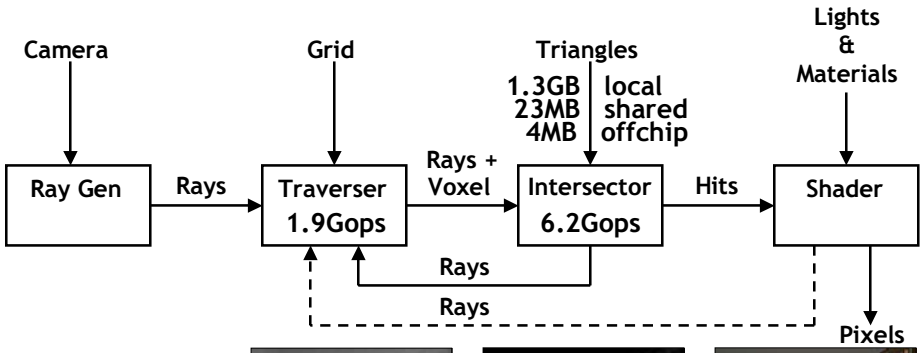
282K Triangles

Eye Rays Only

1 frame

1024x1024 pixels

Architecture Measurements



1.5M Triangles

Eye Rays Only

1 frame

1024x1024 pixels

Performance Analysis

Compute limited

- 3:1 compute to internal bandwidth ratio for intersector
- 10:1 compute to internal bandwidth ratio for traverser
- Ray tracing generally thought to be bandwidth limited

Tuning Smart Memories for ray tracing

- Split FP units (64bit → 2x 32bit) (2x)
- Specialized and/or SSE-like instructions (2-5x)
- Remote data pre-fetching (1.2x)

Excellent performance!

- 32 tiles can perform 1500M intersections/s
- 30 tiles can perform 5000M traversals/s
- 100 times faster than PC with SSE

Summary and Trends

Summary

Ray tracing algorithm

- Embarrassingly parallel
- Global access to database

Ray tracing kernels not unlike pipeline kernels

- Ray-voxel traversal like line drawing
- Ray-triangle intersection like rasterization

Seems to map well to streaming processor

Evolution of the Graphics Pipeline

Evolving the graphics pipeline to accommodate RT

Hybrid algorithms (1st pass using conventional pipe)

- Shadows
- Ray-traced reflections

Full ray tracing

- Ray casting
- Monte Carlo path tracing

Cross-Over

When does ray tracing compete with classic pipeline?

- High model complexity
  - High depth complexity
  - Small triangle size
  - Efficient acceleration structure
- High shading complexity
  - Deferred shading wins

Exactly when?

Convergence of Geometry/Texture

Different mechanisms

- Define and bind textures
- Render immediate mode geometry

Complexity

- Detail in texture
- Detail in geometry

Static vs. dynamic

- Textures static
- Geometry dynamic