GPU Ray Tracing & NVIDIA Advanced Rendering Solutions
Phillip Miller & Lutz Kettner
Agenda

- GPU Ray Tracing Overview
- OptiX
- Iray
- MDL
- Iray+
GPU Ray Tracing - what’s the big deal?

Single GPU speedups in commercial ray tracing app’s vs CPU

- Quadcore CPU
- Iray
- Ansys Fluent
- VRay-RT
- Adobe After Effects
- mental ray AO
GPU Ray Tracing in Commercial Products

- Adobe After Effects: OptiX API
- Arion: CUDA C
- CentiLeo: CUDA C
- finalRender: CUDA C
- Furry Ball: OptiX API
- Iray Photoreal: CUDA C
- Iray Interactive: OptiX API
- mental ray 3.11 (in 3ds Max, Maya, SI): OptiX API
- Octane: CUDA C
- Optis: OpenCL
- RedShift: CUDA C
- V-Ray RT: CUDA C (and OpenCL for x86)
- Pixar (in-house, Lighting): OptiX API
- Weta (in-house, Panta Ray): CUDA C

Out of Core
Ray Tracing with NVIDIA

- **CUDA** - language and computing platform
  - The primary choice for building *entirely custom GPU solutions from scratch*

- **NVIDIA OptiX™** - toolkit for ray tracing developers
  - Good choice for developers with *domain expertise* building *custom ray tracing solutions* who prefer to *leave GPU details to NVIDIA*

- **Iray® & mental ray®** - commercial rendering products
  - Good choice for companies wanting *ready-to-integrate rendering* which is *maintained and advanced for them*
  - Iray focusing on Interactive Design; mental ray focusing on Film Production
NVIDIA Assisting Companies with Rendering

**Developer Type**
- Rendering Companies: Chaos Group, OTOY, Cebas, AAA Studios, Random Control, ...
- In-House Development: BAE, Canon, CCP, Dolby, Honda, Lockheed, Pixar, Sony, USAF, ...
- Companies not focused on Advanced Rendering: Adobe, ANSYS

**In-House Development**
- Autodesk, Dassault, PTC, Bunkspeed

**Libraries**
- CUDA
- OptiX
- mental ray
- Iray
- IndeX

**Licensed Components**
- Q&A, PAPERS
NVIDIA® OptiX™ ray tracing engine

A programmable ray tracing framework enabling the rapid development of high performance ray tracing applications - from complete renderers to discrete functions (collision, acoustics, ballistics, radiation reflectance, signals, etc.)

- Use your techniques, methods, and data for your application with simple programs and a single-ray programming model

- OptiX does the “heavy lifting” of ray tracing, with straightforward APIs, for traversal, intersection, acceleration structures, and (optionally) shading - allowing you to concentrate on technique

- OptiX handles GPU aspects for you, handling load balancing, parallelism, scaling, paging, and optimizing per GPU architecture
OptiX: Analogous to OpenGL

- C-based Shaders/Functions (minimal CUDA exp. reqd.)
- Small, Custom Programs
  - Acceleration Structures Build & Traversal
  - Optimal GPU parallelism and Performance
  - Memory Management
  - Paging

- Application Code & Data Structures
  - OptiX
  - OpenGL or Direct3D

- GPU
OptiX Across Markets and Disciplines

Designed for Generality:

- No assumptions on technique, shading language, geometry type, or data structure
- Supports custom ray generation, material shading, object intersection, scene traversal, ray payloads
- Programmable intersection for custom surface types (procedurals, patches, NURBS, displacement, hair, fur, etc.)
- As many as 1/3 of OptiX applications don’t “render”, and a very simple Traversal API makes this even easier for applications that only need to trace rays – 5 lines of code and you’re ray tracing on the GPU
Ray Tracing in Compositing

Adobe After Effects (starting with Creative Suite 6)
- Ray traced compositing feature
- 100% OptiX - no x86 code
- Includes CPU Fallback
  - Done automatically
  - Transparent to developer
  - Currently exclusive to Adobe
Ambient Occlusion Baking for Games

Original Rendering

Ambient Occlusion

Resulting Higher Quality Game Experience

Courtesy of Wolfire Games
Light Baking for Game Development

Courtesy of Wolfire Games

Geomeric
Sound Propagation for Acoustics
NOTE: Growing CPU time of view-factor computations inhibit proper inclusion of radiation HT effects.

NOTE: GPU time remains low even as view-factor computations grow very large.
Holograph Rendering
OptiX within a Renderer for Maya

- **Furry Ball** for Maya from Art And Animation Studio
- A multi-pass D3D renderer until now
- Added ray tracing via OptiX in just 3 months
OptiX at Pixar Studios

- Pixar Studios showing an in-house tool for Katana built upon OptiX
- Achieves about 100 million rays per second on a single GPU
- Transforms the lighting process - making it interactive

"The Kepler features are key to our next generation of real-time lighting and geometry handling. The added memory and other features allow our artists to see much more of the final scene in a real-time, interactive form, which allows many more artistic iterations."

- Guido Quaroni, VP Software R&D, Pixar

The most powerful GPU for the world’s best artists
OptiX at Pixar Studios

OptiX at Pixar Studios added ray tracing via OptiX in just 3 months.
OptiX at Pixar Studios

Furry Ball for Maya

A multi-pass D3D renderer until now

Added ray tracing via OptiX in just 3 months
OptiX at Pixar Studios

A multi-pas D3D renderer until now. Added ray tracing via OptiX in just 3 months.
OptiX at Pixar Studios

Furry Ball for Maya

A multi-pass D3D renderer until now

Added ray tracing via OptiX in just 3 months
OptiX at Pixar Studios

- Furry Ball for Maya
- A multi-pas D3D renderer until now
- Added ray tracing via OptiX in just 3 months
OptiX at Pixar Studios
OptiX at Pixar Studios
OptiX in mental ray 3.11

- mental ray 3.11 ambient occlusion processing, now shipping in v2014 of Autodesk Maya, Softimage, and 3ds Max
- 1.5sec HLBVH build + 15sec on Quadro 6000 vs. 20 minutes on CPU
- Plan of record is to increase GPU acceleration within mental ray each year

\[
\begin{align*}
<20\text{m tri} &= 25-70X \text{ quadcore} \\
>20\text{m tri} &= 10-20X \text{ quadcore (paging)}
\end{align*}
\]
OptiX in Iray 2013

- Iray Photoreal
  - Interactive but “noisy”
  - The overall scene resolves in a couple of minutes

- Iray Interactive (on OptiX)
  - Interactive with minimal noise
  - The overall scene resolves in a couple of seconds
OptiX - Recent Advances

- Out of Core support (paging to system RAM)
- Optimized for CUDA 5 and NVVM (aka LLVM for CUDA)
- CUDA Interop – for sharing CUDA contexts and pointers
- Callable Programs – for shade trees, etc.
- Much faster Acceleration Structure building - up to 8X faster SBVH
- BVH refitting on all AS Builders
- GPU Direct for faster GL interop buffers
- Eliminated unnecessary compiles at run time
OptiX - Next Up

- Much faster Traversal API
  - As simple as ever to implement
  - A perfect pairing to wavefront approaches

- Faster building, high quality acceleration structures

- Details on this in David McAllister’s following talk

- OptiX “higher tier”
  - An elevated offering bringing a higher level of support, samples and capabilities than is available in the base OptiX version
SIGGRAPH 2013
Shaping the Future of Visual Computing

NVIDIA Iray
NVIDIA Commercial Rendering Offerings

- Solve meaningful rendering problems, while showing what’s possible via GPUs
- Completes a vital Feedback Loop to rapidly influence NVIDIA (long before products alone can)

Result: best of class solutions exploiting & influencing the full ecosystem
Iray - photorealism

Physically based rendering, resulting in a strong parallel to photography
Iray 2013 - in use

CATIA
For Software Developers wanting to add physically based rendering to their applications that is easy to use, highly interactive and scalable

Now with render modes, shared materials, cluster management, and cloud rendering

A rich API handshakes with Application manipulation for interactive updates
Near linear scaling for production rendering
Incremental updates rapidly propagate scene changes, giving full cluster processing after the scene change completes.
Includes a cluster configuration front end for dynamically allocating networked resources.
Scalable rendering power on demand – for WAN
Network protocols to handle private and public clouds (e.g., AWS)
Assets are only ever sent once – for minimal upload times
All edits are handled incrementally – for rapid updates
Scene updates get faster the more you use it
Iray on Grid Tech Demo at SIGGRAPH

- Connecting to an offsite cluster of NVIDIA Grid Appliances using multiple partner applications
- Scaling Interactive Performance as well as Offline
  - Native InfiniBand support
  - Performance limited only by cluster size
- Noiseless, interactive GI is within reach
- Slated to be a commercial option for Iray 2014
Debuting at SIGGRAPH: Iray® for Autodesk Maya®

- Coming from 0X1 Software later this year
- Fully integrated Iray 2013
  - Extremely Interactive
  - Extremely Scalable
  - Latest Material Possibilities
  - Designed to work hand in hand with mental ray for the widest range of production needs

Contact chris@0x1-software.com to apply for private Beta
Iray for Maya - sneak peek

- Fully integrated Iray 2013
  - Extremely Interactive
  - Extremely Scalable
  - Latest Material Possibilities

- Shipping later this year

Fully integrated Iray 2013
contact: chris@0x1-software.com
Multiple rendering modes, provide a quality/speed continuum

<table>
<thead>
<tr>
<th>Iray Realtime</th>
<th>Iray Interactive</th>
<th>Iray Photoreal</th>
</tr>
</thead>
<tbody>
<tr>
<td>120 FPS</td>
<td>15 FPS*</td>
<td>10 FPS</td>
</tr>
<tr>
<td>Stereo</td>
<td>20 FPS</td>
<td>0.5 FPS</td>
</tr>
<tr>
<td>Game Title Quality</td>
<td>Accurate Reflections</td>
<td>Soft Shadows</td>
</tr>
<tr>
<td>Multi-Pass Effects</td>
<td>Accurate Shadows</td>
<td>Glossy Reflections</td>
</tr>
<tr>
<td>Raster AO</td>
<td>Multi-Bounce Diffuse, etc.</td>
<td></td>
</tr>
<tr>
<td>Soft Shadows, etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120 FPS</td>
<td>15 FPS*</td>
<td>10 FPS</td>
</tr>
<tr>
<td>Stereo</td>
<td>20 FPS</td>
<td>0.5 FPS</td>
</tr>
<tr>
<td>Game Title Quality</td>
<td>Accurate Reflections</td>
<td>Soft Shadows</td>
</tr>
<tr>
<td>Multi-Pass Effects</td>
<td>Accurate Shadows</td>
<td>Glossy Reflections</td>
</tr>
<tr>
<td>Raster AO</td>
<td>Multi-Bounce Diffuse, etc.</td>
<td></td>
</tr>
<tr>
<td>Soft Shadows, etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Strength: Very High Resolutions | No / Little Noise while Interacting | Physically Based |
| Weakness: Physically Approximate | Physically Plausible | Noisy while Resolving |

API calls for which mode to use, with what features, what to do on mouse-up, etc. enable applications to have custom personalities for behavior and look

Shipping now as part of Bunkspeed 2014 products
Iray 2013 Rendering Modes

- Iray Photoreal
  - Interactive but “noisy”
  - The overall scene resolves in a couple of minutes

- Iray Interactive
  - Interactive with minimal noise
  - Shadows, glossiness and AA resolve in a couple of seconds
  - Can be made faster with lower quality settings
Iray Photoreal in a few minutes
Iray Interactive in a few seconds
Making Modes Seamless - Shared Materials

- Physically Based for accuracy (BSDF)
- Layered for great flexibility
- Consistent appearance across modes
- Designed to accommodate measured data
MDL - Material Description Language

NOT a shading language, but a canonical representation which renderers can target as they see fit

For material artists:
- Easy to parse and understand
- No algorithmic knowledge required
- Parameters easily exposed

For end users:
- Assign and edit parameters at will
Exploring MDL - NVIDIA Iray’s Material Description Language

Lutz Kettner
NVIDIA Iray 2013 - Rendering Modes

**Goal:**
Share Scene Database & Material Description for a consistent look

---

**Iray Realtime**
- 120 FPS

**Iray Interactive**
- 15 FPS
- 20 FPS

**Iray Photoreal**
- 2 FPS
- 10 FPS
- Minutes

---

**Scene database**
- Camera
- Lights
- Geometry
- Material instances

**Material catalogues**
Traditional Shading Language Elements

Texturing
- Texture lookups
- Procedurals
- uv-transforms
- Projectors
- Noise functions
- Math functions
- Render state

Material Definition
- Glossy reflection
- Transparency
- Translucency

Material Implementation
- Light loops / Trace N rays
- OIT / ray-continuation
- Ray-marching
Procedural Programming Language
- Texture lookups
- Procedurals
- uv-transforms
- Projectors
- Noise functions
- Math functions
- Render State

Declarative Material Definition
- Glossy reflection
- Transparency
- Translucency

Renderer
- Rasterizer
  - Light loops
  - OIT
- Ray Tracer
  - Trace N rays
- Path Tracer
  - Ray-marching
MDL is Not a Shading Language

MDL defines what to compute, *not* how to compute it

- no programmable shading
- no light loops or access to illumination
- no trace call
- no sampling
- no camera dependence
MDL Key Features

- Hierarchical layered material model from NVIDIA Iray
  - More fine grain control than material layering alone
  - Simple elemental building blocks, no monolithic “uber-shader”
  - Easy ways of combining building blocks
    - layering - mixing - modification
    - truly hierarchical: combined parts can be used in further combinations
  - Examples:
    - two materials plus a selection mask to model tile & grout patterns or mud splats on car paint
    - a complex glossy reflection lobe built from a mix of simple lobes as a building block for complex layered materials
MDL Key Features

- Declarative material definition with physically-based material model
  - Easy-to-use paradigm
  - Independence from rendering algorithms
  - Support for modern rendering algorithms: path tracers, MIS, ...
  - Material structure is known without executing any function
  - Early optimizations and simpler compiler
MDL Key Features

- **Parameterized material definitions**
  - Enables flexible, custom-built and domain-specific material catalogues
  - Material re-use and customization

- **Module and package system**
  - Supports packaging and distribution of material catalogues

- **Pure, side-effect free functions and read-only render state**
  - Support for highly parallel execution models
Example

Consistency of MDL materials across NVIDIA Iray Rendering Modes
MDL Material Model

- Physically based
  - Bidirectional Scattering Distribution Functions (BSDF)
    - specular - glossy - diffuse
    - reflective - refractive
  - Emissive Distribution Functions (EDF)
  - Volume Distribution Functions (VDF)

- Hierarchical layered material model from NVIDIA Iray
  - Elemental building blocks - modifiers - layering operators

- Comprehensive texturing functionality
  - Scores of built-in functions provide a flexible instruction set for runtime execution (in NVIDIA Iray 2013, etc.)
  - Full programmability in the future - without losing GPU acceleration
Procedural Programming Language

- Pure side-effect free functions with read-only render state
- Function results feed into material and function parameters
- “Shader graphs” are equivalent to function call graphs

```
coordinate_source
  texture_space: 1

file_texture
  texture: wood
  uww: coordinate_source( texture_space: 1 )
```

Texture lookup
Texture placement
Procedural textures
MDL Material Model

material

- thin_walled
- ior

backface

surface

- bsdf_scattering

emission

- edf_emission
- intensity

volume

- vdf_scattering
- scattering_coefficient
- absorption_coefficient

geometry

- displacement
- cutout_opacity
- normal
Elemental Distribution Functions

Bidirectional Scattering Distribution Functions (BSDF)

- Diffuse Reflection
- Diffuse Transmission
- Simple Glossy
- Backscattering Glossy
- Specular Pure Reflection
- Specular Reflection & Transmission
- Specular Reflection & Transmission
Elemental Distribution Functions

Emissive Distribution Functions (EDF)

- Diffuse
- Spot
- IES Profile

Volume Distribution Functions (VDF)

- Anisotropic Absorption & SSS
- Anisotropic + IOR & Internal Scattering
- Anisotropic w/ Light
Layered Material Example
diffuse
tint: red
diffuse
tint: red
custom-curve layering
diffuse
tint: yellow
weighted layering
glossy
roughness:
diffuse tint: red

diffuse tint: yellow

glossy roughness:

weighted layering

weighted layering
Material Definition: Mixer
Material Definition: Mixer

- mask
- material m1
- material m2
Material Definition: Mixer

Material mixer
- mask
- material m1
- material m2

Dust
- param1
- param2

Paint
- param1
- param2
Material Definition: Mixer

- mask texture
  - param1
  - param2

- dust
  - param1
  - param2

- paint
  - param1
  - param2

- material mixer
  - mask
  - material m1
  - material m2
Material Definition: Mixer

mask texture
- param1
- param2

material mixer
- mask
- material m1
- material m2

dust
- param1
- param2

paint
- param1
- param2
Material Definition: Mixer

- material mixer
- mask
- material m1
- material m2
Material Definition: Mixer

Material Definition:

- mask
- material m1
- material m2
Material Definition: Mixer

- **material mixer**
  - mask
  - material m1
  - material m2
  - weighted_layer
    - weight
    - bsdf_layer
    - bsdf_base
  - surface
    - bsdf_scattering
Material Definition: Mixer

- mask
- material m1
- material m2
- select surface.scattering
- weight
- weighted_layer
  - weight
  - bsdf_layer
    - bsdf_base
    - bsdf_scattering
- surface
Material Definition: Mixer

material mixer

- mask
- material m1
- material m2

weighted_layer
- weight
- bsdf layer
- bsdf base

select surface.scattering

surface
- bsdf scattering
Material Definition: Mixer

- mask
- material m1
- material m2
- select surface.scattering
- weighted_layer
- weight
- bsdf layer
- bsdf base
- surface
- bsdf scattering
- geometry
- normal
Material Definition: Mixer

```
mask
material m1
material m2

select surface.scattering
select surface.scattering

select geometry.normal

weighted_layer
weight

bsdf layer
bsdf base

surface
bsdf scattering

geometry
normal
```
Material Definition: Mixer

```
mask * m1.geometry.normal + (1-mask) * m2.geometry.normal
```
Measured Materials with NVIDIA Iray 2014

Measured Isotropic BRDF

- 3d data
- MDL BSDF type measured_bsdf takes this data
- Can be used just as any other MDL BSDF
- Fully supported in Iray Photoreal and Iray Interactive

Measured Reflectance Curves
BRDF Measurement Example

- BRDF measurement provided by Ford Motor Company
- Data taken directly from Radiant Zemax IS device
BRDF Measurement Example

measured_bsdf

Fresnel layering

specular

measurement: ...
Building Material Systems on top of MDL

MDL provides a few flexible concepts ...
- Constants, Enums, Types
- Functions with input parameters; Return values feed into other parameters
- Materials with input parameters
- Materials are types too and can be used as material parameters

... that can be used to realize many different material paradigms
- Library of monolithic materials (possibly with parameters), or
- Uber-materials with parameters for all uses, or
- Strict sequential layering of materials, or
- Hierarchical layering of materials or material components, or
- Or whatever you dream up!
Increasing Access to Iray with Lightwork Design
David Hutchinson from Lightworks
NVIDIA & Lightworks Unite on Iray®

Lightworks - bringing to Iray decades of experience assisting +80 companies and sites with their rendering solutions

Lightworks will now assist companies wanting to include Iray

Making Iray quicker to integrate and even easier to use with Iray+
Iray®+ from Lightworks

All the power of Iray
- Packaged in a new and intuitive way
- Reduce integration times from months to days
- New approach to presenting MDL materials

Initially
- New level of ease of use to Iray through the Iray+ SDK
- Fast to integrate and evaluate - days not months
- UI concepts presented through the SDK

Going forward
- Iray+ will add more value in response to customer requirements
- Remove barriers to use of Iray and break into new areas
Interested in Iray+?

Contact iray@lightworkdesign.com
Stand #154 at SIGGRAPH

Expertise is servicing:
- CAD Software OEMs
- Point of Sale or Configurator solutions
- Specialised Material requirements
- Corporate Cluster and Cloud Rendering
Plugging into NVIDIA Rendering

www.nvidia-arc.com
iray@lightworks.com
https://developer.nvidia.com/optix
NVIDIA Visualization Theater Booth Talks

Honda: Automotive Design Appearance Simulation
  Tuesday 10:00 AM - 10:30 AM

Dassault Systems: CATIA Live Rendering with NVIDIA Iray
  Tuesday 12:00 PM - 12:30 PM

Pixar: Using OptiX™ for Lighting Preview in a Production Pipeline
  Tuesday 4:00 PM - 4:30 PM
  Wednesday 2:00 PM - 2:30 PM
  Thursday 10:40 AM - 11:10 AM

NVIDIA: Introducing Iray for Maya
  Tuesday 10:40 AM - 11:10 AM

Bunkspeed: Bringing NVIDIA Iray and Ease of Use to Designers
  Thursday 1:20 PM - 1:50 PM