Multi-Scale Global Illumination in Quantum Break

Ari Silvennoinen
Remedy Entertainment
Aalto University

Ville Timonen
Remedy Entertainment

SIGGRAPH 2015: Advances in Real-Time Rendering course
Remedy Entertainment

Max Payne
New York, Fugitive Undercover Cop, Nothing to Lose

Max Payne 2
The Fall of Max Payne
A Film Noir Love Story

Alan Wake
A Psychological Action Thriller

American Nightmare
Custom in-house engine
Physically based light pre-pass renderer
Design Goals and Constraints

Consistency
Design Goals and Constraints

Consistency

Semi-dynamic environments and lighting
Design Goals and Constraints

Consistency
Semi-dynamic environments and lighting
Fully automatic
Large Scale Lighting
Multi-Scale Lighting
Talk Outline

Part I: Large-scale lighting

Part II: Screen-space lighting
Talk Outline

Part I: Large-scale lighting

Part II: Screen-space lighting
Possible Solutions for Global Illumination

Dynamic Approaches

- Virtual Point Lights (VPLs) [Keller97]
- Light Propagation Volumes [Kaplaynan10]
- Voxel Cone Tracing [Crassin11]
- Distance Field Tracing [Wright15]
Possible Solutions for Global Illumination

Dynamic Approaches

- Virtual Point Lights (VPLs) [Keller97]
- Light Propagation Volumes [Kaplaynan10]
- Voxel Cone Tracing [Crassin11]
- Distance Field Tracing [Wright15]

Cost was too high for the **quality** we wanted
Possible Solutions for Global Illumination

Mesh-based Precomputation
- Precomputed Radiance Transfer (PRT) [Sloan02]
- Spherical Harmonic Light Maps

Meshless Precomputation
- Irradiance Volumes [Greger98]
Most traditional rendering algorithms fire rays through each pixel—path tracing, etc.—determine intensity by averaging many samples (MC sampling)—sample all possible paths light can take from source to sensor.

Not useful for finding derivatives.

Mesh-based Precomputation
- Precomputed Radiance Transfer (PRT) [Sloan02]
- Spherical Harmonic Light Maps

Meshless Precomputation
- Irradiance Volumes [Greger98]
Irradiance Volumes

[Greger 1998]
Irradiance Volumes

[Greger 1998]
Irradiance Volumes

[Greger 1998]
Irradiance Volumes

[Greger 1998]
Global Illumination Volumes

Augment irradiance volumes with global illumination data
Lighting Only

Local Irradiance

Indirect Sun Light Transport

Sky Light Transport
Lighting Only

Indirect Sun Light Transport

Local Irradiance

Sky Light Transport
Local Irradiance

Lighting Only

Indirect Sun Light Transport

Sky Light Transport
Global Illumination Volumes

- No UVs
- Works for LOD models
- Volumetric lighting
- Consistent with dynamic objects
Global Illumination Volumes

- No UVs
- Works for LOD models
- Volumetric lighting
- Consistent with dynamic objects

Specular **infeasible** due to data size
Specular Reflections
Specular Reflections
Specular Reflections
Specular Reflections

Screen space info available
Specular Reflections

Fall back to local reflection probes
How to Blend Reflection Probes?

Fall back to local reflection probes
How to Blend Reflection Probes?
Reflection Probe Visibility
Reflection Probe Visibility
Reflection Probe Visibility
Reflection Probe Visibility
Reflection Probe Visibility
Reflection Probe Visibility

Main idea: extend global illumination volumes to store reflection probe visibility
Main idea: extend global illumination volumes to store reflection probe visibility.
Main idea: extend global illumination volumes to store reflection probe visibility
Main idea: extend global illumination volumes to store reflection probe visibility.
Reflection Probe Visibility

Main idea: extend global illumination volumes to store reflection probe visibility
Reflection Probe Visibility

Main idea: extend global illumination volumes to store reflection probe visibility
Reflection Probe Visibility

Main idea: extend global illumination volumes to store reflection probe visibility

Voxel
Reflection Probe Visibility

Main idea: extend global illumination volumes to store reflection probe visibility
Reflection Probe Visibility

Main idea: extend global illumination volumes to store reflection probe visibility

Store best reflection probes in the voxel
Reflection Probes
Reflection Probes
Where to Place Reflection Probes?
Where to Place Reflection Probes?

Not too close to geometry
Where to Place Reflection Probes?

Not too far from geometry
Observation

Maximise **visible surface area**

Minimize **distance** to surface
Automatic Probe Placement

Maximise **visible surface area**

Minimize **distance** to surface
Automatic Probe Placement

Maximise \textit{visible surface area}

Minimise \textit{distance} to surface

Choose K best probe locations
Probe Placement
Global Illumination Data

Local Irradiance

Indirect Sun Light Transport

Sky Light Transport

Specular Probe Visibility

Specular Probe Atlas
Global Illumination Data

Local Irradiance

Indirect Sun Light Transport

Sky Light Transport

Specular Probe Visibility

Specular Probe Atlas
Related Work

GPU Volume Textures
- Can’t use native interpolation due to compression

GPU Sparse Textures
- Too large pages for fine grained tree structure
- May not be available on target platforms for future games
Related Work

Adaptive Volumetric Data Structures

— Irradiance Volumes [Greger98, Tatarchuk05]
— GigaVoxels [Crassin09]
— Sparse Voxel Octrees [Laine and Karras 2010]
— Tetrahedralization, e.g., [Cupisz12], [Bentley14], [Valient14]
— Sparse Voxel DAGs [Kämpe13]
— Open VDB [Museth13]
Adaptive Voxel Tree

**Implicit** spatial partitioning

Branching factor of 64

Multi-scale data
Adaptive Voxel Tree

Implicit spatial partitioning
Branching factor of 64
Multi-scale data
Adaptive Voxel Tree

**Implicit** spatial partitioning

Branching factor of 64

Multi-scale data
Adaptive Voxel Tree

**Implicit** spatial partitioning
Branching factor of 64
Multi-scale data
Adaptive Voxel Tree

**Implicit** spatial partitioning

Branching factor of 64

Multi-scale data
Adaptive Voxel Tree

**Implicit** spatial partitioning
Branching factor of 64
Multi-scale data
Voxel Tree Structure

Voxel Node Array
**Node Structure**

- **Child Mask**: 64 bits
- **Child Block Offset**: 31 bits, 1 bit
- **Terminal Node Bit**: 1 bit

**Voxel Node Array**
Child Mask

Node Structure

Child Mask

Child Block Offset

Terminal Node Bit

Voxel Grid

Voxel Node Array
Child Mask

Node Structure

Child Mask

Child Block Offset

Terminal Node Bit

Voxel Grid

Voxel Node Array

SIGGRAPH 2015: Advances in Real-Time Rendering course
Tree Traversal

Voxel Grid

Node Structure

Child Mask

64 bits

Child Block Offset

31 bits

Terminal Node Bit

1 bit

Voxel Node Array

Child Index = ?
Tree Traversal

Node Structure

Voxel Grid

Child Mask
64 bits

Child Block Offset
31 bits

Terminal Node Bit
1 bit

Child Index =

Voxel Node Array

SIGGRAPH 2015: Advances in Real-Time Rendering course
Tree Traversal

Node Structure

Voxel Grid

Child Mask

64 bits

Child Block Offset

31 bits

Terminal Node Bit

1 bit

Child Index = 3 set bits + Child Block Offset

Voxel Node Array

SIGGRAPH 2015: Advances in Real-Time Rendering course
Tree Traversal

Voxel Grid

Node Structure

Child Mask

64 bits

Child Block Offset

31 bits

Terminal Node Bit

1 bit

Child Index = 3 set bits + Child Block Offset

Voxel Node Array

SIGGRAPH 2015: Advances in Real-Time Rendering course
Payload Index = Child Index = 3 set bits + Child Block Offset

Voxel Grid

Node Structure

Payload Data

Voxel Node Array
## Payload Data

<table>
<thead>
<tr>
<th>Feature</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Light Irradiance</td>
<td><img src="image1.png" alt="Image" /></td>
</tr>
<tr>
<td>Indirect Sun Light Transport</td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>Sky Light Transport</td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>Specular Probe Visibility</td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>

### Voxel Node Array

- Row 1: Voxel 1, Voxel 2, Voxel 3, Voxel 4
- Row 2: Voxel 5, Voxel 6, Voxel 7, Voxel 8
- Row 3: Voxel 9, Voxel 10, Voxel 11, Voxel 12
- Row 4: Voxel 13, Voxel 14, Voxel 15, Voxel 16

---

SIGGRAPH 2015: Advances in Real-Time Rendering course
What About Leaf Nodes?

Leaf nodes are *implicit*: they only show up in the child masks of their parent voxels.

**Compact** trees encoding only the **topology**

Only a **few hundred kilobytes** for an entire level.
First Level Lookup

First level of the tree can have *arbitrary* dimensions

We use a *dense* grid of 8x8x8 meter cells to guarantee coverage for large dynamic objects
Voxel Tree Visualisation

50 cm
Seamless Interpolation

Dynamic and static objects lit by same data
Need seamless interpolation everywhere
Seamless Interpolation

Query point in solid leaf
Seamless Interpolation

Trilinear neighborhood

Query point in solid leaf
Seamless Interpolation

Query point in empty leaf
Seamless Interpolation

Query point in empty leaf

Use dilated tree?
Seamless Interpolation

Query point in empty leaf

Use dilated tree?
Seamless Interpolation

Query point in empty leaf

Trilinear neighborhood
Seamless Interpolation

Query point in empty leaf

Interpolate from parent node

Trilinear neighborhood
Seamless Interpolation

Query point in empty leaf

Apply partial dilation to avoid recursion

Interpolate from parent node

Trilinear neighborhood

SIGGRAPH 2015: Advances in Real-Time Rendering course
Seamless Interpolation

![Image of seamless interpolation with different voxel resolutions: 0.5m, 2m, 8m voxels.]

SIGGRAPH 2015: Advances in Real-Time Rendering course
Seamless Interpolation

0.5m voxels
2m voxels
8m voxels
Multiply trilinear weight with $\max(0, \cos \theta)$
Scaling to Large Scenes

World is divided into a cell grid for streaming

Per cell voxel tree

World Atlas

128 m

128 m

128 m
Scaling to Large Scenes

Linear GPU arrays

World Atlas
Global Illumination
Global Illumination

Screen-Space + Ambient
Performance

Each world cell has max 65K diffuse GI data points
Comparable to **256x256** light map
Performance

Use reflector lights to avoid dynamic fill lights

Local Irradiance
Real-Time Indirect

Direct Only

Reference Indirect

Global Illumination

Real-Time Indirect

Reference Indirect

SIGGRAPH 2015: Advances in Real-Time Rendering course
Volumetric Global Illumination
Global Illumination

Constant Ambient
Summary

**Unified** approach to large scale lighting

Fully **automatic** specular probe system
Talk Outline

Part I: Large-scale lighting

Part II: Screen space lighting
Screen-Space Techniques

Requirements

— Occlude larger scale lighting
— Fill in with screen-space sampled lighting
Screen-Space Ambient Occlusion and Diffuse
GI diffuse occlusion

Screen-Space Diffuse

SIGGRAPH 2015: Advances in Real-Time Rendering course
Screen-Space Ambient Occlusion

Based on Line-Sweep Ambient Obscurance [Timonen2013]:
LSAO locates most contributing occluders
Screen-Space Ambient Occlusion

We scan in 36 directions, long steps (~10px) and short line spacing (~2px apart)

— Scheduling friendly for the GPU
— Scan is 0.75ms on Xbox One at 720p
Screen-Space Ambient Occlusion

regular

jittered
Screen-Space Ambient Occlusion

- An additional near field sample (at ~2px distance)
- Sample normal to clamp occluders
Screen-Space Ambient Occlusion

36 directions too expensive to gather per pixel

- Interleave on a 3x3 neighborhood (4 dirs/pixel)
- Gather using a depth and normal aware 3x3 box filter
Screen-Space Ambient Occlusion — 1.4ms @ 720p on XB1
Screen-Space Ambient Occlusion — 1.4ms @ 720p on XB1
Screen-Space Diffuse Lighting
Screen-Space Diffuse Lighting

LSAO samples are “the most visible”

— Good candidates to sample incident light
— Can’t be occluded by definition (providing self-occlusion)
Screen-Space Ambient Occlusion ON
Screen-Space Diffuse Lighting OFF
Screen-Space Ambient Occlusion OFF
Screen-Space Diffuse Lighting OFF
Screen-Space Ambient Occlusion ON
Screen-Space Diffuse Lighting OFF
Screen-Space Ambient Occlusion ON
Screen-Space Diffuse Lighting ON
Screen-Space Reflections and Occlusion
GI specular occlusion

Screen-Space Specular
Screen-Space Reflections

1 ray per pixel from GGX distribution, evaluated for all surfaces

- Linear search (7 steps)
- Step distances form a geometric series
Screen-Space Reflections

Treating the depth buffer samples

Need to support varying roughness
  — Calculate cone coverage

Need to suit both occlusion and color sampling
  — Also find a single color sample location
Screen-Space Reflections

Depth thickness =
a + b*(distance along the ray)

Depth field extends to/from camera, not along view z!
Screen-Space Reflections

Match the linear term to step size in view space. Otherwise holes on solid geometry:
For occlusion, calculate max coverage of the cone

Clamp the cone’s lower bound to surface tangent!
Screen-Space Reflection Occlusion — 0.8 ms @ 720p on XB1
Screen-Space Reflections

For **color**, we need a single sample location

First, we pick the sample that covered most of the cone
Screen-Space Reflections

Aim the reflection ray towards the center of the coverage

And intersect with the line between the last 2 samples
Screen-Space Reflections

Low sample density: interpolate towards camera direction (in blue)
Screen-Space Reflections

Previous sample above ray: don’t interpolate
Screen-Space Reflections — 0.5 ms @ 720p on XB1

Final image
Screen-Space Reflection Occlusion OFF
Screen-Space Reflections OFF
Screen-Space Reflection Occlusion ON
Screen-Space Reflections OFF
Screen-Space Reflection Occlusion ON
Screen-Space Reflections ON
smooth

rough
Refining the intersections

If neighboring rays have the same direction

- Interleave search
- Take nearest hit distance
Thank You!

Acknowledgments
Tatu Aalto
Janne Pulkkinen
Laurent Harduin
Natalya Tatarchuk
Jaakko Lehtinen
References