The Edge Volume Heuristic

Robust Triangle Subdivision for Improved BVH Performance

Holger Dammertz, Alexander Keller

Holger Dammertz | August 10, 2008
Introduction

Disadvantage of Object Partitioning using Bounding Volumes

- Bounding volume does not pack primitives perfectly
- Primitives are of varying size
- The result is overlap of tree nodes and children wasting traversal time
Introduction

Disadvantage of Object Partitioning using Bounding Volumes

- Bounding volume does not pack primitives perfectly
- Primitives are of varying size
- The result is overlap of tree nodes and children wasting traversal time
Introduction

Disadvantage of Object Partitioning using Bounding Volumes

- Bounding volume does not pack primitives perfectly
- Primitives are of varying size
- The result is overlap of tree nodes and children wasting traversal time

Advantage of Bounding Volume Hierarchies

- Easy and fast traversal
- Memory behaviour deterministic and controllable
- Very small acceleration structures possible
Splitting

Early Split Clipping Approach

- Presented last year on RT07
- Based on surface area of triangles
- Split each triangle at axis aligned plane
- Good performance with correct parameters
Splitting

Early Split Clipping Approach

- Presented last year on RT07
- Based on surface area of triangles
- Split each triangle at axis aligned plane
- Good performance with correct parameters

"by ingenious - OMPF Forum"

There is no good global early clipping heuristic.
Try presplitting the triangles in Soda Hall - unless you’re a magician and just guess the correct splitting threshold, you’ll end up with out of memory exception during tree building :D
Hand Tweaking may be difficult

Problems of Early Split Clipping

- Memory consumption unpredictable
- Clipping against planes may introduce numerical problems
- May split many triangles without any speed improvement
Hand Tweaking may be difficult

Problems of Early Split Clipping
- Memory consumption unpredictable
- Clipping against planes may introduce numerical problems
- May split many triangles without any speed improvement

Animations
- Many different shots
- Scene complexity may change during a shot
Hand Tweaking may be difficult

Problems of Early Split Clipping
- Memory consumption unpredictable
- Clipping against planes may introduce numerical problems
- May split many triangles without any speed improvement

Animations
- Many different shots
- Scene complexity may change during a shot

Advantages of creating smaller triangles
Basic principle similar to \( kd \) reference duplication
- Controllable which triangles to split
- Preprocess before tree construction
Our Approach

Preserve advantage of BVH

- Small memory consumption - split only really bad triangles
- Numerically robust and consistent
- Fast
- Automatic for normal scenes
Our Approach

Preserve advantage of BVH

- Small memory consumption - split only really bad triangles
- Numerically robust and consistent
- Fast
- Automatic for normal scenes

Observation

- Not all large triangles are bad, especially in architectural scenes
- Large triangles are often axis aligned and form quads
- These triangles are no huge problem in SAH based tree construction
Subdivision

Reduce overlap

- Triangle Subdivision at edge midpoint also reduces overlap
Subdivision

Reduce overlap

- Triangle Subdivision at edge midpoint also reduces overlap
**Subdivision**

**Reduce overlap**
- Triangle Subdivision at edge midpoint also reduces overlap

**Edge Volume Heuristic**
- Split a triangle if volume of AABB of a triangle edge exceeds threshold $\varepsilon_v(t)$.
- Use midpoint of edge for split
  - $\frac{A+B}{2}$ is numerically very robust
  - Adjacent triangles sharing an edge are split at the same point
  - Watertight Subdivision: a consistent mesh stays consistent
  - Each triangle is independent
Subdivision

Edge AABB Volume

- Axis aligned edges have no AABB volume
- Very economic splitting: only very bad triangles are split

Choice of Threshold

- Global Threshold based on full scene AABB volume $V$
  \[
  \varepsilon_v(t) := \frac{V}{2^t}
  \]
- Use $t = 14$ as default value.
  - Over a broad range of scenes a very good default value
  - Less than 1% of reference duplication in already fast scenes
Implementation Details

### Preprocess

Before ray tracing by scene exporter

- Larger memory consumption
- Transparent for underlying ray tracer
Implementation Details

Preprocess

Before ray tracing by scene exporter
- Larger memory consumption
- Transparent for underlying ray tracer

On demand

- Compute only bounding boxes prior to tree construction
- Memory efficient (only replicated indices needed)
- Transparent for user
- Pre-Scan is efficient
Implementation Details

Preprocess

Before ray tracing by scene exporter
- Larger memory consumption
- Transparent for underlying ray tracer

On demand

- Compute only bounding boxes prior to tree construction
- Memory efficient (only replicated indices needed)
- Transparent for user
- Pre-Scan is efficient

Additional optimization

- BVH has usually more than one triangle per leaf
- Remove duplicates
Results - Static Scenes

Bunny, Buddha, Dragon...

No triangles are split
Results - Static Scenes

Space Ship

![Image of a space ship](image-url)

![Graph showing the relationship between threshold and performance metrics](graph-url)
Results - Static Scenes

Rotated Kitchen

![Image]

Graph showing the relationship between threshold and the number of triangles and render time percentage.
Results - Simple Animation

Rotated Sponza - Performance Statistics

The graph shows performance statistics for different thresholds during animation. The x-axis represents the frame number, and the y-axis represents the frame time in milliseconds. The graph compares the base performance with performances at different threshold levels: Threshold 12, Threshold 14, and Threshold 16.
Results - Simple Animation

Rotated Sponza - Triangle Statistics
Summary

- Economical heuristic to subdivide triangles
- Efficiently reduces the overlap of triangles
- Numerically robust and topology unaware preprocess
- Simple global threshold

Acknowledgements

- mental images GmbH for support and funding of this research
Summary

▶ Economical heuristic to subdivide triangles
▶ Efficiently reduces the overlap of triangles
▶ Numerically robust and topology unaware preprocess
▶ Simple global threshold

Acknowledgements

▶ mental images GmbH for support and funding of this research

Questions?