

VU Rendering SS 2015

186.101

Thomas Auzinger
Károly Zsolnai

Institute of Computer Graphics and Algorithms (E186)
Vienna University of Technology

<http://www.cg.tuwien.ac.at/staff/ThomasAuzinger.html>

<http://www.cg.tuwien.ac.at/staff/KarolyZsolnai.html>

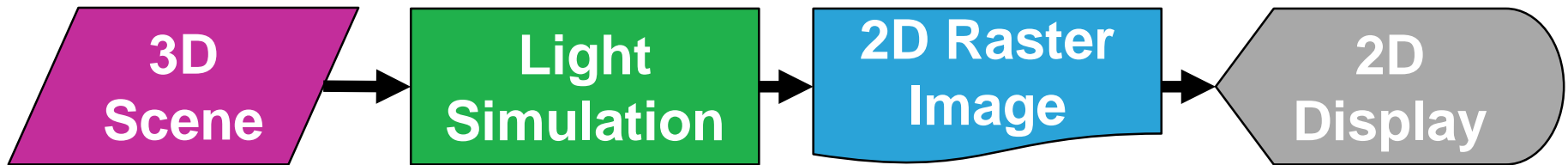


VU Rendering SS 2015

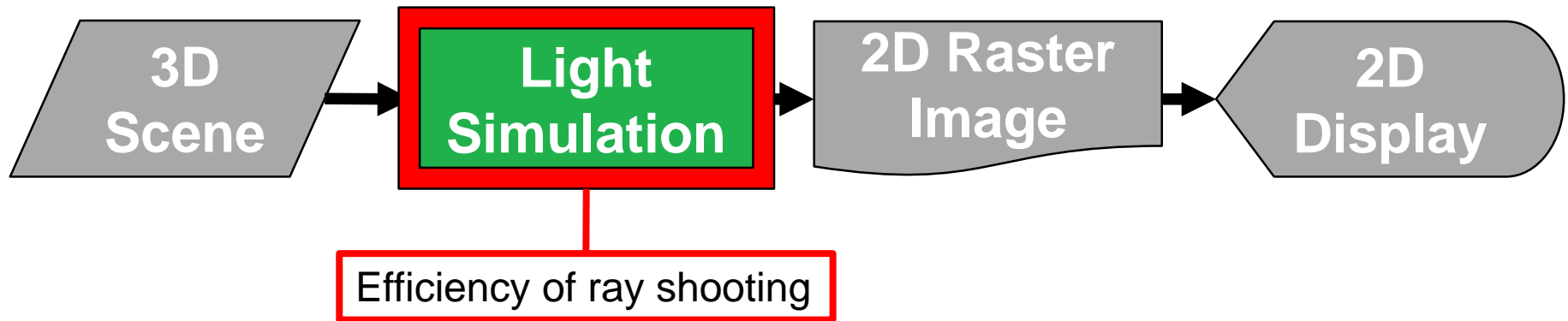
Unit 04 – Spatial Acceleration Structures



Rendering pipeline



Rendering pipeline



Generally, Monte-Carlo methods use ray shooting to sample the integrand of the rendering equation.

Computing the closest intersection with the scene is equivalent to computing the local visibility.

This is very expensive for a large amount of scene objects.



- **Naive approach:**

Intersect ray with all objects to determine the closest intersection ($O(n)$ for n objects)

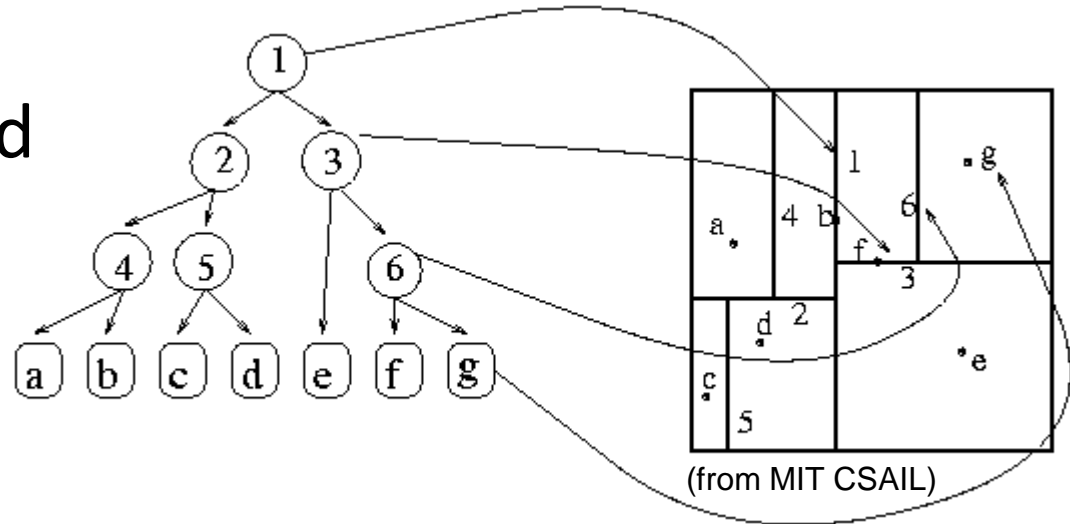
- **Better approach:**

Reorganize the objects into a spatial hierarchy to skip large parts of the scene ($o(n)$ for n objects)



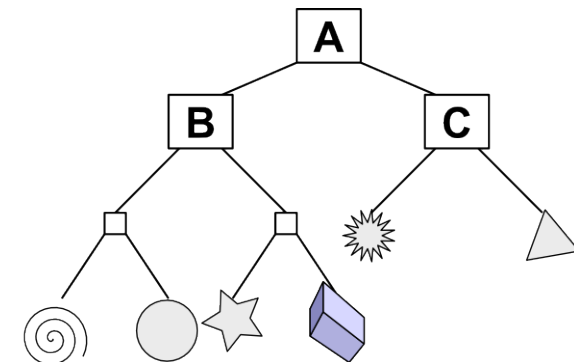
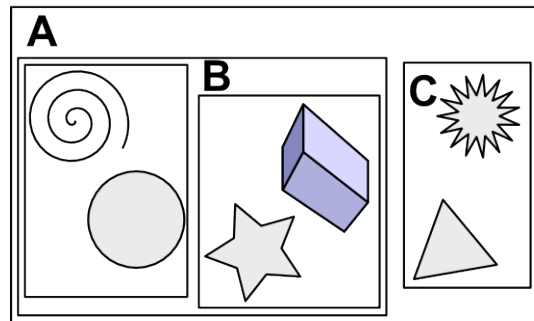
■ k-d tree

Subdivide space and store the objects in overlapping nodes



■ Bounding Volume Hierarchy (BVH)

Group objects recursively into a tree structure



(from wikipedia)



■ k-d tree

+ Faster traversal on the CPU

- Larger amount of nodes (and duplicate references)

■ Bounding Volume Hierarchy (BVH)

+ Faster traversal on the GPU

+ Easier to update (for dynamic scenes)

+ Every object only in one tree leaf

- Spatial overlap of nodes



Object grouping can be done in many ways – optimality is usually scene dependent but heuristics exist.

Surface area heuristic (SAH)

$$SAH = C_{\text{inner}} \sum_I \frac{A_n}{A_{\text{root}}} + C_{\text{leaf}} \sum_L T_n \frac{A_n}{A_{\text{root}}}$$

I ... inner nodes, L ... leaf nodes

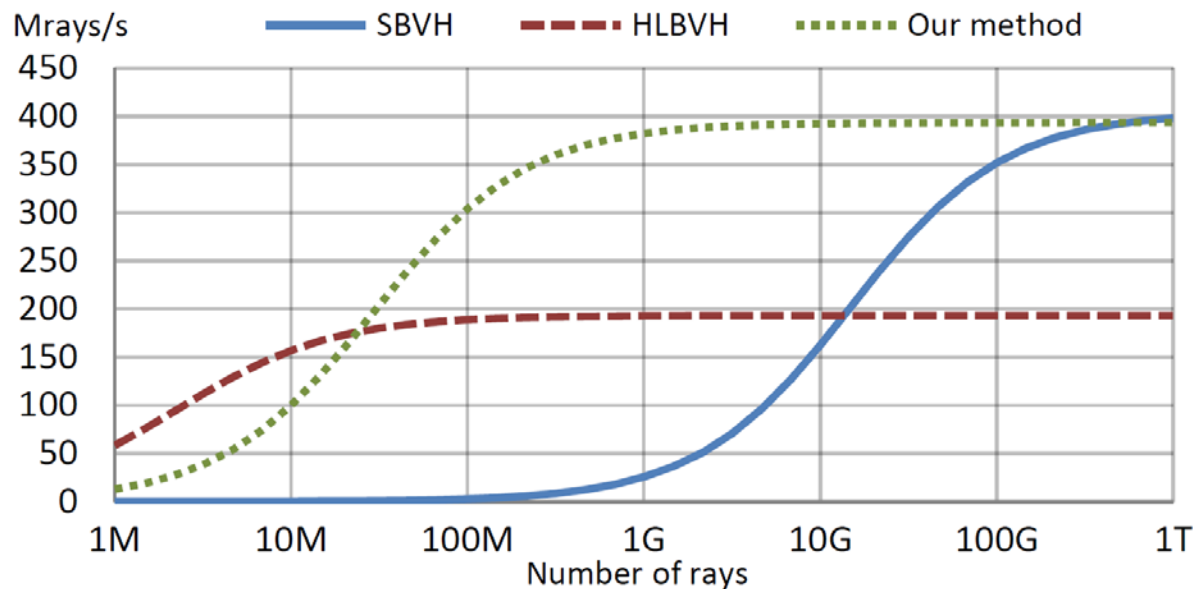
A_n ... surface area node n

A_{root} ... surface area of its root



Constructing the tree with the minimal SAH cost is expensive → usually approximations are used.

This leads to a quality/speed trade-off:



Comparison of different BVH construction methods. From [1].



Textbooks

- PBRT chapter 4
- C. Ericson, *Real-Time Collision Detection*, 2005

Papers

- I. Wald, *On fast Construction of SAH-based BVHs*, IRT 2007
- Z. Wu et al., *SAH KD-Tree construction on GPU*, HPG 2011
- T. Karras, *Maximizing Parallelism in the Construction of BVHs, Octrees and k-d Trees*, HPG 2012
- T. Karras, T. Aila, *Fast Parallel Construction of High-Quality Bounding Volume Hierarchies*, HPG 2013
- M. Doyle et al., *A Hardware Unit for Fast SAH-optimised BVH Construction*, Siggraph 2013



Questions?

model
model: BUNNY.OBJ
triangles: 1500

BVH
nodes: 2999
leafes: 1500
max depth: 14

selection
depth: -1
nodes: 0

