Efficient Sorting and Searching in Rendering Algorithms

Eurographics 2006 Tutorial T4

Organizers and Presenters

Vlastimil Havran
Czech Technical University in Prague

Jiří Bittner
Czech Technical University in Prague
Vienna University of Technology

Version 1.0, June 15, 2006

Abstract

In the proposed tutorial we would like to highlight the connection between rendering algorithms and sorting and searching as classical problems studied in computer science. We will provide both theoretical and empirical evidence that for many rendering techniques most time is spent by sorting and searching. In particular we will discuss problems and solutions for visibility computation, density estimation, and importance sampling. For each problem we mention its specific issues such as dimensionality of the search domain or online versus offline searching. We will present the underlying data structures and their enhancements in the context of specific rendering algorithms such as ray shooting, photon mapping, and hidden surface removal.

Organizers bibliographies

Vlastimil Havran is an assistant professor at the Czech Technical University in Prague since February 2006. He defended his Ph.D. dissertation on ray shooting algorithms in 2001 at the Czech Technical University in Prague. Later he joined the computer graphics group at Max-Planck-Institute for Informatics in Saarbruecken. He became a research associate at the same institute in 2003. He has contributed to the topic of sorting and searching by his dissertation on ray shooting algorithms which started the area of interactive ray tracing. In addition to sorting and searching he worked on various other topics in rendering.

Jiří Bittner holds a Ph.D. in Computer Science from the Czech Technical University in Prague. His main research interests include visibility preprocessing, occlusion culling, real-time rendering, and computational geometry. He has also been active in development of two commercial products dealing with real-time rendering of large scenes. He is currently affiliated with the Vienna University of Technology and the Czech Technical University in Prague.

Organizers contact information

Vlastimil Havran

Czech Technical University in Prague
Karlovo náměstí 13
121 35 Praha 2
Czech Republic
Phone:+420 2435 7263
Fax:+420 22492 3325
e-mail: havran@fel.cvut.cz
URL: http://www.cgg.cvut.cz/~havran

Jiří Bittner

Czech Technical University in Prague
Karlovo náměstí 13
121 35 Praha 2
Czech Republic
Phone:+420 2435 7417
Fax:+420 22492 3325
e-mail: bittner@fel.cvut.cz
URL: http://www.cgg.cvut.cz/~bittner

Vienna University of Technology
Favoritenstrasse 9-11 / E186
A-1040 Wien
Austria
Phone:+431 58801 18685
Fax:+431 58801 18698
e-mail: bittner@cg.tuwien.ac.at
URL: http://www.cg.tuwien.ac.at/staff/JiriBittner.html

Tutorial Contents

Tutorial Web Page

The updated version of this tutorial presented at Eurographics 2006 can be found under the following URL:

`http://www.cgg.cvut.cz/~havran/eg2006tut/`

Acknowledgements

We would like to thank Robert Herzog, Jaroslav Křivánek, Michael Wimmer, Peter Wonka, Tommer Leyvand, David Luebke, and Hansong Zhang for providing us various materials used in the tutorial. This work has been partially supported by the EU under the project no. IST-2-004363 (GameTools) and by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics).

Sorting and Searching in Image Synthesis

Eurographics 2006 Tutorial T4

Vlastimil Havran
Czech Technical University in Prague

Jiří Bittner
Czech Technical University in Prague
Vienna University of Technology

Content

Part One

- Introduction to Rendering
- Sorting and Searching
- Hierarchical Data Structures
- Ray Shooting
- Questions and Answers (5 minutes)

Overview 2

Content

Part Two

- Hidden Surface Removal
- Visibility Culling
- Photon Maps and Irradiance Caching
- Ray Maps
- Other Algorithms
- Questions and Answers (10 minutes)

Overview 3

Tutorial Goals

- Recall that we often use sorting and searching in rendering
- Highlight connections between different problems in rendering
- Briefly show efficient solutions
- Show unusual solutions resulting from twisting searching queries and domains

Overview 4

What is Not Covered Here

- Collision detection algorithms (EG'05 Tutorial)
- Image based rendering
- Radiosity
- Non-photo realistic rendering
- Clustering techniques
- Graph theory and other related problems

Overview 5

Introduction to Rendering

Vlastimil Havran

Czech Technical University in Prague

1

Introduction to Rendering

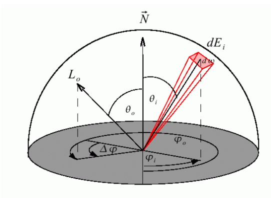
- Rendering equation
- Rendering algorithms
- Sorting and searching in rendering

Introduction to Rendering 2

Rendering Equation

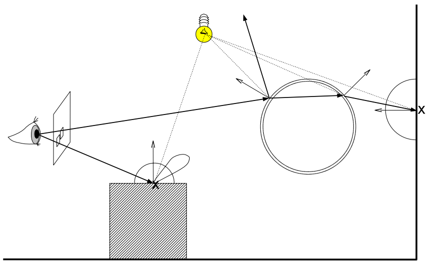
$$L(\vec{\omega_o},x) = L^e(\vec{\omega_o},x) + \int_\Omega L_i(x,\vec{\omega_i}) \cdot f_r(x,\vec{\omega_i},\vec{\omega_o}) \cdot (\vec{n}\cdot\vec{\omega_i}) d\omega_i,$$

- Convolving incoming light with surface reflectance properties



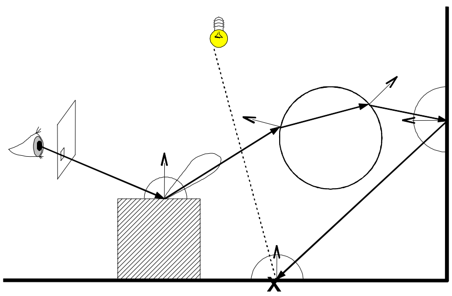
Introduction to Rendering 3

Ray Tracing



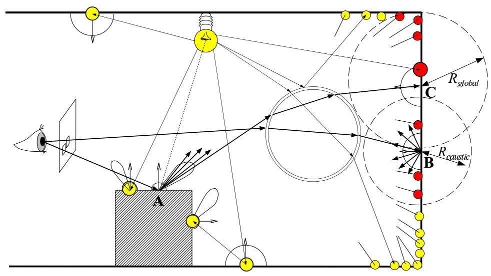
Introduction to Rendering 4

Path Tracing



Introduction to Rendering 5

Photon Mapping



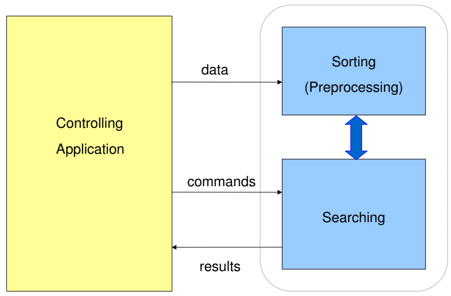
Introduction to Rendering 6

Tutorial Motivation

- Sorting and searching takes usually more than 90% of the rendering time!
- Efficiency of sorting and searching is crucial for the performance
- Examples
  - Ray Tracing
  - Photon Density Estimation

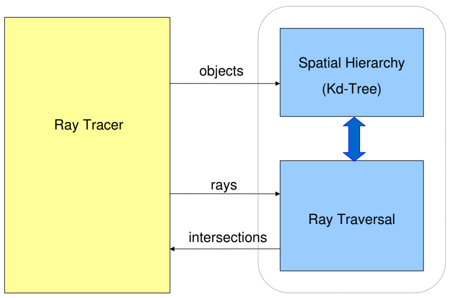
Introduction to Rendering 7

Sorting and Searching General Concept



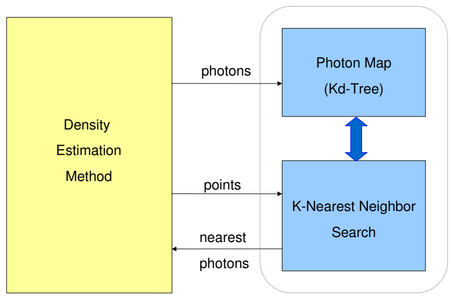
Introduction to Rendering 8

Example 1 Ray Tracing



Introduction to Rendering 9

Example 2 Photon Density Estimation



Introduction to Rendering 10

Introduction to

Sorting and Searching

Jiří Bittner

Czech Technical University in Prague
Vienna University of Technology

Searching

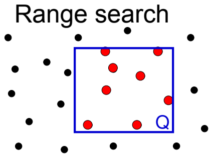
- Searching query definition

$$Q \times S \rightarrow A$$

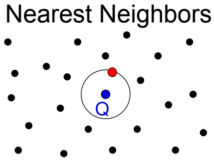
Q set of queries
S search space
A set of answers

Geometric search problems

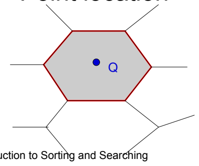
Range search



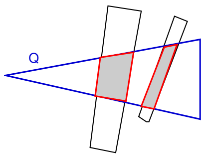
Nearest Neighbors



Point location



Intersection detection

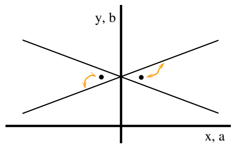


Geometric search problems

- Dimension of S

primitive	R^2	R^3
points	2	3
lines	2	4
spheres	3	4

- Example
  - Lines-points duality in 2D



Geometric search

- Exact vs. Approximate
  - Approximate: finds solution “close to” optimum
- Static vs. dynamic
  - Dynamic: S may change
- Online vs. offline
  - Offline: applied for entire sequence of Q

Search Complexity

- Single result for query
  - O(log n)
- Multiple results for query
  - Reporting O(log n + k)
  - Counting O(log n)

Typical Problems in CG

- Search space
  - Set of points, set of objects, set of oriented discs, set of hemispheres, set of rays
- Queries
  - Point, ray, hemisphere, set of points (polygon, bounding box, bounding sphere)

Introduction to Sorting and Searching 7

Typical Problems in CG

Problem	Q	S	A
Ray shooting	ray	{objects}	{point}
HSR	{rays}	{objects}	{points}
VFC	frustum	{objects}	{objects}
Occlusion culling	{rays}	{objects}	{objects}
Photon maps	point	{points}	{points}
Ray maps	hemisphere	{rays}	{rays}

Introduction to Sorting and Searching 8

Search Problem Transformation

- Halfspace intersection in 2D
  - 2D line maps to 2D point (duality)
  - *Convex hull of points*
- Point and spheres intersection in 3D
  - 3D point maps to 4D point
  - 3D spheres maps to 4D hyperplanes
  - Duality → *Halfspace range reporting in 4D*

Introduction to Sorting and Searching 9

Searching methods

- Sorted arrays
  - Binary search
  - Interpolation search
- Search trees
  - Binary search trees
  - kD-trees, R-trees, AVL-trees….
- Hashing
  - Spatial grid hashing

Introduction to Sorting and Searching 10

Sorting

- Motivation: Improve searching performance
  - Naïve search: O(n) time
  - With sorting: O(log n)!
  - In special cases even O(1)
- Assumption
  - Spatial relations among elements of S (objects, points, rays, normals, …)

Introduction to Sorting and Searching 11

Basic Sorting Algorithms

Algorithm	Method	Best	Average	Worst
Heapsort	Selection	O(n log n)	O(n log n)	O(n log n)
Selection sort	Selection	$O(n^2)$	$O(n^2)$	$O(n^2)$
Quicksort	Partitioning	O(n log n)	O(n log n)	$O(n^2)$
Bucket sort	Distribution	O(n)	O(n)	$O(n^2)$
Merge sort	Merging	O(n log n)	O(n log n)	O(n log n)
Bubble sort	Exchanging	O(n)	$O(n^2)$	$O(n^2)$
Insertion sort	Insertion	O(n)	$O(n^2)$	$O(n^2)$

Space complexity: O(n)

Introduction to Sorting and Searching 12

Examples in CG

- Quicksort
  - Top-down construction of spatial hierarchies
- Bucket sort
  - Z-buffer, voxel grid
- Heapsort
  - Priority queues (occlusion culling, k-NN search)

Introduction to Sorting and Searching 13

Hierarchical Data Structures

Vlastimil Havran

Czech Technical University in Prague

1

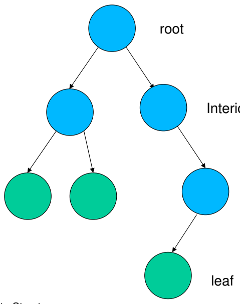
Hierarchical Data Structures (HDS)

- Connection to sorting
- Classification
- Bounding volume hierarchies
- Spatial subdivisions
- Hybrid data structures
- Searching using HDS
- Special techniques on hierarchies

Hierarchical Data Structures 2

Hierarchical Data Structures

= tree or even a graph



Hierarchical Data Structures 3

Connection to Sorting

Hierarchical Data Structures =
implementation of (spatial) sorting

Why ?

- Time complexity is O(N log N)
- For 1D hierarchy over points the construction of HDS is clearly equivalent to quicksort

Hierarchical Data Structures 4

Recall Quicksort

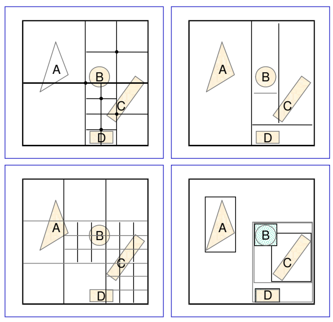
- Pick up a pivot Q
- Organize the data into two subarrays: the left part smaller than pivot Q, the right part larger or equal than pivot Q
- Recurse in both subarrays

Hierarchical Data Structures 5

Examples of HDS in 2D

octree

kd-tree



hierarchy of grids

bounding volume hierarchy

Hierarchical Data Structures 6

HDS Classification

- Data domain organization
- Dimensionality
- Data layout

Hierarchical Data Structures 7

HDS Classification

1) Data domain organization of HDS

- Spatial subdivisions – primarily organizing space (non-overlapping)
- Object hierarchies – primarily organizing objects (overlapping)
- Hybrid data structures
- Transformations and mappings

Hierarchical Data Structures 8

HDS Classification

2) Dimensionality of HDS

- Necessary to represent data entities: 1D, 2D, 3D, 4D, or 5D
- Data entities: points, lines, oriented half-lines, disks, oriented hemispheres, etc.
- Possibility to extend many problems to time domain (so plus one dimension)

Hierarchical Data Structures 9

HDS Classification

3) HDS data layout

- Internal data structures
- External data structures (out of core)
- Cache-aware data structures
- Cache oblivious data structures

Hierarchical Data Structures 10

Types of Nodes in HDS

- An interior node represents a *"pivot"* – according to the data entities are sorted
- Typical content is a subdivision plane or a set of planes plus references to child nodes
- The way of interior node representation with respect to the task is crucial for searching performance

Hierarchical Data Structures 11

Spatial Subdivisions

- **Non-overlapping *regions of child nodes***
- Space is organized by some (cutting) entities , typically by planes, constructed top-down
- Fully covering an original spatial region, every point can be located in some (empty or non-empty) leaf
- They are often called space partitionings

Hierarchical Data Structures 12

Spatial Subdivision Examples

- Kd-trees
- BSP-trees
- Octrees
- Uniform grids
- Recursive grids

Hierarchical Data Structures 13

Object Hierarchies

- Possibly **overlapping** extents of child nodes
- Often called bounding volume hierarchies
- Possibly some spatial regions are not covered
- Constructed top-down or bottom-up
- The shape represented by interior nodes typically a box, but other shapes possible

Hierarchical Data Structures 14

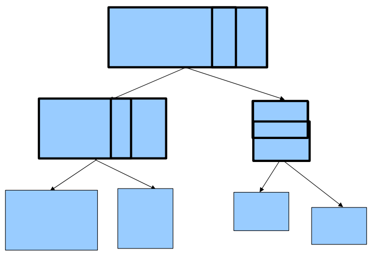
Names used for Object Hierarchies

- Bounding Volume Hierarchies (BVHs)
- R-trees and their many variants
- Box-trees
- Several others (special sort of bounding volumes... sphere trees etc.)

Hierarchical Data Structures 15

Bounding Volume Hierarchies

Constructed Top-Down



Hierarchical Data Structures 16

Hybrid Data Structures

- Combining between various interior nodes
- Possibly combining between spatial subdivisions and object hierarchies
- Sharing pros and cons of both types
- They can be tuned to compromise of some properties, for example efficiency and memory requirements

Hierarchical Data Structures 17

Other HDS

- *Content of the node* – a single splitting plane, more splitting planes, a box, additional information.
- Arity of a node (branching factor)
- A way of constructing a tree (height, weight balancing) + postprocessing
- Data only in leaves or also in interior nodes
- Augmenting data

Hierarchical Data Structures 18

Example of Other HDS

- Cell trees (polyhedral shapes for splitting)
- SKD-trees (two splitting planes at once)
- hB-trees (holey brick B-trees)
- LSD-tree (height balanced kd-tree)
- P-trees (polytope trees)
- BBD-trees (bounding box decomposition trees)
- And many others

(See the surveys listed in tutorial notes)

Hierarchical Data Structures 19

Transformation Approach

- *Input:* A spatial object in 2D or 3D domain, for example a box
- *Output:* A point in 4D or 6D domain
- More complicated mapping is possible, for example a sphere in 3D -> point in 4D
- The transform often changes the searching algorithm used completely

Hierarchical Data Structures 20

HDS Construction Algorithm

Initial Phase: create a node with all elements and put it to the auxiliary structure AS (stack or priority queue).

Top-Down, Divide and Conquer:

(1) Take a node from an auxiliary structure AS. If AS is empty, then we are finished.

(2) Take a set of elements in the node and decide if to subdivide or not. If not, create leaf, go to (1).

(3) Decide how to split the set into two (N) subsets and create new nodes.

(4) Put the new nodes to AS. Recurse.

Hierarchical Data Structures 21

Search Algorithms using HDS

- Start from a root node
- Typically down traversal phase (location phase) + some other phase
- During visiting an interior node use either a stack (LIFO) or priority queue to record the nodes to be visited in future
- Compute incidence computation when visiting a leaf

Note: auxiliary structure implements another sorting phase during searching

Hierarchical Data Structures 22

Search Algorithms using HDS

- *Range queries* – given a range X, find all the incidences of X with data
- *Nearest neighbour* – find a nearest neighbor
- *k-nearest neighbour*
- *Reverse nearest neighbors – given a point Q, find all the points to which Q is nearest neighbor*
- *Ranking* – given a query object Q, report on all the objects in order of distance from Q
- *Intersection search* – given a point Q, find all the objects that contain Q

Hierarchical Data Structures 23

Search Performance Model

- Result = the cost of computation ... C
- Performance is inverse proportional to the quality of the data structures for given problem
- The two uses of performance model
  - *a posteriori:* documenting and testing performance
  - *a priori:* constructing data structures with higher expected performance

Hierarchical Data Structures 24

Search Performance Model

Typical cost model:

C = C_T + C_L + C_R

C = C_TS * N_TS + C_LO * N_LO + C_Access * N_Access

- C_T … cost of traversing the nodes of HDS
- C_L … cost of incidence operation in leaves
- C_R … Cost of accessing the data from internal or external memory

Hierarchical Data Structures 25

HDS Dynamization

- Given changes, only update data structures to reflect these changes
- It is assumed that the performance of searching remains acceptable after update
- Dynamization can require additional bookkeeping data to monitor the cost/quality of a HDS node and its associated subtree
- Techniques known for 1D trees (rotation, balancing) are often not applicable
- It is usually required to update larger amount of data at once (bulk updating)

Hierarchical Data Structures 26

Ray Shooting

Vlastimil Havran

Czech Technical University in Prague

1

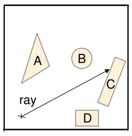
Ray Shooting

- Ray shooting versus ray tracing
- Connection to sorting and searching
- Ray shooting with kd-trees
- Performance studies
- Octrees, uniform grids, recursive grids
- Bounding volume hierarchies
- Offline ray shooting

Ray Shooting 2

Ray Shooting Algorithm (RSA)

Task: Given a ray, find out the first object intersected.



Input: a *scene* and a *ray*

Output: the *object C*

Ray Shooting 3

Ray Tracing *versus* Ray Shooting

- *Ray shooting* – only a single ray
- *Ray tracing* in computer graphics can be:
  – Ray shooting
  – Ray casting – only primary rays from camera
  – Recursive ray tracing
  – Distribution ray tracing and others

Ray Shooting 4

Some Complexity Issues

Computational Geometry
- aimed at worst-case complexity
- restriction to certain class of object shape (polygons, triangles)
- unacceptable memory requirements O(log N) query time induces Omega (N^4) space

Computer Graphics
- aimed at average-case complexity
- practical feasibility and robustness
- implementation issues important for performance

Ray Shooting 5

Some Complexity Results

Lower bound for worst-case complexity: 1997/98

Laszlo Szirmay-Kalos + Gabor Marton – lower bound for space complexity is Omega(N^4) for O(log N) search

Applicability of Computational Geometry techniques in CG for ray tracing
– CGE techniques are not general
– limited to small number of primitives
– no implementations available

Ray Shooting 6

Computer Graphics Techniques Overview

Techniques developed: average-case complexity, no complexity guarantees, many “tricks”

Basic techniques: bounding volumes, spatial subdivision, ray classification

Augmented techniques: macro regions, pyramid clipping, proximity clouds, directed safe zones

Special tricks: ray boxing, mailbox, handling CSG primitives, other types of coherence

Ray Shooting 7

RSA Techniques Classification

A) Subdivision techniques (top down)
- -- binary space partitioning (kd-trees)
- -- octrees
- -- uniform and hierarchical grids
- -- bounding volume hierarchy

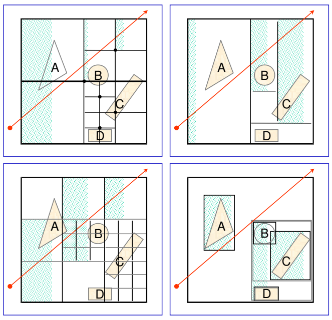
B) Clustering (bottom up)
- -- bounding volume hierarchy

Ray Shooting 8

Some RSA Techniques

octree

kd-tree

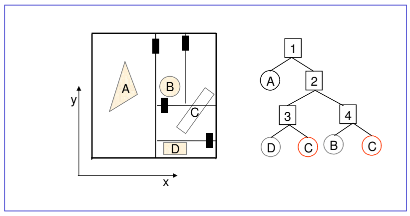


hierarchy of grids

bounding volume hierarchy

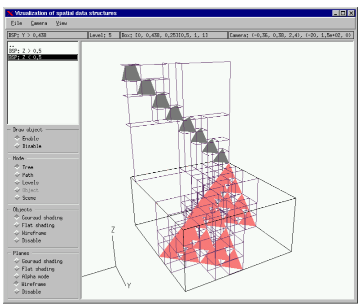
Ray Shooting 9

Kd-tree Construction



Ray Shooting 10

Visualisation of Kd-tree

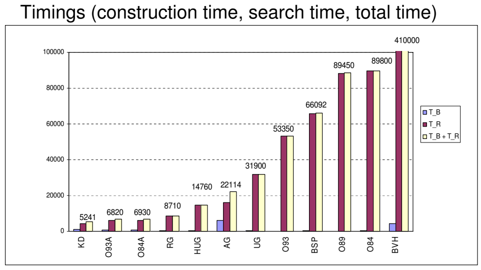


Ray Shooting 11

RSA Techniques Comparison

30 scenes *times* 12 RSAs *times* 4 ray distribution methods
= 1440 measurements, year 2000-2001

Timings (construction time, search time, total time)

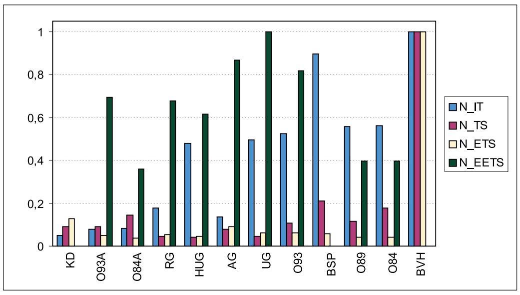


Note: In test BVH constructed in bottom-up fashion !

Ray Shooting 12

RSA Techniques Comparison

Number of operations (ray-object intersection, traversal steps)



Note: values normalized to the worst value.

Ray Shooting 13

RSA based on Kd-tree

Quite an efficient solution used in practice

Construction (in O(N log N) time)

- based on cost function and geometric probability
- automatic termination criteria algorithm
- various efficiency improvements:
  - construction of kd-tree with empty spatial regions
  - reducing objects' axis-aligned bounding boxes
  - preferred ray sets

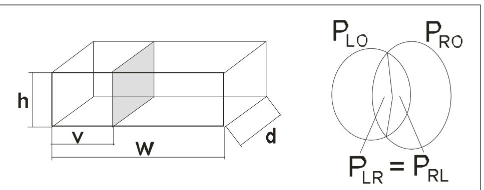
Ray traversal

– in practice achieves expected O(log N) time
– robust recursive ray traversal algorithm

Ray Shooting 14

Geometric Probability of Ray Intersecting a Subdivided Box

$probability_{LEFT} = P_{LO} + P_{LR} + P_{RL}$
$probability_{RIGHT} = P_{RO} + P_{LR} + P_{RL}$

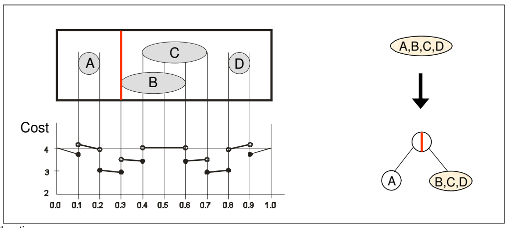


Condition: uniform ray distribution

Ray Shooting 15

Kd-tree Construction Based on Cost Function with Greedy Heuristics

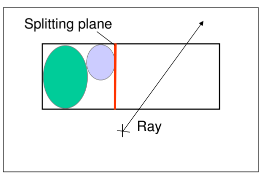
Cost = $probability_{LEFT}$ * $cost\_subtree_{LEFT}$ + $probability_{RIGHT}$ * $cost\_objects_{RIGHT}$



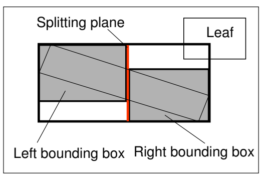
Ray Shooting 16

Kd-tree Efficiency Improvements

Cutting off empty space



Reducing objects' axis-aligned bounding boxes



Ray Shooting 17

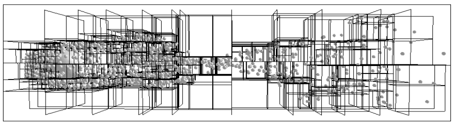
Termination Criteria for Construction

- *Local: using a stack*
  - *Simple local*: maximum depth + number of objects
  - *More complicated local*: a maximum number of cost improvement failures + maximum estimated depth + number of objects
- *Global: using a priority queue*
  - maximum memory used
  - maximum memory used + maximum leaf cost

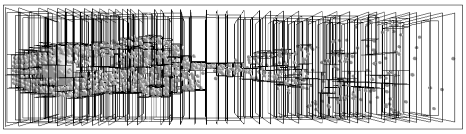
Ray Shooting 18

Kd-tree Construction for Preferred Ray Sets

Idea: using different than uniform distribution of rays



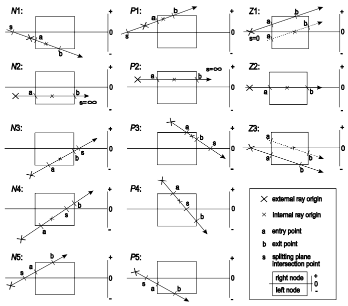
Uniform



Preferred

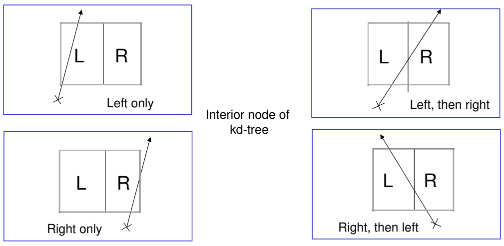
Ray Shooting 19

Recursive Ray Traversal Algorithm for Kd-tree



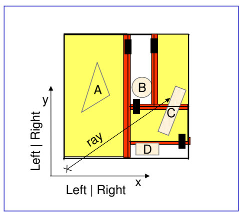
Ray Shooting 20

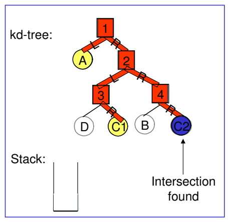
Recursive Ray Traversal Basic Cases Classification



Ray Shooting 21

Recursive Ray Traversal Algorithm

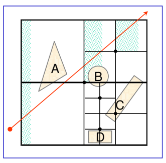




Ray Shooting 22

Ray Shooting with Octrees

- Interior node arity is eight
- Up to four child nodes can be traversed in an interior node
- Traversal algorithm necessarily more complicated than for kd-tree
- About 26 papers about ray tracing with octrees were published
- The octree is less adaptive to the scene geometry than kd-tree
- Geometric probability can be used in the same way as for kd-trees (Octree-R)
- According to the experiments, octrees are less efficient than kd-trees

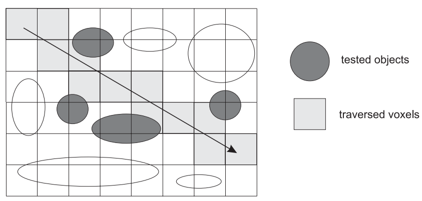


Note: octrees can be simulated by kd-trees

Ray Shooting 23

Ray Shooting with Uniform Grids

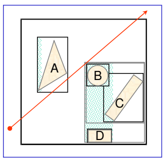
- Arity of a node proportional to the number of objects
- Special traversal method based on 3DDA
- For skewed distributions of objects in the scene is slow
- For highly uniform distributions of objects it is slightly faster than kd-trees



Ray Shooting 24

Ray Shooting with Bounding Volume Hierarchy (BVH)

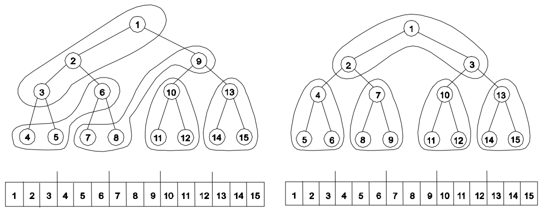
- Each interior node is fully described by a bounding box
- The number of child nodes is usually two for top-down construction (more for bottom-up construction)
- The nodes can overlap
- Each object is referenced only once
- The memory consumption is higher
- Traversal algorithm similar to kd-trees
- Kd-trees can be emulated by BVHs. The other way round is also feasible.



Ray Shooting 25

Data Layout in Memory

Inorder, preorder (depth-first-search), heap (bread-first-search), van Emde Boas



Ray Shooting 26

Performance Model of Ray Shooting

Total cost for RSA =
cost for ray-object intersection tests +
cost for ray traversal of kd-tree +
cost for data move from memory to CPU

- Faster ray-object intersection tests
- Decreasing number of ray-object intersection tests
- Faster traversal step
- Decreasing number of traversal steps
- Reducing CPU-memory traffic

Ray Shooting 27

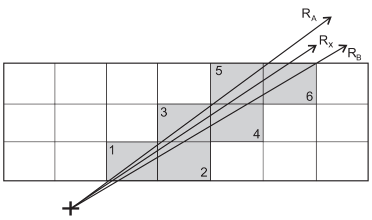
Offline Ray Shooting

- Shooting more rays at once
- Rays are formed by camera, by viewing frustum or by point light sources
- Rays are coherent = similar in direction and origin
- Problem can be formulated as offline setting of searching
- We can amortize the cost of traversal operations though the data structure … the number of traversal steps is decreased typically by 60-70%

Ray Shooting 28

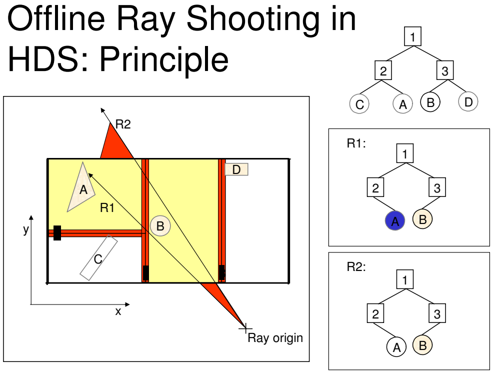
Offline Ray Shooting: Coherence

- If boundary rays traverse the same sequence S of leaves, then all rays in between also traverse the same sequence.
- Proof by convexity

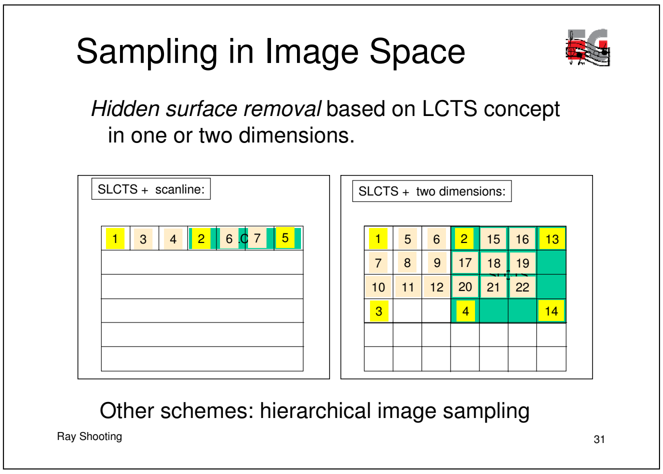


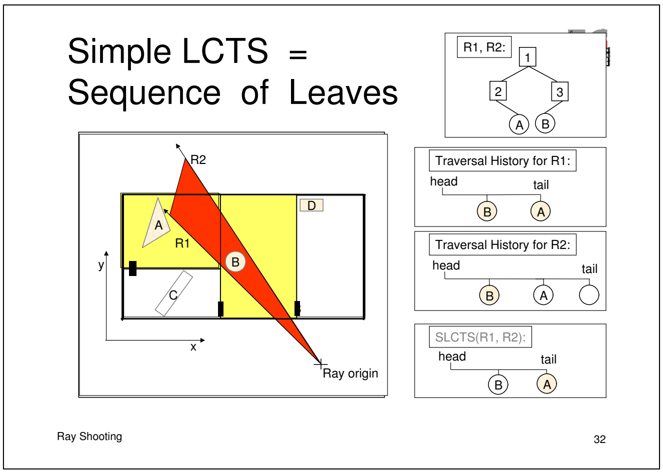
Ray Shooting 29

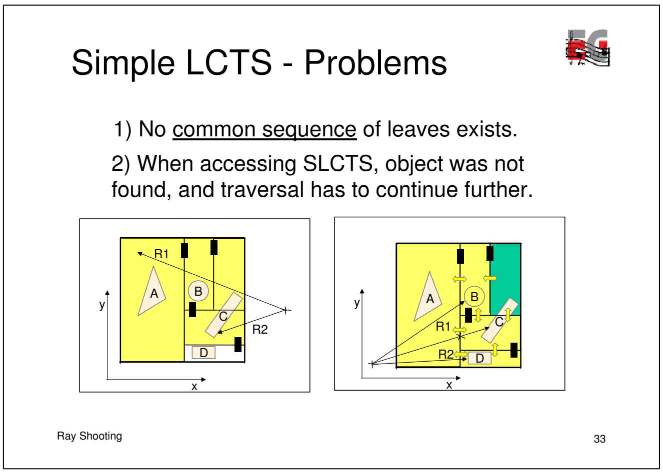
Offline Ray Shooting in HDS: Principle

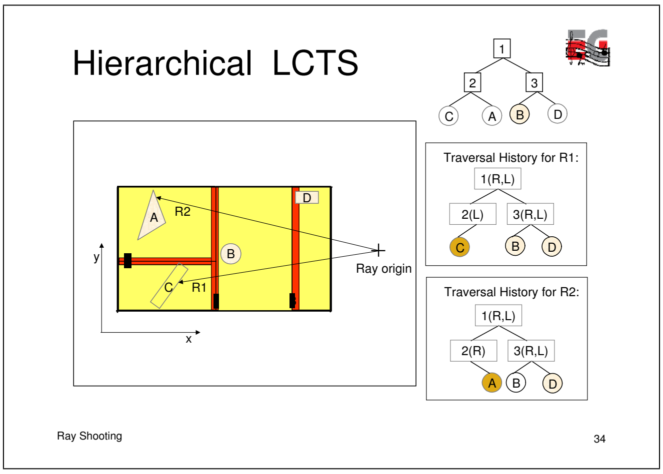


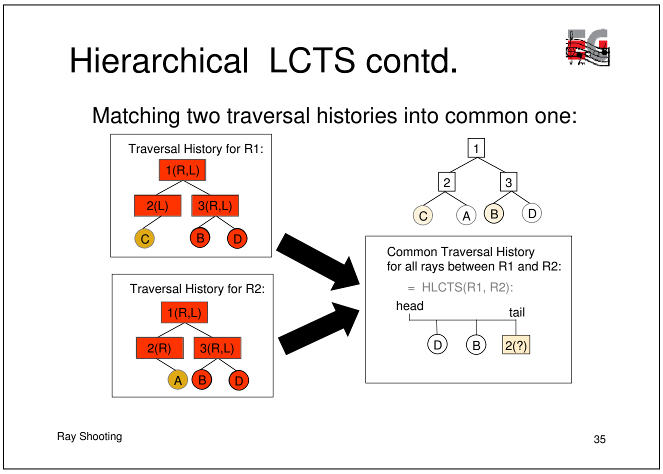
Ray Shooting 30

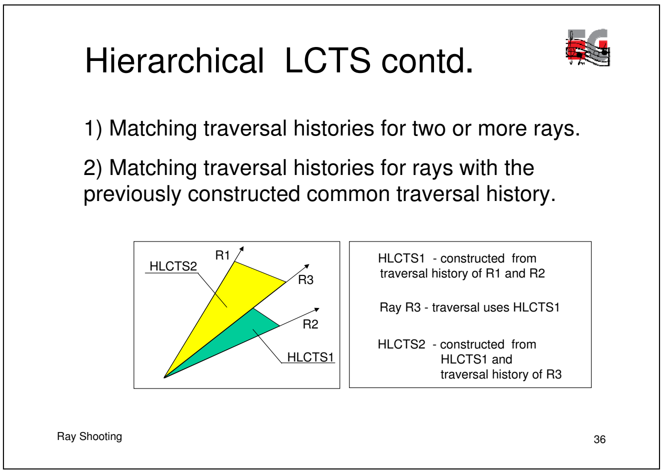

Sampling in Image Space
Hidden surface removal based on LCTS concept in one or two dimensions.
SLCTS + scanline:
1 3 4 2 6 C 7 5
SLCTS + two dimensions:
1 5 6 2 15 16 13
7 8 9 17 18 19
10 11 12 20 21 22
3 4 14
Other schemes: hierarchical image sampling
Ray Shooting
31


Simple LCTS = Sequence of Leaves
R1, R2:
1
2
3
A
B
R2
D
A
R1
B
C
y
x
Ray origin
Traversal History for R1:
head
tail
B
A
Traversal History for R2:
head
tail
B
A
SLCTS(R1, R2):
head
tail
B
A
Ray Shooting
32


Simple LCTS - Problems
1) No common sequence of leaves exists.
2) When accessing SLCTS, object was not found, and traversal has to continue further.
R1
A
B
C
R2
D
y
x
A
B
C
R1
R2
D
y
x
Ray Shooting
33

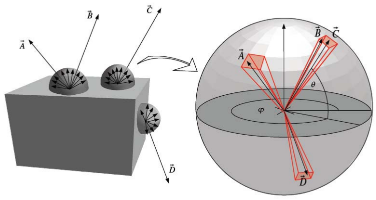

Hierarchical LCTS
1
2
3
C
A
B
D
A
R2
D
B
C
R1
y
x
Ray origin
Traversal History for R1:
1(R,L)
2(L)
3(R,L)
C
B
D
Traversal History for R2:
1(R,L)
2(R)
3(R,L)
A
B
D
Ray Shooting
34


Hierarchical LCTS contd.
Matching two traversal histories into common one:
Traversal History for R1:
1(R,L)
2(L)
3(R,L)
C
B
D
Traversal History for R2:
1(R,L)
2(R)
3(R,L)
A
B
D
1
2
3
C
A
B
D
Common Traversal History for all rays between R1 and R2:
= HLCTS(R1, R2):
head
tail
D
B
2(?)
Ray Shooting
35


Hierarchical LCTS contd.
1) Matching traversal histories for two or more rays.
2) Matching traversal histories for rays with the previously constructed common traversal history.
HLCTS2
R1
R3
R2
HLCTS1
HLCTS1 - constructed from traversal history of R1 and R2
Ray R3 - traversal uses HLCTS1
HLCTS2 - constructed from HLCTS1 and traversal history of R3
Ray Shooting
36

Ray Cache in Final Gathering

- Store the rays into cache according to direction
- When a bucket is filled in, shoot all of them at once
- Improves access pattern for incoherent queries
- Speedup up to 30%



Ray Shooting 37

Surveys on Ray Shooting and Ray Tracing

- G. Simiakakis: Accelerating Ray Tracing with Directional Subdivision and Parallel Processing, 1995
- V. Havran: Heuristics Ray Shooting Algorithms, 2001
- I. Wald: Real Time Ray Tracing and Global Illumination, 2004
- A. Y-H. Chang: Theoretical and Experimental Aspects of Ray Shooting, 2005

Ray Shooting 38

Questions and Answers for Part One

Ray Shooting 39

Part 2 - Content

- Hidden Surface Removal
- Visibility Culling
- Photon Maps and Irradiance Caching
- Ray Maps
- Other Algorithms
- Questions and Answers (10 minutes)

1

Hidden Surface Removal

Jiří Bittner

Czech Technical University in Prague
Vienna University of Technology

2

Hidden Surface Removal

- List priority algorithms
- Area subdivision algorithms
- Scan-line
- Z-buffer
- Ray casting

Hidden Surface Removal 3

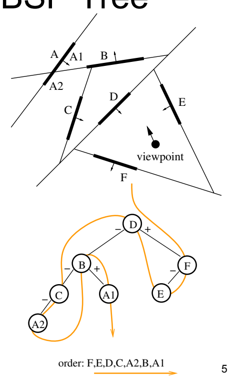
Depth Sort

- Draw faces back to front [Newell72]
- Overwrite the farther ones (painter's alg.)
- Determine strict depth order
  - Resolve cycles of overlapping polygons
- Step 1: depth sort (Z)
  - Quick sort, bubble-sort (temporal coherence)
- Step 2: rasterization (YX)
  - Bucket sort

Hidden Surface Removal 4

Depth Sort with BSP Tree

- BSP build in preprocess
  - Select a plane
  - Partition the polygons in front/back fragments
  - If >1 polygon → recurse
- Quick-sort like
- Heuristics for splitting plane selection



Hidden Surface Removal 5

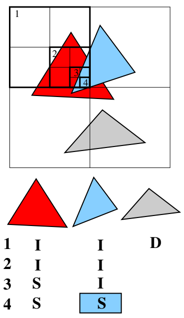
Depth Sort with BSP Tree

- Tree size: $O(n^2)$
- BSP need not be autopartition!
- For manifolds depth order can be predetermined → coarser BSP
- Generalization to all BSP nodes 'Feudal priority tree' [Chen96]

Hidden Surface Removal 6

Area Subdivision

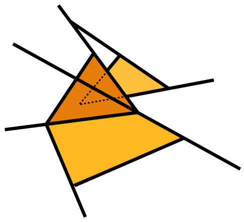
- Subdivide screen space [Warnock69]
- Classify polygons with respect to the area
- Terminate if trivial solution
- Step 1: octree subdivision (XY)
  - Quick sort like
- Step 2: list for octree nodes (Z)
  - Insertion sort



Hidden Surface Removal 7

Naylor’s BSP projection

- Draw polygons front to back
- Clip polygons by 2D BSP of projected polygons
- Step 1: depth sort (Z)
  - 3D BSP built in preprocess
- Step 2: 2D BSP (XY)
  - Quick sort like subdivision of the projection plane



Hidden Surface Removal 8

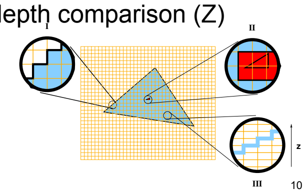
Scan-Line

- Sort by scan-lines (Y)
- Sort spans within a scanline (X)
- Search for closest span (Z)
- [Watkins70]
  - Bubble sort in X and Y
  - O(log n) search in Z

Hidden Surface Removal 9

Z-buffer

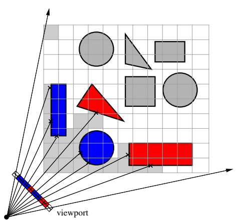
- Rasterize polygons in arbitrary order
- Maintain per pixel depths
- Step 1: rasterization (YZ)
  - Bucket sort
- Step 2: per pixel depth comparison (Z)
  - Min selection



Hidden Surface Removal 10

Ray Casting

- Cast ray for each pixel
- Step 1: spatial data structure (XYZ)
  - Preprocess
  - Trees ~ quick sort
  - Grid ~ distribution sort
- Step 2: search for nearest intersection
  - Min selection with early termination



Hidden Surface Removal 11

Z-buffer vs. Ray Casting

	scan-line coherence	output sensitive	presorting
Z-buffer	+	-	+ (no)
Ray casting	-	+	- (yes)

- Z-buffer better in simple sparsely occluded dynamic scenes
- Ray casting better in complex densely occluded static scenes

Hidden Surface Removal 12

Summary

- HSR algorithms sort in
  - Directions (XY)
  - Depth (Z)
  - Differ in sorting order and methods [SSS74]
- Current winners: z-buffer, ray casting

Hidden Surface Removal 13

Visibility Culling

Jiří Bittner

Czech Technical University in Prague
Vienna University of Technology

1

Visibility Culling - Introduction

- Main goal: reduce linear complexity of z-buffer
  - Render only potentially visible polygons
- Online
  - Applied for every view point at runtime
- Offline
  - Partition view space into view cells
  - Compute Potentially Visible Sets (PVS)

Visibility Culling 2

Visibility Culling – Motivation

- Q: Why visibility culling, when:
  - Object outside screen culled by HW clipping
  - Occluded objects culled by z-buffer
- A: Linear complexity not sufficient!
  - Processing too many invisible polygons
- Goal
  - Render only what can be seen!
  - Make z-buffer output sensitive

Visibility Culling 3

Online Visibility Culling

- For every frame cull whole groups of invisible polygons
- Conservative solution
  - Trading accuracy for speed
  - Leaves a superset of visible polygons
  - Precise visibility solved by z-buffer

Visibility Culling 4

Online Visibility Culling

- Backface culling
- View-frustum culling
- Occlusion culling
  - CPU techniques
  - GPU based (HW occlussion queries)

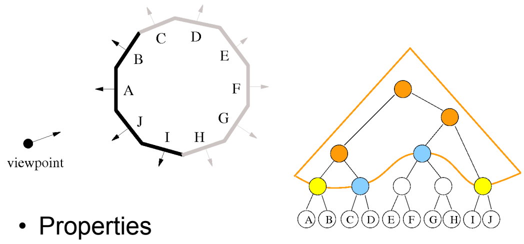
Visibility Culling 5

Backface Culling

- Culls about 50% polygons
- Supported by the GPU
- Alternative: Hierarchical back-face culling [Kummar96]
  - Sort polygons based on their normals into a tree
  - Skip whole groups of backfacing polygons

Visibility Culling 6

Hierarchical Backface Culling



- Properties
  + Skips whole groups of polygons
  - Regroups the scene (discards objects)
  - Limited gain

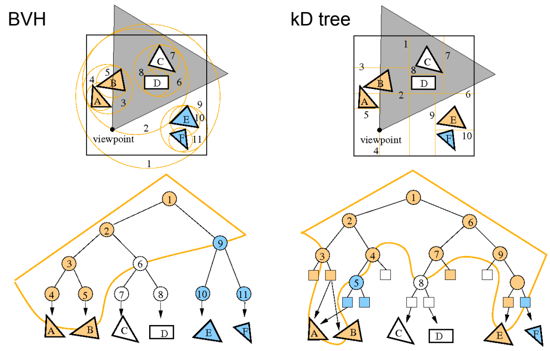
Visibility Culling 7

View Frustum Culling

- Culls 0-100% polygons
- Conservative algorithm
  – Spatial hierarchy: kD-tree, BSP tree, octree, BVH
  – Intersection test with the view frustum
- Optimizations
  – Temporal coherence
  – Efficient intersection test [Assarson00]

Visibility Culling 8

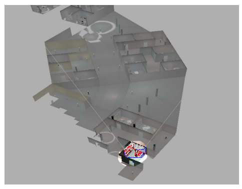
View Frustum Culling - Example



Visibility Culling 9

Occlusion Culling

- Previous methods disregard occlusion
- 99% of scene can be occluded!

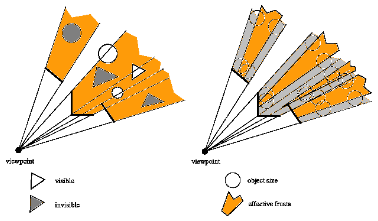


- Solution: Detect and cull also occluded objects

Visibility Culling 10

Shadow Frusta

- Construct shadow frusta for several occluders [Hudson97]
- Object is invisible if inside a shadow frustum
- Queries on the spatial hierarchy



Visibility Culling 11

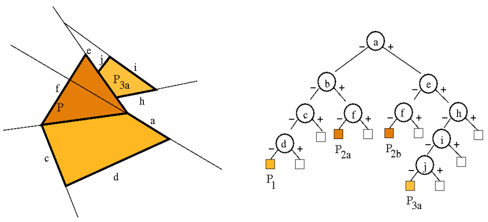
Shadow Frusta - Properties

- Properties
  + Easy implementation
  - No occluder fusion!
  - No occluder sorting - O(n) query time
  - Small number of occluders (~10)

Visibility Culling 12

Occlusion Trees

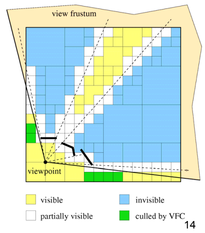
- Occluders sorted into a 2D BSP tree [Bitt98]
- The tree represents fused occlusion
- Example: occlusion tree for 3 occluders



Visibility Culling 13

Occlusion Tree - Traversal

- Visibility test of a node
  - Depth-first-search
  - Found empty leaf → tested object is visible
  - Depth test in filled leaves
- Example of final visibility classification of kD-tree



Visibility Culling 14

Occlusion Tree - Properties

- Presorting occluders
  - Tree size: worst case $O(n^2)$
  - O(log n) visibility test

\+ Allows to use more occluders (~100)

\- Not usable for scenes with small polygons

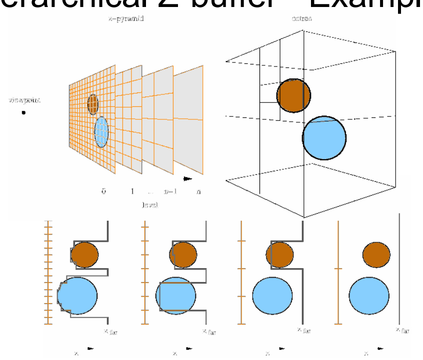
Visibility Culling 15

Hierarchical Z-buffer

- Extension of z-buffer to quickly cull larger objects [Greene 96]
- Ideas
  - octree for spatial scene sorting
  - z-pyramid for accelerated depth test

Visibility Culling 16

Hierarchical Z-buffer - Example



Visibility Culling 17

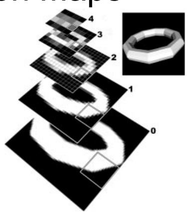
Hierarchical Z-buffer - Usage

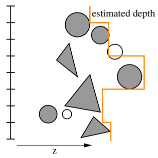
- Hierarchical test for octree nodes
- Find smallest node of z-pyramid, which contains the tested box
- Box depth > node depth → cull
- Otherwise: recurse to lower z-pyramid level
- Optimization: use temporal coherence
  - z-pyramid constructed from polygons visible in the last frame

Visibility Culling 18

Hierarchal Occlusion Maps

- Hierarchical occlusion maps [Zhang97]
- Pyramid of occlusion maps
- Separate occlusion and depth representation
  - Hierarchical occlusion
  - Coarse depth
- Queries on spatial hierarchy





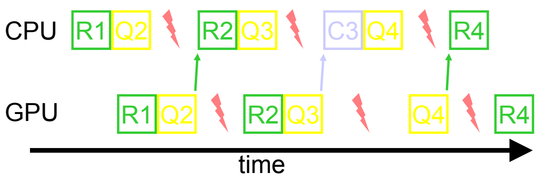
Visibility Culling 19

HW Occlusion Queries

- ARB_occlusion_query, NV_occlusion_query
- Return #pixels passing the depth test
- Feature which has been missing!
- Issues
  1. Latency – the result not readily available
  2. The query costs time

Visibility Culling 20

CPU Stalls GPU Starvation

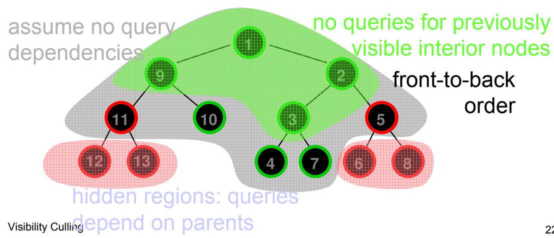


Rx Render object x
Qx Query object x
Cx Cull object x
Waiting time

Visibility Culling 21

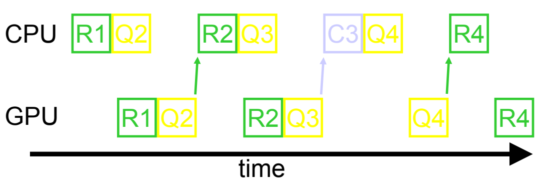
Coherent Hierarchical Culling

- CHC [Bitt04]
  - Interleave queries and rendering
  - Schedule queries based on temporal coherence



Visibility Culling 22

CHC



Rx Render object x
Qx Query object x
Cx Cull object x

Visibility Culling 23

CHC

- Video
- UNC power plant, 12.7M polygons

Visibility Culling 24

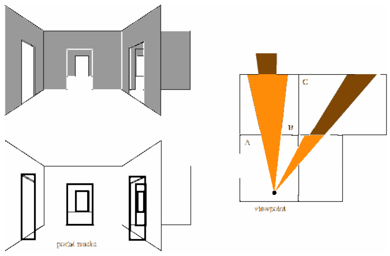
Cells and Portals

- Partition the scene in cells and portals
  - Cells ~rooms
  - Portals ~ doors&windows
- Cell adjacency graph
- Constrained DFS
  - Portal visibility test [Luebke 96]

Visibility Culling 25

Portal Visibility Test

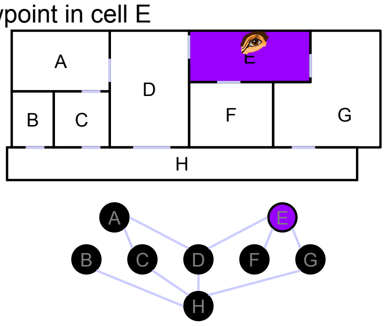
- Intersection of bounding rectangles of portals



Visibility Culling 26

Cells and Portals Example

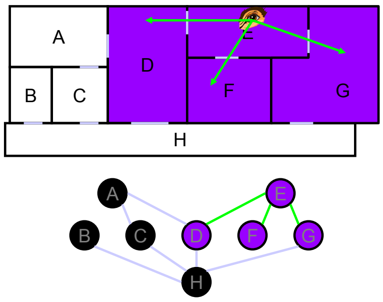
- Viewpoint in cell E



Visibility Culling 27

Cells and Portals - Example

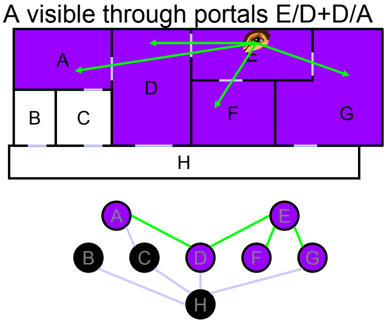
- Adjacent cells DFG



Visibility Culling 28

Cells and Portals - Example

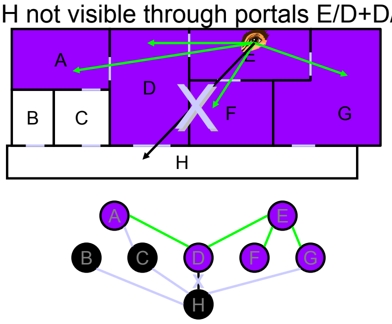
- Cell A visible through portals E/D+D/A



Visibility Culling 29

Cells and Portals - Example

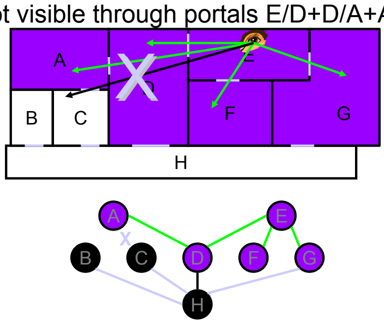
- Cell H not visible through portals E/D+D/H



Visibility Culling 30

Cells and Portals - Example

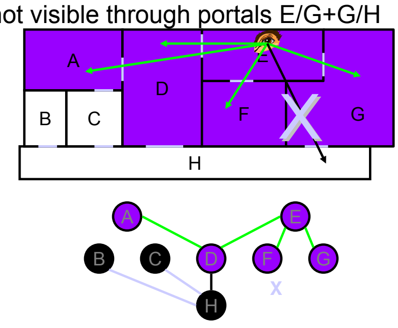
- C not visible through portals E/D+D/A+A/C



Visibility Culling 31

Cells and Portals - Example

- H not visible through portals E/G+G/H



Visibility Culling 32

Visibility Preprocessing

- Preprocessing
  - Subdivide view space into view cells
  - Compute Potentially Visible Sets (PVS)
  - Solves visibility “offline” for all possible view points
- Usage
  1. Find the view cell (point location)
  2. Render the associated PVS

Visibility Culling 33

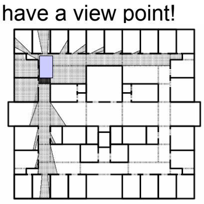
Visibility Preprocessing

- Other benefits
  - Prefetching for out-of-core/network walkthroughs
  - Communication in multi-user environments
- Problems
  - Costly (treats all view points and view directions)
  - 4D domain (online methods only 2D)

Visibility Culling 34

Interiors – Cells and Portals

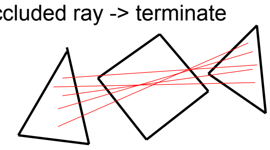
- Subdivide the scene into cells and portals
- Constrained DFS on the adjacency graph
  - Portal visibility test
- More complex than the online algorithm
  - We do not have a view point!



Visibility Culling 35

Interiors – Cells and Portals

- Sampling [Airey90]
  - Random rays
  - Non-occluded ray -> terminate

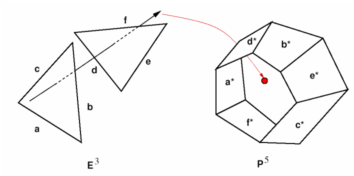


+ Simple implementation
- Approximate solution

Visibility Culling 36

Interiors – Cells and Portals

- Exact computation [Teller 92]
  - Mapping to 5D (Plücker coordinates of lines)
- Portal edges → hyperplanes H_i in 5D
- Halfspace intersection in 5D



Visibility Culling 37

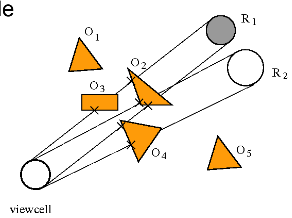
General Scenes - Strong Occlusion

- Occlusion by single object [CohenOr98]
- For each cell and object
  - Cast rays defining convex hull of the cell and object
  - If a convex occluder intersects all rays → invisible

Visibility Culling 38

General Scenes - Strong Occlusion

- Example



- Properties
  - +Simple
  - - No occluder fusion
  - - Too conservative for larger view cells and small objects

Visibility Culling 39

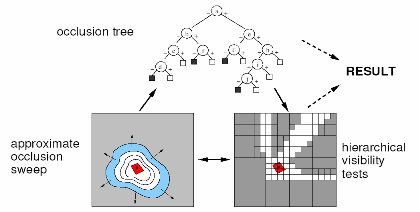
General Scenes
Occlusion Tree

- Extension of the 2D occlusion tree
- 5D BSP tree
  - Plücker coordinates of lines
- The tree represents union of all occluded rays for a given view cell

Visibility Culling 40

General Scenes
Occlusion Tree

- Process polygons in front-to-back order
- Polygon visible → enlarge tree by visible rays
- Polygon invisible → tree not modified



Visibility Culling 41

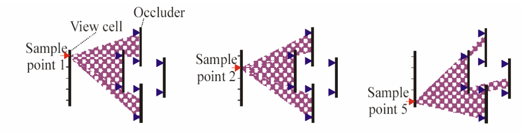
General Scenes
Occlusion Tree

- Properties
  - + Exact solution
  - + Uses visibility coherence
  - - Difficult implementation

Visibility Culling 42

2.5D Scenes Occluder Shadows

- Footprint of occluded volume [Wonka00]
  - Agrregates the shadow polygons using z-buffer
  - Represents intersection of all 'shadows'



Visibility Culling 43

2.5D Scenes Occluder Shadows

- Conservative solution
  - Shrinking occluder polygons
- Properties
  - + Relatively easy implementation
  - + Uses GPU
  - - Large view cells → more conservative solution
  - - Needs high resolution cull map

Visibility Culling 44

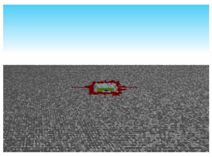
2.5D Scenes Ray Space Factorization

- Main ideas [Leyvand et al. 2003]
  - Occluder in 2.5D → 3D polygon in ray space
  - Polygon shape: defined by 2D footprint
  - Polygon depth: defined by heights

Visibility Culling 45

2.5D Scenes Ray Space Factorization

- Render polygons using z-buffer
- Visible polygons in ray space → visible objects in primal space

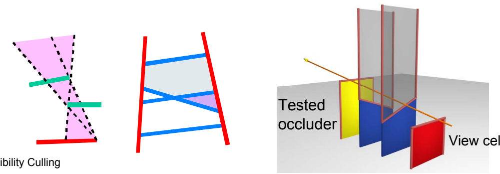



- Properties
  - - Conservative solution
  - + GPU implementation

Visibility Culling 46

2.5D Scenes Occlusion Tree + Virtual Portals

- Occlusion tree for visibility in 2D footprint
- Identifies sequencies of occluders
- Construct virtual portals over the occluders
- Portal visibility test in 5D [Teller 92]



Visibility Culling 47

2.5D Scenes Occlusion Tree + Virtual Portals

- Properties
  - + exact solution for 2.5D scenes
  - + computation time comparable with conservative methods
  - - difficult implementation

Visibility Culling 48

Summary

- Online occlusion culling
  - Computation for every frame
  - No global information
  + Dynamic scenes
  + Easy implementation
- Visibility preprocessing
  + Gives global information
  + Almost no computation at runtime
  - Static scenes
  - Significant preprocessing time
  - Difficult implementation

Visibility Culling 49

Surveys on Visibility

- F. Durand. 3D Visibility: Analytical Study and Applications
- D. Cohen-Or et al.: A survey of visibility for walkthrough applications
- J. Bittner and P. Wonka: Visibility in computer graphics

Visibility Culling 50

Photon Maps and Irradiance Caching

Vlastimil Havran

Czech Technical University in Prague

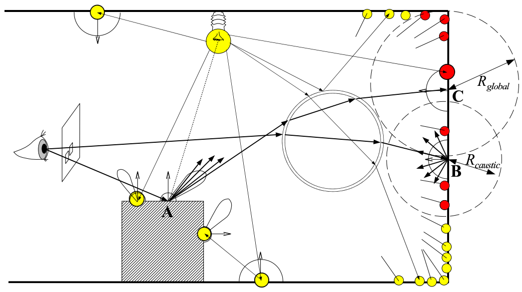
1

Photon Maps and Irradiance Caching

- Final gathering versus direct visualization
- Photon maps
- Irradiance caching
- Offline techniques

Photon Maps and Irradiance Caching 2

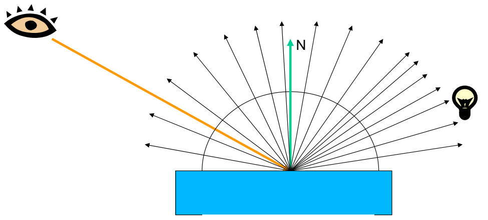
Photon Mapping



Photon Maps and Irradiance Caching 3

Final Gathering

- Shooting many secondary rays (possibly according to BRDF), gathering radiances from the rays
- Integrating the radiances properly to render image

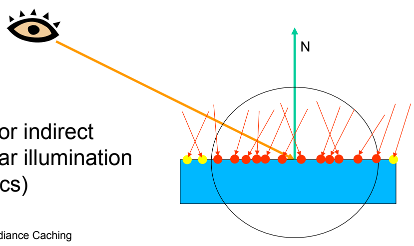


- Used for indirect diffuse illumination

Photon Maps and Irradiance Caching 4

Direct Visualization

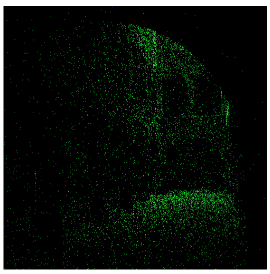
- Do not shoot final gather rays, use directly visible photons from camera
- It is prone to artifacts on object boundaries referred to as bias



- Used for indirect specular illumination (caustics)

Photon Maps and Irradiance Caching 5

Example of Direct Visualization



Photon Hits

Direct Visualization

Photon Maps and Irradiance Caching 6

Estimating Radiance along Final Gather Ray

- Using the density estimation, from the photon hits estimating *PDF*
- It requires K nearest neighbor searching for each final gather ray
- The number of final gather rays (the number of searches) is enormous
- Typically we shoot 200-4000 final gather rays per pixel
- The number of pixels per image 1-6 x 10^6

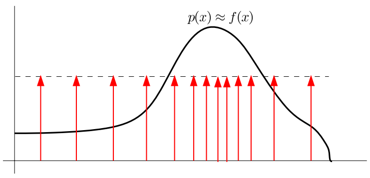
Photon Maps and Irradiance Caching 7

Intro to Density Estimation

- Histogram method – take hits into buckets
- Kernel density estimation
- K-Nearest neighbors estimator
- Variable kernel density estimator
- Multiple pass methods
  - First pass – pilot estimate
  - Second pass – final estimate

Photon Maps and Irradiance Caching 8

Example: Density Estimation in1D

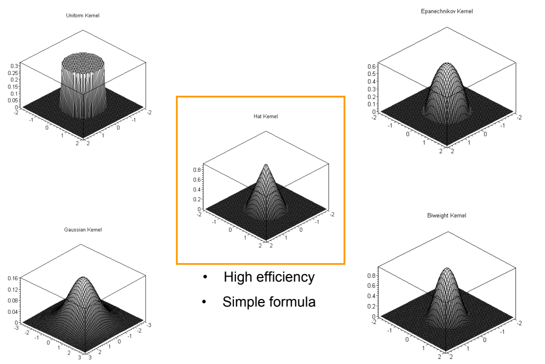


Note: **Importance Sampling**: from given *p(x)* to samples

Density Estimation: from samples to *p(x)* … more complicated

Photon Maps and Irradiance Caching 9

Kernel Types



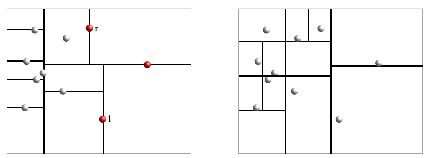
Relation to Searching

- *Range search* – given a fixed range query (sphere, ellipsoid), find all the photons in the range
- *K nearest neighbor search* – given a center of the expanding shape X (sphere, ellipsoid), find K nearest photons
  - Without considering the direction of incoming photons
  - With considering only valid photons with respect to the normal at point X

Photon Maps and Irradiance Caching 11

Search Techniques

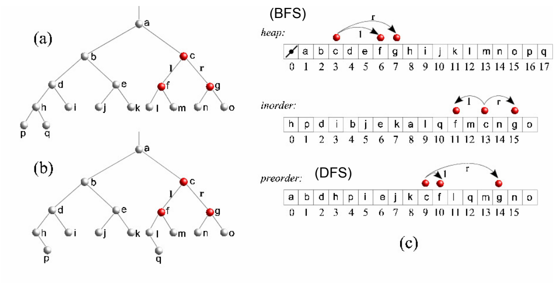
- Use any data structures described in the section "*Hierarchical Data Structures*"
- Typically kd-trees or kd-B-trees are used



Kd-tree Kd-B-tree

Photon Maps and Irradiance Caching 12

KD-tree Layout in Memory



+ van Emde Boas layout (see ray tracing)

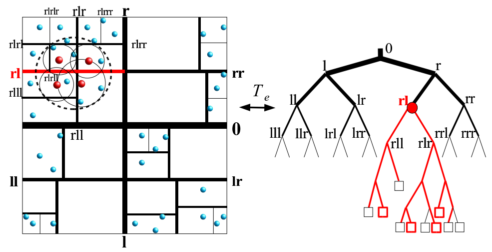
Photon Maps and Irradiance Caching 13

Practical yet Efficient Solution

- Use Kd-B-trees
- Construct a tree over an array of photons
- Use 8 Bytes nodes – good packing
- DFS or van Emde Boas Layout
- Sliding mid-point rule = spatial median + shift to nearest photon if one side empty
- One leaf contains a range of 30-70 photons (two indices to photon array)
- Properties:
  - fast construction time
  - fast search (complexity proved to be optimal)

Photon Maps and Irradiance Caching 14

Aggregate Searching (= Offline Search)



- splitting planes of the spatial kd-tree
- first intersected plane <=> start node
- density estimation point (ep)
- photon hit point
- leaf node containing ep
- searched leaf
- start node for individual queries
- l left child , r right child

Photon Maps and Irradiance Caching 15

Searching Tricks for k-NN Search

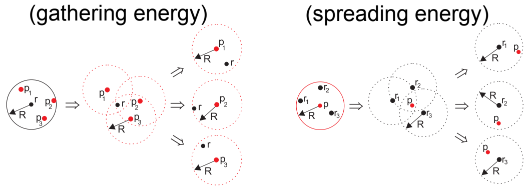
- Do not use uniform grids, they do not work efficiently for skewed distributions
- Try to avoid a priority queue
- Use a fixed radius search where a radius is estimated for given N
- Radius estimate can be based on the properties of the data structure over photons (diagonal of a leaf box) or taken from already computed query
- Use offline search if possible
- Try to change the role of input data to be queried and queries

Photon Maps and Irradiance Caching 16

Reverse Photon Mapping

Normal Photon Mapping (gathering energy)

Reverse Photon Mapping (spreading energy)



r – ends of final gather rays (in black)

p – photons (in red)

Photon Maps and Irradiance Caching 17

Why Does It Work Faster ?

Assume that the number of interactions among photons and final gather rays is the same !

Traditional Photon Mapping – a single tree

- Many searches (~10^9) in a small tree over photons (~10^6)
- kNN search based on the photon density

Reverse Photon Mapping – more involved *(two trees)*

- Smaller number of searches (~10^6) in a larger tree over the ends of final gather rays (~up to 10^9)
- k-NN search is also based on the photon density

Properties

- Search in a tree is logarithmic, reverse photon mapping then faster
- Reverse photon mapping takes more memory

Photon Maps and Irradiance Caching 18

Time Complexity Formulas

- F … number of final gather rays
- K … number of neighbors for kNN search
- V … number of photons
- F * K … number of interactions photon-final gather ray

Traditional Photon Mapping Time

C_PT = C_1 * F * K + C_2 * F * log V

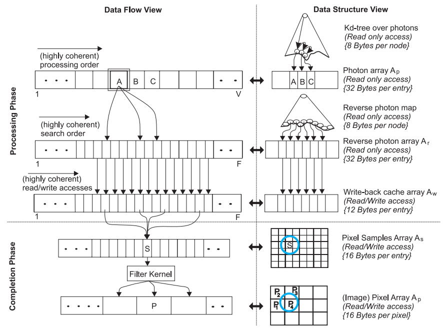
Reverse Photon Mapping Time

C_RPT = C_1 * F * K + C_2 * V * log F

For **F >> V** it is easy to show that **F * log V > V * log F**

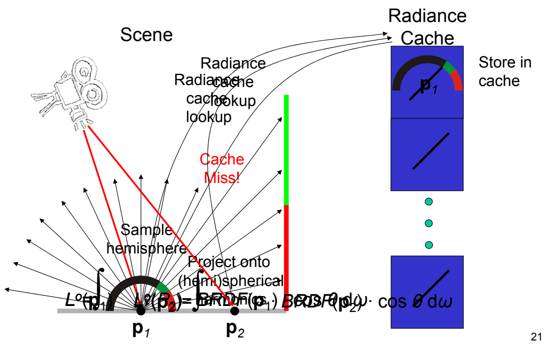
Photon Maps and Irradiance Caching 19

Data Flow + Data Structure View



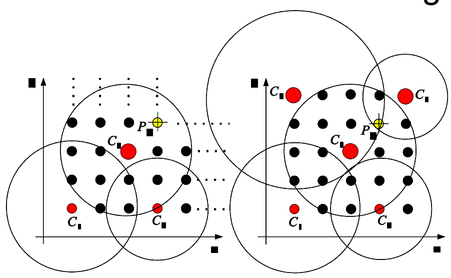
Photon Maps and Irradiance Caching 20

Radiance and Irradiance Caching



21

Data Structures for Caching



- Red – data in the cache
- Black – queries

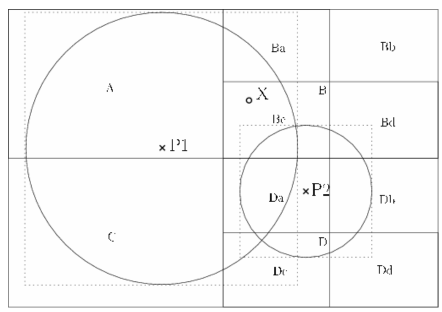
Photon Maps and Irradiance Caching 22

Search Specification

- Records – the irradiance specified by a point and radius of influence
- Query: given a point, find all the sphere in which the point is contained
- Problem is *intersection search*
- Data structures should be dynamic – insertion and deletion is possible

Photon Maps and Irradiance Caching 23

1) Using Octree (Ward et al. 88)



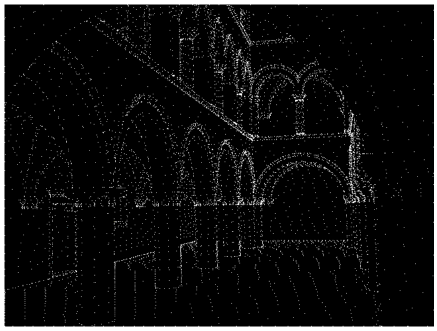
Photon Maps and Irradiance Caching 24

2) Using Mapping to R^4

- A sphere (a,b,c,r) in R^3 as a point in R^4 (t1,t2,t3,t4) by linearization:
  (2.a, 2.b, 2.c, a*a + b*b + c*c - r*r)
- Query: a point (a, b, c) in R^3 as a hyper-plane in R^4 (t1,t2,t3,t4) as follows:
  H: a*t1+b*t2+c*t3-t4 –(a*a+b*b+c*c) > 0
- Compute *half-space range reporting* in R^4 space, it requires a spatial data structure in R^4
- Efficiency depends highly on
  – Position of points with respect to the space origin
  – Efficiency of half-space range reporting

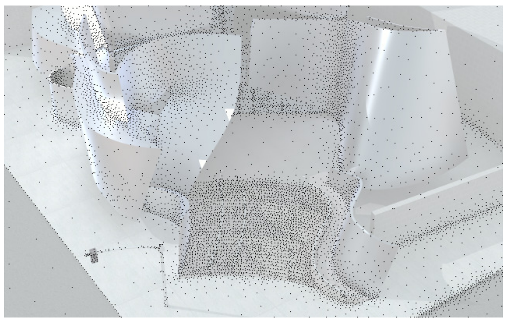
Photon Maps and Irradiance Caching 25

Irradiance Caching Records



Photon Maps and Irradiance Caching 26

Radiance Caching Records



Photon Maps and Irradiance Caching 27

Ray Maps

Jiří Bittner

Czech Technical University in Prague
Vienna University of Technology

1

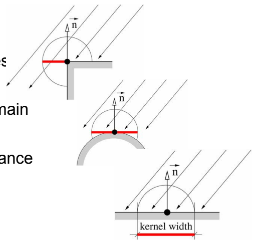
Ray Maps

- Ray map: data structure sorting rays
- Allows efficient searching for rays
  - intersecting a disc/sphere/hemisphere
  - nearest to a point using
  - combination of intersection & NN
- Main application:
  improved density estimation

Ray Maps 2

Density Estimation

- Problems with photon maps
  1. Limited to object surfaces
  2. Boundary bias – darkening on objects' boundaries
  3. Topological bias – assumption on shape of the domain
  4. Proximity bias – tradeoff between noise and variance



- Ray maps

  eliminate 1. and 2., possibly reduce 3.

Ray Maps 3

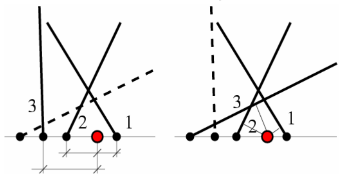
Ray Map Queries

- Queries types
  - intersection search (fixed query domain)
  - k-nearest neighbors (k-NN) search
- Query domains
  - disc
  - sphere
  - hemisphere
  - axis-aligned box
  - \+ possible limitation on ray directions

Ray Maps 4

Metrics for k-NN Search

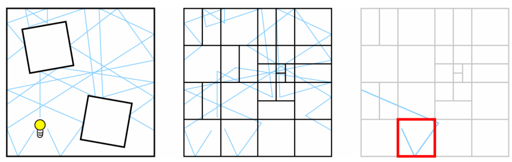
- Distance on the tangent plane
- Distance to the ray segment
- Distance to the supporting line of the ray



Ray Maps 5

Ray Map Implementation

- Kd-tree
- Leaves store references to the rays
- Lazy construction driven by the queries
- Support efficient searching and updating



Ray Maps 6

Construction

- Spatial median split
- Subdivide if #rays > budget
- Classify rays back, front, both
- Termination criteria
  - #ray references per leaf (~32)
  - size of the leaf (~0.1% of the scene box)
  - Max tree depth (~30)

Ray Maps 7

Searching

- Intersection search
  - Locate all leaves containing query domain
  - Gather rays
  - Compute intersections
- k-NN search
  - Priority queue (stack)
  - Locate the leaf containing the query origin
  - If #rays < N get next node from stack

Ray Maps 8

Maintenance

- Deleting a ray
  - Ray cast and remove references
- Adding a ray
  - Ray cast and subdivide if required
- Keeping memory budget
  - Collapsing of unused subtrees nodes
  - LRU strategy

Ray Maps 9

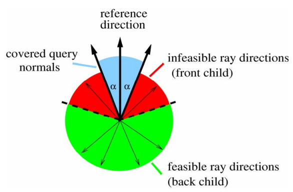
Optimization 1
Exploiting Query Coherence

- Subsequent queries highly coherent
- Store traversed nodes of the previous query
- Initiate the priority queue with the saved nodes
- Top-down traversal is reduced

Ray Maps 10

Optimization 2
Directional Splits

- Queries are oriented
- Many rays in the opposite direction after reflection
- Optimization: inserting directional nodes



Ray Maps 11

k-NN Search with Ray Maps

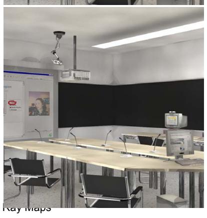
- 1M – 2.5M rays
- Typical memory usage: 16 – 128MB
- Query time: 0.2 – 1.5ms (3.2GHz PC)
- 2.1 to 4.7 times slower than photon maps

Ray Maps 12

Comparison with Photon Maps

Photon maps

Ray maps



Photon maps + convex hull

1,000,000 photons, 500 kNN

Similar Data Structures

- Ray cache [Lastra02]
  - Hierarchy of spheres
- Volumetric ray density estimation [VanHaevre04]
  - Octree
  - Simulation of plant growth

Ray Maps in Line Space

- Idea
  - Ray → 5D point (Plücker coordinates)
  - Query → 5D polyhedron
  - Report all points in the polyhedron
  - Use 5D kD-tree to sort points
- Poor performance
  - Culling only at very bottom of the tree
  - 5D boxes not separate well from the query polyhedron

Ray Maps - Summary

- Sorting rays + efficient searching
- Kd-trees implementation
  - Simple implementation
  - Efficient memory usage control
  - About 2-5x slower than photon maps
- Density estimation
  - New query domains + new metrics
  - Elimination of boundary bias
  - Reduction of topological bias

Other Algorithms

Vlastimil Havran

Czech Technical University in Prague

1

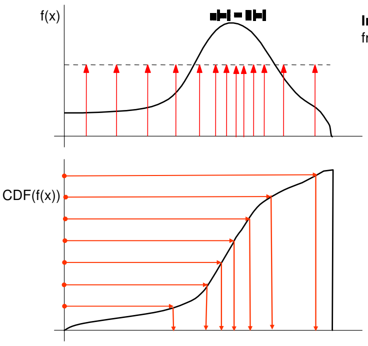
Other Algorithms

- Importance sampling
- Hierarchies over light sources
- Extensions to ray tracing
- Some other techniques

Note: this list on sorting and searching in rendering is definitely not complete !

Other Algorithms 2

Importance Sampling



Importance Sampling: from given *p(x)* to samples

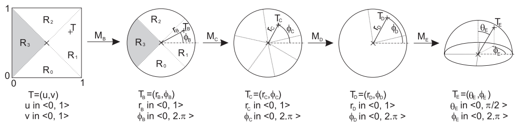
Step 1: compute cumulative distribution function into a table

Step 2: via uniform distribution over CDF generate required distribution f(x)

Other Algorithms 3

Importance Sampling Transforms

- *Results:* samples on the hemisphere (2D domain)
- *Usage:* for BRDF and environment maps
- *Realization:* using four mappings
- *Properties:* bijectivity, continuity in both directions, low distortion
- *Complexity of sampling:* O((log N) * (log M))



Other Algorithms *Havran et al. 03c: Goniometric Diagram Mapping for Hemisphere* 4

Hierarchies over Light Sources

- Another hierarchy (=sorting) if number of light sources is high, approximating or discarding less important light sources
- Papers:
  - Ward92: Adaptive Shadow Testing, discard less important contributions, avoid shadow rays testing
  - Lazanyi and Szirmay-Kalos 04: Speeding up the Virtual Light Sources Algorithm
  - Paquette et al. 98: A Light Hierarchy for Fast Rendering of Scenes with Many Lights
  - Walter et a. 05: Lightcuts: a scalable approach to illumination
  - Walter et al. 06: Multidimensional lightcuts

Other Algorithms 5

Extensions to Ray Tracing

- Spatio-temporal domain
  - Continuous setting (Glassner 88)
  - Multiframe ray tracing (discrete time setting) (Havran et al. 03b)
  - Reprojection for walkthroughs (Havran et al. 03a)
- Approximate ray tracing
  - Szirmay-Kalos et al.: Approximate Ray-Tracing on the GPU with Distance Impostors (2005)
- Fast construction or update for animations
  - Several algorithms proposed in 2005, not yet resolved issue

Other Algorithms 6

Some Other Techniques

- Temporal Photon Mapping and Spatio-Temporal Density Estimation
  - Cammarano and Jensen 02: Time Dependent Photon Mapping
  - Weber et al. 2004: Spatio-Temporal Photon Density Estimation Using Bilateral Filtering
- Reordering the queries for photon mapping
  - Havran et al. 05: Reverse Photon Mapping
  - Steinhurst et al. 05: Reordering for Cache Conscious Photon Mapping
- Changing the role of queries and input data to be queried
  - Havran et al. 05: Reverse Photon Mapping (here in the slides)
  - Laine and Aila 05: Hierarchical Penumbra Casting

Other Algorithms 7

Remainder and Question

- This list on the use of sorting and searching in rendering algorithms is definitely not complete!
- Are you convinced now that sorting and searching is really relevant to rendering?

Other Algorithms 8

Content

Part One

- Introduction to Rendering
- Sorting and Searching
- Hierarchical Data Structures
- Ray Shooting
- Questions and Answers

Other Algorithms 9

Content

Part Two

- Hidden Surface Removal
- Visibility Culling
- Photon Maps and Irradiance Caching
- Ray Maps
- Other Algorithms
- Questions and Answers

Other Algorithms 10

Questions and Answers for Part Two

Other Algorithms 11

References

This section contains selected publications on rendering which use and discuss (either directly or indirectly) sorting and/or searching algorithms. The list of references consists of several parts, which correspond to the topics discussed in tutorial.

Sorting and Searching, Hierarchical Data Structures

[Agar04] P. Agarwal. Range searching. In *CRC Handbook of Discrete and Computational Geometry (J. Goodman and J. O'Rourke, eds.)*, CRC Press, New York, 2004.

[Agar99] P. Agarwal and J. Erickson. Geometric range searching and its relatives. 1999.

[Bare96] G. Barequet, B. Chazelle, L. J. Guibas, J. S. B. Mitchell, and A. Tal. BOXTREE: A Hierarchical Representation for Surfaces in 3D. *Computer Graphics Forum*, Vol. 15, No. 3, pp. 387–396, 1996.

[Chan01] A. Y.-H. Chang. A Survey of Geometrical Data Structures for Ray Tracing. Tech. Rep. TR-CIS-2001-06, 2001.

[Chan04] A. Y.-H. Chang. *Theoretical and Experimental Aspects of Ray Shooting*. PhD thesis, Politechnic University, USA, 2004.

[Gaed98] V. Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, Vol. 30, No. 2, pp. 170–231, 1998.

[Guib98] L. Guibas. Kinetic data structures: A state of the art report. 1998.

[Gutt84] A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In B. Yormark, Ed., *SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, June 18-21, 1984*, pp. 47–57, ACM Press, 1984.

[Hjal03] G. R. Hjaltason and H. Samet. Index-driven similarity search in metric spaces. *ACM Trans. Database Syst.*, Vol. 28, No. 4, pp. 517–580, 2003.

[JaJa00] J. JaJa. A Perspective on Quicksort. *Computing in Science and Engg.*, Vol. 2, No. 1, pp. 43–49, 2000.

[Jark97] M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos, and M. A. Jeusfeld, Eds. *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, Morgan Kaufmann, 1997.

[Knut78] D. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, Reading, MA., 1978.

[Mano05] Y. Manolopoulos, A. Nanopoulos, A. N. Papadopoulos, and Y. Theodoridis. R-Trees Have Grown Everywhere. 2005.

[Mano06] Y. Manolopoulos, A. Nanopoulos, A. Papadopoulos, and Y. Theodoridis. *R-Trees: Theory and Applications*. *Series: Advanced Information and Knowledge Processing*, 2006.

[Mato94] J. Matousek. Geometric Range Searching. *ACM Computing Surveys*, Vol. 26, No. 4, pp. 421–461, 1994.

[Mehl84a] K. Mehlhorn. *Data Structures and Algorithms 1: Sorting and Searching. EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, 1984.

[Mehl84b] K. Mehlhorn. *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*. *EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, 1984.

[Ooi93] B. C. Ooi, R. Sacks-Davis, and J. Han. Indexing in Spatial Databases. 1993. Unpublished Manuscript, available at: `http://www.iscs.nus.edu.sg/~ooibc/`.

[Papa05] A. N. Papadopoulos and Y. Manolopoulos. *Nearest Neighbor Search: A Database Perspective*. *Series: Series in Computer Science*, Springer Verlag, 2005.

[Proc97] O. Procopiuc. Data Structures for Spatial Database Systems. 1997. Unpublished Manuscript, available at `http://www.cs.duke.edu/~tavi/spatial.ps.gz`.

[Same06] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006.

[Same89] H. Samet. *Design and analysis of Spatial Data Structures: Quadtrees, Octrees, and other Hierarchical Methods*. Addison–Wesley, Redding, Mass., 1989.

[Same90] H. Samet. *Applications of Spatial Data Structures*. Addison-Wesley, Reading, Mass., 1990. chapter on ray tracing and efficiency, also discusses radiosity.

[Sell97] T. K. Sellis, N. Roussopoulos, and C. Faloutsos. Multidimensional Access Methods: Trees Have Grown Everywhere. In M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos, and M. A. Jeusfeld, Eds., *VLDB*, pp. 13–14, Morgan Kaufmann, 1997.

[Yorm84] B. Yormark, Ed. *SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, June 18-21, 1984*, ACM Press, 1984.

[Zach03] G. Zachmann and E. Langetepe. Geometric Data Structures for Computer Graphics. ACM SIGGRAPH 2003 Course Notes, 27–31July 2003.

Ray Shooting

[Adam05] B. Adams, R. Keiser, M. Pauly, L. Guibas, M. Gross, and P. Dutre. Efficient Raytracing of Deformable Point-Sampled Surfaces. In *Proceedings of the 2005 Eurographics Conference*, pp. 677–684, 2005.

[Adel95] S. J. Adelson and L. F. Hodges. Generating Exact Ray-Traced Animation Frames by Reprojection. *j-IEEE-CGA*, Vol. 15, No. 3, pp. 43–52, May 1995.

[Agat91] M. Agate, R. L. Grimsdale, and P. F. Lister. The HERO Algorithm for Ray-Tracing Octrees. In R. L. Grimsdale and W. Strasser, Eds., *Advances in Computer Graphics Hardware IV*, pp. 61–73, Springer-Verlag, London, UK, 1991.

[Aman84] J. Amanatides. Ray Tracing with Cones. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, pp. 129–135, July 1984.

[Aman87] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In G. Marechal, Ed., *Eurographics '87*, pp. 3–10, North-Holland, Aug. 1987.

[Aman90] J. Amanatides and D. P. Mitchell. Some Regularization Problems in Ray Tracing. In *Proceedings of Graphics Interface '90*, pp. 221–228, May 1990.

[Ar02] S. Ar, G. Montag, and A. Tal. Deferred, Self-Organizing BSP Trees. *Computer Graphics Journal (Eurographics '02)*, Vol. 21, No. 3, pp. C269–C278, 2002.

[Aron02] B. Aronov, H. Brönnimann, A. Chang, and Y.-J. Chiang. Cost prediction for ray tracing. In *Proceedings of the 18th Annual ACM Symposium on Computational Geometry (SoCG)*, pp. 293–302, ACM Press, Barcelona, Spain, June 2002.

[Aron05] B. Aronov, H. Brönnimann, A. Chang, and Y.-J. Chiang. Cost-driven octree construction schemes: an experimental study. *Computational Geometry: Theory & Applications*, Vol. 31, No. 1-2, pp. 127–148, 2005. Special Issue on the *19th ACM Annual Symposium on Computational Geometry - SoCG 2003*.

[Arvo87] J. Arvo and D. Kirk. Fast Ray Tracing by Ray Classification. In M. C. Stone, Ed., *(SIGGRAPH '87 Proceedings)*, pp. 55–64, July 1987.

[Arvo88] J. Arvo. Linear-time Voxel Walking for Octrees. *Ray Tracing News (available at `htpp://www.acm.org/tog/resources/RTNews/html/rtnews2d.html`*, Vol. 1, No. 5, p. , 1988.

[Arvo89] J. Arvo and D. Kirk. *A survey of ray tracing acceleration techniques*, pp. 201–262. Academic Press, 1989.

[Arvo90] J. Arvo. Ray Tracing with Meta-Hierarchies. In *SIGGRAPH '90 Advanced Topics in Ray Tracing course notes*, ACM Press, Aug. 1990.

[Badt88] J. S. Badt. Two Algorithms for Taking Advantage of Temporal Coherence in Ray Tracing. *The Visual Computer*, Vol. 4, No. 3, pp. 123–132, Sep. 1988.

[Bare96] G. Barequet, B. Chazelle, L. J. Guibas, J. S. B. Mitchell, and A. Tal. BOXTREE: A Hierarchical Representation for Surfaces in 3D. *Computer Graphics Forum*, Vol. 15, No. 3, pp. C387–C396, C484, Sep. 1996.

[Barr86] A. H. Barr. Ray Tracing Deformed Surfaces. *Computer Graphics*, Vol. 20, No. 4, pp. 287–296, Aug. 1986.

[Bart93] W. Barth and W. Sturzlinger. Efficient Ray Tracing for Bezier and B-Spline Surfaces. *Computers & Graphics*, Vol. 17, No. 4, pp. 423–430, July-Aug. 1993.

[Bart94] W. Barth, R. Lieger, and M. Schindler. Ray tracing general parametric surfaces using interval arithmetic. *Visual Computer*, Vol. 10, No. 7, pp. 363–371, 1994.

[Bell94] V. Belloli, S. Callegari, C. Gatti, M. Della Monica, and D. Marini. RayFilling: A new method to accelerate ray casting. *Computers and Graphics*, Vol. 18, No. 5, pp. 723–732, Sep.–Oct. 1994.

[Bent05] C. Benthin, I. Wald, and P. Slusallek. Techniques for Interactive Ray Tracing of Bezier Surfaces. *Journal of Graphics Tools*, 2005. to appear.

[Biar90] L. Biard. Parametric Surfaces and Ray Tracing. In *Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pp. 31–51, Rennes, France, June 1990.

[Boua87] K. Bouatouch, M. O. Madani, T. Priol, and B. Arnaldi. A New Algorithm of Space Tracing Using a CSG Model. In G. Marechal, Ed., *Eurographics '87*, pp. 65–78, North-Holland, Aug. 1987.

[Bron02] H. Brönnimann and M. Glisse. Cost optimal trees for ray shooting. In *Abstracts of the 12th Fall Workshop Computational Geometry*, DIMACS, November 2002.

[Bron85] W. F. Bronsvoort and F. Klok. Ray Tracing Generalized Cylinders. *ACM Transactions on Graphics*, Vol. 4, No. 4, pp. 291–303, Oct. 1985.

[Camp97] S. Campagna, P. Slusallek, and H. Seidel. Ray tracing of spline surfaces: Bézier clipping, Chebyshev boxing, and bounding volume hierarchy - a critical comparison with new results. *The Visual Computer*, Vol. 13, No. 6, pp. 265–282, 1997. ISSN 0178-2789.

[Cass95] T. Cassen, K. R. Subramanian, and Z. Michalewicz. Near-Optimal Construction of Partitioning Trees by Evolutionary Techniques. In *Proceedings of Graphics Interface '95*, pp. 263–271, Canada, June 1995.

[Caza95] F. Cazals, G. Drettakis, and C. Puech. Filtering, Clustering and Hierarchy Construction: A New Solution for Ray-Tracing Complex Scenes. *Computer Graphics Forum*, Vol. 14, No. 3, pp. C/371–382, 1995.

[Caza97] F. Cazals and C. Puech. Bucket-like space partitioning data-structures with applications to ray-tracing. In *13th ACM Symposium on Computational Geometry*, pp. 11–20, Nice, 1997.

[Chan04] A. Y.-H. Chang. *Theoretical and Experimental Aspects of Ray Shooting*. PhD thesis, Politechnic University, USA, 2004.

[Chap90] J. Chapman, T. W. Calvert, and J. Dill. Exploiting Temporal Coherence in Ray Tracing. In *Proceedings of Graphics Interface '90*, pp. 196–204, Canadian Information Processing Society, Toronto, Ontario, May 1990.

[Chap91] J. Chapman, T. W. Calvert, and J. Dill. Spatio-Temporal Coherence in Ray Tracing. In *Proceedings of Graphics Interface '91*, pp. 101–108, June 1991.

[Char90] M. J. Charney and I. D. Scherson. Efficient Traversal of Well-Behaved Hierarchicial Trees of Extents for Ray-Tracing Complex Scenes. *The Visual Computer*, Vol. 6, No. 3, pp. 167–178, June 1990.

[Choi92] H. K. Choi and C. M. Kyung. PYSHA: a shadow-testing acceleration scheme for ray tracing. *Computer-aided design*, Vol. 24, No. 2, pp. 93–104, Feb. 1992. hybrid scheme of light buffer and grid subdivision with cost comparison on the fly.

[Chua95] J.-H. Chuang and W.-J. Hwang. A new space subdivision for ray tracing CSG solids. *IEEE Computer Graphics and Applications*, Vol. 15, No. 6, pp. 56–62, Nov. 1995.

[Clea88] J. G. Cleary and G. Wyvill. Analysis of an algorithm for fast ray tracing using uniform space subdivision. *The Visual Computer*, Vol. 4, No. 2, pp. 65–83, July 1988.

[Cohe94] D. Cohen. Voxel Traversal along a 3D Line. In P. Heckbert, Ed., *Graphics Gems IV*, pp. 366–369, Academic Press, Boston, 1994.

[Coqu84] S. Coquillart and M. Gangnet. Shaded Display of Digital Maps. *j-IEEE-CGA*, Vol. 4, No. 7, pp. 35–42, July 1984.

[Cych92] J. M. Cychosz. Use of Residency Masks and Object Space Partitioning to Eliminate Ray-Object Intersection Calculations. In D. Kirk, Ed., *Graphics Gems III*, pp. 284–287, Academic Press, San Diego, 1992.

[DCoh94] D.Cohen and Z.Sheffer. Proximity clouds - an acceleration technique for 3D grid traversal. *The Visual Computer*, Vol. 11, pp. 27–38, 1994.

[Devi89] O. Devillers. The Macro-regions: an Efficient Space Subdivision Structure for Ray Tracing. In W. Hansmann, F. R. A. Hopgood, and W. Strasser, Eds., *Eurographics '89*, pp. 27–38, Elsevier / North-Holland, Sep. 1989.

[Dmit04] K. Dmitriev, V. Havran, and H.-P. Seidel. Faster Ray Tracing with SIMD Shaft Culling. Research Report MPI-I-2004-4-006, Max-Planck-Institut für Informatik, Saarbrücken, Germany, December 2004.

[Efre05a] A. Efremov. *Efficient Ray Tracing of Trimmed NURBS Surfaces*. Master's thesis, MPI Informatik, Germany, 2005.

[Efre05b] A. Efremov, V. Havran, and H.-P. Seidel. Robust and Numerically Stable Bezier Clipping Method for Ray Tracing NURBS Surfaces. In B. Juettler, Ed., *21st Spring Conference on Computer Graphics (SCCG 2005)*, pp. 123–131, ACM SIGGRAPH and EUROGRAPHICS, ACM, Budmerice, Slovakia, 2005.

[Endl94] R. Endl and M. Sommer. Classification of ray-generators in uniform subdivisions and octrees for ray tracing. *Computer Graphics Forum*, Vol. 13, No. 1, pp. 3–19, March 1994.

[Endl95] R. Endl. An Object-Oriented Ray Tracing Architecture for the Analysis of Ray-Generators in Spatial Subdivisions. In H. P. Santo, Ed., *Compugraphics '95*, pp. 268–277, Dec. 1995. ISBN 972-8342-00-4.

[Enge92] W. Enger. Interval Ray Tracing – a divide and conquer strategy for realistic computer graphics. *The Visual Computer*, Vol. 8, No. 9, pp. 91–104, 1992. ISSN 0178-2789.

[Fori96] T. Foris, G. Márton, and L. Szirmay-Kalos. Ray Shooting in Logarithmic Time. In *Winter School of Computer Graphics 1996*, pp. 84–90, Feb. 1996. held at University of West Bohemia, Plzen, Czech Republic, 12-16 February 1996.

[Four93] A. Fournier and P. Poulin. A Ray Tracing Accelerator Based on a Hierarchy of 1D Sorted Lists. In *Proceedings of Graphics Interface '93*, pp. 53–61, Canadian Information Processing Society, Toronto, Ontario, May 1993.

[Four94] A. Fournier and J. Buchanan. Chebyshev Polynomials for Boxing and Intersections of Parametric Curves and Surfaces. *Computer Graphics Forum*, Vol. 13, No. 3, pp. C/127–C/142, 1994.

[Fuji86] A. Fujimoto, T. Tanaka, and K. Iwata. ARTS: Accelerated Ray Tracing System. *IEEE Computer Graphics and Applications*, Vol. 6, No. 4, pp. 16–26, 1986.

[Garg93] I. Gargantini and H. H. Atkinson. Ray tracing an octree: numerical evaluation of the first intersection. *Computer Graphics Forum*, Vol. 12, No. 4, pp. 199–210, Oct. 1993.

[Garg95] I. Gargantini and J. H. G. Redekop. Evaluating Exact Intersections of an Octree with Full Rays using only Radix-Sort and Meet Operations. In H. P. Santo, Ed., *Compugraphics '95*, pp. 278–284, Dec. 1995. ISBN 972-8342-00-4.

[Gene93] J. Genetti and D. Gordon. Ray Tracing with Adaptive Supersampling in Object Space. In *Proceedings of Graphics Interface '93*, pp. 70–77, Canadian Information Processing Society, Toronto, Ontario, May 1993.

[Gene98] J. Genetti, D. Gordon, and G. Williams. Adaptive Supersampling in Object Space Using Pyramidal Rays. In *Computer Graphics Forum*, pp. 29–54, March 1998.

[Gerv86] M. Gervautz. Three Improvements of the Ray Tracing Algorithm For CSG Trees. *Computers and Graphics*, Vol. 10, No. 4, pp. 333–339, 1986.

[Gerv92] M. Gervautz. Consistent schemes for addressing surfaces when ray tracing transparent CSG objects. *Computer Graphics Forum*, Vol. 11, No. 4, pp. 203–211, Oct. 1992.

[Giga88] M. Gigante. Accelerated Ray Tracing Using Non-Uniform Grids. In *Proceedings of Ausgraph '90*, pp. 157–163, 1988.

[Glas84] A. S. Glassner. Space Subdivision For Fast Ray Tracing. *IEEE Computer Graphics and Applications*, Vol. 4, No. 10, pp. 15–22, Oct. 1984.

[Glas88] A. S. Glassner. Spacetime ray tracing for animation. *IEEE Computer Graphics and Applications*, Vol. 8, No. 2, pp. 60–70, March 1988.

[Glas89] A. S. Glassner. *An Introduction to Ray Tracing*. Academic Press, 1989.

[Gold87] J. Goldsmith and J. Salmon. Automatic Creation of Object Hierarchies for Ray Tracing. *IEEE Computer Graphics and Applications*, Vol. 7, No. 5, pp. 14–20, May 1987.

[Gonz98] P. Gonzalez and F. Gisbert. Object and Ray Coherence in the Optimization of the Ray Tracing Algorithm. In *Proceedings of Computer Graphics International '98 (CGI'98)*, pp. 264–267, IEEE, NY, Hannover, Germany, June 1998.

[Groe93] E. Groeller. Oct-tracing animation sequences. In *Summer school in computer graphics in Bratislava (SCCG93)*, pp. 96–101, June 1993.

[Grol91] E. Groller and W. Purgathofer. Using Temporal and Spatial Coherence for Accelerating the Calculation of Animation Sequences. In W. Purgathofer, Ed., *Eurographics '91*, pp. 103–113, North-Holland, Sep. 1991.

[Gunt06] J. Günther, H. Friedrich, I. Wald, H.-P. Seidel, and P. Slusallek. Ray Tracing Animated Scenes using Motion Decomposition. *Technical Report UUSCI-2006-022*, 2006. (to appear).

[Hain86] E. A. Haines and D. P. Greenberg. The Light Buffer: A Ray Tracer Shadow Testing Accelerator. *IEEE Computer Graphics and Applications*, Vol. 6, No. 9, pp. 6–16, Sep. 1986.

[Hain87] E. A. Haines. A Proposal for Standard Graphics Environments. *IEEE Computer Graphics and Applications*, Vol. 7, No. 11, pp. 3–5, Nov. 1987. Available from `http://www.acm.org/pubs/tog/resources/SPD/overview.html`.

[Hain91a] E. A. Haines. Efficiency Improvements for Hierarchy Traversal. In J. Arvo, Ed., *Graphics Gems II*, pp. 267–273, Academic Press, San Diego, 1991.

[Hain91b] E. A. Haines. Fast Ray-Convex Polyhedron Intersection. In J. Arvo, Ed., *Graphics Gems II*, pp. 247–250, Academic Press, San Diego, 1991. includes code.

[Hanr83] P. Hanrahan. Ray Tracing Algebraic Surfaces. *Computer Graphics (SIGGRAPH '83 Proceedings)*, Vol. 17, No. 3, pp. 83–90, July 1983.

[Hart89] J. C. Hart, D. J. Sandin, and L. H. Kauffman. Ray Tracing Deterministic 3-D Fractals. In J. Lane, Ed., *Computer Graphics (SIGGRAPH '89 Proceedings)*, pp. 289–296, July 1989.

[Hart93] J. C. Hart. Ray Tracing Implicit Surfaces. Tech. Rep. EECS-93-014, Washington State University - School of EECS, 1993.

[Hart96] J. C. Hart. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, Vol. 12, No. 9, pp. 527–545, 1996. ISSN 0178-2789.

[Havr00a] V. Havran and J. Bittner. LCTS: Ray Shooting using Longest Common Traversal Sequences. *Computer Graphics Forum (Proc. Eurographics '2000)*, Vol. 19, No. 3, pp. C59–C70, Aug 2000.

[Havr00b] V. Havran, L. Dachs, and J. Žára. VIS-RT: A Visualization System for RT Spatial Data Structures. In *Proceedings of WSCG'2000, short communication papers*, pp. 28–35, Feb 2000.

[Havr00c] V. Havran. *Heuristic Ray Shooting Algorithms*. PhD thesis, Czech Technical University in Prague, November 2000.

[Havr00d] V. Havran, J. Přikryl, and W. Purgathofer. Statistical Comparison of Ray-Shooting Efficiency Schemes. Tech. Rep. TR-186-2-00-14, Institute of Computer Graphics, Vienna University of Technology, Favoritenstrasse 9/186, A-1040 Vienna, Austria, May 2000. human contact: technical-report@cg.tuwien.ac.at.

[Havr02] V. Havran and J. Bittner. On Improving KD-Trees for Ray Shooting. *Journal of WSCG*, Vol. 10, No. 1, pp. 209–216, February 2002.

[Havr03a] V. Havran, J. Bittner, and H.-P. Seidel. Exploiting Temporal Coherence in Ray Casted Walkthroughs. In K. I. Joy, Ed., *Proceedings of the 19th Spring Conference on Computer Graphics 2003 (SCCG 03)*, pp. 164–172, ACM, Budmerice, Slovakia, April 2003.

[Havr03b] V. Havran, C. Damez, K. Myszkowski, and H.-P. Seidel. An Efficient Spatio-Temporal Architecture for Animation Rendering. In P. Christensen and D. Cohen-Or, Eds., *Rendering Techniques 2003 : 14th Eurographics Workshop on Rendering*, pp. 106–117, Association of Computing Machinery (ACM), ACM, Leuven, Belgium, June 2003.

[Havr03c] V. Havran and W. Purgathofer. On Comparing Ray Shooting Algorithms. *Computers & Graphics*, Vol. 27, No. 4, pp. 593–604, 2003.

[Havr05] V. Havran, R. Herzog, and H.-P. Seidel. Fast Final Gathering via Reverse Photon Mapping. *Computer Graphics Forum (Proceedings of Eurographics 2005)*, Vol. 24, No. 3, pp. 323–333, August 2005.

[Havr97a] V. Havran. Cache Sensitive Representation for the BSP Tree. In *Proceedings of Compugraphics'97*, pp. 369–376, GRASP – Graphics Science Promotions & Publications, Dec. 1997.

[Havr97b] V. Havran, T. Kopal, J. Bittner, and J. Žára. Fast Robust Bsp Tree Traversal Algorithm for Ray Tracing. *Journal of Graphics Tools*, Vol. 2, No. 4, pp. 15–23, Dec. 1997. Published in 1998.

[Havr97c] V. Havran and J. Žára. Evaluation of BSP properties for ray–tracing. In *Proceedings of SCCG'97 (Spring Conference on Computer Graphics)*, pp. 155–162, Budmerice, Jun 1997.

[Havr98] V. Havran, J. Bittner, and J. Žára. Ray Tracing with Rope Trees. In *Proceedings of SCCG'98 (Spring Conference on Computer Graphics)*, pp. 130–139, Budmerice, Slovak Republic, Apr. 1998.

[Havr99a] V. Havran. Analysis of Cache Sensitive Representation for Binary Space Partitioning Trees. *Informatica*, Vol. 23, No. 3, pp. 203–210, May 1999.

[Havr99b] V. Havran and J. Bittner. Rectilinear BSP Trees for Preferred Ray Sets. In *Proceedings of SCCG'99 (Spring Conference on Computer Graphics)*, pp. 171–179, Budmerice, Slovak Republic, Apr/May 1999.

[Heck84] P. S. Heckbert and P. Hanrahan. Beam Tracing Polygonal Objects. *Computer Graphics (SIGGRAPH'84 Proceedings)*, Vol. 18, No. 3, pp. 119–127, July 1984.

[Heid98] W. Heidrich and H.-P. Seidel. Ray-Tracing Procedural Displacement Shaders. In *Graphics Interface*, pp. 8–16, June 1998.

[Held97] M. Held. ERIT: A Collection of Efficient and Reliable Intersection Tests. *journal of graphics tools*, Vol. 2, No. 4, pp. 25–44, 1997.

[Horv92] T. Horvath, G. Márton, P. Risztics, and L. Szirmay-Kalos. Ray coherence between a sphere and a convex polyhedron. *Computer Graphics Forum*, Vol. 11, No. 2, pp. 163–172, June 1992.

[Hsiu92] P.-K. Hsiung and R. H. Thibadeau. Accelerating ARTS. *The Visual Computer*, Vol. 8, No. 3, pp. 181–190, March 1992. nested grid subdivision structures.

[Ip97] H. H. S. Ip, K. C. K. Law, and G. K. P. Fung. Epipolar plane space subdivision method in stereoscopic ray tracing. *The Visual Computer*, Vol. 13, No. 6, pp. 247–264, 1997. ISSN 0178-2789.

[Jans86] F. W. Jansen. Data Structures for Ray Tracing. In L. R. A. Kessener, F. J. Peters, and M. L. P. van Lierop, Eds., *Data Structures for Raster Graphics*, pp. 57–73, Springer-Verlag, New York, 1986. Eurographic Seminar.

[Jeva89] D. Jevans and B. Wyvill. Adaptive voxel subdivision for ray tracing. *Proceedings of Graphics Interface '89*, pp. 164–172, June 1989.

[Jeva92] D. Jevans. Object Space Temporal Coherence for Ray Tracing. In *Proceedings of Graphics Interface '92*, pp. 176–183, Canadian Information Processing Society, Toronto, Ontario, May 1992.

[Joy86] K. I. Joy and M. N. Bhetanabhotla. Ray Tracing Parametric Surface Patches Utilizing Numerical Techniques and Ray Coherence. *Computer Graphics*, Vol. 20, No. 4, pp. 279–285, Aug. 1986.

[Kaji83] J. T. Kajiya. New Techniques For Ray Tracing Procedurally Defined Objects. In *Computer Graphics (SIGGRAPH '83 Proceedings)*, pp. 91–102, July 1983.

[Kalr89] D. Kalra and A. H. Barr. Guaranteed Ray Intersections with Implicit Surfaces. In J. Lane, Ed., *SIGGRAPH '89 Proceedings*, pp. 297–306, July 1989.

[Kapl85] M. Kaplan. *Space-Tracing: A Constant Time Ray-Tracer*, pp. 149–158. July 1985.

[Kapl87] M. R. Kaplan. The Use of Spatial Coherence in Ray Tracing. In D. E. Rogers and R. A. Earnshaw, Eds., *Techniques for Computer Graphics*, pp. 173–193, Springer Verlag, 1987.

[Kay86] T. L. Kay and J. T. Kajiya. Ray Tracing Complex Scenes. In D. C. Evans and R. J. Athay, Eds., *SIGGRAPH '86 Proceedings)*, pp. 269–278, Aug. 1986.

[Ke93] H.-R. Ke and R.-C. Chang. An Efficient Hierarchical Traversal Algorithm for Ray Tracing. *Visual Computer*, Vol. 10, No. 2, pp. 79–87, 1993.

[Ke95] H. R. Ke and R. C. Chang. Ray-cast volume rendering accelerated by incremental trilinear interpolation and cell templates. *The Visual Computer*, Vol. 11, No. 6, pp. 297–308, 1995. ISSN 0178-2789.

[Keat95] M. J. Keates and R. J. Hubbold. Interactive Ray Tracing on a Virtual Shared-Memory Parallel Computer. *Computer Graphics Forum*, Vol. 14, No. 4, pp. 189–202, Oct. 1995.

[Kirk91] D. Kirk and J. Arvo. Improved Ray Tagging for Voxel-Based Ray Tracing. In J. Arvo, Ed., *Graphics Gems II*, pp. 264–266, Academic Press, San Diego, 1991.

[Klim97] K. S. Klimaszewski and T. W. Sederberg. Faster Ray Tracing Using Adaptive Grids. *IEEE Computer Graphics and Applications*, Vol. 17, No. 1, pp. 42–51, Jan./Feb. 1997.

[Kuzm94] Y. P. Kuzmin. Ray Traversal of Spatial Structures. *Computer Graphics Forum*, Vol. 13, No. 4, pp. 223–227, Oct. 1994.

[Kwon98] B. Kwon, D. S. Kim, K.-Y. Chwa, and S. Y. Shin. Memory-Efficient Ray Classification for Visibility Operations. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 4, No. 3, pp. 193–201, july-september 1998.

[Law95] A. Law and R. Yagel. Multi-Frame Thrashless Ray Casting with Advancing Ray-Front. Tech. Rep., Ohio State University, 1995. OSU-CISRC-11/95-TR50.

[Levn88] G. Levner, P. Tassinari, and D. Marini. A simple general method for ray tracing bicubic surfaces. In R. A. Earnshaw, Ed., *Theoretical Foundations of Computer Graphics and CAD*, pp. 805–820, Springer-Verlag, 1988.

[Lext00] J. Lext, U. Assarsson, and T. Möller. BART: A Benchmark for Animated Ray Tracing. Tech. Rep., Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden, May 2000. Available at `http://www.ce.chalmers.se/BART/`.

[Lext01a] J. Lext and T. Akenine-Möller. Towards Rapid Reconstruction for Animated Ray Tracing. In *Eurographics 2001 – Short Presentations*, pp. 311–318, 2001.

[Lext01b] J. Lext, U. Assarsson, and T. Möller. A Benchmark for Animated Ray Tracing. *IEEE Comput. Graph. Appl.*, Vol. 21, No. 2, pp. 22–31, 2001.

[Lin91] T. T. Y. Lin and M. Slater. Stochastic Ray Tracing Using SIMD Processor Arrays. *The Visual Computer*, Vol. 7, No. 4, pp. 187–199, July 1991.

[Lisc90] D. Lischinski and J. Gohczarowski. Improved Techniques for Ray Tracing Parametric Surfaces. *The Visual Computer*, Vol. 6, No. 3, pp. 134–152, June 1990.

[MacD88] D. MacDonald. *Space Subdivision Algorithms for Ray Tracing*. Master's thesis, Department of Computer Science, University of Waterloo, 1988.

[MacD89] J. D. MacDonald and K. S. Booth. Heuristics for Ray Tracing Using Space Subdivision. In *Proceedings of Graphics Interface '89*, pp. 152–63, Canadian Information Processing Society, Toronto, Ontario, June 1989. criteria for building octree (actually BSP) efficiency structures.

[MacD90] J. D. MacDonald and K. S. Booth. Heuristics for Ray Tracing Using Space Subdivision. *Visual Computer*, Vol. 6, No. 6, pp. 153–65, 1990. criteria for building octree (actually BSP) efficiency structures.

[Maho05] J. Mahovsky. *Ray Tracing with Reduced-Precision Bounding Volume Hierarchies*. PhD thesis, University of Calgary, 2005.

[Mail92] J.-L. Maillot, L. Carraro, and B. Peroche. Progressive Ray Tracing. *Third Eurographics Workshop on Rendering*, pp. 9–20, May 1992.

[Mais92] E. Maisel and G. Hégron. A Realistic Image Synthesis of Animation Sequences based on Temporal Coherence. In G. Hegron and D. Thalmann, Eds., *Computer Animation and Simulation '92*, Eurographics, ISSN 1017-4656, Cambridge, England, Sep. 1992. Proceedings of the Eurographics Workshop in Cambridge, England, September 7–11, 1992.

[Mart00] W. Martin, E. Cohen, R. Fish, and P. Shirley. Practical Ray Tracing of Trimmed NURBS Surfaces. *Journal of Graphics Tools: JGT*, Vol. 5, No. 1, pp. 27–52, 2000.

[Mart95a] G. Márton and L. Szirmay-Kalos. On Average-case Complexity of Ray Tracing Algorithms. In *Winter School of Computer Graphics 1995*, pp. 187–196, Feb. 1995. held at University of West Bohemia, Plzen, Czech Republic, 14-18 February 1995.

[Mart95b] G. Márton. Acceleration of Ray Tracing via Voronoi Diagrams. In A. W. Paeth, Ed., *Graphics Gems V*, pp. 268–284, Academic Press, Boston Mass., 1995.

[Mart97] G. Márton. Surfaces for Ray Tracing: A Fast View-Dependent Algorithm. In H. N. H.P. Seidel, B. Girod, Ed., *Proceedings of 3D Image Analysis and Synthesis '97*, pp. 19–26, Nov. 1997.

[Maur93] H. Maurel, Y. Duthen, and R. Caubet. A 4D ray tracing. *Computer Graphics Forum*, Vol. 12, No. 3, pp. C285–C294, Aug 1993.

[McNe92] M. D. J. McNeill, B. C. Shah, M.-P. Hebert, P. F. Lister, and R. L. Grimsdale. Performance of space subdivision techniques in ray tracing. *Computer Graphics Forum*, Vol. 11, No. 4, pp. 213–220, Oct. 1992.

[Mein91] H.-P. Meinzer, K. Meetz, D. Scheppelmann, U. Engelmann, and H. J. Baur. The Heidelberg Ray Tracing Model. *j-IEEE-CGA*, Vol. 11, No. 6, pp. 34–43, Nov. 1991.

[Mitc90] D. P. Mitchell. Robust Ray Intersection with Interval Arithmetic. *Graphics Interface*, pp. 68–74, 1990.

[Mitc94] J. S. B. Mitchell, D. M. Mount, and S. Suri. Query-Sensitive Ray Shooting. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pp. 359–368, 1994.

[Moel95] T. Moeller. A Linear-time simple bounding volume algorithm. In A. W. Paeth, Ed., *Graphics Gems V*, pp. 242–257, Academic Press, Boston Mass., 1995.

[Moll97] T. Möller and B. Trumbore. Fast, Minimum Storage Ray-Triangle Intersection. *Journal of Graphics Tools: JGT*, Vol. 2, No. 1, pp. 21–28, 1997.

[Mont90] C. Montani and R. Scopigno. Ray tracing CSG trees using the sticks representation scheme. *Computers and Graphics*, Vol. 14, No. 3/4, pp. 481–490, 1990.

[Muel88] H. Mueller. Time coherence in computer animation by ray tracing. In *Proceedings of Computational Geometry and Applications*, pp. 187–201, 1988. vol.333 of Lecture Notes in Computer Science, Springer Verlag.

[Muel99] G. Mueller and D. W. Fellner. Hybrid Scene Structuring with Application to Ray Tracing. In *Proceedings of International Conference on Visual Computing (ICVC'99)*, pp. 19–26, Goa, India, Feb. 1999.

[Musg88] F. K. Musgrave. Grid Tracing: Fast Ray Tracing for Height Fields. Technical Report YALEU/DCS/RR-639, Yale University Dept. of Computer Science Research, 1988.

[Naka97] K. Nakamaru and Y. Ohno. Breadth-First Ray Tracing Utilizing Uniform Spatial Subdivision. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 3, No. 4, pp. 316–328, Oct. 1997.

[Nish90] T. Nishita, T. W. Sederberg, and M. Kakimoto. Ray Tracing Trimmed Rational Surface Patches. *Computer Graphics*, Vol. 24, No. 4, pp. 337–345, Aug. 1990.

[Nish94] T. Nishita and E. Nakamae. A Method for Displaying Metaballs by using Bézier Clipping. *Computer Graphics Forum*, Vol. 13, No. 3, pp. C/271–C/280, 1994.

[Ohta87] M. Ohta and M. Maekawa. Ray Coherence Theorem and Constant Time Ray Tracing Algorithm. In T. L. Kunii, Ed., *Computer Graphics 1987 (Proceedings of CG International '87)*, pp. 303–314, Springer-Verlag, 1987.

[Pagl92] D. W. Paglieroni and S. M. Petersen. Parametric height field ray tracing. In *Proceedings of Graphics Interface '92*, pp. 192–200, May 1992.

[Pagl94a] D. W. Paglieroni. Directional Distance Transforms and Height Field Preprocessing for Efficient Ray Tracing. *Graphical Models and Image Processing*, Vol. 59, No. 4, pp. 253–264, July 1994.

[Pagl94b] D. W. Paglieroni and S. M. Petersen. Height Distributional Distance Transform Methods for Height Field Ray Tracing. *ACM Transactions on Graphics*, Vol. 13, No. 4, pp. 376–399, Oct. 1994.

[Pagl98] D. W. Paglieroni. The Directional Parameter Plane Transform of a Height Field. *ACM Transactions on Graphics*, Vol. 17, No. 1, pp. 50–70, Jan. 1998.

[Park98] S. Parker, P. Shirley, Y. Livnat, C. Hansen, and P.-P. Sloan. Interactive Ray Tracing for Isosurface Rendering. In D. Ebert, H. Hagen, and H. Rushmeier, Eds., *IEEE Visualization '98*, pp. 233–238, IEEE, 1998.

[Park99] S. Parker, W. Martin, P.-P. Sloan, P. Shirley, B. Smits, and C. Hansen. Interactive Ray Tracing. In *Symposium on Interactive 3D Graphics: Interactive 3D*, pp. 119–126, April 26-28 1999.

[Pear91a] A. Pearce. Avoiding Incorrect Shadow Intersections for Ray Tracing. In J. Arvo, Ed., *Graphics Gems II*, pp. 275–276, Academic Press, San Diego, 1991.

[Pear91b] A. Pearce. A Recursive Shadow Voxel Cache for Ray Tracing. In J. Arvo, Ed., *Graphics Gems II*, pp. 273–274, Academic Press, San Diego, 1991. includes code.

[Pear91c] A. Pearce and D. Jevans. Exploiting Shadow Coherence in Ray Tracing. In *Proceedings of Graphics Interface '91*, pp. 109–116, June 1991.

[Peng87] Q. Peng, Y. Zhu, and Y. Liang. A Fast Ray Tracing Algorithm Using Space Indexing Techniques. In G. Marechal, Ed., *Eurographics '87*, pp. 11–23, North-Holland, Aug. 1987.

[Phar96] M. Pharr and P. Hanrahan. Geometry Caching for Ray-Tracing Displacement Maps. In X. Pueyo and P. Schröder, Eds., *Eurographics Rendering Workshop 1996*, pp. 31–40, Eurographics, Springer Wein, New York City, NY, June 1996. ISBN 3-211-82883-4.

[Phar97] M. Pharr, C. Kolb, R. Gershbein, and P. Hanrahan. Rendering Complex Scenes with Memory-Coherent Ray Tracing. In T. Whitted, Ed., *SIGGRAPH 97 Conference Proceedings*, pp. 101–108, ACM SIGGRAPH, Addison Wesley, Aug. 1997. ISBN 0-89791-896-7.

[Prad91] B. S. S. Pradhan and A. Mukhopadhyay. Adaptive cell division for ray tracing. *Computers and Graphics*, Vol. 15, No. 4, pp. 549–552, 1991.

[Qin97] K. Qin, M. Gong, Y. Guan, and W. Wang. A new method for speeding up ray tracing NURBS surfaces. *Computers and Graphics*, Vol. 21, No. 5, pp. 577–586, Sep.–Oct. 1997.

[Quai96] M. Quail. Space Time Ray Tracing using Ray Classification. Bachelor thesis, Nov. 1996.

[Rein00] E. Reinhard, B. Smits, and C. Hansen. Dynamic Acceleration Structures for Interactive Ray Tracing. In *Proceedings of the Eurographics Workshop on Rendering*, pp. 299–306, Brno, Czech Republic, June 2000.

[Rein96] E. Reinhard, A. J. F. Kok, and F. W. Jansen. Cost Prediction in Ray Tracing. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pp. 41–50, Springer-Verlag/Wien, New York, NY, 1996.

[Reis97] A. Reisman, C. Gotsmann, and A. Schuster. Parallel Progressive Rendering of Animation Sequences at Interactive Rates on Distributed-Memory Machines. In J. Painter, G. Stoll, and Kwan-Liu Ma, Eds., *IEEE Parallel Rendering Symposium*, pp. 39–48, IEEE, Nov. 1997. ISBN 1-58113-010-4.

[Resh05] A. Reshetov, A. Soupikov, and J. Hurley. Multi-Level Ray Tracing Algorithm. *ACM Transaction of Graphics*, Vol. 24, No. 3, pp. 1176–1185, 2005. (Proceedings of ACM SIGGRAPH).

[Reve00] J. Revelles, C. Urena, and M. Lastra. An Efficient Parametric Algorithm for Octree Traversal. In *Proceedings of WSCG'2000*, pp. 212–219, feb 2000. held at University of West Bohemia, Plzen, Czech Republic, February 2000.

[Ritt90] J. Ritter. A Simple Ray Rejection Test. In A. S. Glassner, Ed., *Graphics Gems*, pp. 385–386, Academic Press, San Diego, 1990.

[Rubi80] S. M. Rubin and T. Whitted. A 3-Dimensional Representation for Fast Rendering of Complex Scenes. In *SIGGRAPH '80 Proceedings*, pp. 110–116, July 1980.

[Same89] H. Samet. Implementing Ray Tracing with Octrees and Neighbor Finding. *Computers and Graphics*, Vol. 13, No. 4, pp. 445–60, 1989. includes code.

[Sand85] J. Sandor. Octree Data Structures and Perspective Imagery. *Computers and Graphics*, Vol. 9, No. 4, pp. 393–405, 1985.

[Sche87] I. D. Scherson and E. Caspary. Data Structures and the Time Complexity of Ray Tracing. *The Visual Computer*, Vol. 3, No. 4, pp. 201–213, Dec. 1987.

[Sede84] T. W. Sederberg and D. C. Anderson. Ray Tracing of Steiner Patches. In H. Christiansen, Ed., *Computer Graphics (SIGGRAPH '84 Proceedings)*, pp. 159–164, July 1984.

[Semw87] S. K. Semwal. *The Slicing Extent Technique for Fast Ray Tracing*. PhD thesis, Department of Computer Science, University of Central Florida, 1987.

[Semw97] S. Semwal and H. Kvarnstrom. Directional Safe Zones & Dual Extent Algorithms for Efficient Grid Traversal. In *Graphics Interface 97*, pp. 76–87, 1997. University of Colorado.

[Sher96] A. Sherstyuk. Ray-tracing implicit surfaces: a generalized approach. Technical Report 1996/290, Monash University, 1996.

[Sher98] A. Sherstyuk. Fast Ray Tracing of Implicit Surfaces. Technical Report 1998/04, Monash University, 1998.

[Shin87] M. Shinya, T. Takahashi, and S. Naito. Principles and Applications of Pencil Tracing. In M. C. Stone, Ed., *Computer Graphics (SIGGRAPH '87 Proceedings)*, pp. 45–54, July 1987.

[Shir91] P. Shirley, K. Sung, and W. Brown. A Ray Tracing Framework for Global Illumination Systems. In *Proceedings of Graphics Interface '91*, pp. 117–128, June 1991.

[Simi94] G. Simiakakis and A. M. Day. Five-dimensional Adaptive Subdivision for Ray Tracing. *Computer Graphics Forum*, Vol. 13, No. 2, pp. 133–140, June 1994.

[Simi95] G. Simiakakis. *Accelerating RayTracing with Directional Subdivision and Parallel Processing*. PhD thesis, University of East Anglia, october 1995.

[Slat92] M. Slater. Tracing a Ray Through Uniformly Subdivided n-Dimensional Space. *The Visual Computer*, Vol. 9, No. 1, pp. 39–46, 1992.

[Smit98] B. Smits. Efficiency Issues for Ray Tracing. *Journal of Graphics Tools: JGT*, Vol. 3, No. 2, pp. 1–14, 1998.

[Snyd87] J. M. Snyder and A. H. Barr. Ray Tracing Complex Models Containing Surface Tessellations. In M. C. Stone, Ed., *Computer Graphics (SIGGRAPH '87 Proceedings)*, pp. 119–128, July 1987.

[Spee92] L. R. Speer. A New Subdivision Method for High-speed, Memory Efficient Ray Shooting. In *Third Eurographics Workshop on Rendering*, pp. 45–60, Bristol, UK, May 1992.

[Stei84] H. A. Steinberg. A Smooth Surface Based on Biquadratic Patches. *j-IEEE-CGA*, Vol. 4, No. 9, pp. 20–23, Sep. 1984.

[Stue96] W. Stuerzlinger. Bounding Volume Construction using Point Clouds. In *Summer school in computer graphics in Bratislava (SCCG96)*, pp. 239–246, June 1996.

[Stur98] W. Sturzlinger. Ray-Tracing Triangular Trimmed Free-Form Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 4, No. 3, pp. 202–214, july-september 1998.

[Subr90a] K. R. Subramanian and D. S. Fussel. Applying Space Subdivision Techniques to Volume Rendering. Aug. 1990.

[Subr90b] K. R. Subramanian and D. S. Fussel. Factors Affecting Performance of Ray Tracing Hierarchies. Tech. Rep. Tx 78712, The University of Texas at Austin, July 1990.

[Subr90c] K. R. Subramanian and D. S. Fussel. A Search Structure based on K-d Trees for Efficient Ray Tracing. Tech. Rep. PhD Dissertation, Tx 78712-1188, The University of Texas at Austin, Dec. 1990.

[Subr91] K. R. Subramanian and D. S. Fussell. Automatic Termination Criteria for Ray Tracing Hierarchies. In *Proceedings of Graphics Interface '91*, pp. 93–100, June 1991.

[Sung91] K. Sung. A DDA Octree Traversal Algorithm for Ray Tracing. In W. Purgathofer, Ed., *Eurographics '91*, pp. 73–85, North-Holland, Sep. 1991.

[Sung92a] K. Sung. Area sampling buffer: tracing rays with Z-buffer hardware. *Computer Graphics Forum*, Vol. 11, No. 3, pp. C299–C310, C480–C481, 1992.

[Sung92b] K. Sung and P. Shirley. Ray Tracing with the BSP Tree. In D. Kirk, Ed., *Graphics Gems III*, pp. 271–274, Academic Press, San Diego, 1992. includes code.

[Swee86] M. Sweeney and R. H. Bartels. Ray Tracing Free-Form B-Spline Surfaces. *IEEE Computer Graphics and Applications*, Vol. 6, No. 2, p. 41, Feb. 1986.

[Szec03] L. Szécsi, B. Benedek, and L. Szirmay-Kalos. Accelerating Animation Through Verification of Shooting Walks. In *Proceedings of SCCG*, pp. 231–238, ACM Press, 2003.

[Szir02] L. Szirmay-Kalos, V. Havran, B. Balázs, and L. Szécsi. On the Efficiency of Ray-shooting Acceleration Schemes. In A. Chalmers, Ed., *Proceedings of the 18th Spring Conference on Computer Graphics (SCCG 2002)*, pp. 89–98, ACM Siggraph, Budmerice, Slovakia, 2002.

[Szir97] L. Szirmay-Kalos and G. Márton. On the Limitations of Worst–case Optimal Ray Shooting Algorithms. In *Winter School of Computer Graphics 1997*, pp. 562–571, Feb. 1997. held at University of West Bohemia, Plzen, Czech Republic, 14-18 February 1997.

[Szir98a] L. Szirmay-Kalos and G. Márton. Worst-Case Versus Average Case Complexity of Ray-Shooting. *Computing*, Vol. 61, No. 2, pp. 103–131, 1998.

[Szir98b] L. Szirmay-Kalos and G. Márton. Analysis and construction of worst-case optimal ray shooting algorithms. *Computers and Graphics*, Vol. 22, No. 2–3, pp. 167–174, March 1998.

[Tell96] S. Teller, K. Bala, and J. Dorsey. Conservative Radiance Interpolants for Ray Tracing. In X. Pueyo and P. Schröder, Eds., *Eurographics Rendering Workshop 1996*, pp. 257–268, Eurographics, Springer Wien, New York City, NY, June 1996. ISBN 3-211-82883-4.

[Tell98] S. Teller and J. Allex. Frustum Casting for Progressive, Interactive Rendering. Tech. Rep. MIT LCS TR-740, MIT, Jan. 1998.

[Thir90] J.-P. Thirion. TRIES: Data Structures Based on Binary Representation for Ray Tracing. In C. E. Vandoni and D. A. Duce, Eds., *Eurographics '90*, pp. 531–541, North-Holland, Sep. 1990.

[Toth85] D. L. Toth. On Ray Tracing Parametric Surfaces. *Computer Graphics*, Vol. 19, No. 3, pp. 171–179, July 1985.

[VanW85] J. J. VanWijk. Ray Tracing Objects Defined By Sweeping a Sphere. *Computers and Graphics*, Vol. 9, No. 3, pp. 283–290, 1985.

[Voor90] D. Voorhies. Space-Filling Curves and a Measure of Coherence. In J. Arvo, Ed., *Graphics Gems*, pp. 257–262, Academic Press, San Diego, 1990.

[Voor91] D. Voorhies and D. Kirk. Ray-Triangle Intersection Using Binary Recursive Subdivision. In J. Arvo, Ed., *Graphics Gems II*, pp. 257–263, Academic Press, San Diego, 1991.

[Wald01] I. Wald, P. Slusallek, C. Benthin, and M. Wagner. Interactive Rendering with Coherent Ray Tracing. In A. Chalmers and T.-M. Rhyne, Eds., *EG 2001 Proceedings*, pp. 153–164, Blackwell Publishing, 2001.

[Wald03] I. Wald, C. Benthin, and P. Slusallek. Distributed Interactive Ray Tracing of Dynamic Scenes. In *Proceedings of the IEEE Symposium on Parallel and Large-Data Visualization and Graphics (PVG)*, 2003.

[Wald04] I. Wald. Realtime Ray Tracing and Interactive Global Illumination. *PhD thesis, Saarland University*, Januar 2004.

[Wald05] I. Wald and H.-P. Seidel. Interactive Ray Tracing of Point Based Models. In *Proceedings of 2005 Symposium on Point Based Graphics*, 2005.

[Wald06a] I. Wald and V. Havran. On building fast kd-trees for ray tracing, and on doing that in O(N log N). *Technical Report, SCI Institute, University of Utah, No UUSCI-2006-009 (submitted for publication)*, 2006.

[Wald06b] I. Wald, T. Ize, A. Kensler, A. Knoll, and S. G. Parker. Ray Tracing Animated Scenes using Coherent Grid Traversal. *ACM SIGGRAPH 2006*, 2006.

[Wang00] S.-W. Wang, Z.-C. Shih, and R.-C. Chang. An improved rendering technique for ray tracing Bézier and B-spline surfaces. *The Journal of Visualization and Computer Animation*, Vol. 11, No. 4, pp. 209–219, 2000.

[Wegh84] H. Weghorst, G. Hooper, and D. P. Greenberg. Improved Computational Methods for Ray Tracing. *ACM Transactions on Graphics*, Vol. 3, No. 1, pp. 52–69, Jan. 1984.

[Whan95] K. Y. Whang, J. W. Song, J. W. Chang, J. Y. Kim, W. S. Cho, C. M. Park, and I. Y. Song. Octree-R: an adaptive octree for efficient ray tracing. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No. 4, pp. 343–349, Dec. 1995. ISSN 1077-2626.

[Whit79] T. Whitted. An improved illumination model for shaded display. *Computer Graphics*, Vol. 13, No. 2, pp. 14–14, Aug. 1979.

[Woo90a] A. Woo. Fast Ray-Box Intersection. In A. S. Glassner, Ed., *Graphics Gems*, pp. 395–396, Academic Press, San Diego, 1990.

[Woo90b] A. Woo. Fast Ray-Polygon Intersection. In A. S. Glassner, Ed., *Graphics Gems*, p. 394, Academic Press, San Diego, 1990. includes code.

[Woo90c] A. Woo and J. Amanatides. Voxel Occlusion Testing: A Shadow Determination Accelerator for Ray Tracing. In *Proceedings of Graphics Interface '90*, pp. 213–220, May 1990.

[Woo92] A. Woo. Ray tracing polygons using spatial subdivision. In *Proceedings of Graphics Interface '92*, pp. 184–191, May 1992.

[Woo93] A. Woo. Efficient shadow computations in ray tracing. *IEEE Computer Graphics and Applications*, Vol. 13, No. 5, pp. 78–83, Sep. 1993.

[Wyvi86] G. Wyvill, T. L. Kunii, and Y. Shirai. Space Division for Ray Tracing in CSG (Constructive Solid Geometry). *IEEE Computer Graphics and Applications*, Vol. 6, No. 4, pp. 28–34, Apr. 1986.

[Yage92] R. Yagel, D. Cohen, and A. Kaufman. Discrete Ray Tracing. *IEEE Computer Graphics and Applications*, Vol. 12, No. 5, pp. 19–28, Sep. 1992.

[Yage97] R. Yagel and J. Meeker. Priority Driven Ray Tracing. *The Journal of Visualization and Computer Animation*, Vol. 8, No. 1, pp. 17–32, Jan. 1997.

[Zemv95] P. Zemčík and A. Chalmers. Optimised CSG Tree Evaluation for Space Subdivision. *Computer Graphics Forum*, Vol. 14, No. 2, pp. 139–146, June 1995.

[Zhen91] J. L. Zheng and C. B. Millham. Linear programming method for ray-convex polyhedron intersection. *Computers and Graphics*, Vol. 15, No. 2, pp. 195–204, 1991.

[Zwaa95] M. van der Zwaan, E. Reinhard, and F. W. Jansen. Pyramid Clipping for Efficient Ray Traversal. In *Proceedings of the Sixth Eurographics Rendering Workshop*, Dublin, Ireland, 1995.

Hidden Surface Removal

[Appe68] A. Appel. Some Techniques for Shading Machine Renderings of Solids. In *AFIPS 1968 Spring Joint Computer Conf.*, pp. 37–45, 1968.

[Berg93] M. de Berg. Generalized hidden surface removal. In A.-S. ACM-SIGGRAPH, Ed., *Proceedings of the 9th Annual Symposium on Computational Geometry (SCG '93)*, pp. 1–10, ACM Press, San Diego, CA, USA, May 1993.

[Berg97] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, New York, 1997.

[Bern94] M. W. Bern, D. P. Dobkin, D. Eppstein, and R. L. Grossman. Visibility with a Moving Point of View. *Algorithmica*, Vol. 11, No. 4, pp. 360–378, Apr. 1994.

[Bitt02] J. Bittner. Efficient Construction of Visibility Maps using Approximate Occlusion Sweep. In *Proceedings of Spring Conference on Computer Graphics (SCCG'02)*, pp. 163–171, Budmerice, Slovakia, 2002.

[Carp84] L. Carpenter. The A-buffer, an Antialiased Hidden Surface Method. In H. Christiansen, Ed., *Computer Graphics (SIGGRAPH '84 Proceedings)*, pp. 103–108, July 1984.

[Catm75] E. E. Catmull. Computer Display of Curved Surfaces. In *Proceedings of the IEEE Conference on Computer Graphics, Pattern Recognition, and Data Structure*, pp. 11–17, May 1975.

[Catm84] E. Catmull. An Analytic Visible Surface Algorithm for Independent Pixel Processing. In *Computer Graphics, Proceedings of Siggraph*, pp. 109–115, July 1984. Published as Computer Graphics, Proceedings of Siggraph, volume 18, number 3.

[Chen96] H. Chen and W. Wang. The Feudal Priority Algorithm on Hidden-Surface Removal. In *Proceedings of SIGGRAPH '96*, pp. 55–64, Aug. 1996.

[Chin89] N. Chin and S. Feiner. Near Real-Time Shadow Generation Using BSP Trees. In *Computer Graphics (Proceedings of SIGGRAPH '89)*, pp. 99–106, 1989.

[Chry92] Y. Chrysanthou and M. Slater. Computing dynamic changes to BSP trees. In *Computer Graphics Forum (EUROGRAPHICS '92 Proceedings)*, pp. 321–332, Sep. 1992.

[Dorw94] S. E. Dorward. A survey of object-space hidden surface removal. *Int. J. Comput. Geometry Appl*, Vol. 4, No. 3, pp. 325–362, 1994.

[Eric99] J. Erickson. Finite-resolution hidden surface removal. *CoRR*, Vol. cs.CG/9910017, 1999.

[Fuch80] H. Fuchs, Z. M. Kedem, and B. F. Naylor. On Visible Surface Generation by a Priori Tree Structures. pp. 124–133, July 1980.

[Gord91] D. Gordon and S. Chen. Front-to-back display of BSP trees. *IEEE Computer Graphics and Applications*, Vol. 11, No. 5, pp. 79–85, Sep. 1991.

[Gran92] C. W. Grant. *Visibility Algorithms in Image Synthesis*. PhD thesis, University of California, Davis, 1992.

[Gras99] J. Grasset, O. Terraz, J.-M. Hasenfratz, and D. Plemenos. Accurate Scene Display by Using Visibility Maps. In *Spring Conference on Computer Graphics and its Applications*, 1999.

[Gree93] N. Greene, M. Kass, and G. Miller. Hierarchical Z-Buffer Visibility. In *Computer Graphics (Proceedings of SIGGRAPH '93)*, pp. 231–238, 1993.

[Gree96] N. Greene. Hierarchical Polygon Tiling with Coverage Masks. In *Proceedings of SIGGRAPH '96*, pp. 65–74, Aug. 1996.

[Grov99] E. F. Grove, T. M. Murali, and J. S. Vitter. The Object Complexity Model for Hidden-Surface Removal. *Int. J. Comput. Geometry Appl*, Vol. 9, No. 2, pp. 207–217, 1999.

[Heck84] P. S. Heckbert and P. Hanrahan. Beam Tracing Polygonal Objects. *Computer Graphics (SIGGRAPH'84 Proceedings)*, Vol. 18, No. 3, pp. 119–127, July 1984.

[Jame98a] A. James and A. Day. Conflict Neutralization on Binary Space Partitioning (extended abstract). In *Eurographics UK Proceedings*, pp. 225 – 229, march 1998.

[Jame98b] A. James and A. Ray. The Priority Face Determination Tree for Hidden Surface Removal. *Computer Graphics Forum*, Vol. 17, No. 1, pp. 55–72, 1998. ISSN 1067-7055.

[Jone71] C. B. Jones. A New Approach to the 'Hidden Line' Problem. *Computer Journal*, Vol. 14, No. 3, pp. 232–237, Aug. 1971.

[McKe87] M. McKenna. Worst-case Optimal Hidden-Surface Removal. *ACM Transactions on Graphics*, Vol. 6, No. 1, pp. 19–28, Jan. 1987.

[Meye98] K. Meyer. A Nearly Output Sensitive Parallel Hidden Surface Removal Algorithm in Object Space. In V. Skala, Ed., *WSCG'98 Conference Proceedings*, 1998.

[Mill96] T. Miller. Hidden-Surfaces: Combining BSP Trees with Graph-Based Algorithms. Tech. Rep. CS-96-15, Department of Computer Graphics, Brown University, Apr. 1996.

[More95] P. Morer, A. M. Garcia-Alonso, and J. Flaquer. Optimization of a Priority List Algorithm for 3-D Rendering of Buildings. *Computer Graphics Forum*, Vol. 14, No. 4, pp. 217–227, Oct. 1995.

[Mulm89] K. Mulmuley. An Efficient Algorithm for Hidden Surface Removal. *Computer Graphics*, Vol. 23, No. 3, pp. 379–388, July 1989.

[Nava87] I. Navazo, J. Fontdecaba, and P. Brunet. Extended octtrees, between CSG trees and boundary representations. In G. Marechal, Ed., *Eurographics '87*, pp. 239–247, North-Holland, Aug. 1987.

[Nayl90a] B. Naylor. Binary Space Partitioning Trees as an Alternative Representation of Polytopes. *Computer–Aided Design*, pp. 250–252, 1990.

[Nayl90b] B. Naylor, J. Amanatides, and W. Thibault. Merging BSP Trees Yields Polyhedral Set Operations. *Computer Graphics (SIGGRAPH '90 Proceedings)*, Vol. 24, No. 4, pp. 115–124, Aug. 1990.

[Nayl92a] B. F. Naylor. Interactive solid geometry via partitioning trees. In *Proceedings of Graphics Interface '92*, pp. 11–18, May 1992.

[Nayl92b] B. F. Naylor. Partitioning tree image representation and generation from 3D geometric models. In *Proceedings of Graphics Interface '92*, pp. 201–212, May 1992.

[Nayl93] B. Naylor. Constructing good partition trees. In *Proceedings of Graphics Interface '93*, pp. 181–191, Toronto, Ontario, Canada, May 1993.

[Newe72] M. E. Newell, R. G. Newell, and T. L. Sancha. A Solution to the Hidden Surface Problem. In *Proceedings of ACM National Conference*, 1972.

[Over94] Overmars and Sharir. An Improved Technique for Output-Sensitive Hidden Surface Removal. *ALGRTHMICA: Algorithmica*, Vol. 11, 1994.

[Sada00] A. Sadagic and M. Slater. Dynamic Polygon Visibility Ordering for Head-Slaved Viewing in Virtual Environments. In *Computer Graphics Forum*, pp. 111–122, Eurographics Association, 2000.

[Shar92] M. Sharir and M. H. Overmars. A Simple Output-Sensitive Algorithm for Hidden Surface Removal. *ACM Transactions on Graphics*, Vol. 11, No. 1, pp. 1–11, Jan. 1992.

[Snyd98] J. Snyder and J. Lengyel. Visibility Sorting and Compositing without Splitting for Image Layer Decomposition. In *Computer Graphics (SIGGRAPH '98 Proceedings)*, pp. 219–230, Addison Wesley, July 1998.

[Stew98] A. J. Stewart and T. Karkanis. Computing the approximate visibility map, with applications to form factors and discontinuity meshing. In *Proceedings of the Ninth Eurographics Workshop on Rendering*, pp. 57–68, 1998.

[Suth74] I. E. Sutherland, R. F. Sproull, and R. A. Schumacker. A Characterization of Ten Hidden-Surface Algorithms. *Computing Surveys*, Vol. 6, No. 1, March 1974.

[Thib87] W. C. Thibault and B. F. Naylor. Set Operations on Polyhedra Using Binary Space Partitioning Trees. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, pp. 153–162, July 1987.

[Torr90] E. Torres. Optimization of the Binary Space Partition Algorithm (BSP) for the Visualization of Dynamic Scenes. In *Proceedings of Eurographics '90*, pp. 507–518, North-Holland, Sep. 1990.

[Tost91] D. Tost. An Algorithm of Hidden Surface Removal Based on Frame-to-Frame Coherence. In F. H. Post and W. Barth, Eds., *Proceedings of the 1991 European Computer Graphics Conference and EXhibition (EG-91)*, pp. 261–274, North-Holland, Amsterdam, Sep. 2–6 1991.

[Warn69] J. Warnock. A Hidden-Surface Algorithm for Computer Generated Half-Tone Pictures. Tech. Rep. TR 4–15, NTIS AD-733 671, University of Utah, Computer Science Department, 1969.

[Watk70] G. S. Watkins. A Real-Time Visible Surface Algorithm. Tech. Rep. UTECH-CSc-70-101, University of Utah, Salt Lake City, Utah, 1970.

[Weil77] K. Weiler and P. Atherton. Hidden Surface Removal Using Polygon Area Sorting. In *Computer Graphics (SIGGRAPH '77 Proceedings)*, pp. 214–222, July 1977.

[Weil80] K. Weiler. Polygon comparison using a graph representation. pp. 10–18, July 1980.

Visibility Culling

[Aila04] T. Aila and V. Miettinen. dPVS: An Occlusion Culling System for Massive Dynamic Environments. *IEEE Computer Graphics & Applications*, pp. 86–97, 2004.

[Aire90] J. M. Airey, J. H. Rohlf, and F. P. Brooks, Jr. Towards Image Realism with Interactive Update Rates in Complex Virtual Building Environments. In *Proceedings of Symposium on Interactive 3D Graphics*, pp. 41–50, ACM SIGGRAPH, March 1990.

[Alia97] D. G. Aliaga and A. A. Lastra. Architectural Walkthroughs Using Portal Textures. In *Proceedings of IEEE Visualization '97*, pp. 355–362, IEEE, Nov. 1997.

[Assa00] U. Assarsson and T. Möller. Optimized View Frustum Culling Algorithms for Bounding Boxes. *Journal of Graphics Tools*, Vol. 5, No. 1, pp. 9–22, 2000.

[Bart98] D. Bartz, M. Meissner, and T. Hüttner. Extending Graphics Hardware for Occlusion Queries in OpenGL. In *Proceedings of the 1998 Workshop on Graphics Hardware, Lisbon, Portugal*, pp. 97–104, 1998.

[Bart99] D. Bartz, M. Meißner, and T. Hüttner. OpenGL-assisted occlusion culling for large polygonal models. *Computers and Graphics*, Vol. 23, No. 5, pp. 667–679, Oct. 1999.

[Bitt01a] J. Bittner and V. Havran. Exploiting Coherence in Hierarchical Visibility Algorithms. *Journal of Visualization and Computer Animation, John Wiley & Sons*, Vol. 12, pp. 277–286, 2001.

[Bitt01b] J. Bittner, P. Wonka, and M. Wimmer. Visibility Preprocessing for Urban Scenes using Line Space Subdivision. In *Proceedings of Pacific Graphics (PG'01)*, pp. 276–284, IEEE Computer Society, Tokyo, Japan, 2001.

[Bitt02] J. Bittner. *Hierarchical Techniques for Visibility Computations*. PhD thesis, Czech Technical University in Prague, Oct. 2002.

[Bitt03] J. Bittner and P. Wonka. Visibility in Computer Graphics. *Environment and Planning B: Planning and Design*, Vol. 30, No. 5, pp. 729–756, Sep. 2003.

[Bitt04] J. Bittner, M. Wimmer, H. Piringer, and W. Purgathofer. Coherent Hierarchical Culling: Hardware Occlusion Queries Made Useful. *Computer Graphics Forum (Proceedings of Eurographics '04)*, No. 3, 2004.

[Bitt98] J. Bittner, V. Havran, and P. Slavík. Hierarchical Visibility Culling with Occlusion Trees. In *Proceedings of Computer Graphics International '98 (CGI'98)*, pp. 207–219, IEEE, 1998.

[Blai98] M. Blais and P. Poulin. Sampling Visibility in Three-Space. In *Proc. of the 1998 Western Computer Graphics Symposium*, pp. 45–52, Apr. 1998.

[Brun01] P. Brunet, I. Navazo, C. Saona-Vázquez, and J. Rossignac. Hoops: 3D Curves as Conservative Occluders for Cell-Visibility. *Computer Graphics Forum (Proceedings of Eurographics '01)*, No. 3, 2001.

[Carl00] I. N. Carlos Andújar, Carlos Saona-Vázquez and P. Brunet. Integrating Occlusion Culling with Levels of Detail through Hardly-Visible Sets. In *Computer Graphics Forum (Proceedings of Eurographics '00)*, pp. 499–506, 2000.

[Caza97] F. Cazals and M. Sbert. Some integral geometry tools to estimate the complexity of 3D scenes. Tech. Rep. RR-3204, The French National Institue for Research in Computer Science and Control (INRIA), July 1997.

[Cham96] B. Chamberlain, T. DeRose, D. Lischinski, D. Salesin, and J. Snyder. Fast rendering of complex environments using a spatial hierarchy. In *Proceedings of Graphics Interface '96*, pp. 132–141, May 1996.

[Chry98a] Y. Chrysanthou, D. Cohen-Or, and D. Lischinski. Fast Approximate Quantitative Visibility for Complex Scenes. In *Proceedings of Computer Graphics International '98 (CGI'98)*, pp. 23–31, IEEE, NY, Hannover, Germany, June 1998.

[Chry98b] Y. Chrysanthou, D. Cohen-Or, and E. Zadicario. Viewspace Partitioning of Densely Occluded Scenes. Abstract of a video presentation, at the 13th Annual ACM Symposium on Computational Geometry, Minnesota, pages 413–414, June 1998.

[Clar76] J. H. Clark. Hierarchical Geometric Models for Visible Surface Algorithms. *Communications of the ACM*, Vol. 19, No. 10, pp. 547–554, Oct. 1976.

[Cohe03] D. Cohen-Or, Y. Chrysanthou, C. Silva, and F. Durand. A Survey of Visibility for Walkthrough Applications. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 9, No. 3, pp. 412–431, 2003.

[Cohe95] D. Cohen-Or and A. Shaked. Visibility and Dead-Zones in Digital Terrain Maps. *Computer Graphics Forum*, Vol. 14, No. 3, pp. C/171–C/180, Sep. 1995.

[Cohe98a] D. Cohen-Or, G. Fibich, D. Halperin, and E. Zadicario. Conservative Visibility and Strong Occlusion for Viewspace Partitioning of Densely Occluded Scenes. In *Computer Graphics Forum (Eurographics '98 Proceedings)*, pp. 243–253, 1998.

[Cohe98b] D. Cohen-Or and E. Zadicario. Visibility Streaming for Network-based Walkthroughs. In *Proceedings of Graphics Interface '98*, pp. 1–7, June 1998.

[Cole89] R. Cole and M. Sharir. Visibility Problems for Polyhedral Terrains. *Journal of Symbolic Computation*, Vol. 7, No. 1, pp. 11–30, Jan. 1989.

[Coor96a] S. Coorg and S. Teller. A Spatially and Temporally Coherent Object Space Visibility Algorithm. Tech. Rep. TM-546, Department of Computer Graphics, MIT, Feb. 1996.

[Coor96b] S. Coorg and S. Teller. Temporally Coherent Conservative Visibility. In *Proceedings of the Twelfth Annual ACM Symposium on Computational Geometry*, Philadelphia, PA, May 1996.

[Coor97] S. Coorg and S. Teller. Real-Time Occlusion Culling for Models with Large Occluders. In *Proceedings of the Symposium on Interactive 3D Graphics*, pp. 83–90, ACM Press, Apr. 1997.

[Deco03] X. Décoret, G. Debunne, and F. Sillion. Erosion Based Visibility Preprocessing. In *Proceedings of the EG Symposium on Rendering*, Eurographics, Eurographics Association, 2003.

[Down01] L. Downs, T. Möller, and C. H. Séquin. Occlusion Horizons for Driving through Urban Scenes. In *Symposium on Interactive 3D Graphics*, pp. 121–124, ACM SIGGRAPH, 2001.

[Dugu02] F. Duguet and G. Drettakis. Robust Epsilon Visibility. In *Computer Graphics (SIGGRAPH '02 Proceedings)*, pp. 567–575, ACM Press/ACM SIGGRAPH, 2002.

[Dura00] F. Durand, G. Drettakis, J. Thollot, and C. Puech. Conservative Visibility Preprocessing Using Extended Projections. In *Computer Graphics (SIGGRAPH '00 Proceedings)*, pp. 239–248, 2000.

[Dura96] F. Durand, G. Drettakis, and C. Puech. The 3D Visibility Complex: A New Approach to the Problems of Accurate Visibility. In *Proceedings of Eurographics Workshop on Rendering*, pp. 245–256, Eurographics, Springer Wein, June 1996.

[Dura97] F. Durand, G. Drettakis, and C. Puech. The Visibility Skeleton: A Powerful and Efficient Multi-Purpose Global Visibility Tool. In *Computer Graphics (SIGGRAPH '97 Proceedings)*, pp. 89–100, 1997.

[Dura99] F. Durand. *3D Visibility: Analytical Study and Applications*. PhD thesis, Universite Joseph Fourier, Grenoble, France, July 1999.

[Egge93] D. W. Eggert, K. W. Bowyer, C. R. Dyer, H. I. Christensen, and D. B. Goldgof. The Scale Space Aspect Graph. *Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11, pp. 1114–1130, Nov. 1993.

[Flor94] L. D. Floriani and P. Magillo. Visibility Algorithms on Triangulated Digital Terrain Models. In *International Journal of Geographical Information Systems*, pp. 13–41, Taylor & Francis, 1994.

[Flor95] L. D. Floriani and P. Magillo. Horizon computation on a hierarchical triangulated terrain model. *The Visual Computer*, Vol. 11, No. 3, pp. 134–149, 1995.

[Fuch80] H. Fuchs, Z. M. Kedem, and B. F. Naylor. On Visible Surface Generation by a Priori Tree Structures. In *Computer Graphics (SIGGRAPH '80 Proceedings)*, pp. 124–133, July 1980.

[Funk93] T. A. Funkhouser. *Database and Display Algorithms for Interactive Visualization of Architectural Models*. PhD thesis, CS Division, UC Berkeley, 1993.

[Geor95] C. Georges. Obscuration Culling on Parallel Graphics Architectures. Tech. Rep. TR95-017, Department of Computer Science, University of North Carolina, Chapel Hill, 1995.

[Gigu90] Z. Gigus and J. Malik. Computing the aspect graph for line drawings of polyhedral objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 2, pp. 113–122, Feb. 1990.

[Gord91] D. Gordon and S. Chen. Front-to-back display of BSP trees. *IEEE Computer Graphics and Applications*, Vol. 11, No. 5, pp. 79–85, Sep. 1991.

[Govi03] N. K. Govindaraju, A. Sud, S.-E. Yoon, and D. Manocha. Interactive visibility culling in complex environments using occlusion-switches. In *Proceedings of the 2003 Symposium on Interactive 3D graphics*, pp. 103–112, ACM Press, 2003.

[Gree93] N. Greene, M. Kass, and G. Miller. Hierarchical Z-Buffer Visibility. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, pp. 231–238, 1993.

[Gree94] N. Greene. Detecting Intersection of a Rectangular Solid and a Convex Polyhedron. In P. Heckbert, Ed., *Graphics Gems IV*, pp. 74–82, Academic Press, Boston, MA, 1994.

[Gree95] D. Green and D. Hatch. Fast Polygon-Cube Intersection Testing. In A. W. Paeth, Ed., *Graphics Gems V*, pp. 375–379, Academic Press, Boston MA, 1995.

[Gree96] N. Greene. Hierarchical Polygon Tiling with Coverage Masks. In H. Rushmeier, Ed., *Computer Graphics (SIGGRAPH '96 Proceedings)*, pp. 65–74, Addison Wesley, Aug. 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[Hain94] E. A. Haines and J. R. Wallace. Shaft Culling for Efficient Ray-Traced Radiosity. In *Photorealistic Rendering in Computer Graphics (Proceedings of Eurographics Workshop on Rendering '94)*, Springer-Verlag, 1994.

[Haum03] D. Haumont, O. Debeir, and F. Sillion. Volumetric Cell-and-Portal Generation. *Computer Graphics Forum (Proceedings of Eurographics '03)*, No. 3, 2003.

[Hey01] H. Hey, R. F. Tobler, and W. Purgathofer. Real-Time Occlusion Culling with a Lazy Occlusion Grid. In *Rendering Techniques (Proceedings of Eurographics Workshop on Rendering '00)*, pp. 217–222, 2001.

[Hink96] A. Hinkenjann and H. Müller. Hierarchical Blocker Trees for Global Visibility Calculation. Research Report 621/1996, University of Dortmund, Aug. 1996.

[Hink98] A. Hinkenjann and H. Muller. Determining Visibility between Extended Objects. In *Proceedings of Computer Graphics International '98 (CGI'98)*, pp. 23–31, IEEE, NY, Hannover, Germany, June 1998.

[Hua02] W. Hua, H. Bao, Q. Peng, and A. R. Forrest. The global occlusion map: a new occlusion culling approach. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 155–162, ACM Press, 2002.

[Huds97] T. Hudson, D. Manocha, J. Cohen, M. Lin, K. Hoff, and H. Zhang. Accelerated Occlusion Culling Using Shadow Frusta. In *Proceedings of ACM Symposium on Computational Geometry*, pp. 1–10, 1997.

[Ione98] A. Iones, S. Zhukov, and A. Krupkin. On Optimality of OBBs for Visibility Tests for Frustrum Culling, Ray Shooting and Collision Detection. In *Proceedings of Computer Graphics International '98 (CGI'98)*, pp. 256–263, IEEE, NY, Hannover, Germany, June 1998.

[Klos00] J. T. Klosowski and C. T. Silva. The Prioritized-Layered Projection Algorithm for Visible Set Estimation. In H. Hagen and D. S. Ebert, Eds., *IEEE Transactions on Visualization and Computer Graphics*, pp. 108–123, IEEE Computer Society, 2000.

[Klos01] J. T. Klosowski and C. T. Silva. Efficient Conservative Visibility Culling Using the Prioritized-Layered Projection Algorithm. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 7, No. 4, pp. 365–379, Oct. 2001.

[Kolt00] V. Koltun, Y. Chrysanthou, and D. Cohen-Or. Virtual Occluders: An Efficient Intermediate PVS Representation. In *Rendering Techniques (Proceedings of Eurographics Workshop on Rendering '00)*, pp. 59–70, 2000.

[Kolt01] V. Koltun, Y. Chrysanthou, and D. Cohen-Or. Hardware-Accelerated From-Region Visibility Using a Dual Ray Space. In *Rendering Techniques (Proceedings of Eurographics Workshop on Rendering '01)*, pp. 205–216, 2001.

[Kuma96a] S. Kumar and D. Manocha. Hierarcical Visibility Culling for Spline Models. In *Proceedings of Graphics Interface '96*, pp. 142–150, Canadian Human-Computer Communications Society, May 1996.

[Kuma96b] S. Kumar, D. Manocha, W. Garrett, and M. Lin. Hierarchical Back-Face Computation. In *Rendering Techniques (Proccedings of Eurographics Workshop on Rendering '96)*, pp. 235–244, Springer Wein, June 1996.

[Kuma97] S. Kumar, D. Manocha, H. Zhang, and K. E. H. III. Accelerated Walkthrough of Large Spline Models. In *Symposium on Interactive 3D Graphics, 1997*, pp. 91–102, 190, 1997.

[Lain05] S. Laine. A General Algorithm for Output-Sensitive Visibility Preprocessing. In *Proceedings of the 2005 Symposium on Interactive 3D graphics*, ACM Press, 2005.

[Lern03] A. Lerner, Y. Chrysanthou, and D. Cohen-Or. Breaking the Walls: Scene Partitioning and Portal Creation. In *Proceedings of Pacific Graphics (PG'03)*, p. 303, IEEE Computer Society, 2003.

[Leyv03] T. Leyvand, O. Sorkine, and D. Cohen-Or. Ray Space Factorization for From-Region Visibility. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, 2003.

[Lisc92] D. Lischinski, F. Tampieri, and D. P. Greenberg. Discontinuity meshing for accurate radiosity. *IEEE Computer Graphics and Applications*, Vol. 12, No. 6, pp. 25–39, Nov. 1992.

[Lloy02] B. Lloyd and P. Egbert, K. Horizon Occlusion Culling for Real-time Rendering of Hierarchical Terrains. In *Proceedings of the conference on Visualization '02*, pp. 403–409, 2002.

[Lowe05] N. Lowe and A. Datta. A New Technique for Rendering Complex Portals. *IEEE Transaction on Visualization and Computer Graphics*, pp. 81–90, 2005.

[Lueb95] D. Luebke and C. Georges. Portals and Mirrors: Simple, Fast Evaluation of Potentially Visible Sets. In *Proceedings of Symposium on Interactive 3D Graphics '95*, pp. 105–106, ACM SIGGRAPH, Apr. 1995.

[McMi97] L. McMillan. An Image-Based Approach to Three-Dimensional Computer Graphics. Ph.D. Thesis TR97-013, University of North Carolina, Chapel Hill, May 1997.

[Moll02] T. Möller and E. Haines. *Real-Time Rendering, 2nd edition*. A. K. Peters, 2002.

[Mora05] F. Mora, L. Aveneau, and M. Mériaux. Coherent and Exact Polygon-to-Polygon Visibility. In *Proceedings of Winter School on Computer Graphics 2005*, pp. 87–94, 2005.

[Nava03] I. Navazo, J. Rossignac, J. Jou, and R. Shariff. ShieldTester: Cell-to-Cell Visibility Test for Surface Occluders. *Computer Graphics Forum (Proceedings of Eurographics '03)*, Vol. 22, No. 3, pp. 291–302, 2003.

[Nayl92] B. F. Naylor. Partitioning tree image representation and generation from 3D geometric models. In *Proceedings of Graphics Interface '92*, pp. 201–212, May 1992.

[Nire02] S. Nirenstein, E. Blake, and J. Gain. Exact From-Region Visibility Culling. In *Proceedings of Eurographics Workshop on Rendering '02*, pp. 199–210, 2002.

[Nire04] S. Nirenstein and E. Blake. Hardware Accelerated Aggressive Visibility Preprocessing using Adaptive Sampling. In *Rendering Technqiues 2004: Proceedings of the 15th symposium on Rendering*, pp. 207–216, Eurographics Association, 2004.

[Orti96] R. Orti, S. Riviere, F. Durand, and C. Puech. Using the Visibility Complex for Radiosity Computation. In *Lecture Notes in Computer Science (Applied Computational Geometry: Towards Geometric Engineering)*, pp. 177–190, Springer-Verlag, Berlin, Germany, May 1996.

[Plan86] W. H. Plantinga and C. R. Dyer. An Algorithm for Constructing the Aspect Graph. In *27th Annual Symposium on Foundations of Computer Science*, pp. 123–131, IEEE Computer Society Press, Los Angeles, Ca., USA, Oct. 1986.

[Plan90] H. Plantinga and C. Dyer. Visibility, Occlusion, and the Aspect Graph. *International Journal of Computer Vision*, Vol. 5, No. 2, pp. 137–160, 1990.

[Plan93] H. Plantinga. Conservative visibility preprocessing for efficient walkthroughs of 3D scenes. In *Proceedings of Graphics Interface '93*, pp. 166–173, Toronto, Ontario, Canada, May 1993.

[Pocc93] M. Pocchiola and G. Vegter. The visibility complex. In *Proceedings of ACM Symposium on Computational Geometry*, pp. 328–337, 1993.

[Pu98] F.-T. Pu. *Data Structures for Global Illumination and Visibility Queries in 3-Space*. PhD thesis, University of Maryland, College Park, MD, 1998.

[Rivi97] S. Rivière. Walking in the Visibility Complex with Applications to Visibility Polygons and Dynamic Visibility. In *Proceedings of 9th Canadian Conference on Computational Geometry*, pp. 147–152, 1997.

[Rohl94] J. Rohlf and J. Helman. IRIS Performer: A High Performance Multiprocessing Toolkit for Real–Time 3D Graphics. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pp. 381–395, July 1994.

[Saon99] C. Saona-Vázquez, I. Navazo, and P. Brunet. The visibility octree: a data structure for 3D navigation. *Computers and Graphics*, Vol. 23, No. 5, pp. 635–643, Oct. 1999.

[Scha00] G. Schaufler, J. Dorsey, X. Decoret, and F. X. Sillion. Conservative Volumetric Visibility with Occluder Fusion. In *Computer Graphics (SIGGRAPH '00 Proceedings)*, pp. 229–238, 2000.

[Schu69] R. A. Schumacker, R. Brand, M. Gilliland, and W. Sharp. Study for Applying Computer-Generated Images to Visual Simulation. Tech. Rep. AFHRL–TR–69–14, U.S. Air Force Human Resources Laboratory, 1969.

[Shar92] M. Sharir and M. H. Overmars. A Simple Output-Sensitive Algorithm for Hidden Surface Removal. *ACM Transactions on Graphics*, Vol. 11, No. 1, pp. 1–11, Jan. 1992.

[Slat97] M. Slater and Y. Chrysanthou. View Volume Culling Using a Probabilistic Caching Scheme. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST '97)*, pp. 71–78, Lausanne, Switzerland, Sep. 1997.

[Sole96] C. Soler and F. Sillion. Accurate Error Bounds for Multi-Resolution Visibility. In *rendering Techniques (Proceedings of Eurographics Workshop on Rendering '96)*, pp. 133–142, Springer Wein, June 1996.

[Stan04] D. Staneker, D. Bartz, and W. Straßer. Occlusion Culling in OpenSG PLUS. *Computer & Graphics*, No. TR-186-2-03-03, pp. 87–92, 2004.

[Stew93] A. J. Stewart and S. Ghali. An Output Sensitive Algorithm for the Computation of Shadow Boundaries. In *Proceedings of Canadian Conference on Computational Geometry*, pp. 291–296, Aug. 1993.

[Stew94] A. J. Stewart and S. Ghali. Fast Computation of Shadow Boundaries Using Spatial Coherence and Backprojections. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pp. 231–238, 1994.

[Stew97] A. J. Stewart. Hierarchical Visibility in Terrains. In *Rendering Techniques (Proceedings of Eurographics Workshop on Rendering '97)*, pp. 217–228, 1997.

[Stew98] A. J. Stewart. Fast Horizon Computation at All Points of a Terrain With Visibility and Shading Applications. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 4, No. 1, pp. 82–93, Jan. 1998.

[Suda96] O. Sudarsky and C. Gotsman. Output-Sensitive Visibility Algorithms for Dynamic Scenes with Applications to Virtual Reality. *Computer Graphics Forum*, Vol. 15, No. 3, pp. C249–C258, Sep. 1996.

[Tell91] S. J. Teller and C. H. Séquin. Visibility preprocessing for interactive walkthroughs. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, pp. 61–69, 1991.

[Tell92a] S. J. Teller. Computing the antipenumbra of an area light source. *Computer Graphics (SIGGRAPH '92 Proceedings)*, Vol. 26, No. 2, pp. 139–148, July 1992.

[Tell92b] S. J. Teller. *Visibility Computations in Densely Occluded Polyhedral Environments*. PhD thesis, CS Division, UC Berkeley, Oct. 1992. Tech. Report UCB/CSD-92-708.

[Tell93] S. Teller and P. Hanrahan. Global Visibility Algorithms for Illumination Computations. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, pp. 239–246, 1993.

[Tell94] S. Teller, C. Fowler, T. Funkhouser, and P. Hanrahan. Partitioning and Ordering Large Radiosity Computations. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pp. 443–450, July 1994.

[Tell98] S. Teller and J. Alex. Frustum casting for progressive, interactive rendering. Tech. Rep. MIT LCS TR–740, MIT, January 1998.

[Tell99] S. Teller and M. Hohmeyer. Determining the Lines Through Four Lines. *Journal of Graphics Tools: JGT*, Vol. 4, No. 3, pp. 11–22, 1999.

[Vegt90] G. Vegter. The Visibility Diagram: a Data Structure for Visibility Problems and Motion Planning. In *In Proceedings of the 2nd Scandinavian Workshop on Algorithm Theory (SWAT '90)*, pp. 97–110, Springer, 1990.

[Wand01] M. Wand, M. Fischer, I. Peter, F. M. auf der Heide, and W. Straßer. The Randomized z-Buffer Algorithm: Interactive Rendering of Highly Complex Scenes. In *Computer Graphics (SIGGRAPH '01 Proceedings)*, pp. 361–370, 2001.

[Wang98] Y. Wang, H. Bao, and Q. Peng. Accelerated Walkthroughs of Virtual Environments Based on Visibility Preprocessing and Simplification. In *Computer Graphics Forum (Proceedings of Eurographics '98)*, pp. 187–194, 1998.

[Welz85] E. Welzl. Constructing the Visibility Graph for n-Line Segments in $O(n^2)$ Time. *Information Processing Letters*, Vol. 20, No. 4, pp. 167–171, May 1985.

[Wonk00] P. Wonka, M. Wimmer, and D. Schmalstieg. Visibility Preprocessing with Occluder Fusion for Urban Walkthroughs. In *Rendering Techniques (Proceedings of Eurographics Workshop on Rendering '00)*, pp. 71–82, 2000.

[Wonk01] P. Wonka, M. Wimmer, and F. X. Sillion. Instant Visibility. In *Computer Graphics Forum (Proceedings of Eurographics '01)*, pp. 411–421, Blackwell Publishing, 2001.

[Wonk99] P. Wonka and D. Schmalsteig. Occluder Shadows for Fast Walkthroughs of Urban Environments. *Computer Graphics Forum (Proceedings of Eurographics '99)*, Vol. 18, No. 3, pp. 51–60, Sep. 1999.

[Yage95] R. Yagel and W. Ray. Visibility Computation for Efficient Walkthrough of Complex Environments. *Presence: Teleoperators and Virtual Environments*, Vol. 5, No. 1, 1995.

[Zhan97a] H. Zhang and K. E. Hoff III. Fast Backface Culling Using Normal Masks. In *Proceedings of 1997 Symposium on Interactive 3D Graphics*, pp. 103–106, ACM SIGGRAPH, Apr. 1997.

[Zhan97b] H. Zhang, D. Manocha, T. Hudson, and K. E. Hoff III. Visibility Culling Using Hierarchical Occlusion Maps. In *Computer Graphics (Proceedings of SIGGRAPH '97)*, pp. 77–88, 1997.

Photon Mapping, Irradiance Caching, and Ray Maps

[Arya98] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions. *Journal of the ACM*, Vol. 45, No. 6, pp. 891–923, 1998.

[Beka03] P. Bekaert, P. Slussalek, R. Cools, V. Havran, and H.-P. Seidel. A Custom Designed Density Estimator for Light Transport. Tech. Rep. MPI-I-2003-4-004, Max-Planck-Institut fur Informatik, Saarbrcken Germany, March 2003.

[Bent79] J. L. Bentley. Multidimensional Binary Search Trees in Database Applications. In *IEEE Trans. on Soft. Eng.*, pp. 333–340, 1979.

[Camm02] M. Cammarano and H. W. Jensen. Time Dependent Photon Mapping. In *Rendering Techniques 2002*, pp. 135–144, June 2002.

[Chri02] P. H. Christensen. Photon Mapping Tricks. *SIGGRAPH Course Notes*, Vol. 43, pp. 93–121, 2002.

[Chri04] P. H. Christensen and D. Batali. An Irradiance Atlas for Global Illumination in Complex Production Scenes. In *Rendering Techniques 2004*, pp. 133–141, Proceedings of Eurographics Symposium on Rendering, 2004.

[Chri99] P. H. Christensen. Faster Photon Map Global Illumination. In *Journal of Graphics Tools*, pp. 1–10, 1999.

[Dick00] M. Dickeerson, C. Duncan, and M. Goodrich. Kd-trees are better when cut at the longest side. In *Proceedings of the 8th Annual European Symposium on Algorithms*, pp. 179–190, 2000.

[Dret97] G. Drettakis and F. X. Sillion. Interactive Update of Global Illumination using a Line-Space Hierarchy. In *Computer Graphics*, pp. 57–64, ACM SIGGRAPH Proceedings, 1997.

[Gaut04] P. Gautron, J. Křivánek, S. Pattanaik, and K. Bouatouch. A Novel Hemispherical Basis for Accurate and Efficient Rendering. In *Rendering Techniques 2004*, Eurographics Symposium on Rendering, 2004.

[Gaut05] P. Gautron, J. Křivánek, K. Bouatoch, and S. Pattanaik. Radiance Cache Splatting: A GPU-Friendly Global Illumination Algorithm. In *Rendering Techniques 2005*, pp. 55–64, Eurographics Symposium on Rendering, 2005.

[Greg98] G. Greger, P. Shirley, P. M. Hubbard, and D. P. Greenberg. The Irradiance Volume. *IEEE Comput. Graph. and Appl.*, Vol. 18, No. 2, pp. 32–43, 1998.

[Haev04] W. V. Haevre, F. D. Fiore, P. Bekaert, and F. V. Reeth. A ray density estimation approach to take into account environment illumination in plant growth simulation. In *SCCG '04: Proceedings of the 20th spring conference on Computer graphics*, pp. 121–131, ACM Press, New York, NY, USA, 2004.

[Hans02] D. Hansson and N. Harrysson. *Fast Photon Mapping using Grids*. Master's thesis, 2002.

[Havr05a] V. Havran, J. Bittner, R. Herzog, and H.-P. Seidel. Ray Maps for Global Illumination. In *Rendering Techniques 2005*, pp. 43–54, Eurographics Symposium on Rendering, 2005.

[Havr05b] V. Havran, R. Herzog, and H.-P. Seidel. Fast Final Gathering via Reverse Photon Mapping. In *Computer Graphics Forum*, pp. 323–333, Proceedings of Eurographics, 2005.

[Herz05] R. Herzog. *Advanced Density Estimation Techniques for Global Illumination*. Master's thesis, Max-Planck-Institute for Informatics, Saarbruecken, Germany, Oct. 2005.

[Hey01] H. Hey and W. Purgathofer. Global Illumination with Photon Mapping Compensation. Tech. Rep. TR-186-2-01-04, Vienna University of Technology, January 2001.

[Jens01] H. W. Jensen. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Natick, 2001.

[Jens95] H. W. Jensen and N. J. Christensen. Photon Maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects. In *Computers & Graphics*, pp. 215–224, 1995.

[Jens96] H. W. Jensen. Global Illumination using Photon Maps. In *Rendering Techniques '96*, pp. 21–30, Proceedings of the Seventh Eurographics Workshop on Rendering, 1996.

[Kvri05] J. Křivánek. *Radiance Caching for Global Illumination Computation on Glossy Surfaces*. PhD thesis, Université de Rennes 1 and Czech Technical University in Prague, December 2005.

[Lars03] B. D. Larsen and N. J. Christensen. Optimizing Photon Mapping using Multiple Photon Maps for Irradiance Estimates. In *WSCG Poster Proceedings, Plzen, Czech Republic*, pp. 77–80, 2003.

[Last02] M. Lastra, C. Urena, J. Revelles, and R. Montes. A Particle-Path Based Method for Monte Carlo Density Estimation. In *In Poster Proceeding of the 13th Eurographics Workshop on Rendering*, pp. 33–40, June 2002.

[Lavi01] F. Lavignotte and M. Paulin. A New Approach of Density Estimation for Global Illumination. In *Proceedings of Graphicon*, University of Nizhny Novgorod, Nizhny Novgorod, Russia, 2001.

[Lavi02] F. Lavignotte and M. Paulin. A New Approach of Density Estimation for Global Illumination. In *Proceedings of WSCG 2002*, pp. 263–270, 2002.

[Lavi03] F. Lavignotte and M. Paulin. Scalable Photon Splatting for Global Illumination. In *Graphite 2003 (International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia), Melbourne, Australia.*, pp. 1–11, ACM SIGGRAPH, 2003.

[Mane99] S. Maneewongvatana and D. Mount. It's okay to be skinny, if your friends are fat. In *Proceedings of the 4th Annual CGC Workshop on Computational Geometry*, 1999.

[Shir95] P. Shirley, B. Wade, P. M. Hubbard, D. Zareski, B. Walter, and D. P. Greenberg. Global Illumination via Density Estimation. In *Rendering Techniques '95*, pp. 219–230, 1995.

[Smyk05] M. Smyk, S. Kinuwaki, R. Durikovic, and K. Myszkowski. Temporally Coherent Irradiance Caching for High Quality Animation Rendering. In *Proceedings of Eurographics*, pp. 401–412, 2005.

[Tabe04] E. Tabellion and A. Lamorlette. An Approximate Global Illumination System for Computer Generated Films. In *ACM Trans. Graph.*, pp. 469–476, 2004.

[Talb00] D. A. Talbert and D. Fisher. An Empirical Analysis of Techniques for Constructing and Searching k-Dimensional Trees. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 26–33, 2000.

[Tawa04] T. Tawara, K. Myszkowski, K. Dmitriev, V. Havran, C. Damez, and H.-P. Seidel. Exploiting Temporal Coherence in Global Illumination. In *SCCG '04: Proceedings of the 20th Spring Conference on Computer Graphics*, pp. 23–33, 2004.

[Wald04] I. Wald, J. Gnther, and P. Slusallek. Balancing Considered Harmful - Faster Photon Mapping Using the Voxel Heuristic. In *Computer Graphics Forum*, Proceedings of Eurographics, 2004.

[Walt98] B. J. Walter. *Density Estimation Techniques for Global Illumination*. PhD thesis, Program of Computer Graphics, Cornell University, Ithaca, NY, August 1998.

[Ward88] G. J. Ward, F. M. Rubinstein, and R. D. Clear. A Ray Tracing Solution for Diffuse Interreflection. In *Computer Graphics (ACM SIGGRAPH '88 Proceedings)*, pp. 85–92, August 1988.

[Ward92] G. J. Ward and P. Heckbert. Irradiance Gradients. In *Third Eurographics Workshop on Rendering*, pp. 85–98, Bristol, UK, May 1992.

Other Publications on Rendering with Sorting and/or Searching

[Camm02] M. Cammarano and H. W. Jensen. Time dependent photon mapping. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering*, pp. 135–144, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2002.

[Glas88] A. S. Glassner. Spacetime ray tracing for animation. *IEEE Computer Graphics and Applications*, Vol. 8, No. 2, pp. 60–70, March 1988.

[Havr03a] V. Havran, J. Bittner, and H.-P. Seidel. Exploiting Temporal Coherence in Ray Casted Walkthroughs. In K. I. Joy, Ed., *Proceedings of the 19th Spring Conference on Computer Graphics 2003 (SCCG 03)*, pp. 164–172, ACM, Budmerice, Slovakia, April 2003.

[Havr03b] V. Havran, C. Damez, K. Myszkowski, and H.-P. Seidel. An Efficient Spatio-Temporal Architecture for Animation Rendering. In P. Christensen and D. Cohen-Or, Eds., *Rendering Techniques 2003 : 14th Eurographics Workshop on Rendering*, pp. 106–117, Association of Computing Machinery (ACM), ACM, Leuven, Belgium, June 2003.

[Havr03c] V. Havran, K. Dmitriev, and H.-P. Seidel. Goniometric Diagram Mapping for Hemisphere. Short Presentations (Eurographics 2003), 2003.

[Lain05] S. Laine and T. Aila. Hierarchical Penumbra Casting. *Computer Graphics Forum*, Vol. 24, No. 3, pp. 313–322, 2005.

[Laz04] I. Laznyi and L. Szirmay-Kalos. Speeding up the Virtual Light Sources Algorithm. In *SCCG '04: Proceedings of the 20th spring conference on Computer graphics*, pp. 112–120, ACM Press, New York, NY, USA, 2004.

[Paqu98] E. Paquette, P. Poulin, and G. Drettakis. A Light Hierarchy for Fast Rendering of Scenes with Many Lights. In N. Göbel and F. N. F. (guest editor), Eds., *Computer Graphics Forum (Eurographics '98 Conference Proceedings)*, pp. 63–74, Eurographics, Sep 1998. held in Li.

[Stei05] J. Steinhurst, G. Coombe, and A. Lastra. Reordering for cache conscious photon mapping. In *GI '05: Proceedings of the 2005 conference on Graphics interface*, pp. 97–104, Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005.

[Szir05] L. Szirmay-Kalos, B. Aszdi, I. Laznyi, and M. Premecz. Approximate Ray-Tracing on the GPU with Distance Impostors. *Computer Graphics Forum (Proceedings of Eurographics 2005)*, Vol. 24, No. 3, pp. 685–704, August 2005.

[Walt05] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, and D. P. Greenberg. Lightcuts: a scalable approach to illumination. *ACM Trans. Graph.*, Vol. 24, No. 3, pp. 1098–1107, 2005.

[Walt06] B. Walter, A. Arbree, K. Bala, and D. P. Greenberg. Multidimensional Lightcuts. *ACM Trans. Graph.*, Vol. 25, No. 3, 2006. to appear.

[Ward91] G. Ward. Adaptive shadow testing for ray tracing. In *Eurographics Workshop on Rendering*, pp. 11–20, May 1991.

[Webe04] M. Weber, M. Milch, K. Myszkowski, K. Dmitriev, P. Rokita, and H.-P. Seidel. Spatio-Temporal Photon Density Estimation Using Bilateral Filtering. In D. Cohen-Or, L. Jain, and N. Magnenat-Thalmann, Eds., *Computer Graphics International (CGI 2004)*, pp. 120–127, IEEE, Crete, Greece, 2004.

