Ray Tracing

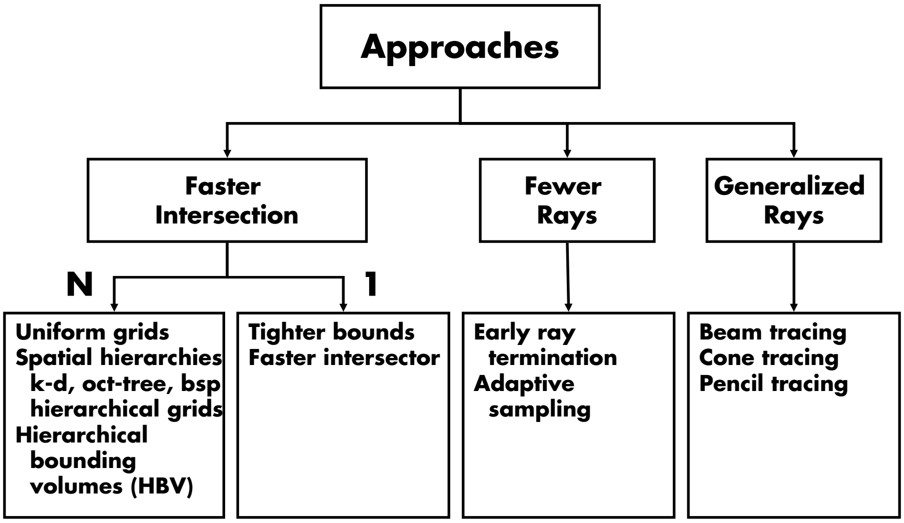
Ray Tracing 1

- Basic algorithm
- Ray-surface intersection (triangles, ...)

Ray Tracing 2

- Brute force $|I| \times |O|$
- Acceleration data structures

Ray Tracing Acceleration Techniques



Approaches

Faster Intersection

Fewer Rays

Generalized Rays

N

1

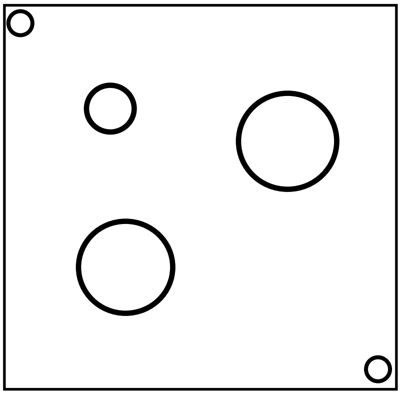
Uniform grids
Spatial hierarchies
k-d, oct-tree, bsp
hierarchical grids
Hierarchical
bounding
volumes (HBV)

Tighter bounds
Faster intersector

Early ray
termination
Adaptive
sampling

Beam tracing
Cone tracing
Pencil tracing

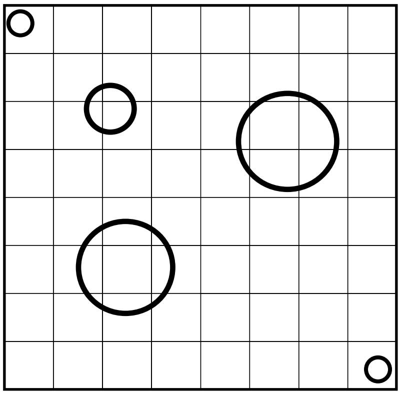
Uniform Grids



Preprocess scene

1. Find bounding box

Uniform Grids

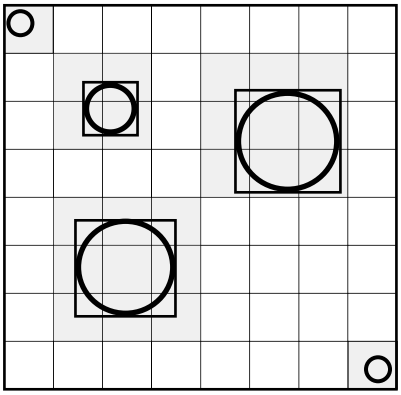


Preprocess scene

1. Find bounding box
2. Determine resolution

$$n^3 = d\,|O|$$

Uniform Grids



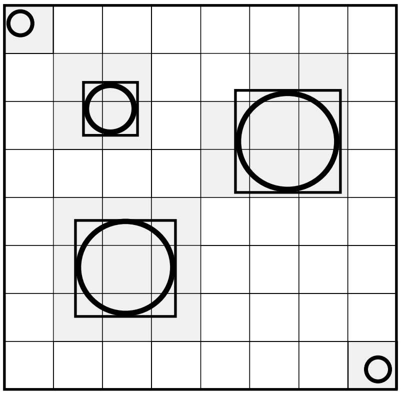
Preprocess scene

1. Find bounding box
2. Determine resolution

$$n^3 = d\,|O|$$

2. Place object in cell, if object overlaps cell

Uniform Grids



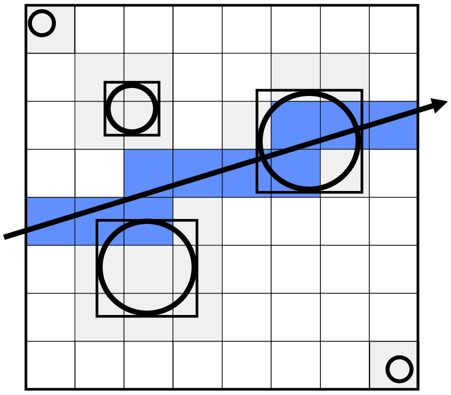
Preprocess scene

1. Find bounding box
2. Determine resolution

$$n^3 = d\,|O|$$

3. Place object in cell, if object overlaps cell
4. Check that object intersects cell

Uniform Grids



Preprocess scene

Traverse grid

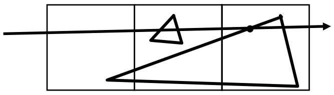
3D line – 3D-DDA

6-connected line

`pbrt` algorithm

Caveat: Overlap

Optimize for objects that overlap multiple cells



Caveat 1:

Intersection must be within cell bound

Caveat 2:

Eliminate redundant intersection tests

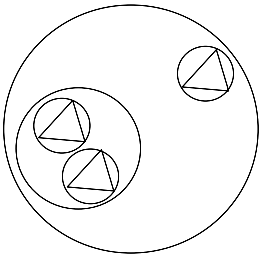
Mailboxes

- Assign each ray a number
- Object intersection cache (mailbox)
  - Store ray number
  - Intersection

Hierarchical Bounding Volumes

Create of tree of bounding volumes

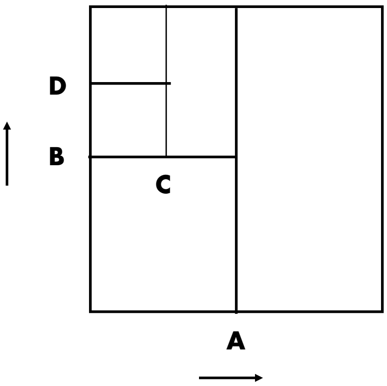
- Children are contained within parent
- Creation preprocess
  - From model hierarchy
  - Automatic clustering

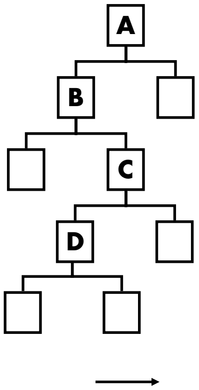


Search

```
intersect(node,ray,hits) {
  if( intersectp(node->bound,ray)
    if( leaf(node) )
      intersect(node->prims,ray,hits)
    else
      for each child
        intersect(child,ray,hits)
}
```

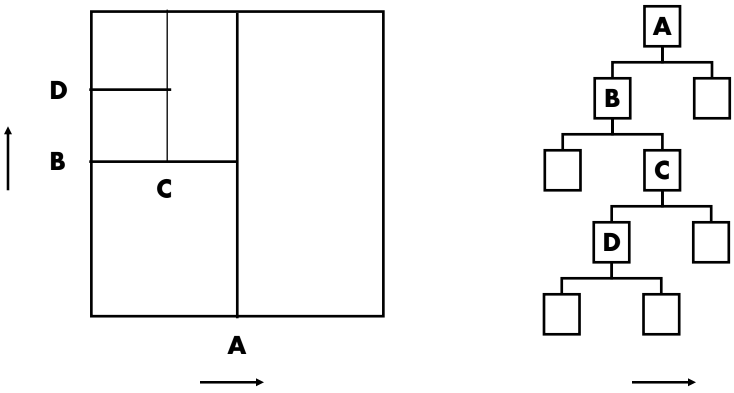
Spatial Hierarchies





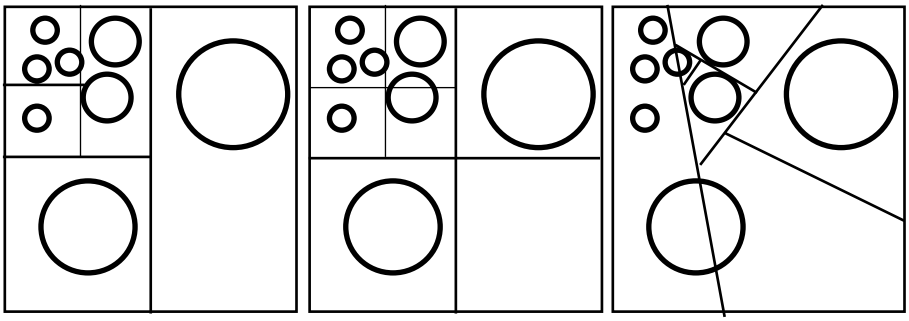
Letters correspond to planes (A, B, C, D)

Spatial Hierarchies



Point Location by recursive search
Leaf nodes correspond to regions

Variations

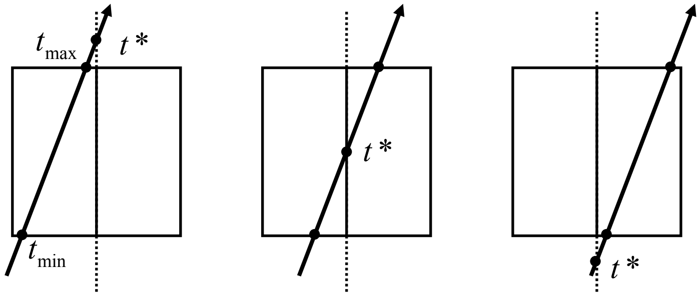


kd-tree oct-tree bsp-tree

Ray Traversal Algorithms

Recursive inorder traversal

Kaplan, Arvo, Jansen



$t_{\max} < t^*$

```
Intersect(L,tmin,tmax)
```

$t_{\min} < t^* < t_{\max}$

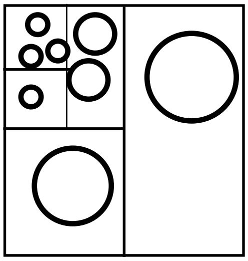
```
Intersect(L,tmin,t*)
Intersect(R,t*,tmax)
```

$t^* < t_{\min}$

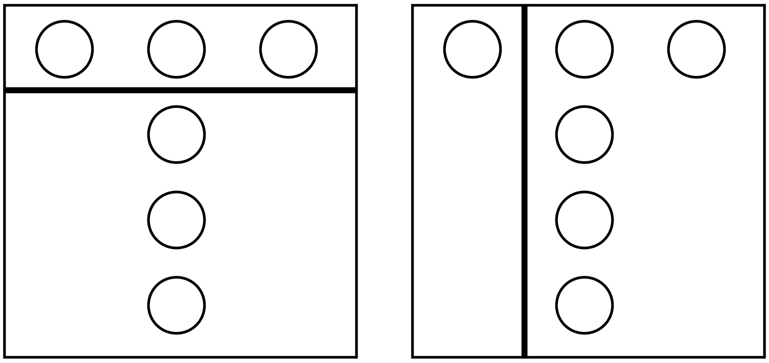
```
Intersect(R,tmin,tmax)
```

Creating Spatial Hierarchies

```
insert(node,prim) {
  if( overlap(node->bound,prim) )
     if( leaf(node) ) {
        if( node->nprims > MAXPRIMS
         && node->depth  < MAXDEPTH ) {
           subdivide(node);
           foreach child in node
             insert(child,prim)
        }
        else
          insertlist(node->prims,prim);
     }
     else
       foreach child in node
         insert(child,prim)
}
// Typically MAXDEPTH=16, MAXPRIMS=2-8
```



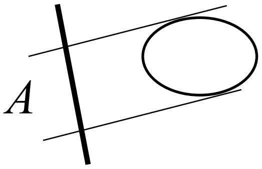
Build Hierarchy Top-Down



Choose splitting plane
- •Midpoint
- •Median cut
- •Surface area heuristic

Surface Area and Rays

Number of rays in a given direction that hit an object is proportional to its projected area



The total number of rays hitting an object is $4\pi\overline{A}$

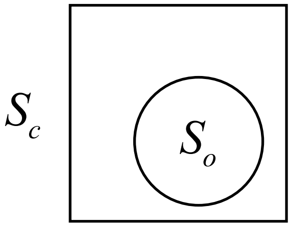
Crofton's Theorem:

For a convex body $\overline{A} = \dfrac{S}{4}$

For example: sphere $S = 4\pi r^2 \quad \overline{A} = A = \pi r^2$

Surface Area and Rays

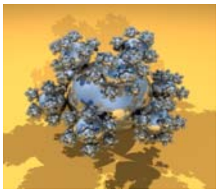
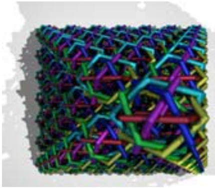
The probability of a ray hitting a convex shape that is completely inside a convex cell equals



$$\Pr[r \cap S_o \mid r \cap S_c] = \frac{S_o}{S_c}$$

Choose hierarchy that minimizes average cost of traversal

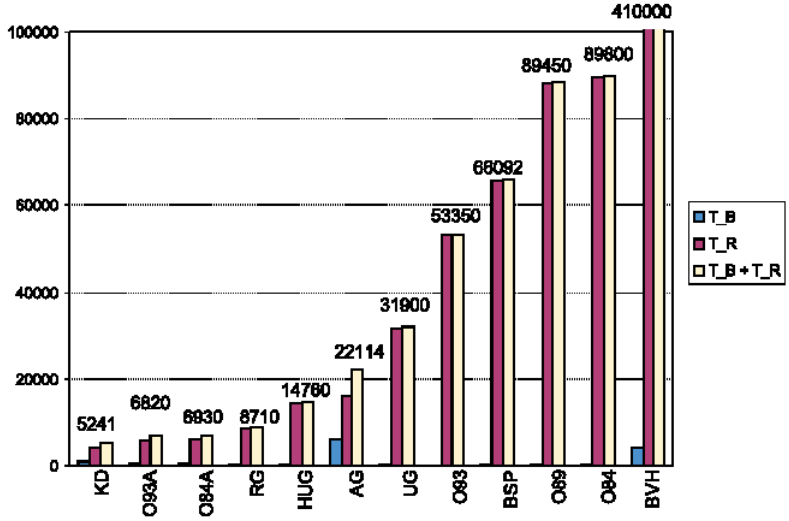
Comparison

Time		Spheres	Rings	Tree
Uniform Grid	d=1	244	129	1517
	d=20	38	83	781
Hierarchical Grid		34	116	34

V. Havran, Best Efficiency Scheme Project
`http://sgi.felk.cvut.cz/BES/`

Comparison



Univ. Saarland RTRT Engine

Ray-casts per second = FPS @ 1K × 1K

RT&Shading Scene	SSE no shd.	SSE simple shd.	No SSE simple shd.
ERW6 (static)	7.1	2.3	1.37
ERW6 (dynamic)	4.8	1.97	1.06
Conf (static)	4.55	1.93	1.2
Conf (dynamic)	2.94	1.6	0.82
Soda Hall	4.12	1.8	1.055

Pentium-IV 2.5GHz laptop
Kd-tree with surface-area heuristic [Havran]

Wald et al. 2003 [`http://www.mpi-sb.mpg.de/~wald/`]

Interactive Ray Tracing

Highly optimized software ray tracers

- Use vector instructions; Cache optimized
- Clusters and shared memory MPs

Ray tracing hardware

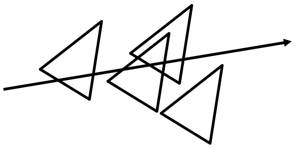
- AR250/350 ray tracing processor
  www.art-render.com
- SaarCOR

Ray tracing on programmable GPUs

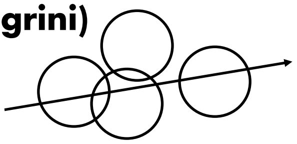
Theoretical Nugget 1

Computational geometry of ray shooting

1. Triangles (Pellegrini)
   - Time: $O(\log n)$
   - Space: $O(n^{5+\varepsilon})$



2. Sphere (Guibas and Pellegrini)
   - Time: $O(\log^2 n)$
   - Space: $O(n^{5+\varepsilon})$

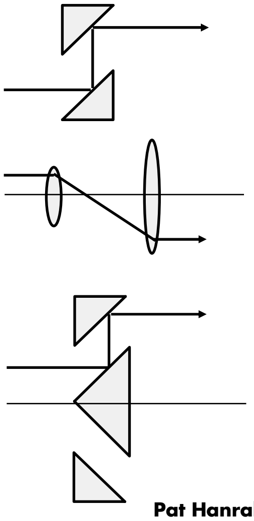


Theoretical Nugget 2

Optical computer = Turing machine

Reif, Tygar, Yoshida

Determining if a ray starting at y0 arrives at yn is undecidable



y = y+1

y = -2*y

if(y>0)