Ray Tracing

Ray Tracing 1

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

Ray Tracing 2

- Constructive solid geometry
- Acceleration data structures

Constructive Solid Geometry

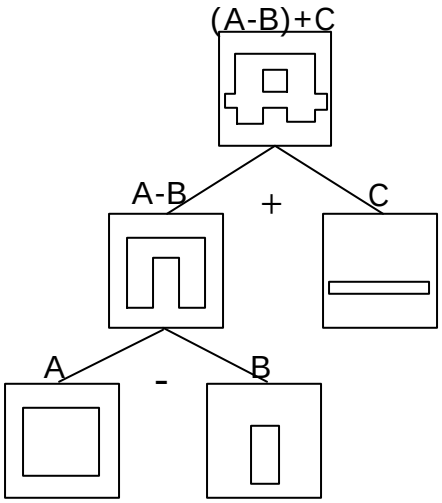
Solid Primitives

- Point Classification

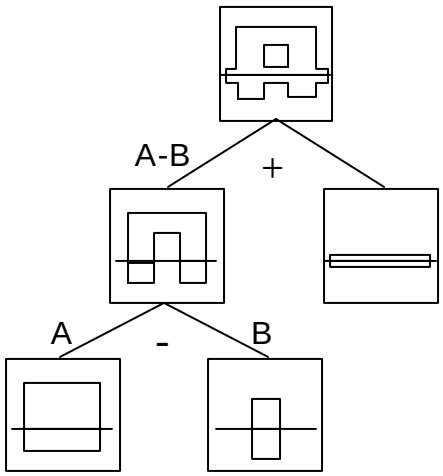
  ?Inside(volume,point)

Spatial set operations

- Intersection
- Union
- Difference



Roth Algorithm

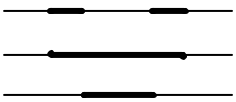


Reduce to 1D calculation
1. Sort
2. Merge

(A-B)+C

(A-B)+C
(A-B)
C

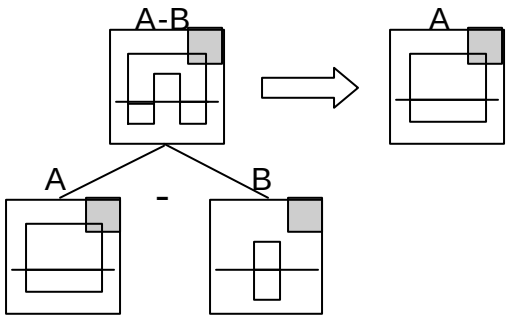
A-B
A
B



CSG Optimizations

Tree Pruning (Broonsvoort et al., Woodwork)

- Within active region simplify CSG tree

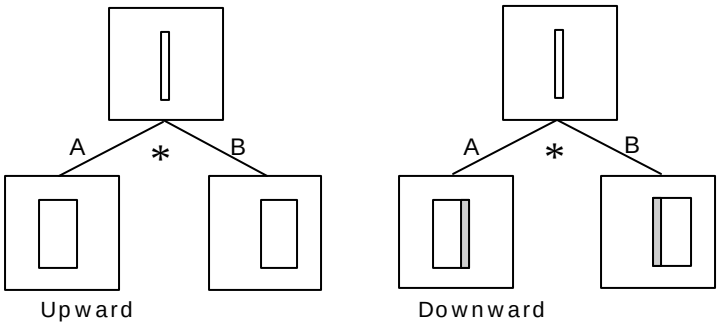


Rules:

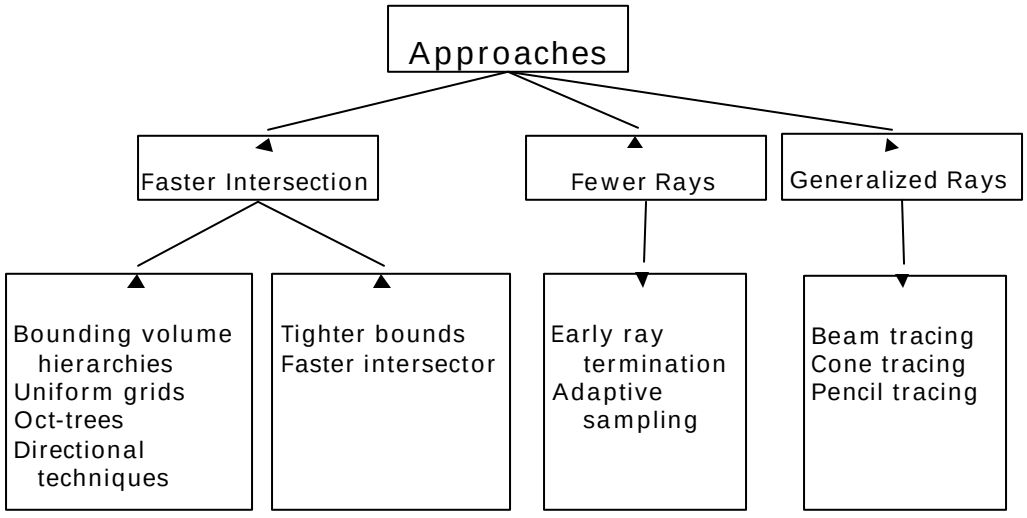
A*B
  B=0 0
  B=1 A
A+B
  B=0 A
  B=1 1
A-B
  B=0 A
  B=1 0

CSG Optimizations

2. S-Bounds (Cameroon)
   - Upward pass: form bounding volumes
   - Downward pass: propagate possible bounds
   - Repeat!



Ray Tracing Acceleration Techniques

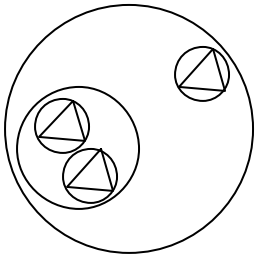


Hierarchical Bounding Volumes

Create of tree of bounding volumes

Children are contained within parent

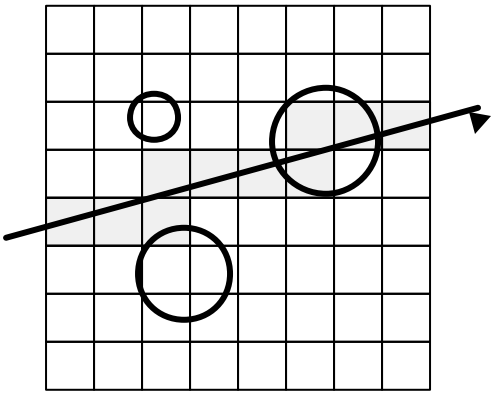
- Creation preprocess
  - From model hierarchy
  - Automatically



Search

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


Uniform Grids



Preprocess scene

1. Determine resolution

   |cells| ~ |objects|

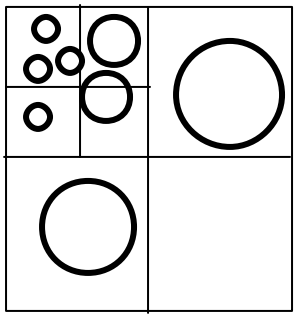
2. Place object in cell if object overlaps cell

Traverse grid

3D line – 3D-DDA

Oct-trees, k-d trees, bsp trees

```
insert(node,prim) {
  if( overlap(node->bound,prim->bound) )
    if( node->nprims > maxprims ) {
      subdivide(node); // and reinsert prims
      foreach child
        insert(child,prim)
    }
    else
      insert(node->prims,prim);
}
```



Oct-trees, k-d trees, bsp trees

Three traversal algorithms:

1. Point location from root

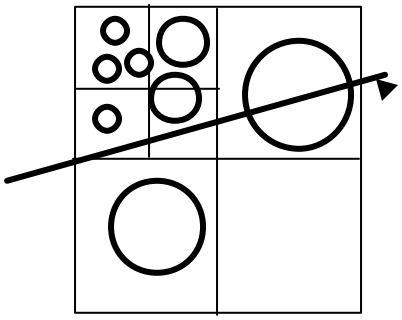
   Glassner, Kaplan

2. Inorder traversal

   Arvo, Jansen

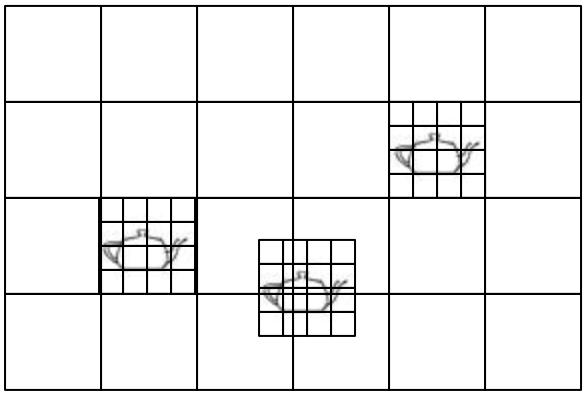
3. Neighbor finding

   Samet



Nested Grids

Good compromise preferred by many practitioners



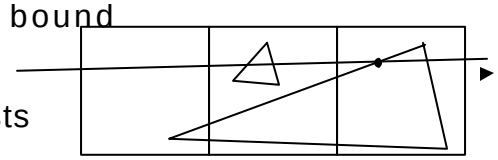
Caveat

Optimize for objects that overlap multiple cells

Caveat 1:

Intesection must be within bound

Caveat 2:



Redundant intersection tests

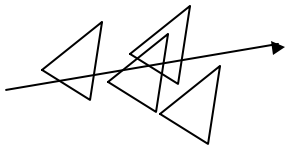
Mailboxes

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

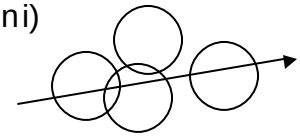
Theoretical Nugget 1

Computational Geometry of Ray Shooting

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



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

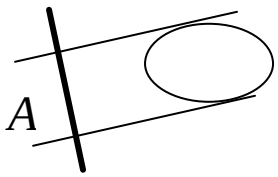


Theoretical Nugget 2

Probability that a ray in a given direction intersects a shape is equal to its projected area.

Probability that any ray intersects a shape is equal to its average projected area.

Theorem: For a convex body $\overline{A} = \frac{S}{4}$



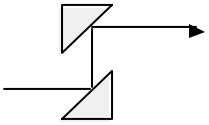
Sphere: $\overline{A} = A = \pi r^2$

$S = 4\pi r^2$

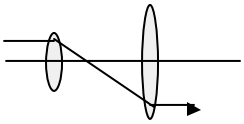
Theoretical Nugget 3

Optical Computer = Turing Machine

y=y+1



y=2*y



if(y>0)