

CS-184: Computer Graphics

Lecture #7: BSP and AABB Trees

Brandon Wang and Prof. James O'Brien
University of California, Berkeley

V2013.5-07.1.0

1

Announcements

Assignment 2: Soon...

2

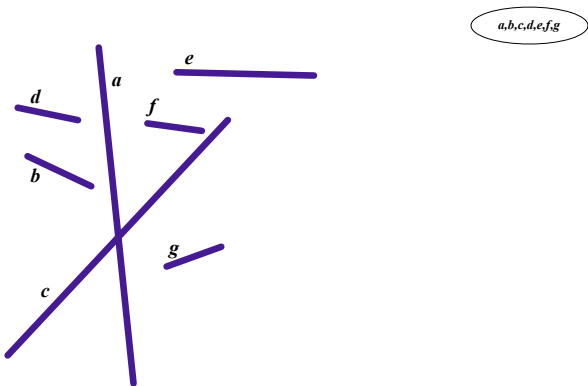
BSP-Trees

- Binary Space Partition Trees
 - Split space along planes
 - Allows fast queries of some spatial relations
- Simple construction algorithm
 - Select a plane as sub-tree root
 - Everything on one side to one child
 - Everything on the other side to other child
 - Use random polygon for splitting plane

3

3

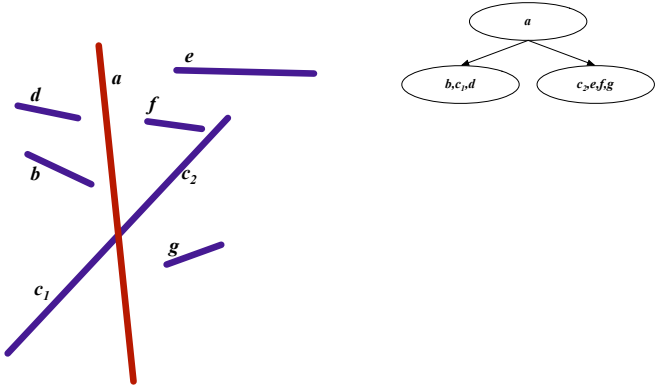
BSP-Trees



4

4

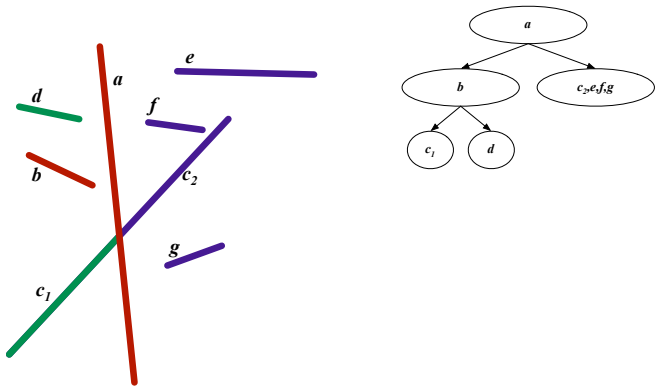
BSP-Trees



5

5

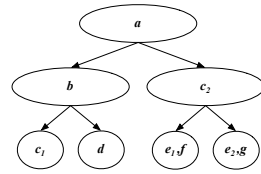
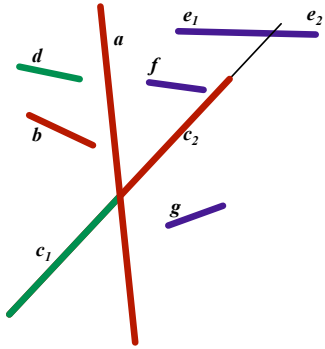
BSP-Trees



6

6

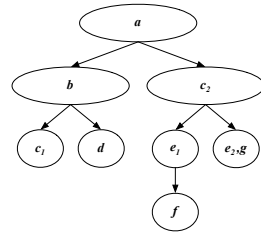
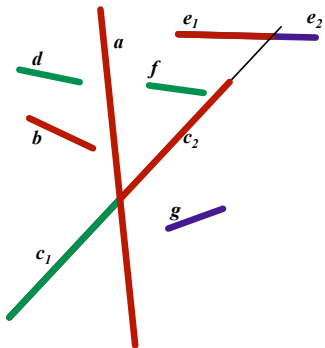
BSP-Trees



7

7

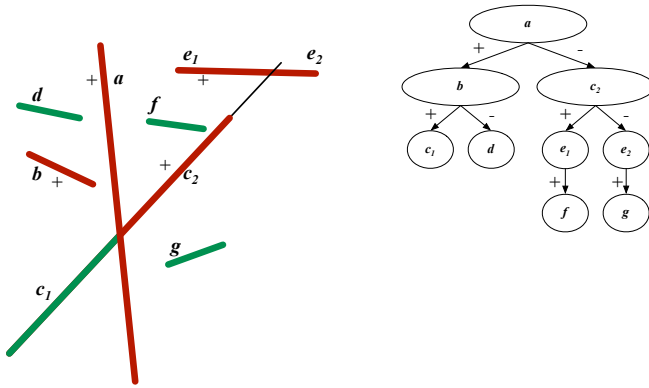
BSP-Trees



8

8

BSP-Trees



9

9

BSP-Trees

- Visibility Traversal
 - Variation of in-order-traversal
 - Child one
 - Sub-tree root
 - Child two
 - Select "child one" based on location of viewpoint
 - Child one on same side of sub-tree root as viewpoint

10

10

Your Ray Tracer

```
RayTrace (image)
```

```
  For ray in camera  
    image[pixel] = Trace(ray)
```

```
Trace (ray)
```

```
  t_hit = infinity  
  For object in scene  
    t_hit = min(object.intersect(ray), t_hit)  
  shade at t_hit  
  possible calls to Trace(new_ray)
```

13

13

Your Ray Tracer

```
RayTrace (image)
```

```
  For ray in camera  
    image[pixel] = Trace(ray)
```

```
Trace (ray)
```

```
  t_hit = infinity  
  For object in scene  
    t_hit = min(object.intersect(ray), t_hit)  
  shade at t_hit  
  possible calls to Trace(new_ray)
```

14

14

Your Ray Tracer

```
RayTrace (image)
```

```
  For ray in camera  
    image[pixel] = Trace(ray)
```

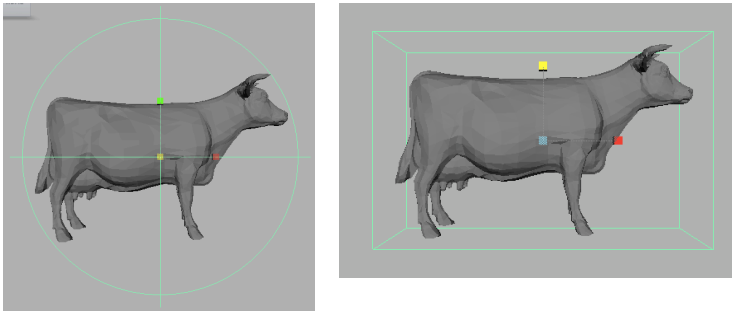
```
Trace (ray)
```

```
  t_hit = infinity  
  For object in scene  
    t_hit = min(object.intersect(ray), t_hit)  
  shade at t_hit  
  possible calls to Trace(new_ray)
```

15

15

Bounding Shapes



- Bounding shape completely encloses associated object
- Rays cannot hit object w/o intersecting bounding shape
- Two objects cannot collide if shapes don't overlap
- Simplicity -vs- tightness

16

16

AABB Trees

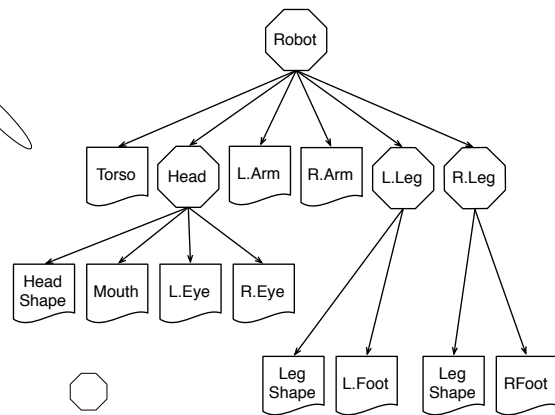
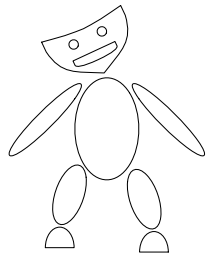




25

25

AABB Trees

One of many variations



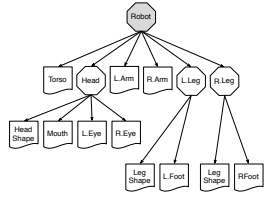
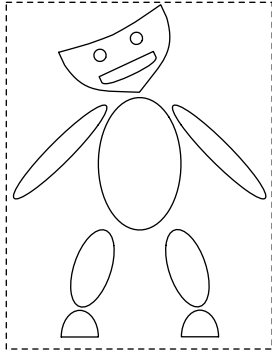
Group node 
Geometry node 
Transformation stored at all nodes

26

26

AABB Trees

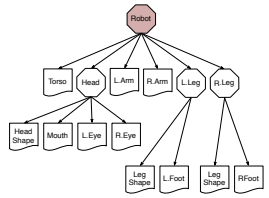
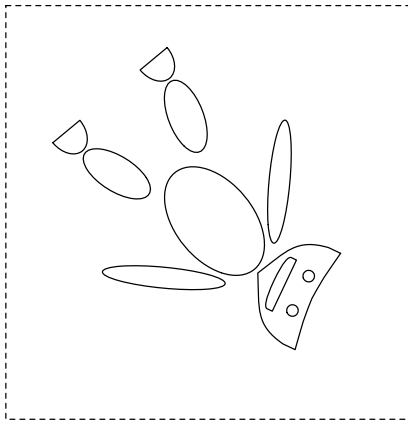
One of many variations



33

AABB Trees

One of many variations

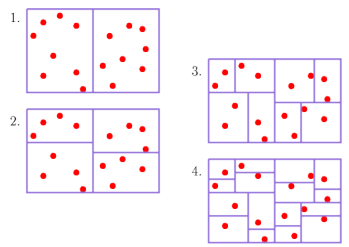


34

Ray Test Against Bound Tree

```
• RayHitSubTree (&ray, node)
  • If RayHitsBB (ray, node.xfBB)
    • ixfRay = Inverse (node.xf) * ray
    • If RayHitsBB (ixfRay, node.BB)
      • If node is group
        • Foreach child in node.children
          • RayHitSubTree (ixfRay, child)
      • else // node not group
        • RayHitGeometry (ixfRay, node.geom)
    • ray.collisionInfo.update (ixfRay)
```

Building the tree



- Sort (or QuickSelect) and split on one axis
- Repeat for the other axis
 - x,y,z

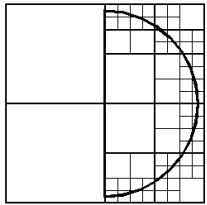
Other Schemes

- Uniform Grid/Octrees
- Spatial Hierarchies
- Etc

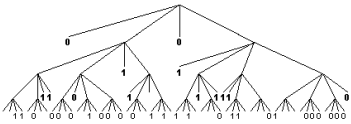
37

Voxels/Octree

VOXELS - OCTREE
(Illustrated with Pixels and Quadtree)



• <http://www.youtube.com/watch?v=sciLNxmMTXM>



Most suitable for: Brain-scan data, representation of a sponge.

38
