# CS-184: Computer Graphics

## Lecture #7: BSP and AABB Trees

Prof. James O'Brien
University of California, Berkeley

V2013-F-07-1.0

1

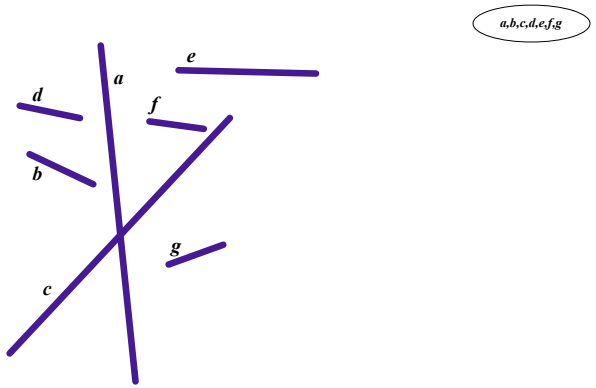
## BSP-Trees

- 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



2

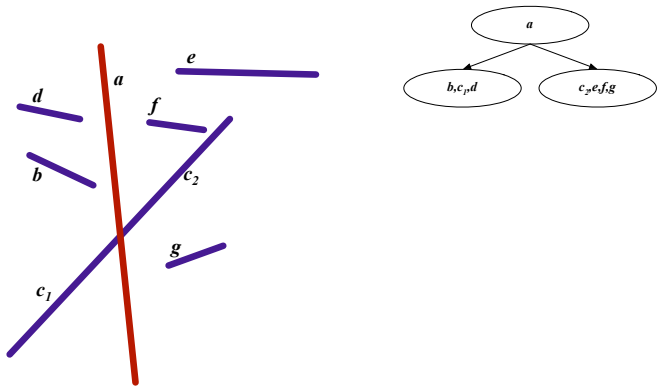
# BSP-Trees




3

3

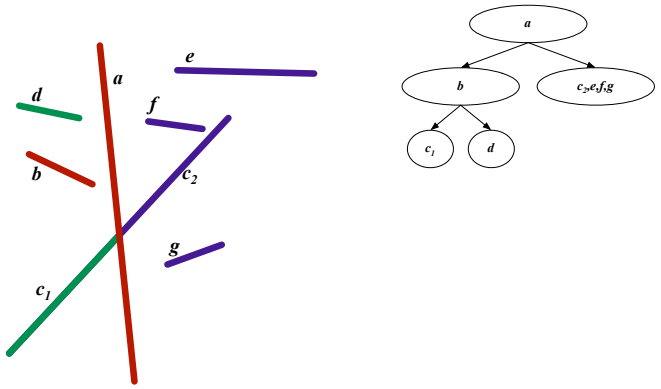
# BSP-Trees




4

4

# BSP-Trees



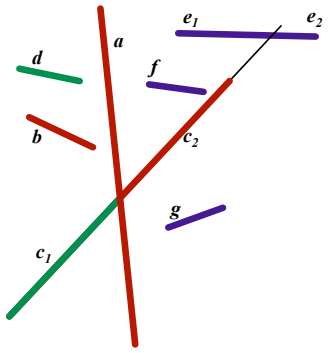


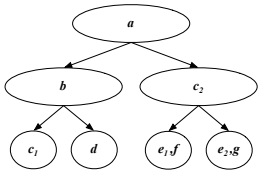


5

5

# BSP-Trees

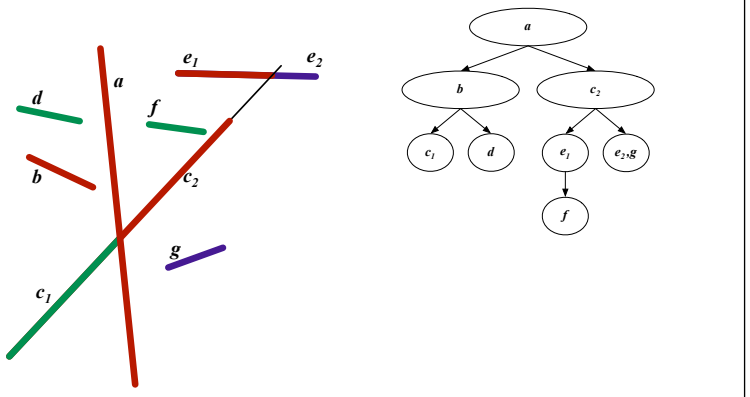



6

6

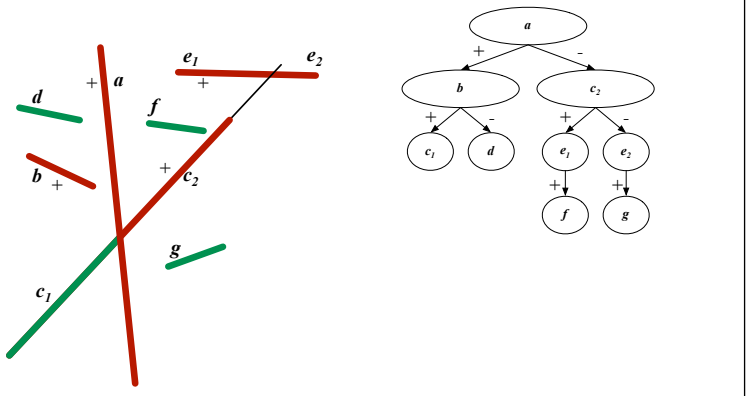
# BSP-Trees




7

7

# BSP-Trees




8

8

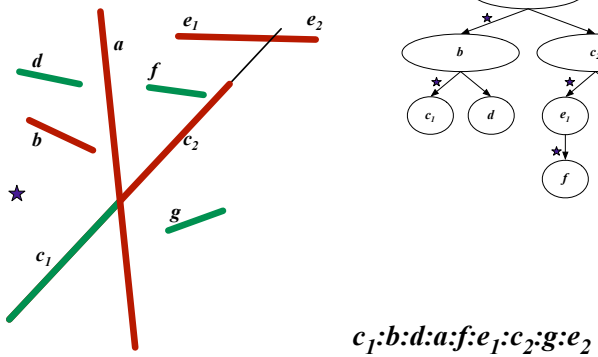
# BSP-Trees

- 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



9

# BSP-Trees





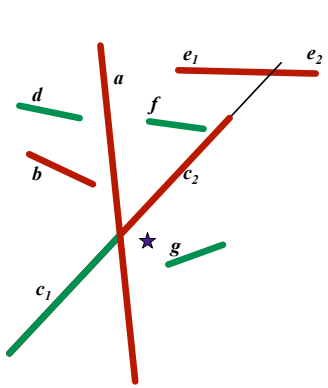


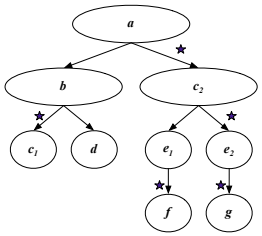
$c_1$:b:d:a:f:$e_1$:$c_2$:g:$e_2$



10

# BSP-Trees




$g{:}e_2{:}c_2{:}f{:}e_1{:}a{:}c_1{:}b{:}d$

11

11

# 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)
```

12

12

# 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

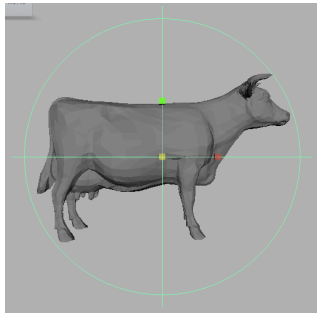
# Your Ray Tracer

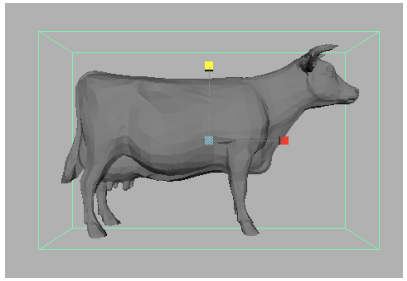
```
RayTrace(image)
   For ray in camera
     image[pixel] = Trace(ray)
Trace(ray)
     t_hit = infinity
     For object in scene
       t_hit = min(object.intersect(ray), t_hit)
     shade at t_hit
     possible calls to Trace(new_ray)
```

14

14

# Bounding Shapes



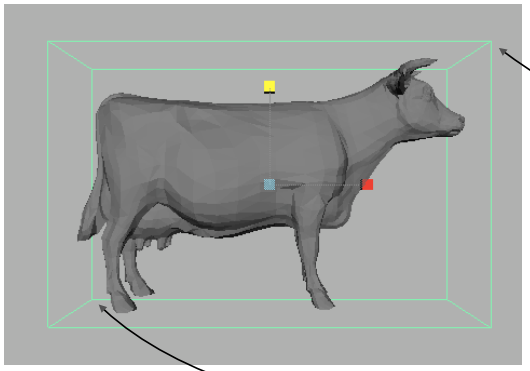


- Bounding shape completely encloses associated object
- Rays cannot hit object w/o intersecting bounding shape
- Two objects cannot collide if shapes don't overlap
- Simplicity -vs- tightness

15

15

# Axis-Aligned Bounding Boxes

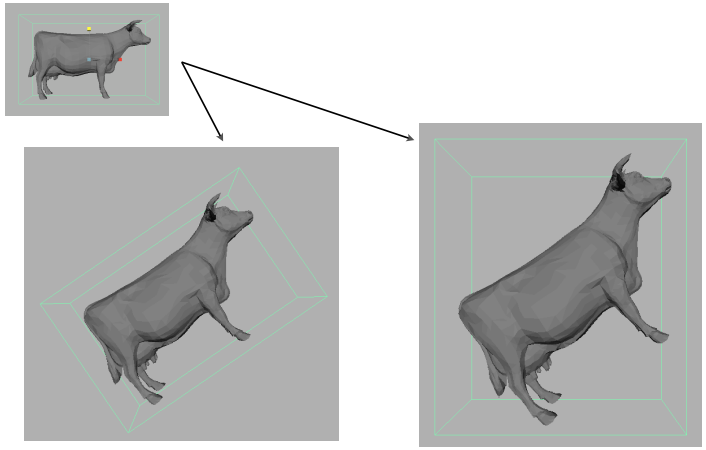



- Axis-aligned bounding box defined by min and max *x,y,z*

16

16

# Axis-Aligned Bounding Boxes



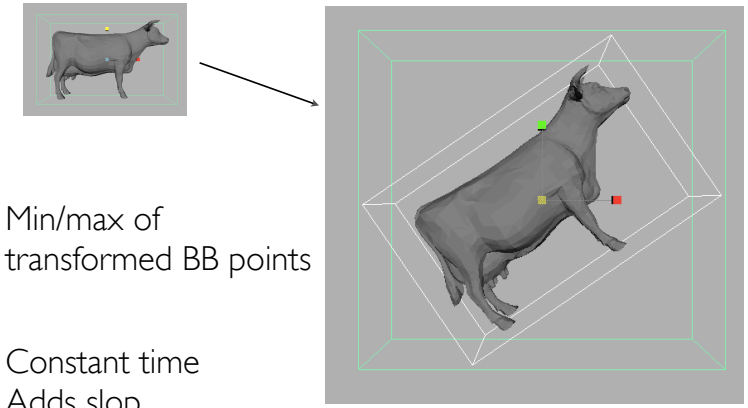
Transform box
Not axis-aligned

Min/max of new points
Linear cost to compute

17

17

# Axis-Aligned Bounding Boxes



Min/max of
transformed BB points

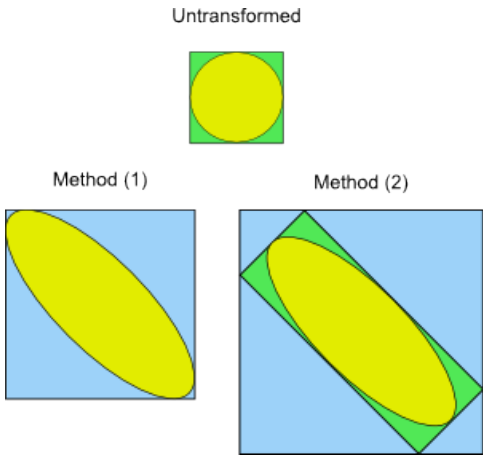
Constant time
Adds slop
Cumulative slop if multiple transforms occur sequentially

**Why would we do this?**

18

18

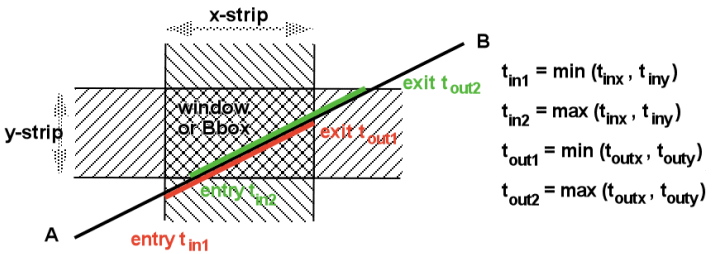
# Tightness




19

19

# Box Intersection




$t_{in1} = \min(t_{inx}, t_{iny})$

$t_{in2} = \max(t_{inx}, t_{iny})$

$t_{out1} = \min(t_{outx}, t_{outy})$

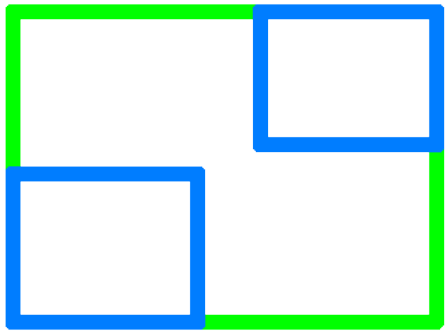
$t_{out2} = \max(t_{outx}, t_{outy})$

Line goes through window/Bbox if intervals overlap

20

20

## Bounding Box of Bounding Boxes



21

21

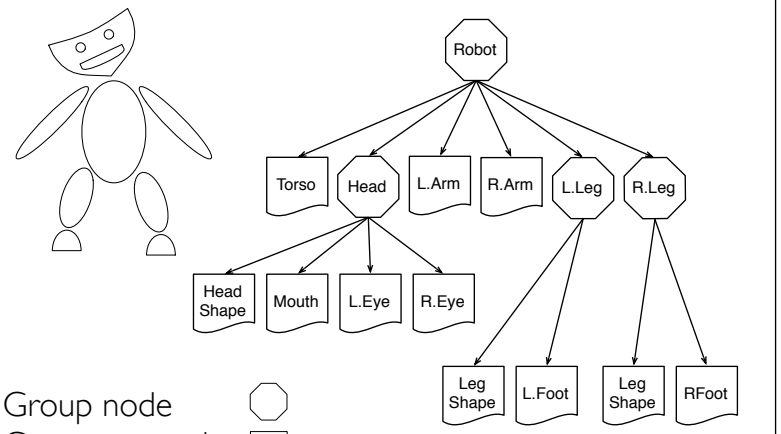
## AABB Trees

22

22

# AABB Trees

**One of many variations**




Group node

Geometry node

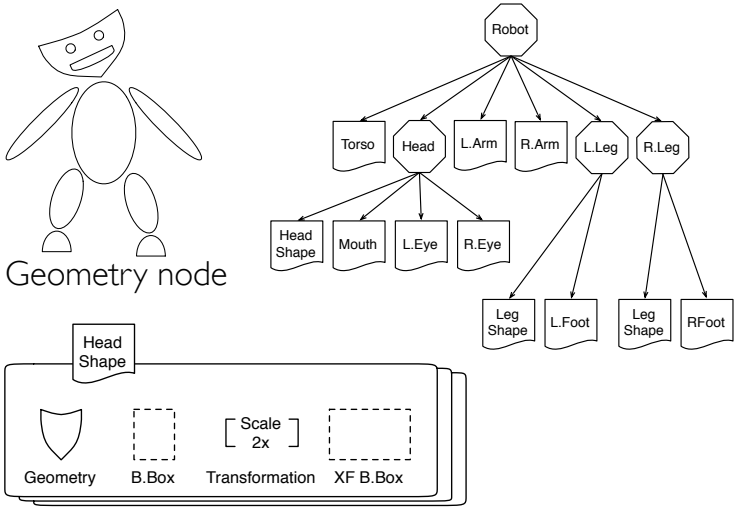
Transformation stored at all nodes

23

23

# AABB Trees

**One of many variations**

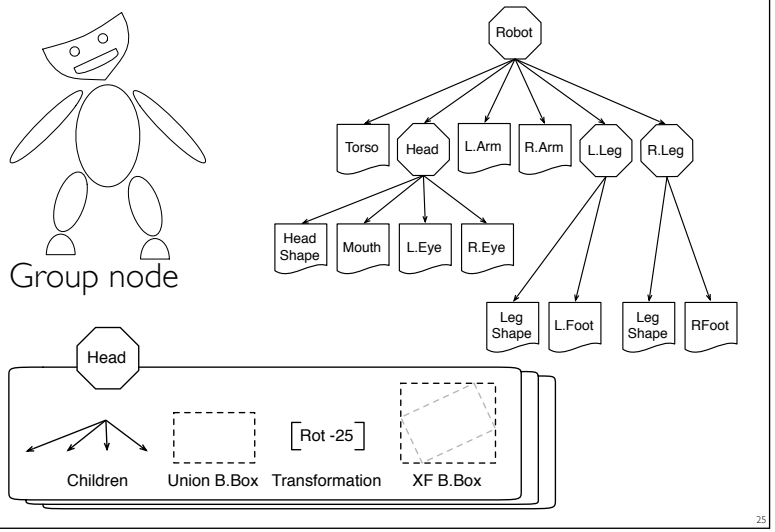



24

24

# AABB Trees

**One of many variations**

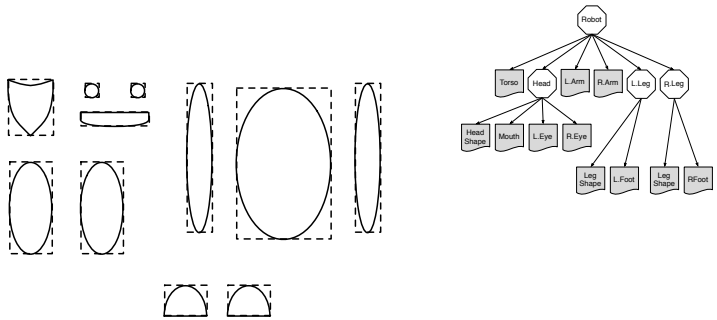



25

25

# AABB Trees

**One of many variations**





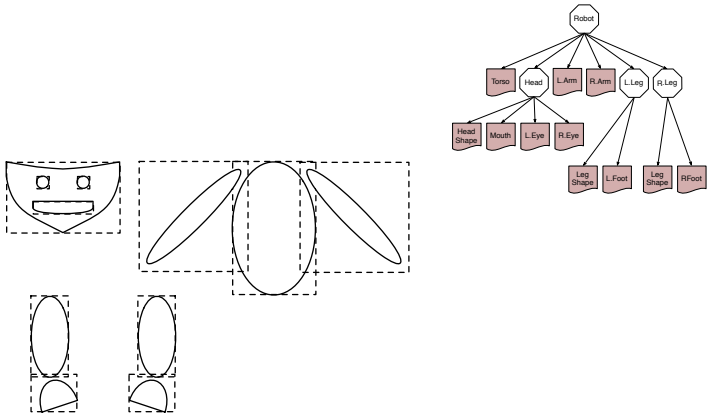

Local Bounding Boxes

26

26

# AABB Trees

**One of many variations**



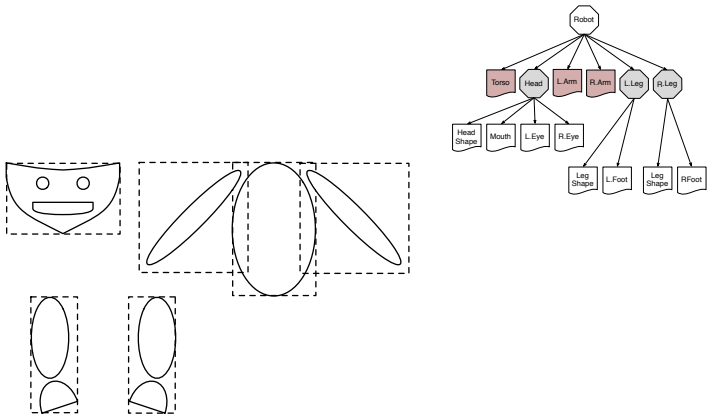

Transformed Bounding Boxes

27

27

# AABB Trees

**One of many variations**

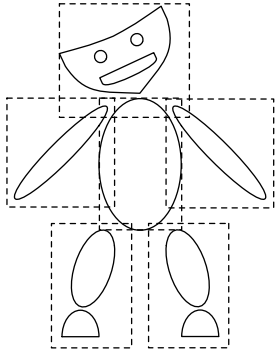


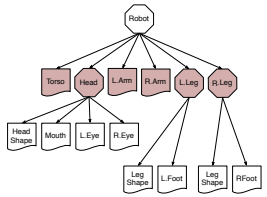

28

28

# AABB Trees

**One of many variations**



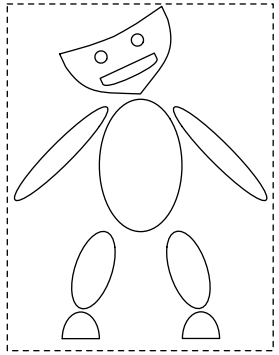


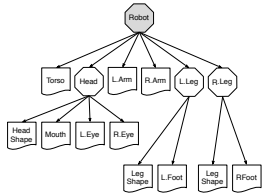

29

29

# AABB Trees

**One of many variations**



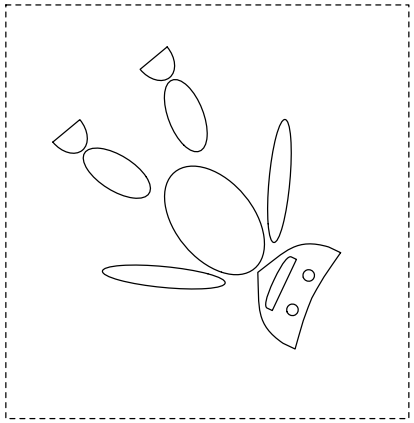


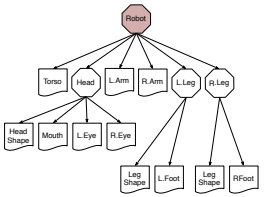

30

30

# AABB Trees

**One of many variations**








31

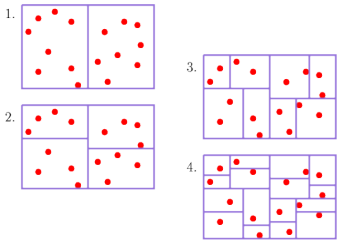
# Ray Test Against Bound Tree

- `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)`



32

# Building the tree




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

33

33

# Other Schemes

- Uniform Grid/Octrees
- Spatial Hierarchies
- Etc

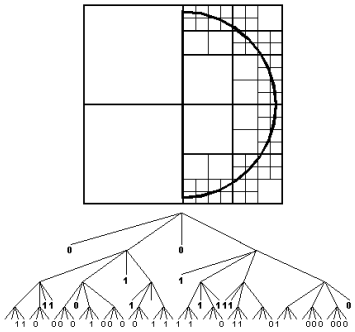
34

34

# Voxels/Octree

**VOXELS - OCTREE**

**(Illustrated with Pixels and Quadtree)**



- http://www.youtube.com/watch?v=sciLNxmMTXM

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

35