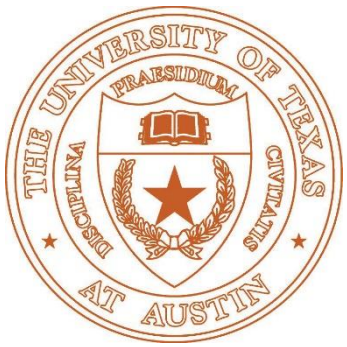
Slide Credit: Michael Wimmer

CS354 Computer Graphics
Spatial Data Structure

Qixing Huang

February 7th 2018

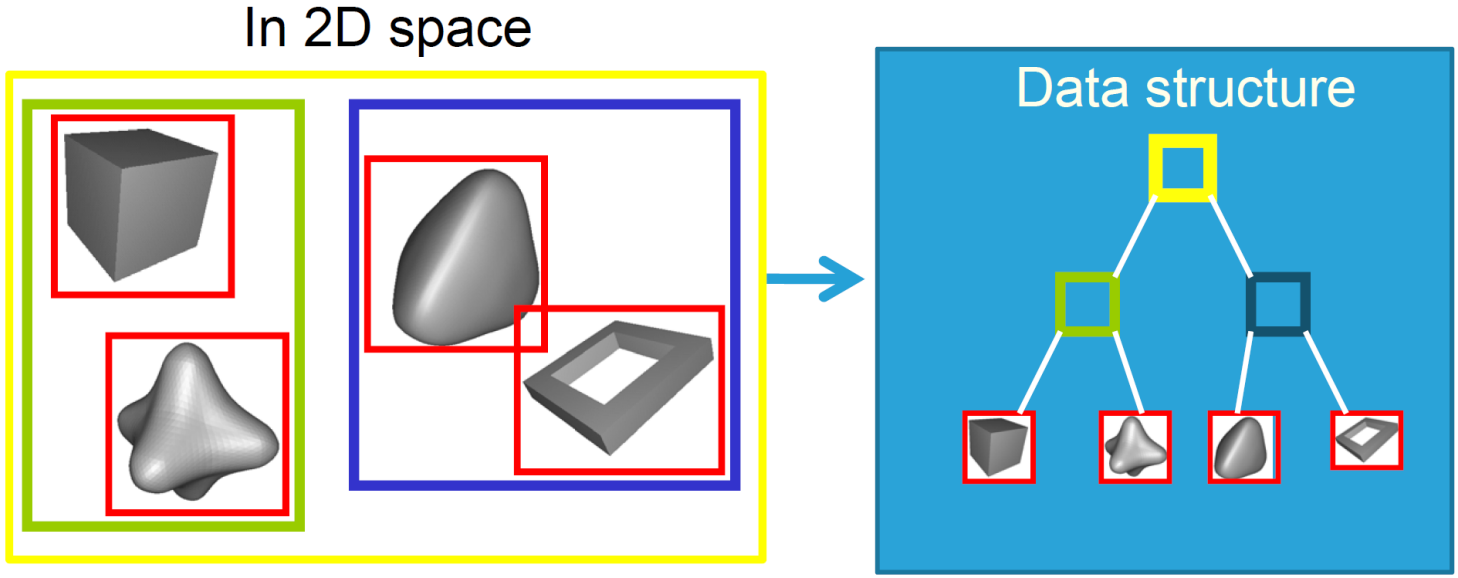


What is it?

- Data structures that organize geometry (point clouds and triangular meshes) in 2D,3D or higher dimensions
- Used for every search related problem
- Very important mathematical tool in CG
  - Ray tracing/Photon mapping
  - Collision/Intersection
  - Culling
  - Data compression
  - Level of detail
- Goal is faster processing and searching

How

- Organize geometry in a hierarchy



Usage of Spatial Data Structures

- Ray intersection (for picking)

- View frustum culling
- Occlusion culling
- Backface culling

- Collision detection (for physics or gameplay)
- Silhouette extraction (for drawing outlines)

Ray Intersection Algorithm

- Recursively descend down the tree

- If ray misses bounding volume, no intersection
- If ray intersects bounding volume, recurse with the enclosed volumes and objects

- Maintain near and far bounds to prune further
- Overall effectiveness depends on model and constructed hierarchy

Culling

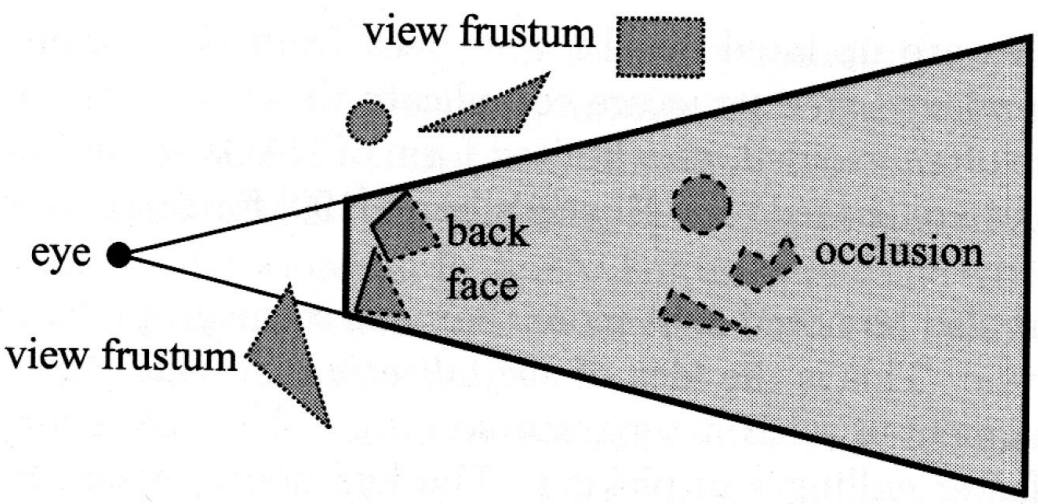
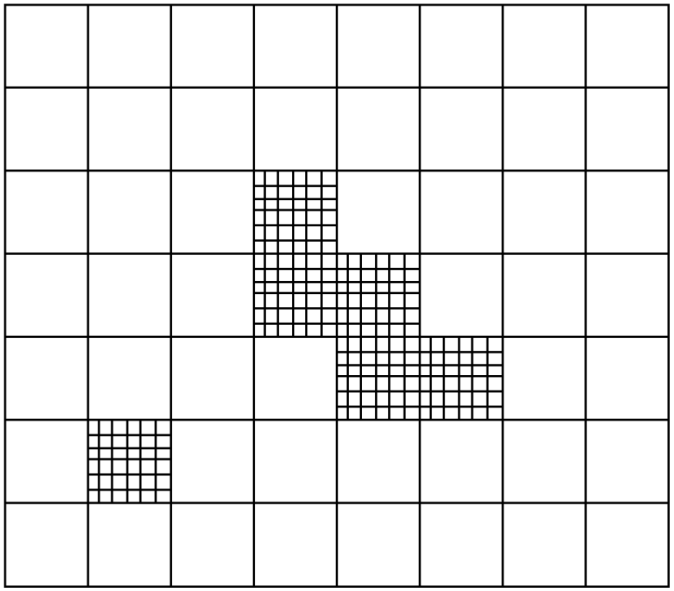


Figure 9.7. Different culling techniques. Culled geometry is dashed. *(Illustration after Cohen-Or et al. [135].)*

[Akenine-Möler & Haines]

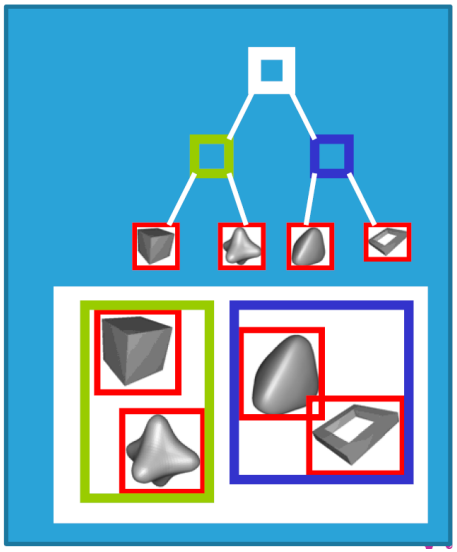
Grid

- Most simple data structure
- Regular subdivision
  - Directly addressable cells
  - Simple neighborhood finding in O(1)
- Problem:
  - Too few/many cells
  - Hierarchical grid
- Good for uniformly distributed problems



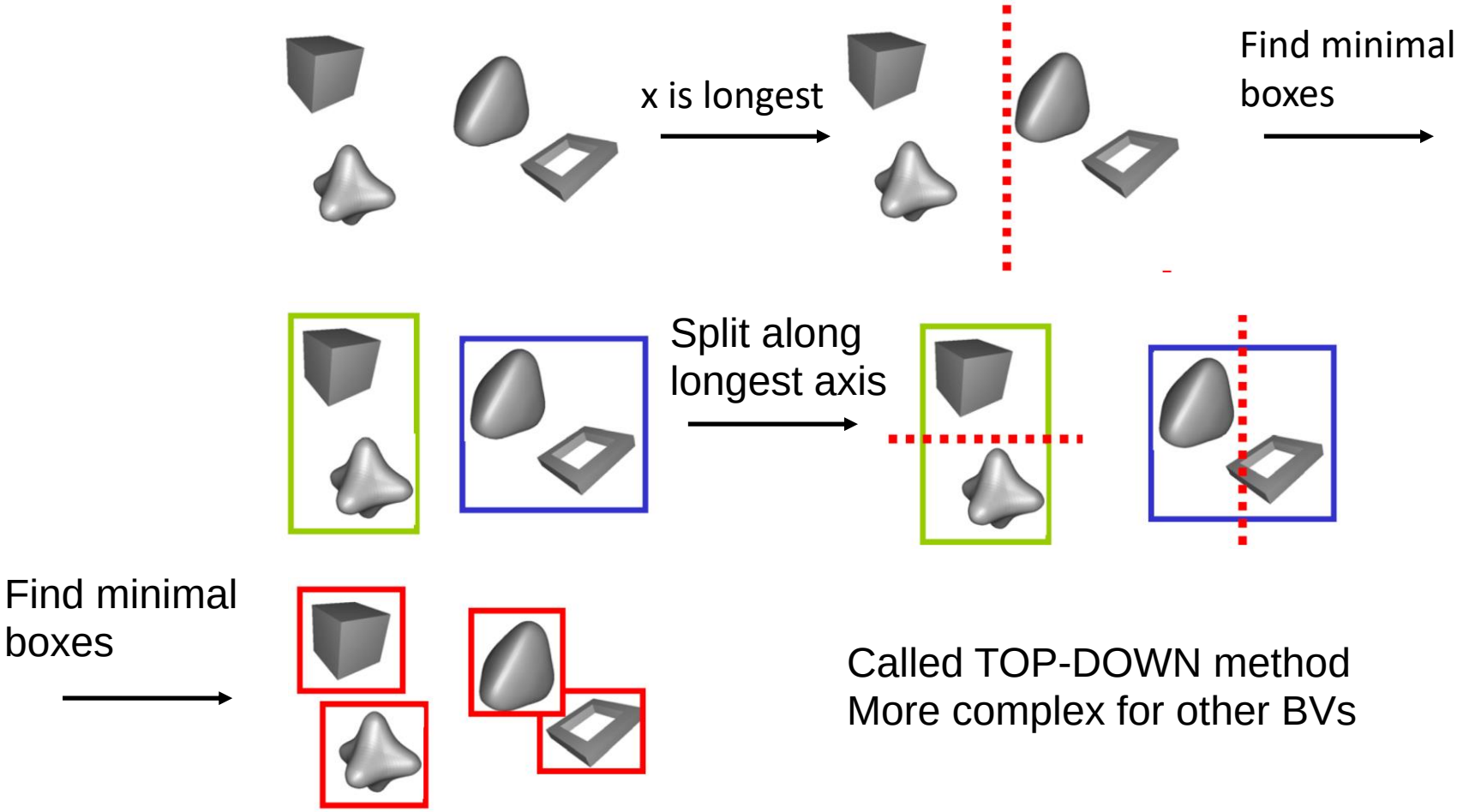
Bounding Volume Hierarchy (BVH)

- Most used structure in real-time graphics
- Most common bounding volumes (BVs):
  - Sphere
  - Boxes (AABB)
- BVs give information about maximum extend of an object
  - Encloses complete object
- Data structure is a k-ary tree
  - Leaves hold geometry
  - Internal nodes have at most k children
  - Internal nodes hold BVs that enclose all geometry in its subtree



How to create a BVH

- Find minimal box, then split along longest axis

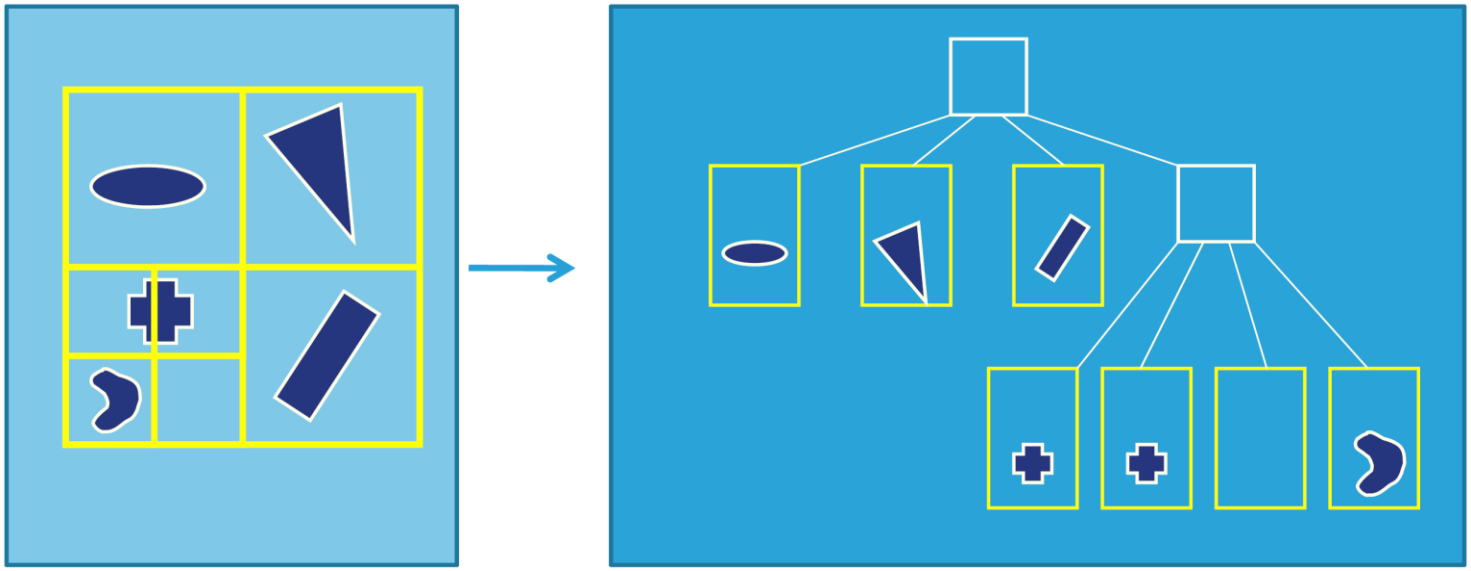


Stopping Criteria for BVH

- We need to stop recursion when:
  - BV is empty
  - Only one primitive (e.g. Triangle, object) is inside Bounding Volume
  - $< n$ primitives is inside BV
  - Recursion level l has been reached

- Similar critera for other BSP trees

Octree (3D) Quadtree (2D)



- Split at half the length axis aligned
  - Always 4 children
- In 3D each square becomes a box with 8 children

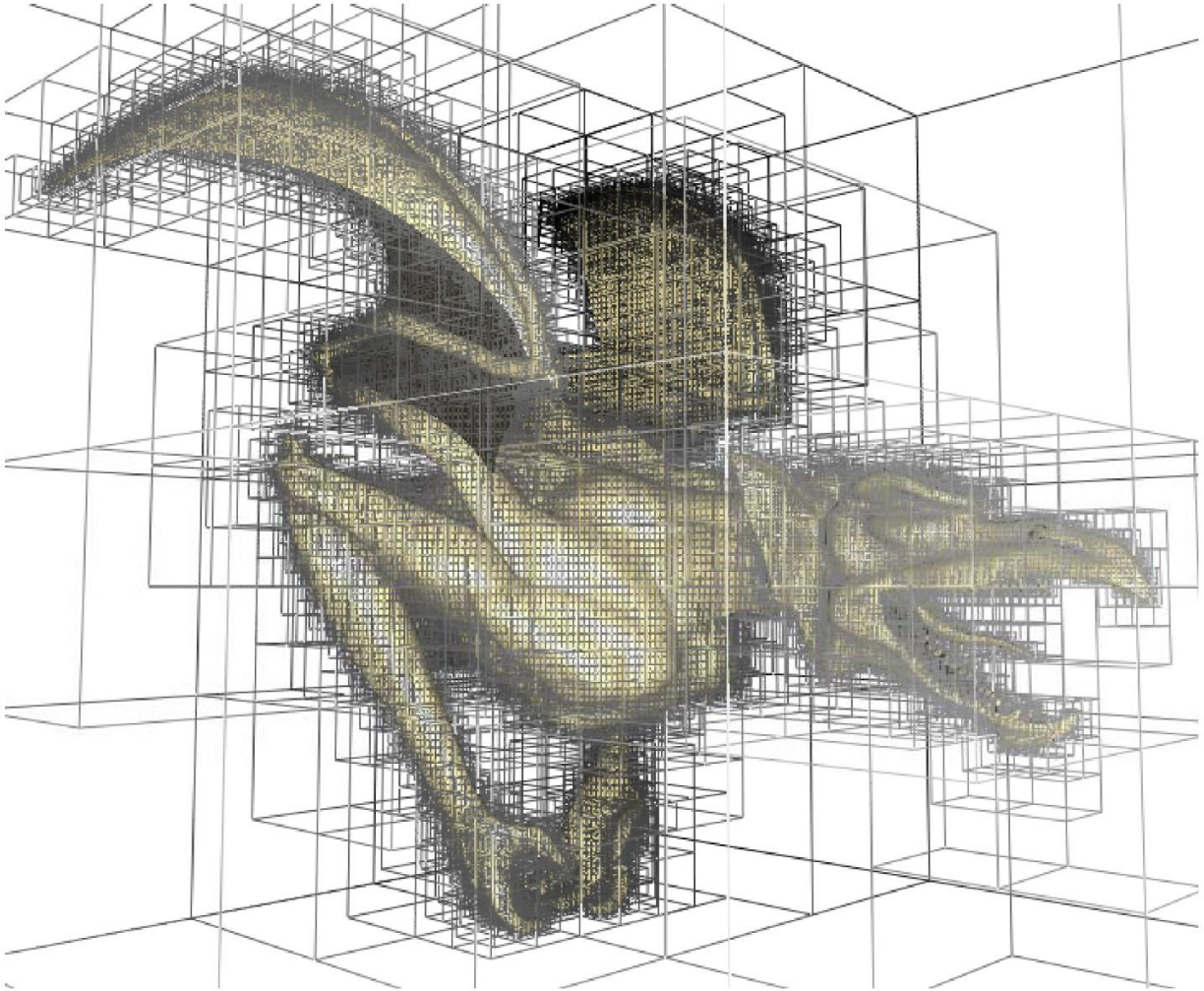
Octree (3D) Quadtree (2D)

- Expensive to rebuild (all BSPs are)

- Easy to implement
  - No geometry analysis needed
  - Just test if something is in leaf

- Used to speed up
  - Culling, Raytracing, Picking

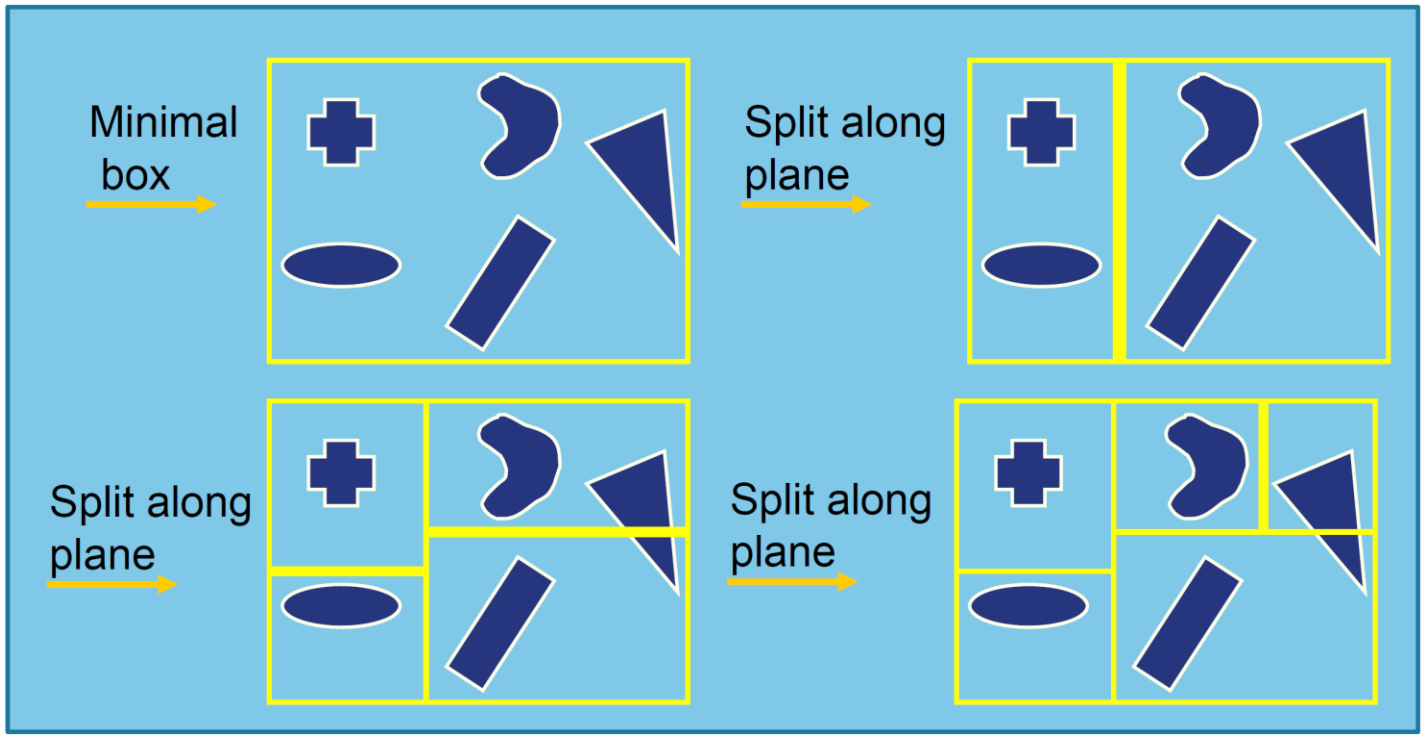
Octree (3D) Quadtree (2D)



Q:Space/Time Complexity of a Quadtree/Octree for n elements

Kd-tree

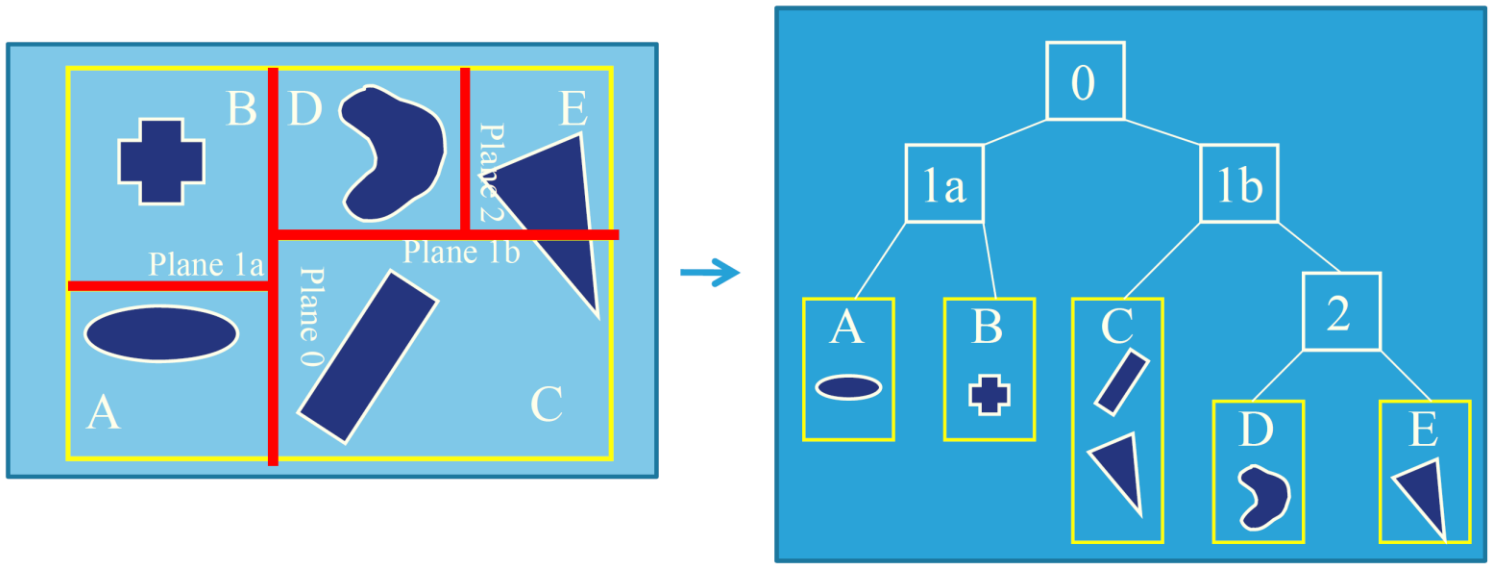
- Split along axis alinged planes/lines which results in minimum search time



Kd-tree

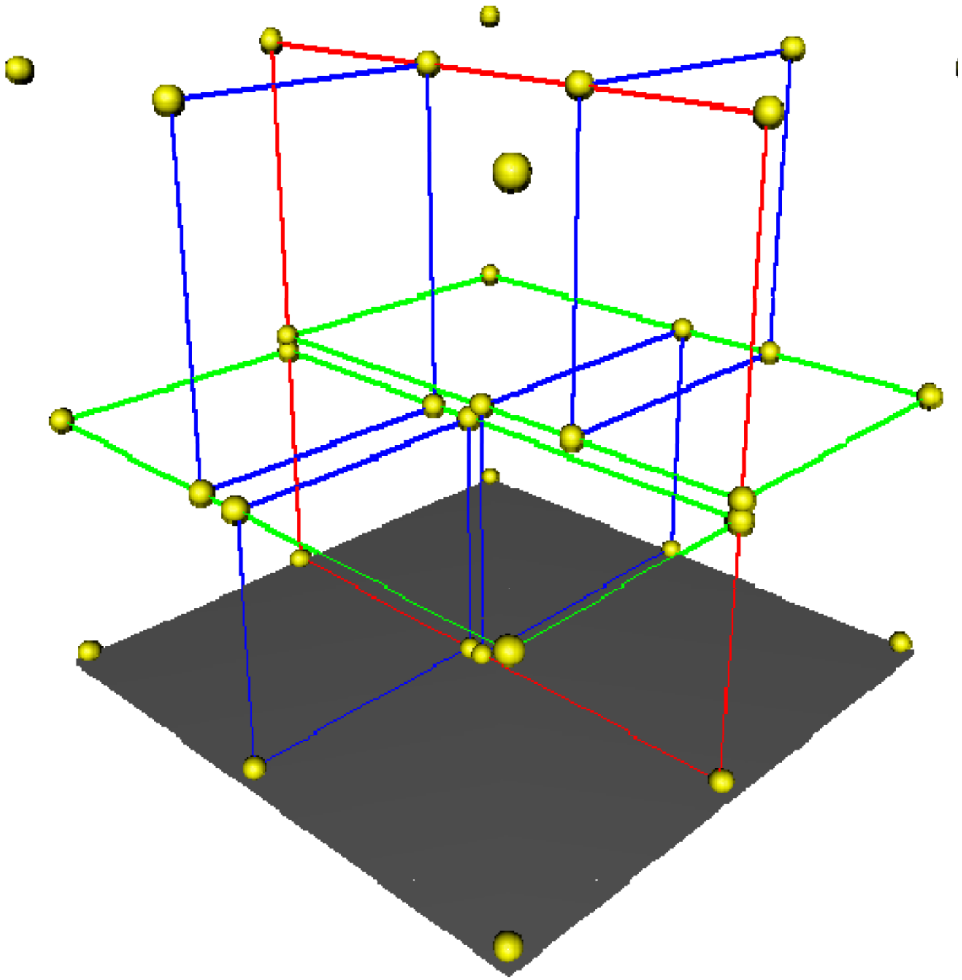
- Test one attribute at a time instead of all simultaneously as in the point quadtree
- Usually cycle through all the attributes
- Shape of the tree depends on the order in which the data is encountered

Kd-tree



- Each internal node holds a divider plane
- Leafs hold geometry
- Problem: splitting criteria is complicated
  - Surface area heuristic (SAH) is best

Kd-tree



Q:Space/Time Complexity of a Kd-tree for n elements

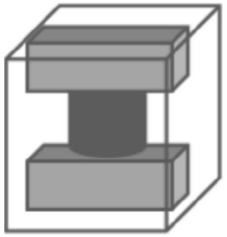
General BSP-Tree

- Same as kd-tree but without axis aligned splitting
  - Splitting criteria is even harder for general generation
  - Intersection calculations more expensive than kd-tree

- Usually used on a per triangle/quad splitting basis
  - Good for per triangle or quad collision detection

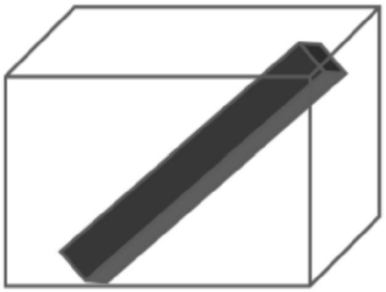
Selection of Bounding Volumes

- Effectiveness depends on:
  - Probability that ray hits bounding volume, but not enclosed objects (tight fit is better)
  - Cost of calculating intersections with bounding volume and enclosed objects
- Break-down into steps the calculations of bounding volumes
- Use heuristics



good

bad



Data Structures Demos

- BSP Tree construction

  http://symbolcraft.com/graphics/bsp/index.html

- KD Tree construction

  http://donar.umiacs.umd.edu/quadtree/points/kdtree.html

Questions?