Johns Hopkins
Engineering for Professionals

605.767 Applied Computer Graphics

Brian Russin

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Module 5A
Spatial Data Structures

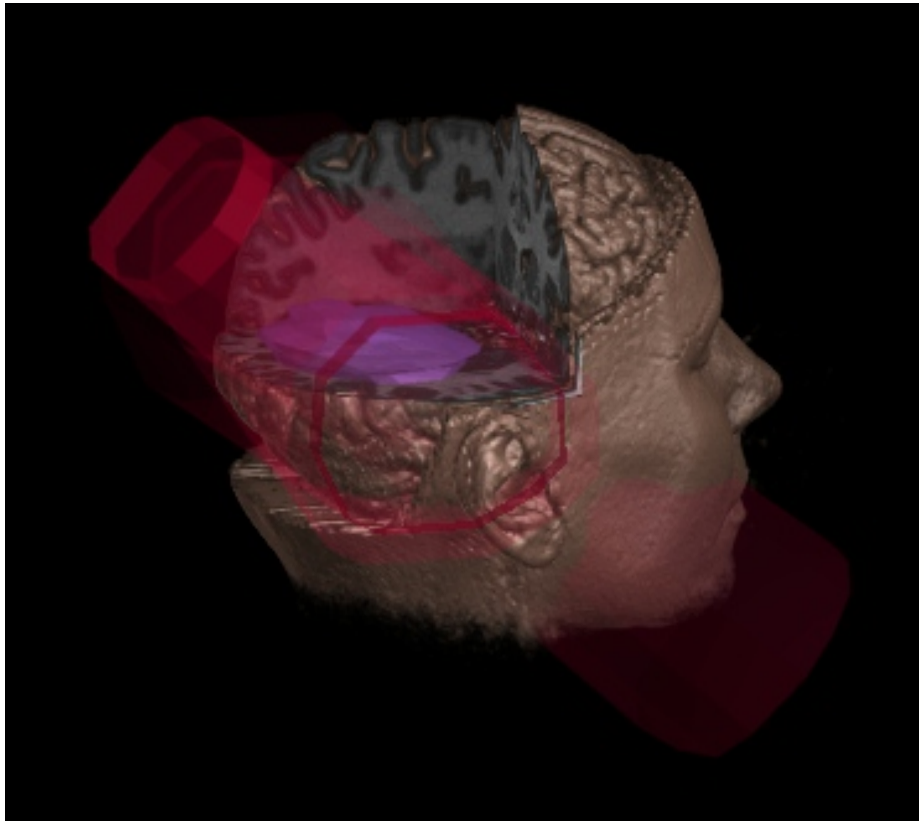
Spatial Data Structures

- Spatial data structures
  - Octrees
  - Bounding Volume Hierarchies
  - BSPs
  - Scene graphs

Goals in Real Time Rendering

- Haines and Moller describe four performance goals
  - More frames per second
    - Interactivity, reduced latency
    - 60-85 frames is considered sufficient
  - Higher resolution and sampling rates
    - Improve image quality – reduce aliasing
  - More realistic materials and lighting
    - Improve realism
  - Increased scene complexity
    - More triangles!
    - Model smaller details of objects
- Conclusion: even with GPU advances there is a need for acceleration methods

Spatial Subdivision



Volume Rendering Techniques
http://graphics.stanford.edu/projects/volume

Spatial Data Structures

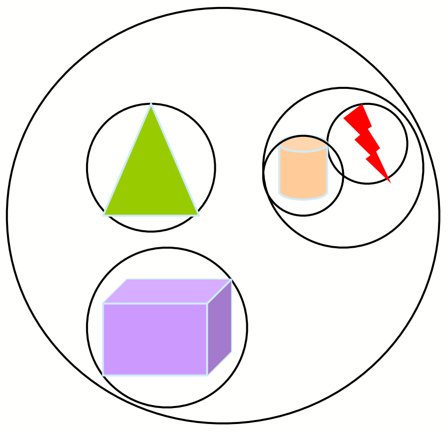
- **Spatial Data Structure**
  - Organizes geometry in an n-dimensional space
- Can be used to accelerate queries about geometry of objects
  - Do they overlap / intersect?
  - Used in culling methods, ray tracing, collision detection
- Spatial data structures often organized in a hierarchical structure
  - Topmost level encloses the level below it, etc.
  - Queries can improve from O(n) to O(log n)
  - Construction of most spatial data structures is expensive
    - Usually done as a pre-process

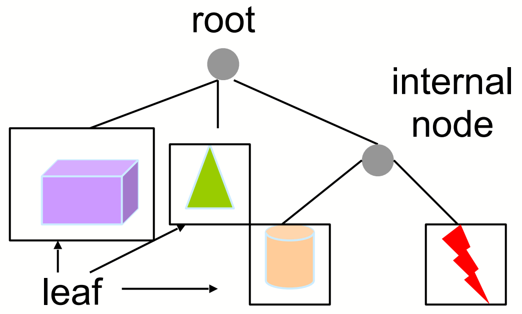
Spatial Data Structures (cont.)

- Examples in 3D
  - **Bounding Volume Hierarchies** (BVH)
  - **Spatial subdivision**
    - **Octrees**
    - **Binary Space Partitioning** (BSP) trees
- Two types
  - Regular – space is divided uniformly
  - Irregular – arbitrary subdivision of space

Bounding Volume Hierarchies

- Bounding Volume (BV) encloses a set of objects
  - Simpler shape than the objects it contains
    - More efficient tests can be performed against the BV
  - Does not get rendered
    - Used to speed rendering and different computations
    - Wireframe can be rendered for debugging collision issues
  - Examples are bounding spheres, AABBs, OBBs
- Scene is organized into a hierarchical tree structure
  - **Root** – topmost node
  - **Internals nodes** have pointers to their children
  - **Leaf nodes** – holds geometry to be rendered
    - No children
- Each node has BV that encloses the geometry in its entire sub tree
  - Root has BV that encloses the entire scene





Bounding Volume Hierarchies (cont.)

- Useful for intersection queries
  - Recursive testing starting at the root
    - Test against the BV of children
  - If ray misses a node's BV testing on the subtree ends
  - If a ray hits a leaf node's BV then the object geometry is tested
  - Nesting BVs allows avoiding testing large amounts of geometry
- Creation of BVH is generally done in a pre-process or initialization (scene definition) method
- Dynamic scenes
  - If an object contained in a BV is moved
    - Check whether it is still in the parent's BV
      - If so, then the BVH is still valid
      - If not, remove the object node and recompute the parent's BV
        - Node is recursively inserted back into the tree
        - Another method is to grow the BV (and parent's BVs up the tree)
  - **Temporal Bounding Volumes**
    - Create a bounding volume around the space where an object can move
      - e.g., a pendulum

Updating the Scene Graph

- Scene graph nodes often contain BVs
  - Scene graph should support several types of updates
    - Moving objects – change in modeling transformation
- Eberly suggests a recursive method to handle moving objects
  - 3D Game Engine Design
  - Update transforms on downward pass: from root to the leaves
    - Multiply matrices and store in the nodes
  - Update BVs on upward pass to the root of the scene graph
    - BV update at a node could affect the parent's BV
      - Need to compute BVs of children before parent's BV can be computed
- Methods to compute parent's BV from BVs of its children
  - Incrementally bound the children
    - Merge BV of first two children, merge that result with the next, etc.
  - Merging AABB is trivial
  - Merging bounding spheres – from Lecture 1