Cornell University CS 569: Interactive Computer Graphics

Spatial data structures for interactive graphics

Lecture 13

Applications 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)

Culling

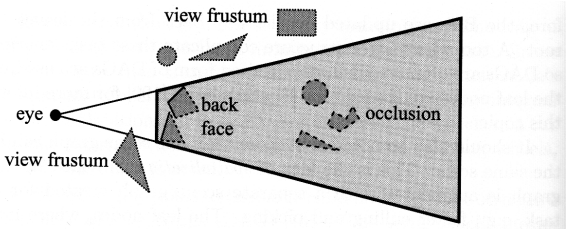


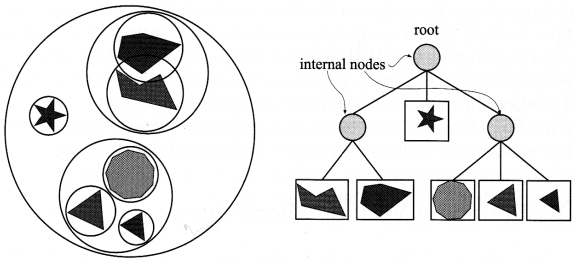
Figure 9.7. Different culling techniques. Culled geometry is dashed. *(Illustration after Cohen-Or et al. [135].)*

[Akenine-Möler & Haines]

Types of spatial data structures

- Object subdivision structures
  - bounding volumes
  - bounding volume hierarchies
- Spatial subdivision structures (regular)
  - regular grid
  - hierarchical grid
  - octree
- Spatial subdivision structures (irregular)
  - cell/portal structures
  - axis-aligned BSP tree (aka *k*-D tree)
  - polygon-aligned BSP tree

Bounding volume hierarchy (BVH)

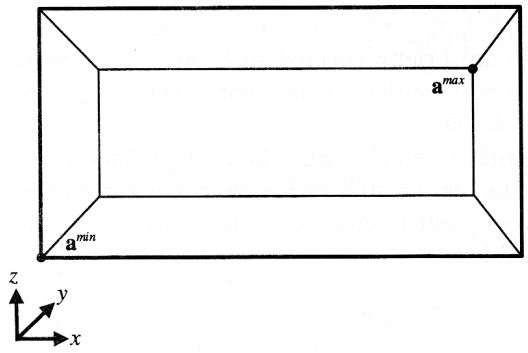


[Akenine-Möler & Haines]

Bounding volumes

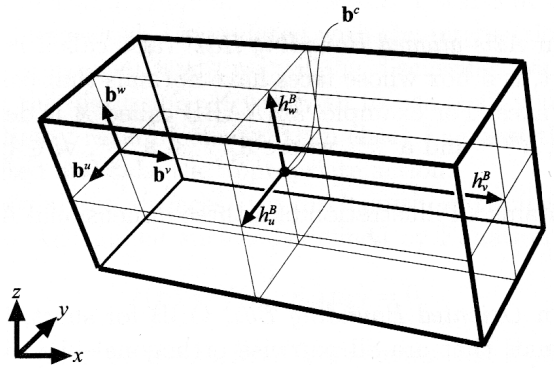
- Applications
  - ray intersection
  - frustum culling
  - collision detection
- Bounding volume choices
  - spheres
  - AABBs (axis-aligned boxes)
  - OBBs (oriented boxes)
  - *k*-DOPs (generalized boxes)
- Construction of BVHs
  - scene structure
  - top-down splitting
  - surface area heuristic

Axis-aligned bounding box (AABB)



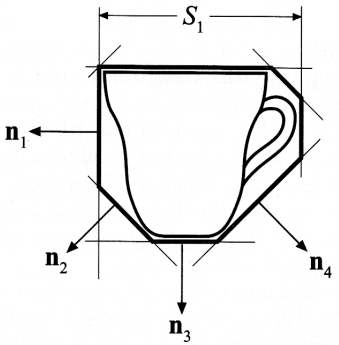
[Akenine-Möler & Haines]

Oriented Bounding Box (OBB)



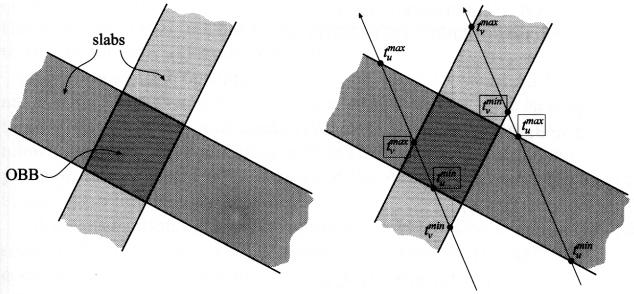
[Akenine-Möler & Haines]

k-DOP bounding volume



[Akenine-Möler & Haines]

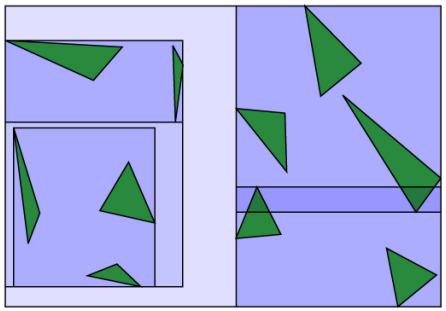
Figure 13.4. An example of a two-dimensional 8-DOP for a tea cup, with all normals, $\mathbf{n}_i$, shown along with the zero'th slab, S_1



[Akenine-Möler & Haines]

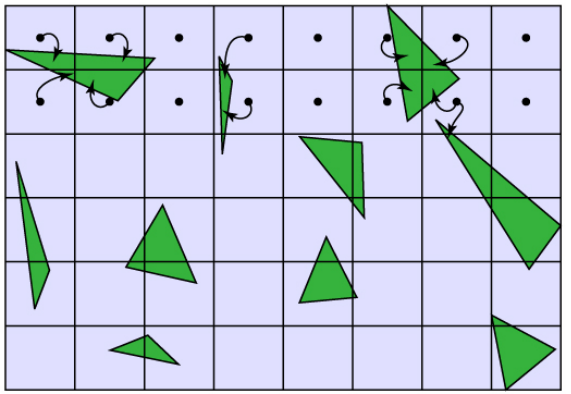
Figure 13.7. The left figure shows a two-dimensional OBB (*Oriented Bounding Box*) formed by two slabs, while the right shows two rays that are tested for intersection with the OBB. All t-values are shown, and they are subscripted with v for the light gray slab and with u for the other. The extreme t-values are marked with boxes. The left ray hits the OBB since $t^{min} < t^{max}$, and the right ray misses since $t^{max} < t^{min}$.

BVH construction example



Regular grid example

- Grid divides space, not objects



Octree

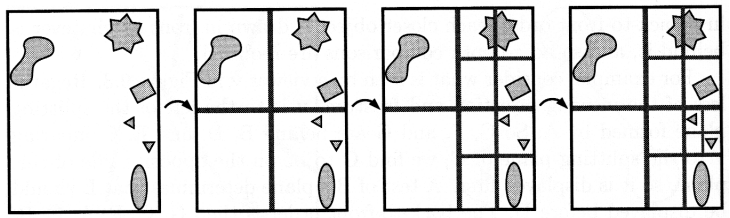


Figure 9.4. The construction of a quadtree (which is the two-dimensional version of an octree). The construction starts from the left by enclosing all objects in a bounding box. Then the boxes are recursively divided into four equal-sized boxes until each box (in this case) is empty or contains one object.

[Akenine-Möler & Haines]

AA BSP tree (aka *k*-D tree)

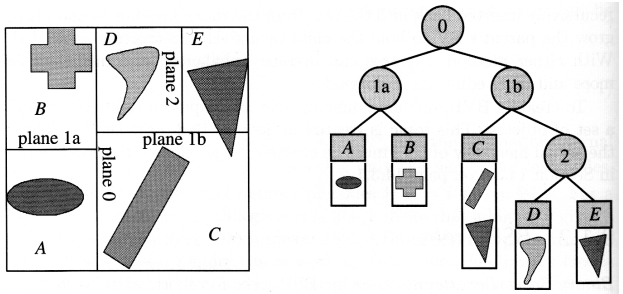
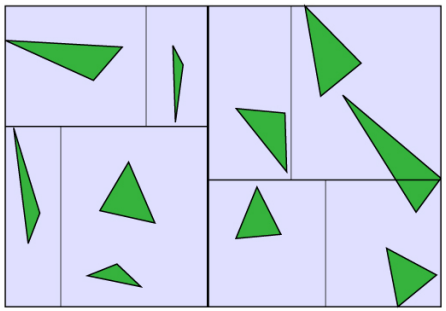


Figure 9.2. Axis-aligned BSP tree. In this example, the space partitions are allowed to be anywhere along the axis, not just at its midpoint. The spatial volumes formed are labeled A through E. The tree on the right shows the underlying BSP data structure. Each leaf node represents an area, with that area's contents shown beneath it. Note that the triangle is in the object list for two areas, C and E, because it overlaps both.

[Akenine-Möler & Haines]

Non-regular space subdivision

- *k*-d Tree
  - subdivides space, like grid
  - adaptive, like BVH



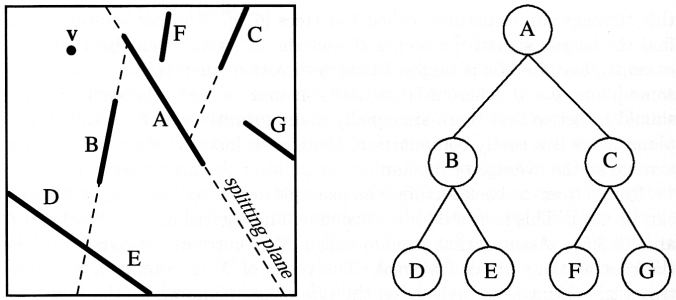


Figure 9.3. Polygon-aligned BSP tree. Polygons A through G are shown from above. Space is first split by polygon A, then each half-space is split separately by B and C. The splitting plane formed by polygon B intersects the polygon in the lower left corner, splitting it into separate polygons D and E. The BSP tree formed is shown on the right.

[Akenine-Möler & Haines]

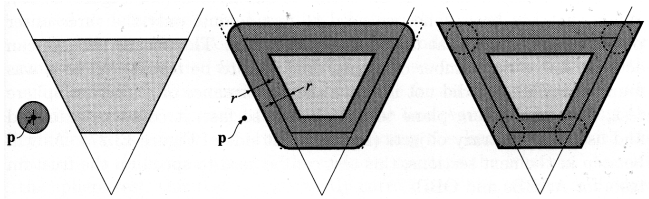


Figure 13.26. At the left, a frustum and a sphere are shown. The exact frustum/sphere test can be formulated as testing **p** against the dark and light gray volumes in the middle figure. At the right is a reasonable approximation of the volumes in the middle. If the center of the sphere is located outside a rounded corner, but inside all outer planes, then it will be incorrectly classified as intersecting even though it is outside the frustum.

[Akenine-Möler & Haines]

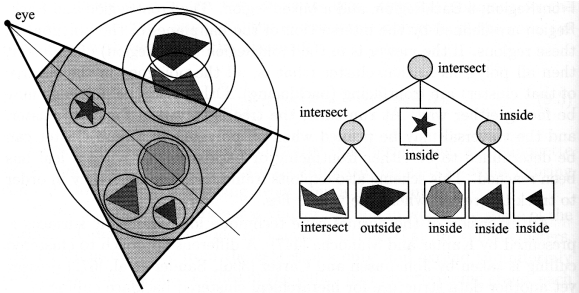


Figure 9.11. A set of geometry and its bounding volumes (spheres) are shown on the left. This scene is rendered with view frustum culling from the point of the eye. The BVH is shown on the right. The BV of the root intersects the frustum, and the traversal continues with testing its children's BVs. The BV of the left subtree intersects, and one of that subtree's children intersects (and thus is rendered), and the BV of the other child is outside and therefore is not sent through the pipeline. The BV of the middle subtree of the root is totally inside and is rendered immediately. The BV of the right subtree of the root is also fully inside, and the entire subtree can therefore be rendered without further tests.

[Akenine-Möler & Haines]

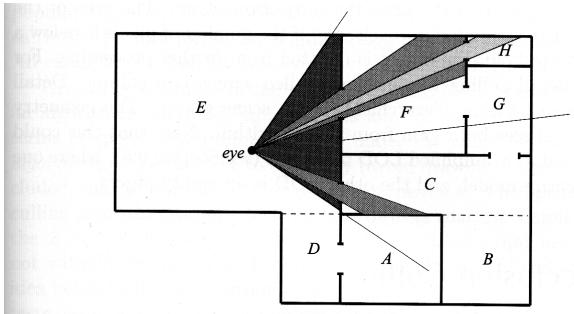


Figure 9.12. Portal culling: Cells are enumerated from A to H, and portals are openings that connect the cells. Only geometry seen through the portals is rendered.

[Akenine-Möler & Haines]

Bounding volumes in direction space

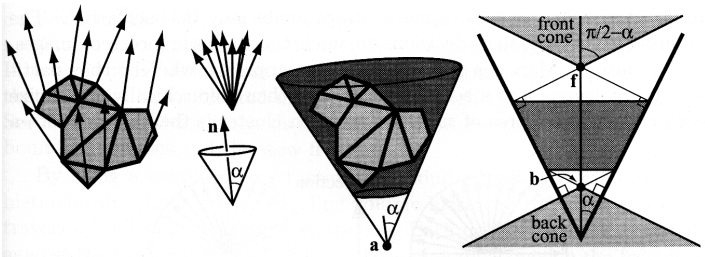


Figure 9.9. Left: A set of polygons and their normals. Middle-left: The normals are collected (top), and a minimal cone (bottom), defined by one normal **n**, and a half-angle, α, is constructed. Middle-right: The cone is anchored at a point **a**, and truncated so that it also contains all points on the polygons. Right: A cross section of a truncated cone. The light gray region on the top is the frontfacing cone, and the light gray region at the bottom is the backfacing cone. The points **f** and **b** are respectively the apexes of the front and backfacing cones.

[Akenine-Möler & Haines]