COM S 567 Physically Based Animation

Collision Detection

(Blackboard Helper Slides)

Doug James

Cornell University

Spring 2007

Narrow Phase Tests

Convex Bounding Volumes

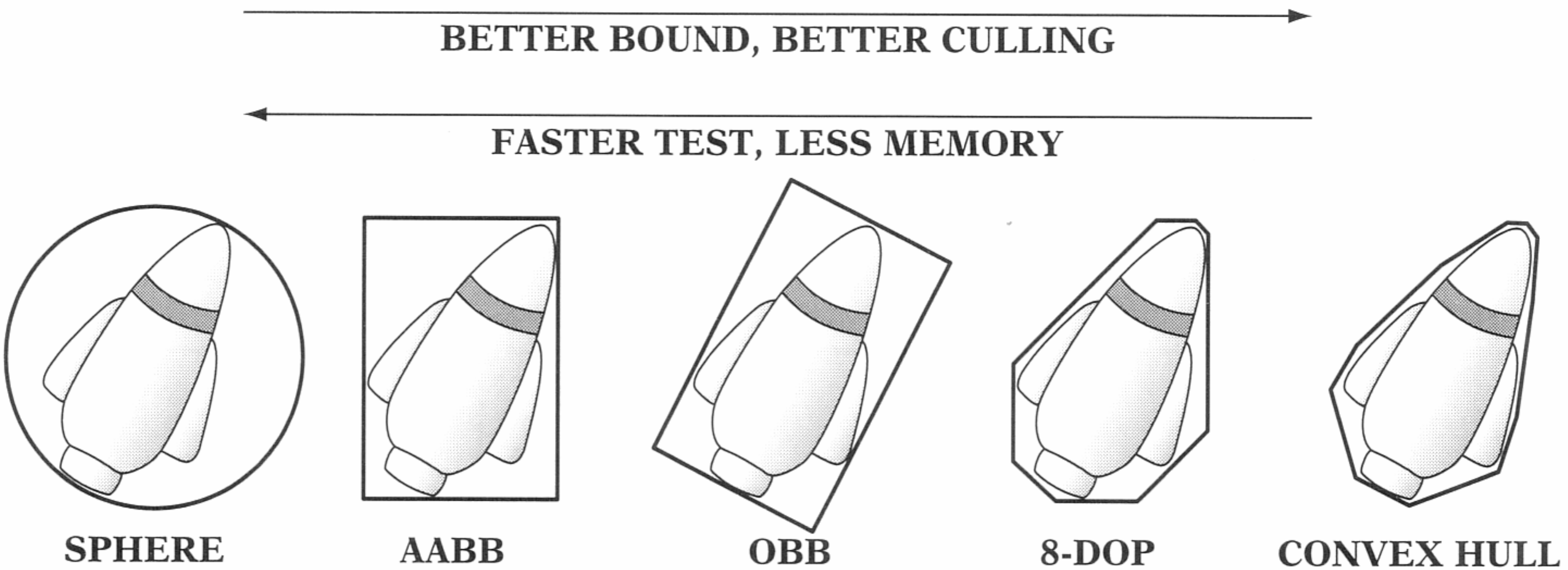


Figure 4.2 Types of bounding volumes: sphere, axis-aligned bounding box (AABB), oriented bounding box (OBB), eight-direction discrete orientation polytope (8-DOP), and convex hull.

Intersect(sphere, polygon)

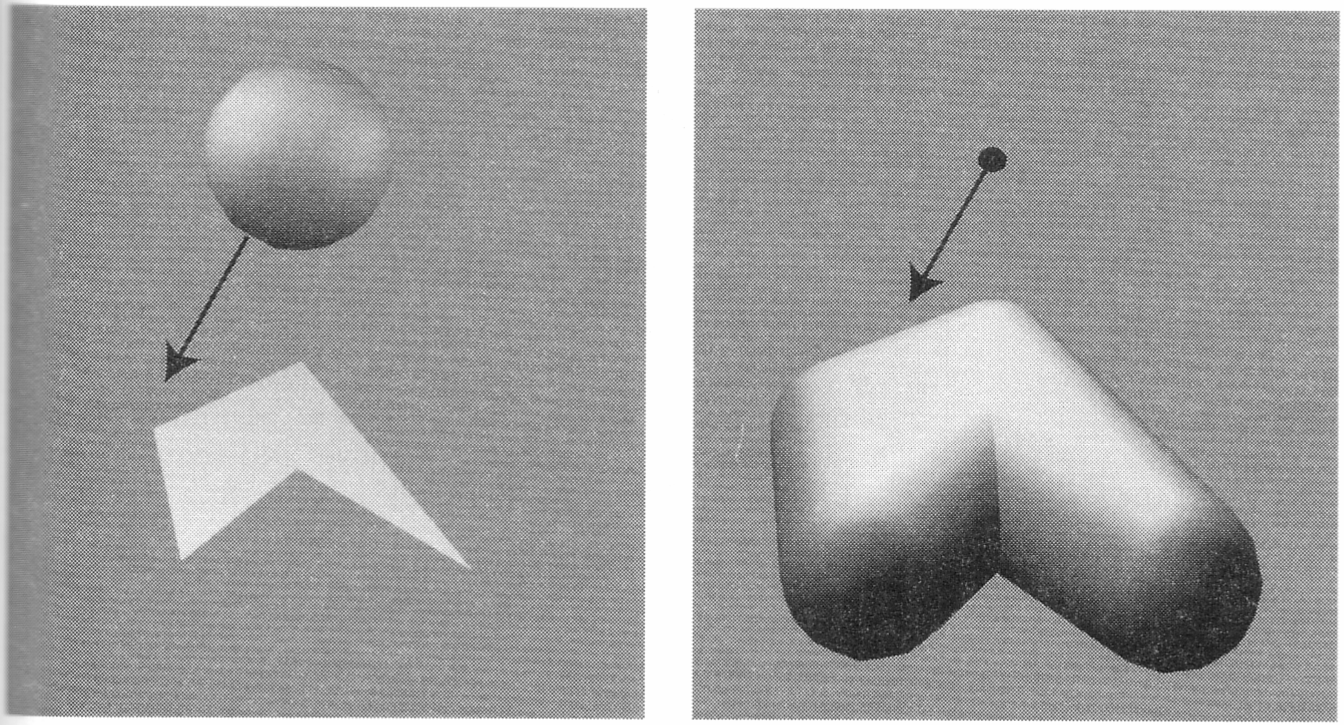


Figure 13.33. In the left figure, a sphere moves towards a polygon. In the right figure, a ray shoots at an "inflated" version of the polygon. The two intersection tests are equivalent.

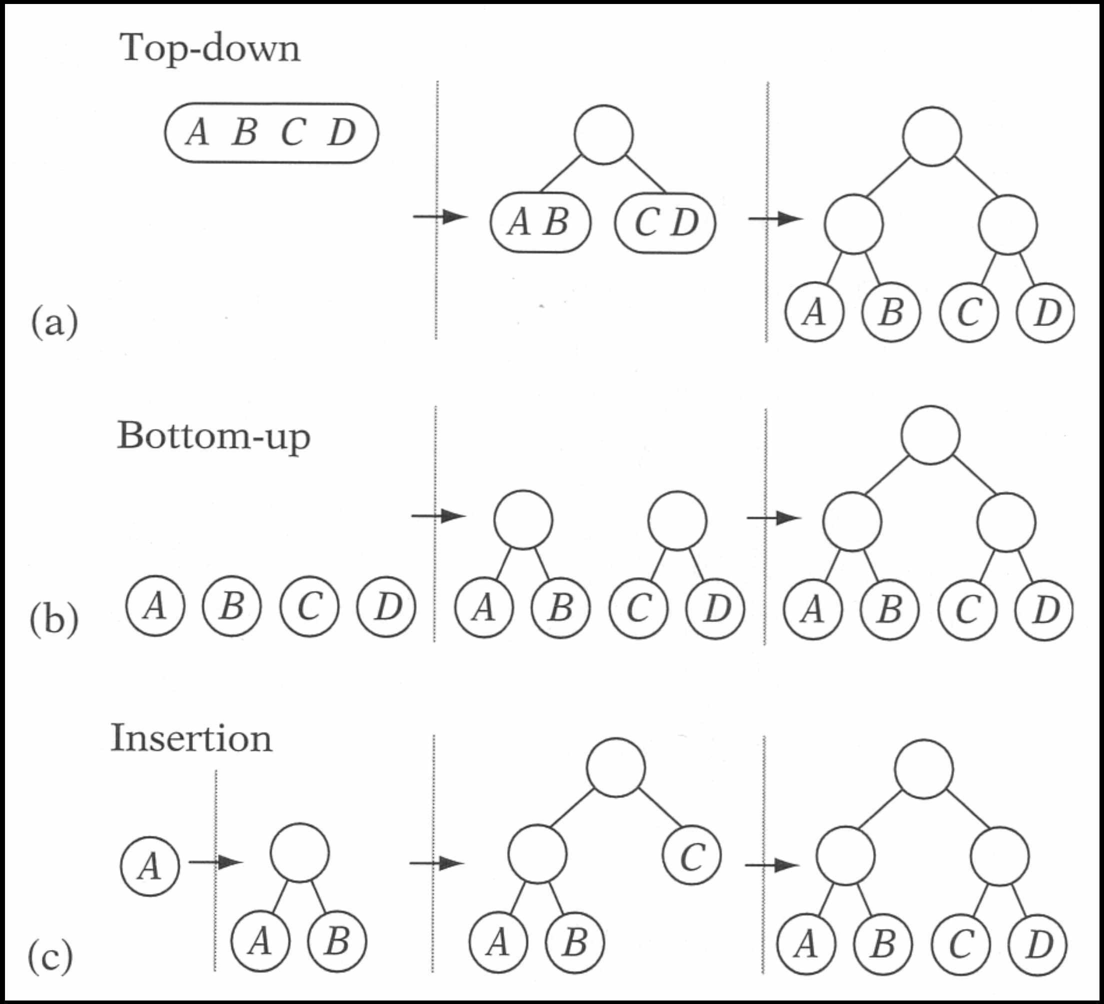
Robust Cloth [Bridson et al. 2002]

..\01 Intro\bridson_curtain.mov

Table 4.1 The 15 separating axis tests needed to determine OBB-OBB intersection. Superscripts indicate which OBB the value comes from.

L	$\lvert T \cdot L \rvert$	r_A	r_B
$\mathbf{u}_0^A$	$\lvert \mathbf{t}_0 \rvert$	$\mathbf{e}_0^A$	$\mathbf{e}_0^B \lvert r_{00} \rvert + \mathbf{e}_1^B \lvert r_{01} \rvert + \mathbf{e}_2^B \lvert r_{02} \rvert$
$\mathbf{u}_1^A$	$\lvert \mathbf{t}_1 \rvert$	$\mathbf{e}_1^A$	$\mathbf{e}_0^B \lvert r_{10} \rvert + \mathbf{e}_1^B \lvert r_{11} \rvert + \mathbf{e}_2^B \lvert r_{12} \rvert$
$\mathbf{u}_2^A$	$\lvert \mathbf{t}_2 \rvert$	$\mathbf{e}_2^A$	$\mathbf{e}_0^B \lvert r_{20} \rvert + \mathbf{e}_1^B \lvert r_{21} \rvert + \mathbf{e}_2^B \lvert r_{22} \rvert$
$\mathbf{u}_0^B$	$\lvert \mathbf{t}_0 r_{00} + \mathbf{t}_1 r_{10} + \mathbf{t}_2 r_{20} \rvert$	$\mathbf{e}_0^A \lvert r_{00} \rvert + \mathbf{e}_1^A \lvert r_{10} \rvert + \mathbf{e}_2^A \lvert r_{20} \rvert$	$\mathbf{e}_0^B$
$\mathbf{u}_1^B$	$\lvert \mathbf{t}_0 r_{01} + \mathbf{t}_1 r_{11} + \mathbf{t}_2 r_{21} \rvert$	$\mathbf{e}_0^A \lvert r_{01} \rvert + \mathbf{e}_1^A \lvert r_{11} \rvert + \mathbf{e}_2^A \lvert r_{21} \rvert$	$\mathbf{e}_1^B$
$\mathbf{u}_2^B$	$\lvert \mathbf{t}_0 r_{02} + \mathbf{t}_1 r_{12} + \mathbf{t}_2 r_{22} \rvert$	$\mathbf{e}_0^A \lvert r_{02} \rvert + \mathbf{e}_1^A \lvert r_{12} \rvert + \mathbf{e}_2^A \lvert r_{22} \rvert$	$\mathbf{e}_2^B$
$\mathbf{u}_0^A \times \mathbf{u}_0^B$	$\lvert \mathbf{t}_2 r_{10} - \mathbf{t}_1 r_{20} \rvert$	$\mathbf{e}_1^A \lvert r_{20} \rvert + \mathbf{e}_2^A \lvert r_{10} \rvert$	$\mathbf{e}_1^B \lvert r_{02} \rvert + \mathbf{e}_2^B \lvert r_{01} \rvert$
$\mathbf{u}_0^A \times \mathbf{u}_1^B$	$\lvert \mathbf{t}_2 r_{11} - \mathbf{t}_1 r_{21} \rvert$	$\mathbf{e}_1^A \lvert r_{21} \rvert + \mathbf{e}_2^A \lvert r_{11} \rvert$	$\mathbf{e}_0^B \lvert r_{02} \rvert + \mathbf{e}_2^B \lvert r_{00} \rvert$
$\mathbf{u}_0^A \times \mathbf{u}_2^B$	$\lvert \mathbf{t}_2 r_{12} - \mathbf{t}_1 r_{22} \rvert$	$\mathbf{e}_1^A \lvert r_{22} \rvert + \mathbf{e}_2^A \lvert r_{12} \rvert$	$\mathbf{e}_0^B \lvert r_{01} \rvert + \mathbf{e}_1^B \lvert r_{00} \rvert$
$\mathbf{u}_1^A \times \mathbf{u}_0^B$	$\lvert \mathbf{t}_0 r_{20} - \mathbf{t}_2 r_{00} \rvert$	$\mathbf{e}_0^A \lvert r_{20} \rvert + \mathbf{e}_2^A \lvert r_{00} \rvert$	$\mathbf{e}_1^B \lvert r_{12} \rvert + \mathbf{e}_2^B \lvert r_{11} \rvert$
$\mathbf{u}_1^A \times \mathbf{u}_1^B$	$\lvert \mathbf{t}_0 r_{21} - \mathbf{t}_2 r_{01} \rvert$	$\mathbf{e}_0^A \lvert r_{21} \rvert + \mathbf{e}_2^A \lvert r_{01} \rvert$	$\mathbf{e}_0^B \lvert r_{12} \rvert + \mathbf{e}_2^B \lvert r_{10} \rvert$
$\mathbf{u}_1^A \times \mathbf{u}_2^B$	$\lvert \mathbf{t}_0 r_{22} - \mathbf{t}_2 r_{02} \rvert$	$\mathbf{e}_0^A \lvert r_{22} \rvert + \mathbf{e}_2^A \lvert r_{02} \rvert$	$\mathbf{e}_0^B \lvert r_{11} \rvert + \mathbf{e}_1^B \lvert r_{10} \rvert$
$\mathbf{u}_2^A \times \mathbf{u}_0^B$	$\lvert \mathbf{t}_1 r_{00} - \mathbf{t}_0 r_{10} \rvert$	$\mathbf{e}_0^A \lvert r_{10} \rvert + \mathbf{e}_1^A \lvert r_{00} \rvert$	$\mathbf{e}_1^B \lvert r_{22} \rvert + \mathbf{e}_2^B \lvert r_{21} \rvert$
$\mathbf{u}_2^A \times \mathbf{u}_1^B$	$\lvert \mathbf{t}_1 r_{01} - \mathbf{t}_0 r_{11} \rvert$	$\mathbf{e}_0^A \lvert r_{11} \rvert + \mathbf{e}_1^A \lvert r_{01} \rvert$	$\mathbf{e}_0^B \lvert r_{22} \rvert + \mathbf{e}_2^B \lvert r_{20} \rvert$
$\mathbf{u}_2^A \times \mathbf{u}_2^B$	$\lvert \mathbf{t}_1 r_{02} - \mathbf{t}_0 r_{12} \rvert$	$\mathbf{e}_0^A \lvert r_{12} \rvert + \mathbf{e}_1^A \lvert r_{02} \rvert$	$\mathbf{e}_0^B \lvert r_{21} \rvert + \mathbf{e}_1^B \lvert r_{20} \rvert$

BVH Construction



From [Ericson 2005]

AABB parent-child relationship

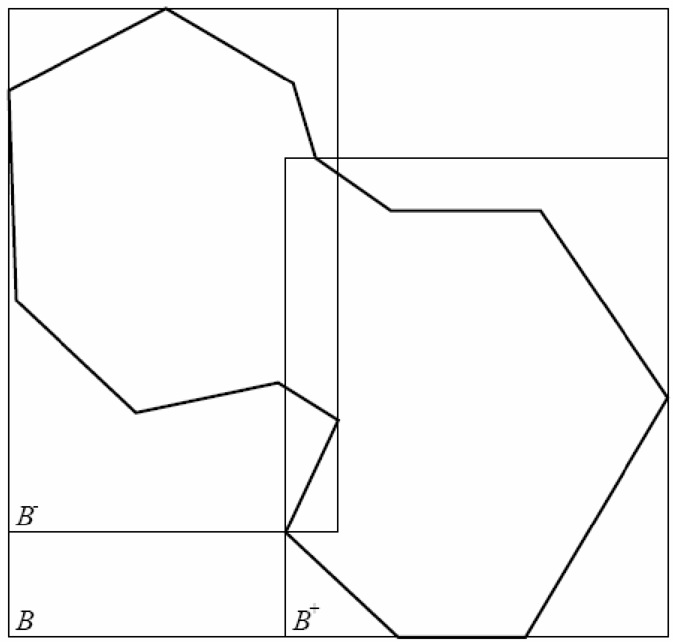
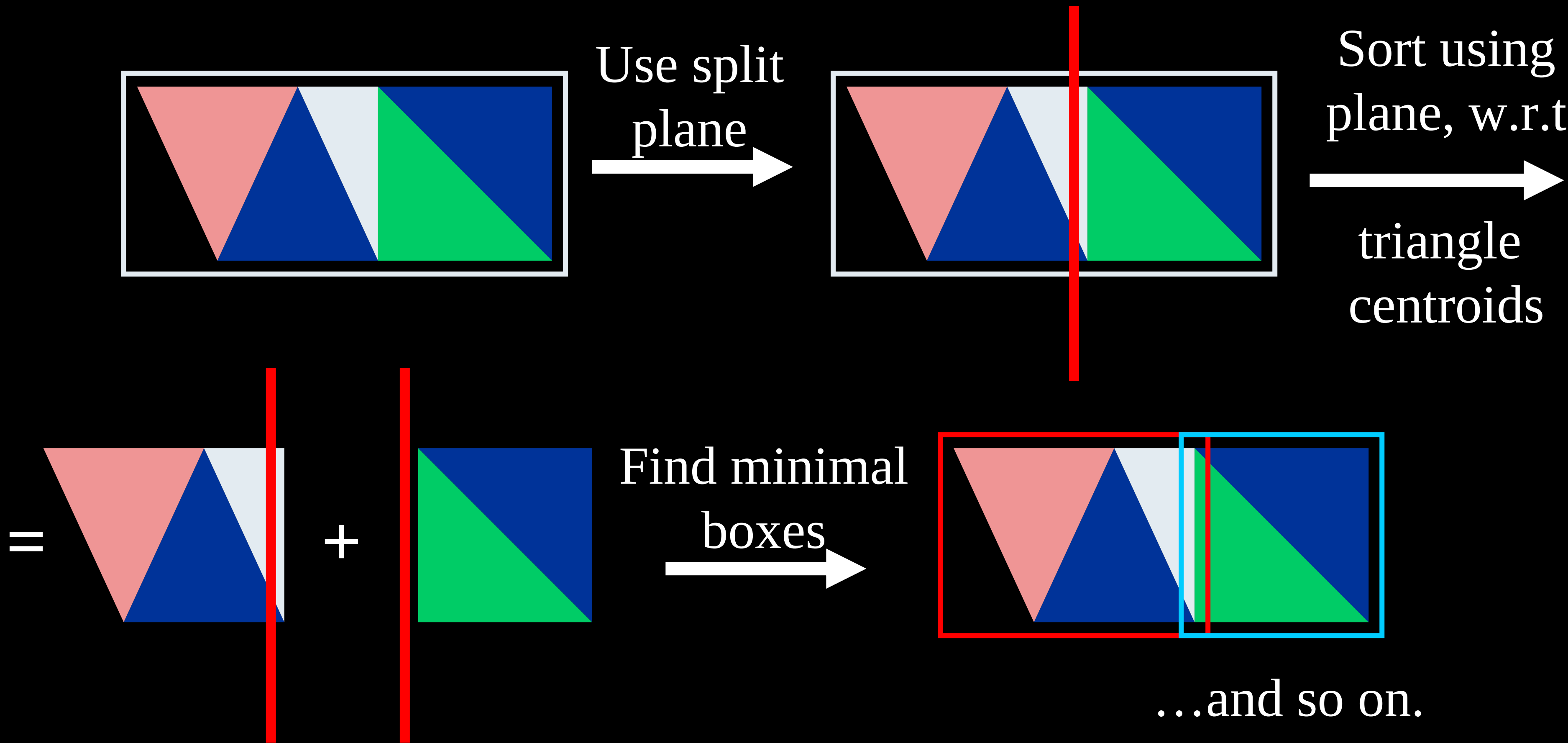


Figure 3: The smallest AABB of a set of primitives encloses the smallest AABBs of the subsets in a partition of the set.

Top-down Construction

- Recursively split & bound geometric primitives



From [Moller and Haines 2002]

Bottom-up construction

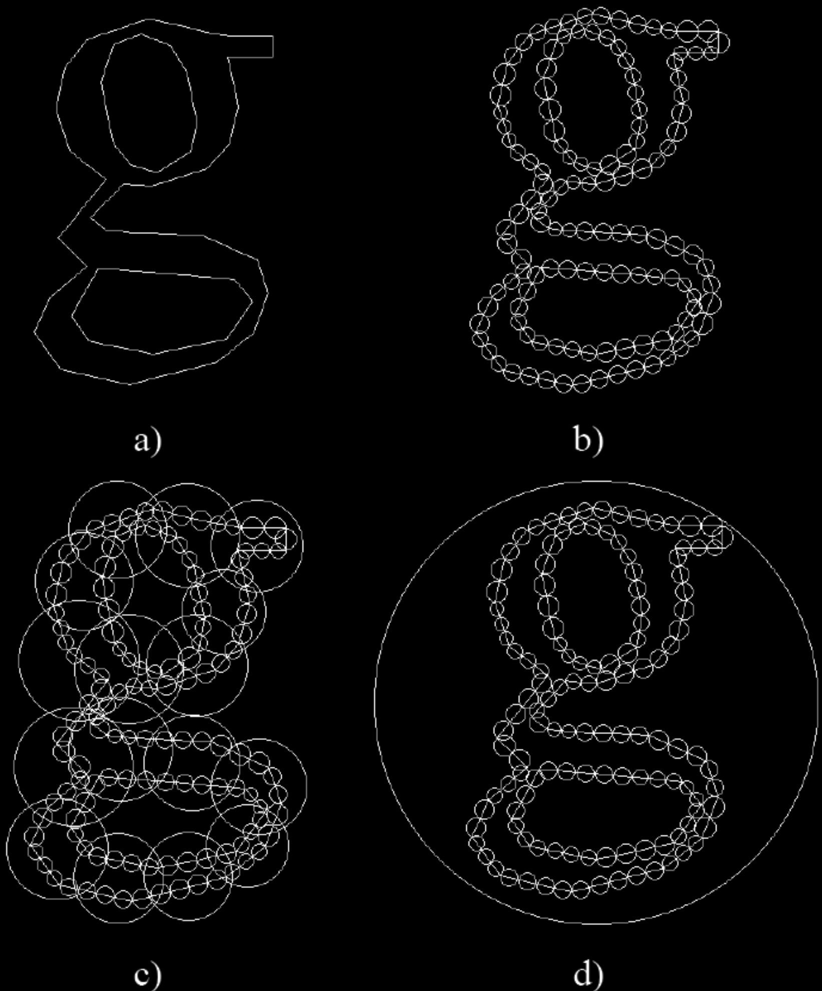


Figure 1. The bounding tree for an object.

From [Quinlan 94]

- Wrap all primitives
  - Tile geometry [Quinlan 94]
- Parent bounds can enclose child bounds for fitting speed (layered hierarchy)

BVH-BVH Overlap Tests

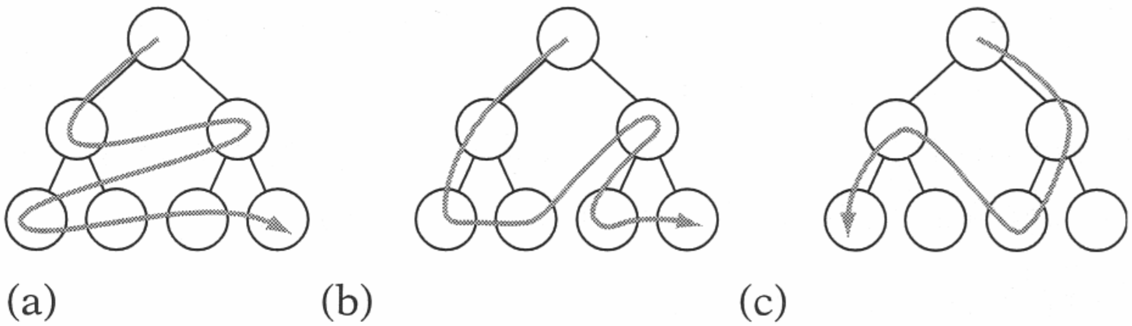


Figure 6.4 (a) Breadth-first search, (b) depth-first search, and (c) one possible best-first search ordering.

From [Ericson 2005]

- Also proximity (distance) queries

Pseudocode: BVH vs BVH Test

NOTE: Should test for overlap here.

```
FindFirstHitCD(A, B)
returns ({TRUE, FALSE});
if(isLeaf(A) and isLeaf(B))
   for each triangle pair T_A ∈ A_c and T_B ∈ B_c
      if(overlap(T_A, T_B)) return TRUE;
else if(isNotLeaf(A) and isNotLeaf(B))
   if(Volume(A) > Volume(B))
      for each child C_A ∈ A_c
         FindFirstHitCD(C_A, B)
   else
      for each child C_B ∈ B_c
         FindFirstHitCD(A, C_B)
else if(isLeaf(A) and isNotLeaf(B))
   for each child C_B ∈ B_c
      FindFirstHitCD(C_B, A)
else
   for each child C_A ∈ A_c
      FindFirstHitCD(C_A, B)
return FALSE;
```

Pseudocode deals with 4 cases:

1) Leaf against leaf node
2) Internal node against internal node
3) Internal against leaf
4) Leaf against internal

From [Moller and Haines 2002]

Comments on pseudocode

- The code terminated when it found the first triangle pair that collided
- Simple to modify code to continue traversal and put each pair in a list

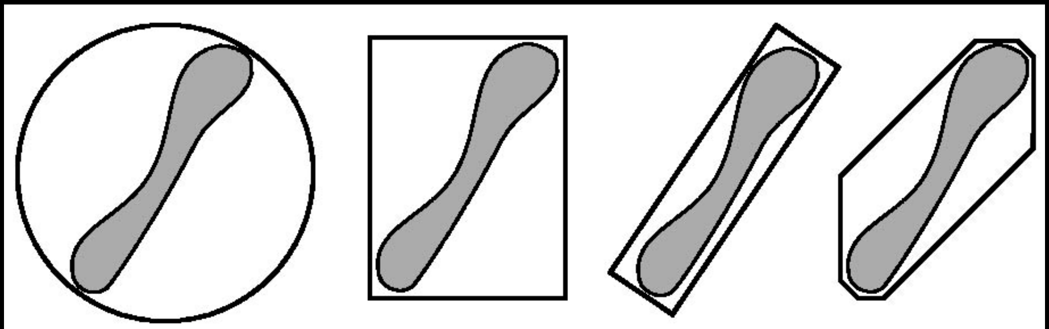
- Reasonably simple to include rotations for objects as well
- Note that if we use AABB for both BVHs, then the AABB-AABB test becomes a AABB-OBB test

From [Moller and Haines 2002]

Tradeoffs

n_v :	number of BV/BV overlap tests
c_v :	cost for a BV/BV overlap test
n_p :	number of primitive pairs tested for overlap
c_p :	cost for testing whether two primitives overlap
n_u :	number of BVs updated due to the model's motion
c_u :	cost for updating a BV

- The choice of BV
  - AABB, OBB, k-DOP, sphere
- In general, the tighter BV, the slower test



- Less tight BV, gives more triangle-triangle tests in the end (if needed)
- Cost function:

$$t = n_v c_v + n_p c_p + n_u c_u$$

From [Moller and Haines 2002]

BVH-BVH Collision Front Tracking

- [Klosowski 1998; Li and Chen 1998]

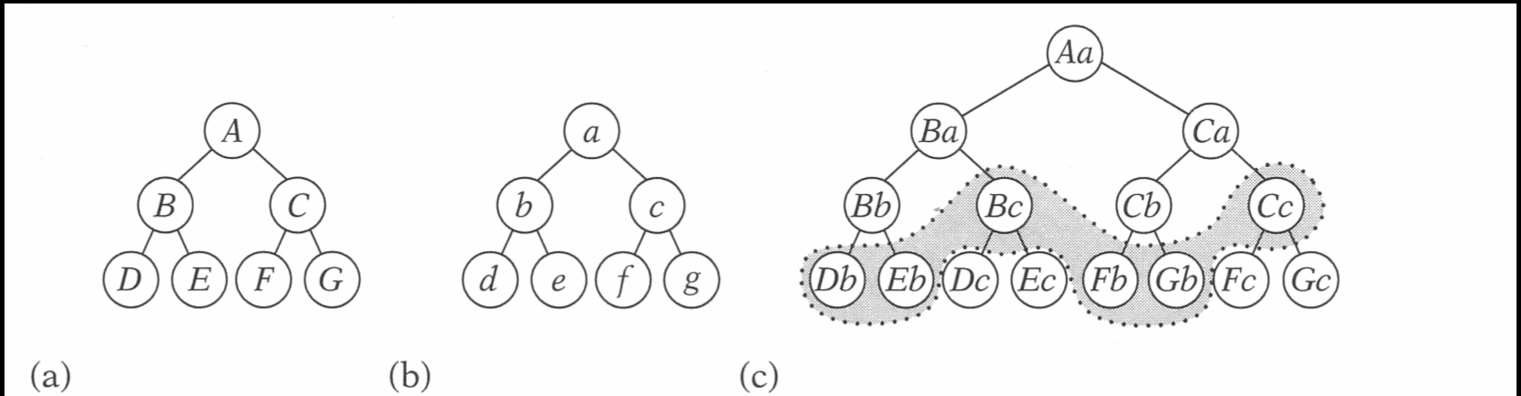
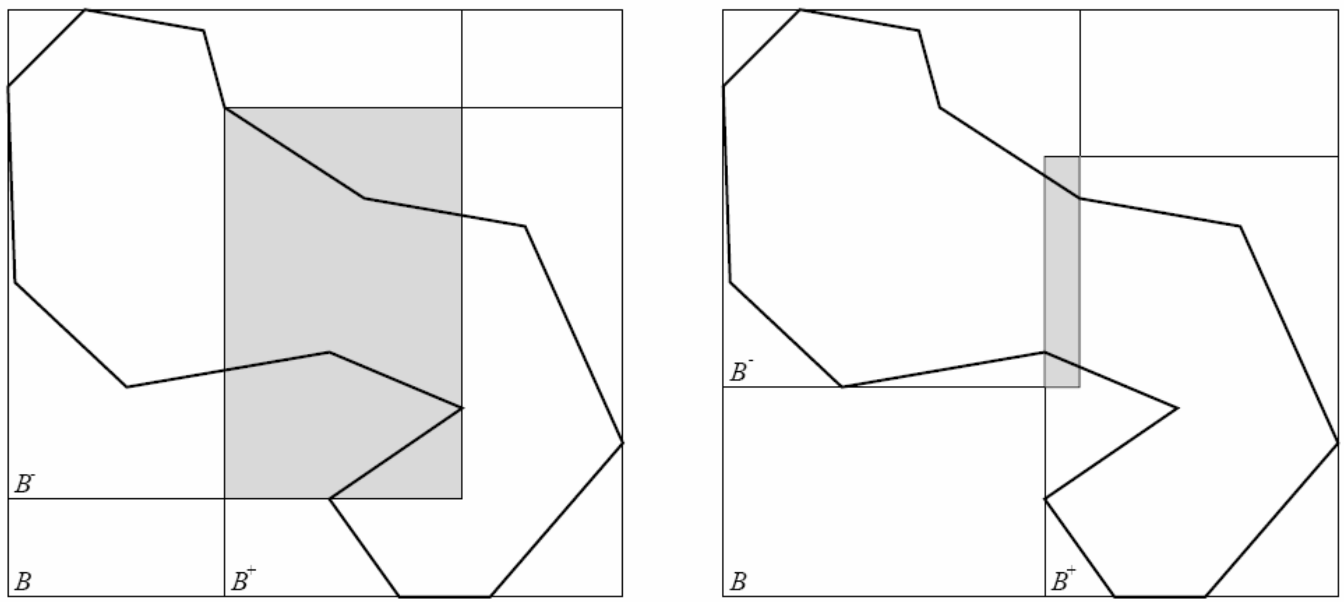


Figure 6.9 (a) The hierarchy for one object. (b) The hierarchy for another object. (c) The collision tree formed by an alternating traversal. The shaded area indicates a front in which the objects are (hypothetically) found noncolliding.

Other issues

- Time-critical tests
- Space-time bounds
- Good reference: Phillip Hubbard's thesis
  - P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210, July 1996.

Refitting after deformation



(a) Refitted (b) Rebuilt

Figure 4: Refitting vs. rebuilding the model in Figure 3 after a deformation

From [van den Bergen 1998]

Broad Phase Tests

Spatial Subdivisions: Uniform Grids

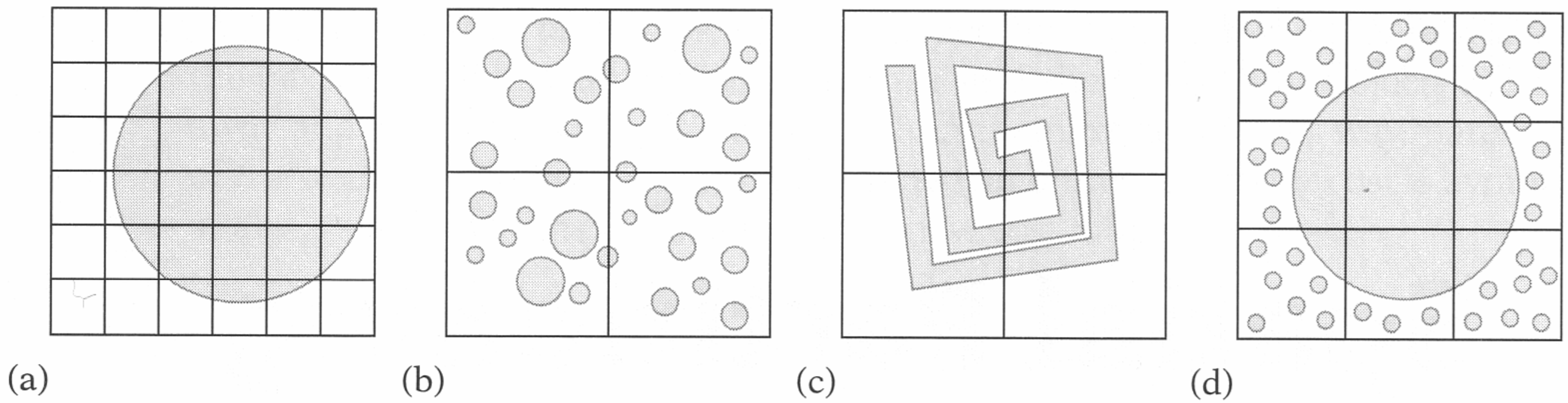


Figure 7.1 Issues related to cell size. (a) A grid that is too fine. (b) A grid that is too coarse (with respect to object size). (c) A grid that is too coarse (with respect to object complexity). (d) A grid that is both too fine and too coarse.

Hierarchical Grids [Brian Mirtich]

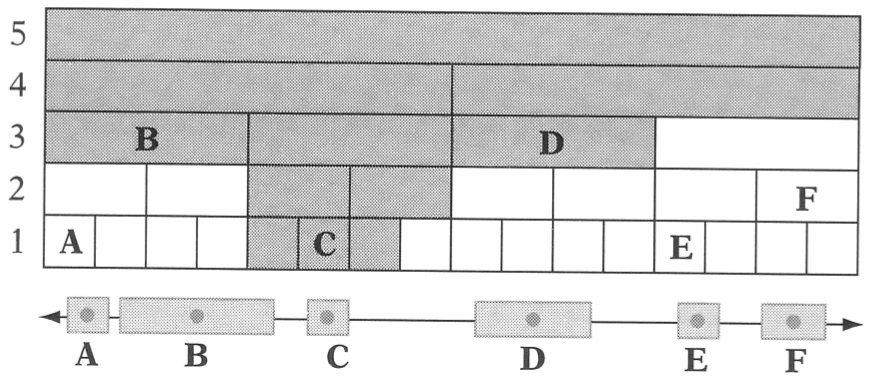


Figure 7.8 A small 1D hierarchical grid. Six objects, *A* through *F*, have each been inserted in the cell containing the object center point, on the appropriate grid level. The shaded cells are those that must be tested when performing a collision check for object *C*.

Hierarchical Grids [Brian Mirtich]

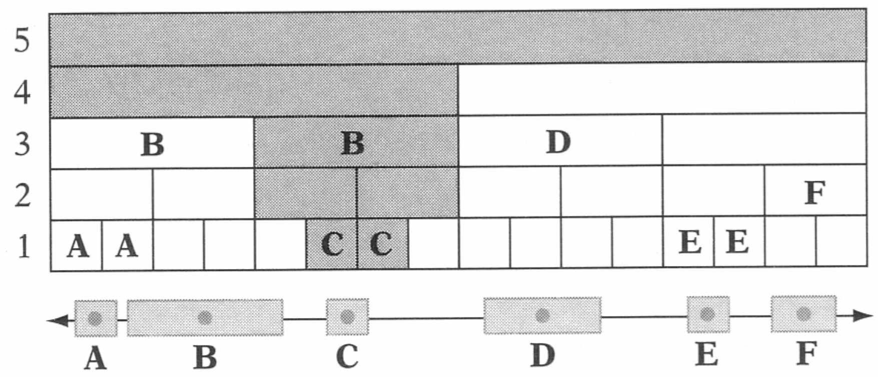
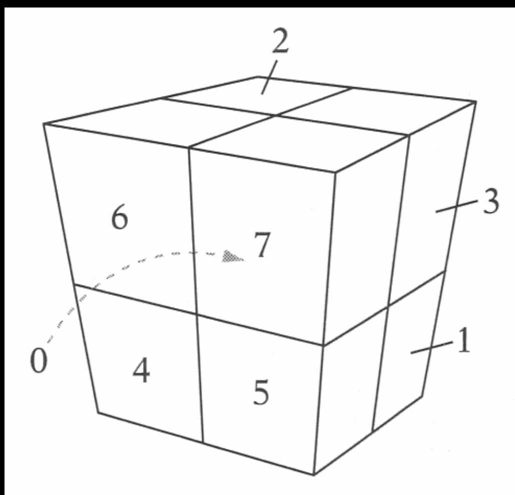


Figure 7.9 In Mirtich's (first) scheme, objects are inserted in all cells overlapped at the insertion level. As in Figure 7.8, the shaded cells indicate which cells must be tested when performing a collision check for object *C*.

Octrees (and Quadtrees)



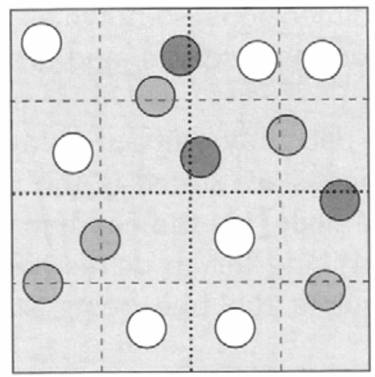


Figure 7.11 A quadtree node with the first level of subdivision shown in black dotted lines, and the following level of subdivision in gray dashed lines. Dark gray objects overlap the first-level dividing planes and become stuck at the current level. Medium gray objects propagate one level down before becoming stuck. Here, only the white objects descend two levels.

Linear Octrees

- Sparse representation
- Avoid pointer storage
- Use Morton index

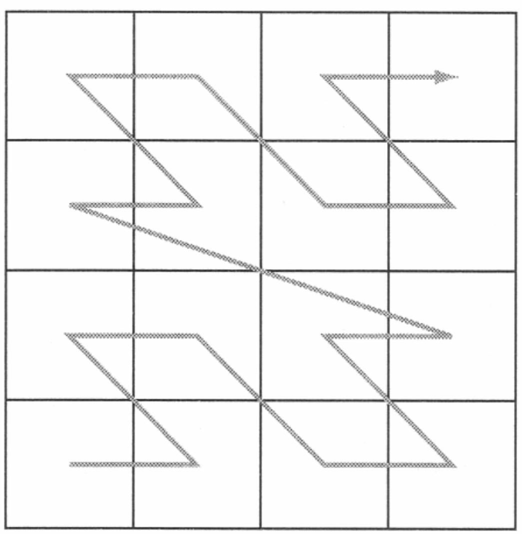


Figure 7.12 The cells of a 4 × 4 grid given in Morton order.

Many other spatial partitions…

- Kd-trees

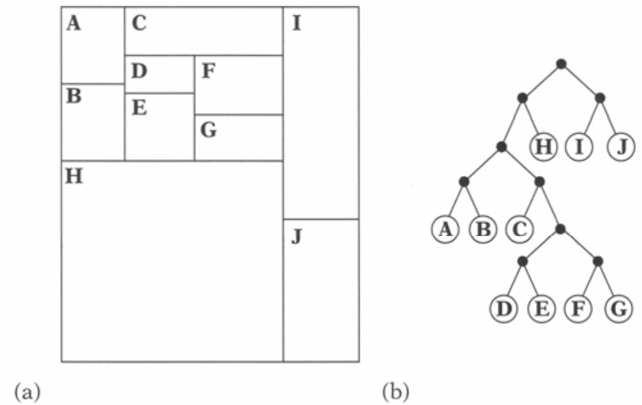


Figure 7.14 A 2D *k*-d tree. (a) The spatial decomposition. (b) The *k*-d tree layout.

- Hybrids

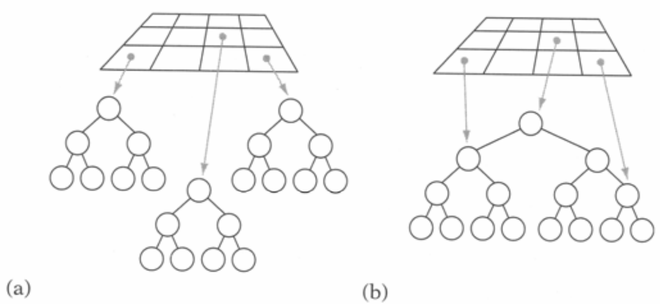


Figure 7.15 (a) A grid of trees, each grid cell containing a separate tree. (b) A grid indexing into a single tree hierarchy, each grid cell pointing into the tree at which point traversal should start.

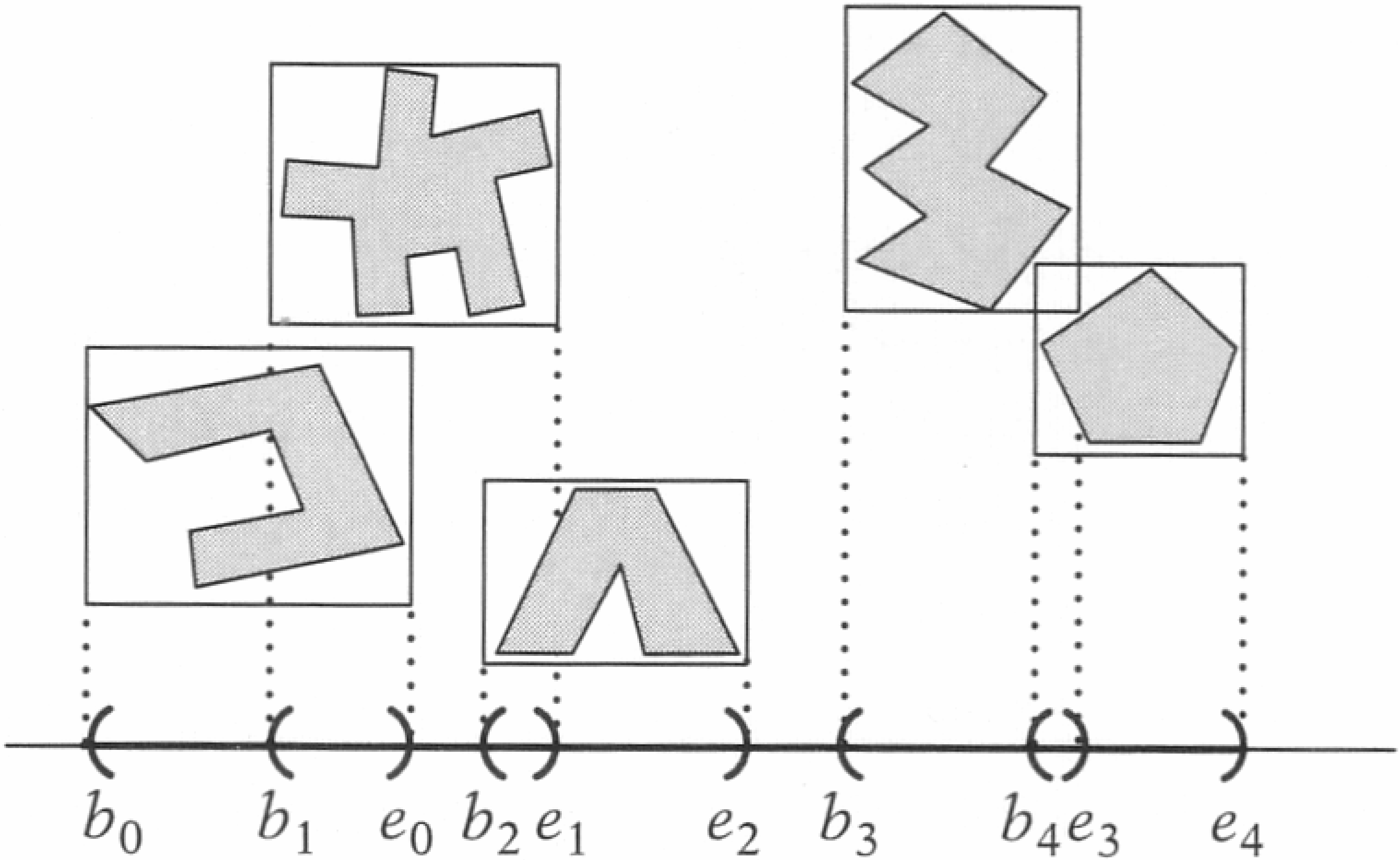
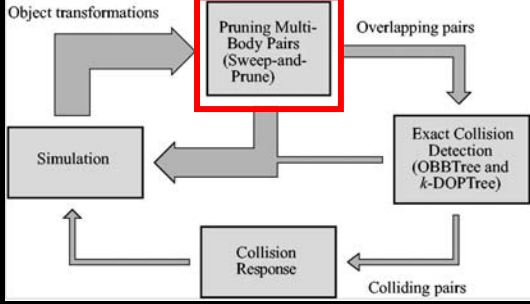


Figure 7.19 Projected AABB intervals on the *x* axis.

CD between many objects



- Why needed?
- Consider several hundreds of rocks tumbling down a slope…
- This system is often called ”First-Level CD”
- We execute this system because we want to execute the 2nd system less frequently
- Assume high frame-to-frame coherency
  – Means that object is close to where it was previous frame
  – Reasonable

From [Moller and Haines 2002]

Sweep-and-prune algorithm [by Ming Lin]

- Assume objects may translate and rotate
- Then we can find a minimal cube, which is guaranteed to contain object for all rotations
- Do collision overlap three times
  - One for x,y, and z-axes
- Let's concentrate on one axis at a time
- Each cube on this axis is an interval, from s_i to e_i, where i is cube number

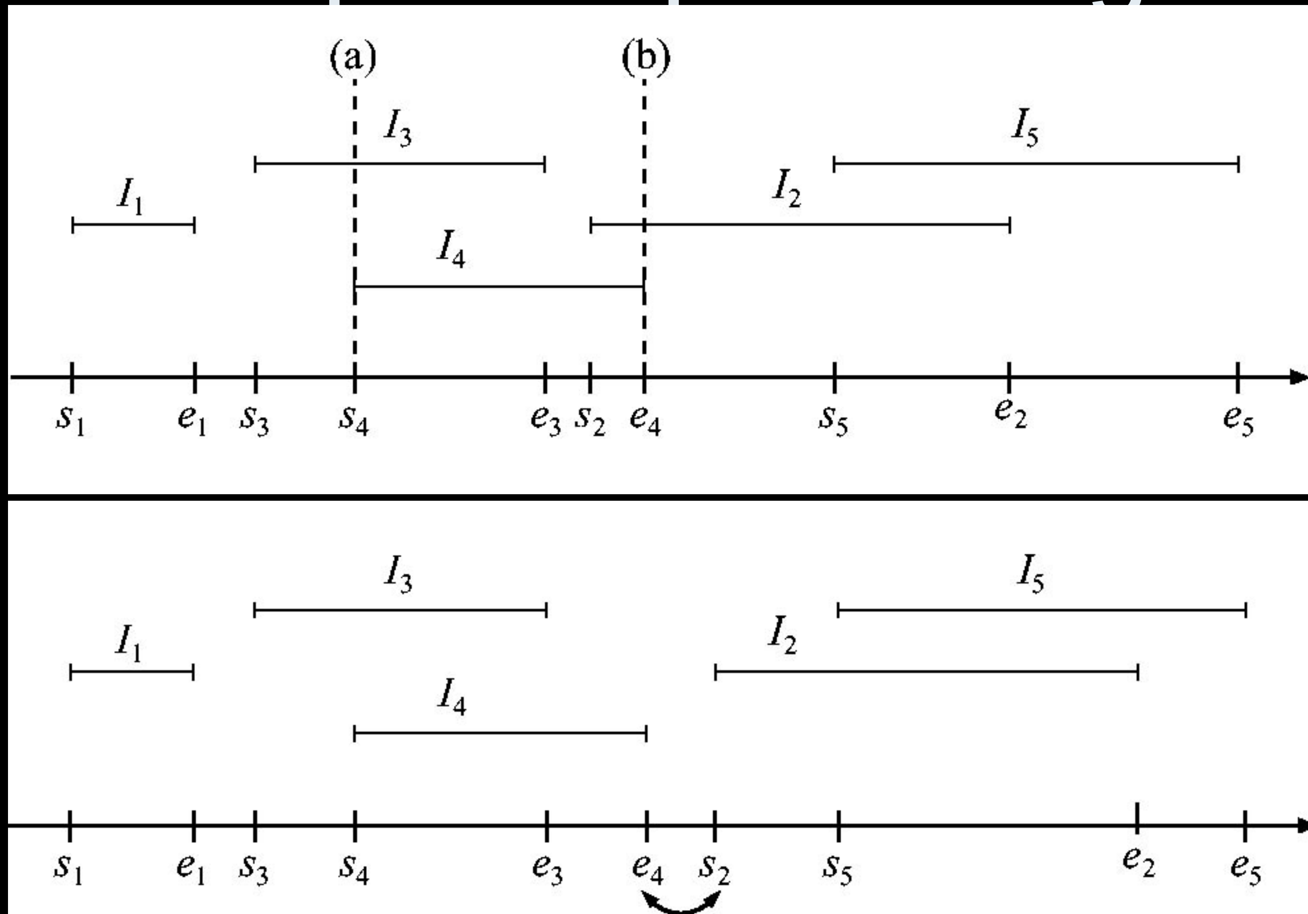
From [Moller and Haines 2002]

Sweep-and-prune algorithm

- Sort all s_i and e_i into a list
- Traverse list from start to end
- When an s is encountered, mark corresponding interval as active in an **`active_interval_list`**
- When an e is encountered, delete the interval in **`active_interval_list`**
- All intervals in **`active_interval_list`** are overlapping!

From [Moller and Haines 2002]

Sweep-and-prune algorithm



- Keep a boolean for each pair of intervals
- Invert when sort order changes
- If all boolean for all three axes are true, → overlap

From [Moller and Haines 2002]

Sweep-and-prune algorithm

- Now sorting is expensive: O(n*log n)
- But, exploit frame-to-frame coherency!
- The list is not expected to change much
- Therefore, ”resort” with bubble-sort, or insertion-sort
- Expected: O(n)

From [Moller and Haines 2002]

Failure Mode

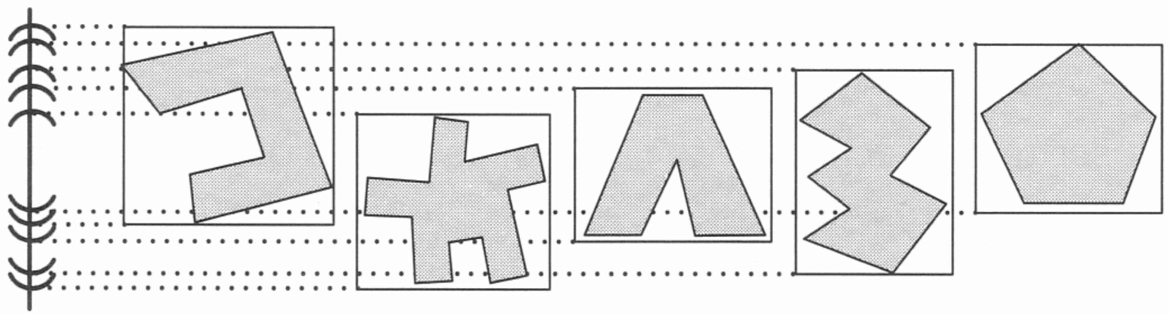


Figure 7.20 Objects clustered on the *y* axis (caused, for example, by falling objects settling on the ground). Even small object movements can now cause large positional changes in the list for the clustered axis.