Collision Detection

Problem Description

Object representations in simulation environments
do not consider impenetrability.

Collision detection: Detection of interpenetrating objects.

- polygonal or non-polygonal surface
- convex, non-convex
- defined volume (closed or open surface)
- rigid or deformable objects
- pair-wise tests or multiple objects
- first contact, all contacts
- intersection, proximity, penetration depth
- static or dynamic
- discrete or continuous time

Outline

Bounding Volumes

Bounding Volume Hierarchies BVH

Generation of BVHs

Comparison

BVHs for Deformable Objects

Bounding Volumes

Simplified conservative surface representation
for fast approximative collision detection test

- **Spheres**
- **Axis-aligned bounding boxes (ABB)**
- **Object-oriented bounding boxes (OBB)**
- **Discrete orientation polytopes (k-DOPs)**

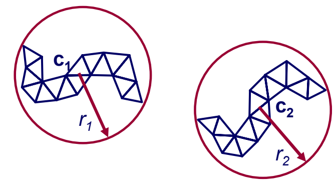
- avoid checking all object primitives.
- check bounding volumes to get the information whether objects **could** interfere. Fast rejection test.
- motivated by **spatial coherence**: Assumption that collisions between objects are rare

Requirements for Bounding Volumes

- should fit the object as tightly as possible to reduce the probability of a query object intersecting the volume but not the object
- overlap tests for bounding volumes should be efficient
- memory efficient
- efficient computation of a bounding volume, if recomputation is required

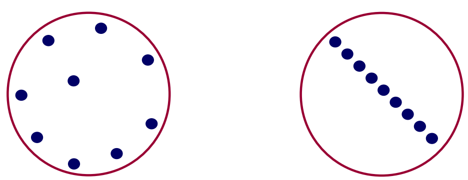
Spheres

sphere is represented by center **c** and radius *r*.



two spheres do not overlap if $(\mathbf{c}_1-\mathbf{c}_2)(\mathbf{c}_1-\mathbf{c}_2) > (r_1+r_2)^2$

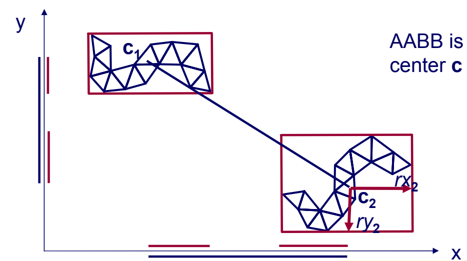
Sphere as Bounding Volume



good choice

bad choice

Axis-Aligned Bounding Box AABB

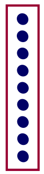


AABB is represented by center **c** and radii *rx*, *ry*.

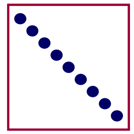
two AABBs do not overlap in 2D if $\left|(\mathbf{c}_1-\mathbf{c}_2)\begin{pmatrix}1\\0\end{pmatrix}\right| > rx_1+rx_2$

or $\left|(\mathbf{c}_1-\mathbf{c}_2)\begin{pmatrix}0\\1\end{pmatrix}\right| > ry_1+ry_2$

AABB as Bounding Volume



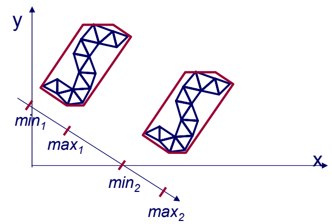
good choice



bad choice

Discrete Orientation Polytope k-DOP

A k-DOP is “a convex polytope whose facets are determined by halfspaces whose outward normals come from a small ***fixed*** set of *k* orientations.” [Klosowski]

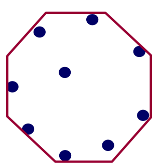


k-DOP is represented by k/2 directions and k/2 pairs of *min*, *max* values (6-, 14, 18-, 26-DOPs)

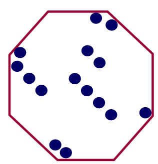
Two k-DOPs do not overlap, if their projections in at least one direction do not overlap.

8-DOPs as Bounding Volumes

larger *k*’s are more flexible than smaller
AABB is a 4-DOP. Is a 4-DOP an AABB?

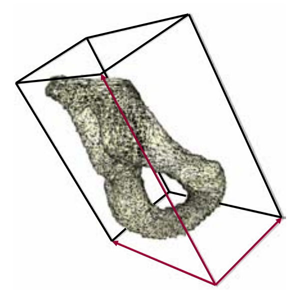


good choice



quite good choice

Oriented Bounding Box OBB



eigenvectors of the covariance matrix

An OBB can be represented by the principal axes of a set of vertices. These axes are ***not fixed***. They move according to object transformations.

vertices: $v \quad v \in \Re^3$

mean: $\mu = \frac{1}{n}\sum_{i=1}^{n} v_i$

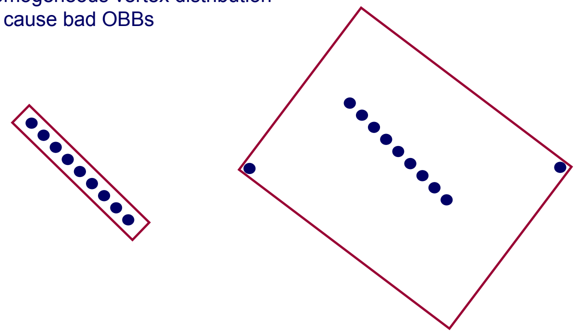
covariance matrix:

$$C_{jk} = \frac{1}{n}\sum_{i=1}^{n} \bar{v}_{ij}\bar{v}_{ik}$$

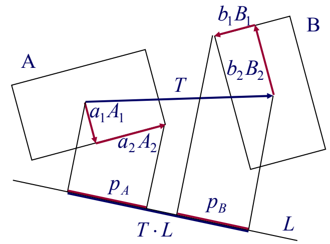
$$\bar{v}_i = v_i - \mu \qquad 1 \le j,k \le 3$$

OBB Examples

- principal axes of an object are not always a good choice for the main axes of an OBB
- inhomogeneous vertex distribution can cause bad OBBs



OBB Overlapping Test in 2D



A_1, A_2, B_1, B_2 • axes of $\mathrm{A, B}$ • unit vectors

a_1, a_2, b_1, b_2 • 'radii' of $\mathrm{A, B}$

L • unit vector

$$p_A = |a_1 A_1 L| + |a_2 A_2 L|$$

$$p_B = |b_1 B_1 L| + |b_2 B_2 L|$$

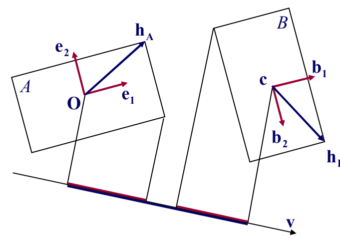
$\mathrm{A, B}$ do not overlap:

$\exists L : |T \cdot L| > p_A + p_B$ or $\exists L \in \{A_1, A_2, B_1, B_2\} : |T \cdot L| > p_A + p_B$

Separating Axis Test SAT

- works with polytopes: line segments, triangles, boxes
- two objects *A* and *B* are disjoint if for some vector **v** the projections of the objects onto the vector do not overlap. In this case, **v** is referred to as separating axis.
- vector **v** has to be a **face orientation** of *A* or *B* or a **cross product of two edges** of *A* and *B*.
- 3D boxes: tests with 3 + 3 + 3 · 3 axes

OBB Overlapping Test in 3D



- $\mathbf{B}=[\mathbf{b}_1\ \mathbf{b}_2\ \mathbf{b}_3]$ is orientation of *B* relative to *A*'s local basis **I**
- **c** is the center of *B* relative to *A*'s local coordinate system
- $\mathbf{h}_A$, $\mathbf{h}_B$ are the extents of *A*, *B*
- **v** is relative to *A*'s basis, $\mathbf{B}^T\mathbf{v}$ is the same vector relative to *B*

- vector **v** is a separating axis iff

$$|\mathbf{v} \cdot \mathbf{c}| > |\mathbf{v}| \cdot \mathbf{h}_A + |\mathbf{B}^T \mathbf{v}| \cdot \mathbf{h}_B$$

OBB Overlapping Test in 3D

$$|\mathbf{v}\cdot\mathbf{c}| > |\mathbf{v}|\cdot\mathbf{h}_A + |\mathbf{B}^T\mathbf{v}|\cdot\mathbf{h}_B$$

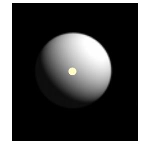
- 15 axes **v** have to be tested
  - 3 coordinate axes of *A*'s orientation **I**
  - 3 coordinate axes of *B*'s orientation **B** = [$\mathbf{b}_1$ $\mathbf{b}_2$ $\mathbf{b}_3$] = [β_{ij}]
  - 9 cross products of a coord. axis of **I** and a coord. axis of **B**
- expressions $\mathbf{B}^T\mathbf{v}$ can be simplified for all axes, e. g.

$$\mathbf{v} = \mathbf{e}_1 \times \mathbf{b}_2 = (0, -\beta_{32}, \beta_{22})^T$$
$$\mathbf{B}^T\mathbf{v} = \mathbf{B}^T(\mathbf{e}_1 \times \mathbf{b}_2) = (-\beta_{13}, 0, \beta_{11})^T$$

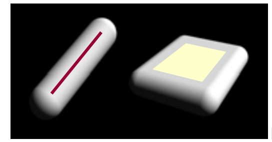
Bounding Volumes Summary

- spheres
- axis-aligned bounding boxes (AABB)
- oriented bounding boxes (OBB)
- discrete orientation polytopes (k-DOPs)

- ellipsoids
- convex Hulls
- swept-Sphere Volumes (SSVs)
  - point Swept Spheres (PSS)
  - line Swept Spheres (LSS)
  - rectangle Swept Spheres (RSS)
  - triangle Swept Spheres (TSS)



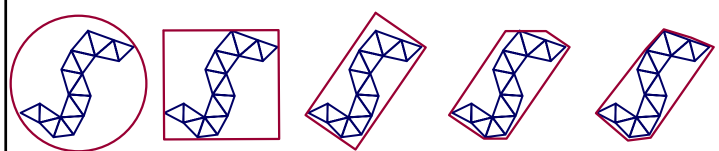
PSS



Lin, UNC

LSS RSS

Optimal Bounding Volume



sphere ABB OBB 6-DOP convex hull

tighter approximation →

← *decreasing complexity and computational expenses for overlap test*

Outline

Bounding Volumes

Bounding Volume Hierarchies BVH

Generation of BVHs

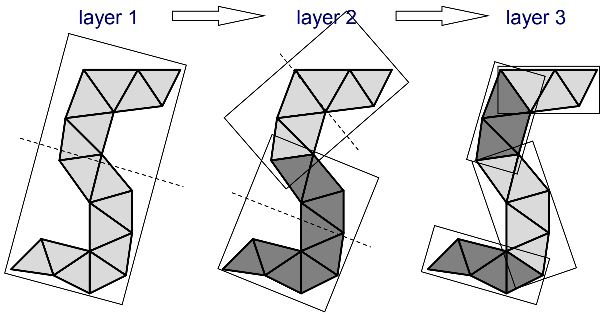
Comparison

BVHs for Deformable Objects

Bounding Volume Hierarchies BVHs

• subdivsion of bounding volumes to generate a hierarchy

• improved object approximation at higher levels

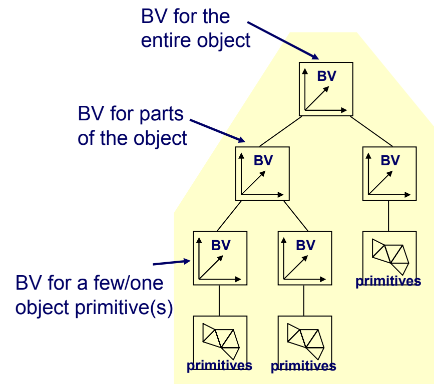


Hierarchy of Bounding Volumes

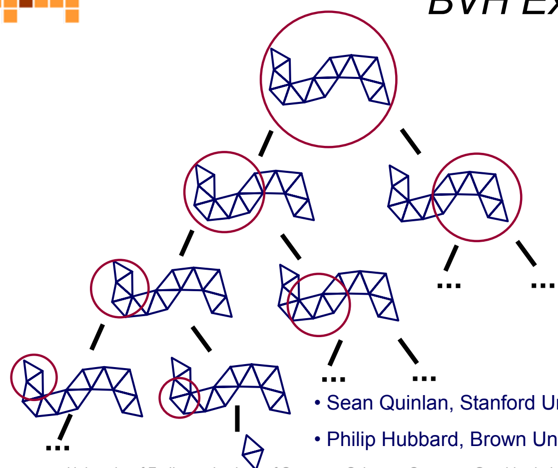
• bounding volume tree (BV tree)

• nodes contain bounding volume information

• leaves additionally contain information on object primitives



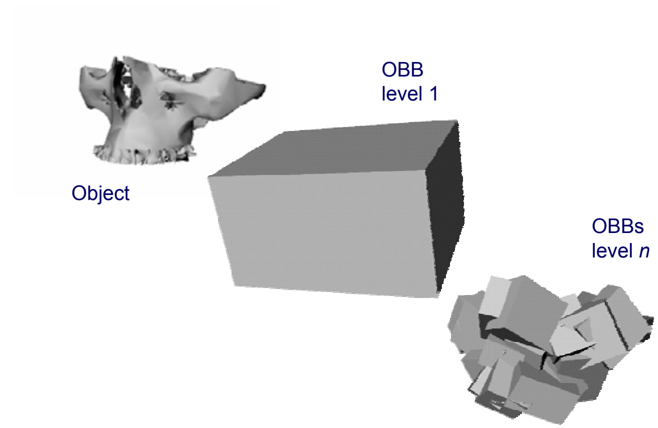
BVH Example



• Sean Quinlan, Stanford Univ

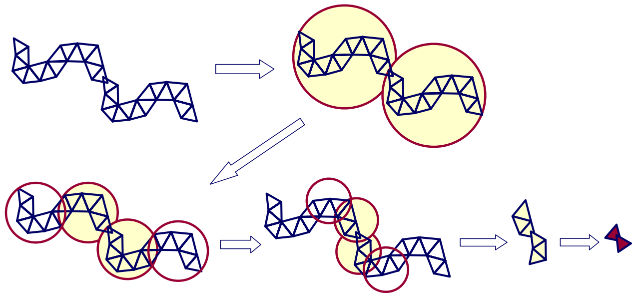
• Philip Hubbard, Brown Univ

OBB Tree



Overlapping Test for BV Tree

- BV-trees speed-up the collision detection test
- if bounding volumes in a hierarchy level overlap, their children are checked for overlapping. If leaves are reached, primitives are checked against each other.



Overlapping Test for BV Tree

Pseudo code

1. interference check for two parent nodes (root)
2. if no interference then “no collision” else
3. all children of one parent node are checked against children of the other parent node
4. if no interference then “no collision” else
5. if at leave nodes then “collision” else go to 3

step 3 checks BVs or object primitives for intersection

Box-Triangle and Triangle-Triangle Test

Box-Triangle Test

a) separating axes test requires 13 axes to be tested (4 face normals, 3 x 3 cross products of edges)

Triangle-Triangle Test

a) separating axes test requires max. 11 axes to be tested (2 face normals, 3 x 3 cross products of edges)

b) testing each edge of one triangle against the other triangle for intersection -> 6 edge-triangle tests

(edge-triangle intersections occur in pairs -> 5 tests are sufficient)

Edge-Triangle Test

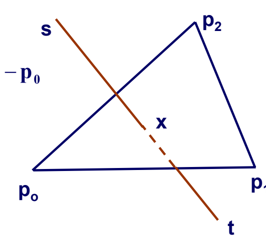
$$\mathbf{x} = \mathbf{p_0} + \mu_1(\mathbf{p_1} - \mathbf{p_0}) + \mu_2(\mathbf{p_2} - \mathbf{p_0}) \quad \mu_1, \mu_2 \geq 0 \quad \mu_1 + \mu_2 \leq 1$$

$$\mathbf{x} = \mathbf{s} + \lambda(\mathbf{t} - \mathbf{s}) \quad 0 \leq \lambda \leq 1$$

$$\mathbf{r} = \mathbf{t} - \mathbf{s} \quad \mathbf{d}_1 = \mathbf{p}_1 - \mathbf{p}_0 \quad \mathbf{d}_2 = \mathbf{p}_2 - \mathbf{p}_0 \quad \mathbf{b} = \mathbf{s} - \mathbf{p}_0$$

$$\mathbf{b} = \mu_1\mathbf{d}_1 + \mu_2\mathbf{d}_2 - \lambda\mathbf{r}$$

$$\begin{pmatrix} \lambda \\ \mu_1 \\ \mu_2 \end{pmatrix} = \frac{1}{-\mathbf{r}\cdot(\mathbf{d}_1 \times \mathbf{d}_2)} \begin{pmatrix} \mathbf{b}\cdot(\mathbf{d}_1 \times \mathbf{d}_2) \\ \mathbf{d}_2\cdot(\mathbf{b} \times \mathbf{r}) \\ -\mathbf{d}_1\cdot(\mathbf{b} \times \mathbf{r}) \end{pmatrix}$$



edge intersects iff

$$-\mathbf{r}\cdot(\mathbf{d}_1 \times \mathbf{d}_2) \neq 0 \quad 0 \leq \lambda \leq 1 \quad \mu_1 + \mu_2 \leq 1 \quad \mu_1, \mu_2 \geq 0$$

Characteristics of BVH

- improved object approximation at higher levels
- fast rejection query
- fast localization of object regions with potential collisions

- additional storage requirements
- generation of BVHs can be expensive
  - BVHs are generally used for rigid models where they can be pre-computed

Computational Costs of BV Trees

Cost function (M. Lin, UNC):

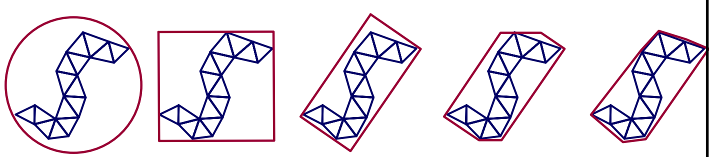
$$F = \underbrace{N_u \times C_u}_{\text{tree generation/update}} + \underbrace{N_{bv} \times C_{bv}}_{\text{BV intersection test}} + \underbrace{N_p \times C_p}_{\text{primitive intersection test}}$$

F:	total cost for interference detection
N_u:	number of bounding volumes updated
C_u:	cost of updating a bounding volume
N_{bv}:	number of bounding volume pair overlap tests
C_{bv}:	cost of overlap test between two bounding volumes
N_p:	number of primitive pairs tested for interference
C_p:	cost of testing two primitives for interference

Optimization

$$F = N_u \times C_u + N_{bv} \times C_{bv} + N_p \times C_p$$

- infrequent BV updates to minimize N_u
- tight-fitting bounding volumes to minimize N_{bv}
- simple intersection test for bounding volumes to minimize C_{bv}

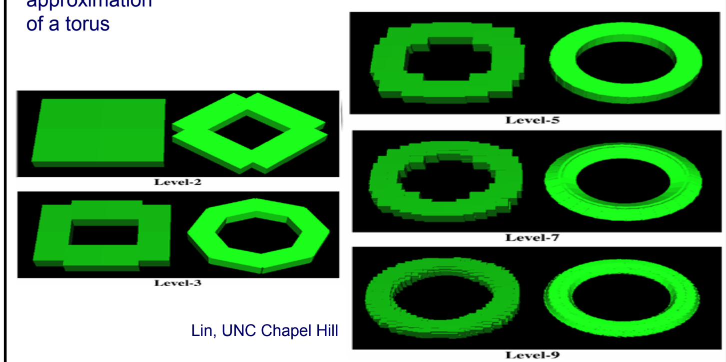


Better approximation →

← *Decreasing computational expenses for overlap test*

AABB vs. OBB Tree

approximation of a torus



Lin, UNC Chapel Hill

Object Transformations

some object transformations can be simply applied
to all elements of the bounding-volume tree:

Spheres

- translation, rotation

Axis-Aligned Bounding Boxes

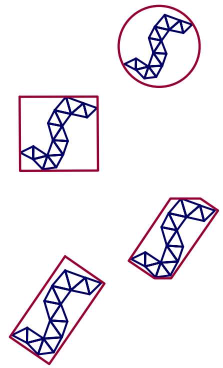
- translation, no rotation

Discrete Orientation Polytopes

- translation, no rotation
  (principal orientations are fixed for all objects)

Object-Oriented Bounding Boxes

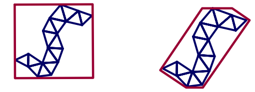
- translation, rotation
  (box orientations are not fixed)



Rotations

Axis-Aligned Bounding Boxes

Discrete Orientation Polytopes



- rotation of the bounding volume is not
  possible due to the respective box overlap test.
  The intersection tests require fixed surface normals.

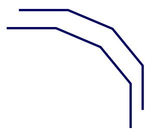
1. recomputation of the BV hierarchy
2. preservation of the tree structure, update of all nodes

a) additional storage of the convex hull which is rotated with the object
  - check if extremal vertices are still extremal after rotation
  - compare with adjacent vertices of the convex hull
  - “climb the hill” to the extremal vertex

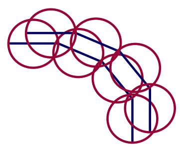
b) computation of an approximate box by rotating the box and
  checking the rotated box for extremal values

Close Proximity

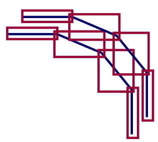
- quality of ***higher***-level
  BV approximation
  influences collision
  detection performance
  in case of close proximity
- quality of higher-level
  BV approximations is
  not very critical
- in case of overlapping BV
  expensive primitive tests
  have to be performed



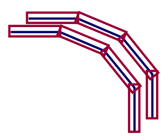
Problem



Spheres



ABB, k-DOP



OBB

Outline

Construction of a BV Tree

Bottom-Up

- start with object-representing primitives
- fit a bounding volume to each primitive
- group primitives or bounding volumes recursively
- fit bounding volumes to these groups
- stop in case of a single bounding volume at a hierarchy level

Top-Down

- start with object
- fit a bounding volume to the object
- split object or bounding volume recursively
- fit bounding volumes
- stop, if all bounding volumes in a level contain less than *n* primitives

Construction of a BV Tree

Parameters

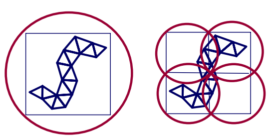
- bounding volume
- top-down vs. bottom-up
- what to subdivide / group: object primitives or bounding volumes
- how to subdivide / group object primitives or bounding volumes
- how many primitives in each leaf of the BV tree
- re-sampling of the object ?

Goals

- balanced tree
- tight-fitting bounding volumes
- minimal redundancy
  (primitives in more than one BV per level)

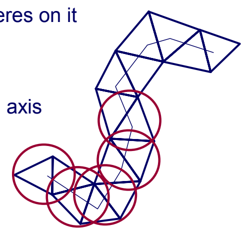
Construction of a BV Tree Spheres

Hubbard, C. O'Sullivan:
- approximate triangles with spheres and build the tree bottom-up by grouping spheres
- cover vertices with spheres and group them
- resample vertices prior to building the tree (homogeneous vertex distribution reduces redundancy)
- build the tree top-down by using an octree
- compute the medial axis and place spheres on it



octree based

medial axis based



Outline

Bounding Volumes

Bounding Volume Hierarchies BVH

Generation of BVHs

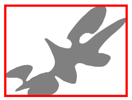
Comparison

BVHs for Deformable Objects

Collision Detection Libraries

SOLID

Axis-aligned bounding box



van den Bergen
Eindhoven University
1997

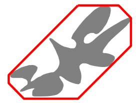
RAPID

Object-oriented bounding box

Gottschalk et al.
University of North Carolina
1995

QuickCD

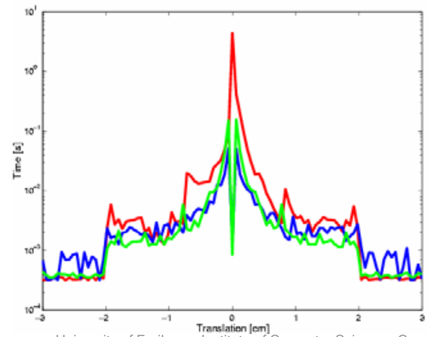
k discrete orientation polytope

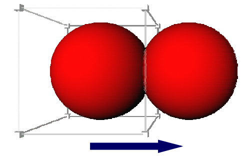


Klosowski et al.
University of New York
1998

Comparison of CD Libraries

- time to compute a collision for two spheres with radius 1 cm
- translation represents the distance of both centers
- QuickCD [Klosowski], RAPID [Gottschalk], SOLID [Bergen]





10,000 triangles per sphere

8-DOP
OBB
ABB

Outline

Bounding Volumes

Bounding Volume Hierarchies BVH

Generation of BVHs

Comparison

BVHs for Deformable Objects

BVHs for Deformable Collision Detection

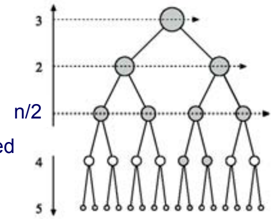
- in case of deformable objects, BVH has to be updated frequently
- hierarchy generation significantly influences performance
- AABBs are commonly used
- AABBs can be updated efficiently compared to OBB, k-DOP, spheres
- however, AABBs do not provide an optimal model approximation

Hybrid Hierarchy Update

- proposed by Larsson / Akenine-Moeller, Eurographics 2001
- AABB hierarchy
- initial hierarchy generation as pre-processing

- lazy hierarchy update during run-time
  - bottom-up update starting at depth n/2
  - very efficient AABB update based on AABBs of children

- update of nodes in depth n/2+1 to n as needed
- this update is only performed if necessary

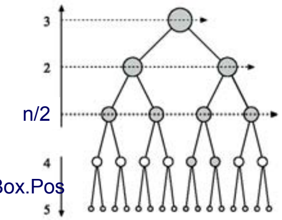


Implementation of Hierarchy Update

- after pre-processing each nodes knows which vertices influence its bounding box

- object is traversed once to update nodes (box information) in layer n/2

- bottom-up merging of AABBs
  - Merge (b1, b2)
    Box.Pos = Min(b1.Pos, b2.Pos)
    Box.Size = Max(b1.Pos+b1.Size, b2.Pos+b2.Size)-Box.Pos



Hierarchical Bounding Volumes - Summary

- bounding volume tree (BV tree) based on spheres or boxes
- nodes contain bounding volume information
- leaves additionally contain information on object primitives

- isolating interesting regions by checking bounding volumes in a top-down strategy

- construction of a balanced, tight-fitting tree with minimal redundancy

- transformation of BV trees dependent on the basic bounding volume

- optimal bounding box hierarchy dependent on application (e. g. close proximity problem)

References

- **S. Quinlan**, "Efficient Distance Computation Between Non-Convex Objects," *Proc. Int. Conf. on Robotics and Automation*, pp. 3324-3329, 1994.
- **P. M. Hubbard**, "Approximating Polyhedra With Spheres for Time-Critical Collision Detection," *ACM ToG*, 15 / 3, pp. 179-210, 1996.
- **S. Gottschalk**, M. C. Lin, D. Manocha, "OBBTree: A Hierarchical Structure for Rapid Interference Detection," *Proc. SIGGRAPH'96, ACM Computer Graphics*, New York, NY, USA, pp. 171-180, 1996.
- **G. van den Bergen**, "Efficient Collision Detection of Complex Deformable Models using AABB Trees," *Journal of Graphics Tools*, 2 / 4, pp. 1-13, 1997.
- **J. T. Klosowski** et al., "Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs," *IEEE Trans on Vis and Computer Graphics*, 4 / 1, pp. 21-36, 1998.
- **Larsson, Akenine-Moeller**, "Collision Detection for Continuously Deforming Objects," *Proc. Eurographics,* 2001.
- **G. van den Bergen**, "*Collision Detection in Interactive 3D Environments*," Elsevier, Amsterdam, ISBN: 1-55860-801-X, 2004.