Simulation in Computer Graphics
Bounding Volume Hierarchies

Matthias Teschner

UNI FREIBURG

Outline

- Introduction
- Bounding volumes BV
- Hierarchies of bounding volumes BVH
- Generation and update of BVs
- Design issues of BVHs
- Performance
- BVHs for deformable objects

UNI FREIBURG

Motivation

- Detection of interpenetrating objects
- Object representations in simulation environments do not consider impenetrability
- Aspects
  - Polygonal, non-polygonal surface
  - Convex, non-convex
  - Defined volume (closed surface), undefined volume
  - Rigid, deformable

UNI FREIBURG

Motivation

- More aspects
  - Pair-wise tests, multiple objects
  - First contact, all contacts
  - Intersection, penetration depth, proximity
  - Static, dynamic
  - Discrete simulation time, continuous time

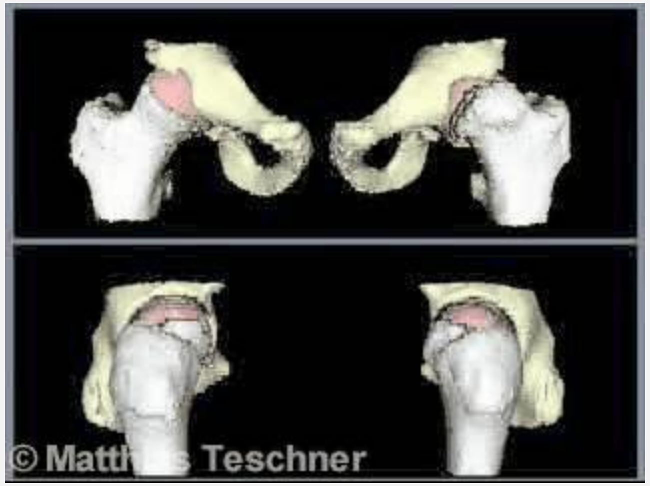
Example

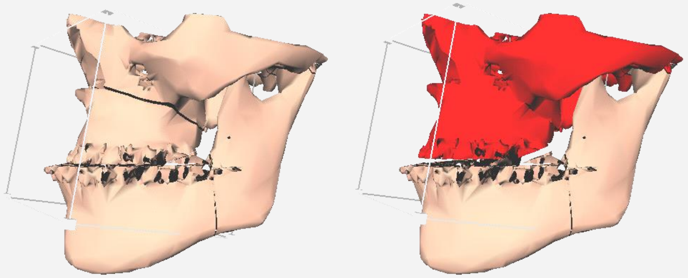
- Collision detection is an essential part of physically realistic dynamic simulations
- In each time step
  - Detect collisions
  - Resolve collisions
  - Compute dynamics

[UNC, Univ of Iowa]

Applications

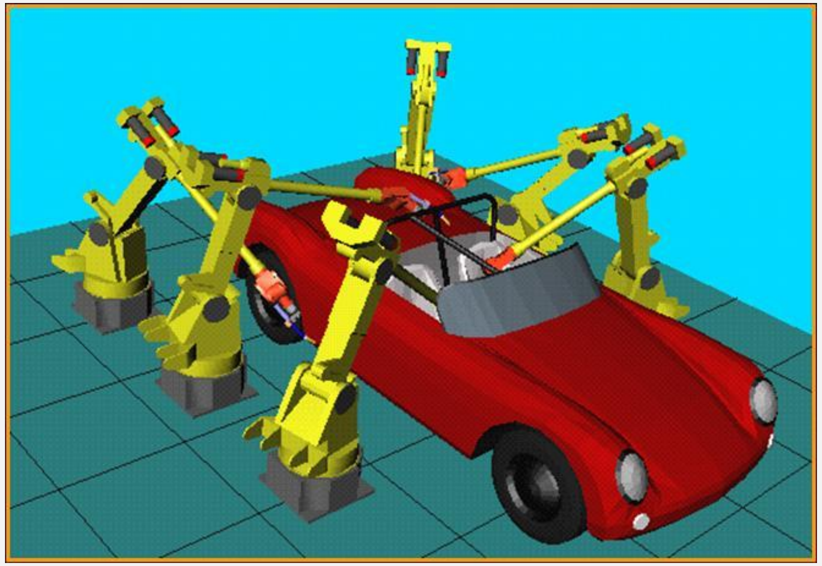
- Surgery planning
  - Dysfunction of the hip joint
    due to reoriented femoral head
  - Simulation of range of motion using triangulated surface representations
  - Craniofacial surgery planning
  - Planning of osteotomies and bone realignment





Applications

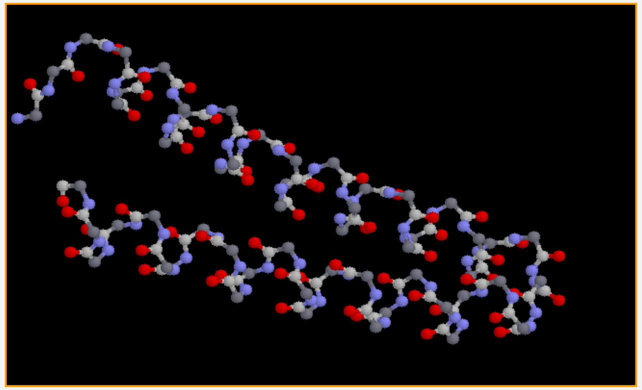
- Robotics
  - Task planning for multiple robots in a virtual environment
  - Continuous time to avoid collisions with small structures
  - Time-critical in case of a break-down
  - Investigated in the context of product assembly since 1979

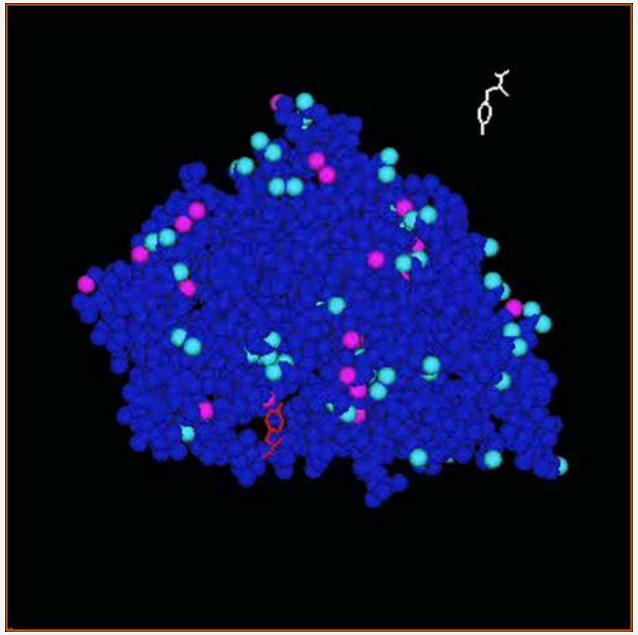


[Stanford]

Applications

- Protein folding
  - Design of drug molecules
  - Computation of low-energy states considering geometric constraints





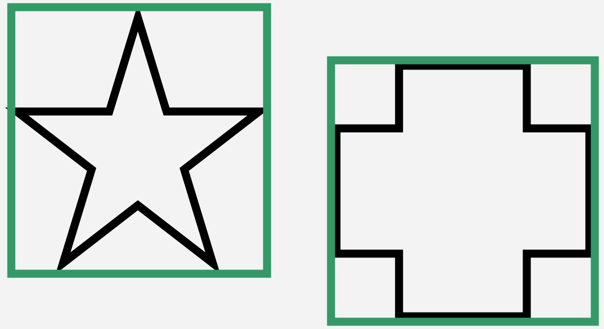
[Stanford]

Outline

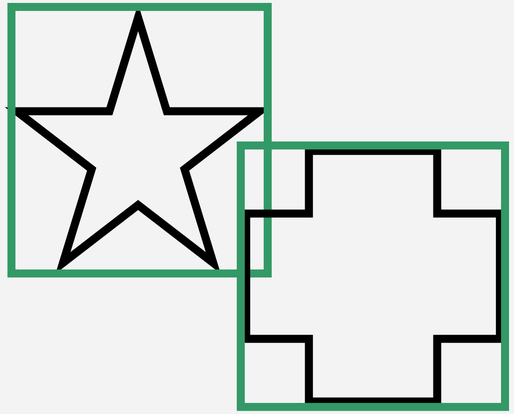
- Introduction
- Bounding volumes BV
- Hierarchies of bounding volumes BVH
- Generation and update of BVs
- Design issues of BVHs
- Performance
- BVHs for deformable objects

Motivation

- Collision detection for polygonal models is in $O(n^2)$
- Simple bounding volumes – encapsulating geometrically complex objects – can accelerate the detection of collisions



No overlapping bounding volumes
→ No collision

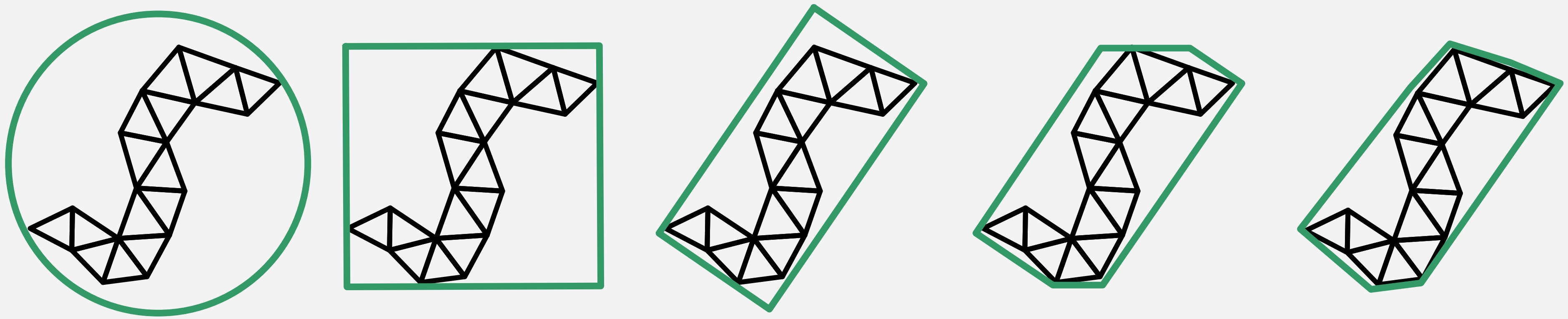


Overlapping bounding volumes
→ Objects **could** interfere

Motivation

- Efficient detection of collision-free state without checking all object primitives
- Overhead for overlapping bounding volumes
  - However, only few bounding volumes overlap in case of multiple objects motivated by spatial coherence
- For some applications, collision information on the bounding volumes might be sufficient
  - Approximate collision detection
  - Accuracy depends on the tight fitting of the bounding volume

Examples and Characteristics



Sphere

Axis-aligned bounding box

Oriented bounding box

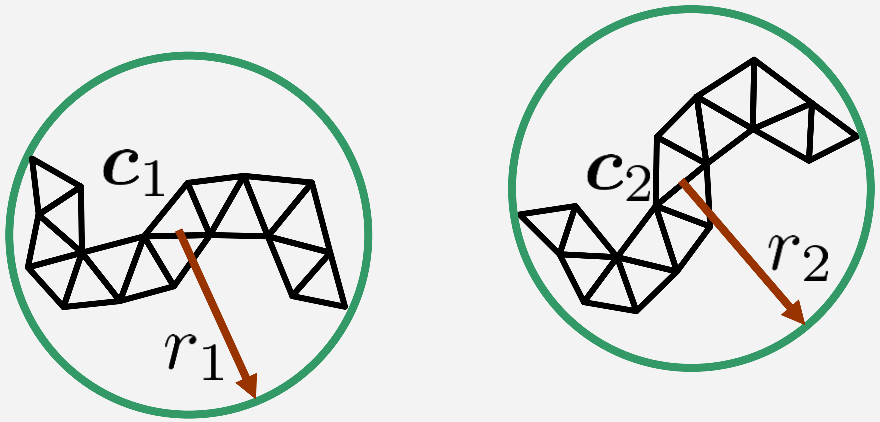
Discrete-orientation polytope

Convex hull

- Desired characteristics
  - Efficient intersection test, memory efficient
  - Efficient generation and update in case of transformations
  - Tight fitting

Sphere

- Spheres are represented by
  - The center position $\boldsymbol{c}$
  - The radius r
- Two spheres do not overlap if $(\boldsymbol{c}_1 - \boldsymbol{c}_2)(\boldsymbol{c}_1 - \boldsymbol{c}_2) > (r_1 + r_2)^2$



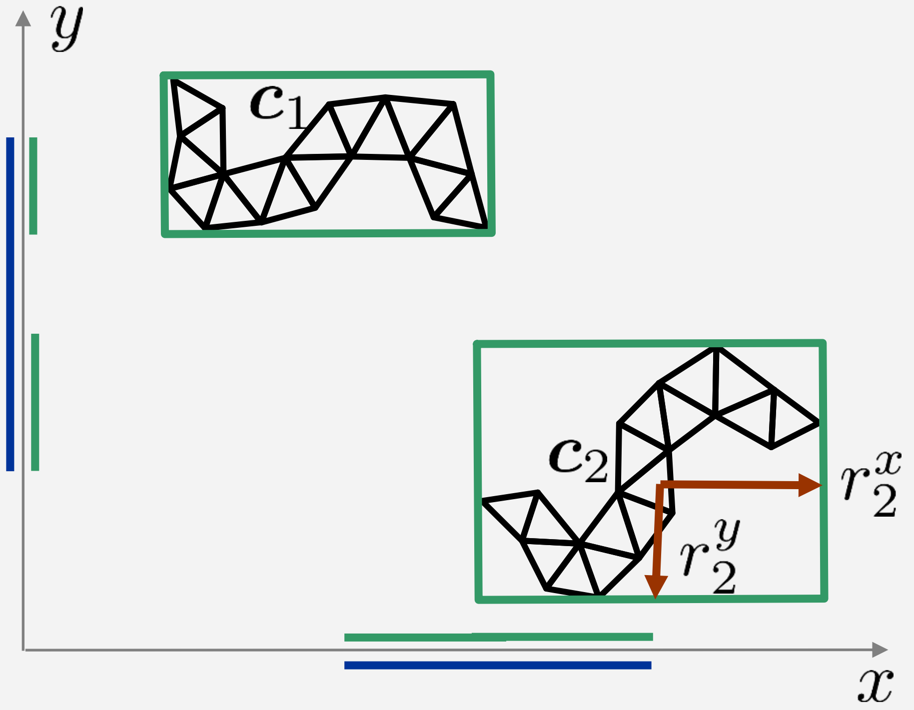
UNI FREIBURG

Axis-Aligned Bounding Box AABB

- AABBs are represented by
  - The center positions $\boldsymbol{c}$
  - The radii r^x, r^y
- Two AABBs in 2D do not overlap if

$$\left| (\boldsymbol{c}_1 - \boldsymbol{c}_2) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right| > r_1^x + r_2^x$$

$$\left| (\boldsymbol{c}_1 - \boldsymbol{c}_2) \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right| > r_1^y + r_2^y$$

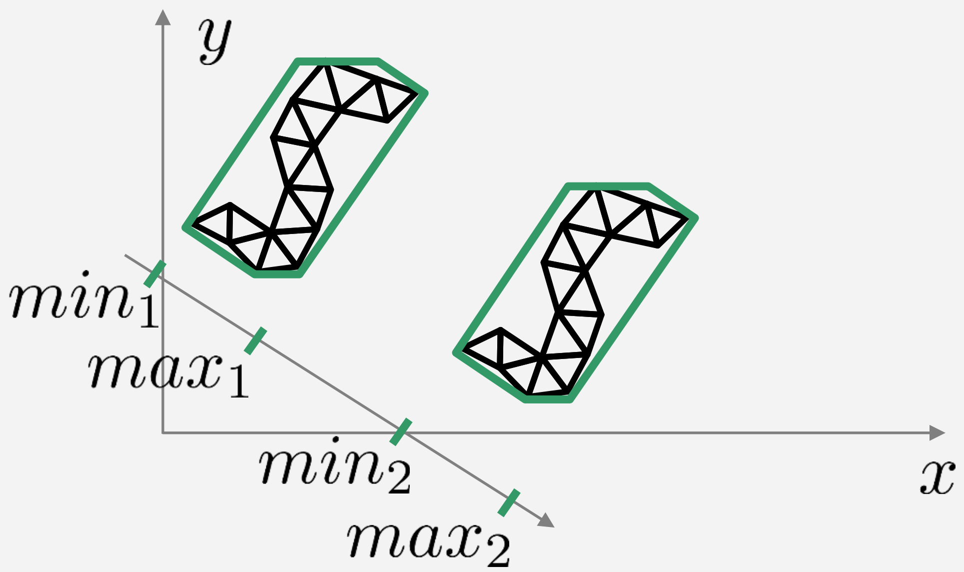


UNI FREIBURG

Discrete-Orientation Polytope k-DOP

- Convex polytope whose faces are determined by a fixed set of normals
- *k*-DOPs are represented by
  - *k* / 2 normals
  - *k* / 2 min-max intervals
- If any pair of intervals does not overlap, *k*-DOPs do not overlap

  $\exists \text{direction} : max_1 < min_2 \lor min_1 < max_2$



UNI FREIBURG

Discrete-Orientation Polytope k-DOP

- Also known as fixed-direction hulls FDHs
- AABB is a 4-DOP. Are all 4-DOPs AABBs?
- All k-DOPs share the same pre-defined normal set
- Only min-max intervals are stored per *k*-DOP
- Larger *k* improves the approximation quality
- Intersection test is more expensive for larger *k*

UNI FREIBURG

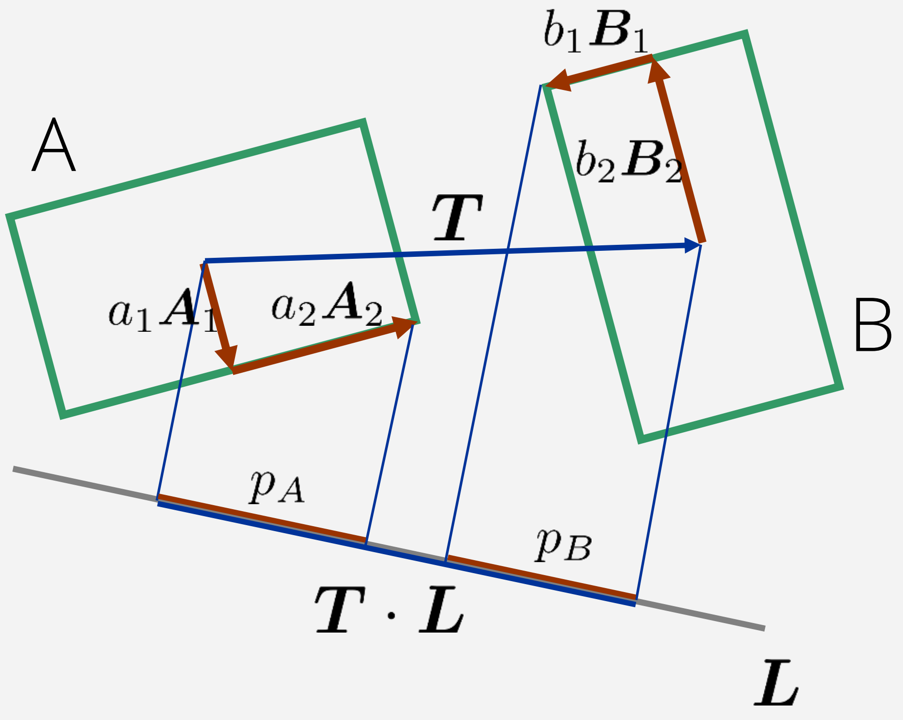
Oriented Bounding Box OBB

- Similar to AABB, but with flexible orientations
- OBBs have not to be aligned with respect to each other or to a coordinate system
- In contrast to AABBs and *k*-DOPs,
  - OBBs can be rotated with an object
  - OBBs are more expensive to check for overlap

UNI FREIBURG

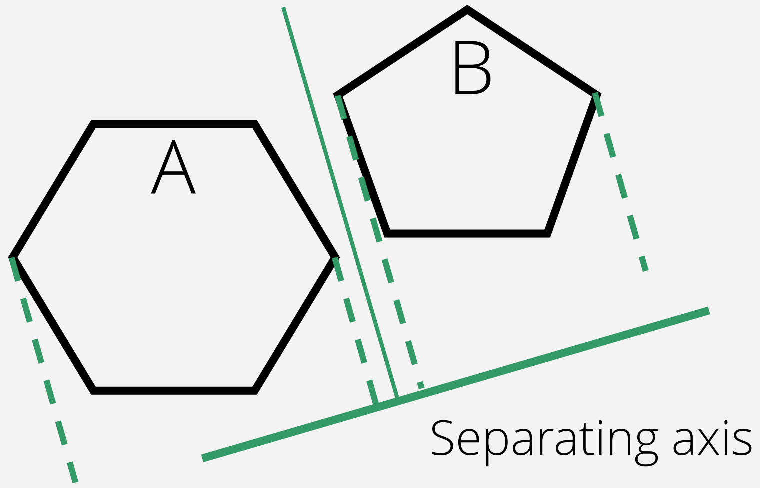
OBB Overlap Test in 2D

- $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{B}_1, \boldsymbol{B}_2$ are normalized axes of A and B
- a_1, a_2, b_1, b_2 are radii of A and B
- $\boldsymbol{L}$ is a normalized direction
- $\boldsymbol{T}$ is the distance of centers of A and B
- $p_A = a_1 \boldsymbol{A}_1 \boldsymbol{L} + a_2 \boldsymbol{A}_2 \boldsymbol{L}$
- $p_B = b_1 \boldsymbol{B}_1 \boldsymbol{L} + b_2 \boldsymbol{B}_2 \boldsymbol{L}$
- A and B do not overlap in 2D if $\exists \boldsymbol{L} : \boldsymbol{T} \cdot \boldsymbol{L} > p_A + p_B$



Separating Axis Test

- Motivation
  - Two objects A and B are disjoint if for some vector $\boldsymbol{v}$ the projections of the objects onto $\boldsymbol{v}$ do not overlap. In this case, $\boldsymbol{v}$ is a separating axis.
  - If A and B are convex, the separating axis exists if and only if A and B do not overlap.



For concave objects, a separating axis does not necessarily exist, if both objects are disjoint.

UNI FREIBURG

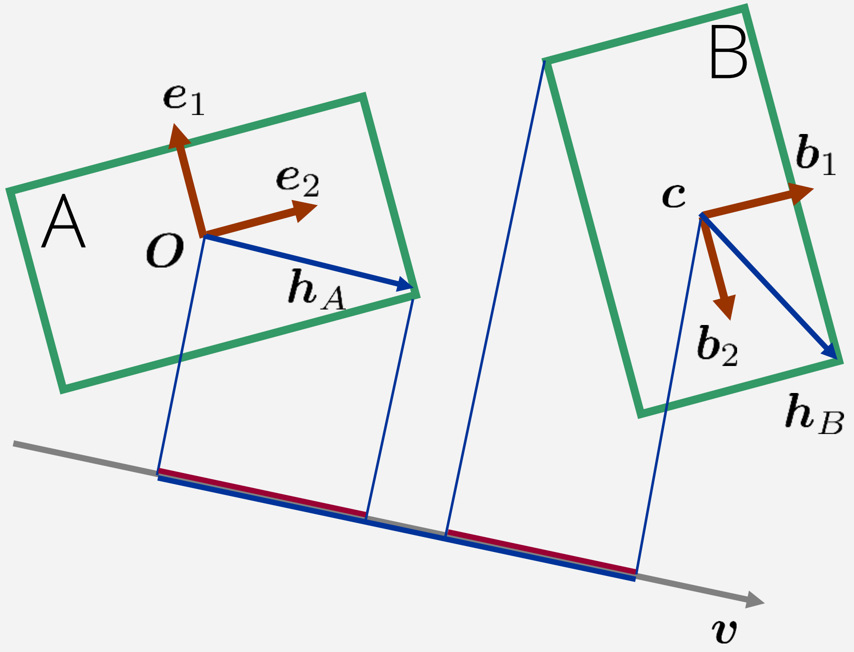
Separating Axis Test

- For polyhedral objects, only a few axes have to be tested
  - Axes parallel to face normals of A
  - Axes parallel to face normals of B
  - Axes parallel to all cross products of edges of A and B
- In case of 3D OBBs, $3 + 3 + 3 \cdot 3$ axes have to be tested
- Efficient and general overlap test
- Does not provide information on the intersection geometry

OBB Overlapping Test in 3D

- $\boldsymbol{B} = (\boldsymbol{b}_1\ \boldsymbol{b}_2\ \boldsymbol{b}_3)$ orientation of B relative to A's basis $\boldsymbol{I}$
- $\boldsymbol{c}$ is the center of B relative to A's coordinate system
- $\boldsymbol{h}_A, \boldsymbol{h}_B$ are the extents of A, B
- $\boldsymbol{v}$ is relative to A's basis, $\boldsymbol{B}^{\mathrm{T}}\boldsymbol{v}$ is the same vector relative to B
- $\boldsymbol{v}$ is a separating axis if

$$\boldsymbol{v} \cdot \boldsymbol{c} > \boldsymbol{v} \cdot \boldsymbol{h}_A + (\boldsymbol{B}^{\mathrm{T}}\boldsymbol{v}) \cdot \boldsymbol{h}_B$$



OBB Overlapping Test in 3D

- $\boldsymbol{v} \cdot \boldsymbol{c} > \boldsymbol{v} \cdot \boldsymbol{h}_A + (\boldsymbol{B}^{\mathrm{T}}\boldsymbol{v}) \cdot \boldsymbol{h}_B$
- 15 axes $\boldsymbol{v}$ have to be tested
  - 3 coordinate axes of A's orientation $\boldsymbol{I}$
  - 3 coordinate axes of B's orientation $\boldsymbol{B} = (\boldsymbol{b}_1 \ \boldsymbol{b}_2 \ \boldsymbol{b}_3) = (\beta_{ij})$
  - 9 cross products of coordinate axes of both orientations
- Computations of $\boldsymbol{B}^{\mathrm{T}}\boldsymbol{v}$ can be simplified, e.g.

$$\boldsymbol{v} = (\boldsymbol{e}_1 \times \boldsymbol{b}_2) = (0, -\beta_{32}, -\beta_{22})^{\mathrm{T}}$$

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

UNI FREIBURG

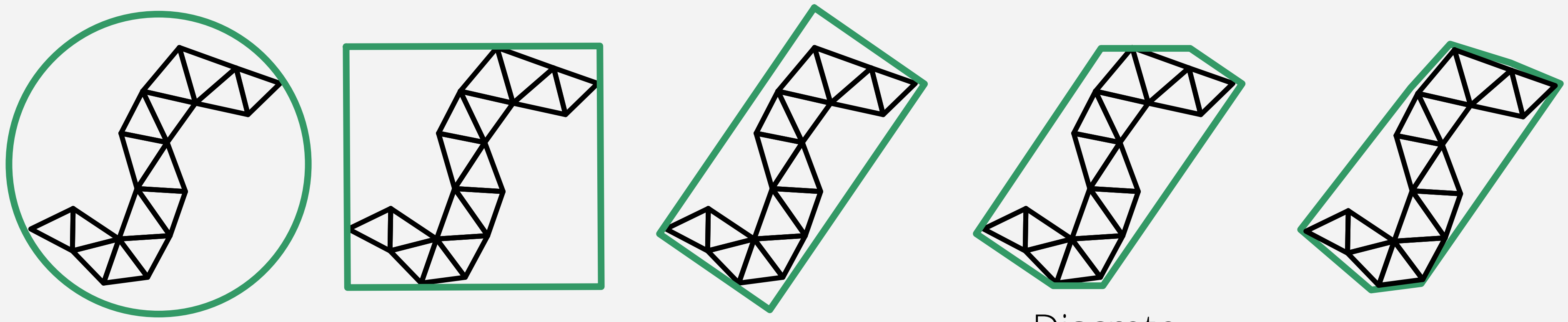
Optimal Bounding Volume

– It depends …

Tight approximation

⇄

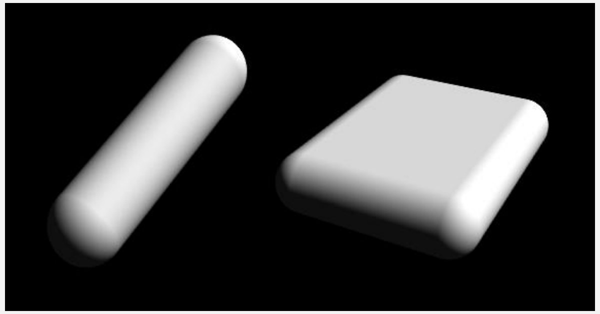
Efficient overlap test



Sphere | Axis-aligned bounding box | Oriented bounding box | Discrete-orientation polytope | Convex hull

Summary - Bounding Volumes

- Simple geometries that encapsulate complex objects
- Efficient overlap rejection test
- Tight object approximation,
  memory efficient, fast overlap test
- Spheres, AABBs, OBBs, k-DOPs, convex hulls
- Further BVs, e.g. swept sphere volumes SSVs
  - Point-swept sphere PSS
  - Line-swept sphere LSS
  - Rectangle-swept sphere RSS


[UNC]

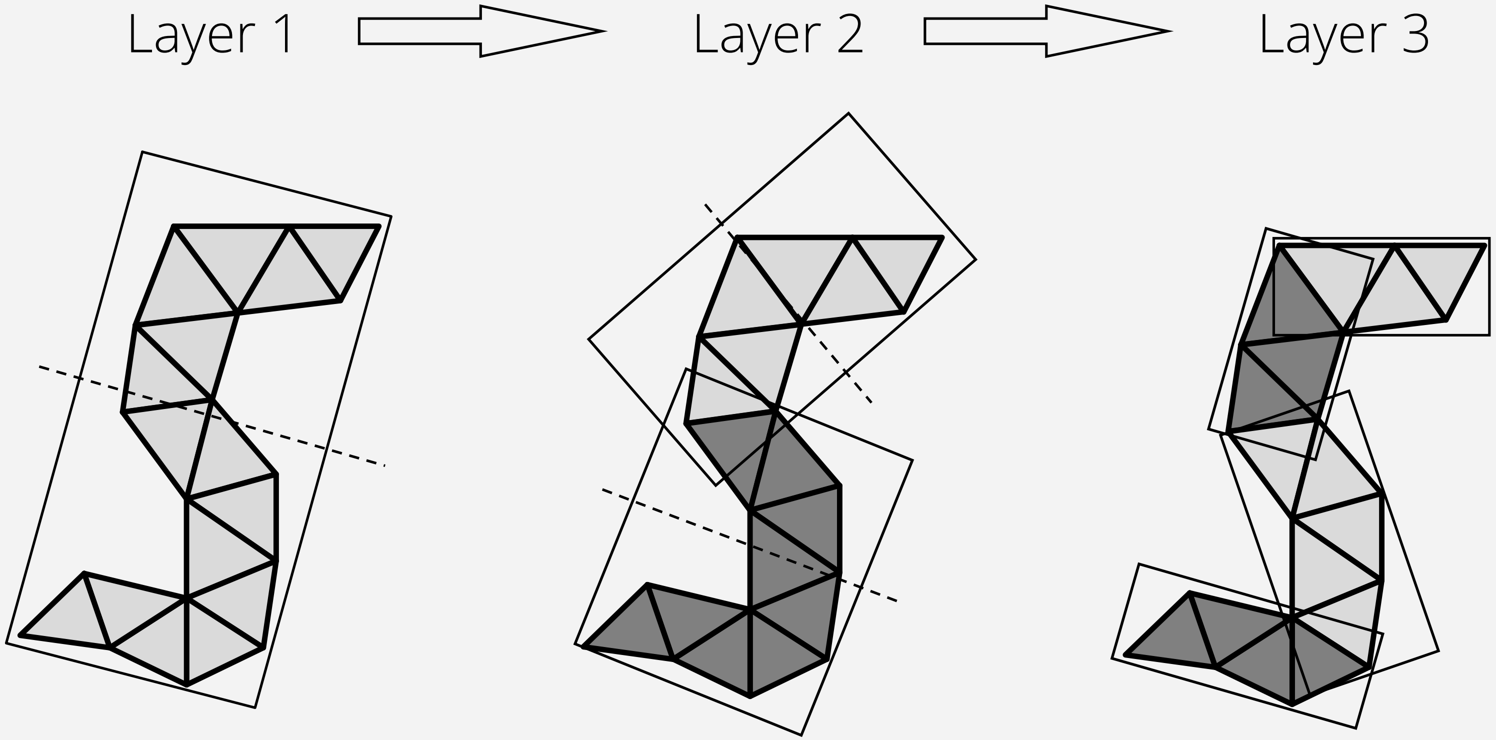
LSS RSS

Outline

- Introduction
- Bounding volumes BV
- Hierarchies of bounding volumes BVH
- Generation and update of BVs
- Design issues of BVHs
- Performance
- BVHs for deformable objects

Bounding Volume Hierarchies

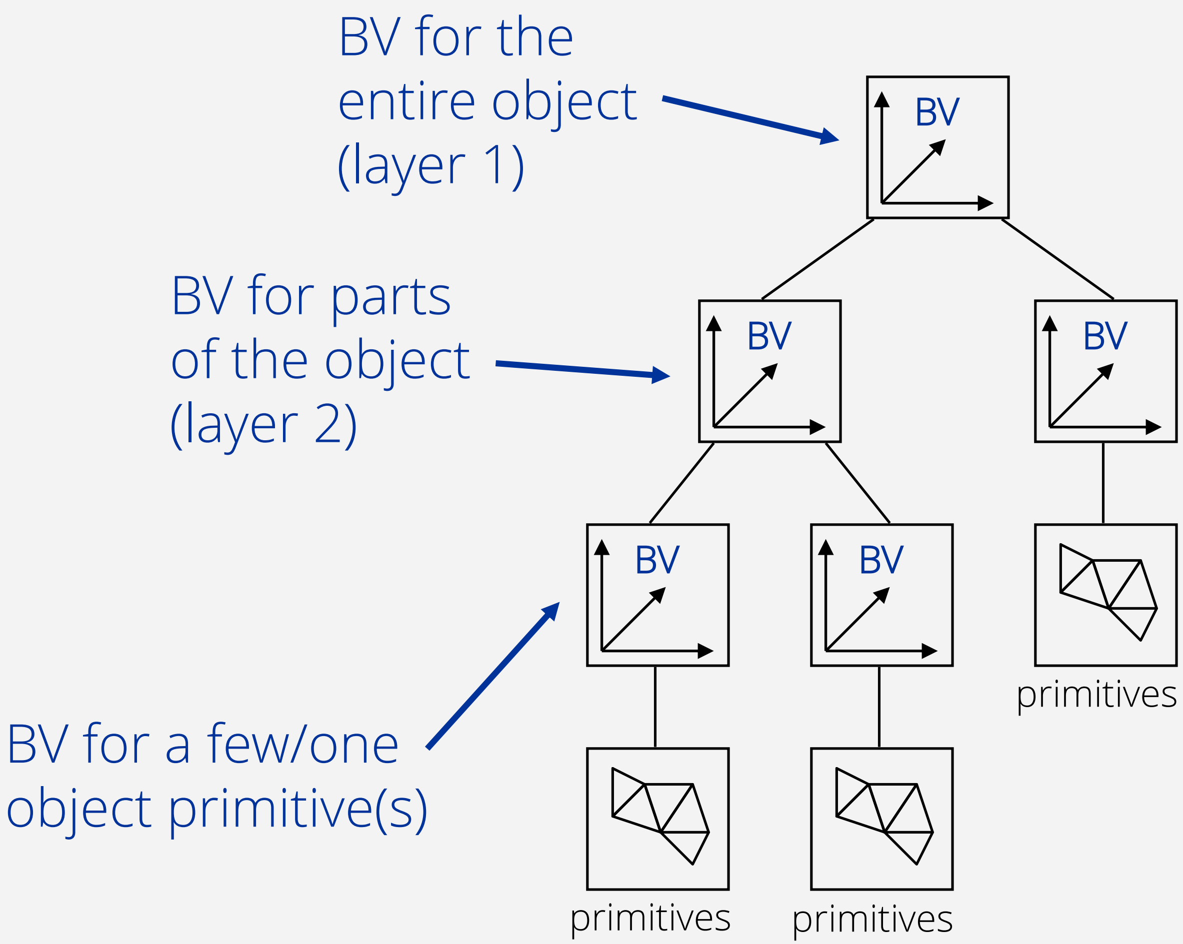
- Efficient overlap rejection test for parts of an objects
- E.g., object can be subdivided to build a BVH



UNI FREIBURG

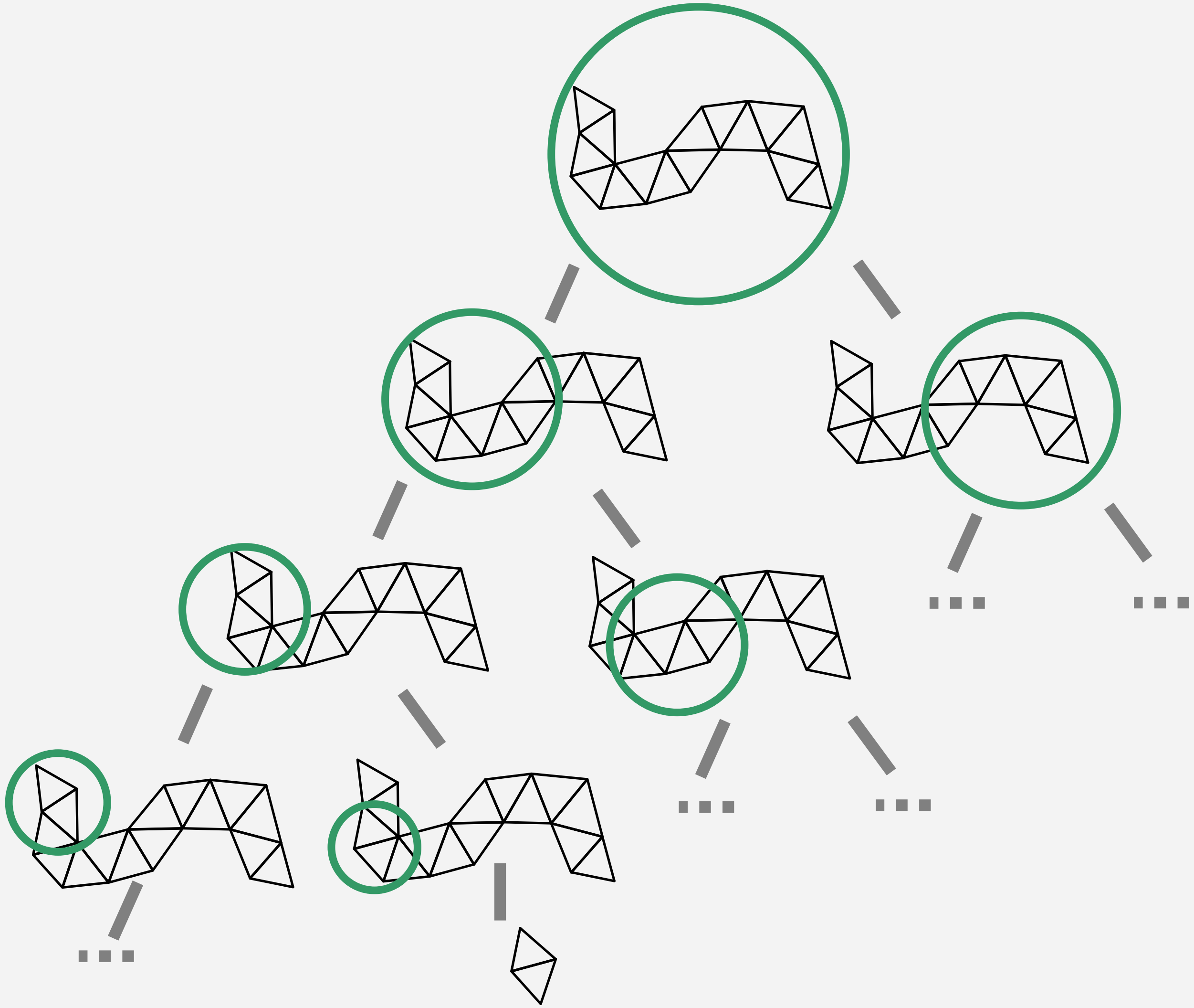
Data Structure

- Tree of bounding volumes
- Nodes contain BV information
- Leaf nodes contain object primitives



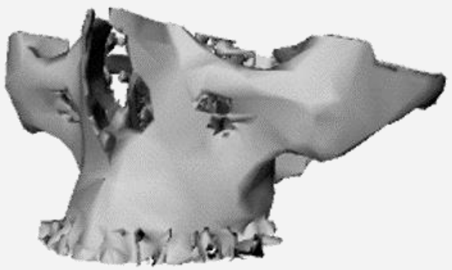
UNI FREIBURG

Example – Sphere Tree

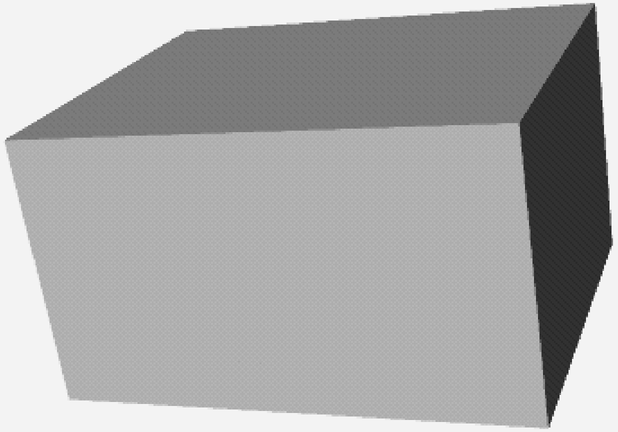


UNI FREIBURG

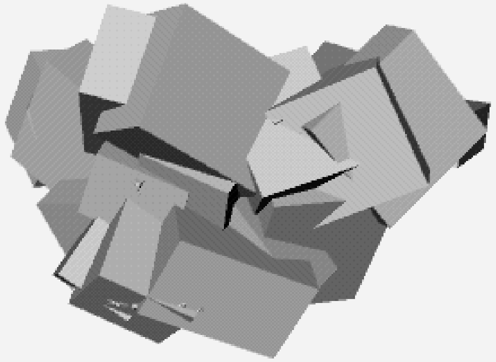
Example – OBB Tree



Object

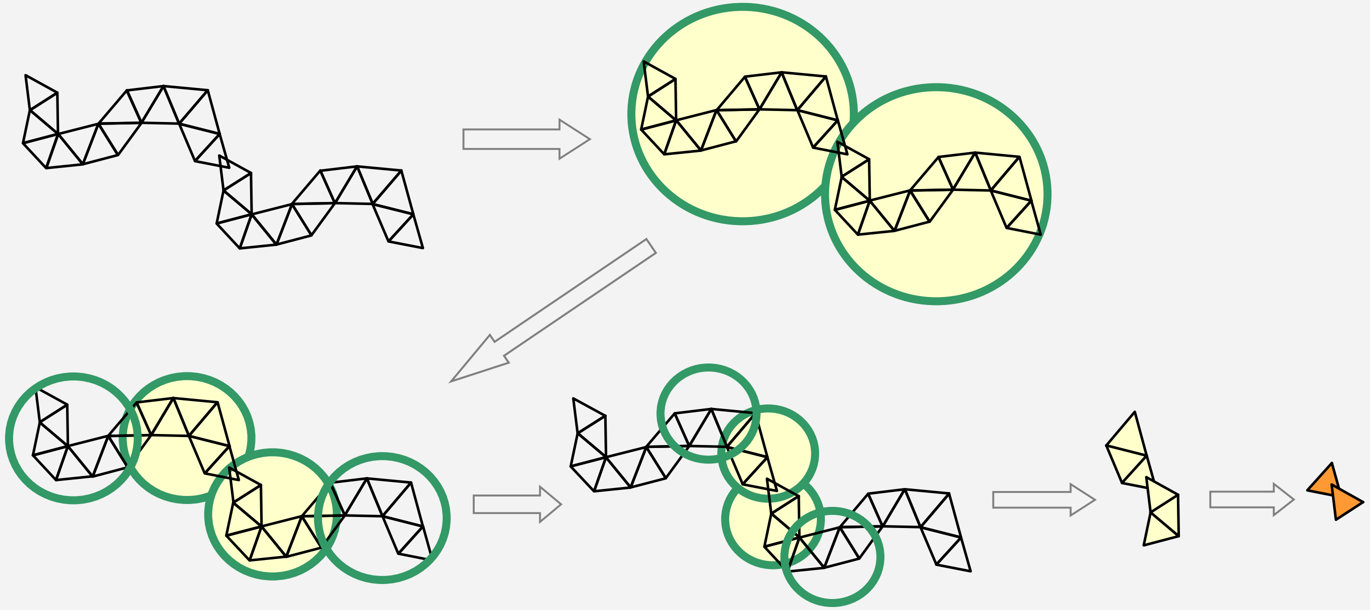


OBB
Layer 1



OBBs
Layer *n*

Overlap Test for BV Trees



- If BVs in a layer overlap, their children are checked
- At a leaf, primitives are tested with BVs and primitives
- Efficient culling of irrelevant object parts

UNI FREIBURG

Pseudo Code

1. Ooverlap test for two parent nodes (root)
2. If no overlap then "no collision" else
3. All children of one parent node are checked against children of the other parent node
4. If no overlap then "no collision" else
5. If at leaf nodes then "collision" else go to 3.

– Step 3. checks BVs or object primitives for intersection
– Required tests: BV-BV, (BV–primitive), primitive-primitive

Implementation - OBB tree, triangulated object

- All tests based on separating axis test
- Box-box
  - $3 + 3 + 3 \cdot 3$ axes have to be tested
- (Box-triangle) – not necessarily implemented
  - $3 + 1$ (face normals) $+ 3 \cdot 3$ (cross products of edges) tests
- Triangle-triangle
  - a) $1 + 1$ (face normals) $+ 3 \cdot 3$ (cross products of edges) tests
  - b) Testing all edge-edge pairs → 6 edge-triangle tests → 5 tests sufficient (intersections occur in pairs)

UNI FREIBURG

Edge-Triangle Test

$$\boldsymbol{x} = \boldsymbol{p}_0 + \mu_1(\boldsymbol{p}_1 - \boldsymbol{p}_0) + \mu_2(\boldsymbol{p}_2 - \boldsymbol{p}_0)$$

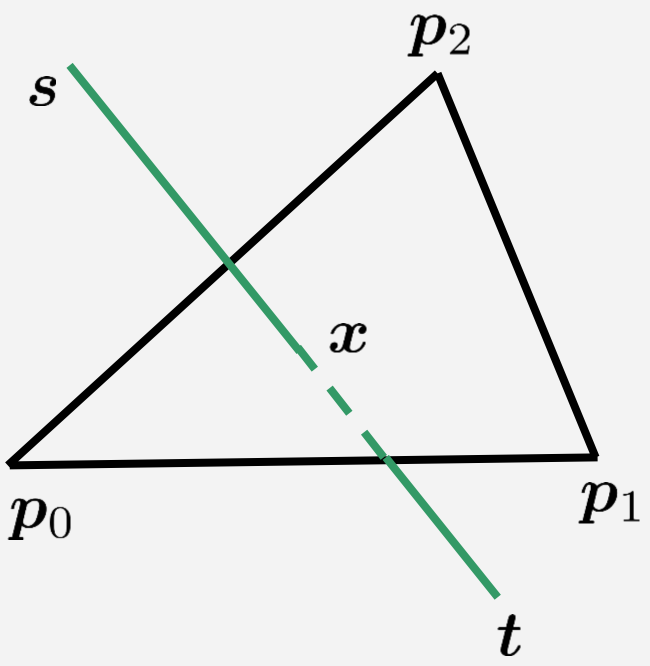
$$\mu_1, \mu_2 \geq 0 \quad \mu_1 + \mu_2 \leq 1$$

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

$$\boldsymbol{r} = \boldsymbol{t} - \boldsymbol{s} \quad \boldsymbol{d}_1 = \boldsymbol{p}_1 - \boldsymbol{p}_0$$

$$\boldsymbol{d}_2 = \boldsymbol{p}_2 - \boldsymbol{p}_0 \quad \boldsymbol{b} = \boldsymbol{s} - \boldsymbol{p}_0$$

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



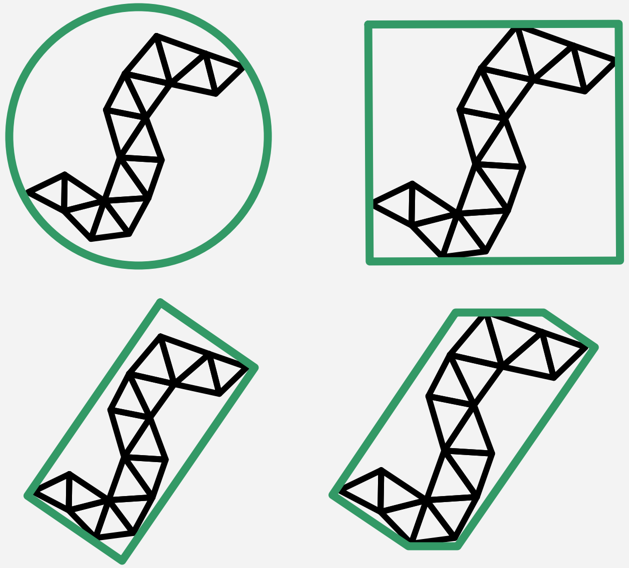
– The edge intersects if

$$\mu_1, \mu_2 \geq 0 \quad \mu_1 + \mu_2 \leq 1 \quad 0 \leq \lambda \leq 1$$

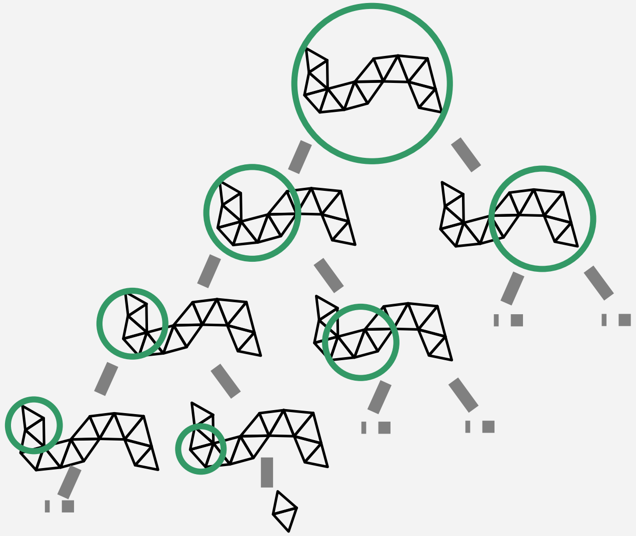
$$-\boldsymbol{r} \cdot (\boldsymbol{d}_1 \times \boldsymbol{d}_2) \neq 0$$

Summary - BVHs

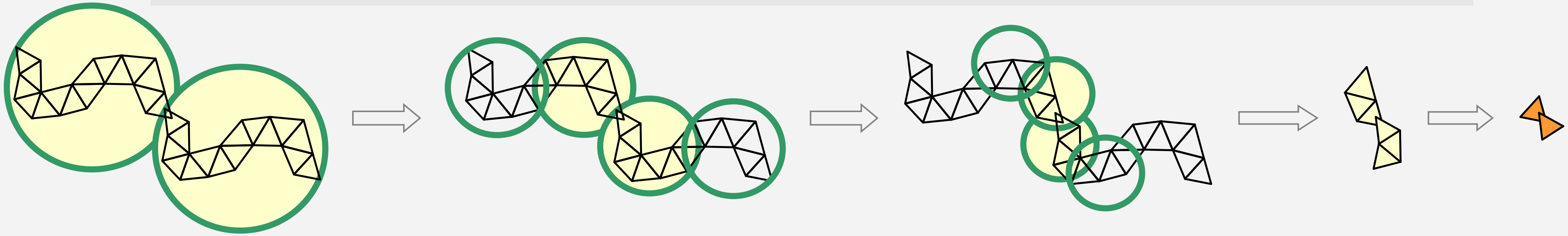
(1) Bounding volumes



(2) Bounding volume tree



(3) Collision detection test



UNI FREIBURG

Outline

- Introduction
- Bounding volumes BV
- Hierarchies of bounding volumes BVH
- Generation and update of BVs
- Design issues of BVHs
- Performance
- BVHs for deformable objects

UNI FREIBURG

Sphere

- Start with AABB of all points
  - The center of the sphere is given by the center of the AABB
  - The radius of the sphere is given by the largest distance from the center to a point
- Iteratively improve an initial guess (Ritter 1990)
  - Six extremal points of an AABB are computed
  - Choose pair of points with largest distance to get the center of the sphere and an initial guess of the radius
  - Iteratively enlarge the radius for points outside the sphere
- Minimum bounding sphere (Welzl 1991)
  - Randomized algorithm that runs in expected linear time
- Spheres can be translated and rotated with objects

AABB

- Compute six extremal points for the center and the radii
- AABBs can be translated with an object
- AABBs cannot be rotated with an object (the overlap test does not work for arbitrarily oriented AABBs)
- Rotation issue is addressed by
  - Computing the AABB for the bounding sphere
  - Update the AABB only considering the new positions of the original extremal points
  - Hill climbing on convex objects or pre-computed convex hulls of concave objects (check adjacent points of the original extremal points to update extremal points)

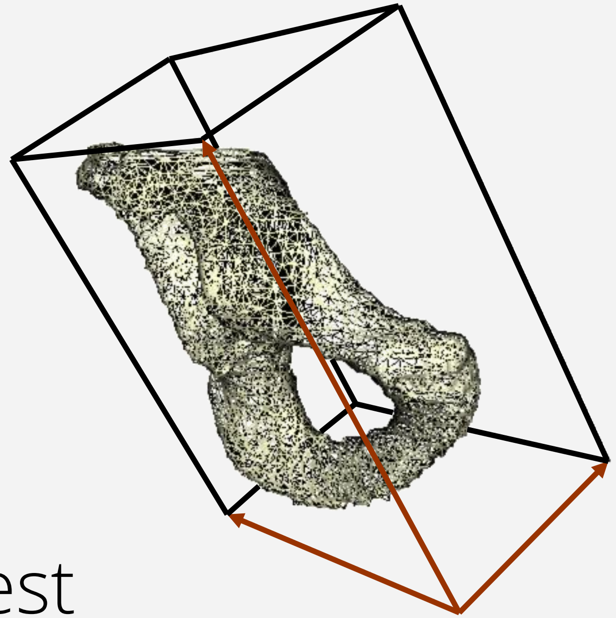
UNI FREIBURG

OBB

- Directions given by eigenvectors of the covariance matrix (PCA) (Barequet 1996)

$$\boldsymbol{C}_{jk} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \mu)_j(\boldsymbol{x}_i - \mu)_k \quad \mu = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$$

- Compute an AABB (Barequet 1999)
  - Choose two extremal points with the largest distance opposite to each other to define the first direction for the OBB
  - Choose two orthogonal directions
- Can be translated and rotated with an object



UNI FREIBURG

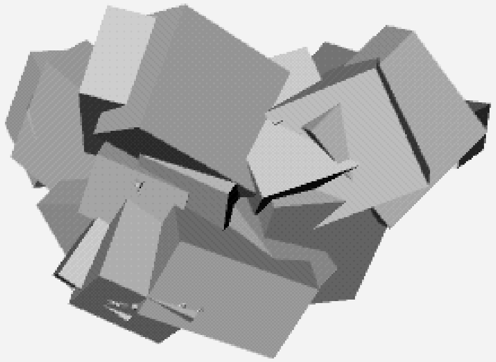
Outline

- Introduction
- Bounding volumes BV
- Hierarchies of bounding volumes BVH
- Generation and update of BVs
- Design issues of BVHs
- Performance
- BVHs for deformable objects

UNI FREIBURG

Construction of BVHs

- Goals
  - Balanced tree
  - Tight-fitting bounding volumes
  - Minimal redundancy
    (primitives in more than one BV per level)
- Parameters
  - BV type
  - Top-down / bottom-up
  - What and how to subdivide or merge: primitives or BVs
  - How many primitives per leaf in the BV tree
  - Re-sampling of the object



UNI FREIBURG

Outline

- Introduction
- Bounding volumes BV
- Hierarchies of bounding volumes BVH
- Generation and update of BVs
- Design issues of BVHs
- Performance
- BVHs for deformable objects

Collision Detection Libraries

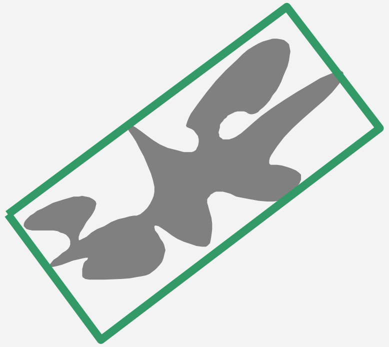
SOLID

Axis-aligned bounding box

van den Bergen
Eindhoven University
1997

RAPID

Object-oriented bounding box



Gottschalk et al.
University of North Carolina
1995

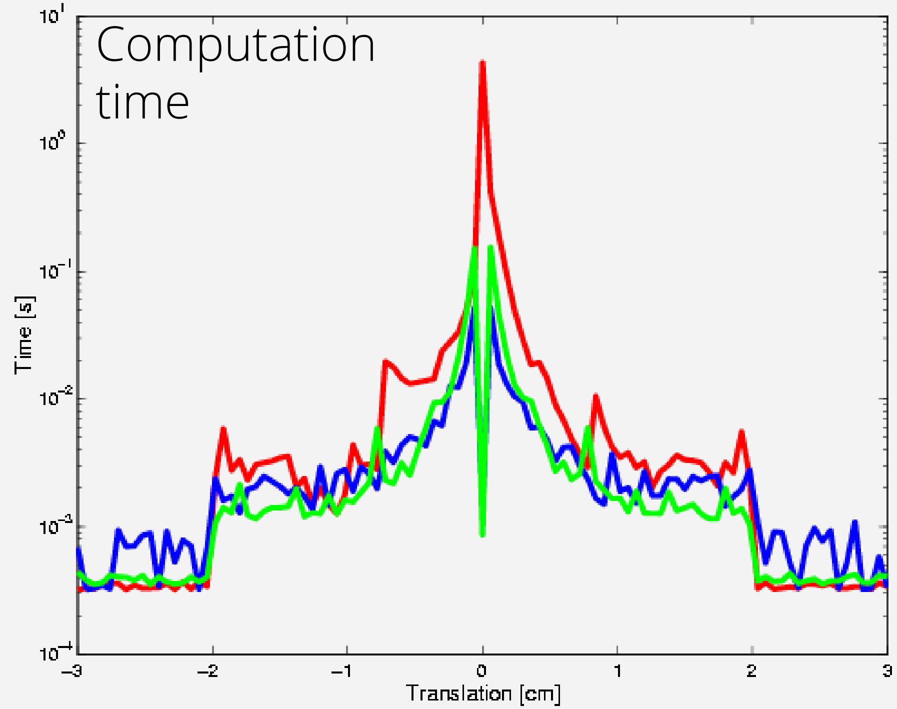
QuickCD

k discrete orientation polytope

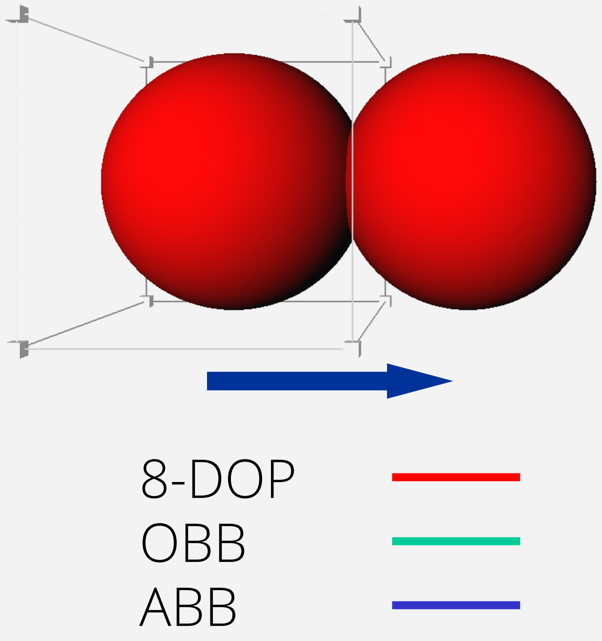
Klosowski et al.
University of New York
1998

Comparison

– Two spheres with radius 1 and 10000 triangles per sphere

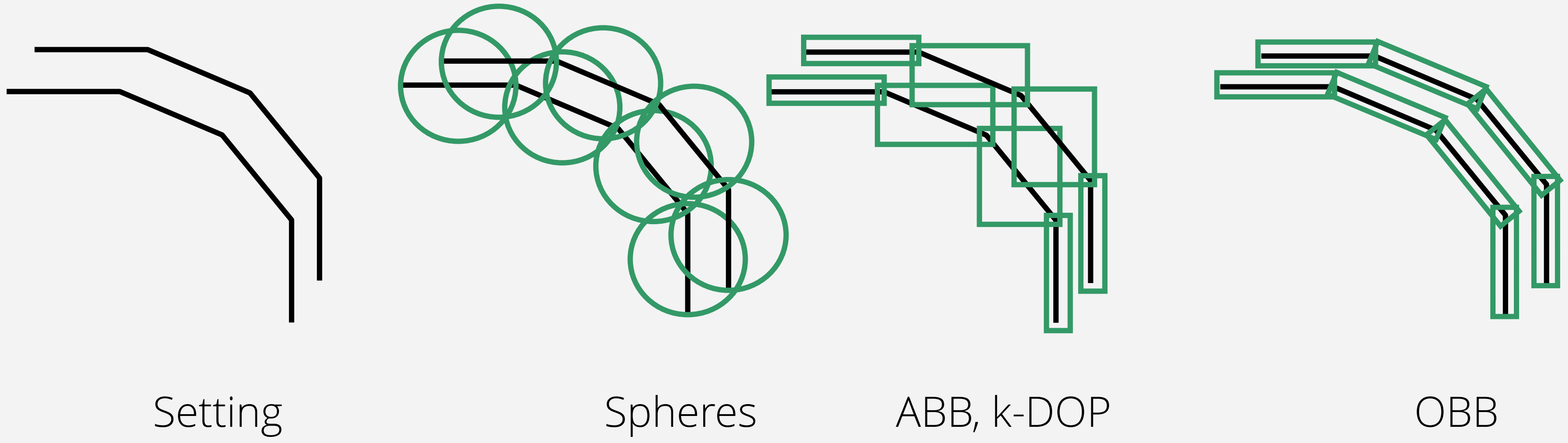


Distance between sphere centers



Close Proximity

- In case of close proximity,
  - Quality of higher layers influences the collision detection performance
  - Quality of lower layers BV approximations is less critical



UNI FREIBURG

Outline

- Introduction
- Bounding volumes BV
- Hierarchies of bounding volumes BVH
- Generation and update of BVs
- Design issues of BVHs
- Performance
- BVHs for deformable objects

UNI FREIBURG

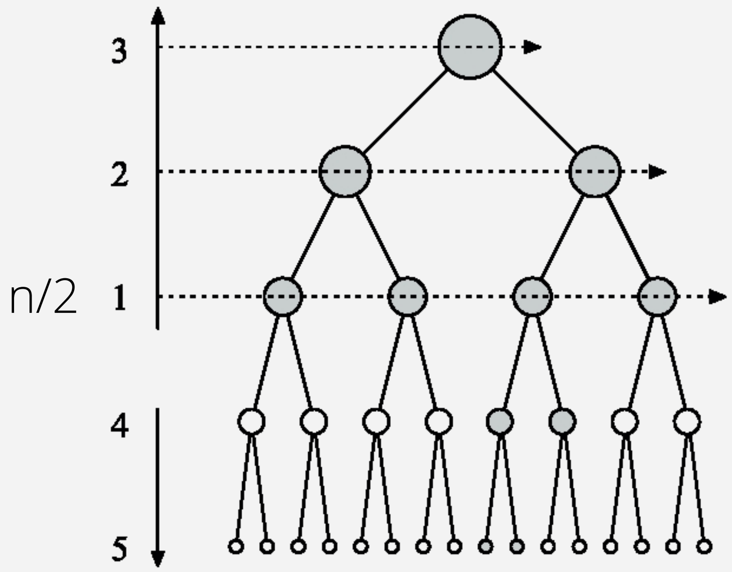
Deformable Objects

- BVH has to be updated in each time step
- Hierarchy generation significantly influences the performance
- AABBs are commonly used
- AABBs can be updated efficiently compared to OBBs or spheres
- AABBs do not optimally approximate an object

UNI FREIBURG

Adaptive Hierarchy Update

– AABB hierarchy
– Initial hierarchy generation
– During run-time
  – Adaptive update
  – Bottom-up update starting at layer *n*/2
  – Object is traversed once to update layer *n*/2
  – Efficient AABB update by merging AABBs of children
– Update of nodes in layer *n*/2+1 to n, only if required



Larsson / Akenine-Moeller
Eurographics 2001

Summary

- Introduction
- Bounding volumes BV
- Hierarchies of bounding volumes BVH
- Generation and update of BVs
- Design issues of BVHs
- Performance
- BVHs for deformable objects

References

- S. Quinlan, “Efficient Distance Computation Between Non-Convex Objects,” Proc. Int. Conf. on Robotics and Automation, pp. 3324-3329, 1994.
- 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.
- C. Ericson, "Real-Time Collision Detection," Elsevier, Amsterdam, ISBN: 1-55860-732-3, 2005.
- M. Teschner et al., "Collision Detection for Deformable Objects," Computer Graphics Forum, vol. 24, no. 1, pp. 61-81, March 2005.