ETH Zurich

Collision Detection

Motivation - Dynamic Simulation

ETH Zurich

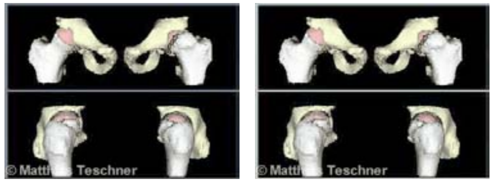
Stefan Gottschalk, UNC
Jim Cremer, Univ of Iowa

- Collision detection is an essential part of physically realistic dynamic simulations.

- For each time step:
    - compute dynamics
    - detect collisions
    - resolve collisions

Motivation - Biomedical Simulation

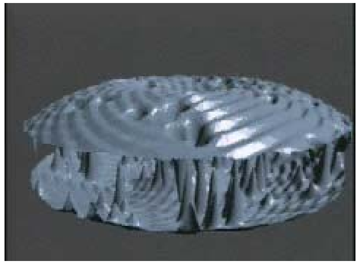
ETH Zurich



- Dysfunction of the hip joint due to reoriented femoral head
- Restricted range of motion in hip joint
- Simulation of the range of motion to decide whether a reorienting osteotomy of the proximal femur can improve the range of motion
- Triangulated surface representation of the bone structure
  -> collision detection problem

Motivation - Path Planning

ETH Zurich



Stefan Gottschalk, UNC
Jean-Claude Latombe, Stanford Univ

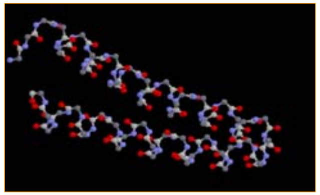
Planning a path for the torus involves collision detection tests.

Motivation - Robotics

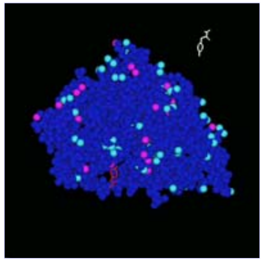
Sanchez, Latombe, Stanford Univ

- Collision detection is required to coordinate all tasks for all robots
- Can be time-critical in case of a break-down

Motivation
Molecular Motion, Protein Folding



Singh, Latombe, Brutlag, Stanford Univ



- Computation of low-energy states for designed drug molecules

---

Outline

Bounding Volumes

Hierarchies of Bounding Volumes

Spatial Partitioning

Distance Fields

Collision Detection
for Deformable Objects

Bounding Volumes

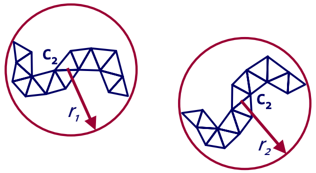
Simplified conservative surface representation for fast approximative collision detection test

- **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.

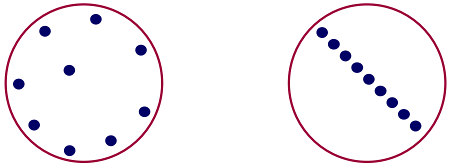
Spheres and Overlapping Test

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 - ABB

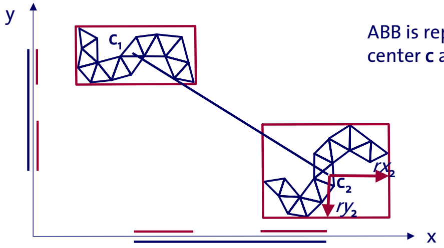


ABB is represented by center **c** and radii *rx*, *ry*.

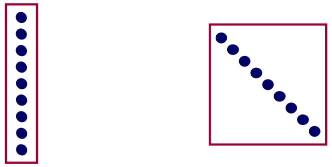
Two ABBs do not overlap in 2D if

$$\left| (\mathbf{c}_1 - \mathbf{c}_2) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right| > rx_1 + rx_2$$

or

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

ABB as Bounding Volume

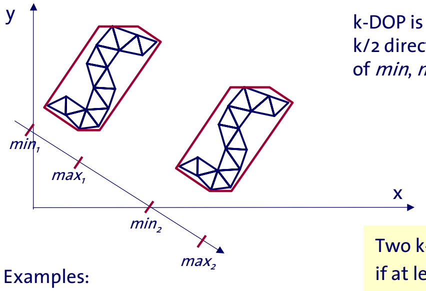


good choice.

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]



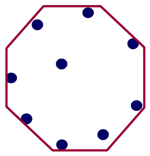
k-DOP is represented by k/2 directions and k/2 pairs of *min*, *max* values.

Examples:
6-, 14-, 18-, 26-DOPs

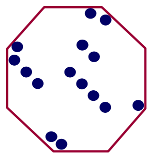
Two k-DOPs do not overlap, if at least the intervals in one direction do not overlap.

k-DOPs as Bounding Volumes

Example: 8-DOP
Larger *k*'s are more flexible than smaller.
ABB is a 4-DOP. Is a 4-DOP an ABB?

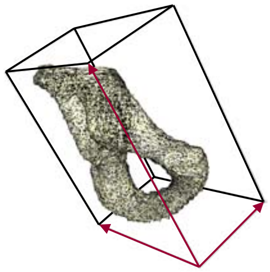


good choice.



quite good choice.

Object-Oriented Bounding Box OBB



Eigenvectors of the covariance matrix

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

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

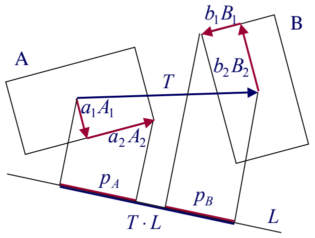
Mean: $\mu = \frac{1}{n} \sum_{i=1}^{n} v_i$

Covariance matrix:

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

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

OBB Overlapping Test in 2D



A_1, A_2, B_1, B_2 • axes of A, B
• unit vectors

a_1, a_2, b_1, b_2 • 'radii' of A, B

L • unit vector

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

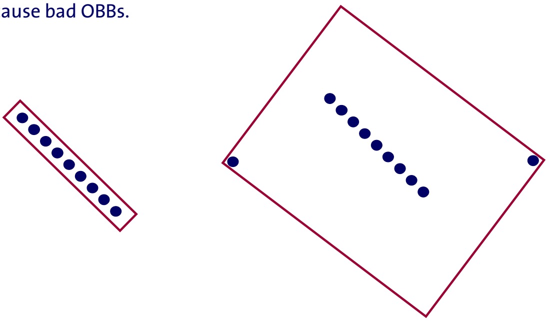
$$p_B = |b_1 B_1 L| + |b_2 B_2 L|$$

A, B do not overlap:

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

OBB Examples

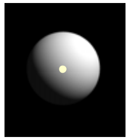
- Principal axes of an object are not always a good choice for the main axes of an OBB!
- Inhomogeneous vertex distribution can cause bad OBBs.



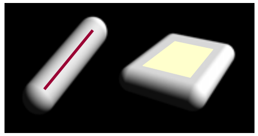
Bounding Volumes - Summary

- Spheres
- Axis-aligned bounding boxes (ABB)
- Object-oriented bounding boxes (OBB)
- Discrete orientation polytopes (k-DOP's)

- Ellipsoids
- Convex Hulls
- Swept-Sphere Volumes (SSV's)
  - Point Swept Spheres (PSS)
  - Line Swept Spheres (LSS)
  - Rectangle Swept Spheres (RSS)
  - Triangle Swept Spheres (TSS)



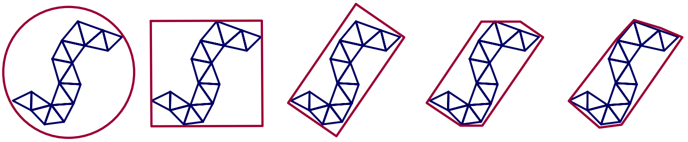
PSS



Lin, UNC

LSS RSS

The Optimal Bounding Volume ?



Sphere ABB OBB 6-DOP Convex Hull

Quality:

Better approximation

→

←

Decreasing complexity and computational expenses for overlap test

---

Outline

Bounding Volumes

Hierarchies of Bounding Volumes

Spatial Partitioning

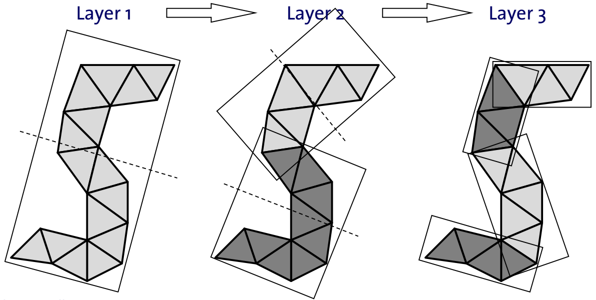
Distance Fields

Collision Detection for Deformable Objects

Hierarchy of Bounding Volumes

ETH Zurich

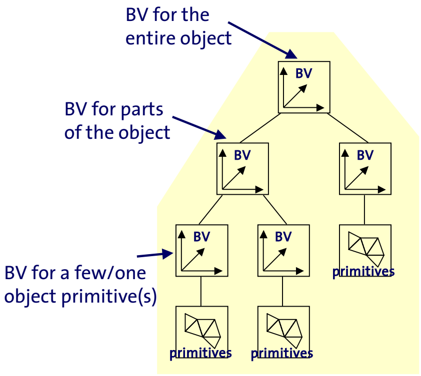
- Not only one bounding volume, but a hierarchical structure of bounding volumes
- Subdivsion of bounding volumes to generate a hierarchy



Hierarchy of Bounding Volumes

ETH Zurich

- Bounding volume tree (BV tree)
- Nodes contain bounding volume information
- Leaves additionally contain information on object primitives

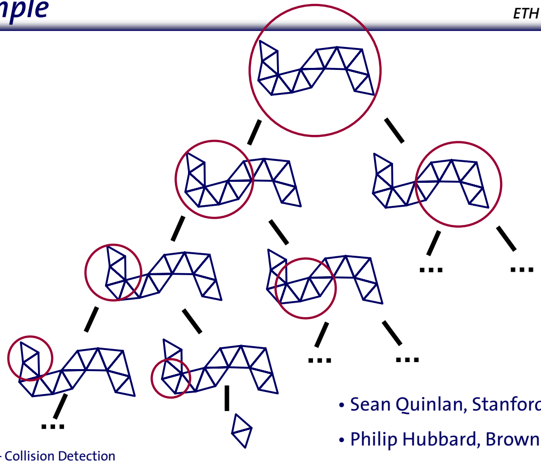

BV for the entire object

BV for parts of the object

BV for a few/one object primitive(s)

Hierarchy of Bounding Volumes - Example

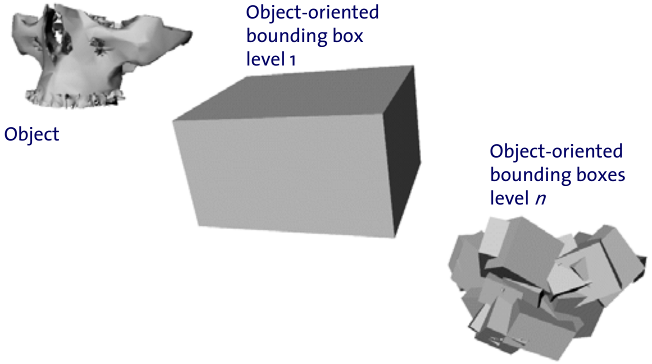
ETH Zurich



- Sean Quinlan, Stanford Univ
- Philip Hubbard, Brown Univ

OBB Tree

ETH Zurich


Object-oriented bounding box level 1

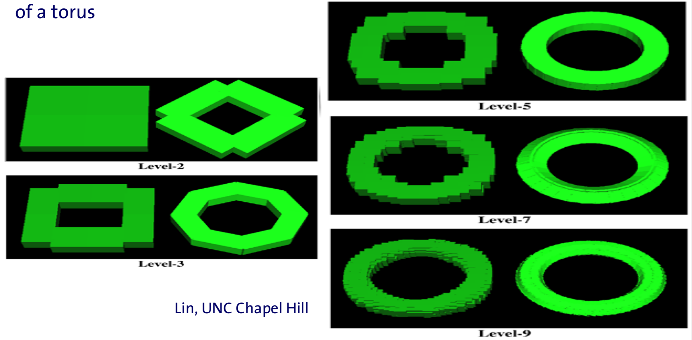
Object

Object-oriented bounding boxes level *n*

ABB vs. OBB Tree

ETH Zurich

Representation
of a torus

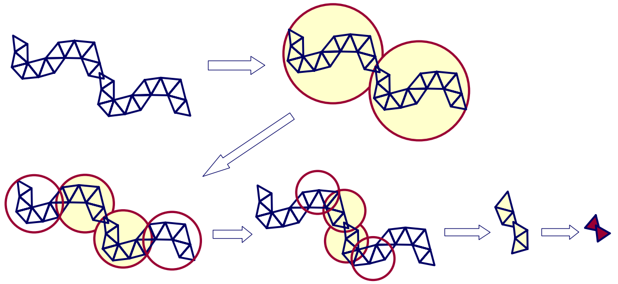


Lin, UNC Chapel Hill

Overlapping Test for BV Trees

ETH Zurich

- 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.



Construction of a BV Tree

ETH Zurich

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

ETH Zurich

Parameters

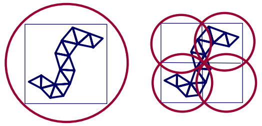
- Bounding volume
- Top-down vs. bottom-up
- What to subdivide / group: object primitives or bounding volumes
- How to subdivide / group object primitives or bounding volumes
- How many primitives in each leaf of the BV tree
- Re-sampling of the object ?

Goals

- Balanced tree
- Tight-fitting bounding volumes
- Minimal redundancy
  (primitives in more than one BV per level)

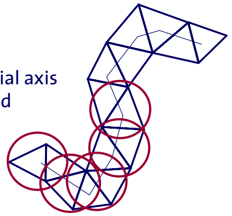
Construction of a BV Tree - Spheres

Hubbard:
- 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

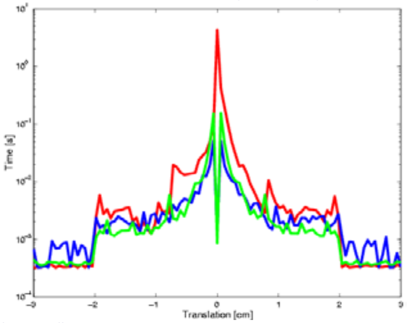


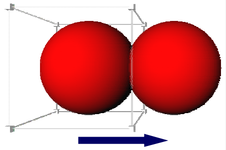
Collision Detection Libraries

SOLID	*RAPID*	*QuickCD*
Axis-aligned bounding box	Object-oriented bounding box	k discrete orientation polytope
		
van den Bergen Eindhoven University 1997	Gottschalk et al. University of North Carolina 1995	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]
- Differences mostly due to programming aspects





10,000 triangles per sphere

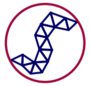
8-DOP

OBB

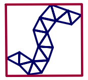
ABB

Object Transformations

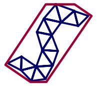
Some object transformations can be simply applied to all elements of the bounding-volume tree:



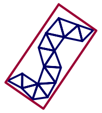
Spheres
- Translation, rotation



Axis-Aligned Bounding Boxes
- Translation, no rotation



Discrete Orientation Polytopes
- Translation, no rotation (principal orientations are fixed for all objects)



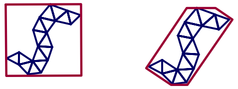
Object-Oriented Bounding Boxes
- Translation, rotation (box orientations are not fixed)

Rotations

ETH Zurich

Axis-Aligned Bounding Boxes

Discrete Orientation Polytopes



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

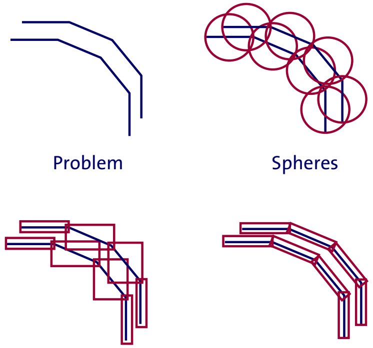
1. Recomputation of the BV hierarchy
2. Preservation of the tree structure, update of all nodes
3. Additional storage of the convex hull which is rotated with the object
   - check if extremal vertices are still extremal after rotation
   - compare with adjacent vertices of the convex hull
   - “climb the hill” to the extremal vertex
4. Computation of an approximate box by rotating the box and checking the rotated box for extremal values

---

Parallel Close Proximity

ETH Zurich

- Quality of *lower*-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

Hierarchical Bounding Volumes - Summary

ETH Zurich

- 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)

Outline

ETH Zurich

Bounding Volumes

Hierarchies of Bounding Volumes

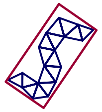
Spatial Partitioning

Distance Fields

Collision Detection
for Deformable Objects

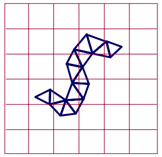
Bounding Volume Hierarchy vs. Spatial Partitioning

Bounding Volume Hierarchy



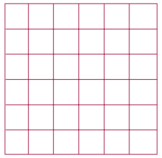
Object related

Spatial Partitioning

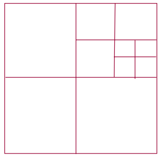


World (space) related

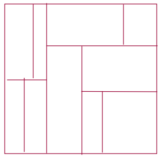
Spatial Data Structures for Collision Detection



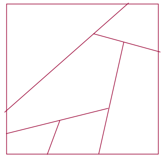
uniform grid



quadtree / octree



kd-tree



BSP-tree

- Each cell contains information whether it is occupied by an object -> Possible collision!
- Information is updated for each object transformation.
- quadtree, kd-tree, and BSP-tree are hybrid approaches. They subdivide space, but object-dependent.

Uniform Grid

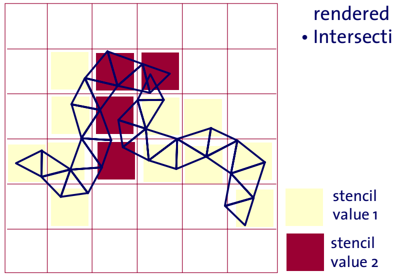
- Each cell contains information which objects are in it. -> Possible collision!

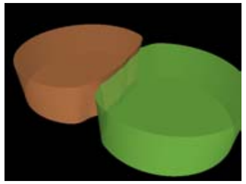
empty	empty	obj 1 obj 2	obj 2	empty	empty
empty	obj 1	obj 1 obj 2	obj 1 obj 2	empty	empty
empty	obj 1	obj 1	obj 2	empty	empty
empty	obj 1	obj 1	obj 2	empty	empty
empty	obj 1	obj 1	empty	empty	empty
empty	empty	empty	empty	empty	empty

Graphics Hardware for 2D Collisions

frame buffer
is a uniform grid

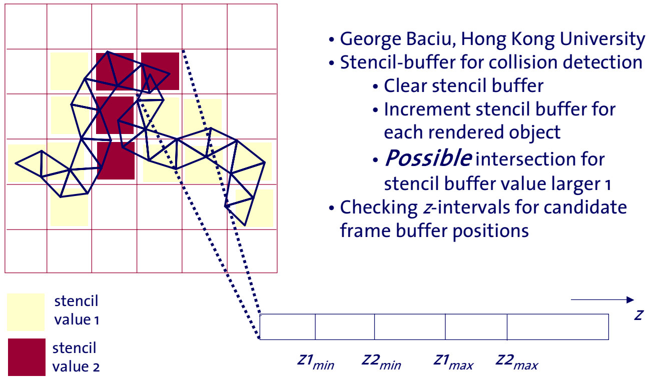
- Kenneth Hoff, UNC
- Stencil-buffer for collision detection
    - Clear stencil buffer
    - Increment stencil buffer for each rendered object
    - Intersection for stencil buffer value larger 1





Hoff, UNC

Graphics Hardware for 3D Collisions



- George Baciu, Hong Kong University
- Stencil-buffer for collision detection
  - Clear stencil buffer
  - Increment stencil buffer for each rendered object
  - ***Possible*** intersection for stencil buffer value larger 1
- Checking *z*-intervals for candidate frame buffer positions

---

Outline

Bounding Volumes

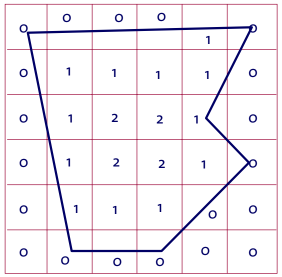
Hierarchies of Bounding Volumes

Spatial Partitioning

Distance Fields

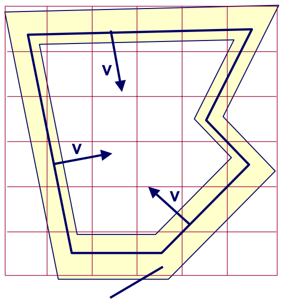
Collision Detection
for Deformable Objects

Distance Fields



- **Distance field**: Minimal distance from each point within an object to the object surface, outside points are zero
- Distance field is transformed with respect to object transformation
- Primitives of other objects are checked against the distance field of an object
- -> value larger 0 -> collision
- Distance value represent penetration depth and can be used for collision response, e.g. penalty forces.

Distance Field Generation



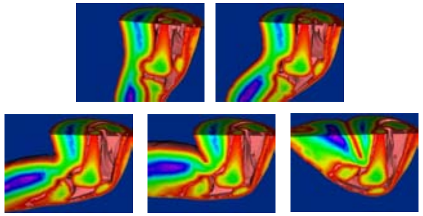
narrow band area

- Susan Fisher, Ming Lin, UNC
- Fast generation of distance fields by using **level set method**
- Initialization of a narrow band area with distance values
- Consideration of the surface as a moving wave
- Computing the arrival time for each point inside the object (linear proportional to the distance to the original surface)
- Very fast algorithm
- Time / distance is derived from adjacent values

Distance Fields - Applications

- Susan Fisher, Ming Lin, UNC
- Collision detection based on distance fields for deformable objects

Fisher, UNC



Fisher, UNC

Outline

Bounding Volumes

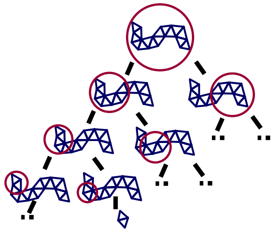
Hierarchies of Bounding Volumes

Spatial Partitioning

Distance Fields

Collision Detection
for Deformable Objects

Sphere Trees for Deformable Objects



- Stephen Sorkin, J.C. Latombe, Stanford Univ
- **Problem**: Sphere tree has to be updated in case of object deformation
- Bottom-up approach for updating the tree
- Update of leafs only for deformed primitives
- Propagating leaf information to the root

Collision Detection for Deformable Objects

Bounding Volume Hierarchy

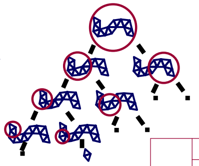
- Time-consuming generation of hierarchies
    - Preprocessing step in case of rigid bodies
    - Focus on quality of the approximation
- (Partly) update of the hierarchy required in case of deformation

Spatial Partitioning

- Same handling of rigid and deformable bodies
- No preprocessing

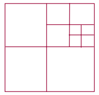
Summary

ETH Zurich



Hierarchies of Bounding Volumes

Simplification of collision detection by using approximate bounding volumes (spheres, boxes, …)

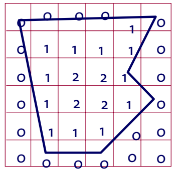


Spatial Partitioning

Using information about the position of object primitives for collision detection

Distance Fields

Computation of surface distance for all object points



Collision Detection for Deformable Objects

Distance fields, bounding volume hierarchies, spatial partitioning

---

References

ETH Zurich

- **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 Transactions on Graphics*, 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.
- **S. Fisher**, M. Lin, ”Deformed Distance Fields for Simulation of Non-Penetrating Flexible Bodies,” *Eurographics Workshop on Computer Animation and Simulation (CAS),* September 2001.
- **G. Baciu**, S. K. Wong "Image-Based Techniques in a Hybrid Collision Detector," *IEEE Transactions on Visualization and Computer Graphics*, Jan 2002.