Collision Detection for Deformable Objects

M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann,
M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Strasser, P. Volino

Computer Graphics Laboratory, University of Freiburg, Germany
Computer Graphics Laboratory, ETH Zurich, Switzerland
WSI/GRIS, Tübingen University, Germany
Miralab, University of Geneva, Switzerland
GRAVIR/IMAG, INRIA Grenoble, France
Bonn University, Germany
Fraunhofer Institute for Computer Graphics, Darmstadt, Germany

- Presentation prepared by: Przemysław Bągard
- Seminar: Innovative applications of computer science
Deformable Collision Detection

• Usages:
  – Cloth simulation
  – Surgery simulation

• Expected key features:
  – Many shapes – precision of 3D model
  – Dynamic – often there are changes of shapes within the same body
  – Measurement precision
  – Speed – Collision Detection and shape deformation should be gained quickly
Aspects of Collision Detection

• Preprocessing
  – We can decrease number of collision detection candidates for example by spatial data structures, where we place those models

• Collisions and self-collisions
  – Model can collide with itself

• Collision information
  – Not only fact of detection, but also data computed e.g. normal vector
Bounding Volume Hierarchies

- Usually built in preprocessing
- In general, BVHs are defined as follows: Each node in the tree is associated with a subset of the primitives of the object, together with a BV that encloses this subset with a smallest containing instance of some specified class of shapes.
- For the collision test of two objects or the self collision test of one object the BVHs are traversed top-down and pairs of tree nodes are recursively tested for overlap.
Bounding Volume (BV)

AABB are preferred
Figure 5: Three levels of an 18-DOP-hierarchy created by splitting the parent DOPs along the longest axis.
Hierarchy Update

- In contrast to hierarchies for rigid objects, hierarchies for deformable objects need to be updated in each time step.

- Question: what is faster? Update or rebuild tree?

- Refitting for Bounded Deformations
  - If the deformation of the object are superpositions of displacement fields

- Refitting for Morphing Objects
  - Object are constructed by interpolating between two or more morph targets.
Stochastic Methods

- Method 1: Using probabilistic methods estimate the possibility of a collision with respect to a given quality criterion.
  - "quality" (or time) can be specified by user
  - algorithm is guided by the probability that a pair of BVs contains intersecting polygons
Method 2: “Guess” colliding pairs by a stochastic sampling within the colliding bodies. The exact colliding regions are then narrowed down by using this principle in conjunction with temporal and spatial coherence.

- If a pair of features is close enough at a time step, it may still be interesting in the next one (track colliding regions over subsequent time steps).
Distance Fields

- Distance fields specify the minimum distance to a closed surface for all points in the field
  - The distance may be signed in order to distinguish between inside and outside.

Figure 13: Happy Buddha and three color-mapped distance field slices. Since the distance field is only valid within a band near the surface, the mapping is faded out at larger distance. Blue maps to close distances, whereas red indicates medium distances.
Distance Fields

- Construction time is slow
- Detection time is high
- Good solution for non-interactive applications, where distance field can be calculated during preprocessing stage
- For interactive applications, distance fields can be used to represent all rigid objects contained within the environment.
Spatial Subdivision

- 3D grid
  - Cells are mapped by hash function
  - As opposed to BSP trees, spatial subdivision with uniform grid is independent of the object
Spatial Subdivision

- Example [THM*03], (spatial hashing with a uniform grid is employed for the detection of collisions for deformable tetrahedral meshes)
  - information on the implicit 3D grid cells of all vertices are mapped to the hash table
  - maps information on all grid cells touched by a tetrahedron to the hash table
  - checks vertices and tetrahedrons within a hash table entry for intersections. If a vertex penetrates a tetrahedron, a collision is detected. If both, the vertex and the tetrahedron belong to the same object, a selfintersection is detected.
Image-Space Techniques

- Example (method with Two Depth buffers)
  - (Depth buffer: for each pixel store distance from viewer)
  - First object to first buffer, second object to second buffer and then check distances between buffers (for each pixel in the first buffer and corresponding pixel in the second one)

- Imprecise data
Thank you