Collision Detection for Medical Applications

Gabriel Zachmann
Clausthal University, Germany
zach@tu-clausthal.de

Eurographics, Crete, 16. February 2008
Applications of Collision Detection

- Physically-based simulation of rigid anatomical parts ...
- ... and of deformable parts

SensAble

Courtesy Raghupathi et al., INRIA
- Force feedback (e.g. in training simulators)
- And numerous apps outside of the medical domain
Rigid Objects

- Standard method: bounding volume hierarchies (BVH)

- Simultaneous traversal of two BVHs = single traversal of one conceptual BV test tree (BVTT)
Variations

- Type of bounding volumes:
  - AABB (axis-aligned b.-box) (R*-trees)
  - k-DOP (discretely oriented polytope)
  - OBB (oriented bounding box)

- Arity of the BVHs:
  - Most prefer 2-ary or 4-ary
  - Particularly well-suited for SSE implementations

- Kind of traversal:
  - Depth-first or breadth-first
Current Performance

![Graph showing performance of Cobra with different models](image)

- **Motivation**
- **Rigid CD**
- **Other Approaches**
- **Deformable Objs**
- **Conclusion**
Object-Space Coll. Detection on the GPU

- Implementation:
  - List of BVs = stream → texture
  - BV intersection test = kernel → fragment program
Time-Critical Collision Detection

- **Goal:**
  - Continuous and controlled balancing between running time and accuracy; i.e.,
  - **Time-critical computation** of collision detection queries

- **Approach:**
  - Stochastic, *average-case* approach
  - Idea: guide traversal of BVTT by probability (→ p-queue)
  - Modification of BVHs: store *simple* description → ADB trees
Result

![Graph showing time in milliseconds against distance]

- Blue line: \(p_{\text{min}} = 0.99\)
- Red line: \(p_{\text{min}} = 0.90\)
- Green line: \(p_{\text{min}} = 0.80\)

**Motivation**

- Rigid CD
- Other Approaches

**Deformable Objs**

**Conclusion**
Collision Detection on Point Clouds

- Motivation: renaissance of points as object representation because of 3D scanners

- Goal:
  - Fast collision detection between 2 given point clouds
  - No polygonal reconstruction
Approach

- Given two point clouds A and B, construct a stochastic sampling of

\[ \mathcal{Z} = \{x \mid f_A(x) = f_B(x) = 0\} \]

- Overall method:
Results

- Theoretical complexity: $O(\log \log N)$
Deformable Objects

- Most objects in medical applications are (probably) deformable

- Use BVHs and update them somehow:
  - Brute-force update bottom-up
  - BV inflation with conservative estimate of motion of vertices
  - **Kinetize** the BVH
    - Augment data structure such that only combinatorial changes, which occur only at discrete points in time, need to be handled
    - Update time is $O(n \log n)$, *independent* from query frequency
Performance of Kinetic AABB

Shirt Scene (~ 100,000 triangles)

Avg time per BVH update / msec

Number of in-between frames

- Kinetic AABB
- Bottom-Up
- Use "naked trees" and compute conservative BVs "as you go"
  - Only for special kinds of deformations, and with limited amounts

- Use BVHs and reconstruct every time
  - Use very simple construction algorithm
  - Reconstruct only the most deteriorated parts

- Use space partitioning scheme and update that
  - Most popular today: grid with hashing
Don't use BVHs nor space partitioning schemes at all:

- Use GPU, compute collision detection by "brute-force" in image space (e.g., clip edges against stencil buffer)
- Use NURBS, tessellate and compute BVs on the fly
- Sample mesh stochastically, update by closest features technique
- Use point clouds with our stochastic approach
Conclusion

- Have not touched on continuous collision detection
- Collision detection for rigid bodies is fairly well researched
- For deformable bodies: still room for improvement