Parallel Collision Detection in Constant Time

Rene Weller, Udo Frese and Gabriel Zachmann
University of Bremen
cgvr.cs.uni-bremen.de
Motivation
Motivation

\[ O(n^2) \]

[Haddadin, 2009]
Previous Works

- Distance of convex polytopes [Lin & Canny, 1991]:
  \[ O(\sqrt{n}) \], \quad n = \# \text{ faces} \\
- All intersections of \( n \) convex polytopes [Suri et al., 1998]:
  \[ O((n+k) \log n) \], \quad k = \# \text{ intersecting pairs} \\
- Voxel based algorithms [McNeely et al., 1999]
  - Worst case: \( O(n) \) \( n=\# \text{ points in pointshell} \)
  - Fast, independent of object’s triangle count
  - Memory consuming, aliasing artifacts
Preliminary Considerations

- Fill object
  - from the *inside*
  - with *non-overlapping*
  - *polydisperse* spheres
- Independent of object representation
- Penetration volume
  - “the most complicate yet accurate method” [Fisher and Lin, 2001]
  - Continuous forces and torques
- Complexity: ?
Theoretic Fundament
Theoretic Fundament

- **Lemma**: A single sphere $s$ intersects a **constant** number of disjoint spheres $A$ with at least the same radius.

- **Theorem**: The maximum number of intersecting pairs of spheres of two polydisperse sphere packings $A$ and $B$ with $n$ spheres is in $O(n)$.

  - Check each sphere $s \downarrow i \in A$ against larger spheres in $B$
  - Check each sphere $s \downarrow j \in B$ against larger spheres in $A$

  $\Rightarrow O(1)$ for each $s \downarrow i \in A$ and $s \downarrow j \in B$

  $\Rightarrow O(n)$ in total
Our Algorithm

Level 0

Level 1

Level 2

Level 3

Level 4

l

l

\frac{l}{2}
### Analysis

For each sphere \( s \in B \):

- Compute hierarchy level \( l \)

For all levels \( 1 \leq l \leq l_{\text{max}} \):

- For all cells \( c_{lj} \) in level \( l \) overlapped by \( s \)

  For all spheres \( s_{jk} \in c_{lj} \)

  Compute overlap volume for \( s \) and \( s_{jk} \)

\[ \Rightarrow O(n) \]

\[ \Rightarrow O(1) \]

\[ \Rightarrow O(1) \]

\[ \Rightarrow O(1) \]

\[ \Rightarrow O(1) \]

\[ \Rightarrow O(1) \]

\[ \Rightarrow O(1) \]
Results

The diagrams illustrate the simulation time (y-axis) versus the simulation time (x-axis) for different numbers of spheres. The graphs show the performance of simulations for different animals: Pig, Cow, and Armadillo, with varying numbers of spheres (4k, 8k, 12k, 16k, 20k, 30k, 40k, 50k, 60k). The graphs indicate how the simulation time increases with the number of spheres.
Conclusions and Future Work

- Proven $O(n)$ complexity for the overlap of two sphere packings
- New collision detection algorithm
  - Worst case sequential time: $O(n)$
  - Worst case parallel time: $O(1)$
- Geforce GTX680: < 1 msec for 30k spheres
- Extension to deformable objects
  - Sphere-Spring-Systems
- Support of thin sheets
Thank You!

- DFG Grant SFB/TR 8 I08 [DextrousSpace], GZ TRR 8/3-2013