Motivation

- Collision Detection is ubiquitous in VR and many physically-based simulation apps
- Obviously: worst-case running time is in $O(n^2)$
- But, we all have seen real-world running time behavior like this:
Goal

- Gain (theoretical) understanding of experienced running times
- Utilize to optimize collision detection
- Better heuristics for probabilistic collision detection

Related Work

- Distance of convex polytopes [Dobkin & Kirkpatrick, 1985]:
  \( O(\log^2 n) \), \( n \) = number of faces
- Distance of convex polytopes [Lin & Canny, 1991]:
  \( O(\sqrt{n}) \), worst-case
  \( O(1) \), expected time, bounded rotation
- General polytopes, fixed trajectory [Schömer & Thiel, 1995]:
  \( O(n^{\frac{5}{3} + \varepsilon}) \)
- All intersections of \( n \) convex polytopes [Suri et al., 1998]:
  \( O((n + k) \log^2 n) \), \( k \) = \# intersecting pairs
Hierarchical CD

- Hierarchical CD is most common technique for rigid bodies
- BV hierarchy (BVH) is constructed in preprocessing:

  ![Diagram of BVH hierarchy]

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

Overview

- Lots of different BVs have been proposed, e.g.:

  - Cylinder [Weghorst et al., 1985]
  - Box, AABB (R*-trees) [Beckmann, Kriegel, et al., 1990]
  - Sphere [Hubbard, 1996]
  - Prism [Barequet, et al., 1996]
  - Spherical shell [Krishnan, et al., 1997]
  - Cylinder [Weghorst et al., 1985]
  - Box, AABB (R*-trees) [Beckmann, Kriegel, et al., 1990]
  - Sphere [Hubbard, 1996]
  - Prism [Barequet, et al., 1996]
  - Spherical shell [Krishnan, et al., 1997]
  - Cylinder [Weghorst et al., 1985]
  - Box, AABB (R*-trees) [Beckmann, Kriegel, et al., 1990]
  - Sphere [Hubbard, 1996]
  - Prism [Barequet, et al., 1996]
  - Spherical shell [Krishnan, et al., 1997]

- In the following: use AABBs
The Cost formula \cite{Weghorst et al. 1984; Gottschalk et al. 1996}:

$$T = N_V C_V + N_P C_P + N_u C_u + C_i$$

- \(N_V, C_V\) = num., costs of BV overlap test, resp.
- \(N_P, C_P\) = num., costs of primitive intersection test
- \(N_u, C_u\) = num., costs of BV update, resp.
- \(C_i\) = initialization costs

- Obviously: \(T(n) \sim N_V(n)\)

- Goal: determine \(E[N_V(n)] = \tilde{N}_V(n)\)
  \(=\) number of nodes in the BVTT that are visited on average

The Model to Determine \(\tilde{N}_V(n)\)

- Assumption: use AABBS
- Estimate probability of BV overlap on some level \(l\)
- Yields product of conditional probabilities
- Estimate conditional probability by geometric reasoning
Terminology

- \( P[A^{(l)} \cap B^{(l)} \neq \emptyset] \) = probability that two AABBs on level \( l \) overlap each other
- In the following, just write \( P[A^{(l)} \cap B^{(l)}] \)
- X-Overlap \( o_x := \) length of overlap of slabs of AABBs

The Chain of Probabilities

- Obviously, the expected total number of BV overlaps is
  \[
  \hat{N}_v(n) = \sum_{i=1}^{d} \hat{N}_v^{(i)} = \sum_{i=1}^{d} 4^i P[A^{(l)} \cap B^{(l)}] \quad (1)
  \]
- Recall that \( X \subseteq Y \Rightarrow P[X] = P[Y] \cdot P[X \mid Y] \)
- Turn \( P[A^{(l)} \cap B^{(l)}] \) into conditional probability that "defers" the probability up one level in the hierarchy:
  \[
  P[A^{(l)} \cap B^{(l)}] = P[A^{(l)} \cap B^{(l)} \mid A^{(l-1)} \cap B^{(l-1)} \wedge o_x^{(l)} > 0] \\
  \cdot P[A^{(l-1)} \cap B^{(l-1)} \wedge o_x^{(l)} > 0]
  \]
• Resolve further:

\[ P[ A^{(i)} \cap B^{(i)} ] = P[ A^{(i)} \cap B^{(i)} \mid A^{(i-1)} \cap B^{(i-1)} \land o_x^{(i)} > 0 ] \]
\[ \cdot P[ A^{(i-1)} \cap B^{(i-1)} ] \]
\[ \cdot P[ o_x^{(i)} > 0 \mid A^{(i-1)} \cap B^{(i-1)} ] \]

• "Unroll" recurrence:

\[ P[ A^{(i)} \cap B^{(i)} ] = \prod_{i=1}^{j} P[ A^{(i)} \cap B^{(i)} \mid A^{(i-1)} \cap B^{(i-1)} \land o_x^{(i)} > 0 ] \cdot \prod_{i=1}^{j} P[ o_x^{(i)} > 0 \mid A^{(i-1)} \cap B^{(i-1)} ] \]

### The Geometric Probability

• Goal: estimate \( P[ A^{(i)} \cap B^{(i)} \mid A^{(i-1)} \cap B^{(i-1)} \land o_x^{(i)} > 0 ] \)

• Terminology:
  - Denote parent boxes by \( A, B \)
  - Denote extents by \( a_x, a_y, b_x, b_y \)
  - Denote child boxes by \( A_1, A_2, B_1, B_2 \)
  - Denote child box extents by \( a'_{x}, a'_{y}, \ldots \)

• Re-stated goal: estimate

\[ o_{ij} := P[ A_i \cap B_j \mid A \cap B \land o_x > 0 ] \]
Assumptions for now:

- **BV diminishing factor**: \( a'_x = \alpha_x a_x, \quad a'_y = \alpha_y a_y, \) etc.
- BVs on same scale, i.e.: \( b_x \approx a_x, \quad b'_x \approx a'_x, \) etc.

Look at \( p_{11} \) first:

- Preconditions:
  - \( x \)-overlap \( \alpha_x > 0 \), and
  - parent boxes overlap

- Probability:
  \[
  p_{11} = \frac{\text{area}(L')}{\text{area}(L)} = \ldots = \alpha y \alpha z.
  \]

Good news:

\[
 p_{22} = p_{12} = p_{21} = \alpha y \alpha z
\]

By analogous reasoning, we get:

\[
 P[\alpha_x^{(l)} > 0 \mid A^{(l-1)} \cap B^{(l-1)}] \approx \alpha_x
\]
Plug all this into Equation (1):

\[ \overline{N}_v(n) \leq \sum_{i=1}^{d} (4\alpha_x\alpha_y\alpha_z)^i \in \mathcal{O}(n^{\log(4\alpha_x\alpha_y\alpha_z)}) \]

Effect of diminishing factor \( \alpha \):

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( T(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>( \mathcal{O}(\log n) )</td>
</tr>
<tr>
<td>( \approx 0.35 )</td>
<td>( \mathcal{O}(\sqrt{n}) )</td>
</tr>
<tr>
<td>3/4</td>
<td>( \mathcal{O}(n^{1.58}) )</td>
</tr>
</tbody>
</table>

Experiments

- Experiment:
  - Construct simple AABB over CAD objects
  - Count number of nodes in BVTT visited by simultaneous traversal

![Diagram showing root BV overlap vs. n]
- Experiments utilizing artificial BVHs:
  - Controlled "layout" and diminishing factor $\alpha$
  - Experimental versus theoretical estimates:

- Interdependence between $\alpha$ and root BV overlap $\delta$:
  
  $\alpha = 0.6$  
  $\alpha = 0.7$
Application

- Time-critical collision detection
- Probabilistic collision detection:
  - Store average $\alpha$ at root of every sub-tree
  - Estimate # BV overlap tests using our model
  - Prioritize traversal based on this number

Conclusions

- Average-case analysis of simultaneous traversal of AABB trees
- New model to estimate the average running time
- Experiments to support correctness of our model
Future Work

- Improve model:
  - variable BV diminishing factor (probably easy)
  - integrate root BV overlap into model
- Consider other BV types (possibly hard)
- Utilize for probabilistic collision detection
- Derive method for average-case analysis of running time for concrete BVHs

Acknowledgements

- Jan Klein, MeVis, Bremen, Germany (formerly PhD student with Paderborn University, Germany)
- René Weller, PhD student, TU Clausthal
- DFG grant ZA 292/1-1 ("Aktionsplan Informatik")
- Anonymous reviewer