Minimal Hierarchical Collision Detection

Gabriel Zachmann
Dept Computer Graphics and Virtual Reality
University Bonn

Motivation

- Fundamental operation:
  - Virtual prototyping
  - Rigid bodies
  - Interaction in VR
  - Haptic rendering

General requirements

- No assumptions about input
- No assumption about motion
- Complexity: >10,000 polygons / object
- Witness(es)
- Construction of aux data structures not too slow
- Small memory footprint

Related Work

- R*-trees [Beckmann, Kriegel, et al., 1990]
- Sphere trees [Hubbard, 1996]
- OBB trees [Gottschalk, et al., 1996]
- DOP trees [Zachmann, 1998; Klosowski, et al., 1998]
- BOXTREE [Barequet, at al., 1996]

Setting

Hierarchical traversal scheme:

```plaintext
traverse(X, Y)
if X,Y do not overlap then return
if X,Y are leaves then check polygons
else for all children pairs do
  traverse(Xi, Yj)
```

Restricted Boxtrees

- Observation: child boxes fit fairly tightly into parent box on most sides

```plaintext
empty space
child
parent
```
Introduction

Restricted Boxtrees

Optimization

Construction

Results

Conclusion

**Definition**

- Combination of k-d tree and AABB:
  - Storage: 1 float, 1 axis ID, 1 pointer

**Overlap Tests**

1. Re-alignment:
   - 12 FLOPs
2. SAT:
   - 82 FLOPs
3. SAT lite:
   - 24 FLOPs
4. Sphere test:
   - 29 FLOPs

**General Optimization**

- Factorization of overlap test costs:
  - \( c_1 = \text{node-specific} \)
  - \( c_2 = \text{pair-specific} \)

  - Brute-force:
    \[
    C(X, Y) = 2c_1 + c_2 + 4(2c_1 + c_2) = 10c_1 + 5c_2
    \]

  - Eager computation:
    \[
    C(X, Y) = 0c_1 + c_2 + 2c_1 + 4c_2 = 2c_1 + 5c_2
    \]

- E.g.: 1.5 mult + 2 add + 5 comp vs. 12 FLOPs

**Constructing Restricted Boxtrees**

- Approach: top-down
  1. Compute BV covering input
  2. Split input into two subsets

  - Splitting criterion:
    - Expected traversal cost:
      \[
      \sum_{i,j} P(X_i, Y_j) C(X_i, Y_j)
      \]

- Estimation of \( P(X_i, Y_j) \):
  \[
  X_i \cap Y_j = \emptyset \iff p \in X_i \oplus Y_j
  \]

  \[
  P(X_i, Y_j) = \frac{\text{vol}(X_i \oplus Y_j)}{\text{vol}(X \oplus Y)}
  = \frac{\text{vol}(X_i \oplus Y_j)}{\text{vol}(X) + \text{vol}(Y)}
  \]

**Algorithm**

- Try three cases:
  1. Lower and upper child boxes
  2. Both lower/upper
  3. Perpendicular

- Find "good" splitting plane
- Find good "seeds"
- Split set of polygons
- Complexity:
  \[
  T(n) = cn + T(an) + T((1-a)n) \in O(n)
  \]
**Results**

- **Suite:**
- **Platform:**
  - Pentium III, 1GHz
- **Construction:**

**Summary**

- New hierarchical BV structure (*Restricted Boxtree*) with extremely small memory footprint (9 bytes/node)
- Very efficient overlap tests for restricted boxes (down to 8.5 FLOPs per BV pair)
- General optimization technique
- Theoretical argument for construction criterion
- Comparison with DOP trees

**Future Work**

- Other applications (ray tracing, occlusion culling, ...)
- "Look-ahead" during construction