Stochastic Methods

Laks Raghupathi
Laboratoire GRAVIR*
Grenoble, France

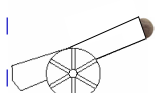
Gabriel Zachmann
Universität Bonn
Germany

*GRAVIR is a laboratory of INRIA, CNRS, UJF and INPG

Motivation

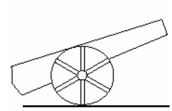
- Absolute exactness not always necessary
- Real-time more important

→ Approximate collision detection



- Whenever only qualitative result or perceived quality matters
- Games, virtual clothes prototyping, medical training, …

Set of „plausible paths" for a cannonball.



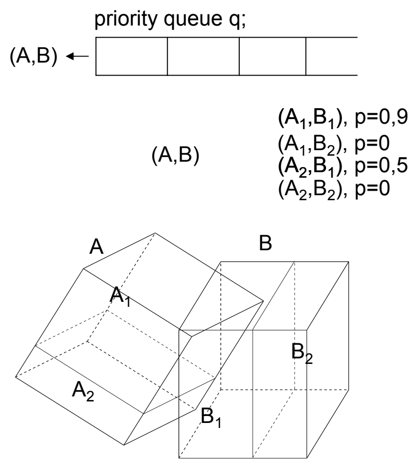
Two Techniques

1. ADB-Trees (Gabriel Zachmann)
2. Stochastic Closest Features Tracking (Laks Raghupathi)

Basic Idea

- Average-case approach:
  - Estimate probability of intersection of 2 *sets* of polygons
- Guide traversal by probabilities (p-queue)
- Can be applied to almost any BV hierarchy

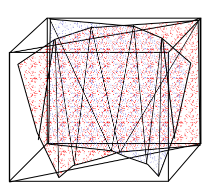
Probability-Guided BVH Traversal



Traverse(A,B)

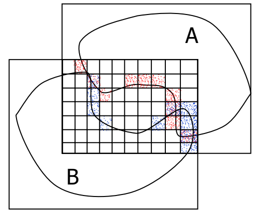
```
p-queue q
q.insert(A,B,1)
while q not empty
   A,B ← q.pop
   forall Ai, Bj
      p ← Pr[ collision in Ai, Bj ]
      if p ≥ pmin
         return "collision"
      if p ≥ 0
         q.insert(Ai, Bj, p)
return "no collision"
```


Thought Experiment ("Gedankenexperiment")



Estimation of the Probability

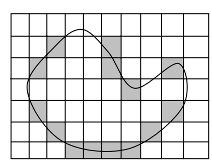
1. Partition A ∩ B by s cells
2. Compute s_A =
   #cells well-filled from A
3. Dito for s_B = #cells from B
4. c(A ∩ B) = # collision cells
5. Compute probability that c ≥ x:



$$Pr[c(A \cap B) \geq x] = 1 - \sum_{t=0}^{x-1} \frac{\binom{s_B}{t}\binom{s-s_B}{s_A-t}}{\binom{s}{s_A}}$$

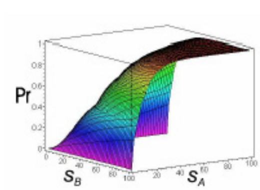
Efficient Estimate

- Preprocessing:
  - Partition each BV of BVH by grid
  - Count number s_A of well-filled cells
  - Store with each node of BVH
- At runtime estimate s_A and s_B:

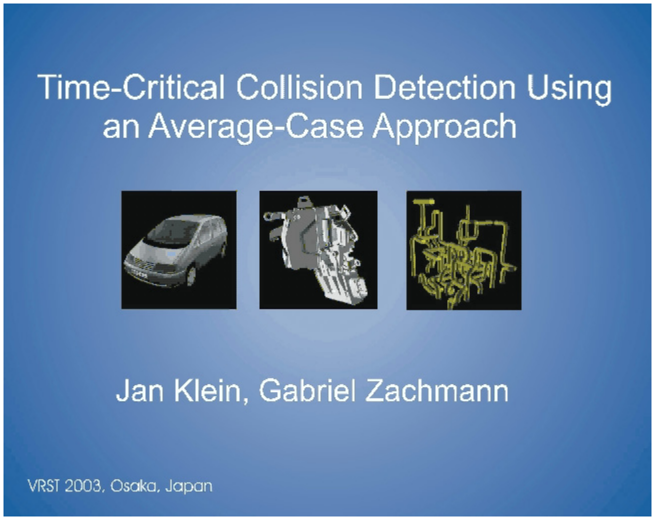


$$s'_A = s_A \frac{\mathrm{Vol}(A)}{\mathrm{Vol}(A \cap B)}$$

- Precompute LUT for function *Pr* for all possible input values



Video



Results

- Time vs. error:

60,000 polygons each