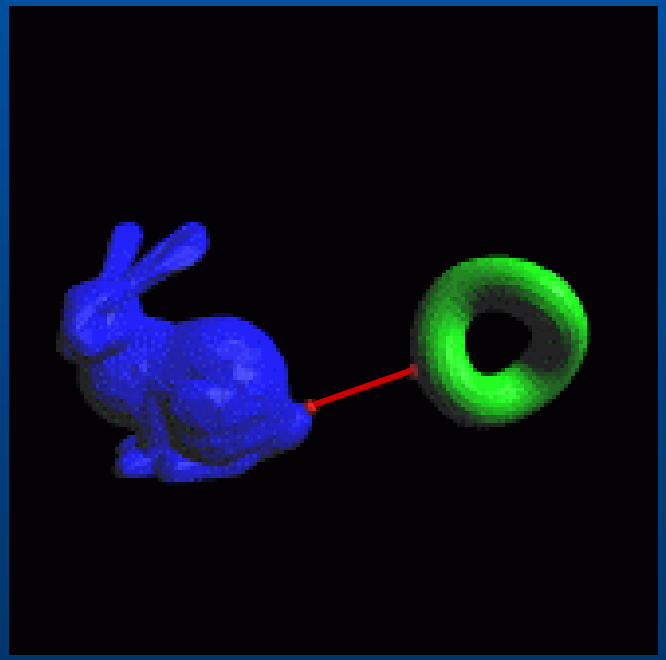
Fast Distance Queries with Rectangular Swept Sphere Volumes

Eric Larsen
Stefan Gottschalk
Ming Lin
Dinesh Manocha

UNC Chapel Hill

Problem

- **Finding translation distance between two polygonal models:**



- **Common in:**
  - **Path planning.**
  - **Virtual Prototyping.**
  - **Dynamics.**

Previous Work

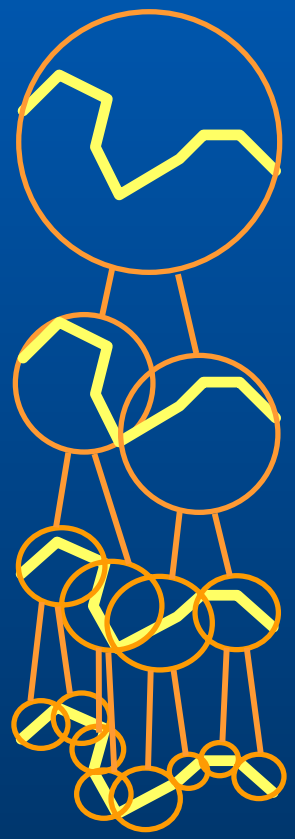
- **Computational Geometry:**
  - Dobkin & Kirkpatrick ‘82
  - Seidel ‘90
  - Chazelle ‘89
- **Distance between Convex Polyhedra:**
  - Gilbert et al. ‘88
  - Lin & Canny ‘91
  - Cameron ‘97
  - Mirtich ‘98
- **Space Partitioning:**
  - Naylor et al. ‘90
  - Bouma and Vanacek ‘91

Previous Work

- **4-dimensional Intersection:**
  - **A. Garica-Alonso et al. ‘94**
  - **Hubbard ‘93**
- **Distance between NURBs models:**
  - **Synder et al. ‘93**
  - **Cameron ‘98**

Previous Work

- **Bounding Volume Hierarchies (BVHs):**
  - **Detecting Intersection:**
    - **Convex Hulls: Lin & Canny ‘91**
    - **Spheres: Hubbard ‘93**
    - **Axis-aligned boxes: Beckmann et al. ‘90, SOLID**
    - **Oriented boxes: Gottschalk et al. ‘96, Barequet et al. ‘96**
    - **K-DOPs: Held et al. ‘96, Klosowski et al. ‘98**
    - **Spherical Shells: Krishnan et al. ‘98**
  - **Distance Computation:**
    - **Spheres: Quinlan ‘94**
    - **OBBs: Johnson & Cohen ‘98**



Bounding Volume Hierarchies

- **Data structure:**
  - Hierarchy of volumes bounding polygon subsets.
- **Algorithm:**
  - Initialize distance estimate.
  - Recursive search:
    - Compute distance between BVs.
    - If less than distance estimate:
      - Recur on pairs of children nodes.
    - When polygons reached:
      - Revise estimate.

Bounding Volume Hierarchies

- **Cost equation:**

$$\text{Total Cost} = N_{BV} \times C_{BV} + N_P \times C_P$$

N_{BV} = number of BV distance tests
C_{BV} = cost of BV distance tests
N_P = number of polygon distance tests
C_P = cost of polygon distance tests

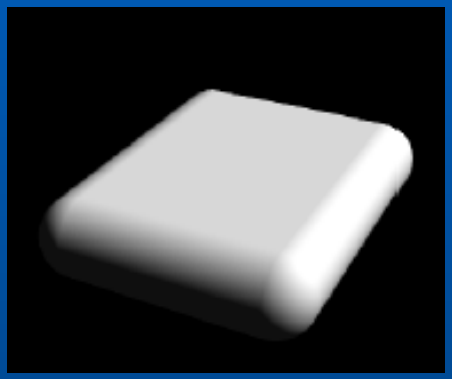
Bounding Volume Hierarchies

- **Impacts on cost equation:**
  - **BV type:**
    - **Tightness of fit.**
    - **Cost of each BV distance test.**
  - **Ways to fit BVs and build hierarchy.**
  - **Order to search BVHs.**
- **Analysis:**
  - **OBB convergence - Gottschalk et al. ‘96.**

Bounding Volume Hierarchies

- Previous variations for distance:
  - Quinlan ‘94:
    - Spheres.
    - Sets of leaf spheres per polygon:
      - maximum leaf sphere size.
  - Johnson & Cohen ‘98:
    - OBBs.
    - Breadth-first search.
- Our goals:
  - Tighter fitting BVs like OBBs.
  - Accelerating BVH searching.

BV Choice



- **Considered OBBs:**
  - **Distance calculation.**
- **Rectangular Swept Sphere:**
  - **Sphere swept across 3D rectangle.**
  - **Distance calculation:**
    - **Distance between 3D rectangles, minus sum of sphere radii.**

BV Choice

- **Rectangle Distance Calculation:**
  - **Specialized approach:**
    - **Voronoi methods, Lin-Canny '91**
    - **Separating axes, Gottschalk et al. '96**
  - **~4 times faster than a general convex polyhedra implementation.**
  - **Conservative estimation in degenerate cases.**

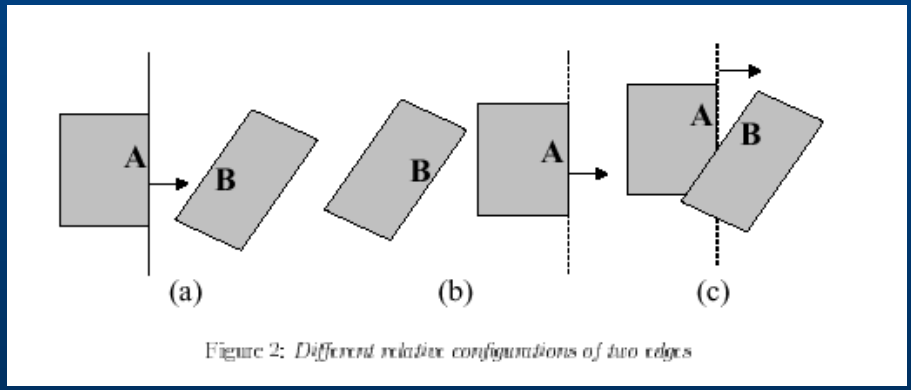


Figure 2: *Different relative configurations of two edges*

BVH Building

- Building methods from Gottschalk et al. ‘96
  - One leaf level BV per polygon
  - Top-down recursive building:
    - fit a set of polygons
    - split into subsets and recur
  - Binary hierarchy

Acceleration Techniques

- **BVH search uses a distance estimate:**
  - **Search most efficient when estimate low.**
- **Polygon Caching:**
  - **Save closest polygons in one query.**
  - **Initial distance estimate next time.**
  - **Empirical 2 times speedup for applications with coherence.**

Acceleration Techniques

- **Priority-directed Search:**
  - **Priority queue to search closest BVs first.**
  - **Queue size limited:**
    - **recursion on closest BVs when full**
  - **May reduce extent of search without coherence.**
  - **O(lg(n)) queue operations.**

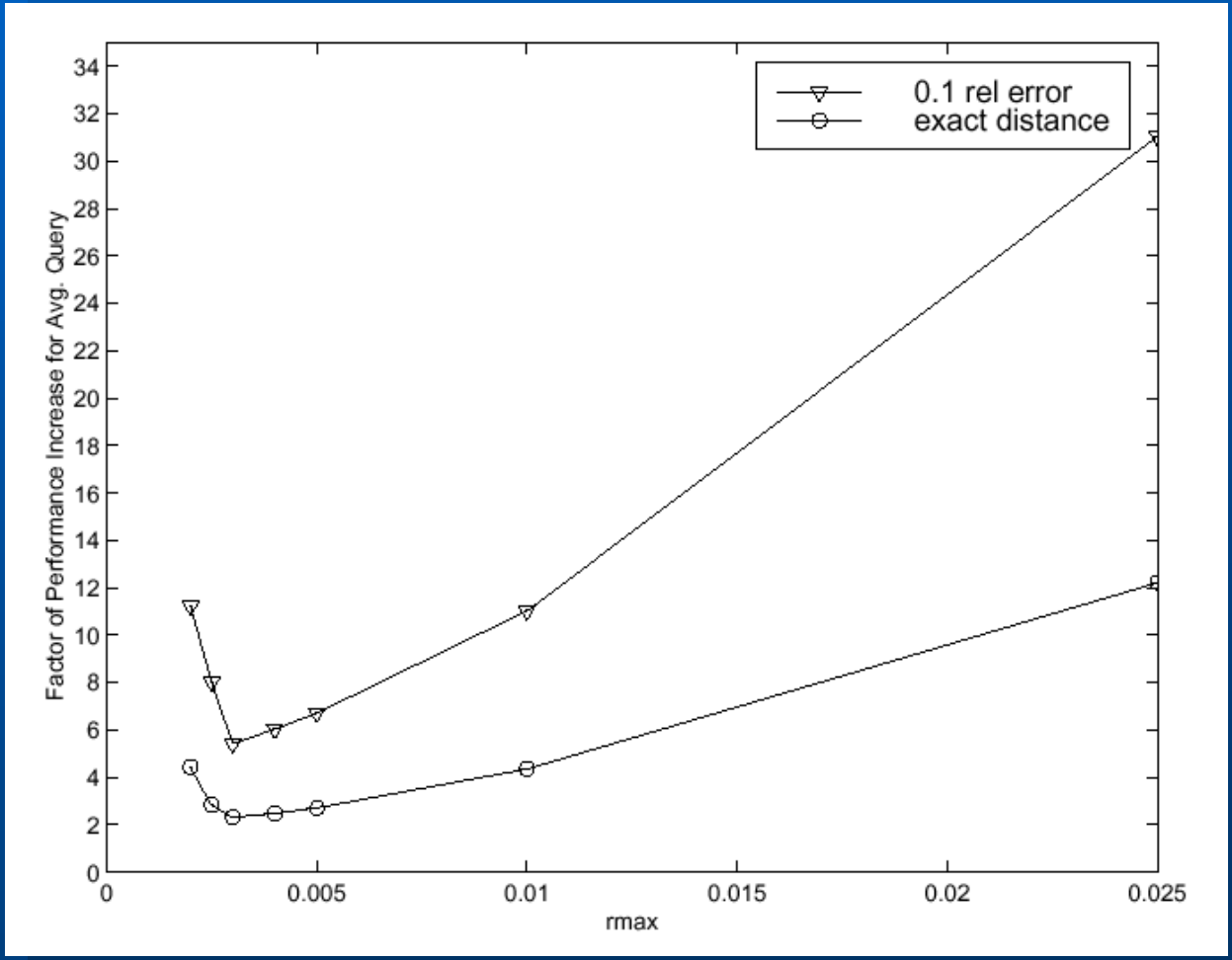
Comparison

- Compared our software to Quinlan's:
  - More than one difference:
    - Type of BV
    - Fitting one vs. many BVs to a polygon.
    - Polygon caching:
      - Quinlan's software might also benefit.

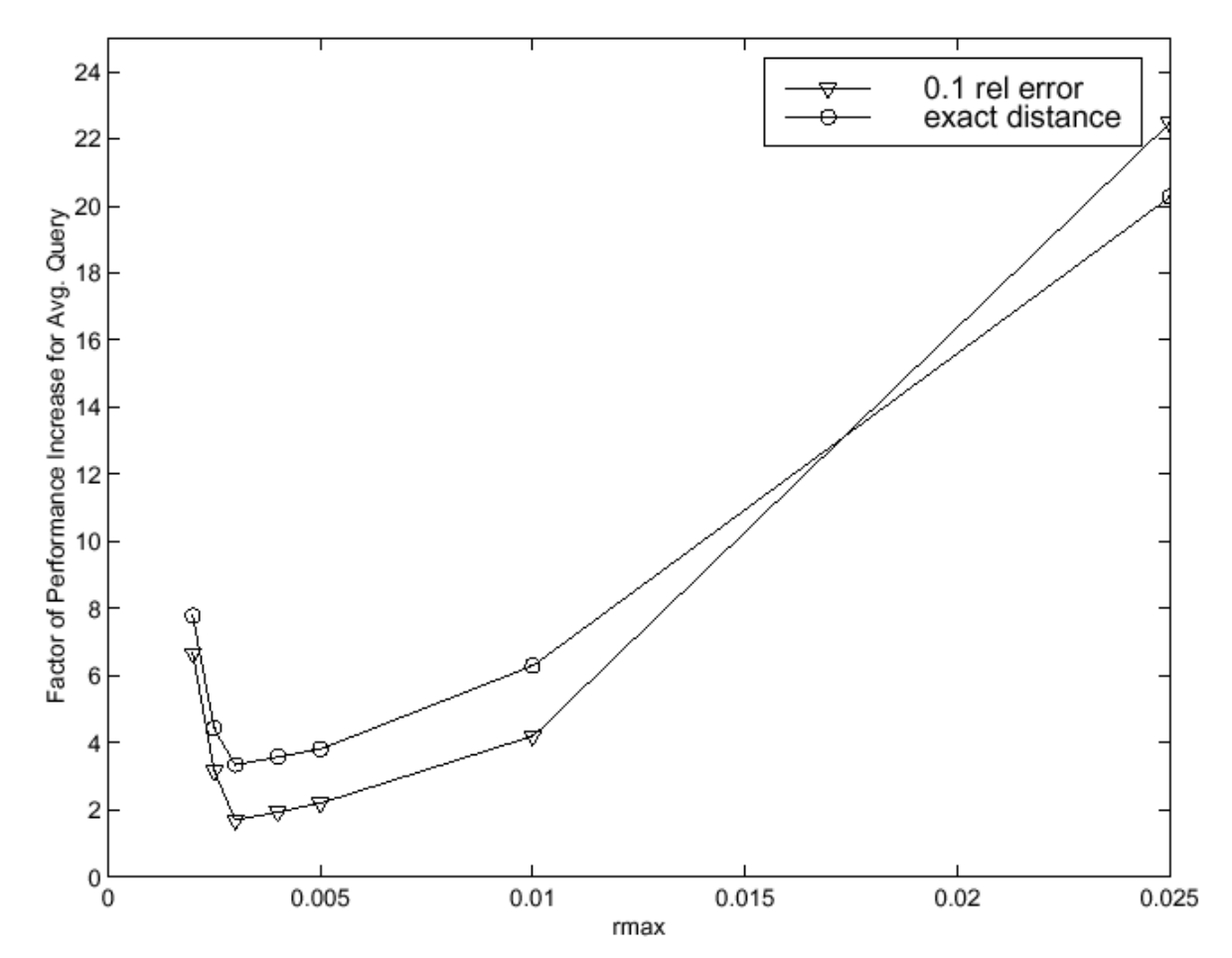
Results

- **Benchmarks:**
  - **For two planner scenarios, tried:**
    - **sequence of calls by randomized planner.**
    - **sequence of calls walking the path found.**
- **Performance:**
  - **Varied maximum leaf BV size in Quinlan's code.**
  - **Plotted relative speed of our system to Quinlan's**

Results


0.1 rel error
exact distance
Factor of Performance Increase for Avg. Query
rmax
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
0
0.005
0.01
0.015
0.02
0.025

Results


0.1 rel error
exact distance
Factor of Performance Increase for Avg. Query
rmax
0
2
4
6
8
10
12
14
16
18
20
22
24
0
0.005
0.01
0.015
0.02
0.025

Results

- **Similar performance for some leaf sizes:**
  - **Lowest speedup about 2 times.**
- **Large speedups at other leaf sizes.**
  - **Picking the wrong size has large consequences.**

Results

- **Priority directed search:**
  - **Reduced our system's performance when walking path:**
    - **Coherence present.**
    - **Polygon caching already in effect.**
  - **Sped up the path planning calls by small factor:**
    - **Low coherence.**

Considerations

- **BVH algorithms:**
  - **Disadvantages:**
    - **Individual queries vary a lot in cost.**
    - **Convex polyhedra algorithms faster.**
  - **Advantages:**
    - **General polygon input:**
      - **No restrictions on convexity or topology.**
      - **Disconnected triangles.**
      - **E.g., some CAD data.**

Software

www.cs.unc.edu/~geom/SSV/