High-performance Penetration Depth Computation for Haptic Rendering

Young J. Kim

http://graphics.ewha.ac.kr
Ewha Womans University
Issues of Interpenetration

• Position and orientation of the haptic probe, governed by the user through the haptic device

• Interpenetration in haptic simulation is unavoidable

probe

object
Penalty-based Response

• Penetration depth (PD) is required for computing penalty-based contact response
Previous Work on PD

- Convex polytopes - [Cameron and Culley86], [Dobkin93], [Agarwal00], [Bergen01], [Kim et al. 04]
- Non-convex polyhedra - [Kim02], [Redon and Lin06], [Lien08a,b], [Hachenberger09]
- Polygon soups – [Je et al. 12]
- Distance fields - [Fisher and Lin01], [Hoff02], [Sud06]
- Pointwise PD - [Tang et al. 09]
- Generalized PD – [Ong and Gilbert96], [Ong96], [Zhang07], [Tang et al. 12]
- Volumetric PD - [Wellner and Zachmann09]
Challenges

• Penetration depth (PD)
  – Is very expensive to compute accurately
  – May not handle arbitrary geometry and topology

• Current practice
  – Hacks
    → Slow, inconsistent, geometrically unstable
Goal

• Recent research results
  – Pointwise
  – Translational
  – Generalized

• Recent results on 6DoF haptic rendering
M. Tang, M. Lee, Y. J. Kim, Interactive Hausdorff Distance Computation for General Polygonal Models, SIGGRAPH 2009

Pointwise Penetration Depth
Pointwise Penetration Depth

• Defined as deepest interpenetrating points
One-sided Hausdorff Distance

\[ h(A, B) = \max_{a \in A} \left( \min_{b \in B} \|a - b\| \right) \]
One-sided Hausdorff Distance

\[ h(\mathcal{B}, \mathcal{A}) \equiv \max_{b \in \mathcal{B}} \left( \min_{a \in \mathcal{A}} \| a - b \| \right) \]
Two-sided Hausdorff Distance

$$H(A, B) \equiv \max \left( h(A, B), h(B, A) \right)$$
Shape Deviation Measure

- Hausdorff distance quantifies deviation between two geometric models.

Large Hausdorff Distance Value

Small Hausdorff Distance Value
1. Find intersection surfaces $\partial A$ and $\partial B$

2. Penetration depth $= H(\partial A, \partial B)$
Pointwise Penetration Depth

Demo (40K Bunny vs 40K Bunny)
Benchmark: Pointwise PD

Model complexity
  – 50K tri
Avg. Performance
  – 3.88ms/pair
Benchmark: Pointwise PD

Model complexity
- 3.5K tri

Avg. performance
- 0.95ms/pair

Translational Penetration Depth
(Translational) Penetration Depth
[Dobkin 93]

- Minimum translational distance to separate overlapping objects
Configuration Space

workspace

configuration space
Translational Configuration Space

workspace

configuration space

Translational C-space = Minkowski Sums
Minkowski Sum

\[ P \oplus Q = \{ p + q \mid p \in P, q \in Q \} \]
\[ P \oplus -Q = \{ p - q \mid p \in P, q \in Q \} \]
Example

Video credit: D. Halperin
PD VS Minkowski Sum

Penetration Depth

$$P \varoplus -Q$$
Combinatorial Explosion

- Complexity of Minkowski Sum
  \[ O(m^3n^3) \] with \( m \) and \( n \) triangles
PD Estimation

Penetration Depth

Boundary of Minkowski Sums
Out-Projection = Continuous Collision Detection

Out-Projection

$q^f$

$q_0$

0

Boundary of Minkowski Sums
Continuous Collision Detection

• Source codes are available
  – http://graphics.ewha.ac.kr/FAST (2-manifold)
  – http://graphics.ewha.ac.kr/C2A (polygon-soups)
  – http://graphics.ewha.ac.kr/CATCH (articulated)
  – http://graphics.ewha.ac.kr/CCQ (for motion planning)
In-Projection $\equiv$ LCP
(Linear Complementarity Problem)
PolyDepth: Iterative Optimization

Out-Projection

In-Projection

Penetration Depth

Boundary of Minkowski Sums
PolyDepth Performance

- Spoon: 1.3K triangles
- Cup: 8.4K triangles
- Time: 1~7 msec
PolyDepth Performance

- Bunny: 40K triangles
- Dragon: 174K triangles
- Time: 2~15 msec
Comparison against Exact Solution

Accuracy

Performance

Generalized Penetration Depth
Generalized Penetration Depth

- Minimal rigid motion to separate overlapping objects
Definition of Generalized PD

• Defined in 6D configuration space

\[ PD^\sigma_g (A, B) = \left\{ \min \left\{ \sigma_A (q, o) \right\} \left| \text{interior} (A(q)) \cap B = \emptyset, q \in F \right. \right\} \]
Distance metric

- Object norm
  - The average squared displacement

\[ \sigma_A(q_0, q_1) = \frac{1}{V} \int_{x \in A} (x(q_0) - x(q_1))^2 \]
PolyDepth++ Algorithm

1. Free-configuration selection
PolyDepth++ Algorithm

2. Contact-space projection

Contact Space
PolyDepth++ Algorithm

3. Constrained optimization

Contact Space

$q_0, q_1$

LLCS
PolyDepth++ Algorithm

4. Re-projection

Contact Space

$q_0, q_1, q_2$
PolyDepth++ Algorithm

5. Iteration until finding a locally-optimal solution
PolyDepth++ for Articulated Model

- Object norm for a link

\[
\sigma_i = \frac{1}{V} \int_{x \in L_i} \left( x(q_i') - x(q_i) \right)^2 dx
\]

\[
\sigma_A(q, q') = \sum_{i=0}^{n-1} \sigma_i
\]
PolyDepth++ for Articulated Model

• Constrained optimization in higher dimension

Minimize $\sigma(q) = \sum_{i=0}^{n-1} \sigma_i$

subject to: $C(q - q_c) \geq 0$

$m \times |q|$
Generalized PD Performance

Generalized PD for Rigid Body
Software Implementations

• Source codes are available
  
  – http://graphics.ewha.ac.kr/polydepth  
    (translational PD)

  – http://graphics.ewha.ac.kr/hdist  
    (Hausdorff distance and pointwise PD)

HAPTIC APPLICATIONS
Penalty-based Haptic Rendering using Translational and Generalized PD

Translational PD

Generalized PD
Penalty-based Haptic Rendering using Translational and Generalized PD

Translational PD

Generalized PD
Benchmarks Setup

6DoF PHANToM Premium 1.5
Performance

For More Details, This Tues, 1:10PM Oral Session V
Summary

• Pointwise PD
• Translational PD
• Generalized PD

• 6DoF haptic rendering with translational and generalized PD
Future Work

• Parallel haptic rendering
  – Asynchronous contact handling
  – GPU-based parallelization

• Haptic rendering for
  – Articulated models
  – Massive models
High Performance GPU-based Collision Queries

5. Results

HW: Intel Quad-core 2.66GHZ CPU
4.0GB memory
NVIDIA Geforce GTX580
SW : Windows 7 & VS 2008
CUDA 4.0

Real-time Collision Culling of a Million Bodies on GPUs
ACM Trans on Graphics 2010

Real-time Adaptive Signed Distance Fields for Rigid and Deformable Models on GPUs
Acknowledgements

• Min Tang, Xinyu Zhang, Minkyoung Lee, Yi Li (Ewha)
• Fuchang Liu (NTU)
• Changsoo Je (Sogang)

• KEIT/MKE (IT core research)
• NRF
Thank you for listening!

http://graphics.ewha.ac.kr