Real-Time Collision Detection for Dynamic Virtual Environments

Distance Fields

Arnulph Fuhrmann

Fraunhofer Institute for Computer Graphics

Darmstadt, Germany

Outline

• Introduction
• Distance Field Generation
• Collision Detection using Distance Fields
• Conclusion

Introduction

• Physically based modeling
  – Cloth, hair, etc.
• Problem
  – Many contact points
• During Simulation
  – Detect Collision
  – Compute Collision Response
    • Proximity or penetration depth
    • Surface normal

Distance Field Definition

• Scalar function
  \[ D : \mathbb{R}^3 \rightarrow \mathbb{R} \]
  \[ \text{dist}(p) = \text{distance to closest point on surface} \]
  \[ \text{sign}(p) = \text{negative if inside object} \]
  \[ D(p) = \text{sign}(p) \cdot \text{dist}(p) \]
Outline

• Introduction
• Distance Field Generation
  • Collision Detection using Distance Fields
• Conclusion

Distance Field Data Structures

• Uniform 3D grid
  – Queries take $O(1)$ time
  – Curved surfaces can be represented quite well
  – $C^0$ continuous
• Adaptively sampled distance fields (ADFs)
  – Frisken et al. '00
  – $C^{-1}$ between different levels
  – Can be resolved

Distance Field Data Structures

• BSP-tree
  – Wu and Kobbelt '03
  – Piecewise linear approximation
  – Generation computationally expensive
  – Discontinuities between cells
  – Compact!
Distance Field Data Structures

- Sparse Block Grids
  - [Bridson '03]
  - Distance values needed only for a small band

- Divide the uniform grid into blocks
  - Coarse grid contains pointers to fine sub-grids
  - Not all sub-grids exist

- Queries (in comparison to uniform grids)
  - More complex
  - Less efficient

Sparse Block Grid Example

Sparse Block Grid – Memory Savings

- Uniform Grid
  - Resolution 378x396x81
  - 48.5 MB

- Sparse Block Grid
  - Same resolution
  - 3x3x3 sub-grids
  - 6.7 MB
  - 86% memory savings

Computation of Distance Fields

- Object representation
  - Triangular mesh

- Problem
  - Computing distances for all grid points
  - Naïve computation too costly

- Collision detection
  - Only a small band needed
Computation of Distance Fields

- Propagation methods
  - Fast Marching methods [Sethian '96]
  - Distance Transforms [Jones and Satherley '01]
- Rasterizing of distance functions
  - Full distance field
    - [Sud et al. '04], [Hoff et al. '99]
  - Bounded Voronoi Regions
    - [Sigg et al. '03], [Breen et al. '01]
    - Bounding polyhedron around Voronoi regions of edges, faces, and vertices

Outline

- Introduction
- Distance Field Generation
- Collision Detection using Distance Fields
- Conclusion

Collision Detection

- [Fuhrmann et al. '03]
- Scenario
  - Deformable object A
  - Static object B
- Collision Detection
  - Sample object A
  - Test sample points for collision with B
- If both objects are deformable
  - Swap and repeat
Collision Detection

- Problem
  - Edges intersect object

- Solution
  - Preserve ε distance at vertices

Queries needed for collision detection

(On a uniform or sparse grid)

- Distance
  - Tri-linear interpolation

- Normal
  - Direction given by the gradient

What about deforming collision objects?

- Multiple distance fields
- Linked rigid objects
  - One distance field per object
- Not possible yet
  - Soft objects like a bending human arm

Other approaches for deforming objects

- [Bridson et al. '03]
  - Clothing and animated characters
  - Pre-computed ADIFs for the body parts
  - Can be used for several cloth simulations

- [Fisher and Lin '01]
  - Deforming geometries
  - Collision detection is done hierarchically
  - Partial DF updates only
  - Internal distance fields for collision response

[Fisher and Lin '01]
Demo Video

- Captured directly from screen
- Simulation runs in java 1.4.1
- Rendering with OpenGL
- Tests made on a Intel Processor at 2.8 GHz
- Buddha model consist of 100k triangles

Outline

- Introduction
- Distance Field Generation
- Collision Detection using Distance Fields
- Conclusion

Summary

- Distance Fields Generation
  - Pre-Processing step
  - Duration: Some seconds
- Collision Detection using Distance Fields
  - Most useful for deformable against rigid objects
  - Efficient computation of
    - Penetration depth / proximity
    - Gradient (Normal)
  - Easy to implement
  - Robust algorithm
Thank You!

Arnulph Fuhrmann  //  afuhr@igd.fhg.de

---

- Frisken et al. '00 Frisken, S., Perry, R., Rockwood, A., and Jones, T. R. 
  Adaptively Sampled Distance Field. SIGGRAPH 2000, pages 249–254.

- Wu and Kobbelt '03 Wu, J., and Kobbelt, L. 

- Bridson '03 Bridson, R. 

- Sethian '96 Sethian, J. 

- Jones and Satherley '01 Jones, M. W., and Satherley, R. 

- Sud et al. '04 Sud, A., Otaduy, M. A., and Manocha, D. 

- Hoff et al. '99 Hoff, K. E., Manocha, D., and Culver, T. 

- Bridson et al. '03 Bridson, R., Paciorek, S., and Fedkiw, R. 

- Breen et al. '01 Breen, D. E., Mauch, S., Whitaker, R., and Mao, J. 

- Sigg et al. '03 Sigg, C., Peikert, R., and Gross, M. 

- Bridson et al. '03 Bridson, R., and Fedkiw, R. 

- Breen et al. '03 Breen, D. E., Mauch, S., and Whitaker, R. 