Geometric Data Structures for Computer Graphics

Dr. Gabriel Zachmann
Dr. Elmar Langetepe
University Bonn
Germany
{zach,langetep}@cs.uni-bonn.de
http://cg.cs.uni-bonn.de/course/

Introduction
• What this tutorial is about
• What it is not about

Overview
1. Quadtrees
2. Bounding Volume Hierarchies
3. Voronoi Diagrams & Delaunay Triangulations
4. Distance Fields
5. Generic Dynamization of Geometric Data Structures

Ray Shooting
• Applications: ray tracing, radiosity, volume visualization, terrain following, etc.
• Simplest solution: grid
• 3D octree
  – Bottom-up
  – Top-down

5D Octree for Rays
• Ray = point + dir. = 5-dim. object
• Octree over rays
  – "Direction cube"
  – One-to-one mapping for dir’s:
    \[ D := [-1,+1] \times \{ \pm x, \pm y, \pm z \} \]
  – All rays in universe \[ R = U \times D, \quad U = [0,1]^3 \]
• Node of 5D octree = beam in 3D:

Construction
• Associate object with node iff obj intersects beam
• Start with root node = \[ U \times [-1,+1]^2 \]
  and all objects in U
• Partition node iff ...
  – too many objects, and
  – cell too large.
  – Distribute set of objects over sub-nodes
Shooting rays

- Algorithm is trivial now:
  1. Convert ray to 5D point
  2. Find leaf of octree
  3. Intersect ray with associated objects
- Optimizations ...

Texture Synthesis

- Properties of textures
  - Stationary under moving window
  - Locality of dependence
- Algorithm
  \[
  \text{for all } p \in \text{new image} \text{ do} \\
  \quad \text{find } p' \in \text{old image so that} \\
  \quad \|N(p) - N(p')\| = \min \\
  \quad \text{set } p := p'
  \]

Examples

Interactive Demo

Other Octree Applications

- Terrain Visualization
- Isosurface Generation
- ...

Bounding Volume Hierarchies

- Definition (informal):
  - Tree, nodes carry BV
  - Leaves carry one (or more) “primitives”
  - BV of node contains BVs of all children
  - Leaf BV contains primitive
- Many variable features
  - Bounding volumes
  - Tightness vs. computational expense
Applications

- Ray shooting
- Nearest-neighbor
- Frustum and occlusion culling
- Geographical data bases
- Collision detection
- ...

Collision Detection

- Simultaneous traversal:
  
  ```plaintext
  traverse(A, B)
  if A, B do not overlap then
    return
  if A and B are leaves then
    check primitives
  else
    forall children Aᵢ, Bⱼ do
    traverse(Aᵢ, Bⱼ)
  ```

Construction

- 3 different strategies
- Most popular: top-down
- Differ by splitting criterion
- Criterion depends on app

Splitting Criterion for Collision Detection

- Expected traversal cost:
  
  $$C(X,Y) = 4(1 + Pr(X_i,Y_j) + \ldots + Pr(X_i',Y_j'))$$
  
  where
  
  $$Pr(X_i,Y_j) = Pr[BV pair (X_i,Y_j) intersects]$$

Movies

- Remaining primitives [Zachmann]
- A simple application [Zachmann]
**Reconstruction using Voronoi diagrams**

- Problem: convert unordered point cloud into 2-manifold triangle mesh ("connect the dots")

**Observation (in 2D)**

- Voronoi diagram intuitively(!) tells how to connect the points
- Sampling must be "dense enough"

**Co-Cone Algorithm**

1. Compute Voronoi diagram
2. Estimate normal \( n \) in \( p \)
   - Choose Voronoi vertex farthest from \( p \) ("pole")
   - Line through \( p \) and pole
3. Construct "co-cone" for each Voronoi site \( p \)
   - Choose suitable angle \( \alpha \)
   - "Co-cone" = cone complement
   - Shows "neighbor" points
4. For each Voronoi edge:
   - Mark edge if intersection with all 3 incident co-cones
5. For each marked edge:
   - Connect Voronoi sites (= points from input) of Voronoi regions incident to that edge

**Delaunay Triangulation on Surfaces**

- Needed for:
  - NURBS tesselation
  - BEM
  - Computer graphics
- Nice properties
**Delaunay condition on surface?**

- Use geodesic distances on surface
- Equivalent to: local evolution in the plane

**Algorithm**

1. Delaunay triangulation in parameter space
2. Evaluate NURBS at Delaunay vertices
3. Improve mesh by adding Steiner points
4. Check Delaunay condition for each edge in 3D
5. Make edge flips, until Delaunay triangulation

**Distance Fields**

- Extension of Voronoi diagrams
- Definition:
  - Surface in 3D = \( S \subseteq \mathbb{R}^3 \)
  - Distance field = function \( D_S : \mathbb{R}^3 \rightarrow \mathbb{R} \)
  - \( D_S(p) = \text{sgn}(p) \cdot \min \{ d(p,q) \mid q \in S \} \)
  - where \( \text{sgn}(p) = \begin{cases} -1 & \text{if } p \text{ inside} \\ +1 & \text{if } p \text{ outside} \end{cases} \)
- Vector distance field

**Examples**

- Simple distance field
- Vector distance field

**Relation to**

- Voronoi diagrams
- Implicit functions

**Other names**

- Potential field
- Distance maps

**Construction**

- Chamfer method:
  - Start with binary, voxelized DF (0 or \( \infty \))
  - Different computation
  - \( D(x, y, z) = \min_{i, j, k} \{ D(i, y + j, z + k) + d_v(i, j, k) \} \)
  - Approximate by convolution in several “scans”:

1. Forward
2. Backward
• Projection of distance functions
  – For each slice, and each feature of polygonal surface
  – Construct shape of distance function
  – Render polygonal approximation
  – Z-Buffer performs Minimum operation

Applications
• Morphing
• Modeling
• Cloth simulation
• Offset surface construction
• …

Morphing
Cohen-Or, Levin, Solomon

Cloth Simulation
Fraunhofer IGD Darmstadt, Arnulph Fuhrmann

Hair Simulation
University Bonn

The Design Space of Geometric Data Structures
Thanks Folks

Enjoy Siggraph!