

# Geometric Data Structures for Computer Graphics

Dr. Gabriel Zachmann  
Dr. Elmar Langetepe

University Bonn  
Germany

{zach, langetepe}@cs.uni-bonn.de  
<http://cg.cs.uni-bonn.de/course/>

## Introduction

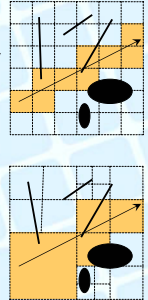
- What this tutorial is about
- What it is *not* about

## Overview

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

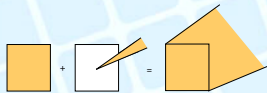
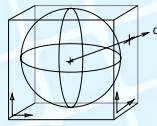
## Ray Shooting

- 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:  
 $D := [-1, +1]^2 \times \{\pm x, \pm y, \pm z\}$
  - All rays in universe  $R = U \times D$ ,  $U = [0, 1]^3$
- Node of 5D octree = beam in 3D:

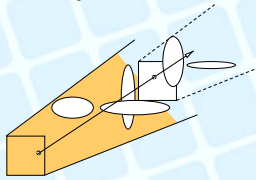


## Construction

- 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 ...



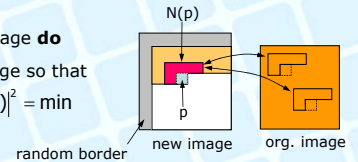
## Texture Synthesis

- Properties of textures
  - Stationary under moving window
  - Locality of dependence



- Algorithm

for all  $p \in$  new image do  
 find  $p' \in$  old image so that  
 $|N(p) - N(p')|^2 = \min$   
 set  $p := p'$

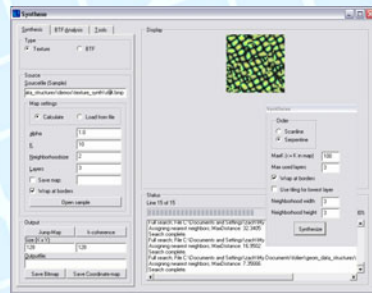


## Examples



Wei & Levoy

## Interactive Demo



University Bonn

## 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

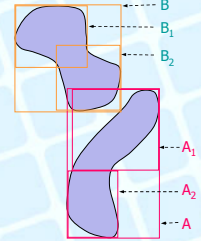
- Ray shooting
- Nearest-neighbor
- Frustum and occlusion culling
- Geographical data bases
- Collision detection
- ...

## Collision Detection

- Simultaneous traversal:
 

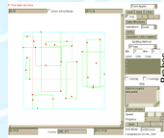
```

            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



Interactive hierarchy construction

## Splitting Criterion for Collision Detection

- Expected traversal cost:

$$C(X, Y) \approx 4(1 + \Pr(X_1, Y_1) + \dots + \Pr(X_2, Y_2))$$

where

$$\Pr(X_i, Y_j) = \Pr[\text{BV pair } (X_i, Y_j) \text{ intersects}]$$

- Reformulate intersection condition:

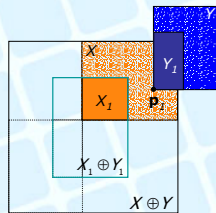
$$X_i \cap Y_j \neq \emptyset \Leftrightarrow \mathbf{p}_i \in X_i \oplus Y_j$$

- Minimize probability:

$$\Pr(X_i, Y_j) = \frac{\text{vol}(X_i \oplus Y_j)}{\text{vol}(X \oplus Y)} = \min$$

$$\Leftrightarrow \text{vol}(X_i) = \min$$

- Splitting criterion: minimize volume



## Movies



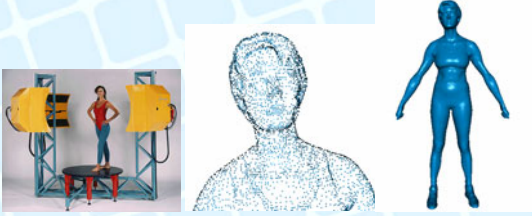
Remaining primitives [Zachmann]



A simple application [Zachmann]

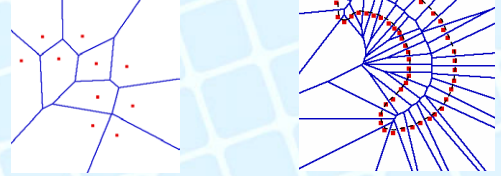
## Reconstruction using Voronoi diagrams

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



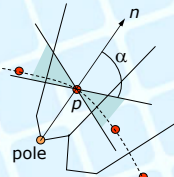
## Observation (in 2D)

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



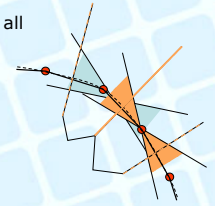
## Co-Cone Algorithm

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

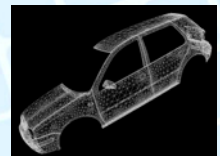
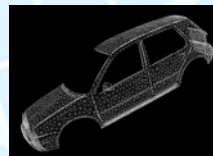
5. For each marked edge:
  - Connect Voronoi sites (= points from input) of Voronoi regions incident to that edge



- Nice about this algorithm:
  - Reconstruction is topologically equivalent to original surface
- More details:
  - Check sampling density
  - Manifold extraction
  - Normal filtering
  - ...

## Delaunay Triangulation on Surfaces

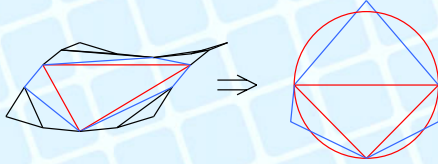
- Needed for:
  - NURBS tessellation
  - BEM
  - Computer graphics
- Nice properties





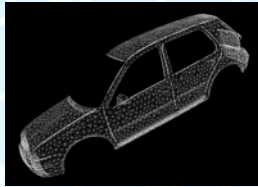
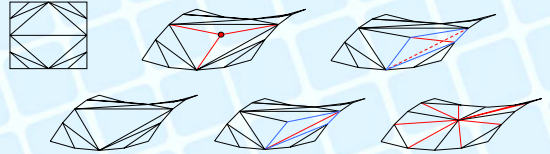
## Delaunay condition on surface?

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

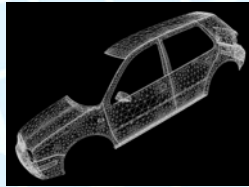


## Algorithm

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



Side lengths optimized



Angles optimized

Unit Bonn

## Distance Fields

- Extension of Voronoi diagrams
- Definition:

– Surface in 3D =  $S \subset \mathbb{R}^3$

– 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

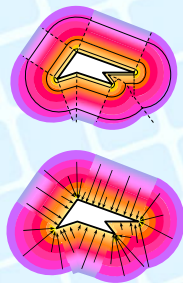
- Simple distance field
- Vector distance field

## Relation to

- Voronoi diagrams
- Implicit functions

## Other names

- Potential field
- Distance maps



## Construction

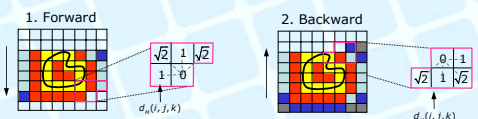
### Chamfer method:

– Start with binary, voxelized DF (0 or  $\infty$ )

– Different computation

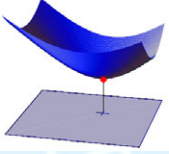
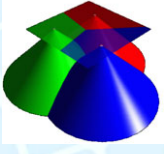
$$D(x, y, z) = \min_{i,j,k \in \mathbb{Z}} \{D(x+i, y+j, z+k) + d_m(i, j, k)\}$$

– Approximate by convolution in several "scans":



SIGGRAPH 2003  
SAN DIEGO

- 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

Introduction Quadrees Bounding Volume Hierarchies Voronoi/Delaunay Distance Fields Dynamization 58

SIGGRAPH 2003  
SAN DIEGO

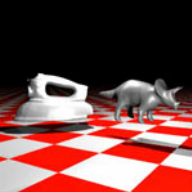
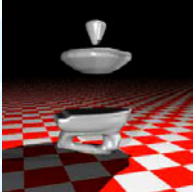
## Applications

- Morphing
- Modeling
- Cloth simulation
- Offset surface construction
- ...

Introduction Quadrees Bounding Volume Hierarchies Voronoi/Delaunay Distance Fields Dynamization 59

SIGGRAPH 2003  
SAN DIEGO

## Morphing

Cohen-Or, Levin, Solomovici

Introduction Quadrees Bounding Volume Hierarchies Voronoi/Delaunay Distance Fields Dynamization 60

SIGGRAPH 2003  
SAN DIEGO

## Cloth Simulation



Fraunhofer IGD Darmstadt, Arnulph Fuhrmann

Introduction Quadrees Bounding Volume Hierarchies Voronoi/Delaunay Distance Fields Dynamization 61

SIGGRAPH 2003  
SAN DIEGO

## Hair Simulation




University Bonn


Introduction Quadrees Bounding Volume Hierarchies Voronoi/Delaunay Distance Fields Dynamization 62

SIGGRAPH 2003  
SAN DIEGO


## The Design Space of Geometric Data Structures




Quadtree




K-d Tree




BSP Tree




BV Hierarchy




$N^3$  - Tree




Hierarchical Grid




Grid




Distance



Hierarch. Distance Field (ADF)



Distance Field



Voronoi Diagram

Introduction Quadrees Bounding Volume Hierarchies Voronoi/Delaunay Distance Fields Dynamization 63

**Thanks Folks**



Enjoy Siggraph!