Introduction to Computer Graphics
Marie-Paule Cani & Estelle Duveau

04/02 Introduction & projective rendering
11/02 Procedural modeling, Interactive modeling with parametric surfaces
25/02 Introduction to OpenGL + lab: first steps & modeling

04/03 Implicit surfaces 1 + lecture/lab: transformations & hierarchies
11/03 Implicit surfaces 2 + Lights & materials in OpenGL
18/03 Textures, aliasing + Lab: Lights & materials in OpenGL
25/03 Textures in OpenGL: lecture + lab

01/04 Procedural & kinematic animation + lab: procedural anim
08/04 Physics: particle systems + lab: physics 1
22/04 Physics: collisions, control + lab: physics 2
29/04 Animating complex objects + Realistic rendering

06/05 Talks: results of cases studies
Drawbacks of Boundary Representations

• Complex shapes with splines?
  – Branches?
  – Arbitrary topological genius?

Partly solved by subdivision surfaces

• Surrounding a volume?
  – Avoid Klein bottles!
  – Prevent self-intersections

Make them impossible?
Solution

Smooth Volume Representation

Discrete volume
Voxels

Smooth volume
Remains smooth when we zoom in
Can be converted to a mesh at any scale
**Implicit surfaces**

Defined by an *Implicit Equation*

\[ S = \{ P(x,y,z) / f(x,y,z) = iso \} \]

- \((f: \mathbb{R}^3 \rightarrow \mathbb{R})\) is the «field function»
- Surface normal : \(N = -\nabla f\)
- Characterizes a volume! \(f(x,y,z) > iso\)
  - «in/out» test (used for collisions, ray tracing…)
- Smoothness: \(S\) and \(f\) have same degree of continuity
History: Solid Geometry

Volumetric primitives \( S = \{ P(x,y,z) \mid f(x,y,z) = \text{iso} \} \)

- Spheres, ellipsoids \( \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \)
- Cylinders, cones …
- Super-ellipsoids \( \frac{x^n}{a^n} + \frac{y^n}{b^n} + \frac{z^n}{c^n} = 1 \)
Constructive Solid Geometry

Developed for Computer Aided Geometric Design (CAGD)

- Solid primitives
- Boolean operators
  - Union (or)
  - Intersection (and)
  - Difference (not)
- Construction tree

Describes the history of construction in a compact, intuitive way
Problem: limited shapes

Free form primitives?

\[ S = \{ P(x,y,z) \ / \ f(x,y,z) = iso \} \]

\( f \) polynomial (algebraic surface), or other smooth function

- What should the equation of \( f \) be to model this?
- How can a user control an implicit shape?
  - Intuitive control
  - Locality
  - Allow deformations
Idea (1982)
Blinn Objects “Blobs”

- Primitive generated by points $S$
  - $f$ decreasing function of the distance
- Union: $f = \max (f_1, f_2)$
- Intersection: $f = \min (f_1, f_2)$
- Blending: $f = f_1 + f_2$
Idea (1982)
Blinn Objects “Blobs”

- Exponential field $f_i = e^{-\frac{d(P-S_i)^2}{2}}$
  + Very smooth
  - No local control
  - Everything is to be recomputed if a point moves

- Extension to blend primitives of different sizes
  $$f_i = k_i e^{-\frac{d(P-S_i)^2}{R(S_i)^2}}$$
Make implicit surfaces local?  
(1985-1990)

Field function with compact support!
- piece-wise polynomial functions in $d(P,S_i)^2$

- **Metaballs** [Nishimura 1985]
  - if $0<d<1/3$\; $f_i = 1 - 3d^2$
  - if $1/3<d<1$\; $f_i = 3/2 \ (1-d^2)$

- **Soft Objects** [Wyvill MP W 1986]
  - if $0<d<1$\; $f_i = -4/9 \ d^6 + 17/9 \ d^4 - 22/9 \ d^2 + 1$
Choice of the field function?

- $e$ gives the thickness of an isolated primitive
- The slope affects the final shape!
- Using $(-f_i)$ instead of $f_i$ carves the shape
  - need of a flat tangent in zero

\[ \text{Distance} \]

\[ f_i \]
Extensions (1990-1995)

Skeleton-based Implicit Surfaces

Idea: Use any primitive $S_i$ as a skeleton

- $S = \{ P / \sum f_i (P) = iso \}$
- $f_i$ decreasing function of $d(P,S_i)$

Point, segments, disc, cylinder

- Intuitive control, deformation, change of topology
Extensions (1990-1995)

Skeleton-based Implicit Surfaces

J. Bloomenthal
1995
Deforming implicit primitives?

- F space deformation
  Ex: Scale, twist, bend, etc

- Deformed implicit surface

\[ f_{\text{deformed}}(P) = f \left( F^{-1}(P) \right) \]
Example of use: Blob tree

- Inspired from CSG trees
  - Blending nodes (+, -, max, min, etc)
  - Unary deformation nodes

- Used for procedural modeling
  - Description file
Displaying implicit surfaces?
Ray Tracing [Blinn 82]

- Use dichotomy to compute ray/surface intersections

Later extensions
- Analytical solutions for intersection
- Sphere tracing
  - adapt the step size based on Lipchisz constants

1980-2000: Several hours for rendering from a single view-point!
Converting implicit surfaces to meshes

**Marching cubes** [Wyvill MP W 86, Lorensen Cline87]

- Space grid
- Facetize voxels that cross the surface
- Mesh can be viewed from different viewpoints
- Extension: file to follow the surface
Converting implicit surfaces to meshes

Marching cubes [Bloomenthal 1993-1994]
- Evaluation of implicit surface tilers
- An implicit surface polygonizer (paper + code in C)
Advanced bibliography

Guaranteeing the Topology of an Implicit Surface Polygonization

[Stander Hart SIGGRAPH 1997]

• Morse theory used to track critical points
• Guaranteed correct topology!

Marching cube correct on (a) (d)
but fails on translated shapes (b) & (c)

Tracking critical points
Advanced bibliography

Adaptive Implicit Surface Polygonization Marching Triangles


• Good quality meshing of implicit surfaces
  – marching triangles, instead of marching cubes
  – Size adapted to local curvature
  – Use in an interactive modeling system
Interactive modeling with implicit surfaces?

Fast visualisation

Particules rendered as splats in the tangent plane

• *[Bloomenthal Wyvill 1991]*
  – Random particles projected along the field gradient

• *[Witking Heckbert 1994]*
  – Attraction/repulsion forces
  – Constrained to remain on the surface
  – Split/death of particles
Interactive modeling with implicit surfaces?

Fast visualisation

- [Desbrun, Tsingos, Cani 1995]
  - Sampling of primitive ‘territories’
  - Piece-wise polygonization
Fast visualization [Cani Hornus 2001]

- Overlapping territories

\[ \{ p | \forall j \neq i, f_i(p) + \eta > f_j(p) \} \]

Real-time rendering using OpenGL

- A closed polygonal mesh for each skeleton curve

Old lips  New lips
Fast visualization [Angelidis Cani 2002]

- Refinement criteria: field well reconstructed?
- Avoid cracks
Subdivision curves & surfaces as skeletons

Results [Angelidis Cani 2002]