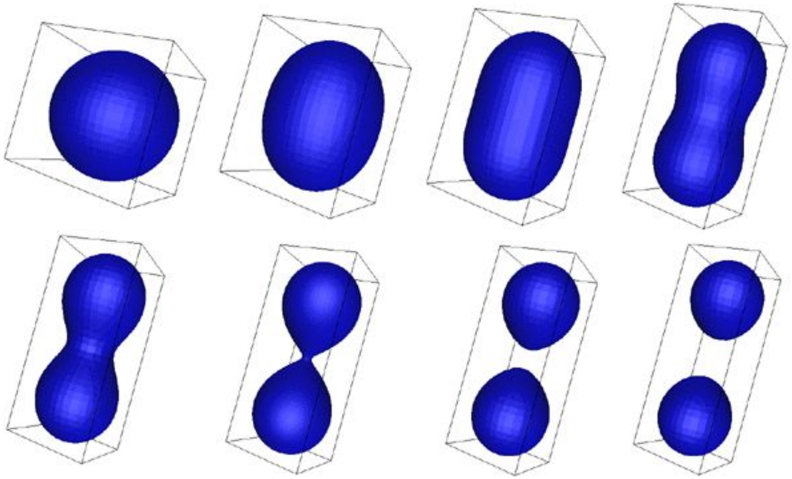
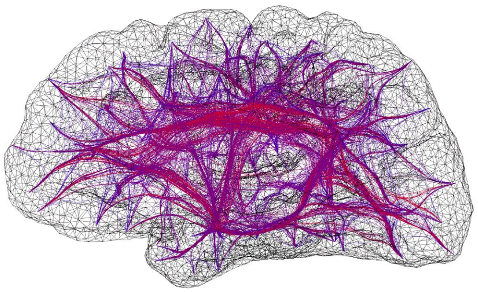
Appendix D:
Implicit surface modeling

Implicit surface modeling[(1)] is a way to produce very ‘organic’ or ‘bulbous’ surfaces very quickly without subdivision or NURBS.

Uses of implicit surface modelling:

- Organic forms and nonlinear shapes
- Scientific modeling (electron orbitals, gravity shells in space, some medical imaging)
- Muscles and joints with skin
- Rapid prototyping
- CAD/CAM solid geometry





[(1)] AKA “*metaball modeling*”, “*force functions*”, “*blobby modeling*”…

How it works

The user controls a set of *control points*; each point in space generates a field of force, which drops off as a function of distance from the point. This 3D field of forces defines an *implicit surface*: the set of all the points in space where the force field sums to a key value.

A few popular force field functions:

- "Blobby Molecules" – Jim Blinn

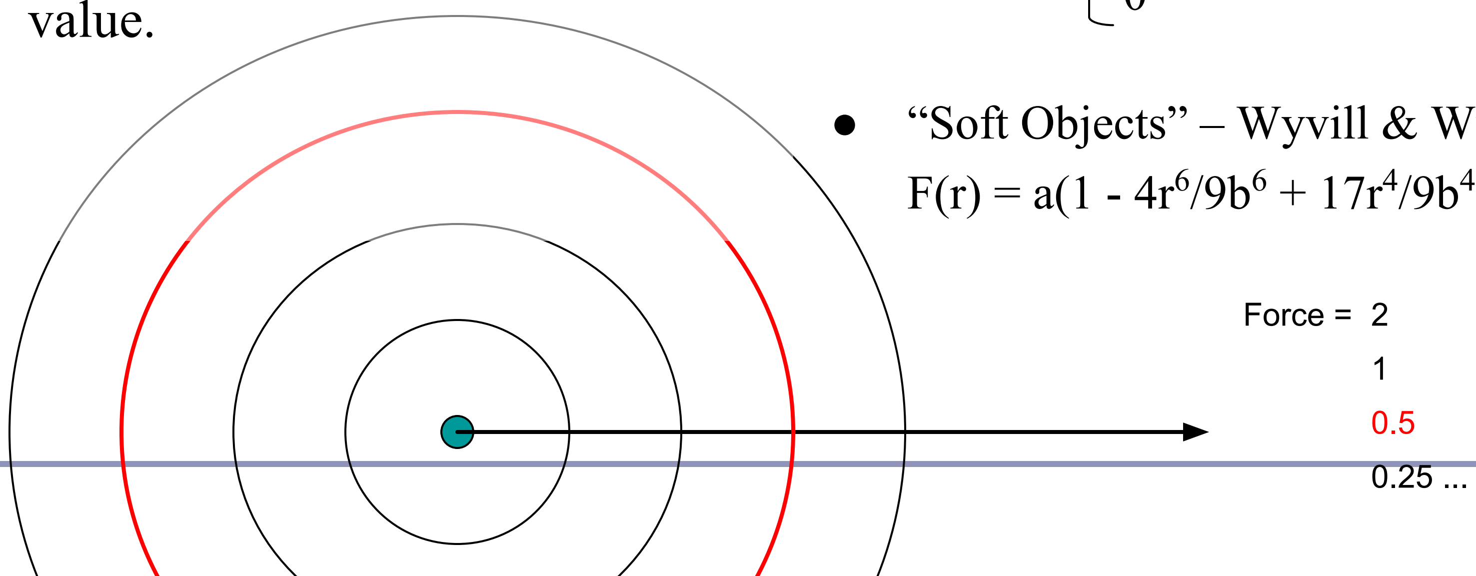
  $F(r) = a\, e^{-br^2}$

- "Metaballs" – Jim Blinn

  $$F(r) = \begin{cases} a(1 - 3r^2 / b^2) & 0 \le r < b/3 \\ (3a/2)(1 - r/b)^2 & b/3 \le r < b \\ 0 & b \le r \end{cases}$$

- "Soft Objects" – Wyvill & Wyvill

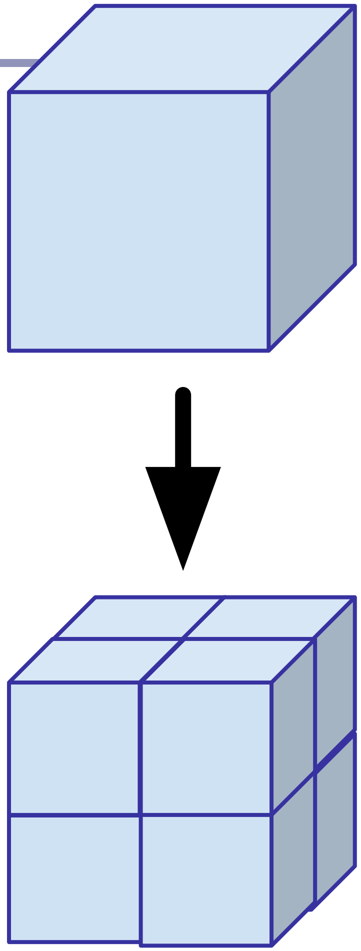
  $F(r) = a(1 - 4r^6/9b^6 + 17r^4/9b^4 - 22r^2 / 9b^2)$



Discovering the surface

An *octree* is a recursive subdivision of space which "homes in" on the surface, from larger to finer detail.

- An octree encloses a cubical volume in space. You evaluate the force function F(*v*) at each vertex *v* of the cube.
- As the octree subdivides and splits into smaller octrees, only the octrees which contain some of the surface are processed; empty octrees are discarded.



Polygonizing the surface

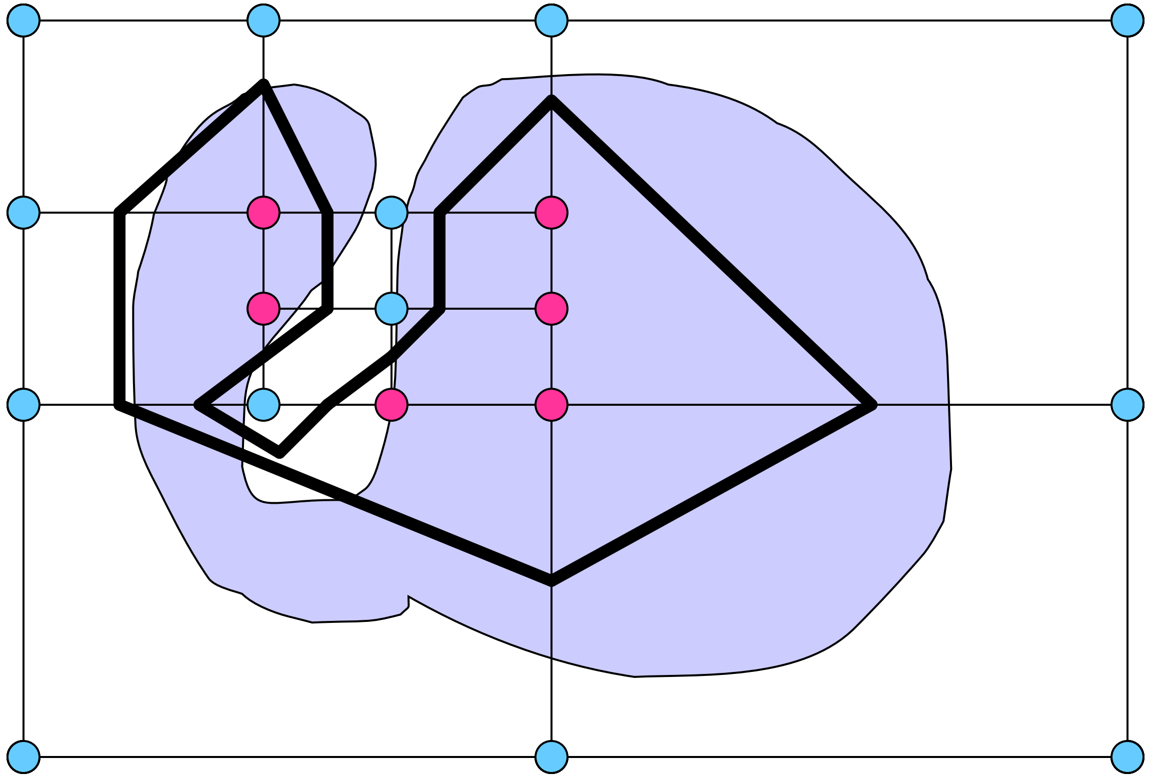
To display a set of octrees, convert the octrees into polygons.

- If some corners are "hot" (above the force limit) and others are "cold" (below the force limit) then the implicit surface crosses the cube edges in between.
- The set of midpoints of adjacent crossed edges forms one or more rings, which can be triangulated. The normal is known from the hot/cold direction on the edges.

To refine the polygonization, subdivide recursively; discard any child whose vertices are all hot or all cold.

Polygonizing the surface

Recursive subdivision (on a quadtree):

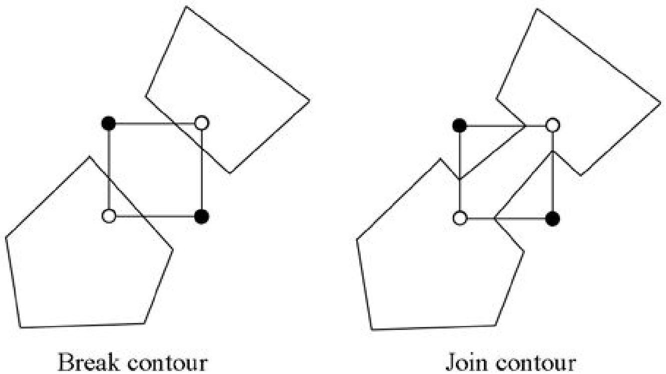


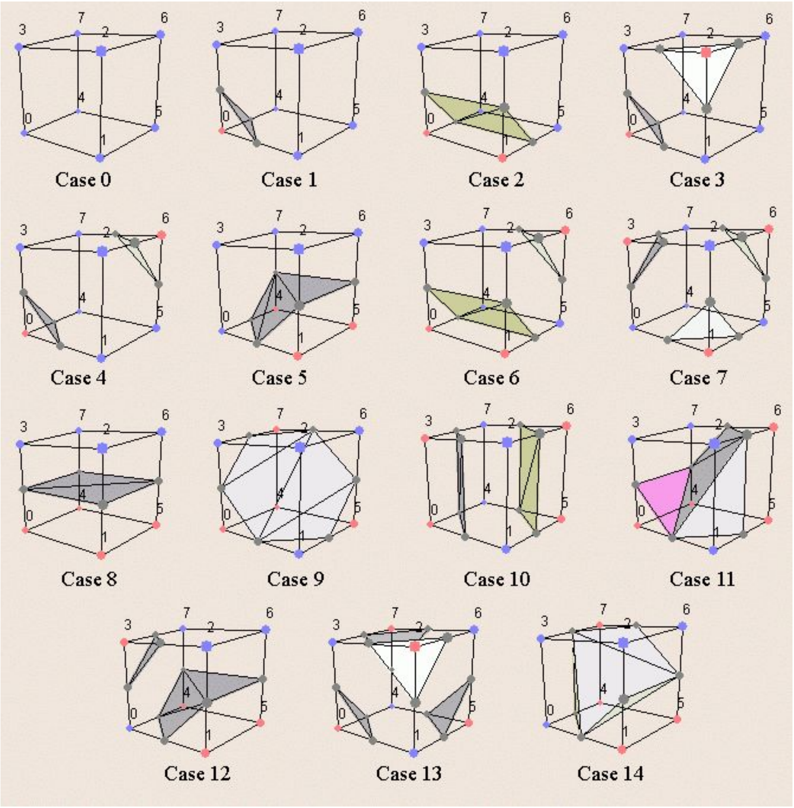
Polygonizing the surface

There are fifteen possible configurations (up to symmetry) of hot/cold vertices in the cube. →

- With rotations, that's 256 cases.

Beware: there are *ambiguous cases* in the polygonization which must be addressed separately. ↓





Images courtesy of [Diane Lingrand]()

Smoothing the surface

Improved edge vertices

- The naïve implementation builds polygons whose vertices are the midpoints of the edges which lie between hot and cold vertices.
- The vertices of the implicit surface can be more closely approximated by points linearly interpolated along the edges of the cube by the weights of the relative values of the force function.
  - $t = (0.5 - F(P1)) / (F(P2) - F(P1))$
  - $P = P1 + t\,(P2 - P1)$

Blobby Modeling - References

D. Ricci, A Constructive Geometry for Computer Graphics, Computer Journal, May 1973

J Bloomenthal, Polygonization of Implicit Surfaces, Computer Aided Geometric Design, Issue 5, 1988

B Wyvill, C McPheeters, G Wyvill, Soft Objects, Advanced Computer Graphics (Proc. CG Tokyo 1986)

B Wyvill, C McPheeters, G Wyvill, Animating Soft Objects, The Visual Computer, Issue 4 1986

http://astronomy.swin.edu.au/~pbourke/modelling/implicitsurf/

http://www.cs.berkeley.edu/~job/Papers/turk-2002-MIS.pdf

http://www.unchainedgeometry.com/jbloom/papers/interactive.pdf

http://www-courses.cs.uiuc.edu/~cs319/polygonization.pdf