Geometric Representations

Stelian Coros

Geometric Representations

- Languages for describing shape
  - Boundary representations
    - Polygonal meshes
    - Subdivision surfaces
    - Implicit surfaces
  - Volumetric models

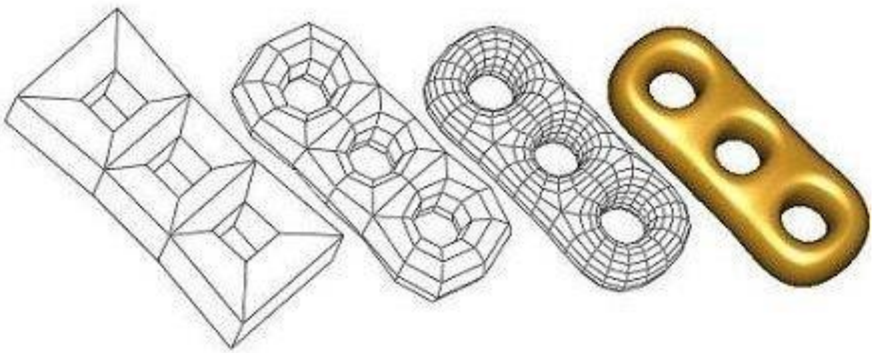
  } Lower Level

  - Parametric models
  - Procedural/generative models

  } Higher Level

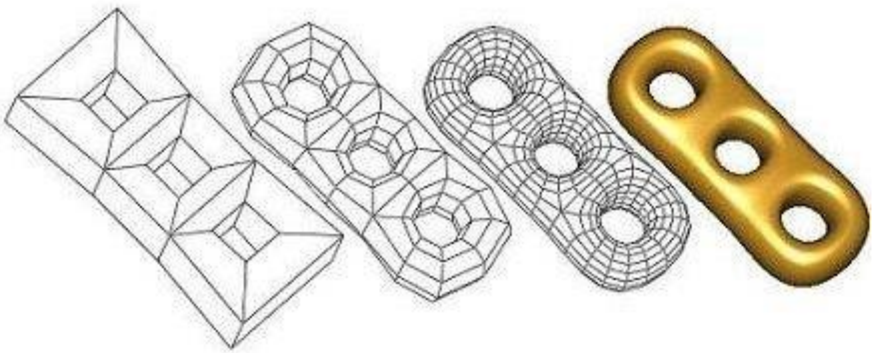
Boundary Representations (B-Reps)

- Only boundary of an object is specified
  - Polygonal mesh
  - Subdivision
  - Implicit



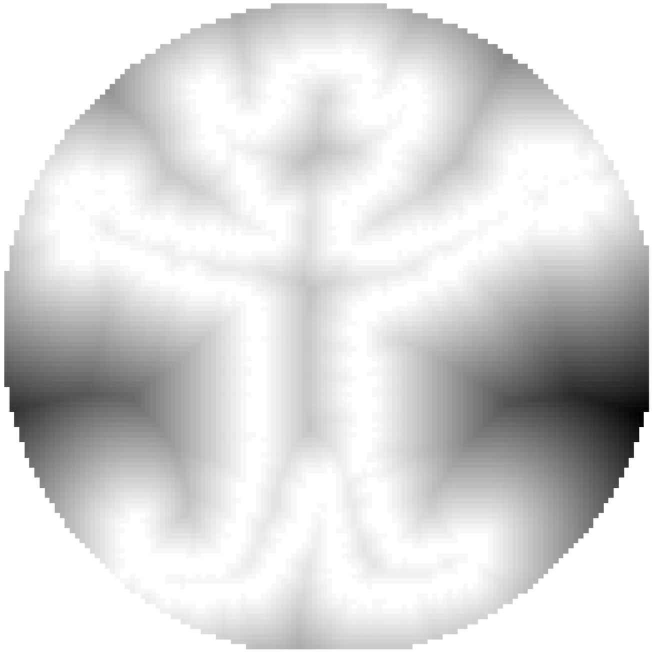
Boundary Representations (B-Reps)

- Only boundary of an object is specified
  - Polygonal mesh
  - Subdivision
  - Implicit



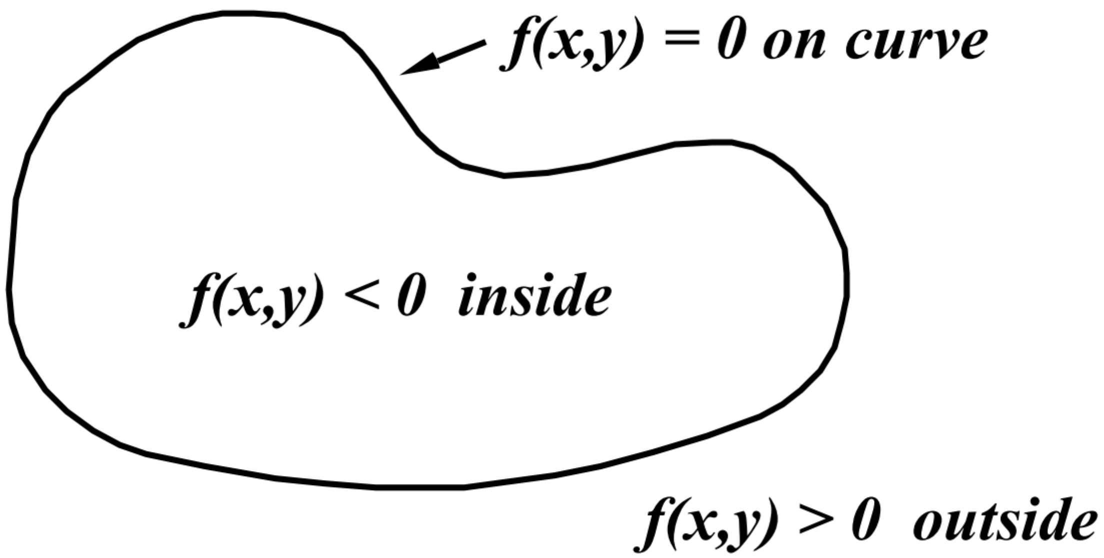
Implicit Surfaces

- Represent surface with function defined over all space



Implicit Surfaces

- Surface defined implicitly by function:
  - $f(x, y, z) = 0$ (on surface)
  - $f(x, y, z) < 0$ (inside)
  - $f(x, y, z) > 0$ (outside)



Implicit Functions

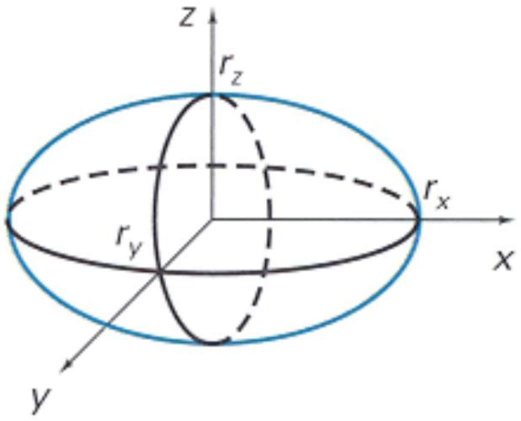
- a relation of the form $R(x_1,\ldots, x_n) = 0$

$$\left(\frac{x}{r_x}\right)^2+\left(\frac{y}{r_y}\right)^2+\left(\frac{z}{r_z}\right)^2-1=0$$

Implicit Surface Properties

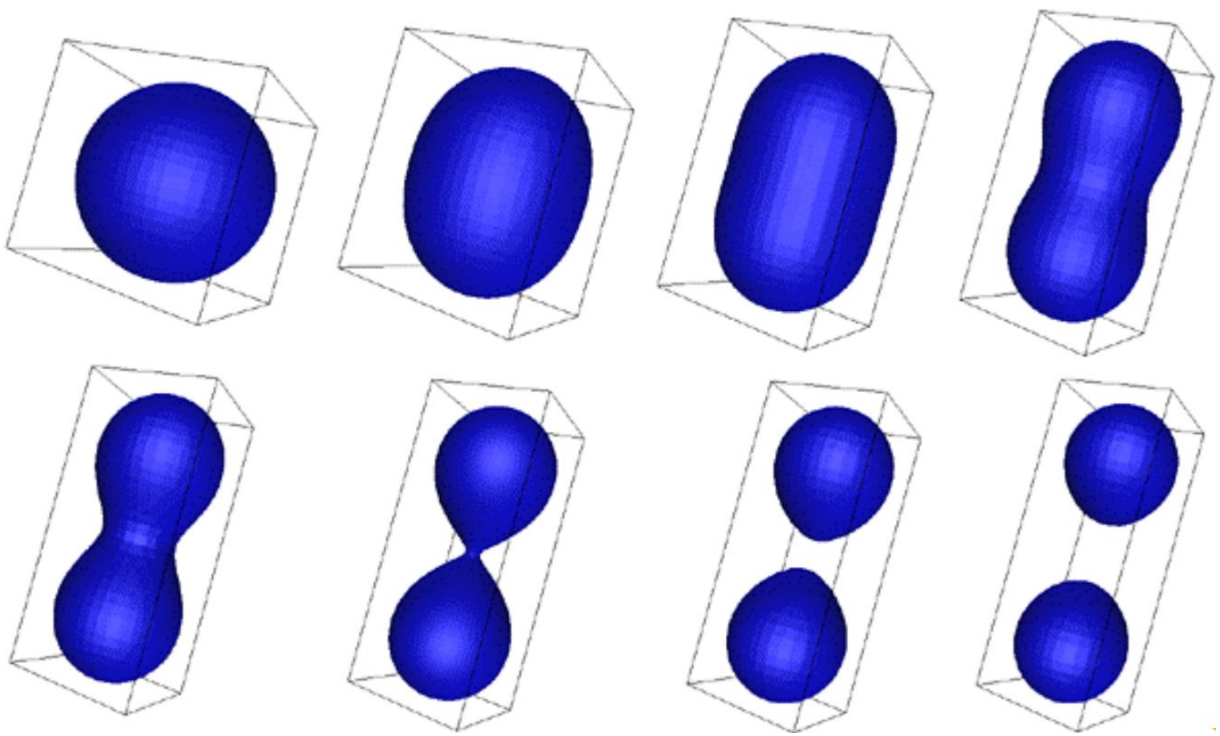
- Efficient check for whether point is inside
  - Evaluate f(x,y,z) to see if point is inside/outside/on

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 - 1 = 0$$



Implicit Surface Properties

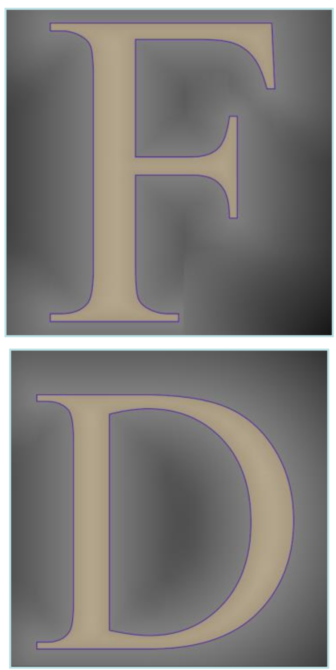
- Efficient topology changes
  - Surface is not represented explicitly!



Implicit Surface Properties

- Efficient to compute boolean operations

$$\text{Intersection: } (A \cap B) = \max(A, B)$$
$$\text{Union: } (A \cup B) = \min(A, B)$$
$$\text{Difference: } (A \setminus B) = \max(A, -B)$$



Union of two shapes

Implicit Surface Representations

- How do we define implicit functions?
  - Algebraic expressions
  - “Blobby” models
  - Samples

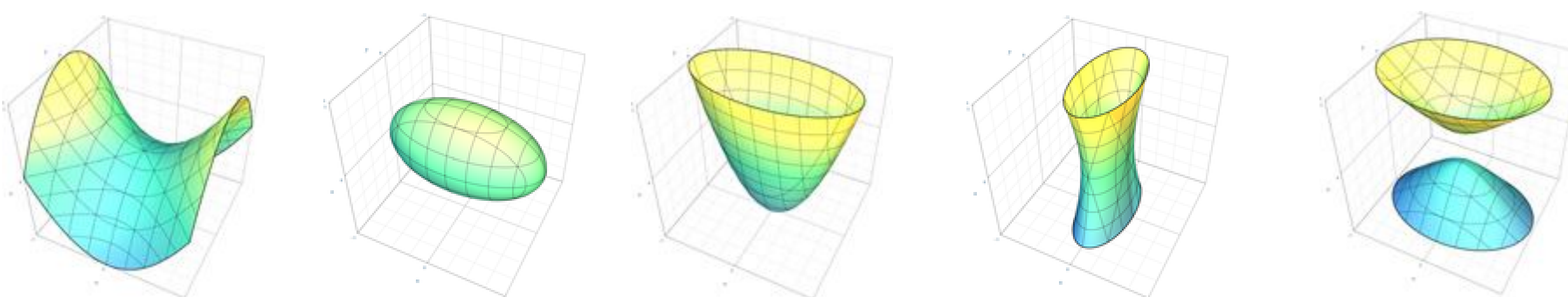
Implicit Surface Representations - Algebraics

- Implicit function is polynomial

$$f(x, y, z) = ax^d + by^d + cz^d + ex^{d-1}y + fx^{d-1}z + gy^{d-1}x + \dots$$

- Most common form: quadrics

$$f(x, y, z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fxz + 2gx + 2hy + 2jz + k$$

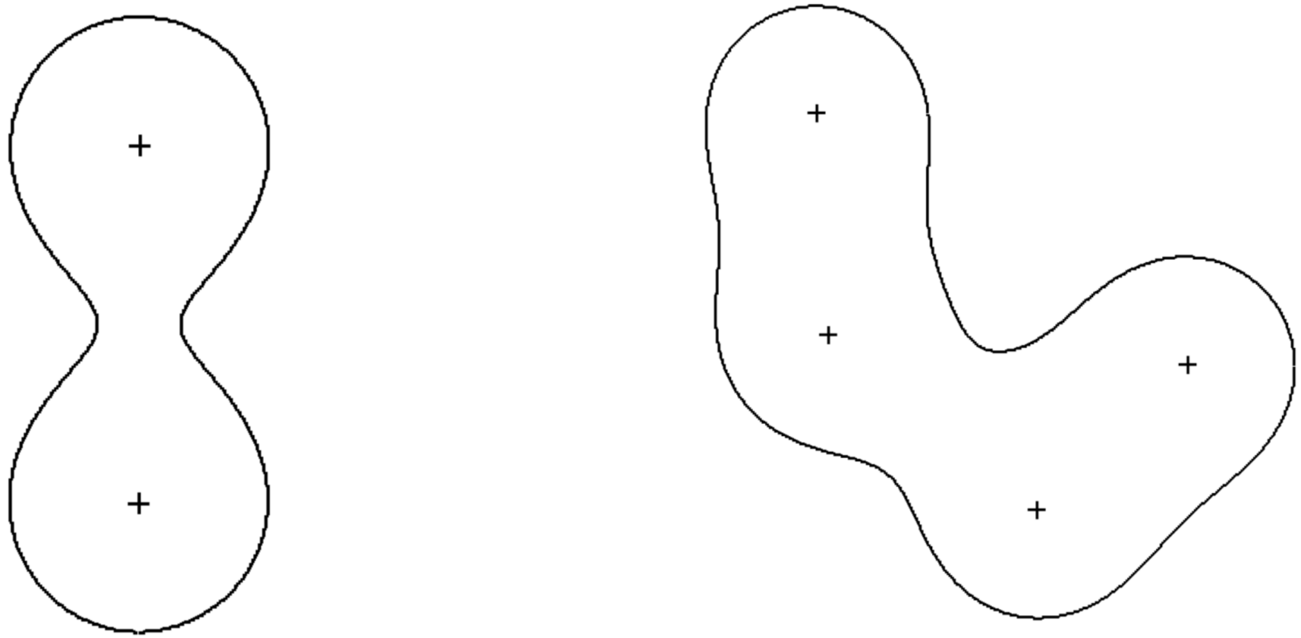


Implicit Surface Representations - Blobbies

- Blobby molecules (radial basis functions):

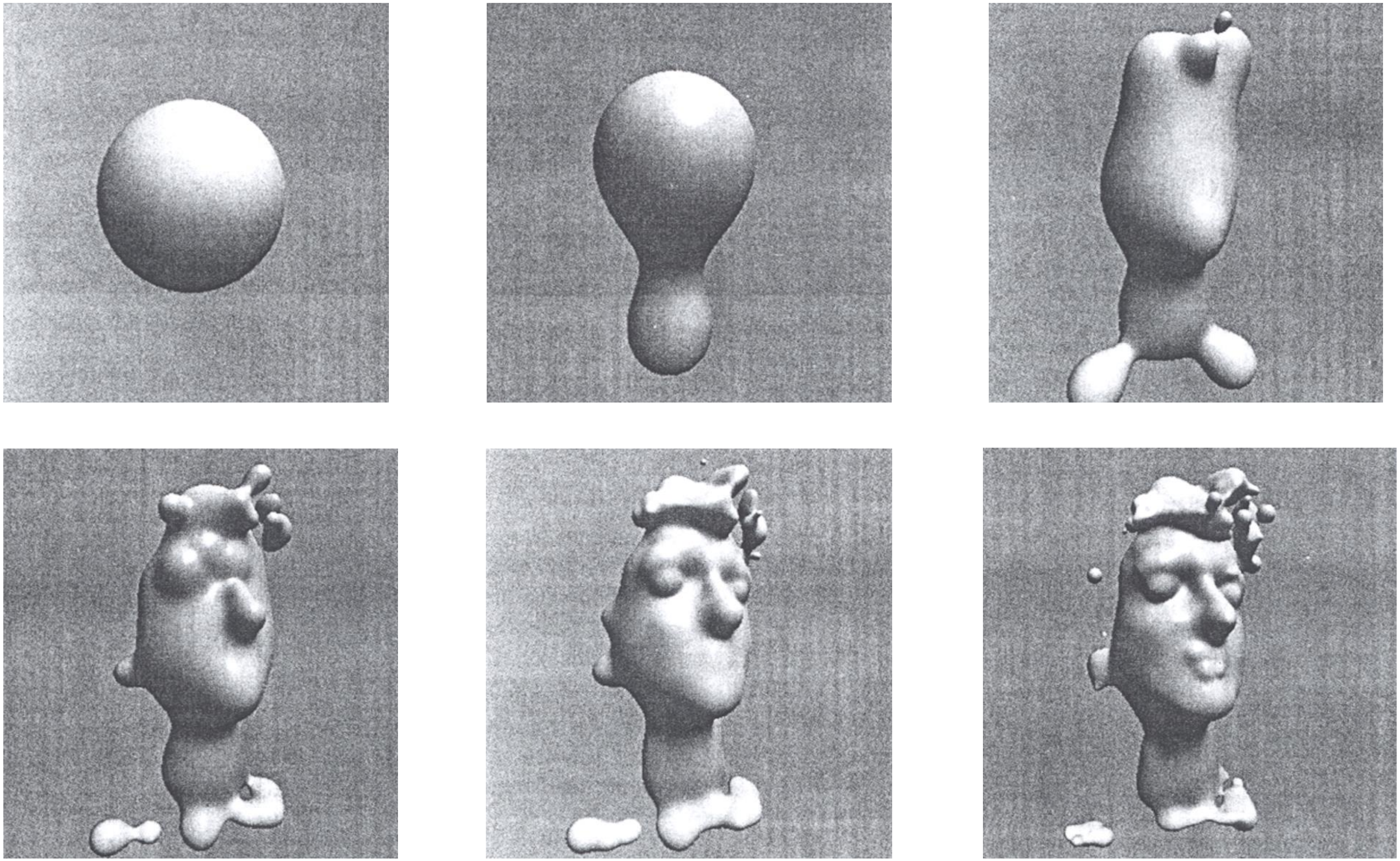
$$D(r) = ae^{-br^2}$$

- Implicit function is sum of blobs



Implicit Surface Representations - Blobbies

N = 1

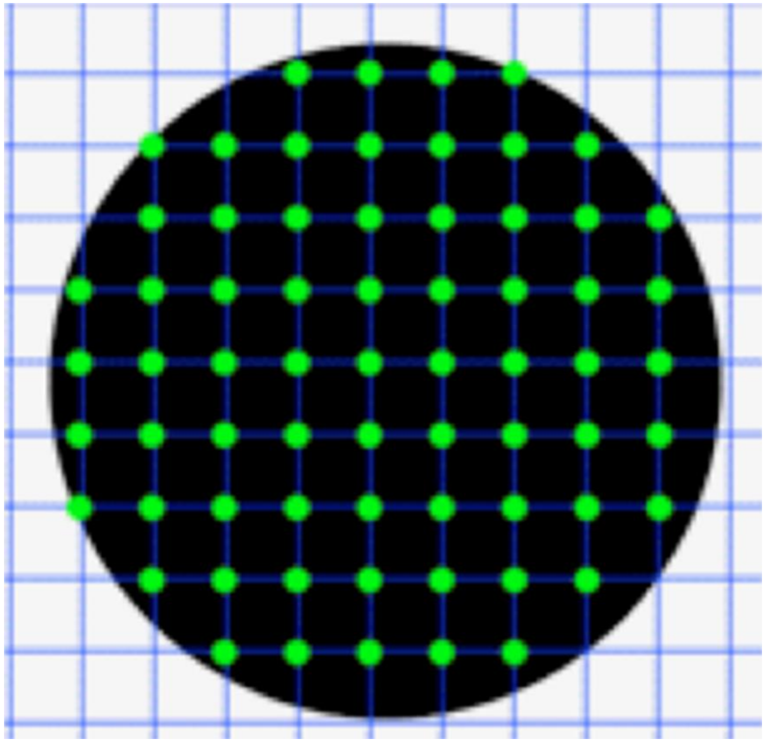


N = 243

Implicit Surface Representations - Samples

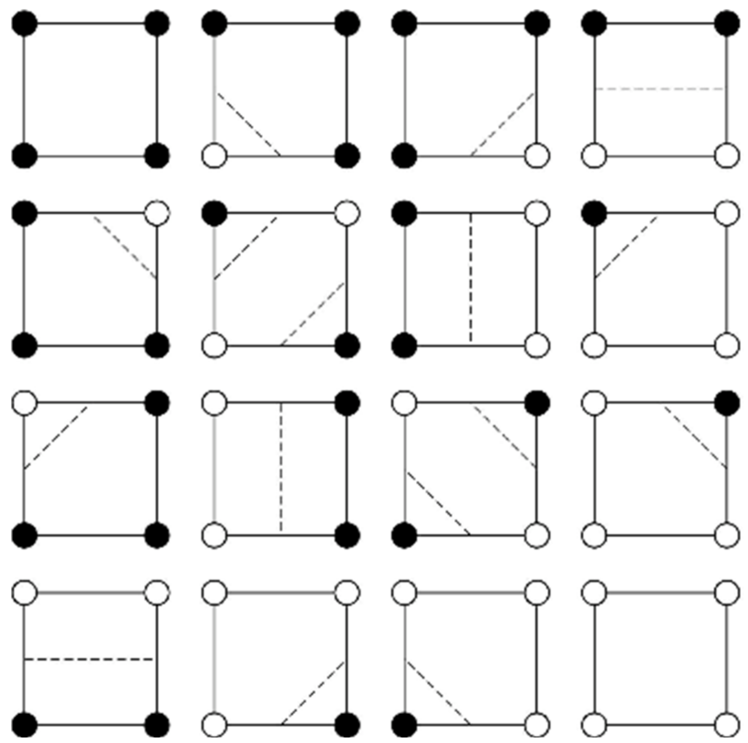
- Function value samples stored explicitly
  - Most common example: voxels (regular grid)
  - Surface?

From sampled implicit functions to surfaces

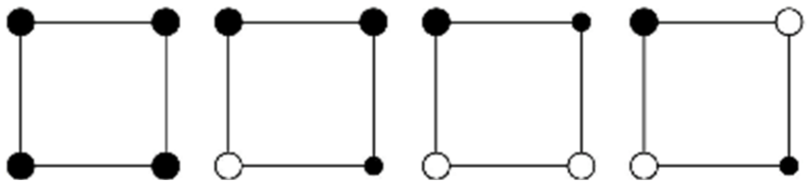


From sampled implicit functions to surfaces

16 cases for vertex labels

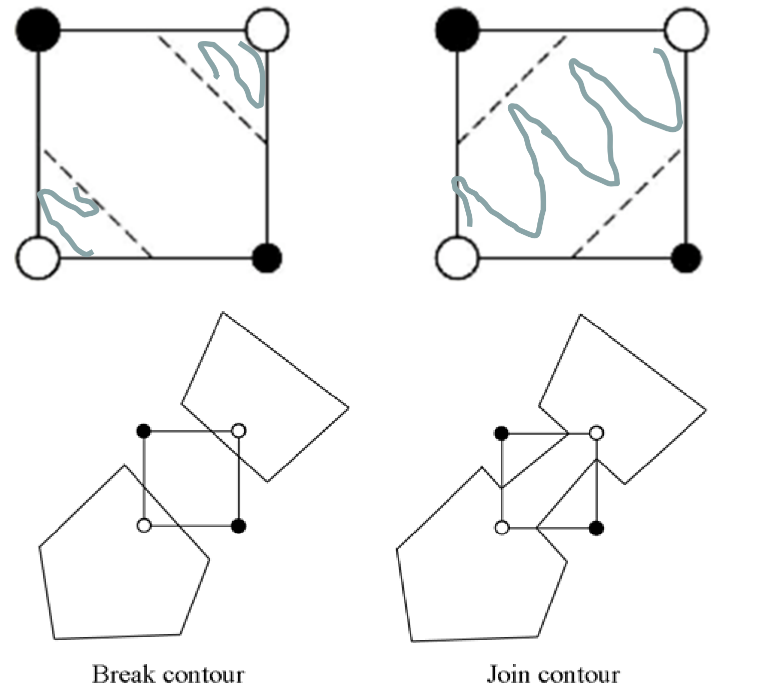


4 unique (based on symmetry)

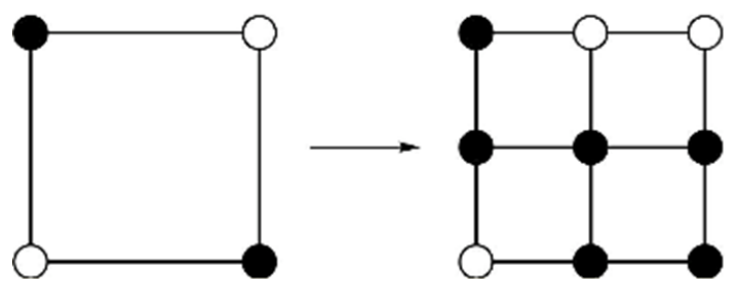


From sampled implicit functions to surfaces

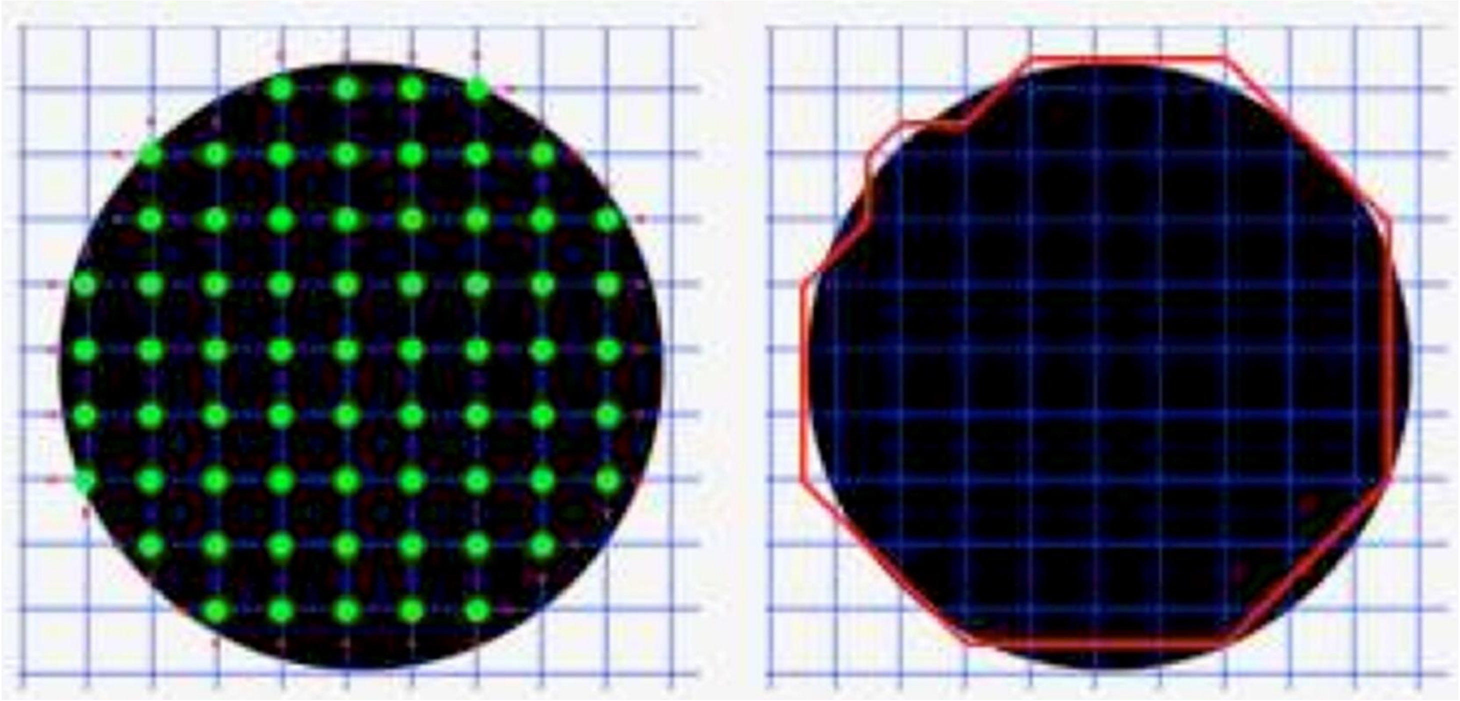
- Ambiguities



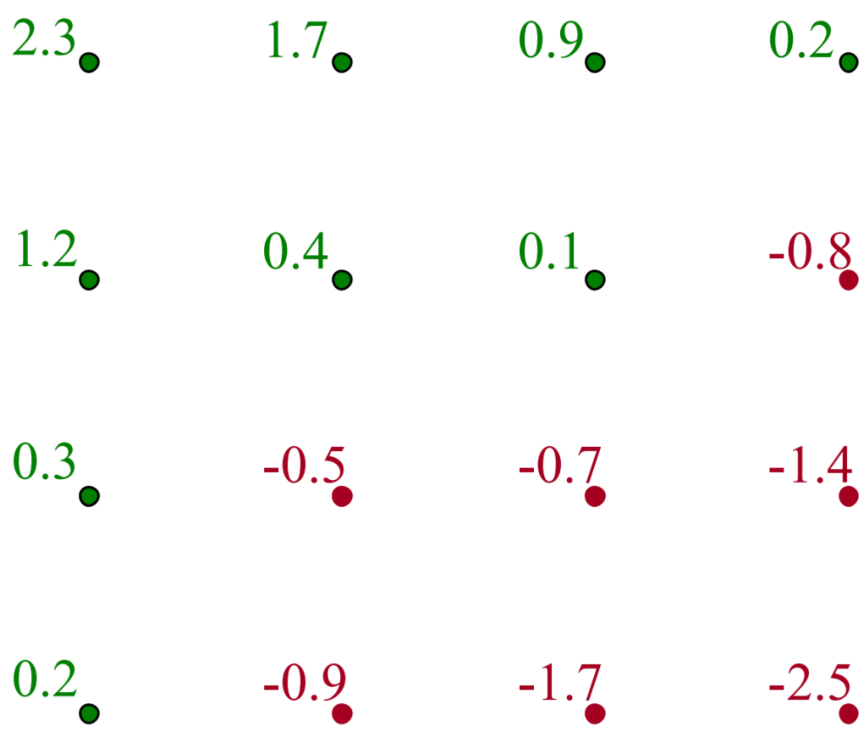
- Use priors (or biases)
- Match neighbors
- If at all possible, subdivide



From sampled implicit functions to surfaces

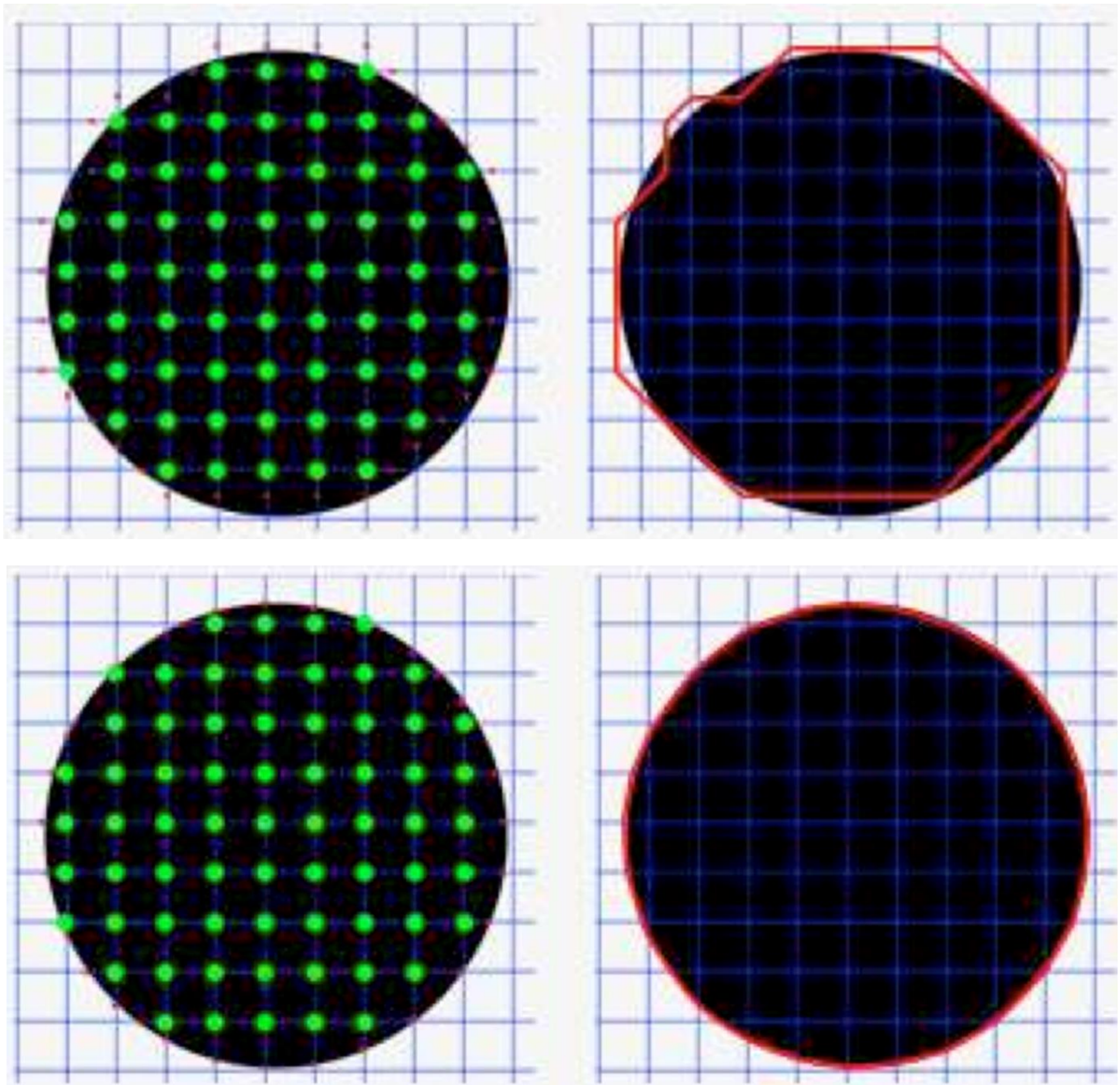


Can we do better?



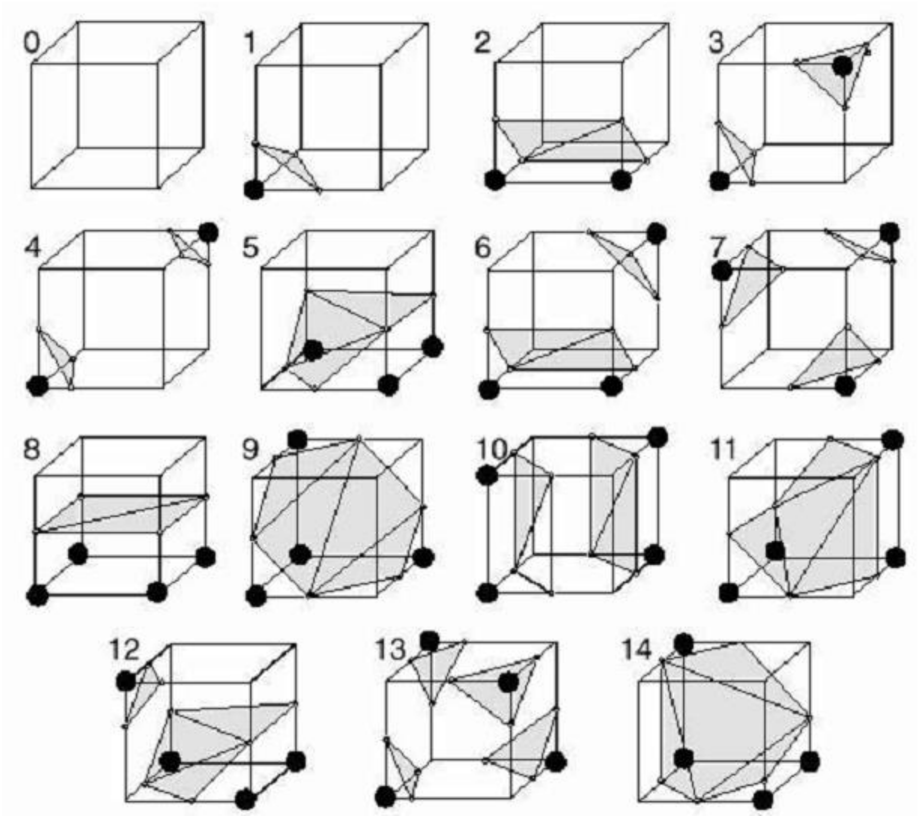
Interpolate!

From sampled implicit functions to surfaces



From sampled implicit functions to surfaces

- Generalization to 3D?
  - Same concept → Marching Cubes
  - How many cases?

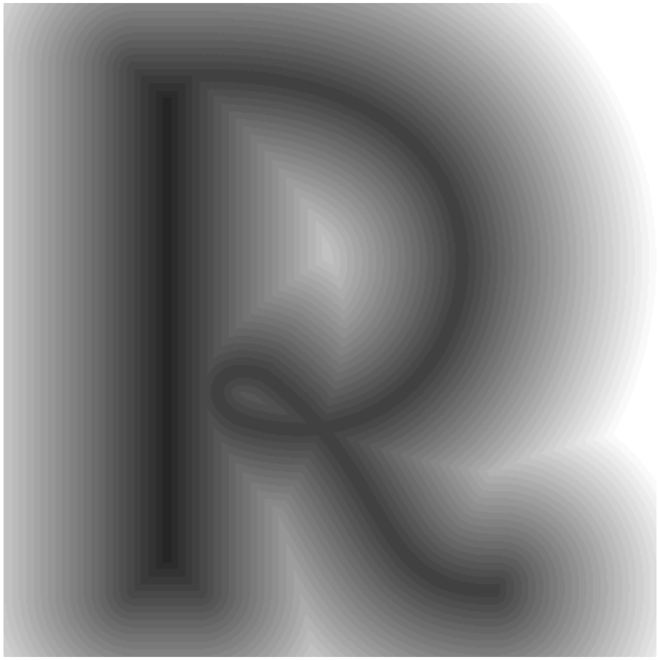


After elimination of symmetric cases!

Important special case: Signed Distance Fields

- An object's distance field represents, for any point in space, the signed distance from that point to the object

R shape



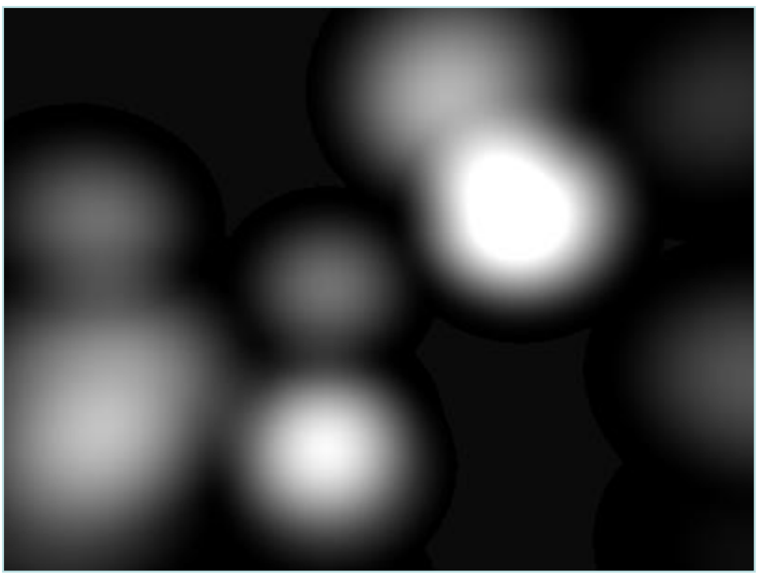
Distance field of R

Distance Fields

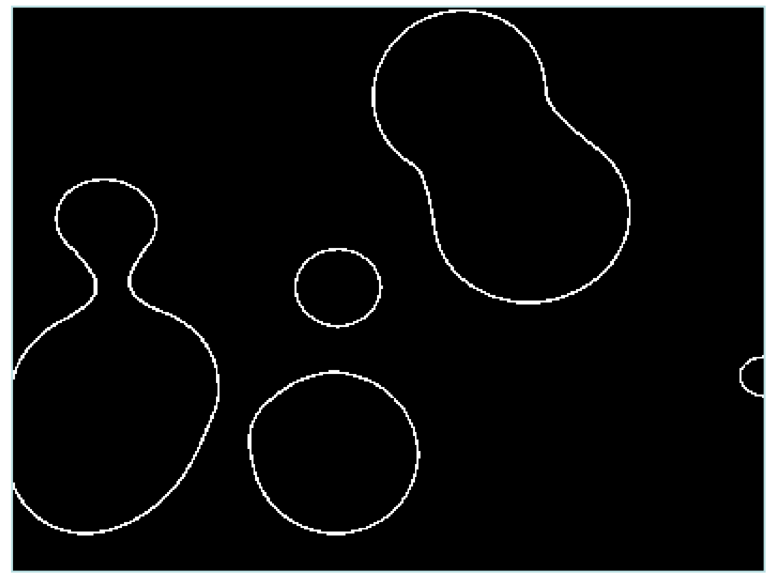
- Distance fields are <u>implicit representations</u> of shape …
  - See *Introduction to Implicit Surfaces* (J. Bloomenthal, ed.), 1997

Distance Fields

- For a shape represented by a distance field, the shape's boundary, Ω, is the zero-valued iso-surface of the distance function



F(x)

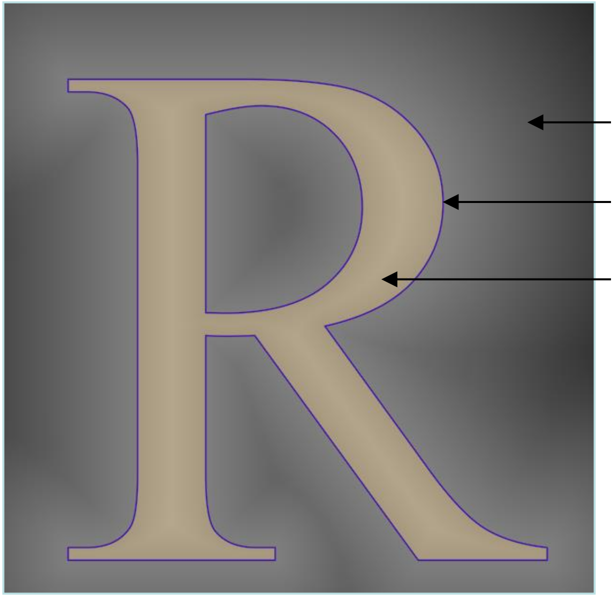


An iso-contour of F(x) where F(x) = 0

Perry and Frisken

Distance Fields

- Distance fields represent more than just the object outline
  - Represent the object interior, exterior, and its boundary

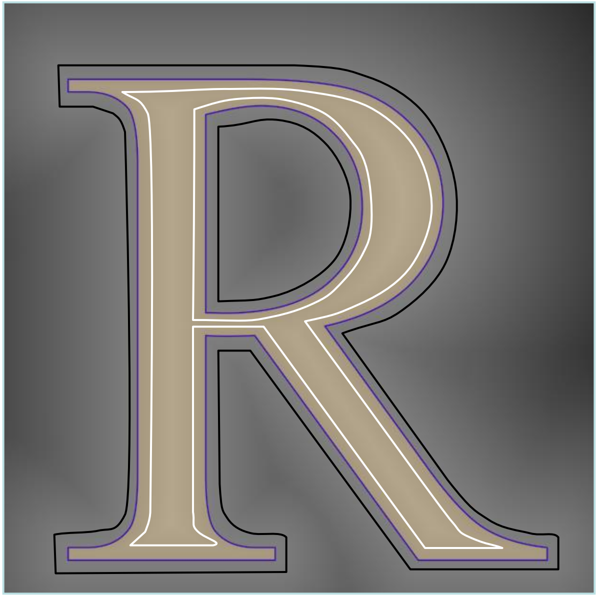


Shape's distance field

Perry and Frisken

Distance Fields

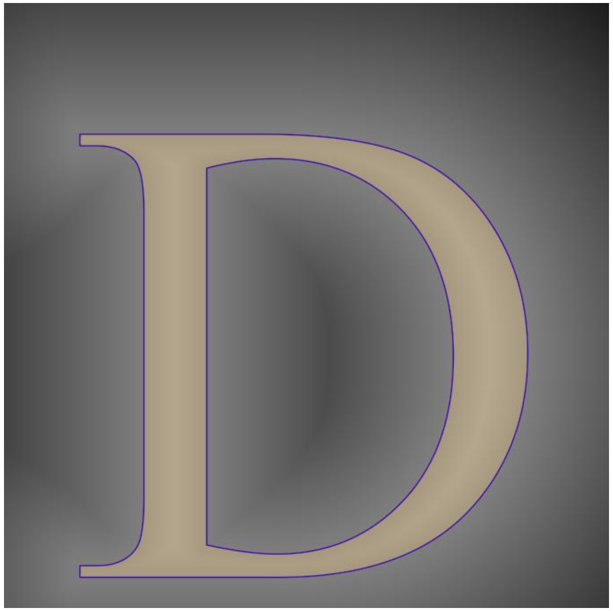
- Distance fields represent more than just the object outline
  - infinite number of offset surfaces – iso-contours



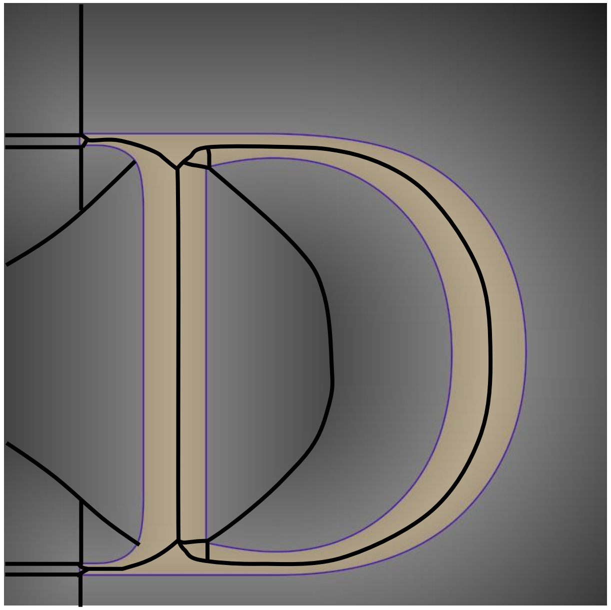
Boundary offsets

Advantages - smoothness and continuity

- Distance fields are C^0 continuous everywhere
- Euclidean distance fields are C^1 continuous except at boundaries of Voronoi regions



Distance field is C^0 continuous

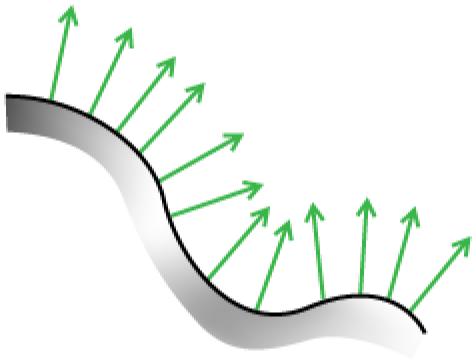


C^1 continuous except at Voronoi boundaries

Advantages

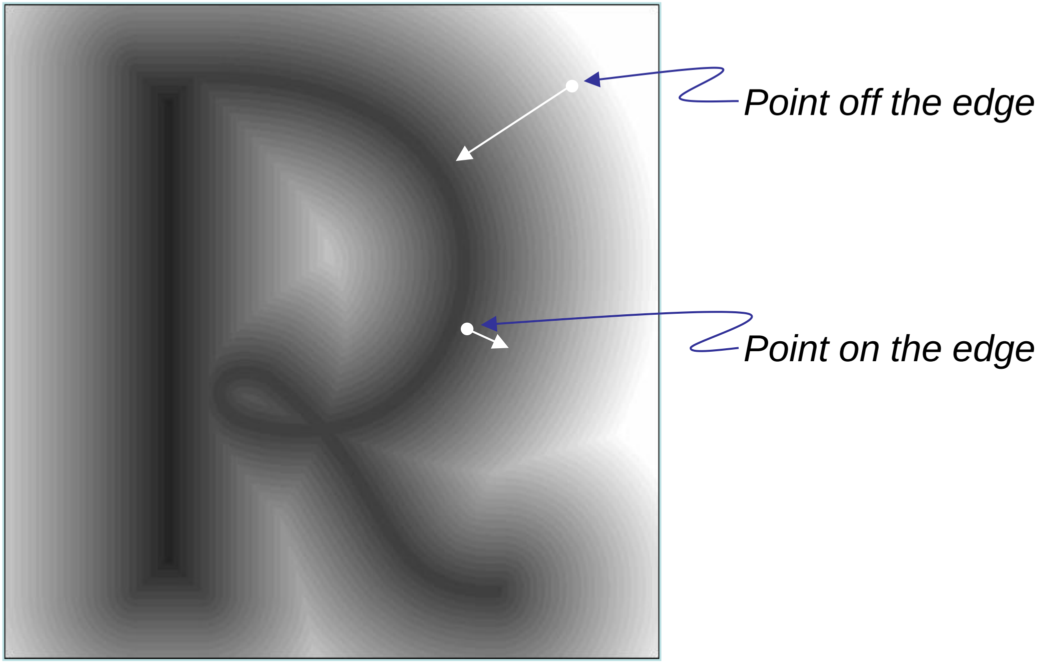
- Gradients can be computed everywhere

$$\vec{\nabla} f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)$$



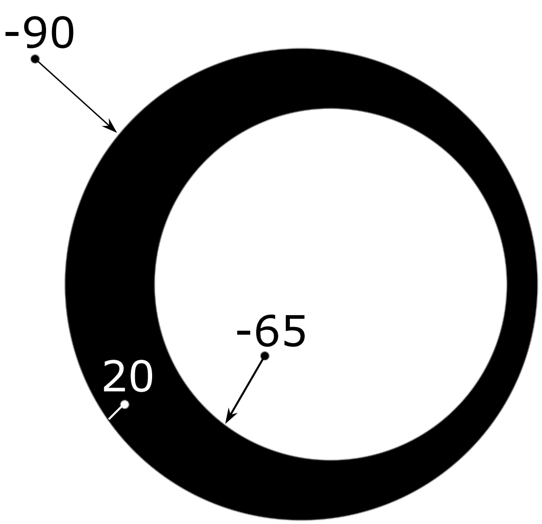
Advantages

- Gradient of the distance field yields
  - Surface normal for points on the edge
  - Direction to the closest point on surface for points outside

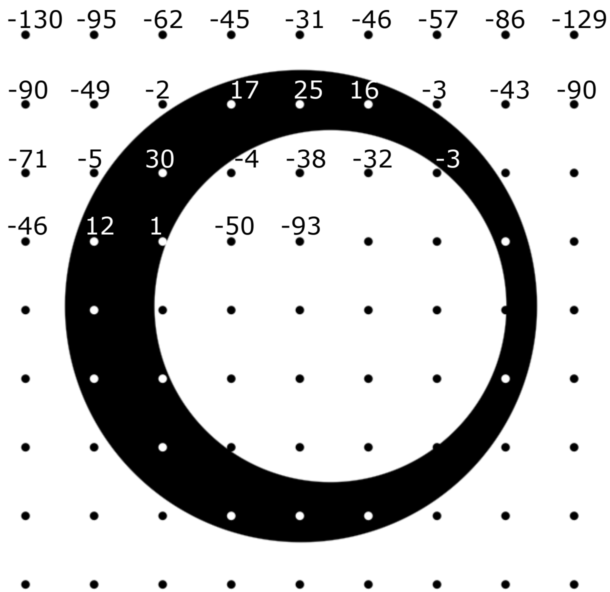


Computing Distance Fields

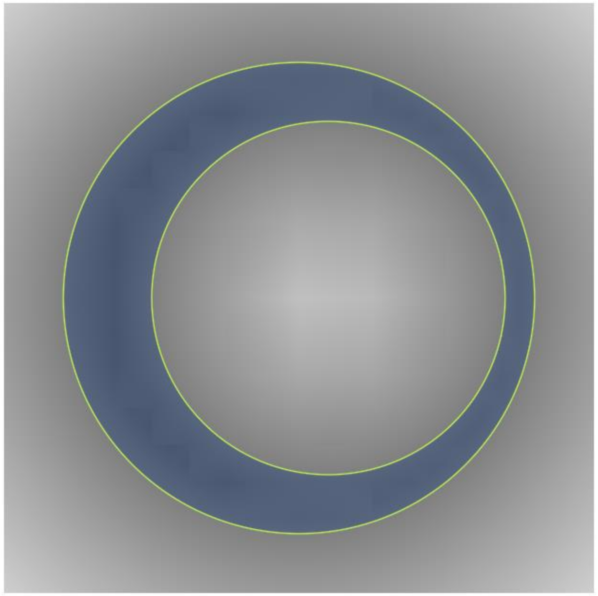
- Sampled volumes
  - Distances are computed and stored in a regular 3D grid
  - Gradients estimated with finite differences
  - Distances/gradients at non-grid locations are interpolated



2D shape



Regularly sampled distance values

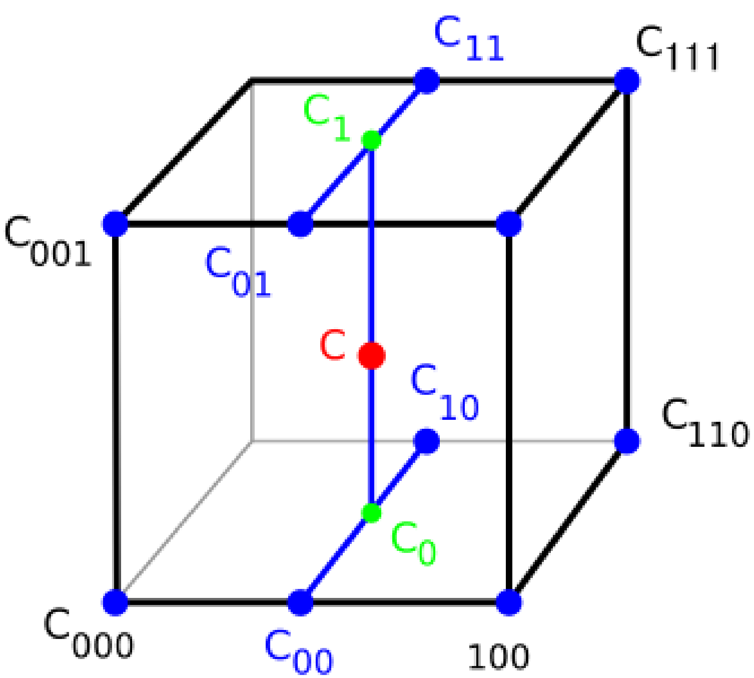


2D distance field

Perry and Frisken

Using Distance Fields

- Distances and gradients are estimated using trilinear interpolation



$$
\begin{aligned}
c_{00} &= V[x_0, y_0, z_0](1 - x_d) + V[x_1, y_0, z_0]x_d \\
c_{10} &= V[x_0, y_1, z_0](1 - x_d) + V[x_1, y_1, z_0]x_d \\
c_{01} &= V[x_0, y_0, z_1](1 - x_d) + V[x_1, y_0, z_1]x_d \\
c_{11} &= V[x_0, y_1, z_1](1 - x_d) + V[x_1, y_1, z_1]x_d
\end{aligned}
$$

Perry and Frisken

Grid resolution

- Sampled volumes
  - Smooth surfaces are well represented by a relatively small number of samples

Radius = 30 voxels
100 x 100 x 100

Radius = 3 voxels
10 x 10 x 10

Radius = 2 voxels
10 x 10 x 10

Radius = 1.5 voxels
10 x 10 x 10

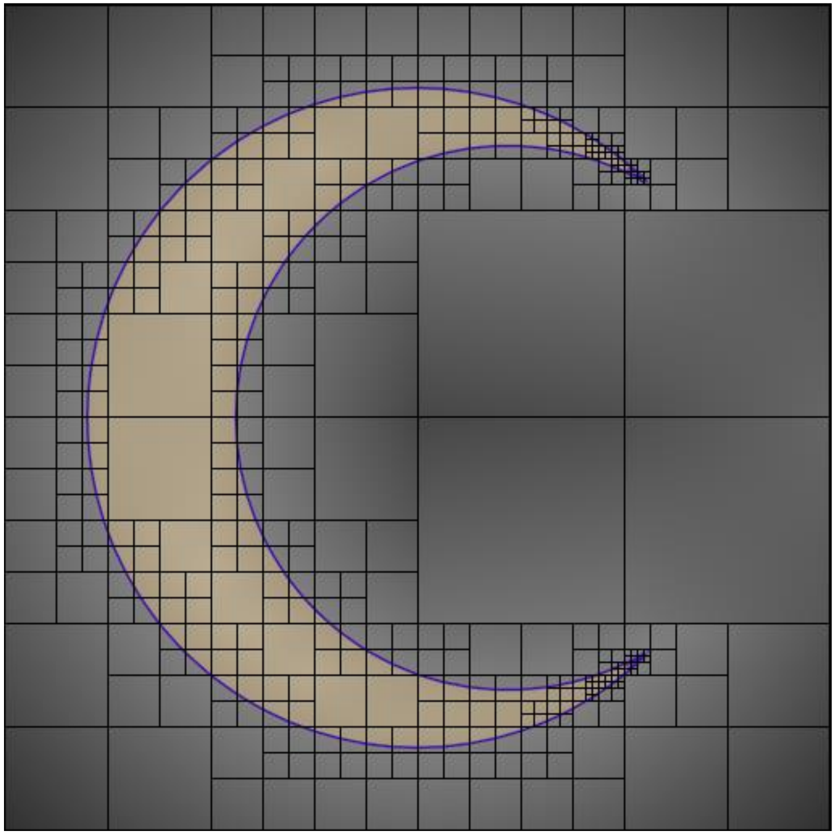
Perry and Frisken

Regularly Sampled Distance Fields

- Distance fields must be sampled at high enough rates to avoid aliasing (jagged edges)
- Very dense sampling is required when fine detail is present
- Regularly sampled distance fields require excessive memory when *any* fine detail is present

Adaptively Sampled Distance Fields

Sample at low rates where the distance field is smooth. Sample at higher rates only where necessary (e.g., near corners).

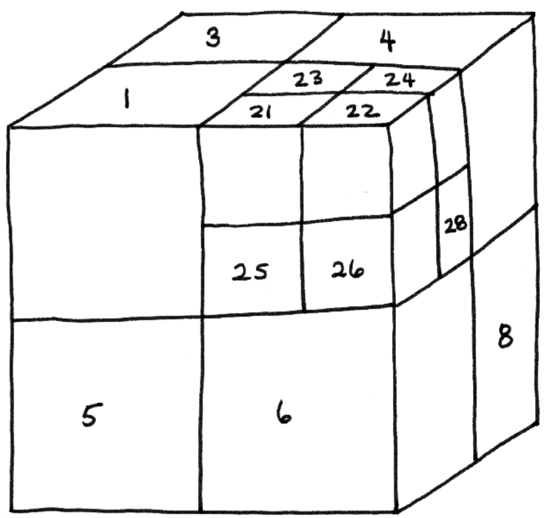


A 2D crescent ADF and its quadtree data structure

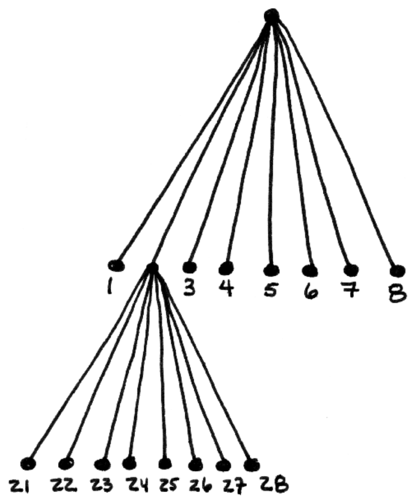
Perry and Frisken

Octree-based ADFs

- Store distance values at cell vertices of an octree



Spatial structure of an octree



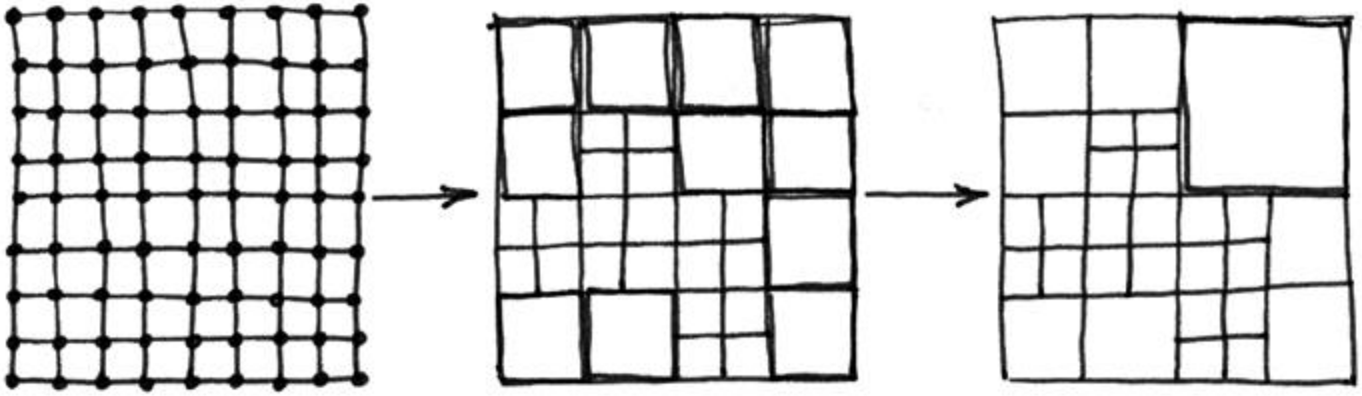
Tree structure of an octree

Perry and Frisken

Adaptively Sampled Distance Fields

- Detail-directed sampling
  - high sampling rates only where needed
- Spatial data structure
  - fast localization for efficient processing
- ADFs consist of
  - adaptively sampled distance values …
  - organized in a spatial data structure …
  - with a method for reconstructing the distance field from the sampled distance values

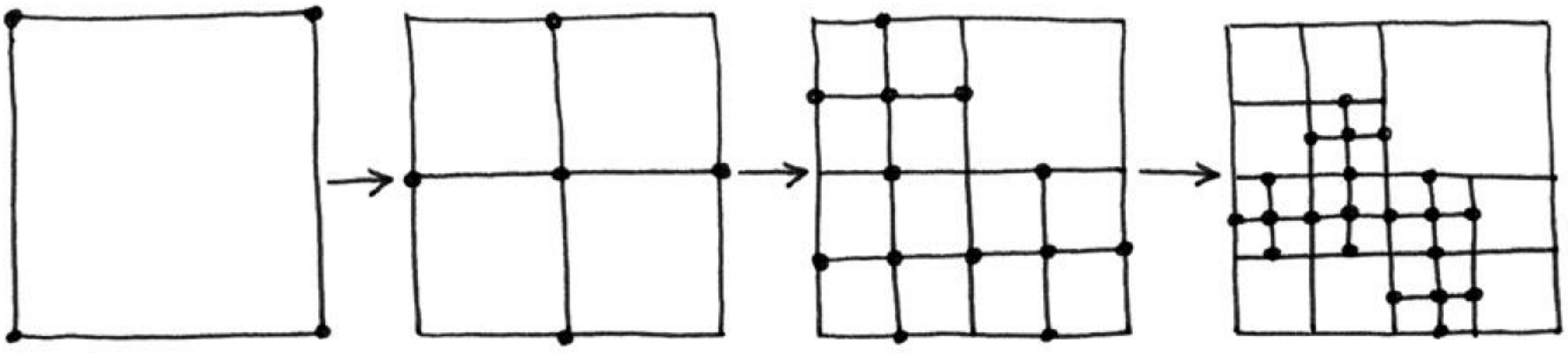
Bottom-up Generation



Fully populate Recursively merge

Perry and Frisken

Top-down Generation



Initialize root cell

Recursively subdivide

Perry and Frisken

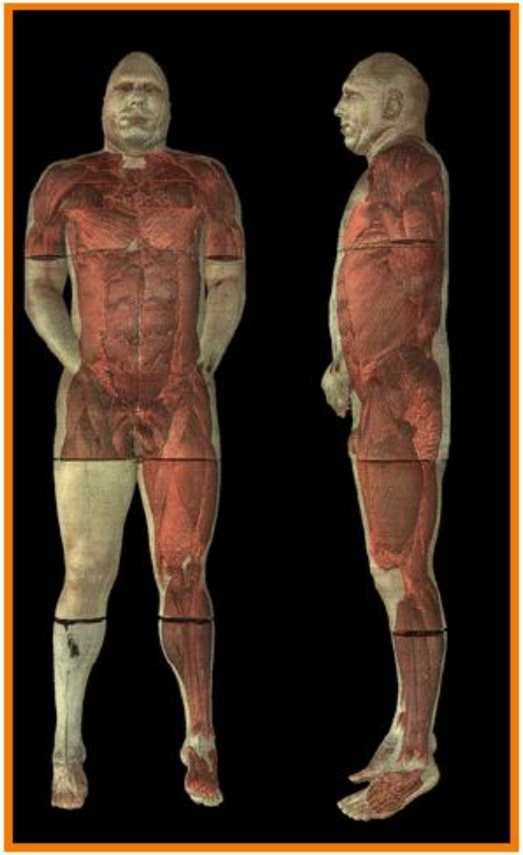
Implicit Surfaces Summary

- Disadvantages:
  - Indirect specification of surface
  - Hard to describe sharp features (but adaptive schemes help)

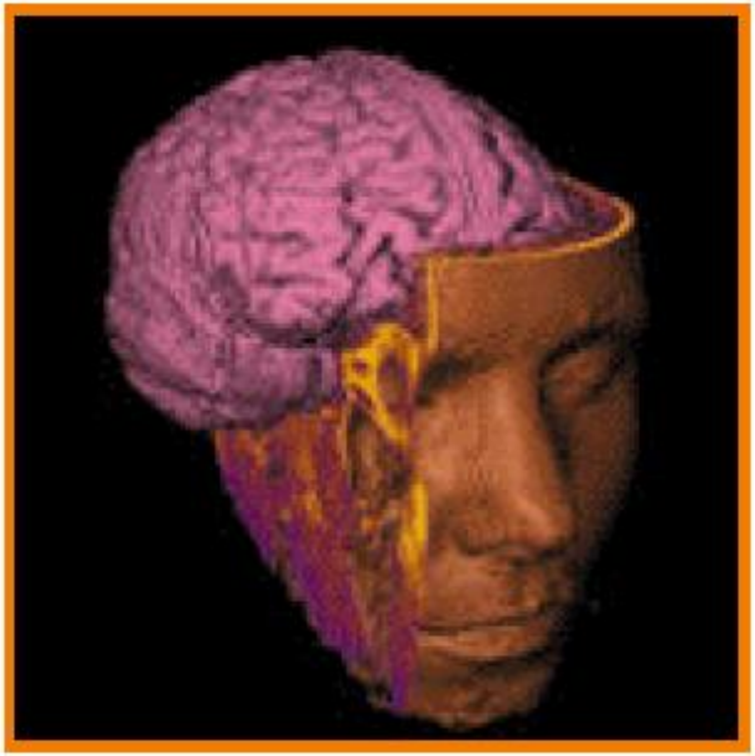
- Advantages:
  - Easy to test if point is on/off surface
  - Gradients are readily available anywhere
  - Easy to compute intersections/unions/differences
  - Easy to handle topological changes
  - Can holds a lot more information than just the surface!!

Solid Modeling

- Represent solid interiors of objects



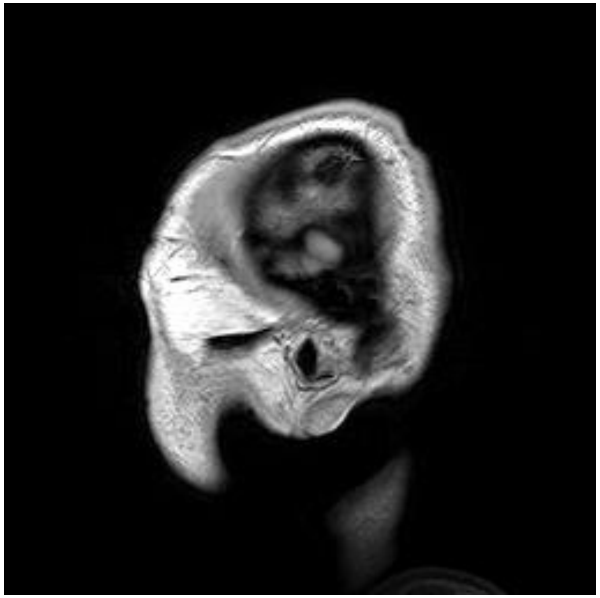
Visible Human
(National Library of Medicine)

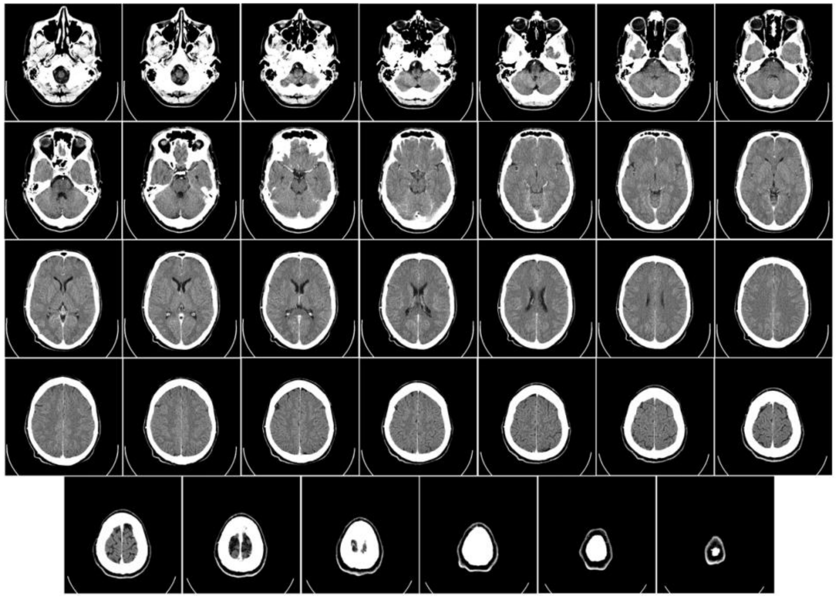


SUNY Stony Brook

Why Volumetric Representations?

- Some acquisition methods generate solids
  - Magnetic Resonance Imaging (MRI)
  - Computed Tomography (CT/ CAT)



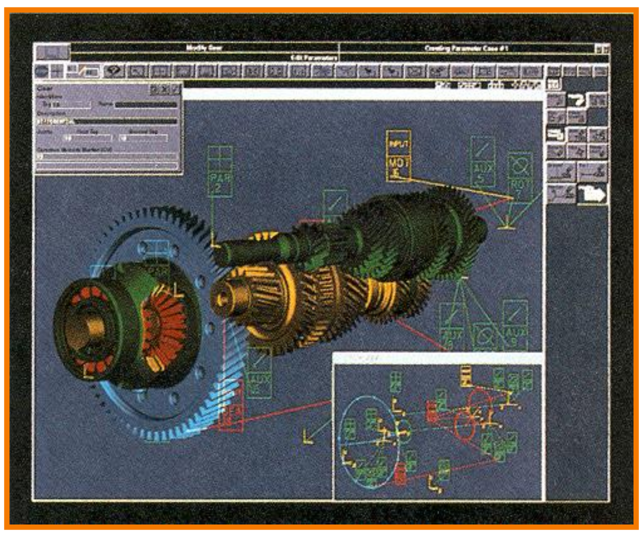


Source: Wikipedia

Why Volumetric Representations?

- Some applications require solids
  - CAD/CAM
  - material(s) need to be specified inside the object

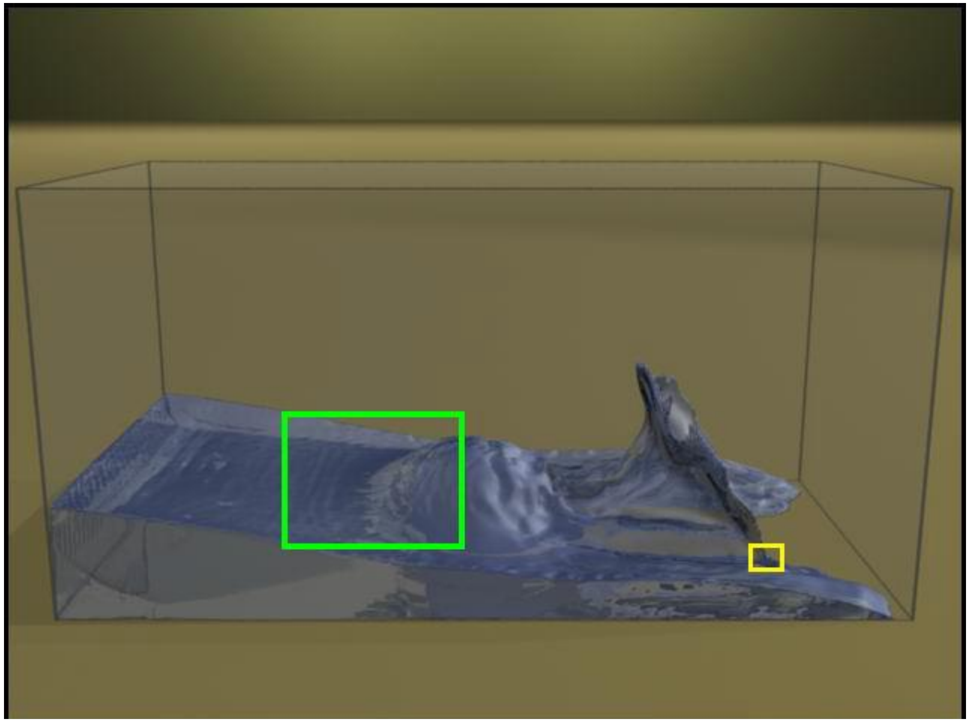
Multi-material 3D Printing

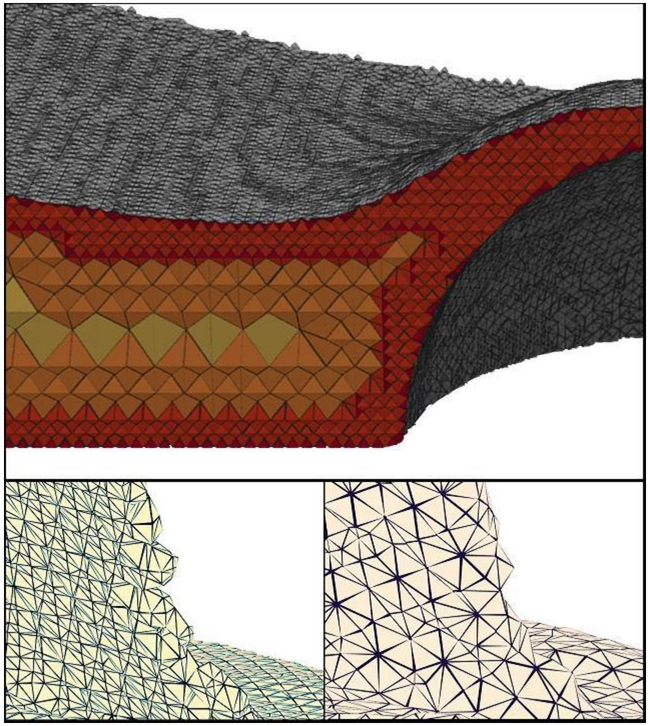


Intergraph Corporation

Why Volumetric Representations?

- Some algorithms require solids
  - Physically-based simulation

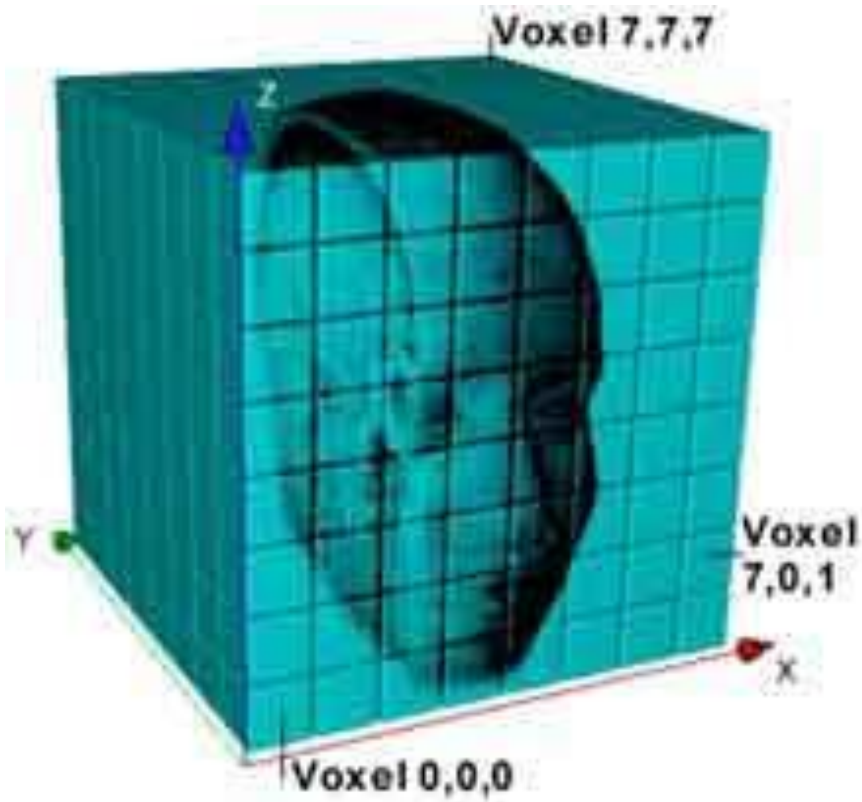




Source: Chentanez

Voxels (Volume Elements)

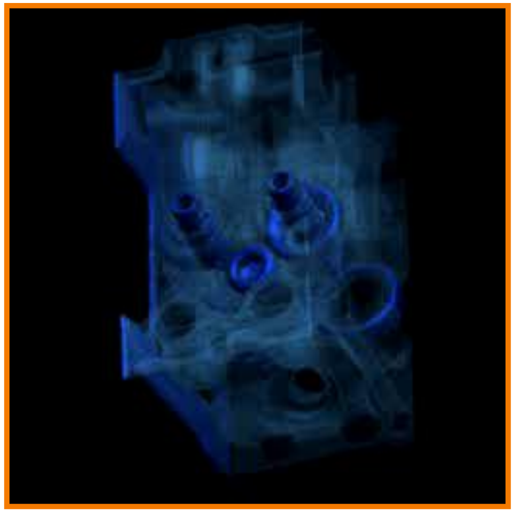
- Partition space into a uniform grid
  - Grid cells are called voxels (like pixels)



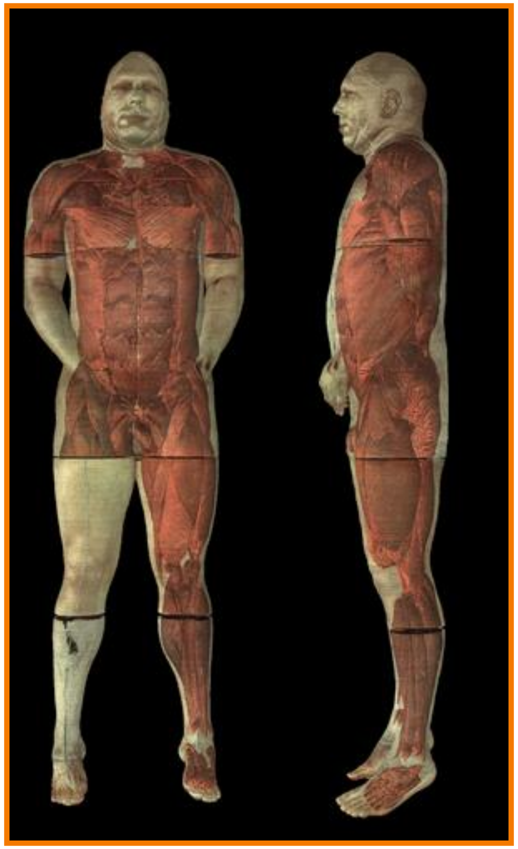
www.volumegraphics.com

Voxels

- Store properties of solid object with each voxel
  - Occupancy
  - Color
  - Density
  - Temperature
  - etc.



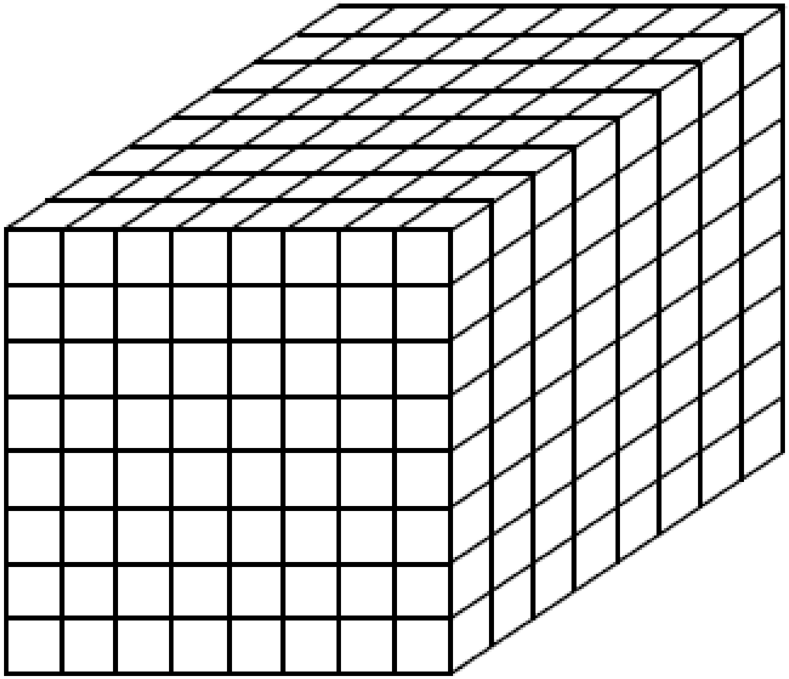
Engine Block
Stanford University



Visible Human
(National Library of Medicine)

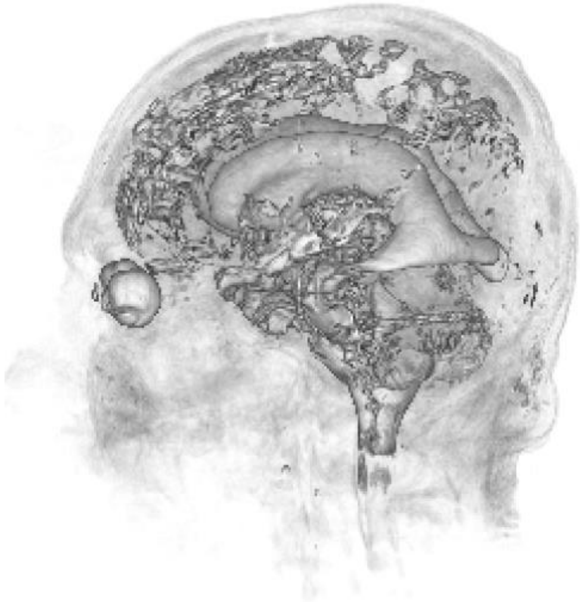
Voxel Storage

- $O(n^3)$ storage for n x n x n grid
  - 1 billion voxels for 1000 x 1000 x 1000



Voxel Processing

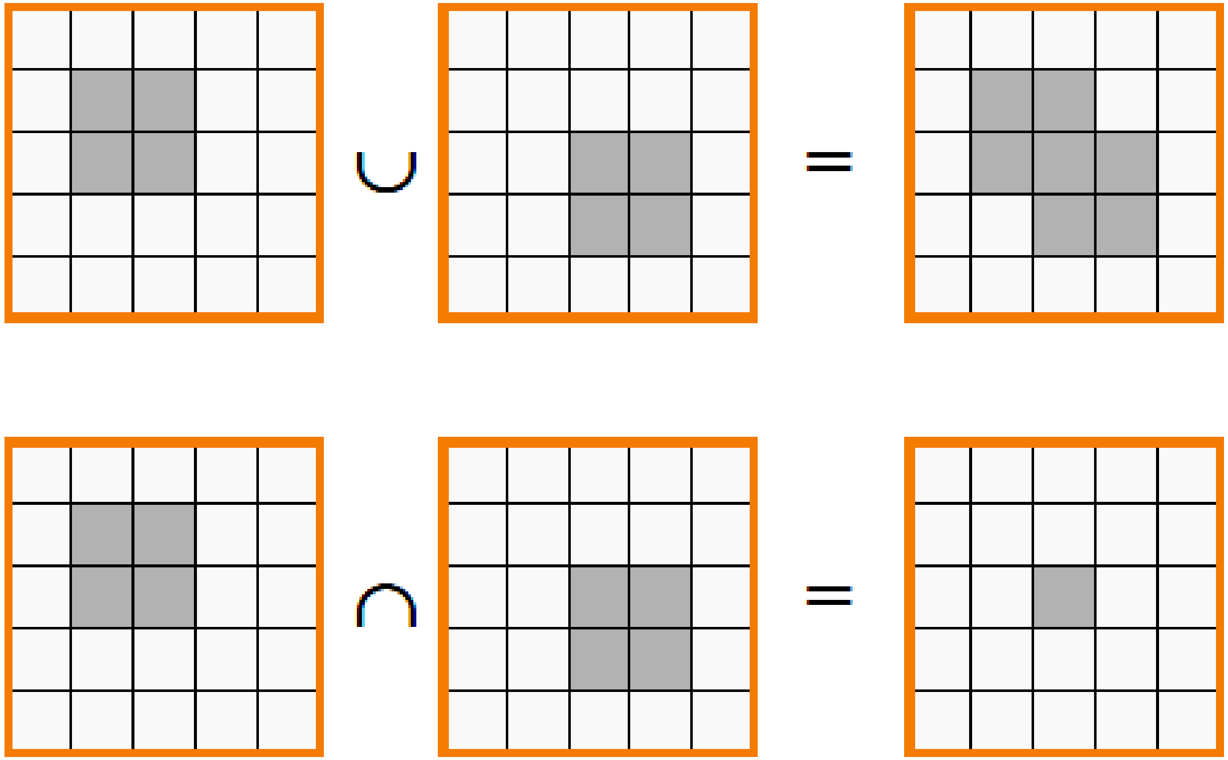
- Signal processing (just like images)
  - Reconstruction
  - Resampling
- Typical operations
  - Blur
  - Edge detection
  - Warp
  - etc.
- Often fully analogous to image processing



www.volumegraphics.com

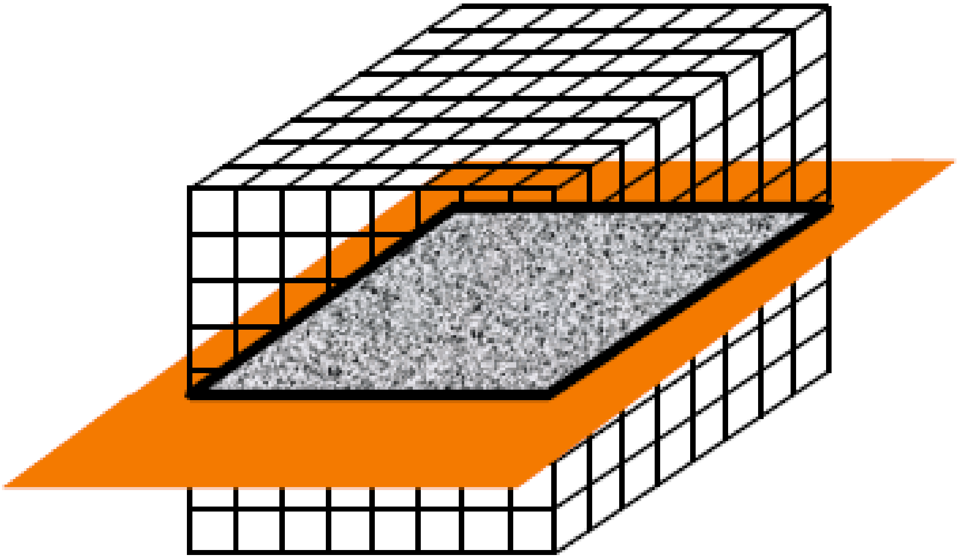
Voxel Boolean Operations

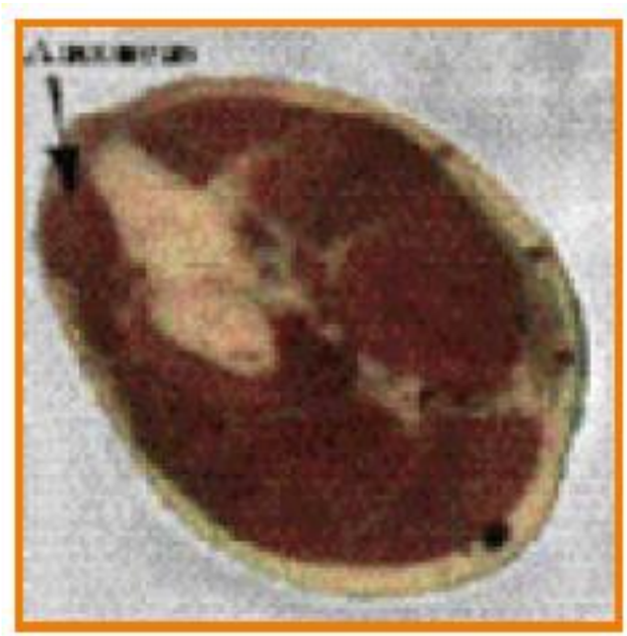
- Compare objects voxel by voxel
  - Trivial



Voxel Display

- Slicing
  - Draw 2D image resulting from intersecting voxels with a plane

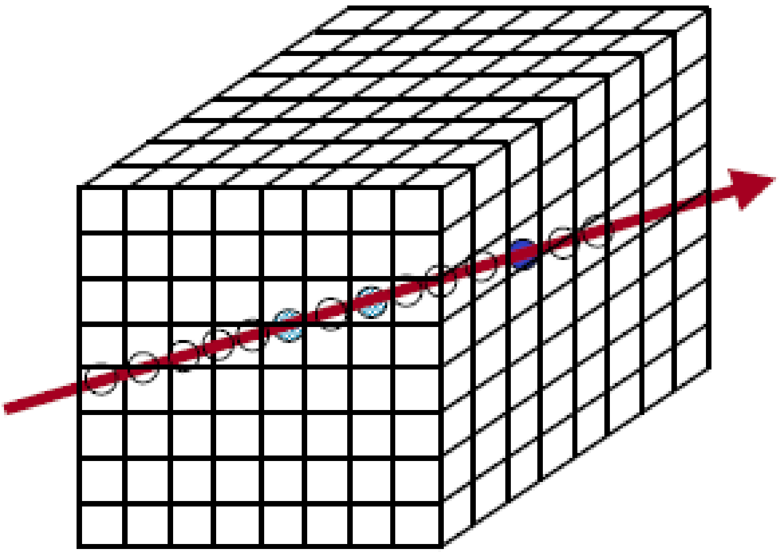


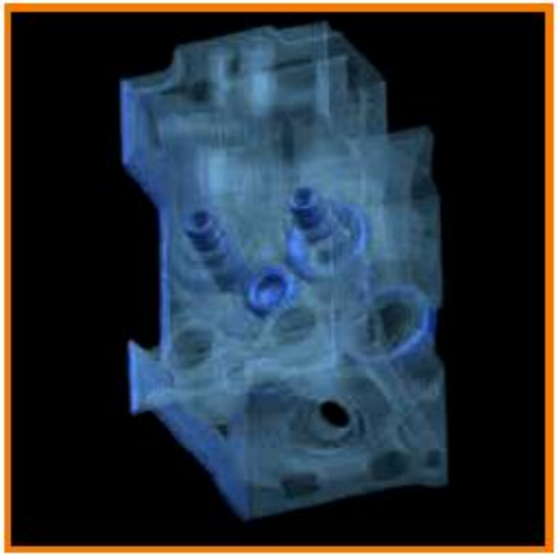


Visible Human
(National Library of Medicine)

Voxel Display

- Ray casting
  - Integrate density along rays trough voxels

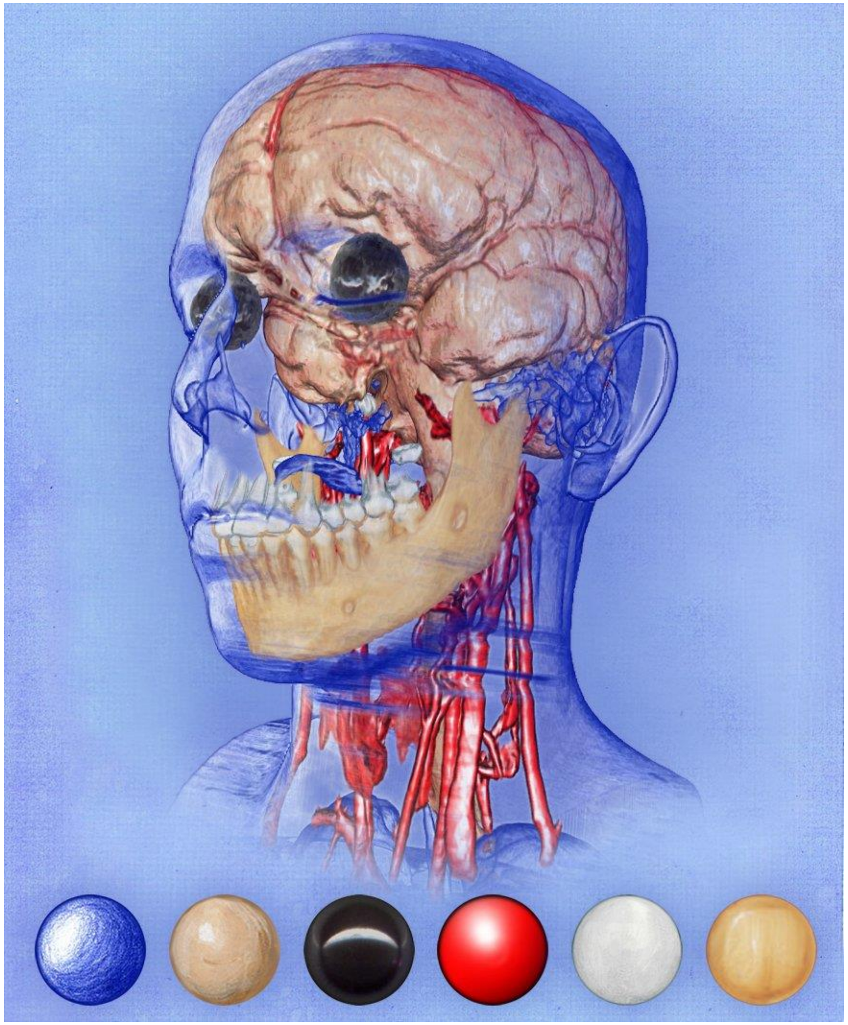




Engine Block
Stanford University

Voxel Display

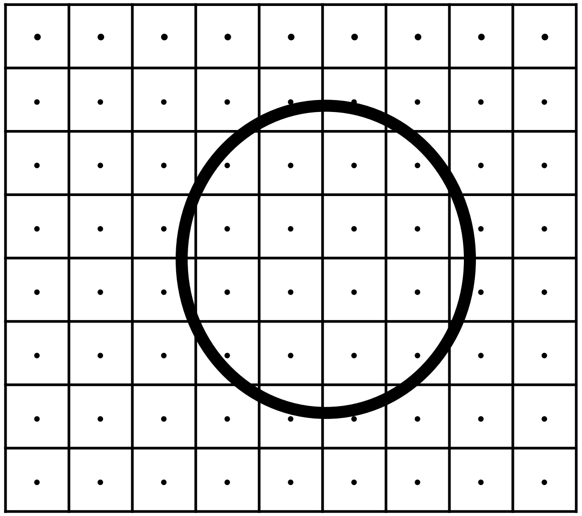
- Extended ray casting
  - Complex transfer functions
  - Map voxel densities to materials
  - Evaluate “normals” at material transitions



Bruckner et al. 2007

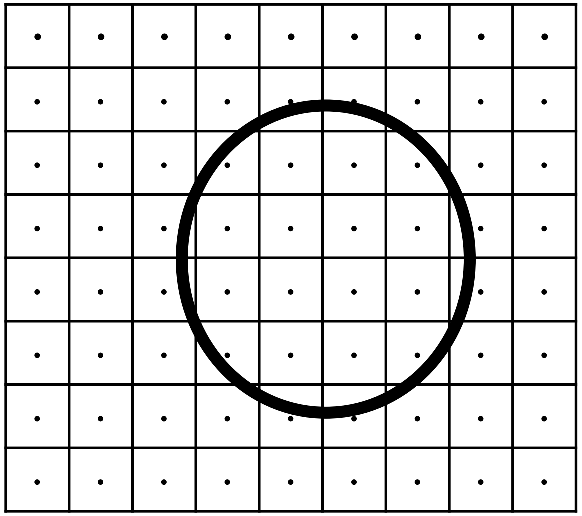
Voxelization: From Surfaces to Voxels

- Binary classification
  - 1: inside the volume, 0: outside
- How?



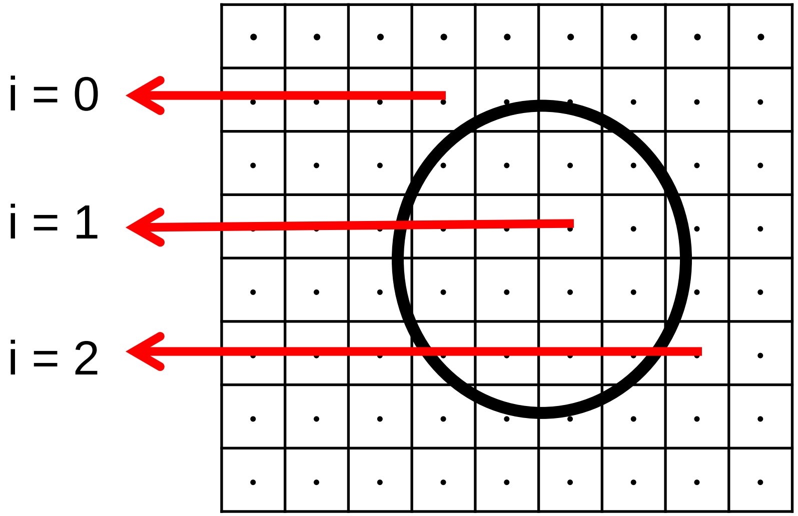
Voxelization: From Surfaces to Voxels

- Common approach (Assignment 1)
  - Ray casting



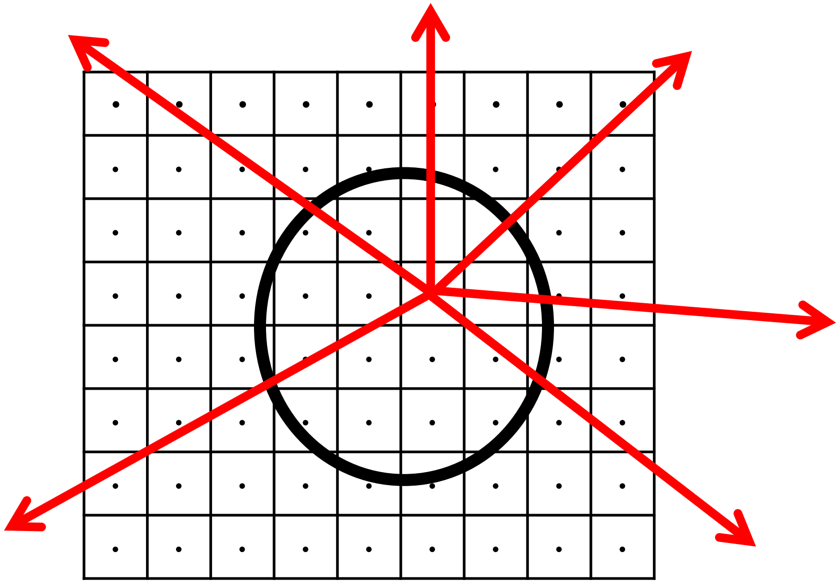
Voxelization: From Surfaces to Voxels

- Ray casting
  - Trace a ray from each voxel center
  - Count intersections
    - Odd: inside
    - Even: outside



Robust Voxelization

- Ray casting
  - Trace many rays in different directions
  - Combine results



Robust Voxelization

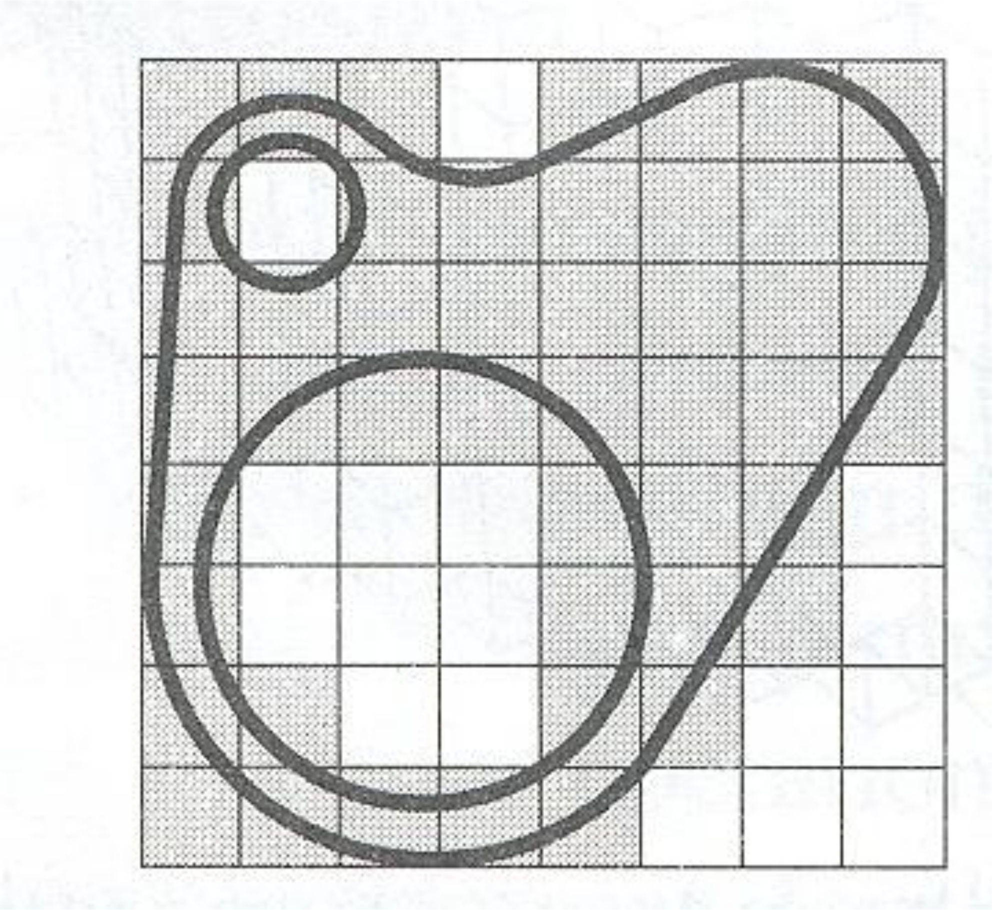
- More on this next class…

Voxels

- Advantages
  - Simple, intuitive, unambigiuous
  - Same complexity for all objects
  - Natural acquisition for some applications
  - Trivial boolean operations
- Disadvantages
  - Approximate
  - Not affine invariant
  - Large storage requirements
  - Expensive display

Voxels

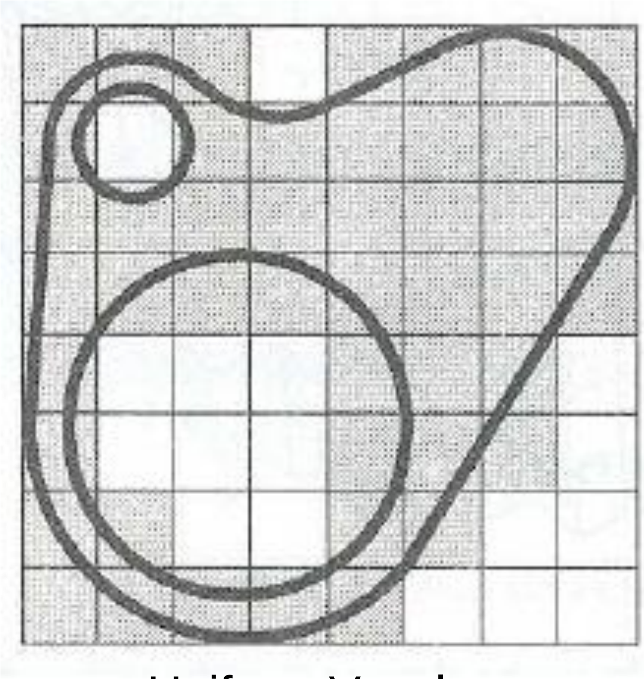
- What resolution should be used?



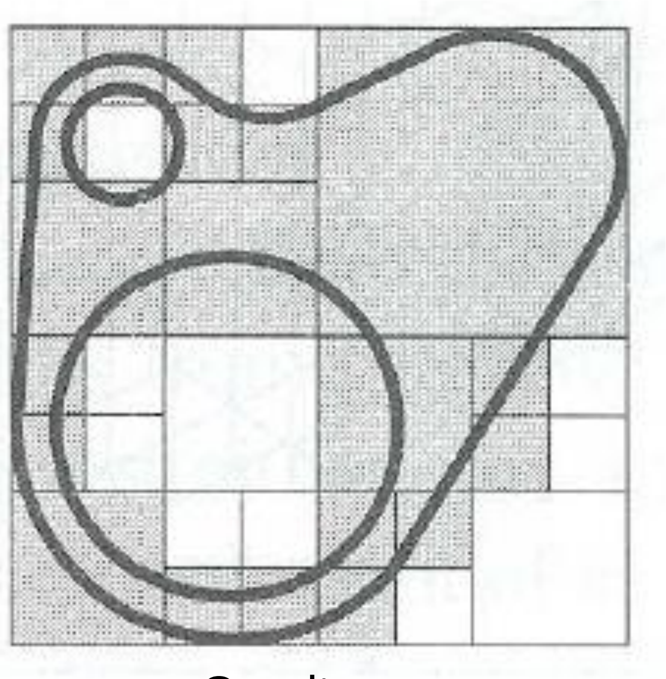
FvDFH Figure 12.21

Octrees

- Refine resolution of voxels hierarchically
  - More concise and efficient for non-uniform objects



Uniform Voxels

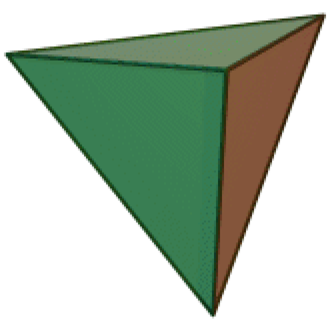


Quadtree

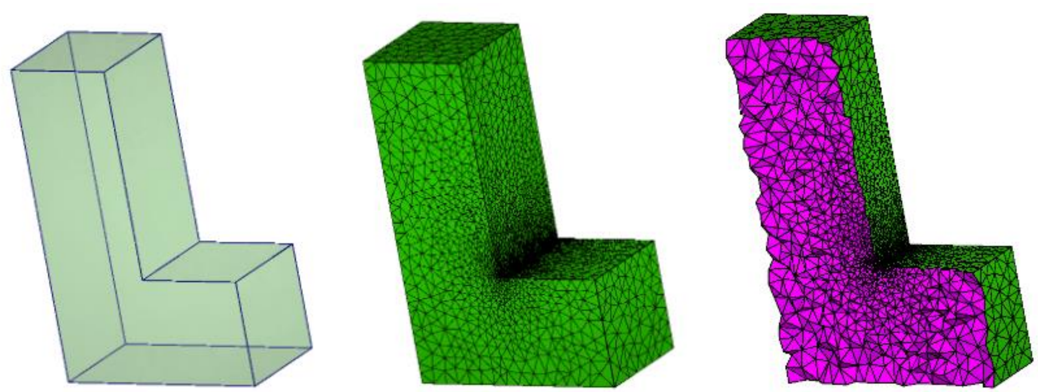
Source: FvDFH Figure 12.20

Tetrahedra as Volume Representations

- Tetrahedron (Tet)
  - Generalization of triangles to volumes
  - 4 vertices, 4 faces
- Tetrahedral mesh (Tet Mesh)
  - Similar to a standard mesh
    - A list of vertices
    - A list of tetrahedra



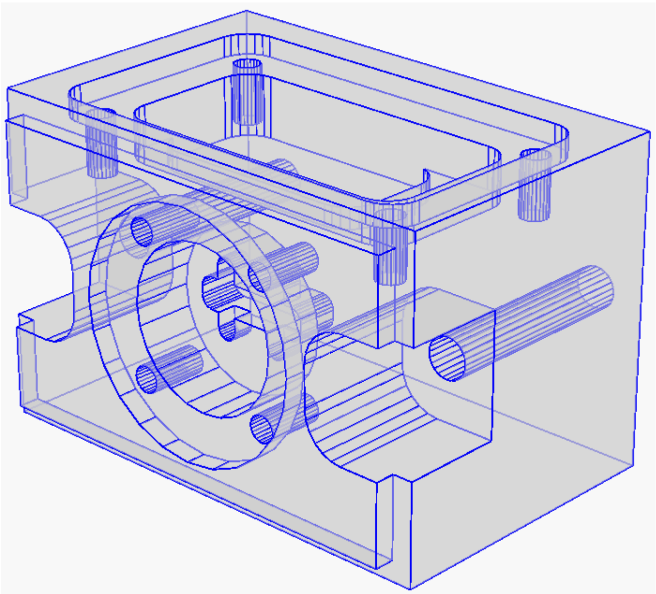
Source: Wikipedia



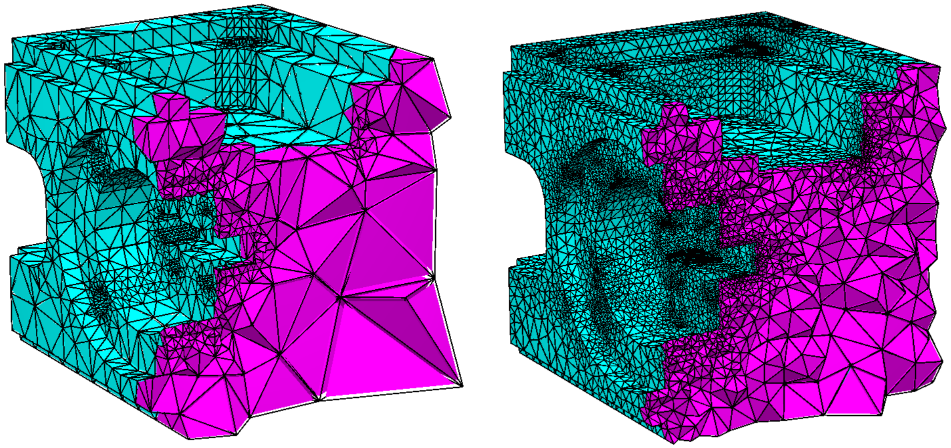
Source: Tetgen.org

Tetrahedralization

- Conversion from a surface representation to a tetrahedral mesh (preserves mesh boundary)
  - Tetgen by Hang Si



Input Mesh



Constrained Delaunay Tetrahedralization

Source: Tetgen.org

That's All For Today

- Readings:
  - Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics
    - http://www.merl.com/publications/docs/TR2000-15.pdf
  - Marching cubes: A high resolution 3D surface construction algorithm
    - http://dl.acm.org/citation.cfm?id=37422
  - Single-pass GPU Solid Voxelization and Applications
    - http://graphics.tudelft.nl/~eisemann/publications/Eisemann2008SolidVoxelization/