CSCI 4560/6560 Computational Geometry

https://www.cs.rpi.edu/~cutler/classes/computationalgeometry/F23/

Lecture 18: Isocontours & Level Sets

Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation

Proposals due Monday Nov 6th

Upload to Submitty

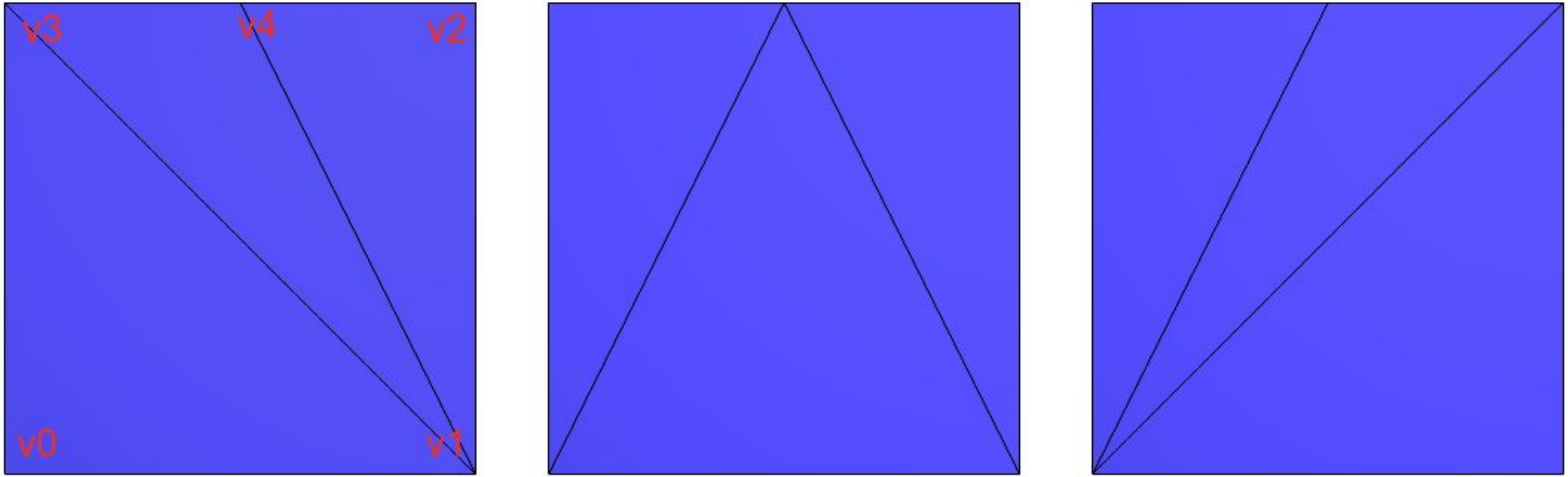
Proposal

As you choose your topic and begin to flesh out the details, keep in mind that implementing new data structures or algorithms can take much longer than anticipated. Also be warned that designing and implementing even relatively simple user interfaces require alot of effort (and is not particularly relevant to this course).

Your proposal should be formatted using pdf. The document should be a minimum of 500 words for an individual project (equivalent of 2 pages double spaced text) or 800 words for a team of two and include:

- A brief summary of the technical problem you are going to investigate.
- A list of the specific research papers and other sources you've collected for background reading. Talk with the instructor if you are unable to find at least 3 relevant academic references. Read and summarize the contributions of each reference and describe how your project relates to this work.
- A timeline for your assignment with a list of the tasks you will execute and *who will do what*. It's ok to list optional tasks that you will work on once the core features are complete. You will be graded relative to the completion of the core tasks, so make sure your plan is feasible.

Homework 7: Delaunay Triangulation Edge Flips

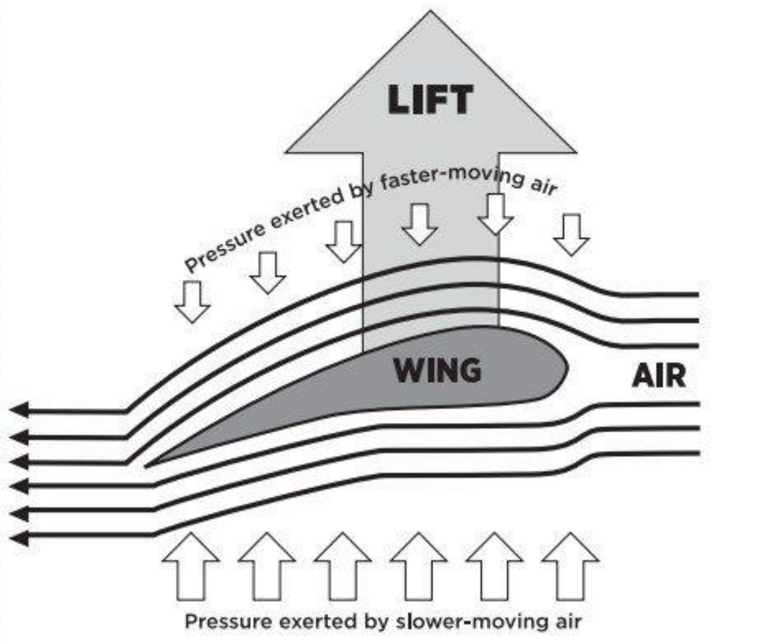



```
v1-v3 -> v0-v4
v1-v4 -> v0-v2
```

Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation

Motivation: Finite Element Modeling (FEM) & Computational Fluid Dynamics (CFD)



https://www.scienceworld.ca/resource/plane-wing-simulator/

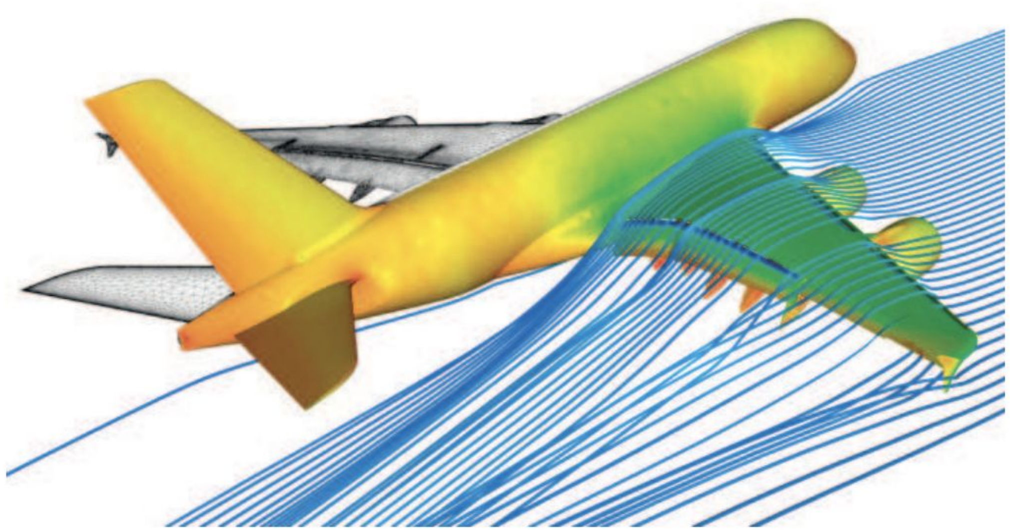
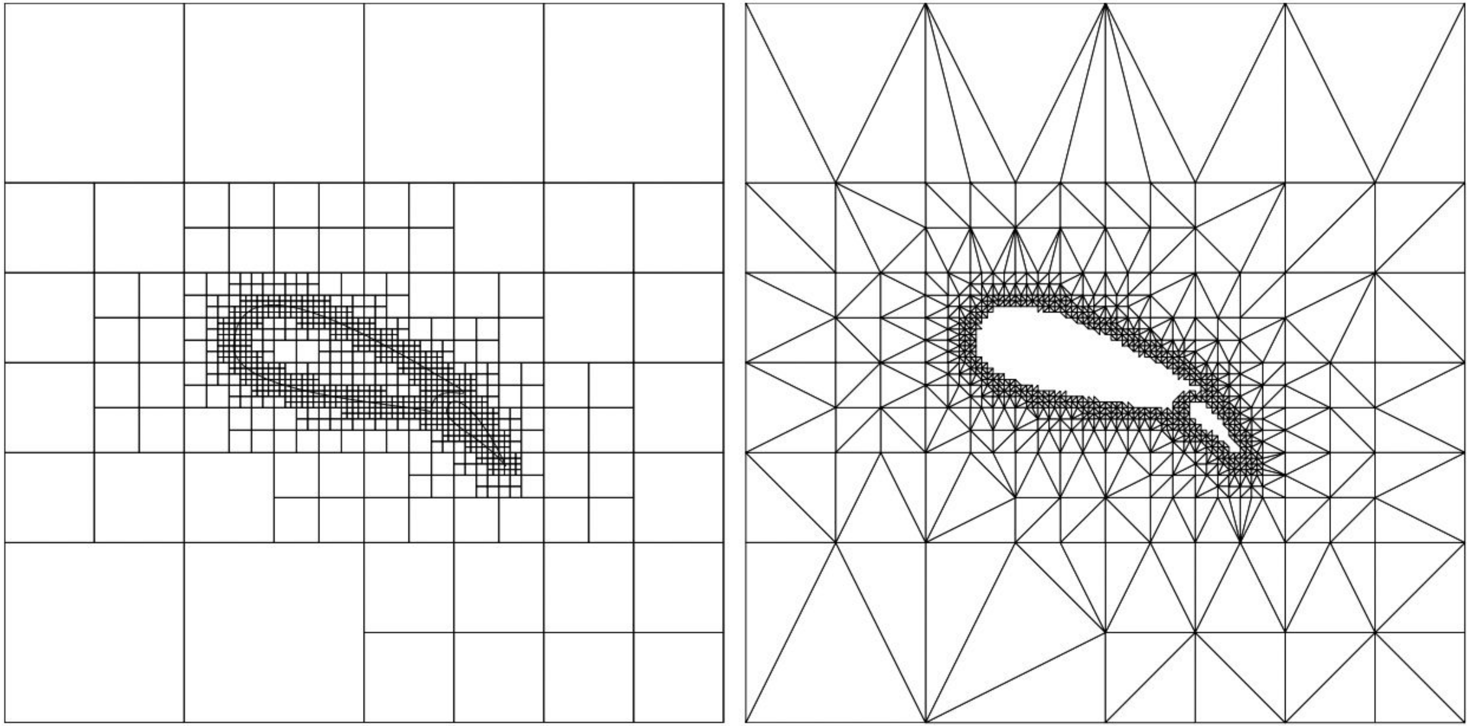


Figure 9: Numerical flow simulation for the Airbus A380 (picture credit: Airbus. Copyright: Dr. Klaus Becker, Senior Manager Aerodynamic Strategies, EGAA, Airbus, Bremen, Germany)

Motivation: Finite Element Modeling (FEM) & Computational Fluid Dynamics (CFD)



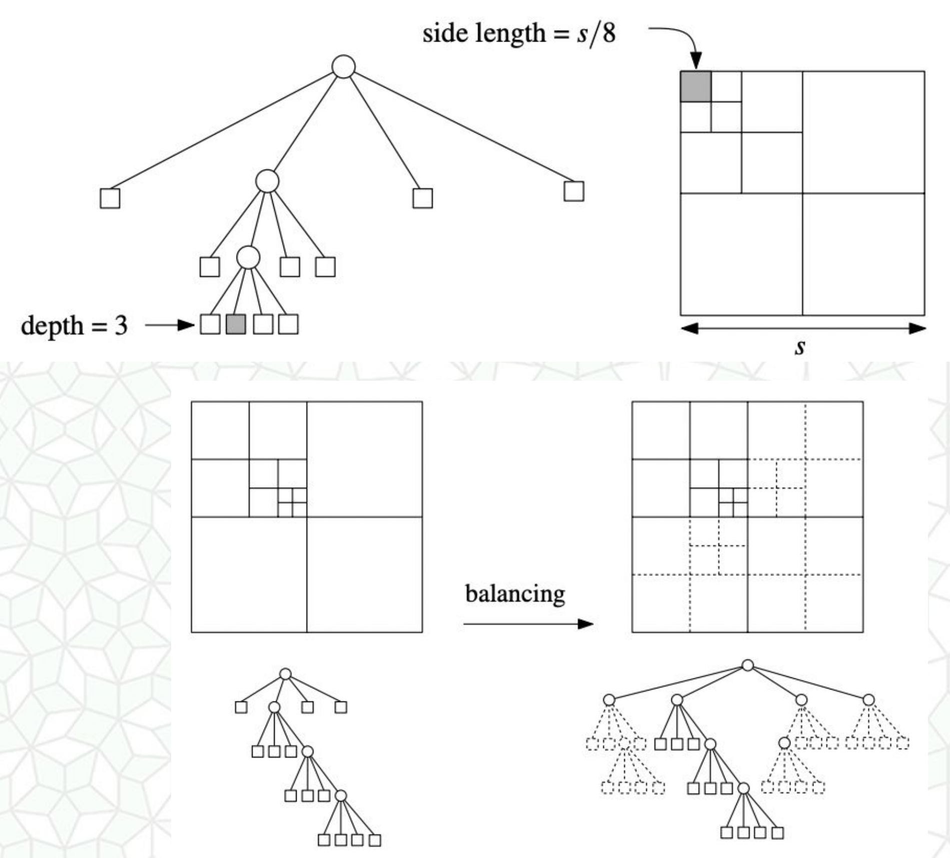
“Delaunay Refinement for Curved Complexes”, Adriano Chaves Lisboa, 2008.

Quad Tree Analysis

- n = # of points
- c = smallest distance between any two points
- s = side length of initial square

- d = depth = $\log(s/c) + 3/2$
- m = # of nodes in unbalanced tree
  = $O((d+1)n)$
- *time to construct* = $O((d+1)n)$

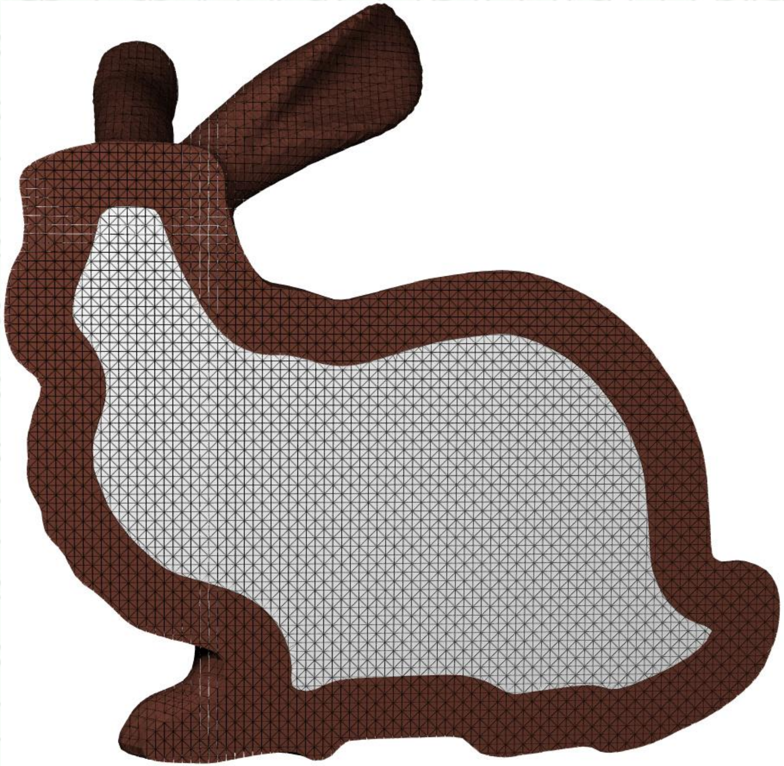
- *# of nodes in balanced tree* = $O(m)$
- *time to balance a tree* = $O((d+1)m)$



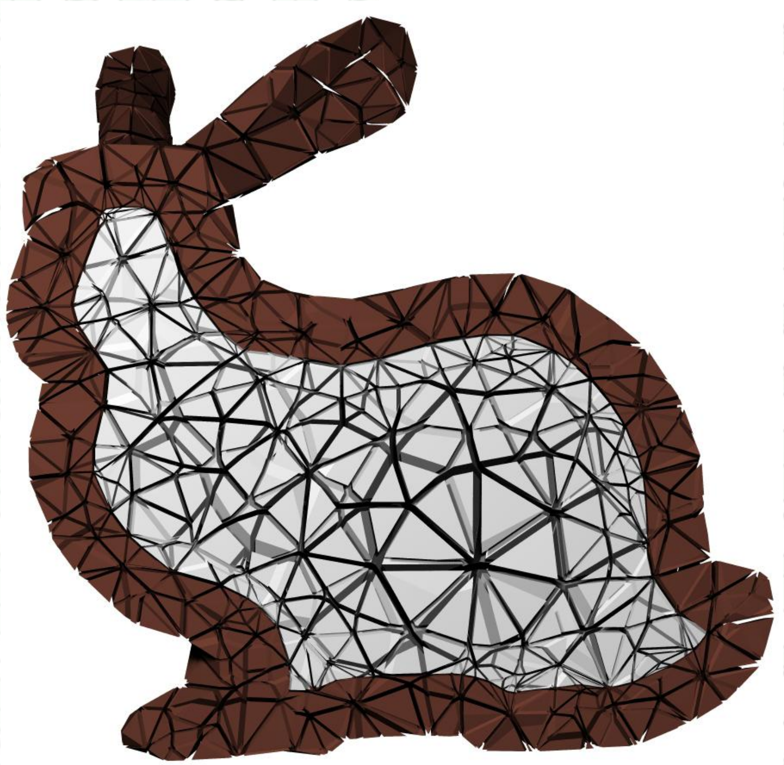
Computational Geometry Algorithms and Applications,
de Berg, Cheong, van Kreveld and Overmars, Chapter 14

3D Mesh Simplification

"Simplification and Improvement of Tetrahedral Models for Simulation"
Cutler, Dorsey, and McMillan
SGP 2004



1,050K tetras
(133K faces)



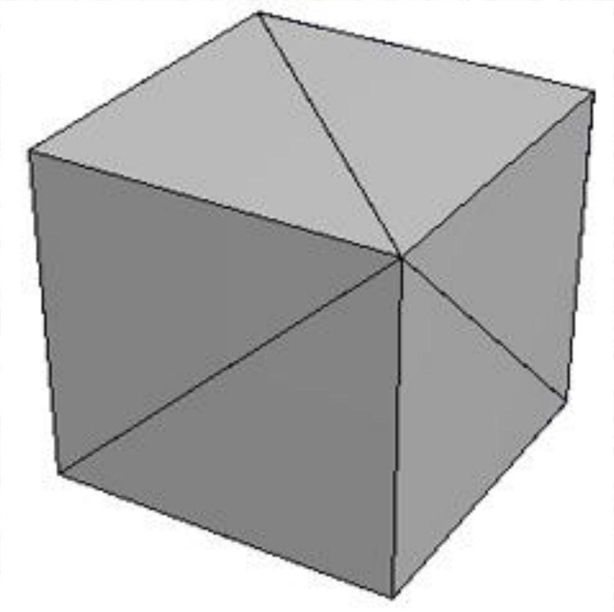
10K tetras
(3K faces)

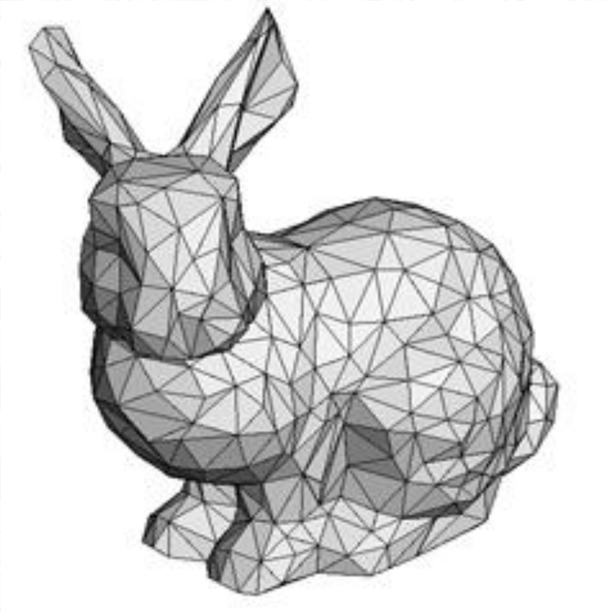
Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation

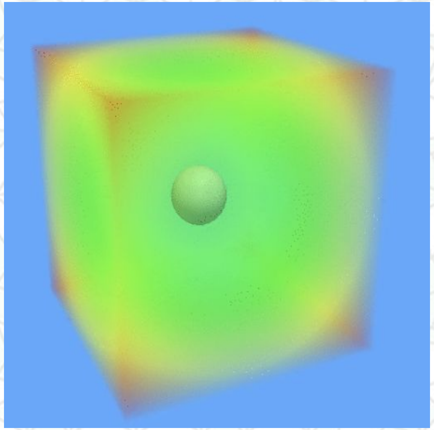
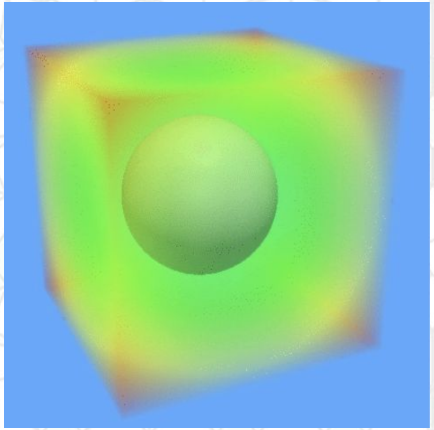
Explicit Surface Mesh Representation

- Often we focus on modeling surfaces with polygon or triangle meshes separating "inside" from "outside"



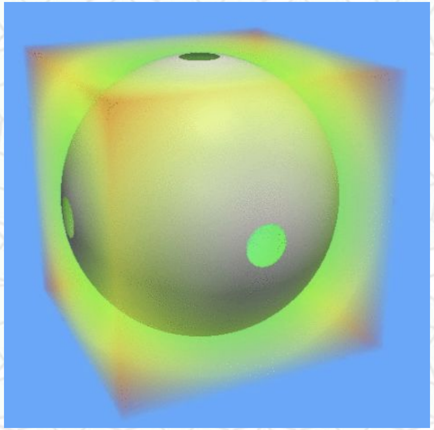
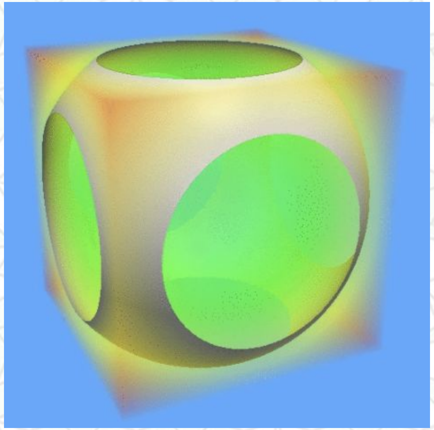
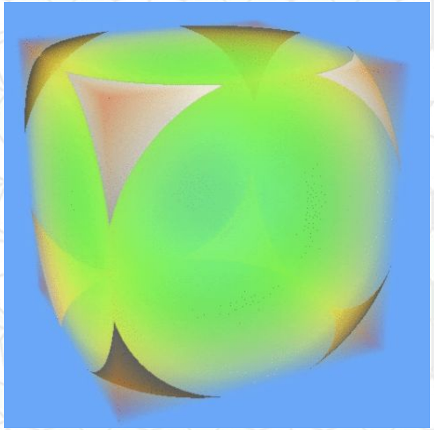


Implicit Surfaces

- Alternately, some objects are easily represented by an equation:
- E.g., a sphere:

  $H(x,y,z) = x^2 + y^2 + z^2 - r^2$

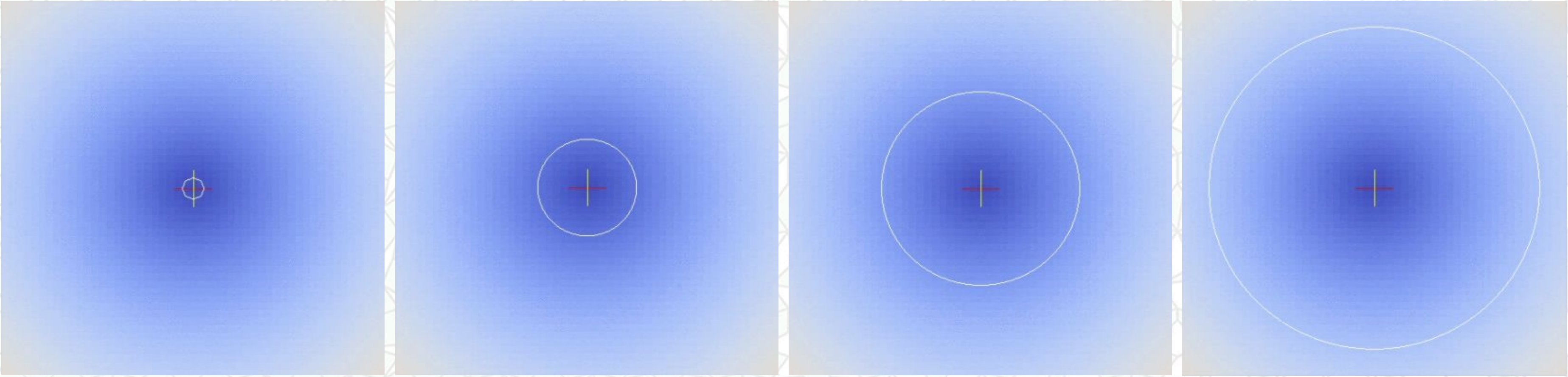




- If $H(x,y,z) = 0$, on surface
- If $H(x,y,z) > 0$, outside surface
- If $H(x,y,z) < 0$, inside surface

Isocontours / Isosurfaces

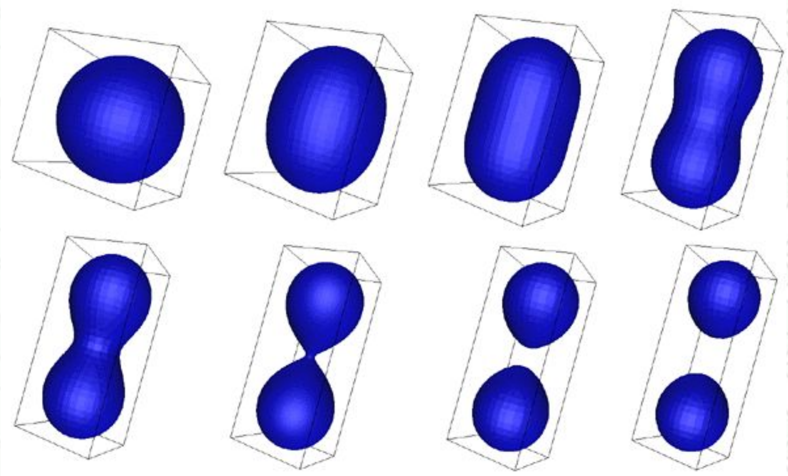
- "iso-" (from Greek word meaning 'equal')
- Everywhere that the data equals a specified value
- E.g., different radii for a circle or sphere centered at the origin

$$H(x,y,z) = x^2 + y^2 + z^2 - r^2$$

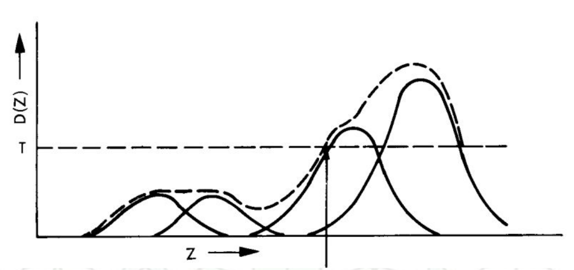


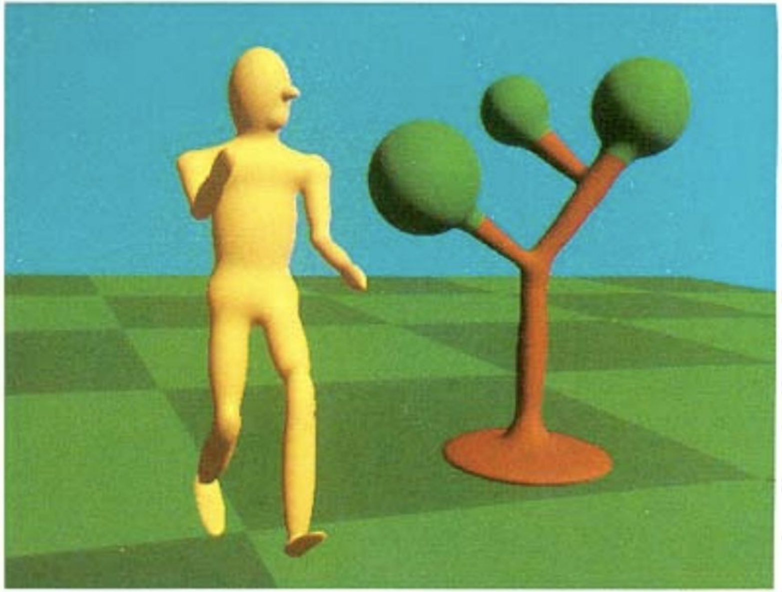
Implicit Surfaces: Blobby Surfaces / Metaballs

- Compact representation to model soft, round objects



http://paulbourke.net/geometry/implicitsurf/index.html



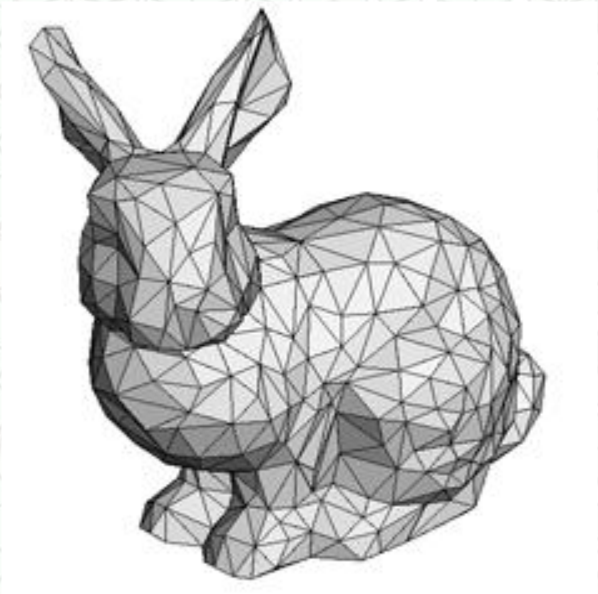


"A Generalization of Algebraic Surface Drawing", Blinn, 1982.

Explicit vs. Implicit Surface Representations

- Some objects can accurately represented either implicitly or explicitly
- Can we convert the bunny mesh into an implicit equation?

  Why might we want to do this?

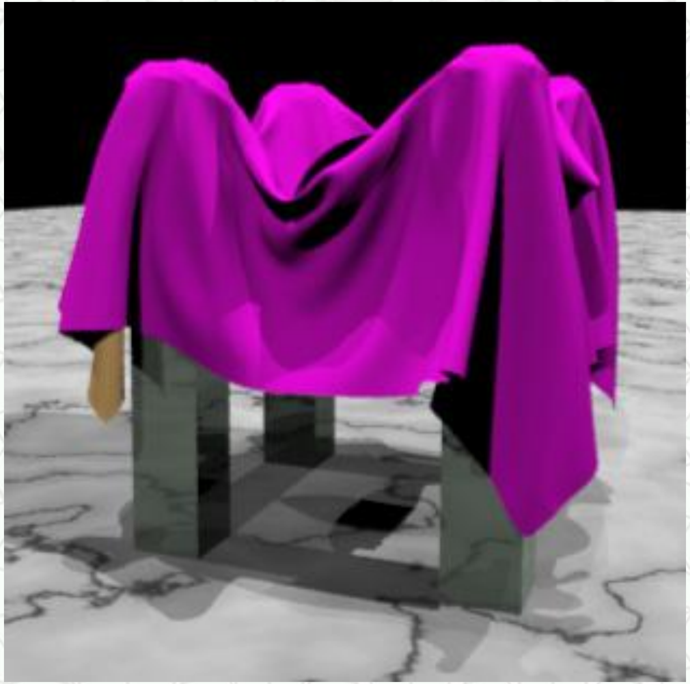


$\longrightarrow$ $H(x,y,z) = ?$

“Robust Treatment of Collisions, Contact and Friction for Cloth Animation”
Bridson, Fedkiw, & Anderson, SIGGRAPH 2002
“Simulation of Clothing with Folds and Wrinkles”,
Bridson, Marino, & Fedkiw, SCA 2003

Motivation: Collision Detection

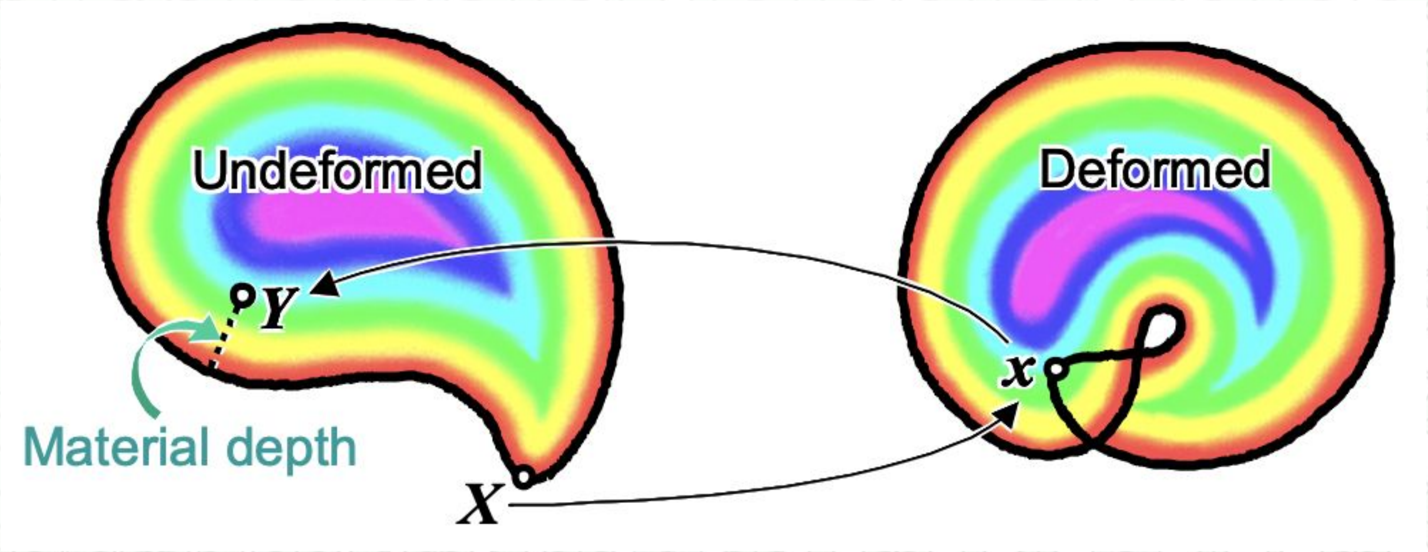
- Detecting Intersections between rigid (or deformable!) objects



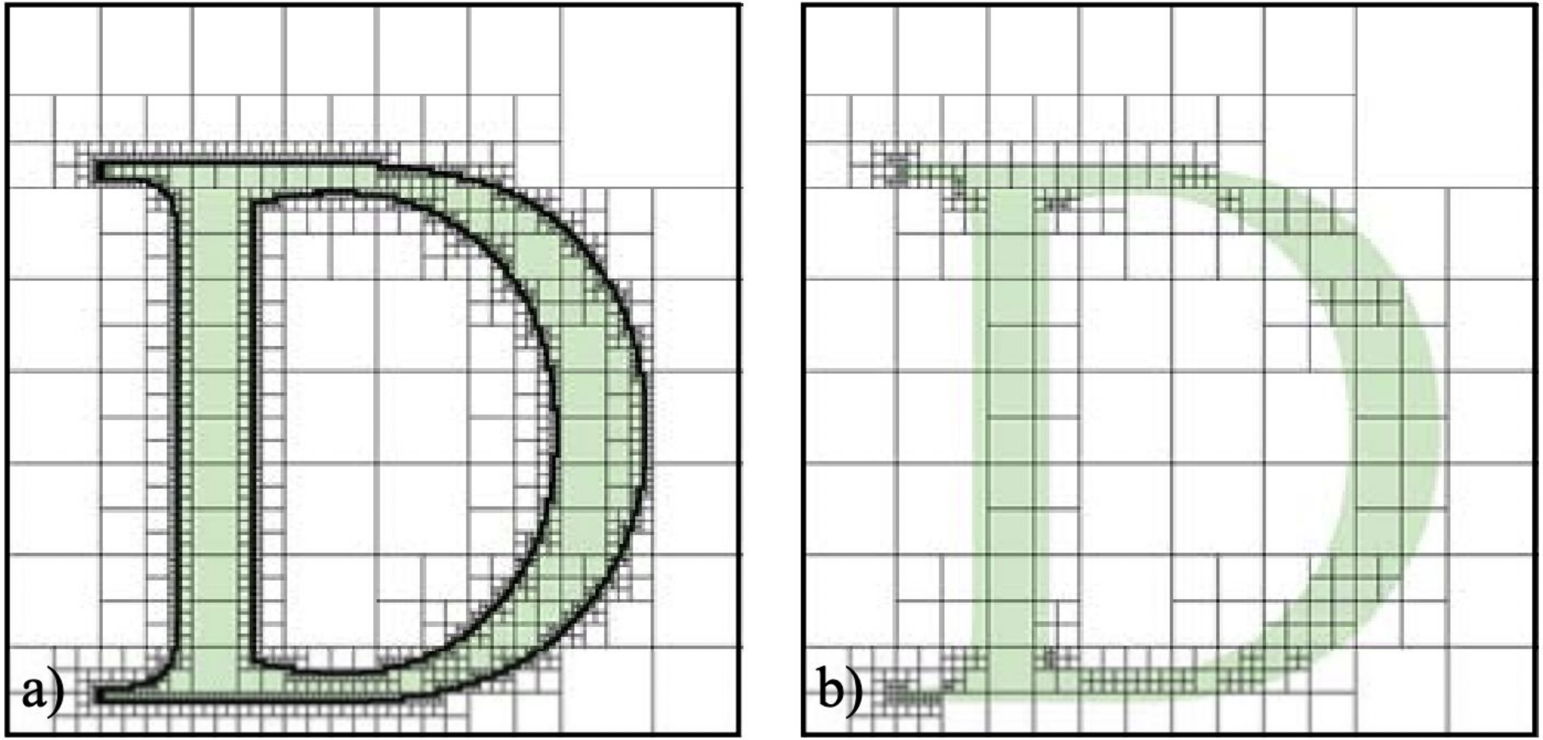
Motivation: Collision Detection

- Detect the intersection
- Depth of intersection penetration
- Gradient & normal of closest surface – Determine penalty force to resolve collision

"An Implicit Finite Element Method for Elastic Solids in Contact", Hirota, Fisher, State, Lee, & Fuchs, SCA 2001

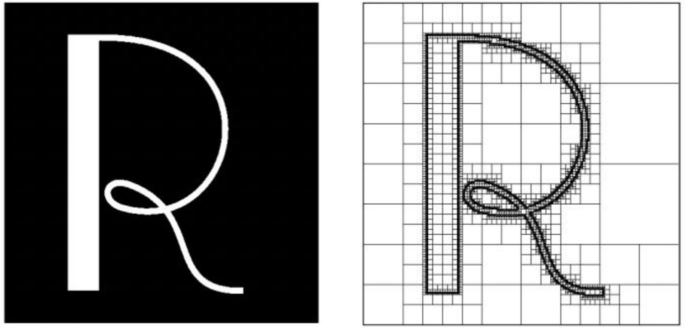


Motivation: Alternate Surface Representation

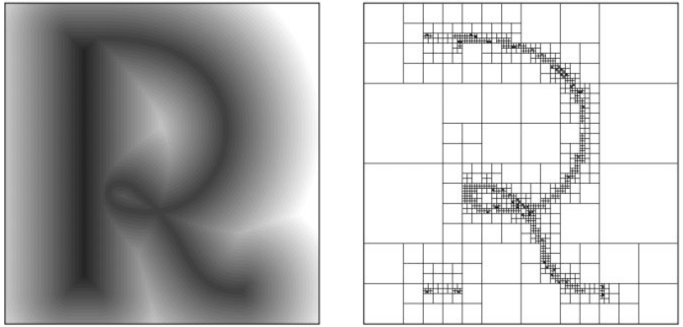


“Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics”,
Frisken, Perry, Rockwood, and Jones, SIGGRAPH 2001
“Designing with Distance Fields”, Frisken and Perry, 2006

Motivation: Surface Sculpting



Figures 4a "R" and **4b** 3-color quadtree containing 23,573 cells.



Figures 4c Distance field of "R" and **4d** ADF containing 1713 cells.

"Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics", Frisken, Perry, Rockwood, and Jones, SIGGRAPH 2001

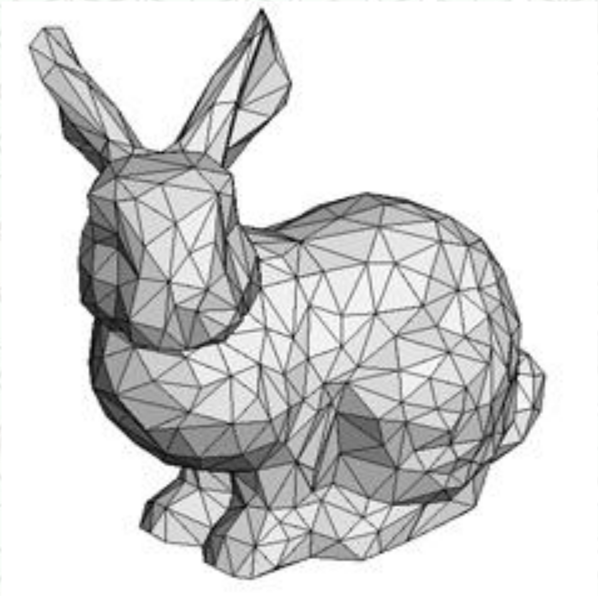
"Designing with Distance Fields", Frisken and Perry, 2006

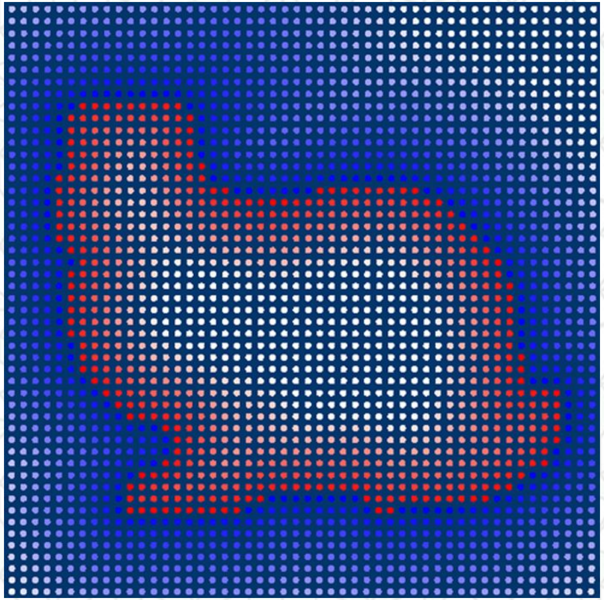
Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation

Explicit vs. Implicit Surface Representations

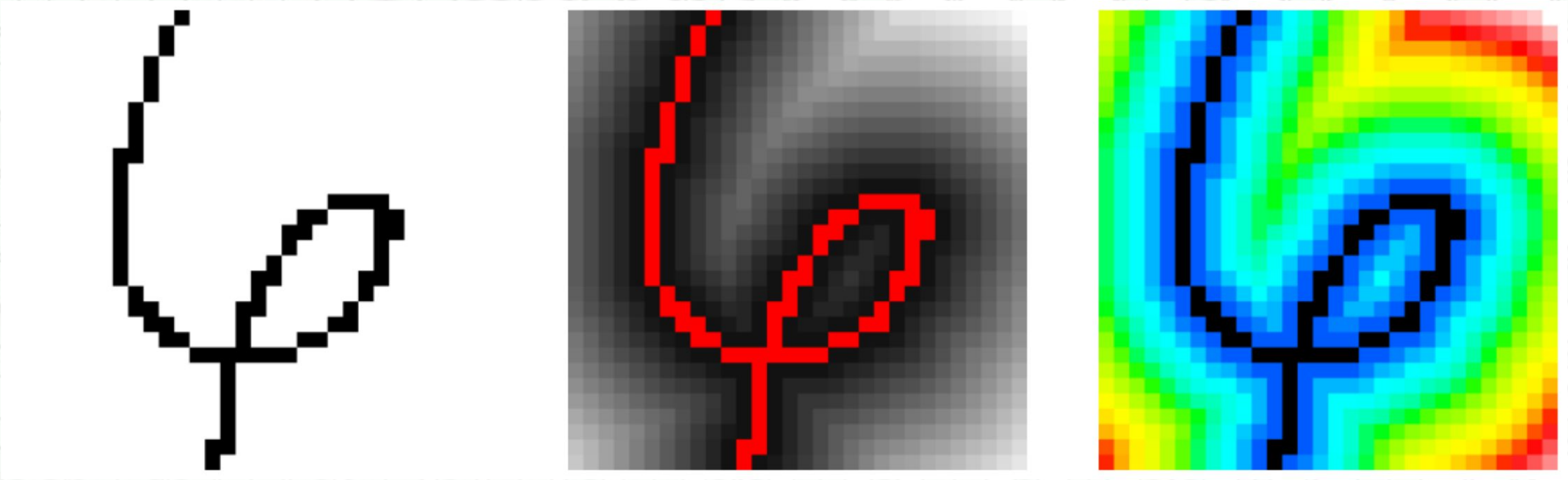
- We may not be able to construct a compact mathematical function…
- But can we convert the bunny mesh into a signed distance field?





Computing a Signed Distance Field

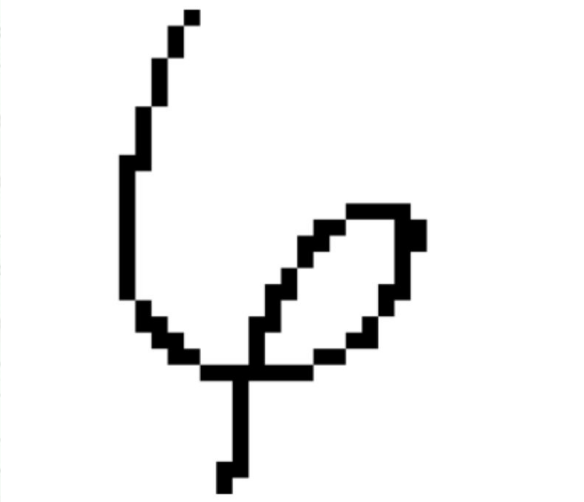
- Given a shape/surface
- Cost to compute shortest distance to original shape for each point (on a grid) in the volume?

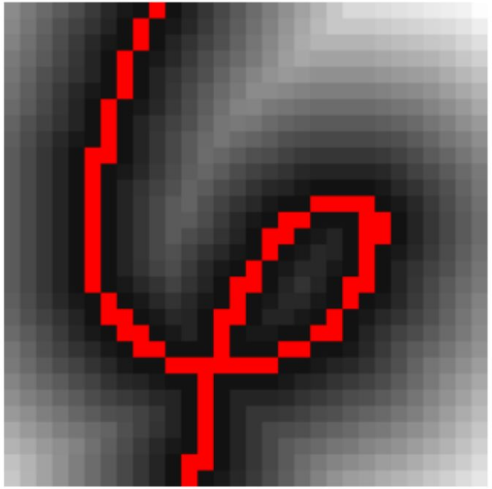


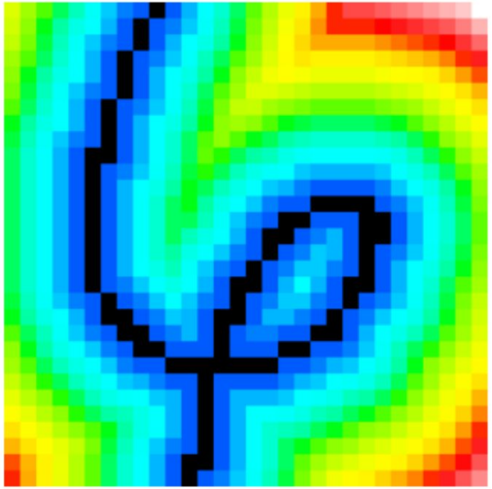
Computing a Signed Distance Field

- Given a shape/surface
- Cost to compute shortest distance to original shape for each point (on a grid) in the volume?

Naive: O(# of volume grid samples * # of surface elements) = $O(w^2h^2)$





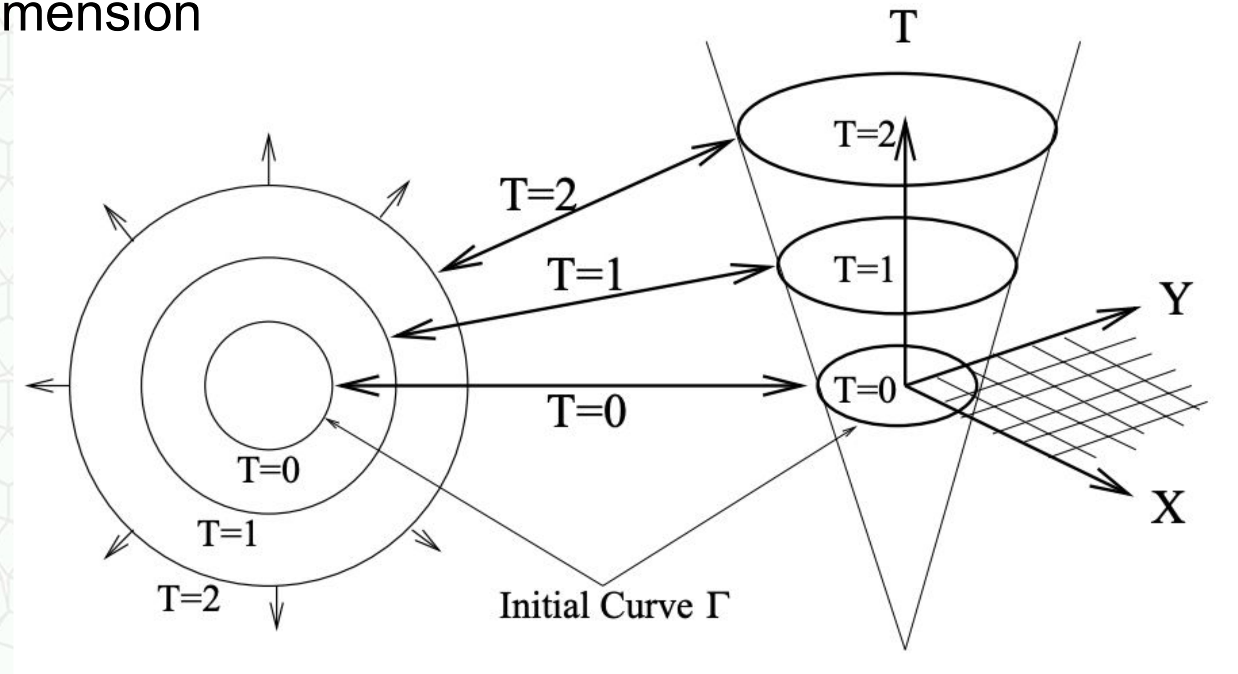


Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation

Level Sets

- For a 2D problem… we can visualize level sets with time (T) as the 3rd dimension

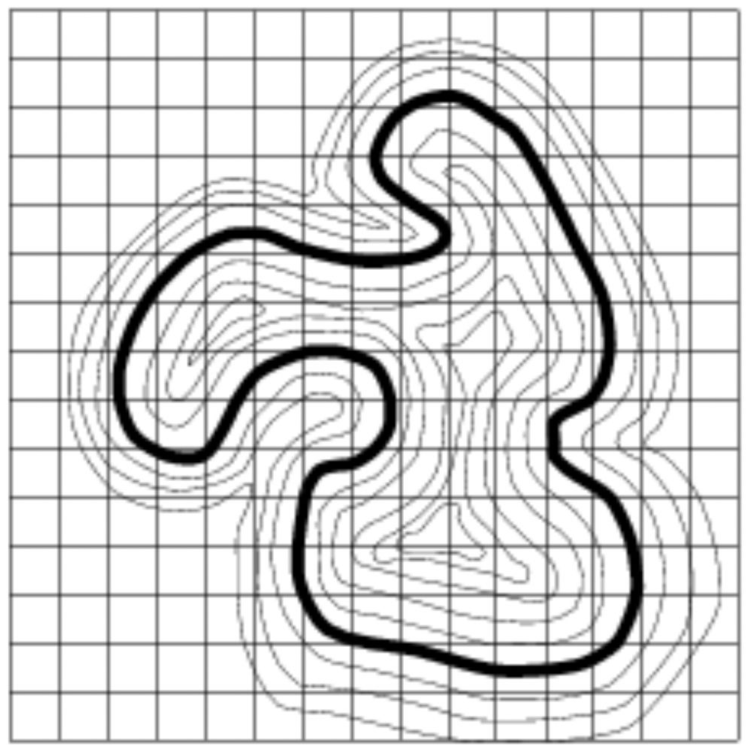


Level Set Methods and Fast Marching Methods, Sethian, 1999

Level Sets - Topology / Connectivity Changes!

- Depending on the application,
  we may want to grow/advance the surface in the outward direction
- Or we may want to shrink the surface in the inward direction

- Sharp corners will round
- Smooth areas may pinch at sharp point

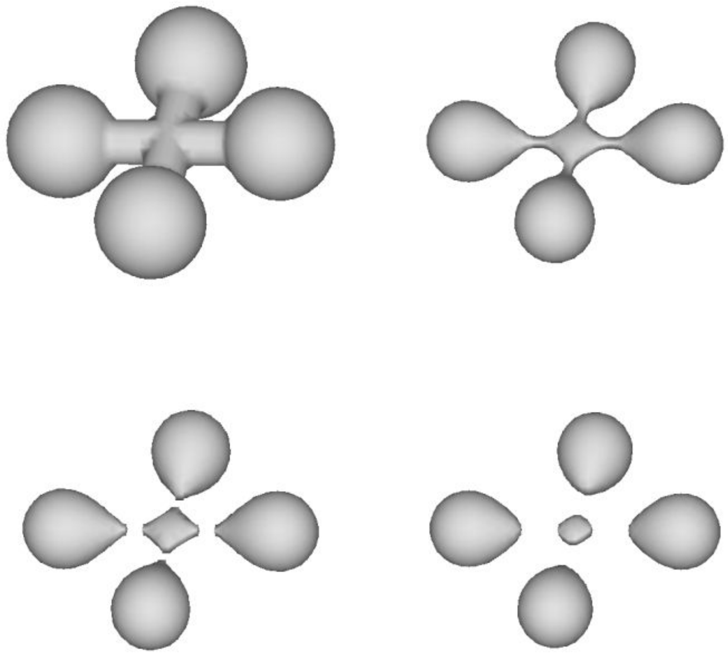


Level Set Methods and Fast Marching Methods, Sethian, 1999

Level Sets - Topology / Connectivity Changes!

- As we trace the level sets the topology of the surface may change!

- The surface may become disconnected
- Disconnected pieces may merge



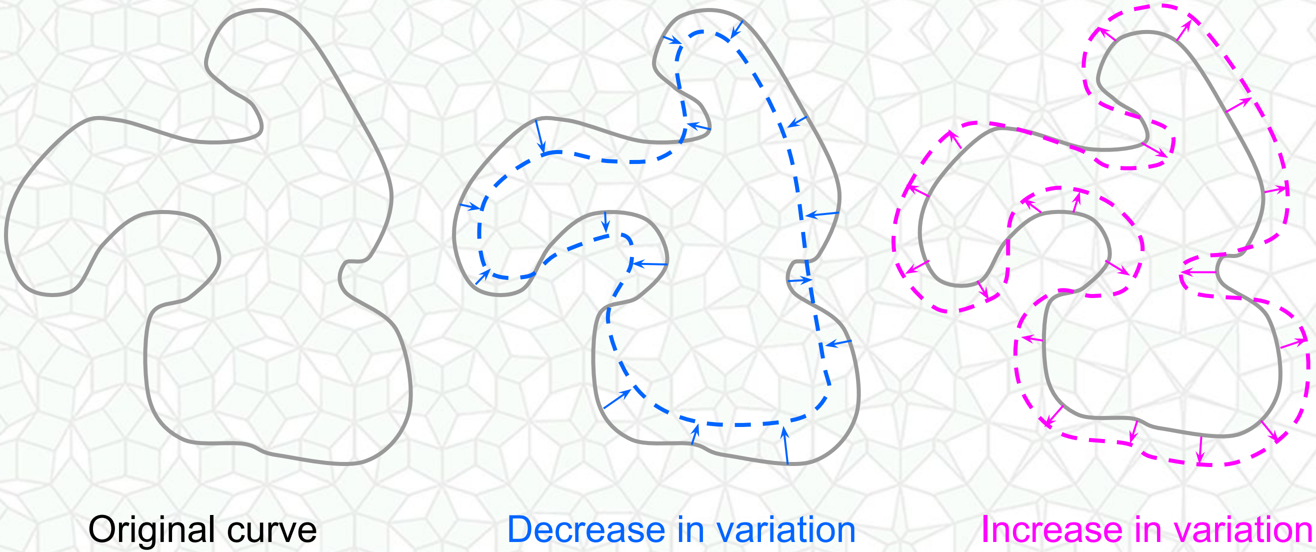
Level Set Methods and Fast Marching Methods, Sethian, 1999

Level Sets - Speed & Direction of Propagation

Depending on the application

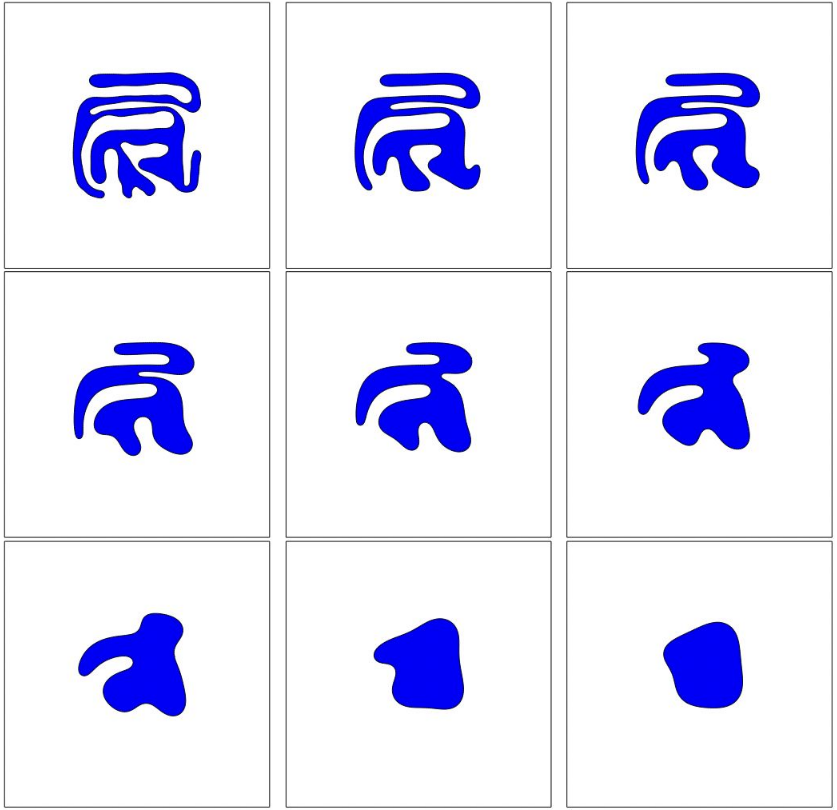
- Speed may not be uniform or constant
- Direction of propagation may be inward and/or outward in different places along the curve/surface!
- And may change over time.

Level Set Methods and Fast Marching Methods,
Sethian, 1999



Level Sets - Topology / Connectivity Changes!

- Locally grow/expand where the curvature is concave
- Locally shrink where the curvature is convex
- All complex curves will collapse to a point!

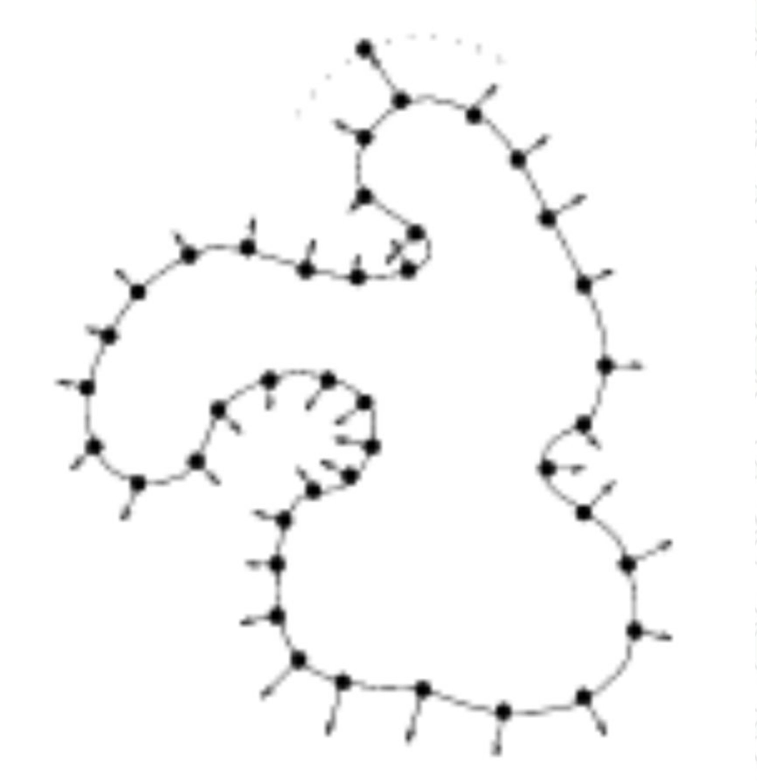


Level Set Methods and Fast Marching Methods, Sethian, 1999

Computing Level Sets / Signed Distance Field

Marker & string method

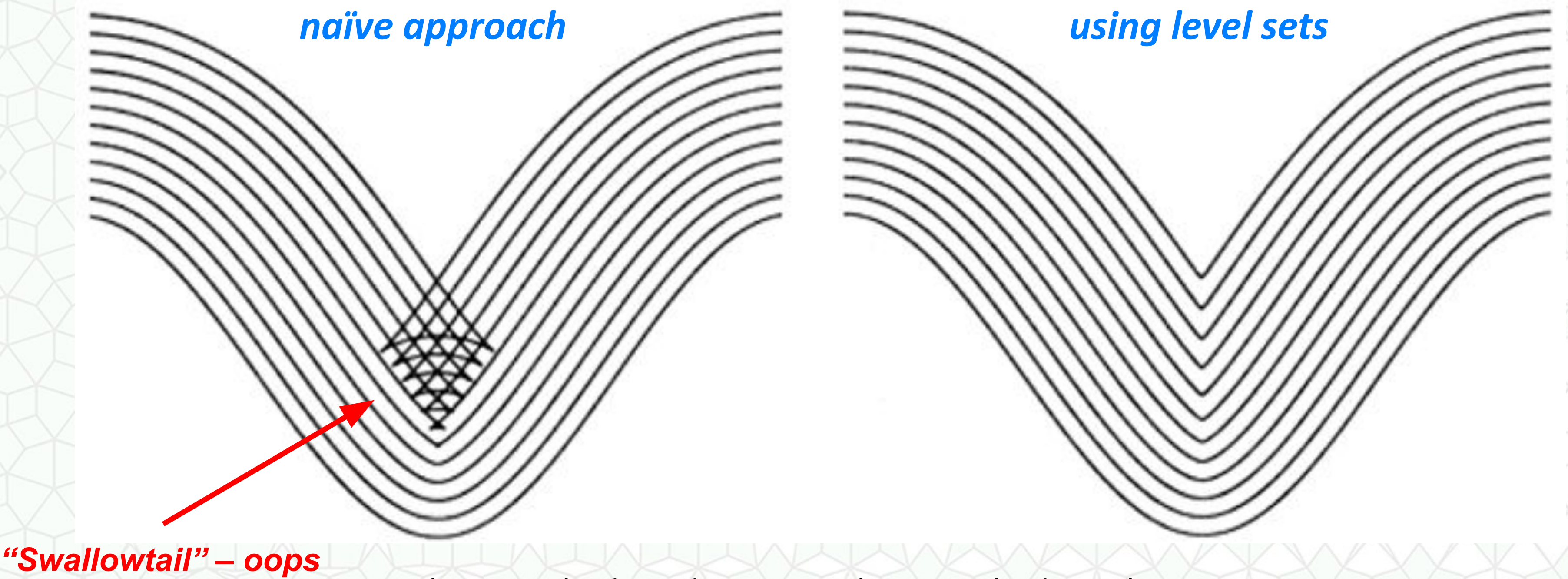
- Copy mesh vertices & edges
- Compute the normal at each vertex (vector perpendicular to the curve)
- Move each vertex a specified distance along the normal
  - *Option: move outward/inward depending on positive/negative curvature*



Level Set Methods and Fast Marching Methods, Sethian, 1999

Computing Level Sets / Signed Distance Field

- Marker & string method: Copy the mesh & move the vertices…



Level Set Methods and Fast Marching Methods, Sethian, 1999

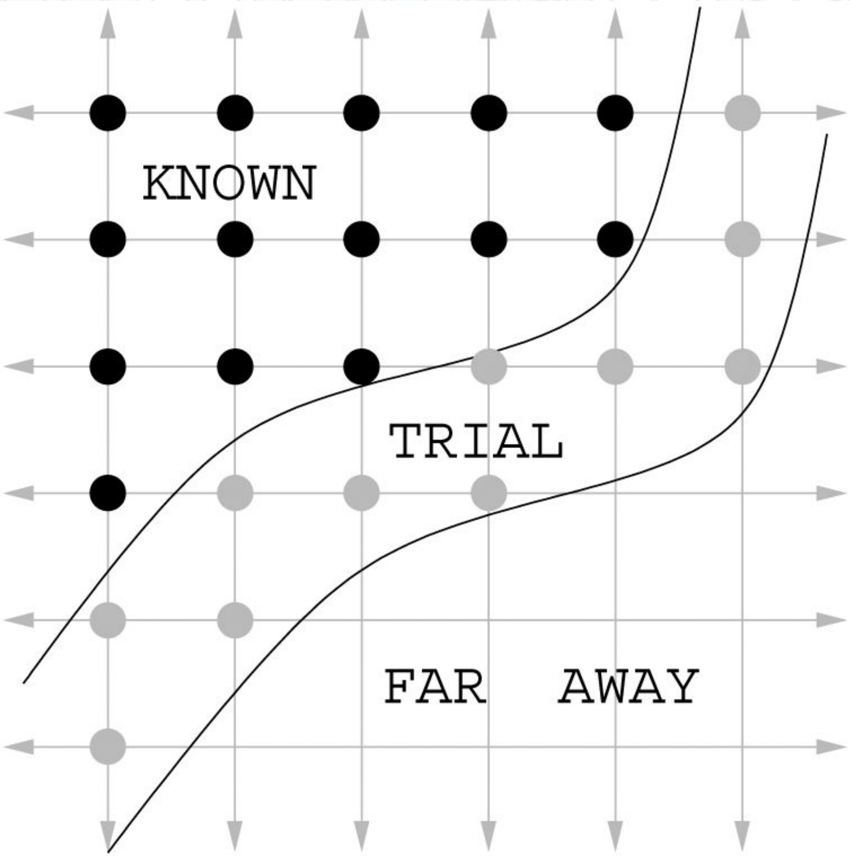
Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation

Fast Marching Method

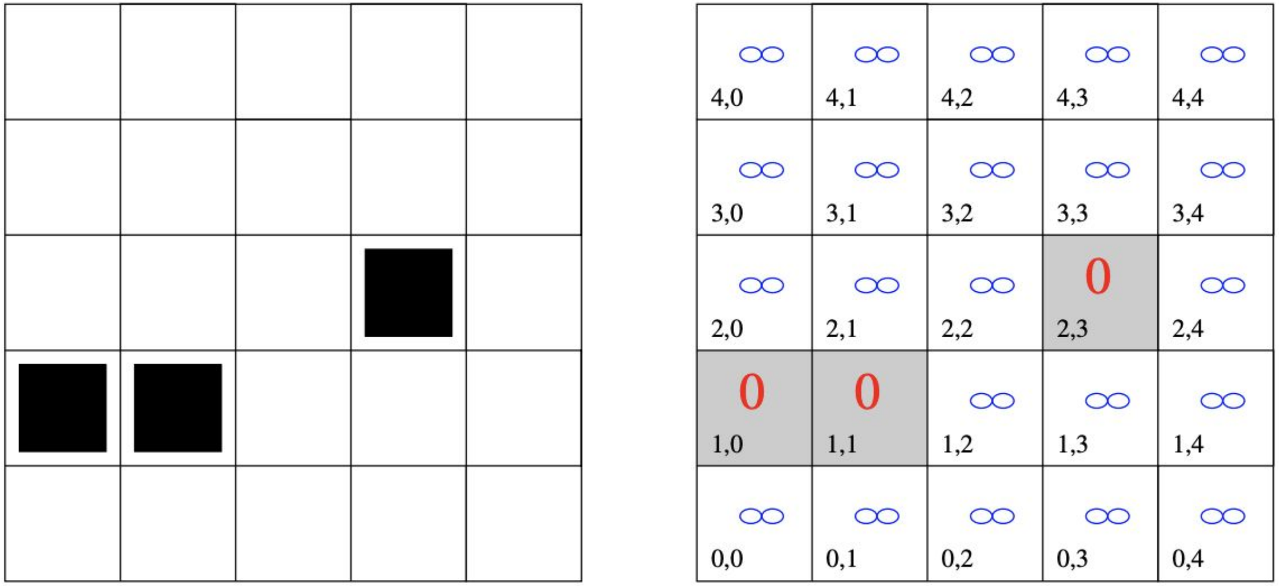
- Efficient method for computing the signed distance field.
- For applications where the front does not *change direction* – it moves outward only (alternately, inward only)

Level Set Methods and Fast Marching Methods, Sethian, 1999



Fast Marching Method Implementation (DS HW!)

Initially, only the surface pixels are "known" to have level set value, a.k.a. distance = 0



input image

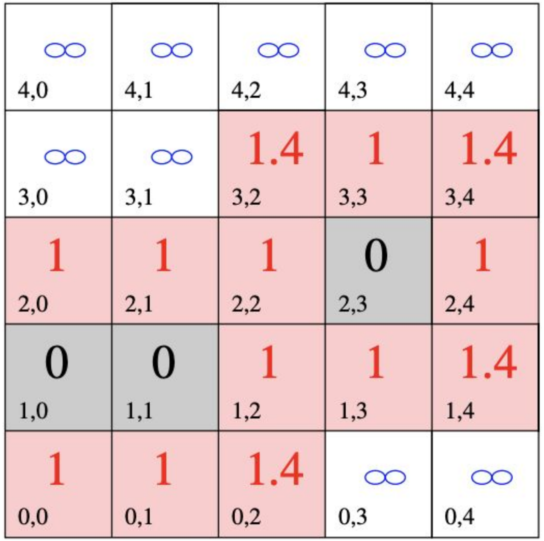
initialization of the signed distance field

https://www.cs.rpi.edu/academics/courses/fall18/csci1200/hw/10_level_sets/hw.pdf

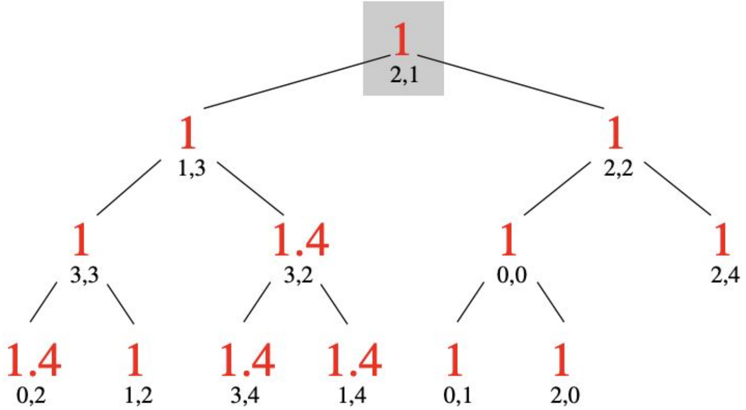
Fast Marching Method Implementation (DS HW!)

We compute the distance of all neighbors of these "known" pixels

Put all these new pixels in a priority queue, ordered by distance



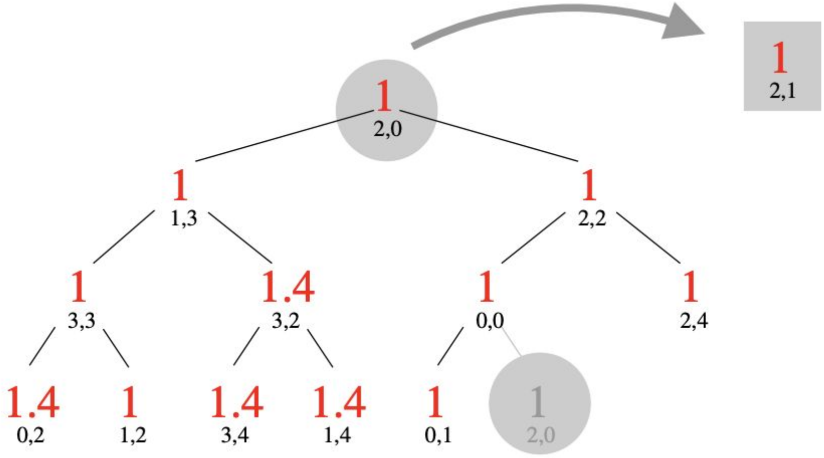
propagating initial values



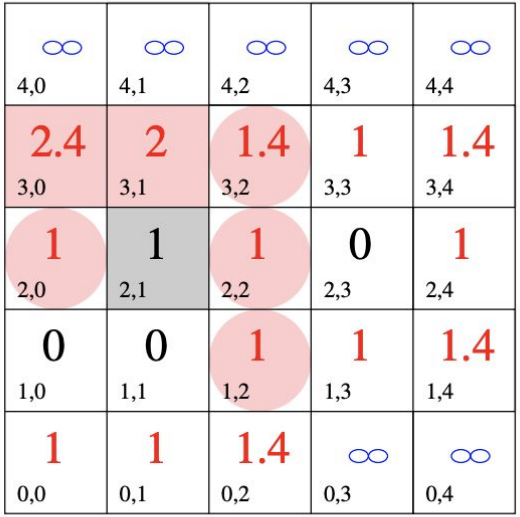
initial priority queue of pixels

Fast Marching Method Implementation (DS HW!)

Grab the top item from the priority queue...



after popping & fixing the top value, grab the last leaf & percolate down



propagate fixed value to neighbors

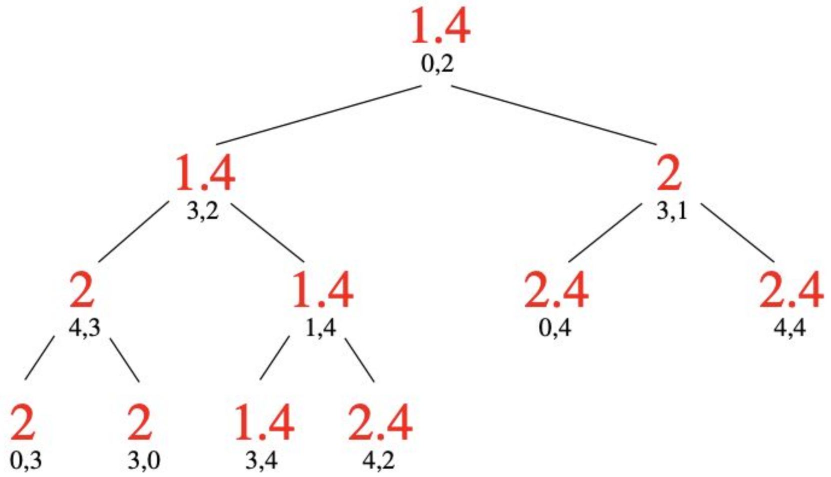
Lock its value, and update its immediate neighbors, update the priority queue

Fast Marching Method Implementation (DS HW!)

Grab the next pixel in the priority queue and repeat….

∞ 4,0	∞ 4,1	2.4 4,2	2 4,3	2.4 4,4
2 3,0	2 3,1	1.4 3,2	1 3,3	1.4 3,4
1 2,0	1 2,1	1 2,2	0 2,3	1 2,4
0 1,0	0 1,1	1 1,2	1 1,3	1.4 1,4
1 0,0	1 0,1	1.4 0,2	2 0,3	2.4 0,4

after fixing all pixels <= 1



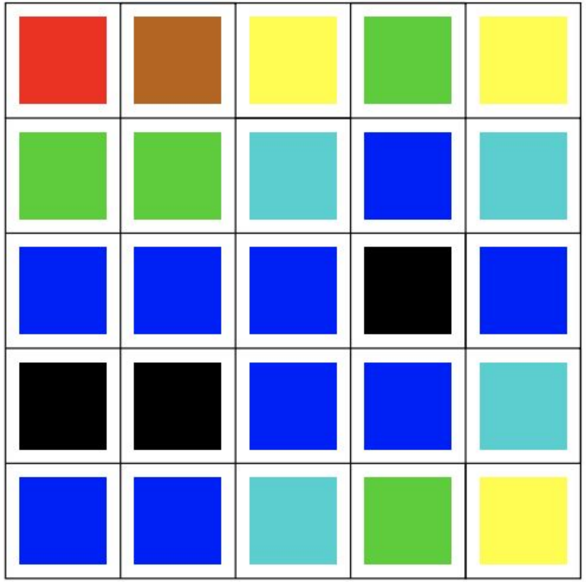
priority queue after fixing all pixels <= 1

Fast Marching Method Implementation (DS HW!)

Final result: Every pixel stores the (approximate) shortest distance to the original surface (black pixels)

3 4,0	2.8 4,1	2.4 4,2	2 4,3	2.4 4,4
2 3,0	2 3,1	1.4 3,2	1 3,3	1.4 3,4
1 2,0	1 2,1	1 2,2	0 2,3	1 2,4
0 1,0	0 1,1	1 1,2	1 1,3	1.4 1,4
1 0,0	1 0,1	1.4 0,2	2 0,3	2.4 0,4

final distance field



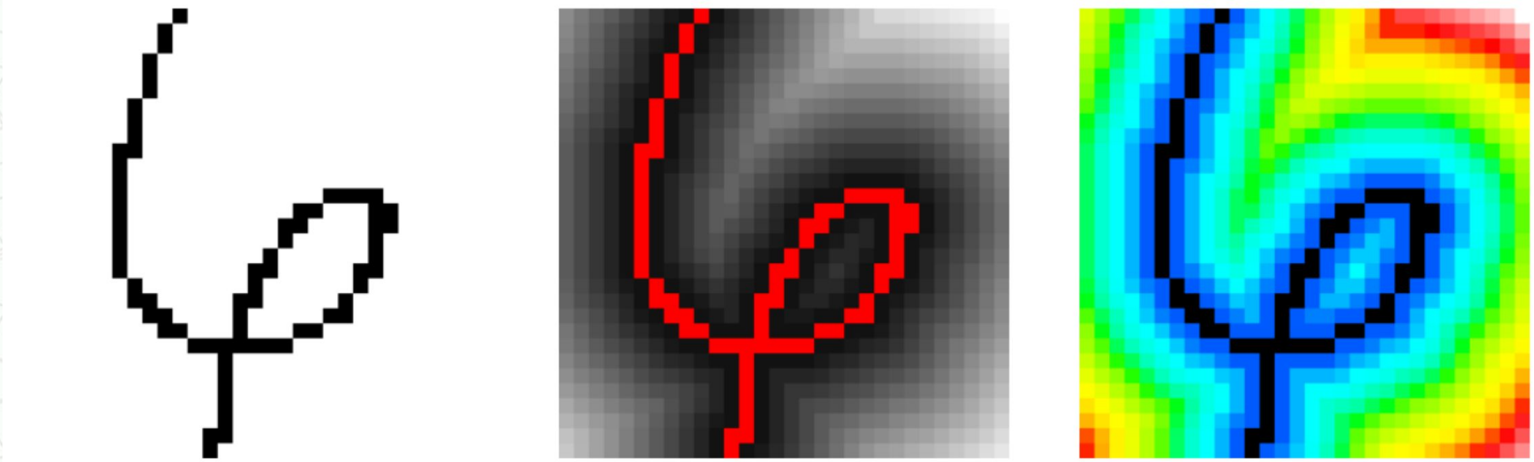
output image

Analysis of Fast Marching Method

- For an image/grid of size *w* x *h*, with *t* pixels/triangles:
- Naive:

  → *O (# of volume grid samples * # of surface elements)* = $O(w^2h^2)$
- Fast Marching:

  →

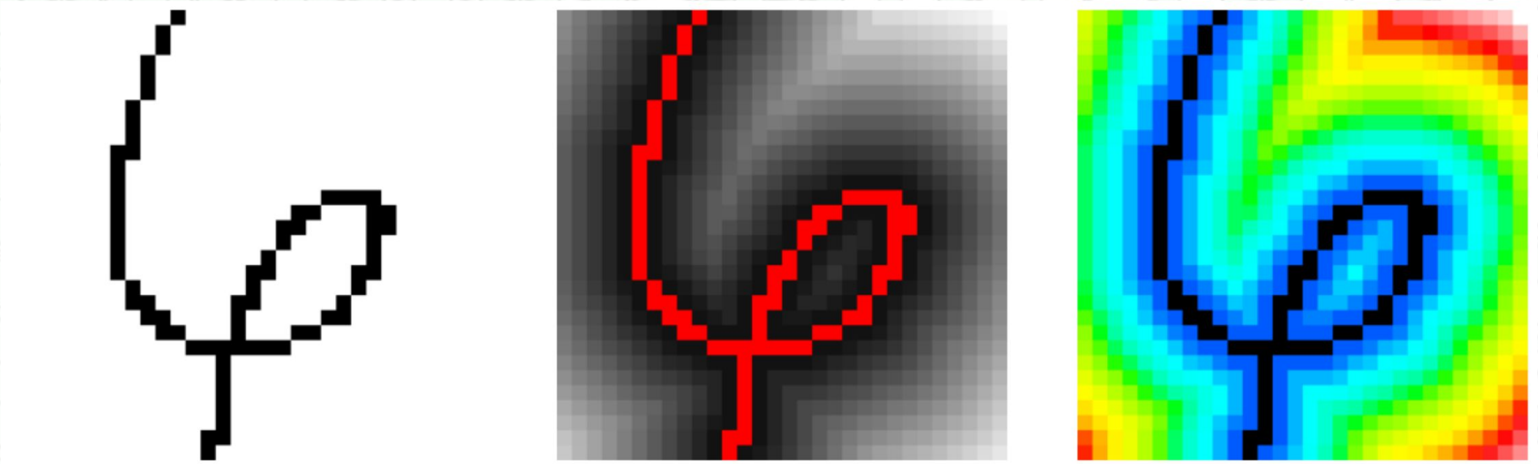


Analysis of Fast Marching Method

- For an image/grid of size *w* x *h*, with *t* pixels/triangles:
- Naive:

  → *O (# of volume grid samples * # of surface elements) =* $O(w^2h^2)$
- Fast Marching:

  → *O (# of volume grid samples * log active front) = O (w*h * log(t))*

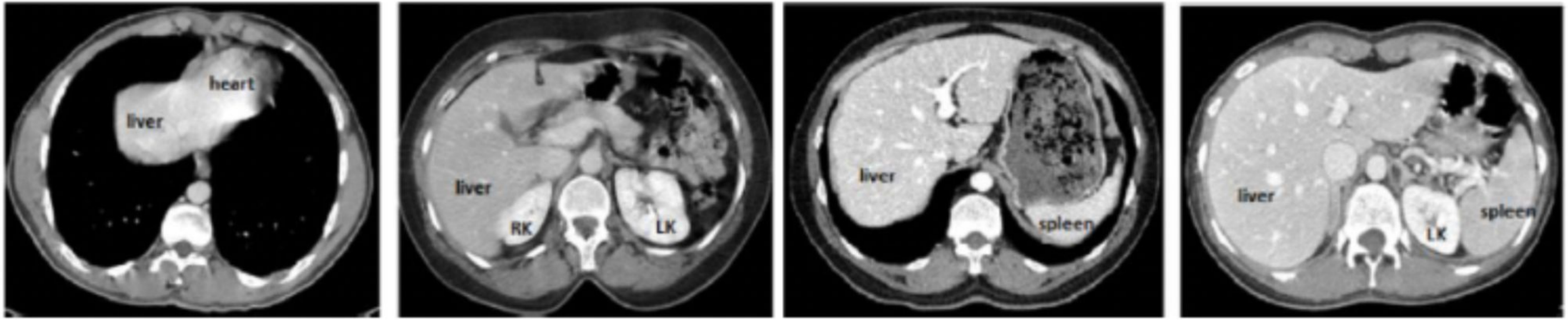


Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation

Motivating Application: Medical Imaging

- Problem Statement: Convert 2D slices of MRI or CT image data into a 3D triangle mesh of the different organs and structures
- This will facilitate more intuitive visualization

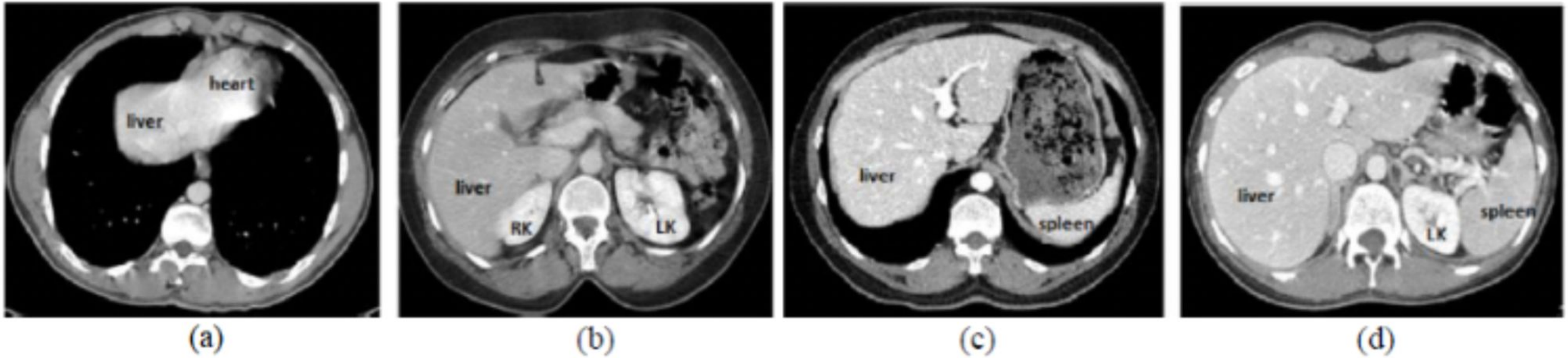


(a) (b) (c) (d)

https://chaos.grand-challenge.org/Data/

Motivating Application: Medical Imaging

- Input: a stack of 2D images, closely spaced parallel “slices” of the 3D object
- Step 1: Segment the different regions (by density / color / texture)
- Step 2: *Marching Cubes!*



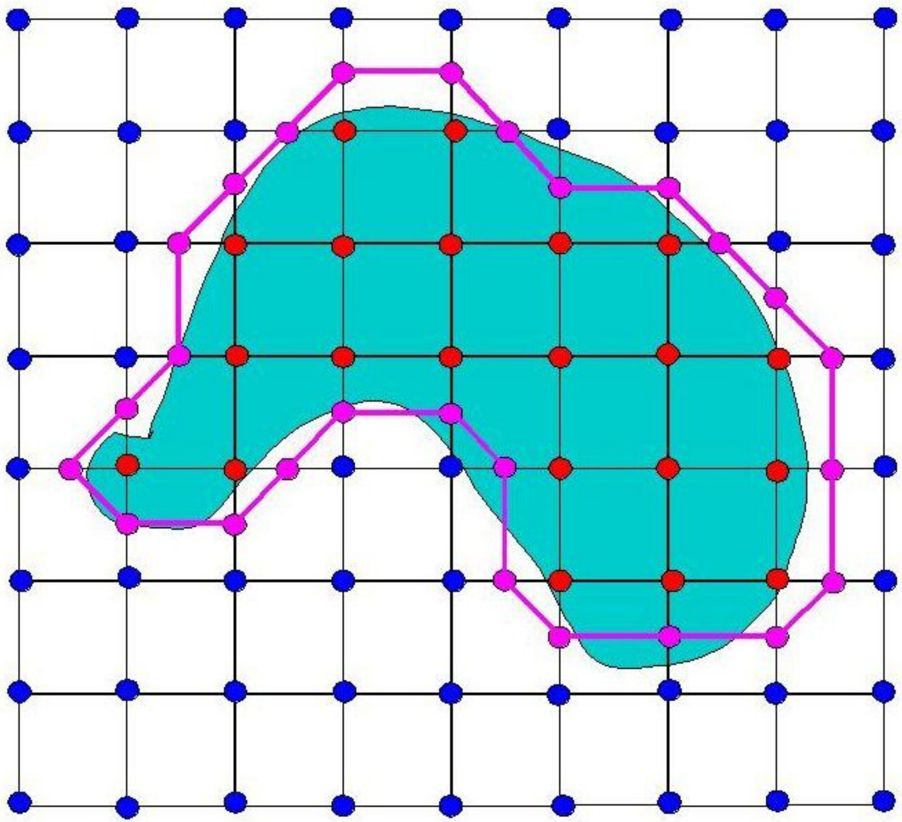
https://chaos.grand-challenge.org/Data/

Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation

Marching Cubes

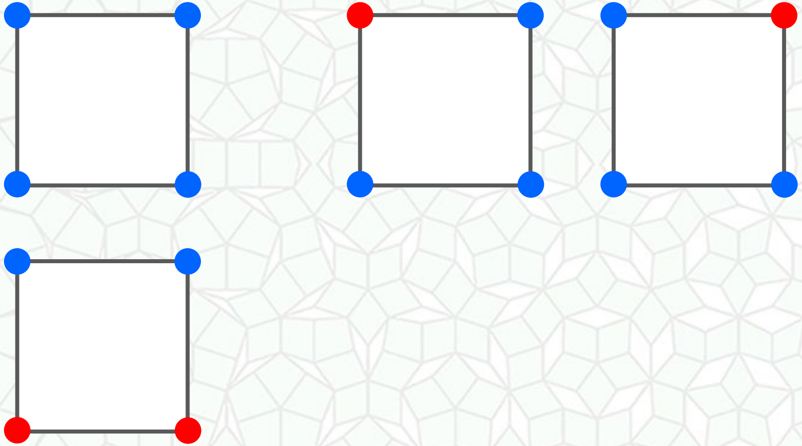
- Each point in the 3D grid is labeled "inside" (red dots) or "outside" (blue dots) the unknown surface.
- Any cell in the grid that has at least one red vertex and at least one blue vertex, must be crossed by the unknown surface.
- We can piecewise construct an approximation of the surface.



http://www.cs.carleton.edu/cs_comps/0405/shape/marching_cubes.html

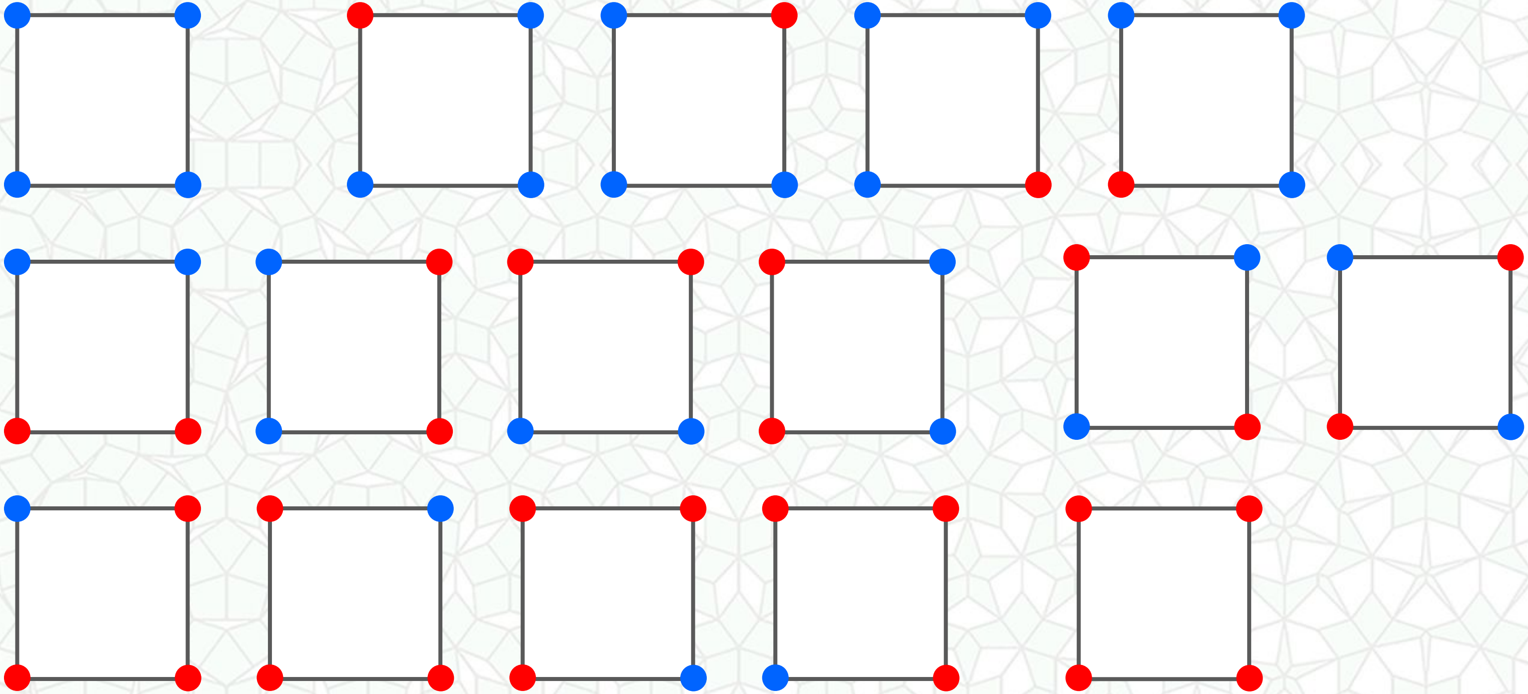
Enumerate Cases in 2D?

- How many cases?
- How many unique cases (excluding rotations)?



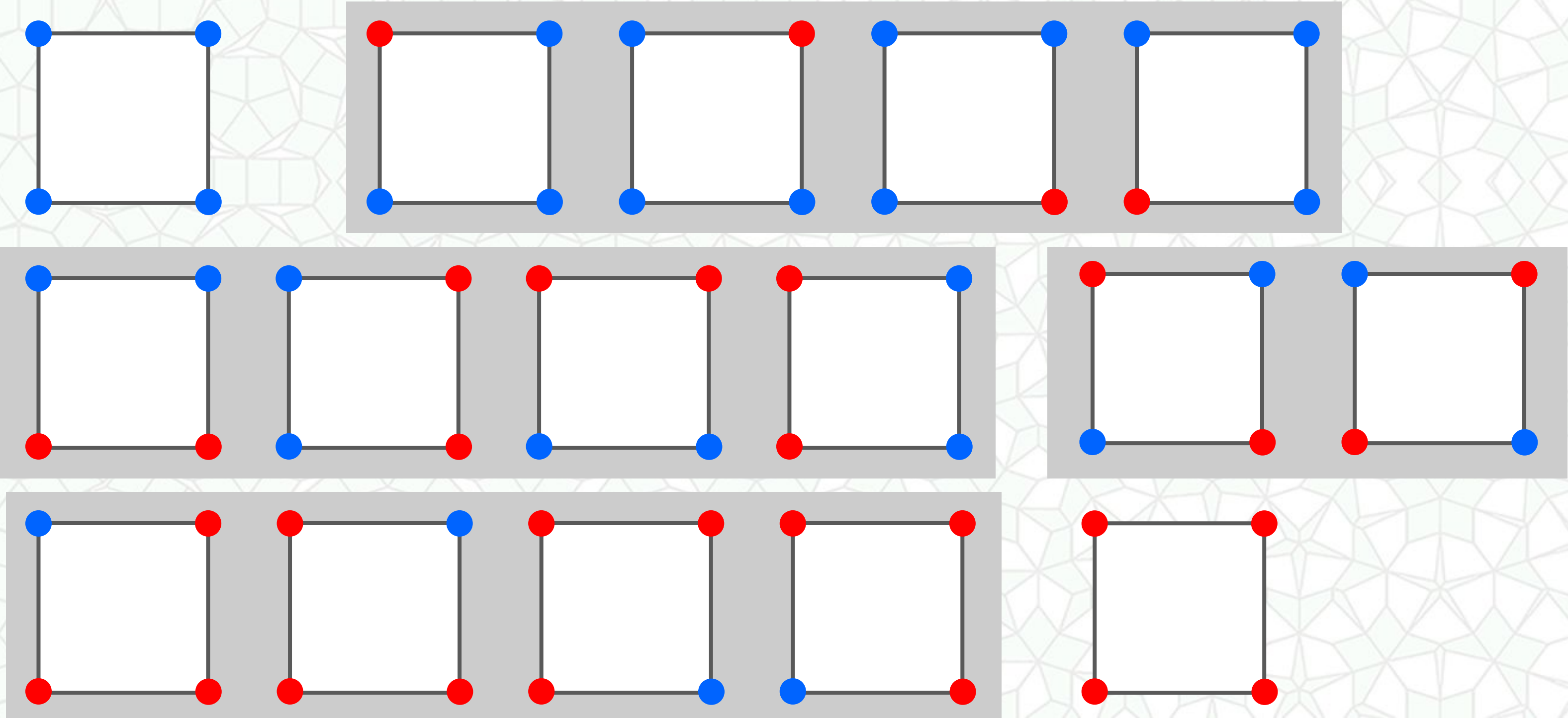
Enumerate Cases in 2D?

- How many cases? **$2^4 = 16$**
- How many unique cases (excluding rotations)?



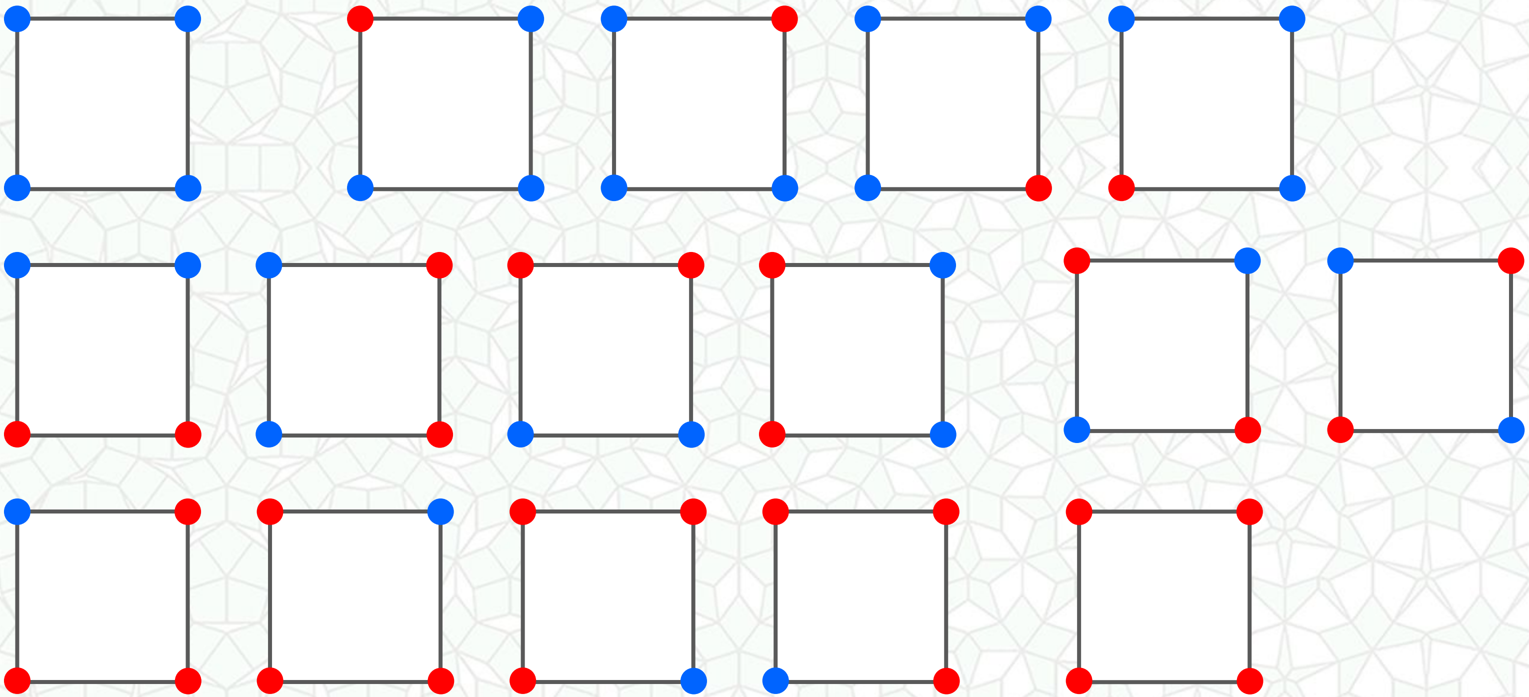
Enumerate Cases in 2D?

- How many cases? **$2^4 = 16$**
- How many unique cases (excluding rotations)? **6**



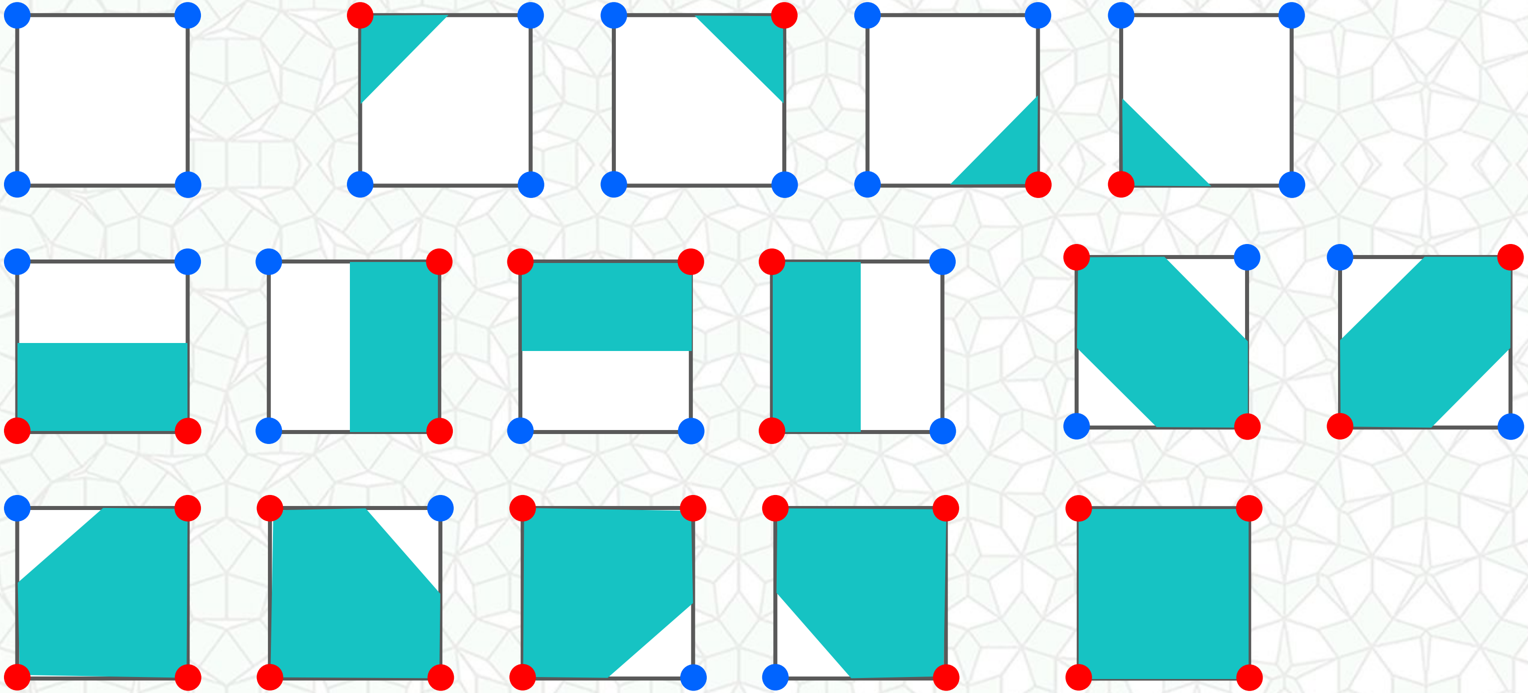
2D Marching Cubes Volume & Surface

- What portion of the cell is inside of the object?
- Where is the surface separating inside from outside?



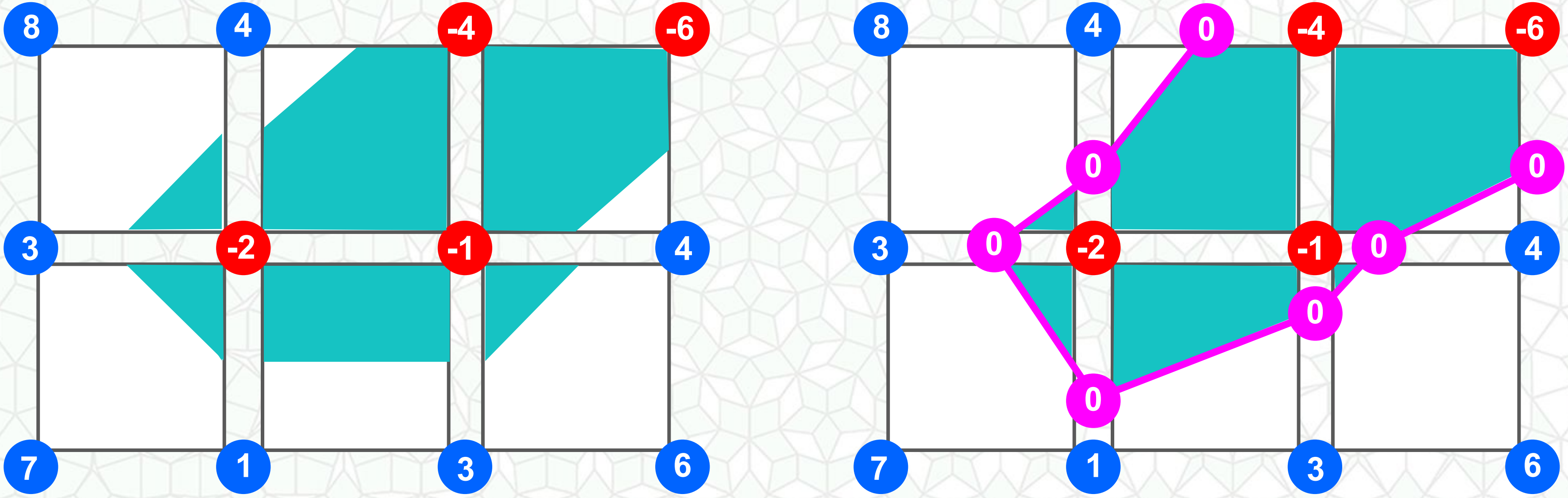
2D Marching Cubes Volume & Surface

- What portion of the cell is inside of the object?
- Where is the surface separating inside from outside?

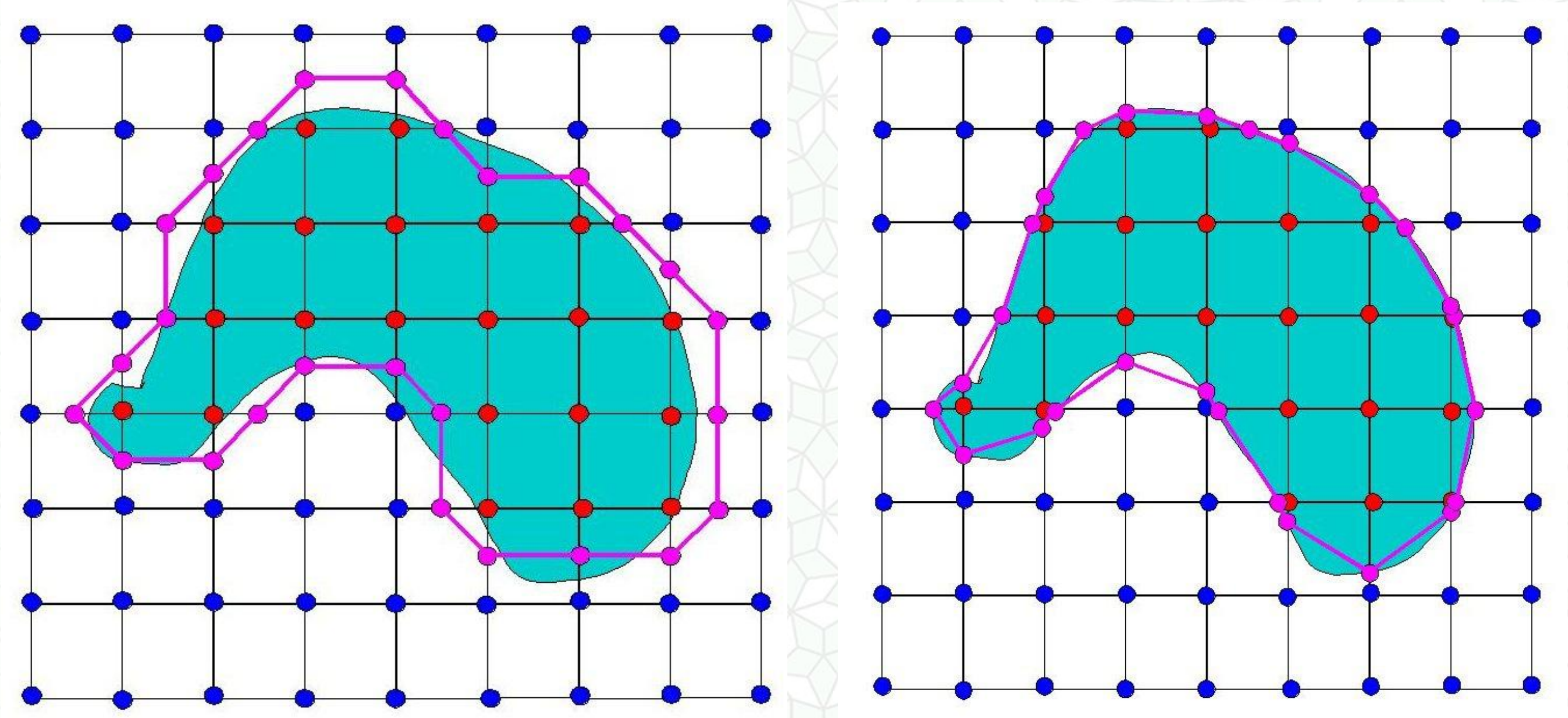


More than Binary – Use Signed Distance Data!

- NOTE: We don't place vertices at the midpoints of cell edges, but at the estimated (interpolated) position of the level set!



More than Binary – Use Signed Distance Data!



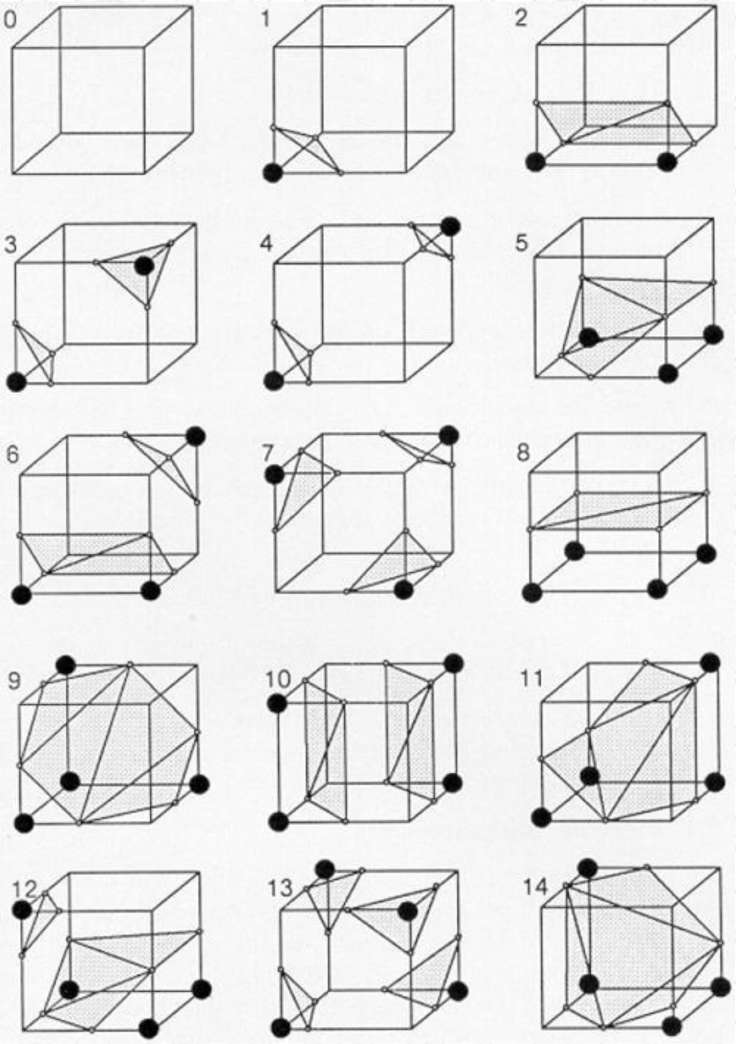
http://www.cs.carleton.edu/cs_comps/0405/shape/marching_cubes.html

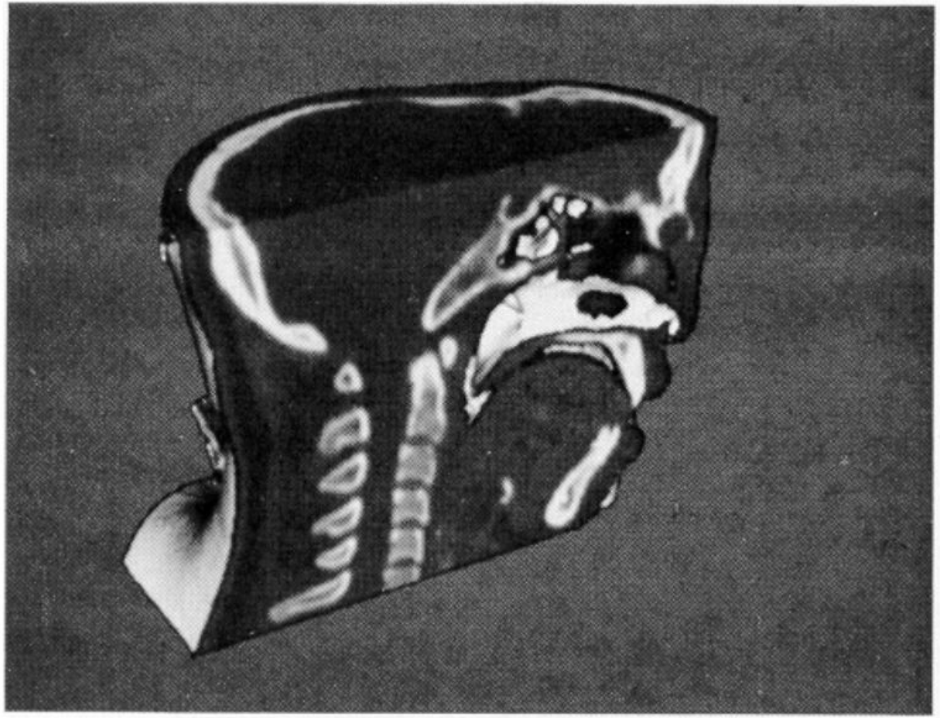
3D Marching Cubes

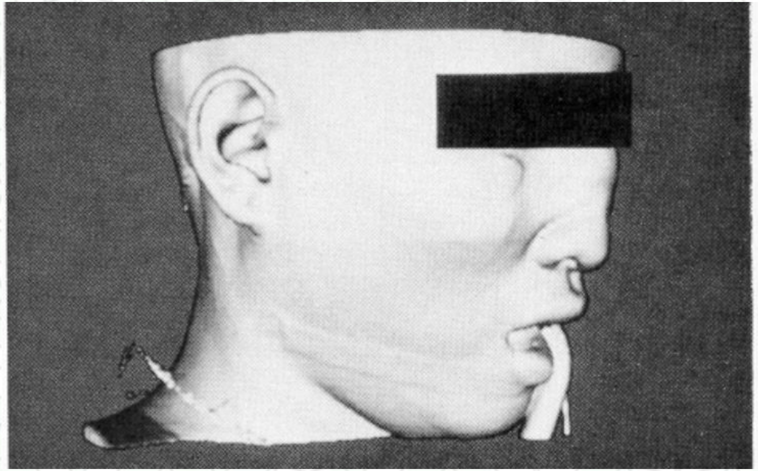
- 256 possible inside/outside labelings of each grid cube.

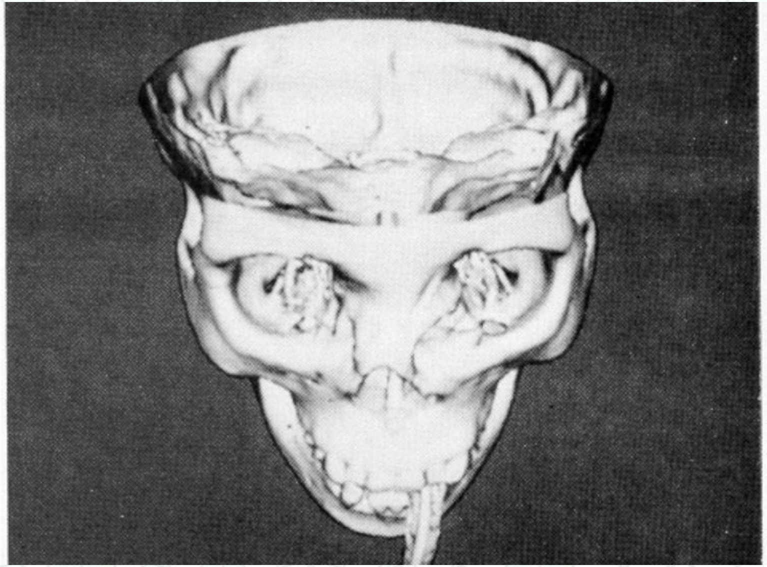
- Merging rotations…
  15 unique cases to implement

"Marching Cubes: A High Resolution 3D Surface Construction Algorithm", Lorensen and Cline, SIGGRAPH '87.





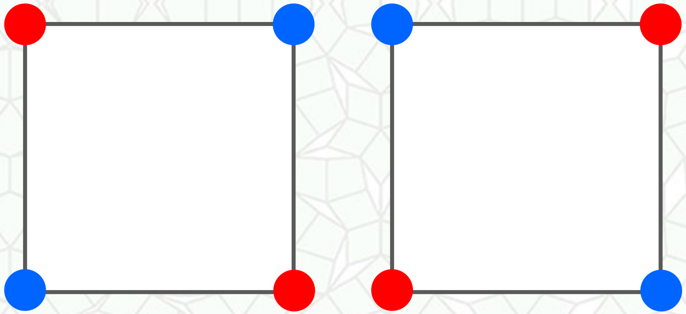




"Marching Cubes: A High Resolution 3D Surface Construction Algorithm", Lorensen and Cline, SIGGRAPH '87.

Uniqueness of 2D Marching Cubes?

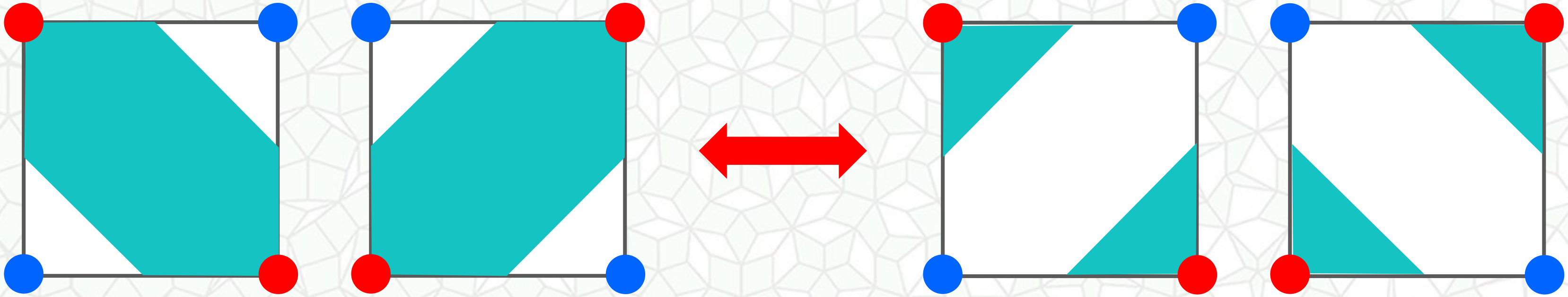
- What portion of the cell is inside of the object?
- *Is the answer unique? Is there any ambiguity in the answer?*



Ambiguity in 2D Marching Cubes

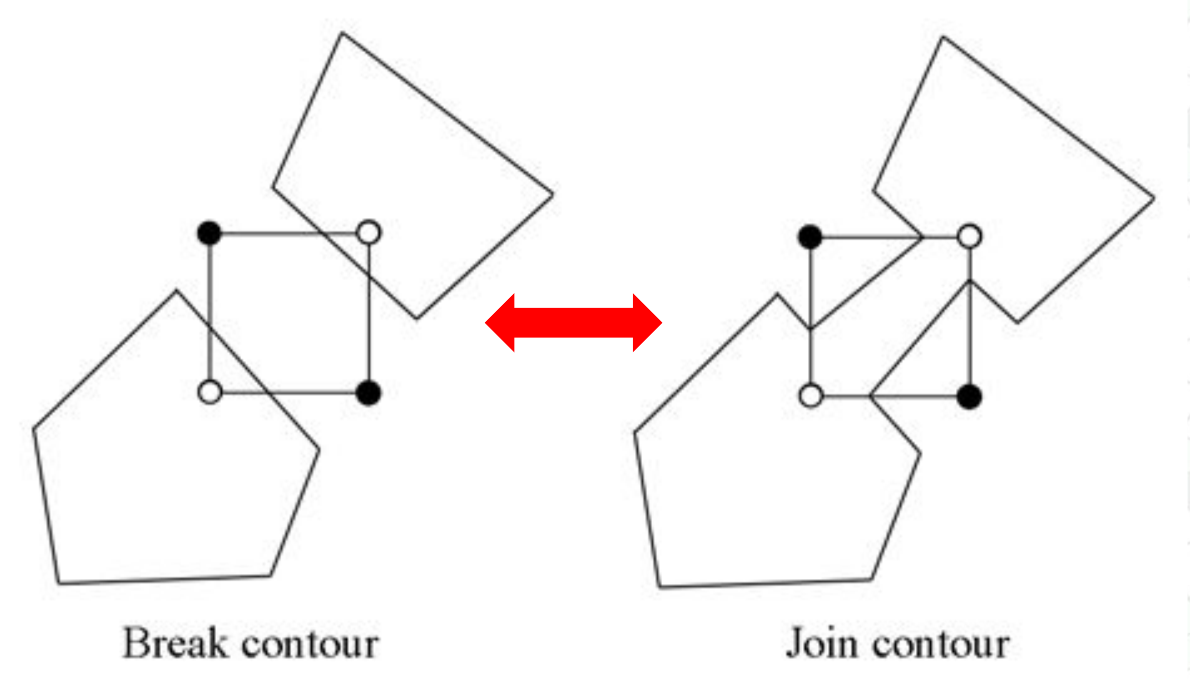
- What portion of the cell is inside of the object?
- *Is the answer unique? Is there any ambiguity in the answer?*

- ***The center of the diagonal cases can either be interior or exterior!***



Ambiguity in 2D Marching Cubes

- ***The choice will affect the global topology of the surface and its connectedness!***



http://users.polytech.unice.fr/~lingrand/MarchingCubes/algo.html

Ambiguity in 3D Marching Cubes

- ***Furthermore in 3D…***

  If the choices made in neighboring cells are inconsistent, the surface can have gaps and not be closed and "watertight"!

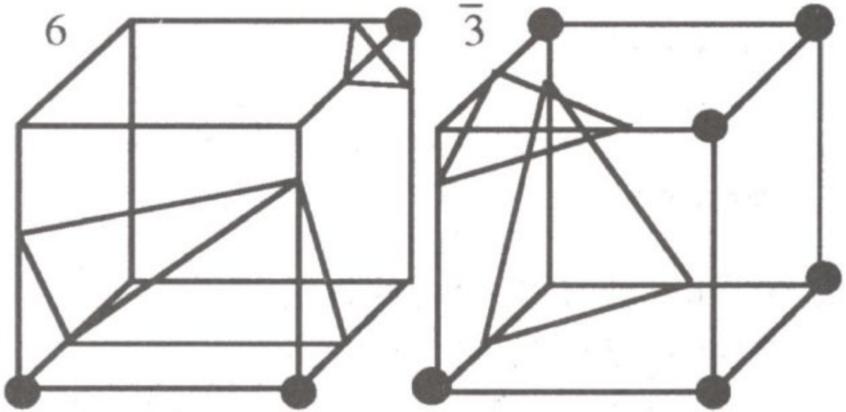
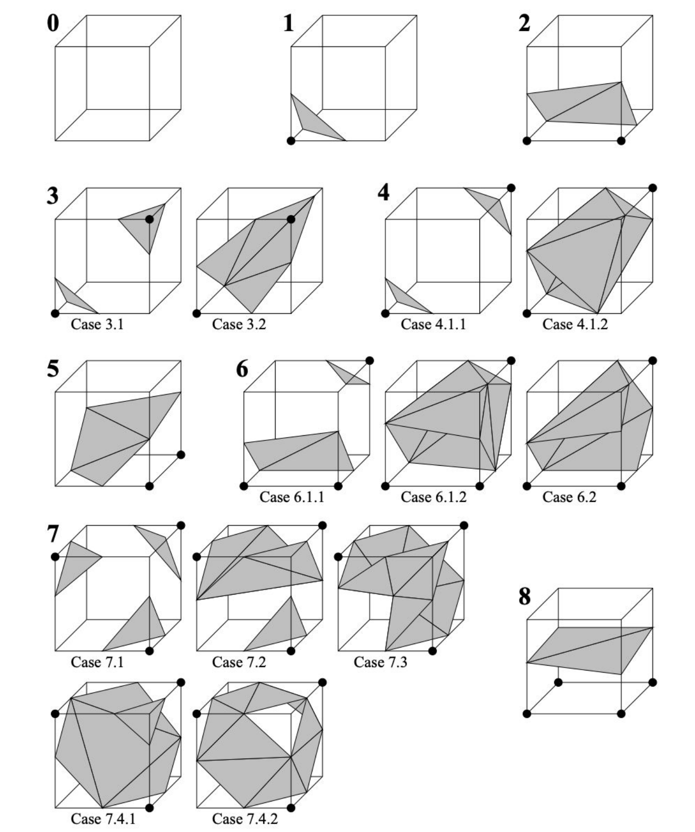


Figure 3. An example illustrating the flaw in the marching cubes method.

"The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes", Nielsen & Hamann, 1991

Fixing Marching Cubes

"Marching Cubes 33: Construction of Topologically Correct Isosurfaces", Chernyaev, 1996



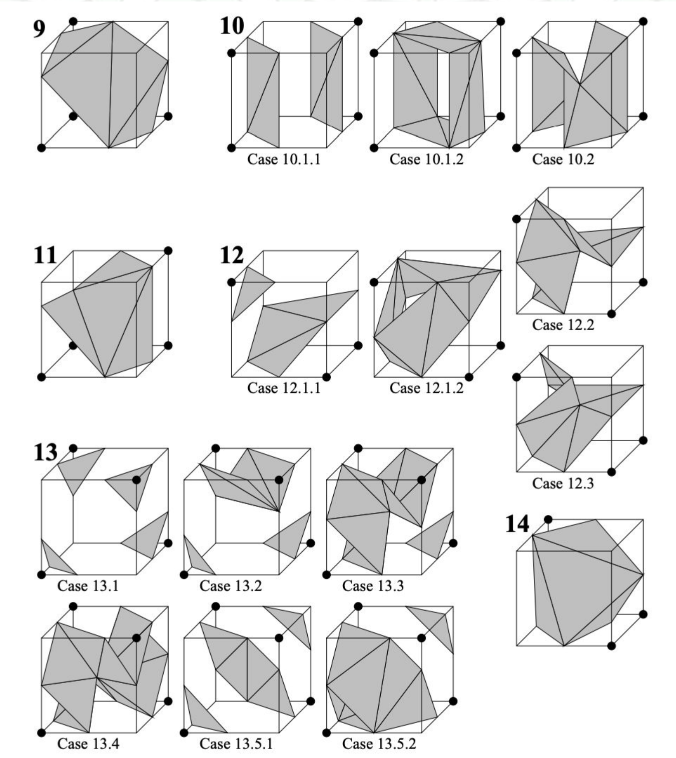
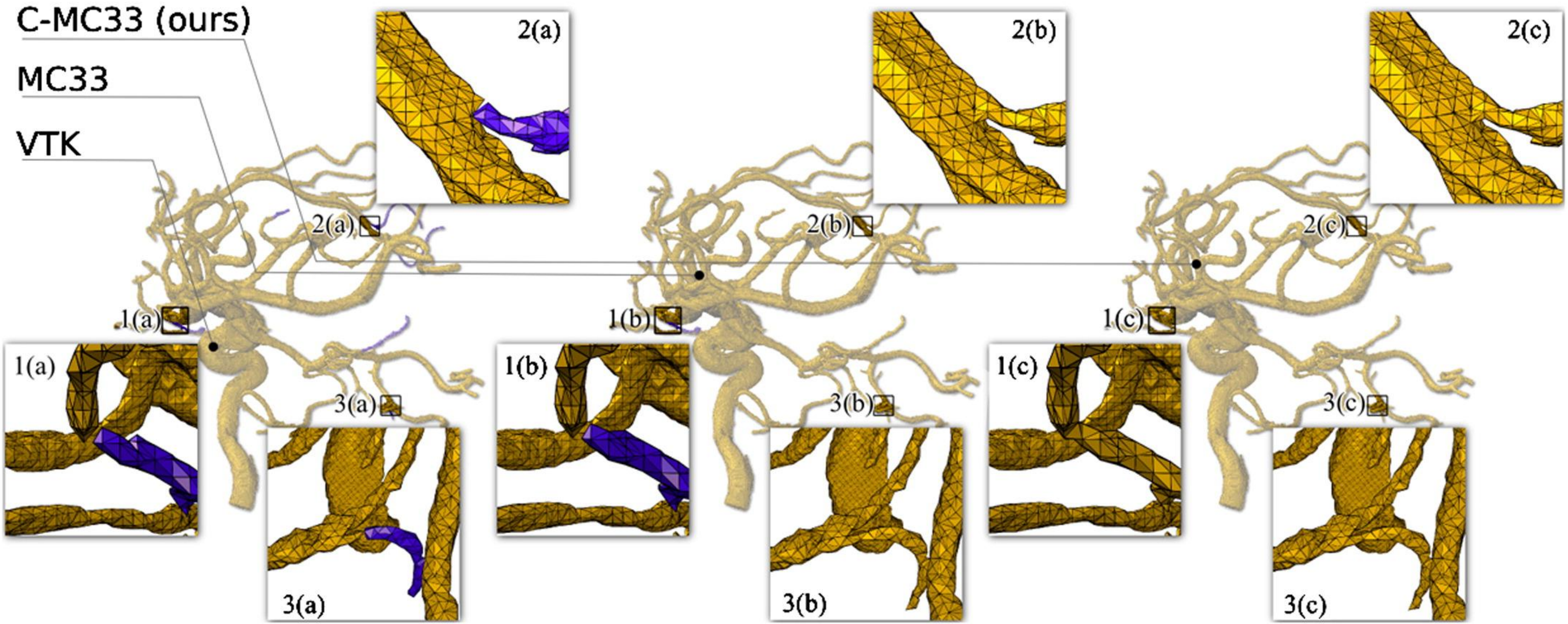
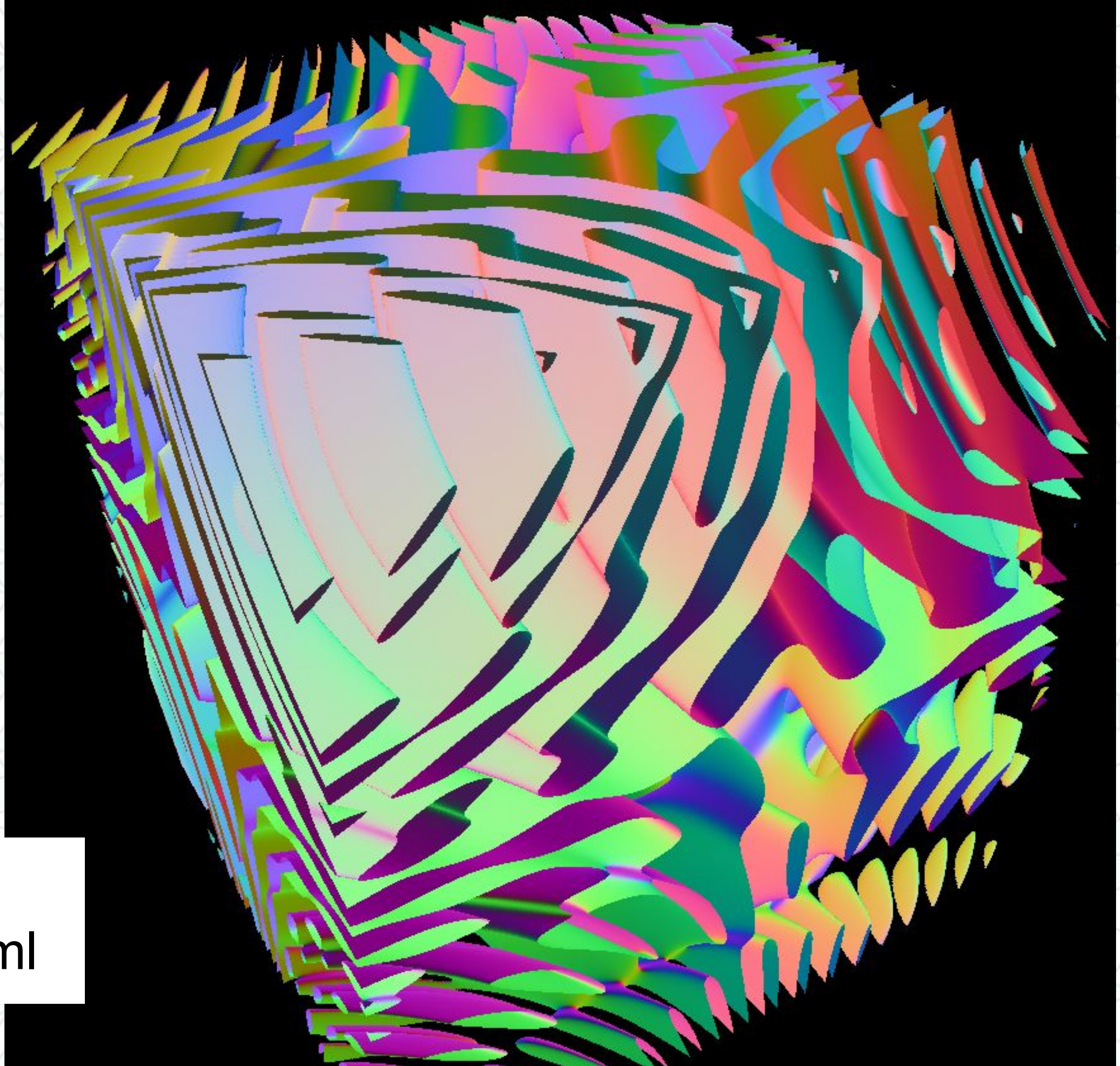


Figure 8: Advanced lookup table

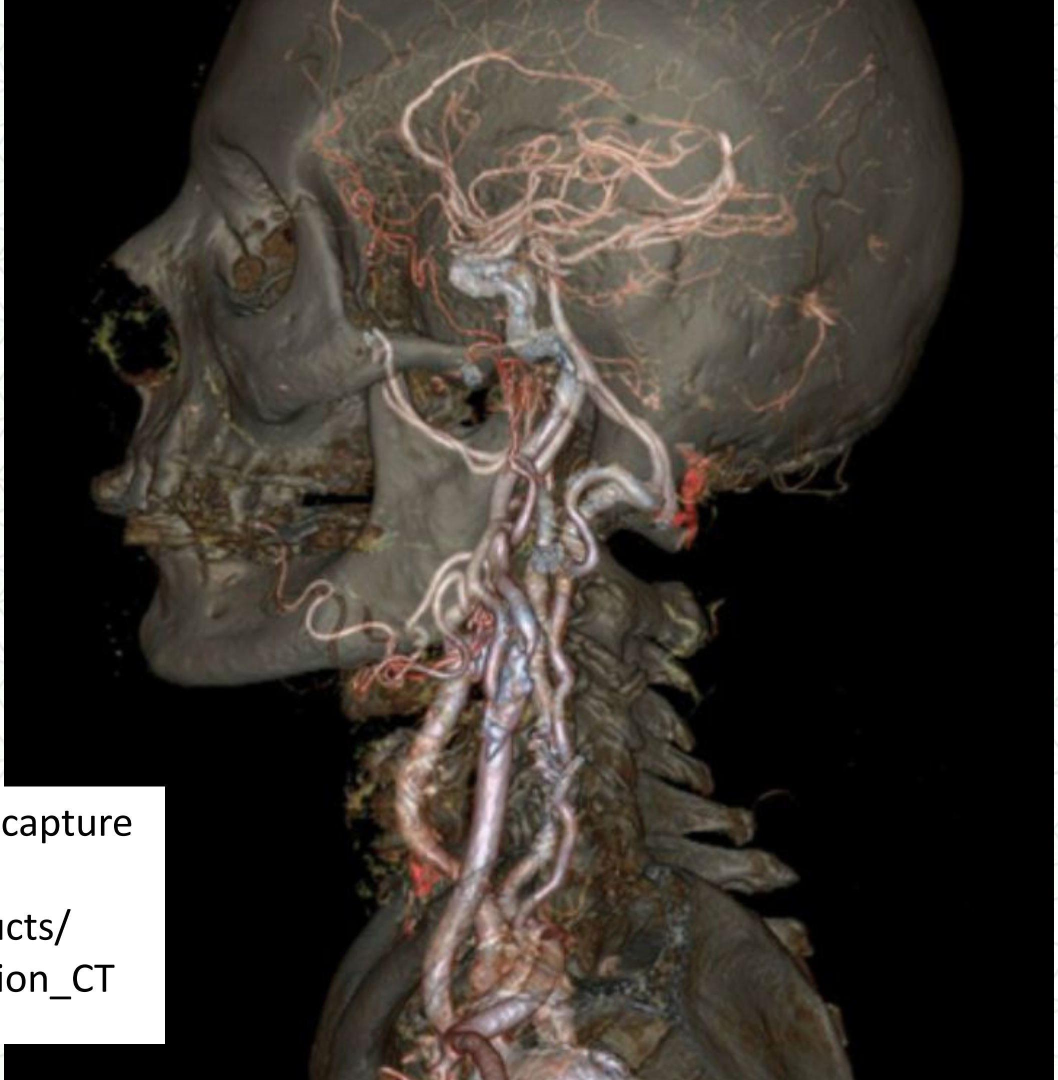
Ambiguity in Marching Cubes

"Practical considerations on Marching Cubes 33 topological correctness"
Custodio, Etiene, Pesco, Silva, 2013





https://graphics.stanford.edu/
~mdfisher/MarchingCubes.html



http://gizmodo.com/ges-new-fast-ct-scanner-captures-insane-images-in-a-he-1482904872
http://www3.gehealthcare.com/en/Products/Categories/Computed_Tomography/Revolution_CT

Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation

BTW: Marching Cubes was Patented…

United States Patent [19]

Cline et al.

[11] **Patent Number:** **4,710,876**

[45] **Date of Patent:** **Dec. 1, 1987**

---

[54] **SYSTEM AND METHOD FOR THE DISPLAY OF SURFACE STRUCTURES CONTAINED WITHIN THE INTERIOR REGION OF A SOLID BODY**

[75] Inventors: **Harvey E. Cline**, Schenectady; **William E. Lorensen**, Ballston Lake, both of N.Y.

[73] Assignee: **General Electric Company**, Schenectady, N.Y.

Graphics" Computer Magazine, IEEE Computer Society Publication, (Oct. 1984), pp. 145–161.

Artzy, Ehud et al., "The Theory, Design, Implementation and Evaluation of a Three–Dimensional Surface Detection. Algorithm", *Computer Graphics and Image Processing*, vol. 15, (1981) pp. 1–24.

Primary Examiner—Jerry Smith

Assistant Examiner—Gail Hayes

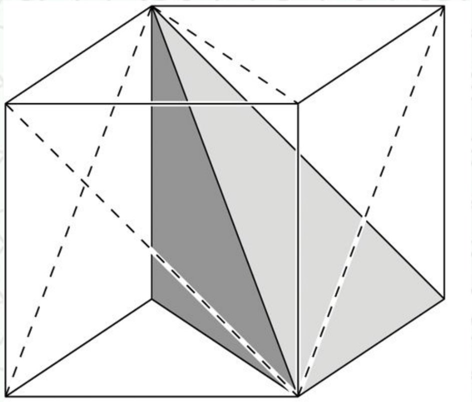
Attorney, Agent, or Firm—Lawrence D. Cutter; James C. Davis, Jr.; Marvin Snyder

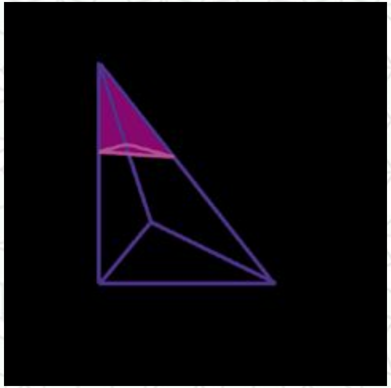
… and the patent expired in 2005

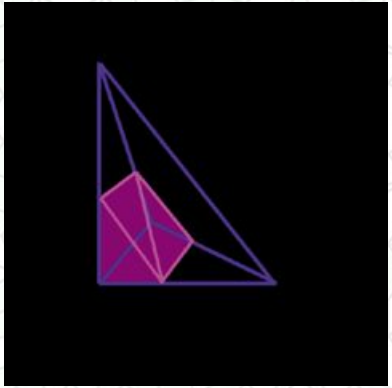
Marching Tetrahedra

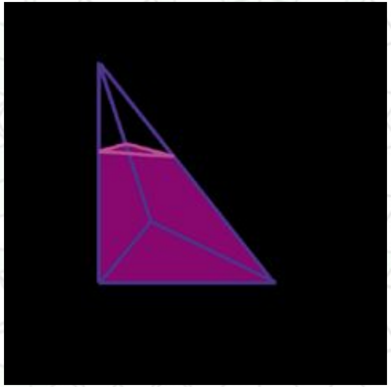
Jules Bloomenthal
"An implicit surface polygonizer"
Graphics Gems IV

- Implementation Shortcut (& Patent Workaround):
  Chop every grid cube into 6 tetrahedra….





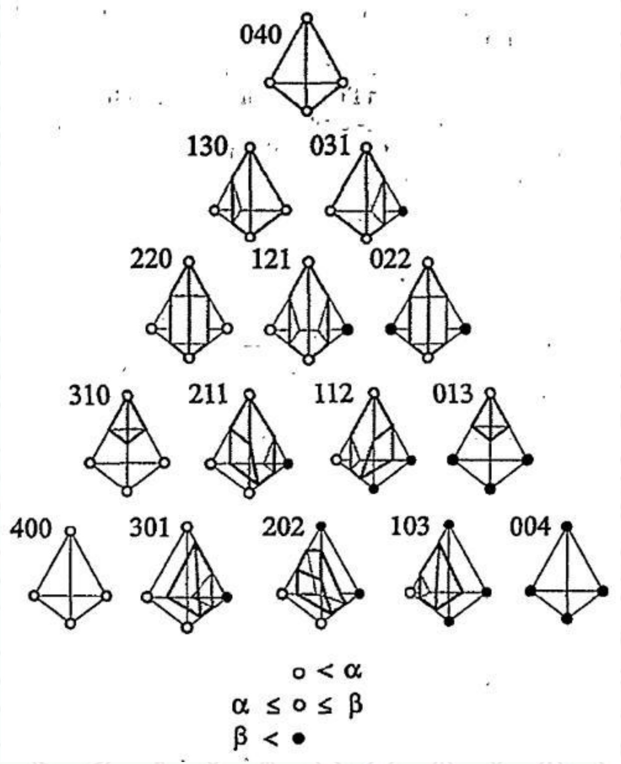




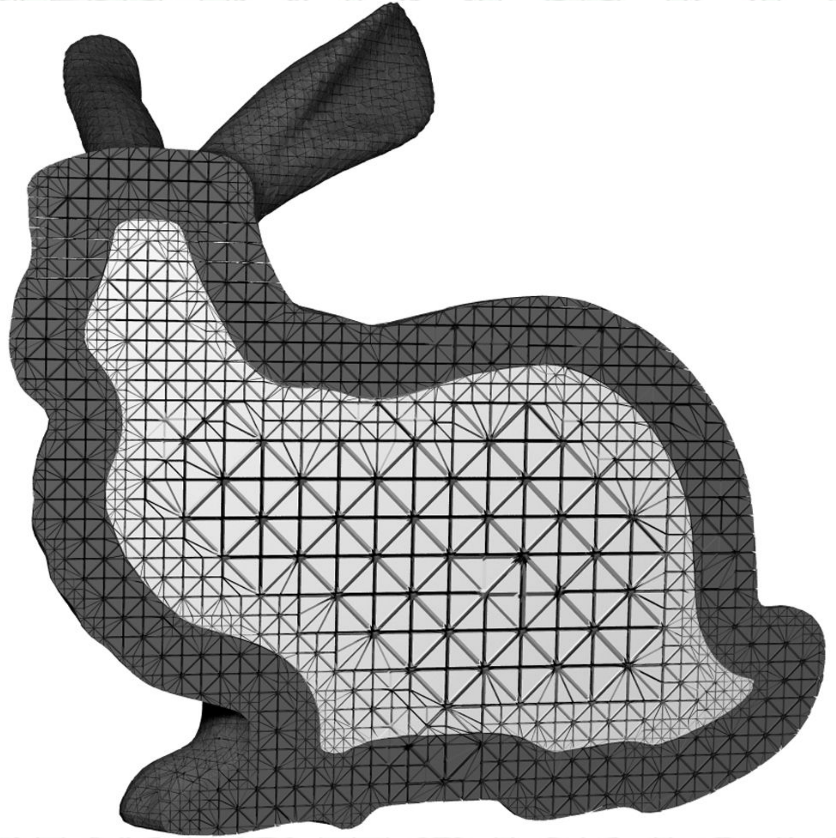
- Now only 3 unique cases for tetrahedra!

"When the Blobs Go Marching Two by Two", *Jeff Lander,* Gamasutra

Volumetric & Multiple Materials

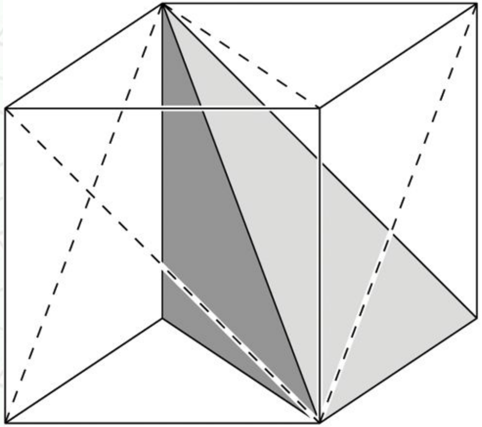


"Interval volume tetrahedrization"
Visualization '97, Nielson & Sung

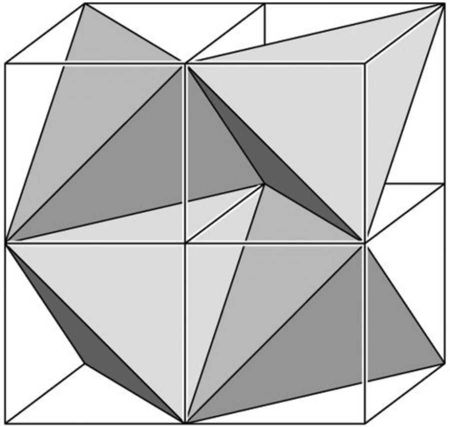


Implementation Details… Marching Tetrahedra

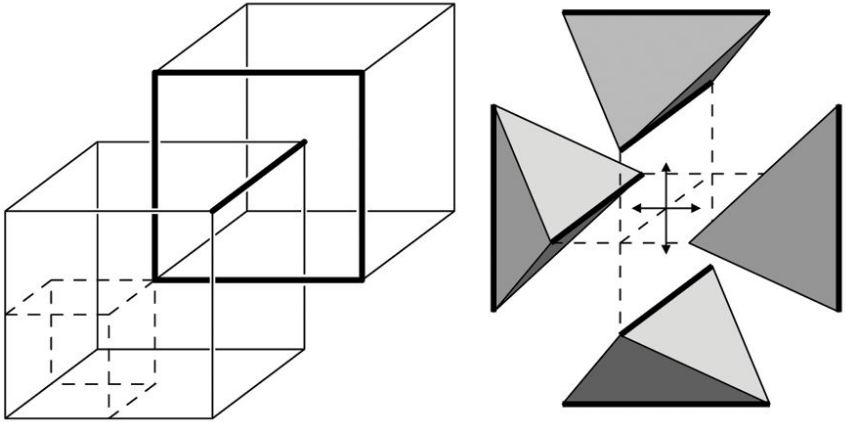
- Which cube → tetrahedra subdivision should we use?



6 tetrahedra
(all equal size & shape)
diagonal bias



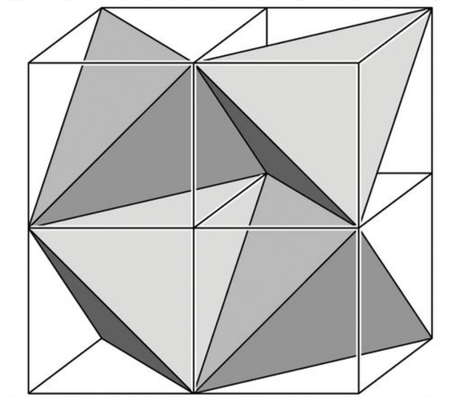
5 tetrahedra (1 equilateral
that is 2X the others in volume)
Orientation must be alternated

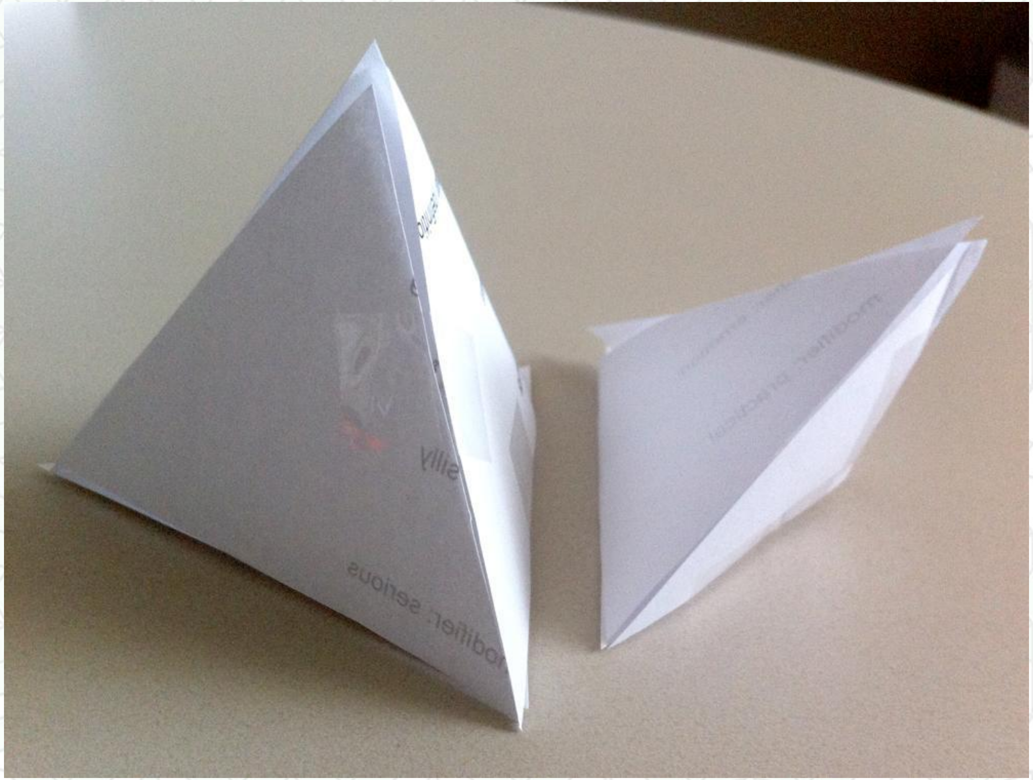


Crystal Lattice
All same size & shape,
but more complicated…

Debugging Marching Tetrahedra

- Drawing (in 2D) didn't work
- Creating an OpenGL visualization didn't work (even with transparency)
- Solution: build lots of paper & tape models





Outline for Today

- Final Project & Homework 7 Questions?
- Last Time: Quad Trees
- Explicit vs. Implicit Surface Representations
- Signed Distance Field
- Level Sets (Surface → Signed Distance)
- Fast Marching Method
- Medical Imaging
- Marching Cubes (Signed Distance → Surface)
- Marching Tetrahedra
- Next Time: Exact Computation