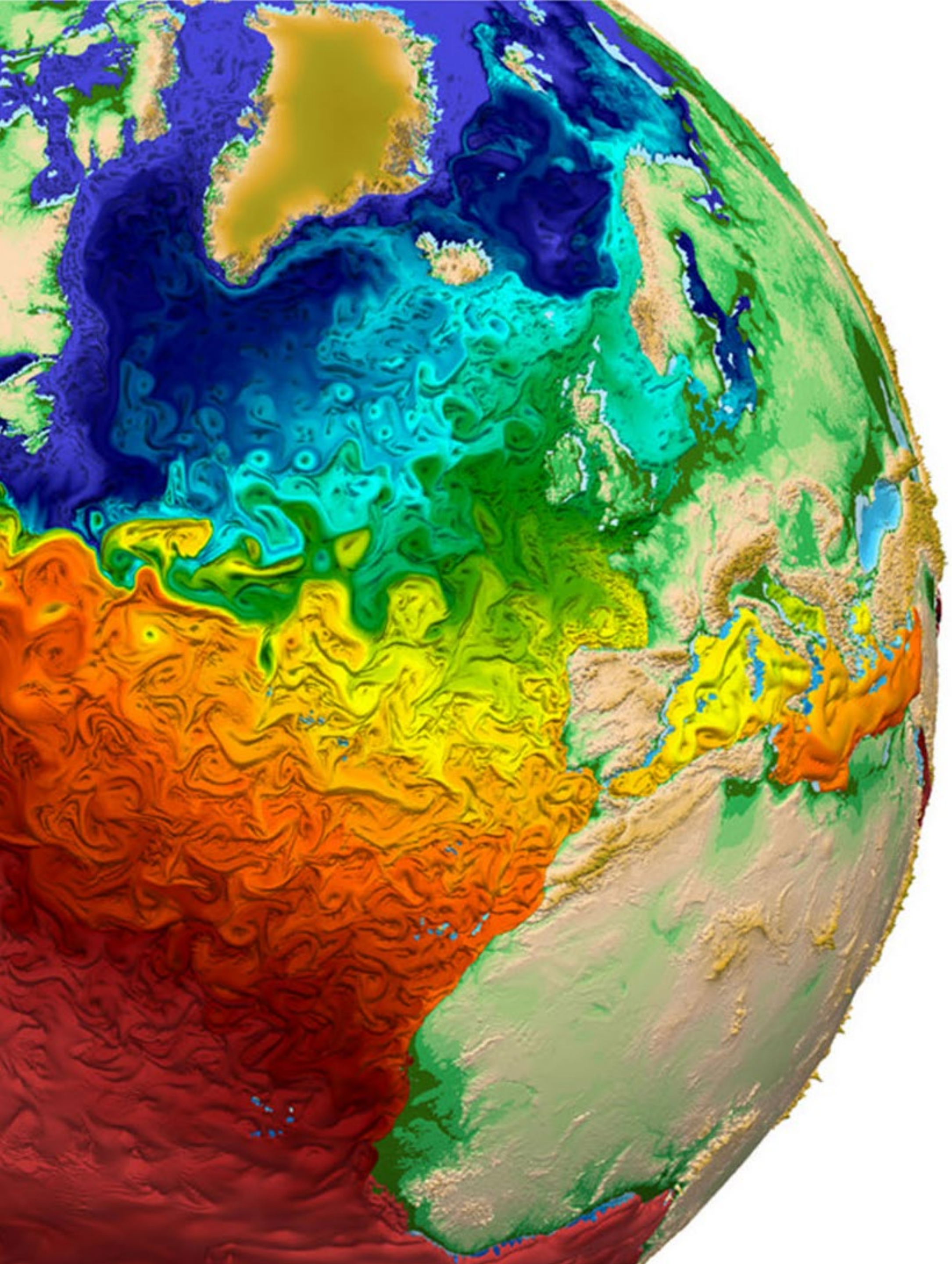


Contouring

Marching Squares

Scientific Visualization

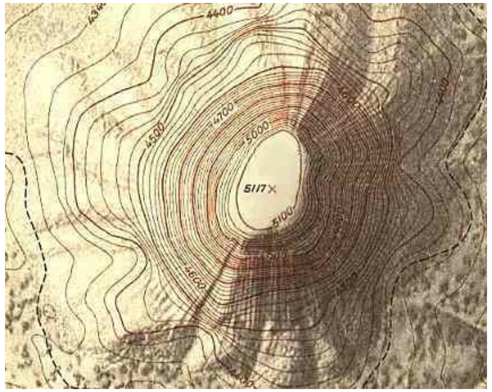
Professor Eric Shaffer

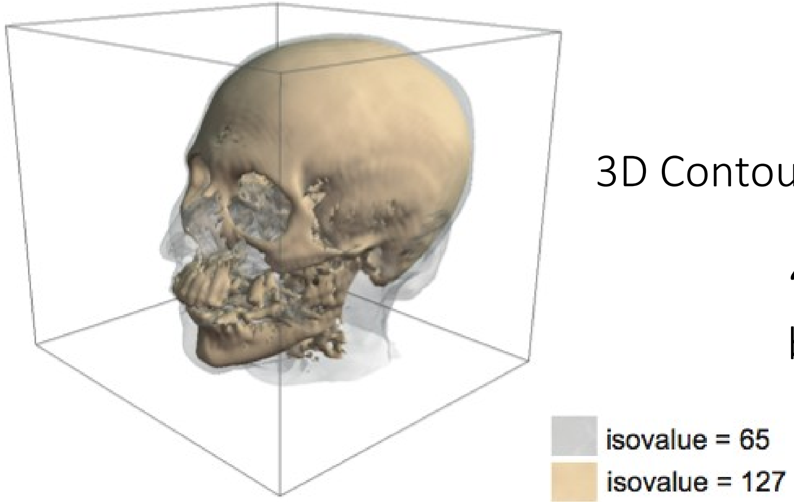
ILLINOIS

Contouring

Contours have been used for hundreds of years in cartography

- also called *isolines* ('lines of equal value')





3D Contouring: Marching Cubes:

"Marching cubes: A high resolution 3D surface construction algorithm",
by Lorensen and Cline (1987)

16,000 citations on Google Scholar

Contour Properties

Definition $I(f_0) = \{x \in D \; f(x) = f_0\}$

Contours are always closed curves (except when they exit *D*)

- why? Recall that *f* is C^0

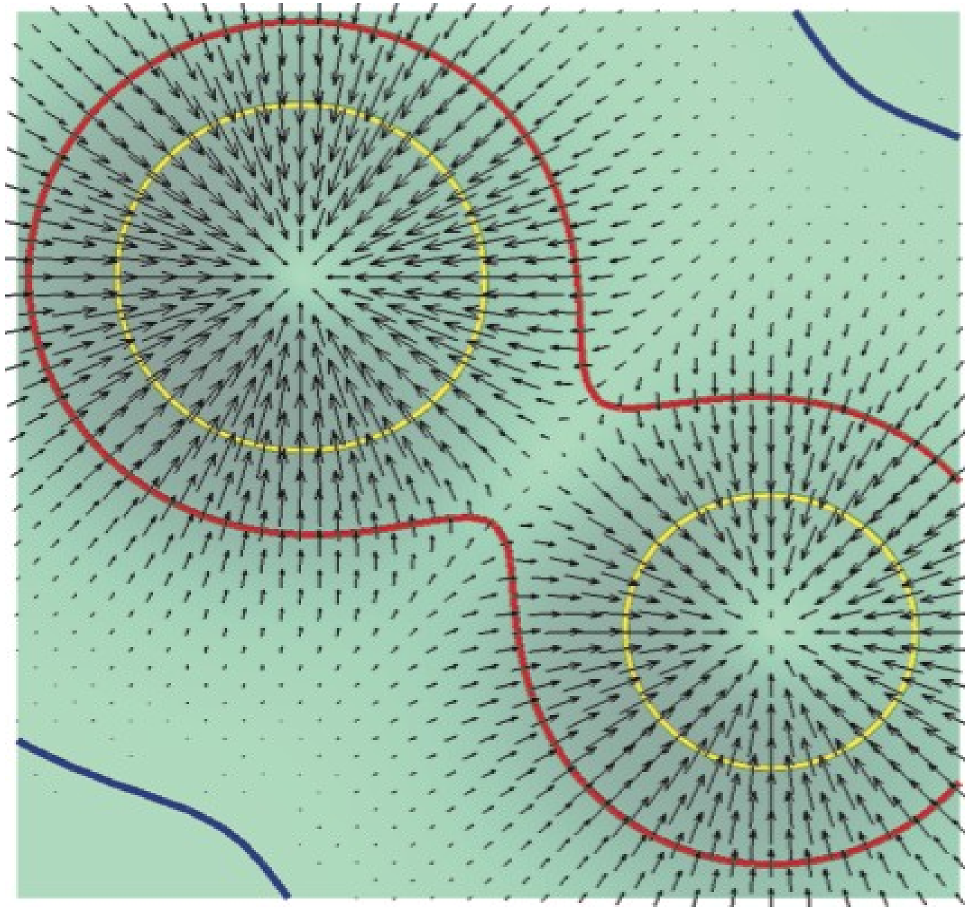
Two different contour lines never intersect, thus are nested

- why? What would it mean if a point belonged to two *different* contours

Contour Properties

Contours are always orthogonal to the scalar value's gradient

- why?



$$I(f_0) = \{x \in D \mid f(x) = f_0\}$$

$$\nabla f(p) \cdot \vec{u} = 0$$

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

$\vec{u}$

p

contour: $\dfrac{\partial f}{\partial I} = 0$ since f constant along I

gradient: $\dfrac{\partial f}{\partial (\nabla f)} = \max$ by definition of gradient direction of greatest increase in f

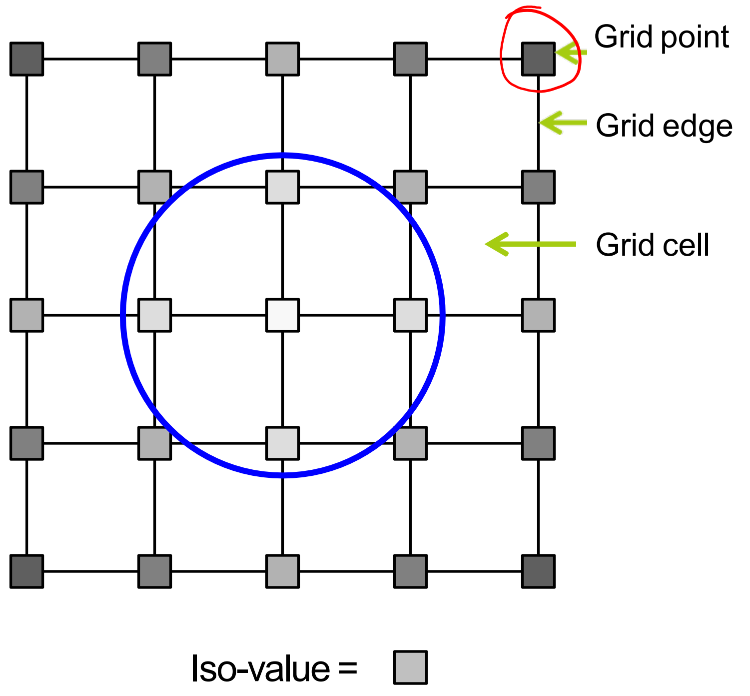
gradient of a scalar field (drawn with arrows) is orthogonal to contours

ILLINOIS

Contouring on a Grid of Sampled Data

Input

- A grid where each grid point has a value
- An iso-value (threshold)



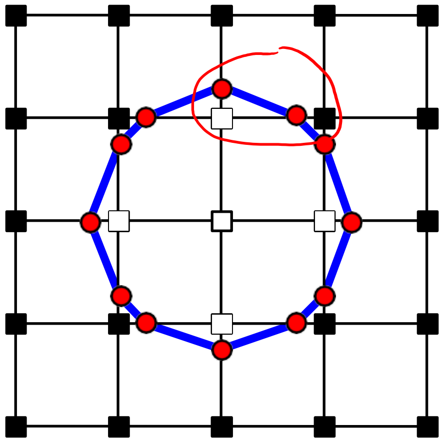
Output

- A closed polyline (2D) or mesh (3D) that separates grid points above or below the iso-value

Algorithms

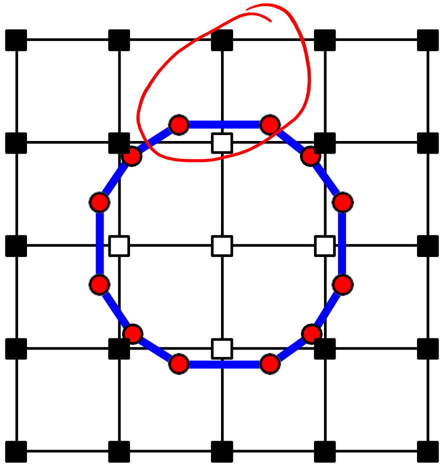
Primal methods

- Marching Squares (2D), Marching Cubes (3D)
- Placing vertices on grid edges

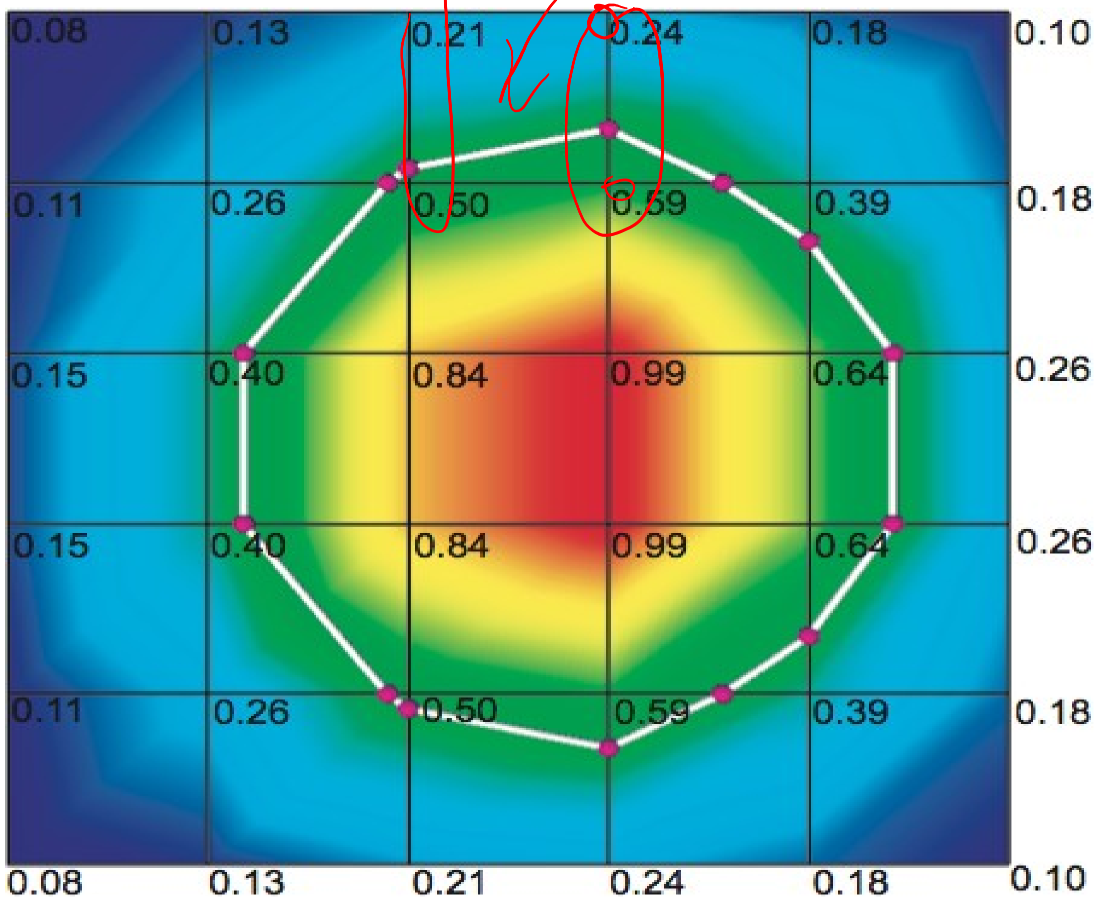


Dual methods

- Dual Contouring (2D,3D)
- Places vertices in grid cells



Contouring in 2D



$S = \varnothing$

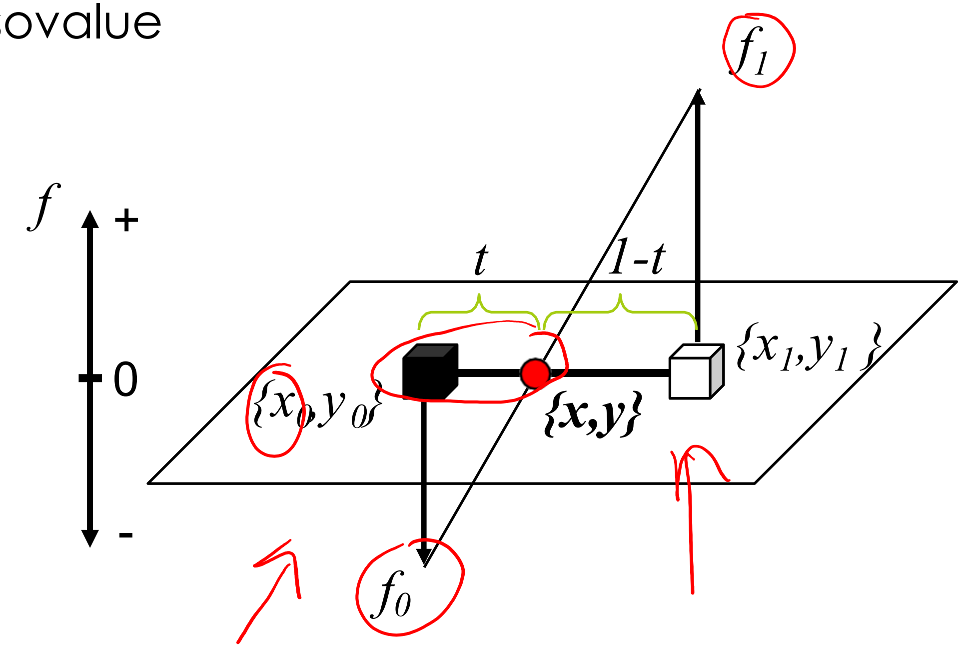
for(each cell c in D)
{
  for(each edge e=(p_i,p_j) of c)
  {
    if($f_i < v < f_j$)
    {
      Compute the intersection point q

      $S = S \cup q$
    }
  }
  connect points in S with lines to build contour;
}

Marching Squares

Creating contour line vertices (x,y)

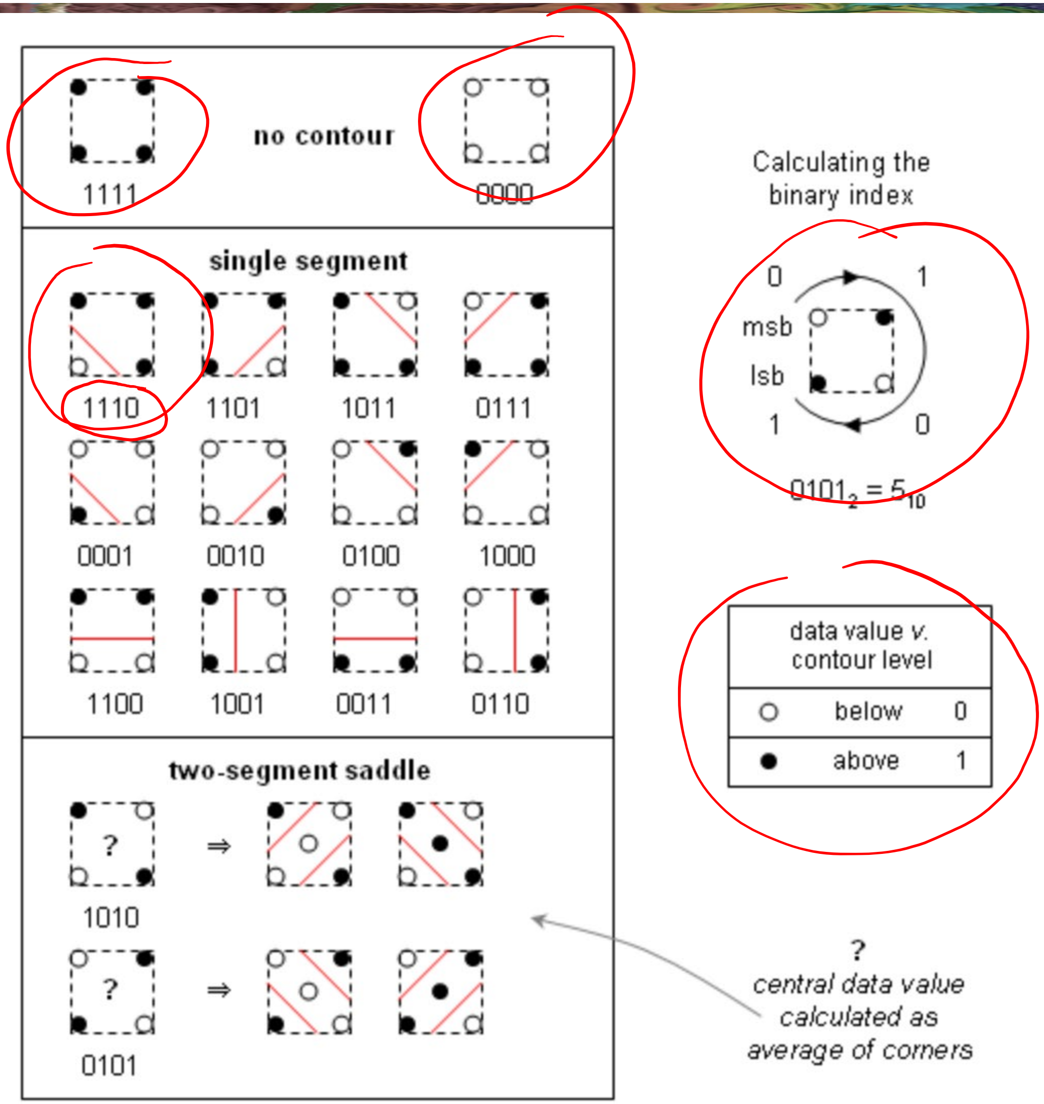
- Assume the underlying, continuous function is linear on the grid edge
- Linearly interpolate the positions of the two grid points
- v is the isovalue



$$\boxed{t = \frac{v - f_0}{f_1 - f_0}}$$

$$\boxed{\begin{aligned} x &= x_0 + t(x_1 - x_0) \\ y &= y_0 + t(y_1 - y_0) \end{aligned}}$$

Marching Squares



2D contouring on quad-cell grids

1. Encode inside/outside state of each vertex in a 4-bit id

2. Process all dataset cells

- for each cell,
  use ids as pointers into a table with 16 cases

- each case has associated code to
  - compute the edge-contour intersection positions
  - connect to already-computed contour vertices from previous cells

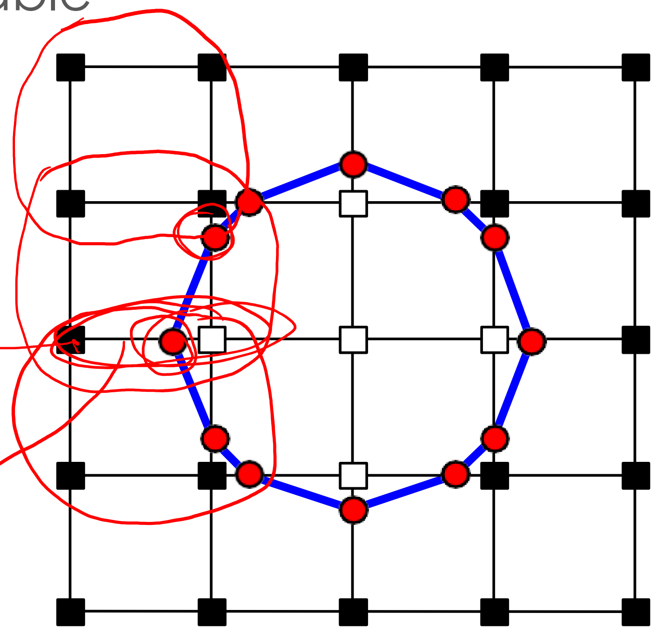
Marching Squares: Implementation

Avoid computing one vertex multiple times

- Compute the vertex location once, and store it in a hash table

When the grid point value is same as the iso-value

- Treat it either as “above” or “below”, but be consistent.



(i, j)

Marching Squares: Example

Use an isovalue of 5

Scalars associated with point to the upper left

Classify points with value 5 as positive

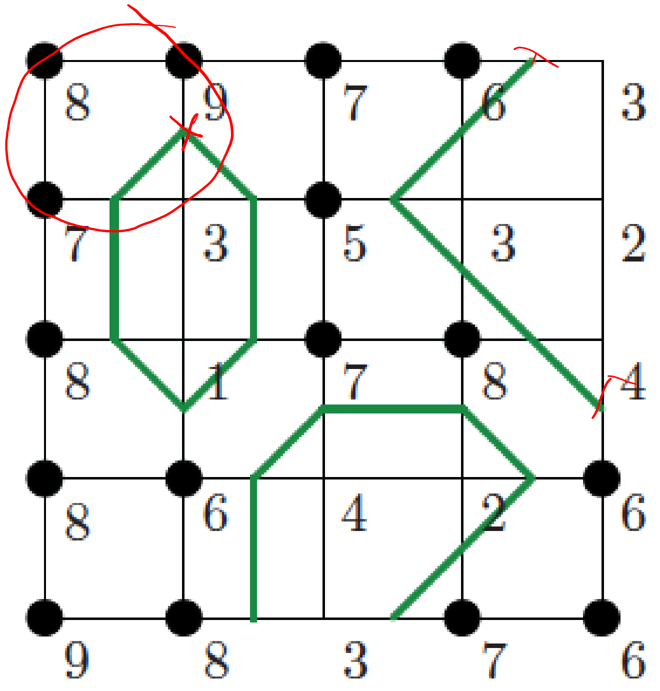
8	9	7	6	3
7	3	5	3	2
8	1	7	8	4
8	6	4	2	6
9	8	3	7	6

(a) Scalar grid.

Marching Squares: Example Using Midpoints

Scalars associated with point to the upper left

Classify points with value 5 as positive



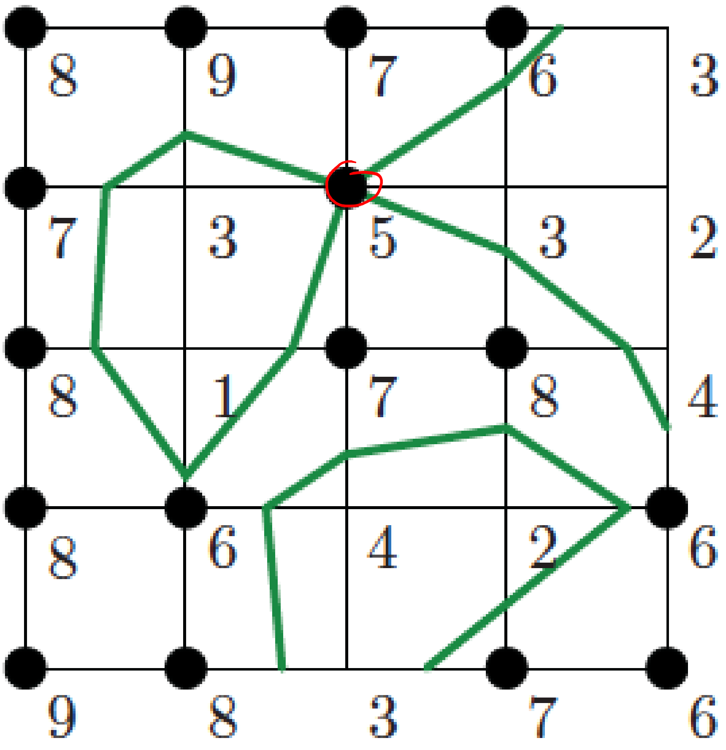
(c) Midpoint vertices.

ILLINOIS

Marching Squares: Example Using Interpolation

Scalars associated with point to the upper left

Classify points with value 5 as positive

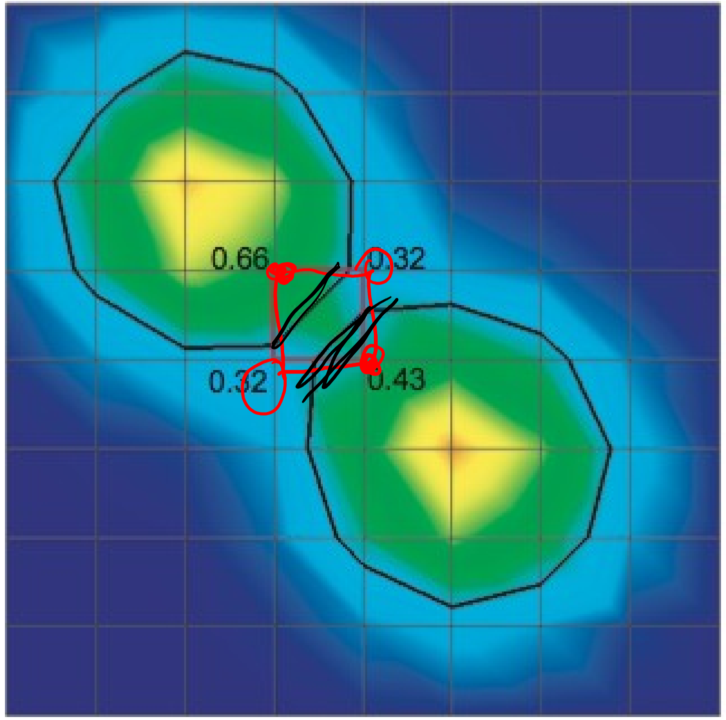


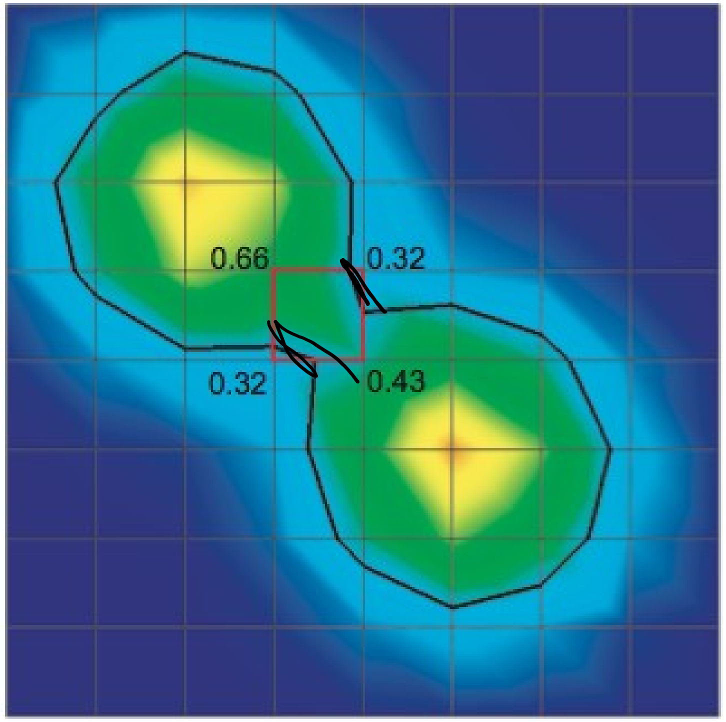
(d) Isocontour.

Contouring: Ambiguity

Each edge of the red cell intersects the contour

- which is the right contour result?

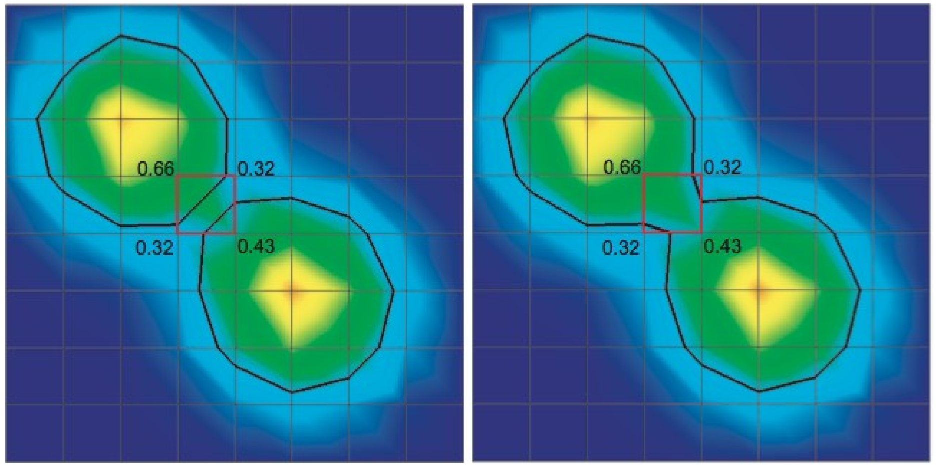




Contouring: Ambiguity

Each edge of the red cell intersects the contour

- which is the right contour result?



Both answers are equally correct!

- we could discriminate only if we had higher-level information (e.g. topology)
- at cell level, we cannot determine more unless we increase sampling rate

Contouring: Ambiguity

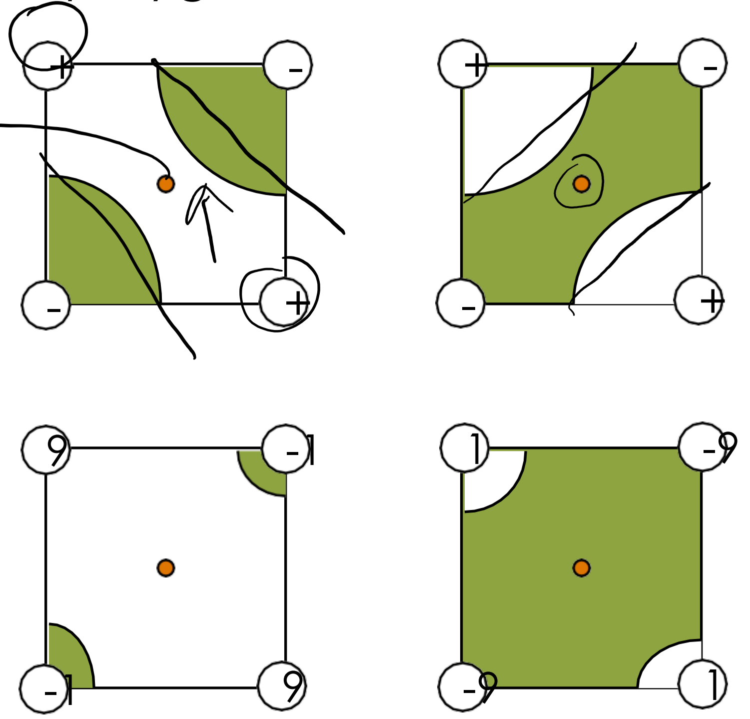
Some cell corner value configurations yield more than one consistent polygon

- In 3-D can yield holes in surface!

How can we resolve these ambiguities?

- Topological Inference
    - Sample a point in the center of the ambiguous face

avg



If data is discretely sampled,
bilinearly interpolate to sample

$p(s,t) = (1\text{-}s)(1\text{-}t)\ a + (1\text{-}t)\ b + (1\text{-}s)\ t\ c + s\ t\ d$

a,b,c, and d are the function values at the 4 corners
s and t are parametric location inside the grid cell
…for midpoint s = ½ t= ½

ILLINOIS