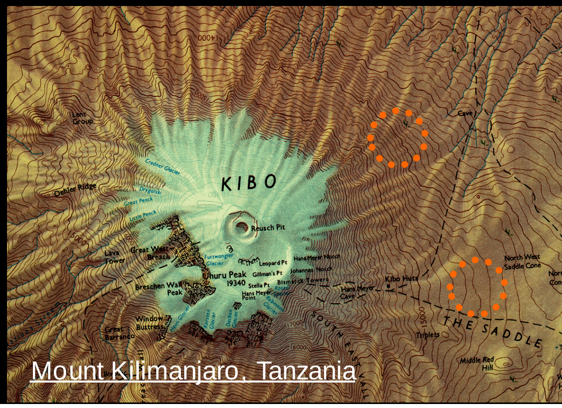
CS5630/6630

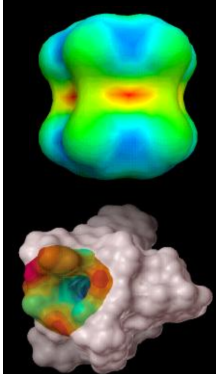
Isosurfacing

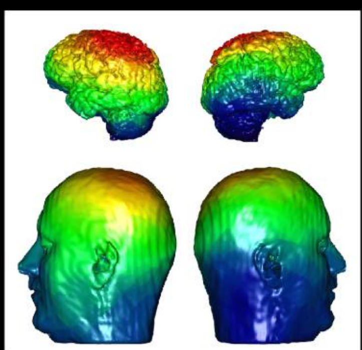
http://www.lib.berkeley.edu/EART/digital/topo.html



Mount Kilimanjaro, Tanzania

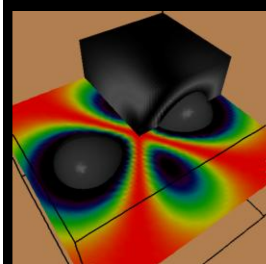
Colored Isosurfaces

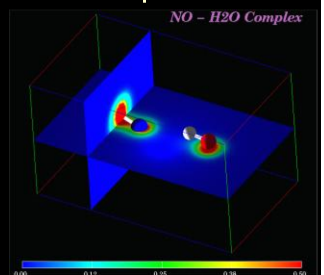




David Weinstein

Slices still have their place

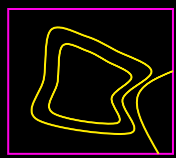




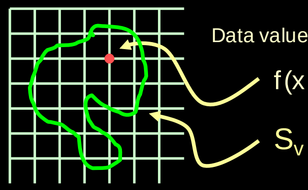
Colormapped slices

Properties of Isocontours

- Preimage of scalar value
  - Concept generalizes to any dimension
  - Manifolds of codimension 1
- Closed (except at boundaries)
- Nested–different values don't cross
  - Can consider the zero-set case (generalizes)
  - F(x, y) = k <–> F(x, y) - k = 0
- Normals given by gradient vector of F



Where are the data values?



Data value f defined on grid points only

But we want a continuous, closed surface

Two solutions:

- Interpolate to get the "right" answer
  - Subsampling or raycasting
  - Dividing Cubes
- Approximate to get a "good" answer
  - Geometric primitives
  - Go cell by cell

Contours in 2D

- Assign gometric primitives to "cells" consisting of 2x2 grid points

Contours in 2D

- Assign gometric primitives to "cells" consisting of 2x2 grid points
  - Line segments

Contours in 2D

- Assign gometric primitives to “cells” consisting of 2x2 grid points
  - – Line segments
- How do we know how to organize the primitives?

Contours in 2D

- Assign gometric primitives to “cells” consisting of 2x2 grid points
  - – Line segments
- How do we know how to organize the primitives?
  - – Signs of the values of corners of cells

Contours in 2D

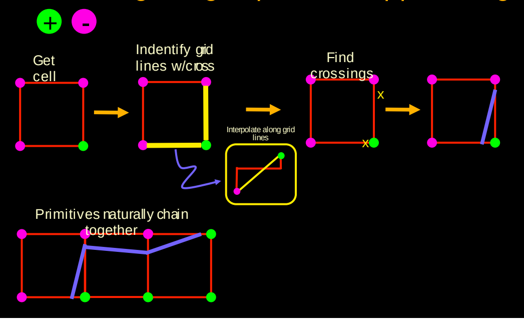
- Assign gometric primitives to “cells” consisting of 2x2 grid points
  - – Line segments
- How do we know how to organize the primitives?
  - – Signs of the values of corners of cells
- How do we know the position of the primitives?

Contours in 2D

- Assign gometric primitives to “cells” consisting of 2x2 grid points
  - – Line segments
- How do we know how to organize the primitives?
  - – Signs of the values of corners of cells
- How do we know the position of the primitives?
  - – Interpolate along grid points

Contours in 2D

- Idea: primitives must cross every grid line connecting two grid points of opposite sign

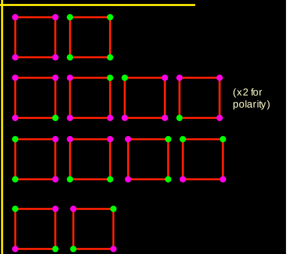


Questions

- How many grid lines with crossings can there be?
- What are the different configurations (adjacencies) of +/- grid points?

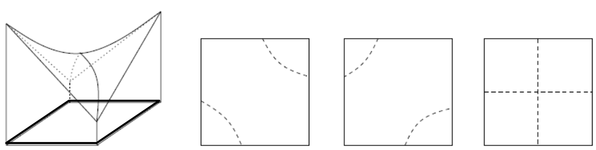
Cases

Case	Polarity	Rotation	Total
No Crossings	x2		2
Singlet	x2	x4	8
Double adjacent	x2	x2 (4)	4
Double Opposite	x2	x1 (2)	2
			16 = 2^4



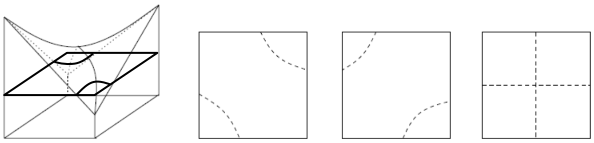
Ambiguities

- Right or wrong?



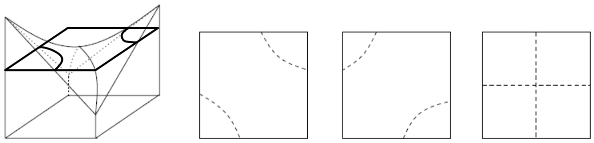
Ambiguities

- Right or wrong?



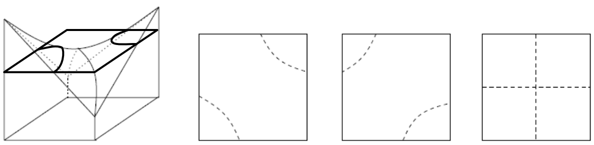
Ambiguities

- Right or wrong?



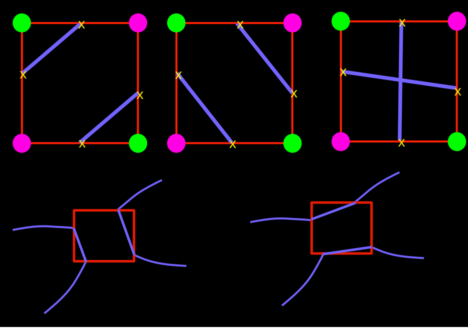
Ambiguities

- Right or wrong?



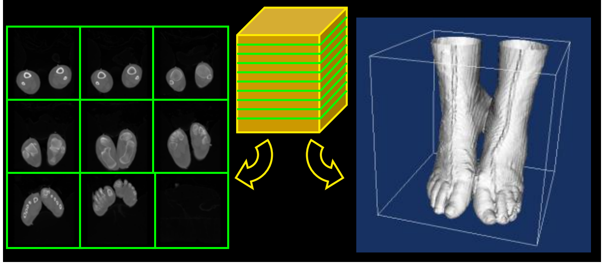
Ambiguities

- Right or wrong?



Isosurfacing

- You're given a big 3D block of numbers
- Make a picture
- Slicing shows data, but not its 3D shape
- Isosurfacing is one of the simplest ways



A little math

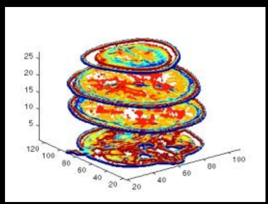
- Dataset: $v = f(x,y,z)$
- $f: R^3 \mapsto R$
- Want to find $S_v = \{(x,y,z) \mid f(x,y,z) = v\}$
- All the locations where the value of f is v
- S_v: <u>isosurface of f at v</u>
  - In 2D: isocontours (some path)
  - In 3D: isosurface
- Why is this useful?

Surface Extraction (Isosurfacing)

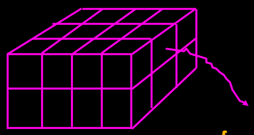
- Surface Extraction
  - SLICING - Take a slice through the 3D volume (often orthogonal to one of the axes), reducing it to a 2D problem
    - Contour in 2D
    - Form polygons with adjacent polylines

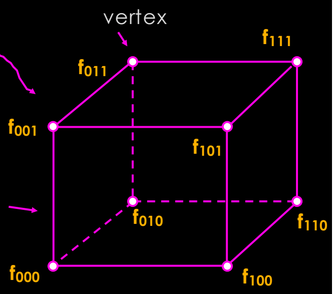
Note analogous techniques in 2D visualization:
1D cross-sections, and contours (=isolines)

Isosurface from slices



Notation





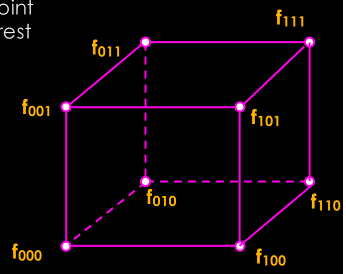
Volume of data

Each voxel transformed
to unit cube

Data Enrichment - Nearest Neighbour Interpolation

Value at any interior point taken as value at nearest vertex

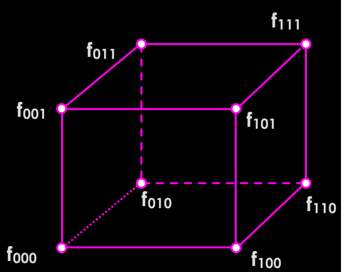
Fast
Discontinuous



Data Enrichment - Trilinear Interpolation

Trilinear interpolant is:

$$f(x,y,z) = f_{000}(1-x)(1-y)(1-z) + f_{100}x(1-y)(1-z) + f_{010}(1-x)y(1-z) + f_{001}(1-x)(1-y)z + f_{110}xy(1-z) + f_{101}x(1-y)z + f_{011}(1-x)yz + f_{111}xyz$$



Data Enrichment - Trilinear Interpolation

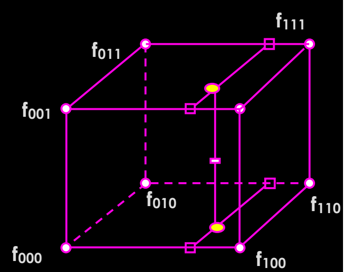
The value at

is found by:

(i) 4 1D interpolations in x

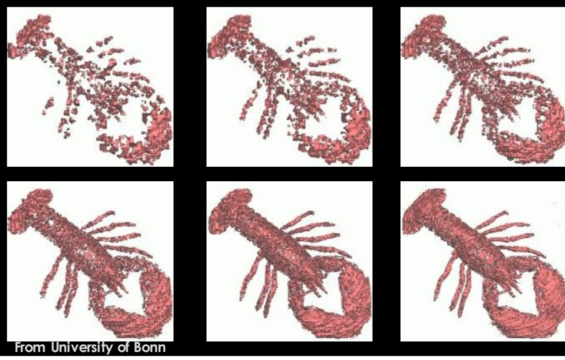
(ii) 2 1D interpolations in y

(iii) 1 1D interpolation in z



Isosurfacing

Lobster – Increasing the Threshold Level



From University of Bonn

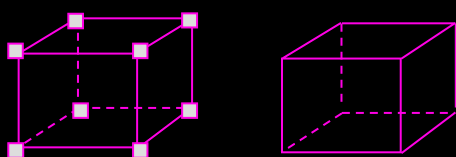
Isosurface Construction

- For simplicity, we shall work with zero level isosurface, and denote

positive vertices as

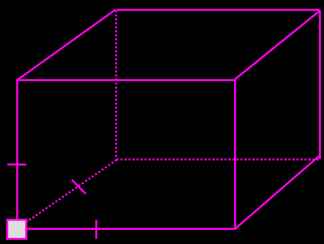
There are EIGHT vertices, each can be positive or negative - so there are $2^8 = 256$ different cases!

These two are easy!



There is no portion of the isosurface inside the cube!

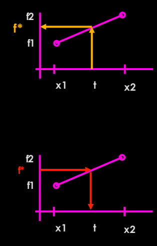
Isosurface Construction - One Positive Vertex - 1



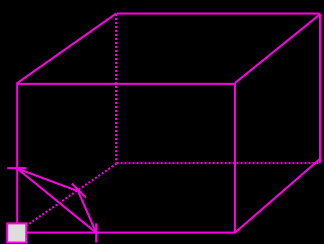
Intersections with edges found by inverse linear interpolation (as in contouring)

Note on Inverse Linear Interpolation

- The linear interpolation formula gives value of f at specified point t:
  f(x*) = f1 + t (f2 - f1)
- Inverse linear interpolation gives value of t at which f takes a specified value f*
  t = (f* - f1)/(f2 –f1)

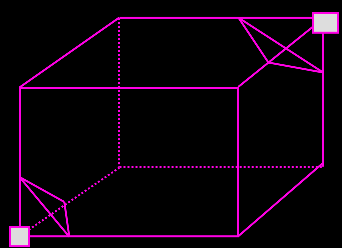


Isosurface Construction - One Positive Vertex - 2



Joining edge intersections across faces forms a triangle as part of the isosurface

Isosurface Construction -Positive Vertices at Opposite Corners



Isosurface Construction

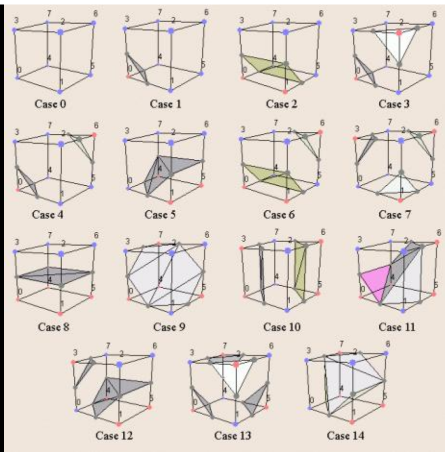
- One can work through all 256 cases in this way - although it quickly becomes apparent that many cases are similar.
- For example:
  - 2 cases where all are positive, or all negative, give no isosurface
  - 16 cases where one vertex has opposite sign from all the rest
- In fact, there are only 15 topologically distinct configurations

Canonical Cases for Isosurfacing

The 256 possible configurations can be grouped into these 15 canonical cases on the basis of complementarity (swapping positive and negative) and rotational symmetry

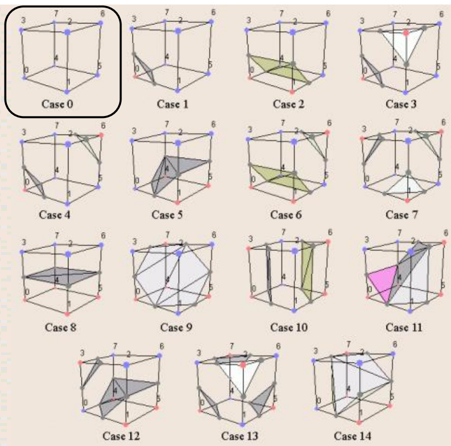
The advantage of doing this is for ease of implementation - we just need to code 15 cases not 256

Case Table



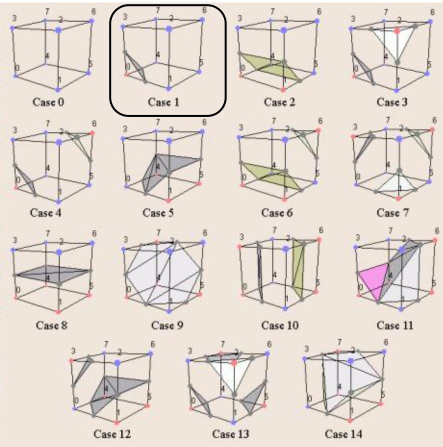
- 8 Above
- 0 Below

1 case



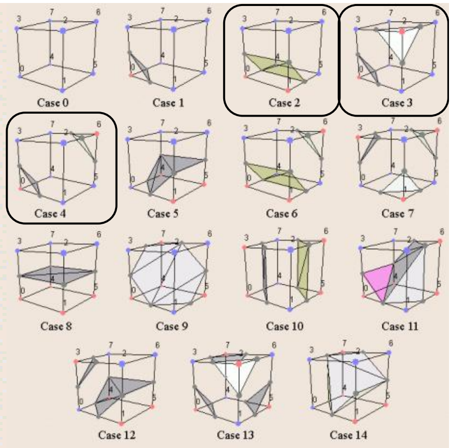
- 7 Above
- 1 Below

1 case



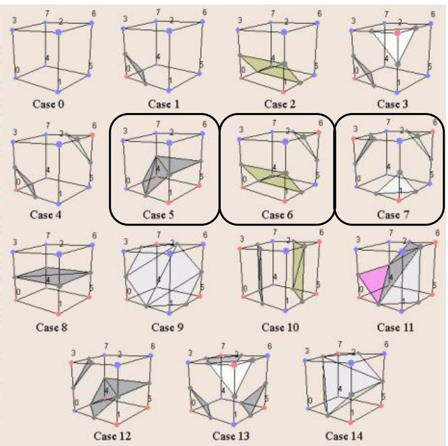
- 6 Above
- 2 Below

3 cases



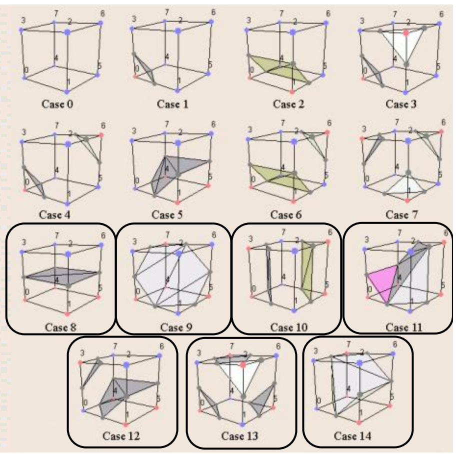
- 5 Above
- 3 Below

3 cases



- 4 Above
- 4 Below

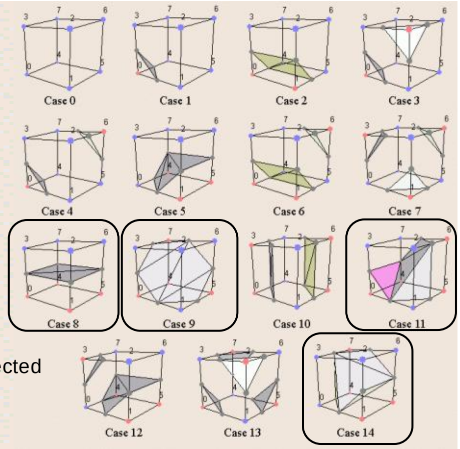
7 cases



- 4 Above
- 4 Below

7 cases

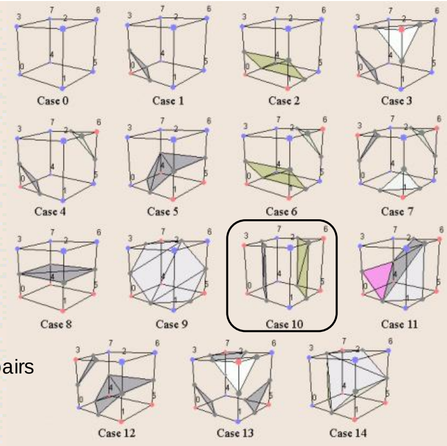
4 edge-connected



- 4 Above
- 4 Below

7 cases

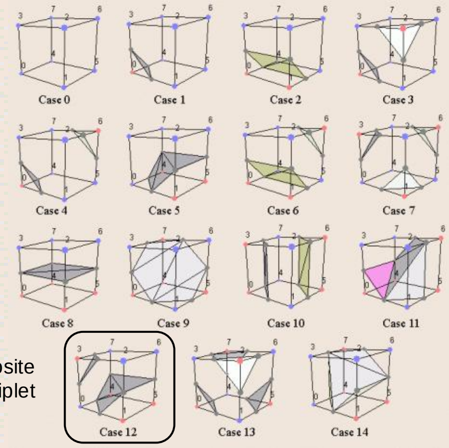
1 opposite pairs



- 4 Above
- 4 Below

7 cases

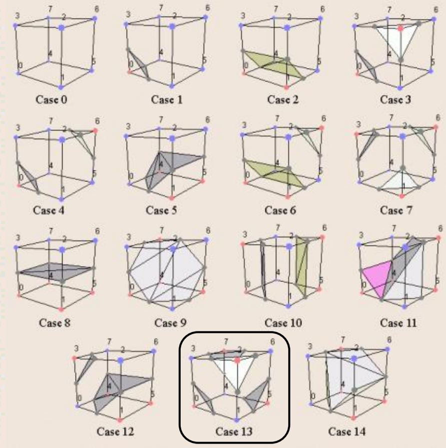
1 vertex opposite
to triplet



- 4 Above
- 4 Below

7 cases

1 isolated
vertices



Isosurface Construction

- In some configurations, just one triangle forms the isosurface
- In other configurations …
  - ...there can be several triangles
  - …or a polygon with 4, 5 or 6 points which can be triangulated
- A software implementation will have separate code for each configuration