Scientific Visualization

Scientific Datasets

- Gaining insight into scientific data by representing the data by computer graphics

- Scientific data sources
  - Computation
    - Real material simulation/modeling (e.g., molecular dynamics simulation, electronic calculations)
    - Solving differential equations (e.g., fluid dynamics, electro-magnetic field)
    - Climate modeling
  - Experiment
    - Medical and biological: magnetic resonance imaging, computer tomography, confocal microscopy,
    - Other data: 3D laser scanner, atomic force microscopy, seismic tomography

Data Challenges

- Scale
  - MRI dataset: 256^3 = 16 MB per slice (each slice is 3 micron thick)

    How many slices to cover a particular organ
  - A million-atom simulation: 7 GB per step (each step is 1 femtosecond)

    How many steps to simulate a particular physical/chemical/biological phenomenon

- Dimensionality
  - 3D volume data
  - 4D space-time data

- Scalar, vector and tensor data
  - Density or temperature distribution
  - Data from flow dynamics
  - Stress-strain data

Scalar Visualization Techniques

Scalar Dataset

- A single quantity that can be expressed as a function of position in space

  $$S = S(x,y,z)$$

  Array ***S*** represents data at discrete locations in space

- Describe the value at any continuous location by defining an interpolation function $F(x,y,z)$

- Volume data (MRI, confocal, finite element modeling)

- Represented through regular grids
  If irregular grids, preprocessing of data to regular grid

- Each data element (cube or cell) often called **voxel**

Different Rendering Techniques

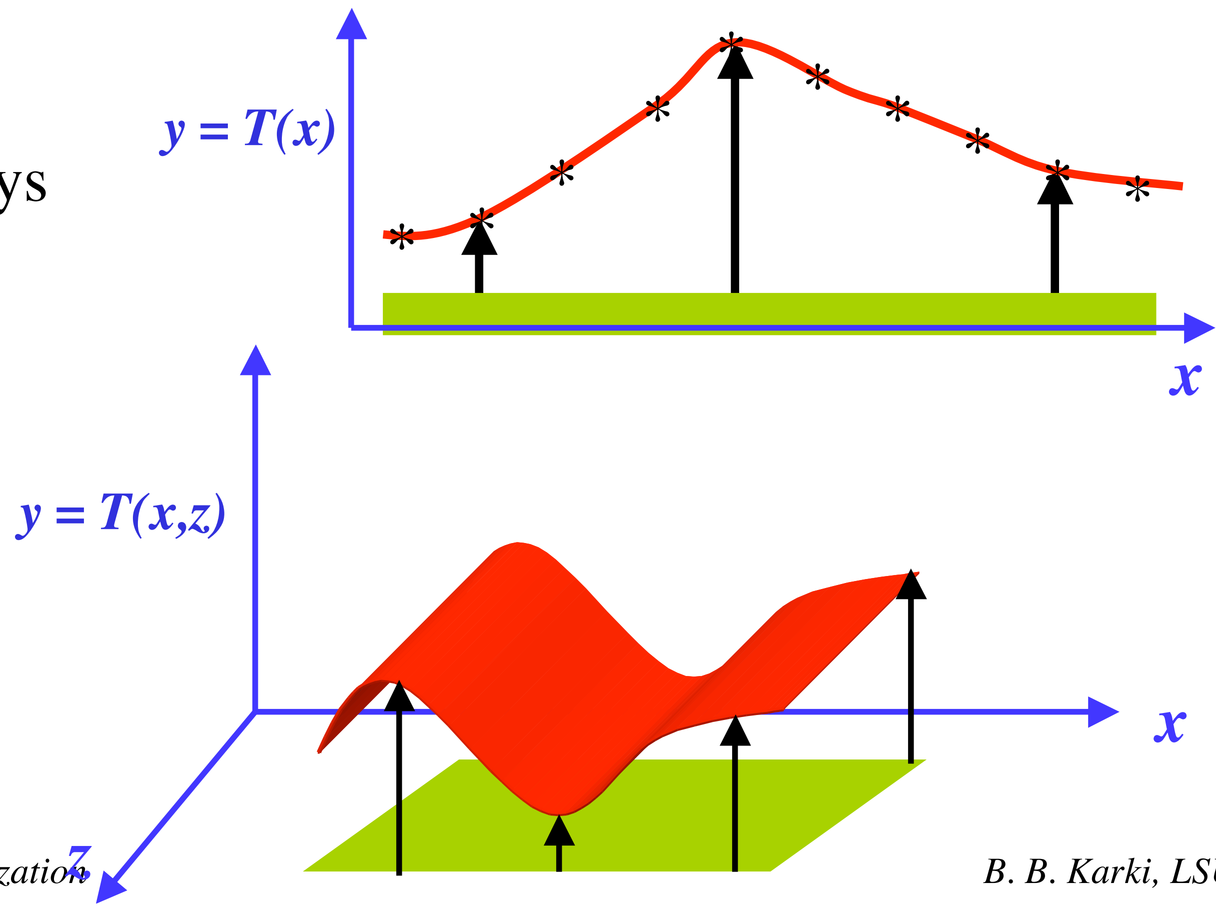
- Simple approaches
  - Symbols, Color mapping, Contour display
- Isosurface rendering
  - Marching cubes algorithm, Fast extraction approaches
- Implicit surfaces
  - Particle sampling, Dividing cubes algorithm, Shape function interpolation
- Volume slicing
  - Clipping, Sampling planes, Interactive clipping, Clip objects
- Volume rendering
  - Object-oriented, Image-oriented, Hybrid techniques

Simple Approaches

Symbols or Off-Path Displays

- Useful for displaying one or two dimensional scalar data
    - ➢ Temperature distribution along a rod or on sheet

- Off-path displays



Color Mapping: Lookup Table

- Useful for scalar visualization in 1D, 2D or 3D
- Map scalar data to colors to display on the screen
- Lookup table:
  - ➢ Holds an array of colors (RGB components)
  - ➢ Scalar values serve as indices

    For each s_i, there is index i

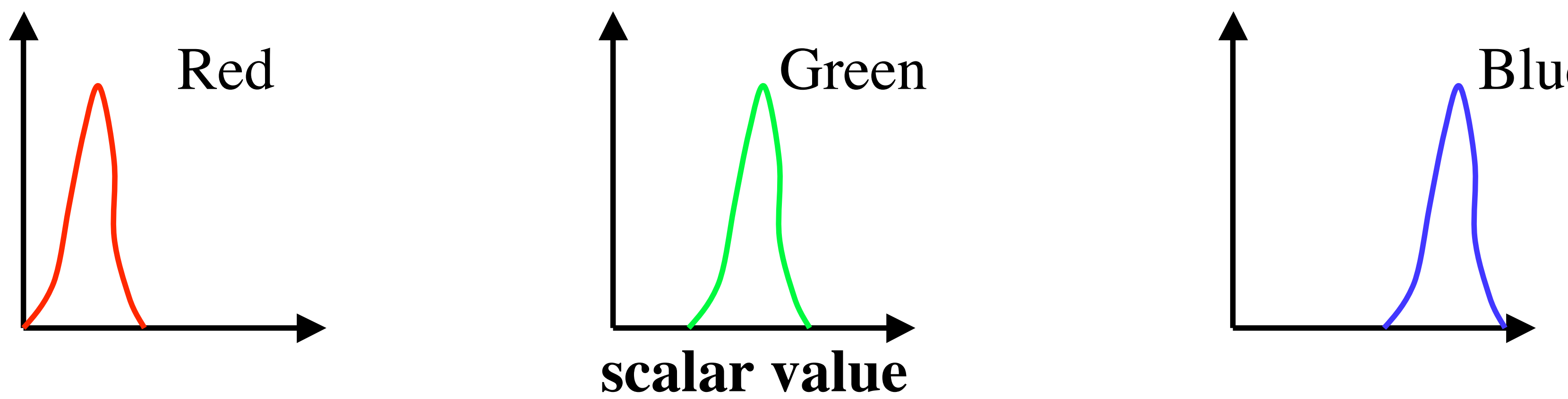
$$i = n\left(\frac{s_i - \min}{\max - \min}\right)$$

i

rgb0
rgb1
rgb2
.
.
rgbn-1

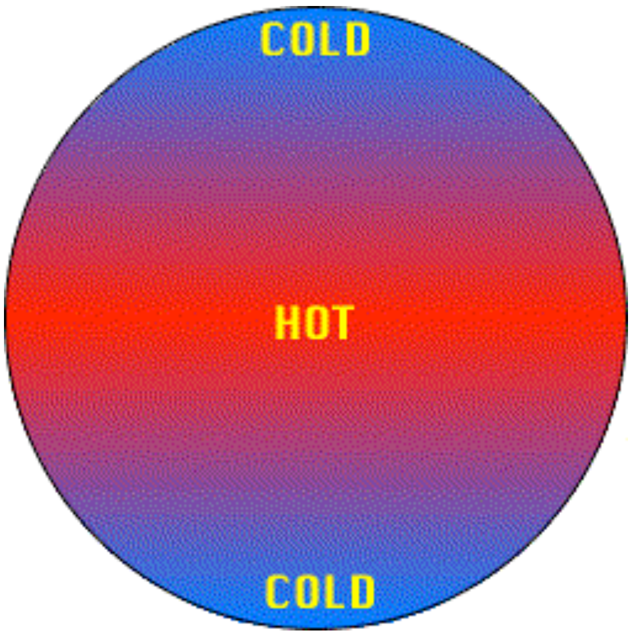
Color Mapping: Transfer Function

- Transfer function
  - ➢ An expression that maps the scalar value into a color specification
  - ➢ Mapping to separate intensity values of R, G and B



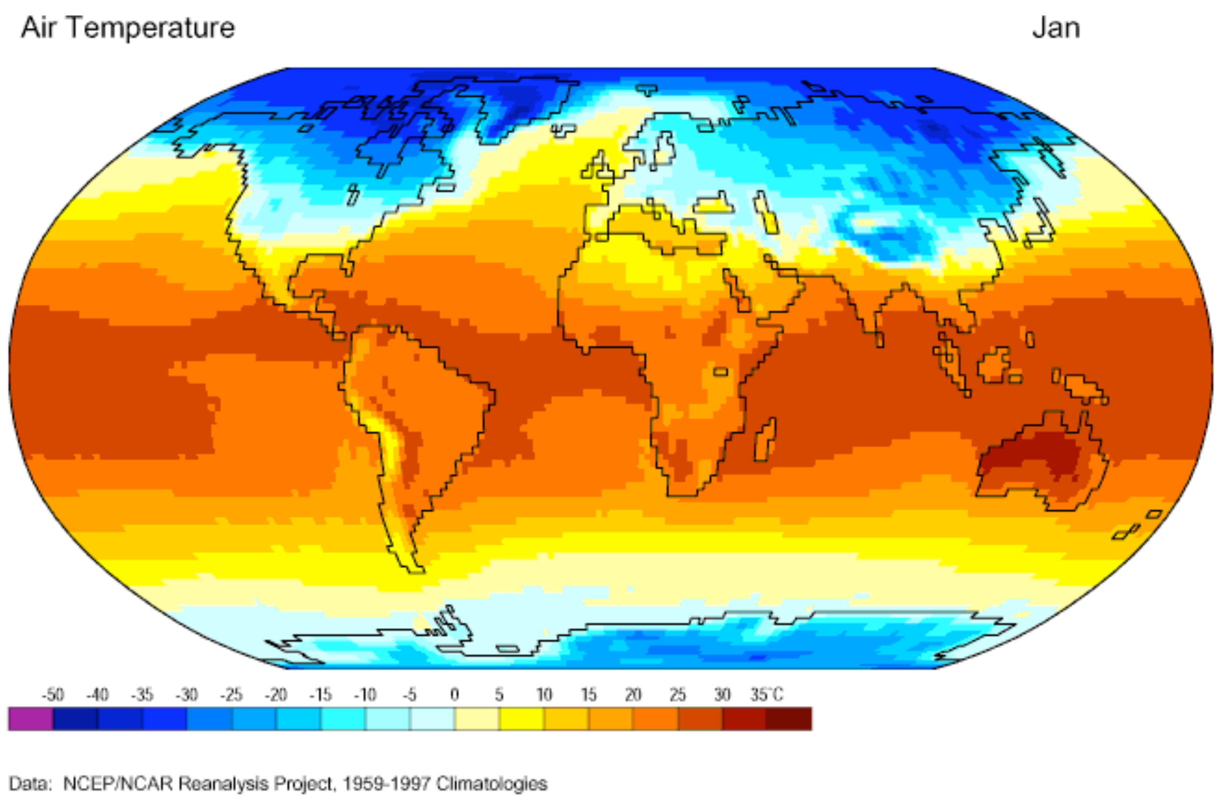
- A lookup table is a discrete sampling of a transfer function

Examples of Color Mapping



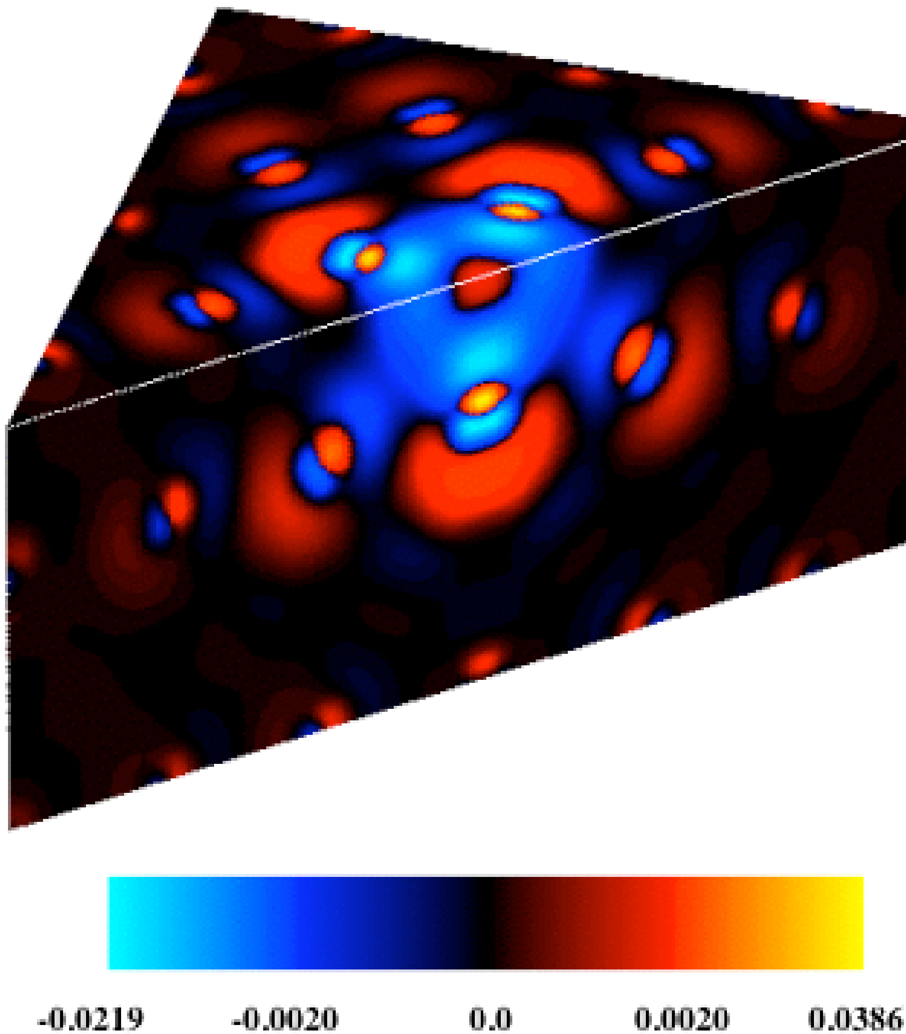
Simple latitudinal zonation temperature

Mean January air temperature on the Earth's surface



www.geog.ouc.bc.ca

Multiscale Color Mapping



Two-level mapping:

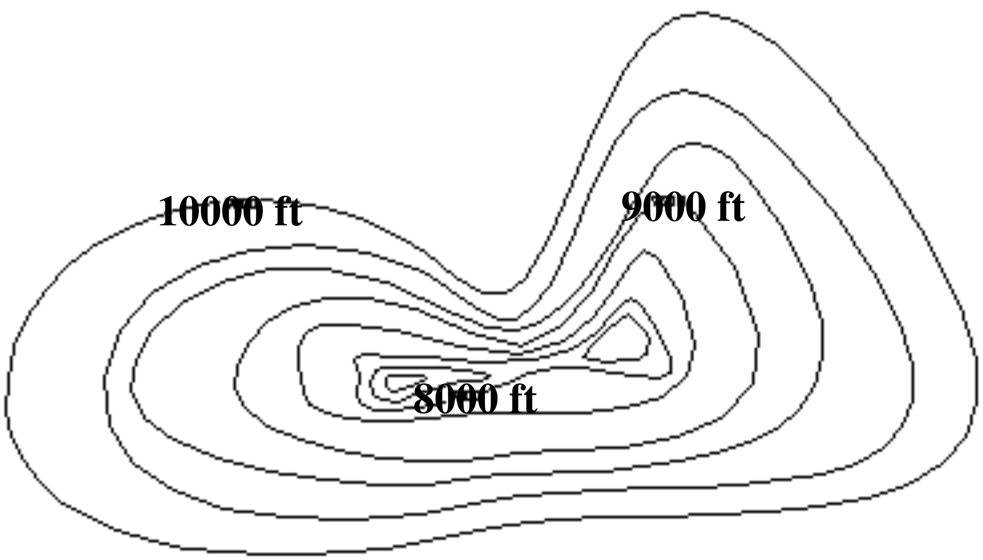
Fine-level scale uses the red and blue colors to represent the positive and negative differences with magnitude up to 0.002 (in units of $Å^{-3}$)

Coarse-level scale adds green color component to red and blue colors to map the positive and negative differences with magnitudes higher than 0.002.

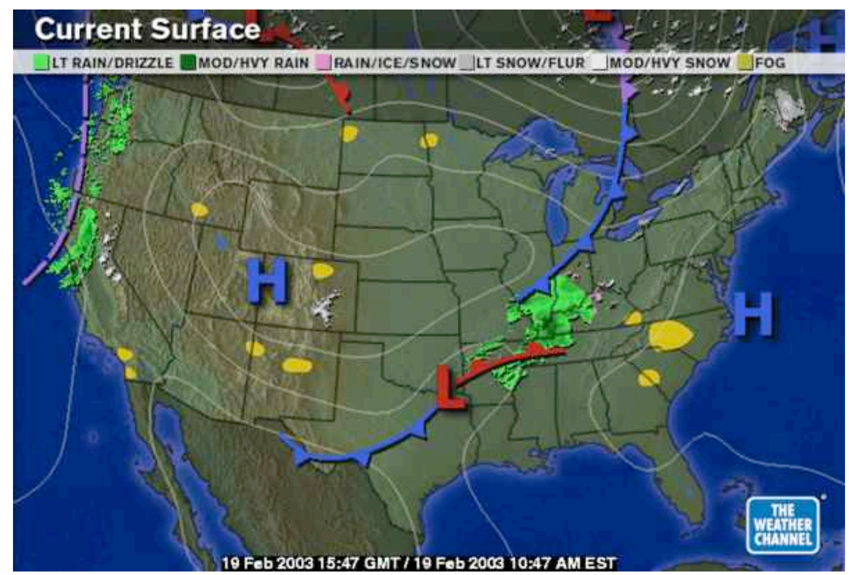
B. B. Karki, LSU

Contour Display

- Common method for displaying scalar data across a surface
- Contour lines: represent a constant value across the surface (isovalue lines)



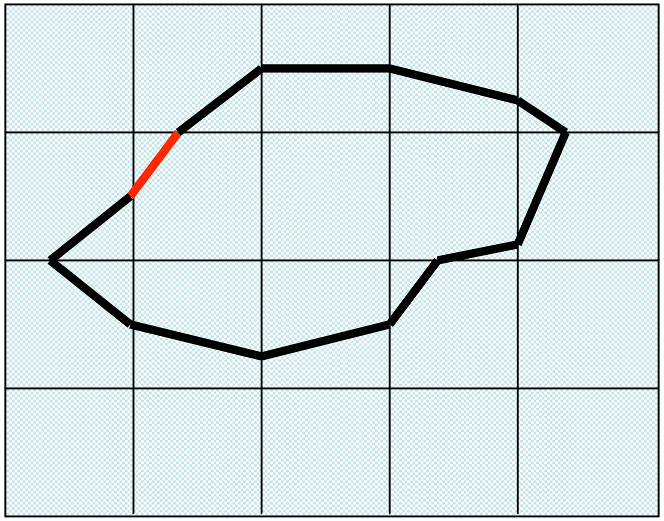
Topographic Map



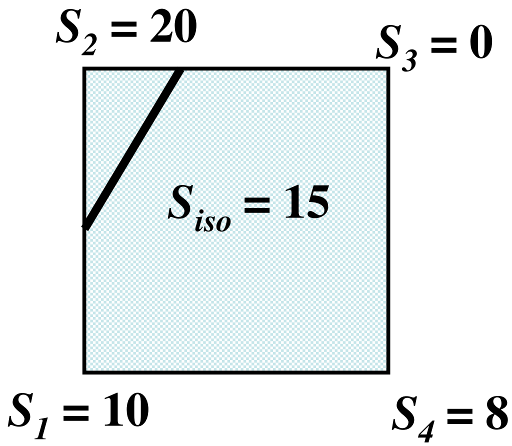
Weather Map

2D Contour Lines

Edge Tracking Algorithm



- Select an element or cell
  - Consider a 4-vertex quadrilateral element with scalar values S_1, S_2, S_3 and S_4
- If all S_i's > S_{iso} or all S_i's < S_{iso}, no contour line passes through the element
- Otherwise, start at the first pair of vertices, determine if the isovalue exists along the edge
  - If one vertex value > S_{iso} while the other vertex value < S_{iso}, isovalue exists, in either order
  - If not, proceed in either clockwise or anticlockwise order until an edge containing the isovalue is found



Edge ….

- Once an edge with S_{iso} is found between vertices i and j, compute isovalue location along the edge by linear interpolation

$$x = x(i) + fac * (x(j) - x(i))$$

$$y = y(i) + fac * (y(j) - y(i))$$

Where $fac = \left(\dfrac{S(j) - S_{iso}}{S(j) - S(i)} \right)$

  - ➢ This isovalue location will be the first point of the contour line

  Default location: mid point

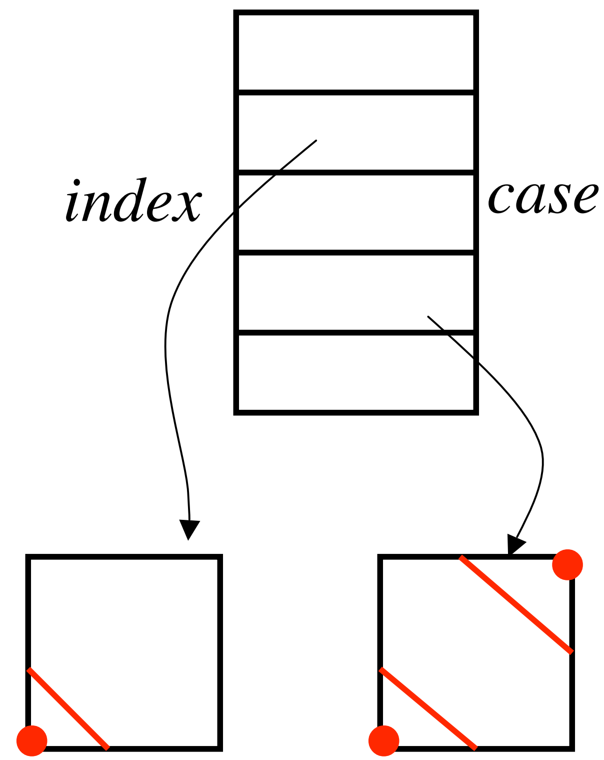
- Examine each subsequent edge until the next edge containing an isovalue is found and repeat previous step
  - ➢ Connect these two points to form the contour segment
  - ➢ Use shape function to give isolines some curvature.

Marching Squares Algorithm

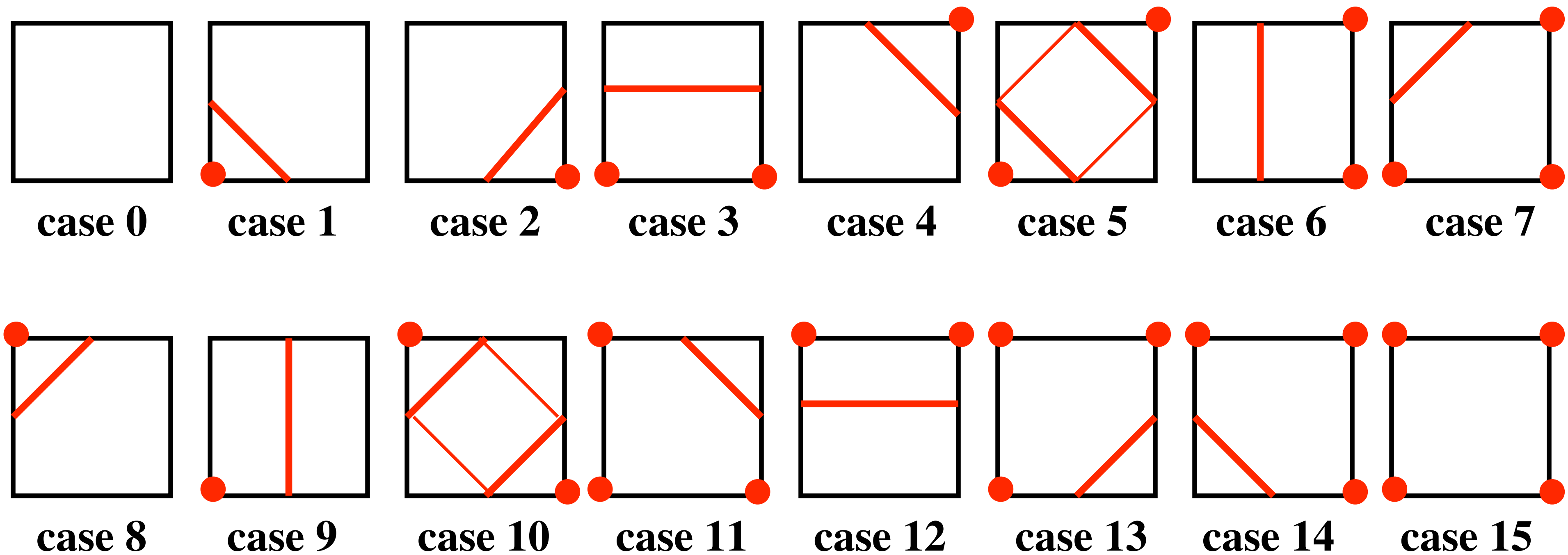
- Select a square element or cell
  - Values at four corners
    - Below isovalue (marked)
    - Above isovalue (unmarked)

- Calculate inside or outside state of each vertex of the cell

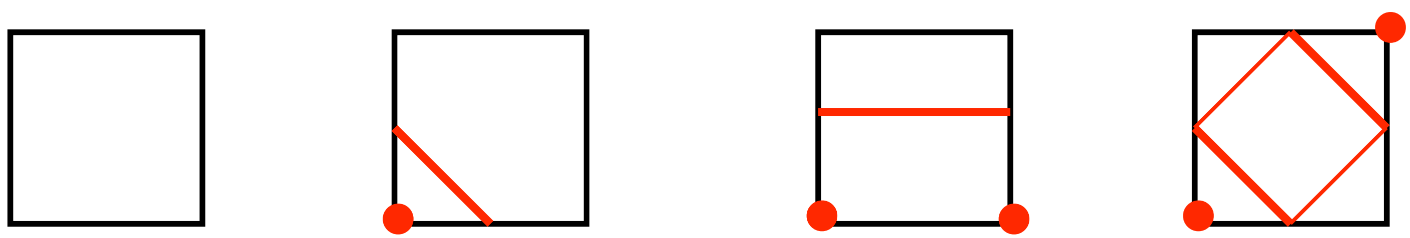
- Determine the topology state of the cell by referring to a case table that has a list of all possible configurations
  - Each square is either inside, outside or intersected
  - 2D cell index: 4-bit, 2^4 (16) cases
- Calculate the contour location (via interpolation) for each edge in the case table
  - No or one intersection per edge



Cases of 2D Cells (Squares)

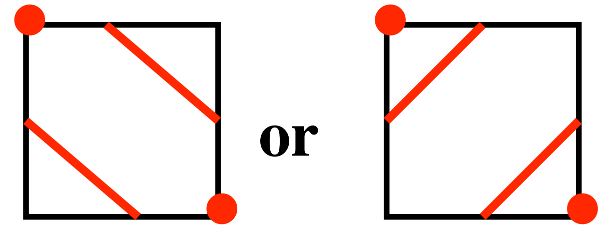


By complementary and rotational symmetries (equivalence), the number of the basic cases is reduced to 4



2D Ambiguous Cases

- Ambiguous cases:
  - **5, 10**

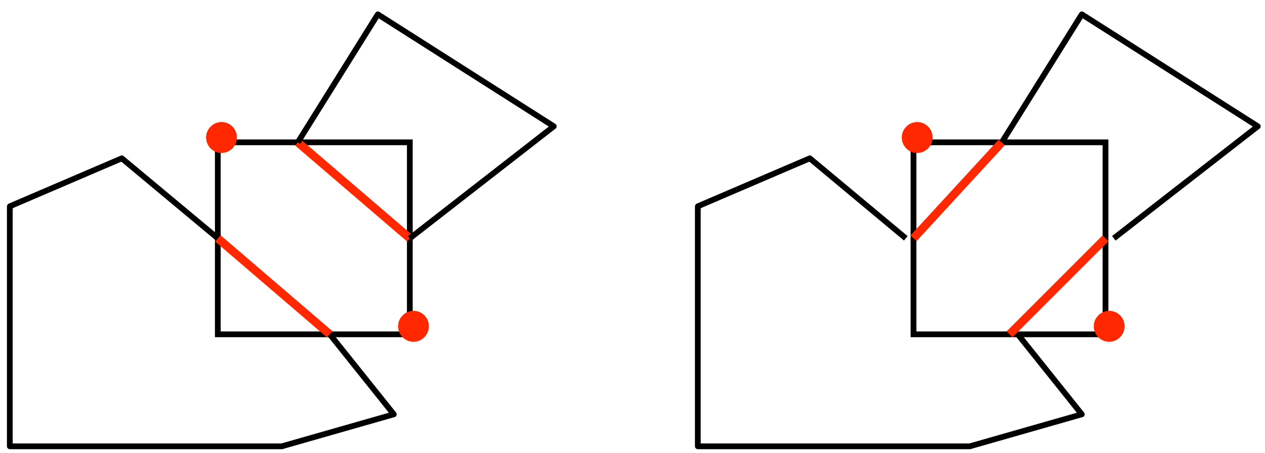


- Contour ambiguity arises when adjacent vertices in different states but diagonal vertices in the same state

- Break contour

  Join contour

- Both are valid

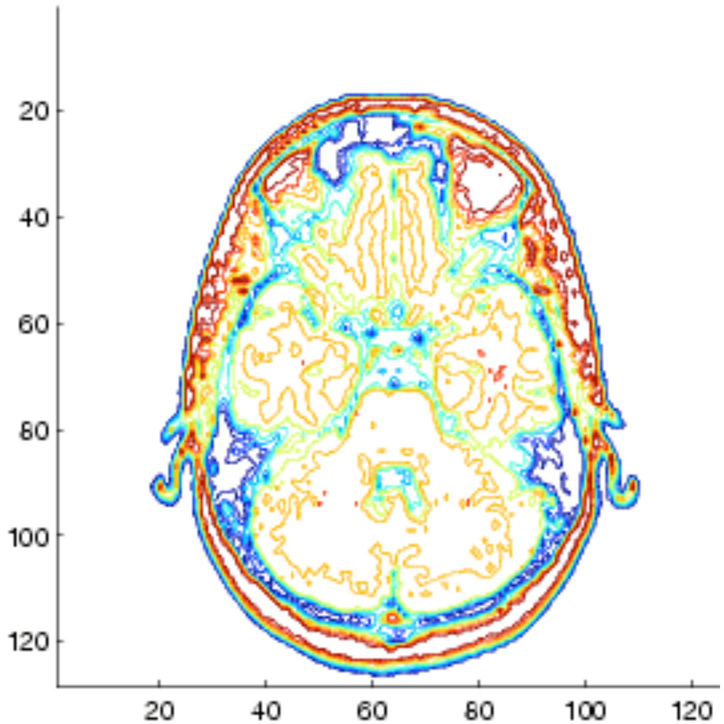


Break contour (two loops)

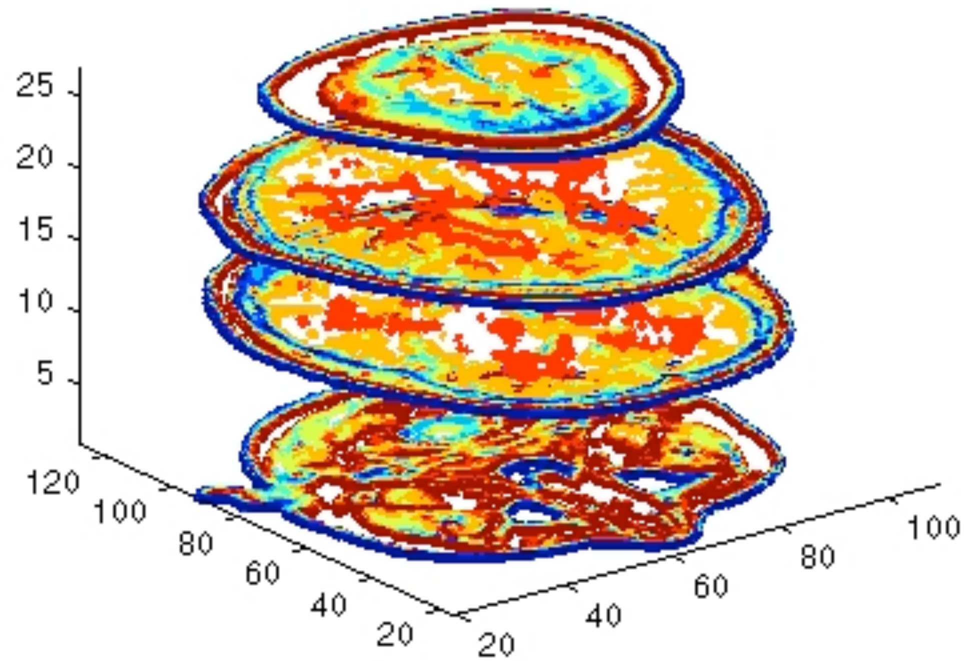
Join contour (single loop)

Contour Lines of MRI Data

Contour display of MRI data of a human head (single image and a stack of four images)



2D contour



3D contour