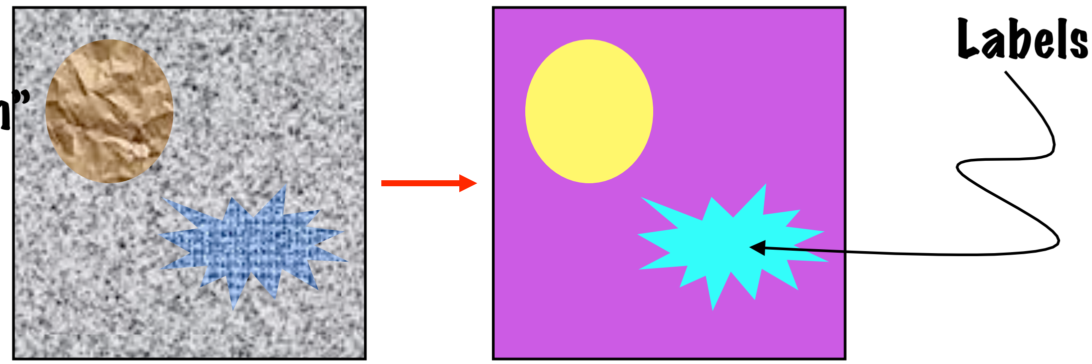
Image Segmentation

Ross Whitaker
SCI Institute, School of Computing
University of Utah

What is Segmentation?

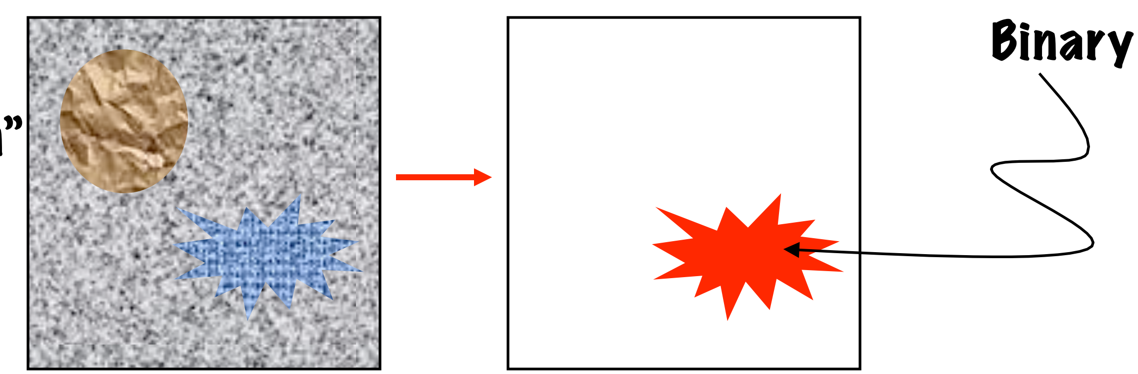
- Partitioning images/volumes into meaningful pieces

"Partitioning problem"



- Isolating a specific region of interest ("find the star" or "bluish thing")

"Delineation problem"



Why?

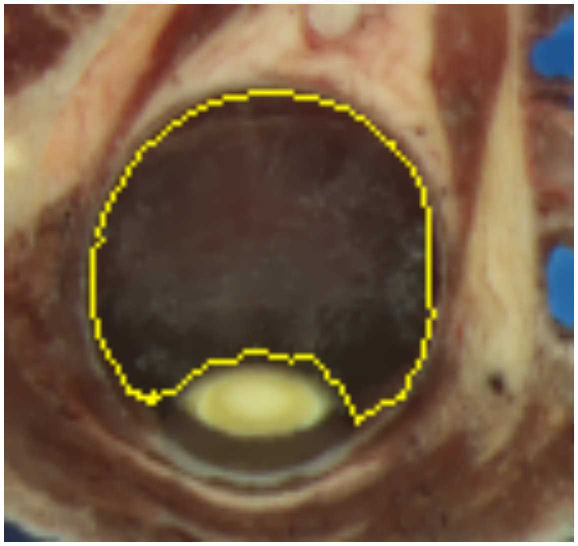
- Detection/recognition
  - Where is the vehicle?
  - What type of vehicle is it?
- Quantifying object properties
  - How big is the tumor? Is is expanding or shrinking?
  - How much of a radiation do I need to treat this prostate?
  - Statistical analyses of sets of biological volumes

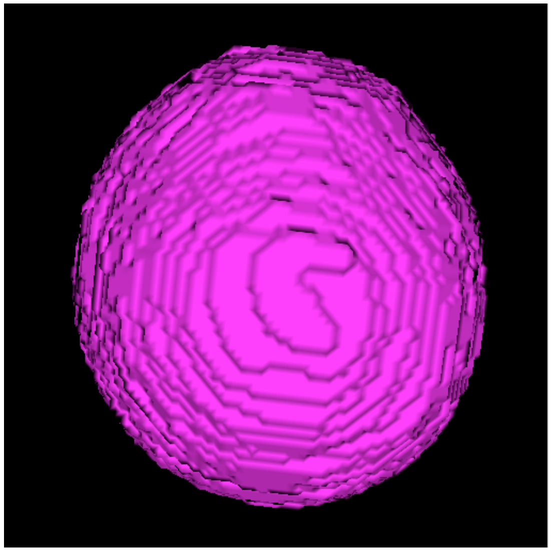
What is The Best Way to Segment Images?

- Depends…
  - Kind of data: type of noise, signal, etc.
  - What you are looking for: shape, size, variability
  - Application specifics: how accurate, how many
- State of the art
  - Specific data and shapes
    - Train a template or model (variability)
    - Deform to fit specific data
  - General data and shapes
    - So many methods
    - So few good ones in practice: hand contouring

Hand Contouring

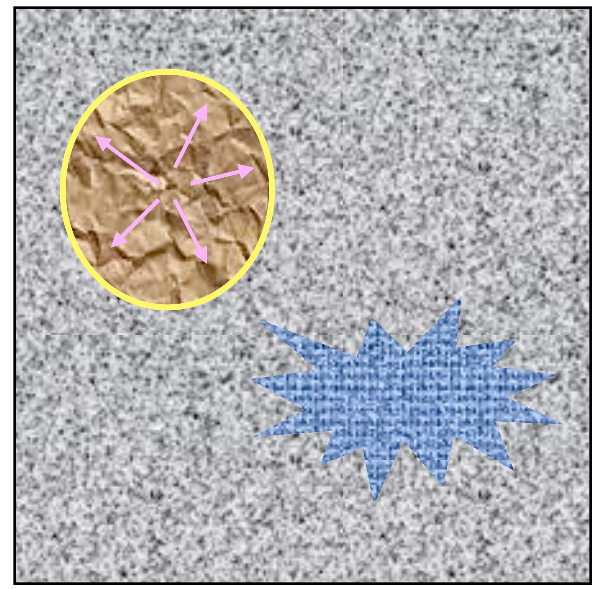
- "Quick and easy" general-purpose seg tool
- Time consuming



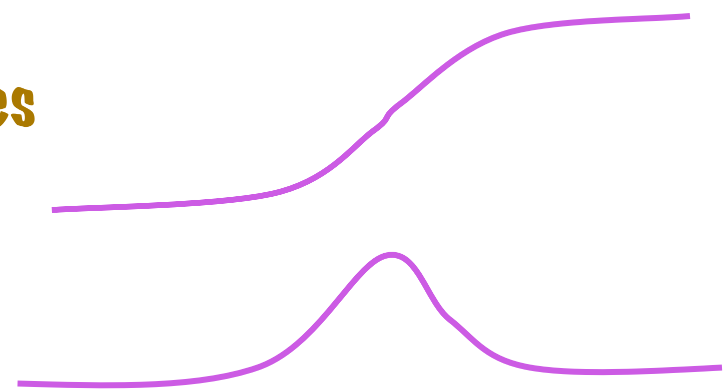


- 3D: slice-by-slice with cursor defining boundary
- User variation (esp. slice to slice)
- Tools available. E.g. Harvard SPL "Slicer"

General Purpose Segmentation Strategies



- Region-based methods (connected)
  - Regions are locally homogeneous (in some property)
  - Regions satisfy some property (to within an tolerance)
  - E.g. Flood fill
- Edge-based methods
  - Regions are bounded by features
  - Features represent sharp contrast in some property (locally maximal constrast)
  - E.g. Canny Edges



Pixel Classification

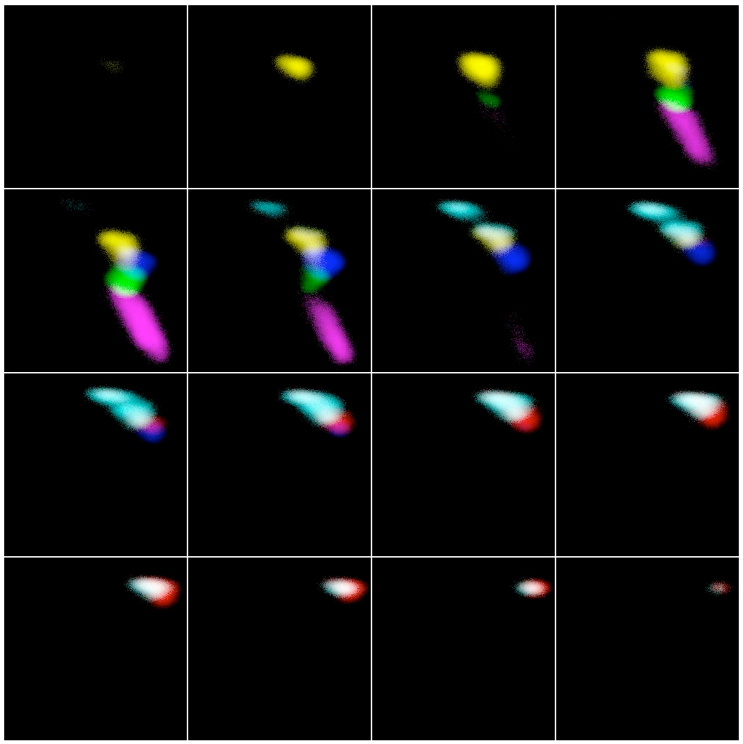
- Simplest: Thresholding
  - Pixels above threshold in class A, below class B
  - Connected components on class label
- Extension of thresholding -> pattern recognition
  - Image intensities not enough
  - Define set of "features" at each pixel

Options for Pixel Features

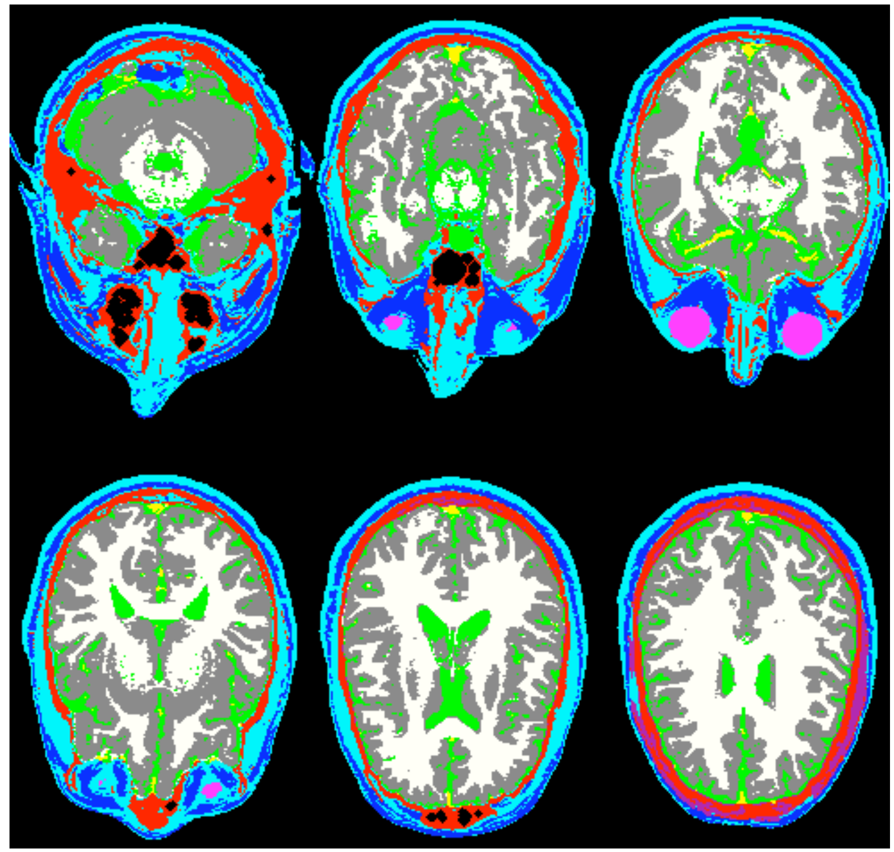
- Intensity
- Derivatives (at different scales)
  - Also differential invariants (e.g. grad mag)
- Neighborhood statistics
  - Mean, variance
  - Neighborhood histogram
  - Texture (e.g. band-pass filters)
- Multivariate data (vector-valued range)
  - Color
  - Spectral MRI

Spectral MRI Classification

T1, T2, PD



Feature Space



Classification

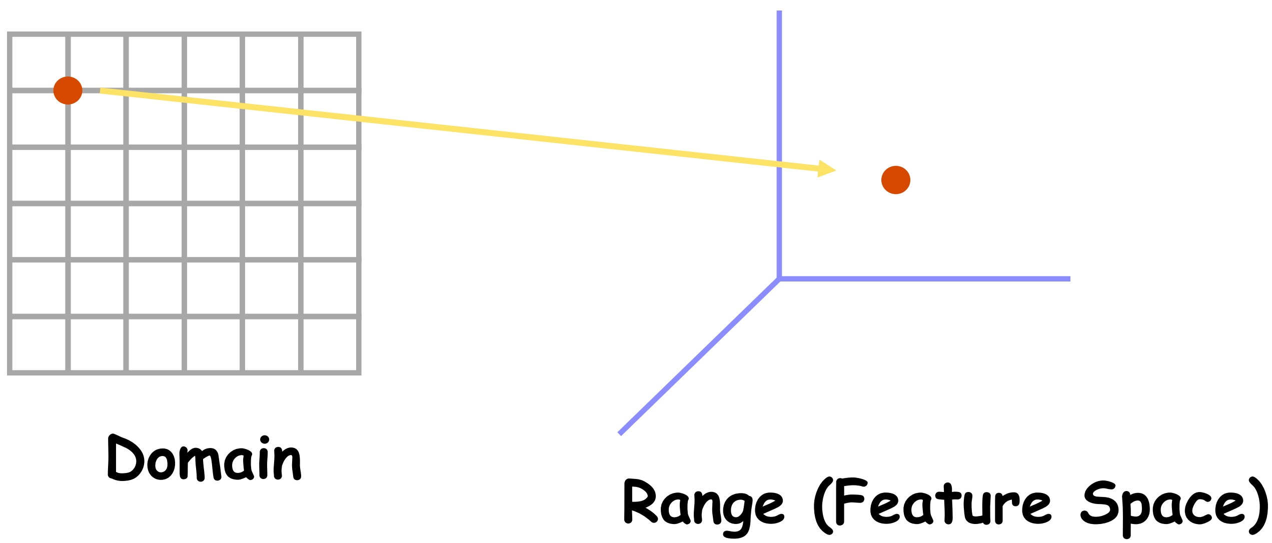
Tasdizen et al.

Pattern Recognition

- Relatively "old" idea (mid 20th century)
- Classify an instance based on multiple measurements (features)
- Statistical decision theory (min. prob. of error)
- For each set of measurements say which class and (maybe) prob.

Concept - Feature Vector

- Set of measurements $x \in \Re^n$
- Position in feature space $x = (x_1, x_2, \ldots, x_n)$



Classification

- Typical approach: construct a function which tells you the extent to which x is in class I

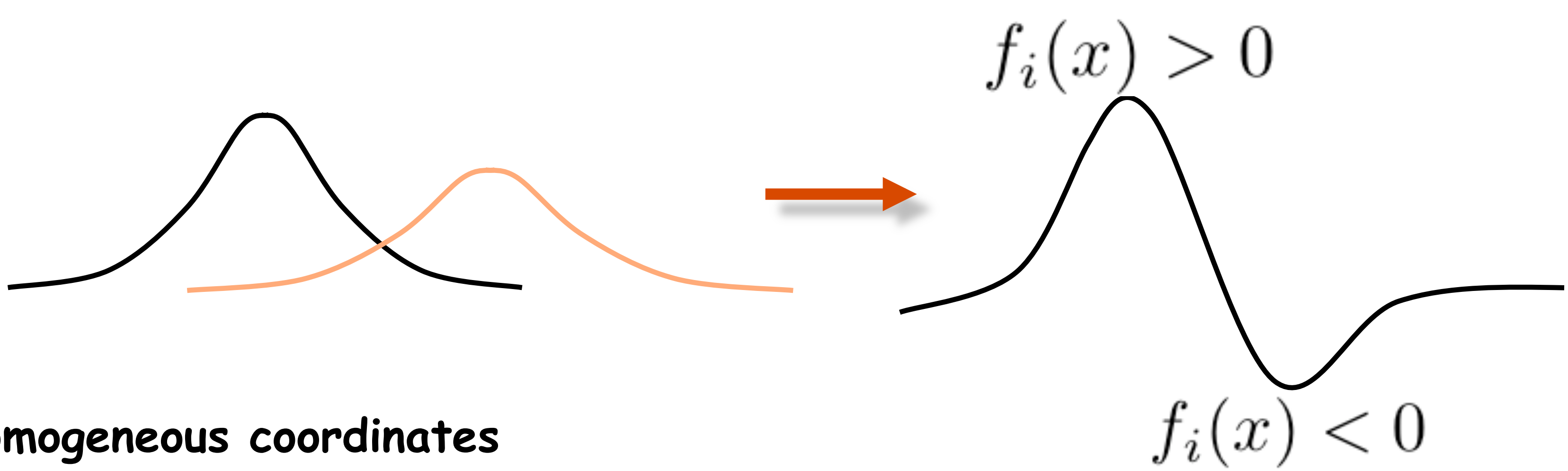
$$f_i : \Re^n \mapsto \Re$$

- Two types of problems
  - Supervised - classified examples are given
  - Unsupervised - only raw data is given

Pattern Recognition

- What is the form of f()?
- Could be anything, but...
  - Linear
  - Difference of Gaussians

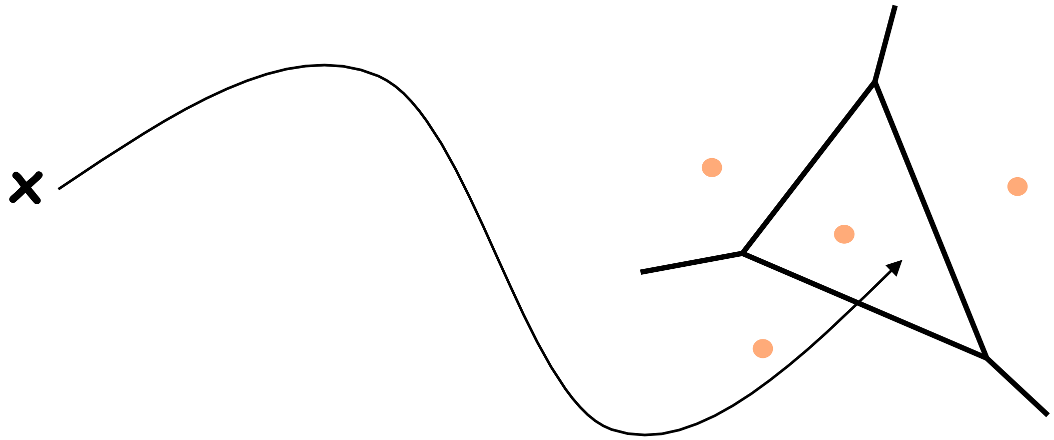
$$f_i(x) = x^* \cdot w_i$$



*homogeneous coordinates

Finding f() From Examples

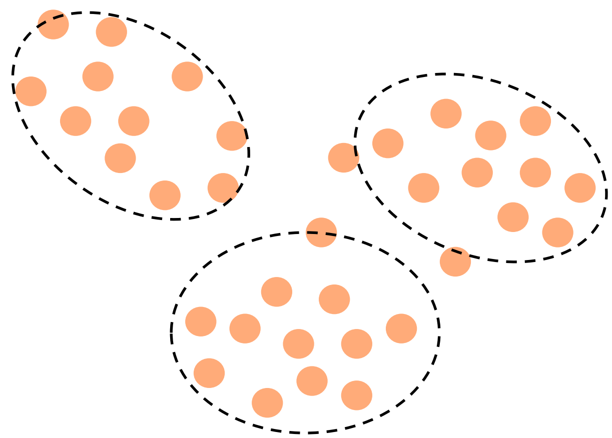
- For each class use *prototype*
  - Classify instance based on nearest prototype



- Neural nets (e.g. perceptron)
  - Learn set of parameters (e.g. Ws) through many examples
- Statistical
  - Construct probability density functions from examples

Unsupervised

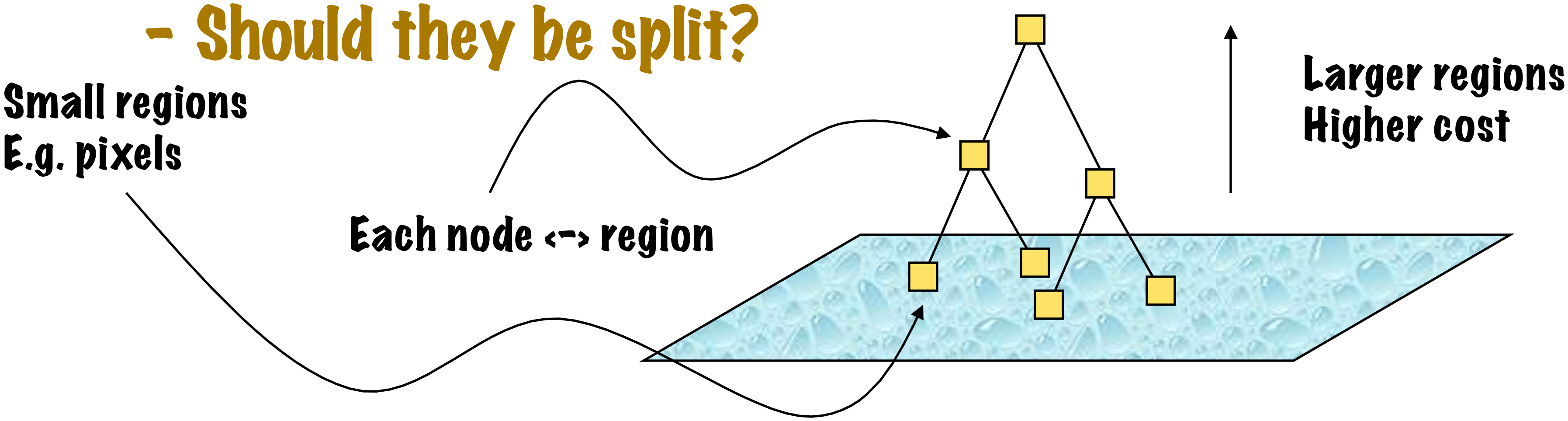
- Find natural structure in data
- E.g. clusters



- K-means alg.
  - Start with k centers (random)
  - Find set of points closest to each center
  - Move center to mean of points
  - Repeat until centers don't move

Hierarchical Grouping Methods

- Splitting, merging of regions
- Construct metric on region configurations M(i)
  - Statistics of region (average intensity, etc).
  - Are two regions similar enough to be merged?
  - Should they be split?

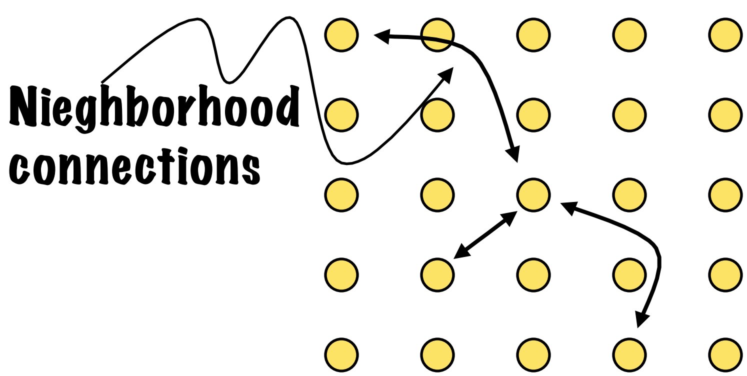


Simple Merging Alogoritm

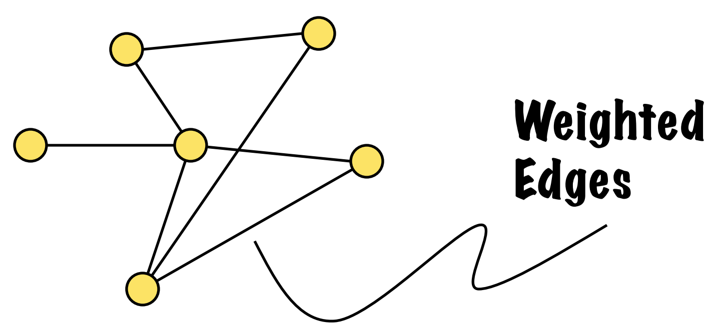
1. Each pixel -> one region
2. For each region, check merge with each neighbor
3. Cost of merge $C(i,j) = M(i \cup j) - [M(i) + M(j)] + k$
4. Sort by cost (e.g. heap) and merge min: region j <- iUj
5. Stop at number of regions or no more merges

Minimum Cut (Shi and Malik `00)

- Treat image as graph
  - Vertices -> pixels
  - Edges -> neighbors
  - Must define a neighborhood stencil (the neigbhors to which a pixel is connected)



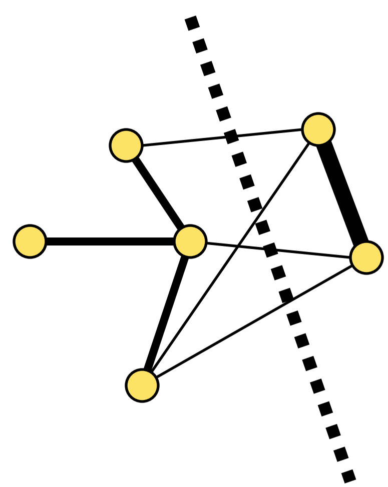
Image



Graph

Minimum Cut - Edge Weights

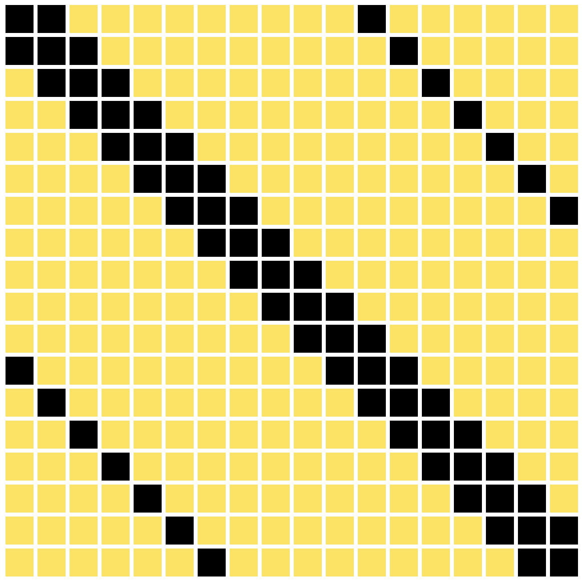
- Edge weights
  - Pixel distance, edges (e.g. Gaussian fall off)
- Say how many regions you want
- Cut graph so that "flow" between regions is minimized (min cut)



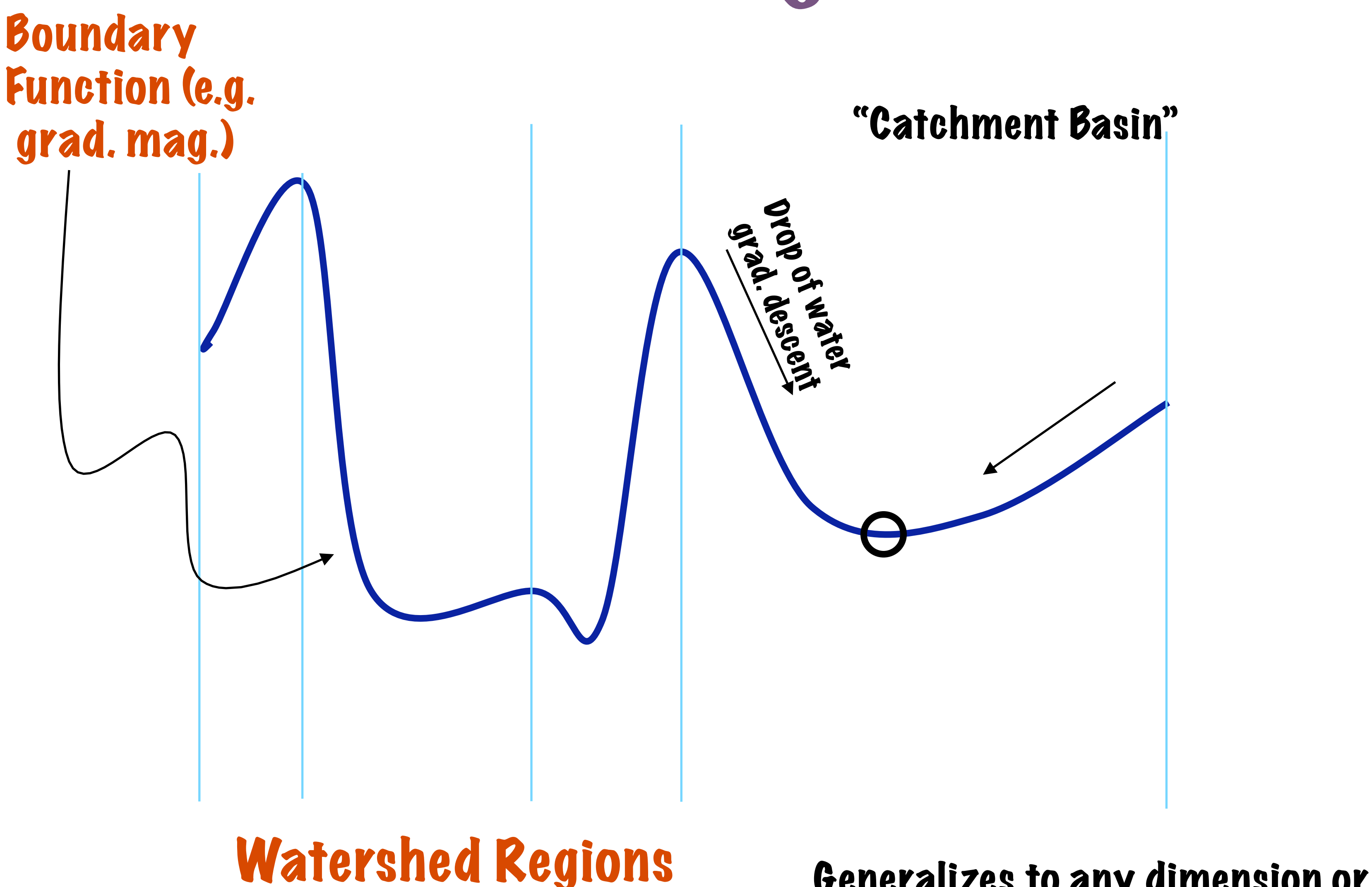
Min cut

Minimum Cut – Solving

- NxN matrix (N number of pixels)
- Min eigen value/vector discribes min cut
- Computationally expensive, but…
- Matrix is sparse because of neighborhood structure
  - I.e. most connections are zero
- Run recursively to get more regions

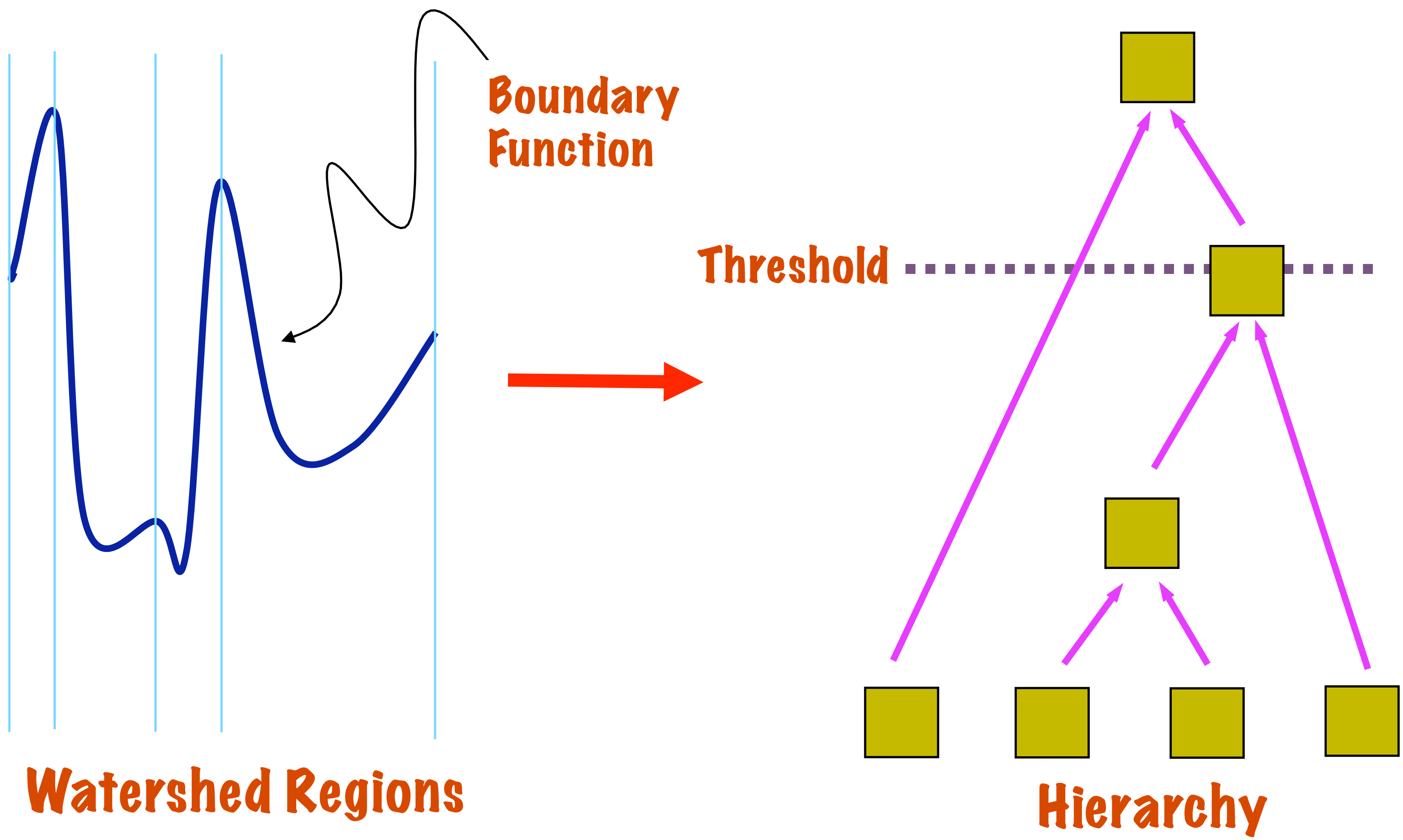


Watershed Segmentation



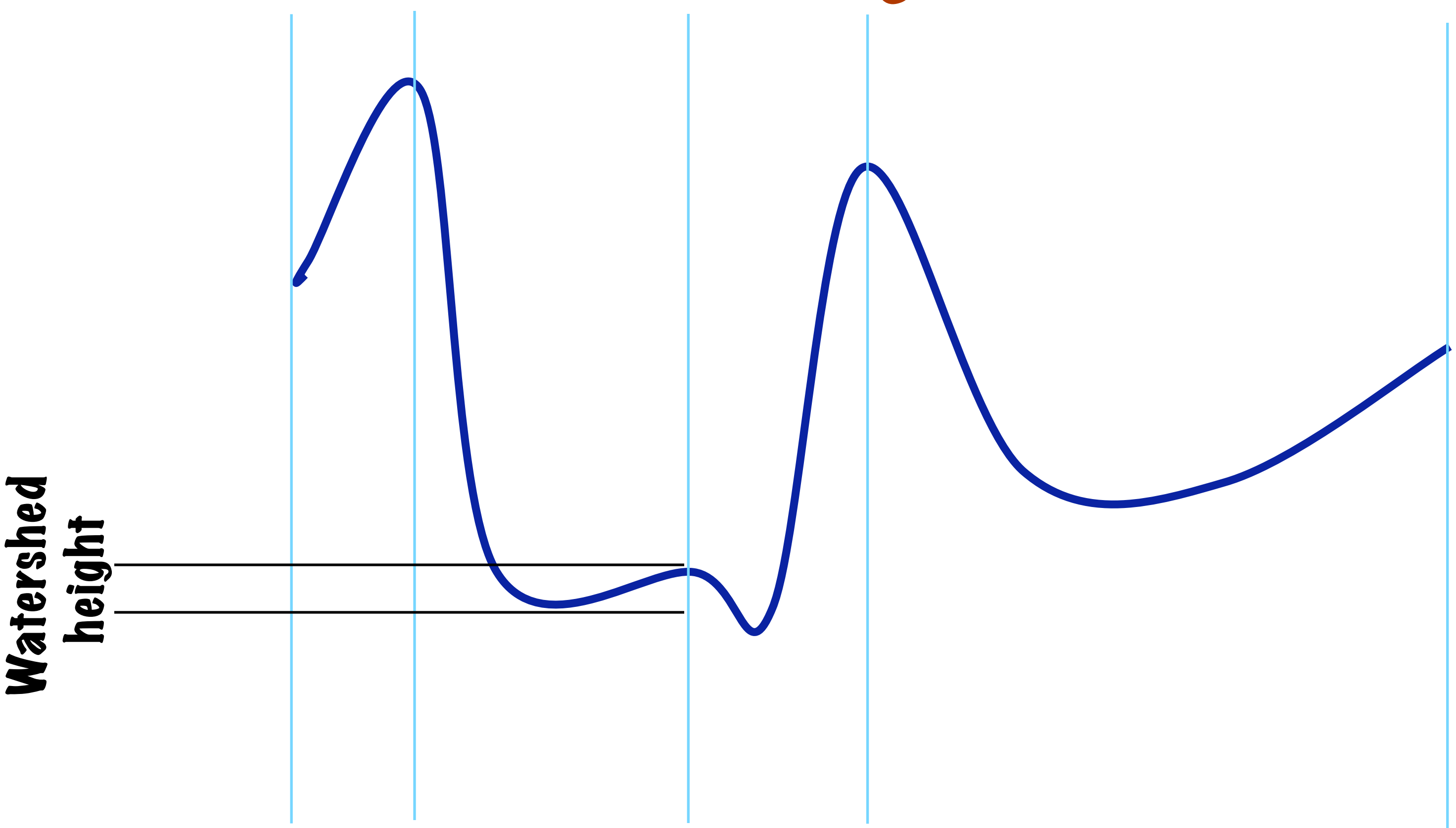
Generalizes to any dimension or boundary measure

Watershed Segmentation



Watershed Saliency

- Height of water before “flooding” neighbor
- Used as “cost of merge” to build hiearchy



Watershed Segmentation Properties

- General
- Non-local - regions can leak
- Boundary based
  - Poor in low-contrast data
  - Sensitive to noise
- Low level (pixel based)
  - Lack of shape model
- Preprocessing
  - Necessary for reliable boundary measure

Edge/Region-Based Segmentation Pipeline

- Noise
  - Filtering/smoothing

- Inhomogeneities
  - Correct for them
  - Model them
  - Use edges (local)

Image/Volume Data

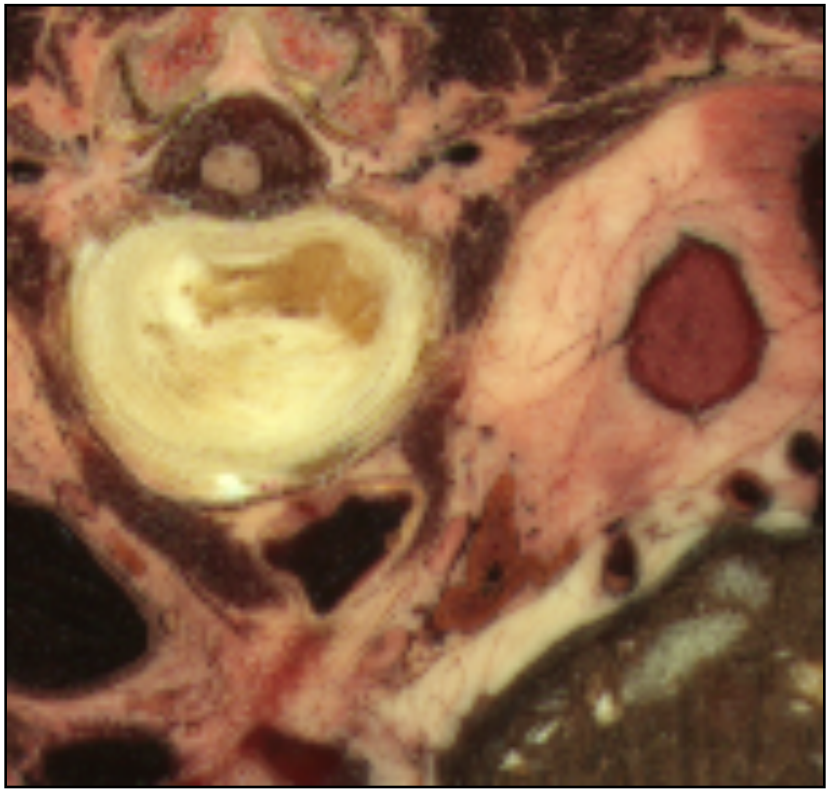
Filtering
- Linear scale-space
- Nonlinear diffusion

Feature Extraction
- Differential Geometry
- Data Fusion

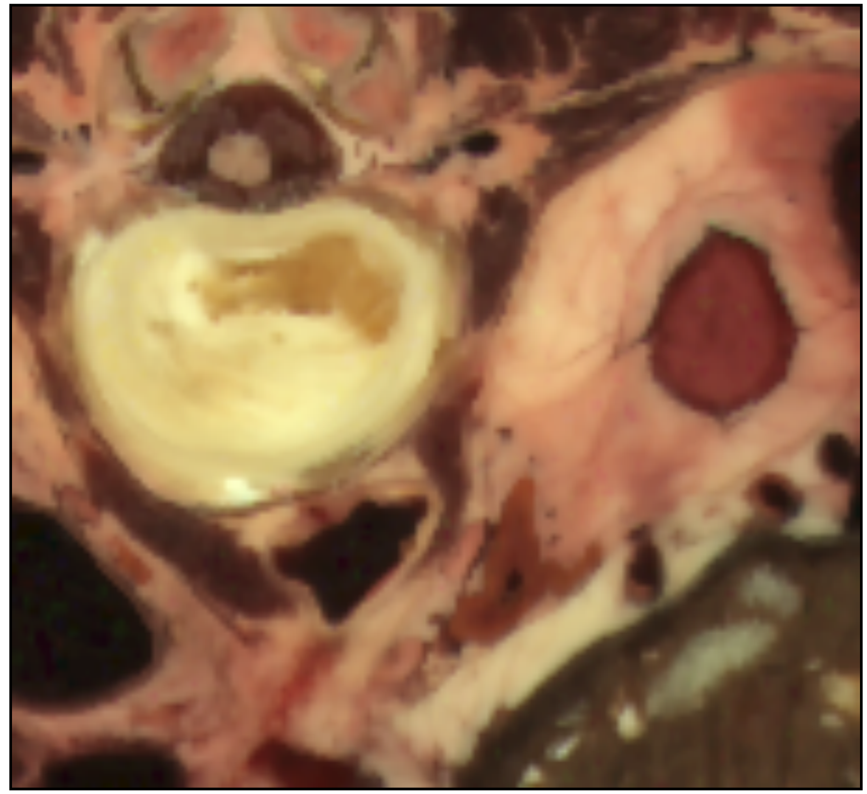
Segmentation
- Watersheds
- Deformable model

Analysis
- Build Atlases
- Analize Variation

Anisotropic Diffusion

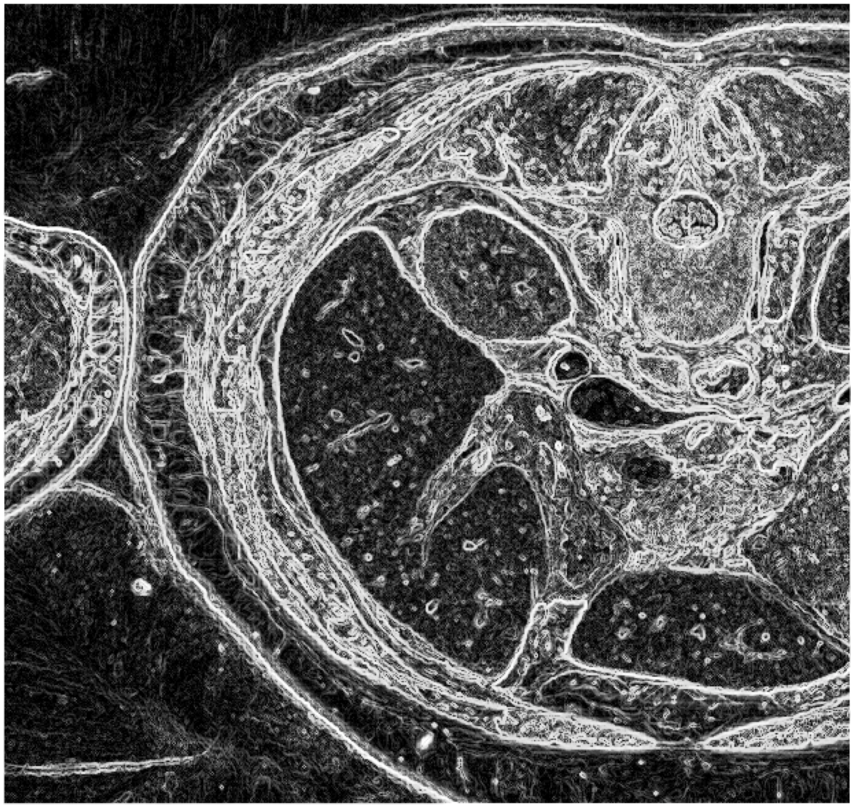


Raw cryosection data

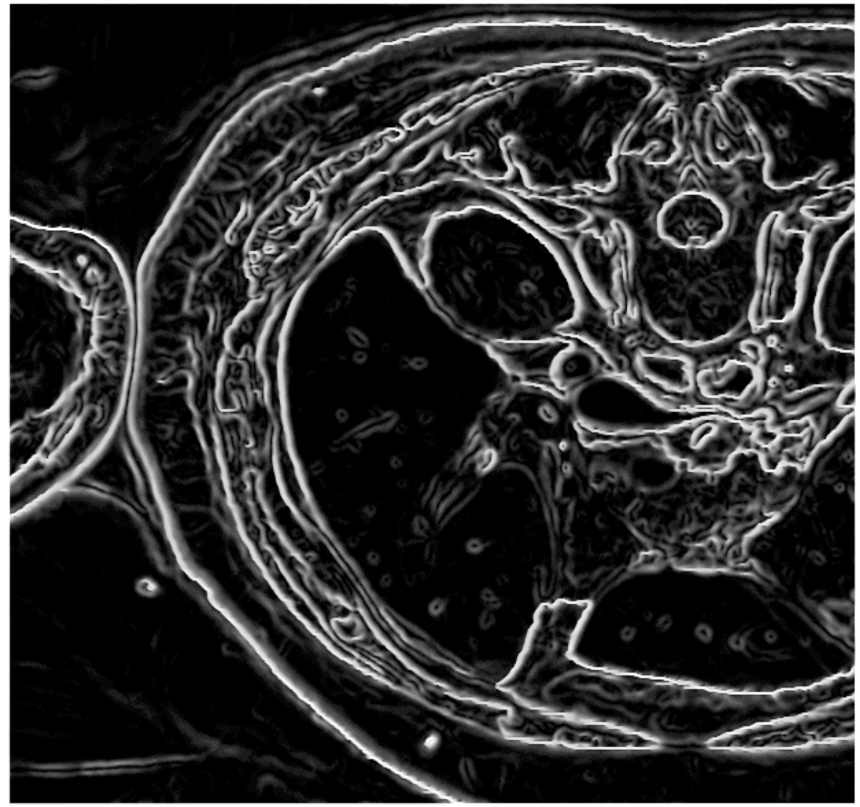


Filtering by anisotropic diffusion

Color Edge Detection Boundary Function



Before Filtering

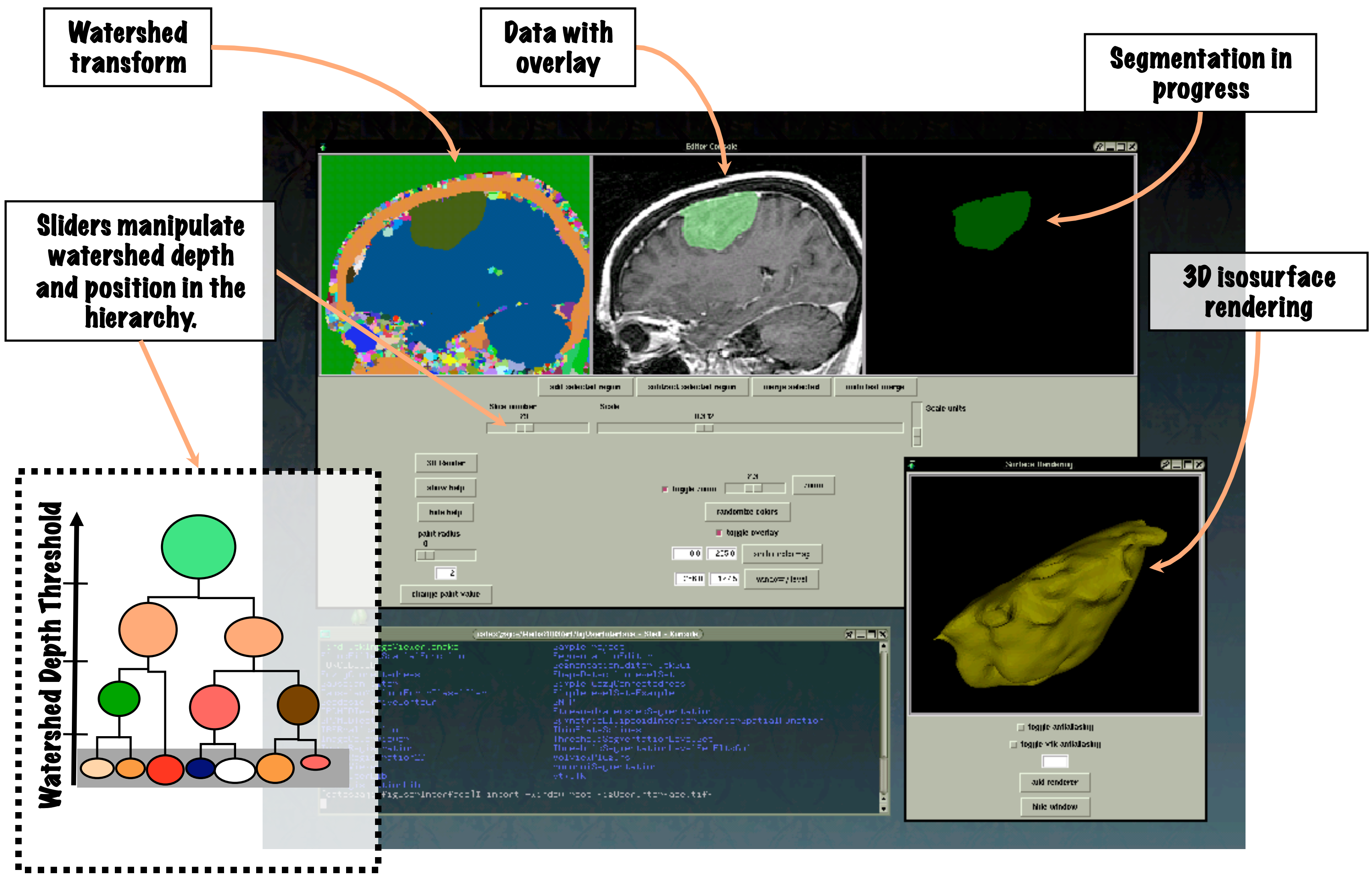


After Filtering

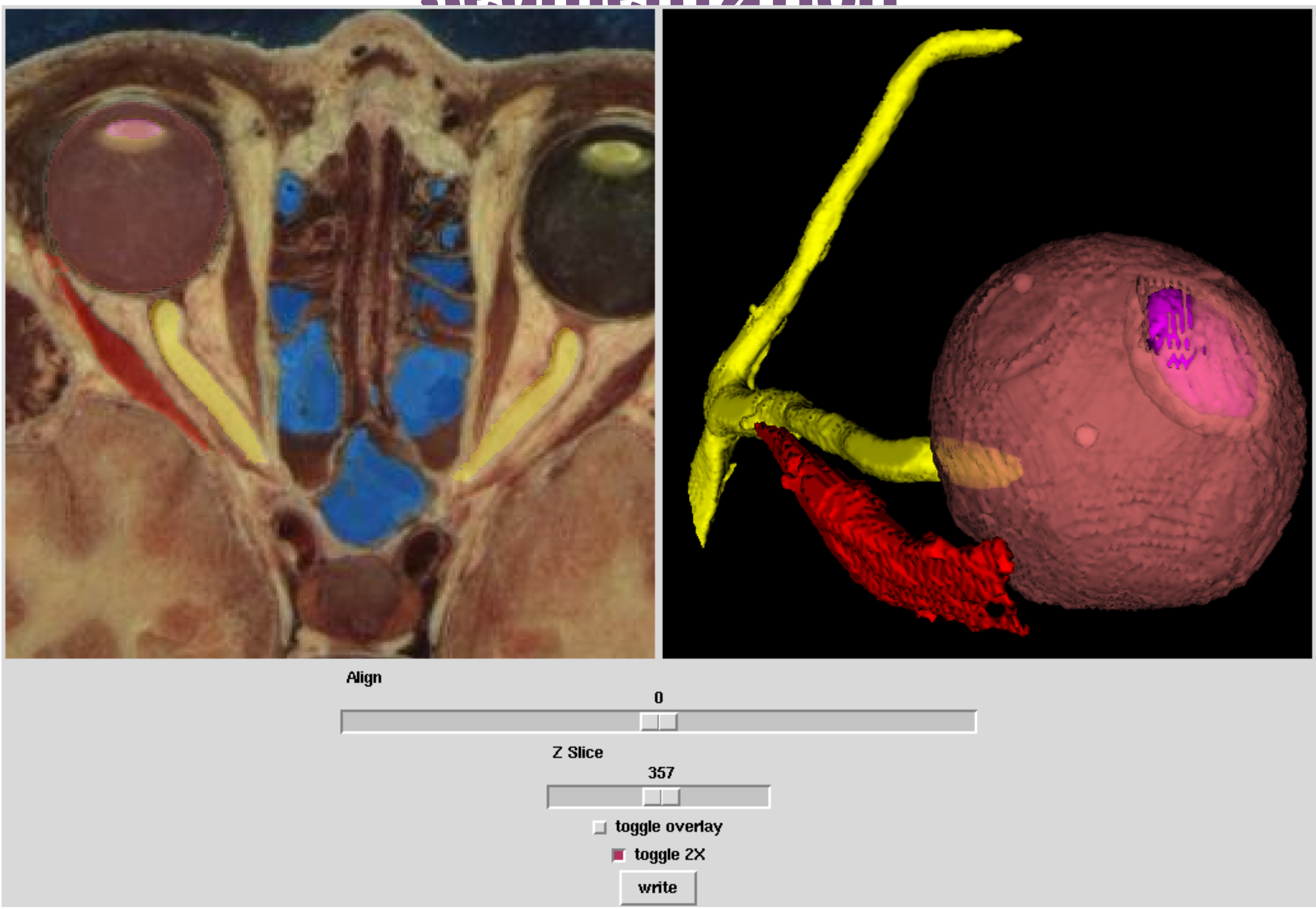
Watershed Segmentation - Level 1

Watershed Segmentation - Level 2

Watershed GUI (Cates `05)



Interactive Watershed Segmentation



Deformable Model

- Object segmentation
- Define a curve that aligns itself with image features to delineate an object
- Issues:
  - What features?
  - How to represent curve?
  - How does it become aligned with data?

Active Contours ("Snakes")
Cass, Witkin, Terzopoulos 87

- Curve $\vec{C}(s) : \Re \mapsto \Re^2$
- Tangent vector $\vec{C}_s / |\vec{C}_s|$
- Define "fitting" energy

$$E[\vec{C}] = \int \left[F(\vec{C}) + \alpha \vec{C}_s^2 + \beta \vec{C}_{ss}^2 \right] ds$$

Attraction to features

Membrane energy (shrink)

Thin-plate energy (stiff)

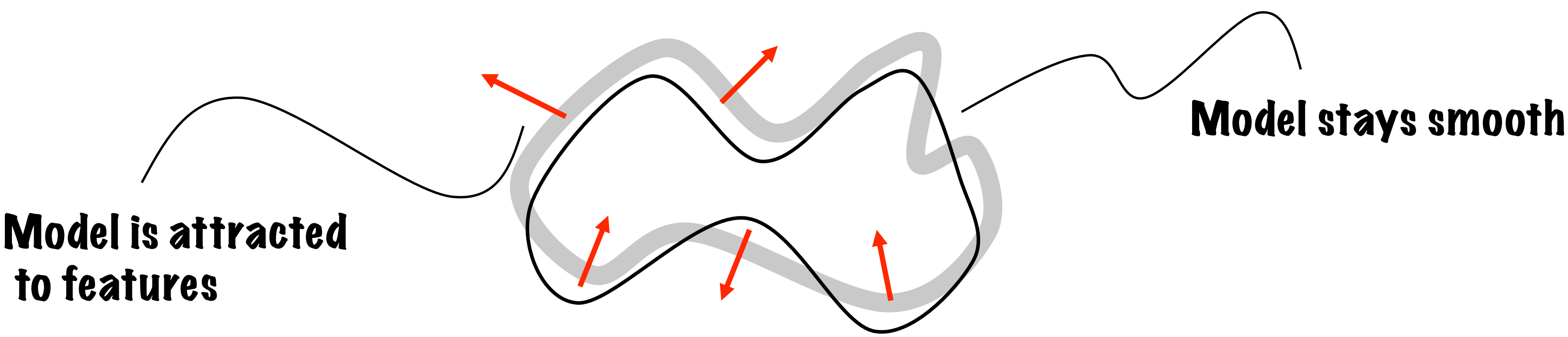
- Minimize/grad. descent -> deformable contour

Snakes: Motion

- First variation gives motion

$$\frac{\partial \vec{C}}{\partial t} = -dE = -\nabla F + \alpha \vec{C}_{ss} + \beta \vec{C}_{ssss}$$

- Snake slides "downhil" on feature image while trying to be "smooth"



Snakes: Computation

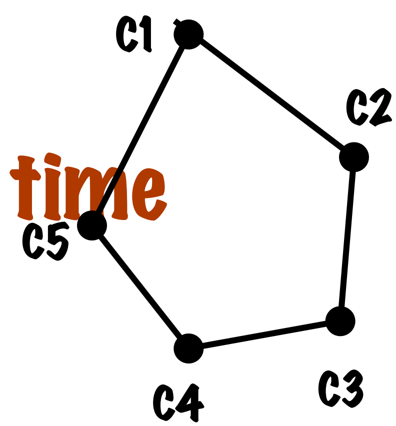
- Represent curve as polyline

$$\vec{C}_i \text{ where i} = 1, \ldots, \text{N}$$

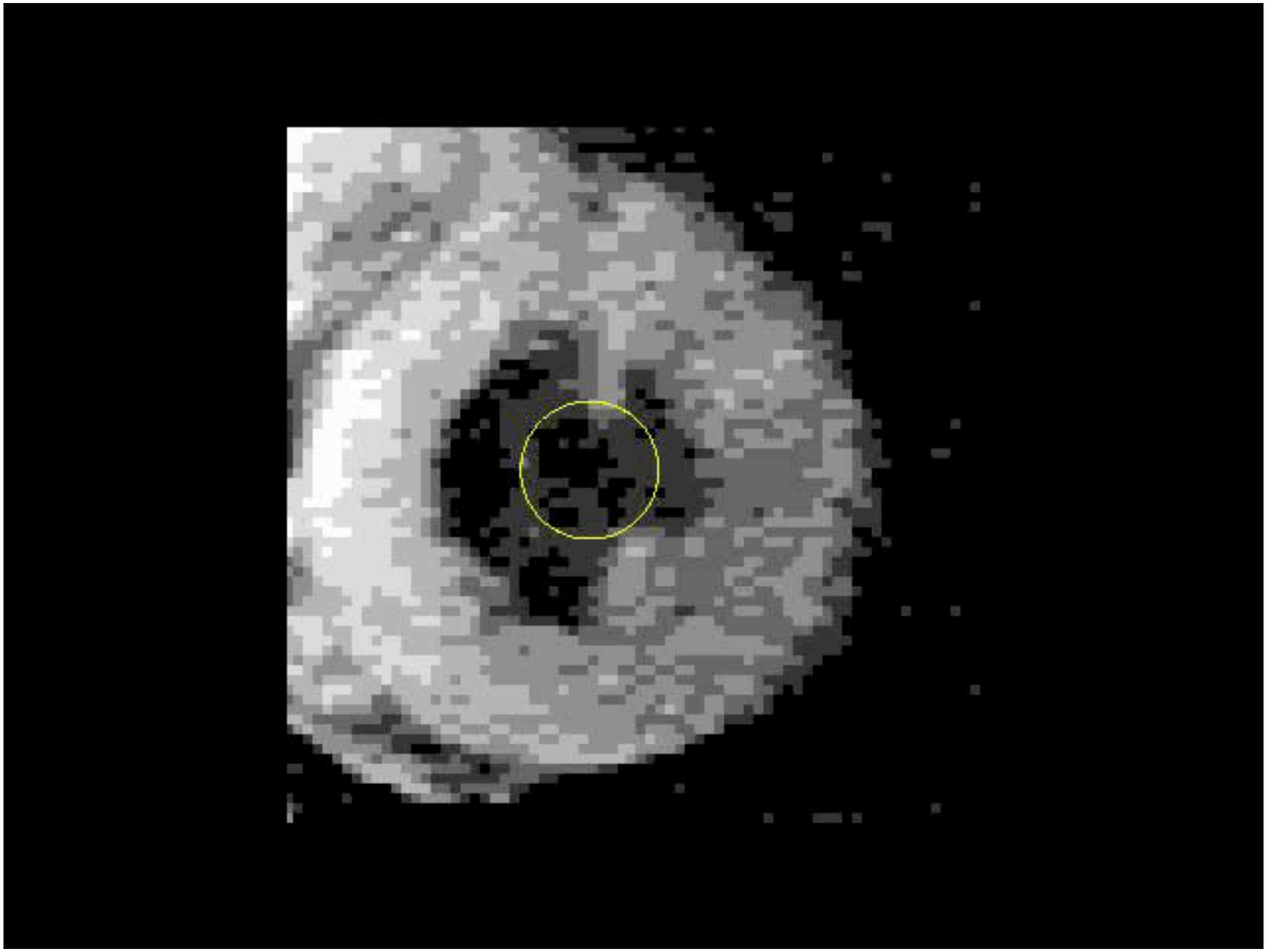
- Approximate derivatives as finite differ

$$\frac{\partial \vec{C}_i}{\partial t} \approx \vec{C}_{i-1} - 2\vec{C}_i + \vec{C}_{i+1}$$

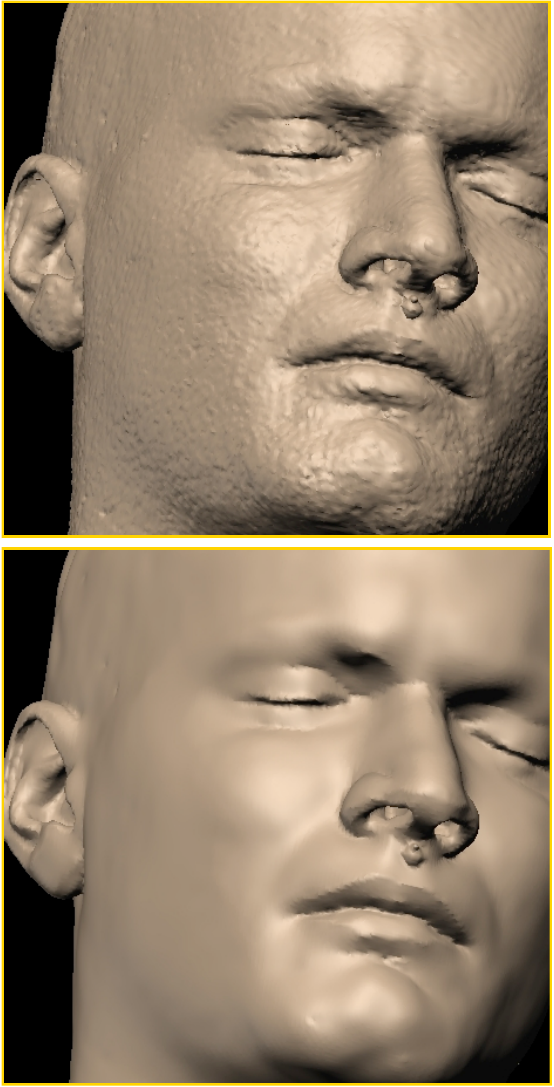
- Update with forward differences in time



Snakes: Example



Deformable Models



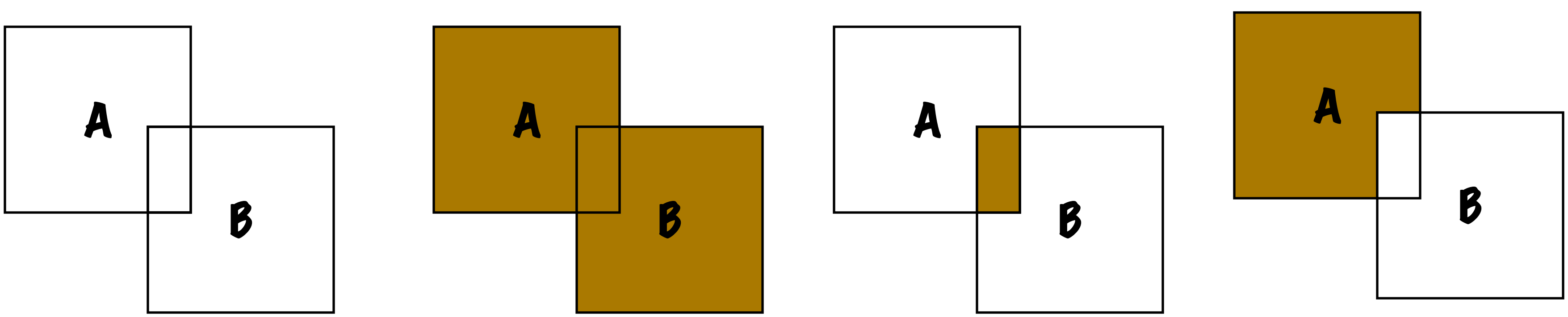
- Spawned many new ideas in segmentation and surface processing
- Extensions that include:
  - Many different kinds of features
  - Combined with statistical classification
  - Spectral/color data
  - 3D surfaces - segmentation and processing
  - Changing topology (split/merge objects)
  - Ties into other PDE-based image processing
  - Other curve/surface

Morphological Image Processing

- Serra - 1980s "Mathematical Morphometry"
- Basic mathematical operations
  - --> theory + algorithms
- Binary images -> greyscale
- Filtering, segmentation, feature detection

Morphometry Concepts

- Sets in images:
  - Regions of value 1 (everything else is 0)
  - Basic operations
    - Intersection, union, subtraction, complement



$A \cup B$ $A \cap B$ $A - B$

Morphometry Concepts

- Translation of set to point z
  - B(z)

$$B_z = \{q | q = w + z \, \forall \, w \in B\}$$

- Reflection of set
  - B'

$$B' = \{z | z = -w \, \forall \, w \in B\}$$

- Dilation of A by B:
  - Makes objects bigger $A \oplus B = \{z | B'_z \cap A \neq \emptyset\}$
- Erosion of A by B:
  - Makes objects smaller $A \ominus B = \{z | B'_z \subset A\}$

Opening and Closing
B – “Structuring Element”

- Opening
  - Generally smoothing by removing material

$$A \circ B = (A \ominus B) \oplus B$$

- Closing
  - Generally smoothing by adding material

$$A \bullet B = (A \oplus B) \ominus B$$

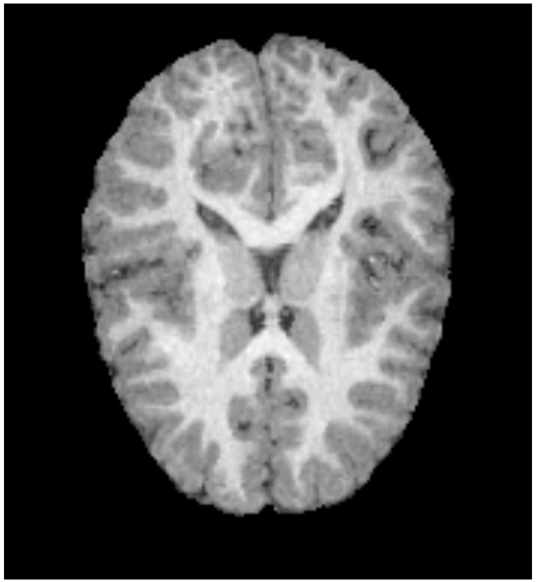
- Idempotent

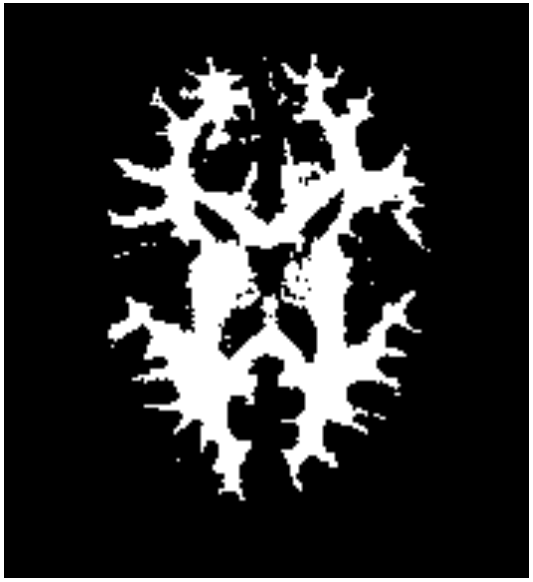
General Concepts

- For smoothing B is normally round
- Discrete implementation is easy
- Dilations of A with B tend to make the result look at little more like B (B convex and normalize for size)
- Openings remove small pieces and connections
- Closings fill in holes and gaps

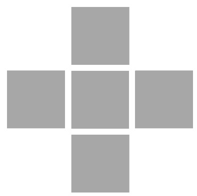
Morphological Filtering

- Thresholding for segmentation
  - "white matter" of the brain from MRI

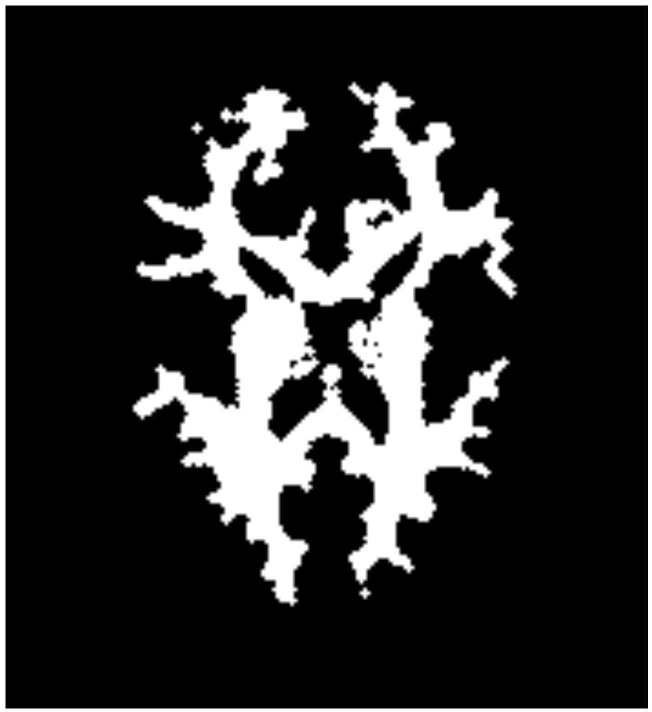




Circular Structuring Element

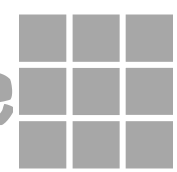


Opening

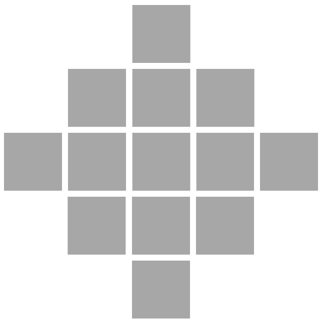


Closing

Note: lots of options for choosing eleme

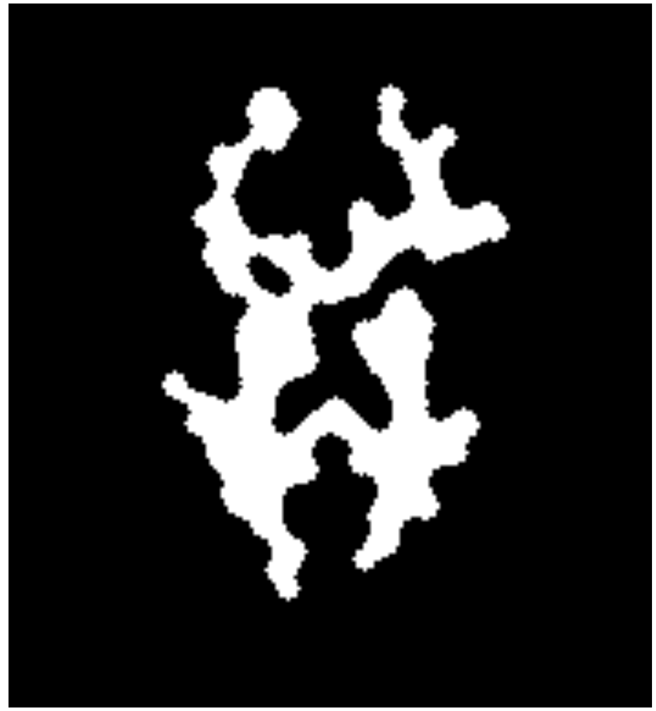


Circular Structuring Element



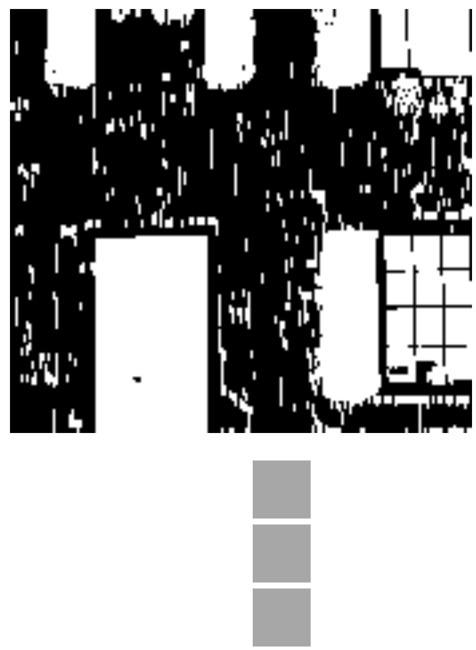
Opening

Closing



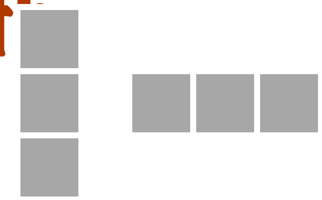
Oriented Structuring Elements

Opening



Oriented Structuring Elements

- Combine with union and close with



Other Useful Algorithms

- Connected components
  - Find all of the pixels (in A) that are connected to a point (in A) via a 4 or 8 connected path in the set A.
- Flood fill
  - Change values of a connected component

Applications

- Work on segmented (e.g. thresholded) regions in images
- Fill in holes
- Remove small, isolated pieces (or connections)
- Smooth boundaries