Image Segmentation

Suyash P. Awate

K-means Clustering

- **Given**: Data points in D-dimensional space
  - Data points can be pixel values
    - Values can be 1-D (intensities) or multi-D
- **Goal**: Partition data points into 'k' clusters
  - 'k' is known / fixed
  - Label each point to belong to one (and exactly one) of the groups
    - Hard (discrete) decision made for each data point

K-means Clustering

- Consider k groups S1,...,Sk
- Consider a **representative** point for each group
  - Called “**mean**”
- **Optimization problem**
  - Find : **Optimal Groups** and their **Means**
  - Penalize sum of squared distances of each data point to the mean of the group it is assigned to
  - Minimize means \mu_i and labels S_i

$$\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

K-means Clustering

$$\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

- Optimization algorithm (2 steps)

  (0) An initial solution for the set of means is given

  - Can equivalently assume an initial solution for the groups

  (1) Given means, what are the optimal groups ?

  - Which group will you assign to each data point so that the objective function is guaranteed to decrease ?
  - Assign the group whose mean is the closest to data point

  (2) Given groups, what are the optimal means ?

  - For a specific group, what mean will minimize the penalty for that group ?
  - Assign mean = average of the data points in that group

K-means Clustering

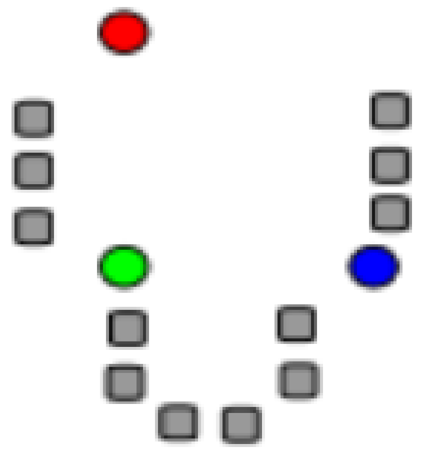
- Optimization algorithm
  - Is it guaranteed to converge / terminate. Why or why not ?
  - Will it converge to a global optimum ? Why or why not ?
  - Is the objective function quadratic ? convex ?

K-means Clustering

- Optimization algorithm
  - Is it guaranteed to converge / terminate. Why or why not ?
    - Yes. Sample size is finite. So # labelings is finite. Each relabeling reduces a non-negative penalty function; else you've stopped.
  - Will it converge to a global optimum ? Why or why not ?
    - No guarantees. See a counter example soon
  - Is the objective function quadratic ? convex ?
    - No. Combinatorial optimization. NP hard.
    - Given groups, objective function for means is quadratic

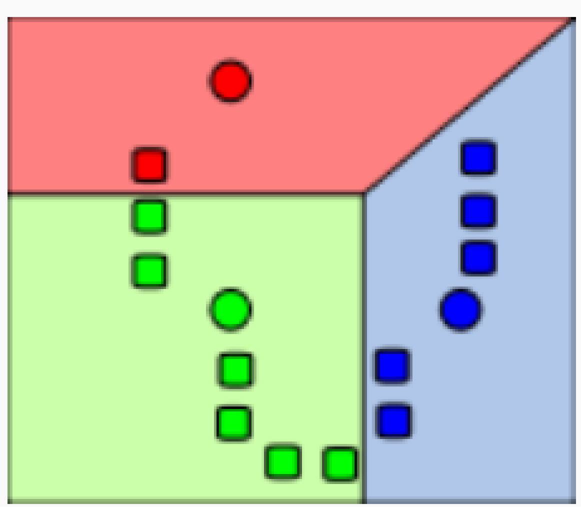
K-means Clustering

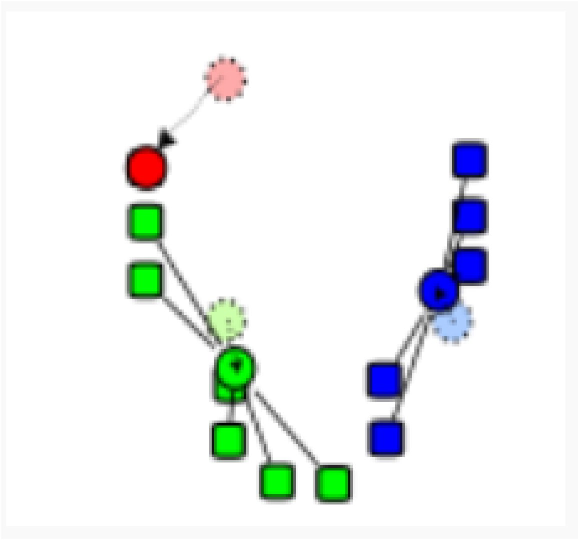
- K-means example
  - Data points in black
  - K = 3 (fixed)
  - Initial means in R, G, B
    - 1st K points in the list ?
    - Randomly selected K points in the list ?
    - Randomly generated K points in the space ?
    - Can we do better that any of these ? Yes. Will see soon.



K-means Clustering

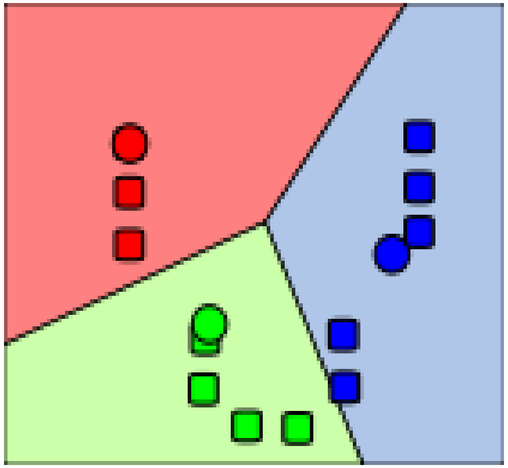
- K means example
  - Given means, update groups
  - Given groups, update means





K-means Clustering

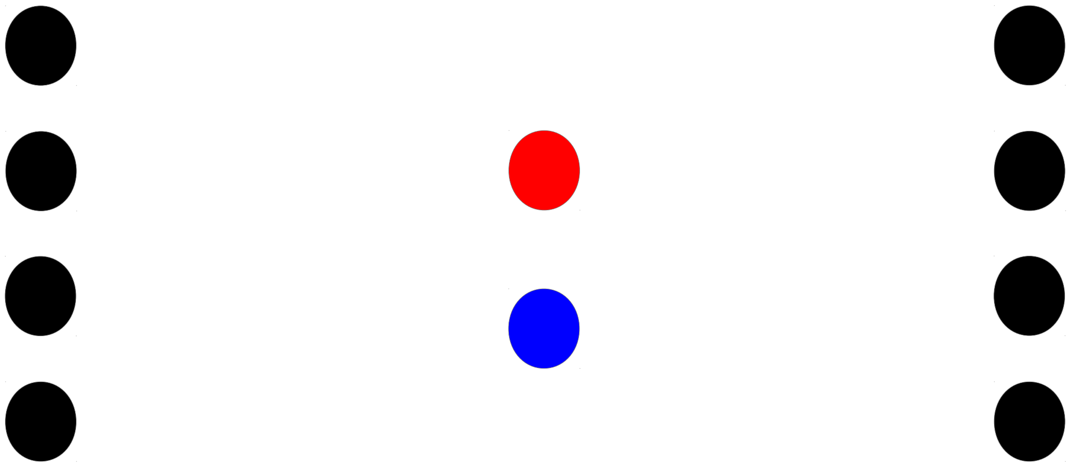
- K means example
  - Given means, update groups



  - Repeat until objective function stops reducing

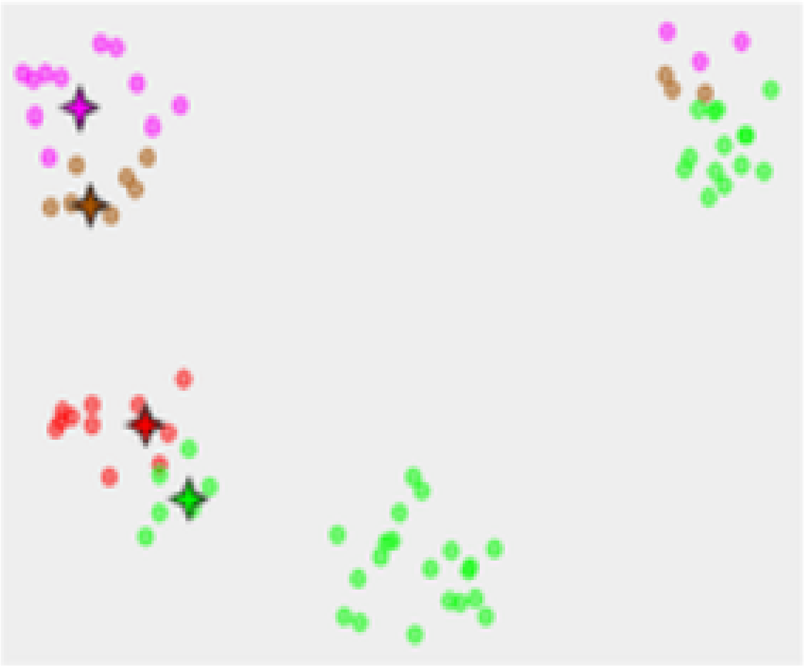
K-means Clustering

- Convergence to local minimum
  - Example 1



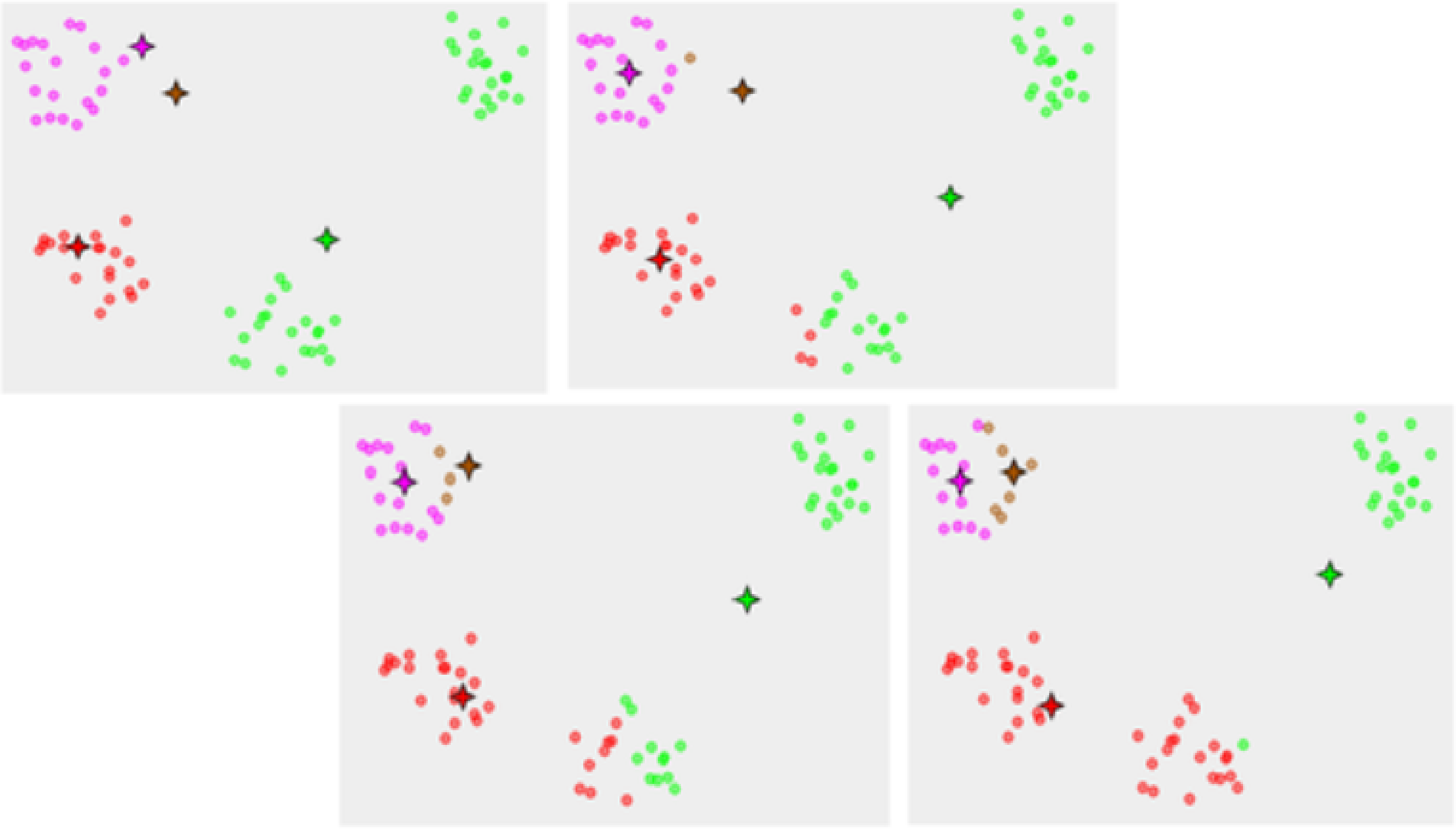
K-means Clustering

- Convergence to local minimum
  - Example 2
    - k = 4
    - Initial means



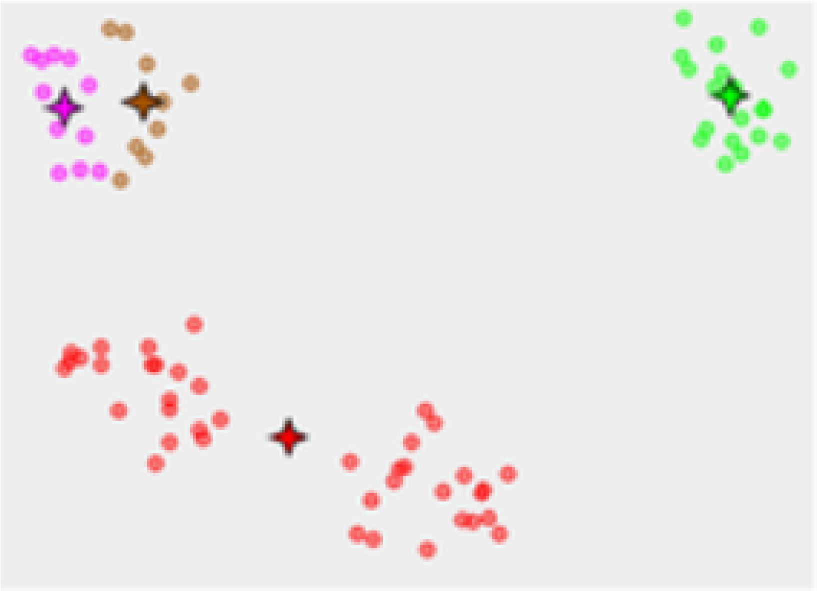
K-means Clustering

- Convergence to local minimum



K-means Clustering

- Convergence to local minimum
  - Example 2
    - Final solution
      - Local minimum



K-means Clustering

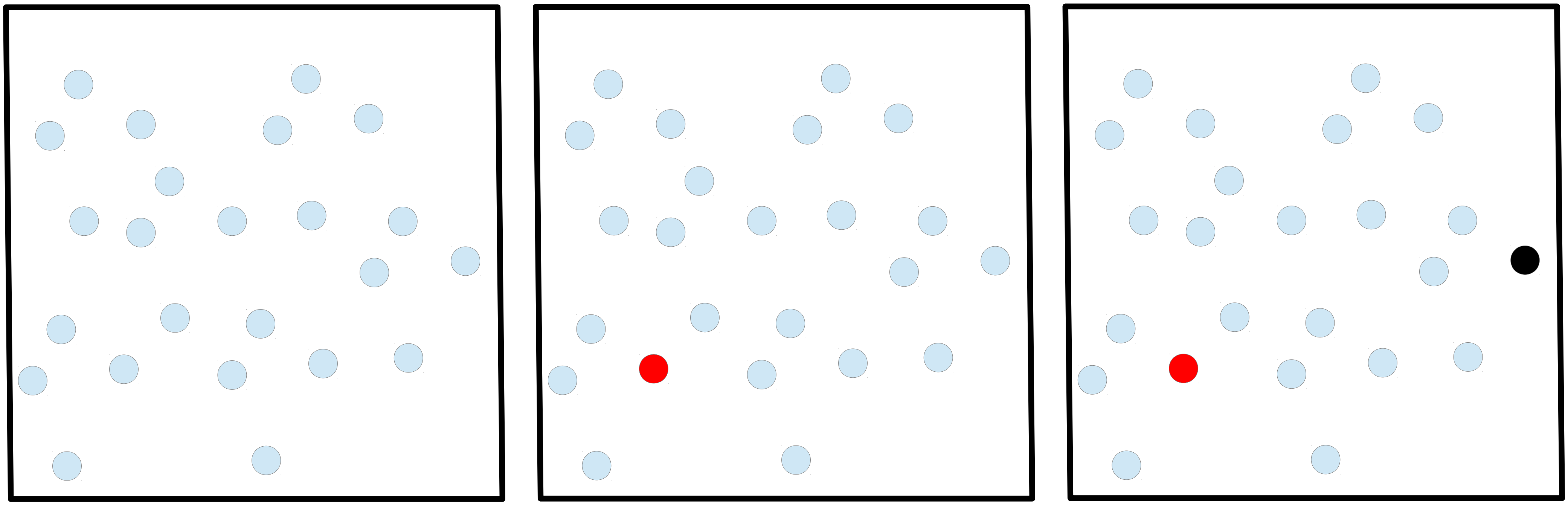
- Hope to get global optimum ?
  - Brute-force approach
    - Check objective-function value for all possible clusterings
    - N points and k clusters → k^N possible clusterings
  - Re-run algorithm using different initial mean estimates
    - Introduce randomization in the initialization
  - Better initialization

K-means Clustering

- Better **initialization**
  - Farthest-point clustering
  - Algorithm
    - Heuristic
    - **Given**: data points, value of 'k'
    - Select 1st “mean” at random from among the data points
    - Select the other means, among the data points, iteratively
    - Assume: at some iteration, (n-1) means have been selected
    - Select the **n-th mean** to be the **data point** that
      maximizes the
      minimum distance among distances to the current (n-1) means

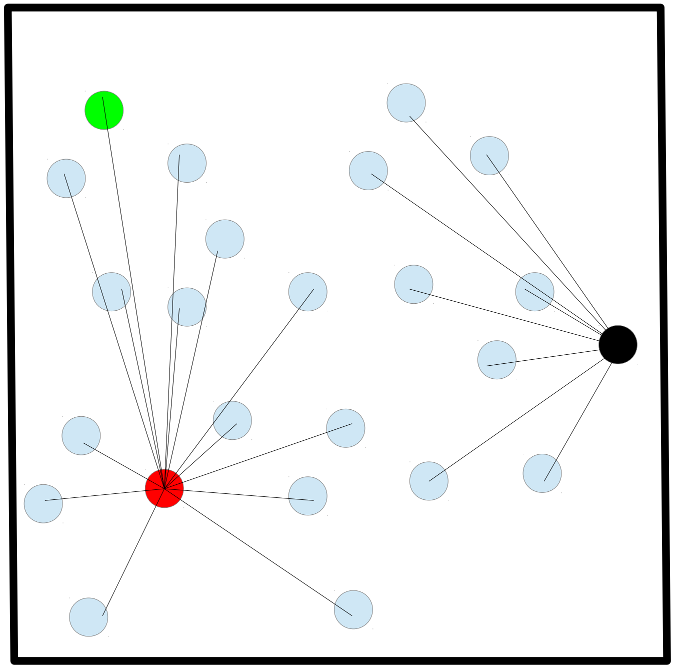
K-means Clustering

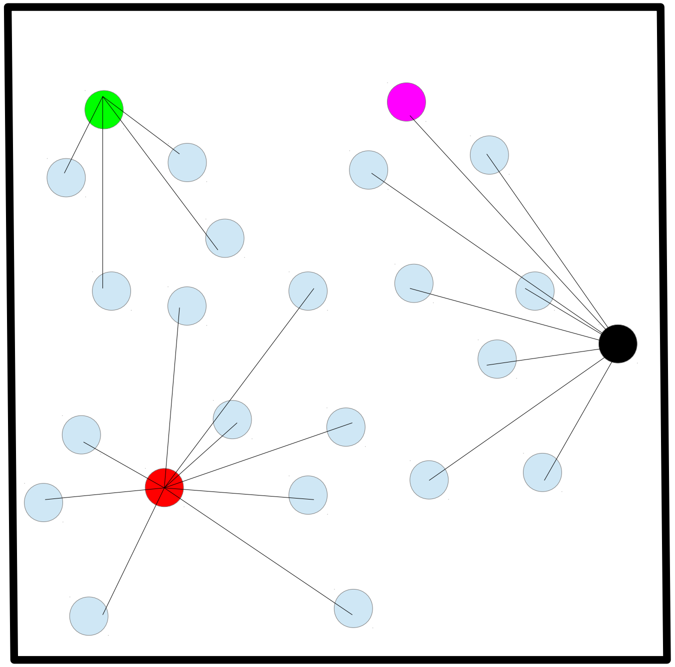
- Better initialization
  - Farthest-point clustering



K-means Clustering

- Better initialization
  - Farthest-point clustering
    - Select the point that is farthest from the current means !
    - Spreads out the selected points (roughly) uniformly





K-means Clustering

- Better initialization
  - Farthest-point clustering
    - Sensitive to outliers
  - K-means++
    - Don't pick farthest point
    - Pick point 'x' with probability directly related to minimum distance

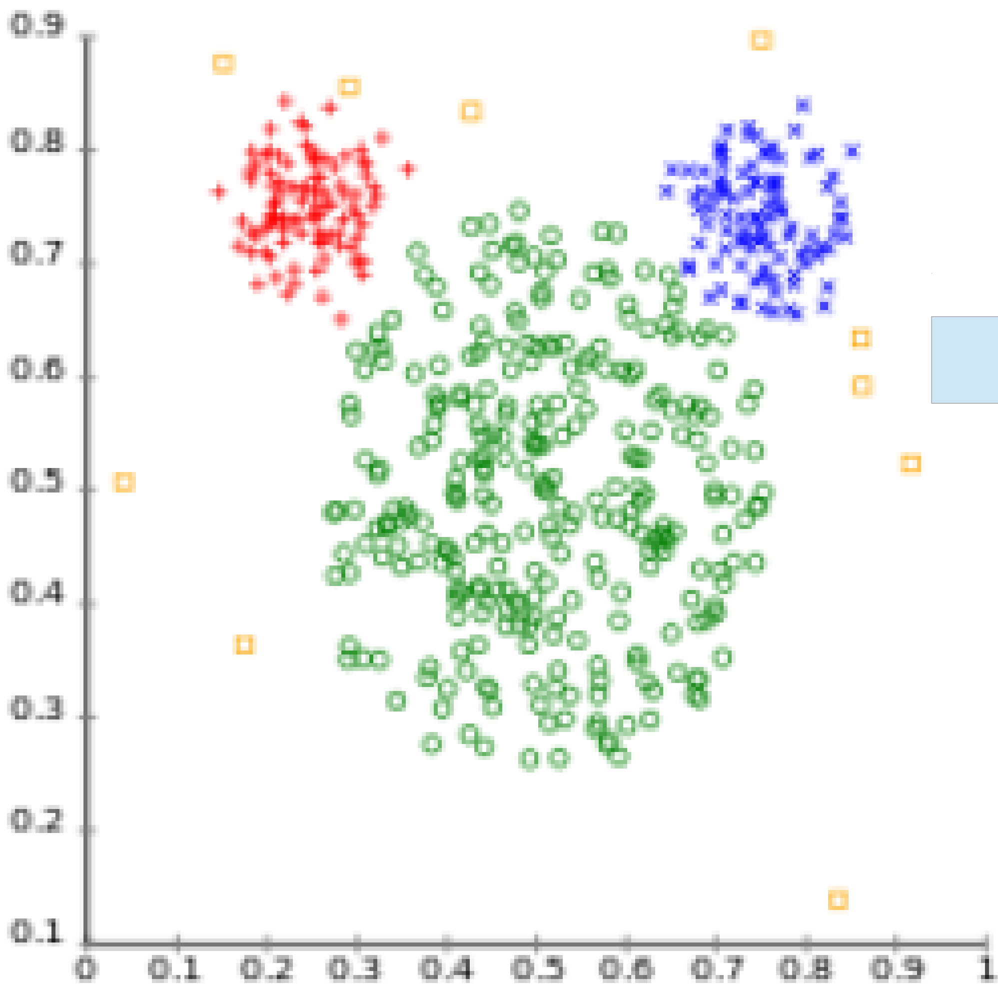
```
pick x ∈ S uniformly at random and set T ← {x}
while |T| < k:
    pick x ∈ S at random, with probability proportional to cost(x, T) = min_{z∈T} ||x − z||^2
    T ← T ∪ {x}
```

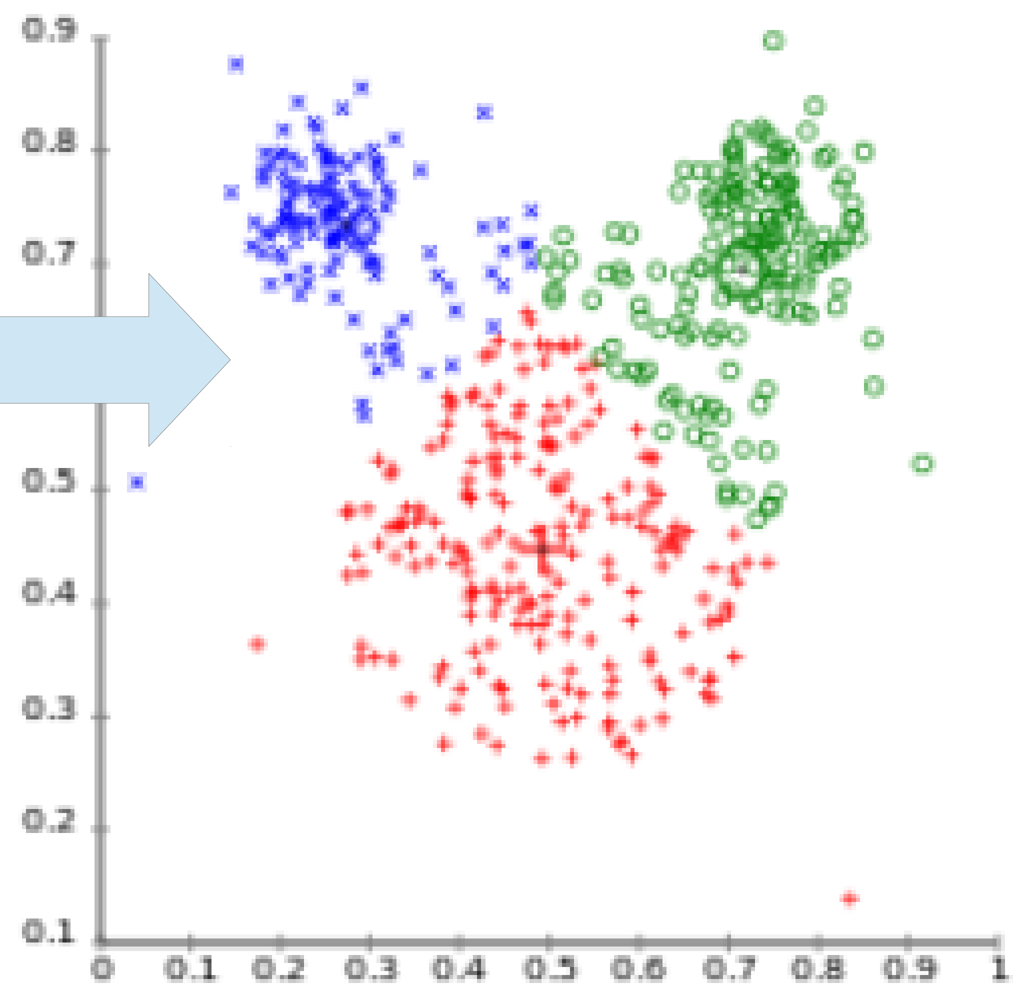
$$\text{pick } x \in S \text{ uniformly at random and set } T \leftarrow \{x\}$$
$$\text{while } |T| < k:$$
$$\quad \text{pick } x \in S \text{ at random, with probability proportional to } \mathrm{cost}(x,T) = \min_{z \in T} \|x - z\|^2$$
$$\quad T \leftarrow T \cup \{x\}$$

- Arthur, D. and Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In ACM-SIAM Symposium on Discrete Algorithms
- https://cseweb.ucsd.edu/~dasgupta/291-geom/kmeans.pdf

K-means Clustering

- Limitations
  - Tendency to produce clusters with equal spreads





K-means Clustering

- Limitations

  1) Binary / hard assignments of each data point to cluster

  - What about points near interface of 2 clusters ?

  2) Ignore covariance of the cluster

  - Assumes spherical clusters
    - Because relies on simple Euclidean distance
  - Doesn't handle ellipsoidal clusters systematically

Evaluating Quality of Clustering

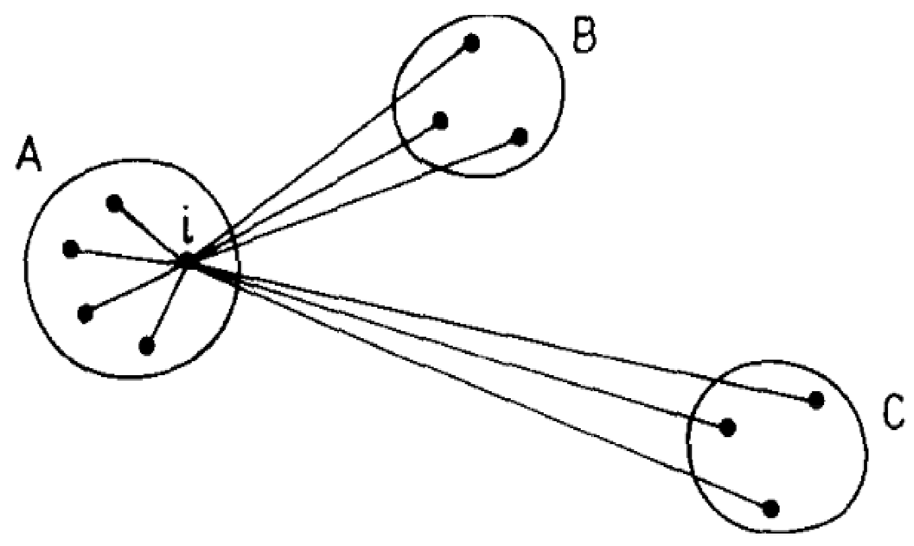
- "Silhouette" analysis
  http://dx.doi.org/10.1016%2F0377-0427%2887%2990125-7
  - Graphical representation
  - Measures how contained is each datum within its cluster

Evaluating Quality of Clustering

- For each datum x_i **assigned to cluster A**
  - **a_i** = avg of distances between x_i & **x_j within cluster A**
    - $a_i > 0$
    - Small $a_i \rightarrow$ more compact cluster
  - **b_i** = ?

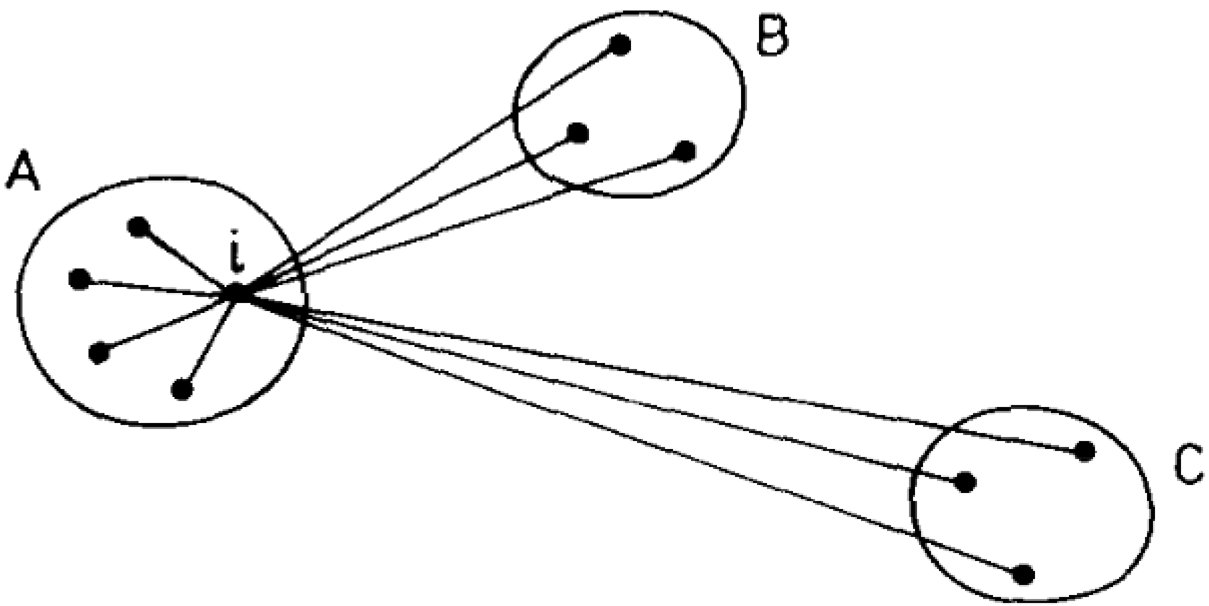
    1) avg of distances between x_i & **x_j within another cluster B**

    2) select minimum of these over all other clusters B
    - $b_i > 0$
    - Large $b_i \rightarrow$ more compact cluster



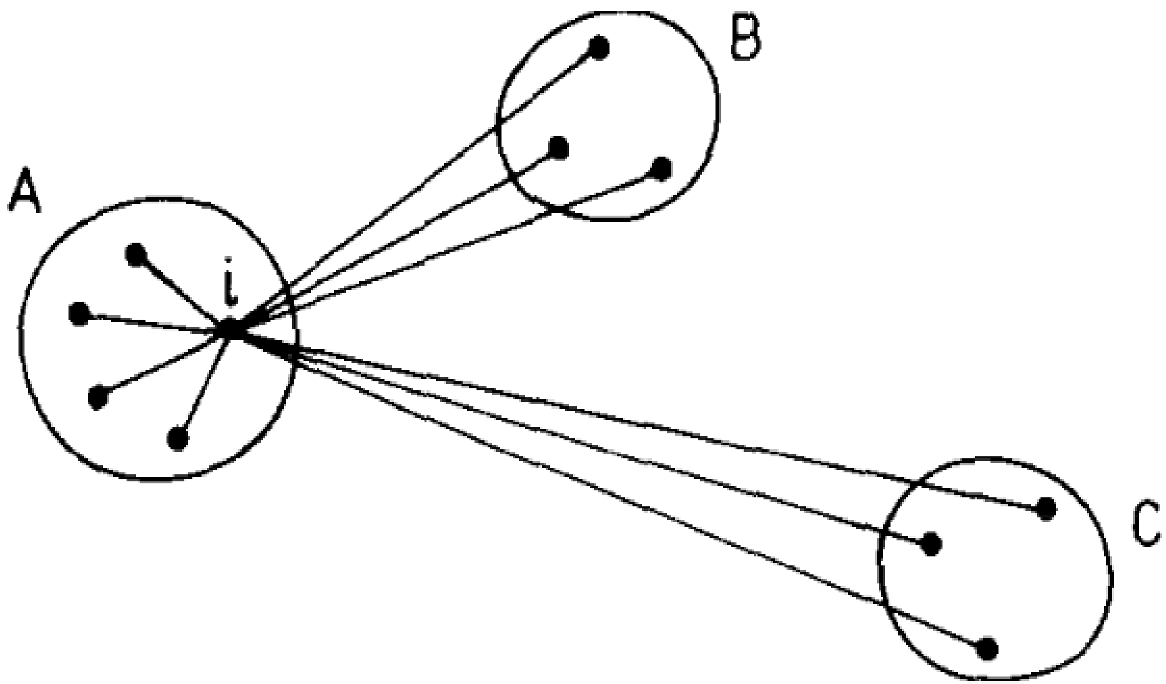
Evaluating Quality of Clustering

- **For each datum, define $s_i = (b_i - a_i) / \max(a_i, b_i)$**
- When $b_i = a_i$
  - $s_i = 0$
- When $b_i > a_i$
  - $s_i = 1 – a_i / b_i$
  - $0 < s_i <= 1$
- When $a_i > b_i$
  - $s_i = b_i / a_i – 1$
  - $-1 <= s_i < 0$



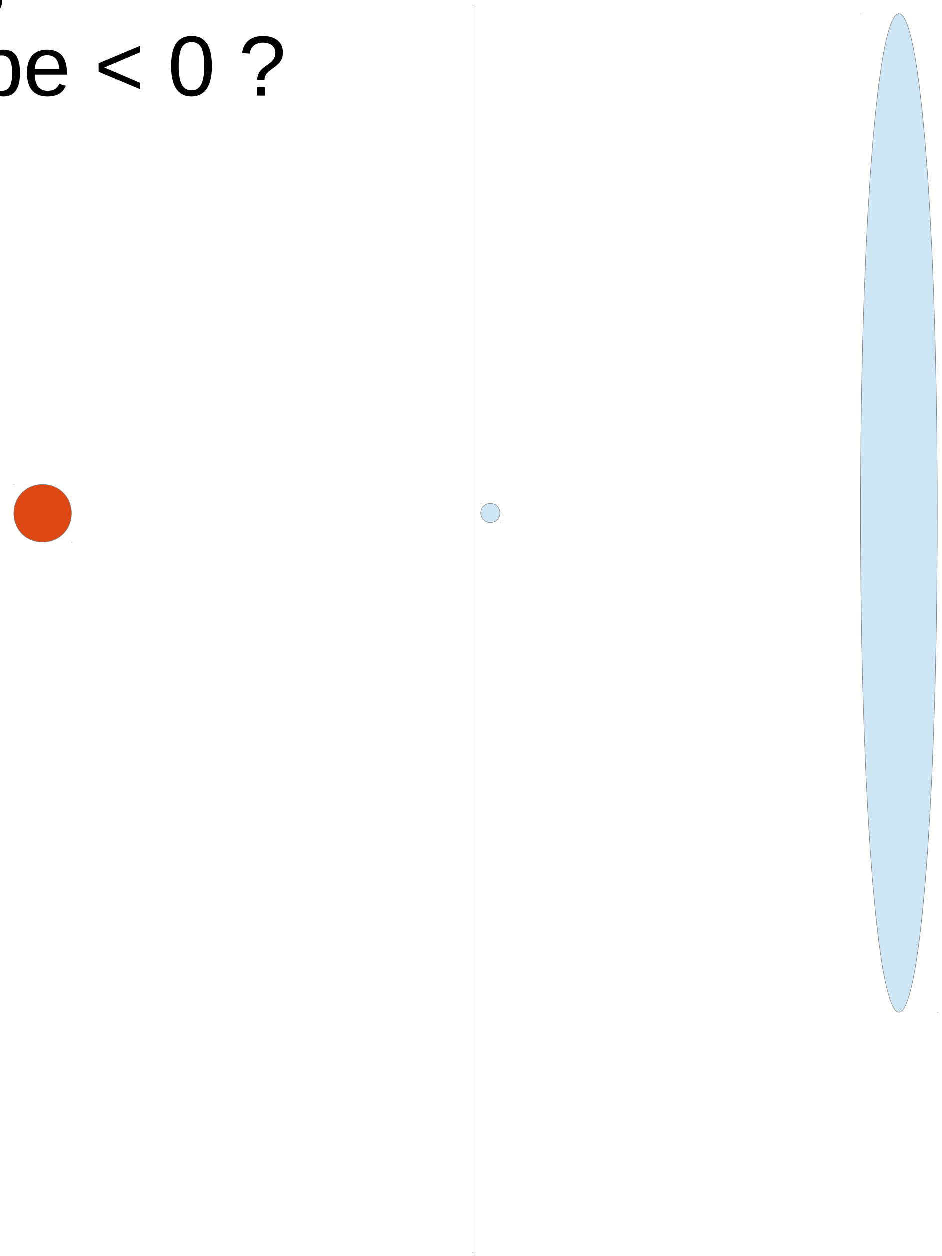
Evaluating Quality of Clustering

- We desire s_i close to +1
  - i.e., $a_i << b_i$
- What does it mean to have s_i close to 0 ?
  - Datum x_i lies ...
- What does it mean to have s_i close to -1 ?
  - Datum x_i lies …



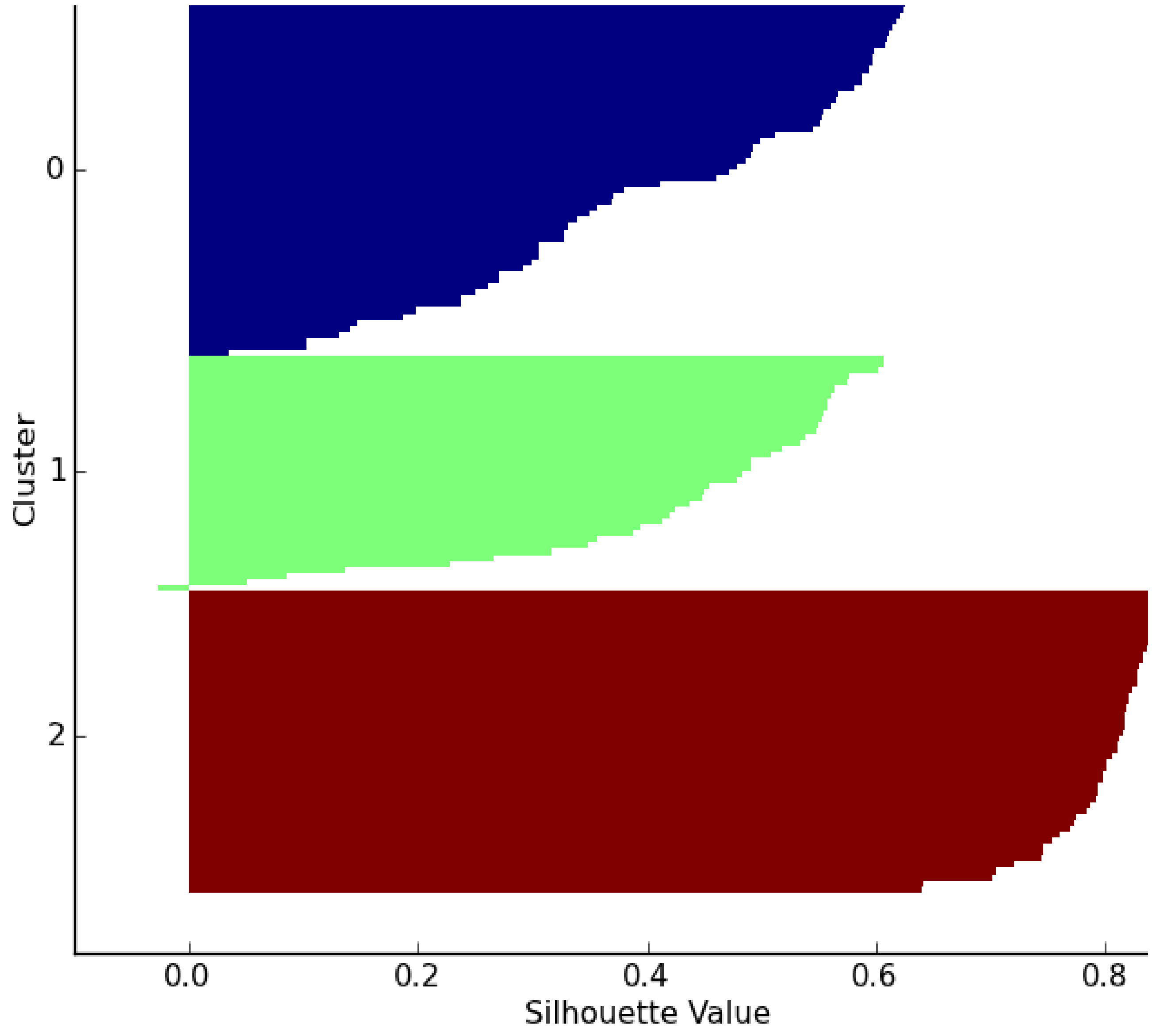
Evaluating Quality of Clustering

- For k-means, when can s_i be < 0 ?



Evaluating Quality of Clustering

- For each cluster, visualize graph of s_i over i
  - This is the “silhouette” / outline
- May be used to tune # of clusters K or other underlying parameters



Fuzzy-C-Means (FCM) Clustering

- Generalization of K-means clustering
  - Assigns, to each data point,
    a membership to belong to each cluster
    - Membership in [0,1]
    - For each data point,
      sum of memberships over all clusters = 1
  - Produces a “soft” / “fuzzy” segmentation
    - K-means produces a “hard” / “crisp” segmentation

Fuzzy-C-Means Clustering

- Given
  - Data = { y_j }, $j = 1,\ldots,N$
  - Number of clusters = K (known / fixed)
- Memberships
  - u_{jk} = membership of j-th point in k-th cluster

(i) memberships are positive: $\forall j, \forall k, u_{jk} \geq 0$ and
(ii) for every data point, memberships sum to one: $\forall j, \sum_k u_{jk} = 1$

- "q" = user-defined parameter
  - Controls fuzziness of the clusters / memberships
  - **q > 1**

Fuzzy-C-Means Clustering

- Objective function to be minimized

$$\sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q}(y_j - c_k)^2$$

  – Penalize squared distance of point j from mean of class k
  – Weight penalty based on membership u_{jk}

- Constraints

$$\forall j, \sum_k u_{jk} = 1$$

  – Positivity constraint on memberships:
    - Isn't enforced explicitly
    - Gets satisfied automatically

Fuzzy-C-Means Clustering

- FCM reduces to K-means
  - What happens if you force u_{jk} to be binary ?
    - What happens to the objective function ?

Fuzzy-C-Means Clustering

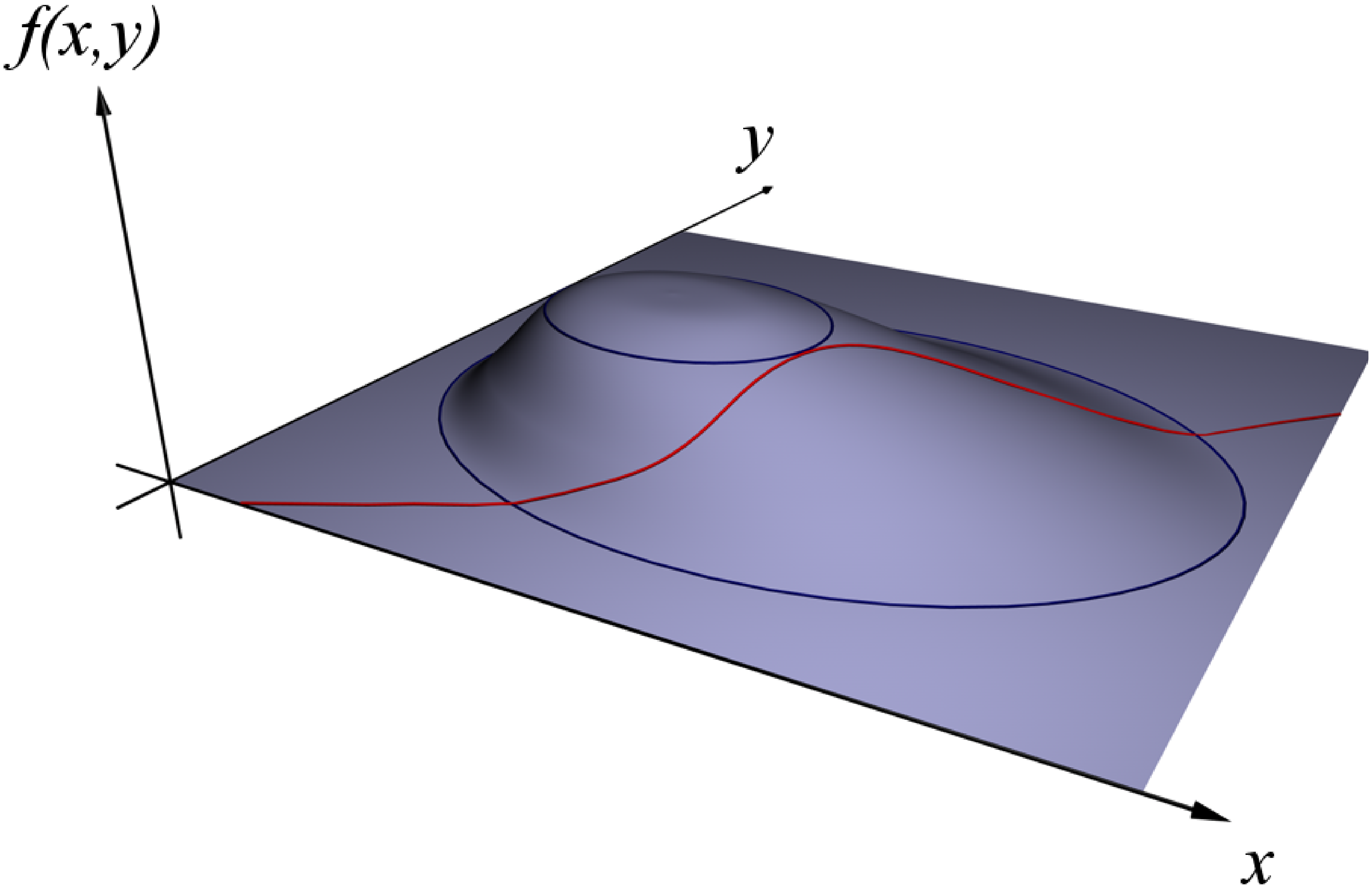
- FCM reduces to K-means
  - What happens if you force u_{jk} to be binary ?
    - u_{jk} = 1 for exactly one cluster k
    - $u_{jk'}$ = 0 for all other clusters k'
    - What happens to the objective function ?
      - Objective function becomes exactly the same as that for K means

Fuzzy-C-Means Clustering

- **Constrained Optimization**
  - Objective function type = Non-convex
  - Constraint type = Equality
  - Approach : Method of Lagrange multipliers

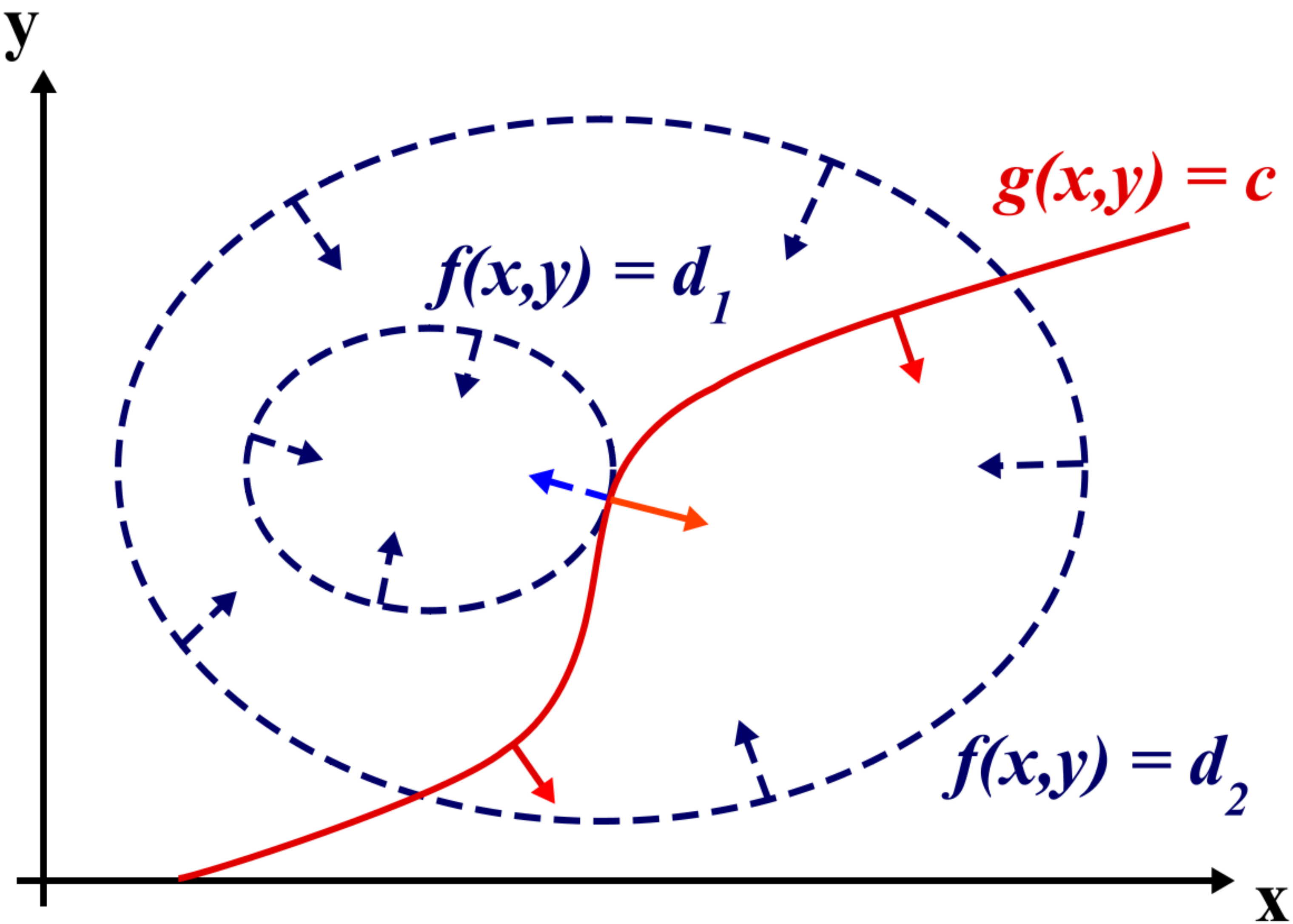
Fuzzy-C-Means Clustering

- **Method of Lagrange Multipliers**
  - Optimization problem
    - Maximize : f (x,y)
    - Subject to (equality / boundary) constraint : g (x,y) = c



Fuzzy-C-Means Clustering

- Method of Lagrange Multipliers
  - What should happen at the optimal solution (x*,y*) ?



Fuzzy-C-Means Clustering

- Method of Lagrange Multipliers
  - What should happen at the optimal solution (x*,y*) ?

    (1) g (x*,y*) = c

    (2) At (x*,y*), gradients of objective function and constraint **<u>must</u>** be parallel to each other

$$\nabla_{x,y} f = -\lambda \nabla_{x,y} g,$$

for some λ

where

$$\nabla_{x,y} f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right), \qquad \nabla_{x,y} g = \left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y} \right)$$

Fuzzy-C-Means Clustering

- Method of Lagrange Multipliers
  - Specify these 2 conditions in the following way
    - Introduce a function called the Lagrangian
      - $L(x, y, \lambda) = f(x,y) + \boldsymbol{\lambda} (g(x,y) - c)$
      - $\boldsymbol{\lambda}$ is called the Lagrange parameter
    - Solve for x, y, λ
      - Assign partial derivatives of Lagrangian, w.r.t. x, y, λ, to zero
      - 3 equations, 3 unknowns
    - Partial derivative w.r.t. 'x' and 'y' gives
      - $d f(x,y) / dx = - \lambda \ d g(x,y) / dx$
      - $d f(x,y) / dy = - \lambda \ d g(x,y) / dy$
    - Partial derivative w.r.t. 'λ' gives us back the constraint
      - $g(x,y) = c$

Joseph-Louis Lagrange

- Advisors: Euler + …
- Students: Fourier, Poisson, …
- It was not until he was 17 that he showed any taste for math
- For 20 years in Berlin, except for a short time when he was ill, he produced ~ 1 paper / month
- Lagrange is one of the 72 prominent French scientists who were commemorated on plaques at the 1st stage of Eiffel Tower

Joseph-Louis Lagrange

Lagrangian [edit]

- Lagrangian
- Lagrangian analysis
- Lagrangian coordinates
- Lagrangian derivative
- Lagrangian drifter
- Lagrangian foliation
- Lagrangian Grassmannian
- Lagrangian intersection Floer hc
- Lagrangian mechanics
- Lagrangian mixing
- Lagrangian point
- Lagrangian relaxation
- Lagrangian submanifold
- Lagrangian subspace
- Nonlocal Lagrangian
- Proca lagrangian
- Special Lagrangian submanifold

Lagrange [edit]

- Euler–Lagrange equation
- Green–Lagrange strain
- Lagrange bracket
- Lagrange–d'Alembert principle
- Lagrange error bound
- Lagrange form
- Lagrange interpolation
- Lagrange invariant
- Lagrange inversion theorem
- Lagrange multiplier
- Lagrange number
- Lagrange point colonization
- Lagrange polynomial
- Lagrange property
- Lagrange reversion theorem
- Lagrange resolvent
- Lagrange stream function

Lagrange's [edit]

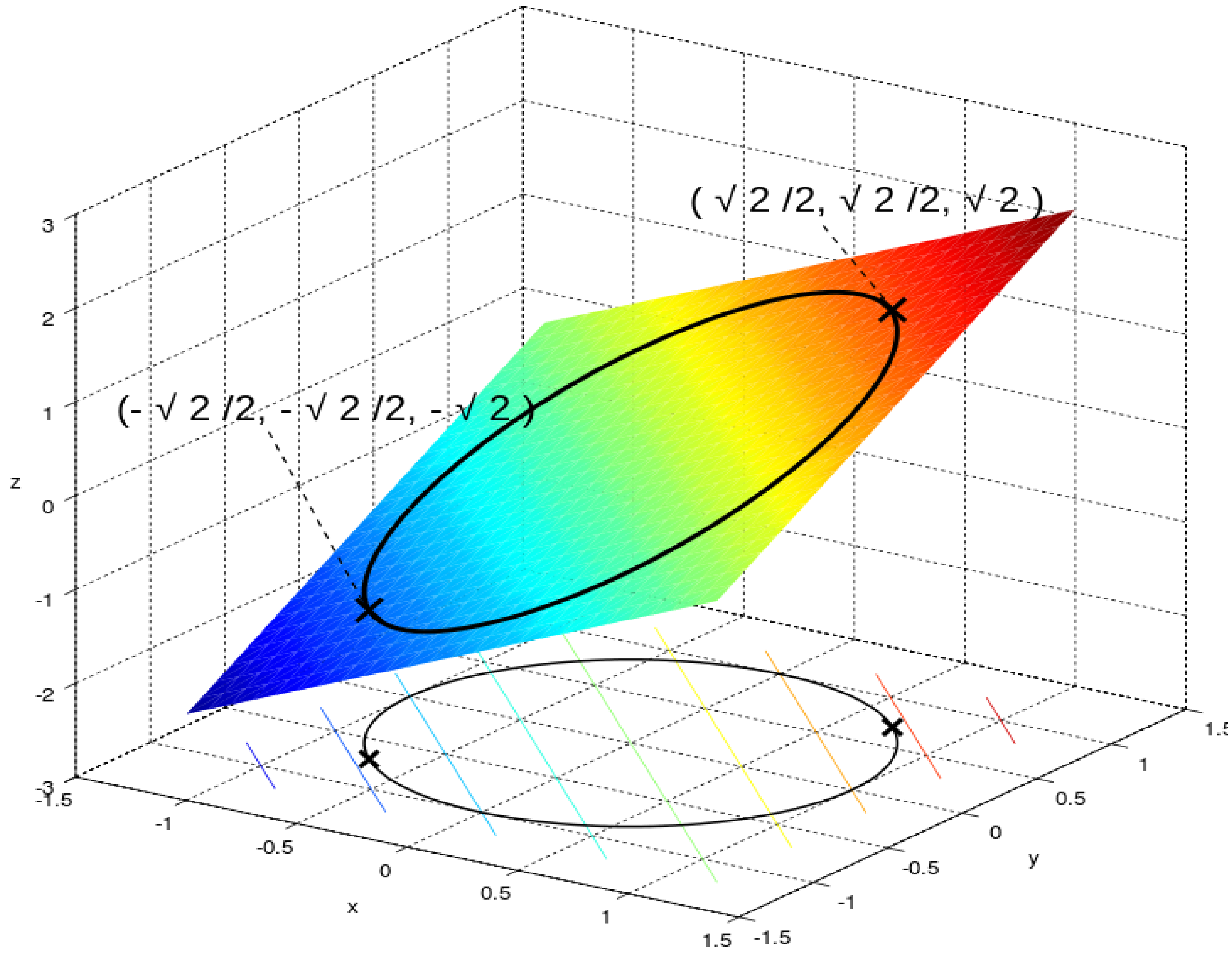
- Lagrange's approximation theorem
- Lagrange's formula (disambiguation)
- Lagrange's identity (disambiguation)
- Lagrange's theorem (group theory)
- Lagrange's theorem (number theory)
- Lagrange's four-square theorem
- Lagrange's trigonometric identities

Fuzzy-C-Means Clustering

- Method of Lagrange Multipliers
  - Lagrangian optimization gives a necessary condition
    - Not a sufficient condition
    - Example: maximize $f(x,y) = x + y$ subject to $x^2 + y^2 = 1$
      - How does the constraint look like (in 2D XY plane) ?
      - How do the objective-function contours look like (in 2D XY plane) ?

Fuzzy-C-Means Clustering

- Method of Lagrange Multipliers
  - Lagrangian optimization gives a necessary condition
    - Not a sufficient condition
    - Example: maximize $f(x,y) = x + y$ subject to $x^2 + y^2 = 1$



Fuzzy-C-Means Clustering

- FCM Optimization via Lagrange Multipliers
  - Start with an initial solution for memberships and cluster means
  - Use method of Lagrange multipliers to iteratively improve solution
  - Lagrangian

$$L(\{u_{jk}\},\{c_k\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q (y_j - c_k)^2 + \lambda_j \left(\sum_{k=1}^{K} u_{jk} - 1\right)$$

  - Denote
    - $d_{jk} := (y_j - c_k)^2$
    - Squared distance of j-th point from k-th cluster mean

Fuzzy-C-Means Clustering

- FCM Optimization via Lagrange Multipliers
  - Keeping d_{jk} fixed, i.e., cluster-means fixed, solve for Memberships and Lagrange Multipliers

$$\frac{\partial L(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{\partial u_{jk}} = 0 = q u_{jk}^{q-1}(y_j - c_k)^2 + \lambda_j$$

$$u_{jk} = \left(\frac{-\lambda_j}{q d_{jk}}\right)^{\frac{1}{q-1}}$$

$$\frac{\partial L(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{\partial \lambda_j} = 0 = \sum_k u_{jk} - 1$$

$$L(\{u_{jk}\},\{c_k\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q (y_j - c_k)^2 + \lambda_j \left(\sum_{k=1}^{K} u_{jk} - 1\right)$$

Fuzzy-C-Means Clustering

- FCM Optimization via Lagrange Multipliers
  - Keeping d_{jk} fixed, i.e., cluster-means fixed, solve for Memberships and Lagrange Multipliers

$$u_{jk} = \left(\frac{-\lambda_j}{q d_{jk}}\right)^{\frac{1}{q-1}} \qquad 0 = \sum_k u_{jk} - 1$$

$$\sum_k \left(\frac{-\lambda_j}{q d_{jk}}\right)^{\frac{1}{q-1}} = 1 \implies (-\lambda_j)^{\frac{1}{q-1}} = \frac{1}{\sum_k \left(\frac{1}{q d_{jk}}\right)^{\frac{1}{q-1}}}$$

Fuzzy-C-Means Clustering

- FCM Optimization via Lagrange Multipliers
  - Keeping d_{jk} fixed, i.e., cluster-means fixed, solve for Memberships and Lagrange Multipliers
    - Substitute lambda value into the expression for u_{jk}

$$(-\lambda_j)^{\frac{1}{q-1}} = \frac{1}{\sum_k \left(\frac{1}{qd_{jk}}\right)^{\frac{1}{q-1}}}$$

$$u_{jk} = \left(\frac{-\lambda_j}{qd_{jk}}\right)^{\frac{1}{q-1}}$$

$$u_{jk} = \frac{\left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}{\sum_k \left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}$$

Fuzzy-C-Means Clustering

- FCM Optimization via Lagrange Multipliers
  - Update for memberships
    - Positivity constraint met
    - Sum-to-1 constraint met

$$u_{jk} = \frac{\left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}{\sum_k \left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}$$

  - Interpretation
    - If data point j has a larger distance d_{jk} to cluster mean ck, then its membership u_{jk} to cluster k … ???
    - If the user-defined parameter $q \rightarrow \infty$, then $\forall j$, $\forall k$ memberships u_{jk} are … ???
    - If the user-defined parameter $q \rightarrow 1$,

      then membership ... ???

Fuzzy-C-Means Clustering

- FCM Optimization via Lagrange Multipliers
  - Update for memberships
    - Positivity constraint met
    - Sum-to-1 constraint met

$$u_{jk} = \frac{\left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}{\sum_k \left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}$$

  - Interpretation
    - If data point j has a larger distance d_{jk} to cluster mean ck, then its membership u_{jk} to cluster k reduces
    - If the user-defined parameter $q \rightarrow \infty$, then $\forall j$, $\forall k$ memberships u_{jk} are constant, i.e., $u_{jk} = 1 / K$
    - If the user-defined parameter $q \rightarrow 1$, then membership $u_{jk} = 1$ iff $d_{jk} < d_{jk'}$ otherwise $u_{jk} = 0$
      - Hard / crisp clustering (equivalent to K means)

Fuzzy-C-Means Clustering

- FCM Optimization via Lagrange Multipliers
  - Keeping memberships and Lagrange multiplier fixed, solve for class means

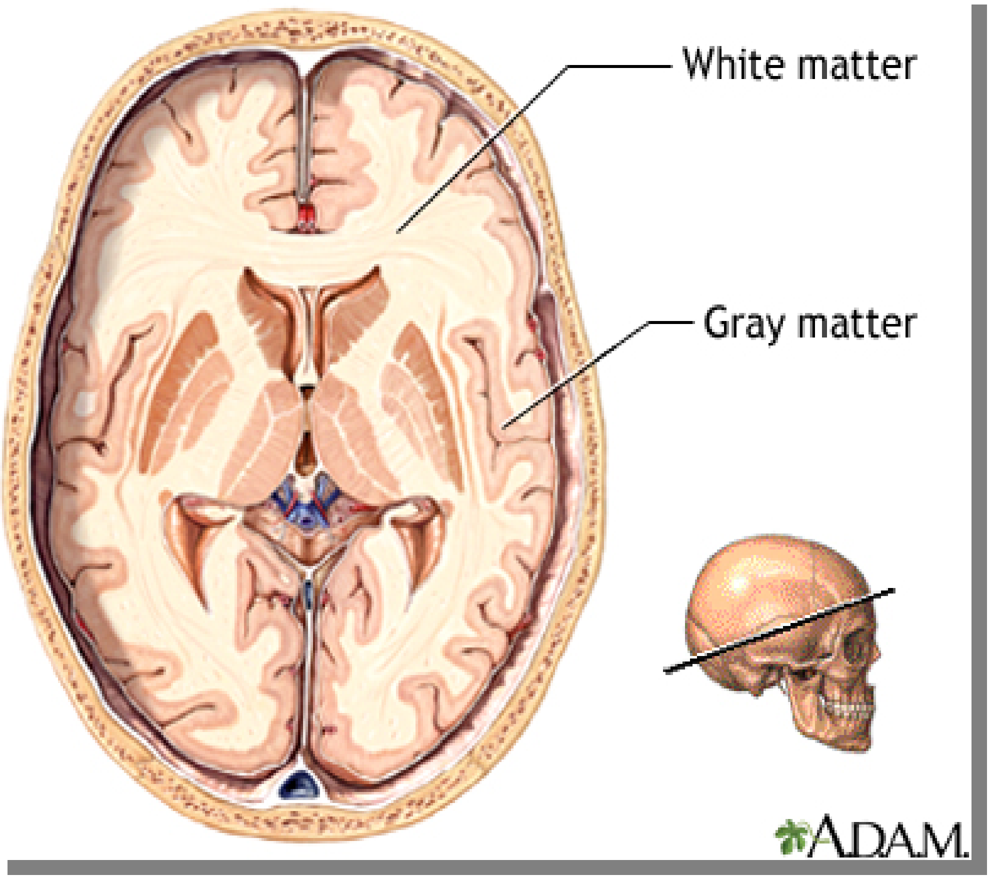
$$\frac{\partial L(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{\partial c_k} = 0 = \sum_{j=1}^{N} u_{jk}^q 2(y_j - c_k) \longrightarrow c_k = \frac{\sum_{j=1}^{N} u_{jk}^q y_j}{\sum_{j=1}^{N} u_{jk}^q}$$

  - Interpretation
    - Class mean is weighted average of data points
    - Weight larger $\leftarrow$ membership larger
  - What happens with memberships are binary ?

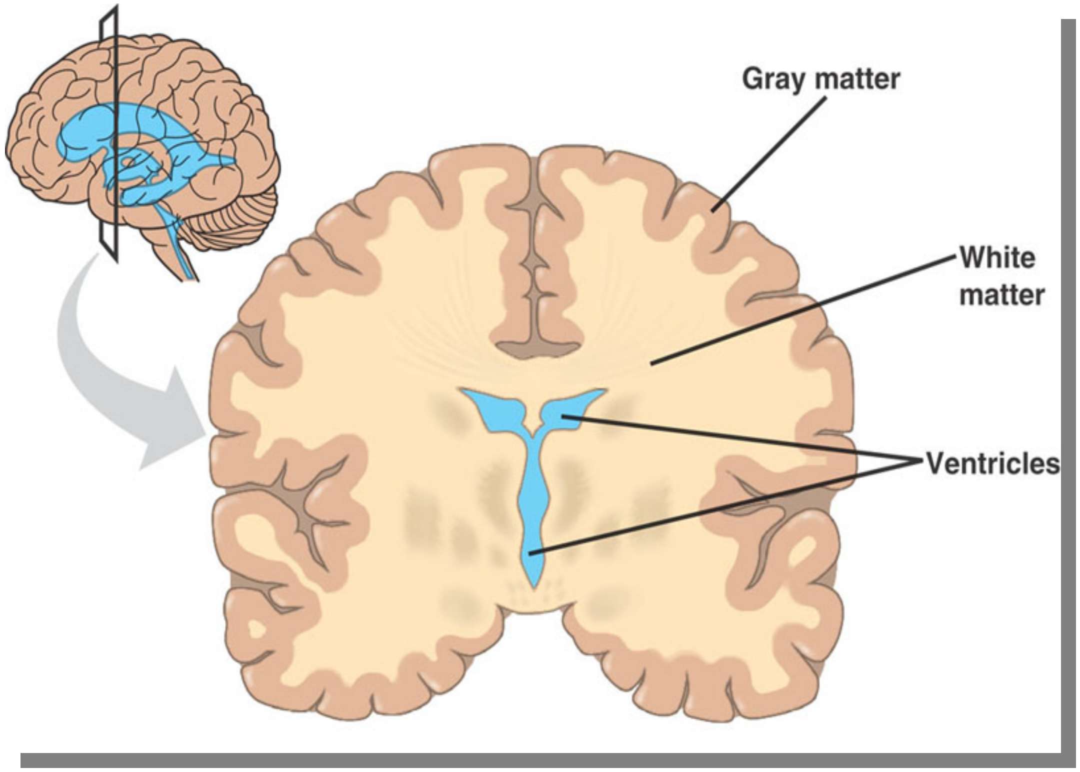
$$L(\{u_{jk}\},\{c_k\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q (y_j - c_k)^2 + \lambda_j \left(\sum_{k=1}^{K} u_{jk} - 1\right)$$

Tissue Segmentation in Brain MRI

- Human brain tissues



www.nlm.nih.gov/medlineplus/ency/imagepages/18117.htm



tgesicsebiology.weebly.com/nervous-system.html

Tissue Segmentation in Brain MRI

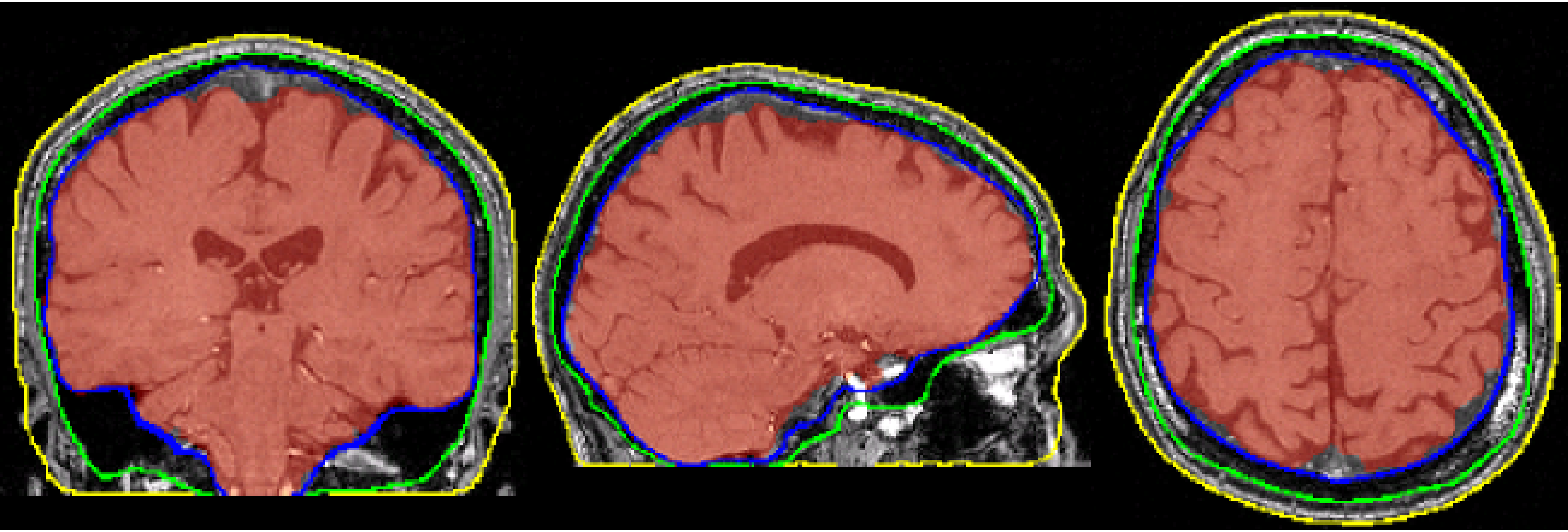
- 2 stages

  (1) Brain extraction from head MR image

Tissue Segmentation in Brain MRI

- 2 stages

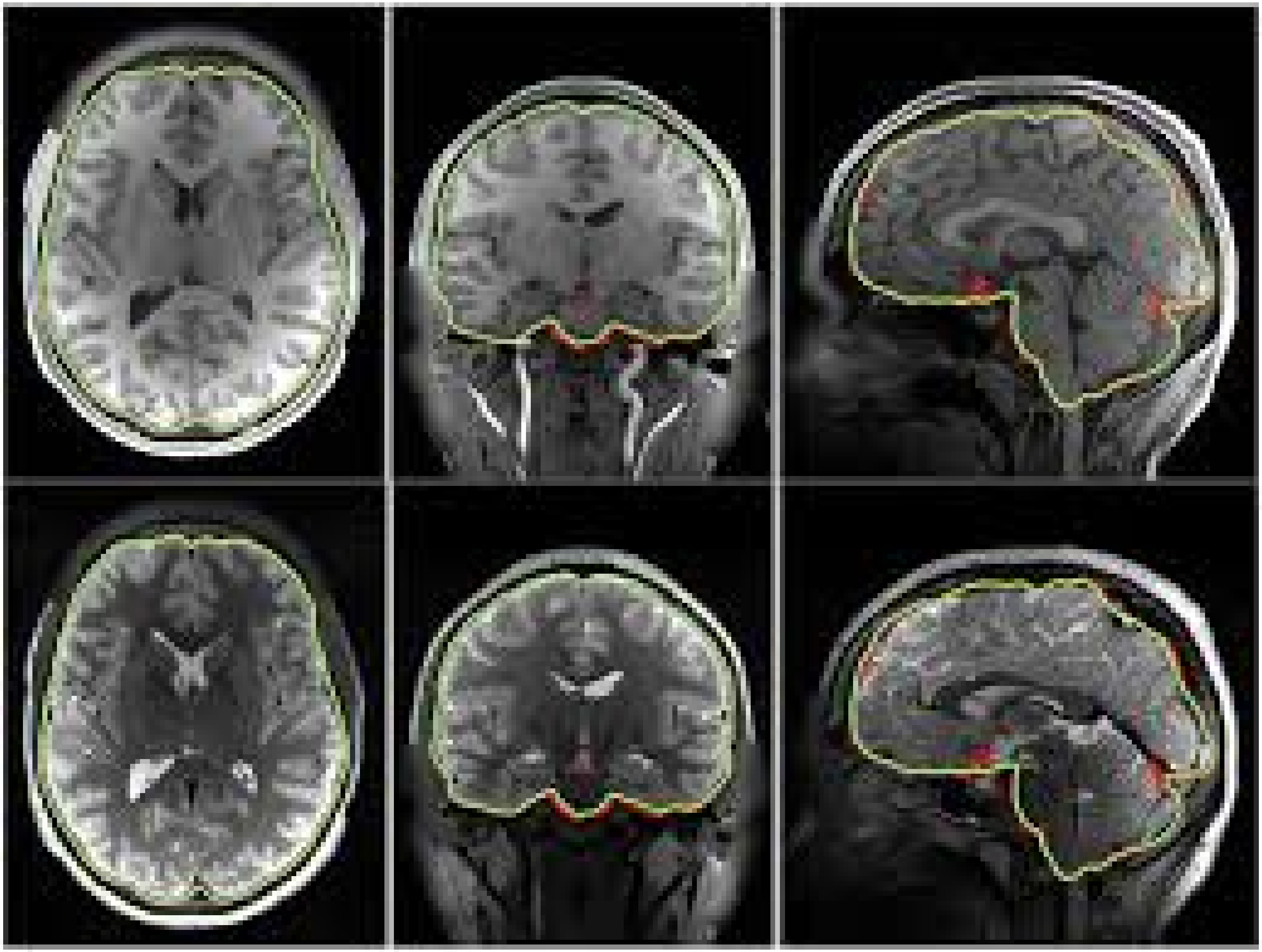
  (1) Brain extraction from head MR image



Tissue Segmentation in Brain MRI

- 2 stages

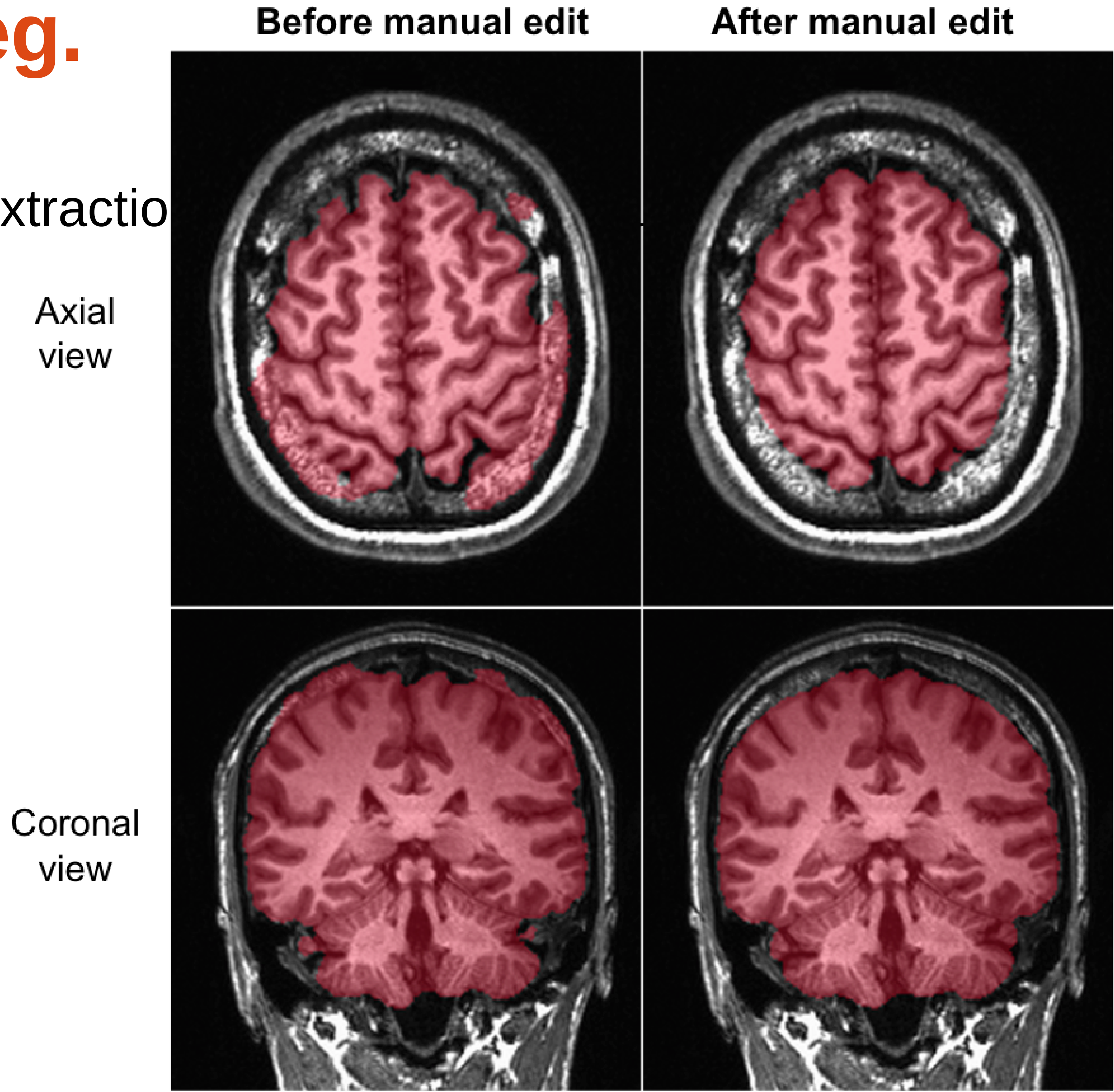
    (1) Brain extraction from head MR image



Tissue Seg.

- 2 stages

    (1) Brain extractio

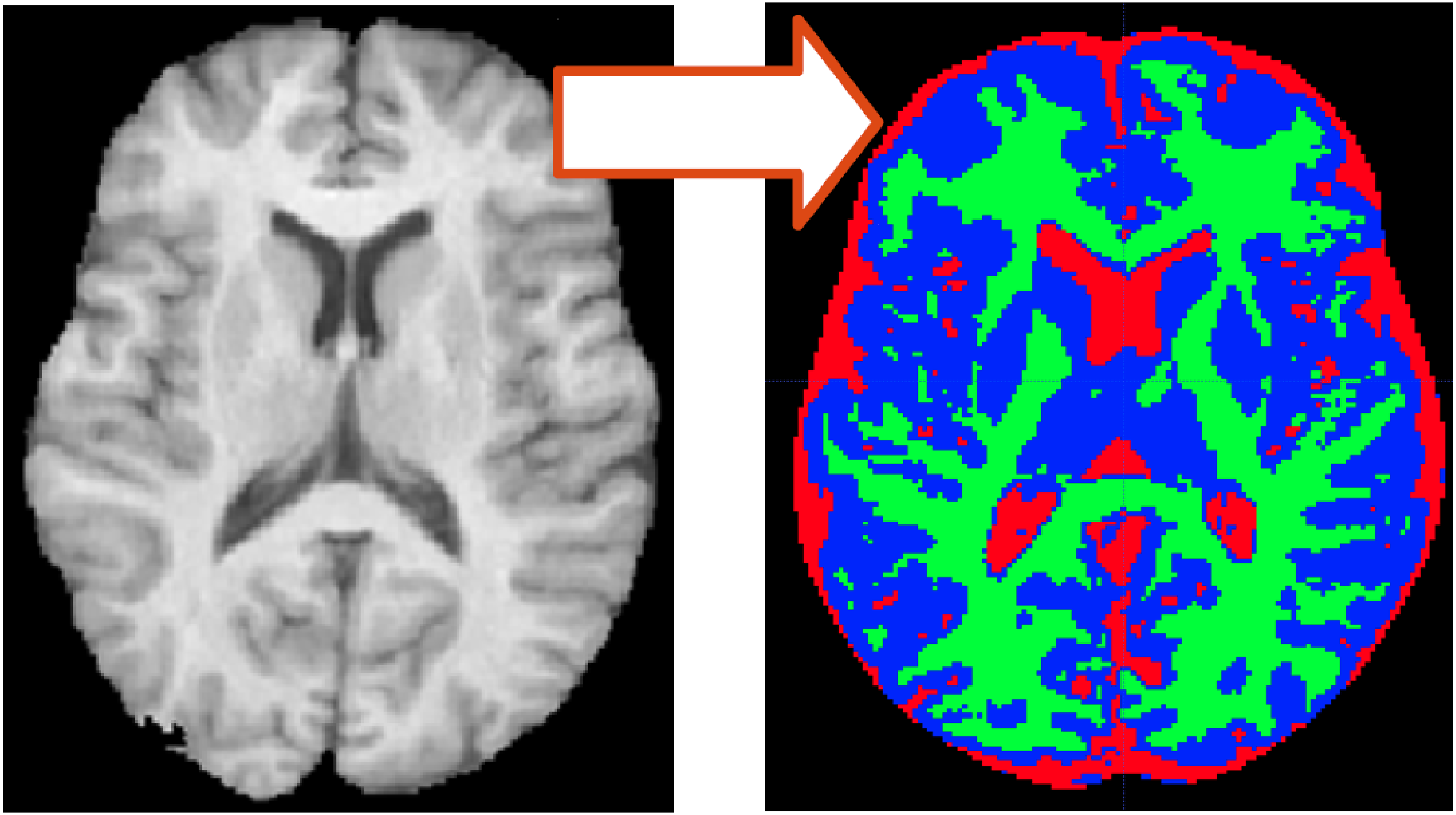


Tissue Segmentation in Brain MRI

- 2 stages

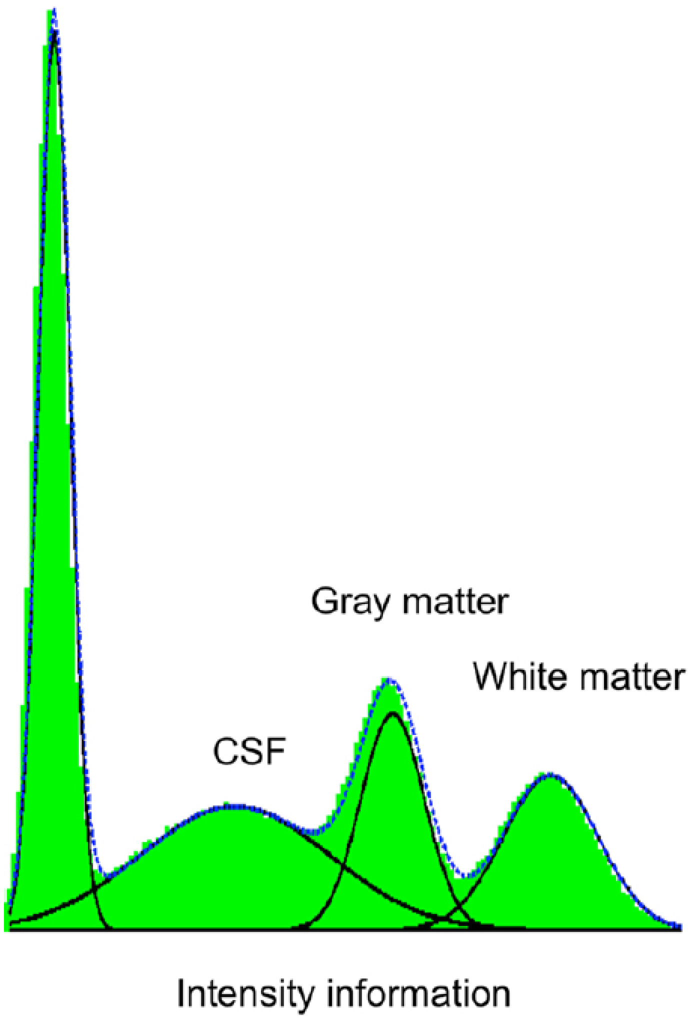
  (1) Brain extraction from head MR image

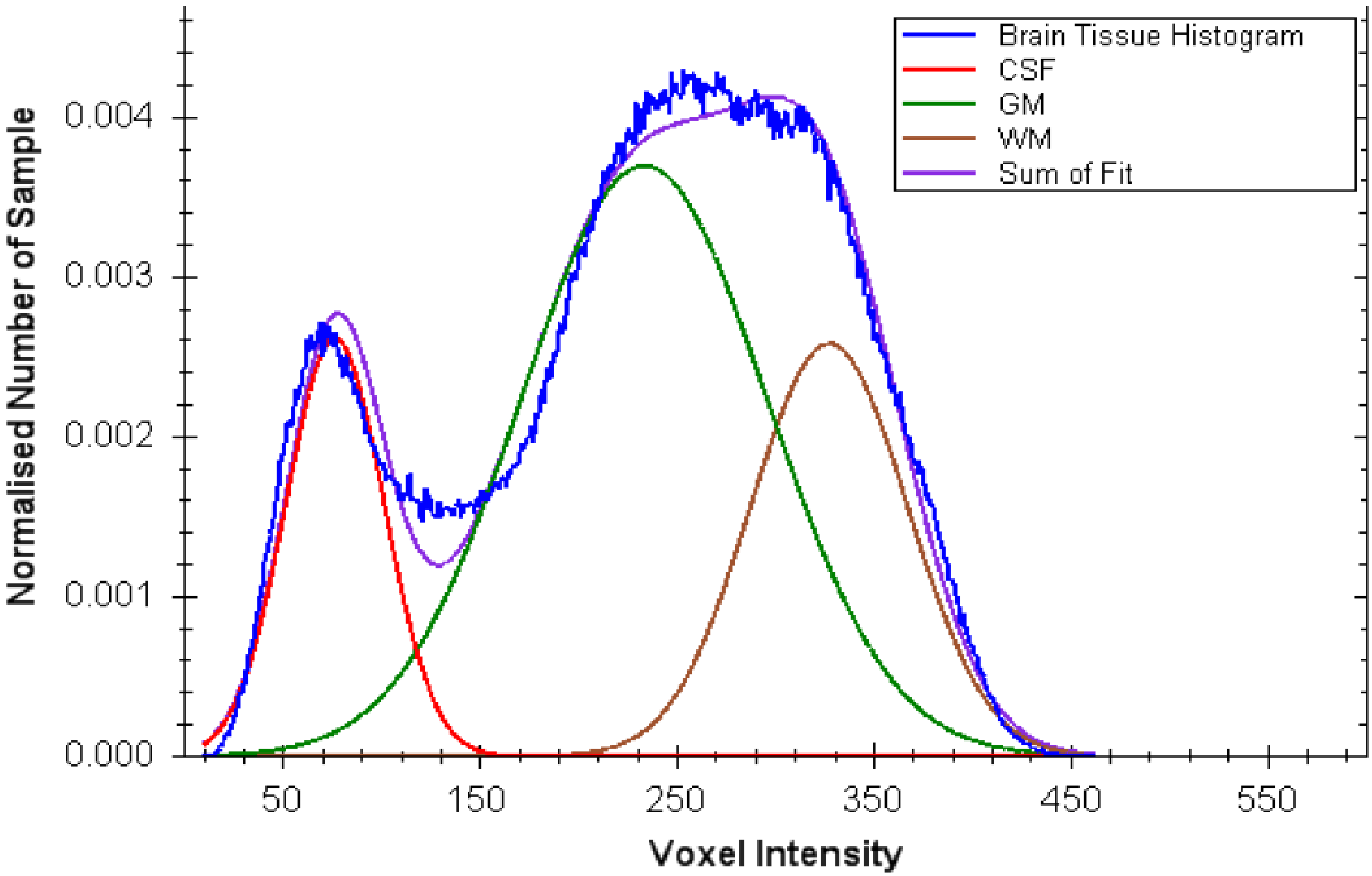
  (2) Tissue segmentation within brain



Tissue Segmentation in Brain MRI

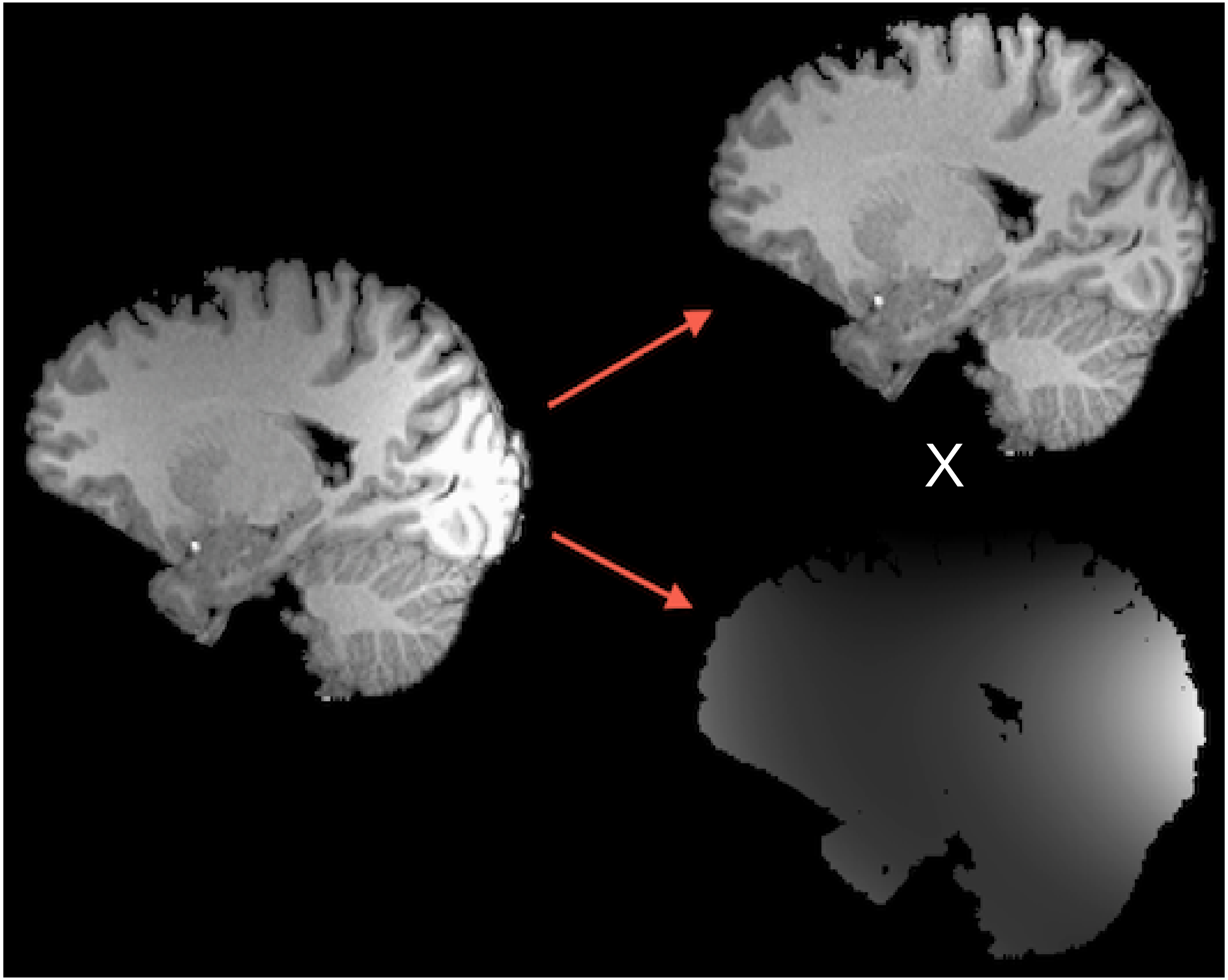
- PDF of intensities in T1-weighted brain MRI





Tissue Segmentation in Brain MRI

- Bias / inhomogeneity field in brain MRI
  - Imperfect static b1 magnetic field (RF pulse)
  - Multiplicative corruption on intensities
  - Spatially smooth



Vignetting

- Reduction of an image's brightness or saturation at the periphery compared to the image center
  - Often unintended / undesired ← cam. settings / lens limits
  - Sometimes introduced to draw attention to frame center

FCM Seg. + Bias-Field Correction

- MR magnitude image with N voxels
- At voxel i, intensity is x_i (unknown)
- At voxel i, inhomogeneity / bias is b_i (unknown)
  - Assumption: Varies smoothly over space
- Consider noise to be i.i.d. additive Gaussian
  - Approximation valid for voxels with high-SNR
  - Otherwise, can work with complex values
- At voxel i, observed intensity is $y_i := x_i b_i + \eta_i$
  - $\eta_i \sim G(0, \sigma^2)$

FCM Seg. + Bias-Field Correction

- 2 more model assumptions
  - Biologically reasonable
  - Image comprises K classes (known K)
  - Each class k has a constant MR (uncorrupted) intensity = ck
- Modeling assumptions lead to :
  - At voxel i,
    if its neighbor j belongs to class k,
    then, xj = ck (constant intensity of tissue)
    bj ≈ bi (spatial smoothness of bias)
    and, thus, xj bj ≈ ck bi

FCM Seg. + Bias-Field Correction

- Strategy
  - At voxel j in neighborhood of voxel i,
    penalize difference between
    (i) observed neighborhood intensities yj and
    (ii) intensities predicted by model xj bj
    - If neighborhood voxel j belongs to class k,
      then, predicted intensity = xj bj ≈ ck bi
  - Because we don’t know which class voxel j is in,
    so penalize differences for all classes
    weighted by membership u_{jk}^q in that class
  - Because bias-constancy is local,
    weight differences based on distance between voxels i, j
    - Weight $w_{ij} \rightarrow 0$ when distance > threshold
    - Implement as a neighborhood “mask”. Weights sum to 1.

FCM Seg. + Bias-Field Correction

- Objective function, per voxel i,

$$J_i := \sum_{j=1}^{N} w_{ij} \sum_{k=1}^{K} \left(u_{jk}^{q} (y_j - c_k b_i)^2 \right)$$

- Constraints
  - Membership, weights $\forall j; \sum_{k=1}^{K} u_{jk} := 1$ $\quad \sum_{j=1}^{N} w_{ij} = 1$

- Objective function $J := \sum_{i=1}^{N} J_i := \sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^{q} \left(\sum_{i=1}^{N} w_{ij} (y_j - c_k b_i)^2 \right)$

- Compare with standard FCM: $\sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^{q} (y_j - c_k)^2$

FCM Seg. + Bias-Field Correction

- Denote

$$d_{kj} := \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2$$

  - dkj depends only on class-number k & voxel-number j
- Rewritten optimization problem

$$\min_{\{u_{jk}\},\{c_k\},\{b_i\}} J(\{u_{jk}\},\{c_k\},\{b_i\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q d_{kj}$$

$$\text{constraints : } \forall j, \sum_k u_{jk} = 1$$

FCM Seg. + Bias-Field Correction

- Optimization via Lagrange Multipliers
  - Lagrangian

$$L(\{u_{jk}\}, \{c_k\}, \{b_i\}, \{\lambda_j\}) := \sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^q d_{kj} + \sum_{j=1}^{N} \lambda_j \left(1 - \sum_{k} u_{jk} \right)$$

$$\boxed{d_{kj} := \sum_{i=1}^{N} w_{ij} (y_j - c_k b_i)^2}$$

FCM Seg. + Bias-Field Correction

- Optimization via Lagrange Multipliers
  - Keeping class means and bias fixed,
    solve for memberships and Lagrange multipliers

$$\frac{\partial L(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{\partial u_{jk}} = 0 = q u_{jk}^{q-1} d_{kj} - \lambda_i$$

$$\frac{dL(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{d\lambda_i} = 0 = 1 - \sum_k u_{jk}$$

FCM Seg. + Bias-Field Correction

- Optimization via Lagrange Multipliers
  - Keeping class means and bias fixed,
    solve for memberships and Lagrange multipliers

$$u_{jk} = \frac{\left(\frac{1}{d_{kj}}\right)^{\frac{1}{q-1}}}{\sum_k \left(\frac{1}{d_{kj}}\right)^{\frac{1}{q-1}}}$$

$$d_{kj} := \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2$$

- Same as in FCM, except that dkj has a different meaning
  - dkj generalizes / smooths the squared distance over neighborhood
  - How to choose width of mask w ?
    - Smoothness of bias field b (a priori)
  - What happens when no bias (bi=1) leads to choice of wide mask (wij=const.) ?

FCM Seg. + Bias-Field Correction

- Optimization via Lagrange Multipliers
  - Keeping memberships, multipliers, and bias fixed, solve for class means

$$\frac{\partial L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\})}{\partial c_k} = 0 = \sum_j u_{jk}^q \frac{\partial d_{kj}}{\partial c_k} = \sum_j u_{jk}^q \sum_i w_{ij} 2(y_j - c_k b_i)(-b_i)$$

  - Rearranging terms :

$$c_k = \frac{\sum_j u_{jk}^q y_j \sum_i w_{ij} b_i}{\sum_j u_{jk}^q \sum_i w_{ij} b_i^2}$$

$$d_{kj} := \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2$$

$$L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q d_{kj} + \sum_{j=1}^{N} \lambda_j \left(1 - \sum_k u_{jk}\right)$$

FCM Seg. + Bias-Field Correction

- Optimization via Lagrange Multipliers
  - Keeping memberships, multipliers, and bias fixed, solve for class means

$$c_k = \frac{\sum_j u_{jk}^q y_j \sum_i w_{ij} b_i}{\sum_j u_{jk}^q \sum_i w_{ij} b_i^2}$$

  - Interpretation
    - Summation over wij performs convolution with the neighborhood mask (which smooths the bias field)
    - If bias field is constant = b, then ck = membership-weighted avg of ratios (yj / b)
    - If bias field varies slowly over neighborhood mask w + memberships are binary, then $c_k \approx \sum_{j \in C^k} y_j b_j / \sum_{j \in C^k} b_j^2$

FCM Seg. + Bias-Field Correction

- Optimization via Lagrange Multipliers
  - Keeping memberships, multipliers, and class means fixed, solve for bias

$$\frac{\partial L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\})}{\partial b_i} = 0 = \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q \frac{\partial d_{kj}}{\partial b_i} = \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q w_{ij} 2(y_j - c_k b_i) c_k$$

- Rearranging terms :

$$b_i = \frac{\sum_{j=1}^{N} w_{ij} y_j \sum_{k=1}^{K} u_{jk}^q c_k}{\sum_{j=1}^{N} w_{ij} \sum_{k=1}^{K} u_{jk}^q c_k^2}$$

$$d_{kj} := \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2$$

$$L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q d_{kj} + \sum_{j=1}^{N} \lambda_j \left(1 - \sum_k u_{jk}\right)$$

FCM Seg. + Bias-Field Correction

- Optimization via Lagrange Multipliers
  - Keeping memberships, multipliers, and class means fixed, solve for bias

$$b_i = \frac{\sum_{j=1}^{N} w_{ij} y_j \sum_{k=1}^{K} u_{jk}^q c_k}{\sum_{j=1}^{N} w_{ij} \sum_{k=1}^{K} u_{jk}^q c_k^2}$$

  - Interpretation
    - If memberships binary, then $b_i \approx \sum_j w_{ij} y_j c_{k(j)} / \sum_j w_{ij} c_{k(j)}^2$
    - If, in addition, all neighbors are in class k,
      then bias bi = spatially-weighted average ratios of (yj / ck)
    - Spatial smoothing
      - Helps counter noise in observed data
      - Ensures a smooth bias-field estimate

FCM Seg. + Bias-Field Correction

- Did we formulate the problem right ?
- Is the model identifiable ?
- Can class means & bias field be uniquely identified ?
  - Model: $y_i := x_i b_i + \eta_i$
    - $y_i := c_k b_i + \eta_i$, if voxel i belongs (100%) to class k
  - What happens when, say, noise is absent ?

$$d_{kj} := \sum_{i=1}^{N} w_{ij} (y_j - c_k b_i)^2$$

$$L(\{u_{jk}\}, \{c_k\}, \{b_i\}, \{\lambda_j\}) := \sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^q d_{kj} + \sum_{j=1}^{N} \lambda_j \left(1 - \sum_{k} u_{jk} \right)$$

K-Means and FCM

- Limitations
  - Can't model clusters with **unequal-spread**
    - How to do that ?
      - Gaussian model (variance)
  - Can't model **non-spherical** clusters (in higher-D)
    - How to do that ?
      - Gaussian model (covariance)
  - Don't enforce **spatial smoothness** on segmentation
    - How to do that ?
      - MRF model on membership image
  - Can't model **multimodal** clusters and '**curved**' clusters
    - How to do that ?

Gauss

- Carl Friedrich Gauss (1777 – 1855)
  - “Prince of mathematicians”
    - “Greatest mathematician since antiquity”
    - Number theory, algebra, statistics, analysis, differential geometry, geophysics, electrostatics,astronomy,optics
  - Child prodigy
    - Published Arithmetical Investigations (age 21), consolidating number theory as a discipline
    - PhD in 1799 (age 22)
      - Provided a new proof of the fundamental theorem of algebra
        - Every non-zero, single-variable, **degree-n polynomial** with complex coefficients has, counted with multiplicity, exactly **n roots**
  - FFT was invented by Gauss in 1805 (age 28)
    - Cooley and Tukey reinvented it in 1965 and popularized it

Gauss

- Carl Friedrich Gauss (1777 – 1855)
  - Refused to publish “unripe” work (“few, but ripe”)
    - Personal diary indicates discoveries before contemporaries
  - Declined to provide intuition behind his elegant proofs
    - Erased all traces of discovery process
    - Wanted proofs to look 'magical'
    - All analysis must be suppressed for the sake of brevity

Bernhard Riemann

- Mathematician
  - Analysis, number theory, differential geometry
  - Advisor: Gauss
- As a child, Riemann showed exceptional math skills, but was shy of public speaking
  - In school, he often outstripped his instructor's knowledge
- Father was a pastor. In high school, he studied Bible intensively, but was often distracted by mathematics
- Went to university to get degree in Theology. Met Gauss, who suggested Riemann to get into math

Riemann

- Clay math institute's Millennium Prize Problems includes 'Riemann hypothesis'

Number theory [edit]

- Riemann–von Mangoldt formula
- Riemann hypothesis
    - Generalized Riemann hypothesis
    - Grand Riemann hypothesis
    - Riemann hypothesis for curves over finite fields
- Riemann theta function
- Riemann Xi function
- Riemann zeta function
- Riemann–Siegel formula
- Riemann–Siegel theta function

Riemannian [edit]

- Pseudo-Riemannian manifold
- Riemannian bundle metric
- Riemannian circle
- Riemannian cobordism
- Riemannian connection
- Riemannian connection on a surface
- Riemannian cubic
- Riemannian cubic polynomials
- Riemannian foliation
- Riemannian geometry
    - Fundamental theorem of Riemannian geometry
- Riemannian graph
- Riemannian group
- Riemannian holonomy
- Riemannian manifold also called Riemannian space
- Riemannian metric tensor
- Riemannian Penrose inequality
- Riemannian polyhedron
- Riemannian singular value decomposition
- Riemannian submanifold
- Riemannian submersion
- Riemannian volume form
- Riemannian wavefield extrapolation
- Sub-Riemannian manifold
- Riemannian symmetric space

Riemann's [edit]

- Riemann's differential equation
- Riemann's existence theorem
- Riemann's explicit formula
- Riemann's minimal surface
- Riemann's theorem on removable singularities

- Free Riemann gas also called primon gas
- Riemann bilinear relations
- Riemann–Cartan geometry
- Riemann conditions
- Riemann curvature tensor also called Riemann tensor
- Riemann form
- Riemann function
- Riemann–Hurwitz formula
- Riemann invariant
- Riemann matrix
- Riemann operator
- Riemann problem
- Riemann–Silberstein vector
- Riemann singularity theorem
- Riemann surface
    - Compact Riemann surface
- Riemann tensor (general relativity)
- The tangential Cauchy–Riemann complex
- Zariski–Riemann space

Analysis [edit]

- Riemann integral
    - Generalized Riemann integral
    - Riemann multiple integral
- Cauchy–Riemann equations
- Riemann mapping theorem
    - Measurable Riemann mapping theorem
- Riemann solver
- Riemann sphere
- Riemann–Hilbert correspondence
- Riemann–Hilbert problem
- Riemann–Lebesgue lemma
- Riemann–Liouville differintegral
- Riemann–Roch theorem
    - Arithmetic Riemann–Roch theorem
    - Riemann–Roch theorem for smooth manifolds
    - Grothendieck–Hirzebruch–Riemann–Roch theorem
    - Hirzebruch–Riemann–Roch theorem
- Riemann–Stieltjes integral
- Riemann series theorem
- Riemann sum

Mixture Model

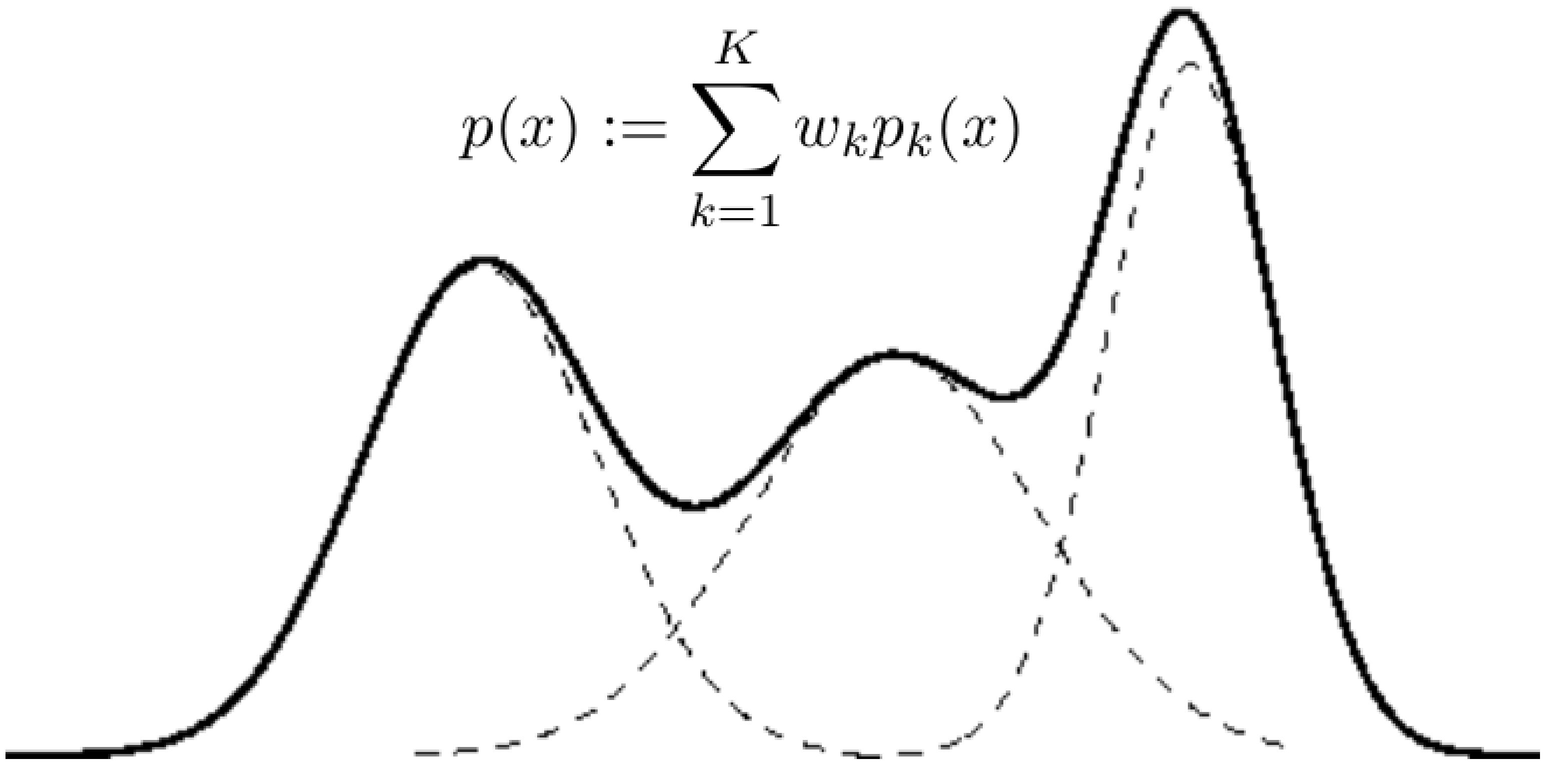
- What is a mixture model ?
  - Probabilistic model
  - Represents a general multi-modal PDF
  - Useful for observed data that is typically derived from multiple subgroups within a population
- Represent PDF p(x) as a convex combination of standard PDFs $p_k(x)$

$$p(x) := \sum_{k=1}^{K} w_k p_k(x)$$

  - Weights wk are positive (why ?) and sum to 1 (why ?)

Mixture Model

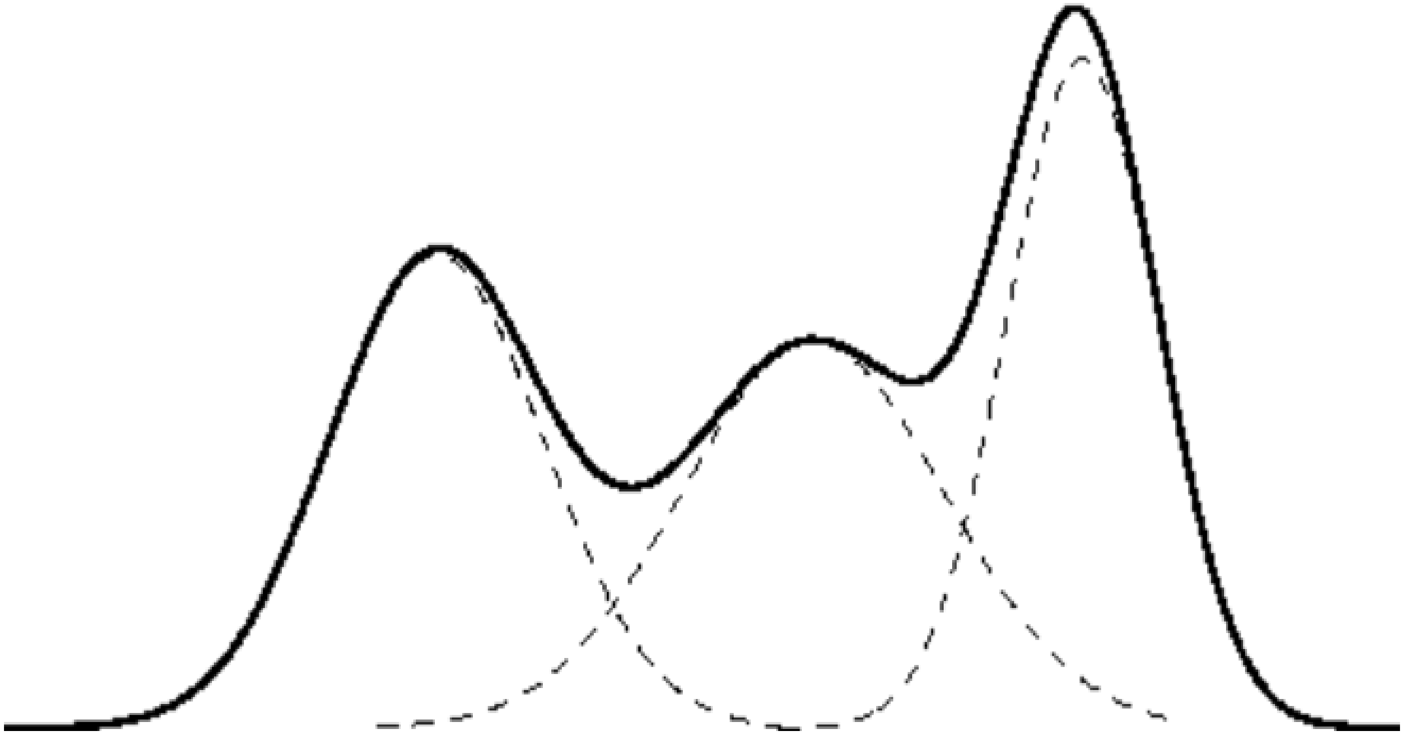
- How to interpret weights ?
  - Larger weight wk $\rightarrow$
    data more likely to be derived from k-th Gaussian
- How to generate data from such a model ?



Mixture Model

- When number of PDFs K is finite
  - Finite mixture model
- When each PDF is a Gaussian
  - Gaussian mixture model (GMM)
- Power of the GMM model
  - Can model unimodal / multimodal / curved / complex PDFs

$$p(x) = \sum_{k=1}^{K} w_k G(x; \mu_k, C_k)$$



Mixture Model

- What happens when we fit a Gaussian to 2D data that lies on a circle ? semi circle ?

Gaussian Mixture Model

- Fitting a GMM to data
  - **Given**: Data y := { yn } n=1,…,N
    - Each observation yn is drawn independently of the others
  - **Goal**: Fit a GMM with K components
    - $P(x) := \sum_{k=1}^{K} w_k G(x; \mu_k, C_k)$
    - *What are the ways in which datum 'x' could be generated ?*
    - Estimate weights wk, means μk, covariances Ck
  - **Strategy**
    - Maximum-likelihood estimation
    - Parameters $\theta = \{w_k, \mu_k, C_k\}_{k=1}^{K}$
    - What is the likelihood function ?

Gaussian Mixture Model

- Fitting a GMM to data
  - Likelihood function

$$L(\theta|y) := P(y|\theta) := \Pi_{n=1}^{N} P(y_n|\theta) := \Pi_{n=1}^{N} \left(\sum_{k=1}^{K} w_k G(y_n; \mu_k, C_k) \right)$$

  - Maximizing the log-likelihood

$$\max_{\theta} L(\theta|y) = \max_{\theta} \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} w_k G(y_n; \mu_k, C_k) \right)$$

    - Log sum can't be simplified further !

Gaussian Mixture Model

- Fitting a GMM to data

  (1) No closed-form updates possible

  (2) Gradient-ascent optimization

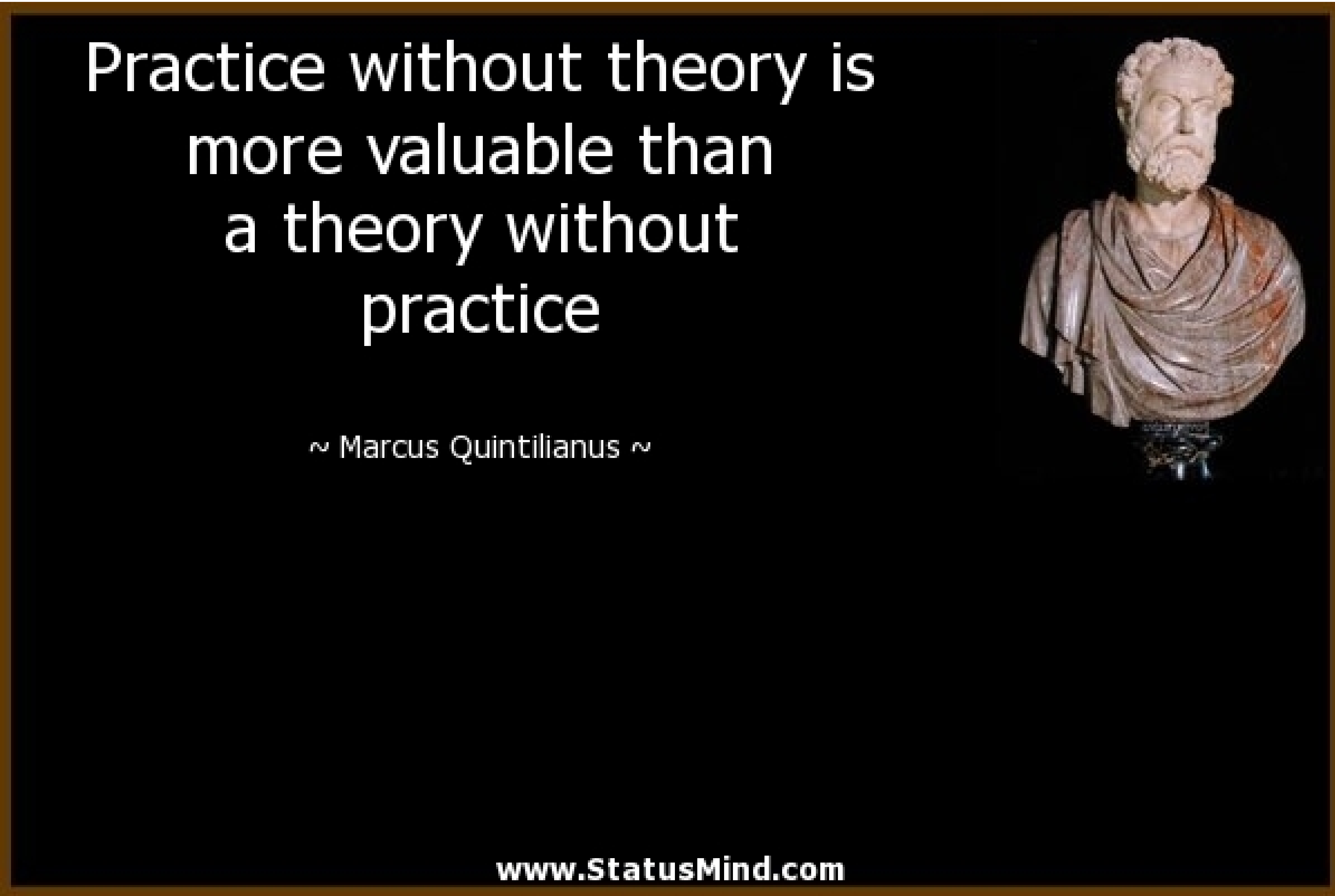
  - Converges very slowly in practice

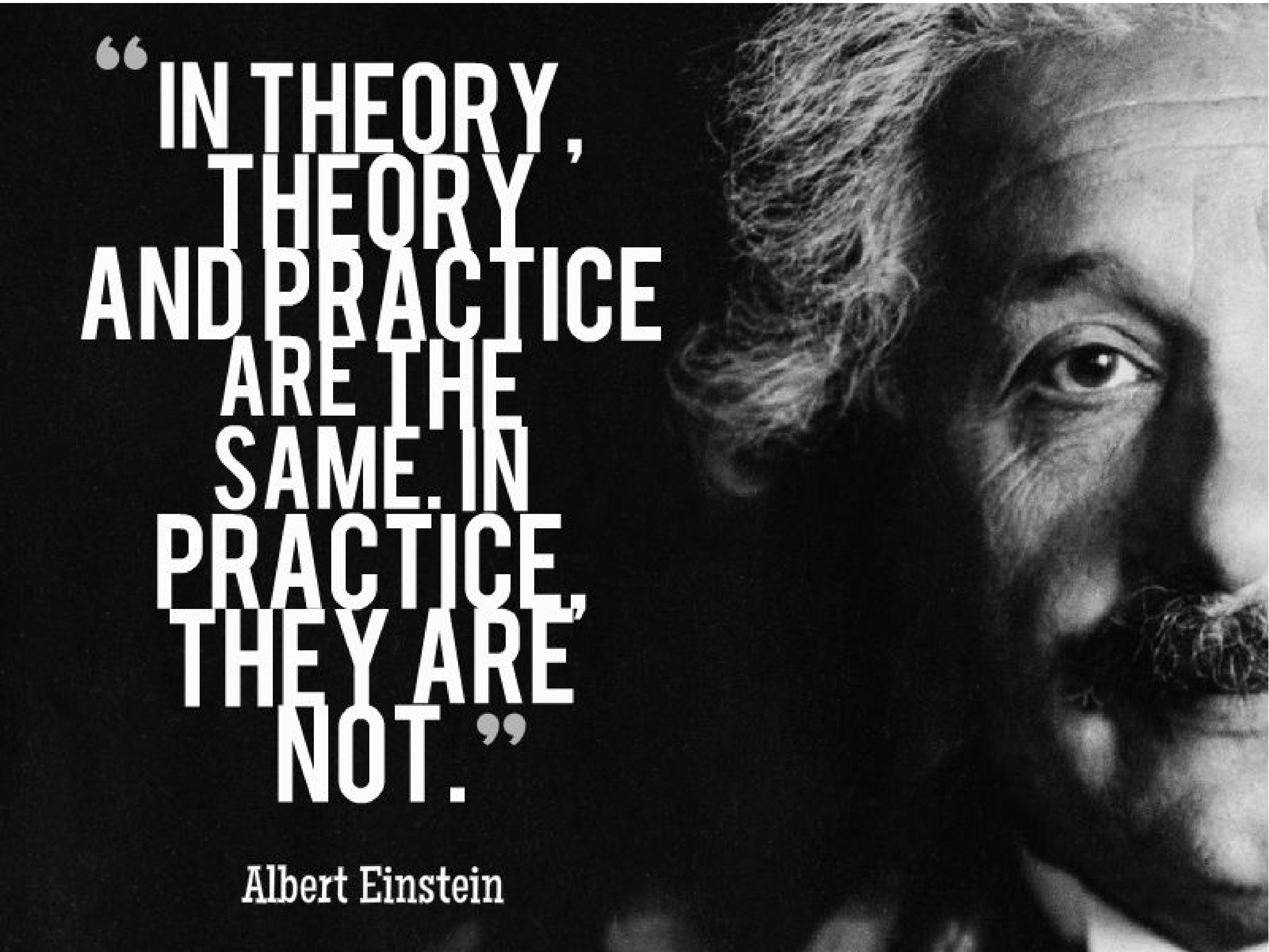
  (3) Is there a better alternative that alleviates both above problems ?

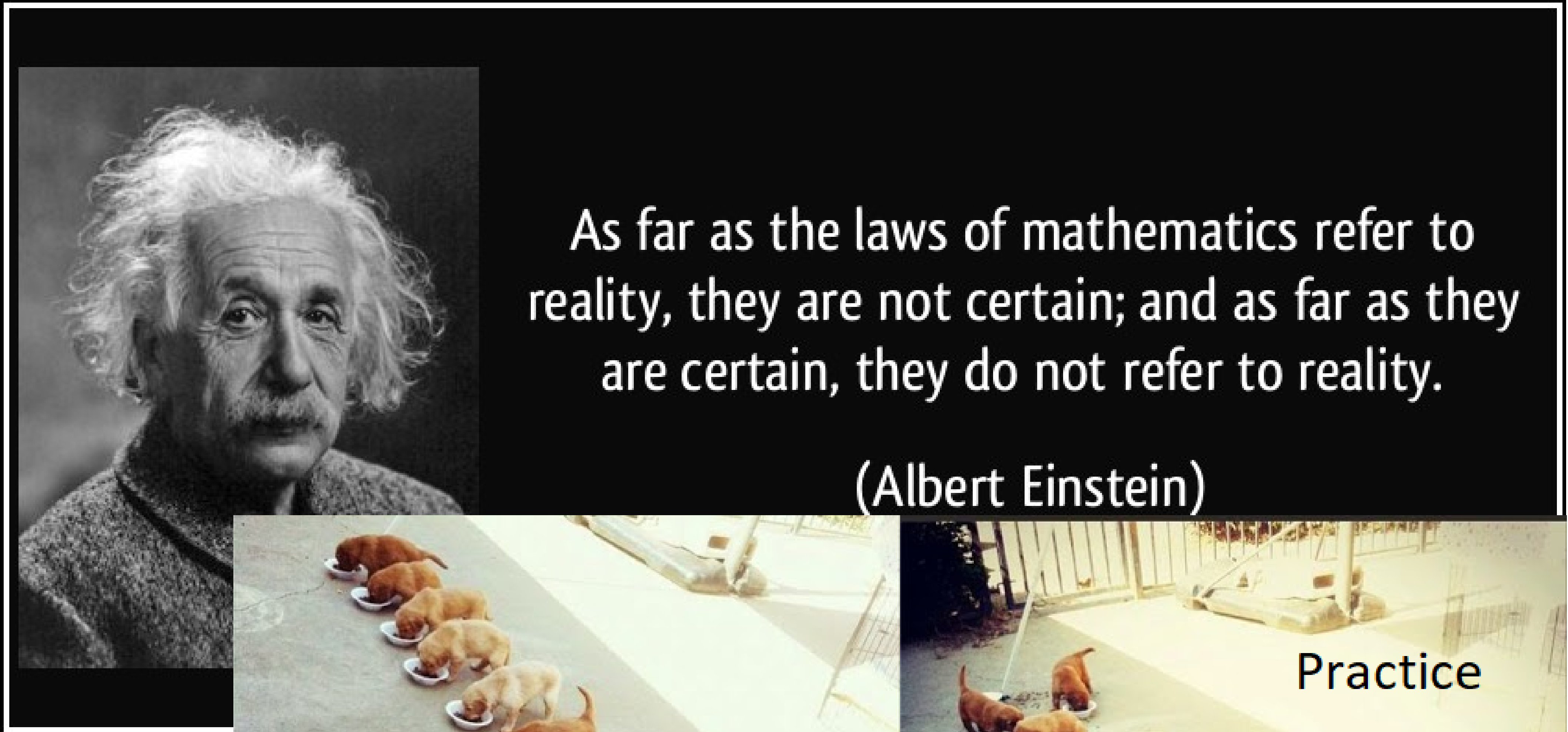
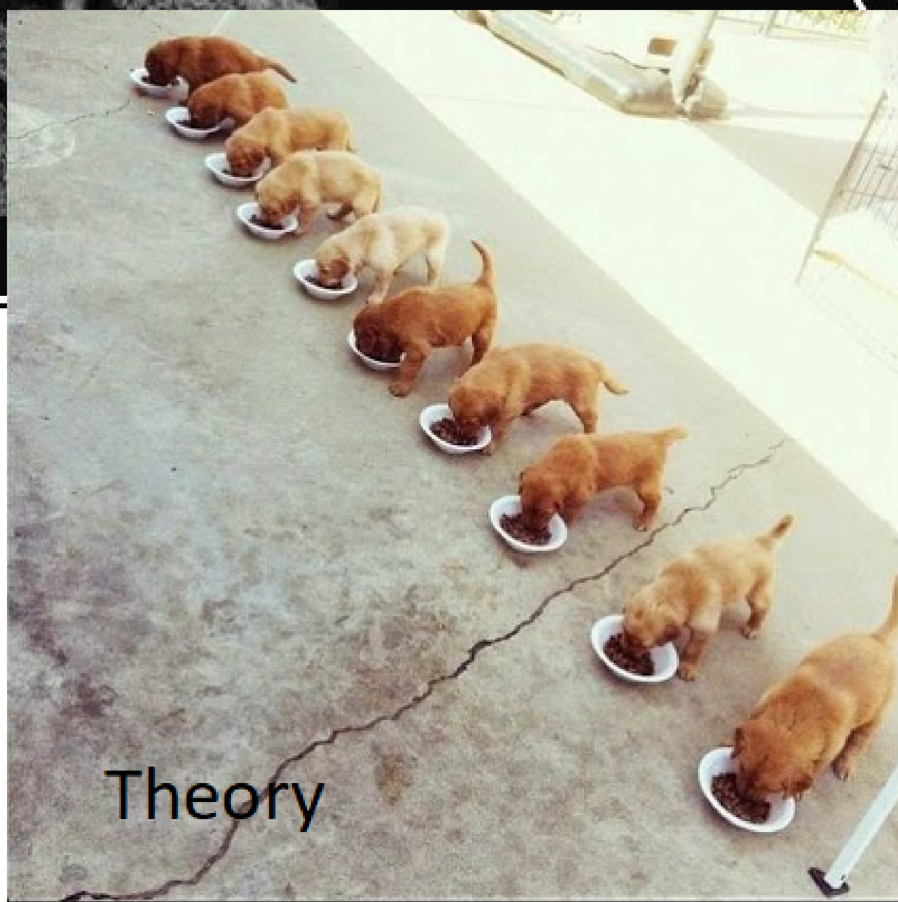
  - Yes
  - Expectation maximization (EM) optimization algorithm

'There is nothing more practical than a good theory'.

*Phrase attributed to Kurt Lewin, German-American psychologist, known as one of the modern pioneers of social, organizational, and applied psychology.


Practice without theory is more valuable than a theory without practice
~ Marcus Quintilianus ~
www.StatusMind.com


"IN THEORY, THEORY AND PRACTICE ARE THE SAME. IN PRACTICE, THEY ARE NOT."
Albert Einstein


As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.
(Albert Einstein)

Theory

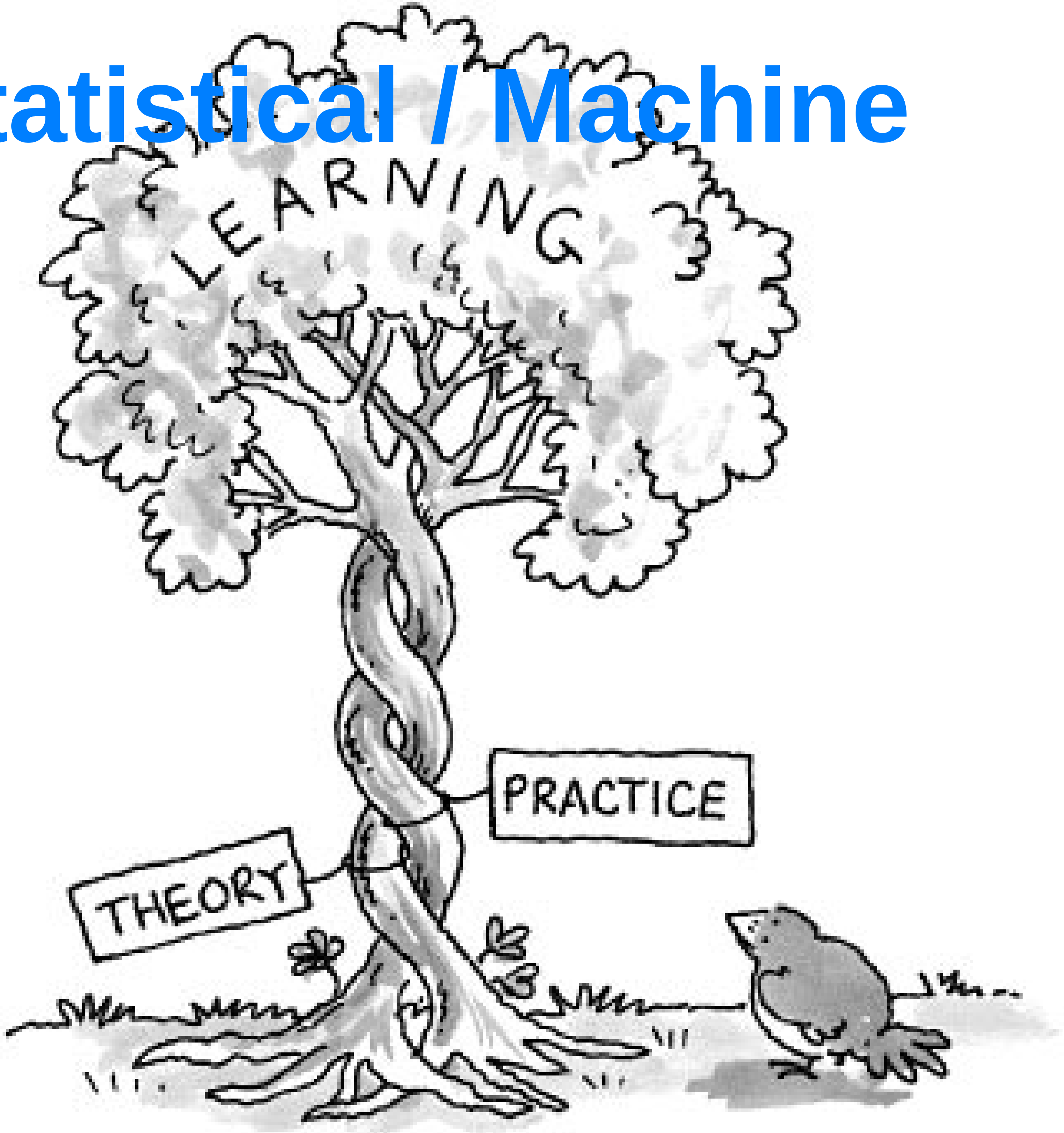
Practice

"Theory without practice is empty.

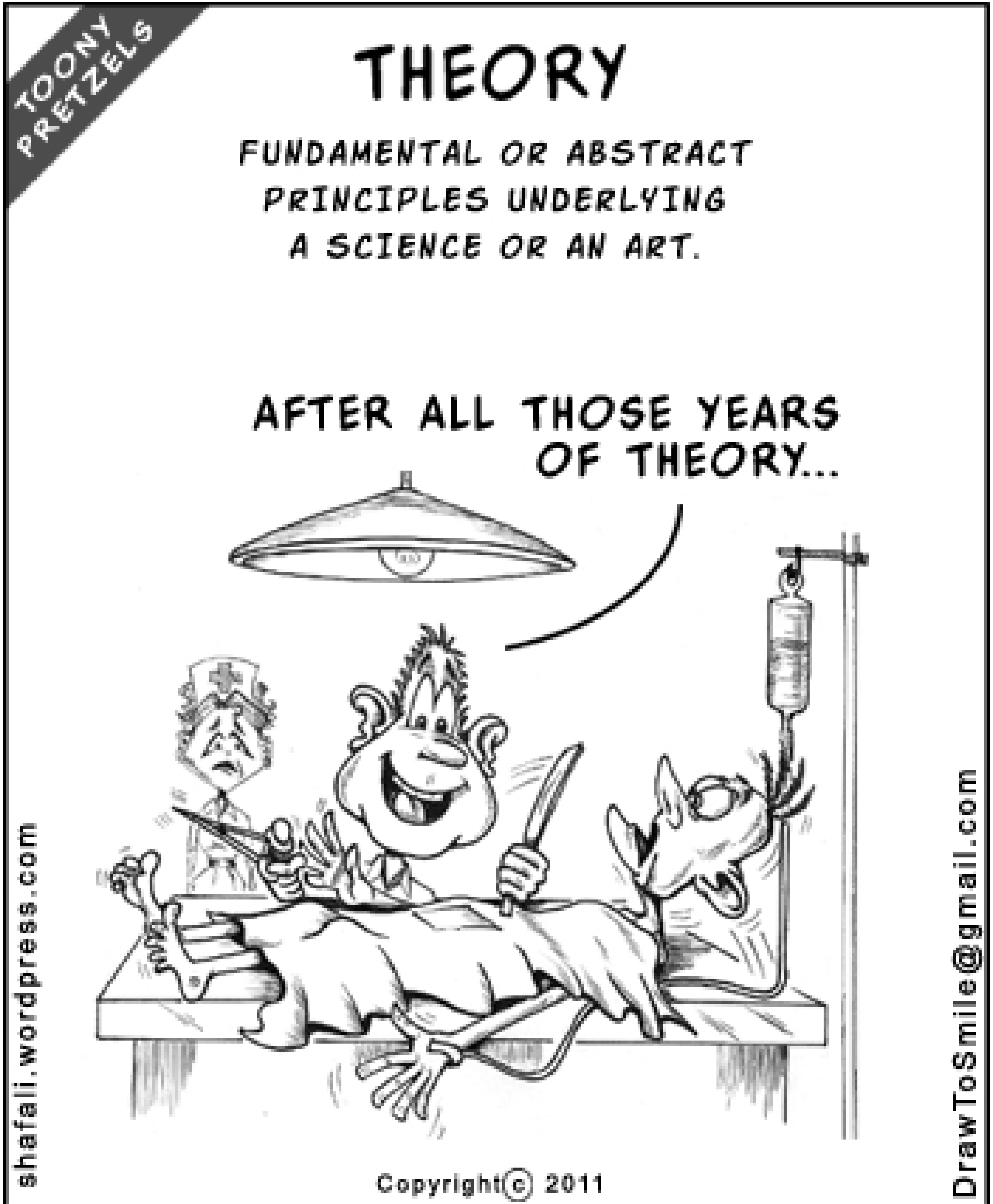
Practice without theory is blind."

J.R. Kidd

Statistical / Machine

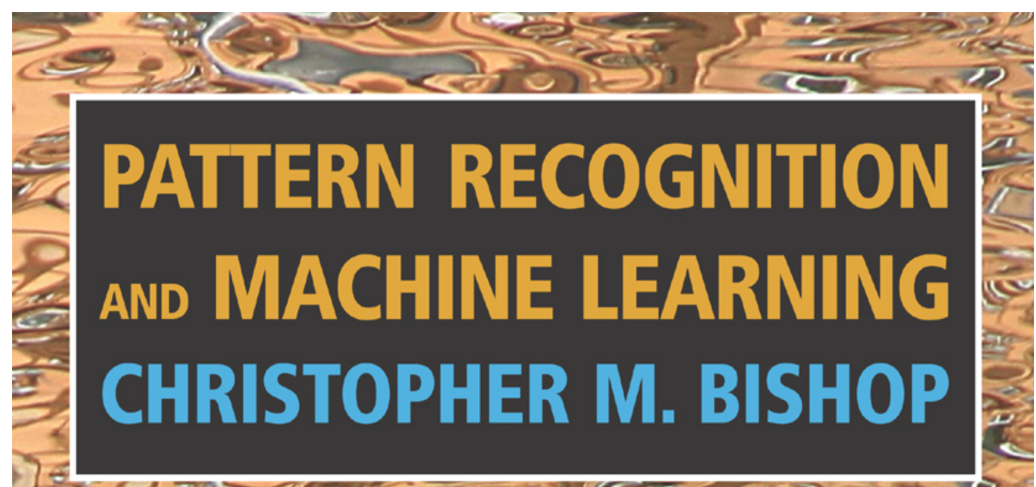


Theory is when you know everything but nothing works.
Practice is when everything works but no one knows why.
In our lab, theory and practice are combined: nothing works and no one knows why.


TOONY PRETZELS
THEORY
FUNDAMENTAL OR ABSTRACT PRINCIPLES UNDERLYING A SCIENCE OR AN ART.
AFTER ALL THOSE YEARS OF THEORY...
shafali.wordpress.com
DrawToSmile@gmail.com
Copyright(c) 2011

EM Optimization

- Observed random variable $Y := \{ Y_n \}$ $n=1,\ldots,N$
  - Leads to data $y := \{ y_n \}$ $n=1,\ldots,N$
- Parameters: θ that needs to be estimated
- Hidden / latent random variable X
  - "Data" x (corresponding to instances of X) are missing
- Complete Data
  = combination of observed + missing data
  $= (x, y) = \{ (x_n , y_n) \}$ $n=1,\ldots,N$



EM Optimization

- Key Idea
  - Hidden variable X is designed (creatively) s.t.
    if missing data x were available,
    then, it would become easy to solve for parameters
- Example : Segmentation
  - Design hidden variable = cluster label corresponding to each data point
    - Let xn be label associated with data point yn s.t.
      xn = cluster label that yn belongs to
    - If we know cluster labels (data points that belong to cluster k), then it is easy to find optimal values of μk, Ck
    - If we know parameters wk, μk, Ck,
      then ...

EM Optimization

- EM optimization performs ML estimation
  - ML estimate is :

$$\max_\theta P(y|\theta) = \max_\theta \int_x P(y,x|\theta)dx$$

  - EM introduces hidden variable, then marginalizes over it
- What is an alternative ?
  - Treat 'x' has parameter
  - Optimize for a value of 'x'
  - Why is EM preferred ?

EM Optimization

- EM performs iterative optimization
- Each EM iteration comprises 2 steps :

  (1) E step

  (2) M step

EM Optimization

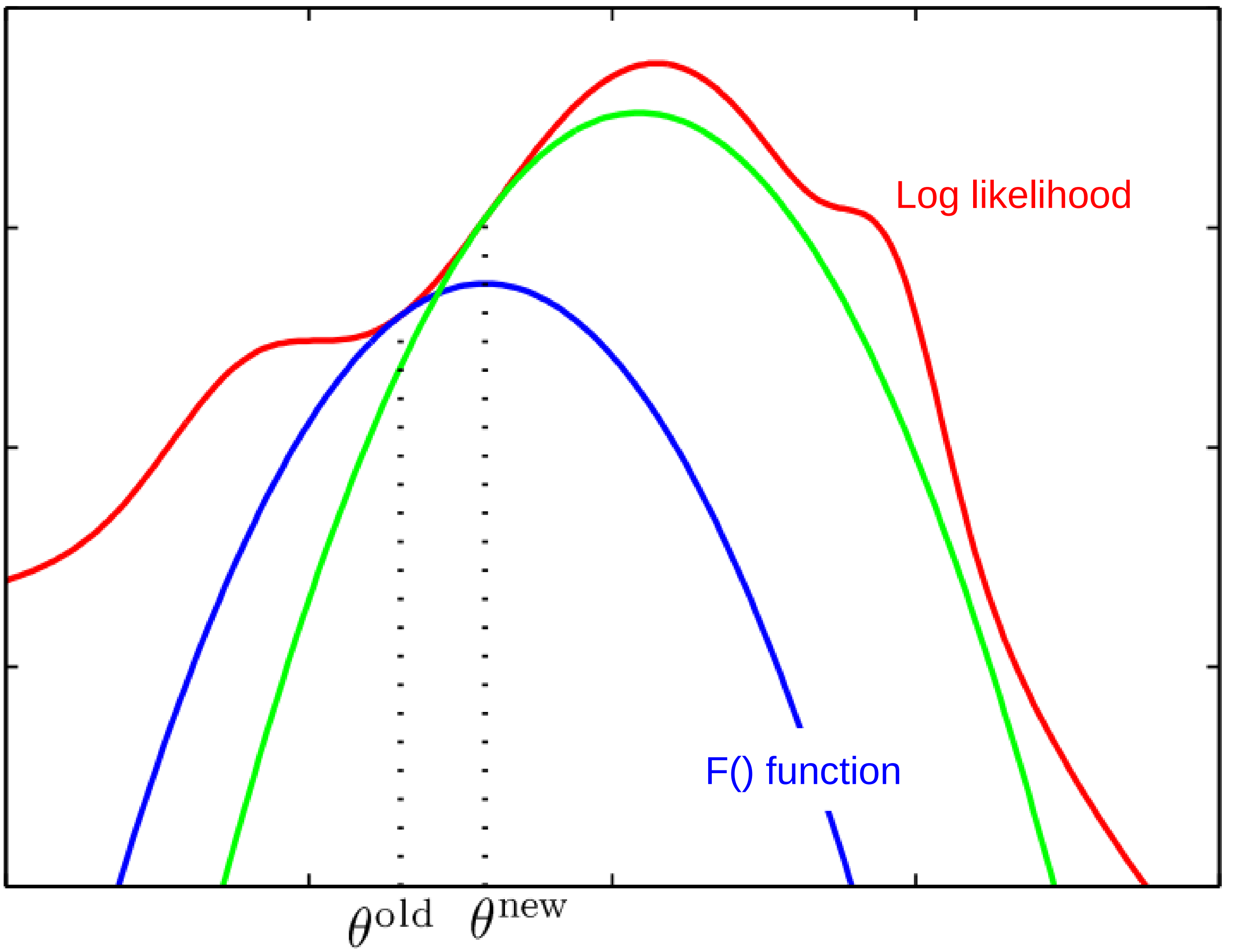
- Assume
  - Iteration i
  - Parameter estimates are θi
- E step
  - Designs a function F(θ; θi) that :

    (1) is a lower bound for (log) likelihood function

    (2) touches likelihood function at current estimate θi

EM Optimization

- EM: E step and M step within 1 iteration



EM Optimization

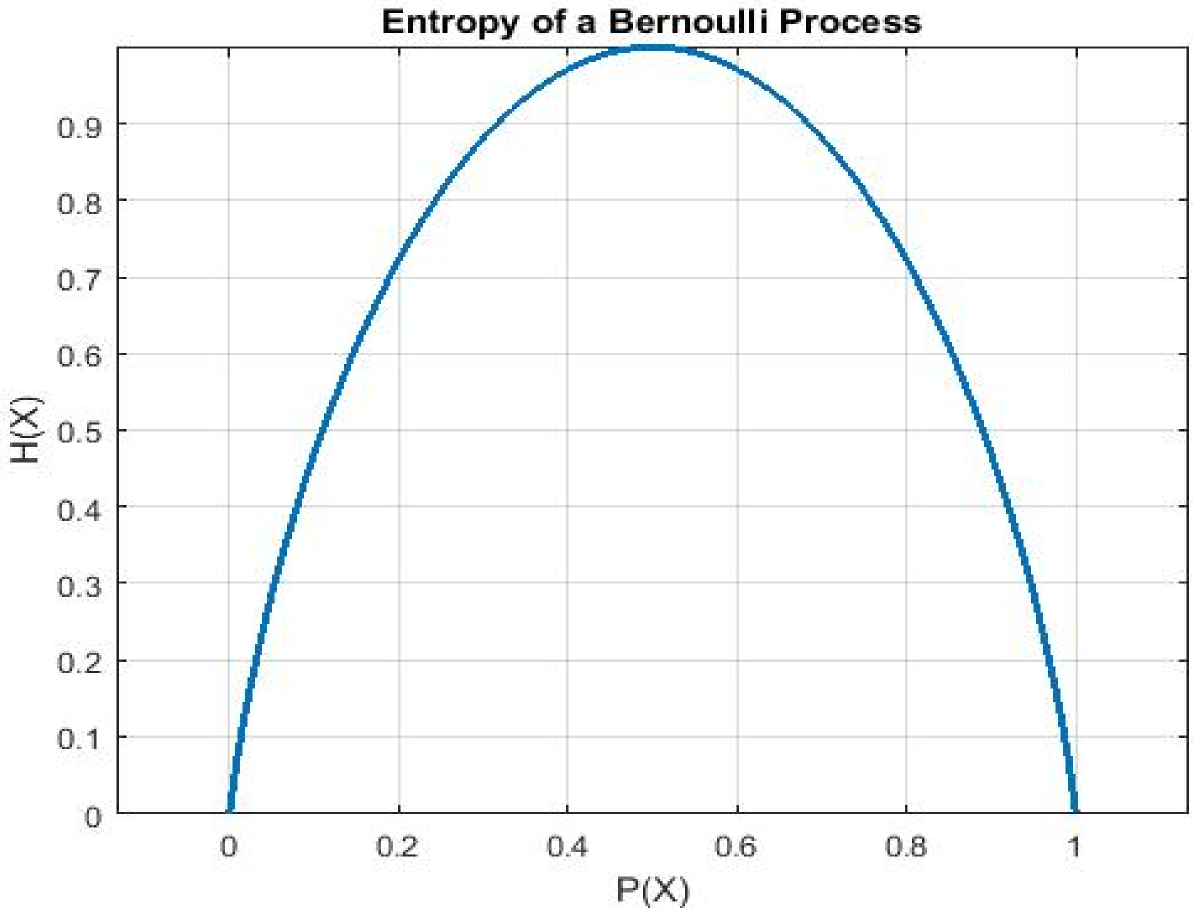
- Assume
  - Iteration i
  - Parameter estimates are θi
- E step
  - Designs a function F(θ; θi) that :

    (1) is a lower bound for (log) likelihood function

    (2) touches likelihood function at current estimate θi
- M step
  - Maximizes F(θ; θi) over θ

Entropy

- Entropy (X) = E [- log P(X)]
  - Shannon
- A measure of:
  - Uncertainty of random variable
  - Spread of PDF



Kullback–Leibler divergence

- Measure of dissimilarity between PDFs / PMFs

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i)\,\ln\frac{P(i)}{Q(i)} \qquad D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x)\,\ln\frac{p(x)}{q(x)}\,\mathrm{d}x$$

  - Expectation of log ratios of PDFs
    - Expectation taken over one of the PDFs
  - Defined only when support of P subset of support of Q
    - q(x) = 0 → p(x) = 0
  - Positive
  - Asymmetric
  - NOT a distance metric

EM Optimization

- Consider the log likelihood = $\log P(y|\theta)$
- Log likelihood can be written as
  - $F(q,\theta) + KL(q \,||\, p(x|y,\theta))$, where
- $F(q,\theta)$ is a:
  - Functional of the PDF **q(x)**
  - Function of parameters θ
  - $F(q,\theta) = \sum_x q(x) \log \{ P(y,x|\theta) / q(x) \}$
- $KL(q \,||\, p(x|y,\theta))$ = KL divergence between PDFs

  $= - \sum_x q(x) \log \{ P(x|y,\theta) / q(x) \}$
  - KL (p1, p2) always >= 0
  - KL (p1, p2) = 0 iff p1 = p2

EM Optimization

- So, $F(q,\theta) = \log P(y|\theta) - KL(q \,||\, p(x|y,\theta))$
- So, $F(q,\theta)$ is a **lower bound** of likelihood $\log P(y|\theta)$
  - For any chosen q(x)
  - $F(q,\theta^*)$ equals $\log P(y|\theta^*)$ iff $q(x)$ = true posterior $p(x|y,\theta^*)$
- We don't know true (ML) parameter values θ^*
- At iteration i, parameter estimates = θi
- **E step** = choose $q(.)$ to maximize $F(q, \theta i)$
  - $q(x) = p(x \mid y, \theta i)$ = posterior PDF of labels given data
  - Thus, at θi, $F(q, \theta i) = \log P(y \mid \theta i)$
  - Thus, F(.) touches log-likelihood function at θi

EM Optimization

- So, $F(q,\theta) = \log P(y|\theta) - KL(q \,||\, p(x|y,\theta))$
- So, $F(q,\theta)$ is a **lower bound** of likelihood $\log P(y|\theta)$
- We don't know true parameter values θ^*
- At iteration i, parameter estimates = θi
- E step fixes $q(x) := p(x \mid y, \theta i)$
- **M step** = choose θ to maximize $F(q, \theta)$
  - Observe (rewrite RHS):

    $F(q, \theta) = E\_q(.) [\log P(x,y|\theta)] + H(q)$, where
  - $E\_q(.) [\log P(x,y|\theta)]$ is our $Q(\theta,\theta i)$ function
  - *H(q)* = entropy of q(.) = *NOT function of θ*

F (q, θ) = \sum_x q(x) log { P (y,x|θ) / q(x) }

EM Optimization

- E step

$$Q(\theta;\theta^i) := E_{P(x|y,\theta^i)}\left[\log P(x,y|\theta)\right]$$

= expectation of the complete-data log likelihood
w.r.t. a posterior PDF of hidden variable x,
where
posterior PDF approximated via current parameter
estimates θi

- M step

$$\theta^{i+1} \leftarrow \arg\max_{\theta} Q(\theta;\theta^i)$$

- Convergence
  - Guaranteed to converge to local maximum of likelihood function P (y|θ)

EM Optimization

- What does the graph of $Q(\theta^{i+1};\theta^{i})$ w.r.t. 'i' look like ?
- Termination criterion ?
  - Relative change between parameter values between 2 consecutive iterations falls below a user-defined threshold
- What if we don't solve for maximum in M step ?
  - What else can we do ?
    - It should converge
    - It should be useful, e.g., optimizes likelihood

GMM Optimization via EM

- Observed random variable: Y := { Yn }   n=1,...,N
  - Leads to data y := { yn } n=1,...,N
- PDF model: $P(x) := \sum_{k=1}^{K} w_k G(x; \mu_k, C_k)$
- Parameters: θ = { wk , μk , Ck }   k=1,...,K
- Hidden random variable: Z := { Zn }   n=1,...,N
  - Zn = zn is label associated with data point yn
  - zn takes values 1,...,K
- Optimization strategy
  - ML estimation of parameters using EM algorithm
  - Constraint: $\sum_k w_k = 1$

GMM Optimization via EM

- Assume, at iteration i, parameter estimates = θ^i
- E step

$$
\begin{aligned}
Q(\theta;\theta^i) &:= E_{P(z|y,\theta^i)}\left[\log P(y,z|\theta)\right] \\
&= E_{P(z|y,\theta^i)}\left[\log \Pi_n P(y_n,z_n|\theta)\right] \quad \text{Independent observations, Independent labels} \\
&= E_{P(z|y,\theta^i)}\left[\log \Pi_n P(y_n|z_n,\theta)P(z_n|\theta)\right] \quad \text{Conditional probability} \\
&= E_{P(z|y,\theta^i)}\left[\sum_n \log\left(P(y_n|z_n,\theta)P(z_n|\theta)\right)\right]
\end{aligned}
$$

GMM Optimization via EM

- E step

$$= E_{P(z|y,\theta^i)}\left[\sum_n \log\left(P(y_n|z_n,\theta)P(z_n|\theta)\right)\right]$$

$$= \sum_n E_{P(z|y,\theta^i)}\left[\log\left(P(y_n|z_n,\theta)P(z_n|\theta)\right)\right]$$ Linearity of expectation

$$= \sum_n E_{P(z_n|y_n,\theta^i)}\left[\log\left(P(y_n|z_n,\theta)P(z_n|\theta)\right)\right]$$ Conditional independence

$$= \sum_n \sum_k P(z_n = k|y_n,\theta^i)\log\left(P(y_n|z_n = k,\theta)P(z_n = k|\theta)\right)$$ Z_n is a discrete RV

GM $\sum_n \sum_k P(z_n = k | y_n, \theta^i) \log \left(P(y_n | z_n = k, \theta) P(z_n = k | \theta) \right)$

- E step
  - Note that $w_k := P(z_n = k|\theta)$
    - Thus, wk is non-negative
  - Denote membership of yn in k-th cluster by :

$$\gamma_{nk} := P(z_n = k | y_n, \theta^i)$$

$$= \frac{P(y_n | z_n = k, \theta^i) P(z_n = k | \theta^i)}{P(y_n | \theta^i)} \quad \text{Bayes rule}$$

$$= \frac{G(y_n | \mu_k^i, C_k^i) w_k^i}{\sum_k G(y_n | \mu_k^i, C_k^i) w_k^i}$$

- Each membership is non-negative
- For each point yn, sum of memberships over K classes is 1

GM

$$\sum_n \sum_k P(z_n = k | y_n, \theta^i) \log \left(P(y_n | z_n = k, \theta) P(z_n = k | \theta) \right)$$

- E step
  - Finally, we have the Q(.) function as :

$$Q(\theta; \theta^i) = \sum_n \sum_k \gamma_{nk} \left(-0.5 \log |C_k| - 0.5 (y_n - \mu_k)' C_k^{-1} (y_n - \mu_k) + \log w_k \right)$$

- Gammas are fixed
- Mahalanobis distance

P C Mahalanobis

- Statistician (1893-1972)
- Founded
  Indian Statistical Institute
  - In 1932
- Sample surveys
  - Using sampling
  - Consumer expenditure,
    tea-drinking habits, public opinion,
    crop acreage, plant disease

P V Sukhatme

- Statistician
- Pioneering work in the 1940s
  by applying random sampling methods
  in agricultural statistics and in biometry
- Made significant contributions in the statistical theory of sampling
  - PhD advisors: J Neyman, ES Pearson
  - ES Pearson ← son of Karl Pearson
  - Karl Pearson founded the world's first university statistics department at University College London in 1911
    - Established the field of mathematical statistics
    - ESP also later became Head of Applied Statistics dept at UCL

Karl Pearson

- Correlation coefficient
- Method of moments
- Chi distance: precursor and special case of Mahalanobis distance
- P value
- Foundations of the statistical hypothesis testing theory and the statistical decision theory
- Pearson's chi-squared test
- PCA
- First introduction of the histogram

GMM Optimization via EM

- M step $Q(\theta;\theta^i) = \sum_n \sum_k \gamma_{nk} \Big(-0.5 \log |C_k| - 0.5(y_n - \mu_k)' C_k^{-1} (y_n - \mu_k) + \log w_k \Big)$
  - Update parameter estimates by solving :

    $$\arg\max_\theta Q(\theta;\theta^i)$$

    under the constraint $\sum_k w_k = 1$

  - Solve for the Gaussian means μk and covariances Ck
    - Take partial derivatives and assign them to 0

GMM Optimization via EM

- M step $Q(\theta;\theta^i) = \sum_n \sum_k \gamma_{nk} \Big(-0.5 \log |C_k| - 0.5 (y_n - \mu_k)' C_k^{-1} (y_n - \mu_k) + \log w_k \Big)$
  - Solve for mean
  - Derivative of a quadratic form
    ∂ (x' A x) = x' (A + A') ∂x, where
    - x' A x is a scalar
    - x' (A + A') is a row vector
    - ∂x is a column vector
    - http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/calculus.html

$$\frac{\partial Q(\theta;\theta^i)}{\partial \mu_k} = 0 = \sum_n \gamma_{nk} C_k^{-1} (y_n - \mu_k)$$

$$\Longrightarrow \quad \mu_k = \frac{\sum_n \gamma_{nk} y_n}{\sum_n \gamma_{nk}}$$

GMM Optimization via EM

- M step $Q(\theta;\theta^i) = \sum_n \sum_k \gamma_{nk}\Big(-0.5\log|C_k| - 0.5(y_n-\mu_k)' C_k^{-1}(y_n-\mu_k) + \log w_k \Big)$
  - Solve for covariance
  - Derivative of a determinant
    ∂ |A| = |A| (A−T) :' ∂A
    - where “ : ” corresponds to vectorization
  - Derivative of an inverse
    ∂ (x' A−1 y) = −(A−T xy' A−T) :' ∂A

$$\frac{\partial Q(\theta;\theta^i)}{\partial C_k} = 0 = -\sum_n \gamma_{nk} C_k^{-T} + \sum_n \gamma_{nk} C_k^{-T}(y_n-\mu_k)(y_n-\mu_k)' C_k^{-T}$$

$$\Longrightarrow\ C_k = \frac{\sum_n \gamma_{nk}(y_n-\mu_k)(y_n-\mu_k)'}{\sum_n \gamma_{nk}}$$

GMM Optimization via EM

- M step $Q(\theta;\theta^i) = \sum_n \sum_k \gamma_{nk}\Big(-0.5\log|C_k| - 0.5(y_n-\mu_k)' C_k^{-1}(y_n-\mu_k) + \log w_k \Big)$
  - Solve for weights wk
  - Constrained optimization problem

$$\arg\max_{\{w_k\}} Q(\theta;\theta^i) \text{ such that } \sum_k w_k = 1$$

$$\arg\max_{\{w_k\}} \sum_n \sum_k \gamma_{nk} \log w_k \text{ such that } \sum_k w_k = 1$$

  - Use method of Lagrange multipliers

GMM Optimization via EM

- M step
  - Solve for weights wk
  - Constrained optimization problem
    - Lagrangian is :

$$L(\{w_k\}) := \sum_n \sum_k \gamma_{nk} \log w_k + \lambda \left(\sum_k w_k - 1 \right)$$

$$= \sum_k \log w_k \gamma_k + \lambda \left(\sum_k w_k - 1 \right)$$

where $\gamma_k := \sum_n \gamma_{nk}$ is the **total membership** associated with the k-th Gaussian

GMM Optimization via EM

- M step
  - Solve for weights wk
  - Constrained optimization problem

$$\frac{\partial L(\{w_k\})}{\partial w_k} = 0 = \frac{\gamma_k}{w_k} + \lambda \implies w_k = \frac{-\gamma_k}{\lambda}$$

$$\frac{\partial L(\{w_k\})}{\partial \lambda} = 0 = \sum_k w_k - 1$$

Thus,

$\lambda = -\sum_k \gamma_k$ and $w_k = \frac{\gamma_k}{\sum_k \gamma_k} = \frac{\gamma_k}{N}$

where we note that $\sum_k \gamma_k = \sum_k \sum_n \gamma_{nk} = \sum_n \left(\sum_k \gamma_{nk} \right) = \sum_n (1) = N$

GMM-EM Segmentation: Limitation

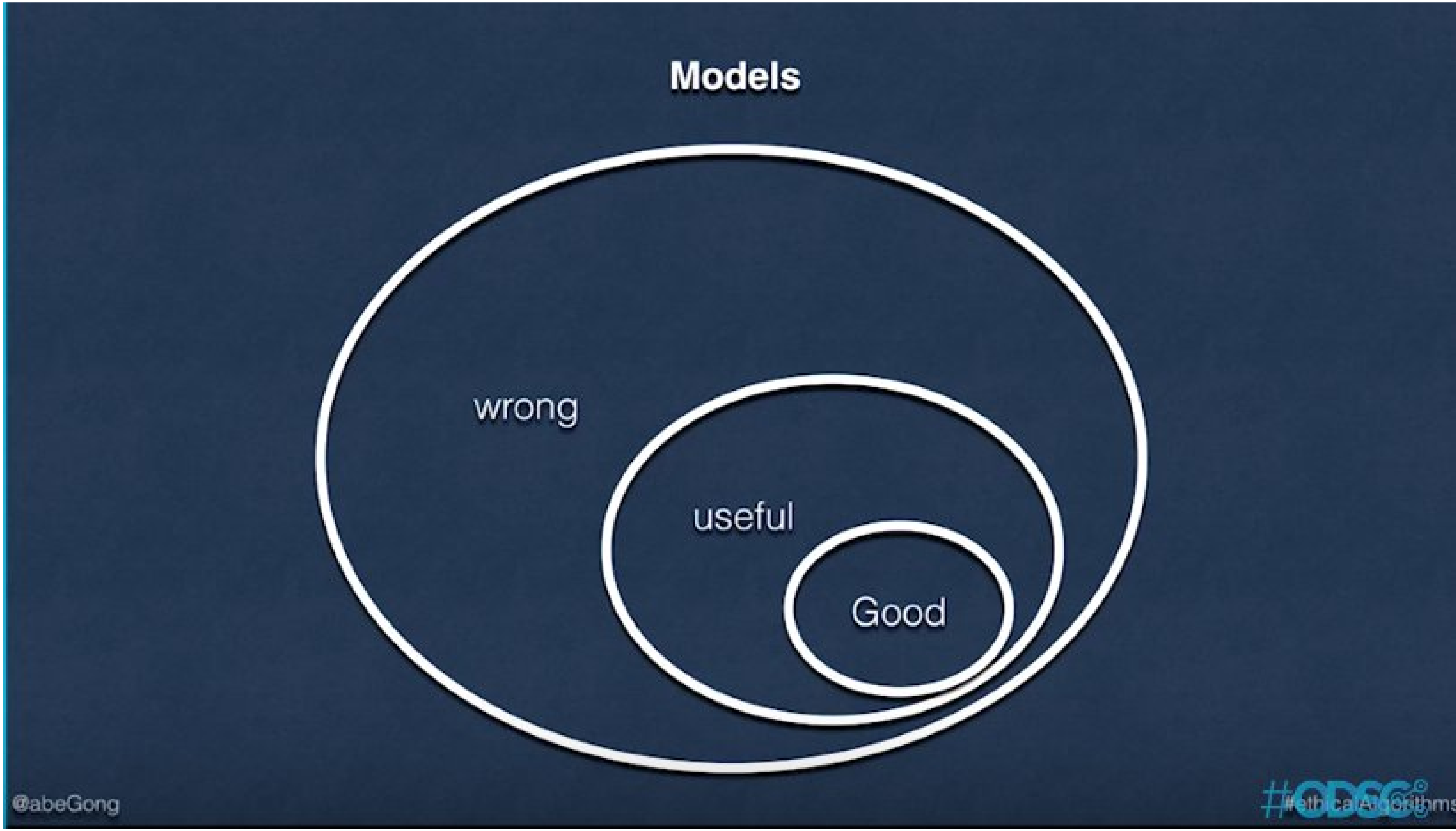
- Doesn't enforce spatial smoothness on segmentation
  - How to do that ?
    - MRF model on label images

MODELS:

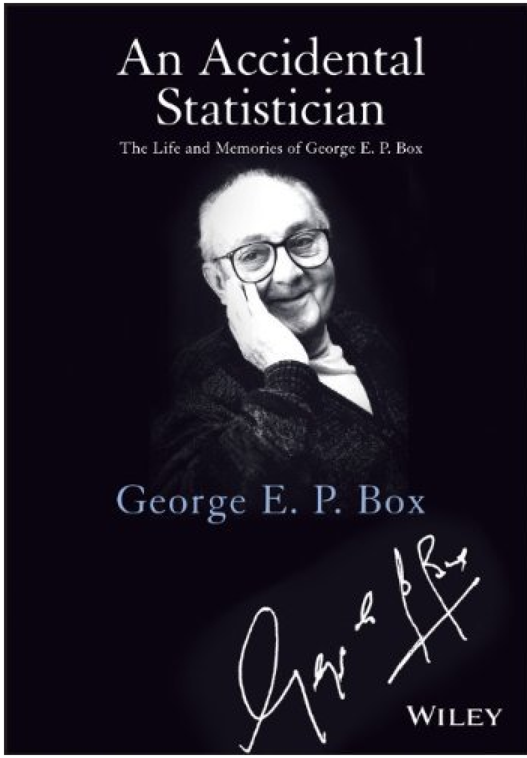
"Essentially, all models are wrong, but some are useful."

"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful."

George E.P. Box


Models
wrong
useful
Good
@abeGong

George E. P. Box

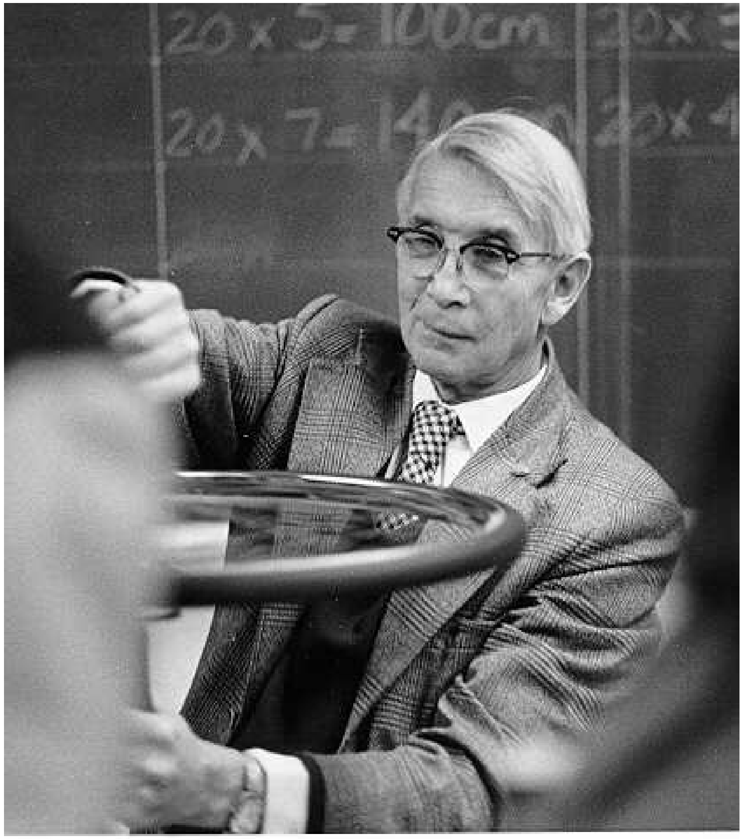


- Statistician (1919 - 2013)
  - “One of the great statistical minds of the 20th century”
  - Quality control, time-series analysis, design of experiments, Bayesian inference
- Advisor:
  ES Pearson, son of Karl Pearson
- During World War II, performed experiments for the British Army exposing small animals to poison gas. To analyze the results of his experiments, he taught himself statistics from available texts
- After war, got PhD from UCL

Priors on Label Images

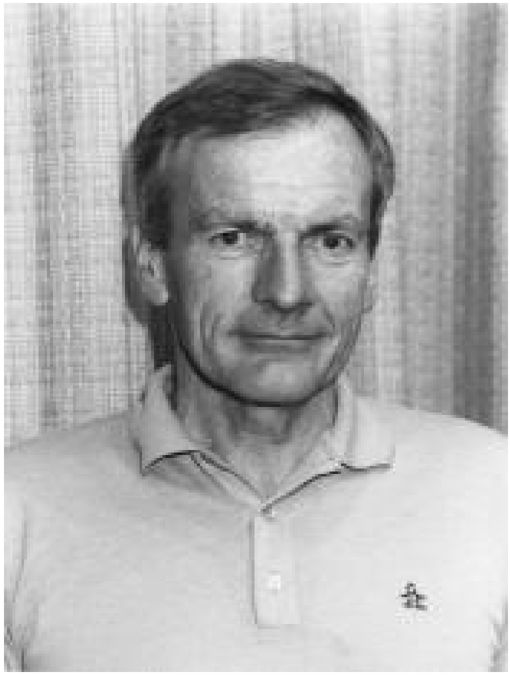
- **MRF on Labels**
  - Binary labels
    - Ising model
  - Labels taking multiple values
    - Potts model
  - MRF where potential function on 2-cliques
    - Gives a constant penalty if (neighboring) labels are unequal
    - Otherwise, gives a zero penalty
    - Does a quadratic error function make sense ?

Ernst Ising



- Physicist (1900 – 1998)
- Advisor: Wilhelm Lenz
  - Studied ferromagnetism with Wolfgang Pauli (Nobel Laureate)
- Finished PhD in 1924
  - PhD work = Ising model
  - Perhaps, start of probabilistic graphical models
- Life
  - Forbidden to teach / research in Hitler's regime (1933+)
    - For some time, worked as a shepherd, railroad worker
  - Immigrated to US in 1947, never published later
  - Realized success of his PhD work in 1949

Renfrey Potts



- Mathematician (1925 – 2005)
- Proposed Potts model in his PhD (1951)
- Switched fields to Operations Research
- Potts model got recognition 20 years after it was first published

EM - 1977

- Arthur P. Dempster (1929 -)
  - Joined Harvard statistics in 1957 (4 faculty in dept.)
  - Advisor: John Tukey (FFT fame)
    - Founding member of Princeton's statistics department
- NM Laird (1943 -) ← PhD student of Dempster
- DB Rubin (1943 -)
  - Worked for ETS
  - In Harvard psychology grad school.
    Asked to take intro stats courses ← insufficient backgrnd.
    Felt insulted, given background in physics @ Princeton.
    Transferred to applied math → PhD in stats @ Harvard.
  - 271K+ citations
  - Rubin ← Cochran ← Wishart ← Karl Pearson

Bayesian Image Segmentation

- **MAP-MRF Segmentation (Hard Seg.)**
  - **Hidden MRF + GMM**
  - Consider **label MRF X** = x, with labels k ∈ 1, 2, ⋯, K
  - Consider observed image **data Y** = y
  - Generative model for intensities given class label : P (Y | X) := Πi P (Yi | Xi)
    - e.g., for each class k , assume that the model is a Gaussian P (Yi | Xi = k) = G (Yi | μk , σk)
  - Let θ = parameters underlying noise model and MRF model (in general)
    - No weight parameters wk
    - Why not ? What is a 'procedure' of generating data ?

Bayesian Image Segmentation

- MAP-MRF Segmentation (Hard Seg.)
  - Optimization problem
    - Obtain segmentation x and parameter estimates θ by optimizing $\max_{x,\theta} P(x \mid y, \theta)$
  - Optimization strategy
    - Alternate between :

      (i) finding optimal parameters $\max_{\theta} P(x \mid y, \theta)$ and

      (ii) finding optimal segmentation $\max_{x} P(x \mid y, \theta)$

Bayesian Image Segmentation

- MAP-MRF Segmentation (Hard Seg.)
  - Optimal parameters
    - Assume MRF model parameters to be fixed
    - Given hard segmentation x,
      if noise model is Gaussian,
      then,
      optimal parameters for each class = ?
      - Sample mean & variance (over pixels labeled to that class)

Bayesian Image Segmentation

- MAP-MRF Segmentation via ICM (Hard Seg.)
  - MAP segmentation
    - $\max_x P(x \mid y, \theta)$
  - Rewrite $P(X \mid y, \theta) =$

$$= P(X_i, X_{\sim i} | y, \theta)$$

$$= P(X_i | X_{\sim i}, y, \theta) P(X_{\sim i} | y, \theta) \text{ Conditional probability}$$

$$= P(X_i | X_{N_i}, y, \theta) P(X_{\sim i} | y, \theta) \text{ Markov assumption on } X$$

$$= P(X_i | X_{N_i}, y_i, \theta) P(X_{\sim i} | y, \theta) \text{ Conditional independence in noise model}$$

Bayesian Image Segmentation

- MAP-MRF Segmentation via ICM (Hard Seg.)
  - At each voxel i, perform :

$$\max_{x_i} P(X|y,\theta) = \max_{x_i} P(X_i|X_{N_i},y_i,\theta)P(X_{\sim i}|y,\theta)$$

$$= \max_{x_i} P(X_i|X_{N_i},y_i,\theta) \text{ Second term doesn't depend on } x_i$$

$$= \max_{x_i} P(y_i|X_i,X_{N_i},\theta)P(X_i|X_{N_i},\theta)/P(y_i|X_{N_i},\theta) \text{ Bayes Rule}$$

$$= \max_{x_i} P(y_i|X_i,X_{N_i},\theta)P(X_i|X_{N_i},\theta) \text{ Denominator doesn't depend on } x_i$$

$$= \max_{x_i} P(y_i|X_i,\theta)P(X_i|X_{N_i},\theta) \text{ Conditional independence assumption in}$$

  - P (yi | Xi,θ) = label likelihood
  - P (Xi | X_{Ni},θ) = label prior
  - How to perform this maximization ?

Bayesian Image Segmentation

- MAP-MRF Segmentation (Hard Seg.)
  - Order of Label Updates
    - Sequentially: Column by column, and then row by row
      - Can lead to artifacts
    - Sequentially: Randomized order each iteration
      - Need to generate random sequence each iteration
      - Are artifacts eliminated ?
    - In Parallel: Doesn’t guarantee increase in image probability
      - No need to generate random sequence
      - Are artifacts eliminated ?
    - In Parallel: Update only if image probability increases !!
      - Typically, it will !
      - May terminate too soon !

Bayesian Image Segmentation

- **MAP-MRF Segmentation (Soft Seg.)**
  - **Hidden MRF + GMM + EM**
    - https://www.ncbi.nlm.nih.gov/pubmed/11293691

IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 20, NO. 1, JANUARY 2001

Segmentation of Brain MR Images Through a Hidden Markov Random Field Model and the Expectation-Maximization Algorithm

Yongyue Zhang*, Michael Brady, and Stephen Smith

Bayesian Ir $\sum_n \sum_k P(z_n = k | y_n, \theta^i) \log \left(P(y_n | z_n = k, \theta) P(z_n = k | \theta) \right)$

- **Hidden MRF + GMM + EM (Soft Seg)**
  - Number of voxels N
  - Observed Data Variable = Y = { Yi }   n=1,...,N
  - Label MRF (Hidden) Variable = X = { Xi }   n=1,...,N
  - Number of classes = L
  - Parameters θ = { μl , σl }   l=1,...,L
    - No weight parameters
    - Weight parameters wk → priors on number of points derived from class k, while computing memberships
    - We replace by MRF prior
  - For voxel i, neighbors are denoted by Ni

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
  - Optimization Strategy
  - Find ML estimates for parameters
    $\max_\theta P(y|\theta)$
  - Using EM

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
  - E step

$$
\begin{aligned}
Q(\theta|\theta^t) &:= E_{P(X|y,\theta^t)}[\log P(X, y|\theta)] \\
&= E_{P(X|y,\theta^t)}[\log(P(y|X,\theta)P(X))] \text{ Conditional Probability} \\
&= E_{P(X|y,\theta^t)}[\log P(X) + \log \Pi_i P(y_i|X_i,\theta)] \text{ i.i.d. noise model} \\
&= E_{P(X|y,\theta^t)}[\log P(X) + \sum_i \log P(y_i|X_i,\theta)]
\end{aligned}
$$

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
  - E step

log

$$= E_{P(X|y,\theta^t)}[P(X)] + E_{P(X|y,\theta^t)}[\sum_i \log P(y_i|X_i,\theta)]$$ linearity of expectation

log

$$= E_{P(X|y,\theta^t)}[P(X)] + \sum_i E_{P(X|y,\theta^t)}[\log P(y_i|X_i,\theta)]$$ linearity of expectation

log

$$= E_{P(X|y,\theta^t)}[P(X)] + \sum_i E_{P(X_i,X_{\sim i}|y,\theta^t)}[\log P(y_i|X_i,\theta)]$$

- We can ignore the first term
  - Why ?
  - What is the purpose of the Q(.) function ?
- E step requires integration w.r.t. posterior prob on X
  - How to handle that ?

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
  - E step
    - Approximate expectation :

$$E_{P(x_i, x_{\sim i}|y, \theta^t)}[\log P(y_i|x_i, \theta)] \approx E_{P(x_i|x_{\sim i}, y, \theta^t)}[\log P(y_i|x_i, \theta)]$$

    - Left hand side
      - Sum over possible labels at i-th voxel,
        accounting for all possibilities of neighbor labels (& their neighbors, & ...)
    - Right hand side
      - Sum over possible labels at i-th voxel, keeping rest of image fixed !
      - This somewhat underestimates sampling variability !
        This also introduces some bias !
    - Label image X fixed to what ? X is hidden !
      - Fix to MAP segmentation, given current parameters. Works for brain seg.

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
  - E step

Calling $C := E_{P(x|y,\theta^t)}\overset{\text{log}}{[P(x)]}$ as a constant independent of θ, we get

$$
\begin{aligned}
Q(\theta|\theta^t) - C &\approx \sum_i E_{P(x_i|\overset{\text{MAP}}{x}_{\sim i},y,\theta^t)}[\log P(y_i|x_i,\theta)] \\
&= \sum_i E_{P(x_i|\overset{\text{MAP}}{x}_{N_i},y,\theta^t)}[\log P(y_i|x_i,\theta)] \\
&= \sum_i \sum_{l=1}^{L} P(x_i = l|\overset{\text{MAP}}{x}_{N_i}, y, \theta^t) \log P(y_i|x_i = l, \theta)
\end{aligned}
$$

  - Below is what we had for GMM + EM (without MRF):

$$
\sum_n \sum_k P(z_n = k|y_n, \theta^i) \log \Big(P(y_n|z_n = k, \theta) P(z_n = k|\theta) \Big)
$$

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
  - E step
    - Lets see how the MRF prior comes in
  - M step
    - Remains very similar to that of GMM fitting via EM (without MRF)
      - What changes ? Only memberships changes. Lets see how.

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
  - E step
  - **Memberships =** $P(x_i = l | \overset{\text{MAP}}{x}_{N_i}, y, \theta^t) =$

$$= P(x_i = l | y_i, \overset{\text{MAP}}{x}_{N_i}, \theta^t) \text{ Conditional Independence of } X_i, Y_{\sim i}, \text{ Given } Y_i, \overset{\text{MAP}}{X}_{N_i}$$

$$= \frac{P(y_i | x_i = l, \overset{\text{MAP}}{x}_{N_i}, \theta^t) P(x_i = l | \overset{\text{MAP}}{x}_{N_i}, \theta^t)}{P(y_i | x_{N_i}, \theta^t)} \text{ Bayes Rule}$$

$$= \frac{G(y_i | \mu_l, \sigma_l) P(x_i = l | \overset{\text{MAP}}{x}_{N_i})}{\sum_{l=1}^{L} G(y_i | \mu_l, \sigma_l) P(x_i = l | \overset{\text{MAP}}{x}_{N_i})} \text{ Conditional Independence of } Y_i, X_{N_i}, \text{ Given } X_i$$

    - What is the prior here ? What is its effect ?
  - Below is what we had for GMM + EM (without MRF):

$$\begin{aligned}
\gamma_{nk} &:= P(z_n = k | y_n, \theta^i) \\
&= \frac{P(y_n | z_n = k, \theta^i) P(z_n = k | \theta^i)}{P(y_n | \theta^i)} \quad \text{Bayes rule} \\
&= \frac{G(y_n | \mu_k^i, C_k^i) w_k^i}{\sum_k G(y_n | \mu_k^i, C_k^i) w_k^i}
\end{aligned}$$

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
  - E step
  - Conditional label probability is:

$$P(X_i|X_{S-\{i\}}) = \frac{\exp\left(-\sum_{a\in A_i} V_a(X_a)\right)}{\sum_{x_i'} \exp\left(-\sum_{a\in A_i} V_a(X_a)\right)} = \frac{\exp\left(-\sum_{a\in A_i} V_a(X_a)\right)}{Z_i}$$

  - What is this probability in case all the following hold:
    - 2 classes + 4 neighborhood
    - 3 neighbors are l1, 1 neighbor is l2
    - V (L1, L2) = beta when L1 ~= L2, where parameter beta > 0
      V (L1, L2) = 0 otherwise

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
  - M step
    - If we call memberships → gamma_{nk}
    - Update for mean and (co)variance

$$\mu_k = \frac{\sum_n \gamma_{nk} y_n}{\sum_n \gamma_{nk}} \qquad C_k = \frac{\sum_n \gamma_{nk}(y_n - \mu_k)(y_n - \mu_k)'}{\sum_n \gamma_{nk}}$$

    - How will you make segmentation smoother ?
      - What parameter will you modify ?
    - What happens if the MRF prior is removed ?

Bayesian Image Segmentation

- Hidden MRF + GMM + EM (Soft Seg)
- Algorithm
  - Initialize parameters: means, covariances
  - **E step (approximation)**
    1) Compute MAP label image, given parameters
       - How to compute this ?

    2) Evaluate memberships
       - Spatially smooth
  - **M Step**
    3) Update means and covariances, for all classes
  - Repeat E and M steps, until convergence
  - **Output:** Memberships (soft, spatially smooth)

Bayesian Image Segmentation

- How to get initial segmentation ?
  - e.g., for brain tissue segmentation

Validation

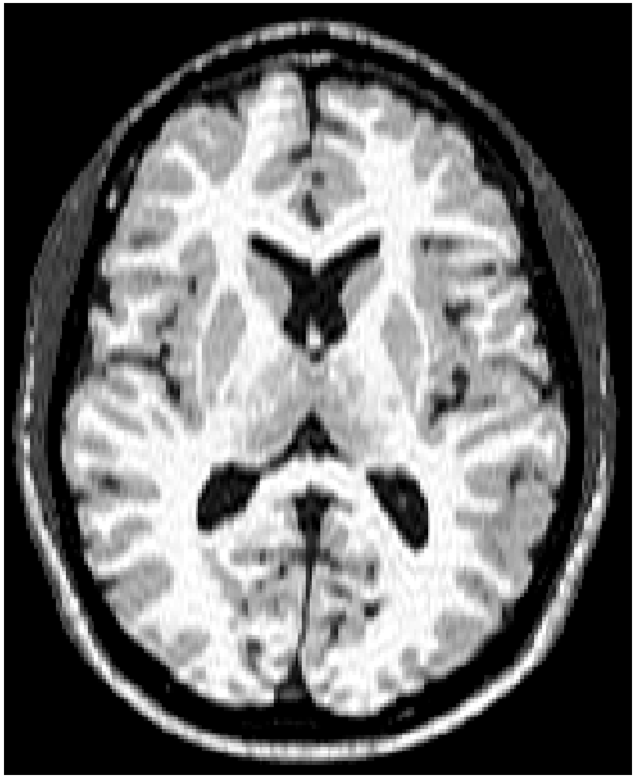
- How to validate an algorithm for real-world applications ?
  (A) real-world phantoms
  (B) human clinical data
  - What about segmentation ?
    - What is “ground truth” ?
    - Kinds of variability
  - What about denoising, reconstruction ?
  - What about segmentation ?

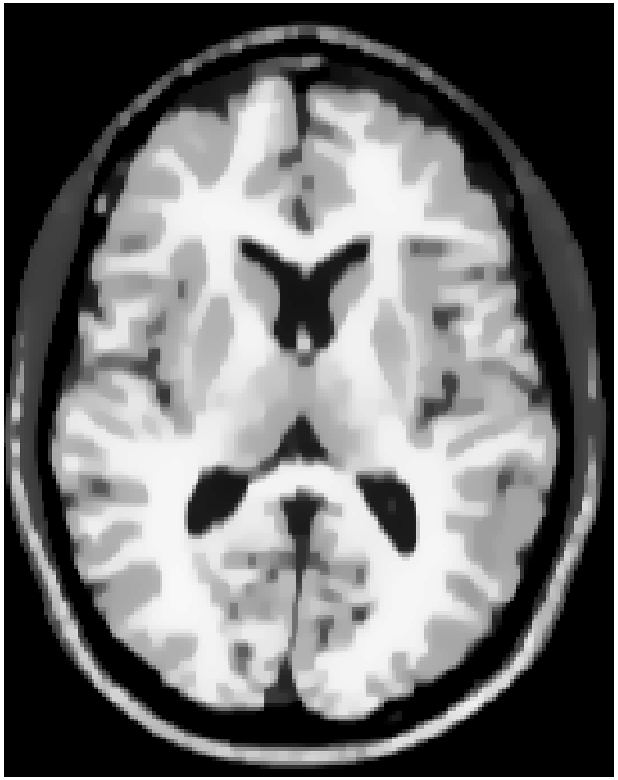
Validation

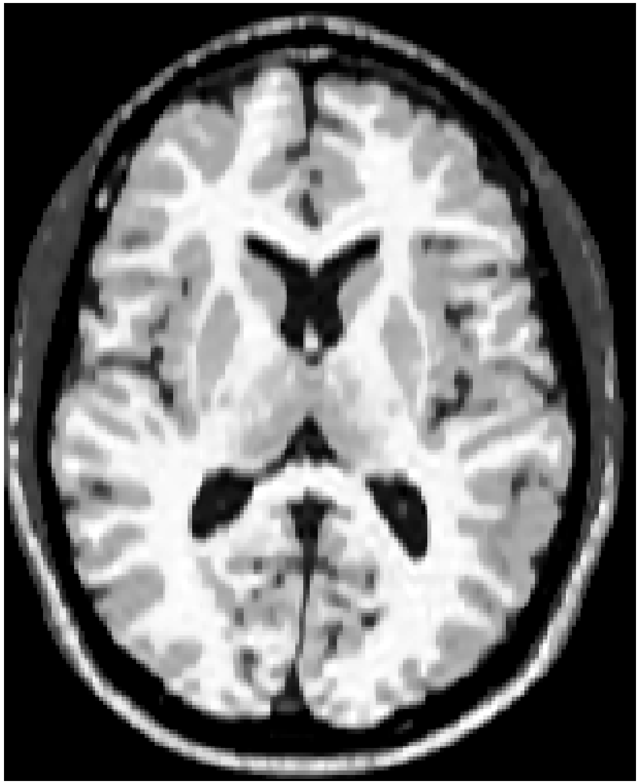
- Examples of denoised images using different values of algorithm parameters

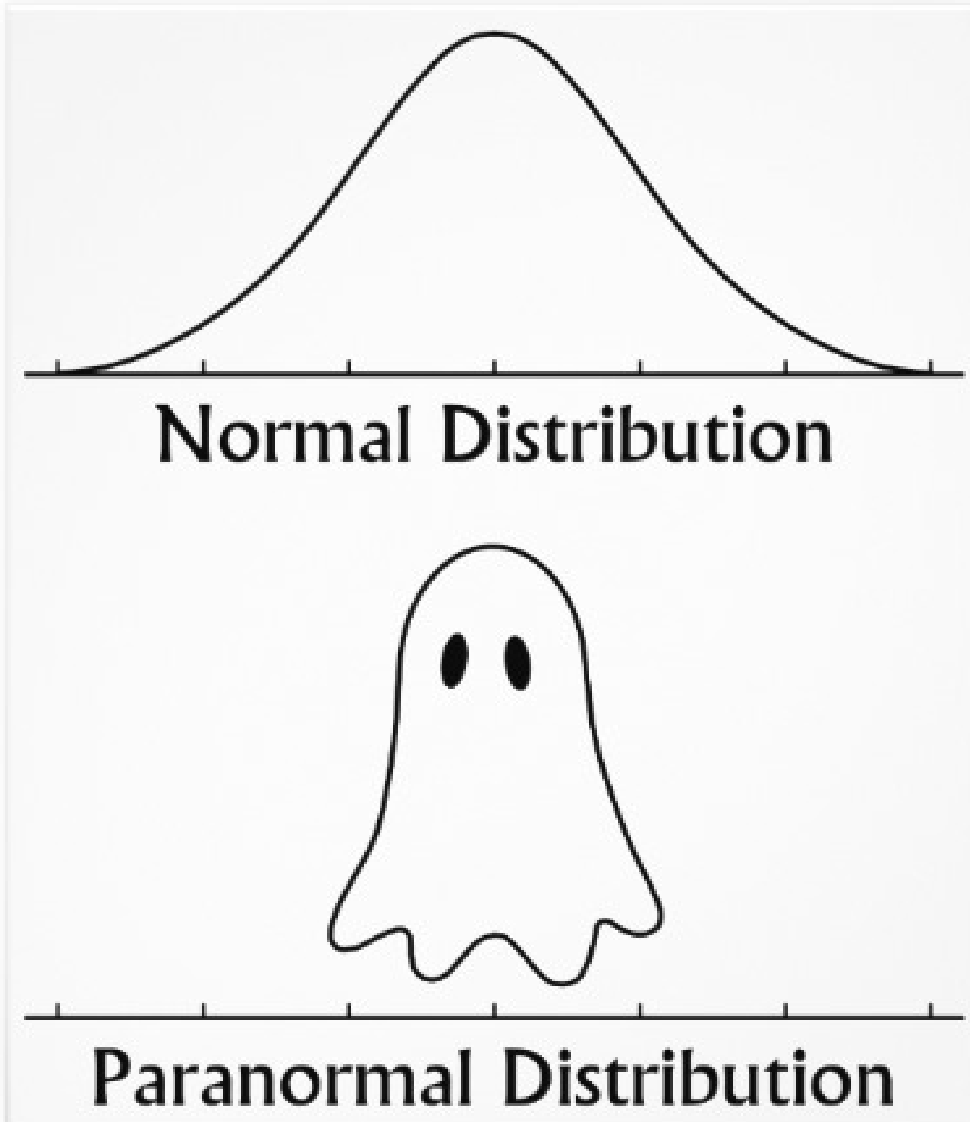
Validation

- Examples of reconstructed images using different values of algorithm parameters








Normal Distribution
Paranormal Distribution

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Number of voxels N
  - Observed Data Variable = Y = { Yi }   n=1,...,N
  - Label MRF Variable = X = { Xi }   i=1,...,N
  - Number of classes = 2
    - Let label values xi be binary, i.e., 0 or 1
  - Generative model for intensities given class label
    $P(Y \mid X) := \Pi_i P(Y_i \mid X_i)$
  - Let θ = parameters underlying noise model
    - Assume MRF model parameters to be fixed

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Optimization Problem
    - Obtain segmentation x and parameter estimates θ by maximizing posterior
  - Optimization Strategy
    - Alternate between
      (i) finding optimal parameters $\max_\theta P(x \mid y, \theta)$ and
      (ii) finding optimal segmentation $\max_x P(x \mid y, \theta)$
  - Optimizing parameters is easy
    - If noise model is Gaussian, given the segmentation, optimal parameters for each class are
      sample mean and sample variance over voxels in that class

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Optimal segmentation, given parameters

$$\max_x P(x|y,\theta)$$

$$= \max_x \log P(x|y,\theta)$$

$$= \max_x (\log P(y|x,\theta) + \log P(x))$$

$$= \max_x \left(\log \Pi_i P(y_i|x_i,\theta) + \log \frac{\exp(0.5 \sum_{(i,j)\in\mathcal{N}} \beta_{ij} V(x_i,x_j))}{Z} \right)$$ Consider only cliques c

where

– Z is a constant that depends only on the MRF parameters that are fixed/known

– $\beta_{ij} \geq 0$: non-negative (convention; relative to how the potential function $V()$ is defined)

– $\beta_{ij} := \beta_{ji}$: neighbor interactions are symmetric

– $\beta_{ii} := 0$: interaction with self isn't allowed

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Assume that potential function $V(\cdot,\cdot)$ is defined, for the case of binary label values, as $V(a, b) := ab + (1 - a)(1 - b)$
    - When neighbor labels are same :
      - $a=b=0$ or $a=b=1$, then $V(a,b) = 1$
      - Leads to higher-prob state because $\beta_{ij} > 0$
    - When neighbor labels are different :
      - $a=1-b=0$ or $a=1-b=1$, then $V(a,b) = 0$
      - Leads to lower-prob state because $\beta_{ij} > 0$

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Rewrite the likelihood function

$$\Pi_i P(y_i|x_i,\theta) = \Pi_i P(y_i|x_i=1,\theta)^{x_i} P(y_i|x_i=0,\theta)^{1-x_i}$$

  - Simplifying the objective function (log Posterior)

$$\max_x P(x|y,\theta) = \max_x \left(\sum_i x_i \log P(y_i|x_i=1,\theta) + (1-x_i)\log P(y_i|x_i=0,\theta) \right.$$
$$\left. +0.5 \sum_i \sum_j \beta_{ij}(x_i x_j + (1-x_i)(1-x_j)) \right)$$

    - Factor 0.5 ensures that pairs aren’t considered twice

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Simplifying the objective function (log Posterior)

$$\max_x P(x|y,\theta) = \max_x \left(\sum_i \lambda_i x_i + 0.5 \sum_i \sum_j \beta_{ij} (2 x_i x_j - x_i - x_j) \right)$$

where

$$\lambda_i := \log P(y_i|x_i = 1, \theta) - \log P(y_i|x_i = 0, \theta) = \log \frac{P(y_i|x_i=1,\theta)}{P(y_i|x_i=0,\theta)}$$

= log of likelihood ratio
(which is a constant independent of the segmentation)

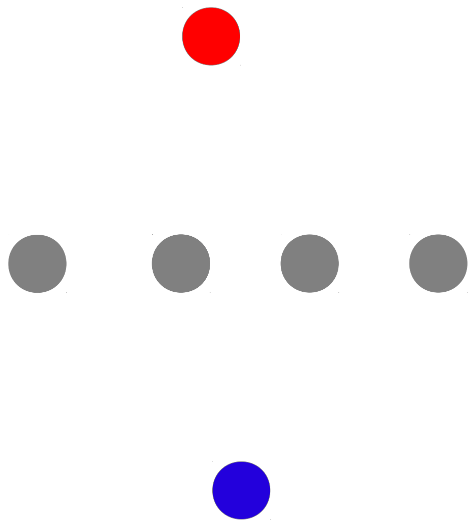
Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Construct an s-t graph as follows :

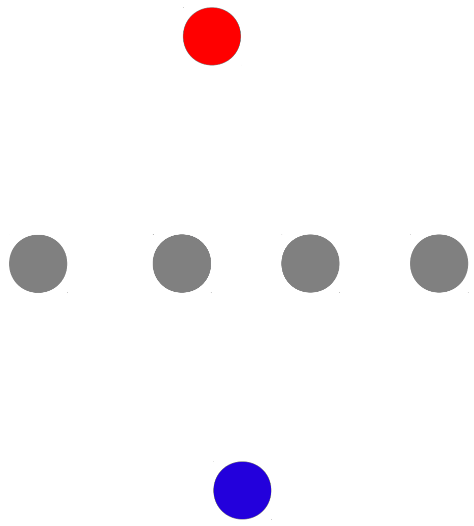
    (1) Add a vertex for each voxel **i**

    (2) Add two additional vertices called **s** and **t**

Example 1



Example 2



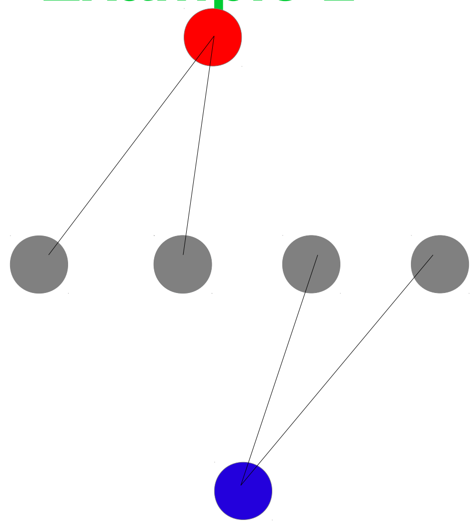
Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Construct an s-t graph as follows :

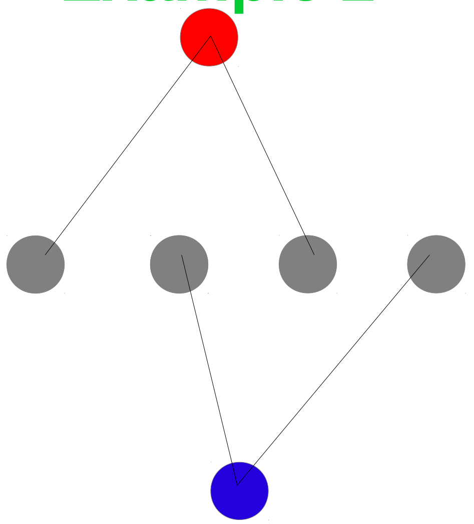
    (3) For each voxel i, if $\lambda_i > 0$, then add an edge from vertex s to vertex i with **cost** $c_{si} := \lambda_i \quad > 0$

    (4) For each voxel i, if $\lambda_i \leq 0$, then add an edge from vertex i to vertex t with **cost** $c_{it} := -\lambda_i \quad > 0$

Example 1



Example 2

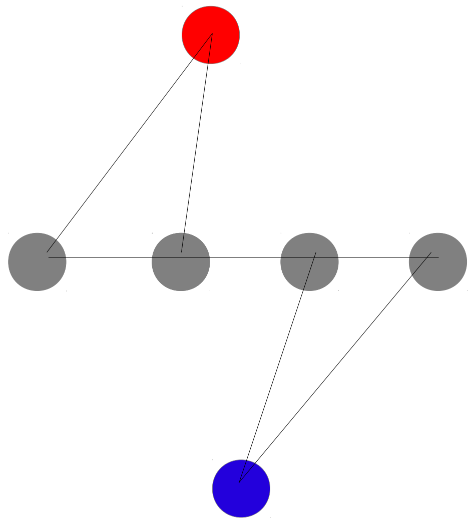


Bayesian Image Segmentation

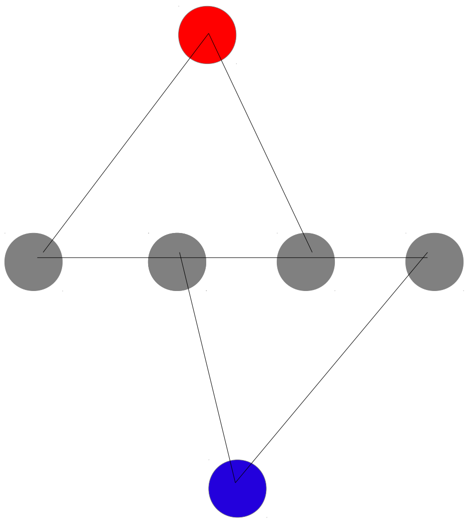
- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Construct an s-t graph as follows :

    (5) For every pair of neighboring voxels i, j in image, add an edge with cost cij := βij > 0

Example 1



Example 2

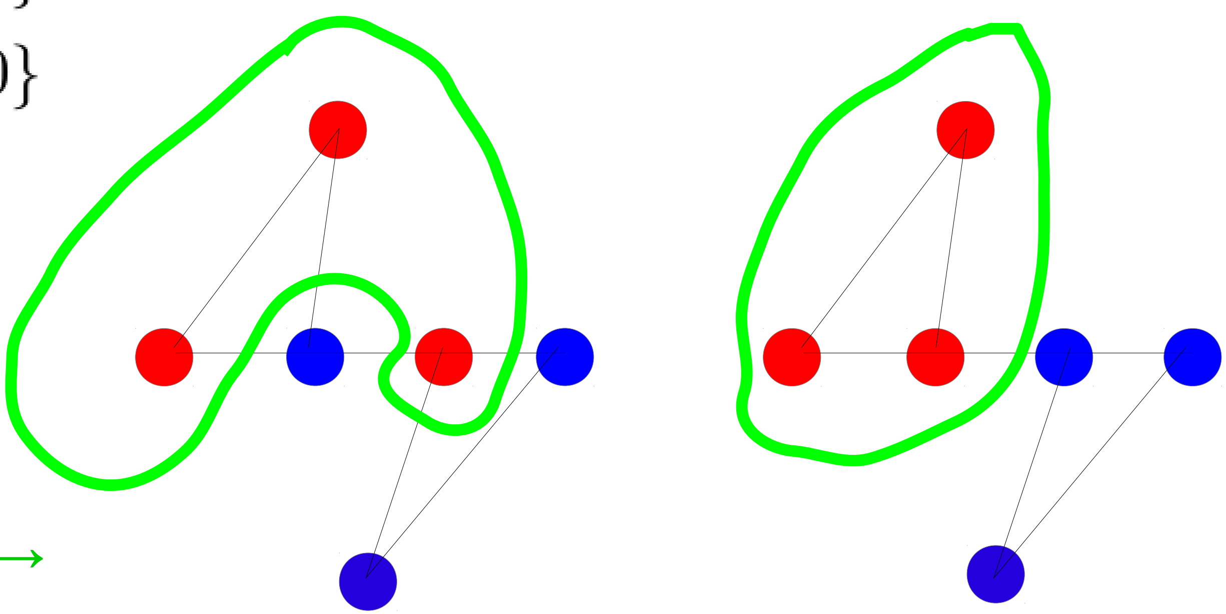


Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - A **cut** of the s-t graph is a partition of the graph into 2 mutually-exclusive & exhaustive sets of vertices s.t.
    - one of the sets contains “s” (& all vertices in set have label 1)
    - the other set contains “t” (& all vertices in set have label 0)

$$S := \{s\} \cup \{i : x_i = 1\}$$
$$T := \{t\} \cup \{i : x_i = 0\}$$

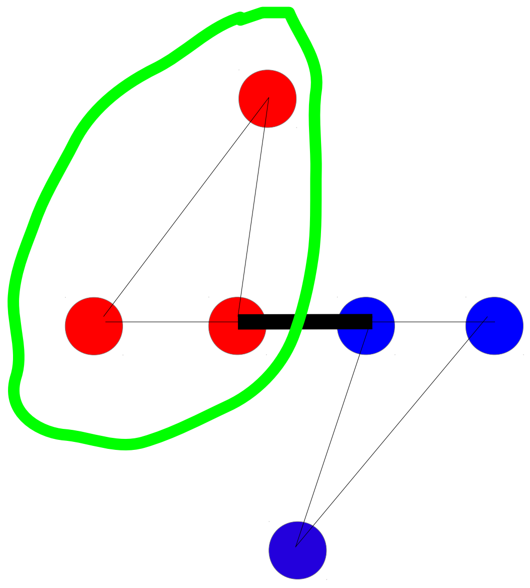
  - *Cut separates vertex s from vertex t*
  - 2 example cuts for graph in Example 1 →

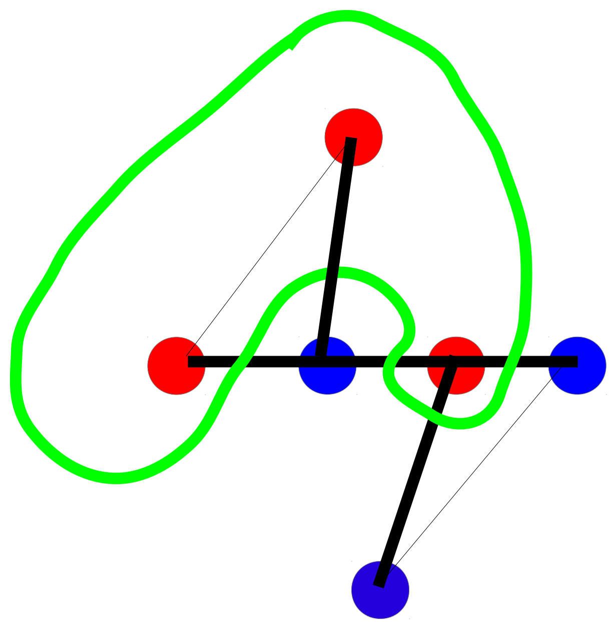


Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - **Capacity** of the cut = ?
  - Capacity of the cut (S,T)
    = sum of costs of each edge with
    one of the vertices in S and
    the other vertex in T

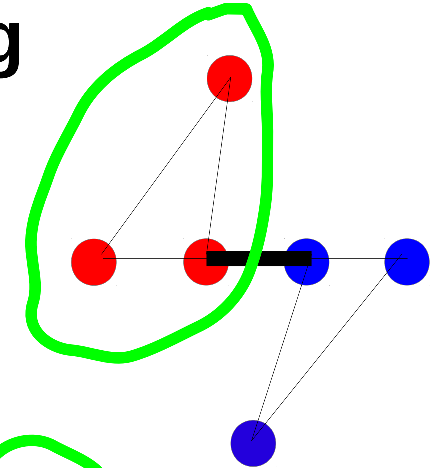
$$C(S,T) := \sum_{u \in S, v \in T} c_{uv}$$

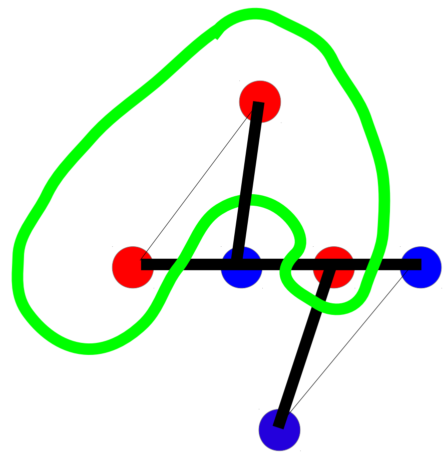




Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - **Claim : Min s-t Cut equals MAP labeling**
  - Is this true ? Lets analyze.
  - For some cut (S,T):
    - Or equivalently, for some labeling x:





$$C(S,T) := \sum_{u \in S, v \in T} c_{uv}$$

$$= \sum_{i \in T} c_{si} \text{ Edges between vertex } s \text{ and vertices in set } T$$

$$+ \sum_{i \in S} c_{it} \text{ Edges between vertices in set } S \text{ and vertex } t$$

$$+ \sum_{i \in S - \{s\}, j \in T - \{t\}} c_{ij} \text{ Edges between vertices in set } S - \{s\} \text{ and vertices in set } T - \{t\}$$

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Claim : Min s-t Cut equals MAP labeling
  - (1) Edges s−i between “s” and vertices in set T exist iff
    - i belongs to T (i.e., label $x_i = 0$) AND
    - an edge exists between s and i (i.e., $\lambda_i > 0$)
  - If edge exists, its cost = λ_i
  - For any voxel i,
    Edge Existence + Edge Cost are captured by :
    $(1 - x_i) \max(0, \lambda_i)$

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Claim : Min s-t Cut equals MAP labeling
  - (2) Edges i–t between vertices in set S and “t” exist iff
    - i belongs to S (i.e., label $x_i = 1$) AND
    - an edge exists between i and t (i.e., $\lambda_i \leq 0$)
  - If edge exists, its cost $= -\lambda_i$
  - For any voxel i,
    Edge Existence + Edge Cost are captured by :
    $x_i \max(0, -\lambda_i)$

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Claim : Min s-t Cut equals MAP labeling
  - (3) Edges i–j between vertices in sets S-s and T-t exist iff
    - labels at the two vertices are different (i.e., $x_i \neq x_j$)
  - If edge exists, its cost = β_{ij}
  - For any voxels i,j
    Edge Existence + Edge Cost are captured by :
    $(x_i - x_j)^2 \beta_{ij}$

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Claim : Min s-t Cut equals MAP labeling
  - Capacity of the cut reduces to :

$$C(S,T) = \sum_i (1 - x_i)\max(0, \lambda_i) \text{ Edges between vertex } s \text{ and vertices in set } T$$

$$+ \sum_i x_i \max(0, -\lambda_i) \text{ Edges between vertices in set } S \text{ and vertex } t$$

$$+ \sum_i \sum_j 0.5(x_i - x_j)^2 \beta_{ij} \text{ Edges between vertices in set } S - \{s\} \text{ and vertices in set } T - \{t\}$$

$$C(S,T) := \sum_{u \in S, v \in T} c_{uv}$$

$$= \sum_{i \in T} c_{si} \text{ Edges between vertex } s \text{ and vertices in set } T$$

$$+ \sum_{i \in S} c_{it} \text{ Edges between vertices in set } S \text{ and vertex } t$$

$$+ \sum_{i \in S - \{s\}, j \in T - \{t\}} c_{ij} \text{ Edges between vertices in set } S - \{s\} \text{ and vertices in set } T - \{t\}$$

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Claim : Min s-t Cut equals MAP labeling
  - Capacity of the cut reduces to :

$$= \sum_i \max(0, \lambda_i) + \sum_i x_i \Big(\max(0, -\lambda_i) - \max(0, \lambda_i) \Big) + \sum_i \sum_j 0.5(x_i + x_j - 2x_i x_j)\beta_{ij}$$

$$= \sum_i \max(0, \lambda_i) + \sum_i x_i \Big(- \lambda_i \Big) + 0.5 \sum_i \sum_j \beta_{ij}(x_i + x_j - 2x_i x_j)$$

Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Proof : Min s-t Cut equals MAP labeling

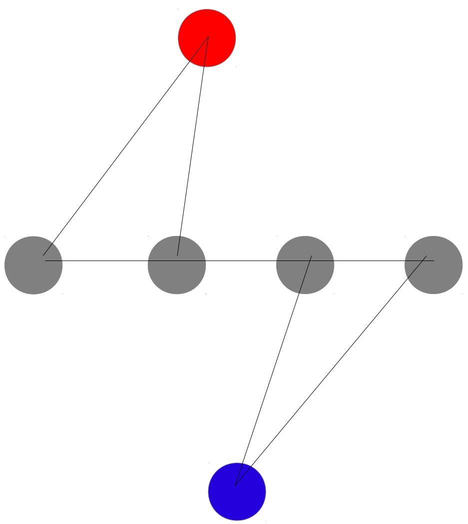
$$
\begin{aligned}
\min_{(S,T)} C(S,T) &= \min_x C(S(x),T(x)) \\
&= \min_x \left(\sum_i \max(0,\lambda_i) + \sum_i x_i \Big(-\lambda_i \Big) + 0.5 \sum_i \sum_j \beta_{ij}(x_i + x_j - 2x_i x_j) \right) \\
&= \min_x \left(-\sum_i x_i \lambda_i - 0.5 \sum_i \sum_j \beta_{ij}(2x_i x_j - x_i - x_j) \right) \\
&= \max_x \left(\sum_i x_i \lambda_i + 0.5 \sum_i \sum_j \beta_{ij}(2x_i x_j - x_i - x_j) \right) \\
&= \max_x P(x|y,\theta)
\end{aligned}
$$

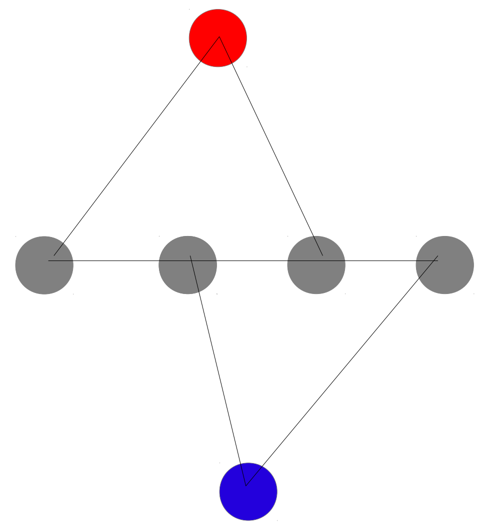
Bayesian Image Segmentation

- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - Strengths
    - Global optimum in polynomial time !
      - Various algorithms : polynomial in number of vertices or edges or both
        - http://heim.ifi.uio.no/~geird/bergen.pdf
        - http://www.cs.princeton.edu/courses/archive/spr05/cos423/lectures/07maxflow.pdf
      - NOT obvious because number of possible cuts grows exponentially with number of vertices
    - Any likelihood (noise) model can be built in
  - Limitations
    - Doesn't produce soft memberships
    - Handles only 2 classes
      - Min cut with > 2 labels is NP hard
        (but fast approximations with bounded sub-optimality exist)
        - http://www.cs.cornell.edu/courses/cs5540/2010sp/lectures/Lec18-graph-cuts.v1.pdf

Bayesian Image Segmentation

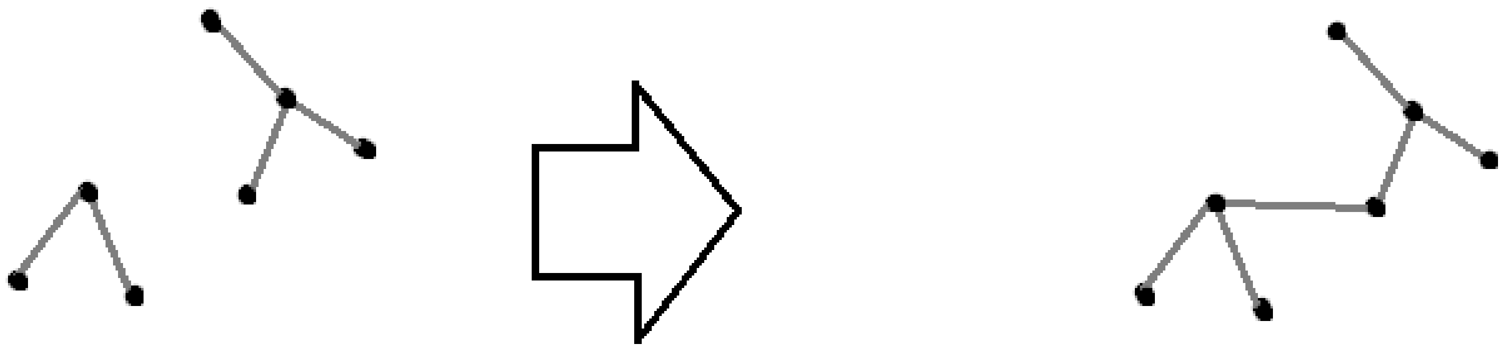
- MAP-MRF Segmentation via s-t Cuts (Hard Seg)
  - **Example 1** **Example 2**





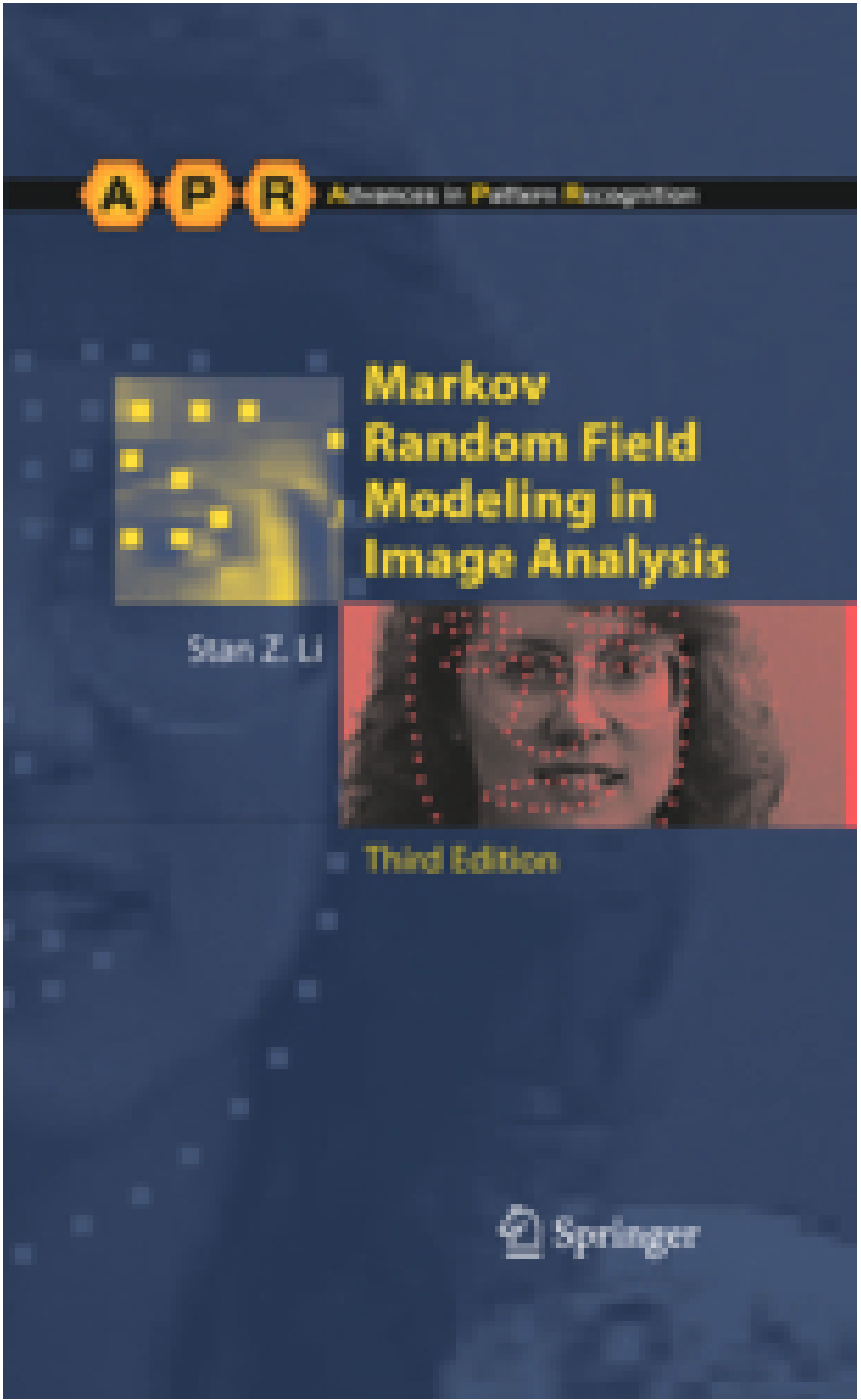
(1) What are the cuts if all βij = 0 ?

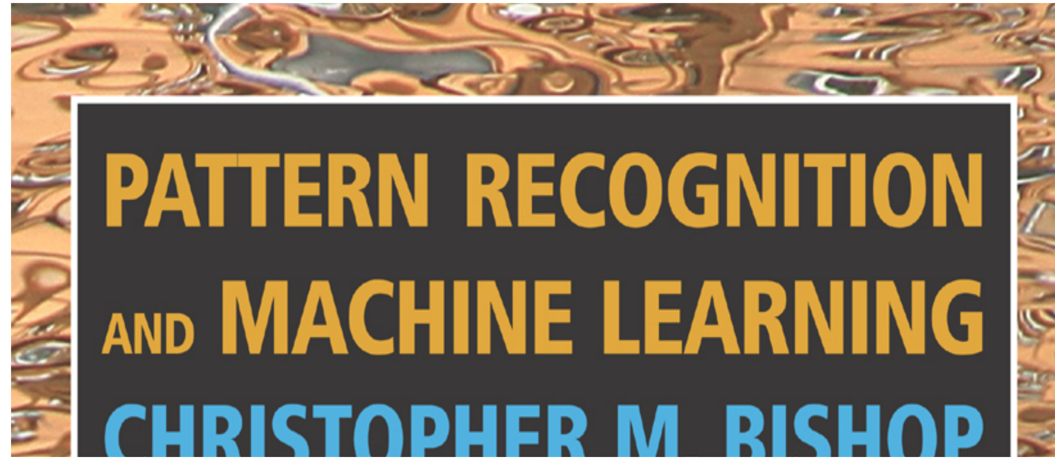
(2) What are possible cuts if βij values >> | λi | values ?

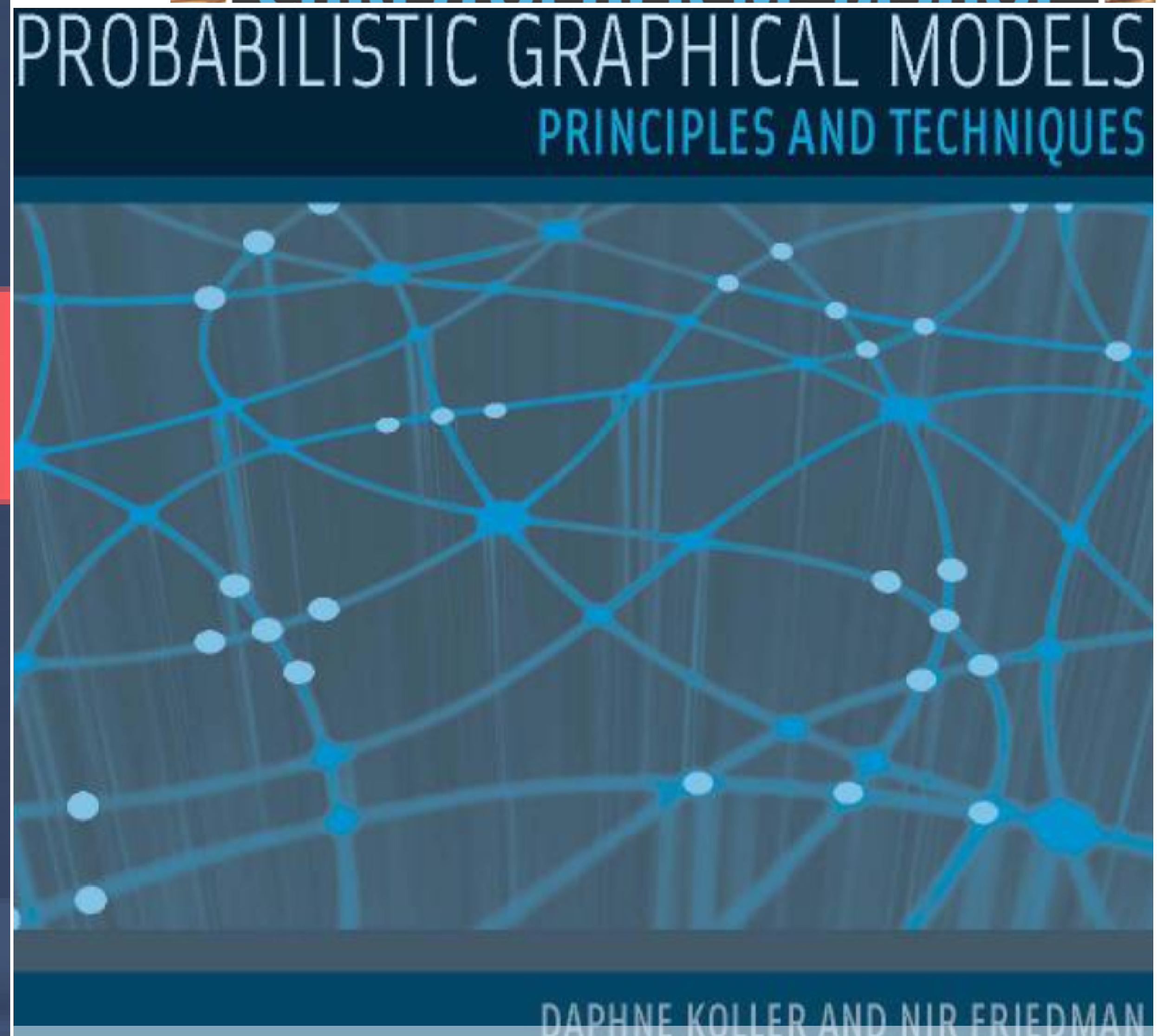


Deforestation:
When adding a branch gives you fewer trees.

For More ...

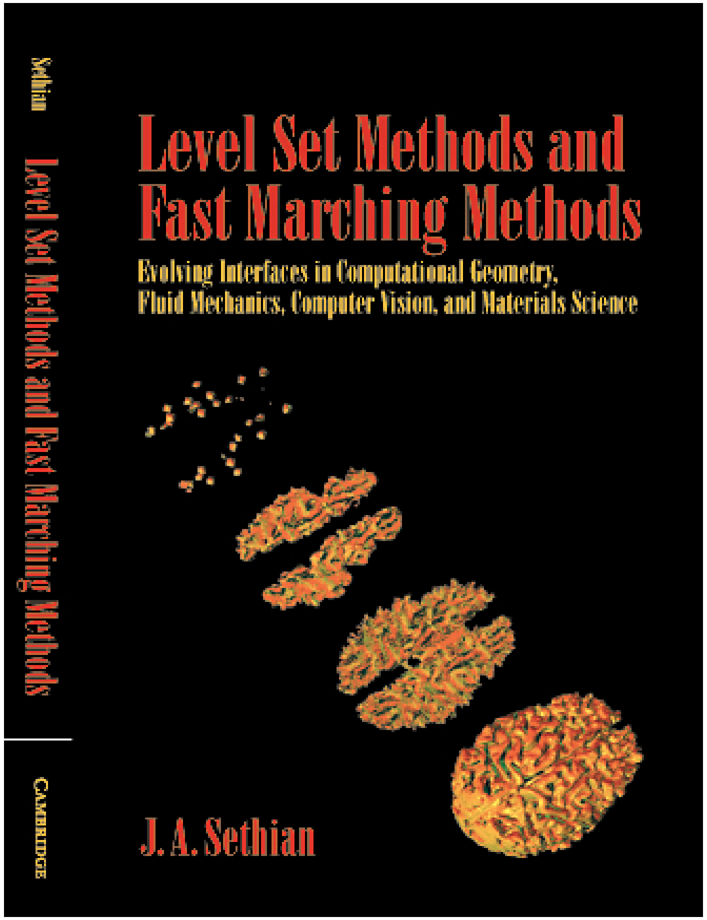

A P R
Markov
Random Field
Modeling in
Image Analysis
Stan Z. Li
Third Edition
Springer

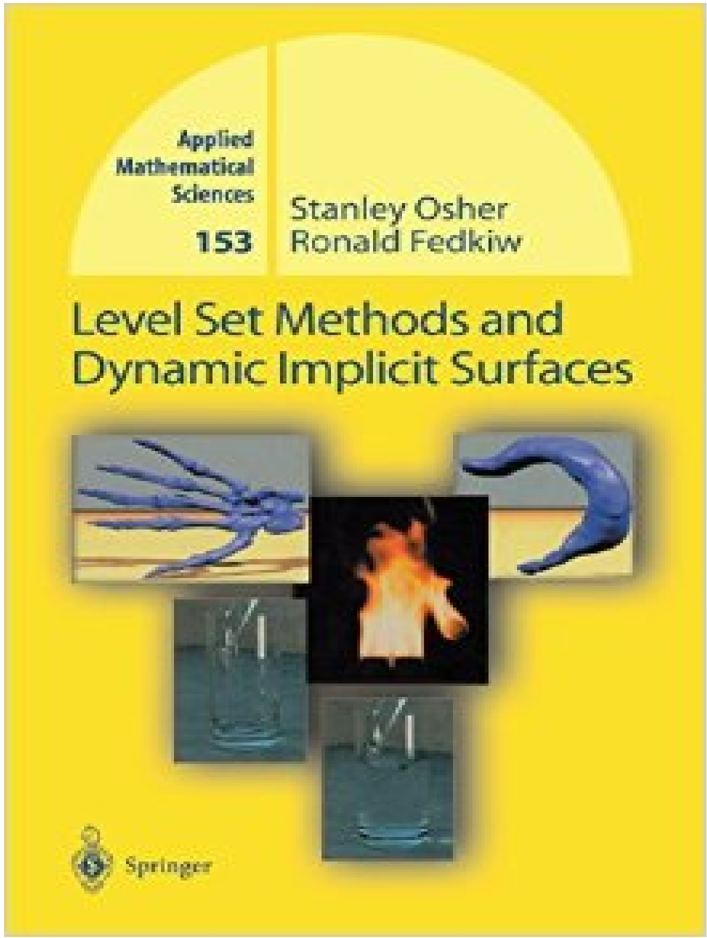

PATTERN RECOGNITION
AND MACHINE LEARNING

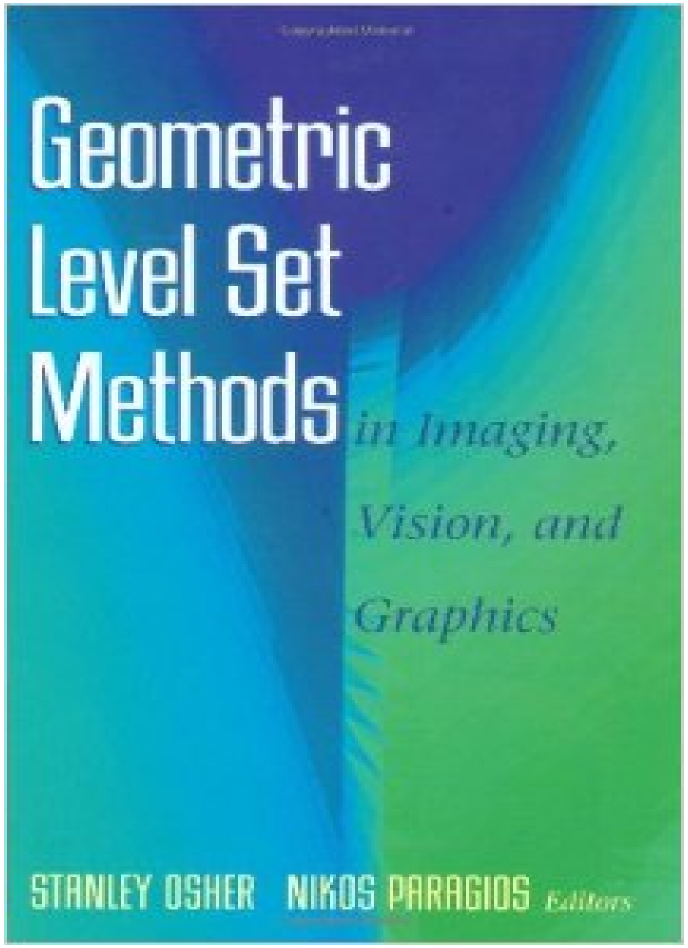

PROBABILISTIC GRAPHICAL MODELS
PRINCIPLES AND TECHNIQUES
DAPHNE KOLLER AND NIR FRIEDMAN

Level-Set Based Segmentation

- A brief overview
  - https://math.berkeley.edu/~sethian/2006/level_set.html
- Books







Level-Set Based Segmentation

- Level set = contour = iso-surface
  - Of some function

Level-Set Based Segmentation

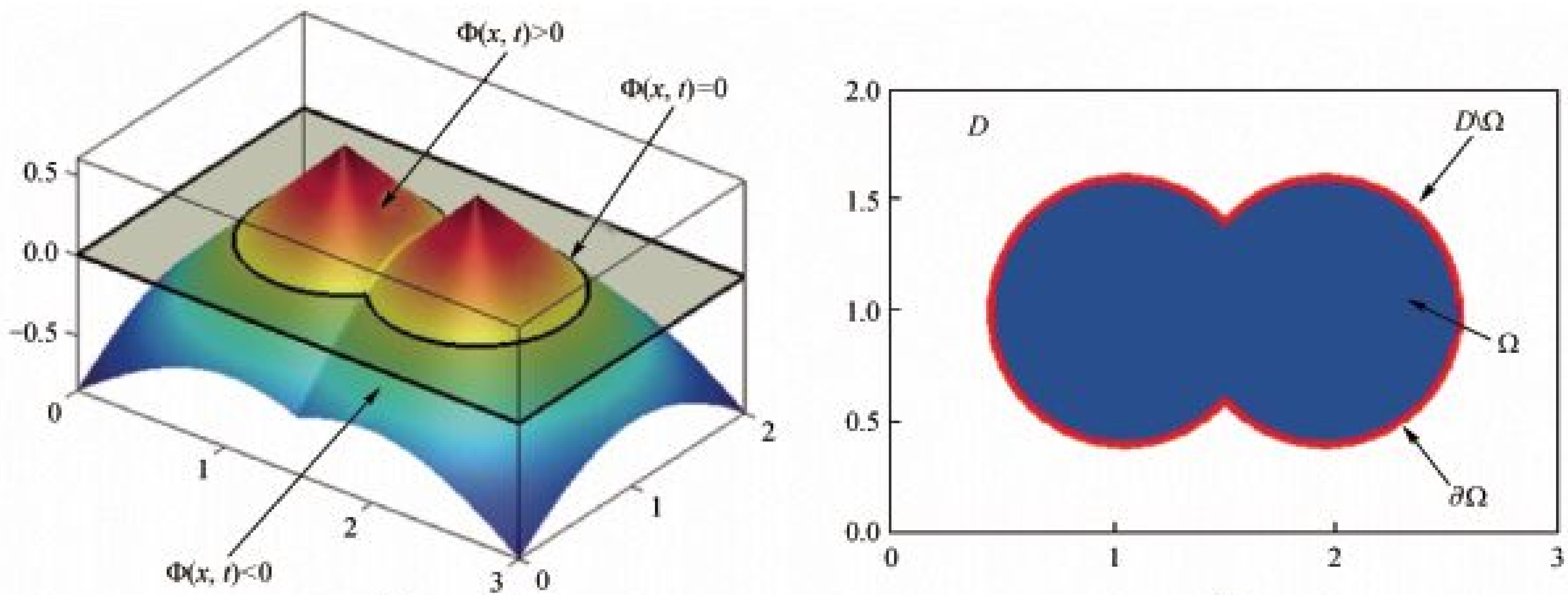
- Key ideas

  (1) Object boundary / surface representation

  (1) Represent object boundary / surface as **zero-level set** of ...

  (2) … **boundary's (signed) distance transform**

  - 1D Curve (2D surface) represented using a 2D (3D) image



Level-Set Based Segmentation

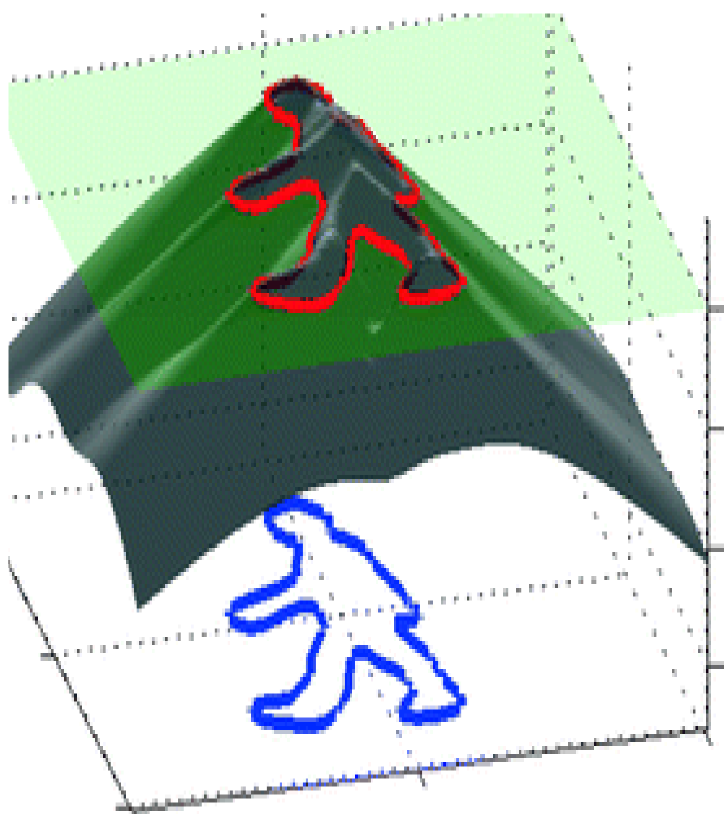
- Key ideas

  (1) Object boundary / surface representation

    (1) Represent object boundary / surface as zero-level set of ...

    (2) … boundary's (signed) distance transform

  - 1D Curve (2D surface) represented using a 2D (3D) image

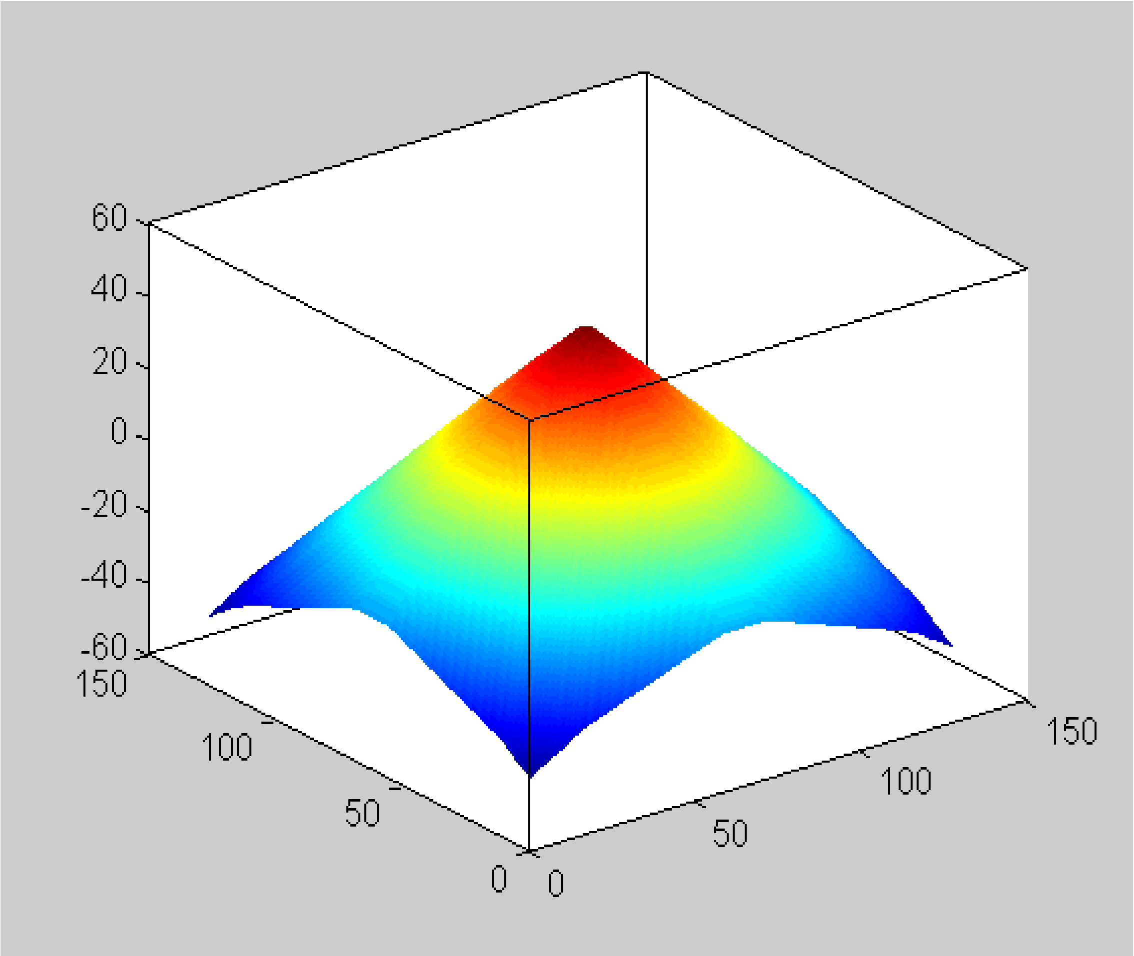


Level-Set Based Segmentation

- Key ideas

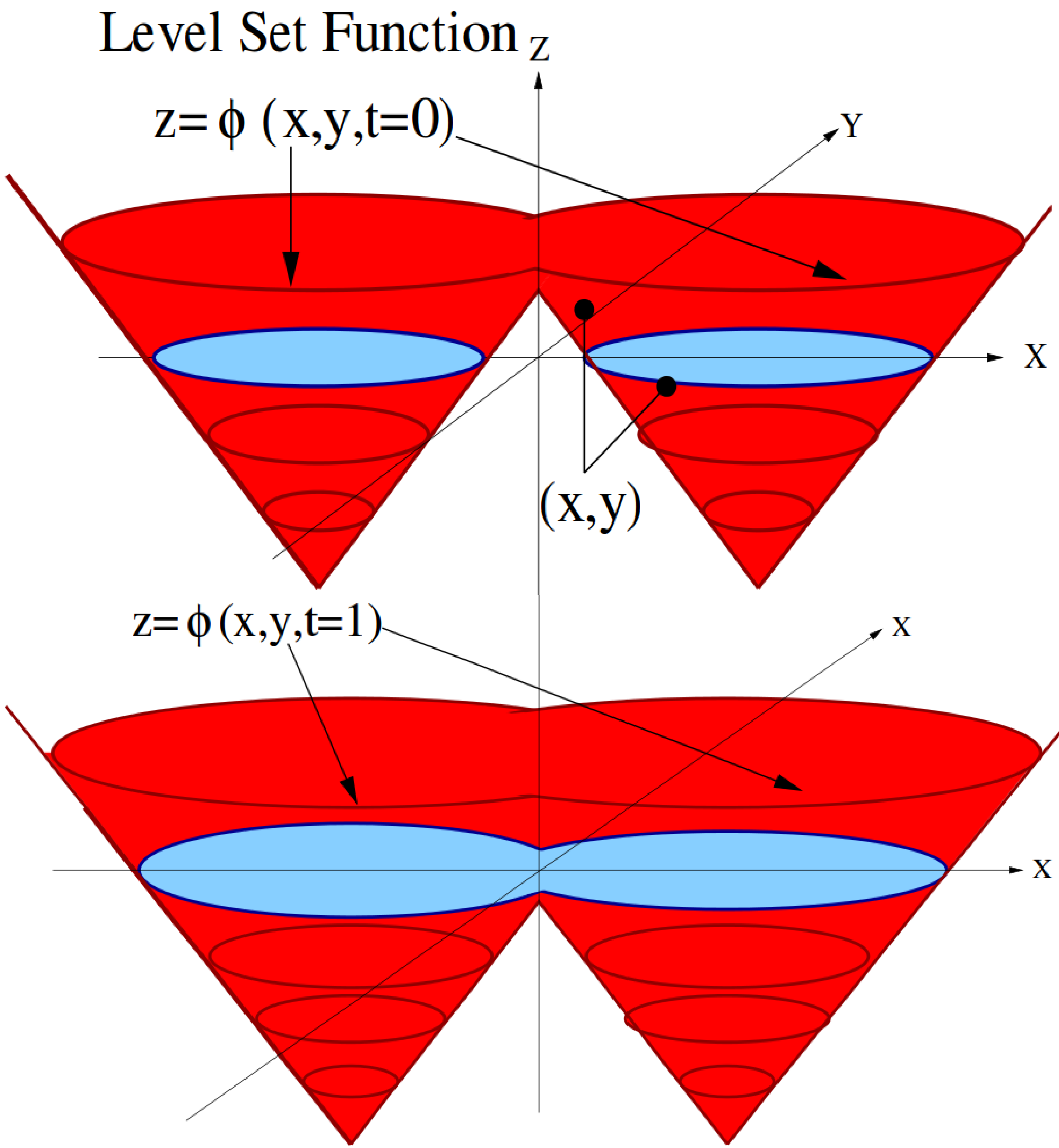
  (2) Surface evolution (to fit to the data)

  - Modify distance transform to move surface
  - http://www.cs.swan.ac.uk/~csjason/rbf/gifs/1circle_levelset.gif



Level-Se

- Key ideas
  - (2) Surface evc
    - Modify distanc

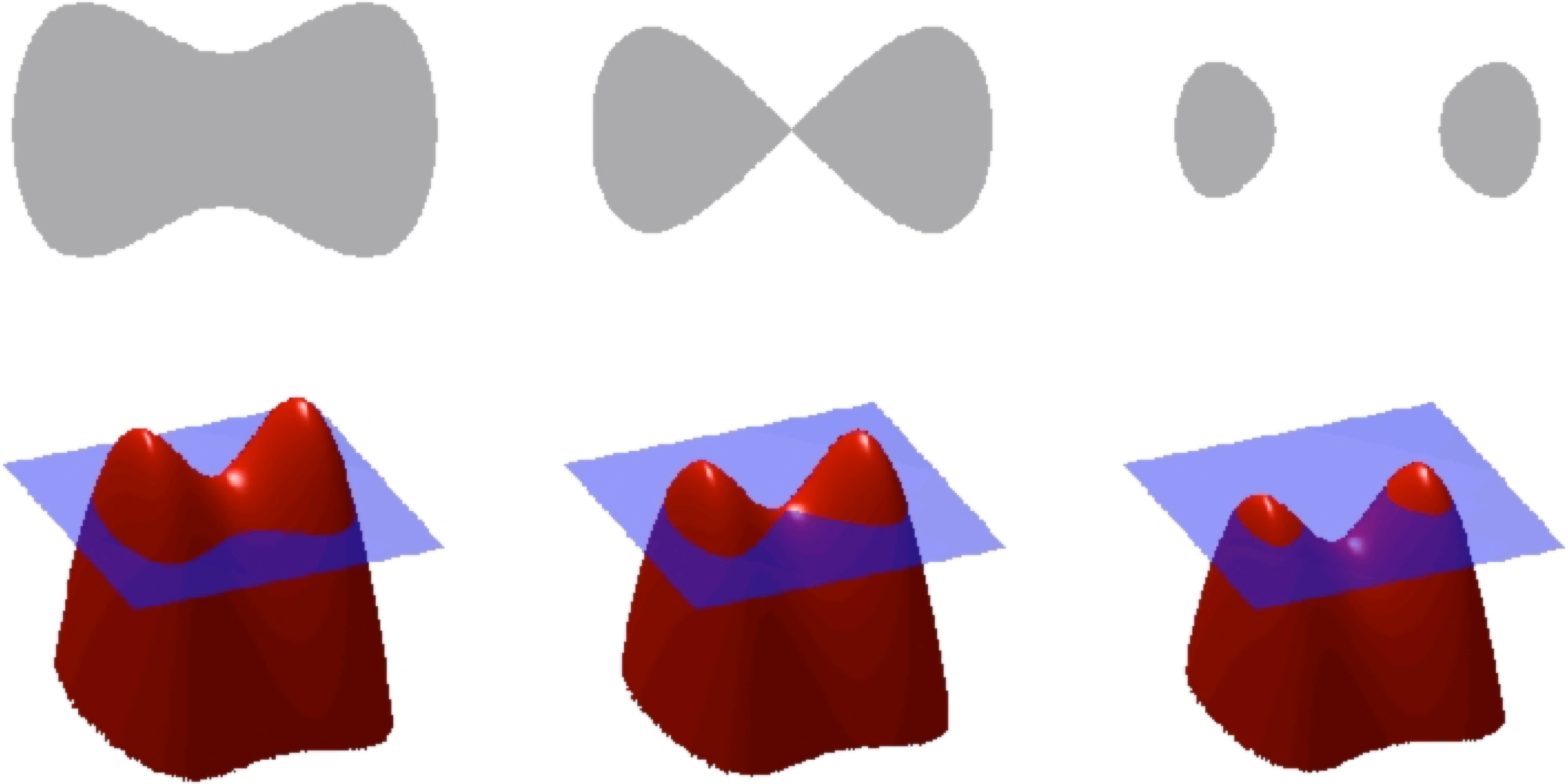


Level-Set Based Segmentation

- Key ideas

  (2) Surface evolution

  - Modify distance transform to move surface

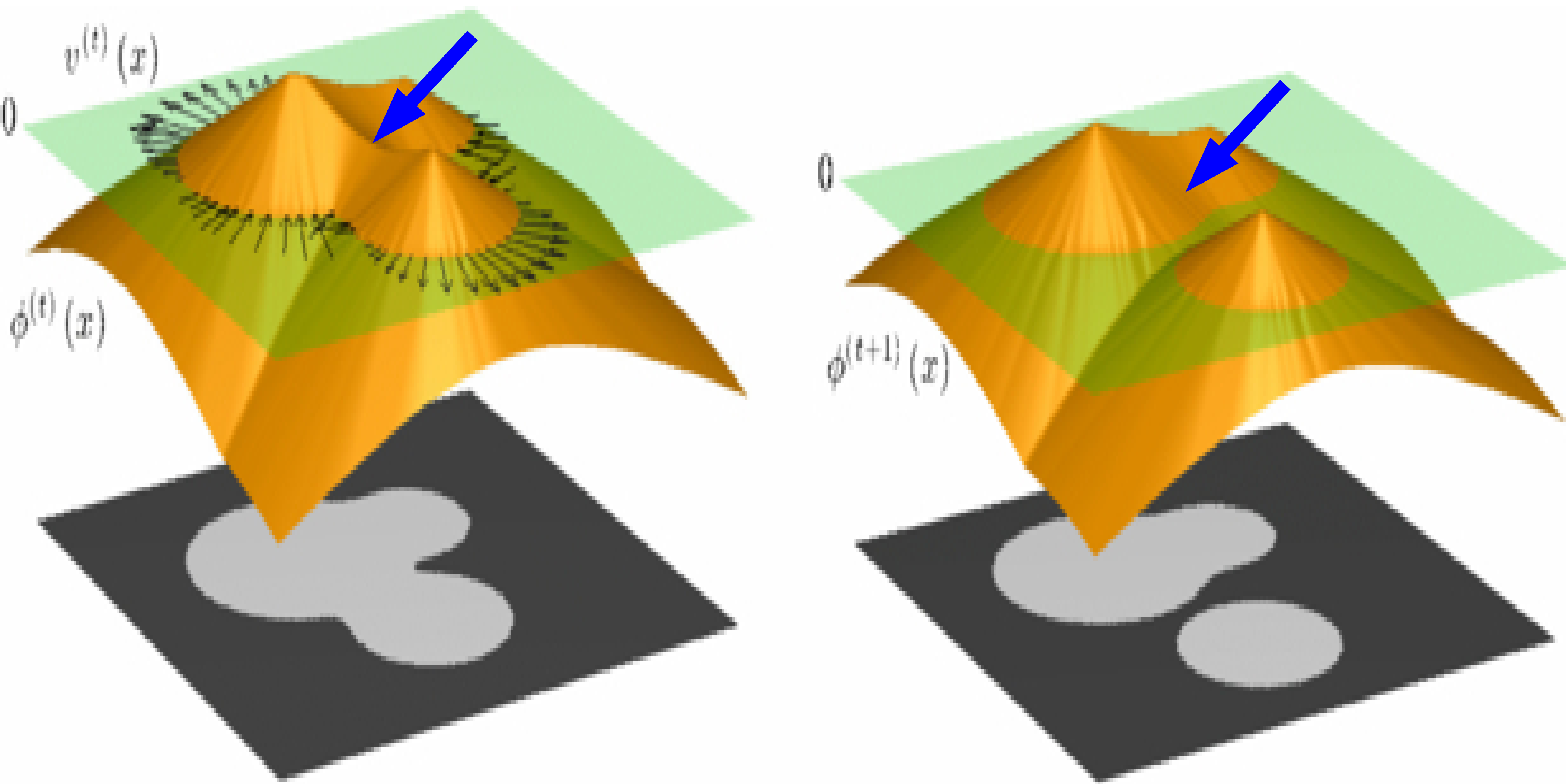


Level-Set Based Segmentation

- Key ideas

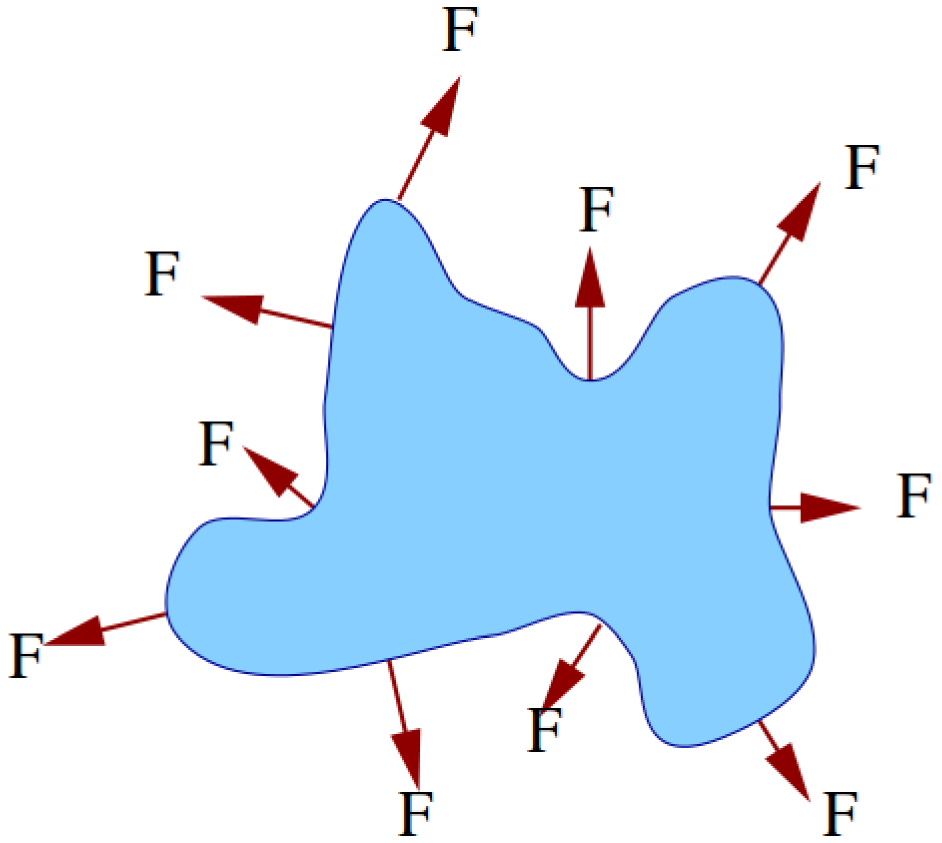
    (2) Surface evolution

    - Modify distance transform to move surface



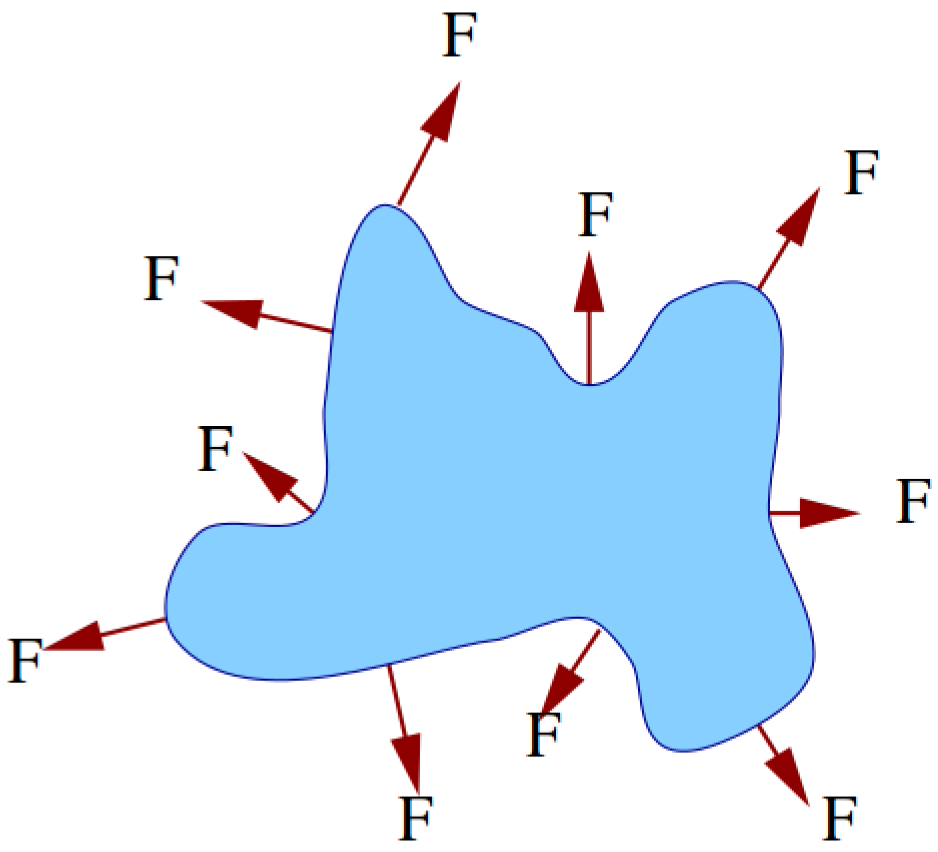
Level-Set Based Segmentation

- **Curve evolution / flow**
  - **Each point moves** according to a **force along its normal**
  - Don't consider / define forces along tangent directions
    - Why ?
    - Such forces only **slide** points along curve
    - Such forces don't change curve



Level-Set Based Segmentation

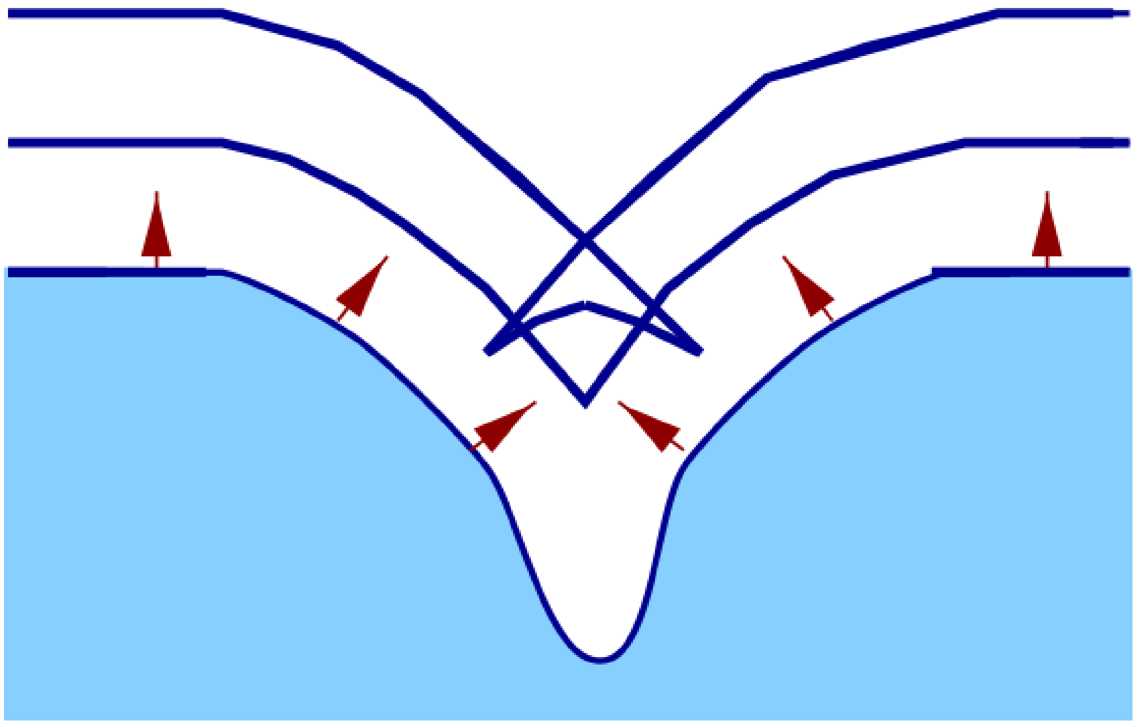
- Advantages over mesh representations
  - Avoid detecting intersections
    - Self intersections
    - Intersections between components

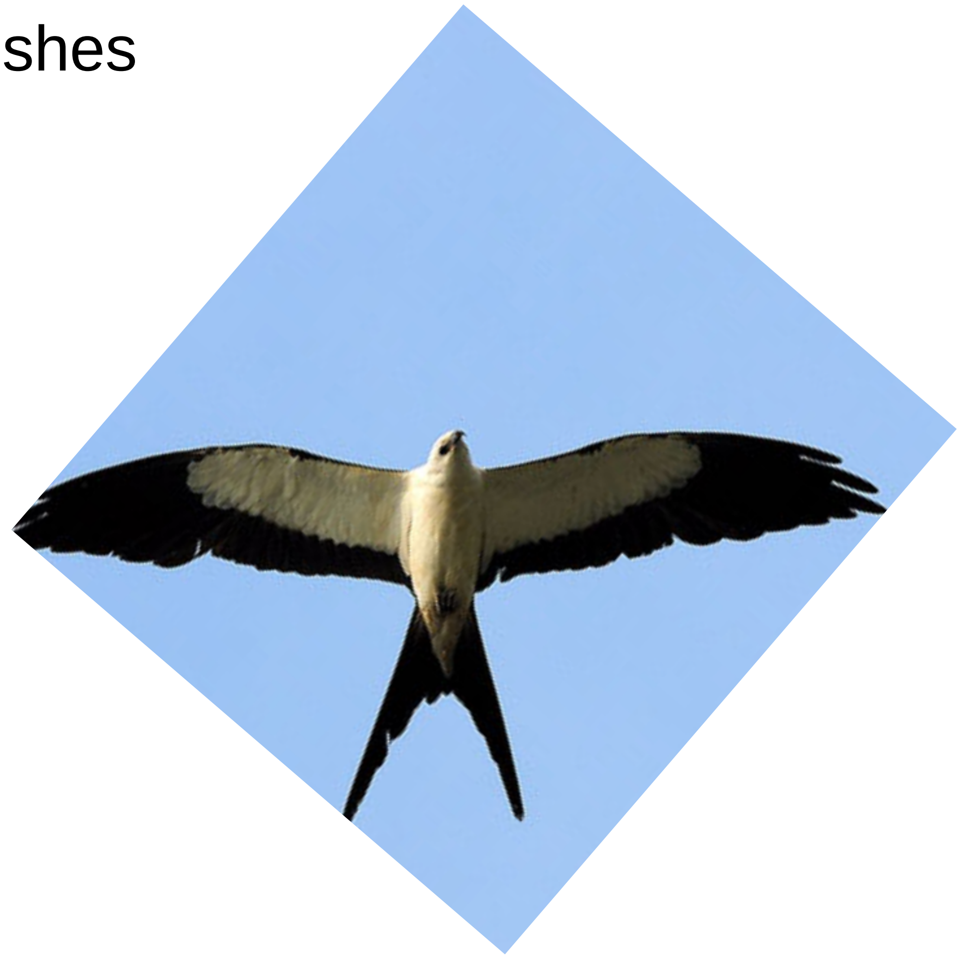


  - Easy to handle splitting of 1 contour into 2 contours
  - Easy to handle merging of 2 contours into 1 contour

Level-Set Based Segmentation

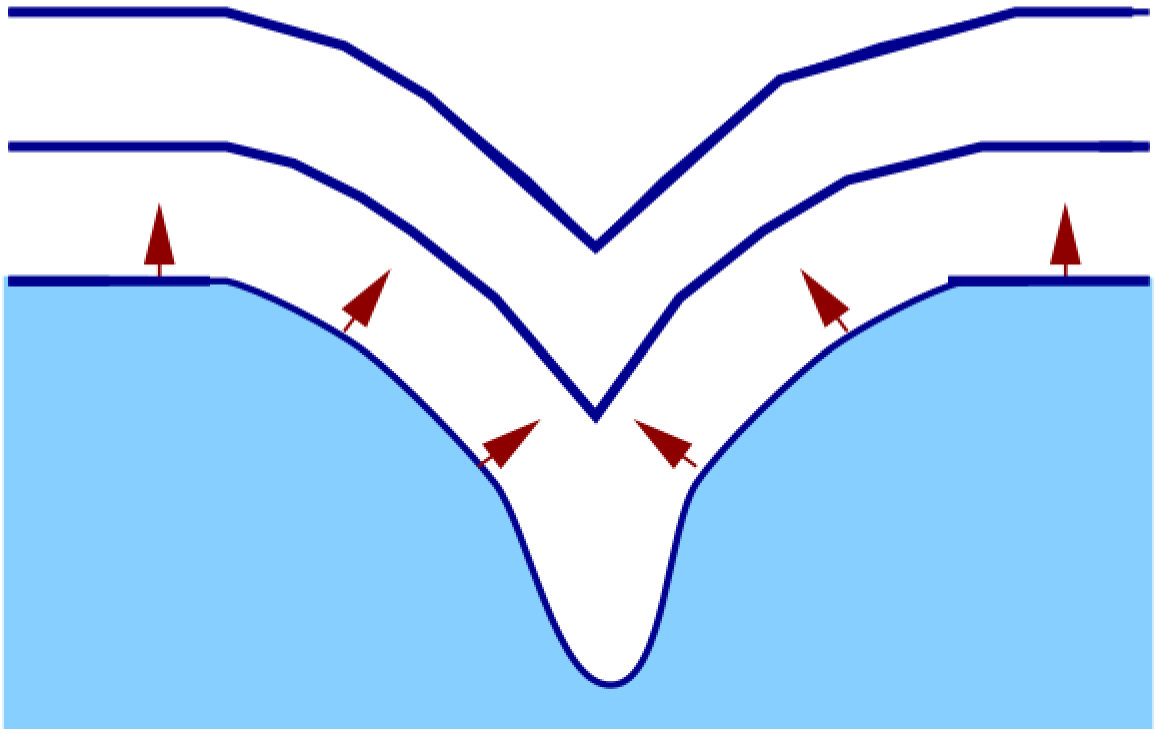
- **Self intersection problem**
  - Consider a sinusoid-like curve
  - Each point moves along its normal with same force F
  - “Swallowtail” problem
    - More difficult to handle with meshes





Level-Set Based Segmentation

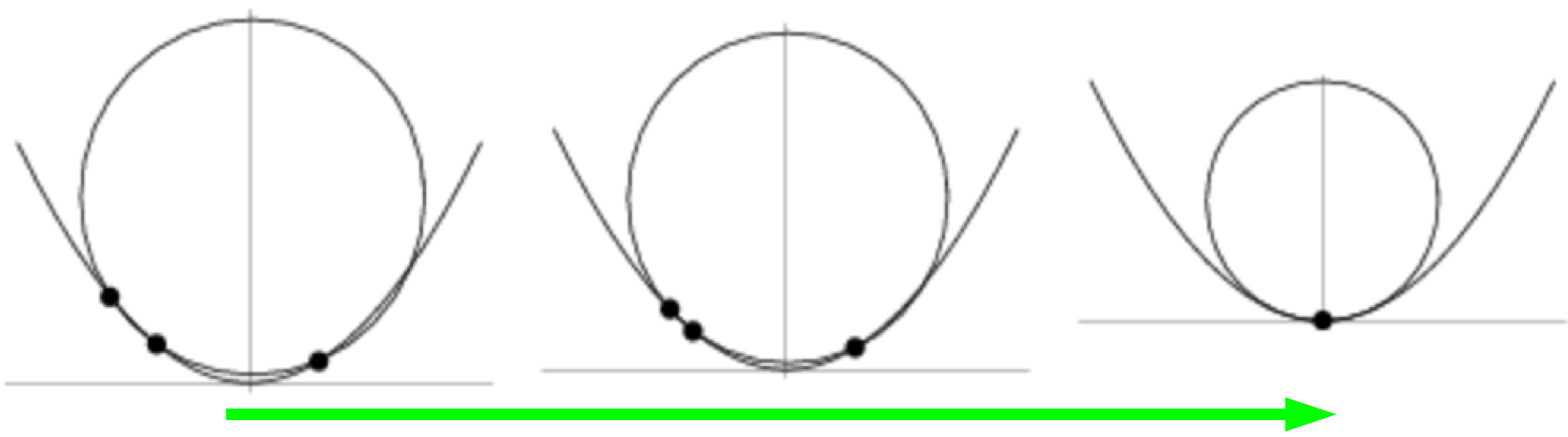
- **Avoiding self intersections**
  - This is more desirable:



  - Must **avoid corners** too → undefined derivatives (e.g., normal direction)

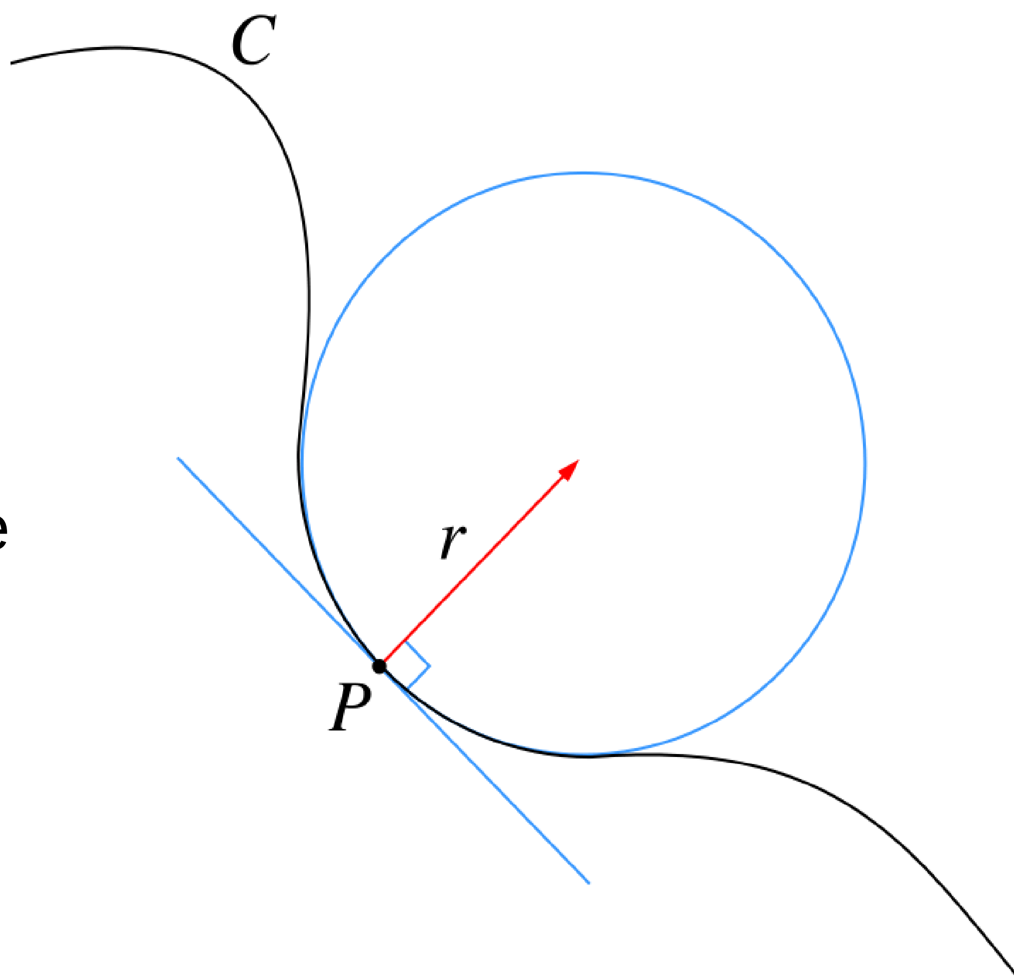
Level-Set Based Segmentation

- **Curvature** of a (smooth) curve, at point P
  = Curvature of its *osculating circle* at point P
  - **Osculating circle**, at P
    - Let point on 2D curve = P = (f(t), g(t)), where t = parameter
    - Limit of a sequence of circles passing through points corresponding to t1, t, t2 (s.t. t1 < t < t2) as t1 → t and t2 → t



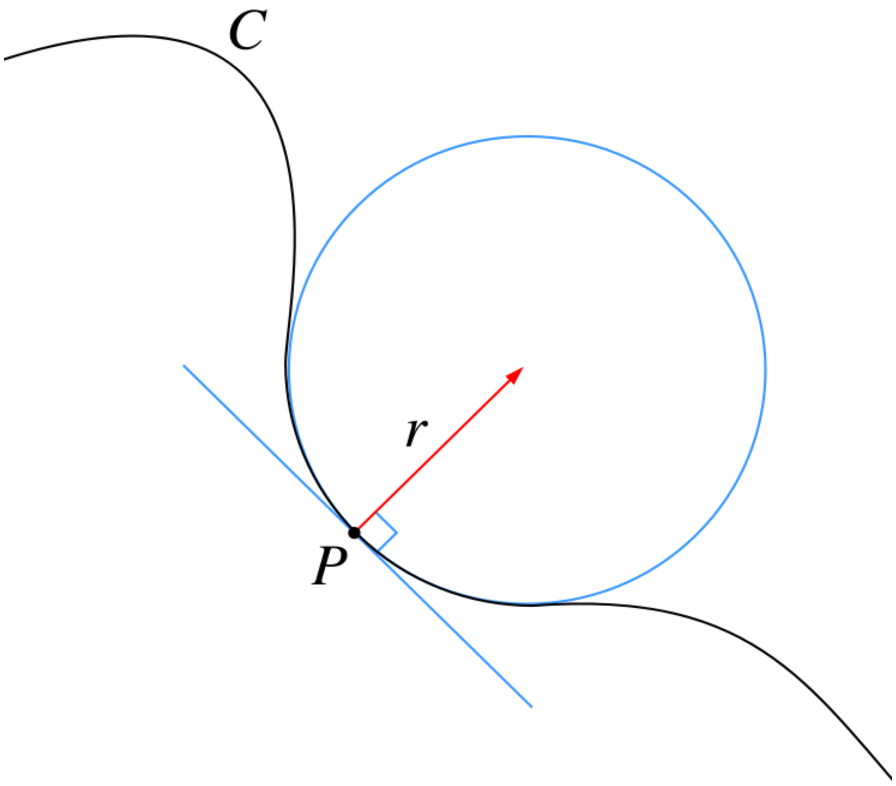
Level-Set Based Segmentation

- **Curvature** of a (smooth) curve, at point P
  = Curvature of its *osculating circle* at point P
  - **Osculating circle**, at P
    - One among circles, all tangent to curve tangent at P, which **best approximates the curve near P**
    - Center = center of curvature
    - Radius r = radius of curvature
    - **Curvature of curve = 1 / r**



Level-Set Based Segmentation

- **Curvature** of a (smooth) curve
  - What is curvature of a **circle** at any point ?
    - Radius of osculating circle = ?
  - What is the curvature of a **line** at any point ?
    - Radius of osculating circle = ?

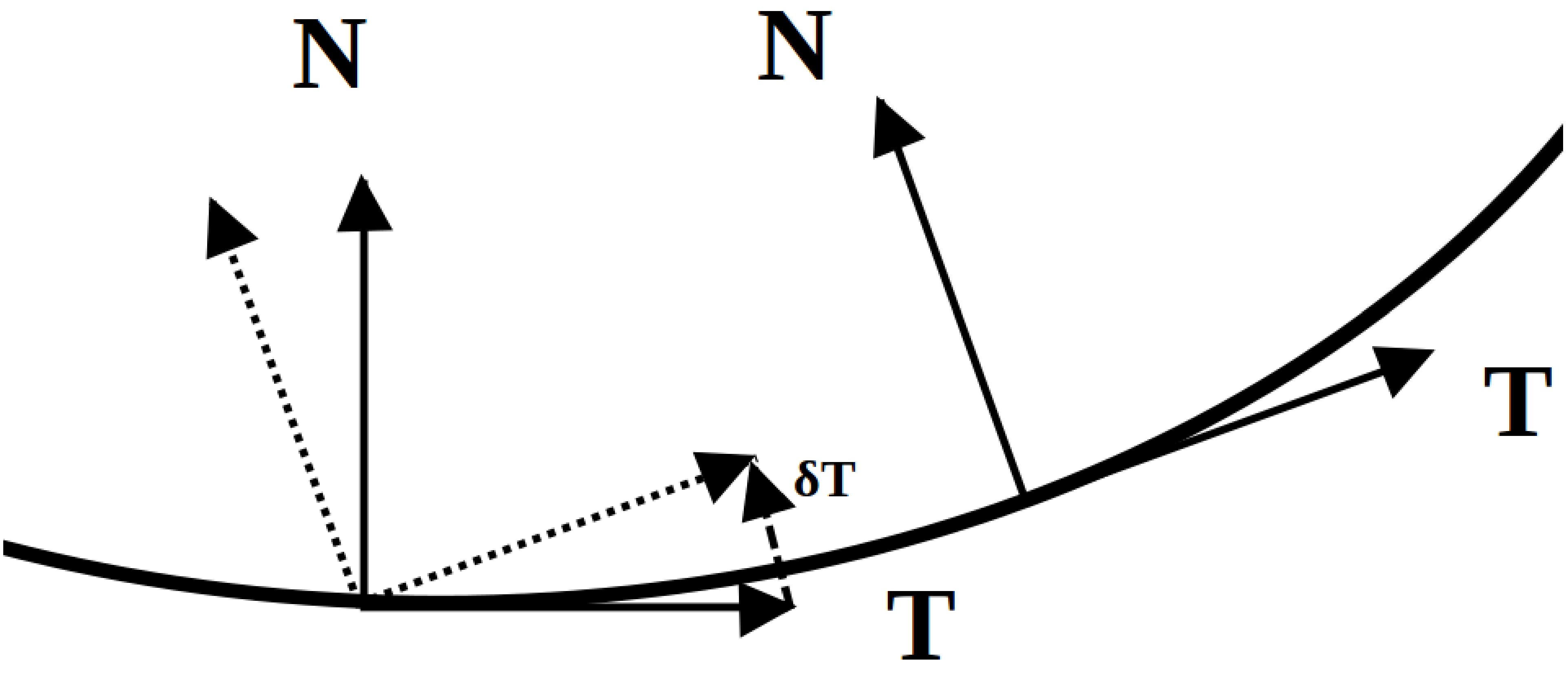


Level-Set Based Segmentation

- **Curvature** of a (smooth) curve

  = magnitude of rate of change of tangent vector (or normal vector)

  = || dT(s) / ds ||, where s = arc length (NOT parameter)

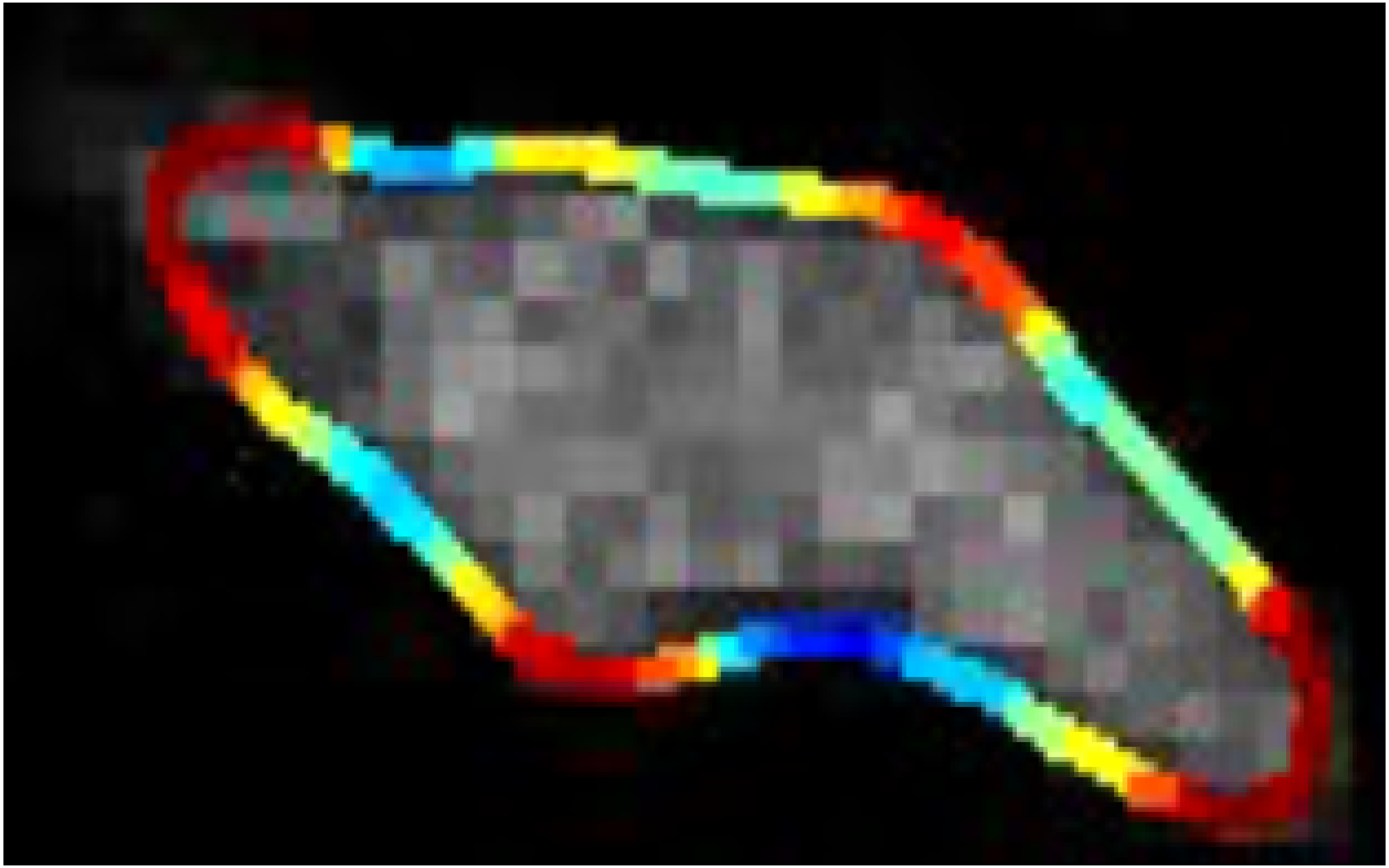


Level-Set Based Segmentation

- **Curvature** of a (smooth) curve
  - **Signed** curvature: depends on **orientation** of curve
    - e.g., whether curve goes from left → right or vice versa
  - Sinusoidal curve has points with positive and negative curvature

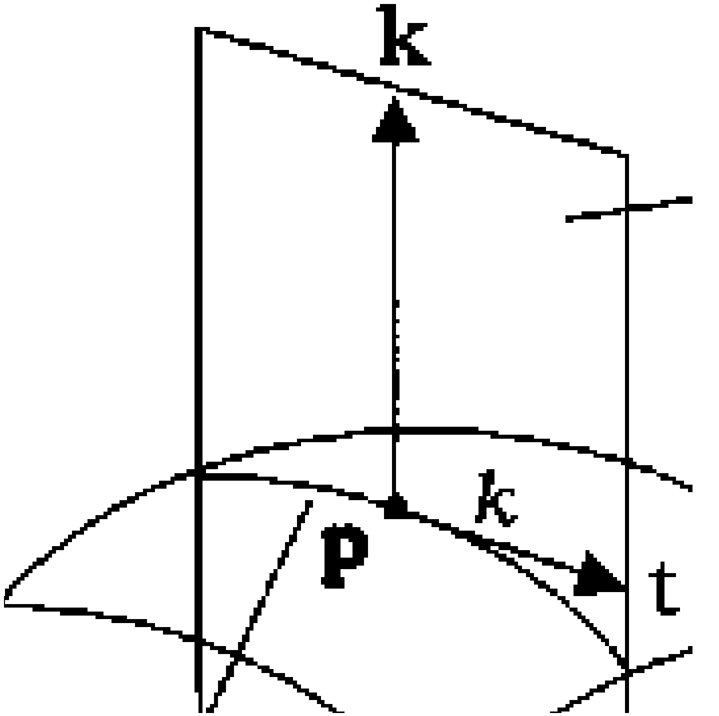
Level-Set Based Segmentation

- Curvature of a (smooth) curve
  - http://upload.wikimedia.org/wikipedia/commons/3/3e/Cell-Shape-Dynamics-From-Waves-to-Migration-pcbi.1002392.s007.ogv
  - Red = large positive
  - Blue = large negative



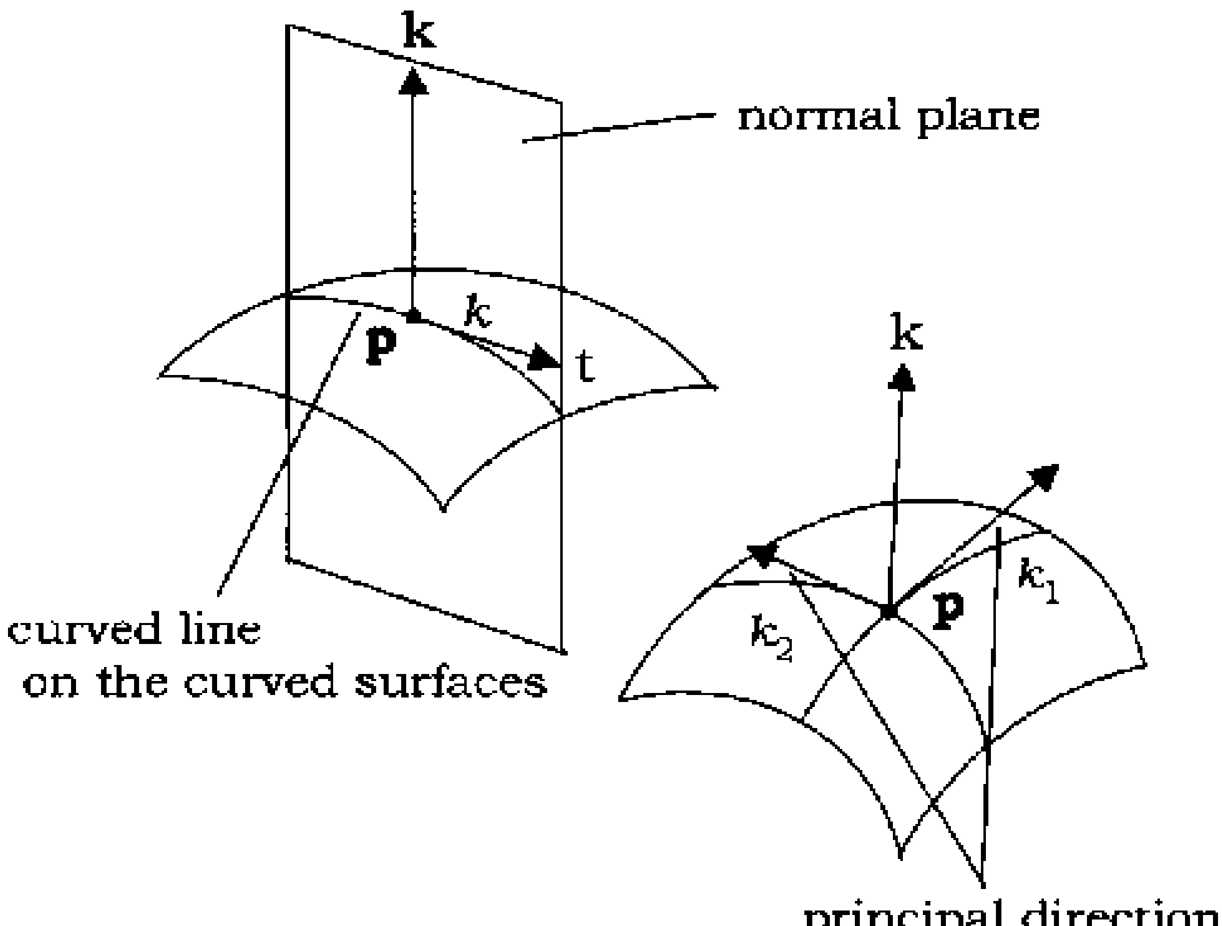
Level-Set Based Segmentation

- **Curvature** of a (smooth) surface
  - Consider a 1D curve on 2D surfaces passing through P
  - Consider a tangent vector 't' at P
    - Many tangent vectors
  - Consider a normal vector 'n' at P
    - Unique upto sign
  - Consider a plane passing through P, and containing a single 't' and 'n'
  - Consider curvature of curve on surface within this plane (at point P)
    - **Normal curvature**



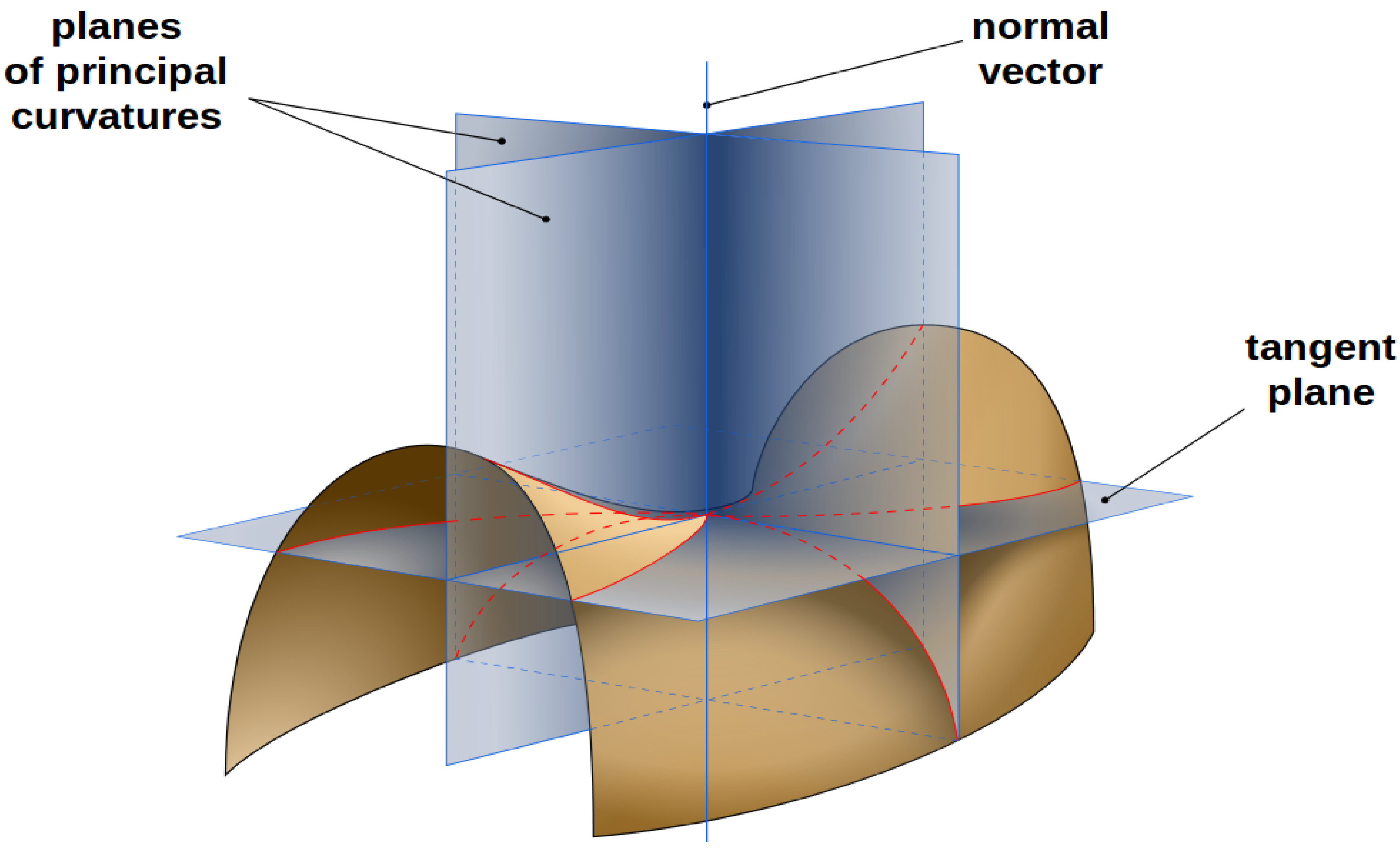
Level-Set Based Segmentation

- **Curvature** of a (smooth) surface
  - Over all possible tangent vectors:
    - Maximum curvature and minimum curvature are called **principal curvatures**



Level-Set Based Segmentation

- **Curvature** of a (smooth) surface
  - Principal curvatures
    - At saddle point in picture: k1 > 0, k2 < 0, |k1| > |k2|

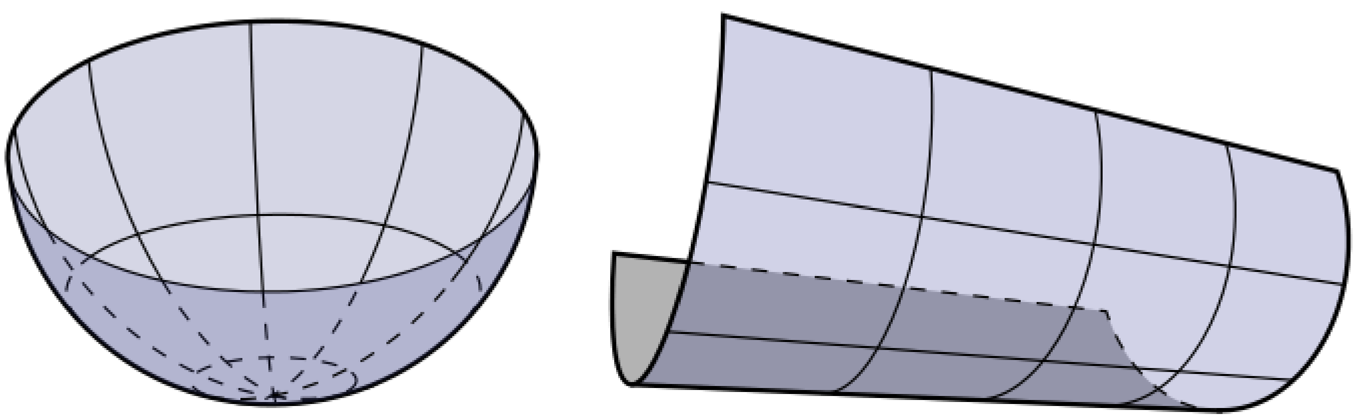


Level-Set Based Se

- Saddle

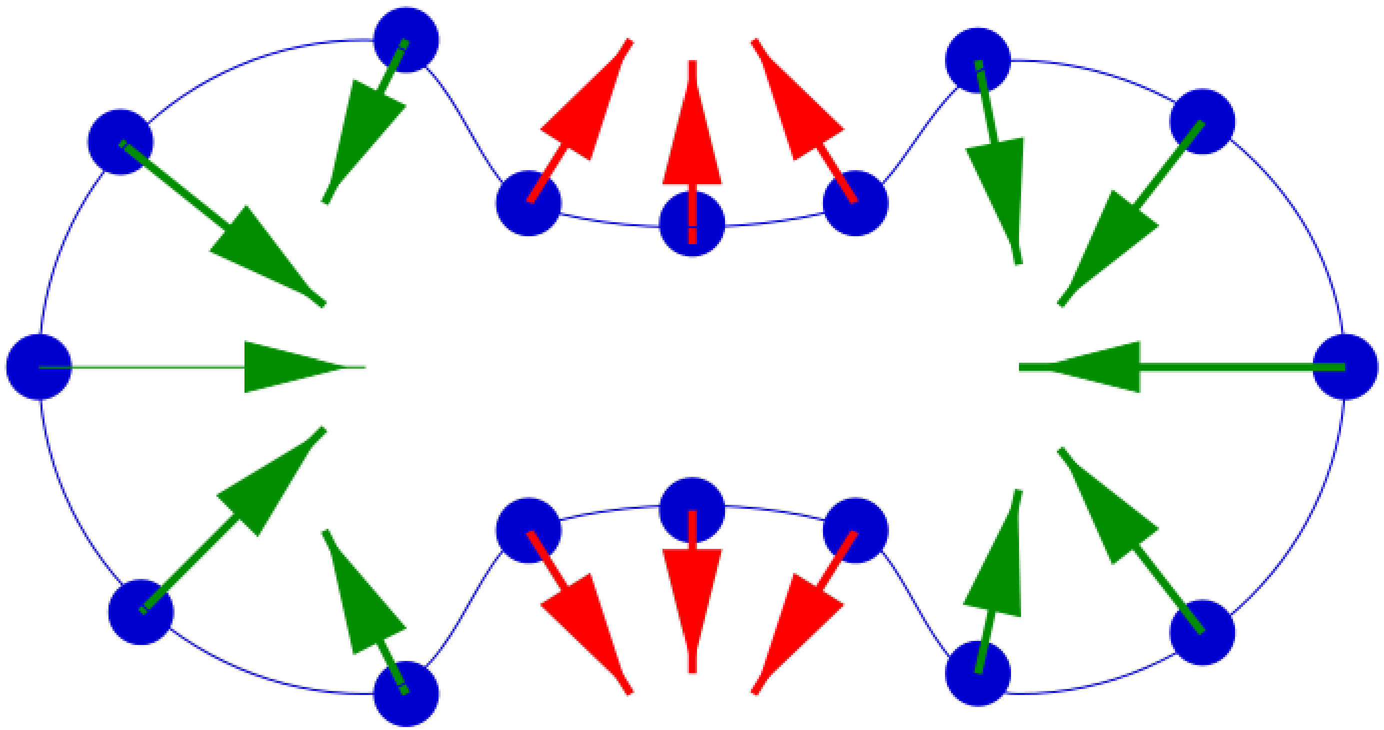
Level-Set Based Segmentation

- **Curvature** of a (smooth) surface
  - Principal curvatures: k1, k2
  - **Gaussian curvature** = k1 * k2
  - **Mean curvature** = (k1 + k2) / 2
- What are the principal curvatures for any point on:
  - A sphere ?
  - A cylinder ?
  - A plane ?



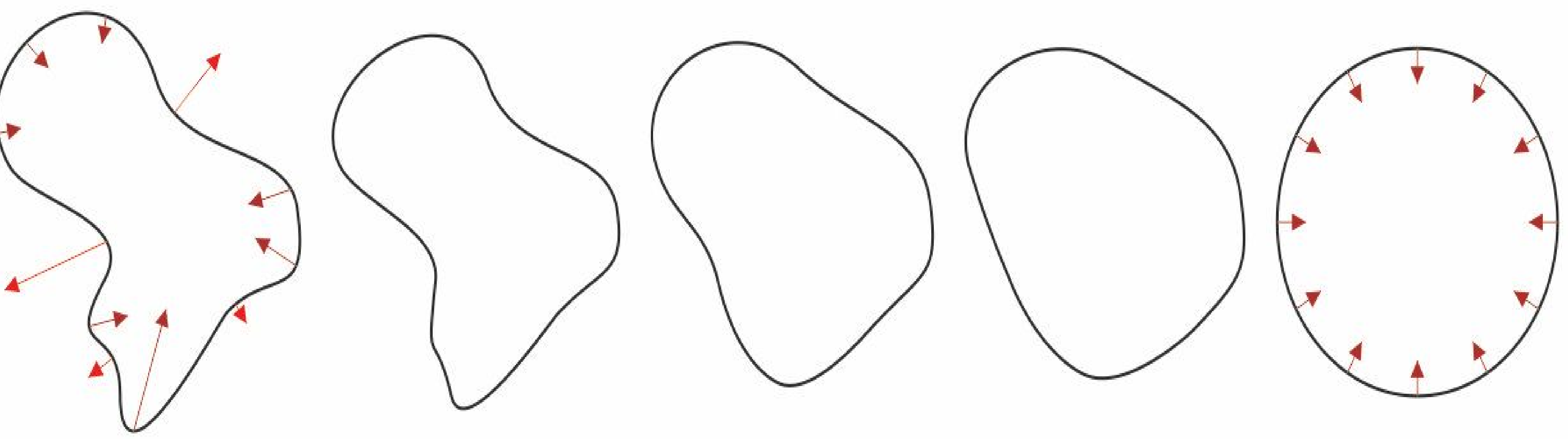
Level-Set Based Segmentation

- **Mean curvature flow**
  - Each point on curve moves along its normal driven by a force proportional to its **mean curvature**
    - Move along **<u>outward</u>** normal
    - Force = **(-1)** * **signed** mean curvature



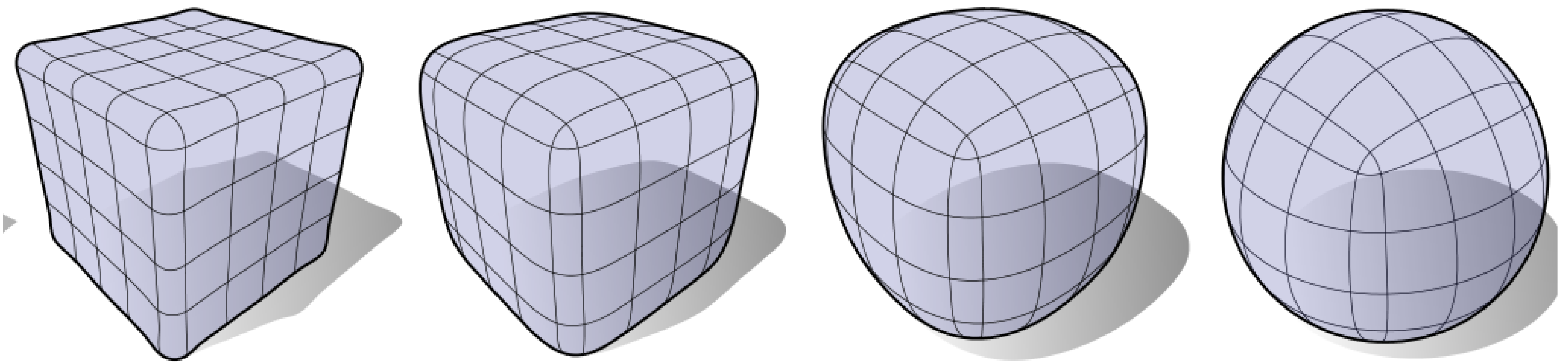
Level-Set Based Segmentation

- Mean curvature flow
  - Each point on curve moves (direction and magnitude) based on **mean curvature**



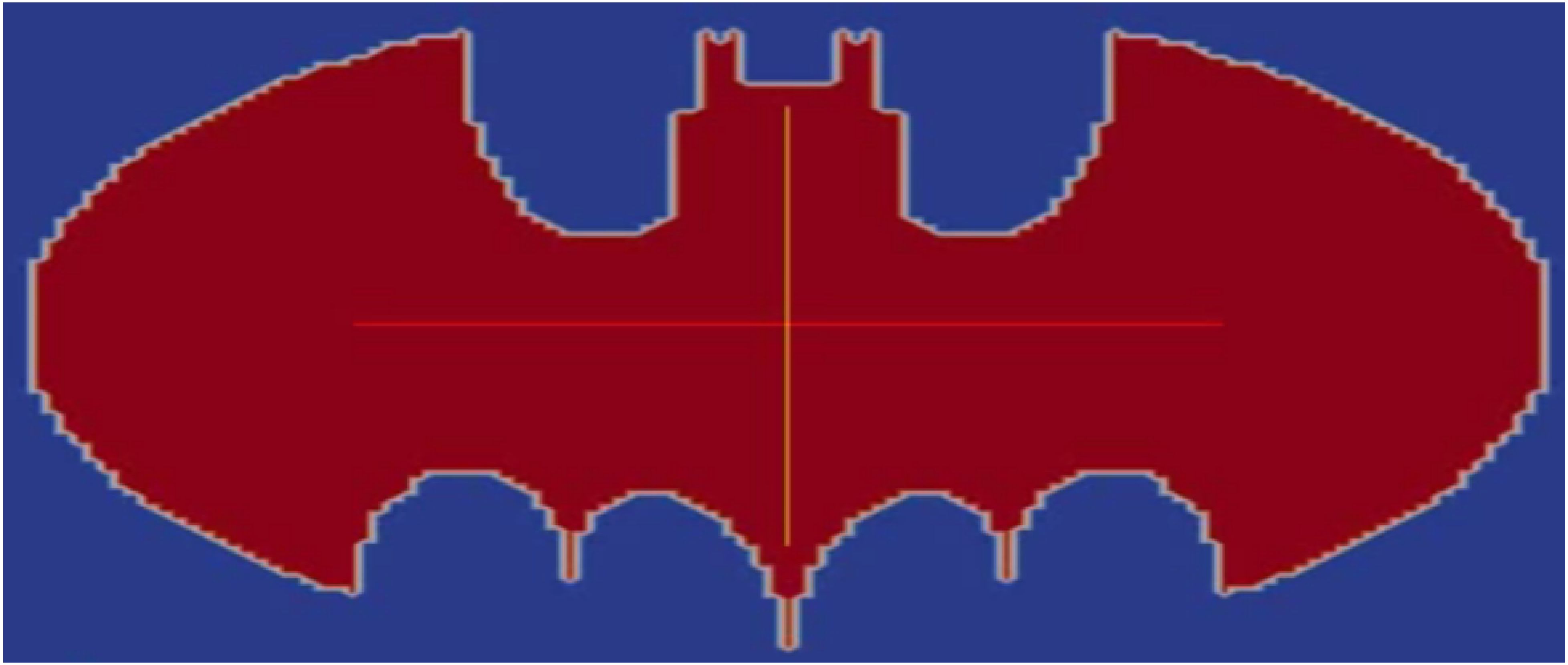
Level-Set Based Segmentation

- Mean curvature flow
  - Each point on curve moves (direction and magnitude) based on **mean curvature**



Level-Set Based Segmentation

- Mean curvature flow
  - Each point on curve moves (direction and magnitude) based on **mean curvature**
  - https://www.youtube.com/watch?v=TeuLhhR7UxM



Level-Set Based Segmentation

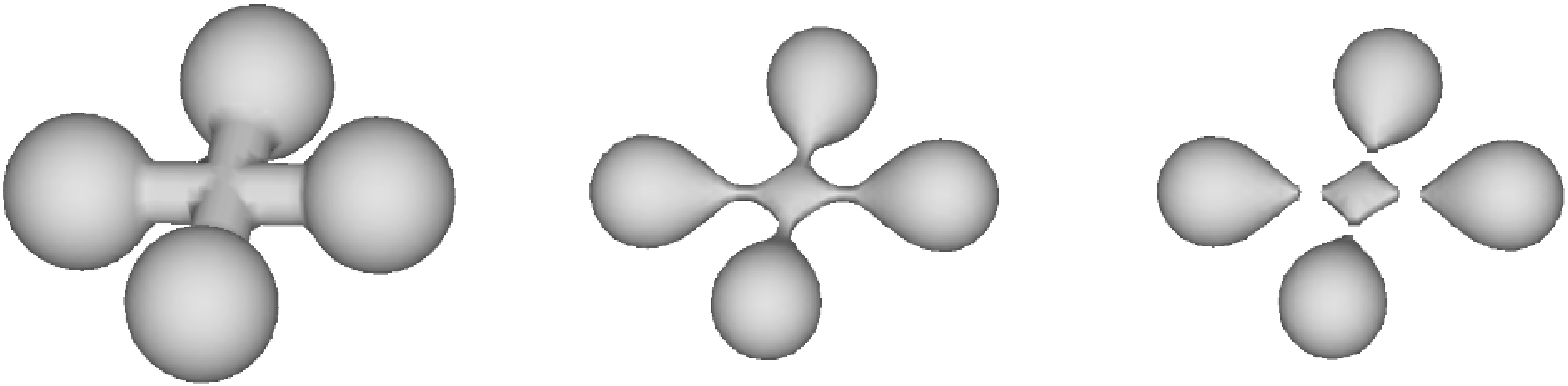
- Mean curvature flow
  - <u>Theorem:</u> **Every** simple closed curve (in 2D), evolving under mean curvature flow, collapses smoothly to a single point, without crossing over itself
  - Implications:
    - No matter how complicated / convoluted a curve might be ...
    - … the curve will evolve into a circle-like curve

Level-Set Based Segmentation

- Mean curvature flow

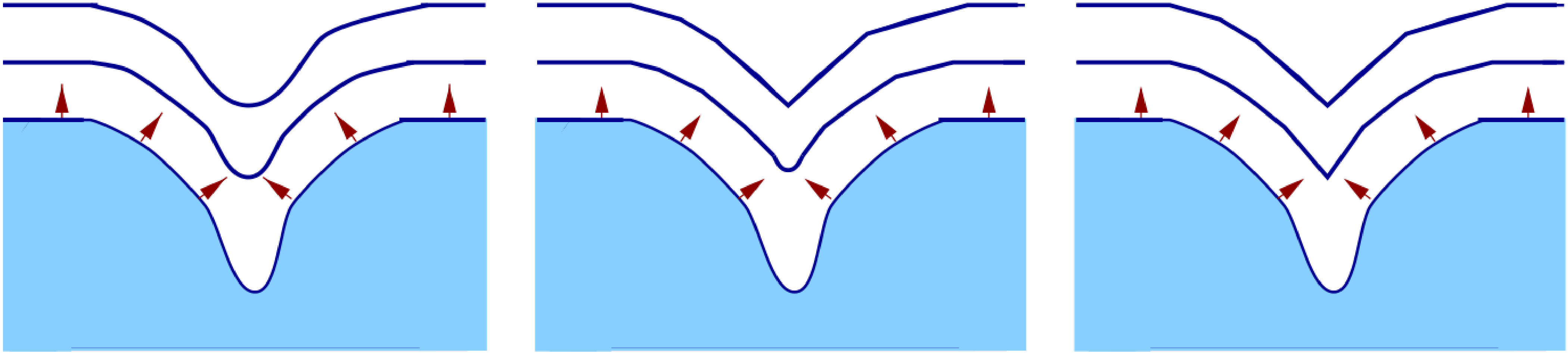
Level-Set Based Segmentation

- Mean curvature flow
  - Slightly different behavior in 3D



Level-Set Based Segmentation

- Smooth out corners using mean curvature



Some Viscosity: $F = 1 - .1\kappa$ Less Viscosity: $F = 1 - .01\kappa$ No Viscosity: $F = 1$

Level-Set Based Segmentation

- **Evolving curves**
- Consider a subset Ω of R^2
- Consider an oriented curve, evolving with time

$$\mathcal{C} \subset \Omega,\ \mathcal{C} : \left\{ \begin{array}{l} [0,1] \times \mathbb{R}^+ \longrightarrow \mathbb{R}^2 \\ (q,t) \mapsto \mathcal{C}(q,t) \end{array} \right.$$

  - q = parametrization
  - t = time
- Embed evolving curve C, at any time t,
  as zero level set of a time-varying function Φ(x,t)
  - $\Phi : \Omega \times R^+ \rightarrow R$,   Φ takes negative values inside curve
- Question: how to evolve Φ ?

Level-Set Based Segmentation

- Evolving curves
- Design principle: **Φ (C(q,t), t) = 0**
  - At any time **t**, Φ should take value 0 when evaluated at points located on curve **C(q,t)** at **same time t**
- Differentiate Φ (C(.,.), .) w.r.t. t

1) Φ(C(q,t'),t'') changes because t'' changes, keeping location C(q,t') fixed

2) Φ(C(q,t'),t'') changes because location C(q,t') changes, keeping time t'' fixed

$$\frac{\partial \Phi}{\partial t}(\mathcal{C}(q,t),t) + < \nabla \Phi(\mathcal{C}(q,t),t), \frac{\partial \mathcal{C}}{\partial t}(q,t) >= 0$$

Level-Set Based Segmentation

- Evolving curves

$$\frac{\partial \Phi}{\partial t}(\mathcal{C}(q,t),t) + < \nabla\Phi(\mathcal{C}(q,t),t), \frac{\partial \mathcal{C}}{\partial t}(q,t) >= 0$$

- For a curve where
each point moves in outward normal direction
based on a force of magnitude F

$$\frac{\partial \mathcal{C}}{\partial t}(q,t) = F\vec{n}(q,t)$$

  - where n(q,t) is outward **unit normal** to curve at point parameterized by q at time t
  - But, we know, **n(q,t) = ∇Φ** (C(q,t)) / **|| ∇Φ** (C(q,t)) **||**

Level-Set Based Segmentation

- Evolving curves

$$\frac{\partial \Phi}{\partial t}(\mathcal{C}(q,t),t) + F|\nabla \Phi(\mathcal{C}(q,t),t)| = 0$$

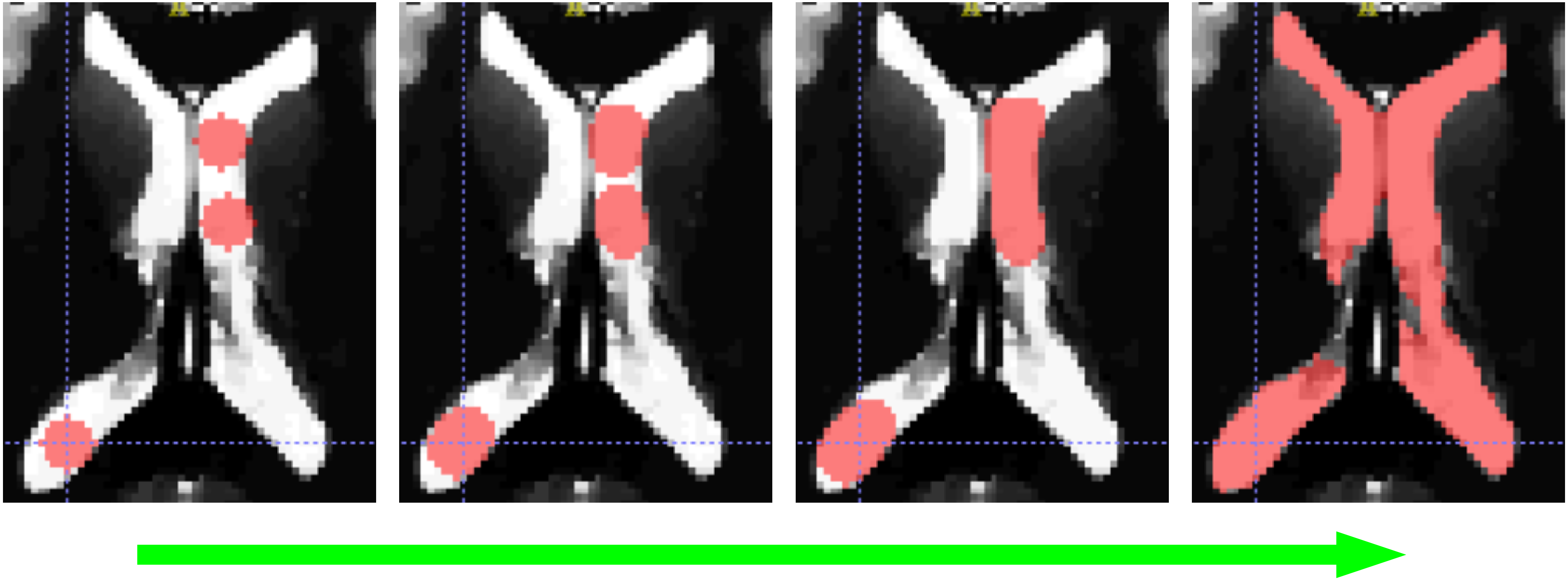
- This equation can be considered as defined on the whole domain $\Omega \times R^{+}$
  - Requires specification of forces F on entire domain (instead of just on one level set)
  - And Φ(.,t) is maintained as a distance transform
- This gives the PDE:

$$\frac{\partial \Phi}{\partial t} + F|\nabla \Phi| = 0, \text{ on } \Omega \times \mathbb{R}^{+}$$

  - with initial condition $\Phi(x,0) = \Phi_0$

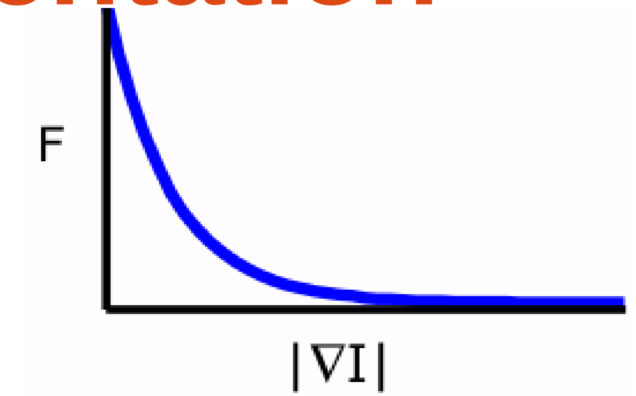
Level-Set Based Segment

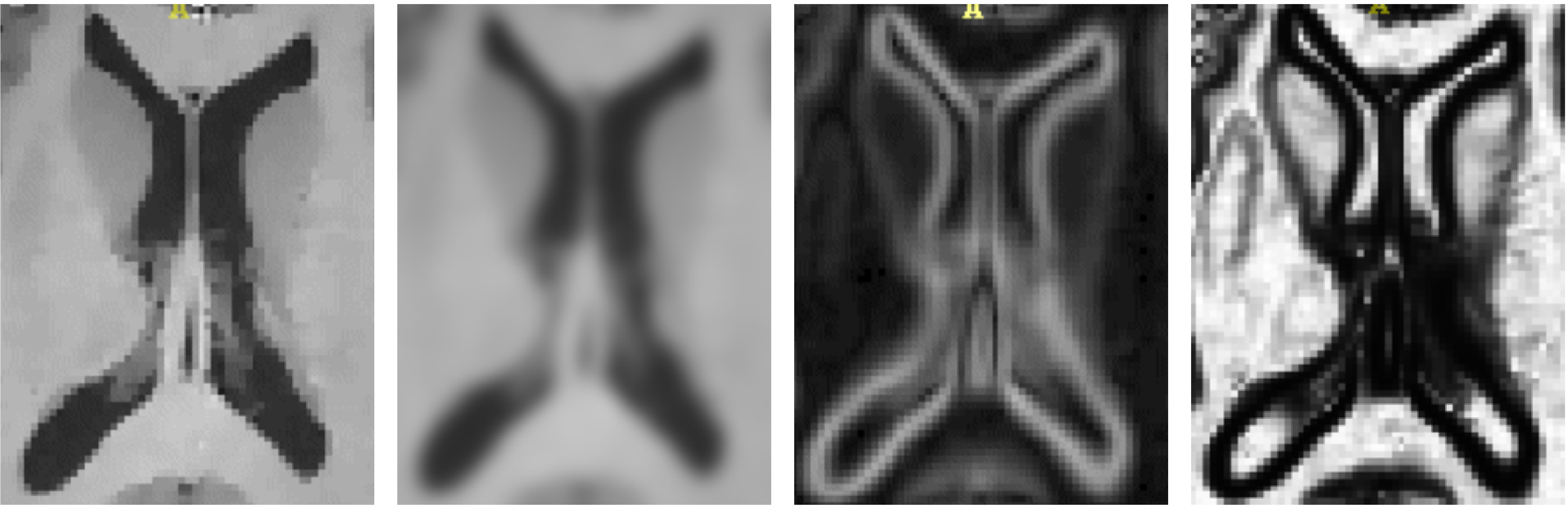
- Anatomical structure segmentation
  - How to make the following happen ?
  - **How to design F(x) ?**



Level-Set Based Segmentation

- Anatomical structure segmentation
  - Designing spatially-varying speed F(x)



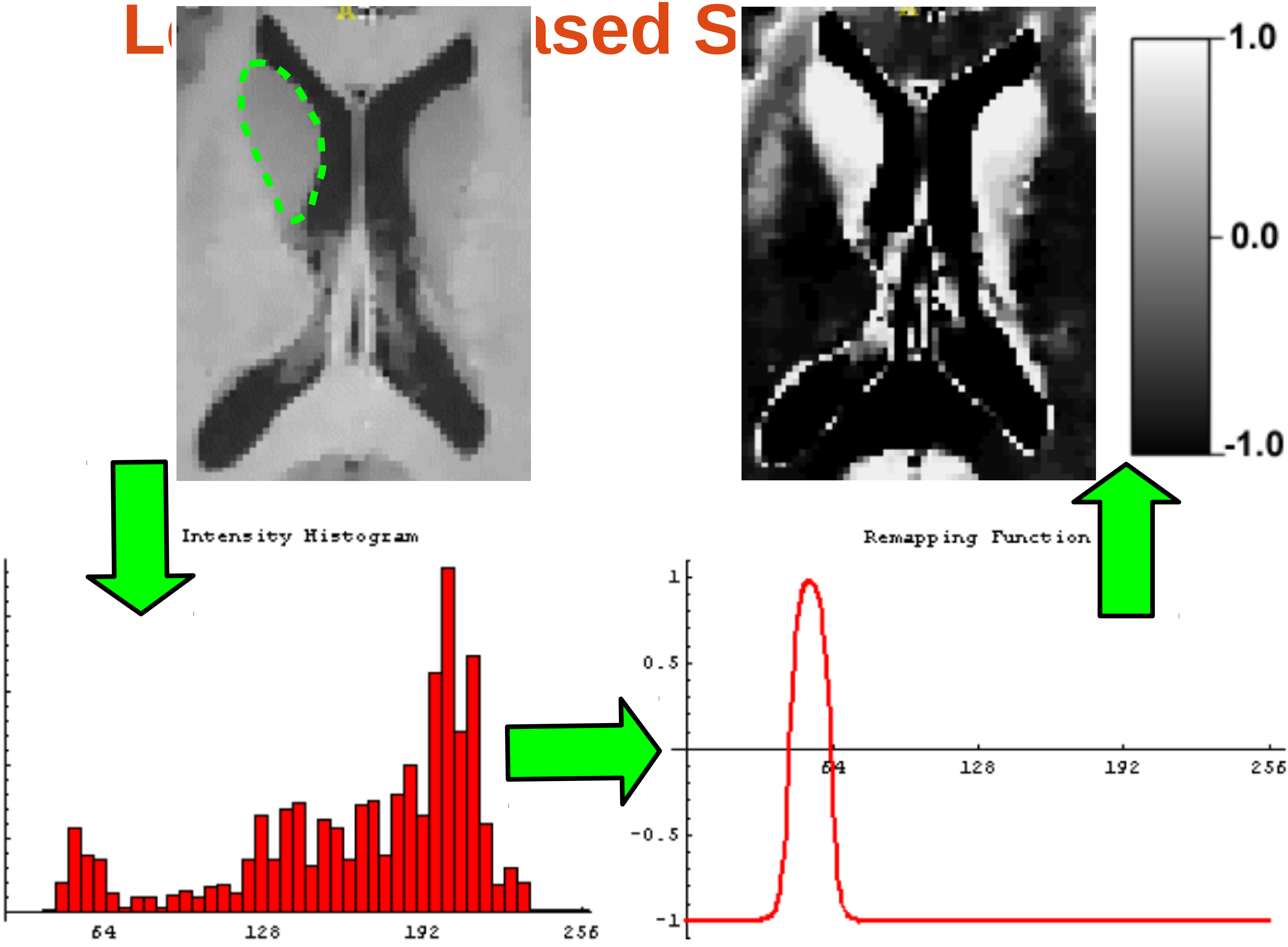


- **What is a limitation of fast-marching based segmentation ?**

Level-Set Based Segmentation

- Limitations of fast-marching segmentation
  - Level-set curve always moves in 1 direction (outwards)
    - $F > 0$, always, for all pixel locations
  - What if we miss an edge ?
- We want the level-set curve to be able to move BOTH outwards and inwards
  - F can be positive and negative
  - Use intensity / texture statistics to design F, instead of using only edge information

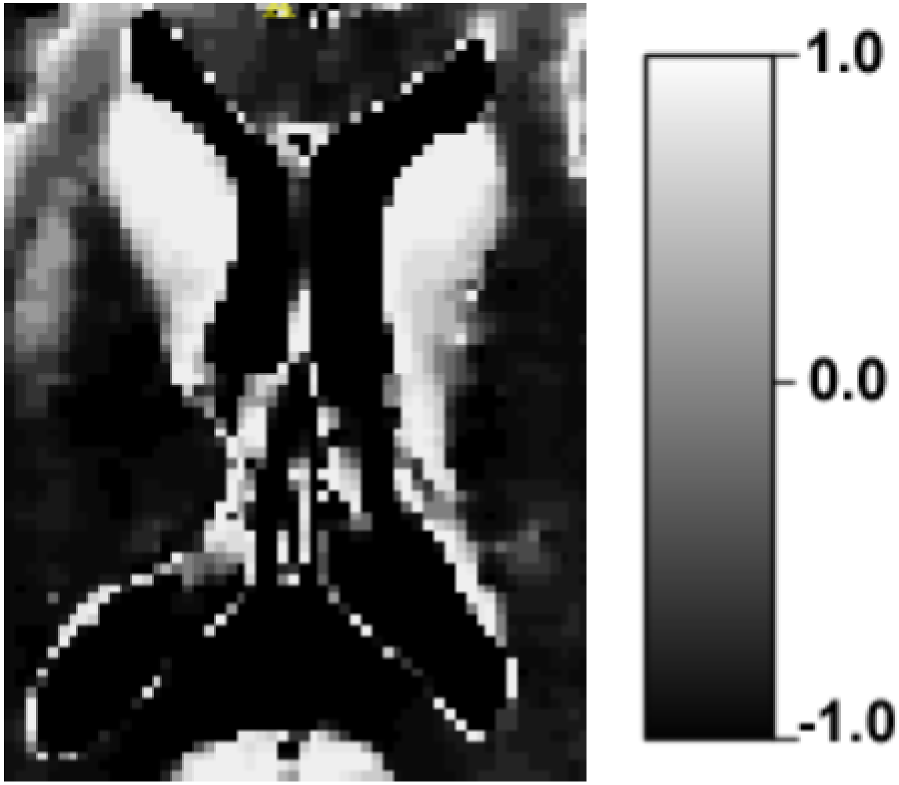
L ased S

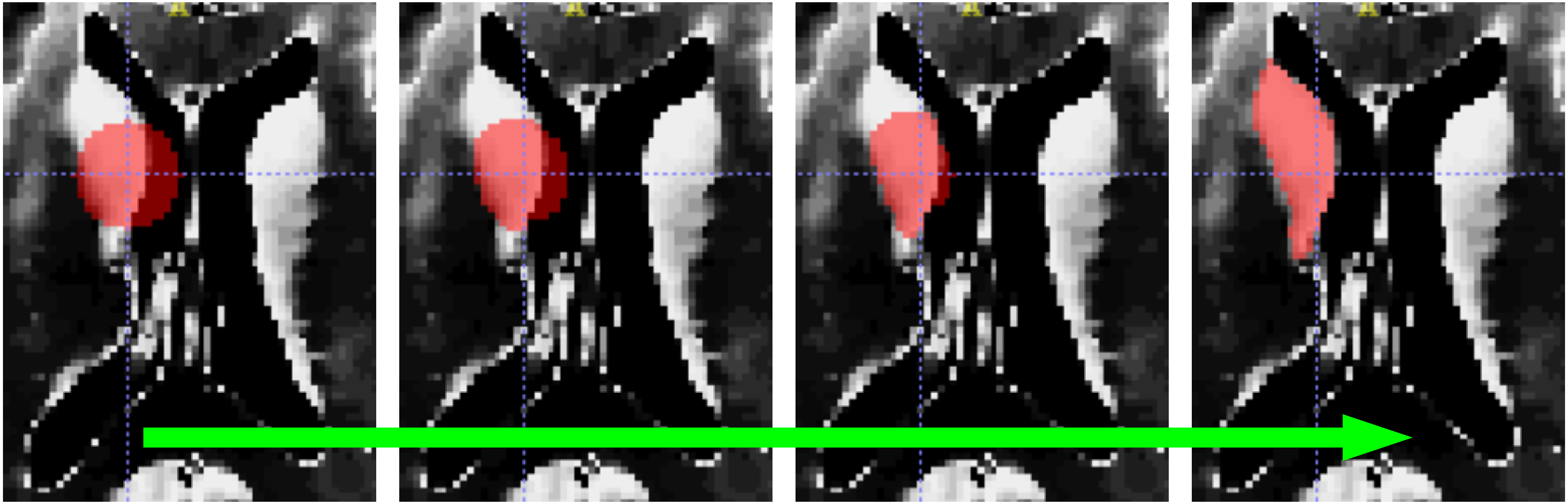


1.0
0.0
-1.0
Intensity Histogram
64
128
192
256
Remapping Function
1
0.5
-0.5
-1
64
128
192
256

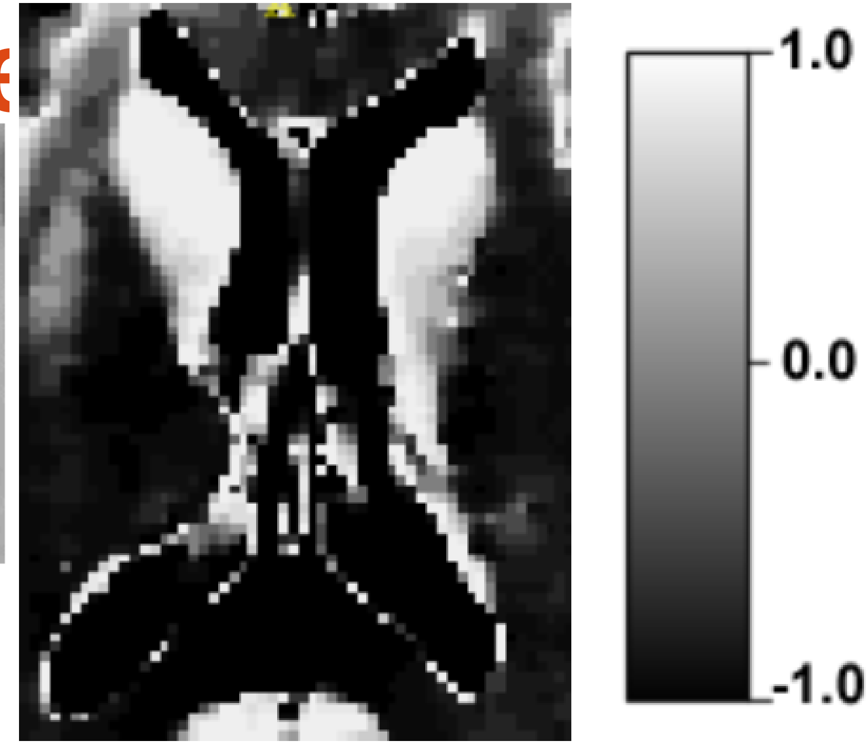
Level-Set Based S

- Level set evolution

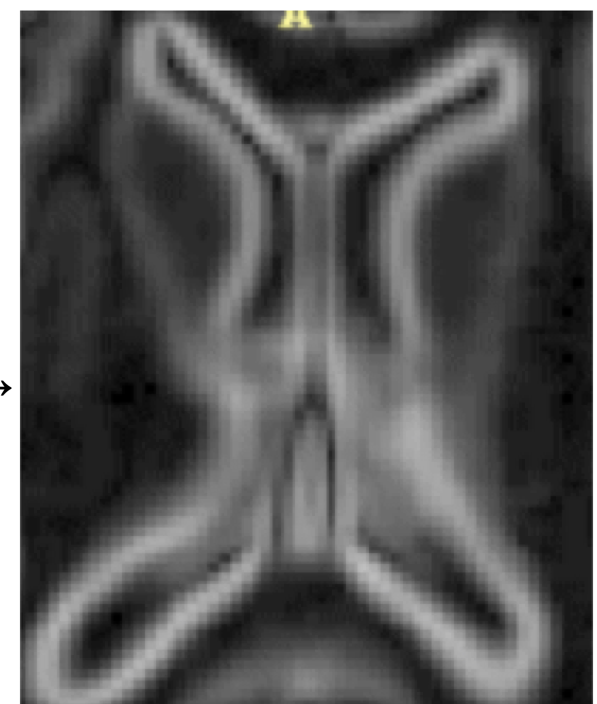




Level-Set Based Se

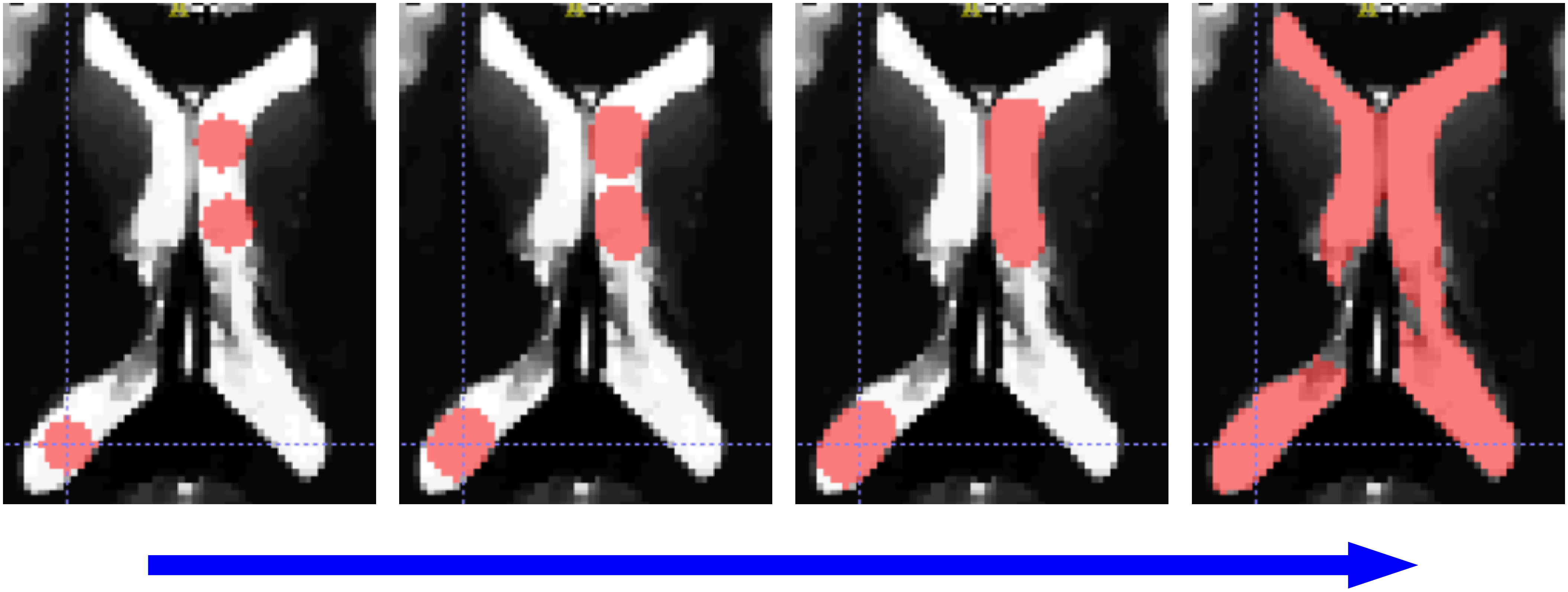


- Types of forces
  - 'Propagation' force
    - Proportional to F(x)
  - Curvature force (MCF)
    - For maintaining smoothness of segmentation
  - 'Advection' force
    - Proportional to inner product of
      (i) level-set normal and
      (ii) gradient of edge (grad mag) image →
    - Can prevent leakage by
      pulling level set to edges



Level-Set Based Segmentation

- Initialization
  - Can be disconnected blobs
    - Merging handled easily



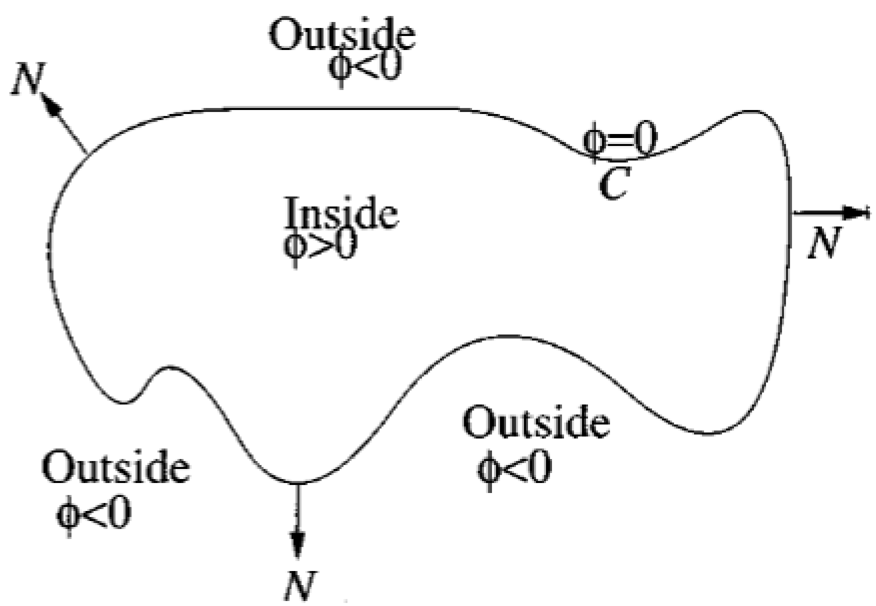
Level-Set Based Segmentation

- **Optimizing segmentation + region-based intensity model**
- Assume 2 regions: foreground (fg) + background (bg)
- Optimize segmentation to minimize:

    Length of boundary

    + (variance of fg * area of fg)

    + (variance of bg * area of bg)



- If $\varphi(x)$ = distance function, then boundary length =

$$\int_\Omega \left|\nabla H\left(\varphi(x)\right)\right| dx = \int_\Omega \delta\left(\varphi(x)\right)\left|\nabla\varphi(x)\right| dx$$

Level-Set Based Segmentation

- Optimizing segmentation + intensity model
- Optimize

$$F(c_1, c_2, \mathcal{C}) = \mu \int_\Omega \delta(\Phi)|\nabla\Phi| dxdy + \lambda_1 \int_\Omega (I(x,y) - c_1)^2 H(\Phi(x,y)) dxdy + \lambda_2 \int_\Omega (I(x,y) - c_2)^2 (1 - H(\Phi(x,y))) dxdy$$

- Given a segmentation Φ(x),
  optimal values for c1, c2 are = ?

Level-Set Based Segmentation

- Optimizing segmentation + intensity model
- Optimize

$$
\begin{aligned}
F(c_1, c_2, \mathcal{C}) = & \mu \int_\Omega \delta(\Phi)|\nabla\Phi| dxdy \\
& + \lambda_1 \int_\Omega (I(x,y) - c_1)^2 H(\Phi(x,y)) dxdy \\
& + \lambda_2 \int_\Omega (I(x,y) - c_2)^2 (1 - H(\Phi(x,y))) dxdy
\end{aligned}
$$

- Given c1, c2, how to modify segmentation Φ(x) ?
- Gradient descent on functional F(.)
  (uses Euler-Lagrange equation)

$$
\frac{\partial\Phi}{\partial t} = \delta_\epsilon(\Phi)\left(\mu \,\mathrm{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right) - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2\right)
$$

Level-Set Based Segmentation

- Optimizing segmentation + intensity model
- Solve a PDE

$$\frac{\partial \Phi}{\partial t} = \delta_\epsilon(\Phi)\left(\mu \operatorname{div}\left(\frac{\nabla \Phi}{|\nabla \Phi|}\right) - \lambda_1 \left(I - c_1\right)^2 + \lambda_2 \left(I - c_2\right)^2\right)$$

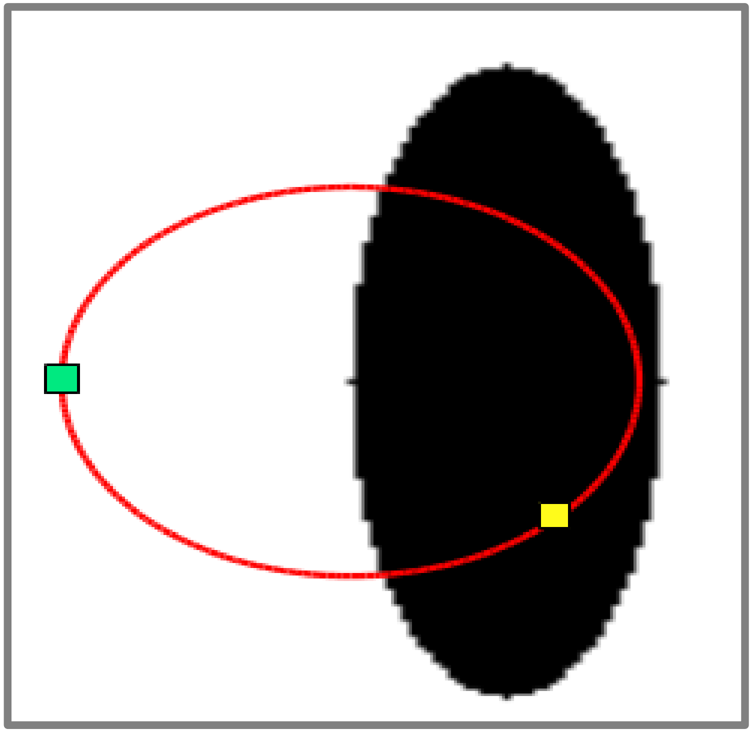
- For surfaces in 3D,
  Divergence of unit normal = mean curvature * (-1)

Level-Set Based Segmentation

- Optimizing segmentation + intensity model

$$\frac{\partial \Phi}{\partial t} = \delta_\epsilon(\Phi)\left(\mu \operatorname{div}\left(\frac{\nabla\Phi}{|\nabla\Phi|}\right) - \lambda_1\left(I - c_1\right)^2 + \lambda_2\left(I - c_2\right)^2\right)$$

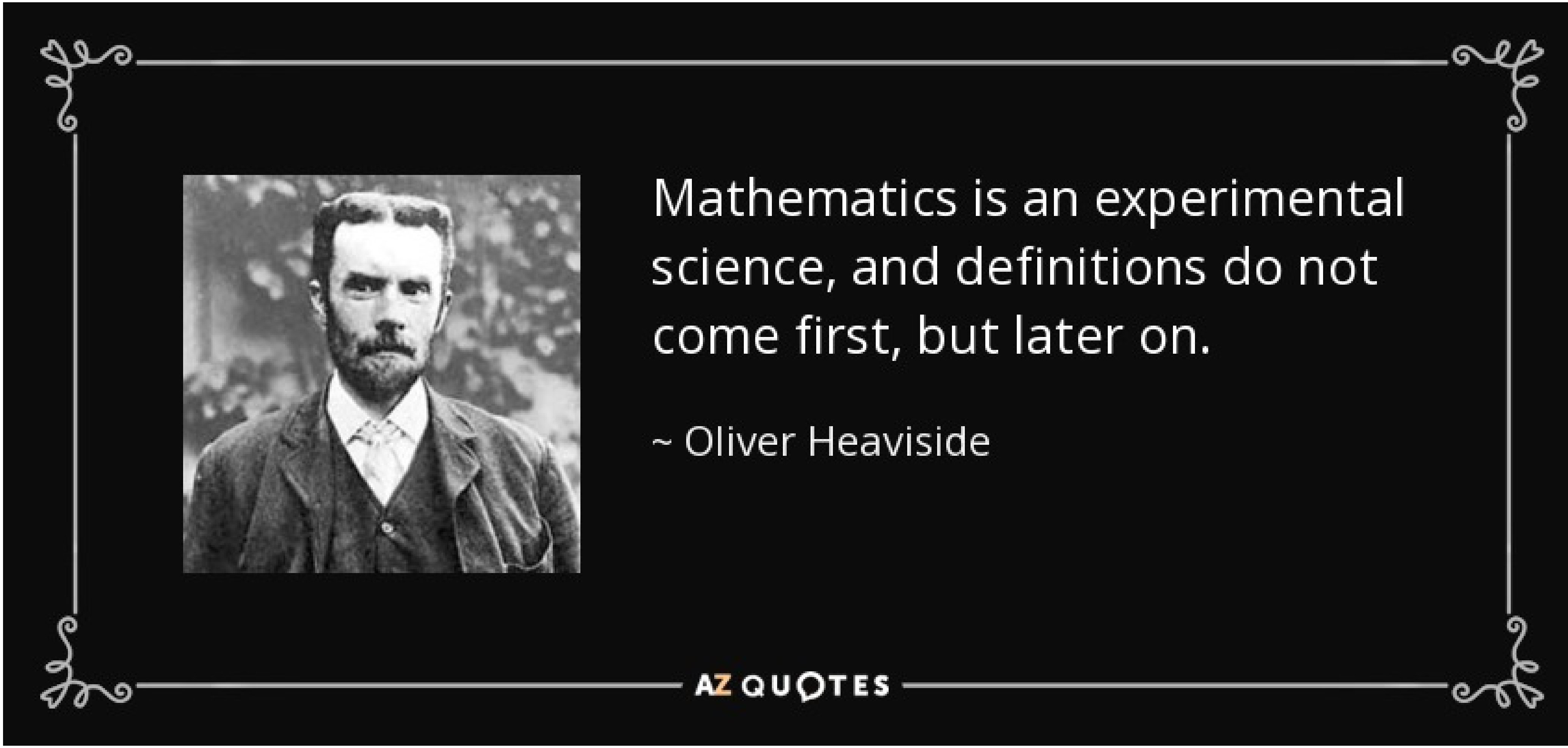
- Example
  - Let c1 : inside : closer to black
  - Let c2 : outside : closer to white
  - **Green** point:
    (I – c2) = small
    (I – c1) = large
    Moves inwards
  - **Yellow** point:
    Moves outwards



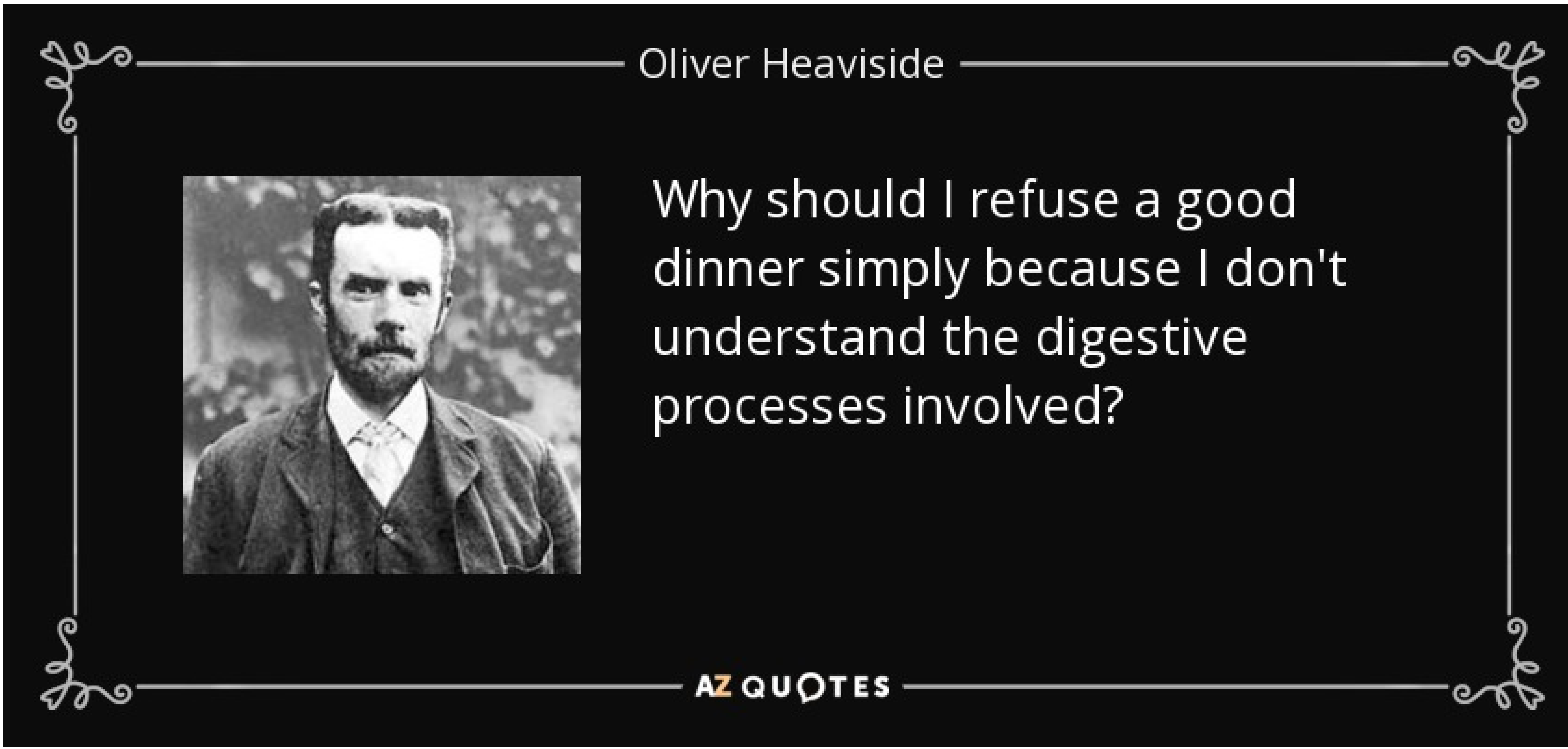
Oliver Heaviside

- Self-taught electrical engineer, mathematician, physicist
- Parents could not keep him at school after he was 16
- 1st winner of Faraday Medal (1922): top medal award by Institution of Engineering and Technology (previously called Institution of Electrical Engineers)
  - Donald Knuth won it in 2011

Oliver Heaviside

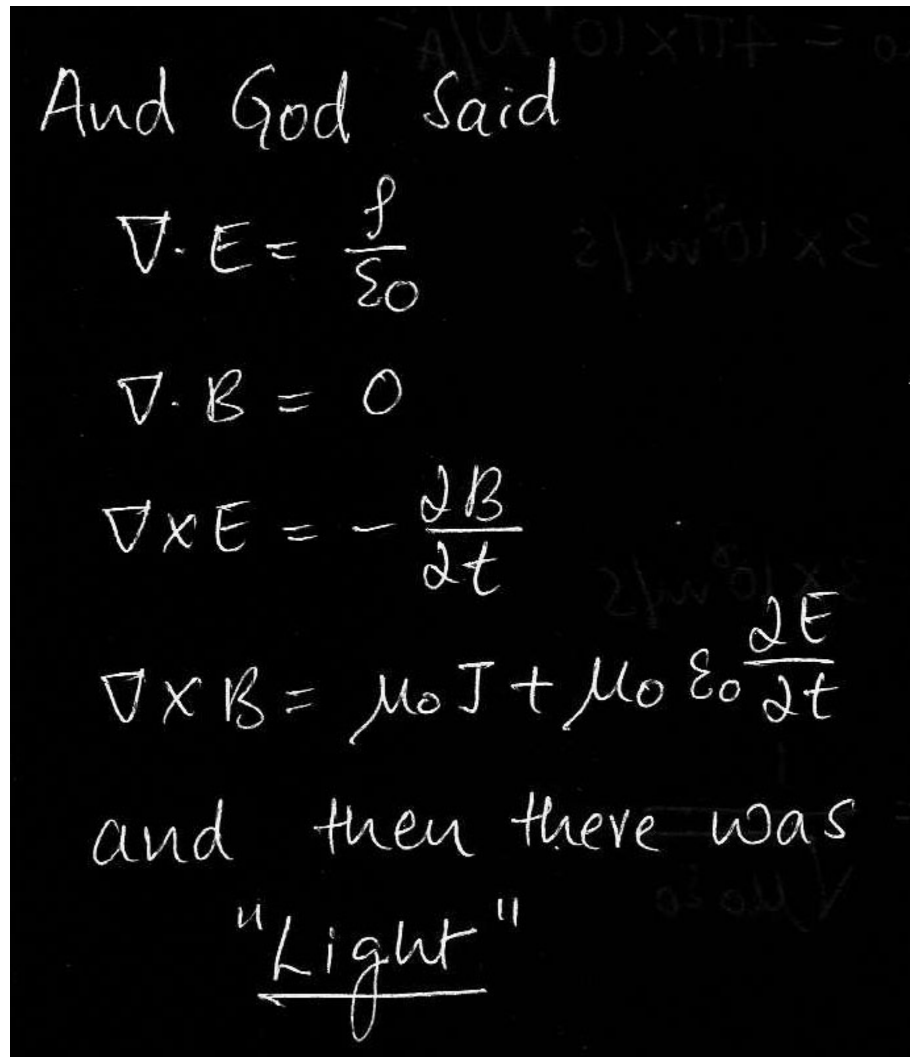


Oliver Heaviside



Oliver Heaviside

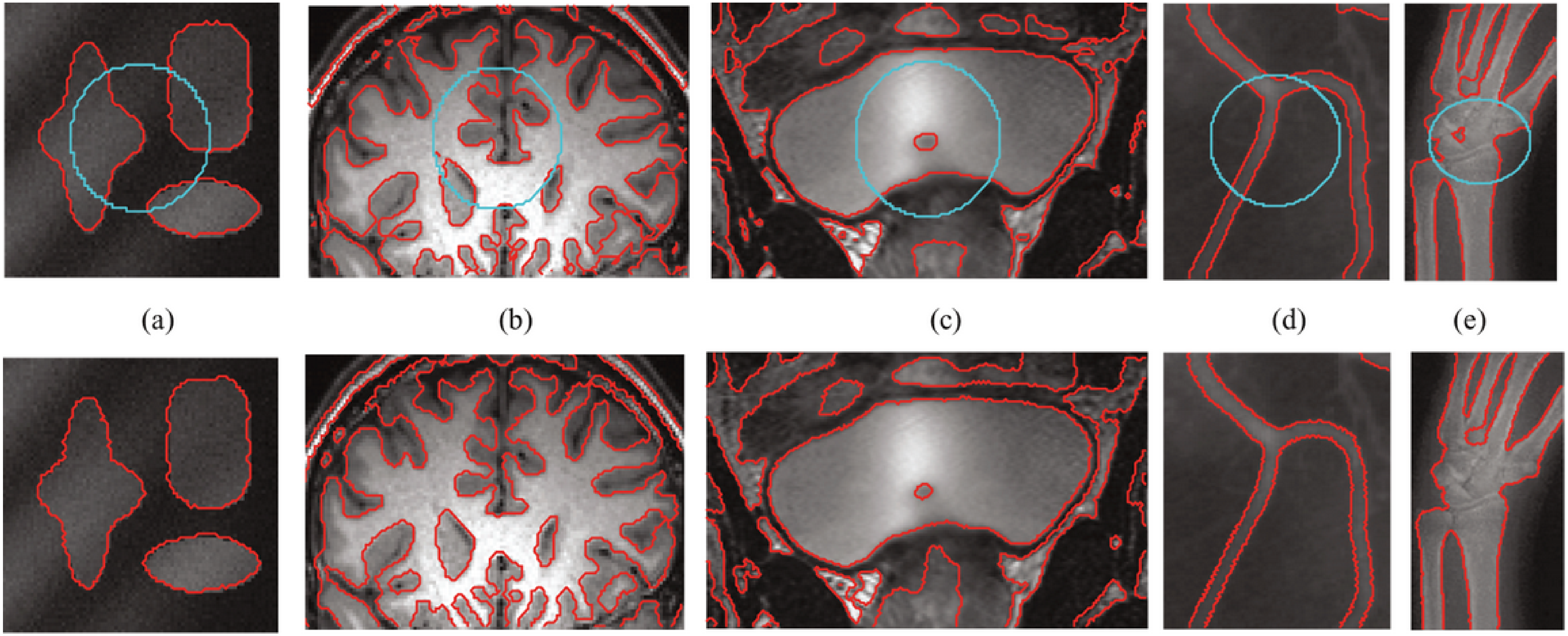
- Maxwell's formulation (1861-62) of electromagnetism consisted of 20 equations in 20 variables.
- Heaviside used curl and divergence operators of vector calculus to reformulate 12 of these 20 equations into 4 equations in 4 variables (B, E, J, ρ), the form by which they have been known ever since



James Watt

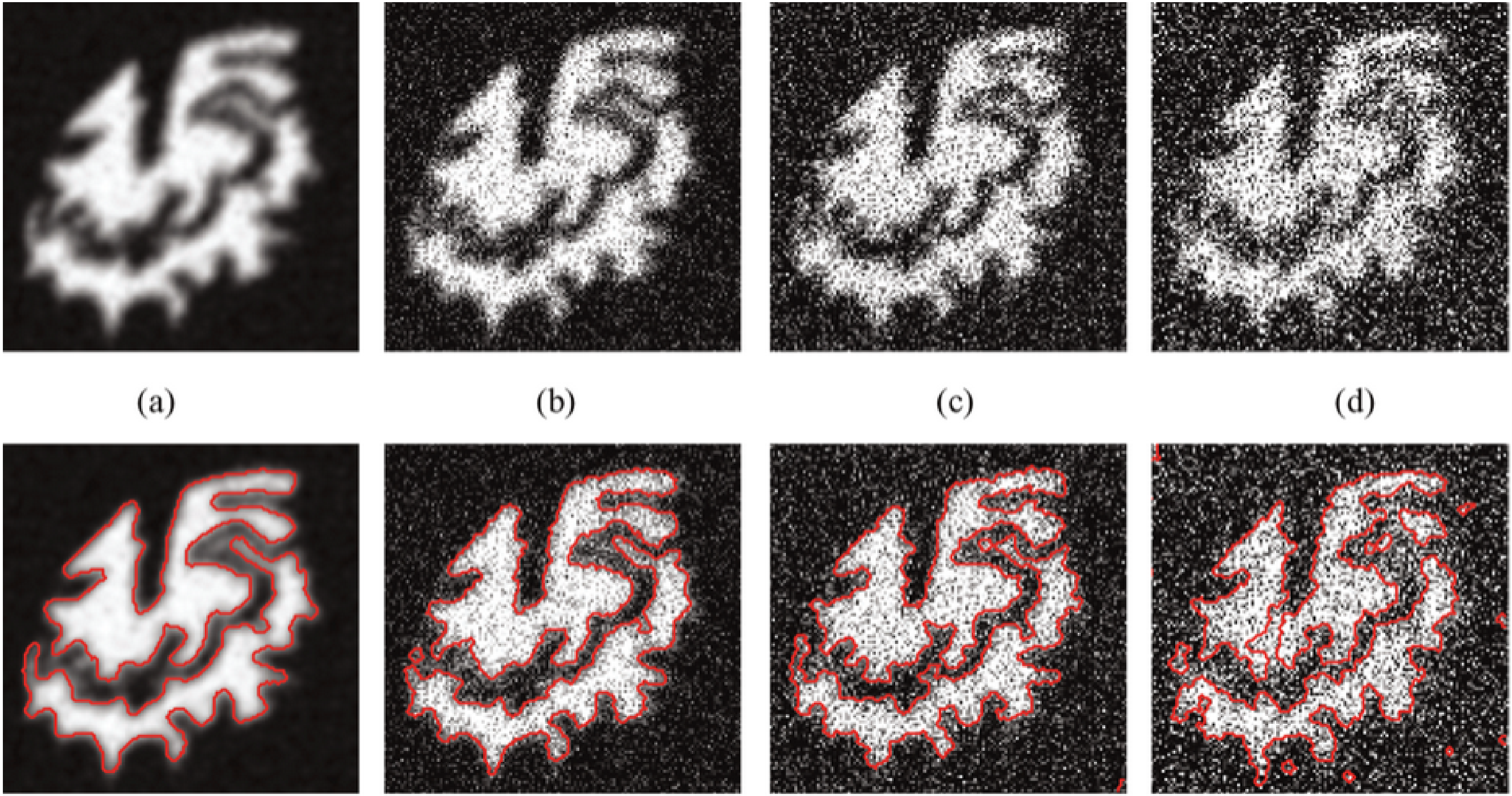
- James Watt built steam engine in 1781
  - Lead to the industrial revolution
- When was theory of thermodynamics developed ?
  - Thermodynamics developed out of a desire to increase the efficiency of early steam engines
  - Carnot, 1824
  - Lord Kelvin, 1854
  - Gibbs, 1873-76, Statistical mechanics / thermodynamics
    - Explained classical thermodynamics as a natural result of statistics, classical mechanics, quantum theory at micro level

Level-Set Based Segmentation



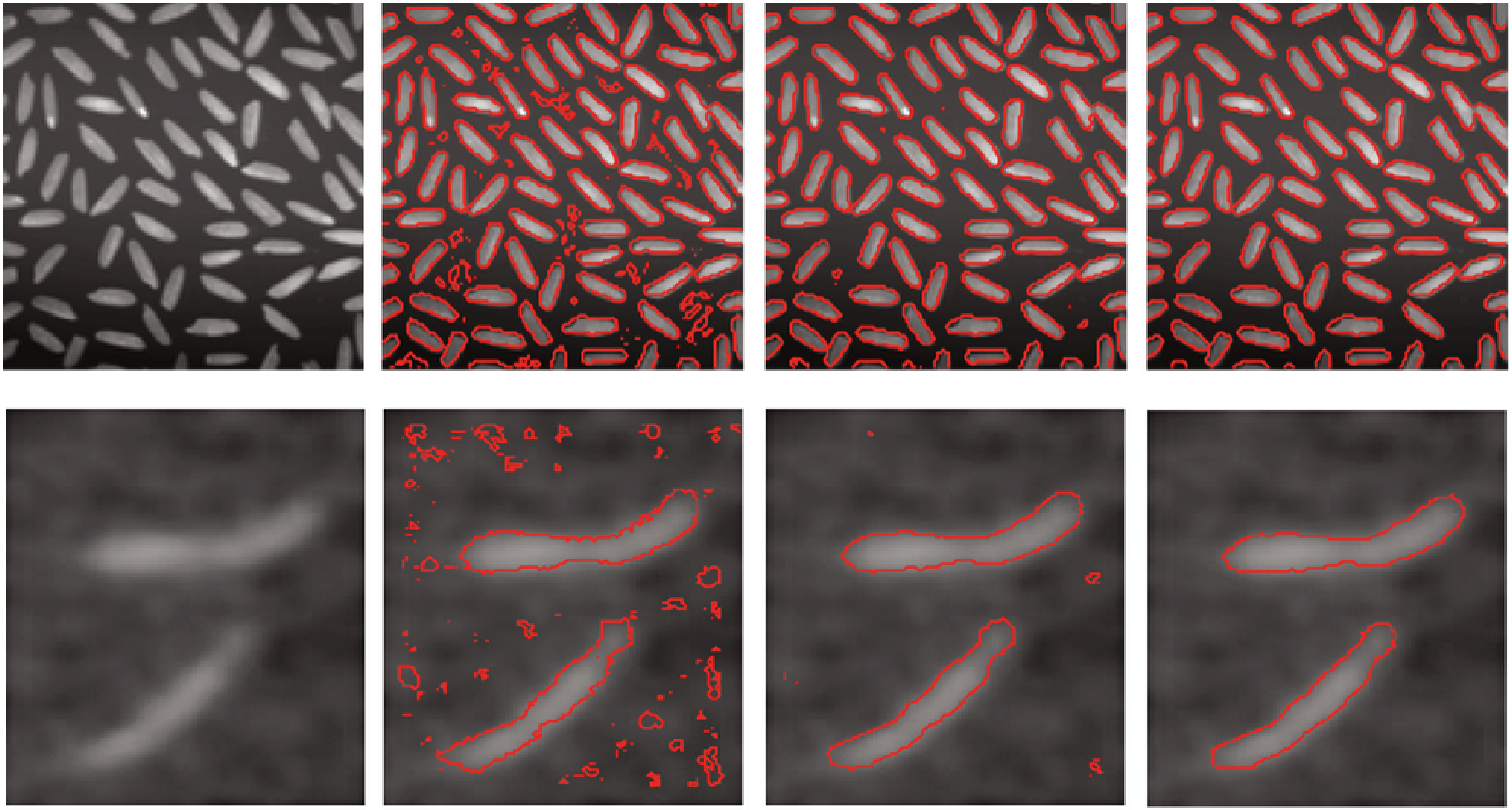
(a) (b) (c) (d) (e)

Level-Set Based Segmentation



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

Level-Set Based Segmentation



Level-Set Based Segmentation

- Increasing regularization μ → ...