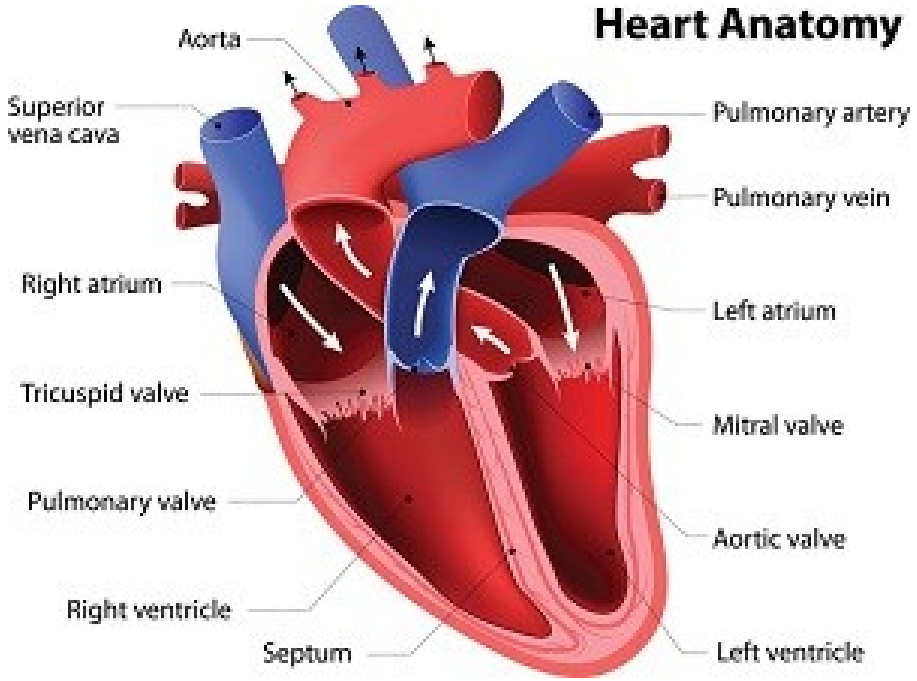
CS463/516

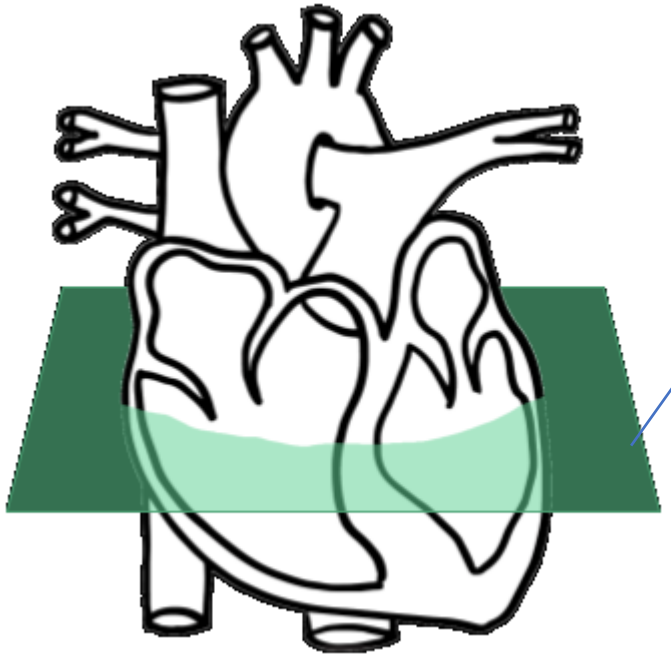
Lecture 14

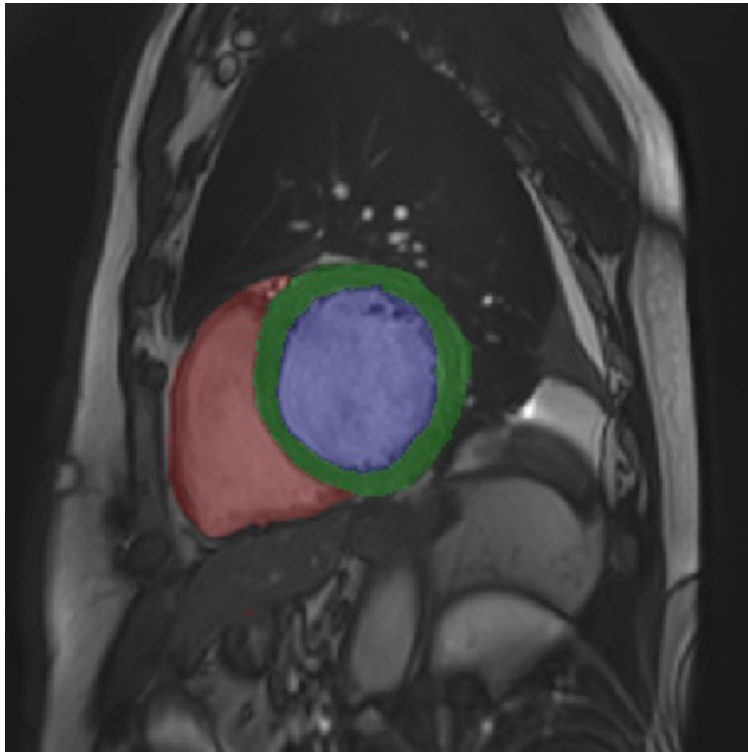
Segmentation

Medical image segmentation

- In medical imaging, segmentation refers to the delineation of specific structures
- Combines *data knowledge* with *domain knowledge* to arrive at result
  - Domain knowledge: information about the objects to be delineated (usually from expert)
  - Data knowledge: assumptions about features (continuity, local smoothness) of image features within segments

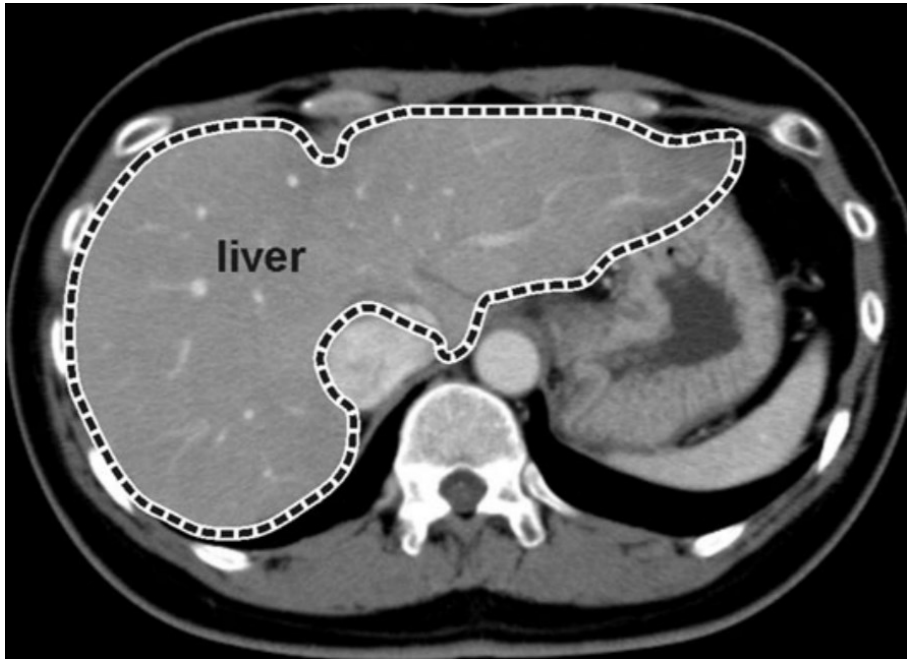






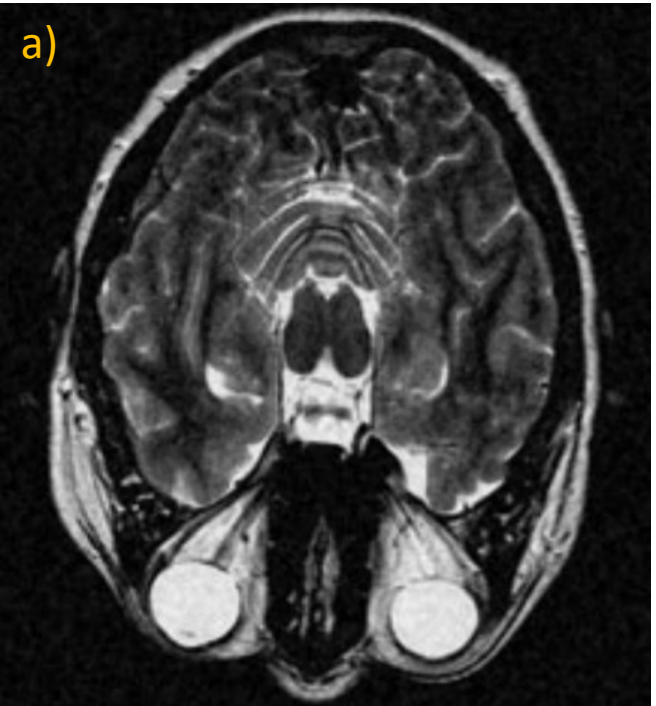
Segmentation strategies

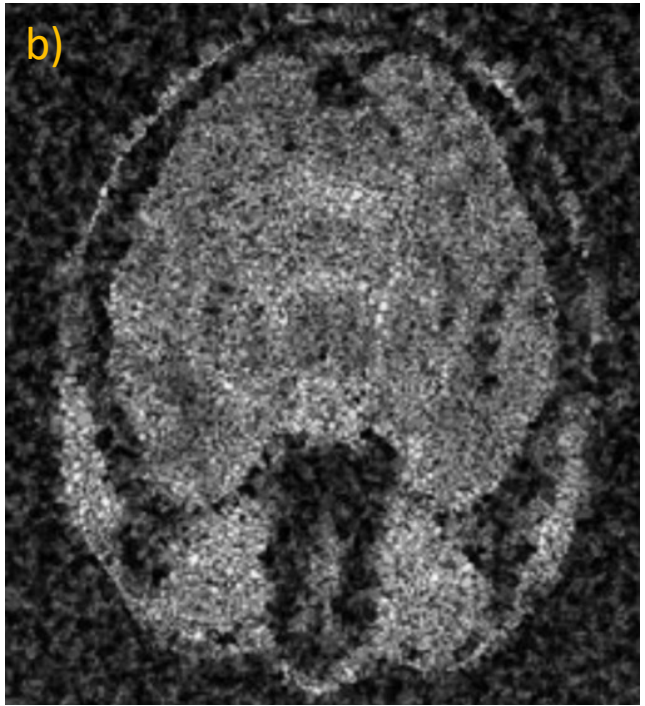
- 1) **foreground segmentation**: focus on a single object in the image,
  - use segmentation criteria that create a good partitioning of foreground objects, quality of background partitioning is irrelevant
  - Requires some model knowledge to be applied after segmentation, for separating foreground segments from the background
  - Example: if only the liver needs to be separated, the segmentation would be successful regardless of errors in regions outside the liver
- Requires some domain knowledge to be applied after the segmentation, for separating foreground segments from background segments
  - Could have a researcher or medical professional point out the foreground segments

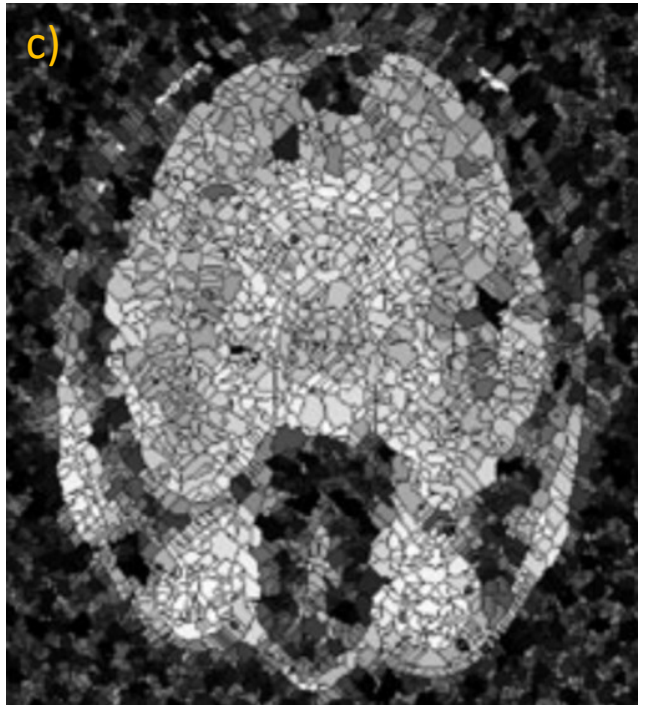


Segmentation strategies

- 2) **hierarchical segmentation**: apply a multi-resolution approach for gradual refinement
  - First segmentation creates segments that are smaller than the smallest object
    - This result sometimes called an ‘over segmentation’
  - Next, some of the small segments are merged into larger segments according to domain knowledge about object appearance
  - Meaningful segments must be definable by common criteria at a single but unknown scale, which is found by analyzing the levels of the segmentation hierarchy





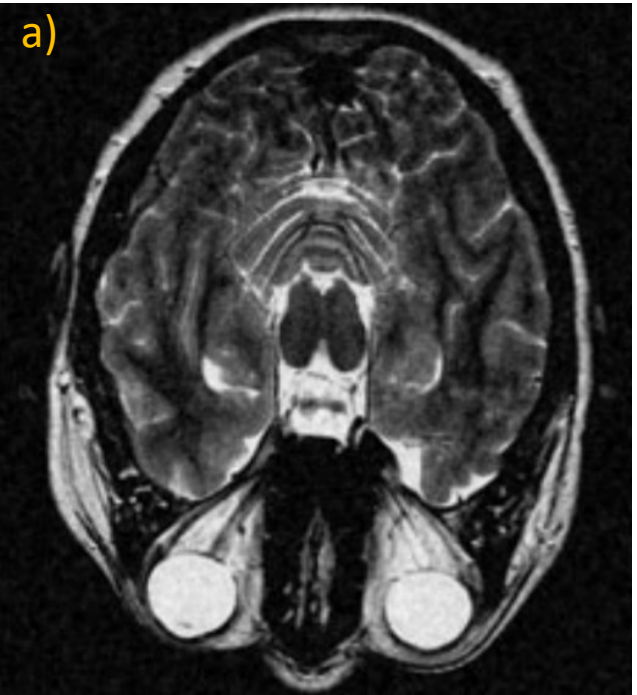


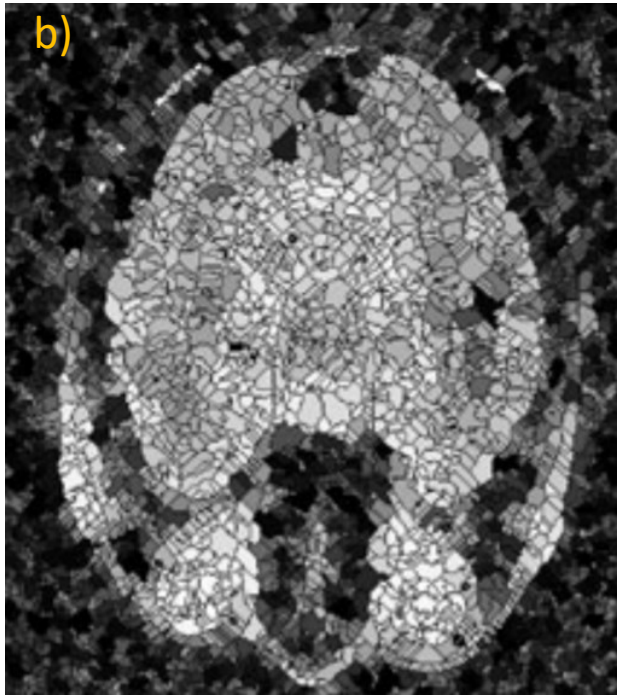
a) Original image
b) Homogeneity constraint 1
c) Homogeneity constraint 2

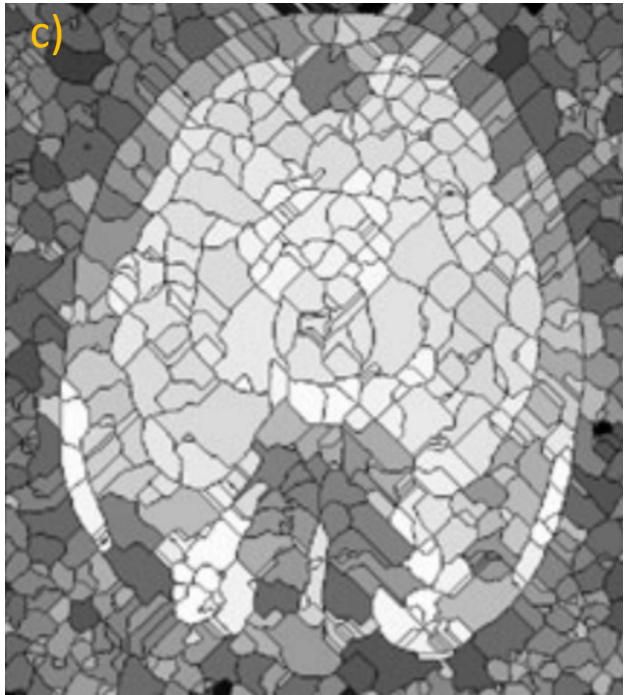
Different homogeneity constraints lead to different size ‘super pixels’

Segmentation strategies

- 3) **multilayer segmentation** – another multi-resolution technique
  - Assumes that a common segmentation criteria exists, but its scale may vary throughout the image
    - Example – structured texture of objects in photo where scale varies with distance from camera
  - Segmentation carried out at different scales, producing *layers of segments*
  - More general than hierarchical segmentation, but analysis of segments is more difficult because an appropriate scale for every segment must be established independently from other segments







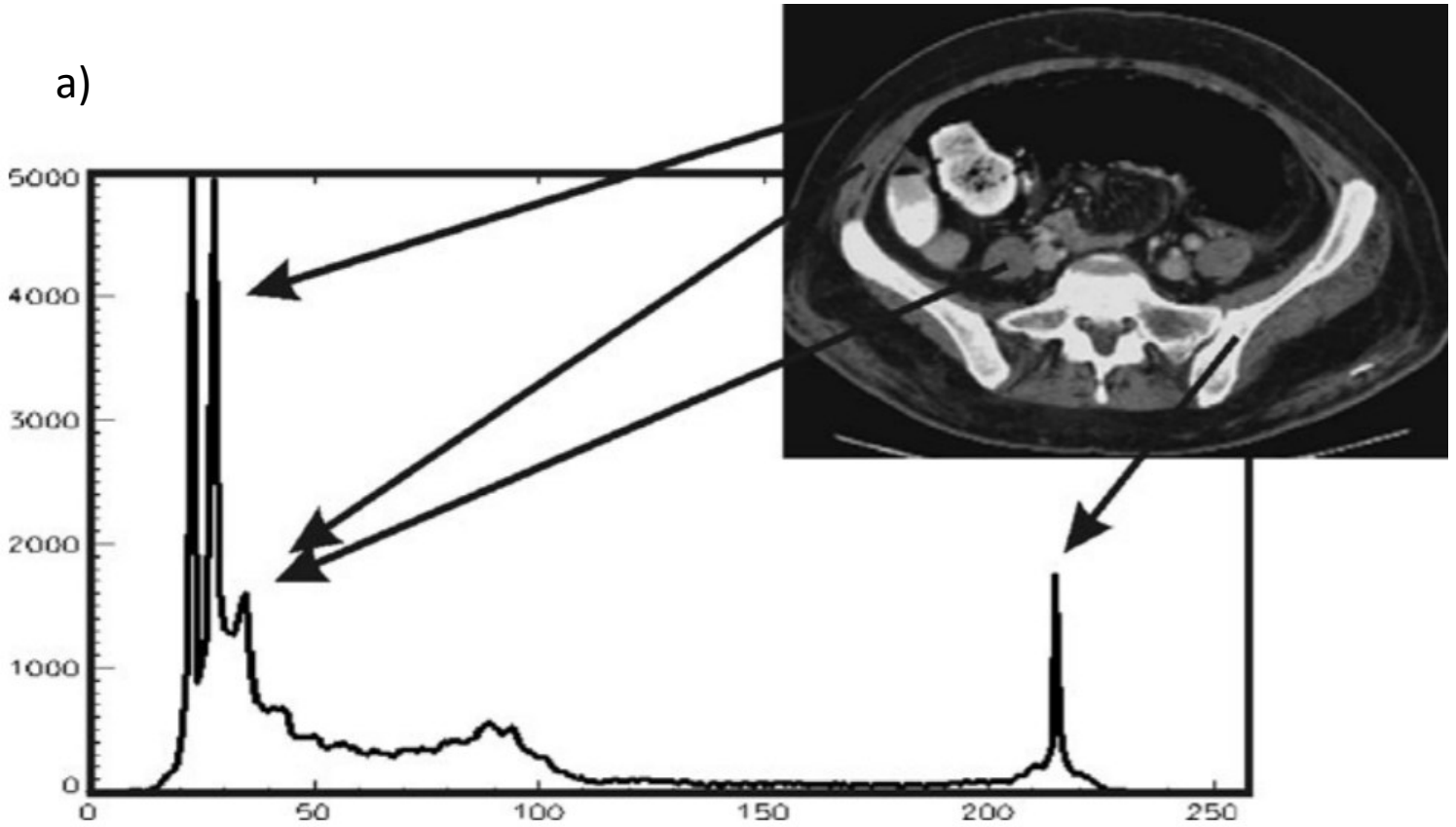
Multilayer segmentation: segmentations at different levels of resolution are created and evaluated in parallel

a) Original image
b) Boundary between fat and bone captured well in high resolution image
c) Boundary between gray matter and CSF captured well in low resolution image

Medical image segmentation

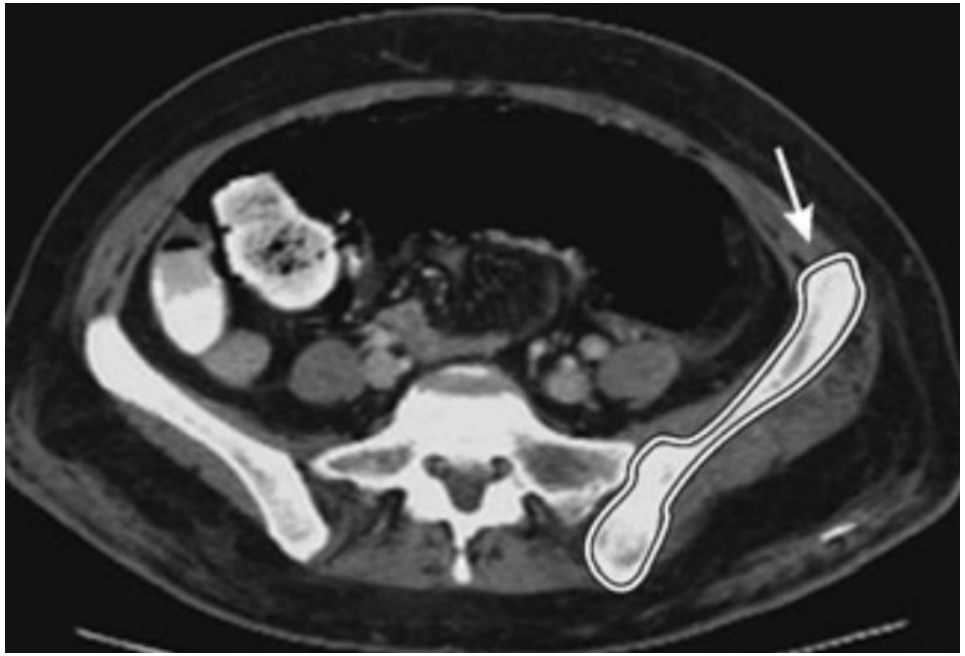
- Often (but not always) the pixel value in a medical image is directly related to diagnostic question we wish to address
  - The acquisition technique was chosen because it offers insight into some diagnostic question

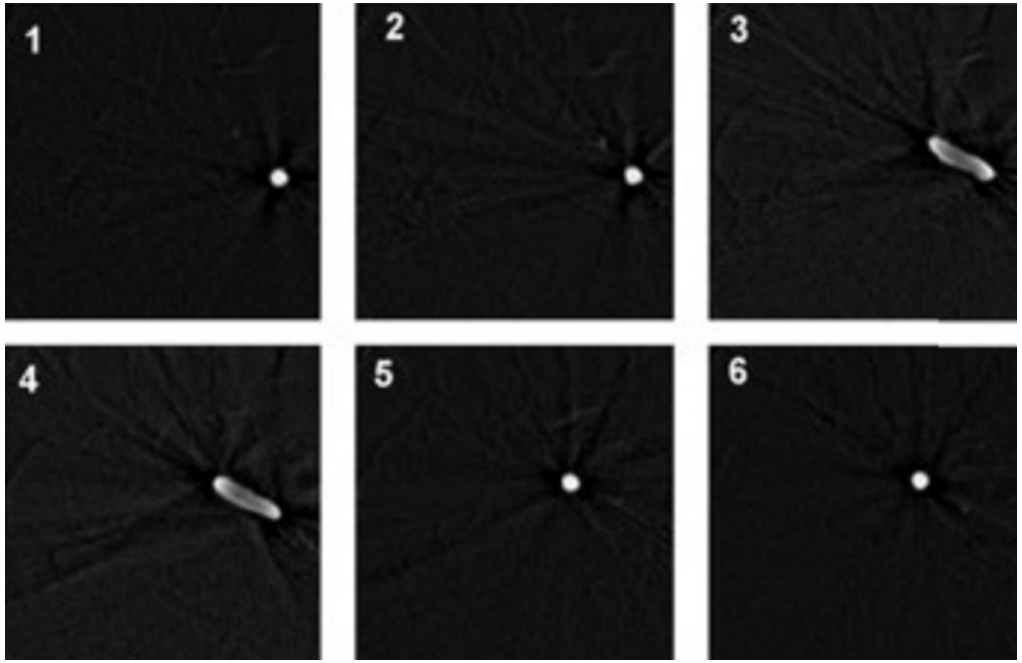
- Example: in some images, such as X-ray or CT, pixel value and membership to an organ class are related (a)
- Hence, can easily apply a threshold to pixel values to segment the structure of interest



Data knowledge

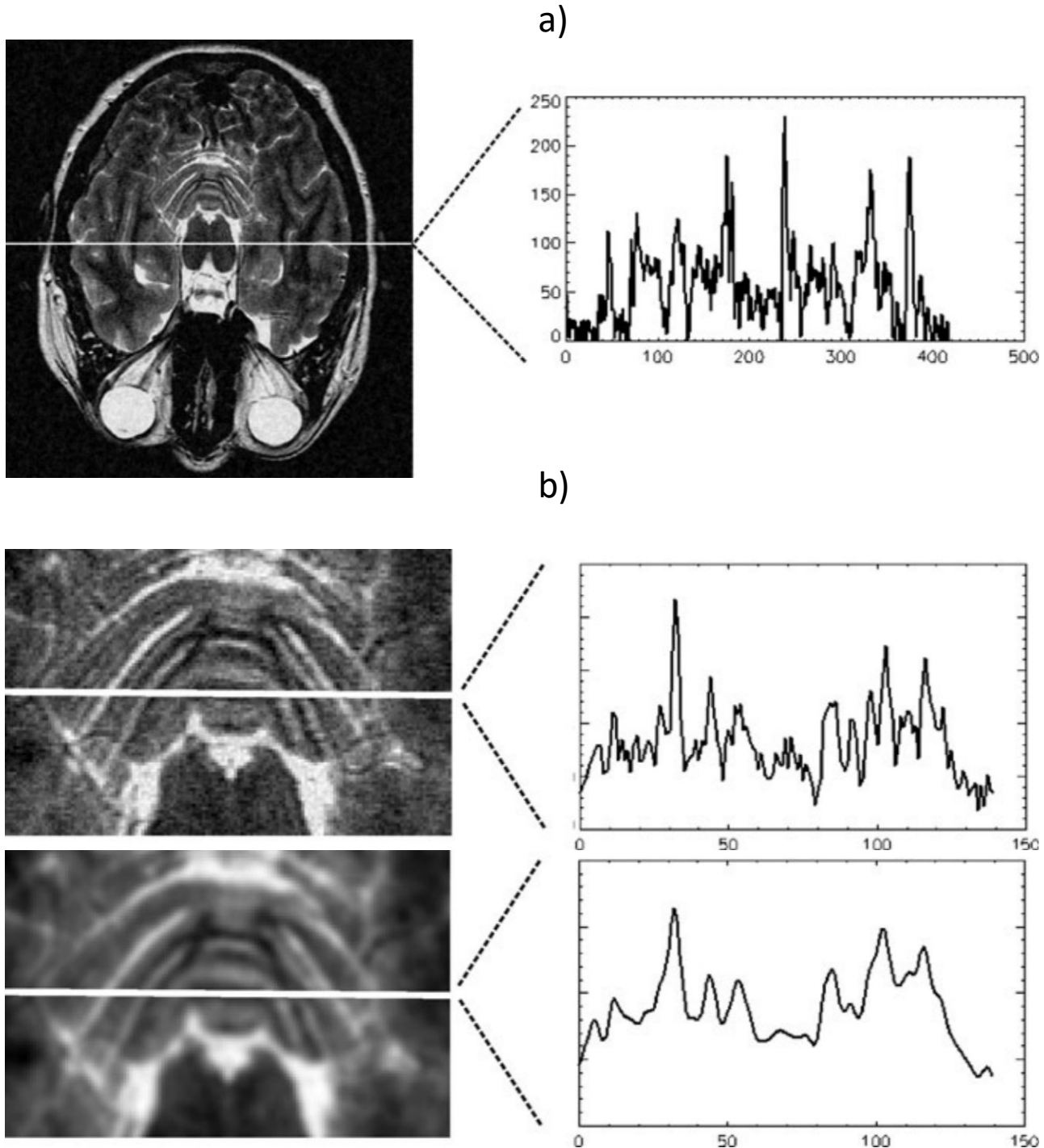
- *Spatial or temporal continuity* is the main data property used for segmentation
- an observable object is assumed to stand out in an image by some homogenous intensity, or texture within a region
- Segmentation based on spatial continuity partitions an image such that homogeneity within segments is larger than between adjacent segments
- a) smooth course of the hip bone boundary is segmented using a spline curve
- b) predicting vessel locations in the sequence of slices by using location from previous slice for initialization





Homogeneity of intensity

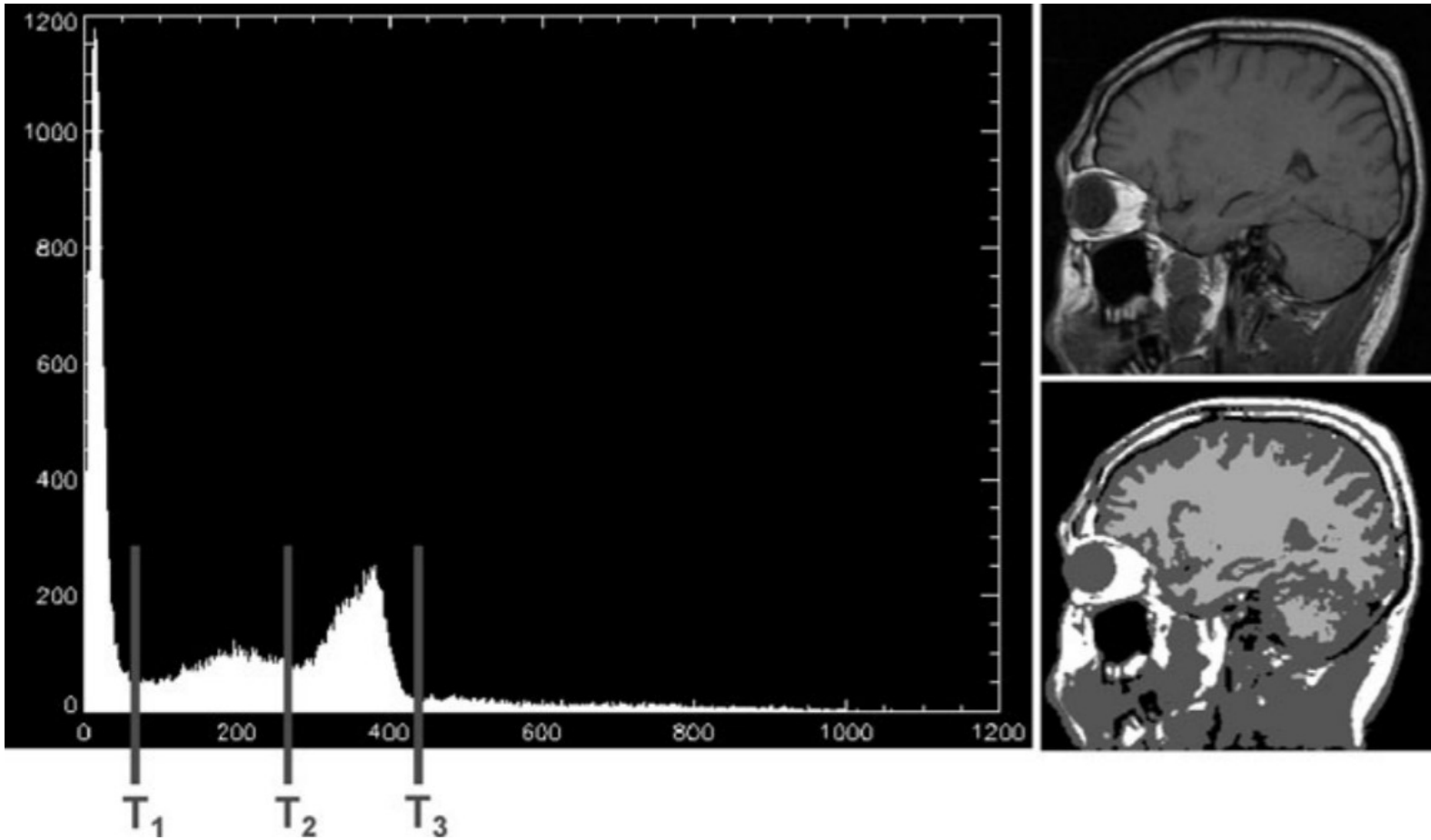
- Pixel intensities of a structure of interest often vary little throughout the structure
  - In consequence, intensity-based segmentation schemes are quite popular
- However, a number of artifacts must be accounted for (primarily noise)
  - a) segmentation in noisy image is more difficult than it seems
  - b) noise reduction may take place before segmentation, but this may cause small details to be removed, which is critical if we want to extract small structures
- Another problem is *shading* (gradual variation in image intensity) which can be removed in several ways



Thresholding

- Often, object of interest stands out with higher intensity than background
- This makes thresholding an often-used tool in segmentation of medical images

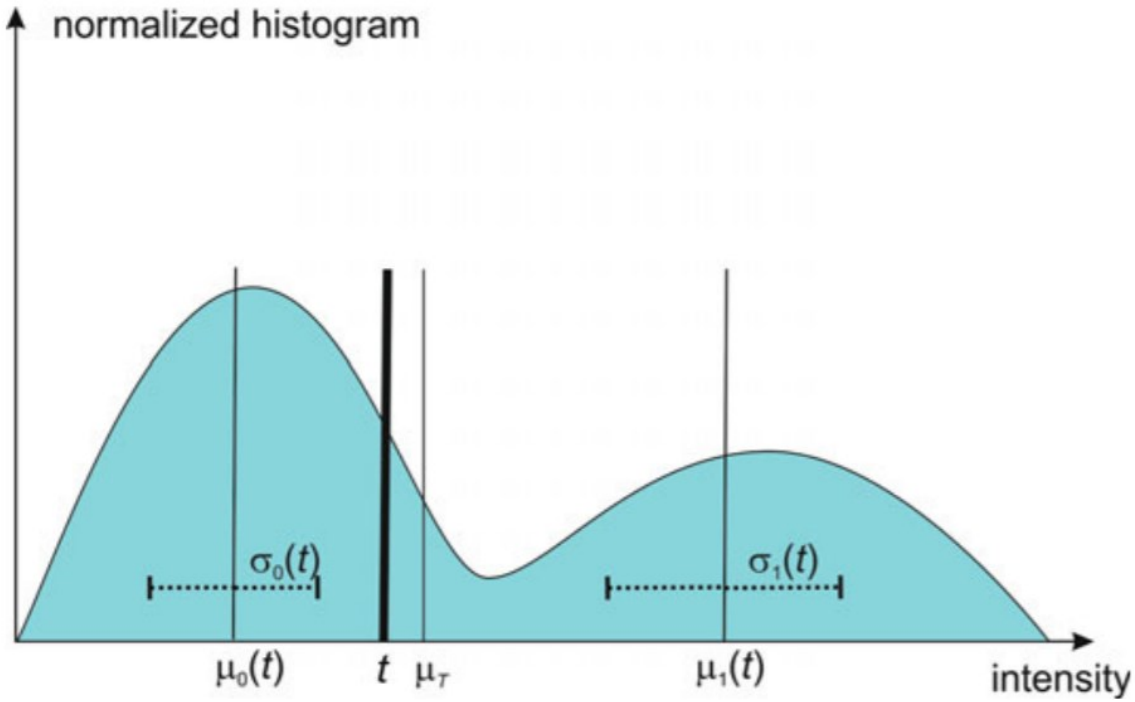
- Segmentation s of an image f by threshold t at voxel $\boldsymbol{v}$ is given by:
- $s(\boldsymbol{v}) = \begin{cases} 1, if\ f(\boldsymbol{v}) > t \\ 0, otherwise \end{cases}$
- Produces segmentation of image pixels into foreground pixels ($s = 1$) and background pixels ($s = 0$)
- If more than one threshold chosen, separation into several different regions is possible:
- Example: T_1, T_2, T_3



Thresholding: Otsu's method

- Otsu presented an iterative procedure to find an optimal threshold for a normalized histogram h
- Produces a threshold t between two classes (0 and 1) such that the variance within each class (q_w) is small and the variance between classes (q_b) is large

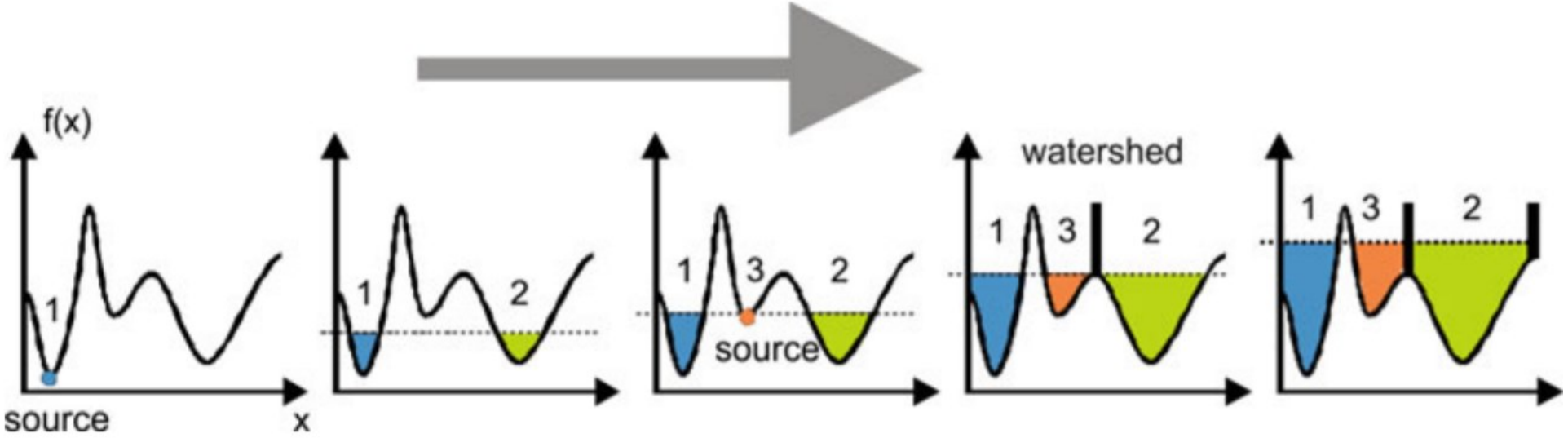
- To do so, maximizes the ratio $\lambda(t) = q_b(t)/q_w(t)$ (a)
- Within-class scatter $q_w(t)$ is computed from gray-level variance $\sigma_0^2(t)$ for all pixels with gray value below t (class 0) and the gray level variance $\sigma_1^2(t)$ from all pixels with gray value greater than or equal to t (class 1)
- Variance of class 0 is $\sigma_0^2(t) = \sum_{g=0}^{t-1} p(g)\big(g - \mu_0(t)\big)^2$, where μ_0 is expected value of all gray levels $< t$
- Within class scatter: $q_w(t) = P_0(t) \cdot \sigma_0^2(t) + P_1(t) \cdot \sigma_1^2(t)$ where $P(0) = \sum_{g=0}^{t-1} p(g)$ and $P(1) = 1 - P(0)$ are the prior probabilities of a pixel belonging to one of the two classes
- Between-class scatter is then the weighted sum of variances of the expected values of the 0-class and 1-class from the expected value μ_T of the histogram:
  $q_b(t) = P_0(t) \cdot (\mu_0(t) - \mu_T)^2 + P_1(t) \cdot (\mu_1(t) - \mu_T)^2$



- Optimal threshold found by searching through all possible values of t.
- Can also maximize $\eta(t) = q_b(t)/\sigma_T^2$ (only requires computing between-class scatter, because variance σ^2 of histogram doesn't depend on t

Watershed transform

- Watershed transform (WST): a popular method that uses edge information as criteria to separate different segments in an image
- Defines segment boundaries by the zero-crossing of the Laplacian of the image
  - Zero-crossings have been shown to be the important features for analysis by humans
- Watershed transform to compute segments is defined as follows:
  - 1) image is treated as a landscape where function values (gradient length) represent height
  - 2) each local minimum in the landscape is a ‘basin’
  - 3) watersheds in landscape are boundaries in the terrain separating regions that drain into different basins
- An image will decompose into as many segments as there are basins



Watershed transform is carried out by ‘flooding’ the image from sources at local minima in the image

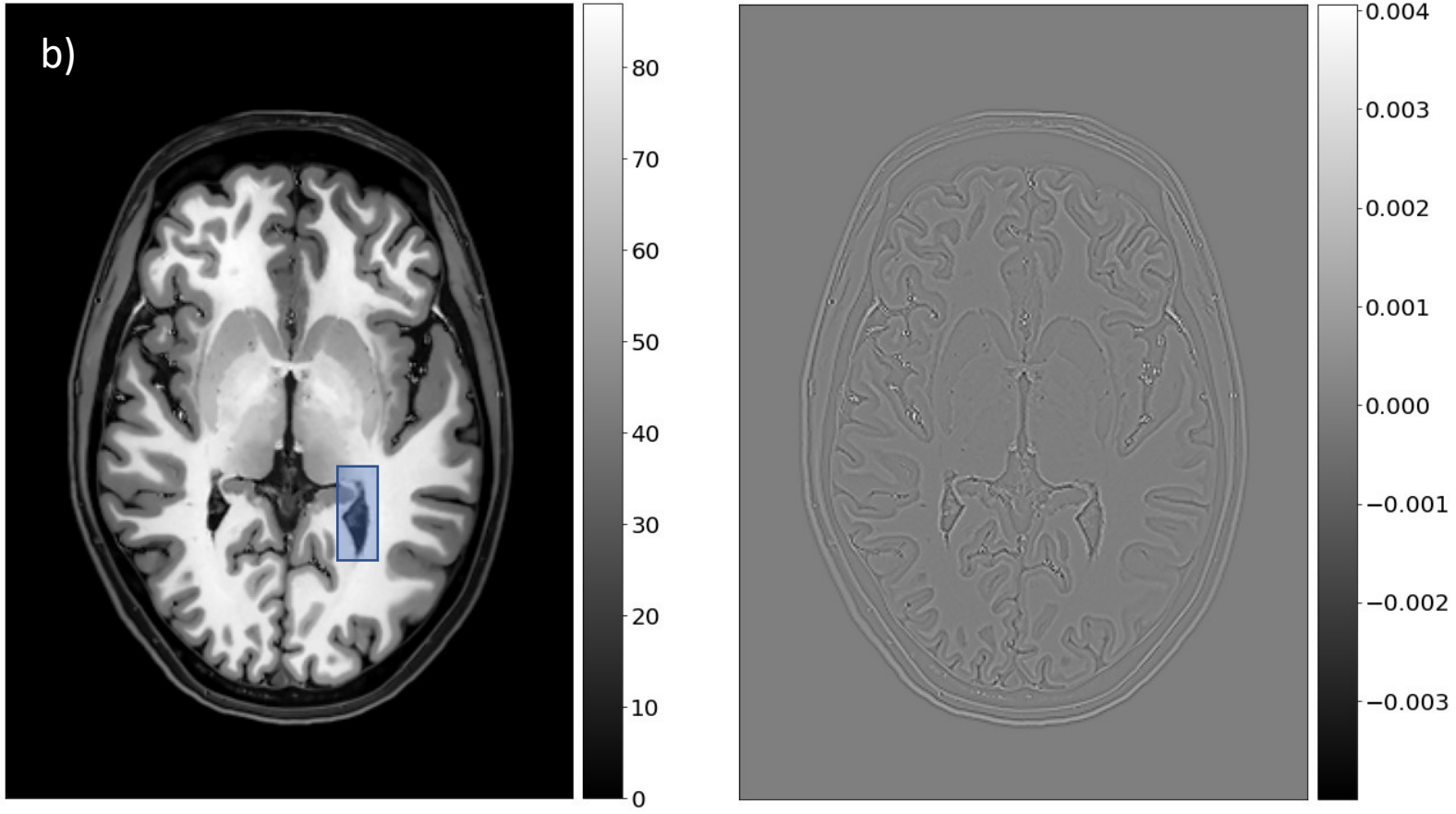
Laplacian

a)

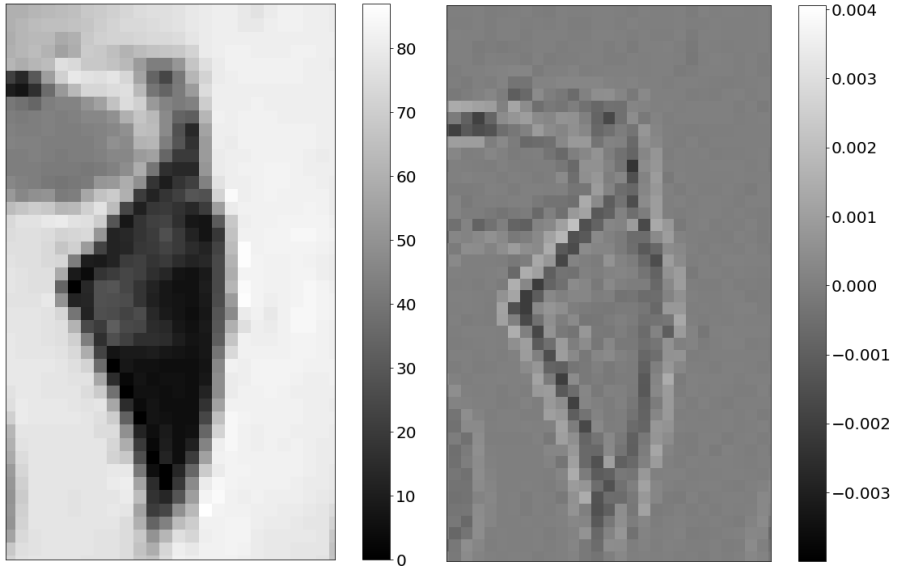
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

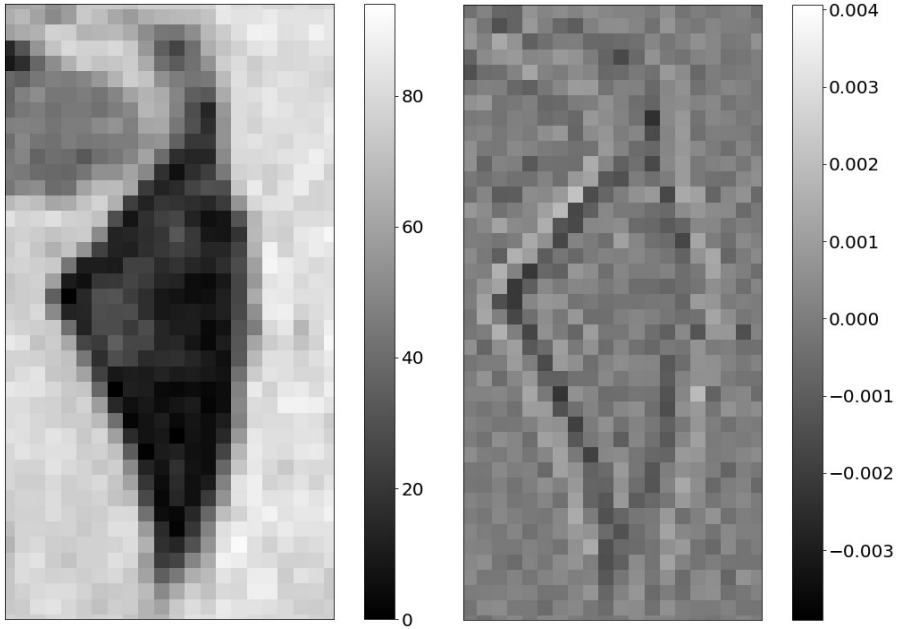
- a) two commonly used laplace filters
- b) denoised T1 + Laplacian of denoised T1
- c) zoom in of ventricle and Laplacian on denoised T1
- d) zoom in of ventricle and Laplacian on non-denoised T1



c)



d)



Zero-crossings easier to find on denoised image

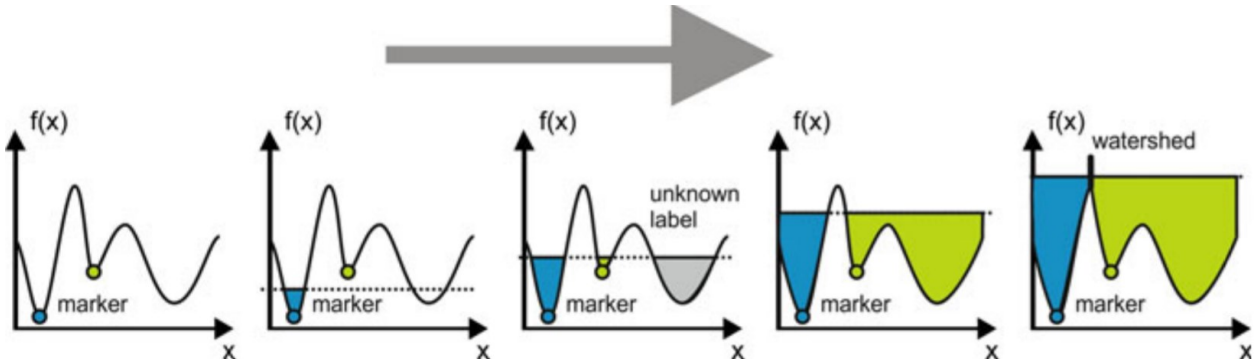
Watershed transform (WST)

- If image contains noise, WST will produce severe over-segmentation because there will be many local minima
- WST algorithm:

set the initial flood level for an image f to $l=\min_{\mathbf{v}}(f(\mathbf{v}))$
while $l<\max_{\mathbf{v}}(f(\mathbf{v}))$ **do**

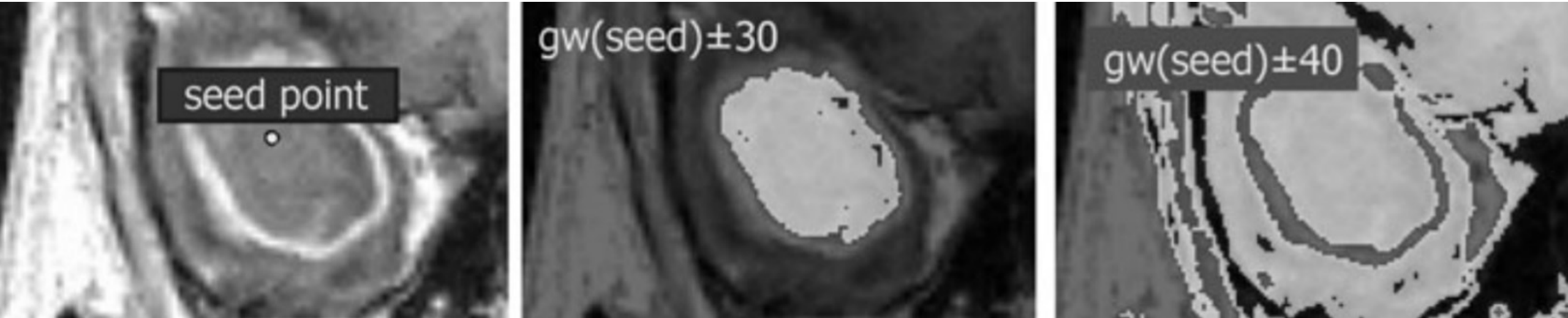
Detect all pixel $\mathbf{v}_l$ that are newly flooded at level l
if a pixel $\mathbf{v}_l$ is connected to pixels which all have a segment label L **then**
segment L is extended by $\mathbf{v}_l$.
if a pixel $\mathbf{v}_l$ is not connected to any pixel that has a segment label **then**
this is the first pixel of a new segment: label it L_{new}.
if a pixel $\mathbf{v}_l$ is connected to at least two pixels with different labels **then**
label the pixel W (for watershed)
$l = l+1$
end_while

- Can also use *marker-based* WST which adds information about objects to be segmented by replacing local minima with pre-specified marker positions
- Regions that are not flooded from a marker position receive a provisional label 'unknown', if an 'unknown' region merges with a marker-labeled region, it receives label from marker-labeled region
- Using markers pre-defines number of segments
- Marker positions can be set manually, or may be computed by a detection process



Region growing

- Region growing combines data-driven constraints with interactively added domain knowledge:
  - User specifies seed point in object
  - Homogeneity criteria characterizes the appearance of region to be segmented
  - Region growing then detects all pixels or voxels that can be reached from the seed point by paths of adjacent pixels for which the homogeneity criteria applies



Region growing

- Simple region growing algorithm:

```
Algorithm Region_growing_segmentation (xseed)
begin
        Label_field[] = not_visited
        Region_grow(xseed)
end

Algorithm Region_grow(x)
begin
    if label_field[x] = not_visited and hom(f(x)) then
        begin
                label_field[x] = visited
                for i=1,nbs do
                        region_grow(neighbor(x,i))
        end
end
```

Starts on image $f(\boldsymbol{x})$ with seed point $xseed$ and homogeneity predicate $\hom\big(f(\boldsymbol{x})\big)$, the function $neighbour(\boldsymbol{x}, i)$ returns i^{th} neighbor of $\boldsymbol{x}$

- Homogeneity needs to be computable for a single pixel, often the variance of the intensity function $f(\boldsymbol{x})$ around some expected value is used
  - This assumes that the foreground object can be separated from background by a lower and upper threshold
- Can be challenging to define homogeneity criterion. Can be done in 2-step process:
  - 1) first pass: seed point set, and homogeneity computed based on estimate of expected value and variance
    - This estimate is based on number of elements in the region already found, and current variance among these elements
    - Initially, region has only a few elements and variance is underestimated. With continuing growth the homogeneity criterion approaches current estimates of expected value and variance of intensity
  - 2) process is repeated with same seed point, but using homogeneity criteria computed in first pass

Segmentation in feature space

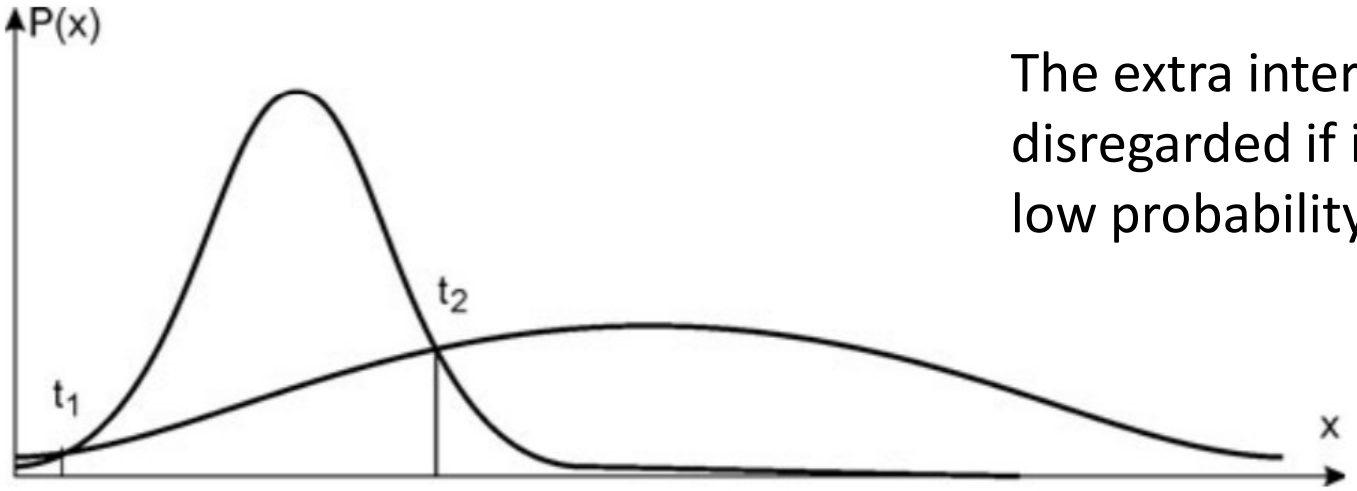
- Image segmentation can be formulated as classification in feature space, where image intensities are the features
- Dimensionality of feature space is usually low, and number of samples characterizing object classes is high
- Classifiers take this into account and estimate likelihood functions from samples
- Classification then done by computing posterior probability for each object class
- Clustering in feature space is also a useful technique
  - Doesn’t require training, may lead directly to segmentation

Segmentation by classification in feature space

- Given an intensity $f(\boldsymbol{v})$ at some scene element $\boldsymbol{v}$ and two different classes (foreground fg and background bg), the optimal solution that minimizes the number of wrong decisions would be to assign the class with highest posterior probability $P(\boldsymbol{v} \in fg | f(\boldsymbol{v}))$ and $P(\boldsymbol{v} \in bg | f(\boldsymbol{v}))$ respectively to $\boldsymbol{v}$
- According to Bayes' Theorem, posterior probabilities can be computed from the prior probabilities $P(\boldsymbol{v} \in fg)$ and $P(\boldsymbol{v} \in bg)$ and likelihood functions $p(f(\boldsymbol{v}) | \boldsymbol{v} \in fg)$ and $p(f(\boldsymbol{v}) | \boldsymbol{v} \in bg)$:
- $$P\big((\boldsymbol{v} \in fg | f(\boldsymbol{v}))\big) = \frac{p\big(f(\boldsymbol{v}) \big| \boldsymbol{v} \in fg\big) P(\boldsymbol{v} \in fg)}{p\big(f(\boldsymbol{v}) \big| \boldsymbol{v} \in fg\big) P(\boldsymbol{v} \in fg) + p\big(f(\boldsymbol{v}) \big| \boldsymbol{v} \in bg\big) P(\boldsymbol{v} \in bg)}$$
- $$P\big((\boldsymbol{v} \in bg | f(\boldsymbol{v}))\big) = \frac{p\big(f(\boldsymbol{v}) \big| \boldsymbol{v} \in bg\big) P(\boldsymbol{v} \in bg)}{p\big(f(\boldsymbol{v}) \big| \boldsymbol{v} \in fg\big) P(\boldsymbol{v} \in fg) + p\big(f(\boldsymbol{v}) \big| \boldsymbol{v} \in bg\big) P(\boldsymbol{v} \in bg)}$$
- Denominator same in both equations, can be omitted when finding most likely class membership

Segmentation by classification in feature space

- Sometimes, the two posterior probabilities intersect only once, giving the threshold for when decision should switch from foreground to background
- The two posterior probabilities can intersect more than once:



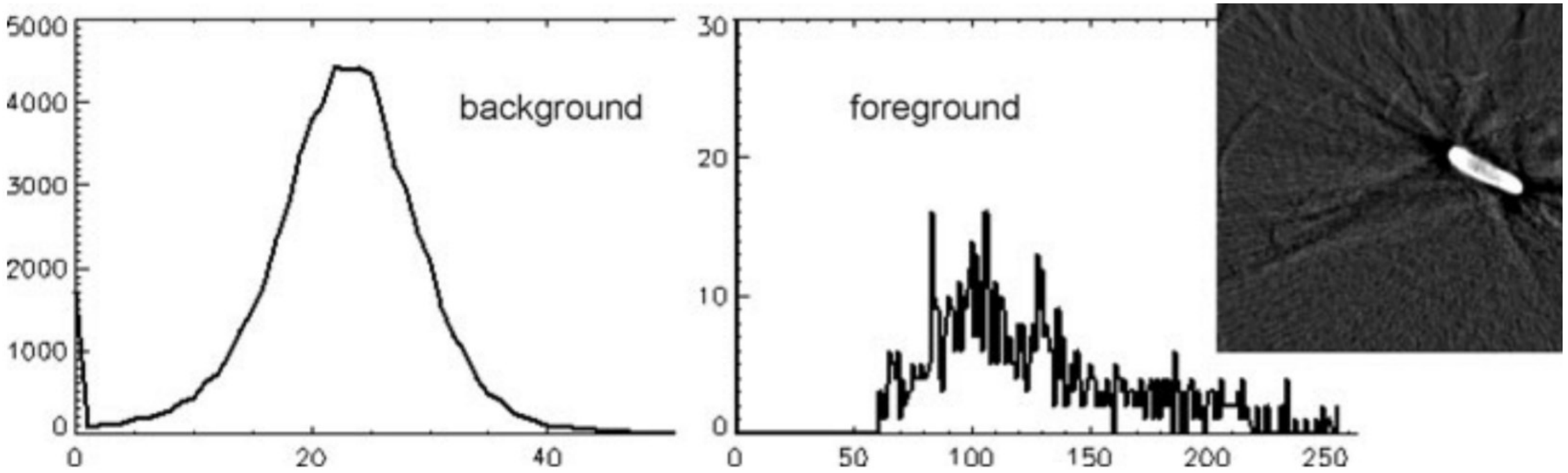
The extra intersection can be disregarded if it occurs at a location with low probability (t_1), so we use only t_2

- Prior probability and likelihood function must be generated from training data
- Data should be representative of the problem, so that is feature values reflect true likelihood function
- If appearance of foreground or background changes across patients or scans, training data should come from several different images (to account for this variability)

Computing the likelihood function

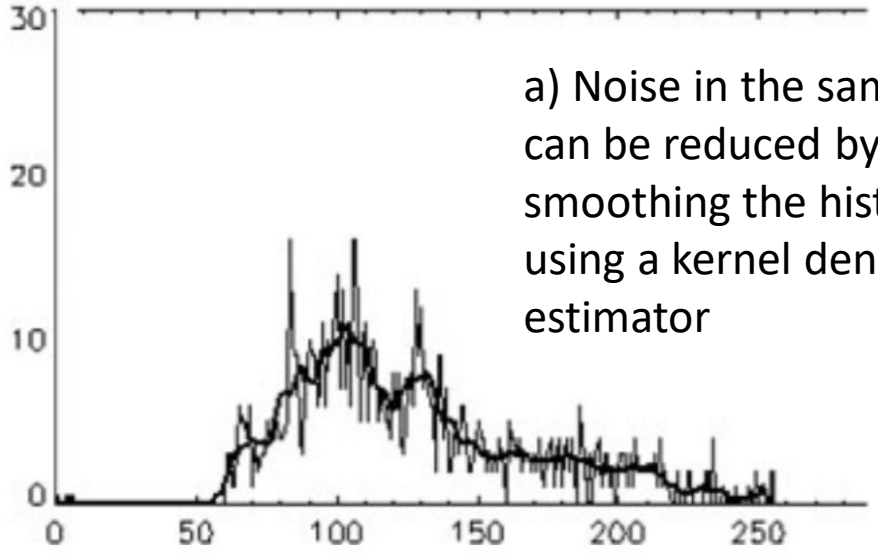
- $p(f(\boldsymbol{v})|\boldsymbol{v} \in fg)$ and $p(f(\boldsymbol{v})|\boldsymbol{v} \in bg)$
- Given classified training data, the estimate of the likelihood function for the two classes are normalized histograms h^n_{fg} and h^n_{bg} of the intensity values of scene elements belonging to either foreground or background
- a) estimates for foreground and background probability generated from a representative sample of a 3d digital angiogram. Background is well approximated by a gaussian, while this is not true of the foreground
- Histograms can be taken directly as likelihood functions, if number of samples is enough to cancel out effects of noise

a)

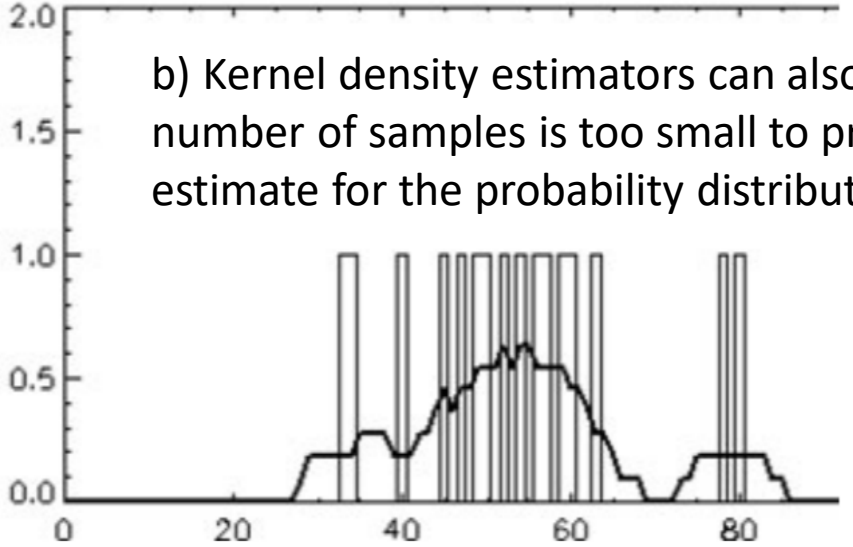


Kernel density estimator

- Using a *kernel density estimator* helps to reduce unwanted effects from undersampling
  - Based on assumption that each sample value $f(\boldsymbol{v})$ is representative of a range of values in the vicinity of $f(\boldsymbol{v})$, hence $f(\boldsymbol{v})$ assumed to be the mean of some unknown density function $d()$
  - a new estimate for likelihood function then computed by convolving normalized histograms with density functions d_{fg} and d_{bg}: $p(\,) = h_{fg}^{n} * d_{fg}$ and $p(\,) = h_{bg}^{n} * d_{bg}$
  - Since d_{fg} and d_{bg} are usually known, they are approximated by Gaussian density functions
  - The variance of the Gaussian is set based on the average sampling density in the normalized histograms, and noise characteristics



a) Noise in the sample data can be reduced by smoothing the histogram using a kernel density estimator

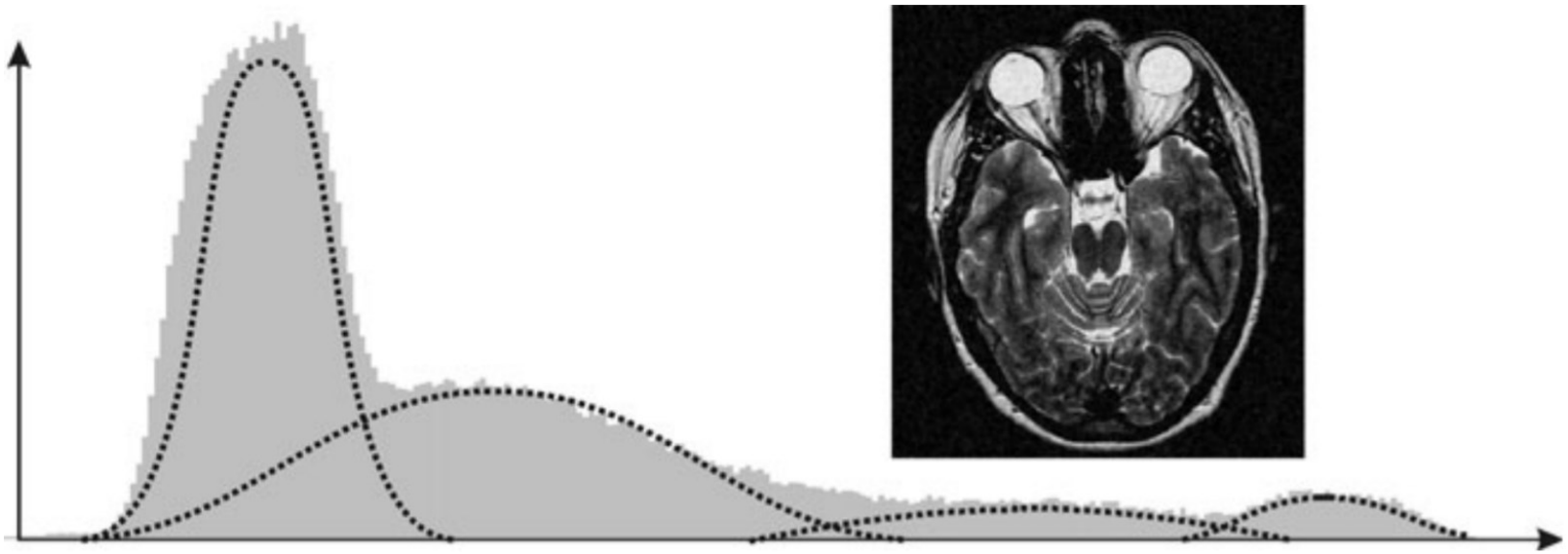


b) Kernel density estimators can also be used if the number of samples is too small to provide a reliable estimate for the probability distribution

Gaussian mixture modeling

- Sometimes it is known beforehand that intensity distribution is a combination of several density functions
  - a) Example: a foreground object consisting of different tissue types
- A *mixture of gaussians* can be assumed with different means, variances, and prior probabilities.
  - Particularly appropriate if the class to be described consists of a mixture of several tissues with different appearances
  - Individual likelihood functions allowed to overlap, since classification of tissues types is not the goal

a)



Gaussian mixture modeling

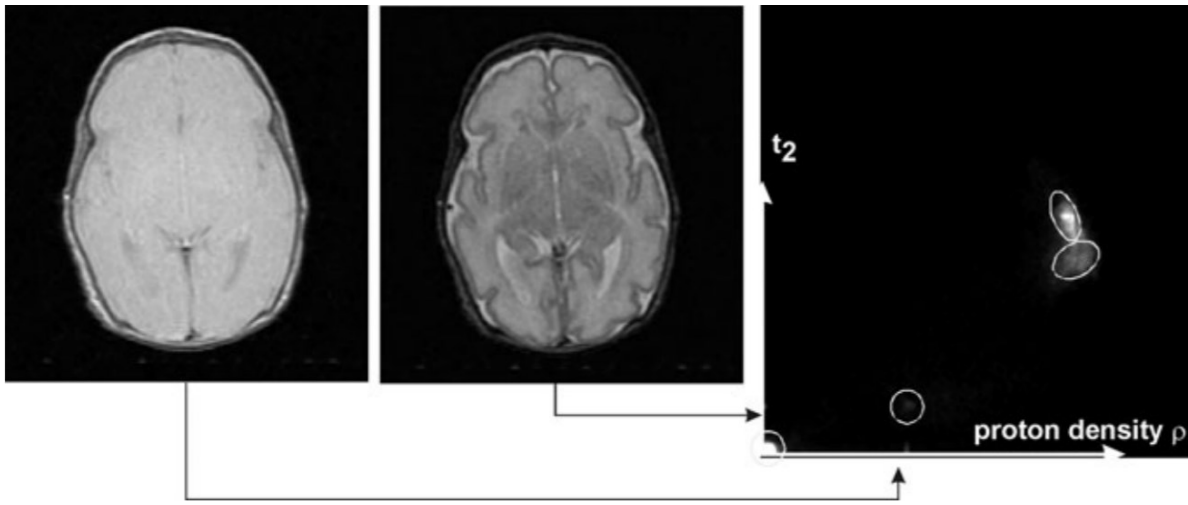
- Gaussian mixture model $\boldsymbol{\Theta}$ for one-dimensional distributions consists of a set of K Gaussian density functions $N(\mu_i, \theta_i)$ with mean μ_i and standard deviation σ_i
- The probability of some event to have value x_j then given by weighted sum of the functions $N(\mu_i, \sigma_i)$: $P\left(x_j \middle| \boldsymbol{\Theta}\right) = \sum_{i=1}^{K} a_i N(x_j; \mu_i, \sigma_i)$
  - Where the weights a_i are the prior probabilities for each distribution. The values x_j are intensity values of the scene element $\boldsymbol{v}$
- Best model for a set of sample values $X = \{x_j\}$ maximizes $P(X|\boldsymbol{\Theta})$:
- $\boldsymbol{\Theta}_{\max} = \mathrm{argmax}_{\boldsymbol{\Theta}} \mathrm{P}(\mathrm{X}|\boldsymbol{\Theta})$, for which there exists no analytical solution
- can use *expectation maximization algorithm* (EM), an iterative procedure to estimate a maximum posterior solution such as the one above

Multi-dimensional feature vectors

- Computation is the same if feature is not a scalar, but a vector
- Particularly easy if features can be assumed to be independent of each other
- Likelihood function for feature vector $\boldsymbol{f}$ is then the product of the likelihood functions of elements $f_1, \ldots f_M$ of the vector:
- $p(\boldsymbol{f}|\boldsymbol{v} \in fg) = \prod_{m=1}^{M} p(f_m|\boldsymbol{v} \in fg)$
- Likelihood function becomes a function of M values if independence cannot be assumed, and probabilities need to be estimated in multi-dimensional feature space (a)

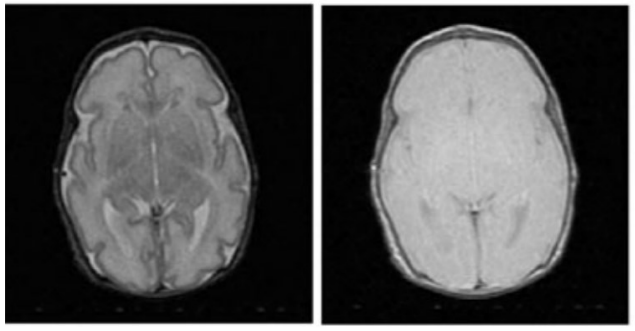
- **Example:** Assume we have taken a single segmented slice with 512x512 pixels for training. if 64000 of the 2^{18} pixels belong to foreground, and intensity ranges from 0 to 4000, of which foreground object occupies a range of 400 gray values, we have an average of 160 samples per bin.
- This would allow to use the sampled, normalized histogram directly as a likelihood function
- However, if we have the same situation but image consists of 3 channels (e.g. T1, T2, and proton density) where the foreground object spanned range of 400 gray levels in each channel, average number of samples would be 0.001 per bin

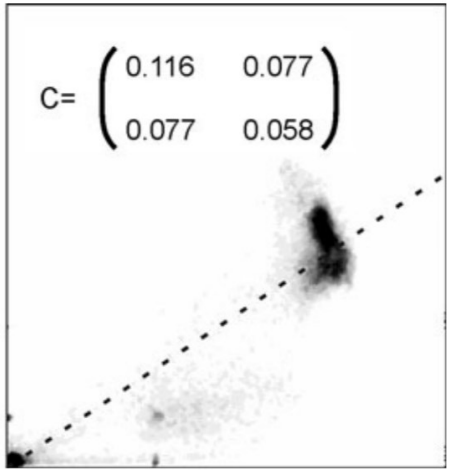
a)



Example continued…

- Even if scans from 50 patients used, each scan with 20 slices where foreground is visible, average number of samples still only 1 per bin
- Therefore, parametric distribution function often used for estimating multi-dimensional likelihood function
  - Normal distribution used as model
- Since features not independent, variances for each element of feature vector replaced by covariances
- Covariance matrix Σ for an M-dimensional feature vector consists of entries σ_{ij} estimated from K sample features $\boldsymbol{f}^K$ by:
- $\sigma_{ij} \approx \frac{1}{K-1}\sum_{k=1}^{K}(f_i^k - \mu_i)(f_j^k - \mu_j)$
- where $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots \mu_{\mathrm{M}})$ is the estimated mean for features $\boldsymbol{f}^k$
- The multi-dimensional normal distribution $N(\boldsymbol{f}; \boldsymbol{\mu}, \Sigma)$ for a feature vector $\boldsymbol{f}$ is then:
- $N(\boldsymbol{f}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^M \det(\Sigma)}} \exp(-\frac{1}{2}(\boldsymbol{f} - \boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{f} - \boldsymbol{\mu})$





Covariance matrix C in 2d feature space is a 2x2 matrix.
The high value of off-diagonal entries compared to diagonal shows there is a lot of covariance between the two features.

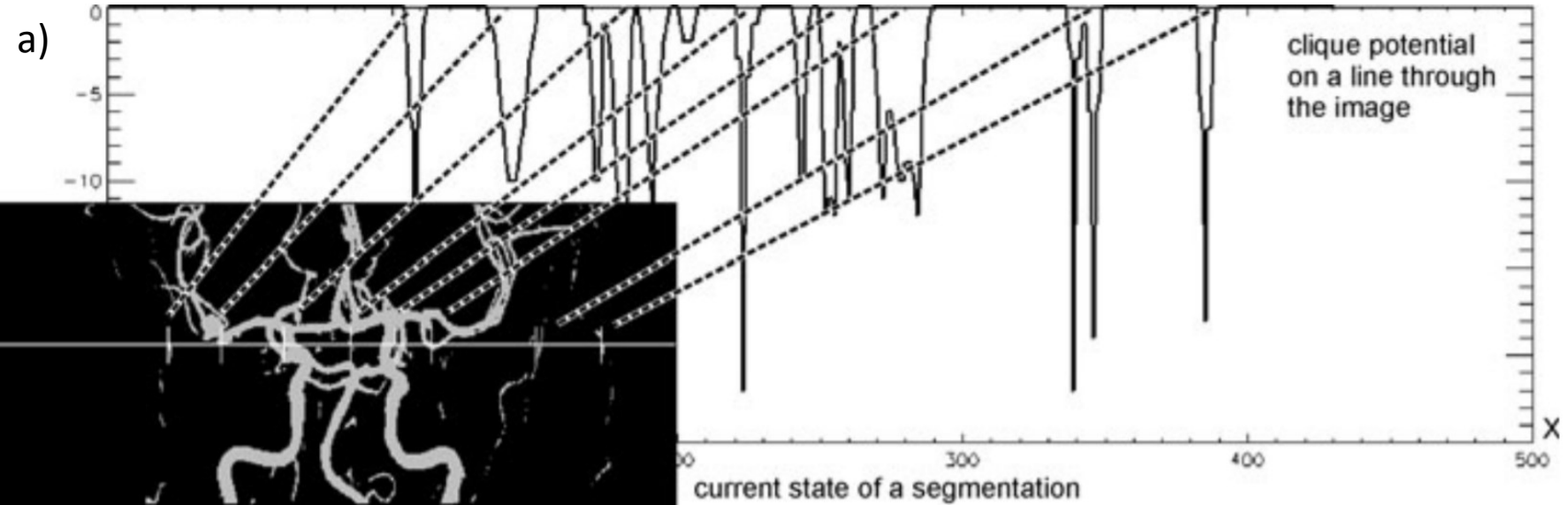
Computing prior probability

- Prior probability represents knowledge about objects irrespective of the data
- Simplest kind of prior for foreground segmentation is ratio of foreground to background pixels
  - Can be computed from training data used to compute likelihood functions
- Prior knowledge may also include neighborhood information, because segment membership is spatially correlated
  - More likely that neighboring scene elements belong to same class
  - this can be represented by a Markov random field (MRF):
  - $P(\boldsymbol{f}|\boldsymbol{g}) \propto P(\boldsymbol{g}|\boldsymbol{f}) \cdot P(\boldsymbol{f}) = \frac{1}{Z_1}\exp(-\frac{(f(\boldsymbol{v})-\mu)}{2\sigma^2}) \cdot \frac{1}{Z^2}\exp(-U(\omega))$
    - Similar to model used in Bayesian image restoration, see previous lecture for details
- For segmentation, use generalized Ising model to model clique potentials V_c that make up $U(\omega)$: $V_c = \begin{cases} \xi_c \; if \; all \; sites \; of \; clique \; have \; same \; label \\ -\xi_c \; otherwise \end{cases}$
- The potential ξ_c is specific to the clique type

Computing prior probability

- Instead of explicitly computing clique potentials U from all cliques in a given neighborhood, an approximation of U can be defined that describes desired segmentation properties
- Segmentation should produce as few segments as possible, so prior probability can simply decrease with number of different labels in neighborhood $\boldsymbol{v}$ (a)
- $P(\boldsymbol{f}) = \frac{1}{Z_2}\exp(-\sum_{\boldsymbol{w}\in Nb(\boldsymbol{v})}\delta(f(\boldsymbol{v}) \neq f(\boldsymbol{w})))$, where δ is the delta function
- Iterative optimization of the MRF then performed using simulated annealing, or other

a)



Example: MRF-based MRI brain segmentation. The potential decreases sharply in regions with different labels in the neighborhood. Further optimization will change labels at these locations, if the likelihood from image is low