Computer Graphics and Image Processing

Part 3: Image Processing

6 – Segmentation

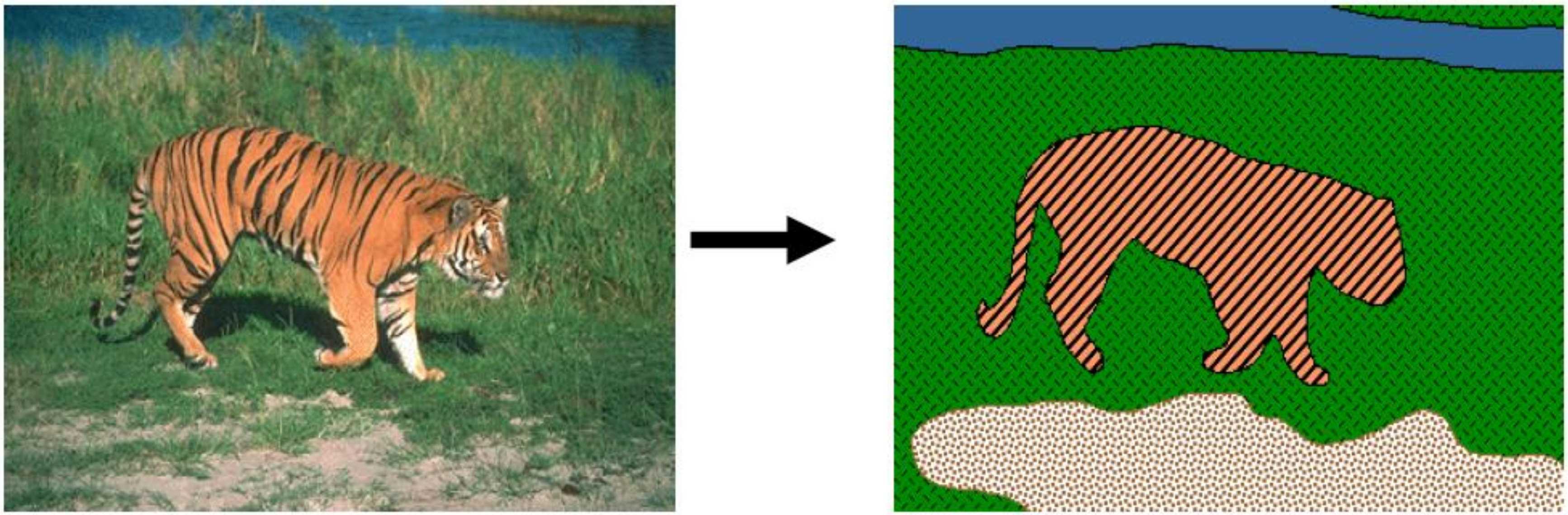
Martin Urschler, PhD

Image segmentation is hard!

http://optical-illusions.wikia.com/wiki/Emergence

Image segmentation definition

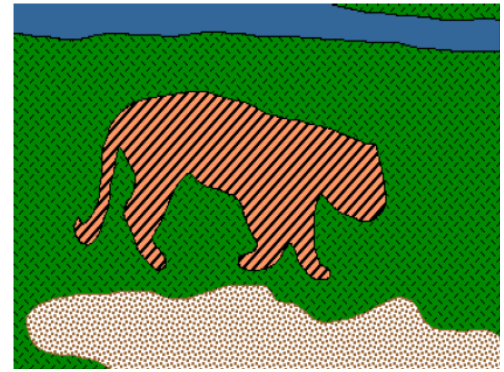
- Process of partitioning the image domain R (all pixels) into n subregions $R_1, R_2, \ldots, R_n$



- For $n = 2$: Binary segmentation (or foreground/background segmentation)

Image segmentation definition

- Process of partitioning the image domain R (all pixels) into n subregions $R_1, R_2, \ldots, R_n$ such that:

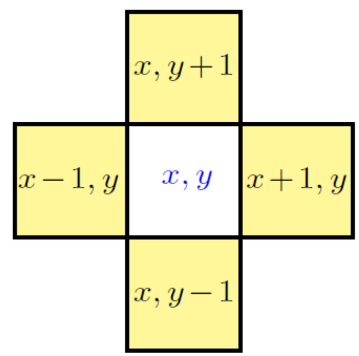


  - The union of the subregions R_i is R again
  - All subregions R_i are "connected" sets
  - Pairs of subregions are disjoint (their intersection is empty)
  - A logical predicate Q is true for each subregion R_i, e.g. all pixels of R_i have "similar" or same greylevel
  - For all pairs of subregions, the same predicate Q between pixels of the respective subregions is false

e.g.

4-neigbourhood:

$\{(x, y \pm 1),\ (x \pm 1, y)\}$

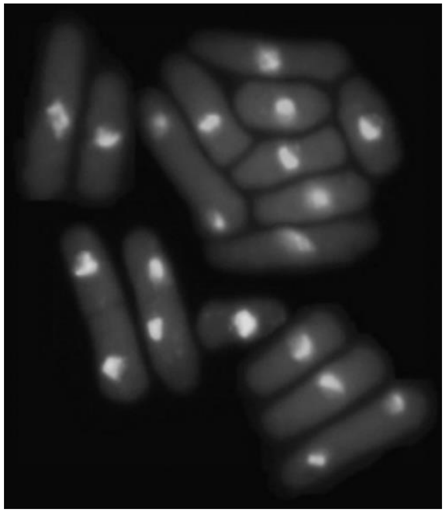


Segmentation problem

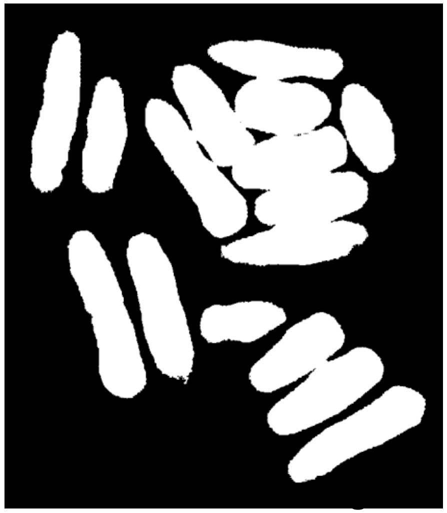
- Partitioning an image into distinct region with similar attributes
  - Meaningful regions relate to objects or features of interest (Semantics)
  - Semantic segmentation is first step from low-level image processing to high-level image analysis (IA)
  - IA describes depicted scenes in terms of features and objects
  - Success of IA depends on reliable segmentation!
  - In general very challenging! (Deep learning)

Semantic segmentation for image analysis

- Types of segmentation:
  - Non-contextual:
    grouping pixels with similar global features
  - Contextual:
    grouping pixels with similar features and in close locations



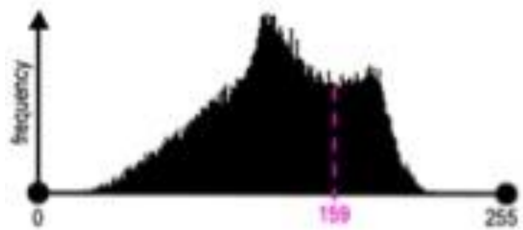
Yeast cells



Grouped pixels

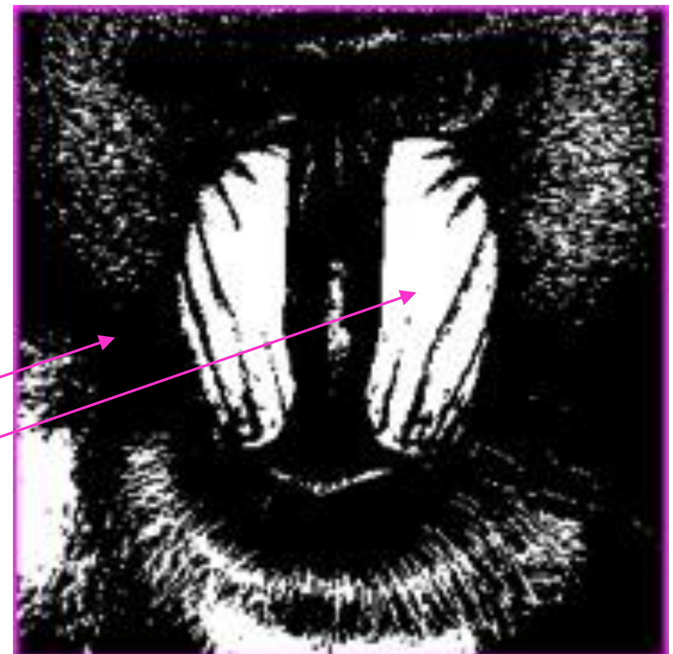
Non-contextual thresholding

- Simplest non-contextual technique
- Single threshold τ: converting a grayscale image g into a binary region map m: $m(x,y) = \begin{cases} 0 & if\ g(x,y) < \tau \\ 1 & if\ g(x,y) \geq \tau \end{cases}$

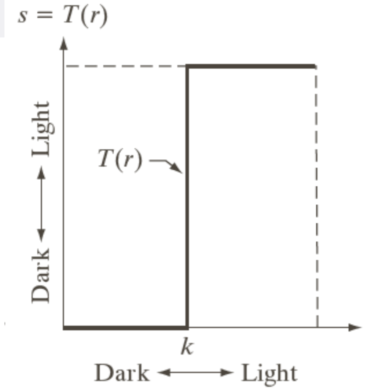


Threshold $\tau = 159$

Region 0 – black
Region 1 - white



Simple thresholding

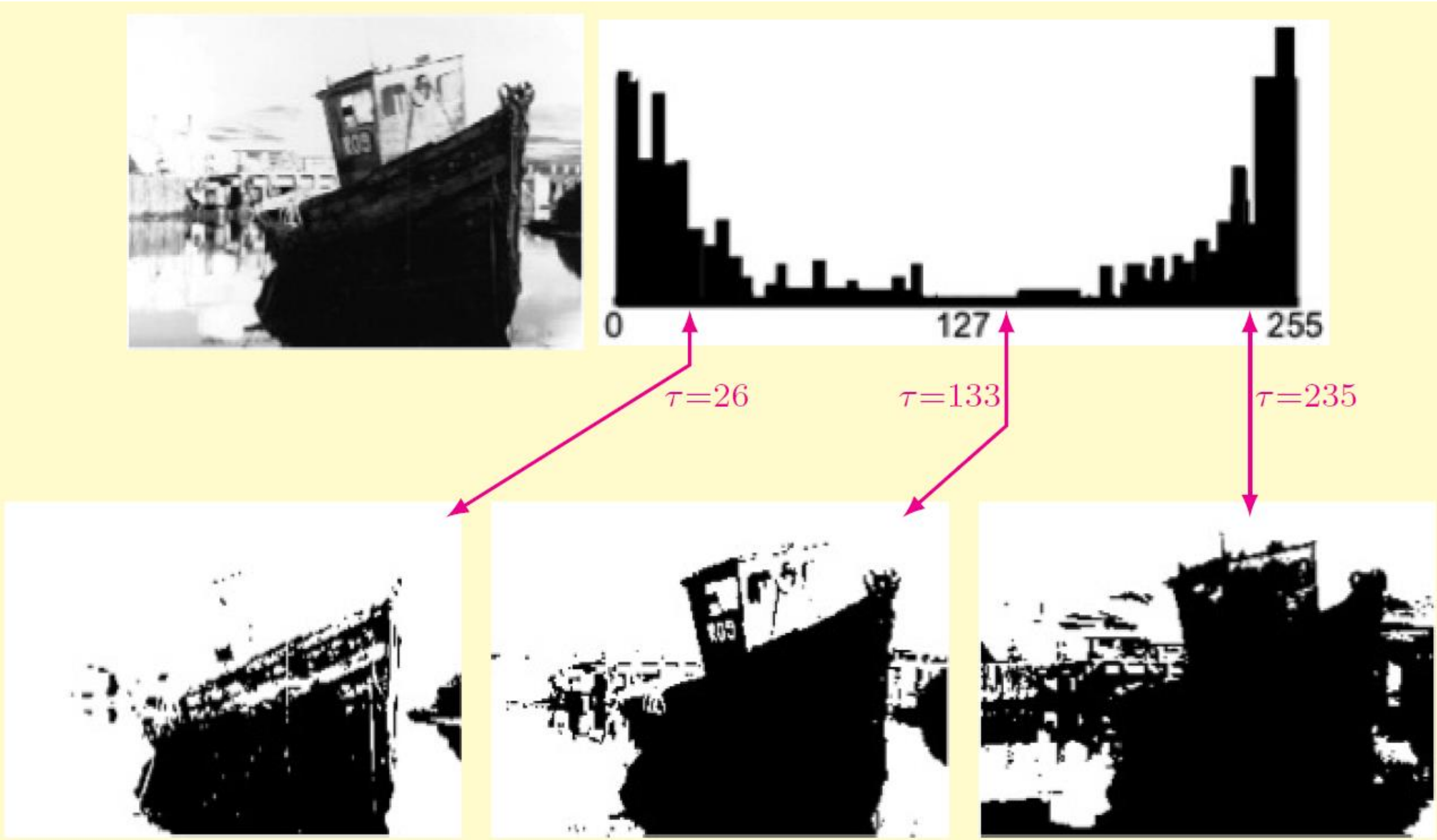


Intensity transformation

- A single threshold produces a binary map with two disjoint regions
  - Region 1 (label 0) with pixel values smaller than threshold
  - Region 2 (label 1 or 255) with values at or above the threshold
  - The regions depend on the image feature (e.g. graylevel) being compared to the threshold and on choice of threshold

- Generally, two or more thresholds can produce two or more types of regions
  - Thresholds may separate ranges of values associated with the region type
  - In principle, one region may combine several ranges of values
    - E.g. $m(x,y) = \begin{cases} 0 & if\ g(x,y) < \tau_1\ OR\ g(x,y) > \tau_2 \\ 1 & if\ \tau_1 \le g(x,y) \le \tau_2 \end{cases}$

Simple thresholding examples



Simple thresholding

Main problems of simple thresholding:

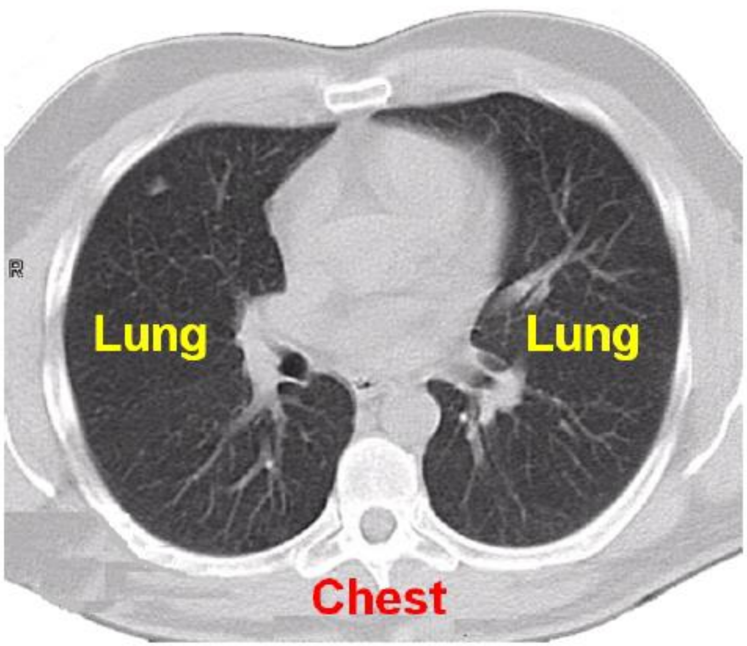
- Whether it is possible and, if yes,
- How to choose an adequate threshold or a number of thresholds to separate one or more desired objects from their background.

In many practical cases the simple thresholding is unable to segment objects of interest.

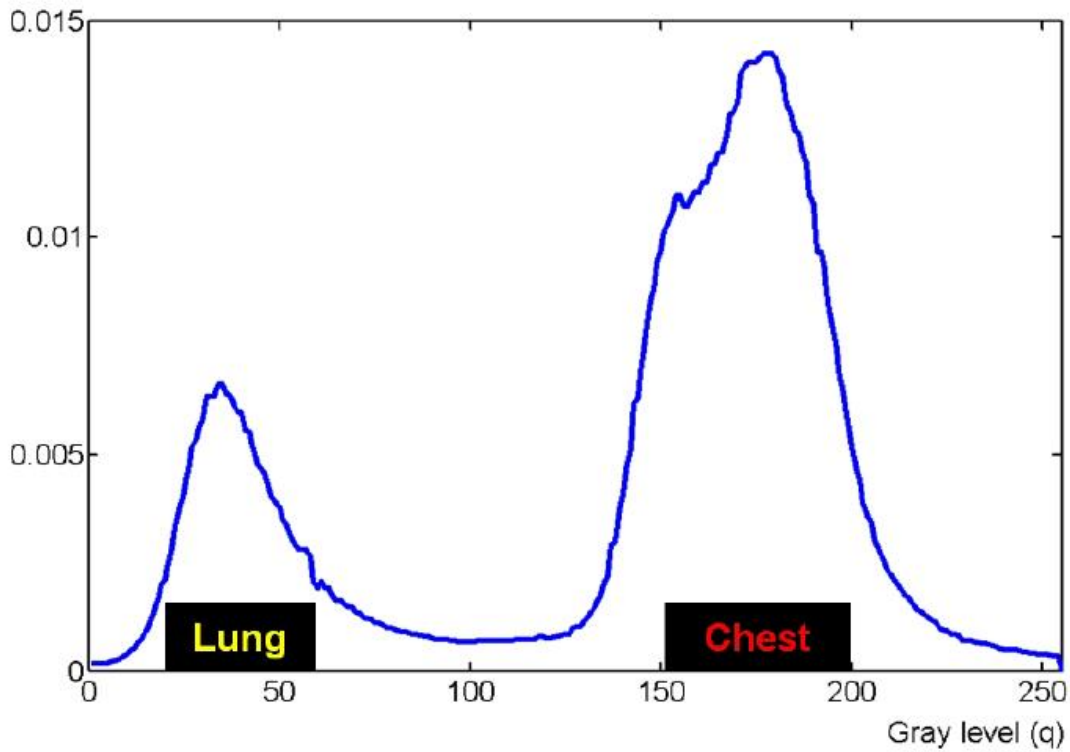
General approach – images are assumed being multimodal:

- Different objects of interest relate to distinct peaks, or modes of the 1D empirical signal histogram.
- Thresholds to optimally separate objects in spite of typical overlaps between signal ranges corresponding to individual peaks.

Multimodal images: Medical example

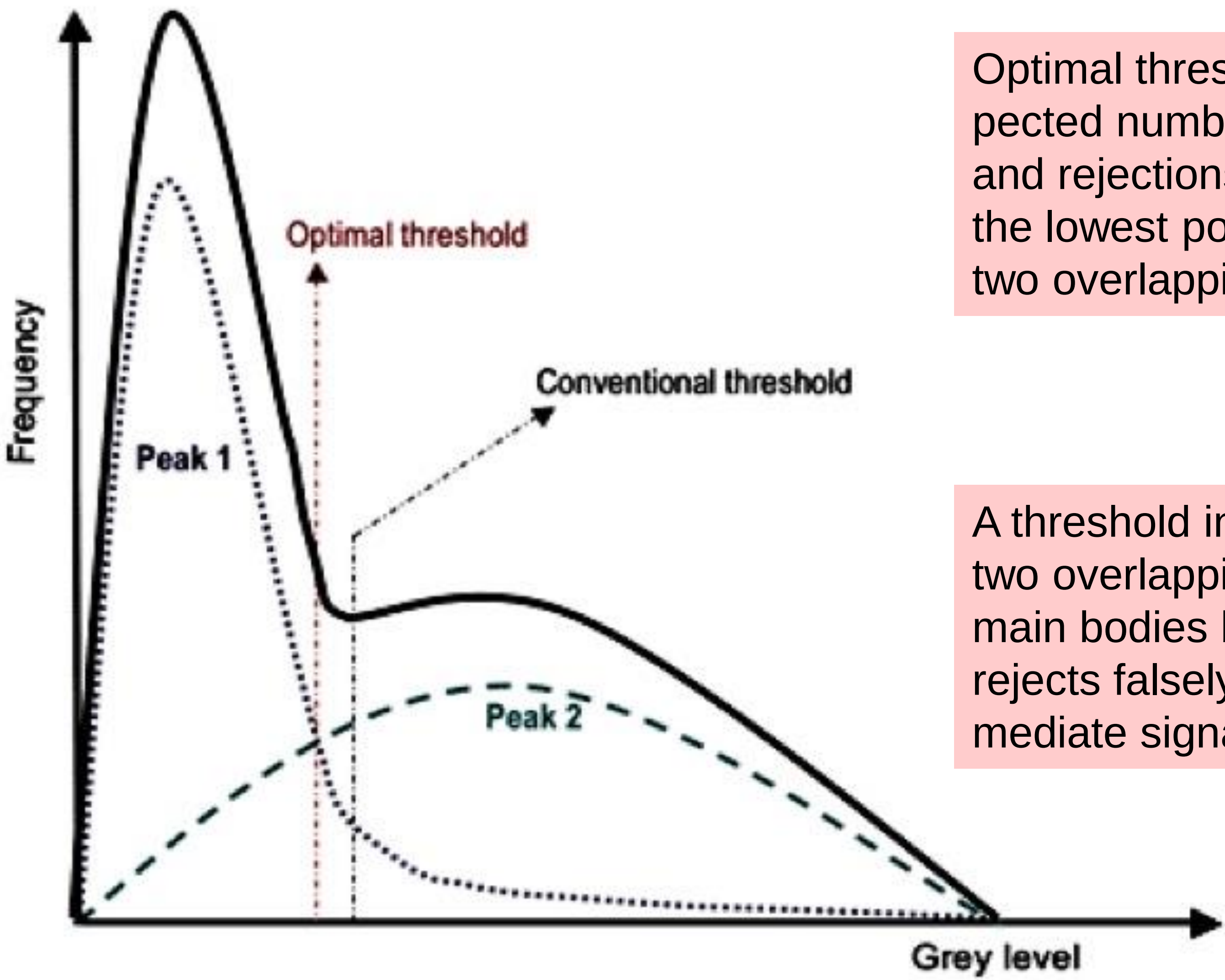


Chest slice



Histogram of occurrences of gray levels for the whole chest image

Thresholding by Mixture Separation



Optimal threshold minimising the expected numbers of false detections and rejections may not coincide with the lowest point in a valley between two overlapping peaks.

A threshold in the valley between two overlapping peaks separates their main bodies but inevitably detects or rejects falsely some pixels with intermediate signals.

Adaptive thresholding

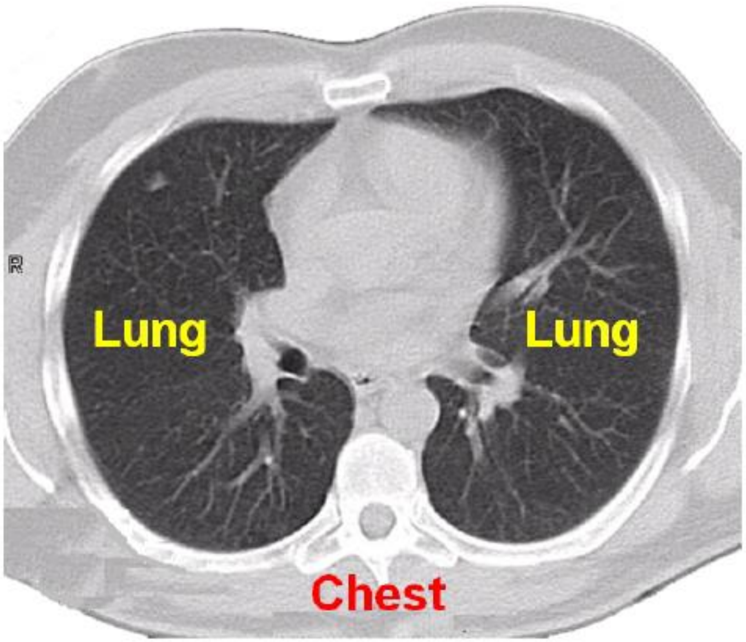
- Threshold separating a background from an object has to equalize two kinds of expected errors:
  - False alarm: assigning a background pixel to the object
  - Missing object: assigning an object pixel to the background

- **Adaptive non-contextual separation** accounts for empirical probability distributions of object (e.g. dark) and background (e.g. bright) pixels
  - E.g. object and background follow *Gaussian distribution*
    - If we describe Gaussian by mean: Our adaptive thresholding (next)
    - If we describe Gaussian by mean and standard deviation: *Otsu thresholding method*

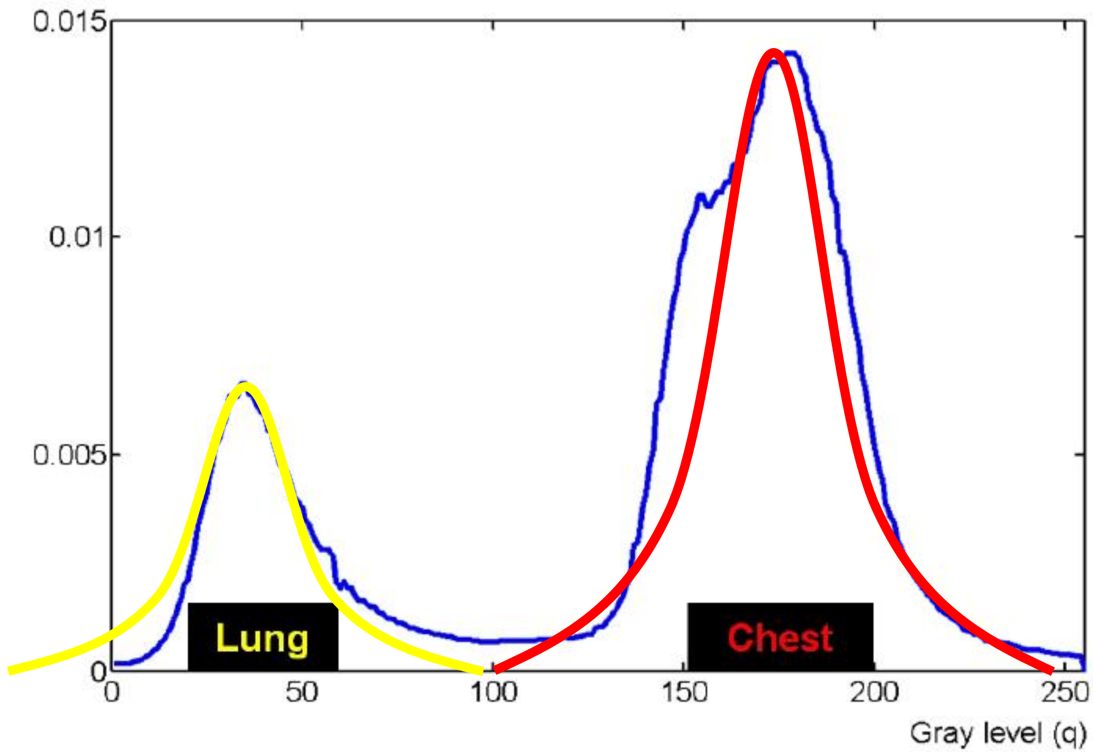
Adaptive thresholding

- Simple iterative adaptation of the threshold:
  - Successive refinement of the estimated peak positions
- Basic assumption: centre-symmetric distributions of object and background grey levels:
  1. Each peak coincides with the mean grey level for all pixels that relate to that peak
  2. Pixel probability decreases monotonically with the growing absolute difference between the pixel grey level and the peak grey level both for object and background
  3. Each grey level is associated with the relevant peak by thresholding
     - The threshold is placed half-way between the peaks!

Multimodal images: Medical example



Chest slice



Histogram of occurrences of gray levels for the whole chest image

Threshold adaptation at iteration j

1. Classify each grey level $g(x, y)$ with a threshold θ_j, having been computed at previous iteration:

$$pixel(x, y) \in \begin{cases} C_{ob:[j]} & if\ g(x, y) \leq \theta_j \\ C_{bg:[j]} & if\ g(x, y) > \theta_j \end{cases}$$

   where $C_{ob:[j]}$ and $C_{bg:[j]}$ are the object and background regions at iteration j, respectively, in the image g

2. Compute mean grey values for each class:

$$\mu_{ob:[j]} = \frac{1}{|C_{ob:[j]}|} \sum_{(x,y) \in C_{ob:[j]}} g(x, y) \qquad \mu_{bg:[j]} = \frac{1}{|C_{bg:[j]}|} \sum_{(x,y) \in C_{bg:[j]}} g(x, y)$$

   where $|C|$ denotes the number of pixels in the region C

3. Compute the new threshold $\theta_{j+1} = \frac{1}{2}\left(\mu_{ob:[j]} + \mu_{bg:[j]}\right)$

Adaptive thresholding: only histogram needed!

Input: an image histogram $\mathbf{H} = (H(q) : q = 0, \ldots, 255)$

Initialisation: $j = 0$; $N = \sum_{q=0}^{255} H(q)$; $\theta_0 = \frac{1}{N} \sum_{q=0}^{255} qH(q)$

while $\theta_{j+1} \neq \theta_j$ **do**

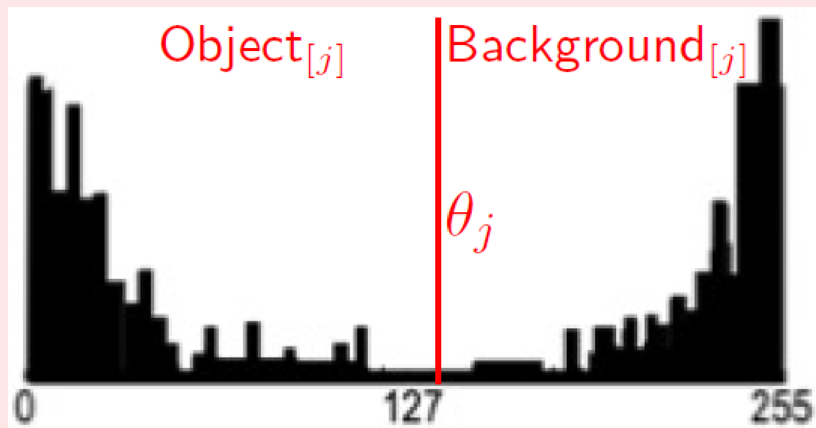
$$N_{\mathsf{ob}:[j]} = \sum_{q=0}^{\theta_j} H(q); \quad N_{\mathsf{bg}:[j]} = \sum_{q=\theta_j+1}^{255} H(q)$$

$$\mu_{\mathsf{ob}:[j]} = \frac{1}{N_{\mathsf{ob}:[j]}} \sum_{q=0}^{\theta_j} qH(q)$$

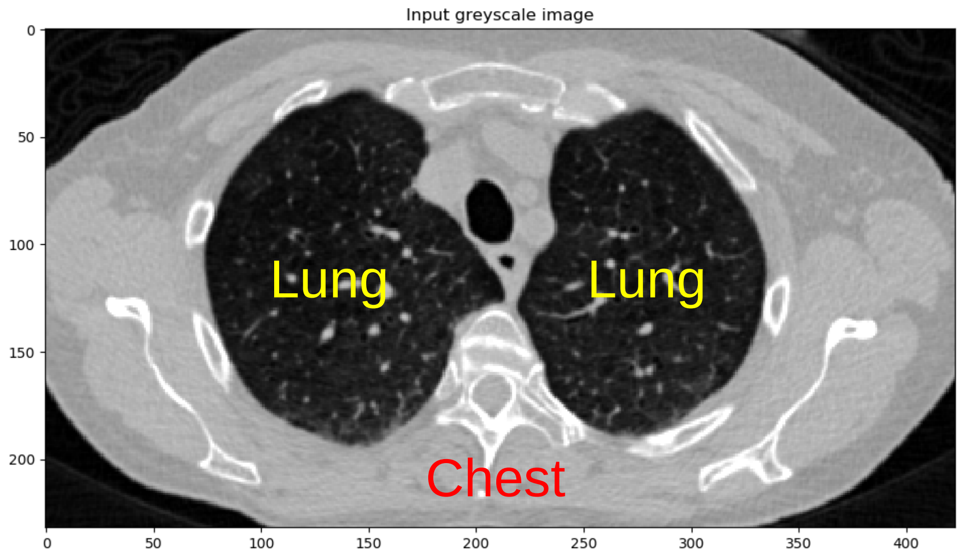
$$\mu_{\mathsf{bg}:[j]} = \frac{1}{N_{\mathsf{bg}:[j]}} \sum_{q=\theta_j+1}^{255} qH(q)$$

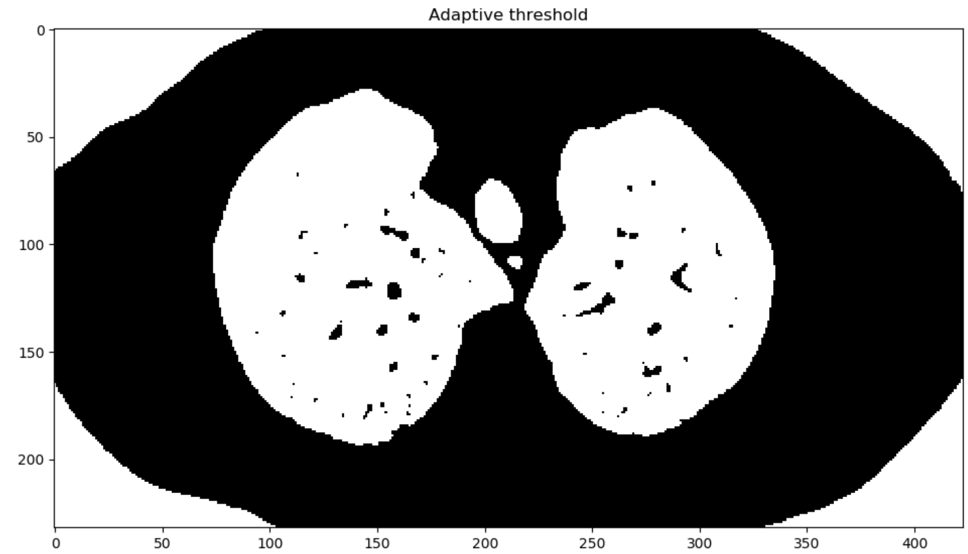
$$\theta_{j+1} = \frac{1}{2}\left(\mu_{\mathsf{ob}:[j]} + \mu_{\mathsf{bg}:[j]}\right)$$

end while



Example: Adaptive lung thresholding





Initialization: $\theta_0 = 116$

$$\theta_1 = 101$$
$$\theta_2 = 100$$
$$\theta_3 = 100$$

Finished after 3 iterations

How to get rid of holes?
Morphological Closing!