FBSP: Point Processing

Nick Yao Larsen

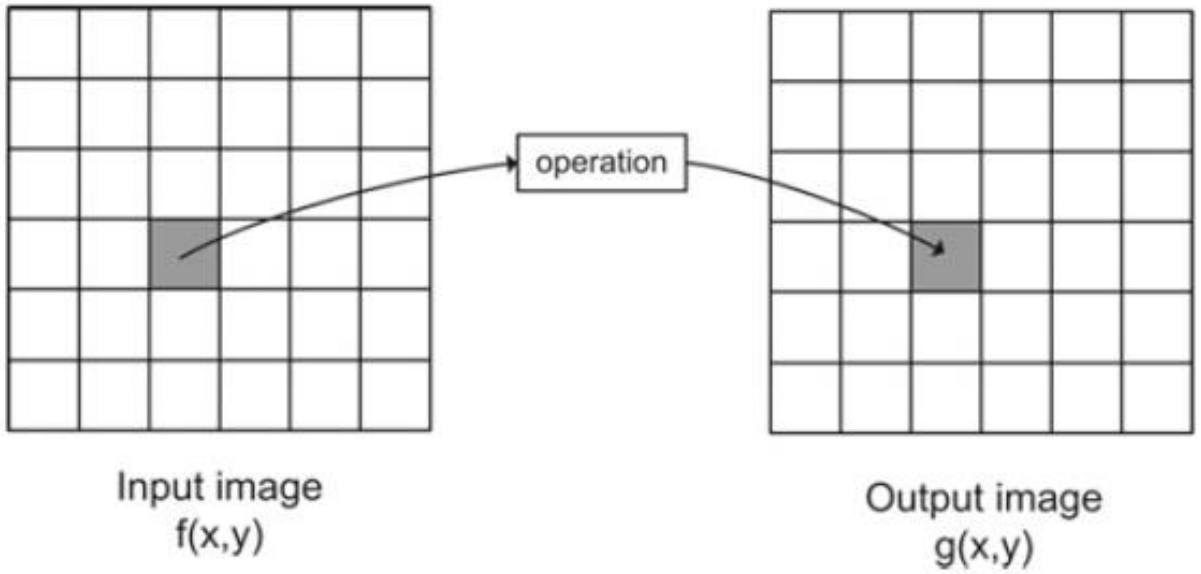
nylarsen@cfin.au.dk

Image as a function

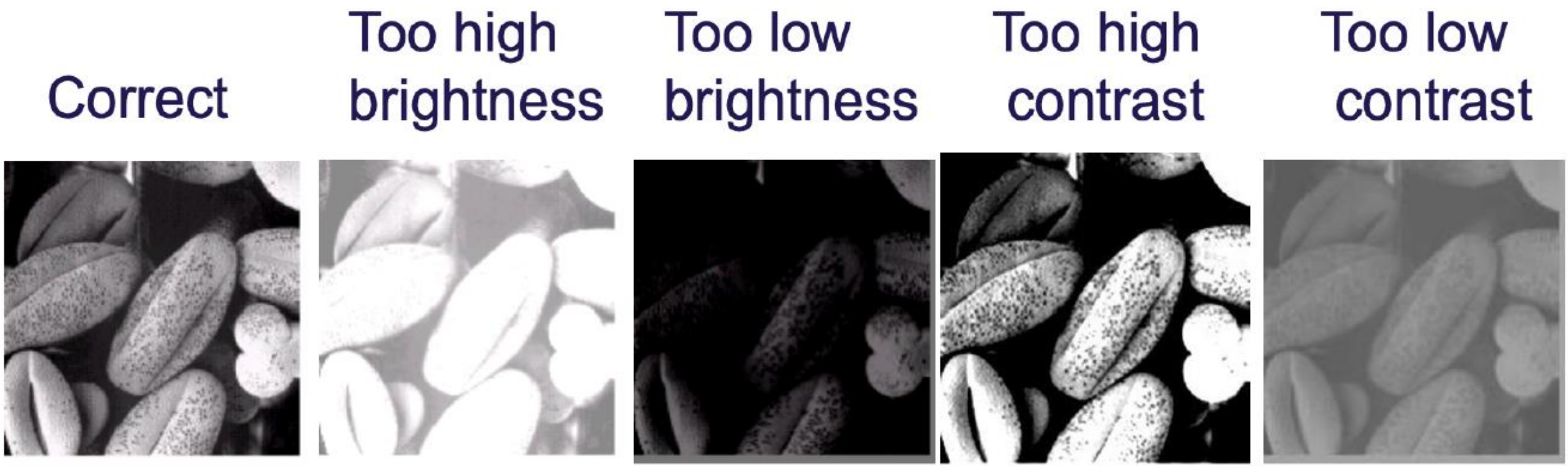
- The spatial coordinates of an image are x and y.
- *f(x,y)* is the intensity postion of *(x,y)*

What is Point (pixel) processing

- Point processing is now defined as an operation which calculates the new value of a pixel in g(x, y) based on the value of the pixel in the same position in f (x, y)
- **Simplest** type of operation because the pixel value itself is independent on the location of other pixel.
- Examples: Changing brightness, contrast, thresholding, histogram stretching



Gray level enhancement



Brightness

- The brightness is the intensity
- Change brightness: $g(x,y) = f(x,y)+b$
  - To each pixel is added the value of b
  - $f(x,y)$ is the input image
  - $g(x,y)$ is the (enhanced) output image

- If b>0 => brighter image
- If b<0 => darker image

- Be aware of image saturation (image bleaching)
  - At some point the pixels cannot exceed its maximum or minimum values

b = 0

b < 0

Decreased brightness

b > 0

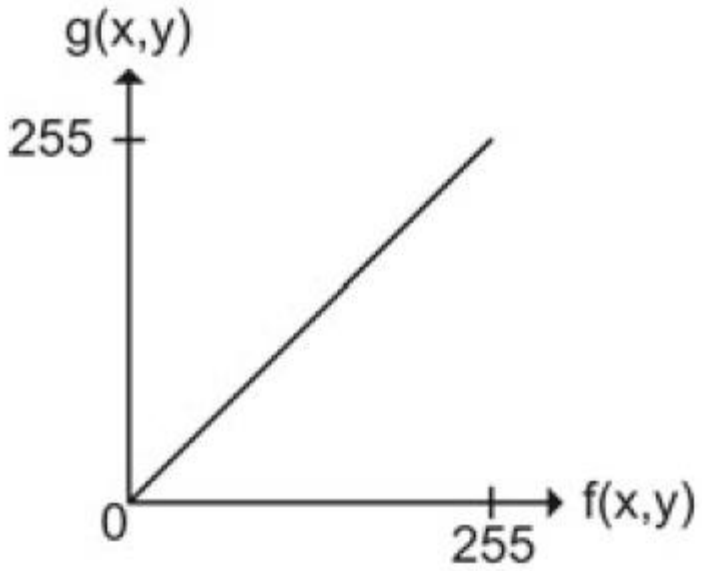
Increased brightness

Contrast

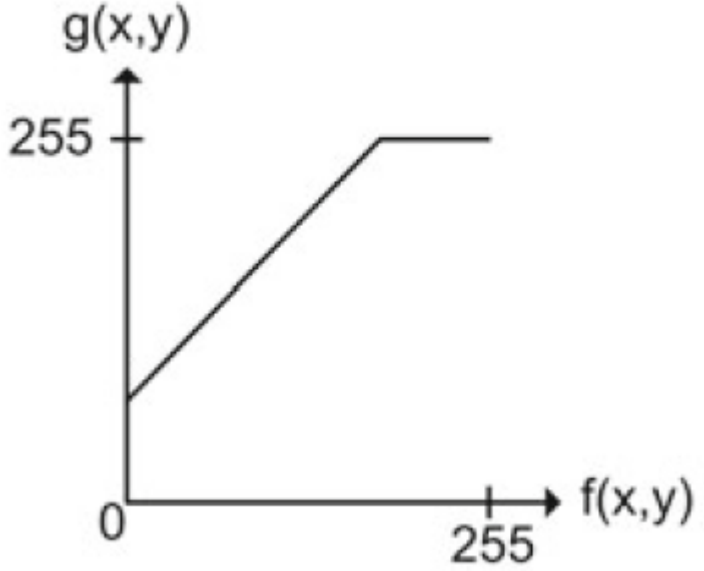
- Contrast describes the level of detail we can see, or the difference between pixel values
- Change contrast: $g(x,y) = a*f(x,y)$
  - Each pixel is multiplied by a
  - $f(x,y)$ is the input image
  - $g(x,y)$ is the output image
- If a>1 => more contrast
- If a<1 => less contrast

- Be aware of image satuation

Gray level intensity transformation

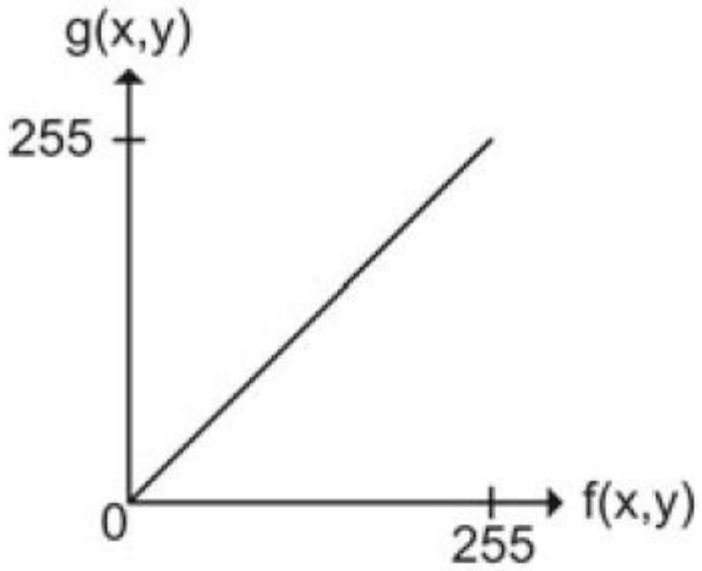


g(x,y), b = 0

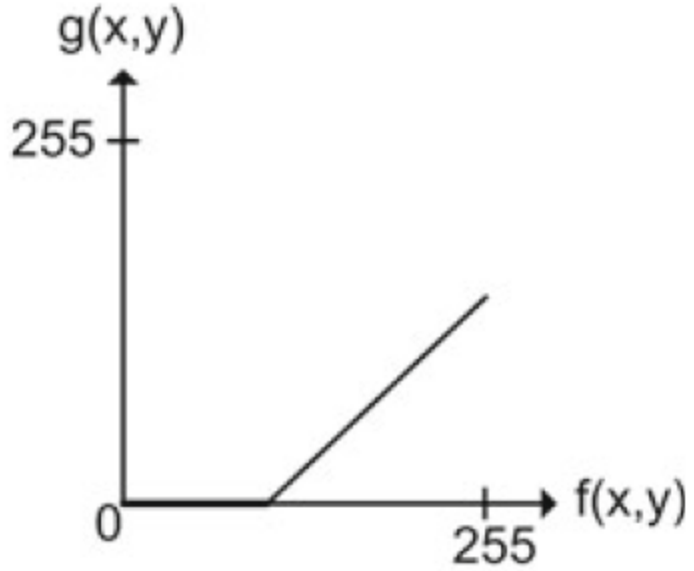


g(x,y), b = 75

Gray level intensity transformation



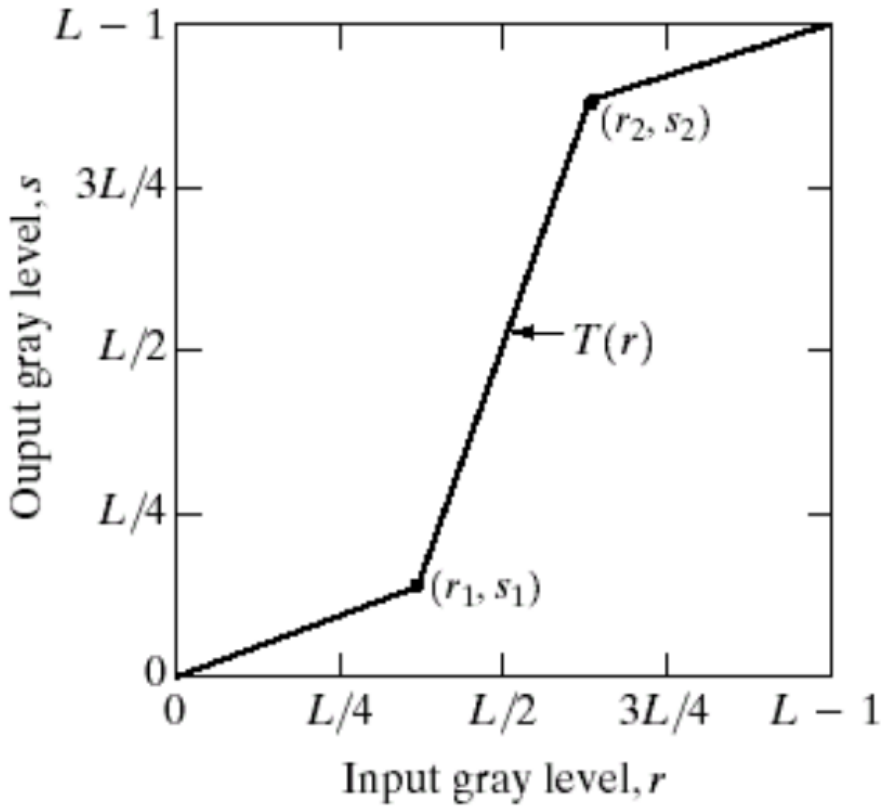
g(x,y), b = 0



g(x,y), b = -100

Gray level intensity transformation

- Apply gray level transformation in “sections”

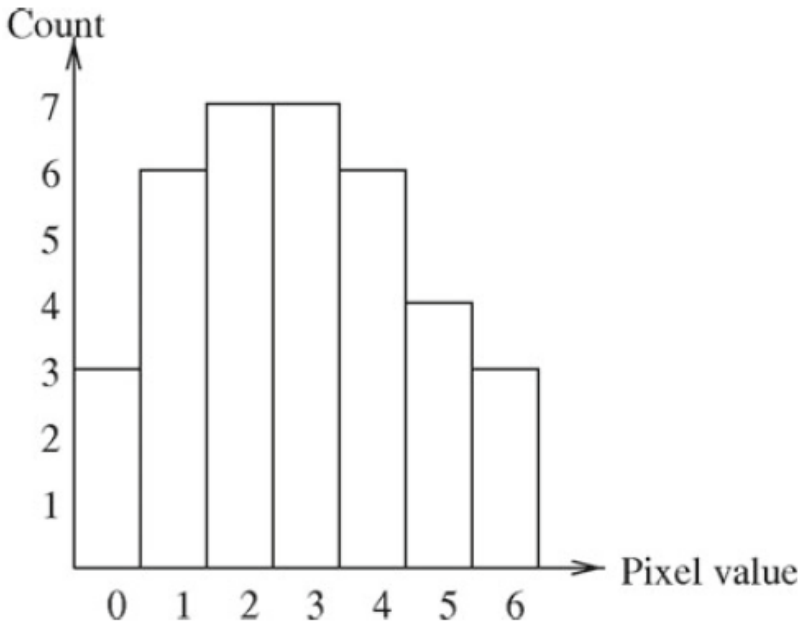


- The slope of T(r):
- - If Slope > 1 —> Contrast increases
- If Slope < 1 —> Contrast decrease
- If Slope = 1 —> no change

Image histograms

- How can we tell if an image is too dark or too bright?
- A histogram is a graphical representation of the frequency of events
  - The horizontal axis represents the possible numbers in a matrix /could be a picture
  - The vertical axis represents the count of the different numbers in the matrix /could be a picture
- Each column is denoted a **bin** and the height of a **bin** corresponds to the number of the numerical value on x-axis.

0	2	6	6	3	3
1	4	3	4	4	4
3	2	5	1	5	2
1	4	2	1	3	1
2	5	3	0	2	0
4	2	5	6	3	1



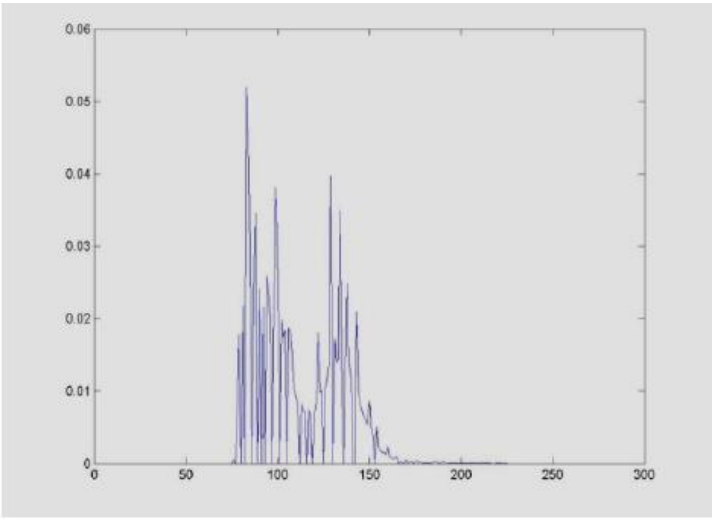
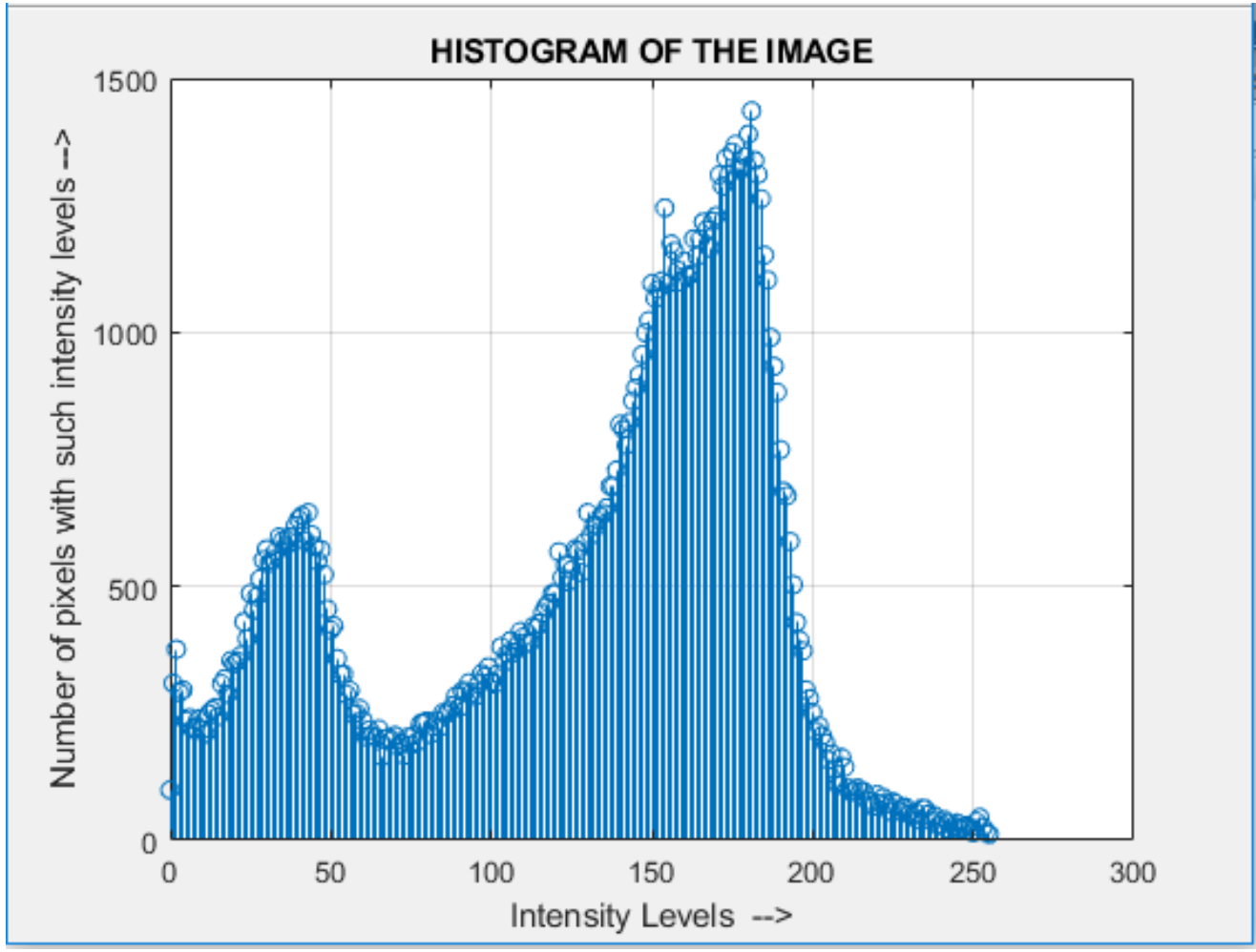
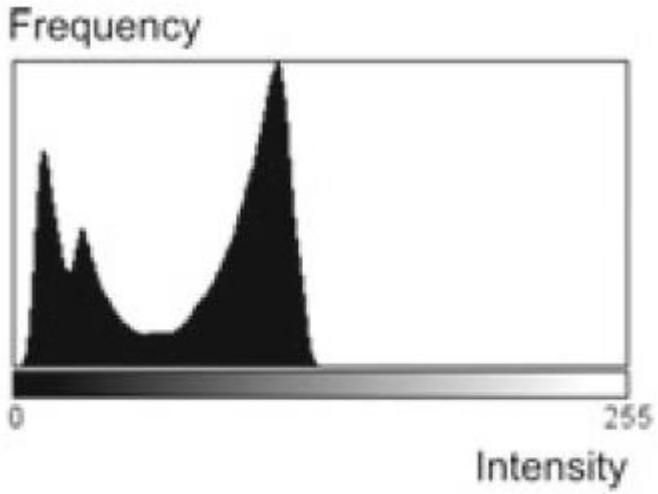


Image histograms

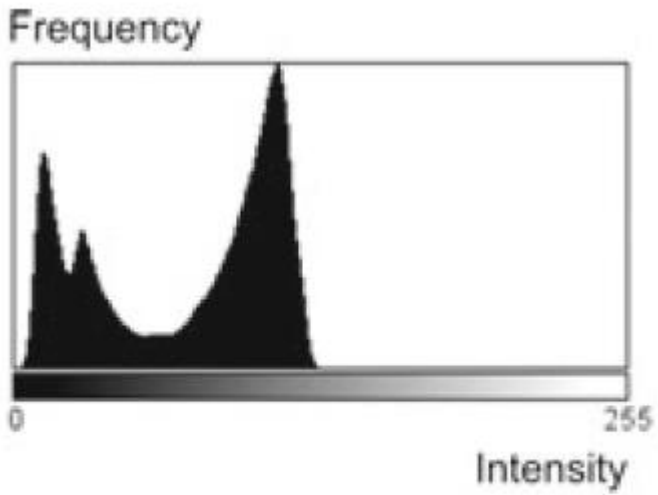
- A histogram is a discreate function
  $p(k) = \frac{n_k}{n}$
- n_k is the number of pixel with the k-th gray level
- n is the total number of pixels
- Histogram entry p(k) gives the probability of k appearing in the image



Interpret histogram

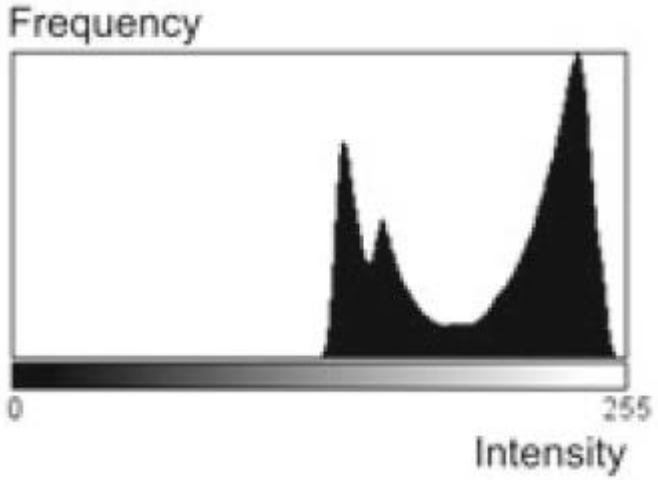


Interpret histogram

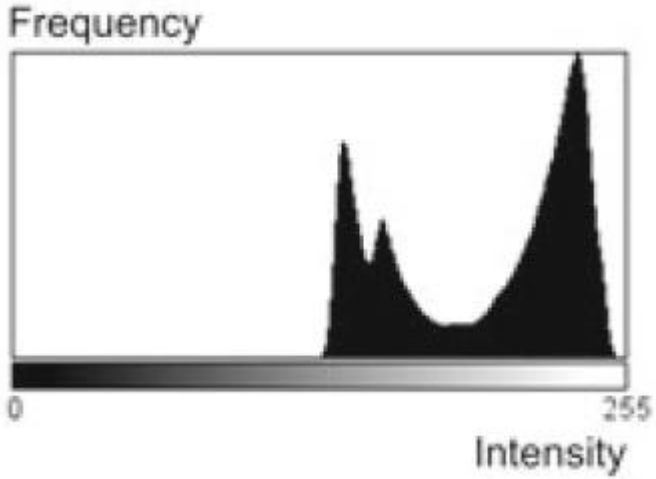


Dark image

Interpret histogram

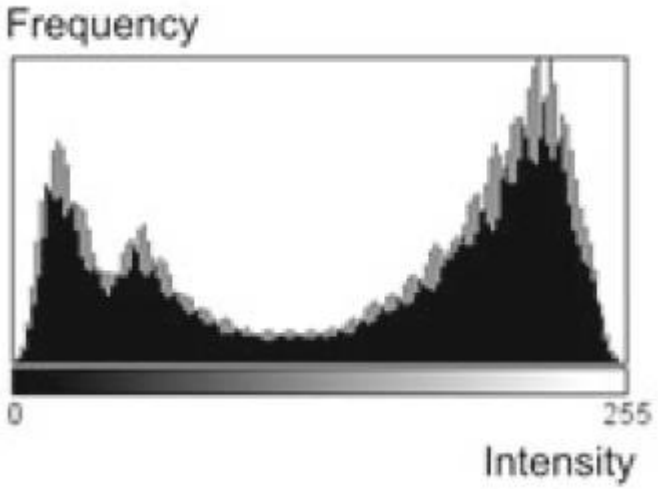


Interpret histogram

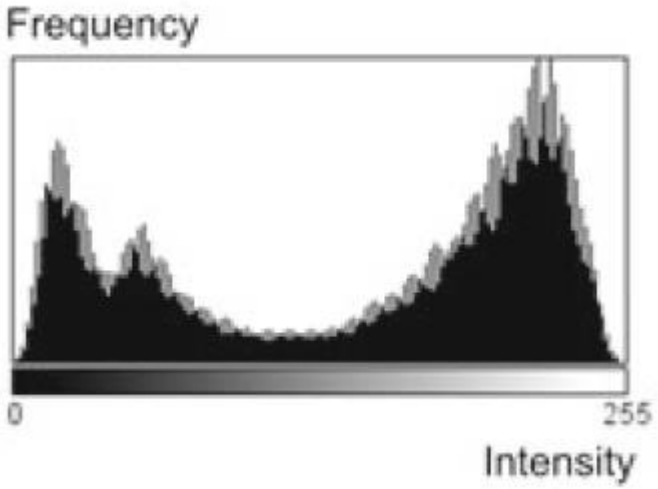


Bright image

Interpret histogram

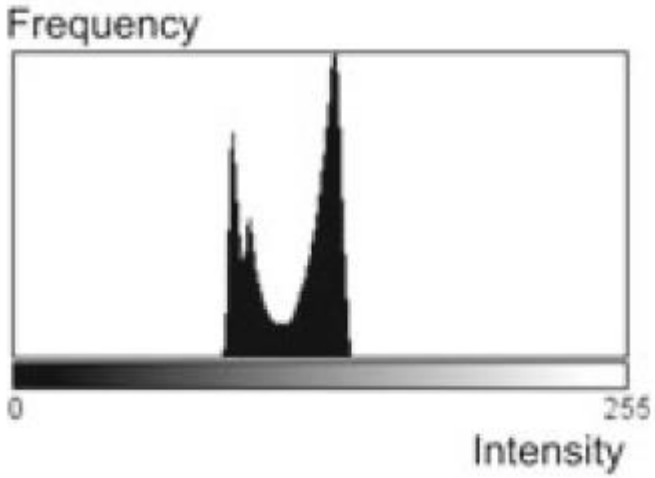


Interpret histogram

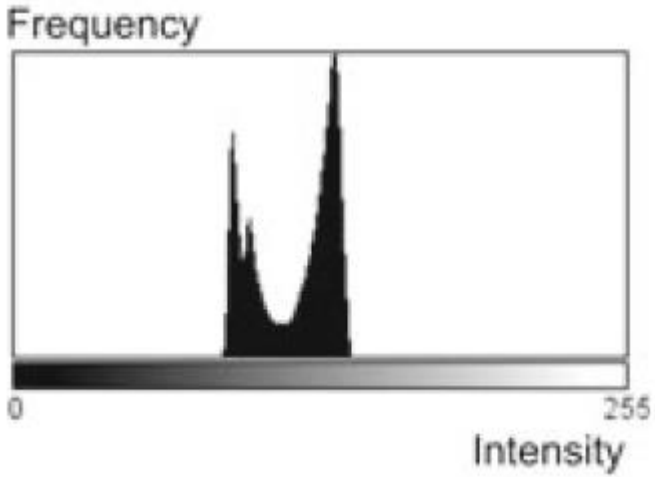


High contrast image

Interpret histogram



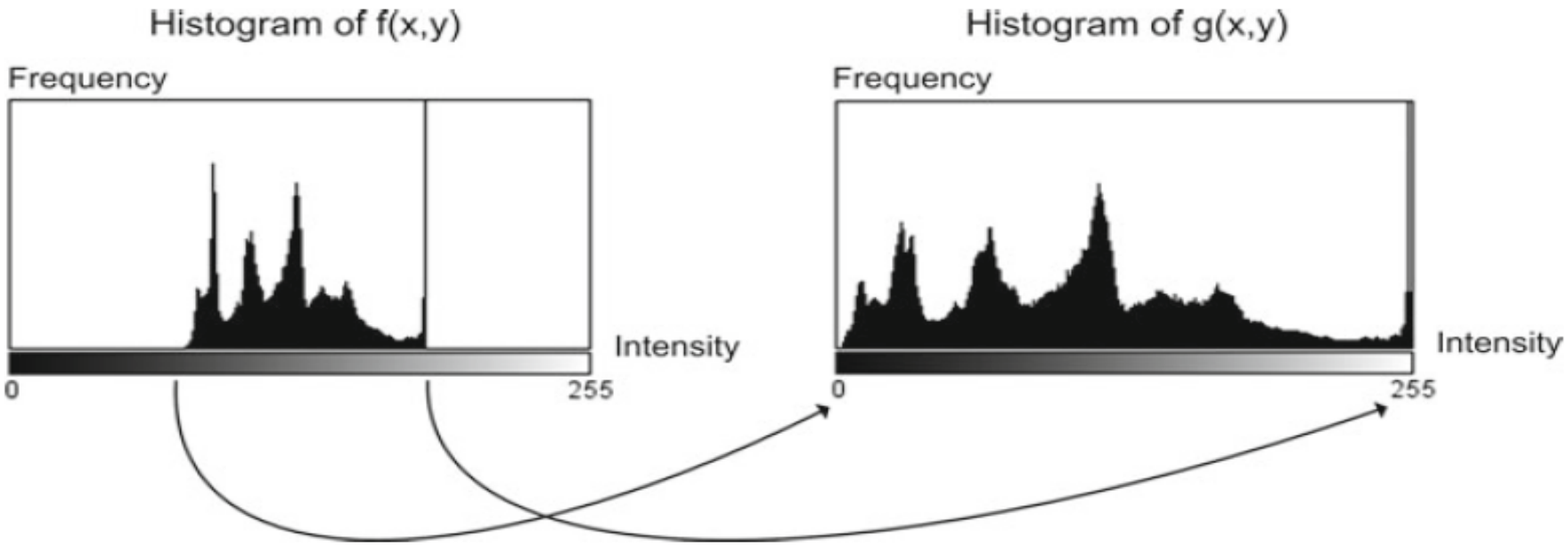
Interpret histogram



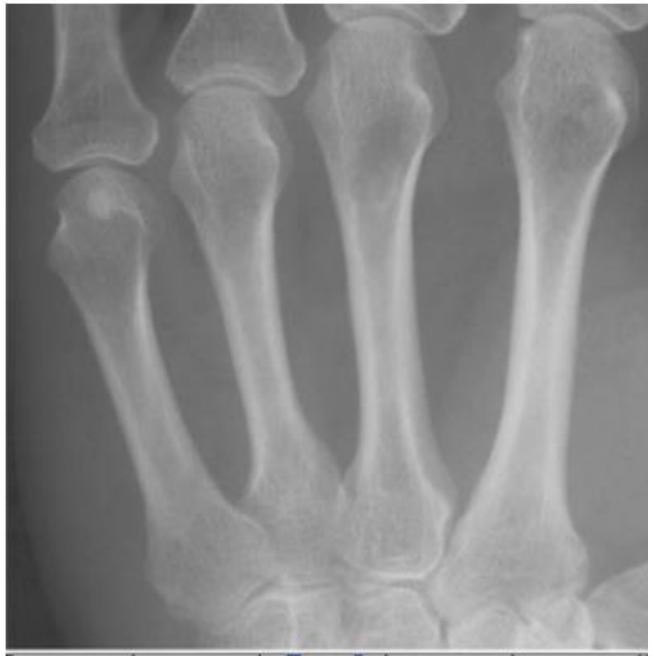
Low contrast image

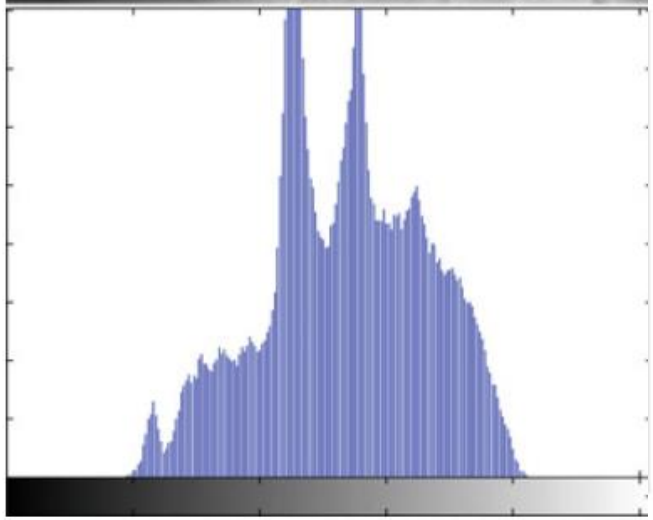
Improving contrast

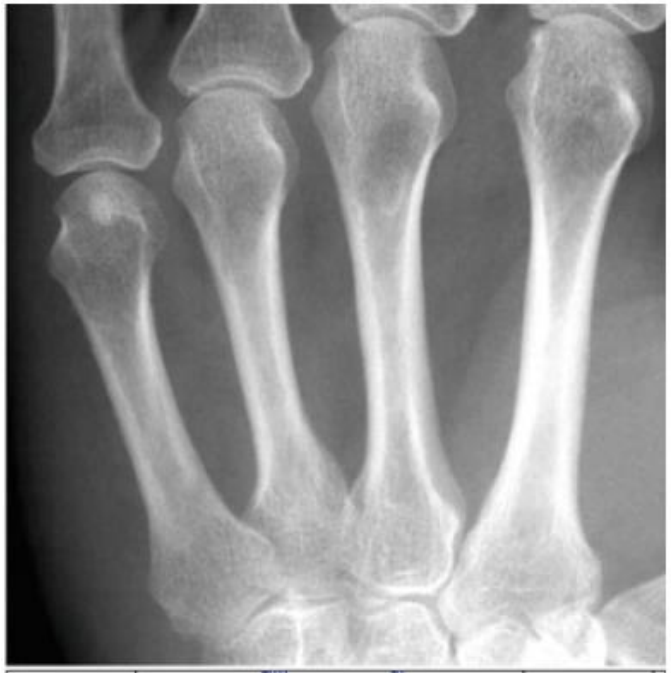
- Humans cannot tell the difference between graylevel values too close to each other
- So, spread out graylevel values
- This is called **histogram stretching**

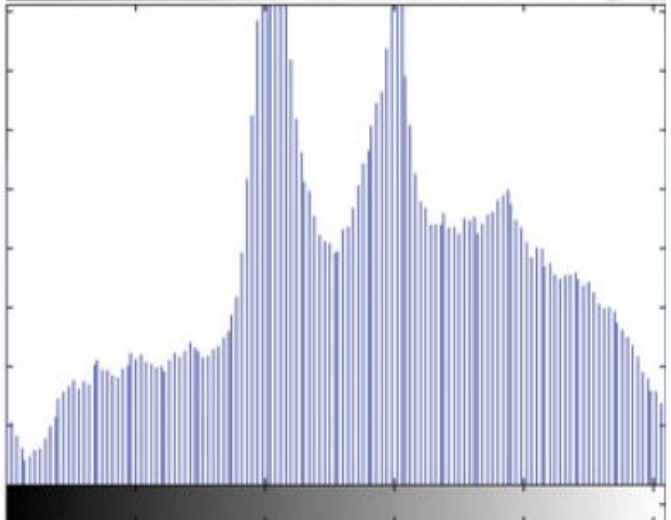


Histogram stretching - Examples





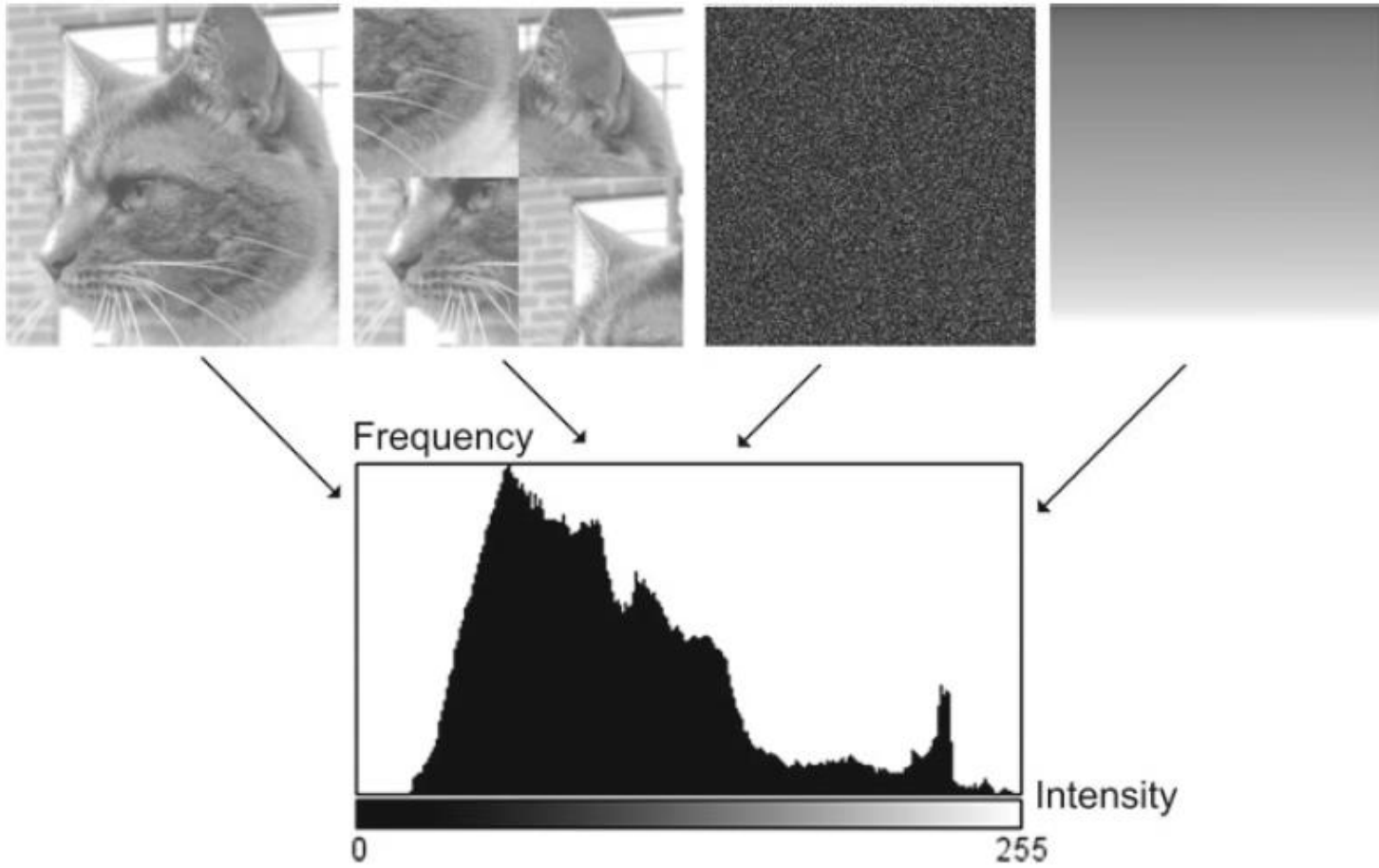




Histogram stretching - Examples

Histogram summary

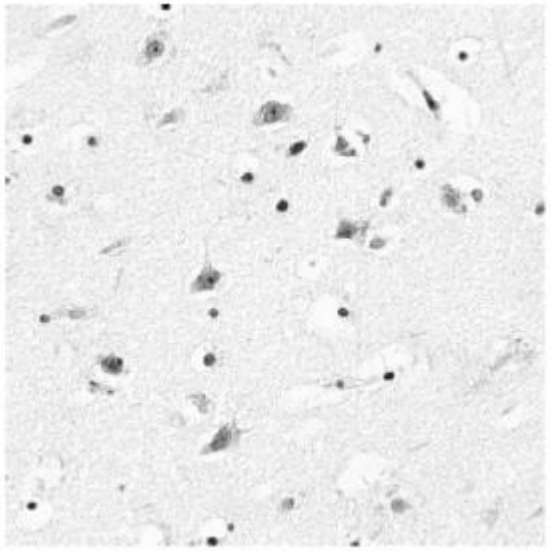
- Histograms tell us about the intensity/color distributions of a (part of a) picture
- Histograms can act as a sort of fingerprint of a picture
  - … but beware, two completely different pictures may have the same histogram

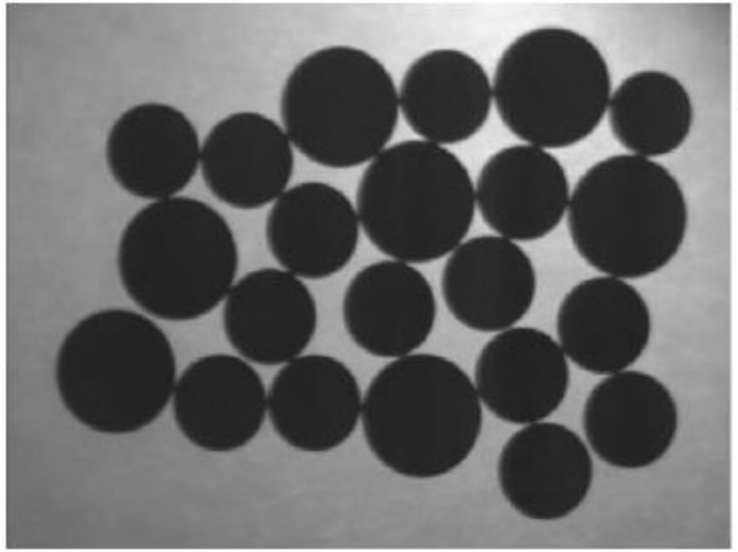


Segmentation

- Segmentation is about image analysis
- Not image manipulation

- The task:
  - Information versus noise
  - Foreground (object) vs. background

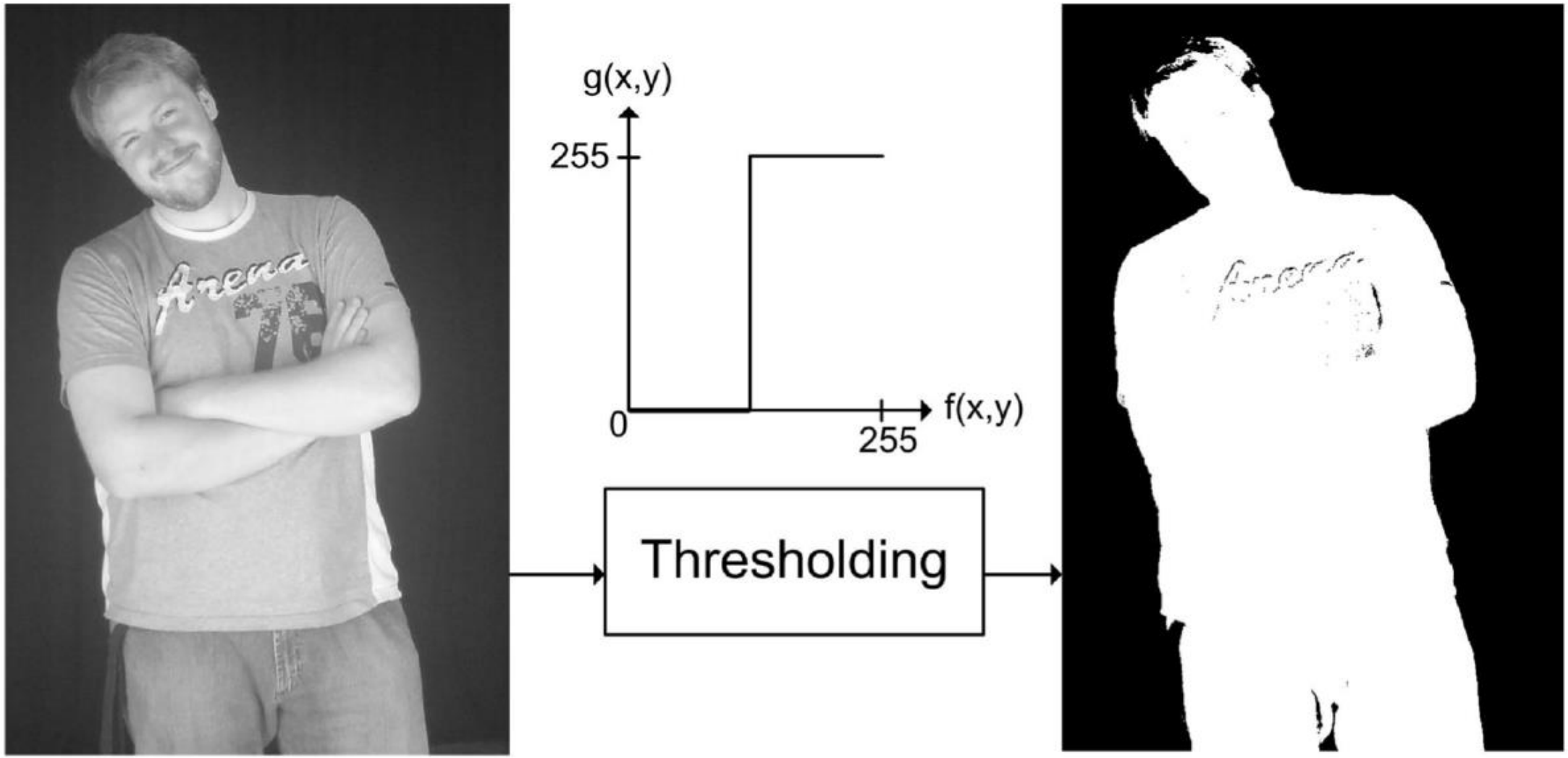




Thresholding

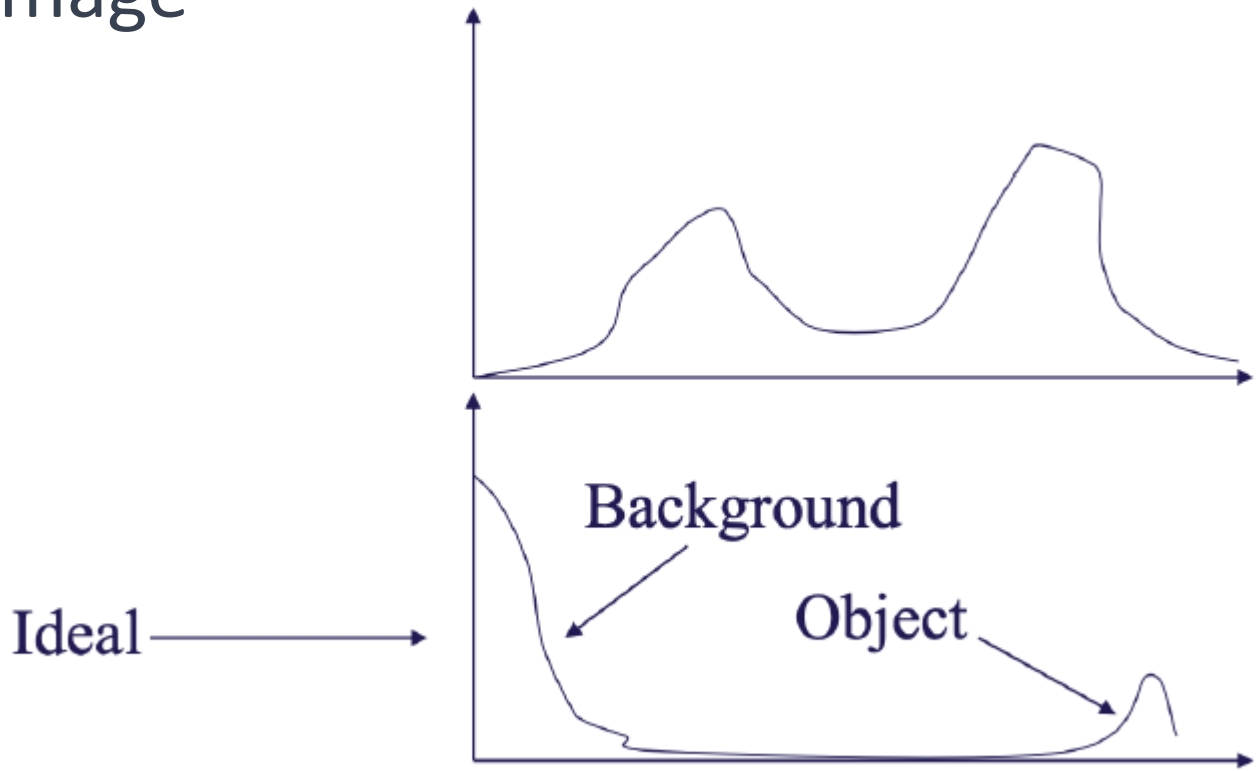
- Use graylevel mapping and the histogram
- When two peaks/nodes of a histogram correspond to object and noise
- Find a threshold value, T, that separates the two peaks. This process is called thresholding.
- Algorithm
  - If $f(x,y) > T$ then $g(x,y) = 1$, else $g(x,y) = 0$
  - (or reverse)
- Results: a binary image where object pixels =1 and noise/backgroun = 0

Thresholding



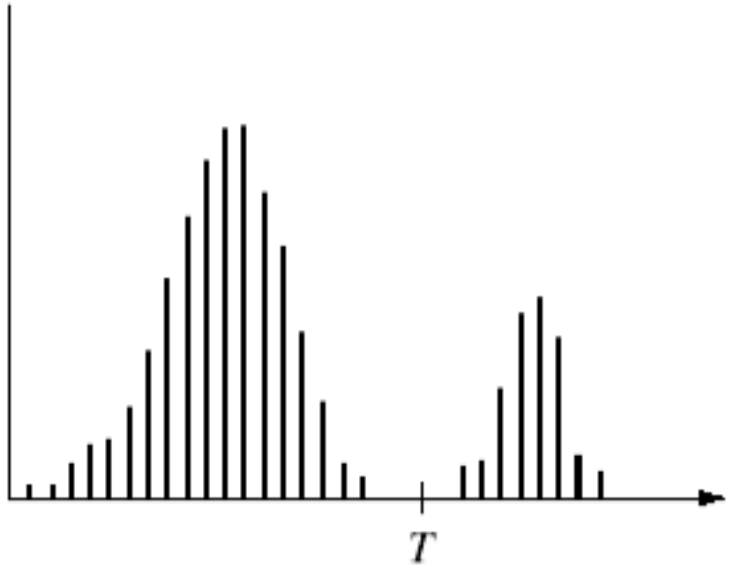
Segmentation

- Often obtained a bi-model histogram is the “sole” purpose of the image acquisition:
  - Lighting
  - Setup
  - Camera
  - Lense

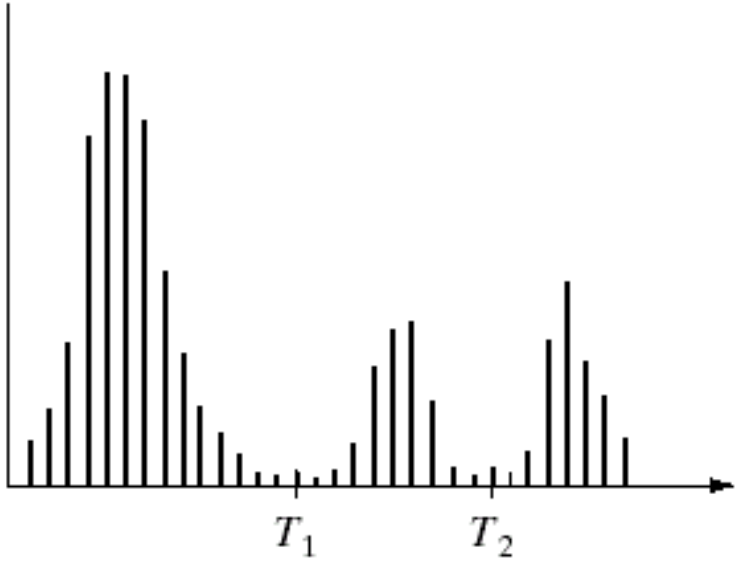


Thresholding

- Bimodal histogram
  - Single global threshold



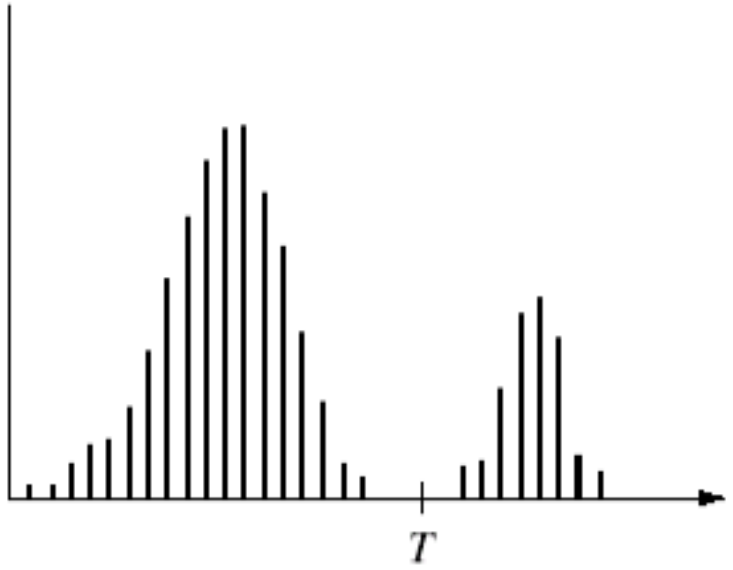
- Histogram with 3 or more ”tops” (often seen in medical applications)
  - Two or more global threshold
  - Region growing



Thresholding

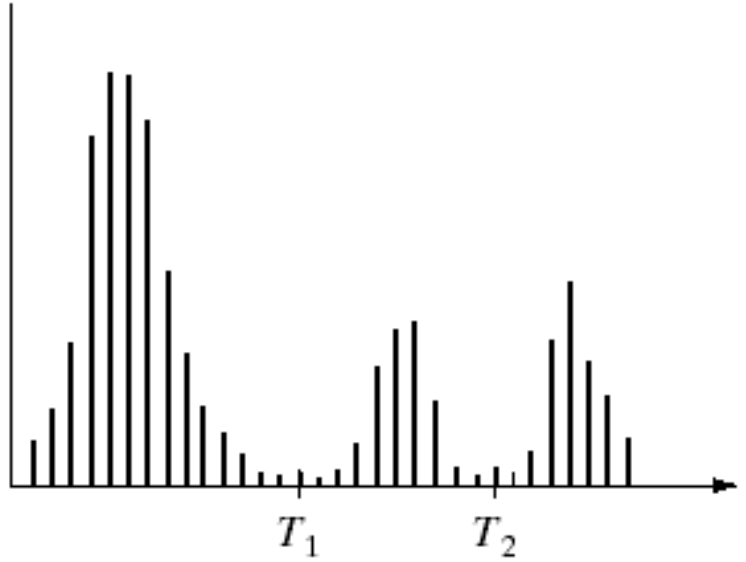
- Global threshold

$$g(x,y) = \begin{cases} 1 \; if \; f(x,y) > T \\ 0 \; if \; f(x,y) \le T \end{cases}$$



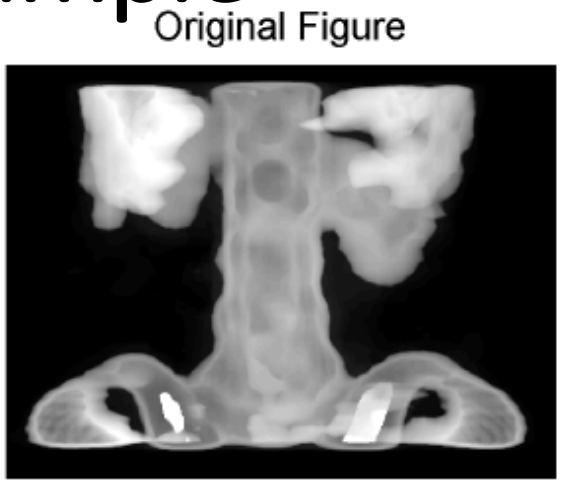
- Multiple thresholds

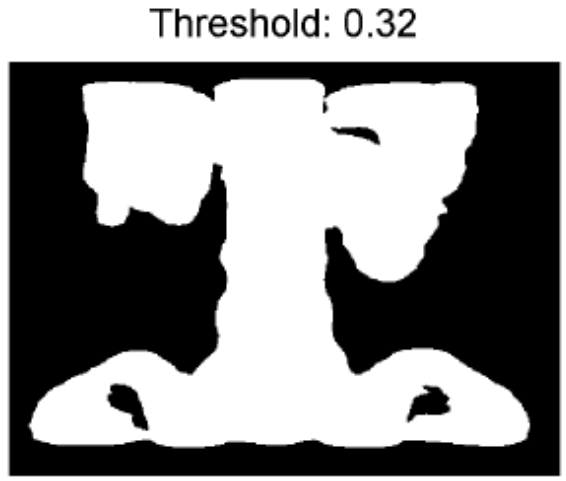
$$g(x,y) = \begin{cases} a & if \; f(x,y) > T_2 \\ b & if \; T_1 < f(x,y) \le T_2 \\ c & if \; f(x,y) \le T_1 \end{cases}$$

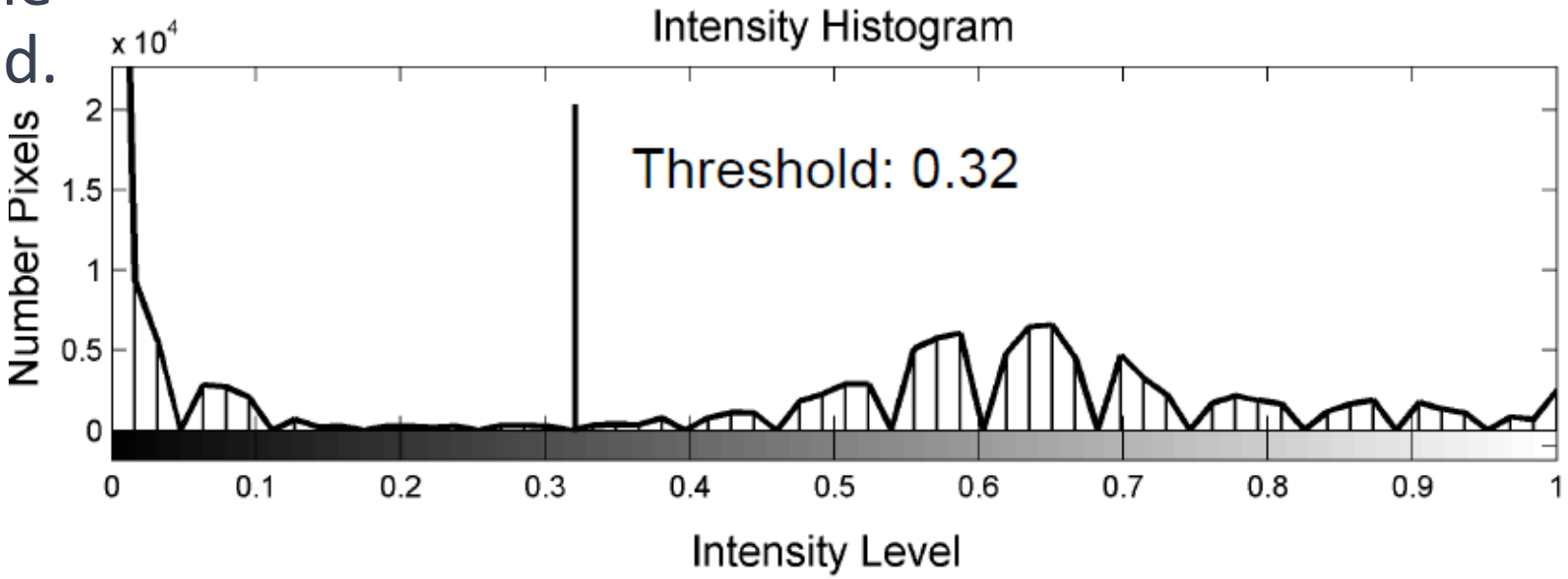


Thresholding – example

- The intensity histogram of an x-ray image of the spine (lower trace).
- Setting a threshold between the two peaks separates the image from the background.

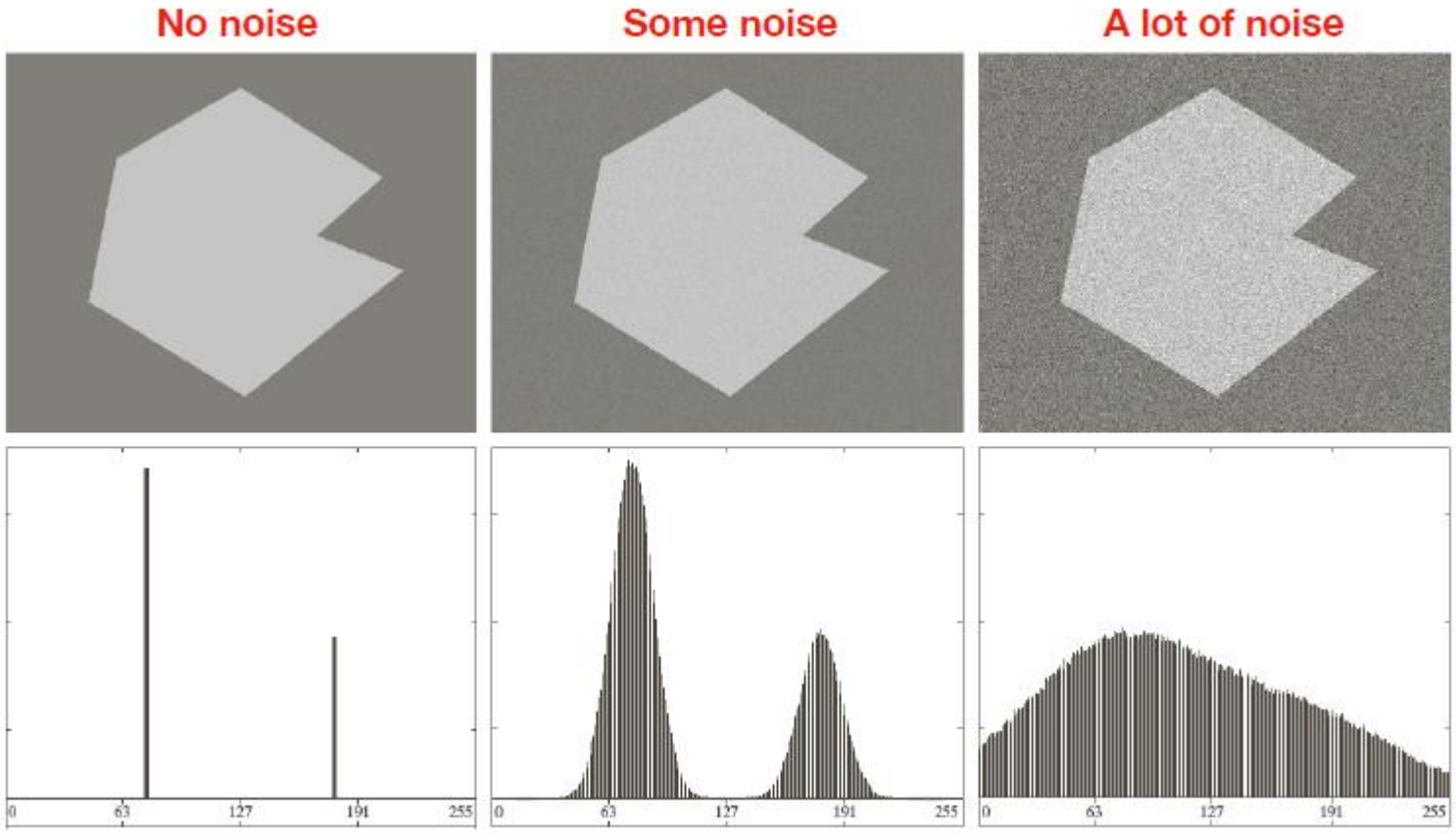






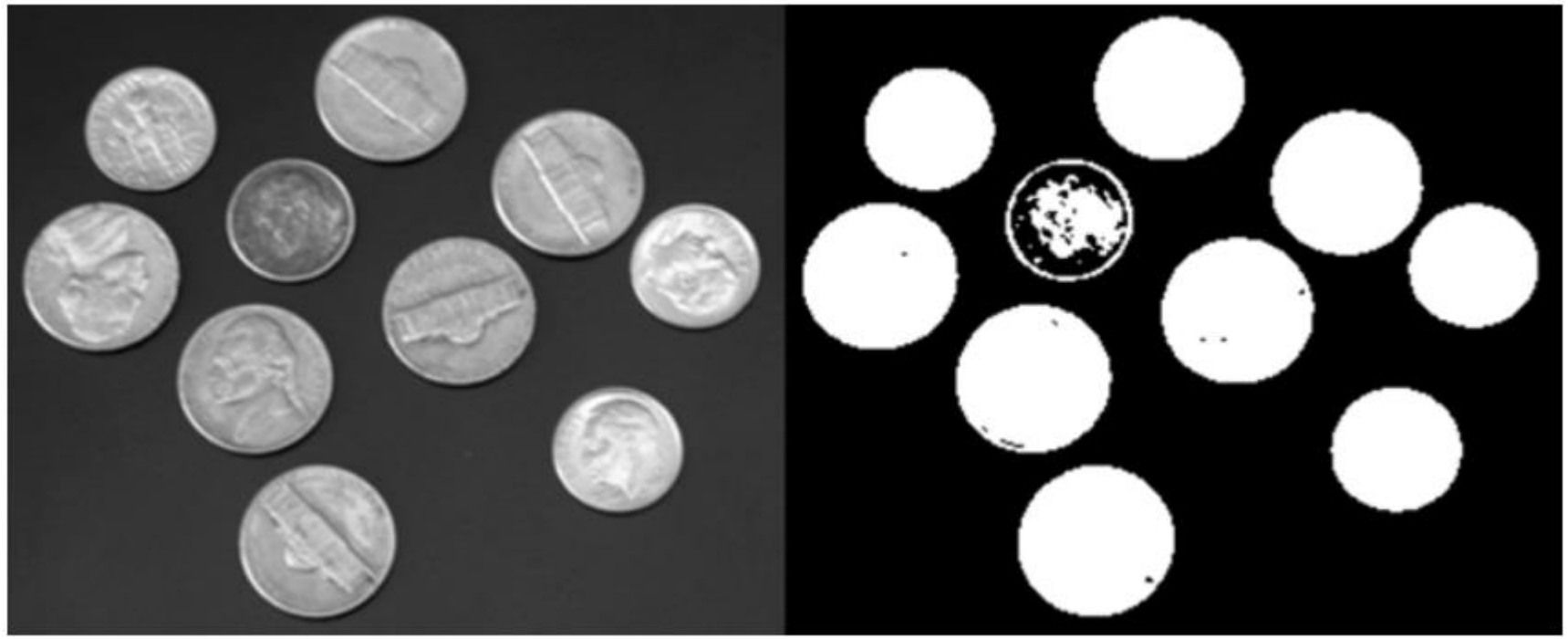
Thresholding

- Noise is a problem in signal processing but also true in images



Global Thresholding

- Automatic (basic for Bimodal histogram)
- Assumption: A separation of the histogram peaks —> separation of the background and object(S)



Global Thresholding

- Automatic (basic for Bimodal histogram)
- Assumption: A separation of the histogram peaks —> separation of the background and object(s)

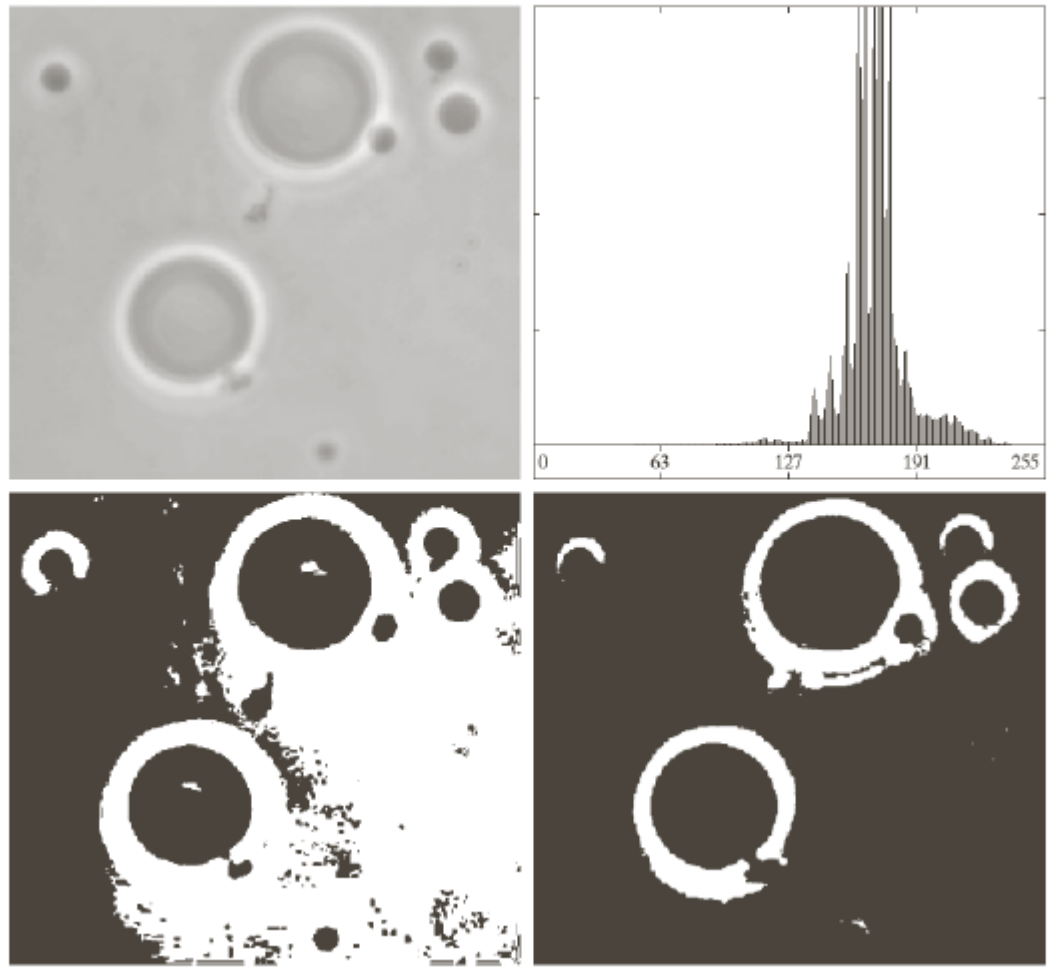
Approach:

- 1. Select an initial estimate for T
- 2. Segment the image using T —> image is divided into two groups G1 and G2.
- 3. Compute the average gray level value m1 and m2 for the pixels in regions G1 and G2.
- 4. Compute a new threshold: $$T = \frac{1}{2}(m_1 + m_2)$$
- 5. Repeat step 2 - 4 until the difference in T is smaller than a predefined parameter T0

Variance minimization: Otsu’ method

- Aim: Find a global threshold by minimizing the overall within-class variance
- One threshold strategy is based on the concept of minimizing the variance between presumed foreground and background elements.
- Although the method assumes two different gray levels, it works well even when the distribution is not bimodal.
- The approach uses an iterative process to find a threshold that minimizes the variance between the intensity values on either side of the threshold level (Otsu’s method).
  - => It makes it calculations for every intensity value

Otsu example



a b
c d

FIGURE 10.39
(a) Original image.
(b) Histogram (high peaks were clipped to highlight details in the lower values).
(c) Segmentation result using the basic global algorithm from Section 10.3.2.
(d) Result obtained using Otsu's method.

Basic automatic thresholding

Otsu's method

Segmentation recap

- About splitting the image into parts:
  - Interesting
  - Not interesting

- Thresholding is one way of solve this
  - If a sufficient nice bimodal histogram is present

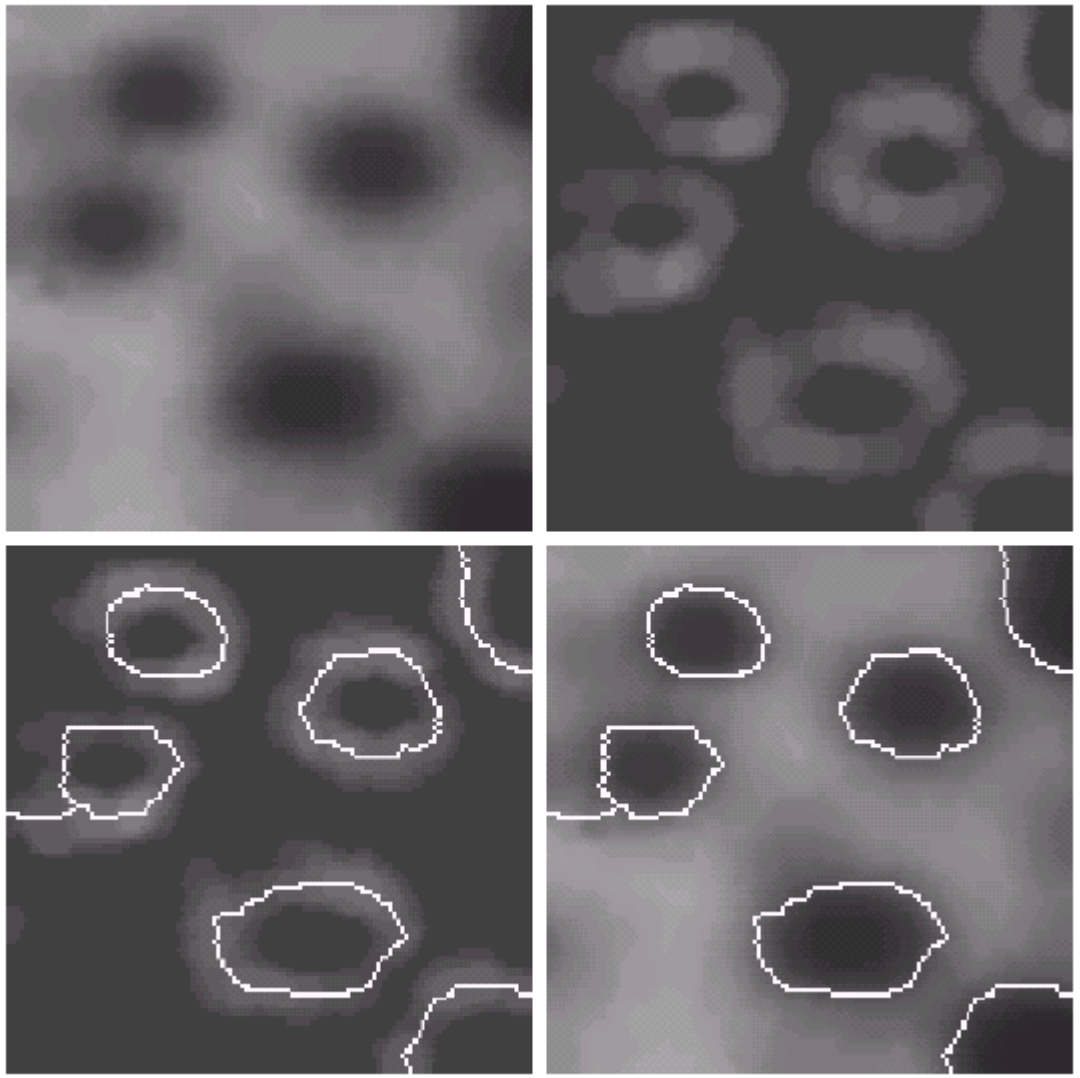


Image arithmetic

- **Arithmetic +, -, x, /**
  - Perform simple math on pixel level
  - Apply the operation as point processing

- **Example: Subtraction (-)**
  - Subtract two images of the same scene at different time instances

    => find the change over time
  - Help you to judge the nature of noise in your setup

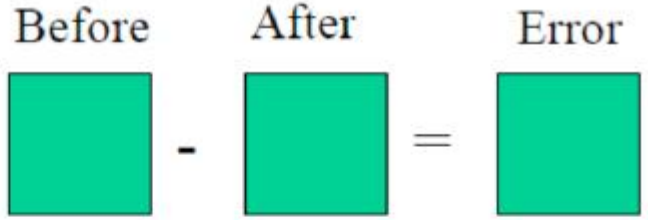
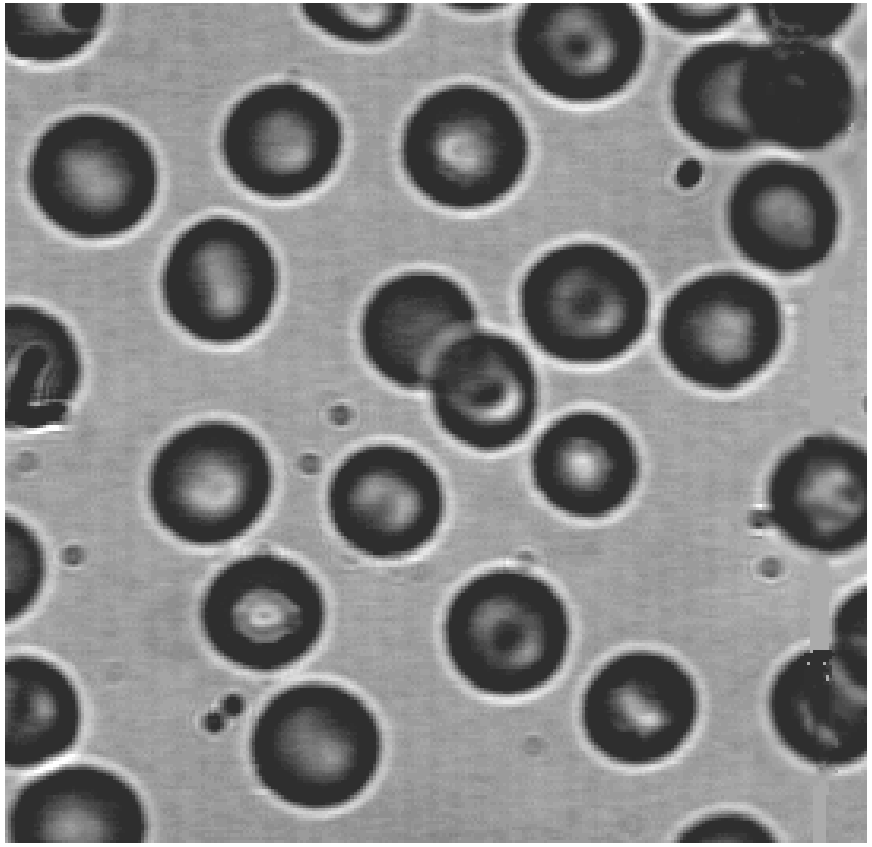


Image arithmetic – Apply mask

- What will the next image look like if we multiply (*) these 2 images?



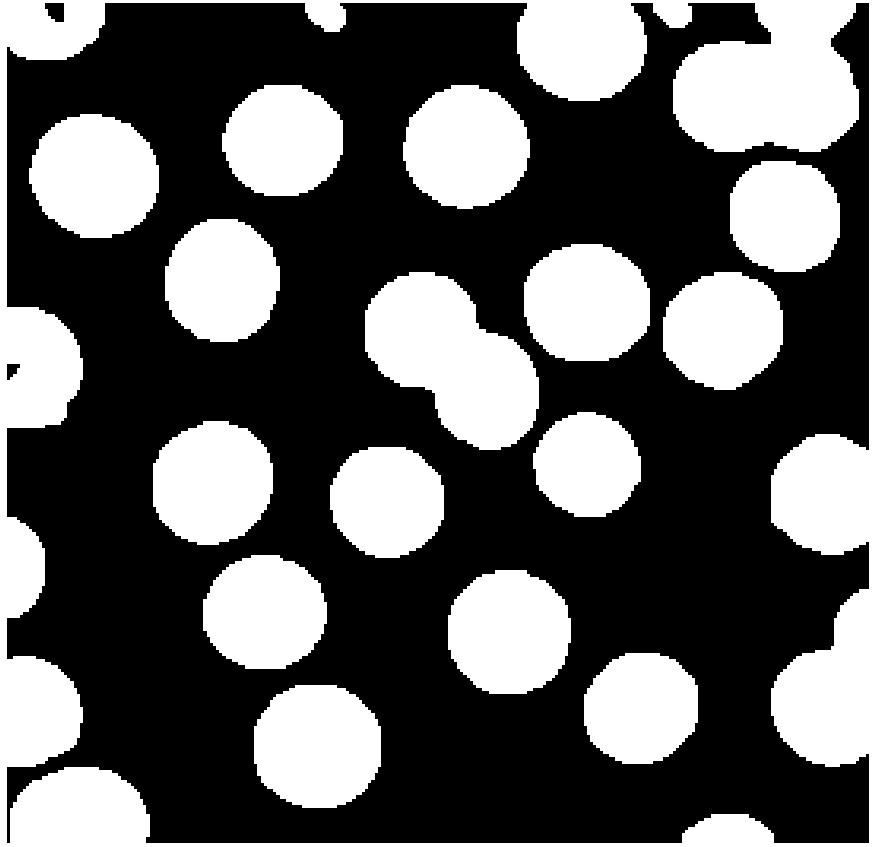


Image arithmetic – Apply mask

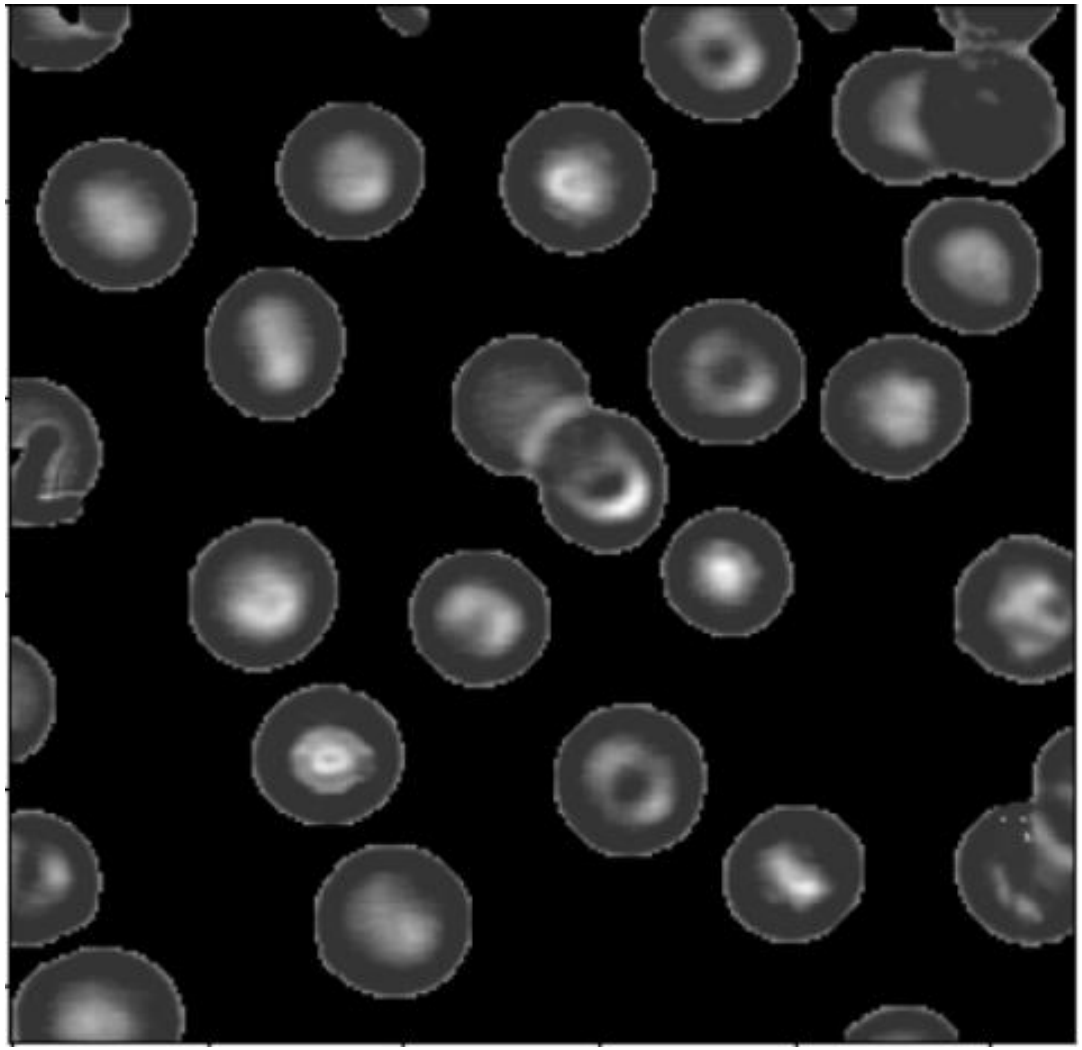


Image arithmetic

- Undesired effects may happen, when computing with images: **Overflow / Underflow**
- Example: Two grey value images A and B with pixel ranges: [0,255]. Their pixel sum may be larger than 255!
- For example: 124+202=326
- Subtraction results may be smaller than 0! For example: 124-202=-78

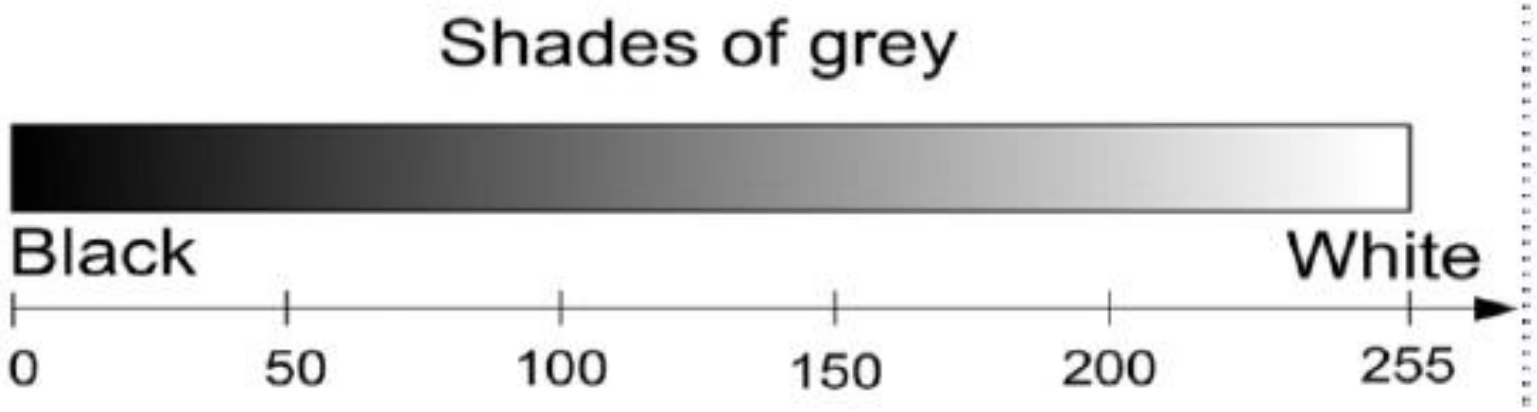


Image arithmetic – solution

- Use an intermediate image:
- Pixel values are float values (32 bits / 4 bytes)
- Can store almost any number
- Write computation results into intermediate image
- Rescale intermediate image to values [0,255] and write results into 8 bit image
- For subtraction you may be interested only in the absolute difference

$$Normalized_{image} = \frac{image - min_{val}}{max_{val} - min_{val}}$$

Summary

- Point operation is when we only take one pixel at the time to do this operation.
- It doesn’t matter the order of it + no information about the neighbours pixel is required. That is why it is a simple and fast operation