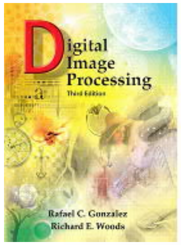
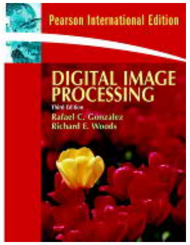
Lecture 2 – Point processes

Wednesday, November 1

Ch. 2.6-2.6.4
3.1-3.3 in
Gonzales & Woods





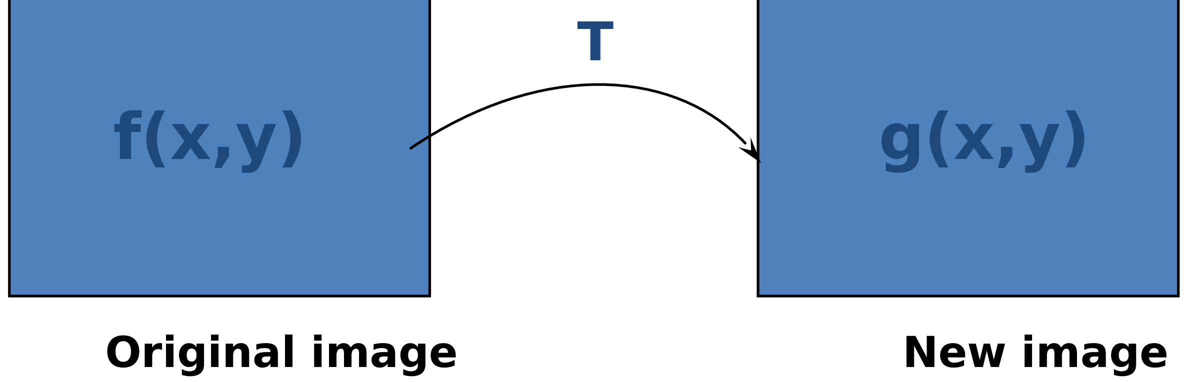
Damian Matuszewski
damian.matuszewski@it.uu.se
Centre for Image analysis
Uppsala University

Image enhancement

https://youtu.be/LhF_56SxrGk

- an image processing technique to enhance certain features of the image

Image processing

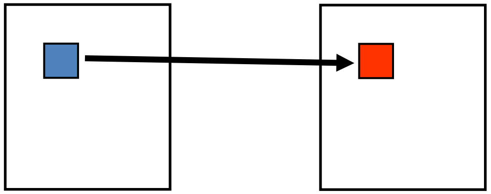


- We want to create an image which is ”better” in some sense.
- For example
- Image restoration (reduce noise)
- Image enhancement (enhance edges, lines etc.)
- Make the image more suitable for visual interpretation
- **Image enhancement does NOT increase image information**

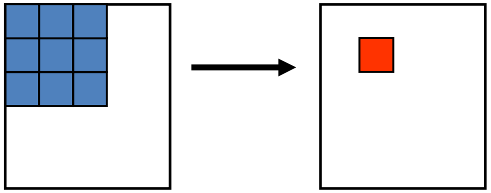
Image processing

- can be performed in the:

- **Spatial domain**
  - Point processes → Lecture 2
    - Works per pixel

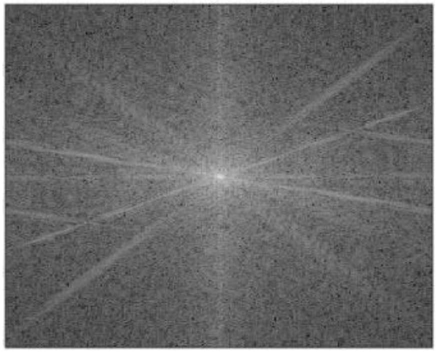


  - Spatial filtering → Lecture 3 *(Filip)*
    - Works on small neighborhood

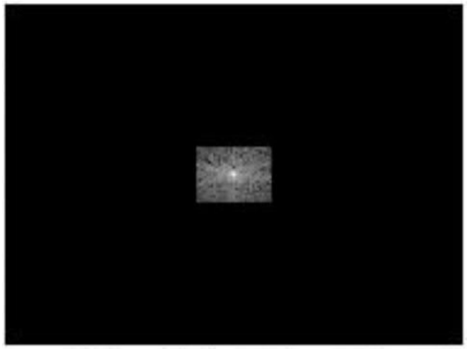


- **Frequency domain** → Lecture 4 *(Filip)*

Original image in spatial domain



Original image in frequency domain



Processed image in frequency domain

Processed image in spatial domain

Problem solving using image analysis: fundamental steps

image acquisition

preprocessing, enhancement

segmentation

feature extraction, description

classification, interpretation, recognition

result

Knowledge about the application

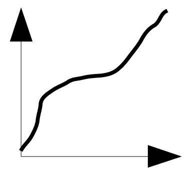
Overview

i. repetition

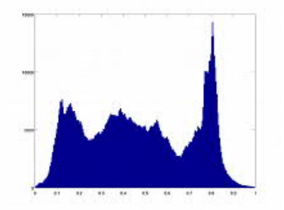
ii. image arithmetics

'+','-', '*'

iii. intensity transfer functions

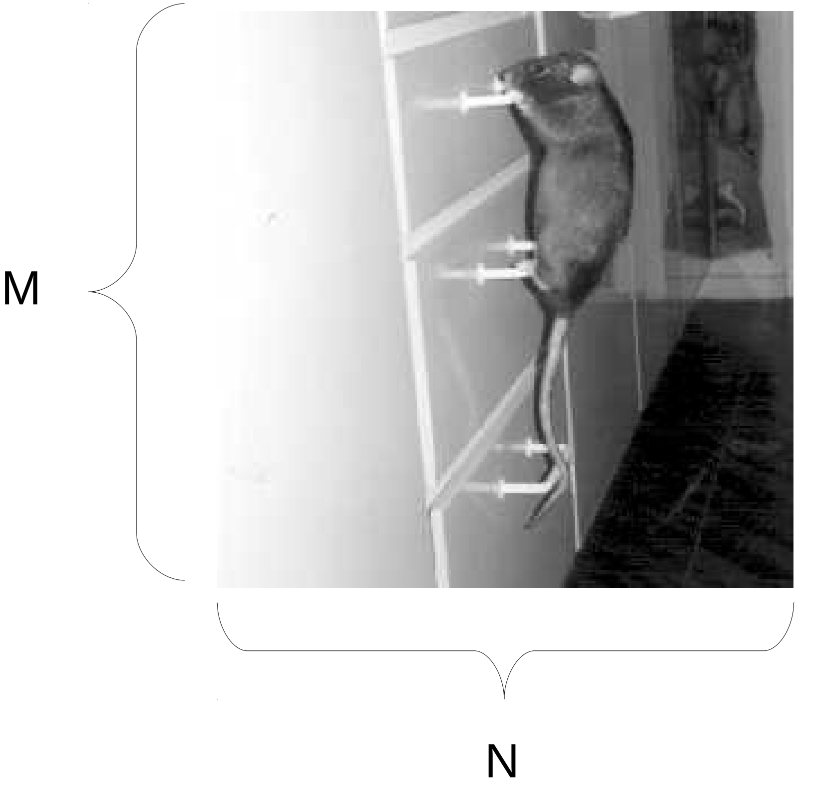


iv. histograms and histogram equalization



Last lecture

- Digitization
  - Sampling in space (x,y)
  - Sampling in amplitude (intensity)
- Pixel/Voxel
- How often should you sample in space to see details of a certain size?



Bit depth

- Number of bits that are used to store the intensity information
- Images are typically of 8- or 16-bit
  - 1bit = 2^1 → 2 steps (0,1)
  - 2 bit = 2^2 → 4 steps
  - 8 bit = 2^8 → 256 steps
  - 16 bit = 2^16 → 65 536 steps
- <u>Not continous!</u>

2 gray levels, 1bit/pixel

64 gray levels, 6bit/pixel

256 gray levels, 8bit/pixel

I. Image arithmetics in the spatial domain

Image arithmetics

- $\mathbf{A(x,y) = B(x,y) \circ C(x,y)}$ for all x,y.

  B, C → images with the same (spatial) dimensions

  → images + constant value

  ○ can be

  - Standard arithmetic operation: **+**, **-**, *****.
  - Logical operator (binary images): **AND**, **OR**, **XOR**,...

- Any pitfalls? -> bit depth, negative values and pixel saturation

Arithmetics with binary images

min value

max value

image1 image2

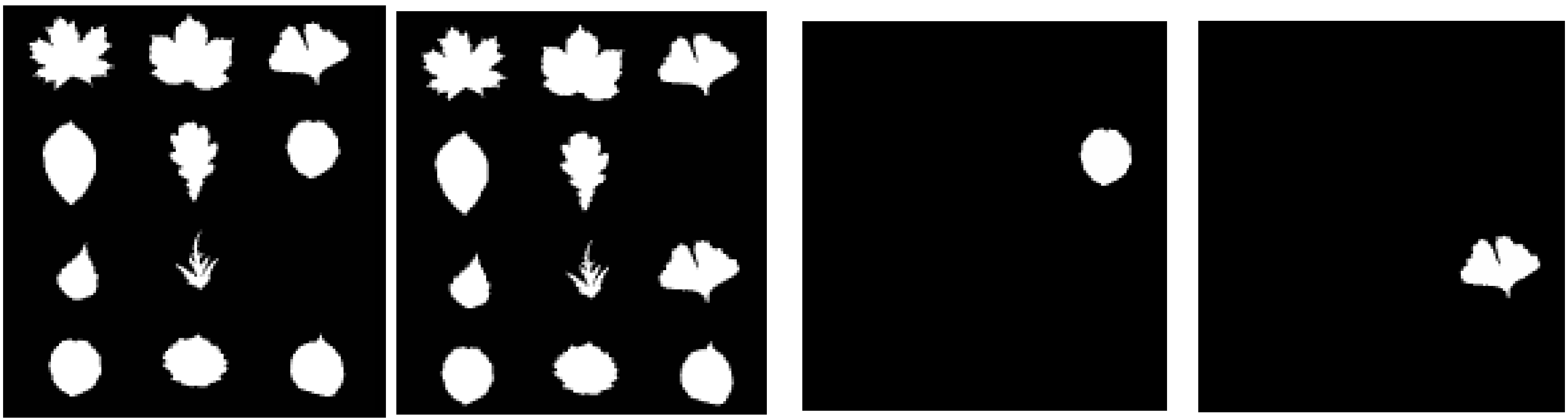
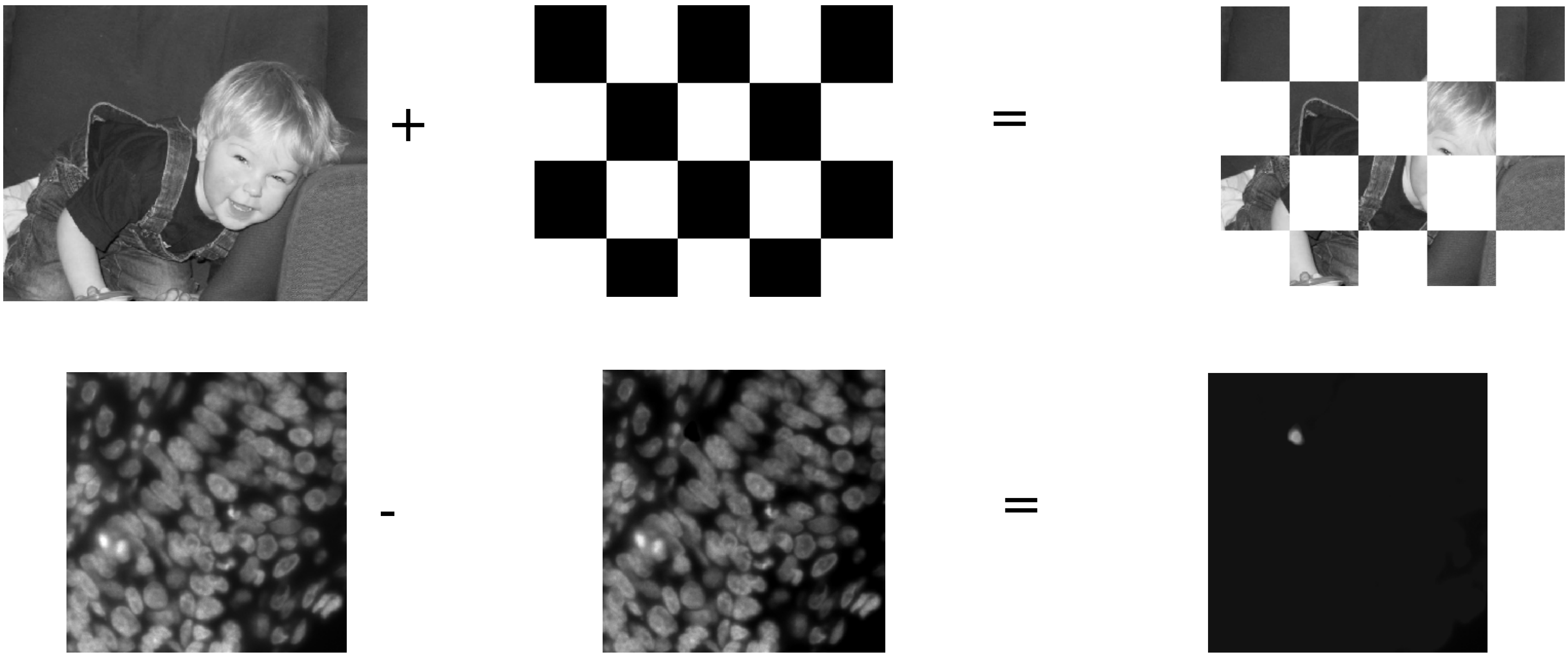
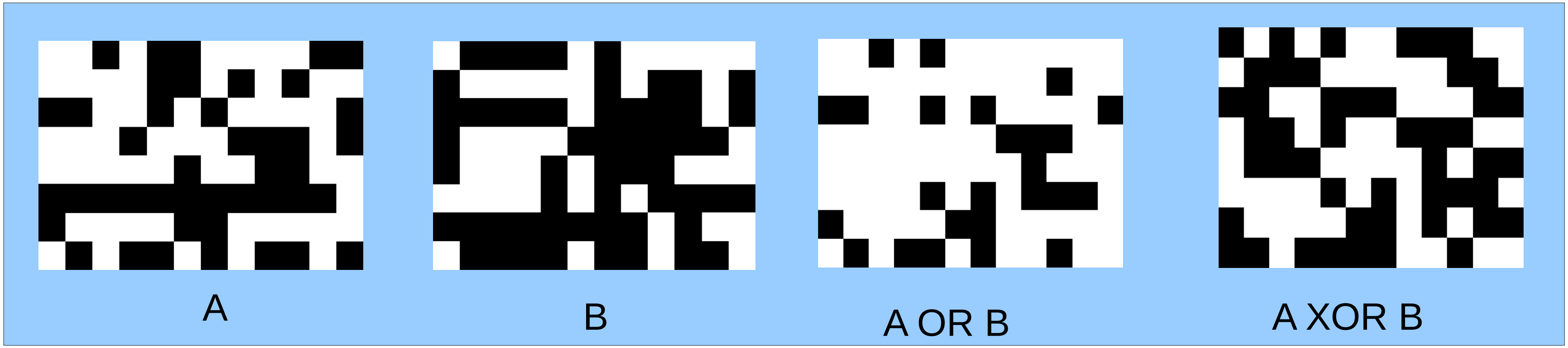


image1-image2 image2-image1

Arithmetics with greyscale images



Logical operations on binary images



INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

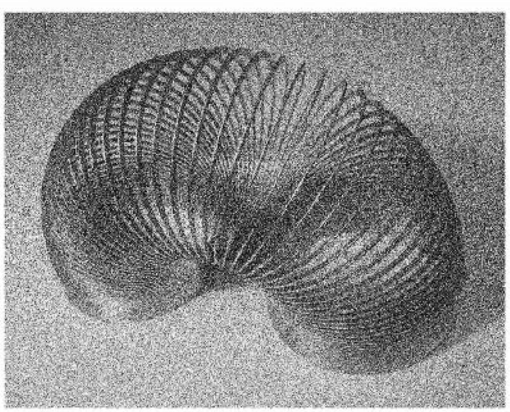
INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Applications

- **Noise reduction** using image mean or median

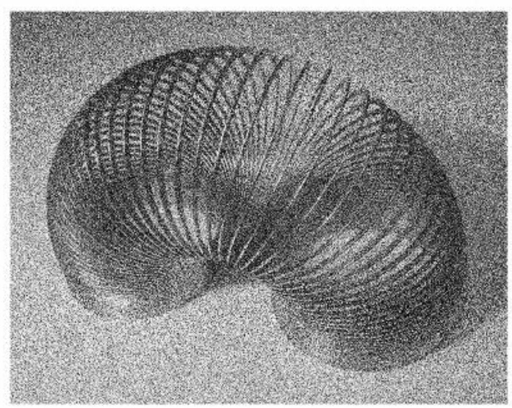
$$I = \frac{1}{n} \sum_{k=1..n} I_k$$

I_1



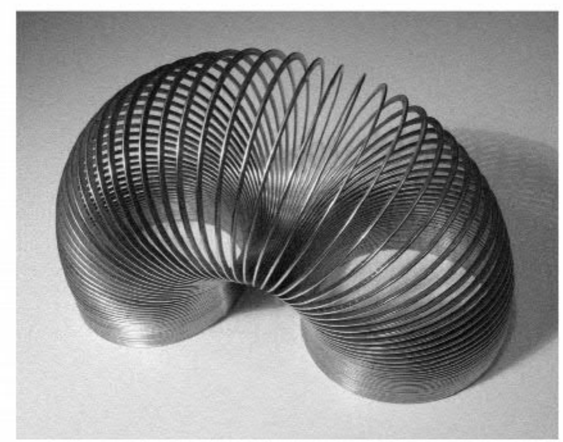
+ … +

I_n



=

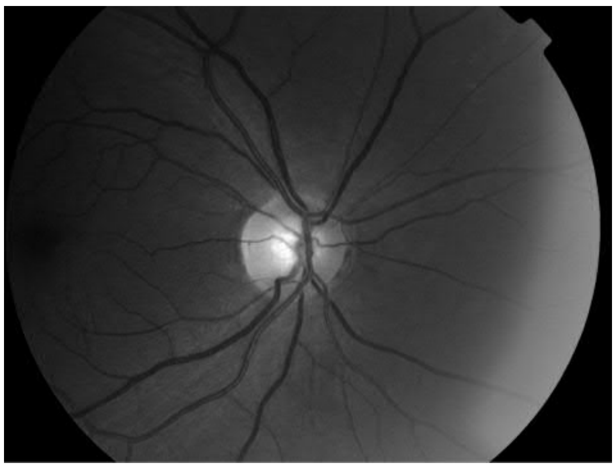
I



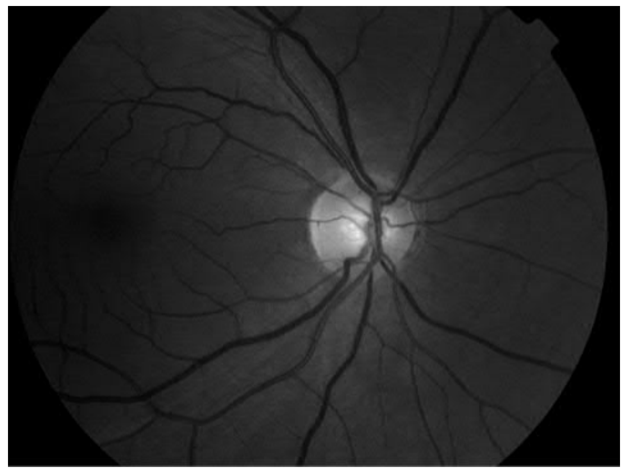
- Useful in astronomy, low light (night) pictures

Applications

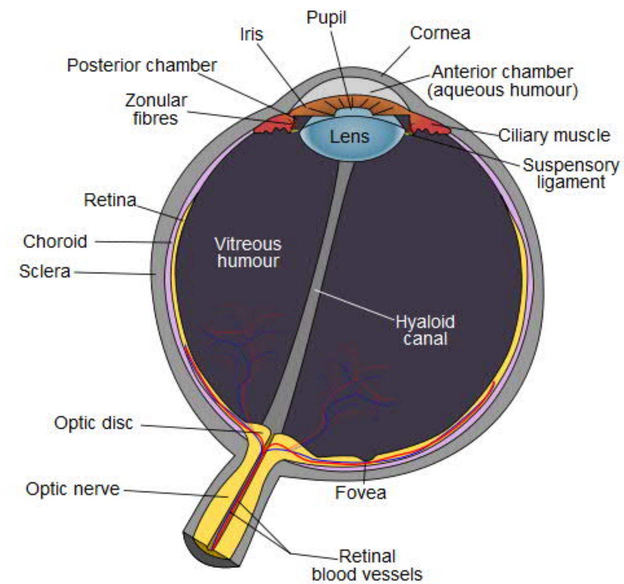
- **Change detection** using subtraction

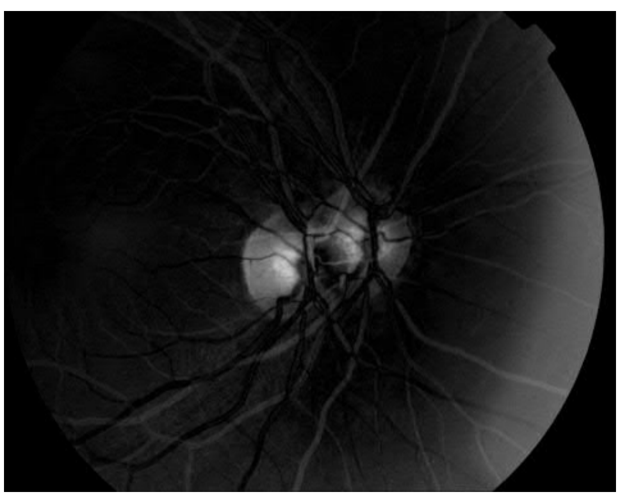


2015

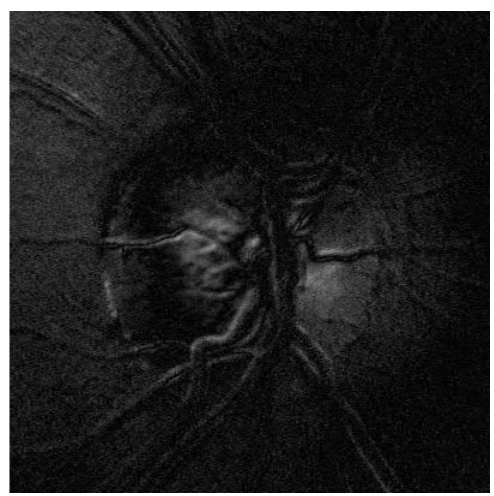


2016





direct difference



difference after registration

Applications

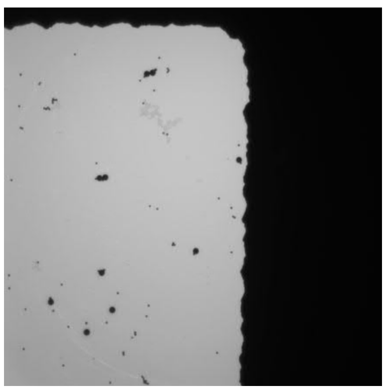
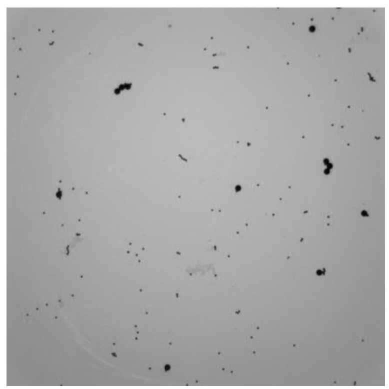
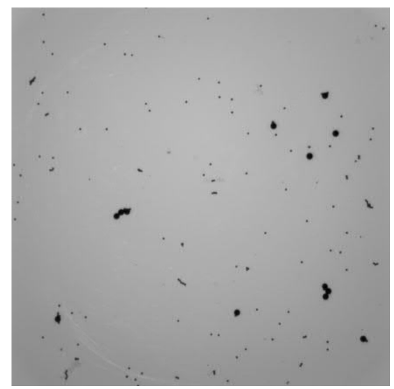
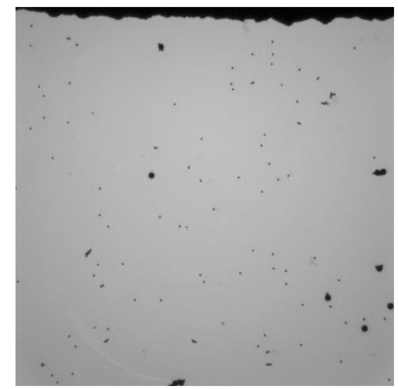
- **Change/motion detection** using subtraction

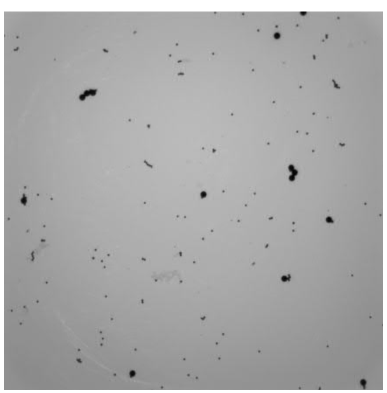
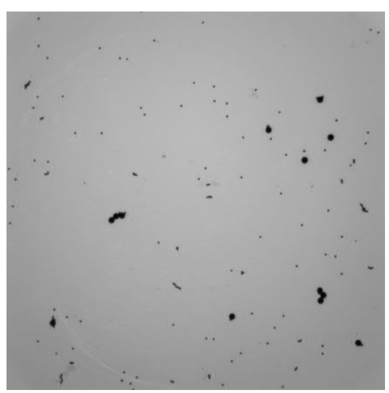
Applications

- **Background removal**

image - background image

- Creating a background image

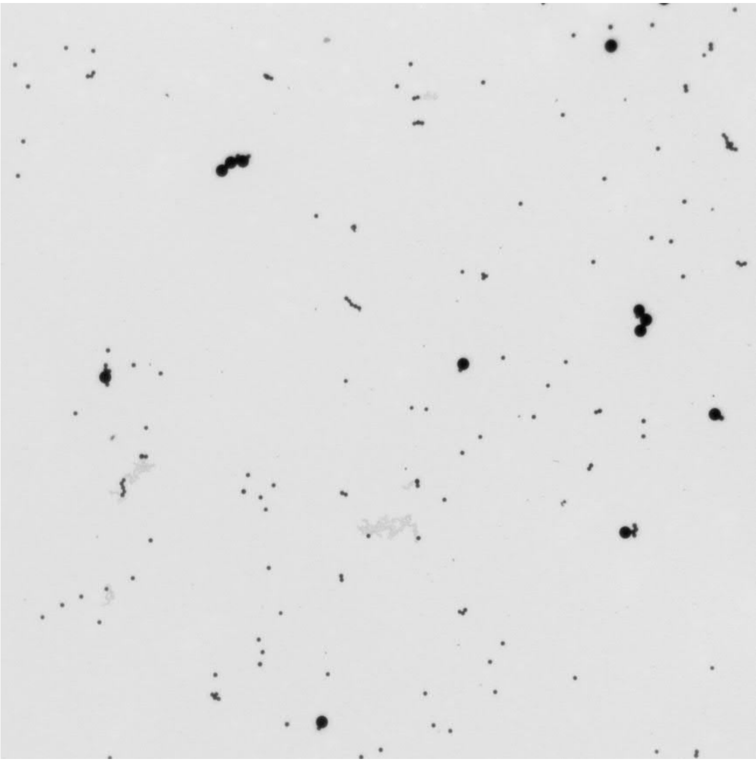





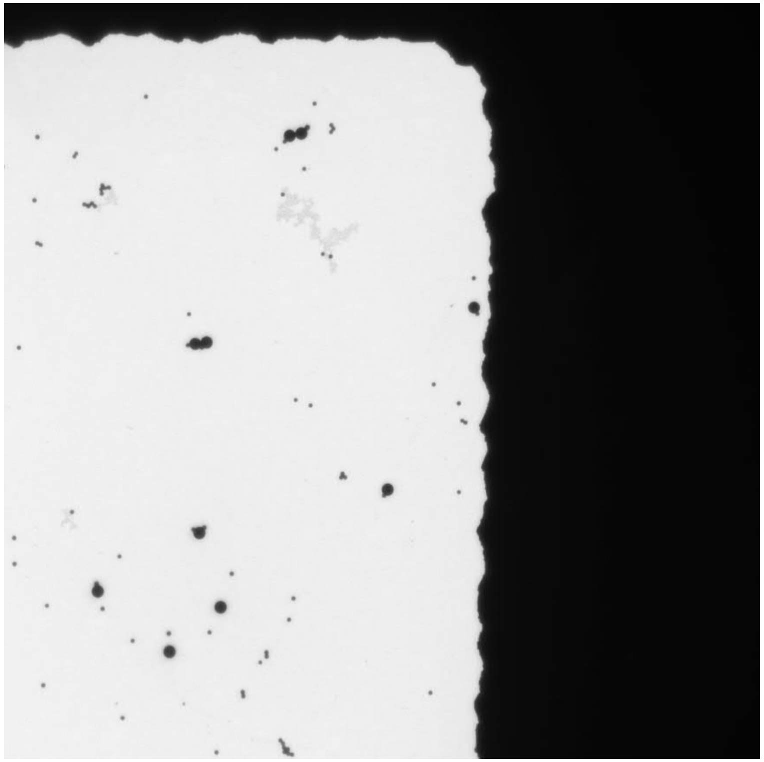



Max or median of the pixel intensities at all positions.

Applications

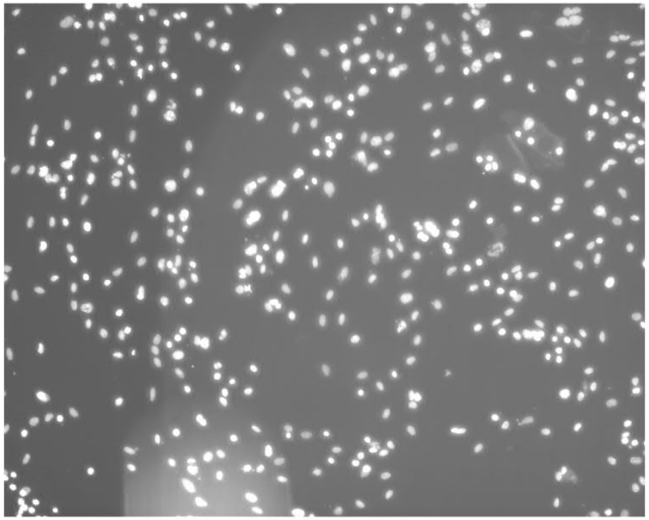
- **Background removal** - result



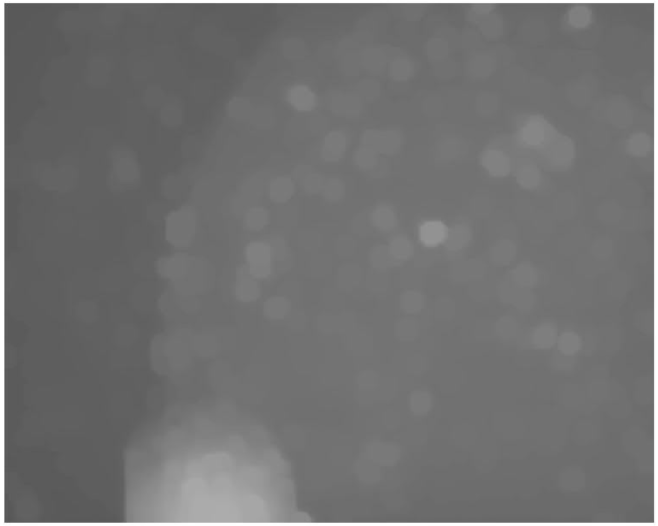


Applications

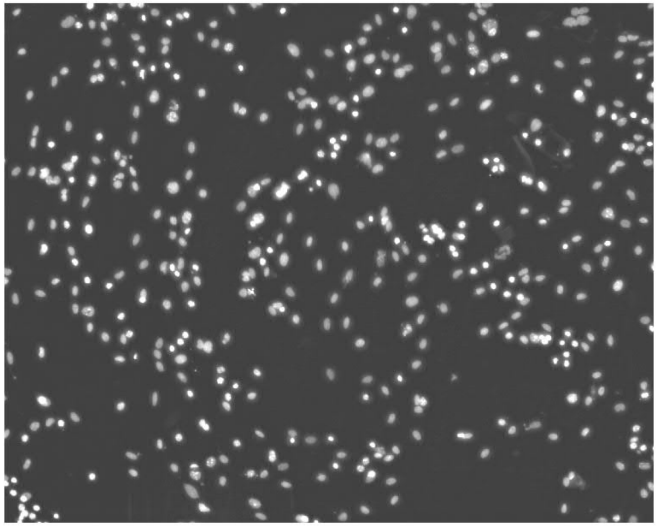
- Subtracting a background image/correcting for uneven illumination



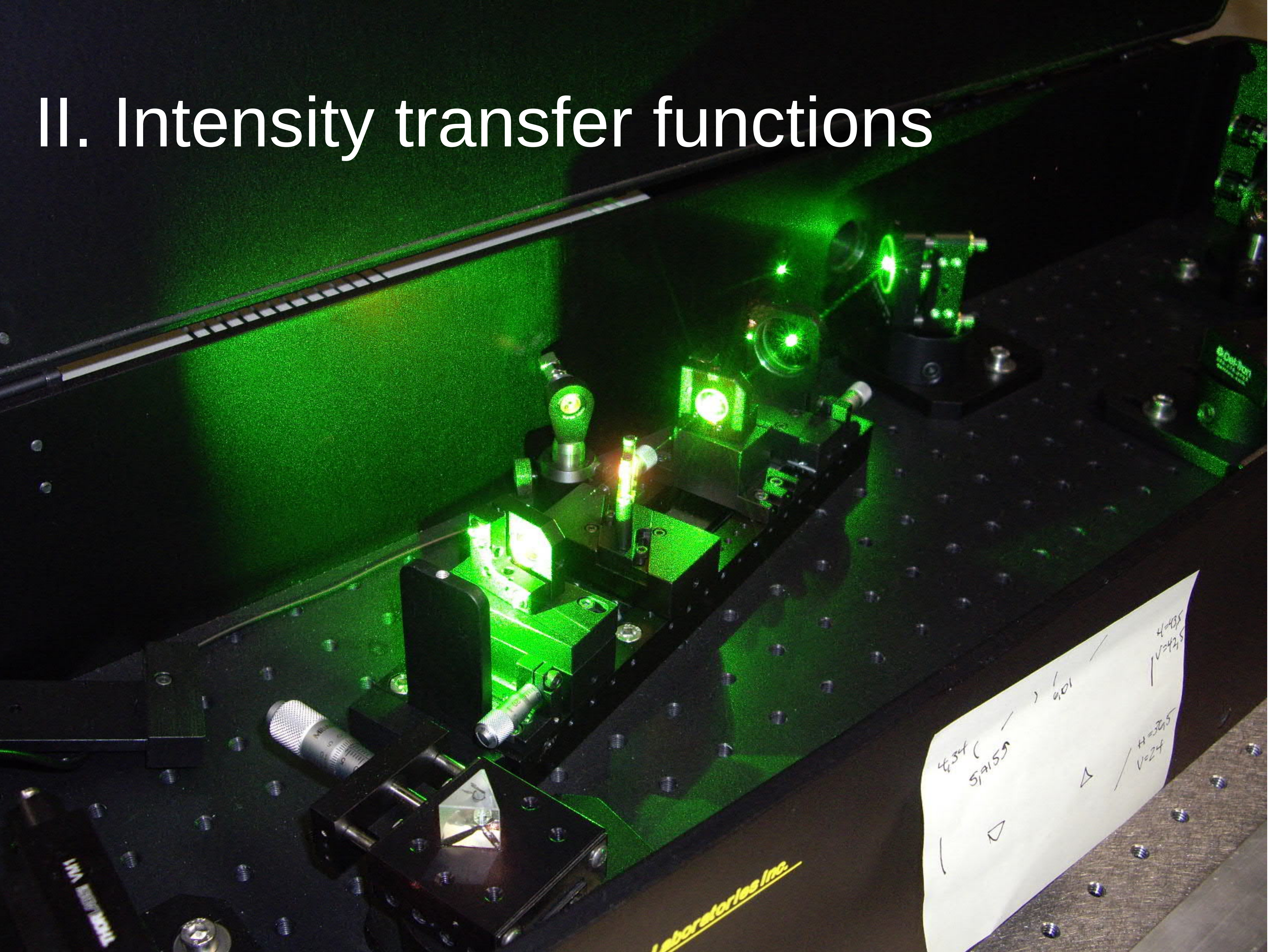
-



=



II. Intensity transfer functions



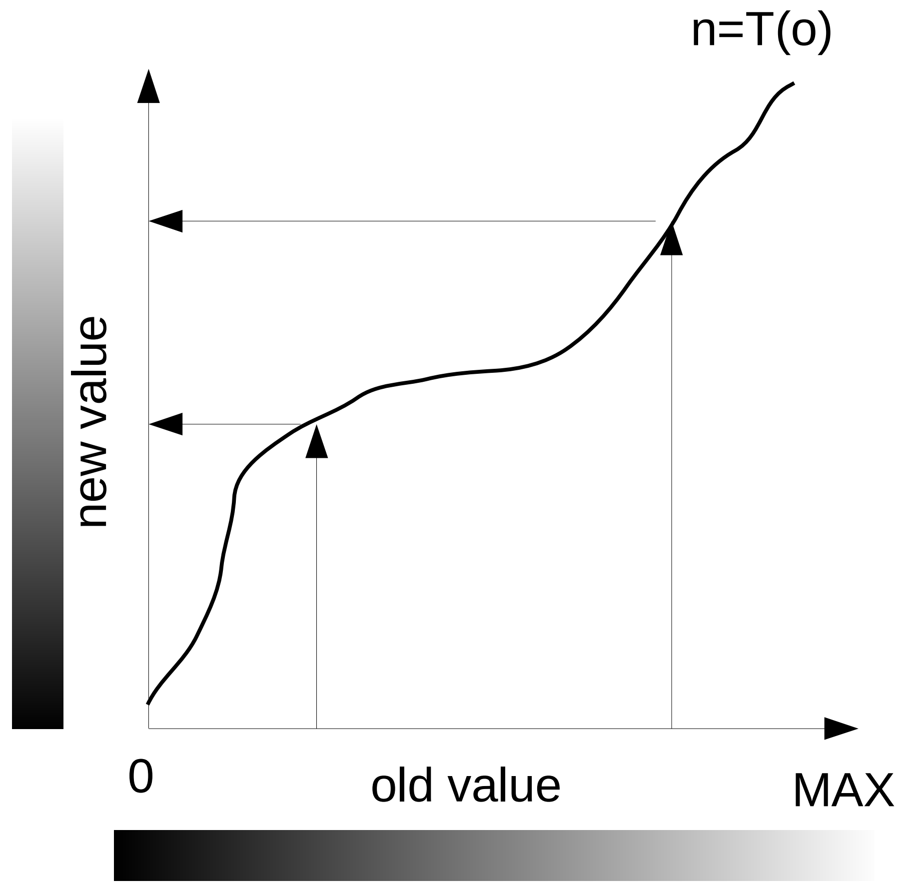
Intensity transfer functions

$$g(x, y) = T f(x, y)$$

i. **linear** (neutral , negative, contrast, brightness)

ii. **smooth** (gamma, log)

iii. **arbitrarily**

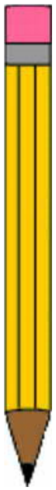


The negative transformation

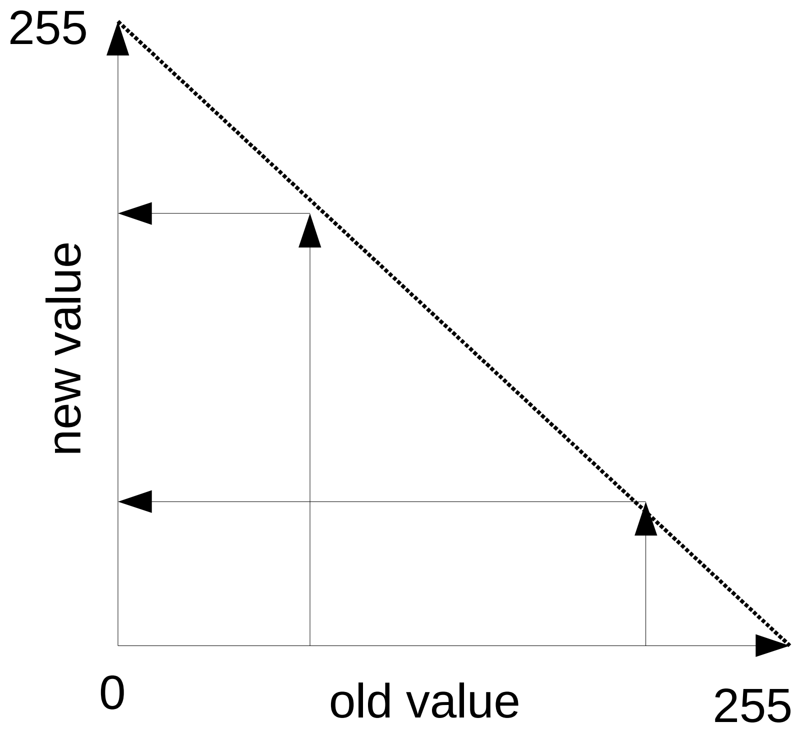
$$g(x, y) = max - f(x, y)$$

- For eight bit image:

$$g(x, y) = 2^8 - 1 - f(x, y)$$



255	254	253
125	130	110
4	3	0

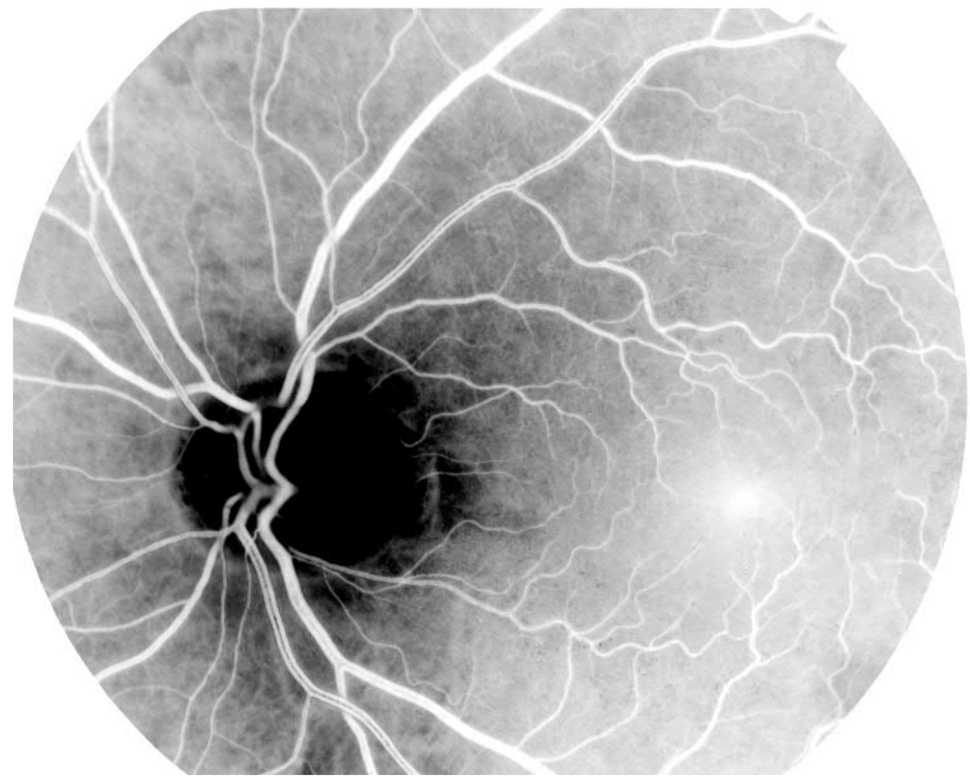


The rules of how to transfer values from the old image to the new one.

The negative transform

- Useful in medical image processing

Original



Negative

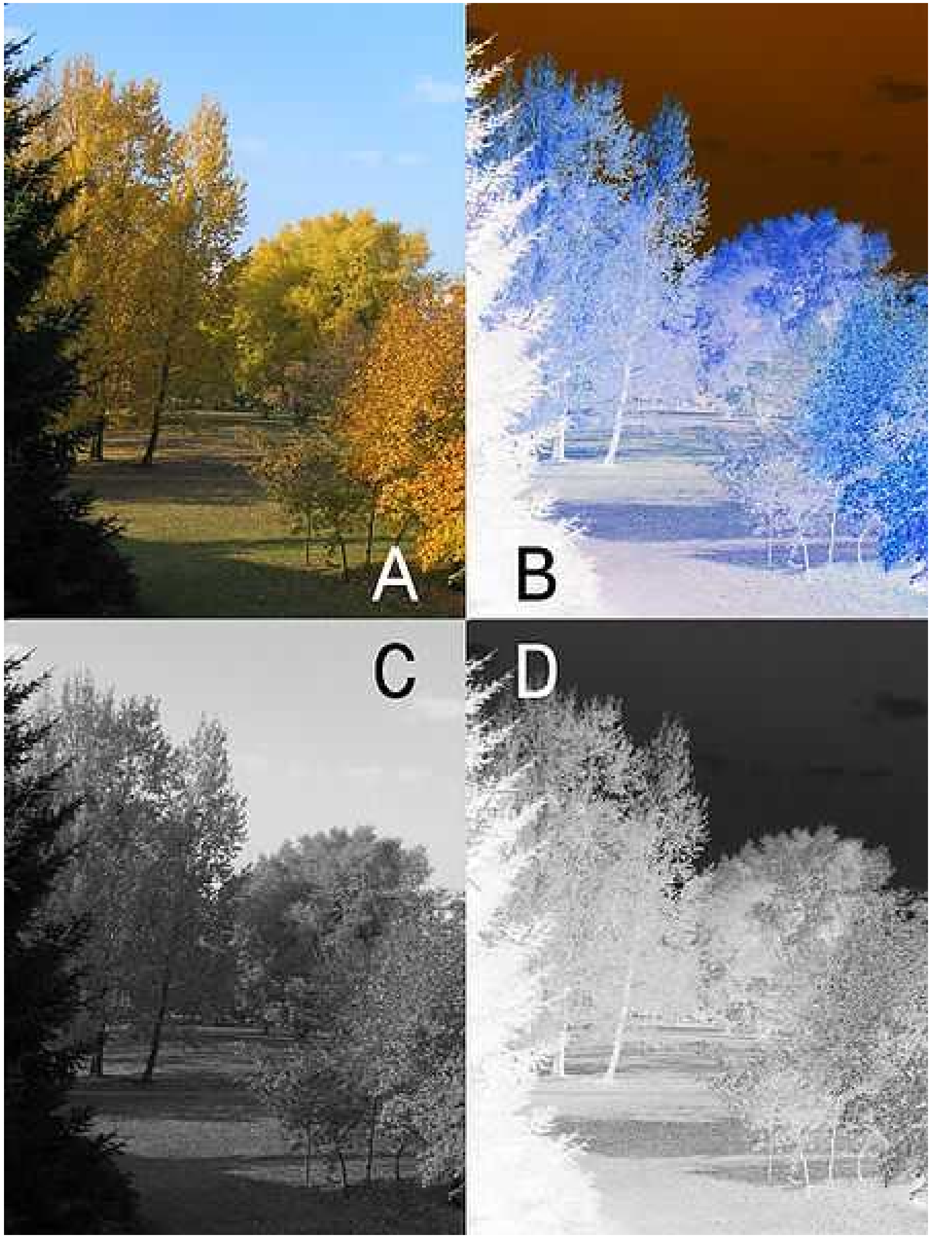
The negative transform

original digital mammogram

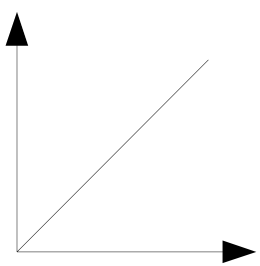
image negative to enhance white or gray details embedded in dark regions

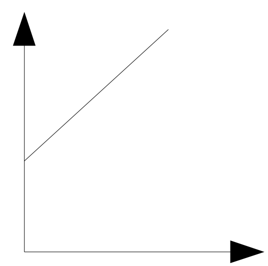
The negative transformation

- Careful with color images

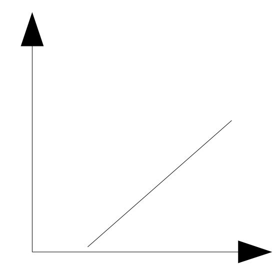


Brightness



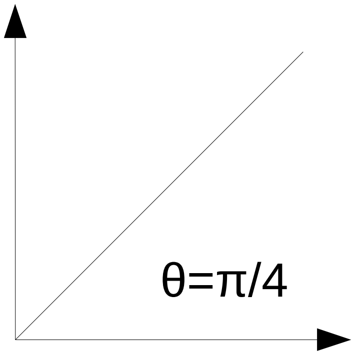


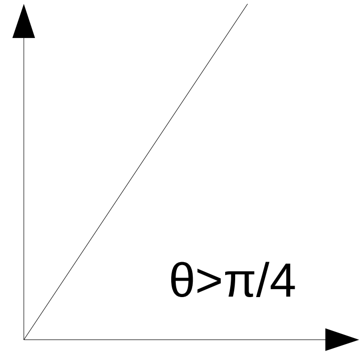
$$g(x, y) = f(x, y) + C$$



$$g(x, y) = f(x, y) - C$$

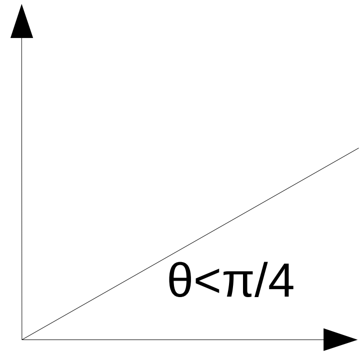
Contrast





$$g(x,y) = f(x,y) \times C$$

$$C > 1$$



$$g(x,y) = f(x,y) \times C$$

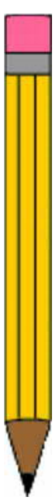
$$C < 1$$

- Decrease the brightness by 10

$$g(x,y) = f(x,y) - 10$$

- Decrease the contrast by 2

$$g(x,y) = f(x,y) \times 0.5$$

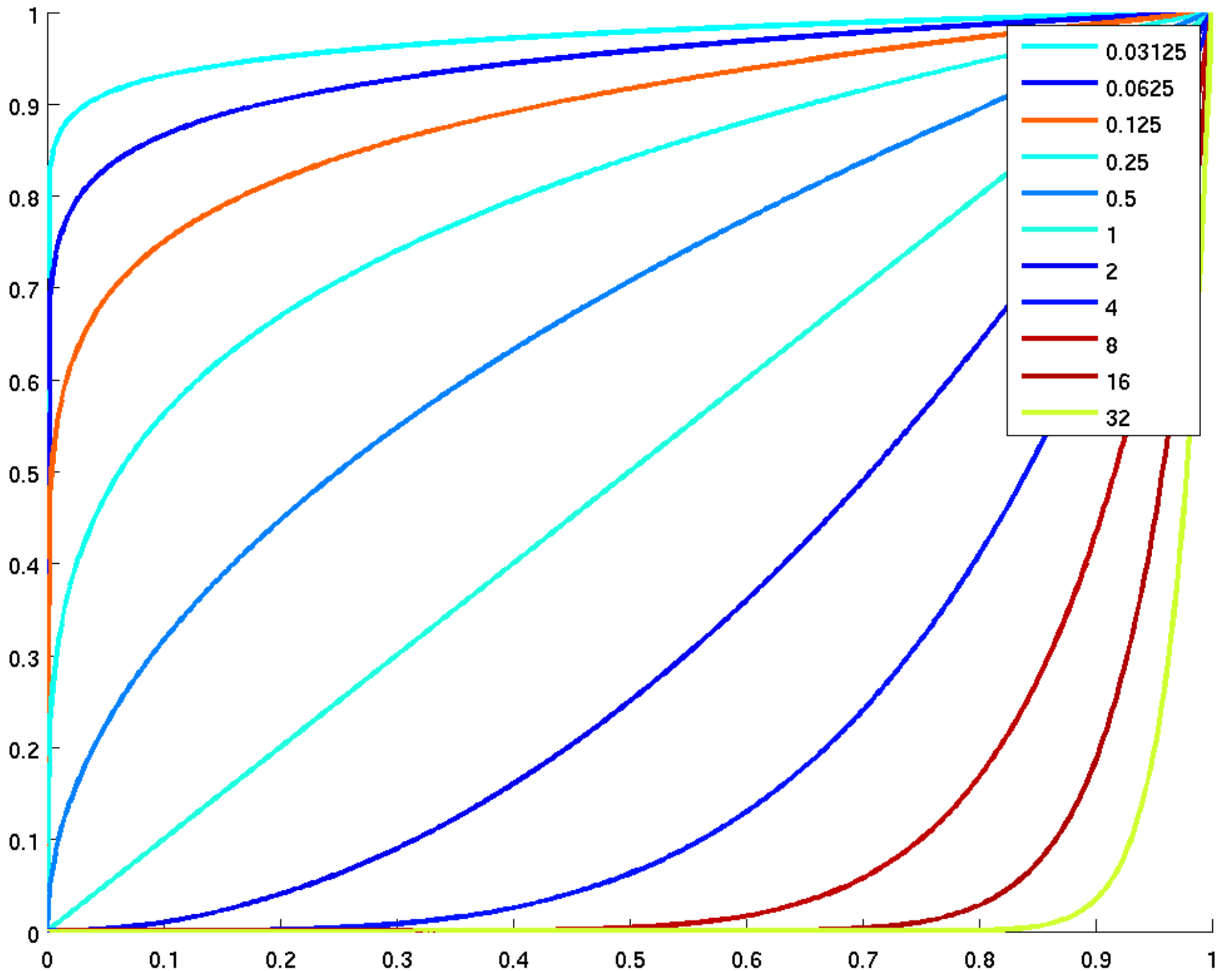


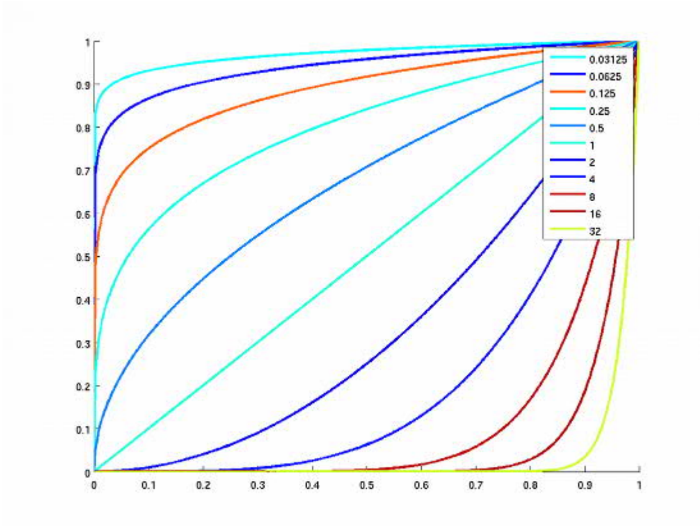
255	254	253
125	130	110
4	3	0

Gamma transformation

$$g(x, y) = C \times f(x, y)^{\gamma}$$

- Computer monitors have $\gamma \sim 2.2$
- Eyes have $\gamma \sim 0.45$
- Microscopes should have $\gamma = 1$





γ=0.25

γ=1

γ=4

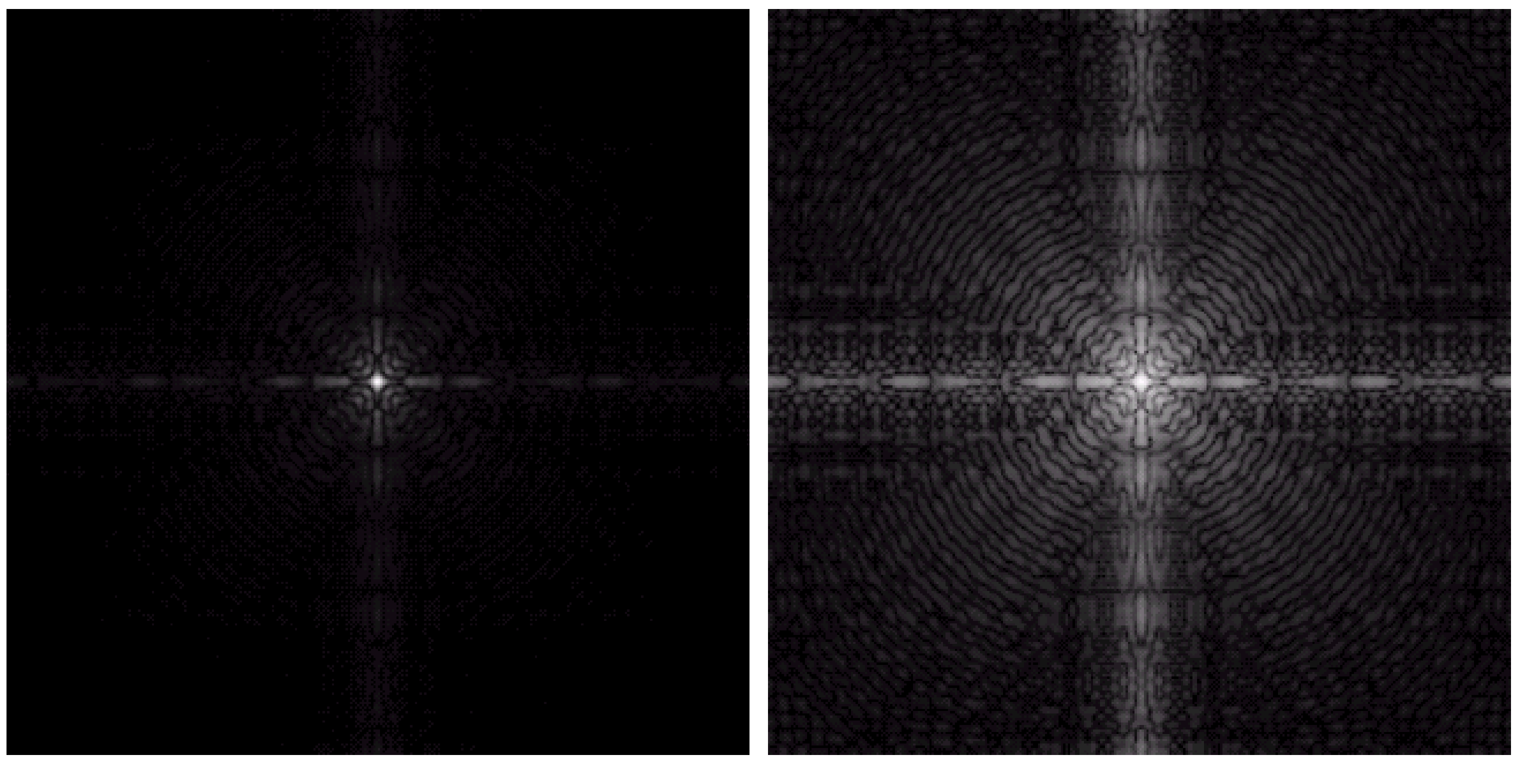
Log transformations

- Log transformation to visualize patterns in the dark regions of an image

$$g(x, y) = C \log(1 + f(x, y))$$

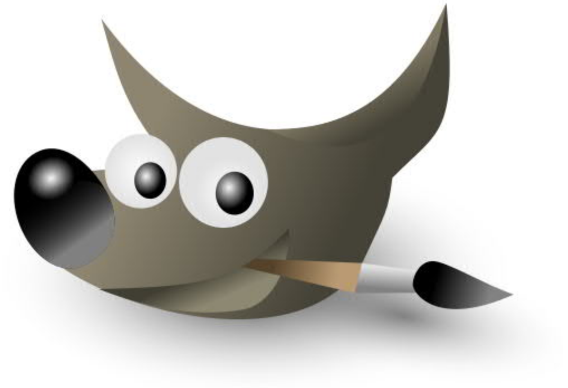
a b

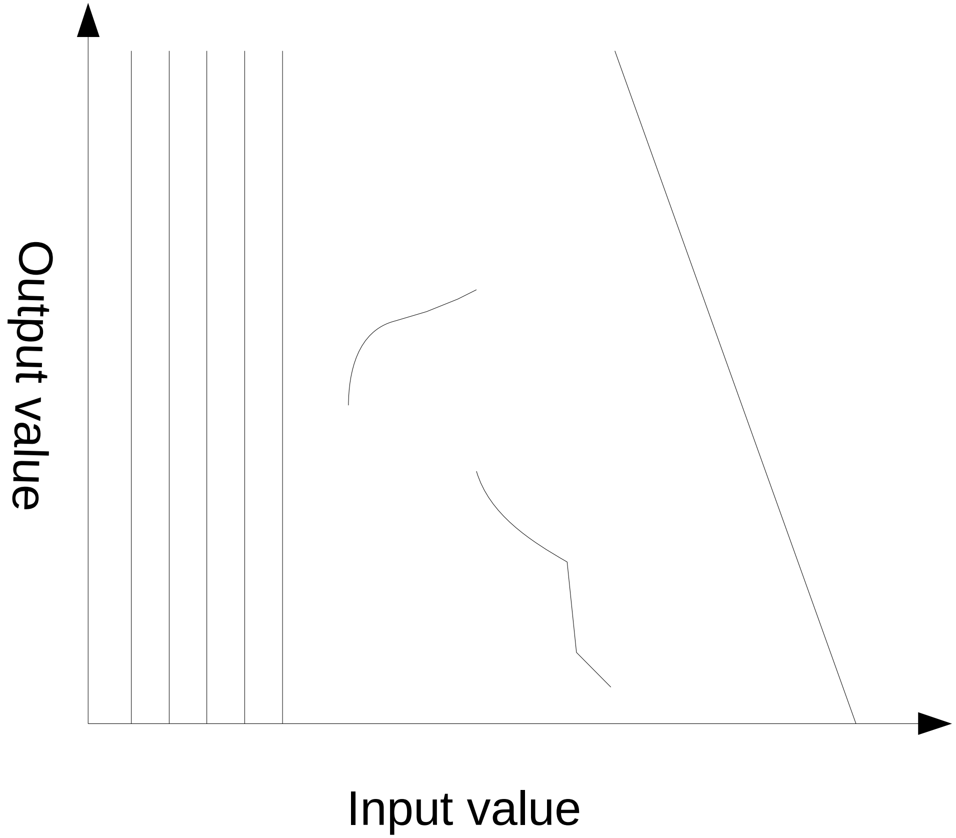
FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Arbitrary transfer functions

- Only one output per input.
- Possibly non-continuous.
- Usually no inverse





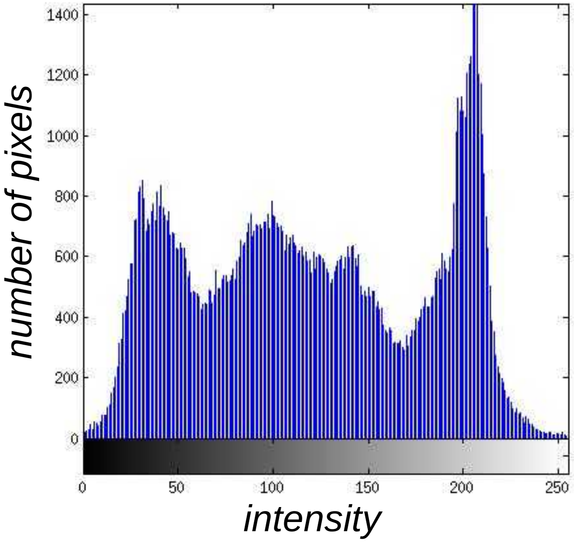
III. Histograms and histogram equalization

Image histogram

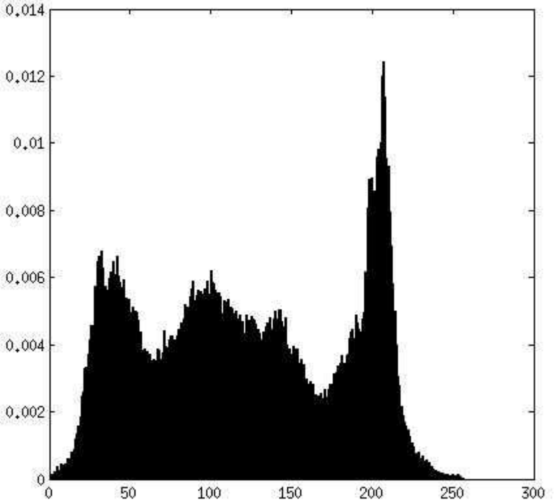
- A gray scale histogram shows how many pixels there are at each intensity level.

- width = 340 px
- height = 370 px
- bit-depth = 8 bits → 0..255

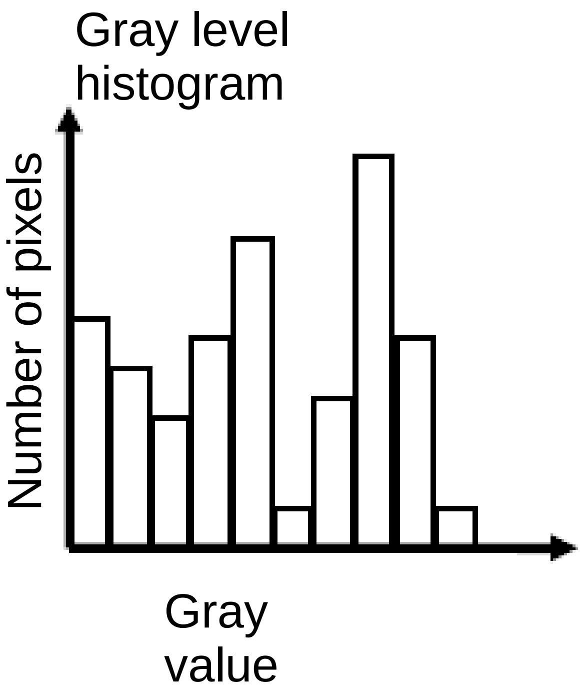
Histogram

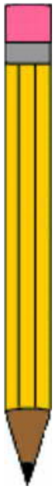


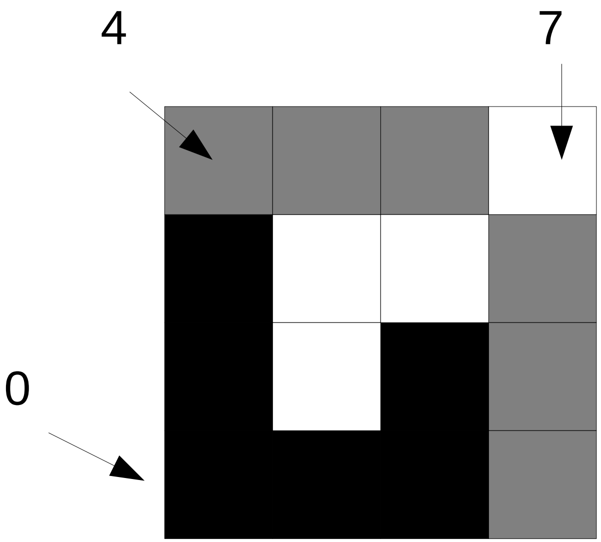
Normalized histogram



Exercise



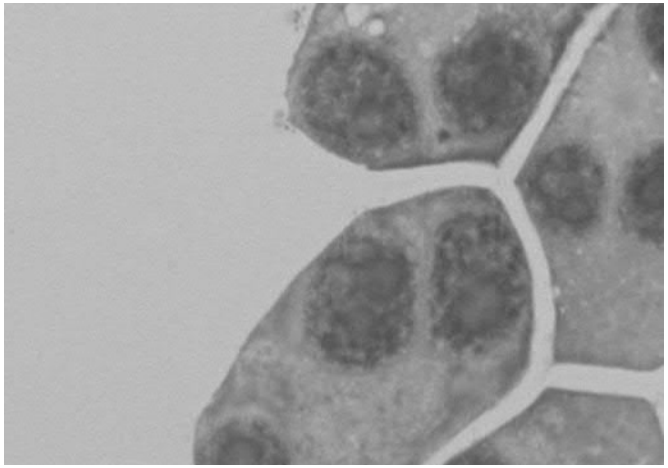


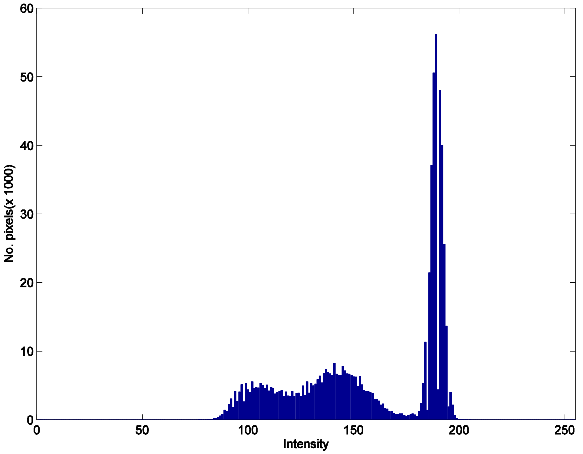


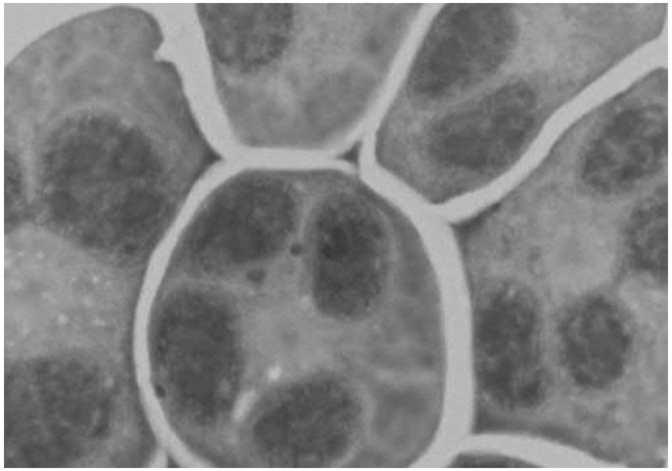
- width = 4px
- height = 4px
- bit-depth = 3 bits

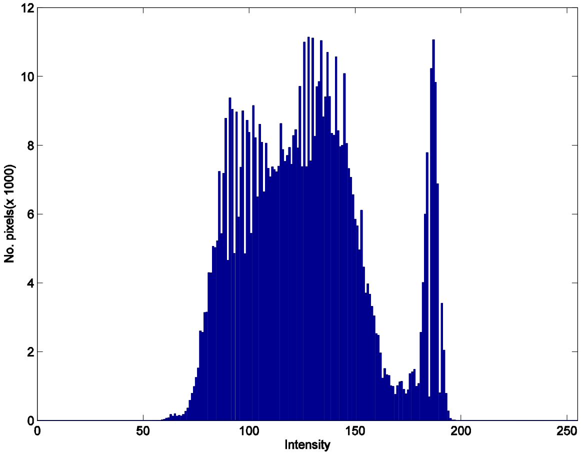
Image histogram

- Gray-level histogram shows intensity distribution



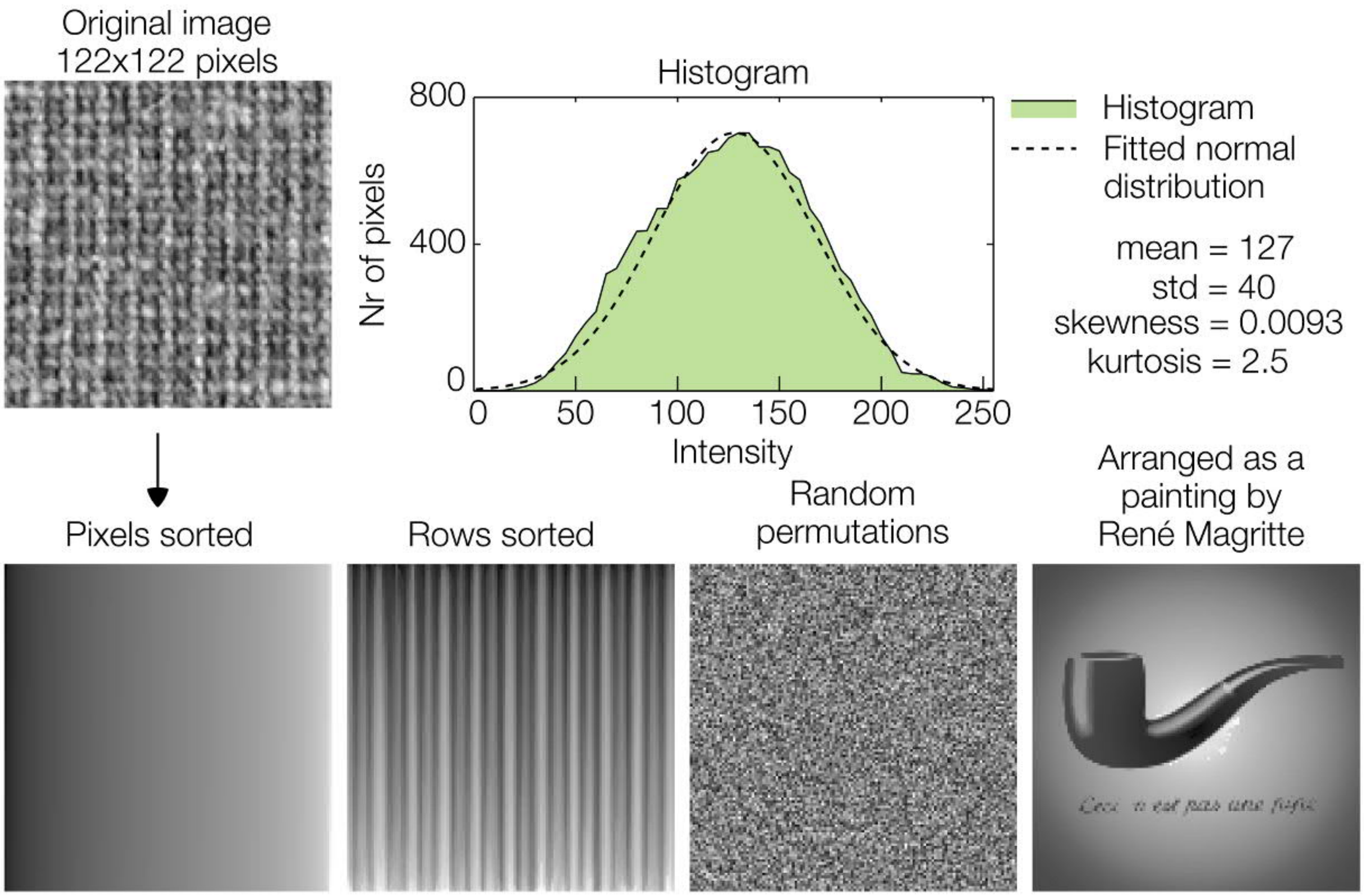






Beware

- Intensity histogram says nothing about the spatial distribution of the pixel intensities

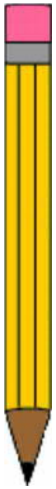


A

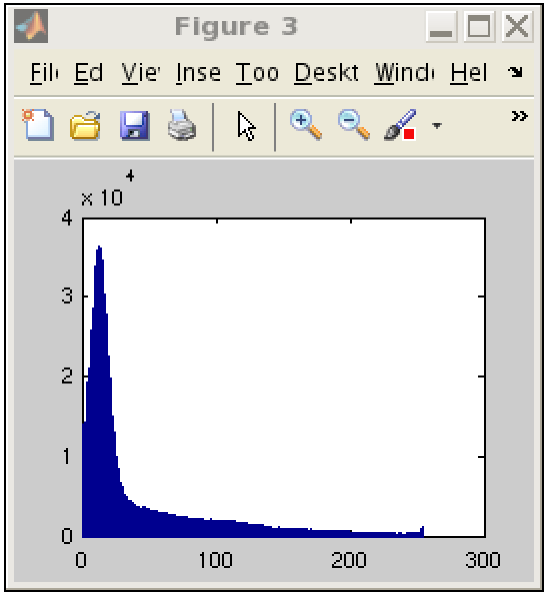
B

C

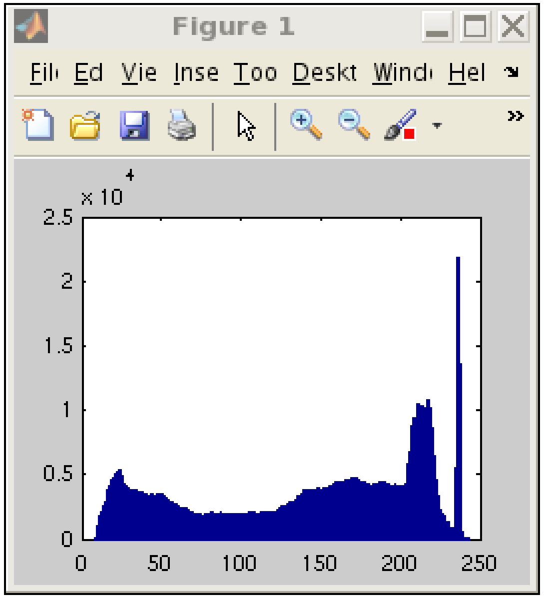
Pair images and histograms!



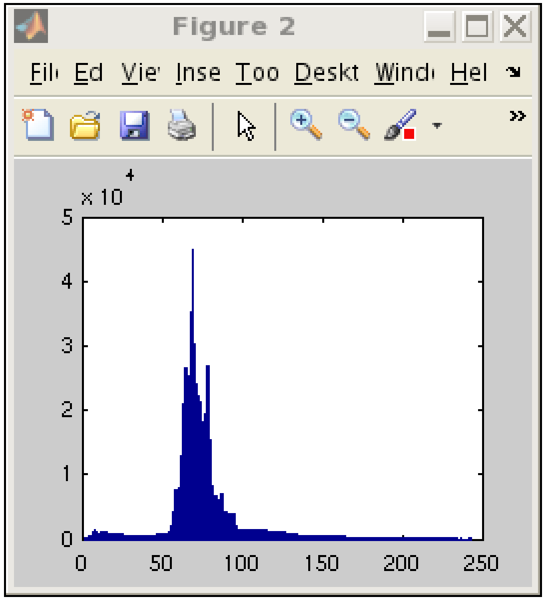
E



F

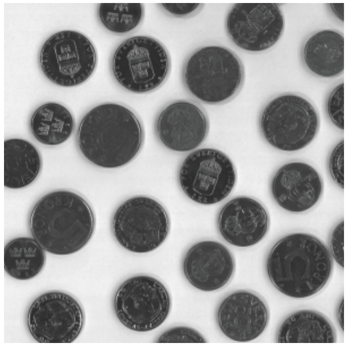
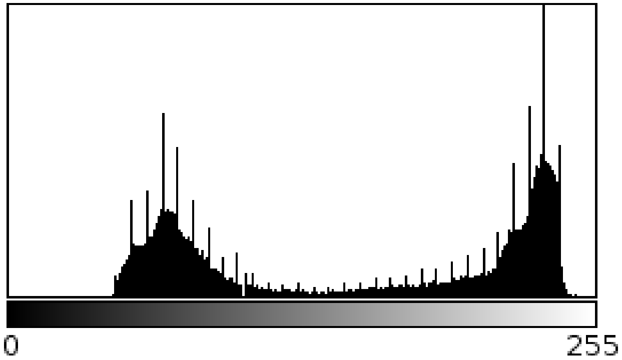


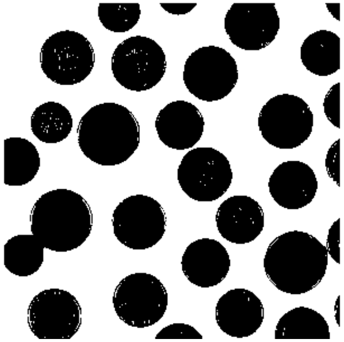
G



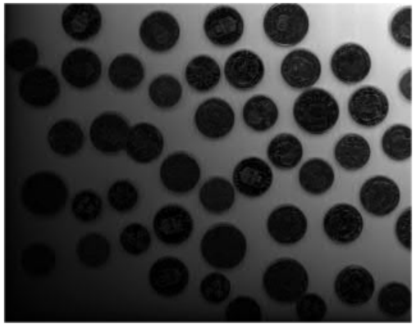
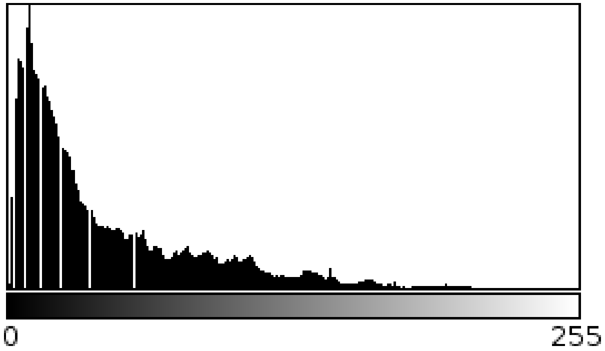
Use of histogram

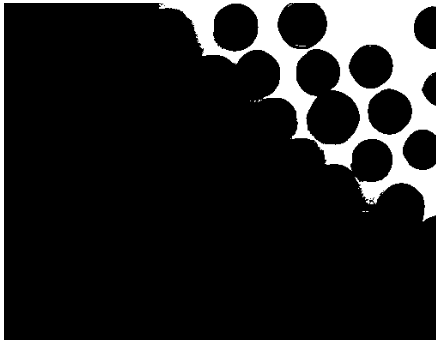
- Thresholding → decide the best threshold value
- ***works well with bi-modal*** **histograms**



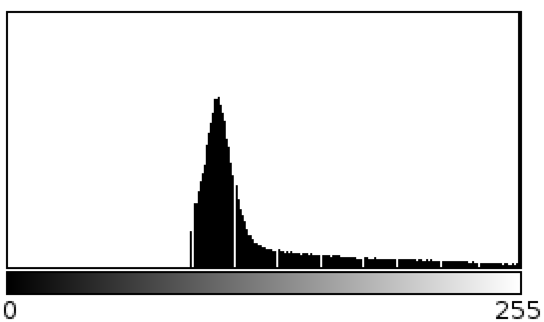
- ***does not work with uni-modal*** **histograms**

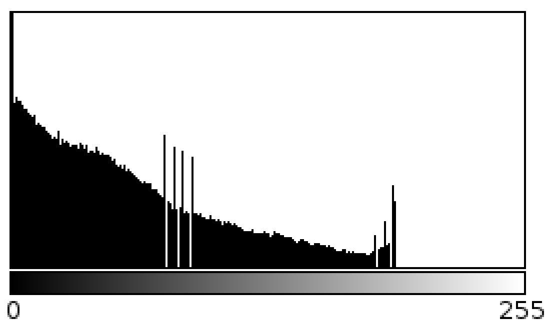





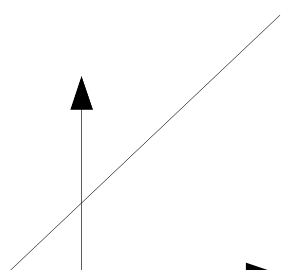
- Analyze the brightness and contrast of an image
- Histogram equalization

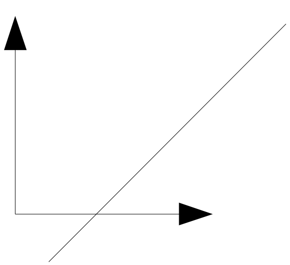
Analyze the brightness





- Increase brightness - shift histogram to the right
- Decrease brightness - shift histogram to the left



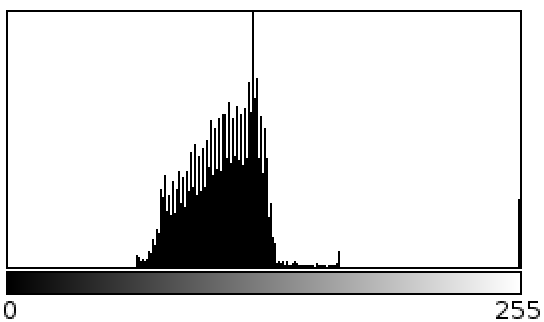


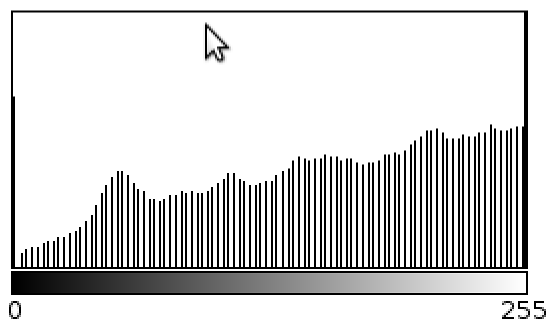
Greylevel transform:

up → increased brightness

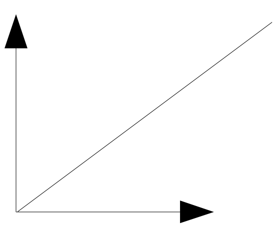
down → decreased brightness

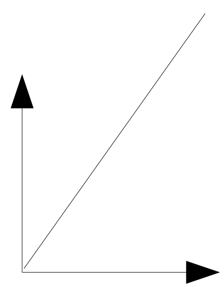
Analyze the contrast





- Decreased contrast - compressed histogram.
- When contrast is increased - the histogram is stretched.



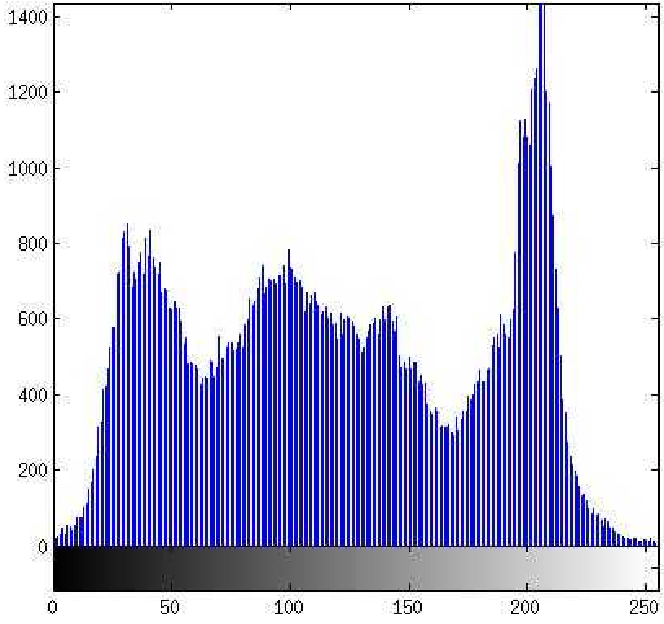


•Greylevel transform:
<45 ° → decreased contrast
>45 ° → increased contrast

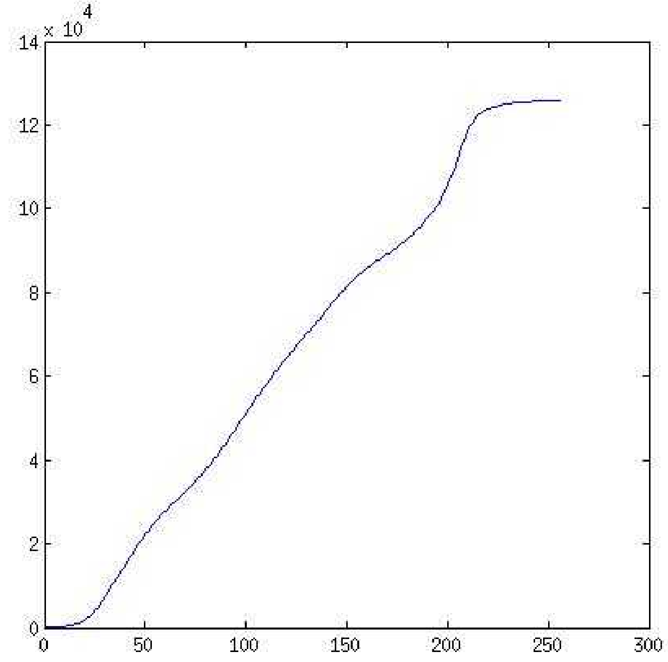
Cumulative histogram

- Easily constructed from the histogram

$$c_j = \sum_{i=0}^{j} h_i$$



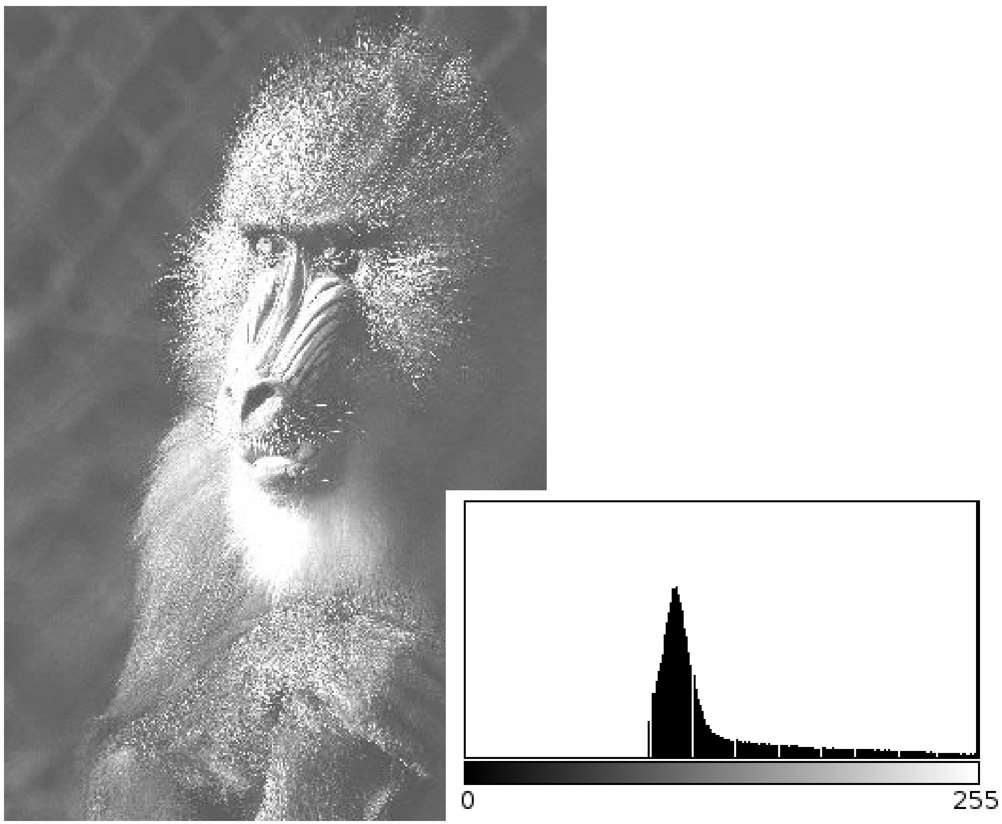
Histogram

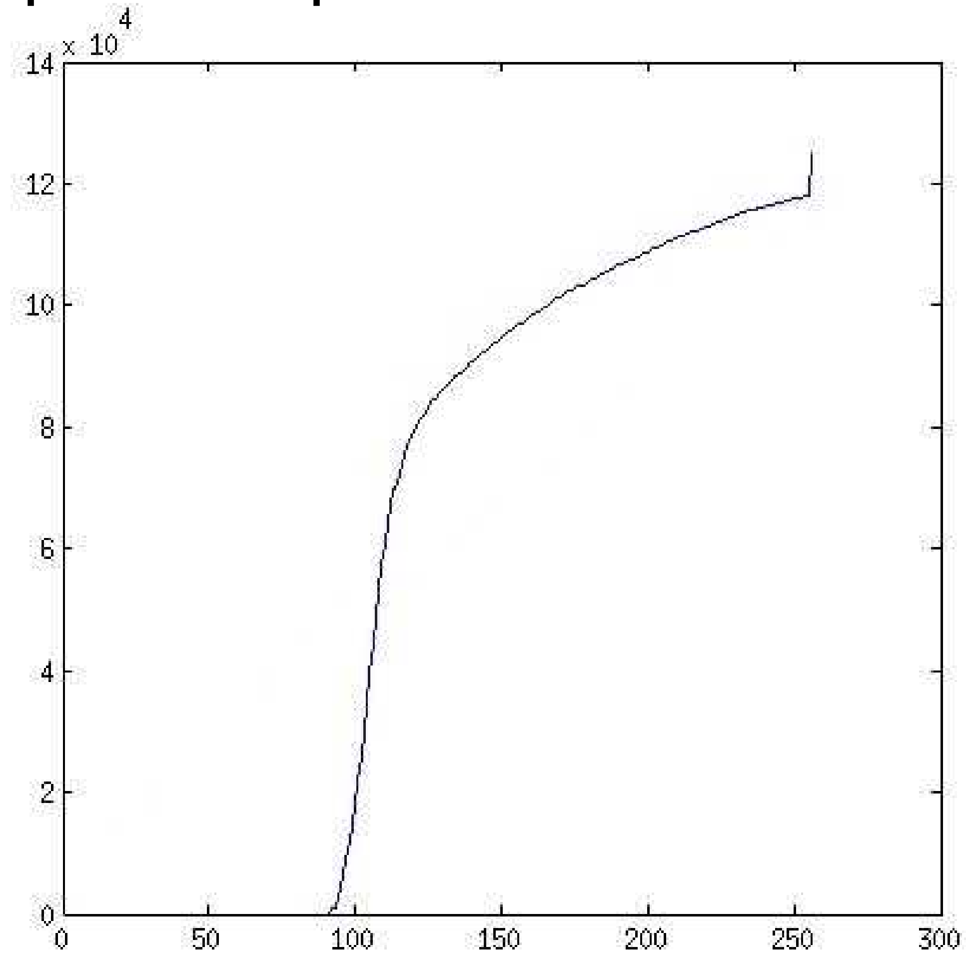


Cumulative histogram

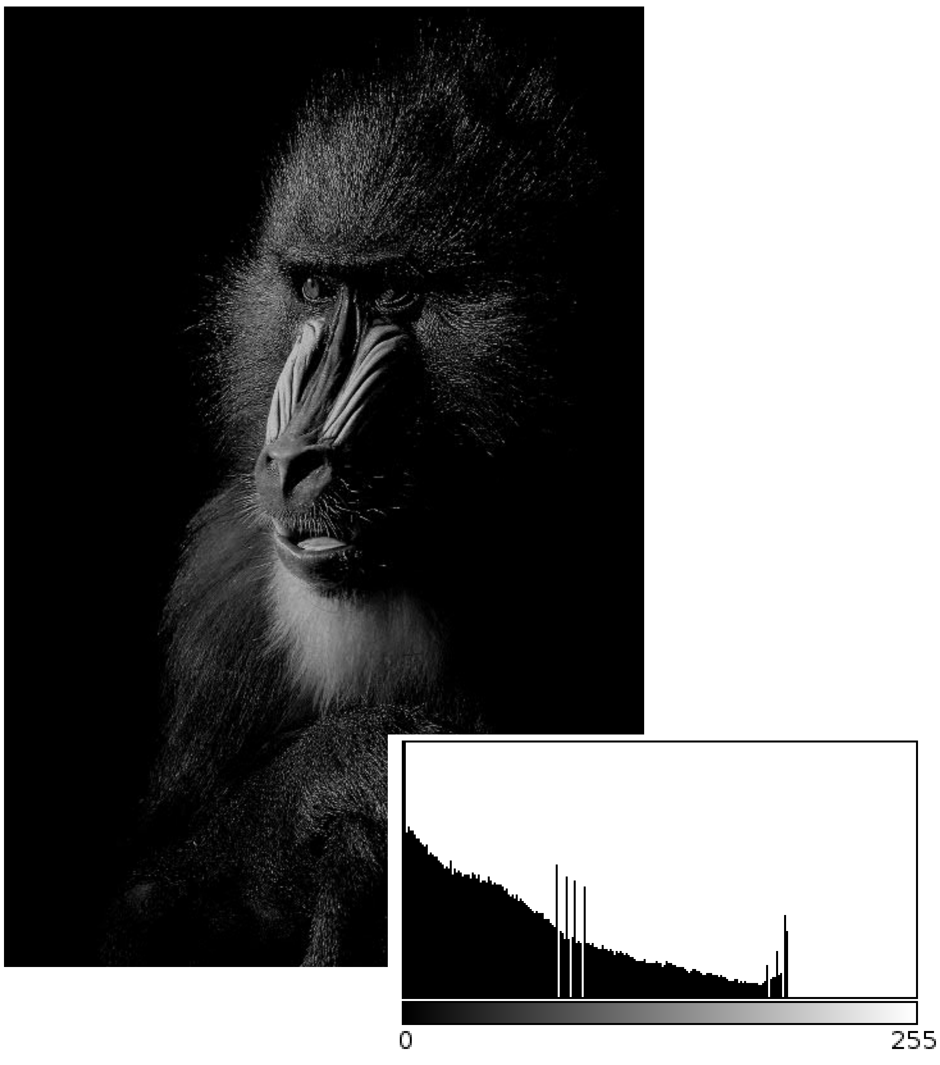
Cumulative histogram

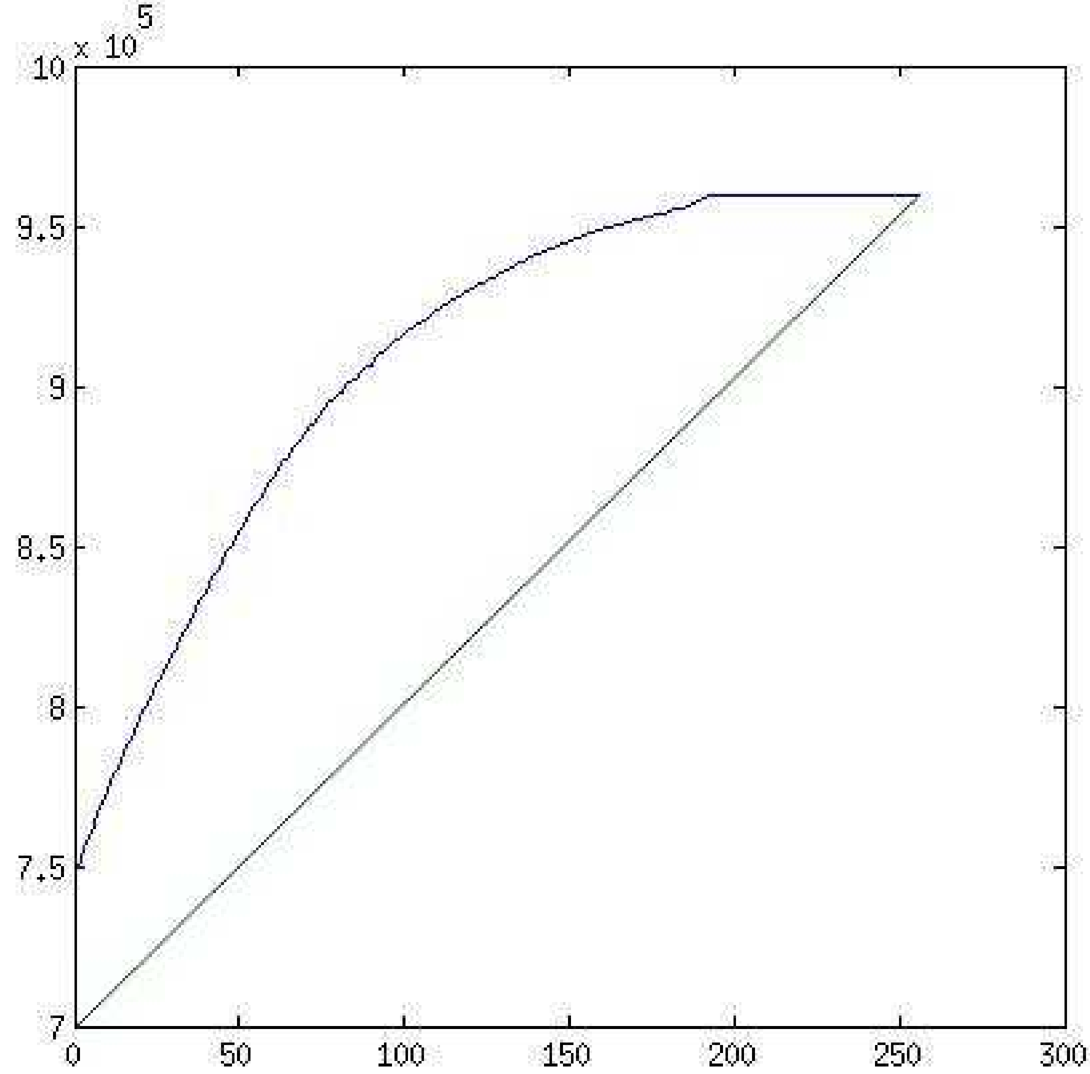
- Slope
  - Steep → intensely populated parts of the histogram
  - Gradual → in sparsely populated parts of the histogram





Cumulative histogram





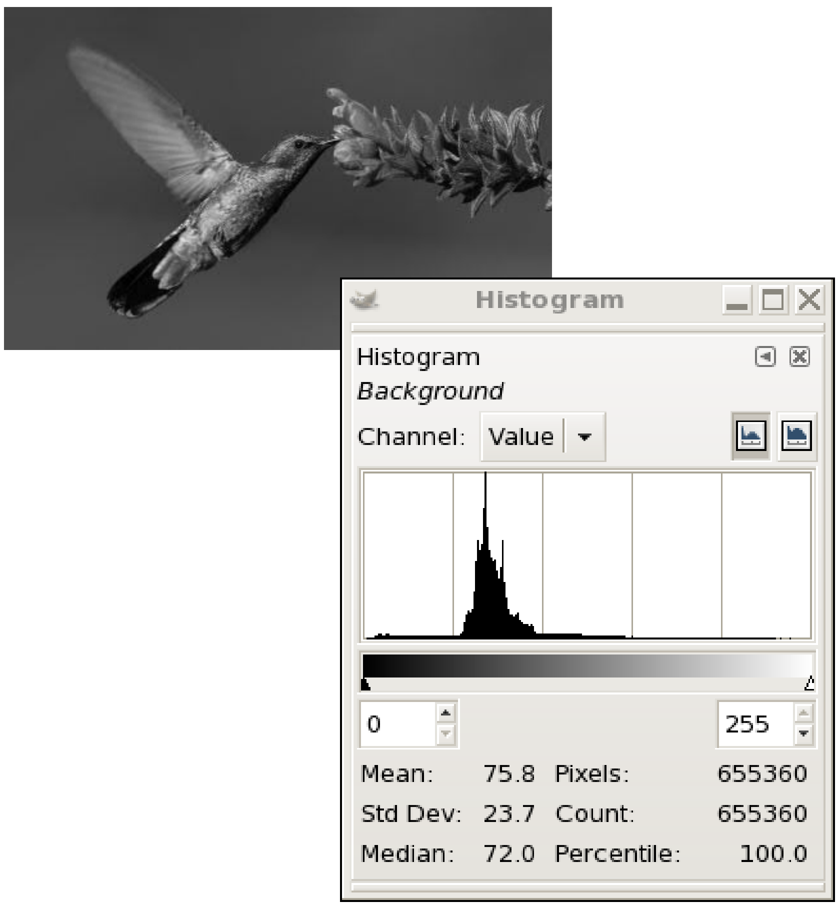
Histogram equalization

- Idea: Create an image with evenly distributed greylevels, for visual contrast enhancement
- Goal: Find the transformation that produces the most even histogram → cumulative histogram curve

- Equalization flattens the histogram or linearize cumulative histogram
- Automatic contrast enhancement

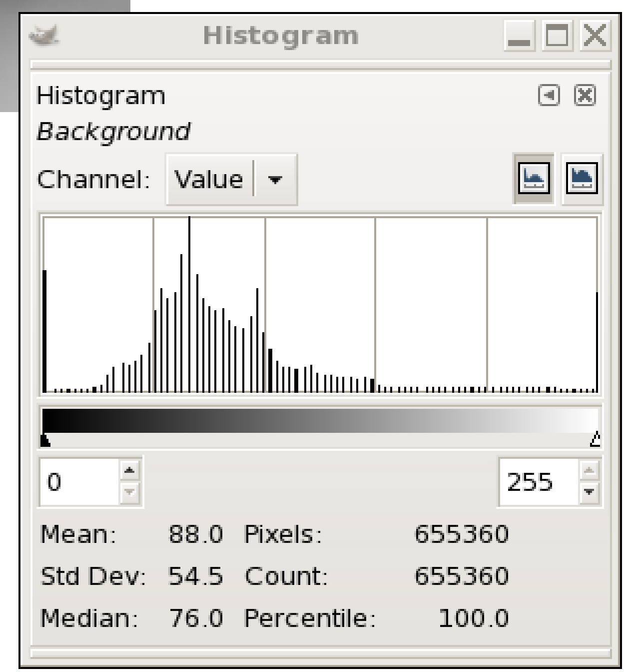
Histogram equalization

original image



result of histogram equalization

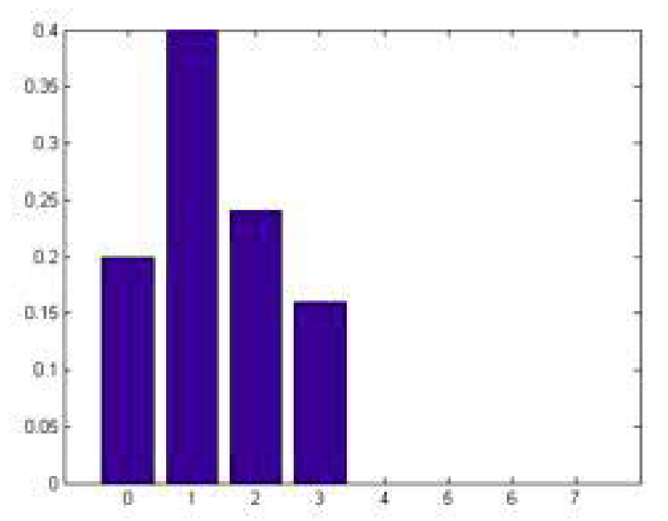
The contrast transform



Hist eq: small example

- Intensity 0 1 2 3 4 5 6 7
- Number of pixels 10 20 12 8 0 0 0 0

- p(0) = 10/50 = 0.2, cdf(0)=0.2
- p(1) = 20/50 = 0.4, cdf(1)=0.6
- p(2) = 12/50 = 0.24, cdf(2)=0.84
- p(3) = 8/50 = 0.16, cdf(3)=1
- p(r) = 0/50 = 0, r = 4, 5, 6, 7 cdf(r)=1

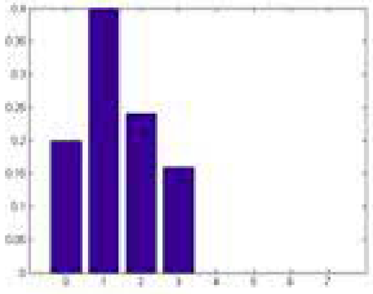


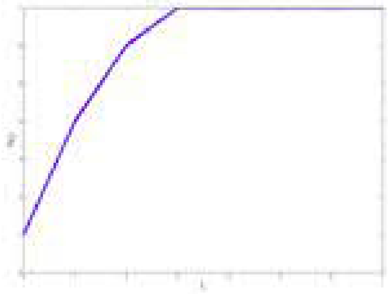
Hist eq: small example

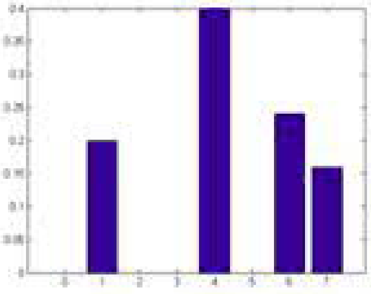
T(x) = MAX * cdf(x)

- T(0) = MAX * cdf(0) = 7 * (p(0)) = 7 * 0.2 = 1.4 ≈ 1
- T(1) = MAX * cdf(1) = 7 * (p(0) + p(1)) = 7 * 0.6 = 3.6 ≈ 4
- T(2) = MAX * cdf(2) = 7 * (p(0) + p(1) + p(2)) ≈ 6
- T(3) = MAX * cdf(3) = 7 * (p(0) + p(1) + p(2) + p(3)) ≈ 7
- T(r) = 7, r = 4, 5, 6, 7

Intensity	0	1	2	3	4	5	6	7
Number of pixels	10	20	12	8	0	0	0	0







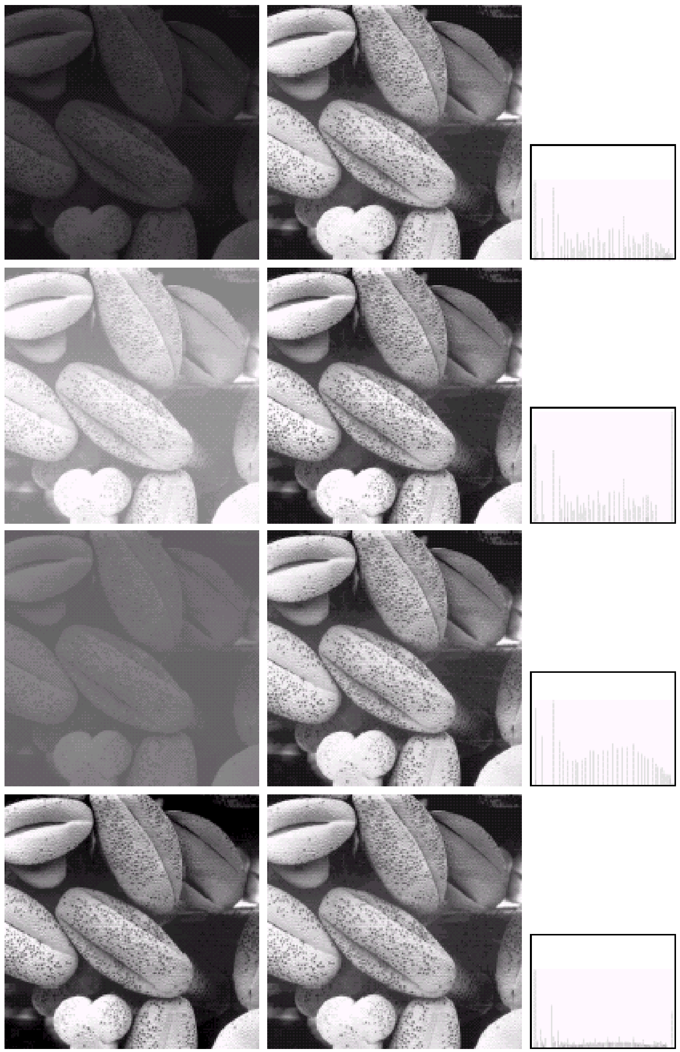
More examples of hist eq

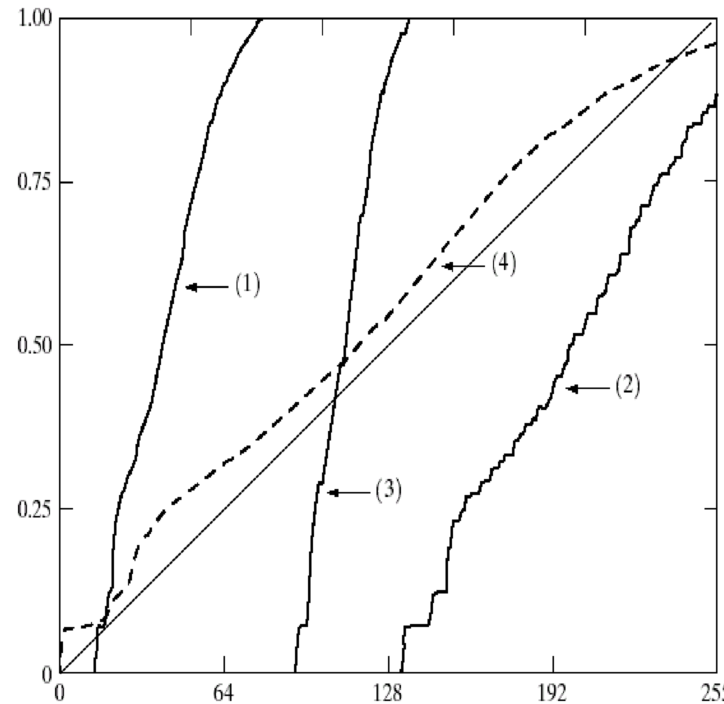
1

2

3

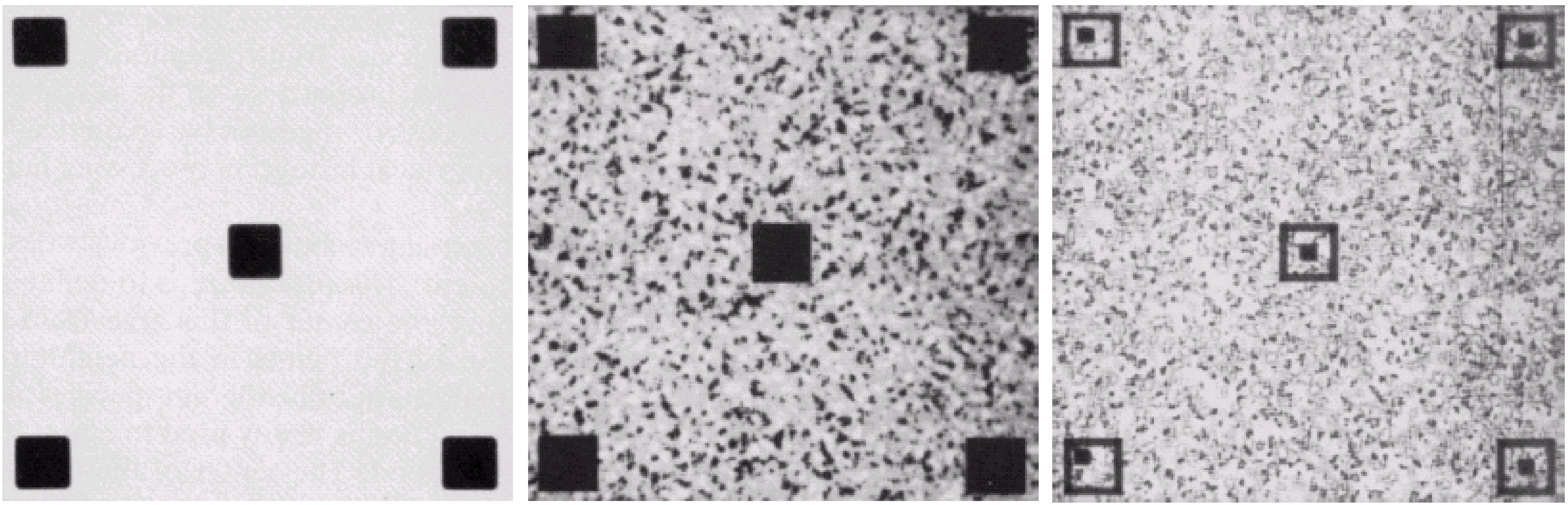
4





Transformations for image 1-4. Note that the transform for figure 4 (dashed) is close to the neutral transform (thin line).

Local histogram equalization

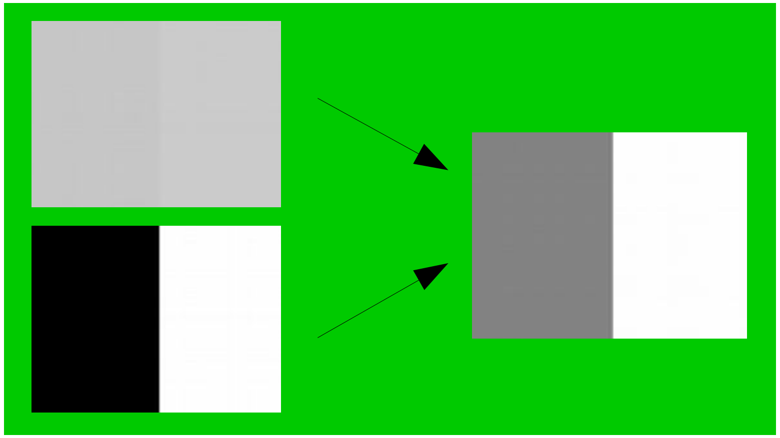


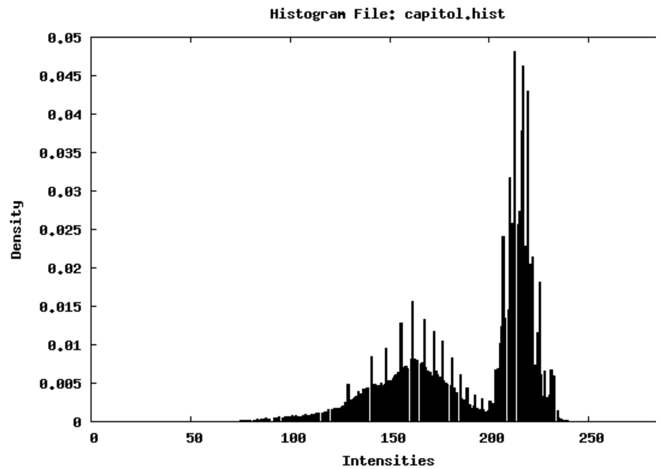
a b c

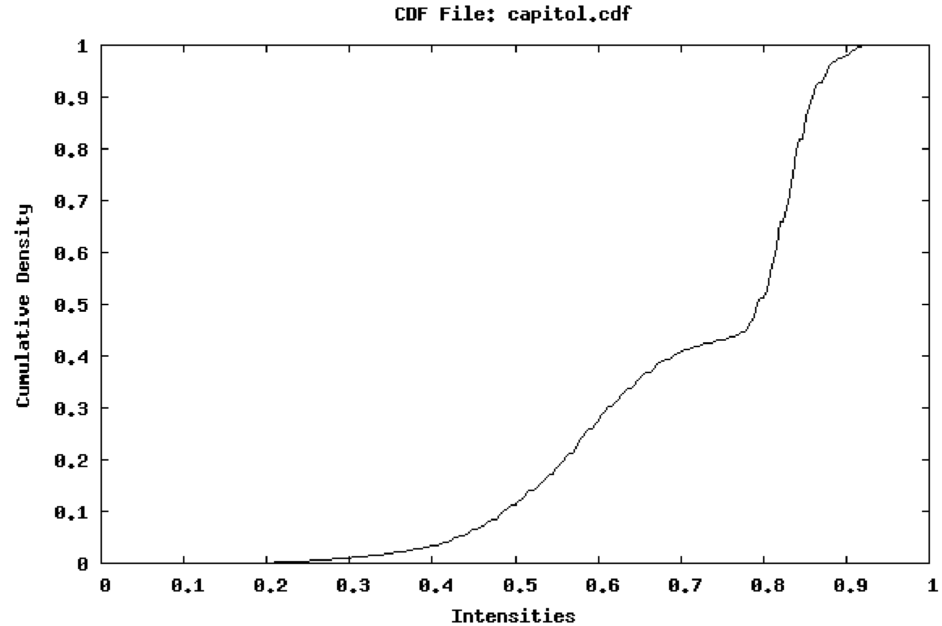
FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.

Histogram equalization

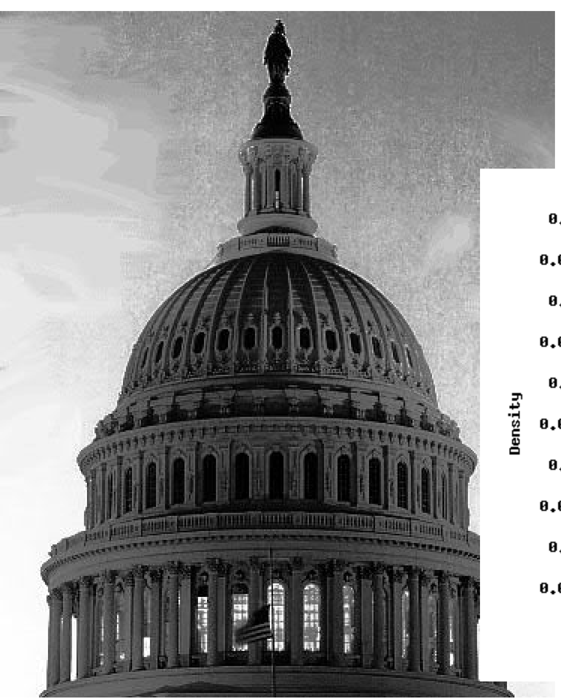
- Useful when much information is in a narrow part of the histogram
- Drawbacks:
  - Amplifies noise in large homogenous areas
  - Can produce unrealistic transformations
  - Information might be lost, no new information is gained
  - Not invertible, usually destructive

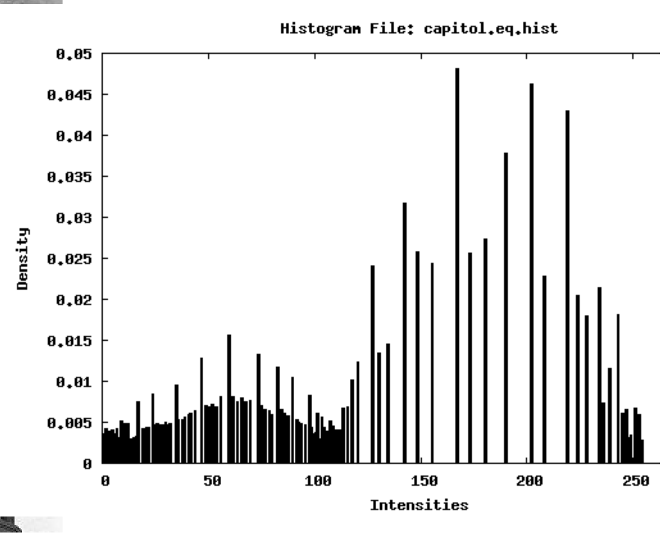


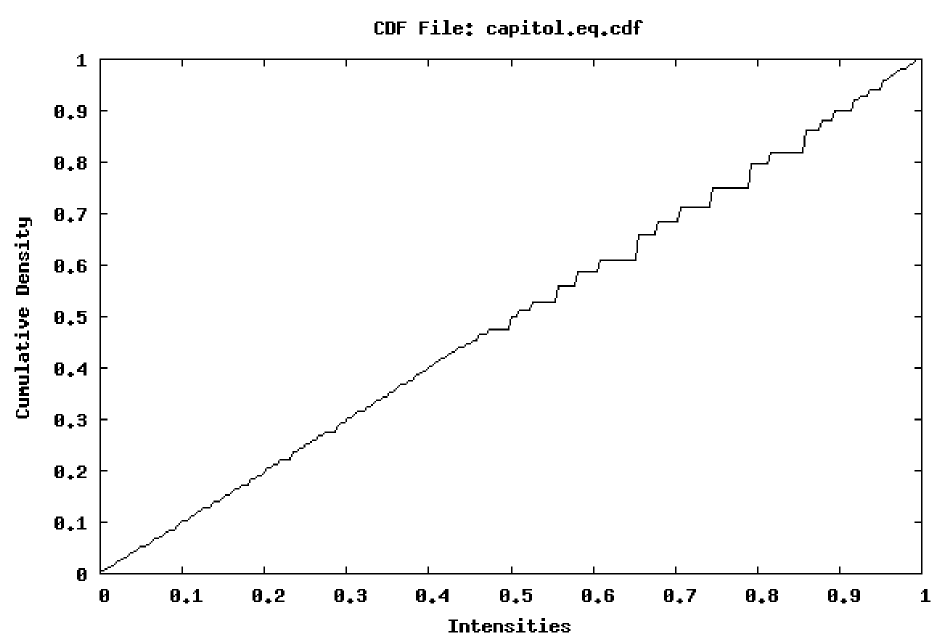




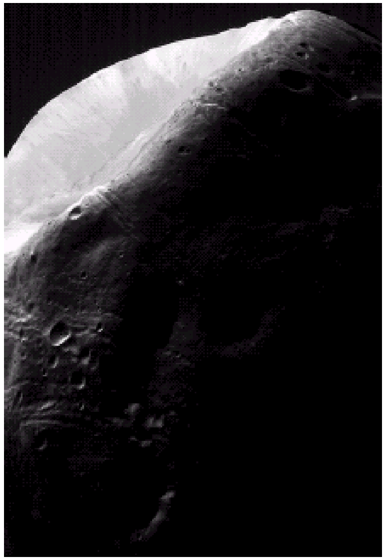
- Does not work well in all cases!







- Histogram equalization is not always “optimal” for visual quality



original image

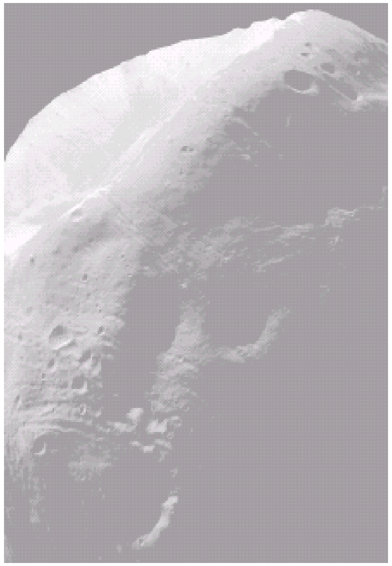
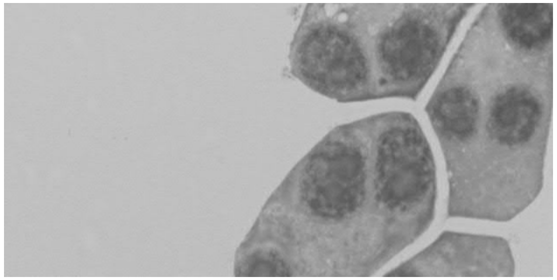
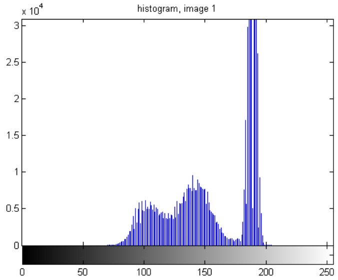


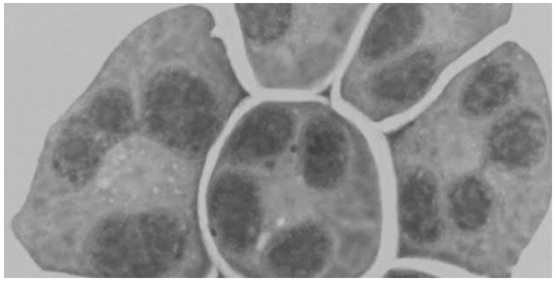
image after histogram equalization

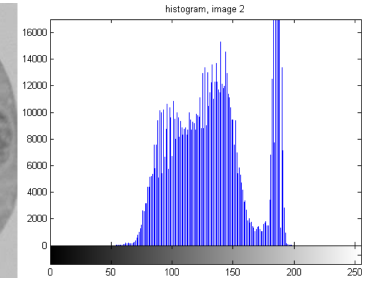
image after manual choice of transform

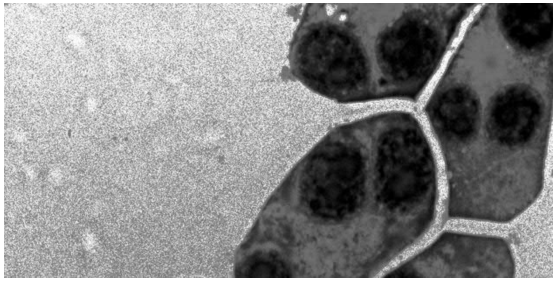
- Histogram eq: the result depends on the amount of different intensities

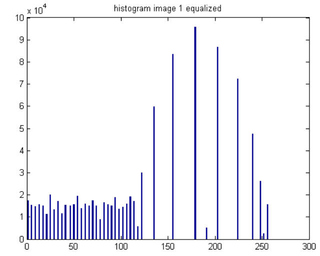


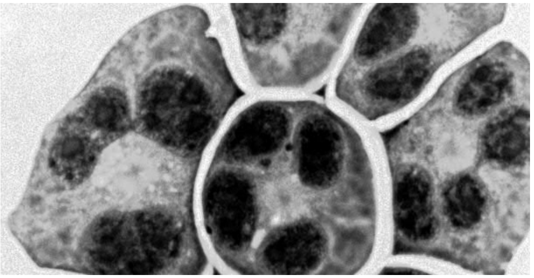


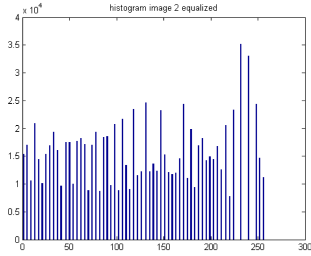










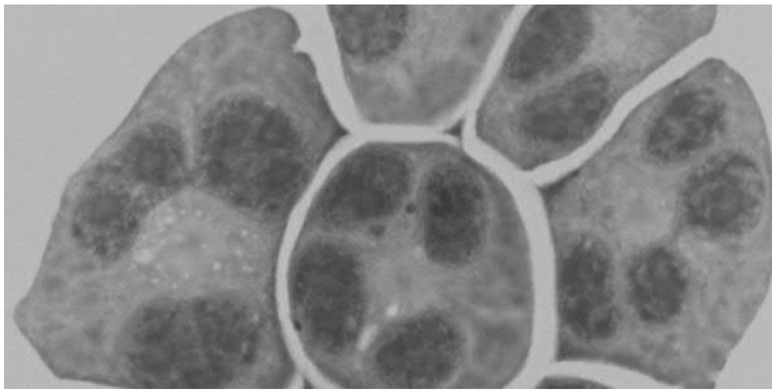


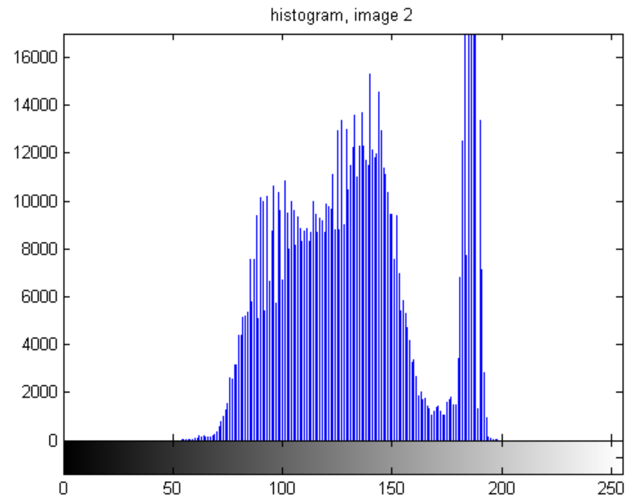
Histogram matching

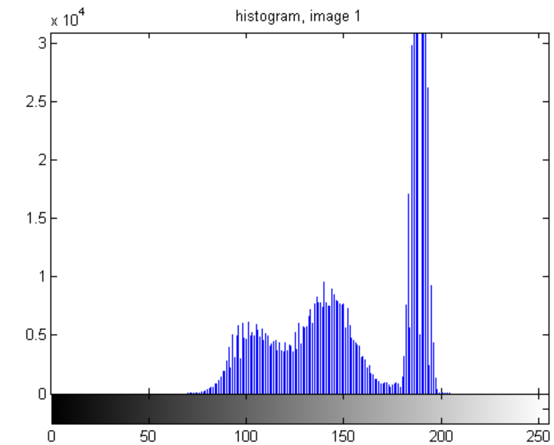
- In histogram equalization a **flat distribution** is the goal
- In histogram matching the **distribution of another image** is the goal

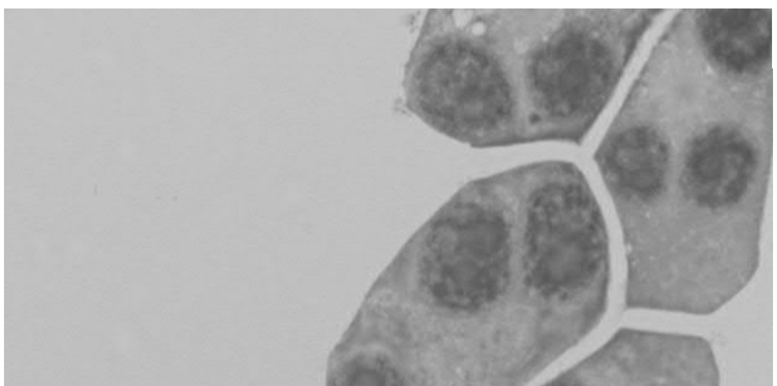
For an image, I, find the transformation, T, that gives the histogram some ideal shape, s.

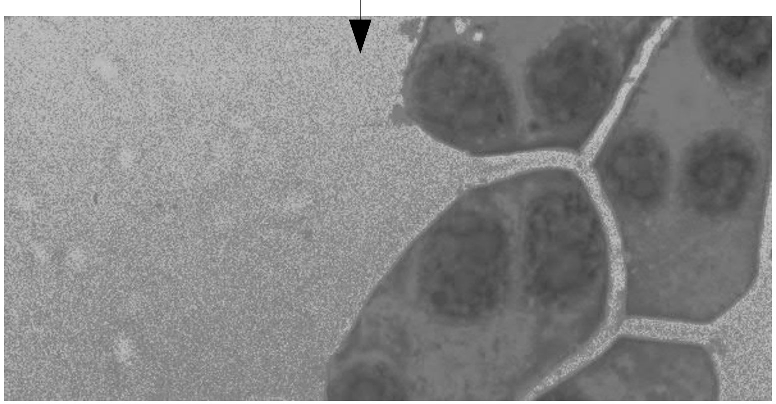
Image 1 histogram matched to image 2











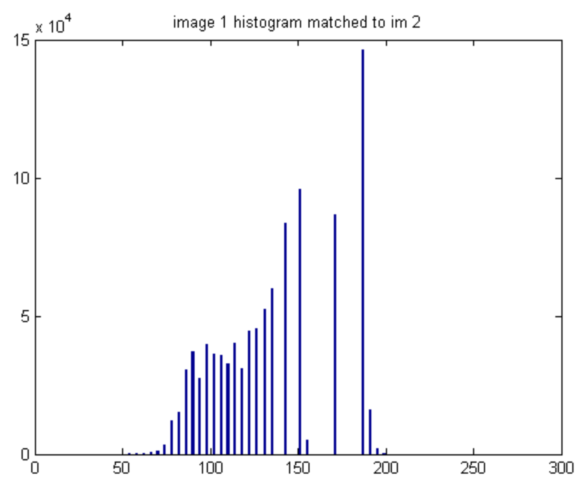
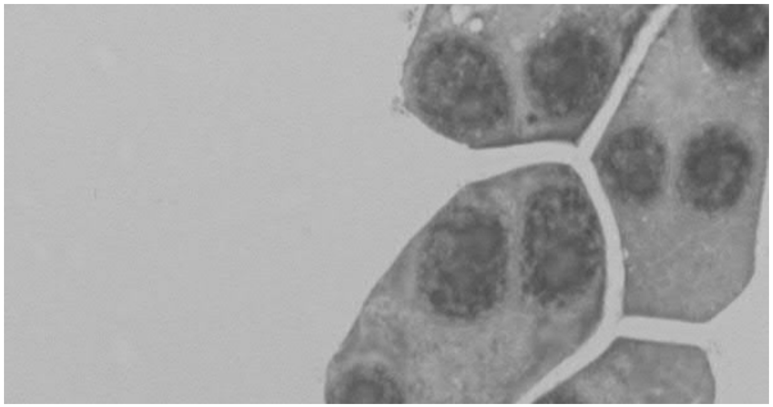
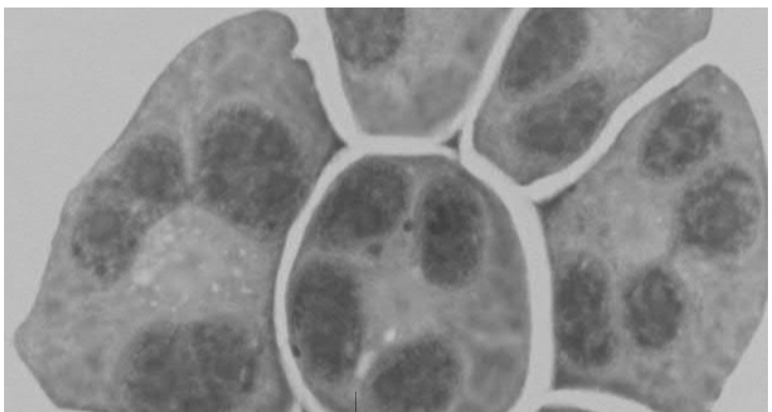
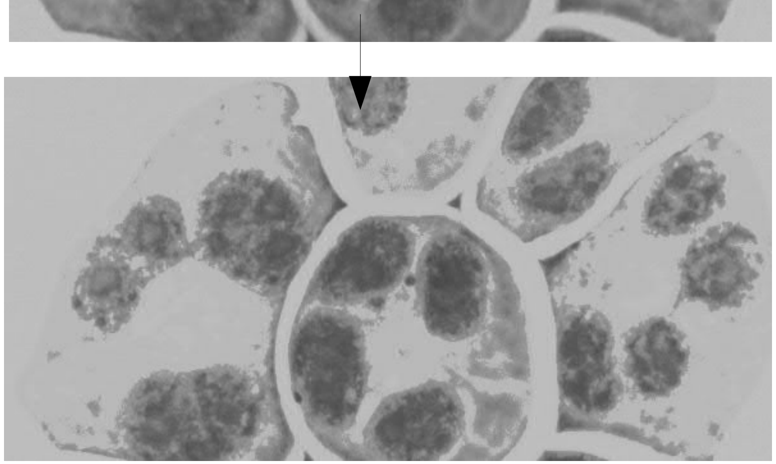
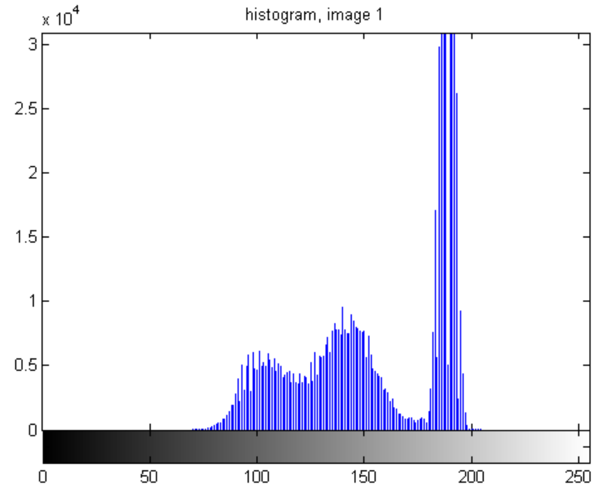


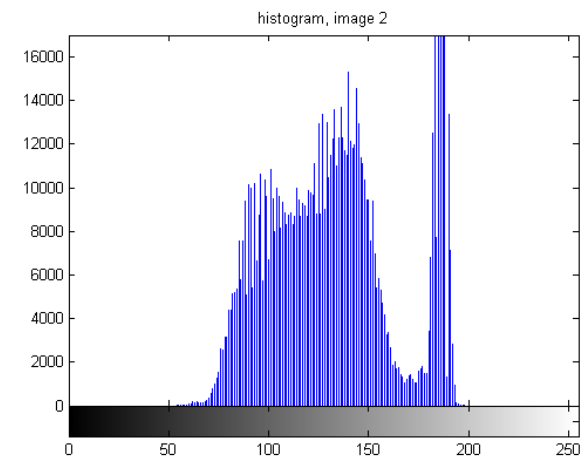
Image 2 histogram matched to image 1

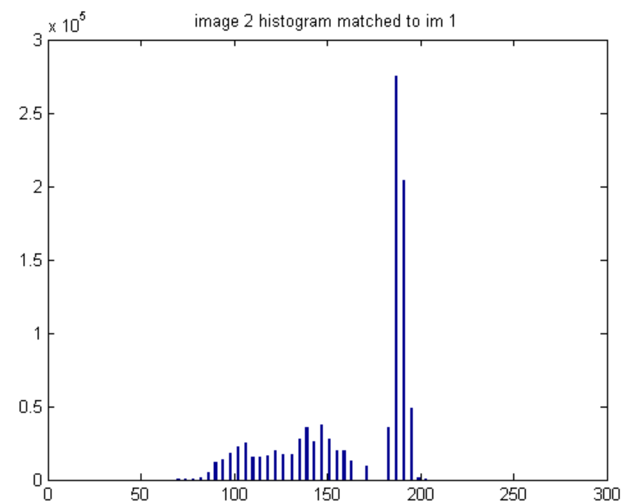






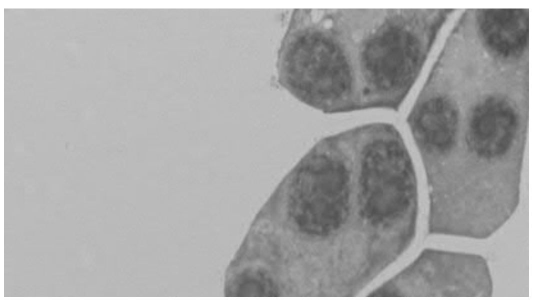


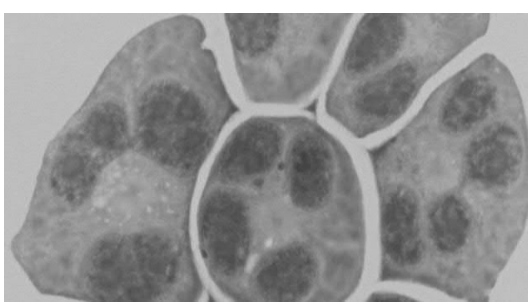


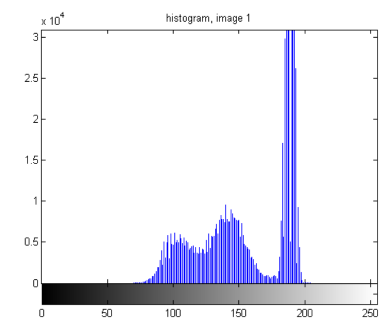


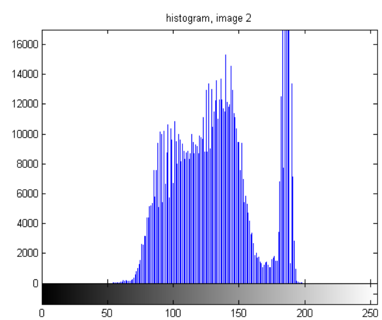
Histogram matching

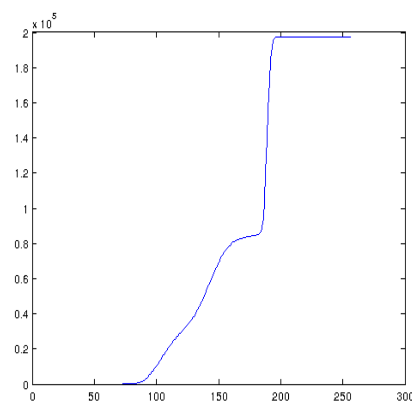
- Compute the histograms for image I_1 , I_2
- Calculate the cumulative distribution function $F_1()$,$F_2()$
- For each gray level G_1 [0,255] find gray level G_2 for which $F_1(G_1)=F_2(G_2)$
  - Histogram matching function $M(G_1) = G_2$



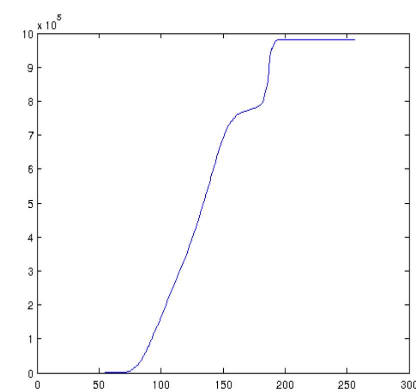








img1



img2

Try at home!

Summary

- Many common tasks can be described by image arithmetics.
- Histogram equalizations can be useful for visualization.
- Watch out for information leaks!

A few things to think about....

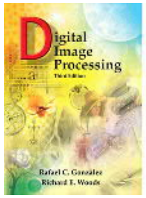
- What is the relation between image arithmetics and linear transfer functions?
- What can you tell about an image from its histogram?
- If you have an 8-bit image, A; how will the 8-bit image B=255*(A+1) look like (exactly!)?
- What conclusions can you draw from the histogram if the first/last column is really high?
- Can you get better resolution by combining multiple images of the same sample?

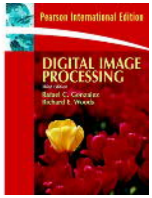
Suggested problems:

2.22, 2.18, 2.9, 3.1, 3.6

Next lecture:

Spatial filtering (Ch. 3.4-3.8)











84-342