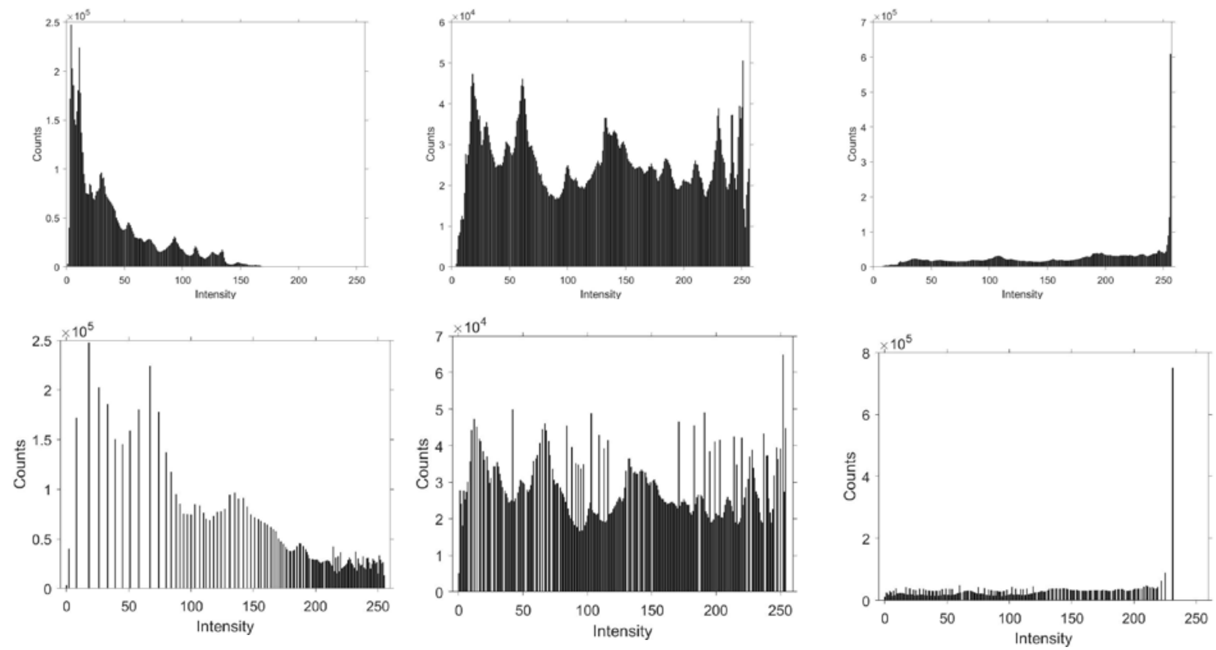
Image Processing II

Histograms &
Intensity transformations



Histograms

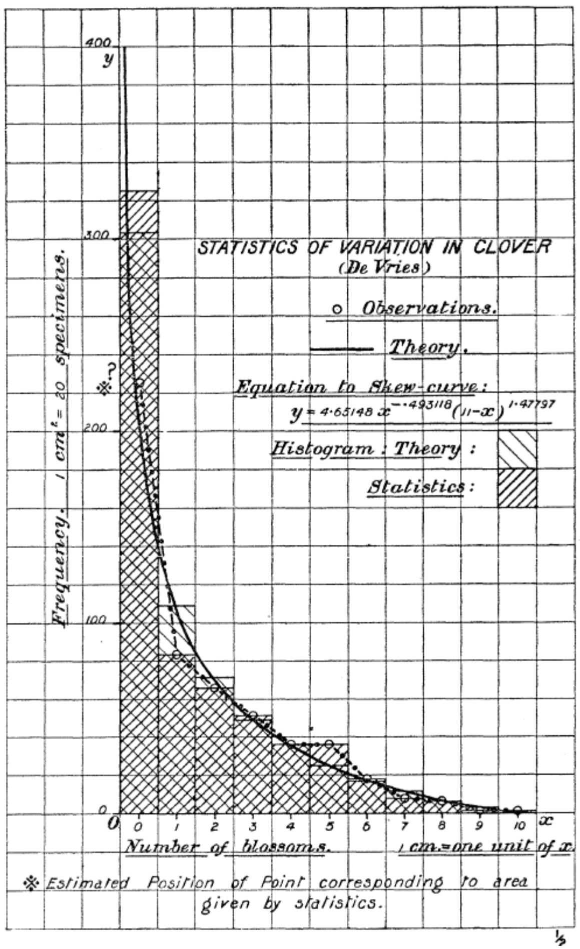
Learning objectives

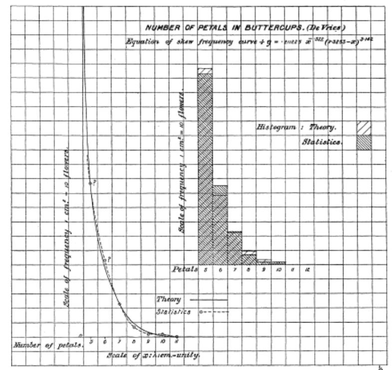
This lecture will cover the following topics:

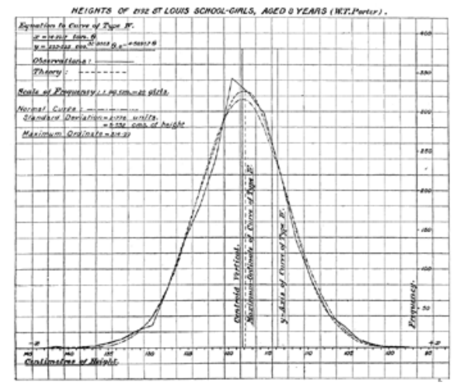
- Generation of image histograms
- Optimization of image acquisition
- Intensity transformations used for image enhancement
  - Brightness and contrast adjustment
  - Histogram equalization
  - Histogram matching

Histograms

Generation of histograms







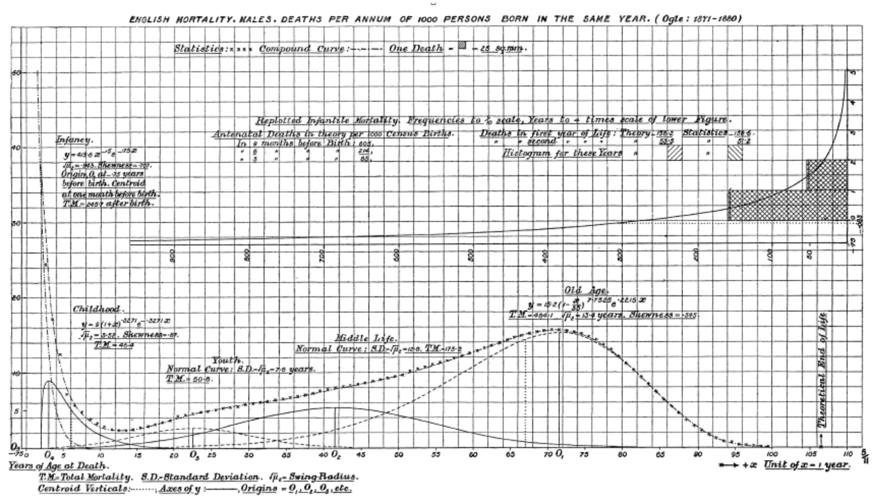
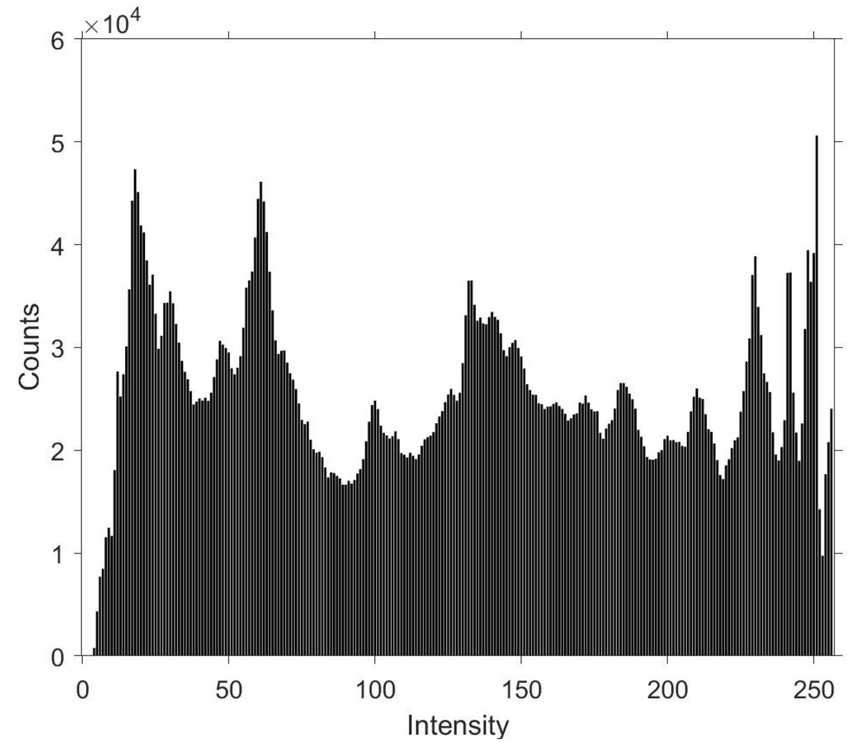


Fig. from: Pearson, Phil. Trans. R. Soc. Lond. A 1895 186, 343-414 (1895)

Histograms

Generation of image histograms



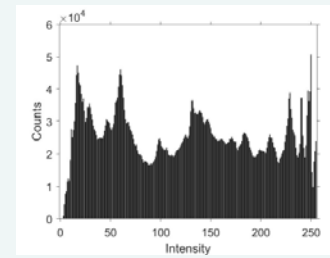
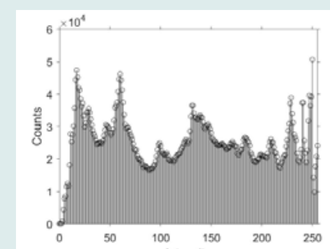
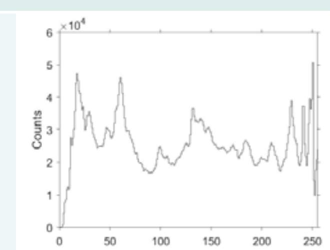
Histograms

Generation of image histograms in MATLAB

Syntax	Description
MATLAB functions for generating histograms	
`[counts,edges] = histcounts(X,edges)`	Sorts data into bins with the bin edges specified by the vector `edges`. Each bin includes the left edge, but does not include the right edge, except for the last bin which includes both edges.
`counts = histc(X,binRange)`	Counts the number of values in `X` that are within each specified bin range. The input `binRange` determines the endpoints for each bin. The output `counts` contains the number of elements from `X` in each bin. Note: Since `histc` has limited overall capabilities, its use in new code is discouraged.
Image processing toolbox functions	
`[counts,binLocations]= imhist(Img)`	Calculates the histogram for the grayscale image `Img`. The `imhist` function returns the histogram counts in `counts` and the bin locations in `binLocations`. The number of bins in the histogram is determined by the image class.
	See MATLAB help page for further information

Histograms

Plotting histograms in MATLAB

Syntax	Description	Example
MATLAB functions for plotting bar graphs		
`h = bar(X,Y)`	Draws bars with the height Y at the locations specified by X. `h.FaceColor = “black”` or `set(h,“FaceColor”,“black”)` choses black as color for the bars.	
`h = stem(X,Y)`	Plots the data sequence Y as stems that extend from the baseline. The data values are indicated by circles terminating each stem. Other symbols can be chosen by setting the property Marker. The color is determined by the property “Color”. Also style of the stems and of the baseline can be changed.	
`h = stairs(X,Y)`	Draws a stairstep graph of the elements in Y at the locations specified by X.	
	See MATLAB help page for further information	

Histograms

Generating and plotting image histograms in MATLAB

Syntax	Description
MATLAB functions for generating and plotting histograms	
`h = histogram(X,edges)`	Sorts X into bins with the bin edges specified by the vector `edges` and creates a histogram plot. Each bin includes the left edge, but does not include the right edge, except for the last bin which includes both edges.
`hist(X,binRange)`	Counts the number of values in X that are within each specified bin range. The input `binRange` determines the endpoints for each bin. Note: Using `hist` in new code is not recommended.
Image processing toolbox functions	
`imhist(Img)`	`imhist` without output parameters calculates and plots the histogram for the grayscale image. If the input image is an indexed image, then the histogram shows the distribution of the pixel values above a colorbar of the colormap.
	See MATLAB help page for further information

Optimizing image acquisition

Perceived image quality of photographs

t = 20 ms

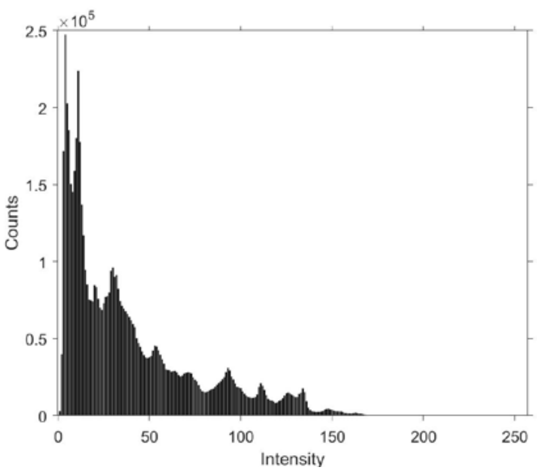
t = 125 ms

t = 250 ms

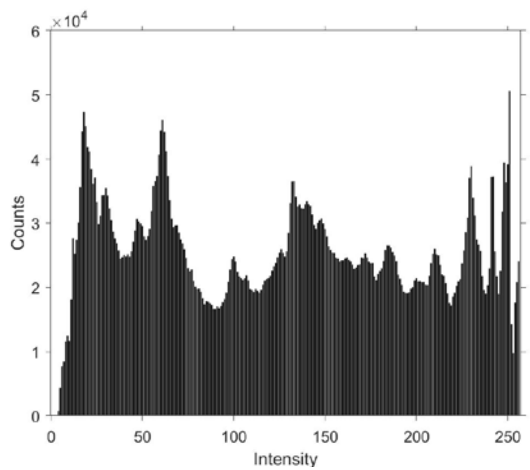
Optimizing image acquisition

Using histograms to assess the image quality

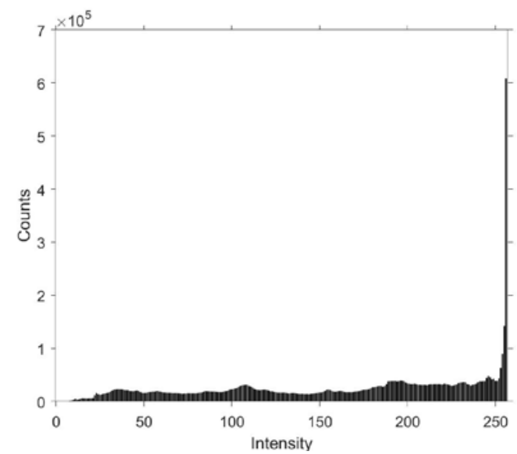
t = 20 ms



t = 125 ms



t = 250 ms



Optimizing image acquisition

Factors affecting the brightness of photographs

Factors affecting the brightness of photographs:

- Exposure time
- ISO value
- Aperture size
- Illumination intensity

Optimizing image acquisition

Factors affecting the brightness of a photograph: The ISO value

Source: Wikipedia, https://commons.wikimedia.org/w/index.php?curid=37351557

Optimizing image acquisition

Factors affecting the brightness of a photograph:
The aperture size

Source: Wikipedia, https://commons.wikimedia.org/w/index.php?curid=78136658

Optimizing image acquisition

Effect of the aperture: F = 1/4

Optimizing image acquisition

Effect of the aperture: F = 1/5.6

Optimizing image acquisition

Effect of the aperture: F = 1/8

Optimizing image acquisition

Effect of the aperture: F = 1/16

Optimizing image acquisition

Effect of the aperture: F = 1/32

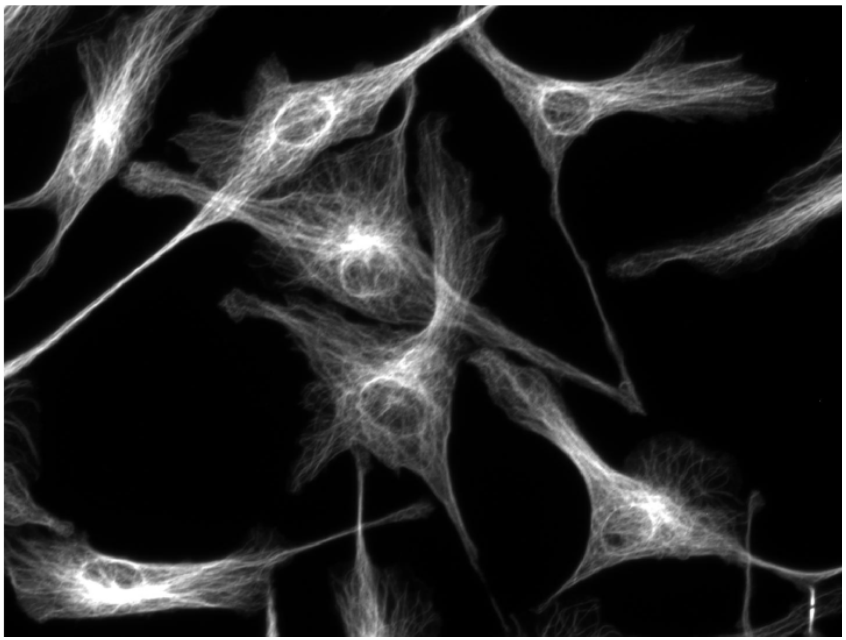
Optimizing image acquisition

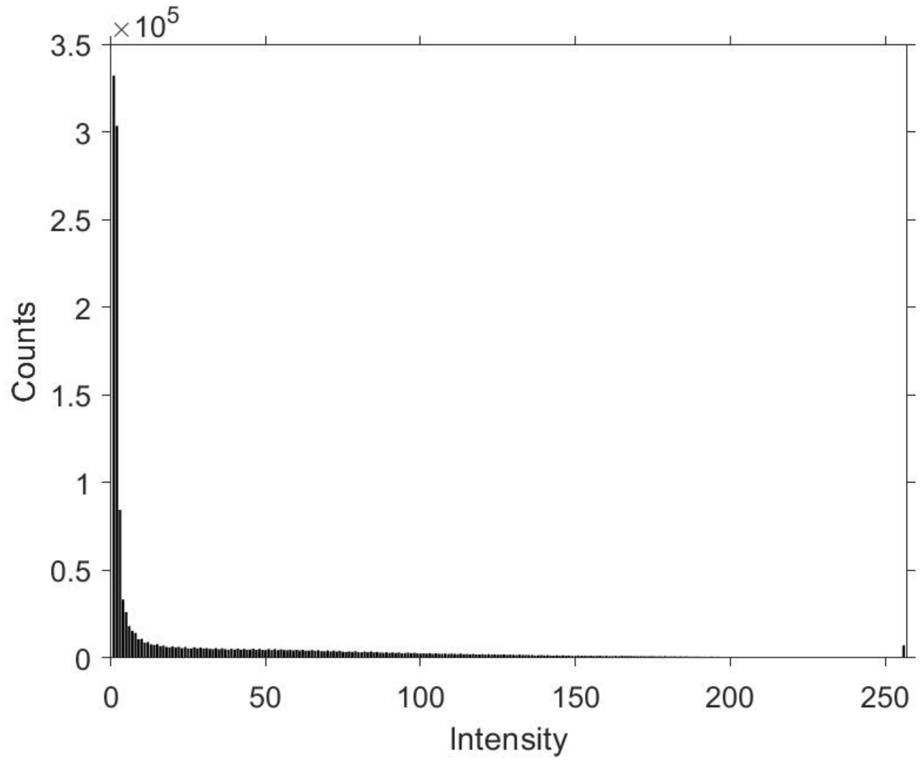
Factors affecting the brightness of photographs

Factors affecting the brightness of photographs:

- Exposure time
- ISO value
- Aperture size
- Illumination intensity

Optimizing image acquisition
Histograms of microscopy images





Optimizing image acquisition

Factors affecting the brightness of microscopy images

Factors affecting the intensity of an epifluorescence micrograph:

- Illumination intensity
- Acquisition time
- Gain of the camera
- Digital gain

Factors affecting the intensity of a confocal laser scanning micrograph

- Illumination intensity, i.e. laser intensity
- Pixel dwell time, i.e. scan speed
- Photomultiplier gain
- Digital gain

Intensity transformations
Definition

General form of intensity transformation functions:

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

where $f(x,y)$ is the input image,
$g(x,y)$ is the output image,
T is an operator on f defined over a specified neighborhood of point (x,y)

Special form of intensity transformation for histogram operations:

$$s = T(r)$$

where r denotes the intensity in point (x,y) of the input image,
s denotes the intensity in the same point (x,y) of the output image,
T is an operator on f acting only on point (x,y)

Intensity transformations

Definition

Intensity transformation for histogram operations:

$$s = T(r)$$

where

r denotes the intensity in point (x,y) of the input image,

s denotes the intensity in the same point (x,y) of the output image,

T is an operator on f acting only on point (x, y)

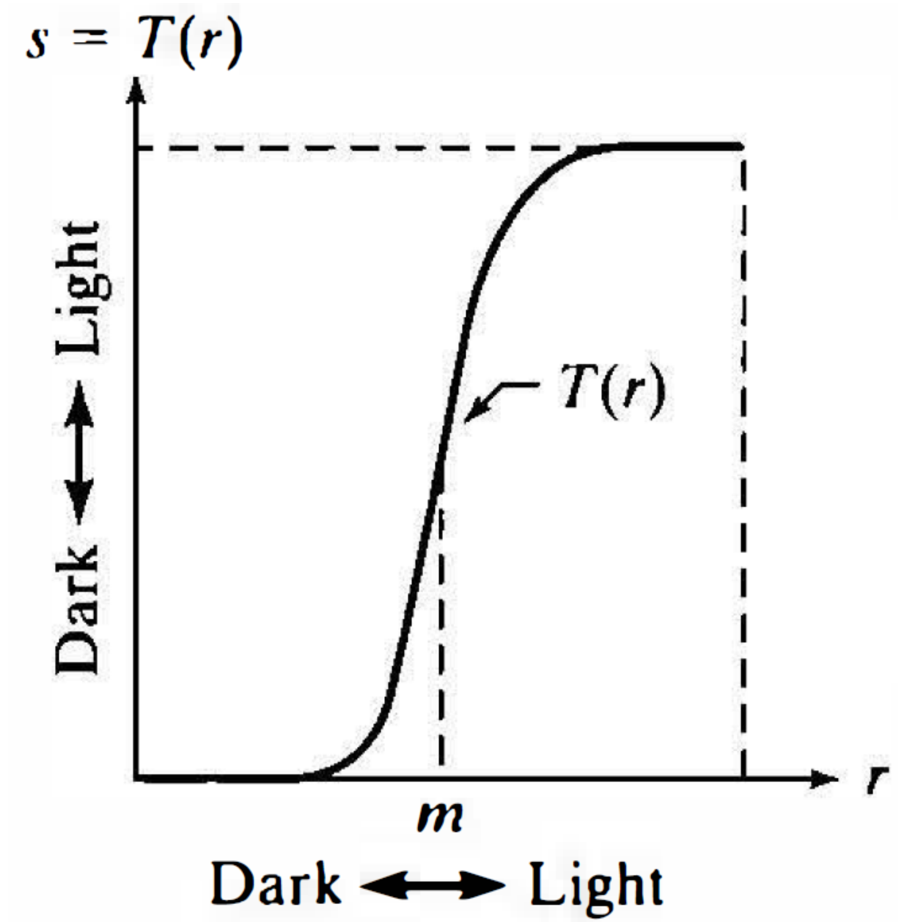
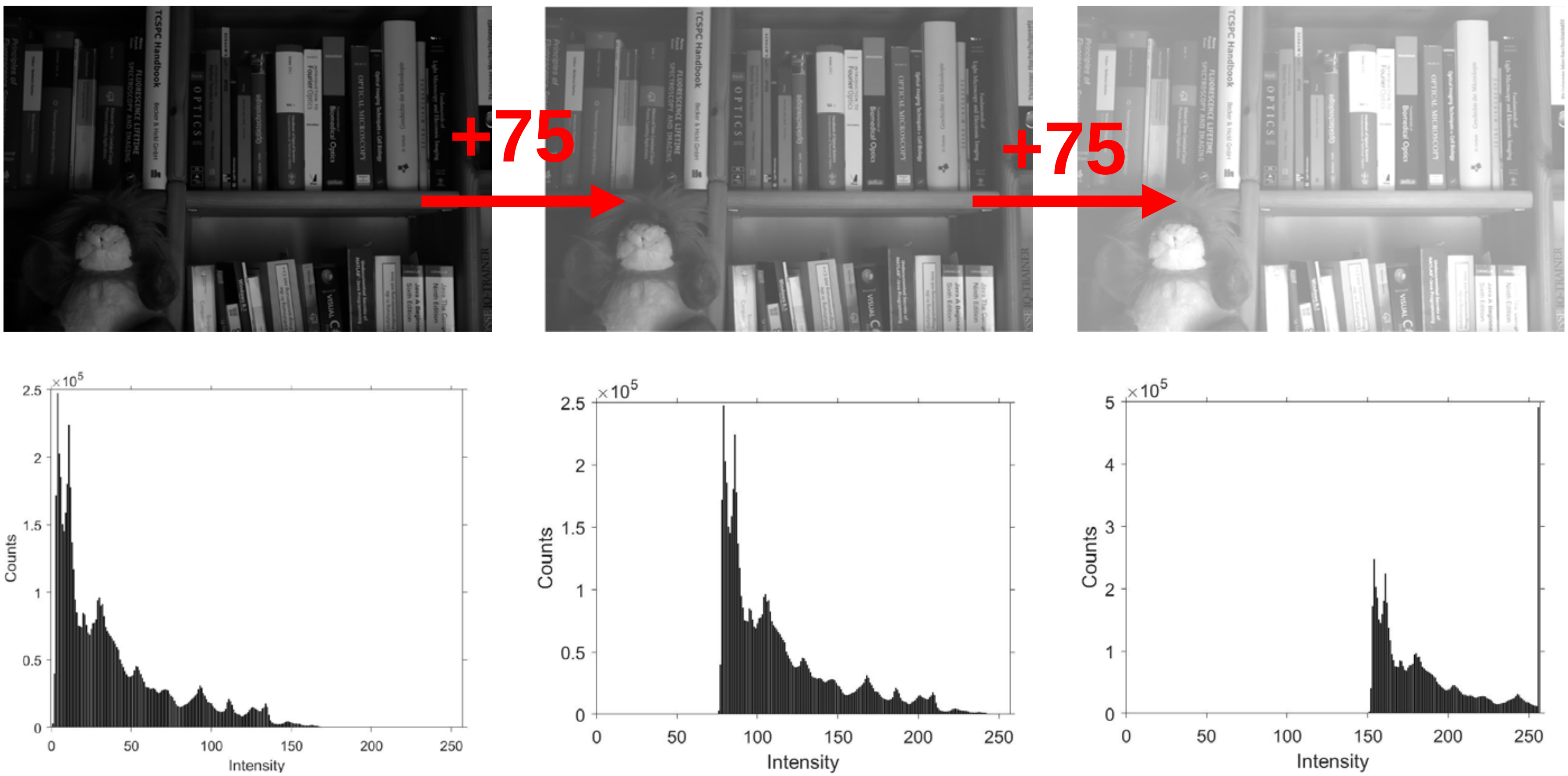


Fig. from: Gonzales, Woods, Eddins, Digital Image Processing using MATLAB, 2nd ed, Gatesmark (2009), p.85.

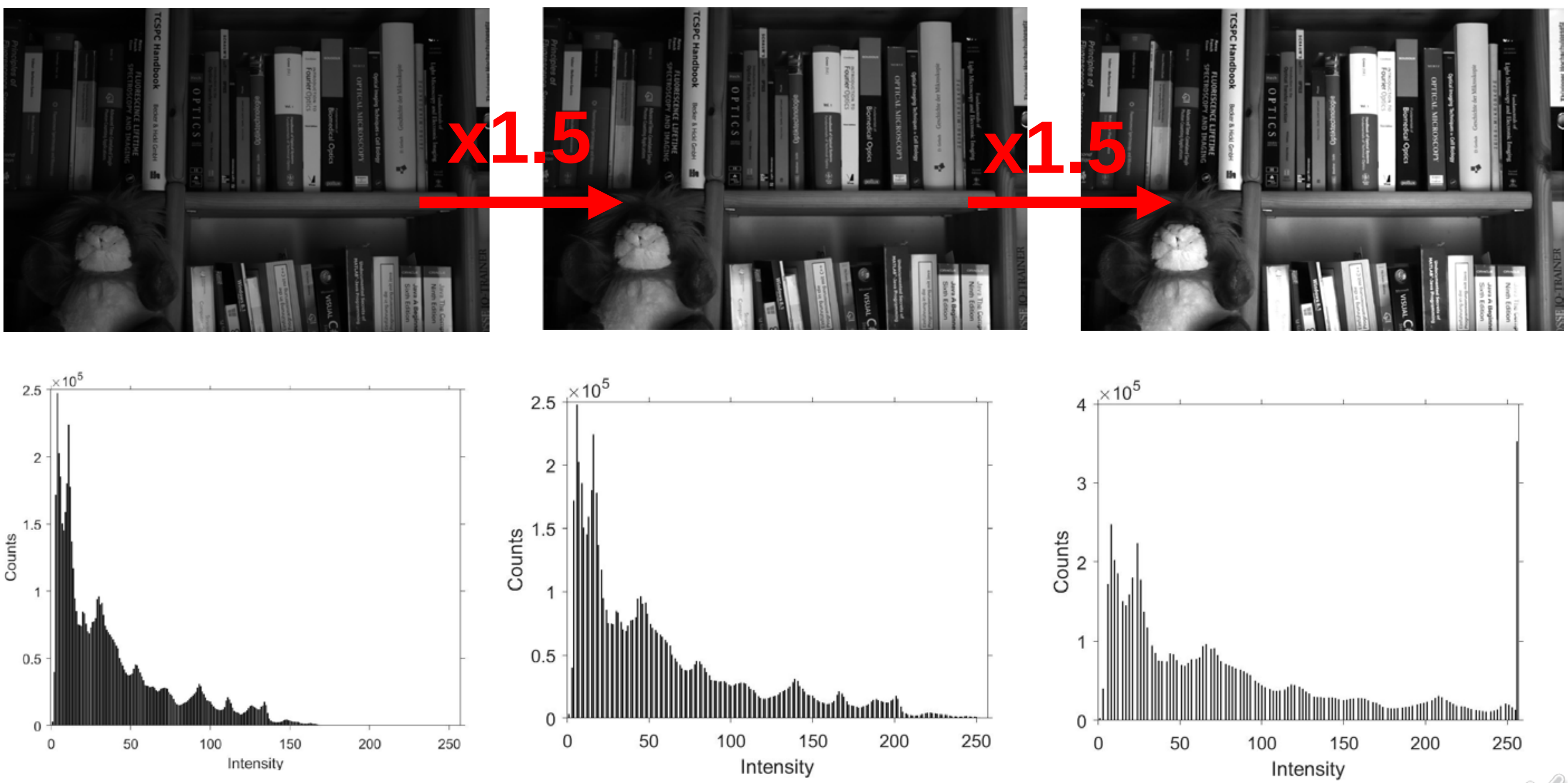
Adjusting brightness & contrast

Increasing the brightness



Adjusting brightness & contrast

Increasing the contrast (= gain)



Adjusting brightness & contrast

Contrast stretching

General scaling transformation to map gray level values I_f of the original image f to the transformed image g:

$$I_g = (I_f - I_{f,min}) \frac{(I_{g,max} - I_{g,min})}{(I_{f,max} - I_{f,min})} + I_{g,min}$$

where I_f gray level value in the original image f

$I_{f,min}, I_{f,max}$ range of interest

I_g mapped gray level value in the transformed image g

$I_{g,min}, I_{g,max}$ range to which the intensities are mapped

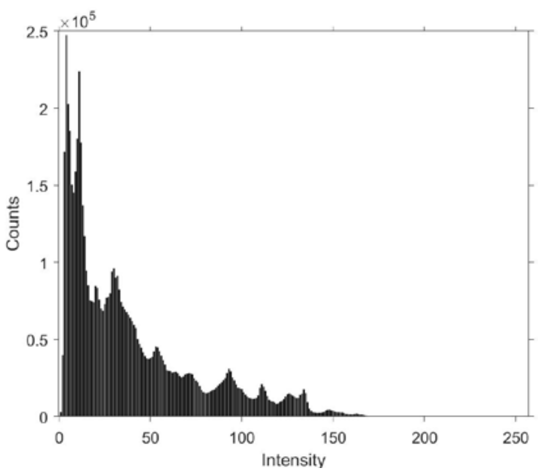
In MATLAB contrast stretching can be achieved with `imadjust`:

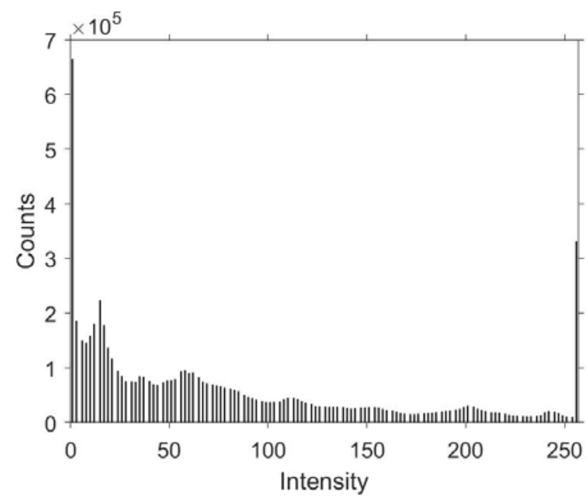
```
ImgOut = imadjust(ImgIn,[lowIn,highIn],[lowOut,highOut])
```

Adjusting brightness & contrast

Contrast stretching

contrast stretching





Adjusting brightness & contrast

Power law (gamma) transformations

General scaling transformation to map gray level values I_f of the original image f to the transformed image g:

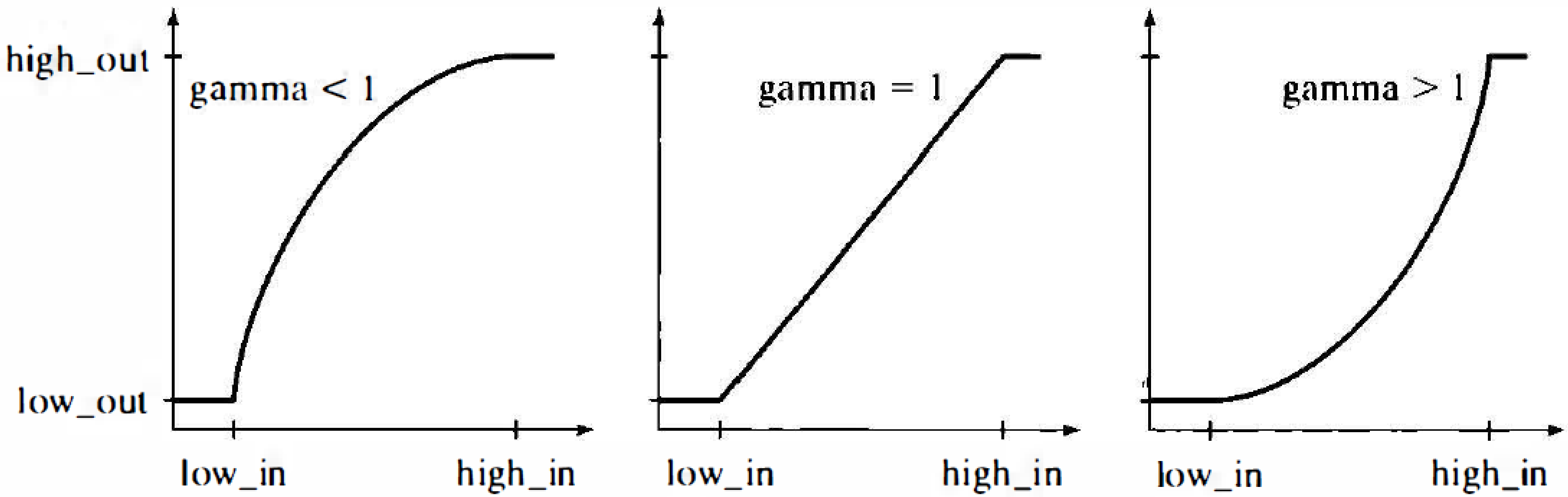
$$I_g = (I_{g,max} - I_{g,min})\left(\frac{I_f - I_{f,min}}{I_{f,max} - I_{f,min}}\right)^{\gamma} + I_{g,min}$$

where

I_f	gray level value in the original image f
$I_{f,min}$, $I_{f,max}$	range of interest
I_g	mapped gray level value in the transformed image g
$I_{g,min}$, $I_{g,max}$	range to which the intensities are mapped
γ	adjustable exponential factor ($\gamma > 0$)

Adjusting brightness & contrast
Power law (gamma) transformations

gamma specifies the shape of the curve that maps the intensity values :



In MATLAB also power law transformations can be done with `imadjust`:

```
ImgOut = imadjust(ImgIn,[lowIn,highIn],[lowOut,highOut],gamma)
```

Histograms

Contrast stretching in MATLAB

Syntax	Description
`ImgOut = imadjust(ImgIn)`	Maps the intensity values in the input image `ImgIn` to new values in the output `ImgOut`. By default, `imadjust` saturates the bottom 1% and the top 1% of all pixel values.
`ImgOut = imadjust(ImgIn, [lowIn,highIn], [lowOut,highOut])`	Maps intensity values in `ImgIn` to new values in `ImgOut` such that values between `lowIn` and `highIn` map to values between `lowOut` and `highOut`. Values below `lowIn` and above `highIn` are clipped. This means, values below `lowIn` map to `lowOut`, and those above `highIn` map to `highOut`. Note: All limits are specified as values between 0 and 1 , independently of the class of `ImgIn`.
`ImgOut = imadjust(ImgIn, [lowIn,highIn], [lowOut,highOut],gamma)`	Maps intensity values in `ImgIn` to new values in `ImgOut`, where gamma specifies the shape of the curve describing the relationship between the values in `ImgIn` and `ImgOut`.
`LowHigh = stretchlim(Img,Tol)`	Computes the lower and upper limits that can be used for contrast stretching. Tol specifies the percentage of pixels that are allowed to saturate at the bottom and at the top. The default is 1%. Limits are returned in `LowHigh`.
	See MATLAB help page for further information

Adjusting brightness & contrast

The test image

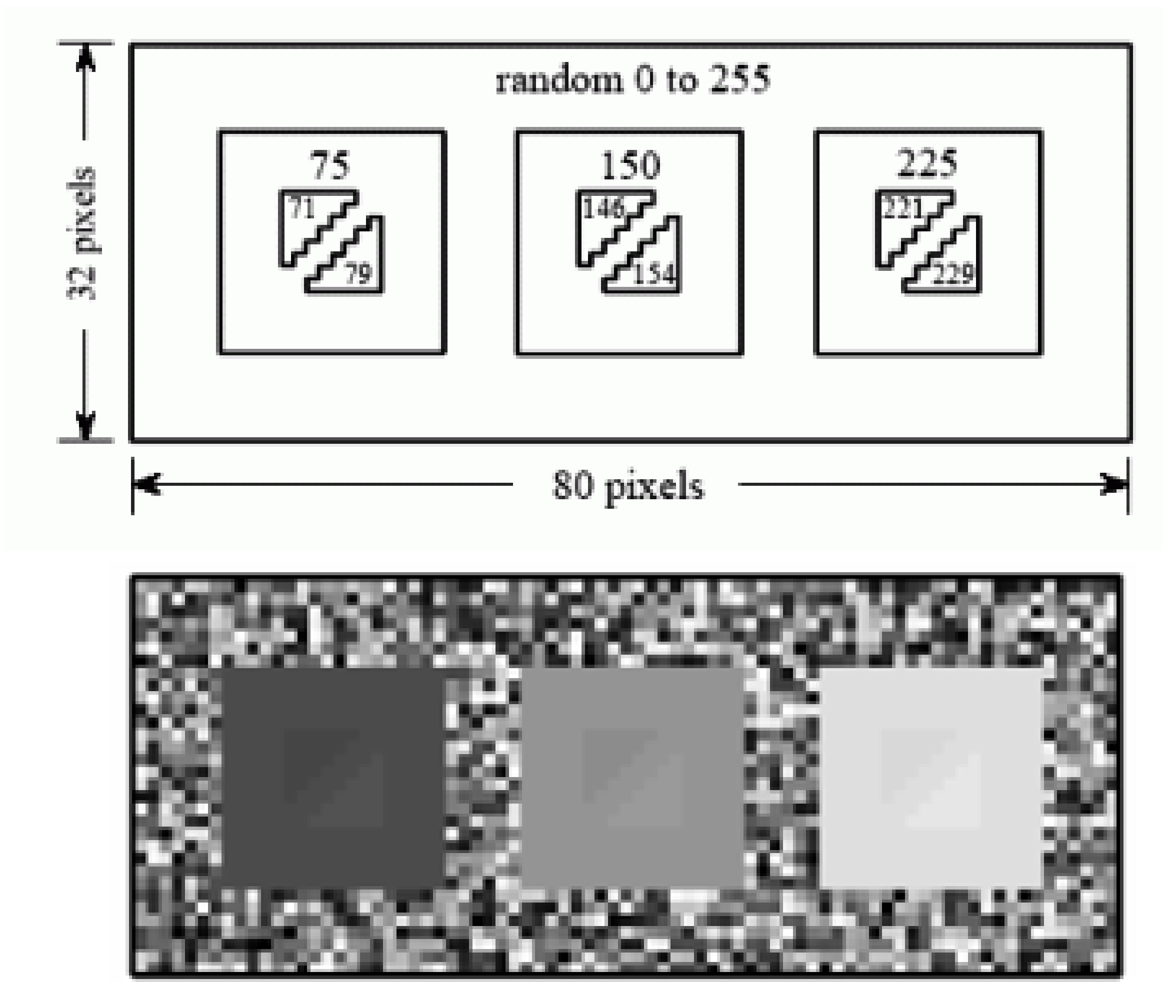


Fig. from: Smith, The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing (1997), p.389

Adjusting brightness & contrast

Adjusting the brightness (= offset)

Output = (Input + B) x C

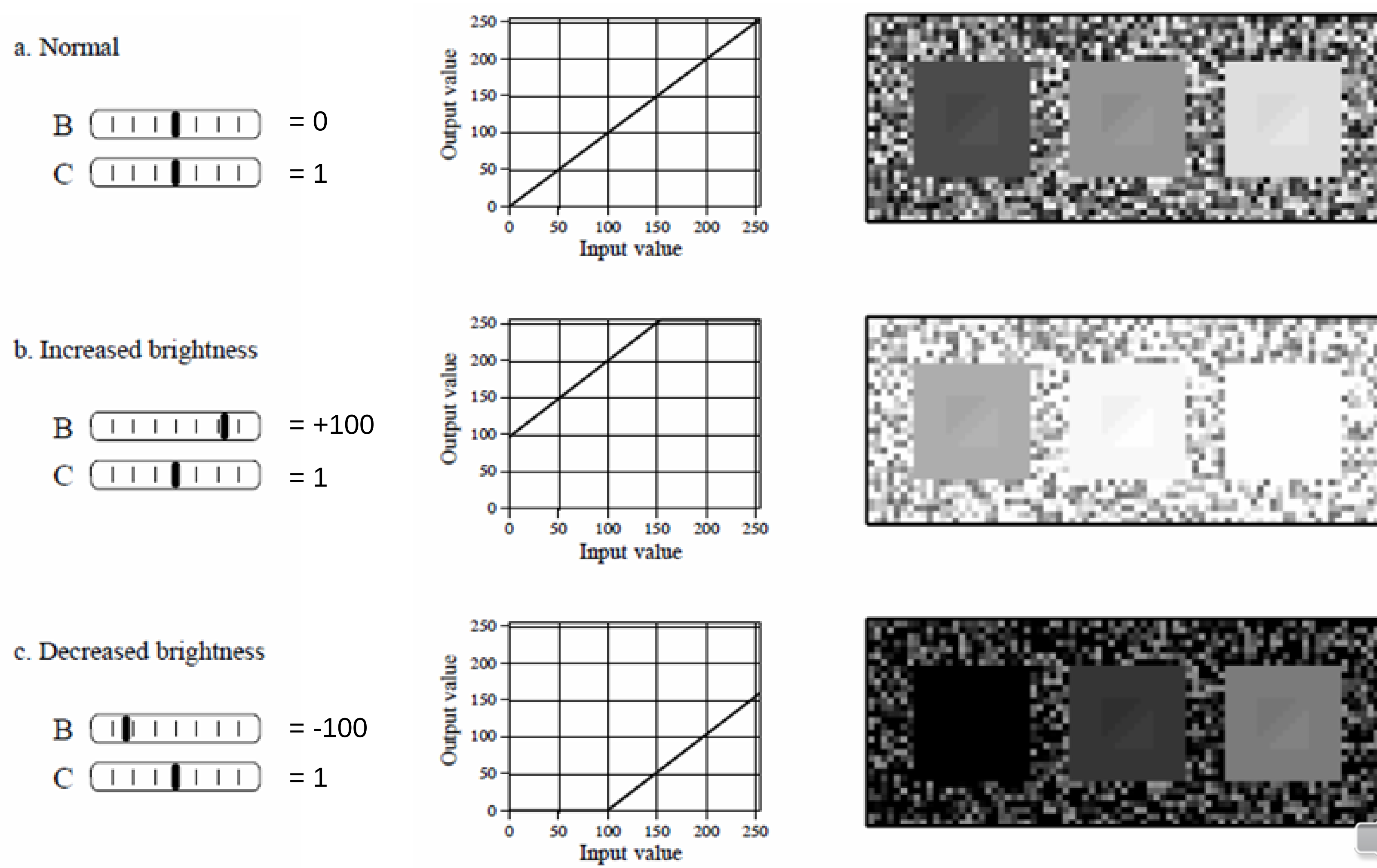


Fig. from: Smith, The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing (1997), p.391

Adjusting brightness & contrast

Adjusting the contrast (= slope)

Output = (Input + B) x C

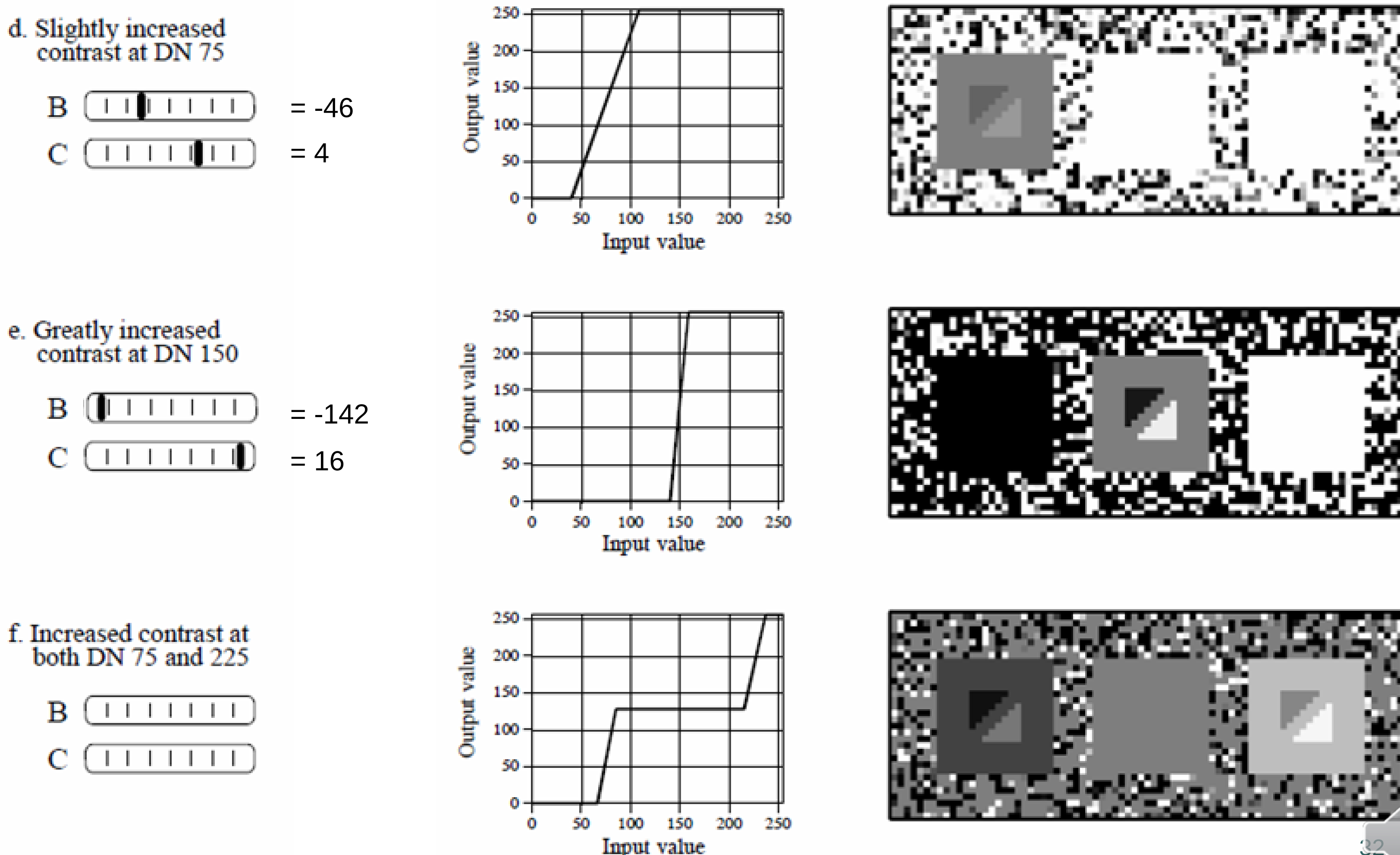


Fig. from: Smith, The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing (1997), p.391

Adjusting brightness & contrast

Adjusting the contrast (= slope)

Output = (Input + B) x C

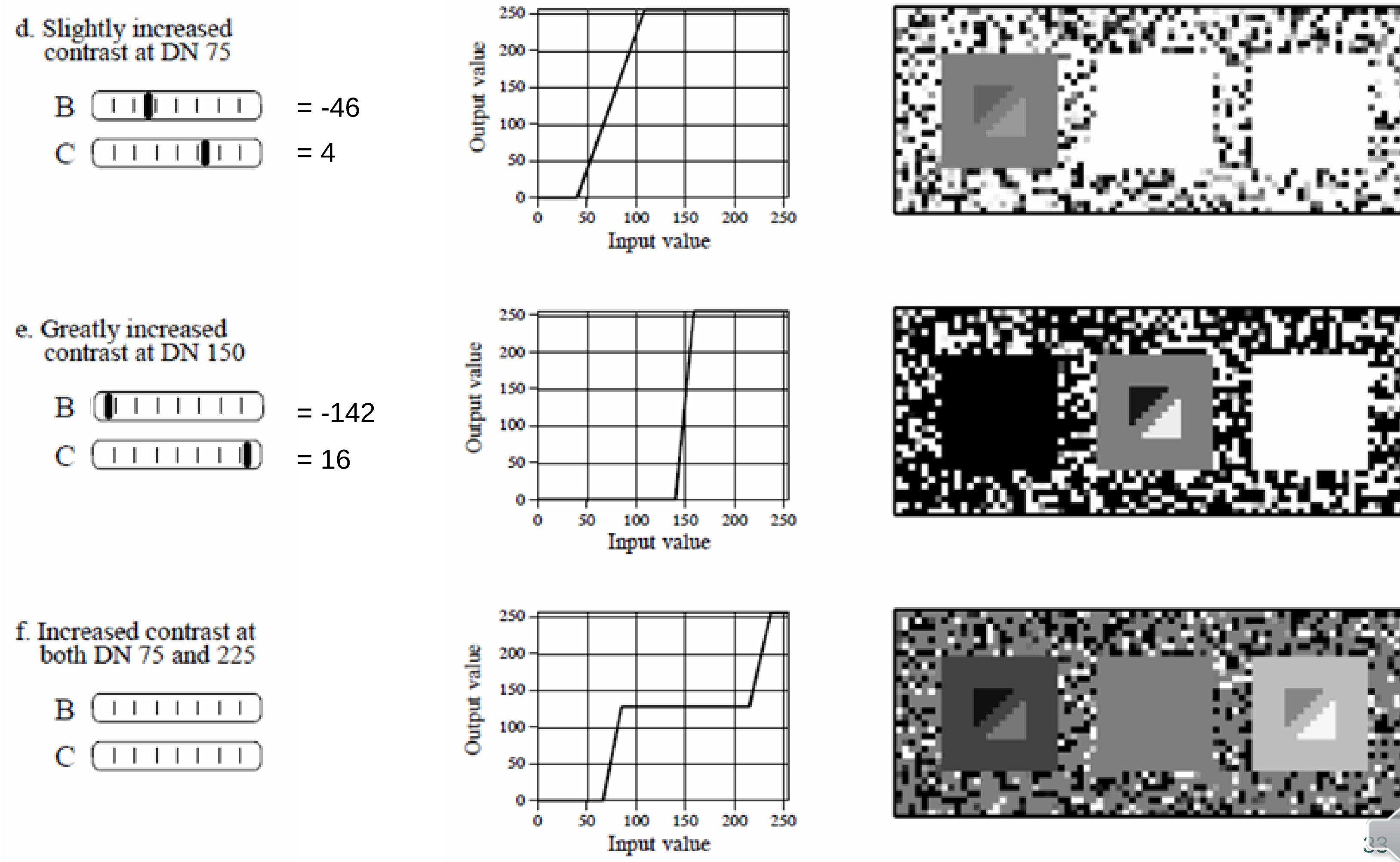


Fig. from: Smith, The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing (1997), p.391

Adjusting brightness & contrast

Adjusting the contrast (= slope)

Output = (Input + B) x C

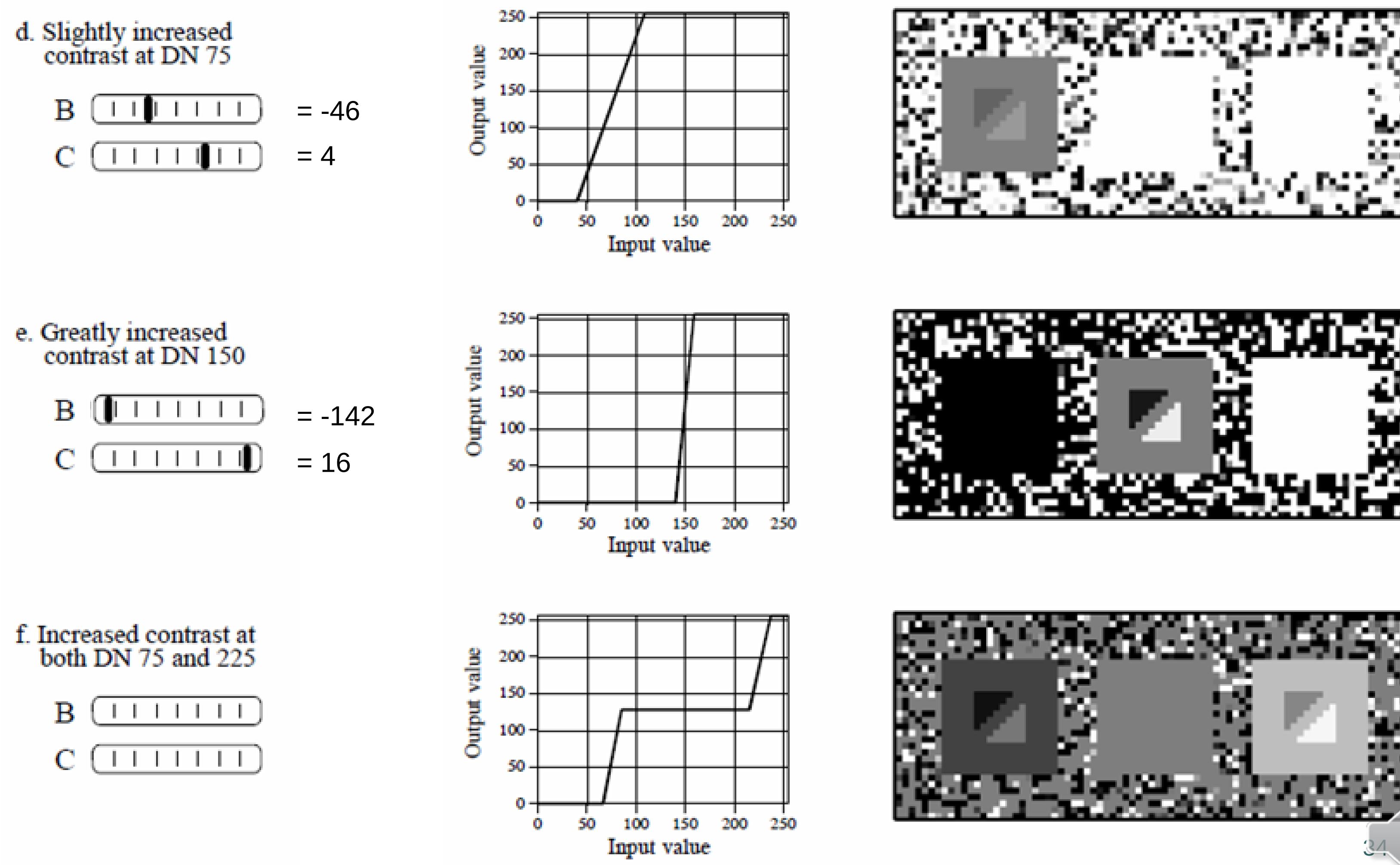
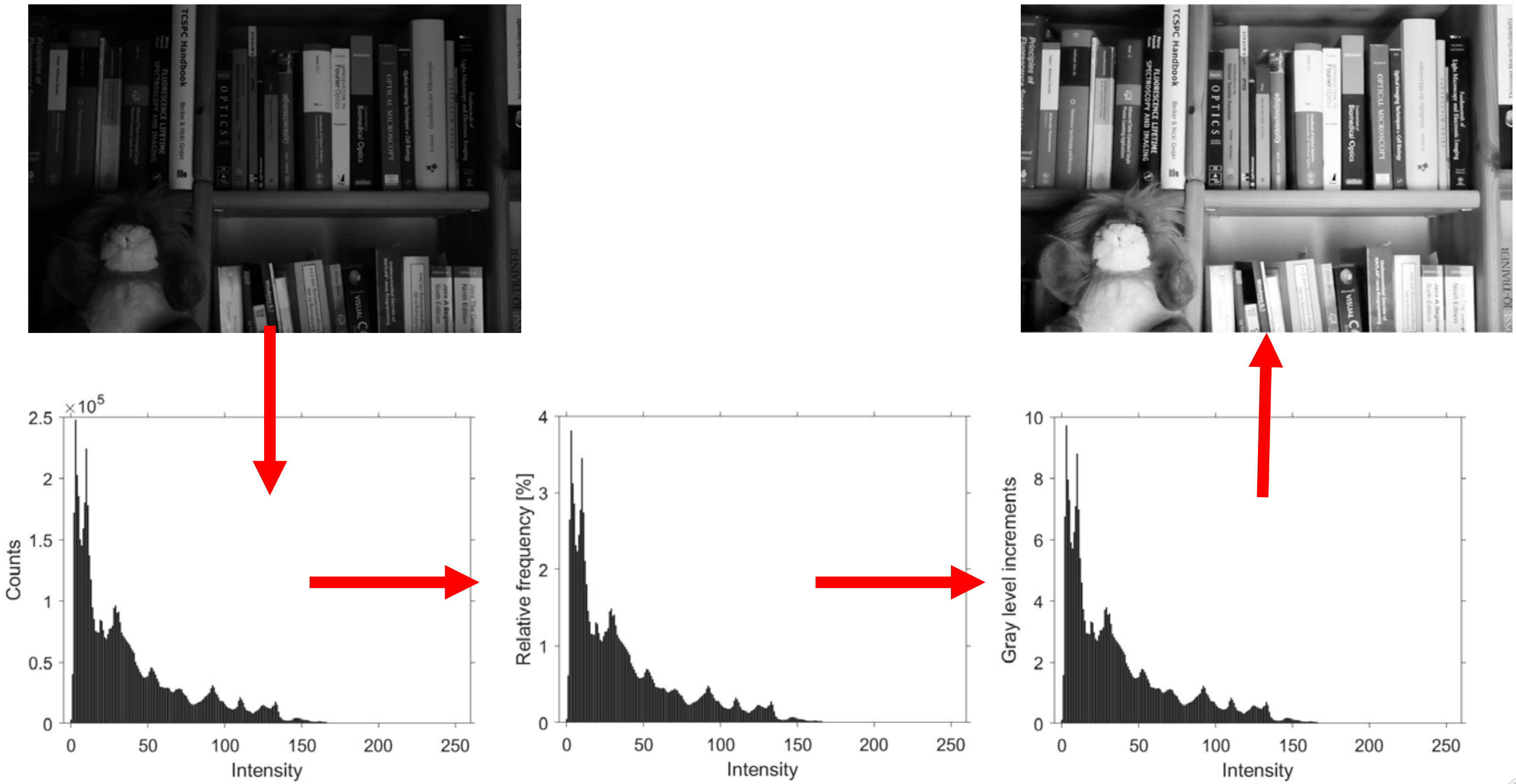


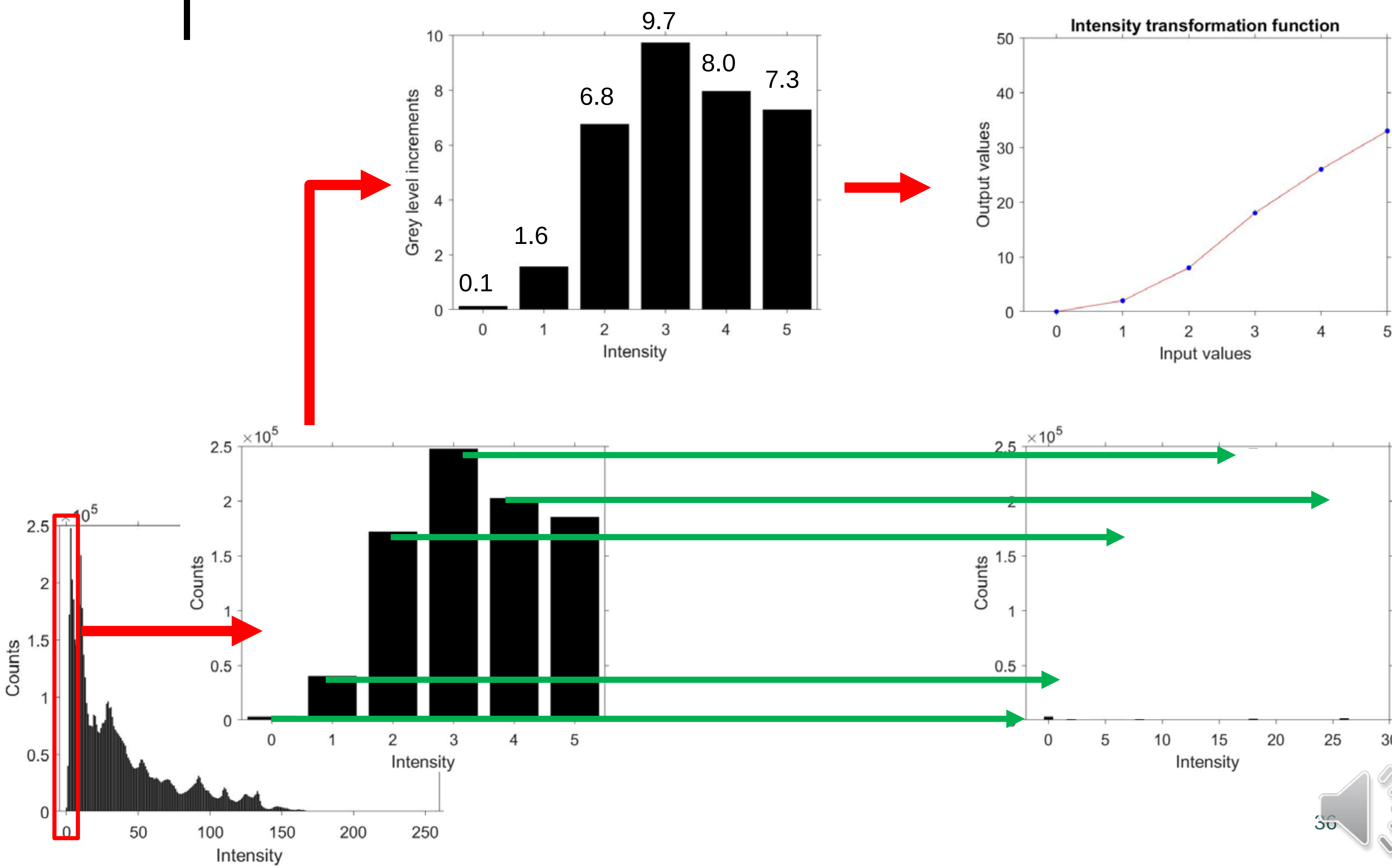
Fig. from: Smith, The Scientist and Engineer's Guide to Digital Signal Processing, California Technical Publishing (1997), p.391

Histogram equalization
Overview



Histogram equalization

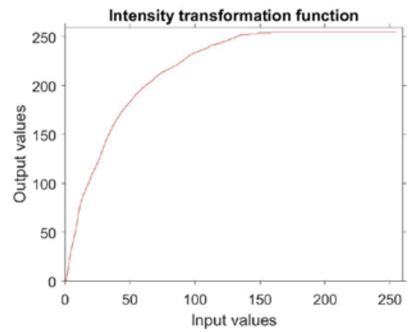
Step by step

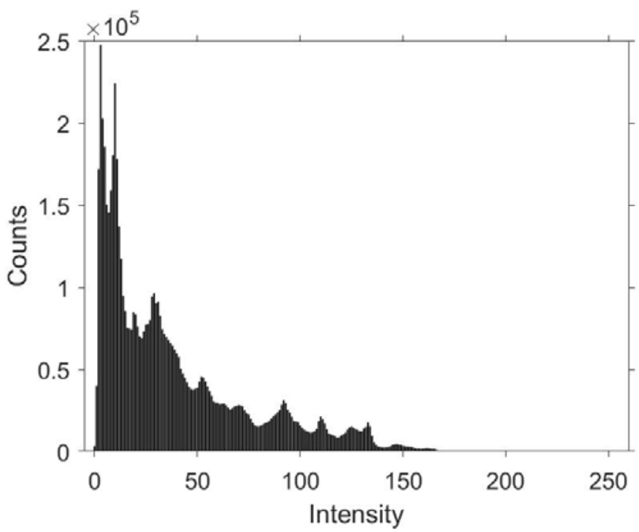


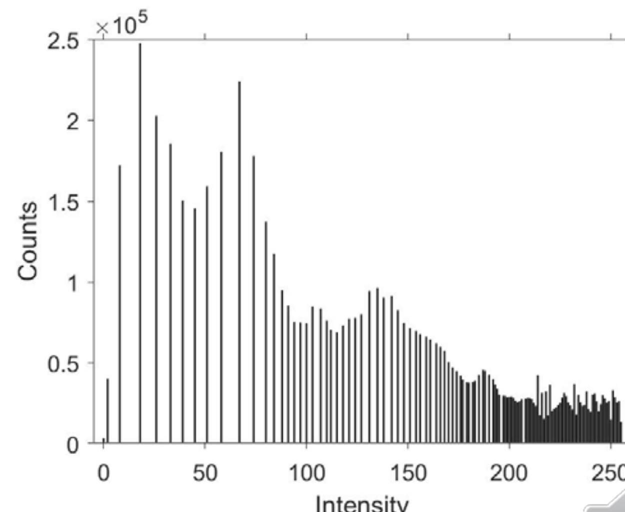
Histogram equalization

Overview

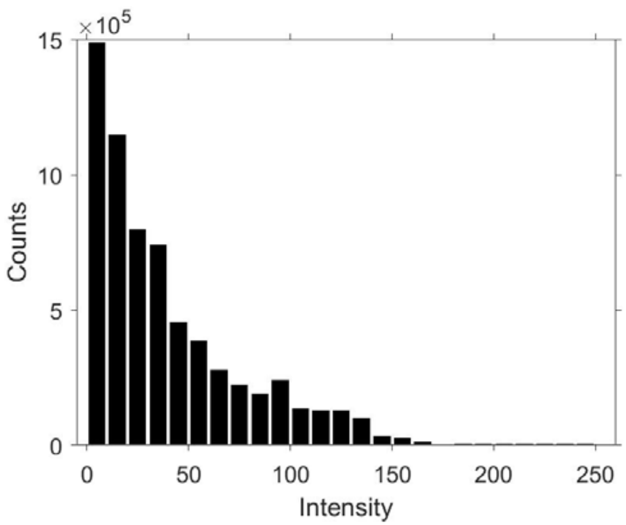
histogram equalization



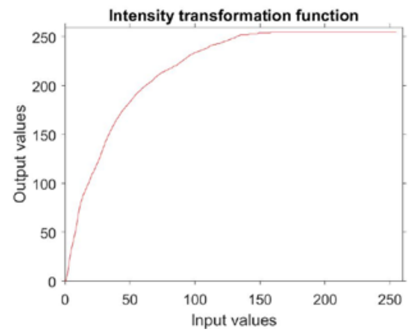


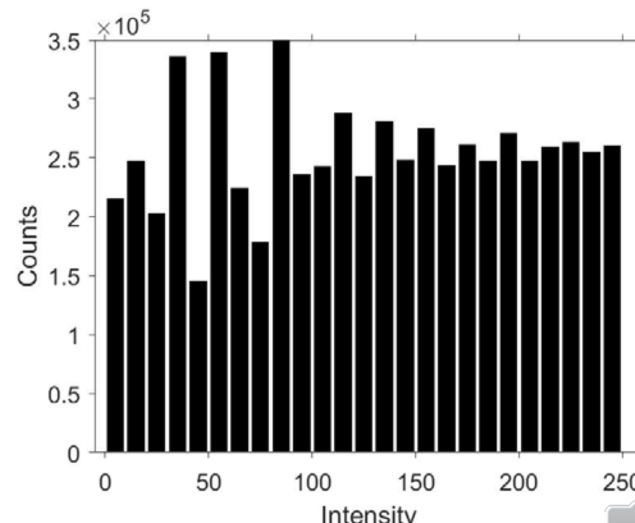


Histogram equalization
Overview



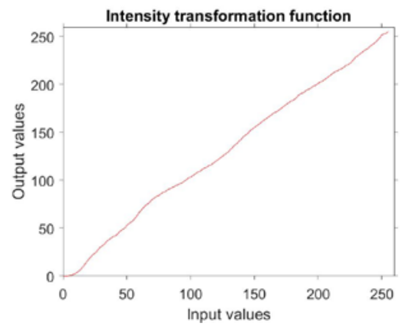
histogram equalization

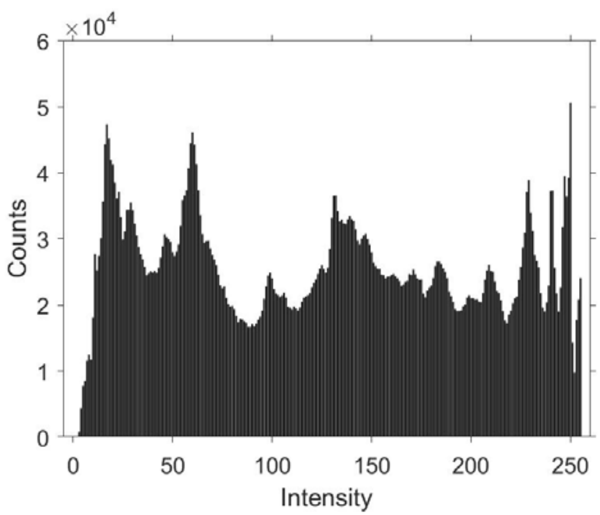


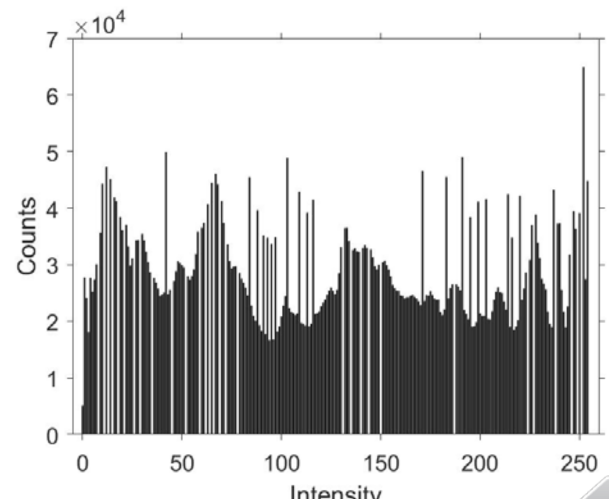


Histogram equalization
Overview

histogram equalization

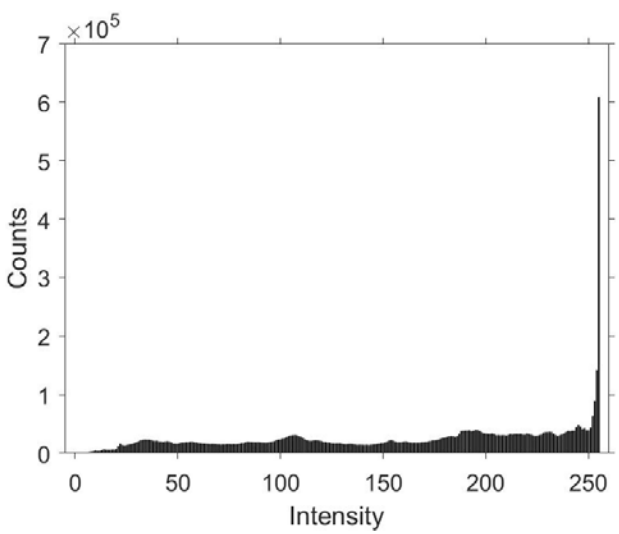




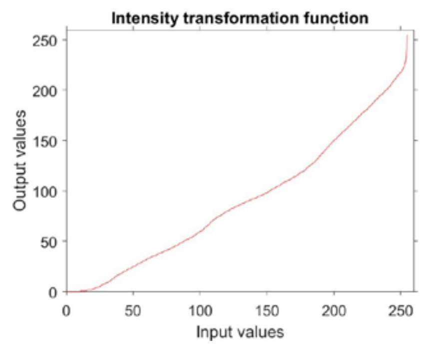


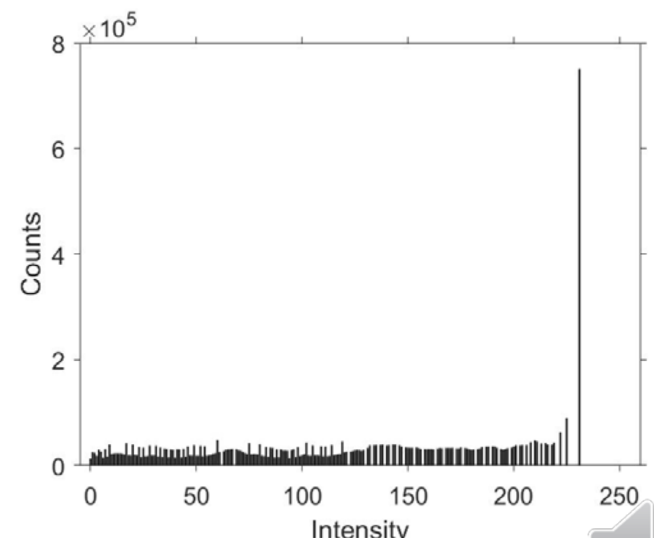
Histogram equalization

Overview



histogram equalization

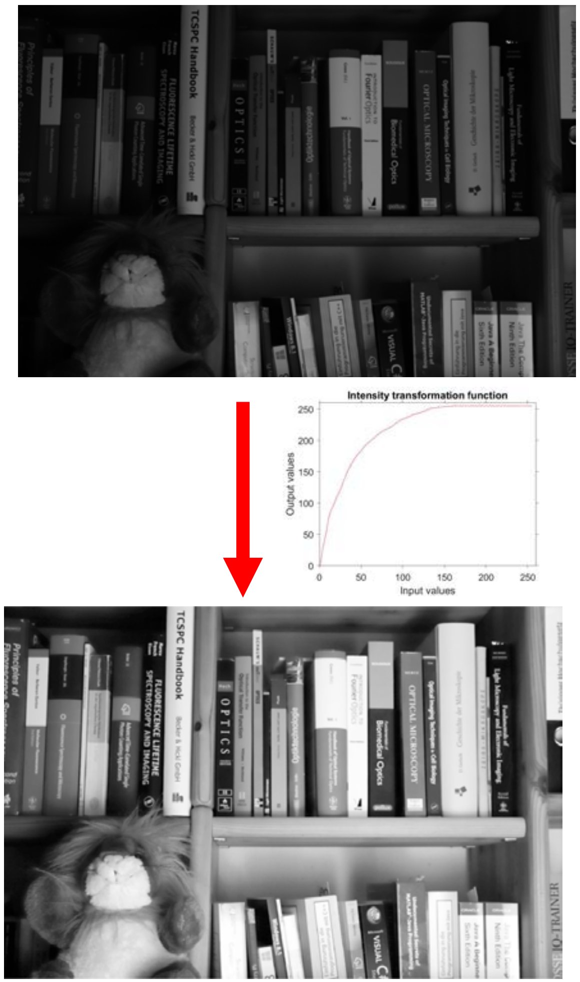




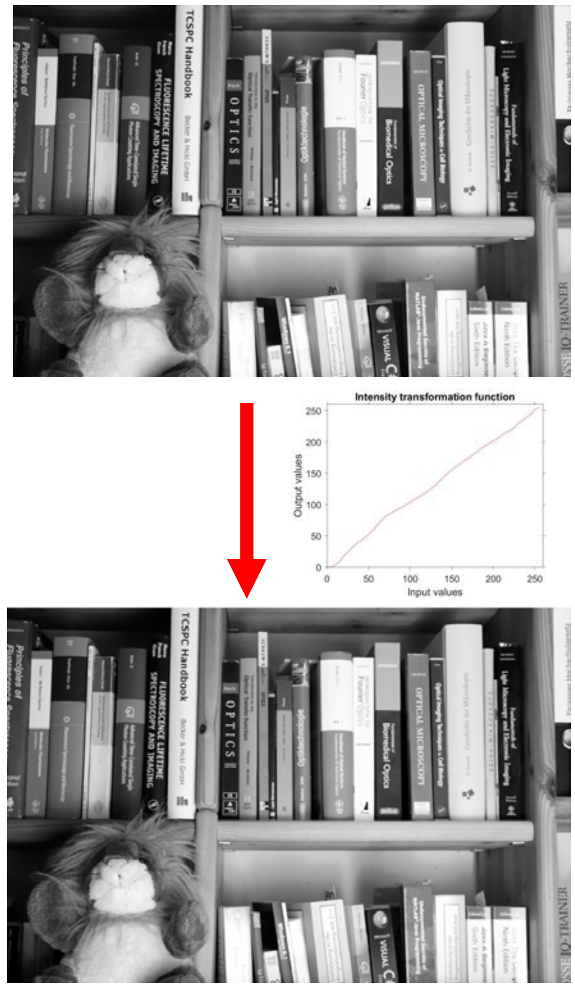
Histogram equalization

Comparison of the results

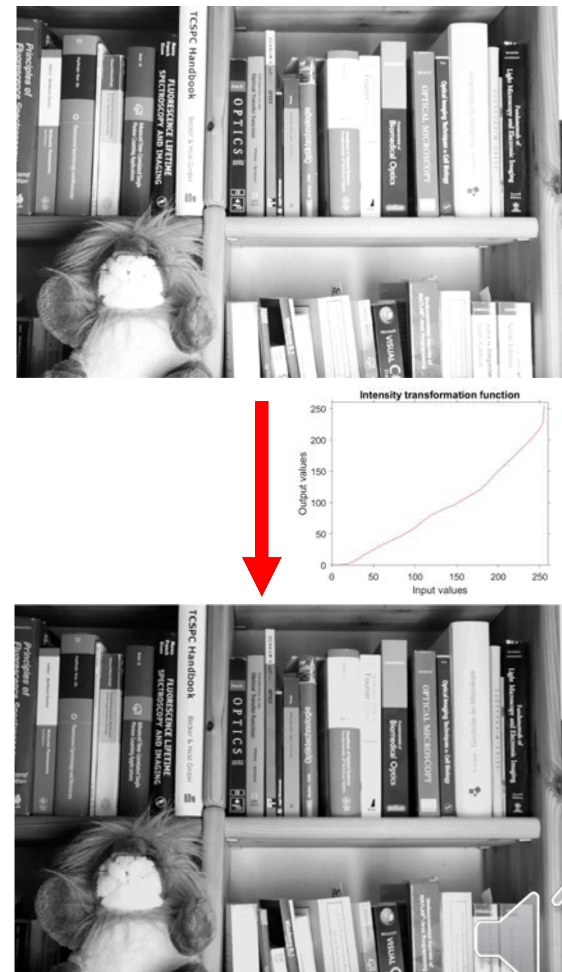
t = 20 ms



t = 125 ms



t = 250 ms



Histogram equalization & matching

Histogram equalization & matching in MATLAB

Syntax	Description
`ImgOut = histeq(ImgIn)`	transforms the grayscale image `ImgIn` to a grayscale image with `ImgOut` with the same number of discrete gray levels. A roughly equal number of pixels is mapped to each of the levels in `ImgOut`, so that the histogram of `ImgOut` is approximately flat.
`ImgOut = histeq(ImgIn,n)`	transforms the grayscale image `ImgIn` to a grayscale image with `ImgOut` with n discrete gray levels. A roughly equal number of pixels is mapped to each of the levels in `ImgOut`, so that the histogram of `ImgOut` is approximately flat. The histogram of `ImgOut` is flatter when n is much smaller than the number of discrete levels in `ImgIn`
`ImgOut = histeq(ImgIn,HistIn)`	transforms the grayscale image `ImgIn` so that the histogram of the output grayscale image `ImgOut` matches approximately the target histogram `HistIn`.
`[ImgOut,T] = histeq(ImgIn)`	returns the transformation function `T` that maps gray levels in the image `ImgIn` to gray levels in `ImgOut`.
	See MATLAB help page for further information