



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Introduction to Image Processing:

Point Operations

Marcus Hudritsch (hsm4)

Image Processing: Point Operations

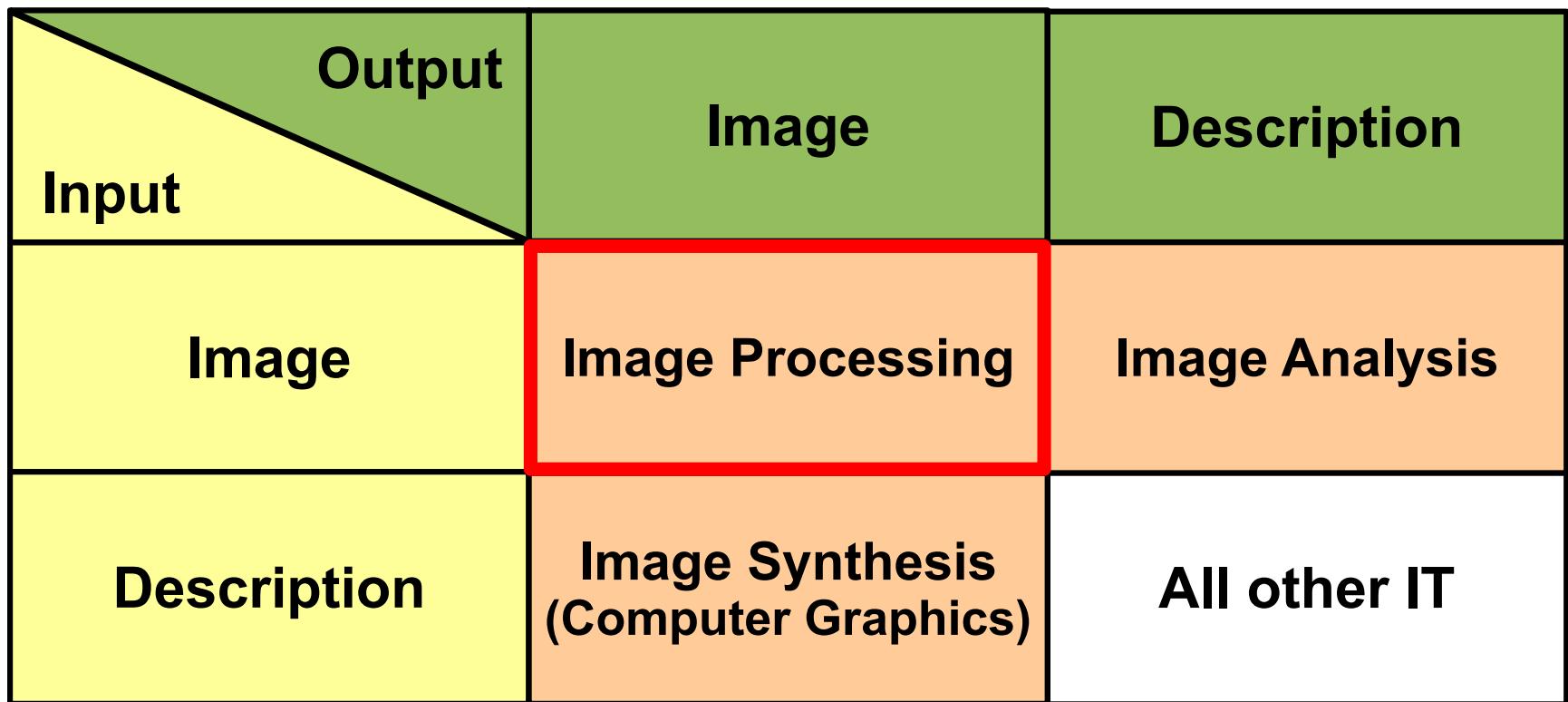


Image Processing vs. Image Analysis

Image Processing

- Contrast & brightness manipulation
- Color space manipulation
- Gray level reduction
- Sharpening
- Noise reduction
- Edge extraction
- Image algebra
- Geometric operations

Image Analysis

- Segmentation
- Region Representation
- Feature Extraction
- Classification
- Tracking

Image Operators

Point Operators

- Binarisation
- Gray level reduction
- Contrast & Brightness manipulations
- Histogram Equalization
- Arithmetic Operations
- Logic Operations

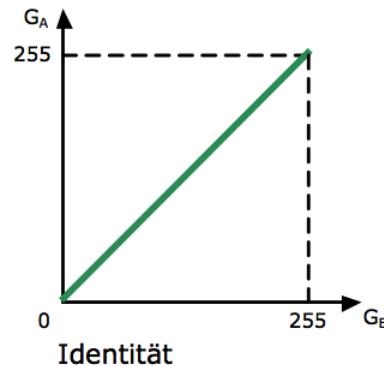
Global Operators

- Discrete Fourier Transform
- Wavelet Transform
- Hough Transform
- Principal Component Analysis

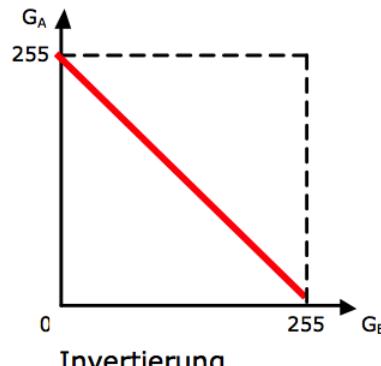
Local Operators

- Filters
 - Low Pass Filter
 - Gauss & Box Filter
 - High Pass Filter
 - Sobel Filter
 - Laplace Filter
- Morphological Operators
 - Erosion & Dilation
 - Opening & Closing
- Rank Order Operators
 - Min. & Max. Filter
 - Median Filter

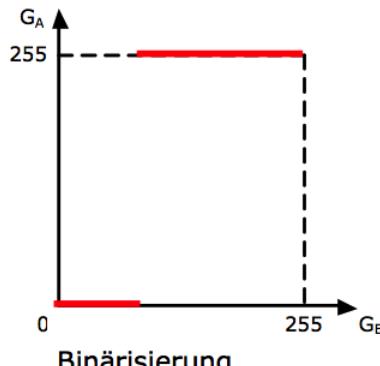
Point Operation: Mapping Function



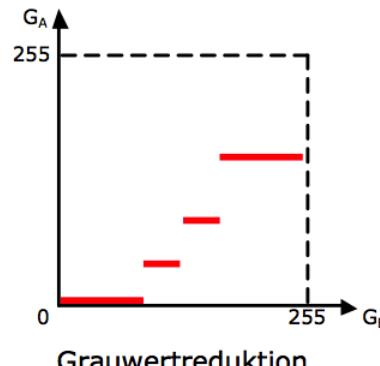
Identität



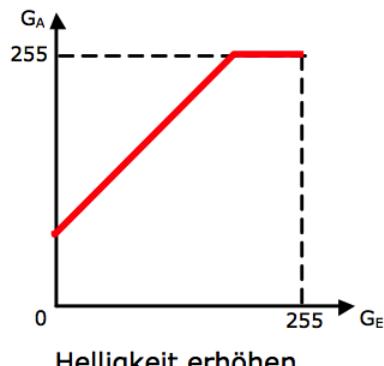
Invertierung



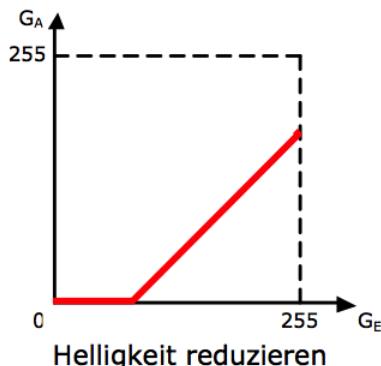
Binärisierung



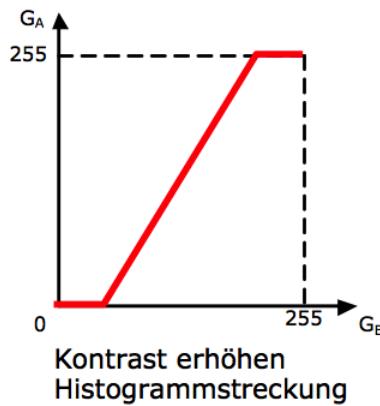
Grauwertreduktion



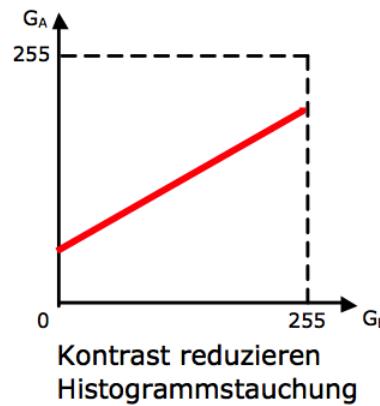
Helligkeit erhöhen



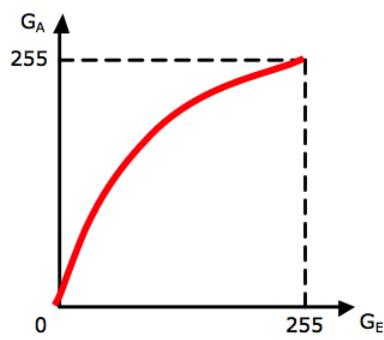
Helligkeit reduzieren



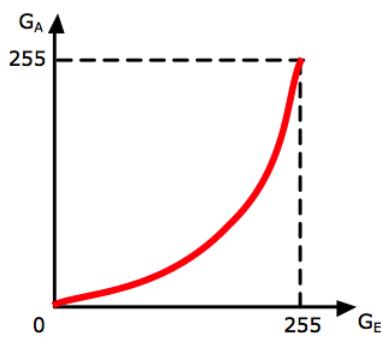
Kontrast erhöhen
Histogrammstreckung



Kontrast reduzieren
Histogrammstauchung

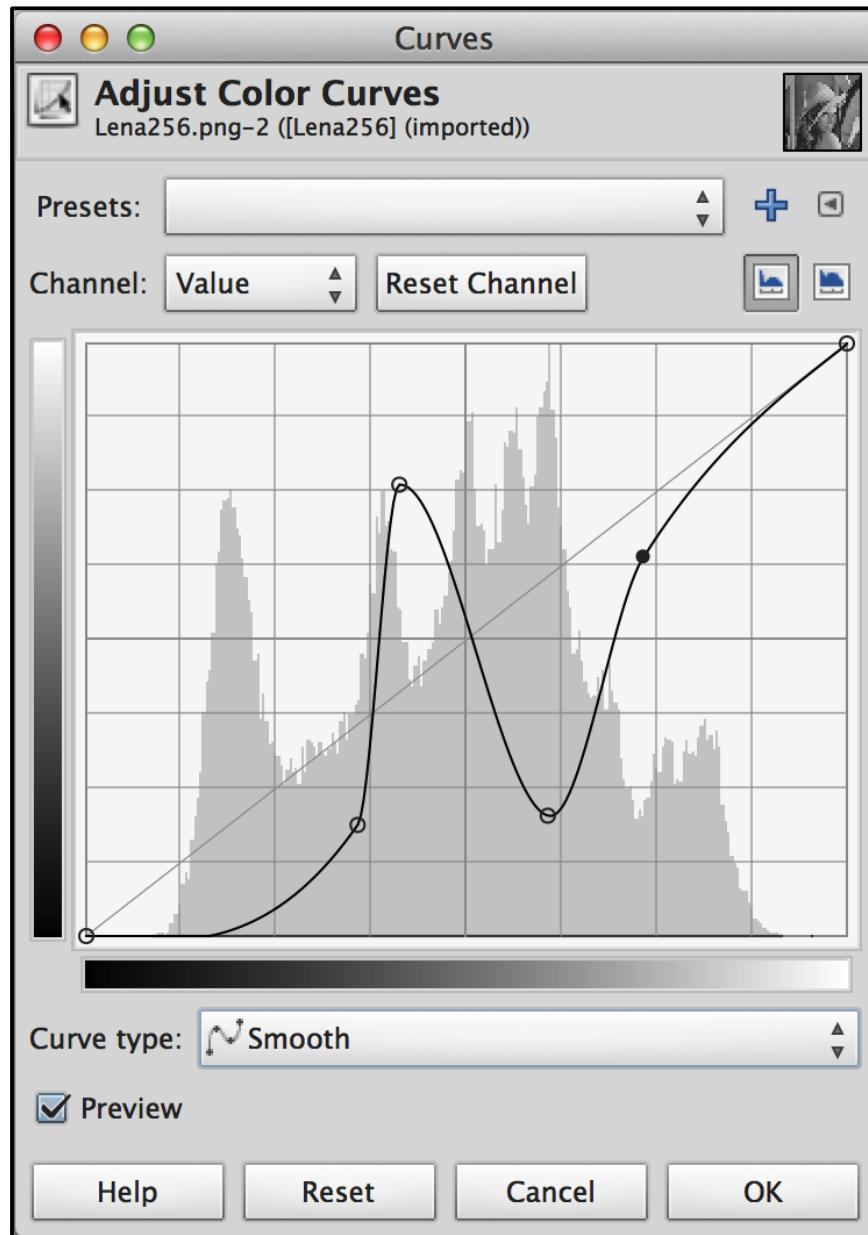


Helligkeit erhöhen
(Gammakorrektur)



Helligkeit reduzieren
(Gammakorrektur)

Point Operation: Mapping Function in Gimp



Point Operation: Mapping Function

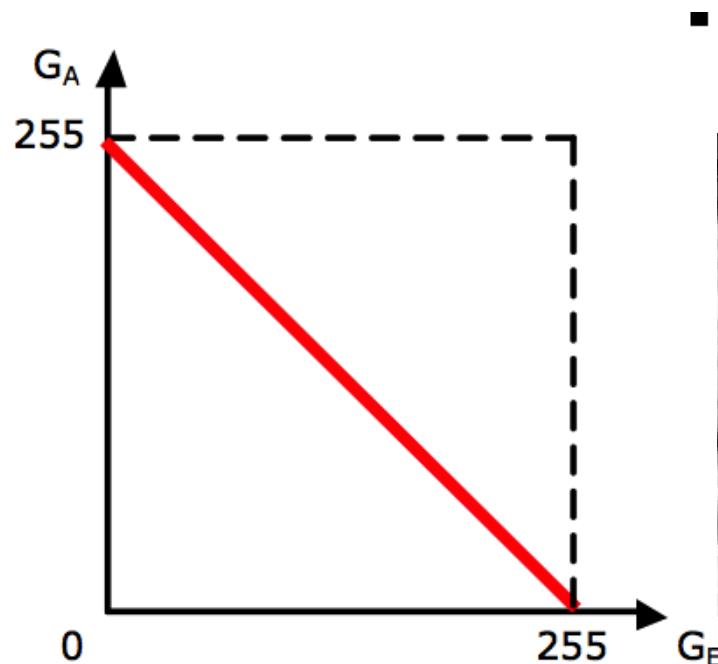
- Mapping functions are first applied to a **lookup table**:

Input	0	1	2	3	4	5	6	7	...	250	251	252	253	254	255
Output	255	254	253	252	251	250	249	248	...	5	4	3	2	1	0

- The point operation is then a **simple and fast lookup operation**.

Point Operation: Invert

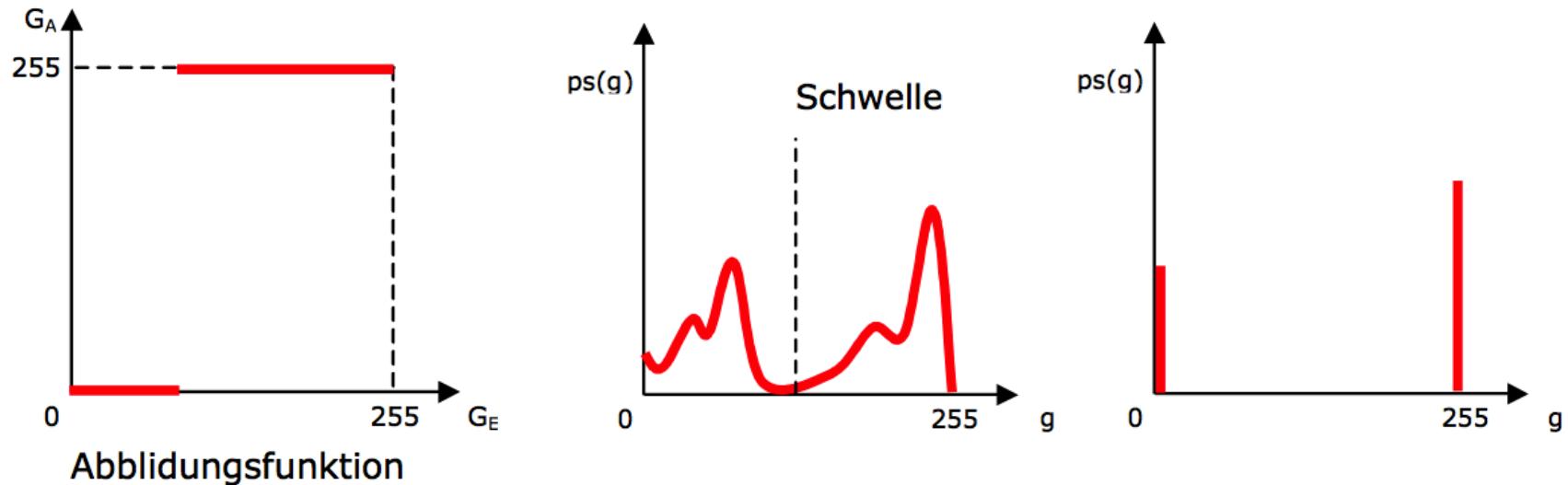
- Invert all gray values: $g_A(x, y) = 255 - g_E(x, y)$



Point Operation: Binarisation

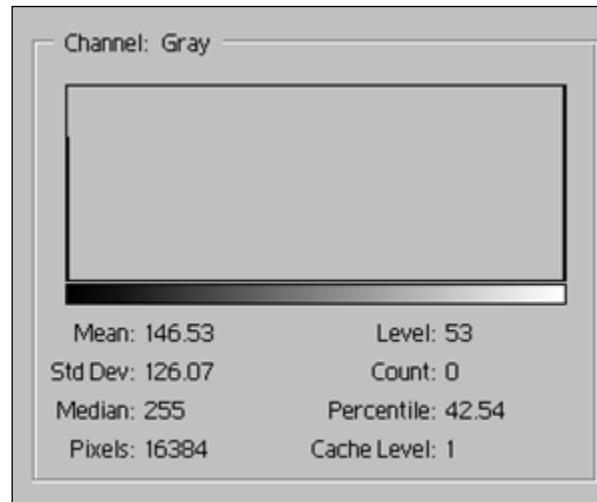
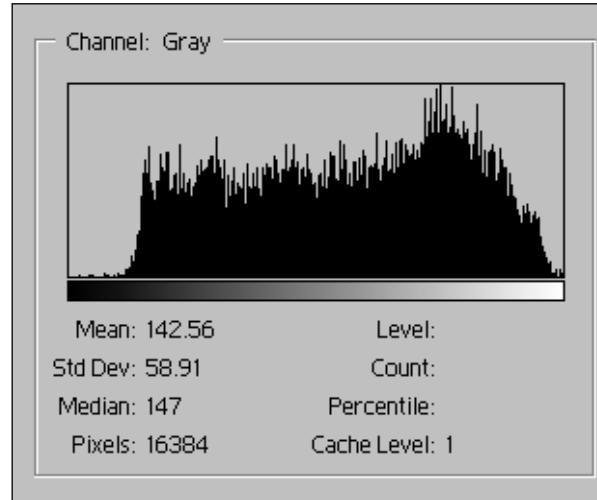
- All gray values below **threshold t** become black and all other become white:

$$g_A(x, y) = \begin{cases} g_1 : g_E(x, y) < t \\ g_2 : g_E(x, y) \geq t \end{cases}$$



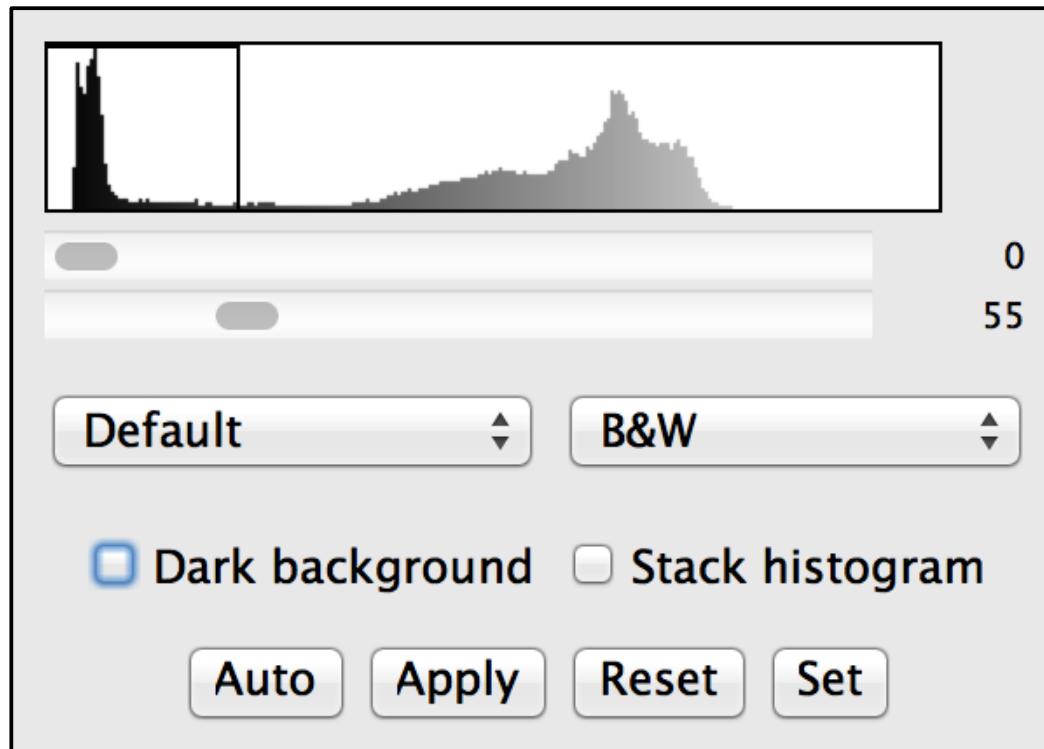
- Calculating the **threshold t automatically** is not always possible.
- Sometimes there is **no global threshold** and an **adaptive threshold** must be calculated.

Point Operation: Binarisation



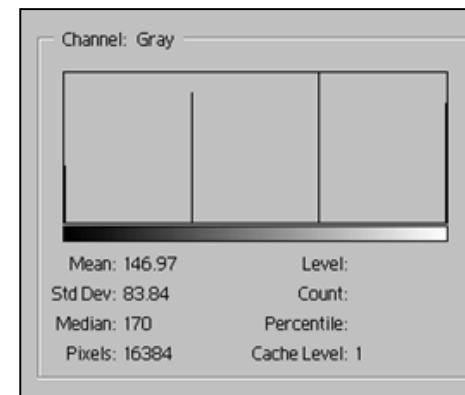
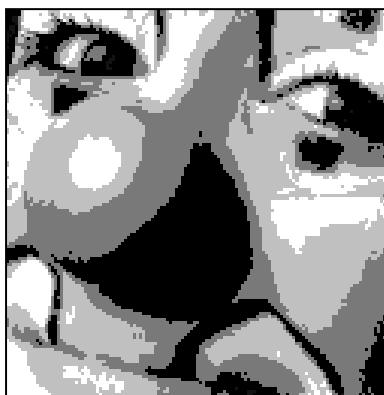
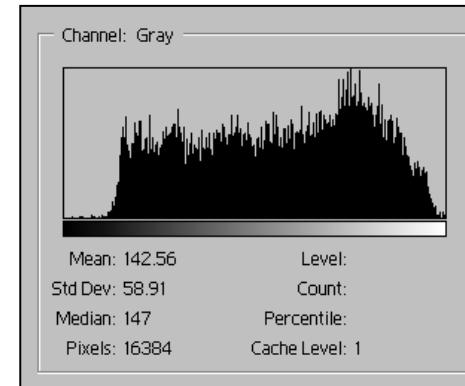
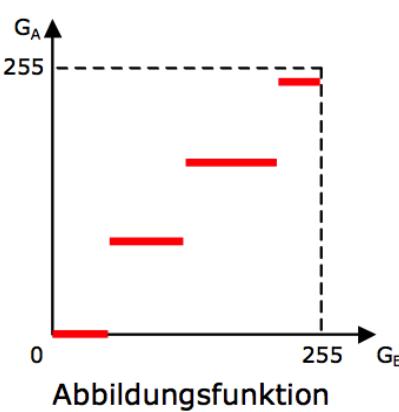
Point Operation: Binarisation

- Manual setting threshold in **ImageJ**: Image > Adjust > Threshold



Point Operation: Gray Level Reduction

- Reducing the NO. of gray levels is also known as gray level slicing.
- It is often a prestep for edge detection.
- The Reduction is mostly done with a **lookup table**.
- Well known algorithm: **Median Cut Algorithm** from Paul Heckbert
- Used for bit depth reduction for image formats (e.g. BMP, PNG & GIF)

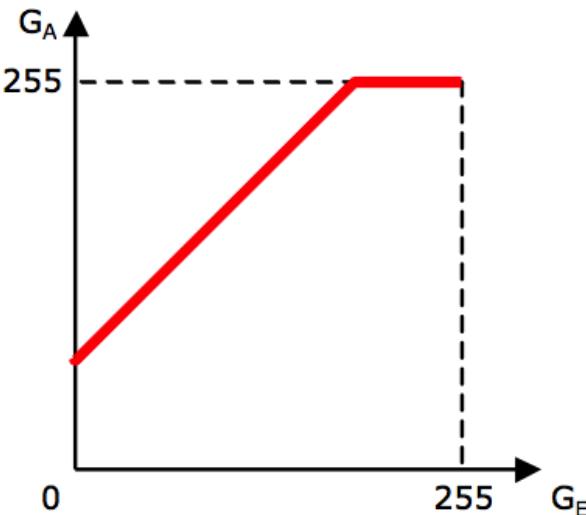


Point Operation: Linear Brightness Correction

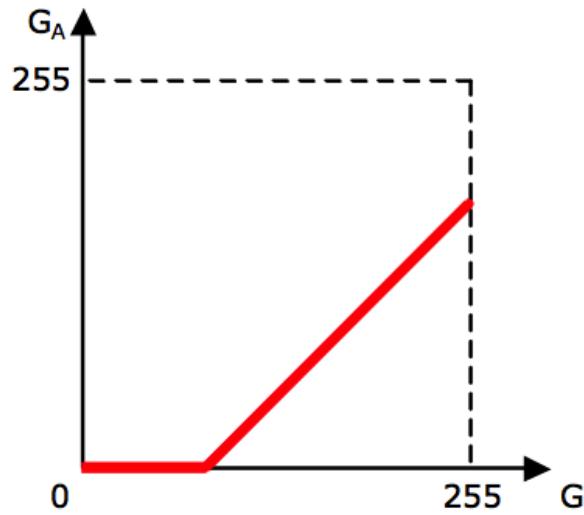
- Linear brightness correction can be expressed in the line equation:

$$g_A(x, y) = c \cdot g_E(x, y) + b$$

where c changes the contrast and **b the brightness.**



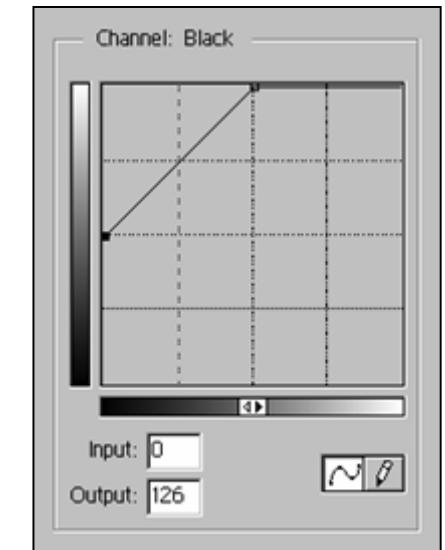
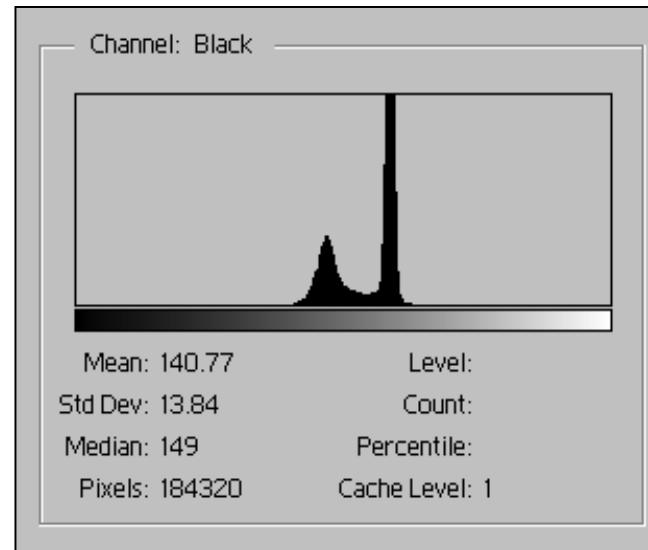
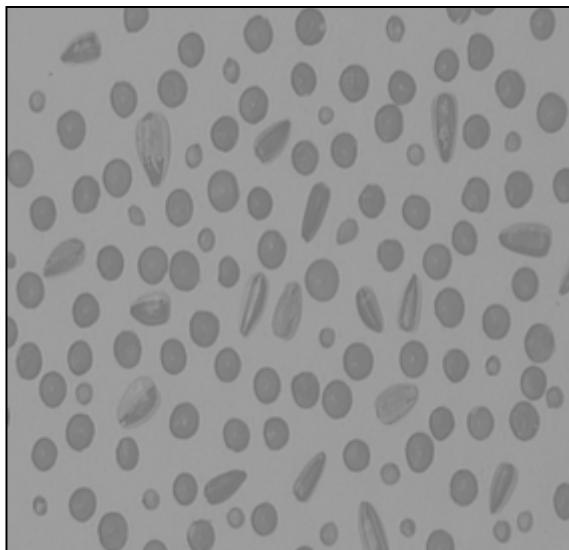
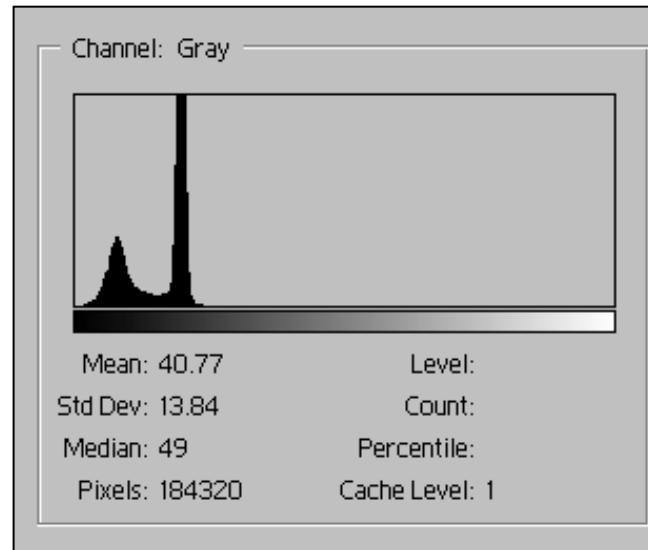
Abbildungsfunktion
Helligkeit erhöhen



Abbildungsfunktion
Helligkeit reduzieren

Point Operation: Linear Brightness Correction

- The histogram is shifted:

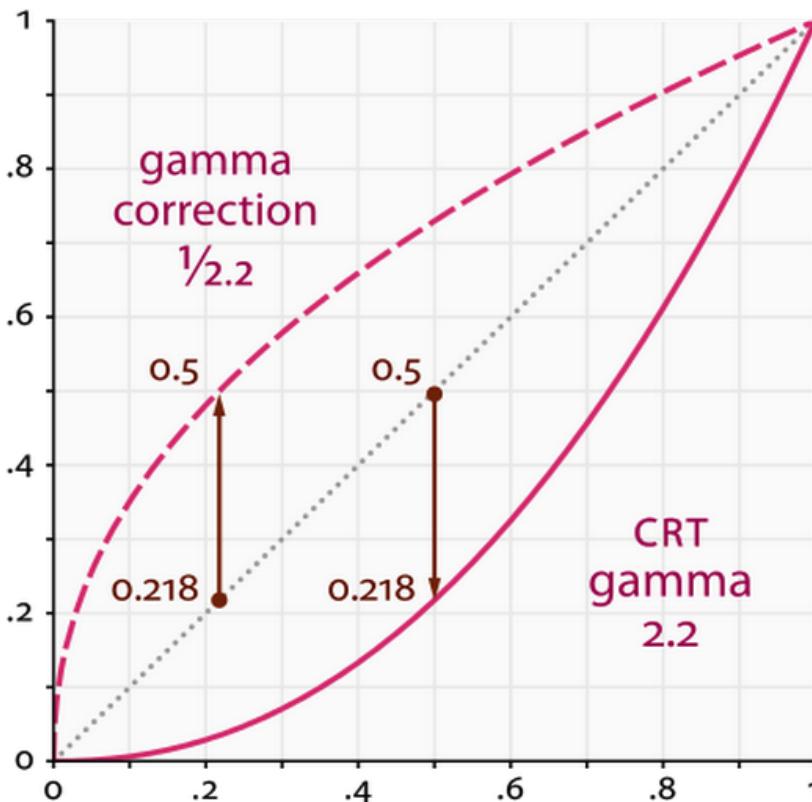


Point Operation: Gamma Correction

- The **gamma correction** is a **non linear** brightness correction:

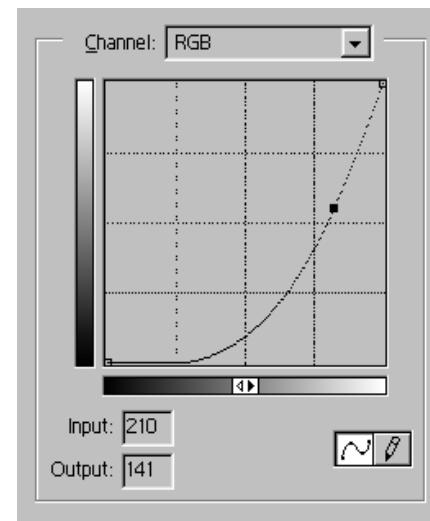
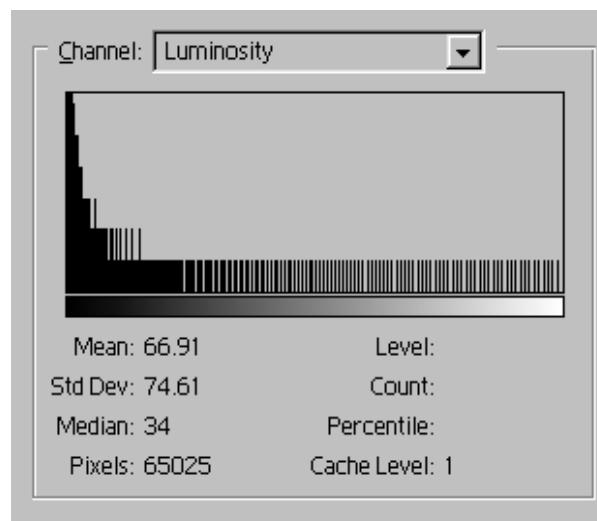
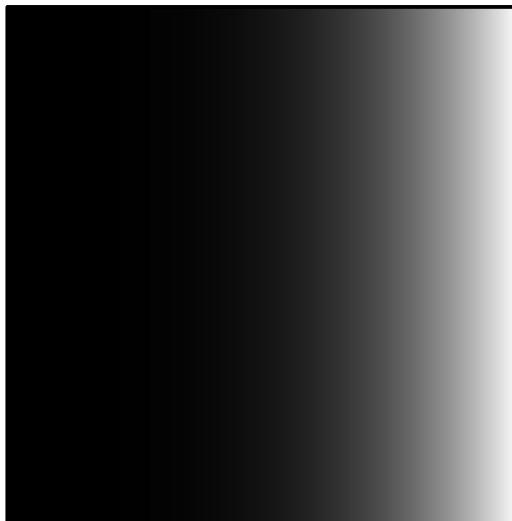
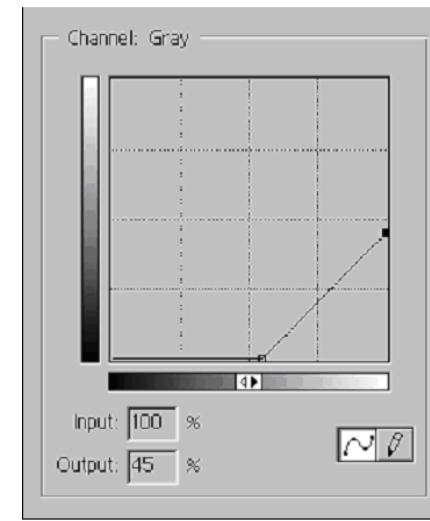
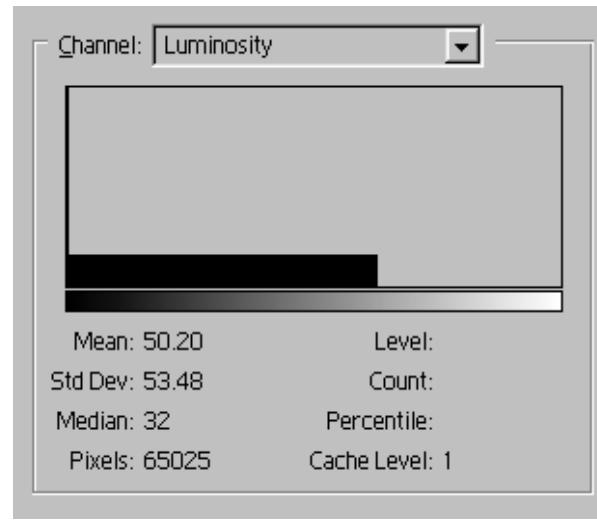
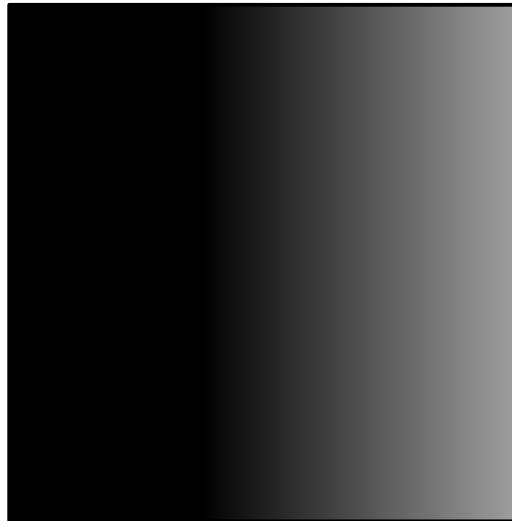
$$g_A(x, y) = 255 \cdot \left(\frac{g_E(x, y)}{255} \right)^\gamma$$

where $\gamma < 1$ **increases** and $\gamma > 1$ **decreases** the brightness:



Point Operation: Gamma Correction

- The gamma correction **preserves information** in the dark & bright areas.



Point Operation: Gamma Correction



$\gamma=2$



$\gamma=1$ (original)



$\gamma=1/2$



$\gamma=1/3$



$\gamma=1/4$

Image: en.wikipedia.org/wiki/Gamma_correction

Point Operation: Gamma Correction

- Gamma correction is important because **analog devices have non linear sensitivity**.
- E.g. analog **film material** has a **non linear sensitivity**:

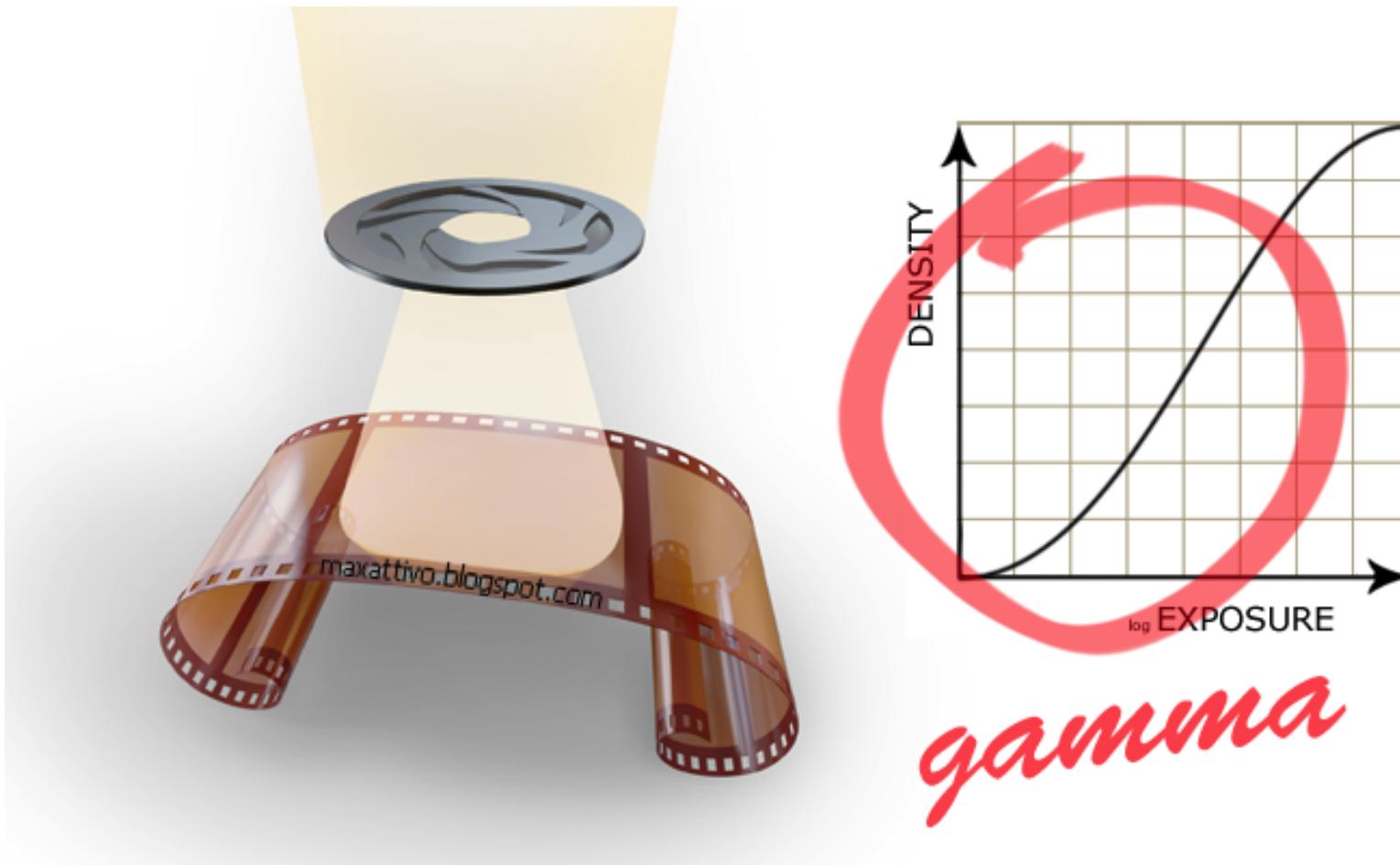


Image: maxattivo.blogspot.ch

11.03.16

BFH-TI > Specialization CPVR > Introduction to Computer Vision

18/43

Point Operation: Gamma Correction

- **CRT Monitors** display power is **not linear**:

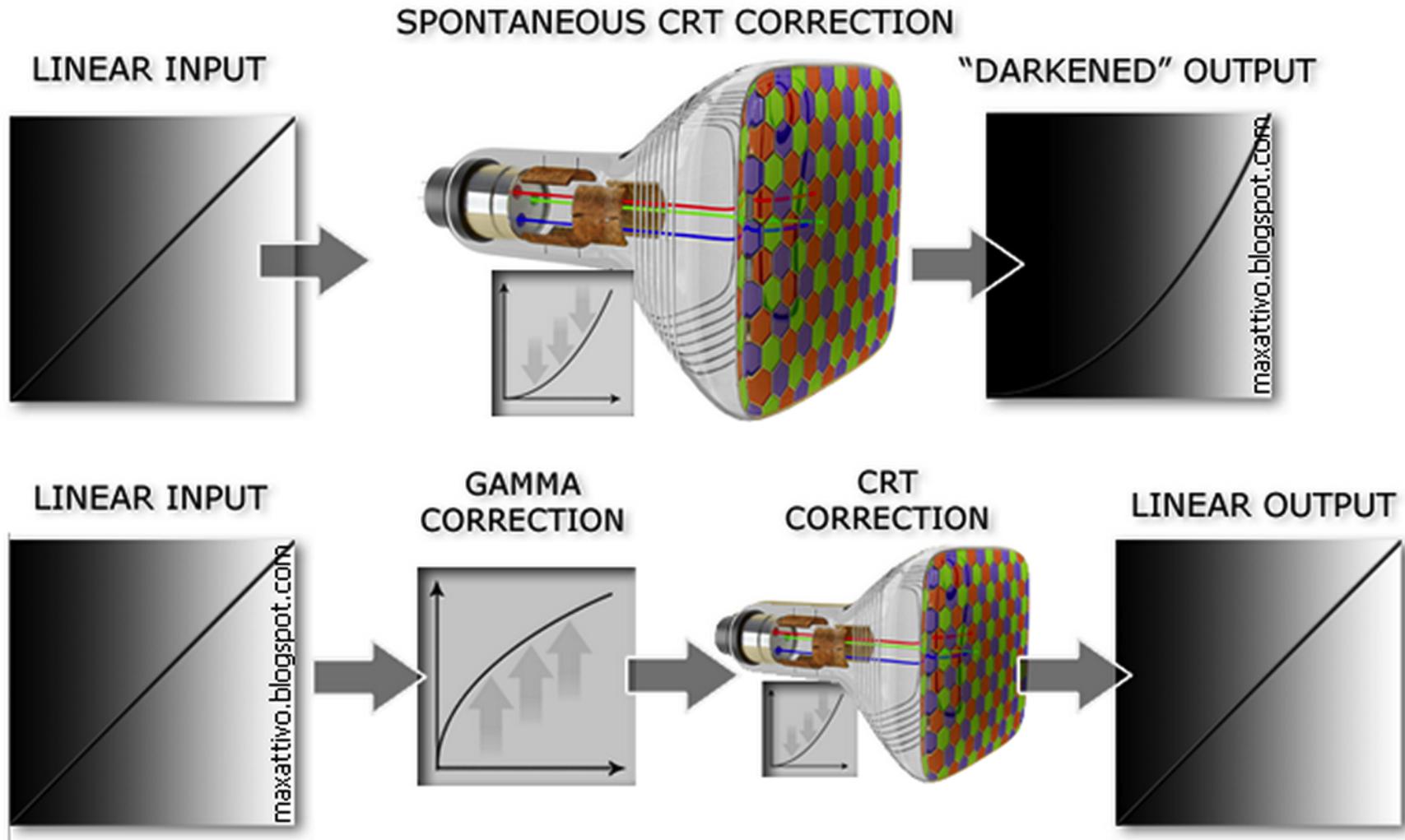


Image: maxattivo.blogspot.ch

Point Operation: Gamma Correction

- The **human light perception** is also **not linear**:

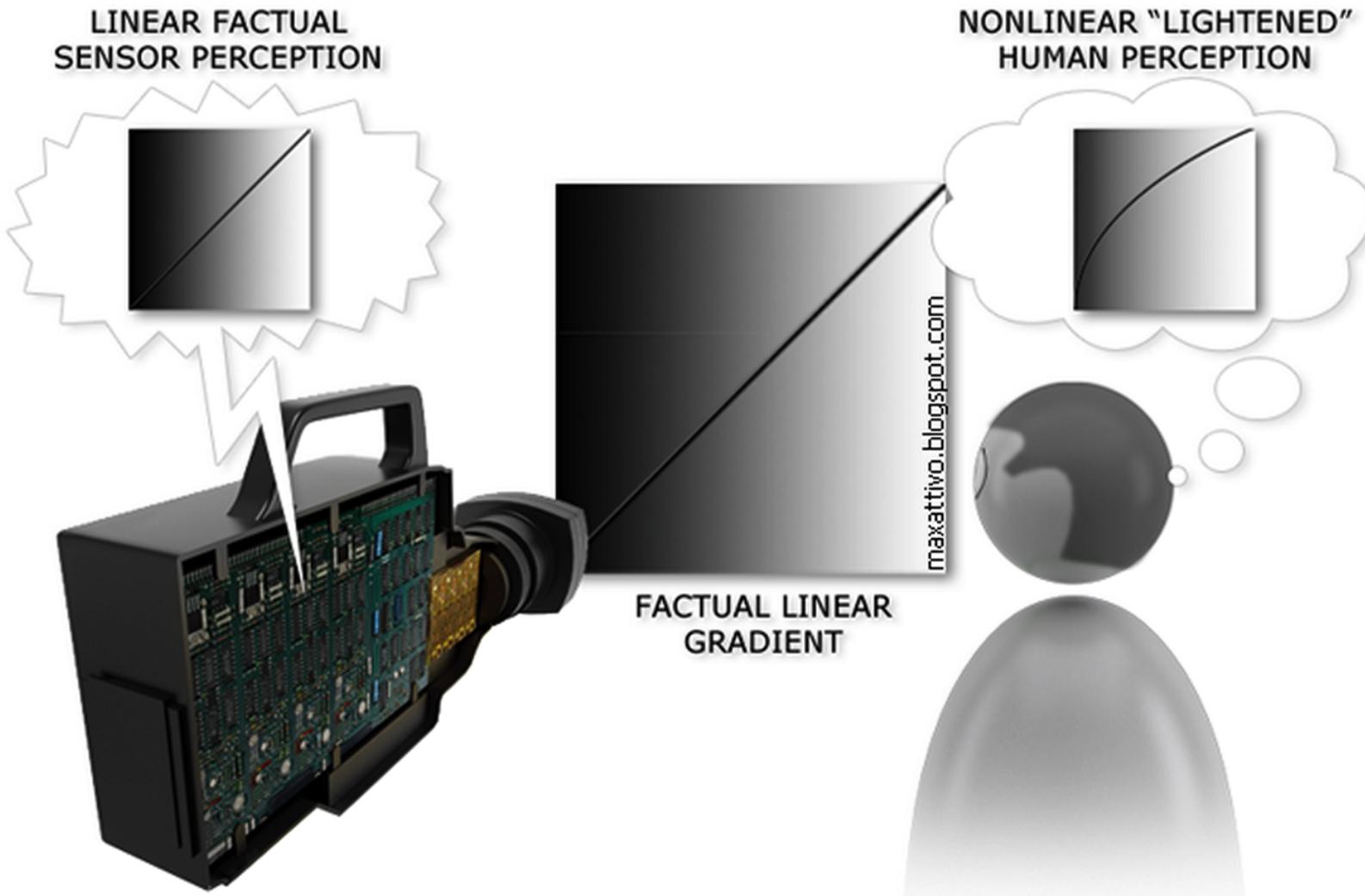


Image: maxattivo.blogspot.ch

Point Operation: Gamma Correction

- Gamma Correction is important in the **digital workflow**:

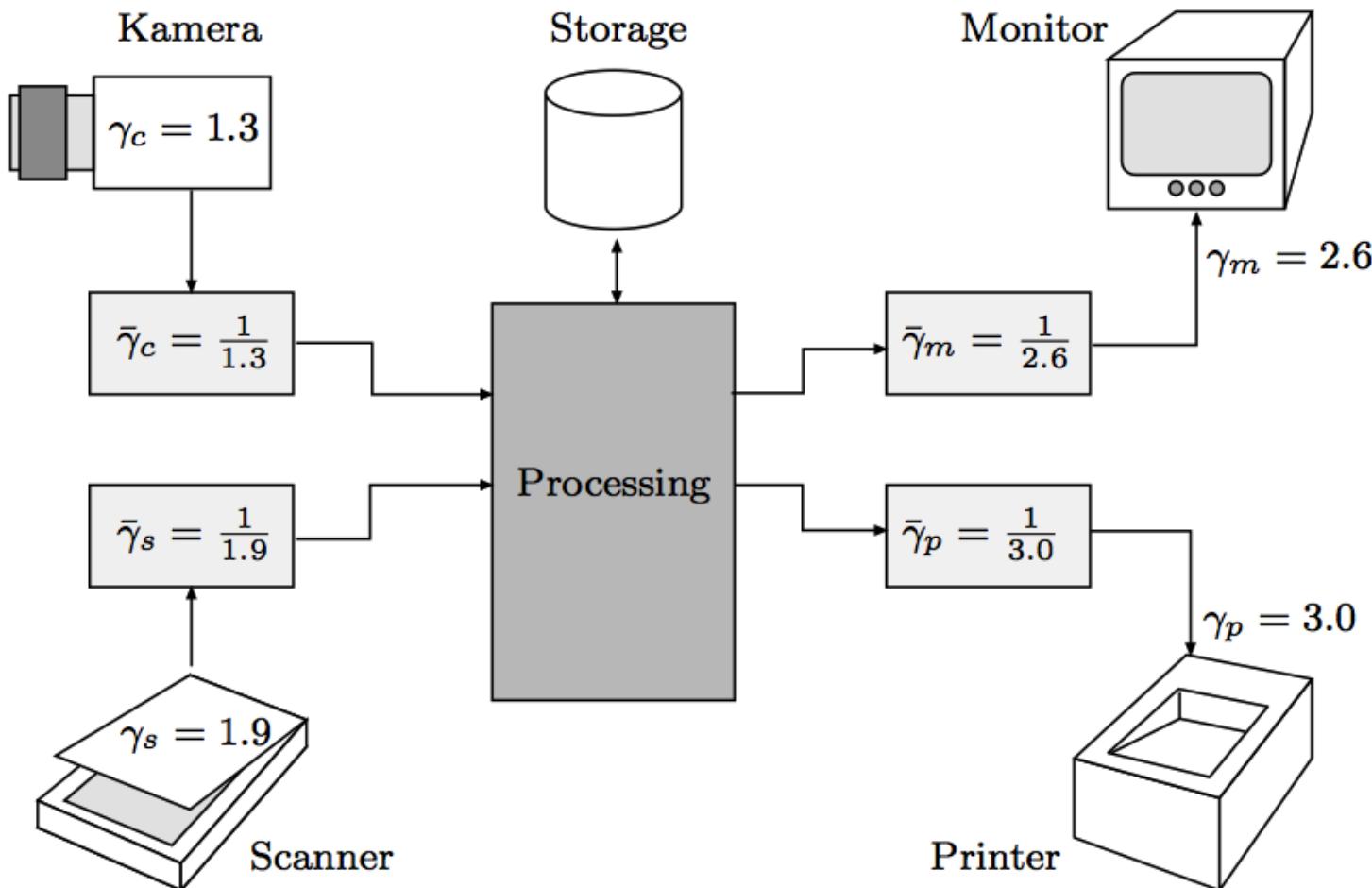


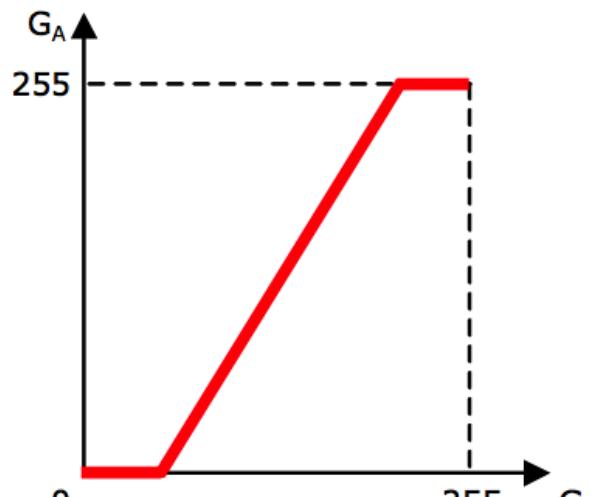
Image: W. Burger: imagingbook.com

Point Operation: Linear Contrast Correction

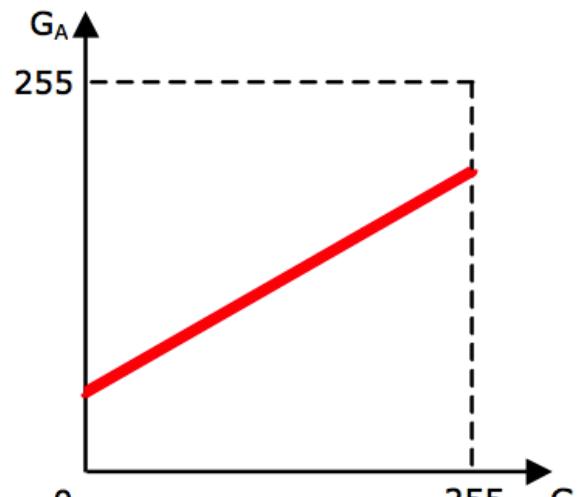
- Linear contrast correction can be expressed in the line equation:

$$g_A(x, y) = c \cdot g_E(x, y) + b$$

where **c changes the contrast** and **b** the brightness.



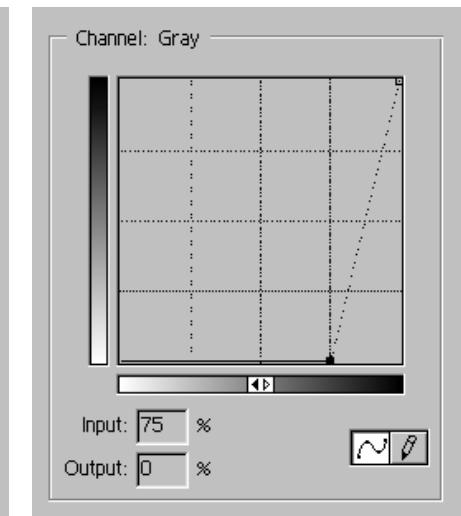
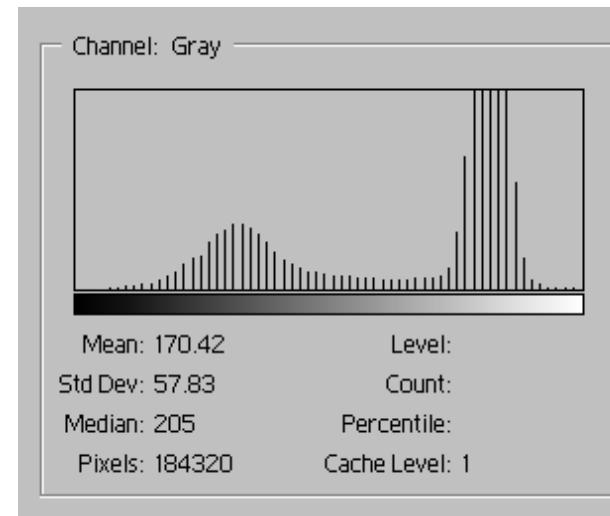
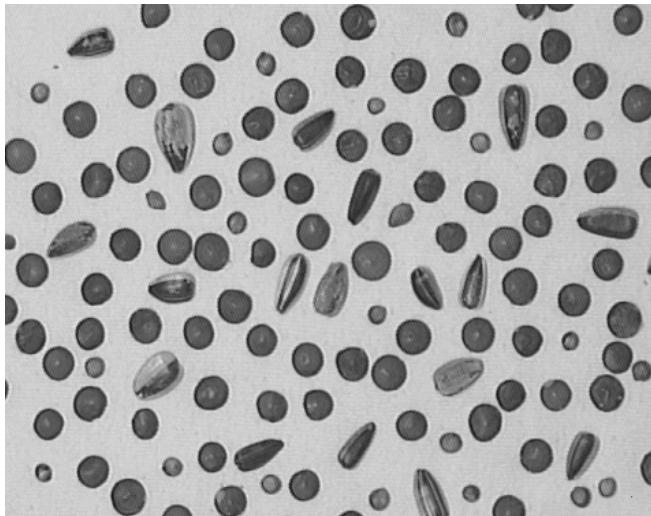
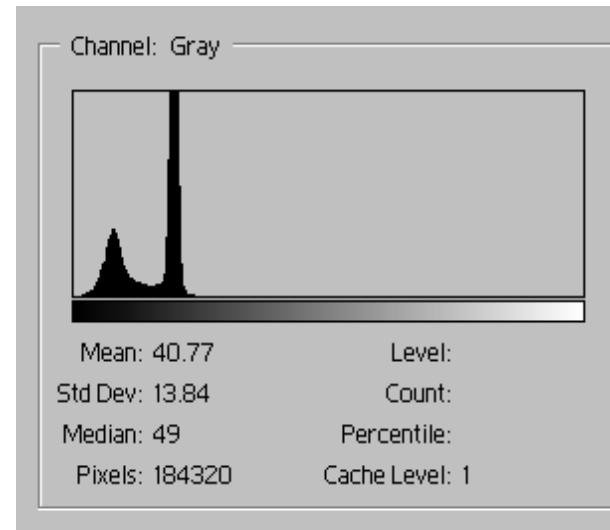
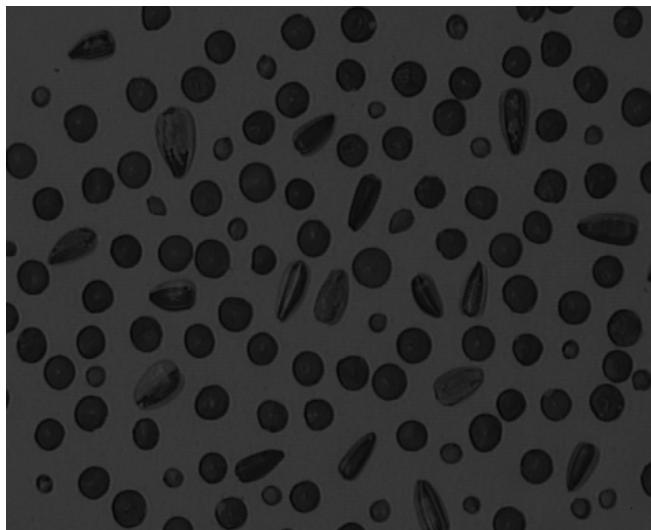
Abbildungsfunktion
Kontrasterhöhung



Abbildungsfunktion
Kontrastverminderung

Point Operation: Linear Contrast Correction

- The histogram is spread:



Point Operation: Linear Contrast Correction

- For **automatic contrast correction** the formula is:

$$g_A(x, y) = (g_E(x, y) - g_{min}) \cdot \frac{255}{g_{max} - g_{min}}$$

- If $g_{min}=0$ and $g_{max}=255$ we cut off a percentage of the brightest and darkest:

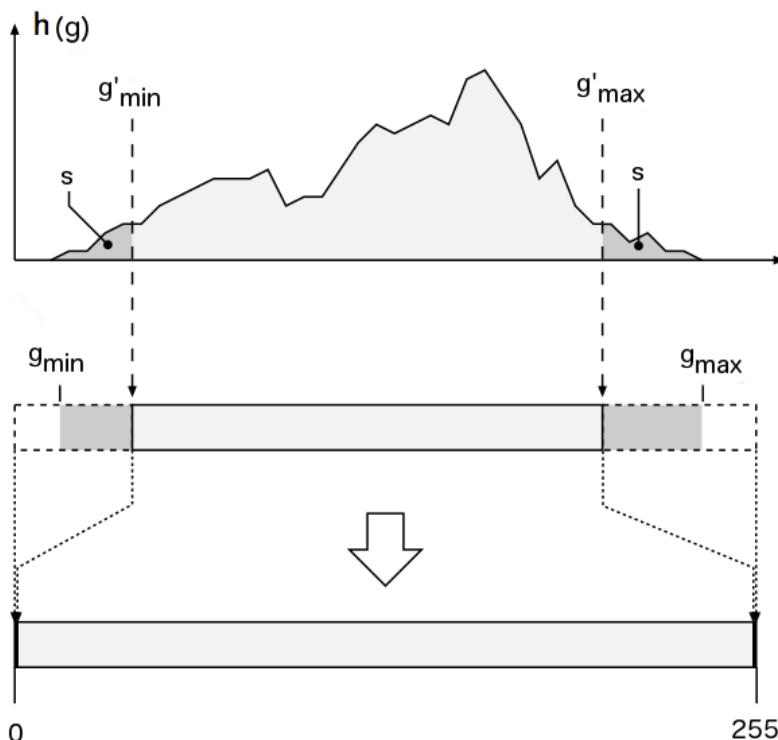


Image: W. Burger: imagingbook.com

Point Operation: Histogram Equalization

- Another automatic contrast correction is the **equalization of the histogram**.
- The algorithm uses the **cummulative histogram**:

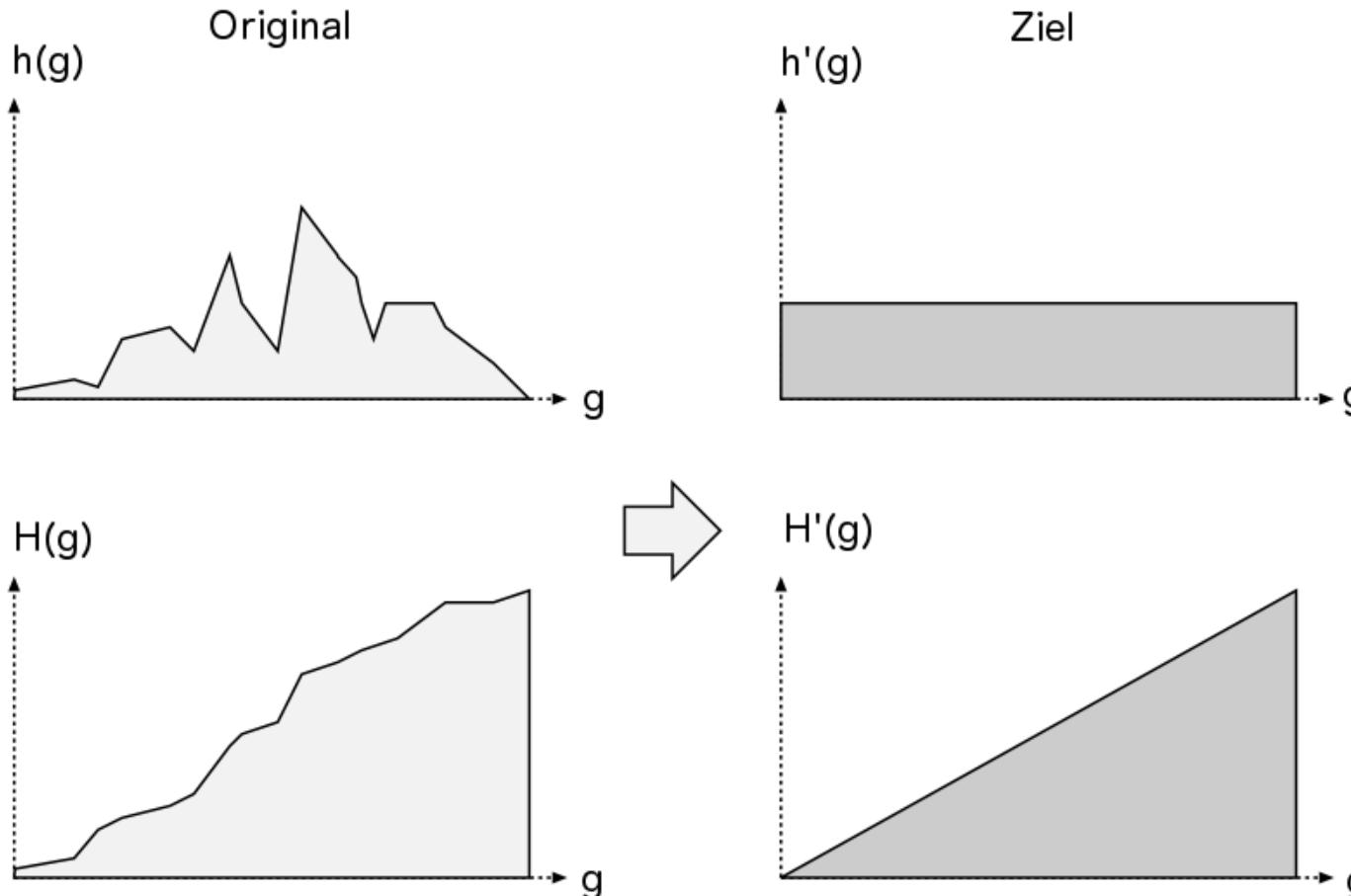
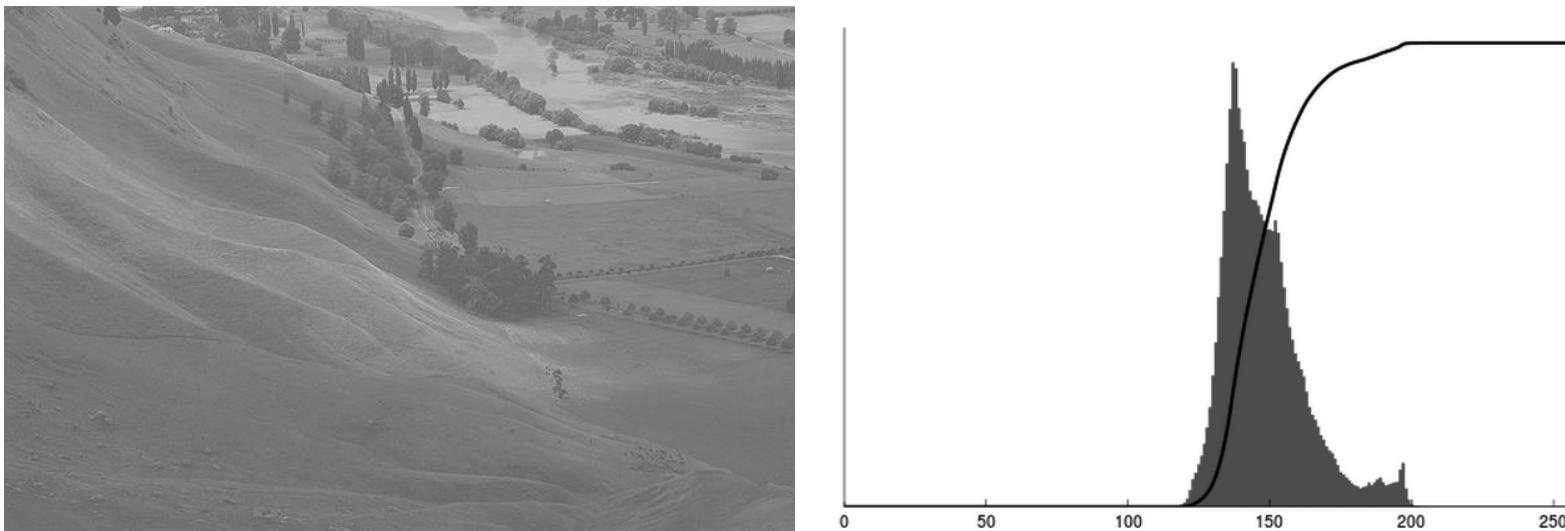


Image: W. Burger: imagingbook.com

Point Operation: Histogram Equalization

- We can use the cumulative histogram as a LUT.
- Gray levels with lots of pixels get spread apart.

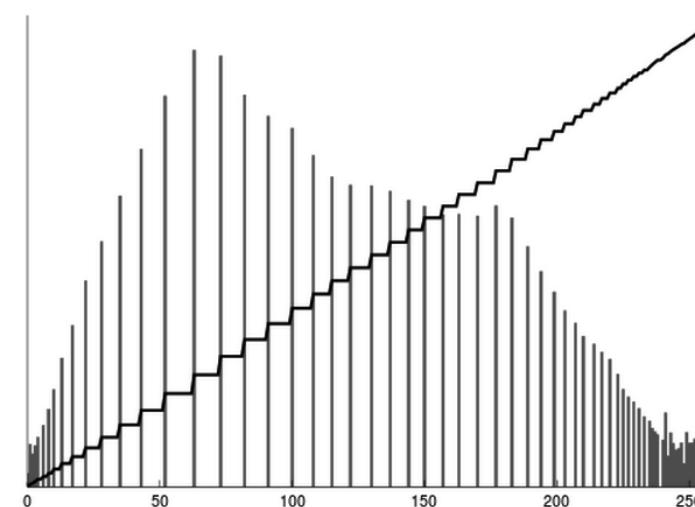
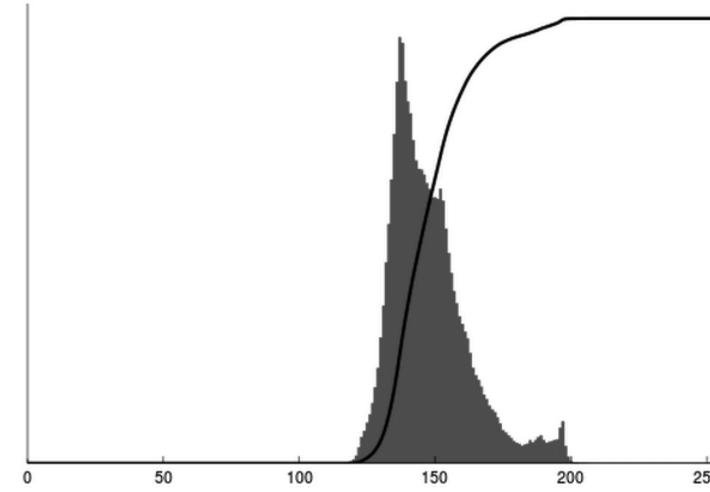


- The scaled cumulative histogram is used as mapping function:

$$f_{eq}(g) = H(g) \cdot \frac{(K-1)}{N}$$

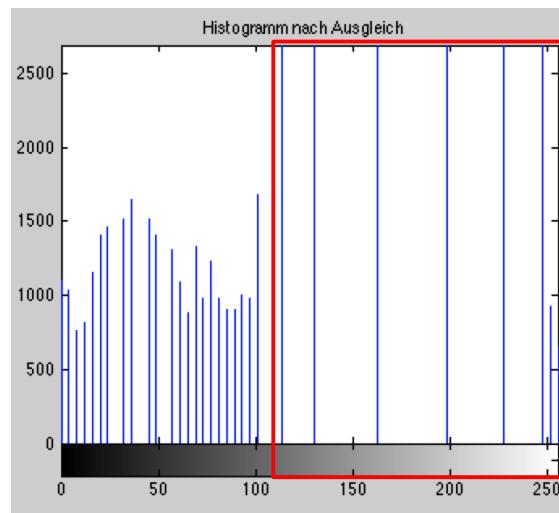
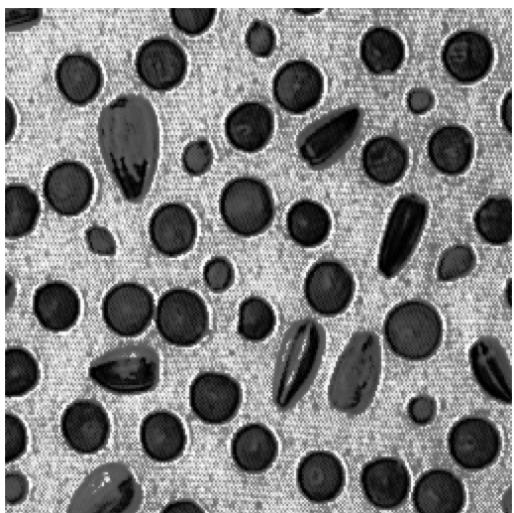
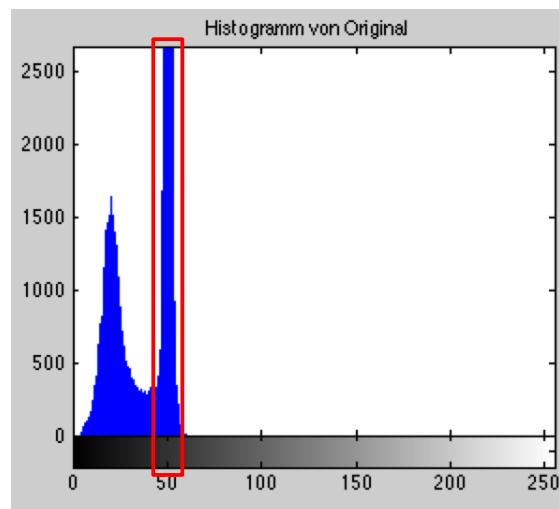
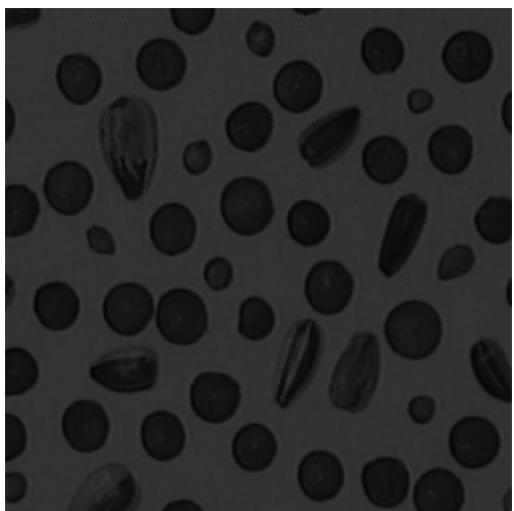
Point Operation: Histogram Equalization

- After the histogram equalization we have equal or less gray levels.
- We often loose information even that the contrast is strong.



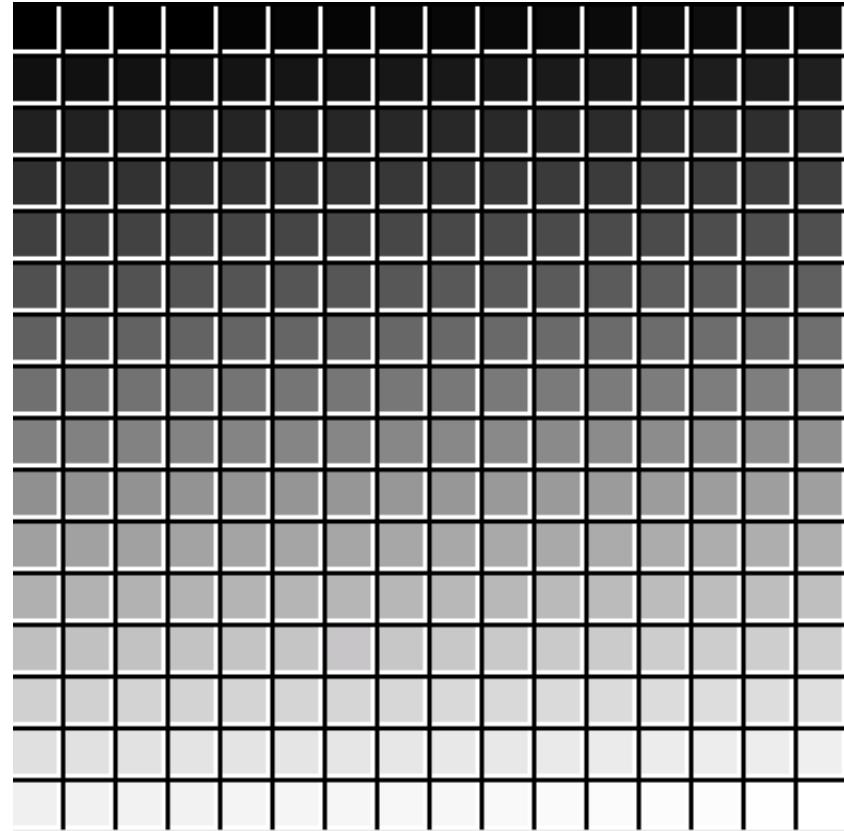
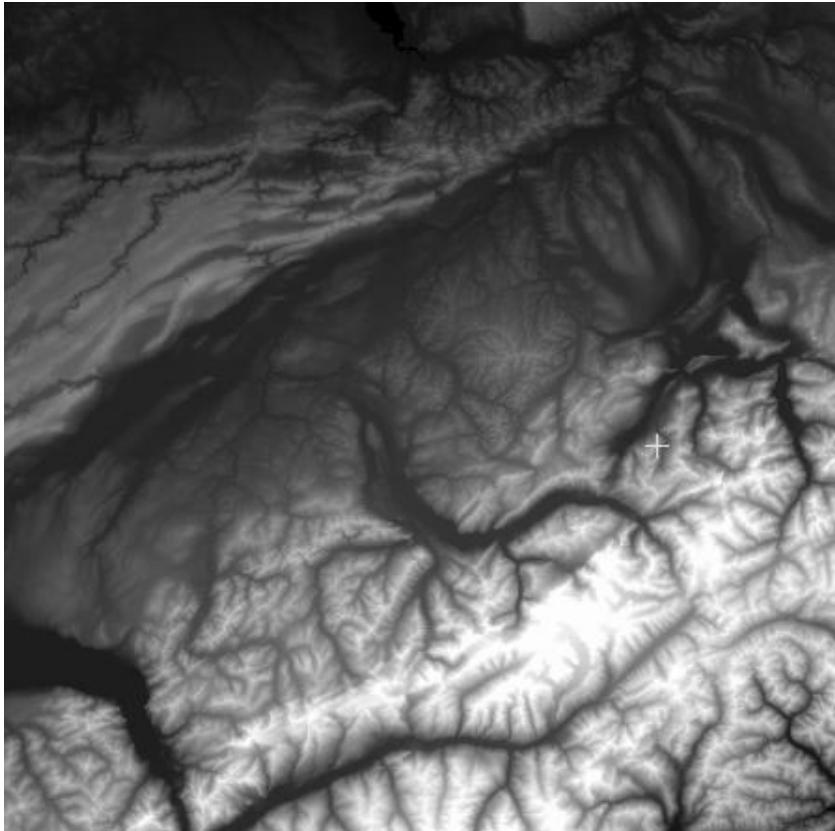
Point Operation: Histogram Equalization

- Histogram Equalization can lead to noise in homogenous regions.
- A few gray levels with lots of pixels get torn apart:



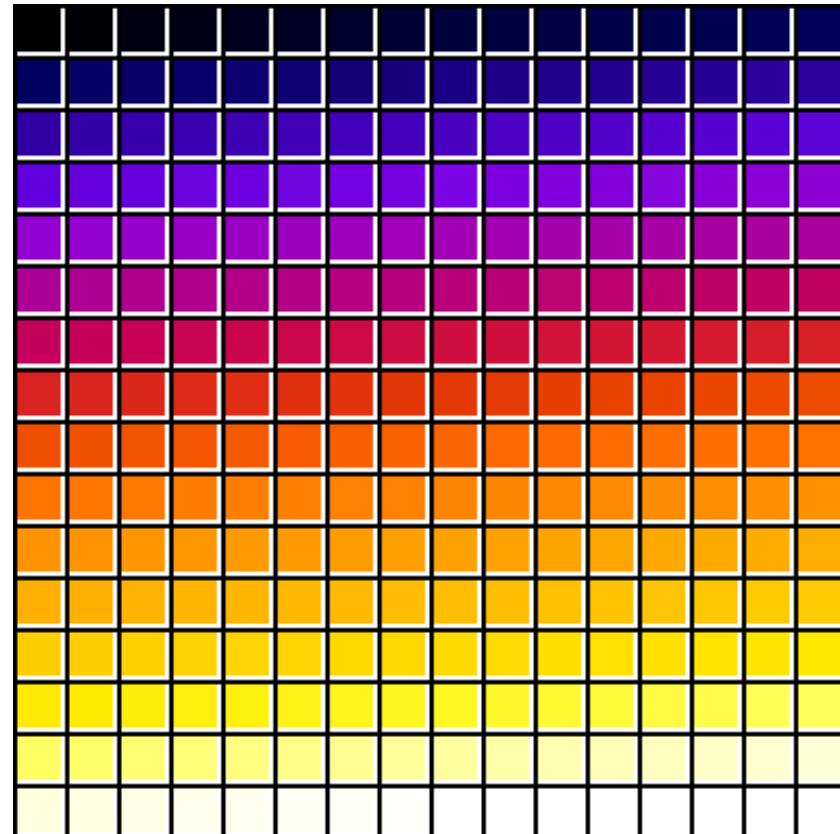
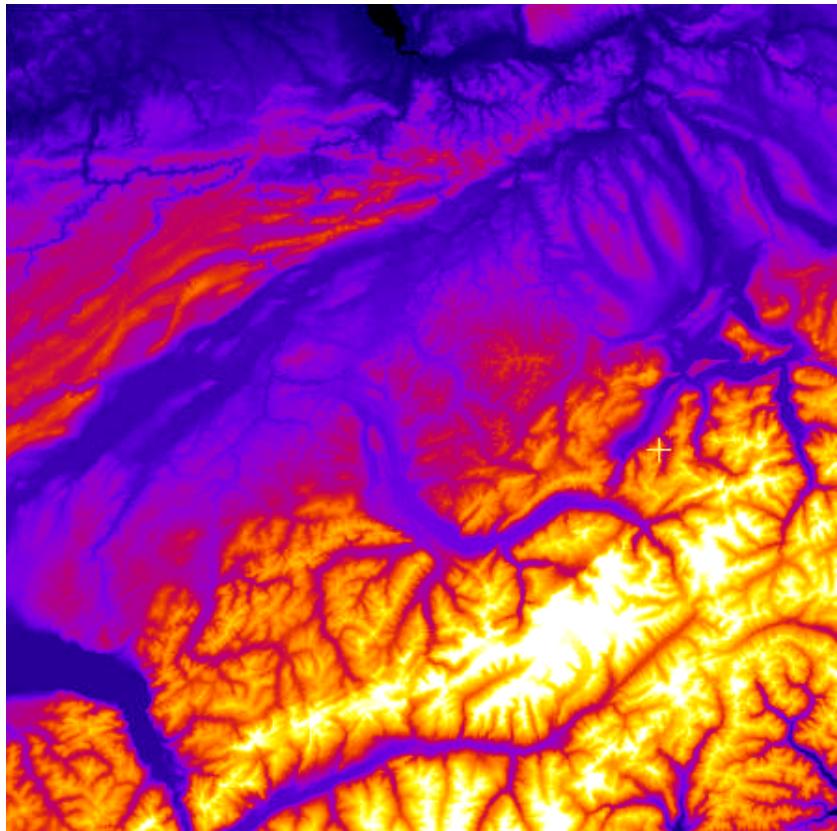
Point Operation: Increase Contrast with Colors

- The human eye can **distinct only 60 gray levels**.
- With **color lookup tables** we can visualize **more information**.
- Height map of Switzerland with 256 gray scales:



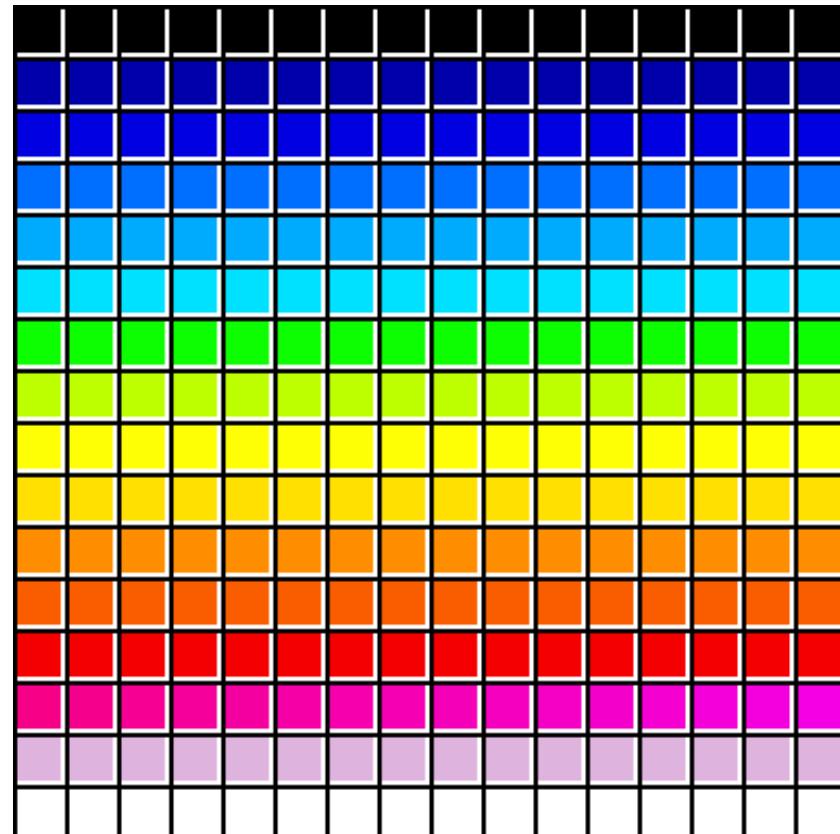
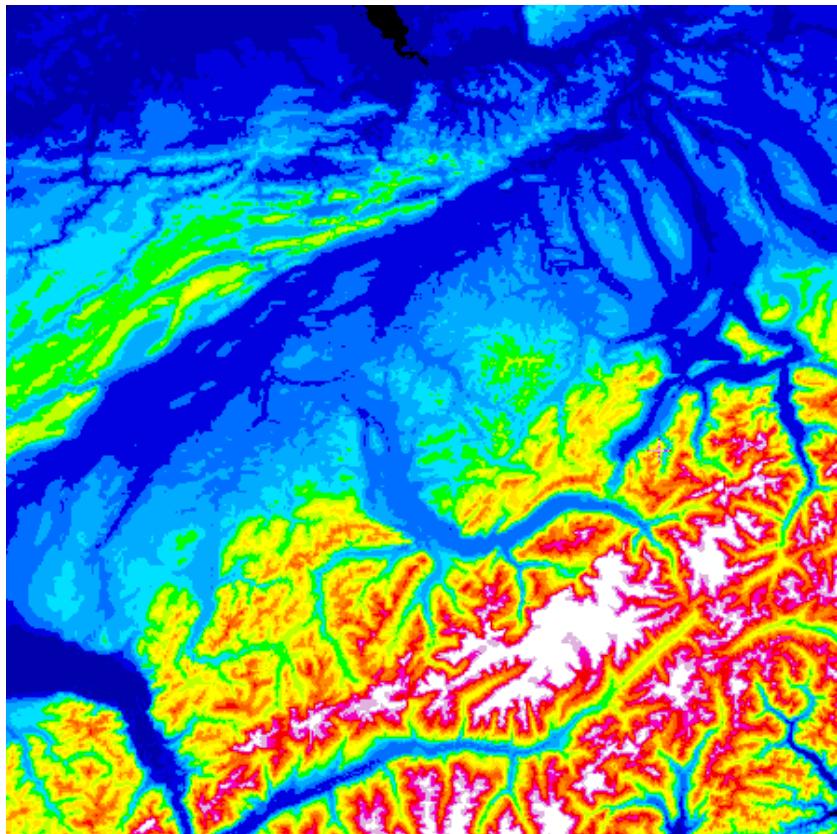
Point Operation: Increase Contrast with Colors

- The human eye can **distinct only 60 gray levels**.
- With **colors** we can visualize **more information**.
- Height map of Switzerland with **ImageJ Fire LUT**:



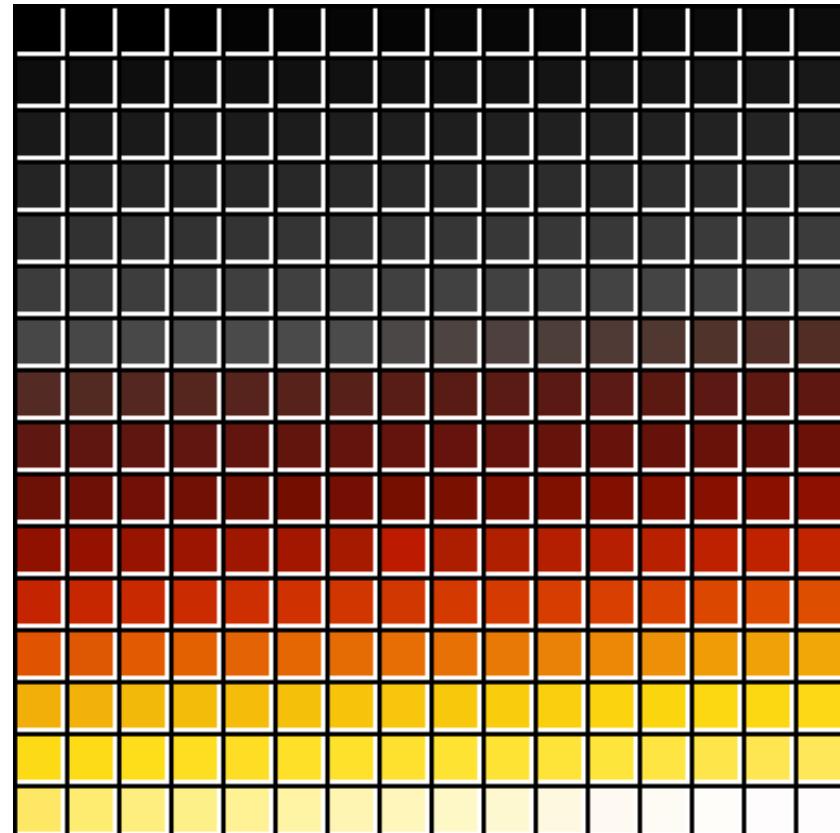
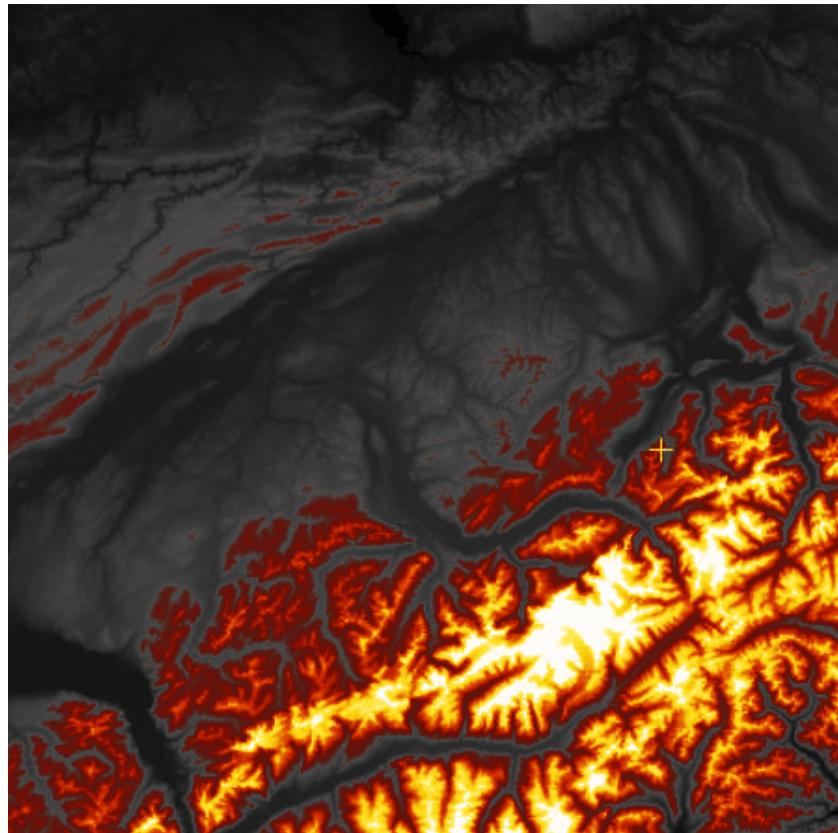
Point Operation: Increase Contrast with Colors

- The human eye can **distinct only 60 gray levels**.
- With **color lookup tables** we can visualize **more information**.
- Height map of Switzerland with **ImageJ 16 Colors LUT**:



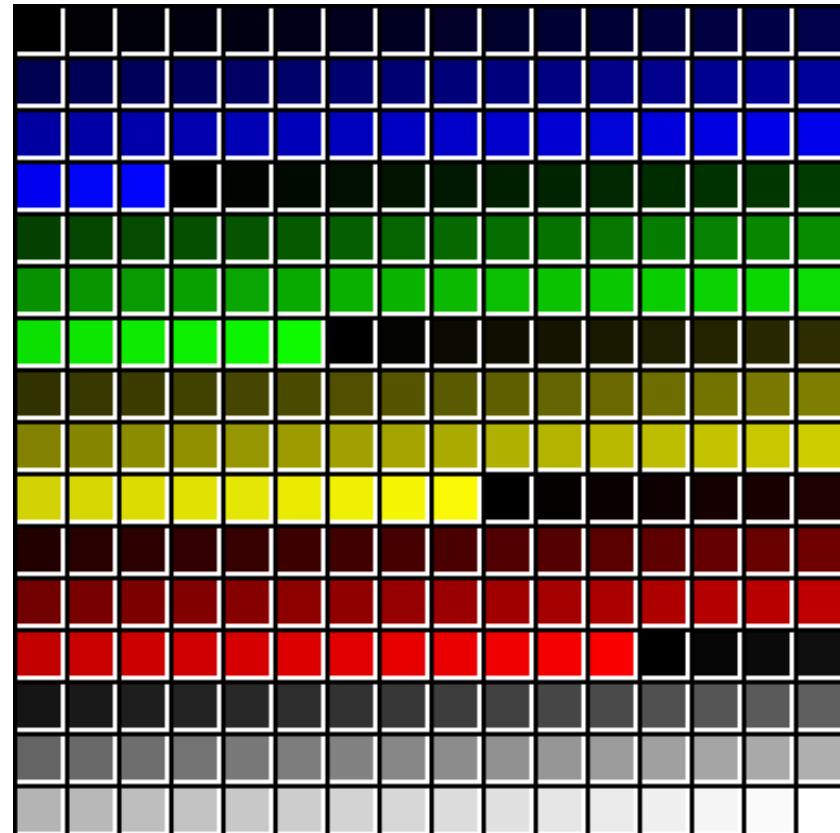
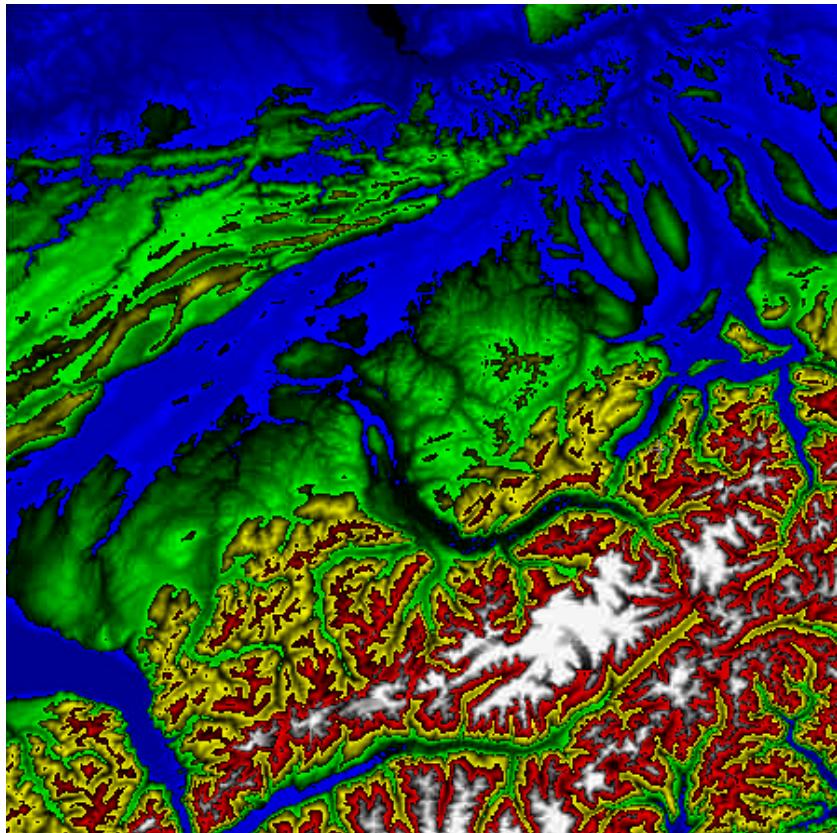
Point Operation: Increase Contrast with Colors

- The human eye can **distinct only 60 gray levels**.
- With **color lookup tables** we can visualize **more information**.
- Height map of Switzerland with **ImageJ Smart LUT**:



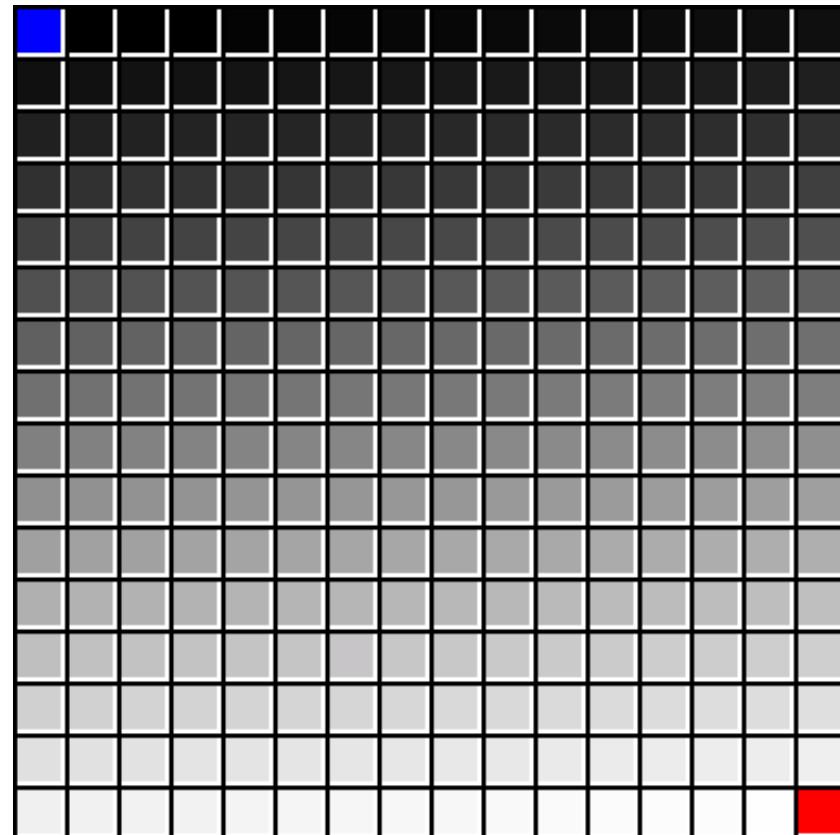
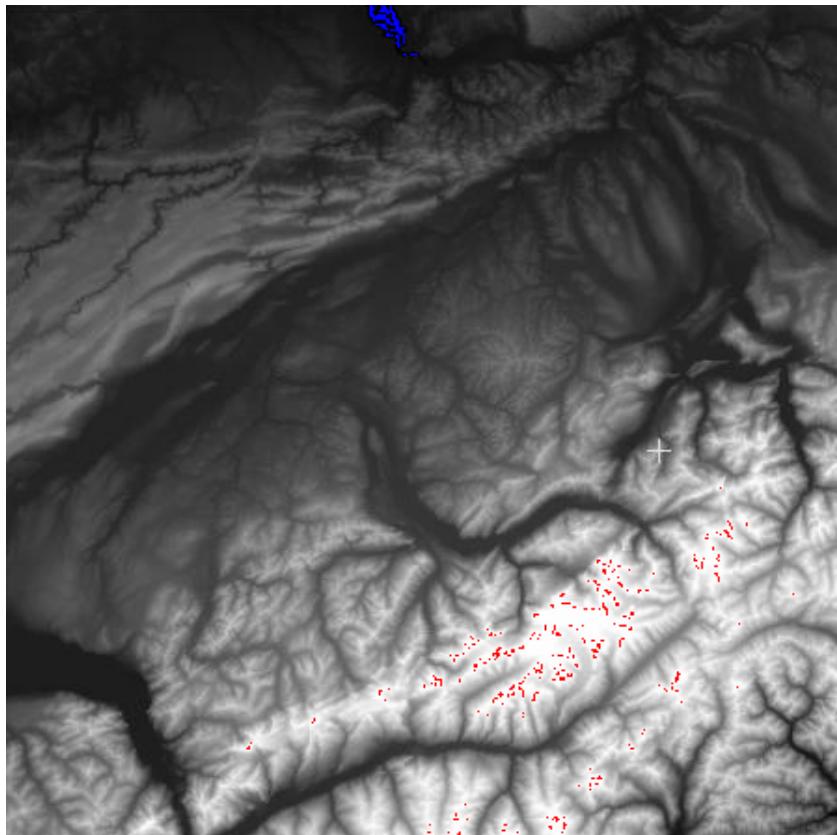
Point Operation: Increase Contrast with Colors

- The human eye can **distinct only 60 gray levels**.
- With **color lookup tables** we can visualize **more information**.
- Height map of Switzerland with **ImageJ 5 Ramps LUT**:



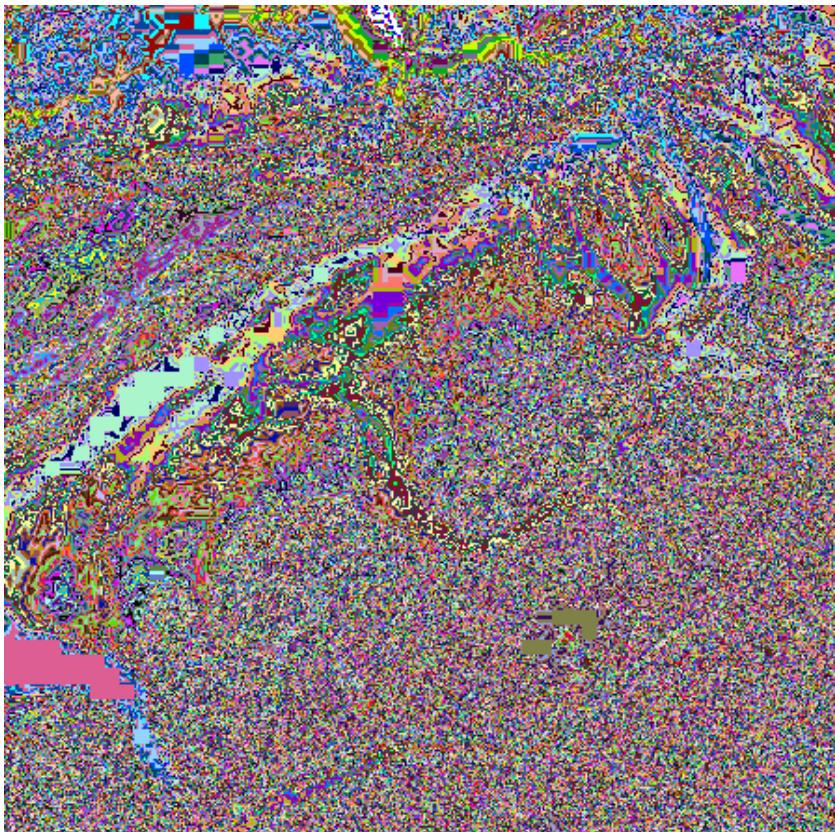
Point Operation: Increase Contrast with Colors

- The human eye can **distinct only 60 gray levels**.
- With **color lookup tables** we can visualize **more information**.
- Height map of Switzerland with **ImageJ HiLo LUT**:



Point Operation: Increase Contrast with Colors

- The human eye can **distinct only 60 gray levels**.
- With **color lookup tables** we can visualize **more information**.
- Height map of Switzerland with **ImageJ HiLo LUT**:



Point Operation: Image Arithmetic

- With brightness & contrast correction we used arithmetics with constant c & b:

$$g_A(x, y) = c \cdot g_E(x, y) + b$$

- We can do also arithmetics with varying values of 2 images.
- With the constants k1 and k2 we take care to be within the range 0-255:

- Addtion:
$$g_A(x, y) = (g_{E1}(x, y) + g_{E2}(x, y)) \cdot k1 + k2$$

- Subtraction:
$$g_A(x, y) = (g_{E1}(x, y) - g_{E2}(x, y)) \cdot k1 + k2$$

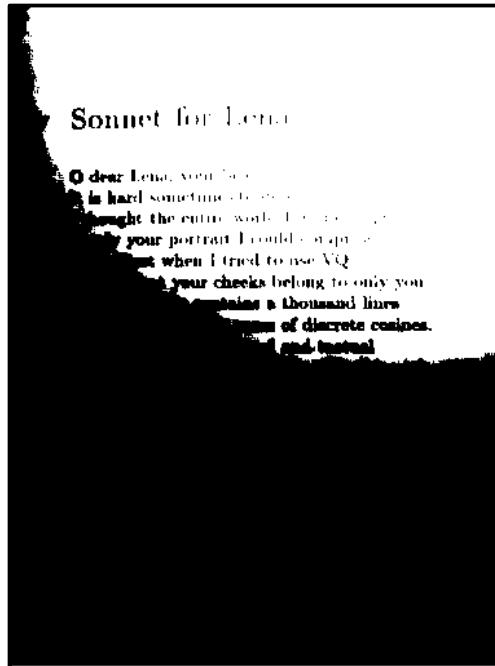
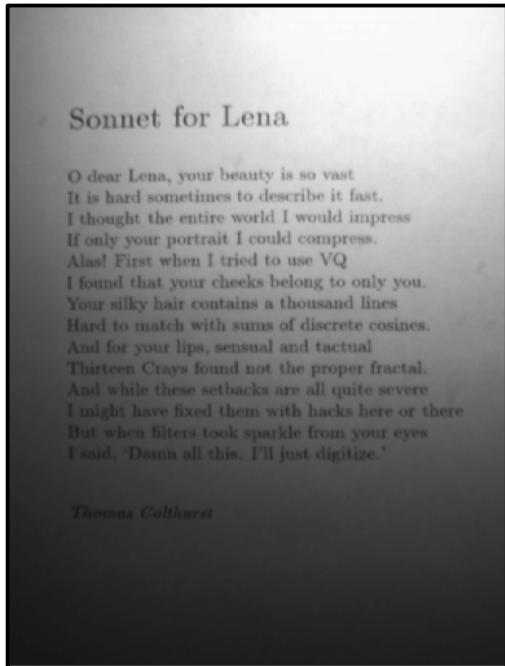
- Multiplication:
$$g_A(x, y) = (g_{E1}(x, y) \cdot g_{E2}(x, y)) \cdot k1 + k2$$

- Division:
$$g_A(x, y) = (g_{E1}(x, y) / g_{E2}(x, y)) \cdot k1 + k2$$

Point Operation: Image Arithmetic

Example: Separate text from background:

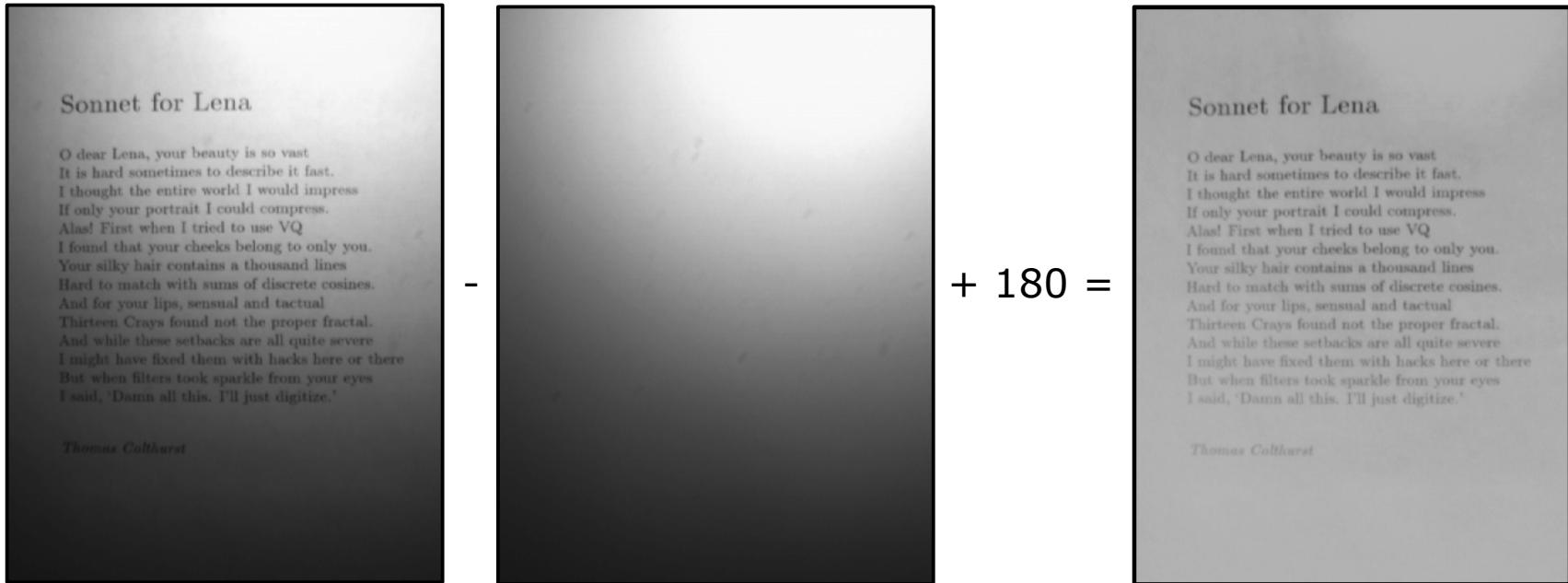
- Not possible with one distinct threshold:



Point Operation: Image Arithmetic

Example: Separate text from background:

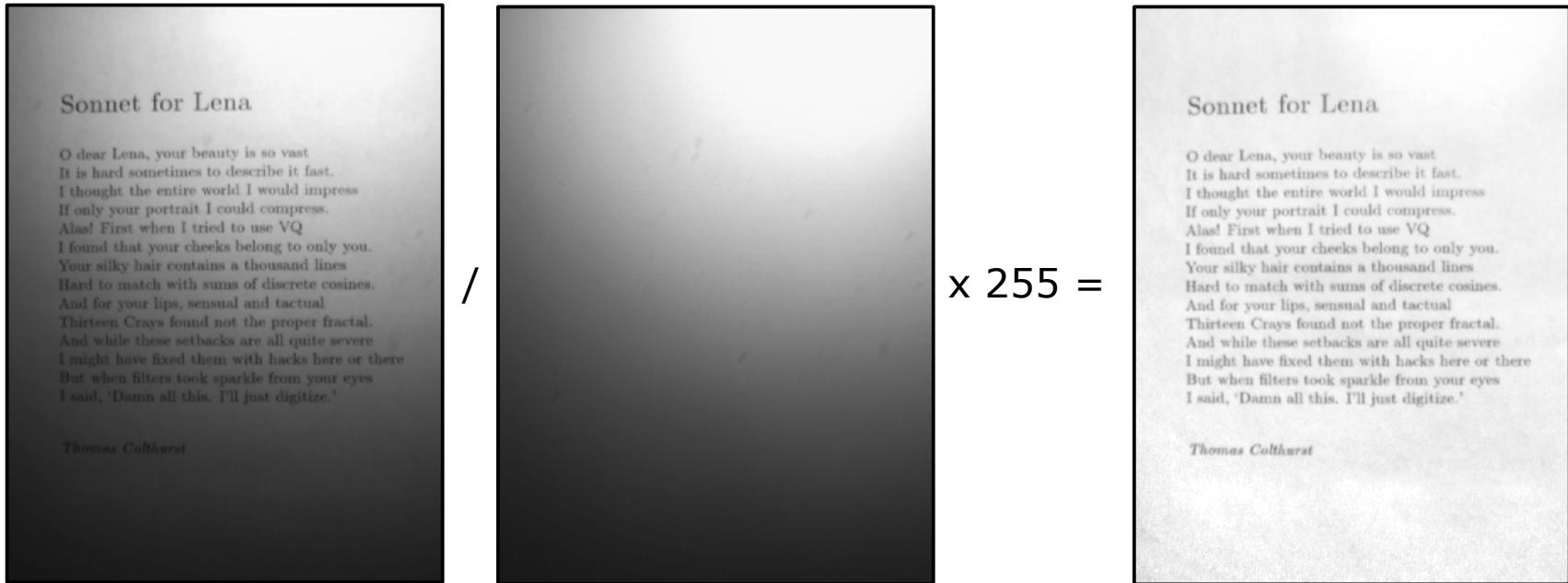
- We can subtract the background:



Point Operation: Image Arithmetic

Example: Separate text from background:

- We can devide by the background:



Remains the question how to get the background?

- With a median filter (denoising filter, chapter 6)
- and/or with a gaussian (averaging filter, chapter 6)

Point Operation: Image Arithmetic

Example: Detecting shifted object with absolute difference:

$$g_A(x, y) = |g_{E1}(x, y) - g_{E2}(x, y)|$$



Point Operation: Logic Operations

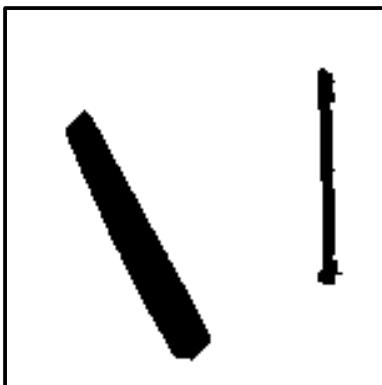
- Boolean operations like AND, OR and XOR can be used to combine image:



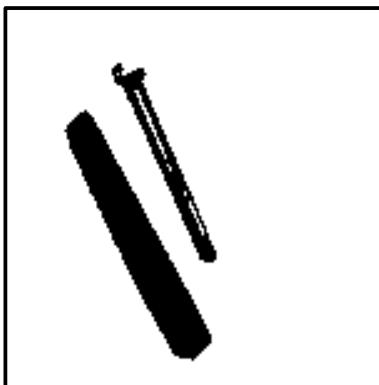
AND



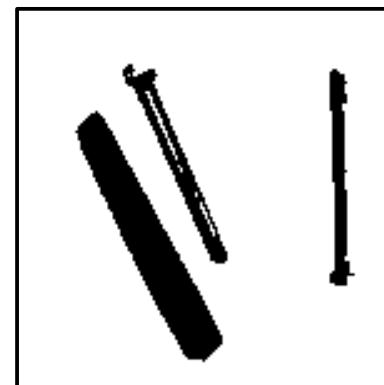
=



AND



=



Point Operation: Logic Operations

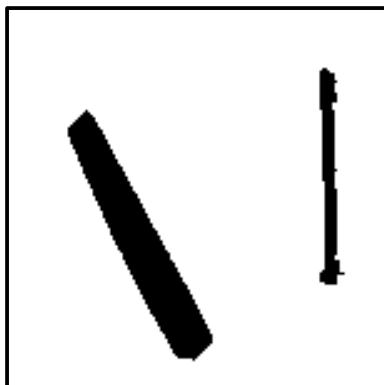
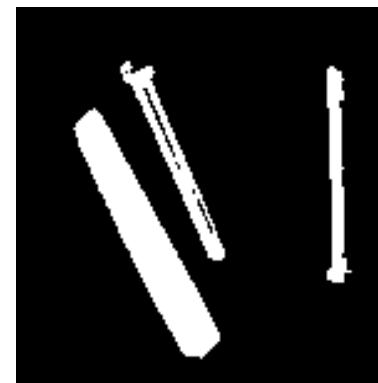
- Boolean operations like AND, OR and XOR can be used to combine image:



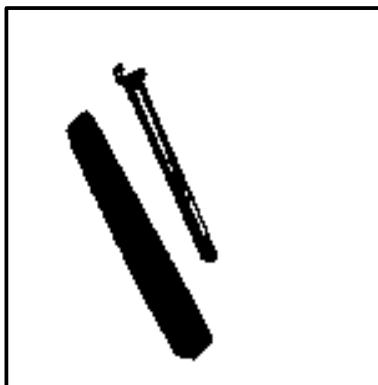
OR



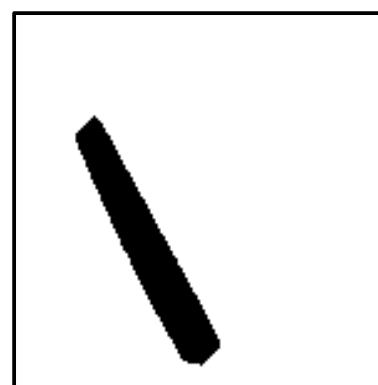
=



OR



=

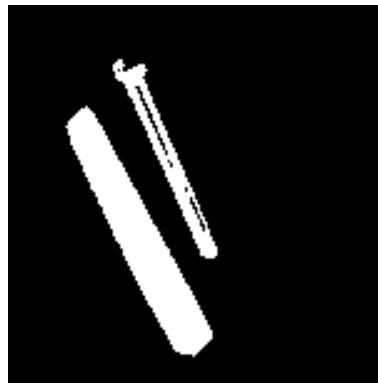


Point Operation: Logic Operations

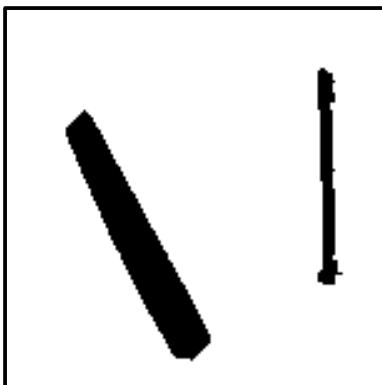
- Boolean operations like AND, OR and XOR can be used to combine image:



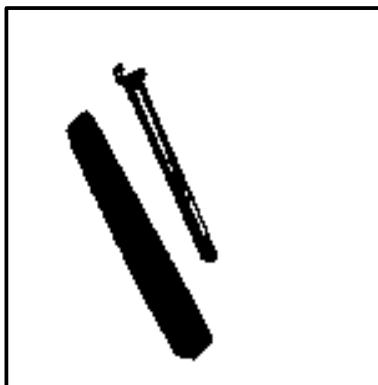
XOR



=



XOR



=

