Lecture

Image Histograms

Cyrill Stachniss

Summer term 2024 – Cyrill Stachniss

5 Minute Preparation for Today

https://www.youtube.com/watch?v=flI_Umo_VAU

Photogrammetry & Robotics Lab

Image Histograms and Simple Point Operators

Cyrill Stachniss

The slides have been created by Cyrill Stachniss.

Variations of an Image

Can We Describe the Color Distribution of an Image and Modifications to it?

Grayscale Image

- 2D grid of intensity measurements
- Can be seen as a I times J matrix with elements taking values in [0…255]

0	255	0
128	0	0
255	0	128

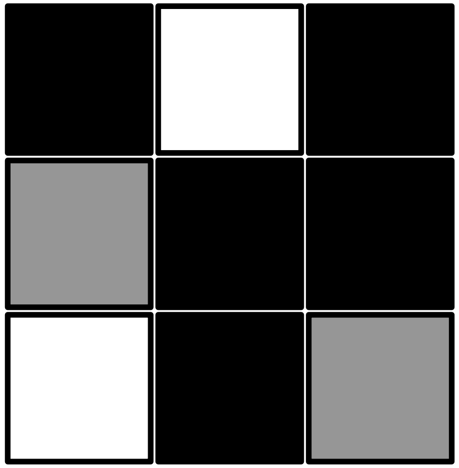


Image as a Function

- Image intensities can be expressed as a function

$$g(i,j) : \mathcal{B} \mapsto \mathcal{G}$$

- with $\mathcal{B} = \mathbb{Z} \times \mathbb{Z}$ and $\mathcal{G} = \mathbb{N}$
- Often, we have

$$\mathcal{B} = [0 \ldots I-1, 0 \ldots J-1]$$

$$\mathcal{G} = [0 \ldots 255]$$

Histogram

Approximate representation of the distribution of numerical data

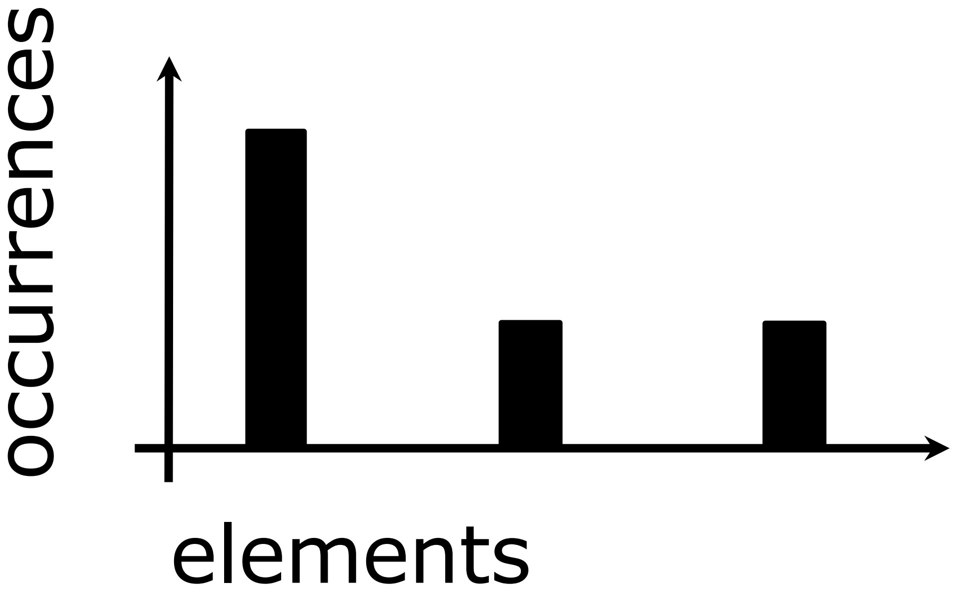


Image Histogram

Representation of the distribution of intensity values of the image

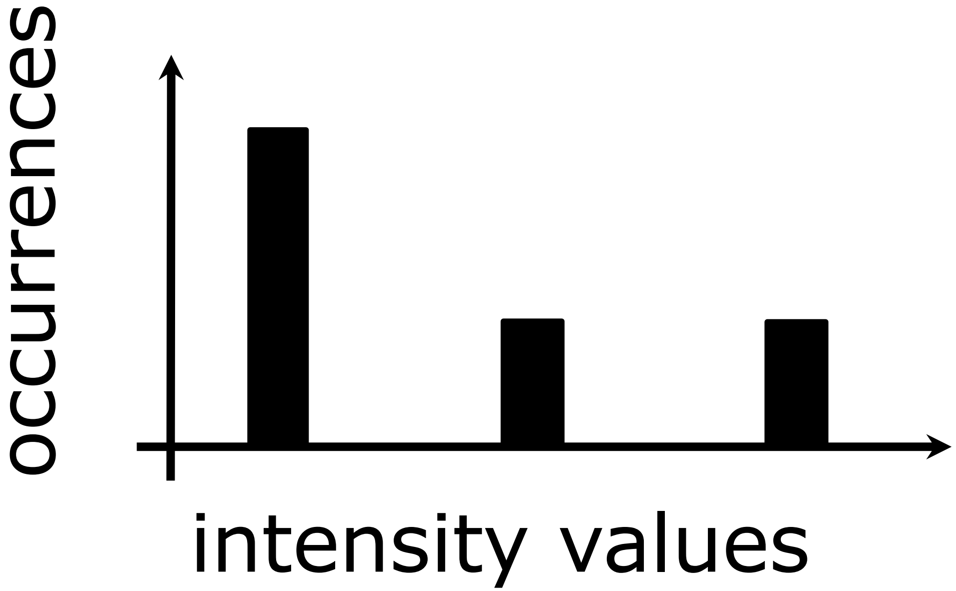


Image Histogram

- A histogram is a frequently used tool for analyzing images
- Def.

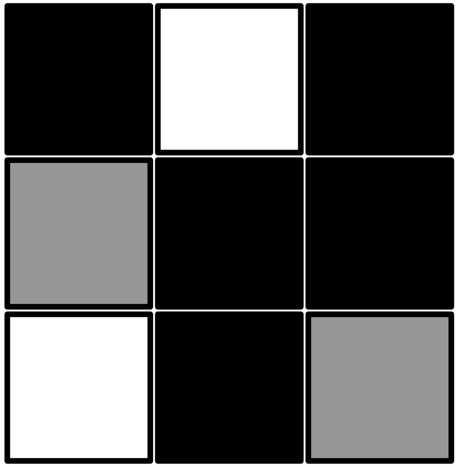
$$h(g) = \#\text{pixel with value } g$$

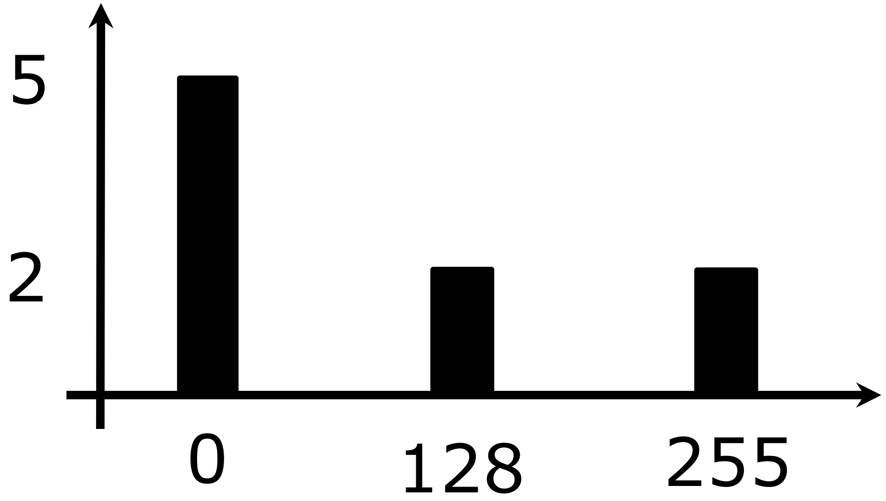
- Counts the number of pixels in the image that take a certain value
- Probability that a random pixel has value g

$$p(g) = \frac{h(g)}{N} \longleftarrow N = I \times J$$

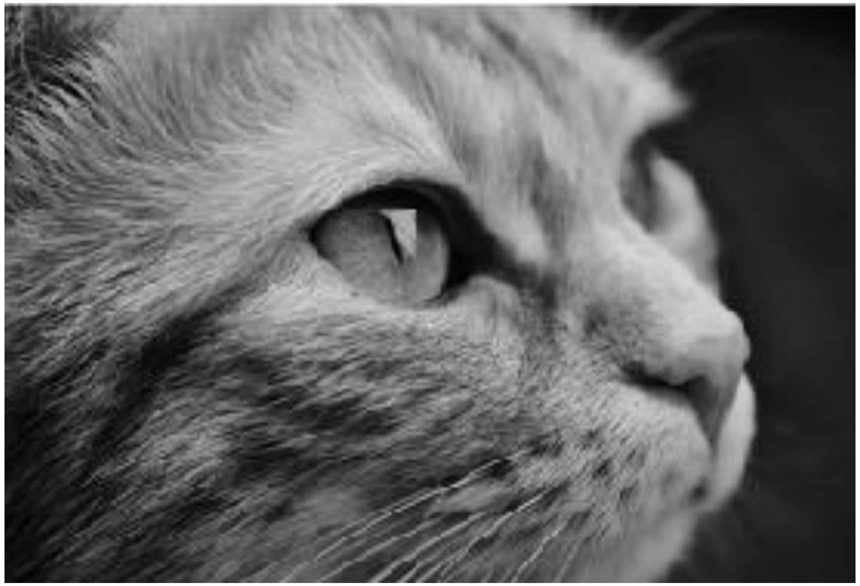
Image to Histogram

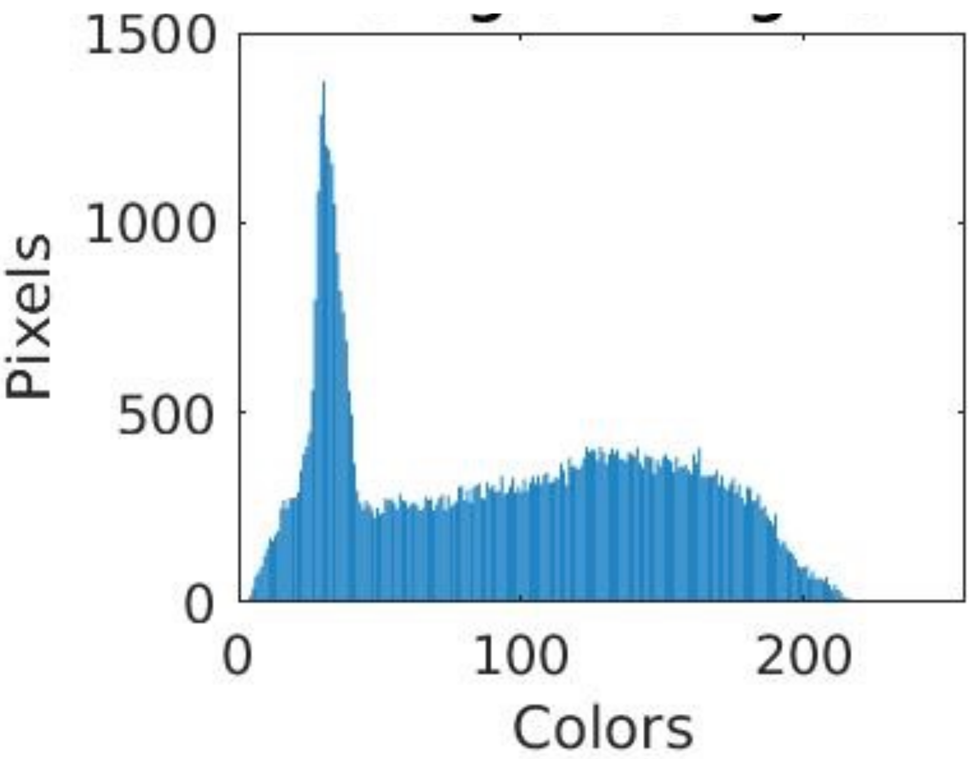
0	255	0
128	0	0
255	0	128





Example





Histogram Computation

A histogram can be computed by analyzing all pixel values

Algorithm

- Create vector h with length 256
- Initialize h with zeros
- For each pixel (i, j) do $h(g(i, j)) + +$

Histogram Computation

A histogram can be computed by analyzing all pixel values

Algorithm

- Create vector h with length 256
- Initialize h with zeros
- For each pixel (i,j) do $h(g(i,j))++$

➡ The computational complexity is linear in the number of pixels: $\mathcal{O}(N)$

Cumulative Histogram

- Definition

$$H(g) = \sum_{x=0}^{g} h(x) = \#\text{pixel with } g(i,j) \leq g$$

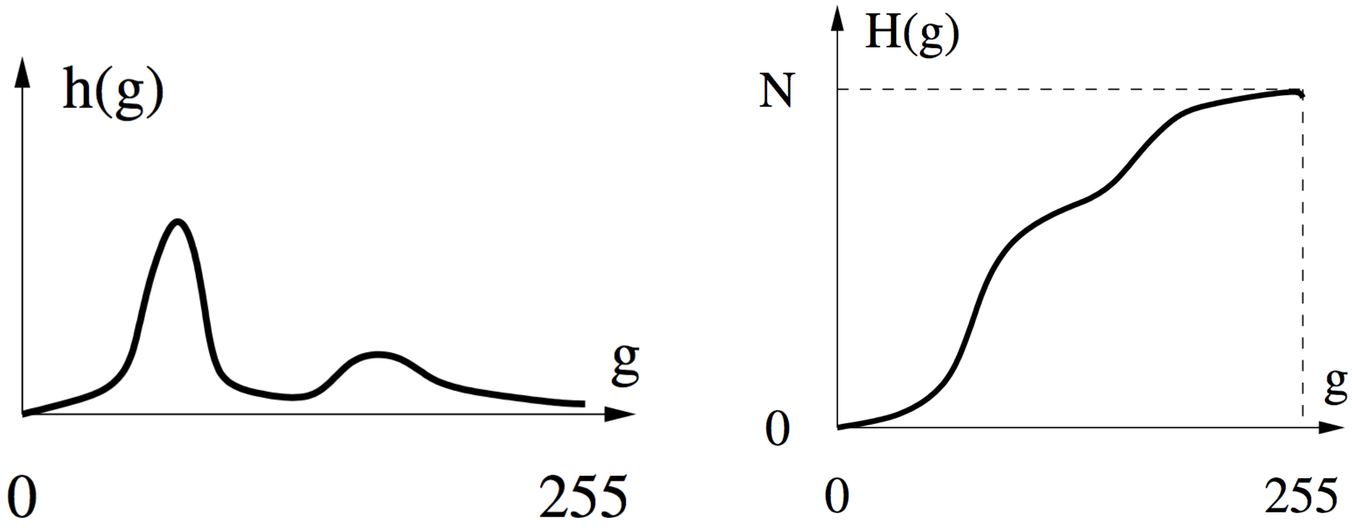
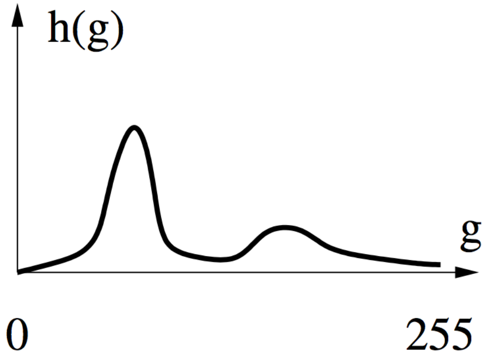


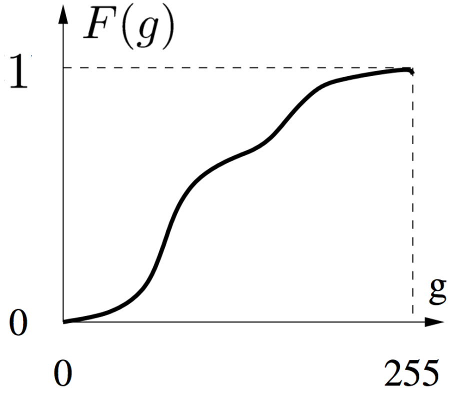
Image courtesy: Förstner

Empirical PDF and CDF

$$p(g) = \frac{h(g)}{N}$$

$$F(g) = P(g(i,j) \leq g) = \frac{H(g)}{N}$$





- $p(g)$ is the empirical probability density function of intensity values
- $F(g)$ is the empirical cumulative distribution function of intensity values

Image Histograms Tell Us Something About the Image

Examples

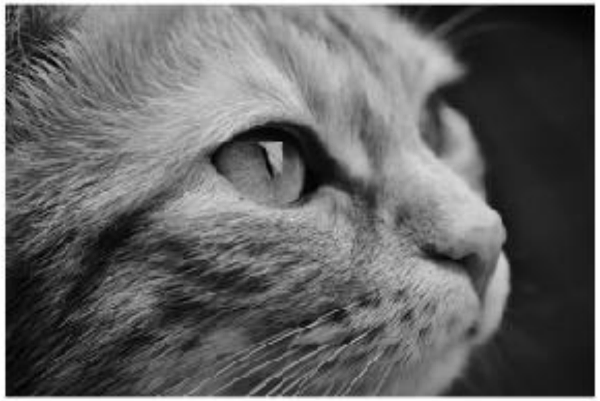


Image original

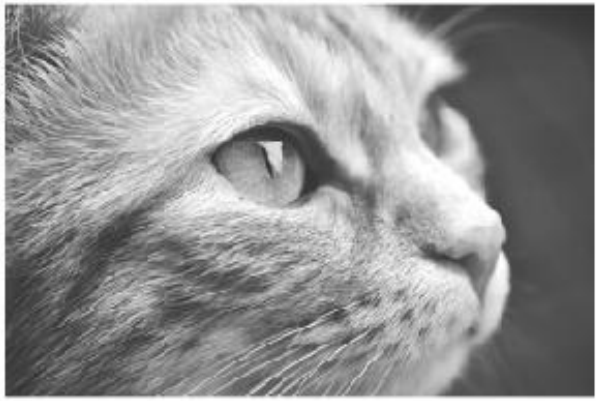
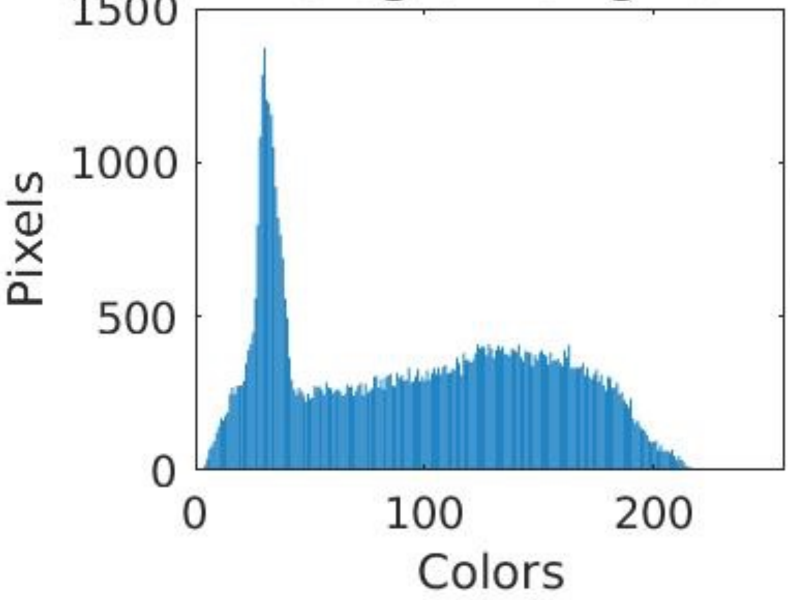
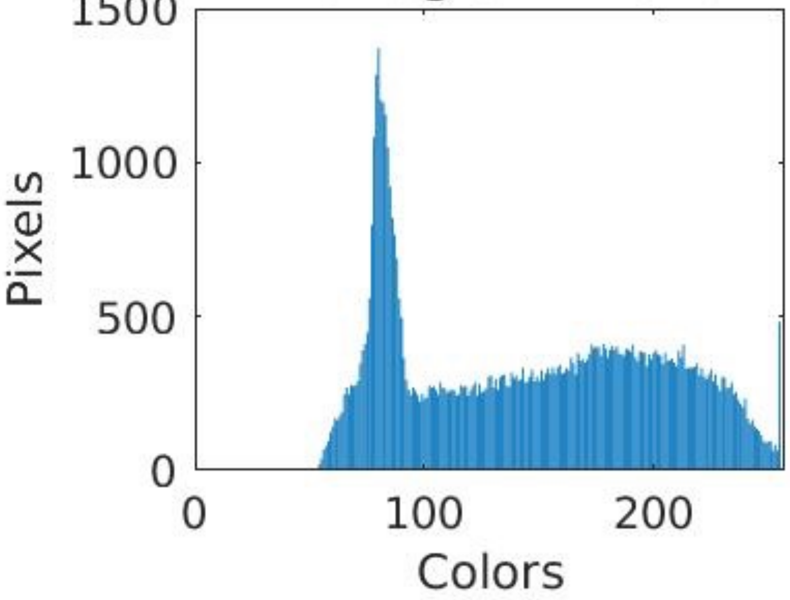


Image transformed



Histogram original



Histrogram Trans

Examples

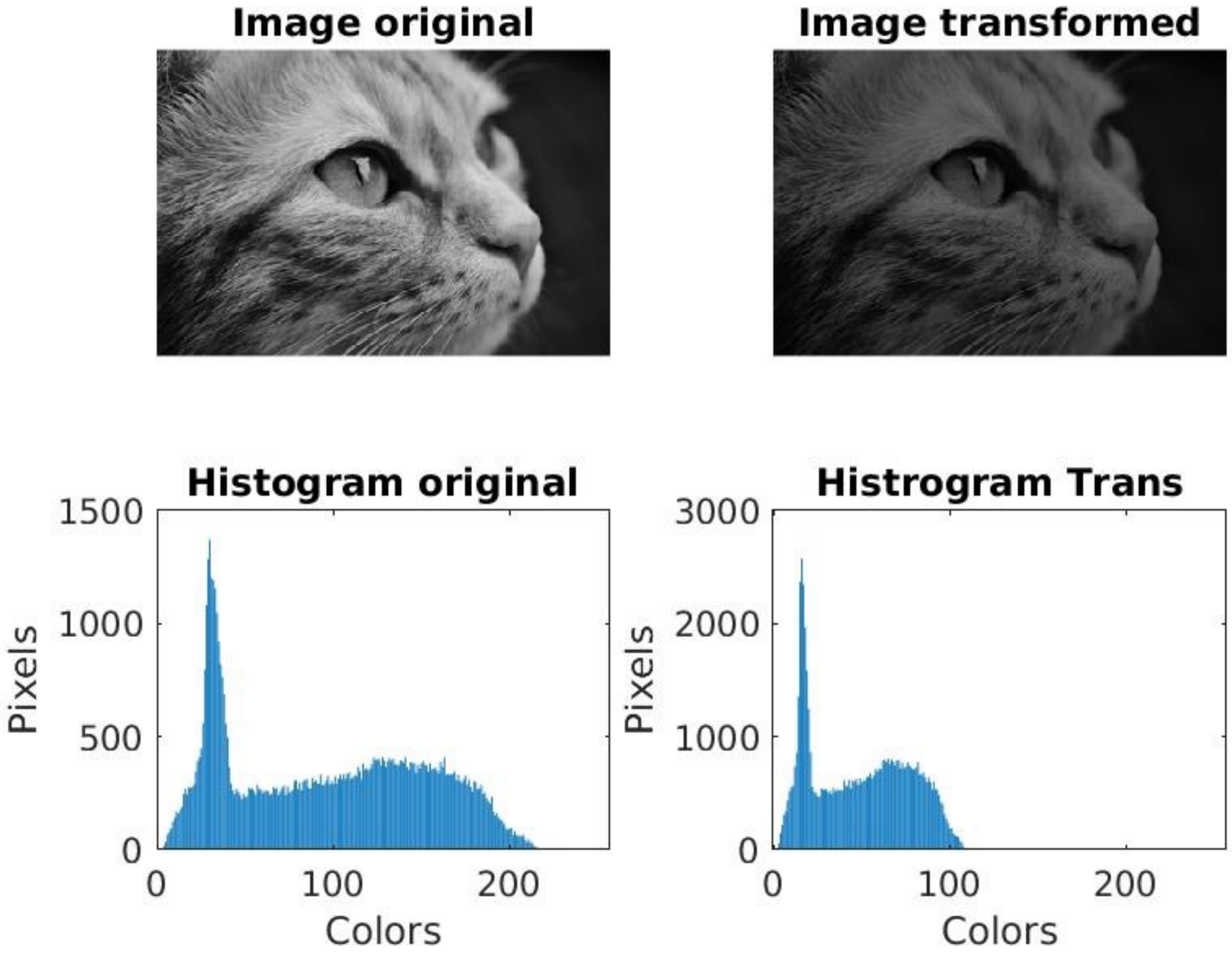


Image original
Image transformed
Histogram original
Histrogram Trans
Pixels
1500
1000
500
0
0
100
200
Colors
Pixels
3000
2000
1000
0
0
100
200
Colors

Examples

Image original

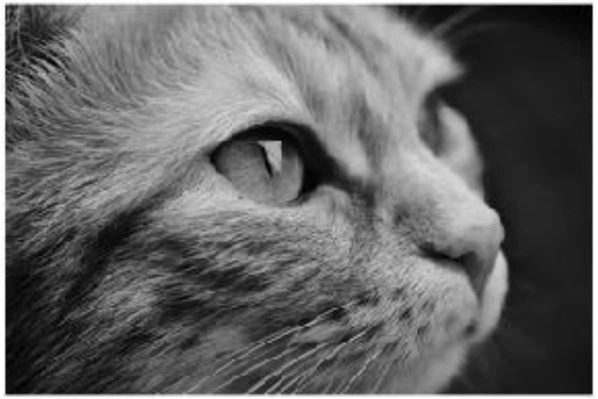
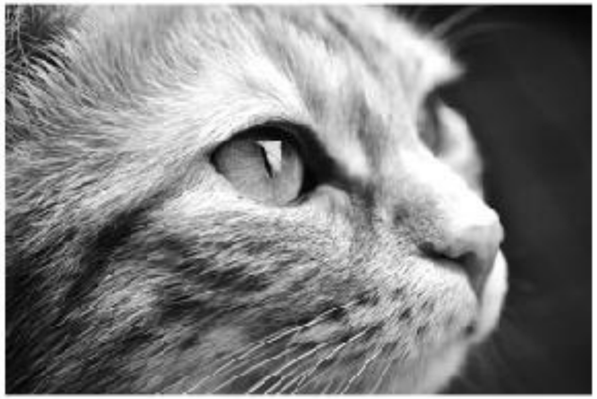
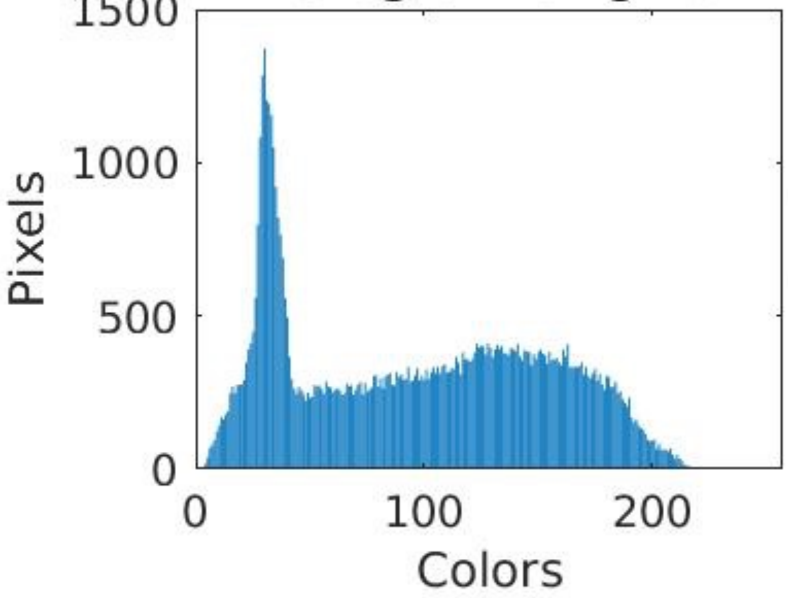


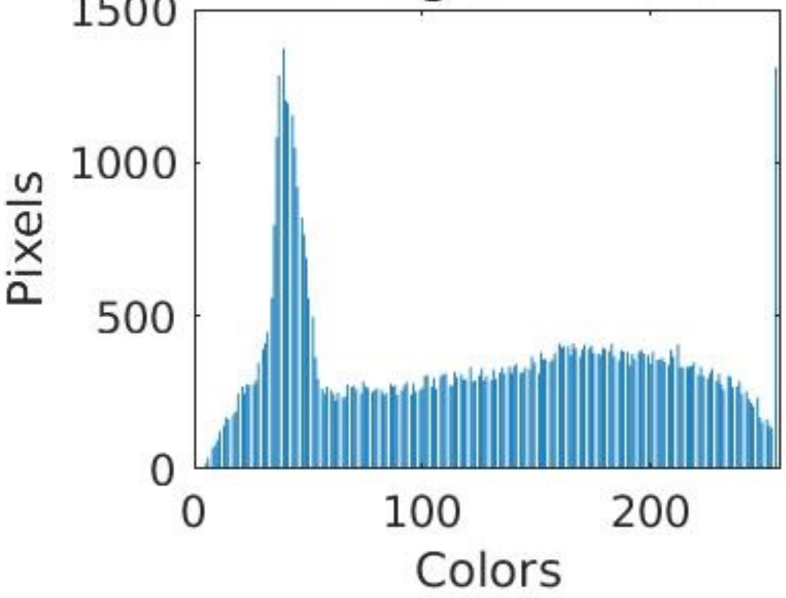
Image transformed



Histogram original



Histrogram Trans

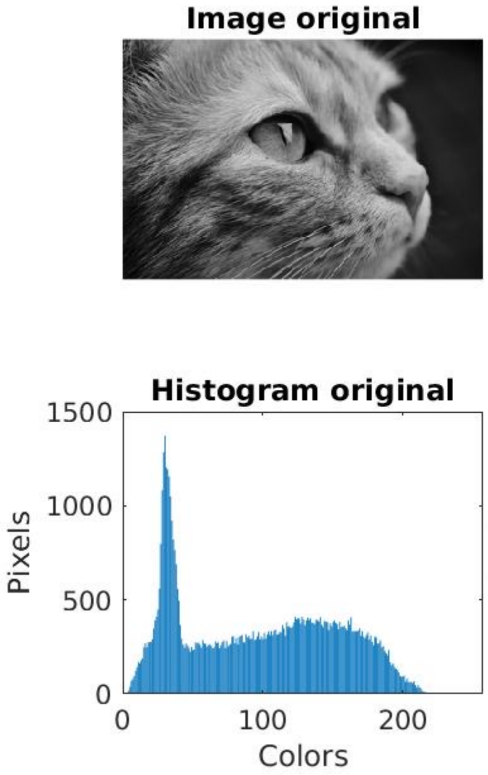


Mean, Variance, Median

- Distribution of intensities describes brightness, contrast and other properties of the image
- **Mean** describes **brightness**
- **Variance** describes **contrast**
- **Median** is a robust description of brightness

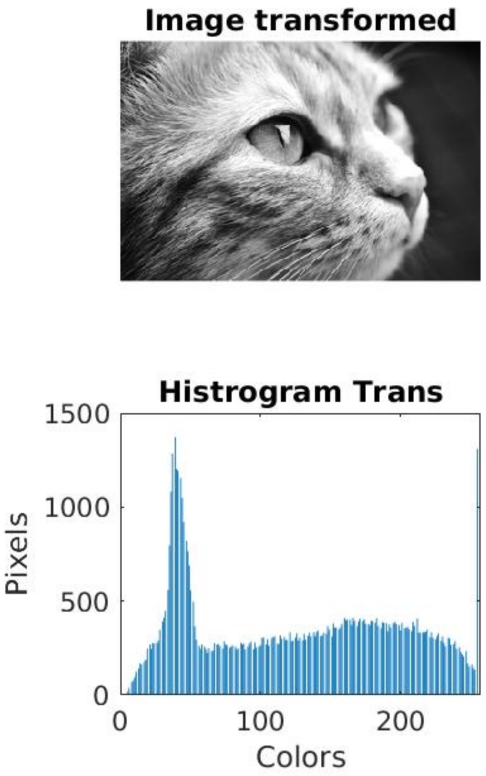
Brightness and Contrast

- A transformation that changes the mean, changes the image brightness
- Analogous for variance and contrast



increasing variance

increases contrast

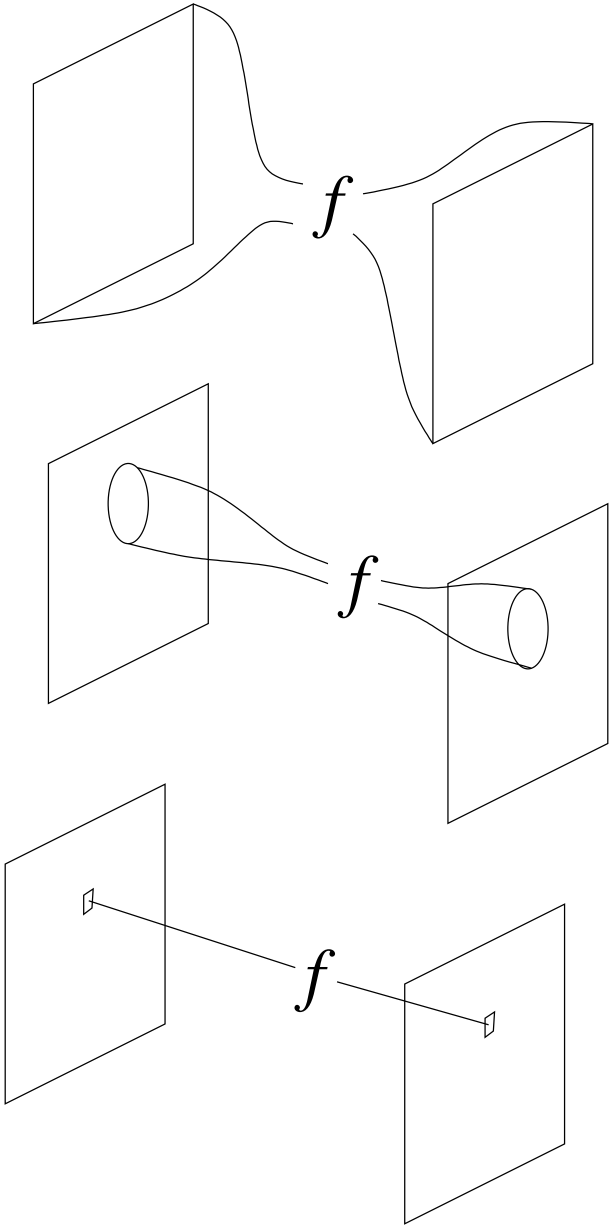


Three Types of such Functions ("Operators")

- Global operator

- Local operator

- Point operator

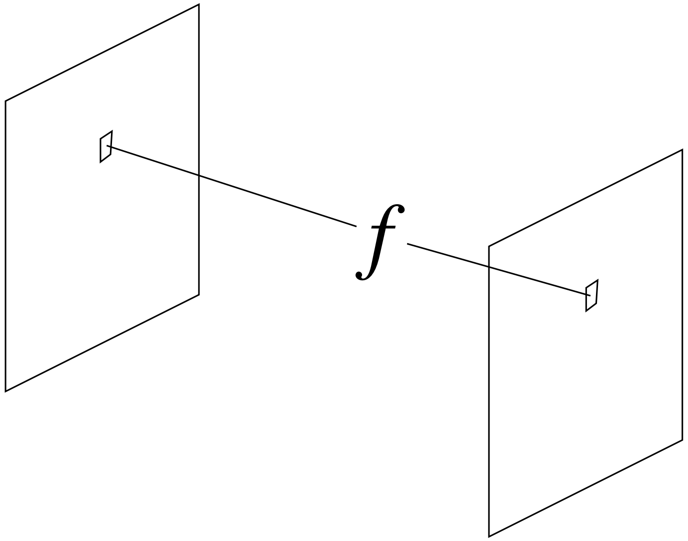


Point Operator

- Maps an intensity value to a new value **only** based on the the input value

$$v_{new} = f(v)$$

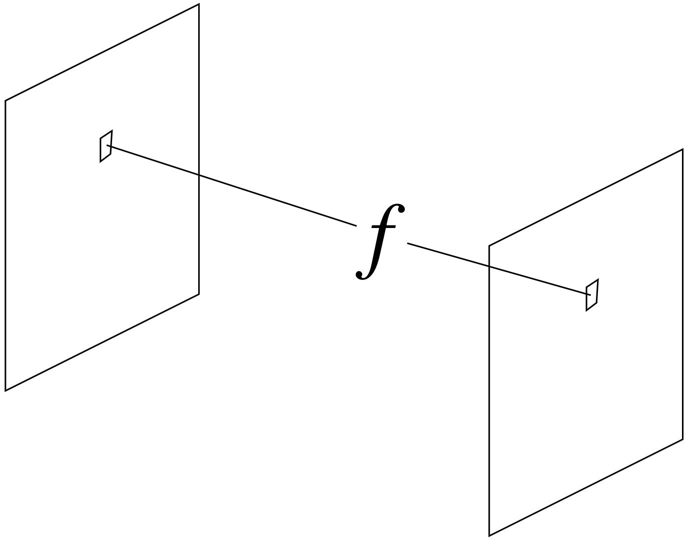
- Mapping is **independent of the values of other pixels**.



Point Operator

$$\underset{\substack{\uparrow\\ \text{output}}}{b(i,j)} = \underset{\substack{\uparrow\\ \text{operator}}}{f}(\underset{\substack{\uparrow\\ \text{input}}}{a(i,j)}, \underset{\substack{\uparrow\\ \text{parameter}}}{\boldsymbol{p}})$$

$$[i,j] \in [1 \dots I] \times [1 \dots J]$$



Linear Function

- Basic class of functions

$$b(i,j) \quad = \quad f(a(i,j),\boldsymbol{p})$$

$$f(a,\boldsymbol{p}) \quad = \quad k + m\,a \qquad \boldsymbol{p} = [k,m]^\top$$

$$b(i,j) \quad = \quad k + m\,a(i,j)$$

Error Propagation under a Linear Function

- The linear function

$$f(a, \boldsymbol{p}) \quad = \quad k + m\,a$$

- transforms mean and standard deviation as

$$\mu_b = k + m\,\mu_a \qquad\qquad \sigma_b = |m|\,\sigma_a$$

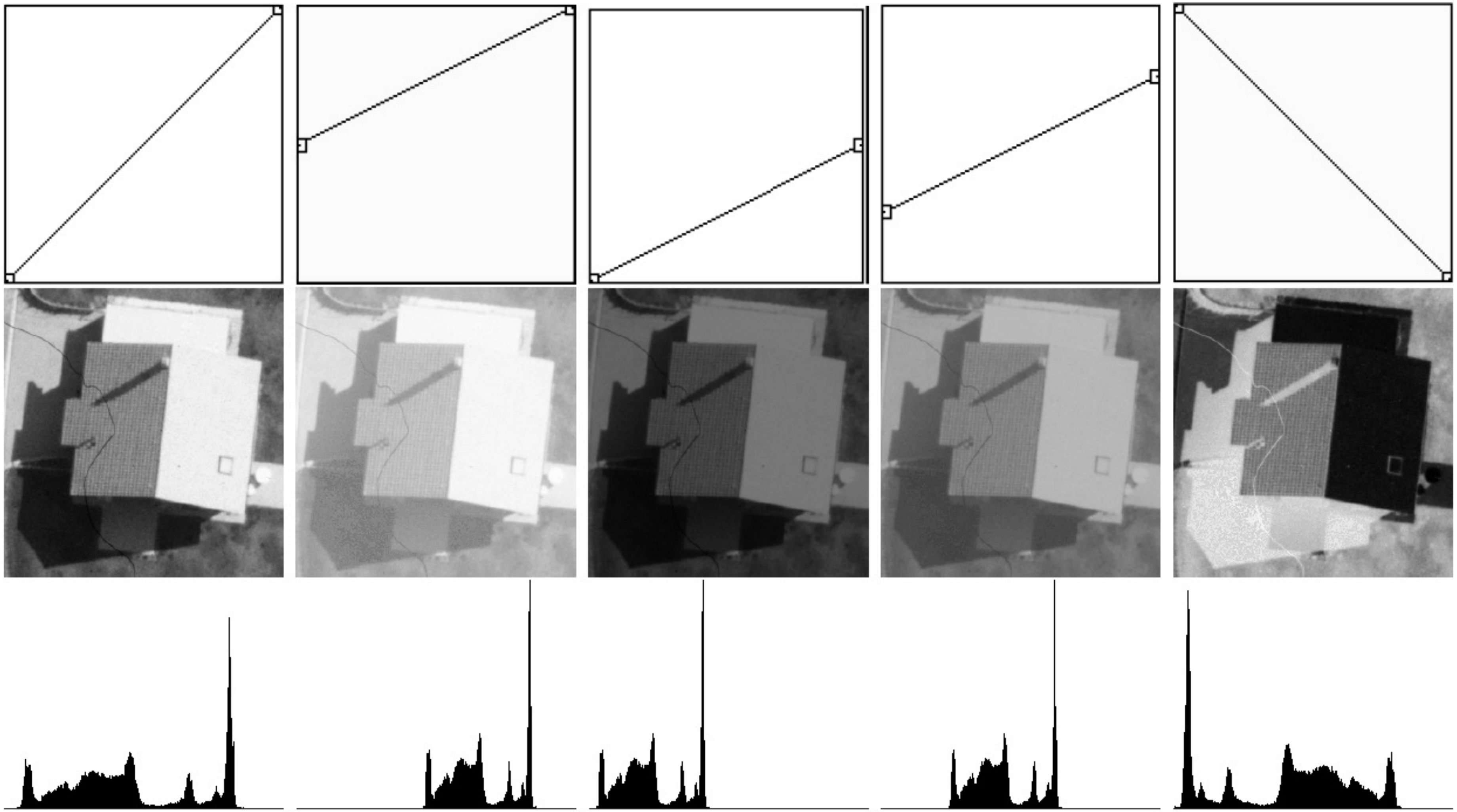
Popular Image Transformations

- Adjust **brightness**
  increase: $k > 0$
  decrease: $k < 0$

- Adjust **contrast**
  increase: $m > 1$
  decrease: $0 < m < 1$

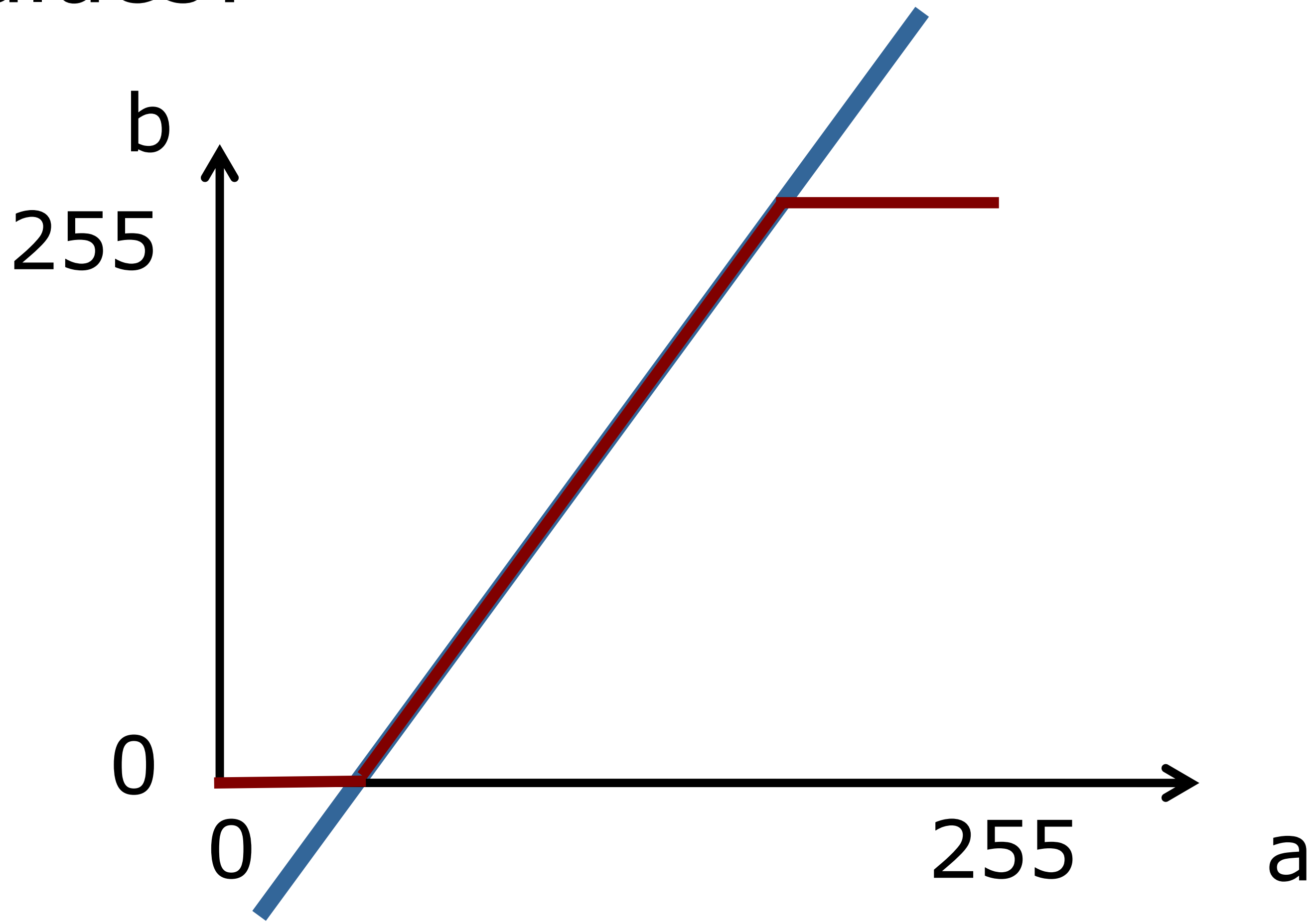
- **Contrast inversion**
  negative image: $m = -1$

Examples



Linear Function?

Does such a linear function really lead to a linear mapping of the intensity values?



Nonlinear Functions

- Example: thresholding

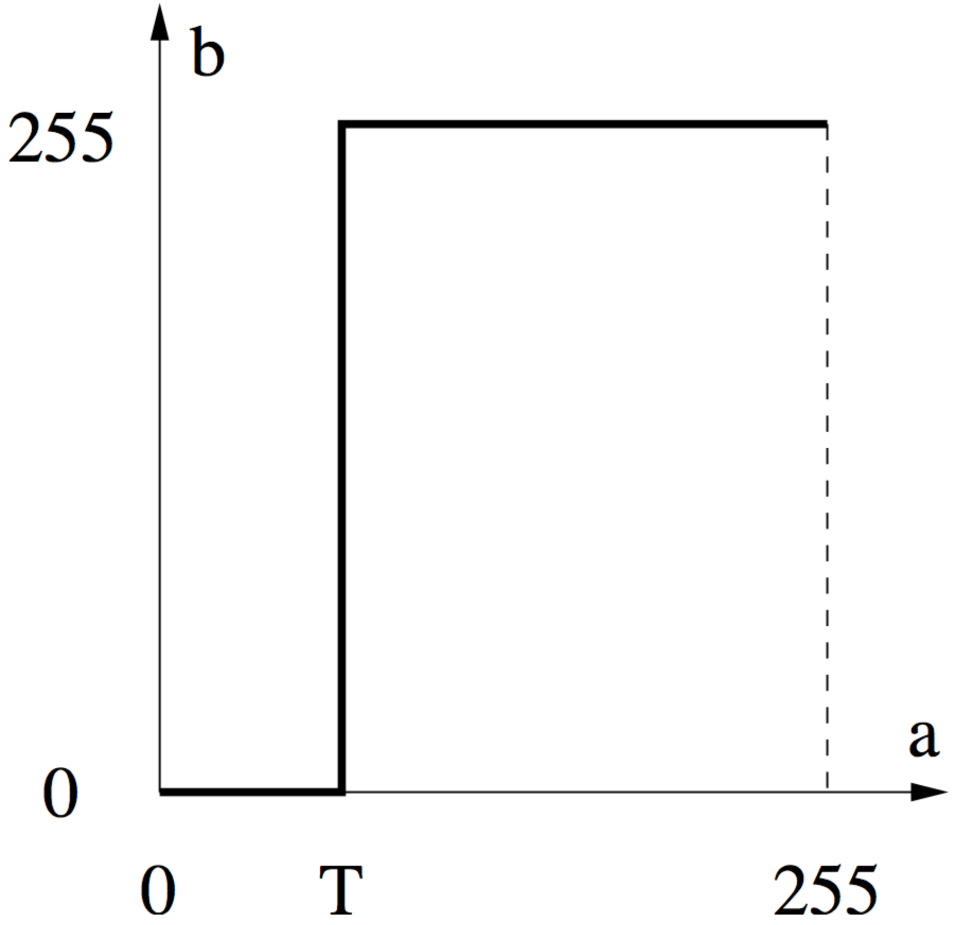
$$b(a) = \begin{cases} b_0 \text{ if } a < T \\ b_1 \text{ otherwise} \end{cases}$$

- Often, we have

$$(b_0, b_1) = (0, 255) \text{ or } = (255, 0)$$

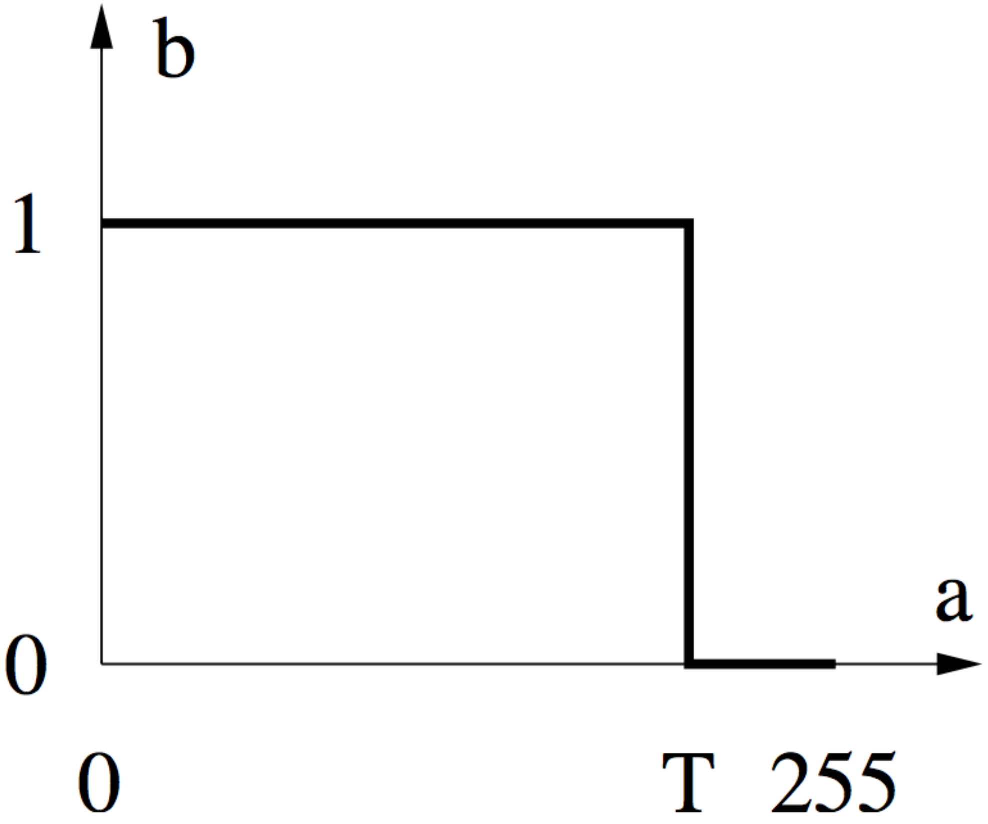
Thresholding

- Example: $b(a) = \begin{cases} 0 \text{ if } a < T \\ 255 \text{ otherwise} \end{cases}$



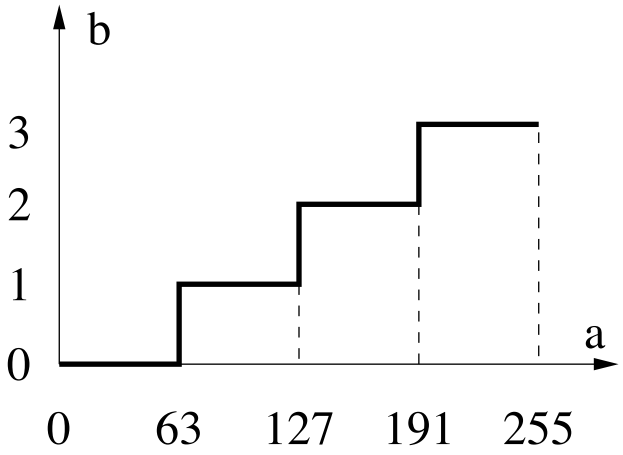
Binary Image

- With $(b_0, b_1) = (1, 0) \text{ or } = (0, 1)$ we obtain a binary image



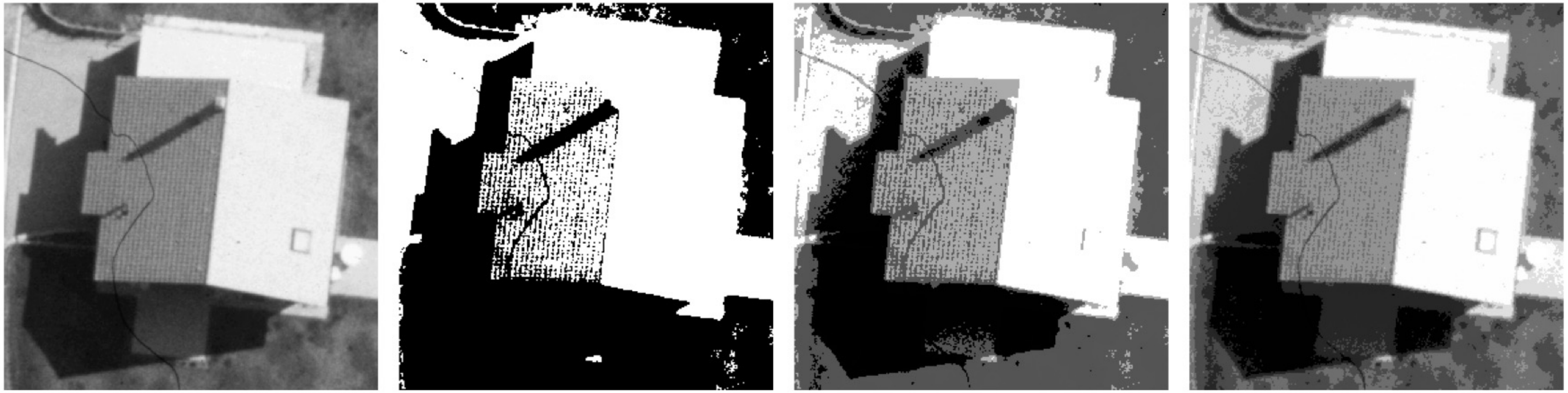
Quantization

- Binary image with two values is a special case of quantization
- Quantization means rounding the gray values to a set of discrete values



Quantization

- Example (256, 2, 4, 8 discrete values)



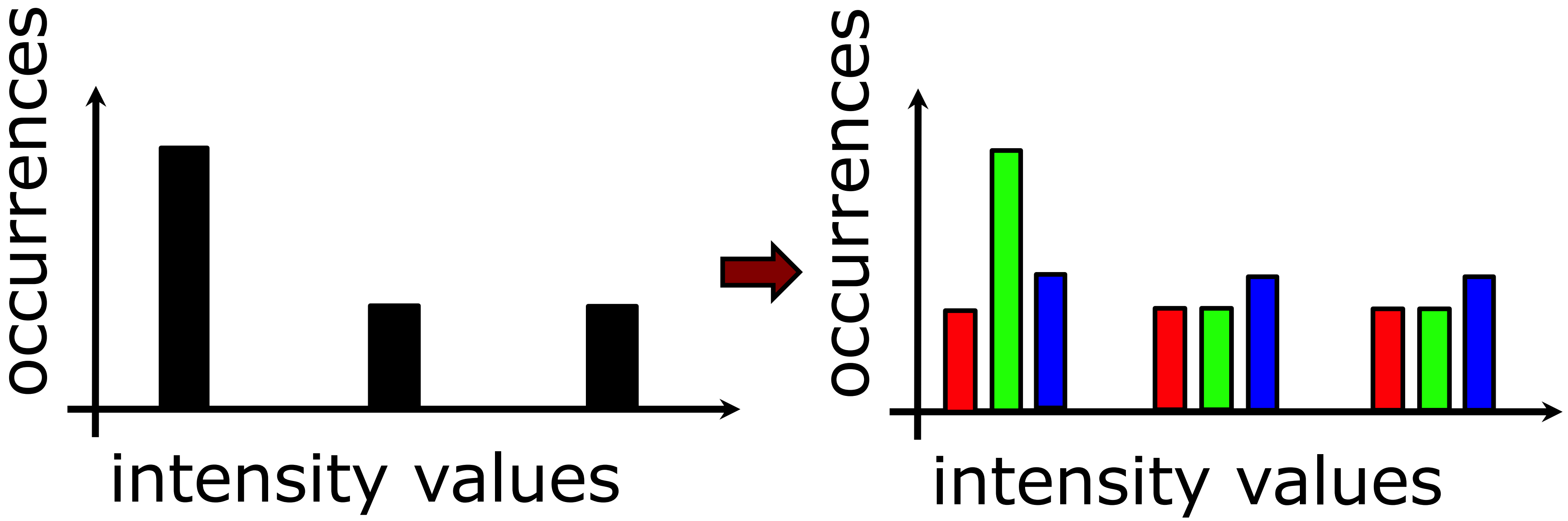
- For an optimal image encoding:
  use the quantization that minimize
  the rounding error given the image

Realizing Point Operators

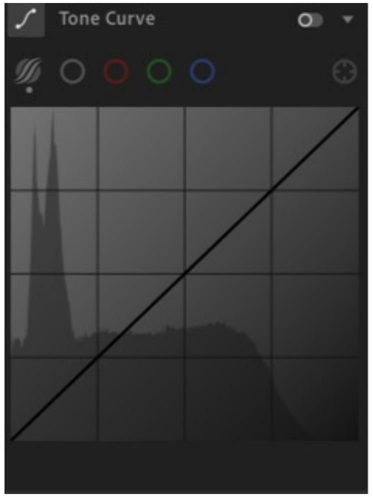
- Point operators can be efficiently computed using look-up tables
- All 256 possible outcomes are stored in an array
- Function evaluation corresponds to reading a byte from memory

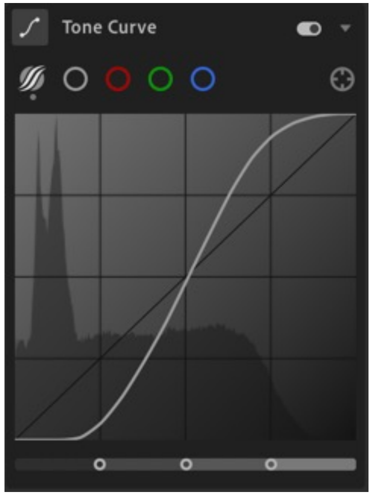
Color Images

Build a histogram for each channel



Tone Curve in Photography





Summary

- Histograms of image intensities
- Histograms allow us to analyze images
- Mean = brightness
- Variance = contrast
- Point operators vs. local/global operators
- Operators as tools for image processing

Slide Information

- The slides have been created by Cyrill Stachniss as part of the photogrammetry and robotics courses.
- **I tried to acknowledge all people from whom I used images or videos. In case I made a mistake or missed someone, please let me know.**
- The photogrammetry material heavily relies on the very well written lecture notes by Wolfgang Förstner and the Photogrammetric Computer Vision book by Förstner & Wrobel.
- Parts of the robotics material stems from the great Probabilistic Robotics book by Thrun, Burgard and Fox.
- If you are a university lecturer, feel free to use the course material. If you adapt the course material, please make sure that you keep the acknowledgements to others and please acknowledge me as well. To satisfy my own curiosity, please send me email notice if you use my slides.

Cyrill Stachniss, cyrill.stachniss@igg.uni-bonn.de