ACV – Applied Computer Vision

Bachelor Medientechnik & Creative Computing

Matthias Zeppelzauer
matthias.zeppelzauer@fhstp.ac.at

Djordje Slijepcevic
djordje.slijepcevic@fhstp.ac.at

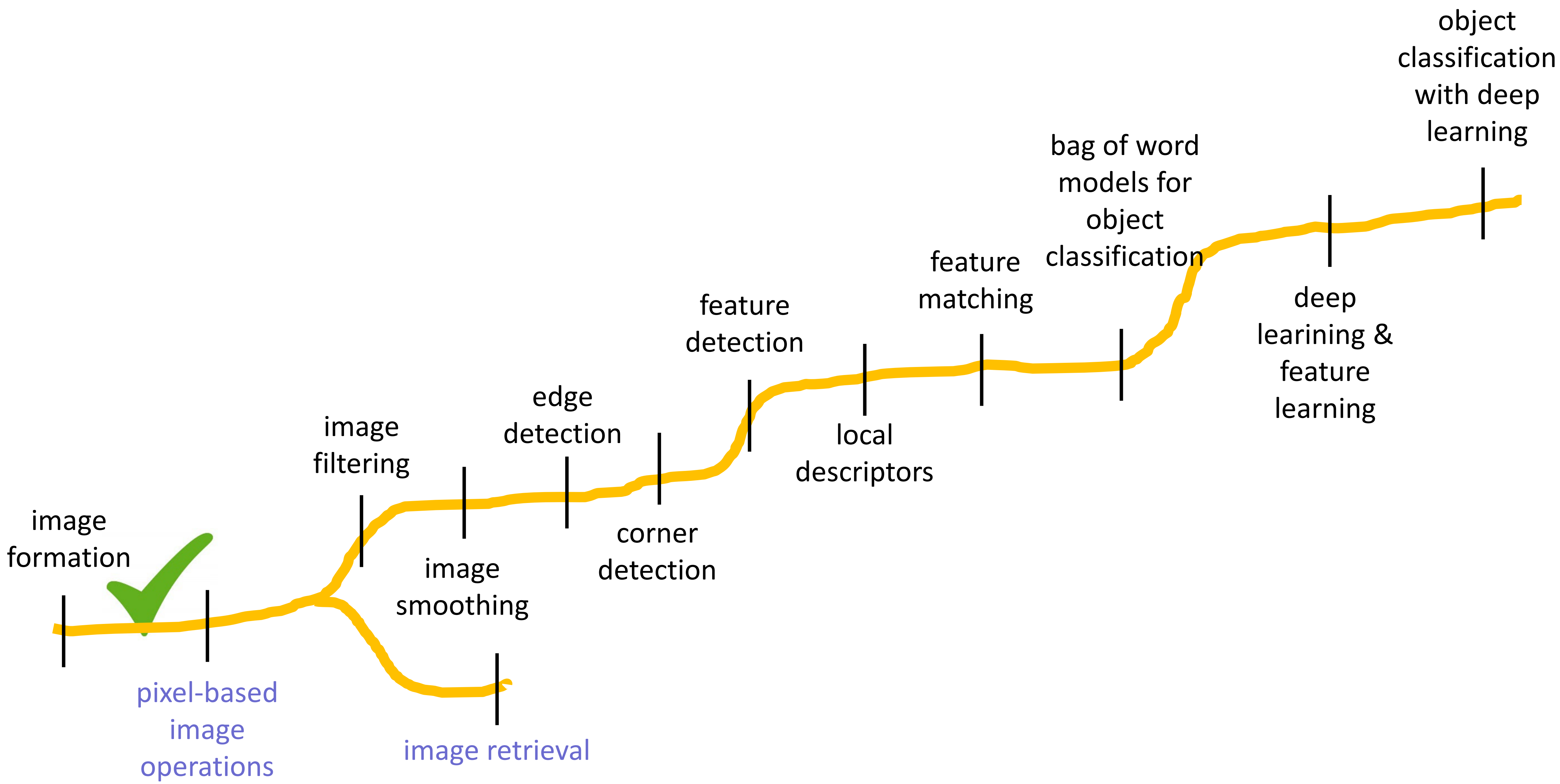
Credits

- Antonio Torralba
- Alexei Efros
- Leon Sigal
- Kristen Grauman
- Steve Seitz
- James Hays

Today

- Last time:
  - Challenges of computer vision
  - Image formation (pinhole camera etc.)
- Basic image operations
- Histograms
- Image Filtering

Roadmap



object classification with deep learning
bag of word models for object classification
feature matching
feature detection
edge detection
image filtering
image formation
deep learining & feature learning
local descriptors
corner detection
image smoothing
pixel-based image operations
image retrieval

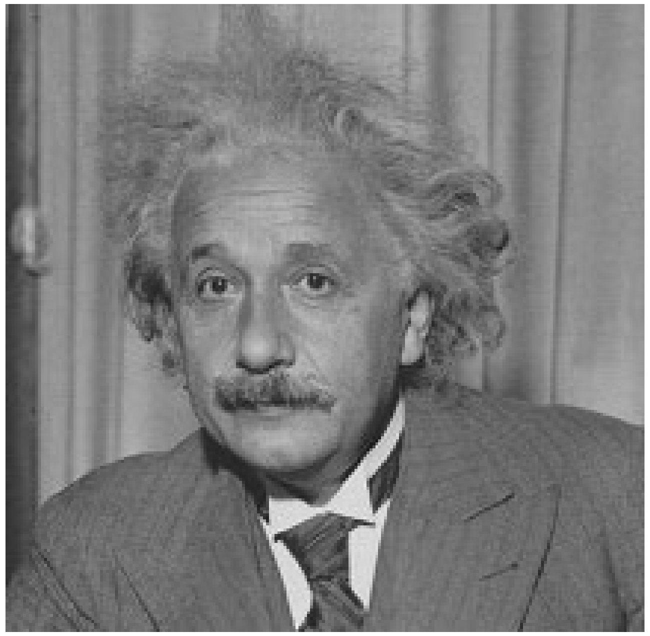
Basic Image Operations

Image Operations - Overview

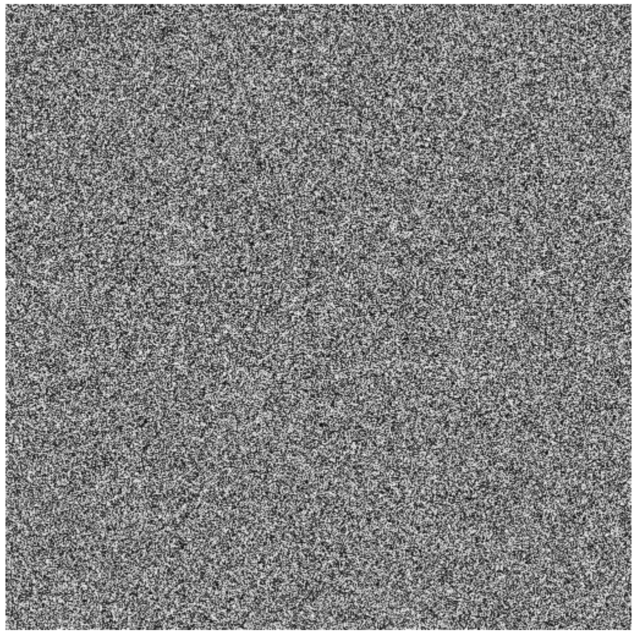
- Arithmetic Operations
  - Add, subtract, multiply, divide, average
  - Not, and, or, xor
- Histograms
  - Applications: contrast improvement, threshold finding
- Thresholding
- Filtering
  - Applications: smoothing, sharpening, edge detection

Arithmetic Operations - Addition

- Addition:
  - Used to „blend“ 2 images
  - Pixels in the same coordinates are added to each other
- Python: im3 = im2 + im1
  - Values become larger, meaning the picture becomes brighter
  - Values could overflow (for example, if you have a range of 0 … 255), so you may need to normalize after adding the images



+



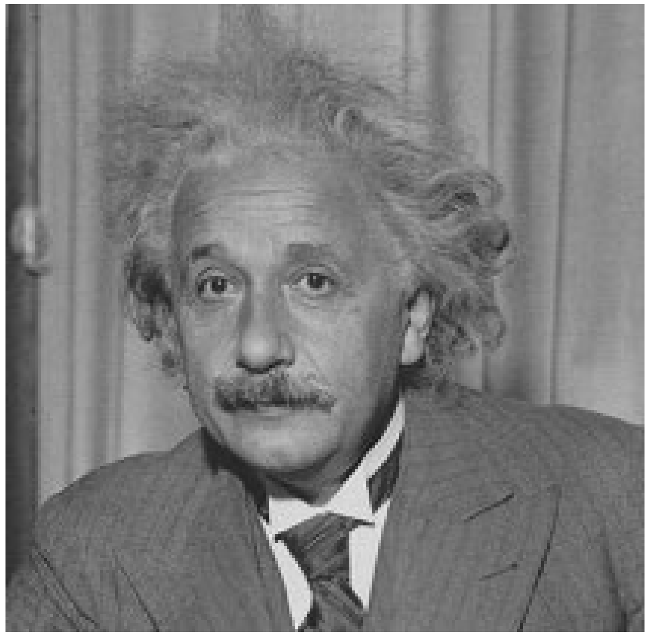
=

Arithmetic Operations - Subtraction

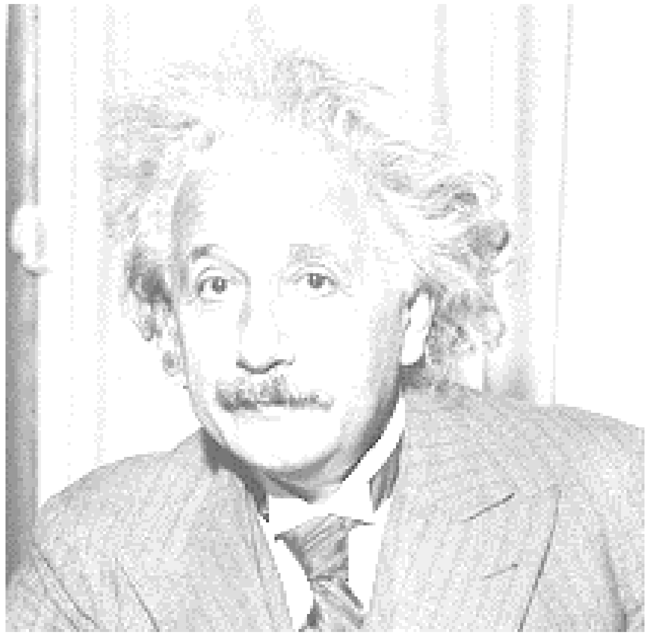
- Substraction:
  - Pixels on the same coordinates are substracted from each other
  - Can be used to find differences in two images of the same scene
  - Areas that did not change turn black
  - Also used to detect motion in subsequent video frames
- Python: im3 = np.abs(im2 - im1)
  - It is important to take the „absolute“ value of the difference, because you cannot display negative values

Arithmetic Operations - Multiplication

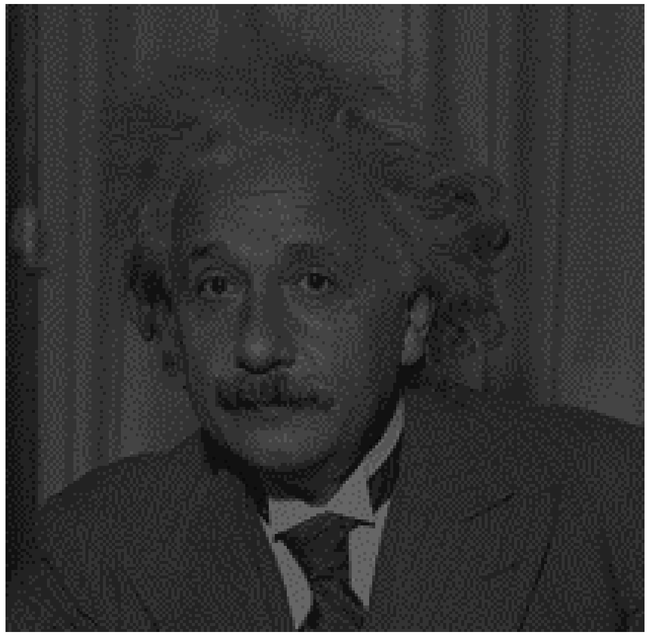
- Multiplication:
  - It scales the grayscale values in the image
  - C > 1: Image becomes brighter, C < 1: Image becomes darker
  - C > 1: You might require clipping, to prevent values from going over the maximum value
- Python: im2 = im1*C
- Using this method for correcting the brightness or contrast of an image is suboptimal, histogram-based approaches provide better results



Original



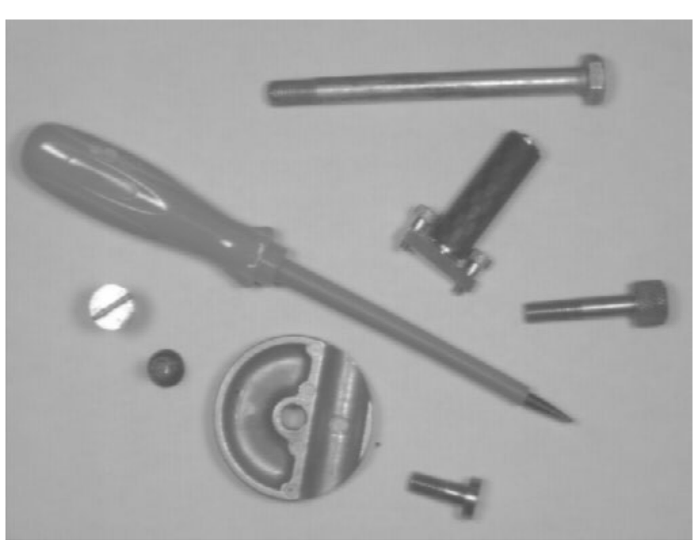
C=2



C=0.5

Arithmetic Operations - Division

- Division:
  - Similar to substraction, can be used to find changes or motion in the image
  - Areas that did not change get the value 1 → different effect on the image depending on the scale you use
  - Depending on the values of the pixels that are not 1, you can deduct which image has the higher value
    - im1 > im2 → [1 255]
    - im1 < im2 → [0 1]
- Python: im3 = im1 / im2



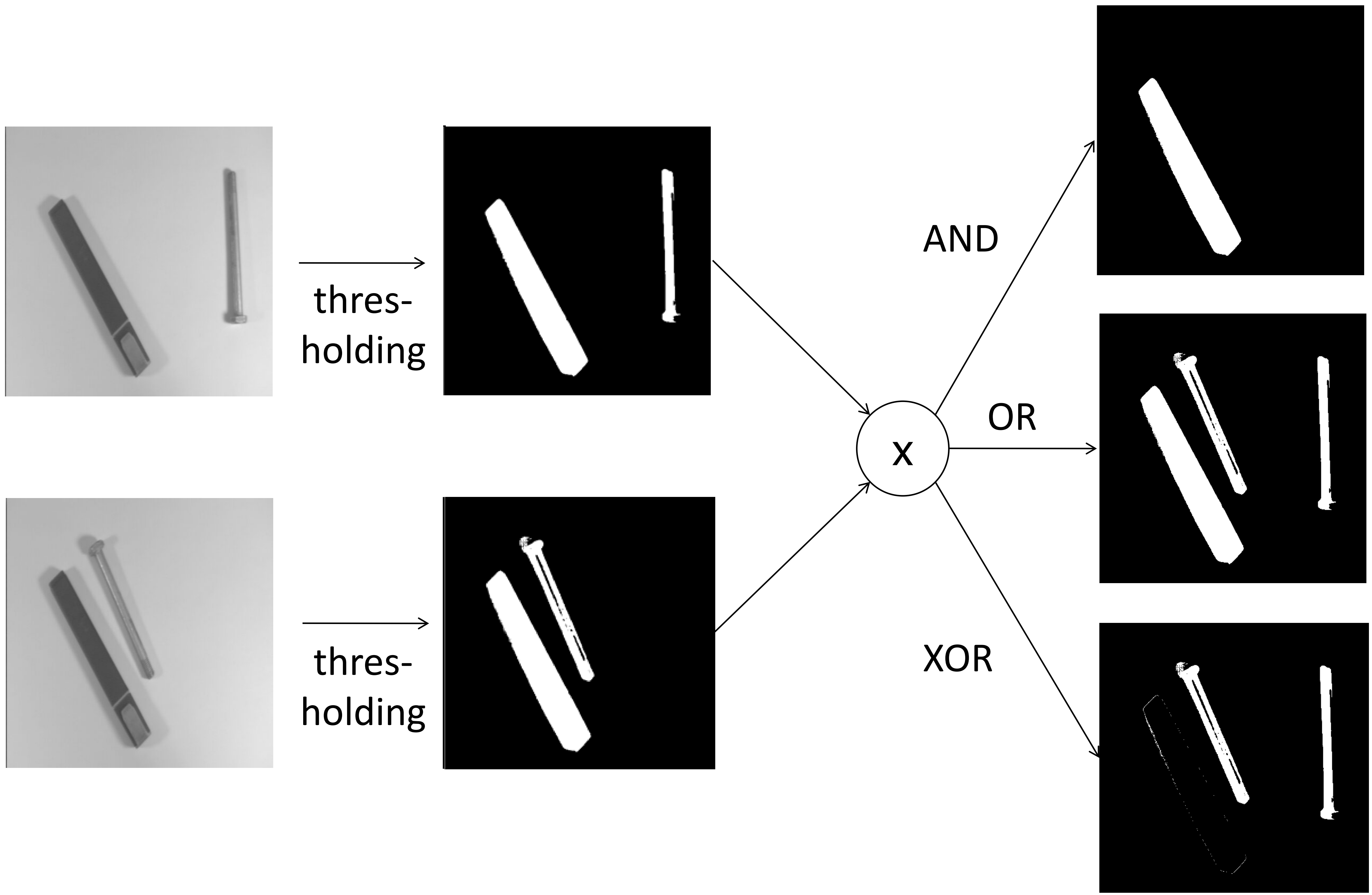
/

=

Logical Operations – AND, OR, XOR, NOT

- Logical Operations:
  - Pixelwise logic arithmetic on an image
  - Only makes sense for binary images, meaning each pixel is either 1 or 0
  - For example, you can mark objects in an image by setting all the pixels belonging to an object to 1 (using thresholding), and then check in the second image which pixels changes (to detect motion perhaps)
- NOT (`not`): Inversion
- AND (`&`, `and`) : Inner join of the images
- OR (`|`, `or`): Union of the images
- XOR (`^`): Detecting changes in the images

Examples – AND, OR, XOR



Questions

- How can arithmetic operations be used to detect motion in a video sequence?

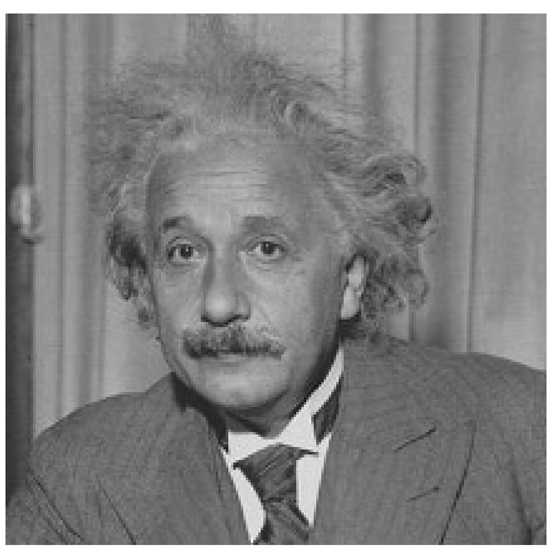
- How can arithmetic operations be used to remove motion in a video sequence?

Image Averaging

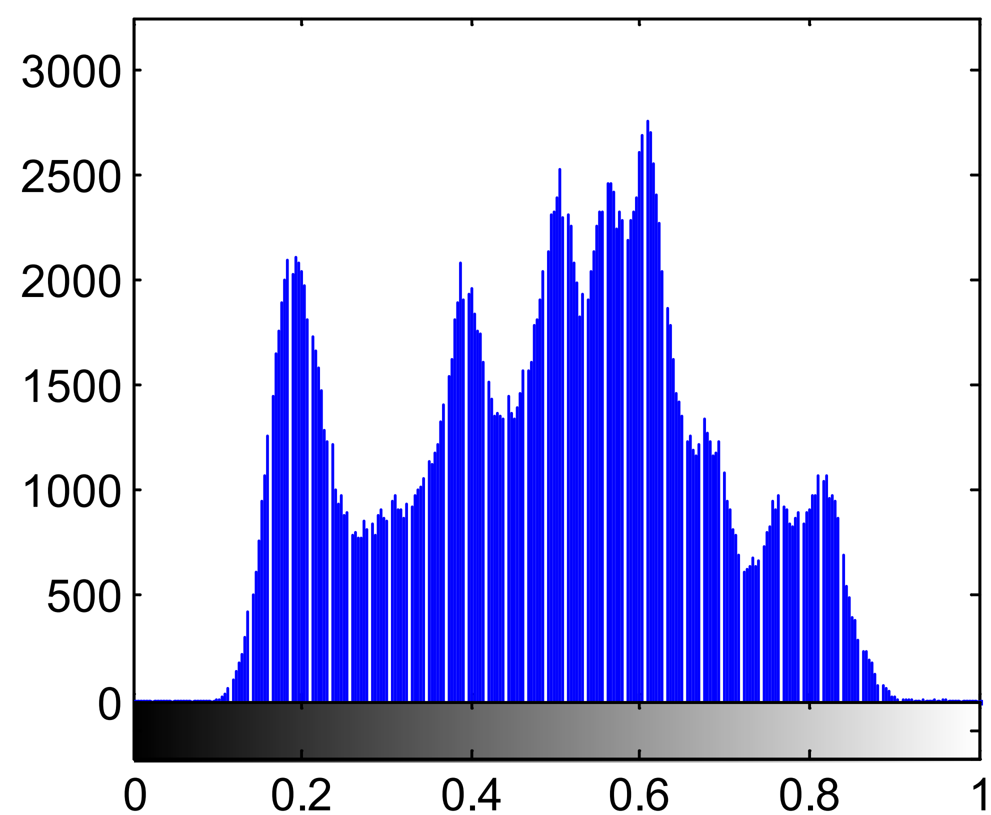
- Motion in images can be removed by image averaging
- Averaging preserves only the constant image content
- Small movements do not prevail in the mean value image
- Example:
  - Webcam of e.g. Stephansplatz
  - Averaging the images over a longer time (1-2 minutes)
  - Result: Image of completely empty Stephansplatz!

Histograms

- Histogram:
  - 1D function that represents the occurrence frequencies of the gray values in an image
  - Normalized histogram represents occurrence probabilities of gray values (normalization factor = number of pixels); corresponds to density function!



`np.histogram(im)` →



Histograms

- Gray level histogram represents the distribution of gray levels in an image
- Calculation: count occurrences of gray levels for a gray lavel range
- Optional: divide histogram by total number of pixels (normalization)
- Properties:
  - Resolution of histogram depends on size of histogram bins
  - Process is dimension-reducing and irreversible!
  - Location information of the image is completely lost
  - Histogram is rotation and translation invariant; normalized histogram also scale invariant
  - But: completely different images can have the same histograms

Histrogram Applications

- **Contrast enhancement**
- Threshold search for image segmentation
- Search for similar images: Content-Based Image Retrieval (CBIR)

Histrogram Applications

- Contrast enhancement of over- and underexposed images

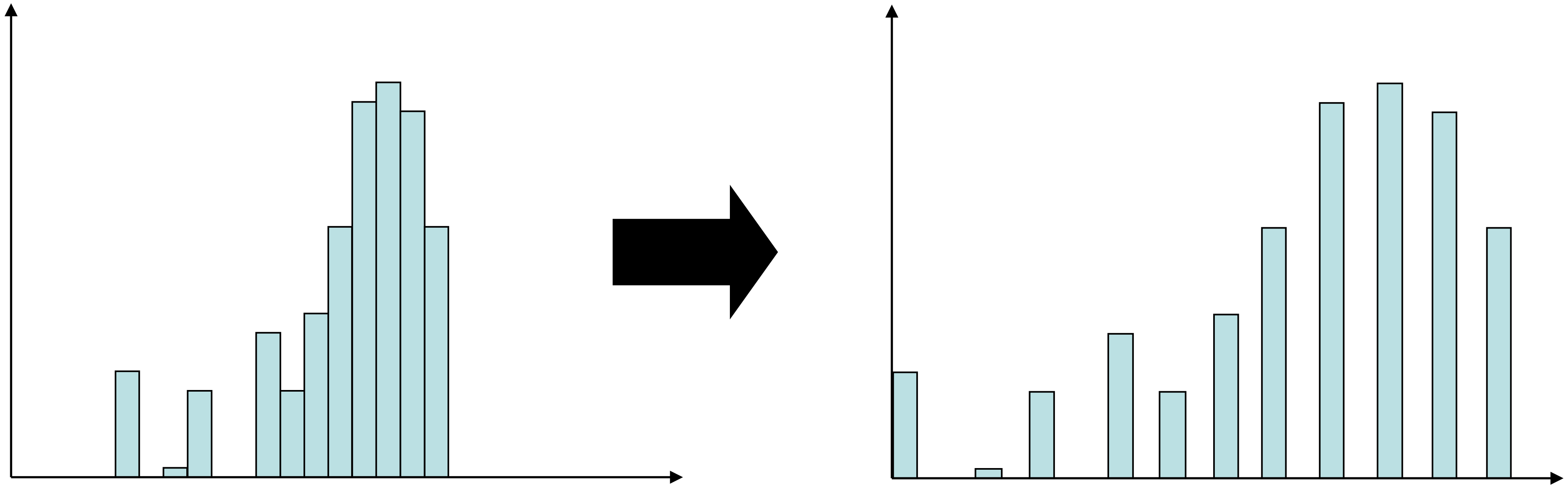
overexposed

underexposed

Contrast enhanced

Contrast Enhancement - Histogram Stretch

- Low contrast narrow histogram not utilizing all gray levels.
- Histogram Stretch (Contrast Stretch):
  - Stretch gray level range linearly
  - Most common linear point operation
  - Utilize entire gray level range



Histogram Stretch

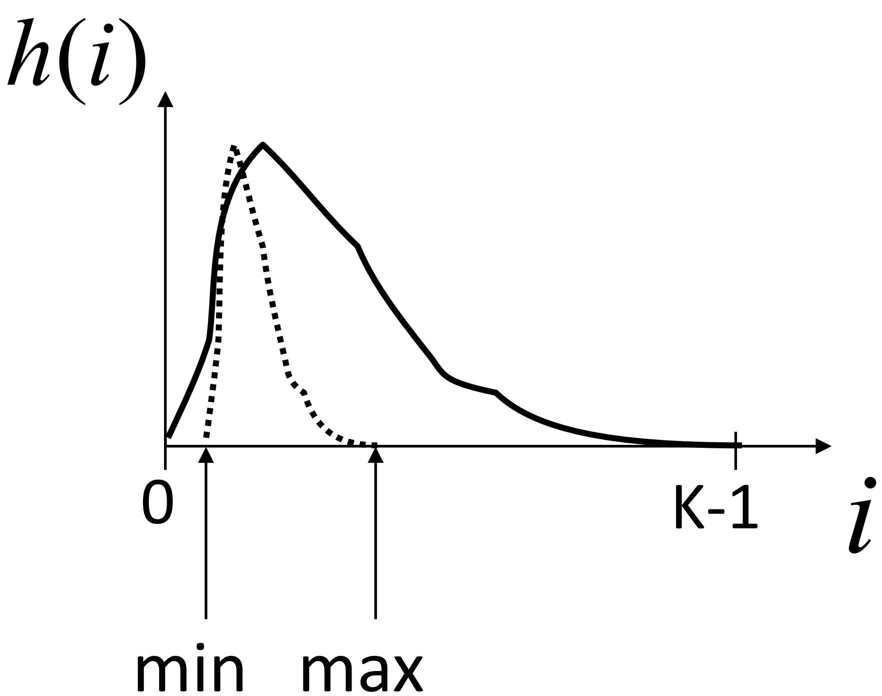
$A(x, y)$

$B(x, y)$

$$B(x, y) = \alpha\, A(x, y) - \beta$$

Scaling

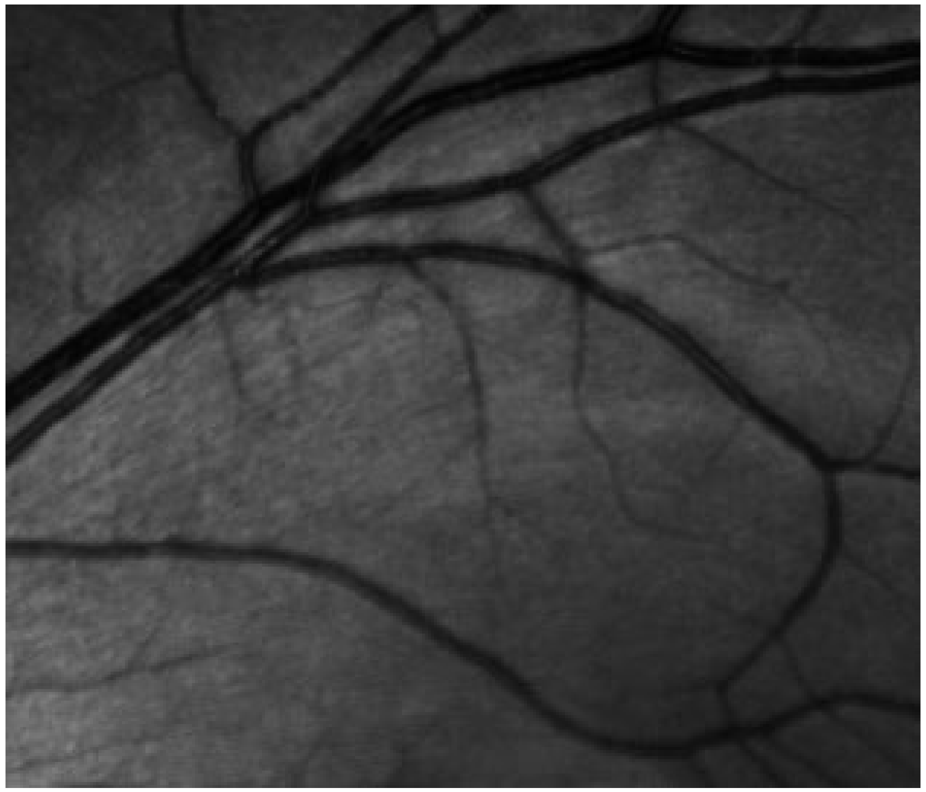
Shift

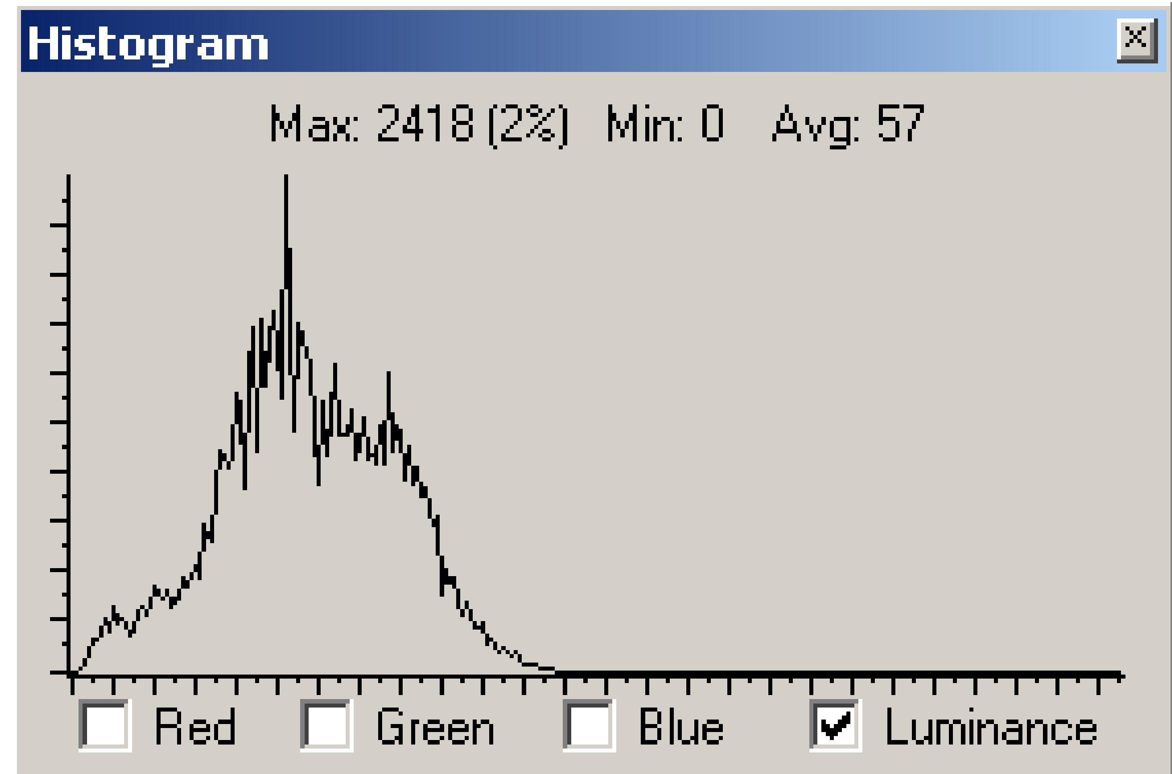


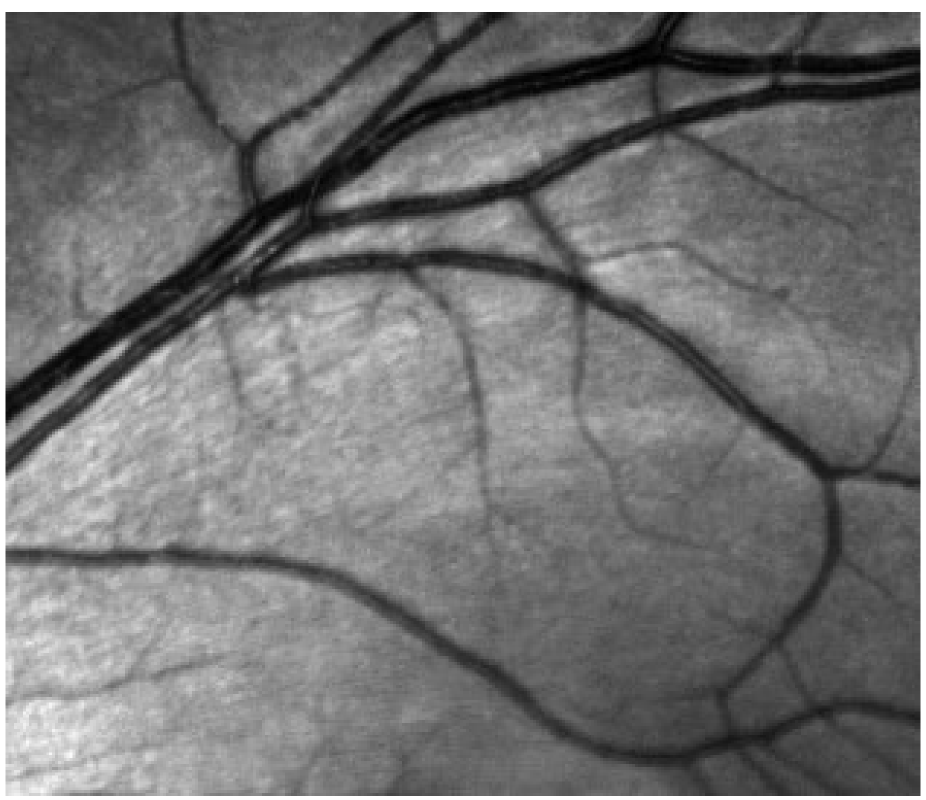
$$\alpha = \frac{(K-1)}{\max - \min}$$

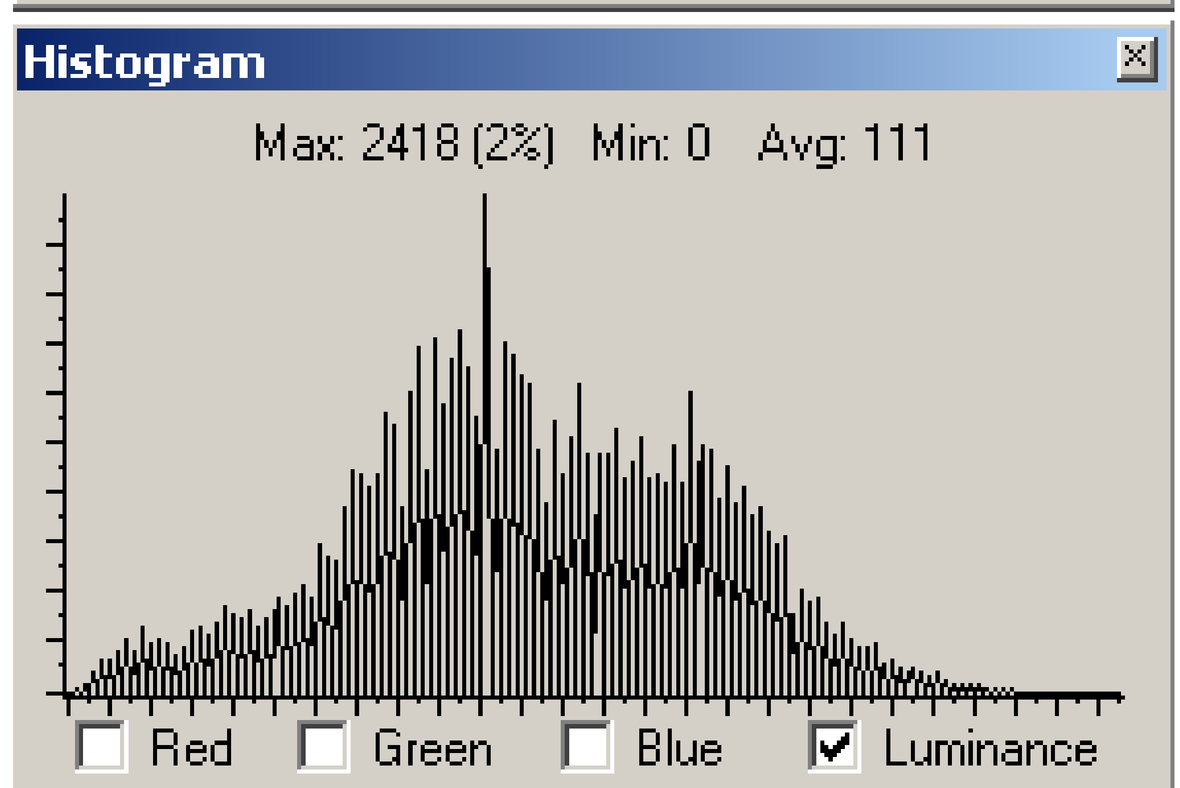
$$\beta = \frac{K-1}{\max - \min} \times \min$$

Histogram Stretch - Example









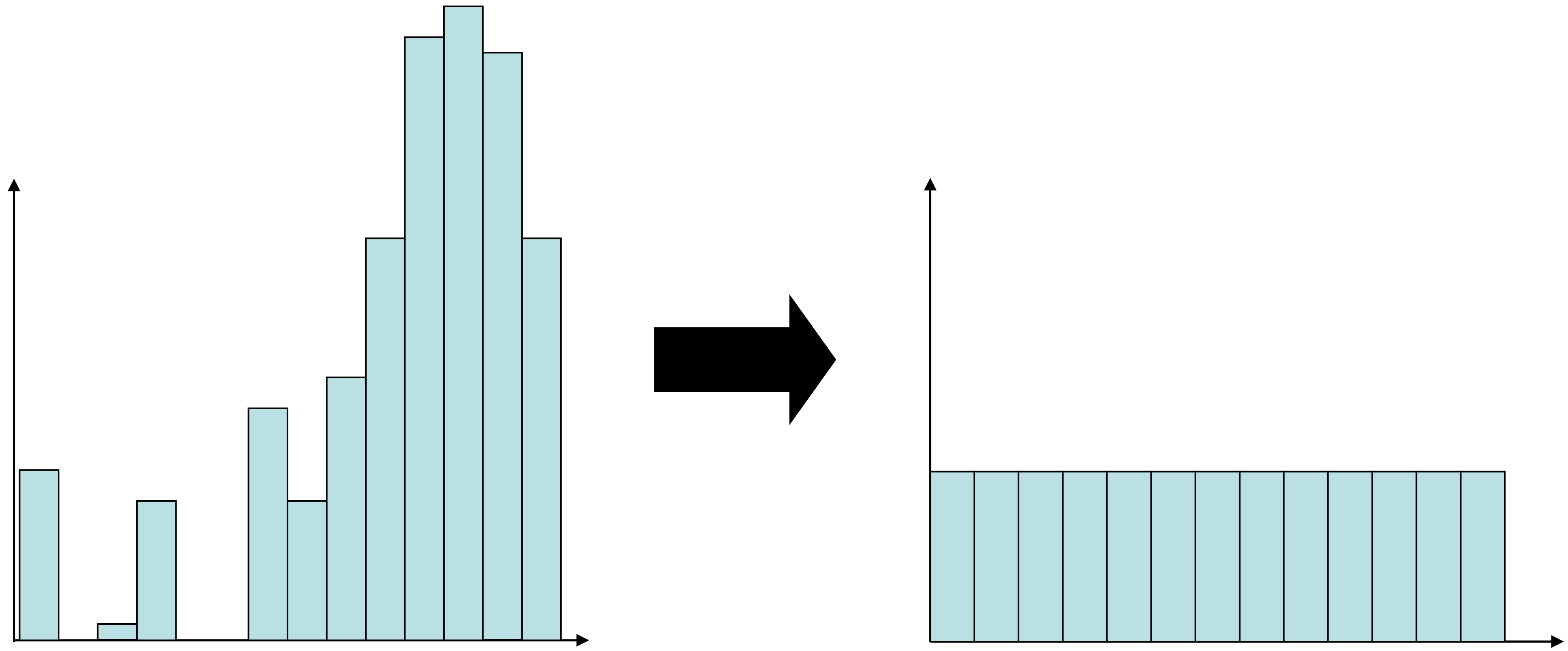
Histogram Stretch - Summary

- Advantage:
  - Simple (linear)
  - Fast

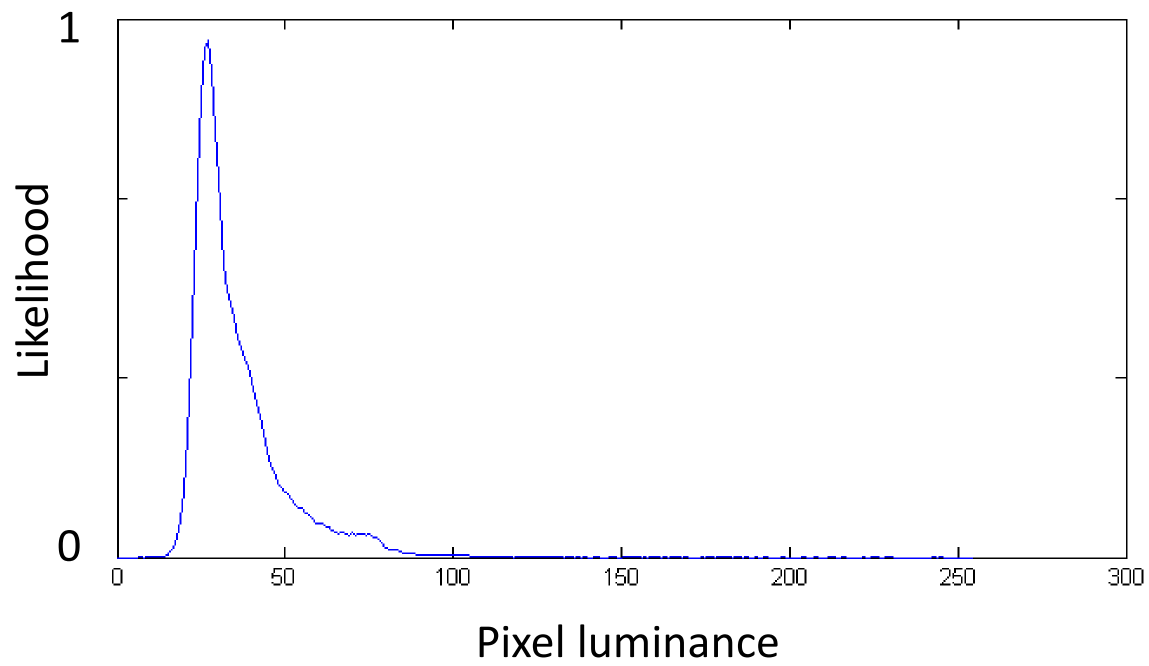
- Disadvantage:
  - Hardly any improvement if gray value distribution is already wide.
  - Sensitive to outliers: Worst case: 1 white and one black pixel no scaling at all!

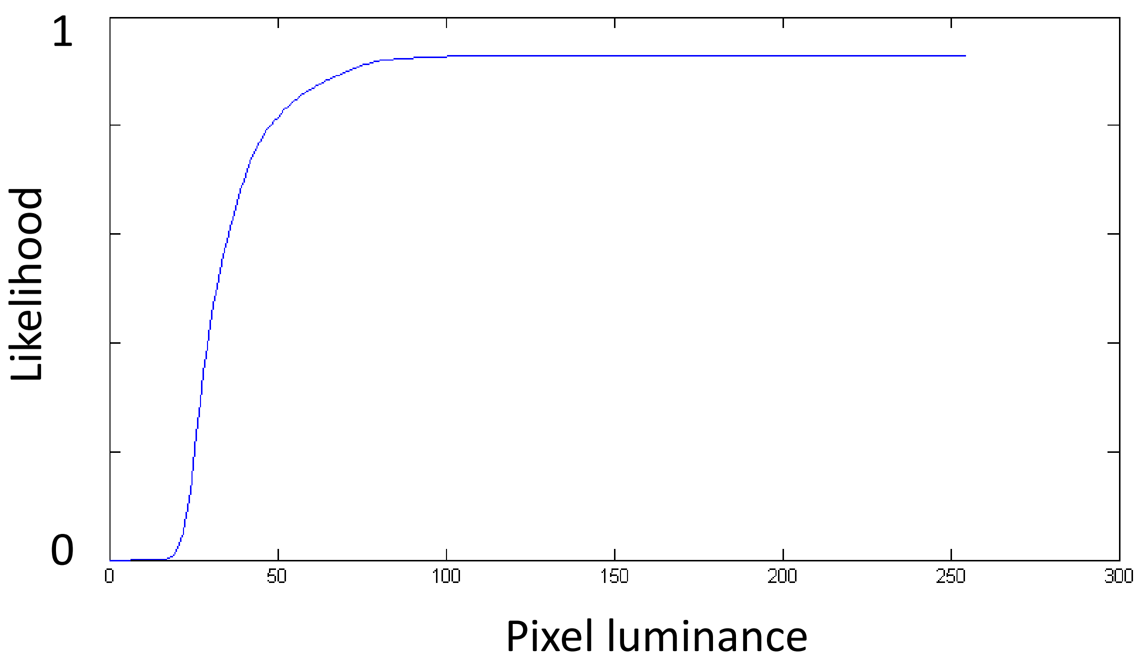
Histogram Equalization

- Process for increasing contrast by spreading the histogram out to be approximately uniformly distributed
- Goal: use entire gray level range and a uniformly distributed (flat) histogram!
- Flat histogram maximizes the information content (cf. entropy)
- Nonlinear point operation → Changes shape of histogram!



Histogram Equalization





Histogram h(x) /
Density function

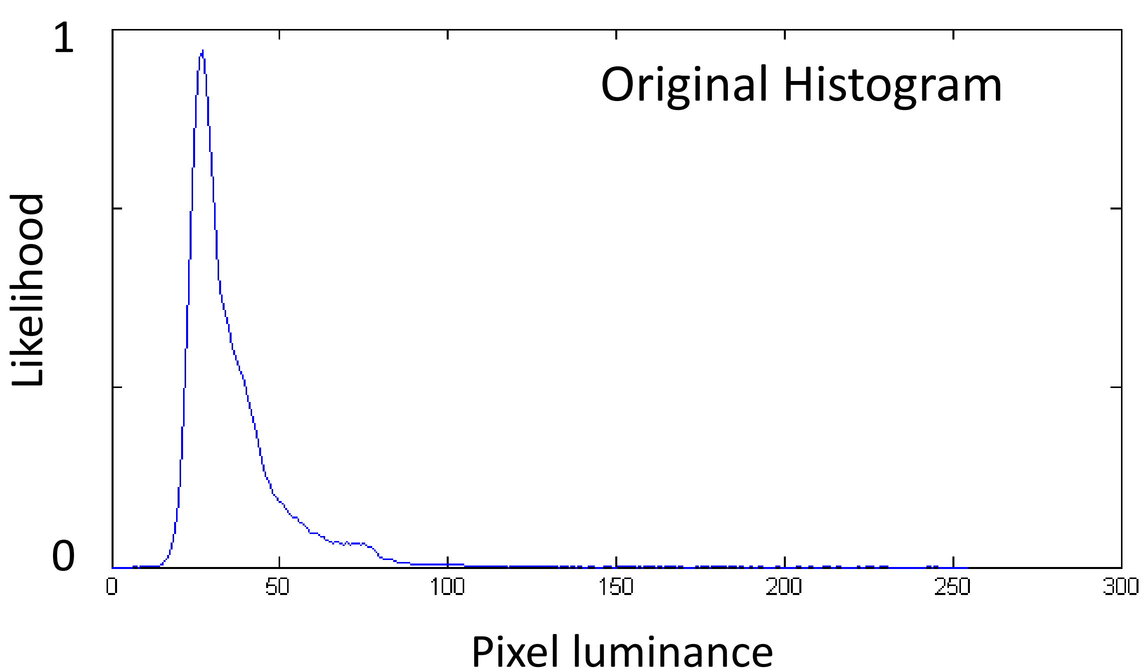
Cumulative histogram /
Cumulative distribution function:
c(i)

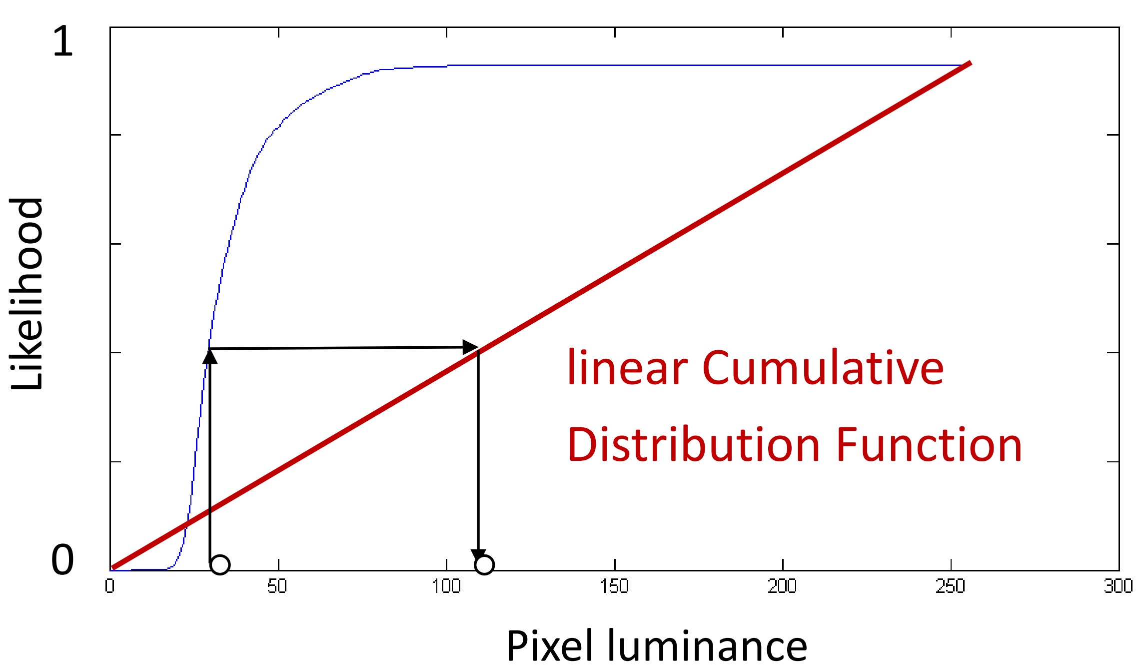
$$c(i) = \int_0^i h(x)dx$$

Histogram Equalization

- Distribution function of a uniform distribution is linear
- Goal: make Distribution function c(i) linear
- For linear distribution function holds:

$$c(i) = \frac{1}{N-1} f(i) \quad \Longrightarrow \quad f(i) = c(i) * (N-1), \qquad N = \#gray\ levels$$



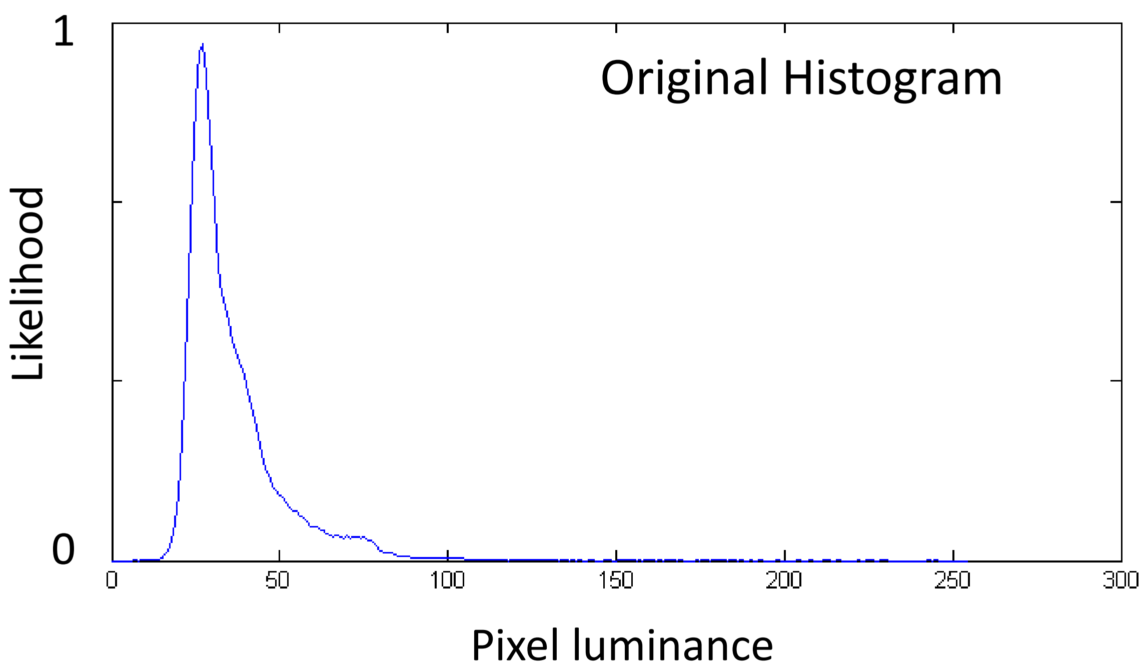


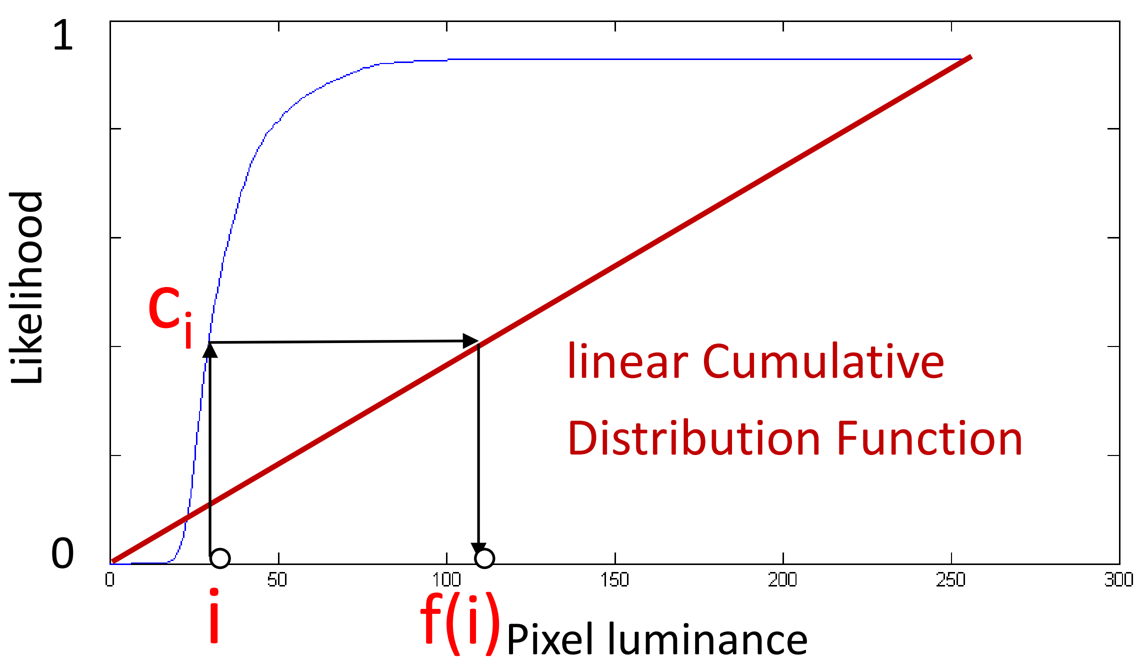
- f(i)is mapping from old gray level to new gray level --> replace gray level i in original image with f(i) in new image

Histogram Equalization

- Distribution function of a uniform distribution is linear
- Goal: make Distribution function c(i) linear
- For linear distribution function holds:

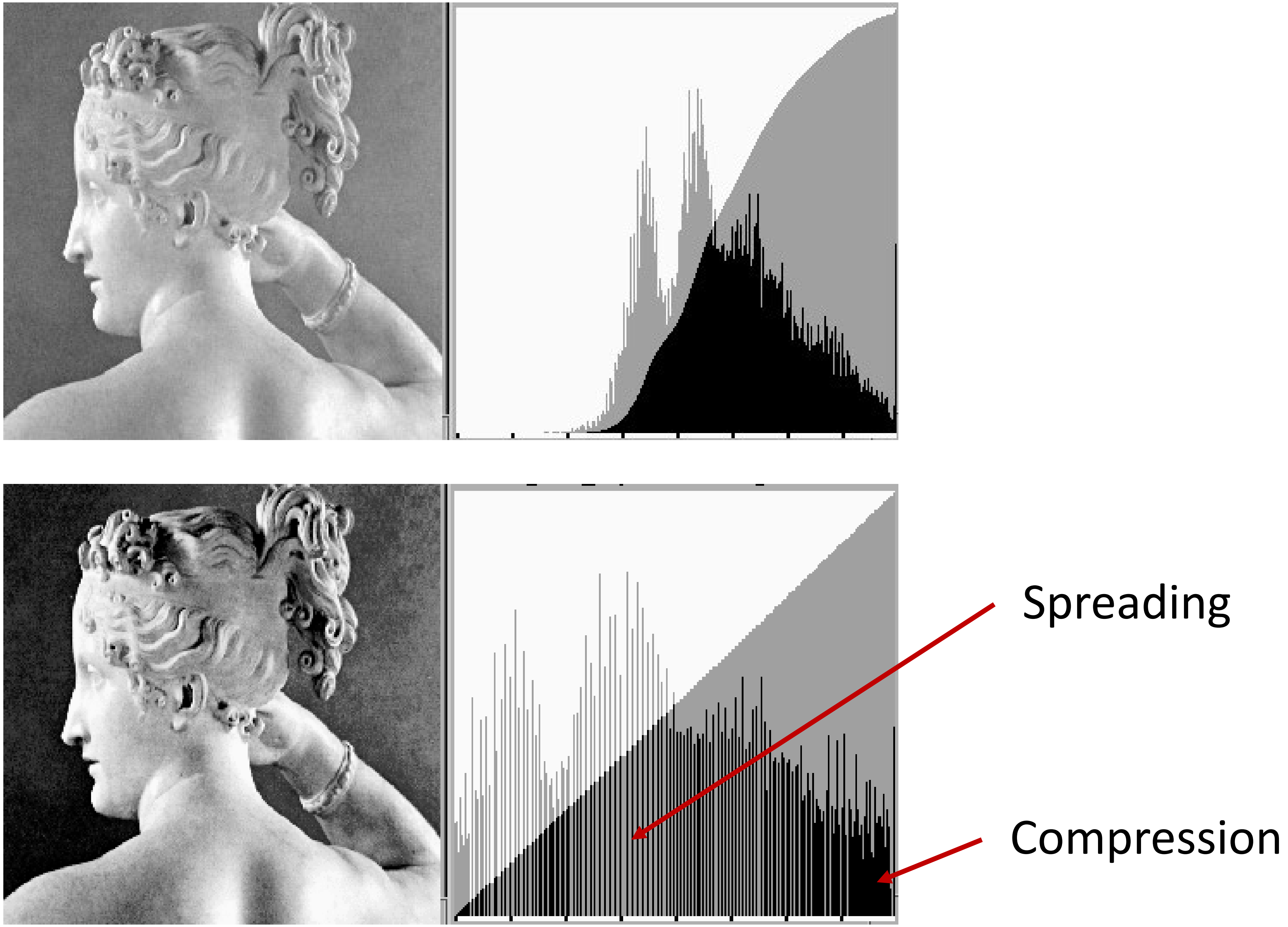
$$c(i) = \frac{1}{N-1} f(i) \implies f(i) = c(i) * (N-1), \qquad N = \#gray\ levels$$





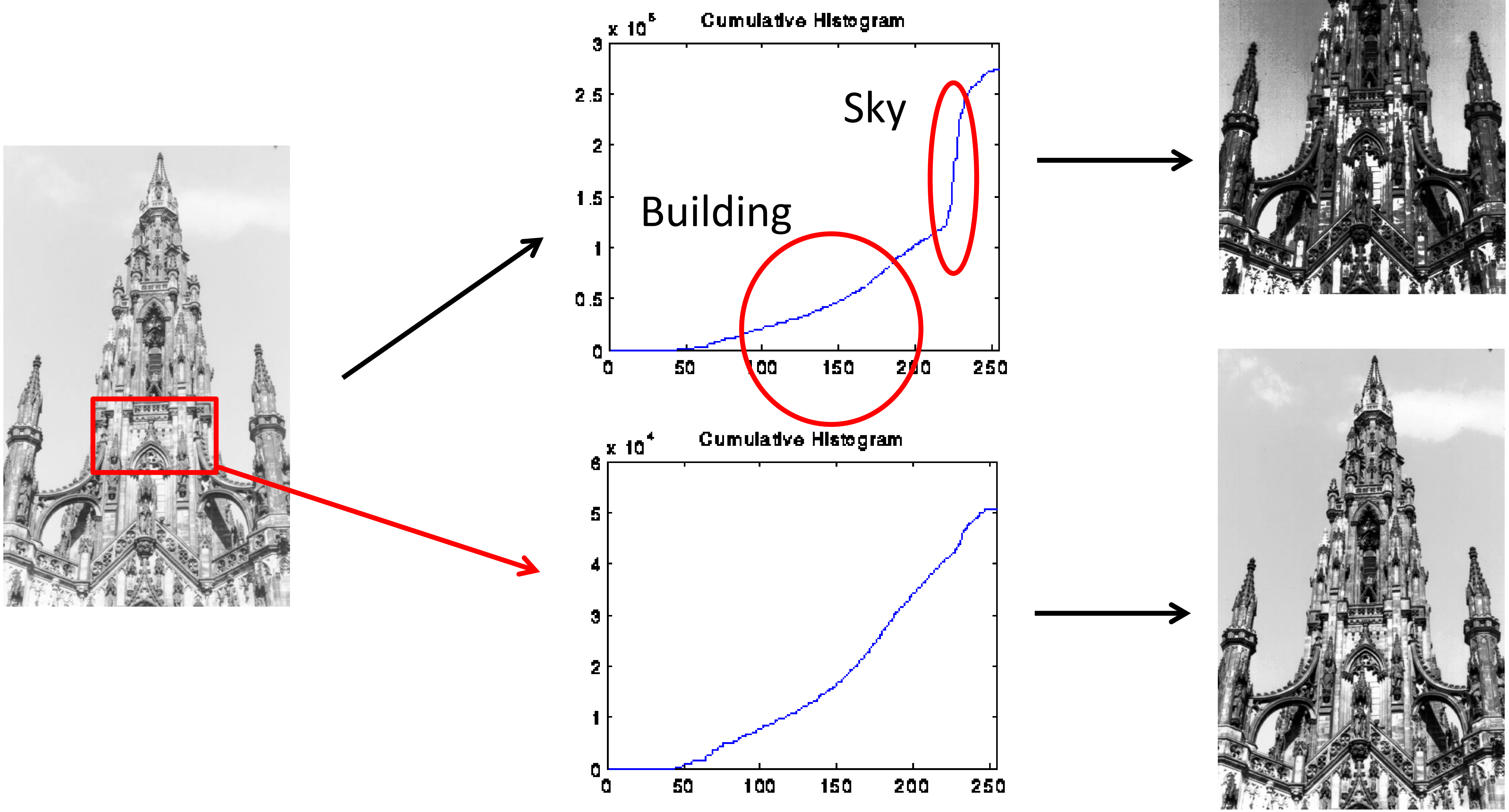
- f(i) is mapping from old gray level to new gray level --> replace gray level i in original image with f(i) in new image

Histogram Equalization - Result



Question

- Why does histogram equalization not work well in this example?



Histogram Equalization - Summary

- Resulting histograms are not really flat (because discrete distribution)
- Peaks in the histogram are preserved
- Histograms are non-linearly distorted. Compression where there are few gray levels, stretching where there are many gray levels
- N input gray levels can be assigned to 1 output gray level (reduction of image information, otherwise no problem)
- Generalization: Histogram Shaping (Histogram of arbitrary shapes)
- Python: `skimage.exposure.equalize_hist(im)`

Histrogram Applications

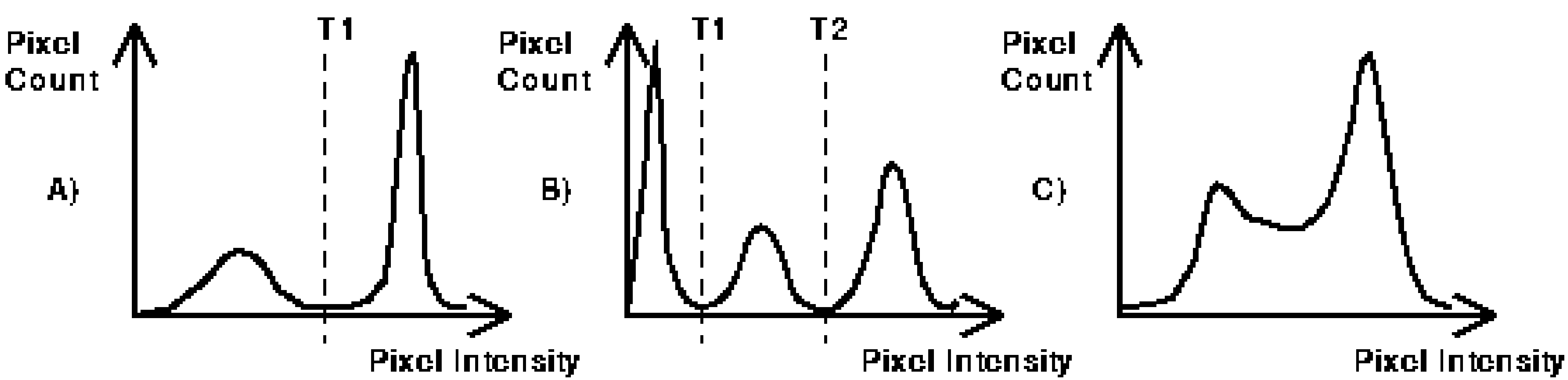
- Contrast enhancement
- **Threshold search for image segmentation**
- Search for similar images: Content-Based Image Retrieval (CBIR)

Thresholding

- Conversion of gray scale image into binary image
- Quantization of gray levels to 2 values → 0 and 1 using threshold T:

$$g(n) = \begin{cases} 1 : f(n) \geq T \\ 0 : f(n) < T \end{cases}$$

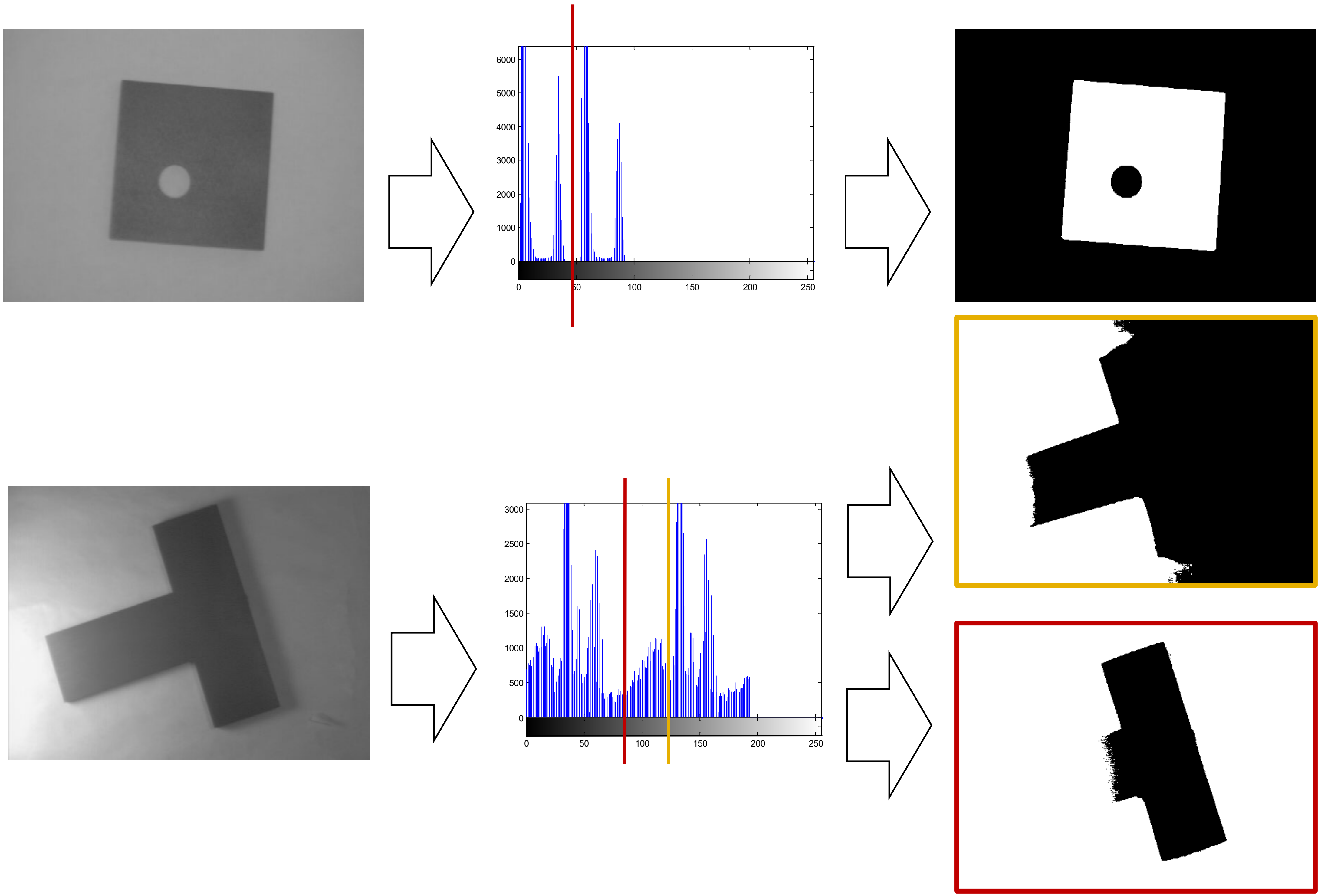
- **How to choose a reasonable threshold value?** Histogram can help if foreground and background have different gray level distributions:



bimodal 2 thresholds ambiguous

Thresholding

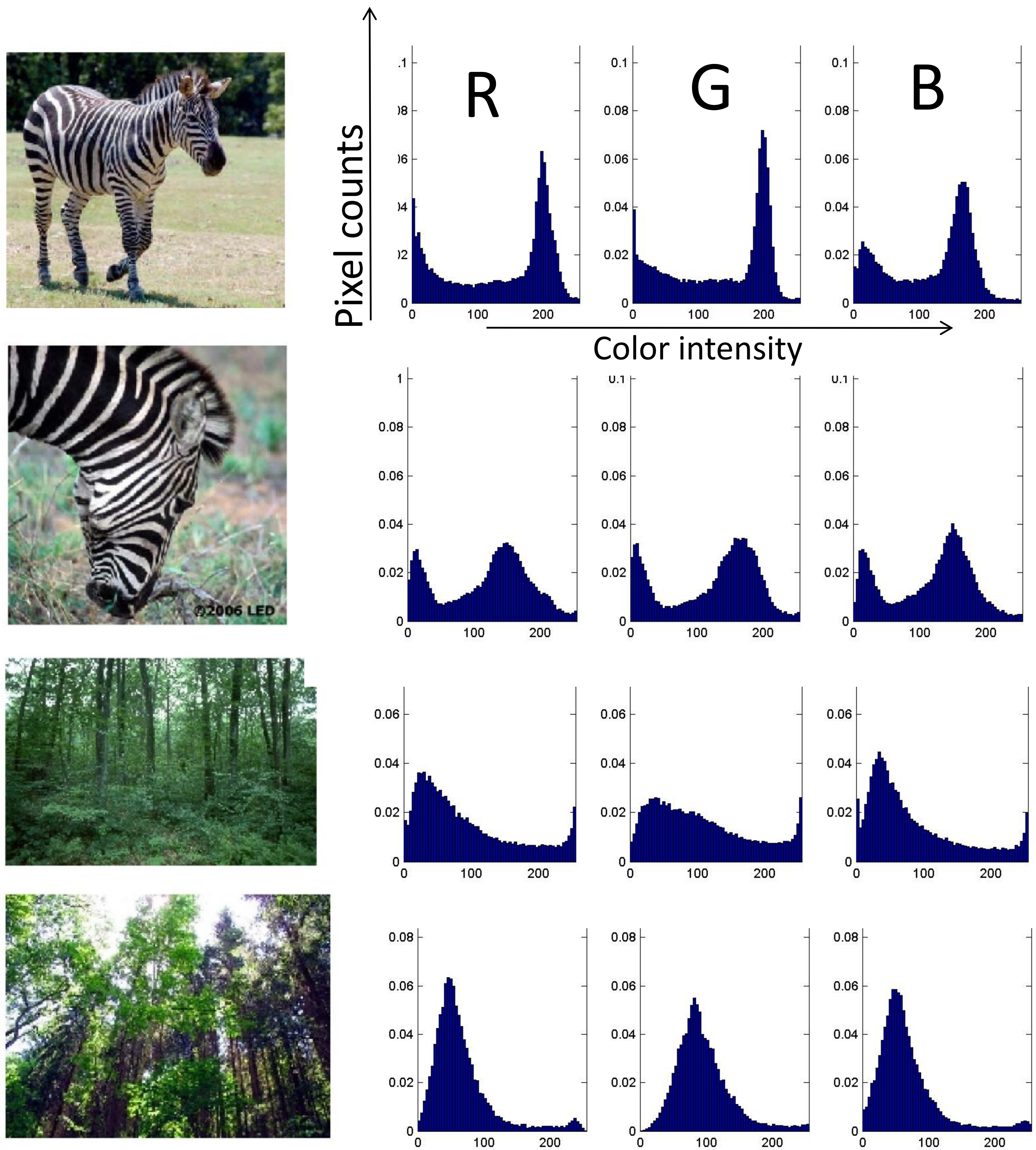
- Application: Separate foreground and background



Histrogram Applications

- Contrast enhancement
- Threshold search for image segmentation
- **Search for similar images: Content-Based Image Retrieval (CBIR)**

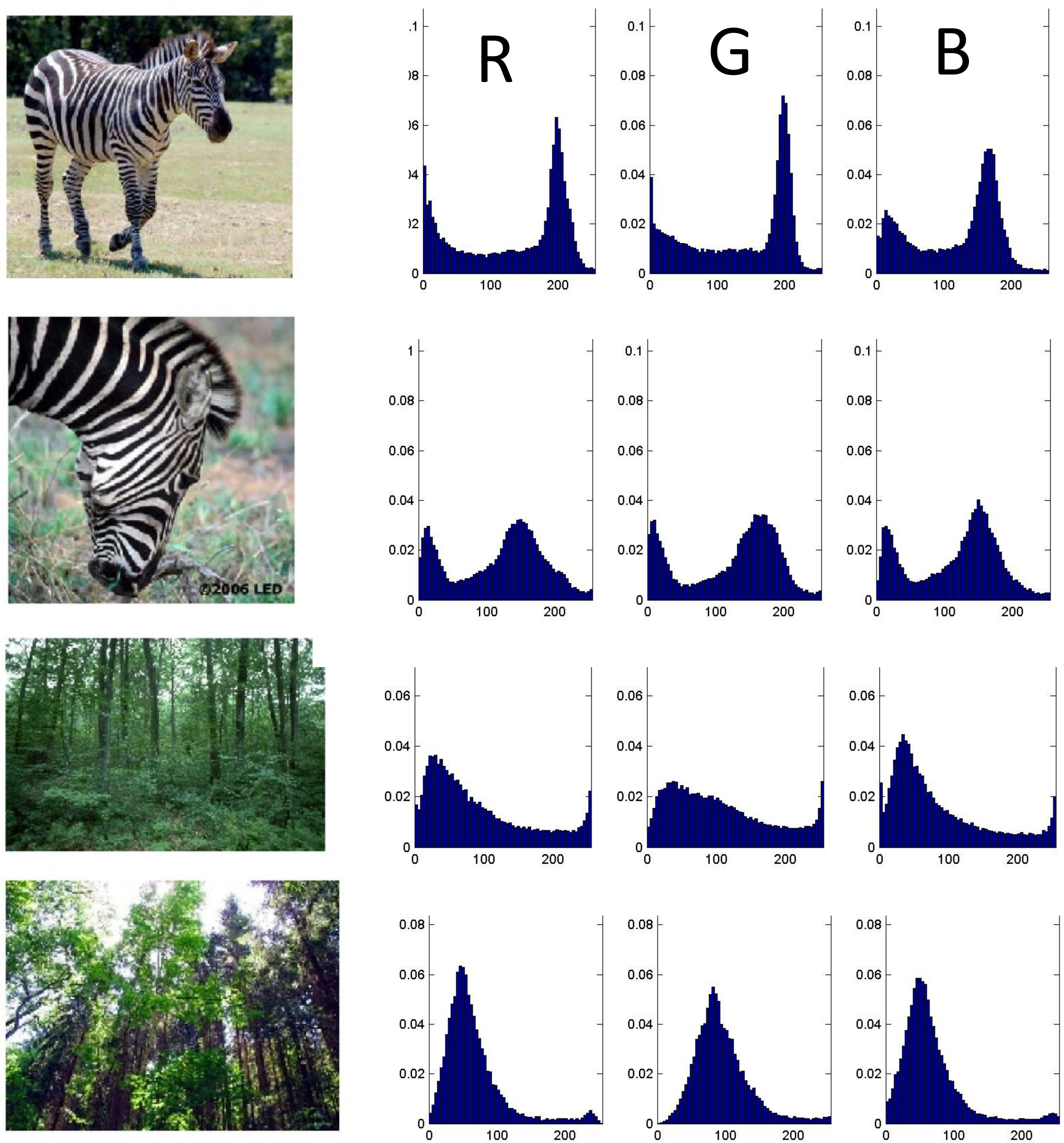
Color as a low-level cue for CBIR



- Color histograms: Use distribution of colors to describe image
- No spatial info – invariant to translation, rotation, scale

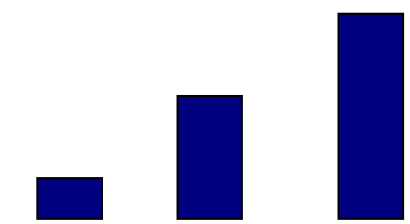
Slide credit: Travor Darrel

Color as a low-level cue for CBIR

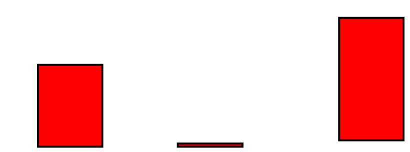


Given two histogram vectors, sum the minimum counts per bin:

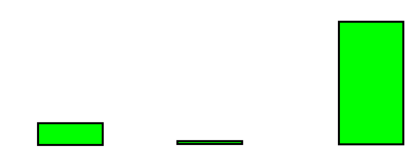
$$I(x,y) = \sum_{i=1}^{n} \min(x_i, y_i)$$



x= [1, 3, 5]



y= [2, 0, 3]



[1, 0, 3]

$\sum_i \min(x_i, y_i) = 4$

Slide credit: Travor Darrel

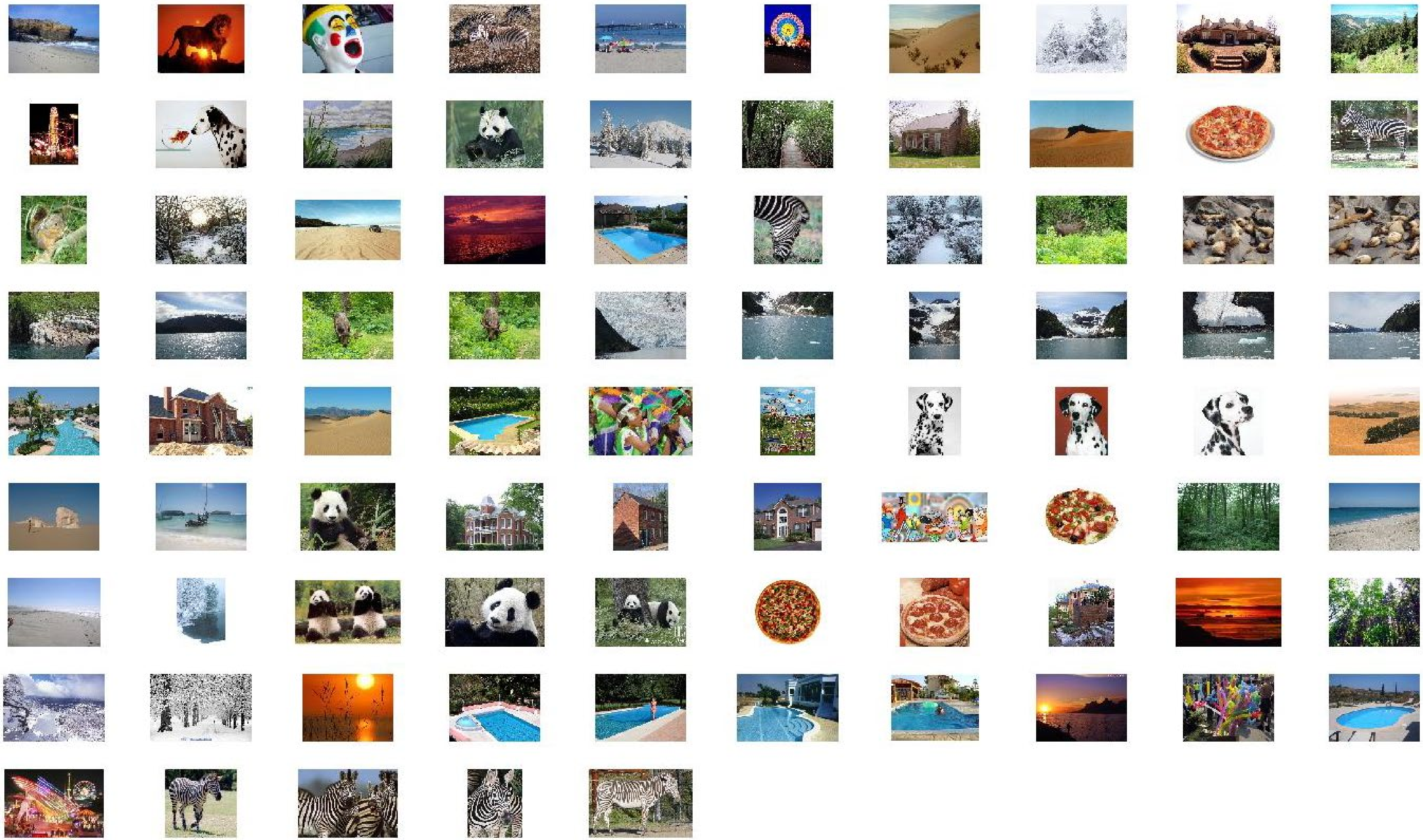
Color-based image retrieval

- Given collection (database) of images:
  - Extract and store one color histogram per image

- Given new query image:
  - Extract its color histogram
  - For each database image: Compute intersection between query histogram and database histogram
  - Sort intersection values (highest score = most similar)
  - Rank database items relative to query based on this sorted order

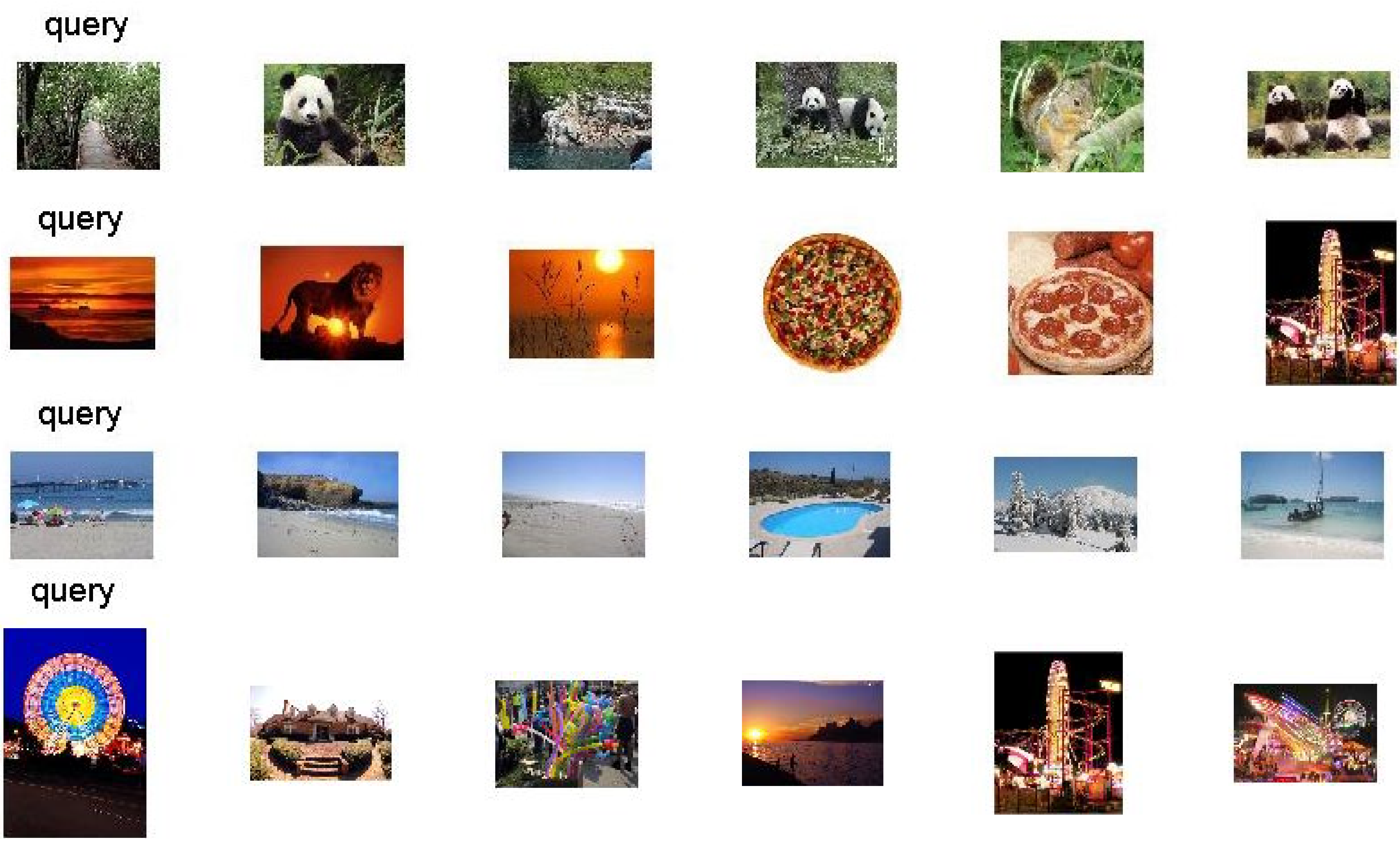
Slide credit: Travor Darrel

Example database



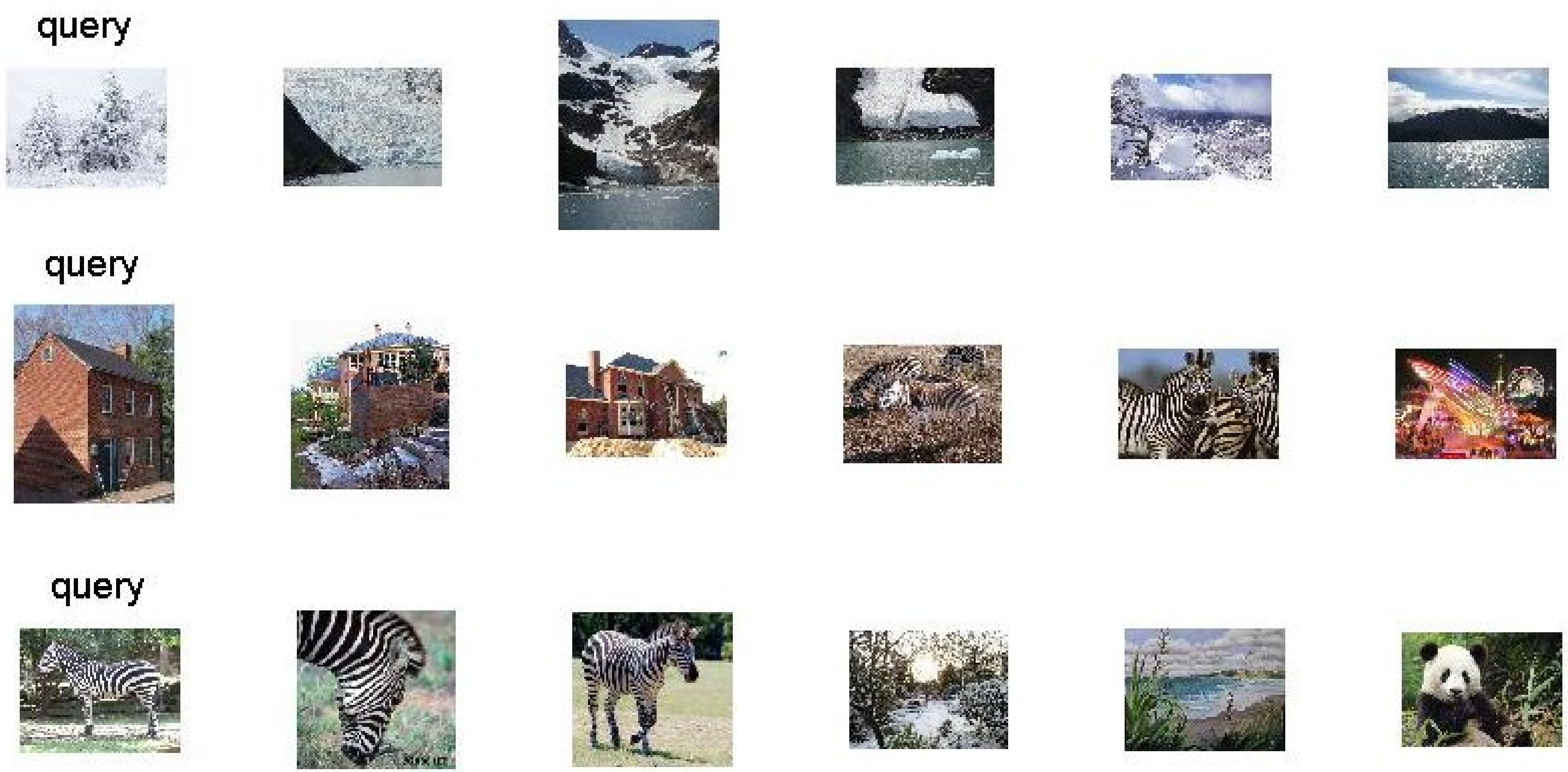
Slide credit: Travor Darrel

Example Retrieval Results



Slide credit: Travor Darrel

Example Retrieval Results



Slide credit: Travor Darrel

Final Remark on Histograms

- Q: What happens if we reshuffle all pixels within the images?

- A: Its histogram won't change!
  → Point-wise processing unaffected
- Need to measure properties relative to small ***neighborhoods*** of pixels

Slide credit: Travor Darrel

Roadmap

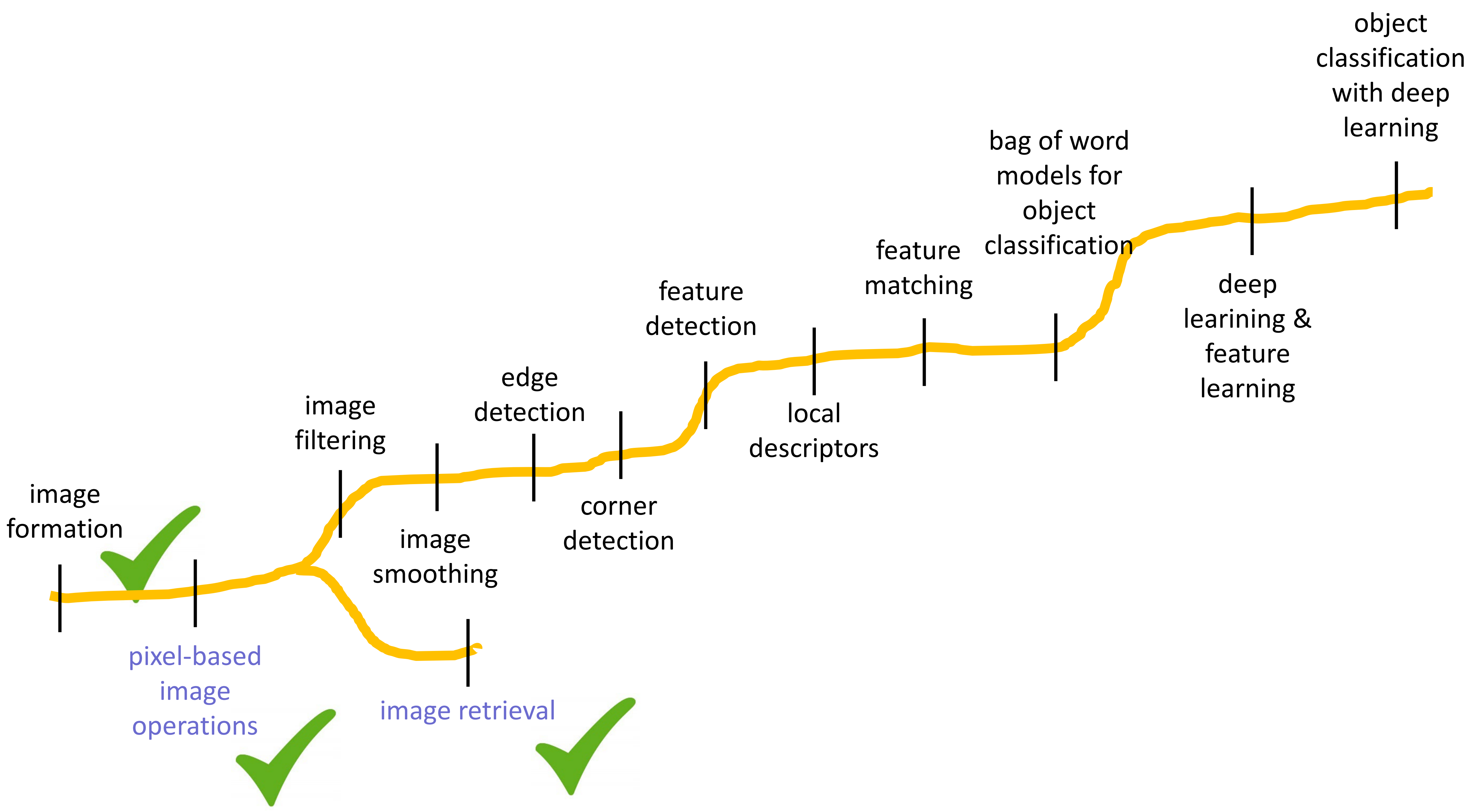


image formation
pixel-based image operations
image filtering
image smoothing
image retrieval
edge detection
corner detection
feature detection
local descriptors
feature matching
bag of word models for object classification
deep learining & feature learning
object classification with deep learning