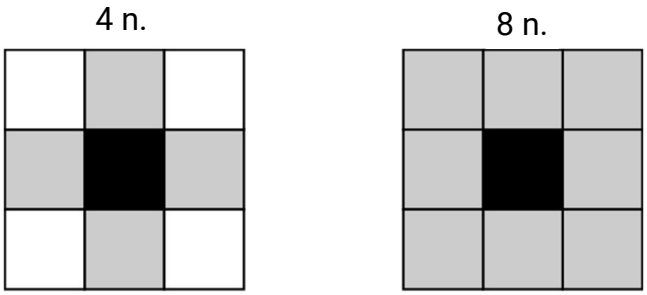
Image basics

Manuel J. Marín-Jiménez (*Univ. of Córdoba*)

@mjmarinj

Pixel connectivity

4 neighbours vs 8 neighbours



Pixel-wise operations

Mathematical operations applied to all pixels in the image

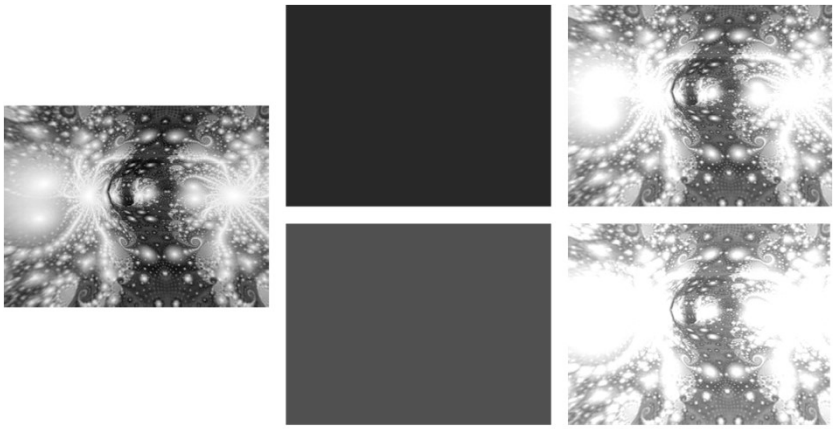
$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} + K$$

$$\text{sqrt}\left(\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}\right)$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

Examples

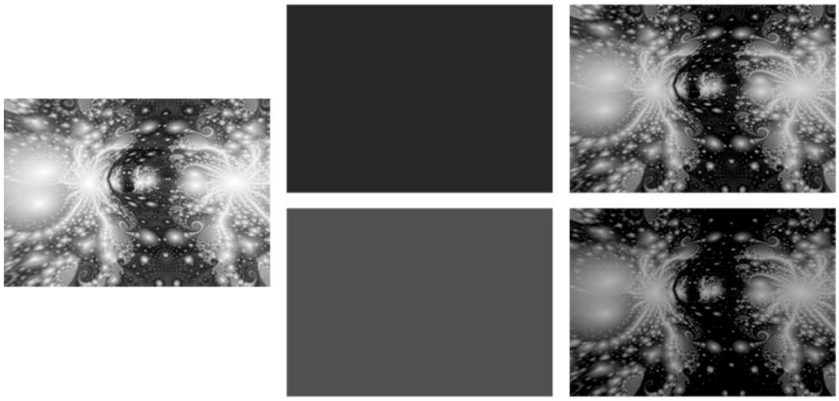
Sum



5

Examples

Subtract



6

Exercise

Write a program in OpenCV that adds a constant value to an image.

- Use the command-line arguments to define the value.
- Control data overflow.
- Show the resulting image.
- Save the image to a file with name given by the user.

7

Examples

Invert = 255 - I

8

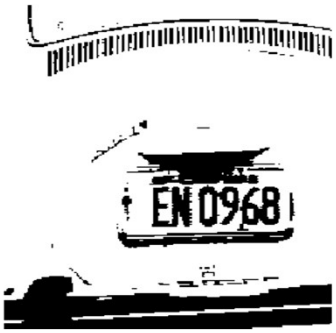
Exercise

Write a program in OpenCV that inverts the values of the input image.

- The user can choose to invert a single channel or all.
- Show the resulting image.
- Save the image to a file with name given by the user.

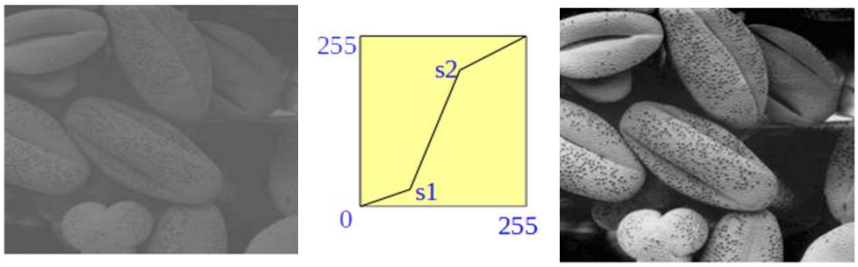
Thresholding

I' = I > t ?



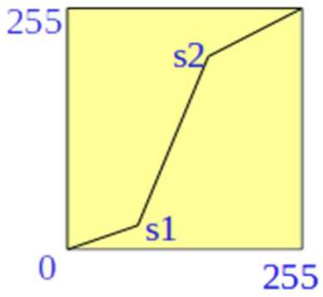
Enhancement

- Linear contrast enhancement (using *transfer function*)



Enhancement

- Linear contrast enhancement (using *transfer function*)

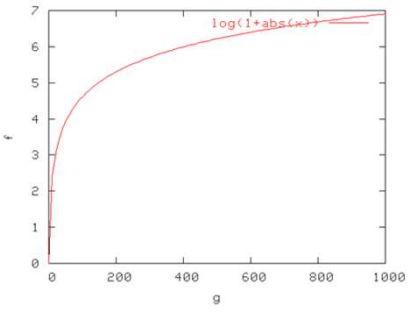


- *f1*: line (0,0) – (s1, v1)
- *f2*: line (s1,v1) – (s2, v2)
- *f3*: line (s2,v2) – (255,255)

Example: s1 = (50, 10)
f1: (0,0) → (50,10)
m = (10-0) / (50-0) = 1/5
n = 0 – 1/5*0

Enhancement

- Logarithmic enhancement → *log(1+abs(p))*



- After applying the *log* function, are the values in the correct range?

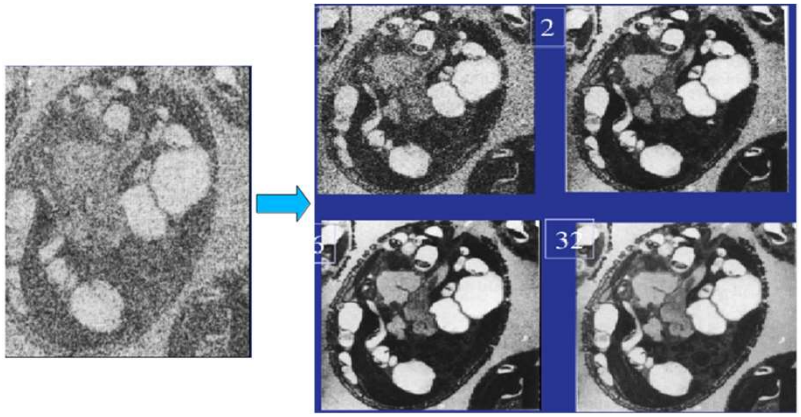
Exercise

Write independent programs in OpenCV to apply the previous operations on images.

- The user can choose the parameters, if needed.
- Show the resulting image.
- Save the image to a file with name given by the user.

Multiple images

- Image average → noise removal
  - Set of images needed



15

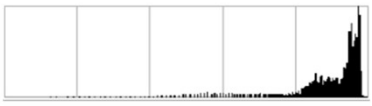
Multiple images

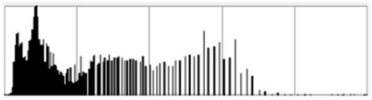
- Image average → background removal
  - Create a background model
  - Difference → foreground detection

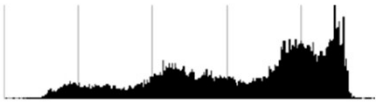
16

Histograms

- Distribution of intensities/colours
  - Frequency



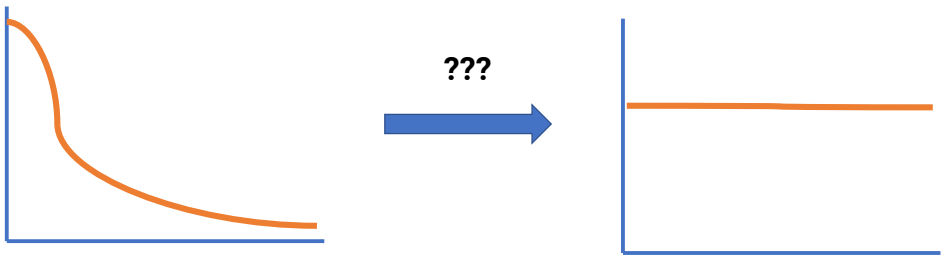




17

Histograms

- Distribution of intensities/colours
  - Frequency

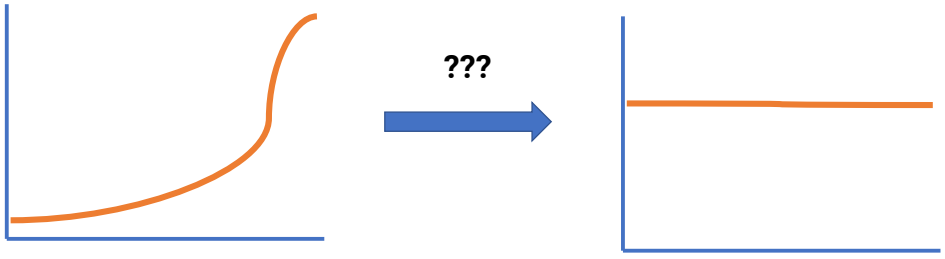


17

18

Histograms

- Distribution of intensities/colours
  - Frequency



18

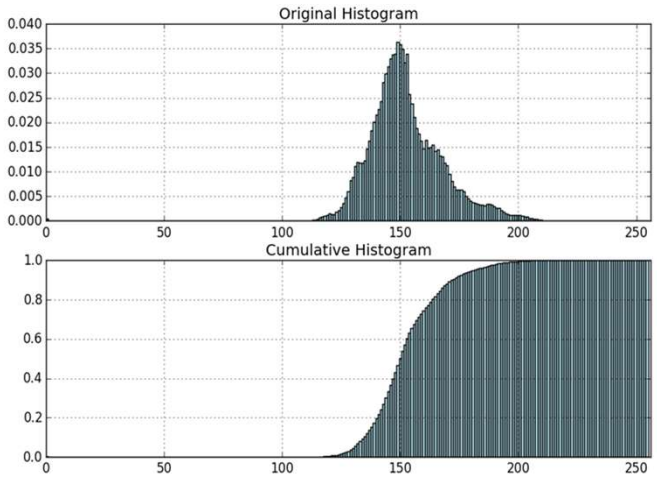
Exercise

Write a program to compute the histogram of intensities of your input image:

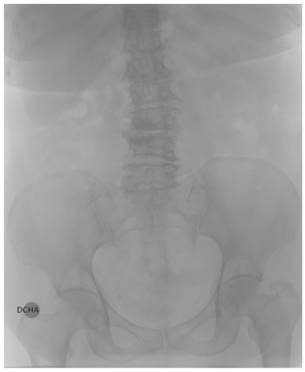
- Create a function to compute the absolute frequency
- The user can select either the colour channel or grey
- Show the frequencies in the terminal

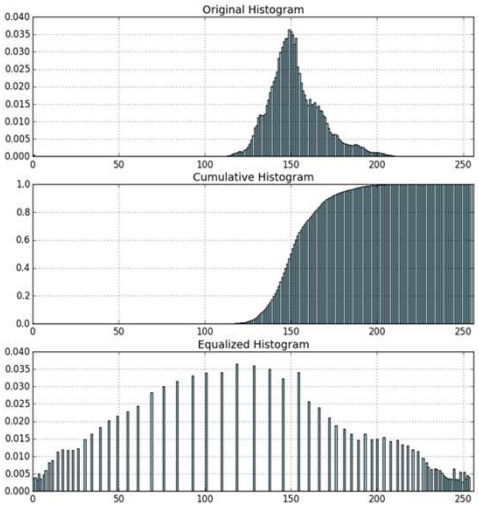
Cumulative Histogram

- c(i) = h(0) + h(1) + h(2) + … + h(i-1) + h(i)



Histogram Equalization





- Create a mapping with the Cumulative hist. → I' = c(I)

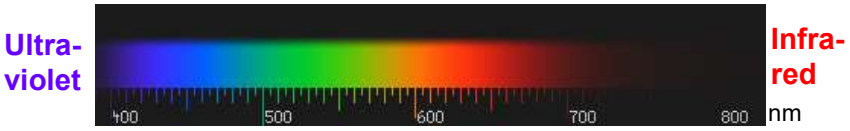
Exercise

Extend your previous program:

- Create a function to compute the cumulative histogram
- Normalize the cumulative histogram to [0,255]
- Verify that your code is correct

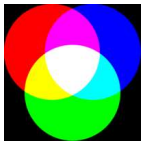
What is *colour*?

- It is a human perception
- Objects do not have colour
  - Some light frequencies are reflected and some absorbed.

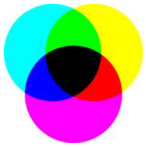


- Two models:

Additive
- Light

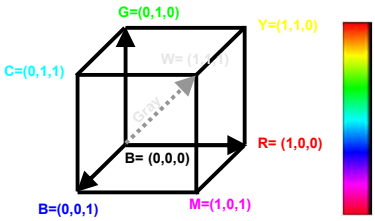


Subtractive
- Pigments

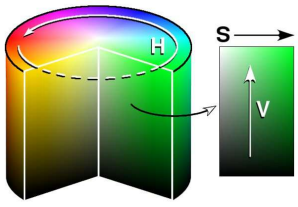


Colour spaces

- RGB: red, green, blue
  - The most used



- HSV/HSL: hue, saturation, value/lightness
  - More human-readable



- CIE-Lab, CMY,...

Interactive: https://color.lukas-stratmann.com/color-systems.html

RGB-Grayscale conversion

- The **lightness** method averages most prominent and least prominent colours: **(max(R, G, B) + min(R, G, B)) / 2**
- The **average** method simply averages the values: **(R + G + B) / 3**
- The **luminosity** method → weighted average to account for human perception. We're more sensitive to green than other colours, so green is weighted most heavily. Computation: **0.21 R + 0.72 G + 0.07 B**

lightness | average | luminosity

Exercise

Investigate and practise with the colour conversion functions that OpenCV provides.

Image basics

Manuel J. Marín-Jiménez (*Univ. of Córdoba*)

@mjmarinj

27