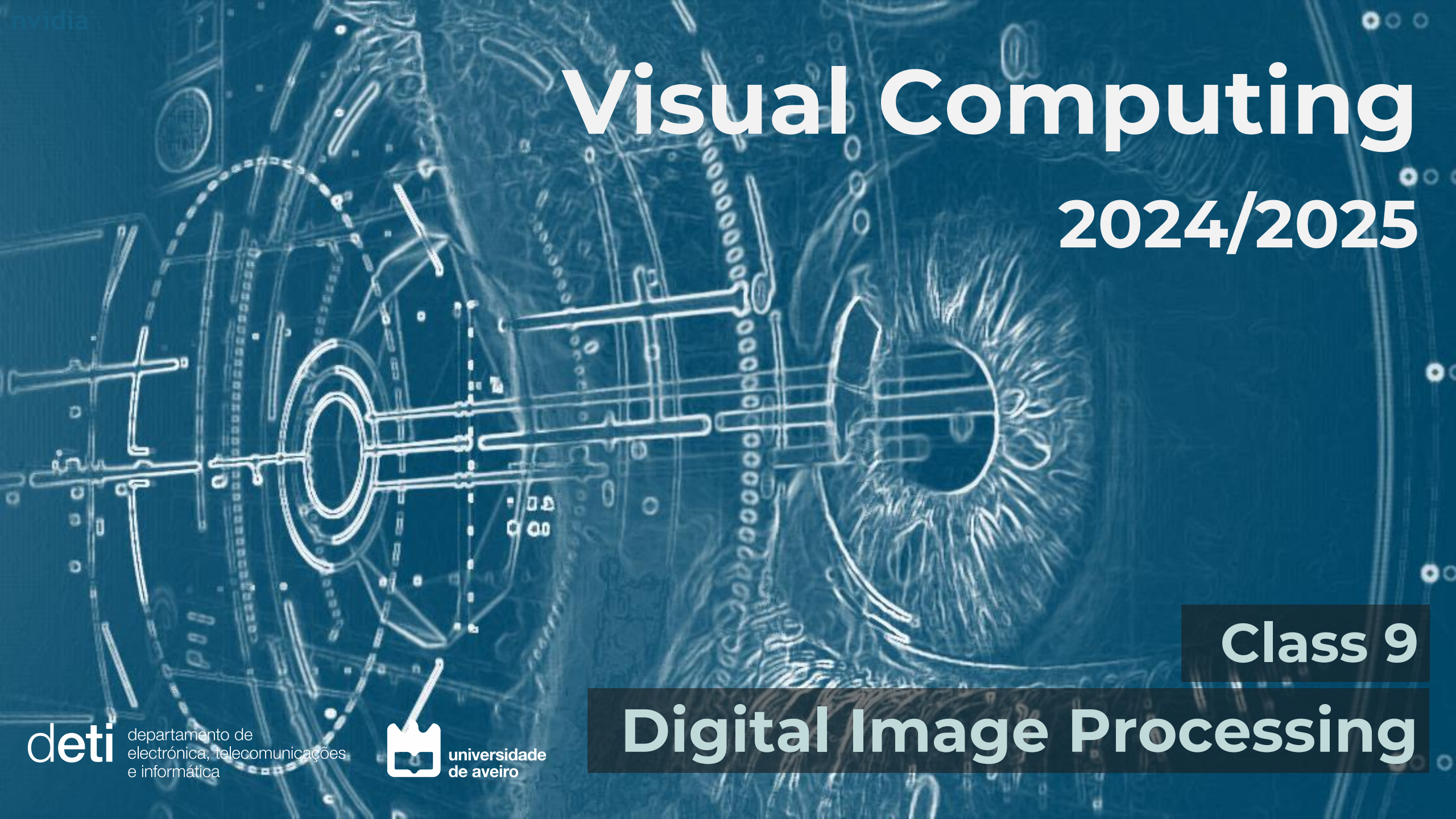


Visual Computing

2024/2025

Class 9

Digital Image Processing

deti departamento de electrónica, telecomunicações e informática

universidade de aveiro

Overview

- Digital Images
- The Processing Pipeline
- Pixel Operations
- Histogram-based Operations
- Filters

Digital Image Processing

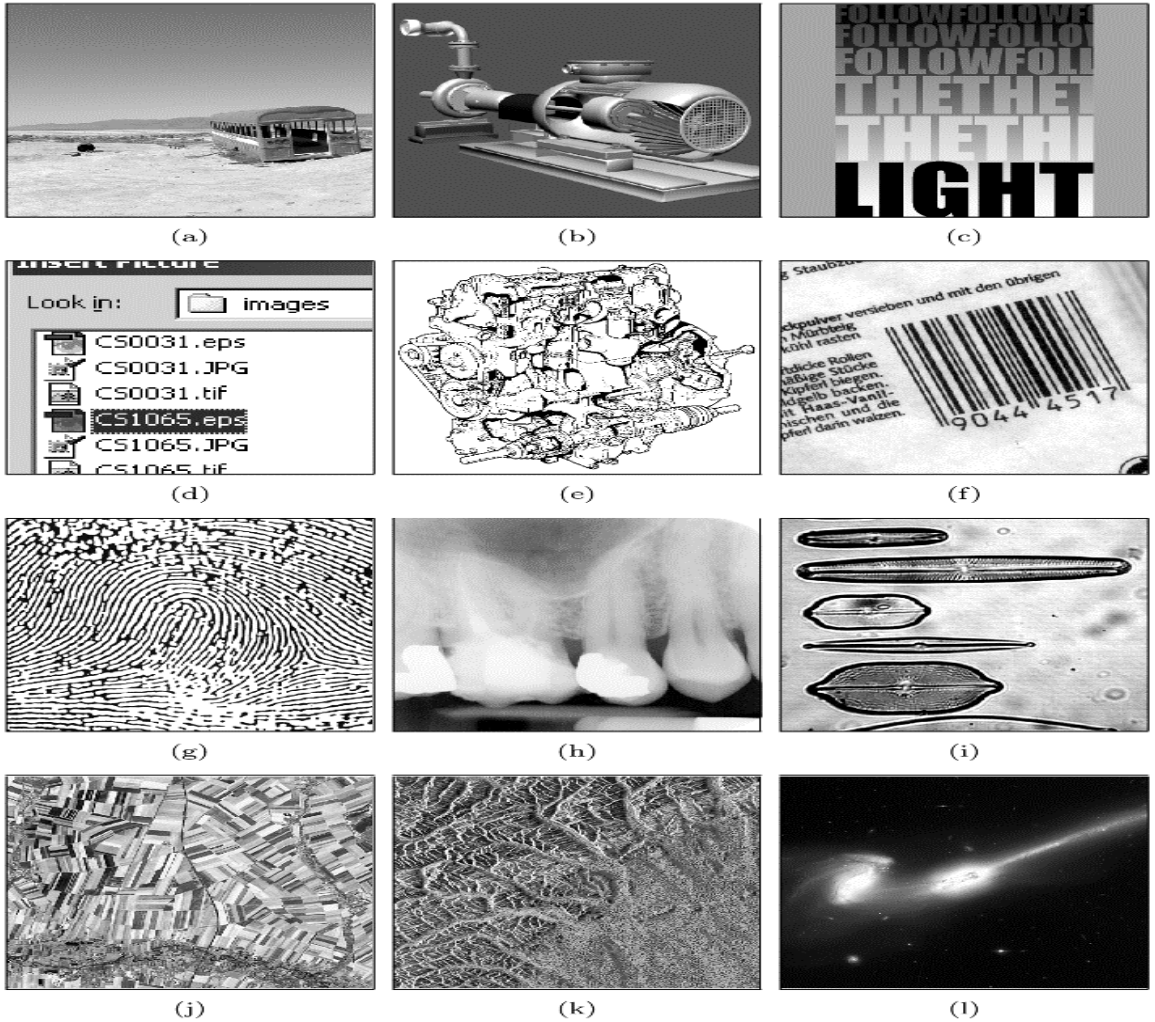
Digital Image Processing, as a scientific area, started when it became possible to use a computer to process images as collections of **pixels** with associated **numerical values**

A traditional goal is to **improve** an image to make it more amenable to processing:

- By a **human observer**
- By some **automatic analysis** process

Digital Images

Wide range of applications



[Burger & Burge]

Digital Image Acquisition

- Continuous light distribution is **spatially sampled**
- Still image created by **time sampling** that discrete distribution
- Resulting values are **quantized** to a finite set of numerical values

Digital Image Acquisition

$F(x, y)$

$\longrightarrow$

$$
\begin{array}{cccccccccc}
148 & 123 & 52 & 107 & 123 & 162 & 172 & 123 & 64 & 89 \cdots \\
147 & 130 & 92 & 95 & 98 & 130 & 171 & 155 & 169 & 163 \cdots \\
141 & 118 & 121 & 148 & 117 & 107 & 144 & 137 & 136 & 134 \cdots \\
82 & 106 & 93 & 172 & 149 & 131 & 138 & 114 & 113 & 129 \cdots \\
57 & 101 & 72 & 54 & 109 & 111 & 104 & 135 & 106 & 125 \cdots \\
138 & 135 & 114 & 82 & 121 & 110 & 34 & 76 & 101 & 111 \cdots \\
138 & 102 & 128 & 159 & 168 & 147 & 116 & 129 & 124 & 117 \cdots \\
113 & 89 & 89 & 109 & 106 & 126 & 114 & 150 & 164 & 145 \cdots \\
120 & 121 & 123 & 87 & 85 & 70 & 119 & 64 & 79 & 127 \cdots \\
145 & 141 & 143 & 134 & 111 & 124 & 117 & 113 & 64 & 112 \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

$I(u, v)$

Greyscale Images

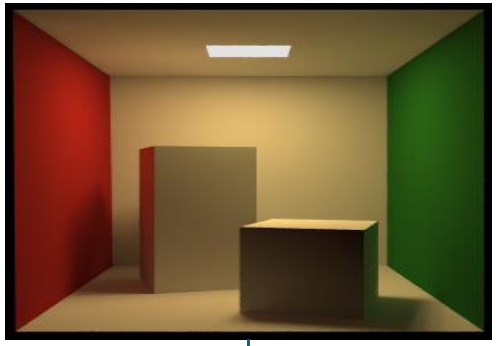
Each pixel stores the brightness / intensity

No color (one chanel)

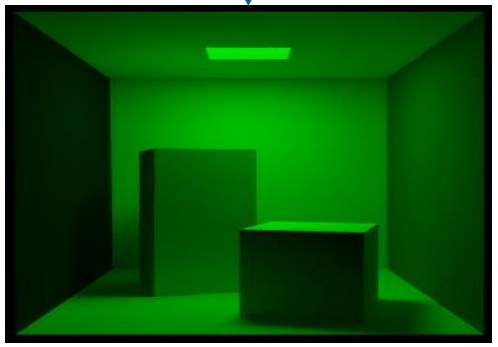
Leonid Afremov

Image Channels

Original RGB image
3 samples per pixel



Red channel
1 sample per pixel



Green channel
1 sample per pixel

Blue channel
1 sample per pixel

[Andy Van Dam]

Convert from Color to Greyscale

?

Leonid Afremov

Convert from Colour to Greyscale

Average method

- **Grey = (Red + Green + Blue) / 3**

But, remember that the human eye perceives colour intensities differently

Weighted method

- **Grey = 0.299 * Red + 0.587 * Green + 0.114 * Blue**
- Can you guess how sensitive hour eyes are to each color component?

Pixel Values

Grayscale (Intensity Images):

Chan.	*Bits/Pix.*	*Range*	*Use*
1	1	0...1	Binary image: document, illustration, fax
1	8	0...255	Universal: photo, scan, print
1	12	0...4095	High quality: photo, scan, print
1	14	0...16383	Professional: photo, scan, print
1	16	0...65535	Highest quality: medicine, astronomy

Color Images:

Chan.	*Bits/Pix.*	*Range*	*Use*
3	24	$[0...255]^3$	RGB, universal: photo, scan, print
3	36	$[0...4095]^3$	RGB, high quality: photo, scan, print
3	42	$[0...16383]^3$	RGB, professional: photo, scan, print
4	32	$[0...255]^4$	CMYK, digital prepress

Special Images:

Chan.	*Bits/Pix.*	*Range*	*Use*
1	16	$-32768...32767$	Whole numbers pos./neg., increased range
1	32	$\pm 3.4 \cdot 10^{38}$	Floating point: medicine, astronomy
1	64	$\pm 1.8 \cdot 10^{308}$	Floating point: internal processing

[Burger & Burge]

Useful Concepts

Contrast

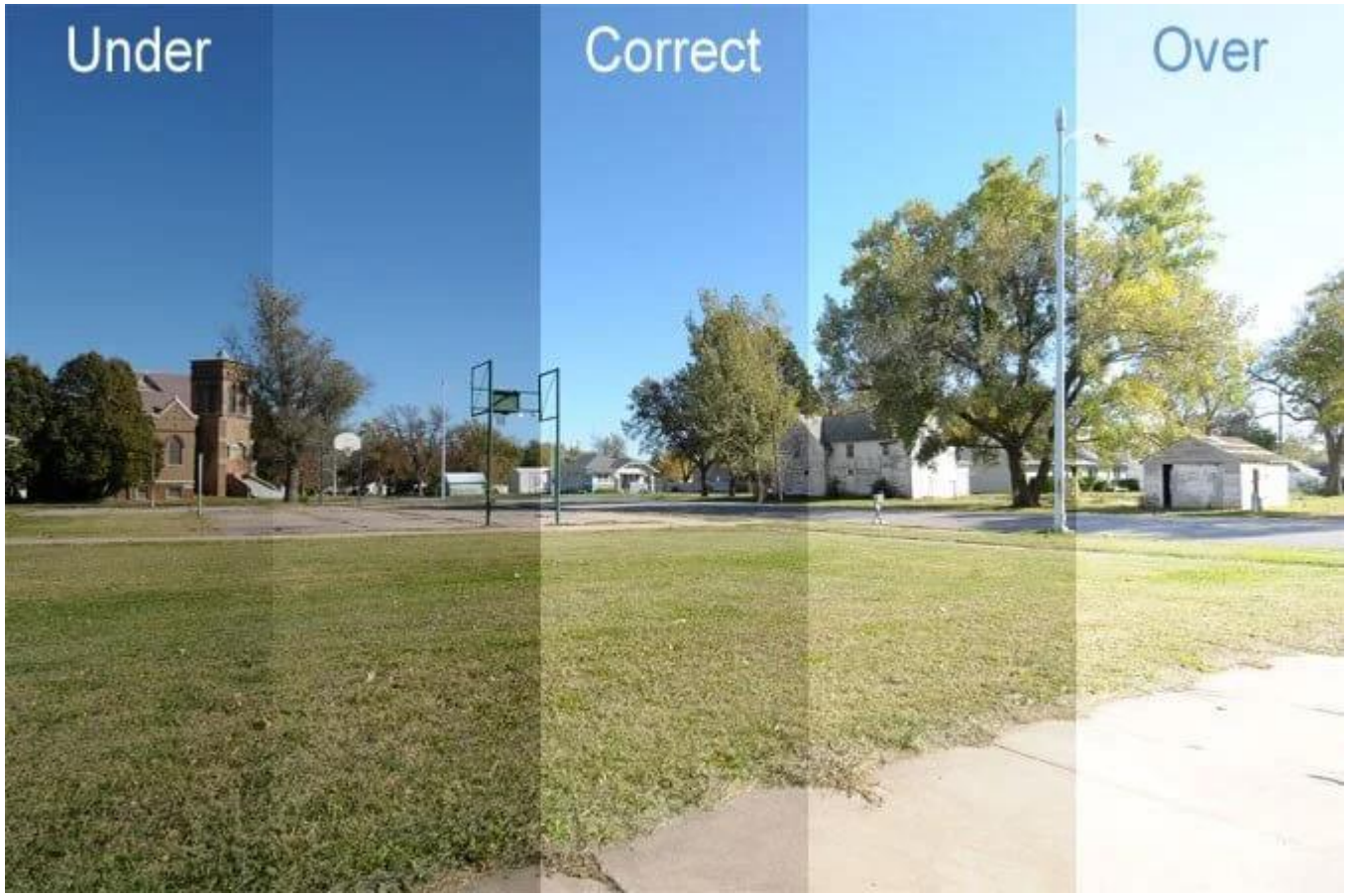
- **Contrast** refers to the amount of noticeable difference among regions of an image
- **High-contrast** images have a larger variation in luminosity than low-contrast images
- Hence, in **low-contrast** images, regions are more hardly distinguishable

Low Contrast Image

High Contrast Image

Image Exposure

- **Overexposed**
  - Details of the image are lost since the **sensor receives much more light** that what it can handle
- **Underexposed**
  - Details are lost since the **sensor is not able to detect** the amount of projected light



www.studiobinder.com


Image
Potato Processing

Image Processing Pipeline

Acquisition, storage, loading

- acquire image,
- store it on disk
- load it for showing or processing

Pre-processing

- Transform, convert to a different format
- Enhance quality (e.g., sharpening)
- Remove noise

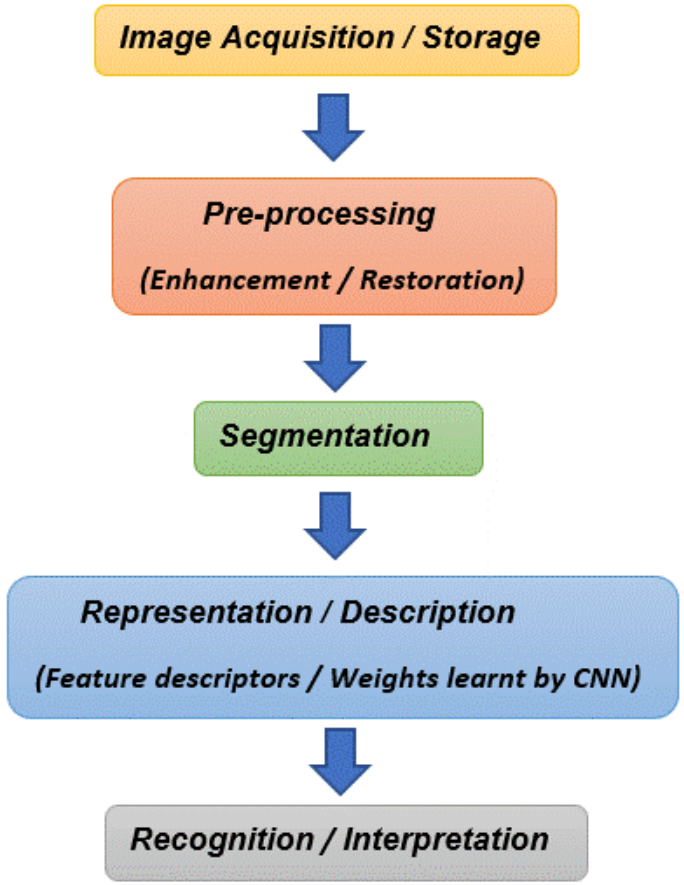


Image Processing Pipeline

- **Segmentation**
  - Extract regions/object of interest from the image
- **Information Extraction / Description**
  - Compute properties / descriptors of the image
- **Recognition / Interpretation**:
  - Image classification
  - Object recognition

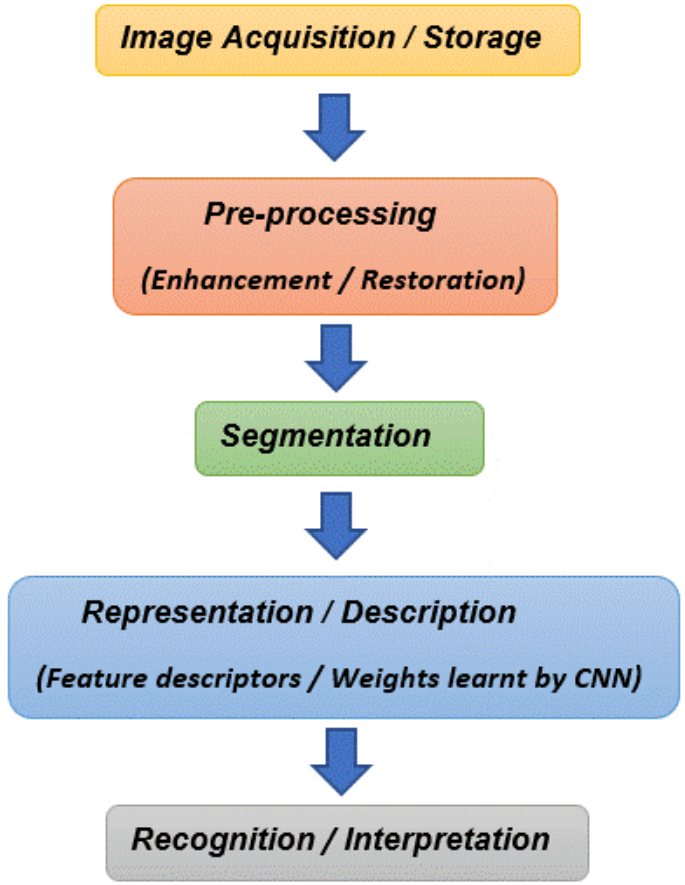


Image Acquisition

- **Synthesis** – images created by a computer
  - 2D painting / 3D rendering / procedural texturing
- **Capture** – images from the “real world”
  - Sampling of an analog signal
  - Digital camera
  - Frames from video
  - Data from sensors (optical, thermal, radiation, …)
  - …

Preprocessing

- Fit image to a given **tone**, **size**, **shape**, etc.
- Match image to a desired feature or to other images
- Make a set of **dissimilar** images appear **similar**
  - E.g., to be composited later
- Make similar parts of an image appear dissimilar
  - E.g., **contrast enhancement**

Original

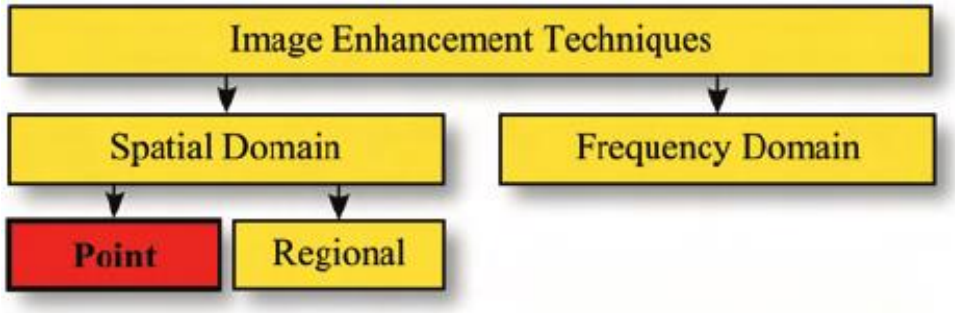
Adjusted grayscale curve

Preprocessing

- We will consider methods for pixel processing in the **space-domain** (they operate on pixel intensity, colour, or position)

- There are other methods that operate on an image after it has been transformed to other domains (e.g., **frequency-domain**)

- **We will keep to the space-domain** (at least today… :D)



Pixel Operations

In **point** (pixel) processing, a **single input** is processed to obtain a **single output** sample

Pixel intensity is modified through some operation, e.g., adding, subtracting, or multiplying

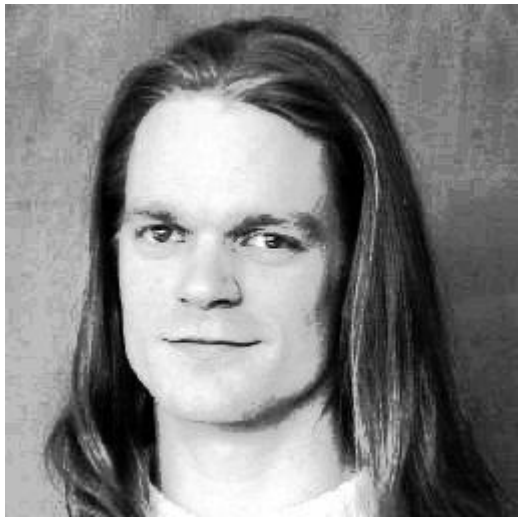
Rescaling (Contrast and Brightness)

Rescaling is the operation by which we alter the contrast and/or brightness of an image

$$S_{\text{output}} = \alpha \cdot S_{\text{input}} + \beta.$$

Alpha is the **gain:** controls overall **contrast**

- > 1, image gets brighter and **more contrasted**
  - E.g., original: 5, 10, diff = 5; converted 5*2, 10*2, diff = 10 (diff got bigger)
- < 1, image gets darker and **less contrasted**
  - E.g., original: 5, 10, diff = 5; converted: 5*.5, 10*.5, diff = ~3 (diff got smaller)



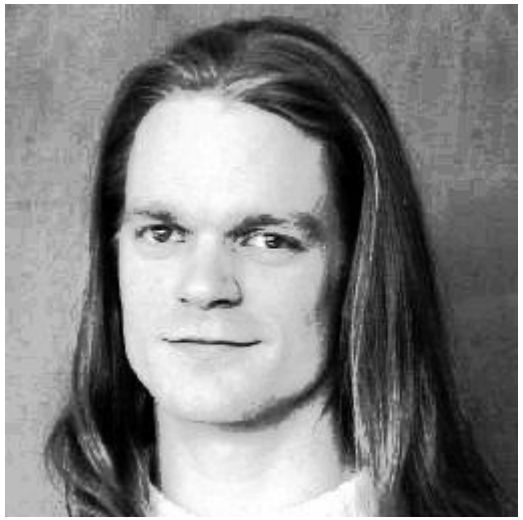
Rescaling (Contrast and Brightness)

Rescaling is the operation by which we alter the contrast and/or brightness of an image

$$S_{\text{output}} = \alpha \cdot S_{\text{input}} + \beta.$$

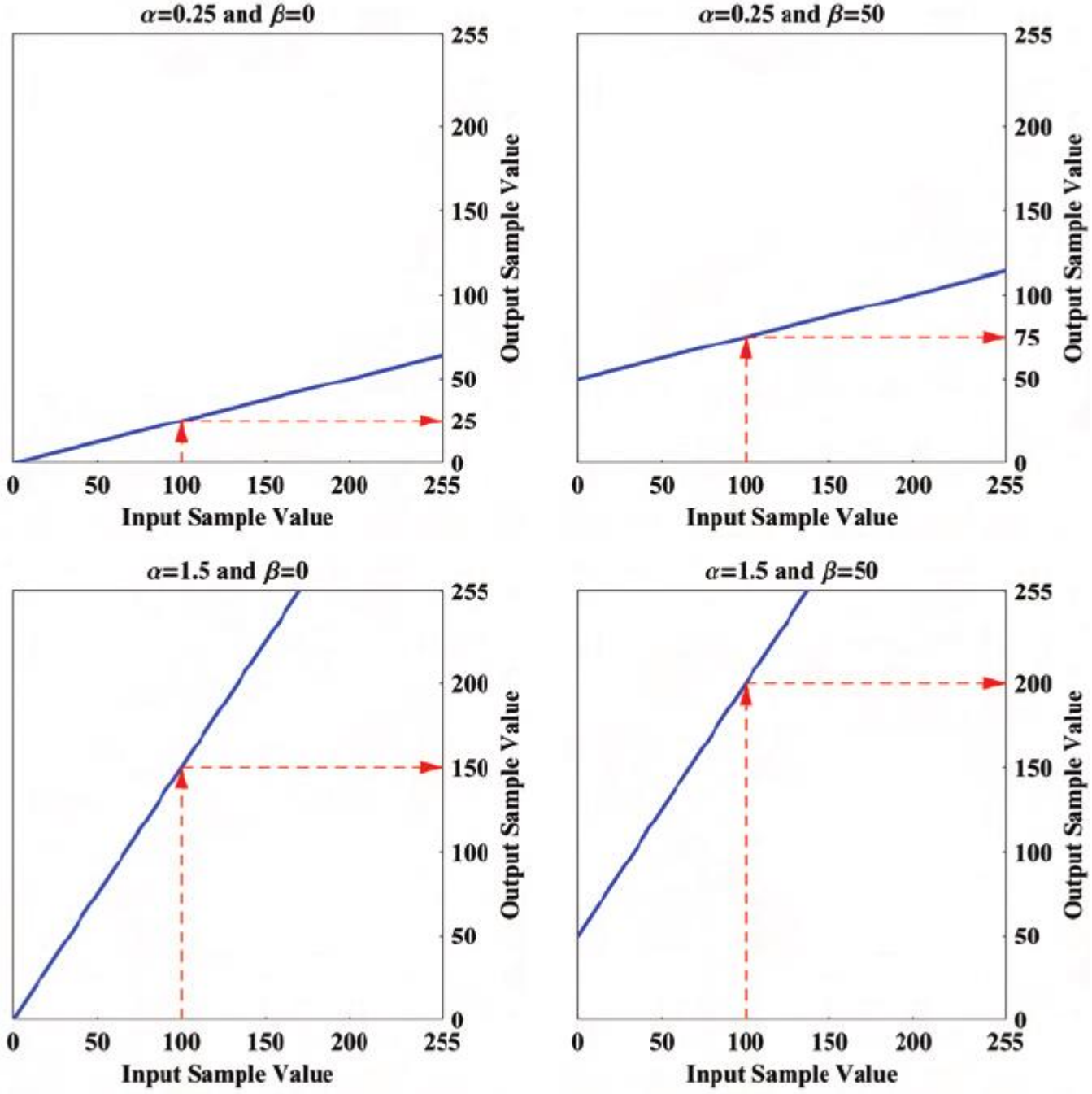
Beta is the **bias:** controls overall **brightness**

- > 0, image gets **brighter**
  - E.g., 10 + 10 = 20
- < 0, image gets **darker**
  - E.g., 10 – 5 = 5



Rescaling

- **Examples** of different values for alpha and beta and how they transform the inputs
- Note how **alpha > 1.0** extends the range of values (**more contrast**)
- Note how **beta** keeps contrast, but **changes mean intensity**



Rescaling (Contrast and Brightness)

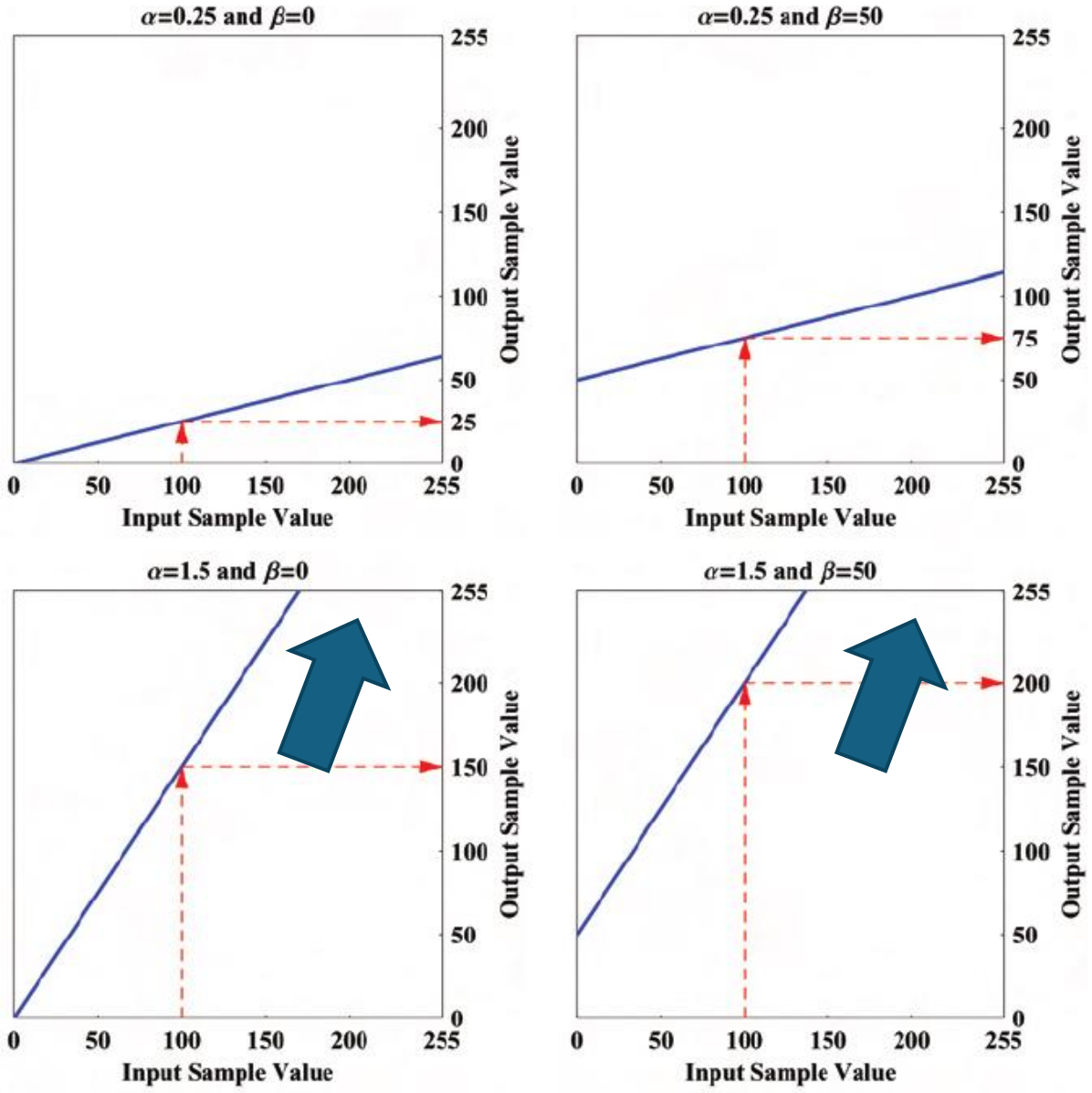
- alpha= 1.3, beta = 40

Notice clipping

Clamping

- Applied after processing the image to **ensure that its pixel values are kept within what is admissible**, e.g., [0..255] for 8 bits, or [0.0... 1.0] for float representations.

$$clamp(x, min, max) = \begin{cases} \min & \text{if } \lfloor x \rfloor \leq \min, \\ \max & \text{if } \lfloor x \rfloor \geq \max, \\ \lfloor x \rfloor & \text{otherwise.} \end{cases}$$



Contrast Stretching

Contrast stretching aims to **maximize the contrast of an image**

Given an image with intensities in the range [*a*, *b*], then:

$$\begin{aligned} \alpha &= \frac{255}{b-a}, \\ \beta &= -\alpha \cdot a \end{aligned}$$

Application Example

- Consider a **thermal camera** making readings of a Labrador on a summer evening
- The outcome ranges from **71.5 to 99.3** degrees Fahrenheit
- We want to visualize it in an image. We:
  - Round values to nearest integer
  - Show them as a greyscale image
- **Nicely done? How would you do it?**

Application Example

Rescale so that:

- 71.5 corresponds to **black**
- 99.3 corresponds to **white**

So, contrast stretching:

$$\begin{aligned} \alpha &= \frac{255}{b-a}, \\ \beta &= -\alpha \cdot a \end{aligned}$$

$$\begin{aligned} \alpha &= \frac{255}{99.3 - 71.5} \\ &= 9.17, \\ \beta &= -\alpha \cdot a \\ &= -9.17 \cdot 71.5 \\ &= -655.85. \end{aligned}$$

Perceptually Uniform Pipeline

- An image travels a long way from acquisition to visualization:
  - **Acquire image with camera**
  - Store into digital form
  - Decode the stored data
  - **Display on one (or more) output devices**

- Ideally, we want that the image seen on the display **is perceptually equivalent** to what was captured by the camera,

but…

Gamma Value

Cameras and displays introduce nonlinear distortions in the images they produce

For a display, this can be described as a **power-law relationship** between the **level of light that is displayed and the amount of light intended for display**:

$$\hat{S}_{\text{displayed}} = \left(\hat{S}_{\text{intended}} \right)^{\gamma}$$

Gamma, in the formula, determines the **amount of distortion** introduced by the output device (1.0 = no distortion)

Computer monitors and TVs have gamma values in the range of [1.2 to 2.5]

Gamma Correction

- Gamma correction aims to **keep a perceptually uniform sample value** throughout an entire imaging pipeline by compensating for this distortion

- The **intended sample is gamma corrected** for the display:

$$\hat{S}_{\text{corrected}} = (\hat{S}_{\text{intended}})^{1/\gamma}$$

When it is shown, then:

$$\begin{aligned}
\hat{S}_{\text{displayed}} &= (\hat{S}_{\text{corrected}})^{\gamma}, \\
\hat{S}_{\text{displayed}} &= \left((\hat{S}_{\text{intended}})^{1/\gamma}\right)^{\gamma} \\
\hat{S}_{\text{displayed}} &= \hat{S}_{\text{intended}}.
\end{aligned}$$

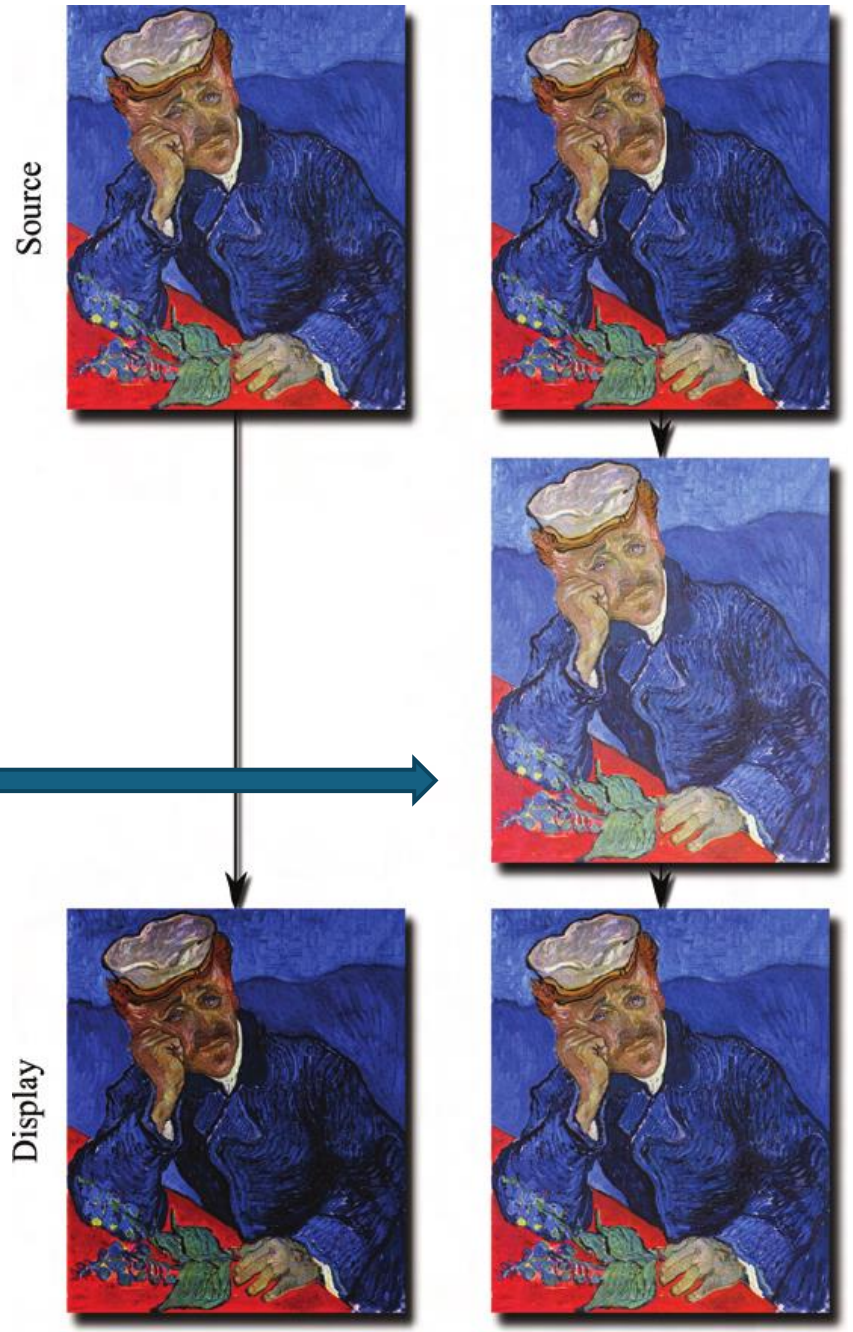


Image Histogram

A histogram shows the number of times each pixel value occurs, in the image

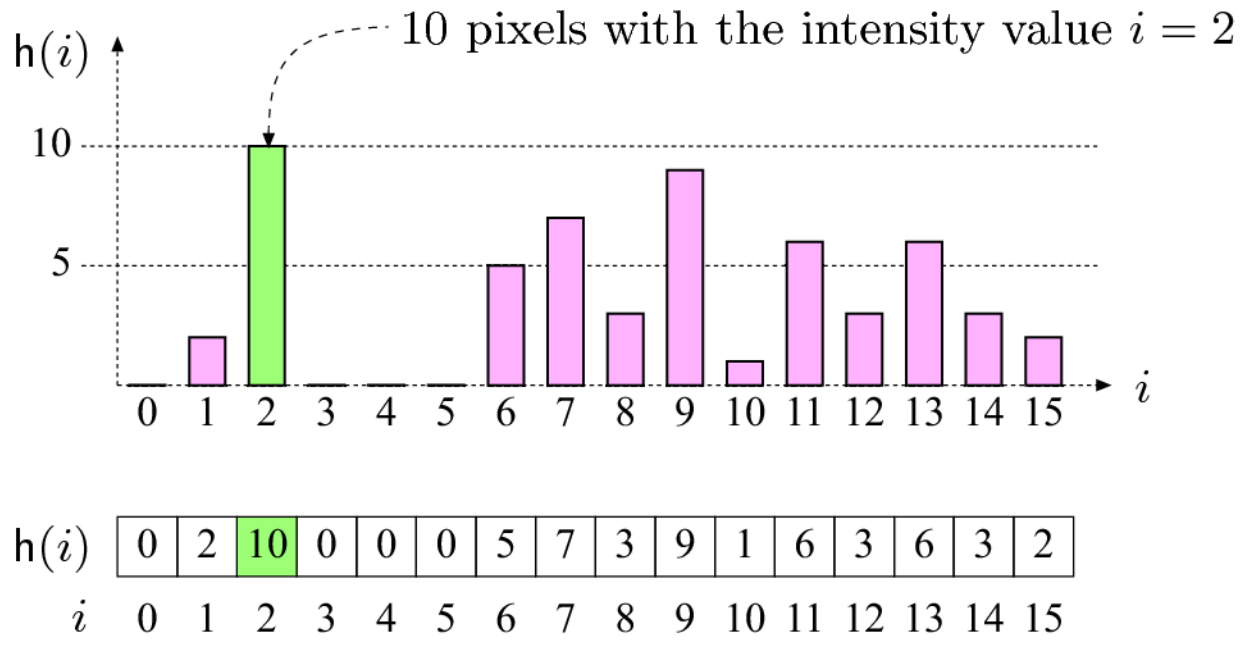


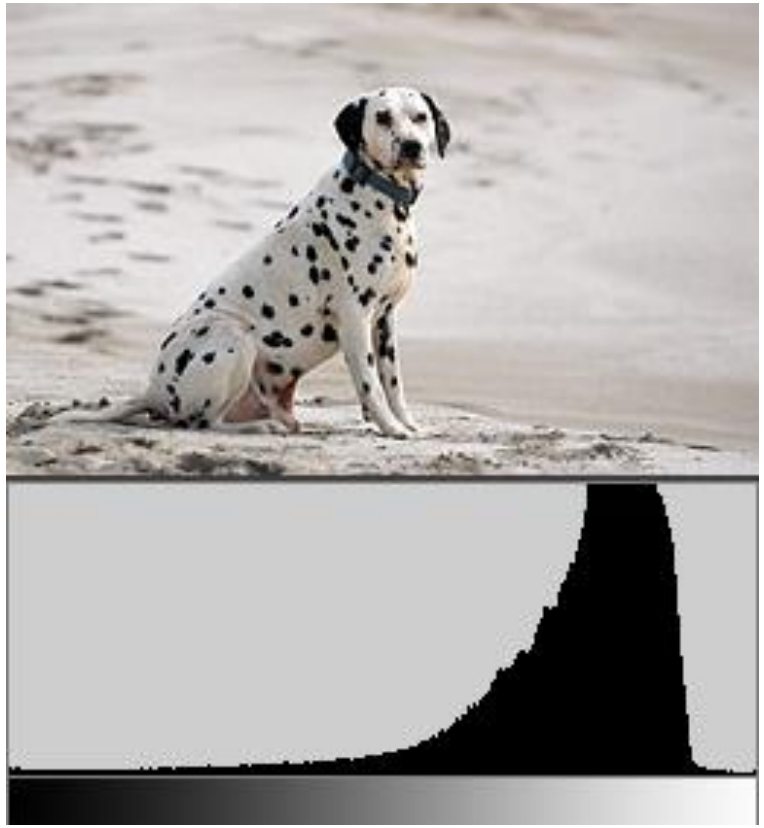
Image Histogram

Provides a **general idea of the image gray levels** and allows choosing an adequate processing operation

Ideally an image histogram should occupy the entire grayscale

Histogram-based processing has **two stages**:

- Computing and analyzing the **histogram**
- **Image enhancement** based on histogram features



Gathering insight from histograms

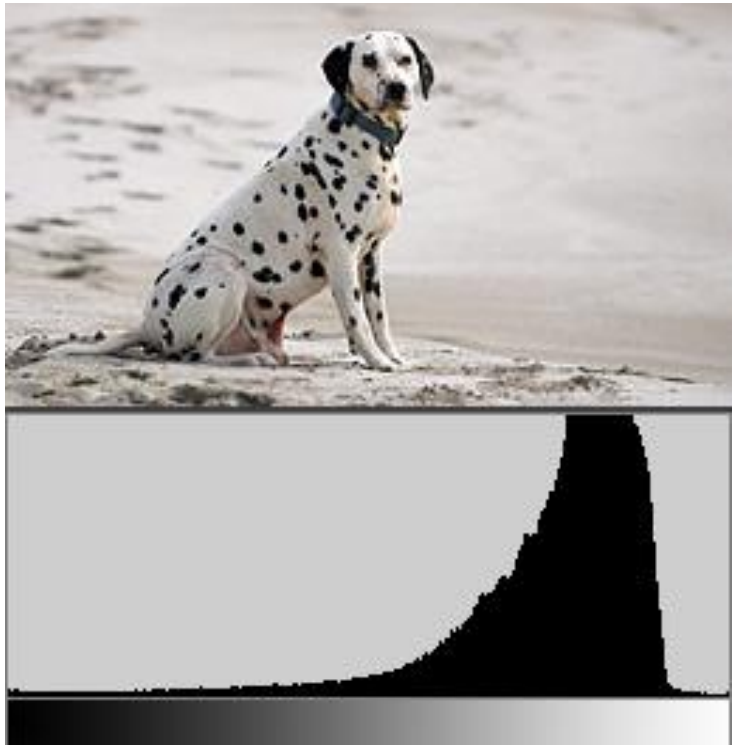
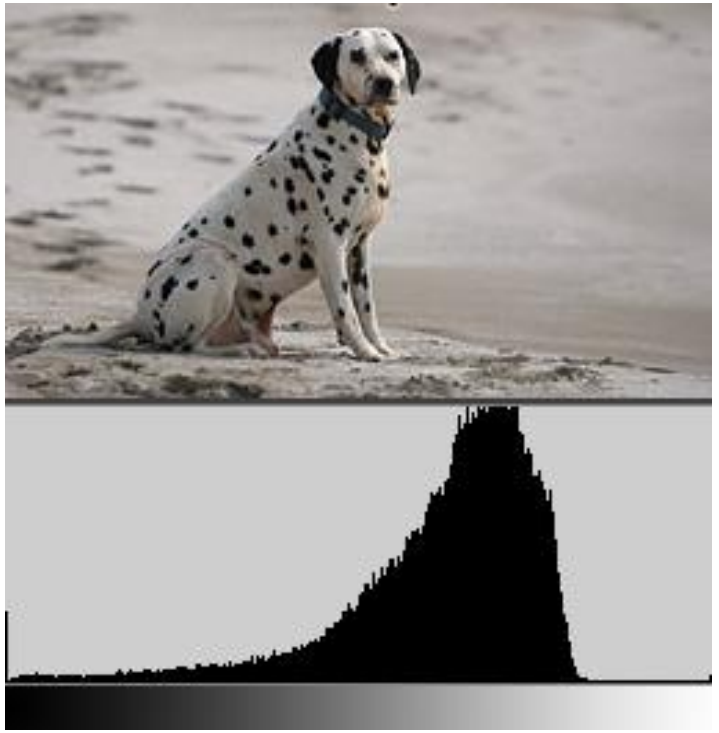


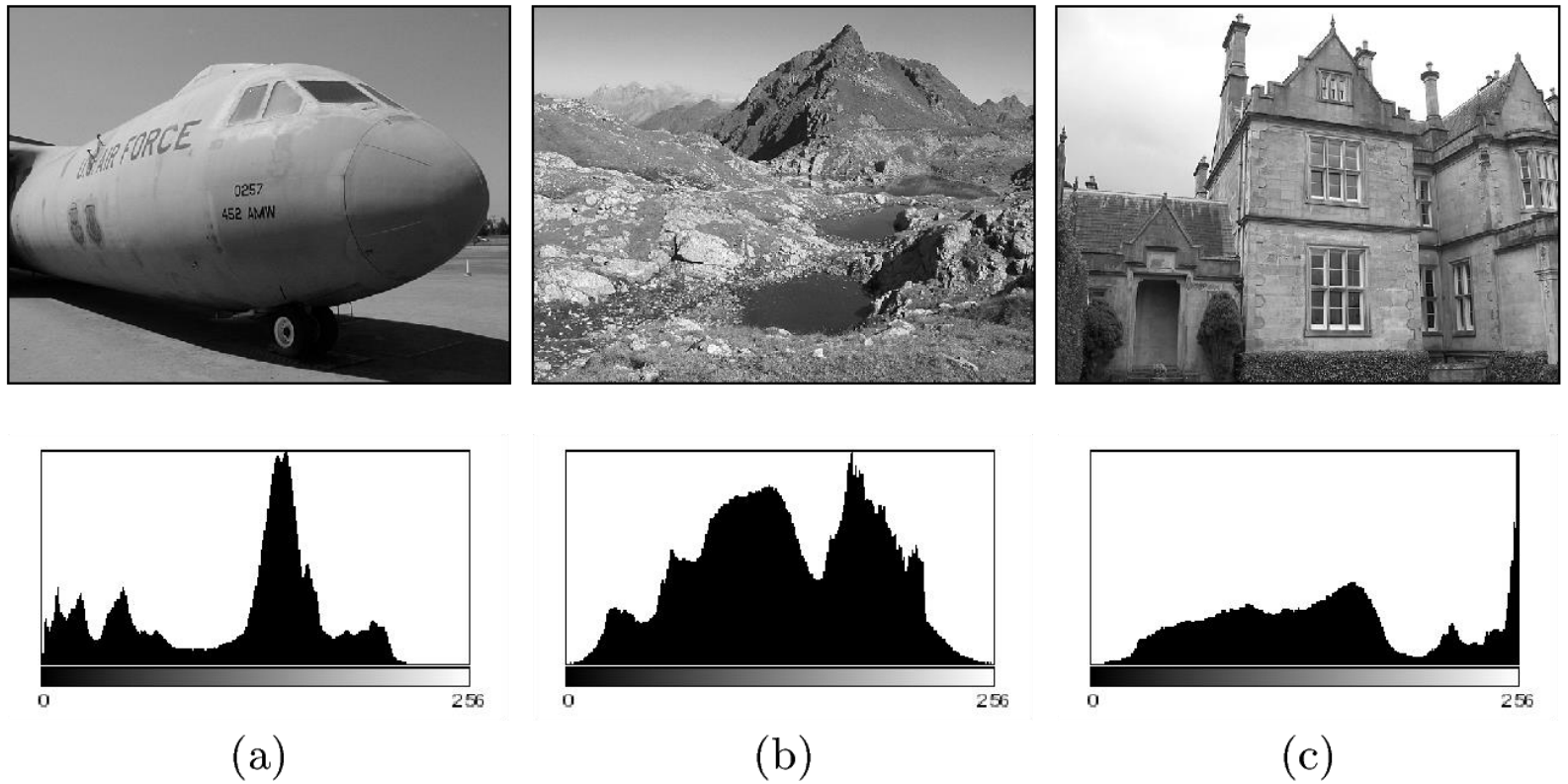
Image and histogram



Underexposed image:
slightly darker

http://www.cambridgeincolour.com/tutorials/histograms1.htm

Gathering insight from histograms



(a) – Underexposed image: loss of detail in the darker areas

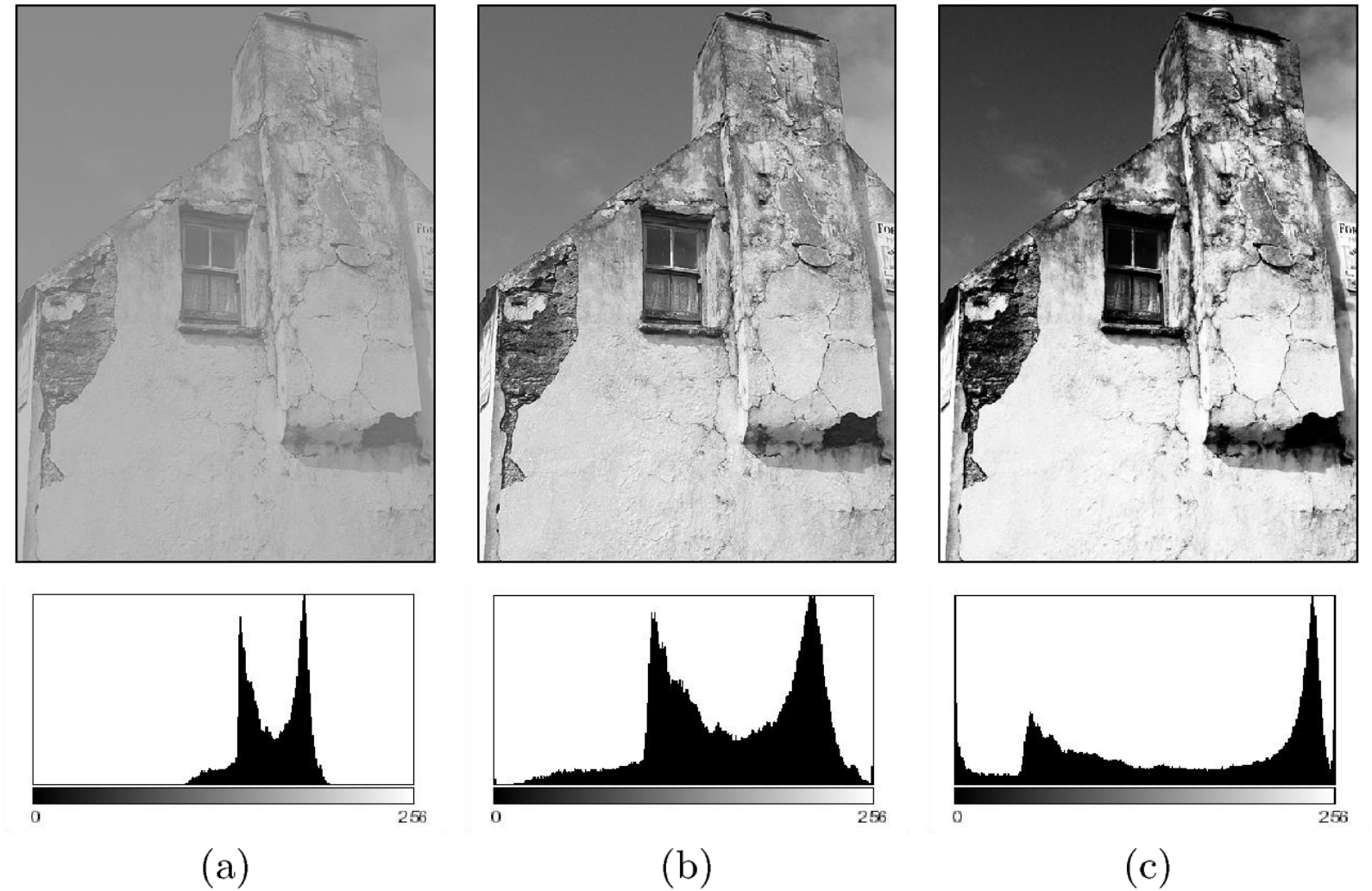
(c) – Overexposed image: loss of detail in the brighter areas

[Burger & Burge]

Gathering insight from histograms

- Contrast

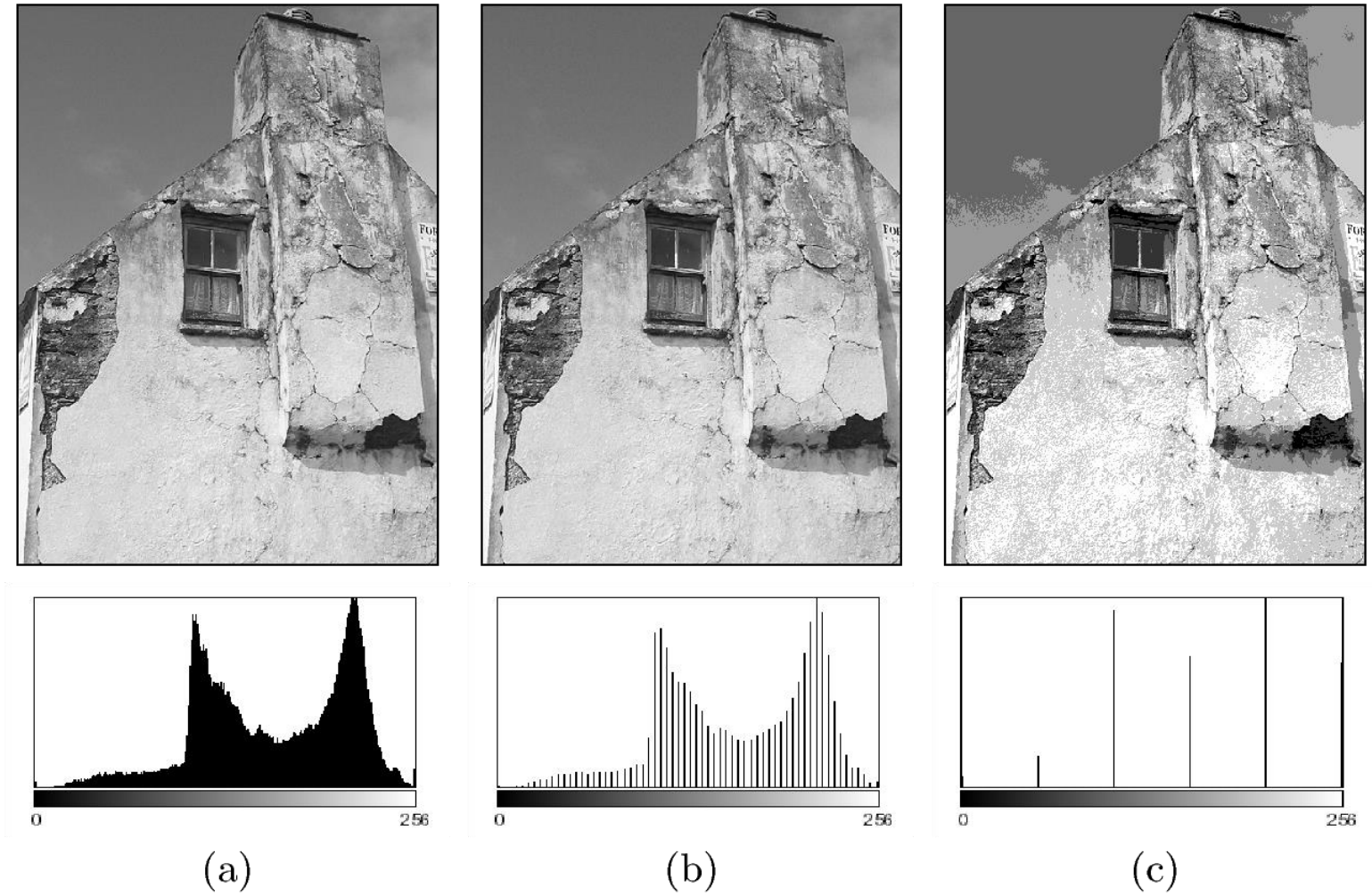
[Burger & Burge]



Gathering insight from histograms

- Number of intensity levels

256, 64 and 6 distinct intensity levels

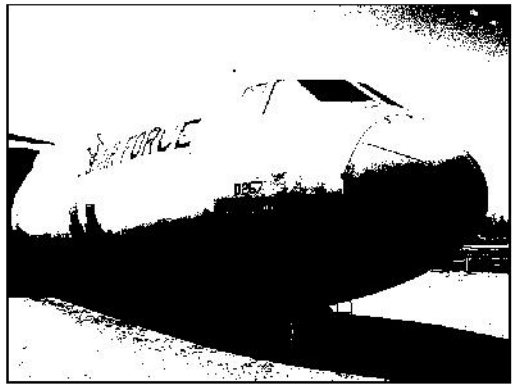


Thresholding

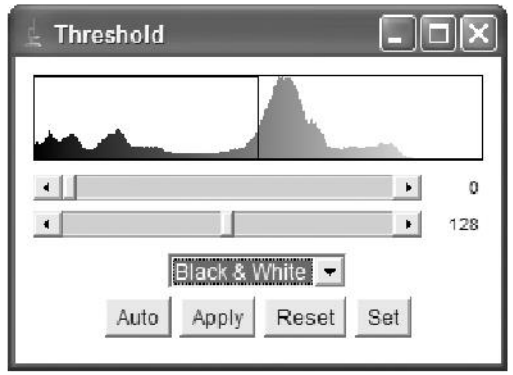
- Thresholding allows selecting an interval of the histogram and change it

[Burger & Burge]

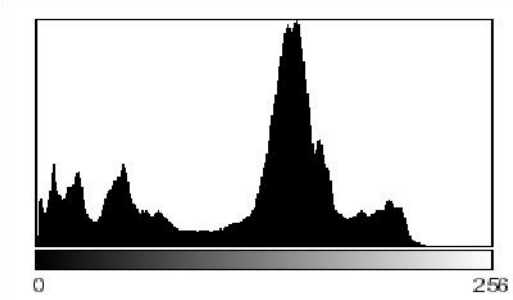
(a)



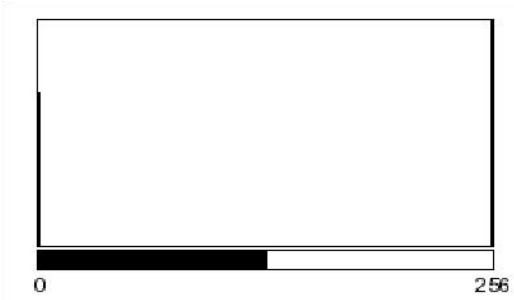
(b)



(e)



(c)



(d)

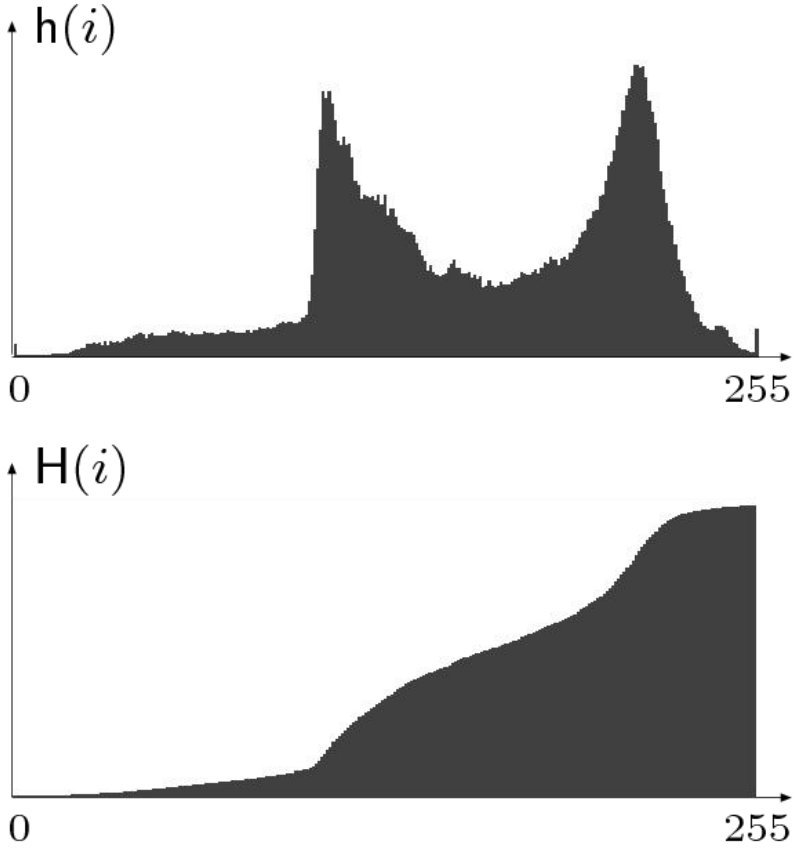
Thresholding

Cumulative histogram

Answers the question:
How many pixels have a value equal or below i?

When normalized it approximates a probability distribution function

[Burger & Burge]



Histogram Equalization

Used to optimize image contrast

Can be seen as a more advanced contrast-stretching

Works by flattening the histogram of an image

Histogram Equalization

The image is analyzed and a function $T(z)$ is computed in order to obtain an image with a **~ equiprobable histogram**

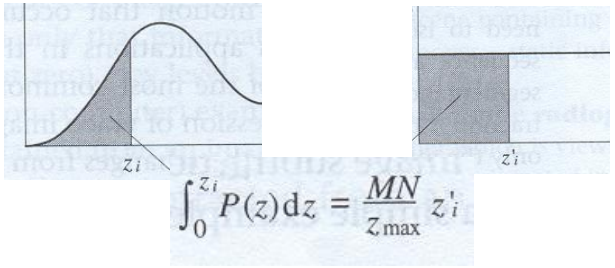
It is assumed that:

- z is continuous
- the image has a continuous probability density function $P(z)$
- the resulting image has an **equiprobable distribution** $Q(z)$

$$\int_0^{z_1} P(z)\,dz = \int_0^{z'_1} Q(z)\,dz = \frac{M\,N}{z_{max}}$$

$$z_i' = T(z_i) = \frac{z_{max}}{M\,N} \int_0^{z_1} P(z)\,dz$$

$$z_i' = \frac{z_{max}}{M\,N} \sum_0^{z_1} H(z)\,dz$$



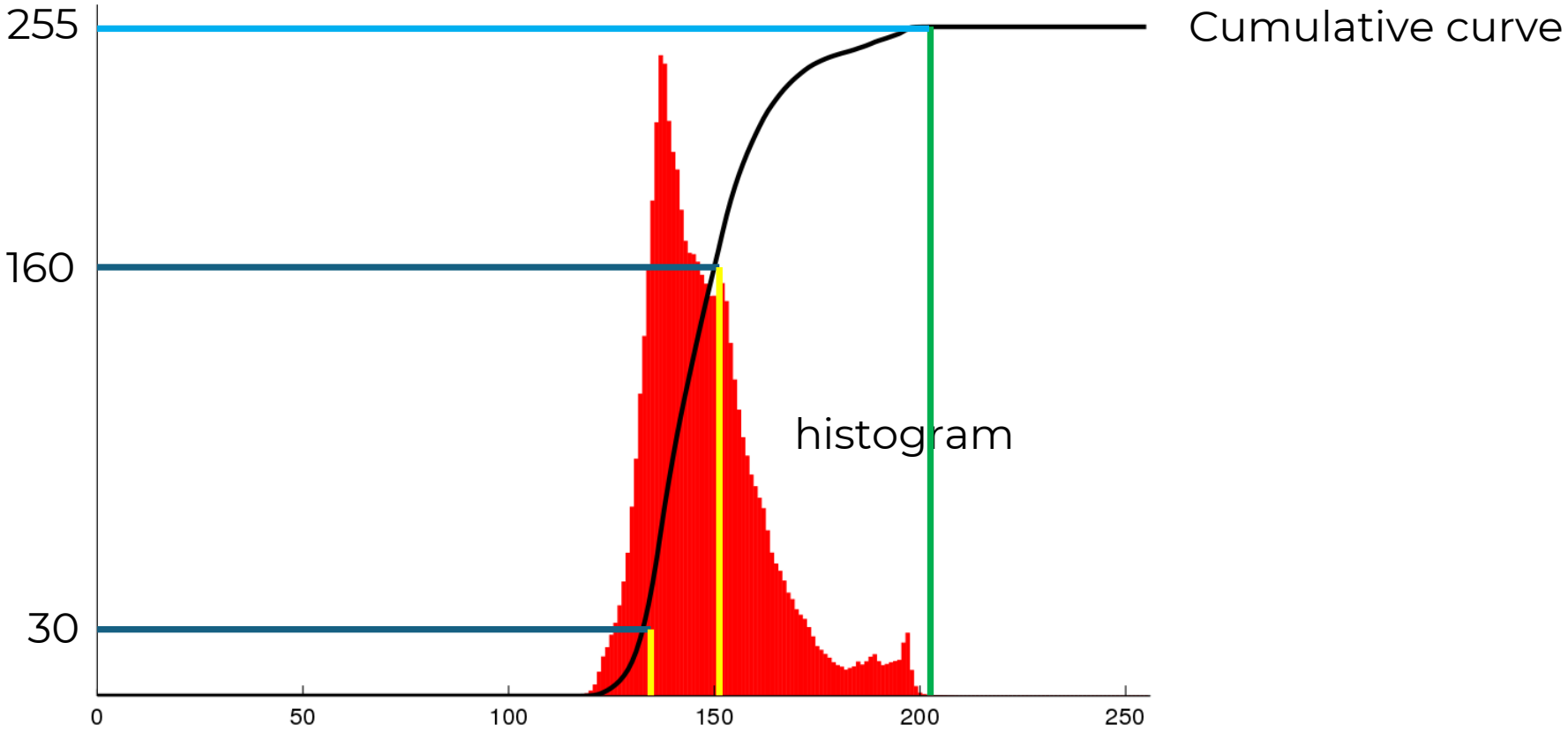
M x N – nº de pixels da imagem

é a aproximação discreta e
H é o histograma discreto

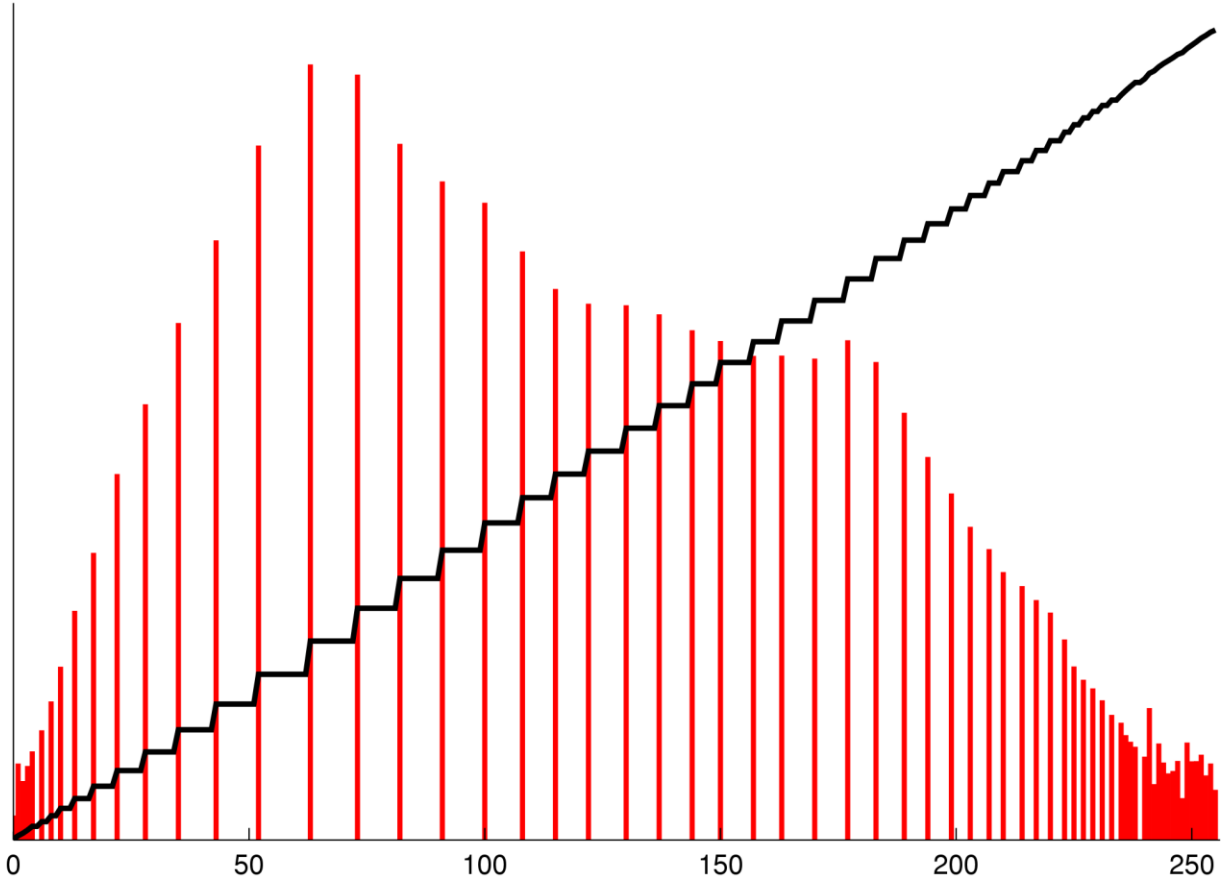
Oh my god, they killed Kenny!

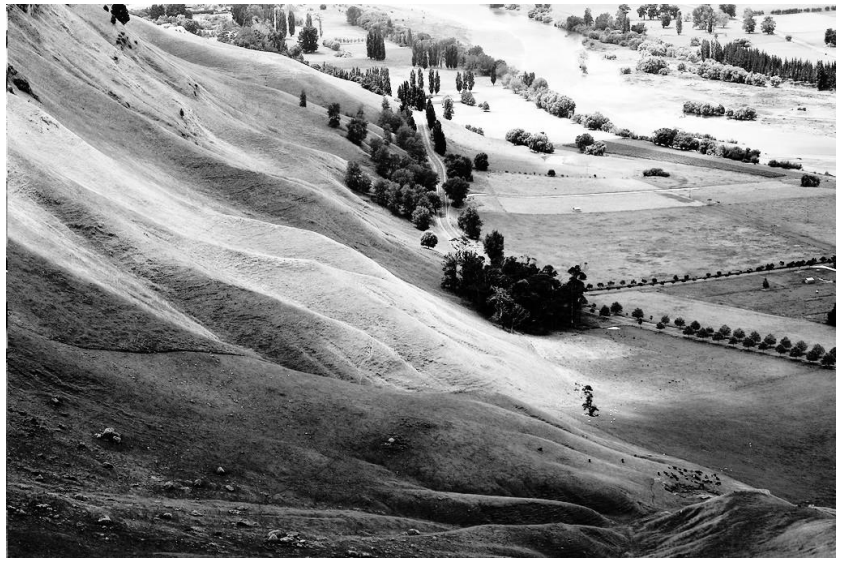
You b$”##rds!!

Histogram Equalization



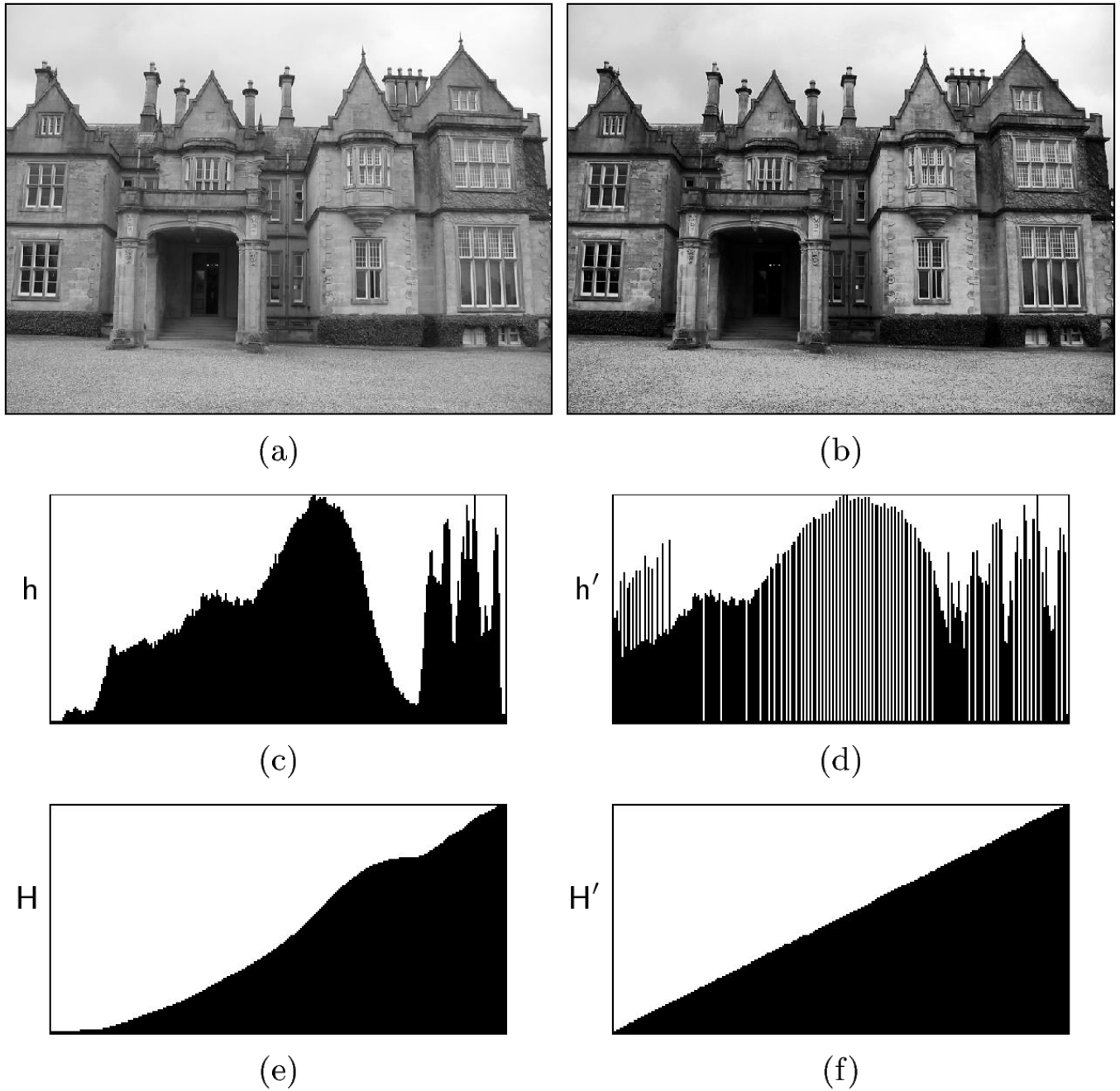
Histogram Equalization





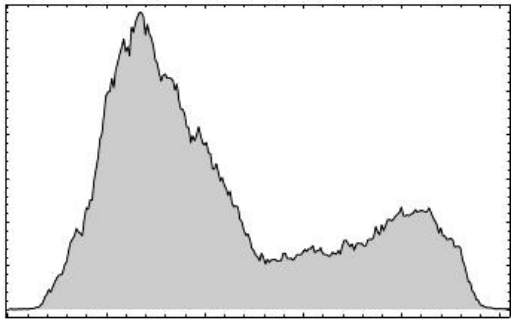
Linear Histogram Equalization

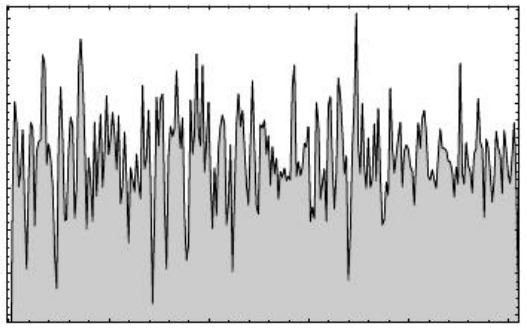
[Burger & Burge]



Cumulative histogram

Histogram Equalization





Arithmetic Operations

Arithmetic Operations

- Images can be added, subtracted, multiplied
- Watch out for saturation or negative values

(a) I_a.

(b) I_b.

(c) $I_a - I_b$.

Alpha Blending

- Two sources are added after being scaled by an alpha value

$$s_{\text{dst}} = \alpha \cdot s_1 + (1 - \alpha) \cdot s_2$$

- Allows the concept of **transparency** to be implemented
- Works for the whole image
- **Alpha compositing** can also be implemented **at the pixel level**
  - RGBA

http://www.alphamatting.com/datasets.php

Alpha Blending

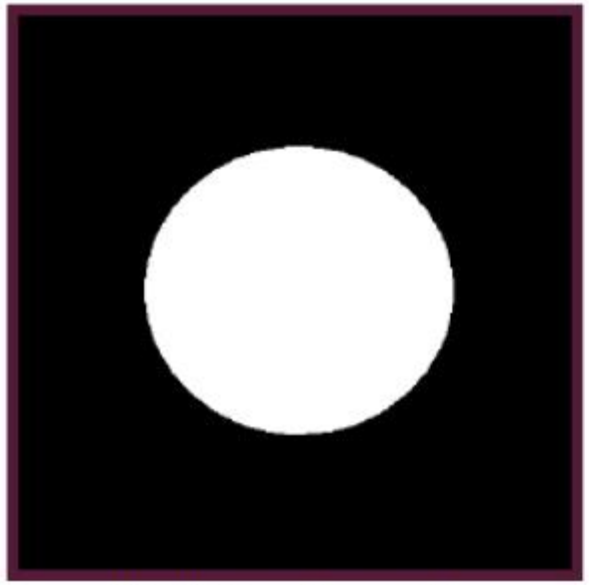
Logical Image Operations

e.g., for masking

(a) Binary Source 1.

(c) Conjunction.

(e) Disjunction.



(b) Binary Source 2.

(d) Exclusive or.

(f) Negation.

Image Filters

Region-based processing
Spatial-domain

https://www.google.com/url?sa=i&url=https%3A%2F%2Fbluebottlecoffee.com%2Fus%2Feng%2Fproduct%2Fblue-bottle-dripper-with-blue-bottle-coffee-filters&psig=AOvVaw2nhT1SofJWqy2q9NNc2o3R&ust=1699691684782000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCMifzoqDuYIDFQAAAAAdAAAAABAI

Regional Processing

In **point** (pixel) processing, a **single input** is processed to obtain a **single output** sample

In **regional** processing, the output sample **depends on multiple inputs** residing in a neighbourhood

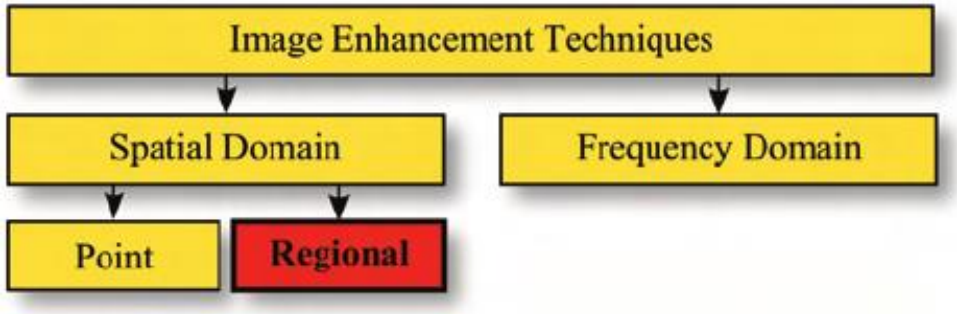
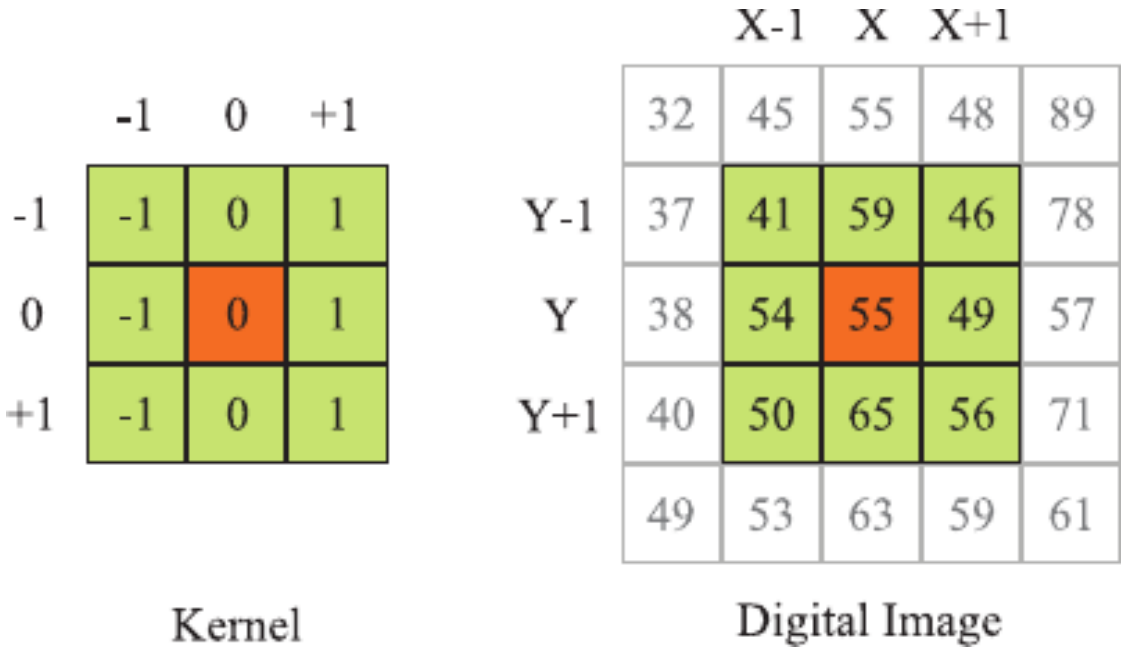


Image Filters

- First, we select the neighbourhood to process around the pixel using a matrix
- In image filtering this matrix is called a **kernel**
- Its values are related with how each neighbour will contribute to the output

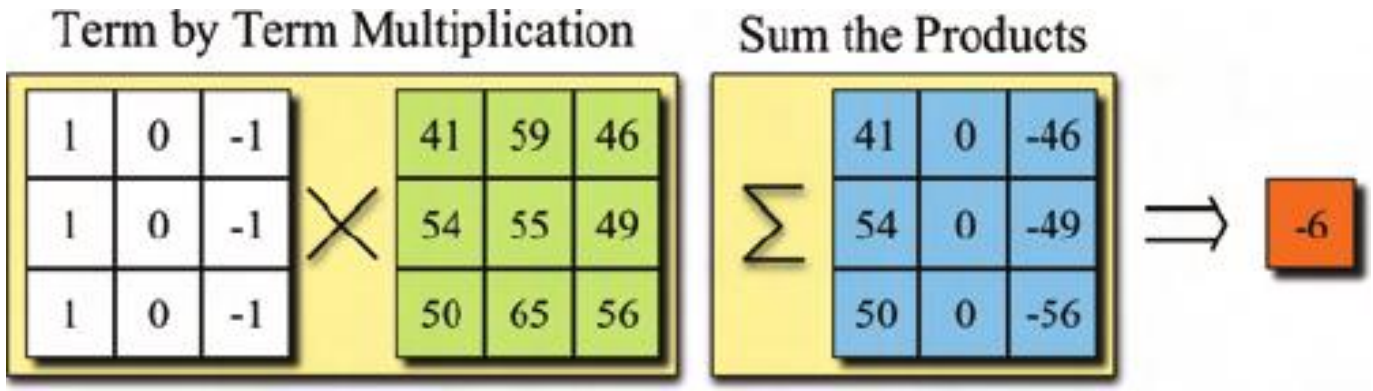


Convolution

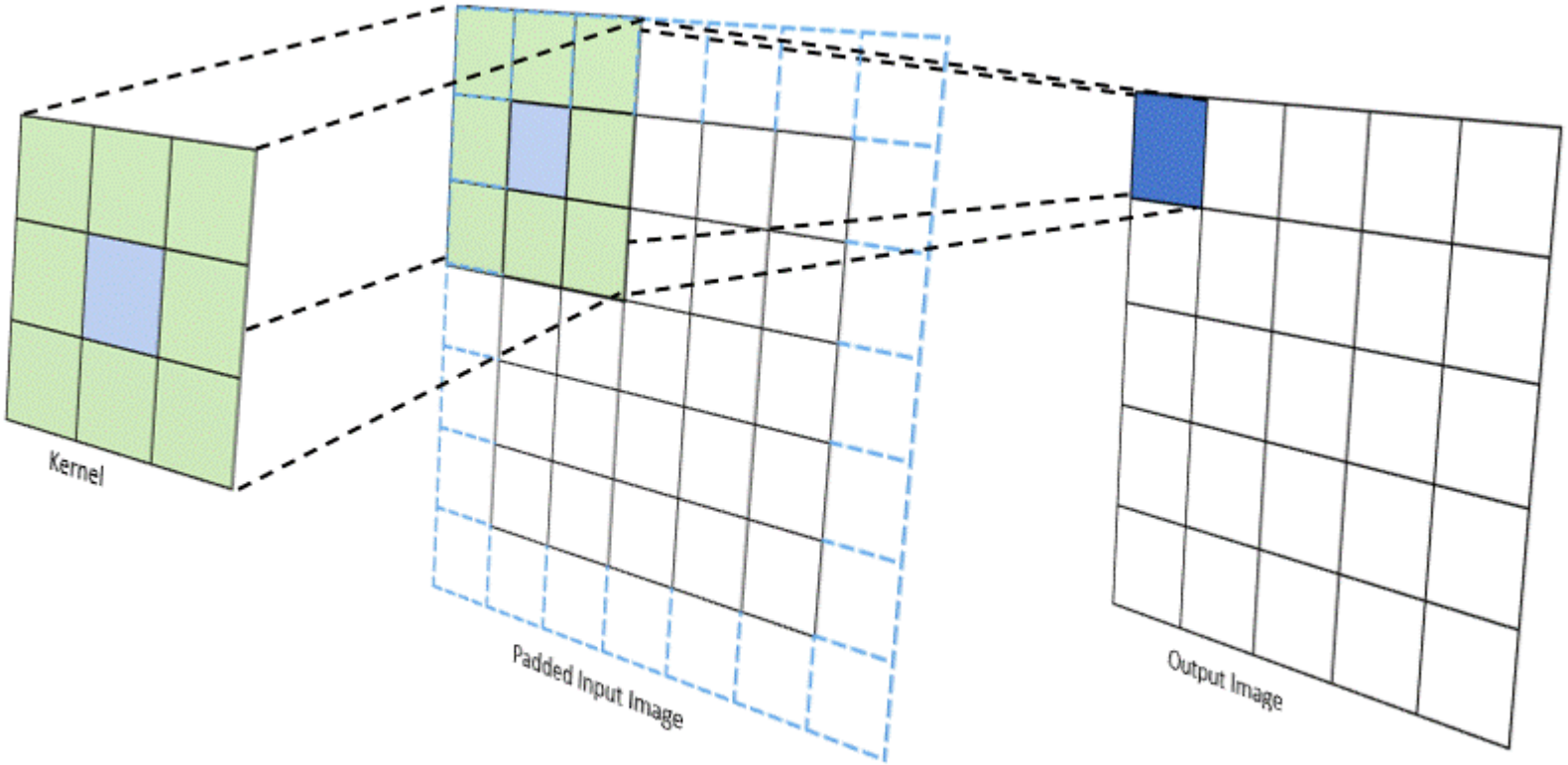
To apply the filter **the kernel is moved over the image stopping at each pixel** and performing the computation

Each **pixel value** at the origin is **transformed** into another pixel value at the destination **based on the kernel**

This operation of multiplying the kernel is called **convolution**



Convolution



https://miro.medium.com/v2/resize:fit:1400/1*juGbsuRpNjg15OCMwSmrmA.gif

Linear Filters

Smoothing

Also called **neighborhood averaging**, can be used to reduce image noise

The **kernel** slides over the original image and the **intensity** of the resulting pixel in the final image is given by:

$$\sum_{i=0}^{N-1} W_i \cdot I_i$$

Other filters allow reducing noise **without blurring** (e.g., **median filter**)

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3 x 3 Averaging Filter
All weights are 1/9

Why 1/9?

Smoothing / Blurring

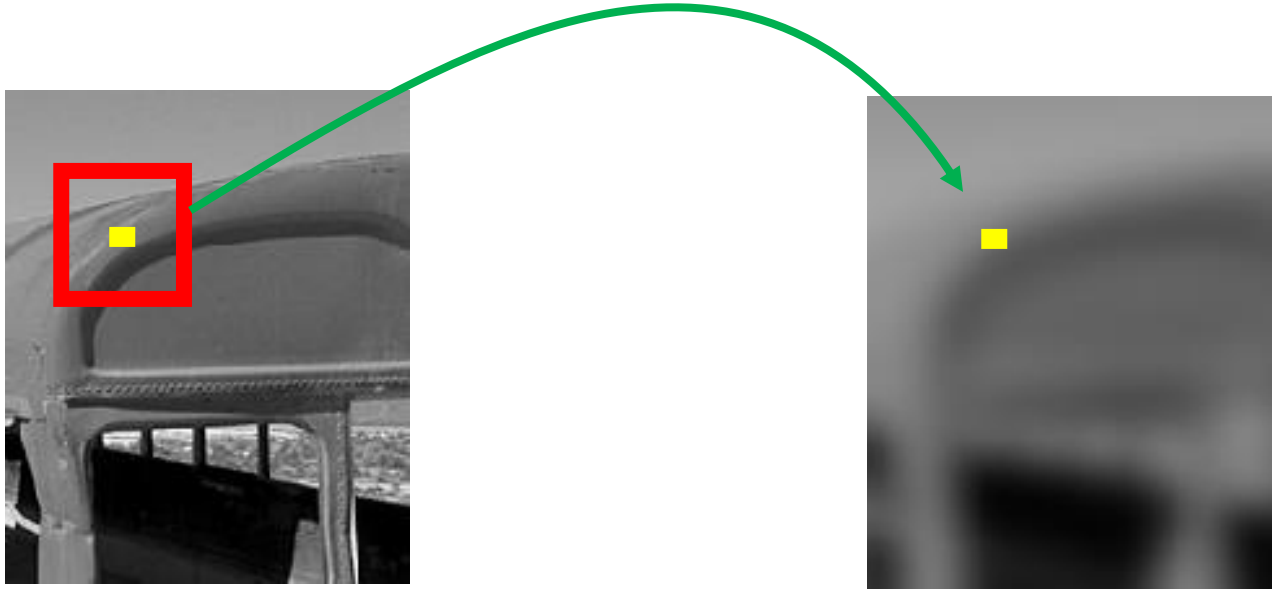
[Burger & Burge]

Averaging is an easy method for smoothing, but it also causes blurring

The bigger the kernel, bigger the smoothing (blurring)

Smoothing / Blurring

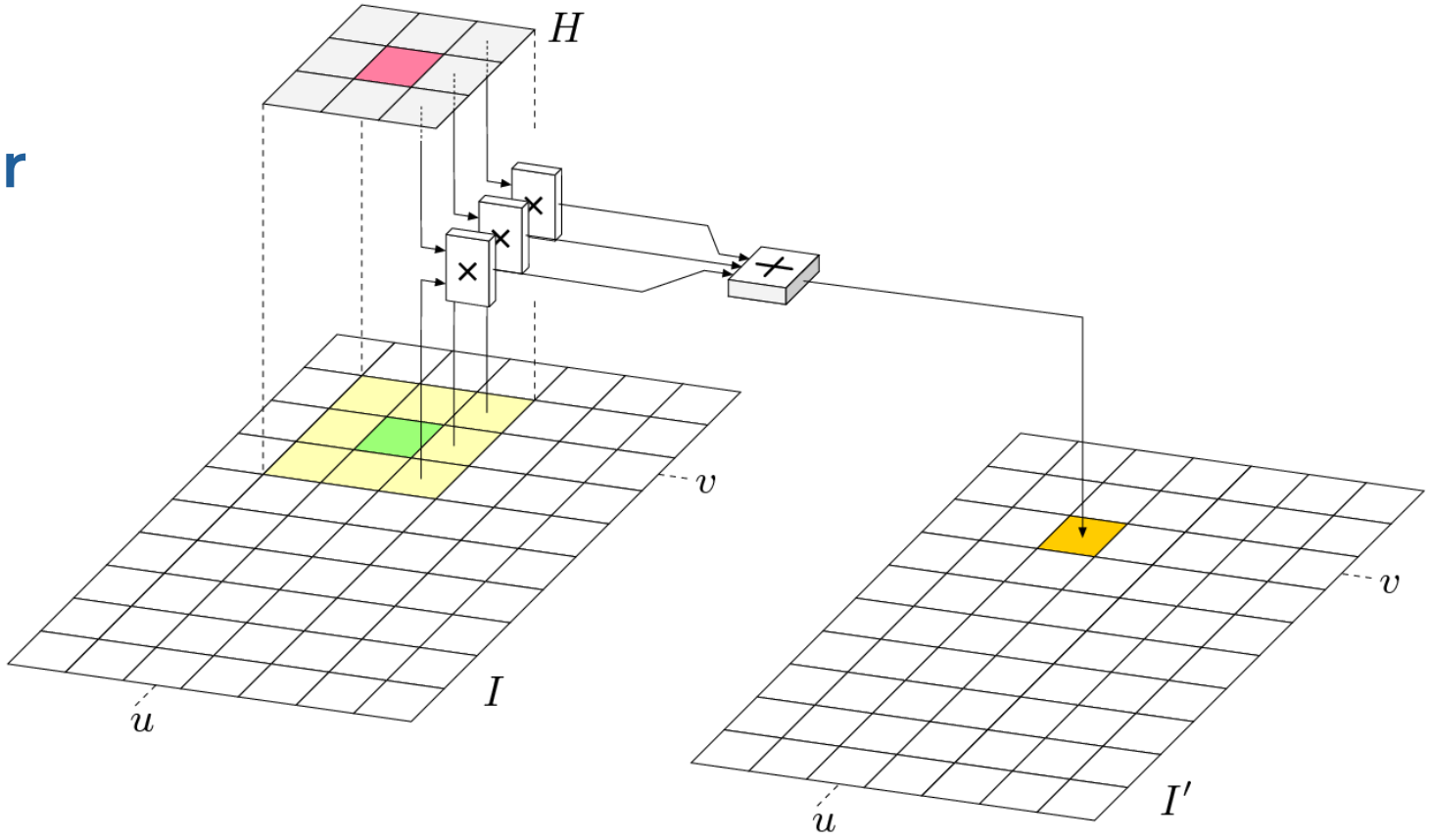
This is the basic principle of all filtering performed with kernel convolution. What changes is the weights



Linear Filter

Averaging is a type of **linear filter**

Pixel value is obtained by a linear operation over its neighbourhood



Smoothing kernels

A kernel where **all nonzero coefficients are identical** is called a **uniform filter**

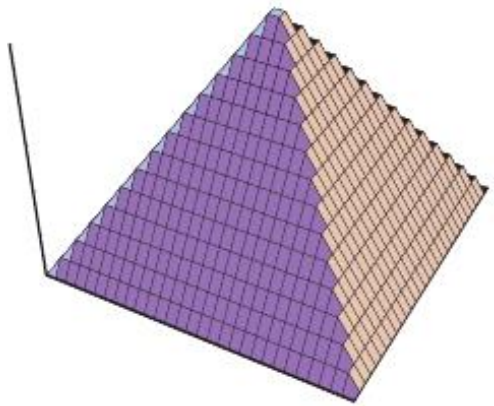
$$\frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(a) Box filter.

$$\frac{1}{21} \times \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

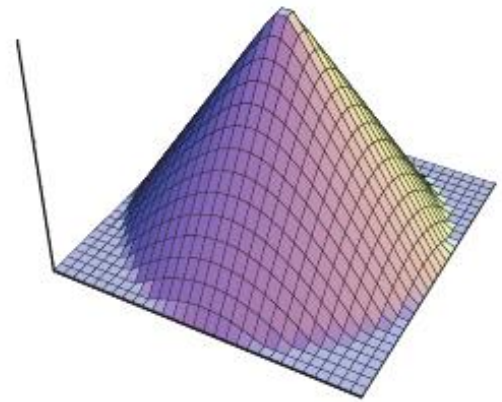
(b) Circular box filter.

Smoothing Kernels



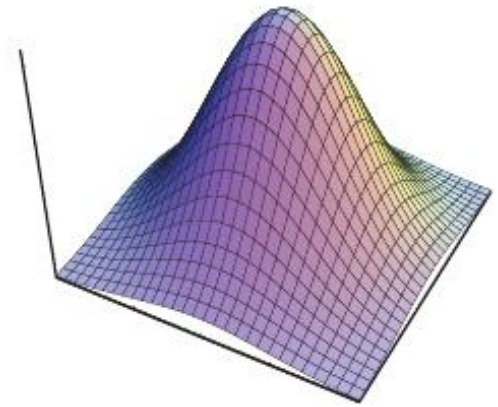
1	2	3	2	1
2	4	6	4	2
3	6	9	6	3
2	4	6	4	2
1	2	3	2	1

(a) Pyramid.



0	0	1	0	0
0	2	2	2	0
1	2	5	2	1
0	2	2	2	0
0	0	1	0	0

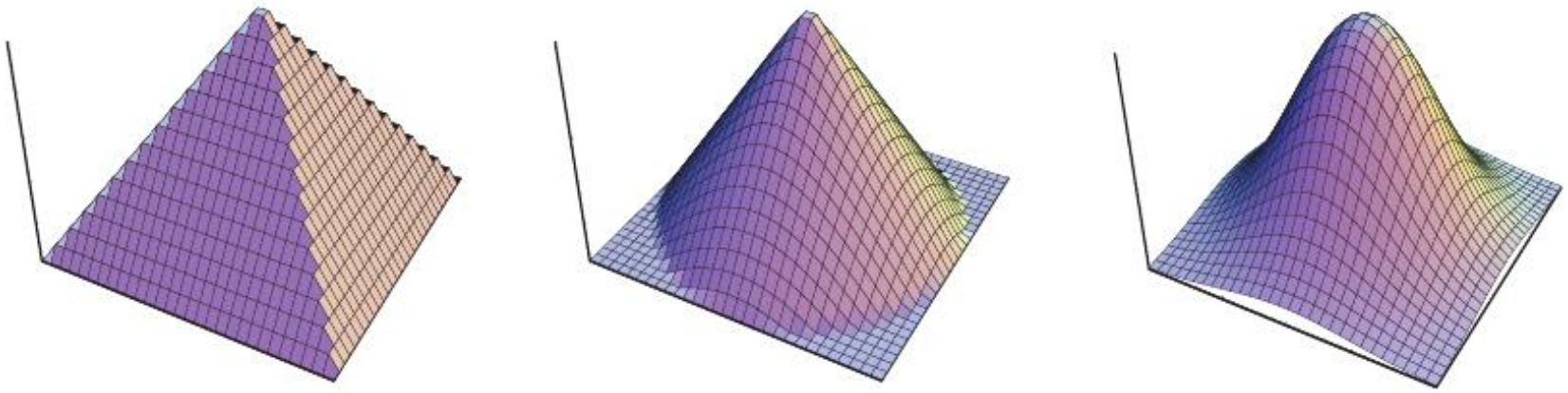
(b) Cone.



1	4	7	4	1
4	16	28	16	4
7	28	49	28	7
4	16	28	16	4
1	4	7	4	1

(c) Gaussian.

Smoothing Kernels



The Gaussian kernel is the preferred given its smoothness

Smoothing

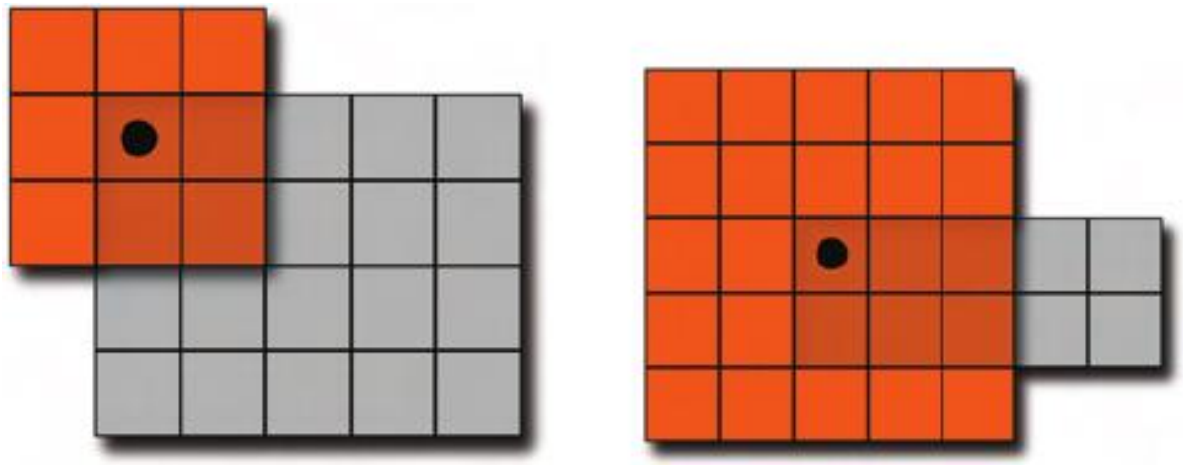
source

17 x 17 box

Gaussian

The Boundary Problem

- When applying convolution to process an image, what happens…
  - When we process **border pixels?**
  - When the **kernel is bigger than the image?**



How to handle image borders?

- Simple solutions can be:
  - Zero
  - Keep the same value
- Solutions that behave better include:
  - Padding
  - Mirroring
  - Circular tiling
  - Extend border values

[Burger & Burge]

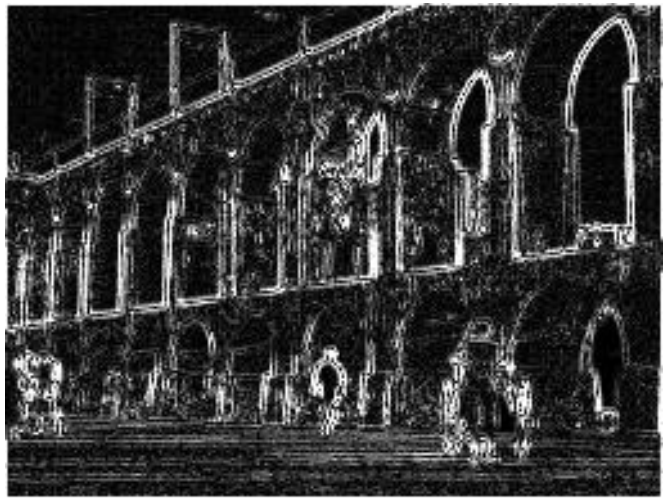
Sharpening

-1	-1	-1
-1	8	-1
-1	-1	-1

It is an **approximation** to the **high-pass filter** in the frequency-domain

Implemented in the space-domain

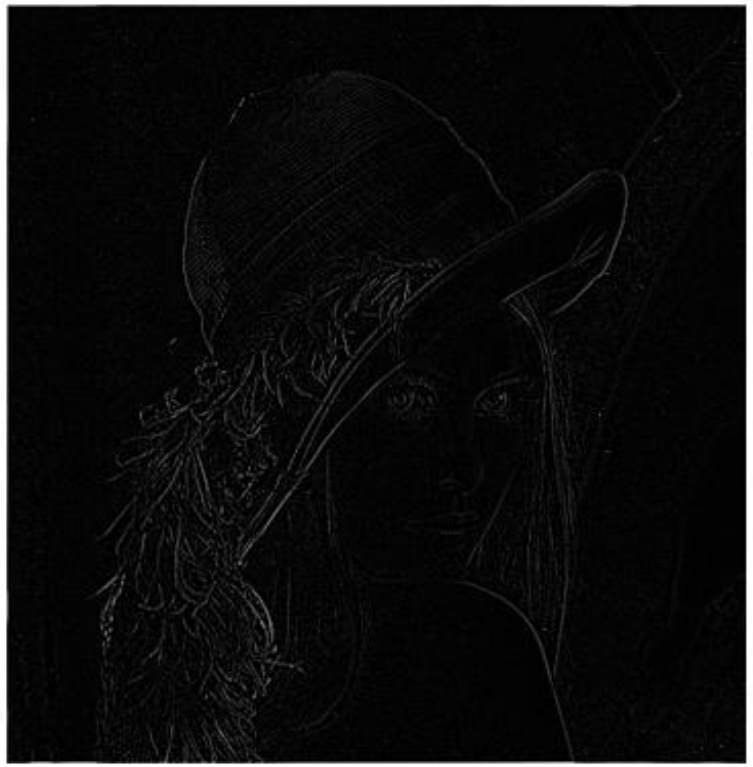
If all the pixels in the neighborhood have the **same intensity** value (gray level) the corresponding pixel in the result image has value **zero** (black)



Sharpening
Laplacian Filtering

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1



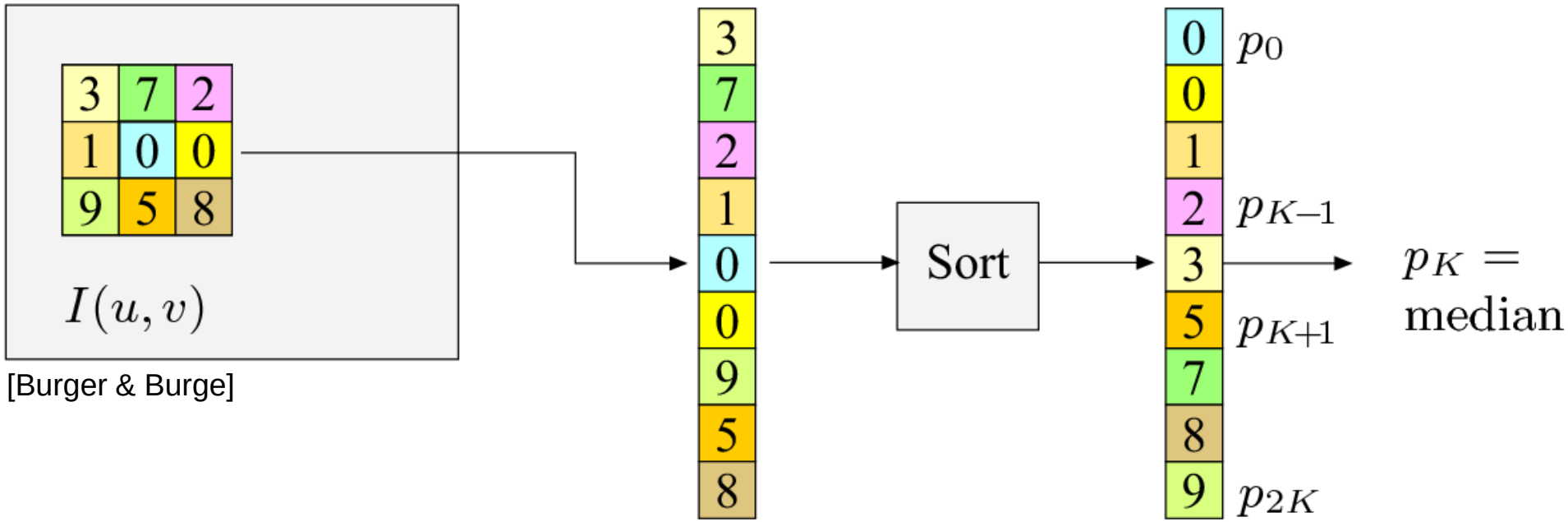
Sharpening

wikipedia

Rank Filters (aka Non-Linear)

Based on a statistical analysis of the neighbourhood

Median Filter



[Burger & Burge]

Median vs Averaging

A median filter is better than the averaging filter at smoothing salt and pepper noise.

Why?

original

median

Gaussian

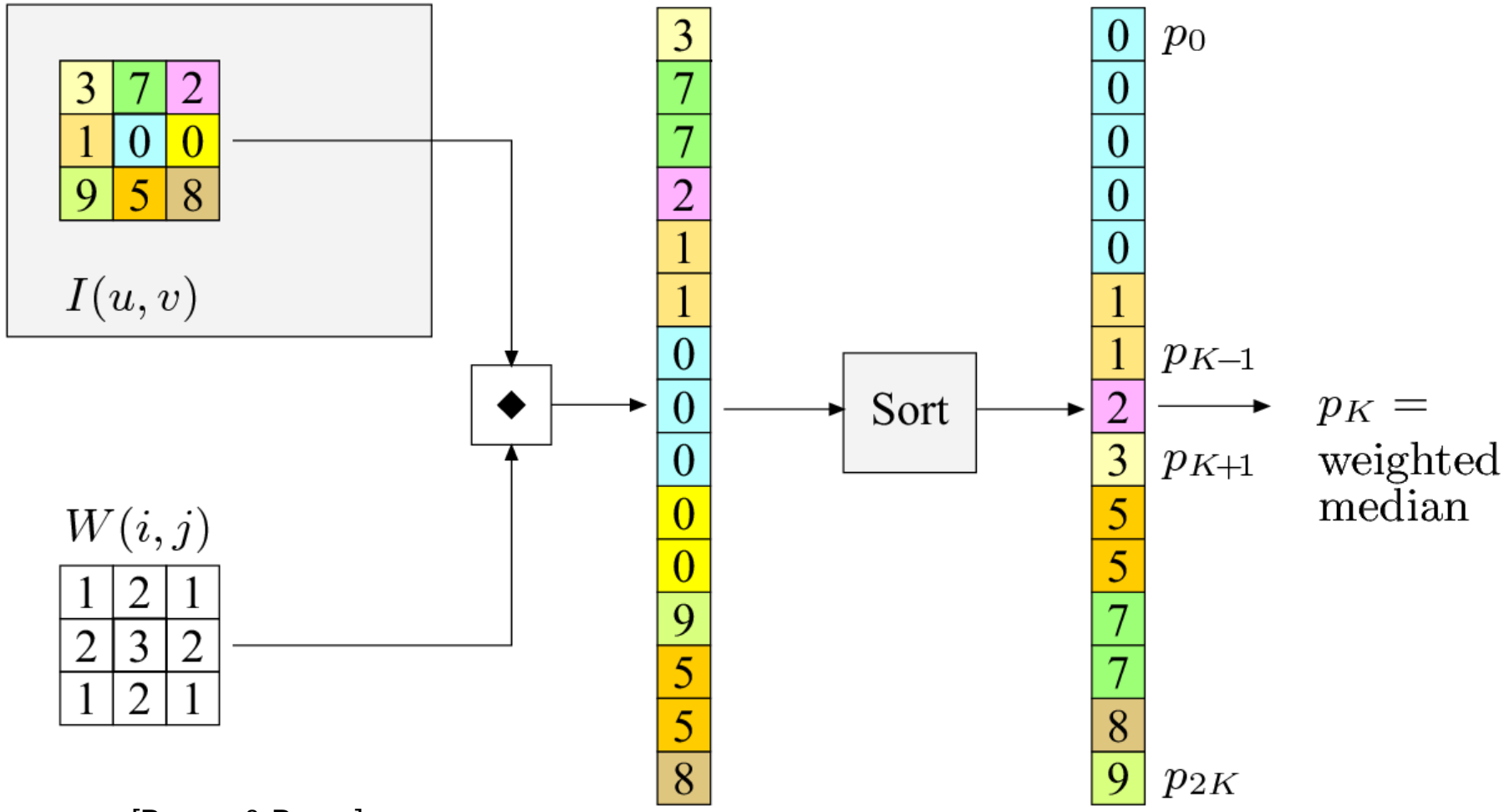
Averaging Filter vs Median Filter

[Burger & Burge]

original averaging median

by the way, this is Lena

Weighted Median Filter



[Burger & Burge]

Minimum Filter vs Maximum Filter

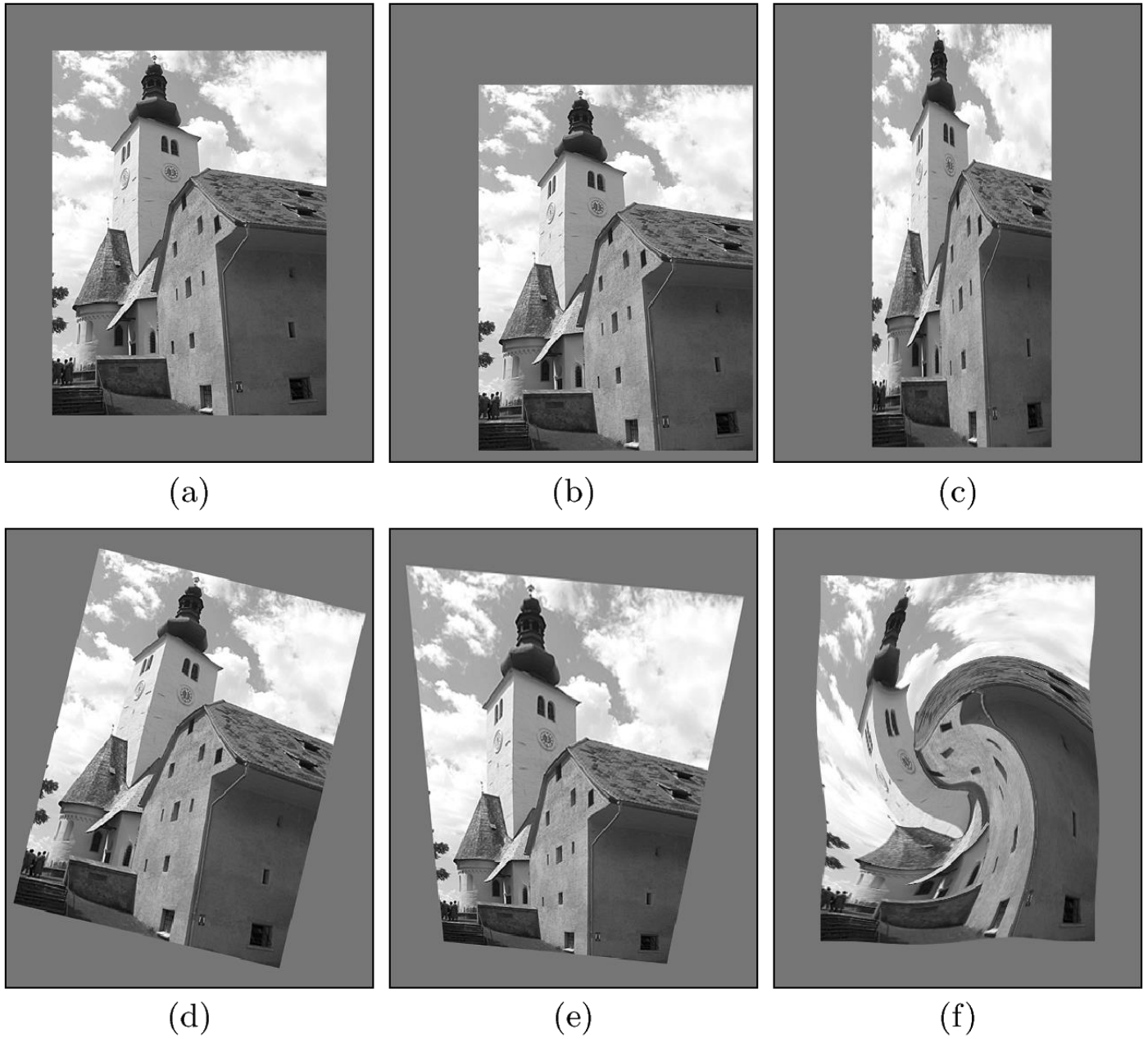
original minimum maximum

Geometric Transformations

- Another possibility is to **operate on pixel position** and not on intensity or color values:
  - Correction of geometric distortion due to acquisition effects
  - Distortion of an image into another (image warping, morphing)
  - Texture mapping

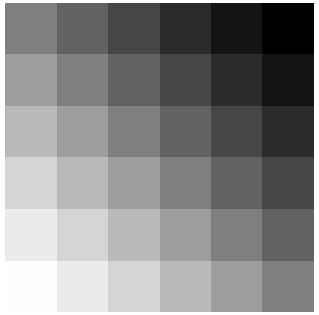
Global geometric transformations

[Burger & Burge]

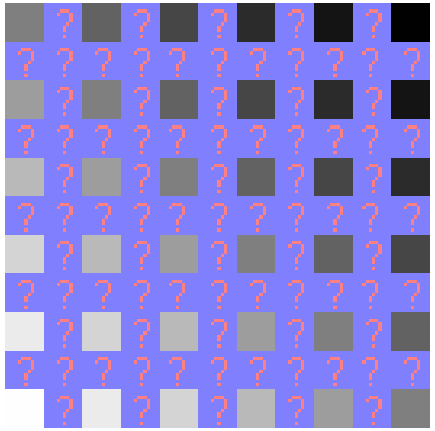


Interpolation

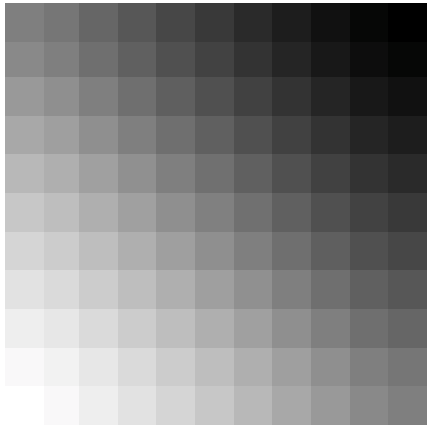
- When we **resize** an image, how do we fill the gaps?
- **Interpolation** works by using known data to estimate values of unknown points
- Interpolation **does not** add detail



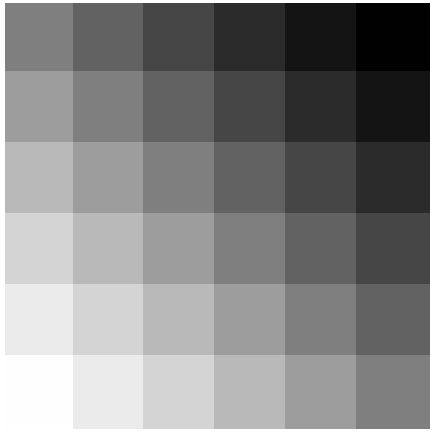
original



Before interpolation

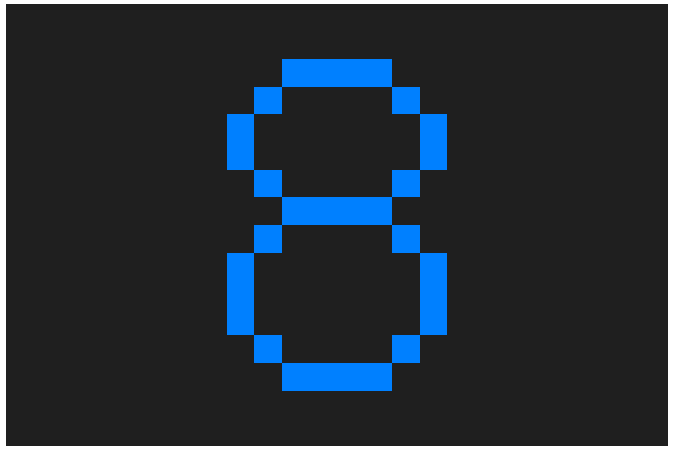


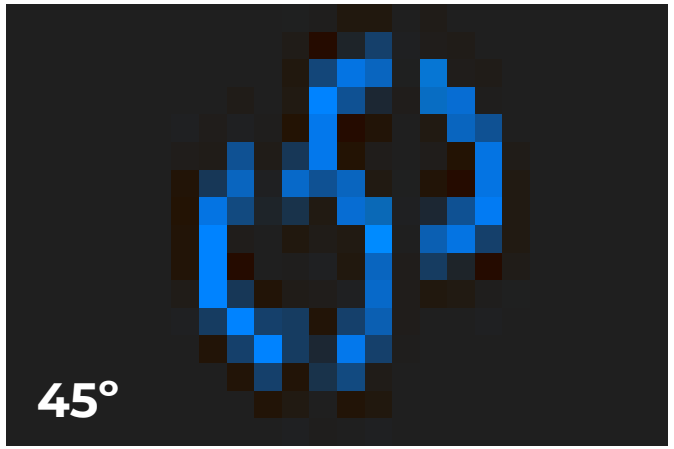
After interpolation



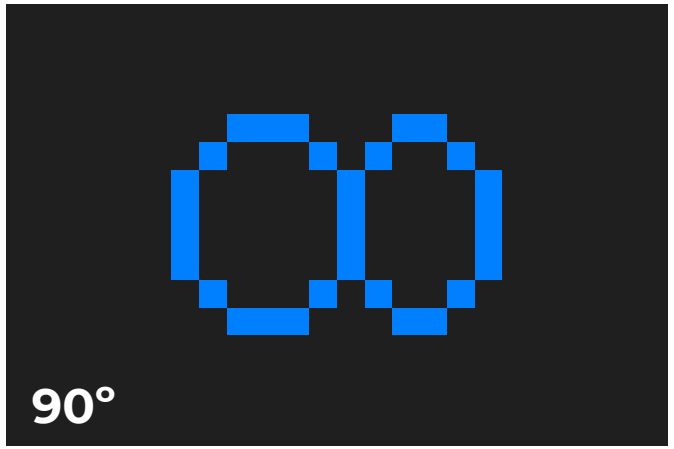
No interpolation

Interpolation

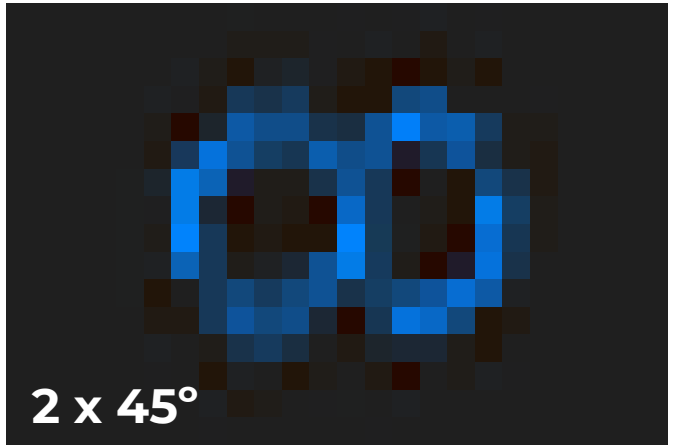




- Interpolation also happens when other transformations are applied to the image
- **Rotations**, for instance, can have a **strong impact on image quality** due to interpolation
- **Cumulative rotations** can be **catastrophic**



- Avoid rotating images beyond what is necessary
- Perform the **rotation in a single operation**



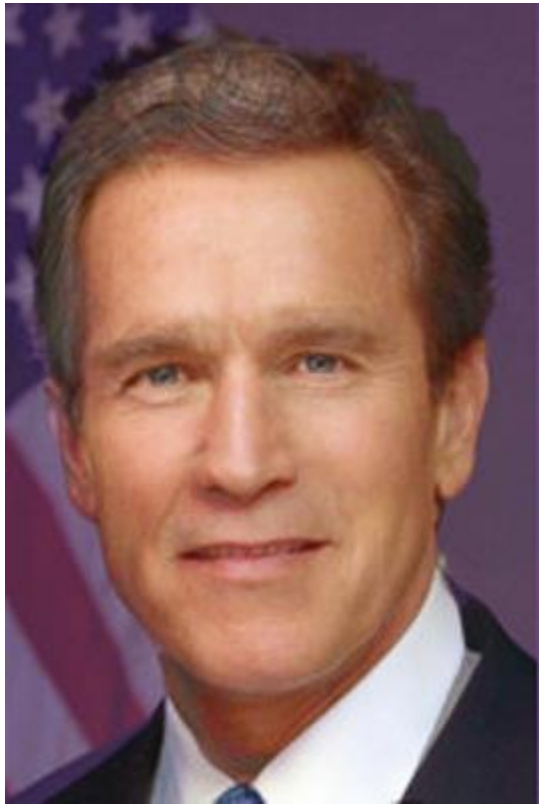
https://www.cambridgeincolour.com/

Interpolation

Multiple methods for interpolation are available and provide increasing degrees of quality:

- Nearest neighbour
- Bilinear
- Bicubic
- Spline
- …

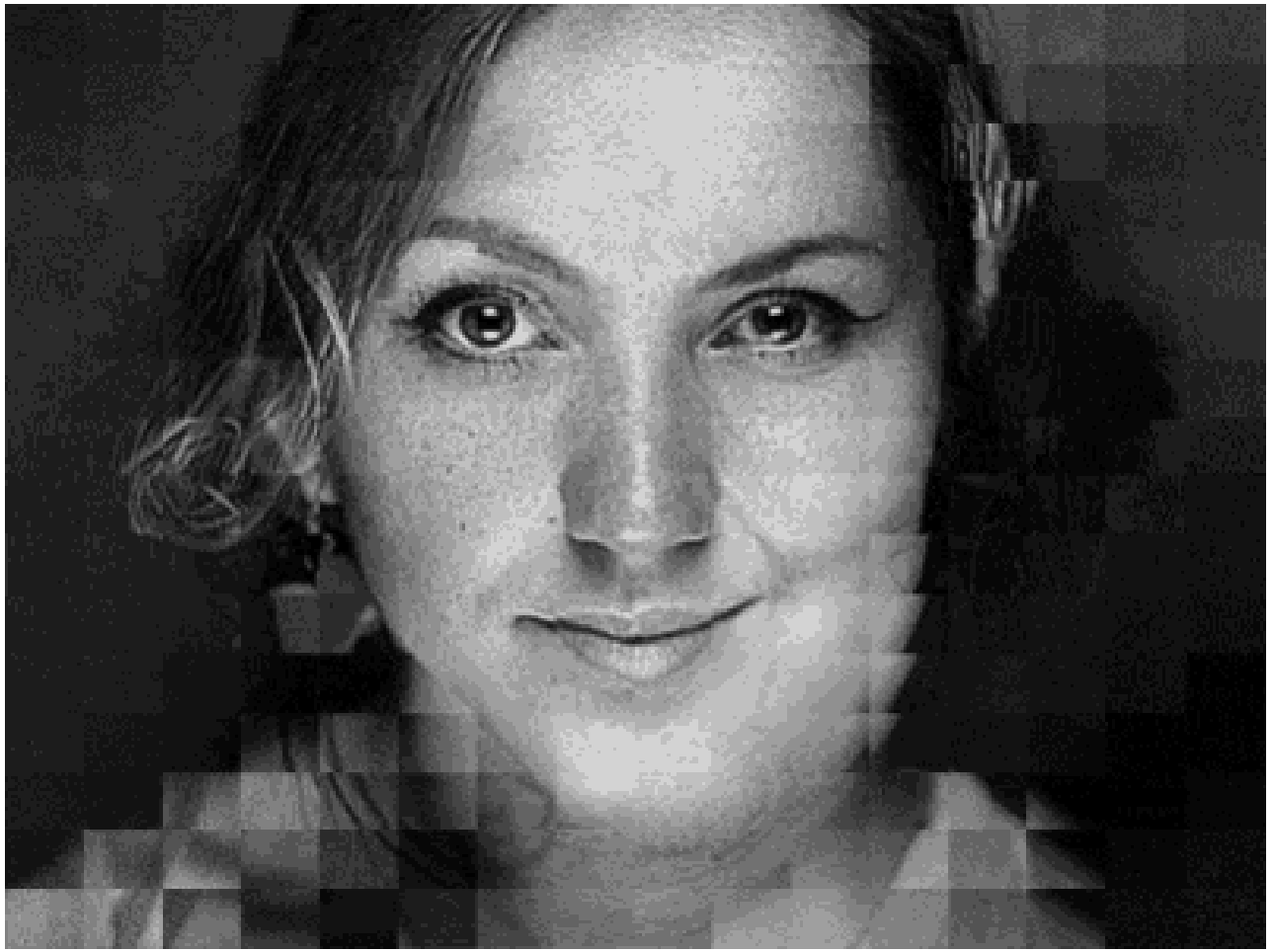
Who is this guy?



Morphing

https://educalingo.com/

Morphing



Bibliography

- Kenny A. Hunt, “The Art of Image Processing with Java”, chapters 3, 5 and 6, AK Peters, 2010 https://learning.oreilly.com/library/view/the-art-of/9781439865590/

- Alasdair McAndrew, “A Computational Introduction to Digital Image Processing”, 2nd ed., chapters 4, 5 and 6, CRC Press, 2016 https://learning.oreilly.com/library/view/a-computational-introduction/9781482247336/