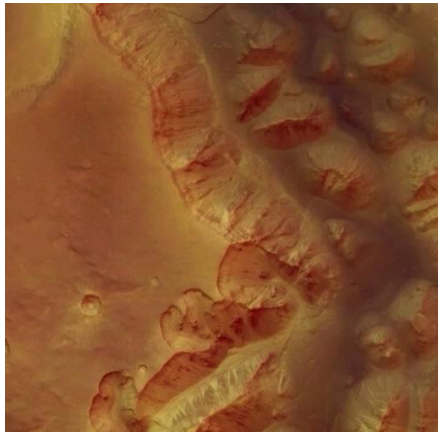
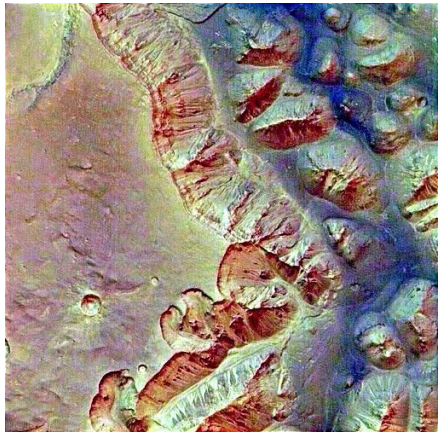
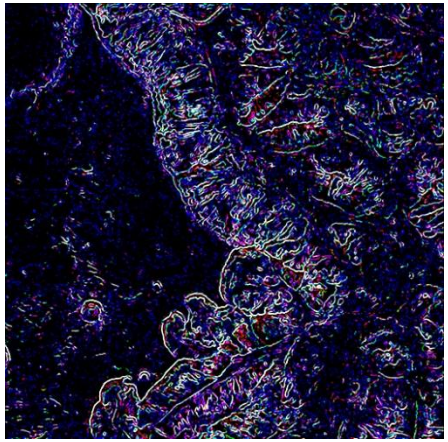
Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

An Introduction to Image Processing

Computação Visual

Beatriz Sousa Santos, J. Madeira

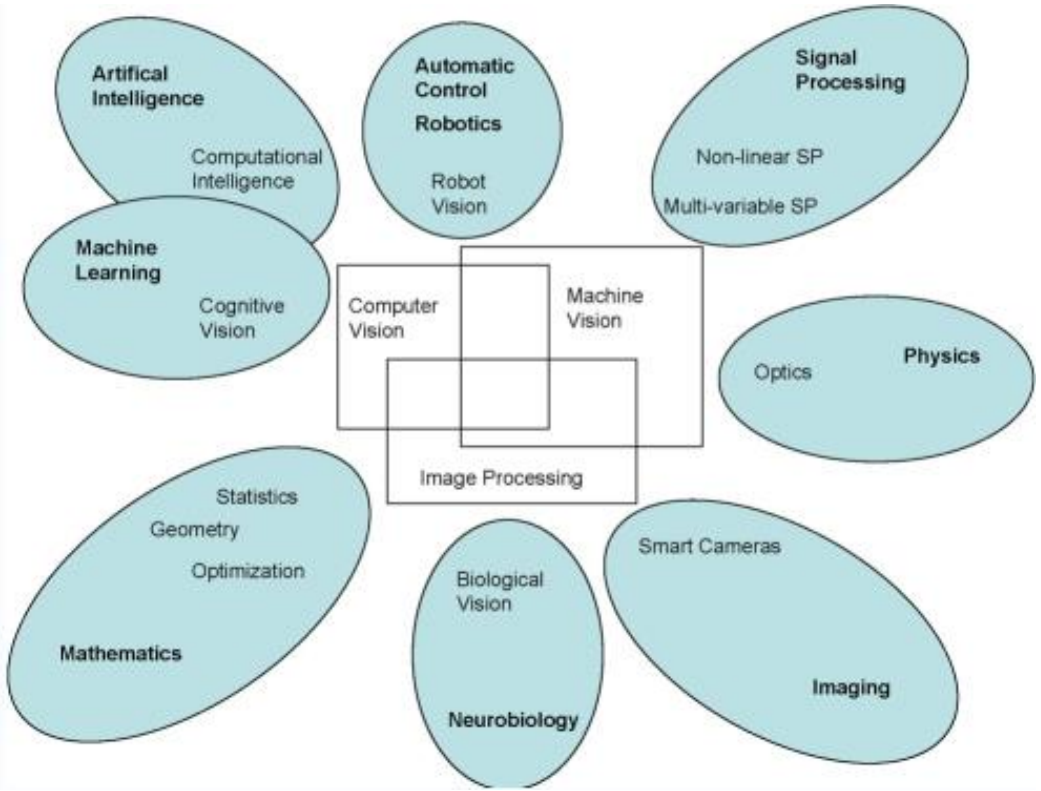
Overview

- Digital Image Processing
- The Processing Pipeline
- Pixel Intensity Operations
- Histogram-based Operations
- Filters
- Geometric Transformations

INTRODUCTION

- **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**

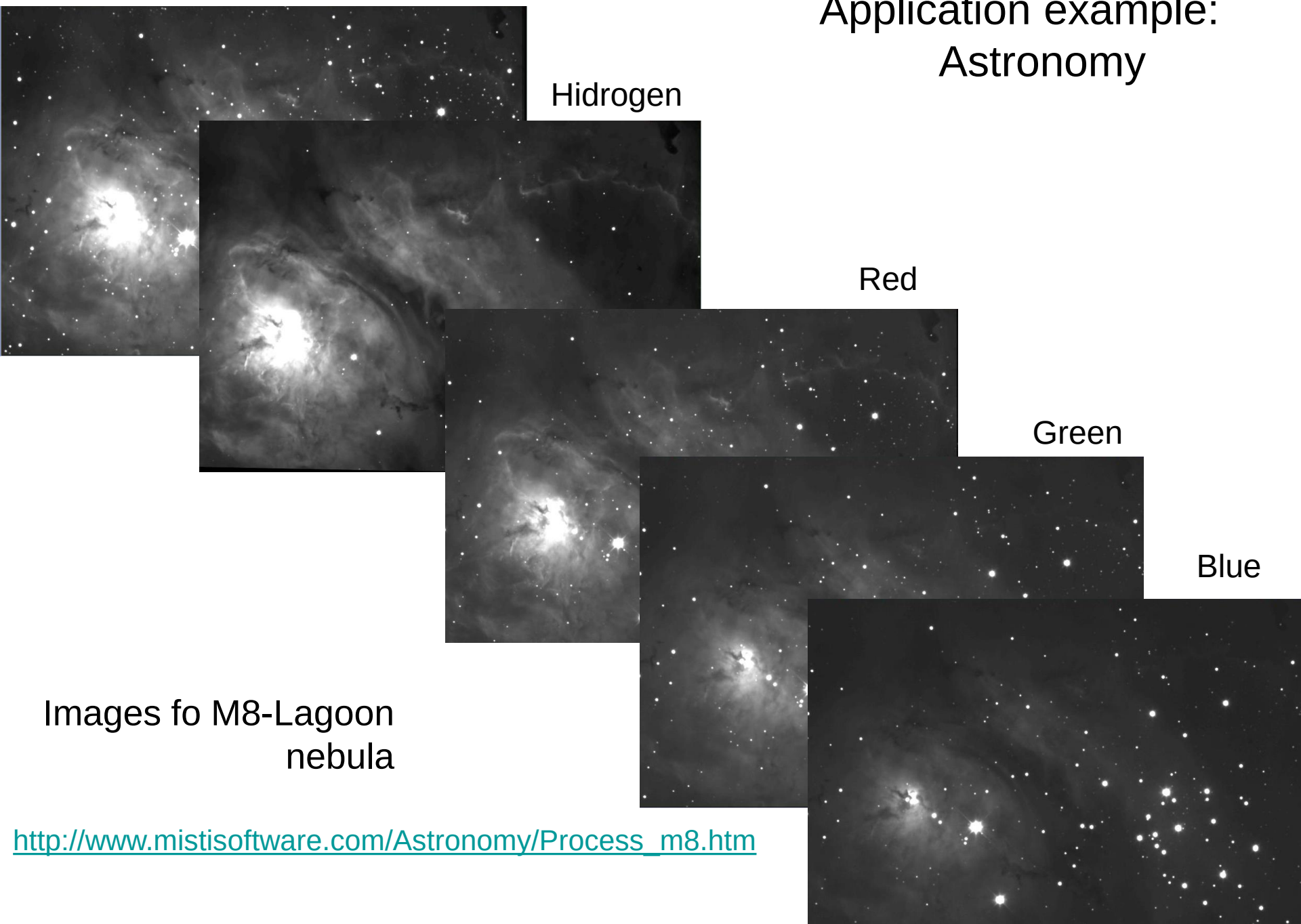
- It is related to other scientific areas



- A traditional goal is to improve an image to make it more amenable to processing:
  - By a human observer
  - By some automatic analysis process

An example of “manual” image processing

Application example: Astronomy



Images fo M8-Lagoon nebula

http://www.mistisoftware.com/Astronomy/Process_m8.htm

M8-Lagoon nebula

Noise reduction

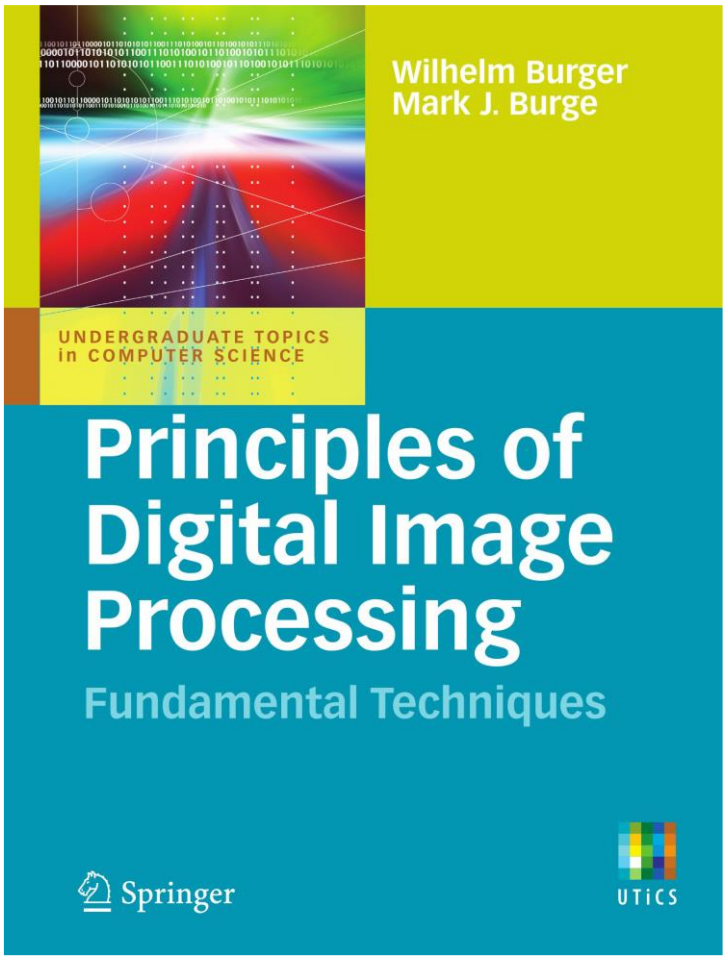
Using a median filter

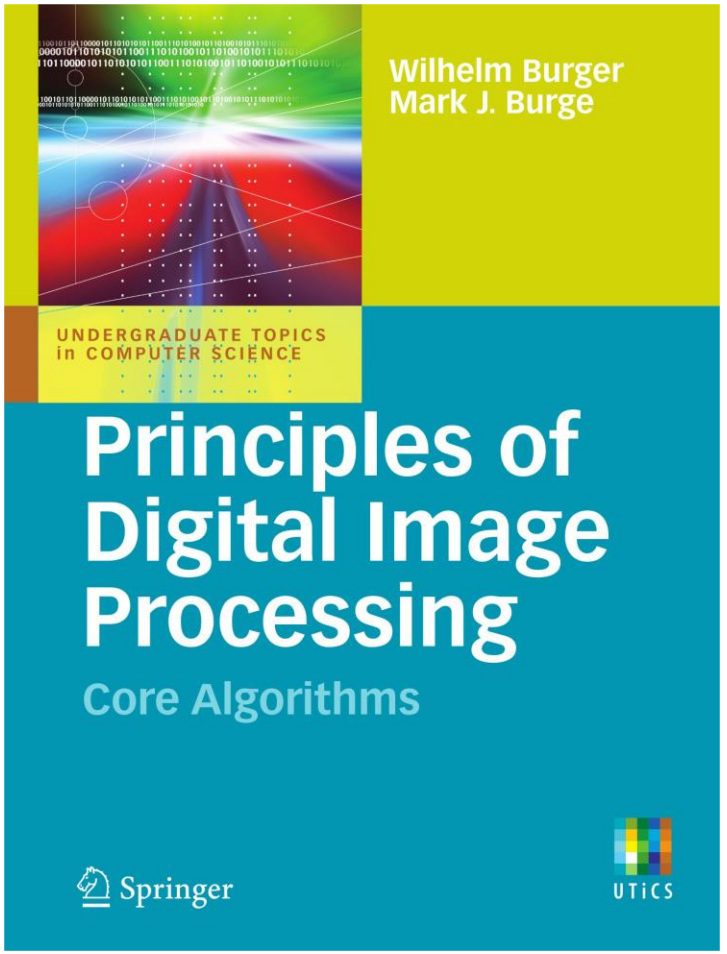
Processing satellite images

Computer Graphics vs Image Processing vs …

		Output	
		Model	Image
Input	Model	Geometric Modeling	Computer Graphics
	Image	Computer Vision	Image Processing

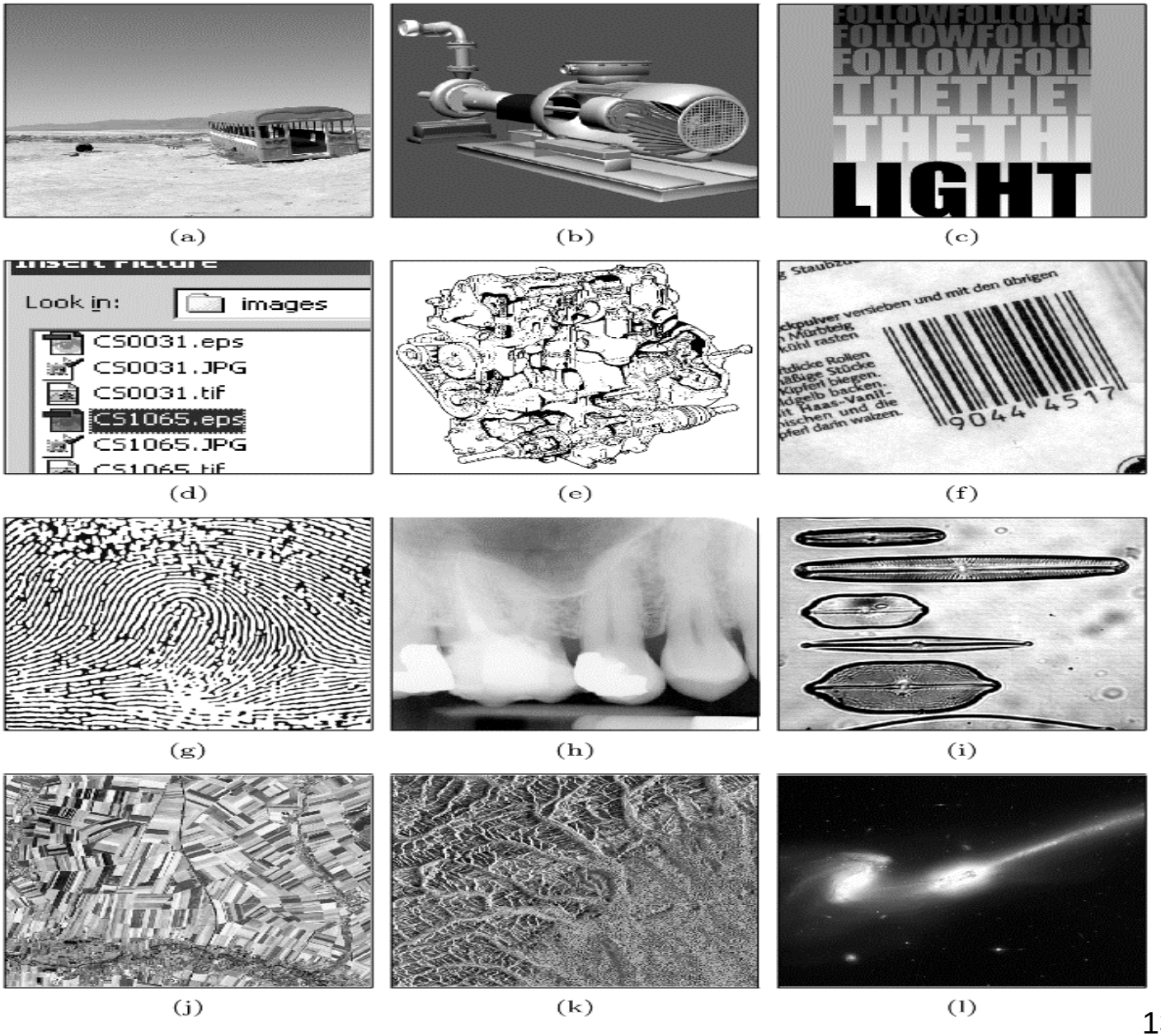
Possible reference books





DIGITAL IMAGES

Digital Images



[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

$F(x, y)$

$\longrightarrow$

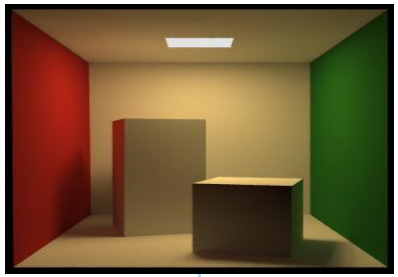
$$
\begin{array}{cccccccccc}
148 & 123 & 52 & 107 & 123 & 162 & 172 & 123 & 64 & 89 \cdots \\
147 & 130 & 92 & 95 & 98 & 130 & 171 & 155 & 169 & 163 \cdots \\
141 & 118 & 121 & 148 & 117 & 107 & 144 & 137 & 136 & 134 \cdots \\
82 & 106 & 93 & 172 & 149 & 131 & 138 & 114 & 113 & 129 \cdots \\
57 & 101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

$I(u, v)$

[Burger & Burge]

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]

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

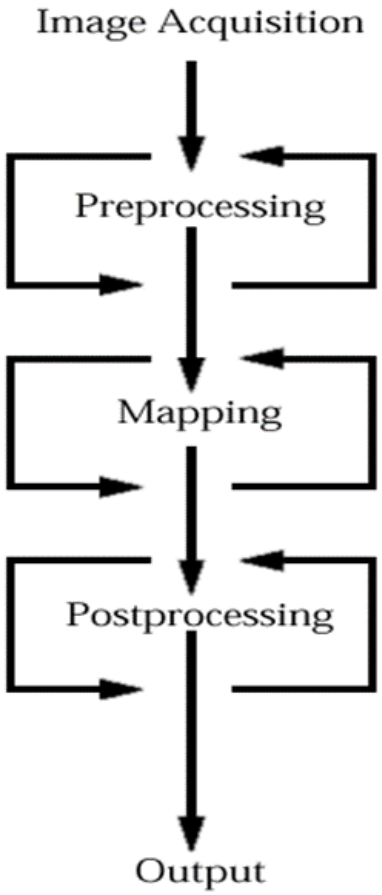
[Burger & Burge]

IMAGE PROCESSING PIPELINE

Idealized Image Processing Pipeline

- Acquisition – obtain images
- Preprocessing – image crop, mask, filter, etc.
- Mapping –image transformations or image composition
- Postprocessing – texturizing, color remapping, etc.
- Output – archiving, printing or displaying on a screen

- In practice, stages may be skipped
- Middle stages are often interlaced



[Andy Van Dam]

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, …)
  - …

[Andy Van Dam]

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

[Andy Van Dam]

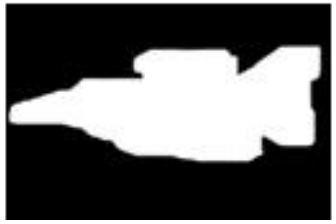
Preprocessing

- Adjusting color or grayscale curve


- Cropping


- Masking (cutting out part of an image)



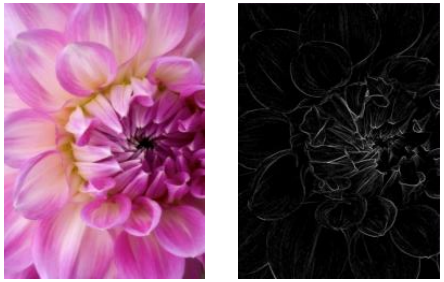



[Andy Van Dam]

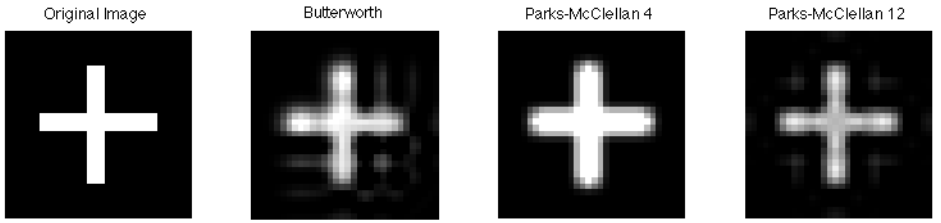
Preprocessing

- Blurring / Sharpening

- Edge detection / enhancement



- Filtering / Anti-aliasing

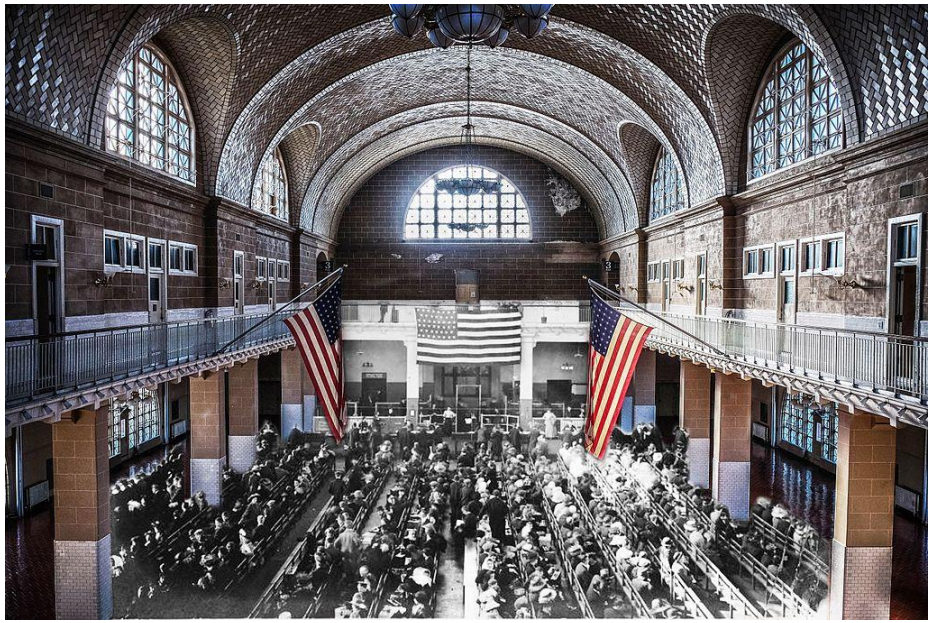


- Super-sampling / Sub-sampling

[Andy Van Dam]

Mapping

- Apply geometrical transformations
  - Rotating / Scaling / Shearing / …
  - Warping / Morphing
- Combine images – compositing
  - Overlay
  - Smooth blending
  - …



[Wikipedia]

Postprocessing

- Create global effects across an entire image or selected area
- Art effects
  - Posterizing / Ageing / …

- Technical effects
  - Color remapping / Conversion to Gray or B&W
  - Color separation for printing
  - Color / contrast balancing
  - …

[Andy Van Dam]

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

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

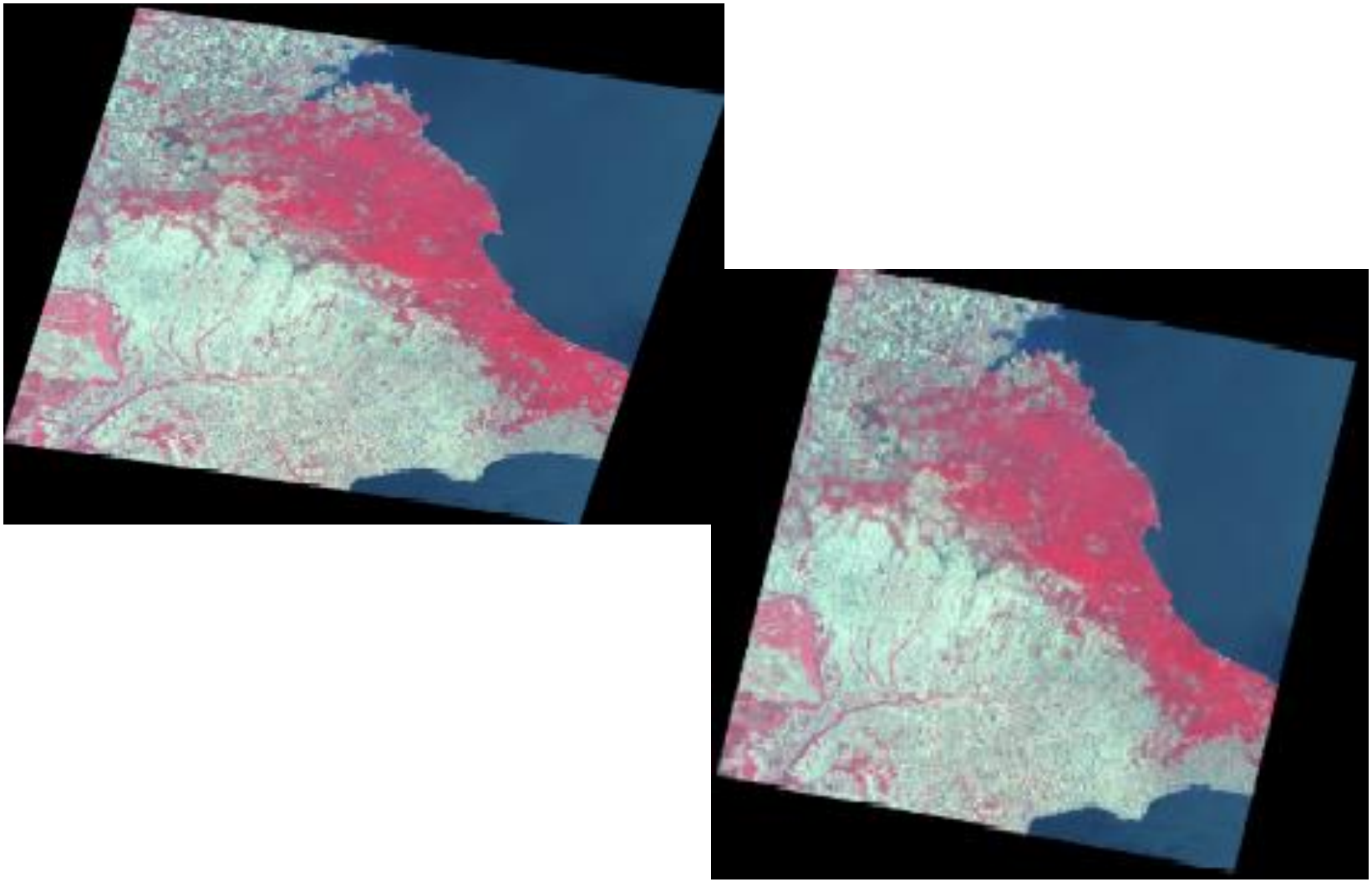
- Another possibility is to operate on pixel position and not on intensity or color values:

  - Correction of geometric distortion due to acquisition effects

  - Distortion of an image into another (*image* **warping**, **morphing**)

  - **Texture mapping**

Geometric correction in satellite imagery



Texture mapping

http://cgkit.sourceforge.net/tutorials/baking/baking.html

INTENSITY TRANSFORMATIONS

Pixel Processing

- One of the simplest operations that can applied to transform an image
- A transformation is applied to the values (intensity, color) associated with image pixels
- Such a transformation can be established based on global image features (usually the histogram)

  or

- Considering the pixel values in the neighborhood of each image pixel

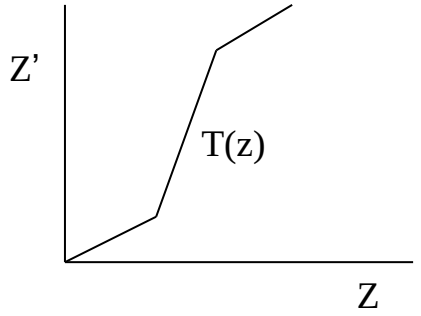
1. Intensity transformations

- The same transformation is applied to all image pixels
- **Contrast stretching is one of the mostly used transformations**
- For grayscale images that do not use the entire dynamic range (possible gray values), the histogram can be modified by a transformation function T

$$z' = T(z)$$

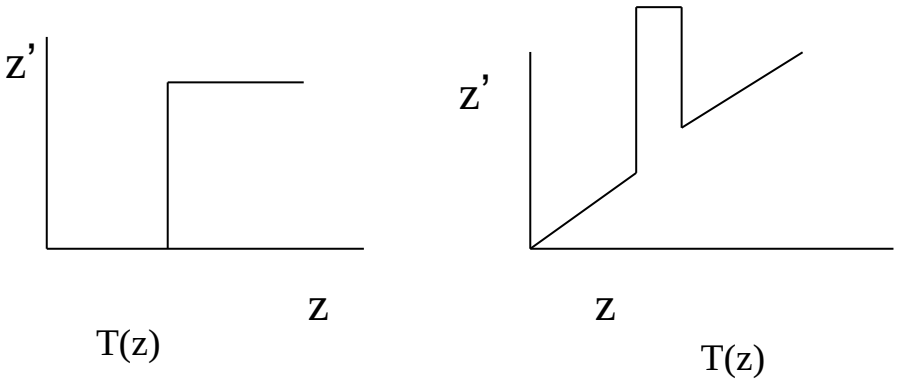
New gray value

Original gray value

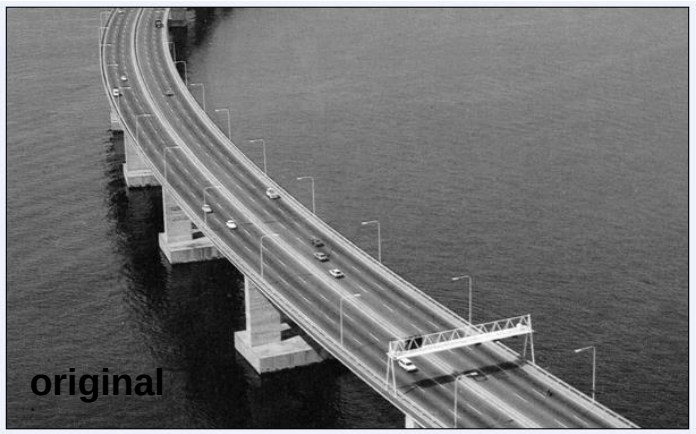


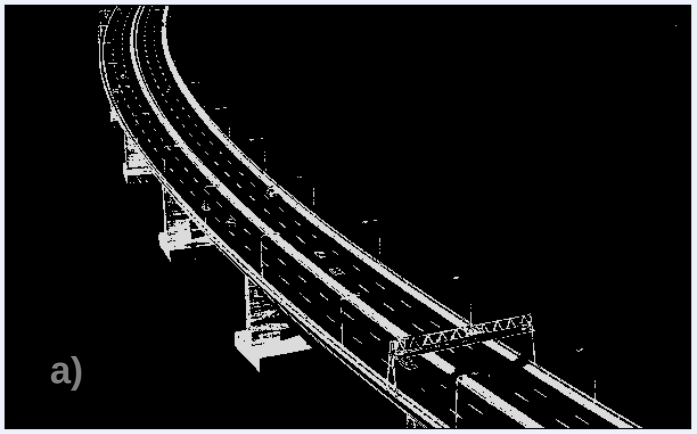
2. **Intensity level slicing**

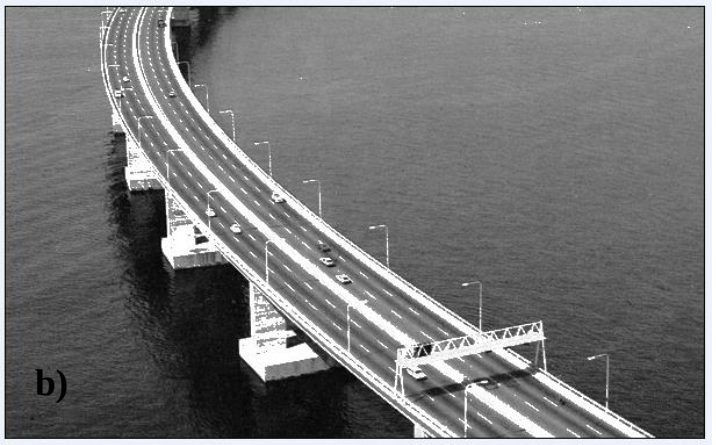
- Highlighting a contiguous set of intensity values
- **Thresholding** is a particular case and can be used as a segmentation technique



a) Thresholding *b) Highting an intensity band of interest*







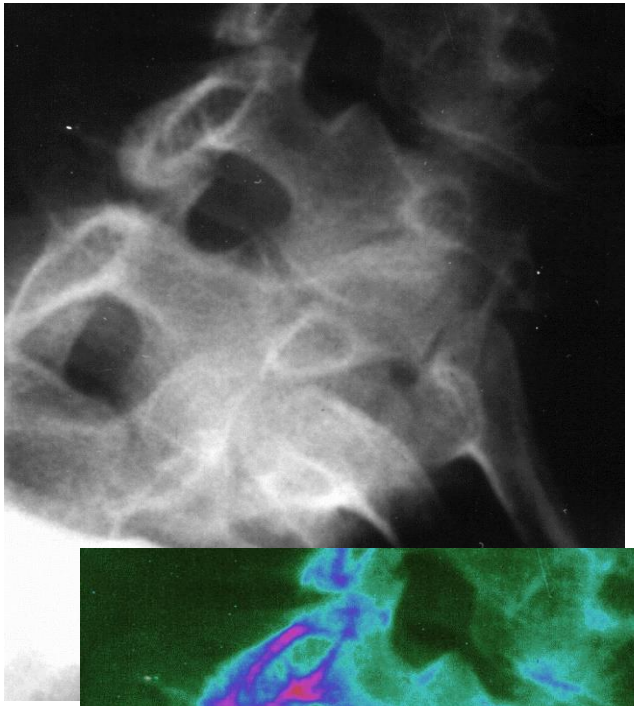
3. Intensity level to color – pseudo-color

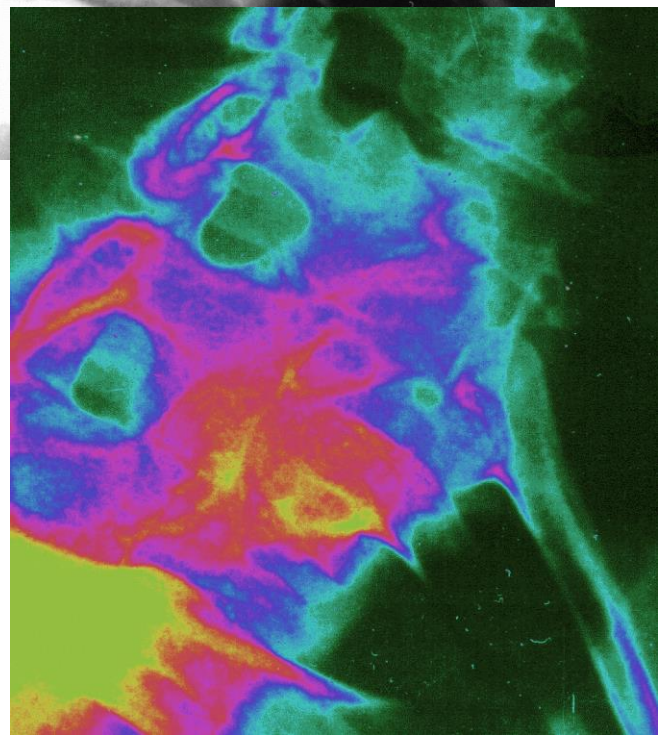
- Transform a grayscale image into a color image
- Recodification of the original image

$$I_R(x,y) = T_1(I(x,y))$$
$$I_G(x,y) = T_2(I(x,y))$$
$$I_B(x,y) = T_3(I(x,y))$$

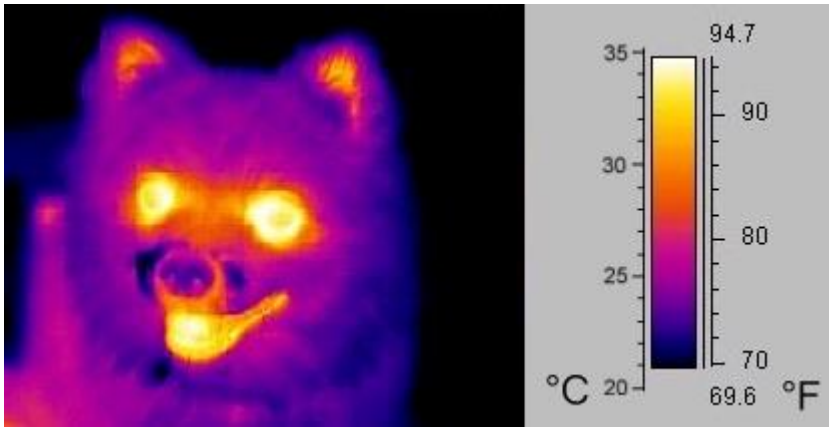
- Use in images that are not photos and depict some kind of data (Data Visualization)
- Use with care, since false contours might be perceived

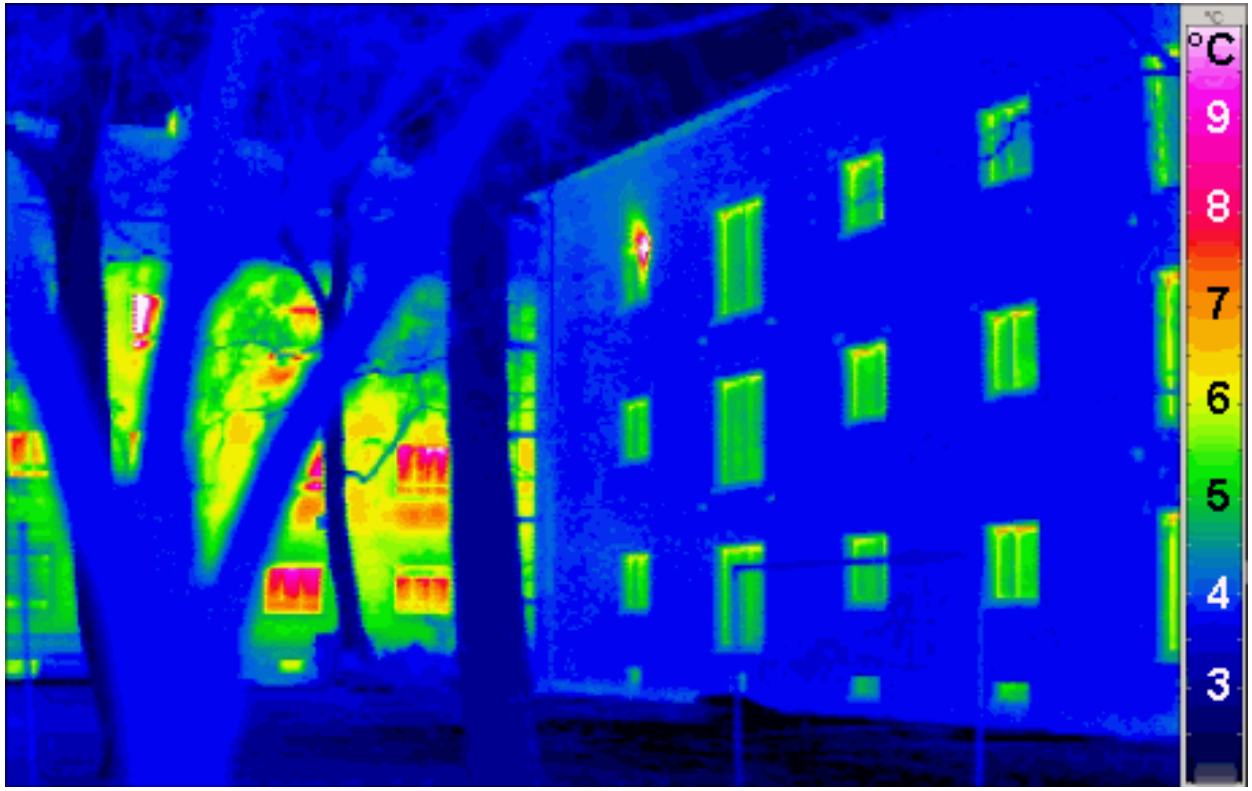
Example – “rainbow” color-scale is not adequate for most situations, although it is often used





Pseudo-color examples (thermography)





Pseudo-color vs *false-color*

- Given an original image, with 3 non-RGB channels, the false-color transforms that image into an RGB image
- E.g., infrared, ultraviolet

- Recodification of the original image

$$I_R(x,y) = T_1(I_1(x,y))$$
$$I_G(x,y) = T_2(I_2(x,y))$$
$$I_B(x,y) = T_3(I_3(x,y))$$

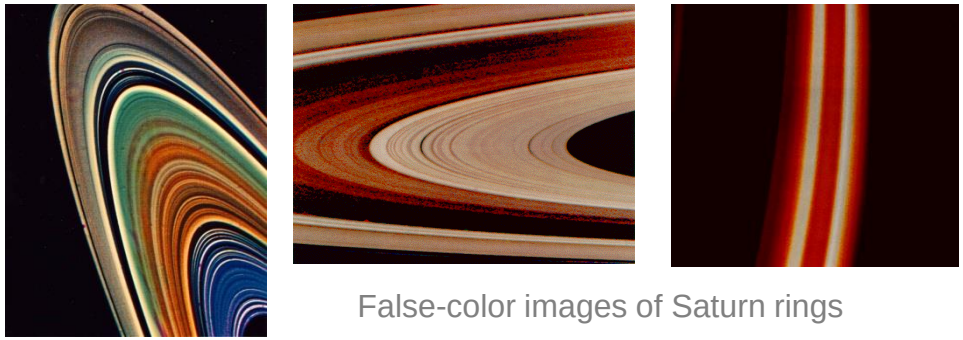
- Used in remote detection imagery (e.g., Landsat)

Example – “true-color” and “false-color” images of Chesapeake Bay and Baltimore

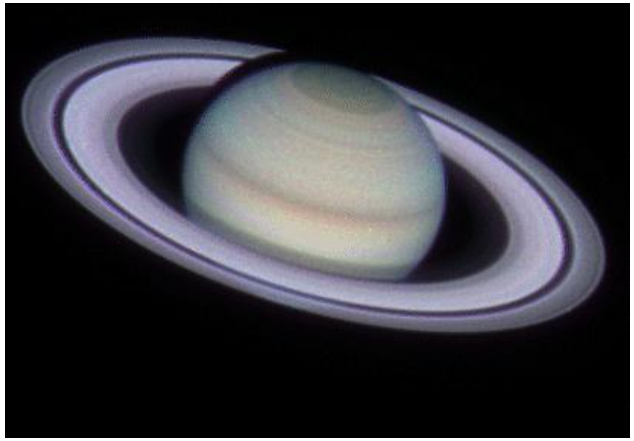
False-color examples

color enhancing

http://www.windows.ucar.edu/tour/link=/saturn/saturn_il.html

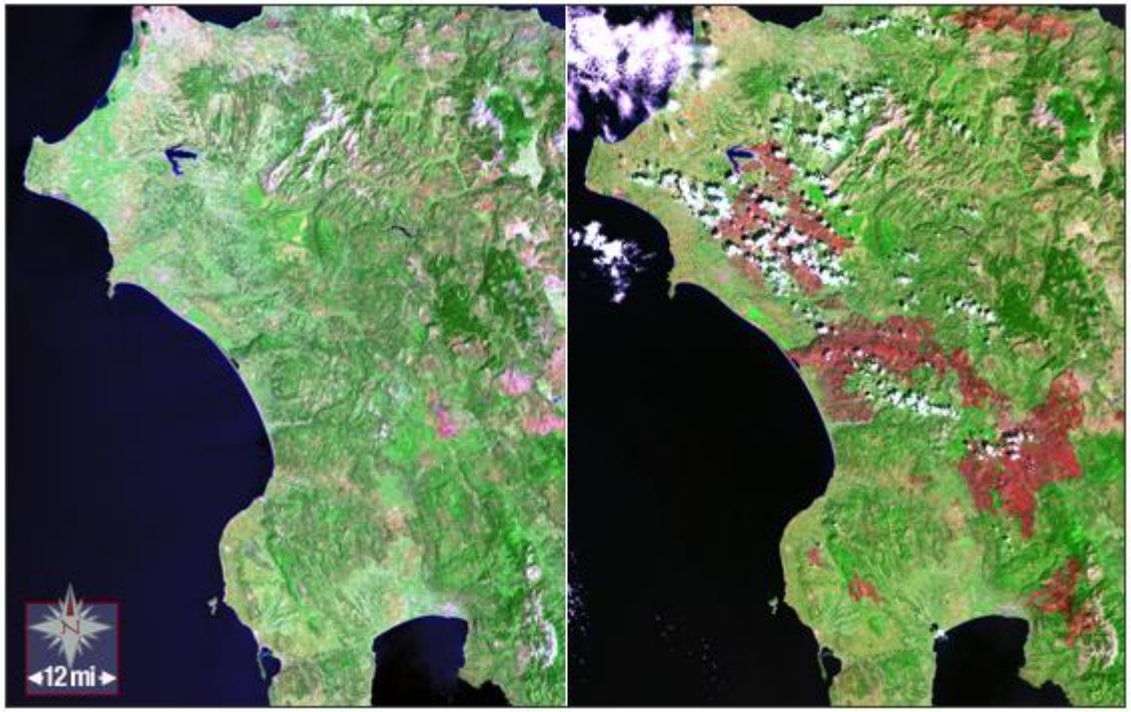


False-color images of Saturn rings



color image of Saturn taken on August 26, 1990

http://landsat.gsfc.nasa.gov/images/archive/e0012.html

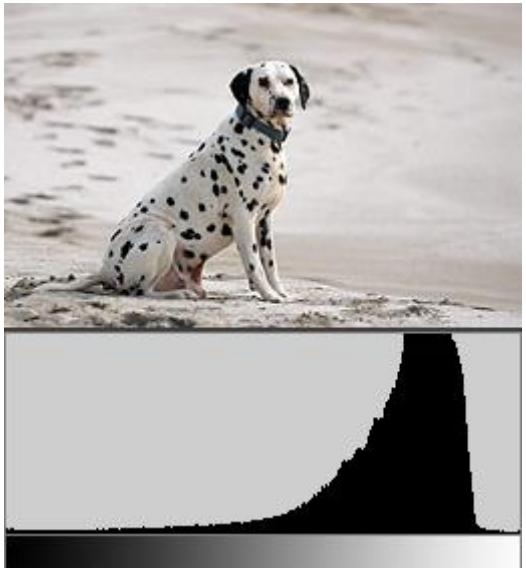


Landsat 7 images of southwestern Greece before (left, 7/18/07) and after (right, 9/4/07) this summer's severe fires.

False-color view of Saturn and its rings Voyager 1 spacecraft on Oct. 18, 1980

HISTOGRAM-BASED PROCESSING

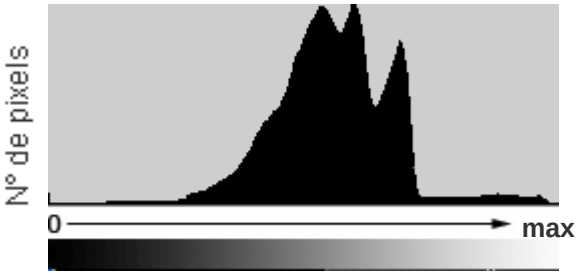
4. Histogram-based processing

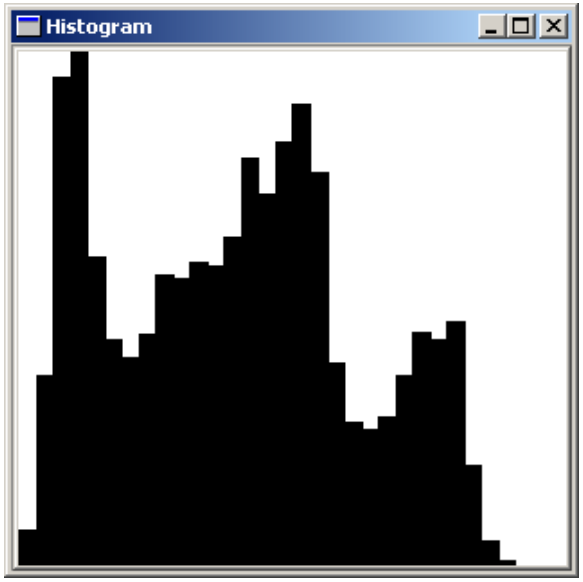


- A histogram provides a general idea of the image gray levels and allows choosing an adequate processing operation
- Ideally an image histogram should occupy the entire grayscale
- Histogram-based processing has two stages:
  - Computing and analyzing the histogram
  - Image enhancement based on histogram features

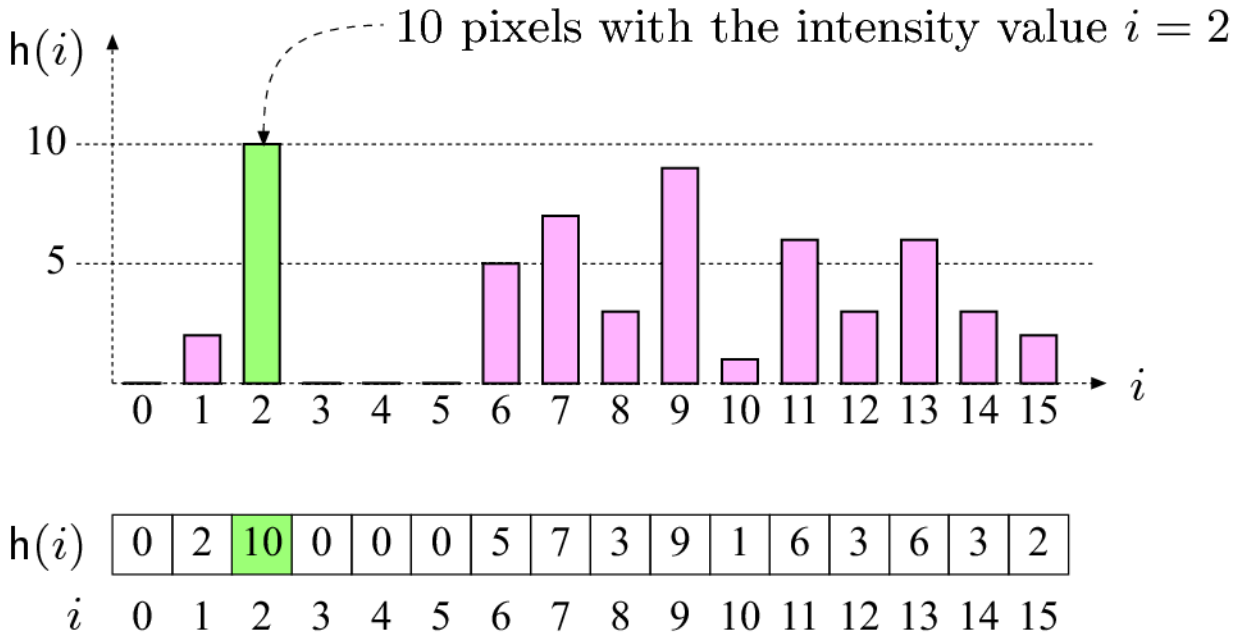
Image histogram

- A histogram provides a general idea of the image gray levels



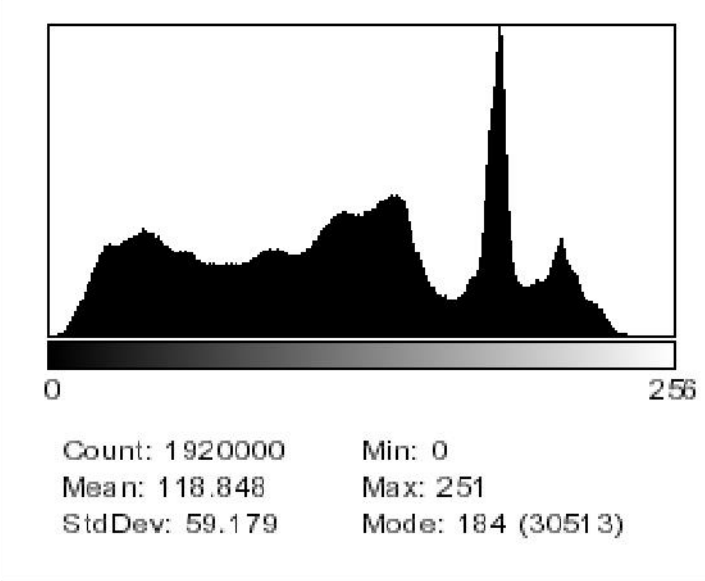


Histogram – How to compute?



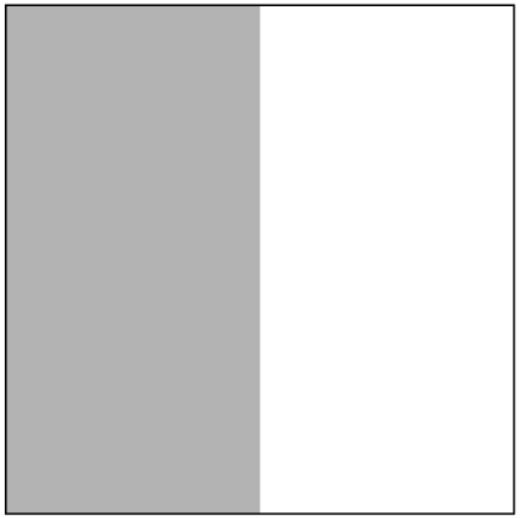
[Burger & Burge]

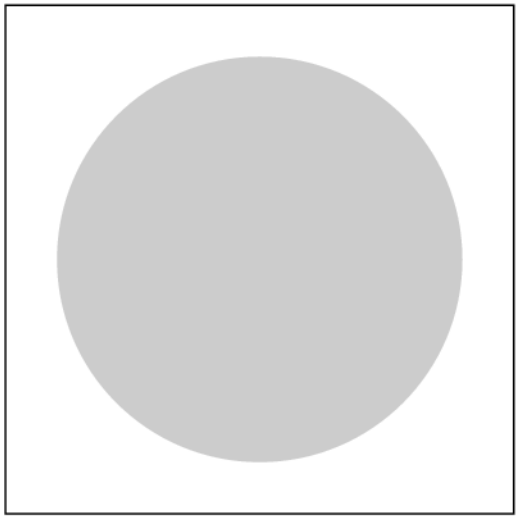
Histogram – Another example

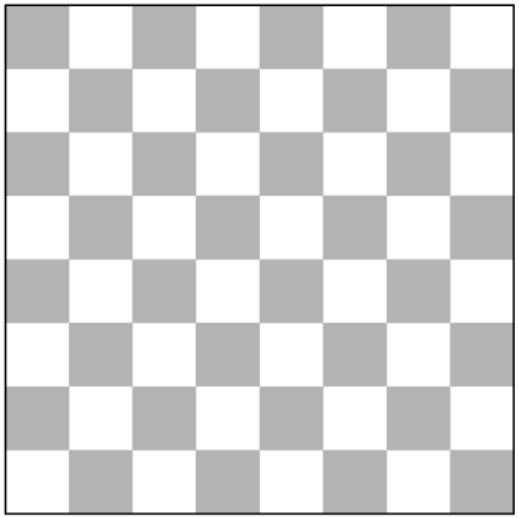


[Burger & Burge]

Same or different histogram?







[Burger & Burge]

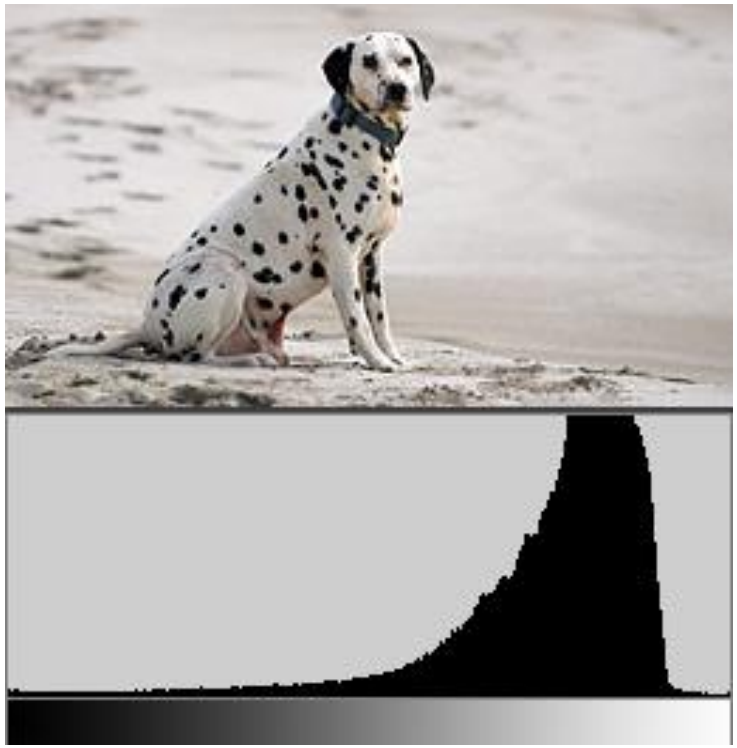
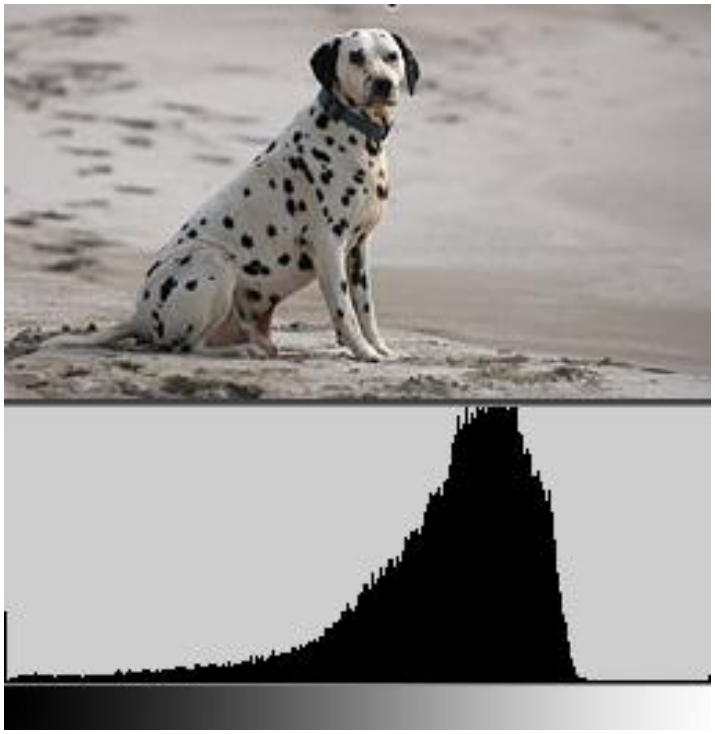


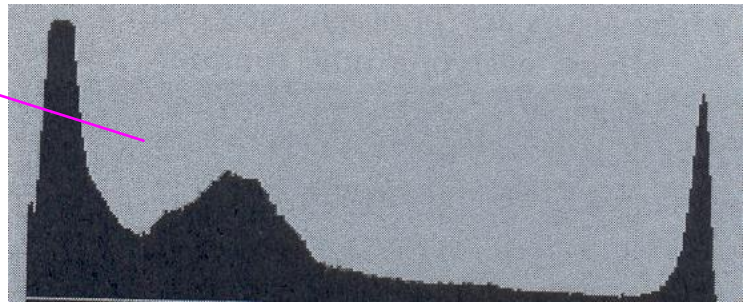
Image and histogram



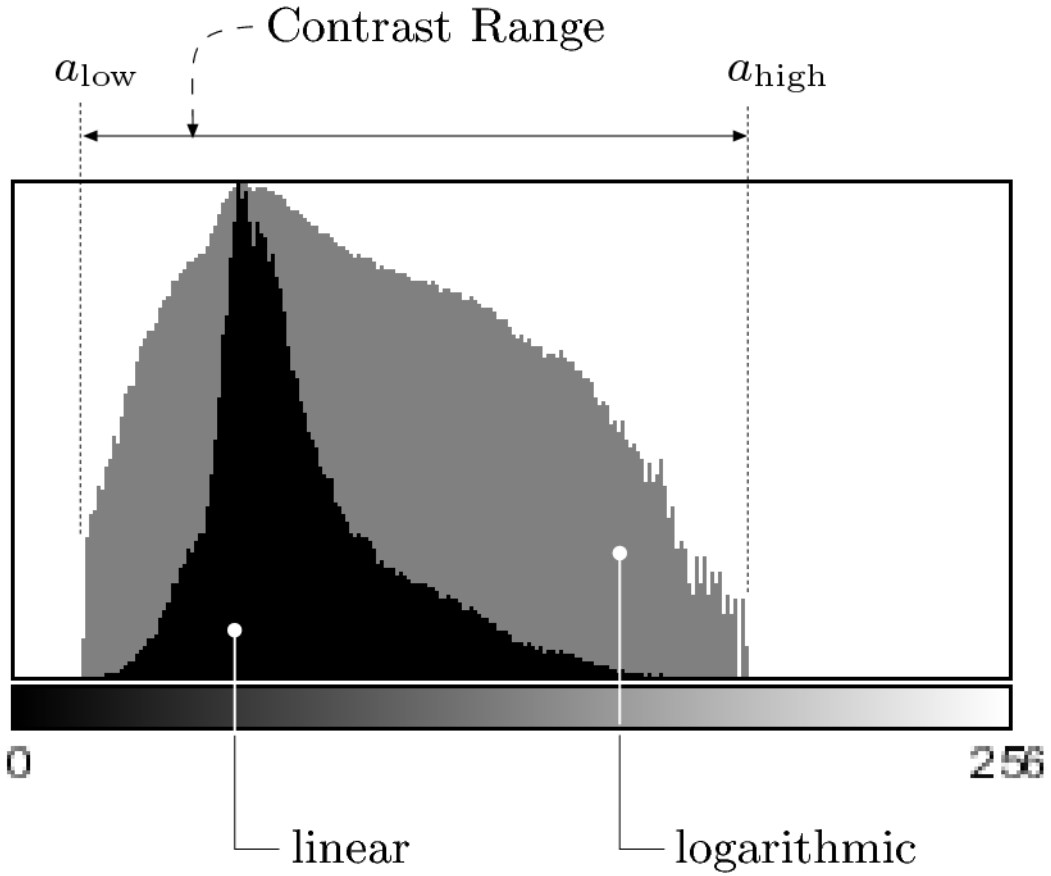
Underexposed image:
slightly more dark

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

What is the histogram of each image?

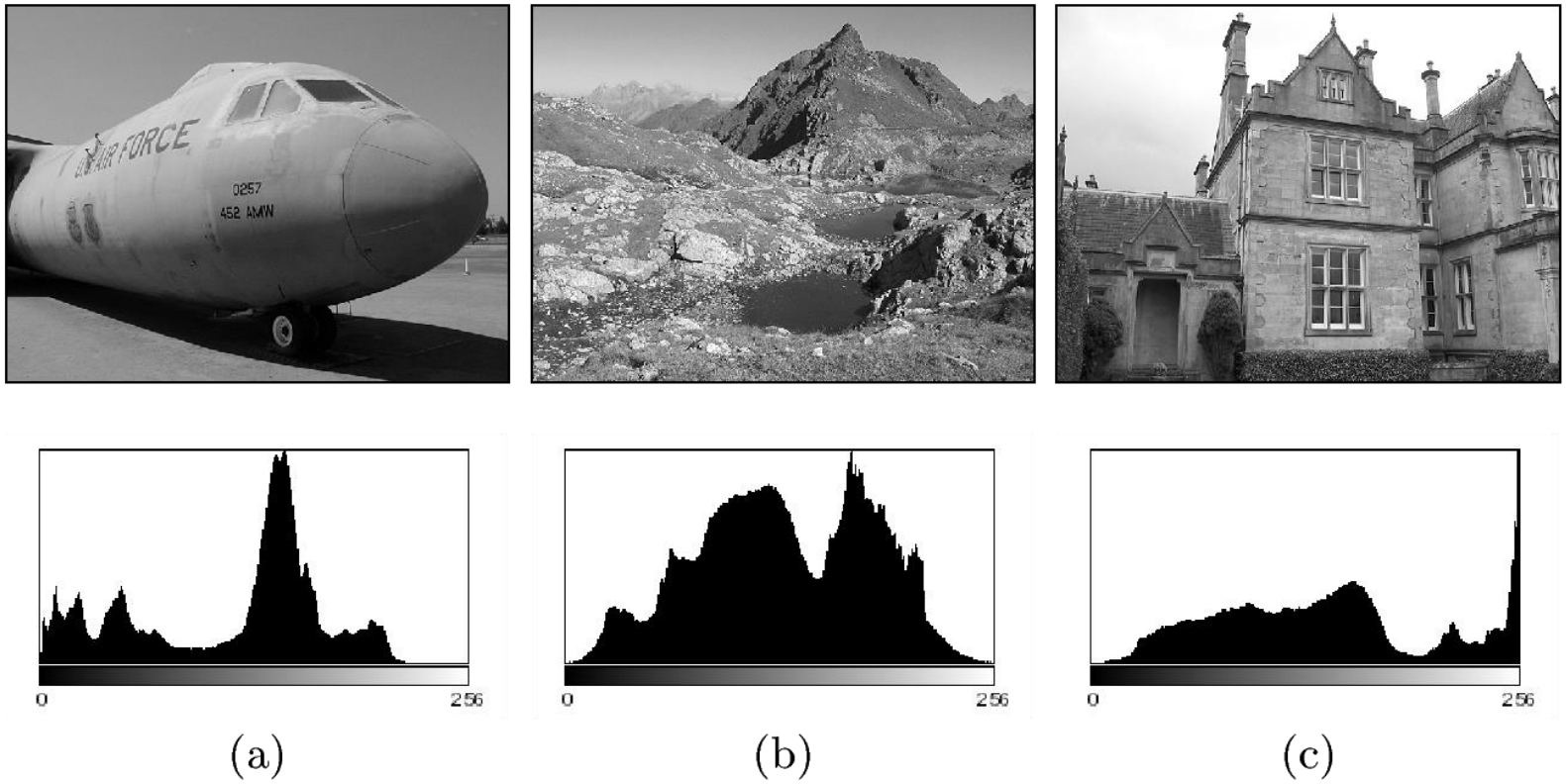


Linear vs Logarithmic representation



[Burger & Burge]

Image features – Exposure



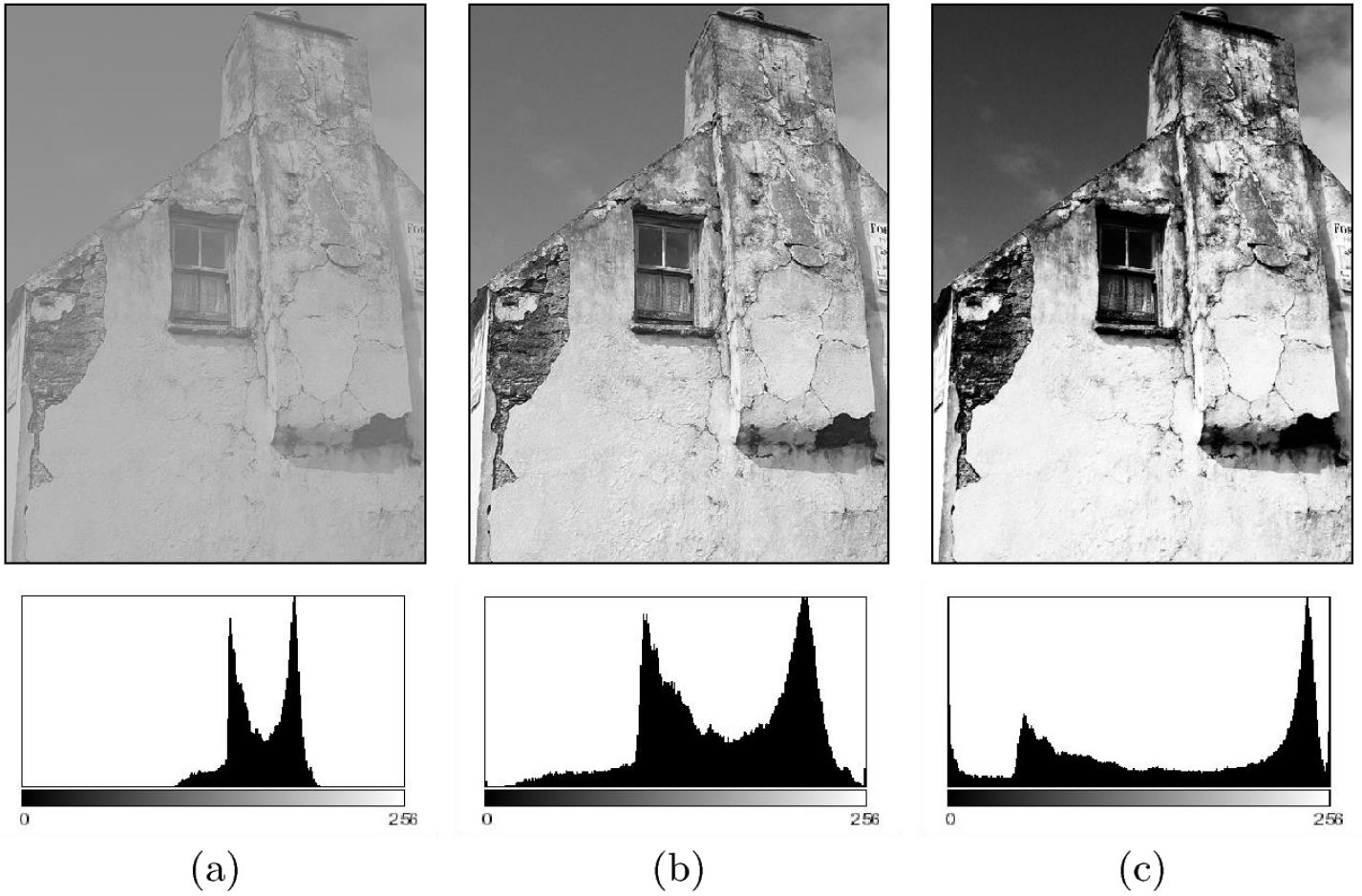
(a) (b) (c)

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

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

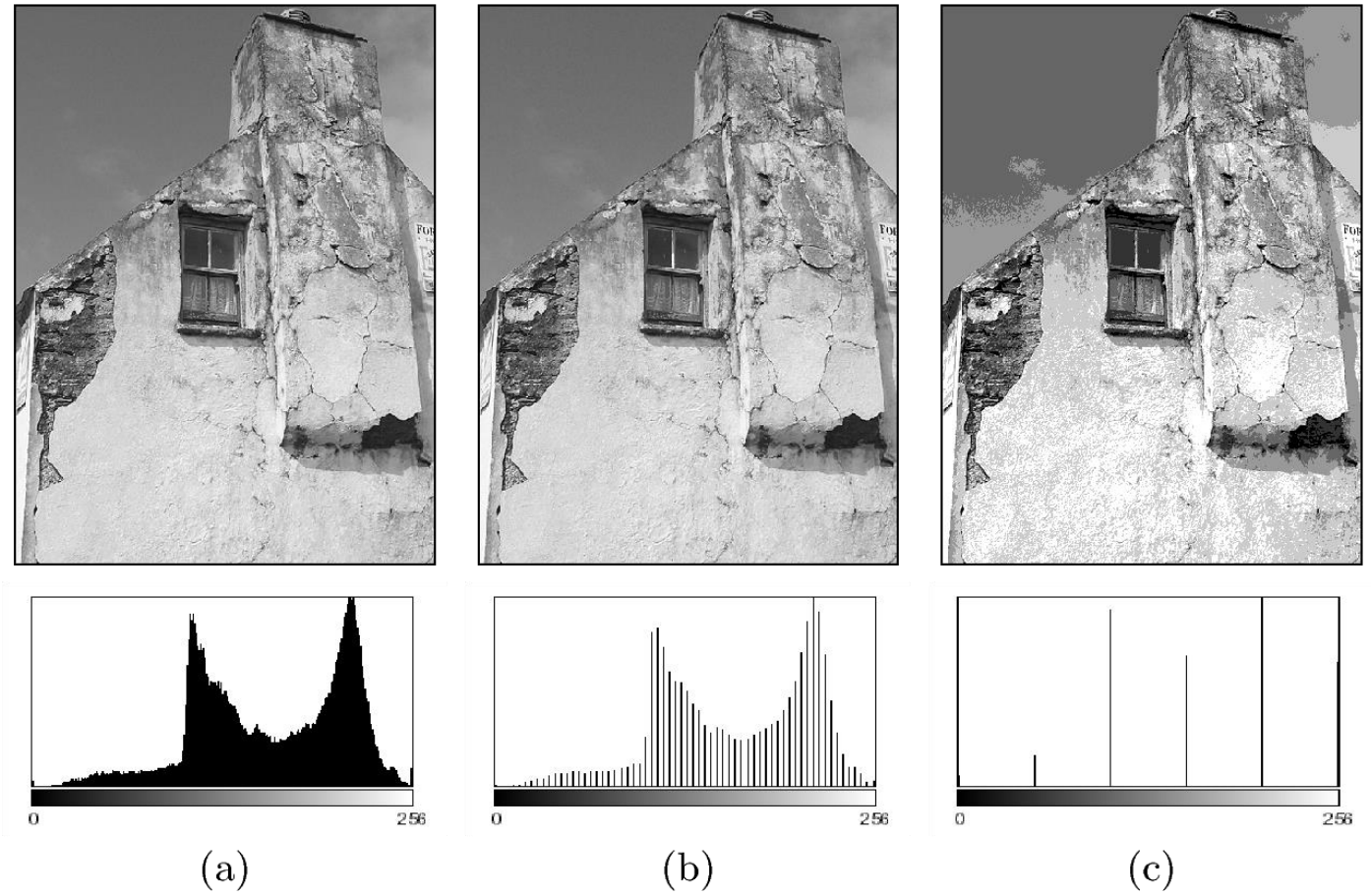
[Burger & Burge]

Image features – Contrast



[Burger & Burge]

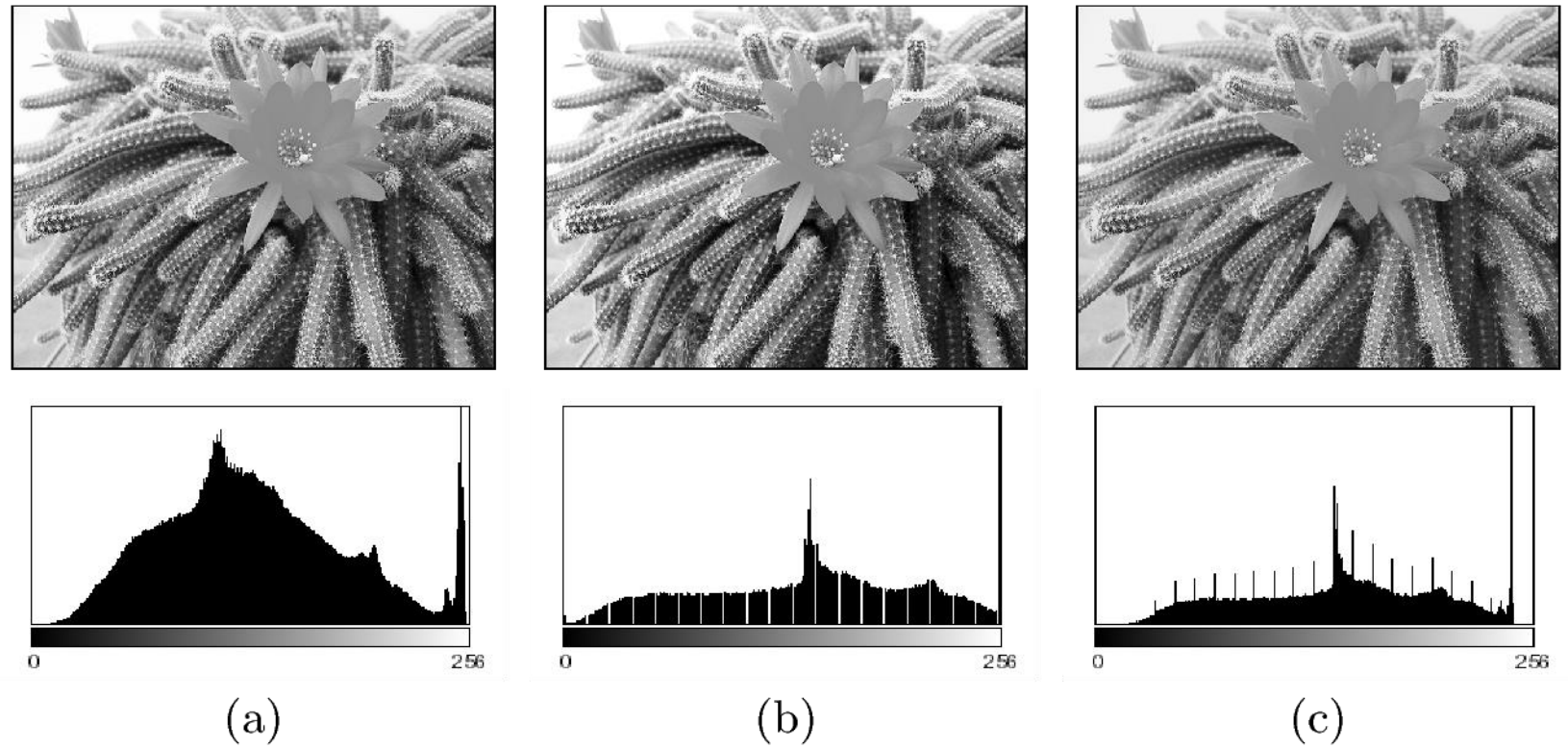
Image features – Number of intensity levels



256, 64 and 6 distinct intensity levels

[Burger & Burge]

Problems – Saturation

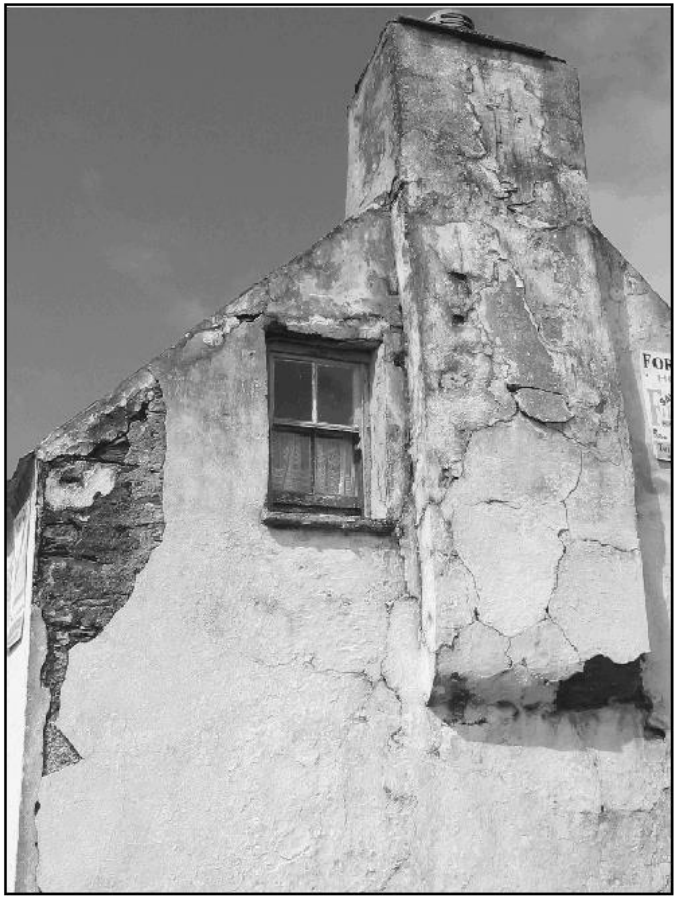


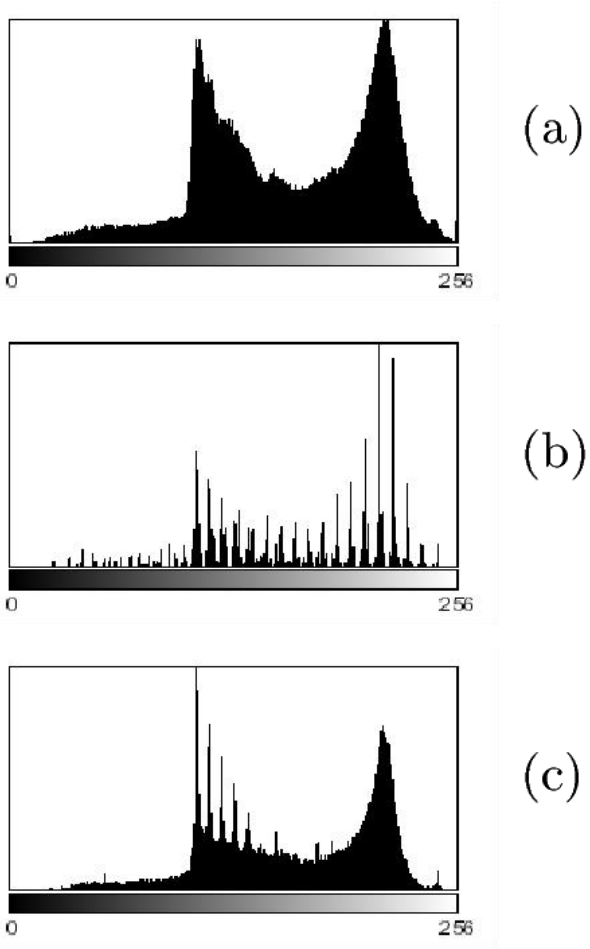
Saturation at the right-end of the histogram !!

Results of unsuccessful correction attempts (contrast modifications)

[Burger & Burge]

Problems – Image compression (GIF)





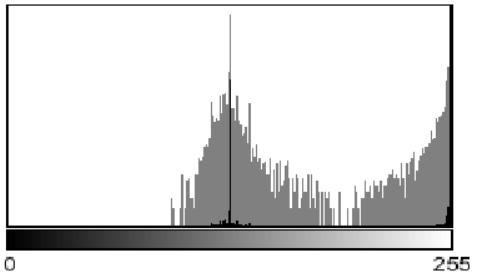
[Burger & Burge]

Problems – Image compression (JPEG)

(a)

(b)

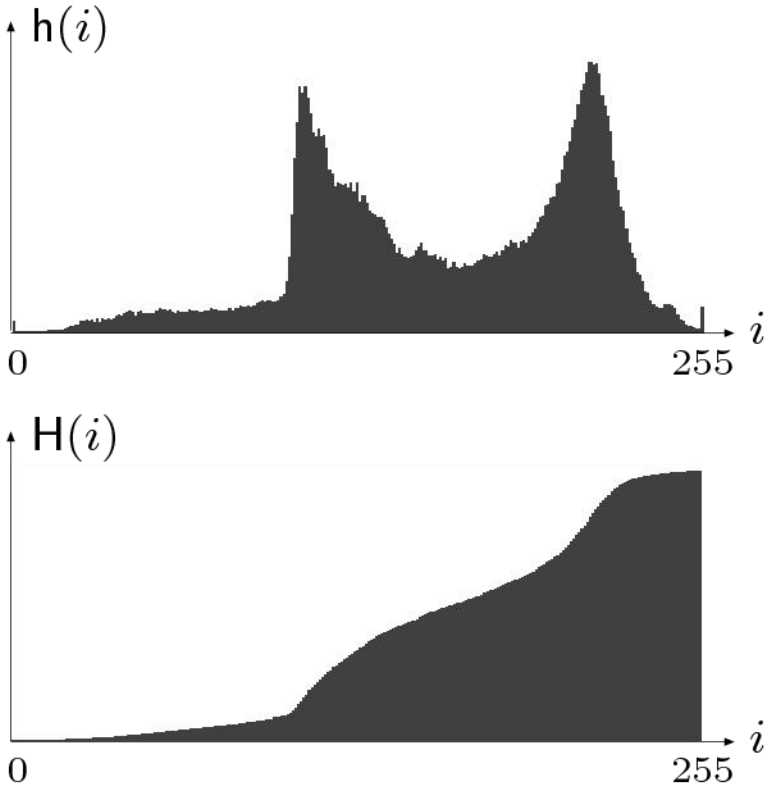
(c)



(d)

[Burger & Burge]

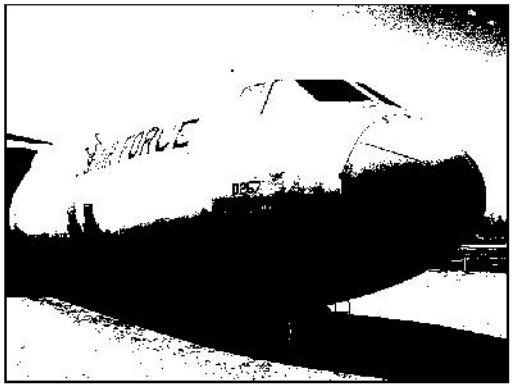
Cumulative histogram



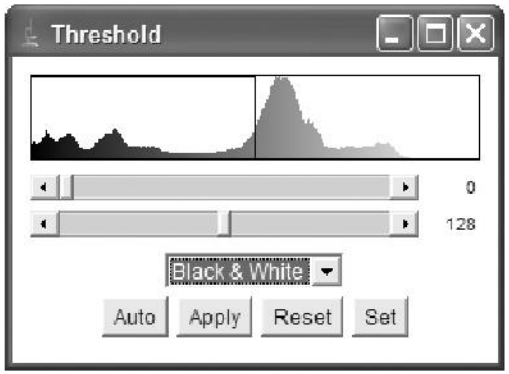
[Burger & Burge]

Thresholding – Another example

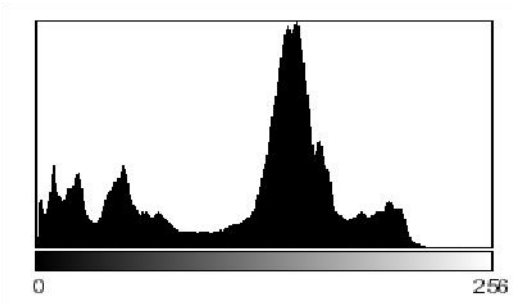
(a)



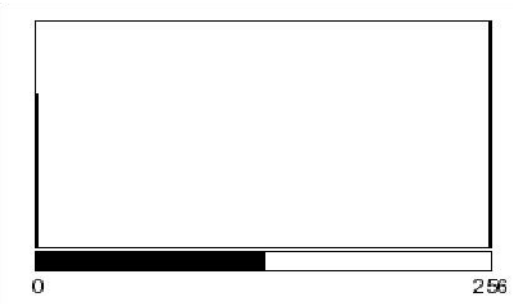
(b)



(e)



(c)

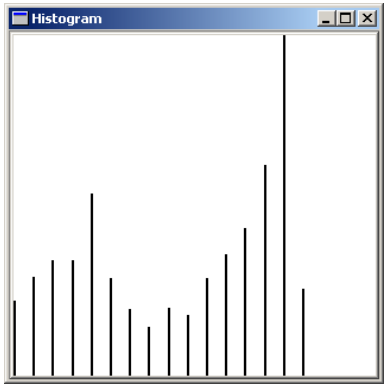


(d)

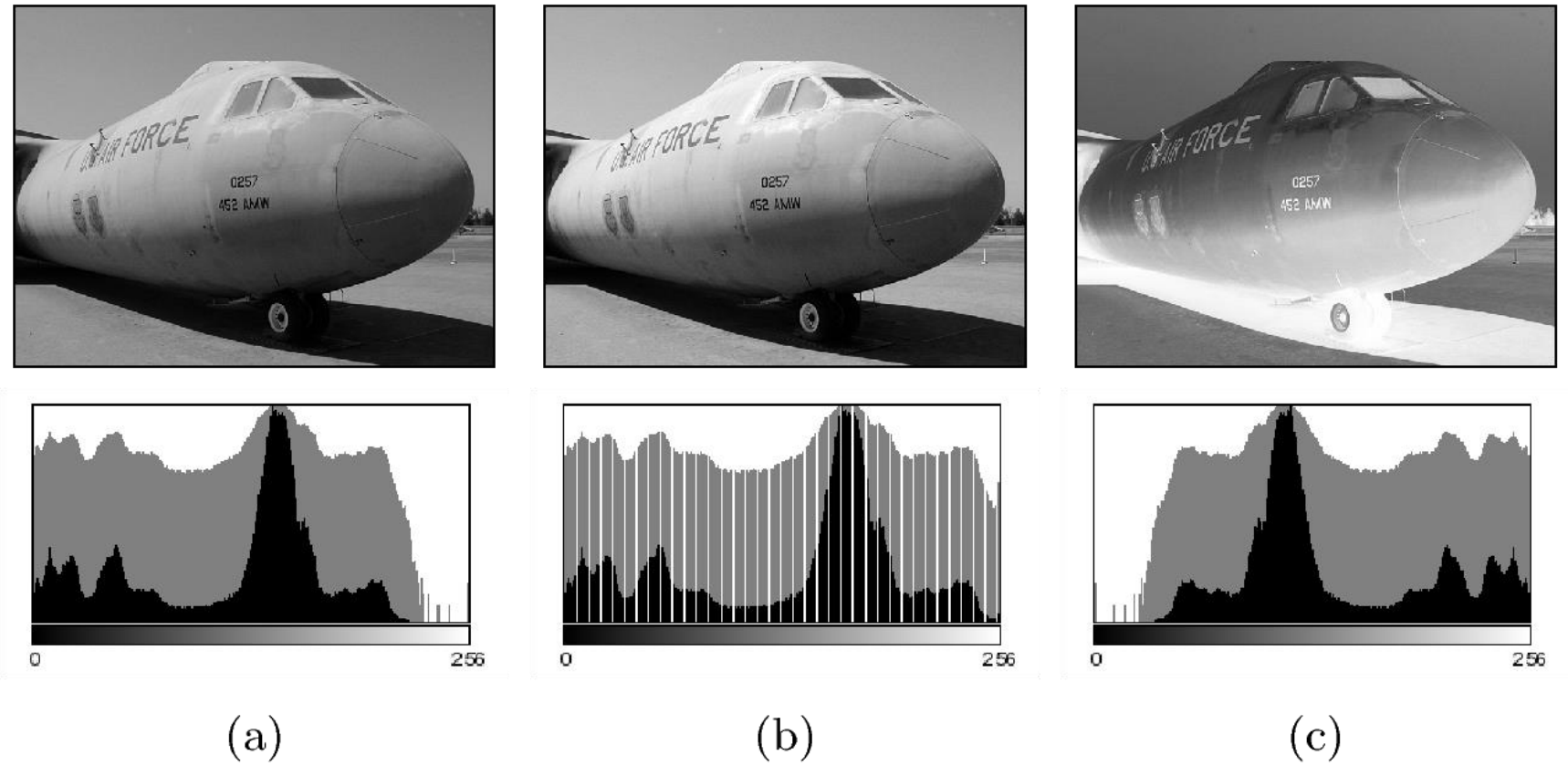
[Burger & Burge]

Contrast-stretching – Another example

$$final[x,y] = \frac{original[x,y] - \min}{\max - \min} \times 255$$



Another example

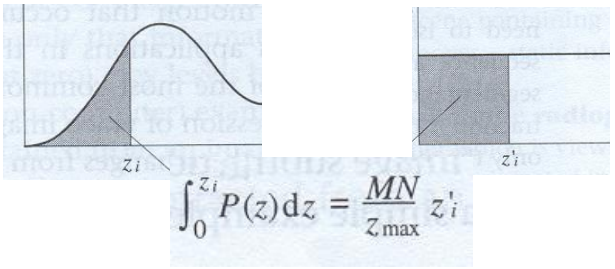


[Burger & Burge]

5. Histogram Equalization

- It can be regarded as an extension of ***contrast-stretching***
- 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 $Q(z)$

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



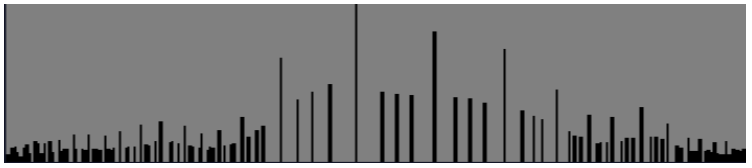
$$z_i' = T(z_i) = \frac{z_{max}}{M\,N} \int_0^{z_1} P(z)\,dz$$

M x N – nº de pixels da imagem

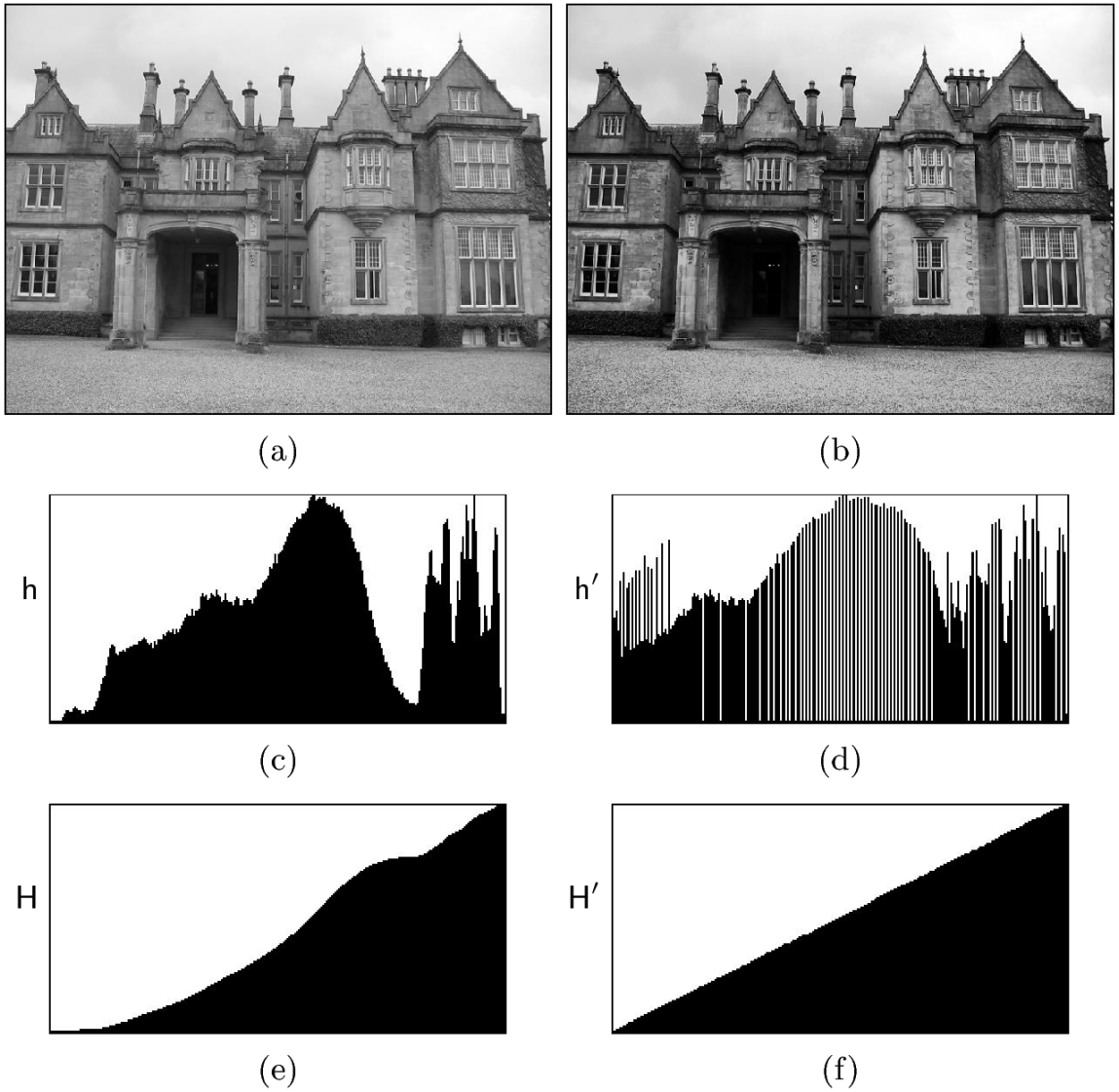
$$z_i' = \frac{z_{max}}{M\,N} \sum_0^{z_1} H(z)\,dz$$

é a aproximação discreta e
H é o histograma discreto

Histogram Equalization – Example



Linear Histogram Equalization



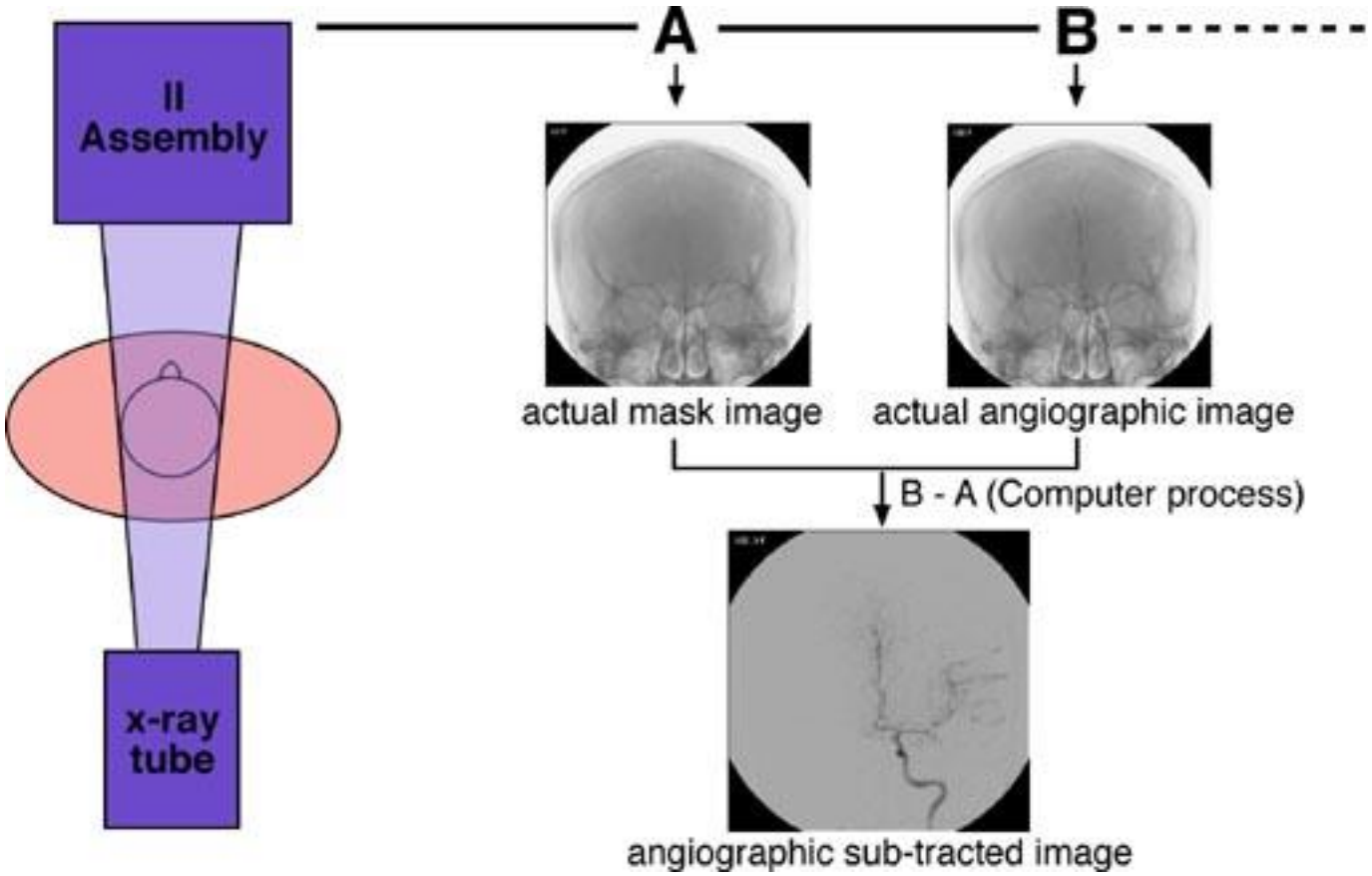
[Burger & Burge]

IMAGE SUBTRACTION

6. Image Subtraction

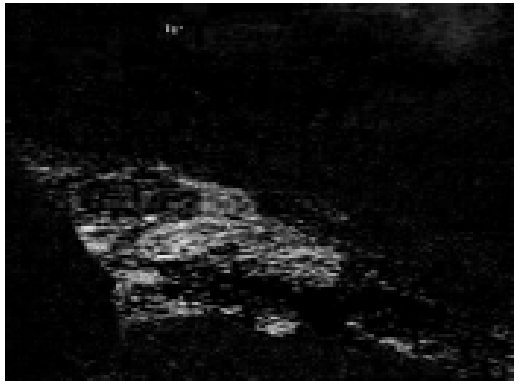
- One of the simplest *pixel processing* operations
- Can be used to isolate movement happening between two successive images
- As well as for image segmentation and enhancement

Example – Digital Subtraction Angiography



http://www.droid.cuhk.edu.hk/web/service/angio/dsa.htm

Estimating flow velocity after a typhoon



http://www.aars-acrs.org/acrs/proceeding/ACRS2006/Papers/H-2_P70.pdf

FILTERING

7. Image smoothing

- Also called *neighbourhood averaging*, can be used to reduce image noise
- Or as pre-processing step prior to edge detection
- It is a simple filtering operation, where the filter kernel slides over the original image and the intensity (gray level) of the resulting pixel in the final image is computed as:

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

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

Pixel intensity values in the original image

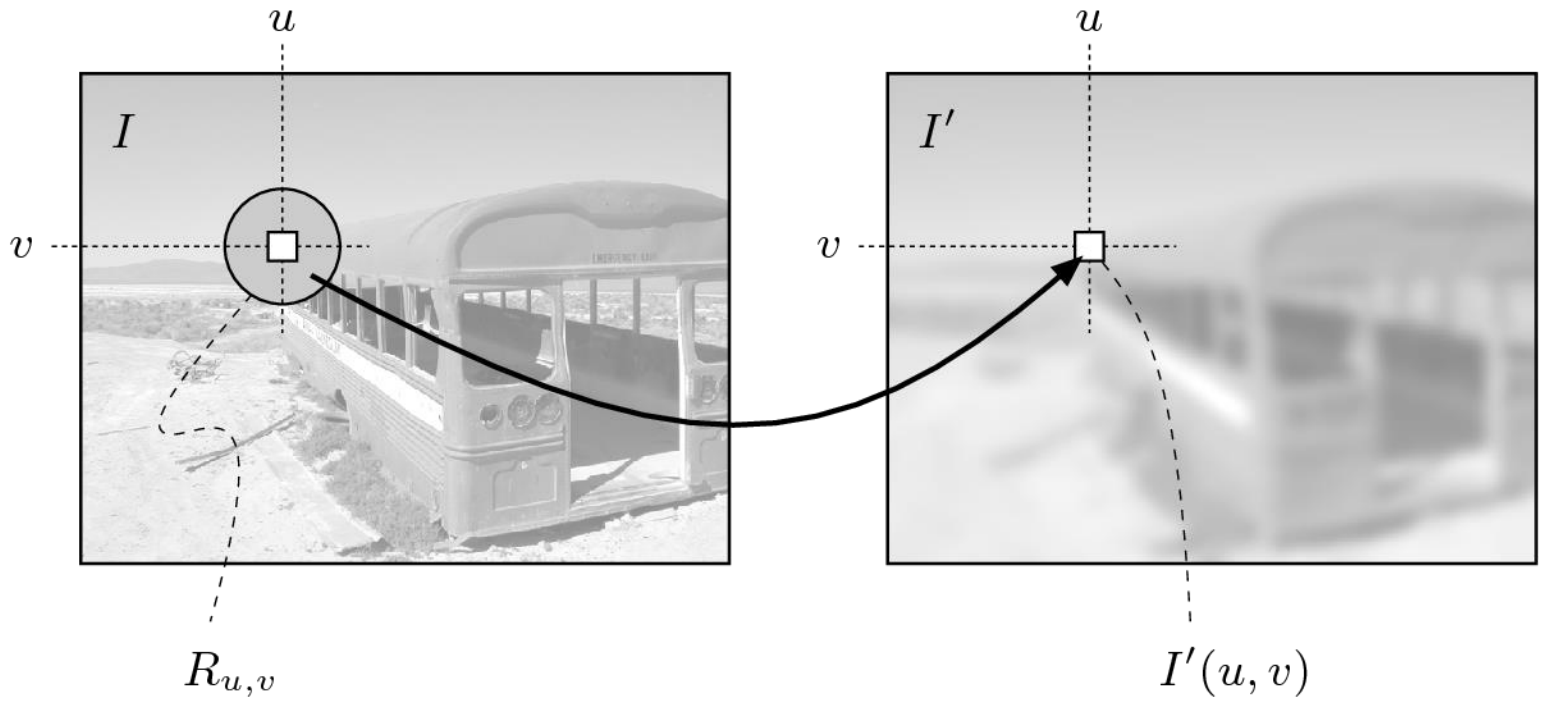
Filter weights

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

Smoothing / Blurring

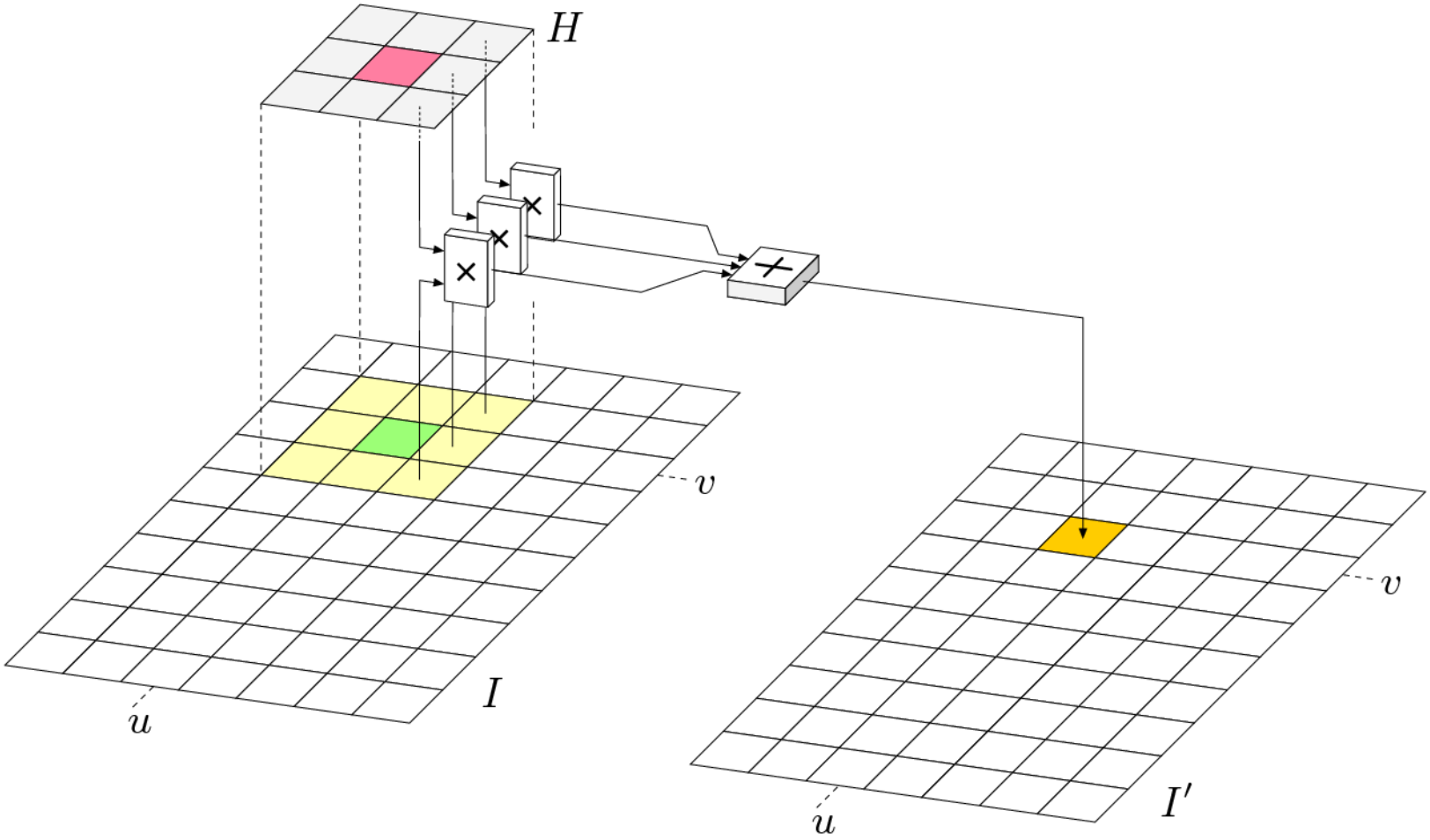
[Burger & Burge]

Operation



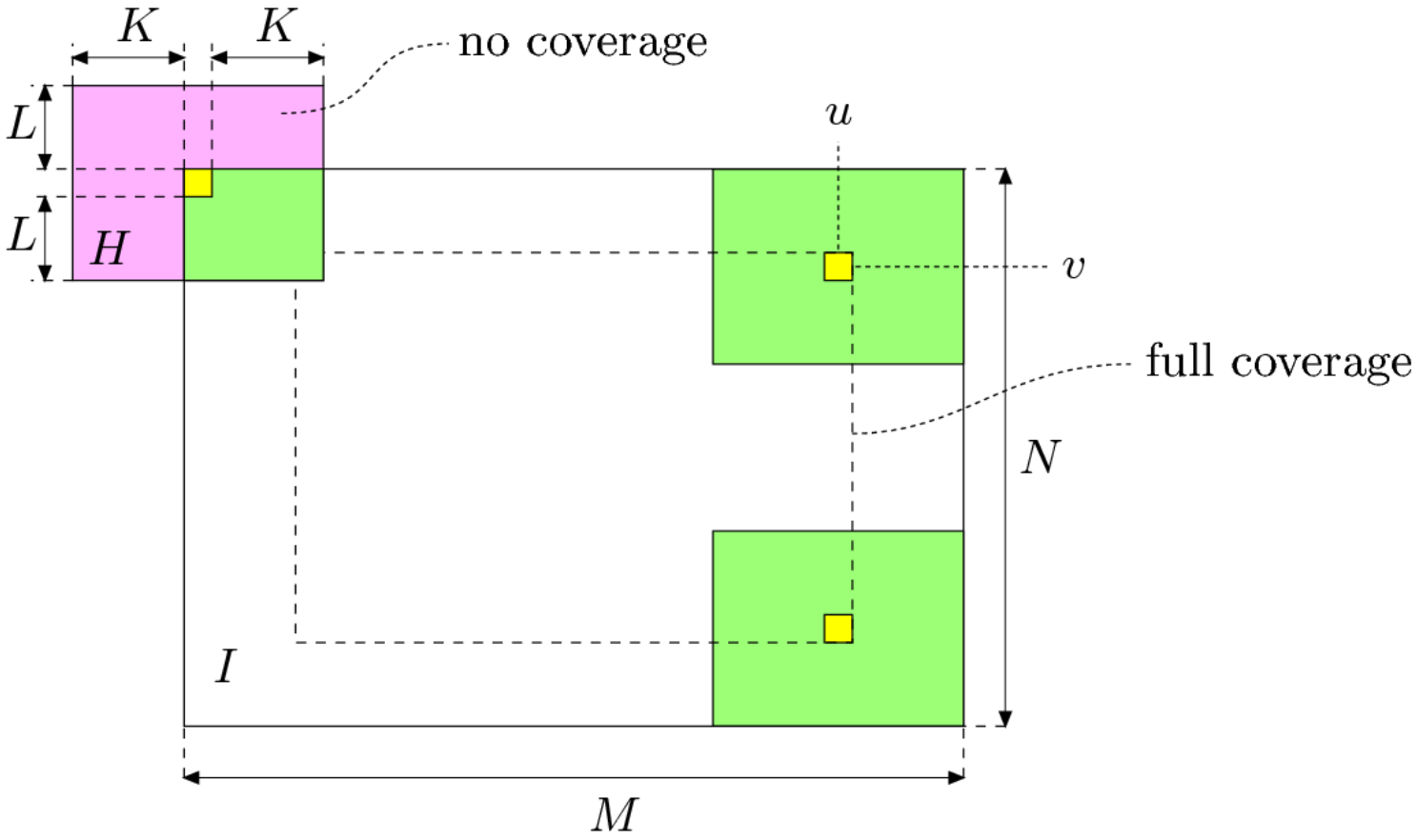
[Burger & Burge]

Linear Filter



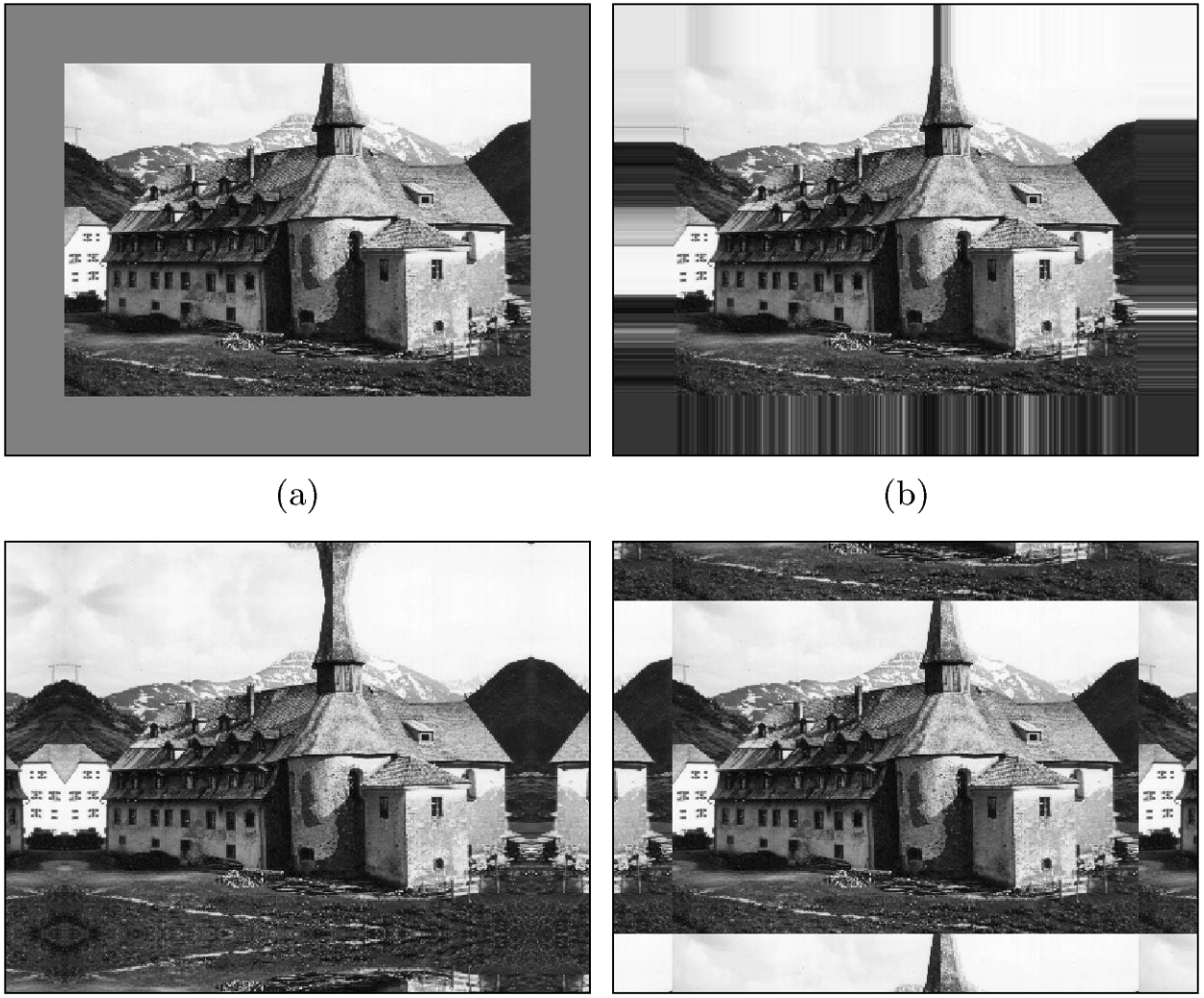
[Burger & Burge]

How to handle image borders?



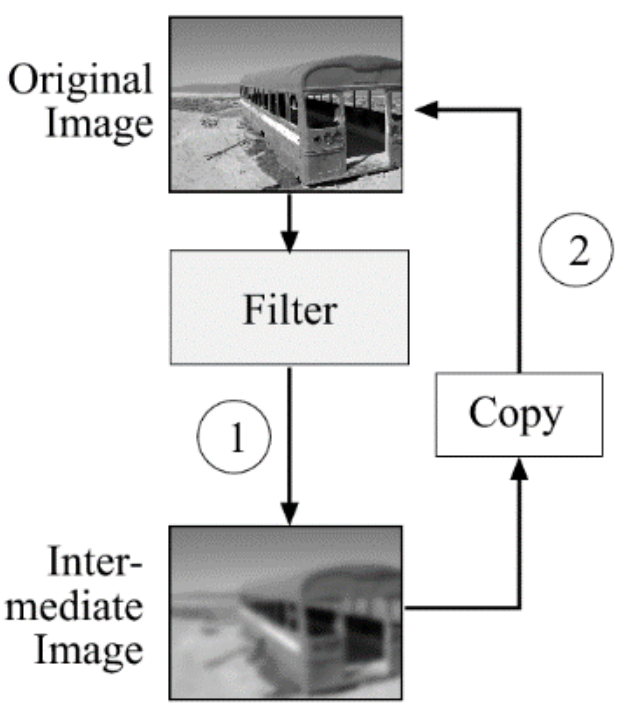
[Burger & Burge]

How to handle image borders?

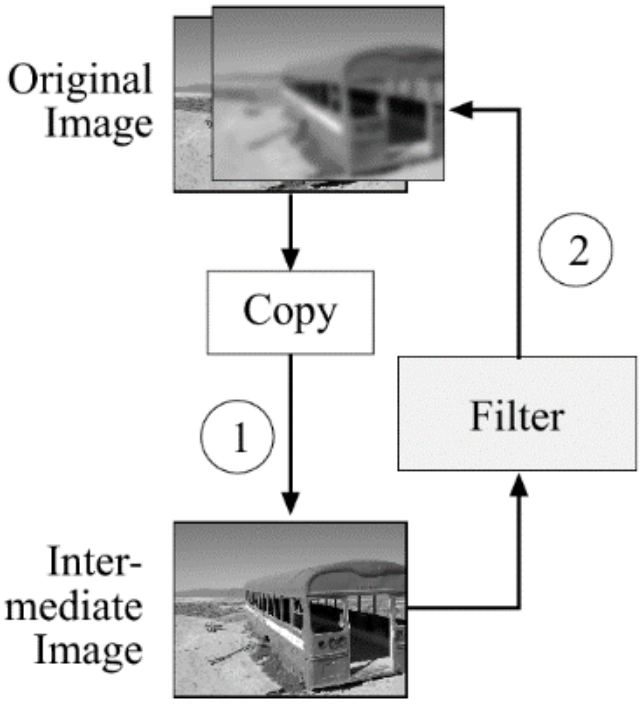


[Burger & Burge]

How to implement?



Variante A



Variante B

[Burger & Burge]

Original image

Image with noise

Final image

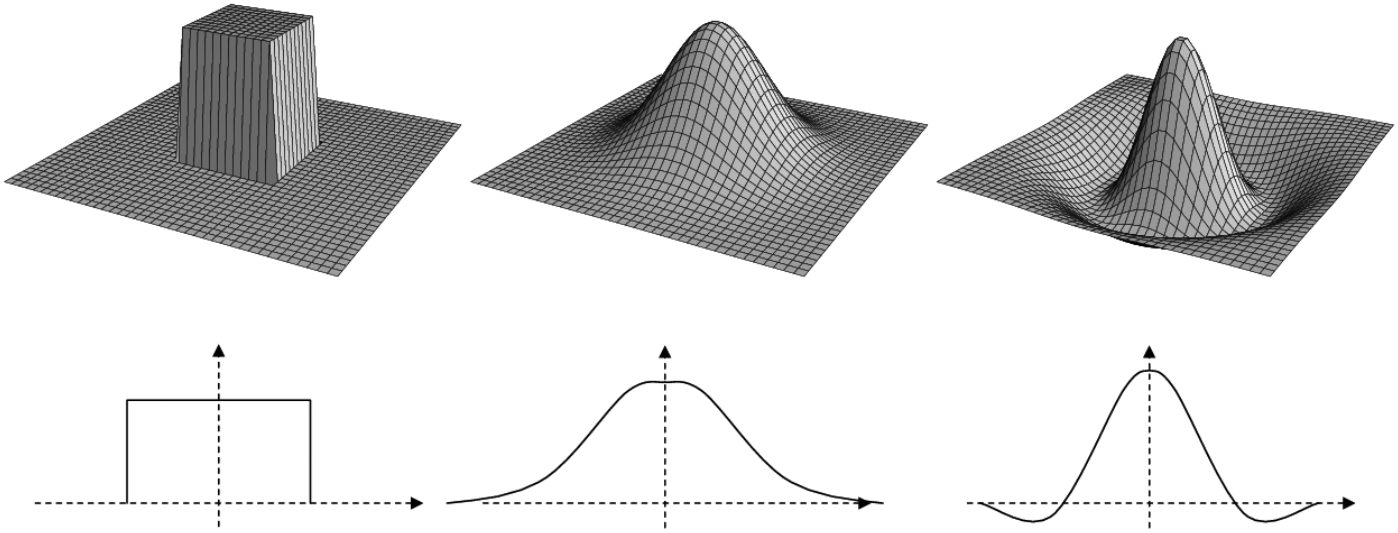
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Smoothing with an averaging filter

- The averaging filter is a simple approximation to the low-pass filter in the frequency-domain, where, the larger the filter area the lower its cutoff frequency

- There are two main differences:
  - It is empirical, it is not a precise equivalent to the low-pass filter
  - Smoothing is carried out as an operation in the space-domain, there is no need for a conversion to the frequency-domain, followed by filtering and a final conversion to the space-domain

Linear Filters – Examples



0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

(a)

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

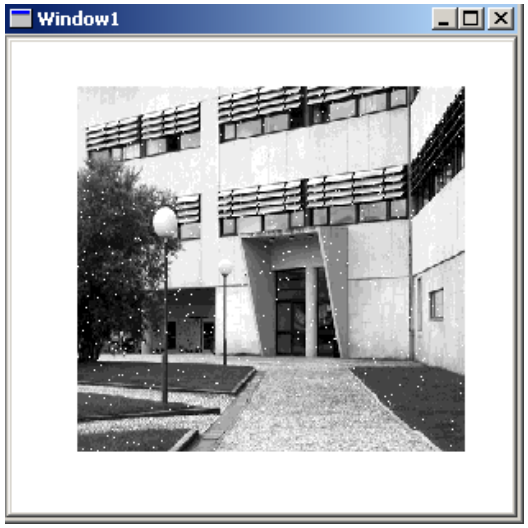
(b)

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(c)

[Burger & Burge]

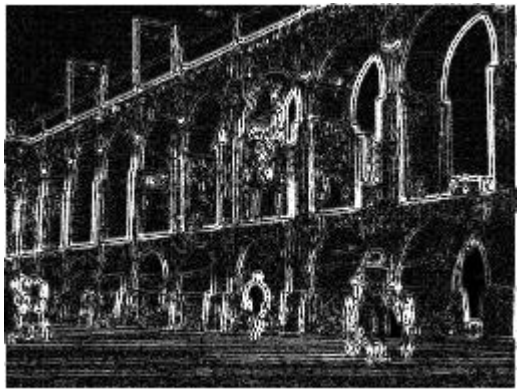
Examples – *Blurring* in OpenCV



8. Sharpening / detail enhancement

-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)

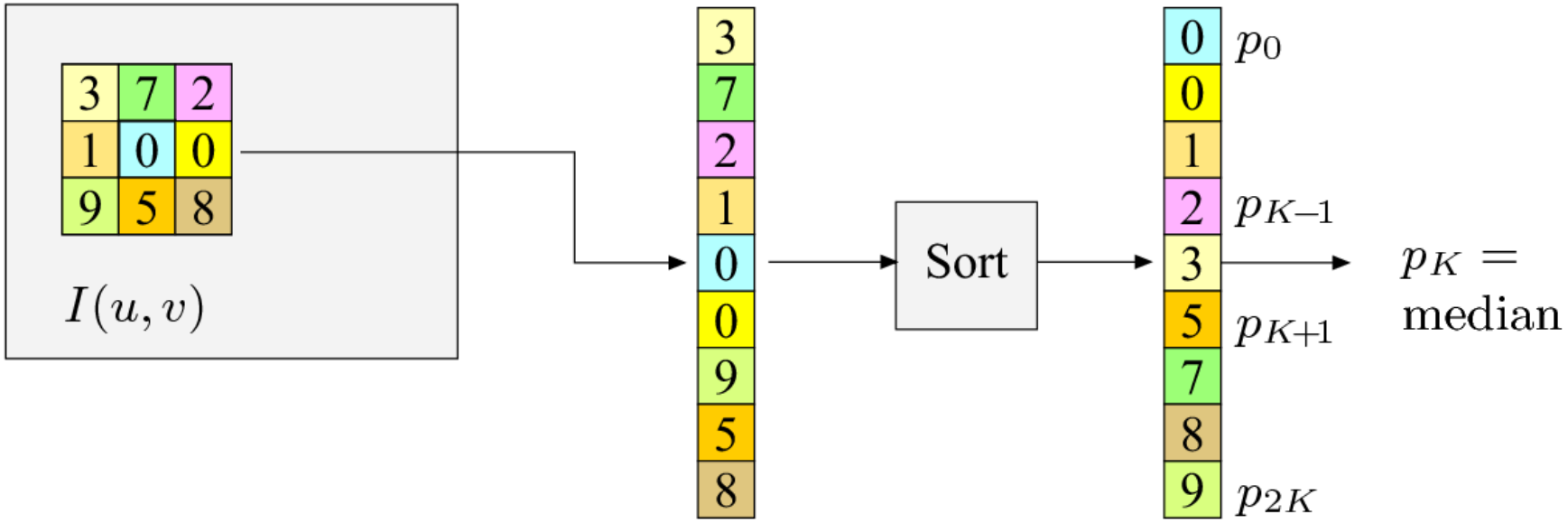


Minimum Filter vs Maximum Filter

(a) (b) (c)

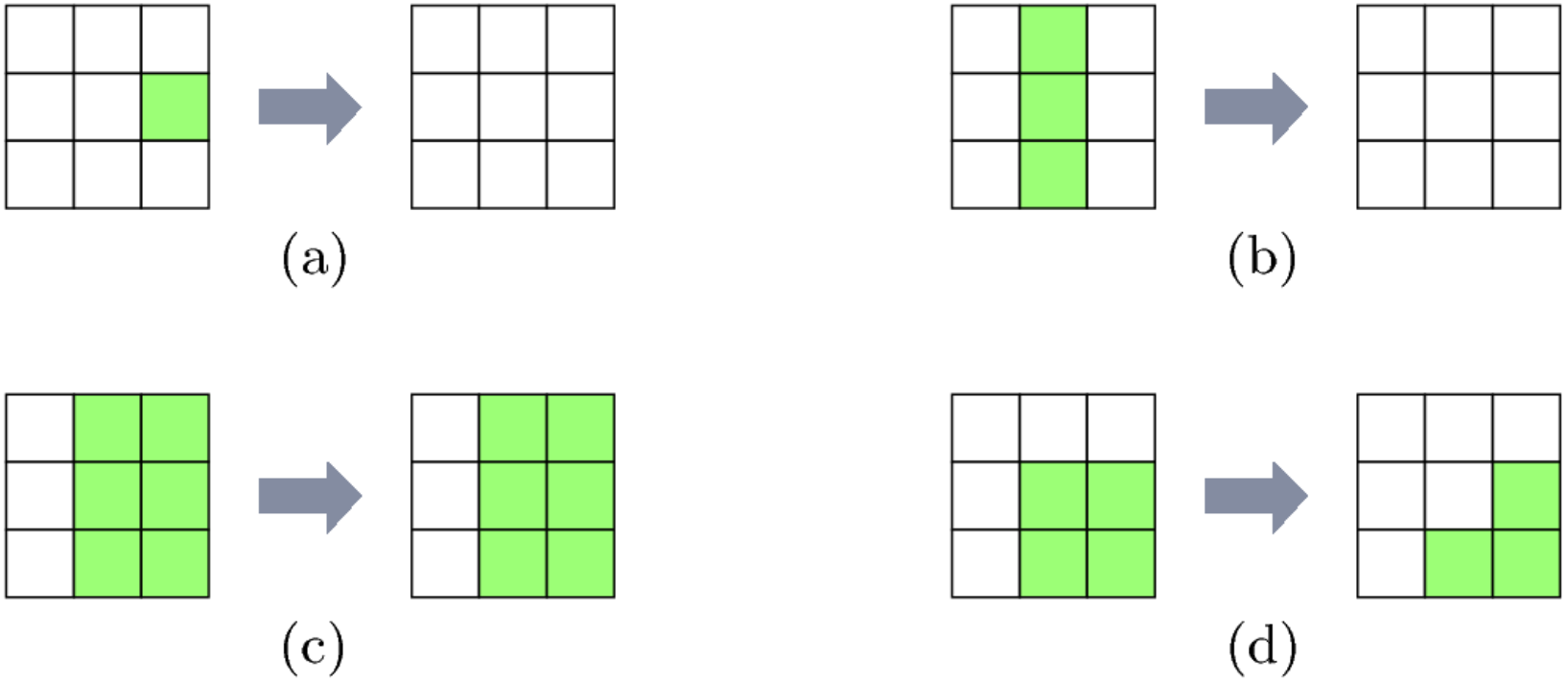
[Burger & Burge]

Median Filter



[Burger & Burge]

Median Filter



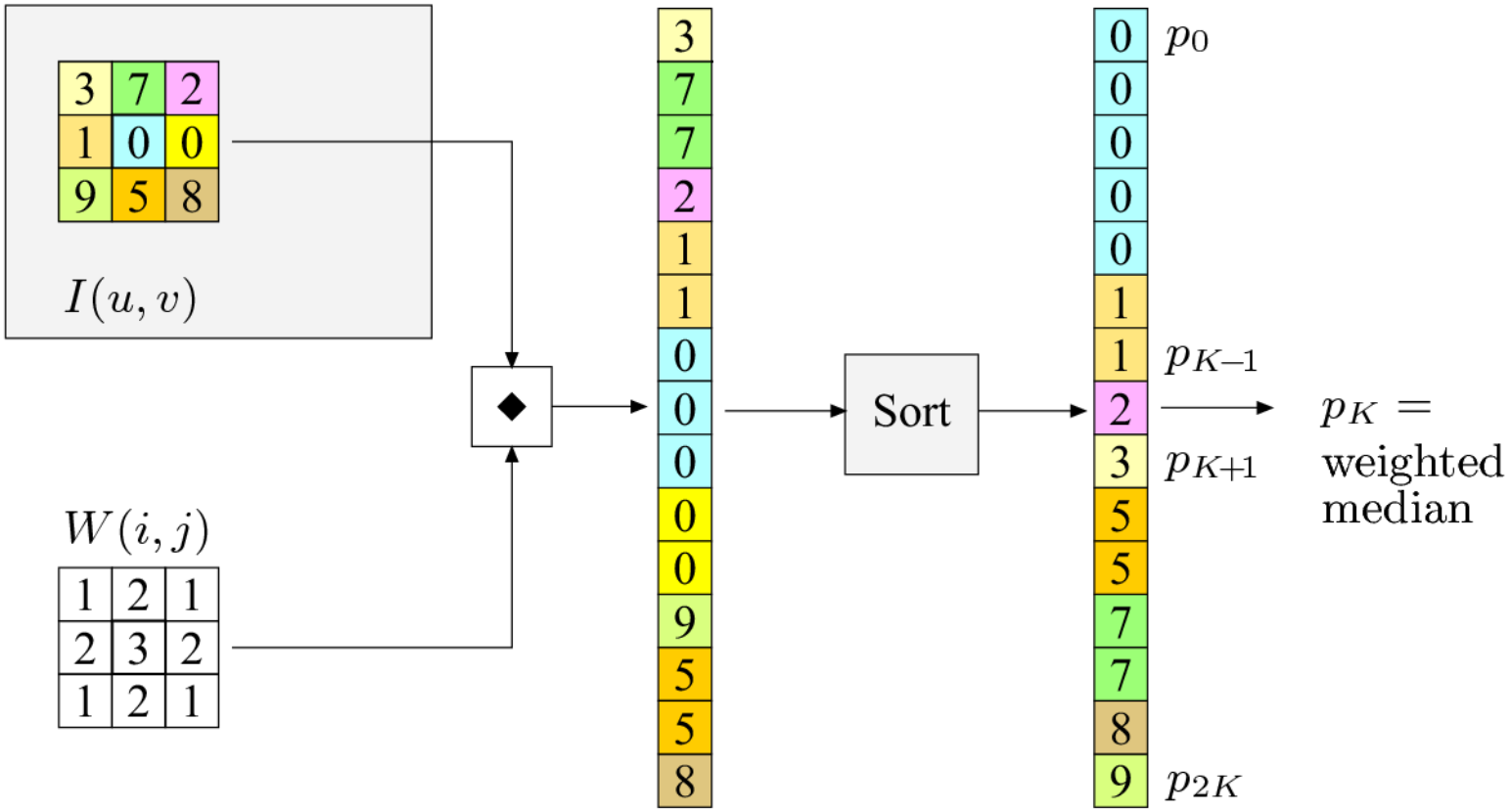
[Burger & Burge]

Averaging Filter vs Median Filter

(a) (b) (c)

[Burger & Burge]

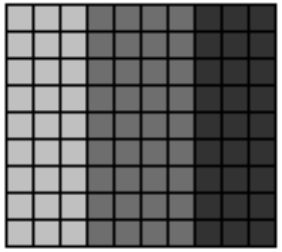
Weighted Median Filter



[Burger & Burge]

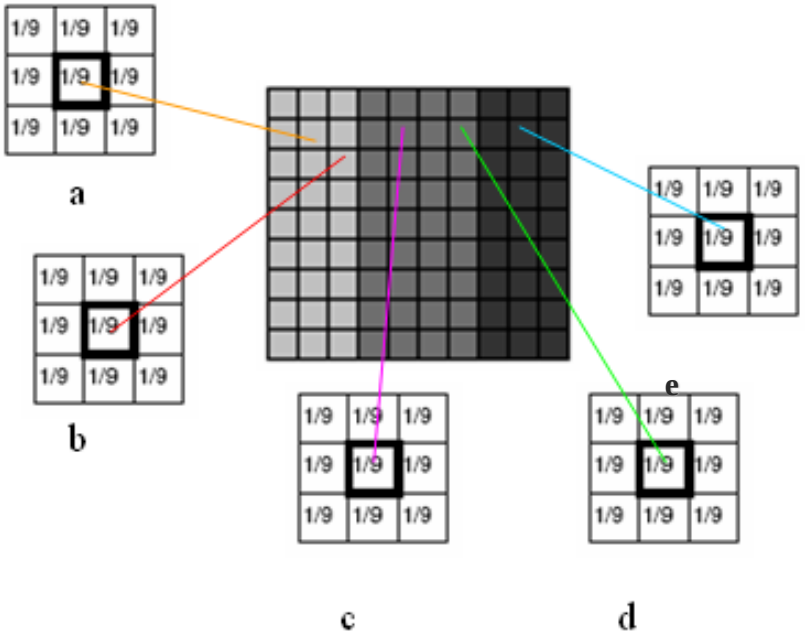
APPLICATION EXAMPLES

Application Example



Consider the 10 x 10 *pixels* image in the figure, with gray-levels 200, 100 e 50, in the range 0 to 255 (2^8 = 256 possible gray values).

1- Compute the pixel values in the resulting image when applying an **averaging filter** with a 3 x 3 kernel to the pixels referenced in the figure.

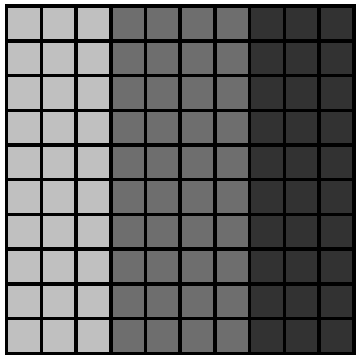


Applying the averaging filter to one row of the input image

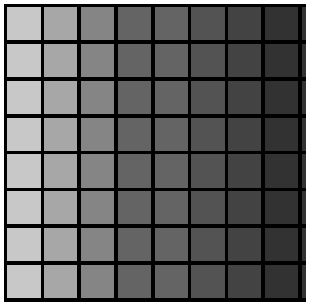
200	200	200	100	100	100	100	50	50	50
	200	166,7	133,3	100	100	83,3	66,7	50	

(10) Row of the input image

(8) Row of the output image



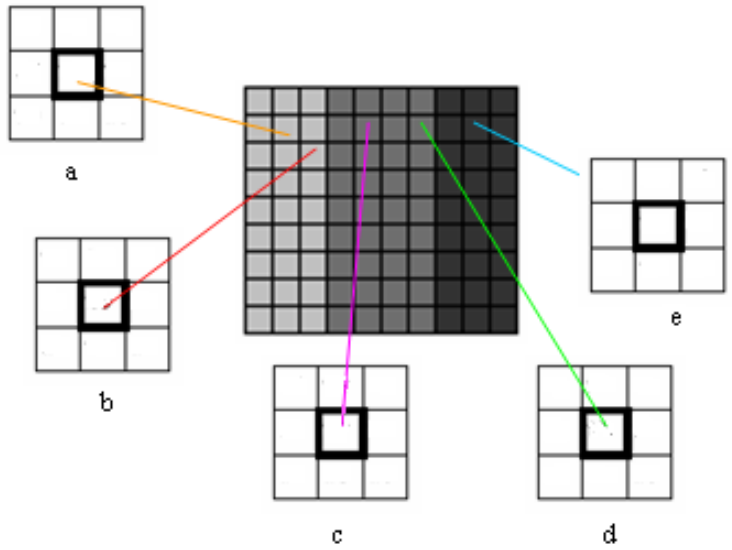
Original image



Resulting image

Extra: Apply a 5 x 5 kernel

2- Compute the pixel values in the resulting image when applying a **median filter** with a 3 x 3 kernel to the pixels referenced in the figure.



a- The *kernel* encompasses an area of constant intensity = 200:

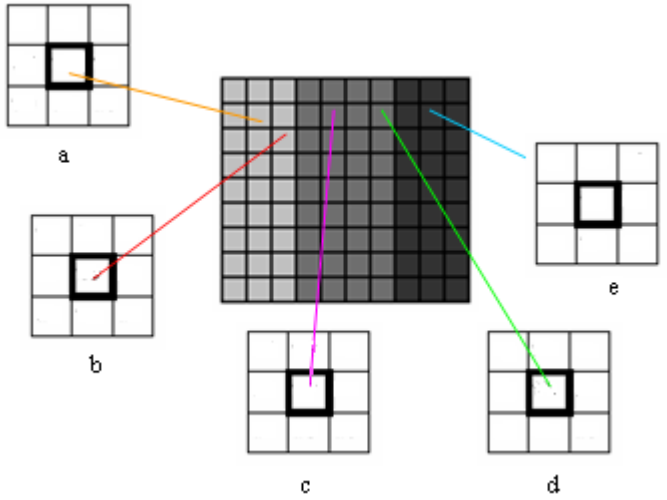
$I_o = 200$

b- The *kernel* encompasses 6 pixels with value 200 and 3 pixels with value 100:

100 100 100 200 200 200 200 200 200

$I_o = 200$ Median value

2- (cont.)

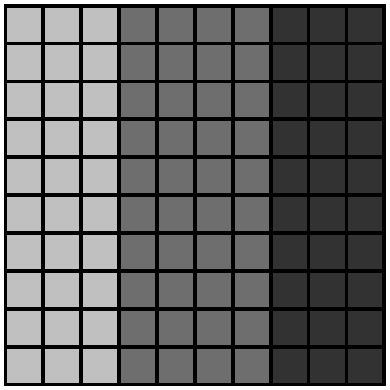


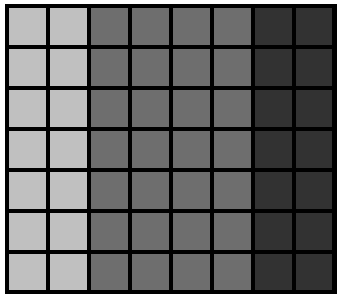
c- The *kernel* encompasses an area of constant intensity = 100:

$$I_o = 100$$

d- …

e- …



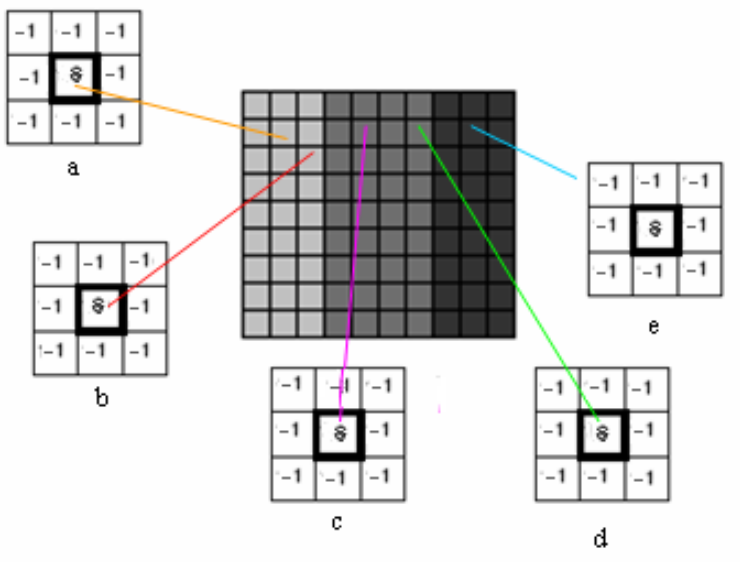


Original image

Resulting image

Extra: Apply a 5 x 5 kernel

3- Compute the pixel values in the resulting image when applying a **sharpening filter** with a 3 x 3 kernel to the pixels referenced in the figure



a- The *kernel* encompasses an area of constant intensity = 200:

$$I_o = 0$$

b- The *kernel* encompasses an area with 3 pixels of value = 100 and 6 pixels of value = 200, one of wich is the *kernel* center.

$$Io = 5 \times 200 \times (-1) + 8 \times 200 + 3 \times 100 \times (-1) = 1600 - 1300 = 300$$

c- …

d- The *kernel* encompasses an area with 3 pixels of value = 50 and 6 pixels of value = 100, one of wich is the *kernel* center.

$$Io = 5 \times 100 \times (-1) + 8 \times 100 + 3 \times 50 \times (-1) = 800 - 650 = 150$$

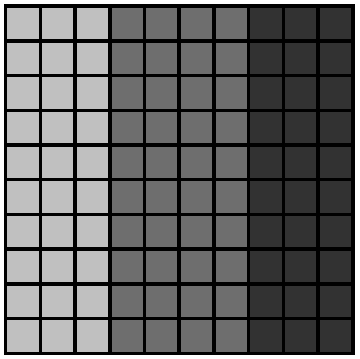
e- …

Applying the sharpening filter to one row of the input image

200	200	200	100	100	100	100	50	50	50
	0	300	?	0	0	150	?		0

(10) Row of the input image

(8) Row of the output image



Original image

?

Resulting image

Extra: Apply a 5 x 5 kernel

PROCESSING COLOR IMAGES

9. Processing color images

- The first Image Processing methods were developed for grayscale images
- Currently, color images are common and their processing has become and important issue

- Some of the previous operations can be generalized for color images
- Color images are usually defined using the 3 RGB components
- But can be represented using other color models (e.g., HSI)

Imagem Original

Componente R

Componente G

Componente B

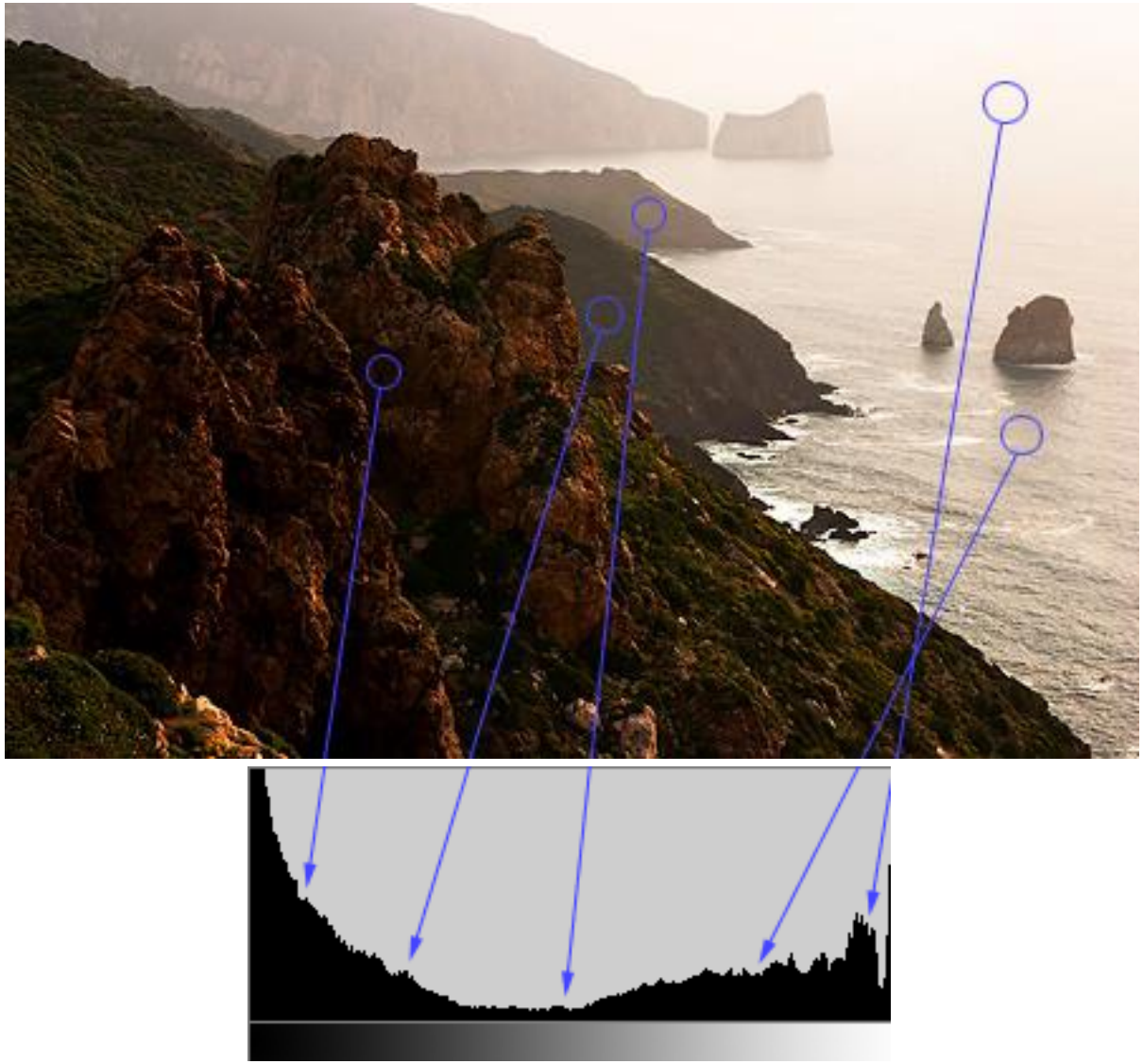
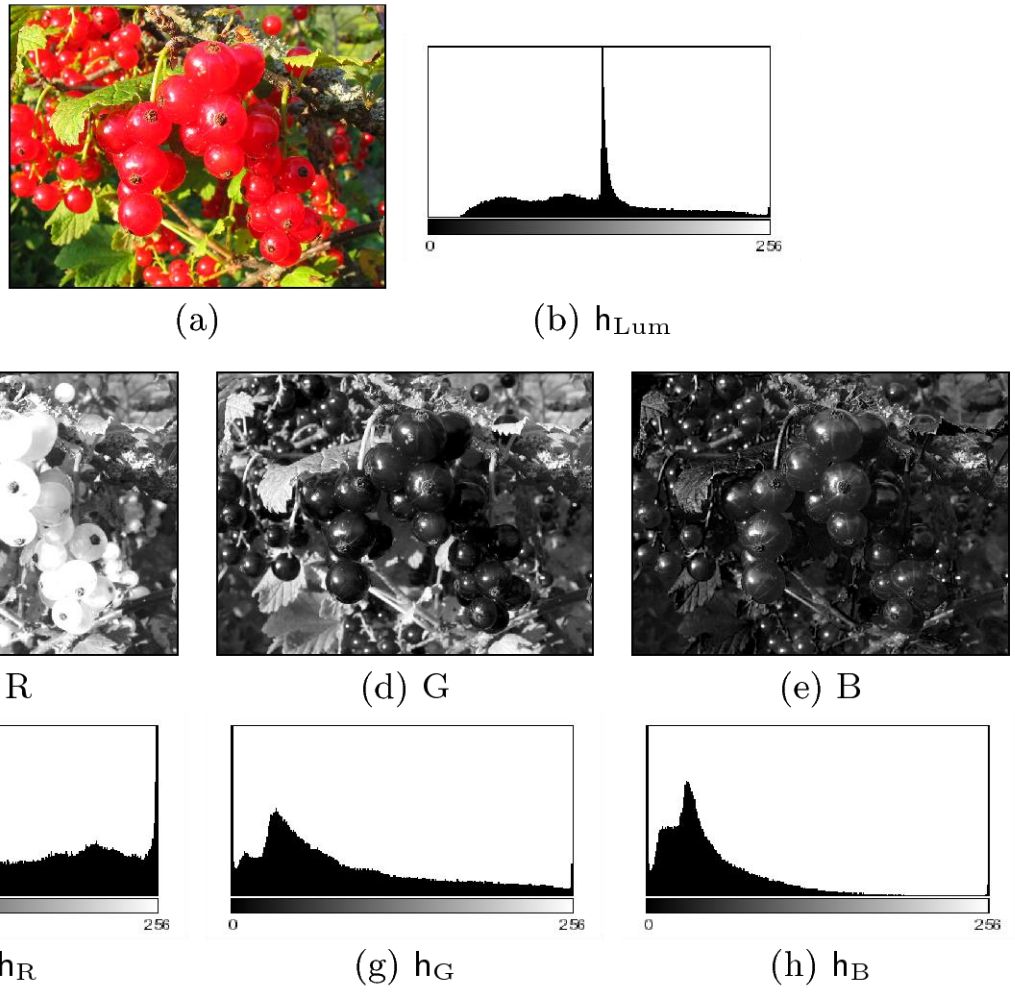


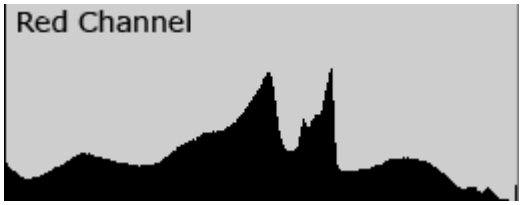
Image and Luminosity Histogram

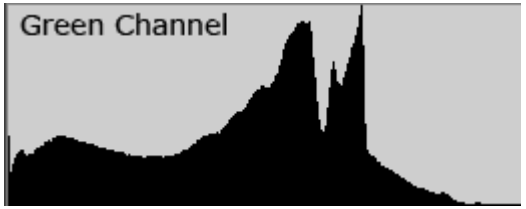
Histograms for a color image

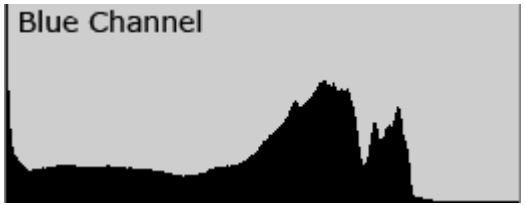


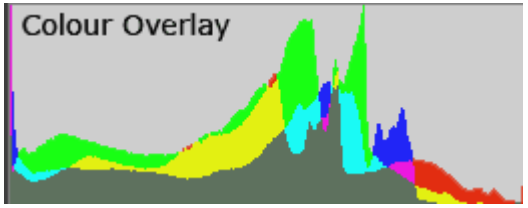
[Burger & Burge]

Histograms for a color image

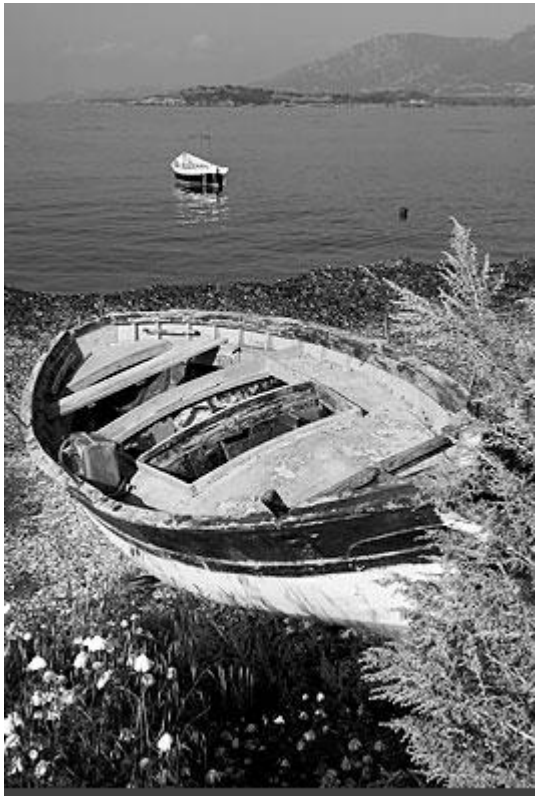


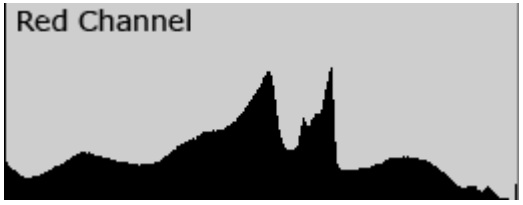


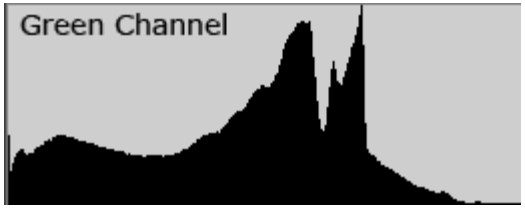


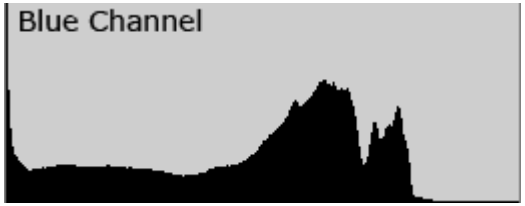


Luminosity









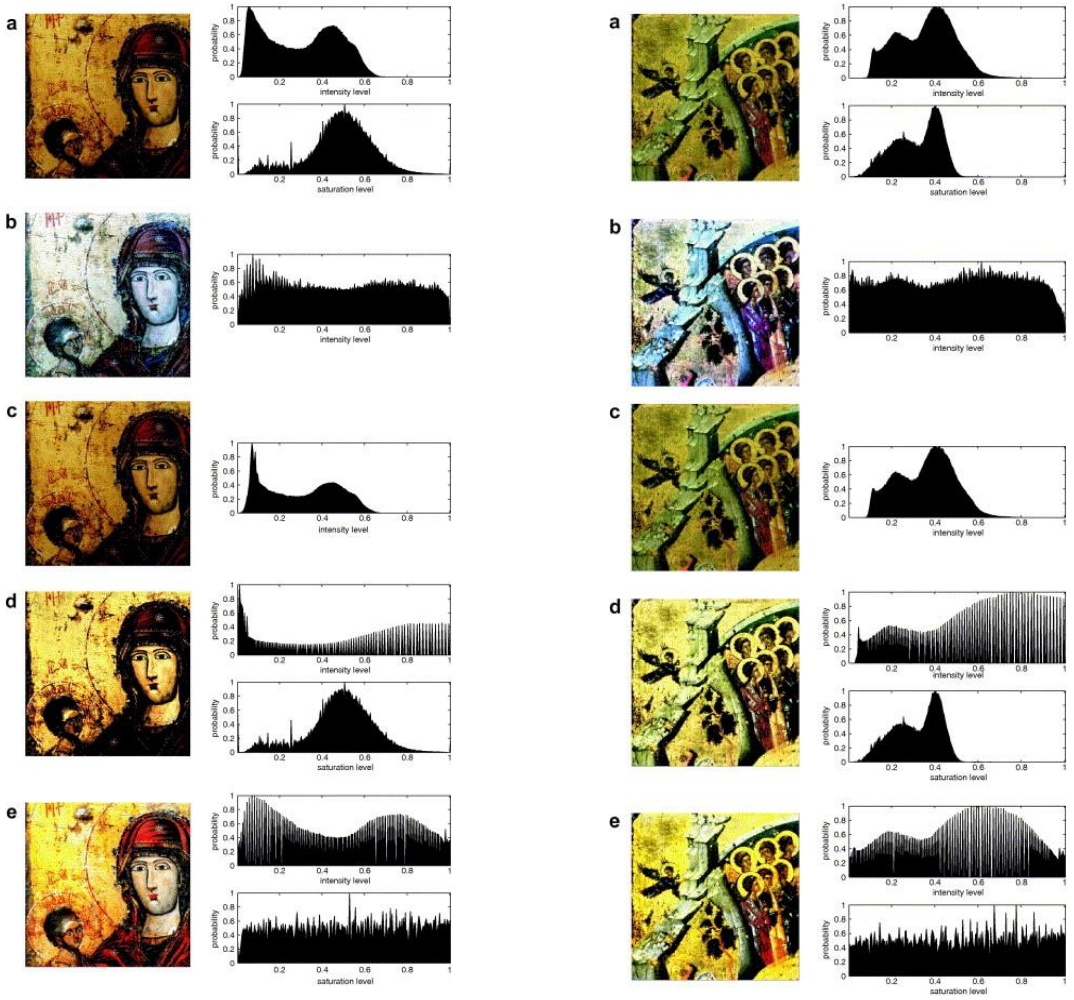
http://www.cambridgeincolour.com/tutorials/histograms2.htm

Color Images – Methods for histogram equalization

Original images

RGB-based methods

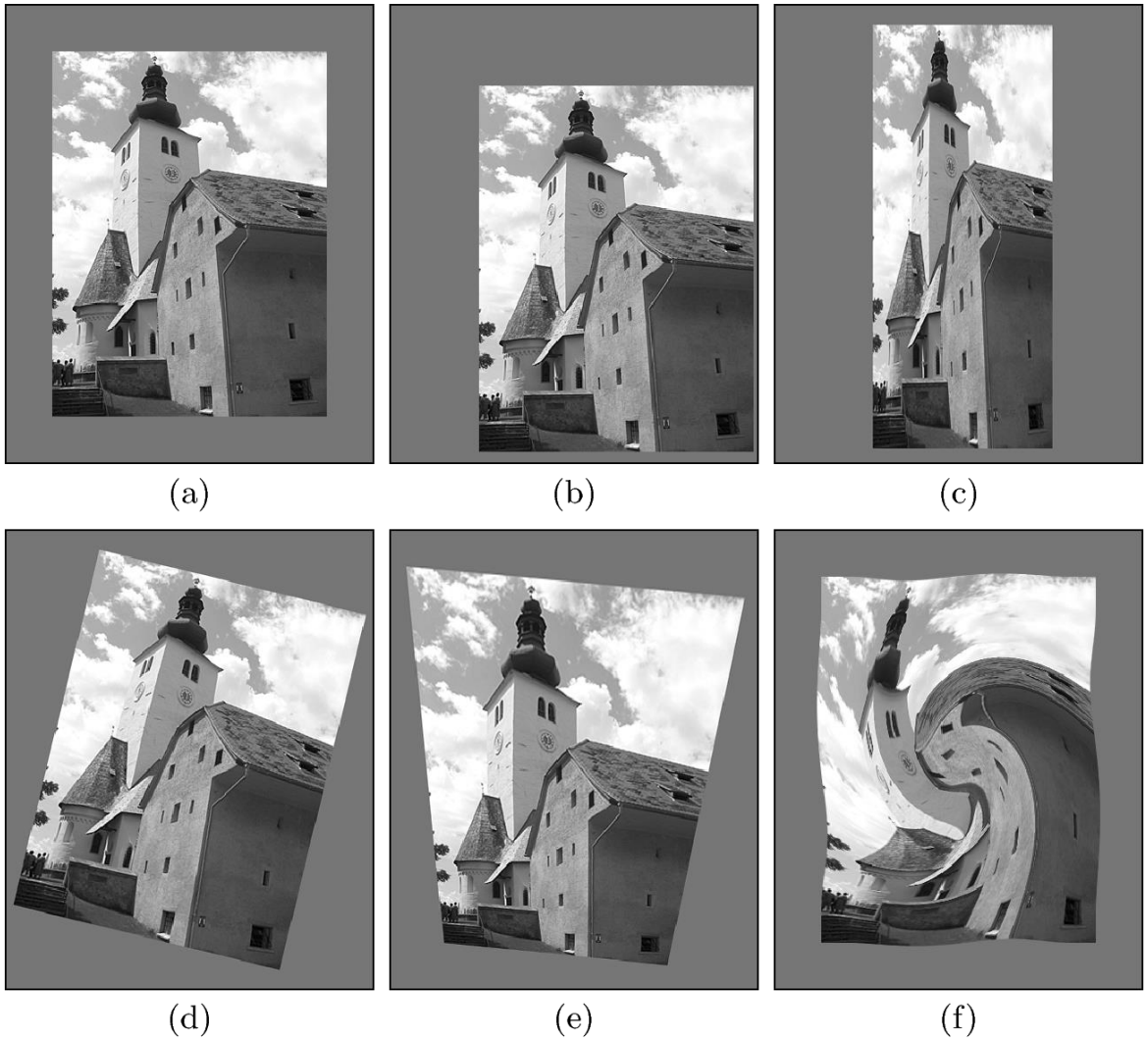
HSI-based methods



N. Bassiou and C.Kotropoulos, “Color image histogram equalization by absolute discounting back-off” *Computer Vision and Image Understanding*, Vol 107, July August 2007, pp. 108-122

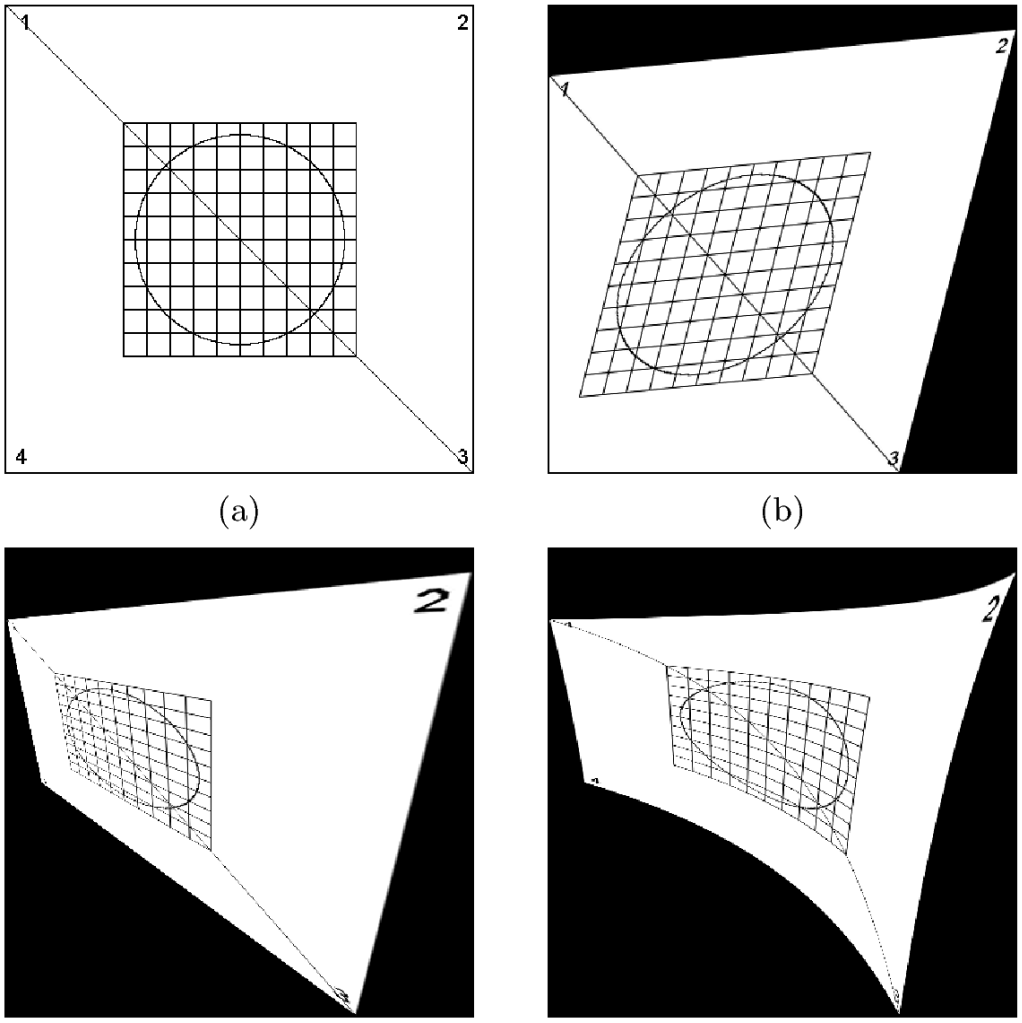
GEOMETRIC TRANSFORMATIONS

Global geometric transformations – What are they?



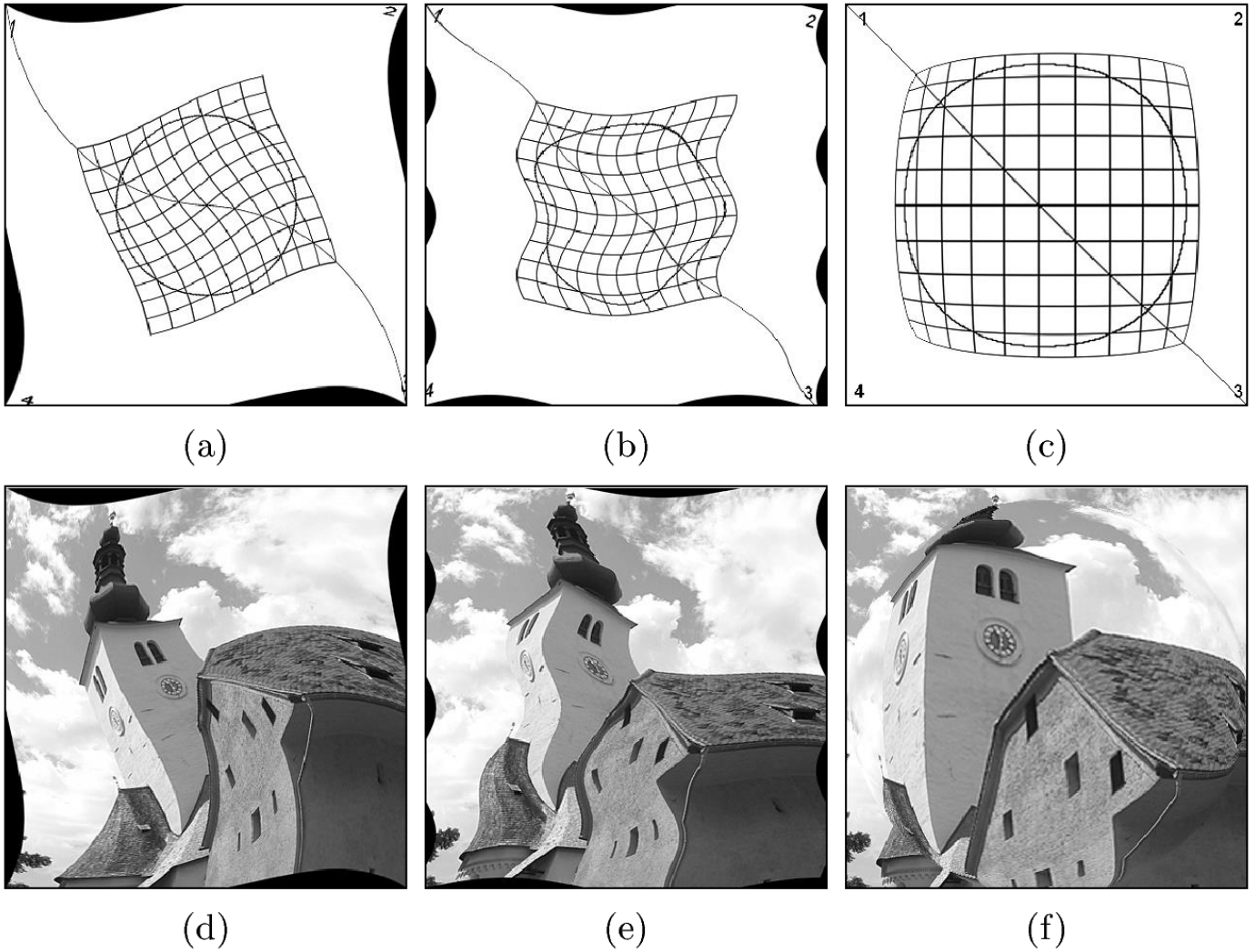
[Burger & Burge]

Geometrical transformations: affine, projective, bilinear



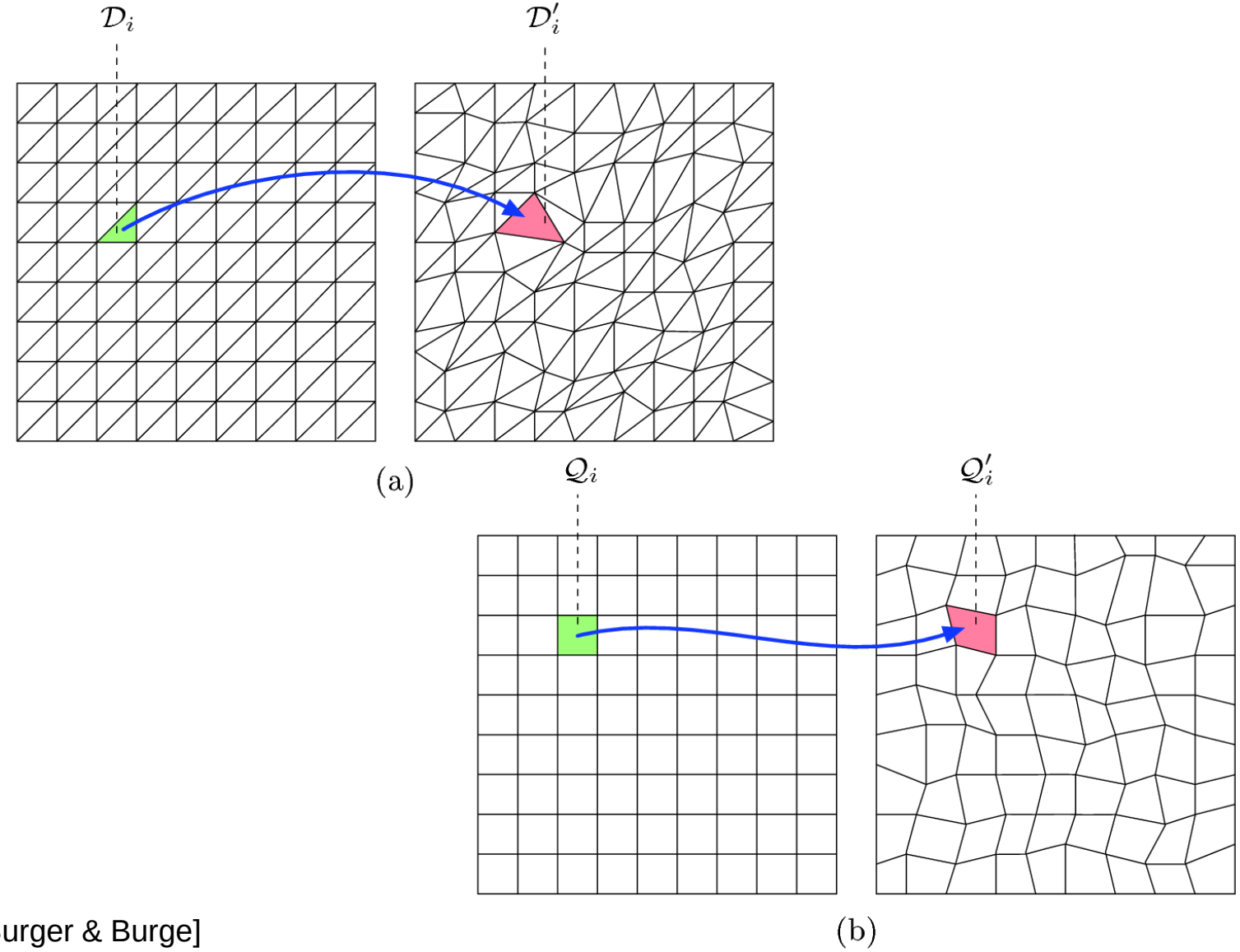
[Burger & Burge]

Non-linear transformations: *Twirl*, *Ripple*, *Sphere*



[Burger & Burge]

Local geometric transformations: *Meshing*



[Burger & Burge]

Geometrical transformations – *Image Warping*

- Usual application:
  - Distorsion correction (e.g., after image acquisition)
  - *Texture mapping* (Computer Graphics)
  - *Morphing* for special effects (e.g., movie industry)

- Two possible approaches to *warping*:
  - *Forward mapping:*

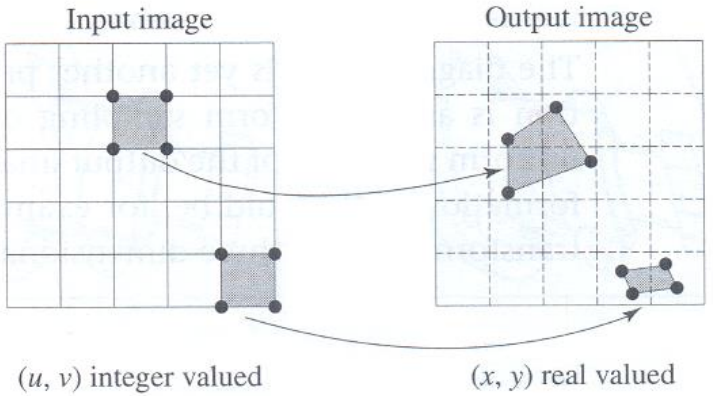
    $(x,y) = (X(u,v), Y(u,v))$

  - *Backward mapping:*

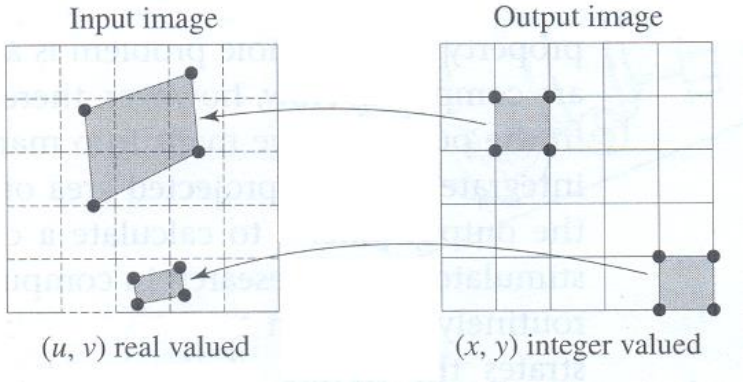
    $(u,v) = (U(x,y), V(x,y))$

- Relative advantages and disadvantages

Forward mapping:

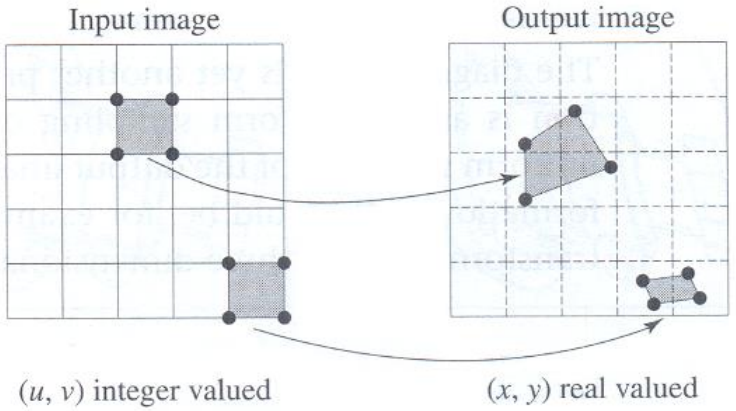


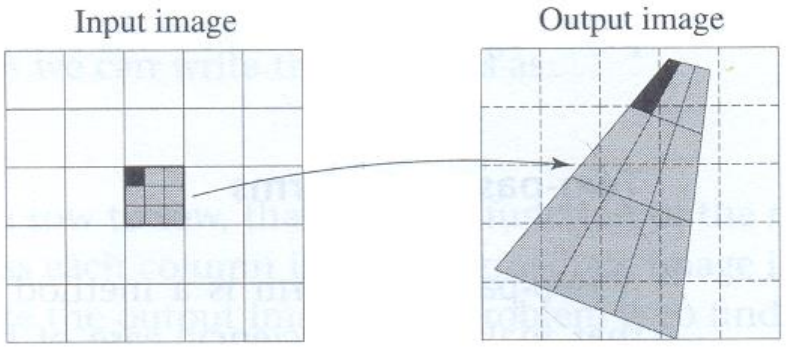
Inverse mapping:



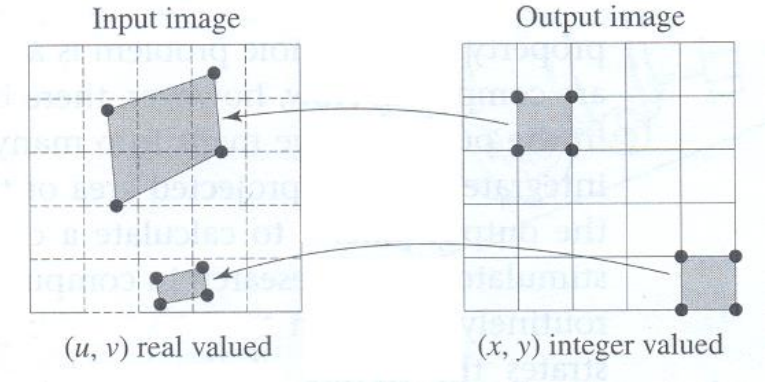
Forward mapping:

- Gaps and superpositions can happen in the output image
- Computing the intensity of the output image pixels is not straighforward
- Magnification (input pixel mapped on more than one output pixel) can induce a “blocky” appearance
  - Can be solved with additional sampling (input pixel subdivision)

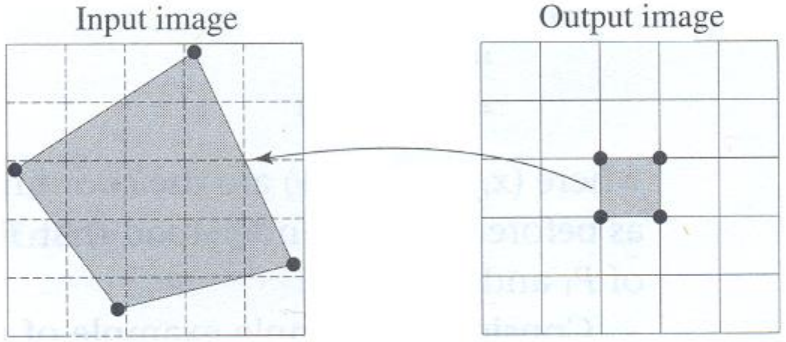




Inverse mapping



- All output image pixels are computed
- Gaps no longer occur
- If an output pixel is mapped on several input pixels, it is necessary to integrate the input pixels values to obtain the value of the output pixel
- Example: animating a chessboard textured model



Local warps

a. Unwarped image

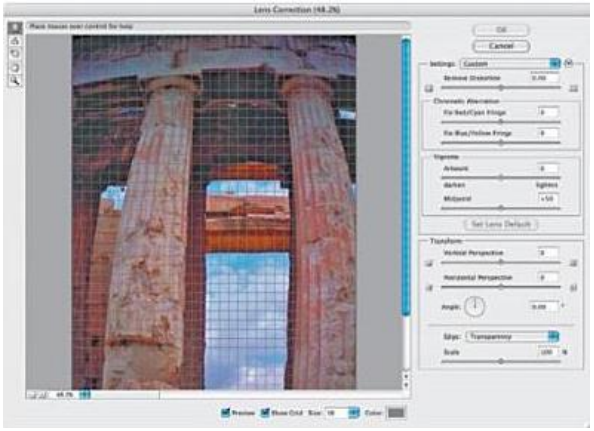
b. Translation

c. Scaling

d. Rotation

A. Gustafsson, Interactive Image Warping, Helsinky, 1993

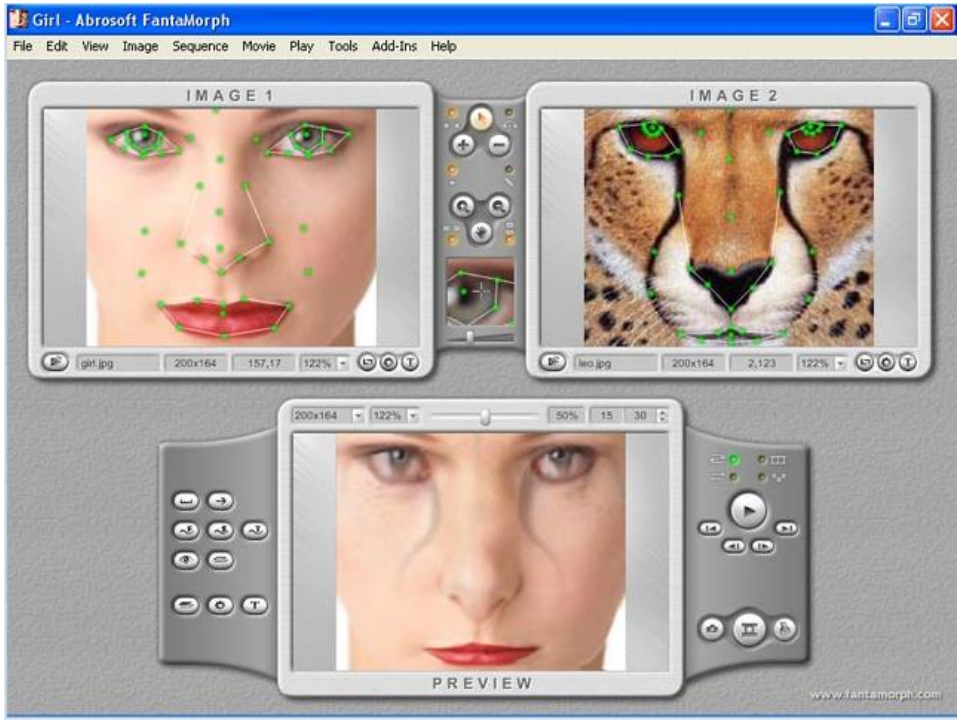
Correcting image distortion

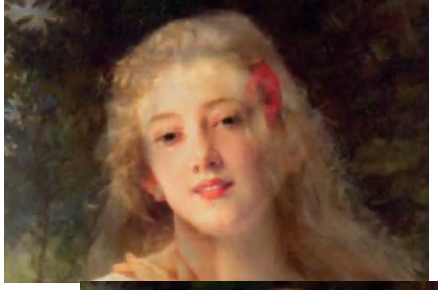


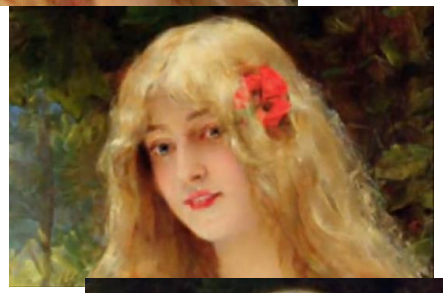
Texture mapping

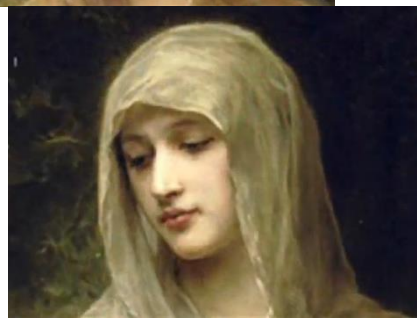
http://www.photoshopsupport.com/tutorials/tt-cs2/image-warp.html

Morphing





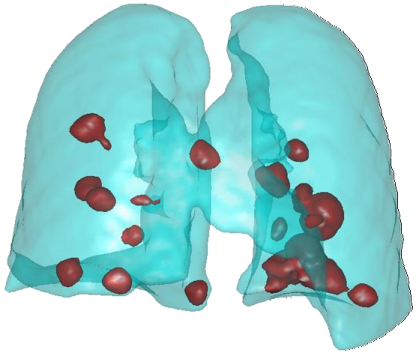




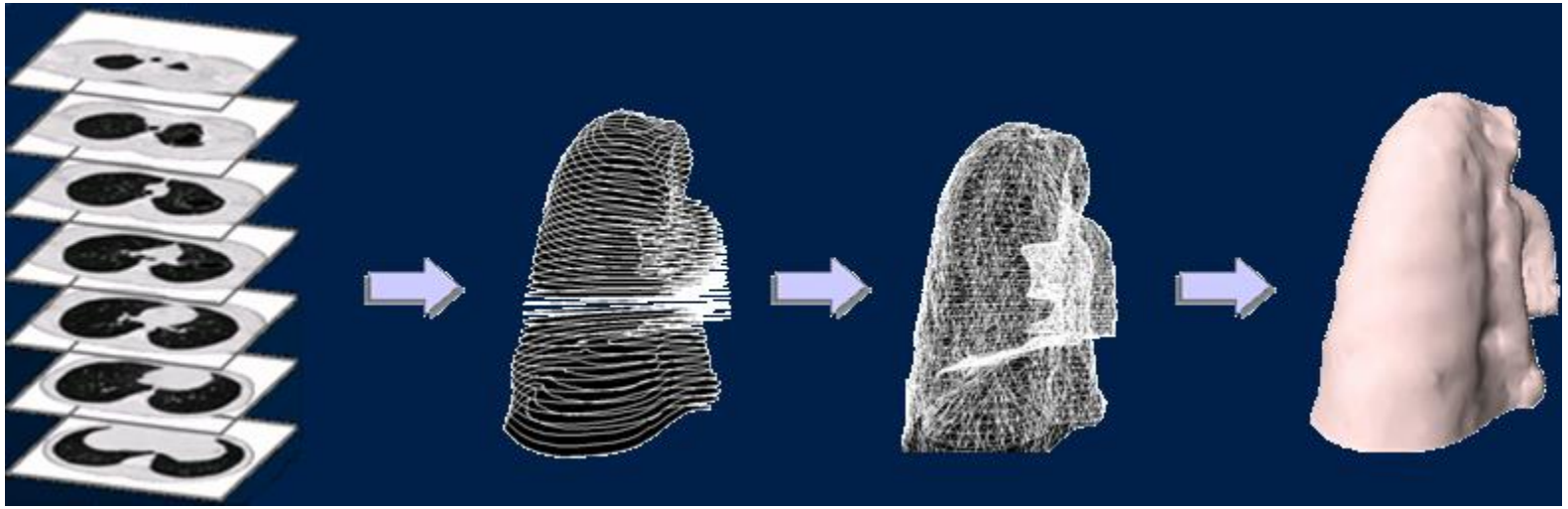
http://www.youtube.com/watch?v=nUDIoN-_Hxs

http://www.fantamorph.com/?gclid=CL245faKjp4CFZQA4wodghq9qg

Application Examples in Medical Imaging



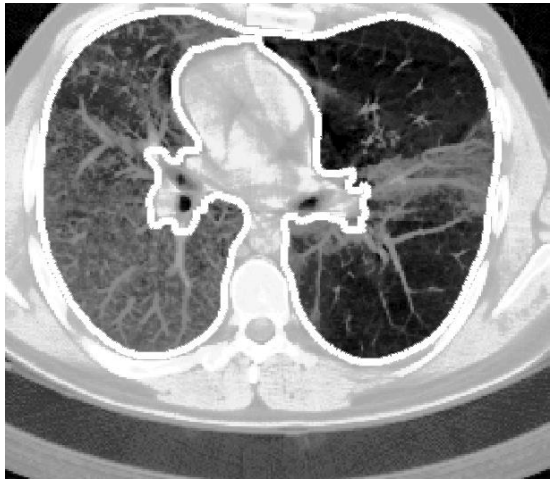
Processing chest CT images and lung segmentation

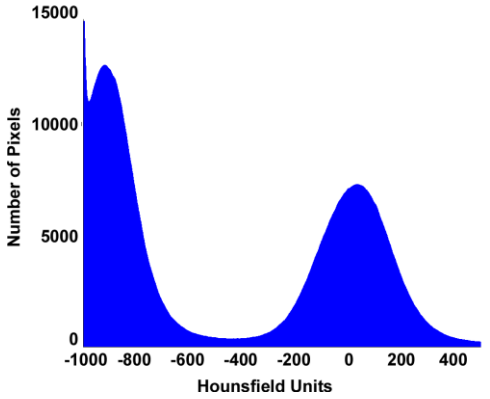


- Set of CT sections (images) define a volume
- Reduce noise and enhance images
- Detect lung contours in each image
- Associate contours to define the lung surfaces

Processing chest CT images and segmentation

- Detecting contours of the lungs and of lung air bubbles on chest CT images
- Different gray levels correspond to different tissues and air in each CT image
- Determine the histogram of each image and the intensity values corresponding to the transition between the lungs and other tissues
- Identify the lung contour pixels in each image
- "Enhance" the contours

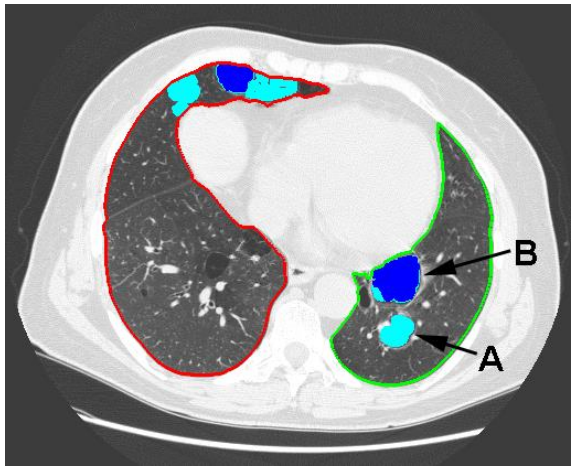


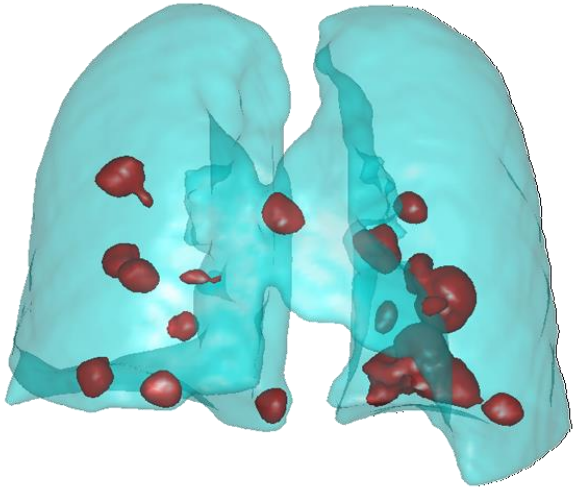


CT image histogram

Processing chest CT images and segmentation

- Detecting contours of the lungs and of lung air bubbles on chest CT images
- Identify the contour pixels in each image
- “Enhance” the contours
- Volumetric visualization of lungs and air bubbles





REFERENCES

References

- W. Burger, M. J. Burge, Principles of Digital Image Processing, Vol.1 and Vol. 2, Springer, 2009

- A. Watt, F. Policarpo, The Computer Image, Addison Wesley, 1998

- R. Gonzalez, R Woods, Digital Image Processing, 2nd Ed., Addison Wesley, 2002

Acknowledgements

- To Prof. Paulo Dias and PhD students