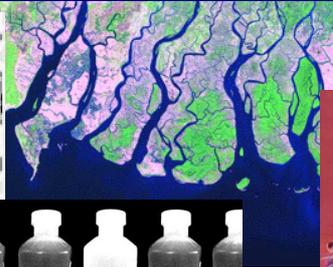
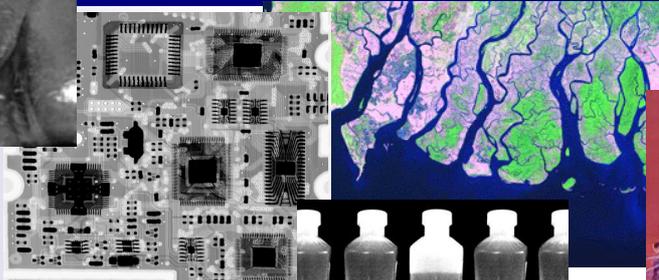




Digital Image Processing



INSA GE – option TdSI – Master GE/GP

Thomas Grenier

Olivier Bernard



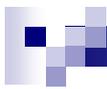
Digital Image Processing

Module overview

*Département Génie Electrique
5GE - TdSi*

Summary

- I. Introduction
 - DIP ?, Examples, Fundamental steps, components
- II. Digital Image Fundamentals
 - Visual perception, Light
 - Image sensing, acquisition, sampling, quantization
 - Linear, and non linear operation
- III. Discrete 2D Processing
 - Vector space, Convolution
 - Unitary Transform
- IV. Image Improvement
 - Enhancement, restoration, geometrical modification
- V. Image Analysis (Oliver Bernard) (6h)
 - Edge detection, Segmentation, shape analysis



Module organization

- Courses

- 16h

- TD

- 2x2h image

- + 2x2h related exercises

- TP

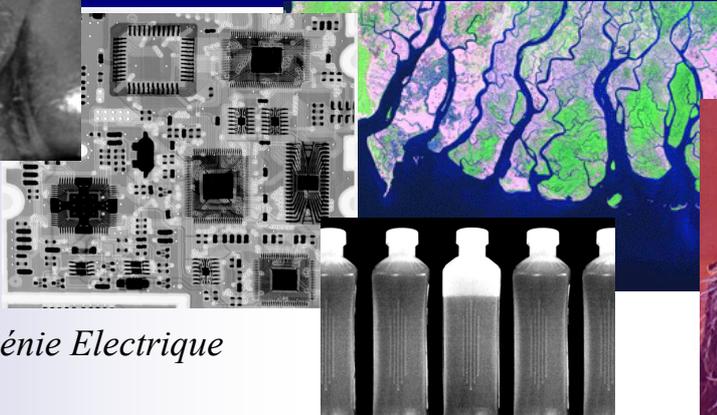
- 2x4h

- Project (*mini-projet*)

- matlab

Exam 2h-3h

Digital Image Processing



Introduction



Département Génie Electrique
5GE - TdSi

Summary

I. Introduction

- DIP ?, Examples, Fundamental steps, components

II. Digital Image Fundamentals

- Visual perception, light
- Image sensing, acquisition, sampling, quantization
- Linear, and non linear operation

III. Discrete 2D Processing

- Vector space, Convolution
- Unitary Transform

IV. Image Improvement

- Enhancement, restoration, geometrical modifications

Introduction

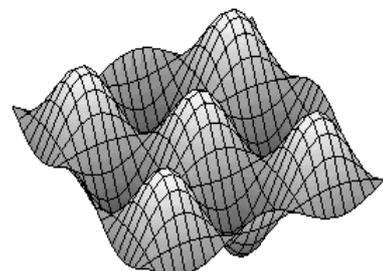
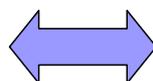
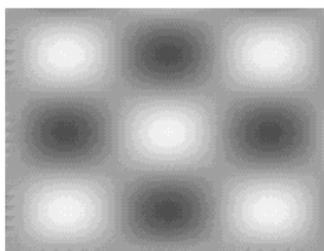
- What is Digital Image Processing?
- Examples of fields that use DIP
- Fundamental steps in DIP
- Components of an image processing system

→ Book

Digital Image Processing, Gonzales, Prentice Hall (3Ed.)

What is a *DIP* ?

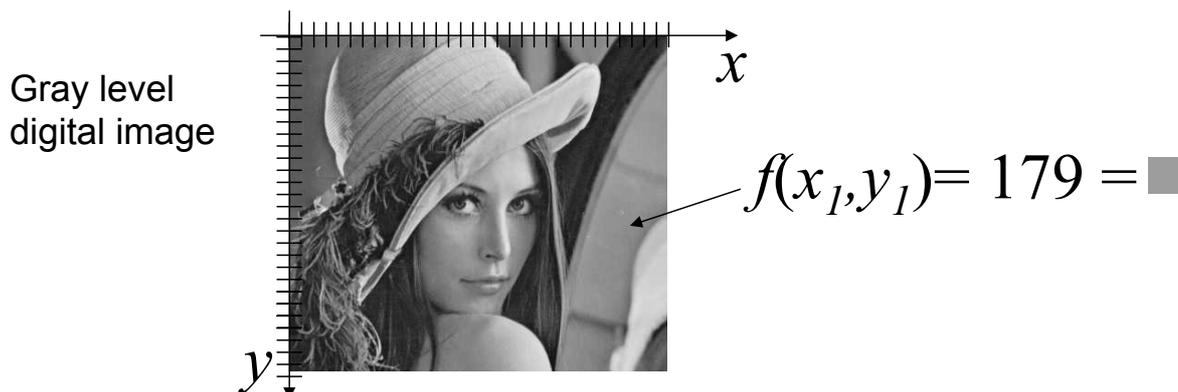
- Image definition
 - An image may be defined as a two-dimensional function, $f(x,y)$
 - x and y are spatial (plane) coordinates
 - the amplitude of f at any pair of coordinates (x,y) is called **intensity** or **gray level** of the image at that point



What is a *DIP* ?

■ Image definition

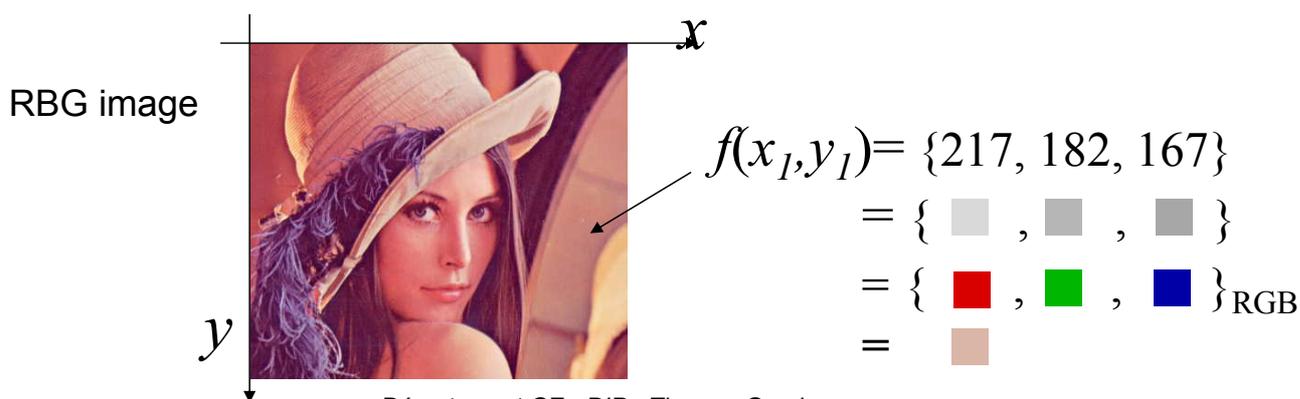
- When f , x and y are all finite and discrete quantities, the image is called a **digital image**



What is a *DIP* ?

■ Image definition

- The definition of f may be extended:
 - as a n -dimensional function,
 - i.e. 3D: $f(x, y, z)$ or image sequence $f(x, y, t)$
 - with amplitudes composed as a vector of data,
 - i.e. Color image: 3 components at each point, Complex number



What is a *DIP* ?

■ Pixel

- A digital image is composed of a finite number of elements, each of which has a particular location and value
- These elements are referred to as *picture elements*, *image elements*, *pels*, and **pixels**
- *Pixel* is the term most widely used to denote the elements of a digital image

What is a *DIP* ?

■ Digital Image Processing & related areas

- Image processing
 - Low-level processes
 - noise reducing, contrast enhancement, ...
- Image analysis
 - Mid-level processes
 - segmentation (partitioning an image into regions or objects)
 - classification (recognition) of objects, ...
- Computer vision
 - Ultimate goal: emulate human vision
 - High-level processes
 - learning, inferences making, actions taking
 - giving a sense to a set of recognized objects
 - perform the cognitive functions normally associated with vision
- no clear-cut boundaries...

What is a *DIP* ?

- Digital Image Processing and human vision
 - The field of DIP refers to processing digital images by means of a digital computer

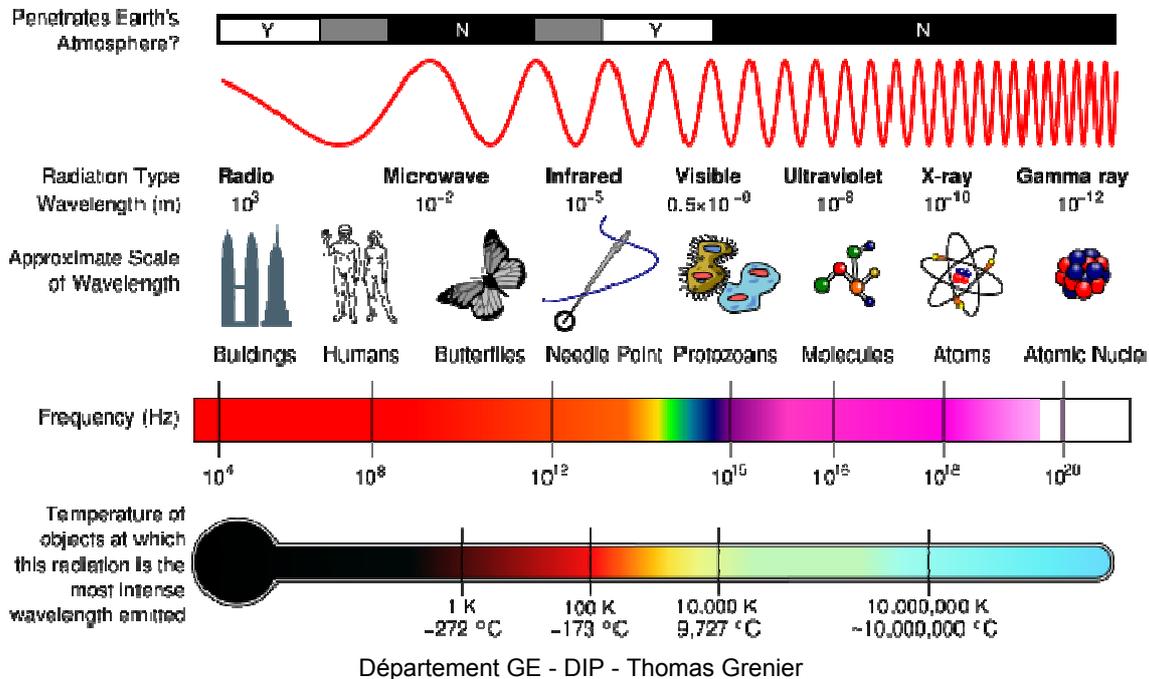
	humans	imaging machines & DIP +CV
electromagnetic spectrum of images	visible band	Full spectrum
sources of images	accustomed to be associated with image	all (Ultrasound, electron microscopy, ...)
processing by	<i>brain</i>	computer(s)
analysis	<i>hand (manually)</i>	computer(s)

Examples of fields that use DIP

- Many applications...
 - **Industrial inspection** (anomalies detection, measuring (bench), tracking, monitoring...)
 - Medical imaging (visualization, tumor detection, reconstruction, artifact correction, diseases quantification, ...)
 - **Satellite Imaging** (weather, environmental conditions monitoring,..)
 - microscopy (pharmaceutical, micro inspection, materials characterization,...)
 - Telecommunication (transmission, compression,...)
 - Cinema, image synthesis, scientific visualization
 - Law enforcement (license plate reading, speed, finger print...)
 - ...

Examples of fields that use DIP

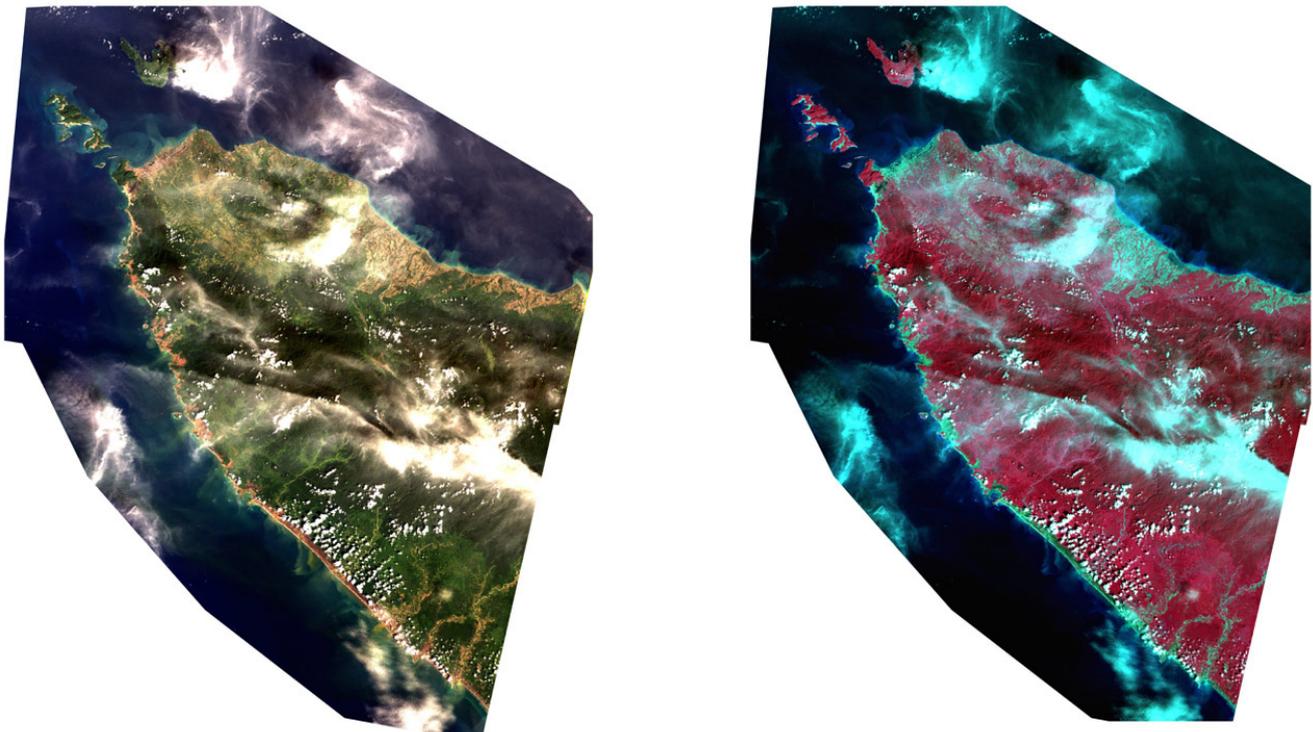
- Images based on radiation from electromagnetic spectrum



11

Examples

Landsat images after the tsunami in Indonesia 2004,



Left: in natural color (Landsat ETM+ bands 1,2,3 RGB)

Right: in false-color composite (Landsat ETM+ bands 4(near infrared),3,2 RGB). In this image vegetation appears in red, pink, and maroon; water appears in blue to black; urban and non-vegetated areas (including the tsunami damage regions) appear in bluish-greens and grays.

Département GE - DIP - Thomas Grenier

12

Examples

Landsat images before the tsunami in Indonesia 2004,

Before
(14/05/02)



In false-color composite (*Landsat* ETM+ bands 4,3,2 RGB). In this image vegetation appears in red, pink, and maroon; water appears in blue to black; urban and non-vegetated areas (including the tsunami damage regions) appear in bluish-greens and grays.

Département GE - DIP - Thomas Grenier

13

Examples

Landsat images after the tsunami in Indonesia 2004,

After
(29/12/04)



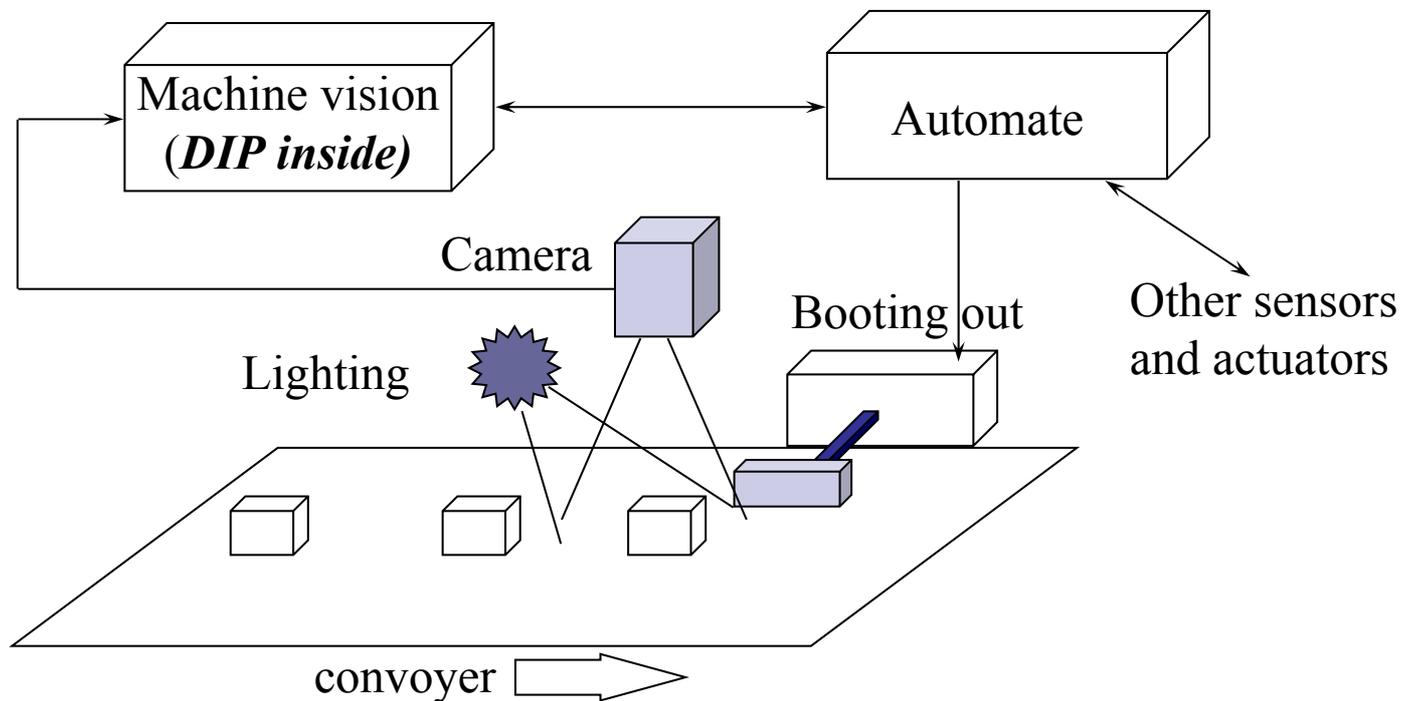
In false-color composite (*Landsat* ETM+ bands 4,3,2 RGB). In this image vegetation appears in red, pink, and maroon; water appears in blue to black; urban and non-vegetated areas (including the tsunami damage regions) appear in bluish-greens and grays.

Département GE - DIP - Thomas Grenier

14

Examples of fields that use DIP

■ Industrial inspection, computer vision



Département GE - DIP - Thomas Grenier

15

Computer vision constrains

■ Robust in respect of

- snapshot conditions
 - Lighting, camera settings...
- (tolerated) variations of the product to control or monitor
 - Shape, position, color....
- environment
 - Temperature, dust, moisture, place ...
- Human being
 - User-friendly, efficiency, ...

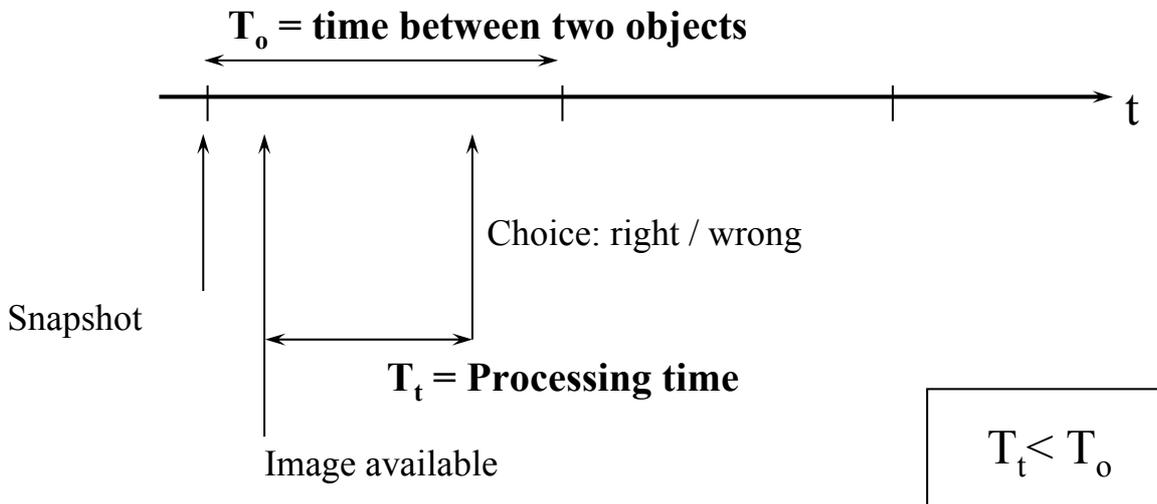
Département GE - DIP - Thomas Grenier

16

Computer vision constrains

- Real Time processing

→ Rate of the objects to control

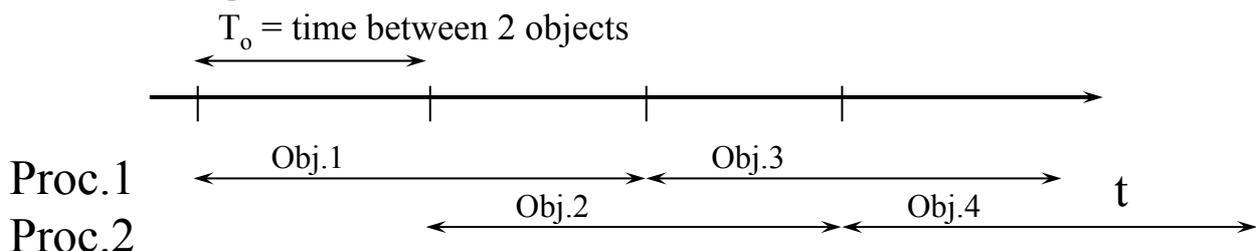


Technical solutions

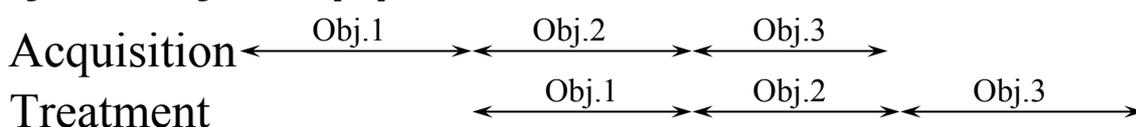
- A good lighting, a good snapshot are better than an elaborate processing

- Image analysis can not bring any information that are not present in the image

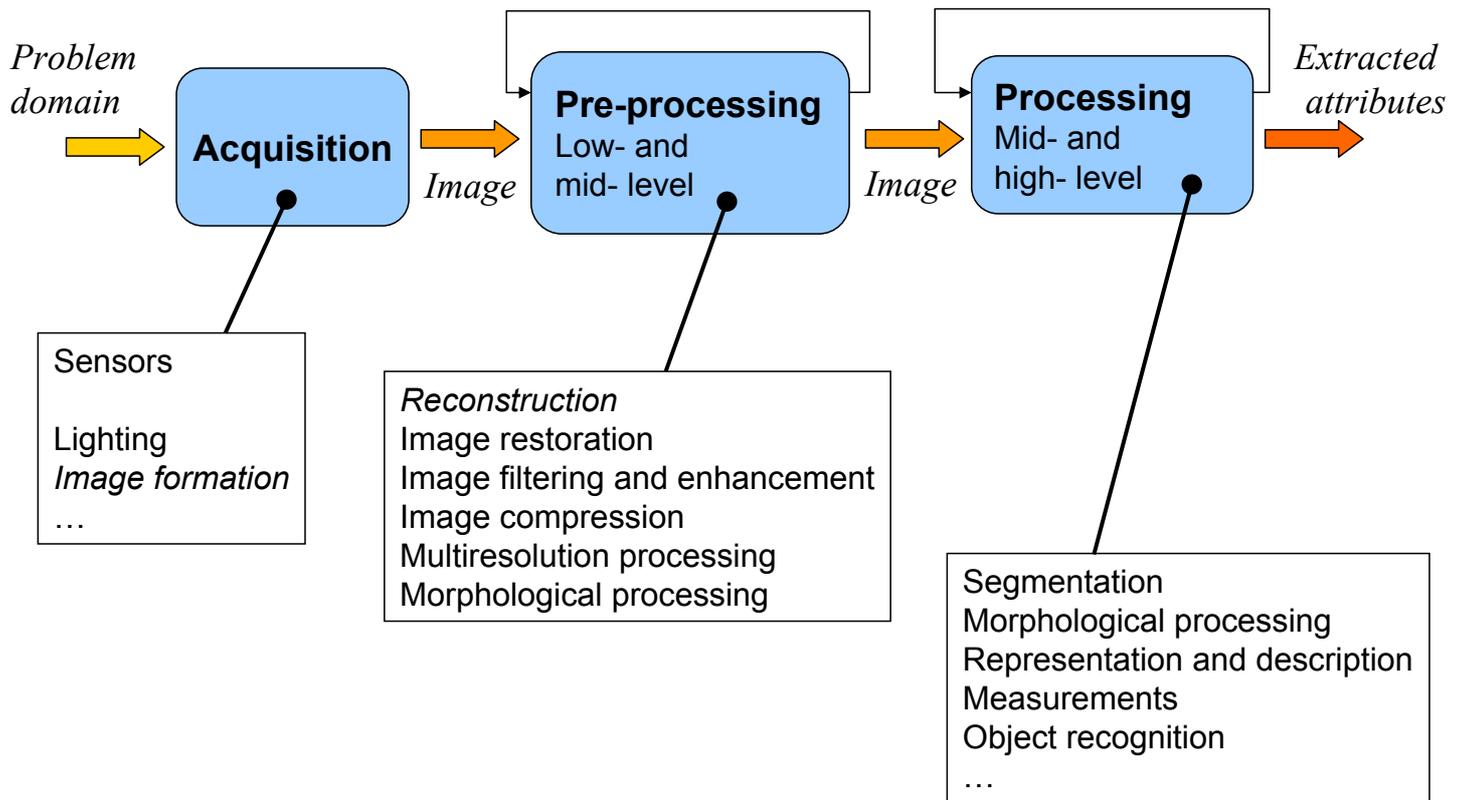
- For elaborate processing, you may use parallel processing



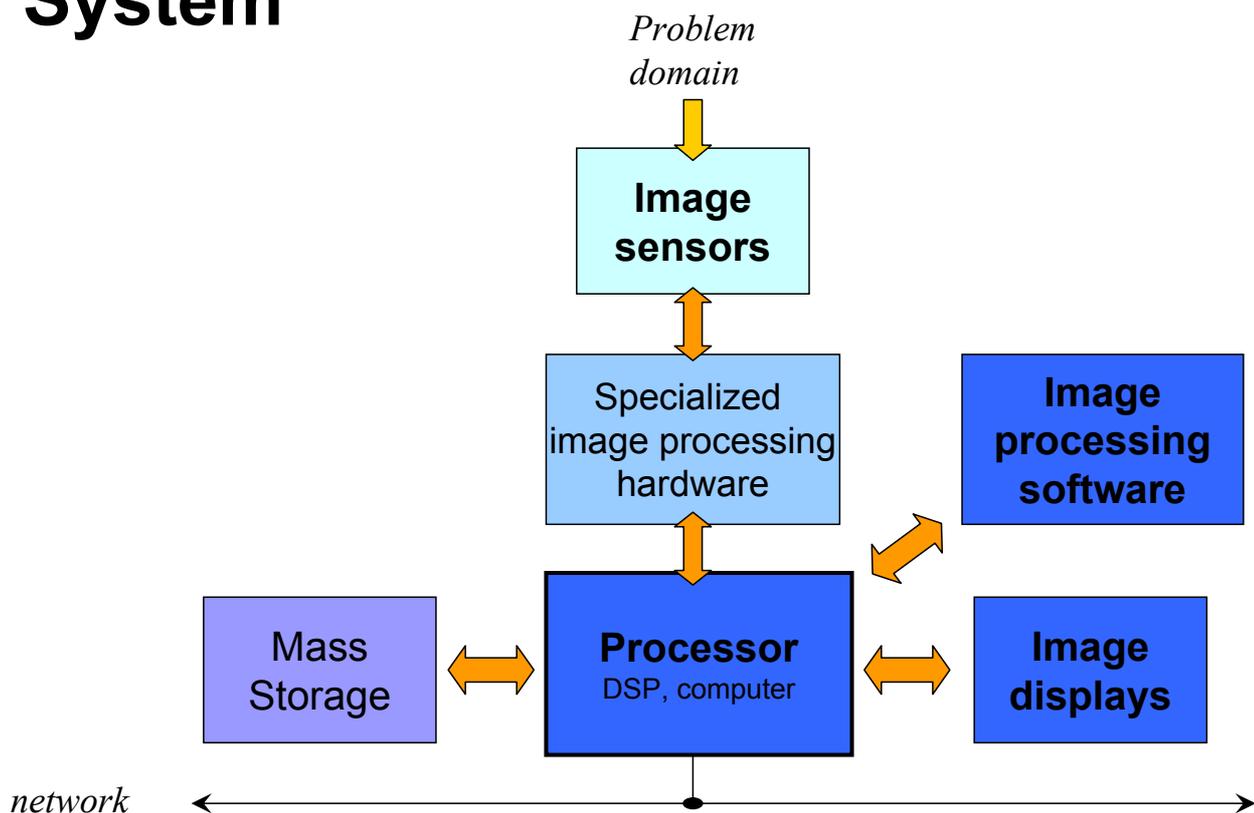
- Or you may use *pipelines*



Fundamental steps in DIP



Components of an image processing System



And you ?

- Technical skills needed in computer vision
(including *DIP*)
 - Optics, physics
 - Mechanics
 - Electronics
 - Control theory
 - Image processing
 - Artificial intelligence
 - Computer science
 - interpersonal relationship
 - ...



Mars Rover

Digital Image Processing

Digital Image Fundamentals

*Département Génie Electrique
5GE - TdSi*

Summary

I. Introduction

- DIP ?, Examples, Fundamental steps, components

II. Digital Image Fundamentals

- Visual perception, light
- Image sensing, acquisition, sampling, quantization
- Linear, and non linear operations

III. Discrete 2D Processing

- Vector space, Convolution
- Unitary Transform

IV. Image Improvement

- Enhancement, restoration, geometrical modifications

Digital Image Fundamentals

- Visual perception
- Light
- Image Sensing and Acquisition
- Image Sampling and Acquisition
- Basic relationships between pixels
- Linear and nonLinear operations

→ Books :

Digital Image Processing, Gonzales, Prentice Hall (3Ed.)

Digital Image Processing, Jahne, Springer (6Ed.)

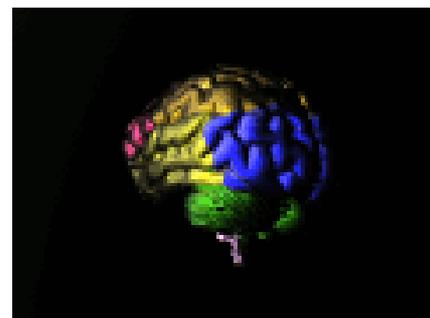
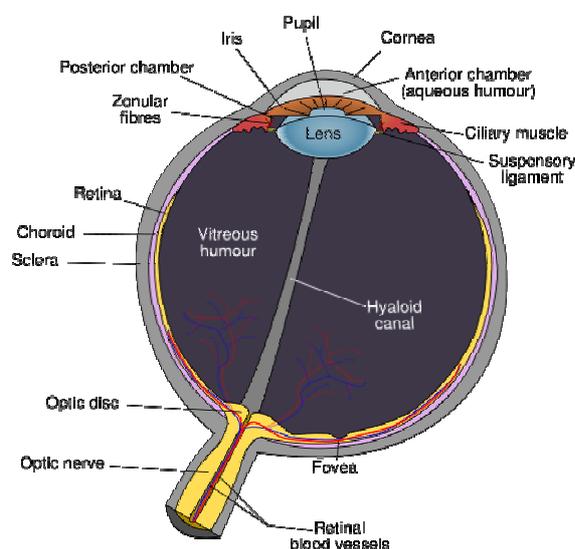
Visual perception

- Human visual perception

Eyes

+

Brain



Visual perception

■ Brightness adaptation and discrimination

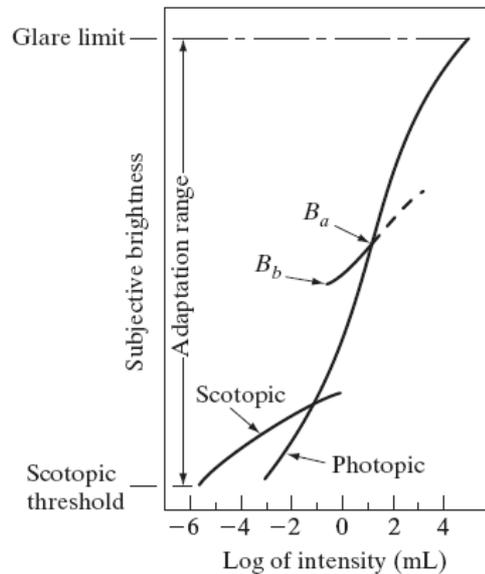
□ Digital images are displayed as a discrete set of intensities

➔ Range of light intensity levels to which the human visual system can adapt?

Enormous: on the order of 10^{10}

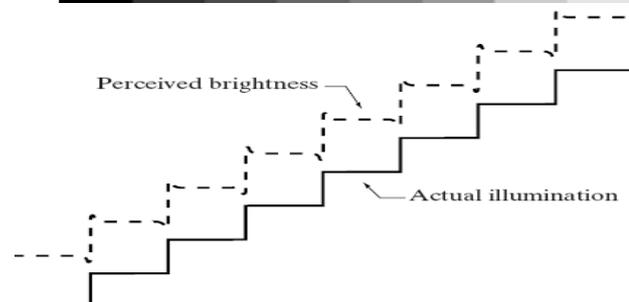
Scotopic vision is the monochromatic vision of the eye in low light

Photopic vision is the vision of the eye under well-lit conditions. In humans and many animals, photopic vision allows color perception, mediated by cone cells.



Visual perception

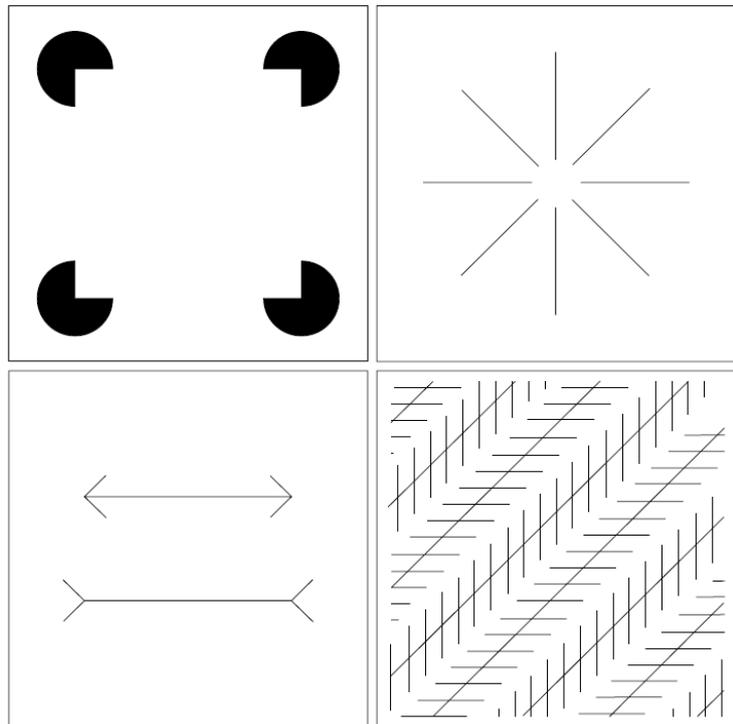
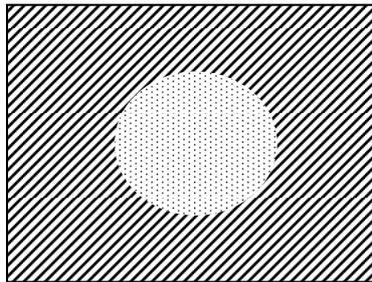
■ Brightness adaptation & discrimination



Visual perception

■ Other examples of human perception

phenomena:
optical illusions



Département GE - DIP - Thomas Grenier

7

Light and electromagnetic spectrum

- A 'light' (illumination) source is needed to generate an image. The elements of the 'scene' should reflect or absorb the energy of that source (or be that source)
- Imaging is based predominantly on energy radiated by electromagnetic waves
- Other methods for image generation exist
 - Sound reflected on objects (ultrasound image)
 - Electron beams (microscopy)
 - Synthetic images (graphics and visualization)
 - Mechanic sensors
 - ...

Département GE - DIP - Thomas Grenier

8

■ Electromagnetic spectrum

Wavelength (m) $\rightarrow \lambda = \frac{c}{f}$

Photon energy (eV, J) $\rightarrow E = \frac{hc}{\lambda}$

$\rightarrow E = hf$ \leftarrow frequency

Where:

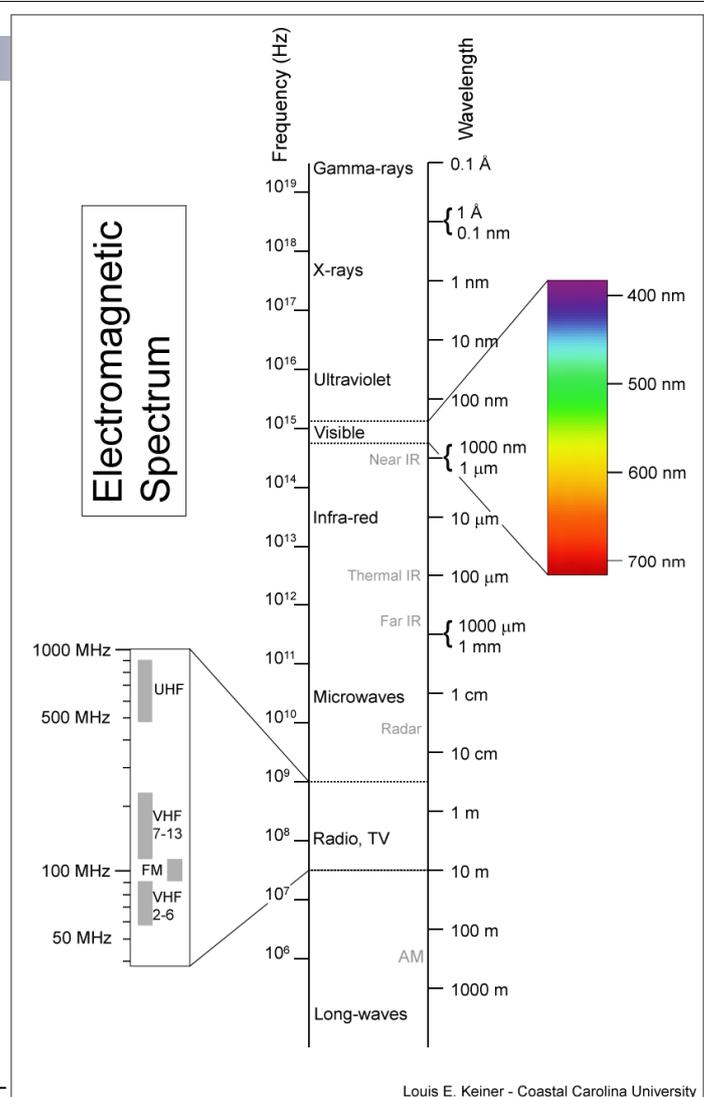
c is the speed of light
 h is the Planck's constant

$$c = 299792.458 \text{ m/s}$$

$$h = 6.626069 \cdot 10^{-34} \text{ J}\cdot\text{s}$$

$$\approx 4.13567 \text{ meV/Hz}$$

Département GE -



Louis E. Keiner - Coastal Carolina University

9

Light sources

■ Monochromatic (or achromatic) light

- Light that is void of color

■ Chromatic (color) light

- Light that spans the electromagnetic energy spectrum (from 0.43 to 0.79 μm)

■ Basic quantities used to describe the quality of sources

- **Radiance** (in Watts): total amount of energy that flows the light source
- **Luminance** (in lumens, lm): gives a measure of the amount of energy that an observer perceives from a light source
- **Brightness** (measurable ??): subjective descriptor of light perception (one of the key factors in describing color sensation)

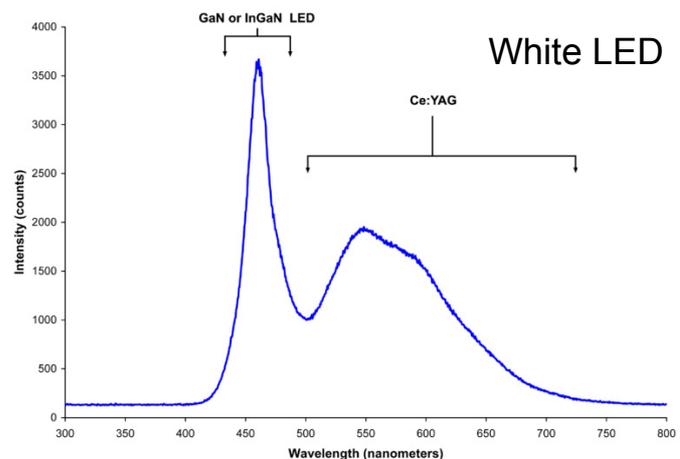
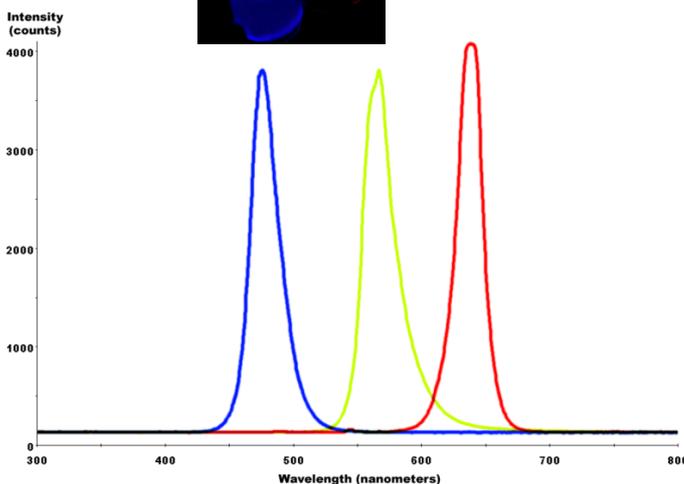
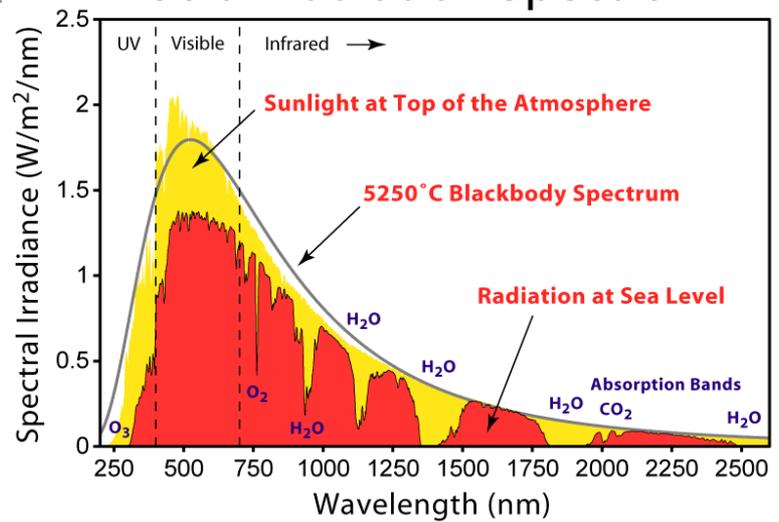
Light sources

Quality of these sources?

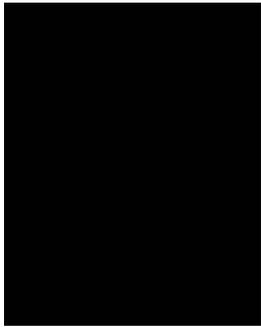
- Sun
- Candle
- Conventional incandescent light bulbs
- Fluorescent lamp
- Light-Emitting Diode (LED)
- Laser
- Xenon flash lamp



Solar Radiation Spectrum



- Flashes can be used for capturing quickly moving objects



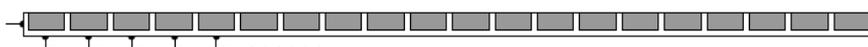
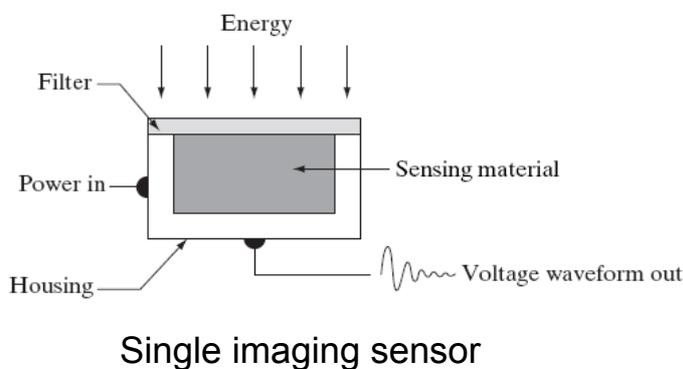
Xenon flash



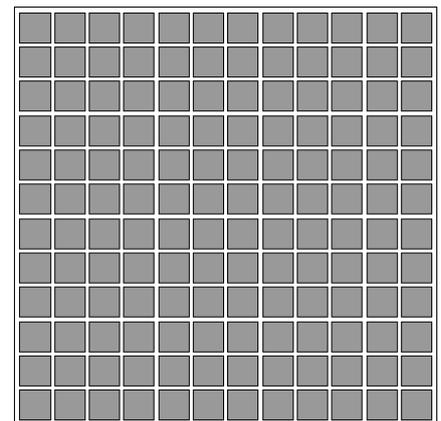
Running water "frozen" by flash

Image Sensing and Acquisition

- Sensor arrangements used to transform illumination energy into digital images



Line sensor



Array sensor

Image Sensing and Acquisition

■ Image acquisition (single and line sensors)

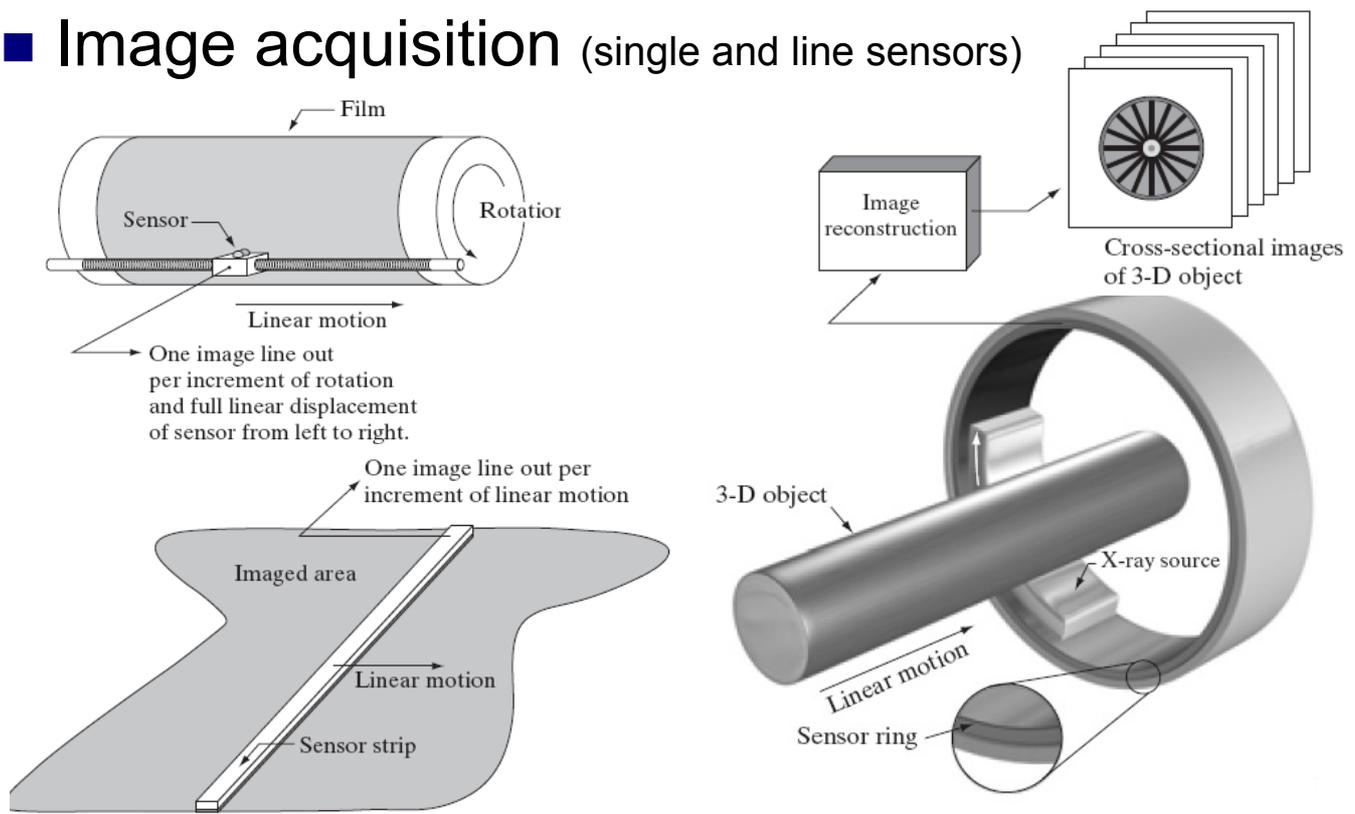


Image Sensing and Acquisition

■ Image acquisition (array sensor)

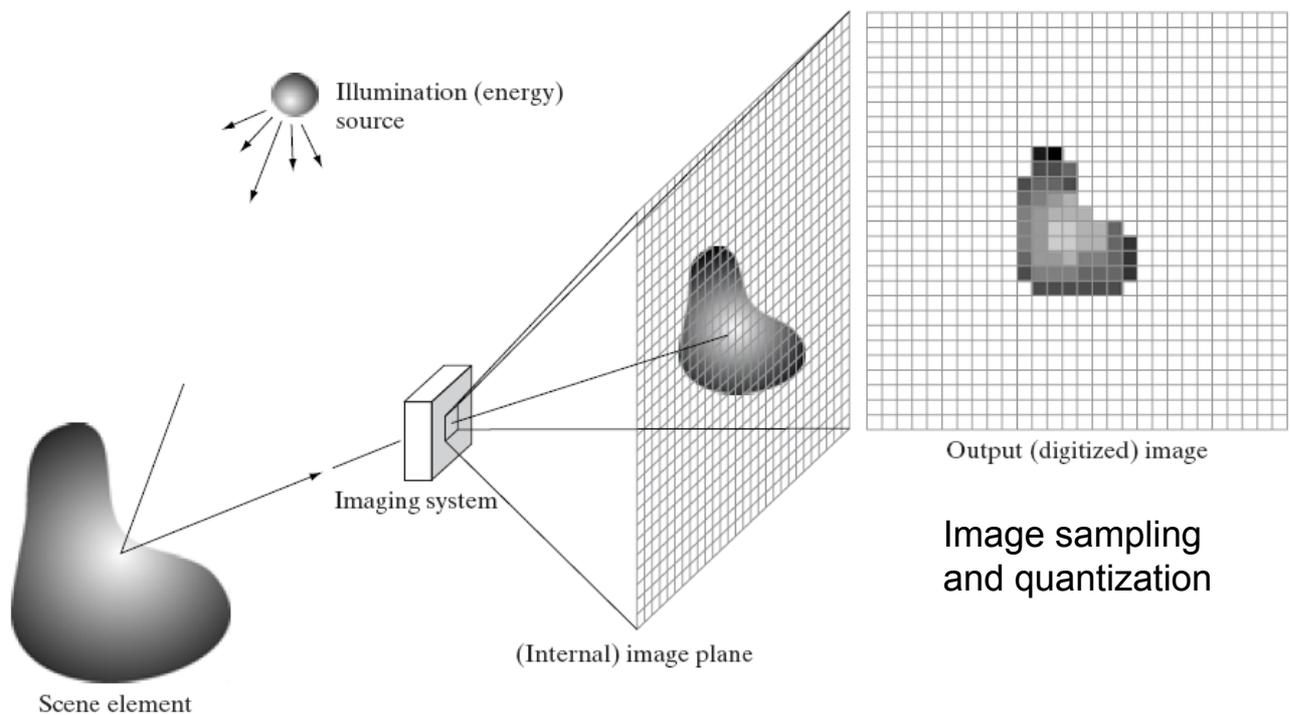


Image Sensing and Acquisition

■ Image acquisition

- camera, object and source positions

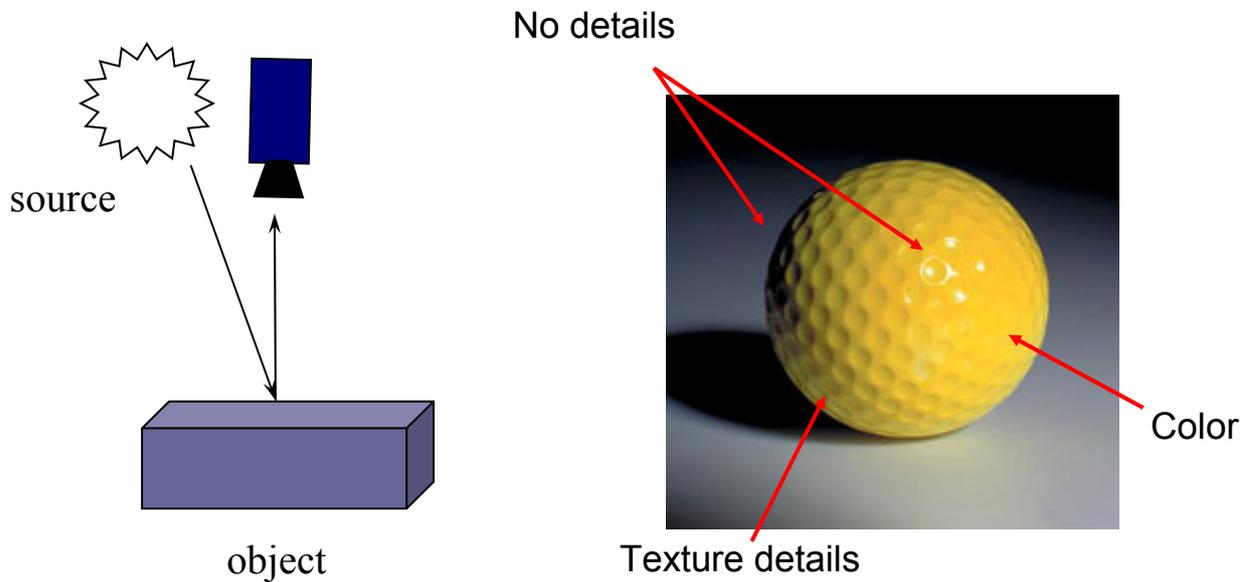


Image Sensing and Acquisition

■ Image acquisition

- camera, object and source positions

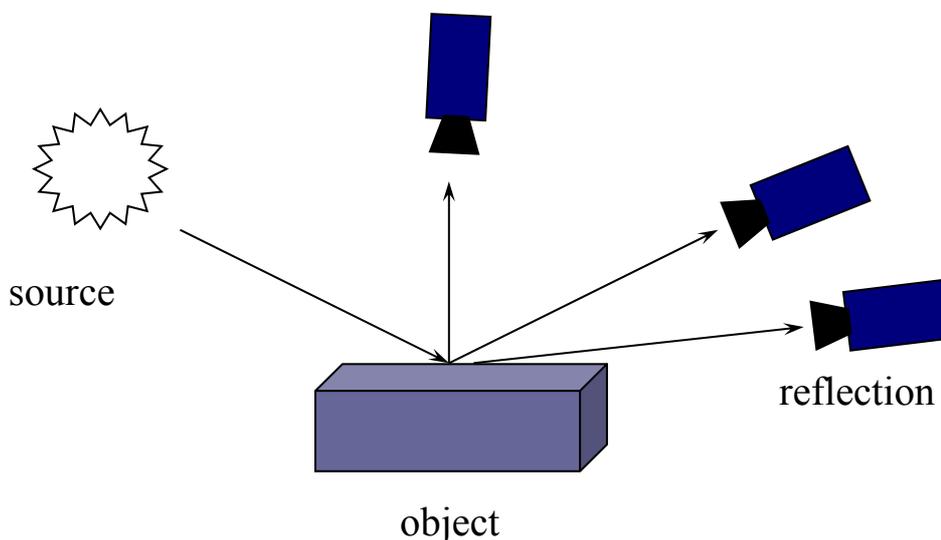
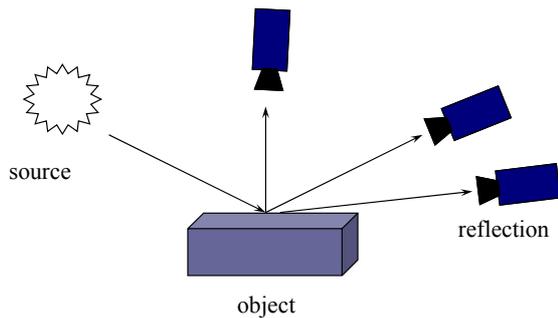


Image Sensing and Acquisition

■ Image acquisition:

- camera, object and source positions



Département GE - DIP

19

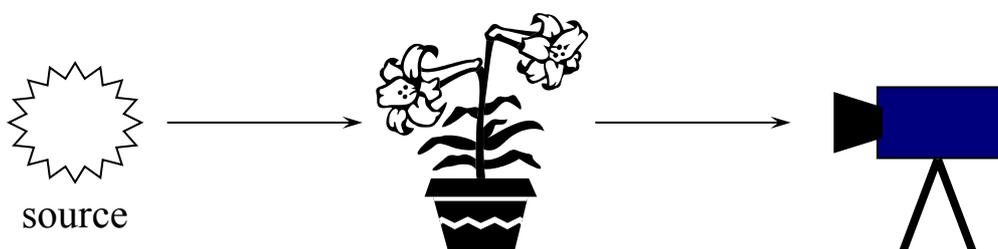
Image Sensing and Acquisition

■ Image acquisition: contre-jour (or backlighting)

Transmission: X, PET

- This effect usually

- Hides details,
- Causes a stronger contrast between light and dark,
- Creates silhouettes,
- Emphasizes lines and shapes.

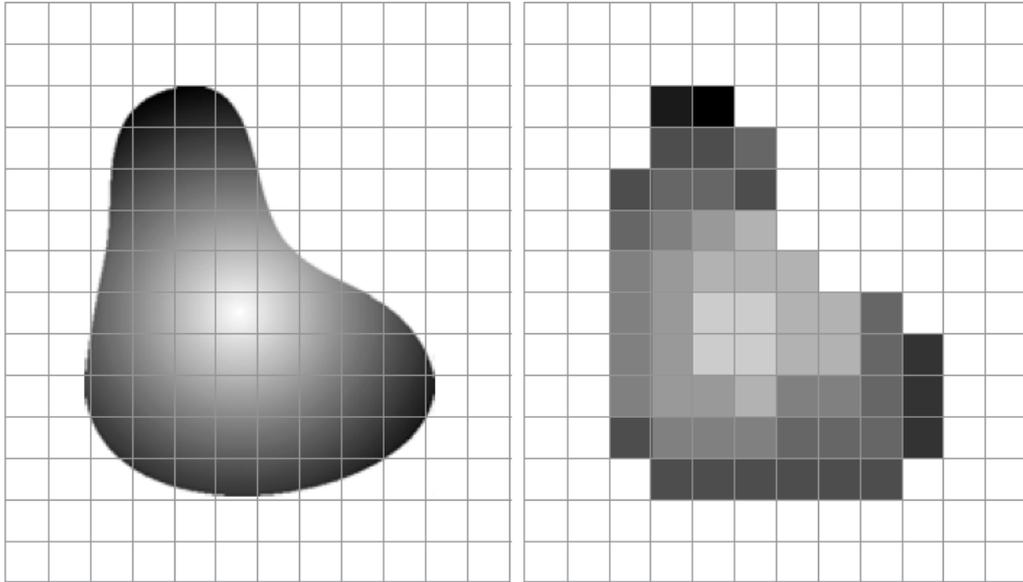


Département GE - DIP - Thomas Grenier

20

Image Sampling and Quantization

Digitization : sampling and quantization



Continuous image projected
on an array sensor

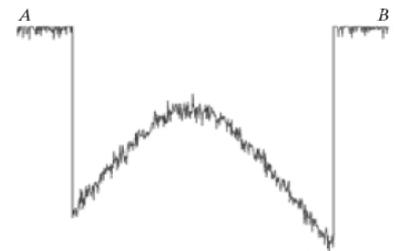
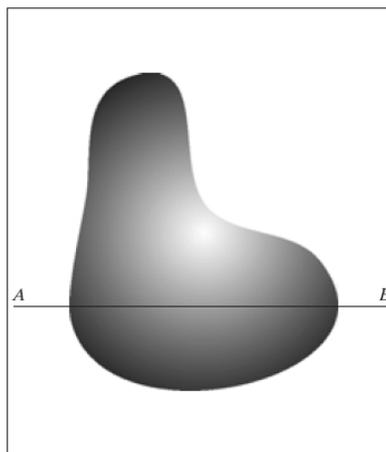
Result of image sampling
and quantization

Image Sampling and Quantization

Definitions

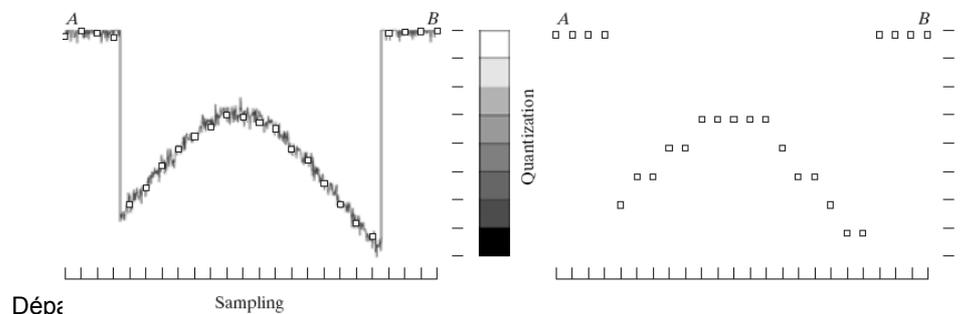
Sampling:

Digitizing the coordinate values
(spatial resolution)



Quantization:

Digitizing the amplitude values
(intensity levels)



Quantization

→ Number of intensity levels typically is an integer power of two (often 256 : 1 Byte = 8bits per pixel), the discrete levels are equally spaced

■ Effects of reducing the number of intensity levels



8 bits (256 levels)



4 bits (16 levels)



2 bits (4 levels)

- false contours appear
- quantification noise
- visible (eyes) effect under 6/7 bits
- quantification for the display : 8bits

Département GE - DIP - Thomas Grenier

23

Sampling

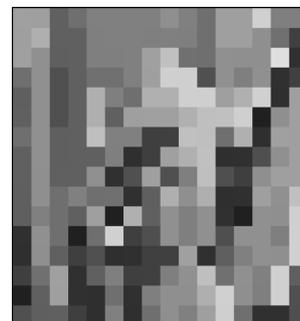
■ Effects of reducing spatial resolution



256 x 256 pixels



64 x 64 pixels



16 x 16 pixels

- ugly contours (by steps)
- blur effect
- Details are less precise / detectable
- Resolution loose

Département GE - DIP - Thomas Grenier

24

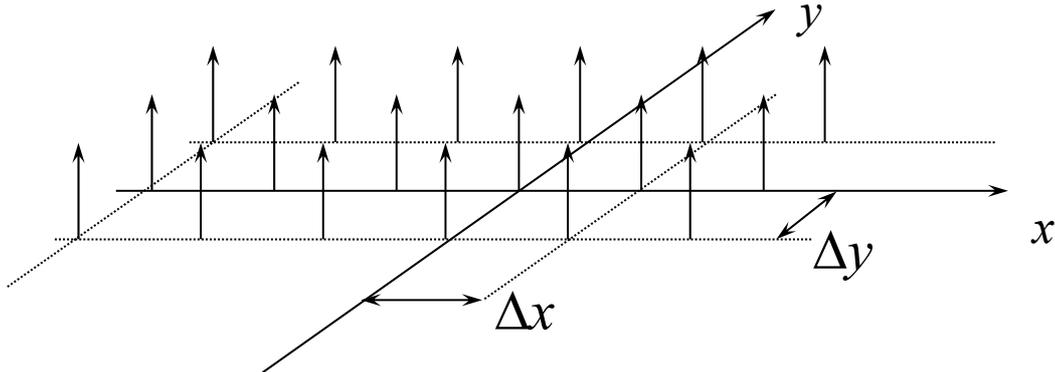
Sampling

■ Sampling of function $f(x,y)$

$$f_e(x,y) = f(x,y) \cdot \sum_i \sum_j \delta(x - i \Delta x, y - j \Delta y)$$

Δx sampling distance in direction of x

Δy sampling distance in direction of y



$\sum_i \sum_j \delta(x - i \Delta x, y - j \Delta y)$: 2D Dirac comb

→ Δx and Δy limits ?

Sampling

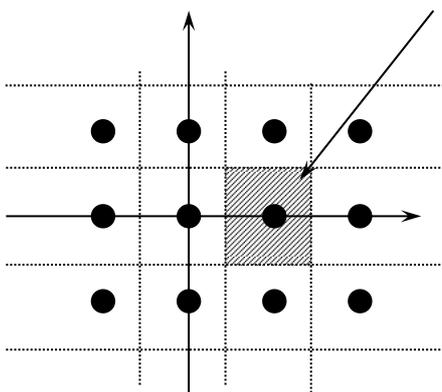
$$f_e(x,y) = \sum_i \sum_j \tilde{f}(i\Delta x, j\Delta y) \delta(x - i\Delta x, y - j\Delta y)$$

■ The dirac weight of $\tilde{f}(i\Delta x, j\Delta y)$ is:

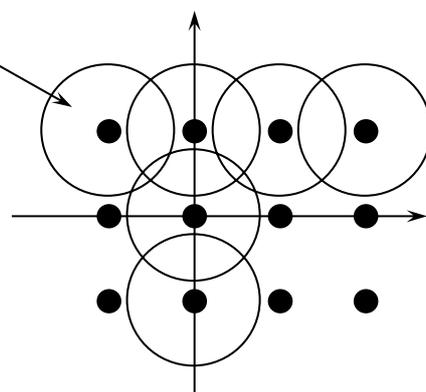
- The value of $f(x,y)$ at $x = i \Delta x$ and $y = j \Delta y$
- The 'mean' value of $f(x,y)$ around $(i \Delta x, j \Delta y)$

$f(x,y)$ is weighted by h and integrated over R

$$\tilde{f}(i\Delta x, j\Delta y) = \int_R f(x,y) h(i\Delta x - x, j\Delta y - y) dx dy$$



i.e: CCD Camera



i.e: video camera tube or pickup tube

Sampling

■ Sampling distances limits

- Sampling leads to a reduction in resolution,
 - structures of about the scale of the sampling distance and finer will be lost
- Sampling theorem
 - We will obtain a periodic structure correctly only if we take at least two samples per wavelength
 - Else : Moiré effect (*'aliasing'* for 1D signal)

Sampling distances: *Moiré effect*



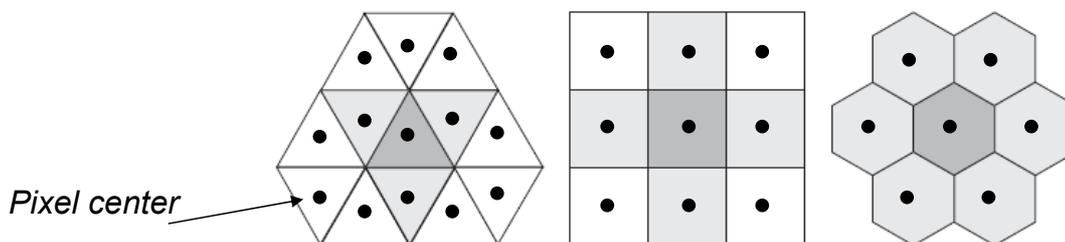
Properly sampled image of brick wall.



Spatial aliasing in the form of a Moiré pattern

Image representation

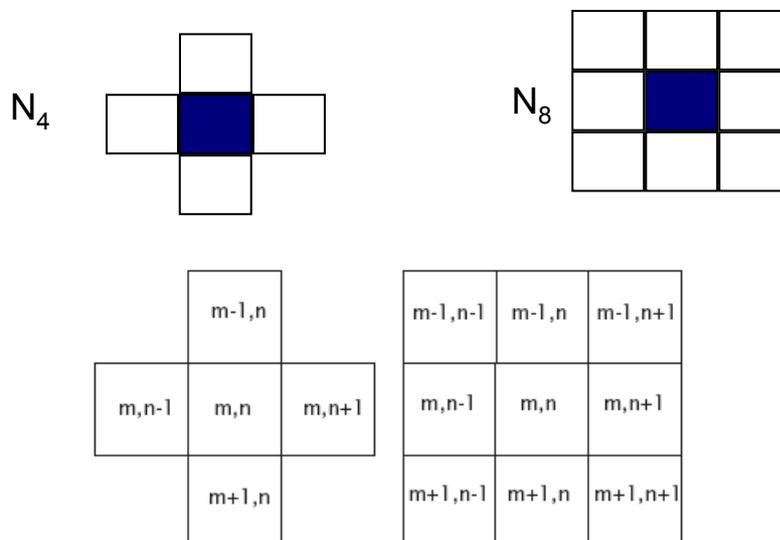
- A rectangular grid is only the simplest geometry for a digital image ($\Delta x = \Delta y$)
- Other geometrical arrangements of the pixels and geometric forms of the elementary cells are possible



The three possible regular grids in 2-D: **left** triangular grid, **center** square grid, **right** hexagonal grid

Basic relationships between pixels

- Neighborhoods
 - 4- and 8- neighbors of a pixel



Basic relationships between pixels

■ Adjacency

- Two pixels p and q are **4-adjacent** if q is in the set $N_4(p)$
- Two pixels p and q are **8-adjacent** if q is in the set $N_8(p)$

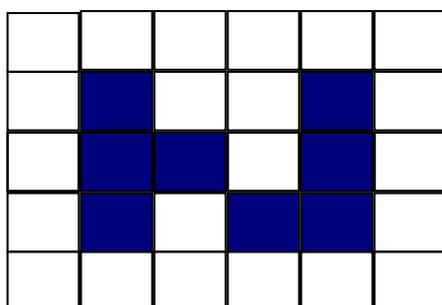
■ Connectivity

- A (digital) **path** from pixel p to pixel q is a sequence of distinct pixels where the next pixels is adjacent to the previous one
- Let S represent a subset of pixels in an image. Two pixels p and q are said to be **connected** in S if there exists a **path** between them consisting entirely of pixels in S

Basic relationships between pixels

■ Region

- Let R be a subset of pixels in an image. We call R a region of the image if R is a connected set
- Regions that are not adjacent are said to be *disjoint*



Two regions if 4-adjacency is used
One region if 8-adjacency is used

Basic relationships between pixels

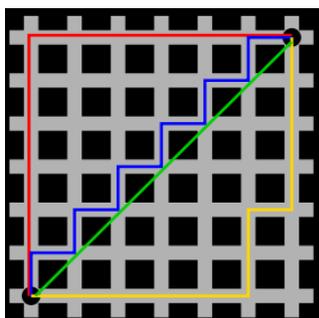
■ Boundaries

- The boundary (or border, or contour) of a region R is the set of points that are adjacent to points in the complement of R (*the background*)
- Or... the border of a region is the set of pixels in the region that have at least one 'background' neighbor

Basic relationships between pixels

■ Distances between to pixels $f[i,j]$ et $f'[k,l]$

- Euclidian Distance $d_e(f, f') = \sqrt{(i-k)^2 \Delta x^2 + (j-l)^2 \Delta y^2}$
- *City-Block* distance $d_c(f, f') = |i-k|\Delta x + |j-l|\Delta y$
Taxicab, Manhattan distance, D_4 distance
- Chessboard distance $d_b(f, f') = \max(|i-k|\Delta x, |j-l|\Delta y)$
 D_8 distance
- ...



Euclidian vs City block distances

3	2	2	2	2	2
3	2	1	1	1	2
3	2	1	X	1	2
3	2	1	1	1	2
3	2	2	2	2	2
3	3	3	3	3	3

Chessboard distance

Linear and nonLinear operations

- Important classification of an image-processing method

- Is it a linear or a a nonlinear method ?

- Let H be a general operator

$$H[f(x, y)] = g(x, y)$$

- H is said to be a **linear operator** if

$$\begin{aligned} H[a_i f_i(x, y) + a_j f_j(x, y)] &= a_i H[f_i(x, y)] + a_j H[f_j(x, y)] \\ &= a_i g_i(x, y) + a_j g_j(x, y) \end{aligned}$$

- ➔ Tools: vector-matrix, PDE, set, *and related operators (+, *, convolution)...*

Digital Image Processing

Discrete 2D Processing

*Département Génie Electrique
5GE - TdSi*

Summary

I. Introduction

- DIP, Examples, Fundamental steps, components

II. Digital Image Fundamentals

- Visual perception, light
- Image sensing, acquisition, sampling, quantization
- Linear and non linear operations

III. Discrete 2D Processing

- Vector space, color space
- Operations (arithmetic, geometric, convolution, ...)
- Image Transformations

IV. Image Improvement

- Enhancement, restoration, geometrical modifications

Discrete 2D Processing

- Vector space, colour space
- Operations on images
 - Arithmetic operations
 - Set and logical operations
 - Spatial operations
 - Geometric
 - Convolution
- Image transformations
 - Unitary Transforms

Vector space and Matrix

■ Vector and Matrix Operations

□ Vector

- Spatial position of pixel
- Pixel intensities

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad \text{i.e. : RGB}$$

➔ once pixels have been represented as vectors, we can use the tools of vector-matrix theory

□ Euclidean distance

$$D(\mathbf{a}, \mathbf{b}) = [(\mathbf{a} - \mathbf{b})^T (\mathbf{a} - \mathbf{b})]^{1/2}$$

□ Linear transformations

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{b})$$

Vector space and Matrix

Vector and Matrix Operations

Other vector forms

- Joint Spatial-range domain

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_s \\ \mathbf{x}_r \end{bmatrix} \begin{array}{l} \leftarrow \text{location} \\ \leftarrow \text{intensities} \end{array} \quad \text{i.e.} \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ r \\ g \\ b \end{bmatrix}$$

Image

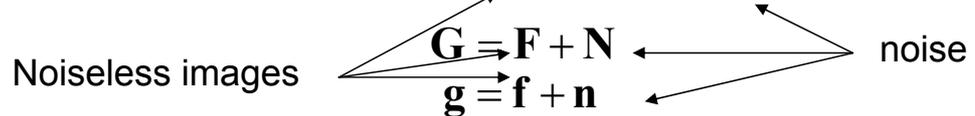
- As function
- As matrix ($M \times N$)
- As vector ($MN \times 1$)
- i.e. adding noise:

$$f(x, y) \Leftrightarrow f(\mathbf{x}_s) = \mathbf{x}_r$$

$$\mathbf{I}[i, j] \Leftrightarrow f(i \cdot \Delta x, i \cdot \Delta y) = \mathbf{x}_r$$

$$\mathbf{f}[k] = (\mathbf{x}_s, \mathbf{x}_r) \Leftrightarrow \mathbf{I}[k \% M, \lfloor k / N \rfloor] = \mathbf{x}_r$$

$$g(x, y) = f(x, y) + \eta(x, y)$$



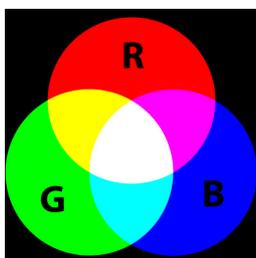
Color spaces

Color fundamentals

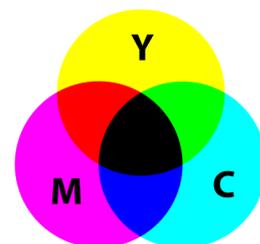
- Colors are seen as variable combinations of **primary colors**: Red, Green and Blue
- Standardization of the specific wavelength values by the CIE (Commission Internationale de l'Eclairage)
 - Red = 700nm, green=546.1nm, blue=435.8nm
- 2 kinds of mixture
 - Mixture of light (additive primaries)
 - Mixture of pigment (subtractive primaries)



additive



subtractive



Color spaces

- 3 Characteristics used to distinguish one color from another:
 - **Brightness**: achromatic notion of intensity
 - **Hue**: dominant color perceived (wavelength)
 - **Saturation**: relative purity (or the amount of white light within a hue)
- Chromaticity: hue+saturation
- ➔ A color can be characterized by its brightness and chromaticity

Color spaces

- Tristimulus values:
 - amount of red (X) green (Y) and blue(Z)
- Trichromatic coefficients (not spatial position!)

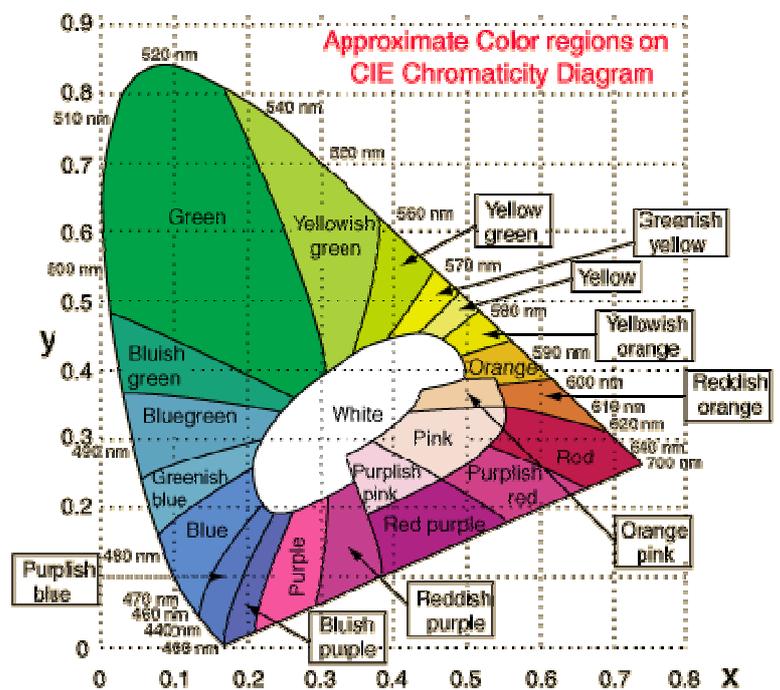
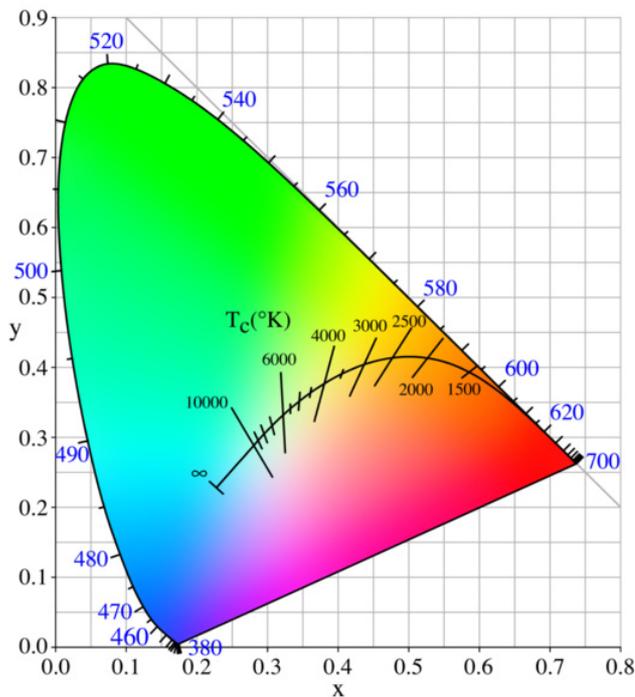
$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad z = \frac{Z}{X+Y+Z}$$

$$\Rightarrow x + y + z = 1$$

- Example:
 - a yellowish green :
 - x = 30% Red; y=60% Green; (z=10% Blue)

Color spaces

■ CIE Chromaticity diagram



Département GE - DIP - Thomas Grenier

9

Color spaces

- A color model (also called color space or color systems) is a specification of a coordinate system.
- Color models in use may be
 - Hardware-driven (monitors, printers)
 - Application-driven (creation, color graphic animation, ...)
- Commonly used color models
 - RGB (monitors, color video cameras)
 - CMY, CMYK (printing)
 - YUV (TV, MPEG, ...)
 - HSI (similar to human description and interpretation of colors)
 - *Pseudocolor and Look Up Tables*

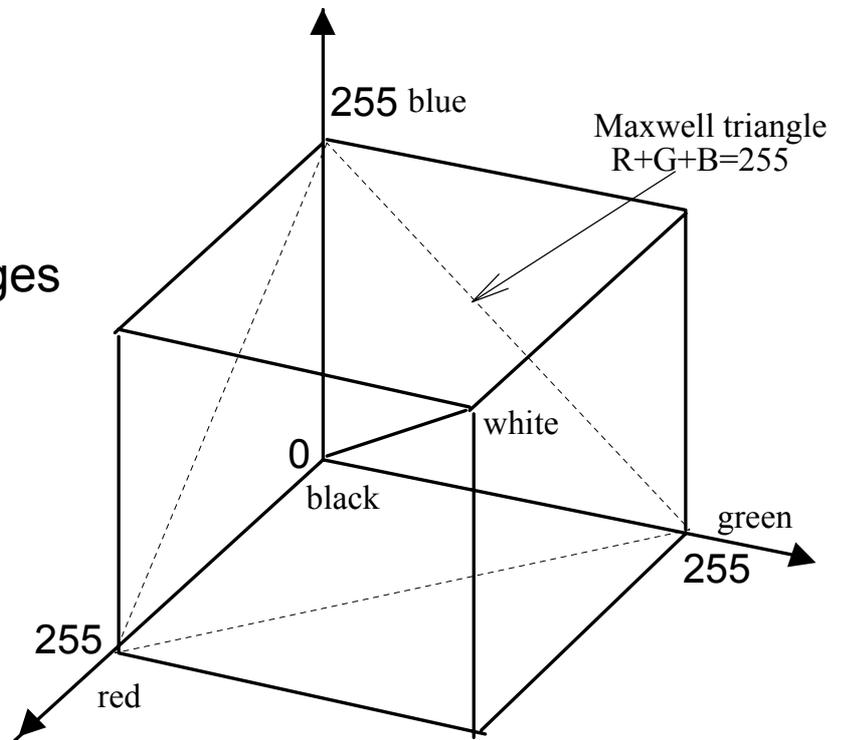
Département GE - DIP - Thomas Grenier

10

RGB color model

■ Additive synthesis of color (mixtures of light)

- Display hardware
 - monitors
 - CRT, LCD, plasma
 - graphic card
- 24 bits (3*8 bits) Images
- About 16M colors
- Grey-levels images
 - R=G=B



CMY Color model

■ Cyan Magenta Yellow

- Subtractive color model
 - Describes the color reflected by an illuminated surface absorbing certain wavelengths
 - Is needed by devices that deposit colored pigments
 - i.e. on paper
 - CMYK: black color added for true black

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \quad \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \rightarrow \begin{bmatrix} C - \min(C, M, Y) \\ M - \min(C, M, Y) \\ Y - \min(C, M, Y) \\ \min(C, M, Y) \end{bmatrix} = \begin{bmatrix} C' \\ M' \\ Y' \\ K \end{bmatrix}$$

YUV Color model

- YUV: PAL; YIQ: NTSC
- Y → intensity (grey-levels!) (not yellow ☺)
- UV or IQ: chromaticity

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & -0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.522 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

→ YUV is better than RGB for information decorrelation

→ Compression of color images

$$\begin{bmatrix} R : 33.2 \\ V : 36.2 \\ B : 30.6 \end{bmatrix} \leftrightarrow \begin{bmatrix} Y : 93 \\ U : 5.3 \\ V : 1.7 \end{bmatrix}$$

HSI Color model

- Hue Saturation and ...
 - Intensity HSI, Brightness HSB, Lightness HSL, Value HSV (not exactly the same)
- This model attempts to describe perceptual color relationships more accurately than RGB

$$I = \frac{1}{3}(R + G + B) \quad \text{RGB} \rightarrow \text{HSI}$$

$$S = 1 - \frac{3 \cdot \min(R, G, B)}{R + G + B}$$

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1} \left(\frac{\frac{1}{2}((R - G) + (R - B))}{((R - G)^2 + (R - B)(G - B))^{\frac{1}{2}}} \right)$$

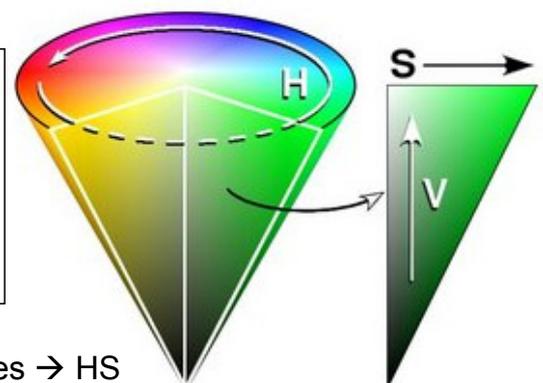
$$\text{YUV} \rightarrow \text{HSL}$$

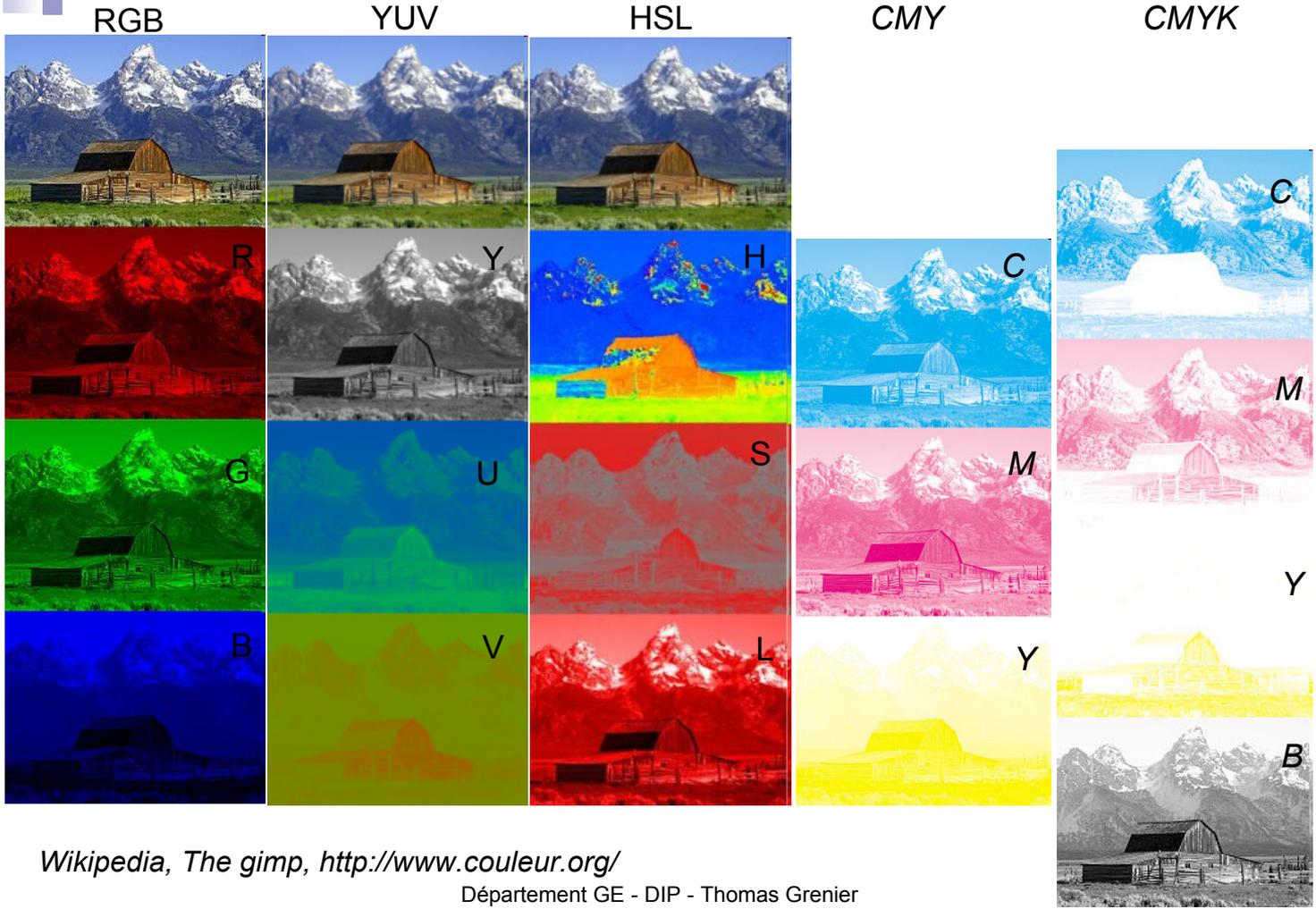
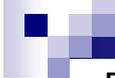
$$L = Y$$

$$H_{UV} = \tan^{-1}(V / U)$$

$$S_{UV} = \sqrt{U^2 + V^2}$$

UV → polar coordinates → HS



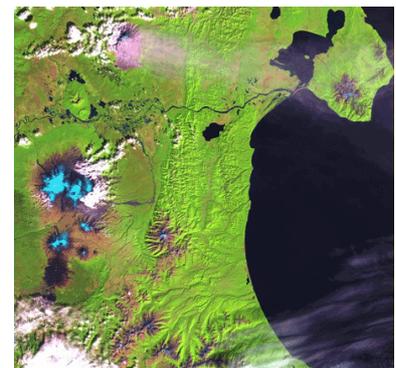


Pseudocolor

- Or false color visualization consists of assigning colors to gray values based on specified criteria, function(s), table(s) (LUT)



LUT



Example of a 3-color image with mixed components

Operations on images

■ Arithmetic Operations

□ $+, -, \times, \div$

□ Application: Corrupted image g obtained by adding the noise η to a noiseless image f

$$g(x, y) = f(x, y) + \eta(x, y)$$

■ assumptions :

- at every pair of coordinates (x, y) the noise is uncorrelated
- the noise has zero average value

→ Noise reduction by summing (averaging) a set of noisy images

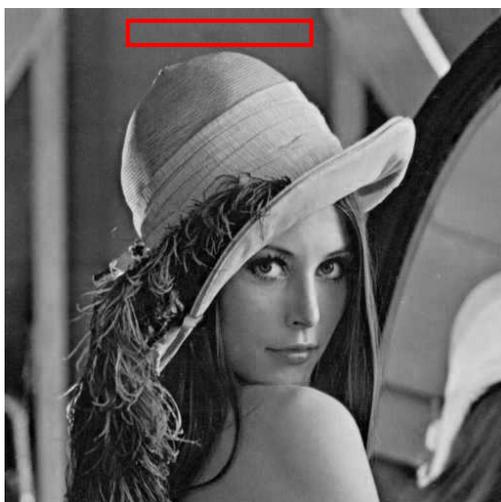
$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}} \cdot \sigma_{\eta(x,y)}$$

Département GE - DIP - Thomas Grenier

17

□ Noise Reduction

Original



$\sigma = 3,55$

Mean



1 $\sigma = 25,45$



2 $\sigma = 18,3$



4 $\sigma = 13,5$



6 $\sigma = 11,6$

Département GE - DIP - Thomas Grenier

18

□ Shading correction examples

***, /



=



Shading pattern



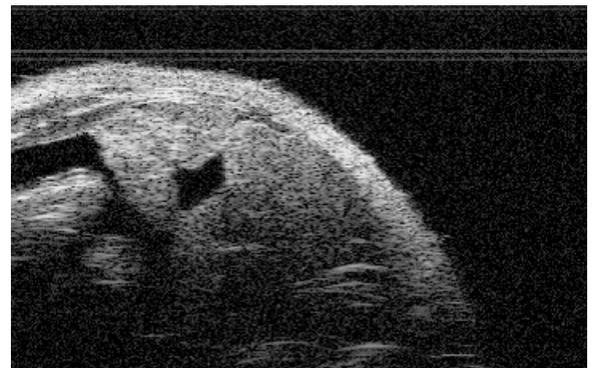
Grenier

19

□ Shading correction examples



*Knee of Guinea pig
MRI, 7 Telsa*



Ultrasound image

And:

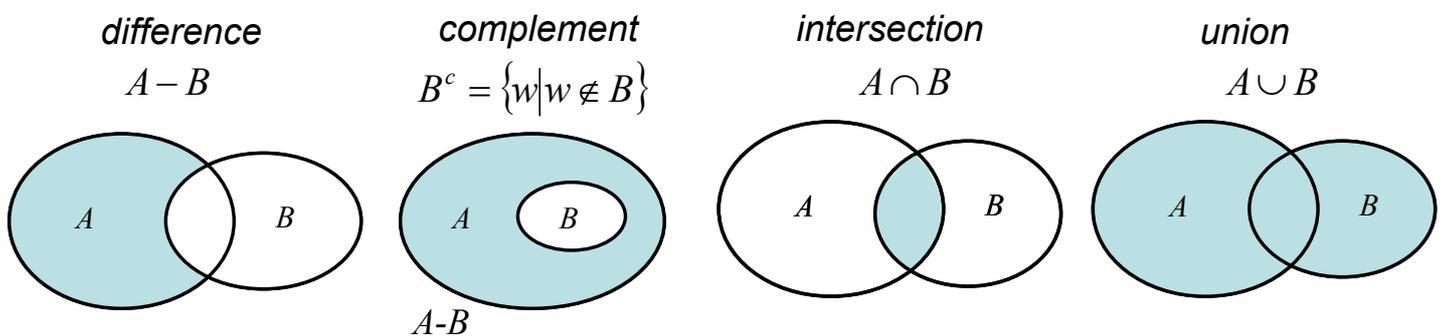
- perspective
- non uniform lighting (bulb filament, spot light)

Operations on images

■ Set and Logical Operations

□ Basic set operations

- a is an element of the set A : $a \in A, b \notin A$
- A set is represented with two braces $\{ \bullet \}$
- The set with no elements is called null or empty set: \emptyset
- Some operations



Département GE - DIP - Thomas Grenier

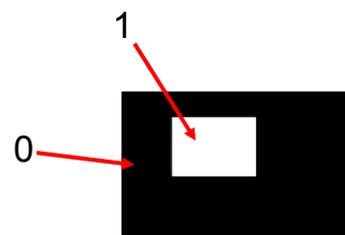
21

Operations on images

■ Set and Logical Operations

□ Logical operations

- Binary image:
 - $1 \rightarrow$ foreground
 - $0 \rightarrow$ background
- Operations:
 - AND, OR, NOT, XOR, AND-NOT, ...
 - Applicable on binary images or gray-level images!



□ Fuzzy sets

- For gradual transition from 0 to 1 (and 1 to 0)

Département GE - DIP - Thomas Grenier

22

Operations on images

■ Spatial Operations

- Spatial operations are performed directly on the pixels of a given image.
- 3 categories:
 - Single pixel operations
 - Neighborhood operations, Convolution
 - Geometric spatial transformations and image registration

Operations on images

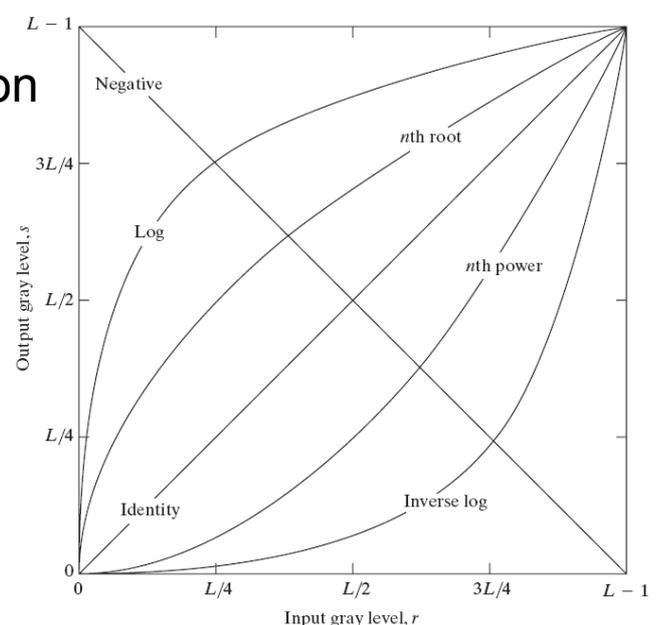
■ Spatial Operations - Single pixel operations

- r : original intensity,
- s : new intensity,
- T : a transformation function

$$s = T(r)$$

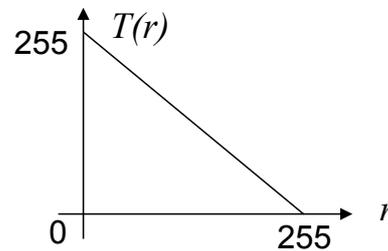
Some basic grey-levels transformation functions used for image enhancement

Range of intensities:
 $[0, L-1]$



Operations on images

- Spatial Operations - Single pixel operations
Example



Original



Negative



Operations on images

- Spatial Operations - Neighborhood operations
 - S_{xy} : set of coordinates of a neighborhood centered on a point (x,y) in an image f .
 - Neighborhood processing generates one corresponding pixel in the output image g at the same (x,y) coordinates.
 - The value of that pixel in g is determined by a operation involving the pixels in S_{xy} .

Operations on images

Spatial Operations - Neighborhood operations

Mean of S_{xy}

$$g(x, y) = \frac{1}{m.n} \sum_{(r,c) \in S_{xy}} f(r, c)$$

FIR filters...

Département GE - D

27

Operations on images

Convolution

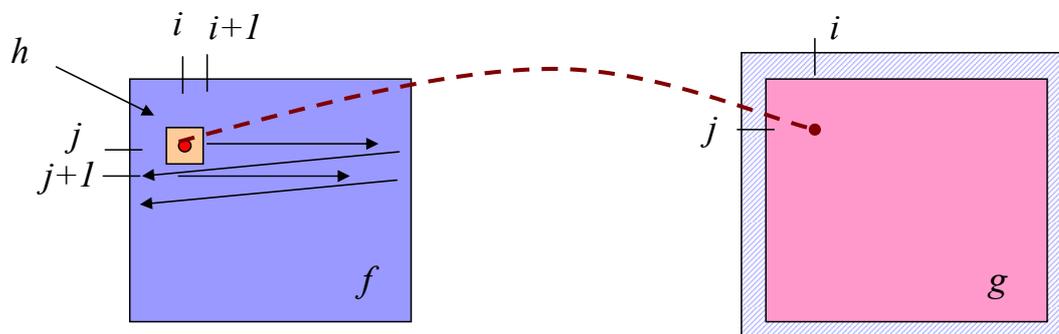
$$g(x, y) = h(x, y) * f(x, y) \quad (\text{two-dimensional convolution})$$

$$g(i, j) = \sum_{(k,l) \in H} h(k, l) f(i-k, j-l)$$

Output image

Convolution mask
Impulse response

Input image



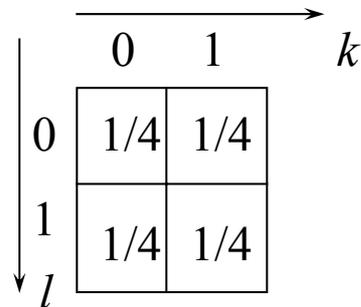
■ Example of Mean Filter

$$h(k,l) = \begin{cases} \frac{1}{(2M+1)(2N+1)} & \text{if } -M < k < M \text{ and } -N < l < N \\ 0 & \text{else} \end{cases}$$

W : 2x2 neighborhood

➤ $k = \{0;1\}$ $l = \{0;1\}$

$h(k,l) = 1/4$ for each (k,l)

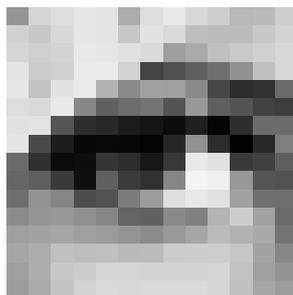


$$\begin{array}{cccc}
 0 & 1 & 2 & 2 \\
 1 & 1 & 2 & 1 \\
 1 & 2 & 0 & 0
 \end{array}
 \rightarrow
 \begin{array}{cccc}
 3/4 & 6/4 & 7/4 & x \\
 5/4 & 5/4 & 3/4 & x \\
 x & x & x & x
 \end{array}
 \xrightarrow{\text{(rounded to the integer part)}}
 \begin{array}{cccc}
 0 & 1 & 1 & x \\
 1 & 1 & 0 & x \\
 x & x & x & x
 \end{array}$$

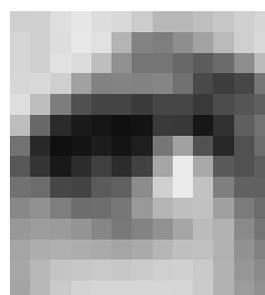
■ Example of Mean Filter



2x2
Mean Filter



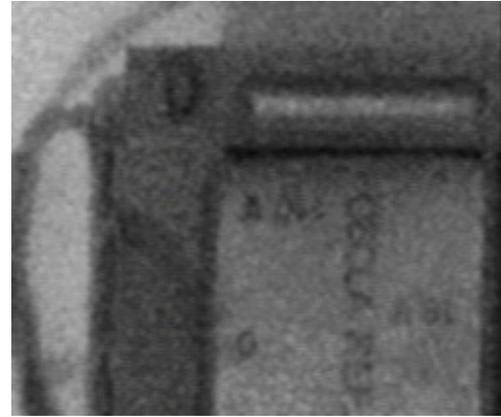
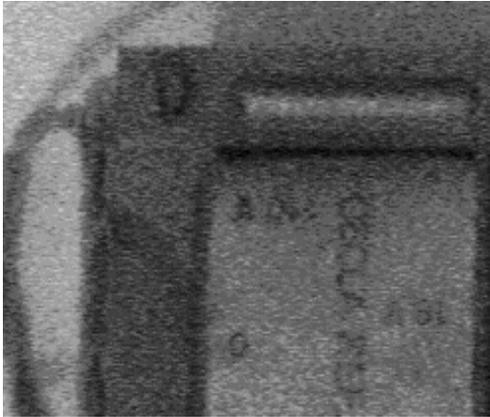
(zoom)



(... interpolation)

■ Example Mean Filter → noise reduction

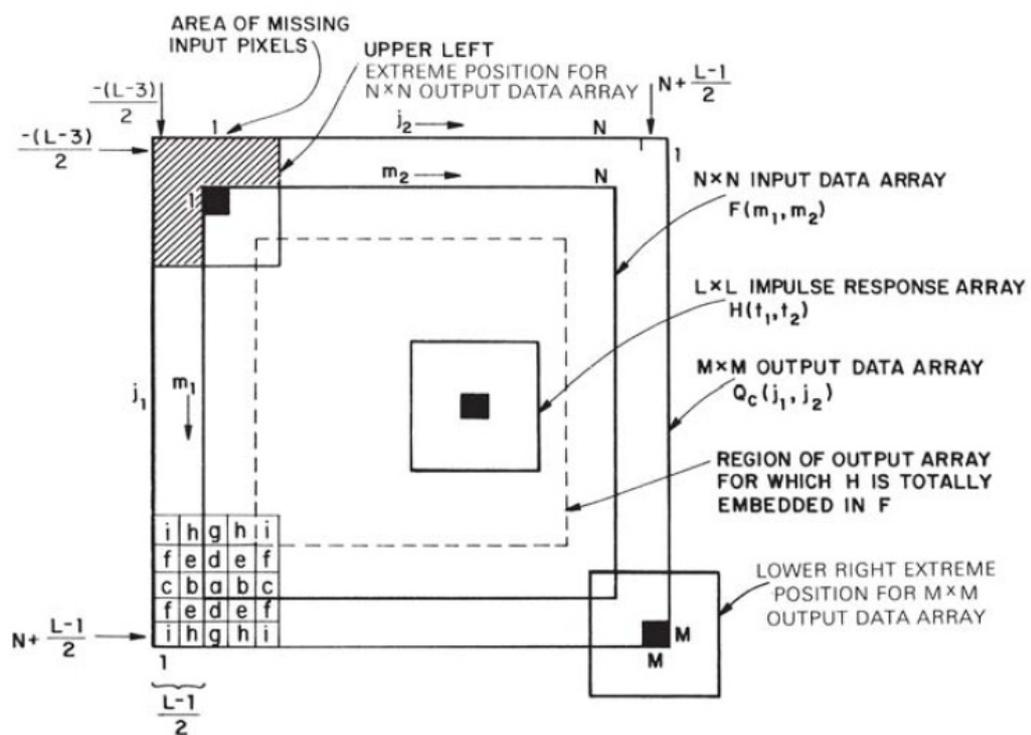
Mean Filter 3x3 ($k=-1,0,1 \quad l=-1,0,1$),
Constant value $h(k,l)=1/9$



→ Problems?

Operations on images

■ Domain of convolution



Operations on images

- Geometric spatial transformations and image registration
 - Geometric transformations modify the spatial relationship between pixels in an image
 - In terms of digital image processing, a geometric transformation consists of
 - A **spatial transformation** of coordinates $(x', y') = T\{(x, y)\}$
 - **Intensity interpolation** that assigns values to the spatially transformed pixel

■ Spatial transformations

□ Example

- Shrink image to half its size $(x', y') = T\{(x, y)\} = (x/2, y/2)$

□ Affine transform:

$$[x', y', 1] = [x, y, 1] \cdot \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

□ Higher order

$$[x', y', 1] = [x, y, x^2, y^2, xy, \dots, 1] \cdot \mathbf{T}$$

...

□ Affine transform:

identity $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{cases} x' = x \\ y' = y \end{cases}$

scaling $\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{cases} x' = c_x \cdot x \\ y' = c_y \cdot y \end{cases}$

translation $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \Rightarrow \begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$

Shear (vertical) $\begin{bmatrix} 1 & 0 & 0 \\ s_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{cases} x' = x + s_x \cdot y \\ y' = y \end{cases}$

Shear (horizontal) $\begin{bmatrix} 1 & s_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{cases} x' = x \\ y' = y + s_y \cdot x \end{cases}$

identity



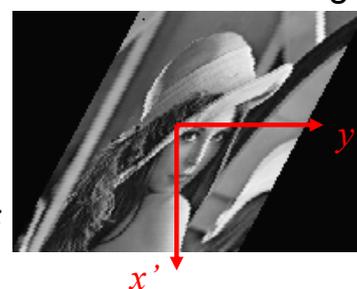
scaling



translation



horizontal shearing



$$\begin{cases} x' = x \\ y' = y + 0.5x \end{cases}$$

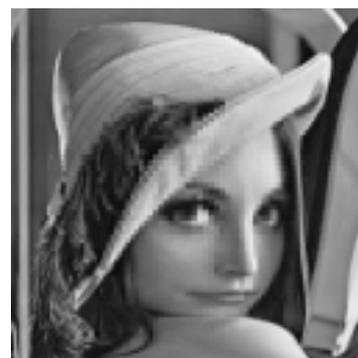
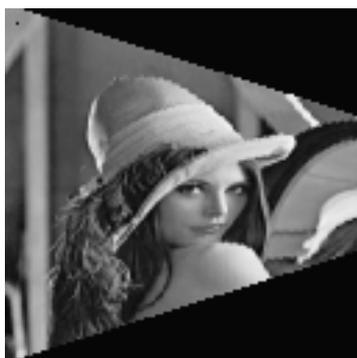
□ Affine transform, rotation

Rotation $T = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

35° degrees rotation

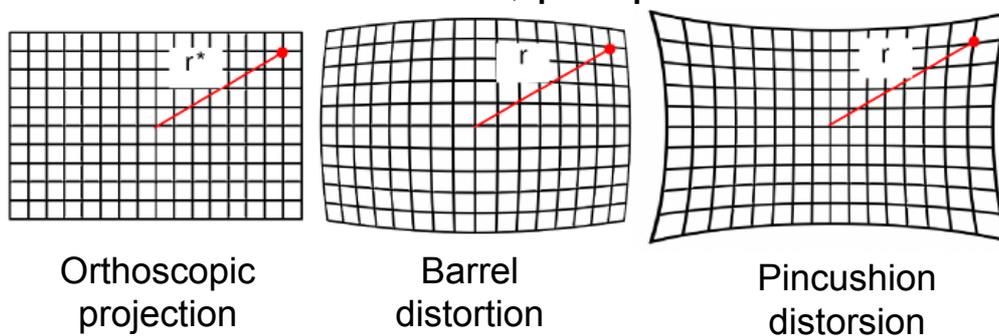


Higher order transforms



Applications :

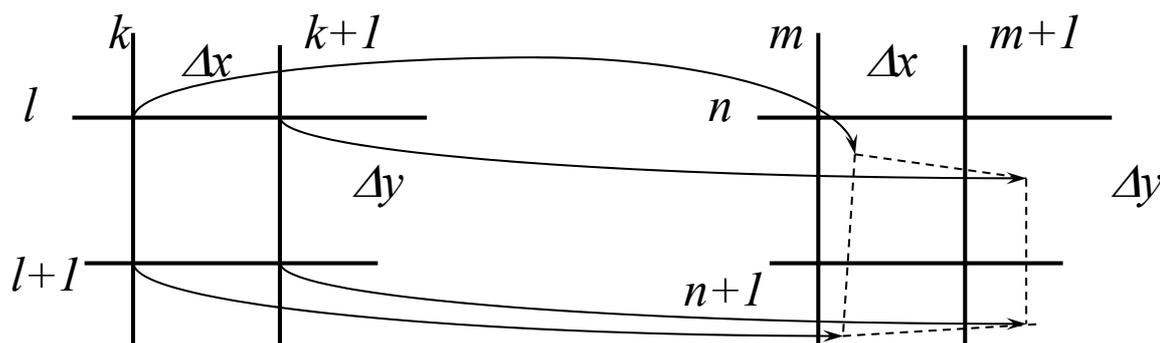
Lens distortion correction, perspective



Intensity interpolation

Problem:

x, y are discrete values (sampled image): $x = kD_x$, $y = lD_y$
 and $x' = h_1(kD_x, lD_y)$ et $y' = h_2(kD_x, lD_y)$ are not
 necessary multiple integer of D_x and D_y



□ Solution: intensity interpolation

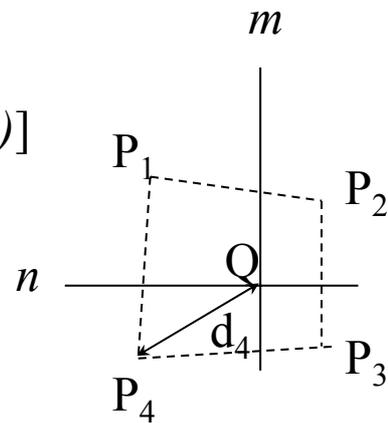
$$f'(Q) = f'(m\Delta x, n\Delta y) = G[f(P_1), f(P_2), f(P_3), f(P_4)]$$

With $f(P_1) = f(k\Delta x, l\Delta y)$

$$f(P_2) = f((k+1)\Delta x, l\Delta y)$$

$$f(P_3) = f((k+1)\Delta x, (l+1)\Delta y)$$

$$f(P_4) = f(k\Delta x, (l+1)\Delta y)$$



- Nearest neighbor: $f(Q) = f(P_k)$, $k : d_k = \min\{d_1, d_2, d_3, d_4\}$
- linear interpolation

$$f(Q) = \frac{\sum_{k=1}^4 f(P_k) / d_k}{\sum_{k=1}^4 1 / d_k}$$

- Bilinear interpolation, ideal interpolation, spline interpolation,....

□ Interpolation, example with rotation

$$T = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

35° degrees rotation

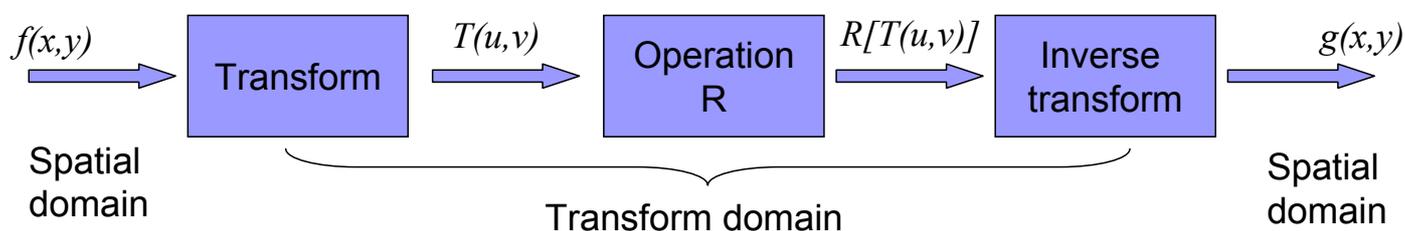
Without interpolation

With interpolation

Département

Images Transformation

- Previous methods work in spatial domain
 - In some cases, image processing tasks are best formulated in a transform domain.
 - i.e. frequency \rightarrow Fourier
- \rightarrow Many transformations exist



Images Transformation

- A particularly important class of 2D linear transforms can be expressed in the general form

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot r(x, y; u, v)$$

forward transform Input image Forward transformation kernel

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) \cdot s(x, y; u, v)$$

Recovered image Forward transform inverse transformation kernel

Separable kernel: $r(x, y, u, v) = r_1(x, u) \cdot r_2(y, v)$

Symmetric kernel: $r(x, y, u, v) = r_1(x, u) \cdot r_1(y, v)$

Images Transformation, DFT

■ 2-D Discrete Fourier Transform (DFT)

Forward kernel $r(x, y, u, v) = e^{-j2\pi(ux/M+vy/N)}$

Inverse kernel $s(x, y, u, v) = \frac{1}{MN} e^{j2\pi(ux/M+vy/N)}$

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-2j\pi(ux/M+vy/N)}$$

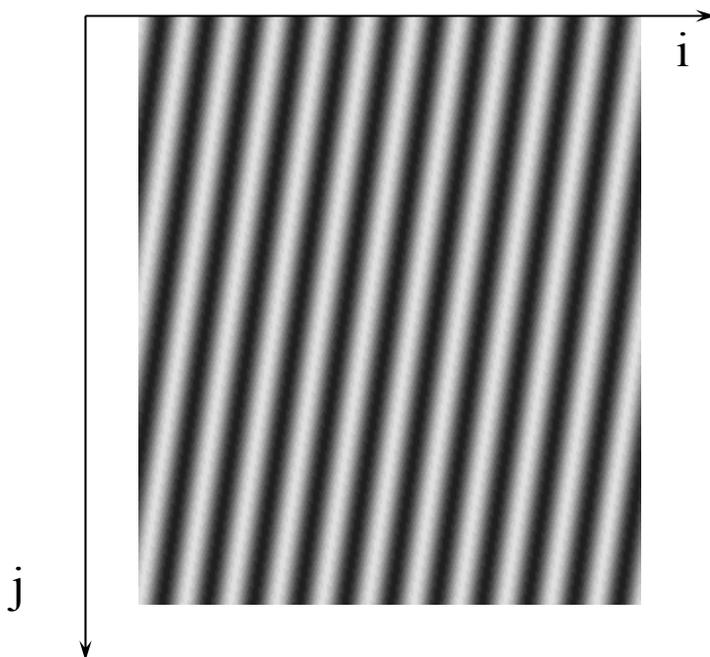
$$f(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} T(u, v) \cdot e^{2j\pi(ux/M+vy/N)}$$

→ Complex numbers...

→ Modulus and phase (angle)

→ coefficients...

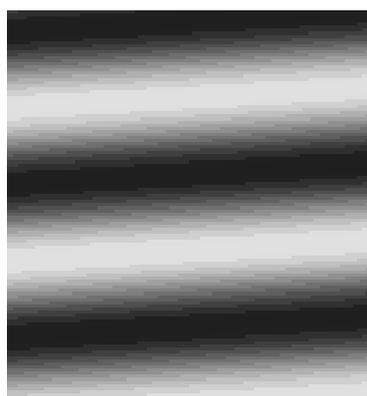
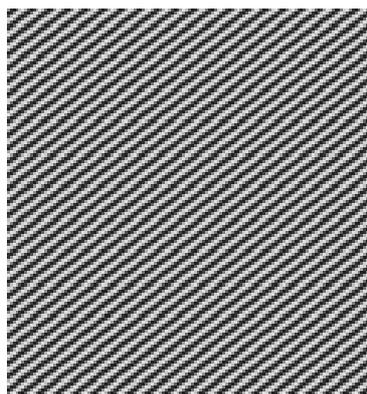
■ Example: 2-D Fourier transform



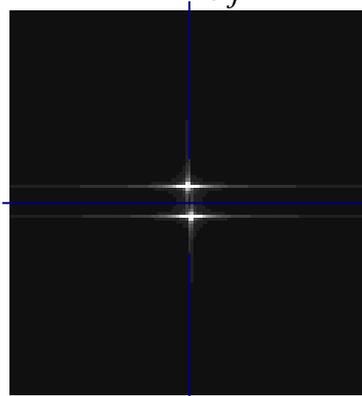
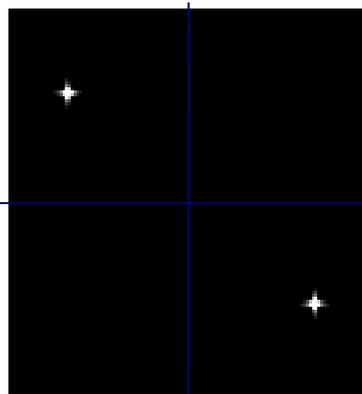
Speed (frequency) of
sinusoïdal variation of pixel
intensity in a given direction

$$f(i, j) = A \sin(2\pi f_i i + 2\pi f_j j + \varphi_i + \varphi_j)$$

Sine images



2 Dirac delta functions



High frequency
 f_i

f_j

Low frequency
 f_i

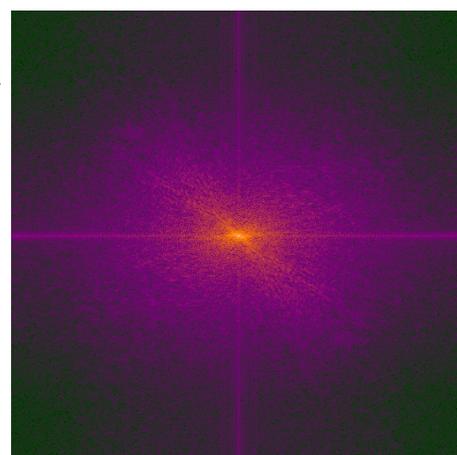
f_j

Image example: 2-D Fourier transform

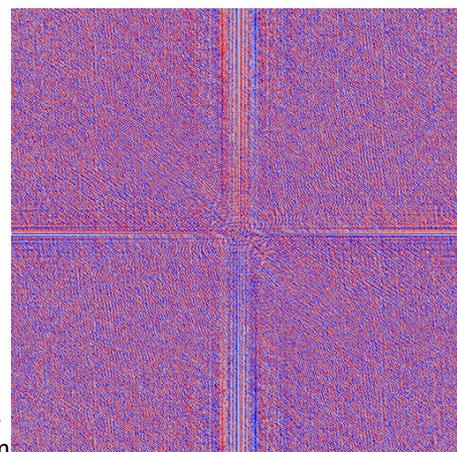


DFT

DFT⁻¹

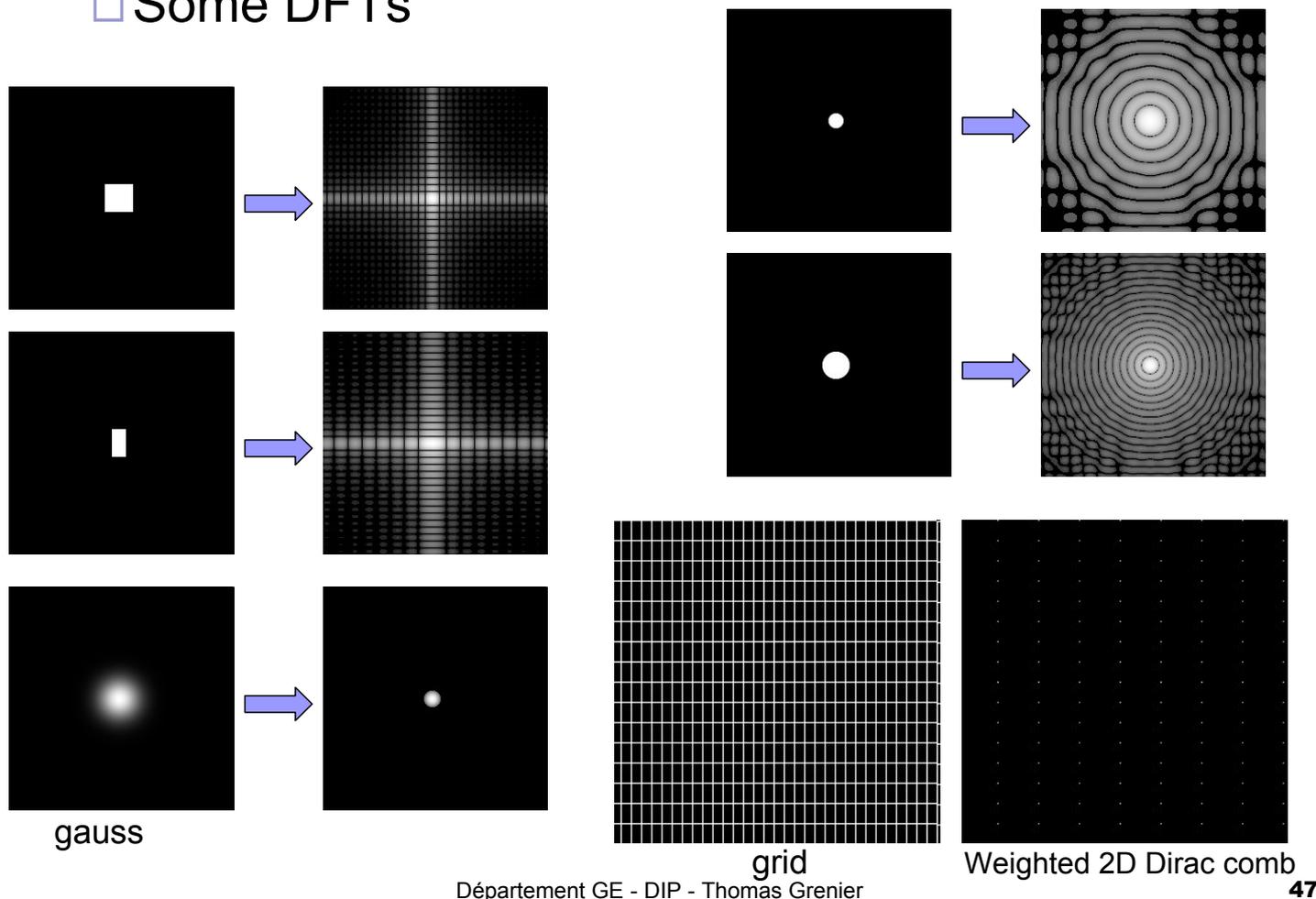


modulus



phase

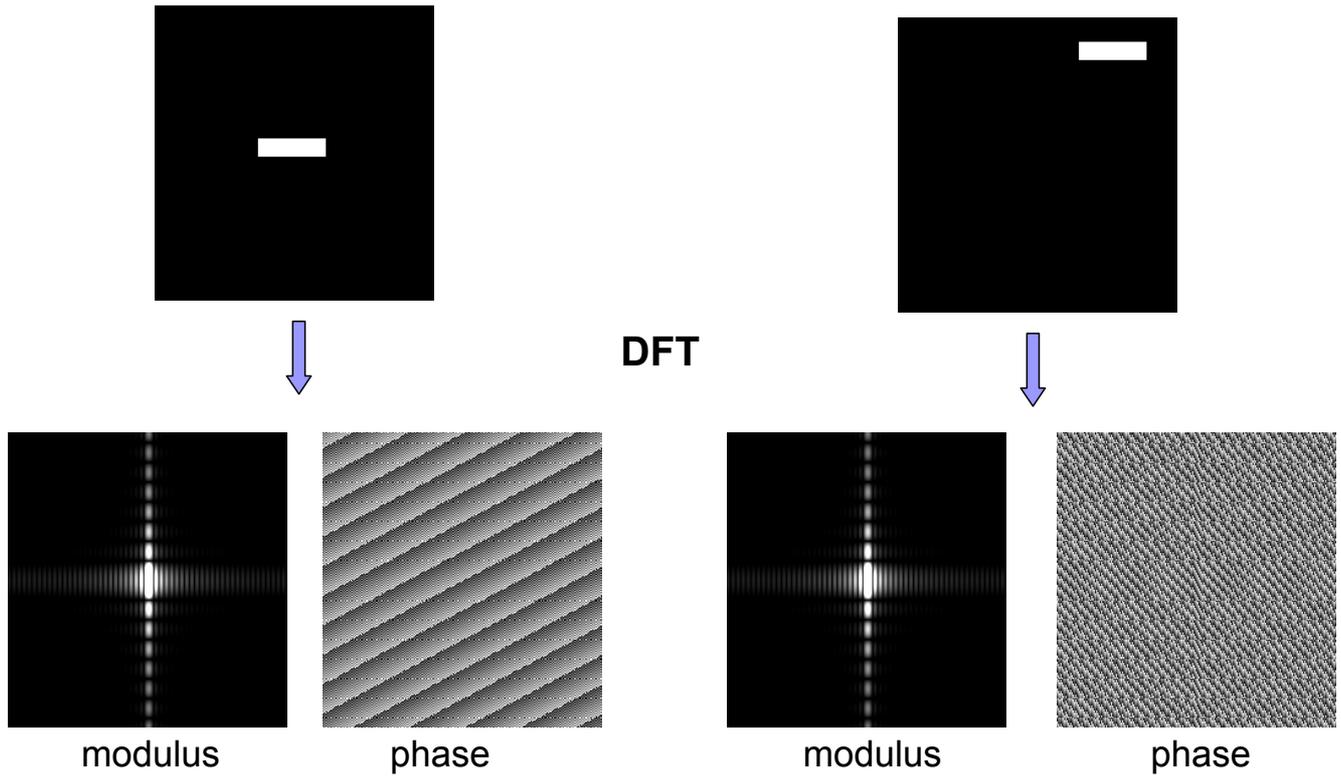
□ Some DFTs



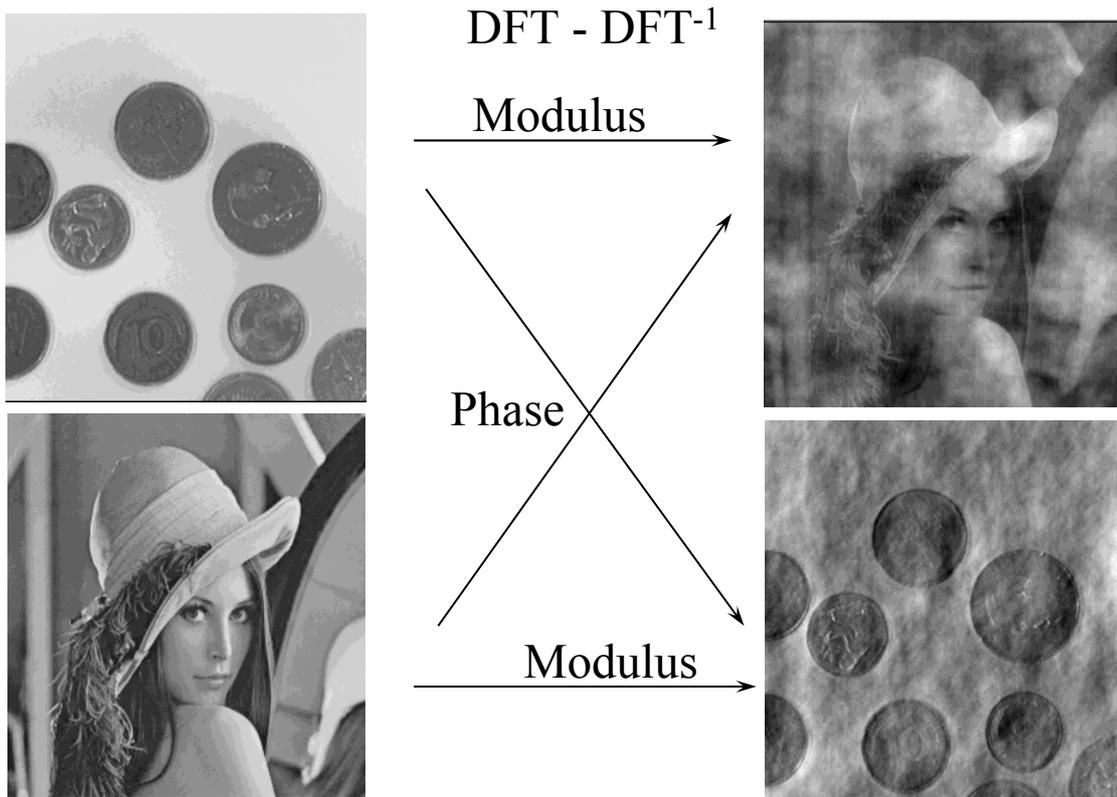
□ 2-D DFT and DFT⁻¹ properties

- The same as in the 1D case
- Periodic in the u and v directions (period = M, N)
- $F(0, 0)$ = dc component = mean of grey-levels
- Energy conservation $\rightarrow \sum \sum |f(x, y)|^2 = \sum \sum |F(u, v)|^2$
- $f(x, y)$ real $\rightarrow F(u, v)$ is conjugate symmetric $F^*(u, v) = F(-u, -v)$
 - (and Real part is even, Imaginary part is odd)
- Separable and symmetric kernel
- Fast algorithm (FFT, many forms) : $N^2 \cdot \log_2(N)$
- Circular convolution (periodic extension of function) = DFT

□ Phase influence:



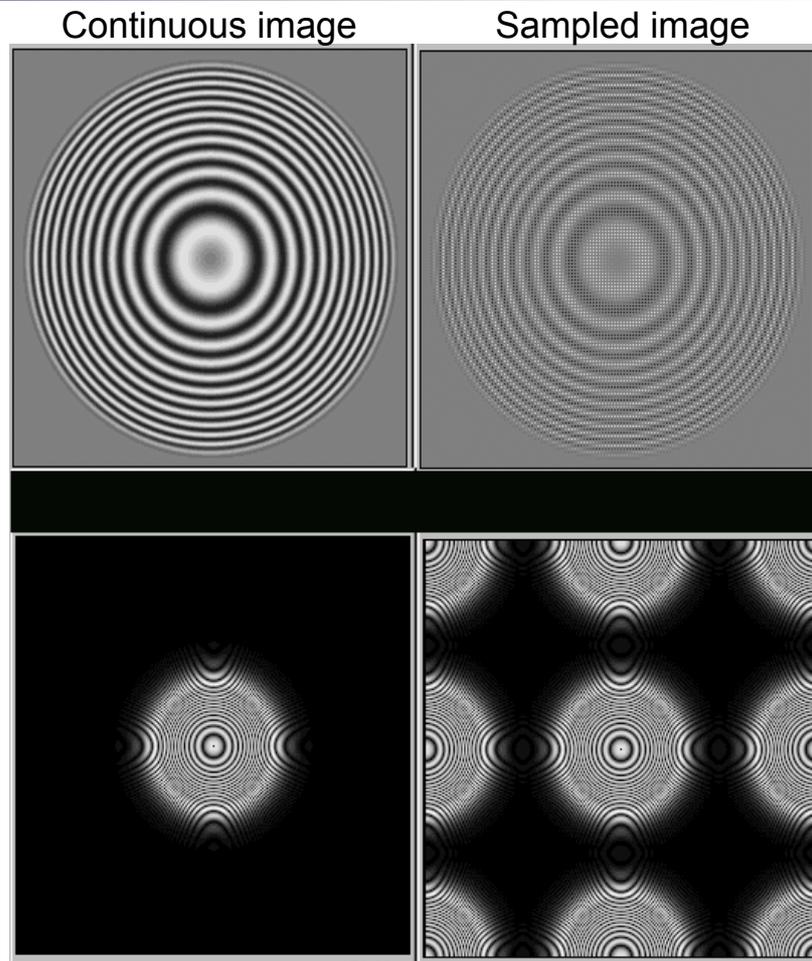
□ Phase influence:



□ Notes on aliasing...

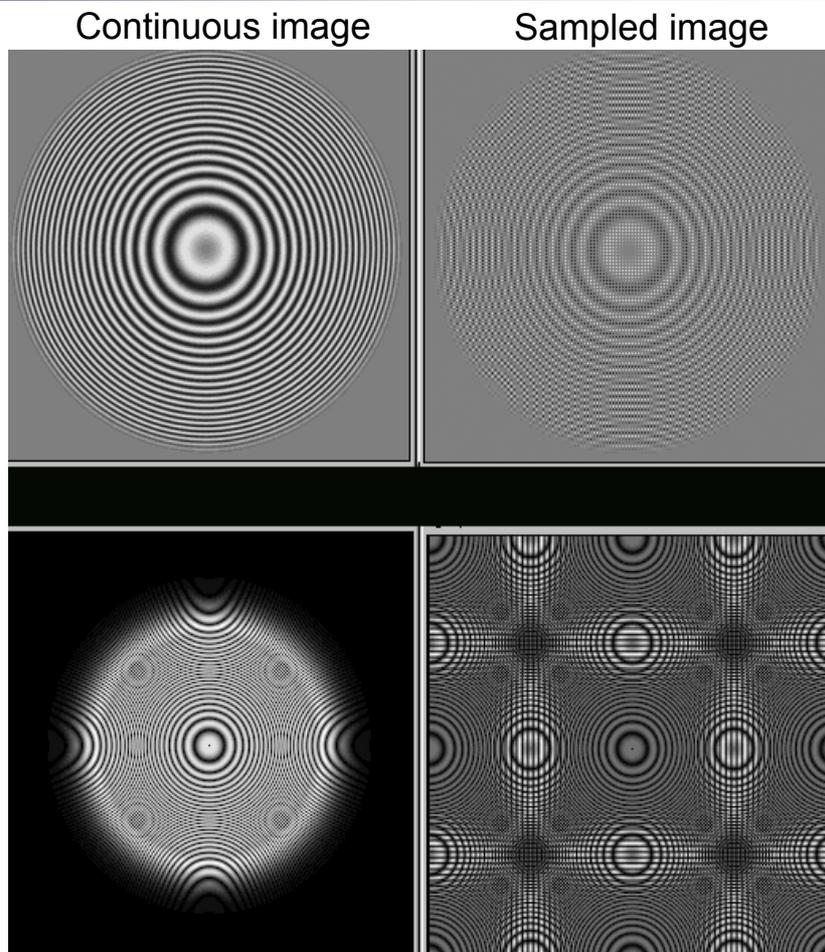
Without aliasing

Note
Periodic DFT



□ Notes on aliasing...

With aliasing



Images Transformation

■ Other transformations

□ Unitary transforms

■ Radon

- used to reconstruct images from medical computed tomography scans

■ Cosine (DCT)

- JPEG, MPEG (mDCT: AAC, Vorbis, WMA, MP3)

■ Sine

■ Wavelet

■ ...

□ Hough

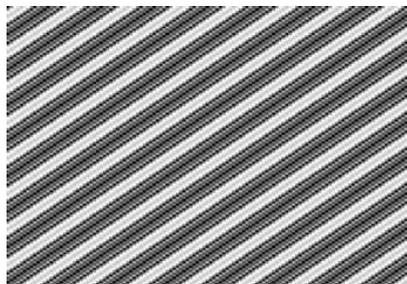
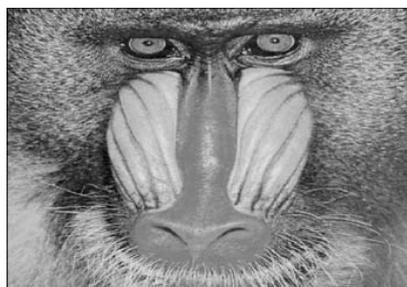
□ Discrete Cosine Transform

DCT-II

$$C(u, v) = \frac{4 \cdot c(u) \cdot c(v)}{M \cdot N} \cdot \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} x(i, j) \cdot \cos\left(\frac{(2i+1)u\pi}{2N}\right) \cdot \cos\left(\frac{(2j+1)v\pi}{2M}\right)$$

$$\text{with } c(m) = \begin{cases} \sqrt{2/N} & \text{if } m \neq 0 \\ \sqrt{1/N} & \text{if } m = 0 \end{cases}$$

Examples



Sine wave

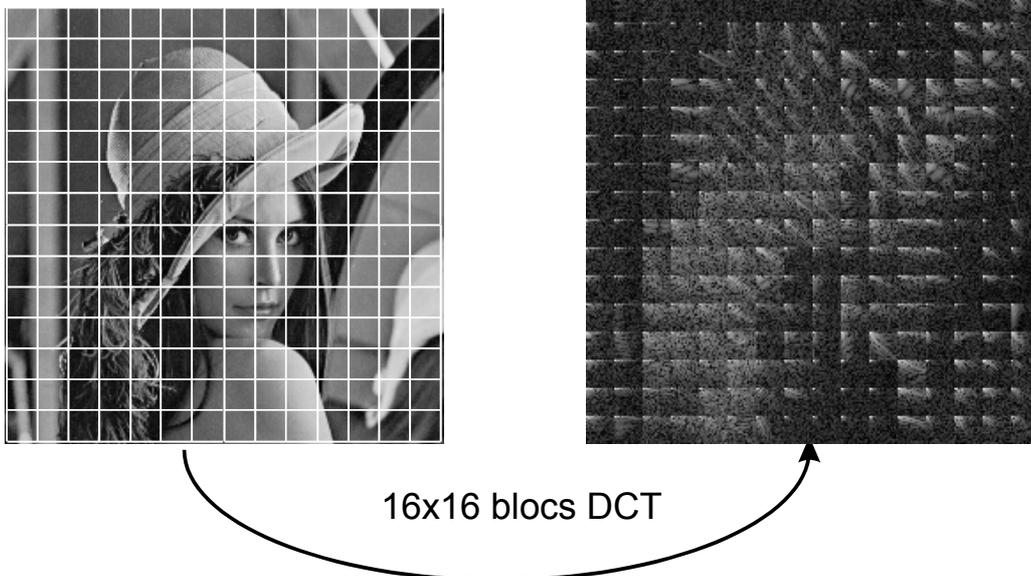


□ Discrete Cosine Transform, properties

- Linear, separable
- **real coefficients**
- $C(0,0)$ = dc component = mean of grey-levels
- DCT concentrates most of the power on the lower frequencies
- Fast algorithms (like FFT) : $N^2 \cdot \log_2(N)$

→ Image compression (JPEG, MPEG)

Note on jpeg: DCT per bloc



□ Hough Transform

➔ Features extraction technique

- The classical Hough transform was concerned with the identification of lines in the image
- The Hough transform has been extended to identify positions of arbitrary shapes
 - Circles
 - Ellipses

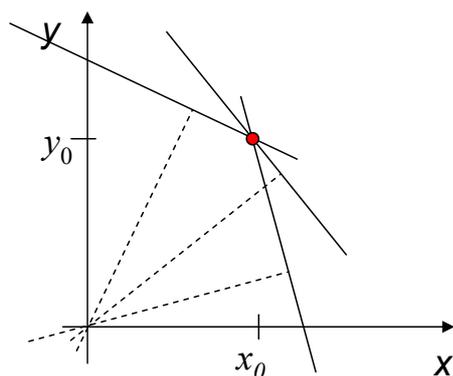
□ Hough Transform

Equation of a line

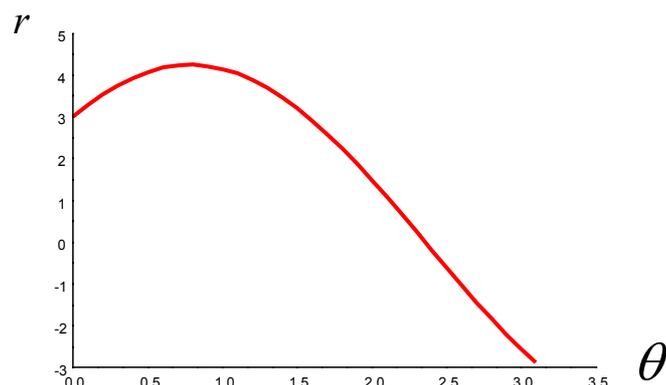
$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \left(\frac{r}{\sin \theta} \right) \longrightarrow r = x \cdot \cos \theta + y \cdot \sin \theta$$

Distance to origin
Angle of the vector from the origin to the closest point of the line

➔ For a point with (x_0, y_0) coordinates in the image plane, all the lines that go through it verify : $r(\theta) = x_0 \cdot \cos \theta + y_0 \cdot \sin \theta$

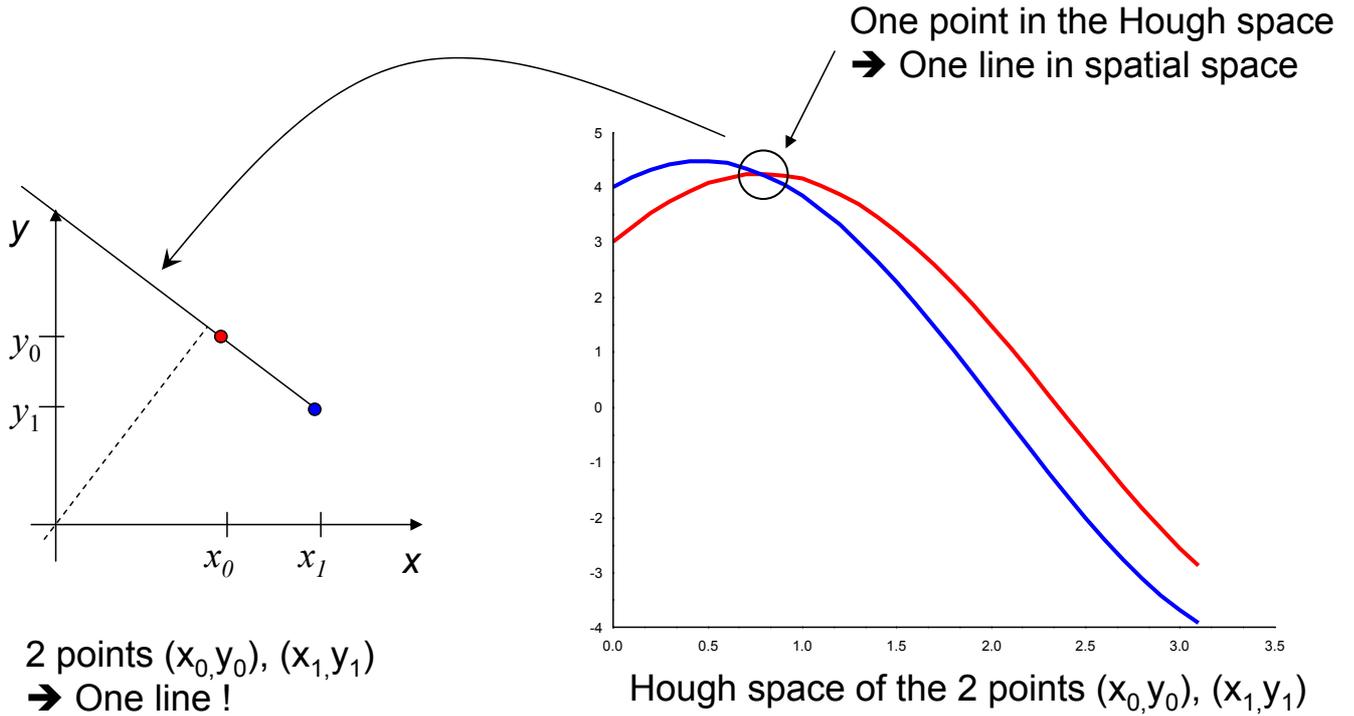


3 lines through (x_0, y_0)



All the lines through (x_0, y_0)

□ Hough Transform



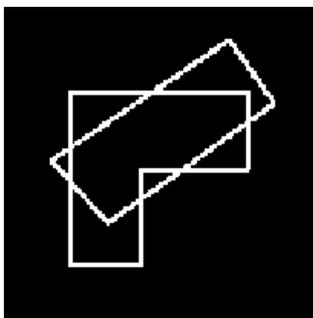
2 points $(x_0, y_0), (x_1, y_1)$
 → One line !

→ $(x_0, y_0) = (3, 3); (x_1, y_1) = (4, 2);$

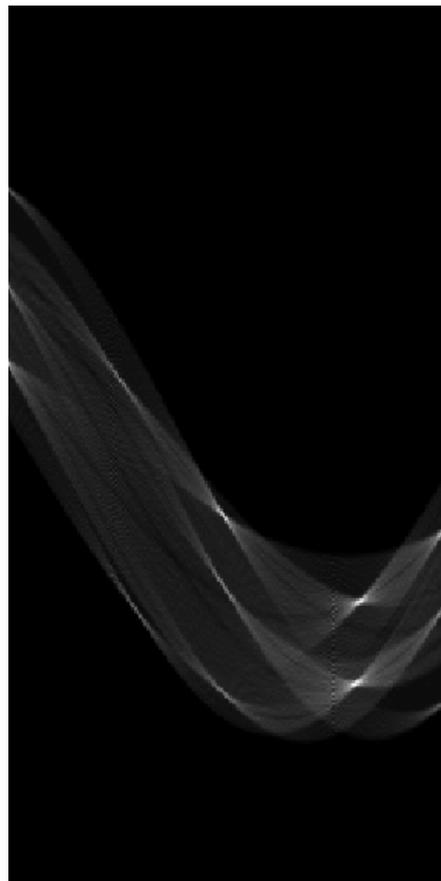
→ $(r, \theta) \sim (4.25, 0.785)$

□ Hough Transform

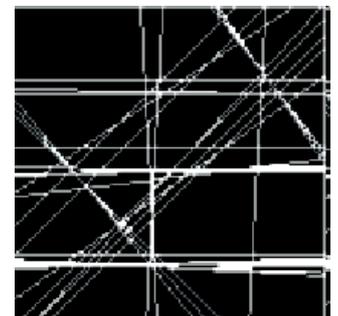
■ Application: line detection



Binary image



Hough's space



Projection of some lines

→ Implementation ?
 → Sinogram and Radon

Digital Image Processing

Image Improvement

*Département Génie Electrique
5GE - TdSi*

Summary

I. Introduction

- DIP , Examples, Fundamental steps, components

II. Digital Image Fundamentals

- Visual perception, light
- Image sensing, acquisition, sampling, quantization
- Linear, and non linear operation

III. Discrete 2D Processing

- Vector space, Convolution
- Unitary Transformation

IV. Image Improvement

- Enhancement, restoration, geometrical modifications

Image Improvement

- Image improvement denotes three types of image manipulation processes:
 - Image enhancement entails operations that improve the appearance to a human viewer, or operations to convert an image to a format better suited to machine processing
 - Image restoration has commonly been defined as the modification of an observed image in order to compensate for defects in the imaging system that produced the observed image
 - Geometrical image modification includes image magnification, minification, rotation and nonlinear spatial warping

Image Improvement

- Image enhancement
 - Contrast and histogram
 - Noise cleaning
 - Edge enhancement
 - *Color/multispectral image enhancement*
- Image restoration
- Geometrical image modification

Image Enhancement

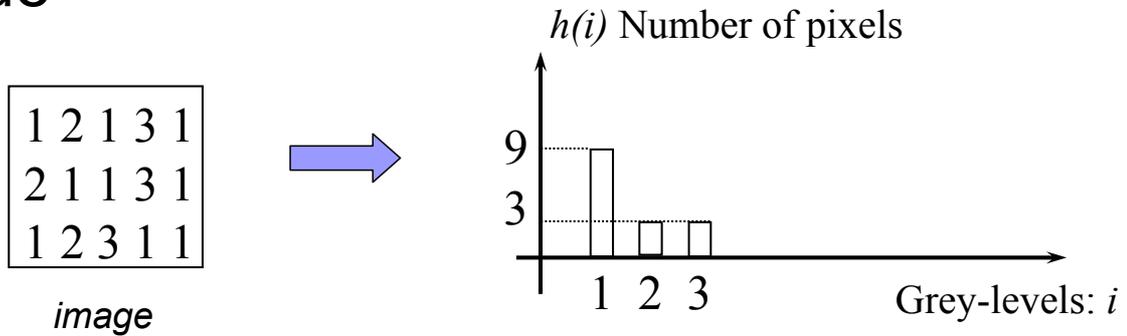
- Improve the visual appearance of an image or to convert the image to a form better suited for analysis by a human or a machine
- A lot of techniques exist
- There is no general unifying theory of image enhancement at present because there is no general standard of image quality that can serve as a design criterion for an image enhancement processor

Contrast improvement

- The most common defects of photographic or electronic images is poor contrast resulting from a reduced, and perhaps nonlinear, image amplitude range
- Image contrast can often be improved by amplitude rescaling of each pixel
 - Histogram
 - Transformation functions

Histogram

- Number of pixels that have a given intensity value

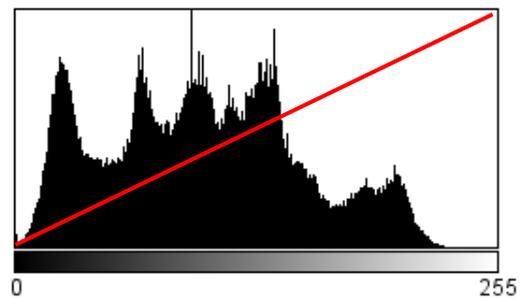


- Similar to the probability density function

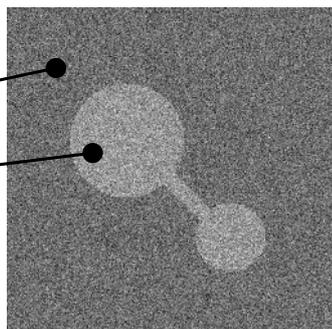
$$p(i) = h(i) / nb_of_pixels$$

Histogram

- Examples



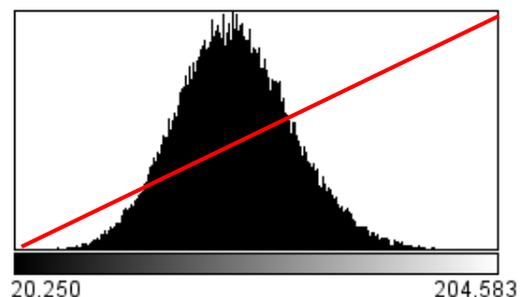
Count: 262144 Min: 0
 Mean: 99.434 Max: 243
 StdDev: 52.585 Mode: 93 (2760)



Mean = 100

Mean = 130

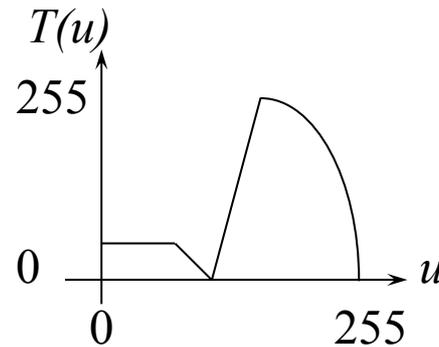
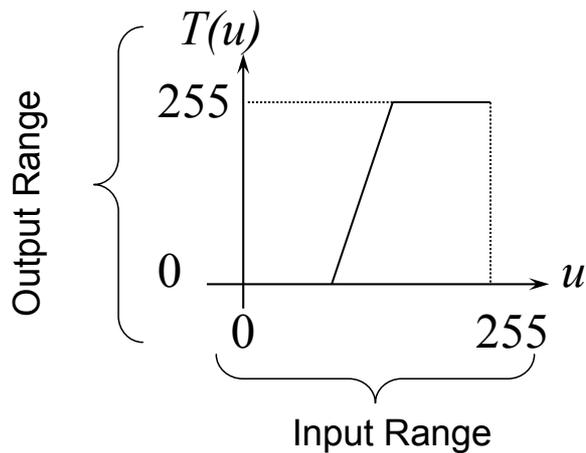
$\sigma = 20$



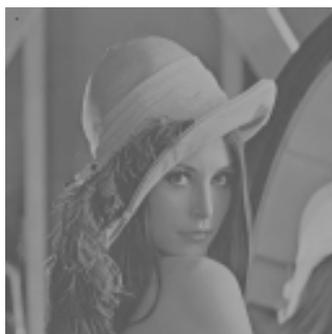
Count: 65536 Min: 20.250
 Mean: 104.219 Max: 204.583
 StdDev: 22.501 Mode: 103.416 (908)
 Bins: 256 Bin Width: 0.720

Histogram manipulation

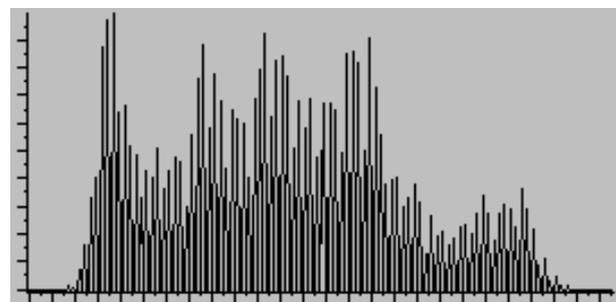
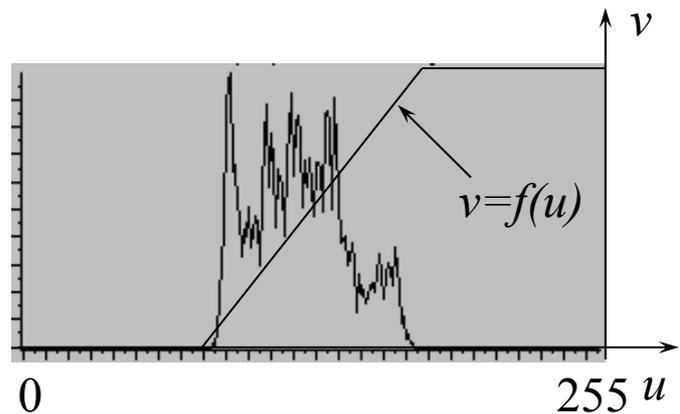
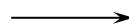
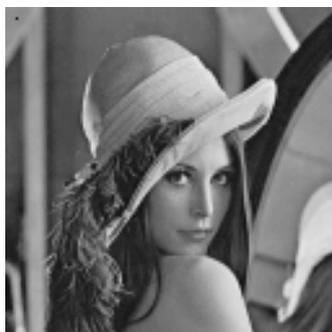
- Use a transformation function
 - Try an existing (classical) one
 - Build your own!



Histogram manipulation

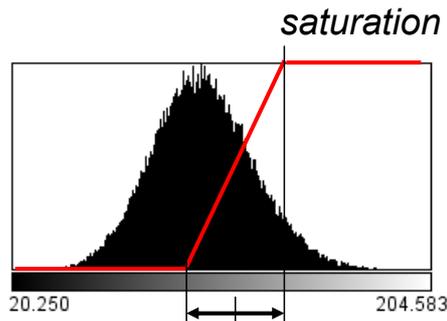
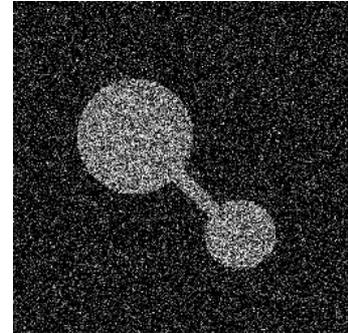
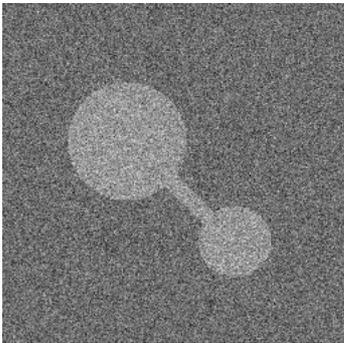


$$v=f(u)$$



Histogram manipulation

Brightness and contrast



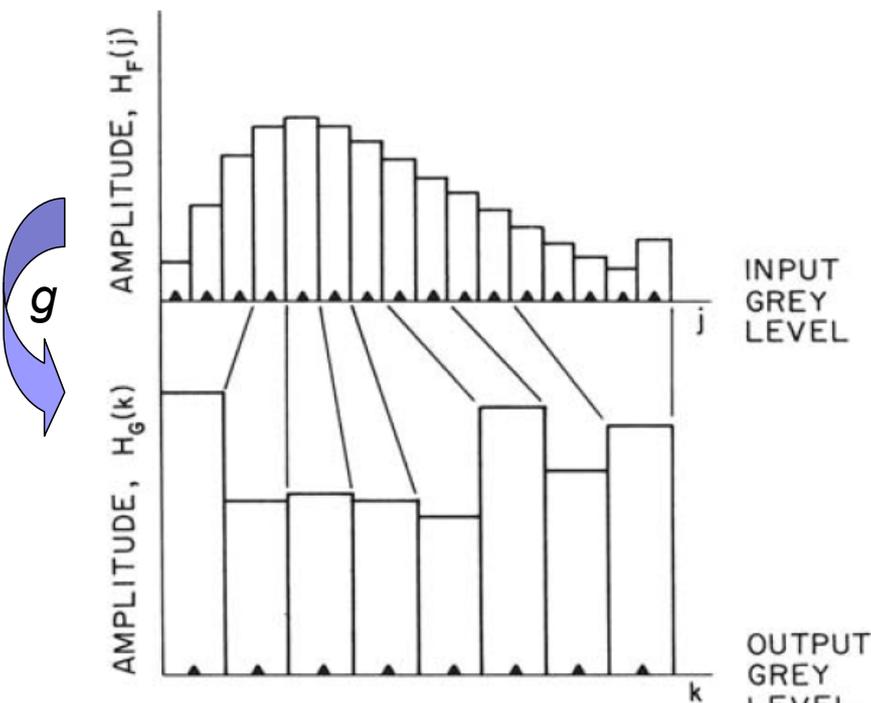
Brightness (level) \rightarrow middle value of the contrast window

Contrast (window) \rightarrow range of the contrast window

\rightarrow Demo
 \rightarrow Best choice ?

Histogram manipulation

Grey level histogram equalization



'g' function ?

$$\int_{g_{\min}}^g p_g(g).dg = \int_{f_{\min}}^f p_f(f).df$$

$$\int_{g_{\min}}^g p_g(g).dg = P_f(f)$$

Histogram manipulation

■ Histogram equalization

$$\int_{g_{\min}}^g p_g(g).dg = P_f(f)$$

□ Examples

- The output density is forced to be the uniform density

$$p_g(g) = \frac{1}{g_{\max} - g_{\min}} \quad g_{\min} \leq g \leq g_{\max}$$

$$g = (g_{\max} - g_{\min})P_f(f) + g_{\min}$$

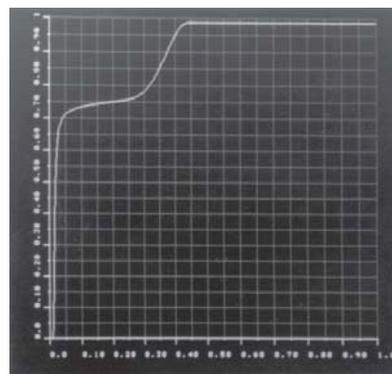
- Other functions for the output density (exponential, logarithmic)

$$p_g(g) = \alpha \exp\{-\alpha(g - g_{\min})\} \quad g \leq g_{\min} \quad \longrightarrow \quad g = g_{\min} - \frac{1}{\alpha} \ln\{1 - P_f(f)\}$$

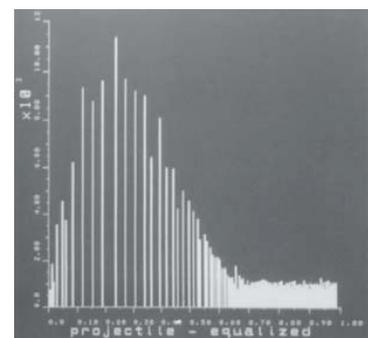
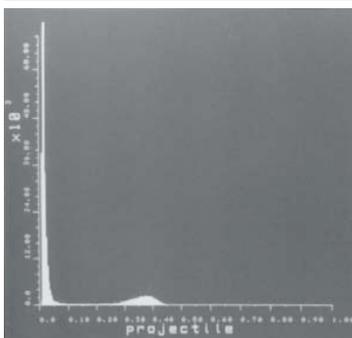
$$p_g(g) = \frac{1}{g[\ln\{g_{\max}\} - \ln\{g_{\min}\}]} \quad \longrightarrow \quad g = g_{\min} \left(\frac{g_{\max}}{g_{\min}}\right)^{P_f(f)}$$

Histogram manipulation

■ Histogram equalization, example



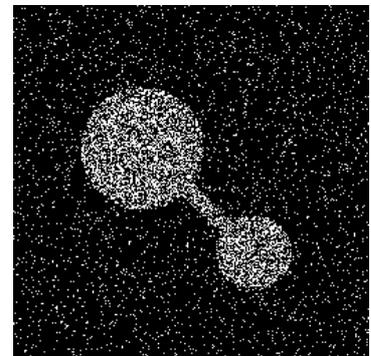
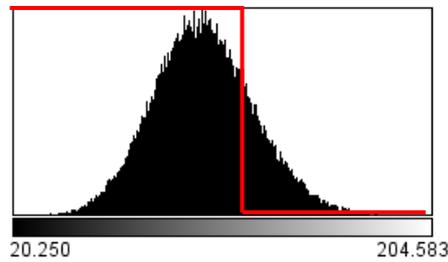
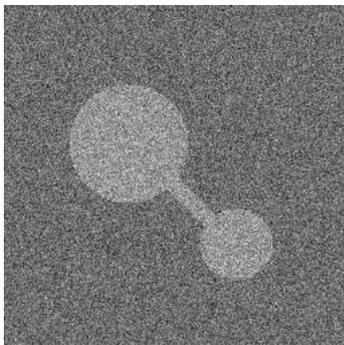
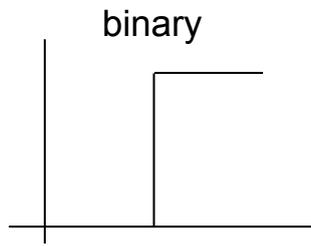
Transfer function
($P_f(f)$)



X-ray projectile image
and histogram

Histogram manipulation

■ Threshold

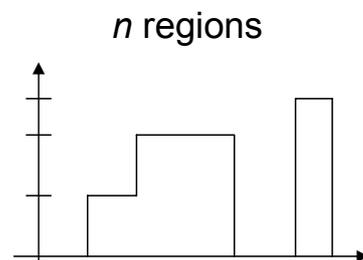
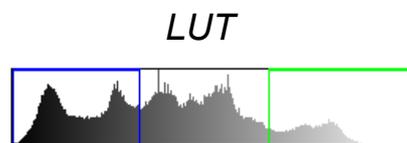


Département GE - DIP - Thomas Grenier

15

Histogram manipulation

■ Threshold



Département GE - DIP - Thomas Grenier

16

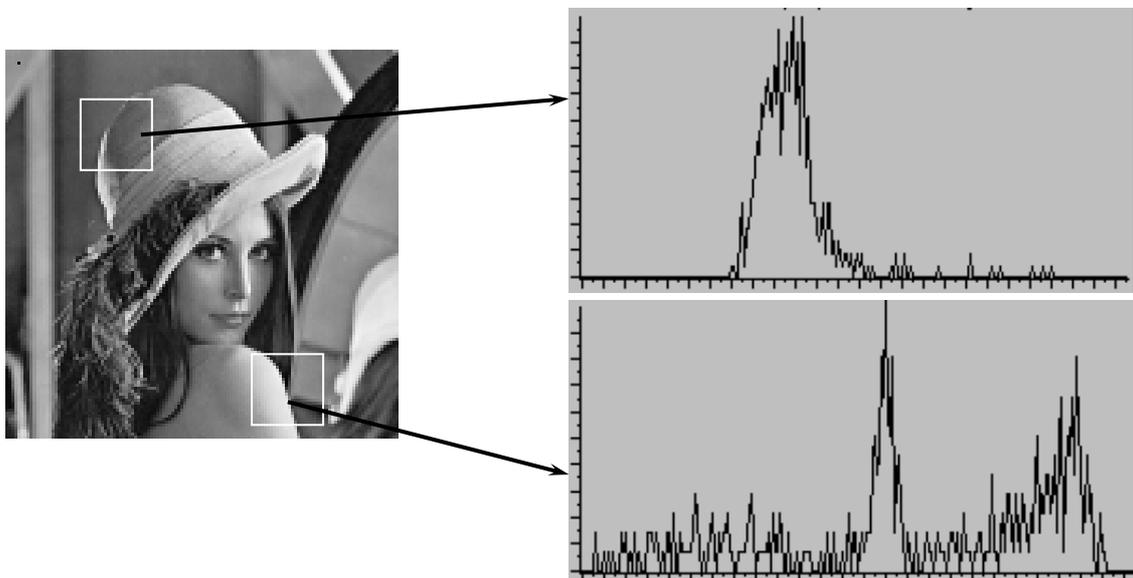
Histogram manipulation

■ Limitations

- Histogram equalization is not well adapted for good quality images
- Histogram threshold is not a noise removing technique!
- Histogram equalization should be adaptive!
 - Some methods exist (local equalization)

Local Histogram analysis

- features measured on the smallest neighborhood (1 pixel) grey-level (NG), color, quantitative value (Bq/cc, ...) ...
- features measured on a neighborhood → local histogram



Common computed values from the density probability function $p(x)$ (based on histogram, local or not)

- Moments

$$m_i = E[x^i] = \sum_{n=0}^{N-1} x^i p(x)$$

N: number of grey levels

- Centered Moments

$$\hat{m}_i = E[(x - E[x])^i] = \sum_{n=0}^{N-1} (x - m_1)^i p(x)$$

- Entropy

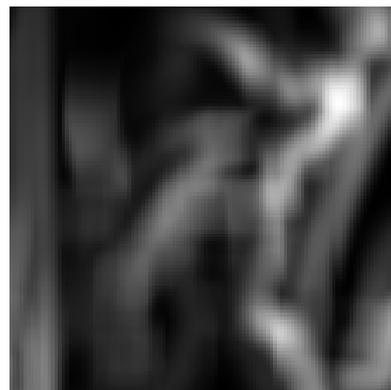
$$H = - \sum_{x=0}^{N-1} p(x) \log_2(p(x))$$

- Absolute moments, median value, max/min value, mode, percentiles, invariant moments (Legendre, ...) ...

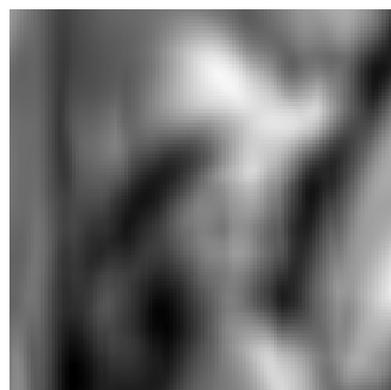


128x128 pixels

Computed on a 16x16 neighborhood



Variance



Mean

Noise cleaning

- Mean filter (linear filtering)
- Median filter (nonlinear filtering)
- Frequency domain filter (LP, HP, band-rejection (*notch*), ...)
- ...

Noise cleaning

- Many types of noise...



Gaussian Noise
(sd 25)



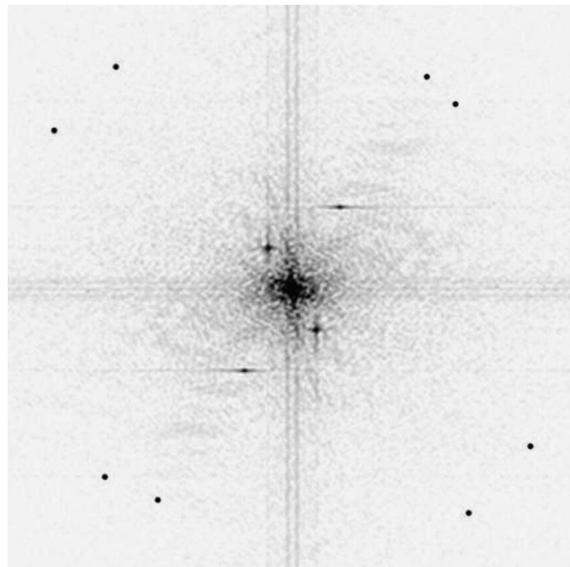
Salt and pepper noise

Noise cleaning

■ Periodic and quasi periodic noise



Quasi periodic noise



Fourier amplitude spectrum

Linear filtering

■ Mean filter

$$g(i, j) = \sum_{(k, l) \in W} h(k, l) f(i - k, j - l)$$

W: 25 neighbors

$$H = (1/25) \cdot I$$

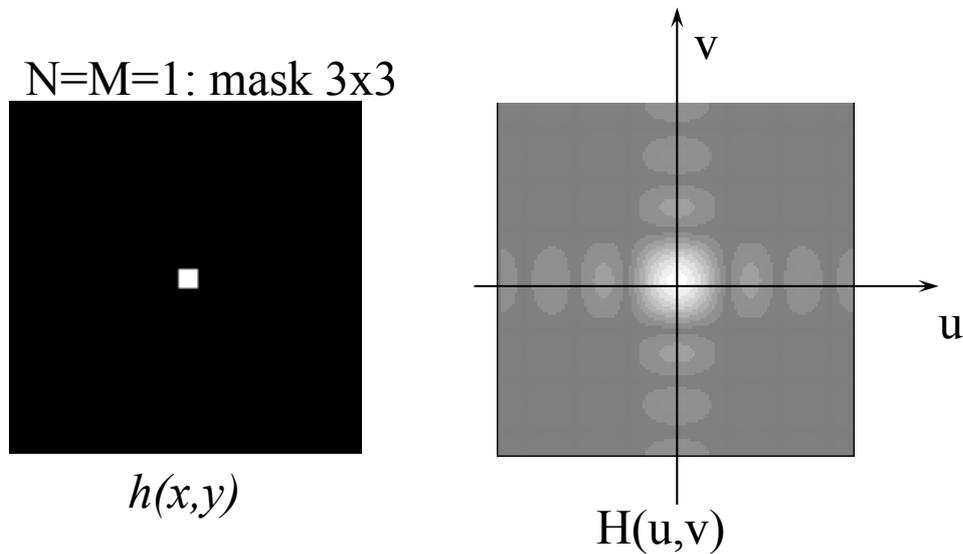
Result on Gaussian noise

Result on Salt & Pepper noise



Linear filtering

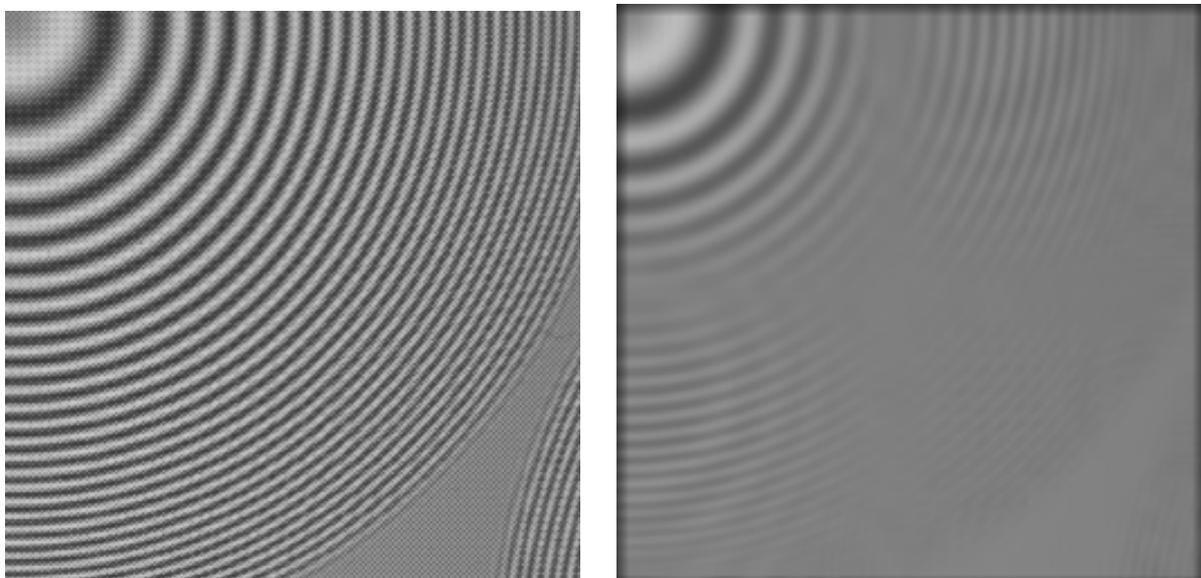
- Mean filter and DFT



→ *Anisotropic low pass filter with poor selectivity*

Linear filtering

- Mean filter anisotropy, (mask 11x11)



Linear filtering

■ Gaussian Filter

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Filter with infinite impulse response! → Approximate the ideal filter by truncating and windowing the infinite impulse response to make a FIR



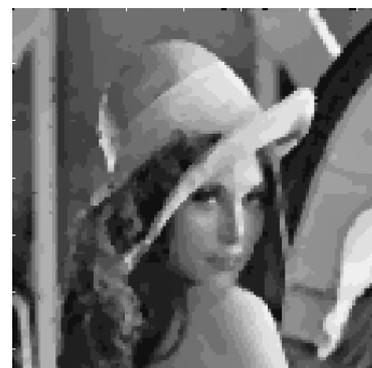
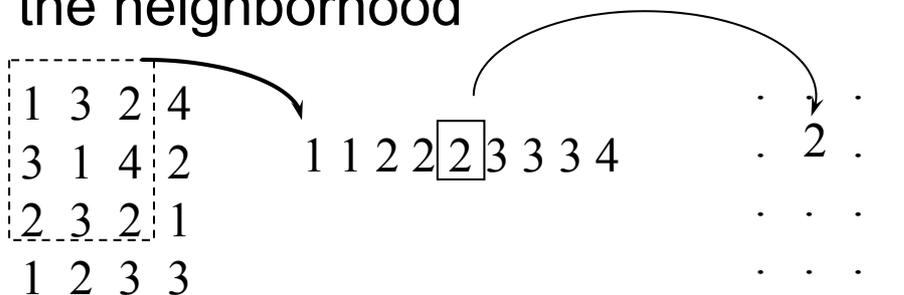
Isotropic low pass filter with poor selectivity
 $H(u,v)$ is a gaussian

- Many other types of filter
 - high pass, low pass, band-stop, derivative...
 - Ideal, Butterworth,...

Non linear filtering

■ Median filter

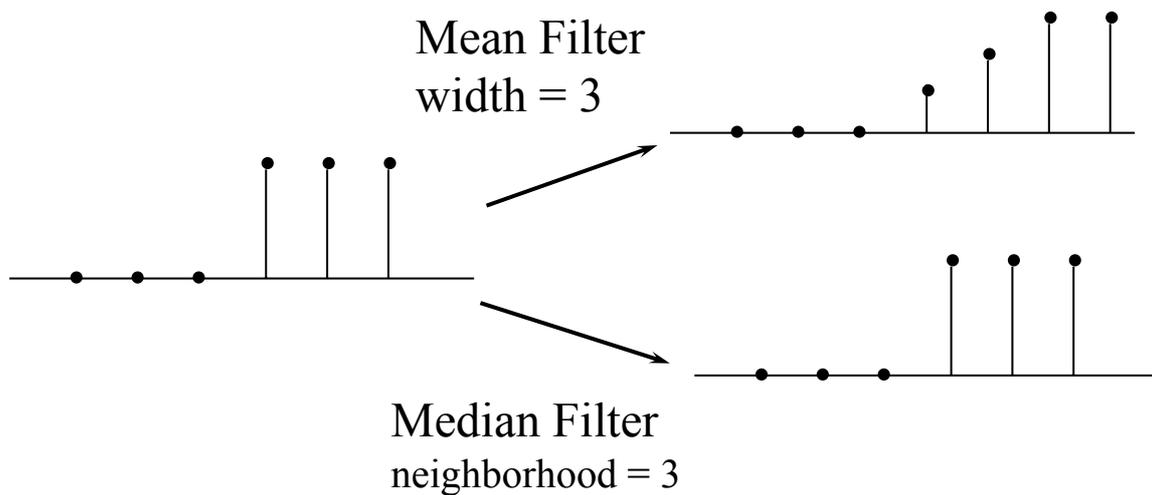
- Replace the central value by the median value of the neighborhood



Non linear filtering

■ Median filter

- Advantage of median filtering over linear filtering: edges are preserved



Non linear filtering

■ Median filter

Result on Gaussian noise

W=25



W=9

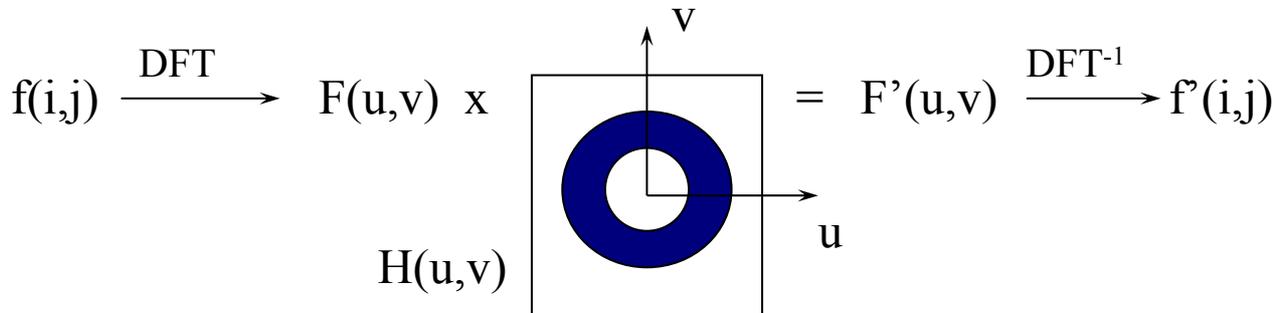
Result on salt & pepper noise



Frequency domain filtering

■ Approach

- Region selection in the frequency space (u,v)

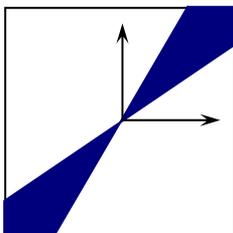


- to keep $f'(i,j)$ real, regions must be symmetric about the origin
- In frequency space (u,v), region boundaries can be
 - Steep (but ... oscillations can appear: Gibbs)
 - Smooth (but less selective)

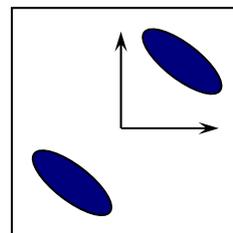
Frequency domain filtering

■ Approach

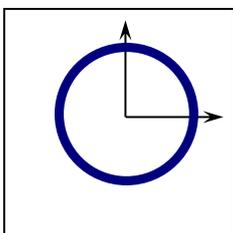
- Many forms for regions



Choice of one direction



Choice of the direction and frequency bands



Choice of frequency bands

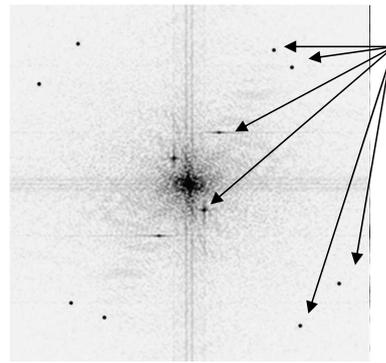


- Low-pass filters
- High-pass filters
- Band-pass filters
- Band-reject filters...

Frequency domain filtering

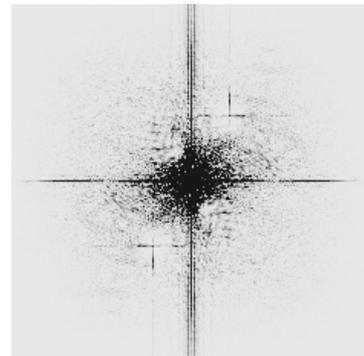
■ Notch filter

Periodic noise



Remove!

Clean dots and lines



Département GE - DIP - Thomas Grenier

33

Edge enhancement

- **Edges**: Changes or discontinuities of amplitude in an image
- Edges provide an indication of the physical extent of objects within the image → **Contours**

■ Edge detection

- Differential detection
- Model fitting

■ Edge enhancement filters

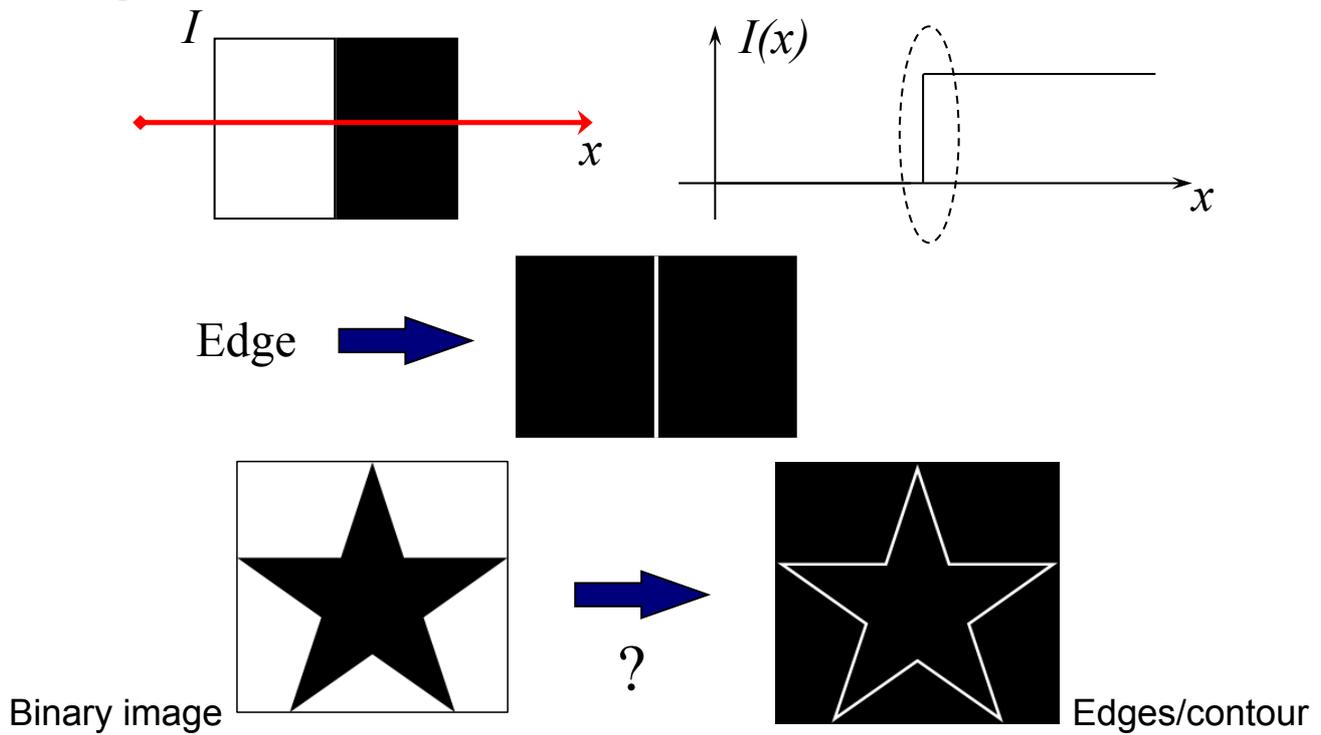


Département GE - DIP - Thomas Grenier

34

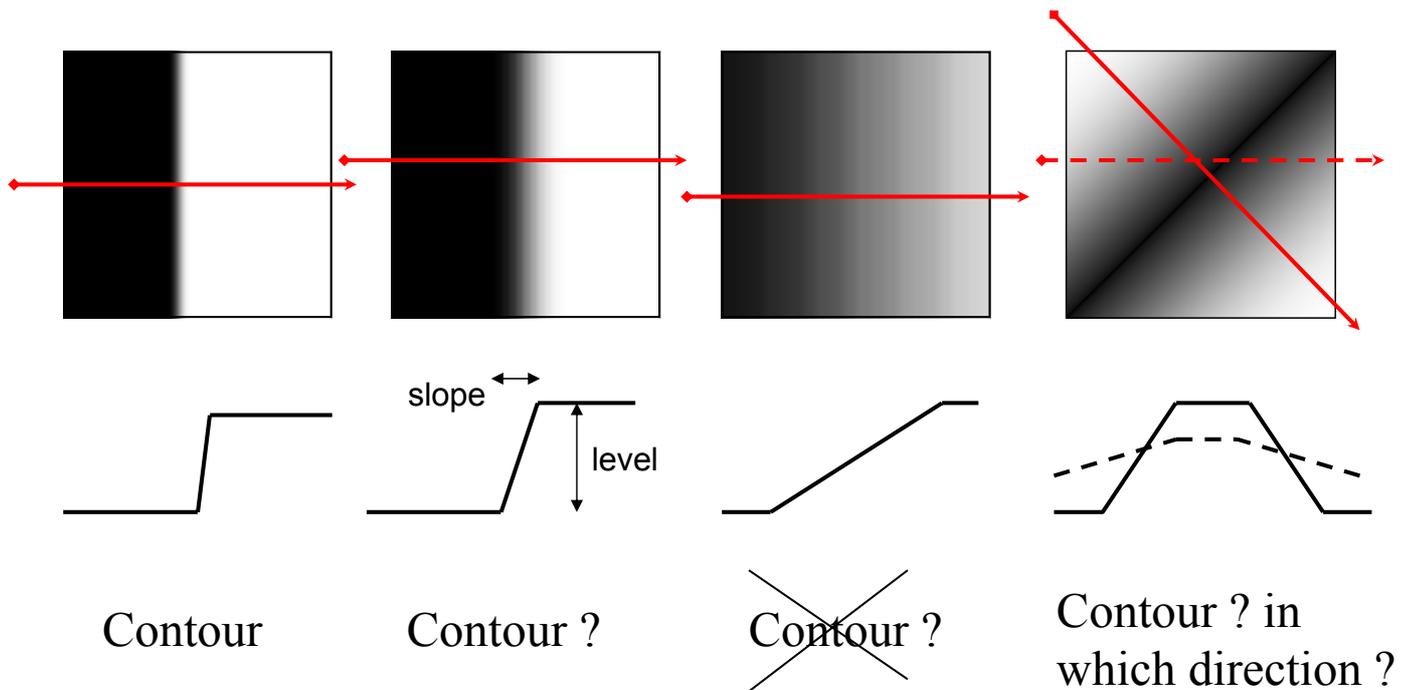
Edge enhancement

■ Edge detection



Département GE - DIP - Thomas Grenier

35



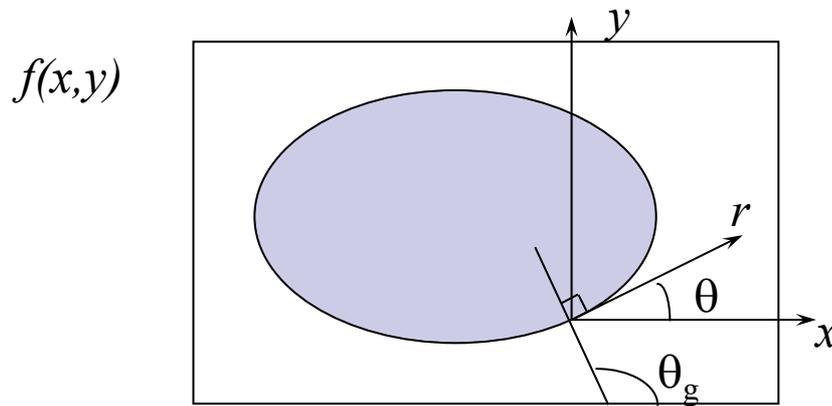
➔ What are a change or a discontinuity ?

➔ What about the direction (in image) ?

Département GE - DIP - Thomas Grenier

36

□ Definition of continuous contour and gradient



Continuous one-dimensional gradient $\mathbf{g}(x,y)$ of $f(x,y)$ along a line normal to the edge slope which is at an angle θ with respect to the horizontal axis:

$$\frac{\partial f(x,y)}{\partial r} \text{ max for } \theta_g = \theta + \frac{\pi}{2} \quad \Rightarrow \quad \frac{\partial}{\partial \theta} \left(\frac{\partial f(x,y)}{\partial r} \right) = 0$$

$$\mathbf{g}(x,y) = \nabla f(x,y) = \begin{Bmatrix} \partial f(x,y) / \partial x \\ \partial f(x,y) / \partial y \end{Bmatrix}$$

Département GE - DIP - Thomas Grenier

37

$$\frac{\partial f}{\partial r} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial r} = \frac{\partial f}{\partial x} \cos(\theta) + \frac{\partial f}{\partial y} \sin(\theta)$$

$$\frac{\partial}{\partial \theta} \left(\frac{\partial f(x,y)}{\partial r} \right) = 0 \quad \Rightarrow \quad -\frac{\partial f}{\partial x} \sin(\theta) + \frac{\partial f}{\partial y} \cos(\theta) = 0$$

Direction

$$\theta_g = \arctan \left(\frac{\partial f / \partial y}{\partial f / \partial x} \right)$$

Amplitude

$$\left(\frac{\partial f}{\partial r} \right)_{\text{max}} = \sqrt{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2}$$

Département GE - DIP - Thomas Grenier

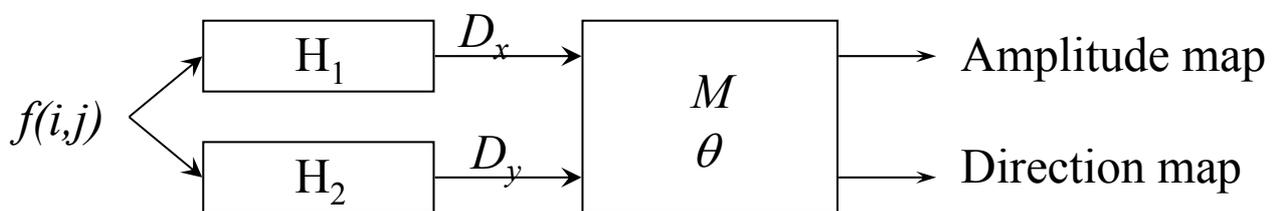
38

□ Gradient in discrete domain

- For each pixel (i,j): gradient computation in two orthogonal directions $\longrightarrow D_x, D_y$

- Gradient amplitude $M = \sqrt{D_x^2 + D_y^2}$

- Gradient direction $\theta = \text{Arctan}\left(\frac{D_y}{D_x}\right)$



- For computational efficiency, the gradient amplitude is sometimes approximated by the magnitudes combination

$$M = |D_x| + |D_y|$$

- If the gradient amplitude M is large enough (i.e., above some threshold value), an edge is deemed present
- The direction (angle) map is used to follow the contour
- Many H1 H2 operators exist:

Pixel difference $D_x(i, j) = f(i, j) - f(i-1, j)$ $\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$
 $D_y(i, j) = f(i, j) - f(i, j-1)$

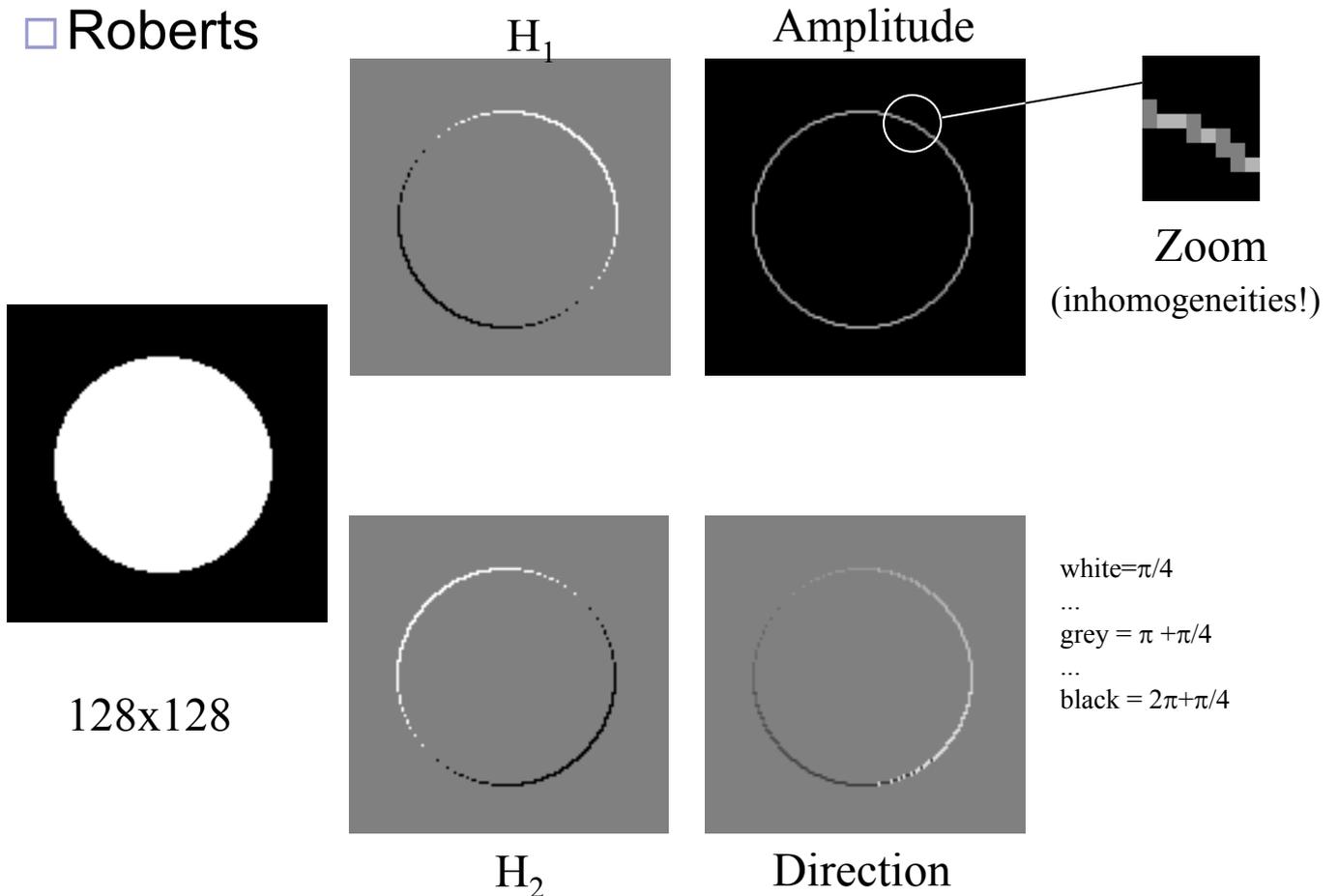
Separated pixel difference $D_x(i, j) = f(i+1, j) - f(i-1, j)$
 $D_y(i, j) = f(i, j+1) - f(i, j-1)$

→ Convolution windows!

□ H_1, H_2

Operator	Row gradient H_1	Column gradient H_2
Pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Separated pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
Roberts	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Prewitt	$\frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Frei-Chen	$\frac{1}{2 + \sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{2 + \sqrt{2}} \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$

□ Roberts



□ Roberts



H_1



Amplitude



H_2



Direction

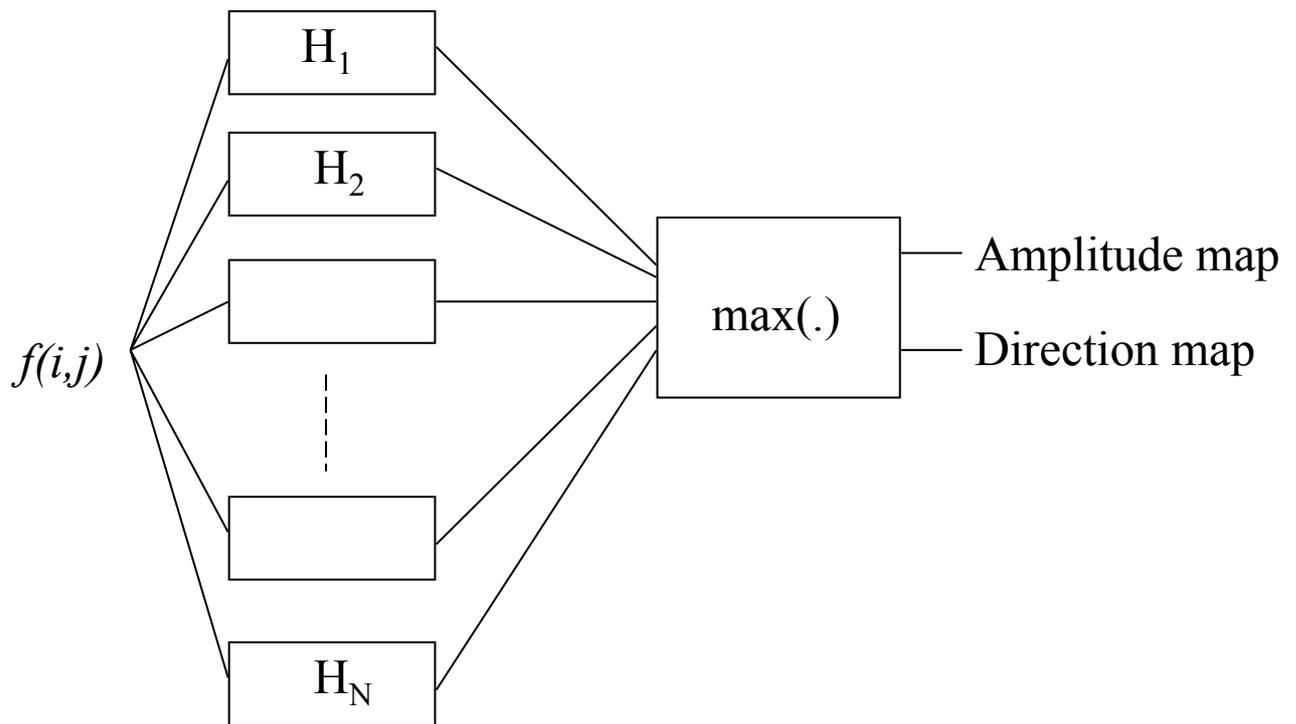


- Increasing the contrast, many edges appear due to noise
- Edge detectors are high-pass filters



□ Compass operator

- Computation of the gradient in N directions
- Selection of the maximum value



Département GE - DIP - Thomas Grenier

45

■ Examples

Gradient direction	Prewitt compass gradient	Kirsch	Robinson 3-level	Robinson 5-level
East H_1	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$
Northeast H_2	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$
North H_3	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Northwest H_4	$\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$
West H_5	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
Southwest H_6	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$
South H_7	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
Southeast H_8	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$
Scale factor	$\frac{1}{5}$	$\frac{1}{15}$	$\frac{1}{3}$	$\frac{1}{4}$

46

□ Laplacian

$$\Delta f(x, y) = \nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

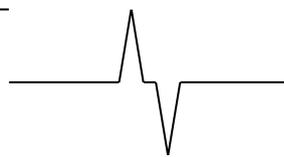
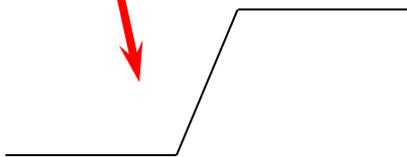
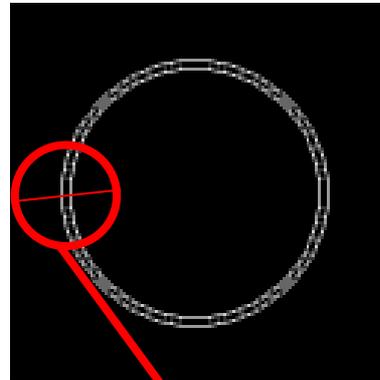
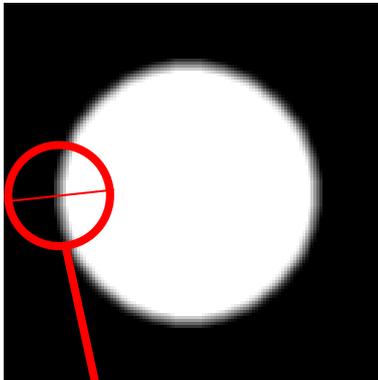
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\left. \begin{matrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \end{matrix} \right\}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

then absolute value



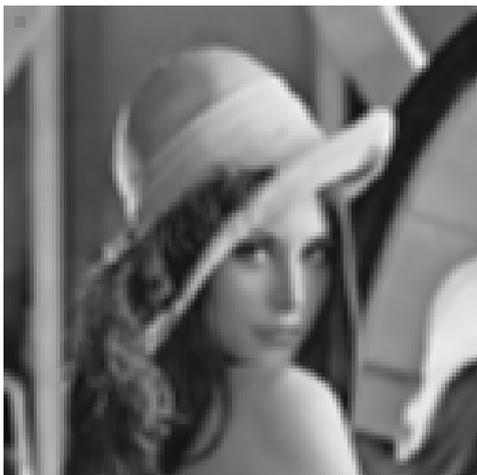
■ Emphasis filter

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \rightarrow$$



$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \text{Input Image} + \text{Laplacian}$$

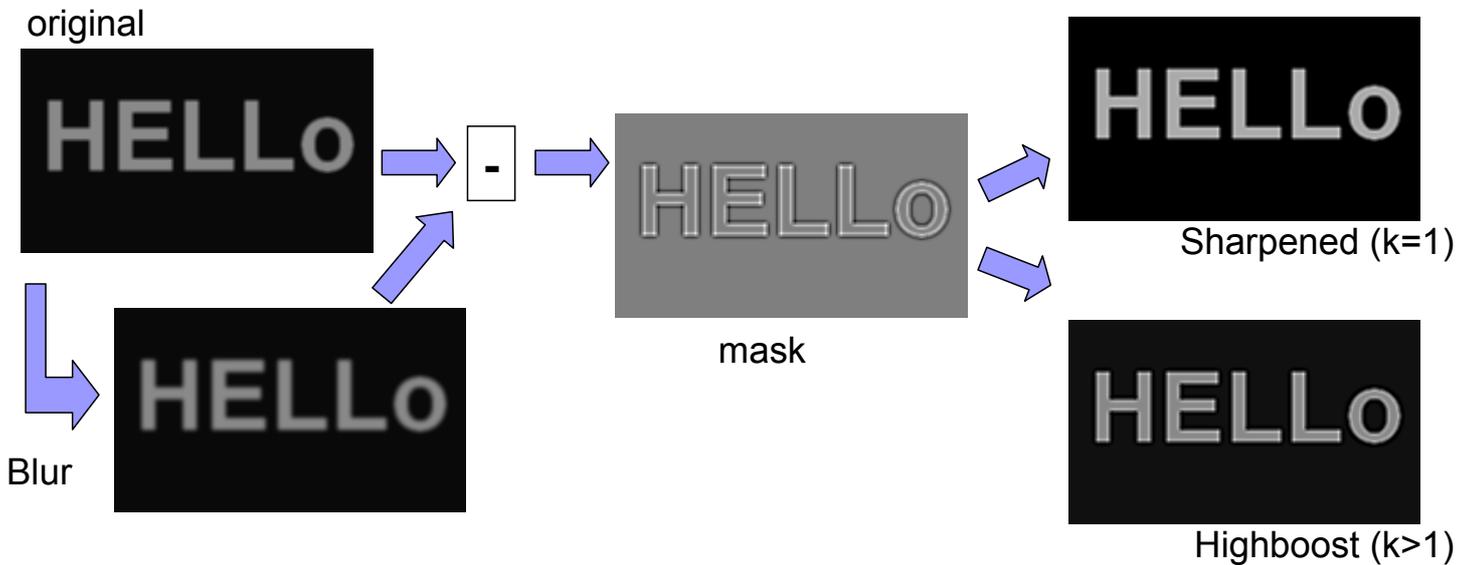
➔ Enhancement of high frequencies



■ Unsharp masking, Highboost Filtering

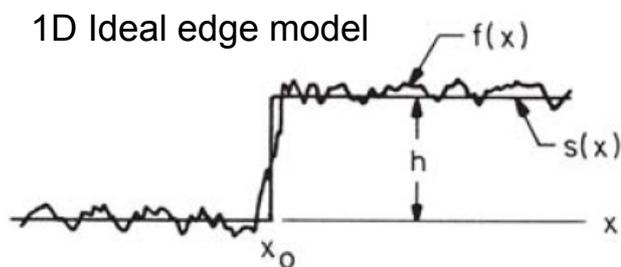
□ Used by the printing and publishing industry

- 1- Blur the original image
- 2- Subtract the blurred image from the original (the result is called the mask)
- 3- Add the mask (multiplied by k) to the original

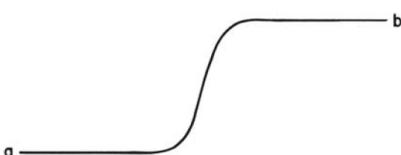


Edge fitting

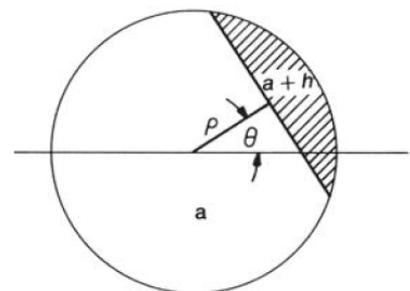
■ Image data f can be fitted to an ideal edge model s



Hyperbolic Edge model 1D



2D Ideal edge model



→ An edge is assumed present if the Mean Square Error is below a threshold value

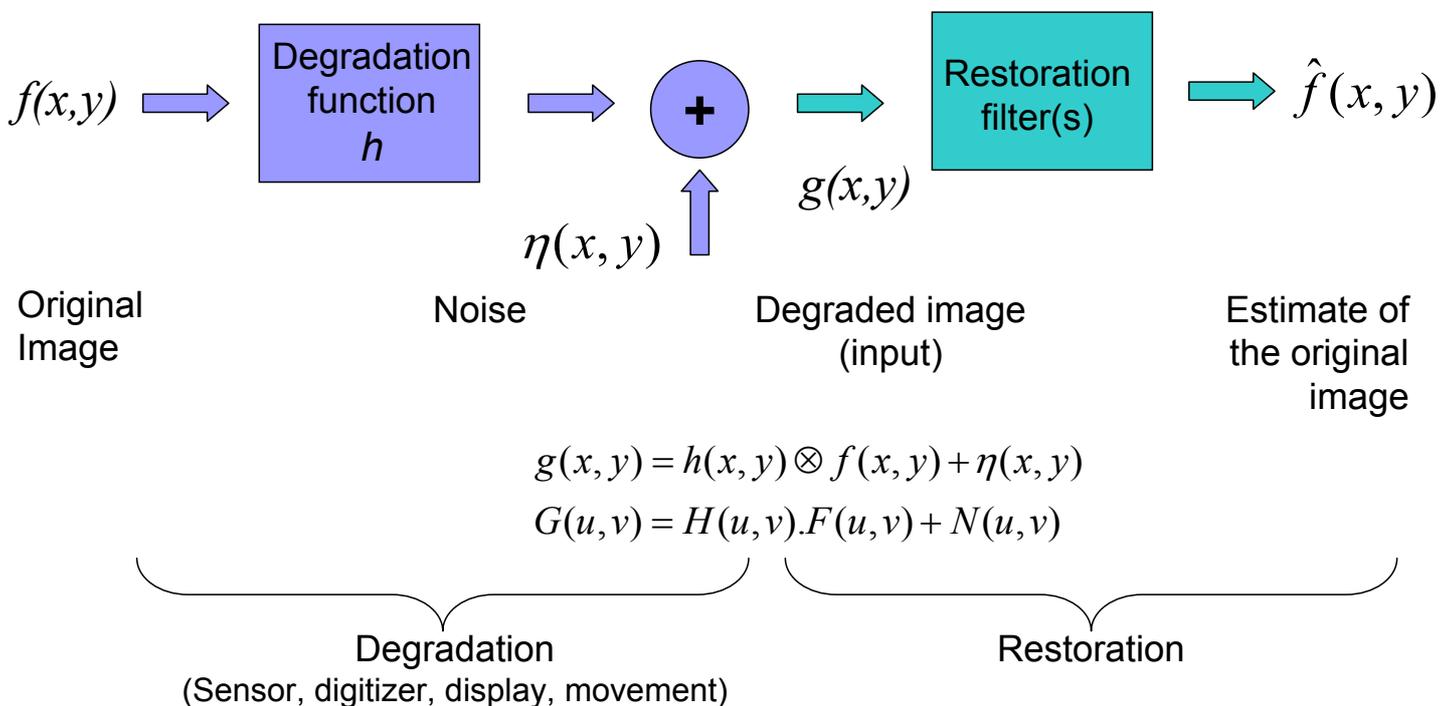
$$MSE = \int_{x_0-L}^{x_0+L} [f(x) - s(x)]^2 \cdot dx$$

Model+minimization... image restoration

Image Restoration

- Image restoration attempts to recover an image that has been degraded using a priori knowledge of degradation phenomenon
 - Modeling the degradation
 - Applying the inverse process (in order to recover the original image)
- ➔ Involves formulating a criterion of goodness that will yield an optimal estimate of the desired result

- A model of the image degradation/restoration process



■ Noise Models

- Gaussian (normal) noise

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

- Uniform noise

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$$

- Impulse (salt and pepper) noise

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

- Erlang (gamma) noise

- Rayleigh noise

$$p(z) = \begin{cases} \frac{2}{b} \cdot (z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

Restoration: presence of noise only

$$g(x, y) = f(x, y) + \eta(x, y)$$

■ Mean filters

- Arithmetic, geometric, harmonic
- $$\frac{1}{m.n} \sum_{i,j \in S_{xy}} g(i, j) \quad \left[\prod_{i,j \in S_{xy}} g(i, j) \right]^{\frac{1}{m.n}} \quad \frac{m.n}{\sum_{i,j \in S_{xy}} \frac{1}{g(i, j)}}$$

■ Order statistic filters

- Median, min & max, midpoint
- $$\hat{f}(x, y) = \underset{s,t \in S_{xy}}{\text{median}}(g(s, t)) \quad \hat{f}(x, y) = \max_{s,t \in S_{xy}}(g(s, t)) \quad \frac{1}{2} \max + \frac{1}{2} \min$$
- $$\hat{f}(x, y) = \min_{s,t \in S_{xy}}(g(s, t))$$

■ Adaptive filters

- Local noise reduction, adaptive median, ...

Adaptive Median Filter

Notations

- z_{\min} = minimum intensity value in S_{xy}
- z_{\max} = maximum intensity value in S_{xy}
- z_{med} = median of intensity values in S_{xy}
- z_{xy} = intensity value at coordinates (x,y)
- S_{\max} = maximum allowed size of S_{xy}

Algorithm for a pixel (x,y)

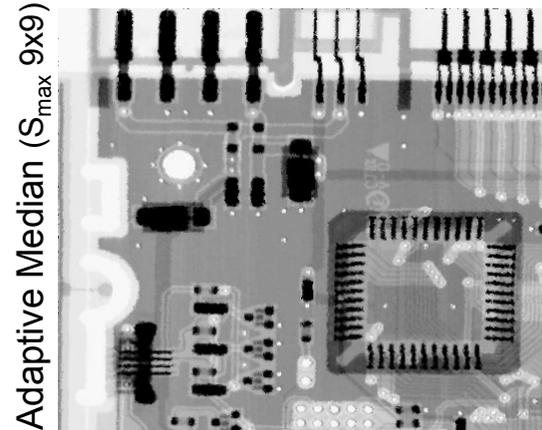
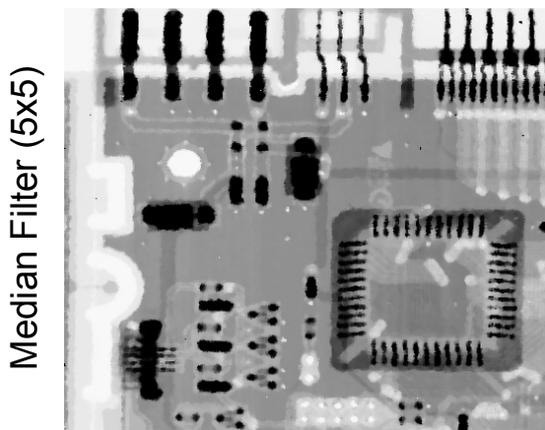
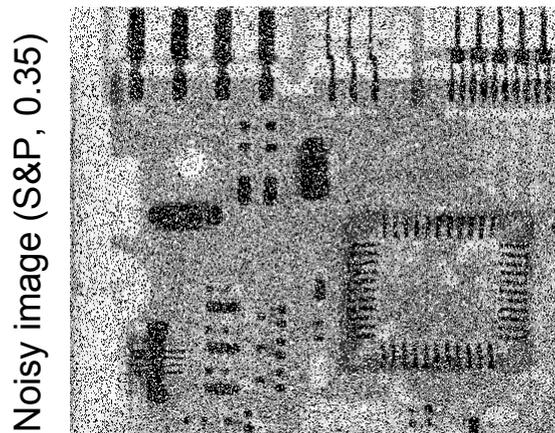
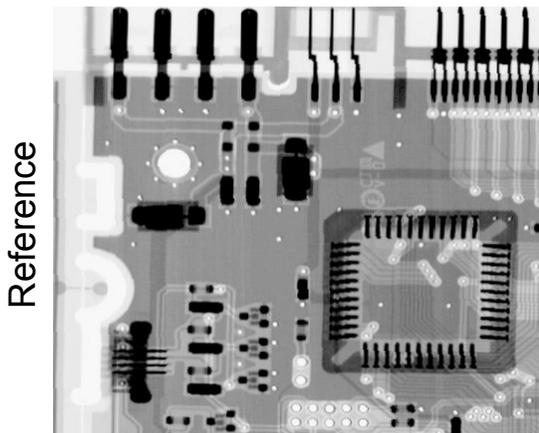
Stage 1 :

Compute z_{\min} , z_{\max} , z_{med}
 $A1 = z_{\text{med}} - z_{\min}$
 $A2 = z_{\text{med}} - z_{\max}$
 if $A1 > 0$ and $A2 < 0 \rightarrow$ stage 2
 increase the window size
 if window size $< S_{\max} \rightarrow$ stage 1
 else output z_{med}

Stage 2:

$B1 = z_{xy} - z_{\min}$
 $B2 = z_{xy} - z_{\max}$
 if $B1 > 0$ and $B2 < 0$
 output z_{xy}
 else output z_{med}

Result of Adaptive Median filter



Restoration: Periodic noise reduction

$$g(x, y) = f(x, y) + \eta(x, y)$$

■ By frequency domain filtering

- Bandreject filter
- Notch filter → optimum notch

Build H_{NP} (Notch Pass) by placing a notch pass filter at the location of each spike.
Interference noise pattern is:

$$N(u, v) = H_{NP}(u, v) \cdot G(u, v)$$

$$\text{then } \eta(x, y) = \text{FT}^{-1}[N(u, v)]$$

$$\text{thus } \hat{f}(x, y) = g(x, y) - w(x, y) \cdot \eta(x, y)$$

Estimate of $f(x, y)$

Weighted function (minimizes the effect of components not present in the estimate of η)

→ How to select $w(x, y)$?

■ $w(x, y)$?

$w(x, y)$ is selected so that the local variance of the estimate of f is minimized (optimum choice of w)

$$\frac{\partial \sigma^2(x, y)}{\partial w(x, y)} = 0$$

$$\Rightarrow w(x, y) = \frac{\overline{g(x, y) \cdot \eta(x, y)} - \bar{g}(x, y) \cdot \bar{\eta}(x, y)}{\eta^2(x, y) - \bar{\eta}^2(x, y)}$$

→ Prove the validity of this equation

Hints:

→ estimate the variance in a small neighborhood

→ assume that w remains essentially constant over the neighborhood

Restoration: linear, position-invariant degradation

$$g(x, y) = H[f(x, y)] + \eta(x, y)$$

- H is linear,

$$H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)]$$

- If H is position invariant then (for any a and b):

$$H[\delta(x-a, y-b)] = h(x-a, y-b)$$

→ **g(x,y):** $g(x, y) = h(x, y) \otimes f(x, y) + \eta(x, y)$

$$G(u, v) = H(u, v) \cdot F(u, v) + N(u, v)$$

Estimating the degradation function

Blind deconvolution

- 3 principal ways

- Observation

$$H_s(u, v) = \frac{G_s(u, v)}{\hat{F}_s(u, v)}$$

- Small rectangular section containing samples structures (part of an object, background)

- Experimentation

$$H(u, v) = \frac{G(u, v)}{A}$$

- Obtain the impulse response of the degradation function by imaging an impulse (small dot of light)

- Modeling

- Mathematical model that take into account environmental conditions that cause degradation
- Derive a mathematical model starting from basic principles

And after ?

■ Inverse filtering

Without noise

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)}$$

With noise

$$\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$$

■ We have to know N!

■ What happens for small values of H(u,v) ?

■ Minimum Mean Square Error (Wiener) Filtering

■ Constrained Least Squares Filtering

■ ...

Geometrical image modification

■ Spatial transformations

Example

■ Shrink image to half its size

$$(x', y') = T\{(x, y)\} = (x/2, y/2)$$

Affine transform:

$$[x', y', 1] = [x, y, 1] \cdot \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

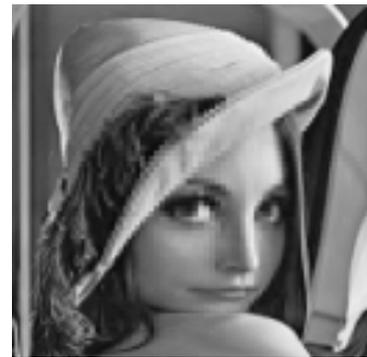
Higher order

$$[x', y', 1] = [x, y, x^2, y^2, xy, \dots, 1] \cdot \mathbf{T}$$

→ Estimate (or compute) the inverse matrix

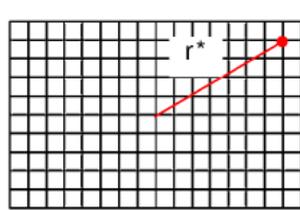
→ If needed, use interpolation

□ Higher order transforms

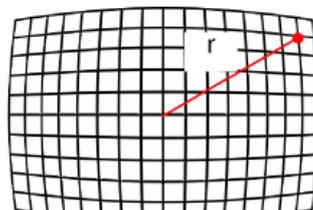


Applications :

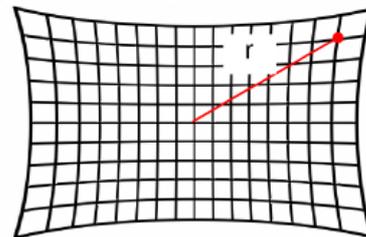
Lens distortion correction, perspective



Orthoscopic
projection



Barrel
distortion



Pincushion
distorsion

Département GE - DIP - Thomas Grenier

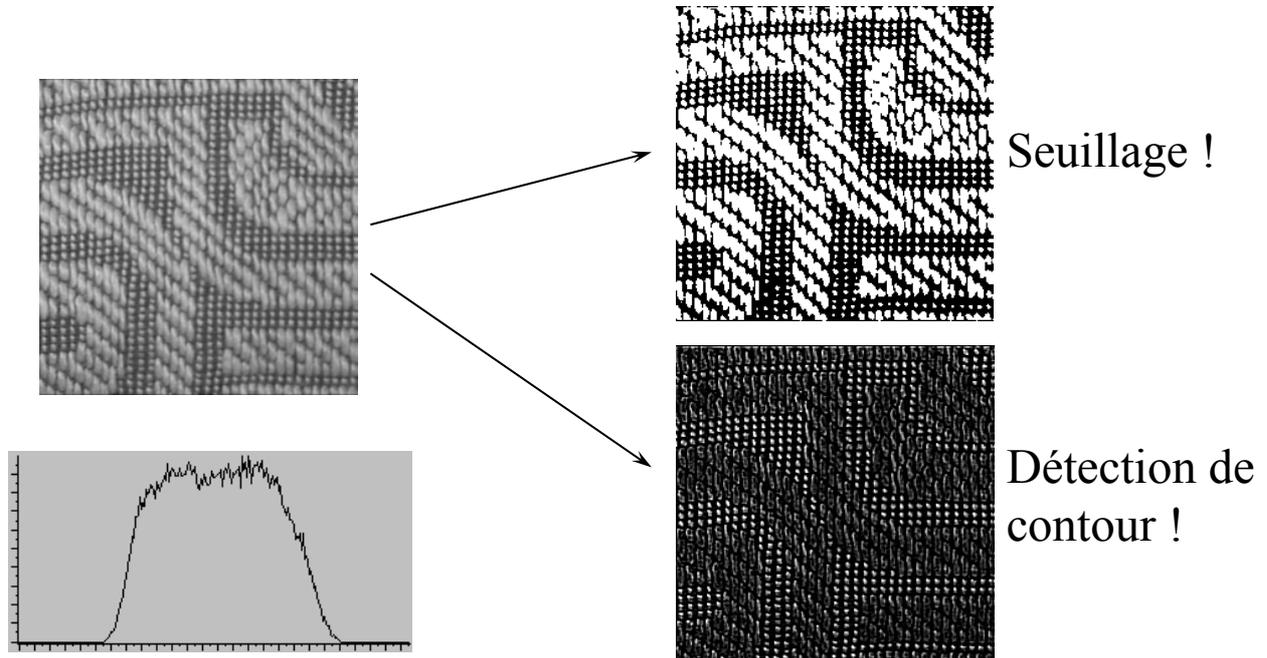
63

■ End ...

□ Before texture

Analyse de texture

Région \neq zone de NG ou de couleur homogène



Texture = information visuelle qualitative:

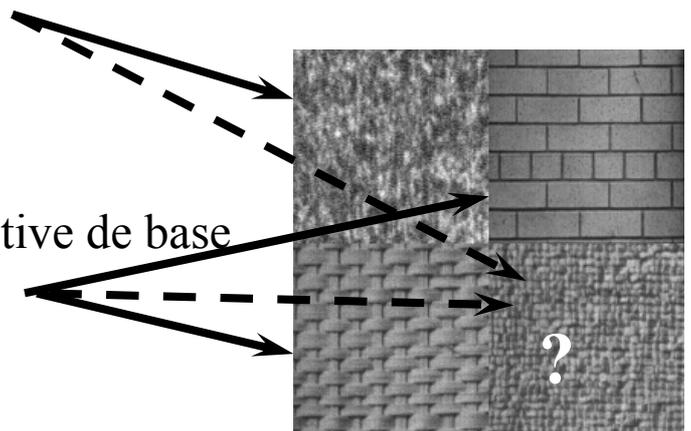
Grossière, fine, tachetée, marbrée, régulière, périodique...

Région homogène: Assemblage plus ou moins régulier de primitives plus ou moins similaires.

Analyse de texture = formalisation de ces critères

Texture microscopique: Aspect chaotique mais régulier, primitive de base réduite.

Texture macroscopique: primitive de base évidente, assemblage régulier.



Méthodes d'analyse de texture:

Structurelles: recherche de primitives de base bien définies et de leur organisation (règles de placement)

Méthodes peu utilisées

Stochastiques: primitives mal définies et organisation +/- aléatoire.

Principe: évaluation d'un paramètre dans une petite région

(fenêtre de taille dépendant de la texture (!)):

Analyse fréquentielle, statistiques, comptage

d'événements, corrélation,....

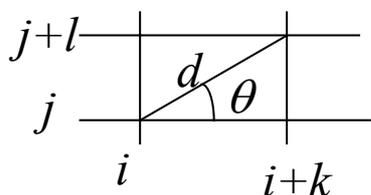
Pas de modèle général de texture \longrightarrow

Nombreuses méthodes ad-hoc.

Exemple de méthode: Matrices de co-occurrence

Statistique du second ordre:

$$\text{Pr.}(f(i,j)=a \text{ et } f(i+k,j+l)=b) = p(k,l;a,b) = p(d,\theta;a,b)$$



$$d = 1, \theta = 0^\circ \quad (k=1, l=0)$$

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

$a \backslash b$	0	1	2	3
0	2	2	1	0
1	0	2	0	0
2	0	0	3	1
3	0	0	0	1

$a \backslash b$	0	1	2	3
0	4	2	1	0
1	2	4	0	0
2	1	0	6	1
3	0	0	1	2

(en symétrique

$\theta = 0^\circ d = 1$ et $d = -1$)

Quelques Paramètres extraits des matrices de co-occurrence

Moyenne locale: $\sum_{i=1}^{NG} \sum_{j=1}^i (i+j)p(i,j)$ (i,j : ligne et colonne de la matrice de co-occurrence p)

Energie ou second moment: $\sum_{i=1}^{NG} \sum_{j=1}^i p(i,j)^2$

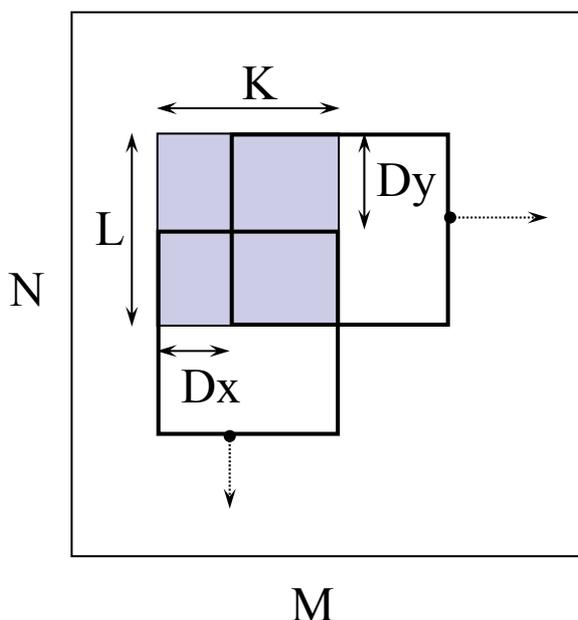
Inertie ou moment d'ordre deux des différences : $\sum_{i=1}^{NG} \sum_{j=1}^i (i-j)^2 p(i,j)$

Autocorrélation: $\sum_{i=1}^{NG} \sum_{j=1}^i i.j p(i,j)$

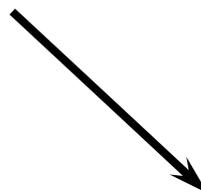
Contraste: $\sum_{i=1}^{NG} \sum_{j=1}^i (i+j)^2 p(i,j)$

- Il y en a d'autres
- L'interprétation visuelle est difficile.

Application de l'analyse de texture



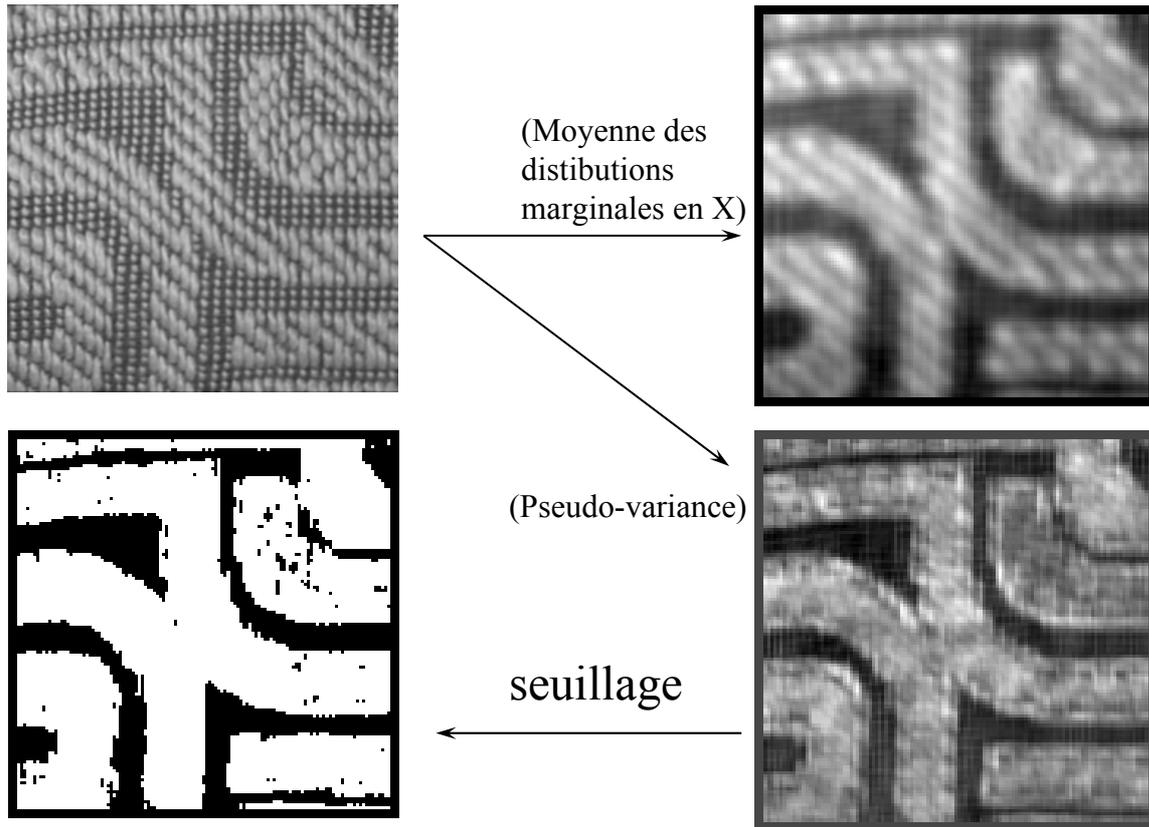
Mesure de paramètres dans une fenêtre de taille K,L
Avec un pas de déplacement Dx, Dy



Cartes de paramètres

Application des matrices de co-occurrence

Fenêtre 16x16, pas 2x2, $k=1, l=0$



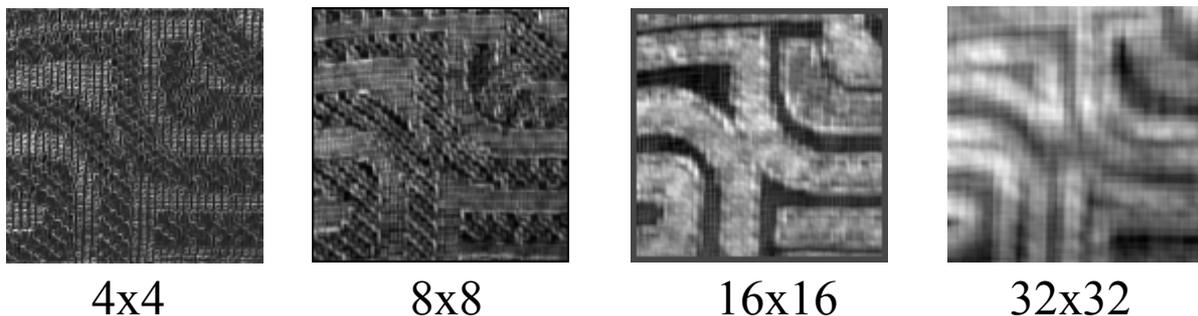
Département GE - DIP - Thomas Grenier

71

Influence des paramètres

- Exemple : Choix de la taille de la fenêtre

(Matrice de co-occurrence : Pseudo-variance)



Le choix et les réglages des paramètres sont difficiles. Il faut souvent faire de nombreux essais.

Les paramètres obtenus doivent être pertinents pour l'opération suivante de **segmentation**.

Département GE - DIP - Thomas Grenier

72

Traitement d'images numériques

Analyse d'images: 1^{ère} partie

*Département Génie Electrique
5GE - TdSi*

Sommaire

- I. Introduction
- II. Fondamentaux en imagerie numérique
- III. Traitement discret 2D
- IV. Amélioration d'image

V. Analyse d'images (6 heures)

Analyse d'images

■ Sources qui ont inspirées cette partie:

- ★ *Henri Maître (2003) : Le traitement des images, Hermes-Lavoisier*
- ★ *J.P. Cocquerez (1995) : Analyse d'images: filtrage et segmentation, Hermes-Lavoisier*
- ★ *http://www.ensta.fr/~manzaner/Support_Cours.html*
- ★ *<http://www.tsi.enst.fr/~tupin/TDI.html>*

Analyse d'images: présentation

- ▶ *A quoi peut servir l'analyse d'images?*

Analyse d'images: présentation

▶ *A quoi peut servir l'analyse d'images?*

- *Reconnaissance*
- *Suivi*
- *Anticipation*
- *Détection*
- *Prise de décision*
- *...*

Analyse d'images: présentation

▶ *A quoi peut servir l'analyse d'images?*

- *Reconnaissance*



Analyse d'images: présentation

► A quoi peut servir l'analyse d'images?

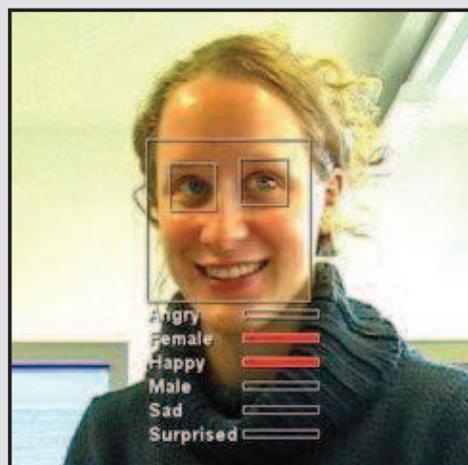
● Reconnaissance



Analyse d'images: présentation

► A quoi peut servir l'analyse d'images?

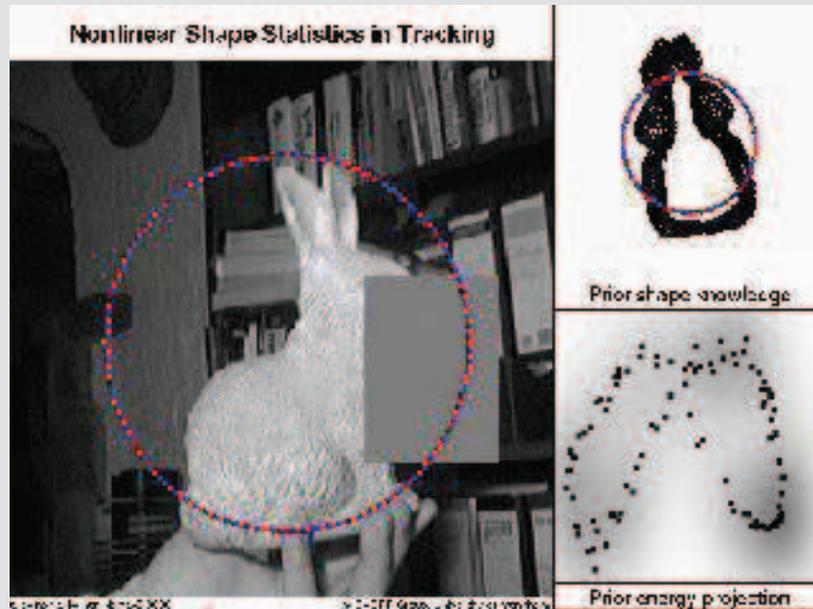
● Reconnaissance



Analyse d'images: présentation

- ▶ *A quoi peut servir l'analyse d'images?*

- *Reconnaissance*



9

Analyse d'images: présentation

- ▶ *A quoi peut servir l'analyse d'images?*

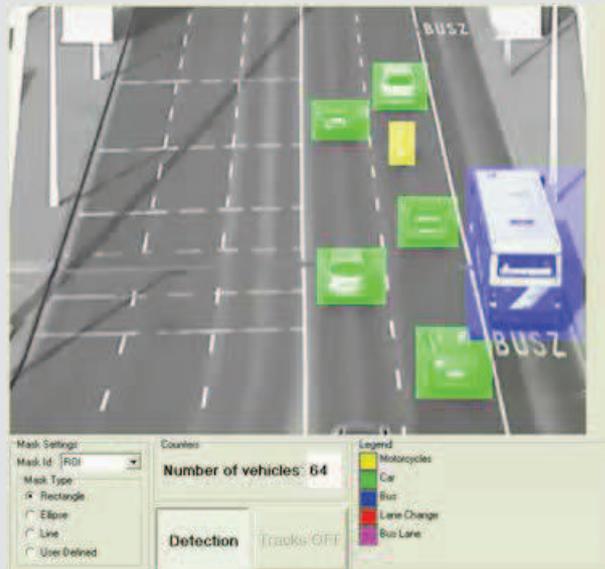
- *Suivi*



Analyse d'images: présentation

► A quoi peut servir l'analyse d'images?

● Suivi



Analyse d'images: présentation

► A quoi peut servir l'analyse d'images?

● Suivi



Analyse d'images: présentation

- ▶ *Dans quelles applications trouve t'on de l'analyse d'images ?*

Analyse d'images: présentation

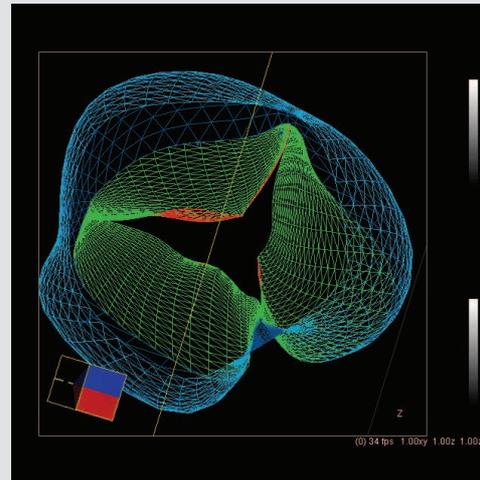
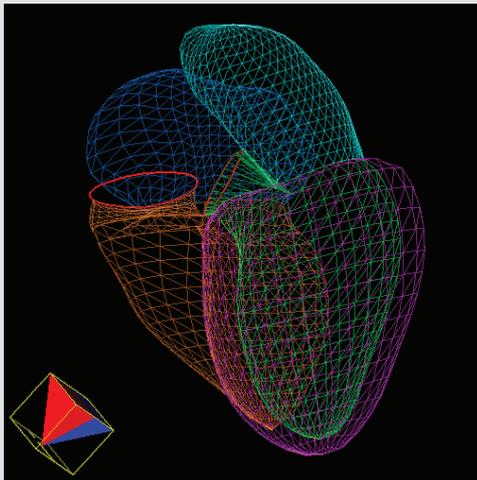
- ▶ *Dans quelles applications trouve t'on de l'analyse d'images ?*

- Industrie (vision industrielle)
- Imagerie médicale
- Domaine multimédia
 - Internet (video, applications type google map)
 - Téléphone portable
 - Appareil photo numérique
 - GPS, ...

Analyse d'images: présentation

- ▶ *Dans quelles applications trouve t'on de l'analyse d'images ?*

- **Imagerie médicale**



Analyse d'images: présentation

- ▶ *Dans quelles applications trouve t'on de l'analyse d'images ?*

- **Domaine multimédia**



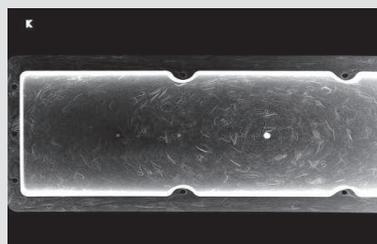
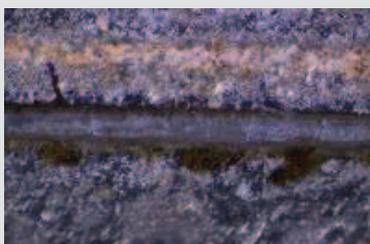
Analyse d'images: présentation

- ▶ *L'ensemble des résultats du cours obtenus en utilisant matlab est téléchargeable à l'adresse internet suivante*

www.creatis.insa-lyon.fr/~bernard/AnalyseImage

Analyse d'images: présentation

- ▶ *A la fin de ce module, vous devrez réaliser des mini-projets industriels pendant un créneau horaire de 16 heures.*
- ▶ *Date de présentation des mini-projets: mercredi 1^{er} décembre*



Analyse d'images: présentation

- ▶ *Le but de ce module est de vous rendre autonome pour la réalisation d'applications ayant comme support l'image*
- ▶ *Exemple de sujet d'Examen 2009/2010*

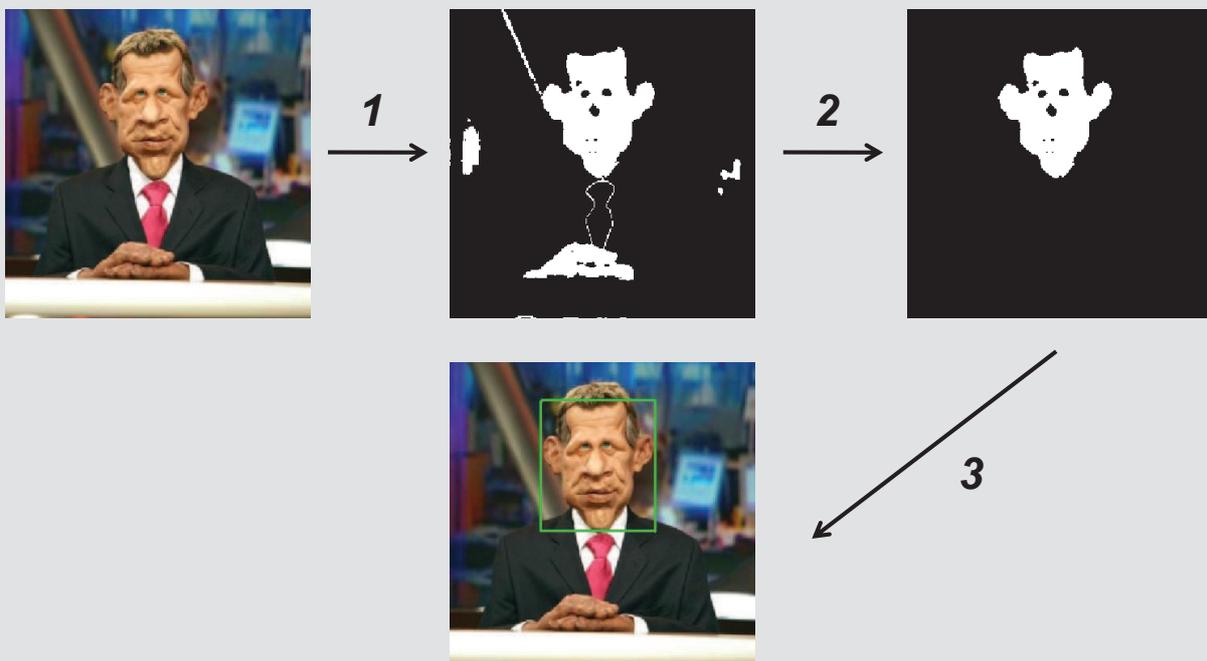


Département GE - TI - Olivier Bernard

19

Analyse d'images: présentation

- ▶ *Exemple de sujet d'Examen 2009/2010*



Département GE - TI - Olivier Bernard

20

Sommaire

V. Analyse d'images

- ✓ **1^{ère} partie**
 - Morphologie mathématique
 - Détection et analyse de contours
- ✓ **2^{ème} partie**
 - Segmentation par régions
- ✓ **3^{ème} partie**
 - Analyse de formes
 - Reconnaissance de formes

Sommaire

V. Analyse d'images

- ✓ **1^{ère} partie**
 - Morphologie mathématique
 - Détection et analyse de contours
- ✓ **2^{ème} partie**
 - Segmentation par régions
- ✓ **3^{ème} partie**
 - Analyse de formes
 - Reconnaissance de formes

Morphologie mathématique

■ Contexte

- Définition
- Principe
- Propriétés

■ Les opérateurs classiques

- Dilatation / Erosion
- Gradient / Laplacien morphologique
- Ouverture / Fermeture
- Filtres alternés séquentiels
- Autres

Morphologie mathématique

■ Contexte

- Définition
- Principe
- Propriétés

■ Les opérateurs classiques

- Dilatation / Erosion
- Gradient / Laplacien morphologique
- Ouverture / Fermeture
- Filtres alternés séquentiels
- Autres

Morphologie mathématique

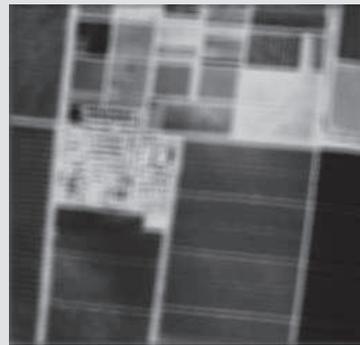
■ Contexte - Définition

En traitement linéaire des images, filtrer c'est éliminer certaines composantes fréquentielles des images

➔ Filtrage = convolution



Image initiale



Filtre moyenneur

Morphologie mathématique

■ Contexte - Définition

En morphologie mathématique, filtrer c'est simplifier l'image en supprimant certaines structures géométriques



Image initiale



Filtre morphologique

Morphologie mathématique

■ Contexte - Définition

- ▶ **Objectif**: mettre en évidence la morphologie (forme) d'un objet
- ▶ **Outils mathématiques** : théorie des ensembles, treillis (ensemble muni d'une relation d'ordre), topologie des fermés, probabilités
- ▶ **Exploitation d'opérateurs ensemblistes classiques** (union, intersection, inclusion, complémentation)

Morphologie mathématique

■ Contexte

- Définition
- Principe
- Propriétés

■ Les opérateurs classiques

- Dilatation / Erosion
- Gradient / Laplacien morphologique
- Ouverture / Fermeture
- Filtres alternés séquentiels
- Autres

Morphologie mathématique

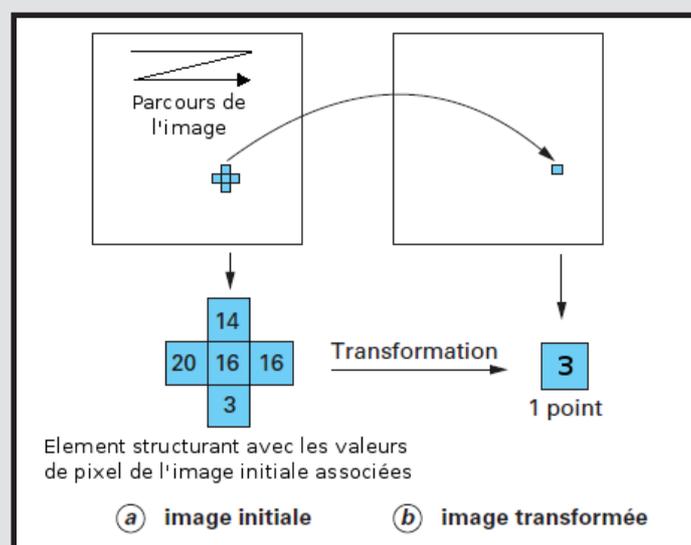
■ Contexte - Principe

- ▶ Soit une image A à traiter
- ▶ On définit une surface appelée élément structurant B
- ▶ B se déplace sur A comme un opérateur de convolution
- ▶ En chaque position de B, des opérations logiques (essentiellement non linéaire) sont effectuées entre A et B

Morphologie mathématique

■ Contexte - Principe

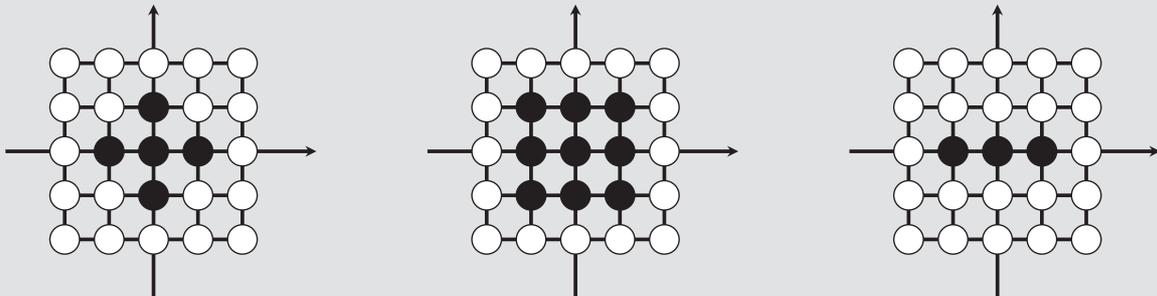
- ▶ Schéma de principe des filtres morphologiques



Morphologie mathématique

■ Contexte - Principe

▶ Exemple d'éléments structurants



▶ Les résultats dépendent de la taille et de la forme de l'élément structurant utilisé

Morphologie mathématique

■ Contexte

- Définition
- Principe
- Propriétés

■ Les opérateurs classiques

- Dilatation / Erosion
- Gradient / Laplacien morphologique
- Ouverture / Fermeture
- Filtrés alternés séquentiels
- Autres

Morphologie mathématique

■ Contexte - Propriétés

- ▶ Le filtre morphologique simplifie l'image en préservant la structure, mais il perd en général de l'information

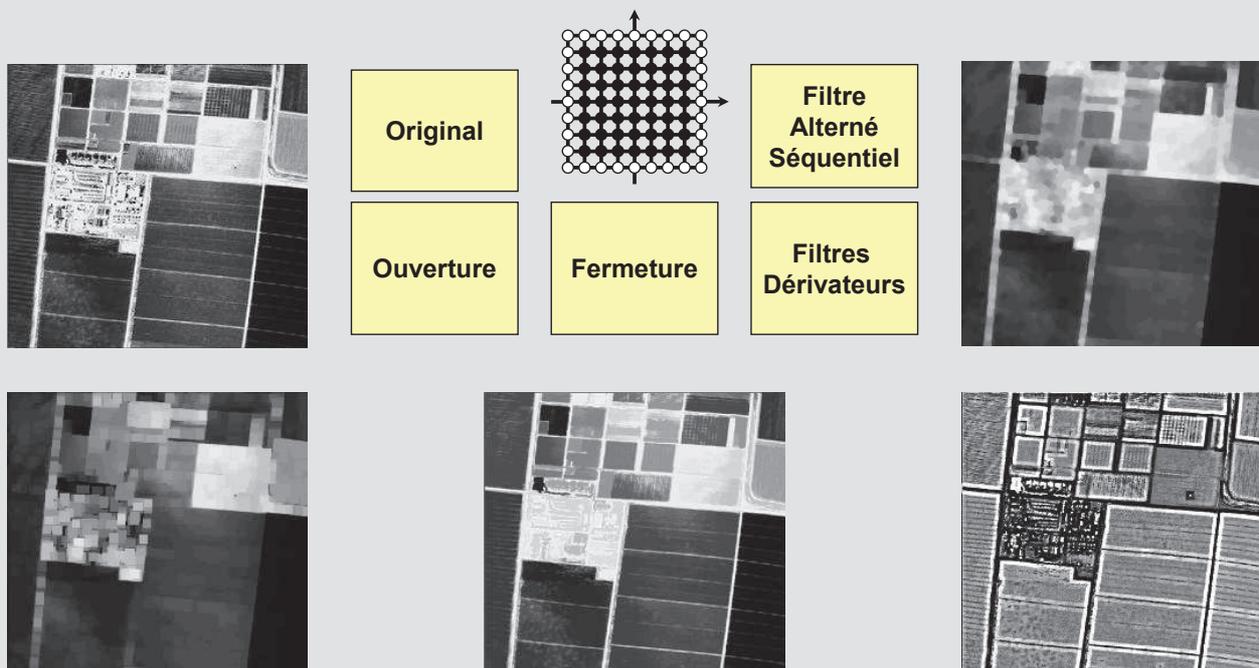
➡ Propriété de croissance

$$\forall X, Y, \quad X \subset Y \Rightarrow \Psi(X) \subset \Psi(Y)$$

- ▶ Le filtre morphologique est stable et possède une classe d'invariance connue

Morphologie mathématique

■ Contexte – Les opérateurs étudiés



Morphologie mathématique

■ Contexte

- Définition
- Principe
- Propriétés

■ Les opérateurs classiques

- Dilatation / Erosion
- Gradient / Laplacien morphologique
- Ouverture / Fermeture
- Filtres alternés séquentiels
- Autres

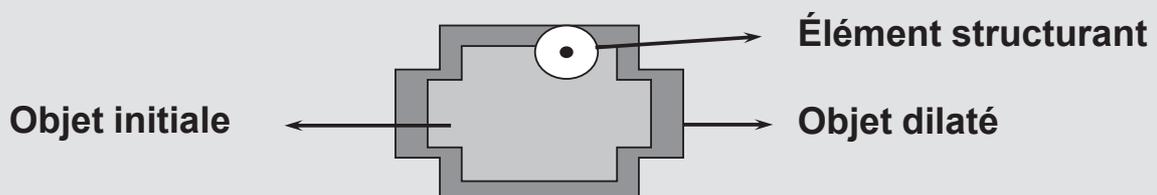
Morphologie mathématique

■ Dilatation binaire

▶ Définition

Ensemble des pixels recouverts par l'élément structurant, lorsque celui-ci parcourt l'objet (son centre restant dans l'objet)

▶ Illustration



Morphologie mathématique

► Code matlab correspondant - base

```
% Lecture image
in = imread('leafBin.png');
in = double(in);
[dimI,dimJ] = size(in);

% Element structurant
se=[-1  0 0 0 1;
     0 -1 0 1 0];

% Dilatation de base
out=zeros(dimI,dimJ);
for i=2:(dimI-1)
    for j=2:(dimJ-1)
        if (in(i,j)>0)
            for k=1:size(se,2)
                out(i+se(1,k),j+se(2,k))=1;
            end
        end
    end
end
```

Morphologie mathématique

► Code matlab correspondant - base

```
% Lecture image
in = imread('leafBin.png');
in = double(in);
[dimI,dimJ] = size(in);

% Element structurant
se=[-1  0 0 0 1;
     0 -1 0 1 0];

% Dilatation de base
out=zeros(dimI,dimJ);
for i=2:(dimI-1)
    for j=2:(dimJ-1)
        if (in(i,j)>0)
            for k=1:size(se,2)
                out(i+se(1,k),j+se(2,k))=1;
            end
        end
    end
end
```

Lecture d'une image



in

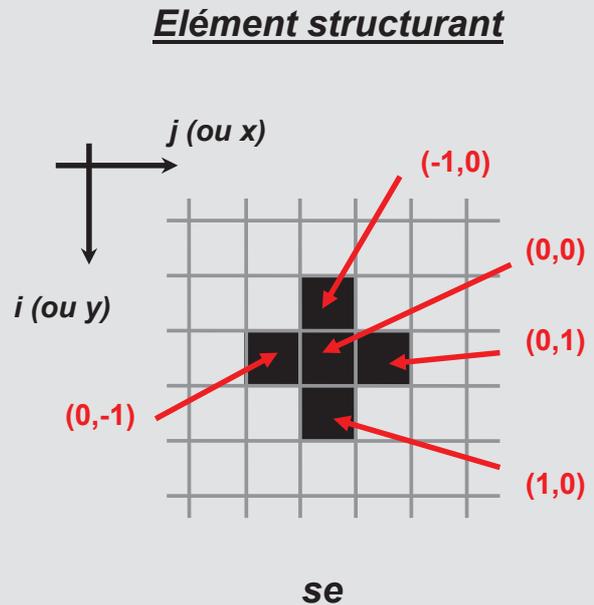
Morphologie mathématique

► Code matlab correspondant - base

```
% Lecture image
in = imread('leafBin.png');
in = double(in);
[dimI,dimJ] = size(in);

% Element structurant
se=[-1 0 0 0 1;
    0 -1 0 1 0];

% Dilatation de base
out=zeros(dimI,dimJ);
for i=2:(dimI-1)
    for j=2:(dimJ-1)
        if (in(i,j)>0)
            for k=1:size(se,2)
                out(i+se(1,k),j+se(2,k))=1;
            end
        end
    end
end
```



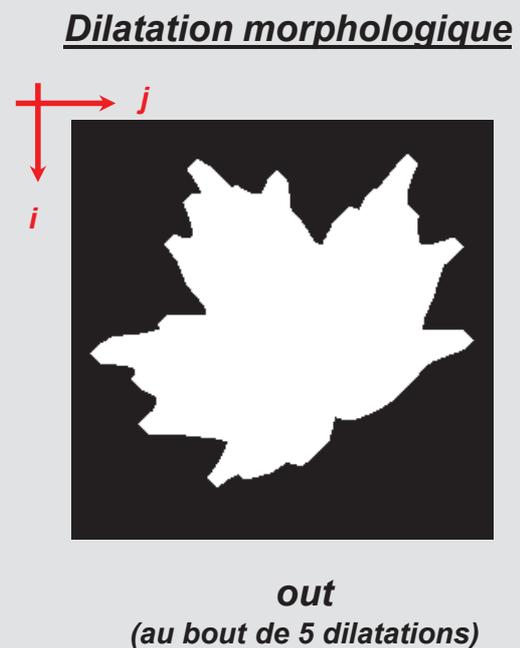
Morphologie mathématique

► Code matlab correspondant - base

```
% Lecture image
in = imread('leafBin.png');
in = double(in);
[dimI,dimJ] = size(in);

% Element structurant
se=[-1 0 0 0 1;
    0 -1 0 1 0];

% Dilatation de base
out=zeros(dimI,dimJ);
for i=2:(dimI-1)
    for j=2:(dimJ-1)
        if (in(i,j)>0)
            for k=1:size(se,2)
                out(i+se(1,k),j+se(2,k))=1;
            end
        end
    end
end
```



Morphologie mathématique

- ▶ *Code matlab correspondant - optimisé*



in

```
% Lecture image
in = imread('leafBin.png');
in = double(in);

% Element structurant
se = strel('disk',1);

% Dilatation de base
out = imdilate(in,se);
```



out
(au bout de 5 dilatations)

Morphologie mathématique

■ Dilatation binaire

- ▶ Expressions

$$\begin{cases} \delta_B(X) = X \oplus B \\ \delta_B(X) = \{z \mid X \cap B_z \neq \emptyset\} \end{cases} \quad \text{où } B_z \text{ est l'élément structurant centré en } z$$

- ▶ Propriétés

Permet d'ajouter à l'objet initial les pixels qui lui sont proches, dans des directions dépendant de l'élément structurant choisi

Morphologie mathématique

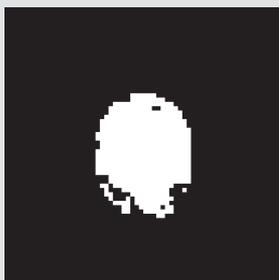
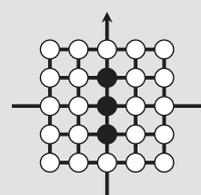
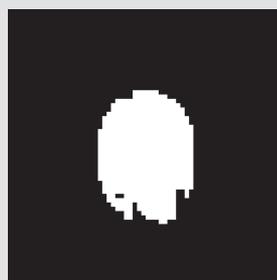
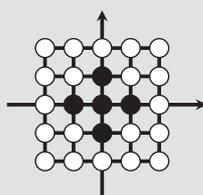
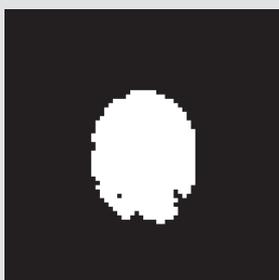
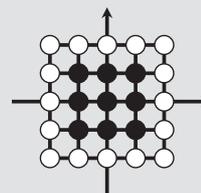
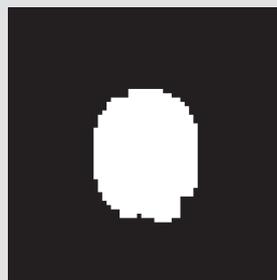


Image initiale



Morphologie mathématique

■ Dilatation fonctionnelle

► Expression

$$\delta_B(f) = \sup_{y \in B_x} f(y)$$

où f est un fonction
(par exemple une image)



Image initiale

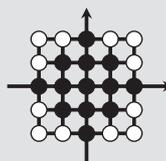


Image dilatée

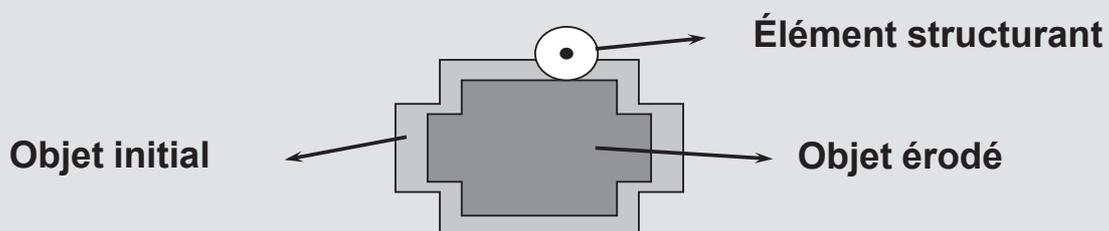
Morphologie mathématique

■ Érosion binaire

▶ Définition

- Opération complémentaire à la dilatation
- Ensemble des pixels non recouverts par l'élément structurant, lorsque celui-ci parcourt l'objet (son centre restant hors de l'objet)

▶ Illustration



Morphologie mathématique

■ Érosion binaire

▶ Expressions

$$\begin{cases} \varepsilon_B(X) = X \ominus B \\ \varepsilon_B(X) = \bigcap_z B_z \subset X \end{cases} \quad \text{avec } B_z \text{ l'élément structurant} \\ \text{centré en } z$$

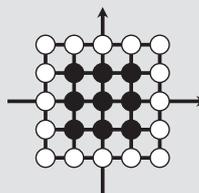
▶ Propriétés

Permet d'enlever à l'objet initial les pixels qui sont proches de ses bords, dans des directions dépendant de l'élément structurant choisi

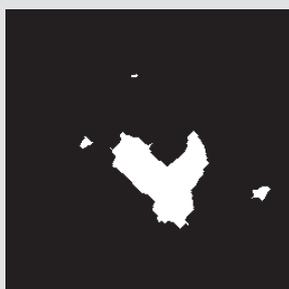
Morphologie mathématique



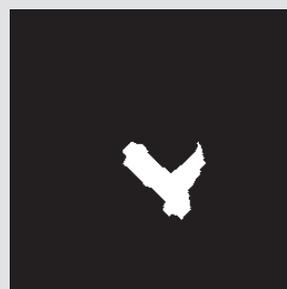
Image initiale



2 érosions



6 érosions



12 érosions

Morphologie mathématique

■ Erosion fonctionnelle

► Expression

$$\varepsilon_B(f) = \inf_{y \in B_x} f(y)$$

où f est un fonction
(par exemple une image)



Image initiale

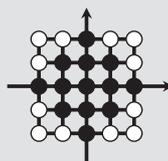


Image érodée

Morphologie mathématique

■ Dilatation, érosion

- ▶ Tous les opérateurs morphologiques sont basés sur une opération combinatoire des opérateurs de dilatation et d'érosion

- ▶ Exemples:

$$G_B(f) = |\delta_B(f) - \varepsilon_B(f)|$$

$$\lambda_B(f) = \delta_B(f) + \varepsilon_B(f) - 2f$$

$$\gamma_B(f) = \delta_B(\varepsilon_B(f))$$

$$\varphi_B(f) = \varepsilon_B(\delta_B(f))$$

Morphologie mathématique

■ Contexte

- Définition
- Principe
- Propriétés

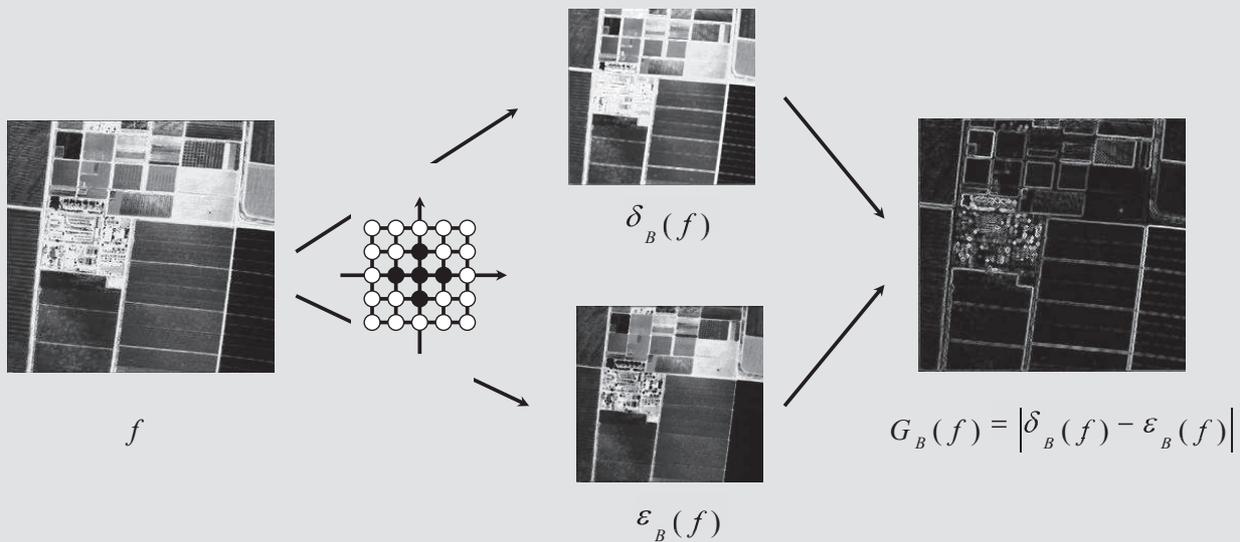
■ Les opérateurs classiques

- Dilatation / Erosion
- Gradient / Laplacien morphologique
- Ouverture / Fermeture
- Filtrés alternés séquentiels
- Autres

Morphologie mathématique

■ Gradient fonctionnelle

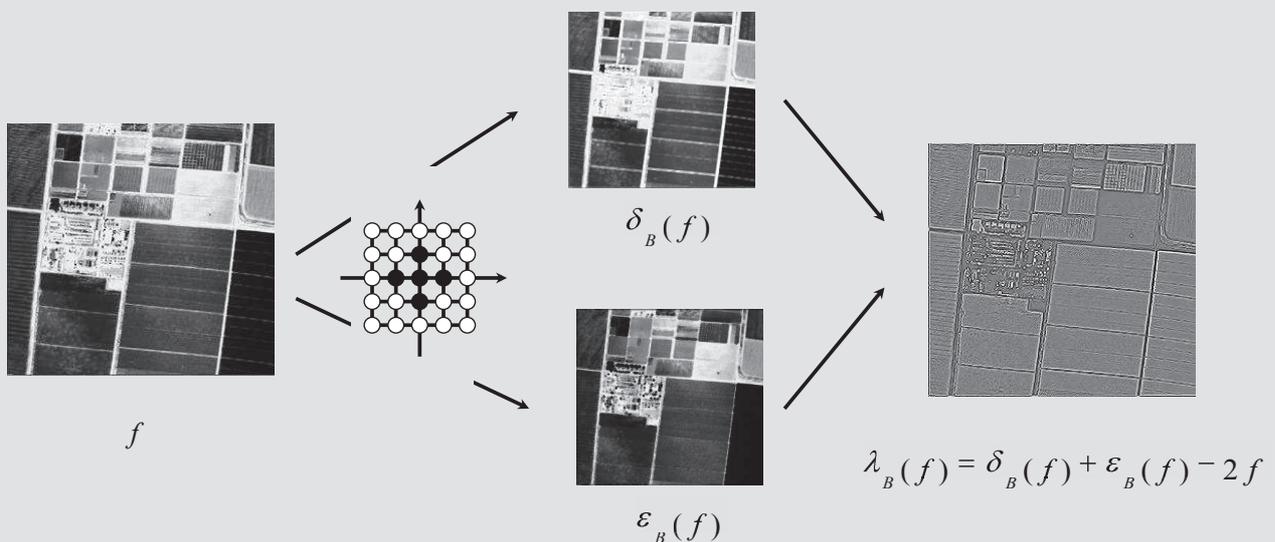
► $G_B(f) = |\delta_B(f) - \varepsilon_B(f)|$



Morphologie mathématique

■ Laplacien fonctionnelle

► $\lambda_B(f) = \delta_B(f) + \varepsilon_B(f) - 2f$



Morphologie mathématique

■ Contexte

- Définition
- Principe
- Propriétés

■ Les opérateurs classiques

- Dilatation / Erosion
- Gradient / Laplacien morphologique
- Ouverture / Fermeture
- Filtres alternés séquentiels
- Autres

Morphologie mathématique

■ Ouverture

▶ Expressions

$$\begin{cases} \gamma_B(f) = f \circ B \\ \gamma_B(f) = \delta_B(\varepsilon_B(f)) = (f \ominus B) \oplus B \end{cases}$$

▶ Propriétés

Lisse (nettoie) les contours, supprime les petits îlots ou parties saillantes tout en gardant la taille originale de l'objet

Morphologie mathématique

■ Fermeture

▶ Expressions

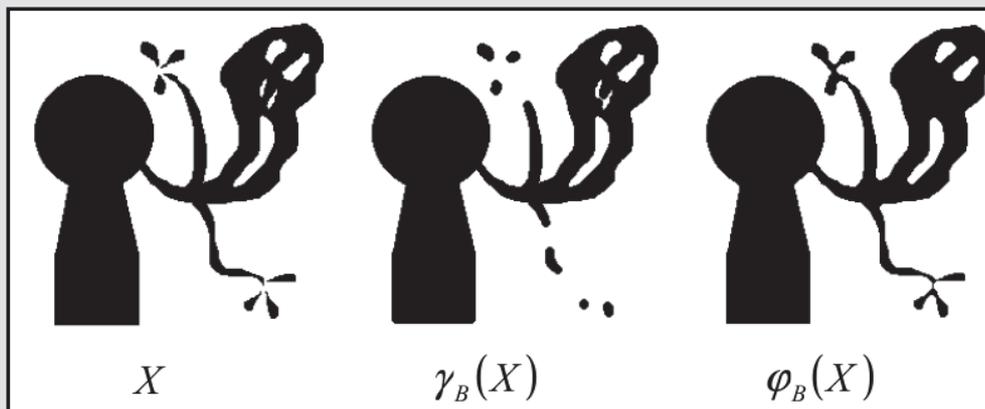
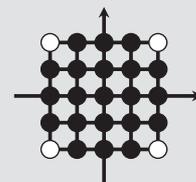
$$\begin{cases} \varphi_B(f) = f \bullet B \\ \varphi_B(f) = \varepsilon_B(\delta_B(f)) = (f \oplus B) \ominus B \end{cases}$$

▶ Propriétés

Supprime les trous, complète les défauts de forme, connecte les objets proches

Morphologie mathématique

▶ Exemple: ouverture et fermeture binaires



X

$\gamma_B(X)$

$\varphi_B(X)$

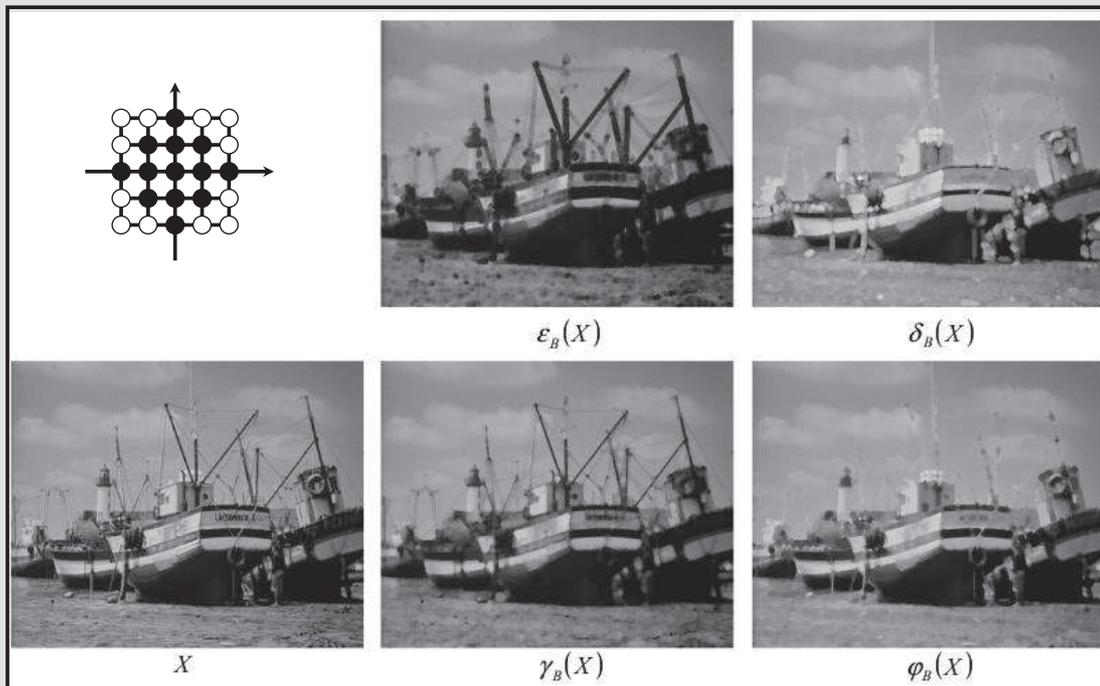
image originale

ouverture

fermeture

Morphologie mathématique

- ▶ Exemple: ouverture et fermeture fonctionnelles



Département GE - TI - Olivier Bernard

57

Morphologie mathématique

■ Contexte

- Définition
- Principe
- Propriétés

■ Les opérateurs classiques

- Dilatation / Erosion
- Gradient / Laplacien morphologique
- Ouverture / Fermeture
- Filtres alternés séquentiels**
- Autres

Département GE - TI - Olivier Bernard

58

Morphologie mathématique

■ Filtres alternés séquentiels

▶ Définition

Succession d'ouvertures et de fermetures utilisant des éléments structurants de taille croissante

▶ Expressions

$$\begin{cases} \Xi_1(f) = \gamma_1(\varphi_1(f)) \\ \Xi_n(f) = \gamma_n(\varphi_n(\Xi_{n-1}(f))) \end{cases} \quad \begin{cases} \Theta_1(f) = \varphi_1(\gamma_1(f)) \\ \Theta_n(f) = \varphi_n(\gamma_n(\Theta_{n-1}(f))) \end{cases}$$

▶ Propriétés

Conduisent à une bonne réduction du bruit grâce à une élimination progressive des pics et de creux

Morphologie mathématique

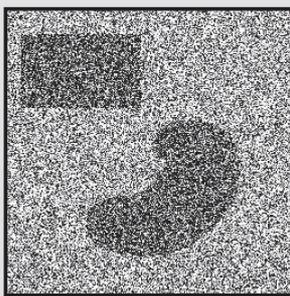
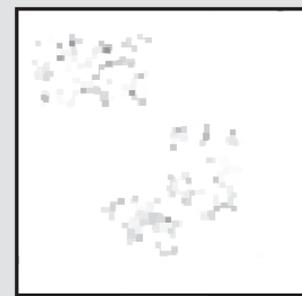
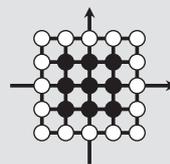
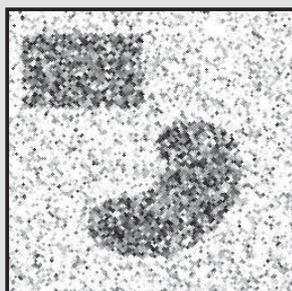


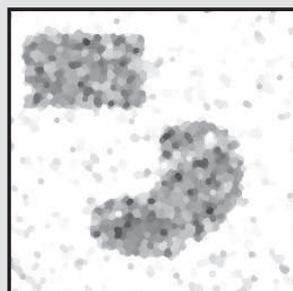
Image initiale



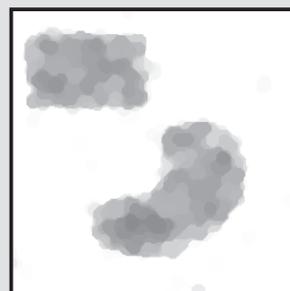
$\gamma_2 \varphi_2$



Ξ_1



Ξ_2

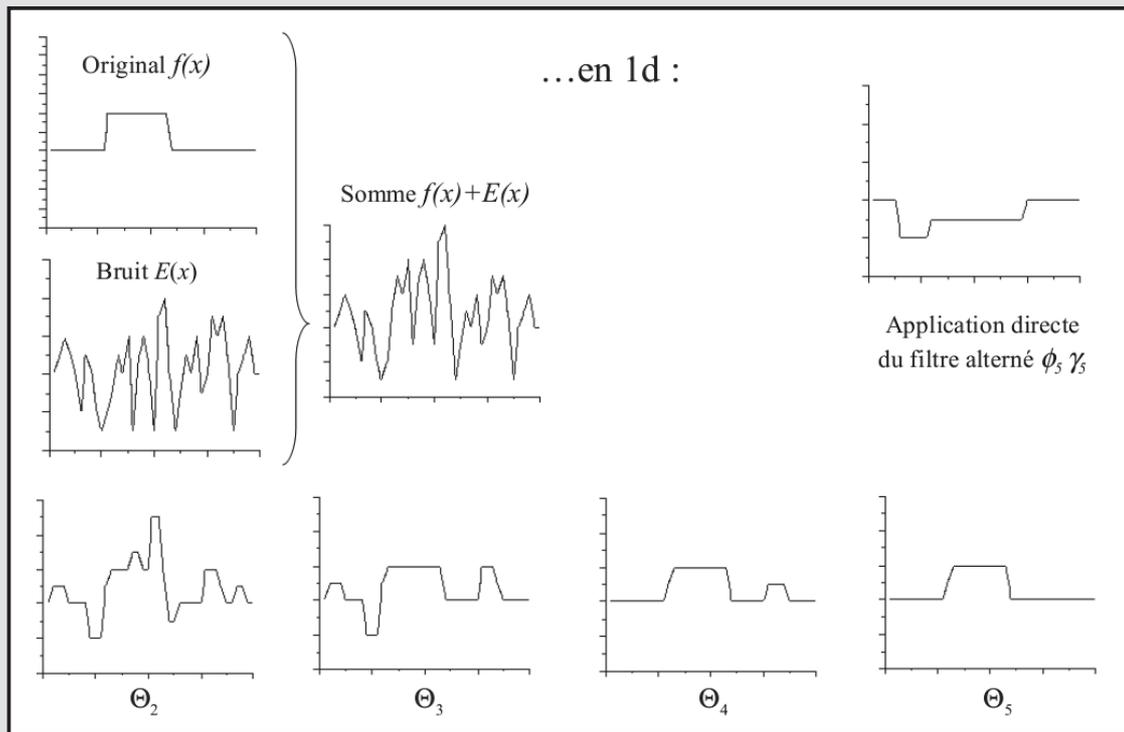


Ξ_5



Ξ_8

Morphologie mathématique



Morphologie mathématique

■ Contexte

- Définition
- Principe
- Propriétés

■ Les opérateurs classiques

- Dilatation / Erosion
- Gradient / Laplacien morphologique
- Ouverture / Fermeture
- Filtres alternés séquentiels
- Autres

Morphologie mathématique

■ Autres

▶ Filtres connexes

Éliminent les structures en préservant les contours

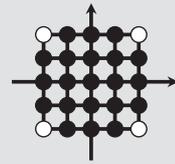


Image initiale



Ouverture par reconstruction

Morphologie mathématique

■ Autres

▶ Filtres géodésiques

Basée sur la longueur du plus court chemin entre deux points (mesures géodésiques)

▶ Application: étiquetage

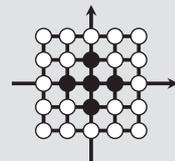
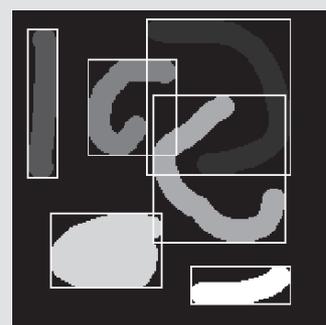


Image initiale



étiquetage

Morphologie mathématique

Autres

► Filtres géodésiques

Application: bouchage de trous

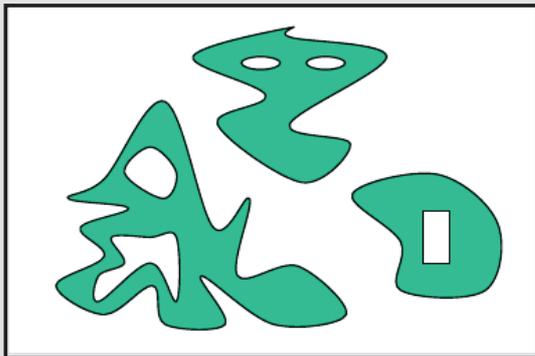
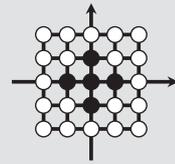


Image initiale

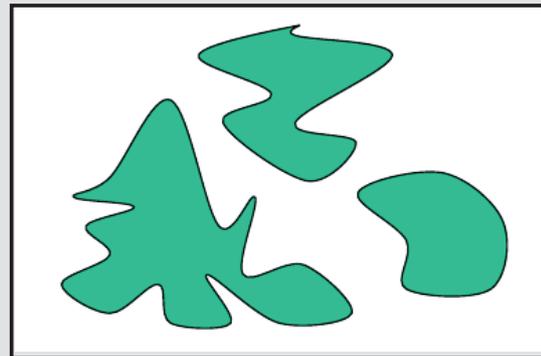


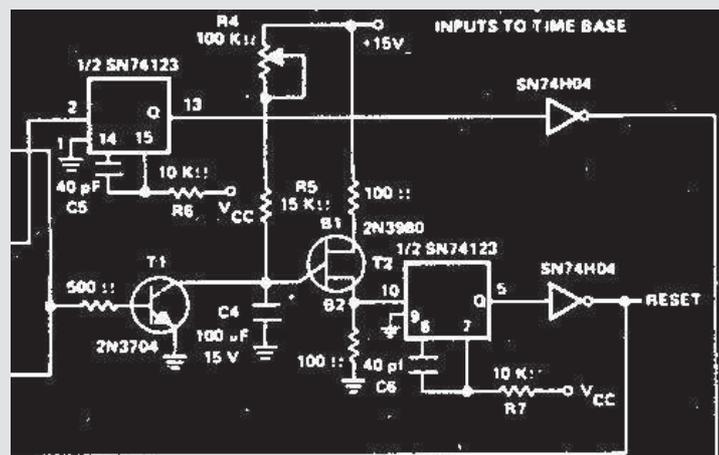
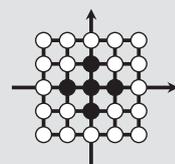
Image traitée

Morphologie mathématique

Autres

► Filtres d'ouvertures de région

Exemple: supprimer du texte (objets de taille inférieure à un seuil)



► **Code matlab correspondant – suppression texte**

```
% Lecture image
in = imread('carte_piste_avant.jpg');
in = rgb2gray(in);
% Seuillage
level = graythresh(in);
BW = im2bw(in,level);
% Suppression regions inf. a 200 pixels
BW2 = bwareaopen(BW,200);
```

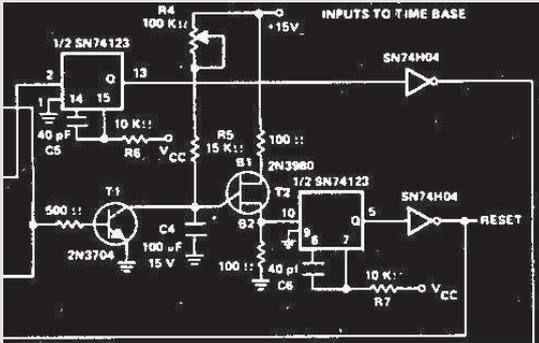


Image initiale

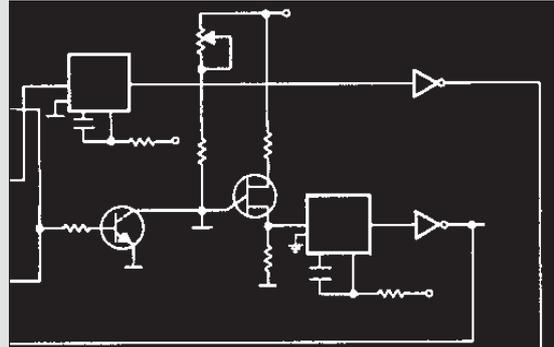
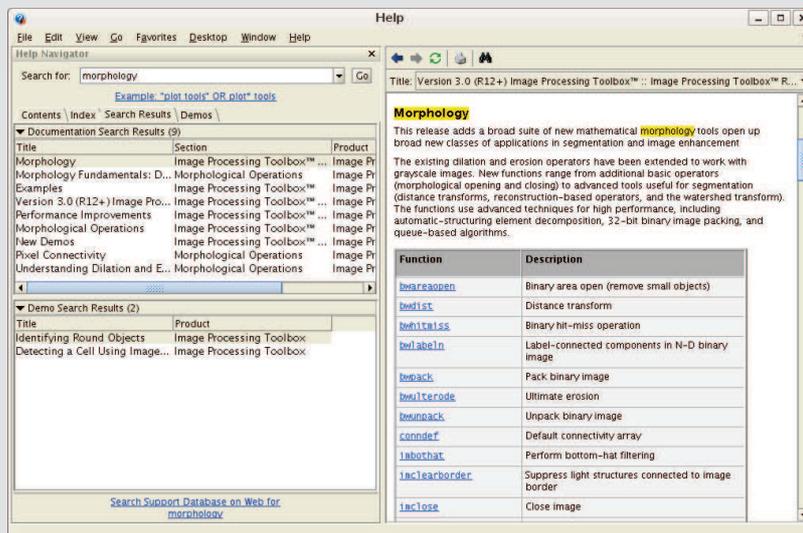


Image traitée

Morphologie mathématique

■ Si vous êtes intéressé par le filtrage morphologique

Liste des fonctions de morphologie mathématique accessibles dans la documentation matlab en écrivant « morphology » dans l'onglet content



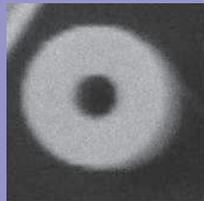
Sommaire

V. Analyse d'images

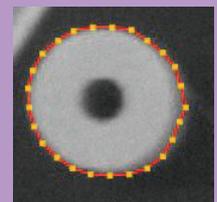
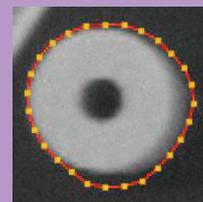
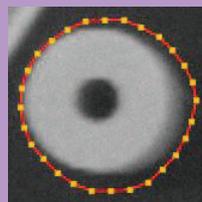
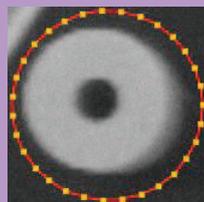
- ✓ **1^{ère} partie**
 - Morphologie mathématique
 - **Détection et analyse de contours**
- ✓ **2^{ème} partie**
 - Segmentation par régions
- ✓ **3^{ème} partie**
 - Analyse de formes
 - Reconnaissance de formes

Détection et analyse de contours

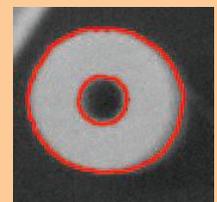
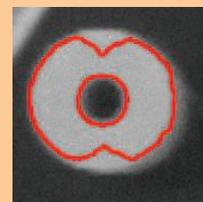
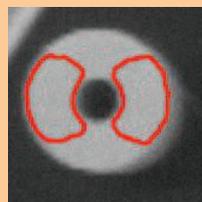
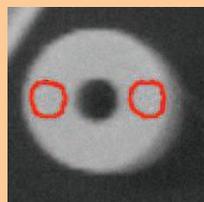
Méthodes classiques



Méthodes « snake »



Méthodes géodésiques



Détection et analyse de contours

■ Contexte

- Définition
- Modèle continu de contour
- Les familles d'analyse de contours

■ Les approches classiques

- Préfiltrage
- Détection de contours
- Extraction de contours
- Poursuite et fermeture de contours

■ Les contours actifs (snakes)

■ Les modèles géodésiques actifs

Détection et analyse de contours

■ Contexte

- Définition
- Modèle continu de contour
- Les familles d'analyse de contours

■ Les approches classiques

- Préfiltrage
- Détection de contours
- Extraction de contours
- Poursuite et fermeture de contours

■ Les contours actifs (snakes)

■ Les modèles géodésiques actifs

Détection et analyse de contours

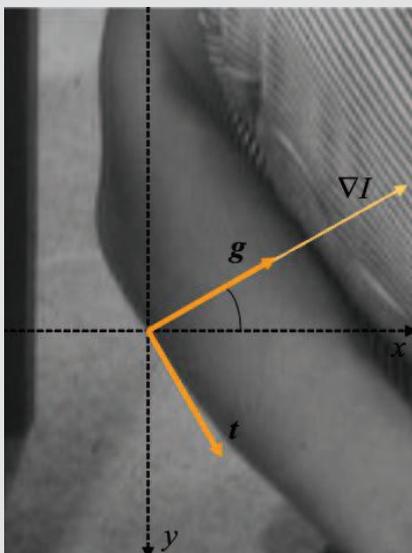
■ Contexte - définition

La détection de contours est une technique de réduction d'information dans les images

Technique qui consiste à transformer l'image en un ensemble de courbes, pas forcément fermées, formant les frontières significatives de l'image.

Détection et analyse de contours

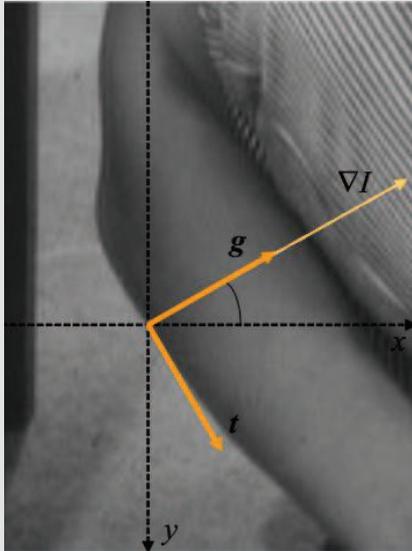
■ Contexte – modèle continu



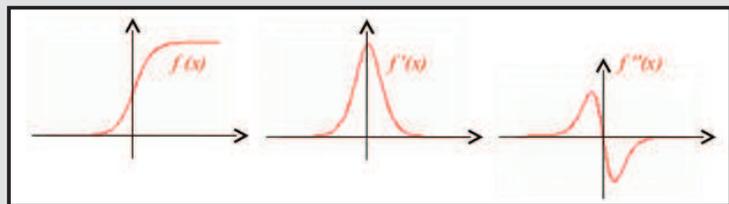
- ▶ Soit une image continue $I(x, y)$
- ▶ Un contour apparait comme une ligne où sont localisées les très fortes variations de $I(x, y)$

Détection et analyse de contours

■ Contexte – modèle continu

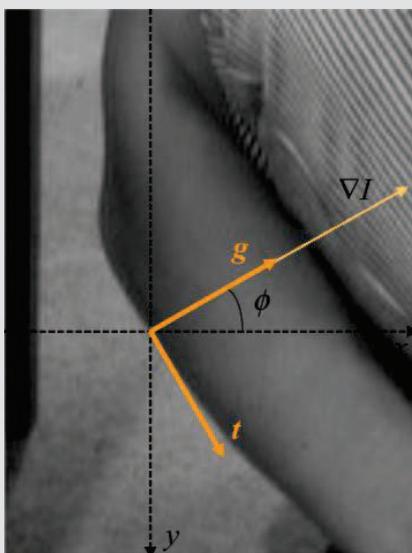


- ▶ En 1D, un contour correspond à un maximum de la dérivée première, c'est-à-dire à un passage par zéro de la dérivée seconde



Détection et analyse de contours

■ Contexte – modèle continu



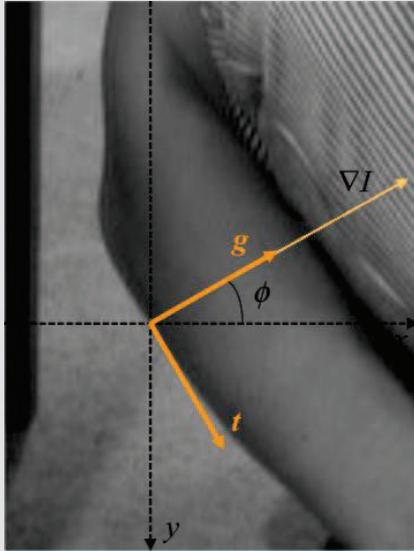
- ▶ Pour retrouver ces propriétés en 2D, il faut se placer dans la direction du gradient

Définition du gradient

$$\left\{ \begin{array}{l} \|\vec{\nabla}_I\|_2 = \left[\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2 \right]^{1/2} \\ \vec{\nabla}_I = \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix}^T \end{array} \right.$$

Détection et analyse de contours

■ Contexte – modèle continu



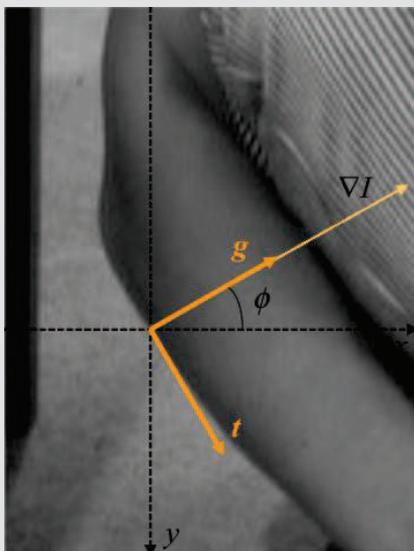
- ▶ En exprimant la fonction image et ses dérivées dans le repère local au gradient (g,t) (Jacobien)

$$\begin{cases} \frac{\partial I}{\partial g} = \frac{\partial I}{\partial x} \cos \phi + \frac{\partial I}{\partial y} \sin \phi \\ \frac{\partial I}{\partial t} = \frac{\partial I}{\partial x} \sin \phi - \frac{\partial I}{\partial y} \cos \phi \end{cases}$$

avec $\phi = \arctan \left(\frac{\partial I / \partial x}{\partial I / \partial y} \right)$

Détection et analyse de contours

■ Contexte – modèle continu

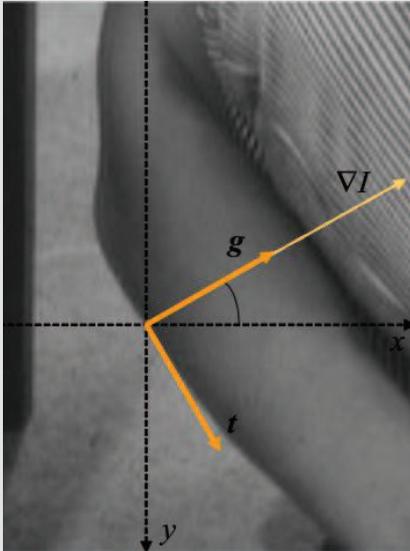


- ▶ En remplaçant dans les deux équations ϕ par son expression on obtient:

$$\begin{cases} \frac{\partial I}{\partial g} = \|\overrightarrow{\nabla I}\|_2 \\ \frac{\partial I}{\partial t} = 0 \end{cases}$$

Détection et analyse de contours

■ Contexte – modèle continu

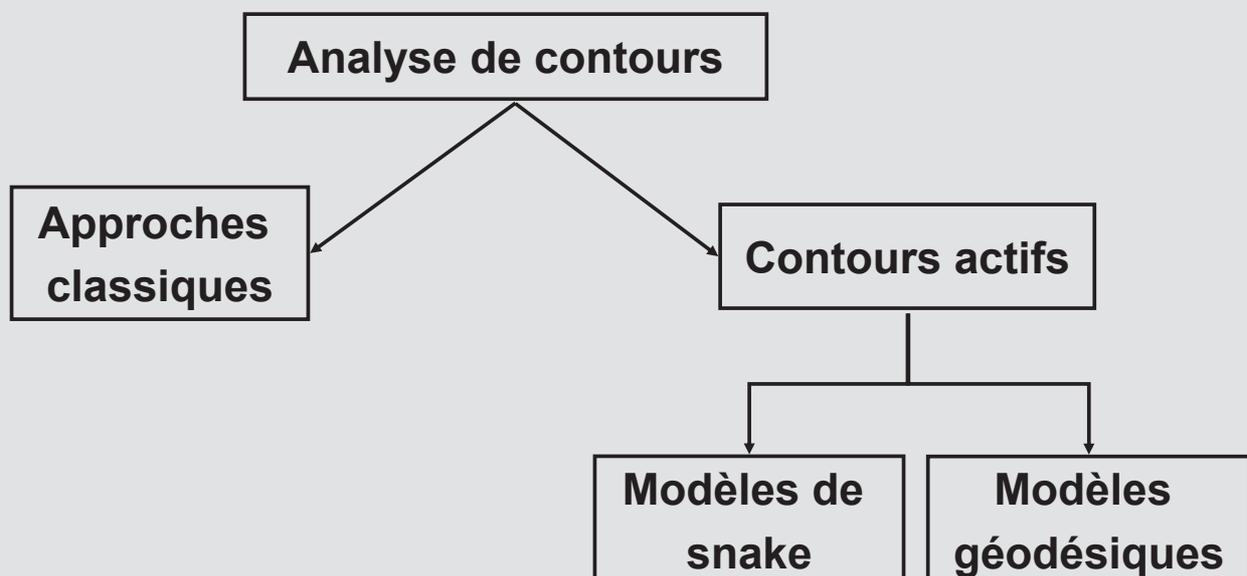


Conclusion

- ▶ Pour retrouver ces propriétés en 2D, il faut se placer dans la direction du gradient
- ▶ Un contour est alors défini comme le lieu des maxima de la dérivée première dans la direction du gradient

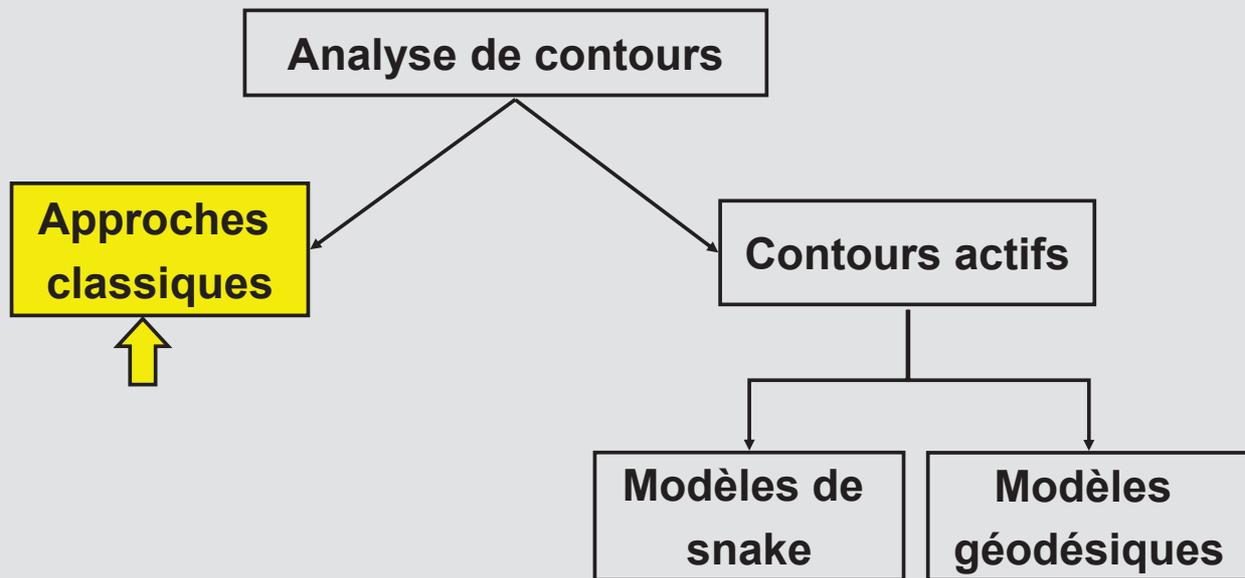
Détection et analyse de contours

■ Contexte – les familles d'analyse de contours



Détection et analyse de contours

■ Contexte – les familles d'analyse de contours



Détection et analyse de contours

■ Contexte

- Définition
- Modèle continu de contours
- Les familles d'analyse de contours

■ Les approches classiques

- Préfiltrage
- Détection de contours
- Extraction de contours
- Poursuite et fermeture de contours

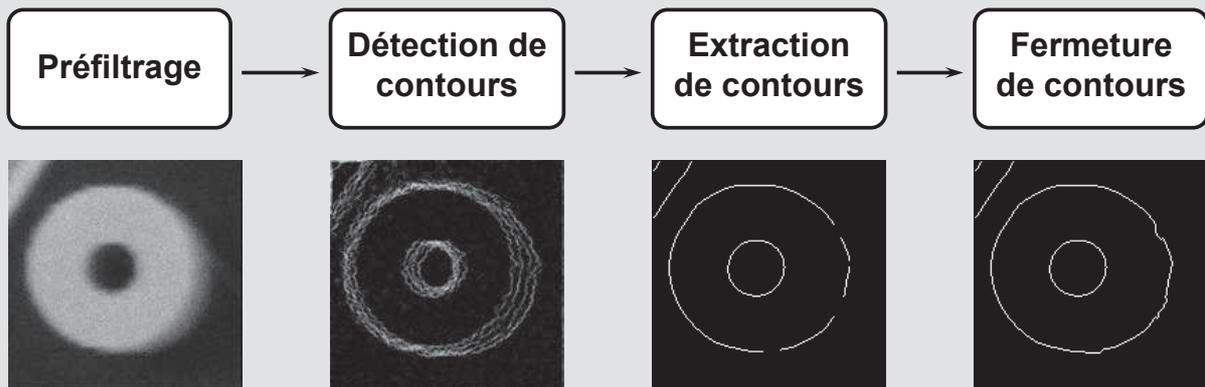
■ Les contours actifs (snakes)

■ Les modèles géodésiques actifs

Détection de contours - approches classiques

■ Démarche générale

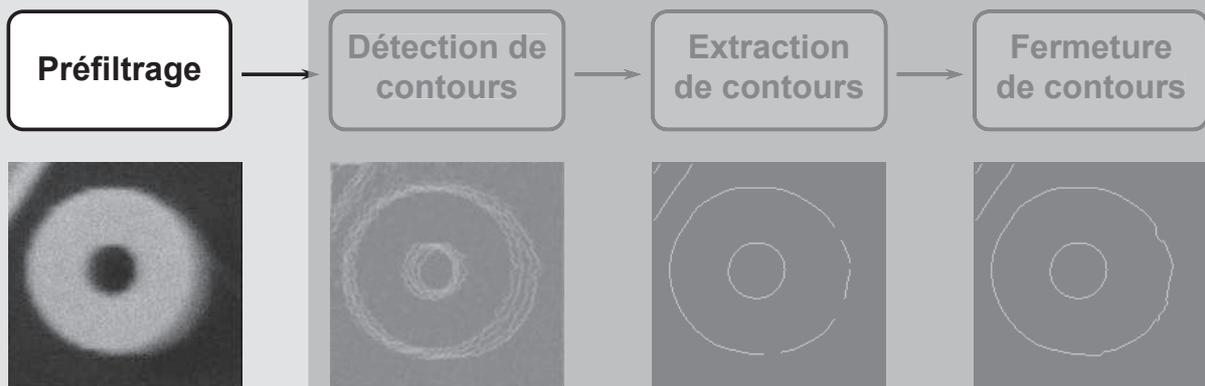
Mise en place d'une chaîne de traitements afin de définir un ensemble connexe de points caractérisant le contour détecté



Détection de contours - approches classiques

■ Démarche générale

Mise en place d'une chaîne de traitements afin de définir un ensemble connexe de points caractérisant le contour détecté



Détection de contours - approches classiques

■ Préfiltrage

Le but est de diminuer le bruit tout en préservant les contours

- ▶ Filtre de lissage, médian, méthode des mean-shift,...



Image initiale



Image obtenue après filtrage médian

Détection et analyse de contours

■ Contexte

- Définition
- Modèle continu de contours
- Les familles d'analyse de contours

■ Les approches classiques

- Préfiltrage
- Détection de contours
- Extraction de contours
- Poursuite et fermeture de contours

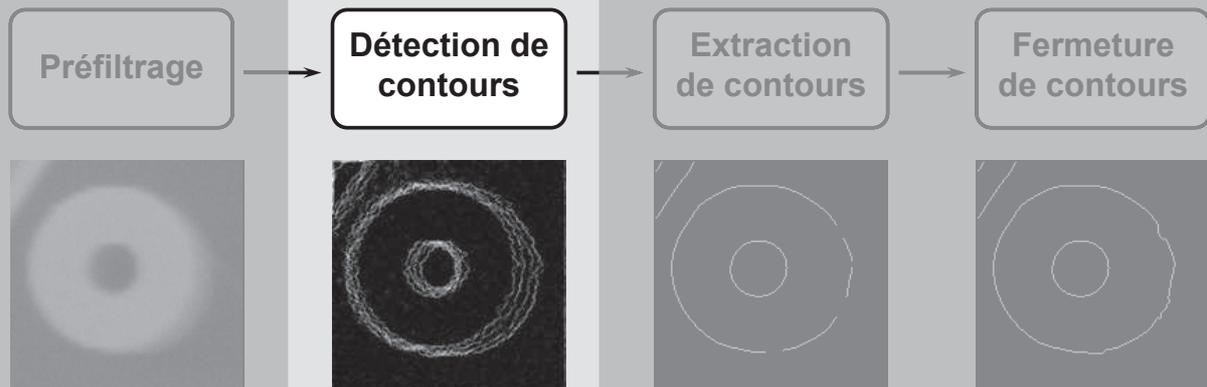
■ Les contours actifs (snakes)

■ Les modèles géodésiques actifs

Détection de contours - approches classiques

■ Démarche générale

Mise en place d'une chaîne de traitements afin de définir un ensemble connexe de points caractérisant le contour détecté



Détection de contours - approches classiques

■ Détection de contours

- ▶ Il existe trois grandes catégories:
 - Détection de contours par filtrage
 - Détection de contours par masques
 - Détection de contours par passage par zéro

Détection de contour – approches classiques

■ Détection de contours par filtrage

La théorie de filtrage des signaux s'applique aux images

- ▶ Exemple: filtre passe haut d'une image par filtre de Butterworth d'ordre n

Détection de contour – approches classiques

■ Détection de contours par filtrage

- ▶ Exemple: filtre passe haut d'une image par fonction de Butterworth d'ordre n

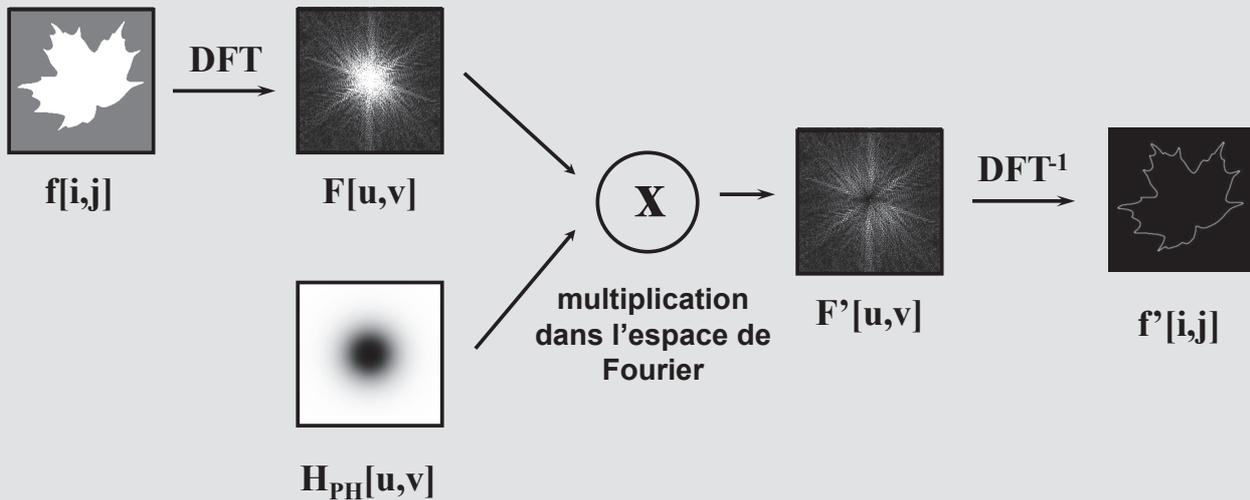
$$\underbrace{\left|H_{PH}(f)\right|^2 = \frac{1}{1 + \left(\frac{f_C}{f}\right)^{2n}}}_{\text{Fonction de transfert 1D}} \quad \longrightarrow \quad \underbrace{\left|H_{PH}(f_x, f_y)\right|^2 = \frac{1}{1 + \left(\frac{f_C}{\sqrt{f_x^2 - f_{cx}^2 + f_y^2 - f_{cy}^2}}\right)^{2n}}}_{\text{Fonction de transfert 2D centrée en } (f_{cx}, f_{cy})}$$

Fonction de transfert 1D

Fonction de transfert 2D centrée en (f_{cx}, f_{cy})

Détection de contour – approches classiques

- ▶ Exemple: filtre passe haut d'une image par fonction de Butterworth d'ordre n



Détection de contour – approches classiques

- ▶ Exemple matlab



Image initiale

```
% Lecture image
img = imread('ellipse.png');
img = double(img(:,:,1));

% parameter
F_cut = 50; cx = 128;
cy = 128; ButtOrd = 2;
[dimI,dimJ] = size(img);

% Calcul des fft
imf=fftshift(fft2(img));
H=zeros(dimI,dimJ);
for i=1:dimI
    for j=1:dimJ
        d=sqrt((i-cx)^2+(j-cy)^2);
        H(i,j)=sqrt(1/(1+...
            ((F_cut/d)^(2*ButtOrd))));
    end
end

% Extraction du contour
outf = imf .* H;

% Calcul des ifft
out = abs(ifft2(outf));
```

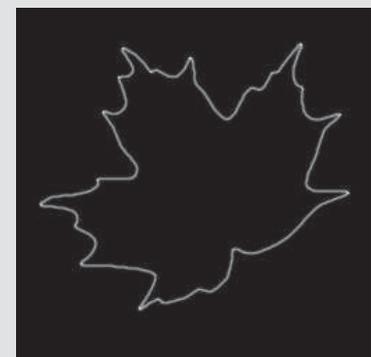


Image des hautes fréquences

Détection de contours - approches classiques

■ Détection de contours

- ▶ Il existe trois grandes catégories:
 - Détection de contours par filtrage
 - Détection de contours par masques
 - Détection de contours par passage par zéro

Détection de contour – approches classiques

■ Détection de contours par masques

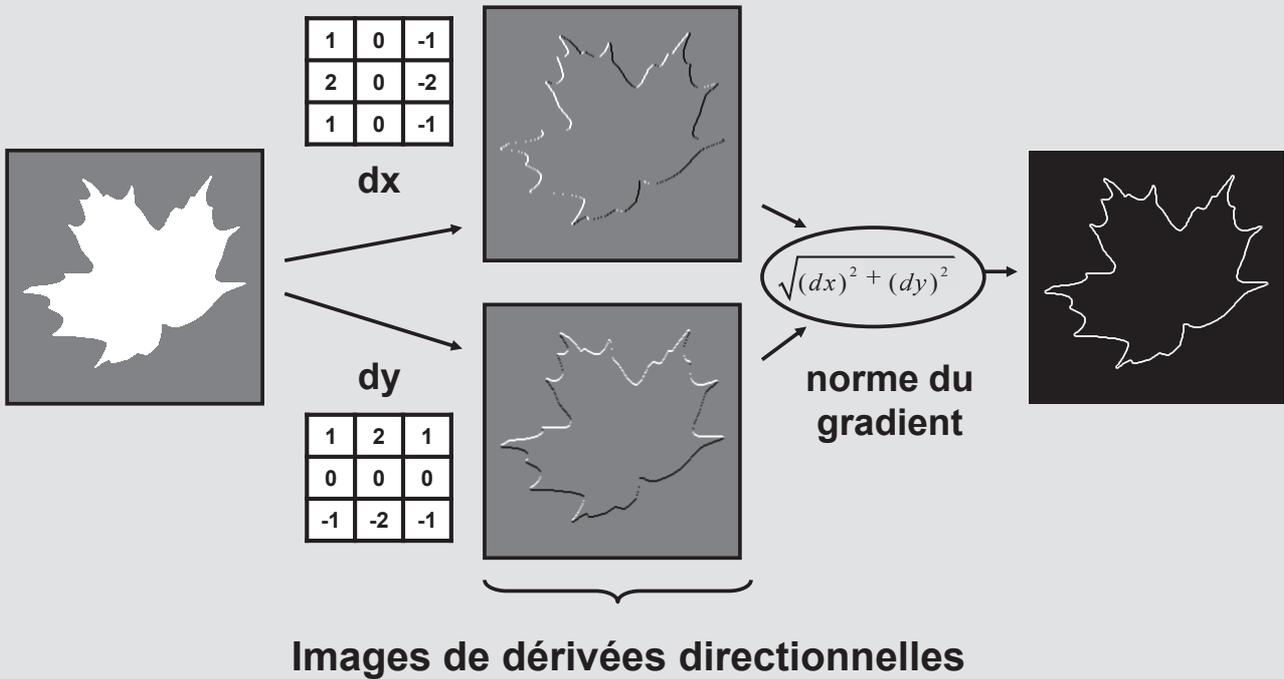
Proviennent d'estimateurs locaux des dérivées discrètes d'une image

- ▶ Correspondent à la convolution d'une image par un filtre 2D de réponse impulsionnelle finie

$$f_{new}[i, j] = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} f[k, l] \cdot h[i - k, j - l]$$

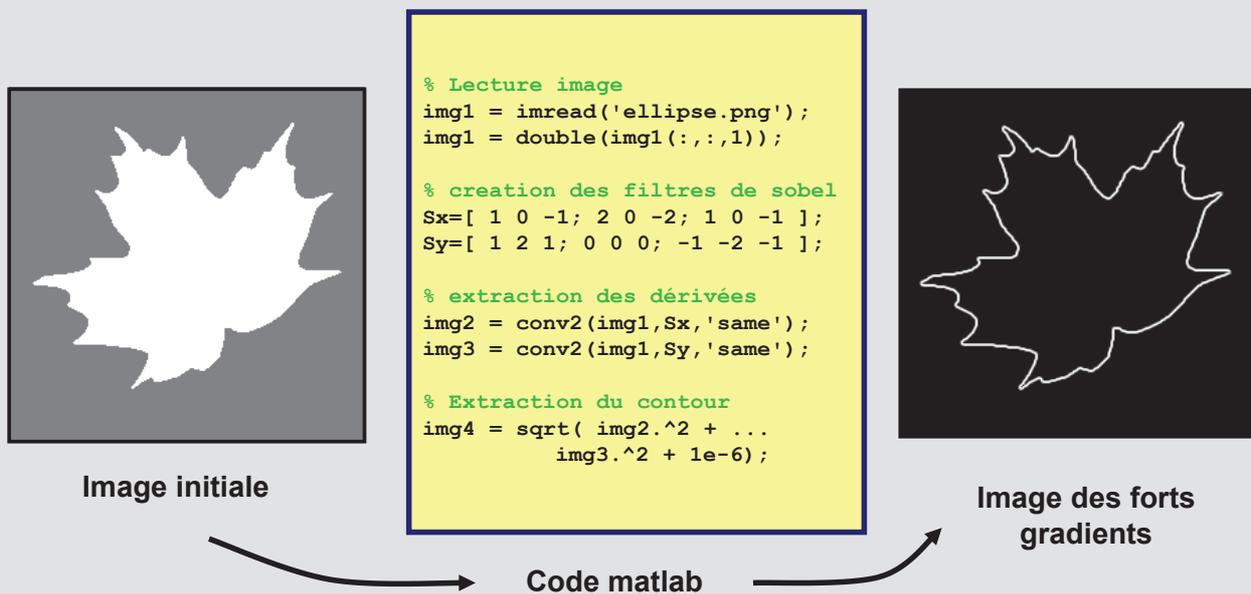
Détection de contour – approches classiques

► Exemple: filtre de Sobel



Détection de contour – approches classiques

► Exemple matlab: filtre de Sobel



Détection de contours - approches classiques

■ Détection de contours

- ▶ Il existe trois grandes catégories:
 - Détection de contours par filtrage
 - Détection de contours par masques
 - **Détection de contours par passage par zéro**

Détection de contours - approches classiques

■ Détection de contours par passage par zéro

- ▶ Utilisation de l'opérateur Laplacien



Image initiale



Laplacien de l'image

Détection et analyse de contours

■ Contexte

- Définition
- Modèle continu de contours
- Les familles d'analyse de contours

■ Les approches classiques

- Préfiltrage
- Détection de contours
- Extraction de contours
- Poursuite et fermeture de contours

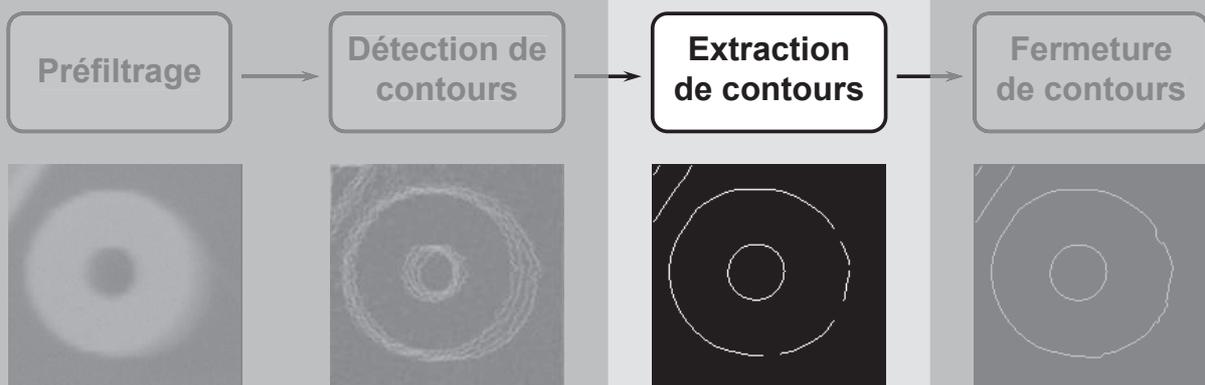
■ Les contours actifs (snakes)

■ Les modèles géodésiques actifs

Détection de contours - approches classiques

■ Démarche générale

Mise en place d'une chaîne de traitements afin de définir un ensemble connexe de points caractérisant le contour détecté



Détection de contours - approches classiques

■ Extraction de contours par seuillage

- ▶ Généralement l'étape de détection fournit trop d'information de contour
- ▶ Le but est donc de conserver que les composantes principales des contours détectées lors de l'étape précédente
- ▶ Utilisation de méthodes de seuillage

Détection de contours - approches classiques

■ Extraction de contours par seuillage

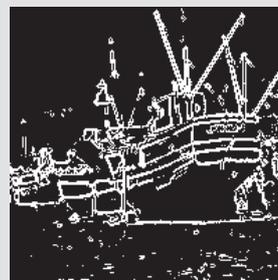
- ▶ Exemple: seuillage de la norme du gradient



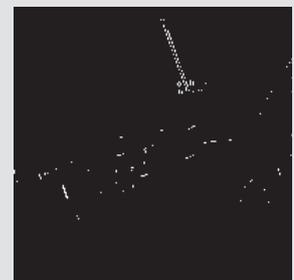
Image initiale



Norme du gradient



Norme du gradient
seuillée à 25



Norme du gradient
seuillée à 70

Méthodes très sensibles au paramètres de seuillage

Détection de contours - approches classiques

■ Extraction de contours par seuillage

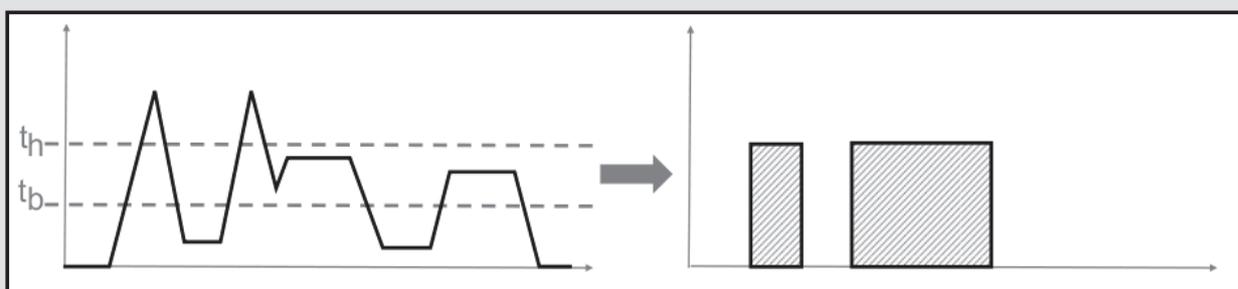
▶ Exemple: seuillage par hystérésis

Le but est de diminuer l'importance du seuil lors du seuillage du gradient

Détection de contours - approches classiques

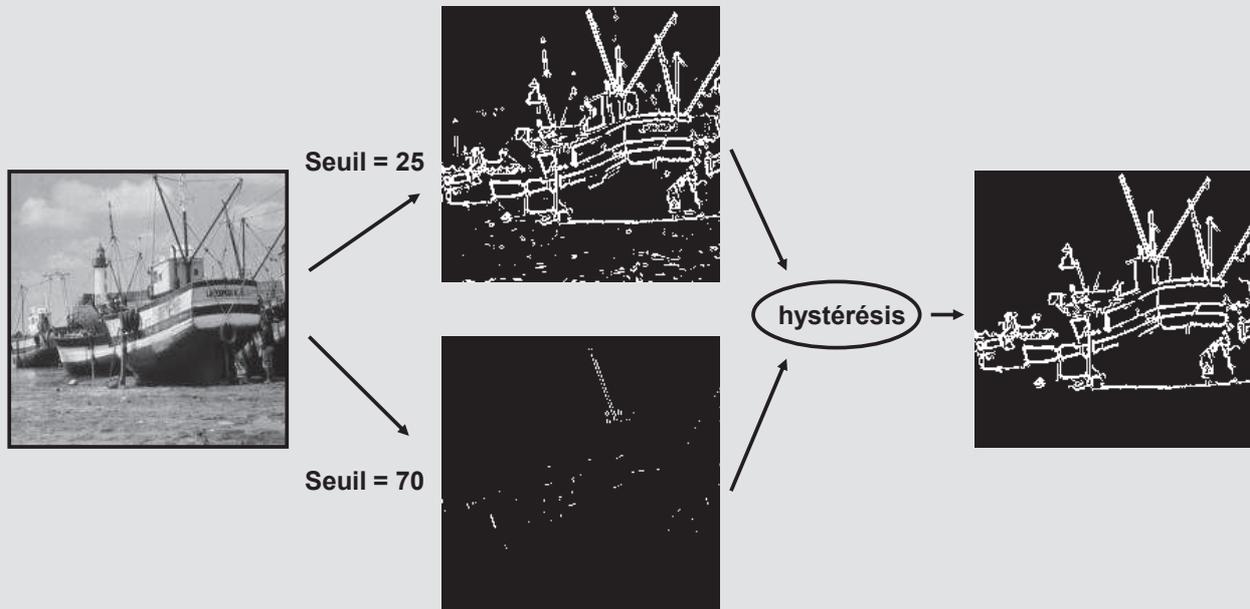
▶ Exemple: seuillage par hystérésis - Principe

- Deux seuils : un seuil haut t_h , un seuil bas t_b
- On sélectionne d'abord les points au dessus de t_h
- Puis on applique le seuil t_b en ne gardant que les composantes connexes qui contiennent au moins un point au dessus de t_h



Détection de contours - approches classiques

► Exemple: seuillage par hystérésis



Détection de contours - approches classiques

► Exemple matlab: seuillage par hystérésis (connexité 4)

```
function out = hysteresis(in1,in2)

out = in2; count = 1; k=0;
MAXITERATION = 200;
while ( ( count ~= 0 ) && ( k < MAXITERATION ) )
    count = 0;
    for i=2:(size(out,1)-1)
        for j=2:(size(out,2)-1)
            if ( out(i,j) > 0 )
                if ( in1(i-1,j) > 0 )
                    out(i-1,j) = 255; count = count + 1;
                end
                if ( in1(i+1,j) > 0 )
                    out(i+1,j) = 255; count = count + 1;
                end
                if ( in1(i,j-1) > 0 )
                    out(i,j-1) = 255; count = count + 1;
                end
                if ( in1(i,j+1) > 0 )
                    out(i,j+1) = 255; count = count + 1;
                end
            end
        end
    end
    k = k + 1;
end
```

Détection et analyse de contours

■ Contexte

- Définition
- Modèle continu de contours
- Les familles d'analyse de contours

■ Les approches classiques

- Préfiltrage
- Détection de contours
- Extraction de contours
- Poursuite et fermeture de contours

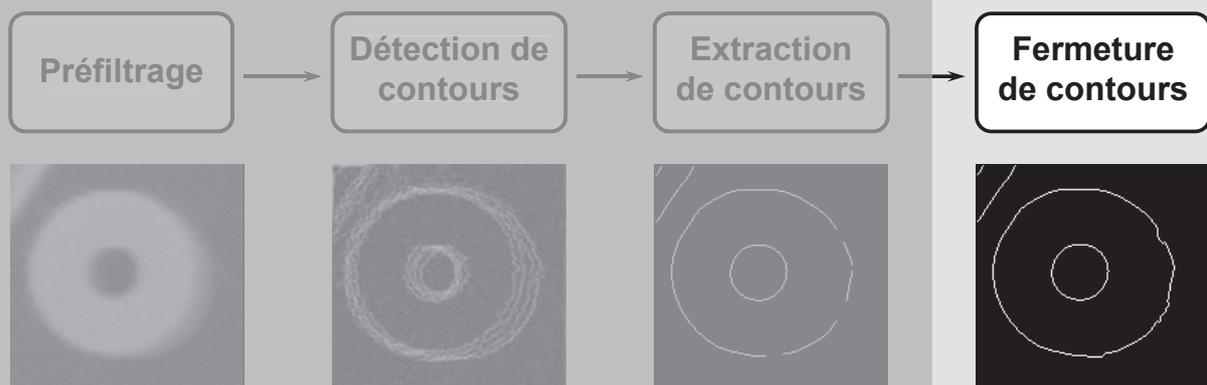
■ Les contours actifs (snakes)

■ Les modèles géodésiques actifs

Détection de contours - approches classiques

■ Démarche générale

Mise en place d'une chaîne de traitements afin de définir un ensemble connexe de points caractérisant le contour détecté



Détection de contours - approches classiques

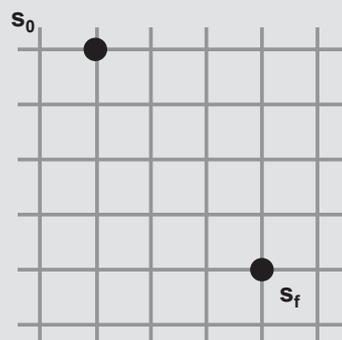
■ Poursuite et fermeture de contours

- ▶ Souvent, l'étape de conservation des contours importants n'est pas suffisante pour détecter des objets topologiquement clos
- ▶ Ajout d'étapes de poursuite et de fermeture de contours

Détection de contours - approches classiques

■ Poursuite et fermeture de contours

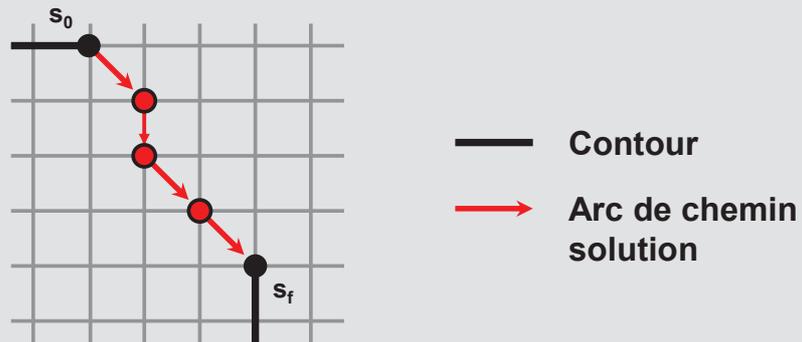
- ▶ Les méthodes de recherche dans des graphes
 - L'image numérique peut être associée à un maillage ou un graphe) où chaque sommet est un site s_i ou un pixel



Détection de contours - approches classiques

■ Les méthodes de recherche dans des graphes

La fermeture correspond à la recherche d'un chemin entre les sommets du graphe associés aux deux sites s_0 et s_f qui sont les extrémités de la lacune à combler



Détection de contours - approches classiques

■ Les méthodes de recherche dans des graphes

► Comment trouver un chemin ?

- Définition d'une fonction coût qui peut dépendre:
 - d'une énergie d'attache aux données en chaque sommet
 - de la longueur du contour (longueur minimale)
 - de la courbure du contour (courbure minimale)
- Le chemin optimal sera le chemin ayant un coût minimal (ou maximal suivant la fonction utilisée)

Détection de contours - approches classiques

- ▶ **Exemple: contours prolongés le long des lignes de crête du gradient**
 - **Extraction préalable des contours par un opérateur donnant des contours d'une largeur d'un pixel et fournissant la norme du gradient en chaque pixel**
 - **Extraire les contours « sûrs » en fixant un seuil suffisamment élevé pour ne pas détecter le bruit**
 - **A partir de chaque extrémité du contour, élaborer un chemin qui suit les lignes de crête dans l'image de la norme du gradient**

Détection de contours - approches classiques

- ▶ **Exemple: contours prolongés le long des lignes de crête du gradient**
 - **1^{ère} étape : détection de contours « sûrs »**

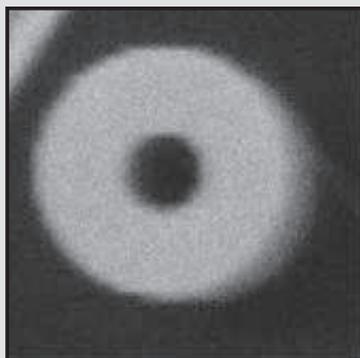


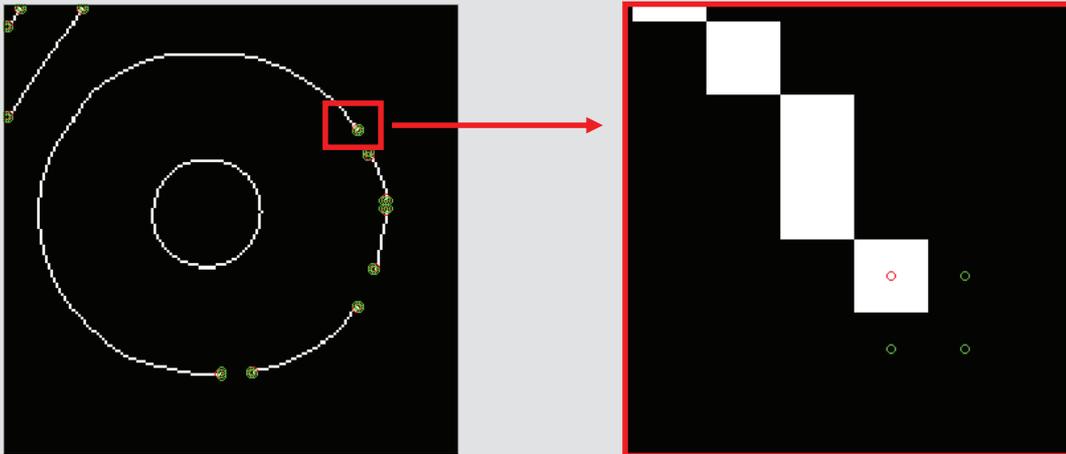
Image initiale



Image obtenue par contour multi-échelles

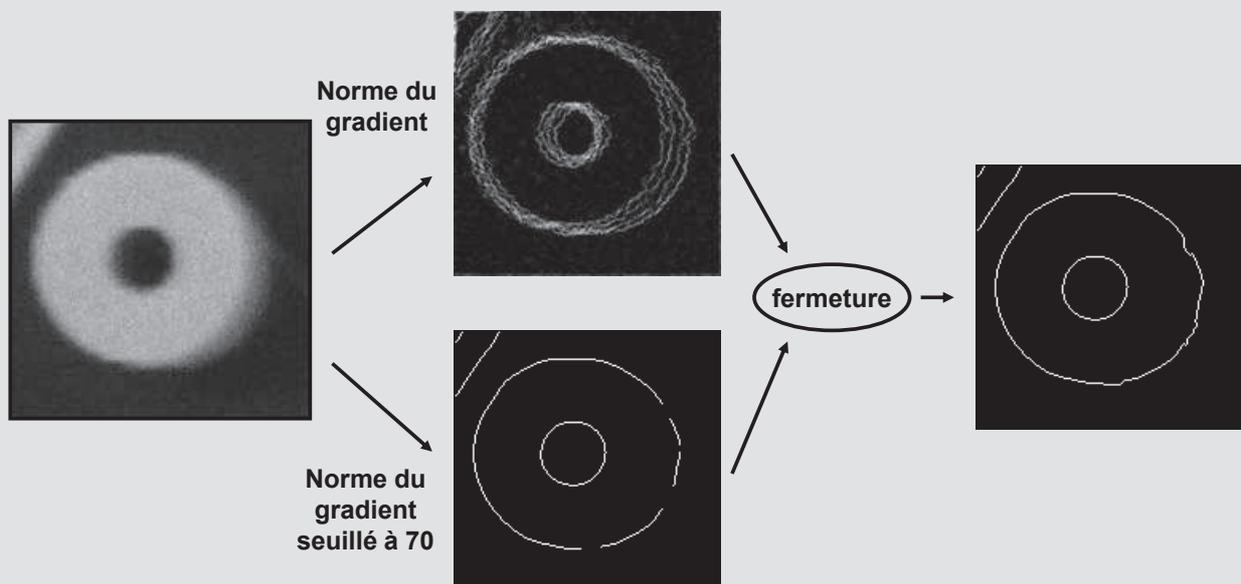
Détection de contours - approches classiques

- ▶ Exemple: contours prolongés le long des lignes de crête du gradient
 - 2^{ème} étape : détection de points extrêmes



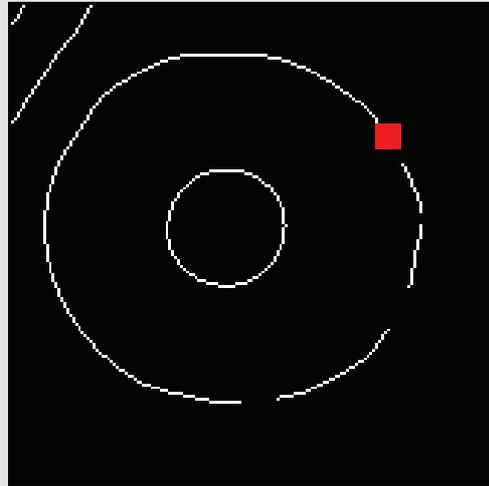
Détection de contours - approches classiques

- ▶ Exemple: contours prolongés le long des lignes de crête du gradient



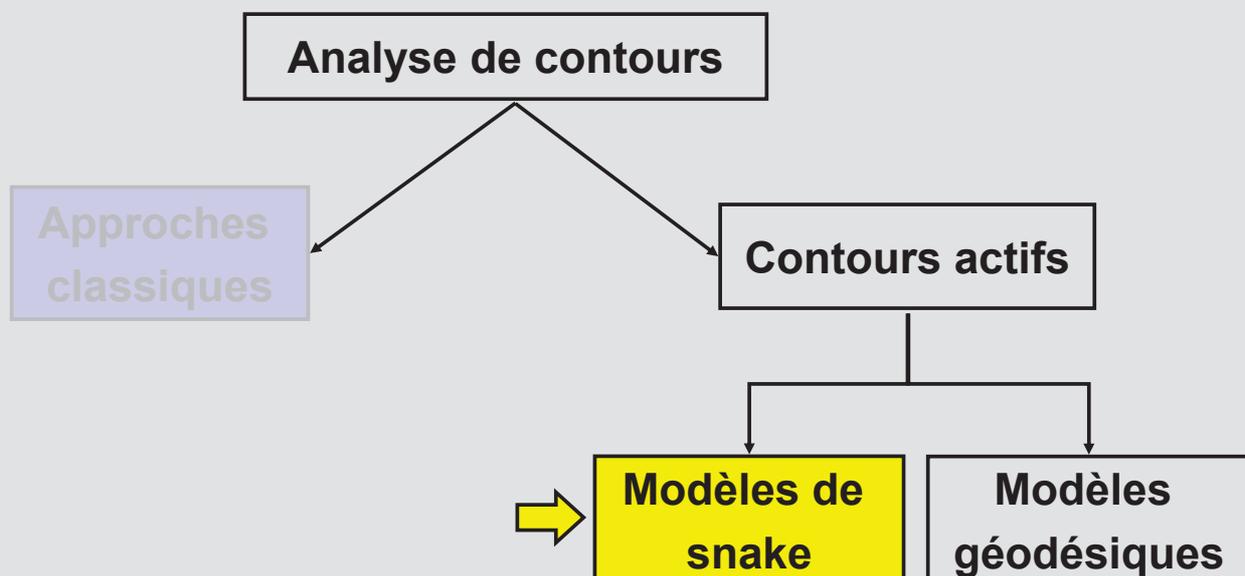
Détection de contours - approches classiques

- ▶ Exemple: contours prolongés le long des lignes de crête du gradient



Détection de contours - approches classiques

■ Les familles d'analyse de contours



Détection et analyse de contours

■ Contexte

- Définition
- Modèle continu de contours
- Les familles d'analyse de contours

■ Les approches classiques

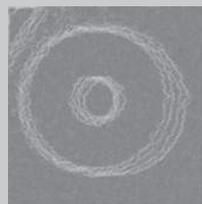
- Préfiltrage
- Détection de contours
- Extraction de contours
- Poursuite et fermeture de contours

■ Les contours actifs (snakes)

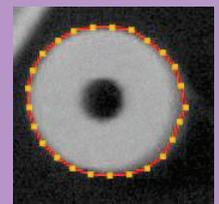
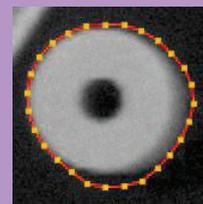
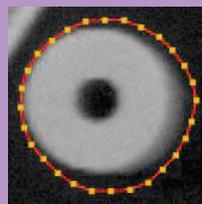
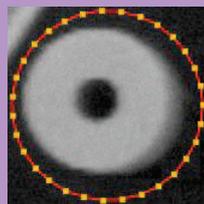
■ Les modèles géodésiques actifs

Détection et analyse de contours

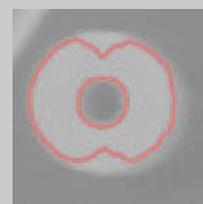
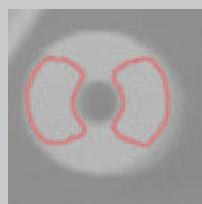
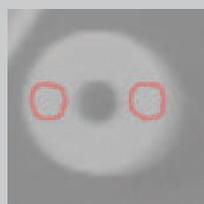
Méthodes classiques



Méthodes « snake »



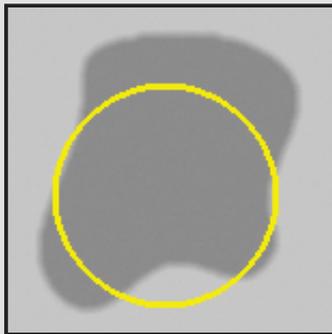
Méthodes géodésiques



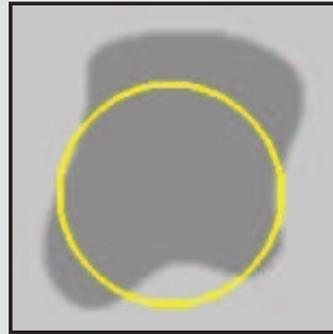
Détection de contours - contours actifs (snakes)

■ Principe

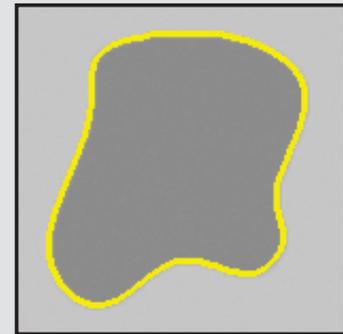
Evolution d'une courbe dans une image dont l'état final définit les contours de l'objet détecté



initialisation



évolution



à convergence

Détection de contours - contours actifs (snakes)

■ Comment faire évoluer le contour actif ?

▶ Modélisation mathématique

- 1 - choix de la représentation du contour actif**
- 2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter**
- 3 - évolution du contour guidée par la minimisation de la fonction d'énergie**

Détection de contours - contours actifs (snakes)

■ Comment faire évoluer le contour actif ?

► Modélisation mathématique

1 - choix de la représentation du contour actif

2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter

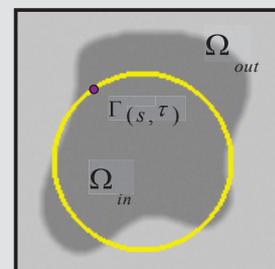
3 - évolution du contour guidée par la minimisation de la fonction d'énergie

Détection de contours - contours actifs (snakes)

► Choix de la représentation du contour actif

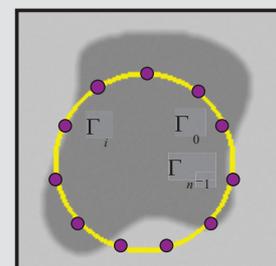
Représentation continue

- $\Gamma(s, \tau) : [0, 1] \times [0, \infty[\rightarrow \mathbb{R}^2$
- $\Omega_{in}, \Omega_{out}$ région intérieure et extérieure à Γ



Représentation discrète

- Le contour est échantillonné
- $\Gamma(\tau) = [\Gamma_0^t(\tau), \Gamma_1^t(\tau), \dots, \Gamma_{n-1}^t(\tau)]^t$
- avec $\Gamma_i(\tau) = x_i(\tau), y_i(\tau)^t$



Détection de contours - contours actifs (snakes)

■ Comment faire évoluer le contour actif ?

▶ Modélisation mathématique

1 - choix de la représentation du contour actif

2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter

3 - évolution du contour guidée par la minimisation de la fonction d'énergie

Détection de contours - contours actifs (snakes)

▶ Conception d'une fonction d'énergie

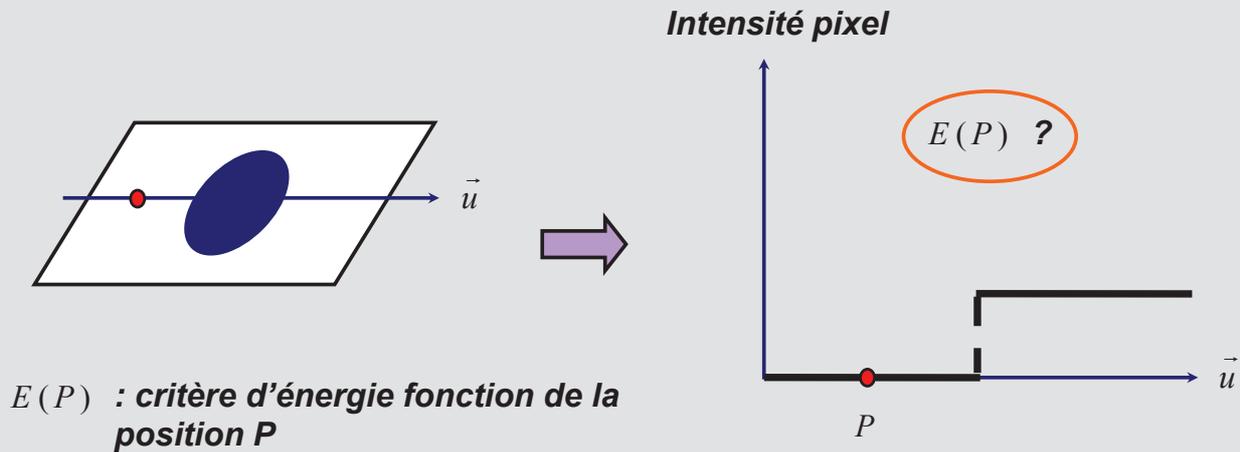
Qu'est ce qu'un critère d'énergie ?

- Fonction positive (appelée par exemple E)
- Fonction dérivable
- Fonction dont le minimum délimite les contours de l'objet à détecter
- Le minimum correspond généralement à $E = 0$

Détection de contours - contours actifs (snakes)

► Conception d'une fonction d'énergie

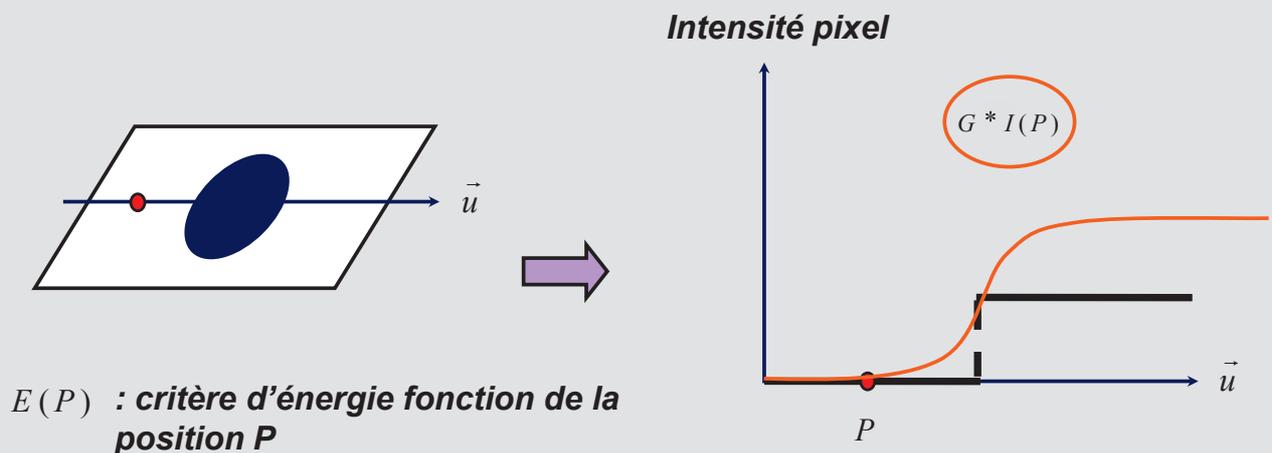
Illustration: étude 1D



Détection de contours - contours actifs (snakes)

► Conception d'une fonction d'énergie

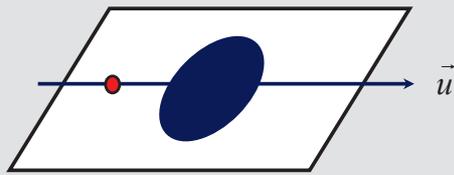
Illustration: étude 1D



Détection de contours - contours actifs (snakes)

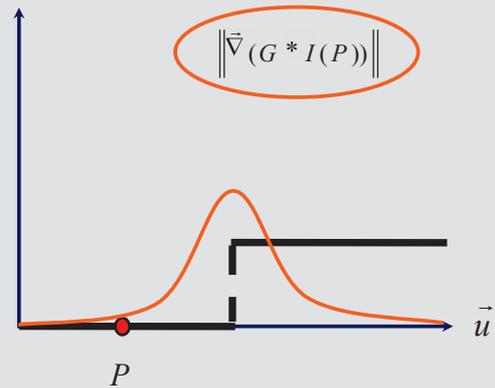
► Conception d'une fonction d'énergie

Illustration: étude 1D



$E(P)$: critère d'énergie fonction de la position P

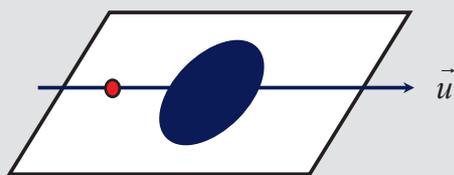
Intensité pixel



Détection de contours - contours actifs (snakes)

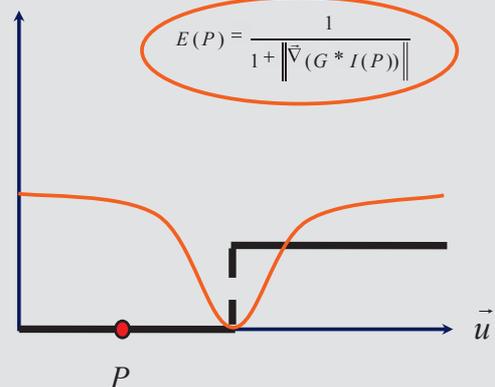
► Conception d'une fonction d'énergie

Illustration: étude 1D



$E(P)$: critère d'énergie fonction de la position P

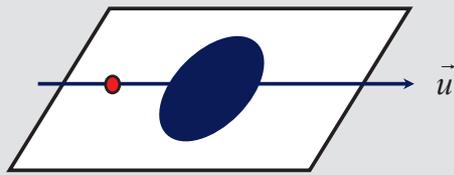
Intensité pixel



Détection de contours - contours actifs (snakes)

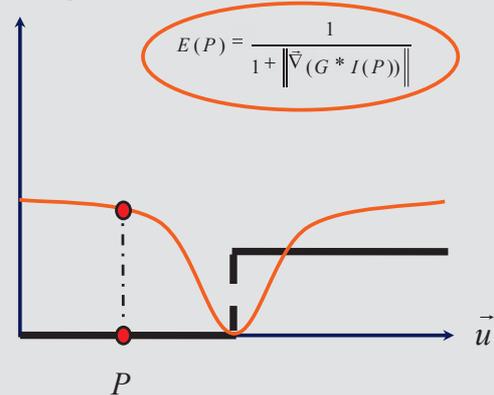
► Conception d'une fonction d'énergie

Illustration: étude 1D



$E(P)$: critère d'énergie fonction de la position P

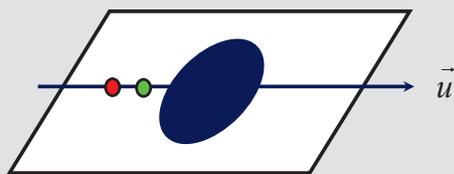
Intensité pixel



Détection de contours - contours actifs (snakes)

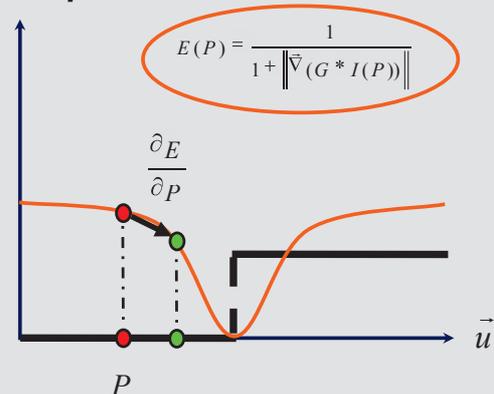
► Conception d'une fonction d'énergie

Illustration: étude 1D



$E(P)$: critère d'énergie fonction de la position P

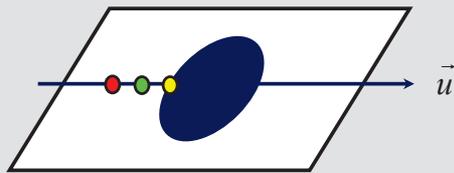
Intensité pixel



Détection de contours - contours actifs (snakes)

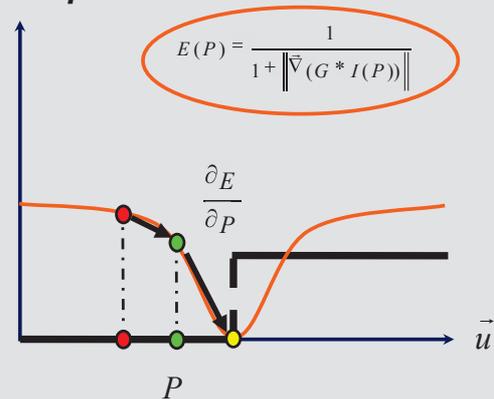
► Conception d'une fonction d'énergie

Illustration: étude 1D



$E(P)$: critère d'énergie fonction de la position P

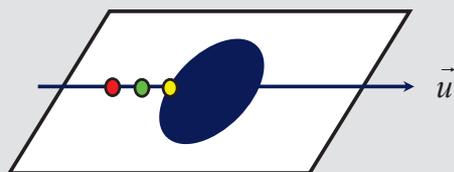
Intensité pixel



Détection de contours - contours actifs (snakes)

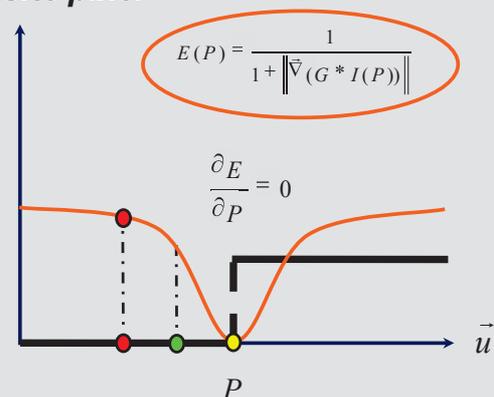
► Conception d'une fonction d'énergie

Illustration: étude 1D



$E(P)$: critère d'énergie fonction de la position P

Intensité pixel



Détection de contours - contours actifs (snakes)

► Conception d'une fonction d'énergie

Fonction d'énergie 2D classique

$$E_{totale} \Gamma = E_{interne} \Gamma + E_{image} \Gamma + E_{externe} \Gamma$$

- $E_{interne} \Gamma$: énergie propre, assimilée à l'énergie mécanique de tension et de torsion de la courbe
- $E_{image} \Gamma$: énergie potentielle imposée par l'image qui vise à plaquer la courbe sur les contours
- $E_{externe} \Gamma$: énergie utilisée pour traduire des contraintes spécifiques au problème (ex: a priori de forme)

Détection de contours - contours actifs (snakes)

► Conception d'une fonction d'énergie

Fonction d'énergie 2D classique – énergie interne

$$E_{interne} \Gamma = \int_0^l \left[\alpha(s) \left| \frac{\partial \Gamma(s)}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 \Gamma(s)}{\partial s^2} \right|^2 \right] ds$$

- agit sur la longueur de Γ
- lié à la rigidité (tension)

- agit sur la courbure de Γ
- lié à l'élasticité (flexion)

Détection de contours - contours actifs (snakes)

► Conception d'une fonction d'énergie

Fonction d'énergie 2D classique – énergie image

$$E_{image}(\Gamma) = \int_0^1 \frac{1}{1 + \|\nabla(G * I(\Gamma(s)))\|} ds$$

- Permet d'attirer la courbe vers des zones de l'image à très fort gradient (\Rightarrow détection de contours)

Détection de contours - contours actifs (snakes)

► Conception d'une fonction d'énergie

Fonction d'énergie 2D classique – énergie externe

- Permet d'introduire des contraintes de haut niveau appliquées interactivement par l'utilisateur, par exemple:
 - \Rightarrow pour placer la courbe sur la région désirée
 - \Rightarrow pour placer des points de contrôle
 - \Rightarrow pour indiquer une partie du contour déjà détectée
 - \Rightarrow uniquement utilisée que dans des cas bien spécifiques

Détection de contours - contours actifs (snakes)

■ Comment faire évoluer le contour actif ?

▶ Modélisation mathématique

1 - choix de la représentation du contour actif

2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter

3 - évolution du contour guidée par la minimisation de la fonction d'énergie

Détection de contours - contours actifs (snakes)

▶ Évolution du contour guidée par la minimisation de la fonction d'énergie

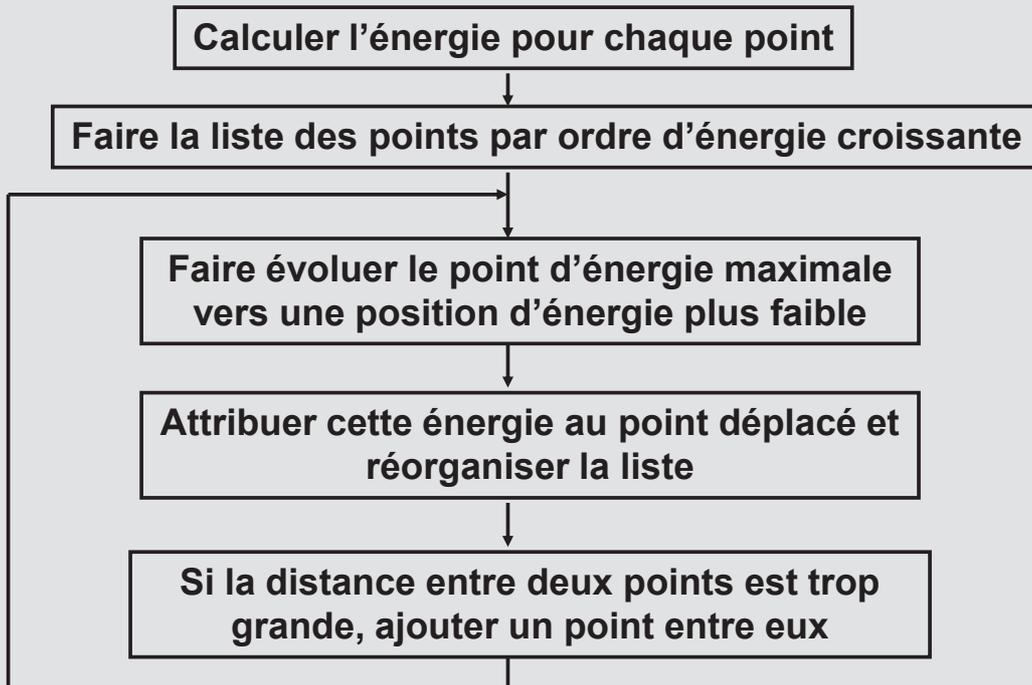
- Cette étape dépend explicitement du mode de représentation du contour

⇒ pour une représentation continue, utilisation d'outils mathématiques adaptés tels que l'équation d'Euler Lagrange permettant de dériver une équation explicite de l'évolution du contour

⇒ pour une représentation discrète, utilisation d'un modèle de graphe pour trouver un successeur à chaque point du contour permettant de diminuer la fonction d'énergie

Détection de contours - contours actifs (snakes)

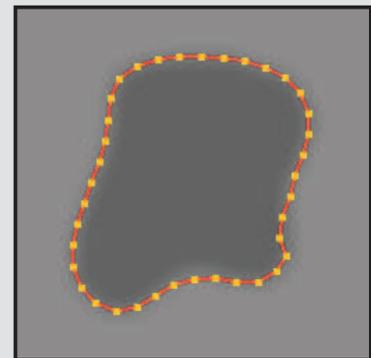
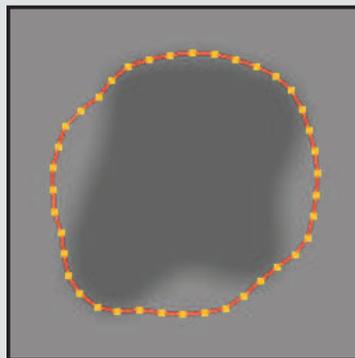
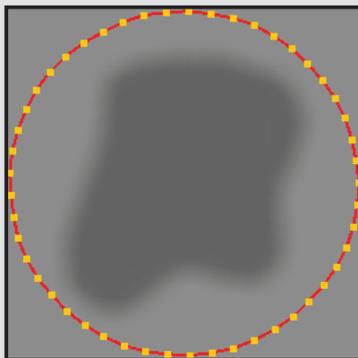
Exemple d'algorithme pour une représentation discrète



Détection de contours - contours actifs (snakes)

Exemple d'algorithme pour une représentation discrète

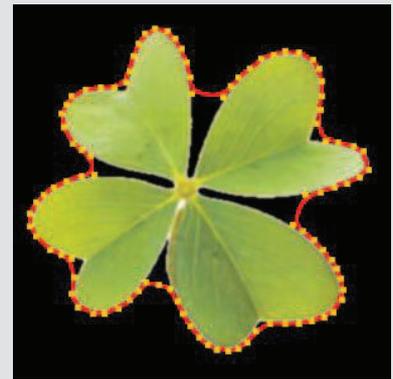
- Illustration sur une image de simulation



Détection de contours - contours actifs (snakes)

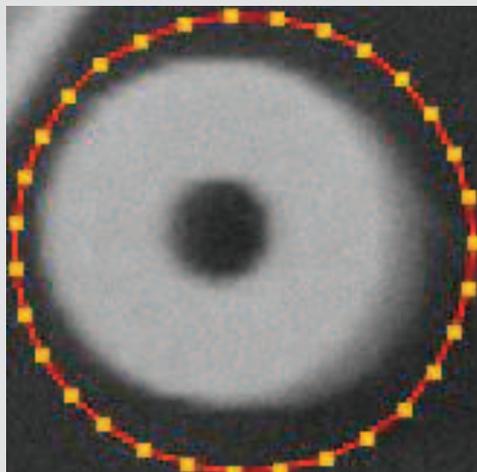
Exemple d'algorithme pour une représentation discrète

- ▶ Difficulté à aller détecter des coins
- ▶ Gestion de l'apparition de nouveaux points délicate
- ▶ Nombre de paramètres de réglage peut être conséquent



Détection de contours - contours actifs (snakes)

Exemple d'algorithme pour une représentation discrète



Analyse d'images

Fin de la première partie

ANNEXES

- ✓ *Exemple de code matlab*
- ✓ *Les modèles géodésiques actifs*

ANNEXES

- ✓ *Exemple de code matlab*
- ✓ *Les modèles géodésiques actifs*

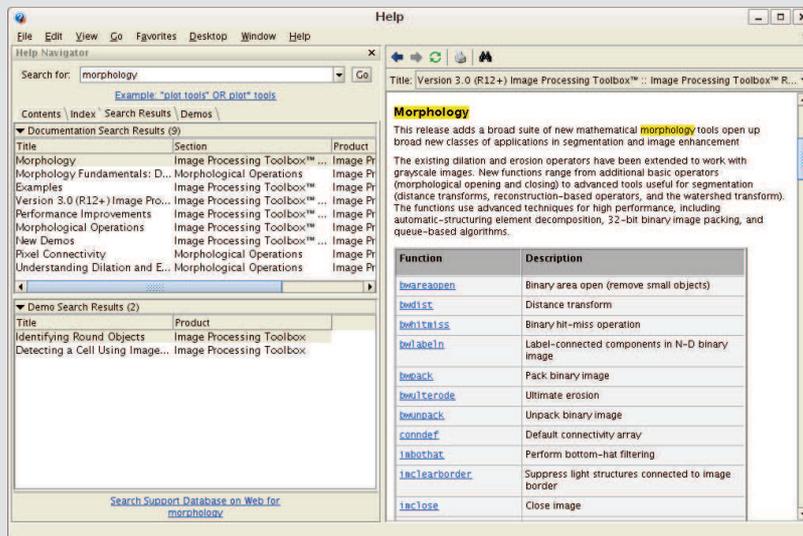
Analyse d'images - Annexes

- ▶ *L'ensemble des résultats du cours obtenus en utilisant matlab est téléchargeable à l'adresse internet suivante*

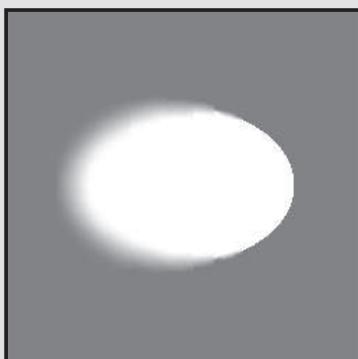
www.creatis.insa-lyon.fr/~bernard/AnalyseImage

Analyse d'images - Annexes

- **Liste des fonctions de morphologie mathématique accessible dans la documentation matlab en écrivant « morphology » dans l'onglet content**



Annexes - Détection de contour



```
% Lecture image
img = imread('ellipse.gif');
img = double(img(:,:,1));

% Calcul de la norme du gradient
[FX,FY] = gradient(img);
NG = sqrt( FX.*FX + FY.*FY );

% Affichage de l'image
figure; colormap(gray);
imagesc(NG); axis image;
```



Image initiale

Norme du gradient de l'image

Code matlab

Détection de contour

Exemple: gradient

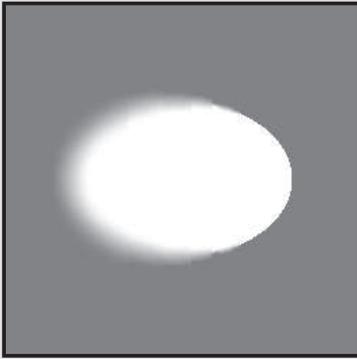


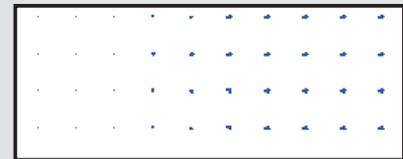
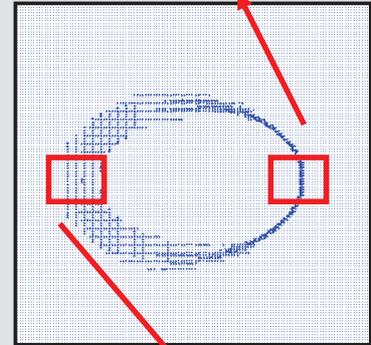
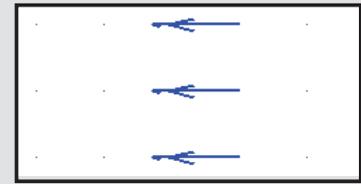
Image initiale

```
% Lecture image
img = imread('ellipse.gif');
img = double(img(:,:,1));

% Calcul du gradient
[FX,FY] = gradient(img);
xSpace=(1:2:size(img,1));
ySpace=(1:2:size(img,2));
qx=interp2(FX,xSpace,...
ySpace');
qy=interp2(FY,xSpace,...
ySpace');

% Affichage de l'image
figure; quiver(xSpace,...
ySpace,qx,qy);
axis image;
```

Code matlab



gradient (vecteur) de l'image

Département GE - TI - Olivier Bernard

151

Annexes – approches classiques

Exemple: Fourier



Image initiale

```
% Lecture image
img = imread('ellipse.png');
img = double(img(:,:,1));

% parameter
F_cut = 50; cx = 128;
cy = 128; ButtOrd = 2;
[dimx,dimy] = size(img);

% Calcul des fft
imf=fftshift(fft2(img));
H=zeros(dimx,dimy);
for i=1:dimx
    for j=1:dimy
        d=sqrt((i-cx)^2+(j-cy)^2);
        H(i,j)=sqrt(1/(1+...
((F_cut/d)^(2*ButtOrd))));
    end
end

% Extraction du contour
outf = imf .* H;

% Calcul des ifft
out = abs(ifft2(outf));
```

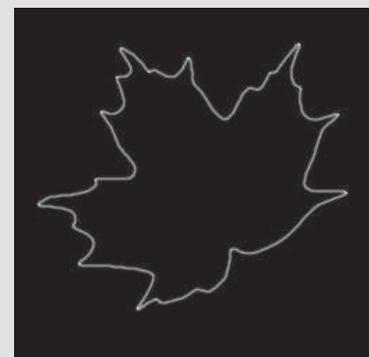


Image des hautes fréquences

Département GE - TI - Olivier Bernard

152

Annexes – approches classiques

► Exemple: filtre de Sobel



Image initiale

```
% Lecture image
img1 = imread('ellipse.png');
img1 = double(img1(:,:,1));

% creation du filtre de sobel
S = [ 1 0 -1; 2 0 -2; 1 0 -1 ];

% extraction des dérivées
img2 = conv2(img1,S,'same');
img3 = conv2(img1,S','same');

% Extraction du contour
img4 = sqrt( img2.^2 + ...
            img3.^2 + 1e-6);
```

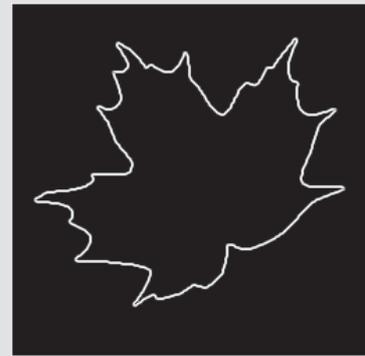


Image des forts gradients

Code matlab

Annexes – Seuillage par hystérésis

► Code matlab de la fonction hystérésis (connexité 4)

```
function out = hysteresis(in1,in2)

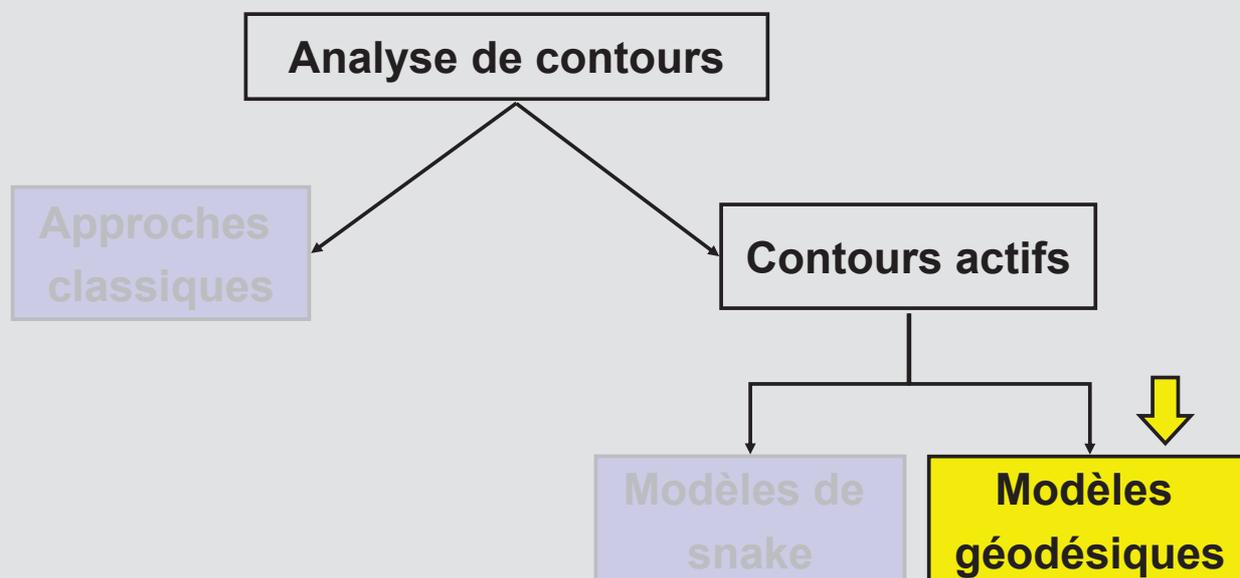
out = in2; count = 1; k=0;
MAXITERATION = 200;
while ( ( count ~= 0 ) && ( k < MAXITERATION ) )
    count = 0;
    for i=2:(size(out,1)-1)
        for j=2:(size(out,2)-1)
            if ( out(i,j) > 0 )
                if ( in1(i-1,j) > 0 )
                    out(i-1,j) = 255; count = count + 1;
                end
                if ( in1(i+1,j) > 0 )
                    out(i+1,j) = 255; count = count + 1;
                end
                if ( in1(i,j-1) > 0 )
                    out(i,j-1) = 255; count = count + 1;
                end
                if ( in1(i,j+1) > 0 )
                    out(i,j+1) = 255; count = count + 1;
                end
            end
        end
    end
    k = k + 1;
end
```

ANNEXES

- ✓ *Exemple de code matlab*
- ✓ **Les modèles géodésiques actifs**

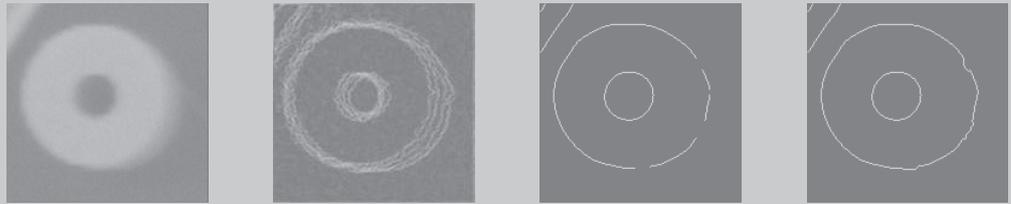
Détection de contours

■ Les familles d'analyse de contours

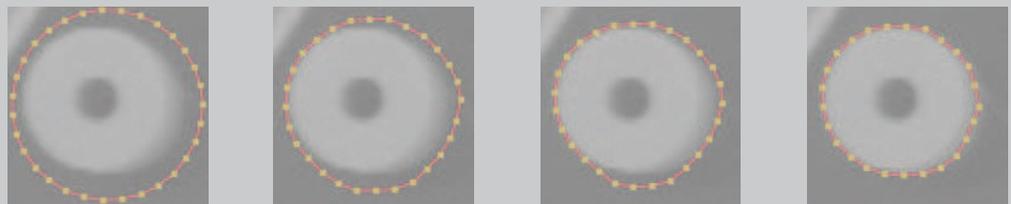


Détection et analyse de contours

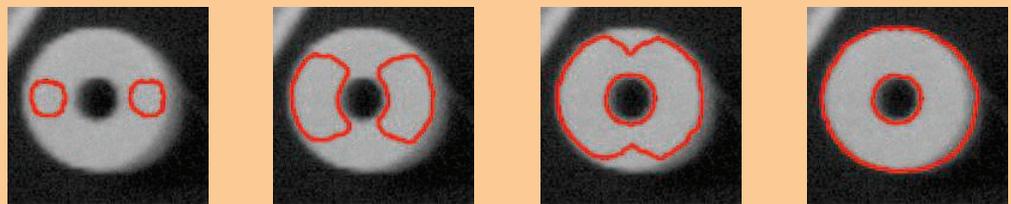
Méthodes classiques



Méthodes « snake »



Méthodes géodésiques



Annexes - modèles géodésiques actifs

■ Comment faire évoluer le contour actif ?

▶ Modélisation mathématique

- 1 - choix de la représentation du contour actif**
- 2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter**
- 3 - évolution du contour guidée par la minimisation de la fonction d'énergie**

Annexes - modèles géodésiques actifs

■ Comment faire évoluer le contour actif ?

▶ Modélisation mathématique

1 - choix de la représentation du contour actif

2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter

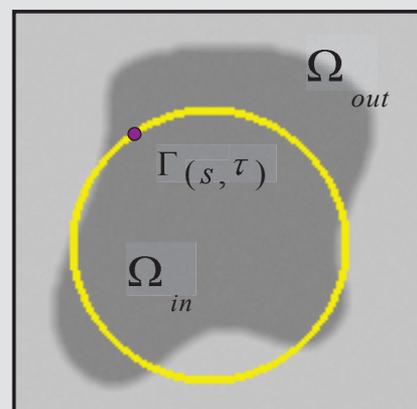
3 - évolution du contour guidée par la minimisation de la fonction d'énergie

Annexes - modèles géodésiques actifs

▶ Choix de la représentation du contour actif

Représentation continue

- $\Gamma(s, \tau) : [0, 1] \times [0, \infty[\rightarrow \mathbb{R}^2$
- $\Omega_{in}, \Omega_{out}$ région intérieure et extérieure à Γ



Annexes - contours actifs (snakes)

■ Comment faire évoluer le contour actif ?

▶ Modélisation mathématique

1 - choix de la représentation du contour actif

2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter

3 - évolution du contour guidée par la minimisation de la fonction d'énergie

Annexes - modèles géodésiques actifs

▶ Fonction d'énergie

$$E(\Gamma) = \int_0^1 g(I(\Gamma(s))) \left| \frac{\partial \Gamma(s)}{\partial s} \right| ds$$

avec $\begin{cases} I : \text{image traitée} \\ g : \mathbb{R}^2 \rightarrow \mathbb{R} : \text{fonction de pondération décroissante} \end{cases}$

Annexes - modèles géodésiques actifs

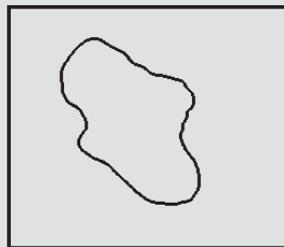
Illustration du critère géodésique

$$E(\Gamma) = \int_0^1 g \left| \frac{\partial \Gamma(s)}{\partial s} \right| ds$$

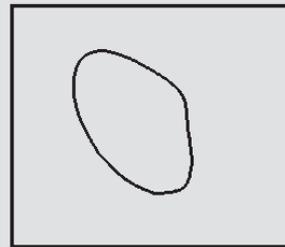
- ▶ $g(\cdot) = 1$ (*pas d'attache aux données*)
- ▶ $E(\Gamma) = \int_0^1 \left| \frac{\partial \Gamma(s)}{\partial s} \right| ds = \int_0^1 ds$
- ▶ *minimisation de la longueur du contour*



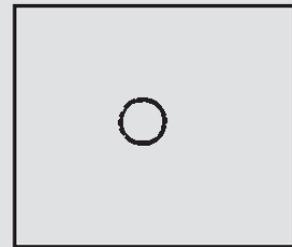
$t = 0$



$t = 1$



$t = 2$



$t = 3$

Annexes - modèles géodésiques actifs

Illustration du critère géodésique

$$E(\Gamma) = \int_0^1 g \left| \frac{\partial \Gamma(s)}{\partial s} \right| ds$$

- ▶ $g(\cdot) \neq 1$
- ▶ *pondération de longueur d'arc de cercle par $g(\cdot)$*
- ▶ *l'algorithme converge pour des points du contour où $g = 0$*



$g(I)$

Annexes - modèles géodésiques actifs

■ Comment faire évoluer le contour actif ?

▶ Modélisation mathématique

1 - choix de la représentation du contour actif

2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter

3 - évolution du contour guidée par la minimisation de la fonction d'énergie

Annexes - modèles géodésiques actifs

▶ Évolution du contour guidée par la minimisation de la fonction d'énergie

● Utilisation d'outil mathématique de calcul variationnel

$$\frac{\partial \Gamma(\tau)}{\partial \tau} = V(\tau) \vec{N}$$

$$\text{avec } V = \underbrace{g \cdot \kappa}_{\text{terme de lissage}} - \underbrace{g \cdot \vec{N}}_{\text{terme d'attache aux données}}$$

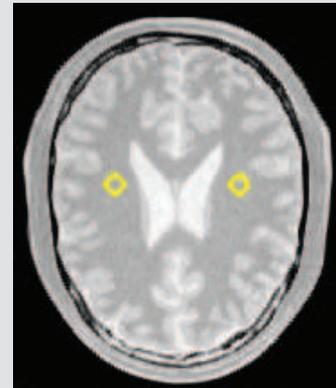
terme de
lissage

terme d'attache
aux données

Annexes - modèles géodésiques actifs

► Évolution du contour guidée par la minimisation de la fonction d'énergie

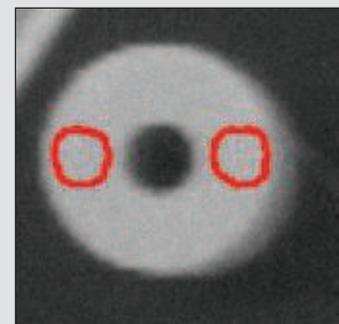
- Implémentation particulière faisant intervenir la notion d'ensemble de niveaux
- Permet la gestion intrinsèque des changements de topologie de la courbe au cours de son évolution



Annexes - modèles géodésiques actifs

► Évolution du contour guidée par la minimisation de la fonction d'énergie

- Implémentation plus complexe que le modèle des snake
- Mais moins de paramètres de réglage
- Gestion intrinsèque des variations de topologie du contour actif



Traitement d'images numériques

Analyse d'images: 2^{ème} partie

*Département Génie Electrique
5GE - TdSi*

Sommaire

- I. Introduction
- II. Fondamentaux en imagerie numérique
- III. Traitement discret 2D
- IV. Amélioration d'image

V. Analyse d'images (6 heures)

Analyse d'images

■ Sources qui ont inspirées cette partie:

- ★ *Henri Maître (2003) : Le traitement des images, Hermes-Lavoisier*
- ★ *J.P. Cocquerez (1995) : Analyse d'images: filtrage et segmentation, Hermes-Lavoisier*
- ★ *http://www.ensta.fr/~manzaner/Support_Cours.html*
- ★ *<http://www.tsi.enst.fr/~tupin/TDI.html>*

Sommaire

V. Analyse d'images

- ✓ *1^{ère} partie*
 - Morphologie mathématique
 - Détection et analyse de contours
- ✓ *2^{ème} partie*
 - Segmentation par régions
- ✓ *3^{ème} partie*
 - Analyse de formes
 - Extraction de caractéristiques

Sommaire

V. Analyse d'images

- ✓ *1^{ère} partie*
 - Morphologie mathématique
 - Détection et analyse de contours

- ✓ *2^{ème} partie*
 - **Segmentation par régions**

- ✓ *3^{ème} partie*
 - Analyse de formes
 - Extraction de caractéristiques

Segmentation par régions

- **Contexte**
 - Définition
 - Principe

- **Les différentes méthodes**
 - Méthodes par histogramme
 - Méthodes par transformation de région
 - Méthodes par optimisation

Segmentation par régions

■ Contexte - Définition

La segmentation consiste à partitionner une image en un ensemble de régions connexes

- ▶ Repose sur la recherche de zones de l'image possédant des attributs communs, comme la luminosité ou plus rarement la texture

Segmentation par régions

■ Contexte - Définition

- ▶ L'intérêt de détecter des régions est de pouvoir les manipuler ensuite pour extraire des caractéristiques de forme, de position, de taille ...



*Exemple d'utilisation de
segmentation par régions:
étiquetage de pièces,
d'objets*

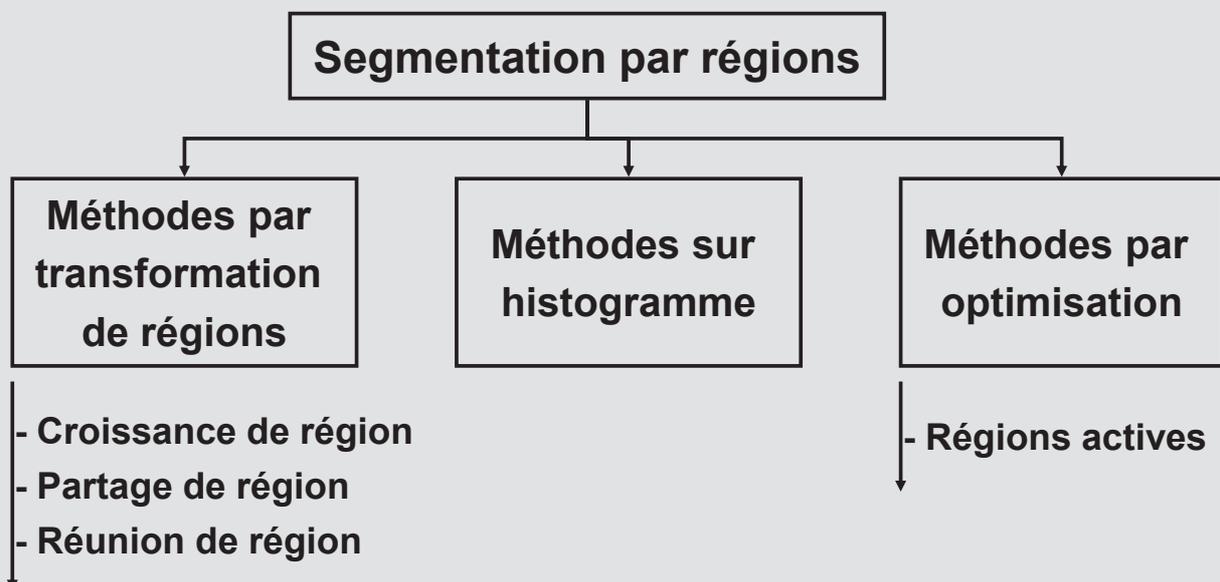
Segmentation par régions

■ Contexte - Définition

- ▶ Le problème de segmentation par régions est mal posé \Rightarrow il n'existe pas de segmentation idéale
- ▶ Le choix d'une méthode est lié
 - à la nature des images (éclairage, texture, ...)
 - à la forme des primitives à extraire
 - aux contraintes de temps

Segmentation par régions

■ Contexte – Les différentes méthodes



Segmentation par régions

■ Contexte

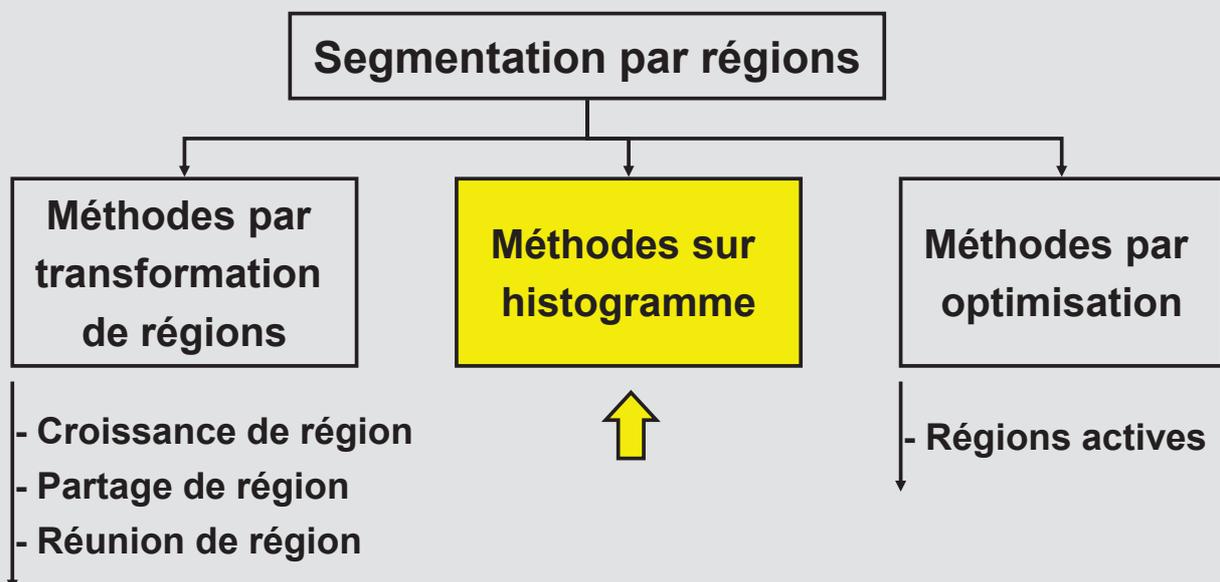
- Définition
- Principe

■ Les différentes méthodes

- Méthodes par histogramme
- Méthodes par transformation de région
- Méthodes par optimisation

Segmentation par régions

■ Contexte – Les différentes méthodes



Segmentation par régions

■ Méthodes sur histogramme

L'idée générale de ces méthodes consiste à isoler des pics de l'histogramme, puis de seuiller les zones correspondantes

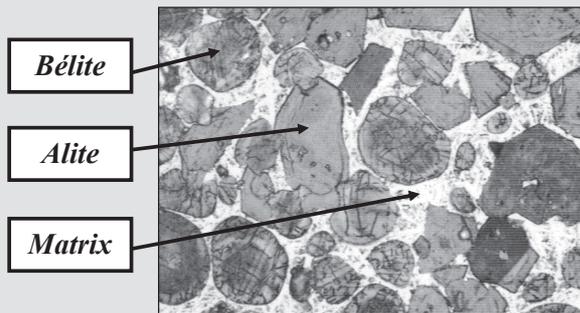


Image originale



Séparation des classes Alite et Belite de la classe cérite (le fond)

Segmentation par régions

■ Méthodes sur histogramme - Principe

- ▶ Localisation sur un histogramme d'un mode isolé
- ▶ Détection de la zone correspondante par seuillage
- ▶ Parmi les zones de l'image contribuant à ce mode, sélection de la région connexe la plus grande grâce à l'utilisation de méthodes haut-niveaux (exemple: morphologie mathématique)
- ▶ Itération sur un autre mode isolé de l'histogramme

Segmentation par régions

■ Méthodes sur histogramme - Exemple

- ▶ *Localisation sur l'histogramme d'un mode isolé*

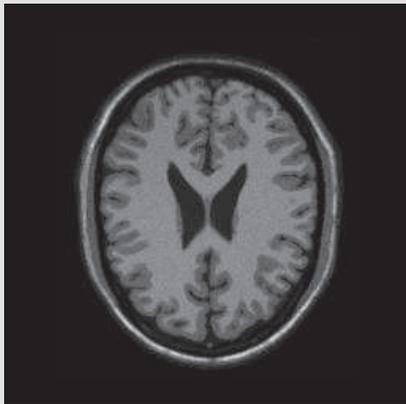
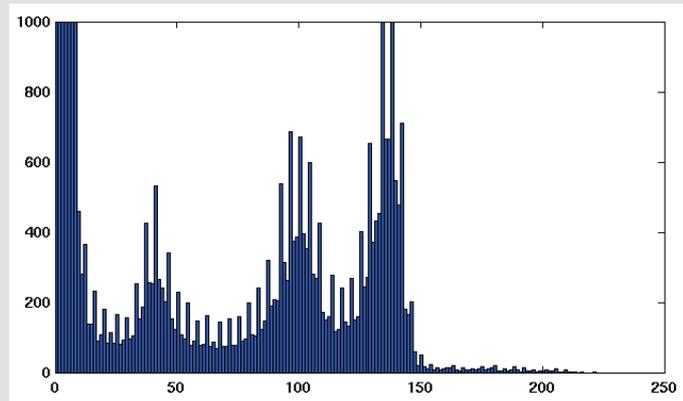


Image originale

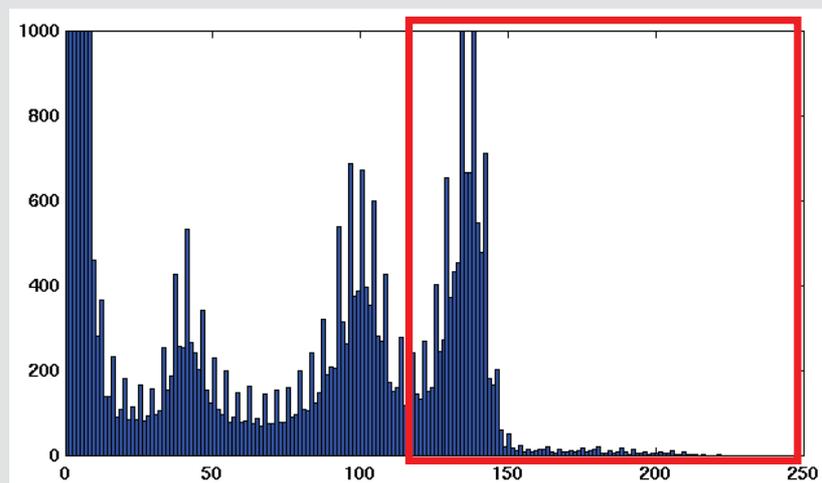


Histogramme de l'image

Segmentation par régions

■ Méthodes sur histogramme - Exemple

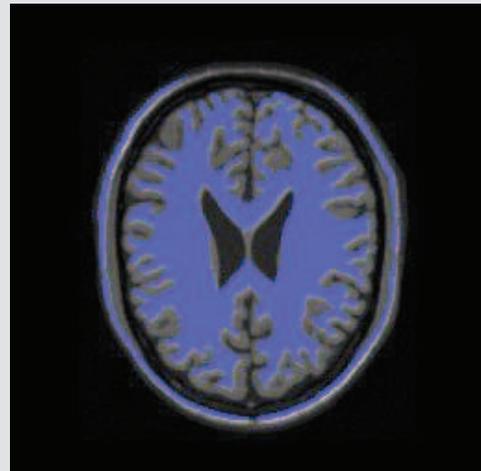
- ▶ *Localisation sur l'histogramme d'un mode isolé*



Segmentation par régions

■ Méthodes sur histogramme - Exemple

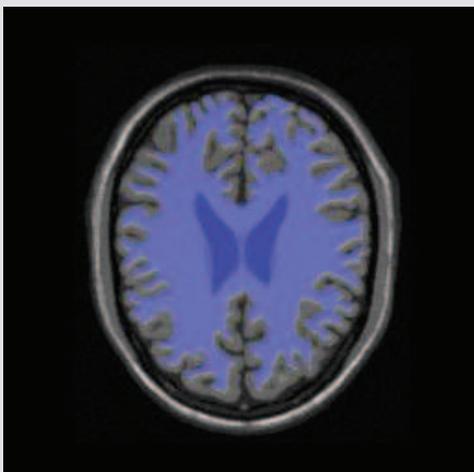
- ▶ *Détection de la zone correspondante par seuillage*



Segmentation par régions

■ Méthodes sur histogramme - Exemple

- ▶ *Sélection de la région connexe la plus grande*



Algorithmes utilisés

- *Seuillage par hystérésis*
- *Morphologie mathématique (érosion, bouchage de trous, fermeture)*

Segmentation par régions

► Code matlab correspondant

```
% Lecture image
img = imread('brain.png');
img = double(img(:,:,1));

% Selection des pixels correspondant a un
% mode de l'histogramme
e = find(img>118);
mask = zeros(size(img));
mask(e) = 255;

% Erosion de la region par morpho math
se = strel('disk',3);
maskErode = imerode(mask,se);

% Seuillage par hysteresis
maskHyst = hysteresis(mask,maskErode);

% Bouche les trous par morpho math
maskFill = imfill(maskHyst,'holes');

% Fermeture de la region par morpho math
se = strel('disk',1);
final = imclose(maskFill,se);
```

Segmentation par régions

► Code matlab correspondant

```
% Lecture image
img = imread('brain.png');
img = double(img(:,:,1));

% Selection des pixels correspondant a un
% mode de l'histogramme
e = find(img>118);
mask = zeros(size(img));
mask(e) = 255;

% Erosion de la region par morpho math
se = strel('disk',3);
maskErode = imerode(mask,se);

% Seuillage par hysteresis
maskHyst = hysteresis(mask,maskErode);

% Bouche les trous par morpho math
maskFill = imfill(maskHyst,'holes');

% Fermeture de la region par morpho math
se = strel('disk',1);
final = imclose(maskFill,se);
```

Sélection des pixels d'un mode



mask

Segmentation par régions

► Code matlab correspondant

```
% Lecture image
img = imread('brain.png');
img = double(img(:,:,1));

% Selection des pixels correspondant a un
% mode de l'histogramme
e = find(img>118);
mask = zeros(size(img));
mask(e) = 255;

% Erosion de la region par morpho math
se = strel('disk',3);
maskErode = imerode(mask,se);

% Seuillage par hysteresis
maskHyst = hysteresis(mask,maskErode);

% Bouche les trous par morpho math
maskFill = imfill(maskHyst,'holes');

% Fermeture de la region par morpho math
se = strel('disk',1);
final = imclose(maskFill,se);
```

Erosion de la région extraite



maskErode

Segmentation par régions

► Code matlab correspondant

```
% Lecture image
img = imread('brain.png');
img = double(img(:,:,1));

% Selection des pixels correspondant a un
% mode de l'histogramme
e = find(img>118);
mask = zeros(size(img));
mask(e) = 255;

% Erosion de la region par morpho math
se = strel('disk',3);
maskErode = imerode(mask,se);

% Seuillage par hysteresis
maskHyst = hysteresis(mask,maskErode);

% Bouche les trous par morpho math
maskFill = imfill(maskHyst,'holes');

% Fermeture de la region par morpho math
se = strel('disk',1);
final = imclose(maskFill,se);
```

Seuillage par hystérésis



maskHyst

Segmentation par régions

► Code matlab correspondant

```
% Lecture image
img = imread('brain.png');
img = double(img(:,:,1));

% Selection des pixels correspondant a un
% mode de l'histogramme
e = find(img>118);
mask = zeros(size(img));
mask(e) = 255;

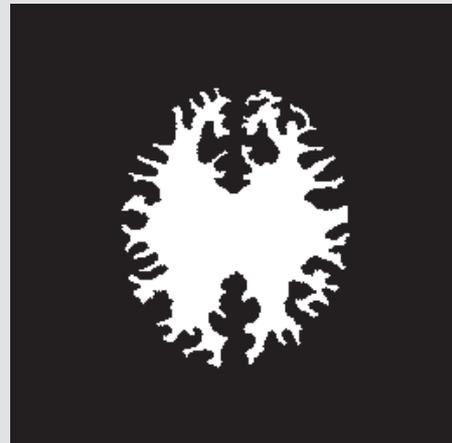
% Erosion de la region par morpho math
se = strel('disk',3);
maskErode = imerode(mask,se);

% Seuillage par hysteresis
maskHyst = hysteresis(mask,maskErode);

% Bouche les trous par morpho math
maskFill = imfill(maskHyst,'holes');

% Fermeture de la region par morpho math
se = strel('disk',1);
final = imclose(maskFill,se);
```

Bouchage de trous



maskFill

Segmentation par régions

► Code matlab correspondant

```
% Lecture image
img = imread('brain.png');
img = double(img(:,:,1));

% Selection des pixels correspondant a un
% mode de l'histogramme
e = find(img>118);
mask = zeros(size(img));
mask(e) = 255;

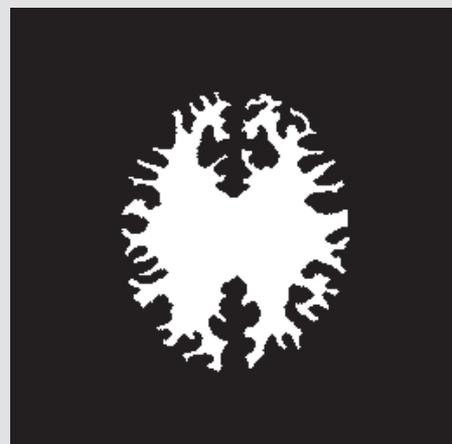
% Erosion de la region par morpho math
se = strel('disk',3);
maskErode = imerode(mask,se);

% Seuillage par hysteresis
maskHyst = hysteresis(mask,maskErode);

% Bouche les trous par morpho math
maskFill = imfill(maskHyst,'holes');

% Fermeture de la region par morpho math
se = strel('disk',1);
final = imclose(maskFill,se);
```

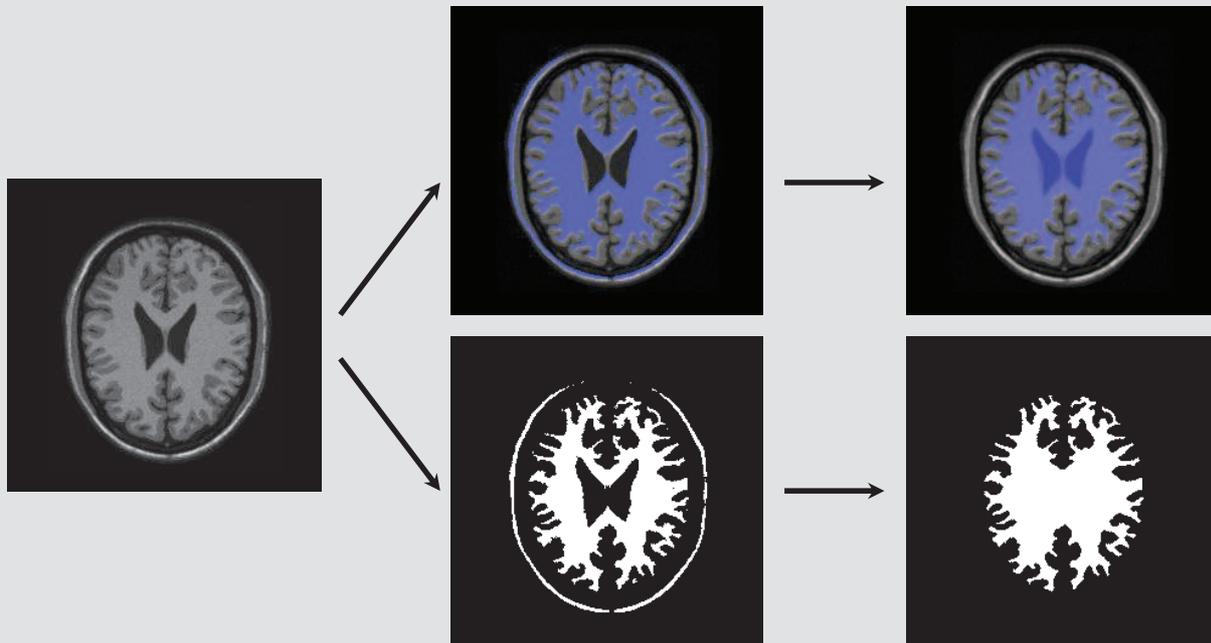
Fermeture de la région



final

Segmentation par régions

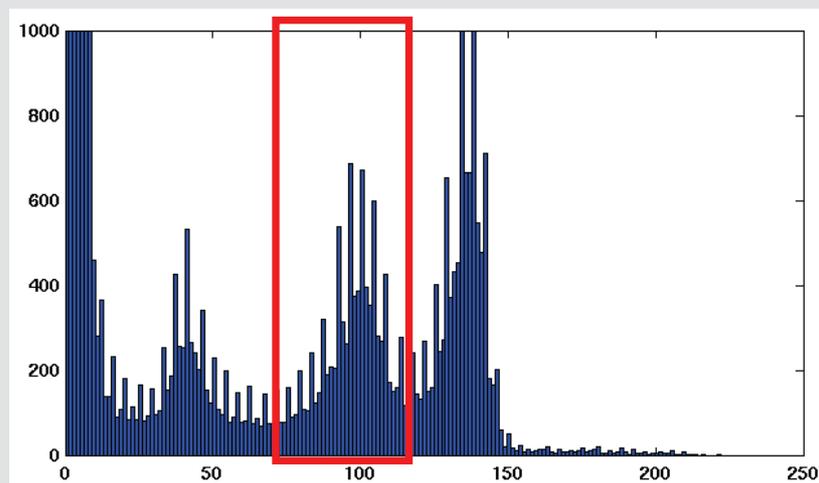
■ Méthodes sur histogramme - Exemple



Segmentation par régions

■ Méthodes sur histogramme - Exemple

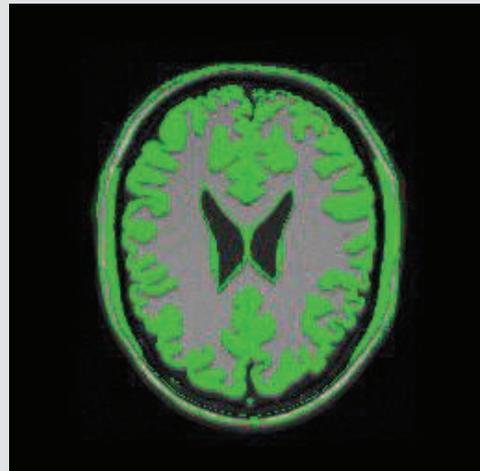
- ▶ *Localisation sur l'histogramme d'un mode isolé*



Segmentation par régions

■ Méthodes sur histogramme - Exemple

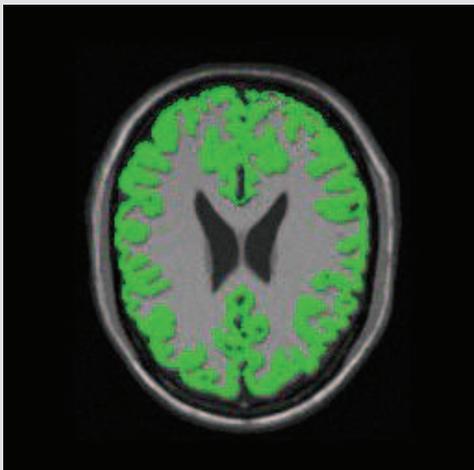
- ▶ *Détection de la zone correspondante par seuillage*



Segmentation par régions

■ Méthodes sur histogramme

- ▶ *Sélection de la région connexe la plus grande*



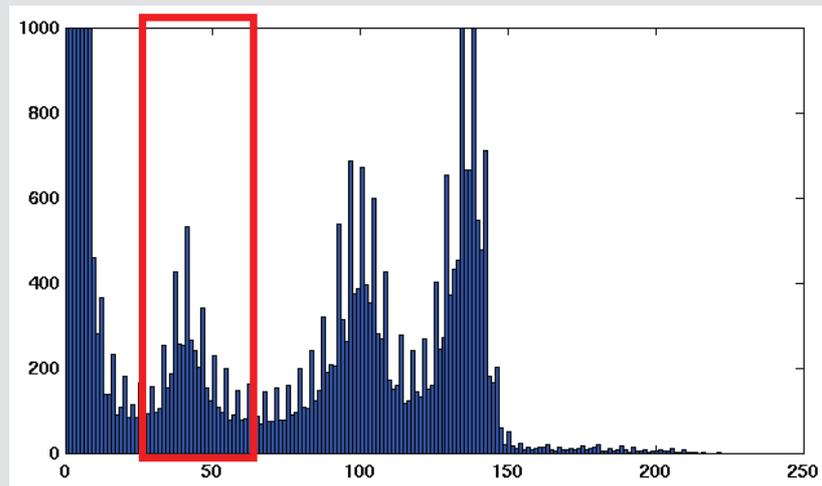
Algorithmes utilisés

- *Seuillage par hystérésis*
- *Morphologie mathématique (érosion, fermeture)*

Segmentation par régions

■ Méthodes sur histogramme - Exemple

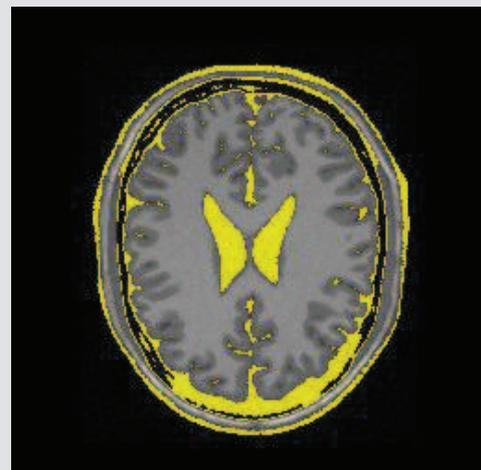
- ▶ *Localisation sur l'histogramme d'un mode isolé*



Segmentation par régions

■ Méthodes sur histogramme - Exemple

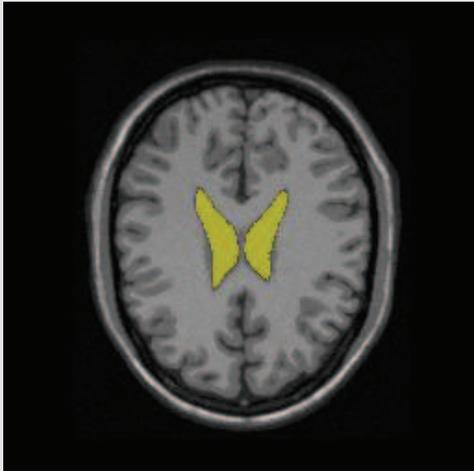
- ▶ *Détection de la zone correspondante par seuillage*



Segmentation par régions

■ Méthodes sur histogramme

- ▶ *Sélection de la région connexe la plus grande*



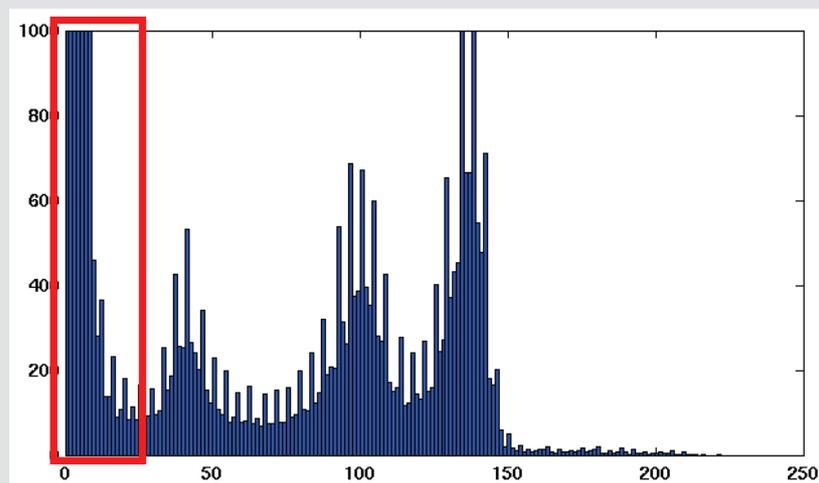
Algorithmes utilisés

- *Seuillage par hystérésis*
- *Morphologie mathématique (érosion, fermeture)*

Segmentation par régions

■ Méthodes sur histogramme - Exemple

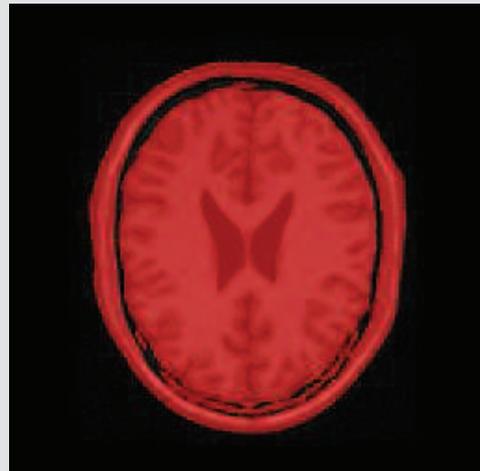
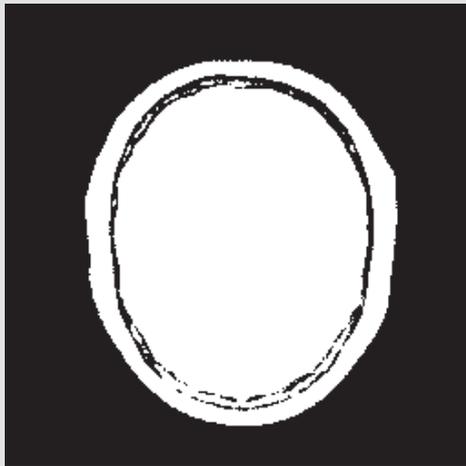
- ▶ *Localisation sur l'histogramme d'un mode isolé*



Segmentation par régions

■ Méthodes sur histogramme - Exemple

- ▶ *Détection de la zone correspondante par seuillage*



Segmentation par régions

■ Méthodes sur histogramme

- ▶ *Sélection de la région connexe la plus grande*



Algorithmes utilisés

- *Morphologie mathématique (bouchage de trous, fermeture)*

Segmentation par régions

■ Méthodes sur histogramme

▶ *Résultat final*

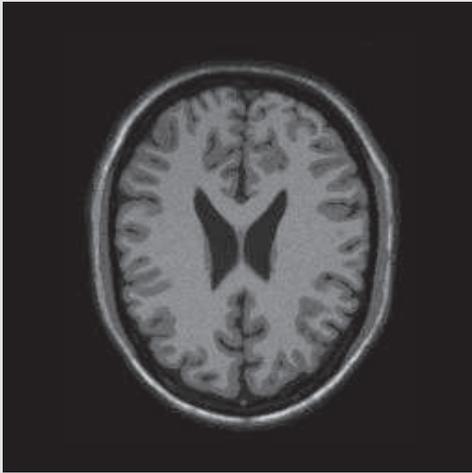


Image originale

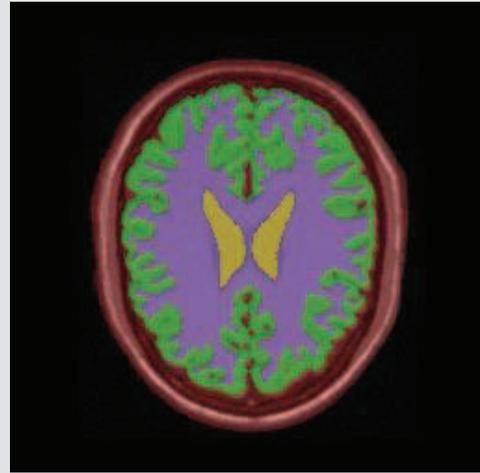


Image segmentée

Segmentation par régions

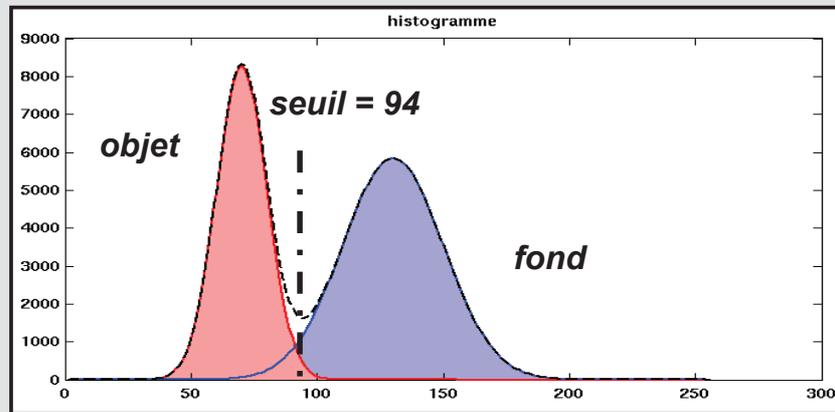
■ Méthodes sur histogramme – Choix du seuil

- ▶ Comment choisir judicieusement la valeur du seuil ?
- ▶ Il existe de nombreuses méthodes basées sur l'univers des probabilités
- ▶ Exemple: seuillage avec apprentissage

Segmentation par régions

■ Méthodes sur histogramme – Choix du seuil

▶ Exemple: seuillage avec apprentissage



But: détection de vallées, en prenant le minimum de l'histogramme situé entre deux pics

Segmentation par régions

■ Méthodes sur histogramme – Choix du seuil

▶ Méthode analytique

- Modélisation de chaque pic par des distributions a priori (par exemple des gaussiennes p_1 et p_2)
- Maximisation de la fonctionnelle de coût suivante

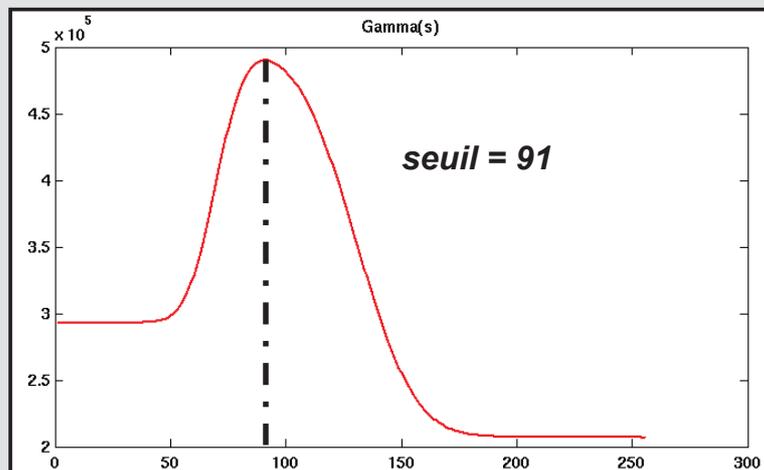
$$\left\{ \begin{array}{l} \Gamma(s) = \int_0^s p_1(n)dn + \int_s^N p_2(n)dn \\ \Gamma(s) = 1 - F_2(n=s) + F_1(n=s) \end{array} \right.$$

avec F_i fonction de répartition

Segmentation par régions

■ Méthodes sur histogramme – Choix du seuil

- Maximisation de la fonctionnelle $\Gamma(s)$



- ◆ *Approximation d'une intégrale continue en intégrale discrète*

Segmentation par régions

■ Méthodes sur histogramme – Choix du seuil

▶ Méthode empirique

- Trouver le seuil qui minimise les variances des deux classes
- Soit $h[n]$ l'histogramme
- Centre de gravité G d'une classe
- Variance var d'une classe

$$G_i(s) = \frac{\sum_{n \in C_i} n h[n]}{\sum_{n \in C_i} h[n]}$$

$$var_i(s) = \sum_{n \in C_i} (n - G_i)^2 h[n]$$

Segmentation par régions

■ Méthodes sur histogramme – Choix du seuil

▶ Méthode empirique

- Trouver le seuil qui minimise la somme des variances

$$s_{opt} = \arg \min_s (\text{var}_1(s) + \text{var}_2(s))$$

- En simplifiant les termes en carré, cela revient à maximiser la fonctionnelle $J(s)$ suivante :

$$J(s) = \frac{\left(\sum_{n \in C_1} nh[n] \right)^2}{\sum_{n \in C_1} h[n]} + \frac{\left(\sum_{n \in C_2} nh[n] \right)^2}{\sum_{n \in C_2} h[n]}$$

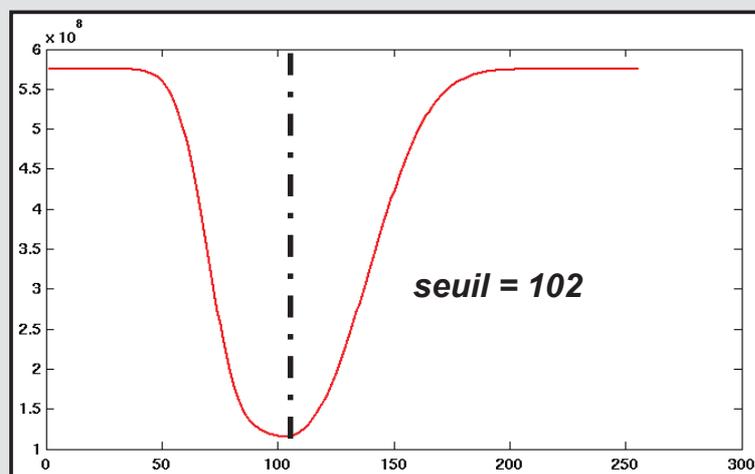
Département GE - TI - Olivier Bernard

41

Segmentation par régions

■ Méthodes sur histogramme – Choix du seuil

- Minimisation de la fonctionnelle S_{opt}



- ◆ Approximation d'une intégrale continue en intégrale discrète

Département GE - TI - Olivier Bernard

42

Segmentation par régions

■ Méthodes sur histogramme - Propriétés

- ▶ Mise en œuvre assez simple
- ▶ Performance assez réduite puisqu'elles ne tirent pas profit de l'aspect spatial de l'information image
- ▶ Méthodes à utiliser lorsque:
 - les images présentent des classes évidentes (objets très contrastés)
 - lorsque les images sont définies sur de nombreux canaux (images multi-spectrales) ce qui enrichit l'information portée par l'histogramme

Segmentation par régions

■ Contexte

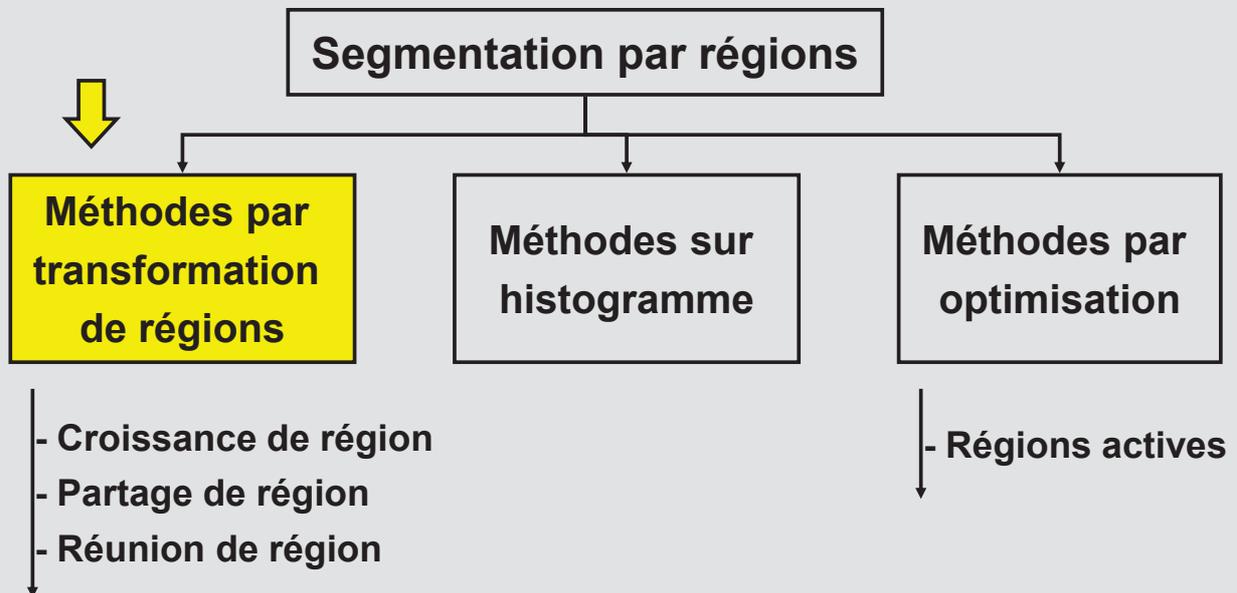
- Définition
- Principe

■ Les différentes méthodes

- Méthodes par histogramme
- Méthodes par transformation de régions
- Méthodes par optimisation

Segmentation par régions

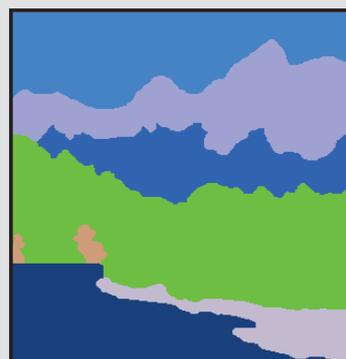
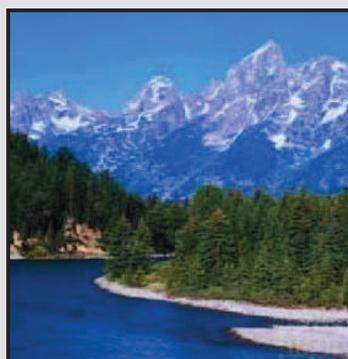
■ Contexte – Les différentes méthodes



Segmentation par régions

■ Méthodes par transformation de régions

Traitement qui consiste à créer une partition de l'image I en sous-ensembles R_i , appelés régions



Segmentation par régions

■ Méthodes par transformation de régions

- ▶ Méthodes qui s'appuient sur la notion de prédicat P et sur celle de partition
- ▶ Un prédicat est une proposition logique dont la valeur dépend de son argument

Segmentation par régions

■ Méthodes par transformation de régions

- ▶ Prédicat en segmentation: la région R_i est homogène
- ▶ Arguments utilisés pour vérifier le prédicat précédent
 - **Contraste sur la région**

$$P(R_i) = \text{vrai} \Leftrightarrow \max_{R_i} I(x, y) - \min_{R_i} I(x, y) < \sigma$$

Segmentation par régions

■ Méthodes par transformation de régions

- ▶ Prédicat en segmentation: la région R_i est homogène
- ▶ Arguments utilisés pour vérifier le prédicat précédent

- Écart-type sur la région

$$P(R_i) = \text{vrai} \Leftrightarrow \sqrt{\frac{1}{N} \sum_{R_i} (I(x, y) - m)^2} < \sigma$$

avec $\begin{cases} N = \text{Card}(R_i) \\ m = \frac{1}{N} \sum_{R_i} I(x, y) \end{cases}$

Segmentation par régions

■ Méthodes par transformation de régions

- ▶ Prédicat en segmentation: la région R_i est homogène
- ▶ Arguments utilisés pour vérifier le prédicat précédent

- L'entropie (mesure de quantité d'information)

$$P(R_i) = \text{vrai} \Leftrightarrow - \sum_{R_i} p(I) \log(P(I)) < \sigma$$

Segmentation par régions

■ Méthodes par transformation de régions

- ▶ Méthodes qui s'appuient sur la notion de prédicat P et sur celle de partition
- ▶ Une partition Π est un ensemble de régions R_i de l'image, vérifiant

$$\Pi \Leftrightarrow \begin{cases} \forall i, j (i \neq j) R_i \cap R_j = \emptyset \\ \bigcup_i R_i = \text{Support}(\text{image}) \\ \forall i R_i \neq \emptyset \end{cases}$$

Segmentation par régions

■ Méthodes par transformation de régions

- ▶ Il existe un très grand nombre de partitions d'une image
- ▶ Il existe un très grand nombre de partitions qui vérifie le prédicat de segmentation

Segmentation par régions

■ Méthodes par transformation de régions

- ▶ Comment choisir entre plusieurs partitions qui vérifient le même prédicat ?
 - Le cardinal de la partition (à minimiser)
 - La taille de la plus petite région (à maximiser)
 - Une distance entre régions (somme des distances entre zones adjacentes à maximiser)

Segmentation par régions

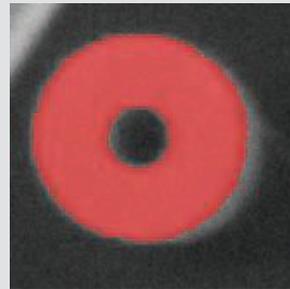
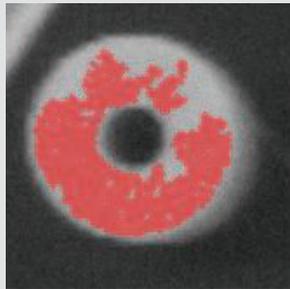
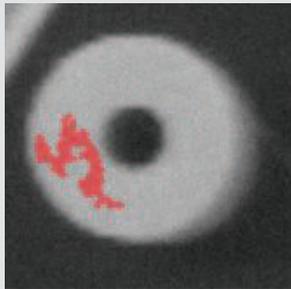
■ Méthodes par transformation de régions

- ▶ Il existe trois grandes catégories
 - La croissance de région
 - Le partage de région (*quad-tree*)
 - Les graphes d'adjacence

Segmentation par régions

■ La croissance de région

Méthode basée sur l'agglomération de pixels voisins à une région vérifiant l'argument d'un prédicat donné



Segmentation par régions

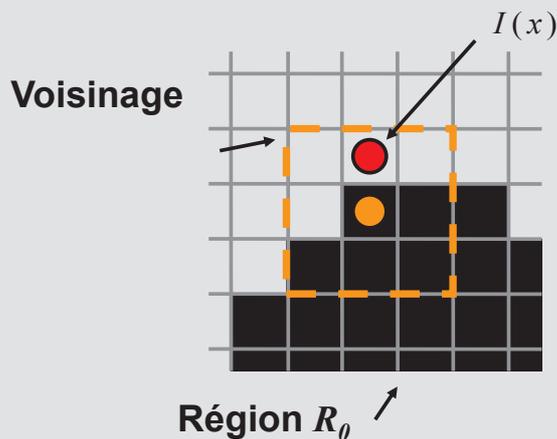
■ La croissance de région - Principe

- ▶ Initialisation d'une région R_0 à un pixel ou un groupe de pixels (germes)
- ▶ Ajout à R_0 de tous les pixels voisins dont l'argument vérifie le prédicat
- ▶ Itération jusqu'à convergence

Segmentation par régions

■ La croissance de région - Principe

- ▶ Exemple d'argument: l'homogénéité suivant la moyenne de la région μ_R (mise à jour à chaque itération)



argument

$$|I(x) - \mu_R| < \text{seuil}$$

Segmentation par régions

■ La croissance de région – Exemple 2D



Après quelques itérations



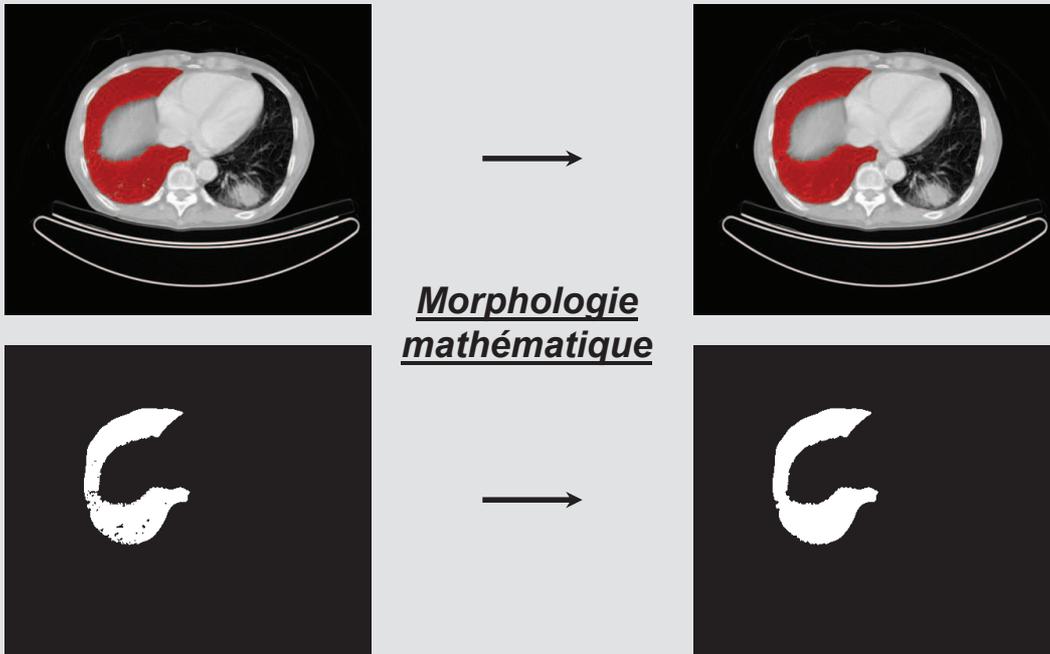
Après 5000 itérations



Convergence obtenue à 10119 itérations

Segmentation par régions

■ La croissance de région – Exemple 2D



Segmentation par régions

■ La croissance de région – Exemple 2D



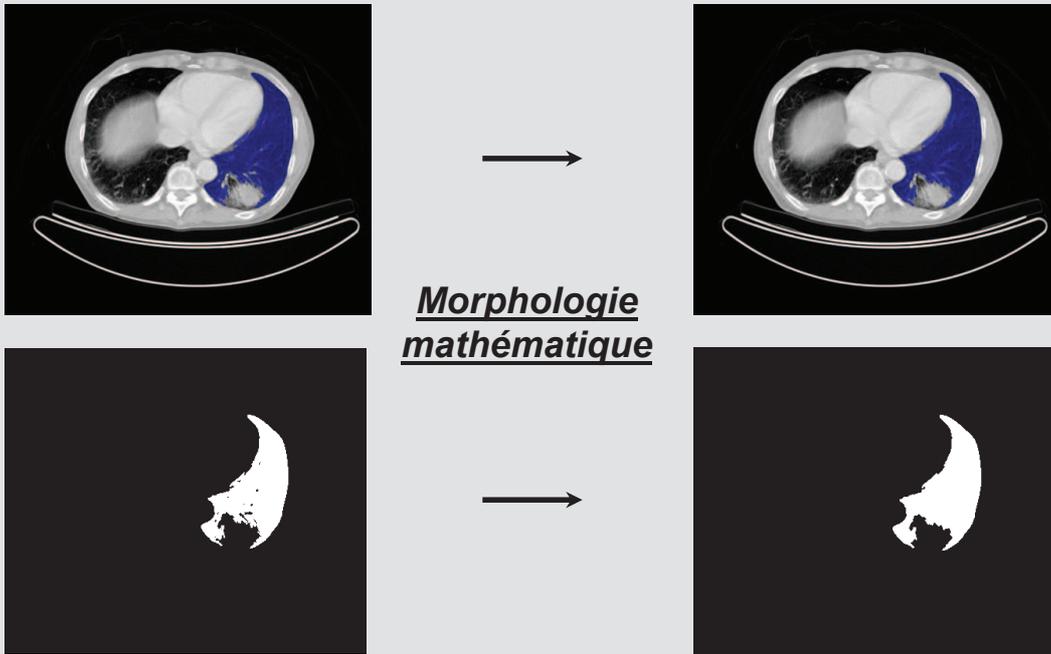
Après quelques itérations

Après 5000 itérations

Convergence obtenue à 10085 itérations

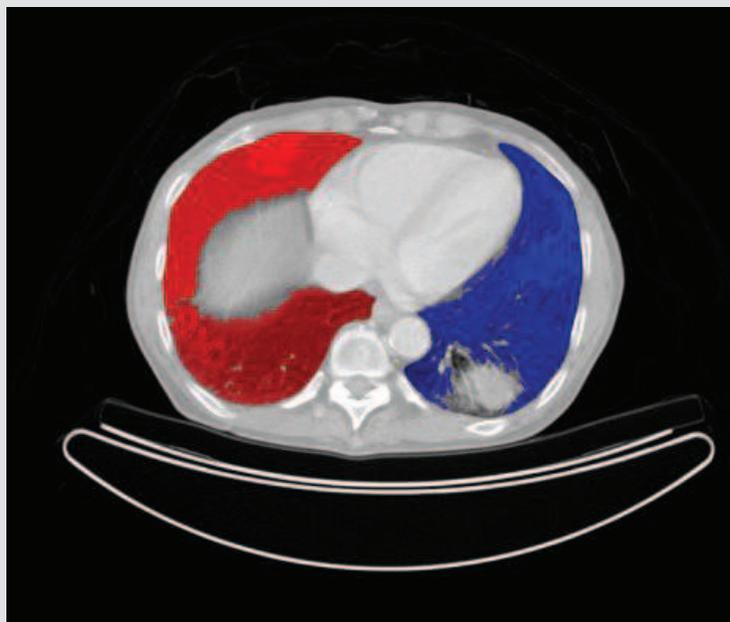
Segmentation par régions

■ La croissance de région – Exemple 2D



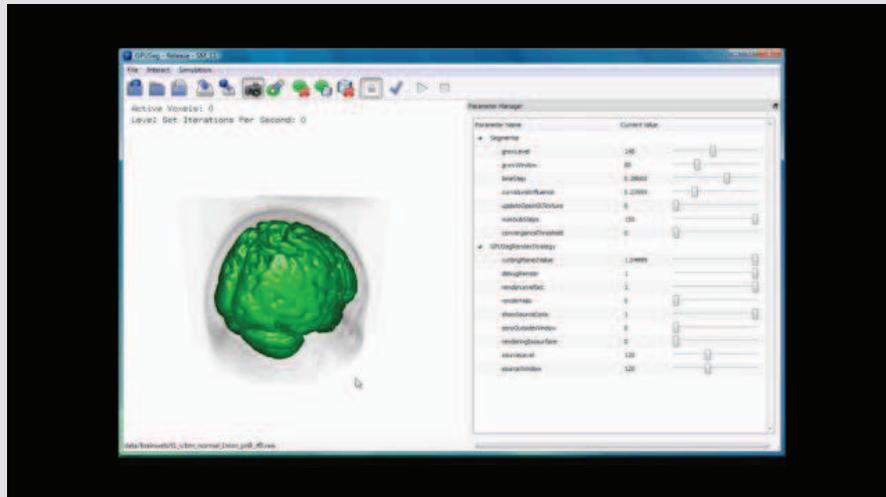
Segmentation par régions

■ La croissance de région – Exemple 2D



Segmentation par régions

■ La croissance de région – Exemple 3D



[Lien](#)

Segmentation par régions

■ La croissance de région - Propriétés

- ▶ La performance des résultats dépend de l'emplacement des germes initiaux
- ▶ L'ordre dans lequel sont ajoutés les pixels dans une région a une influence sur le résultat
- ▶ Implémentation relativement simple et temps d'exécution rapides

Segmentation par régions

■ Méthodes par transformation de régions

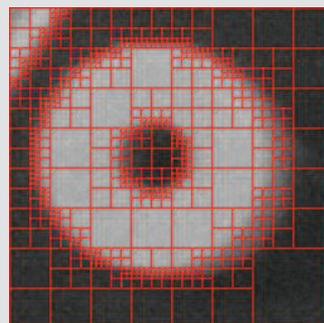
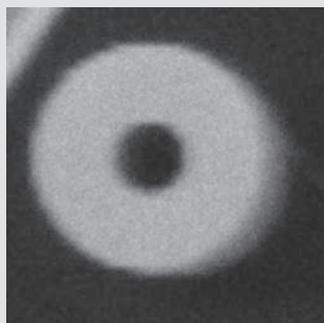
▶ Il existe trois grandes catégories

- La croissance de région
- Le partage de région (*quad-tree*)
- Les graphes d'adjacence

Segmentation par régions

■ Le partage de région

Le but de ces méthodes est de partager de façon automatique une image en un ensemble de régions proches du point de vue du prédicat choisi



Segmentation par régions

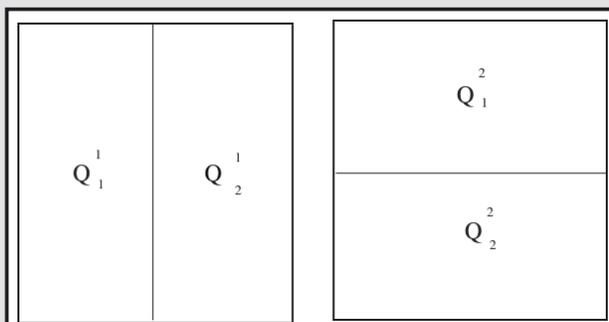
■ Le partage de région - Principe

- ▶ On part de l'image entière I . On appelle R cette région
- ▶ Application de plusieurs divisions δ produisant de nouvelles régions R_i^δ
- ▶ Pour chaque R_i^δ on test le prédicat P et on retient la meilleure subdivision δ , c'est à dire
 - celle qui conduit à des sous-régions vérifiant toute P
 - ou celle qui donne le plus de sous-régions vérifiant P
- ▶ Chaque sous-région ne vérifiant pas P devient alors une région R passible du traitement ci-dessus

Segmentation par régions

■ Le partage de région

- ▶ Exemple de partage de régions à deux partitions



Partitions d'une zone: choix entre partition verticale ou horizontale

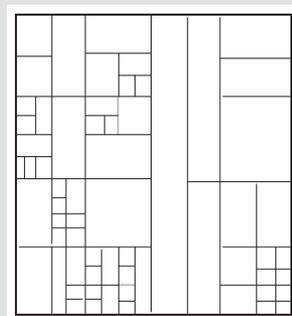


Image partitionnée

Segmentation par régions

■ Le partage de région - Exemple

- ▶ *Quad-tree (une seule partition carrée)*

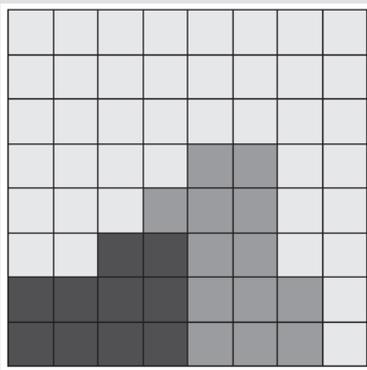
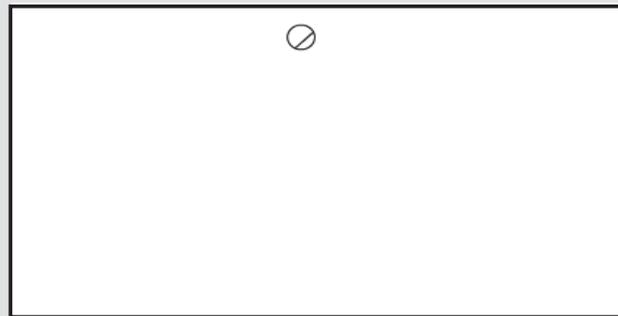


Image originale



Construction du Quad-tree correspondant

Segmentation par régions

■ Le partage de région - Exemple

- ▶ *Quad-tree (une seule partition carrée)*

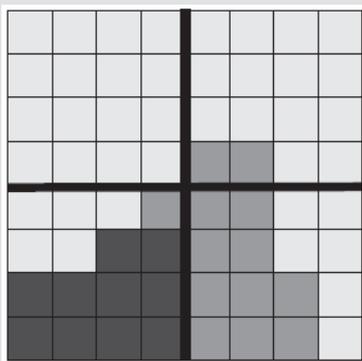
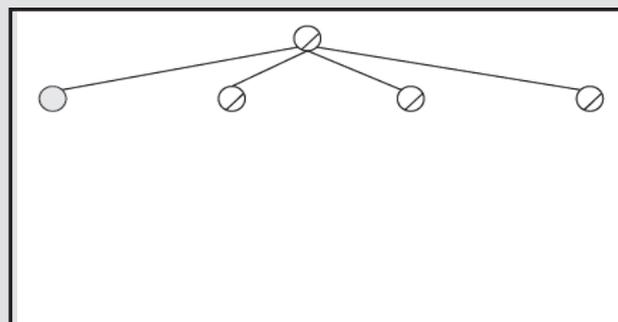


Image originale



Construction du Quad-tree correspondant

Segmentation par régions

■ Le partage de région - Exemple

- ▶ *Quad-tree (une seule partition carrée)*

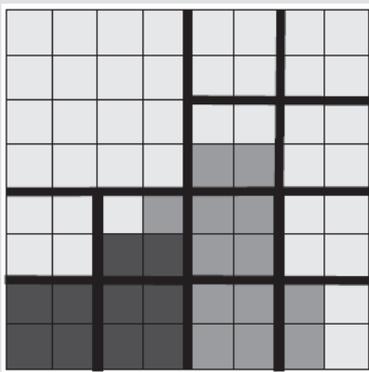
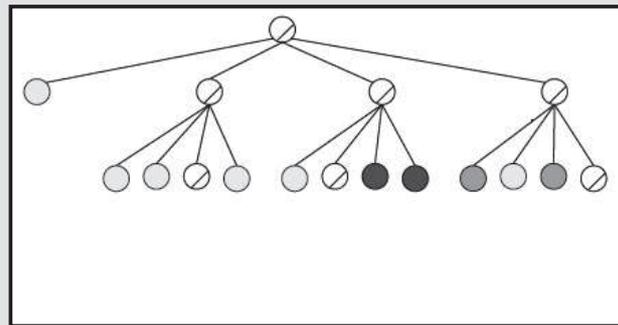


Image originale



Construction du Quad-tree correspondant

Segmentation par régions

■ Le partage de région - Exemple

- ▶ *Quad-tree (une seule partition carrée)*

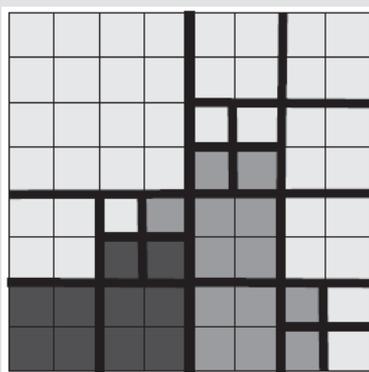
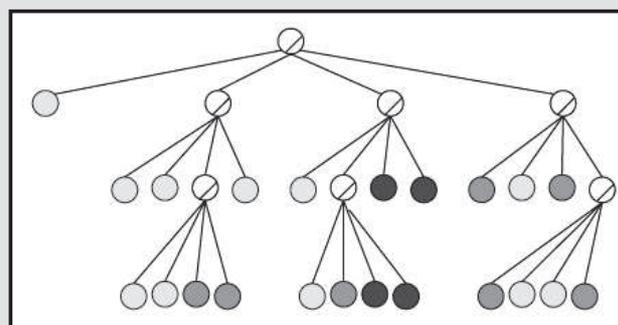


Image originale



Construction du Quad-tree correspondant

Segmentation par régions

■ Le partage de région - Propriétés

- ▶ La géométrie de découpage a une influence directe sur le résultat de segmentation
- ▶ Par exemple la méthode *quad-tree* fait apparaître des régions carrées
- ▶ Il existe d'autres type de partage (triangle, pyramide)
- ▶ Le choix du type de partage se fait en fonction des formes que l'on souhaite segmenter

Segmentation par régions

▶ Exemple d'exploitation d'un Quad-tree

➡ *Détection de variation d'intensité locale*

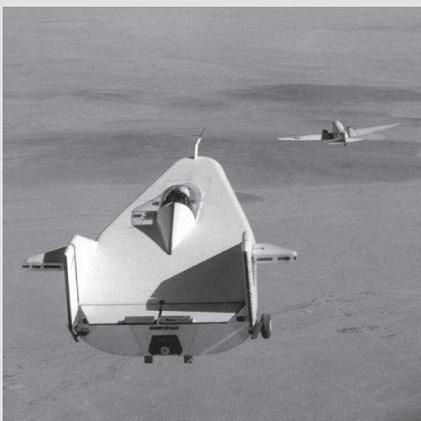
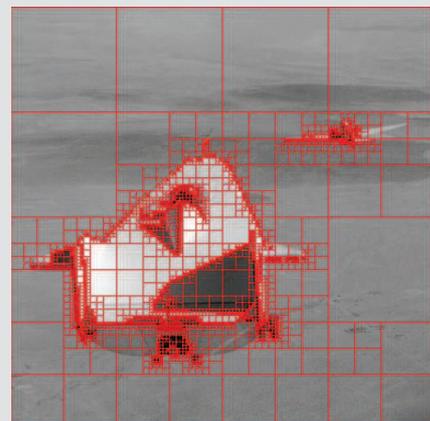


Image originale

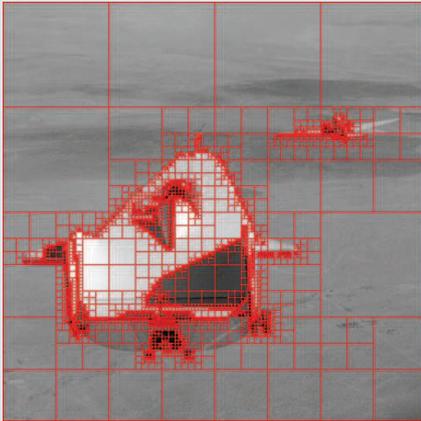


Construction du Quad-tree correspondant

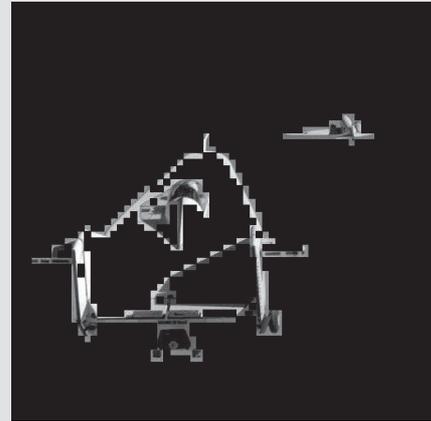
Segmentation par régions

▶ Exemple d'exploitation d'un Quad-tree

➡ *Idée: analyser que les plus petits blocs*



Construction du Quad-tree correspondant



Après analyse du Quad-tree

Segmentation par régions

■ Méthodes par transformation de régions

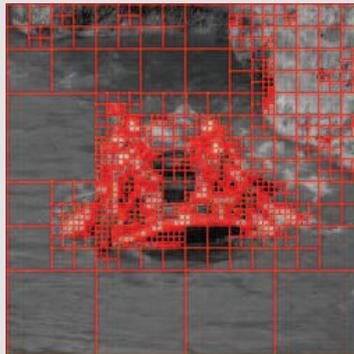
▶ Il existe trois grandes catégories

- La croissance de région
- Le partage de région (*quad-tree*)
- Les graphes d'adjacence

Segmentation par régions

■ Les graphes d'adjacence

Méthodes qui consiste à plonger les régions obtenues après segmentation dans une structure de graphe puis d'utiliser la notion de graphe pour effectuer des regroupements



Département GE - TI - Olivier Bernard

77

Segmentation par régions

■ Les graphes d'adjacence - Principe

- ▶ A partir d'une segmentation initiale, définition d'un graphe d'adjacence où une région est un nœud et un arc une relation d'adjacence
- ▶ Définition d'une fonction de similarité entre deux nœuds
- ▶ On trie tous les couples de nœuds adjacents dans une liste ordonnée
- ▶ On regroupe les deux meilleurs candidats
- ▶ On remet à jour la liste et on itère

Département GE - TI - Olivier Bernard

78

Segmentation par régions

■ Les graphes d'adjacence - Exemple

- ▶ Création d'un graphe au cours d'un *quad-tree*

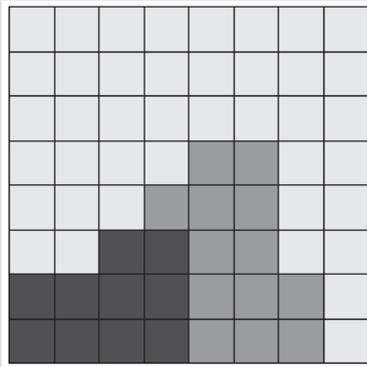
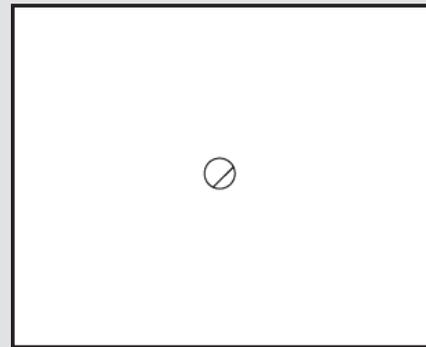


Image originale



Construction du Graphe d'adjacence correspondant

Segmentation par régions

■ Les graphes d'adjacence - Exemple

- ▶ Création d'un graphe au cours d'un *quad-tree*

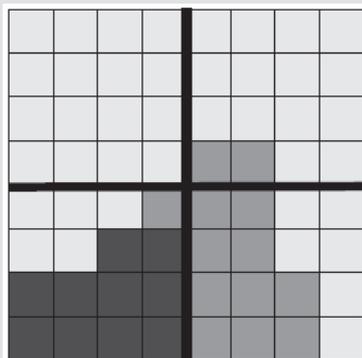
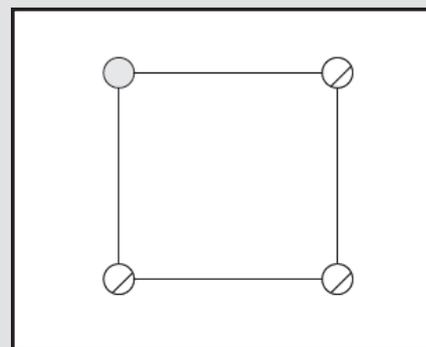


Image originale



Construction du Graphe d'adjacence correspondant

Segmentation par régions

■ Les graphes d'adjacence - Exemple

- ▶ Création d'un graphe au cours d'un *quad-tree*

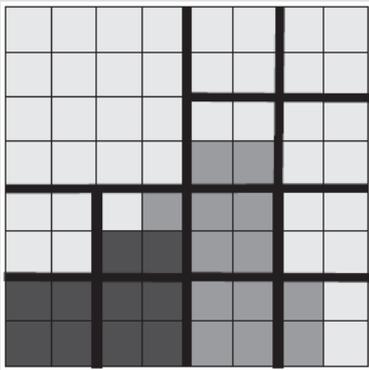
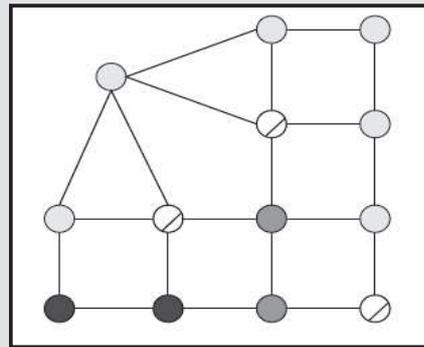


Image originale



Construction du Graphe d'adjacence correspondant

Segmentation par régions

■ Les graphes d'adjacence - Exemple

- ▶ Création d'un graphe au cours d'un *quad-tree*

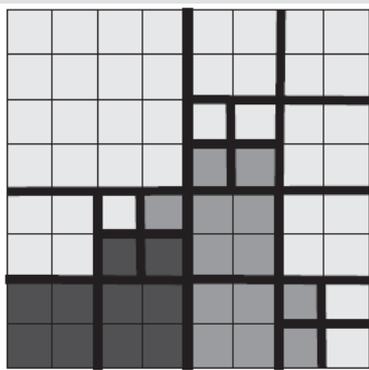
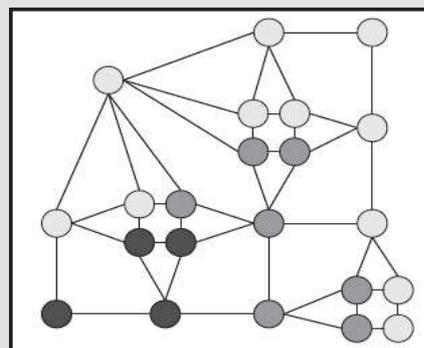


Image originale



Construction du Graphe d'adjacence correspondant

Segmentation par régions

■ Les graphes d'adjacence - Exemple

► Exploitation d'un graphe d'adjacence

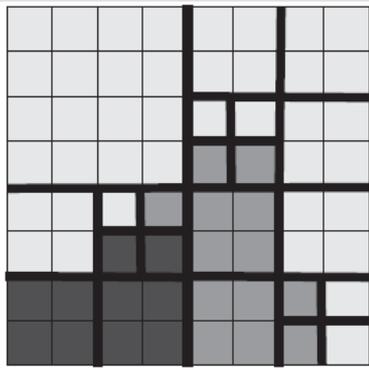
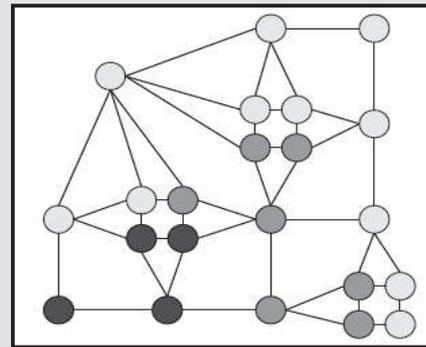


Image originale



Construction de la segmentation par régions correspondante

Segmentation par régions

■ Les graphes d'adjacence - Exemple

► Exploitation d'un graphe d'adjacence

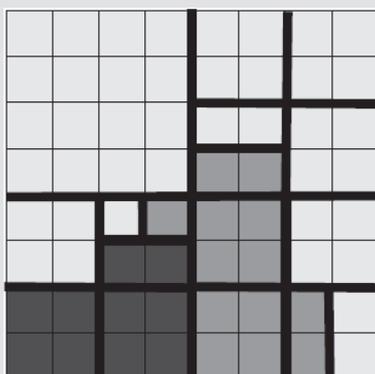
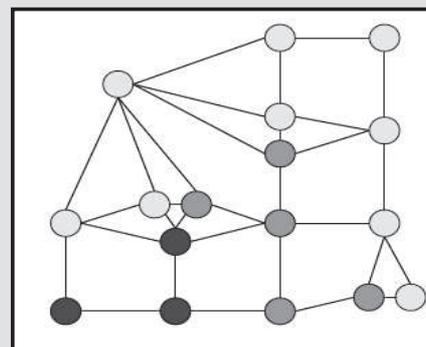


Image originale



Construction de la segmentation par régions correspondante

Segmentation par régions

■ Les graphes d'adjacence - Exemple

► Exploitation d'un graphe d'adjacence

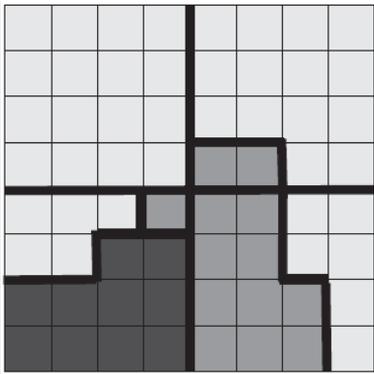
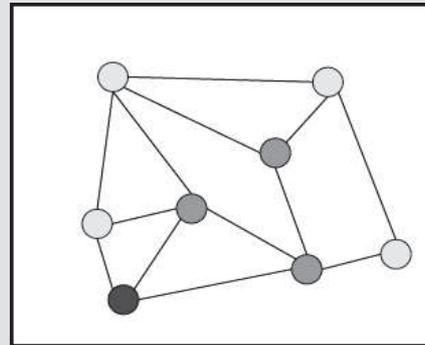


Image originale



Construction de la segmentation par régions correspondante

Segmentation par régions

■ Les graphes d'adjacence - Exemple

► Exploitation d'un graphe d'adjacence

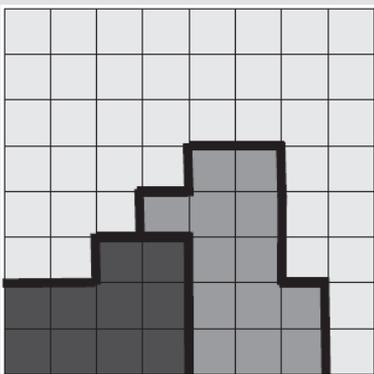
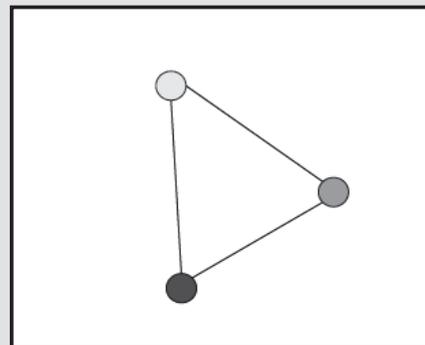


Image originale



Construction de la segmentation par régions correspondante

Segmentation par régions

■ Les graphes d'adjacence - Propriétés

- ▶ L'ordre dans lequel s'effectue le regroupement des régions a une influence sur le résultat
- ▶ En général, il est préférable de regrouper les petites régions en premier

Analyse d'images

Fin de la deuxième partie

ANNEXES

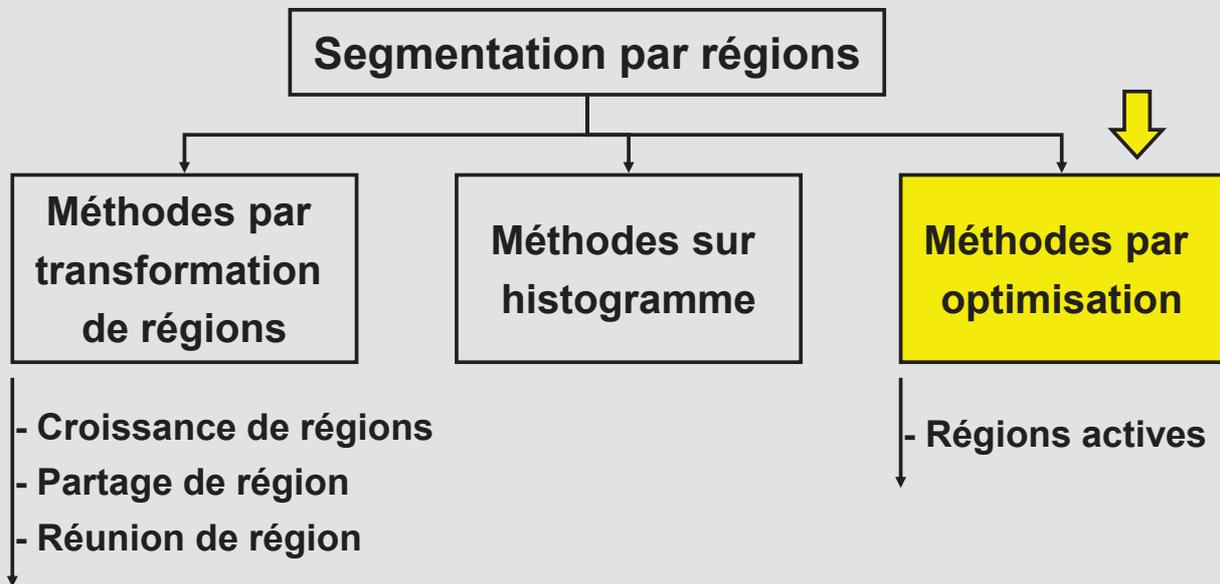
- ✓ *Méthodes par optimisation*
- ✓ *Exemple code matlab*

Segmentation par régions

- **Contexte**
 - Définition
 - Principe
- **Les différentes méthodes**
 - Méthodes par histogramme
 - Méthodes par transformation de régions
 - Méthodes par optimisation**

Segmentation par régions

■ Contexte – Les différentes méthodes



Segmentation par régions

■ Les méthodes par optimisation

Le problème de segmentation est formalisé par l'estimation d'une fonction f bidimensionnelle devant rester le plus proche possible de l'image I sous contraintes particulières

- ▶ Les contraintes peuvent être de type
 - régularité des contours
 - f constante par morceaux
 - régulière sur l'ensemble de définition

Segmentation par régions

■ Les méthodes par optimisation

▶ Principe

Le but des méthodes par optimisation est de trouver un compromis entre les différentes propriétés de la fonction, en minimisant une fonctionnelle d'énergie

➔ Similarité avec les contours actifs

Segmentation par régions

■ Les méthodes par optimisation

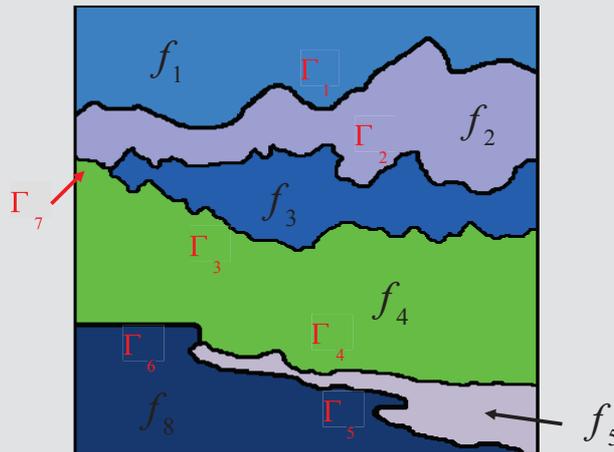
▶ Formalisme mathématique

- I : l'image à traiter
- $\Omega_{i \in P}$: la partition (segmentation) calculée
- $\Gamma_{i \in Q}$: les courbes frontières associées à la segmentation
- f : la fonction recherchée, représentant l'image I segmentée

Segmentation par régions

■ Les méthodes par optimisation

- ▶ La fonction f est représentée par ses restrictions f_i sur chaque région R_i : $f \equiv f_i \text{ }_{i \in P}$



Département GE - TI - Olivier Bernard

95

Segmentation par régions

■ Les méthodes par optimisation - Fonctionnelle

$$E(I, f, R, \Gamma) = \underbrace{\mu \sum_{i \in P} \iint_{R_i} |I(x, y) - f_i(x, y)|^2 dx dy}_{1} + \underbrace{\sum_{i \in P} \iint_{R_i} \|\nabla f_i(x, y)\|^2 dx dy}_{2} + \underbrace{\nu \sum_{j \in Q} \int_{\Gamma_j} ds}_{3}$$

- ① Terme de ressemblance à l'image initiale
- ② Terme de régularité des fonctions sur chaque région
- ③ Terme de simplicité géométrique (longueur des frontières)

Département GE - TI - Olivier Bernard

96

Segmentation par régions

■ Les méthodes par optimisation - Simplification

- ▶ On impose que chaque fonction f_i doit être constante sur la région R_i
- ▶ f_i est alors égale à la valeur moyenne de I sur la région R_i
- ▶ Le problème revient alors à minimiser la fonctionnelle suivante

$$E(I, f, R, \Gamma) = \sum_{i \in P} \iint_{R_i} I(x, y) - f_i(x, y)^2 dx dy + \frac{\nu}{\mu} \sum_{j \in Q} \int_{\Gamma_j} ds$$

Segmentation par régions

■ Les méthodes par optimisation - Propriétés

- ▶ Il n'existe pas de solution directe au problème de minimisation de la fonctionnelle d'énergie
- ▶ Implémentation de méthodes d'optimisation permettant d'obtenir un minimum local

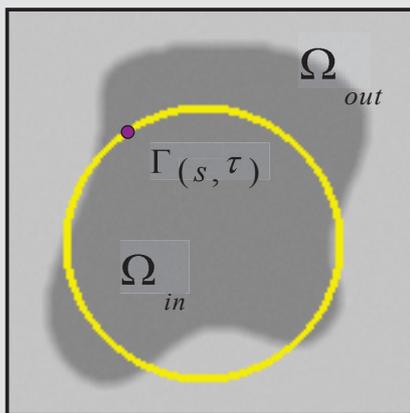


Méthodes variationnelles à partir de courbes fermées

Segmentation par régions

■ Exemple de méthode variationnelle

- ▶ Recherche de deux régions homogènes en niveaux de gris (par exemple l'objet et le fond de l'image)



Les contours actifs ?
Les régions actives !!

Segmentation par régions

■ Les régions actives

- ▶ Modélisation mathématique

1 - choix de la représentation du contour actif

2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter

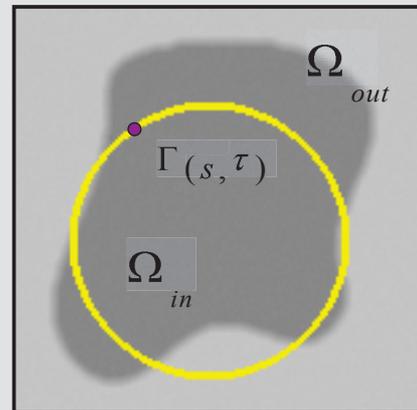
3 - évolution du contour guidée par la minimisation de la fonction d'énergie

Segmentation par régions – les régions actives

► Choix de la représentation du contour actif

Représentation continue

- $\Gamma(s, \tau) : [0, 1] \times [0, \infty[\rightarrow \mathbb{R}^2$
- $\Omega_{in}, \Omega_{out}$ région intérieure et extérieure à Γ



Segmentation par régions – les régions actives

■ Comment faire évoluer le contour actif ?

► Modélisation mathématique

1 - choix de la représentation du contour actif

2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter

3 - évolution du contour guidée par la minimisation de la fonction d'énergie

Segmentation par régions – les régions actives

► Fonction d'énergie

$$E(\Gamma) = \int_{\Omega_{in}} (I(x, y) - f_{in})^2 dx dy + \int_{\Omega_{out}} (I(x, y) - f_{out})^2 dx dy + \int_0^1 \left| \frac{\partial \Gamma(s)}{\partial s} \right| ds$$

avec

- I : image traitée
- f_{in} : valeur moyenne à l'intérieur du contour (à mettre à jour à chaque itération)
- f_{out} : valeur moyenne à l'extérieur du contour (à mettre à jour à chaque itération)

Segmentation par régions – les régions actives

■ Comment faire évoluer le contour actif ?

► Modélisation mathématique

- 1 - choix de la représentation du contour actif
- 2 - conception d'une fonction énergie dont le minimum correspond aux contours de l'objet à segmenter
- 3 - évolution du contour guidée par la minimisation de la fonction d'énergie**

Segmentation par régions – les régions actives

- ▶ Évolution du contour guidée par la minimisation de la fonction d'énergie
 - Utilisation d'outil mathématique de calcul variationnel

$$\frac{\partial \Gamma(\tau)}{\partial \tau} = V(x, y) \vec{N}$$

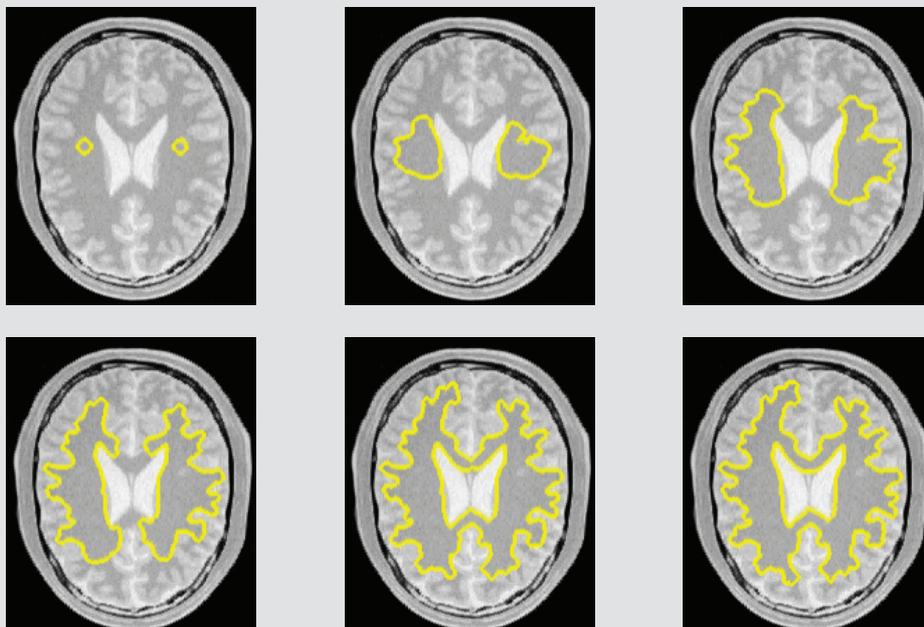
$$\text{avec } V(x, y) = \underbrace{\kappa}_{\text{terme de lissage}} + \underbrace{(I(x, y) - f_{in})^2 - (I(x, y) - f_{out})^2}_{\text{terme d'attache aux données}}$$

terme de
lissage

terme d'attache
aux données

Segmentation par régions – les régions actives

- ▶ Évolution du contour guidée par la minimisation de la fonction d'énergie



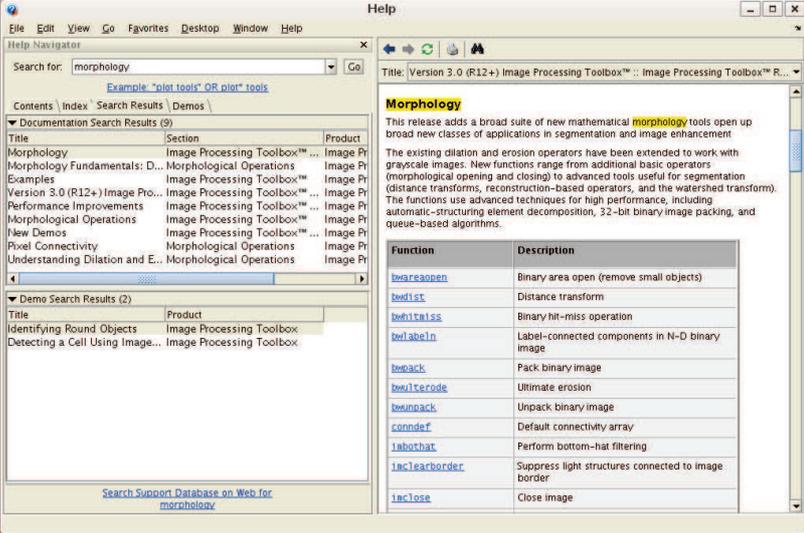
Analyse d'images - Annexes

- **L'ensemble des résultats du cours obtenus en utilisant matlab est téléchargeable à l'adresse internet suivante**

www.creatis.insa-lyon.fr/~bernard/AnalyseImage

Analyse d'images - Annexes

- **Liste des fonctions de morphologie mathématique accessible dans la documentation matlab en écrivant « morphology » dans l'onglet content**



The screenshot shows the MATLAB Help Navigator window. The search bar contains 'morphology'. The 'Contents' tab is active, displaying a list of search results. The results are organized into two sections: 'Documentation Search Results (9)' and 'Demo Search Results (2)'. The 'Documentation Search Results' section lists various topics related to morphology, such as 'Morphology Fundamentals', 'Examples', 'Performance Improvements', 'Morphological Operations', 'New Demos', 'Pixel Connectivity', and 'Understanding Dilation and Erosion'. The 'Demo Search Results' section lists 'Identifying Round Objects' and 'Detecting a Cell Using Image...'. The right pane shows the 'Morphology' section of the documentation, which includes a description of the new mathematical morphology tools and a table of functions.

Function	Description
bwareopen	Binary area open (remove small objects)
bwdist	Distance transform
bwhitmiss	Binary hit-miss operation
bwlabeled	Label-connected components in N-D binary image
bwpack	Pack binary image
bwulterode	Ultimate erosion
bunpack	Unpack binary image
conndef	Default connectivity array
inbothat	Perform bottom-hat filtering
inlearborder	Suppress light structures connected to image border
inlclose	Close image

Analyse d'images - Annexes

► Code matlab de la fonction hystérésis (connexité 4)

```
function out = hysteresis(in1,in2)

out = in2; count = 1; k=0;
MAXITERATION = 200;
while ( ( count ~= 0 ) && ( k < MAXITERATION ) )
    count = 0;
    for i=2:(size(out,1)-1)
        for j=2:(size(out,2)-1)
            if ( out(i,j) > 0 )
                if ( in1(i-1,j) > 0 )
                    out(i-1,j) = 255; count = count + 1;
                end
                if ( in1(i+1,j) > 0 )
                    out(i+1,j) = 255; count = count + 1;
                end
                if ( in1(i,j-1) > 0 )
                    out(i,j-1) = 255; count = count + 1;
                end
                if ( in1(i,j+1) > 0 )
                    out(i,j+1) = 255; count = count + 1;
                end
            end
        end
    end
    k = k + 1;
end
```

Traitement d'images numériques

Analyse d'images: 3^{ème} partie

*Département Génie Electrique
5GE - TdSi*

Sommaire

- I. Introduction
- II. Fondamentaux en imagerie numérique
- III. Traitement discret 2D
- IV. Amélioration d'image

V. Analyse d'images (6 heures)

Analyse d'images

■ Sources qui ont inspirées cette partie:

- ★ *Henri Maître (2003) : Le traitement des images, Hermes-Lavoisier*
- ★ *J.P. Cocquerez (1995) : Analyse d'images: filtrage et segmentation, Hermes-Lavoisier*
- ★ *http://www.ensta.fr/~manzaner/Support_Cours.html*
- ★ *<http://www.tsi.enst.fr/~tupin/TDI.html>*

Sommaire

V. Analyse d'images

- ✓ *1^{ère} partie*
 - Morphologie mathématique
 - Détection et analyse de contours
- ✓ *2^{ème} partie*
 - Segmentation par régions
- ✓ *3^{ème} partie*
 - Analyse de formes
 - Reconnaissance de formes

Sommaire

V. Analyse d'images

- ✓ *1^{ère} partie*
 - Morphologie mathématique
 - Détection et analyse de contours

- ✓ *2^{ème} partie*
 - Segmentation par régions

- ✓ *3^{ème} partie*
 - **Analyse de formes**
 - Reconnaissance de formes

Analyse de formes

- **Contexte**
 - Définition
 - Propriétés

- **Les méthodes classiques**
 - Boîtes englobantes
 - Descripteur de Fourier
 - Autres

Analyse de formes

■ Contexte - Définition

Etape qui consiste à reconnaître un objet à partir de sa seule silhouette (forme)

- ▶ Généralement intervient après une étape de segmentation d'une image

Analyse de formes

■ Contexte - Définition

- ▶ L'étape d'analyse de forme est basée sur l'utilisation d'un descripteur de forme
- ▶ Il existe de nombreux descripteurs de formes
 - boîtes englobantes
 - représentation par les moments
 - polygones de Guzman
 - chaînes de Freeman
 - descripteur de Fourier

Analyse de formes

■ Contexte - Propriétés

- ▶ **Un bon descripteur de formes doit posséder les propriétés suivantes :**
 - **bonne fidélité à la forme initiale**
 - **bonne discrimination de formes différentes**
 - **bonne adaptation aux opérations de reconnaissance de formes :**
 - **invariance par translation**
 - **invariance par rotation**
 - **invariance par changement d'échelle**

Analyse de formes

■ Contexte - Applications

- ▶ **L'étape de reconnaissance d'objet peut intervenir dans différents domaines :**
 - **application de tri (objets arrivant sur un convoyeur)**
 - **application de reconnaissance de caractères**
 - **application de surveillance ou de guidage**

Analyse de formes

■ Contexte

- Définition
- Propriétés

■ Les méthodes classiques

- Boîtes englobantes
- Descripteur de Fourier
- Autres

Analyse de formes

■ Boîtes englobantes

Création d'une boîte qui englobe l'objet à décrire et dont ses axes sont alignés avec les axes de l'image

- ▶ **Ce descripteur est simplement défini par ses dimensions**

$$\begin{cases} (x_{\max} - x_{\min}) \\ (y_{\max} - y_{\min}) \end{cases}$$

Analyse de formes

■ Boîtes englobantes - Exemple

- ▶ Détection de structures filaires



Image binaire obtenue après segmentation

Analyse de formes

■ Boîtes englobantes - Exemple

- ▶ 1^{ère} étape: étiquetage des objets



Algorithmes utilisés

- Morphologie mathématique
(étiquetage)

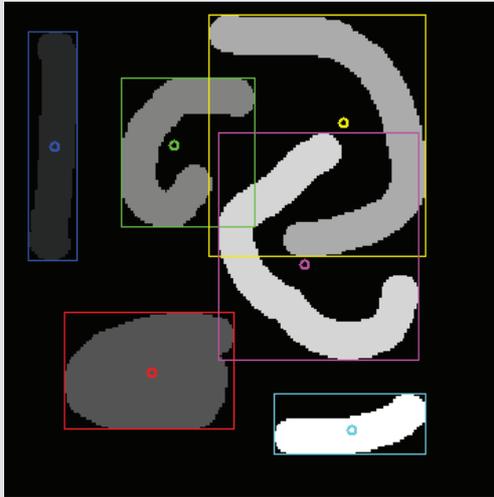
```
% Lecture image
img = imread('pieces.png');
img = double(img(:,:,1));

% creation de l'etiquetage
result = bwlabel(img,8);
```

Analyse de formes

■ Boîtes englobantes - Exemple

- ▶ 2^{ème} étape: description de chaque forme



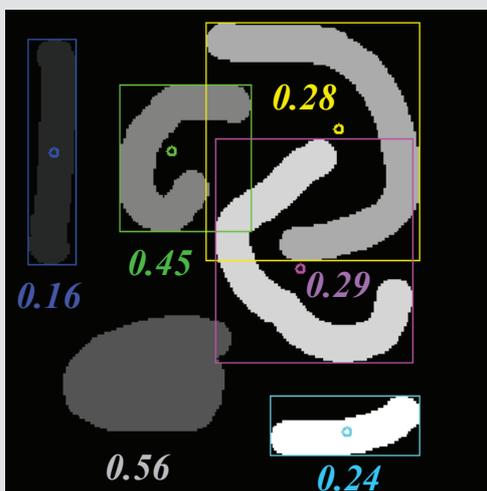
Algorithmes utilisés

- Description par boîtes englobantes

Analyse de formes

■ Boîtes englobantes - Exemple

- ▶ 3^{ème} étape: calcul de paramètres discriminants



Paramètre utilisé

- Rapport des surfaces objet / carré englobant
- On garde les objets dont le paramètre correspondant est inférieur à 0.5

Analyse de formes

■ Boîtes englobantes - Propriétés

- ▶ Descripteurs utilisés lorsque les formes à étudier sont très irrégulières
- ▶ Invariances en translation uniquement

Analyse de formes

■ Contexte

- Définition
- Propriétés

■ Les méthodes classiques

- Boîtes englobantes
- Descripteur de Fourier
- Autres

Analyse de formes

■ Descripteur de Fourier

Outil permettant de décrire l'enveloppe externe d'une forme, c'est-à-dire son contour

- ▶ Il existe deux types de descripteur de Fourier
 - descripteur par tangente
 - descripteur par représentation complexe

Analyse de formes

■ Descripteur de Fourier

Outil permettant de décrire l'enveloppe externe d'une forme, c'est-à-dire son contour

- ▶ Il existe deux types de descripteur de Fourier
 - descripteur par tangente
 - descripteur par représentation complexe

Analyse de formes

■ Descripteur complexe - Principe

- ▶ Description discrète de la forme par un ensemble $\{M_i\}$ de points du contour
- ▶ Représentation de la forme dans le plan complexe : on associe à chaque point M_i un nombre complexe $z_i = x_i + j y_i$

Analyse de formes

■ Descripteur complexe - Principe

- ▶ On appelle alors descripteurs de Fourier, les coefficients de la transformée de Fourier discrète Z de z

$$Z_k = \frac{1}{N} \sum_{i=0}^{N-1} z_i \exp\left(-j2\pi \frac{ik}{N}\right)$$

Avec $k \in 0, N-1$

Analyse de formes

■ Descripteur complexe - Propriétés

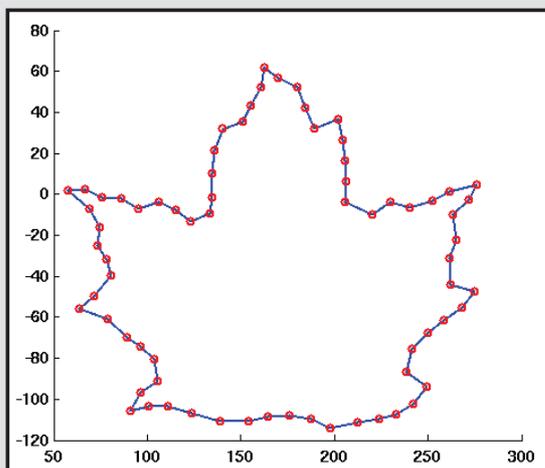
▶ Influence du coefficient Z_0

- Z_0 est le centre de gravité de la forme
- Si l'on impose $Z_0 = 0$, la description est invariante par translation

Analyse de formes

■ Descripteur complexe - Propriétés

▶ Influence du coefficient Z_0



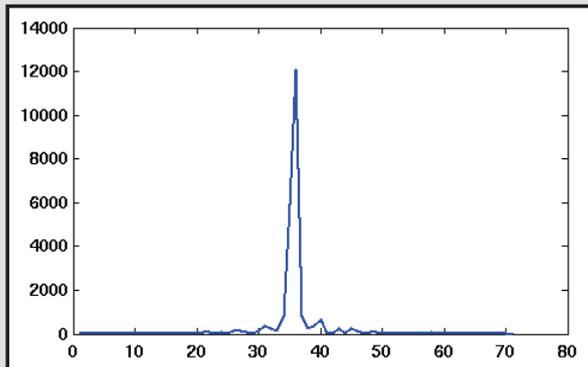
Contour initial

```
% Creation d'une liste de nombres  
% complexes  
ListCmp = complex(ListPoint(:,1), ...  
                  ListPoint(:,2));  
  
% Affichage de cette liste  
figure; plot(ListCmp, 'or');
```

Analyse de formes

■ Descripteur complexe - Propriétés

► Influence du coefficient Z_0



Spectre du contour initial

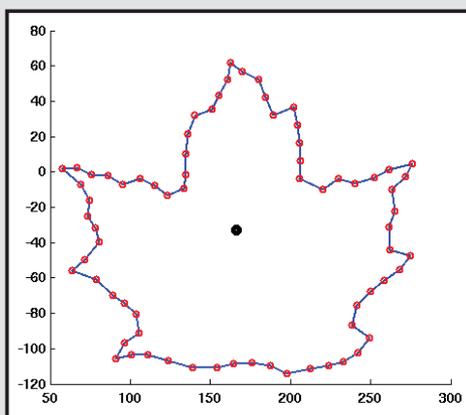
```
% Calcul du spectre correspondant
FD = fft(ListCmp);

% Affichage du spectre
figure; plot(abs(fftshift(FD)));
```

Analyse de formes

■ Descripteur complexe - Propriétés

► Influence du coefficient Z_0



*Contour initial + contour reconstruit
(noir) correspondant au centre de
gravité du contour*

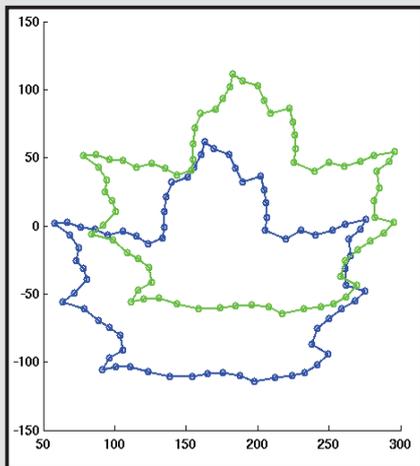
```
% Calcul du nouveau spectre
FDNew = FD;
FDNew(2:end) = 0;

% Affichage des contour correspondant
figure; plot(iff(FD), 'or');
hold on; plot(iff(FDNew), 'ok');
```

Analyse de formes

■ Descripteur complexe - Propriétés

► Influence du coefficient Z_0



*Contour initial (bleu) +
contour translaté (vert)*

```
% Calcul du contour translaté
ListTransCmp(:,1) = ListCmp(:,1) + 20;
ListTransCmp(:,2) = ListCmp(:,2) + 50;

% Affichage des contours correspondant
figure; plot(ListCmp,'ob');
hold on; plot(ListTransCmp,'og');

% Calcul du spectre correspondant
FD = fft(ListCmp);
FDTrans = fft(ListTransCmp);

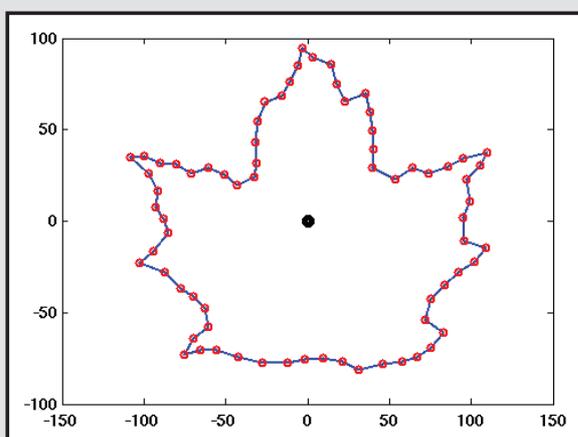
% invariance par translation
FD(1) = 0;
FDTrans(1) = 0;
```

27

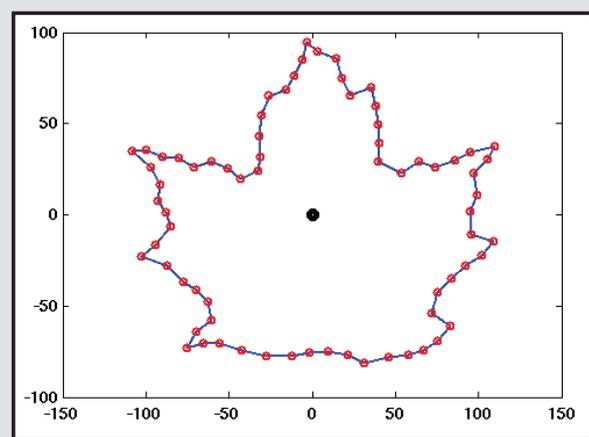
Analyse de formes

■ Descripteur complexe - Propriétés

► Influence du coefficient Z_0



*Reconstruction du contour initial
après invariance par translation*



*Reconstruction du contour
translaté après invariance par
translation*

28

Analyse de formes

■ Descripteur complexe - Propriétés

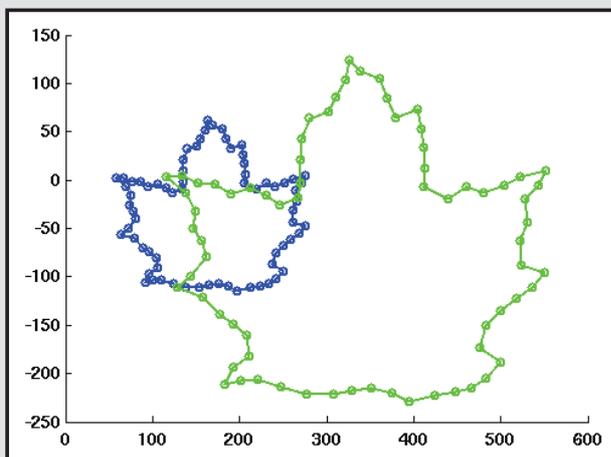
► Influence du coefficient Z_1

- Si tous les coefficients Z_k sont nuls sauf pour $k=1$, la forme est un cercle de rayon Z_1
- Z_1 joue le rôle de facteur d'échelle
- La normalisation par $|Z_1|$ rend la forme invariante par homothétie

Analyse de formes

■ Descripteur complexe - Propriétés

► Influence du coefficient Z_1



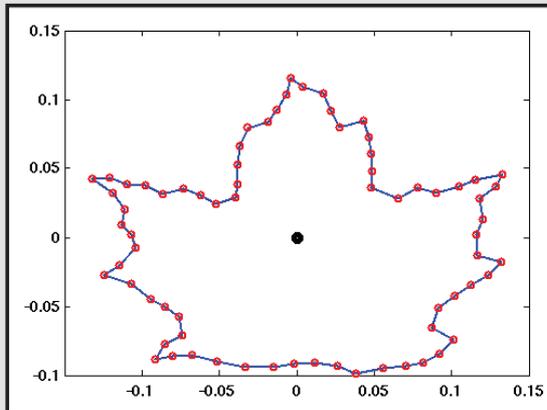
Contour initial (bleu) + contour avec changement d'échelle (vert)

```
% Calcul du contour par changement  
% d'échelle  
ListCECmp(:,1) = ListCmp(:,1) * 2;  
ListCECmp(:,2) = ListCmp(:,2) * 2;  
  
% Affichage des contours correspondant  
figure; plot(ListCmp,'ob');  
hold on; plot(ListCECmp,'og');  
  
% Calcul du spectre correspondant  
FD = fft(ListCmp);  
FDCE = fft(ListCECmp);  
  
% invariance par translation  
FD(1) = 0; FDCE(1) = 0;  
  
% invariance par changement d'échelle  
FD = FD / abs(FD(2));  
FDCE = FDCE / abs(FDCE(2));
```

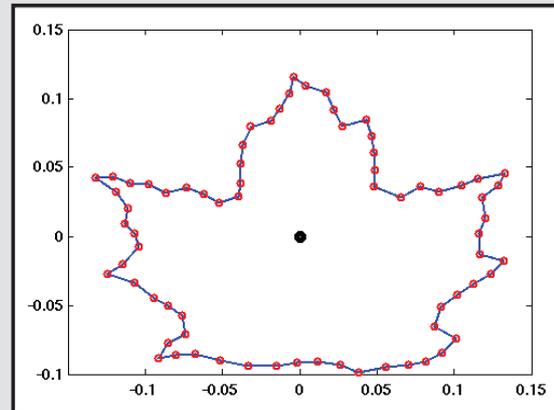
Analyse de formes

■ Descripteur complexe - Propriétés

▶ Influence du coefficient Z_1



Reconstruction du contour initial après invariance par translation et changement d'échelle



Reconstruction du contour modifié après invariance par translation et changement d'échelle

31

Analyse de formes

■ Descripteur complexe - Propriétés

▶ Etude des coefficients Z_k

- Les coefficients Z_k et Z_{1-k} (pour $k \neq 0$ et $k \neq 1$) jouent des rôles symétriques mais opposés
- L'ordre k indique le nombre d'actions sur le cercle unité (1 action pour $k=2$ et $k=-1$, 2 actions pour $k=3$ et $k=-2$, etc)
- Les actions sont réparties régulièrement autour du cercle unité

Analyse de formes

■ Descripteur complexe - Propriétés

▶ Etude des coefficients Z_k

- Les valeurs de $k > 0$ indiquent des actions de traction sur la courbe, pour la déformer vers l'extérieur du cercle unité
- Les valeurs de $k < 0$ indiquent des actions de pression sur la courbe, pour creuser la courbe vers son centre

Analyse de formes

■ Descripteur complexe - Propriétés

▶ Etude des coefficients Z_k

- La phase du nombre complexe Z_k exprime le lieu, sur le cercle unité, où s'exerce l'action

Analyse de formes

■ Descripteur complexe - Propriétés

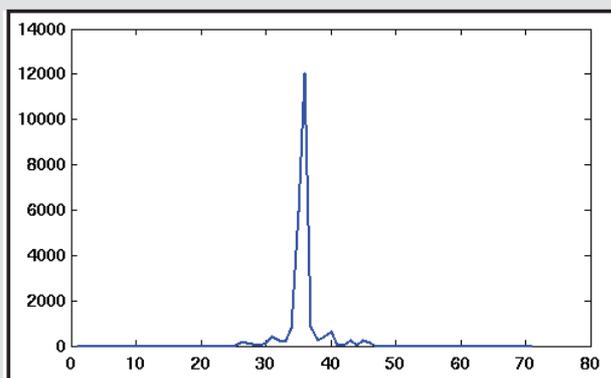
▶ Etude des coefficients Z_k

- La troncature du développement de Fourier permet de représenter des formes plus lisses qui restent fermées (on diminue le nombre d'actions exercées sur la courbe)

Analyse de formes

■ Descripteur complexe - Propriétés

▶ Etude des coefficients Z_k



Spectre du contour simplifié

```
% Calcul du spectre correspondant
FDS = fftshift(fft(ListCmp));

% Simplification du spectre
% correspondant
FDS(1:25) = 0;
FDS(47:71) = 0;

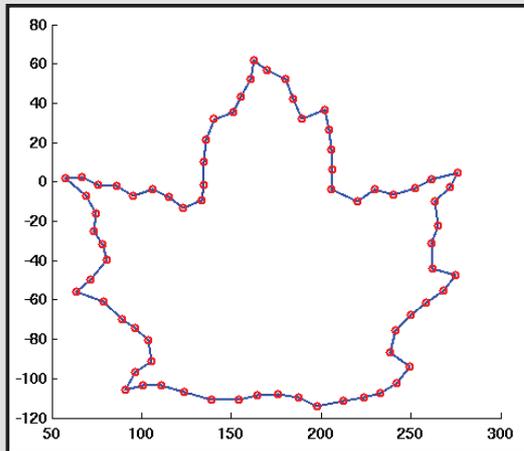
% Reconstruction du contour
Contour = ifft(ifftshift(FDS));

% affichage du contour reconstruit
figure; plot(Contour, 'or');
```

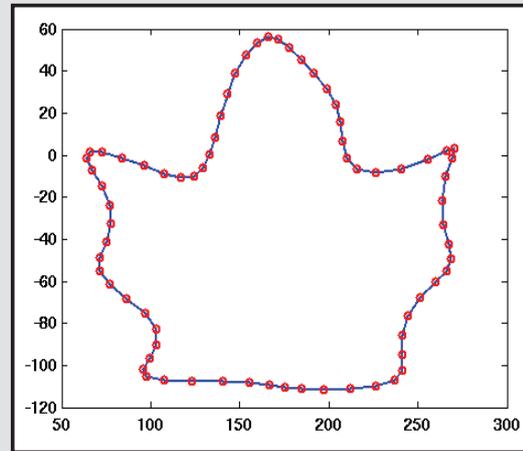
Analyse de formes

■ Descripteur complexe - Propriétés

► Etude des coefficients Z_k



*Reconstruction du contour
avant simplification*



*Reconstruction du contour
après simplification*

Sommaire

V. Analyse d'images

- ✓ *1^{ère} partie*
 - Morphologie mathématique
 - Détection et analyse de contours
- ✓ *2^{ème} partie*
 - Segmentation par régions
- ✓ *3^{ème} partie*
 - Analyse de formes
 - **Reconnaissance de formes**

Reconnaissance de formes

- **Contexte**

- Définition

- **Les méthodes classiques**

- Approximation d'une droite

- Transformation de Hough

Reconnaissance de formes

- **Contexte - Définition**

A partir d'un nuage de points (pixels) obtenu après une phase de traitement, on cherche à retrouver une forme connue a priori qui décrit « au mieux » les points extraits

Reconnaissance de formes

- Contexte

- Définition

- Les méthodes classiques

- Approximation d'une droite

- Transformation de Hough

Reconnaissance de formes

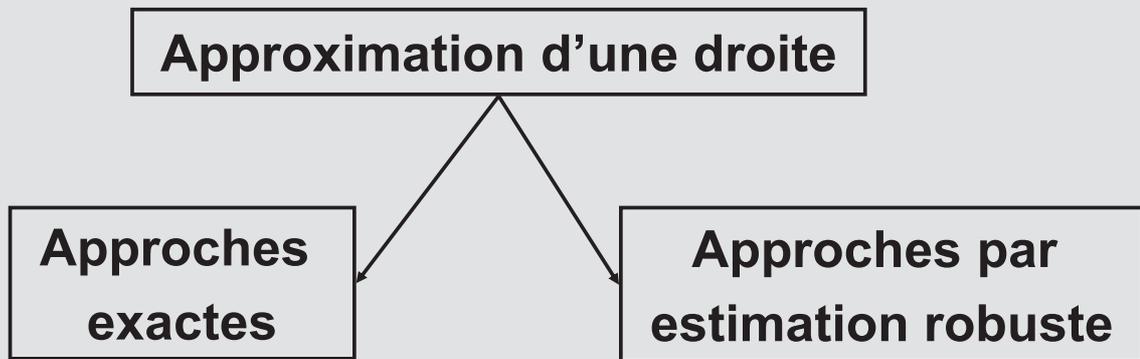
- Approximation d'une droite

Méthodes basées sur une approche par moindres carrés obtenue à partir de la minimisation d'une distance

- ▶ Il existe de nombreuses méthodes pour résoudre ce problème apparemment simple

Reconnaissance de formes

■ Approximation d'une droite - familles



Reconnaissance de formes

■ Méthodes exactes

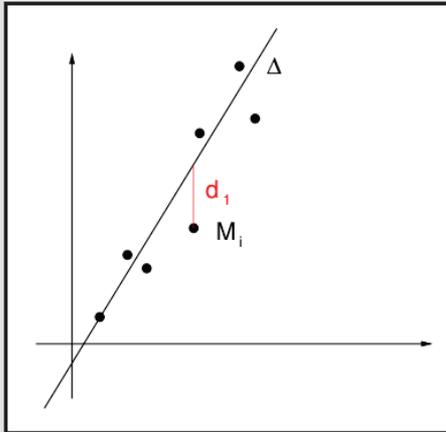
On considère que tous les points extraits participent avec le même poids à la description de la droite

- ▶ Il existe deux grandes méthodes
 - par régression linéaire
 - par axe principal d'inertie (annexes)

Reconnaissance de formes

■ Approximation par régression linéaire

- ▶ Recherche de la droite $\Delta: y = a_0 + a_1x$ qui minimise la distance suivante



$$d_1^2 = \sum_{i=1}^N y_i - (a_0 + a_1x_i)^2$$

Reconnaissance de formes

■ Approximation par régression linéaire

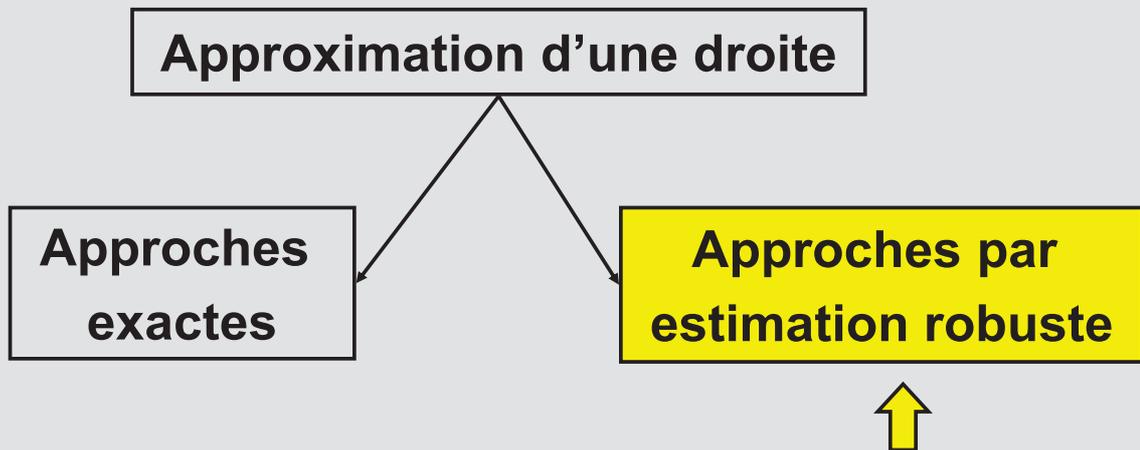
- ▶ La solution est simplement donnée à partir de la liste de points (x_i, y_i) par

$$A = (X^t X)^{-1} X^t Y$$

avec $X = \begin{bmatrix} 1 & x_1 \\ \dots & \dots \\ 1 & x_N \end{bmatrix}$ $Y = y_1, y_2, \dots, y_N^t$ $A = a_0, a_1^t$

Reconnaissance de formes

■ Approximation d'une droite - familles



Reconnaissance de formes

■ Approximation par estimation robuste

Recherche de la meilleure droite représentant au mieux l'ensemble des points sous l'hypothèse d'un bruit entachant la position des points

- ▶ Il existe de nombreuses méthodes, entre autres
 - estimation médiane
 - estimation d'un mélange de droites (annexes)

Reconnaissance de formes

■ Approximation par estimation médiane

▶ Principe (1/2)

- On choisit deux points M_i et M_j de l'ensemble
- On leur associe une droite qui fournit les coefficients a_{ij} et b_{ij}
- Par combinatoire sur l'ensemble des points, on obtient deux ensembles de coefficients $a = \{a_{ij}\}$ et $b = \{b_{ij}\}$

Reconnaissance de formes

■ Approximation par estimation médiane

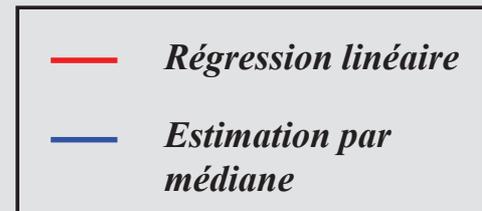
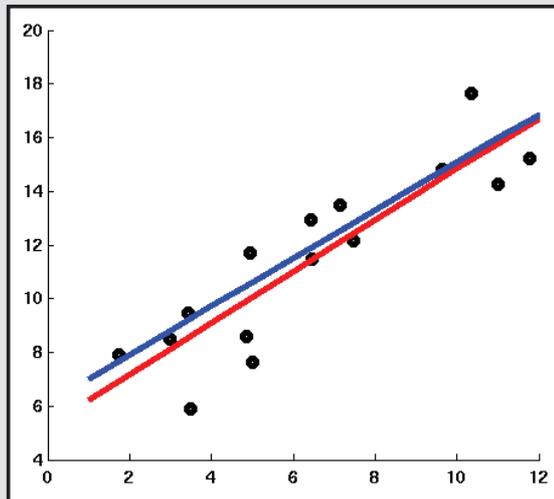
▶ Principe (2/2)

- Les ensembles a et b sont triés séparément et l'on choisit les valeurs médianes a_m et b_m comme estimateurs de paramètres de la droite
- Cette estimateur peut tolérer jusqu'à 50% de points erronés (perturbés par du bruit non gaussien)

Reconnaissance de formes

■ Approximation par estimation médiane

▶ Exemple



Reconnaissance de formes

■ Contexte

Définition

■ Les méthodes classiques

Approximation d'une droite

Transformation de Hough

Reconnaissance de formes

■ Transformation de Hough

On appellera transformation de Hough associée à une forme paramétrée par les variables $\{a_j\}$ la transformation qui fait passer de l'espace image I à l'espace de paramètre H

- ▶ S'applique à toute forme paramétrique (ex: droite, cercle, ellipse, paraboles, traces sinusoïdales)

Reconnaissance de formes

■ Transformation de Hough - Exemple

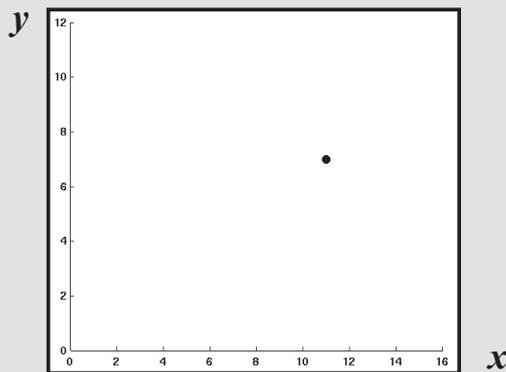
- ▶ Détection d'une droite (1/5)
 - On associe à un point $M_i = (x_i, y_i)$ de l'espace image I , toutes les droites du plan
 - Ces droites sont définies dans l'espace de paramètre H par $b = -ax_i + y_i$

Reconnaissance de formes

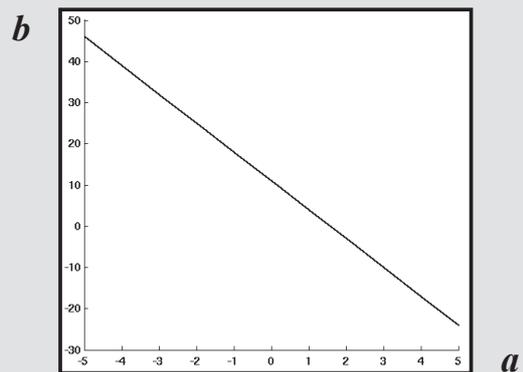
■ Transformation de Hough - Exemple

▶ Détection d'une droite (2/5)

- A un point de I est associé une droite dans H



Espace image



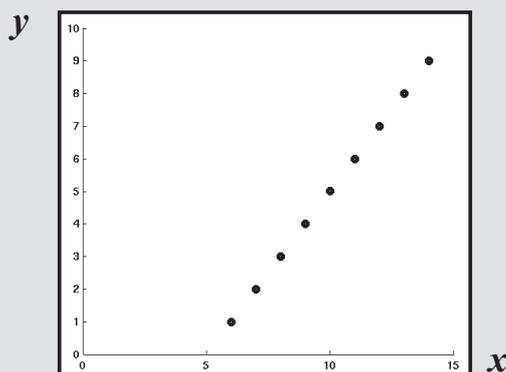
Espace de paramètre

Reconnaissance de formes

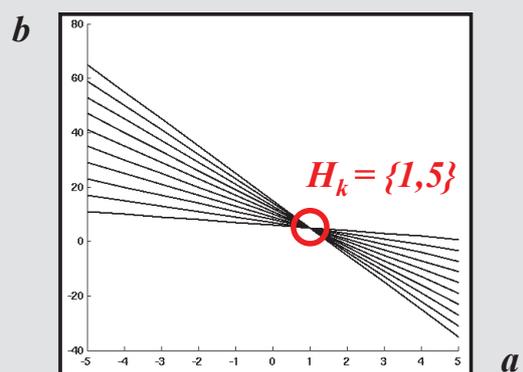
■ Transformation de Hough - Exemple

▶ Détection d'une droite (3/5)

- En appliquant ce procédé à tous les pixels de l'image, on associe à I un ensemble de droites qui, idéalement, se coupent en des points H_k



Espace image

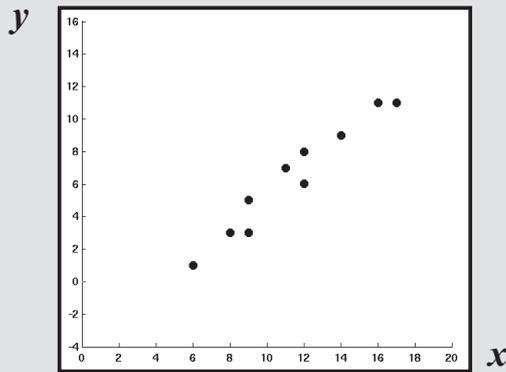


Espace de paramètre

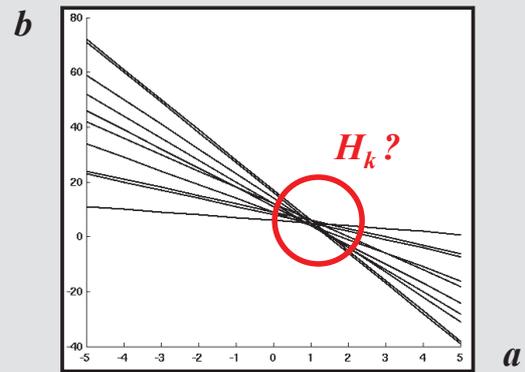
Reconnaissance de formes

■ Transformation de Hough - Exemple

▶ Détection d'une droite (3/5)



Espace image



Espace de paramètre

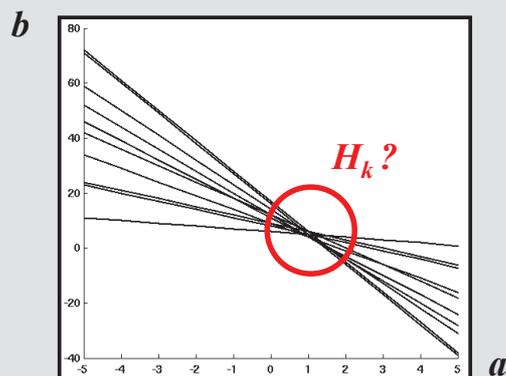
57

Reconnaissance de formes

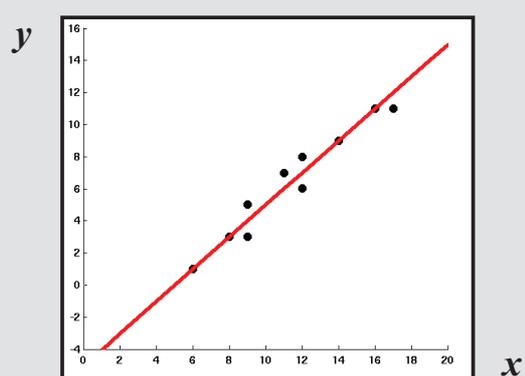
■ Transformation de Hough - Exemple

▶ Détection d'une droite (4/5)

- Les points d'intersection H_k correspondent à des droites recherchées dans l'espace I



Espace de paramètre



Espace image

58

Reconnaissance de formes

■ Transformation de Hough - Exemple

▶ Détection d'une droite (5/5)

- Il suffit de détecter dans l'espace de paramètre les points d'intersection (ou nuages de points denses) pour trouver les droites dans l'espace image

- On préférera comme espace de représentation ρ, θ

$$\rho = x_i \cos \theta + y_i \sin \theta$$

➔ maîtrise des bornes de l'espace H

$$\Delta\theta = -\pi, +\pi \quad \text{et} \quad \Delta\rho = \left[0, \sqrt{2}L \right], \quad \text{où } L \text{ est le coté de l'image}$$

Reconnaissance de formes

■ Transformation de Hough

▶ Exemple: étude d'un circuit imprimé

```
% Lecture image
img = imread('circuit.pgm'); img = double(img);

% Extraction du nuage de points correspondant
BW = edge(img, 'canny');
```

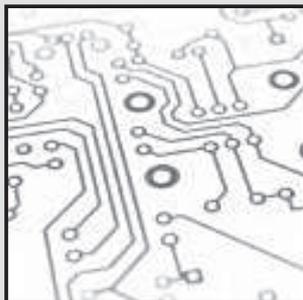


Image originale

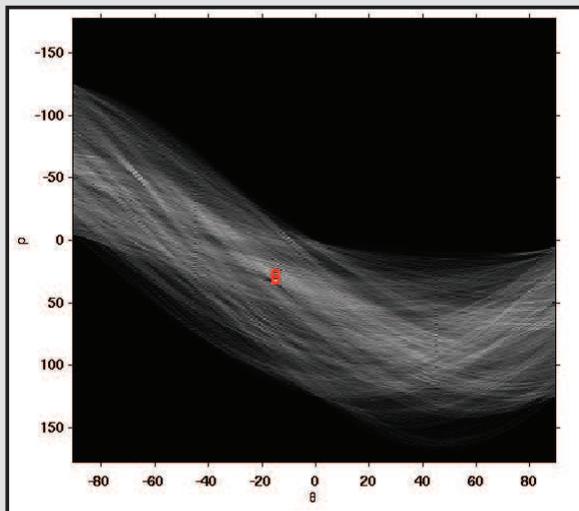


Image contour

Reconnaissance de formes

■ Transformation de Hough

▶ Exemple: étude d'un circuit imprimé



Espace de paramètre

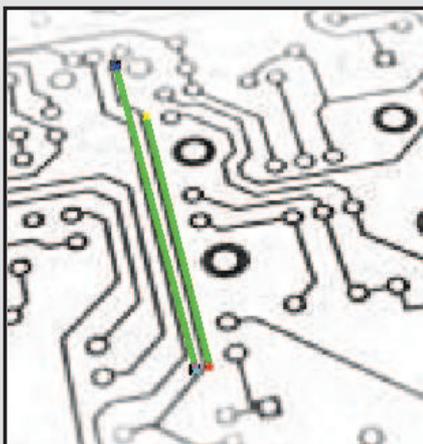
```
% Calcul de la transformée de hough  
[H,T,R] = hough(BW) ;  
  
% Détection des pics dans l'espace de hough  
P = houghpeaks(H,2,'threshold', ...  
    ceil(0.8*max(H(:)))) ;
```

61

Reconnaissance de formes

■ Transformation de Hough

▶ Exemple: étude d'un circuit imprimé



Les 2 courbes les plus probables extraites

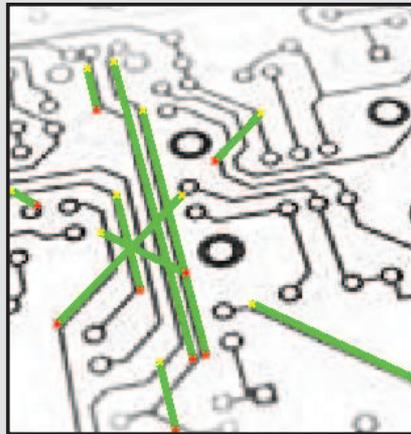
```
% Extraction des lignes correspondantes  
lines = houghlines(BW,T,R,P, ...  
    'FillGap',5,'MinLength',8) ;
```

62

Reconnaissance de formes

■ Transformation de Hough

- ▶ Exemple: étude d'un circuit imprimé - difficultés



Les 5 courbes les plus probables extraites

63

Reconnaissance de formes

■ Transformation de Hough

- ▶ Exemple: étude d'un circuit imprimé - idée

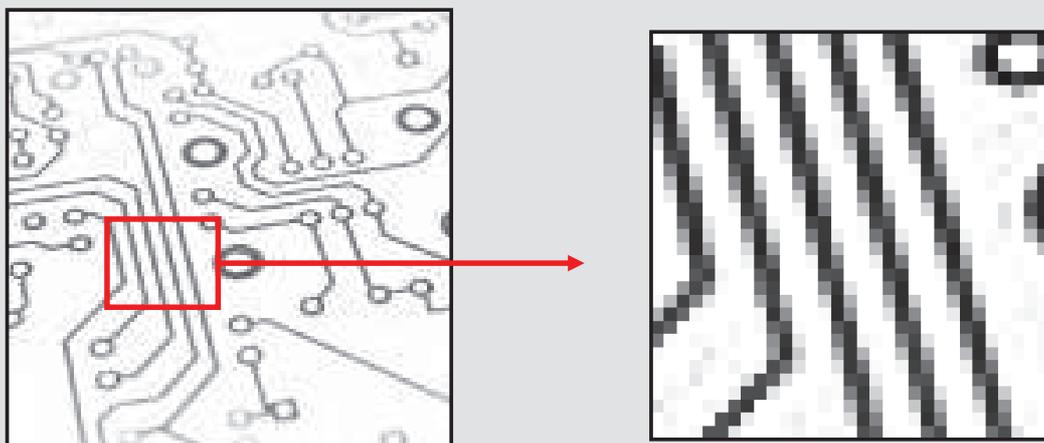


Image originale

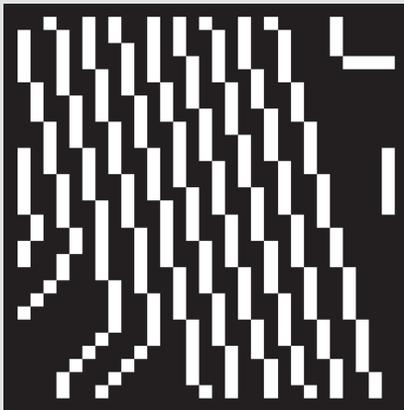
Image zoomée

64

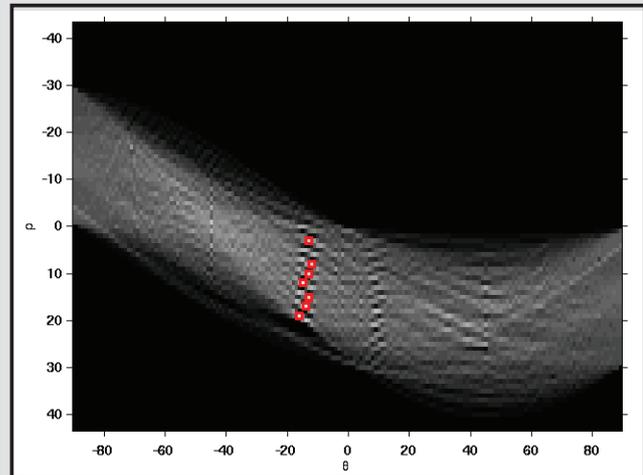
Reconnaissance de formes

■ Transformation de Hough

- ▶ Exemple: étude d'un circuit imprimé - idée



Espace Image



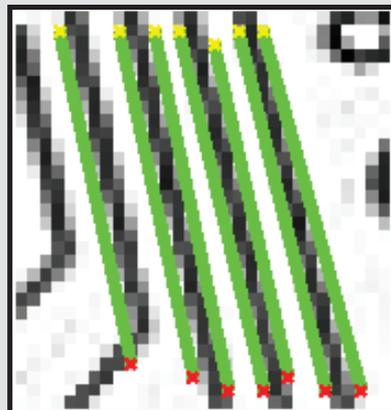
Espace de paramètre

65

Reconnaissance de formes

■ Transformation de Hough

- ▶ Exemple: étude d'un circuit imprimé - idée



Les 7 courbes les plus probables extraites

66

Reconnaissance de formes

■ Transformation de Hough - généralisation

▶ La méthode de transformation de Hough s'applique à toute forme paramétrique

- les paraboles (5 paramètres si l'on ne connaît pas leur axe)
- les cercles (3 paramètres)
- les ellipses (5 paramètres)
- les traces sinusoïdales (5 paramètres)

Analyse d'images

Fin de la troisième partie

ANNEXES

- ✓ *Descripteur de Fourier - tangente*
- ✓ *Complément sur l'approximation d'une droite*
- ✓ *Approximation de formes coniques*

ANNEXES

- ✓ *Descripteur de Fourier - tangente*
- ✓ *Acomplément sur l'approximation d'une droite*
- ✓ *Approximation de formes coniques*

Analyse de formes

■ Descripteur de Fourier

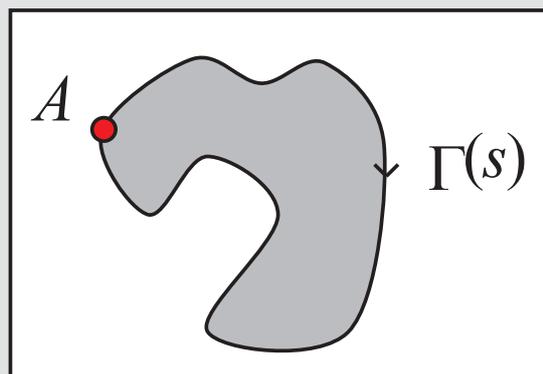
Outil permettant de décrire l'enveloppe externe d'une forme, c'est-à-dire son contour

- ▶ Il existe deux types de descripteur de Fourier
 - descripteur par tangente
 - descripteur par représentation complexe

Analyse de formes

■ Descripteur par tangente - Principe

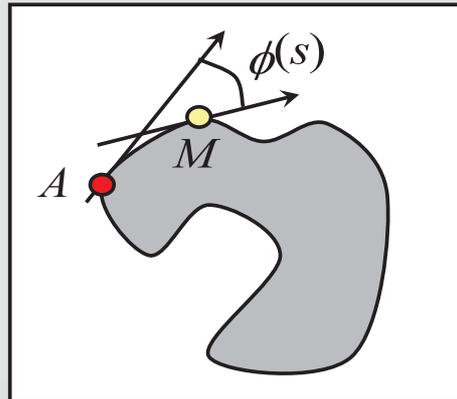
- ▶ On considère le contour de la forme à décrire comme une courbe continue définie par son abscisse curviligne s à partir d'une origine A



Analyse de formes

■ Descripteur par tangente - Principe

- ▶ Paramétrisation de la courbe par l'angle fait par le vecteur tangent en chaque point et celui au point origine $\phi(s)$



73

Analyse de formes

■ Descripteur par tangente - Expression

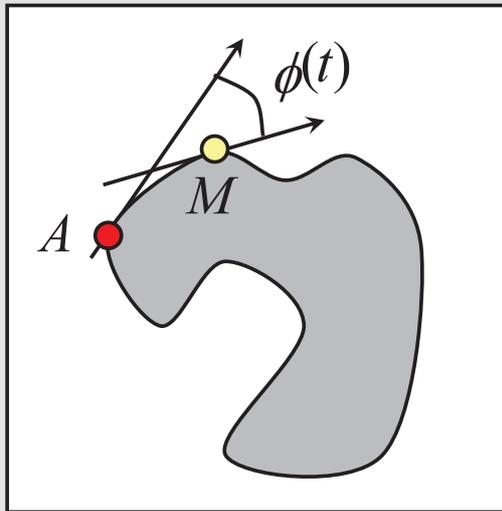
- ▶ Création de la variable réduite t qui prend ses valeurs entre 0 et 2π

$$t = \frac{2\pi \cdot s}{L}$$

Avec L : longueur complète de la courbe

Analyse de formes

■ Descripteur par tangente - Expression



$$\phi(t) = \phi \left[\frac{2\pi \cdot s}{L} \right] - \frac{2\pi \cdot s}{L}$$

- Le terme correctif prend en compte l'enroulement de 2π de la tangente pour un tour du contour

Analyse de formes

■ Descripteur par tangente - Propriétés

► Descripteur invariant par :

- translation
- changement d'échelle (puisque t est normalisé)
- rotation (puisque l'on choisi la différence d'angle entre 2 tangente)
- changement d'origine A

Analyse de formes

■ Descripteur par tangente - Propriétés

- ▶ Pour comparer deux formes, on compare leurs descripteurs par ordre croissant
- ▶ Cependant, l'interprétation géométrique n'est pas évidente
- ▶ Idée



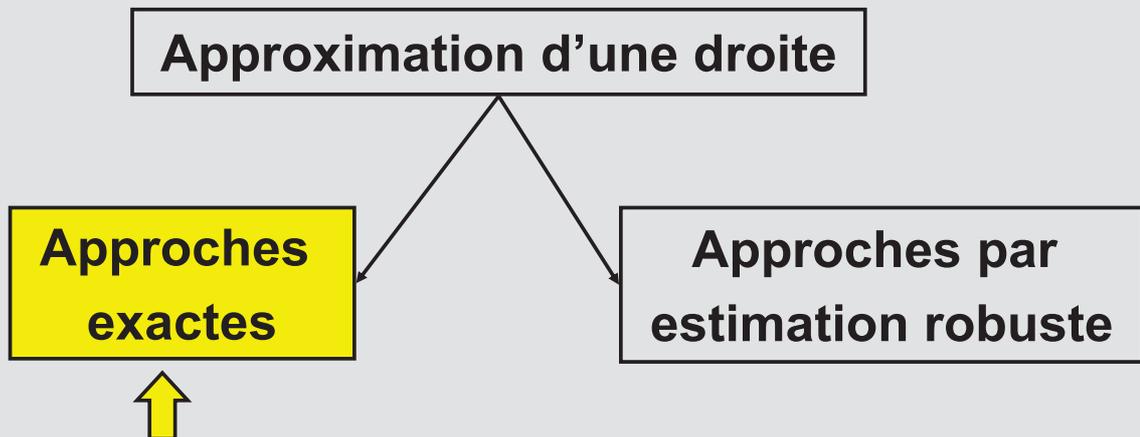
Création d'une autre représentation permettant une interprétation géométrique plus simple

ANNEXES

- ✓ *Descripteur de Fourier - tangente*
- ✓ **Complément sur l'approximation d'une droite**
- ✓ *Approximation de formes coniques*

Reconnaissance de formes

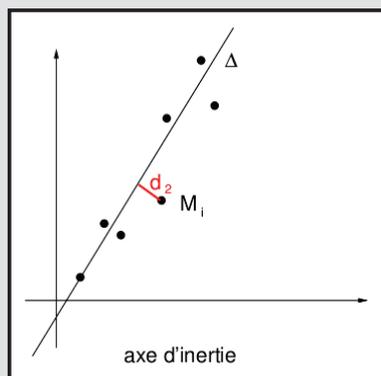
■ Approximation d'une droite - familles



Reconnaissance de formes

■ Approximation par axe principal d'inertie

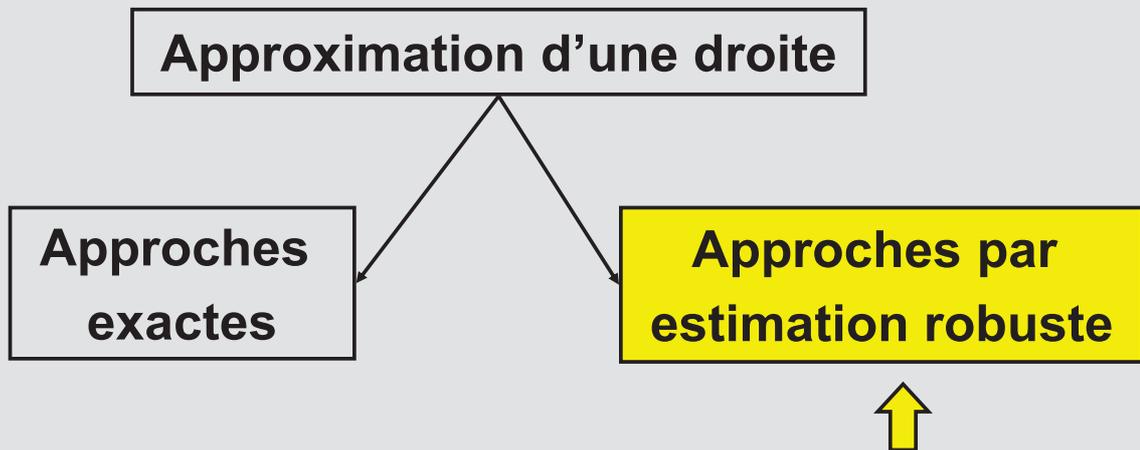
- ▶ Recherche de la droite $\Delta: y = a_0 + a_1x$ qui minimise la somme de la distance de tous les points à la droite \wedge



$$d_2^2 = \sum_{i=1}^N \frac{y_i - (a_0 + a_1x_i)^2}{a_1^2 + 1}$$

Reconnaissance de formes

■ Approximation d'une droite - familles



Reconnaissance de formes

■ Approximation par mélange de droites

▶ Principe (1/2)

- On initialise le processus en choisissant n droites (issues par exemple de l'ensemble des points par tirage aléatoire) représentées par leurs paramètres $a_0^i, a_1^i, i=1, \dots, n$
- On attribue chaque point à la droite qui minimise une distance (régression linéaire ou axe principal d'inertie)

Reconnaissance de formes

■ Approximation par mélange de droites

▶ Principe (2/2)

- Après classification de tous les points, on estime pour chaque droite ses paramètres a_0^i, a_1^i par une méthode classique
- On itère le processus enchaînant classification et estimation jusqu'à convergence

ANNEXES

- ✓ *Descripteur de Fourier - tangente*
- ✓ *Complément sur l'approximation d'une droite*
- ✓ *Approximation de formes coniques*

Reconnaissance de formes

■ Approximation de formes coniques

Extension des méthodes d'approximation d'une droite à la recherche de formes représentées par des cercles ou des ellipses

- ▶ Il existe de nombreuses méthodes pour résoudre ce problème

Reconnaissance de formes

■ Approximation de formes coniques

- ▶ Description d'une conique par une représentation matricielle

$$\left[(P - q_i)^t A_i (P - q_i) \right] = r_i^2$$

Avec : - q_i : centre de la conique

- P : point appartenant à la conique de coordonnées (x,y)

- A_i : matrice d'ellipticité

- r_i : rayon de la conique

Reconnaissance de formes

■ Approximation de formes coniques

- ▶ A partir d'un nuage de N points P_k , recherche de la conique $\{q_i, A_i, r_i\}$ qui minimise la distance suivante

$$d_3^2 = \sum_{k=1}^N \left[(P_k - q_i)^t A_i (P_k - q_i)^{1/2} - r_i^2 \right]^2$$