Introduction to image processing

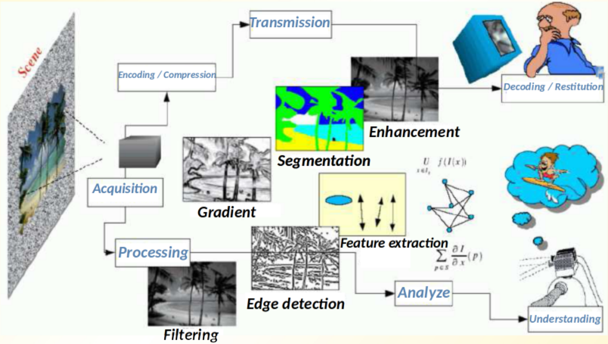
Youssef El Rhabi

Centrale
CASABLANCA

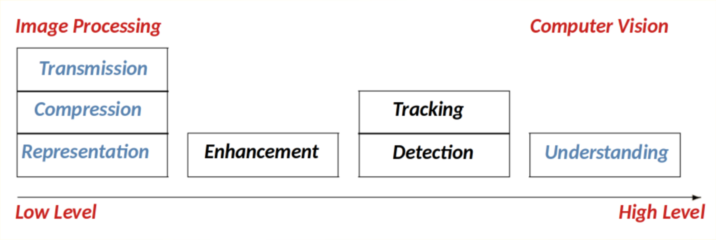
Plan

1. Introduction
2. Linear and non linear filtering
3. Image restoration

A Global Vision

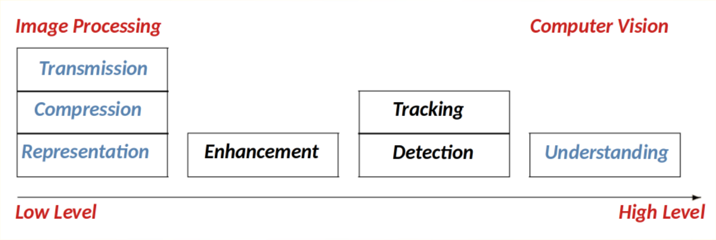


From Image Processing to Computer Vision



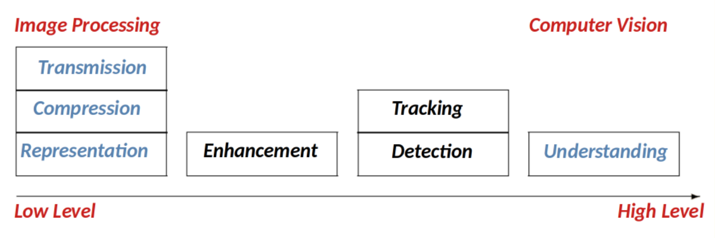
- Why the image processing ?

From Image Processing to Computer Vision



- Why the image processing ?
  - The futur is multimedia : images are everywhere !

From Image Processing to Computer Vision



- Why the image processing ?
  - The futur is multimedia : images are everywhere !
  - Possibilities for application are numerous and varied.

Cours 1 : Domaine d'application

- Teledetection : meteorology, cartography, astronomy,

Cours 1 : Domaine d'application

- Teledetection : meteorology, cartography, astronomy,
- Medical imaging : diagnostic assistance, tomography, auto-tracking, 3D-reconstruction

Cours 1 : Domaine d'application

- Teledetection : meteorology, cartography, astronomy,
- Medical imaging : diagnostic assistance, tomography, auto-tracking, 3D-reconstruction
- Military applications : missile guidance, ground reconnaissance

Cours 1 : Domaine d'application

- Teledetection : meteorology, cartography, astronomy,
- Medical imaging : diagnostic assistance, tomography, auto-tracking, 3D-reconstruction
- Military applications : missile guidance, ground reconnaissance
- Robotics and industry : recognition and assembly of parts, autonomous cars, quality assurance

Cours 1 : Domaine d’application

- Teledetection : meteorology, cartography, astronomy,
- Medical imaging : diagnostic assistance, tomography, auto-tracking, 3D-reconstruction
- Military applications : missile guidance, ground reconnaissance
- Robotics and industry : recognition and assembly of parts, autonomous cars, quality assurance
- Security : face detection and identification, fingerprint recognition, watermarking, data hiding

Cours 1 : Domaine d'application

- Teledetection : meteorology, cartography, astronomy,
- Medical imaging : diagnostic assistance, tomography, auto-tracking, 3D-reconstruction
- Military applications : missile guidance, ground reconnaissance
- Robotics and industry : recognition and assembly of parts, autonomous cars, quality assurance
- Security : face detection and identification, fingerprint recognition, watermarking, data hiding
- Entertainment : HD/FHD (4K/1080P), high-quality images, compression (standards JPEG, JPEG 2000, MPEG4 . . .)

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception
- Restoration : reduce the distortion due to the acquisition process (noise, blur, . . .)

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception
- Restoration : reduce the distortion due to the acquisition process (noise, blur, . . .)
- Compression : storage and transmission

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception
- Restoration : reduce the distortion due to the acquisition process (noise, blur, . . .)
- Compression : storage and transmission
- Segmentation : partitioning a digital image into multiple objects

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception
- Restoration : reduce the distortion due to the acquisition process (noise, blur, . . .)
- Compression : storage and transmission
- Segmentation : partitioning a digital image into multiple objects
- 3D-reconstruction : creation of three-dimensional images from a set of 2D-images

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception
- Restoration : reduce the distortion due to the acquisition process (noise, blur, . . .)
- Compression : storage and transmission
- Segmentation : partitioning a digital image into multiple objects
- 3D-reconstruction : creation of three-dimensional images from a set of 2D-images
- Representation

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception
- Restoration : reduce the distortion due to the acquisition process (noise, blur, . . .)
- Compression : storage and transmission
- Segmentation : partitioning a digital image into multiple objects
- 3D-reconstruction : creation of three-dimensional images from a set of 2D-images
- Representation
  - Low level : textures, intensity, colors, patterns . . .

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception
- Restoration : reduce the distortion due to the acquisition process (noise, blur, . . .)
- Compression : storage and transmission
- Segmentation : partitioning a digital image into multiple objects
- 3D-reconstruction : creation of three-dimensional images from a set of 2D-images
- Representation
    - Low level : textures, intensity, colors, patterns . . .
    - High level : features, machine learning, graphs.

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception
- Restoration : reduce the distortion due to the acquisition process (noise, blur, . . .)
- Compression : storage and transmission
- Segmentation : partitioning a digital image into multiple objects
- 3D-reconstruction : creation of three-dimensional images from a set of 2D-images
- Representation
  - Low level : textures, intensity, colors, patterns . . .
  - High level : features, machine learning, graphs.
- Analyze : to convert in information

Some Examples of image processing

- Enhancement : enhance the quality of the visual perception
- Restoration : reduce the distortion due to the acquisition process (noise, blur, ...)
- Compression : storage and transmission
- Segmentation : partitioning a digital image into multiple objects
- 3D-reconstruction : creation of three-dimensional images from a set of 2D-images
- Representation
  - Low level : textures, intensity, colors, patterns ...
  - High level : features, machine learning, graphs.
- Analyze : to convert in information
- Recognition / Understanding : content-based image recognition

Real image : a mathematical definition

Definition

- An image is a **projection** on a horizontal plane of a 3D scene

Real image : a mathematical definition

Definition

- An image is a **projection** on a horizontal plane of a 3D scene
- An image can be defined as a fonction of 2 variables $f(x, y)$:

Real image : a mathematical definition

Definition

- An image is a **projection** on a horizontal plane of a 3D scene
- An image can be defined as a fonction of 2 variables $f(x,y)$:
  - (x,y) is the projected **position** of a 3D scene point on the projection plane

Real image : a mathematical definition

Definition

- An image is a **projection** on a horizontal plane of a 3D scene
- An image can be defined as a fonction of 2 variables $f(x,y)$:
    - (x,y) is the projected **position** of a 3D scene point on the projection plane
    - $f(x,y)$ is the **luminance intensity** (or **luminance**) at the point (x,y)

Real image : a mathematical definition

Definition

- An image is a **projection** on a horizontal plane of a 3D scene
- An image can be defined as a fonction of 2 variables $f(x, y)$:
  - (x, y) is the projected **position** of a 3D scene point on the projection plane
  - $f(x, y)$ is the **luminance intensity** (or **luminance**) at the point (x, y)
- An image is an analog plane where the luminance intensities are reals

Real image : a definition in computer science

Definition

- A Matrice of numbers. Each number represent a discretised luminance intensity

Real image : a definition in computer science

Definition

- A Matrice of numbers. Each number represent a discretised luminance intensity
- A discretised plane from an analog image after a **digitization** :

Real image : a definition in computer science

Definition

- A Matrice of numbers. Each number represent a discretised luminance intensity
- A discretised plane from an analog image after a **digitization** :
    - **Spatial samping** : obtained from the coordinates of the real image by a discretization

Real image : a definition in computer science

Definition

- A Matrice of numbers. Each number represent a discretised luminance intensity
- A discretised plane from an analog image after a **digitization** :
  - **Spatial samping** : obtained from the coordinates of the real image by a discretization
  - **Quantization of the luminances** : discretization of the real luminance intensities

Definition

- Define the **pixel density** of the image (spatial resolution)

Spatial sampling

Definition

- Define the **pixel density** of the image (spatial resolution)
  - The step size of the image plane partition : number of elements per unit

Spatial sampling

Definition

- Define the **pixel density** of the image (spatial resolution)
  - The step size of the image plane partition : number of elements per unit
  - The tiniest detail in the image

Spatial sampling

Definition

- Define the **pixel density** of the image (spatial resolution)

Spatial sampling

Definition

- Define the **pixel density** of the image (spatial resolution)
  - The step size of the image plane partition : number of elements per unit

Warning !

- Too low resolution cause some aliasing or pixelation effects

Spatial sampling

Definition

- Define the **pixel density** of the image (spatial resolution)
  - The step size of the image plane partition : number of elements per unit
  - The tiniest detail in the image

Warning !

- Too low resolution cause some aliasing or pixelation effects

Definition

- The luminance intensity I is quantized at m bits and et can take $L = 2^m$ values : $I \in \{0, ..., 2^m - 1\}$

Definition

- The luminance intensity I is quantized at m bits and et can take $L = 2^m$ values : $I \in \{0, ..., 2^m - 1\}$
- The tiniest intensity variation

Warning !

- A strong (excessive) quantization causes some false edges

Definition

- The luminance intensity I is quantized at m bits and et can take $L = 2^m$ values : $I \in \{0, ..., 2^m - 1\}$
- The tiniest intensity variation

Warning !

- A strong (excessive) quantization causes some false edges

Example

- $m = 1$: 2 possible values (**binary** images)
- $m = 8$: 256 possible values possibles (**grayscale** images)
- $m = 16$: 65535 valeurs possibles (**colors** images)

Feature of an image

Definition

- Surface divided on small rectangles with a fixed size named **pixels** (picture elements), definied by :

Feature of an image

Definition

- Surface divided on small rectangles with a fixed size named **pixels** (picture elements), definied by :
  - The image width N and the image height M defining the $N \times M$ pixels of the image (obtained after the spatial sampling)

Example

Feature of an image

Definition

- Surface divided on small rectangles with a fixed size named **pixels** (picture elements), definied by :
  - The image width N and the image height M defining the $N \times M$ pixels of the image (obtained after the spatial sampling)
  - The level of the luminance intensities (contrast) (after the quantization)

Example

Feature of an image

Definition

- Surface divided on small rectangles with a fixed size named **pixels** (picture elements), definied by :
  - The image width N and the image height M defining the $N \times M$ pixels of the image (obtained after the spatial sampling)
  - The level of the luminance intensities (contrast) (after the quantization)

Example

- Grayscal image : (8 bits) with the size 128×128 :

$$128 \times 128 \times 8 = 16 \text{ Koctets}$$

Feature of an image

Definition

- Surface divided on small rectangles with a fixed size named **pixels** (picture elements), definied by :
  - The image width N and the image height M defining the $N \times M$ pixels of the image (obtained after the spatial sampling)
  - The level of the luminance intensities (contrast) (after the quantization)

Example

- Grayscal image : (8 bits) with the size 128×128 :

$$128 \times 128 \times 8 = 16 \text{ Koctets}$$

- Colors Image : (32 bits) withe the size 256×256 :

$$256 \times 256 \times 32 = 256 \text{ Koctets}$$

Digital Image Standards

Standard

- Raw data,

Digital Image Standards

Standard

- Raw data,
- free standards : JPG, PNG, GIF, BMP, bitmap TIFF, ppm, eps . . .

Example

Digital Image Standards

Standard

- Raw data,
- free standards : JPG, PNG, GIF, BMP, bitmap TIFF, ppm, eps . . .
- medical standard : DICOM, ACR-NEMA, . . .

Example

Digital Image Standards

Standard

- Raw data,
- free standards : JPG, PNG, GIF, BMP, bitmap TIFF, ppm, eps . . .
- medical standard : DICOM, ACR-NEMA, . . .
- private standards : JPEG2000, Philips, Siemens . . .

Example

Digital Image Standards

Standard

- Raw data,
- free standards : JPG, PNG, GIF, BMP, bitmap TIFF, ppm, eps . . .
- medical standard : DICOM, ACR-NEMA, . . .
- private standards : JPEG2000, Philips, Siemens . . .

Example

- BMP (Bitmap) : matrix of bits coded in colors (up to 24 bits/pixel),

Digital Image Standards

Standard

- Raw data,
- free standards : JPG, PNG, GIF, BMP, bitmap TIFF, ppm, eps . . .
- medical standard : DICOM, ACR-NEMA, . . .
- private standards : JPEG2000, Philips, Siemens . . .

Example

- BMP (Bitmap) : matrix of bits coded in colors (up to 24 bits/pixel),
- PNG : lossless data compression,

Digital Image Standards

Standard

- Raw data,
- free standards : JPG, PNG, GIF, BMP, bitmap TIFF, ppm, eps . . .
- medical standard : DICOM, ACR-NEMA, . . .
- private standards : JPEG2000, Philips, Siemens . . .

Example

- BMP (Bitmap) : matrix of bits coded in colors (up to 24 bits/pixel),
- PNG : lossless data compression,
- JPG : method of lossy compression for digital images (example for digital photography),

Digital Image Standards

Standard

- Raw data,
- free standards : JPG, PNG, GIF, BMP, bitmap TIFF, ppm, eps . . .
- medical standard : DICOM, ACR-NEMA, . . .
- private standards : JPEG2000, Philips, Siemens . . .

Example

- BMP (Bitmap) : matrix of bits coded in colors (up to 24 bits/pixel),
- PNG : lossless data compression,
- JPG : method of lossy compression for digital images (example for digital photography),
- JPEG2000 : an optimal method of compression.

Luminance intensity of images

Definition

- Grayscale image : value of the luminance intensity $I(x, y)$ at pixel (x, y)

Definition

- Grayscale image : value of the luminance intensity $I(x, y)$ at pixel (x, y)

Example

- Binary image : 2 possible values for the luminance intensity I, 0 or 1 for each pixel :

Luminance intensity of images

Definition

- Grayscale image : value of the luminance intensity $I(x, y)$ at pixel (x, y)

Example

- Binary image : 2 possible values for the luminance intensity I, 0 or 1 for each pixel :
- Grayscale image :

Luminance intensity of images

Definition

- Grayscale image : value of the luminance intensity $I(x, y)$ at pixel (x, y)

Example

- Binary image : 2 possible values for the luminance intensity I, 0 or 1 for each pixel :
- Grayscale image :
  - Quantization of the luminance intensities on $[0, 255]$

Luminance intensity of images

Definition

- Grayscale image : value of the luminance intensity $I(x, y)$ at pixel (x, y)

Example

- Binary image : 2 possible values for the luminance intensity I, 0 or 1 for each pixel :
- Grayscale image :
  - Quantization of the luminance intensities on $[0, 255]$
  - Codage on bits (1 octet) : $2^0 - 1 \leq k \leq 2^8 - 1$

Luminance intensity of images

Definition

- Grayscale image : value of the luminance intensity $I(x, y)$ at pixel (x, y)

Example

- Binary image : 2 possible values for the luminance intensity I, 0 or 1 for each pixel :
- Grayscale image :
  - Quantization of the luminance intensities on $[0, 255]$
  - Codage on bits (1 octet) : $2^0 - 1 \leq k \leq 2^8 - 1$
  - Common convention : black $= 0$, white $= 255$

Quality

- Contrasts : the dynamique range of the luminance intensities.

Quality

- Contrasts : the dynamique range of the luminance intensities.
- Noise : random variation of brightness or color information in images, and is usually an aspect of electronic noise. Its distribution is generally unknown.

Quality

- Contrasts : the dynamique range of the luminance intensities.
- Noise : random variation of brightness or color information in images, and is usually an aspect of electronic noise. Its distribution is generally unknown.
- Geometrical distortions : defects due to the axis difference between the acquisition sensor and the center of the observed scene.

Content

- Texture : a quantitative measure of the arrangement of intensities in a region of the image.

Content

- Texture : a quantitative measure of the arrangement of intensities in a region of the image.
- Contour : boundary of 2 groups of pixels where the difference of grayscale or colors level is significant (“enough large”).

Content

- Texture : a quantitative measure of the arrangement of intensities in a region of the image.
- Contour : boundary of 2 groups of pixels where the difference of grayscale or colors level is significant (“enough large”).
- Region : group of pixels with similar features (luminance intensity, colors, texture . . .).

Content

- Texture : a quantitative measure of the arrangement of intensities in a region of the image.
- Contour : boundary of 2 groups of pixels where the difference of grayscale or colors level is significant (“enough large”).
- Region : group of pixels with similar features (luminance intensity, colors, texture . . .).
- Object : region (group of region) bounded by a contour.

Read and display an Image

Code

```
% Load an Image with Octave :
close all, clear, clc
X = imread("lena.jpg");
% The image is loaded in the variable X
% with other images? :
% X = imread('name of your image.entension');

%Visualization : displaying Images in grayscale
imshow(X);
colormap gray;

% Displaying a second figure :
figure(2);
colormap gray;
XX=imread("cameraman.png");
imshow(XX);
```

Function noise - Adding a noise to an image

```
% gaussian noise (standard deviation s) of an image I

function out = gaussian_noise(I,s)

[m,n]=size(I);
J=zeros(m,n);
% create the gaussian noise
J=s*randn(m,n);

% adding the gaussian noise to the image I

out = I+J;
```

An image and a noisy image

Display the image and the noisy image

```
% Adding an artificial noise to an image :
close all , clear , clc
X = double(imread("lena.jpg"));

% the image is loaded in the variable X
% here we prefer the double precision

%Visualization :
imagesc(X);
colormap gray;
% Displaying the image in grayscale

% adding the noise with the function noise
% Try with s = 5, 15, 25 and 30
XX=noise (X,s);

% Displaying the image in grayscale
figure(2);
colormap gray;
imagesc(XX);
```

An image and its boundary

Some frameworks to restore a noisy image

- The image I can be assimilated to an open bounded domaine Ω of $\mathbb{R}^2$.

An image and its boundary

Some frameworks to restore a noisy image

- The image I can be assimilated to an open bounded domaine Ω of $\mathbb{R}^2$.
- We will use the variations of the image to restore it $\Longrightarrow$ We need some assumption on the boudaries of Ω, namely :

$$\frac{\partial I}{\partial \vec{n}} = \langle \nabla I, \vec{n} \rangle = 0$$

  where $\vec{n}$ is the normal vector to the boundaries of I $\partial\Omega$ and $\langle ., . \rangle$ is the (euclidian) dot product (inner product),

An image and its boundary

Some frameworks to restore a noisy image

- The image I can be assimilated to an open bounded domaine Ω of $\mathbb{R}^2$.
- We will use the variations of the image to restore it $\Longrightarrow$ We need some assumption on the boudaries of Ω, namely :

$$\frac{\partial I}{\partial \vec{n}} = \langle \nabla I, \vec{n} \rangle = 0$$

  where $\vec{n}$ is the normal vector to the boundaries of I $\partial\Omega$ and $\langle ., . \rangle$ is the (euclidian) dot product (inner product),
- in this course, this boundary conditions will be assumed and in practice, we obtain it by replace the real boudaries of the image I by “reflection symmetry”

An image and its boundary

Some frameworks to restore a noisy image

- The image I can be assimilated to an open bounded domaine Ω of $\mathbb{R}^2$.
- We will use the variations of the image to restore it $\Longrightarrow$ We need some assumption on the boudaries of Ω, namely :

$$\frac{\partial I}{\partial \vec{n}} = \langle \nabla I, \vec{n} \rangle = 0$$

  where $\vec{n}$ is the normal vector to the boundaries of I $\partial\Omega$ and $\langle .,. \rangle$ is the (euclidian) dot product (inner product),
- in this course, this boundary conditions will be assumed and in practice, we obtain it by replace the real boudaries of the image I by "reflection symmetry"
- These conditions are the Neumann boundary conditions.

Neumann Conditions

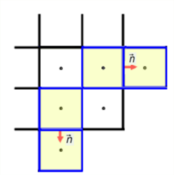
Some details and example

- Any image I could not verify the neumann boundary condition.

Neumann Conditions

Some details and example

- Any image I could not verify the neumann boundary condition.
- That's why we pose $\widetilde{I}$ an extension image of I constructs such that $\widetilde{I}$ verifies the Neumann boudary conditions, namely :
  - The size of $\widetilde{I}$ is $(N+p)\times(M+r)$, where p is the extension of I among the $x-$axis and r the extension of I among the $x-$axis,
  - p and r depend on the problem to be solved,
  - else the values of $\widetilde{I}$ are obtained by "reflection symmetry" (see below for a simple example) :

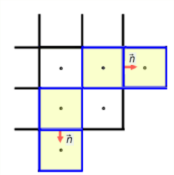


At $x=N$ and $y=2$: $\dfrac{\partial \widetilde{I}}{\partial \vec{n}}=\dfrac{\partial \widetilde{I}}{\partial x}=\dfrac{\widetilde{I}_{N+1,2}-\widetilde{I}_{N,2}}{\Delta x}=0$

$$\Longrightarrow \boxed{\widetilde{I}_{N+1,2}=\widetilde{I}_{N,2}=I_{N,2}}.$$

Some details and example

- Any image I could not verify the neumann boundary condition.
- That's why we pose $\widetilde{I}$ an extension image of I constructs such that $\widetilde{I}$ verifies the Neumann boudary conditions, namely :
  - The size of $\widetilde{I}$ is $(N+p)\times(M+r)$, where p is the extension of I among the $x-$axis and r the extension of I among the $x-$axis,
  - p and r depend on the problem to be solved,
  - on $[1,N]\times[1,M]$, $\widetilde{I}=I$,
  - else the values of $\widetilde{I}$ are obtained by "reflection symmetry" (see below for a simple example) :



At $x=N$ and $y=2$: $\dfrac{\partial \widetilde{I}}{\partial \vec{n}} = \dfrac{\partial \widetilde{I}}{\partial x} = \dfrac{\widetilde{I}_{N+1,2}-\widetilde{I}_{N,2}}{\Delta x} = 0$

$$\Longrightarrow \boxed{\widetilde{I}_{N+1,2}=\widetilde{I}_{N,2}=I_{N,2}}.$$

Function boundary - extension of an image - Neumann conditions

```
function B=boundary(A,d)
%Extension of an A by reflection symmetry (for d pixels)
% Neumann Conditions
%syntaxe : bord(A,d)
[m,n]=size(A);
% Create the matrice B with the right size
M=m+2*d;
N=n+2*d;
B=zeros(M,N);
B(d+1:M-d,d+1:N-d)=A;
% Extension by a reflection symmetry
for i=1:m
    for j=1:d B(i+d,j)=A(i,d-j+1);end;
    for j=N-d+1:N B(i+d,j)=A(i,n+N-j-d);end;
end;
for j=1:N
    for i=1:d B(i,j)=B(2*d-i+1,j);end;
    for i=M-d+1:M B(i,j)=B(2*M-i-2*d,j);end;
end;
```

Display the image and the extended image with the Neumann conditions

```
close all ,clear ,clc
X = double(imread("lena.jpg"));

% the image is loaded in the variable X
% here we prefer the double precision

%Visualization :
imagesc(X);
colormap gray;
% Displaying the image in grayscale

% Extend the image with s pixel
% Try with s = 5, 15, 25 and 30

XX=boundary(X,s);
% Displaying the image in grayscale
figure(2);
colormap gray;
imagesc(XX);
```

Reminder

First order's derivatives

$f : D \subset \mathbb{R}^{\mathbf{2}} \to \mathbb{R}$ a $\mathscr{C}^1(D)$ function then for all $(x,y) \in D$, we have :

$$\frac{\partial f(x,y)}{\partial x} = \lim_{h \to 0} \frac{f(x+h,y) - f(x,y)}{h} \text{ and } \frac{\partial f(x,y)}{\partial y} = \lim_{h \to 0} \frac{f(x,y+h) - f(x,y)}{h}.$$

In other words :

$$\frac{\partial f(x,y)}{\partial x} \simeq \frac{f(x+h,y) - f(x,y)}{h} \text{ and } \frac{\partial f(x,y)}{\partial y} \simeq \frac{f(x,y+h) - f(x,y)}{h}.$$

For an image I, $h = 1$ (we can not see under 1 pixel) :

$$\frac{\partial I(x,y)}{\partial x} \simeq I(x+1,y) - I(x,y) \text{ and } \frac{\partial I(x,y)}{\partial y} \simeq I(x,y+1) - I(x,y).$$

Reminder

The forward difference

Now in its discrete form, if $(x, y) = (i, j)$ with $I(i, j) = I_{i,j}$, we can rewrite the previous result as the following :

$$\frac{\partial I_{i,j}}{\partial x} \simeq I_{i+1,j} - I_{i,j} \text{ and } \frac{\partial I_{i,j}}{\partial y} \simeq I_{i,j+1} - I_{i,j}.$$

Reminder

The forward difference

Now in its discrete form, if $(x, y) = (i, j)$ with $I(i, j) = I_{i,j}$, we can rewrite the previous result as the following :

$$\frac{\partial I_{i,j}}{\partial x} \simeq I_{i+1,j} - I_{i,j} \text{ and } \frac{\partial I_{i,j}}{\partial y} \simeq I_{i,j+1} - I_{i,j}.$$

The backward difference

$$\frac{\partial I_{i,j}}{\partial x} \simeq I_{i,j} - I_{i-1,j} \text{ and } \frac{\partial I_{i,j}}{\partial y} \simeq I_{i,j} - I_{i,j-1}.$$

Reminder

The forward difference

Now in its discrete form, if $(x, y) = (i, j)$ with $I(i, j) = I_{i,j}$, we can rewrite the previous result as the following :

$$\frac{\partial I_{i,j}}{\partial x} \simeq I_{i+1,j} - I_{i,j} \text{ and } \frac{\partial I_{i,j}}{\partial y} \simeq I_{i,j+1} - I_{i,j}.$$

The backward difference

$$\frac{\partial I_{i,j}}{\partial x} \simeq I_{i,j} - I_{i-1,j} \text{ and } \frac{\partial I_{i,j}}{\partial y} \simeq I_{i,j} - I_{i,j-1}.$$

The central difference

$$\frac{\partial I_{i,j}}{\partial x} \simeq \frac{I_{i+1,j} - I_{i-1,j}}{2} \text{ and } \frac{\partial I_{i,j}}{\partial y} \simeq \frac{I_{i,j+1} - I_{i,j-1}}{2}.$$

First order's image Gradient computation & Octave

The central difference

```
function [I_x,I_y] = compute_derivatives(I)

    [ny,nx,s]=size(I);
    % estimate first derivatives - center derivative scheme
    I_x = (I(:,[2:nx nx],:)-I(:,[1 1:nx-1],:))/2;
    I_y = (I([2:ny ny],:,:)-I([1 1:ny-1],:,:))/2;
```

First order's image Gradient computation & Octave

Display the image and its first derivatives

```
% Adding an artificial noise to an image :
close all ,clear ,clc
X = double(imread("lena.jpg"));
% the image is loaded in the variable X in double precision
%Visualization in grayscale
imagesc(X);
colormap gray;
% Compute the derivatives of the image I
[X_x,X_y]=compute_derivatives(X);
% Displaying the x-derivative of the image in grayscale
figure(2);
colormap gray;
imagesc(X_x);
% Displaying the y-derivative of the image in grayscale
figure(3);
colormap gray;
imagesc(X_y);
```

Reminder

Second order's derivatives

$f : D \subset \mathbb{R}^{\mathbf{2}} \to \mathbb{R}$ a $\mathscr{C}^2(D)$ function then for all $(x,y) \in D$, we have :

$$\frac{\partial^2 f(x,y)}{\partial x^2} = \lim_{h\to 0} \frac{f(x+h,y) - 2f(x,y) + f(x-h,y)}{h^2},$$
$$\frac{\partial^2 f(x,y)}{\partial y^2} = \lim_{h\to 0} \frac{f(x,y+h) - 2f(x,y) + f(x,y-h)}{h^2}.$$

And the cross partial derivatives :

$$\frac{\partial^2 f(x,y)}{\partial x \partial y} = \lim_{h\to 0} \frac{f(x+h,y+h) - f(x+h,y-h) - f(x-h,y+h) + f(x-h,y-h)}{4h^2}.$$

Reminder

Second order's derivatives

In other words :

$$\frac{\partial^2 f(x,y)}{\partial x^2} \simeq \frac{f(x+h,y) - 2f(x,y) + f(x-h,y)}{h^2},$$

$$\frac{\partial^2 f(x,y)}{\partial y^2} \simeq \frac{f(x,y+h) - 2f(x,y) f(x,y-h)}{h^2}.$$

And the cross partial derivatives :

$$\frac{\partial^2 f(x,y)}{\partial x \partial y} \simeq \frac{f(x+h,y+h) - f(x+h,y-h) - f(x-h,y+h) + f(x-h,y-h)}{4h^2}.$$

Reminder

Second order's derivatives

For an image I, $h = 1$ (we can not see under 1 pixel) :

$$\frac{\partial^2 I(x,y)}{\partial x^2} \simeq I(x+1,y) - 2\,I(x,y) + I(x-1,y),$$

$$\frac{\partial^2 I(x,y)}{\partial y^2} \simeq I(x,y+1) - 2\,I(x,y) + I(x,y-1).$$

And the cross partial derivatives :

$$\frac{\partial^2 f(x,y)}{\partial x \partial y} \simeq \frac{f(x+1,y+1) - f(x+1,y-1) - f(x-1,y+1) + f(x-1,y-1)}{4}.$$

The second order derivatives of an image

Now in its discrete form, if $(x, y) = (i, j)$ with $I(i, j) = I_{i,j}$, we can rewrite the previous result as the following :

$$\frac{\partial^2 I_{i,j}}{\partial x^2} \simeq I_{i+1,j} - 2\, I_{i,j} + I_{i,j-1},$$

$$\frac{\partial^2 I_{i,j}}{\partial y^2} \simeq I_{i,j+1} - 2\, I_{i,j} + I_{i,j-1}.$$

And the cross partial derivative :

$$\frac{\partial^2 I_{i,j}}{\partial x\, \partial y} \simeq \frac{I_{i+1,j+1} - I_{i+1,j-1} - I_{i-1,j+1} + I_{i-1,j-1}}{4}.$$

second order's image Gradient computation & Octave

Second order finite difference

```
function [I_xx,I_yy,I_xy] = compute_2derivatives(I)

    [ny,nx,s]=size(I);
    % second order's derivatives estimates
    I_xx = I(:,[2:nx nx],:)+I(:,[1 1:nx-1],:)-2*I;
    I_yy = I([2:ny ny],:,:)+I([1 1:ny-1],:,:)-2*I;
    Dp = I([2:ny ny],[2:nx nx],:)+I([1 1:ny-1],[1 1:nx-1],:);
    Dm = I([1 1:ny-1],[2:nx nx],:)+I([2:ny ny],[1 1:nx-1],:);
    I_xy = (Dp-Dm)/4;
```

Display the image and its second order's derivatives

```
% Adding an artificial noise to an image :
close all ,clear ,clc
X = double(imread("lena.jpg"));
% the image is loaded in the variable X
% here we prefer the double precision

%Visualization in grayscale
imagesc(X);
colormap gray;
%
% Compute the derivatives of the image I
[X_xx,X_yy,X_xy]=compute_2derivatives(X);

% Displaying the x-derivative of the image in grayscale
figure(2);colormap gray;imagesc(X_xx);
% Displaying the y-derivative of the image in grayscale
figure(3);colormap gray;imagesc(X_yy);
% Displaying the cross partial derivative of the image in grayscale
figure(4);colormap gray;imagesc(X_xy);
```

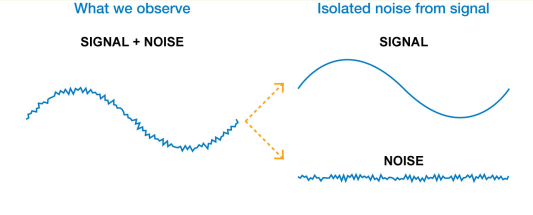
Image denoising - a model

A model

If I denotes the "ideal" image and n, the gaussian noise (assumption), it is usual to suppose that this noise is added by the sensor to the "ideal" image :

$$I^{noise} = I + n,$$

To fix our idea, let us consider an simple 1D-example with a signal :



Our objective is to extract the signal from the noise (or isolated noise from the signal). This noise "increase the variations" of the signal, a first path forward : if we could rationally "control or reduce" these variations . . . Toward regularization methods ?

Image denoising - heat equation

The heat equation

The first Partial Differential Equation (PDE) used in image processing was the heat equation. The main idea is compute the solution (the denoised image) of the heat equation where the initial condition is the observed image I^{noise} (the noisy image !) :

$$\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} = \Delta u(x,y,t) \text{ on } \Omega\times]0,T[\\ u(x,y,0) = I^{noise}(x,y) \text{ on } \Omega \\ \dfrac{\partial u(x,y,t)}{\partial \vec{n}} = 0 \text{ on } \partial\Omega\times]0,T[. \end{cases}$$

where u is function defined on $\Omega \times [0,T]$, $\vec{n}$ the normal vector, Ω is the domain of the image I^{noise}, and T is the maximum time.

Image denoising - Heat equation - The discrete form of the laplacian

Reminder : the operator Laplacian Δu

$u : \Omega \subset \mathbb{R}^{\mathbf{2}} \to \mathbb{R}$ is a $\mathscr{C}^2(\Omega)$ function :

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

For an image I, if we use the discrete form of the second order derivatives of u, we have :

$$\Delta I_{i,j} = I_{i+1,j} + I_{i-1,j} + I_{i,j+1} + I_{i,j-1} - 4\, I_{i,j}$$

Discrete scheme

We recall the heat equation : $\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} = \Delta u(x,y,t) \text{ on } \Omega\times]0,T[\\ u(x,y,0) = I^{noise}(x,y) \text{ on } \Omega \\ \dfrac{\partial u(x,y,t)}{\partial \vec{n}} = 0 \text{ on } \partial\Omega\times]0,T[. \end{cases}$ To solve this PDE's, we need a discretization of $\dfrac{\partial u(x,y,t)}{\partial t}$, we begin with the definition :

$$\frac{\partial u(x,y,t)}{\partial t} = \lim_{\delta t \to 0} \frac{u(x,y,t+\delta t) - u(x,y,t)}{\delta t}$$

where $t \in]0,T[$, in other words :

$$\frac{\partial u(x,y,t)}{\partial t} \simeq \frac{u(x,y,t+\delta t) - u(x,y,t)}{\delta t}$$

Discrete scheme

For u, if we use the discrete form of the second order derivatives of u and we use the following time discretization :

$$t_0 = 0, t_1 = \delta t, t_2 = 2\delta t, \ldots, t_n = n\delta t = T,$$

with $u(x,y,t_k) = u(x,y,k\delta t) = u^k(x,y)$. Moreover if $(x,y) \in \Omega$, Ω the domain of the image I, then we have $(x,y) = (i,j)$ and $u^k(x,y) = u^k(i,j) = u^k_{i,j}$.
Now, it is obvious to obtain :

$$\frac{\partial u^k_{i,j}}{\partial t} \simeq \frac{u^{k+1}_{i,j} - u^k_{i,j}}{\delta t}$$

By using the discrete form of the laplacian :

$$\Delta u^k_{i,j} = u^k_{i+1,j} + u^k_{i-1,j} + u^k_{i,j+1} + u^k_{i,j-1} - 4\,u^k_{i,j}$$

Image denoising - Heat equation - A discrete scheme

Discrete scheme

We can rewrite the heat equation in its discrete form :

$$\frac{u_{i,j}^{k+1} - u_{i,j}^{k}}{\delta t} = \Delta u_{i,j}^{k},$$

or in this more convenient form :

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \delta t \left(u_{i+1,j}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k} + u_{i,j-1}^{k} - 4\, u_{i,j}^{k} \right).$$

with $u_{i,j}^{0} = I_{i,j}^{noise}$, $1 \leq i \leq N$, $1 \leq j \leq M$, $0 \leq k \leq n$ and the Neumann conditions at the boundary $\partial\Omega$ of Ω (the domaine of the image I^{noise}).

Exercice

1. Write the octave program of the heat equation using a noisy version of the image “lena” (lena+noise) then with the image “test”.
2. What can you conclude about this restoration ?

Image denoising - The divergence operator div

Reminder : the divergence operator div

$\mathbf{u} : \Omega \subset \mathbb{R}^{2} \to \mathbb{R}^{2}$ is a $\mathscr{C}^1(\Omega)$ function, where $\mathbf{u} = \left(u^1, u^2\right)$:

$$\mathrm{div}\left(\mathbf{u}(x,y)\right) = \frac{\partial u^1(x,y)}{\partial x} + \frac{\partial u^2(x,y)}{\partial y}$$

In mathematics, it is well-known that, we have a strong relation between the divergence div and gradient ∇ operators, namely :

$$\langle \varphi, \mathrm{div}\,\mathbf{u} \rangle = -\langle \nabla\varphi, \mathbf{u} \rangle$$

$\forall \varphi$, a $\mathscr{C}^1(\Omega)$ function with compact support and $\langle ., . \rangle$ is the dot product and its discrete version for 2 images I and J :

$$\langle I, J \rangle_d = \sum_{i,j} I_{i,j} J_{i,j}.$$

This relation is very important (but its explanation is beyond the scope of this course) that is why we would like to preserve it in the discrete case !

Image denoising - The discrete divergence operator div in octave

Reminder : the discrete divergence operator div & octave

```
function M = div(px,py)
% compute the divergence of a vector (p1,p2)
% px and py have the same size
% (satistiying div = -(grad)*)
% Syntaxe : div(px,py)
px_x = (px(:,[2:nx nx],:) - px(:,[1 1:nx-1],:))/2;
py_y = (py([2:ny ny],:,:) - py([1 1:ny-1],:,:))/2;
   M = px_x+py_y;
```

Remark : $\mathrm{div}\,(\nabla u) = \Delta u$, compare this result with the previous one (on the laplacian operator).

The Perona-Malik equation

To enhance the result of the heat equation, Perona and Malik introduced their model by modifiying the heat equation. The main idea is to introduce a edge detection process. Then with this new model, we compute the solution (the denoised image) where the initial condition is the observed image I^{noise} (the noisy image !) :

$$\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} = \mathrm{div}\,\left(c(|\nabla u|)\nabla u(x,y,t)\right) \text{ on } \Omega\times]0,T[\\ u(x,y,0) = I^{noise}(x,y) \text{ on } \Omega \\ \dfrac{\partial u(x,y,t)}{\partial \vec{n}} = 0 \text{ on } \partial\Omega\times]0,T[. \end{cases}$$

where u is function defined on $\Omega \times [0,T]$, $\vec{n}$ the normal vector, Ω is the domain of the image I^{noise}, T is the maximum time and the function c an edge detector.

Image denoising - Perona-Malik equation - A discrete scheme

Discrete scheme

Exercice

1. As the heat equation, write the octave program of the Perona-Malik equation using a noisy version of the image “lena” (lena+noise) then with the image “test” and with the function $c(t) = \dfrac{1}{\sqrt{1+\left(\frac{t}{\alpha}\right)^2}}$ then with $c(t) = \dfrac{1}{1+\left(\frac{t}{\alpha}\right)^2}$.
2. What can you conclude about this restoration ?

Image denoising - An enhancement of the Perona-Malik equation

The alternative equation

To enhance the result of the Perona-Malik equation, we can put a smoothing on the gradient of the image in the function c. This smoothing will regularize the edge detector avoiding to consider the noise as an edge. Then with this new model, we compute the solution (the denoised image) where the initial condition is the observed image I^{noise} (the noisy image !) :

$$\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} = \mathrm{div}\left(c(|G_\sigma * \nabla u|)\nabla u(x,y,t)\right) \text{ on } \Omega\times]0,T[\\ u(x,y,0) = I^{noise}(x,y) \text{ on } \Omega \\ \dfrac{\partial u(x,y,t)}{\partial \vec{n}} = 0 \text{ on } \partial\Omega\times]0,T[. \end{cases}$$

where u is function defined on $\Omega \times [0,T]$, $\vec{n}$ the normal vector, Ω is the domain of the image I^{noise}, T is the maximum time, c an edge detector and G_σ a gaussian filter.

Exercice (discrete scheme)

1. As for the previous equations (heat and Perona-Malik), write the octave program using a noisy version of the image "lena" (lena+noise) then with the image "test" and with the function $c(t) = \dfrac{1}{\sqrt{1+\left(\frac{t}{\alpha}\right)^2}}$ then with

$$c(t) = \frac{1}{1+\left(\frac{t}{\alpha}\right)^2}.$$

```
% Don't forget to load the octave package image processing
% for fspecial and imfilter!
pkg load image
% try for sigma =1, 3, 10, 15
sigma = 3
k=fspecial("gaussian",sigma);
% Convolution k*I where I is an image
Iconv = imfilter(I,k,'conv','symmetric')
```

2. What can you conclude about this restoration?

Image restoration-The mathematical model

- $\Omega \subset \mathbb{R}^2$: the domain of the image
- $u : \Omega \longrightarrow \mathbb{R}$: the original image.
- $u_0 : \Omega \longrightarrow \mathbb{R}$: the observed image (a degradation of u).

$$u_0 = u + n,$$

- n : is the noise (a random variable).

An approximation of u is given by solving the minimization problem

$$\min_u \frac{1}{2}\int_\Omega |u-u_0|^2 dx + \lambda \int_\Omega \varphi(|\nabla u|)dx.$$

Formally, u, the minimum, satisfies the equation

$$u - \lambda \mathrm{div}\left(\frac{\varphi'(|\nabla u|)}{|\nabla u|}\nabla u\right) = u_0.$$

Image restoration-Regularization function

Thus, the function φ could be chosen such that

- Encourage *smoothing* (isotropic diffusion) in regions of *weak variations* of u ($\nabla u \approx 0$).
- Direct the diffusion along the edges and not across them (to preserve edges).

$$\lim_{t \longrightarrow +\infty} \frac{t\varphi''(t)}{\varphi'(t)} = 0.$$

Example 1 : $\varphi(t) = \sqrt{1+t^2}$
Example 2 : $\varphi(t) = t$ (Total variation model).
(the usual function $\varphi(t) = t^2$ does not work : edges are not preserved).

Image restoration-A descent algorithm with a constant stepsize

For each u Set $w(u) = (u - u_0) - \lambda \mathrm{div}\left(\frac{\varphi'(|\nabla u|)}{|\nabla u|}\nabla u\right)$.

- *Initial data :* $n = 0$, $u^{(0)} = u_0$, α.
- *Step 1 :* if $w(u^{(n)}) = 0$, then stop.
- *Step 2 :* $d^{(n)} = w(u^{(n)})$.
- *Step 4 :* $u^{(n+1)} = u^{(n)} - \alpha d^{(n)}$.
  $n \leftarrow n + 1$. Go to Step 1.

Image restoration-A descent algorithm with a constant stepsize

Exercice (discrete scheme)

1. Write the octave program using a noisy version of the image “lena” (lena+noise) then with the image “test” and with $\varphi(t) = t^2$ then with the function $\varphi(t) = \sqrt{1+t^2}$.
2. What can you conclude about this restoration ?

Thank you for your attention!