Image mining:
Introduction – Image models

Masters IP Paris & Paris-Saclay
Data AI - AI

Antoine MANZANERA
ENSTA Paris / U2IS

ENSTA
IP PARIS

Image Mining Course: Objectives

Images and Videos represent a *major source of information* today.

There is an increasing demand to find *automated methods* to *organise* huge collections of image data, and to *interpret* images and videos by computer.

The *Image Mining* course deals with the problem of increasing the semantics of visual data by:

(1) *reducing* its information to *relevant* data

(2) *adding* to it specific information related to a *model* and/or to a previous *knowledge*, in order to facilitate its retrieval and interpretation by a machine.

The course does not assume previous knowledge in Image Processing, but basics in signal processing, information theory and pattern recognition are useful.

Image Mining Course: Content

***- Introduction - Image basics and processing models*:**
Image sampling and quantization. Linear models and convolutions. Frequency models and Fourier transforms. Differential models, Scale-space and EDPs. Discrete models, set based models and mathematical morphology. Statistical and probabilistic models.

***- Image clustering and classification*:**
Unsupervised clustering for images: PCA, K-means... Supervised classification in images: Bayesian methods, SVM...

- Feature extraction:
Multiscale derivatives, gradient, Hessian and curvature. Contours extraction. Interest point extractors. Basics of segmentation.

- Image representation and description:
Local and regional descriptors. Differential invariants. Histograms of orientation. Local Binary patterns. Visual bag-of-words representations. Hough based representations.

- Visual learning for image recognition and mining:
Applications of deep convolutional networks for visual recognition: image categorisation, object recognition, semantic segmentation, image captioning,...

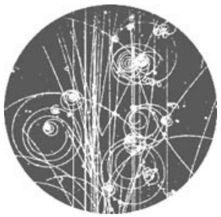
- Video, motion estimation and object tracking:
Optical flow estimation. Basic of object tracking methods. Video structuring and indexing.

- Application / Case study:
Satellite image mining

Introduction and Image Models

A (very) brief History of Image Processing

1950



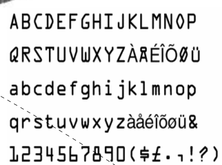
Bubble chamber images

EMPIRICISM

Restoration

Improvement

ACTIVE VISION



Typographic characters

RECONSTRUCTIONISM

PDEs & SCALE SPACE

Satellite and Aerial imagery

MATHEMATICAL MORPHOLOGY

Classification

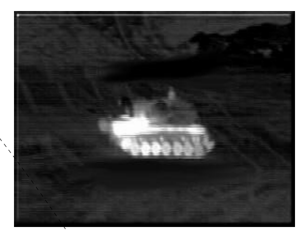
Medical imaging

Videosurveillance and Defence

LARGE SCALE MACHINE LEARNING

Reconstruction

Industrial control



Detection

Tracking

Mobile Robotics

Localisation

Multimedia

Compression

Recognition

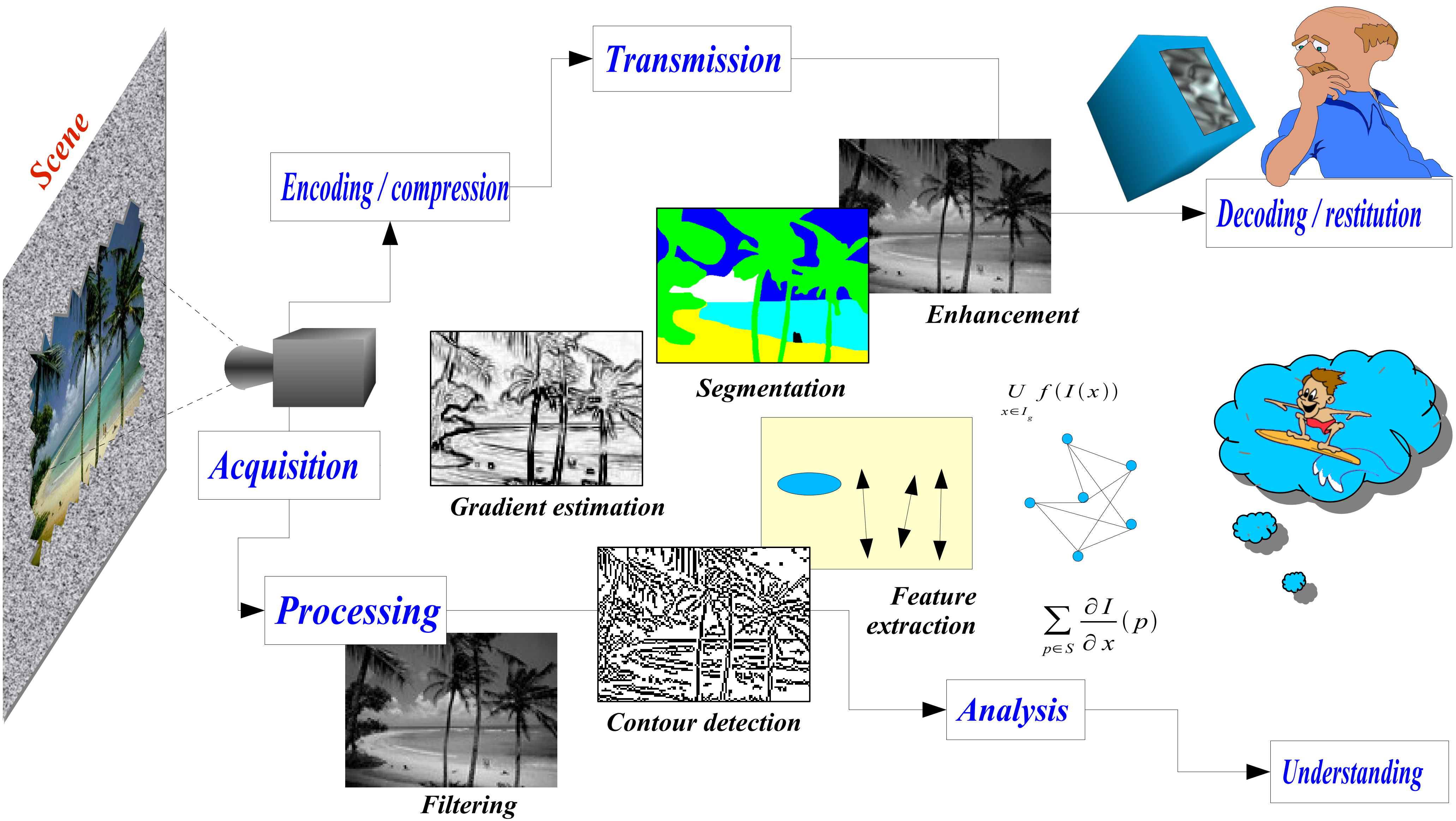
Mining

Augmented Reality

Autonomous Driving

2020

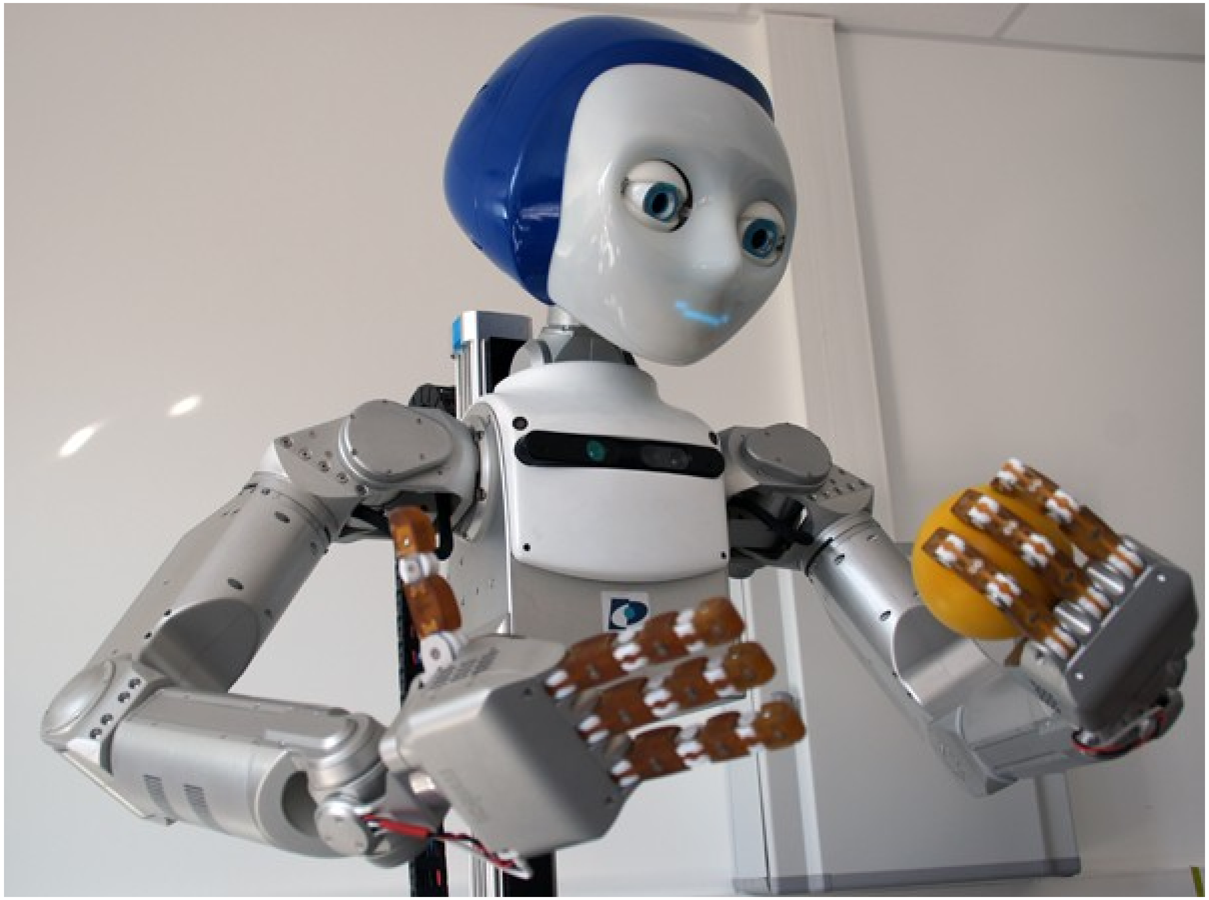
Image Processing Systems?



Vision for AI...

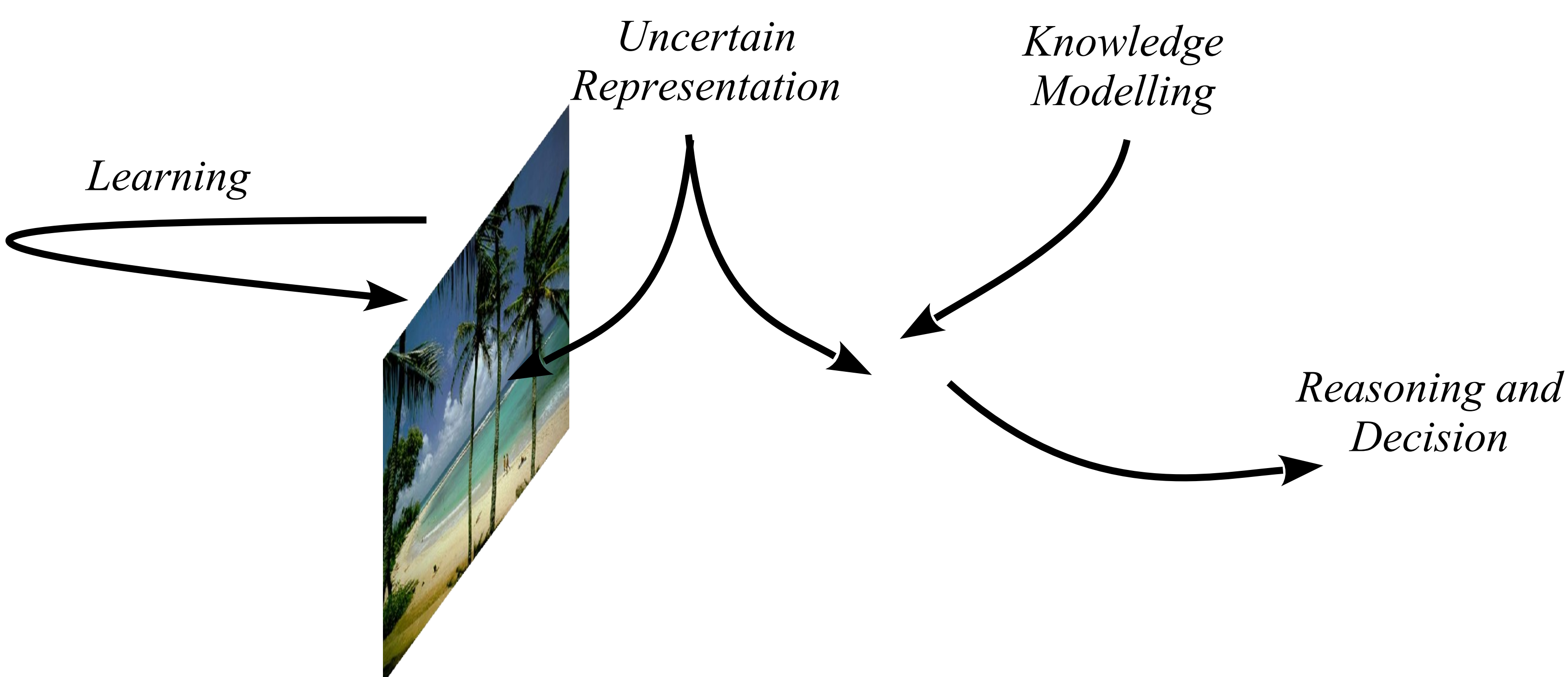
In our modern conception of *embodied* Artificial Intelligence, the machine *acts* onto the external world, eventually *moves*, and needs to *perceive* its environment so that it can *adapt* to it.

Vision is an extremely *rich* source of information, that allows the machine to *localise*, *recognise* objects or people, at a small cost, a low energy, and in a *passive* way (i.e. without emitting signal).



...and AI for Vision

Reciprocally, image processing and computer vision exploit knowledge and ML techniques to address the adaptation problem to a dynamic environment, the uncertain information, the heterogeneous knowledge and the different levels of decision making.



IP and visual perception

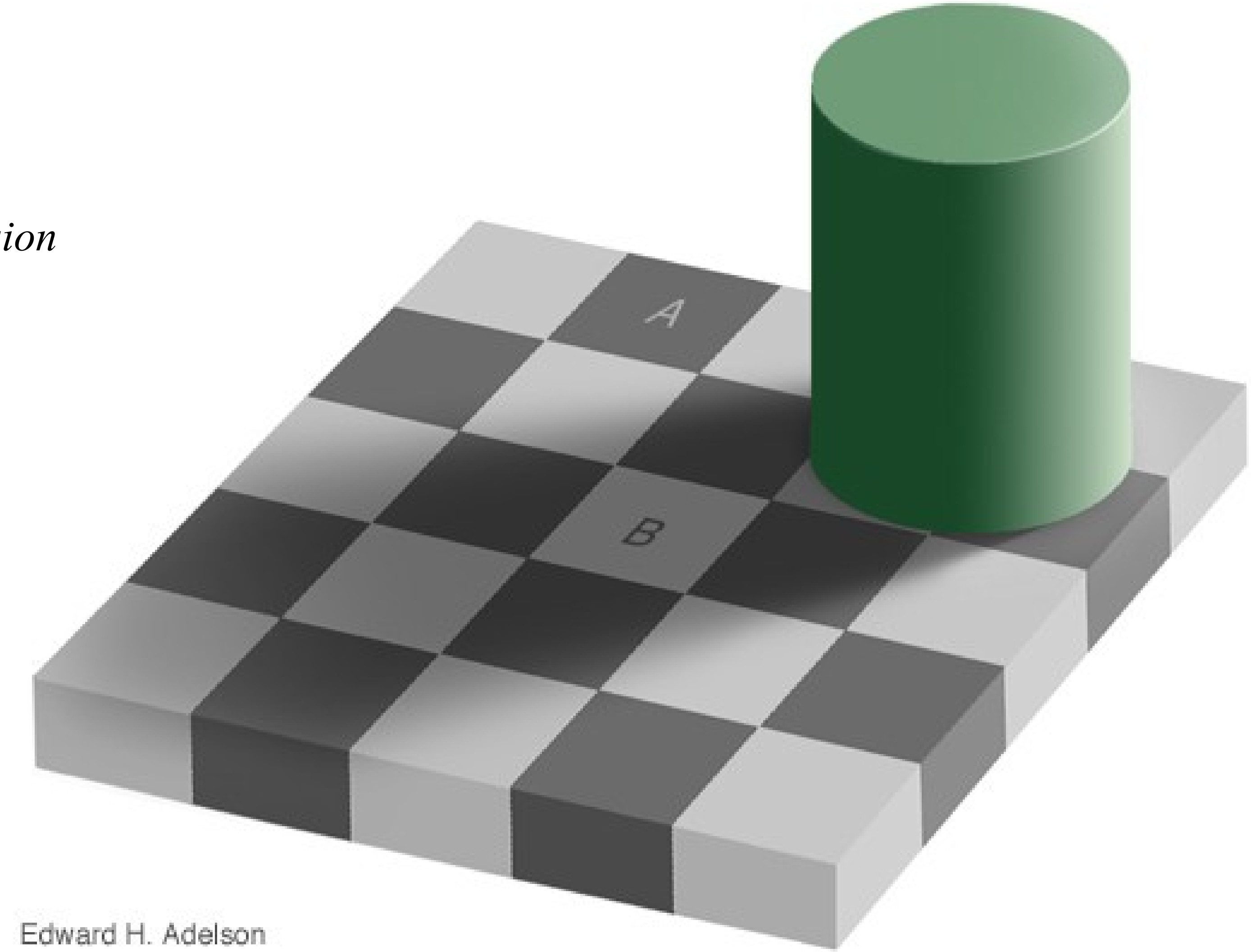
A fundamental difficulty of computer vision comes from the lack of deep knowledge of the biological mechanisms of image understanding. Human vision is extremely powerful (navigation, reading, recognition), but without any conscious feedback on the underlying mechanisms (as opposed to many “difficult” tasks like playing chess or calculating a division for example). In this sense studying physiological and psychological mechanisms of vision are a major source of information and inspiration.

Examples:

- Retinal / Cortical processes.
- Contrast enhancement mechanisms.
- Retina and multi-resolution.
- Motion and frog vision.
- .../...

IP and visual perception

Example:
The checker board illusion

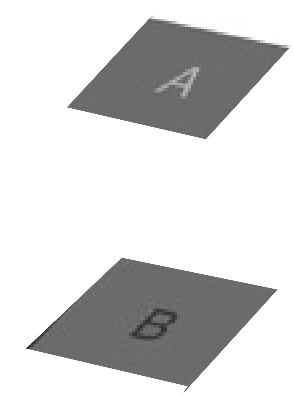


IP and visual perception

Checker-shadow illusion:
The squares marked A and B
are the same shade of gray.

Example:
The checker board illusion

Several mechanisms are operating here, from the very early level of perception (local enhancement of contrasts) to the very high level of understanding (interpretation of the cast shadow and recognition of the checker board)



Edward H. Adelson

Modalities and Sensors...

Physical phenomenon	*Measured quantity*	*Sensor*
Emission and Reflection of visible light	Intensity, Reflectivity,...	CCD Camera, CMOS Sensor...
Infrared emission	IR luminance (heat), ...	Bolometers,...
Ultrasound echo	Distance, tissue densities,...	Echography, sonar,...
Magnetic Resonance	Presence of a chemical body...	IRM, RMN,...
Electromagnetic echo	Distance, surface specularities,..	Radar, SAR,...
X-ray Absorption	Tissue densities,...	Radiography, CT scanners,...

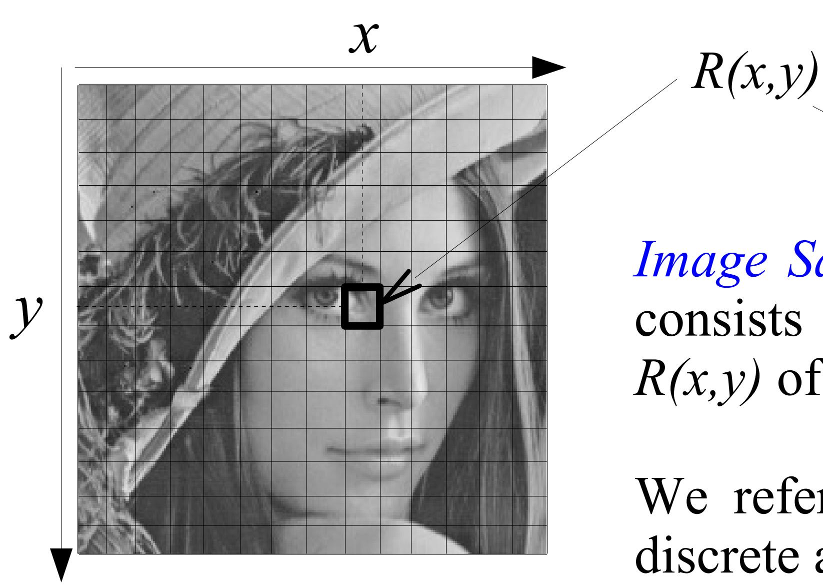
Models in Image Processing

Different models coexist in Image Processing.

A model formally defines an image, and then, operators and algorithms that are applied to it. We will consider the following families of modelling:

- **Linear model** / **An Image ↔ A vector**
- **Statistical model** / **An image ↔ A random field**
- **Frequency based model** / **An image ↔ A periodic function**
- **Differential model** / **An image ↔ A differentiable function**
- **Set based model / An image ↔ A set**
- **Discrete model / An image ↔ A discrete set or function**

Digital images



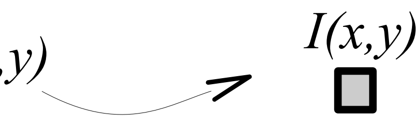


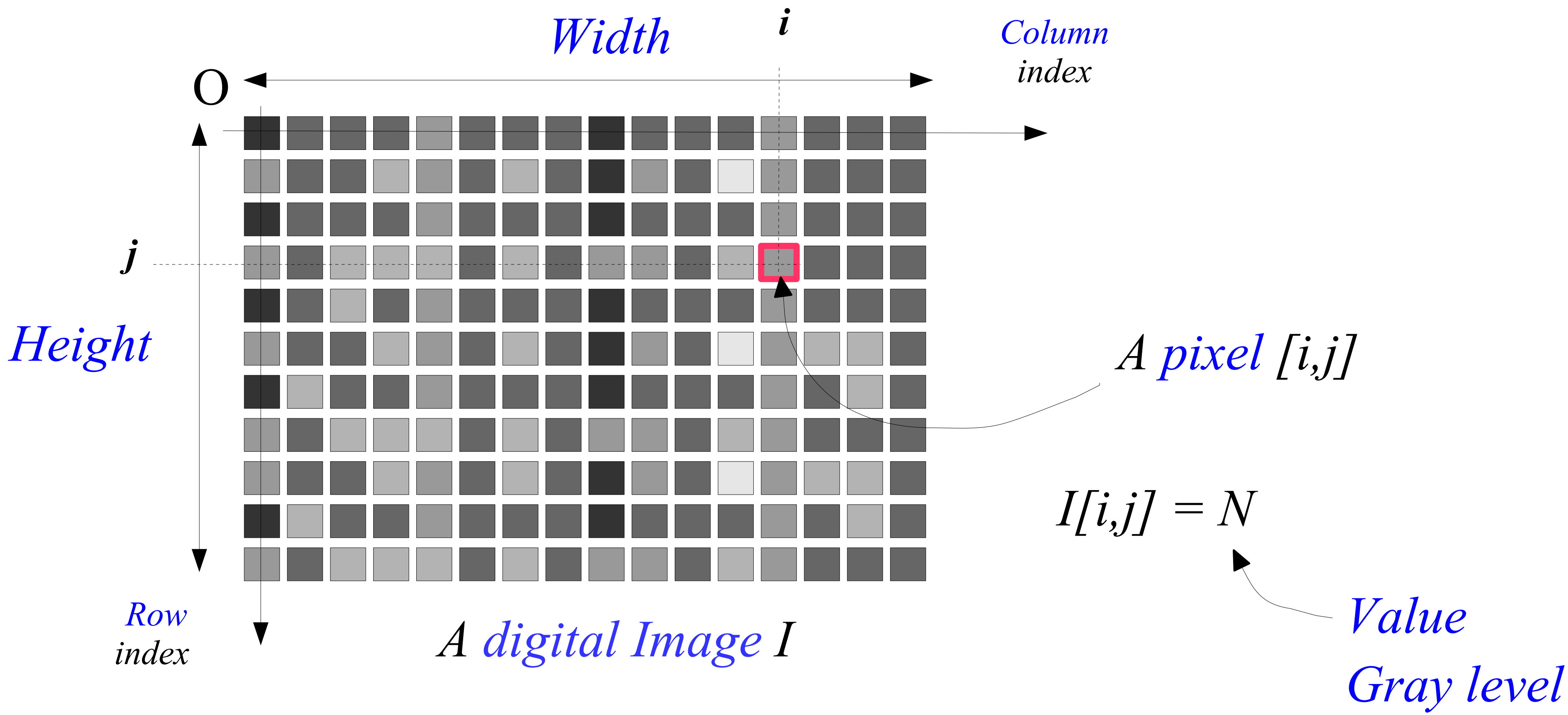
Image Sampling is the process of spatial digitization that consists in associating to each rectangular zone (tile) *R(x,y)* of a continuous image a unique value *I(x,y)*.

We refer to *sub-sampling* when the image is already discrete and the number of samples is decreased.

Quantization refers to limiting the number of distinct values of *I(x,y)*.

A *digital image* is both *sampled* and *quantized*.

Pixels and gray level



$$N \in [N_{min}, N_{max}]$$

$$(N_{max} - N_{min}) = \text{number of gray levels}$$

$$Log_2(N_{max} - N_{min}) = \text{dynamics}$$

Sampling and Quantization

Changing the resolution...

...in the spatial domain: *Sampling*

256x256

128x128

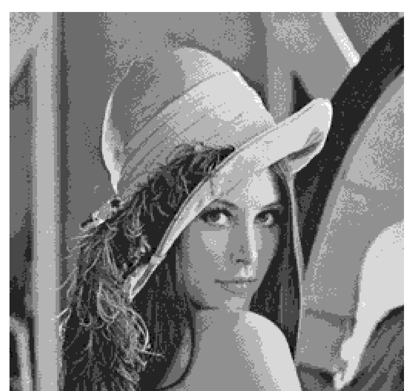
64x64

32x32

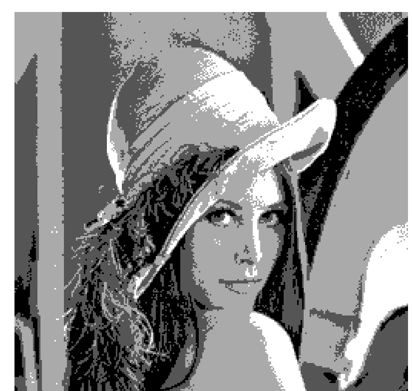
...in the tonal domain: *Quantization*

6 bits

4 bits



3 bits

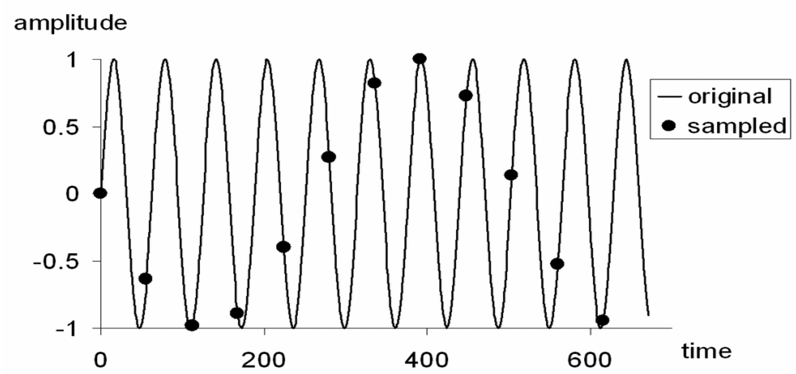


2 bits

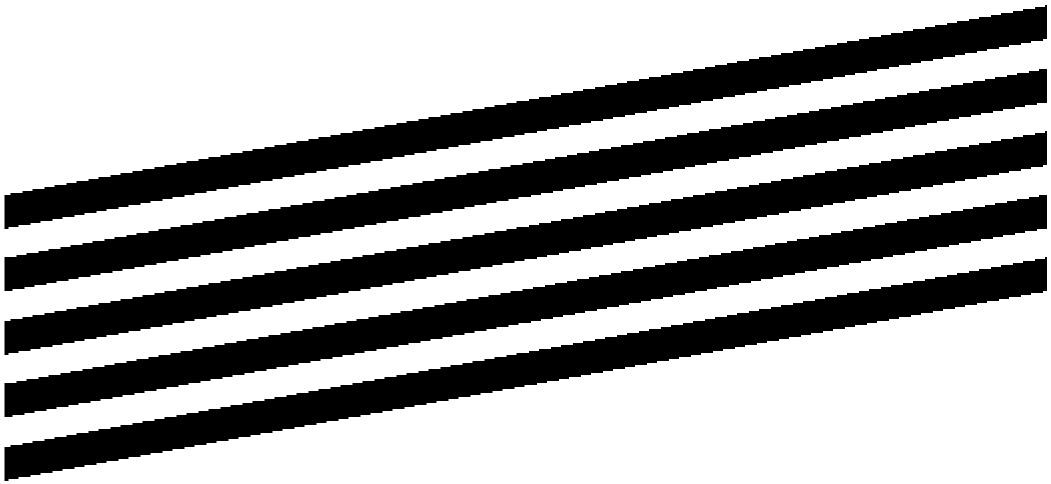
1 bit

Sampling and information

Sampling is a fundamental step that must take into account the relevant information from the image to be analysed. On the example on the right (in 1d), the sample signal looks like a sinusoid with a frequency 8 times smaller than the continuous one:

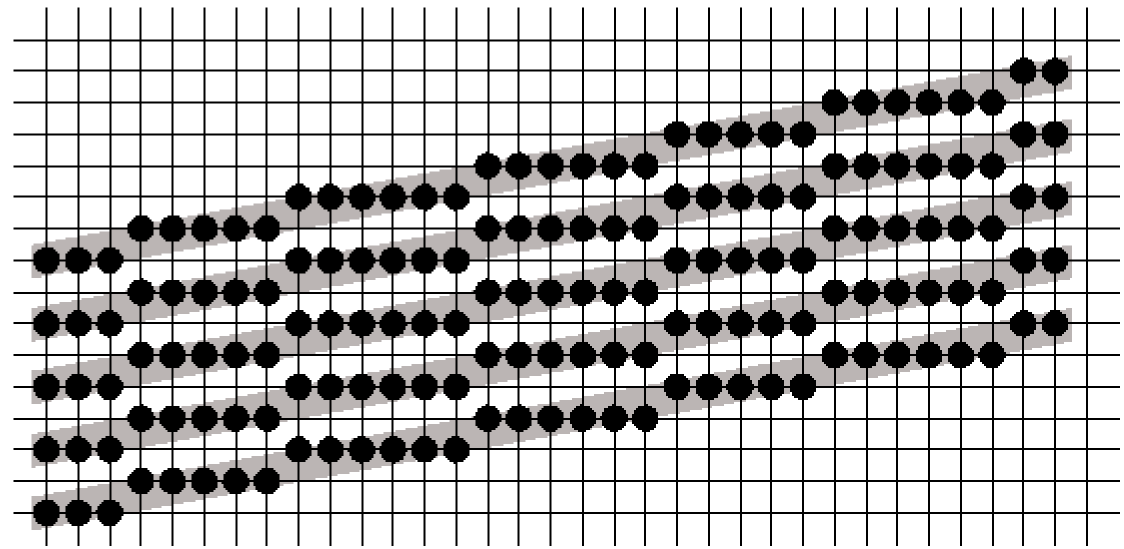


This phenomenon called *aliasing* is somewhat worse in 2d, since it affects *frequency and orientation* of periodical structures. Suppose for example that we wish to sample the image on the right with the black stripes:

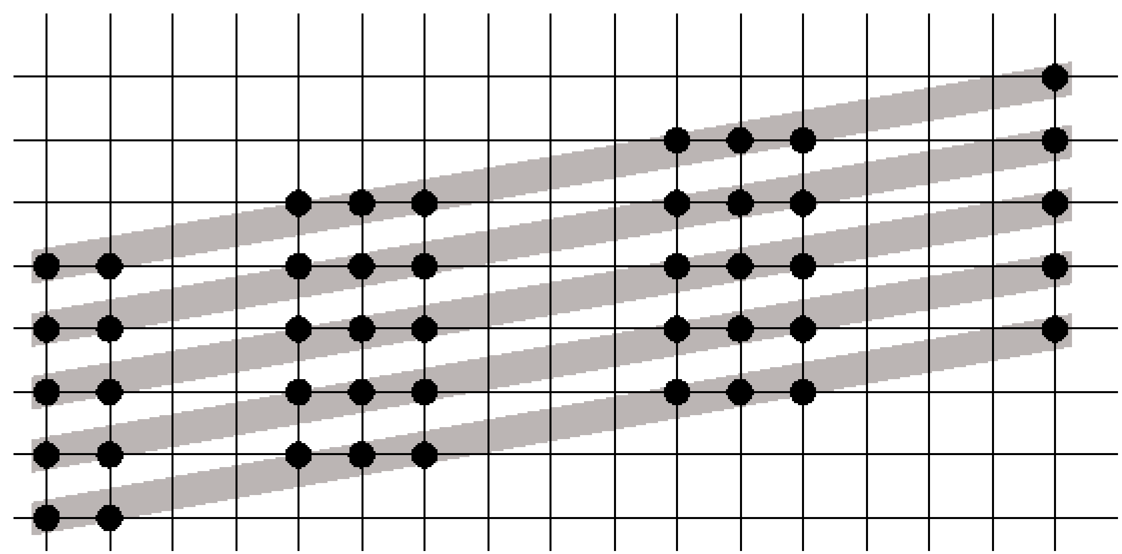


Sampling and information

With an adapted sampling, the digital image shows structures that are conform to the information present in the continuous image:

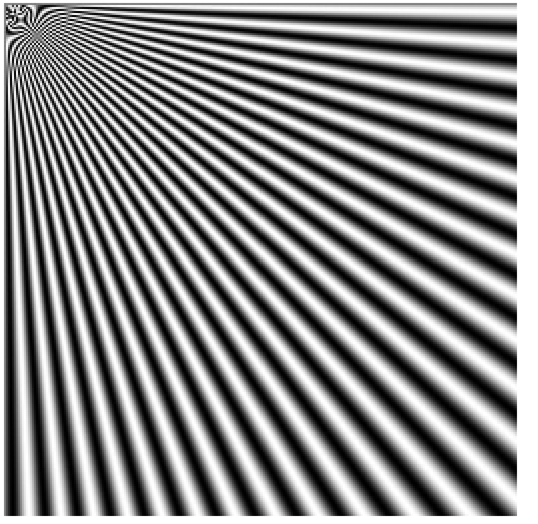


But if we consider only 1 sample over 2 in each dimension, a different structure appears, whose analysis (here thicker and vertical stripes) will not conform to the real object:



Sampling and information

Another example, on a synthesis image:



...and on a natural image:

Original image

Sub-sampled image

Quantization and information

Quantization also creates distortions in images:

$$I_{quant} = \left\lfloor \frac{I}{n_{quant}} \right\rfloor$$

As for sampling, there are rules to determine the right quantization (the right number of bits) to encode digital images.
One is depending on the *sensor*, and its effective capability to distinguish signals with different magnitudes: the *signal-to-noise ratio*.
It is defined from the ration between the *amplitude of gray levels* that can be measured by the sensor (n_{max} - n_{min}) and the *level of noise*, corresponding to the standard deviation s_n of the random perturbation that affects the gray levels. By taking the logarithm, we get the number of useful bits to encode the images.

Quantization and information

Aside from the sensor capabilities, the number of bits actually necessary to encode an image varies according to the *information content*.
This is related to the *entropy*, defined from the distribution of gray levels (see further, the statistical model).

$$E = \sum_{i \leqslant N} - p_i \log_2(p_i)$$

Where N is the number of gray levels, p_i is the ratio ($0 < p_i < 1$) of pixels with gray level equal to i. This quantities measures the average number of bits per pixel necessary to encode the whole information. It is used in lossless compression techniques to adapt the volume of image data to their information content.

III Models and fundamental tools

We now present an introduction to the most common digital imaging processing tools. For tutorial purposes, the presentation is structured according to the main *mathematical models* employed to process images. HOWEVER those different models are neither exclusive nor clearly separated and the distinction will be hardly visible in the next lectures.

Some *fundamental tools* are associated to each different model, that have (or have had) an important role in image processing, from a theoretical or practical point of view. Let us mention: *convolution*, *Fourier transform*, *histogram*, *pyramids*, *correlation*, wav*elets*...

Exploring the different models is seen here as the opportunity to introduce some of those tools. In this introduction the focus will be put on:
Convolution, Fourier transform, histogram, partial derivatives.

III-1: The linear model

In the linear model, the underlying mathematical structure is the *Vector Space*. Basic operators are then those that preserve the structure of vector space, that is the *linear applications*:

$$f(I+J)=f(I)+f(J)$$
$$f(\lambda I)=\lambda f(I)$$

For images (translation invariance), this correspond to *convolutions*:

<u>Convolution</u> :

This is the basic operator of linear image processing. Appeared very early in the first image processing systems for the sake of simplicity, it was later justified by physical considerations and by the theoretical links with filters in signal processing.

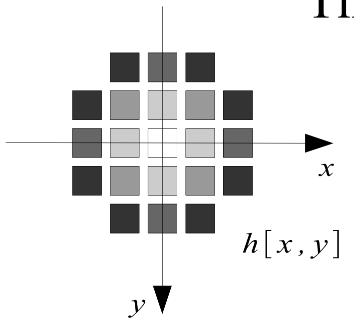
Convolution

Let I be a digital image.
Let h be a real-valued function from $[x_1,x_2] \times [y_1,y_2]$.

The *convolution* of I by h is defined as follows:

$$(I*h)[x,y]=\sum_{i=x_1}^{x_2}\sum_{j=y_1}^{y_2} h[i,j]\cdot I[x-i,y-j]$$

The function h is called *convolution kernel*



<u>Properties of convolution</u>:

COMMUTATIVITY $h*g=g*h$

ASSOCIATIVITY $(h*g)*k=h*(g*k)=h*g*k$

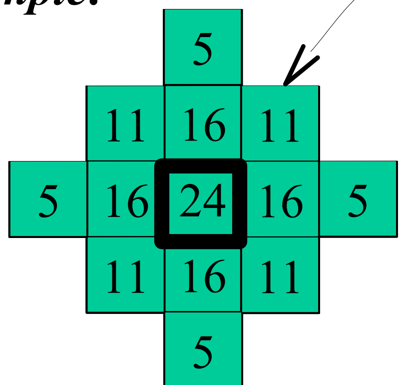
DISTRIBUTIVITY / + $h*(g+k)=(h*g)+(h*k)$

The new values of each pixel are calculated through the *scalar product* between the convolution kernel and the corresponding neighbourhood of the pixel.

Convolution

Example:

1/152

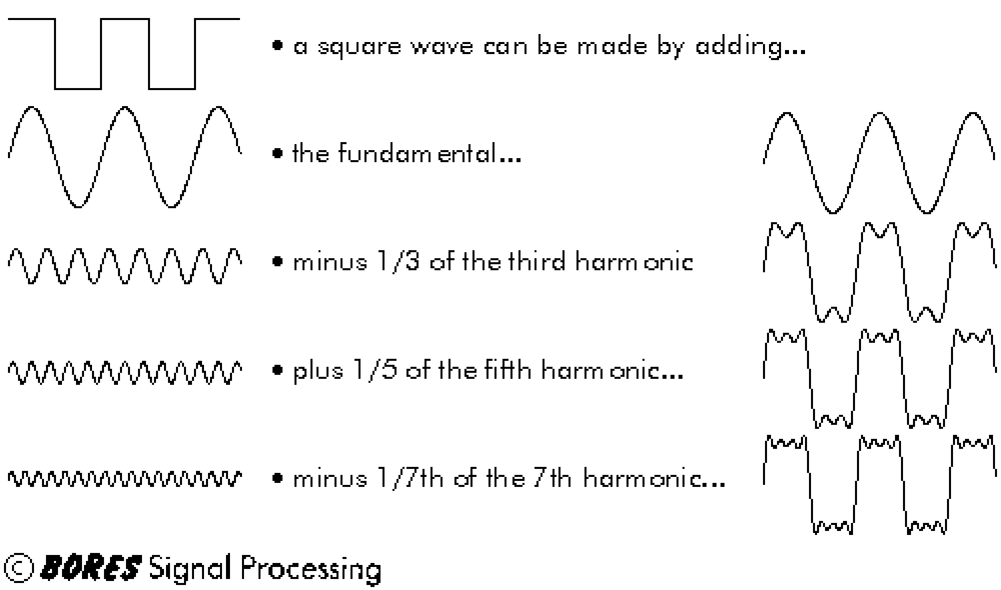


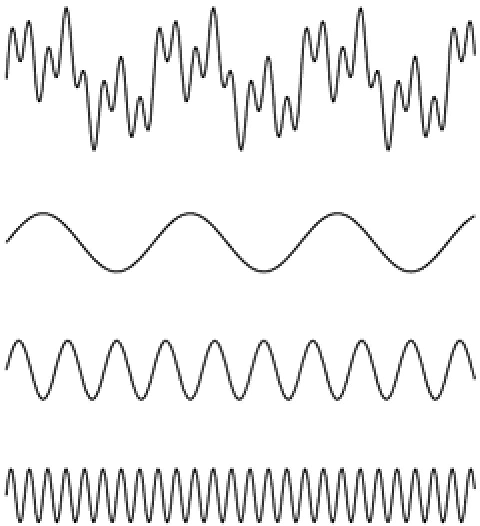
To calculate a convolution, the value of every pixel is replaced by the scalar product between the convolution kernel and the neighbourhood of the considered pixel (with respect to the origin (0,0) of the convolution kernel).

Attention: “parallel” implementation.

III-2: Frequency-based model

The frequency-based model describes the image in terms of *periodic structures*, by decomposing it into a *basis of simple periodic functions*, like sinusoids:





Fourier Analysis: the complex wave at the top can be decomposed into the sum of the three simple waves shown below.

Fourier transform:

Fundamental tool of signal processing, the 2d counterpart of the Fourier Transform and its discrete version can be applied to digital images. Its use as analytic tool has been mostly abandoned to the benefit of other approaches more adapted to the spatial location of frequencies (like wavelets), however its theoretical and tutorial role remains important: see formalisation of the aliasing (spectrum folding) and of sampling conditions.

Fourier Transform

The Fourier transform allows decomposing a signal *f* in *linear combination of complex sinusoids*, whose coefficients *F[u,v]* referred to as *Fourier coefficients*, provide information on *frequencies (u,v)* and allow *frequency domain* image manipulations.

2d discrete Fourier transform:

(x,y) are the coordinates in the *spatial domain*

Direct:

$$F[u,v]=\sum_{x=0}^{w-1}\sum_{y=0}^{h-1} f[x,y]\,e^{-2j\pi(ux/w+vy/h)}$$

(u,v) are the coordinates in the *frequence domain*

Inverse:

$$f[x,y]=\frac{1}{wh}\sum_{u=0}^{w-1}\sum_{v=0}^{h-1} F[u,v]\,e^{2j\pi(ux/w+vy/h)}$$

Properties of the Fourier Transform (1):

MODULUS / ARGUMENT FORMULATION $\quad F[u,v]=\|F[u,v]\|\,e^{j\varphi[u,v]}$

PERIODICITY $\quad F[u,v]=F[u+w,v+h]$

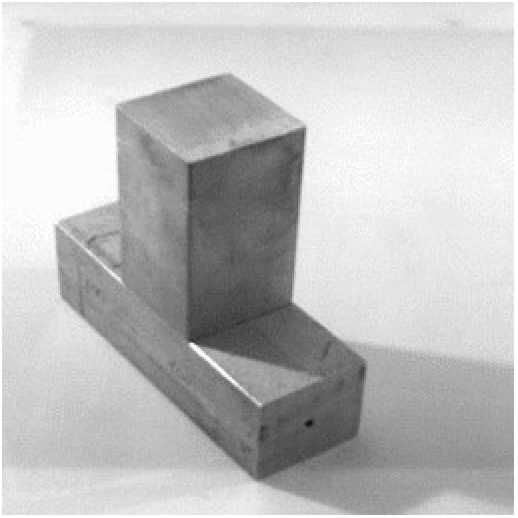
SYMMETRY

If F is the Fourier Transform of a real function f:

$F[u,v]=\overline{F[-u,-v]}$ and then: $\|F[u,v]\|=\|F[-u,-v]\|$ et $\varphi[u,v]=-\varphi[-u,-v]$

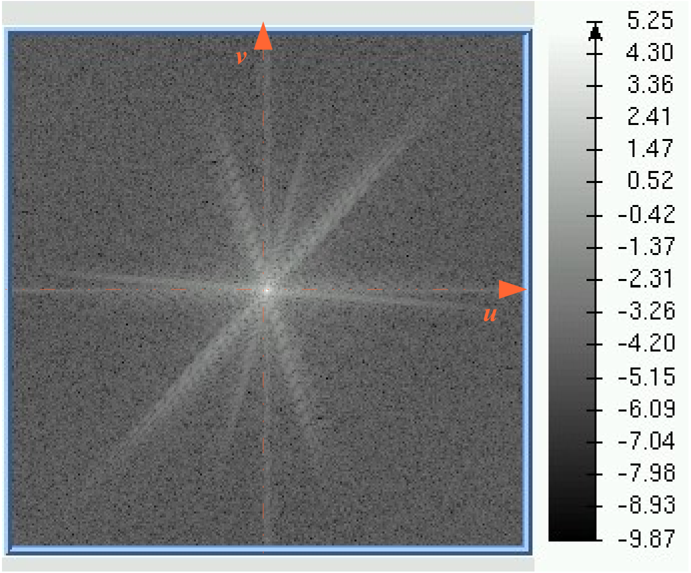
Fourier Transform

Image



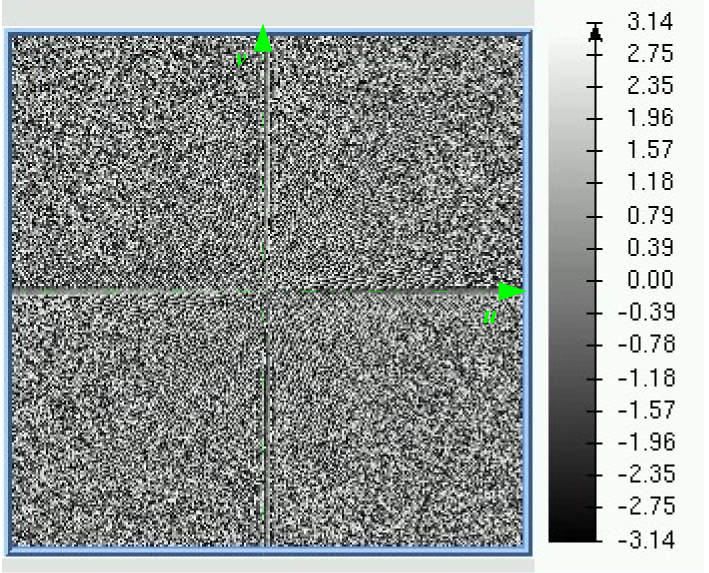
$f[x,y]$

Modulus



$\ln(\|F[u,v]\|)$

Phase



$\varphi[u,v]$

Properties of the Fourier Transform (2):

$$si \begin{cases} f[x,y] \xrightarrow{TF} F[u,v] \\ f_1[x,y] \rightarrow F_1[u,v] \\ f_2[x,y] \rightarrow F_2[u,v] \end{cases}$$

CORRESPONDENCE CONVOLUTION / PRODUCT

$$f_1[x,y] * f_2[x,y] \rightarrow F_1[u,v] \cdot F_2[u,v]$$
$$f_1[x,y] \cdot f_2[x,y] \rightarrow F_1[u,v] * F_2[u,v]$$

DERIVATION

$$\frac{\partial f[x,y]}{\partial x} \rightarrow juF[u,v] \; et \; \frac{\partial f[x,y]}{\partial y} \rightarrow jvF[u,v]$$

LINEARITY

$$a \cdot f_1[x,y] + b \cdot f_2[x,y] \rightarrow a \cdot F_1[u,v] + b \cdot F_2[u,v]$$

SPATIAL / FREQUENTIAL TRANSLATIONS

$$f[x-x', y-y'] \rightarrow F[u,v] \cdot e^{-2j\pi(ux'/w + vy'/h)}$$
$$f[x,y] \cdot e^{2j\pi(u'x/w + v'y/h)} \rightarrow F[u-u', v-v']$$

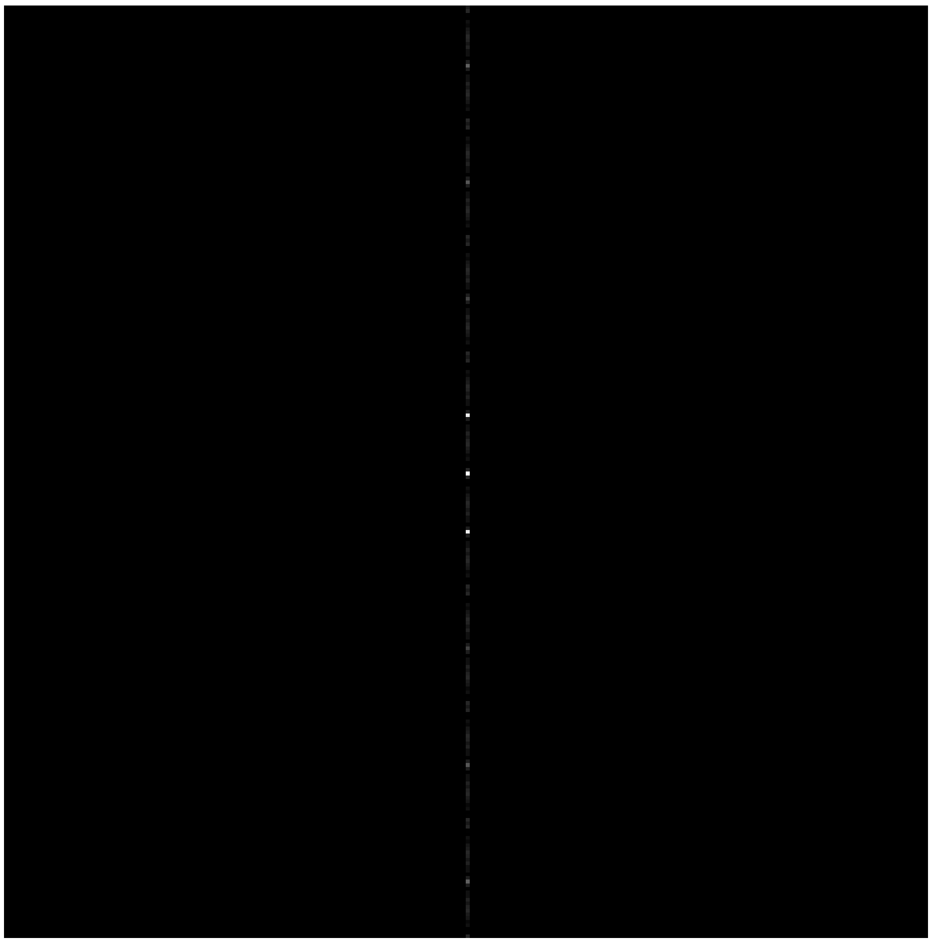
PARSEVAL THEOREM

$$\sum_{x=0}^{w-1} \sum_{y=0}^{h-1} \|f[x,y]\|^2 = \frac{1}{wh} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} \|F[u,v]\|^2$$

Understand the 2d Fourier spectrum

A simple 2d sine map:

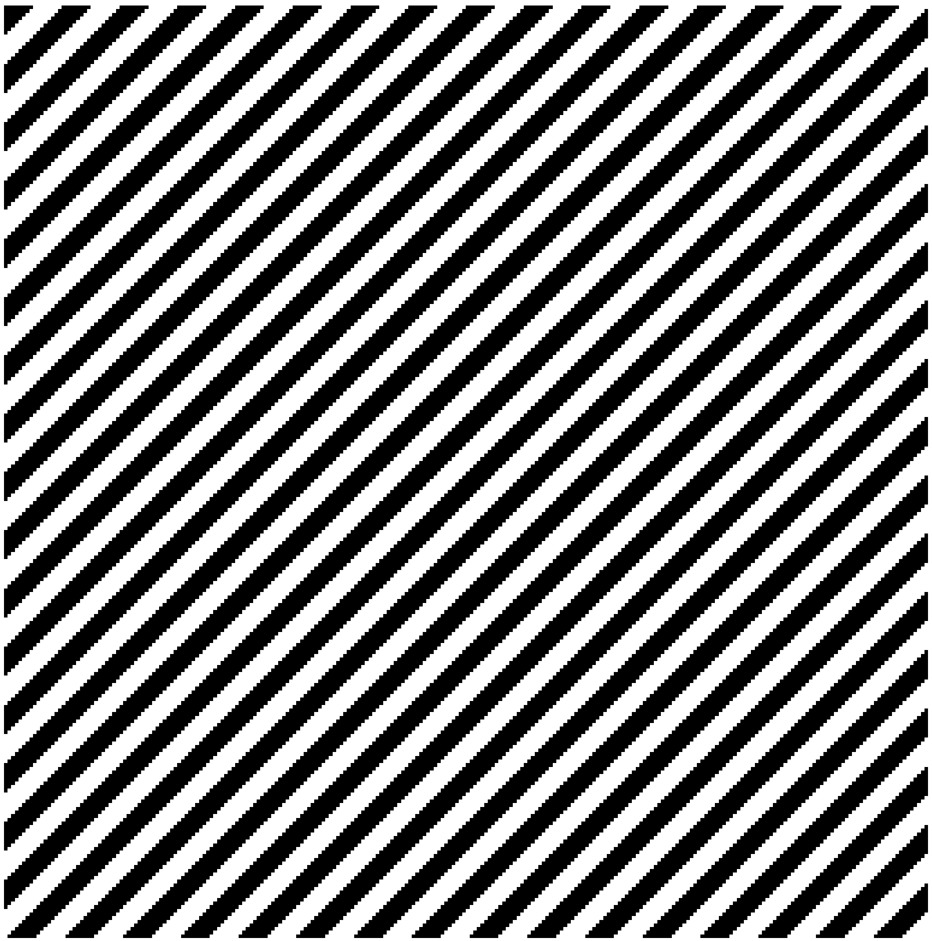
f



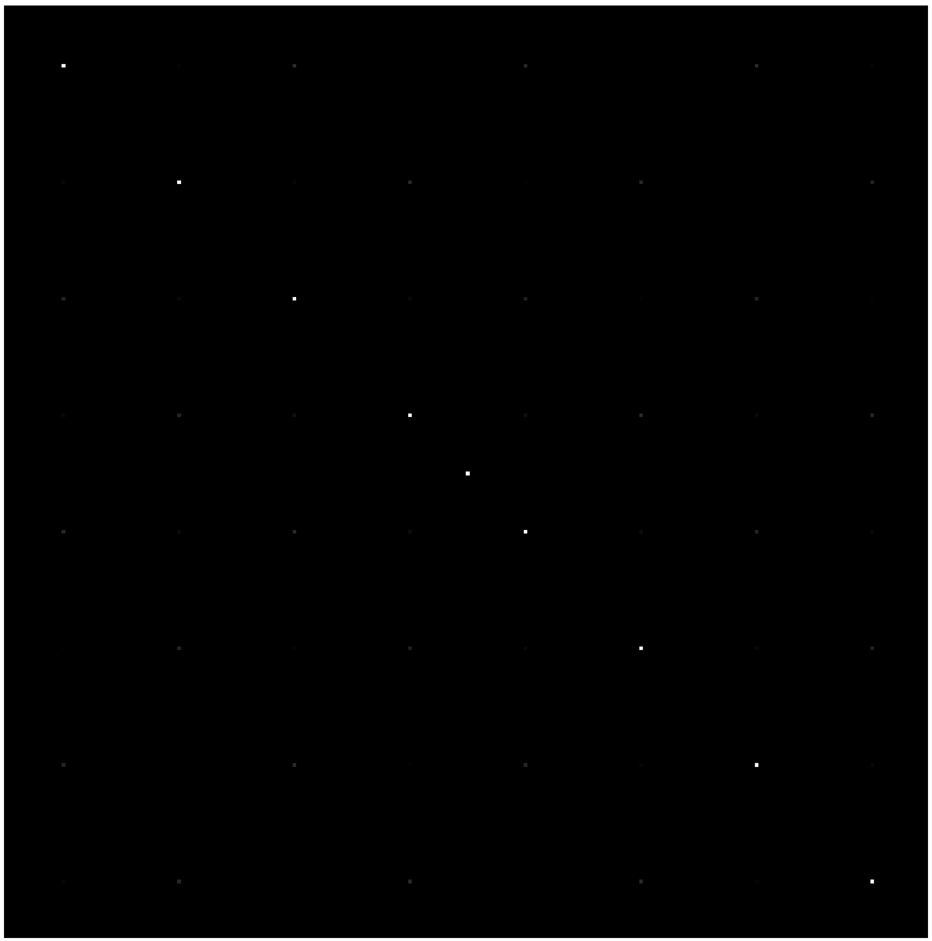
Log ||F||

Understand the 2d Fourier spectrum

A 2d square signal:



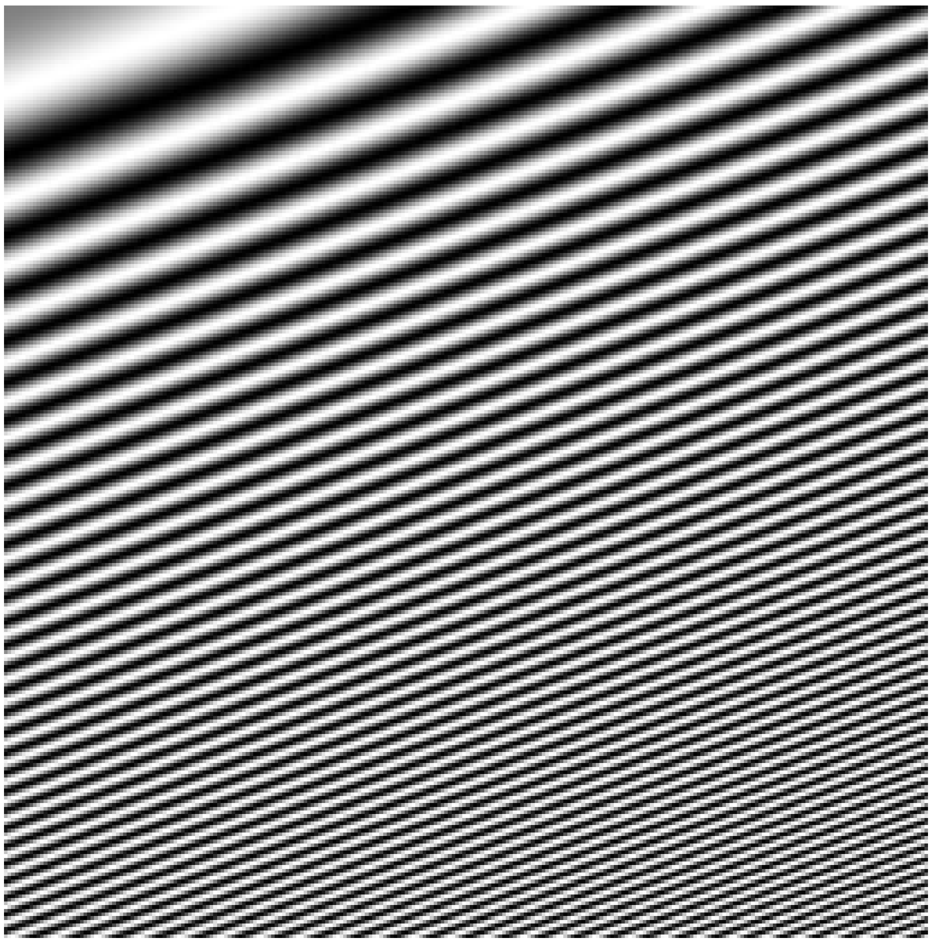
f



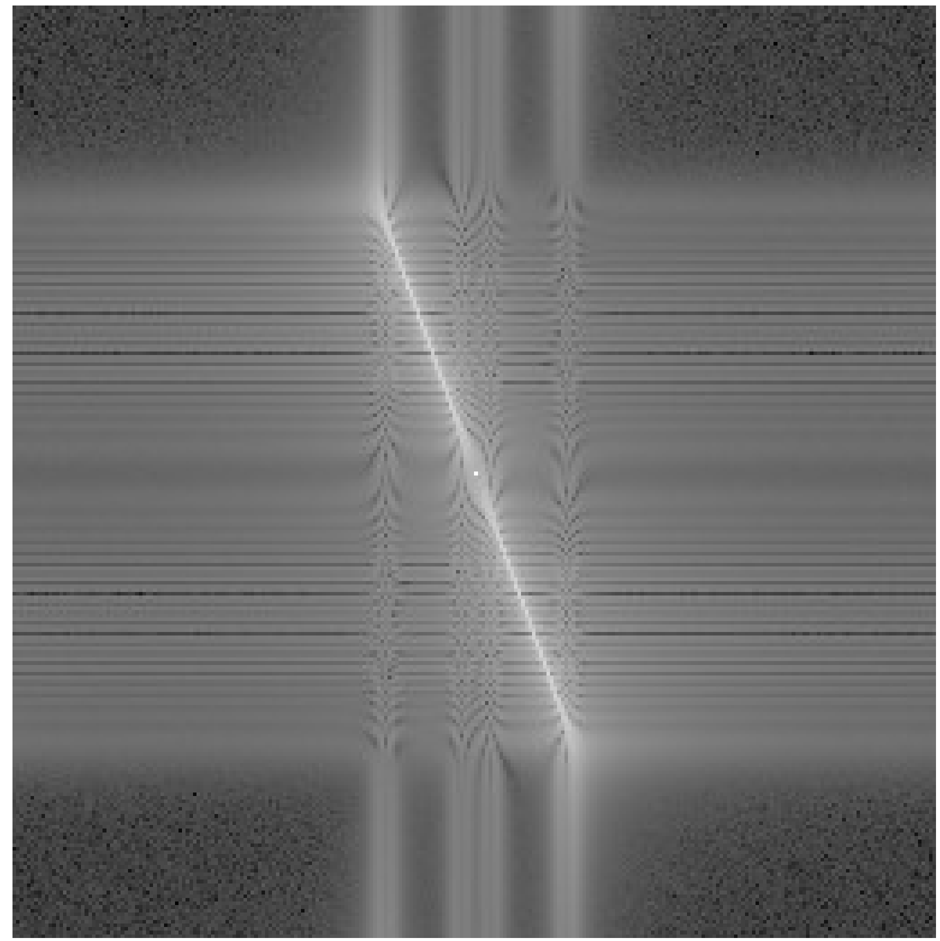
Log ||F||

Understand the 2d Fourier spectrum

A visual chirp:



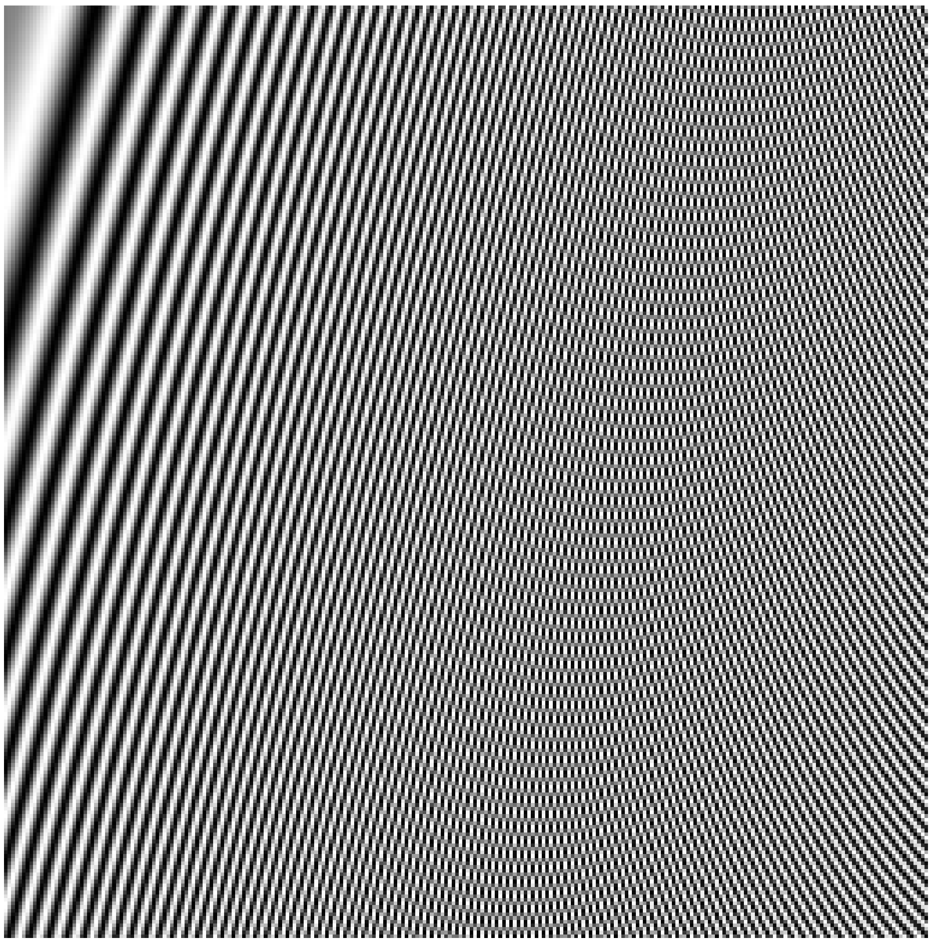
f



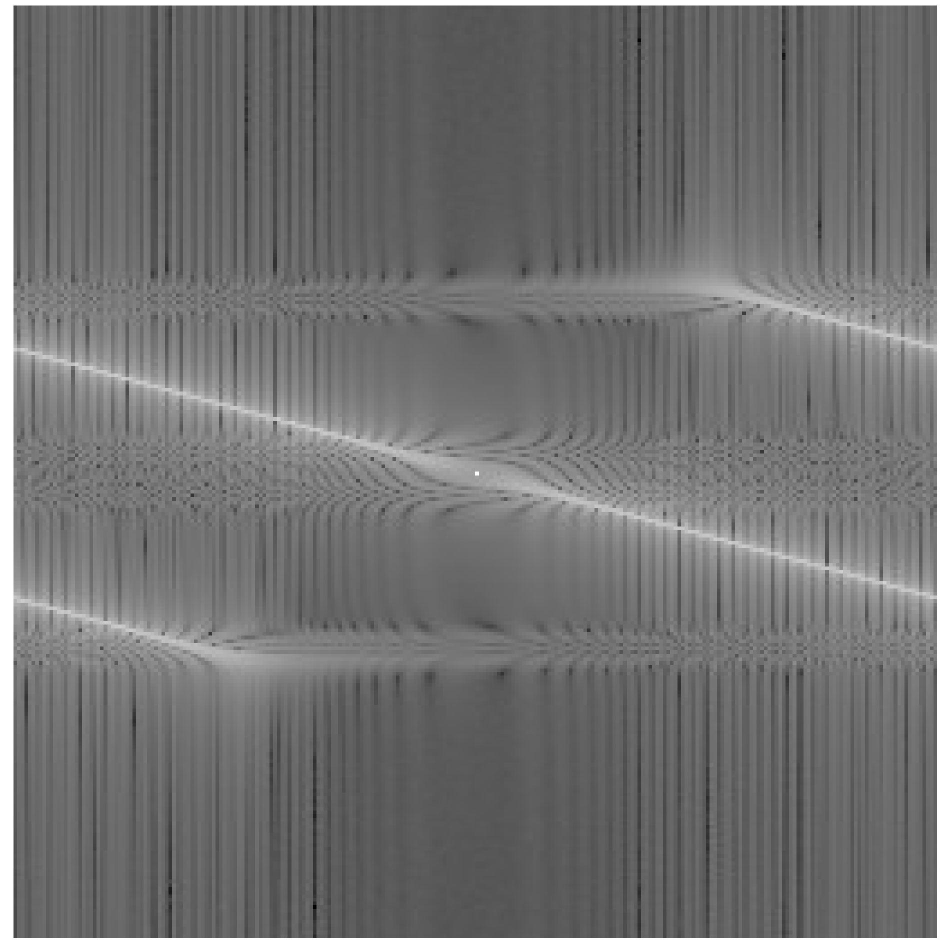
Log ||F||

Understand the 2d Fourier spectrum

Another visual chirp:



f

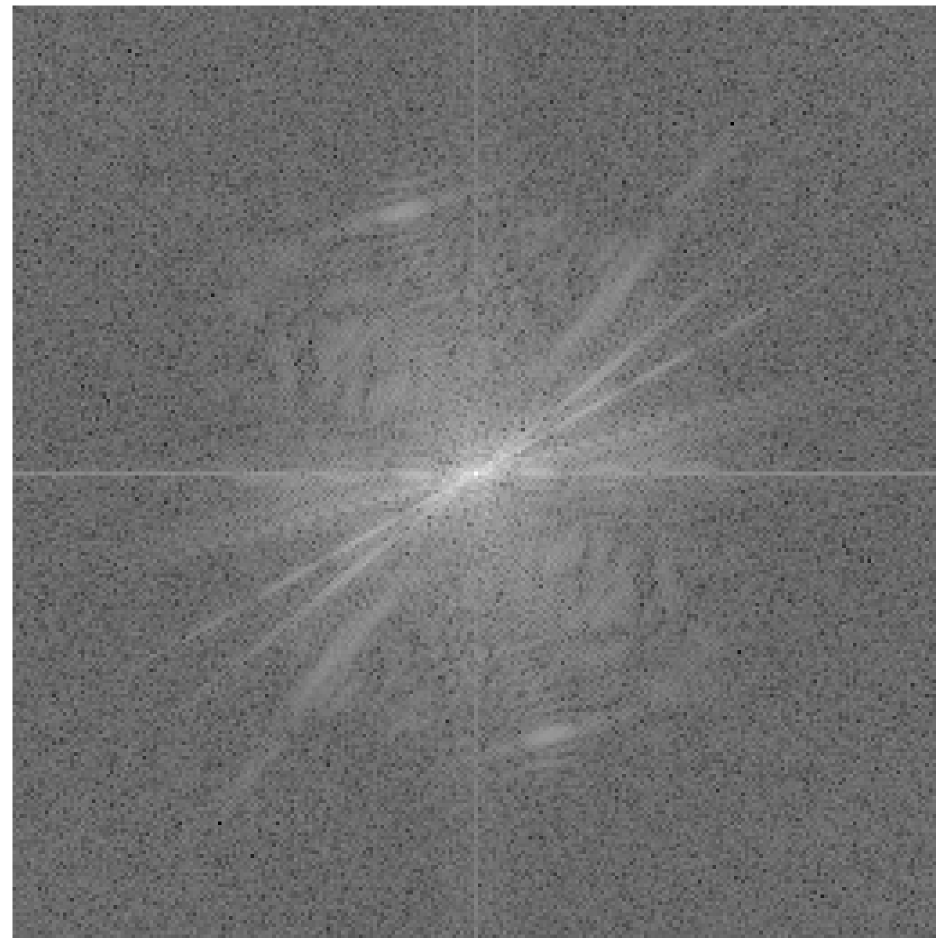


Log ||F||

Understand the 2d Fourier spectrum

A "natural" image:

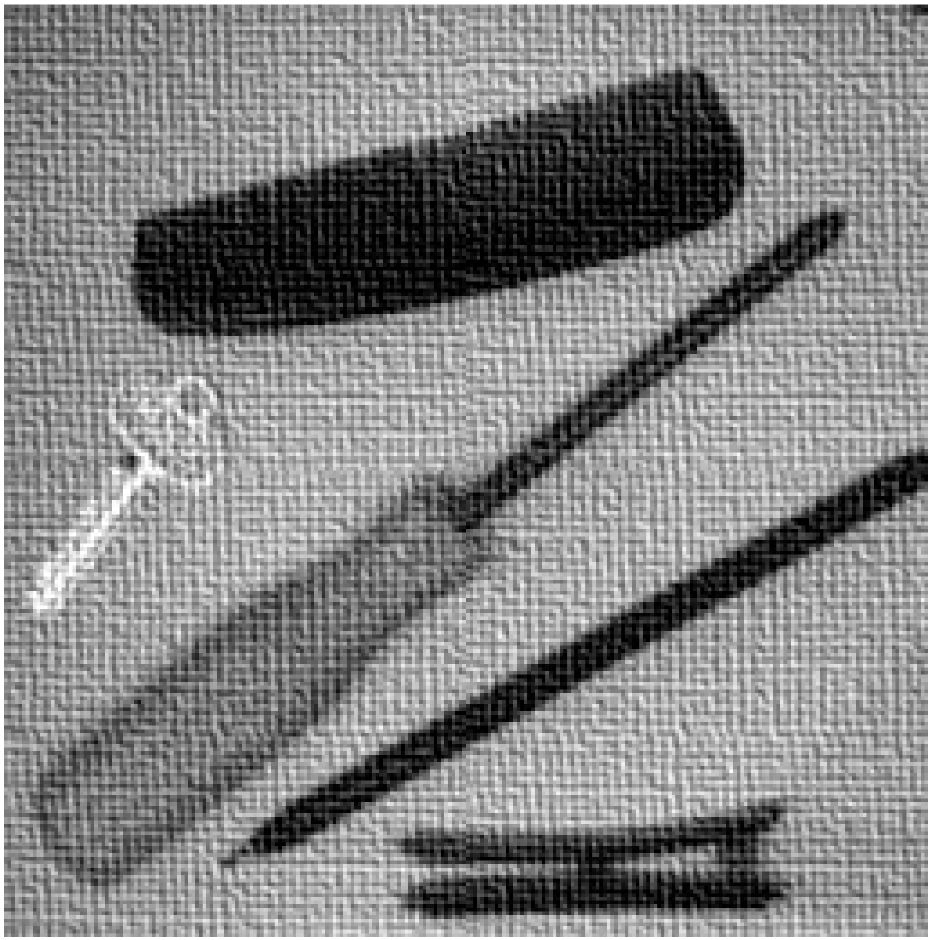
f



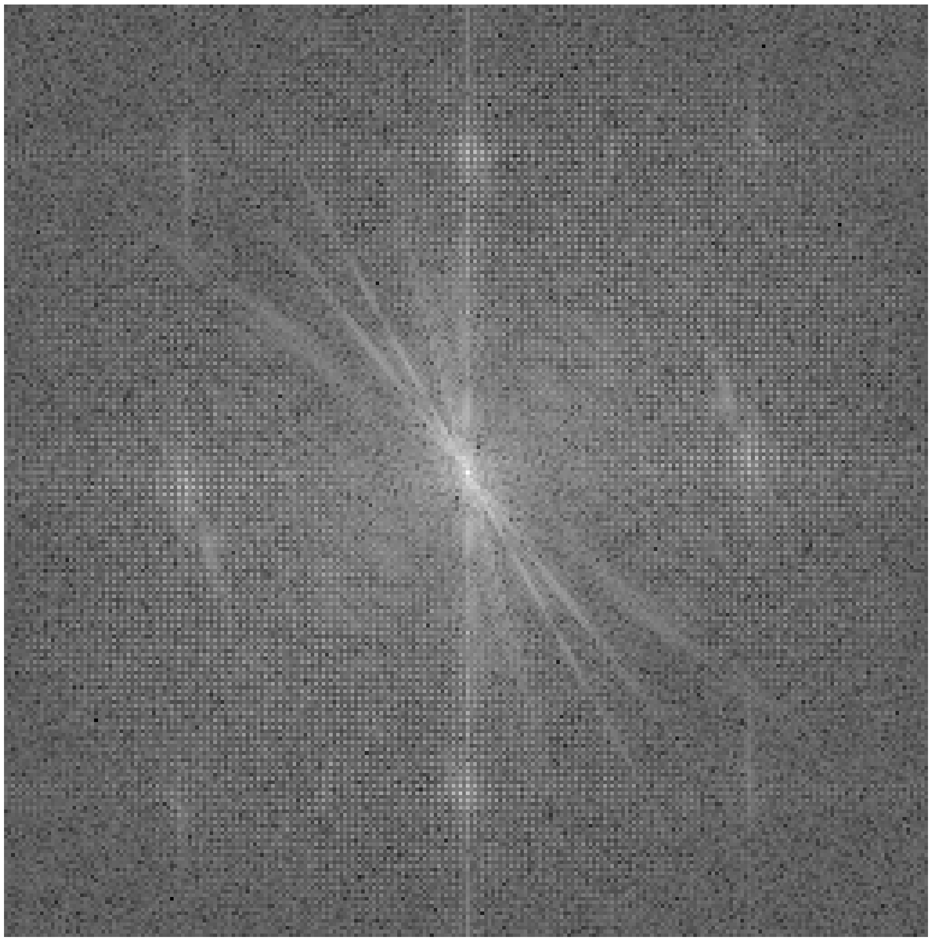
Log ||F||

Understand the 2d Fourier spectrum

Same, with rotation and texturing:



f

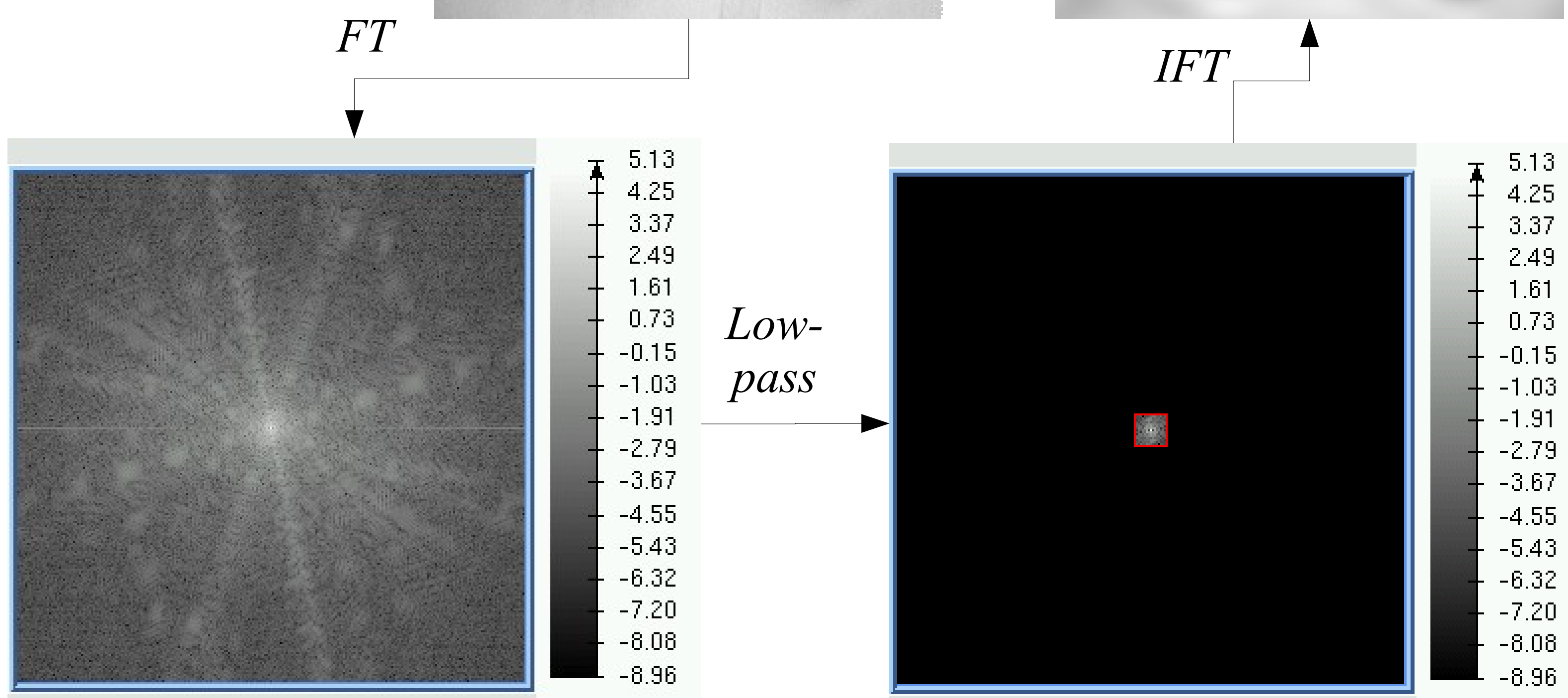


Log ||F||

Smoothing in the Fourier domain

Low-pass Filter

The (ideal) low-pass filter is the multiplication in the frequency space by a *gate function* (indicator function of the 2d interval: $[-u_{max},u_{max}]\times[-v_{max},v_{max}]$)

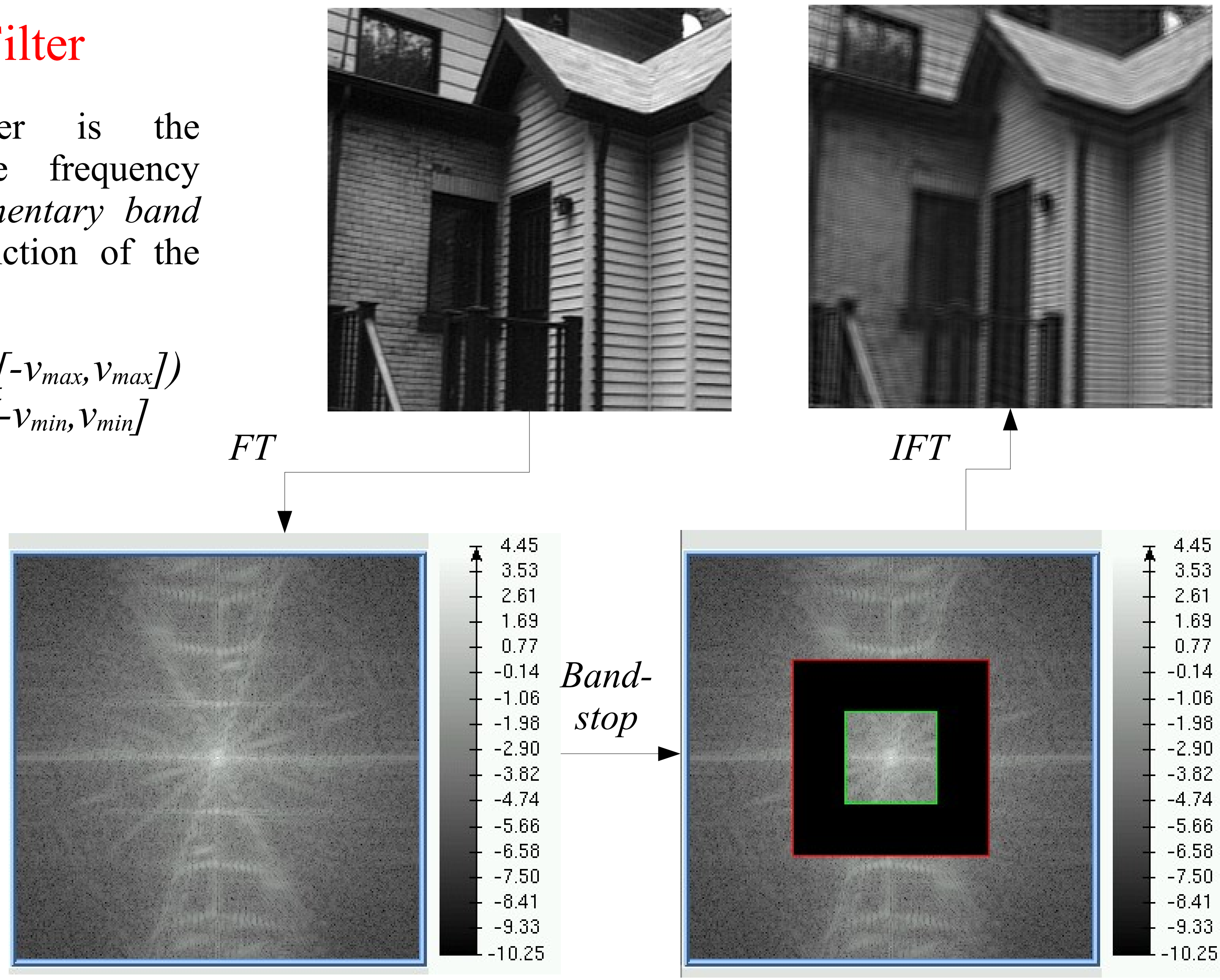


Band-stop in the Fourier domain

Band-stop Filter

The band-stop filter is the multiplication in the frequency domain by a *complementary band function,* indicator function of the set:

$$(\mathbb{R}^2 \setminus [-u_{max},u_{max}]\times[-v_{max},v_{max}]) \cup [-u_{min},u_{min}]\times[-v_{min},v_{min}]$$



Note that in this case and in the previous one, the *F[0,0]* component remains unchanged:

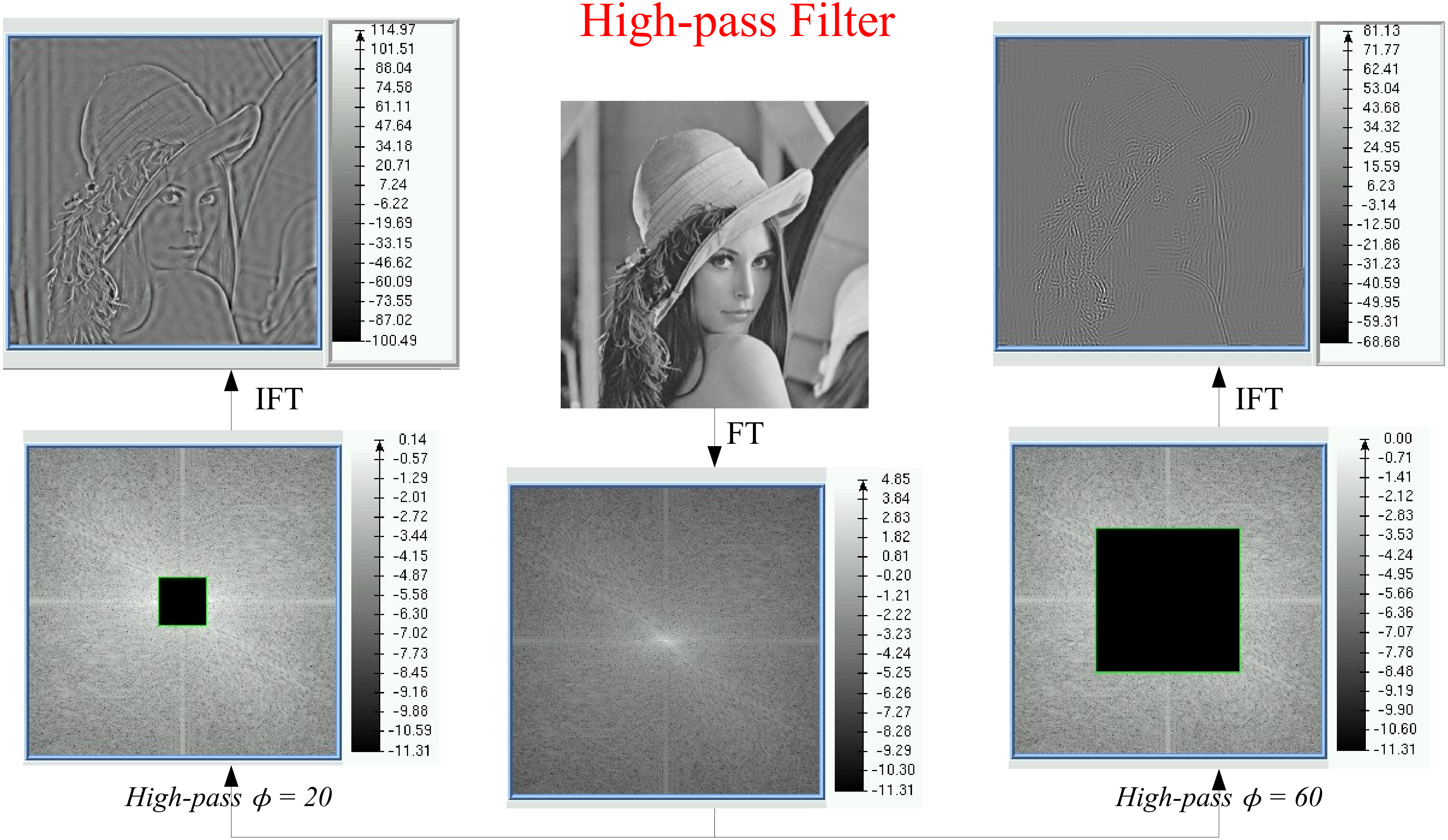
$$F[0,0]=\sum_{x=0}^{w}\sum_{y=0}^{h} f[x,y]$$

The sum of intensities is the spatial domain is then constant.

Details enhancing in the Fourier domain

The high-pass filter is the multiplication in the frequency domain by the complementary of a gate function.

High-pass Filter

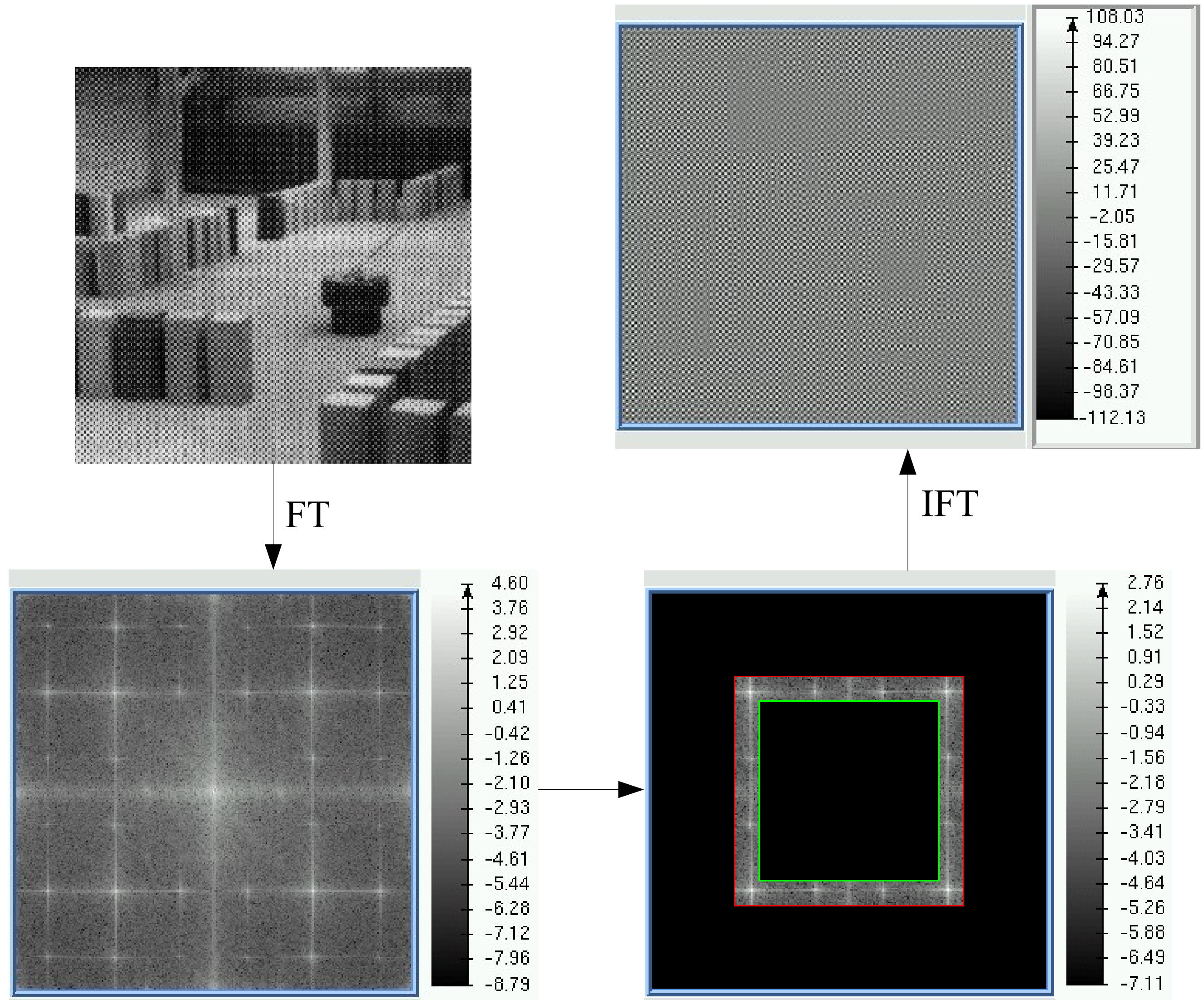


Band-pass in the Fourier domain

The band-pass filter is the multiplication in the frequency domain by a symmetric band function.

In this case and the previous one, the F(0,0) component is set to zero. The sum of the intensities in the spatial domain is 0, which means that the image has positive and negative values.

Band-pass Filter

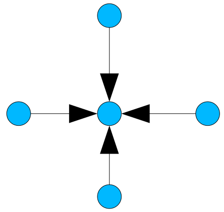


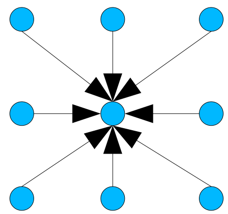
III-3: The statistical model

It deals with statistical properties of images: distribution of gray level values among pixels, correlation between spatially close pixels, occurrence frequency of some spatial structures...

Statistical measures provide empirical functions that can be used as probabilistic models in many algorithms.

For example, the Markov Random Field model considers the image as the realisation (event) of a random field (where each pixel corresponds to a random variable), where the value of a pixel only depends on its neighbours values (according to a certain discrete topology, see discrete models further).

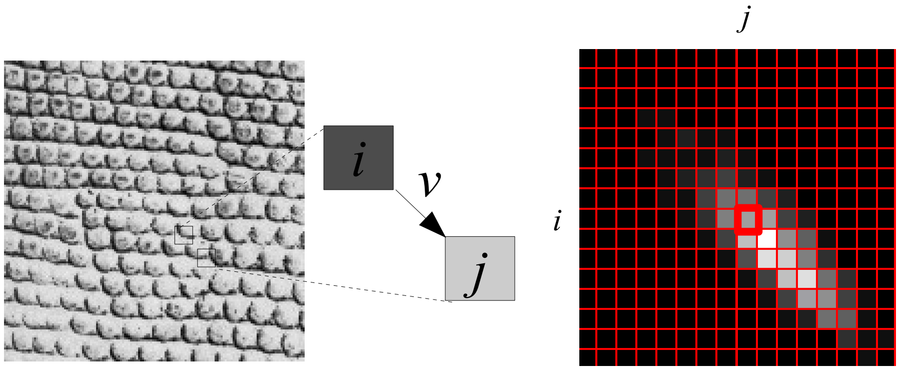




III-3: The statistical model

Another remarkable example of statistical analysis is the use of co-occurrence matrices to represent textures.

The co-occurrence matrix M_v associated to the shift vector v, is the N x N matrix (N is the number of gray levels), such that $M_v(i,j)$ is the frequency of occurrence of couple (i,j) amongst value couples of pixels $(x,x+v)$.

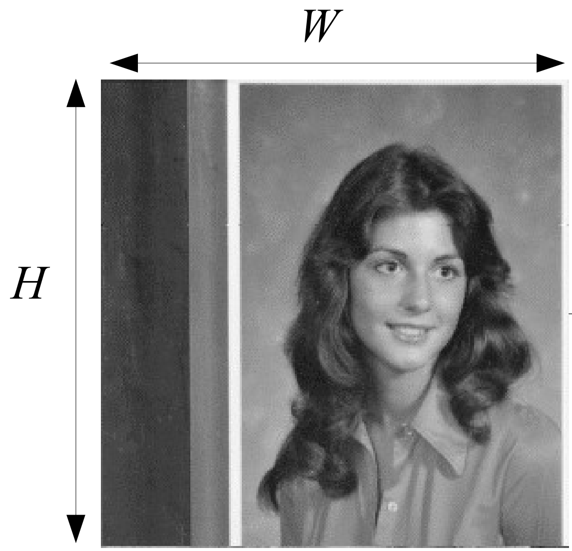


In this introductory lecture, we develop the first and most basic of these tools: the histogram.

<u>The histogram</u>:

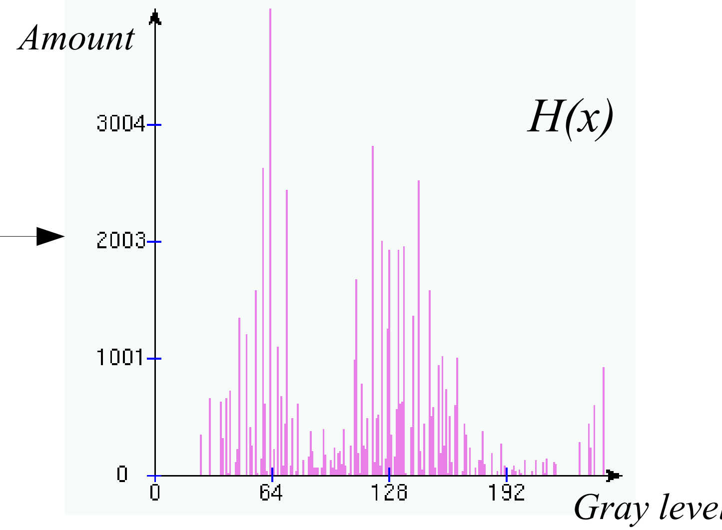
Basic tool for studying a sensor or the dynamics of a scene, it is used in some pre-processing operators. However the histogram of gray levels should not be considered as a fundamental analysis tool for images, since it may be dramatically transformed without changing significantly the image.

Histogram



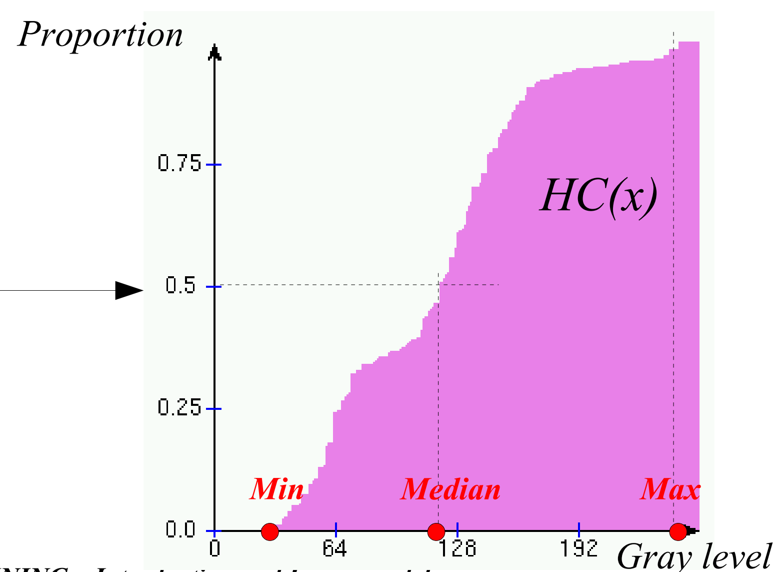
The histogram shows the repartition of pixels according to they (gray level) value. It provides diverse information like order statistics, entropy, and can be used in some specific cases to isolate objects.

Histogram



H(x) is the amount of pixels whose gray level is equal to x.

Normalised cumulative histogram



$$HC(x) = \frac{\sum_{i=0}^{x} H(x)}{W \times H}$$

HC(x) is the ratio of pixels whose gray level is less than x.

Histogram based processing

A few histogram based operators are shown in the following. It should be noted that they are often calculated within the sensor during the acquisition, and that their relevance greatly depends on the acquisition conditions.

(1) Normalisation

→ make the most of the whole encoding dynamics

(2) Equalisation

→ balance the encoding dynamics and globally enhance the contrast

(3) Segmentation

→ simplify the image by grouping pixels according to their values

Histogram: normalisation

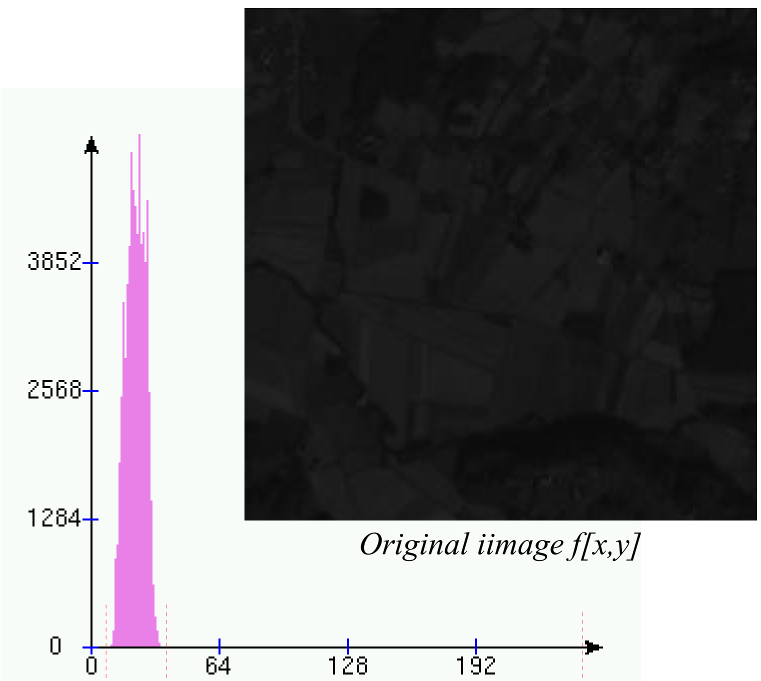
Histogram normalisation, or *dynamics expansion*, is an *affine function* of the gray level so that the image uses *the whole encoding dynamics*.

- ***D***: *dynamics*
- ***Nmin***: *smallest value in the image*
- ***Nmax***: *largest value in the image*

$$f_{new}[x,y] = (f[x,y] - Nmin) \cdot \frac{2^D - 1}{Nmax - Nmin}$$

To be less sensitive to outlier values, the normalisation can use a parameter β, 0<β<1, and take:

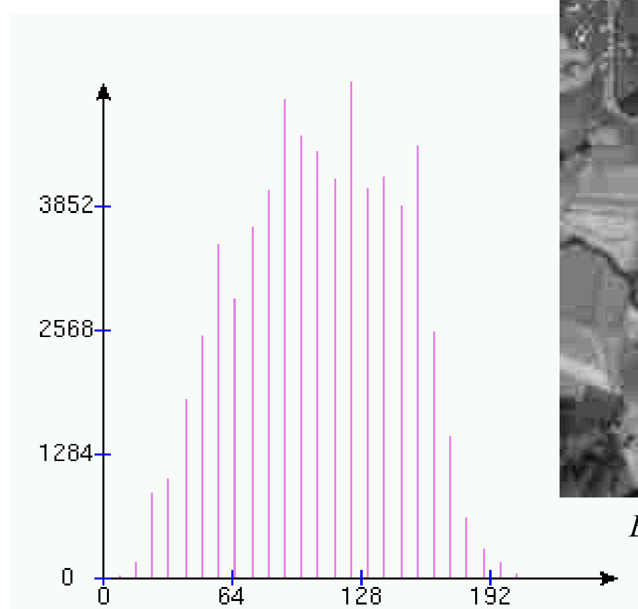
$$Nmin \in HC^{-1}(\beta)$$
$$Nmax \in HC^{-1}(1-\beta)$$



Original iimage f[x,y]

Nmin **Nmax** **2^D - 1**

Original histogram



Normalised histogram

Expanded dynamics $f_{new}[x,y]$

Histogram: equalisation

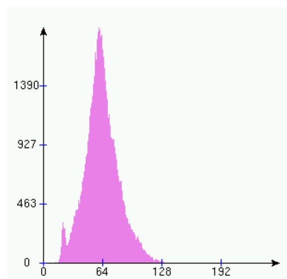
Histogram equalisation is a function of gray levels whose purpose is to *balance as uniformly as possible* the gray level distribution (The idea is to get as close as possible to a *flat histogram*).

- *The classic technique consists in linearising the cumulative histogram through the following operation:*

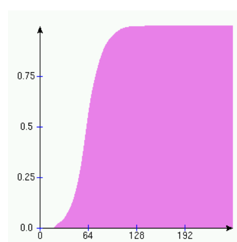
$$f_{new}[x,y] = (2^D - 1) \cdot \frac{HC(f[x,y])}{wh}$$

- ***D**: dynamics*
- ***(w,h)**: image dimensions*
- ***HC(.)**: cumulative histogram*

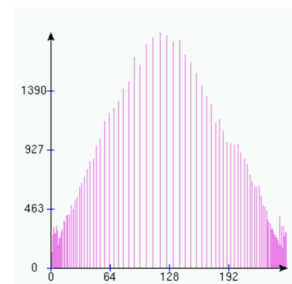
Original f[x,y]



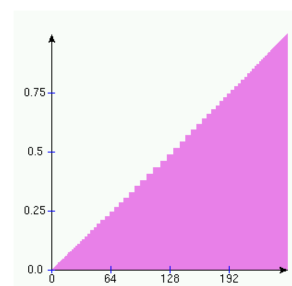
Histogram of f



Cumulative histogram of f



Histogram de f_{new}



Cumulative histogram of f_{new}

After equalisation $f_{new}[x,y]$

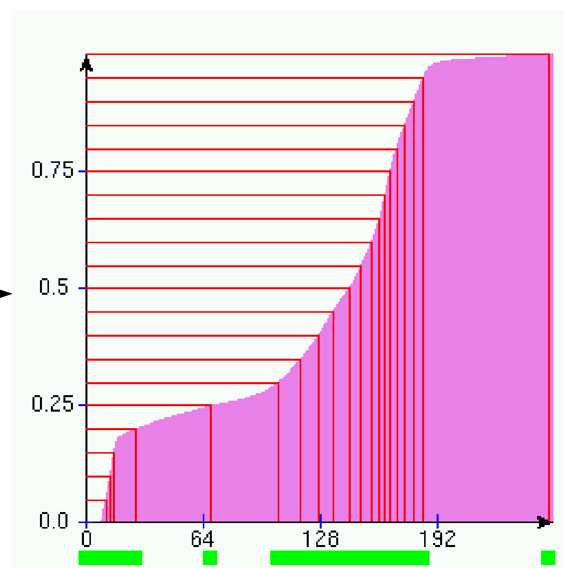
It results in a *global contrast augmentation* in the image. Note on the example above the enhancement of defaults also, like the fixed pattern noise of the uncooled infrared sensor.

Histogram: segmentation

There exists "segmentation" techniques based on grouping the gray levels from the histogram. These techniques are not effective in the general case since they only consider the values of pixels without regard to any geometrical or topological criterion.

For example, the method below calculates a given number (here 20) of quantiles from the cumulative histogram, then group them into classes based on a distance criterion, and attributes the same labels to pixels whose value belongs to that class:

Original image

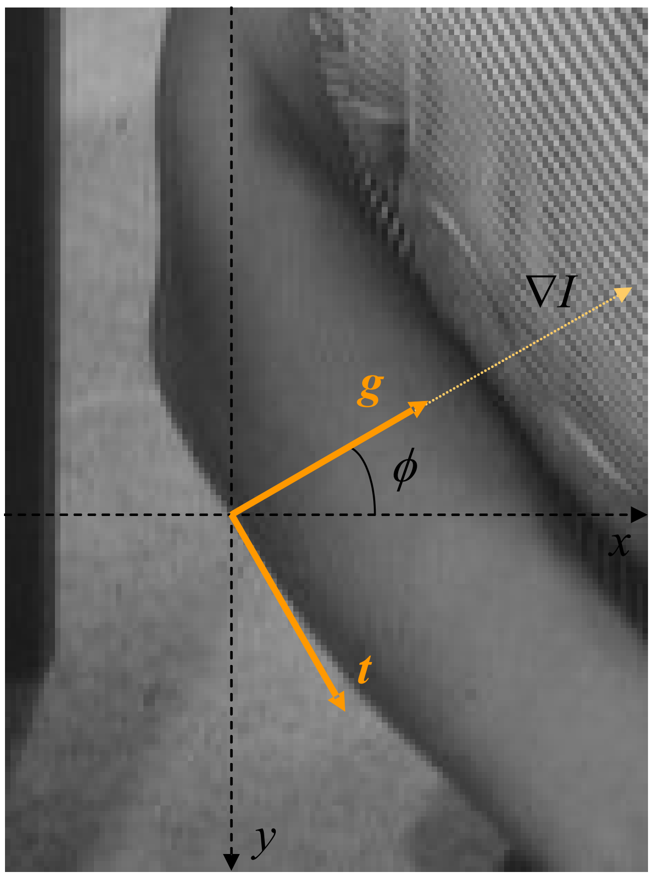


Cumulative histogram
showing quantile aggregation

"Segmented" image

III-4: Differential model

In the differential model, the image is seen as a continuous and et differentiable function $f(x,y)$, whose local behaviour is studied using its partial derivatives.
Such study, founded on Taylor's expansion formula, only makes sense if the function f is *regular* enough, which is the key problem of differential methods.



At the first order, to each pixel (x,y) may be associated a local frame (g,t), where the vector g corresponds to the *gradient* direction (i.e. line of steepest ascent), and t to the *isophote* direction (i.e. line of iso-gray-value).

Thanks to the continuous approximation, this model allows in addition to express a large number of analysis operations in terms of Partial Derivative Equations (PDEs), which provides mathematical / physical foundations to many image processing operators, and also methods to calculate them using numerical resolution schemes.

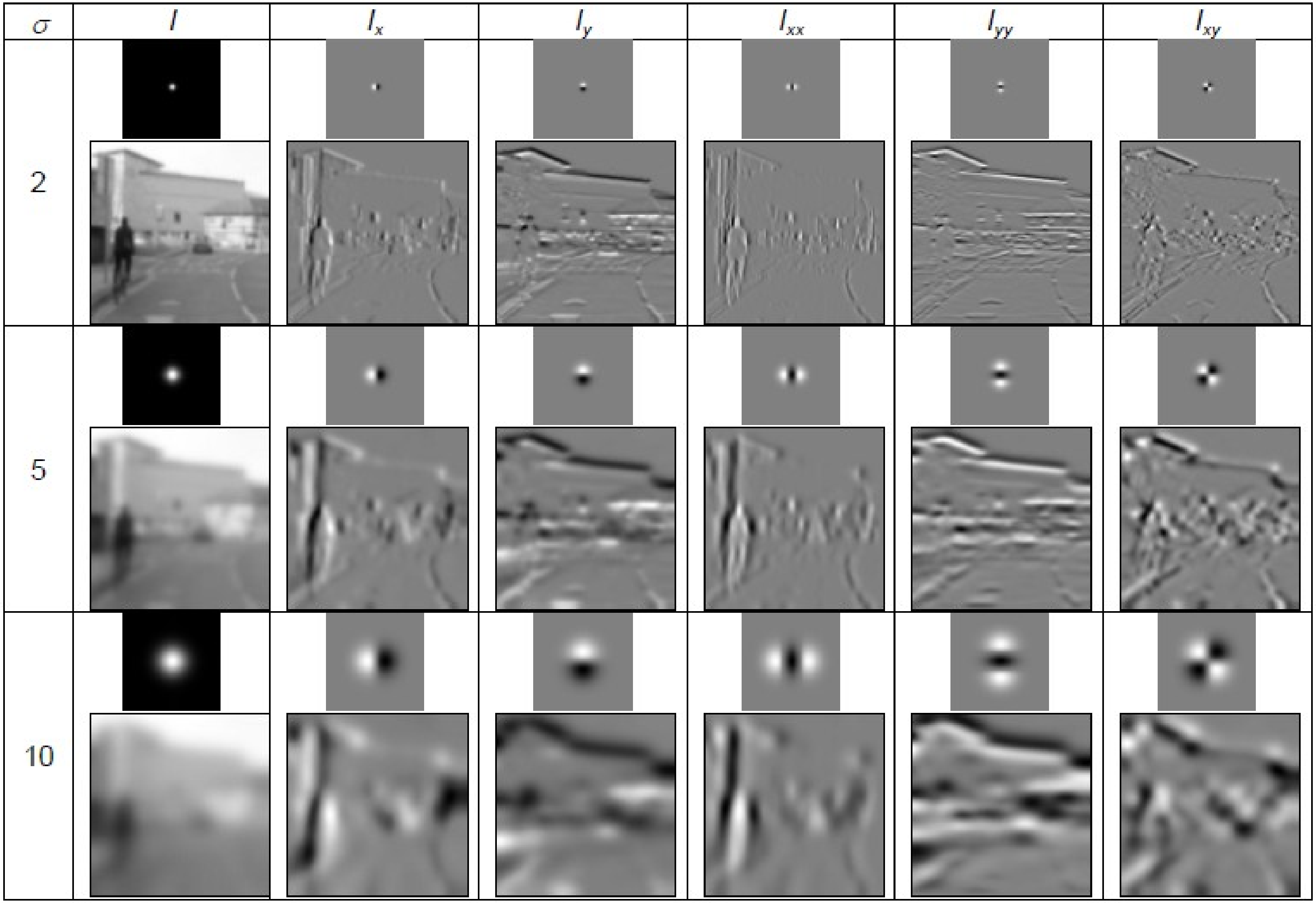
Differential quantities

Analysing local geometry in images: contrast, orientation, curvature... is naturally approached by the differential geometry framework.
Estimating the spatial derivatives then plays a major role in image processing.
Nowadays this estimation is strongly related to the *scale space theory*: a derivative in a physical signal only makes sense *up to a scale factor*.
It is based on the fact that convolution commutes with derivation, and then estimating the derivative at a given scale is done by convolving the image by the derivative of a convolution kernel, where the spatial scope of the kernel corresponds to the scale:

$$\frac{\partial}{\partial u}(f * G_s) = f * \frac{\partial G_s}{\partial u}$$

- s: estimation scale
- G_s: smoothing filter (scale s)
- $\frac{\partial G_s}{\partial u}$: derivative filter (scale s)

Multi-scale derivatives



Derivatives to the order 2, estimated at 3 different scales. The estimation is obtained by convolving the original image (200x200) with the 2d (derivative of) Gaussian, corresponding to the convolution kernel (100x100 icon) displayed above the result image.

Finite differences: order 1

The simplest approximations of spatial derivatives are obtained by finite differences, corresponding to convolutions with small kernels:

E.g.: $[-1\,\mathbf{1}]$, for approximating $\frac{\partial f}{\partial x}$, and : $\begin{bmatrix}-1\\ \mathbf{1}\end{bmatrix}$, for approximating $\frac{\partial f}{\partial y}$

Usually $[-1\,\mathbf{0}\,1]$ and $\begin{bmatrix}-1\\ \mathbf{0}\\ 1\end{bmatrix}$ are preferred, since they produce thicker, but well centred (zero-phased) frontiers.

These operations being very noise sensitive, they are usually combined with a smoothing filter in the direction orthogonal to derivation, for example using the following kernel (or its transpose): $[1\,2\,1]$

Finally the first order spatial derivatives in *x* and *y* may be estimated by convolving the image with the following kernels, respectively:

$$f_x[i,j]=(f*h_x)[i,j]$$
$$f_y[i,j]=(f*h_y)[i,j]$$

, with: $h_x=\begin{bmatrix}-1 & 0 & 1\\ -2 & \mathbf{0} & 2\\ -1 & 0 & 1\end{bmatrix}$ $h_y=\begin{bmatrix}-1 & -2 & -1\\ 0 & \mathbf{0} & 0\\ 1 & 2 & 1\end{bmatrix}$ (Sobel's masks)

Then to compute the norm of the gradient:

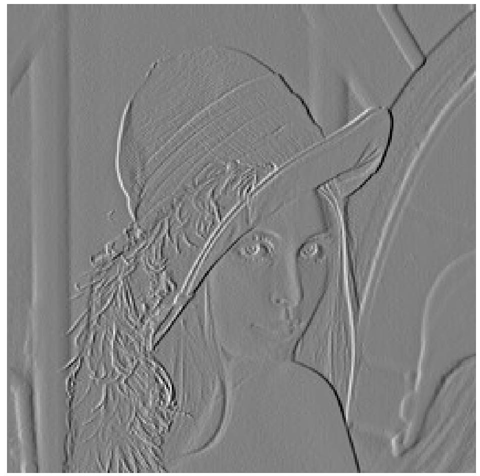
$$\|\nabla f[i,j]\|_2=\sqrt{f_x[i,j]^2+f_y[i,j]^2}$$
$$\|\nabla f[i,j]\|_1=|f_x[i,j]|+|f_y[i,j]|$$
$$\|\nabla f[i,j]\|_\infty=max\{|f_x[i,j]|,|f_y[i,j]|\}$$

...and its orientation:

$$arg(\nabla f[i,j])=\arctan\left(\frac{f_y[i,j]}{f_x[i,j]}\right)$$

Finite differences: order 1

Original



*Kernel [-1 **1**]*

*Kernel [-1 **0** 1]*

Horizontal Sobel

Vertical Sobel

Sobel gradient magnitude

Finite differences: order 2

The simplest finite difference approximation of the second derivative is obtained by the convolution with the following kernel:

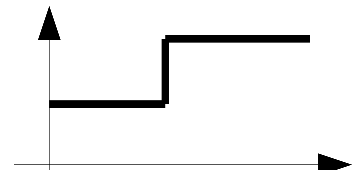
$[\,1\ \mathbf{-2}\ 1\,]$, for approximating $\dfrac{\partial^2 f}{\partial x^2}$, and: $\begin{bmatrix} 1 \\ \mathbf{-2} \\ 1 \end{bmatrix}$, for approximating $\dfrac{\partial^2 f}{\partial y^2}$

The Laplacian $\Delta f = \dfrac{\partial^2 f}{\partial x^2} + \dfrac{\partial^2 f}{\partial y^2}$ can then be approximated by the following linear operator:

$\begin{bmatrix} & 1 & \\ 1 & \mathbf{-4} & 1 \\ & 1 & \end{bmatrix}$ *4-connected Laplacian* , or also $\begin{bmatrix} 1 & 1 & 1 \\ 1 & \mathbf{-8} & 1 \\ 1 & 1 & 1 \end{bmatrix}$ *8-connected Laplacian*

Exercise: Unsharp masking

- Explain why convolution by $[\,1\ \mathbf{-2}\ 1\,]$ approximates $\frac{\partial^2 f}{\partial x^2}$
- Show how it behaves on a simple 1d step, like:



- Explain why the following convolution kernel enhances the local contrast (see lab 1):

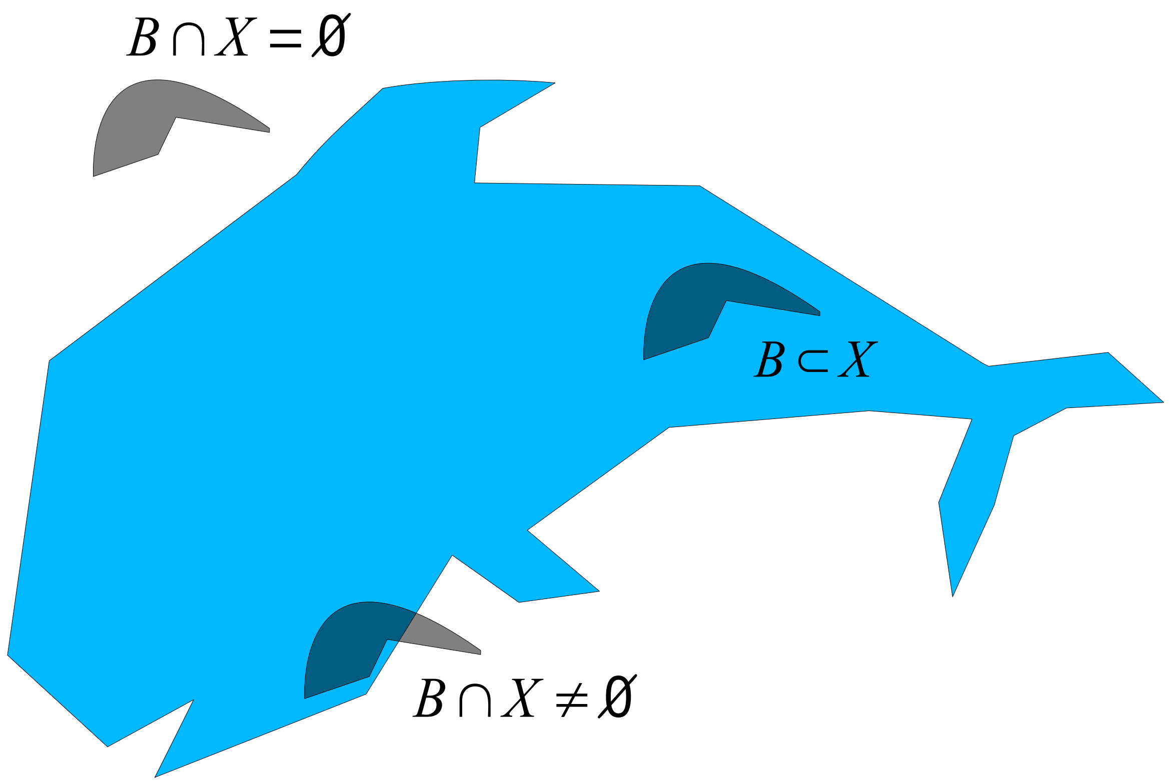
$$\begin{bmatrix} & -1 & \\ -1 & \mathbf{5} & -1 \\ & -1 & \end{bmatrix}$$

f

$f * \begin{bmatrix} & -1 & \\ -1 & \mathbf{5} & -1 \\ & -1 & \end{bmatrix}$

III-5: Set based model

In *mathematical morphology*, the image is seen as a *set*, whose properties are studied with respect to local relations with a reference set (the structuring element) in terms of *intersection* and *inclusion* (hit-or-miss relations).



Erosion and Dilation

Morphological transformations are defined from 2 basic (and dual) set-based operators: *erosion* and *dilation*

Original (Matisse - 1952)

$\varepsilon_B(X) = \{x \in \mathbb{R}^2 \,;\, B_x \subset X\}$

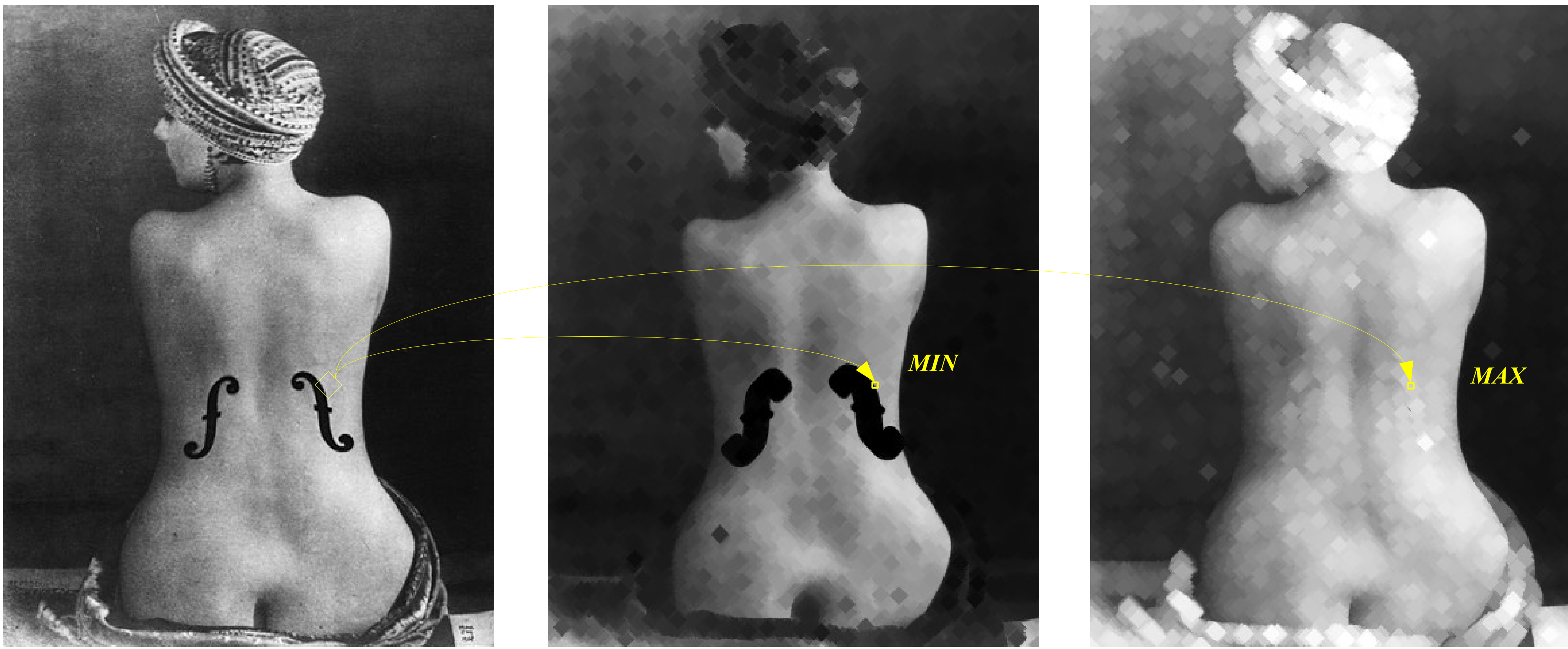
$\delta_B(X) = \{x \in \mathbb{R}^2 \,;\, B_x \cap X \neq \emptyset\}$

(structuring element: disk)

Erosion and Dilation

Erosion and dilation, and then all the morphological transformations, generalise from sets (binary images) to functions (gray level images) through the *level sets*:

$$I_n = \{x \in \mathbb{R}^2 \,;\, I(x) \leqslant n\}$$

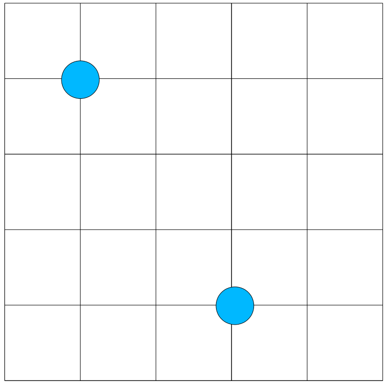


Original (Man Ray - 1924) $\varepsilon_B(I)$ $\delta_B(I)$

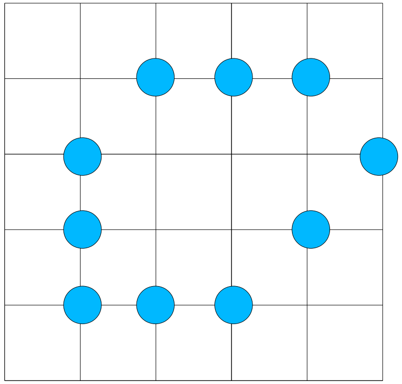
(structuring element: diamond)

III-6: the discrete model

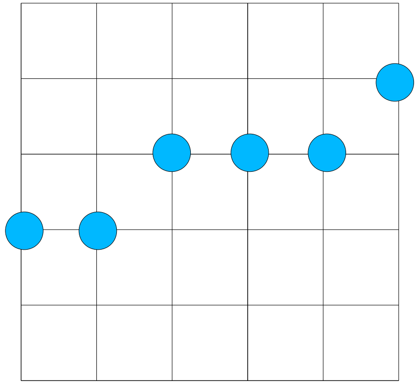
Discrete geometry is at least as old as Image Processing. Whereas the differential model considers the geometric structures (curves, surfaces, lines, etc.) as numerical approximations of their continuous counterparts, or the frequency based model interprets the digitization in terms of information loss, the discrete model, in contrast, integrates the sampled space as its mathematical framework, and aims to provide a rigorous formalism to geometric structures, including definitions, properties, theorems,...



What is the distance between the 2 points?



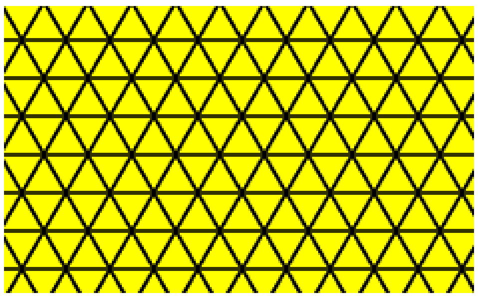
What is a hole?



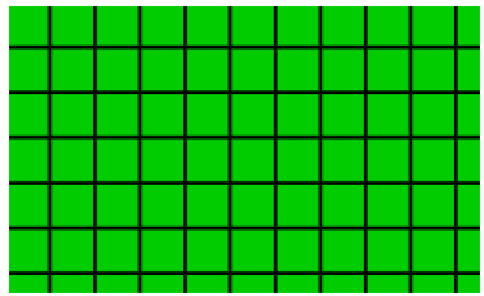
What is a line?

Tessellation of the plane

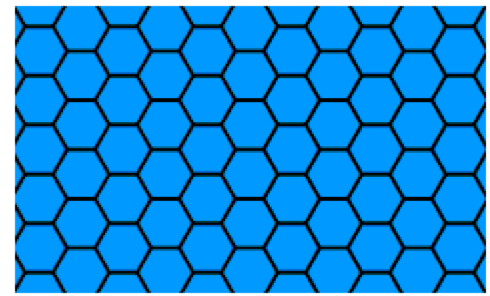
A tessellation (tiling) of the plane is its partition into elementary cells (pixels).
There only exist 3 regular tessellations:



triangular

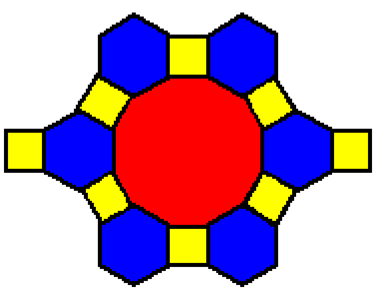


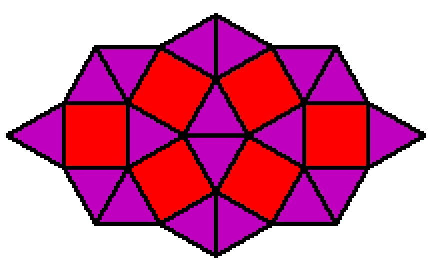
square

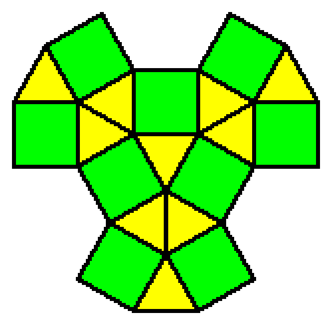


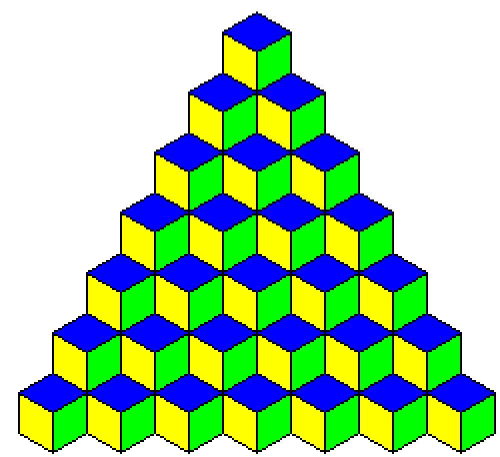
hexagonal

... but many irregular ones:



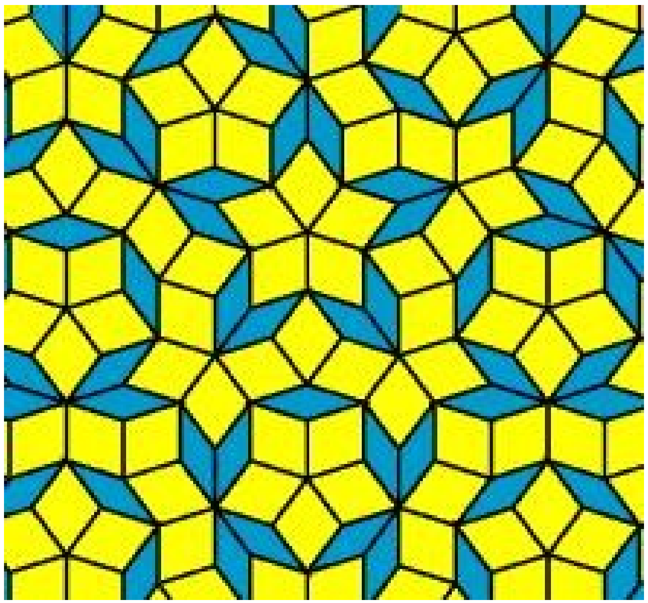




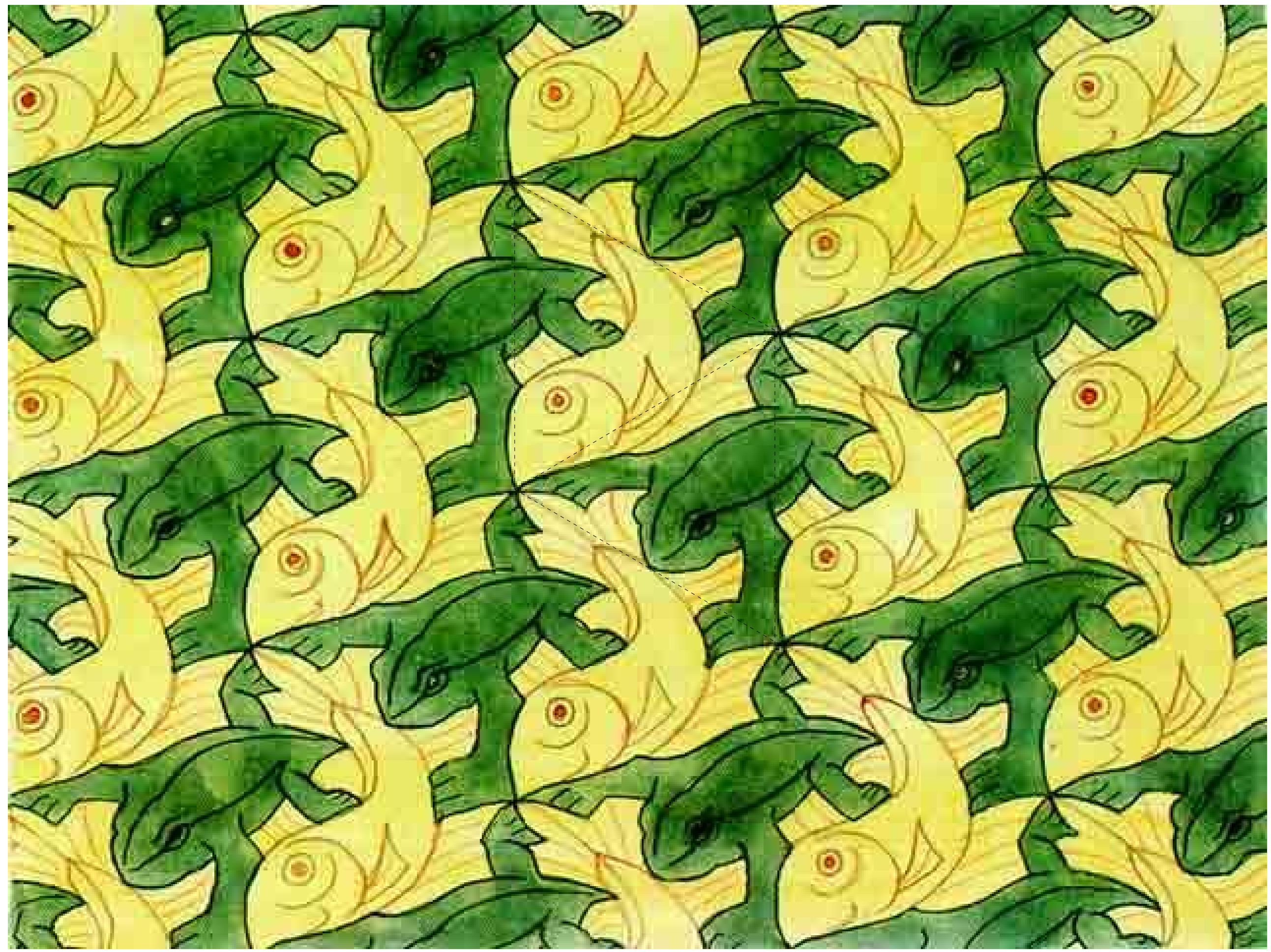


Tessellation of the plane

Other irregular tessellations of the plane...



Penrose's aperiodic tessellation

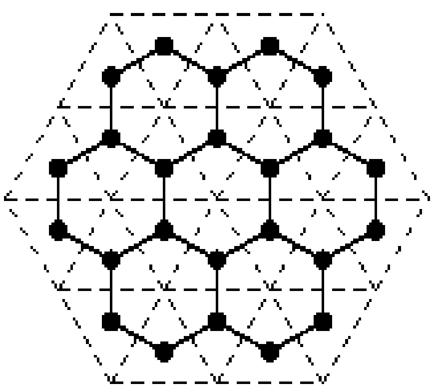


Escher's periodic tessellation

Tessellations and meshes

Every tessellation can be associated to a *graph* whose *vertices* represent elementary cells, and whose *edges* represent the *adjacency* relation between cells (2 cells are adjacent if they share an edge). Such graph is referred to as a *mesh* of the plane.

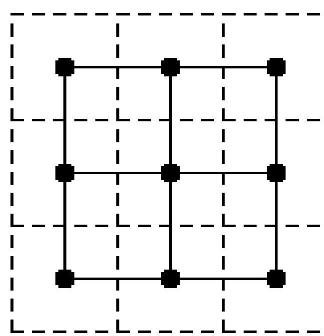
Regular tessellations and meshes are *dual*:



Triangular tessellation

↕

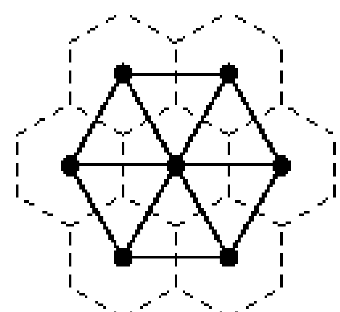
Hexagonal mesh



Square tessellation

↕

Square mesh



Hexagonal tessellation

↕

Triangular mesh

Questions:

- representation in Z^2?
- how many directions?
- recursivity?

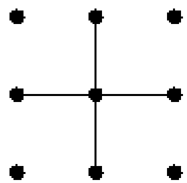
Meshes and connectivity

Topological relations in discrete images are defined from the connectivity relation induced by the mesh graph (X,S), where X is the set of vertices and S the set of edges.

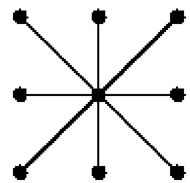
$$X \subset \mathbb{Z}^2 ; S \subset X^2$$

Let x and y be 2 points of X, by definition x and y are neighbours if:

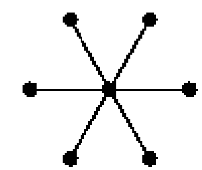
$$x \approx y \Leftrightarrow (x , y) \in S$$



4-connected square mesh



8-connected square mesh



6-connected triang. mesh

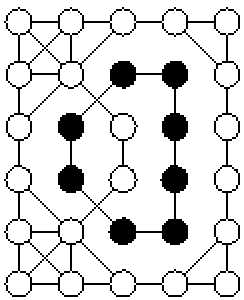
The transitive closure of the neighbourhood (adjacency) relation is an equivalence relation “there exists a connected path between x and y”:

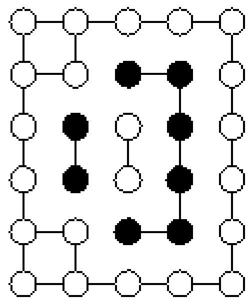
$$x \sim y \Leftrightarrow \exists \{x_{1,} \ldots , x_n\} / x \approx x_{1,} \ldots , x_i \approx x_{i+1} , \ldots , x_n \approx y$$

The equivalence classes of this relation are called the *connected components* of X

Topologies in the square mesh

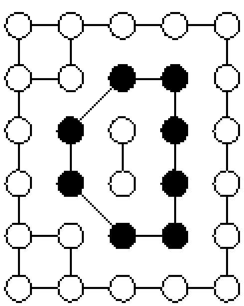
In the square mesh, the notion of *hole* in an object X ($X \subset Z^2$), that should correspond to a finite connected component of the complementary X^c, is not well defined...

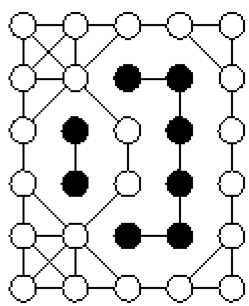

8-connectivity


4-connectivity

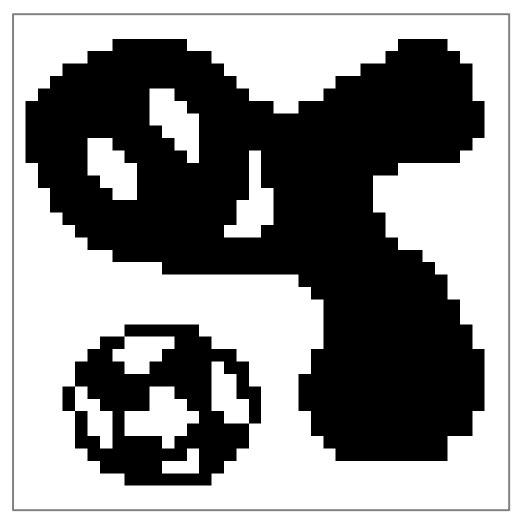
This is related to the validity of *Jordan's theorem*, which stated that a simple closed (Jordan's) curve separates the plane into 2 connected components,one of which is bounded.

...except if *different connectivities* are considered for *X and X^c*:


(8,4)-connectivity


(4,8)-connectivity

Jordan's theorem is valid for these hybrid connectivities.



Questions:

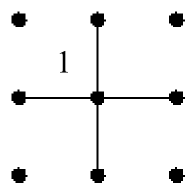
How many connected components, and how many holes are there in the image on the left?
- in (8,4)-connectivity?
- in (4,8)-connectivity?

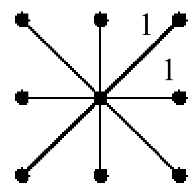
Metrics in the square meshes

The graph of the mesh also induces a metrics in the discrete space, the distance between 2 points *x* and *y* being defined by the length of the shortest path between them. By weighting each edge by 1, we get:

distance of the 4-connectivity

$$d_4(x,y)=|x_1-y_1|+|x_2-y_2|$$

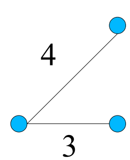




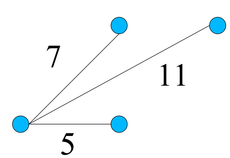
distance of the 8-connectivity

$$d_8(x,y)=max(|x_1-y_1|,|x_2-y_2|)$$

We can also weight differently the edges of the 8-connected mesh, or even use more complex meshes (i.e. larger neighbourhoods):



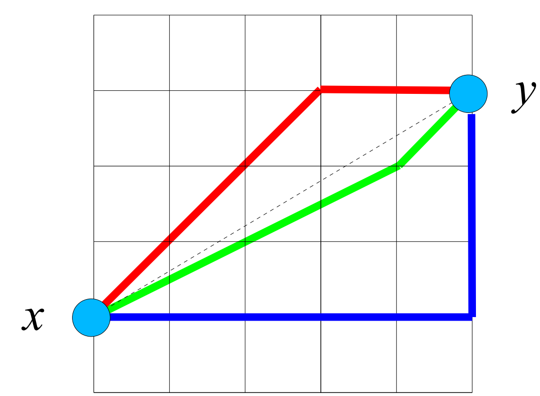
(3,4) chamfer distance



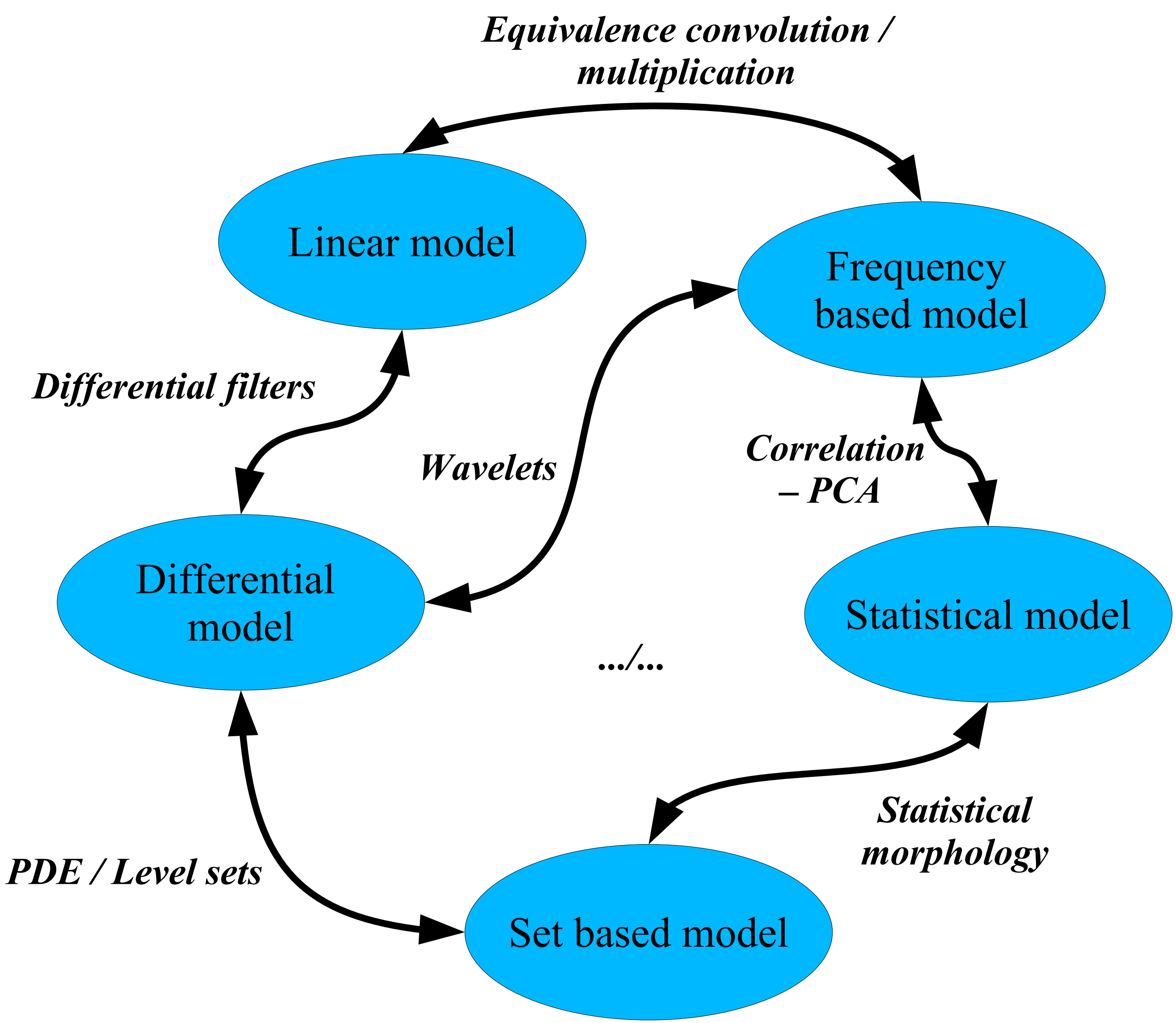
(5,7,11) chamfer distance

Questions:

calculate distances $d_4(x,y)$, $d_8(x,y)$, $d_{ch(3,4)}(x,y)$, $d_{ch(5,7,11)}(x,y)$ between the two 2 points *x* and *y* on the right:



Conclusion



Key notions:

- digital image
- sampling
- quantization
- histogram
- convolution
- frequency representations
- estimation of derivatives
- discrete connectivity and distance

Sources and bibliography

• R.C. Gonzalez et Woods “*Digital Image Processing - 2d edition”* Addison Wesley 2002.

• A. Rosenfeld et A.C. Kak “*Digital picture processing”* Academic Press - London 1982.

• J.R. Parker “*Algorithms for Image Processing and Computer Vision”* Wiley & Sons 1997.

• I.T. Young, J.J. Gerbrands et L.J. Van Vliet “*Fundamentals of Image Processing”* University of Delft.

• B.M. ter Haar Romeny “Front-End Vision and Multi-Scale Image Analysis” Springer 2003.

• R. Szeliski “Computer Vision: Algorithms and Applications” Springer 2010

• S.J.D. Prince “Computer vision: models, learning and inference” Cambridge University Press 2012

Course description, program, practical details, and slides available on:

https://perso.ensta-paris.fr/~manzaner/Cours/Masters_ParisSaclay/Image_Mining/