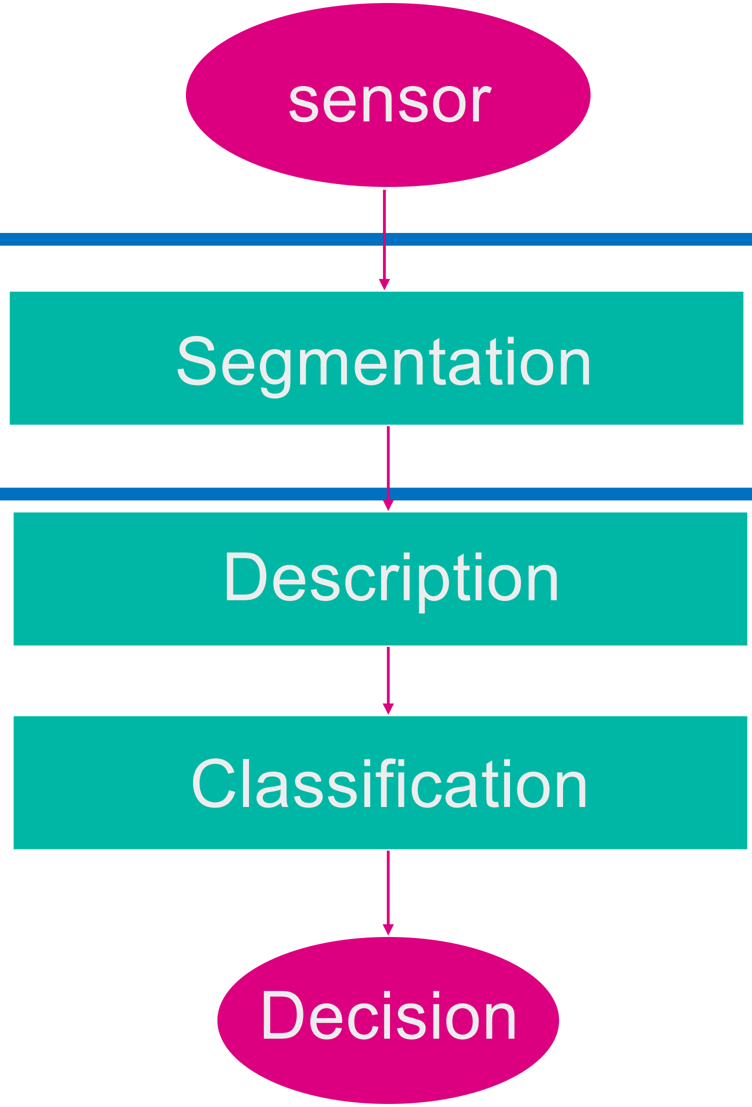
Image analysis and Pattern Recognition

Lecture 1 : image pre-processing

Prof. Jean-Philippe THIRAN
JP.Thiran@epfl.ch

The Pattern Recognition System



- Electricity, electronics, metrology
- Signal Processing
- Image Processing
- Geometry, topology, …
- Statistics
  - AI
- Robotics, …

- Chapter 1 : Image Segmentation
  - Digital images, properties & consequences of those properties
  - Pre-processing
    - *Histogram Equalization*
    - *Denoising and image restoration*
  - Segmentation:
    - *Contour-based approach*
    - *Region-based approach*
    - *Mathematical Morphology*
    - *(Hough Transform)*
  - Some examples

Lecture 1 (Digital images through Denoising and image restoration)

Lecture 2 (Segmentation through Some examples)

Digital images & properties

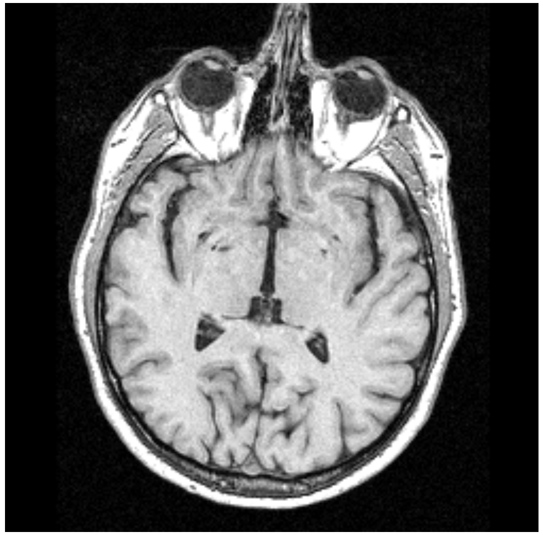
- An image is a function of (at least) 2 spatial variables :

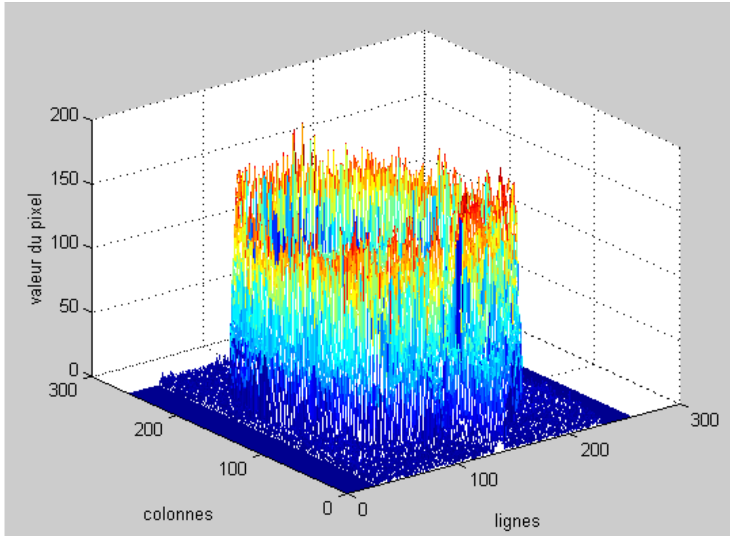
$$f : \Re^m \rightarrow \Re^n, \vec{x} \mapsto \vec{y} = f(\vec{x})$$

  - 2D Images: $m = 2, \quad \text{i.e.} \quad \vec{x} = (x, y)$
  - Monochromatic images: $n=1$
  - Digital images: discrete domain:

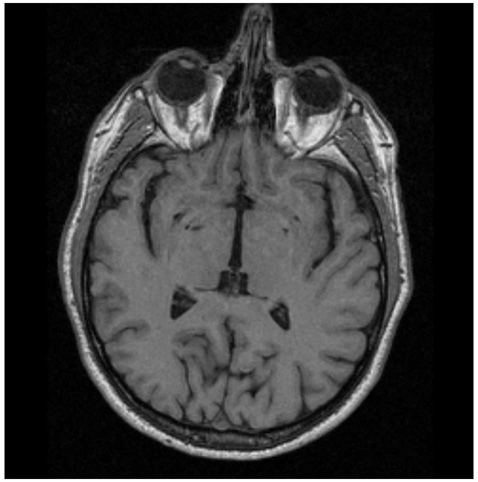
$$f : \mathbb{N}^m \rightarrow \Re^n$$

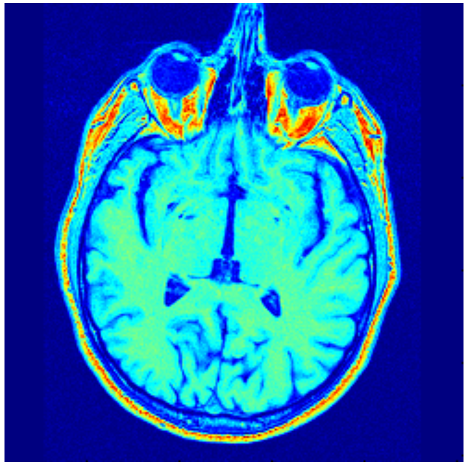
    - *The points are pixels (voxels in 3D)*





- Monochromatic images: 1 scalar / pixel
  - Often displayed as grey levels
  - Choice of other color lookup tables (CLUT) : « false colors »





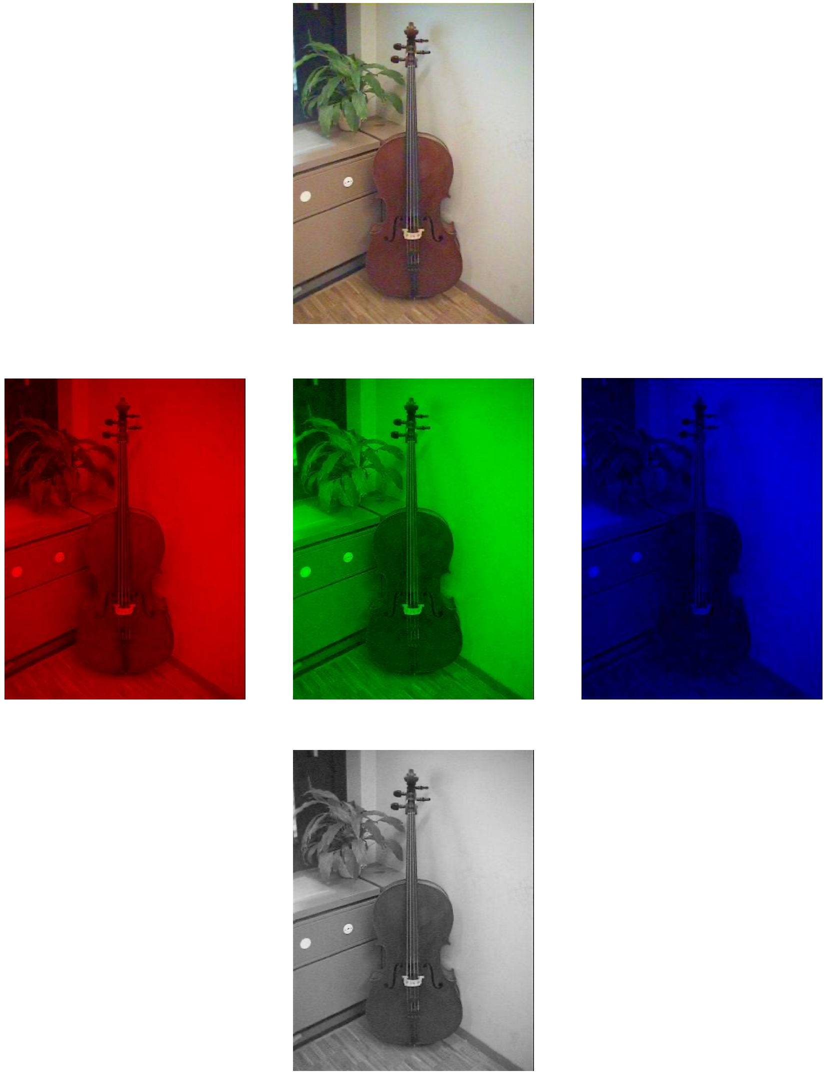
Valeur	R	G	B
0	0/255	0/255	0/255
1	1/255	1/255	1/255
2	2/255	2/255	2/255
3	3/255	3/255	3/255
…	…	…	…
255	255/255	255/255	255/255

Valeur	R	G	B
0	0	0	0.5
1	0	0	052
…	…	…	…
150	1	0.34	0
…	…	…	…
255	0.5	0	0

- Multi-spectral images:
  1 vector/pixel
  - Example : color images
    - *Each pixel has three color components, in a given color space*
    - *Example : display on a screen: RGB*
    - *Example : TV diffusion: YUV ou YIU (Y=luminance)*

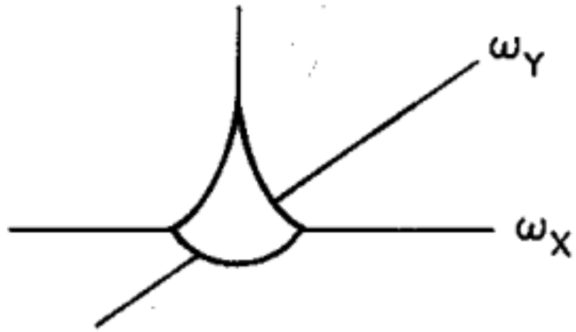
$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.331 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

  - Luminance is a good component to identify objects



- Let $f(x,y)$ *be a 2D image*
  - *Its Fourier Transform is*

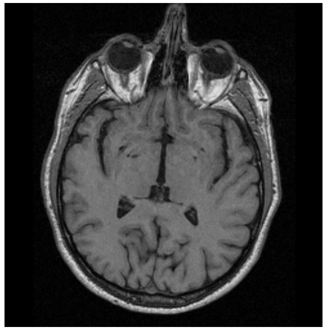
$$F(f_x, f_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi(f_x x + f_y y)} dxdy$$

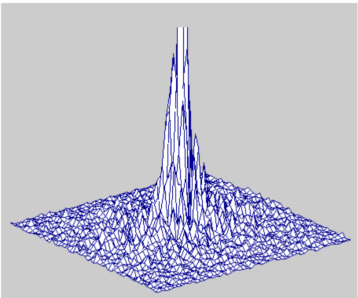


  - *It is separable :*

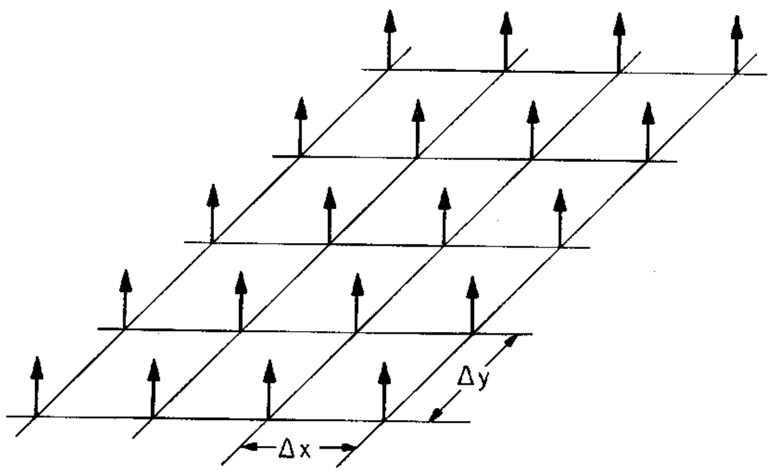
$$F_y(f_x, y) = \int_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi f_x x} dx$$

$$F(f_x, f_y) = \int_{-\infty}^{\infty} F_y(f_x, y)\, e^{-j2\pi f_y y} dy$$





- Sampling a continuous function (image) *f(x,y)* means taking samples at every Δx and Δy
- f_{ex} and f_{ey} are the vertical and horizontal sampling frequencies, respectively
- Mathematically, this means multiplying the analog image f(x,y) by a grid of Dirac impulses



- Thus, the spectrum of the sampled image can be obtained by a convolution of the spectrum of the analog image with the FT of the grid of Dirac impulses:

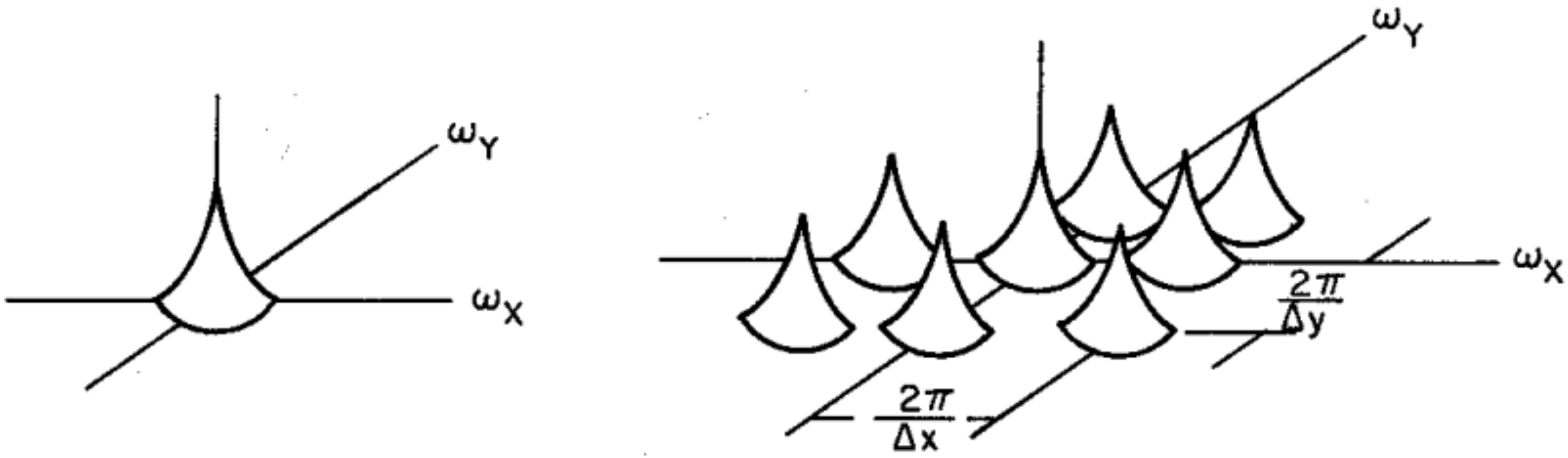
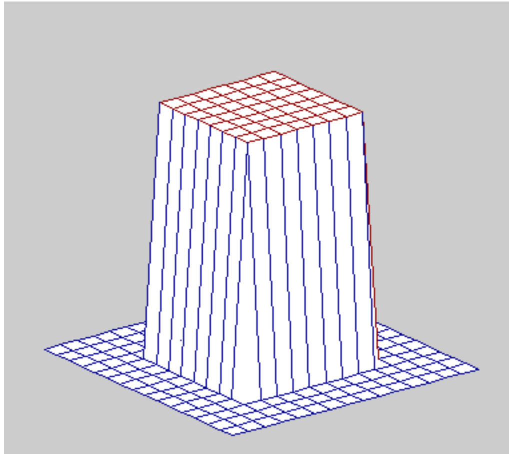
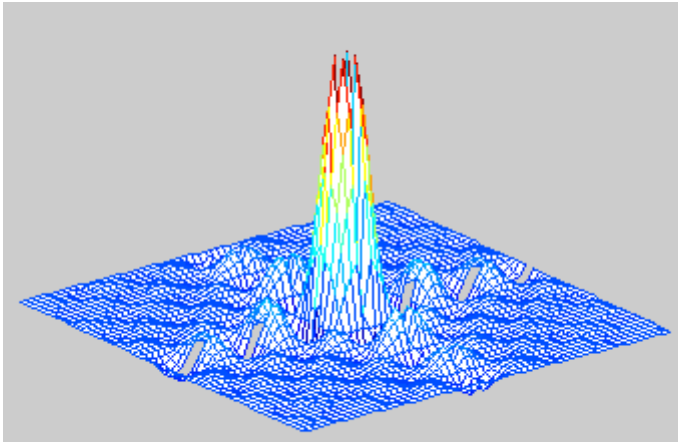


Image reconstruction

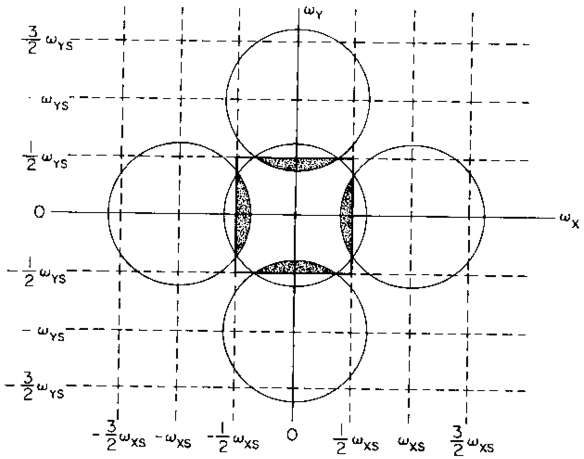
- From the spectrum of the sampled image, on can try to reconstruct the original image by low pass filtering:
  - Multiply the spectrum by a rectangle function in 2D
  - Convolve the sampled image by a signal of type *sin(x)/x* in 2D

$$R(x,y)=\frac{K\omega_{xL}\omega_{yL}}{\pi^2}\frac{\sin(\omega_{xL}x)}{\omega_{xL}x}\frac{\sin(\omega_{yxL}y)}{\omega_{yxL}y}$$





- Practical implication: scale change (« zoom »)
  - Up-sampling : interpolation by $R(x,y)$
  - Down-sampling: low-pass filtering + interpolation, otherwise aliasing



Aliasing

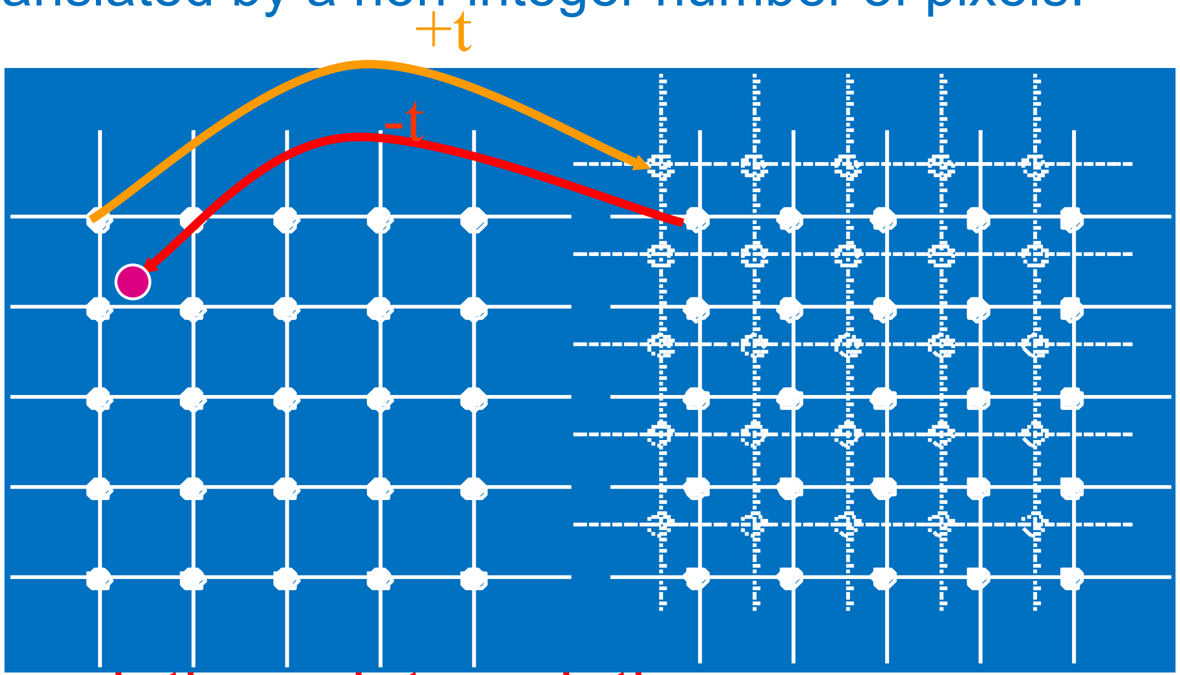

Originale Image


Aliasing

- Following the same considerations, it is possible to perform geometrical operations on images, as a pre-processing step:
  - translation : thanks to interpolation, it is possible to reconstruct an image translated by a non-integer number of pixels:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

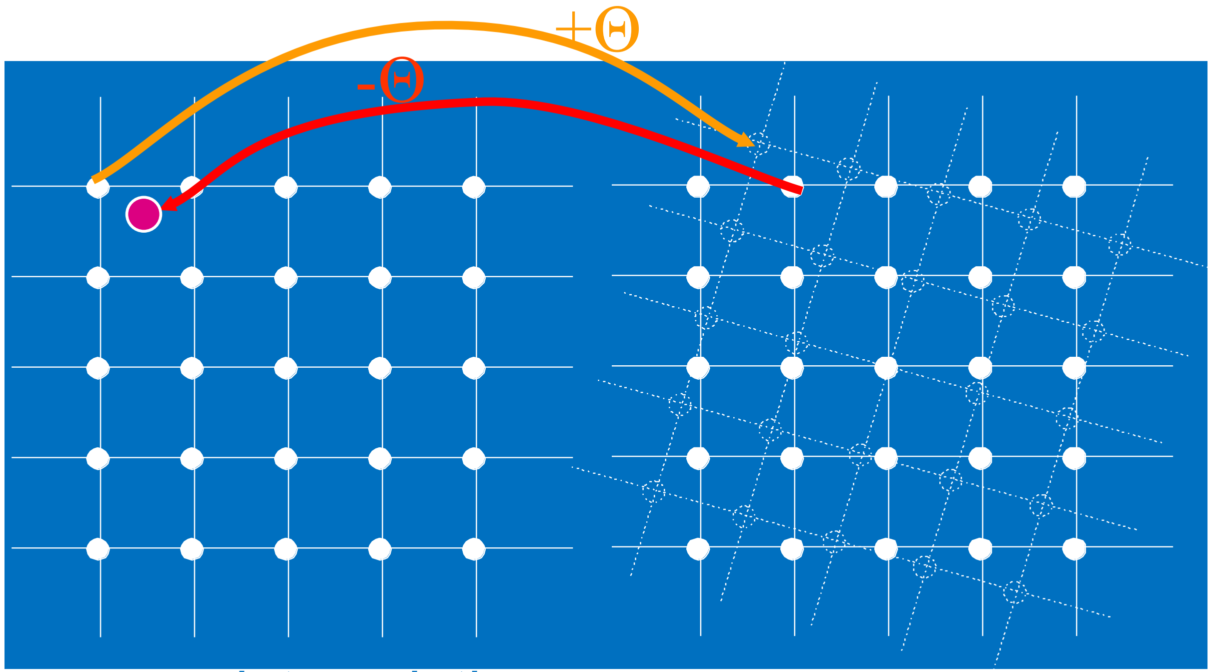


  - Inverse translation + interpolation

- Other operations of the same type:
  - rotation : idem

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



    - *rotation inverse + interpolation*
  - And similarly for all the possible geometrical transformations

- More generally:
  - Linear transformations :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_0 \\ b_1 & b_2 & b_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

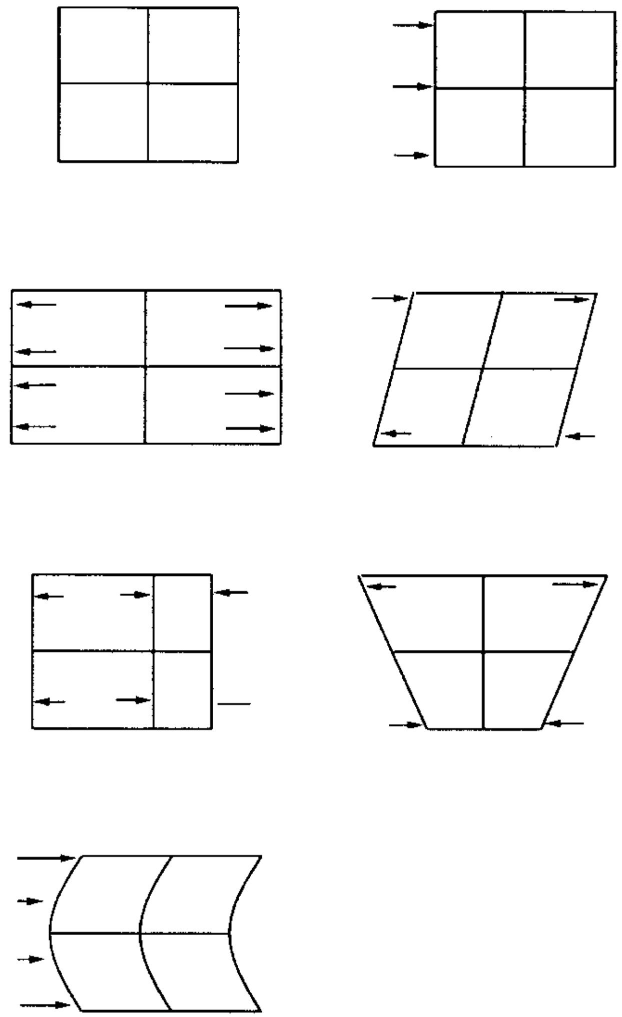
$$\text{Translation}: \mathrm{a}_0 = -t_x, a_1 = 1, a_2 = 0, b_0 = -t_y, b_1 = 0, b_2 = 1$$

$$\text{Rotation}: \mathrm{a}_0 = 0, a_1 = \cos\theta, a_2 = \sin\theta, b_0 = 0, b_1 = -\sin\theta, b_2 = \cos\theta$$

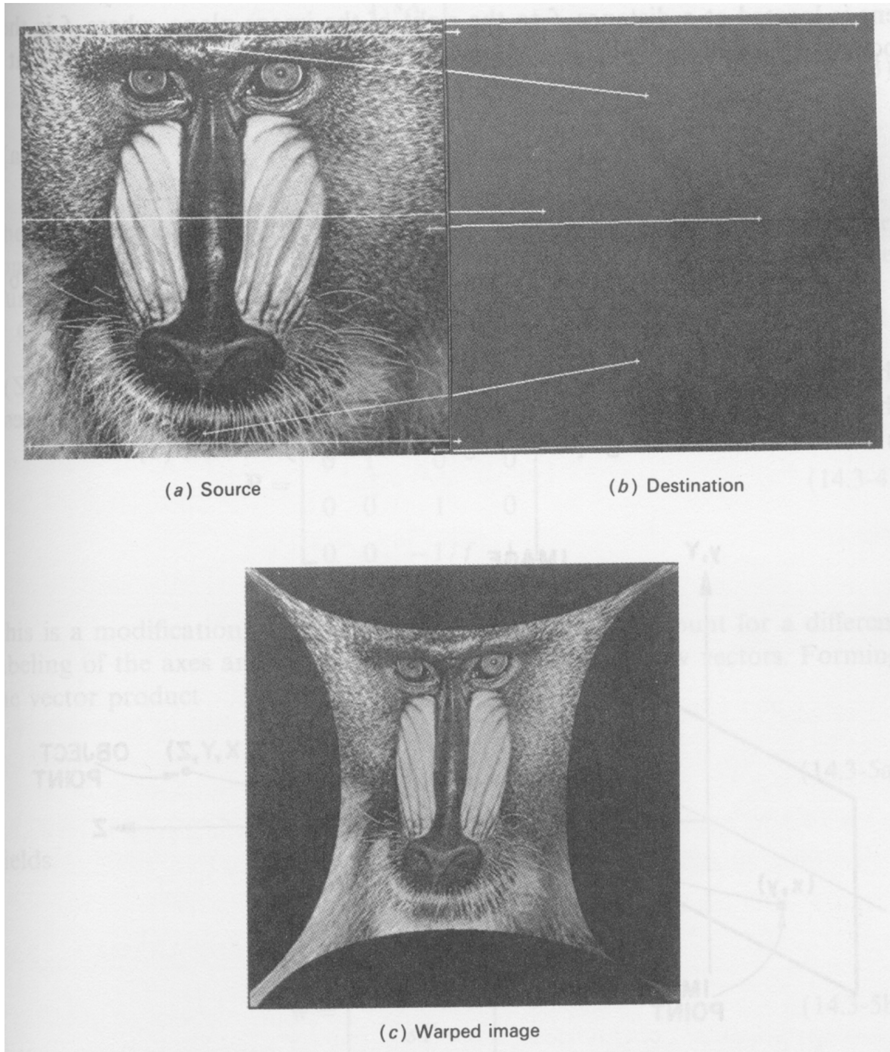
$$\text{Zoom}: \mathrm{a}_0 = 0, a_1 = 1/s, a_2 = 0, b_0 = 0, b_1 = 0, b_2 = 1/s$$

- Other geometrical transformations (non-linear):
  - Polynomial deformation
    - *Ex: order 2*

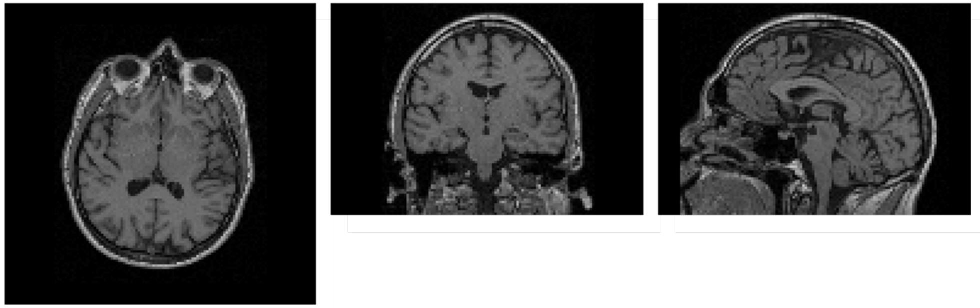
$$u = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2$$
$$v = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy + b_5 y^2$$



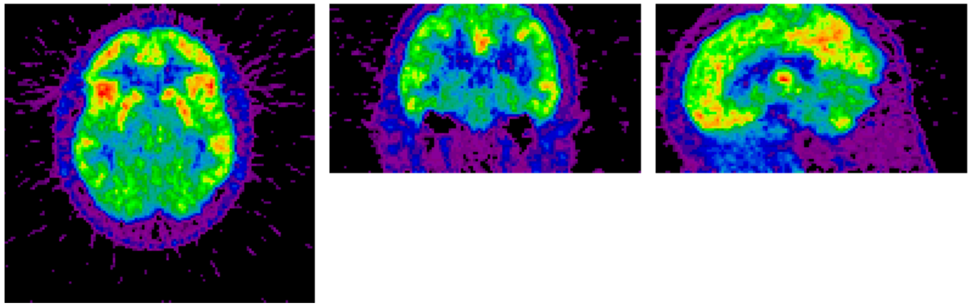
- Example of polynomial transformation



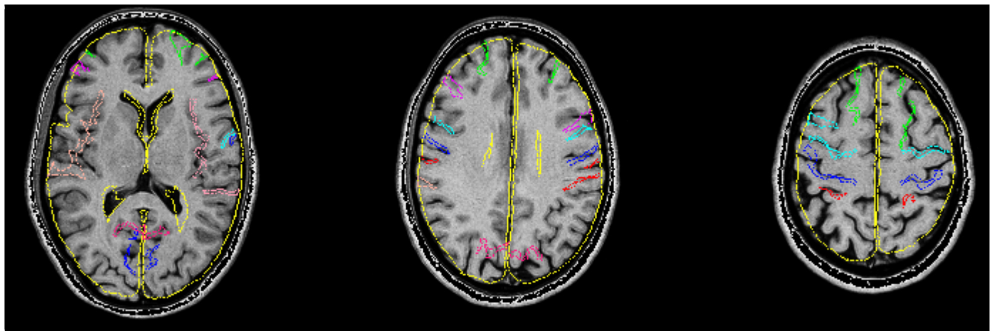
- Examples in medical imaging
  - compensation of the difference in position between two patients
    - *rotation-translation*



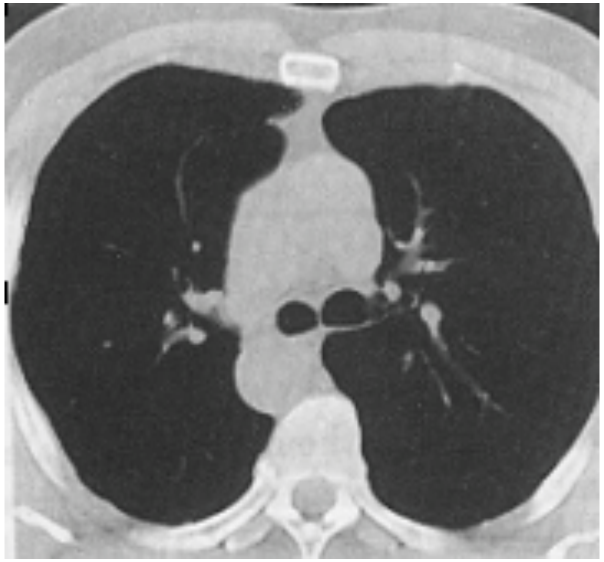
  - Registration in functional MRI
    - *rotation-translation*

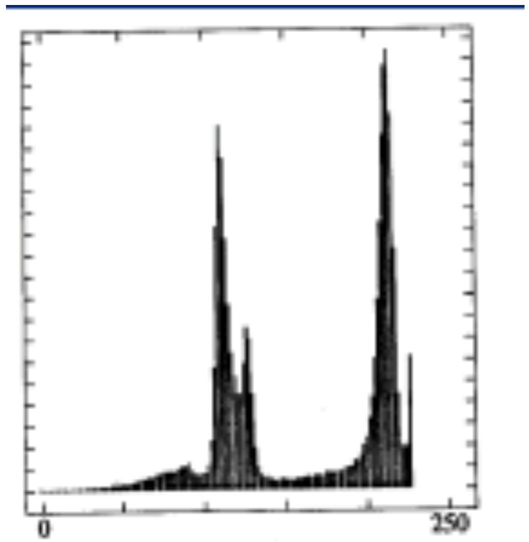


  - Registration between different patients
    - *Complex non-rigid registration (polynomial)*



- Let us observe the histogram of the CT image

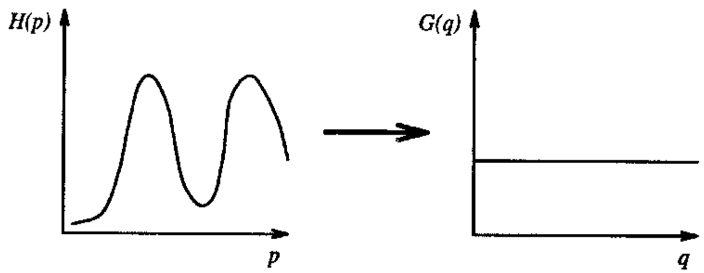




Histogram Equalization

- Goal: create an image with a uniform histogram, by a transformation $q=T(p)$

$$\sum_i G(q_i) = \sum_i H(p_i) \text{ et } G(q_i) = \frac{N^2}{q_k - q_0}$$

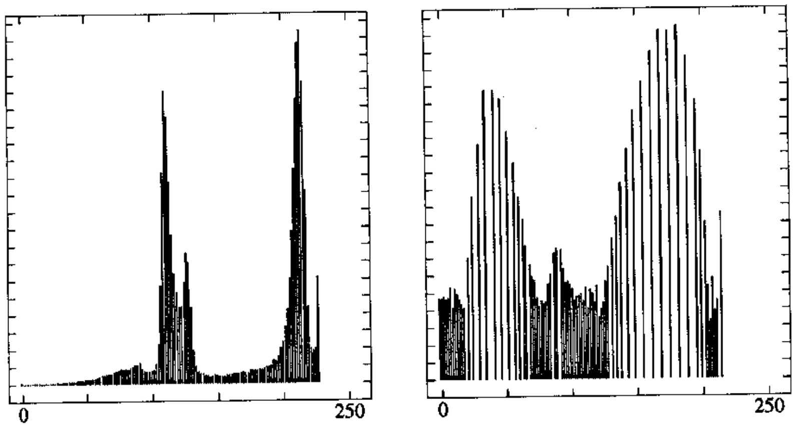


$$\int_{q_0}^{q} \frac{N^2}{q_k - q_0} ds = \frac{N^2}{q_k - q_0}(q - q_0) = \int_{p_0}^{p} H(s) ds$$

$$q = T(p) = \frac{q_k - q_0}{N^2} \int_{p_0}^{p} H(s) ds + q_0$$

$$q = T(p) = \frac{q_k - q_0}{N^2} \sum_{i=p_0}^{p} H(i) + q_0$$

- Example



- Vast topic!
- Source of noise in images:
  - All the interferences on the measurements (electrical, mechanical, …)
  - Signal quantification
  - etc.
- Characteristics of the noise
  - Random signal
  - Often, in real situations, noise can be described as white, additive and Gaussian
  - But it may have sometimes other statistics (Reyleigh)
  - additive : $y(t) = x(t)+n(t)$

- One can define the importance of the noise by the signal to noise ratio:

$$SNR = 10\log\frac{P_S}{P_N} \qquad [\mathrm{dB}]$$

- If one can acquire several realization of the noisy signal, one can try to denoise it by exploiting the statistical properties (e.g. zero mean) of the noise.
  - Example : for an additive noise with zero mean, one can calculate the mean of the observations, which will reduce the noise:

$$\text{If } y(t) = x(t) + n(t)$$

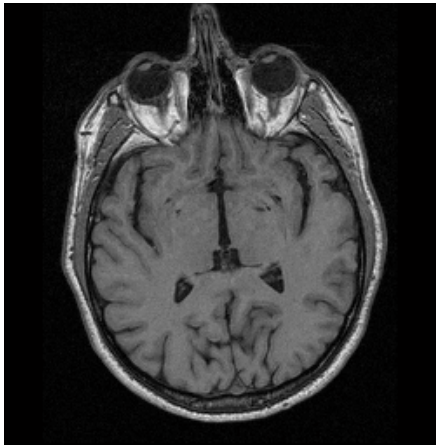
$$\text{Then } E\left(y(t)\right) = E\left(x(t) + n(t)\right) = E\left(x(t)\right) + E\left(n(t)\right) = E\left(x(t)\right)$$

Denoising: linear filtering

- If one has access to only one image: we have to use more general aspects:
  - Noise has a large frequency spectrum, containing also high frequencies (more than the images)
  - A low-pass filtering should reduce the noise
  - But there is a risk to alter the image as well!
- Low-pass filtering:
  - 2D convolution: $G(i,j)=\sum_{m}\sum_{n}F(m,n)H(m-i,n-j)$
  - Some simple low-pass filters:

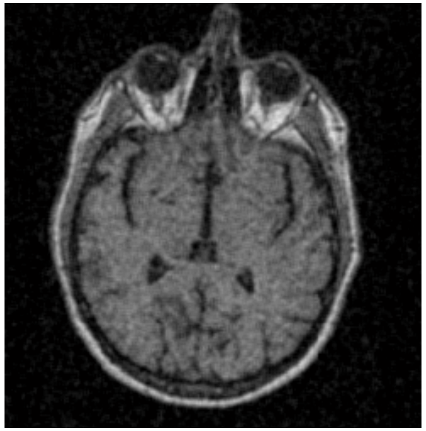
$$H=\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\qquad H=\frac{1}{10}\begin{bmatrix}1&1&1\\1&2&1\\1&1&1\end{bmatrix}\qquad \mathrm{H}=\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

- Example :

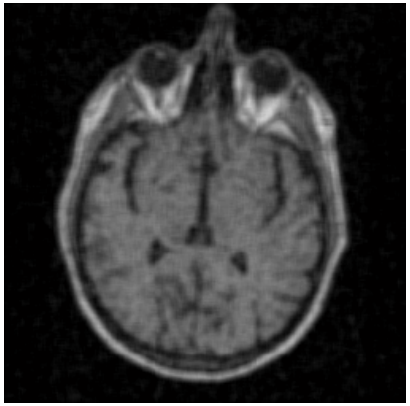


Original

With additive white Gaussian noise



Filtered (3x3 filter)



Filtered (5x5 filter)

- Sometimes the noise can be multiplicative
  - Noise depends on the intensity of the signal
  - example : nuclear medicine
- Then the model of the observed image is:
  - Initial image $f_i(i,j)$ multiplied by the noise $n(i,j)$

$$f_o(i,j) = f_i(i,j)n(i,j)$$

  - By taking the logarithm, we can come back to an additive model, that we can filter

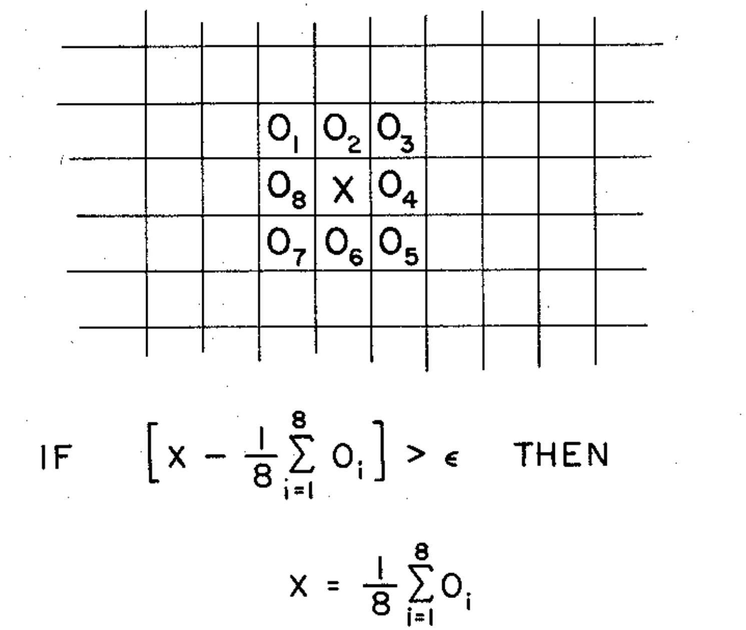
$$\log(f_o(i,j)) = \log(f_i(i,j)) + \log(n(i,j))$$

- Linear filtering works quite well for additive Gaussian noise
  - Even if it degrades the contours
- When the noise is of type “impulses”, we would need a very strong filter to suppress it, and the image would be very degraded
- Non-linear techniques often offer a good compromise between filtering power and respect of the image details

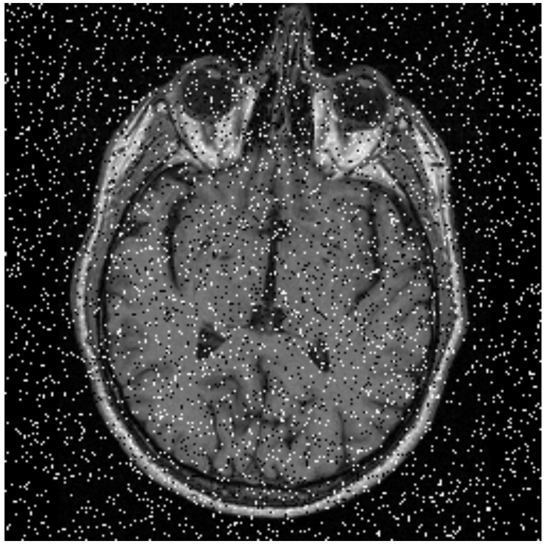
Denoising : « Outliers »

- Compare the pixel value with the mean of its neighbors
  - If the difference is greater than a certain threshold, the pixel is considered as noisy, and replaced by the mean of the neighbors
  - Can be seen as a conditional convolution by

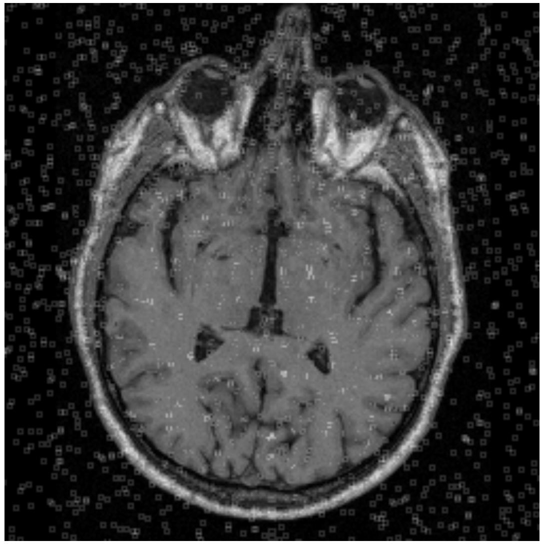
$$H = \frac{1}{8}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



- Example



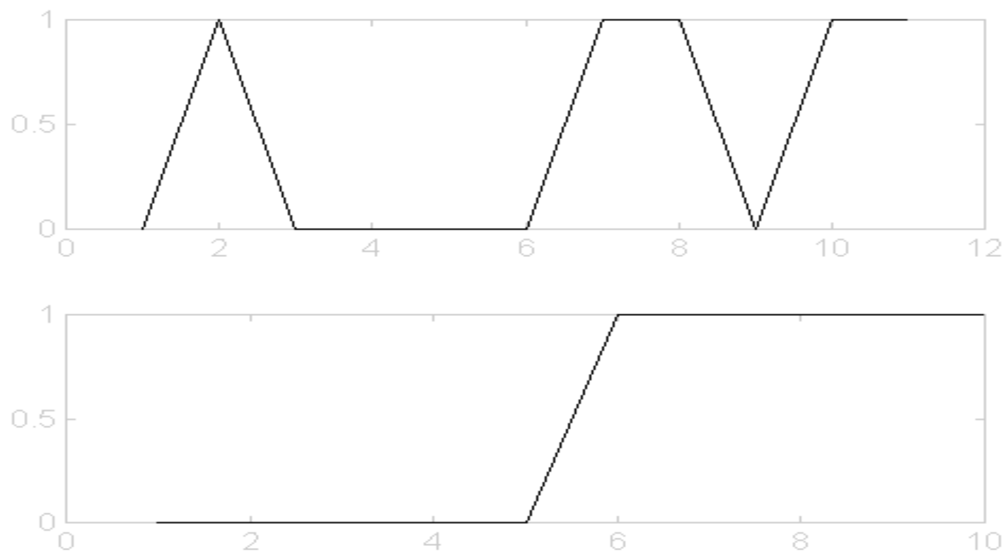
Noisy image
« salt & pepper »



Denoised image

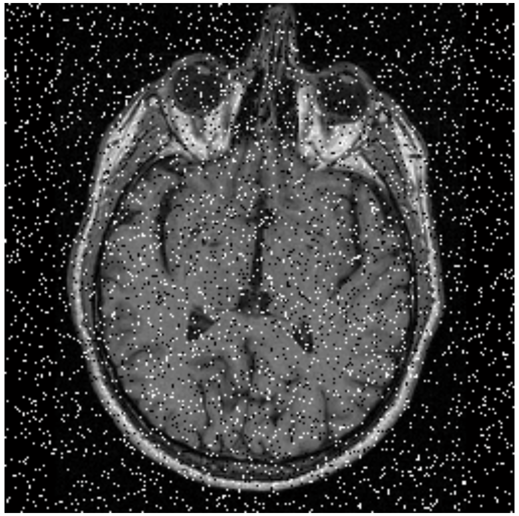
Denoising: median filter

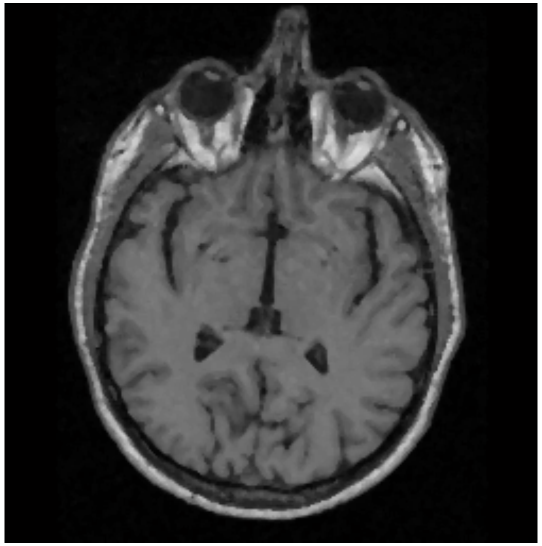
- Let us consider the neighbors of a pixel, on a neighborhood of size *nxn*
- Let us sort the values of those pixels in ascending order
- Let us set the median value of this list (not the mean) as the value of the current pixel
  - The median is the value of the middle element of the list
- Advantages :
  - Suppresses the small variations
  - Keeps the contours



Denoising: median filter

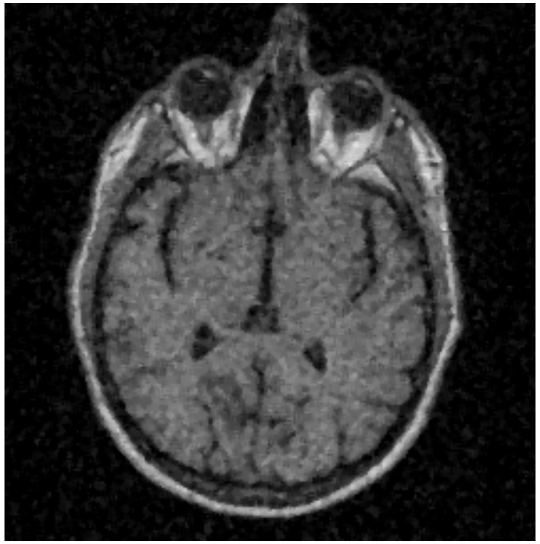
Noisy image
« salt & pepper »





Denoised image
« salt & pepper »

Noisy image
Gaussian noise

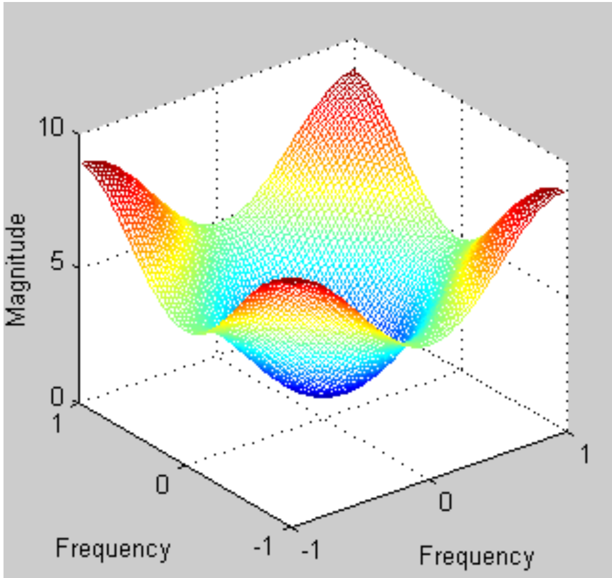


Denoised image
Gaussian noise

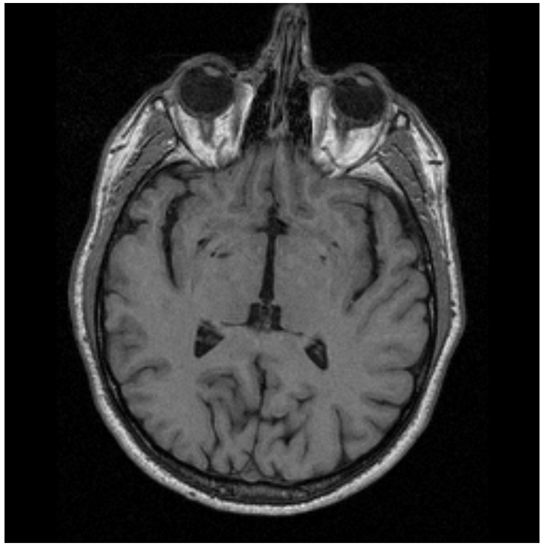
- General Principle:
  - Enhance the high frequencies
    - *Filtering by a high pass filter, added to the original image*
- Some examples of high pass filters:

$$H_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \mathrm{H}_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$



- Example : MRI



Original

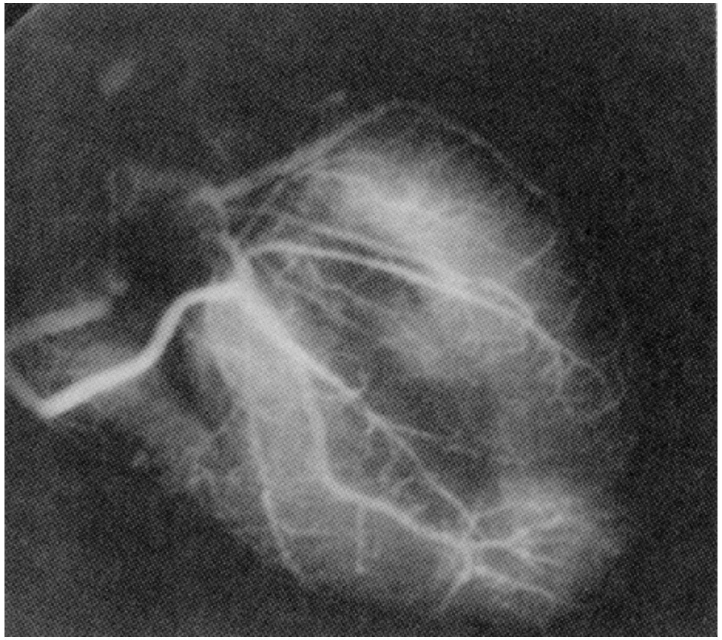
Enhanced : filter H_1

- Example : X ray

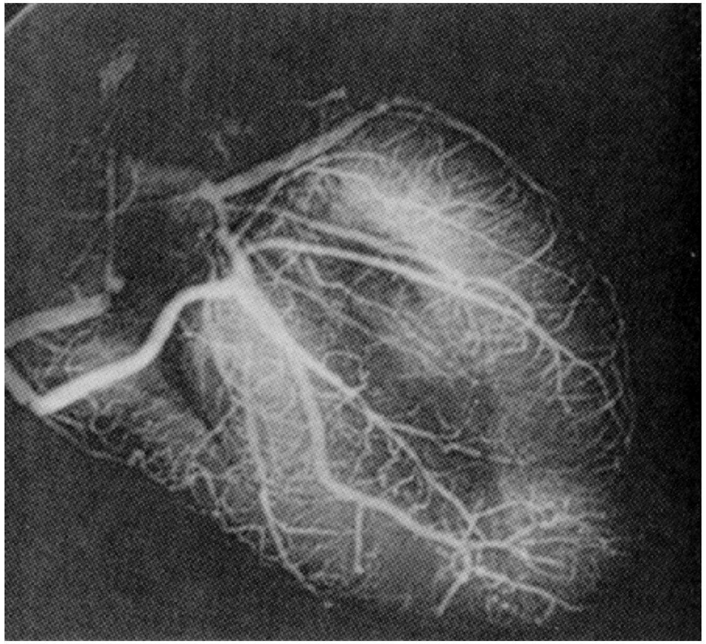
Original

Enhanced : filter H_1

- Example : DSA



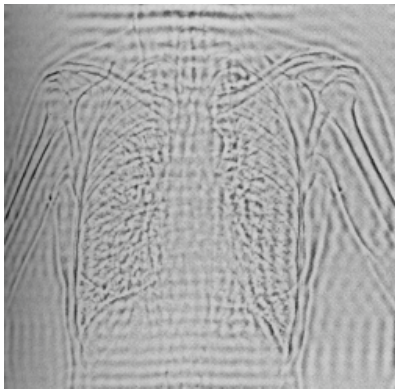
Original



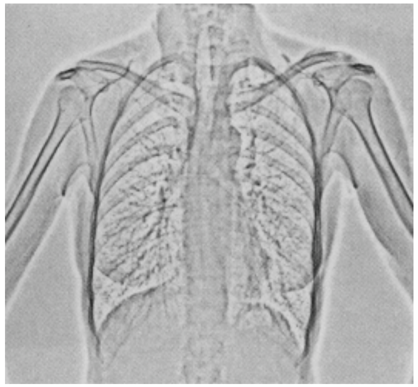
Enhanced : filter H_1

Contour enhancement

- Contour Enhancement in the Fourier domain
  - Filtering by a high pass filter
  - Keep the original image and add the high pass component
  - Warning: do not use too strict filters, because of the ringing effect
  - Prefer a smooth filter like Butterworth



Too strict filter



Butterworth

- Restoration : invert non-wanted effects
- Typical application: deconvolution
  - Let us consider the ideal image f_i that has been degraded by an undesired (low pass) filtering effect
  - Let f_o be the observed image
  - Moreover, there is an additive noise n,

$$f_o(x,y) = f_i(x,y) ** h_D(x,y) + n(x,y)$$

- Goal : try to restore the initial image, using a model for the original image and for the noise

Image restoration: inverse filtering

- Inverse filtering: let us find a filter h_R that will best restore the image f_i
- The restored image will thus be

$$\hat{f}_i(x,y) = f_o(x,y) ** h_R(x,y)$$

- By substitution in the previous equation, we get

$$\hat{f}_i(x,y) = [f_i(x,y) ** h_D(x,y) + n(x,y)] ** h_R(x,y)$$

- By FT:

$$\hat{F}_i(\omega_x,\omega_y) = [F_i(\omega_x,\omega_y) H_D(\omega_x,\omega_y) + N(\omega_x,\omega_y)] H_R(\omega_x,\omega_y)$$

- Thus,, the solution consists in taking a filter h_R with a frequency response inverse of that of h_D :

$$H_R(\omega_x,\omega_y) = \frac{1}{H_D(\omega_x,\omega_y)}$$

- The spectrum of the restored image is thus

$$\hat{F}_i(\omega_x,\omega_y) = F_i(\omega_x,\omega_y) + \frac{N(\omega_x,\omega_y)}{H_D(\omega_x,\omega_y)}$$

- And by inverse FT, the restored image will be

$$\hat{f}_i(x,y) = f_i(x,y) + \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \frac{N(\omega_x,\omega_y)}{H_D(\omega_x,\omega_y)} e^{j(\omega_x x+\omega_y y)} dxdy$$

- Without noise, the restoration is perfect
- With noise, the error can be important:
  - Often h_D will be a low-pass filter (blur, …)
  - Noise will thus be amplified

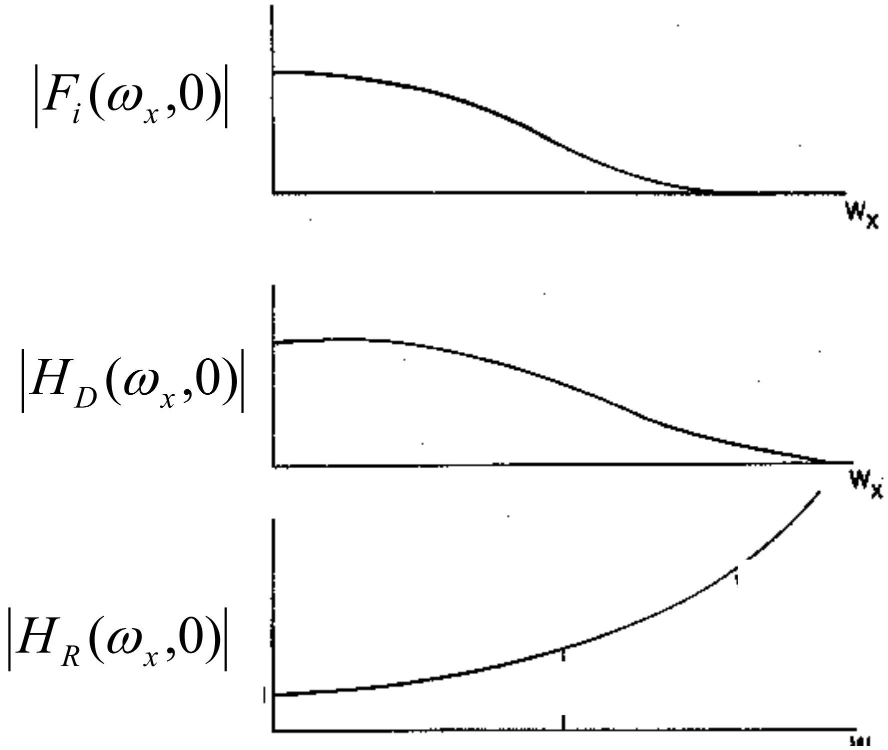


Image restoration: inverse filtering

- Example

Original

Blured image (filtered)

Noisy and blured

- Example (cont.)

Restoration of the blured image

Restoration of the noisy blured image

Image restoration: Wiener filtering

- The previous problem comes from the fact that the filter ignores the presence of noise in the
  - solution : Wiener filtering, that considers both a model of the image and of the noise
- Wiener filtering: hypotheses :
  - Images are 2D random variables, with zero mean (can be obtained by subtracting the mean to the images)
- Goal: find a filter h_R that will minimize the quadratic error

$$\varepsilon = E\left\{\left[f_i(x,y) - \hat{f}_i(x,y)\right]^2\right\}$$

- Calculating the 1st derivative, the error is minimal when

$$E\left\{\left[f_i(x,y)-\hat{f}_i(x,y)\right]f_o(x',y')\right\}=0$$

- By replacing $\hat{f}_i$ by its value, we get

$$E\{f_i(x,y)f_o(x',y')\}=\int_{-\infty}^{\infty}\int E\{f_o(i,j)f_o(x',y')\}\ h_R(x-i,y-j)didj$$

- The expectations of this products are the intercorrelation and the autocorrelation of the variables:

$$K_{f_i f_o}(x-x',y-y')=\int_{-\infty}^{\infty}\int K_{f_o}(i-x',j-y')\ h_R(x-i,y-j)didj$$

$$K_{f_i f_o}(x-x', y-y') = \int_{-\infty}^{\infty}\int K_{f_o}(i-x', j-y')\, h_R(x-i, y-j)\, didj$$

- By FT, we obtain

$$H_R(\omega_x, \omega_y) = \frac{P_{f_i f_o}(\omega_x, \omega_y)}{P_{f_o}(\omega_x, \omega_y)}$$

$P_{f_i f_o}(\omega_x, \omega_y)$ is the power interspectrum

$P_{f_o}(\omega_x, \omega_y)$ is the power spectrum of f_o

Image restoration: Wiener filtering

- When the noise is additive, we can write, by the Wiener-Kintchine theorems :

$$P_{f_o}(\omega_x,\omega_y) = \left|H_D(\omega_x,\omega_y)\right|^2 P_{f_i}(\omega_x,\omega_y) + P_N(\omega_x,\omega_y)$$

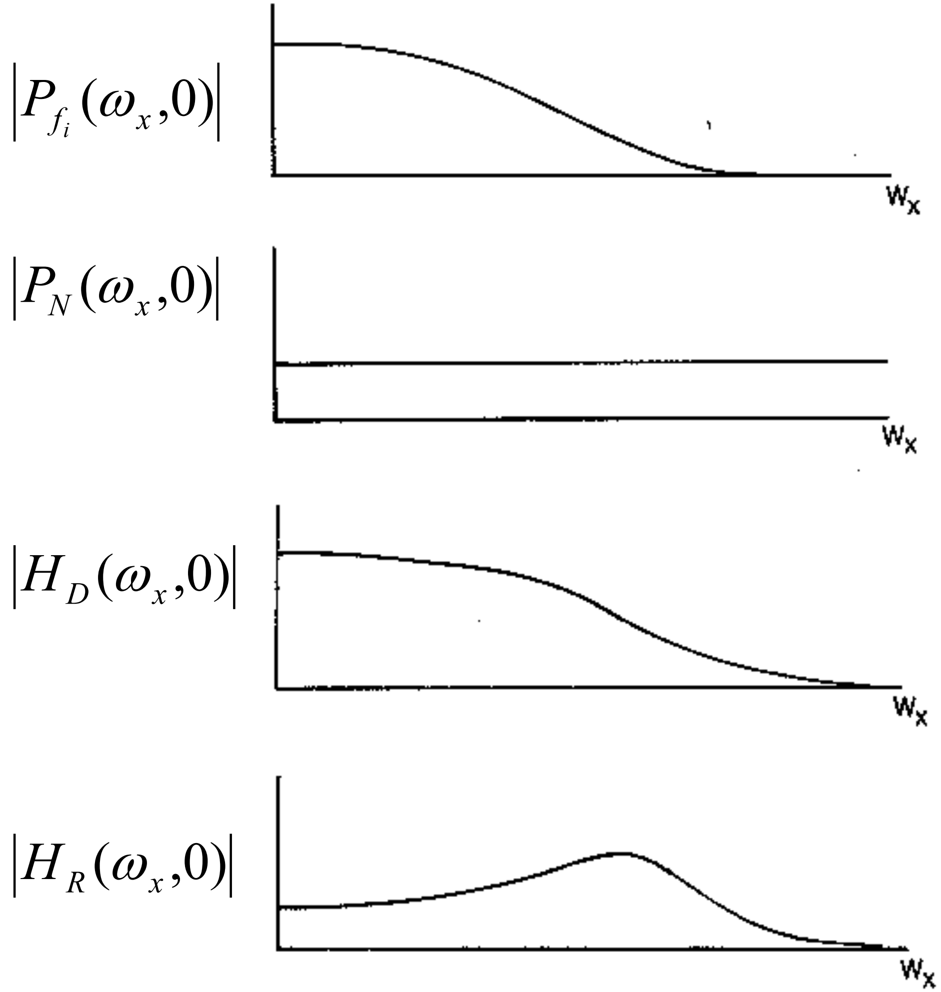
and

$$P_{f_o f_i}(\omega_x,\omega_y) = H_D^*(\omega_x,\omega_y) P_{f_i}(\omega_x,\omega_y)$$

- And we finally obtain the Wiener filter, with frequency response:

$$H_R(\omega_x,\omega_y) = \frac{H_D^*(\omega_x,\omega_y)}{\left|H_D^*(\omega_x,\omega_y)\right|^2 + \dfrac{P_N(\omega_x,\omega_y)}{P_{f_i}(\omega_x,\omega_y)}}$$

- Conclusions :
  - The Wiener filter is a **adaptive band-pass filter**
  - It behaves like the inverse filter at low frequencies and like a low-pass filter for high frequencies

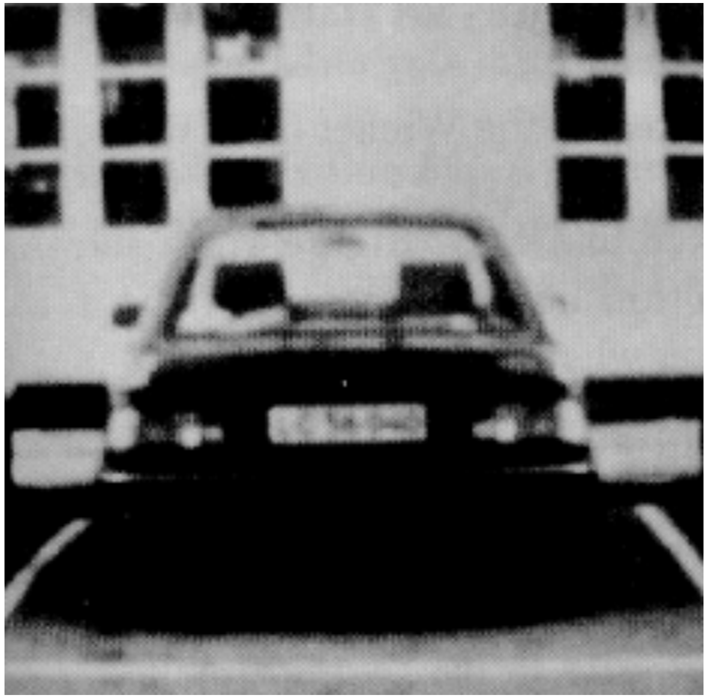


- Examples :

Motion blur

Restored Image

- Examples (cont.):



Out-of-focus blur

Restored image

- How to estimate h_D and the power spectrum of the noise?
  - h_D is the impulse response of the system. Thus if we find in the image a place that should contain a punctual object, one can deduce h_D
  - Similarly, a clear edge allows to evaluate the index response, integral of the impulse response
- For the power spectrum of the noise:
  - A uniform region in the image show the noise. The FT of its autorcorrelation gives an estimation of the power spectrum of the noise