Politecnico di Milano
Master of Science program in Biomedical Engineering

Biomedical Image Processing Lab class

(5 credits)

Class IV

April 13th 2012

Enrico Caiani, PhD

Image Noise

There are two ways to corrupt an image f(x,y) with noise n(x,y):

- use additive noise: $g(x,y)=f(x,y) + n(x,y)$
- use multiplicative noise: $g(x,y)=f(x,y) \bullet n(x,y)$

The former is easier to be removed, using spatial linear filtering or frequency filtering. To remove multiplicative noise, non linear methods are needed.

For the moment, we will consider only uncorrelated noise: changes in videointensity don't have any dependence with the pixels spatial position.

The classification of the noise type in the image is based on the shape of its probability density function (noise histogram), that can be obtained by selecting a region of interest in the image backgound. This is also at the base of the image restoration process.

Image Noise with Matlab

With Matlab, it is possible to add different types of noise to the image:

G = **imnoise**(I,TYPE,PARAMETERS)

where TYPE can be:

'gaussian': Gaussian white noise with constant mean and variance
'localvar': Zero-mean Gaussian white noise with an intensity-dependent variance
'poisson': Poisson noise
'salt & pepper': "On and Off" pixels
'speckle': Multiplicative noise

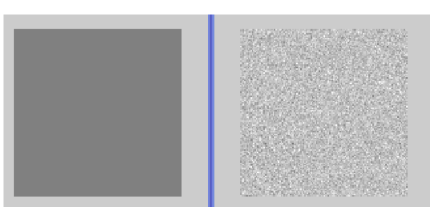
The mean and variance parameters for 'gaussian', 'localvar', and 'speckle' noise types are always specified as if for a double image in the range [0, 1]. If the input image is of class uint8 or uint16, the function converts the image to double, adds noise according to the specified type and parameters, and then converts the noisy image back to the same class as the input.

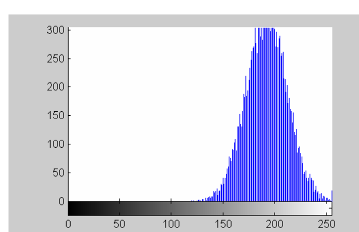
If I has more than two dimensions it is treated as a multidimensional intensity image and not as an RGB image.

Gaussian noise:

G = **imnoise**(I,'gaussian',MEDIA,DEV.ST.^{2}); [0, 0.01]

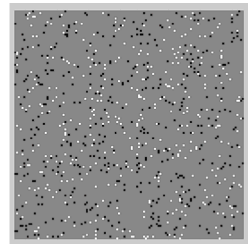
It is used to model unknown sources of additive noise as, for the central limit theorem, the sum of a large number of independent noise sources approximates a Gaussian noise.
Given the mean and standard deviation, the 99.7% of the noise is included into the ±3 st.dev interval, around the mean. If the range is out of the allowed values (0-1), there will be saturation of the corresponding extreme level.





G = **imnoise**(I,'gaussian',64/255,400/(255)^2);

<u>Salt & Pepper noise</u>:



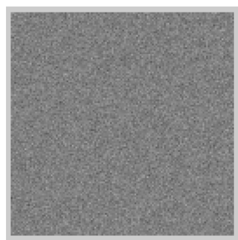
G=**imnoise**(I,'salt & pepper',d);   [0.05]

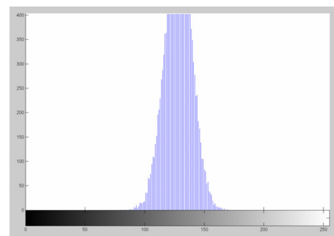
This noise is not additive nor multiplicative. It is obtained by substituting d/2% (between 0 and 1) of the image pixels with pixels 0 (pepper) and d/2% with pixels 1 (salt).

<u>Poisson noise</u>:

H=**imnoise**(I,'poisson');

It generates a Poisson noise overimposed to I. The videointensity in the input image should correspond to the number of photons counted in PET and SPECT.



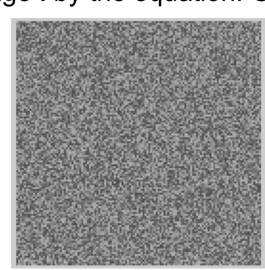


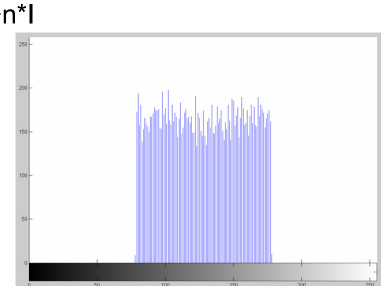
Caiani 13/04/2012

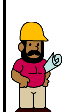
<u>Multiplicative noise</u>:

G=**imnoise**(I,'speckle',V)   [0.04]

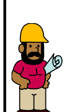
A uniform multiplicative noise n(x,y), of zero mean and variance V is added to the image I by the equation: G=I+n*I







Generate a 128x128 image with all elements equal to 128.
Add to it the different types of noise available, applying different parameters, and see how the histogram is modified.

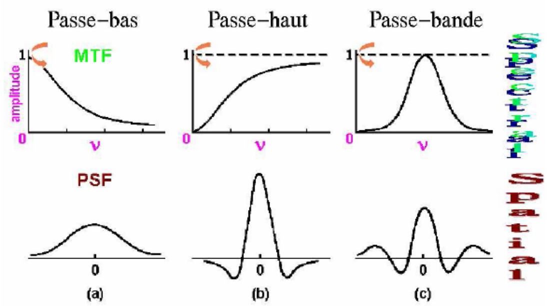


Load an image.
Separately, add Gaussian, salt and pepper,and speckle noise, observing the histogram variations in a selected ROI.

Caiani 13/04/2012

FILTRAGE SPATIALE
INTRODUCTION

Trois Types de Filtrage



- **PSF** : Point Spread Function (ou Fonction d'Étalement Spectrale)
- **MTF** : Modulation Transfer Function (ou Fonction de Transfert)

▶ Filtre Passe-bas : diminue le bruit mais atténue les détails de l'image
▶ Filtre Passe-haut : accentue les contours et les détails de l'image mais amplifie le bruit
▶ Filtre Passe-bande : élimine certaines fréquences indésirables présentes dans l'image

Caiani 13/04/2012

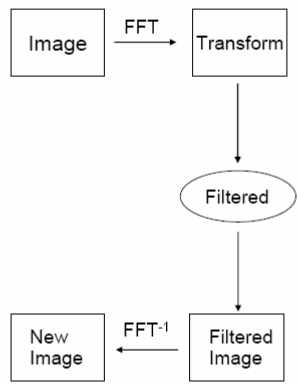
Spatial Filters

- Operate on raw image data in the *(r,c)* space, by considering small neighborhoods, 3x3, 5x5, 7x7, and moving sequentially across and down the image
- Returns a result based on a linear or nonlinear operation
- Consists of three types of filters:
  - Mean filters
  - Median filters
  - Enhancement filters

- Many spatial filters are linear filters implemented with a convolution mask: the result is a weighted sum of a pixel and its neighbors
- Mask coefficients tend to effect the image in the following general ways:
  - Coefficients are positive: blurs the image
  - Coefficients are alternating positive and negative: sharpens the image
  - Coefficients sum to 1: brightness retained
  - Coefficients sum to 0: dark image

Caiani 13/04/2012

Image Operation in the Frequency Domain

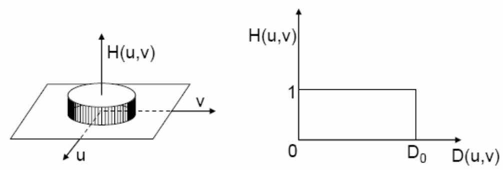


H(u,v) - Ideal Low Pass Filter

$$H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) > D_0 \end{cases}$$

$D(u,v) = \sqrt{u^2 + v^2}$

D_0 = cut off frequency



Frequency Domain and Convolution Mask Lowpass Filtering

- Frequency domain lowpass filters smooth images by attenuating high frequency components that correspond to rapidly changing brightness values in the original image
- Ideal filters cause undesirable artifacts, but a Butterworth filter does not
- The amount of information suppressed is determined by the cutoff frequency of the filter
- The convolution theorem allows us to use filter masks in the spatial domain to perform filtering
- The masks are spatial domain approximations to frequency domain filters
- For lowpass filtering these masks are some form of average (mean) filters

Convolution Mask Lowpass Filtering

- Example spatial convolution masks for lowpass filtering:

$$\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\quad\begin{bmatrix}1&1&1\\1&2&1\\1&1&1\end{bmatrix}\quad\begin{bmatrix}2&1&2\\1&4&1\\2&1&2\end{bmatrix}\quad\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

Standard arithmetic mean filters

Approximations to Gaussian filters

Note: the normalizing multiplier is implied or is handled by postprocessing remapping

- The coefficients for these filter masks are all positive, unlike the highpass filters where the center is surrounded by negative coefficients
- For application specific reasons, the coefficients in the mask may be biased, that is, they may not all be 1's
- For image smoothing, the results from most of the spatial mean filters are visually similar

Caiani 13/04/2012

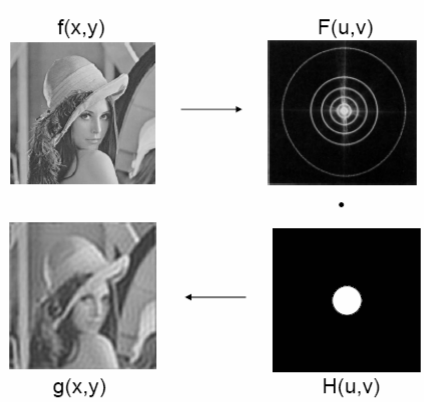
Low pass Filter

spatial domain — frequency domain

$f(x,y) \longrightarrow F(u,v)$

$G(u,v) = F(u,v) \cdot H(u,v)$ — filter

$g(x,y) \longleftarrow G(u,v)$



Blurring - Ideal Low pass Filter

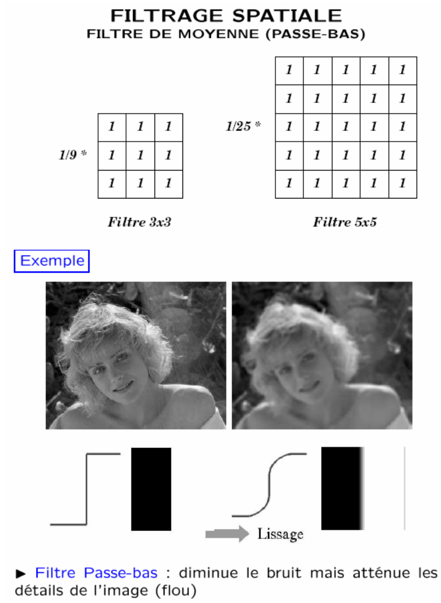
Caiani 13/04/2012

Mean filters

- Averaging filters
- Tend to blur the image
- Adds a softer look to the image
- Example 3x3 convolution mask

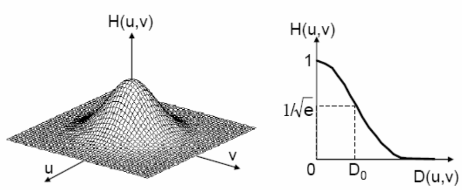
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Various types of mean filters are most effective with different types of noise
- We can use a larger mask size for a greater smoothing effect
- Gaussian filters create a more natural effect and less noticeable blurring as the mask size increases, compared to arithmetic mean filters



Gaussian filters

H(u,v) - Gaussian Filter



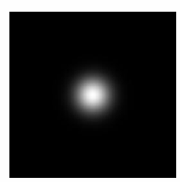
$$H(u,v) = e^{-D^2(u,v)/(2D^2_0)}$$

$$D(u,v) = \sqrt{u_2 + v_2}$$

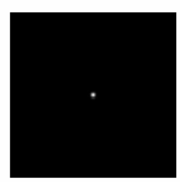
Softer Blurring + no Ringing

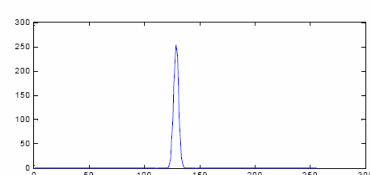
The Gaussian Lowpass Filter

Freq. domain

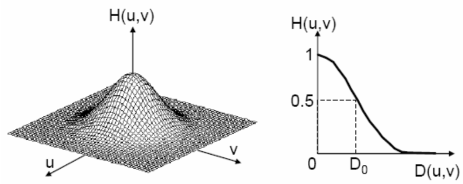


Spatial domain



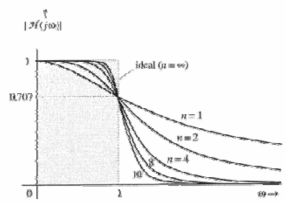


H(u,v) - Butterworth Filter



$$H(u,v) = \frac{1}{1 + (D(u,v)/D_0)^{2n}}$$

$$D(u,v) = \sqrt{u_2 + v_2}$$



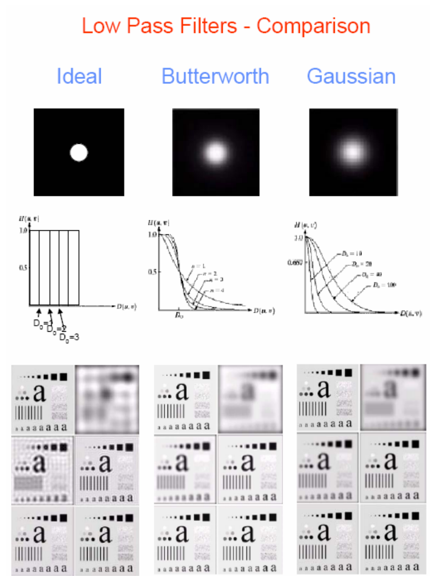
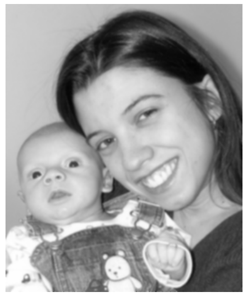


Image Smoothing

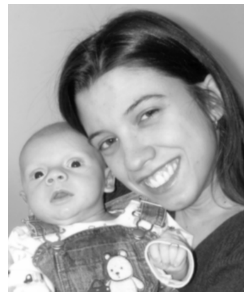
- Image smoothing will give the image a softer look
- Image smoothing is also used for noise mitigation
- Image smoothing is performed in spatial domain and frequency domain
- In the spatial domain image smoothing is accomplished by considering a pixel and its neighbors and eliminating any extreme values with median filters or by averaging with neighboring pixels with mean filters
- In the frequency domain, image smoothing is accomplished by some form of lowpass filtering

Image Smoothing

a) Original image



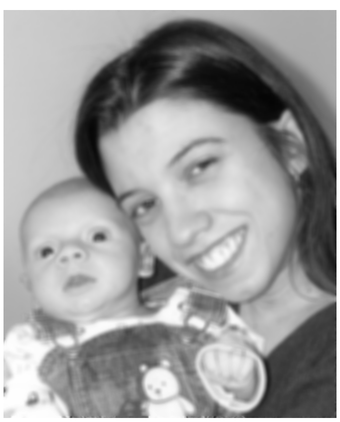
b) Arithmetic mean, 3x3



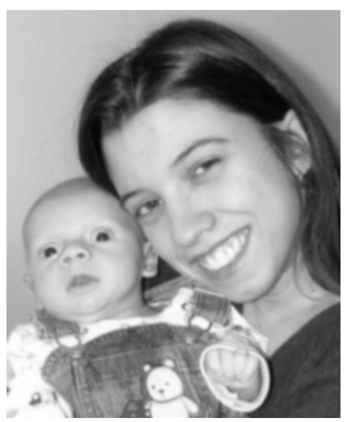
c) Gaussian, 3x3

Caiani 13/04/2012

Image Smoothing



d) Arithmetic mean, 5x5

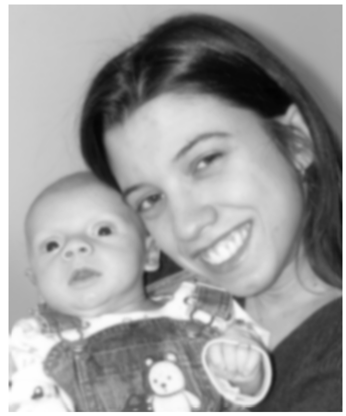


d) Gaussian, 5x5

Caiani 13/04/2012

Image Smoothing

e) Arithmetic mean, 7x7



f) Gaussian, 7x7

Caiani 13/04/2012

FILTRAGE SPATIALE

AUTRES FILTRES PASSE-BAS

Filtre Binomial

Les coefficients de ce filtre sont obtenus par le binome de Newton. Un filtre 1D Binomial du quatrième ordre donne le vecteur (1/16)(1 4 6 4 1). Le filtre 2D est

$$\frac{1}{256}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

Filtre Pyramidal

$$\frac{1}{81}\begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 3 & 2 & 1 \\ \hline 2 & 4 & 6 & 4 & 2 \\ \hline 3 & 6 & 9 & 6 & 3 \\ \hline 2 & 4 & 6 & 4 & 2 \\ \hline 1 & 2 & 3 & 2 & 1 \\ \hline \end{array}$$

Filtre Conique

$$\frac{1}{25}\begin{array}{|c|c|c|c|c|}\hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 2 & 2 & 2 & 0 \\ \hline 1 & 2 & 5 & 2 & 1 \\ \hline 0 & 2 & 2 & 2 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}$$

Load an image.
Implement a 3x3 filter mask of mean filter and gaussian filter; implement the other masks here aside.
Apply them to the original image you chose before, and their noise corrupted versions, observing when the better performance is achieved.

Caiani 13/04/2012

SPATIAL FILTERING IN MATLAB

By the command **fspecial** it is possible to create predefined spatial filter masks:

w=**fspecial**('type',parameters)

Type	Syntax and Parameters
'average'	fspecial('average', [r c]). A rectangular averaging filter of size r × c. The default is 3 × 3. A single number instead of [r c] specifies a square filter.
'disk'	fspecial('disk', r). A circular averaging filter (within a square of size 2r + 1) with radius r. The default radius is 5.
'gaussian'	fspecial('gaussian', [r c], sig). A Gaussian lowpass filter of size r × c and standard deviation sig (positive). The defaults are 3 × 3 and 0.5. A single number instead of [r c] specifies a square filter.
'laplacian'	fspecial('laplacian', alpha). A 3 × 3 Laplacian filter whose shape is specified by alpha, a number in the range [0, 1]. The default value for alpha is 0.5.
'log'	fspecial('log', [r c], sig). Laplacian of a Gaussian (LoG) filter of size r × c and standard deviation sig (positive). The defaults are 5 × 5 and 0.5. A single number instead of [r c] specifies a square filter.
'motion'	fspecial('motion', len, theta). Outputs a filter that, when convolved with an image, approximates linear motion (of a camera with respect to the image) of len pixels. The direction of motion is theta, measured in degrees, counterclockwise from the horizontal. The defaults are 9 and 0, which represents a motion of 9 pixels in the horizontal direction.
'prewitt'	fspecial('prewitt'). Outputs a 3 × 3 Prewitt mask, wv, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: wh = wv'.
'sobel'	fspecial('sobel'). Outputs a 3 × 3 Sobel mask, sv, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: sh = sv'.
'unsharp'	fspecial('unsharp', alpha). Outputs a 3 × 3 unsharp filter. Parameter alpha controls the shape; it must be greater than 0 and less than or equal to 1.0; the default is 0.2.

TABLE 3.4 Spatial filters supported by function fspecial.

Once the filter is defined, it is possible to visualize their coefficient (spatial kernel) and its frequency response by **freqz2**(w).

Caiani 13/04/2012

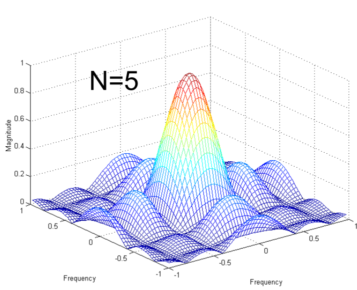
MEAN (average) filter

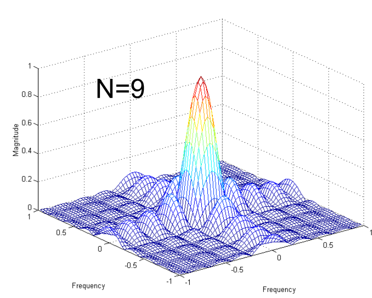
Low-pass filter, appropriate to remove Gaussian noise; its application blurs the contours.

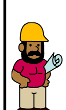
w=**fspecial**('average',N)   with N=mask dimensions.

By this filtering, the dev.st. of the Gaussian noise is reduce by 1/**sqrt** (N).

w=**fspecial**('disk',r)



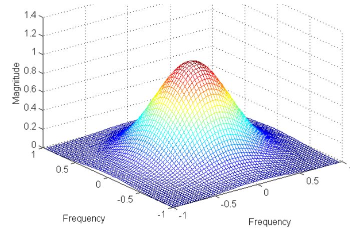




Apply the filter (with average and disk), with different N, to the image previously corrupted with Gaussian noise. Observe the filter frequency responce and its effect on the image.

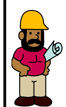
Caiani 13/04/2012

Gaussian filter



h=**fspecial**('gaussian',N,SIGMA);   [3x3, 0.5]

with N=dimensions and SIGMA= st. dev. in pixel. Differently from the mean filter, it is circular symmetric.

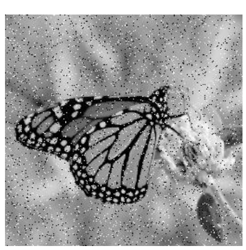


Apply the Gaussian filter, with different N and SIGMA, to the image previously corrupted with Gaussian noise. Observe the filter frequency responce and its effect on the image.

Caiani 13/04/2012

Median Filtering

- Nonlinear filter
- Sorts the pixel values in a small neighborhood and replaces the center pixel with the middle value in the sorted list
- Output image needs to be written to a separate image (a buffer), so that results are not corrupted
- Neighborhood can be of any size but 3x3, 5x5 and 7x7 are typical



Original image with
salt and pepper noise

Median filtered image (3x3)

Caiani 13/04/2012