Image Enhancement

Part 2: Neighborhood operations

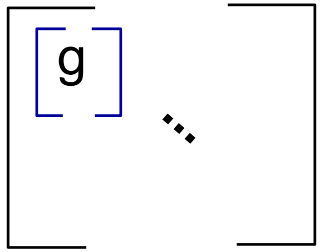
Neighborhood operations

- Correlation $\leftrightarrow$ pattern recognition
- Convolution $\leftrightarrow$ Linear filtering
  - Edge detection
  - Denoising

Correlation

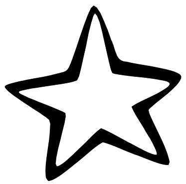
- Correlation
  - Measures the similarity between two signals
  - Difference from convolution: no ‘minus’ signs in the formula
    - the signals need only to be translated

$$C(m,n) = \sum_k \sum_r f[m,n] h_{template}[m+k, n+r]$$

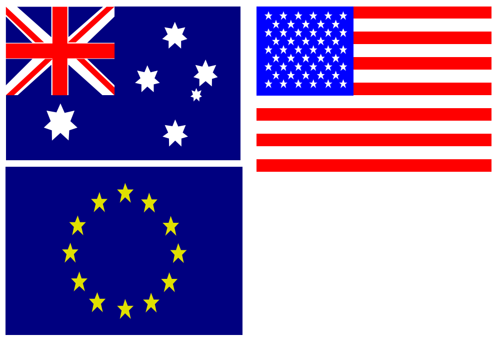


Application

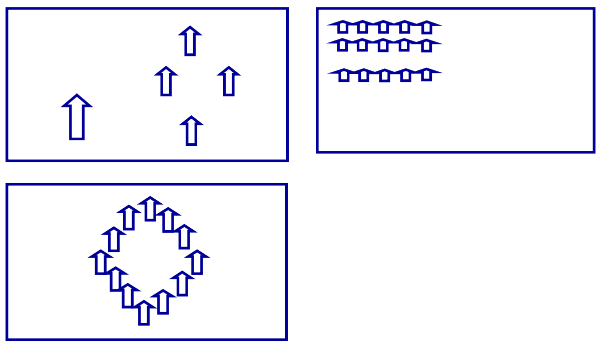
pattern



database



matching result



Convolution

$$g[m,n] = f[m,n] * h_{filter}[m,n] = \sum_{k,r} f[m,n] h_{filter}[k-m, r-n]$$

$$G(j\omega_x, j\omega_y) = F(j\omega_x, j\omega_y) H_{filter}(j\omega_x, j\omega_y)$$

$f[m,n]$: original (input) image

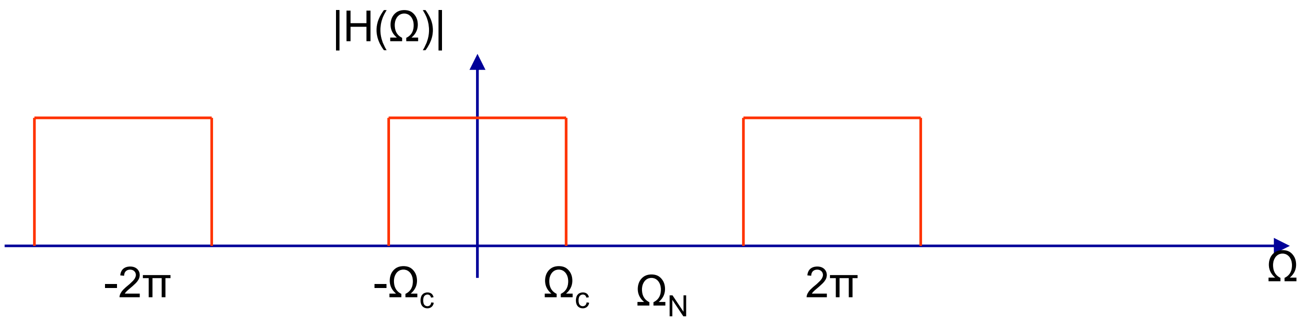
$g[m,n]$: filtered (output) image

$h_{filter}[m,n]$: filter impulse response

Convolution and digital filtering

- Digital filtering consists of a convolution between the image and the impulse response of the filter, which is also referred to as convolution kernel.
- Warning: both the image and the filter are matrices (2D).
  - If the filter is separable, then the 2D convolution can be implemented as a cascade of two 1D convolutions
- Filter types
  - FIR (Finite Impulse Response)
  - IIR (Infinite IR)

Ideal interpolation *digital* filter



Digital LP filter (discrete time)

The boundary between the pass-band and the stop-band is sharp

The spectrum is periodic (the signal is sampled)

The repetitions are located at integer multiples of 2π

The low-pass filtered signal is *still* a digital signal, but with a different frequency content

The impulse response h[n] in the signal domain is discrete time and corresponds to the `sinc[]` function

Reconstruction LP filter (continuous time)

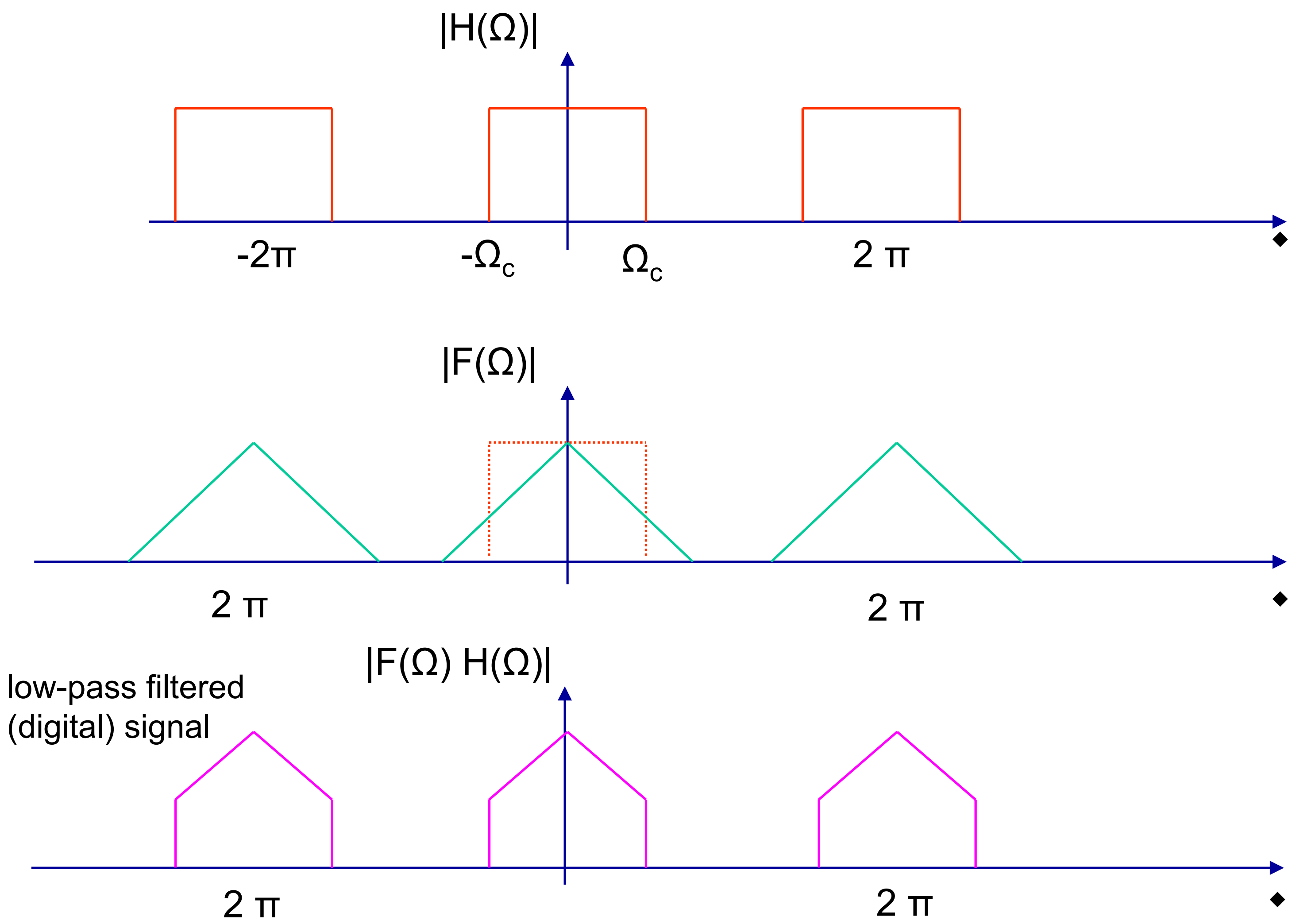
The boundary between the pass-band and the stop-band is sharp

The spectrum consists of ***one repetition only (the resulting signal is CT, ideally)***

The low-pass filtered signal is a continuous time signal, that might have a different frequency content

The impulse response h(t) in the signal domain is continuous time and corresponds to the `sinc()` function

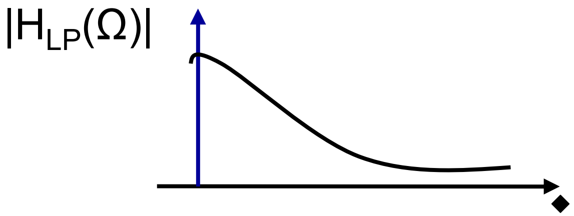
Digital LP filtering

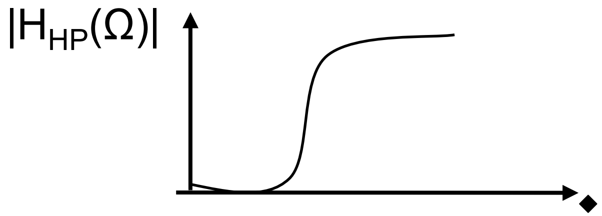


LP and HP filtering

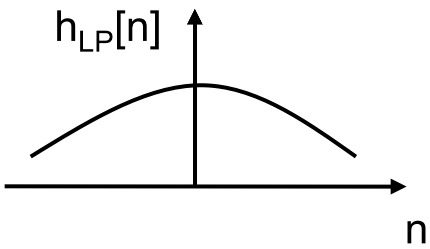
Low-pass High-pass

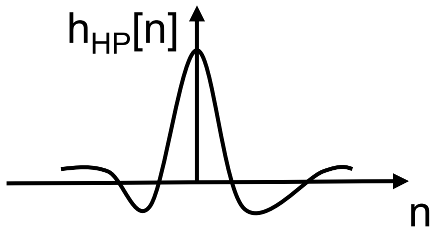
Frequency domain





Signal domain





integration

averaging

differentiation

Notation reminder

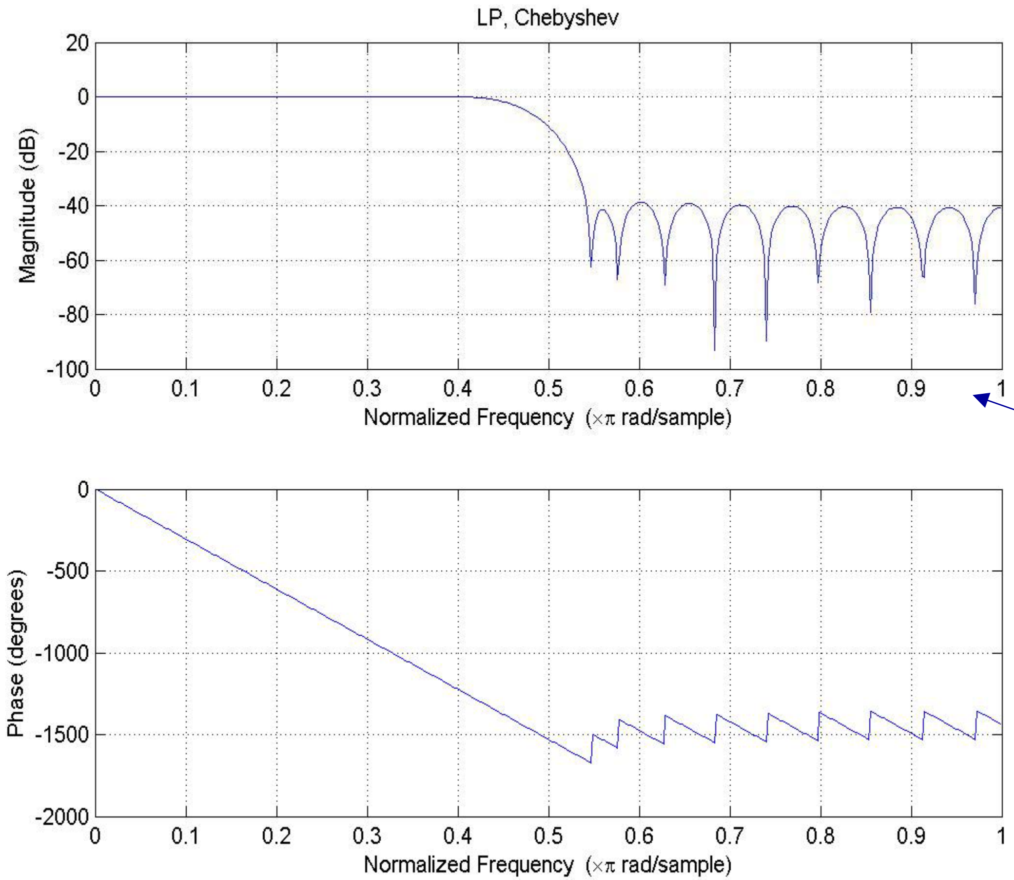
$\omega_s = \dfrac{2\pi}{T_s}$ periodicity in Fourier domain for sampled signals (Ts=sample distance)

$\Omega_s = 2\pi$ periodicity in FD after frequency normalization (Ts=1)

$\Omega = \omega T_s$ relation among frequency and normalized frequency

$u = \dfrac{\Omega}{2\pi}$ unitary frequency (periodicity in frequency domain=1)

Example: Chebyshev LP

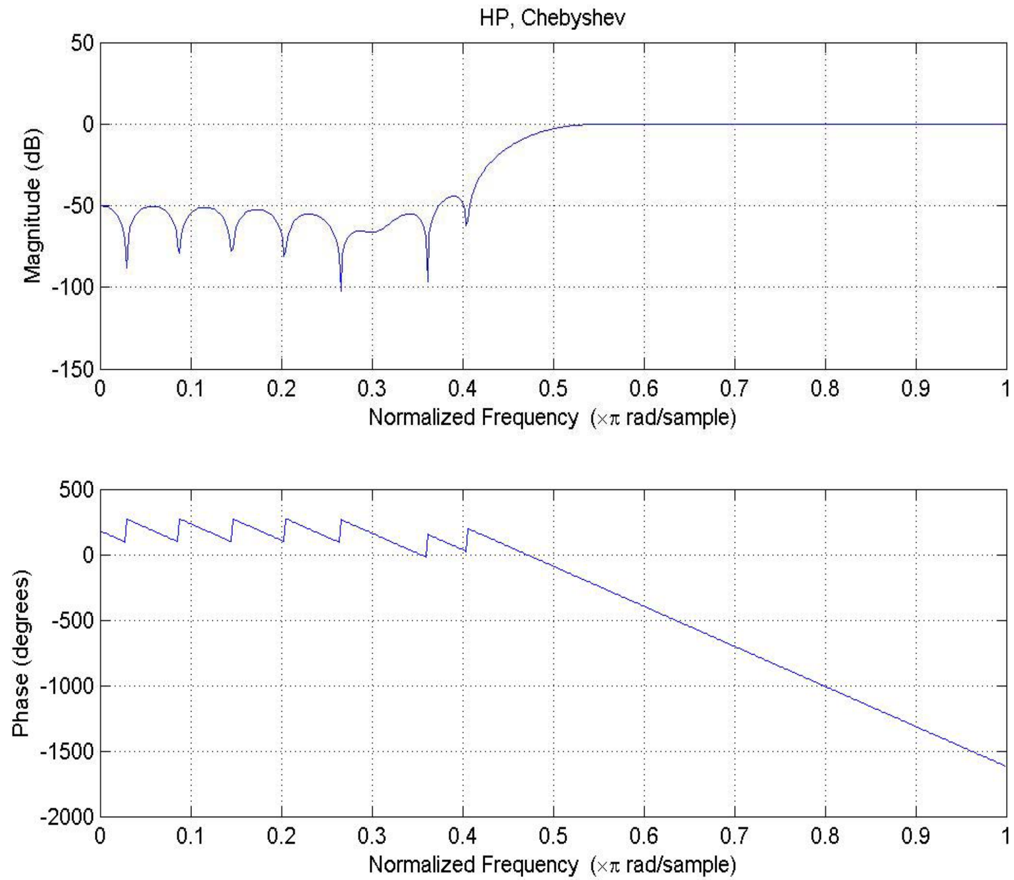


Reminder:

$\Omega=2\pi u$

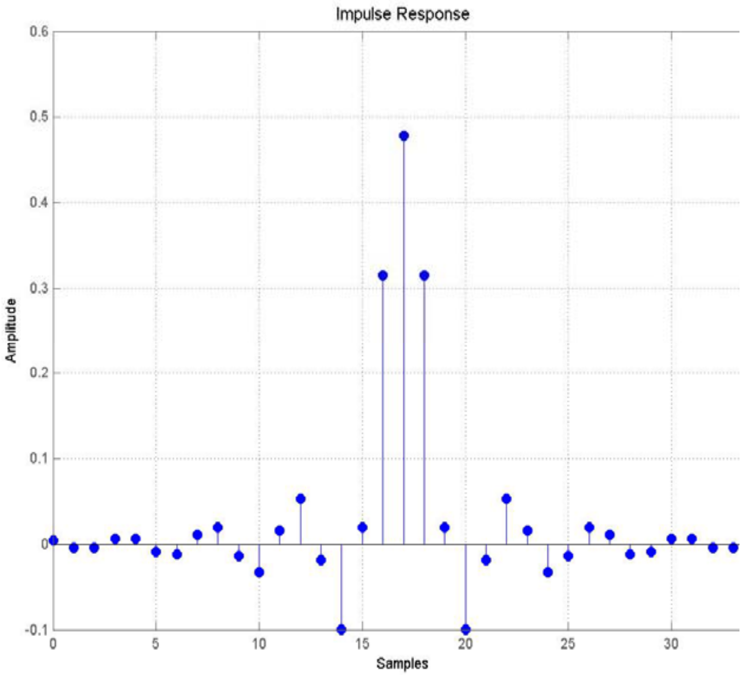
this is *u*

Example: Chebyshev HP

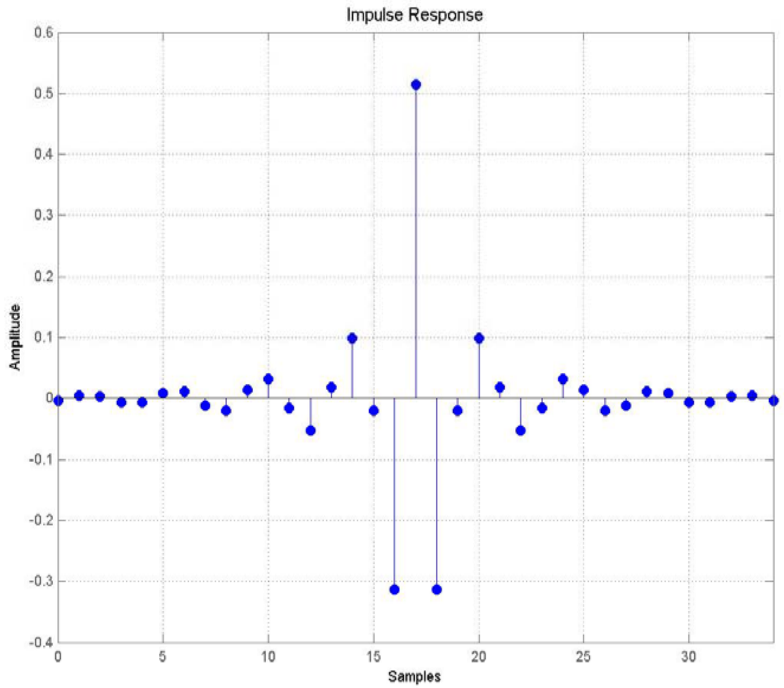


Example: Chebyshev Impulse Response

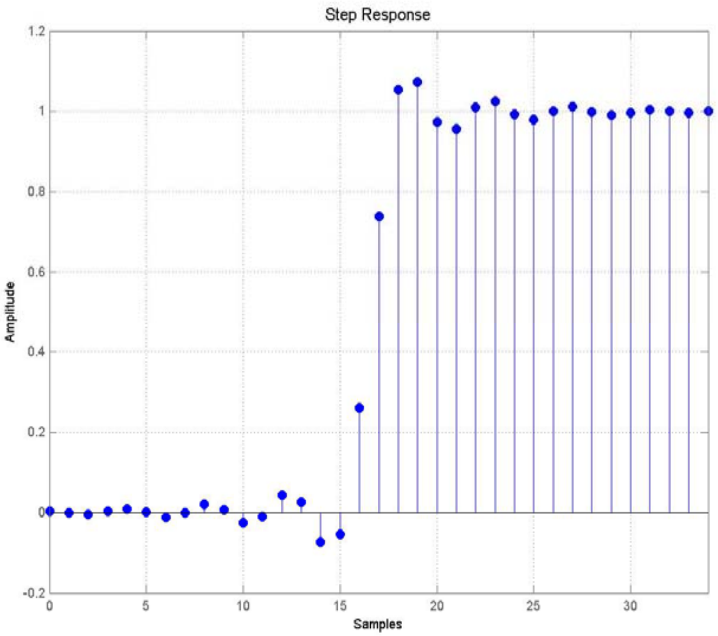
LP

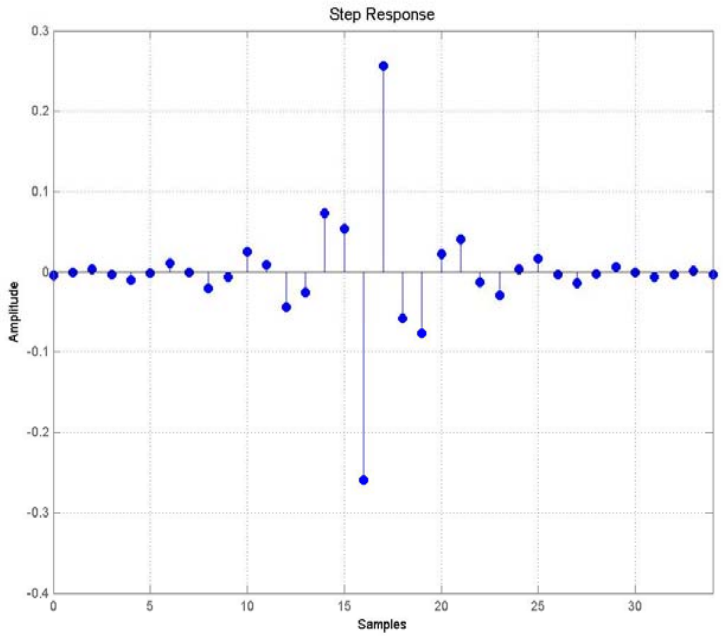


HP

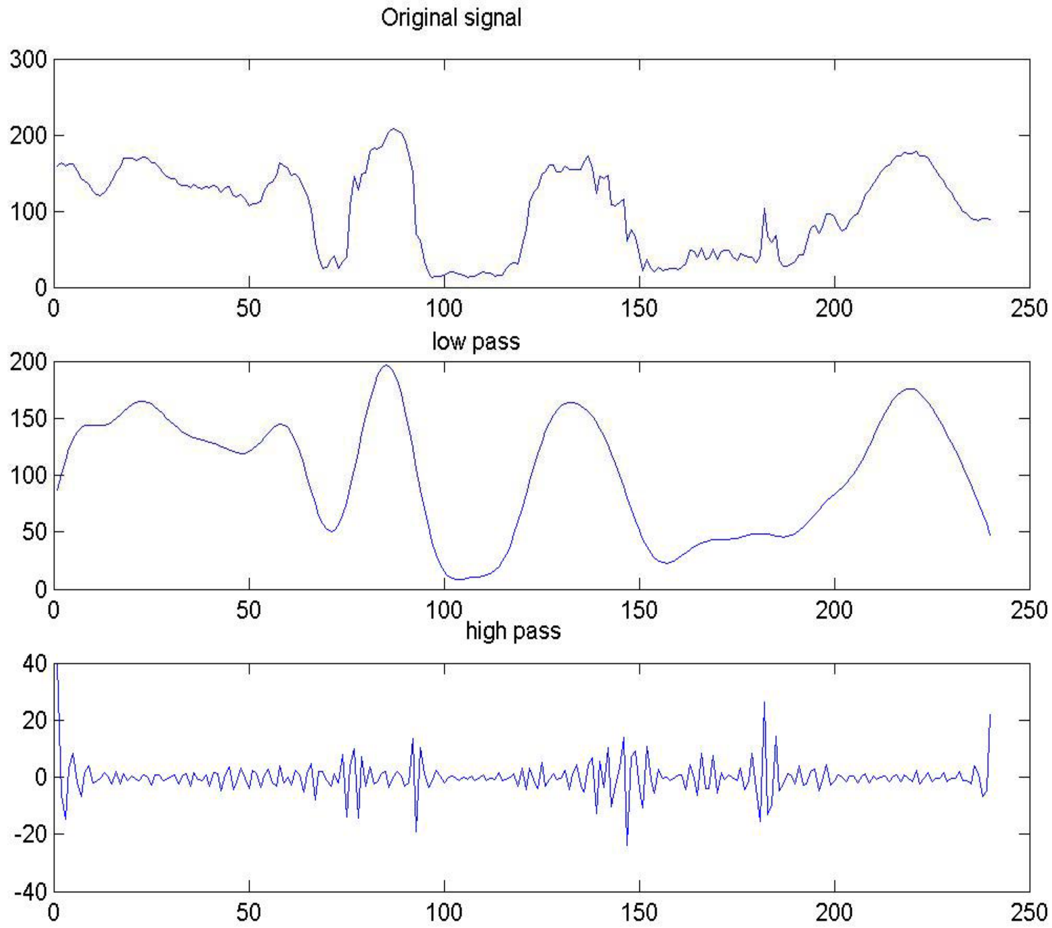


Example: Chebyshev Step Response





Example: filtered signal



The transfer function (or, equivalently, the impulse response) of the filter determines the characteristics of the resulting signal

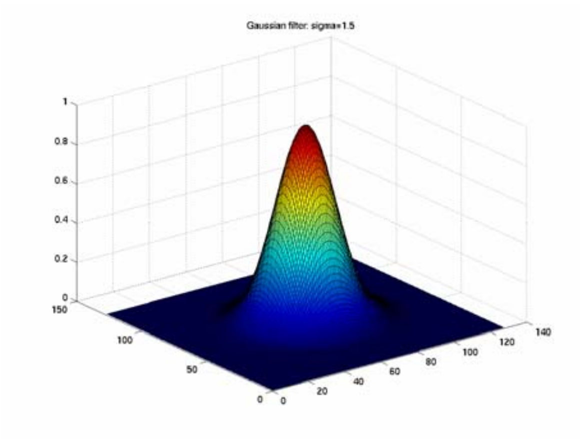
Switching to images

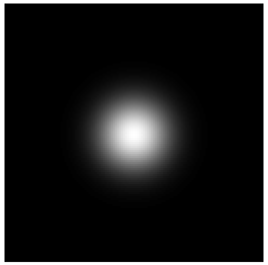
- Images are 2D digital signals (matrices) → filters are matrices
- Low-pass ↔ *averaging* (*discrete* interpolation) ↔ smoothing
- High-pass ↔ differentiation ↔ emphasize image *details* like lines, and, more in general, sharp variations of the luminance
- Band-pass: same as high pass but selecting a predefined *range* of spatial frequencies

►►Setting apart low-pass and high-pass image features is the ground of *multi-resolution*. It is advantageous for many tasks like contour extraction (edge detection), image compression, feature extraction for pattern recognition, image denoising etc.

2D filters

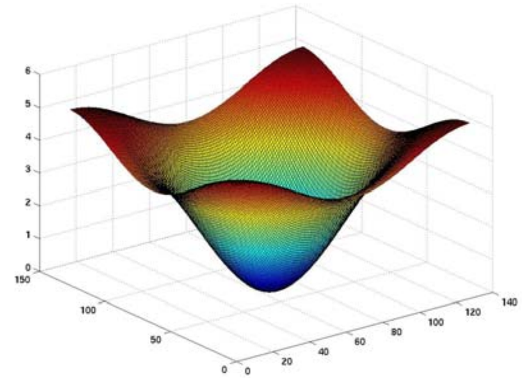
Low-pass

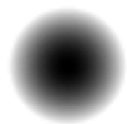




$$h_{lowpass} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

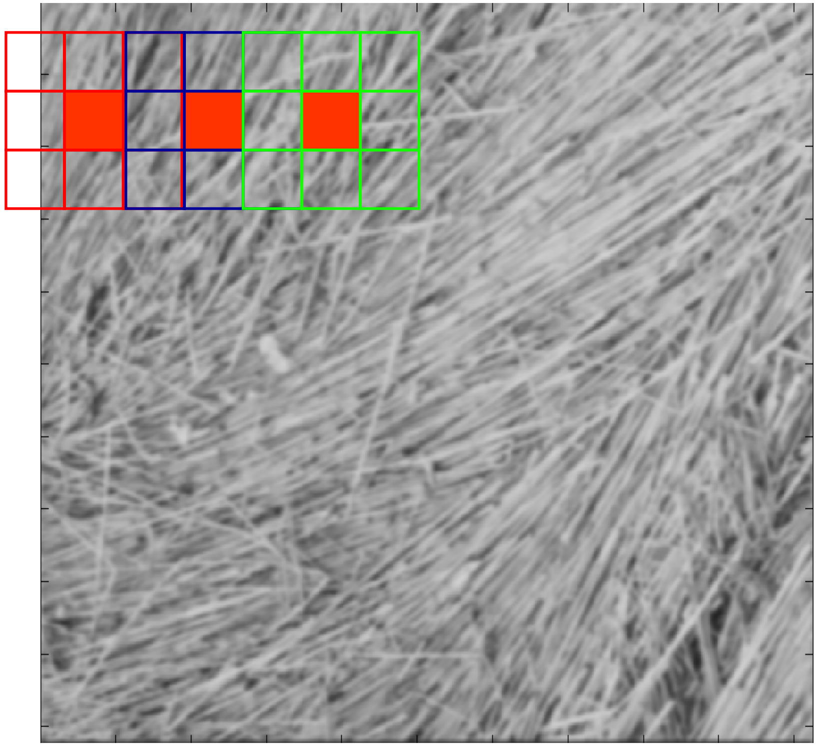
High-pass





$$h_{highpass} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

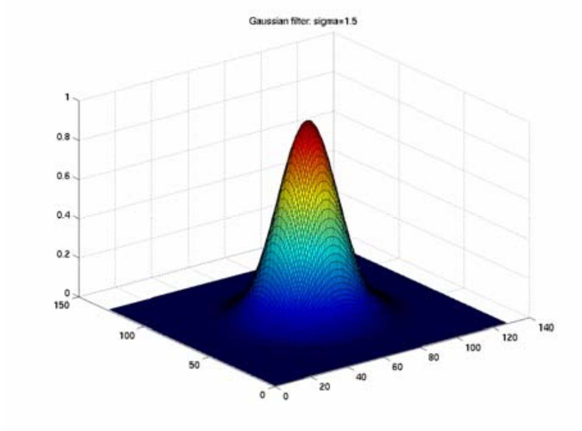
Filtering in image domain

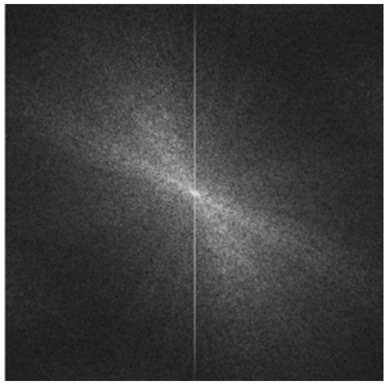


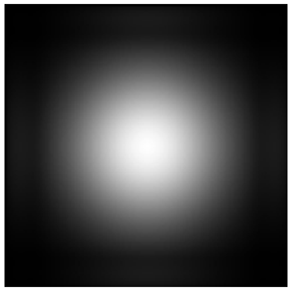
Filtering in image domain is performed by *convolving* the image with the filter kernel. This operation can be though of as a pixel-by-pixel product of the image with a *moving kernel*, followed by the sum of the pixel-wise output.

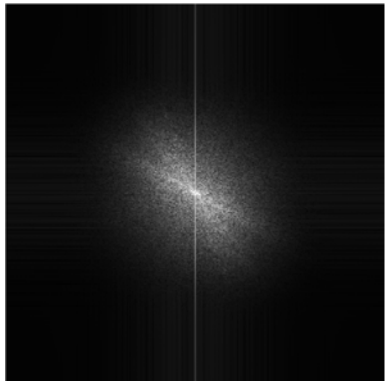
Low-pass filtering: example

$f[m,n]$ $h_{lowpass}[m,n]$ $g[m,n]$

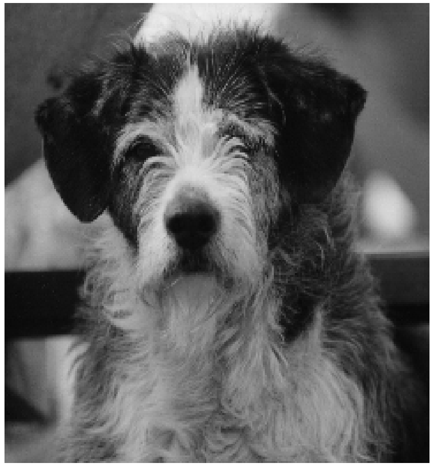




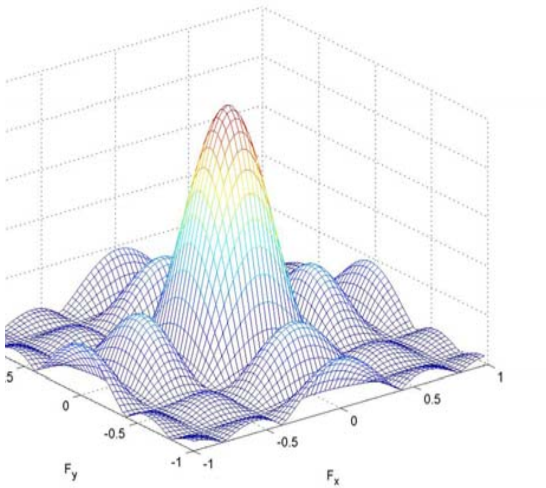




Averaging



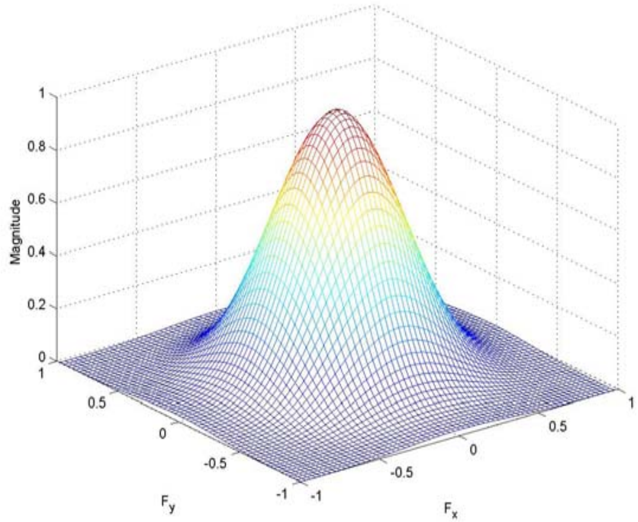
$$h_{lp} = 1/25 \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix}$$



Gaussian

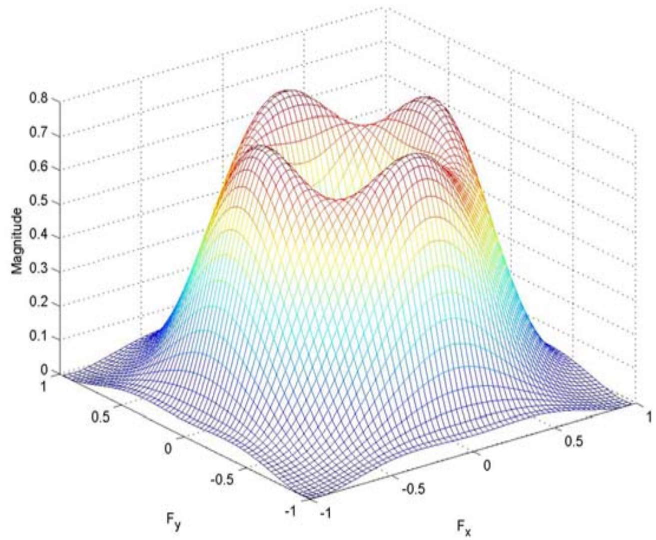
h_{lp}=

0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
0.0133	0.0596	0.0983	0.0596	0.0133
0.0030	0.0133	0.0219	0.0133	0.0030



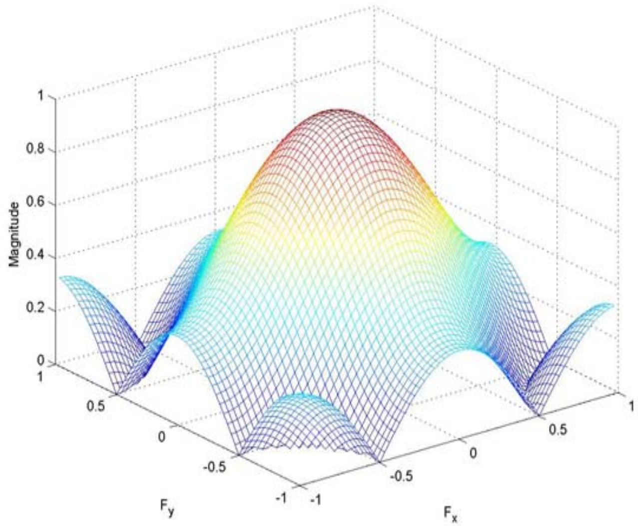
Laplacian of Gaussian (LoG)

$$h_{hp}=\begin{matrix} 0.0239 & 0.0460 & 0.0499 & 0.0460 & 0.0239 \\ 0.0460 & 0.0061 & -0.0923 & 0.0061 & 0.0460 \\ 0.0499 & -0.0923 & -0.3182 & -0.0923 & 0.0499 \\ 0.0460 & 0.0061 & -0.0923 & 0.0061 & 0.0460 \\ 0.0239 & 0.0460 & 0.0499 & 0.0460 & 0.0239 \end{matrix}$$

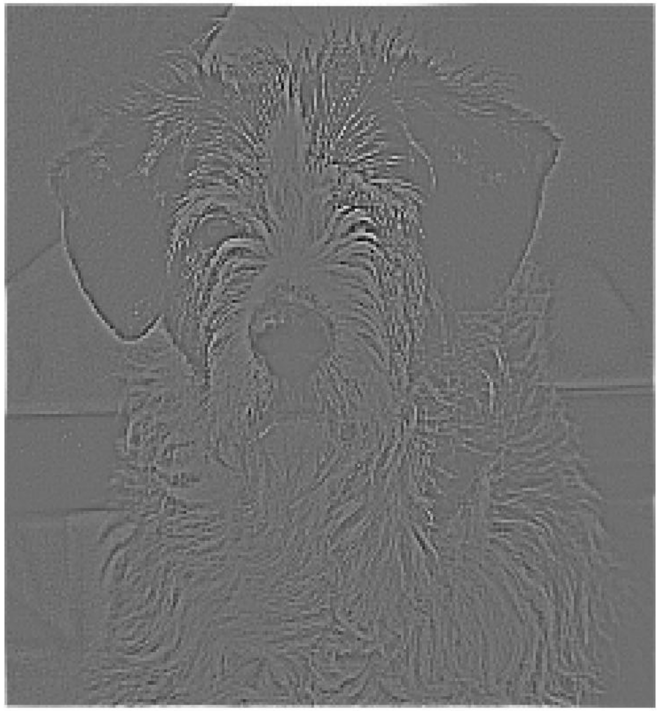


Asymmetric LP

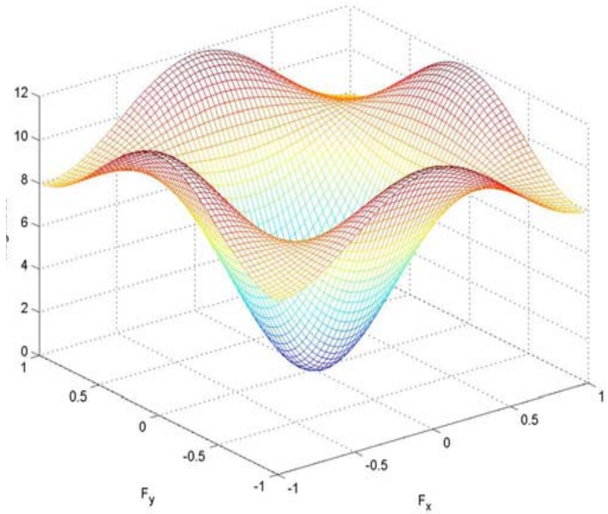
$$h_{lp}=1/6 \begin{matrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{matrix}$$



Asymmetric HP



$$h_{lp}= \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$



Basic Highpass Spatial Filtering

- The filter should have positive coefficients near the center and negative in the outer periphery:

Laplacian mask

$$\frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

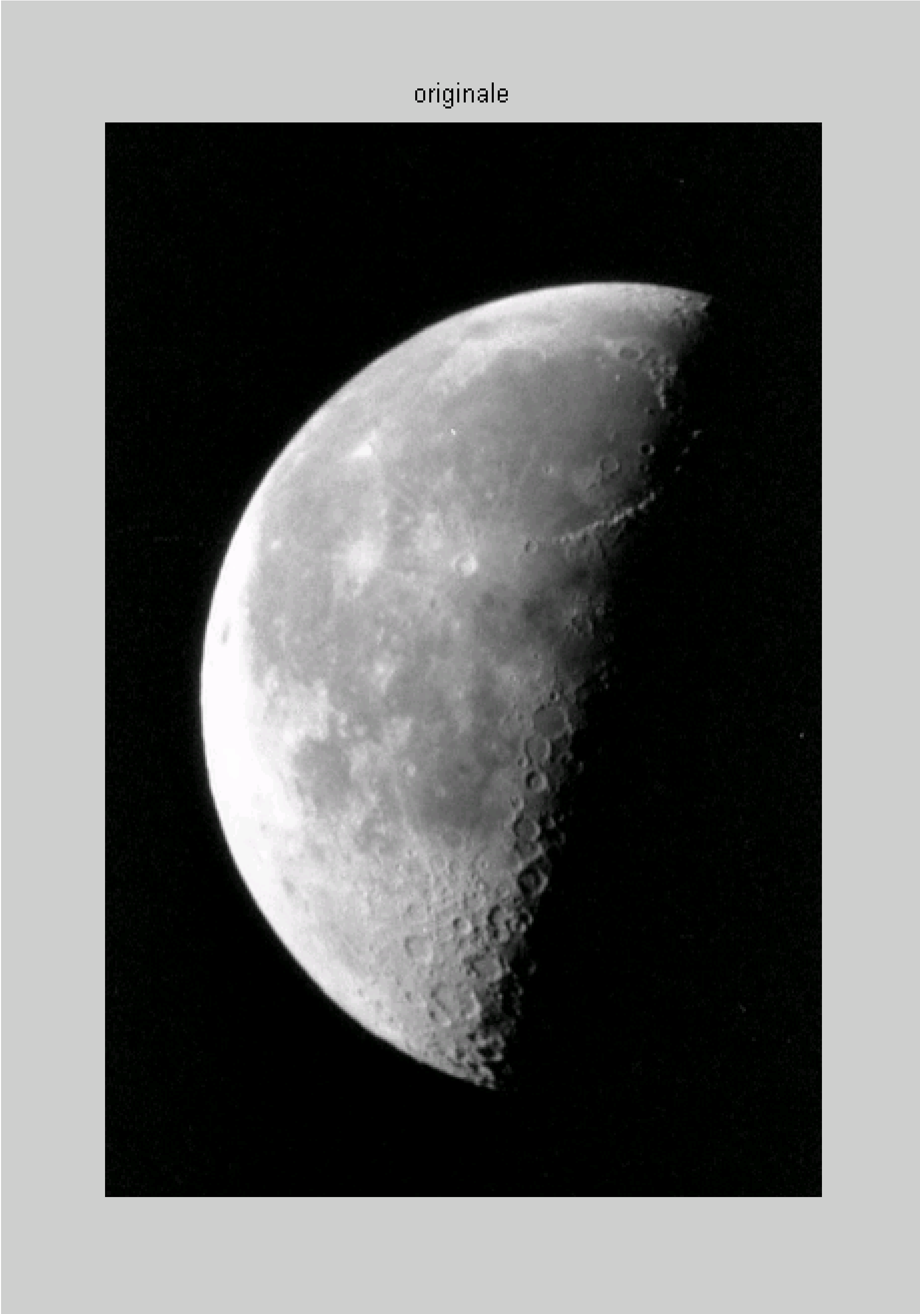
a b
c d

FIGURE 3.39 (a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

Other Laplacian masks (normalization factor is missing)

Basic Highpass Spatial Filtering

- The sum of the coefficients is 0, indicating that when the filter is passing over regions of almost stable gray levels, the output of the mask is 0 or very small.
- The output is high when the center value differ from the periphery.
- The output image does not look like the original one.
- The output image depicts all the fine details
- Some scaling and/or clipping is involved (to compensate for possible negative gray levels after filtering).


originale

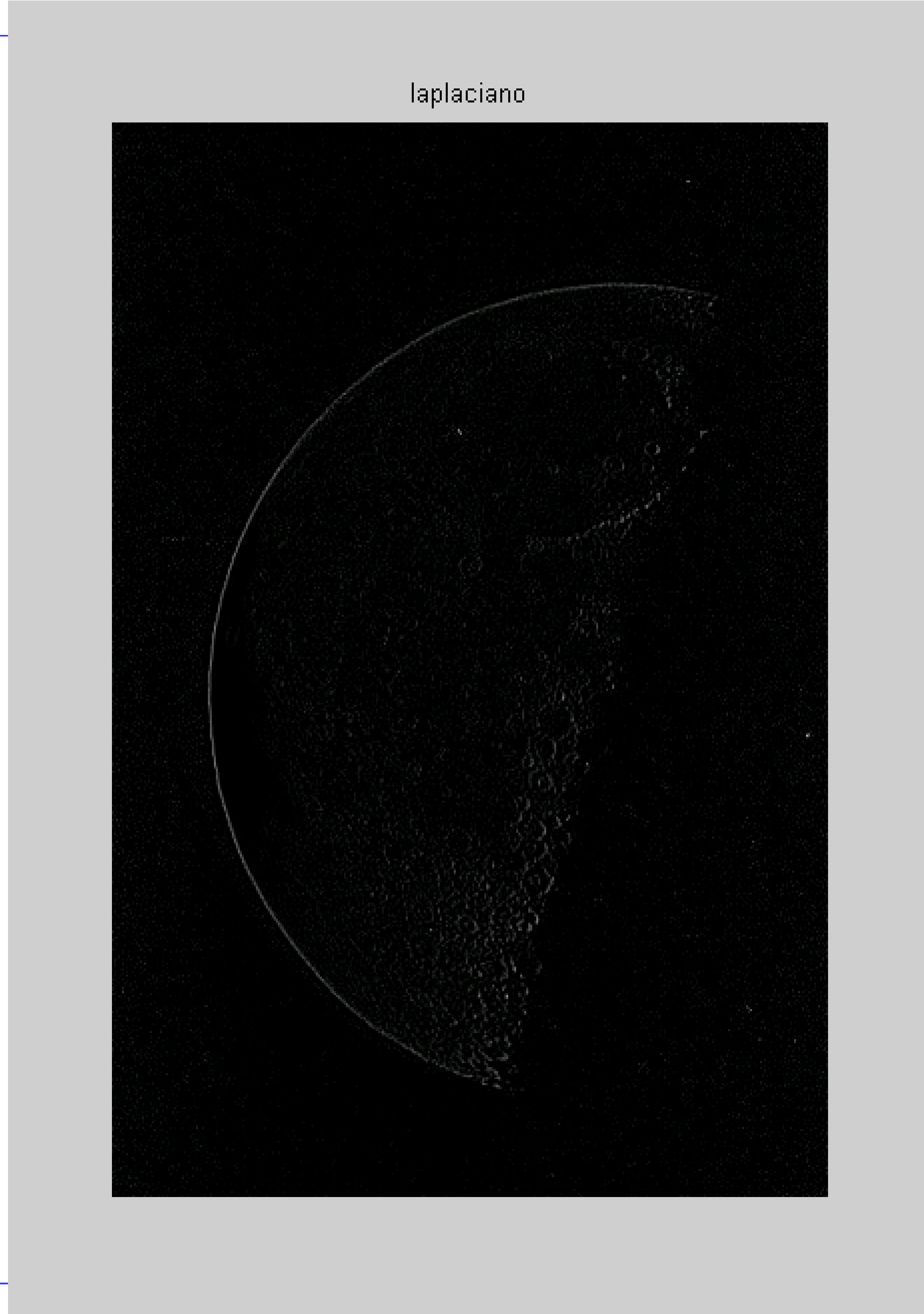

laplaciano

Image enhancement by filtering

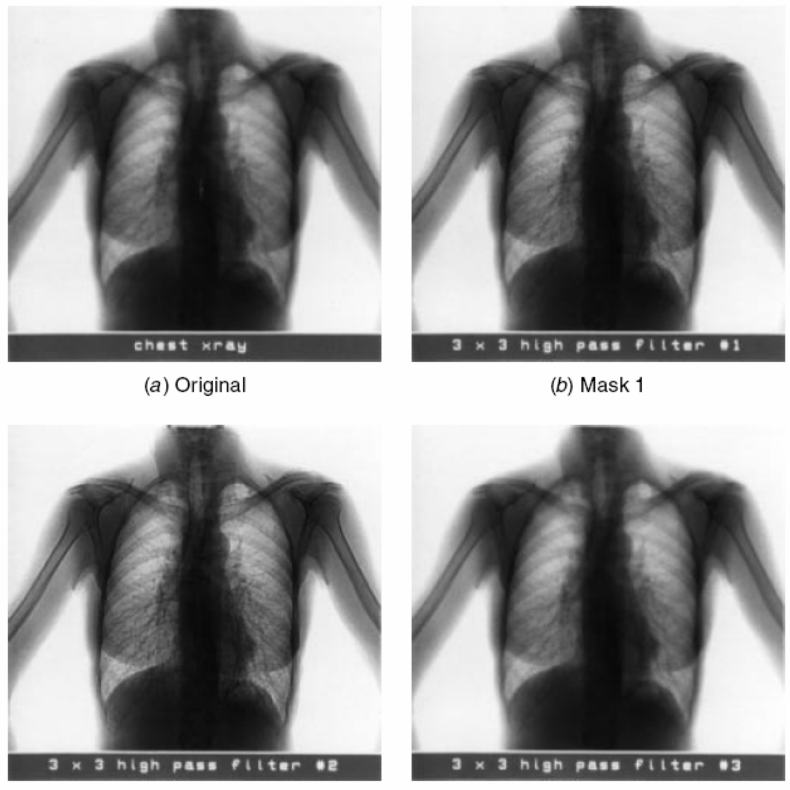
Edge crispening

- Edge crispening is performed by a discrete convolution with a high pass filter

$$\mathbf{H} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

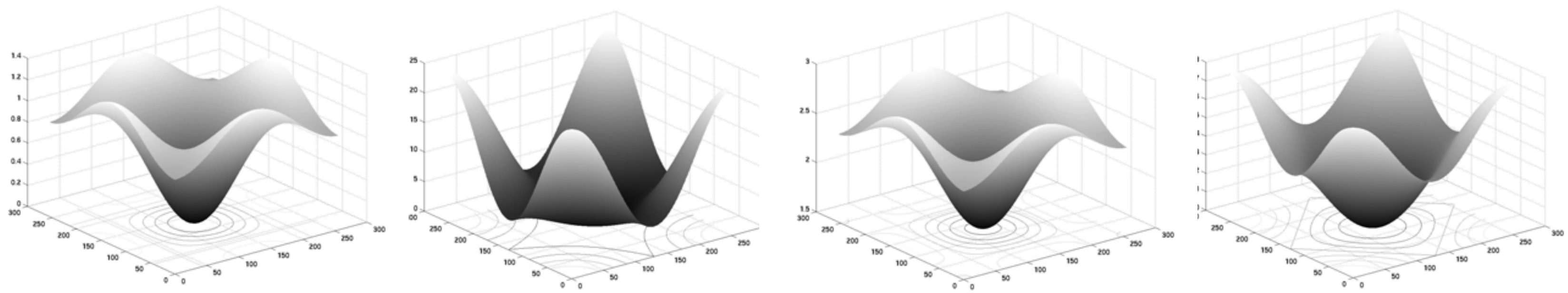
$$\mathbf{H} = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$



(*a*) Original (*b*) Mask 1 (*c*) Mask 2 (*d*) Mask 3

Sharpening

- Goal: “*improve*” image quality
- Solutions
  - increase *relative* importance of details, by increasing the relative weight of high frequency components
    - Increase a subset of high frequencies (non symmetric HP)
    - *High-boost* filter
    - Laplacian gradient
  - The original image is assumed to be available

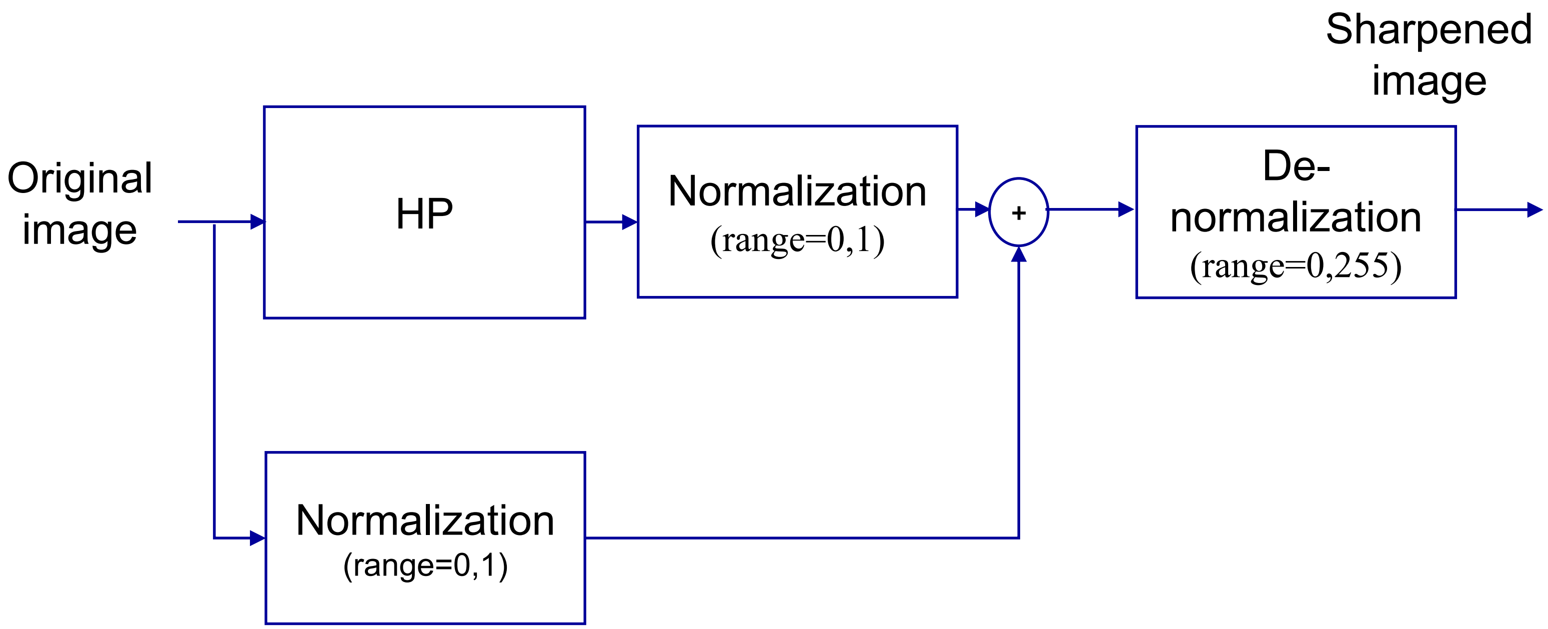


Sharpening Filters

- To highlight fine detail or to enhance blurred details
  - Averaging filters smooth out noise but also blur details
- Sharpening filters *enhance* details
- May also create artifacts (amplify noise)
- Background: Derivative is higher when changes are abrupt

- Categories of sharpening filters
  - Basic highpass spatial filtering
  - High-boost filtering

Sharpening



The normalization step subtracts the mean and scales the amplitude of the resulting image by dividing it for the dynamic range of the resulting image (graylevel values are now in the range 0-1)

For the sharpening to be visible, the sharpened and original images must then be displayed using the same set of graylevel values

High boost

$$I_{highboost} = cI_{original} + I_{highpass} = (cW_{allpass} + W_{highpass}) * I_{original} = W_{highboost} * I_{original}$$

Examples

$$W_{highboost} = cW_{allpass} + W_{highpass} = c\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4+c & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$W_{highboost} = cW_{allpass} + W_{highpass} = c\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8+c & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

High-boost

- Highpass filtered image = Original – lowpass filtered image

- If A is an amplification factor, then:

High-boost = A · original – I · lowpass
= (A-I) · original + I · (original – lowpass)
= (A-I) · original + I · highpass

Unsharp masking (if A=2I)

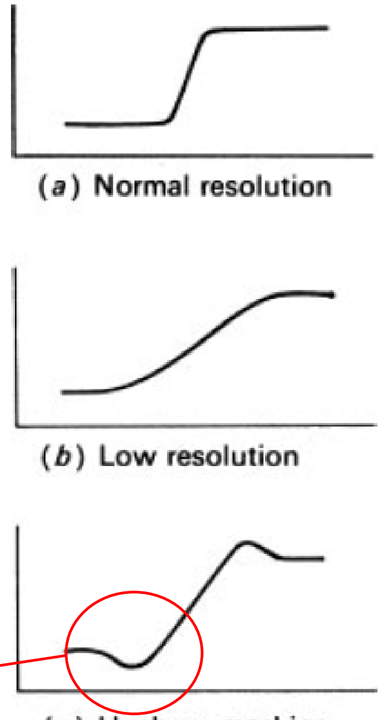
Unsharp masking

$$G(j,k) = \frac{c}{2c-1}F(j,k) - \frac{1-c}{2c-1}F_L(j,k)$$

original

low resolution

c: weighting constant, usually ranging between 3/5 and 5/6



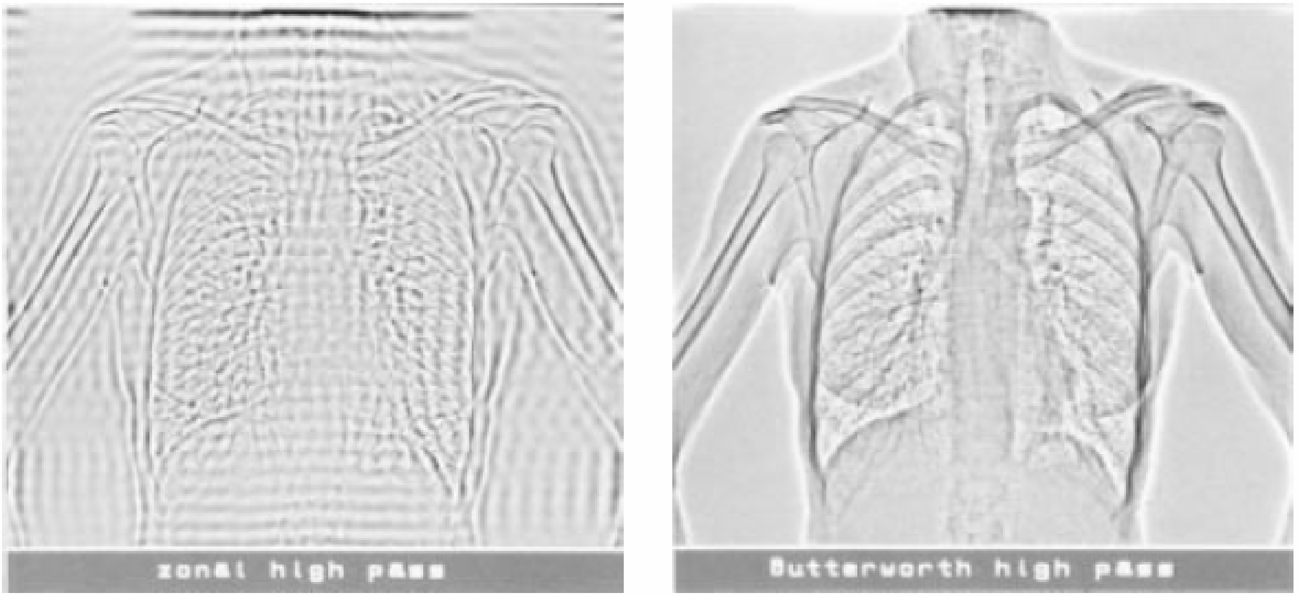
(*a*) Normal resolution

(*b*) Low resolution

(*c*) Unsharp masking

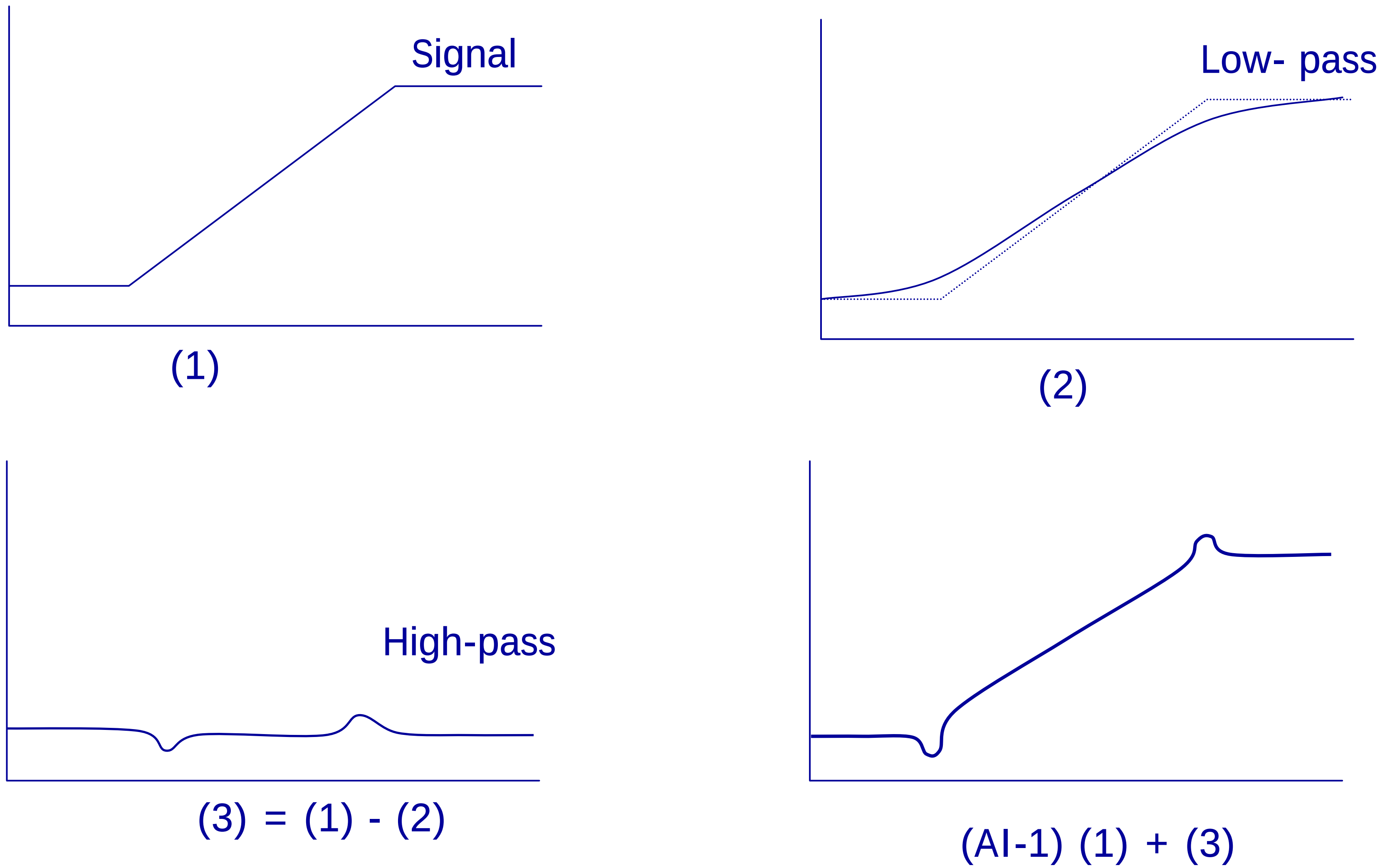
ringing – this can be reduced by using a HP filter with smooth cut-off

Unsharp masking - ringing



(*a*) Zonal filtering (*b*) Butterworth filtering

Unsharp Masking and Sharpening operation



High-boost Filtering

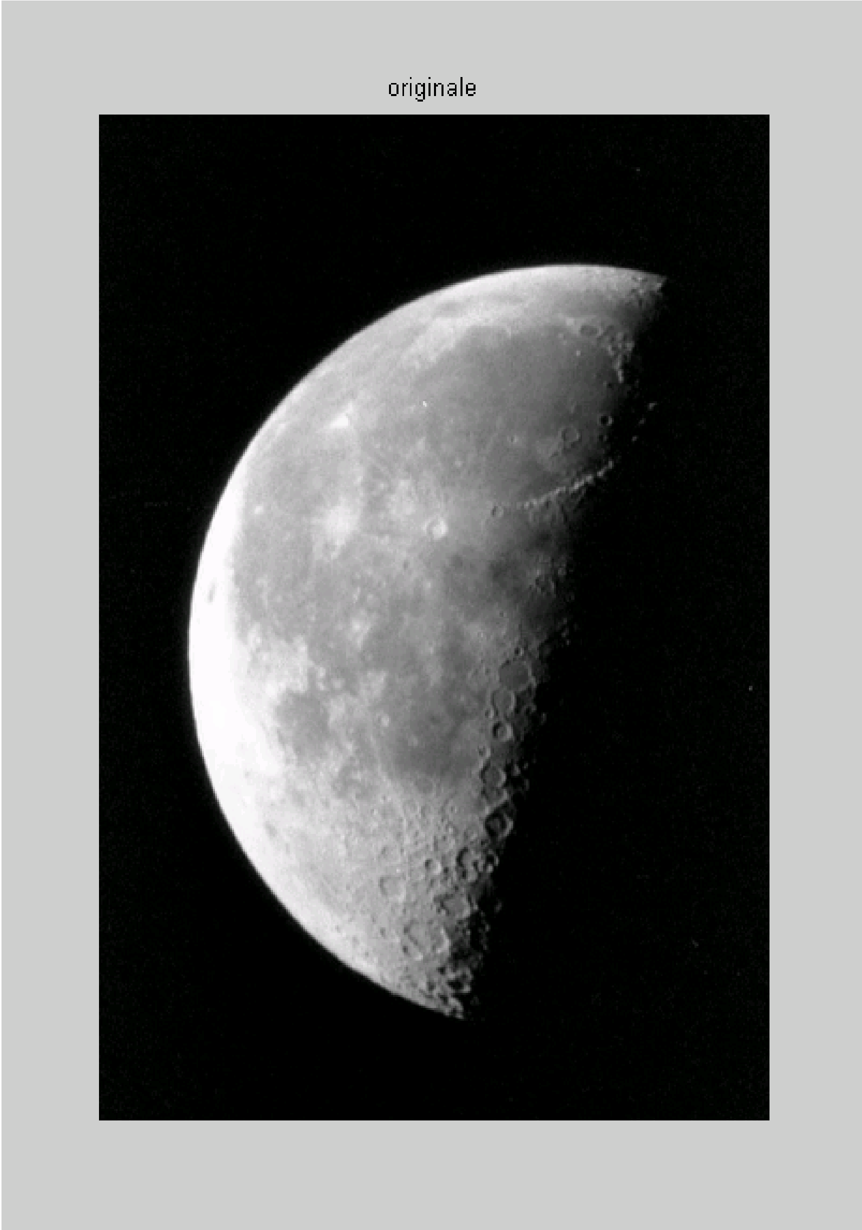
- A=1 : standard highpass result

- A > 1 : the high-boost image looks more like the original with a degree of edge enhancement, depending on the value of A.

$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & w & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

$w=9A-1, \ A \geq 1$

A > 1 → Unsharp masking


originale

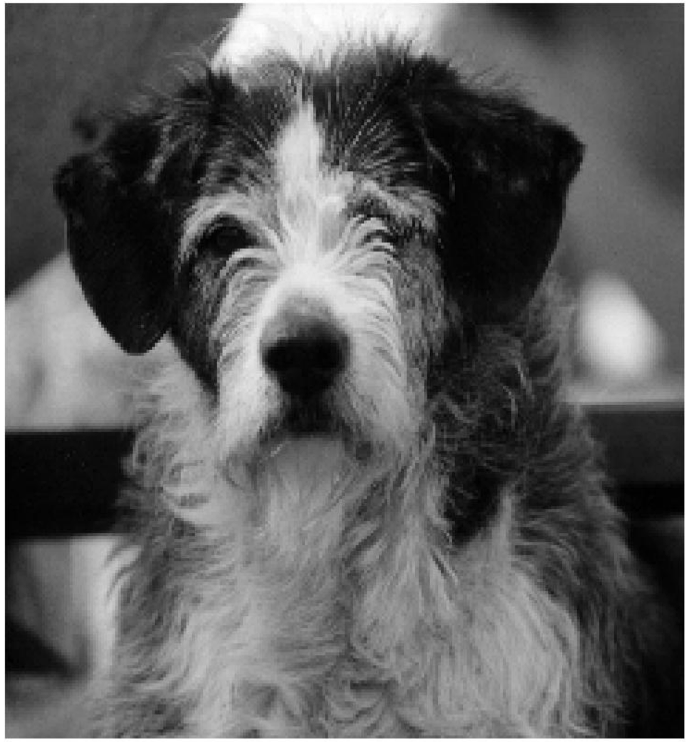

originale+laplaciano

originale+laplaciano

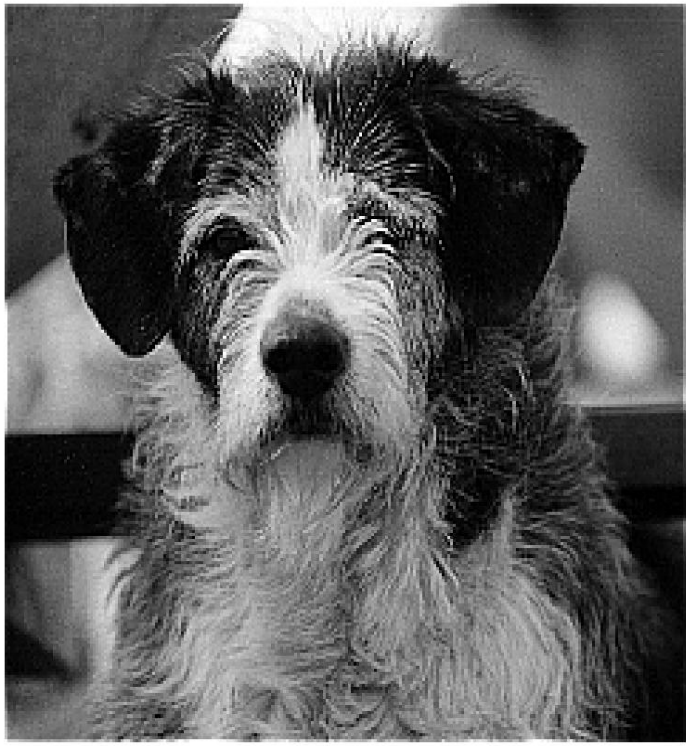
unsharp masking

Sharpening: asymmetric HP

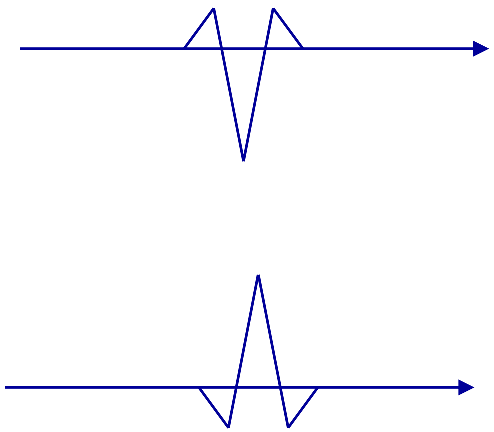
Original image

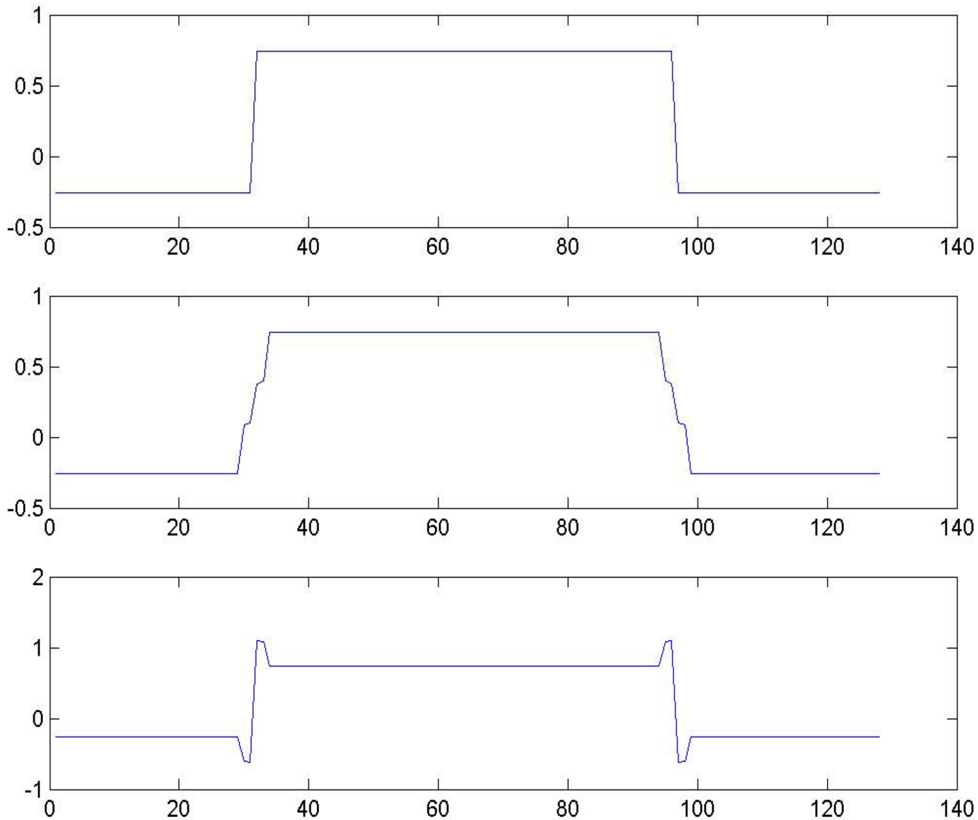


Sharpened image

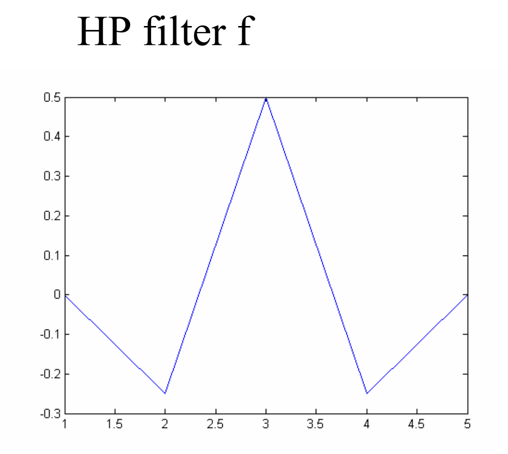


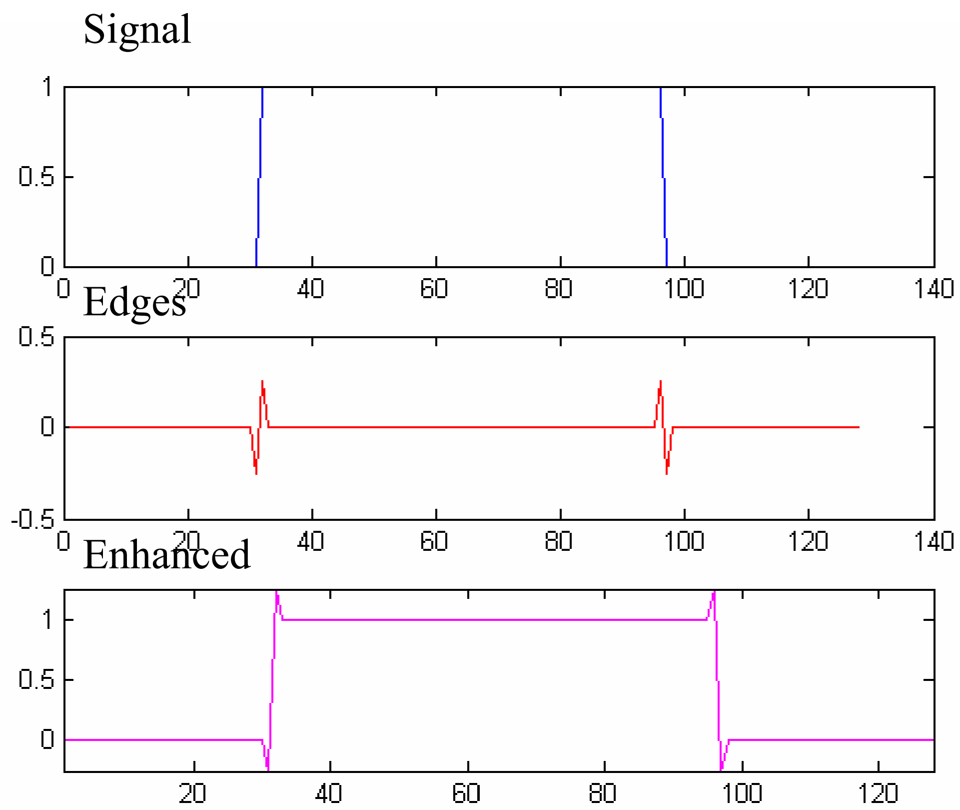
Sharpening: the importance of phase



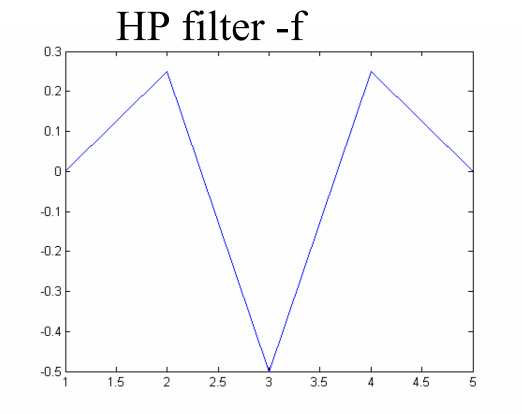


Phase

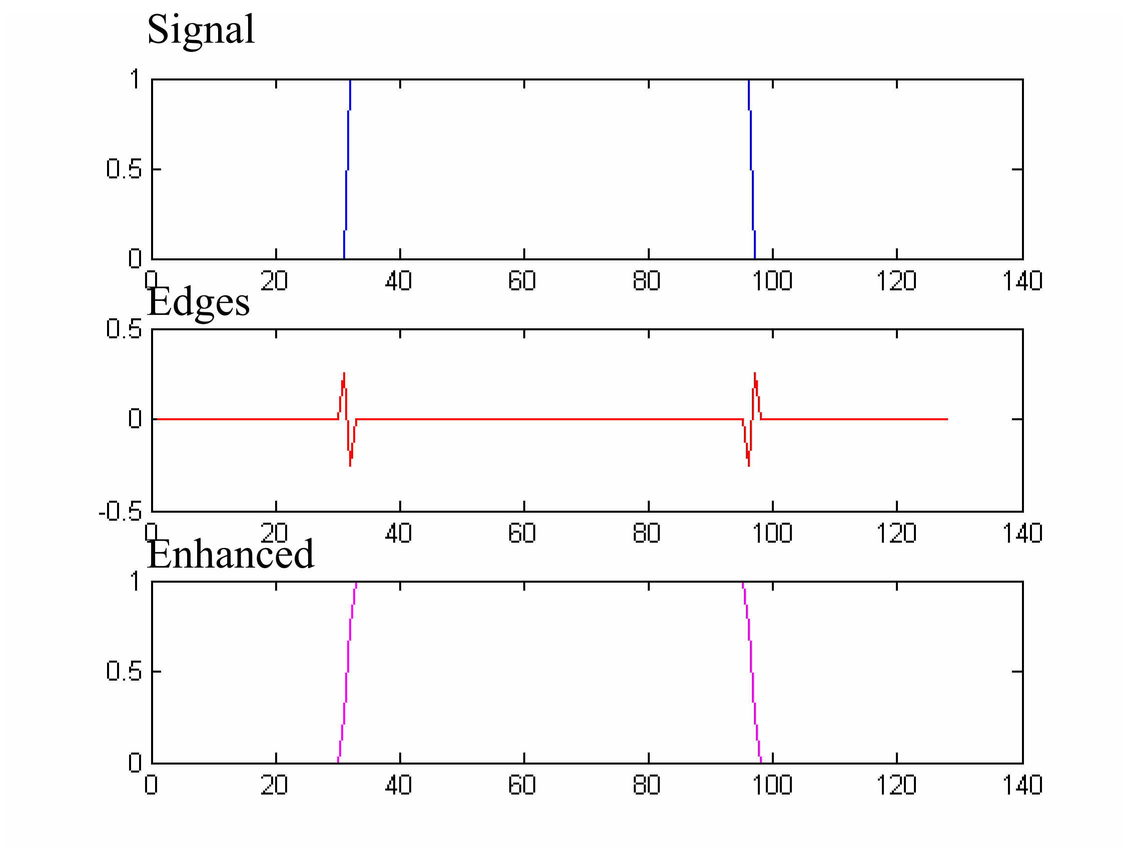




Phase

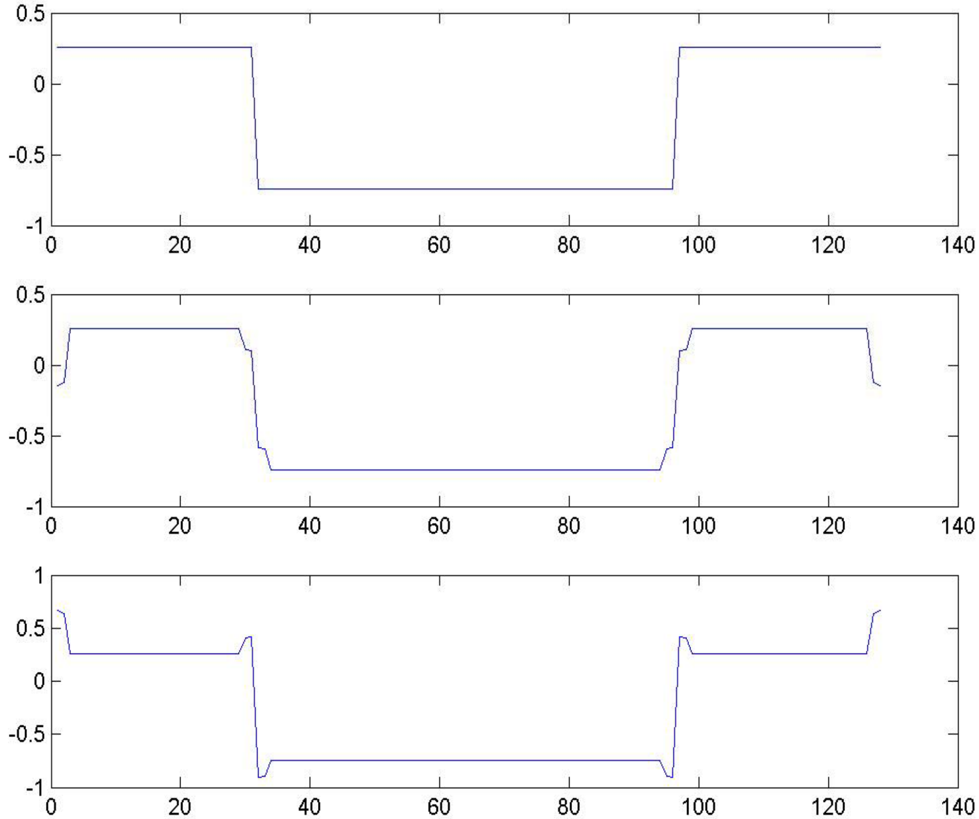


HP filter -f
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

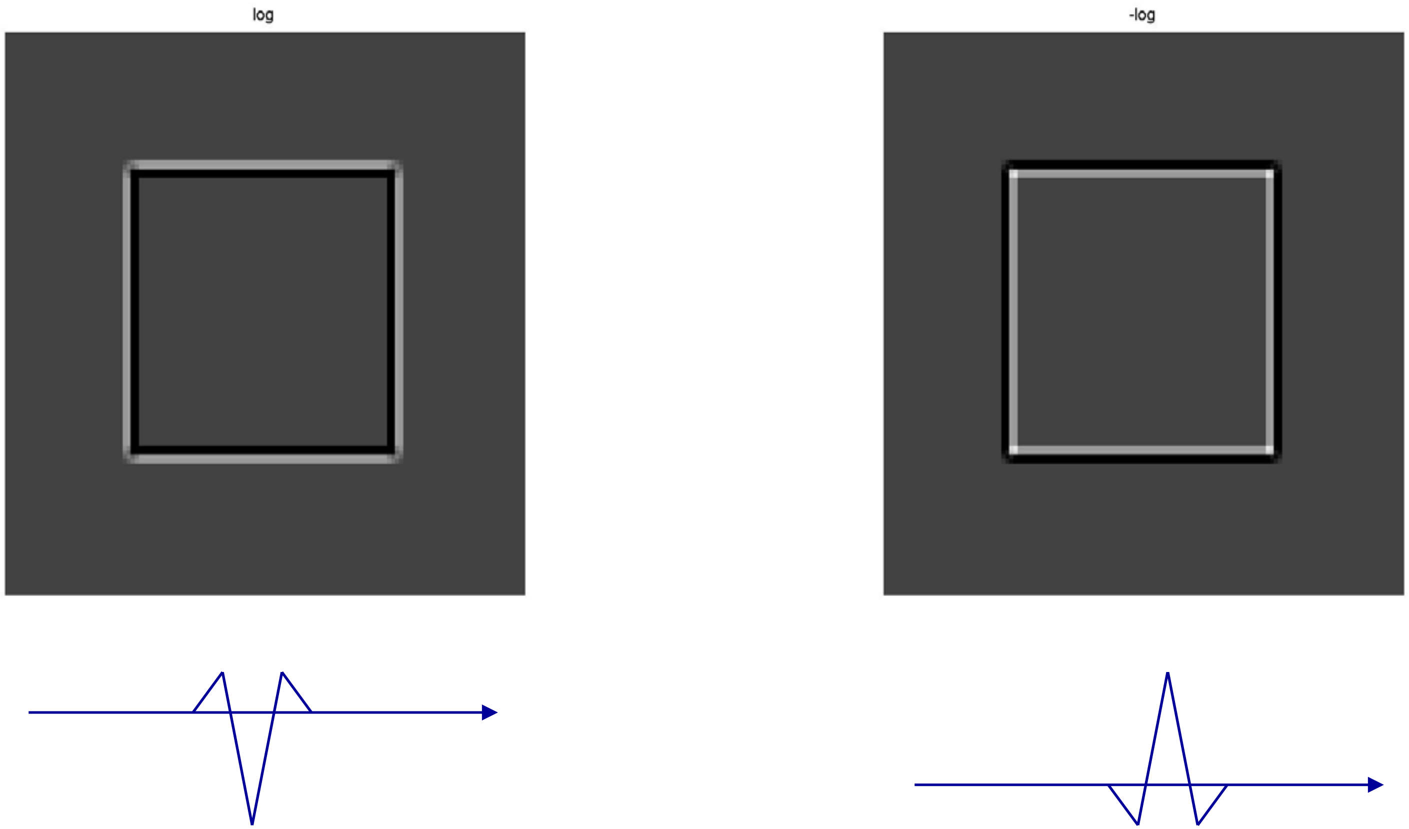


Signal
1
0.5
0
0
20
40
60
80
100
120
140
Edges
0.5
0
-0.5
Enhanced

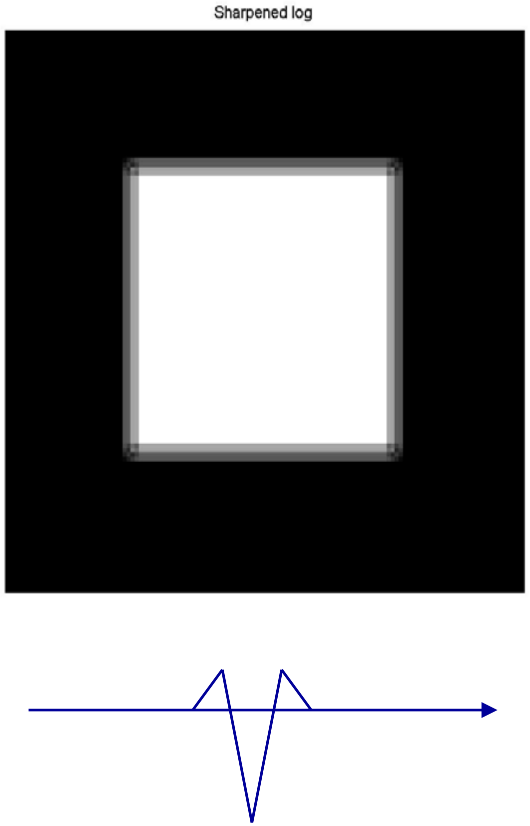
Sharpening: the importance of phase

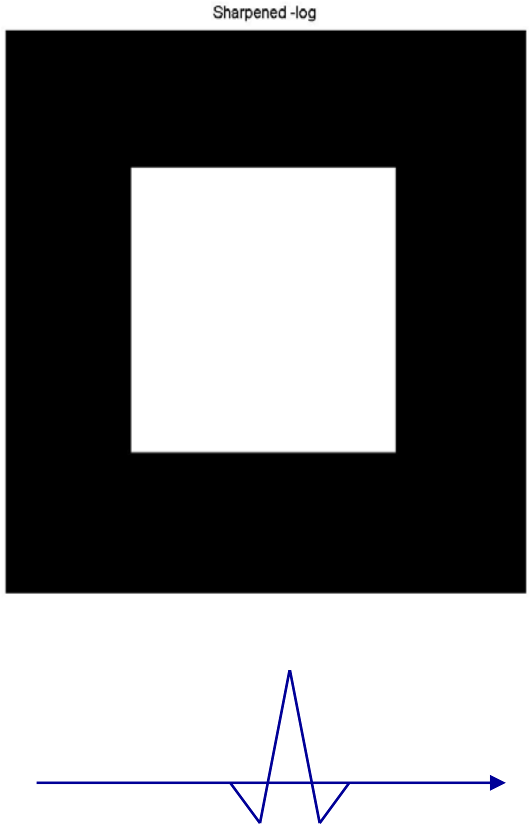


Sharpening: the importance of phase



Sharpening: the importance of phase

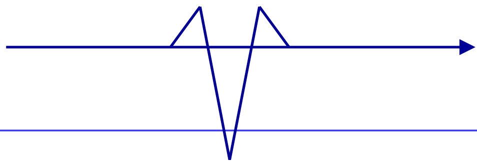




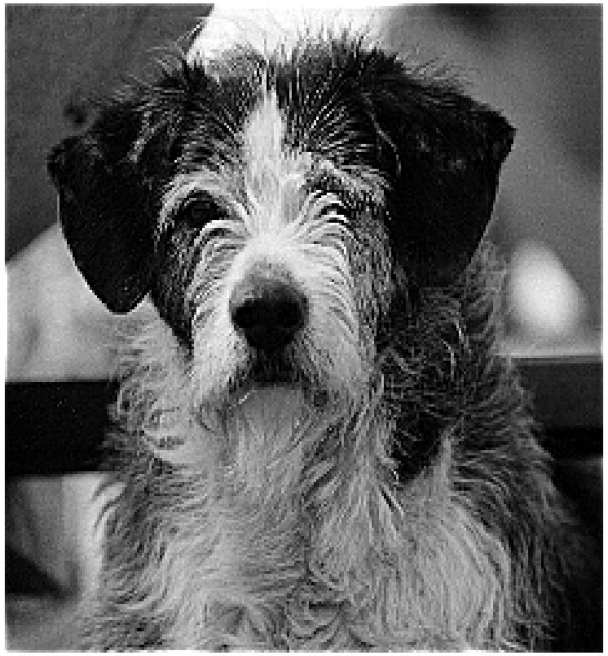
Original image

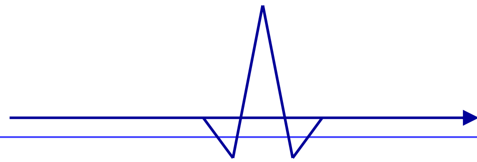
Back to the natural image

Sharpened image



Sharpened image





Edge crispening by statistical differencing

$$G[j,k] = \frac{F[j,k]}{D[j,k]}$$

F[j,k]: original image
G[j,k]: enhanced image
D[j,k]: local standard deviation

$$D[j,k] = \frac{1}{W}\left[\sum_{m=j-w}^{j+w}\sum_{n=j-w}^{j+w}\left(F[m,n]-M[m,n]\right)^2\right]^{\frac{1}{2}}$$

$$W = 2w+1$$

$$M[m,n] = \frac{1}{W^2}\sum_{m=j-w}^{j+w}\sum_{n=j-w}^{j+w}F[m,n]$$

local mean value

The enhanced image is increased in amplitude with respect to the original at pixels that *deviate significantly from their neighbors*, and is decreased in relative amplitude elsewhere.

Statistical differencing: Wallis operator

- The enhanced image is forced to a form with desired first- and second-order moments.
- *Wallis operator* is defined by

$$G(j,k) = [F(j,k) - M(j,k)]\frac{A_{\max} D_d}{A_{\max} D(j,k) + D_d} + [pM_d + (1-p)M(j,k)]$$

  – where
    - M_d and D_d represent desired average mean and standard deviation factors,
    - A_{max} is a maximum gain factor that prevents overly large output values when D[j,k] is small
    - $0 \le p \le 1$ is a mean proportionality factor controlling the background flatness of the enhanced image

Statistical differencing: Wallis operator

- Alternative expression

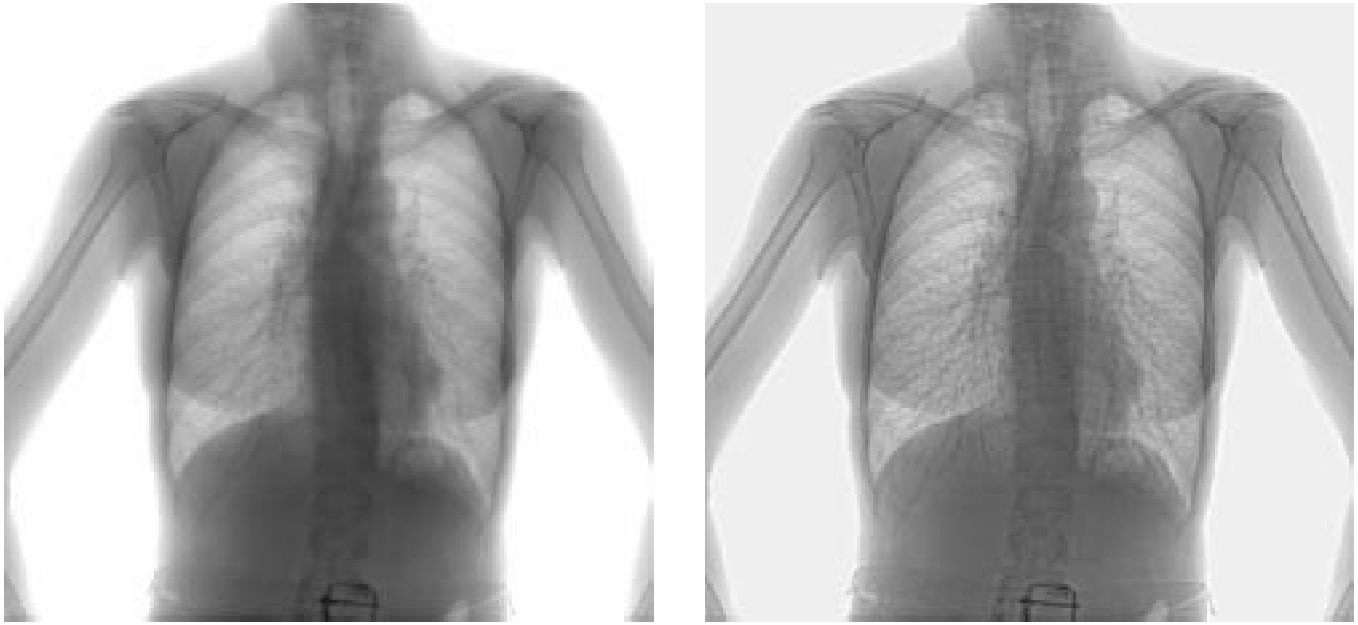
$$G(j,k) = [F(j,k) - M(j,k)]A(j,k) + B(j,k)$$

  - where A[j,k]is a *spatially dependent gain factor* and B[j,k] is a *spatially dependent background factor*
  - It is convenient to specify the desired average standard deviation D_d such that the spatial gain ranges between maximum A_{max} and minimum A_{min} limits. This can be accomplished by setting D_d to the value

$$D_d = \frac{A_{\min} A_{\max} D_{\max}}{A_{\max} - A_{\min}}$$

Dmax is the maximum value of D[j,k]

Wallis enhancement



(a) Original (b) Wallis enhancement

Wallis enhancement

(*a*) Original

(*b*) Mean, 0.00 to 0.98

(*c*) Standard deviation, 0.01 to 0.26

(*d*) Background, 0.09 to 0.88

Wallis enhancement

(*e*) Spatial gain, 0.75 to 2.35 (*f*) Wallis enhancement, – 0.07 to 1.12

FIGURE 10.4-5. Wallis statistical differencing on the `bridge` image for $M_d = 0.45$, $D_d = 0.28$, $p = 0.20$, $A_{max} = 2.50$, $A_{min} = 0.75$ using a 9×9 pyramid array.

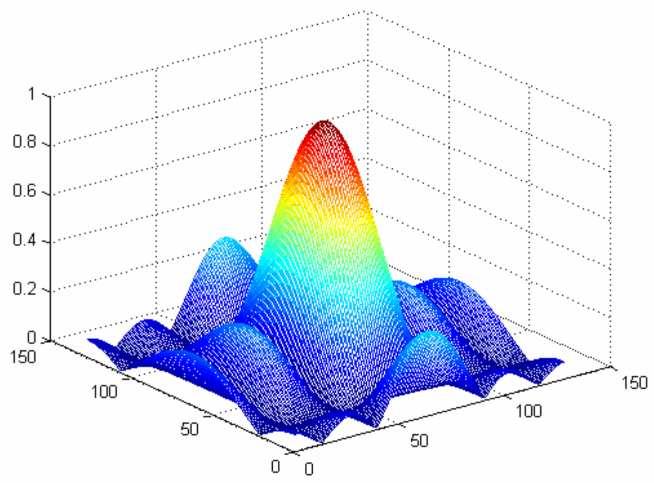
Color image enhancement

- Usually performed after transformation to an opposed channels model featuring lower correlation
  - Lab, Luv, YIQ, YCbCr
- Often enhancing the luminance, or lightness, component is enough
  - Low-pass behaviour of color vision
    - Because of the high-spatial-frequency response limitations of human vision, edge crispening of the chroma or chrominance components may not be perceptible
- The channels are processed independently and then “recombined” after enhancement
  - Care must be taken to preserve the average value of the three channels in every point to avoid color artifacts

Color image sharpening

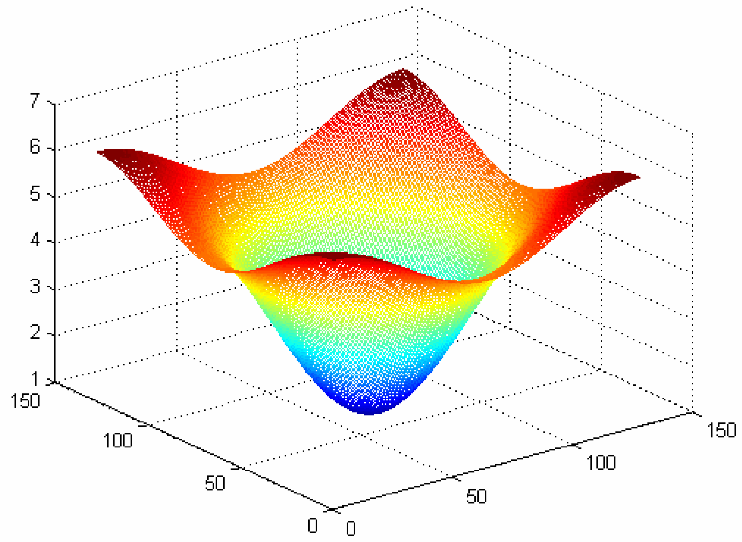
1. RGB to YCbCr
2. Smoothing the three components, respectively
3. Unsharpening of the lightness (Y) component
4. Reconstruction

$$h = 0{,}04 \times \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix} \quad \text{low pass}$$



unsharpening filter

$$h_u = \begin{matrix} -0.1667 & -0.6667 & -0.1667 \\ -0.6667 & 4.3333 & -0.6667 \\ -0.1667 & -0.6667 & -0.1667 \end{matrix}$$



Example

smoothed Y

smoothed chrominances

Smoothed chrominances

all smoothed

sharpened Y

Unsharped lightness