Image Analysis in Neuroinformatics

Removal of Artifacts

Pavan Ramkumar Xavier Mínguez Bernal

Table of contents

- Characterization of Arfitacts
  - Statistics
  - Diferent types
- Matrix Representation of Images
  - Images
  - Transforms
  - Convolutions
- Denoising Techniques
  - Mutliframe Averaging
  - Spatial convolutions and order statistics
  - Frequency domain filters
  - Optimal filtering: Wiener

Characterization of Arfitacts

Characterization of Artifacts I

Random noise

- Pdf

$$p_\eta(\eta)$$

- Mean

$$\mu_\eta = E[\eta] = \int_{-\infty}^{\infty} \eta p_\eta(\eta)\, \delta\eta$$

- Variance

$$\sigma^2{}_\eta = E\left[(\eta - \mu_\eta)^2\right] = \int_{-\infty}^{\infty} (\eta - \mu_\eta)^2 p_\eta(\eta)\, \delta\eta$$

$$\sigma^2{}_\eta = E\left[\eta^2\right] - E[\eta]^2$$

Characterization of Artifacts II

- Image

$$g(x,y) = f(x,y) + \eta(x,y)$$

Detected image

Image frame

Noise
(typically additive)

$$E[g] = \mu_g = \mu_f + \mu_\eta$$

$$E[(g - \mu_g)^2] = \sigma^2{}_g = \sigma^2{}_f + \sigma^2{}_\eta$$

Characterization of Artifacts III

- Statistical expectation
  - Estimated mean (M observations)

$$\mu_f(x_1, y_1) = \lim_{M\to\infty} \frac{1}{M} \sum_{k=1}^{M} f_k(x_1, y_1)$$

  - Autocorrelation function

$$r_f(x_1, x_1+\alpha, y_1, y_1+\beta) = E[f^*(x_1, y_1) f(x_1+\alpha, y_1+\beta)]$$

$$\check{r}(\alpha, \beta) = \frac{1}{XY} \sum_{x_1=0}^{X} \sum_{y_1=0}^{Y} f^*(x_1, y_1) f(x_1+\alpha, y_1+\beta)$$

$$r_f(0,0) \Rightarrow energy$$

Characterization of Artifacts IV

- Stationarity in strict sense

  Statistics not affected by a shift in time or space

- Stationarity in wide sense

  Constant mean and autocorrelation depends only upon the shift in time or space

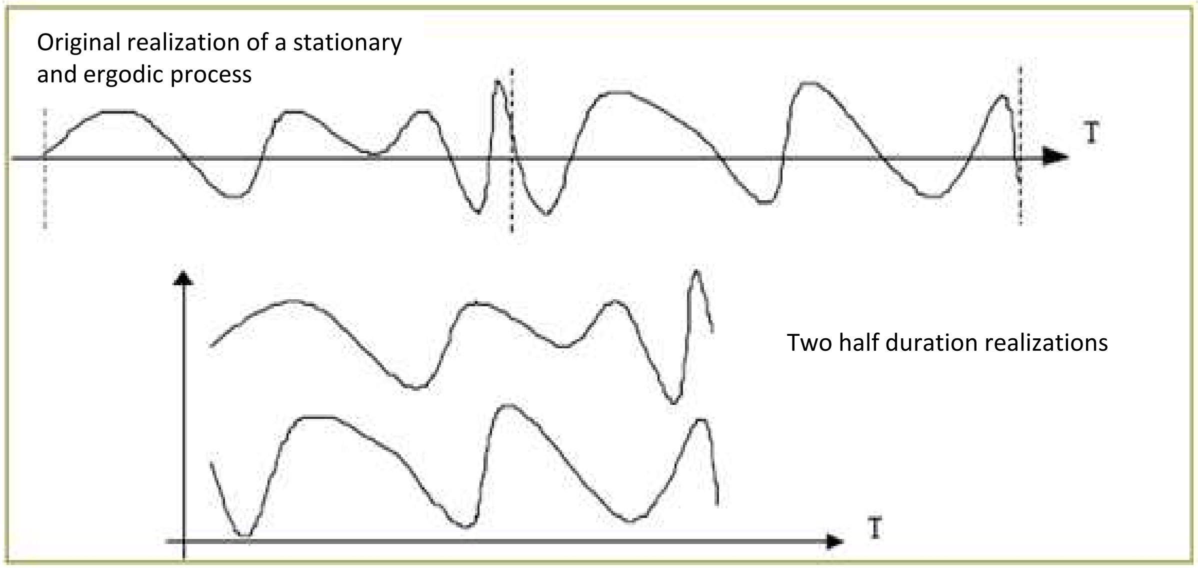
$$\mu_f(k) = \mu_f$$

$$r_f(t_1, t_1 + \tau) \Rightarrow r_f(\tau)$$

$$r_f(x_1, x_1 + \alpha, y_1, y_1 + \beta) \Rightarrow r_f(\alpha, \beta)$$

Characterization of Artifacts V

- Ergodicity
  - Temporal statistics independent of the sample observed
  - Statistics may be computed from a single observation

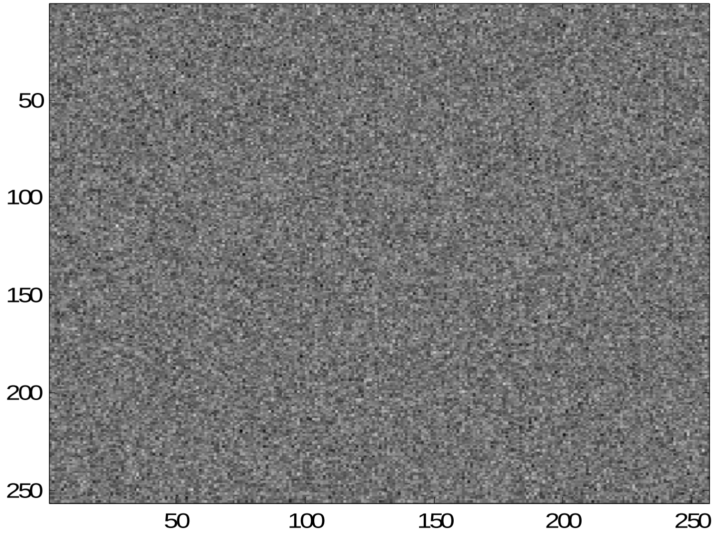


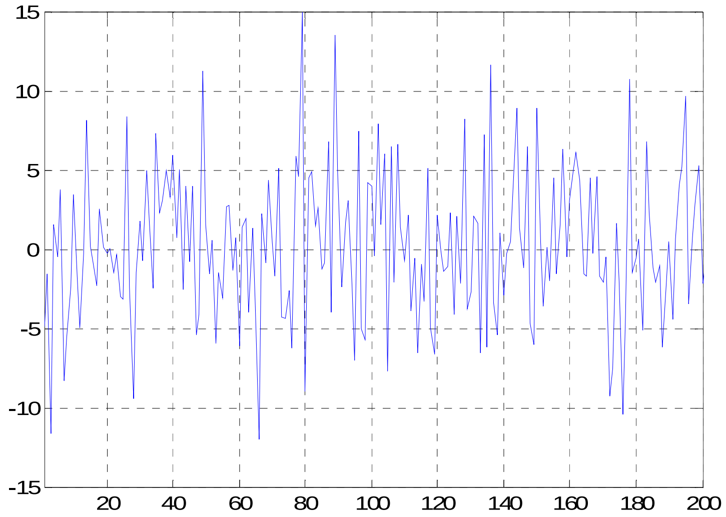
Gaussian noise I

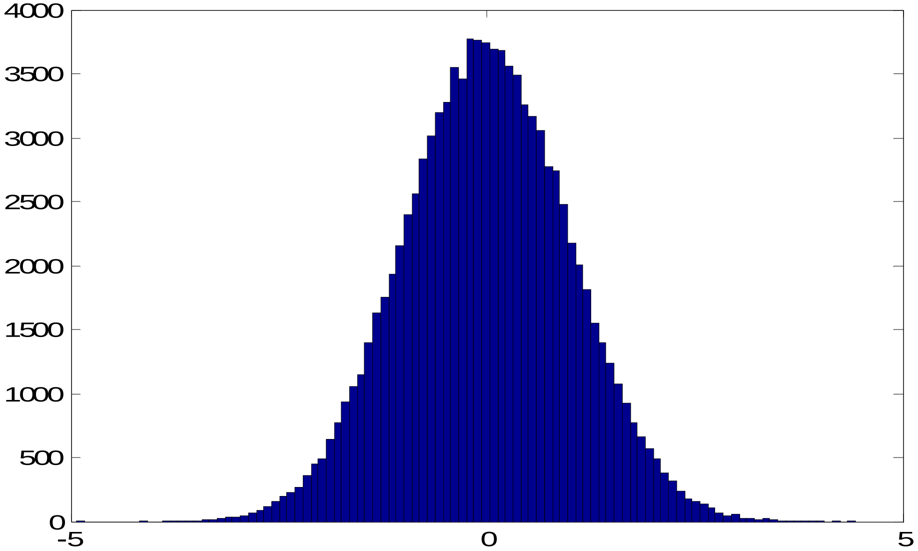
- Completely specified by the mean and variance
- Important ⇨ *central limit teorem*
- Termal noise ⇨ electronic components

$$p(x) = \frac{1}{\sqrt{2\pi\sigma_x^{\,2}}}\, e^{-\left(\frac{(x-\mu_x)^2}{2\sigma^2{}_x}\right)}$$

Gaussian noise II



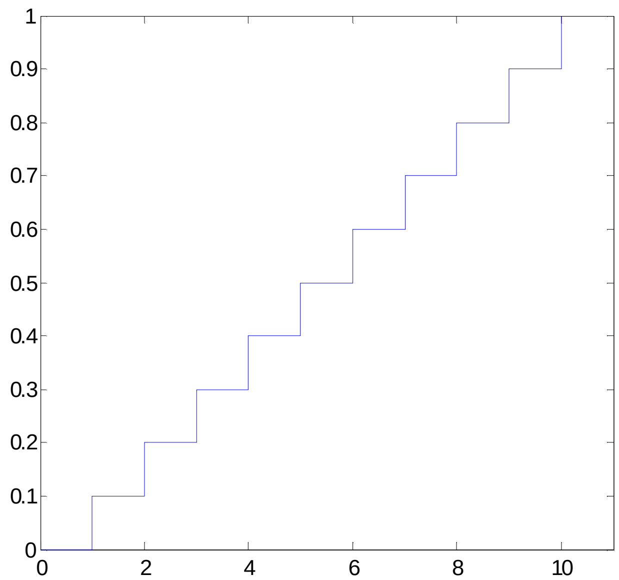


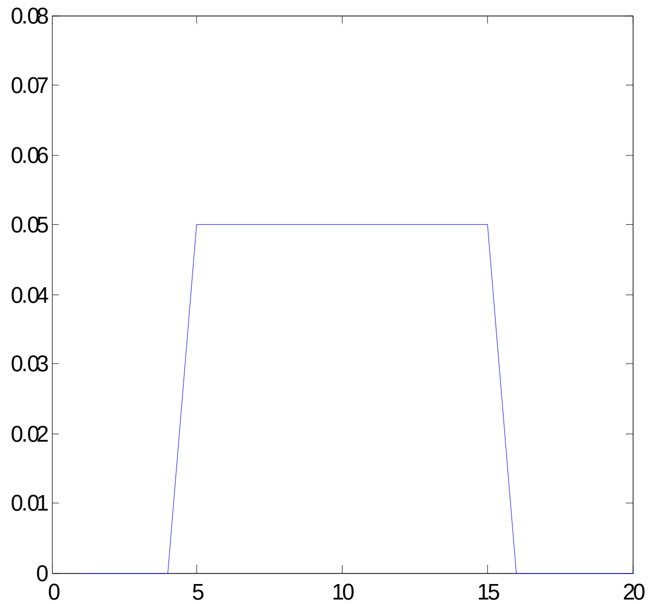


Uniform distributed noise

- Quantization noise

$$N \propto \left(\frac{dinamic\ range}{2^Q}\right)^2$$

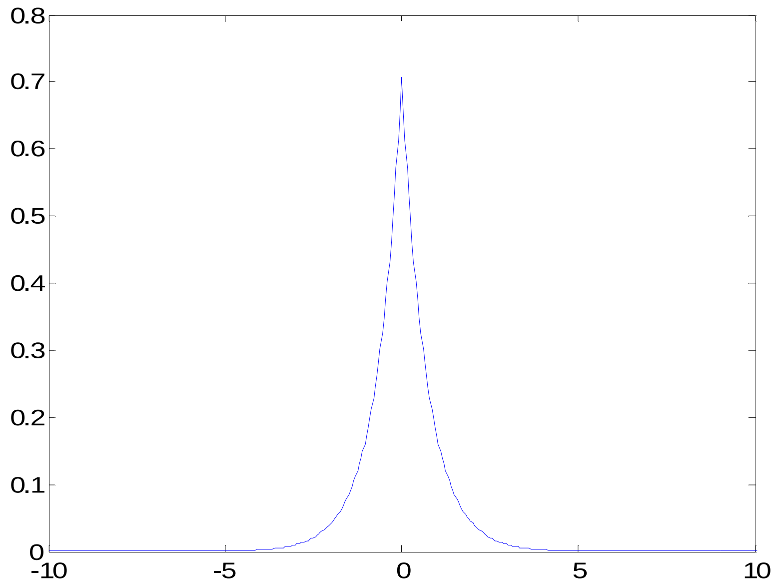




Laplacian distributed noise

- Errors in linear prediction

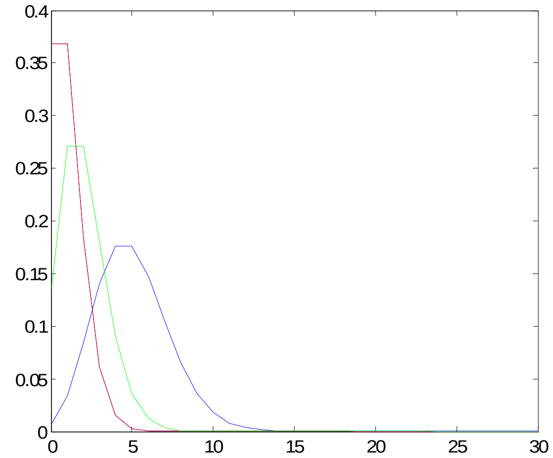
$$p_x(x) = \frac{1}{\sqrt{2\pi\sigma_x{}^2}} e^{-\left(\frac{\sqrt{2}\,|x - \mu_x|}{\sigma_x}\right)}$$



Poisson noise

- Quantum noise, photon noise
- Signal dependant
- Systems in low-light conditions

$$p(k) = \frac{\mu^k}{k!} e^{-\mu}$$



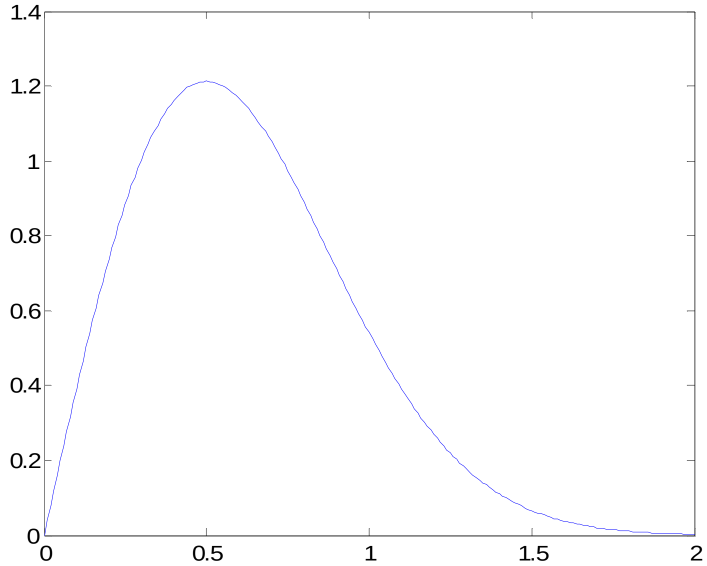
- Degraded image

$$p(g_0(m,n)|f(m,n),\lambda) = \frac{[\lambda f(m,n)]^{g_0(m,n)}}{g_0(m,n)!} e^{-\lambda f(m,n)}$$

Speckle noise

- Caused by roughness surface
- Signal depedant
- Rayleigh distributed

$$p_x(x) = \frac{2}{b}(x-a)\,e^{-\left(\frac{(x-a)^2}{b}\right)}u(x-a)$$



Other types of noise and artifact

- Structured noise
  - Power line interference
  - Grid artifact ⇨ parallel periodic strips
  - Surgical implants
- Physiological interference
  - Effect of breathing
  - Cardiovascular activity
- Others
  - Salt and pepper noise
  - Shot noise

Matrix Representation of Images

Matrix Representation of Images

- $F = \{F(m,n), m = 0 \ldots M-1, n + 0 \ldots N-1\}$

- Non-negativity and Upperbound Constraint
  - $Fmin <= F(m,n) <= Fmax$

- Finite energy
  - $ЄF = \Sigma\Sigma F2(m,n) <= Emax$

- Smoothness
  - $F(m,n) – mean(Fnbd(m,n)) <= S$

Vectorization

- Useful representation prior to application of transformation, estimation, optimization
    - F = [1 2
    - 3 4]
  - Row ordering
    - $f = [1\ 2\ 3\ 4]^T$
  - Column ordering
    - $f = [1\ 3\ 2\ 4]^T$
  - F is MxN, f is MNx1

Some definitions

- Energy $\epsilon = f^T f = Tr[ff^T]$
- Mean $\mu = E[f]$
- Covariance $\sigma = E[(f - \mu)(f - \mu)^T] = E[ff^T] - \mu\mu^T$
- Auto Correlation or scatter matrix

$$\Phi = E[ff^T]$$

- For 2 images f,g we can define:
- Uncorrelatedness, Orthogonality, Statistical Independence (look up last week's slides)

Matrix Representation of Transforms I

- 1D Transforms

- A signal may be represented as a linear combination of orthogonal basis functions

$$f(t) = \sum_{k=0}^{\infty} a_k \varphi_k(t)$$

$$\int_{to}^{to+T} \varphi_k(t)\varphi_l^*(t)dt = 1, k = l, 0, otherwise$$

$$a_k = \int_{to}^{to+T} f(t)\varphi_k^*(t)$$

Matrix Representation of Transforms II

- 1D Transforms
- The set {ak} represents a 'Transform space'
- If a 1D signal is sampled at N points, the transform may be represented as matrix multiplication

$$F = Lf$$

$$f = L^* f$$

$$L(k,n) = \varphi_k(n)$$

Matrix Representation of Transforms III

- 2D Transforms
- For an NxN image f(m,n) and its transform F(k,l) are related by

$$F(k,l) = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f(m,n) \varphi(m,n,k,l)$$

$$f(m,n) = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) \psi(m,n,k,l)$$

- ϕ(m,n,k,l) is the forward kernel and
- Ψ(m,n,k,l) is the inverse kernel

2D Transforms I

- If ϕ(m,n,k,l) = ϕ1(m,n) ϕ2(k,l), ϕ is said to be separable
- If ϕ1(m,n) = ϕ2(m,n), it is said to be symmetric
- If ϕ is symmetric and separable, the 2D transform can be computed as 2 1D transforms sequentially

$$F_1(m,l) = \sum_{n=0}^{N-1} f(m,n)\varphi(n,l)$$

$$F(k,l) = \sum_{m=0}^{N-1} F_1(m,l)\varphi(m,k)$$

2D Transforms II

- Example of separable, symmetric kernels:
- 2D DFT

$$\varphi(m,n,k,l) = \exp\left[-j\frac{2\pi}{N}(mk+nl)\right]$$

$$\varphi(m,n,k,l) = \exp\left[-j\frac{2\pi}{N}mk\right]\exp\left[-j\frac{2\pi}{N}nl\right]$$

- f is the NxN image, W is a symmetric NxN matrix and F is the NxN 2D DFT

$$W(k,m) = \exp\left[-j\frac{2\pi}{N}km\right]$$

Matrix Representation of Convolutions I

- 1D Convolution with Causal IIR filter

$$g(n) = \sum_{\alpha=0}^{n} f(\alpha) h(n-\alpha)$$

- Can be represented in matrix form as

$$\begin{pmatrix} g(0) \\ g(1) \\ \vdots \\ g(N) \end{pmatrix} = \begin{pmatrix} h(0) & 0 & \cdots & 0 \\ h(1) & h(0) & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h(N) & h(N-1) & \cdots & h(0) \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(N) \end{pmatrix}$$

Matrix Representation of Convolutions II

- 1D Convolution with Non-Causal FIR filter

$$g(n) = \sum_{\alpha=0}^{M} f(\alpha + n - \frac{M}{2}) h(\frac{M}{2} - \alpha)$$

- Can be represented in matrix form as

$$\begin{pmatrix} g(0) \\ g(1) \\ \vdots \\ g(N) \end{pmatrix} = \begin{pmatrix} h(\frac{M}{2}) & h(\frac{M}{2}-1) & \cdots & h(0) & \cdots & h(1-\frac{M}{2}) & h(-\frac{M}{2}) & 0 & \cdots & 0 \\ 0 & h(\frac{M}{2}) & h(\frac{M}{2}-1) & \vdots & h(0) & & h(1-\frac{M}{2}) & h(-\frac{M}{2}) & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & & & & \ddots & \vdots \\ 0 & \cdots & h(\frac{M}{2}) & h(\frac{M}{2}-1) & \cdots & h(0) & & & \cdots & h(-\frac{M}{2}) \end{pmatrix} \begin{pmatrix} f(-\frac{M}{2}) \\ f(1-\frac{M}{2}) \\ \vdots \\ f(N+\frac{M}{2}) \end{pmatrix}$$

Matrix Representation of Convolutions III

- Periodic or circular convolution for finite, periodic f(n) and h(n)

$$g(n) = \sum_{\alpha=0}^{n} f(\alpha) h([n-\alpha] \bmod N)$$

- Circular convolution of 2 signals of duration N is of length N (obtained also by IDFT of product of 2 DFTs)
- Linear convolution of 2 signals of duration N is of length 2N-1
- Circular and linear convolution can be made identical by zero padding the signals to length 2N-1

Matrix Representation of Convolutions IV

- Circular convolution in matrix form

$$\begin{pmatrix} g(0) \\ g(1) \\ \vdots \\ g(N-2) \\ g(N-1) \end{pmatrix} = \begin{pmatrix} h(0) & h(N-1) & \cdots & h(2) & h(1) \\ h(1) & h(0) & \ddots & h(3) & h(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h(N-2) & h(N-1) & \ddots & h(0) & h(N-1) \\ h(N-1) & h(N-2) & \cdots & h(1) & h(0) \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-2) \\ f(N-1) \end{pmatrix}$$

- This is called a circulant matrix.
- Important property: Diagonalized by DFT

2D Convolution I

- 2D LSI convolution

$$g(m,n)=\sum_{\alpha=0}^{M-1}\sum_{\beta=0}^{N-1} f(\alpha,\beta)h(m-\alpha,n-\beta)$$

- 2D Circular convolution

$$g(m,n)=\sum_{\alpha=0}^{M-1}\sum_{\beta=0}^{N-1} f(\alpha,\beta)h([m-\alpha]\bmod M,[n-\beta]\bmod N)$$

2D Convolution II

- Matrix representation

$$\mathbf{g} = \mathbf{hf}$$

- Representation of **h** as block circulant matrix
- Insert eqn 3.102 (pg. 221)
- Submatrices $\mathbf{h_m}$ are given by
- Insert eqn. 3.103 (pg. 221)
- Each $\mathbf{h_m}$ is circulant and **h** is block circulant

Denoising Techniques

Mutliframe Averaging I

- Gated ensemble averaging
- Successive frames which are gated (phase locked) to a 'physiological state' in a recurring cycle are averaged to remove additive noise
- 'Recurring physiological state' is not very clearly defined as far as the brain is concerned!

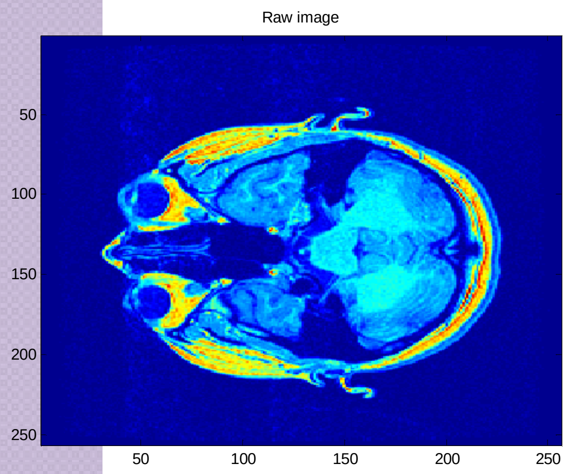
$$g = f + \eta$$

- Law of large numbers suggests

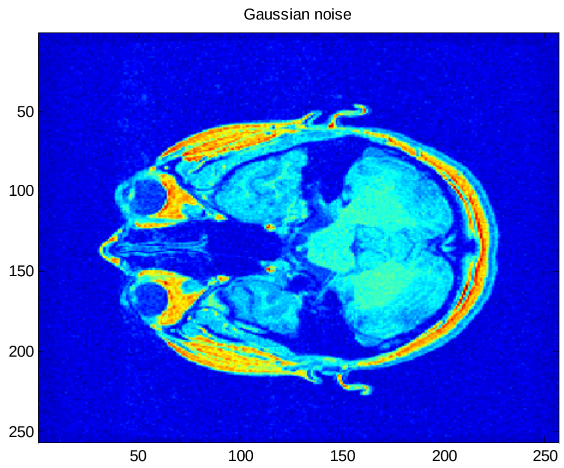
$$\sigma(g) = 1/\mathrm{sqrt}(N)\ \sigma(\eta)$$

Multiframe Averaging II

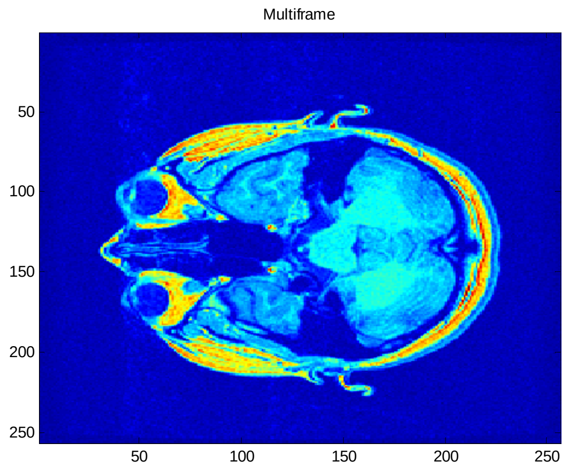
5 zero mean Gaussian noise (σ=5) frames were added to an anatomical image and averaged



Raw image



Gaussian noised image (MSE = 24.87)



Multiframe avgd. Image (MSE = 4.85)

Spatial Domain Local Statistics Filters

- Cannot rely on ensemble of images to obtain properties of the image
- A (spatially) moving window is used to gather local statistics
- If the statistic is a linear combination of intensities in the nbd, it can be expressed as a LSI convolution

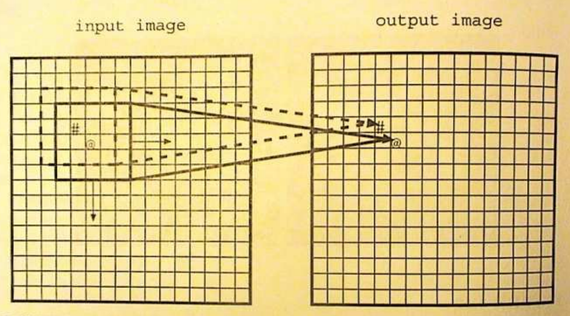
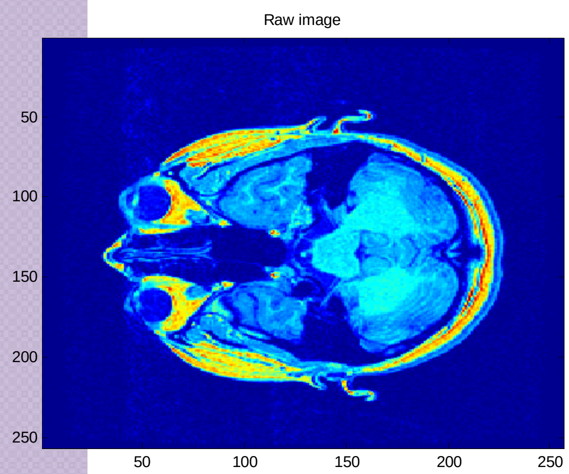


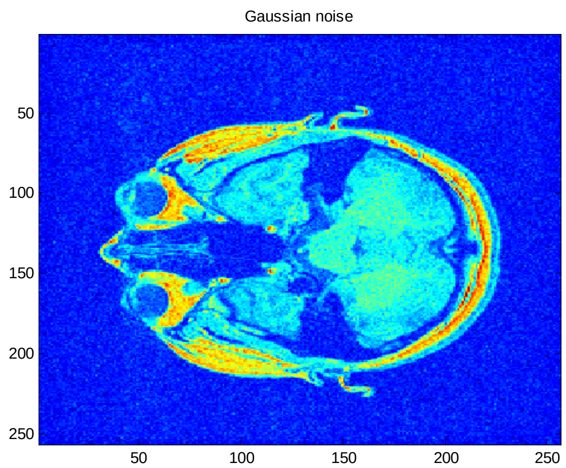
FIGURE 3.13
Moving-window filtering of an image. The size of the moving window in the illustration is 5 × 5 pixels. Statistics computed by using the pixels within the window are applied to the pixel at the same location in the output image. The moving window is shown for two pixel locations marked # and @.

Mean Filter

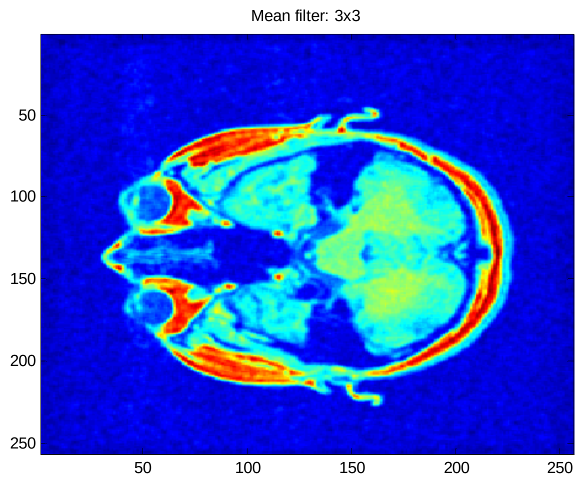
- A local neighbourhood of each pixel is considered as an ensemble
- Each pixel is substituted by its local spatial average



Raw image



Gaussian noised image (MSE = 99.03)

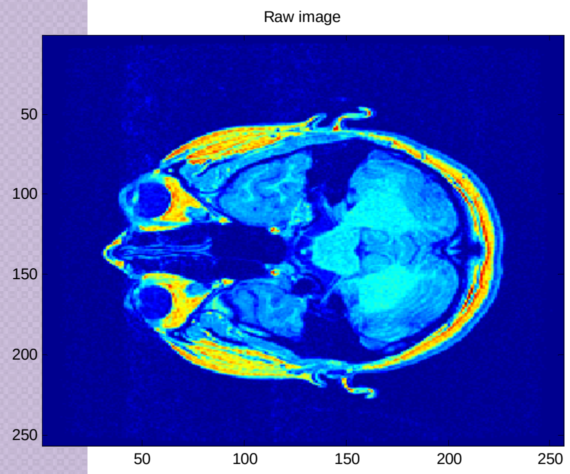


Mean filtered image (MSE = 80.09)

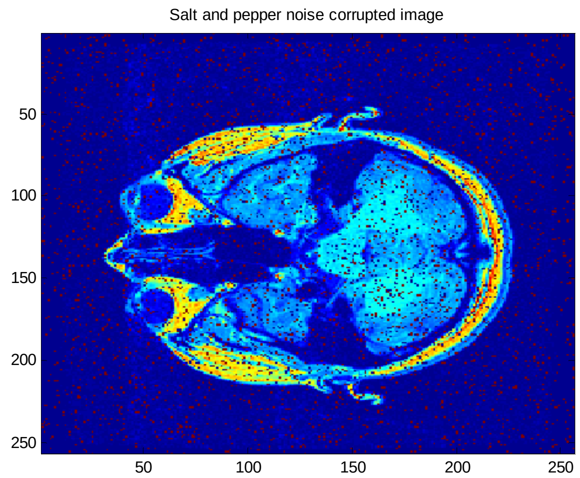
Oversmoothing may be avoided by selectively applying to 'non edge' pixels. This however, makes the filter non linear

Median Filter

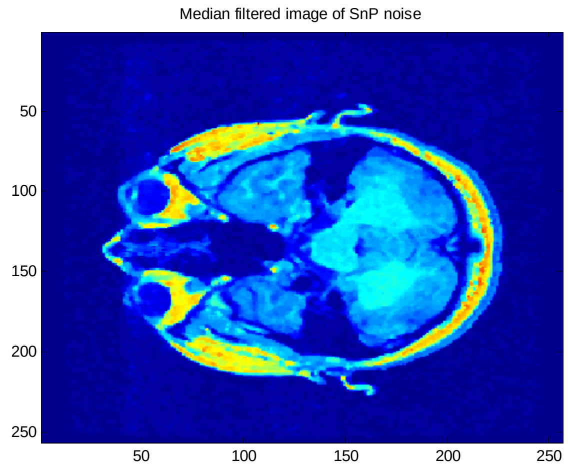
- Each pixel is replaced by the median of its local neighborhood
- Uselful to remove outliers in the histogram (eg. Impulse noise)



Raw image



S&P noised image (MSE = 2903.1)



Median filtered image (MSE = .2)9

Order Statistics Filters

- A class of non linear filters
- The pixels in the nbd are ordered by intensity
- i-th entry is the output of the i-th order statistic filter
- Eg. 1st entry is the min-filter
- 2nd entry is the max-filter
- Middle entry is the median-filter
- α trimmed mean filter: α percent of the top and bottom of the list are rejected and the mean of the rest is chosen

Frequency-domain filters

Take advantage using the frequency domain

⬇

Most image vary slowly and smootly across space

⬇

Energy concentrated in small region around (k,l)=0

⬇

Remove high-frecuency components

Procedure

1
- 2D Fourier transform of the image, padding the image with 0
  - **F(k,l)**

2
- Design or select the appropiate 2D filter transfer function
  - **H(k,l)**

3
- Obtain the filtered image in Fourier Domain (center or fold)
  - **G(k,l)=H(k,l)F(k,l)**

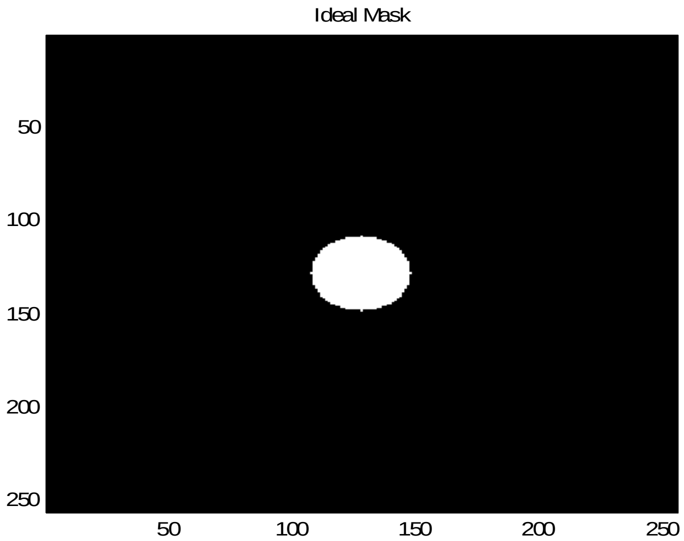
4
- Inverse Fourier Fourier Transform of G(k,l), (unfold)
  - **G(m,n)**

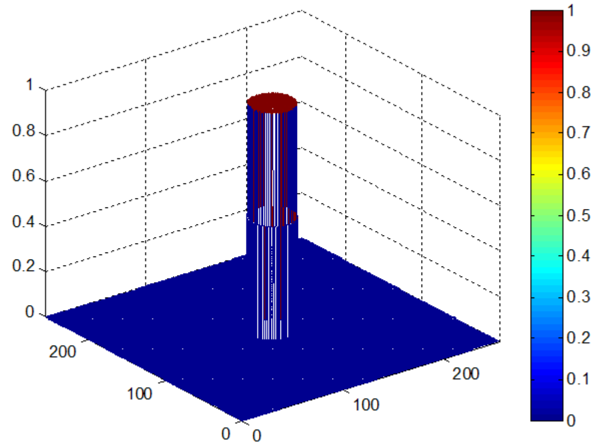
5
- Trim the resulting image g(m,n), if it was zero-padded

Ideal lowpass filter

D0 = 20

$$H(u,v) = \begin{cases} 1, & D(k,l) < D_0 \\ 0, & otherwise \end{cases} \qquad D(k,l) = \sqrt{k^2 + l^2}$$





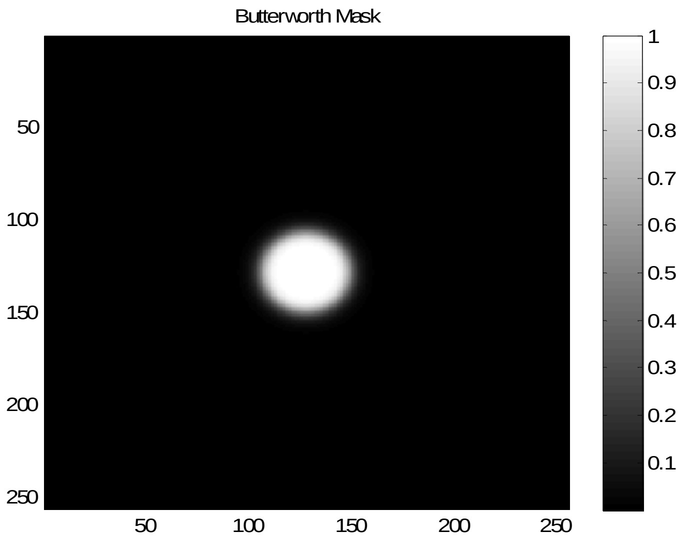
Butterworth lowpass filter

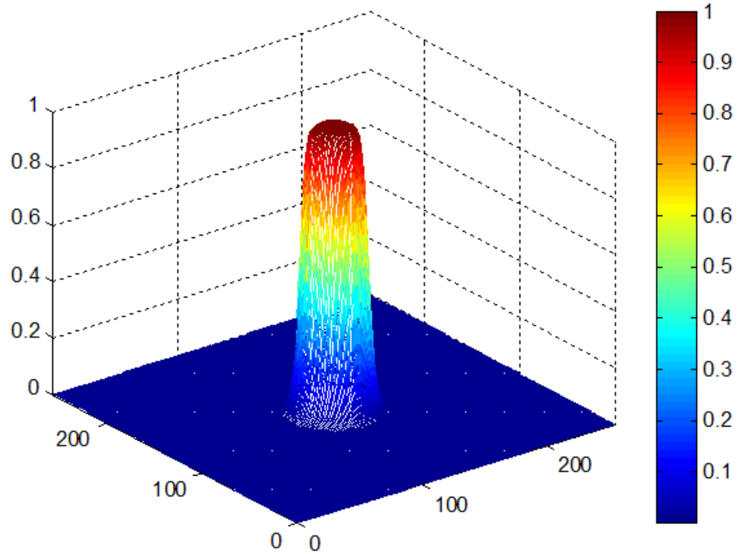
n=5 D0 = 20

$$H(k,l) = \frac{1}{1 + (\sqrt{2} - 1)\left[\frac{D(k,l)}{D_0}\right]^{2n}}$$

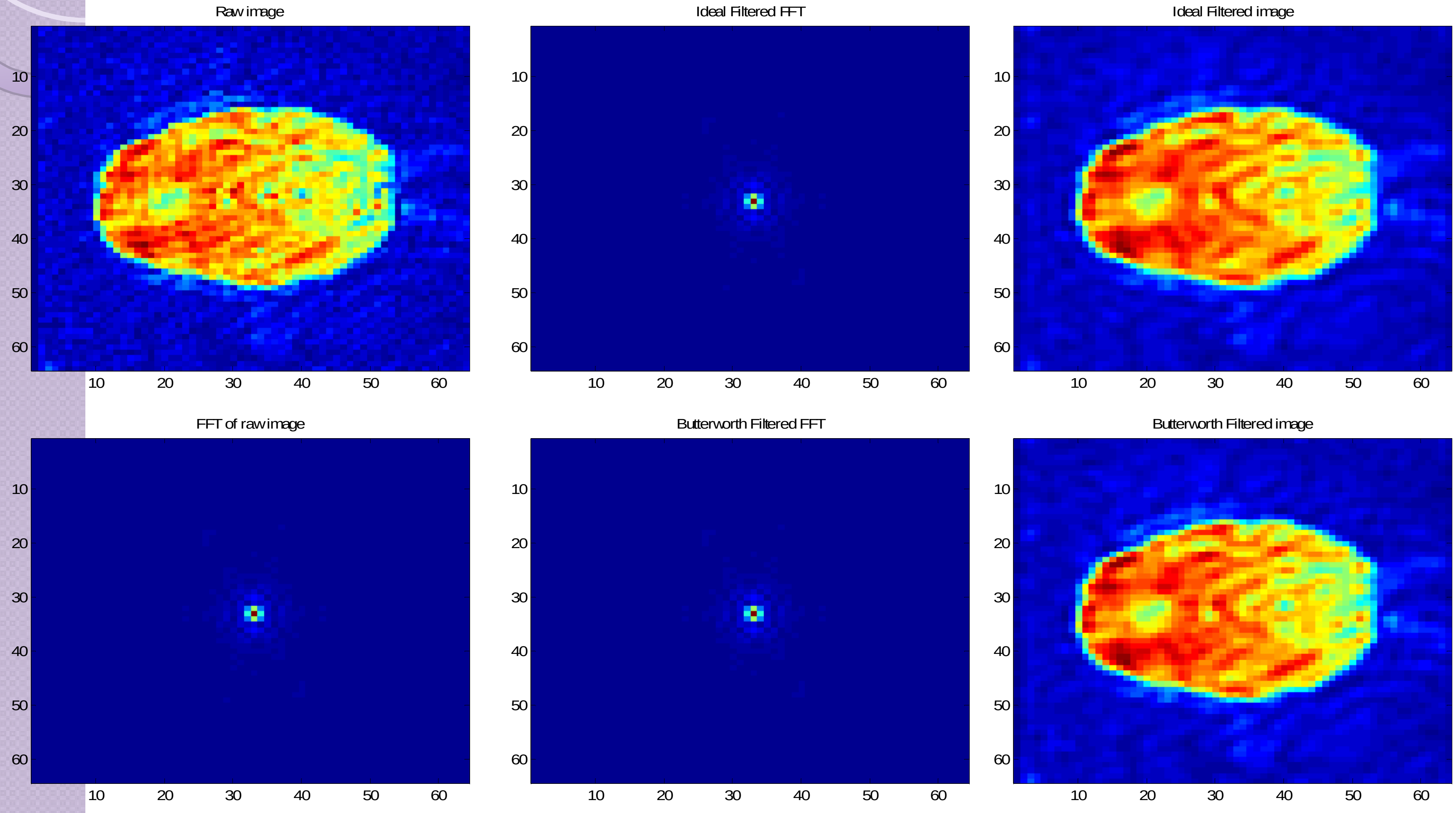
$$D(k,l) = \sqrt{k^2 + l^2}$$

$$D_0 \rightarrow D(k,l) = \frac{1}{\sqrt{2}}$$

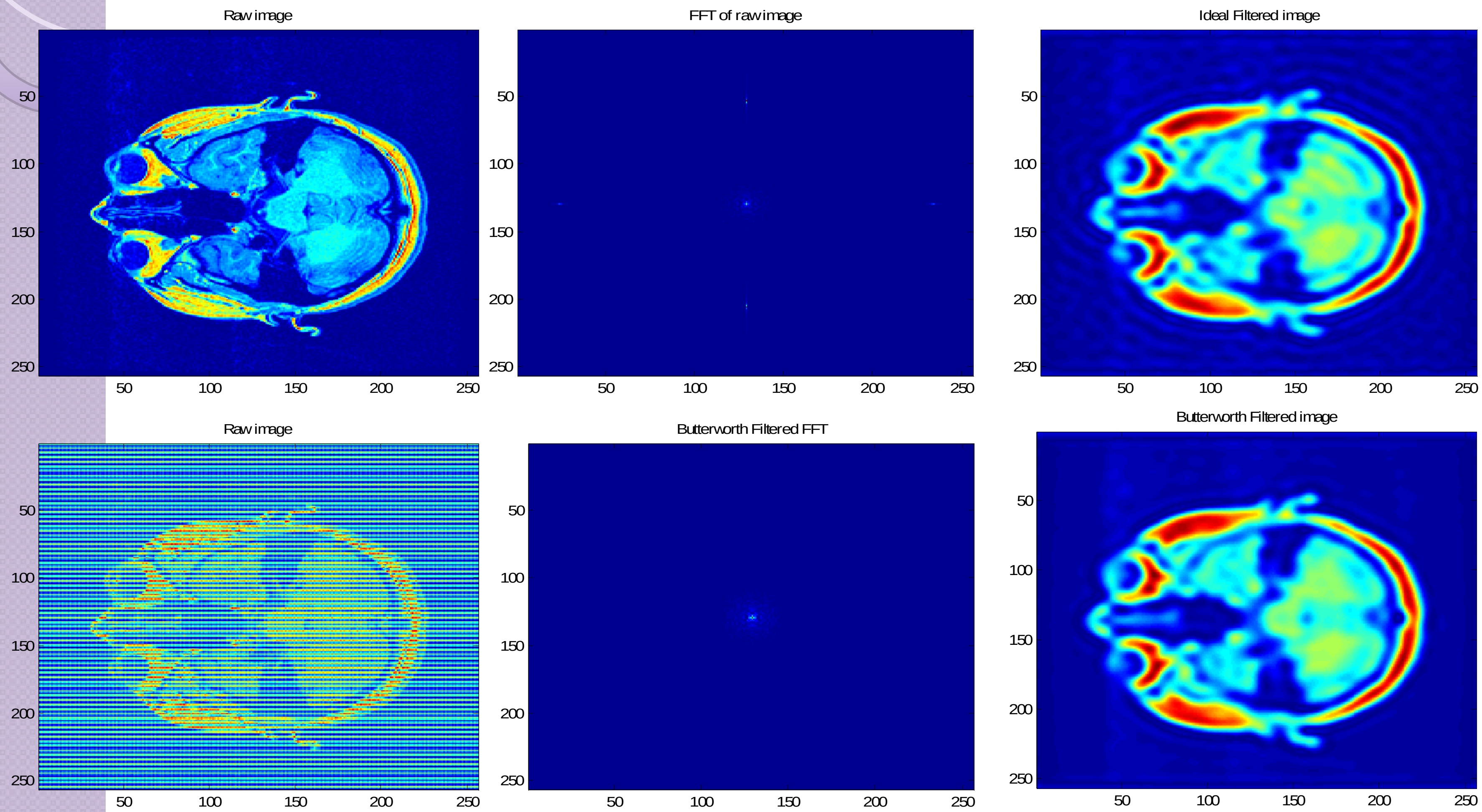




Removal of high-frequency noise

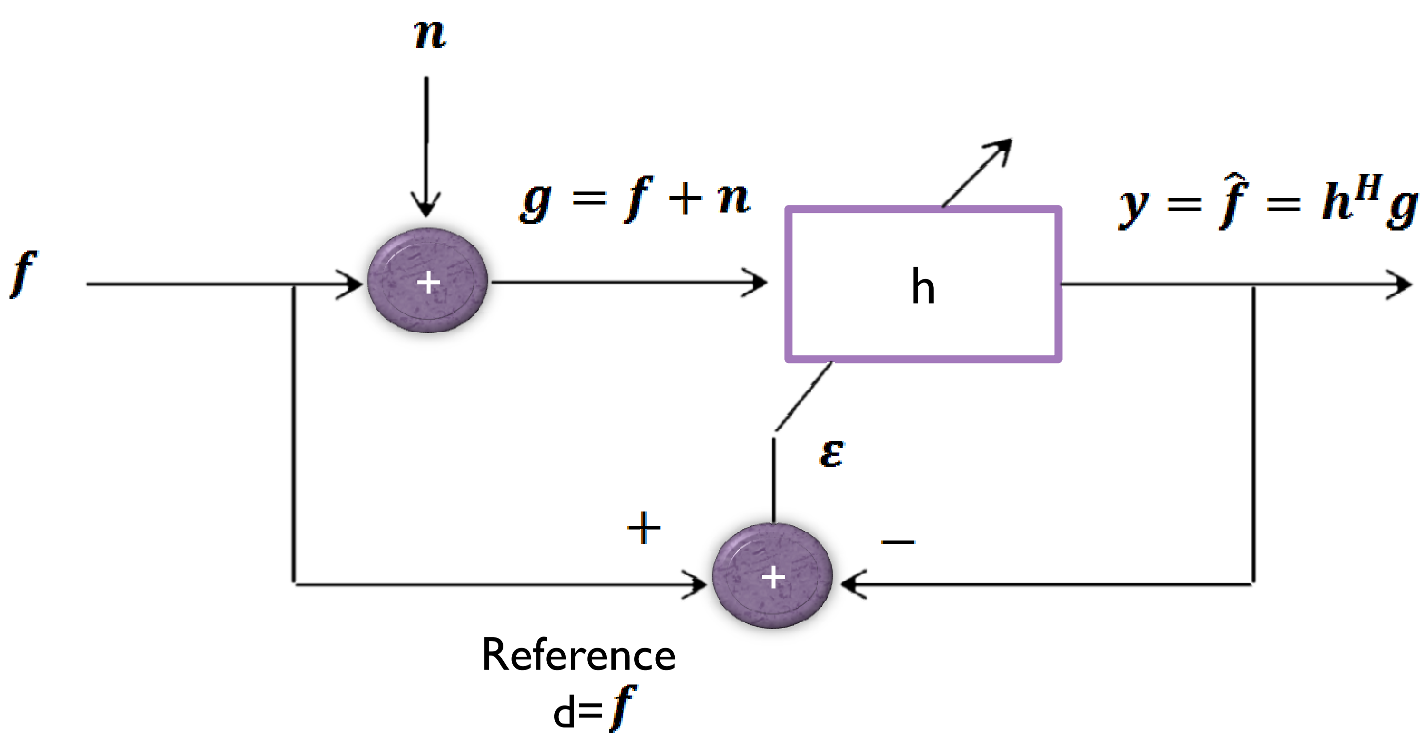


Removal of peridodic artifacts



Optimal filtering: Wiener filtering

Aim ⇨ minimize the mean square error



Wiener filtering: equations I

$$\xi = E\{\varepsilon(n)^2\} = E\{|(d-y)|^2\} = E\{|(f-\hat{f})|^2\}$$

$$\xi = E\{|(f-h^H g)|^2\} = P_f + h^h E\{gg^T\}h - h^H E\{fg^H\} - E\{gf^H\}h$$

if $P = E\{fg^H\}$ ⇨ $\xi = P_f + h^H R_g h - h^H P - P^H h$

$$\Delta_{h^H}\xi = R_g h - P = 0$$

$$h_{optim} = R_g^{-1} P = R_g^{-1} fg^H \qquad h_{optim,book} = R_f(R_f + R_n)^{-1}$$

Wiener filtering: equations II

$$y = \hat{f} = h_{opt}{}^{H} g = P^{H} R_{g}{}^{-1} g \quad y = \hat{f}_{book} = h_{opt}{}^{H} g = R_f (R_f + R_n)^{-1} g$$

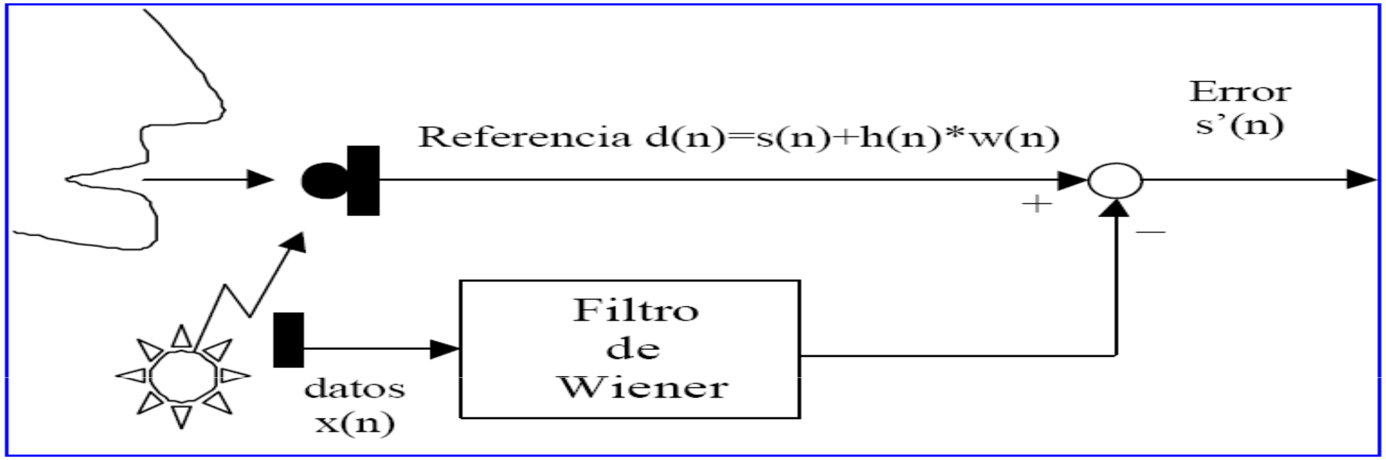
Mse:

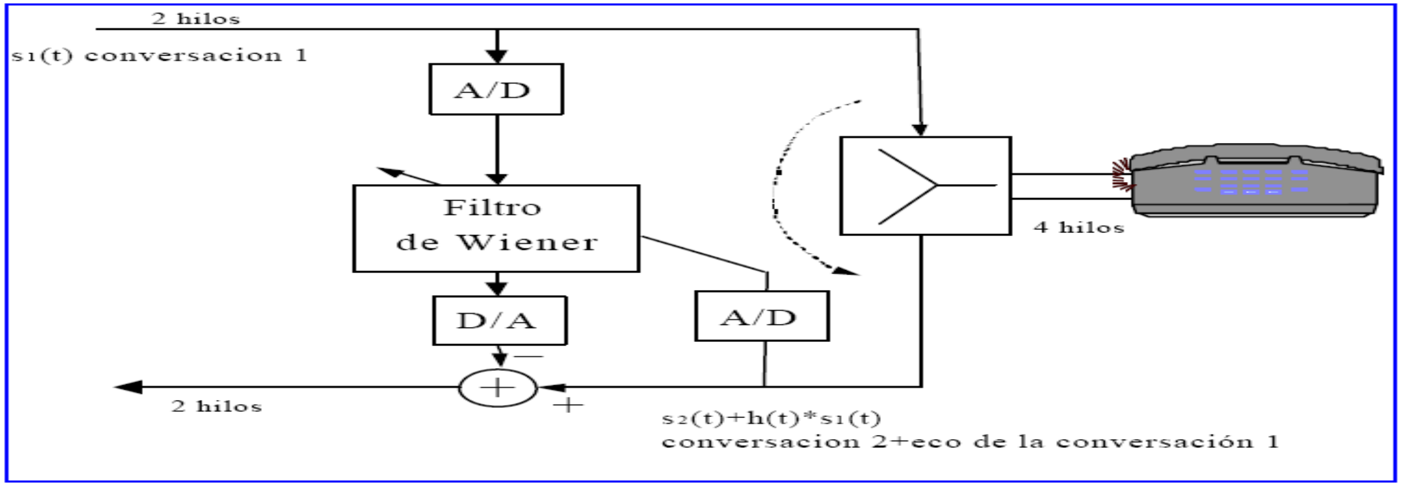
$$\xi_{min} = P_f - P^H R_g{}^{-1} P$$

Fourier domain:

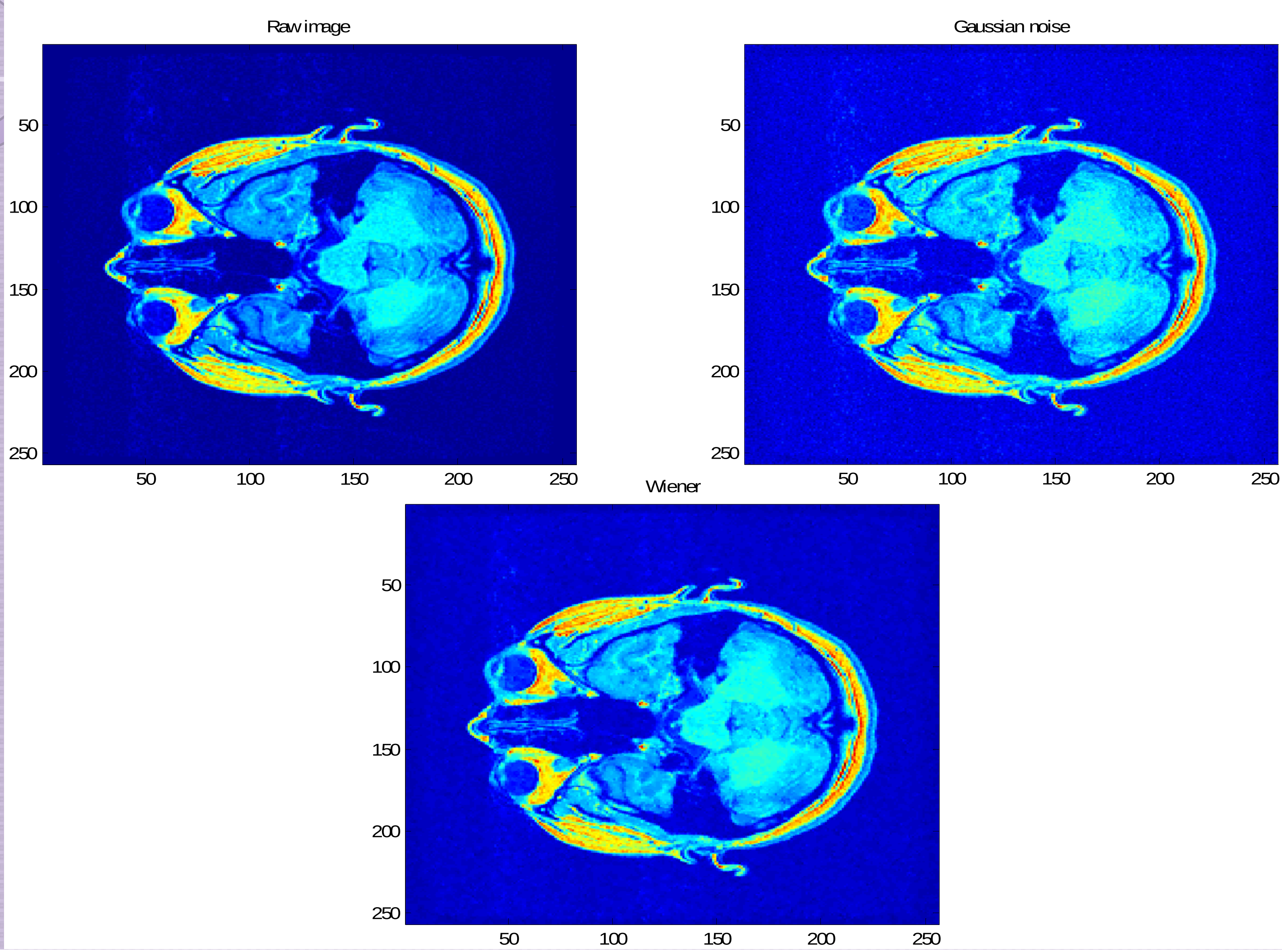
$$\widetilde{F(k,l)} = \left[\frac{1}{1 + \frac{S_{\eta(k,l)}}{S_{f(k,l)}}} \right] G(k,l)$$

Wiener filtering: more aplications





Removal of noise with Wiener filtering



End of the presentation

Questions round