Image restoration: linear solutions

— Inverse filtering and Wiener filtering —

Jean-François Giovannelli

Groupe Signal – Image
Laboratoire de l'Intégration du Matériau au Système
Univ. Bordeaux – CNRS – BINP

Topics

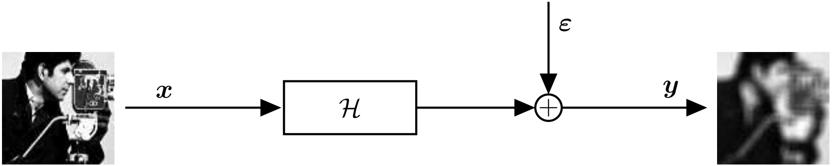
- Image restoration, deconvolution
  - Motivating examples: medical, astrophysical, industrial, vision,...
  - Various problems: deconvolution, Fourier synthesis, denoising...
  - Missing information: ill-posed character and regularisation

- Three types of regularised inversion
  1. Quadratic penalties and linear solutions
     - Closed-form expression
     - Computation through FFT
     - Optimisation (*e.g.*, gradient), system solvers (*e.g.*, splitting)
  2. Non-quadratic penalties and edge preservation
     - Half-quadratic approaches, including computation through FFT
     - Optimisation (*e.g.*, gradient), system solvers (*e.g.*, splitting)
  3. Constraints: positivity and support
     - Augmented Lagrangian and ADMM, including computation by FFT
     - Optimisation (*e.g.*, gradient), system solvers (*e.g.*, splitting)

- Bayesian strategy: a few incursions
  - Tuning hyperparameters, instrument parameters,...
  - Hidden / latent parameters, segmentation, detection,...

Inversion: standard question

$$\boldsymbol{y} = \mathcal{H}(\boldsymbol{x}) + \boldsymbol{\varepsilon} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\varepsilon} = \boldsymbol{h} \star \boldsymbol{x} + \boldsymbol{\varepsilon}$$



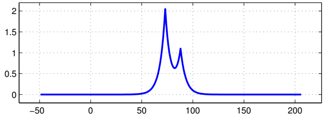
$$\widehat{\boldsymbol{x}} = \widehat{\mathcal{X}}(\boldsymbol{y})$$

Restoration, deconvolution-denoising

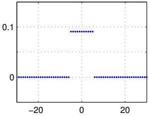
- General problem: ill-posed inverse problems, *i.e., lack of information*
- Methodology: regularisation, *i.e., information compensation*
  - Specificity of the inversion / reconstruction / restoration methods
  - Trade off and tuning parameters
- Limited quality results

Example due to Hunt ("square" response) [1970]

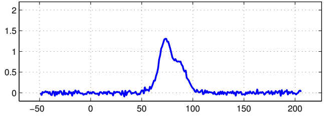
- Convolutive model: $\boldsymbol{y} = \boldsymbol{h} \star \boldsymbol{x} + \boldsymbol{\varepsilon}$
- Samples averaging



True $\boldsymbol{x}$



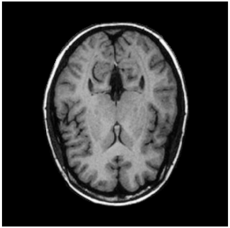
Response $\boldsymbol{h}$



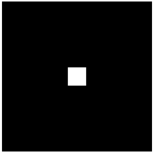
Observation $\boldsymbol{y}$

Example: brain ("square" response)

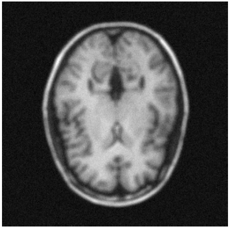
- Convolutive model: $\boldsymbol{y} = \boldsymbol{h} \star \boldsymbol{x} + \boldsymbol{\varepsilon}$
- Pixels averaging
- Think also about the Fourier domain



True $\boldsymbol{x}$



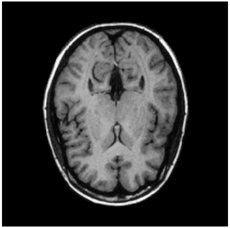
Spatial response $\boldsymbol{h}$



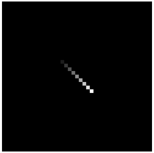
Observation $\boldsymbol{y}$

Example: brain (“motion blur”)

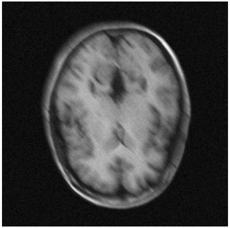
- Convolutive model: $\boldsymbol{y} = \boldsymbol{h} \star \boldsymbol{x} + \boldsymbol{\varepsilon}$
- Pixels averaging
- Think also about the Fourier domain



True $\boldsymbol{x}$



Spatial response $\boldsymbol{h}$

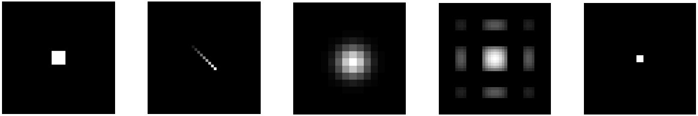


Observation $\boldsymbol{y}$

Convolution

- Examples of response

Square Motion Gaussian Diffraction Point



- Convolutive model (1D, 2D and also 3D,...)

$$
\begin{aligned}
z(n) &= \sum_{p=-P}^{+P} h(p)\,x(n-p) \\
z(n,m) &= \sum_{p=-P}^{+P} \sum_{q=-Q}^{+Q} h(p,q)\,x(n-p,m-q)
\end{aligned}
$$

- Response: $h(p)$ or $h(p,q)$
  - impulse response, convolution kernel,...
  - ...point spread function, stain image

Matrix form: 1D convolution

- Linear $\rightsquigarrow$ matricial relation: $\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x}$
- Shift invariance $\rightsquigarrow$ Tœplitz structure
- Short response $\rightsquigarrow$ band structure

$$
\begin{aligned}
& \vdots \\
z_{n-1} &= \\
z_n &= h_P x_{n-P} + \cdots + h_1 x_{n-1} + h_0 x_n + h_{-1} x_{n+1} + \cdots + h_{-P} x_{n+P} \\
z_{n+1} &= \\
& \vdots
\end{aligned}
$$

$$
\boldsymbol{H} = \left[\begin{array}{cccccccccccc}
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & \\
 & h_P & \dots & h_0 & \dots & h_{-P} & 0 & 0 & 0 & 0 & 0 & \\
\dots & 0 & h_P & \dots & h_0 & \dots & h_{-P} & 0 & 0 & 0 & 0 & \dots \\
\dots & 0 & 0 & h_P & \dots & h_0 & \dots & h_{-P} & 0 & 0 & 0 & \dots \\
\dots & 0 & 0 & 0 & h_P & \dots & h_0 & \dots & h_{-P} & 0 & 0 & \dots \\
\dots & 0 & 0 & 0 & 0 & h_P & \dots & h_0 & \dots & h_{-P} & 0 & \dots \\
 & 0 & 0 & 0 & 0 & 0 & h_P & \dots & h_0 & \dots & h_{-P} & \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots &
\end{array}\right]
$$

Dealing with side effects

- Extend convolution matrix into a circulant matrix
- Approximation "periodic objects"

$$
\left[\begin{array}{cccccccccccccc}
 & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \\
 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 & h_P & h_{P-1} & h \\
 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & h_P & h \\
 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & h \\
 & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \\
 & \cdots & 0 & h_P & \cdots & h_0 & \cdots & h_{-P} & 0 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & h_P & \cdots & h_0 & \cdots & h_{-P} & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & h_P & \cdots & h_0 & \cdots & h_{-P} & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & 0 & h_P & \cdots & h_0 & \cdots & h_{-P} & 0 & \cdots & \\
 & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \\
h_{-P} & 0 & 0 & 0 & \cdots & & & & & & & & & \\
h_{-P+1} & h_{-P} & 0 & 0 & \cdots & & & & & & & & & \\
h_{-P+2} & h_{-P+1} & h_{-P} & 0 & 0 & \cdots & & & & & & & & \\
 & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & &
\end{array}\right.
$$

Never compute that. Never.

Circulant case: diagonalization

- Circulant matrices diagonalization...

$$\bar{\boldsymbol{H}} = \boldsymbol{F}^{\dagger} \boldsymbol{\Lambda}_h \boldsymbol{F}$$

- ...in the Fourier basis

$$\boldsymbol{F} = N^{-1/2} \left[e^{-j2\pi(k-1)(l-1)/N} \right]_{k,l \in 1,\ldots,N}$$

- Reminder of properties

$$\begin{aligned} & \boldsymbol{F}^{\mathrm{t}} && = && \boldsymbol{F} \\ \boldsymbol{F}^{\dagger}\boldsymbol{F} = {} & \boldsymbol{F}\boldsymbol{F}^{\dagger} && = && \boldsymbol{I}_N \\ & \boldsymbol{F}^{-1} && = && \boldsymbol{F}^{\dagger} = \boldsymbol{F}^{*} \end{aligned}$$

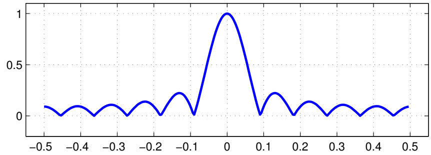
$$\begin{aligned} \boldsymbol{F}\boldsymbol{x} &= \texttt{FFT(x)} \\ \boldsymbol{F}^{\dagger}\boldsymbol{x} &= \texttt{IFFT(x)} \end{aligned}$$

Circulant case: eigenvalues

- Eigenvalues $\sim$ frequency response

$$\mathring{\boldsymbol{h}} = \begin{bmatrix} \mathring{h}_0 \\ \mathring{h}_1 \\ \vdots \\ \mathring{h}_{N-2} \\ \mathring{h}_{N-1} \end{bmatrix} = \sqrt{N}\ \boldsymbol{F} \begin{bmatrix} \cdot \\ 0 \\ h_{-P} \\ \cdot \\ h_0 \\ \cdot \\ h_P \\ 0 \\ \cdot \end{bmatrix} = \texttt{fft(h,N)}$$

- Eigenvalues "read" on the frequency response graph



- Comment: conditioning of $\boldsymbol{H}$ and low pass character

Circulant case: convolution through FFT

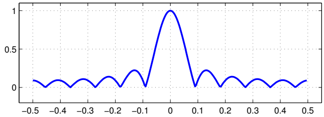
- Matricial form of the convolution through FFT

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} = \boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_h\boldsymbol{F}\boldsymbol{x}$$

$$\begin{aligned} \boldsymbol{F}\boldsymbol{z} &= \boldsymbol{\Lambda}_h\ \boldsymbol{F}\boldsymbol{x} \\ \mathring{\boldsymbol{z}} &= \boldsymbol{\Lambda}_h\ \mathring{\boldsymbol{x}} \end{aligned}$$

$$\mathring{\boldsymbol{z}} = \mathring{\boldsymbol{h}}\ .*\ \mathring{\boldsymbol{x}}$$

$$\mathring{z}_n = \mathring{h}_n\,\mathring{x}_n \quad \text{pour } n = 1, \ldots, N$$

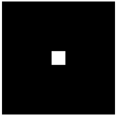


- Frequency attenuation, lowpass character, ill-conditioned character
- Possibly non-invertible system

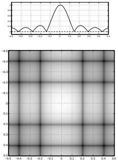
- Remark: exact convolution calculus always possible by FFT but…

Example: photographed photographer (“square” response)

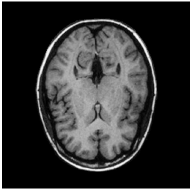
- Convolutive model (pixels averaging): $\boldsymbol{y} = \boldsymbol{h} \star \boldsymbol{x} + \boldsymbol{\varepsilon}$
- Fourier domain:
$$\begin{array}{cccccccc} \mathring{y}(\nu) & = & \mathring{h}(\nu) & . & \mathring{x}(\nu) & + & \mathring{\varepsilon}(\nu) \\ \mathring{\boldsymbol{y}} & = & \mathring{\boldsymbol{h}} & .* & \mathring{\boldsymbol{x}} & + & \mathring{\boldsymbol{\varepsilon}} \end{array}$$



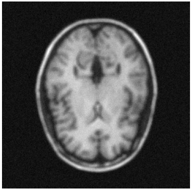
Spatial response



Frequency response



True ($\boldsymbol{x}$)

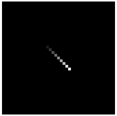


Observation ($\boldsymbol{y}$)

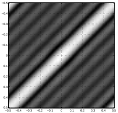
Example: photographed photographer ("motion blur")

- Convolutive model (pixels averaging): $\boldsymbol{y} = \boldsymbol{h} \star \boldsymbol{x} + \boldsymbol{\varepsilon}$
- Fourier domain:

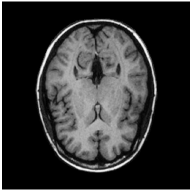
$$\begin{array}{ccccccc} \mathring{y}(\nu) & = & \mathring{h}(\nu) & . & \mathring{x}(\nu) & + & \mathring{\varepsilon}(\nu) \\ \mathring{\boldsymbol{y}} & = & \mathring{\boldsymbol{h}} & .* & \mathring{\boldsymbol{x}} & + & \mathring{\boldsymbol{\varepsilon}} \end{array}$$



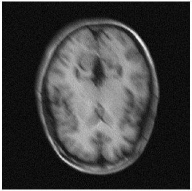
Spatial response



Frequency response



True ($\boldsymbol{x}$)



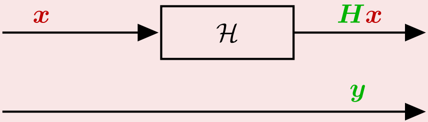
Observation ($\boldsymbol{y}$)

First approach for restoration

First restoration: least squares

- Compare observations $\boldsymbol{y}$ and model output $\boldsymbol{H}\boldsymbol{x}$
  - Unknown: $\boldsymbol{x}$
  - Known: $\boldsymbol{H}$ and $\boldsymbol{y}$

Comparison experiment – model



- Quadratic criterion: distance observation – model output

$$\mathcal{J}_{\mathrm{LS}}(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2$$

- Least squares solution

$$\widehat{\boldsymbol{x}}_{\mathrm{LS}} = \arg\min_{\boldsymbol{x}} \mathcal{J}_{\mathrm{LS}}(\boldsymbol{x})$$

  - Solution $\widehat{\boldsymbol{x}}_{\mathrm{LS}}$: the best one to reproduce data
  - Faut que "*ça colle*", it must fit, it must match

Calculational aspects

Linear model + least squares $\rightsquigarrow$ Quadratic criterion

$$\begin{aligned}\mathcal{J}_{\mathrm{LS}}(\boldsymbol{x}) \;=\; \|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2 &= (\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x})^{\mathrm{t}}\,(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}) \\ &= \boldsymbol{x}^{\mathrm{t}}\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H}\boldsymbol{x} - 2\boldsymbol{y}^{\mathrm{t}}\boldsymbol{H}\boldsymbol{x} + \boldsymbol{y}^{\mathrm{t}}\boldsymbol{y}\end{aligned}$$

- Gradient calculus $\rightsquigarrow$ linear

$$\boldsymbol{g}(\boldsymbol{x}) = \frac{\partial \mathcal{J}_{\mathrm{LS}}}{\partial \boldsymbol{x}} = 2\,\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H}\boldsymbol{x} - 2\,\boldsymbol{H}^{\mathrm{t}}\boldsymbol{y} = -2\boldsymbol{H}^{\mathrm{t}}(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x})$$

- And Hessian calculus $\rightsquigarrow$ constant

$$\boldsymbol{Q} = \frac{\partial^2 \mathcal{J}_{\mathrm{LS}}}{\partial \boldsymbol{x}^2} = 2\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H} \qquad\qquad (\geqslant 0 \;\text{ or }\; > 0)$$

- Gradient nullification $\rightsquigarrow$ linear system $\rightsquigarrow$ matrix inversion

$$\begin{aligned}(\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H})\,\widehat{\boldsymbol{x}}_{\mathrm{LS}} &= \boldsymbol{H}^{\mathrm{t}}\boldsymbol{y} \\ \widehat{\boldsymbol{x}}_{\mathrm{LS}} &= (\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{t}}\boldsymbol{y}\end{aligned}$$

Computational aspects and implementation

Various options and many relationships...

- Direct calculus, compact (closed) form, matrix inversion
- Algorithms for linear system
  - Gauss, Gauss-Jordan
  - Substitution
  - Triangularisation,...
- Numerical optimisation
  - gradient descent ... and various modifications
  - Pixel wise, pixel by pixel
- Diagonalization
  - Circulant approximation and diagonalization by FFT
- Special algorithms, especially for 1D case
  - Recursive least squares
  - Kalman smoother or filter (and fast versions)

Computations through FFT

$$
\begin{aligned}
\widehat{\boldsymbol{x}} &= (\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{t}}\boldsymbol{y} \\
&\simeq (\bar{\boldsymbol{H}}^{\mathrm{t}}\bar{\boldsymbol{H}})^{-1}\bar{\boldsymbol{H}}^{\mathrm{t}}\boldsymbol{y} \\
&= (\boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_h^{\dagger}\boldsymbol{F}\boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_h\boldsymbol{F})^{-1}\boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_h^{\dagger}\boldsymbol{F}\boldsymbol{y} \\
&= \boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_h^{-1}\boldsymbol{F}\boldsymbol{y}
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{F}\widehat{\boldsymbol{x}} &= \boldsymbol{\Lambda}_h^{-1}\boldsymbol{F}\boldsymbol{y} \\
\mathring{\widehat{\boldsymbol{x}}} &= \boldsymbol{\Lambda}_h^{-1}\mathring{\boldsymbol{y}}
\end{aligned}
$$

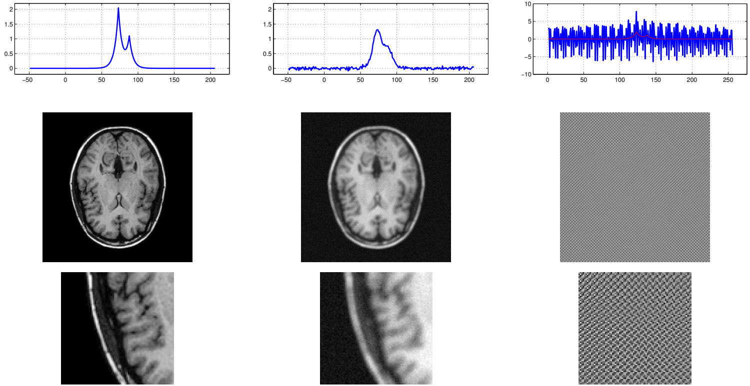
$$
\mathring{\widehat{x}}_n = \frac{\mathring{y}_n}{\mathring{h}_n} \quad \text{pour } n = 1, \ldots, N
$$

It is just inverse filtering !!

Matlab Pseudo-code

```
ObjetEstLS = IFFT( FFT(Data) ./ fft(IR) )
```

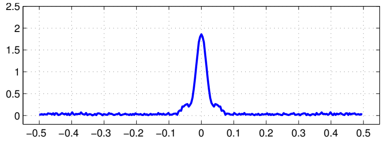
Least squares solution (Hunt and Brain)



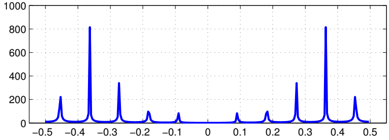
True | Observation | LS solution

Advantages / Disadvantages

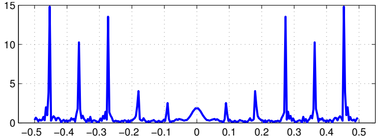
- Very general, hyper-fast, no parameter to tune
- ... but does not work ... (except if ...)



Observation $\boldsymbol{y}$



Inverse filter $\mathring{\boldsymbol{h}}$

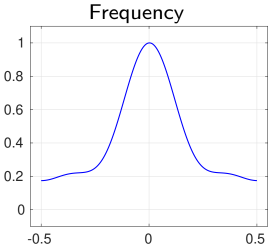


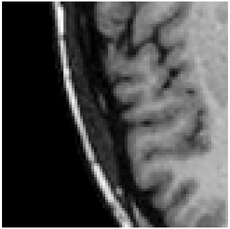
LS solution $\widehat{\boldsymbol{x}}$

Default of the least squares solution

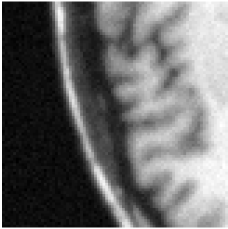
- Unacceptable solution, explosive, noise dominated
- Three analyses: signal-image, numerical, statistical
  - Bandwidth, non-observed or badly-observed frequencies
  - Badly-scaled matrix, eigenvalues, numerical instabilities
  - Strong variance (even if minimal variance and unbiased)
- Ill-posed problem
  - Missing information
  - Regularisation

Least squares solution: well conditioned problem





True



Observation

LS solution

Regularisation: generalities

- Data insufficiently informative
  $\rightsquigarrow$ Account for prior information

  1. Question of compromise, of competition
  2. Specificity of methods

  $\rightsquigarrow$ Here: *smoothness* of images, ideally *edge preserving*
  $\rightsquigarrow$ Explicitation of prior information

- Regularisation
  - Through penalty, constraint, re-parametrisation
  - ...or other ideas ...but regularisation anyhow

- Regularisation by penalty

$$\mathcal{J}_{\text{PLS}}(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 + \mu\,\mathcal{P}(\boldsymbol{x})$$

- Restored image

$$\widehat{\boldsymbol{x}}_{\text{PLS}} = \arg\min_{\boldsymbol{x}} \mathcal{J}_{\text{PLS}}(\boldsymbol{x})$$

Quadratic penalty (1)

- Differences, higher order derivatives, generalizations,...

$$
\begin{aligned}
\mathcal{P}(\boldsymbol{x}) &= \sum_n (x_{n+1} - x_n)^2 \\
\mathcal{P}(\boldsymbol{x}) &= \sum_n (x_{n+1} - 2x_n + x_{n-1})^2 \\
\mathcal{P}(\boldsymbol{x}) &= \sum_n (\alpha\ x_{n+1} - x_n + \alpha'\ x_{n-1})^2 \\
\mathcal{P}(\boldsymbol{x}) &= \sum_n (\boldsymbol{\alpha}_n^{\mathrm{t}}\, \boldsymbol{x})^2
\end{aligned}
$$

- Linear combinations (wavelet, other-truc-en-et,...dictionaries,...)

$$
\mathcal{P}(\boldsymbol{x}) = \sum_n (\boldsymbol{w}_n^{\mathrm{t}} \boldsymbol{x})^2 = \sum_n \left(\sum_m w_{nm} x_m \right)^2
$$

  - Redundant or not
  - Link with Haar wavelet and other

Quadratic penalty (2)

- 2D aspects: derivative, finite differences, gradient approximations

$$
\begin{aligned}
\mathcal{P}(\boldsymbol{x}) &= \sum_{p\sim q}(x_p - x_q)^2 \\
&= \sum_{n,m}(x_{n+1,m} - x_{n,m})^2 + \sum_{n,m}(x_{n,m+1} - x_{n,m})^2
\end{aligned}
$$

- Norms of filtered images
  - Through rows and columns

$$
\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}
$$

  - Jointly both of them

$$
\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}
$$

- Remark
  - Notion of neighborhood and Markov field
  - Any highpass filter, contour detector (Prewitt, Sobel,. . .)
  - Linear combinations: wavelet, contourlet and other truc-en-et,. . . dictionaries,. . .

Quadratic penalty (3)

- Other possibilities (slightly different)
  - Enforcement towards a known shape $\boldsymbol{x}_0$

$$\mathcal{P}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_0)^{\mathrm{t}} (\boldsymbol{x} - \boldsymbol{x}_0)$$

  - Usual Euclidean norm (separable terms)

$$\mathcal{P}(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{t}} \boldsymbol{x}$$

- More general form

$$\mathcal{P}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_0)^{\mathrm{t}} \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{x}_0)$$

- In the following development

$$\mathcal{P}_1(\boldsymbol{x}) = \sum_{p \sim q} (x_p - x_q)^2 = \|\boldsymbol{D}\boldsymbol{x}\|^2 = \boldsymbol{x}^{\mathrm{t}} \boldsymbol{D}^{\mathrm{t}} \boldsymbol{D} \boldsymbol{x}$$

und $\boldsymbol{D} = \dots$

Penalised least squares restoration

- Remind the criterion...

$$\mathcal{J}_{\text{PLS}}(\boldsymbol{x}) \quad = \quad \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 + \mu \, \|\boldsymbol{D}\boldsymbol{x}\|^2$$

- ...and its gradient...

$$\boldsymbol{g}(\boldsymbol{x}) = \frac{\partial \mathcal{J}_{\text{PLS}}}{\partial \boldsymbol{x}} = -2\boldsymbol{H}^{\text{t}}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}) + 2\mu \, \boldsymbol{D}^{\text{t}}\boldsymbol{D}\boldsymbol{x}$$

- ...and its Hessian...

$$\boldsymbol{Q} = \frac{\partial^2 \mathcal{J}_{\text{PLS}}}{\partial \boldsymbol{x}^2} = 2\boldsymbol{H}^{\text{t}}\boldsymbol{H} + 2\mu \, \boldsymbol{D}^{\text{t}}\boldsymbol{D}$$

- ...the normal system of equation...

$$(\boldsymbol{H}^{\text{t}}\boldsymbol{H} + \mu\boldsymbol{D}^{\text{t}}\boldsymbol{D}) \, \widehat{\boldsymbol{x}}_{\text{PLS}} \quad = \quad \boldsymbol{H}^{\text{t}}\boldsymbol{y}$$

- ...and the minimiser ...

$$\widehat{\boldsymbol{x}}_{\text{PLS}} \quad = \quad (\boldsymbol{H}^{\text{t}}\boldsymbol{H} + \mu\boldsymbol{D}^{\text{t}}\boldsymbol{D})^{-1} \boldsymbol{H}^{\text{t}}\boldsymbol{y}$$

Computational aspects and implementation

Various options and many relationships...

- Direct calculus, compact (closed) form, matrix inversion
- Algorithms for linear system
  - Gauss, Gauss-Jordan
  - Substitution
  - Triangularisation,...
- Numerical optimisation
  - gradient descent... and various modifications
  - Pixel wise, pixel by pixel
- Diagonalization
  - Circulant approximation and diagonalization by FFT
- Special algorithms, especially for 1D case
  - Recursive least squares
  - Kalman smoother or filter (and fast versions)

Circulant form

- Circulant approximation
  - for $\boldsymbol{H}$ as previously
  - for $\boldsymbol{D}$ in addition
- One extra-term of interaction

$$\bar{\boldsymbol{D}} = \left[\begin{array}{ccccccc} 1 & -1 & 0 & 0 & \dots & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & \dots & 0 \\ \vdots & & \ddots & \ddots & & & \vdots \\ \vdots & & & \ddots & \ddots & & \vdots \\ 0 & \dots & \dots & 0 & 1 & -1 & 0 \\ 0 & \dots & \dots & 0 & 0 & 1 & -1 \\ \hline -1 & \dots & \dots & 0 & 0 & 0 & 1 \end{array}\right] \qquad (N \times N)$$

- Diagonalization of $\bar{\boldsymbol{D}}$

$$\bar{\boldsymbol{D}} = \boldsymbol{F}^{\dagger} \boldsymbol{\Lambda}_d \boldsymbol{F}$$

- Eigenvalues by FFT

$$\mathring{\boldsymbol{d}} = \texttt{fft([-1,1])}$$

Computations by FFT

$$
\begin{aligned}
\widehat{\boldsymbol{x}} &= (\bar{\boldsymbol{H}}^{\mathrm{t}}\bar{\boldsymbol{H}} + \mu\bar{\boldsymbol{D}}^{\mathrm{t}}\bar{\boldsymbol{D}})^{-1}\bar{\boldsymbol{H}}^{\mathrm{t}}\boldsymbol{y} \\
&= (\boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_h^{\dagger}\boldsymbol{F}\boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_h\boldsymbol{F} + \boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_d^{\dagger}\boldsymbol{F}\boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_d\boldsymbol{F}\mu)^{-1}\boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_h\boldsymbol{F}\boldsymbol{y} \\
&= \boldsymbol{F}^{\dagger}(\boldsymbol{\Lambda}_h^{\dagger}\boldsymbol{\Lambda}_h + \mu\boldsymbol{\Lambda}_d^{\dagger}\boldsymbol{\Lambda}_d)^{-1}\boldsymbol{\Lambda}_h^{\dagger}\boldsymbol{F}\boldsymbol{y}
\end{aligned}
$$

$$
\mathring{\widehat{\boldsymbol{x}}} = (\boldsymbol{\Lambda}_h^{\dagger}\boldsymbol{\Lambda}_h + \mu\boldsymbol{\Lambda}_d^{\dagger}\boldsymbol{\Lambda}_d)^{-1}\boldsymbol{\Lambda}_h^{\dagger}\mathring{\boldsymbol{y}}
$$

$$
\mathring{\widehat{x}}_n = \frac{\mathring{h}_n^{\,*}}{|\mathring{h}_n|^2 + \mu|\mathring{d}_n|^2}\,\mathring{y}_n \qquad \text{for } n = 1, \dots N
$$

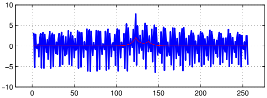
This is just Wiener filtering !!

Matlab pseudo-code

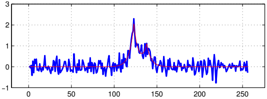
```
Gain = fft(IR)^* ./ (|fft(IR)|^2 + mu * |fft([-1,1])|^2)
ObjetEstPLS = IFFT( FFT(Data) .* Gain )
```

Solutions by quadratic penalties (Hunt)

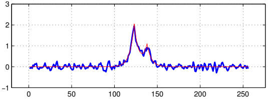
- Evolution with μ



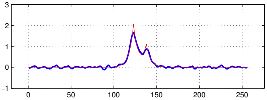
$\mu = 0$



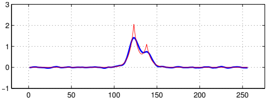
$\mu = 10^{-3}$



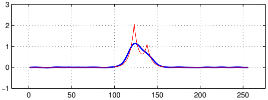
$\mu = 10^{-2}$



$\mu = 10^{-1}$

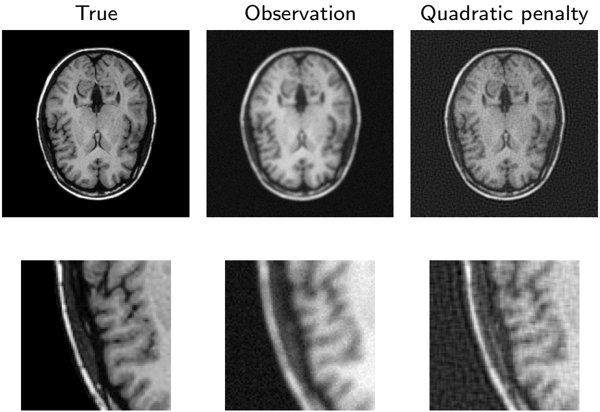


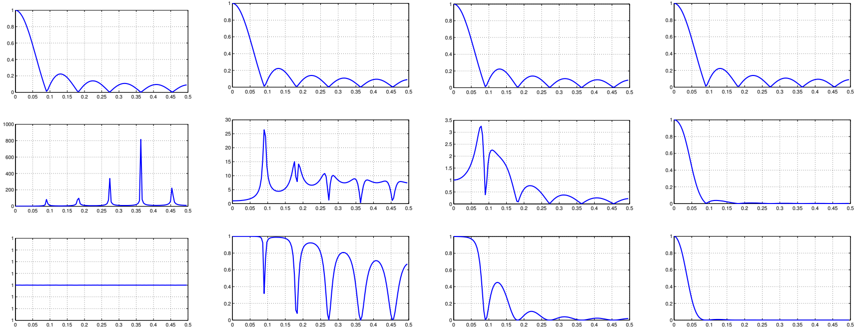
$\mu = 1$



$\mu = 10^{+1}$

Brain example

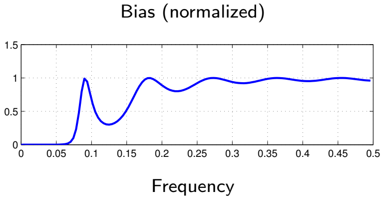


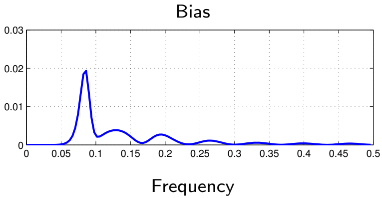


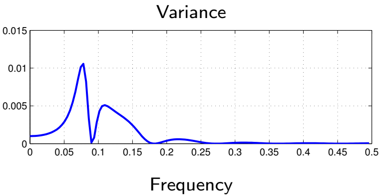
$\mu = 0$ $\quad$ $\mu = 10^{-3}$ $\quad$ $\mu = 10^{-1}$ $\quad$ $\mu = 10^{1}$

- Depending on the considered frequency
  - equalisation in the correctly observed bandwidth
  - nullification in the uncorrectly observed bandwidth

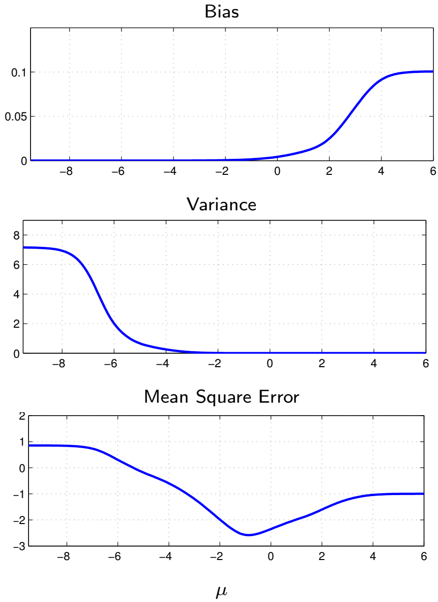
Bias / Variance (1)







Bias / Variance (2)



Photographed photographer: circulant or not

True

Observation

PLS
(circulant)

PLS
(non-circulant)

Conclusions

Synthesis

- Image deconvolution
- Quadratic penalties and smoothness of solution
  - Gradient of gray level and extensions (and other transforms)
- Closed-form expression and linear w.r.t. the data
- Numerical computations
  - Circulant case (diagonalization) $\rightsquigarrow$ FFT only
  - Numerical optimisation, system solvers,. . .

Extensions (next lectures)

- Also available for
  - non-invariant linear direct model
  - colour images, multispectral and hyperspectral
  - also signal, 3D and more, video, 3D+t. . .
- Extension to non-quadratic $\rightsquigarrow$ better image resolution
- Including constraints $\rightsquigarrow$ better image resolution
- Hyperparameters estimation, instrument parameter estimation,. . .