université de BORDEAUX

Practical work on image restoration

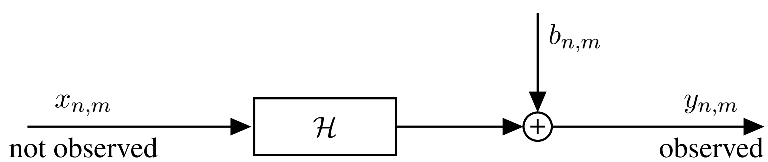
Image deconvolution : Wiener-Hunt method

Prerequisite: prepare in advance the answer to the questions no. 1 and 2.

This practical work deals with the problem of image de-blurring, that is how to retrieve a sharp image starting from the blurred (and noisy) version of it. This problem belongs to the class of inverse problems which arise when one is working with real measurement systems. This in view of the fact that each measurement system, *e.g.,* thermometer, CCD camera, spectrograph,... has its limitations imposed by the underlying physics, *e.g.,* a finite precision, a finite dynamical range, a non-zero response time,.... This means that the measured quantity is a deformed version, in a given sense and within a given measure, of the physical quantity of interest.

In most cases of interest, the measurements as directly given by the measuring system are thankfully of a sufficient precision and robustness. However, there are situations in which the former is not necessarily true. To alleviate this, at least partially if not completely, special signal and image processing techniques have been developed. In the following we will look at such techniques and as a simple illustrative example, we will consider the case of an image taken with an improperly set focus. In this case, the image of a point will actually be a spot. The captured image will suffer from blur because it is the result of the superposition of the spots generated by each point from the true image.

The simplest model to describe such a transformation is a linear invariant filter, *i.e.,* a convolution. A schematic representation of such a system is given in the following figure.



In the above schematic, $x_{n,m}$ represents the real or true image and $y_{n,m}$ represents what is measured or, better said, the image that is taken by the camera. The additive component $b_{n,m}$ is added to account for measurement and modelling errors. The equation that describes the depicted measurement process is:

$$y_{n,m} = \sum_{p=-P}^{P} \sum_{q=-Q}^{Q} h_{p,q}\, x_{n-p,m-q} + b_{n,m} \tag{1}$$

for each observed pixel (n, m). In this relation, P and Q are given integer numbers.

It is important to highlight that usually the filter type is a low-pass one. This is in line with the operational characteristics of a vast array of measuring instruments, in the sense that they are not able to accurately reproduce in their output the faster components of the input signal or image. This usually means that the high frequency components are either strongly attenuated or rejected completely, which easily explains why the inverse problem of recovering the true signal or image is difficult: one must recover the high frequency components when they are either not be present at all or they are "incorrectly" observed.

In this practical work, the problem of image deconvolution is tackled using linear methods. They rely on a least squares criterion coupled with a quadratic penalisation. The main theoretical results for such method are presented in the suite where the focus is on the criteria and their minimisers, which actually represent the reconstructed image. The technicalities behind the implementation are also discussed wherein a method based on circulant approximations allowing for fast numerical computations is presented.

1 Deconvolution 1D

In the following section the focus will be on the 1D case, that is signal deconvolution. This case allows a more in depth analysis of the deconvolution problem while still allowing one to easily grasp the concepts and ideas associated with it. The 2D case is then introduced and seen as an extension of the 1D case. The *Matlab* implementation part concerns only the 2D case.

1.1 Modelisation 1D

In the single dimensional case, the observation model given in (1) becomes:

$$y_n = \sum_{p=-P}^{P} h_p \, x_{n-p} + b_n \,, \tag{2}$$

and if we dispose of N samples, we can rewrite the corresponding N equations in a matrix form:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{b} \,. \tag{3}$$

The vector $\boldsymbol{y}$ contains all the N observed values (in the 2D case, it will contain the blurred image), the vector $\boldsymbol{x}$ contains the samples of the restored signal and $\boldsymbol{b}$ contains the noise samples. The matrix $\boldsymbol{H}$, called the convolution matrix, has the following classical structure:

$$\boldsymbol{H} = \begin{bmatrix} & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & \\ & h_P & \dots & h_0 & \dots & h_{-P} & 0 & 0 & 0 & 0 & \\ \dots & 0 & h_P & \dots & h_0 & \dots & h_{-P} & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & h_P & \dots & h_0 & \dots & h_{-P} & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & h_P & \dots & h_0 & \dots & h_{-P} & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & h_P & \dots & h_0 & \dots & h_{-P} & \dots \\ & & \vdots & & \vdots & & \vdots & & \vdots & & \end{bmatrix} .$$

It is a square matrix[1] of size $N \times N$ and it has a Tœplitz structure. The signal deconvolution problem is thus recast as the problem of estimating the vector $\boldsymbol{x}$ being given the observed signal $\boldsymbol{y}$ and knowing the convolution matrix $\boldsymbol{H}$.

1.2 Penalized least squares

The proposed reconstruction strategy is based on the penalized least squares method. It comprises two terms:

1. A term which penalises differences between successive samples of the restored signal with the aim of ensuring that it has a certain regular structure;
2. A term which quantifies the similarity between the re-convolution of the restored signal and the observed signal thus ensuring that the reconstructed signal is in line with the observed signal.

[1]... aside from side effects (first and last samples).

The criterion takes on the following expression:

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

where $\boldsymbol{D}$ is the difference matrix of order 1 and size $(N-1)\times N$, defined as follows:

$$
\boldsymbol{D} = \begin{bmatrix}
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\dots & -1 & 1 & 0 & 0 & \dots & 0 & \dots \\
\dots & 0 & -1 & 1 & 0 & \dots & 0 & \dots \\
\dots & \vdots & & \ddots & \ddots & & & \dots \\
\dots & 0 & \dots & 0 & -1 & 1 & 0 & \dots \\
\dots & 0 & \dots & 0 & 0 & -1 & 1 & \dots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
\end{bmatrix}. \tag{4}
$$

1. *Show that the minimiser of the penalized least squares criterion is:*

$$
\widehat{\boldsymbol{x}} = (\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H} + \mu\boldsymbol{D}^{\mathrm{t}}\boldsymbol{D})^{-1}\boldsymbol{H}^{\mathrm{t}}\boldsymbol{y}\,, \tag{5}
$$

that is the estimate of $\boldsymbol{x}$ in the penalized least square sense. Analyse and comment upon the resulting expression of the criterion and of the minimiser for the particular case $\mu = 0$.

1.2.1 Circulant approximation

The matrix $\boldsymbol{H}^{\mathrm{t}}\boldsymbol{H}+\mu\boldsymbol{D}^{\mathrm{t}}\boldsymbol{D}$ is of size $N\times N$ and N attains a high value, especially when working with images: for example, for an image of size 1000×1000 the matrix to invert is of size $10^6\times 10^6$. And far more large in a three dimensional context. . . The inversion becomes very costly computational-wise or impossible. There are several methods to overcome the difficulty and to compute $\widehat{\boldsymbol{x}}$, some of which are faster than the others.

The method used in this practical work makes use of the properties of the circulant matrices. This allows one to “replace” the matrices from (5) with diagonal ones. Thus, matrix operations the likes of multiplication or inversion, have a very small computational cost. The matrices are “transformed” into diagonal matrices through FFT. However, this requires approximating the matrices $\boldsymbol{H}$ and $\boldsymbol{D}$ with the circulant matrices $\tilde{\boldsymbol{H}}$ and $\tilde{\boldsymbol{D}}$.

The circulant approximation consists in modifying the top-right and/or bottom-left parts of the matrices such that they have a circulant structure. This approximation brings with it a particular assumption about the start and end of the signal/image. . . This is discussed in greater depth in the lectures.

The matrices $\tilde{\boldsymbol{H}}$ and $\tilde{\boldsymbol{D}}$ being circulant convolution matrices are easily diagonalised in the Fourier basis:

$$
\tilde{\boldsymbol{H}} = \boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_h\boldsymbol{F} \quad \text{et} \quad \tilde{\boldsymbol{D}} = \boldsymbol{F}^{\dagger}\boldsymbol{\Lambda}_d\boldsymbol{F} \tag{6}
$$

as it was shown in the exercises. The matrix $\boldsymbol{\Lambda}_h$ is diagonal and its elements on the diagonal are the eigenvalues of $\boldsymbol{H}$. The eigenvalues are readily obtained by taking the FFT of the first row of $\boldsymbol{H}$, that is calculating the N-point FFT of the impulse response which represent samples of the frequency response. The same applies for the matrix $\tilde{\boldsymbol{D}}$ and its eigenvalues by replacing the impulse response with $[-1, 1]$.

2. *By replacing (6) in (5) and using simple matrix manipulations, show that:*

$$
\mathring{\widehat{\boldsymbol{x}}} = (\boldsymbol{\Lambda}_h^{\dagger}\boldsymbol{\Lambda}_h + \mu\boldsymbol{\Lambda}_d^{\dagger}\boldsymbol{\Lambda}_d)^{-1}\boldsymbol{\Lambda}_h^{\dagger}\mathring{\boldsymbol{y}}\,, \tag{7}
$$

and give $\mathring{\boldsymbol{y}}$ and $\mathring{\widehat{\boldsymbol{x}}}$. Hint: $\tilde{\boldsymbol{H}}$ is a real matrix, so it is also its complex conjugate, and $\tilde{\boldsymbol{H}} = \tilde{\boldsymbol{H}}^{\dagger}$; same idea for $\tilde{\boldsymbol{D}}$. Analyse and comment upon the expression of the criterion and of the minimiser for the case $\mu = 0$.

To finalise our discussion, we shall first construct the vector $\boldsymbol{g}_{\text{PLS}}$ whose components are given by

$$g^n_{\text{PLS}} = \frac{\mathring{h}^*_n}{|\mathring{h}_n|^2 + \mu|\mathring{d}_n|^2} \quad \text{for } n = 1, 2, \ldots, N \tag{8}$$

FR ref

such that the vector $\mathring{\widehat{\boldsymbol{x}}}$ is obtained by performing the element-wise product between the vectors $\boldsymbol{g}_{\text{PLS}}$ and $\mathring{\boldsymbol{y}}$:

$$\mathring{\widehat{\boldsymbol{x}}} = \boldsymbol{g}_{\text{PLS}} \;.\!*\; \mathring{\boldsymbol{y}}\,. \tag{9}$$

Observe that the deconvolution problem writes as a filtering operation carried out in the Fourier domain, for which $\boldsymbol{g}_{\text{PLS}}$ represents the discrete transfer function.

The deconvolution problem is summarised in what follows:

1. Construct $\mathring{\boldsymbol{h}}$ as the N-point FFT of the impulse response.
2. Construct $\mathring{\boldsymbol{d}}$ as the N-point FFT of $[1; -1]$.
3. Construct the vector $\boldsymbol{g}_{\text{PLS}}$ containing the transfer function (8).
4. Construct $\mathring{\boldsymbol{y}}$ as the FFT of the observation.
5. Compute $\mathring{\widehat{\boldsymbol{x}}}$ as the product between the transfer function $\boldsymbol{g}_{\text{PLS}}$ and $\mathring{\boldsymbol{y}}$.
6. Compute the inverse FFT of $\mathring{\widehat{\boldsymbol{x}}}$ to obtain the solution $\widehat{\boldsymbol{x}}$ in the spatial domain.

2 Implementation

2.1 2D Approach

Equations similar to the one for the 1D case could have also been given for the 2D case. However, the matrices that are involved have a more complex structure: block-Tœplitz with each block having itself a Tœplitz structure. This makes the circulant approximation in both directions much more difficult to handle. As such, the 2D case is presented just as an extension of the 1D case in what concerns the *Matlab* implementation. The following remarks apply:

- The images, the impulse response, the regularisation term are two-dimensional, this means that one has to use FFT-2D instead of FFT.
- More exactly, if the image to restore has N rows and N columns, then the FFT-2D must be computed on N rows and N columns.
- The frequency transfer function is also two-dimensional, with one dimension for each spatial frequency.

Remark 1 — *Under no circumstances the matrices $\boldsymbol{H}$ and $\boldsymbol{D}$ are to be constructed in the Matlab code.*

2.2 The observed image

The first step is to load the data given in the files `Data1` and `Data2`. This is accomplished with the use of the function `load`. Each file contains: the blurred image (`Data`), the true image (`TrueIma`), for comparison purposes, and the impulse response of the convolution filter (`IR`). One should now take time to analyse each data set on its own and with respect to the other.

3. *One can use the following sequence of instructions in order to display the blurred image:*

```
% Clean the workspace and load the data
    close all, clear all
    load Data1
    %load Data2

% Create a window to display the image
    figure(1), clf

% Display the image, grayscale
    imagesc(Data);
    colormap('gray');colorbar

% Scale the axes and eliminate the grading
    axis('square','off')
```

Comment upon the degree of blurriness of each observed image. Analyse the images in the frequency domain (using both a linear and a logarithmic scale) and provide appropriate comments. Take the time to identify the two frequency axes, which must be properly graded, the zero frequency, the low frequencies, the high frequencies,. . . . Properly present these results in your report.

4. *Display the two impulse responses $h_{n,m}$ and their associated transfer functions $H(\nu_x, \nu_y)$, first using the function `imagesc` and then using the function `plot`. Properly grade the axes in either case. Properly present these results in your report. What is the type of each filter? High-pass or low-pass? What are the differences between the two filters?*

2.3 Inversion

The *Matlab* code shall implement the deconvolution, in the 2D case, using the penalized least squares method with a quadratic penalisation term. The criterion is to be minimised using the circulant approximation presented in the section 1 and summarized on page 4.

In what concerns the regularisation term, it relies on differences between neighboring pixels on comlumns and rows of the image. It reads:

$$\|\boldsymbol{D}\boldsymbol{x}\|^2 = \sum_{n,m} (x_{n,m} - x_{n,m+1})^2 + (x_{n,m} - x_{n+1,m})^2$$

and it will be implemented based on two filters

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

that compute differences between pixels on columns and rows. One can also use a unique filter with the following impulse responses

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ or } \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \text{ or } \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

that implement various approximations of gradient of images.

5. *Implement the deconvolution inside a function which takes as input the observed data, the impulse response and the regularisation parameter.*

6. *First consider the case of the simple inverse filter, i.e., $\mu = 0$. Analyse and comment the results one obtains for either data set.*

7. *Analyse the results one obtains for various values of μ on a $\log_{10}$ scale. Comment the obtained results both in the spatial domain and in the frequency domain. How does the value of μ influence the deconvolved image with respect to each data set? By trial and error and through a visual examination of the results determine an appropriate value for μ.*

2.4 Role of the hyper-parameters

The previous point allowed one to asses the inherent difficulty associated with the deconvolution problem. It also showed that better results can be obtained by taking into account available information with respect to the expected regularity of the reconstructed image. This approach thus allows one to perform a compromise between the two sources of information: the observed data and the available *a priori* information regarding regularity. This is carried out through the value of the parameter μ. In this work, its value is chosen empirically such that the deconvolved image is neither too smooth nor too irregular.

In the "toy example" investigated here, the true image is known, *i.e.,* the image from which the observed data has been generated is known. Thus it is possible to compute the numerical difference between the deconvolved image $\widehat{\boldsymbol{x}}$ and the true image $\boldsymbol{x}^\star$ as a function of the regularisation parameter μ. The following three distance functions are to be considered for that purpose:

$$
\begin{aligned}
\Delta_2(\mu) &= \frac{\sum_{p,q}(\widehat{x}_{p,q}(\mu) - x^\star_{p,q})^2}{\sum_{p,q}(x^\star_{p,q})^2} = \frac{\|\widehat{\boldsymbol{x}}(\mu) - \boldsymbol{x}^\star\|^2}{\|\boldsymbol{x}^\star\|^2} \\
\Delta_1(\mu) &= \frac{\sum_{p,q}|\widehat{x}_{p,q}(\mu) - x^\star_{p,q}|}{\sum_{p,q}|x^\star_{p,q}|} = \frac{\|\widehat{\boldsymbol{x}}(\mu) - \boldsymbol{x}^\star\|_1}{\|\boldsymbol{x}^\star\|_1} \\
\Delta_\infty(\mu) &= \frac{\max_{p,q}(|\widehat{x}_{p,q}(\mu) - x^\star_{pq}|)}{\max_{p,q}|x^\star_{pq}|} = \frac{\|\widehat{\boldsymbol{x}}(\mu) - \boldsymbol{x}^\star\|_\infty}{\|\boldsymbol{x}^\star\|_\infty}
\end{aligned}
$$

The distances is near to 0 as the restored image is similar to the true image and close to 1 when the restored image is zero everywhere.

8. *Compute the values of the distances for a set of values for μ, for example the set of logarithmic spaced values between 10^{-10} and 10^{10}. Give the values for μ which minimise each distance function and compare them to previously empirically chosen value. One should mainly focus only on the second data set* `Data2`.