Signal processing
Lab 3: Wiener deconvolution

Maxim Berman – maxim.berman@ecp.fr

Friday 27/02/2015

Abstract

This lab will build upon the work you did on signal denoising in Lab 2. This time, we will be interested in finding the optimal Wiener filter of a given signal to perform denoising. The result will be compared with the best linear gaussian filter optimizing the SNR ratio. Finally, we will use *iterative Wiener filtering* in order to study a plausible implementation, where we would not be given the ground truth image.

Deliverable: To be submitted before Friday 13/03/2015.

1 Wiener Filter Theory

In image denoising, we are given an image $\mathbf{y}$ which is a noisy observation of a "ground truth" image $\mathbf{x_0}$. The goal is to reconstruct an image $\mathbf{x}$ from $\mathbf{y}$ that recovers the original image $\mathbf{x_0}$ as closely as possible, removing the effect of the noise.

In Lab 2, we have seen how, given a model of white noise with variance σ_n^2, we can empirically find the best linear gaussian filter of variance σ_r^2 optimizing the PSNR ratio of the recovered image $\mathbf{h} \star \mathbf{y}$ with respect to the ground truth $\mathbf{x_0}$.

In this lab, we will see how, when we are given a model not only for the noise but also for the signal, we can theoretically determine the optimal filter to use to optimize the PSNR, the *Wiener filter*.

Remember the expression of the PSNR for images encoded as a grayscale vector in $[0,1]^{n_x n_y}$:

$$\text{PSNR} = -10 \cdot \log_{10} \left(\frac{1}{n_x n_y} \sum_{i,j} (x^{i,j} - x_0^{i,j})^2 \right).$$

Obviously maximizing the PSNR is equivalent to minimizing the squared error or L^2 norm

$$\|\mathbf{x} - \mathbf{x_0}\|^2 = \sum_{i,j} (x^{i,j} - x_0^{i,j})^2. \tag{1}$$

We will again use a model of additive noise $\mathbf{y} = \mathbf{x} + \boldsymbol{\xi}$. In a white noise model $\boldsymbol{\xi}$ is a matrix of independent gaussian variables of variance σ_n^2 – however we stay in the general case for now. As in Lab 2, we are interested in finding a translation-invariant linear denoising operator parametrized by a convolutional kernel $\mathbf{h}$, such that the reconstruction

$$\mathbf{x} = \mathbf{h} \star \mathbf{y} \tag{2}$$

is as close to x_0 under the L^2 norm (1). In order to give a theoretical solution, this time we consider that the ground truth image $\mathbf{x_0}$ is itself the realization of a random process. We are now interested in minimizing the expected value of the norm (1):

$$R(\mathbf{h}) = \mathbb{E}_{\mathbf{w},\mathbf{x0}}(\|\mathbf{x} - \mathbf{x_0}\|^2) = \mathbb{E}_{\boldsymbol{\xi},\mathbf{x0}}(\|\mathbf{h} \star \mathbf{y} - \mathbf{x_0}\|^2)$$

with respect to $\mathbf{h}$. Under our additive noise model the risk writes as

$$R(\mathbb{h}) = \mathbb{E}_{\boldsymbol{\xi},\mathbf{x0}}(\|\mathbf{h} \star (\mathbf{x_0} + \boldsymbol{\xi}) - \mathbf{x_0}\|^2). \tag{3}$$

Remember Parseval's theorem for Fourier transforms: the squared $L2$-norm of a function $f(x,y)$ is proportional to the squared L2-norm of its Fourier transform $\widehat{f}(\omega_u, \omega_v)$:

$$\|f(x,y)\|^2 = C \cdot \|\widehat{f}(\omega_u, \omega_v)\|^2; \tag{4}$$

the proportionality constant depend on the normalization used for the Fourier transform and is not important here.

Question 1.1 Show that *under the assumption of independence between* $\mathbf{h}$ *and* $\mathbf{x_0}$, minimizing the risk (3) accounts to minimizing the risk in Fourier space

$$\widetilde{R}(\mathbf{h}) = (1 - \hat{\mathbf{h}})(1 - \hat{\mathbf{h}})^* \mathbb{E}_{\boldsymbol{\xi},\mathbf{x0}}(\|\widehat{\mathbf{x_0}}\|^2) + \hat{\mathbf{h}}\hat{\mathbf{h}}^* \mathbb{E}_{\boldsymbol{\xi},\mathbf{x0}}(\|\hat{\boldsymbol{\xi}}\|^2). \tag{5}$$

One shows that the solution to this optimization problem is

$$\widehat{h}(\omega_u, \omega_v) = \frac{P_{x_0}(\omega_u, \omega_v)}{P_{x_0}(\omega_u, \omega_v) + P_{\xi}(\omega_u, \omega_v)} \tag{6}$$

where $P_{x_0} = \mathbb{E}(|x_0(\omega_u, \omega_v)|^2)$ and $P_\xi = \mathbb{E}(|\xi(\omega_u, \omega_v)|^2)$ are the power spectral density of the source and the noise.

Question 1.2 Interpret this expression: what is the Wiener filter doing?

In the case of white gaussian noise the power spectral density is the variance σ_n^2. In order to compute the Wiener filter, the spectral density of the source source can be approximated by the particular realization x_0 that we have, as

$$P_{x_0}(\omega_u, \omega_v) \simeq \frac{1}{N} |\hat{x}_0(\omega_u, \omega_v)|^2$$

where $N = n_x n_y$ is the number of pixels in the image.

Again, note that this Wiener filter needs the ground truth image x_0 to be computed: in real applications, one may not have access to this unperturbed signal.

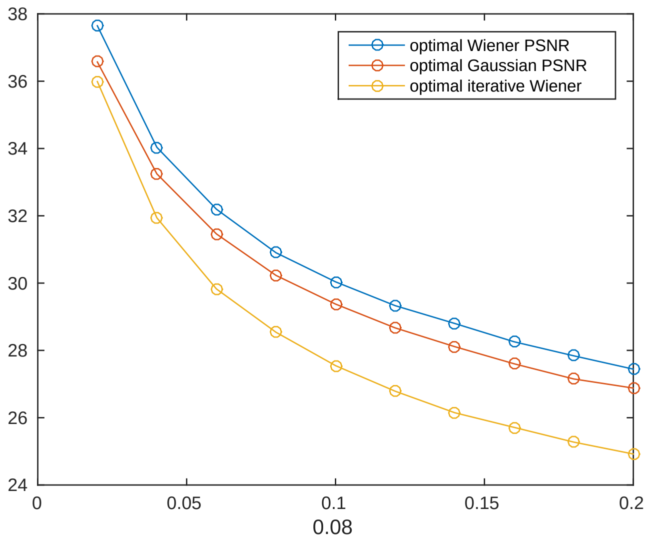

Figure 1: Optimal PSNR attained by a Wiener and a Gaussian linear filter, and by an iterative Wiener filter process.

2 Implementation

The script provided loads the same image as in Lab 2 and adds some white noise to it.

Question 2.1 Write a function `h = wiener_filter(x0, sigma_n)` that takes an image as argument and compute its associated Wiener convolutional kernel for a gaussian white noise of standard deviation σ_n. *You may use Matlab functions `fft2` and `ifft2`.*

Question 2.2 Use the provided code to visualize the image denoised by Wiener filtering. Include this image and the value of the PSNR in your report. Is the noise completely removed? Do we recover the informations contained in $x0$?

Question 2.3 Visualize the kernel **h** using `imagesc`. Make sure you center the filter using `fftshift`.

Question 2.4 Write a function `PSNR = best_wiener_psnr(x0, sigma_n)` to compute the PSNR atteigned by the Wiener filter on an input image **x0** using additive gaussian noise of parameter σ_n. Use the provided code to visualize the evolution of the best Wiener PSNR with the noise.

Question 2.5 Adapt the code from Lab 2 in order to write a function `PSNR = best_gaussian_psnr(x0, sigma_n)` to compute the PSNR atteigned by the optimal gaussian filter. Superpose the plot obtained with the plot obtained in question 2.4. You should obtain the plot of Figure 1 (without the yellow curve which shall be computed in the next section).

Question 2.6 What are the limits of our method? Can you criticize the assumptions made in deriving the Wiener filter?

3 Iterative Wiener filtering

In this section, we will implement Wiener filtering in a real application, where we do not have access to the ground truth **x0**. We may rely on approximation methods to estimate the power spectrum of **x0** from the noisy observation, but this requires further modeling of the signal. Another approach is to iteratively apply Wiener filtering.[1] We first evaluate the power spectrum of the noisy observation to compute the Wiener filter, and then we iteratively compute a Wiener filter on the ouput. This converges to a fixed point which is not an optimal Wiener filtered image; however a good approximation can be found after only a few iterations.

Question 3.1 We suppose that σ_n is known: on a commercial camera, this standard deviation could e.g. be estimated on calibration images. Implement Iterative Wiener filtering: start from the approximation $\mathbf{x}^{(\mathbf{0})} = \mathbf{y}$ and then iteratively compute and apply the Wiener filter on $\mathbf{x}^{(\mathbf{i})}$ to obtain the approximation $\mathbf{x}^{(\mathbf{i+1})}$. Show the images obtained over the first 5 iterations.

Question 3.2 Plot the graph of the PSNR of $\mathbf{x}^{(\mathbf{i})}$ as a function of the iteration i (up to 30 iterations). Use this result and qualitative observation of the images observed in **Question 3.1** to show that we are only interested in doing a few iteratons of iterative Wiener filtering.

Question 3.3 Write a function `PSNR = best_it_wiener_psnr(x0, sigma_n)` that finds the best value of the PSNR attained by iterative Wiener filtering for an input image $x0$ and a white noise of parameter σ_n. Adapt the code given previously to superimpose this plot to the two other similar curves, as in Figure 1.

Question 3.4 One sees that Iterative Wiener filtering performs worse than the two other denoising methods: why is it still interesting?

[1] This iterative Wiener filtering of images is introduced in Hillery, A. D., & Chin, R. T. (1991). Iterative Wiener filters for image restoration. *IEEE Transactions on Signal Processing*, 39(8), 1892-1899.