Università del Salento and INFN Lecce

# Introduction to Wiener Filtering

*Mosè Giordano*

26 November 2014

# The purpose of Wiener filtering

Reduce degradation and noise in images, audio signals, etc.


Picture of the Moon taken by the Galileo spacecraft on 7 December 1992

# The theory

Suppose you have a signal $S(\boldsymbol{t})$ in the time domain (whatever "time" is: actual time, point in space, pixel of an image, etc), degraded by known blurring shift-invariant function $B(\boldsymbol{t})$ and additive noise $N(\boldsymbol{t})$

$$X(\boldsymbol{t}) = (B * S)(\boldsymbol{t}) + N(\boldsymbol{t})$$

The Fourier transform in the frequency domain of this degraded signal $X(\boldsymbol{t})$ is

$$\hat{X}(\boldsymbol{f}) = \mathcal{F}(X)(\boldsymbol{f}) = \hat{B}(\boldsymbol{f})\hat{S}(\boldsymbol{f}) + \hat{N}(\boldsymbol{f})$$

# The theory (cont.)

We want to find an appropriate filter $W(\boldsymbol{f})$ such that the function $W(\boldsymbol{f})\hat{X}(\boldsymbol{f})$ is as much close as possible to the Fourier transform of the original signal $\hat{S}(\boldsymbol{f}) = \mathcal{F}(S)(\boldsymbol{f})$, i.e. we want to minimize the quantity

$$\left\langle W(\boldsymbol{f})\hat{X}(\boldsymbol{f}) - \hat{S}(\boldsymbol{f})\right\rangle = \left\langle W(\boldsymbol{f})(\hat{B}(\boldsymbol{f})\hat{S}(\boldsymbol{f}) + \hat{N}(\boldsymbol{f})) - \hat{S}(\boldsymbol{f})\right\rangle$$

This condition is fulfilled by

$$W(\boldsymbol{f}) = \frac{\hat{B}^*(\boldsymbol{f})}{|\hat{B}(\boldsymbol{f})|^2 + |\hat{N}(\boldsymbol{f})|^2/|\hat{S}(\boldsymbol{f})|^2}$$

This is the Wiener filter function and $\mathcal{F}^{-1}(W\hat{X})(\boldsymbol{t})$ is the filtered signal. This function gives more importance to frequencies with higher signal to noise ratio. In absence of blurring

$$W(\boldsymbol{f}) = \frac{|\hat{S}(\boldsymbol{f})|^2}{|\hat{S}(\boldsymbol{f})|^2 + |\hat{N}(\boldsymbol{f})|^2}$$

# The theory (cont.)

Theoretically, in order to calculate the Wiener filter function we need to know

- the original signal
- the blurring function
- the noise

or at least their power spectra. Actually, power spectra need not to be known exactly (noise power spectrum can often be easily estimated, e.g. white noise has constant spectrum) because

- most signals of the same class have fairly similar power spectra
- the Wiener filter is insensitive to small variations in the original signal power spectrum

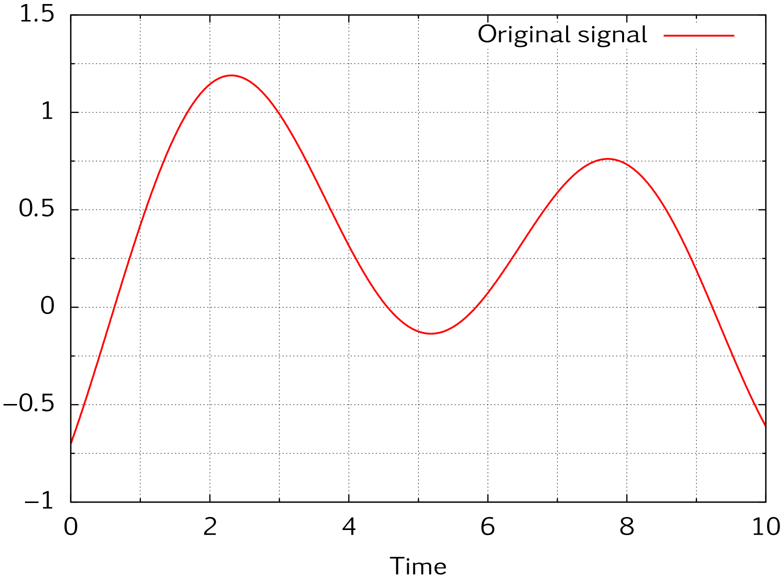
We can estimate the original signal power spectrum using a representative of the class of signals being filtered

# Wiener filter applied to a temporal signal

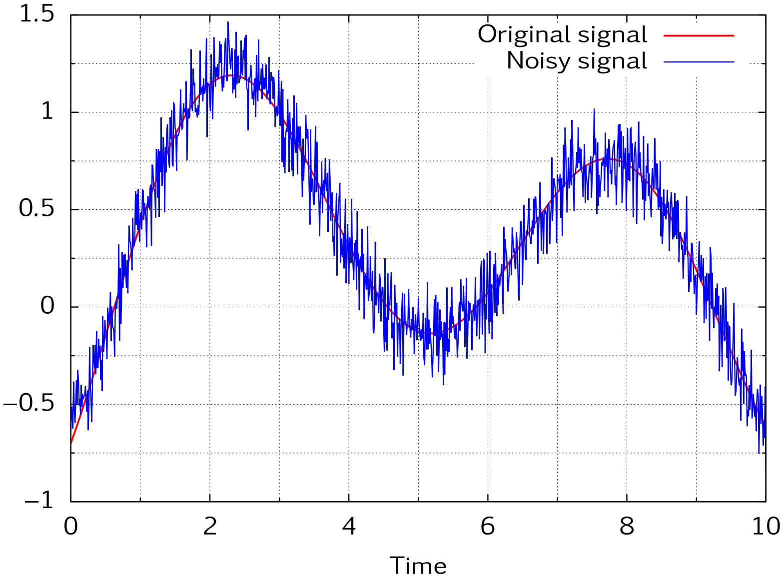
IDL/GDL code:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;  Temporal Signal  ;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
ntime = 1001
ff = 100
;; Define the times array.
time = findgen(ntime)/ff
;; The orignal signal.
sign = sin(time) - 0.7*cos(0.7*time) + 0.5*sin(0.5*time)^2
;; The noise.
noise = 0.3*randomu(null, ntime)*cos(10*randomu(null, ntime)*time)
;; Signal + noise.
sign_noise = sign + noise
;; Its Fourier transform.
ft = fft(sign_noise)

;; Determine the power spectra of the signal and the noise.
signal_power_spectrum = abs(fft(sign))^2
noise_power_spectrum = abs(fft(noise))^2
;; Calculate the Wiener filter.
filter = signal_power_spectrum/(signal_power_spectrum + noise_power_spectrum)
;; Get the filtered signal + noise.
result = fft(ft*filter, /inverse)
```

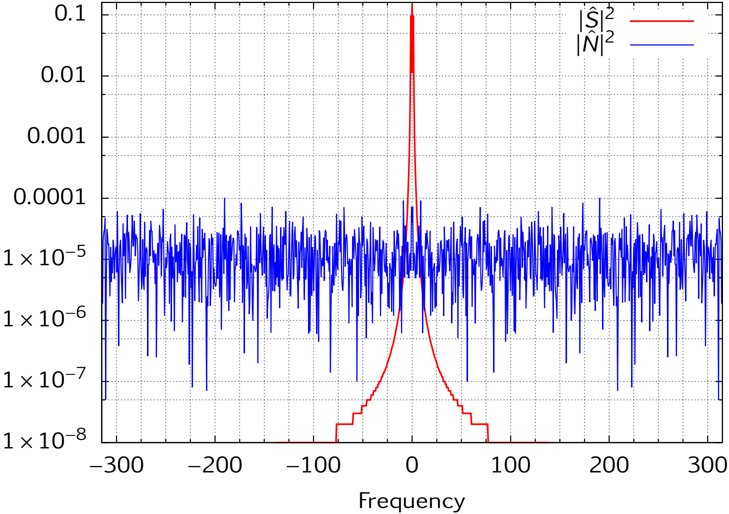
# Wiener filter applied to a temporal signal (cont.)



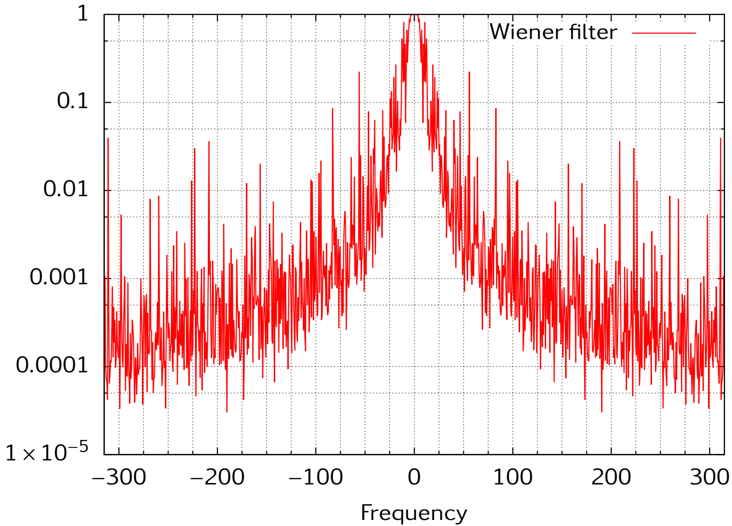
# Wiener filter applied to a temporal signal (cont.)



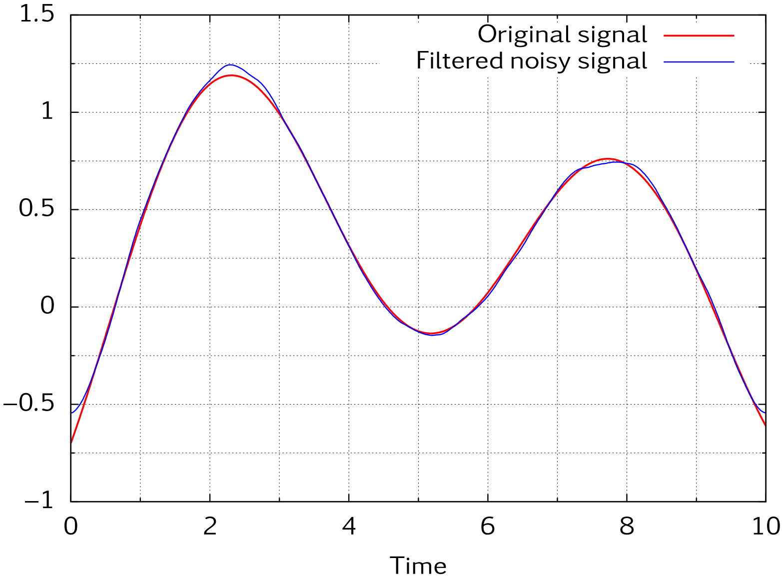
# Wiener filter applied to a temporal signal (cont.)



# Wiener filter applied to a temporal signal (cont.)



# Wiener filter applied to a temporal signal (cont.)



# Wiener filter applied to an image

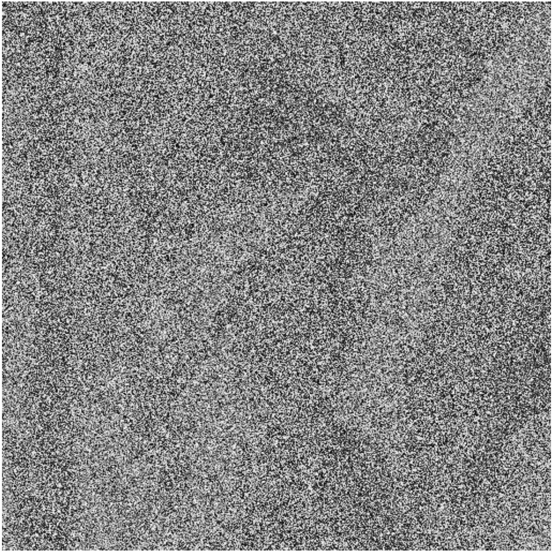
IDL/GDL code:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;   Image   ;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Read the Lena image.
read_jpeg, "lena.jpg", lena, /grayscale
;; Add a large noise to Lena.
img_noise = 1.8d*mean(lena)*randomu(systime(/seconds), 512, 512)
degraded_img = lena + img_noise
;; Fourier transform of the degraded image.
ftimg = fft(degraded_img)

;; For the Wiener filter, use a completely different picture.
read_jpeg, "elaine.jpg", elaine, /grayscale
;; Determine the power spectrum of Elaine.
elaine_power_spectrum = abs(fft(elaine))^2
;; To further increase entropy, calculate a new noise.
img_noise_new = 2d*mean(lena)*randomu(systime(/seconds), 512, 512)
;; Power spectrum of the new noise.
img_noise_power_spectrum = abs(fft(img_noise_new))^2

;; Calculate the Wiener filter.
filter = elaine_power_spectrum/(elaine_power_spectrum + $
                                img_noise_power_spectrum)
;; Get the filtered picture.
result_img = fft(ftimg*filter, /inverse)
```

# Wiener filter applied to an image (cont.)

The original image, file `lena.jpg`

# Wiener filter applied to an image (cont.)



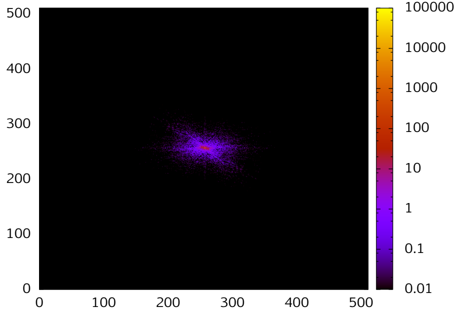
The image has been degraded with a large noise

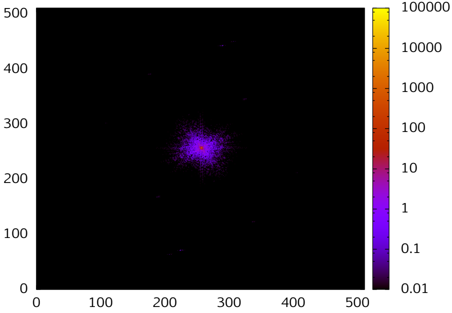
# Wiener filter applied to an image (cont.)

The picture used to filter the degraded image, file `elaine.jpg`

# Wiener filter applied to an image (cont.)

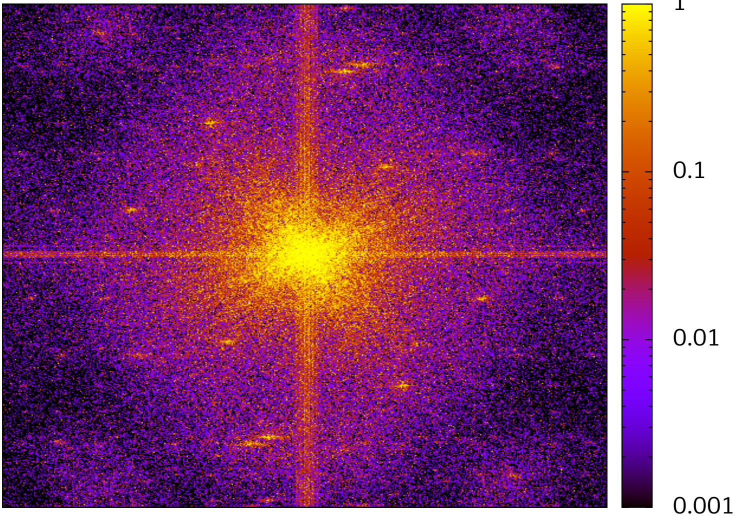
We can use a different picture to filter the degraded image because they have similar power spectra




Left: power spectrum of `lena.jpg` file; right: power spectrum of `elaine.jpg` file

# Wiener filter applied to an image (cont.)




The Wiener filter function in the frequency domain

# Wiener filter applied to an image (cont.)

The filtered image

# References and further reading


- S. Eddins. *Image deblurring – Wiener filter.* Nov. 2, 2007. URL: http://blogs.mathworks.com/steve/2007/11/02/image-deblurring-wiener-filter/.
- W. Press. *Computational Statistics with Application to Bioinformatics – Unit 19: Wiener Filtering (and some Wavelets).* 2008. URL: http://www.nr.com/CS395T/lectures2008/19-WienerFiltering.pdf.
- E. Turkel. *Summary Wiener Filter.* 2004. URL: http://www.math.tau.ac.il/~turkel/notes/wiener7-2.pdf.
- *WIENER_FILTER (IDL Reference).* URL: http://www.exelisvis.com/docs/WIENER_FILTER.html.