TP de Technologies de l'Information

Restauration et filtrage de Wiener

Prof. Jean-Philippe THIRAN
JP.Thiran@epfl.ch

- Restoration : invert non-wanted effects
- Typical application: deconvolution
  - Let us consider the ideal image f_i that has been degraded by an undesired (low pass) filtering effect
  - Let f_o be the observed image
  - Moreover, there is an additive noise n,

$$f_o(x,y) = f_i(x,y) ** h_D(x,y) + n(x,y)$$

- Goal : try to restore the initial image, using a model for the original image and for the noise

Image restoration: inverse filtering

- Inverse filtering: let us find a filter h_R that will best restore the image f_i
- The restored image will thus be

$$\hat{f}_i(x,y) = f_o(x,y) ** h_R(x,y)$$

- By substitution in the previous equation, we get

$$\hat{f}_i(x,y) = [f_i(x,y) ** h_D(x,y) + n(x,y)] ** h_R(x,y)$$

- By FT:

$$\hat{F}_i(\omega_x,\omega_y) = [F_i(\omega_x,\omega_y) H_D(\omega_x,\omega_y) + N(\omega_x,\omega_y)] H_R(\omega_x,\omega_y)$$

- Thus, the solution consists in taking a filter h_R with a frequency response inverse of that of h_D :

$$H_R(\omega_x,\omega_y) = \frac{1}{H_D(\omega_x,\omega_y)}$$

- The spectrum of the restored image is thus

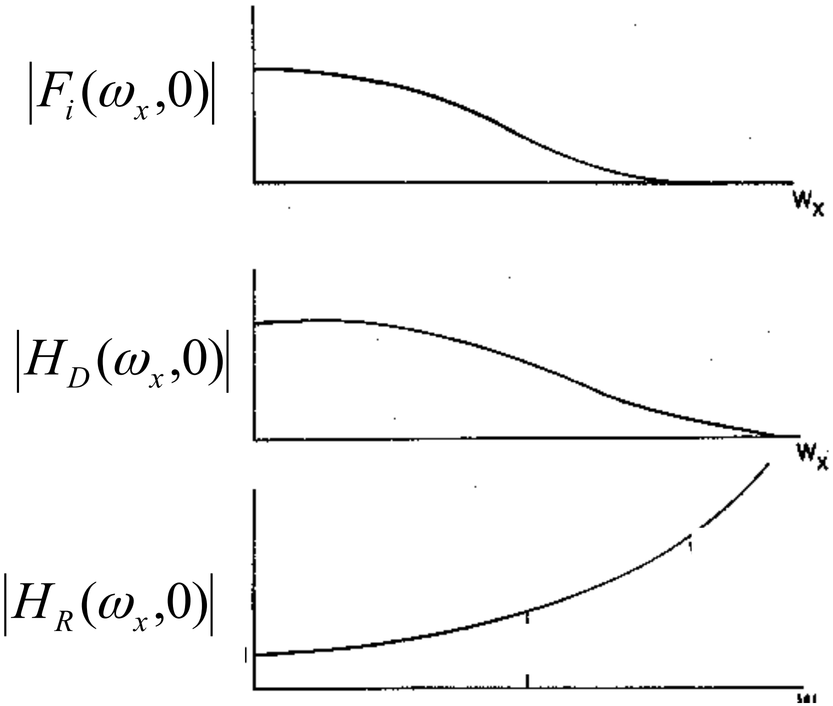
$$\hat{F}_i(\omega_x,\omega_y) = F_i(\omega_x,\omega_y) + \frac{N(\omega_x,\omega_y)}{H_D(\omega_x,\omega_y)}$$

- And by inverse FT, the restored image will be

$$\hat{f}_i(x,y) = f_i(x,y) + \frac{1}{4\pi^2}\iint_{-\infty}^{\infty} \frac{N(\omega_x,\omega_y)}{H_D(\omega_x,\omega_y)} e^{j(\omega_x x+\omega_y y)} dxdy$$

Image restoration: inverse filtering

- Without noise, the restoration is perfect
- With noise, the error can be important:
  - Often h_D will be a low-pass filter (blur, …)
  - Noise will thus be amplified



- Example

Original

Blured image (filtered)

Noisy and blured

- Example (cont.)

Restoration of the blured image

Restoration of the noisy blured image

Image restoration: Wiener filtering

- The previous problem comes from the fact that the filter ignores the presence of noise in the signal
  - solution : Wiener filtering, that considers both a model of the image and of the noise
- Wiener filtering: hypotheses :
  - Images are 2D random variables, with zero mean (can be obtained by subtracting the mean to the images)
- Goal: find a filter h_R that will minimize the quadratic error

$$\varepsilon = E\left\{\left[f_i(x,y) - \hat{f}_i(x,y)\right]^2\right\}$$

- Calculating the 1st derivative, the error is minimal when

$$E\left\{\left[f_i(x,y)-\hat{f}_i(x,y)\right]f_o(x',y')\right\}=0$$

- By replacing $\hat{f}_i$ by its value, we get

$$E\left\{f_i(x,y)f_o(x',y')\right\}=\int_{-\infty}^{\infty}\int E\left\{f_o(i,j)f_o(x',y')\right\}h_R(x-i,y-j)didj$$

- The expectations of this products are the intercorrelation and the autocorrelation of the variables:

$$K_{f_if_o}(x-x',y-y')=\int_{-\infty}^{\infty}\int K_{f_o}(i-x',j-y')\,h_R(x-i,y-j)didj$$

Image restoration: Wiener filtering

$$K_{f_i f_o}(x-x', y-y') = \int_{-\infty}^{\infty}\int K_{f_o}(i-x', j-y')\, h_R(x-i, y-j)\,di\,dj$$

- By FT, we obtain

$$H_R(\omega_x, \omega_y) = \frac{P_{f_i f_o}(\omega_x, \omega_y)}{P_{f_o}(\omega_x, \omega_y)}$$

$P_{f_i f_o}(\omega_x, \omega_y)$ is the power interspectrum

$P_{f_o}(\omega_x, \omega_y)$ is the power spectrum of f_o

- When the noise is additive, we can write, by the Wiener-Kintchine theorems :

$$P_{f_o}(\omega_x,\omega_y) = \left|H_D(\omega_x,\omega_y)\right|^2 P_{f_i}(\omega_x,\omega_y) + P_N(\omega_x,\omega_y)$$

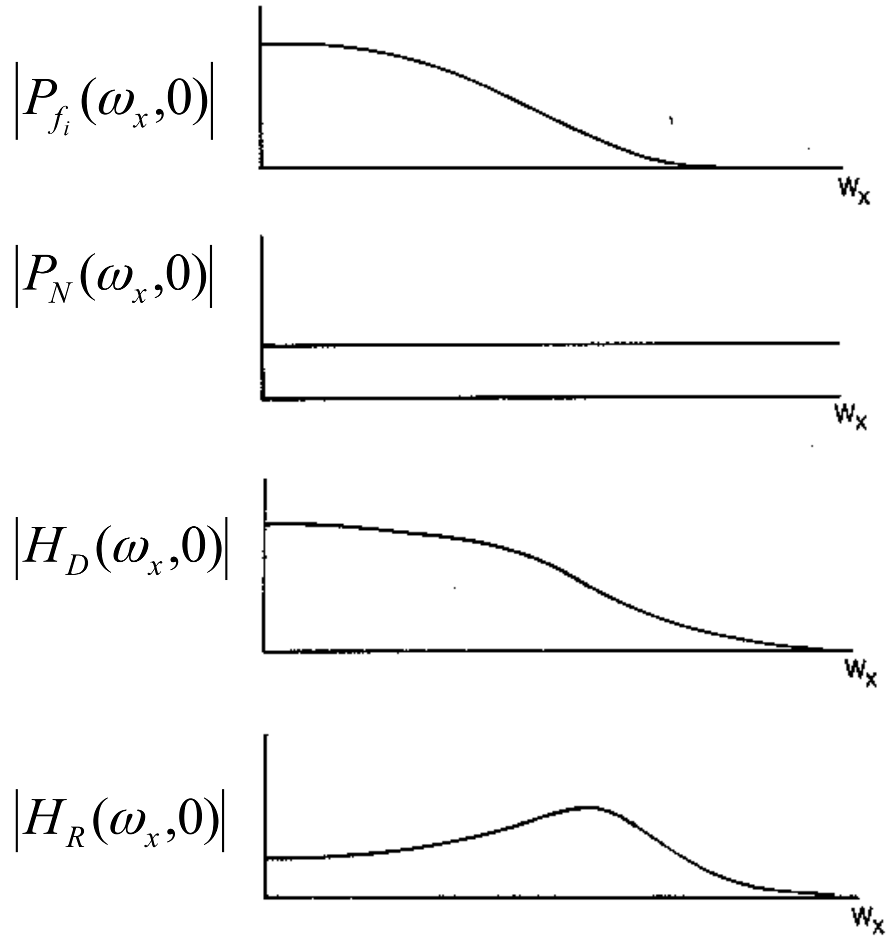
and

$$P_{f_o f_i}(\omega_x,\omega_y) = H_D^*(\omega_x,\omega_y) P_{f_i}(\omega_x,\omega_y)$$

- And we finally obtain the Wiener filter, with frequency response:

$$H_R(\omega_x,\omega_y) = \frac{H_D^*(\omega_x,\omega_y)}{\left|H_D^*(\omega_x,\omega_y)\right|^2 + \dfrac{P_N(\omega_x,\omega_y)}{P_{f_i}(\omega_x,\omega_y)}}$$

- Conclusions :
  - The Wiener filter is a **adaptive band-pass filter**
  - It behaves like the inverse filter at low frequencies and like a low-pass filter for high frequencies



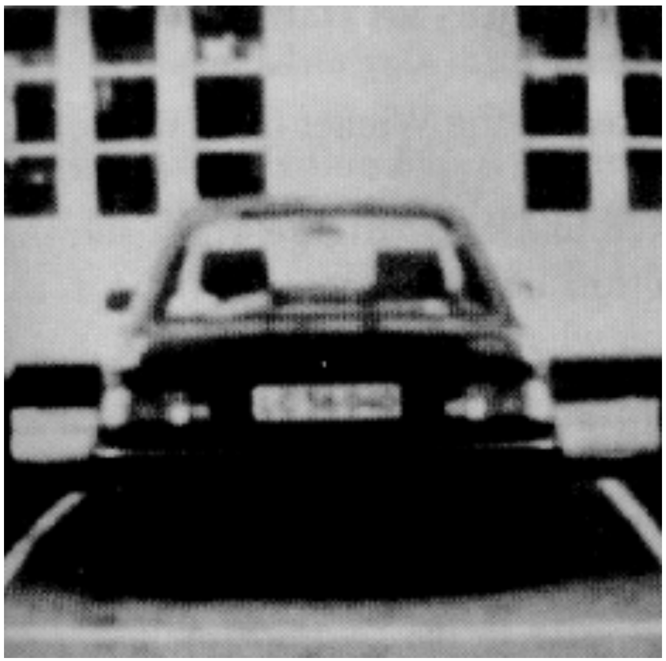
- Examples :

Motion blur

Restored Image

Image restoration: Wiener filtering

- Examples (cont.):



Out-of-focus blur

Restored image