# Digital Image Processing

14 December 2006

**Dr. ir. Aleksandra Pizurica**
Prof. Dr. Ir. Wilfried Philips

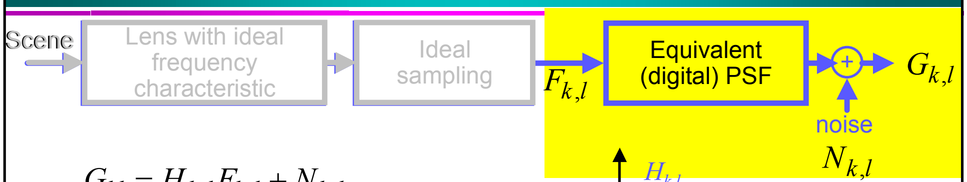
Aleksandra.Pizurica @telin.UGent.be
Tel: 09/264.3415

UNIVERSITEIT
GENT
**Telecommunicatie en Informatieverwerking**

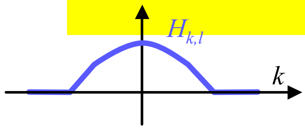
INSTITUUT VOOR PERMANENTE VORMING

# Image Deblurring

# …Degradation model




$$G_{kl} = H_{k,l} F_{k,l} + N_{k,l}$$

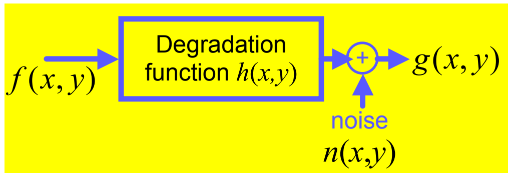



equivalent

equivalent

$$g(x_k, y_k) = g(x_k, y_k) * f(x_k, y_k) + n(x_k, y_k)$$

For compactness we write $x,y$ instead of $x_k,y_k$




Digital filter: models imperfections in the lens, form of the pixels, …
(if aliasing appears equivalent with analogue PSF may not hold)





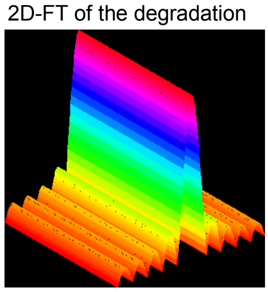

# Example: linear blur

Linear blur arises e.g., due to linear **motion of a camera**: a given pixel value is an aggregation of pixel intensities from points along the line of the camera's motion.

The degradation function is in this case a **box** function and the resulting spectrum is **sinc** function (sinc(x)=sin(x)/x)

31-Point horizontal blur

2D-FT of the degradation



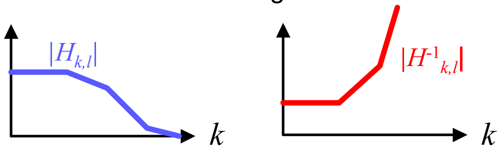
# Inverse filtering…

$$G_{kl} = H_{k,l}F_{k,l} + \cancelto{0}{N_{k,l}}$$ suppose noise can be neglected

A naive solution is to filter by dividing the 2-dimensional fft of the blurred function by the reciprocal of $H_{k,l}$

$$G_{kl} / H_{k,l} = H_{k,l}F_{k,l} / H_{k,l} = F_{k,l}$$

This is commonly reffered to as the inverse filtering method where $1/H_{k,l}$ is the inverse filter.




In practice:

- $H_{k,l}$ is low-pass $\Rightarrow$ inverse filter is high-pass
- $H_{k,l} \approx 0$ at high frequencies
  - $\Rightarrow$ numerical problems (division by a close-to-zero number)
  - $\Rightarrow$ huge amplification of noise: $N/H$ becomes big even for small $N$

Not applicable, e.g., to motion blur. Why?






# Pseudo inverse filtering…

If there are few values of $H_{k,l}$ which are close to zero then the ideal inverse filter can be approximated with a stabilized version of $1/ H_{k,l}$

$$H_{k,l}^{pinv} = \begin{cases} 1/H_{k,l} & \text{if } |H_{k,l}| > T \\ 0, \text{otherwise} \end{cases}$$

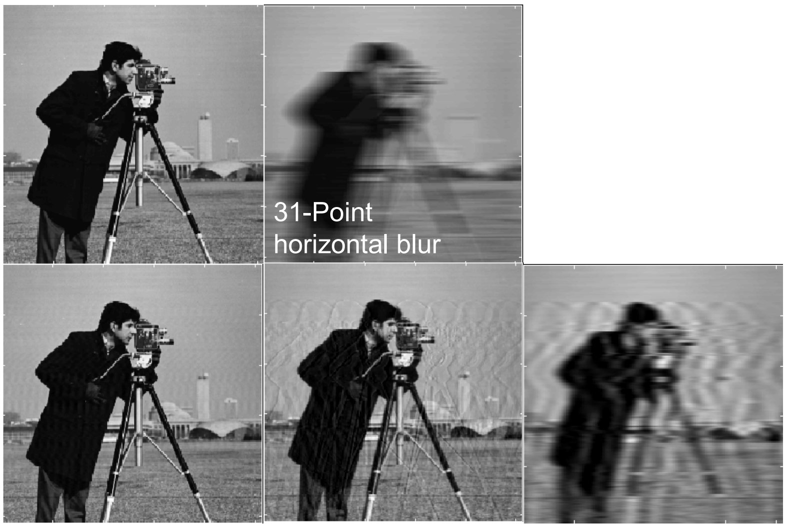
The estimated image spectrum is then

$$\hat{F}_{k,l} = G_{k,l}H_{k,l}^{pinv}$$

This works well only if few elements of $H$ have a magnitude below the threshold.

This filter is hence applicable only in some very specific cases

# …Pseudo inverse filtering




4% values of Hinv=0 44% values of Hinv=0 91% values of Hinv=0






# Wiener filtering

Degradation model $G_{kl} = H_{k,l} F_{k,l} + N_{k,l}$

The Wiener estimate is derived from a least squared method:

$$\hat{F}_{k,l} = \frac{H_{k,l}{}^*}{|H_{k,l}|^2 + S^{\eta}_{k,l} / S^{f}_{k,l}} G_{k,l}$$

$H^*_{k,l}$ - complex conjugate of the degradation function $H_{k,l}$

$S^{\eta}_{k,l}$ - power spectrum of the noise $|N_{k,l}|^2$

$S^{f}_{k,l}$ - power spectrum of the noise $|F_{k,l}|^2$

In practice, we often use:

$$\hat{F}_{k,l} = \frac{H^*_{k,l}}{|H_{k,l}|^2 + K} G_{k,l}$$

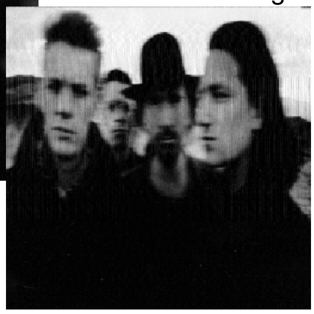
where $K$ is an experimentally optimized constant

# Example of Wiener filtering

original

Blur+noise

after Wiener filtering








# Example of Wiener filtering



Original is first low-pass filtered with Gaussian PSF with $\sigma$=5, then noise is added with standard deviation $\sigma_n$ =5 resp. $\sigma_n$ =10

# Example of Wiener filtering: K=1/100



For big K the filter does not sharpen the image but the noise is suppressed






# Example of Wiener filtering: K=1/500



With decreasing $K$ the high frequencies are amplified
$\Rightarrow$ sharpening, but also more noise

# Example of Wiener filtering: K=1/1000



When the noise level is smaller a better compromise can be found between sharpening and avoiding artefacts






# Example of Wiener filtering: K=1/10000



In this case the Wiener filter is quasi-equal to the inverse filtyer and inverse filter is instable (divisions by 0)

# Wavelet regularized deblurring (WaRD)

A common approach is to combine frequency domain deconvolution (Wiener filter or Richardson-Lucy) with wavelet-domain denoising

Example (WaRD algorithm):

Step 1: Wiener deblurring with small K to sharpen the image

Step 2: Wavelet domain denoising to suppress the excessive noise







# Bayesian Image restoration using MRFs

Assume: $g(x,y) = h(x,y) * f(x,y) + n(x,y)$

data distr.

MAP estimate: $\hat{\mathbf{f}} = \arg\max_{\mathbf{f}\in\Im} p(\mathbf{f}\,|\,\mathbf{g}) \;= \arg\max_{\mathbf{f}\in\Im} p(\mathbf{g}\,|\,\mathbf{f})P(\mathbf{f})$

prior

For white Gaussian noise with zero mean and standard deviation $\sigma$:

$$p(\mathbf{g}\,|\,\mathbf{f}) = A\exp(-\sum_l (g_l - \sum_k h_{l-k} f_k)^2 / 2\sigma^2)$$

MRF prior: $p(\mathbf{f}) = Z\exp(-\sum_{c\in\zeta} V_C(\mathbf{f})) = Z\exp(-\sum_{\langle k,l\rangle} V_2(f_k, f_l))$

over all pairs of pixels

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}\in\Im} \log(p(\mathbf{g}\,|\,\mathbf{f})P(\mathbf{f})) \;= \arg\min_{\mathbf{f}\in\Im} E(\mathbf{f}\,|\,\mathbf{g})$$

posterior energy

$$E(\mathbf{f}\,|\,\mathbf{g}) = \underbrace{\sum_l (g_l - \sum_k h_{l-k} f_k)^2}_{\text{data fit term}} + \lambda \sum_{\langle k,l\rangle} V_2(f_k, f_l)$$

($\lambda$ is a constant)

data fit term

prior knowledge (context)

# Bayesian Image restoration using MRFs

Image restoration using a Markov Random Field prior yields potentially better results than Wiener filtering, provided that a-priori model is well specified

Critical: specification of discontinuity-adaptive clique potentials

Aother problem with these approaches is computation time






# Deblurring summary

- In cases where noise can be neglected a naive approach is inverse filtering
- Inverse filtering does not work because it implies divisions by small numbers (close to zero) and because it boosts noise
- In some very specific cases, pseudo-inverse filtering may work
- A good approach is Wiener filtering
- The results of the Wiener filter can be improved by using a small regularization constant and removing extra noise by a subsequent noise reduction filter
- Image restoration techniques employing MRF priors are powerful but also computationally expensive