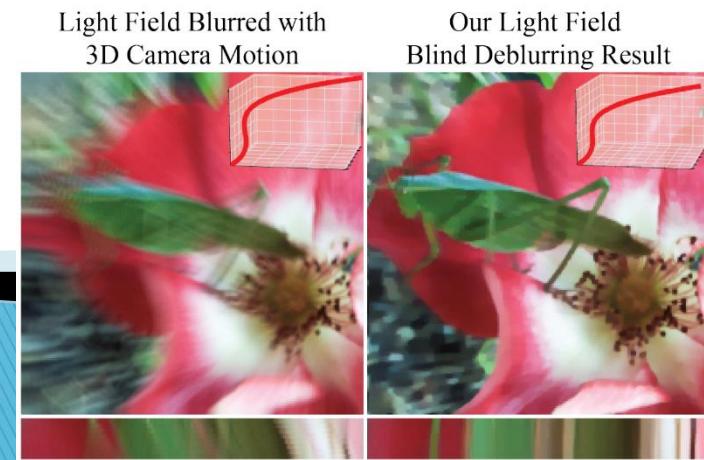
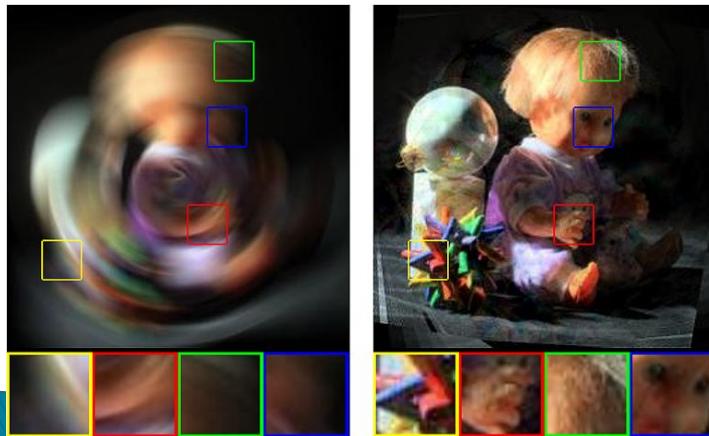


# Restauro de imagem *Deblurring* Filtro de Wiener



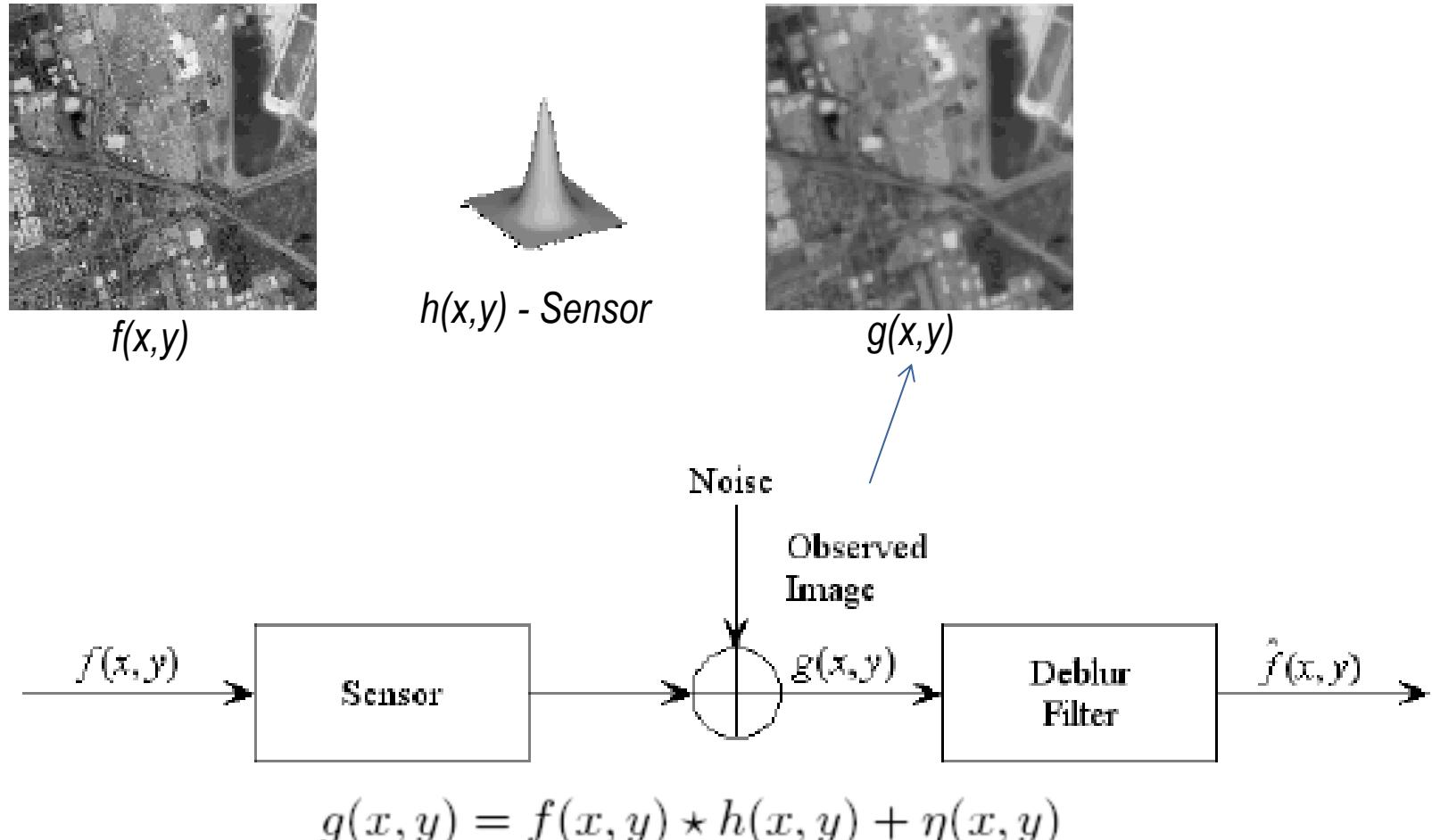
# Restauração / Deblurring de Imagem

- ▶ A desfocagem das imagens deve-se normalmente ao movimento do sensor ou à má focagem da lente.
- ▶ Aplicações de Deblurring podem ser encontradas em diferentes campos desde a astronomia até produtos de consumo
- ▶ É possível restaurar uma imagem desfocada?
- ▶ Como se obtém?

# É possível “refocar” uma imagem?

- ▶ As técnicas de refocagem baseiam-se no conhecimento do modelo da distorção introduzida pela incorreta focagem ou pela “tremideira”.
- ▶ É necessário ter uma descrição matemática de como ocorreu a desfocagem ou ter uma estimativa dessa transformação
- ▶ Como tal, é normalmente necessário conhecer qual o ruído que afetou a imagem ou, pelo menos, uma aproximação a esse ruído.
- ▶ No entanto, esta necessidade nem sempre é realista...

# Modelo tradicional



# No domínio da frequência

$$g(x, y) = f(x, y) * h(x, y) + \eta(x, y) \quad \rightarrow \quad G(u, v) = F(u, v)H(u, v) + N(u, v)$$

- ▶ O objectivo é construir um filtro  $T(u, v)$  tal que possamos obter uma estimativa de  $F(u, v)$

$$\hat{F}(u, v) = G(u, v)T(u, v)$$

- ▶ Temos portanto dois objectivos antagónicos que são retirar ruído (*blur*) e focar a imagem (*sharpen*) pelo que temos de os equilibrar entre si

# Como refocar a imagem?

- ▶ Podemos começar por tentar a transformada inversa do modelo de desfocagem

$$T(u, v) = H^{-1}(u, v)$$

- ▶ O que dará:

$$G(u, v) = F(u, v)H(u, v) + N(u, v)$$

$$\begin{aligned}\hat{F}(u, v) &= G(u, v)H^{-1}(u, v) = \\&= F(u, v)H(u, v)H^{-1}(u, v) + N(u, v)H^{-1}(u, v) = \\&= F(u, v) + N(u, v)H^{-1}(u, v)\end{aligned}$$

- ▶ O principal problema será o termo  $N(u, v)H^{-1}(u, v)$  que terá uma preponderância muito significativa mesmo com quantidades de ruído limitadas dado o ganho nas altas frequências introduzido pelo termo  $H^{-1}$

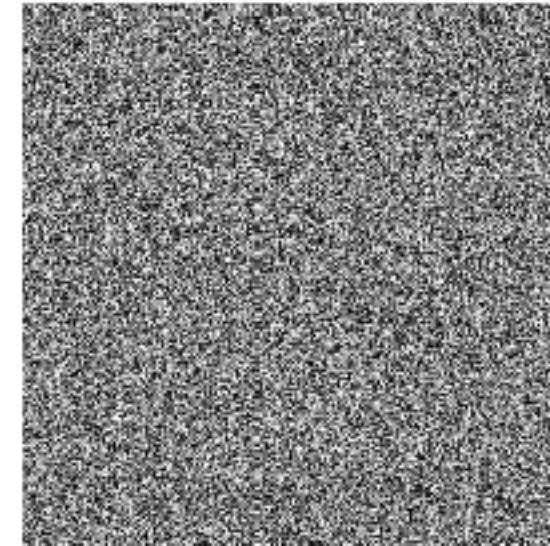
# Exemplo de aplicação de filtragem inversa



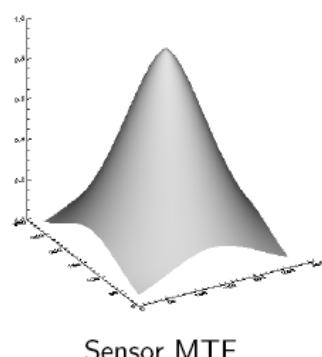
Original Image



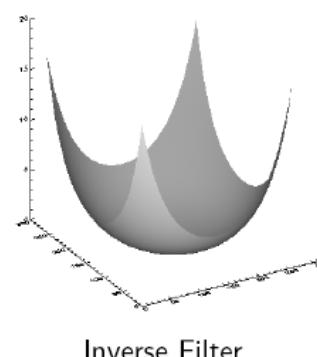
Blurred Image



Restored with  $H^{-1}(u, v)$



Sensor MTF



Inverse Filter

# Atenuando as altas frequências

- Podemos tentar atenuar as altas frequências aplicando uma função de Butterworth à função inversa:

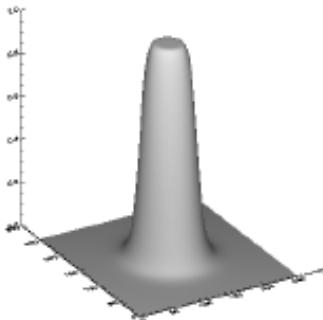


Imagen original filtrada  
pelo Butterworth

$$B(u, v) = \frac{1}{1 + \left( \frac{D(u, v)}{R^2} \right)^{2n}} = \frac{1}{1 + \left( \frac{u^2 + v^2}{R^2} \right)^n}$$

$$\hat{F}(u, v) = (F(u, v)H(u, v) + N(u, v))T(u, v)$$

$$= F(u, v)B(u, v) + \frac{N(u, v)B(u, v)}{H(u, v)}$$

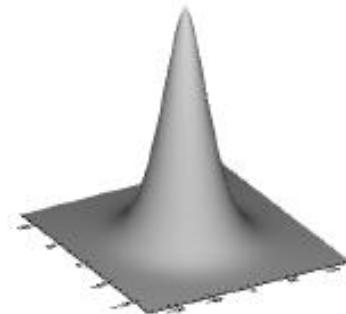
$$T(u, v) = H^{-1}(u, v)$$

$$T(u, v) = \frac{B(u, v)}{H(u, v)}$$

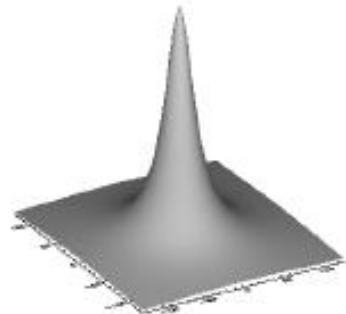
Ruído atenuado pelo  
Butterworth e filtrado pelo  
inverso da desfocagem

- O problema é agora escolher  $B(u, v)$  tal que a imagem seja minimamente recuperada e o ruído devidamente atenuado

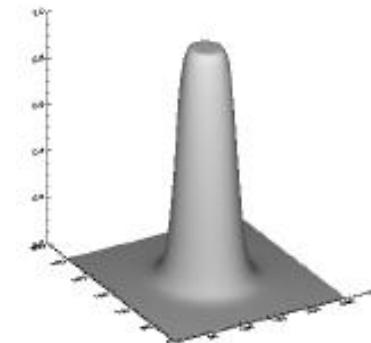
# Exemplos de Butterworths



$H(u, v)$



$B(u, v): R = 20, n = 1$



$R = 40, n = 1$



Blurred Image  $G(u, v)$

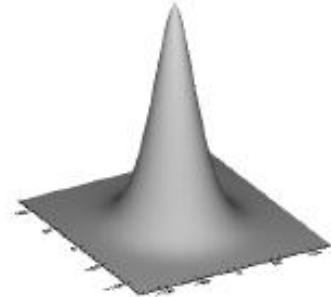


Restored using  $R = 20$

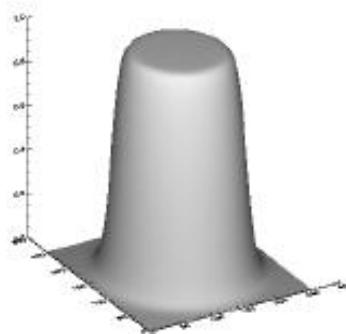


Restored using  $R = 40$

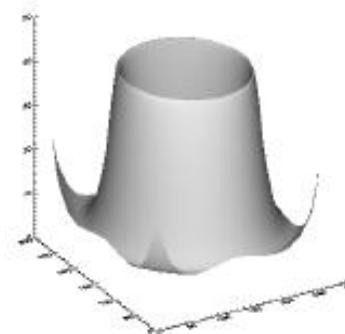
# Exemplos de Butterworths



$H(u, v)$



$B(u, v): R = 90, n = 8$



Inverse  $B/H$



Original Image  $G(u, v)$



Blurred using  $R = 20$



Restored

# Norbert Wiener 1894–1964

- Born in 1894 in Columbia, Missouri
- 1918 – Ph.D. in Philosophy, Harvard, age 18
- Instructor of Mathematics at MIT in 1919, Assistant Professor in 1929 and Professor in 1931
- 1942 – “The Yellow Peril” (classified)
- 1948 – “Cybernetics or Control and Communication in the Animal and the Machine”
- 1949 – “Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications”



Norbert Wiener

# Filtro de Wiener

$$\hat{F}(u, v) = G(u, v)T(u, v)$$

$$G(u, v) = F(u, v)H(u, v) + N(u, v)$$

O filtro de Wiener minimiza o erro:

$$e^2 = E[(f(x, y) - \hat{f}(x, y))^2]$$

No domínio da frequência a sua função é:

$$T(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + S_\eta(u, v)/S_f(u, v)}$$

$H(u, v)$  = the degradation function

$$|H(u, v)|^2 = H^*(u, v)H(u, v)$$

$H^*(u, v)$  = complex conjugate of  $H(u, v)$

$S_\eta(u, v) = |N(u, v)|^2$  = the power spectrum of the noise

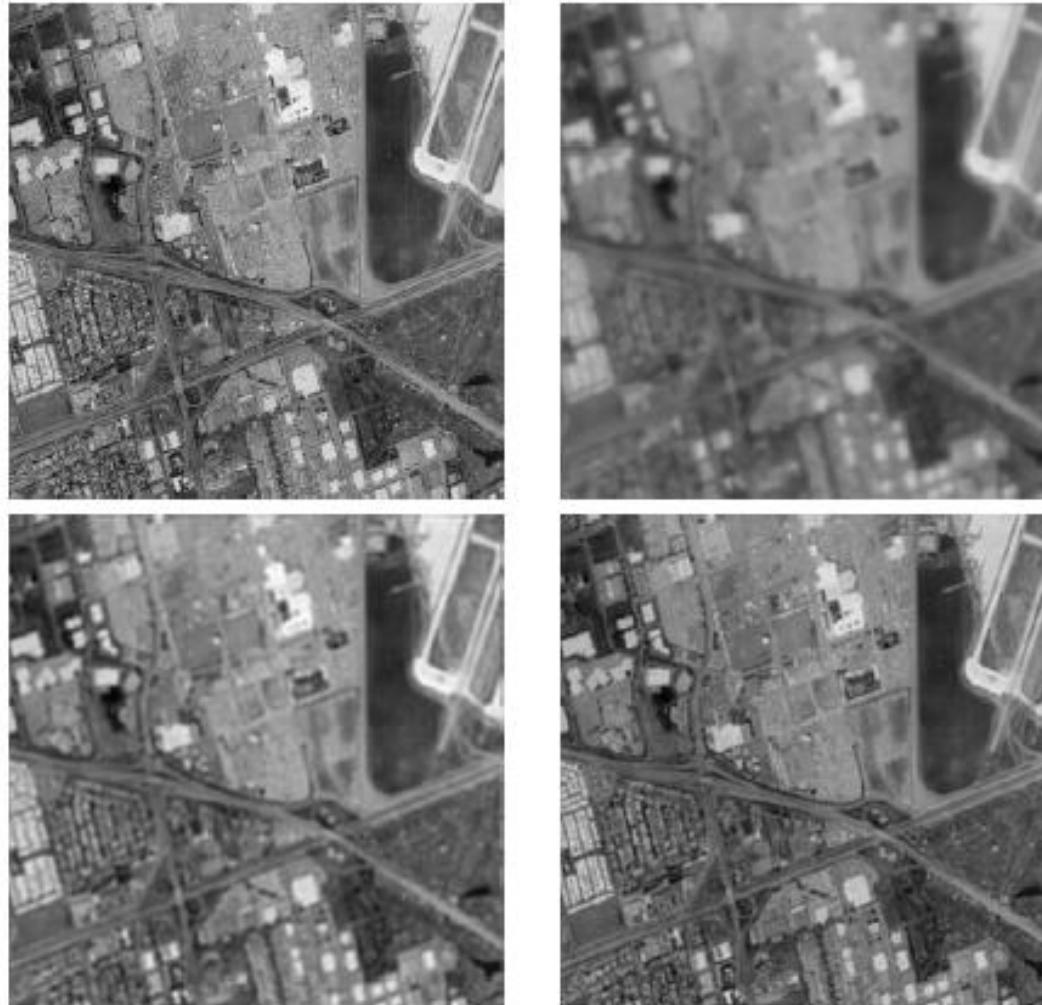
$S_f(u, v) = |F(u, v)|^2$  = power spectrum of the original image

Quando se desconhecem os espectros de potência do sinal e do ruído pode usar-se a aproximação:

$$T(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K} \quad K = \text{constant}$$

# Aplicações do filtro de Winner

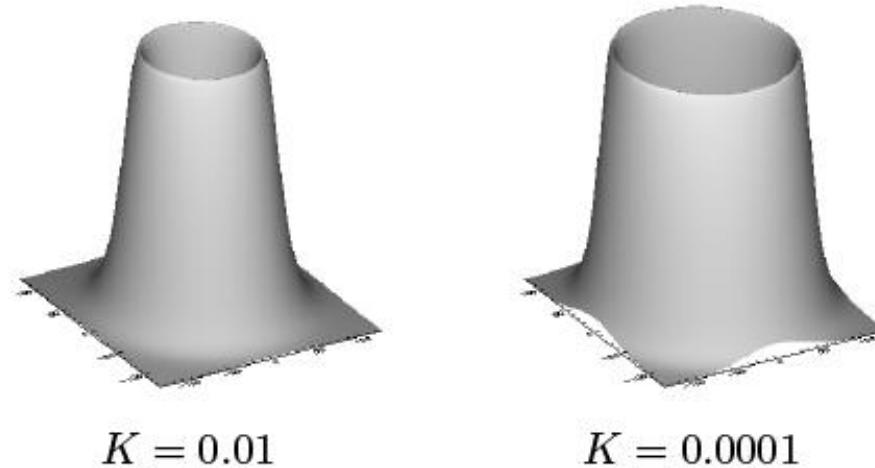
Original image (left),  
blurred image (right)



Restored with Wiener  
filters with  $K = 0.01$   
(left) and  $K = 0.0001$   
(right)

# Resposta do filtro de Wiener

- ▶ A resposta de frequência de um filtro de Wiener com  $K=0.01$  e  $K=0.0001$



- ▶ Como se pode observar não existe um ganho catastrófico em nenhuma frequência

# Comparação Wiener / Butterworth

- ▶ Os resultados são muito idênticos



Original

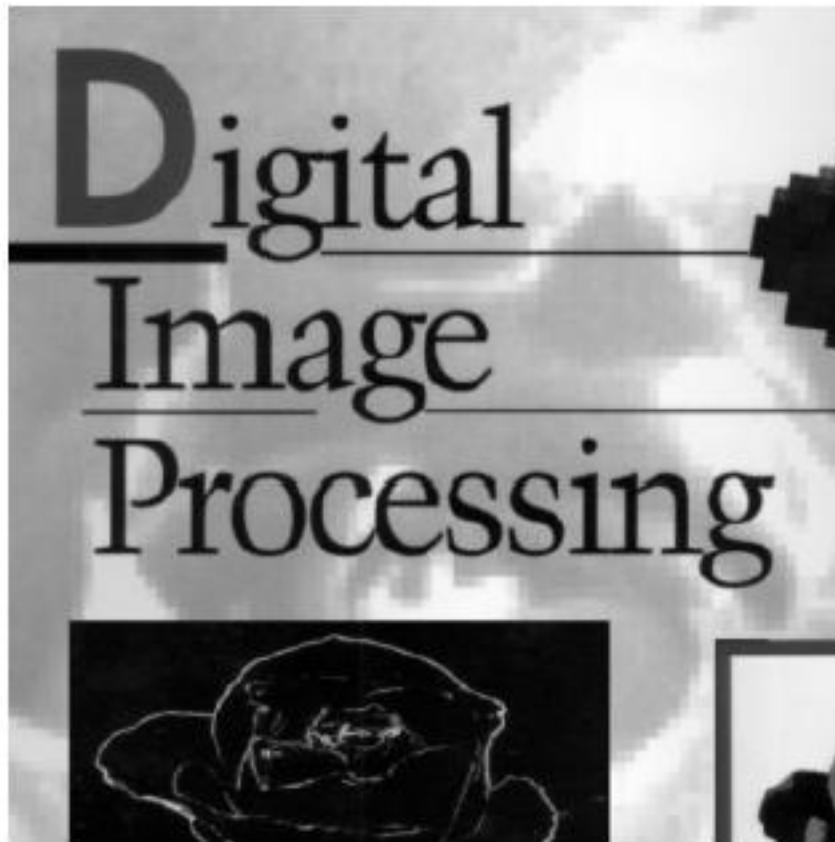


Wiener  $K = 0.0001$



Inverse Butterworth [90, 8]

# Compensação de imagens “tremidas”



# Compensação de imagens “tremidas”

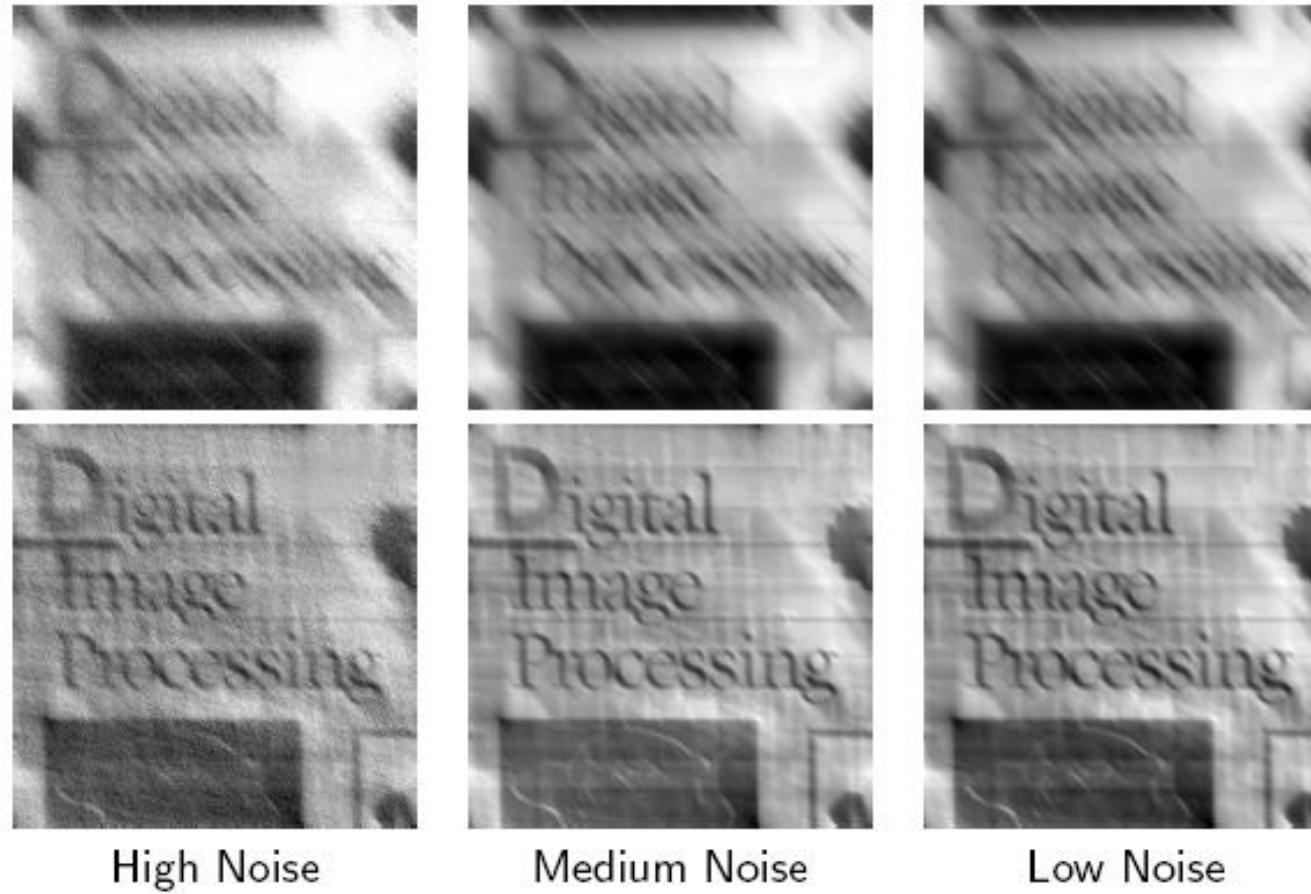
- ▶ Uma imagem “tremida” deve-se ao movimento do sensor durante a captação da imagem.
- ▶ Supondo que a velocidade do sensor é constante e que em T segundos se move de  $f(x,y)$  para  $f(x-\alpha, y-\beta)$  então teremos:

$$g(x, y) = \int_0^T f\left(x - \frac{\alpha t}{T}, y - \frac{\beta t}{T}\right) dt \quad G(u, v) = F(u, v)H(u, v)$$

- ▶ Sendo a função de transferência:

$$H(u, v) = T \frac{\sin \pi(\alpha u + \beta v)}{\pi(\alpha u + \beta v)} e^{-i\pi(\alpha u + \beta v)}$$

# Compensação de imagens “tremidas”



Foi utilizada a expressão simplificada com K encontrado por tentativa e erro de forma a obter o melhor resultado visual

$$T(u,v) = \frac{|H^*(u,v)|}{|H(u,v)|^2 + K}$$

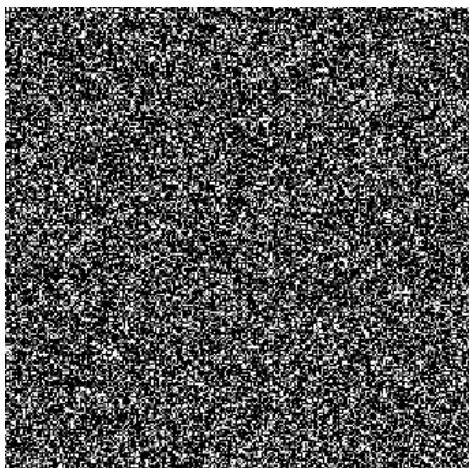
# Filtro de Wiener – exemplo de aplicação



Imagen original



Imagen “desfocada” / tremida



Ruído somado  
à imagem  
(multiplicado  
por 100x)



Imagen recuperada

# Exemplos



# Outros Métodos

## ► Regularização Tikhonov

- $y$  – parâmetro de regularização
- $P(u, v)$  – Fourier-function of Laplace operator (matrix  $3 * 3$ ).

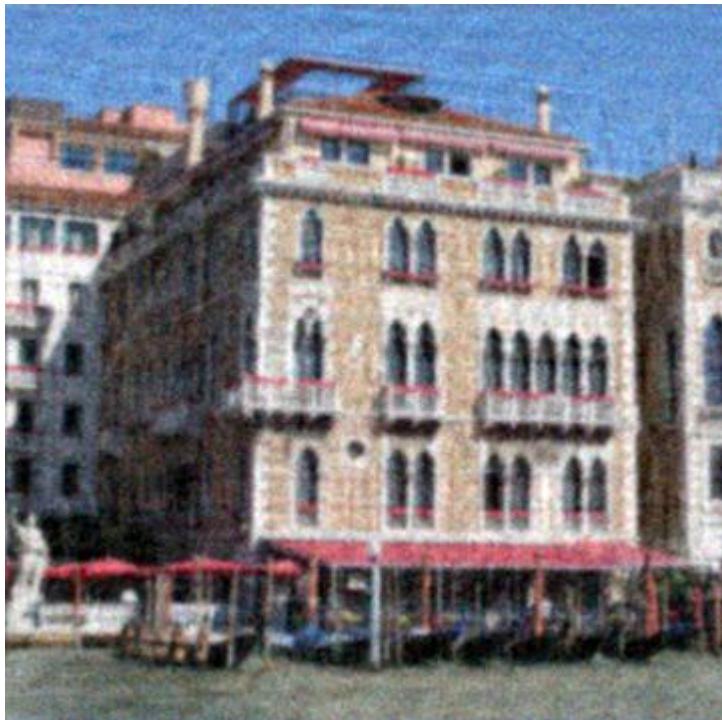
$$\hat{F}(u, v) = \left( \frac{H^*(u, v)}{|H(u, v)|^2 + y|P(u, v)|^2} \right) G(u, v)$$

## ► Filtro Lucy–Richardson

- Sem uso da transformada de Fourier – tudo é feito no domínio espacial
- Método iterativo – requer uma condição de paragem e pode levar várias horas para completar

$$\hat{f}_{k+1}(x, y) = \hat{f}_k(x, y) \left[ h(-x, -y) * \frac{g(x, y)}{h(x, y) * \hat{f}_k(x, y)} \right]$$

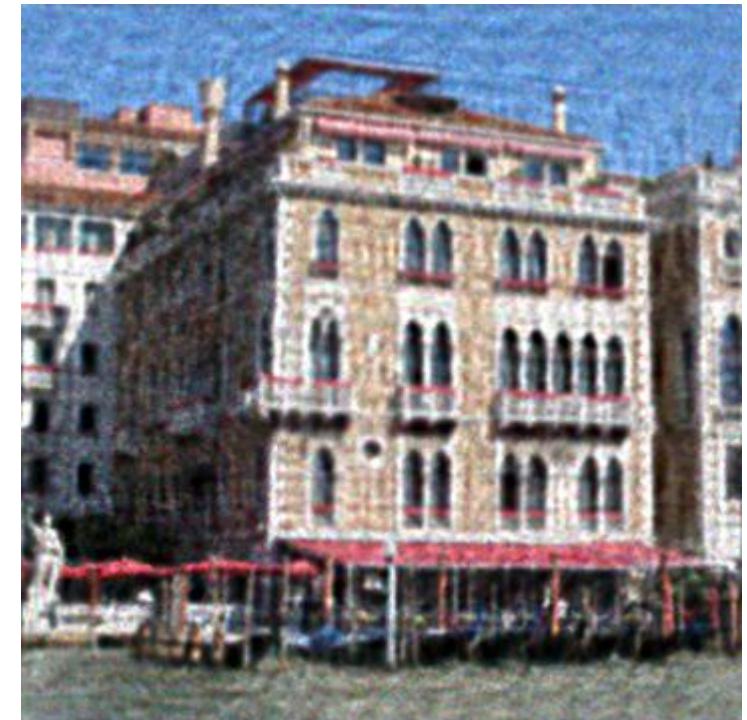
# Exemplo de comparação



Wiener



Tikhonov Regularization



Lucy–Richardson filter

# Fontes

FONTE:

[http://www.cis.rit.edu/class/simg782/lectures/lecture\\_16/lec782\\_05\\_16.pdf](http://www.cis.rit.edu/class/simg782/lectures/lecture_16/lec782_05_16.pdf)