Digital Image Processing

Two Dimensional Signals Processing

Emad Fatemizadeh
Winter 1399

Contents

- Image Processing in Frequency Domain
- Filters
- Power Spectral Density
- Image Restoration
- Inverse Filter
- Wiener Filter
- Adaptive Local Wiener Filter

Fourier Domain Image Processing

- Same as signals processing in frequency domain:

$$g(x,y) = Real\{\mathcal{F}^{-1}(H(u,v)F(u,v))\}$$

- Practical Consideration:
  - Zero Padding (Circular convolution and Linear Convolution)
  - Centering both image and filter in each domain

Fourier Domain Image Processing

- Sequence of operation:

a b c
d e f
g h

FIGURE 4.35
(a) An $M \times N$ image, f.
(b) Padded image, f_p of size $P \times Q$.
(c) Result of multiplying f_p by $(-1)^{x+y}$.
(d) Spectrum of F. (e) Centered Gaussian lowpass filter transfer function, H, of size $P \times Q$.
(f) Spectrum of the product HF.
(g) Image g_p, the real part of the IDFT of HF, multiplied by $(-1)^{x+y}$.
(h) Final result, g, obtained by extracting the first M rows and N columns of g_p.

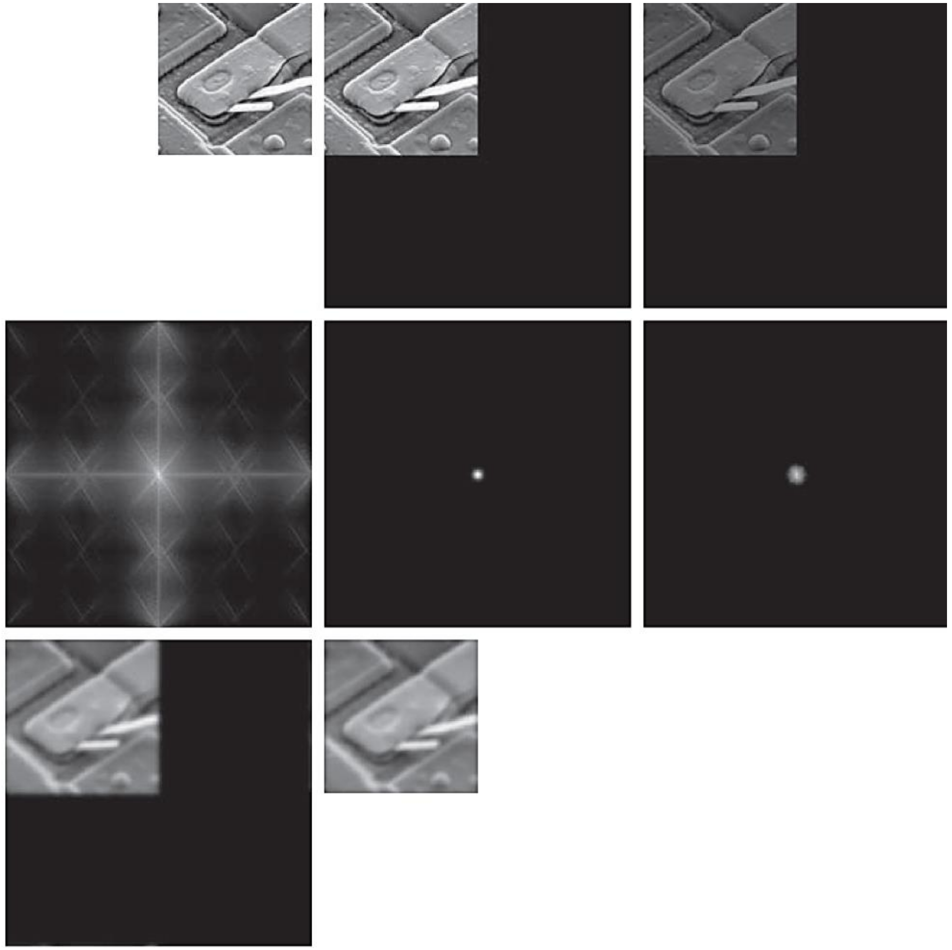
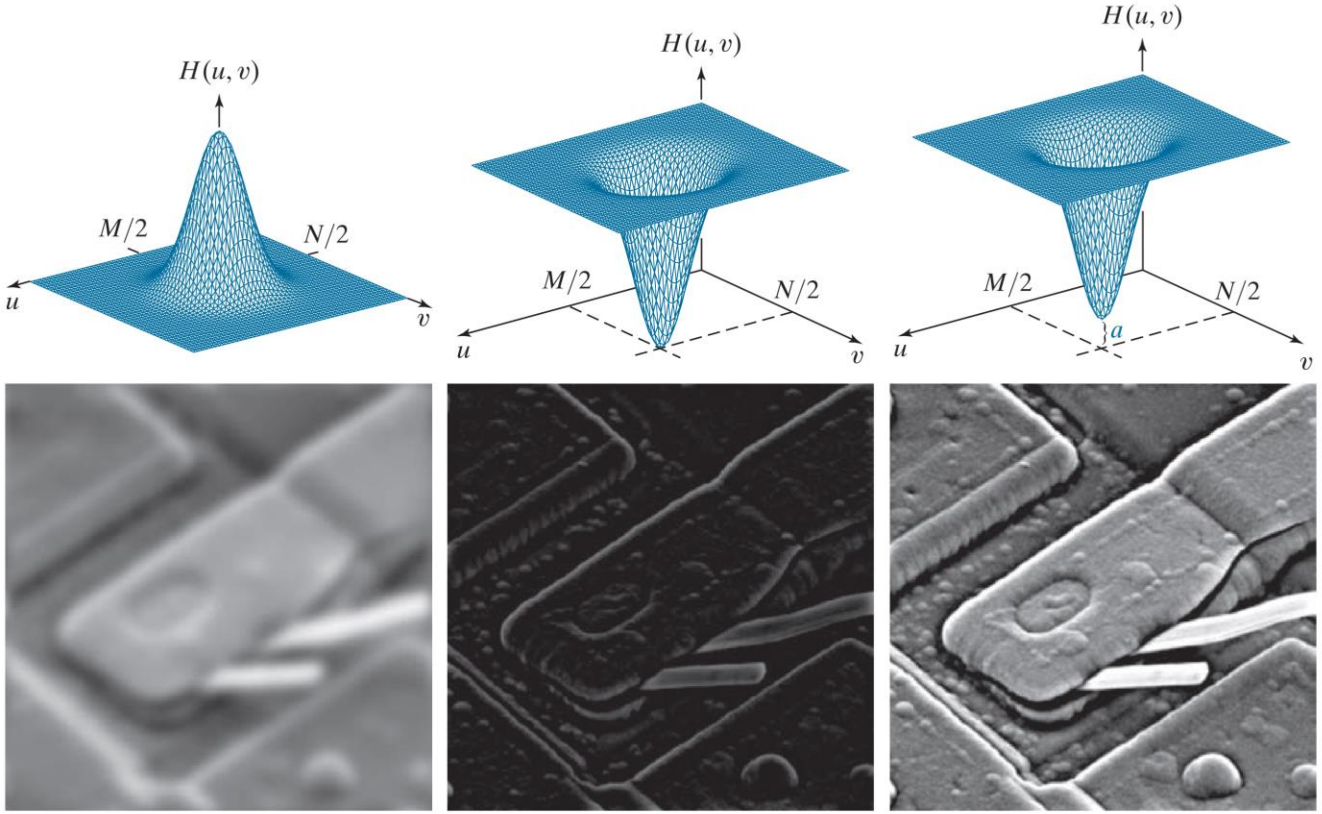


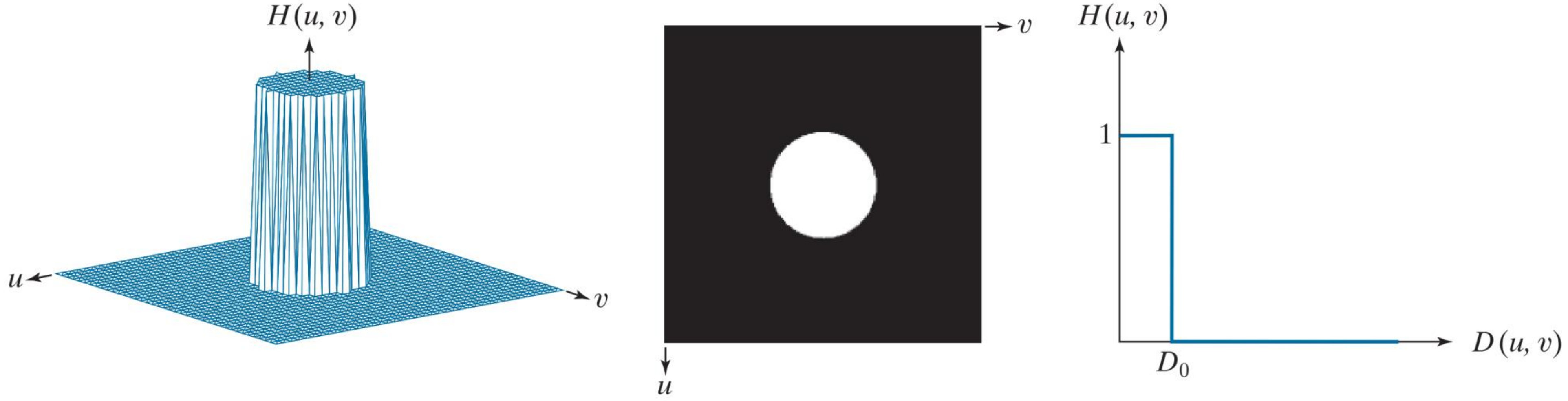
Image Filtering

› Lowpass (left), Highpass (middle), and Highboost (right)



Ideal Lowpass Filter

› Ideal filter is implementable in image domain (why?)

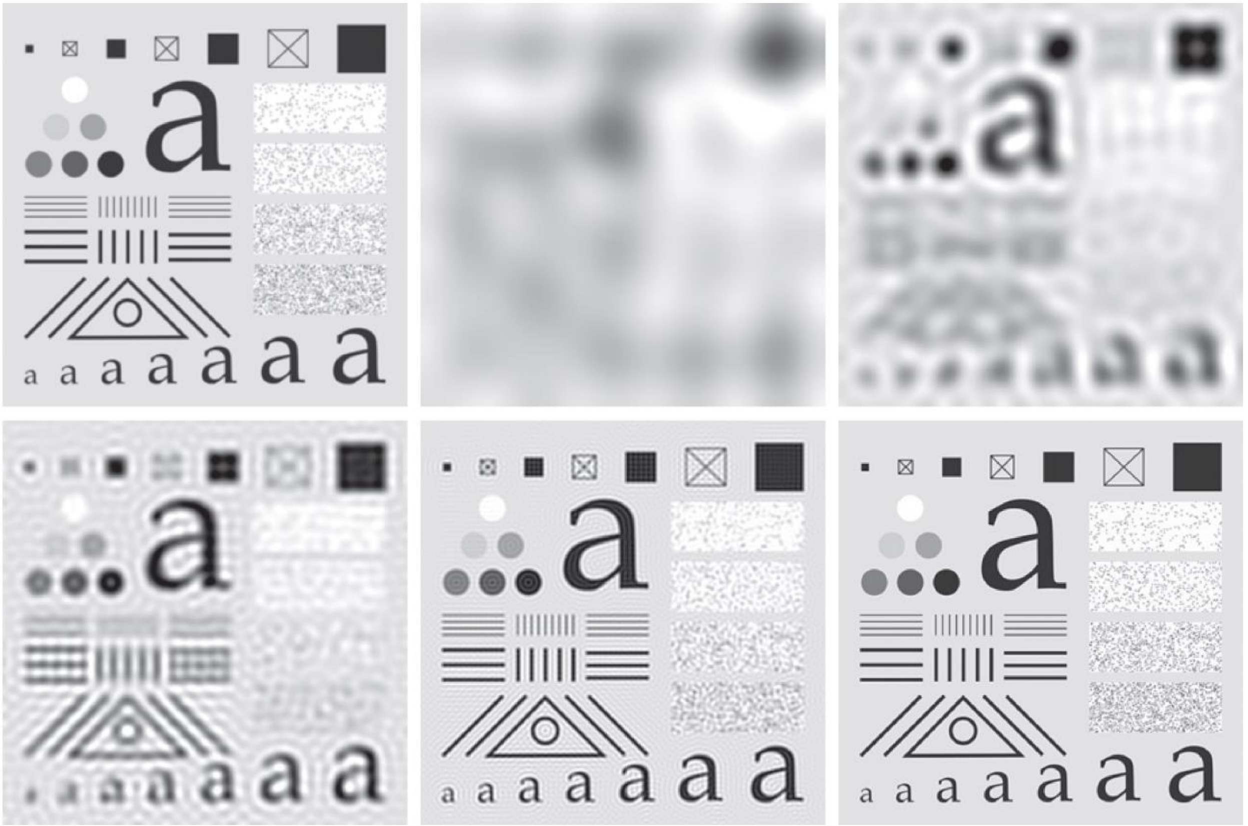


Ideal Lowpass Filter

› Ideal filter in NOT used:

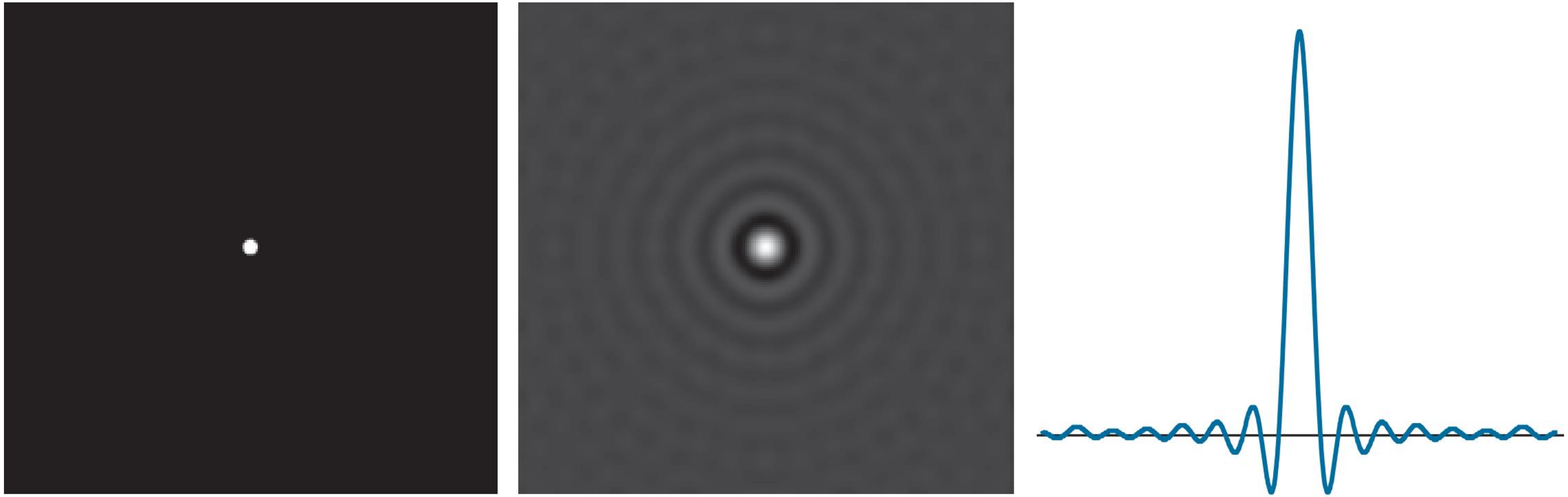
› Blurring Effect

› Ringing Effect



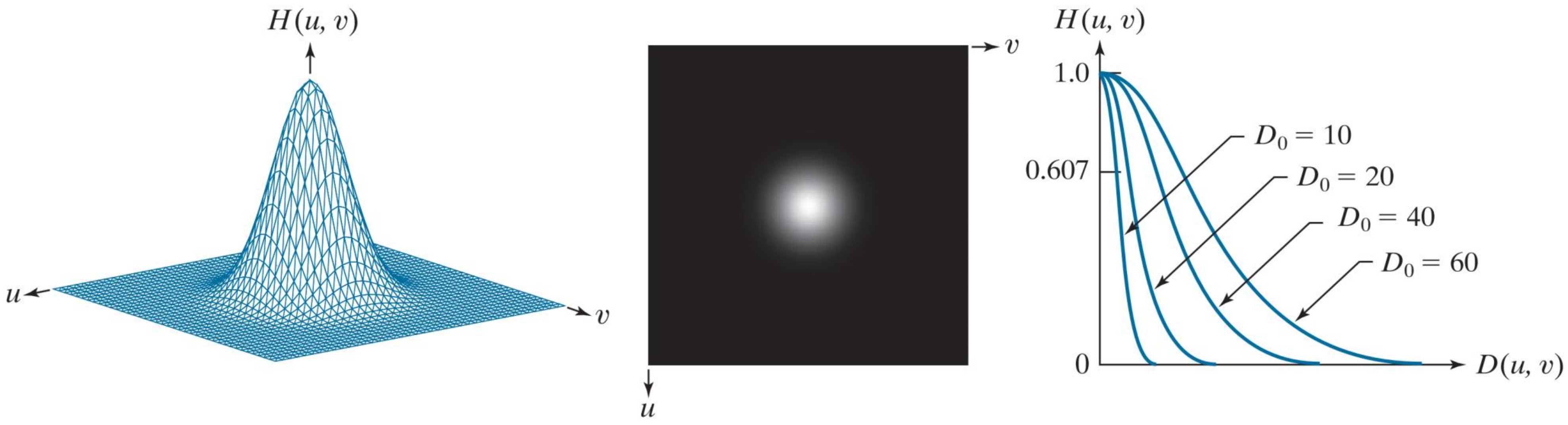
Ideal Lowpass Filter PSD

› PSF of ideal low pass filter:



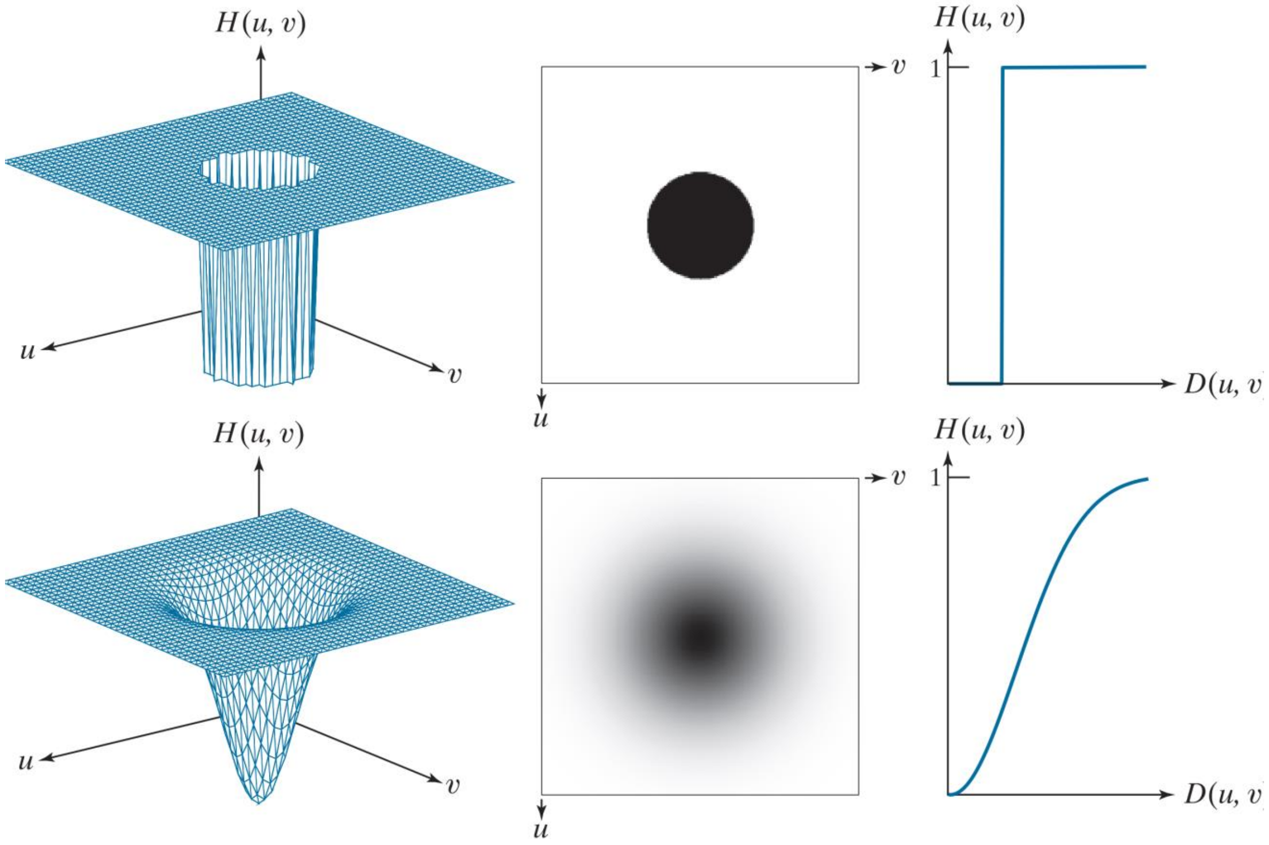
Gaussian Lowpass Filter

- There is no ringing effect but blurring effect:
- $G(u,v) = exp(-D^2(u,v)/2D_0^2),\ D(u,v) = \sqrt{u^2+v^2}$
- $g(x,y) = 2\pi D_0^2 exp\left(-2\pi^2 D_0^2 r^2(x,y)\right),\ r(x,y) = \sqrt{x^2+y^2}$



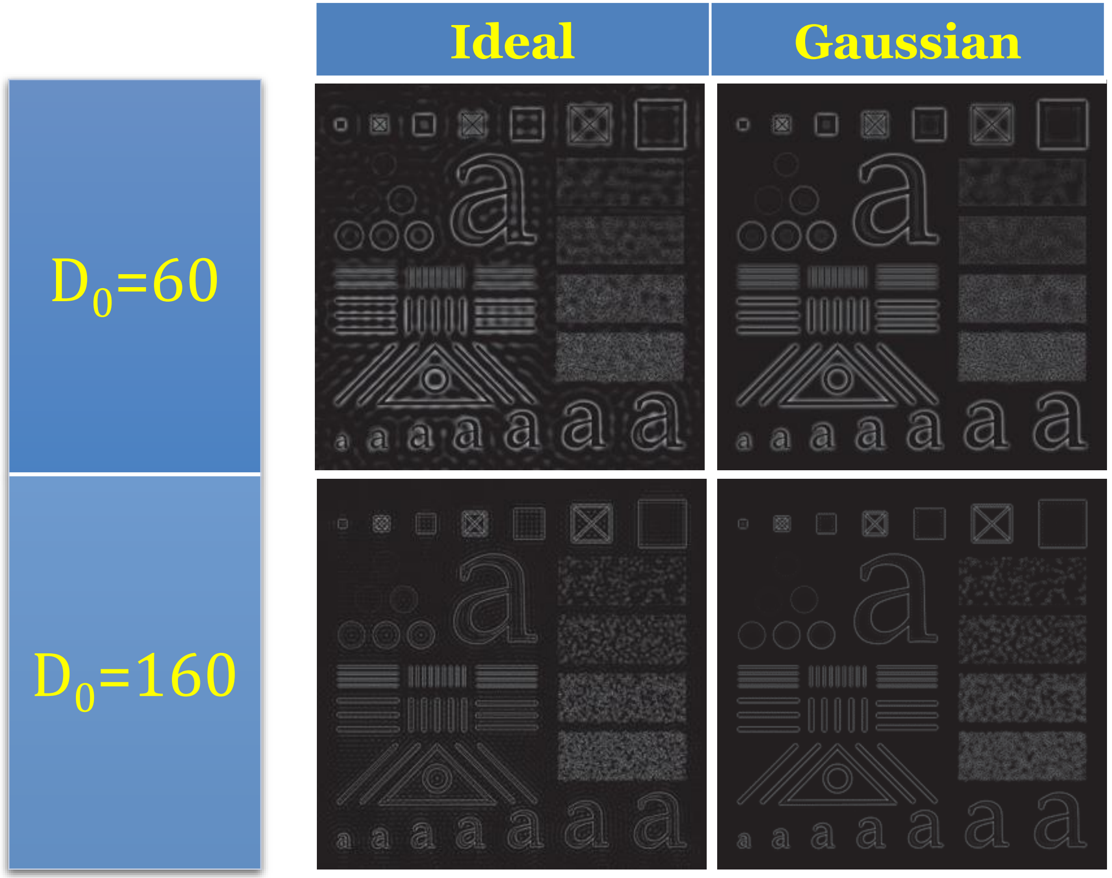
Ideal and Gaussian Highpass Filter

› Highpass filter from lowpass filter: $HPF = 1 - LPF$



Ideal and Gaussian Highpass Filter

› Ideal (left) and Gaussian (right) highpass filter



Laplacian in Frequency Domain

- Mathematical formulation:

$$g = -\nabla^2 f = -\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial Y^2}\right) \Rightarrow G(u,v) = 4\pi^2(u^2+v^2)F(u,v)$$

$$H(u,v) = 4\pi^2(u^2+v^2) = 4\pi^2 D^2(u,v)$$

- Is it equivalent to spatial domain Laplacian?

Power Spectral Density

- PSD of image $f(m,n)$:

$$P_{FF}(f_x, f_y) = \lim_{N\to\infty} \frac{1}{(2N+1)^2} \left|F_N(f_x, f_y)\right|^2$$

$$F_N(f_x, f_y) = \sum_{m=-N}^{N} \sum_{n=-N}^{N} f(m,n)\, e^{-j2\pi(f_x m + f_y n)}$$

- For real world images (limited size) and discrete frequency:

$$f_x = \frac{u}{M}, f_y = \frac{v}{N}$$

$$P_{FF}(u,v) \cong E\{|F(u,v)|^2\} \xrightarrow{\text{single observation}} |F(u,v)|^2$$

Cross Power Spectral Density

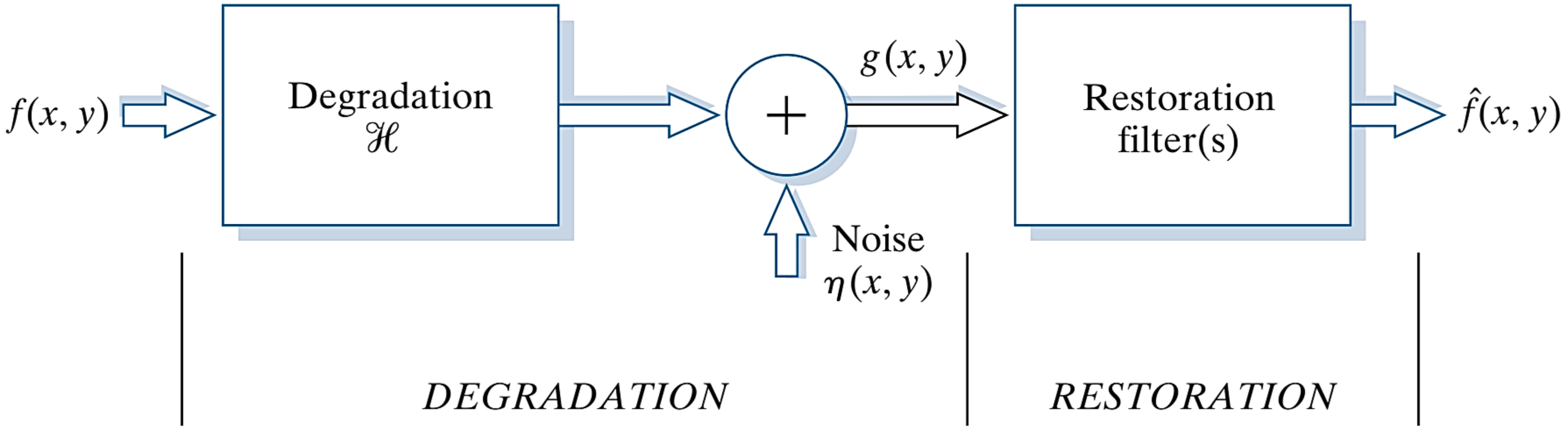
- For two image f and g:

$$P_{FG}(u,v) \cong E\{F(u,v)G^*(u,v)\} \xrightarrow{\text{single observation}} F(u,v)G^*(u,v)$$

Image Restorations

› Definition: Restoration attempts to recover an image that has been degraded by using a priori knowledge of the degradation phenomenon and noise properties.

› Common Degradation Model:



Linear Degradation and Restoration Model (1)

- We assume LSI system of degradation and restoration:

$$g(x,y) = f(x,y) * h(x,y) + \eta(x,y)$$

$$\hat{f}(x,y) = g(x,y) * w(x,y)$$

- $g(x,y)$: Degraded observation
- $f(x,y)$: Original Image
- $h(x,y)$: PSF of degradation function
- $\eta(x,y)$: uncorrelated additive noise

Linear Degradation and Restoration Model (2)

- In frequency domain:

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

- What we know:
  - Noise PSD ($P_{NN}(u,v)$) and/or noise pdf
    - We assume noise is zero mean, known variance, i.i.d, and uncorrelated
  - PSF ($h(x,y)$) of degradation function

Inverse Filtering

- Noise free formulation:

$$W(u,v) = \frac{1}{\hat{H}(u,v)}$$

$$\hat{F}(u,v) = \frac{G(u,v)}{\hat{H}(u,v)} = \frac{F(u,v)H(u,v)}{\hat{H}(u,v)} \cong F(u,v)$$

- Main drawback:
  - Division by zero even we know exact PSD $(\frac{0}{0})$

Inverse Filtering

- In presence of noise:

$$W(u,v) = \frac{1}{\hat{H}(u,v)}$$

$$\hat{F}(u,v) = \frac{G(u,v)}{\hat{H}(u,v)} = \frac{F(u,v)H(u,v) + N(u,v)}{\hat{H}(u,v)} \cong F(u,v) + \frac{N(u,v)}{\hat{H}(u,v)}$$

- Main drawback:
  - Division by zero even we know exact PSD $\left(\frac{0}{0}\right)$
  - Noise amplification!

Pseudo Inverse Filtering

- To overcome division by zero problem:

$$W(u,v) = \begin{cases} \dfrac{1}{\hat{H}(u,v)}, & \left|\hat{H}(u,v)\right| \geq \delta \\ 0, & \left|\hat{H}(u,v)\right| < \delta \end{cases}$$

- Phase corruption!
- Another implementation:

$$W(u,v) = \frac{\hat{H}^*(u,v)}{\left|\hat{H}(u,v)\right|^2 + \varepsilon}$$

Wiener Filtering

- Least means square estimation:

$$W(u,v) = \underset{W}{argmin}\, E\{|F(u,v) - W(u,v)G(u,v)|^2\}$$

$$\therefore W(u,v) = \frac{P_{FG}(u,v)}{P_{GG}(u,v)}$$

Wiener Filtering – Noise Only Corruption

- We have:

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

- $P_{GG} = E\{|F+N|^2\} = E\{|F|^2 + |N|^2 + FN^* + NF^*\} = P_{FF} + P_{NN}$
- $P_{FG} = E\{F(F+N)^*\} = E\{|F|^2 + FN^*\} = P_{FF}$
- Note: for zero means, i.i.d, and uncorrelated noise:

$$E\{NF^*\} = E\{N\}E\{F^*\} = 0, \qquad E\{FN^*\} = E\{F\}E\{N^*\} = 0$$

- Thus:

$$W(u,v) = \frac{P_{FG}(u,v)}{P_{GG}(u,v)} = \frac{P_{FF}}{P_{FF} + P_{NN}} = \frac{P_{GG} - P_{NN}}{P_{GG}}$$

Wiener Filtering – General Corruption

› We have:

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

› $P_{GG} = E\{|FH + N|^2\} = E\{|H|^2|F|^2 + |N|^2 + FHN^* + NF^*N^*\} = |H|^2 P_{FF} + P_{NN}$

› $P_{FG} = E\{F(FH + N)^*\} = E\{H^*|F|^2 + FN^*\} = H^* P_{FF}$

› Thus:

$$W(u,v) = \frac{P_{FG}(u,v)}{P_{GG}(u,v)} = \frac{H^* P_{FF}}{|H|^2 P_{FF} + P_{NN}} = \frac{1}{H} \frac{|H|^2 P_{FF}}{|H|^2 P_{FF} + P_{NN}}$$

Phase in Wiener Filter

- Wiener recap:

$$W(u,v)=\frac{1}{H}\frac{|H|^2P_{FF}}{|H|^2P_{FF}+P_{NN}}\Rightarrow \measuredangle W(u,v)=-\measuredangle H(u,v)$$

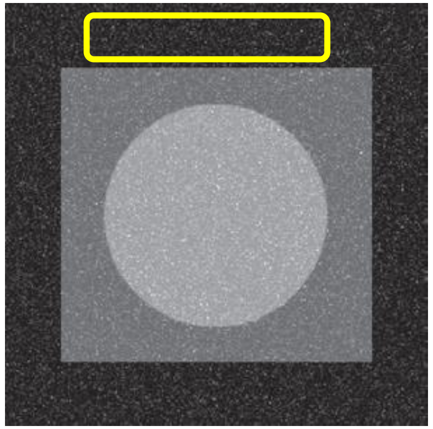
- There is no need to phase compensation

Wiener Filtering – Practical Issues (1)

- Noise PSD (P_{NN})
  - For zero mean, white noise:

$$P_{NN} = \sigma^2$$

  - Where σ^2 is noise variance
- Noise variance estimation:
  - A Naïve approach is sample variance in flat region



Wiener Filtering – Practical Issues (2)

- Original image PSD (P_{FF}) :
  - Noise Only corruption:

$$W(u,v) = \frac{P_{FG}(u,v)}{P_{GG}(u,v)} = \frac{P_{FF}}{P_{FF} + P_{NN}} = \frac{P_{GG} - P_{NN}}{P_{GG}}$$

  - Noise and degradation corruption:

$$W(u,v) = \frac{P_{FG}(u,v)}{P_{GG}(u,v)} = \frac{H^* P_{FF}}{|H|^2 P_{FF} + P_{NN}} = \frac{1}{H} \frac{|H|^2 P_{FF}}{|H|^2 P_{FF} + P_{NN}} = \frac{1}{H} \frac{P_{GG} - P_{NN}}{P_{GG}}$$

Wiener Filtering – Practical Issues (3)

› Average preservation:

$$W(u,v)\Big|_{(0,0)} \neq 1$$

› There is change in average of image brightness!

› Solution:

$$\left(g(x,y) - \bar{g}(x,y)\right) * w(x,y) + \bar{g}(x,y)$$

Iterative Wiener Filter

› Noise only formulation:

1. $k \leftarrow 0$
2. $P_{FF}^{(k)} = P_{GG}$
3. $W^{(k+1)} = \frac{P_{FF}^{(k)}}{P_{FF}^{(k)} + P_{NN}}$
4. $F^{(k+1)} = W^{(k+1)} G$
5. $P_{FF}^{(k+1)} = \left| F^{(k+1)} \right|^2$
6. $k \leftarrow k + 1$
7. Repeat steps 3-4-5-6 until convergence

Adaptive Local Wiener Filter (1)

- Image are non-stationary!
- Need adaptive Wiener filter which is locally optimal.
- Assume image within small region are stationary
- Clean image model within a small window:

$$f(x,y) = \mu_f(x,y) + \sigma_f^2(x,y)\eta(x,y)$$

- Where:
  - $\eta(x,y)$: zero mean unity variance white noise
  - $\mu_f(x,y)$ and $\sigma_f^2(x,y)$ are constant within a small window centered at (x,y)

Adaptive Local Wiener Filter (2)

- Noise image model:

$$g(x,y) = f(x,y) + v(x,y)$$

- where:
  - $f(x,y)$: clean image
  - $v(x,y)$: Additive noise (zero mean and known variance)
- Adaptive Local Wiener formulation:

$$\hat{f}(x,y) = \left(g(x,y) - \mu_g(x,y)\right) * w(x,y) + \mu_g(x,y)$$

$$W(u,v) = \frac{P_{ff}}{P_{ff} + P_{vv}} = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_v^2} \Rightarrow w(x,y) = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_v^2}\delta(x,y)$$

Adaptive Local Wiener Filter (3)

- Local Wiener formulation:

$$\hat{f}(x,y) = \left(g(x,y) - \mu_g(x,y)\right) * w(x,y) + \mu_g(x,y)$$

$$\hat{f}(x,y) = \left(g(x,y) - \mu_g(x,y)\right)\frac{\sigma_f^2(x,y)}{\sigma_f^2(x,y) + \sigma_v^2} + \mu_g(x,y)$$

- $\mu_f(x,y) = \mu_g(x,y)$: $v(x,y)$ is assumed to be zero mean
- $\mu_g(x,y)$: Local noisy image average
- $\sigma_g^2(x,y)$: Local noisy image variance:

$$\sigma_g^2(x,y) = \sigma_f^2(x,y) + \sigma_v^2$$

Adaptive Local Wiener Filter (4)

- Final formulation:

$$\hat{f}(x,y) = \left(g(x,y) - \mu_g(x,y)\right)\frac{\sigma_f^2(x,y)}{\sigma_f^2(x,y) + \sigma_v^2} + \mu_g(x,y)$$

$$\hat{f}(x,y) = \left(g(x,y) - \mu_g(x,y)\right)\frac{\sigma_g^2(x,y) - \sigma_v^2}{\sigma_g^2(x,y)} + \mu_g(x,y)$$

- Practical consideration:

$$\hat{f}(x,y) = \left(g(x,y) - \mu_g(x,y)\right)\frac{max\left(\sigma_g^2(x,y) - \sigma_v^2, 0\right)}{\sigma_g^2(x,y)} + \mu_g(x,y)$$

Wiener Filtering vs Inverse Filter

- Wiener recap:

$$W(u,v) = \frac{H^* P_{FF}}{|H|^2 P_{FF} + P_{NN}} = \frac{1}{H}\frac{P_{GG} - P_{NN}}{P_{GG}} = \begin{cases} \dfrac{1}{H}\dfrac{P_{GG} - P_{NN}}{P_{GG}}, & |H| \neq 0 \\ 0, & |H| = 0 \end{cases}$$

- For noise free case: $P_{NN} \longrightarrow 0$

$$W(u,v) = \frac{H^* P_{FF}}{|H|^2 P_{FF} + \underbrace{P_{NN}}_{P_{NN} \longrightarrow 0}} = \begin{cases} \dfrac{1}{H}, & |H| \neq 0 \\ 0, & |H| = 0 \end{cases}$$

Matlab Command

- fft2, ifft2, fftshift, ifftshift
- fspecial (average, disk , gaussian, laplacian, log, motion, prewitt, sobel, unsharp)
- deconvblind: Restore image using blind deconvolution
- deconvlucy: Restore image using accelerated Richardson-Lucy algorithm
- deconvreg: Restore image using Regularized filter
- deconvwnr: Restore image using Wiener filter
- wiener2: Perform 2-D adaptive noise-removal filtering

The End

› AnY QuEsTiOn?