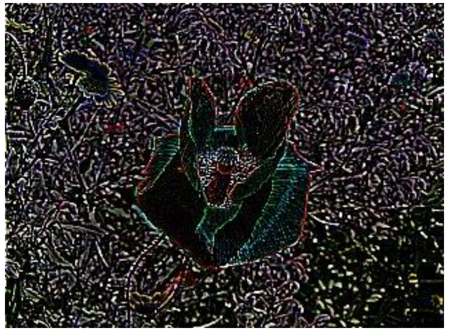
ELEC 474 – Machine Vision

IMAGE FILTERS

M.T. Ahmed, M. Greenspan

Contents

Introduction

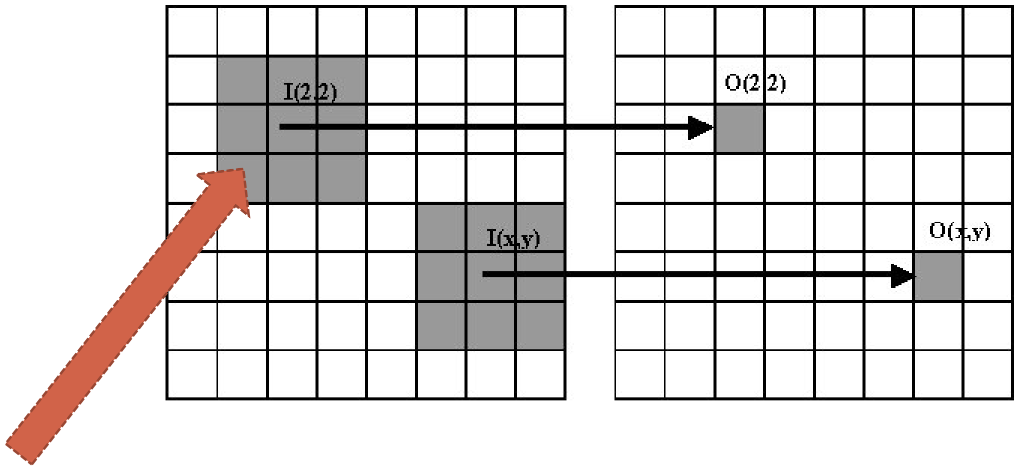




Contents

Spatial Domain Filtering

- Most widely used technique
- Position of the pixel is important
  - Neighborhood Operations around the position.
- Main Idea
  - Subimage/window/template/ structuring element/ kernel/mask
  - Convolution
    - Spatial Domain Filtering is to convolve a mask with the image



$$f(x) * g(x) = \int_{-\infty}^{\infty} f(u)g(x-u)du$$

Convolution

- A mask is a small subimage
  - Preferred odd size, square matrix
  - Well defined centre, symmetric
  - Isotropic, so that they don't favour any specific direction. Invariant to 90 degree rotations
- Steps

1. Translate the mask to every image location
2. At each translation, multiply the underlying image pixels with the overlying mask pixels, and sum their values
3. Replace the image pixel located at the mask center with this summed value

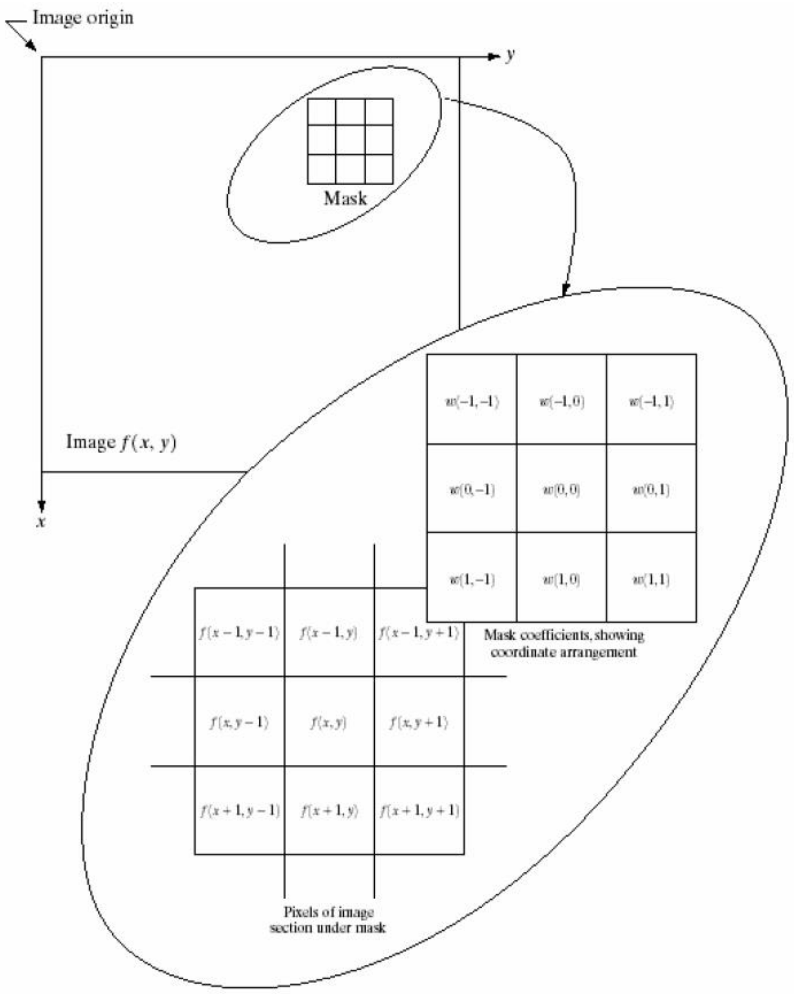


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

Smoothing Filters

- Reduce or remove sharp transitions
  - e.g. noise, unwanted fine detail
- Preprocessing step to many image processing algorithms

Smoothing Filters

- Input image $f(x,y)$, mask $w(s,t)$, output image $g(x,y)$:

$$g(x,y) = \frac{1}{nm}\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

- The image is of width M and height N, and the mask is of width m and height n
- The summation is limited to the region of overlap between the mask and the image
  - The output image $g(x,y)$ is smaller than $f(x,y)$ because:

$$x \in [a, M-a], y \in [b, N-b]$$

Average Filters

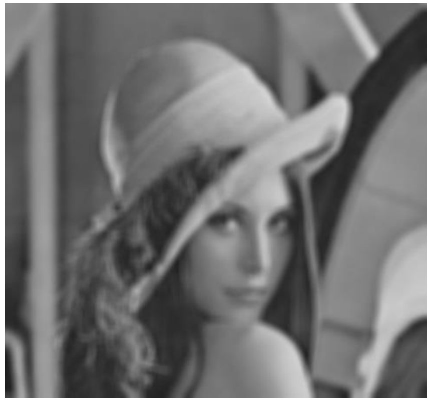
Original

3x3

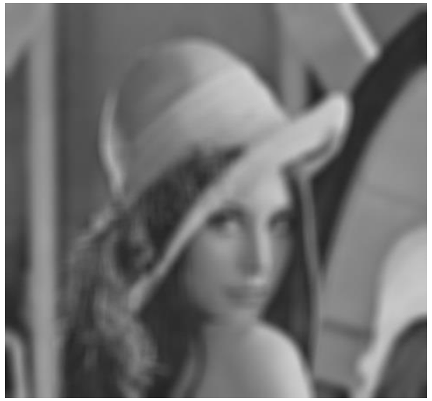
5x5

7x7

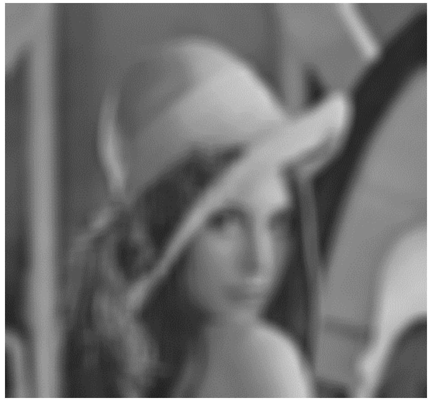
9x9



11x11



15x15



19x19

Order Statistics Filters

- The spatial averaging filter is linear
- The *median filter* is a very useful non-linear spatial smoothing filter
  - more generally known as an *order statistics filter*
  - sometimes called *a rank order filter*

Order Statistics Filters

- rank order filter algorithm:
  1. Produce an ordinal ranking (i.e. a list, ordered by increasing value) of the image pixels that lie under the mask
  2. Replace the image pixel that falls at the center of the mask with:
     - the middle (median) value (in the case of the median filter)

       or/

     - the p^{th} percentile value (in the case of a general order statistics filter)
- Median filter = 50^{th} percentile order statistics filter

Average vs. Median Filter

- Average Filter 3x3

Average vs. Median Filter

- Median Filter 3x3

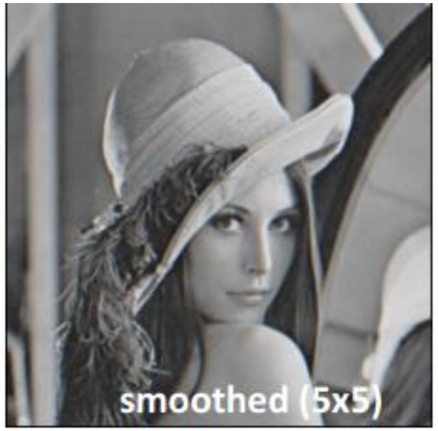
Average vs. Median Filter

- Which filter is better ?

- Median filter is superior at removing salt and pepper noise, but is more computationally expensive
  - This is because the pixels underlying the mask have to be sorted at each mask location, which is expensive
- An average filter is $O(NM)$, where N is the size of the image and M is the size of the mask
- Q/ What is Big Oh of median filter?

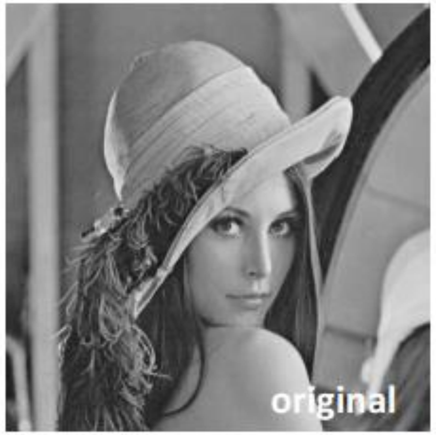
Sharpening Filter

−



=

Let’s add it back:



$+ \alpha$

=

Sharpening Spatial Filters

- Enhances details, such as high contrast areas
- Whereas smoothing filters were a form of integration, sharpening filters are a form of differentiation
- Univariate differentiation in the discrete domain is approximated as:

$$\frac{df}{dx} = f(x+1) - f(x)$$

- Similarly, multivariate partial differentiation is represented as:

$$\begin{cases} \dfrac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y) \\ \dfrac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y) \end{cases}$$

Sharpening Spatial Filters

- Approximate differentiation
  1. The quantity $\partial f(x, y)/\partial x$ is implemented as:

-1	0	1
-2	0	2
-1	0	1

  2. Similarly, $\partial f(x, y)/\partial y$ is implemented as:

-1	-2	-1
0	0	0
1	2	1

  - Sobel Operator
- Rectangular Masks
  - Thinner output

-1	-2	-1
1	2	1

-1	1
-2	2
-1	1

Image Gradient

- The gradient of an image *f*(*x*,*y*) is:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f(x,y)/\partial x \\ \partial f(x,y)/\partial y \end{bmatrix}$$

- The two components of ∇f are combined into single image called the *gradient magnitude:*

$$\text{mag}\,(\nabla f) = \sqrt{G_x^2 + G_y^2}$$

which is sometimes approximated as:

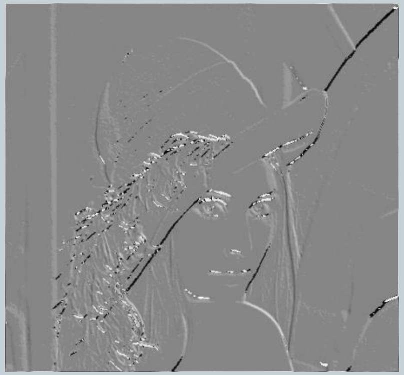
$$\text{mag}\,(\nabla f) \approx |G_x| + |G_y|$$

- The direction of the gradient can be computed as:

$$\theta(x,y) = \text{atan}(G_y(x,y)/G_x(x,y))$$

  - This quantity is useful in edge detection (as we will see)

$\sqrt{G_x^2 + G_y^2}$



$|G_x| + |G_y|$

G_x

G_y

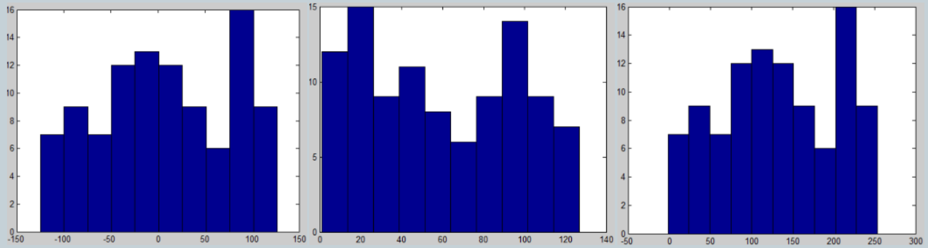
Image Gradient

- Negative Values
- Absolute image
- Scaling
  - A better alternative is to scale the results according to: $u = ax + b$
  - where $b = |\min x_i|$, and:

$$a = \frac{255}{\max x_i + |\min x_i|}$$

  - The maximum dynamic range is achieved, and that the ordinal relationship of the output pixels is maintained

$$\underset{f}{\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}} * \underset{g}{\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



$G_x = |x|$

$G_x = ax + b$

M.T. Ahmed, M. Greenspan

Laplacian

- The Laplacian is the second derivative of an image:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial x^2}$$

- Extending the previous expression for discrete differentiation:

$$\frac{df}{dx} = f(x+1) - f(x)$$

$$\frac{d^2 f}{dx} = f(x+2) - f(x+1) - f(x+1) + f(x)$$

- In practise, this expression is typically shifted, so that it is centered on x:

$$\frac{d^2 f}{dx} = f(x+1) - 2f(x) + f(x-1)$$

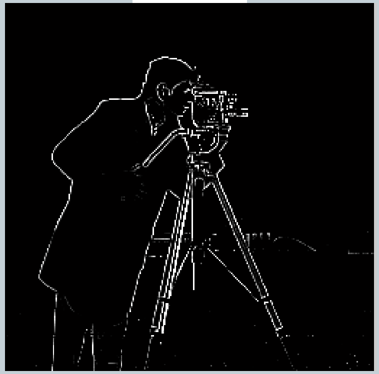
Laplacian

- The summation of these expressions into the Laplacian is represented by the mask
- As with the gradient, there are different variations of the Laplacian. Sometimes the signs are reversed:
- and sometimes the center and corner values are weighted differently:

0	1	0
1	-4	1
0	1	0

0	-1	0
-1	4	-1
0	-1	0

1	1	1
1	-8	1
1	1	1

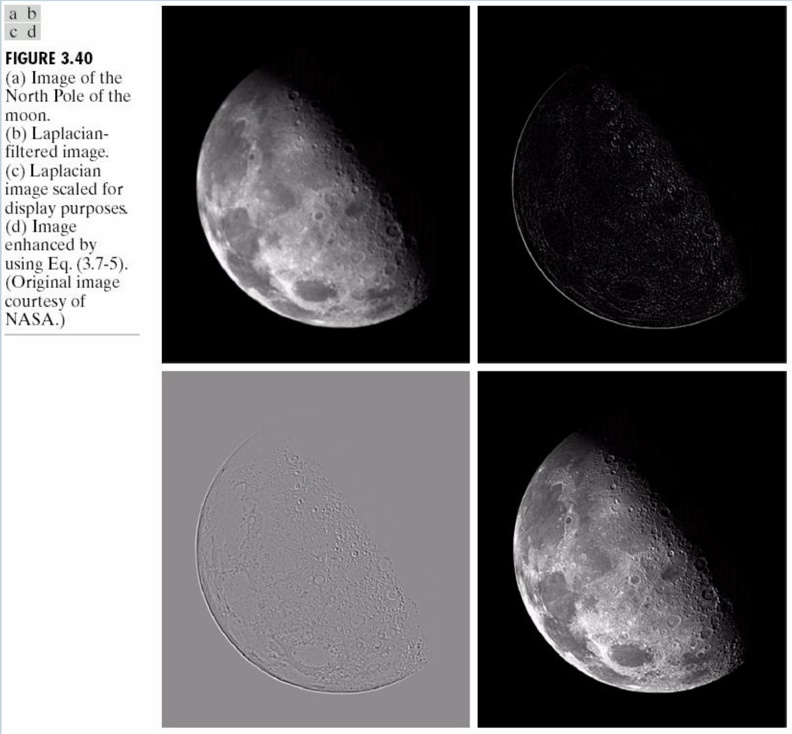


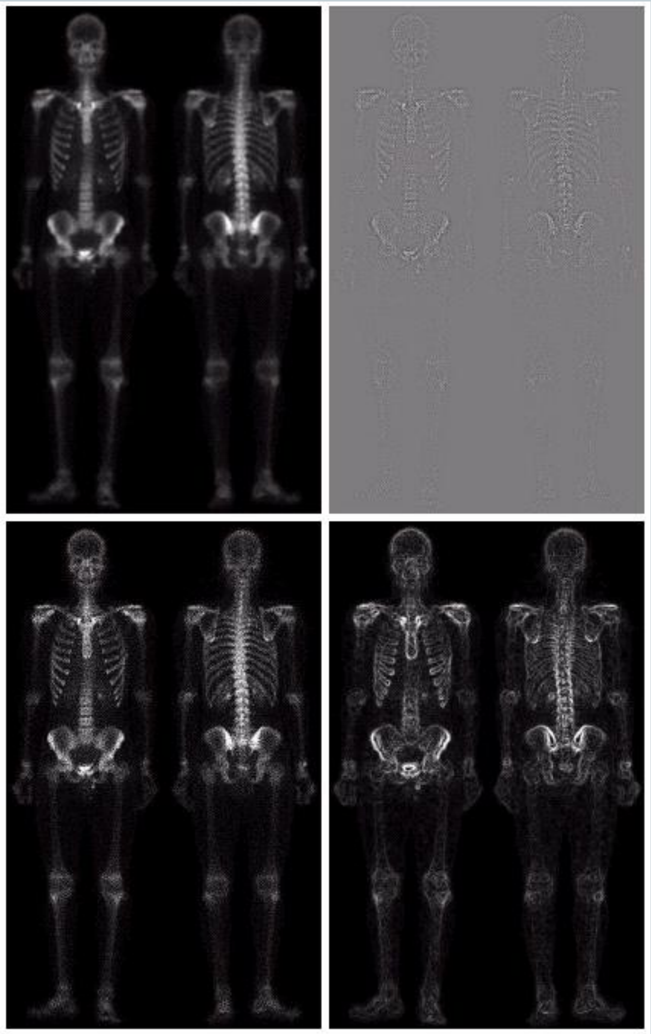
1	1	1
1	-12	1
1	1	1

Image Enhancement using Laplacian

- Add the Laplacian of an image back to the original image:

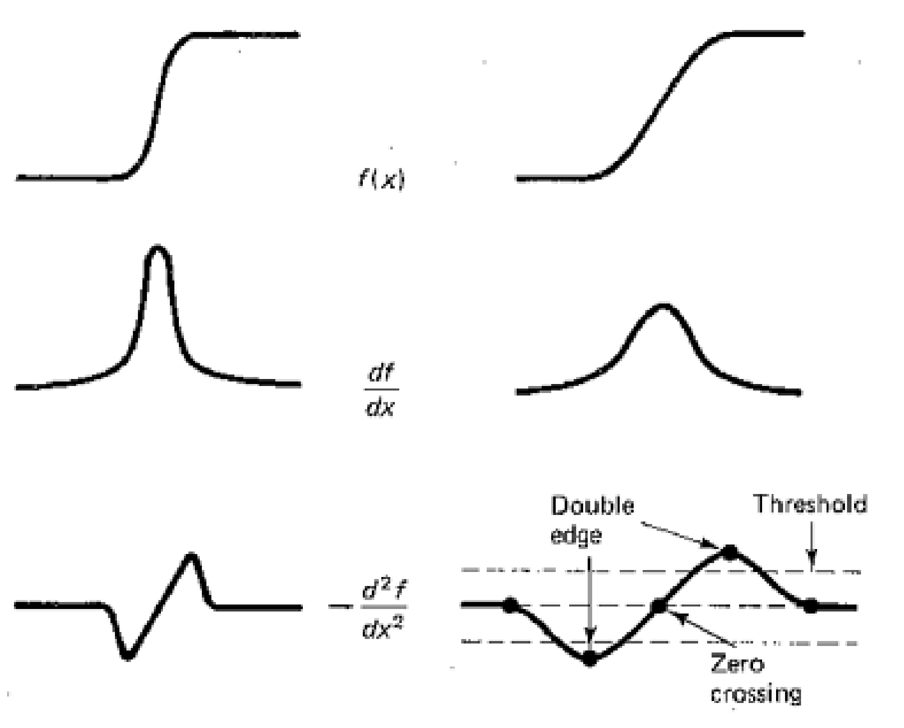
$$g(x,y) = f(x,y) + \nabla^2 f(x,y)$$





Comparison of Gradient and Laplacian

- Laplacian (second derivative) is more sensitive to changes
- Laplacian exhibits a “double edge” effect
- Gradient provides edge direction information, but Laplacian does not



Contents

Frequency Domain Image Processing

- Applications in wide range of image processing
  - Image Analysis
  - Image Filtering
  - Image Reconstruction
  - Image Compression
- Images are discrete
  - Discrete Fourier Transform
  - Discrete Cosine Transform
    - JPEG image compression
  - Discrete Wavelet Transform
    - JPEG 2000

Frequency Domain Filters

- Duality of Spatial and Frequency Domain Filters:

  Every convolution operation in the spatial domain has an equivalent multiplication operation in the frequency domain, and vice versa.

$$h(x,y) * f(x,y) \Leftrightarrow H(u,v)F(u,v)$$

Duality of Spatial and Frequency Domain Filters

A convolution operation in the spatial domain has an equivalent multiplication operation in the frequency domain, and vice versa:

$$\mathcal{F}[f(t)] = F(w) \quad \text{and} \quad \mathcal{F}[g(t)] = G(w)$$

$$\Longrightarrow \begin{cases} \mathcal{F}[f(t) * g(t)] = F(w)G(w) \\ \\ \mathcal{F}^{-1}[F(w) * G(w)] = f(t)g(t) \end{cases}$$

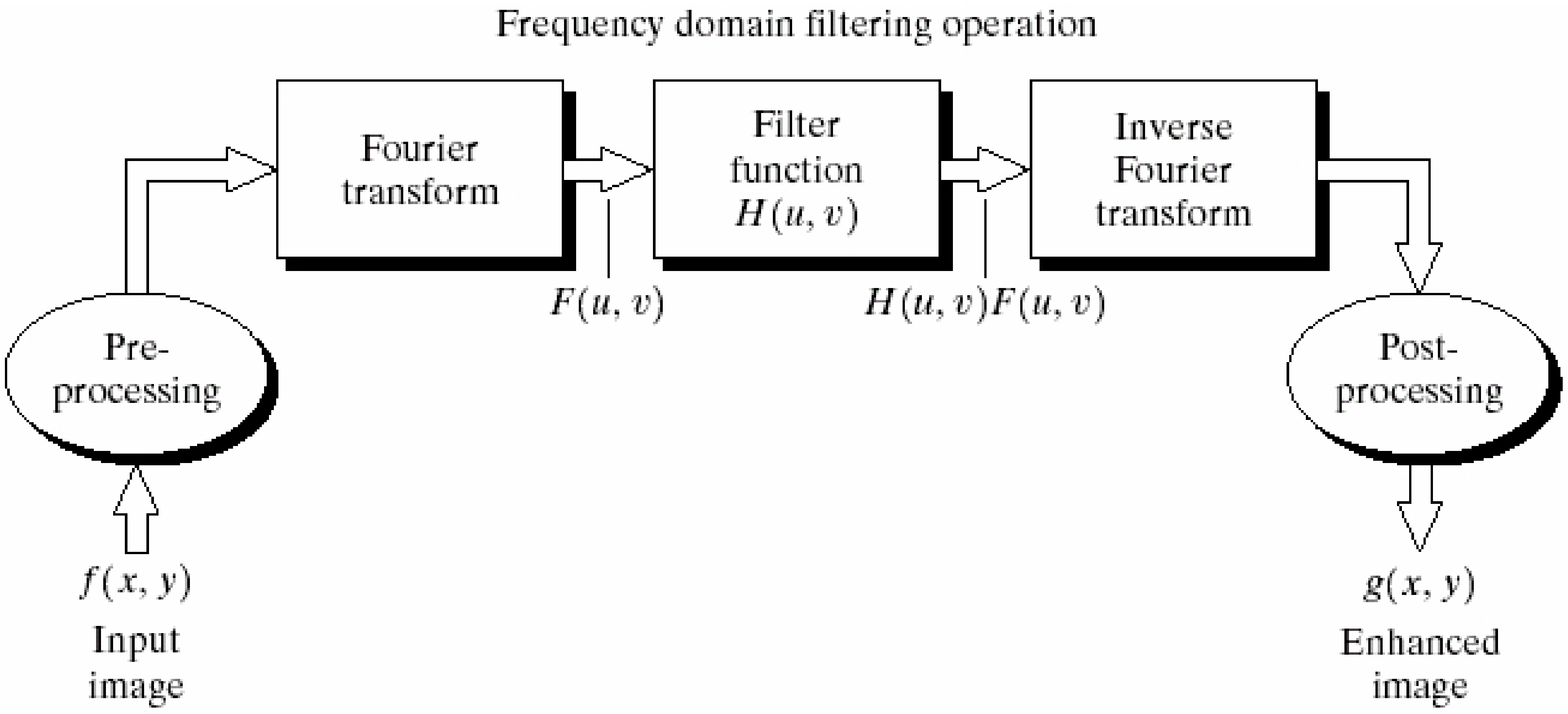
2D Fourier Transform

The extension of the Fourier Transform to handle 2-D signals (e.g. images) is straightforward:

$$\begin{cases} F(u,v) = \int\int_{-\infty}^{\infty} f(x,y)e^{-2\pi i(ux+vy)}dxdy \\ \\ f(x,y) = \int\int_{-\infty}^{\infty} F(u,v)e^{2\pi i(ux+vy)}dudv \end{cases}$$

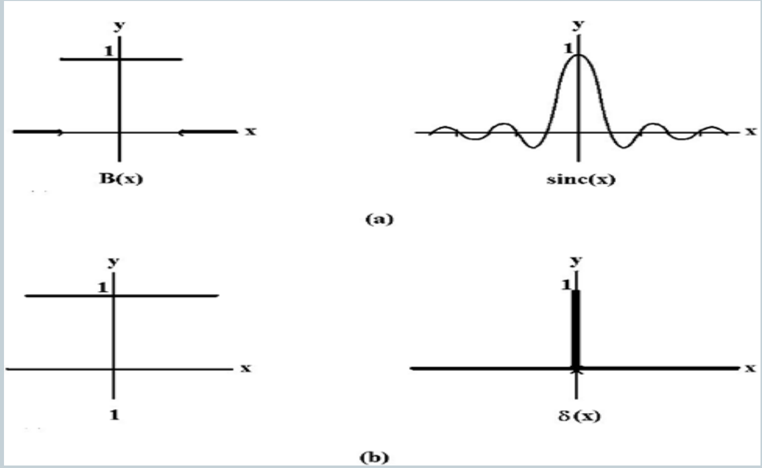
The properties of the 2-D Fourier Transform are the same as those of the 1-D Fourier Transform.

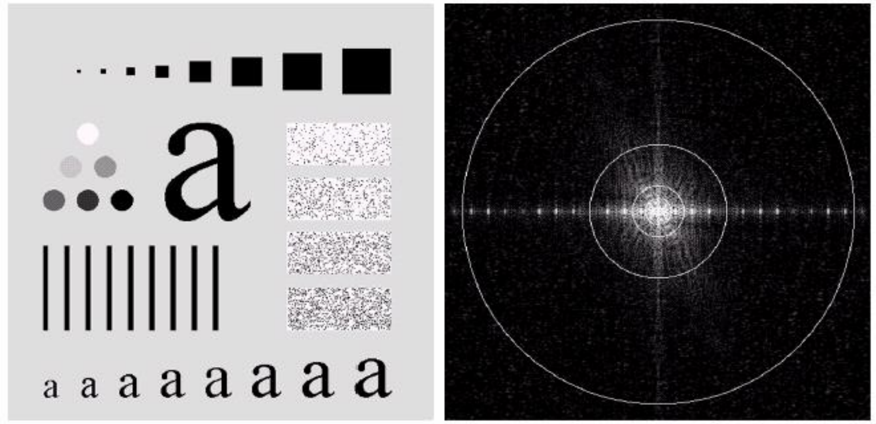
Basic Steps for Frequency Domain Filtering



Frequency Response

- For sharper spatial signals, higher frequencies are required to describe it
  - e.g. *Dirac Delta Function* $\delta(x)$
- For smoother spatial signals, lower frequencies are required to describe it





a b

FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

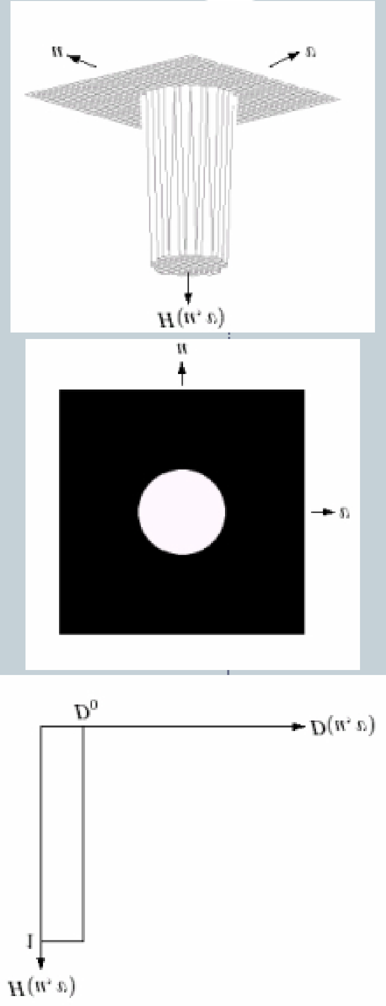
Smoothing Filter

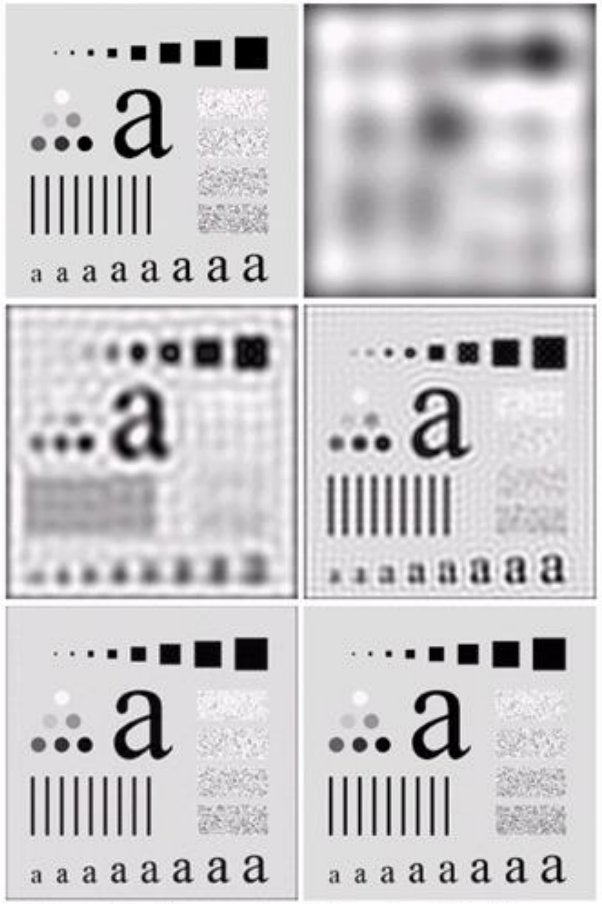
- Objective: remove the sharp transitions in the image
  - Filter out the high frequency components of the signal
  - Maintain the low frequencies

  "Low Pass Filter"

$$H(u,v) = \begin{cases} 1 \; if \; D(u,v) \leq D_o \\ 0 \; if \; D(u,v) > D_o \end{cases}$$

- Two effects
  - Blurring
  - Ringing





a b
c d
e f

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

Smoothing Filter

- Butterworth low pass filter

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

- Gaussian Filter

$$H(u,v) = e^{-D^2(u,v)/2D_o^2}$$

  - the Fourier Transform of a Gaussian function is still a Gaussian

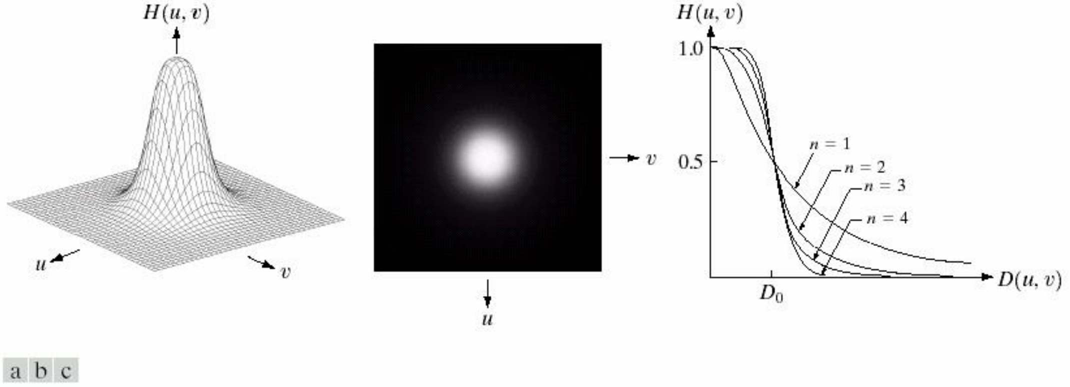


FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

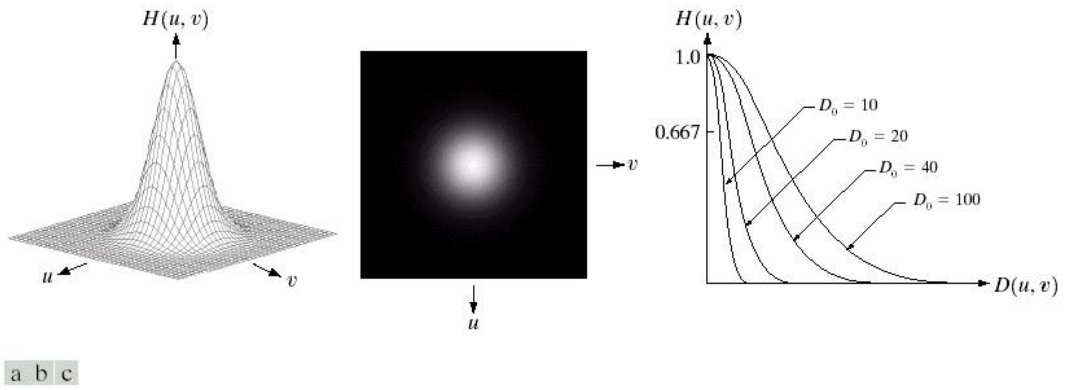


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Smoothing Filter

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

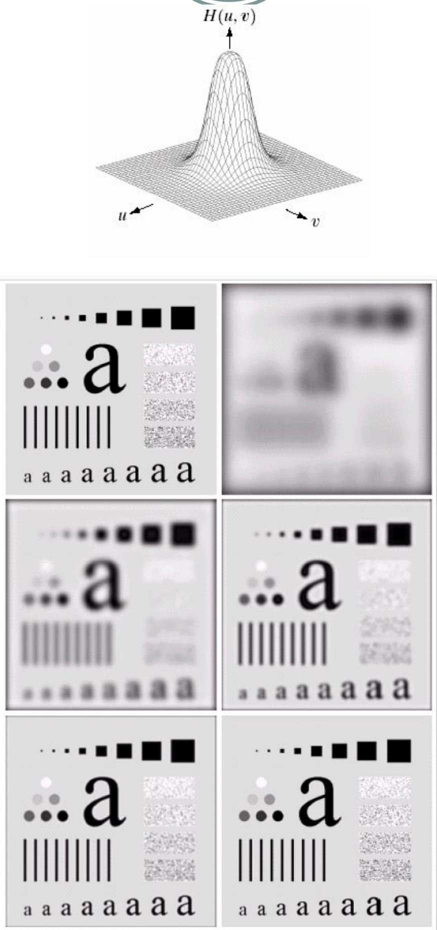


FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

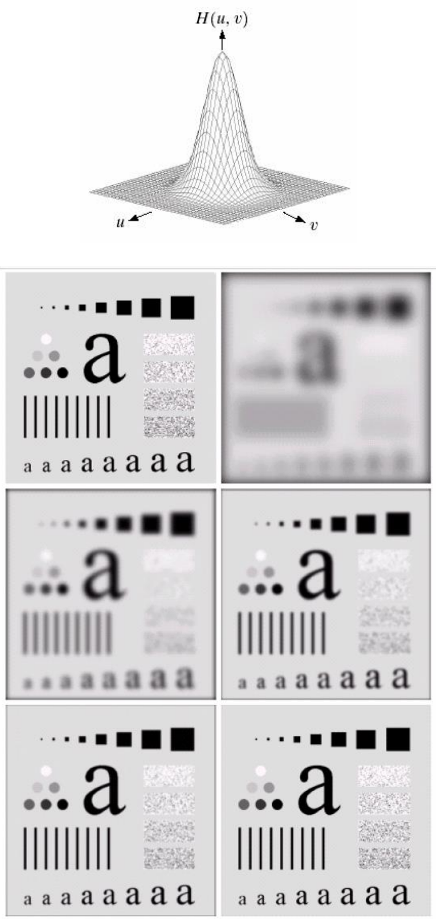


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

Sharpening Filters

- Using only high frequency components when applying inverse Fourier transform results in sharp images

  "High Pass Filter"

- All the low pass filters discussed above can be converted into high pass using

$$H_{hp} = 1 - H_{lp}$$

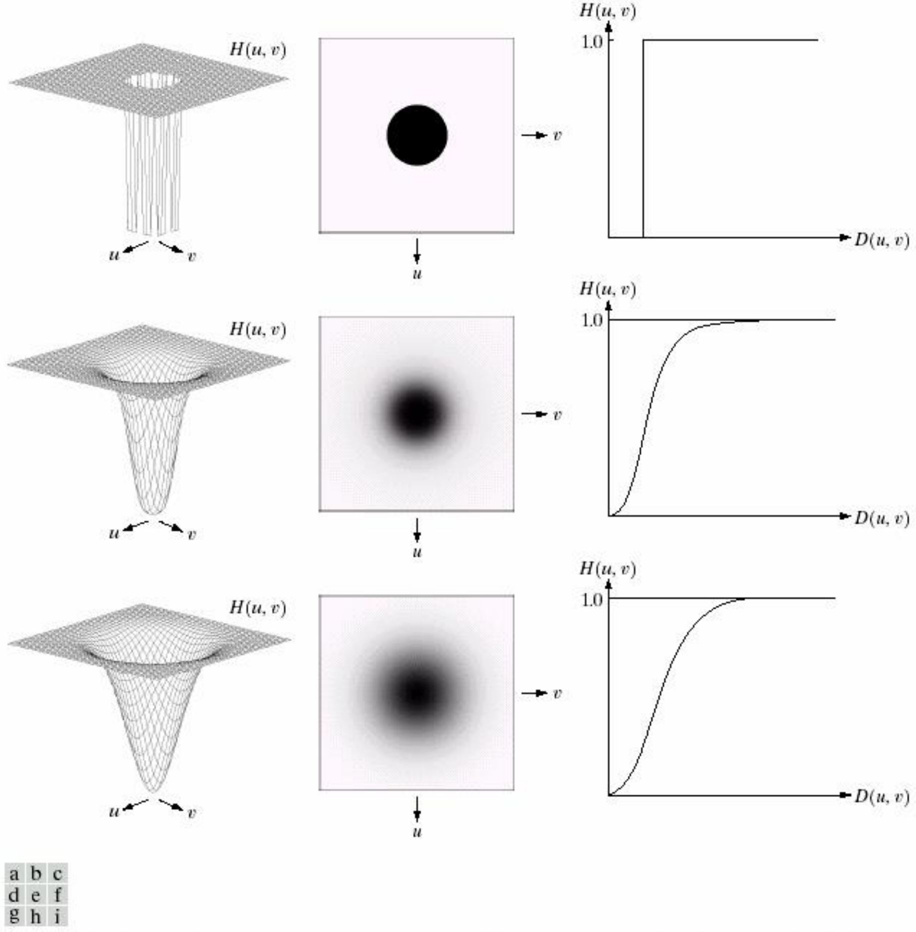
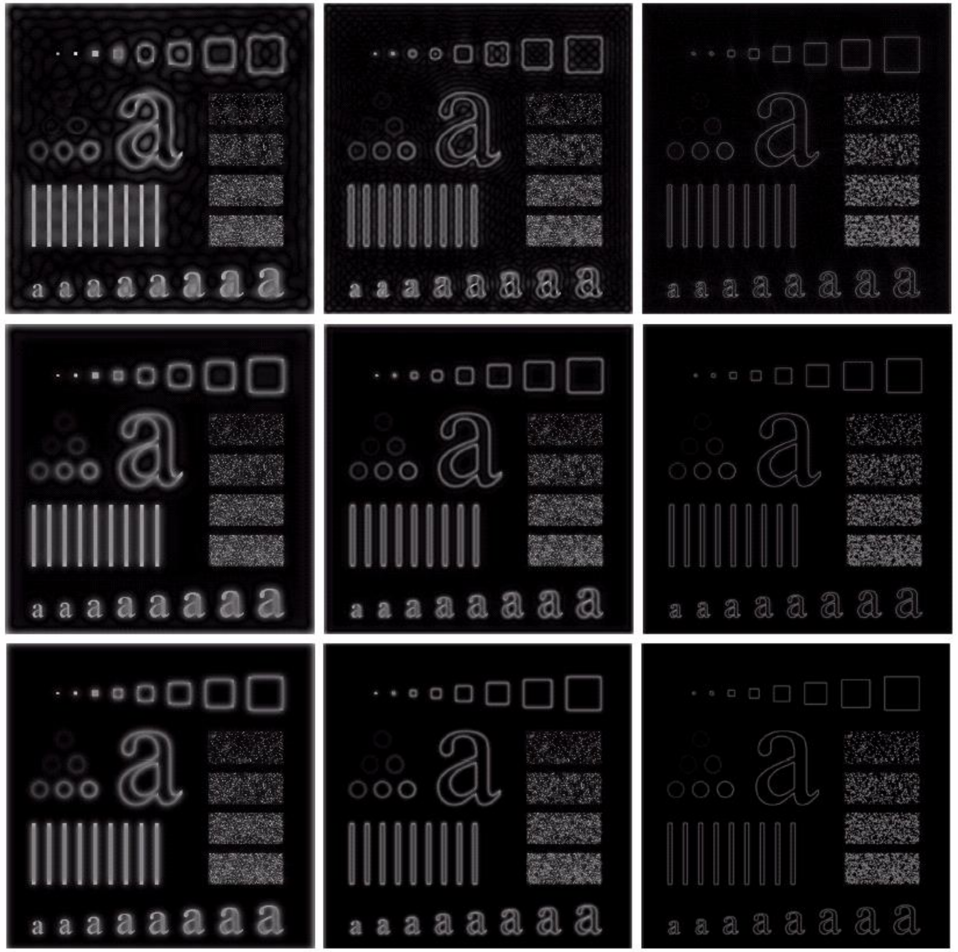


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Sharpening Filters

- Ideal high pass filter :
- Butterworth high pass filter :
- Gaussian high pass filter :



a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

Resources

- Gonzalez and Woods, *Digital Image Processing*, Chapter 3 and 4.8.