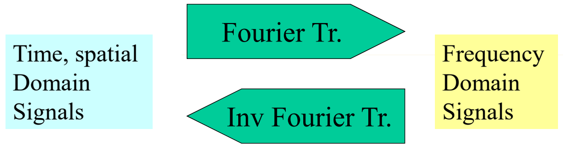
Digital Image Processing
Chapter 4:
Image Enhancement in the Frequency Domain

Fourier Tr. and Frequency Domain



Time, spatial Domain Signals

Fourier Tr.

Inv Fourier Tr.

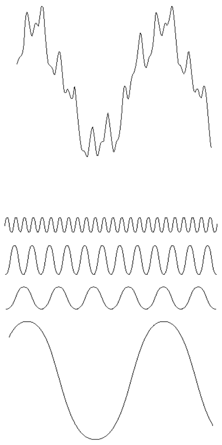
Frequency Domain Signals

1-D, Continuous case

Fourier Tr.: $$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

Inv. Fourier Tr.: $$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

Background: Fourier Series



Fourier series:

Any periodic signals can be viewed as weighted sum of sinusoidal signals with different frequencies

Frequency Domain: view frequency as an independent variable

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Fourier Tr. and Frequency Domain (cont.)

1-D, Discrete case

Fourier Tr.: $$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \qquad u = 0,\ldots,\text{M-1}$$

Inv. Fourier Tr.: $$f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \qquad x = 0,\ldots,\text{M-1}$$

$F(u)$ can be written as

$$F(u) = R(u) + jI(u) \quad \text{or} \quad F(u) = |F(u)| e^{-j\phi(u)}$$

Fourier Tr. and Frequency Domain (cont.)

Euler's formula $e^{j\theta} = \cos\theta + j\sin\theta$

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)[\cos 2\pi ux/M - j\sin 2\pi ux/M]$$
$$= R(u) + jI(u)$$

Or in polar coordinates $F(u) = |F(u)|e^{-j\phi(u)}$

Where the magnitude of spectrum and phase angle are

$$|F(u)| = \sqrt{R(u)^2 + I(u)^2} \qquad \phi(u) = \tan^{-1}\left(\frac{I(u)}{R(u)}\right)$$

Spectral density $P(u) = F(u)^2 = R(u)^2 + I(u)^2$

Relation Between Δx and Δu

For a signal $f(x)$ with M points, let spatial resolution Δx be space between samples in $f(x)$ and let frequency resolution Δu be space between frequencies components in $F(u)$, we have

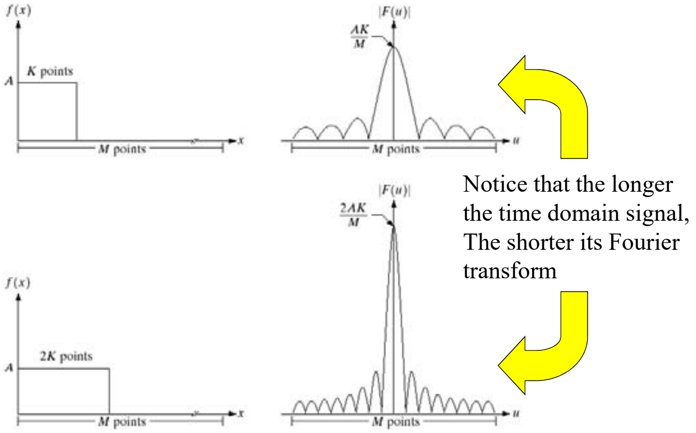
$$\Delta u = \frac{1}{M\Delta x}$$

Example: for a signal f(x) with sampling period 0.5 sec, 100 point, we will get frequency resolution equal to

$$\Delta u = \frac{1}{100\times 0.5} = 0.02 \text{ Hz}$$

This means that in $F(u)$ we can distinguish 2 frequencies that are apart by 0.02 Hertz or more.

Example of 1-D Fourier Transforms



Notice that the longer the time domain signal, The shorter its Fourier transform

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

2-Dimensional Discrete Fourier Transform

For an image of size MxN pixels

2-D DFT

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

u = frequency in x direction, u = 0 ,…, M-1
v = frequency in y direction, v = 0 ,…, N-1

2-D IDFT

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$$

x = 0 ,…, M-1
y = 0 ,…, N-1

2-Dimensional Discrete Fourier Transform (cont.)

$F(u,v)$ can be written as

$$F(u,v) = R(u,v) + jI(u,v) \quad \text{or} \quad F(u,v) = |F(u,v)|e^{-j\phi(u,v)}$$

where

$$|F(u,v)| = \sqrt{R(u,v)^2 + I(u,v)^2} \qquad \phi(u,v) = \tan^{-1}\left(\frac{I(u,v)}{R(u,v)}\right)$$

For the purpose of viewing, we usually display only the Magnitude part of $F(u,v)$

Relation Between Spatial and Frequency Resolutions

$$\Delta u = \frac{1}{M\Delta x} \qquad \Delta v = \frac{1}{N\Delta y}$$

where

Δx = spatial resolution in x direction
Δy = spatial resolution in y direction

(Δx and Δy are pixel width and height.)

Δu = frequency resolution in x direction
Δv = frequency resolution in y direction
N,M = image width and height

Computational Advantage of FFT Compared to DFT

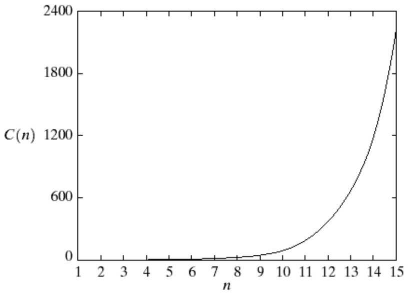
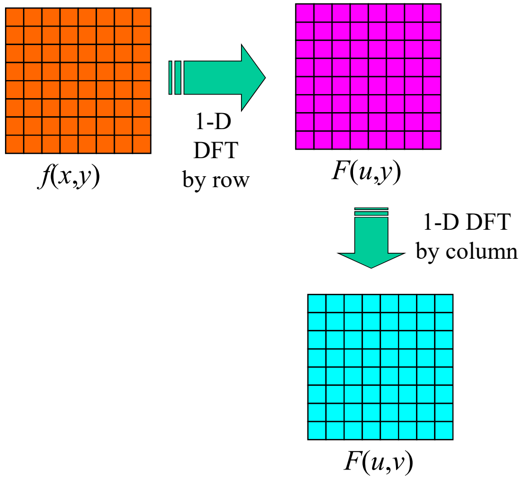


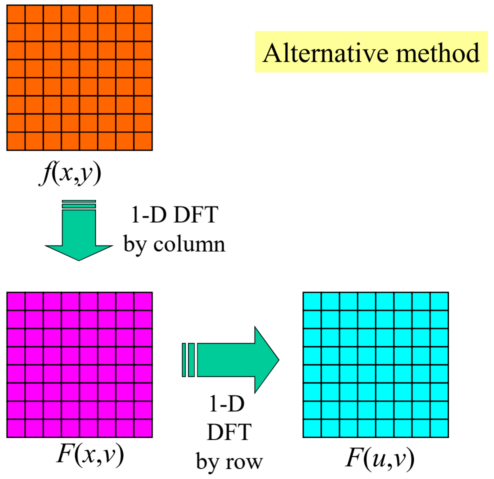
FIGURE 4.42 Computational advantage of the FFT over a direct implementation of the 1-D DFT. Note that the advantage increases rapidly as a function of n.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

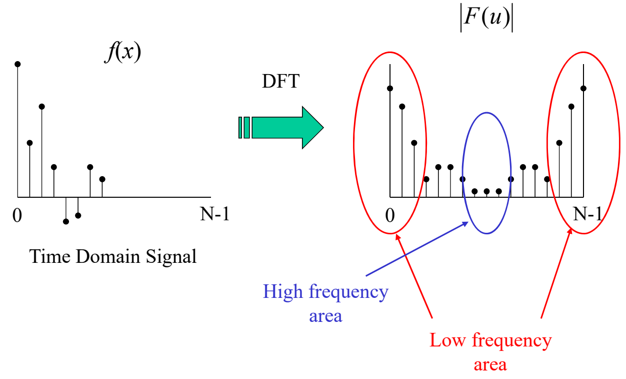
How to Perform 2-D DFT by Using 1-D DFT



How to Perform 2-D DFT by Using 1-D DFT (cont.)



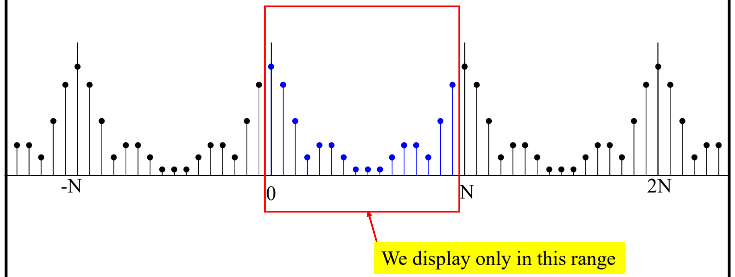
Conventional Display for 1-D DFT



The graph $F(u)$ is not easy to understand !

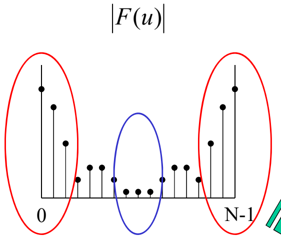
Periodicity of 1-D DFT

From DFT: $$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M}$$

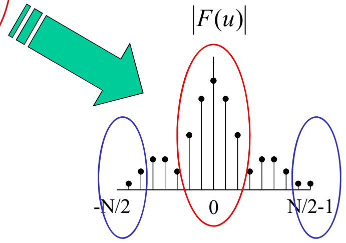


DFT repeats itself every N points (Period = N) but we usually display it for $n = 0, \ldots, N\text{-}1$

Conventional Display for DFT : FFT Shift



FFT Shift: Shift center of the graph $F(u)$ to 0 to get better Display which is easier to understand.



○ High frequency area

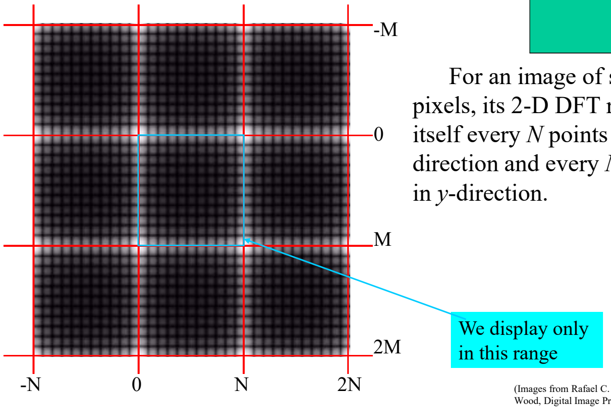
○ Low frequency area

Periodicity of 2-D DFT

2-D DFT: $F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M}$

g(*x*,*y*)

For an image of size *N*x*M* pixels, its 2-D DFT repeats itself every *N* points in *x*-direction and every *M* points in *y*-direction.

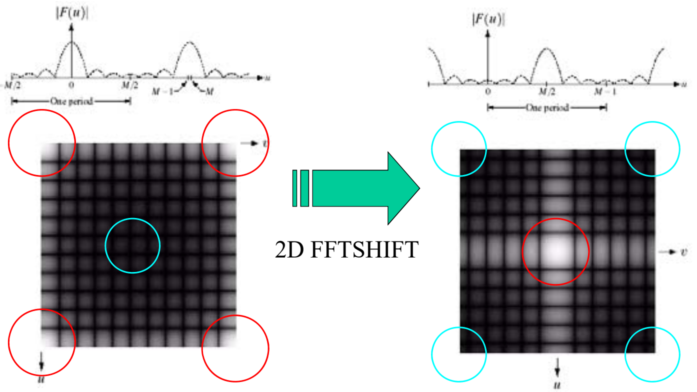


We display only in this range

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

2-D FFT Shift : Better Display of 2-D DFT

2-D FFT Shift is a MATLAB function: Shift the zero frequency of *F*(*u*,*v*) to the center of an image.

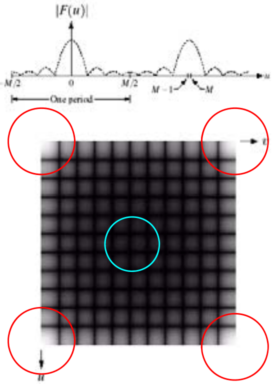


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

○ High frequency area ○ Low frequency area

Conventional Display for 2-D DFT

F(*u*,*v*) has low frequency areas at corners of the image while high frequency areas are at the center of the image which is inconvenient to interpret.

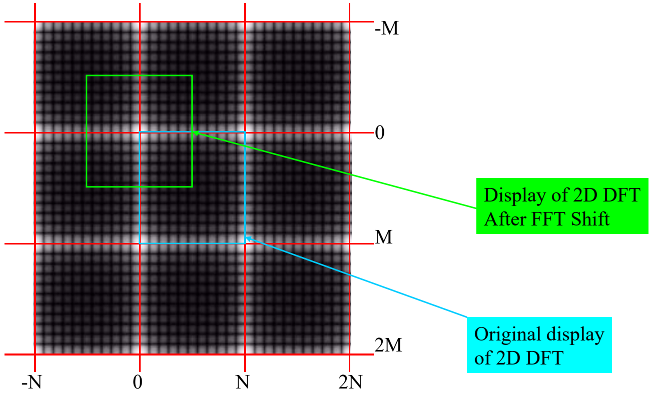


○ High frequency area

○ Low frequency area

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

2-D FFT Shift (cont.) : How it works



Display of 2D DFT After FFT Shift

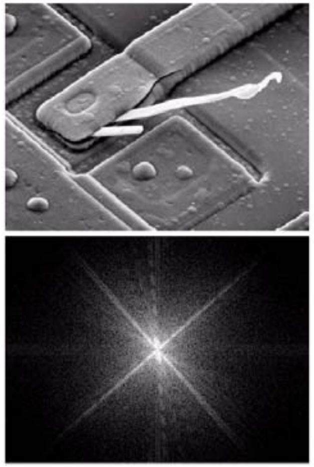
Original display of 2D DFT

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

2-Dimensional FFT shift

$$\Im[f(x,y)(-1)^{x+y}] = F(u - M/2, v - N/2)$$

Example of 2-D DFT



a
b

FIGURE 4.4
(a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

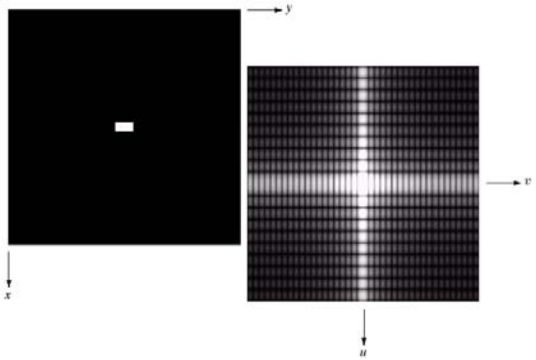
Notice that direction of an object in spatial image and Its Fourier transform are orthogonal to each other.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example of 2-D DFT

a b

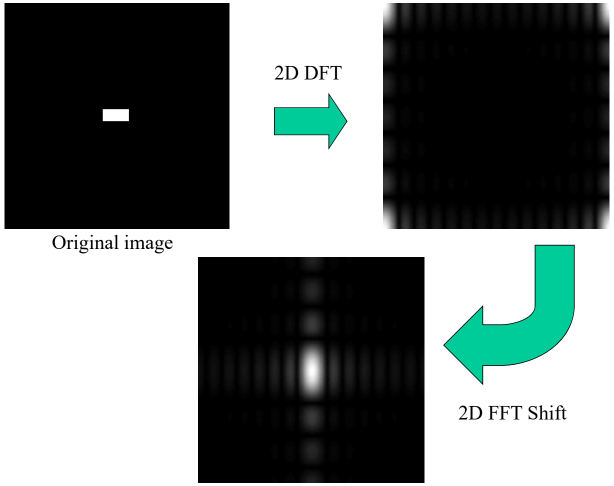
FIGURE 4.3
(a) Image of a 20 × 40 white rectangle on a black background of size 512 × 512 pixels. (b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.



Notice that the longer the time domain signal,
The shorter its Fourier transform

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example of 2-D DFT

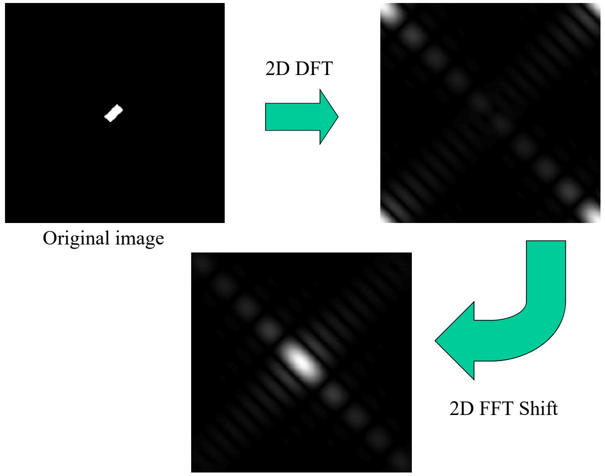


2D DFT

Original image

2D FFT Shift

Example of 2-D DFT



Spatial filtering

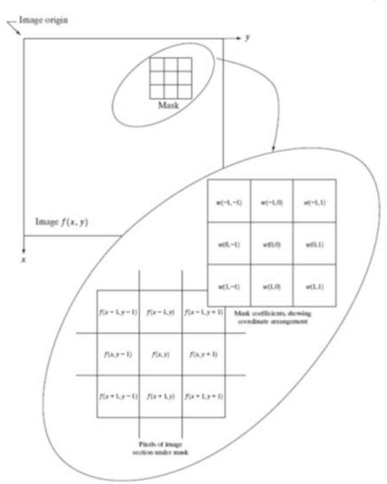


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

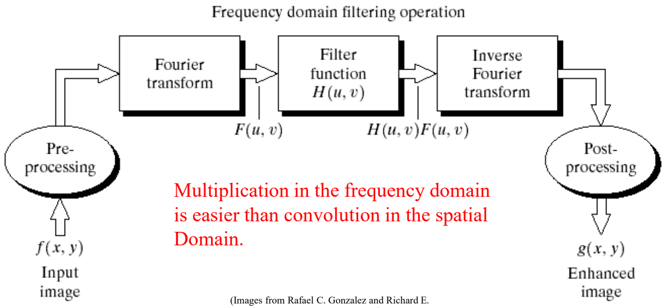
$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Basic Concept of Filtering in the Frequency Domain

From Fourier Transform Property:

$$g(x,y) = f(x,y) * h(x,y) \Leftrightarrow F(u,v) \cdot H(u,v) = G(u,v)$$

We cam perform filtering process by using

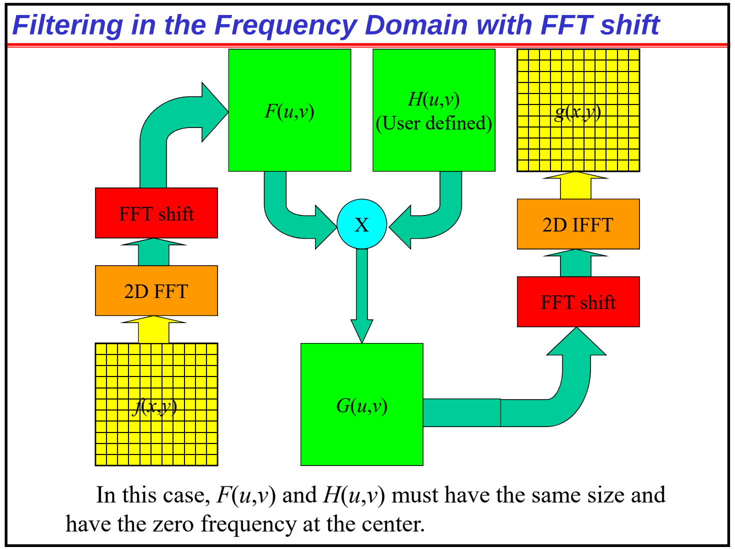


Multiplication in the frequency domain is easier than convolution in the spatial Domain.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

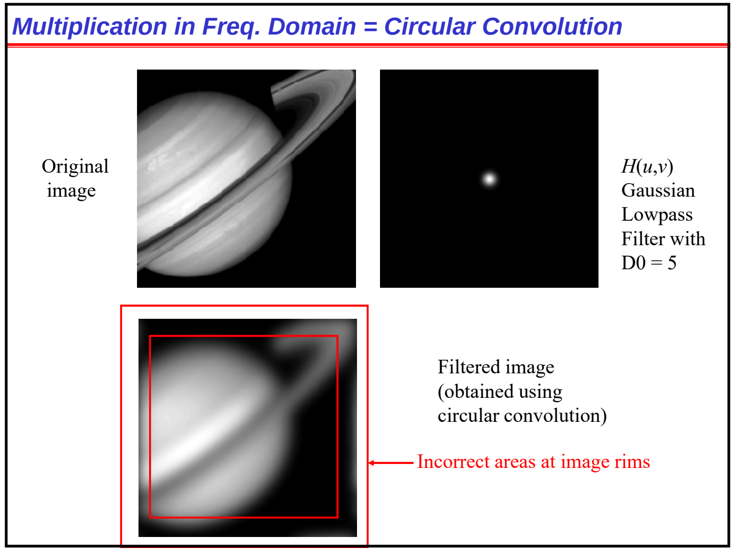
Discrete convolution of f(x,y) and h(x,y)

$$f(x,y) * h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x-m, y-n)$$



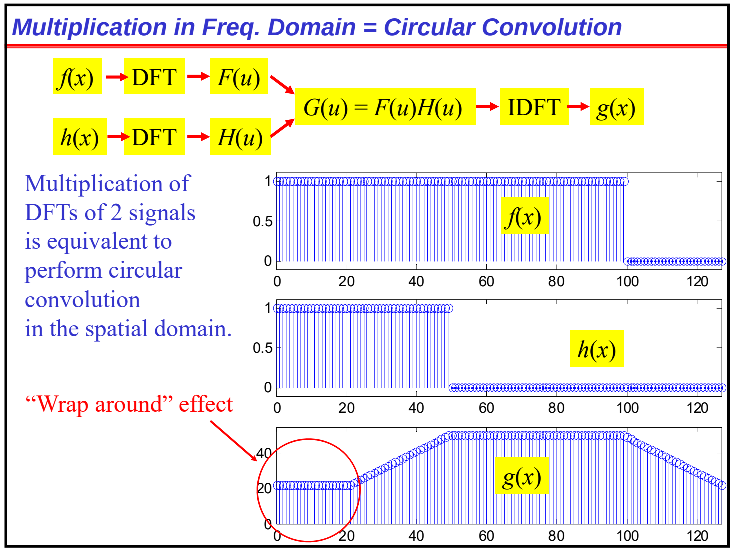
Filtering in the Frequency Domain with FFT shift

In this case, *F*(*u*,*v*) and *H*(*u*,*v*) must have the same size and have the zero frequency at the center.



Multiplication in Freq. Domain = Circular Convolution

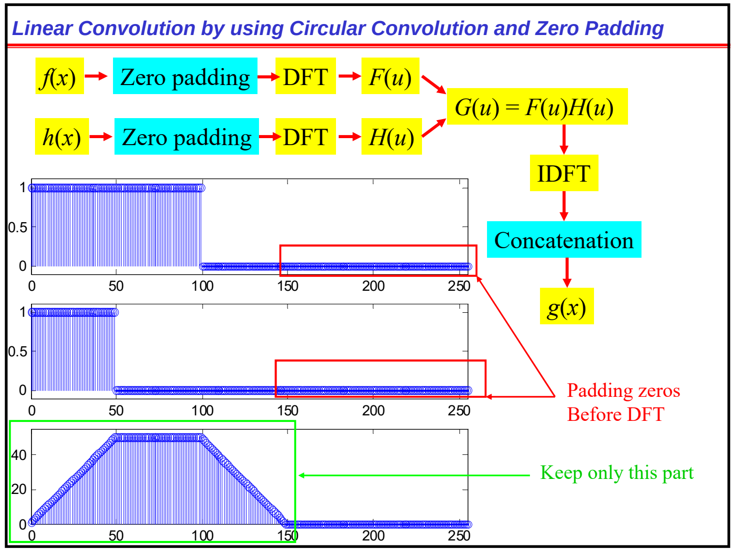
Original image
H(u,v) Gaussian Lowpass Filter with D0 = 5
Filtered image (obtained using circular convolution)
Incorrect areas at image rims



Multiplication in Freq. Domain = Circular Convolution

Multiplication of DFTs of 2 signals is equivalent to perform circular convolution in the spatial domain.

"Wrap around" effect



Linear Convolution by using Circular Convolution and Zero Padding

The need for Padding

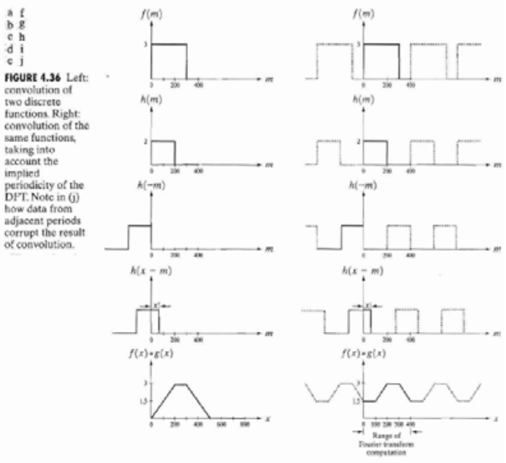


FIGURE 4.36 Left: convolution of two discrete functions. Right: convolution of the same functions, taking into account the implied periodicity of the DFT. Note in (j) how data from adjacent periods corrupt the result of convolution.

The need for Padding

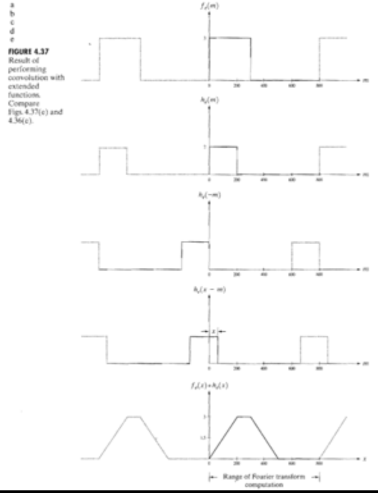


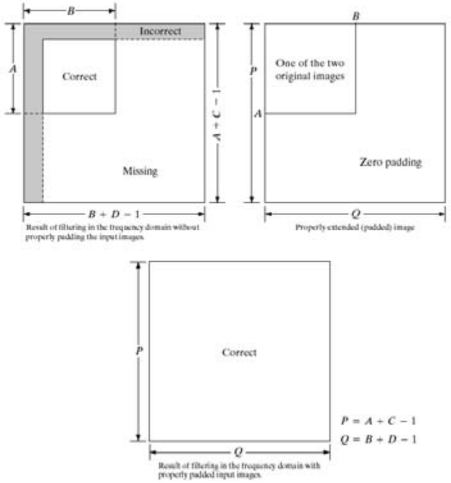
FIGURE 4.37 Result of performing convolution with extended functions. Compare Figs. 4.37(c) and 4.36(e).

The need for Padding

Failure to handle the periodicity issue properly will give incorrect results if the convolution function is obtained using the Fourier transform. The result will have erroneous data at the beginning and have missing data at the end.

The solution to this problem is straightforward. Assume that f and h consist of ***A*** and ***B*** points**,** respectively**.** We append ***zeros*** to both functions so that they have identical periods, denoted by P.

Linear Convolution by using Circular Convolution and Zero Padding

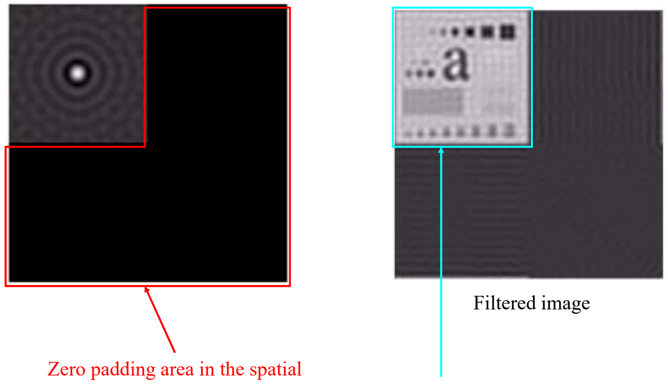

B
Incorrect
A
Correct
Missing
A + C − 1
B + D − 1
Result of filtering in the frequency domain without properly padding the input images.
B
P
One of the two original images
A
Zero padding
Q
Properly extended (padded) image
P
Correct
Q
Result of filtering in the frequency domain with properly padded input images.
P = A + C − 1
Q = B + D − 1

a b
c

FIGURE 4.38 Illustration of the need for function padding. (a) Result of performing 2-D convolution without padding. (b) Proper function padding. (c) Correct convolution result.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Linear Convolution by using Circular Convolution and Zero Padding



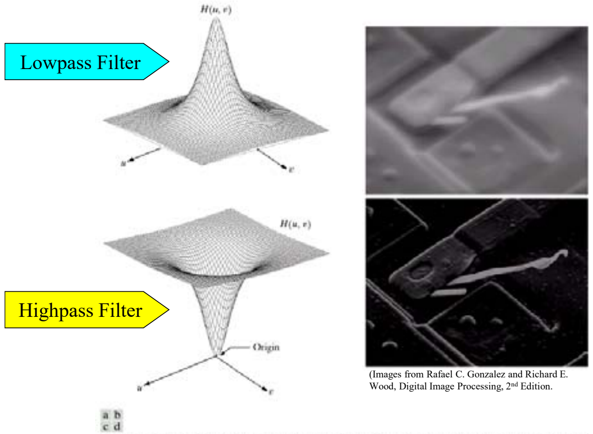
Filtered image

Zero padding area in the spatial Domain of the mask image (the ideal lowpass filter)

Only this area is kept.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

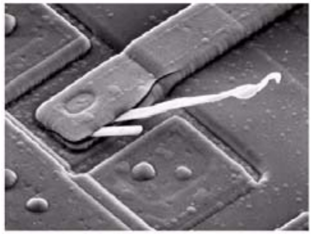
Filtering in the Frequency Domain : Example



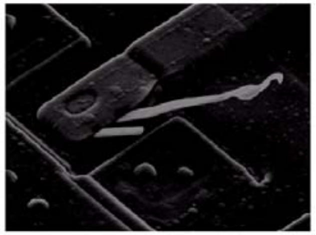
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

Filtering in the Frequency Domain : Example



In this example, we set F(0,0) to zero which means that the zero frequency component is removed.

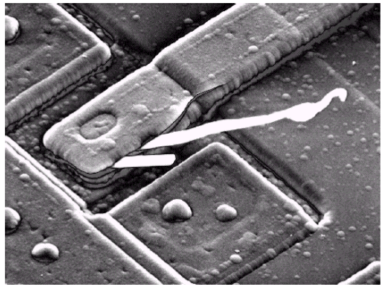


Note: Zero frequency = average intensity of an image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Filtering in the Frequency Domain : Example (cont.)

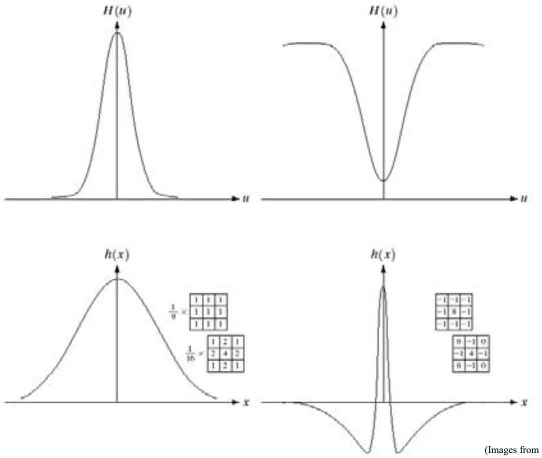
FIGURE 4.8 Result of highpass filtering the image in Fig. 4.4(a) with the filter in Fig. 4.7(c), modified by adding a constant of one-half the filter height to the filter function. Compare with Fig. 4.4(a).



Result of Sharpening Filter

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Filter Masks and Their Fourier Transforms

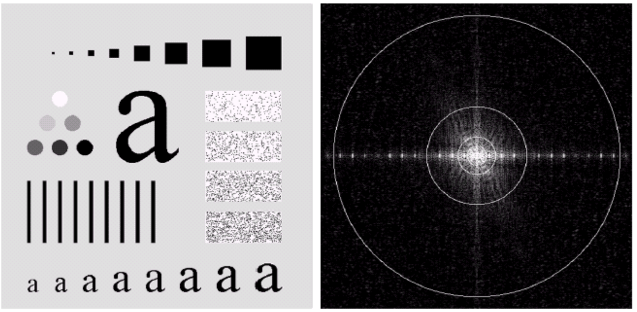


a b
c d

FIGURE 4.9 (a) Gaussian frequency domain lowpass filter. (b) Gaussian frequency domain highpass filter. (c) Corresponding lowpass spatial filter. (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Examples of Ideal Lowpass Filters



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

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.

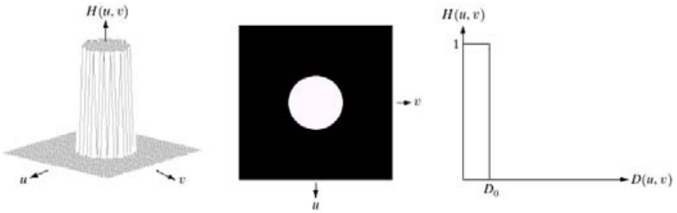
The smaller D_0, the more high frequency components are removed.

Ideal Lowpass Filter

Ideal LPF Filter Transfer function

$$H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) > D_0 \end{cases}$$

where $D(u,v)$ = Distance from (u,v) to the center of the mask.

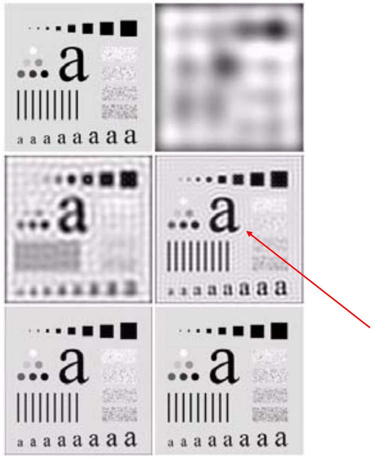


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

a b c

FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

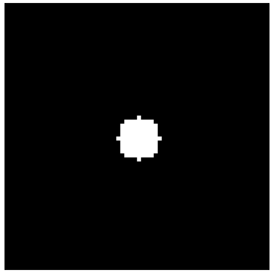
Results of Ideal Lowpass Filters



Ringing effect can be obviously seen!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

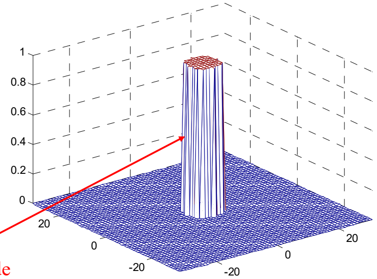
How ringing effect happens



Ideal Lowpass Filter
with $D_0 = 5$

$$H(u,v) = \begin{cases} 1 & D(u,v) \le D_0 \\ 0 & D(u,v) > D_0 \end{cases}$$

Surface Plot



Abrupt change in the amplitude

How ringing effect happens (cont.)

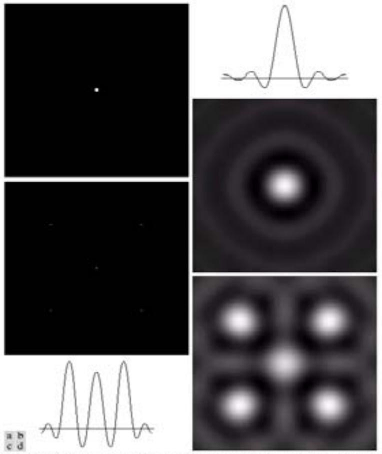
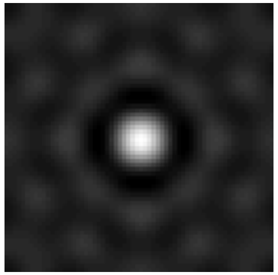


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

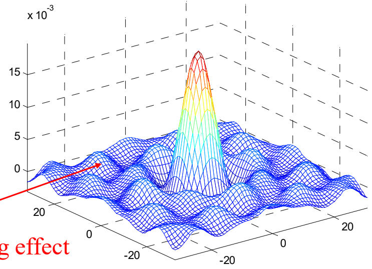
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

How ringing effect happens (cont.)



Spatial Response of Ideal Lowpass Filter with $D_0 = 5$

Surface Plot



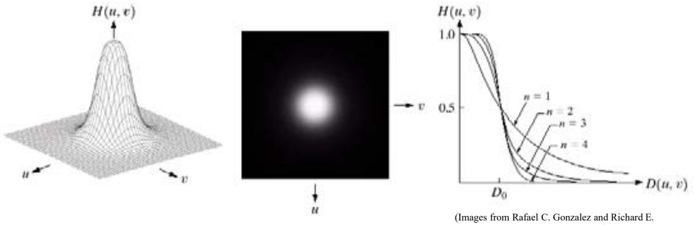
Ripples that cause ringing effect

Butterworth Lowpass Filter

Transfer function

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2N}}$$

Where D_0 = Cut off frequency, N = filter order.

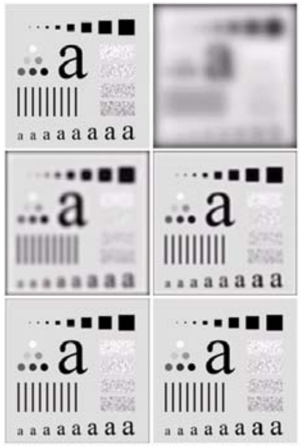


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

a b c

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.

Results of Butterworth Lowpass Filters



a b c d e f 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.

There is less ringing effect compared to those of ideal lowpass filters!

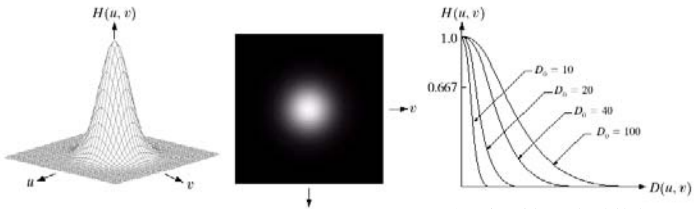
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Gaussian Lowpass Filter

Transfer function

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

Where D_0 = spread factor.



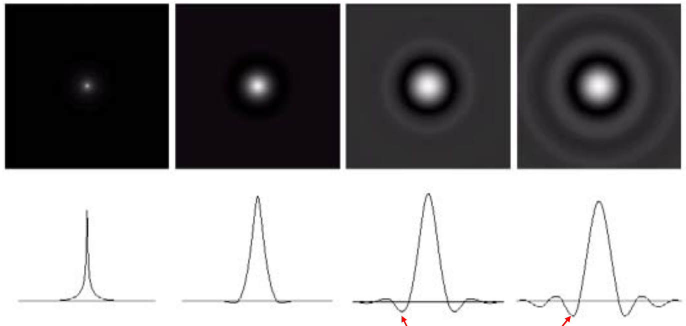
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

a b c

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.

Note: the Gaussian filter is the only filter that has no ripple and hence no ringing effect.

Spatial Masks of the Butterworth Lowpass Filters



a b c d

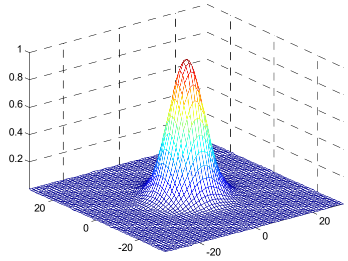
FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Some ripples can be seen.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

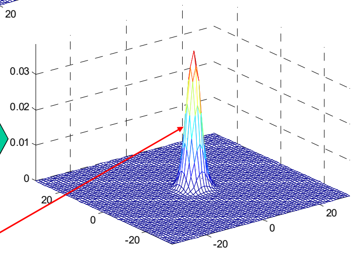
Gaussian Lowpass Filter (cont.)

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$



Gaussian lowpass filter with $D_0 = 5$

Spatial respones of the Gaussian lowpass filter with $D_0 = 5$



Gaussian shape

Results of Gaussian Lowpass Filters

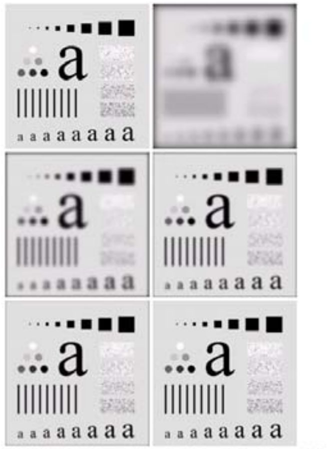


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

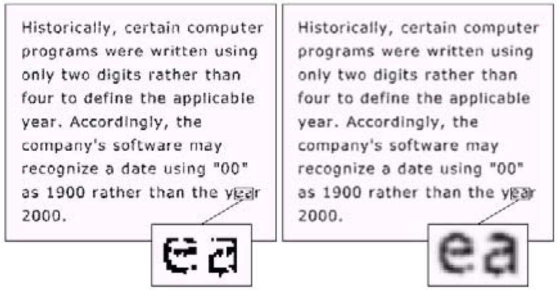
No ringing effect!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Application of Gaussian Lowpass Filters

a b

FIGURE 4.19 (a) Sample text of poor resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

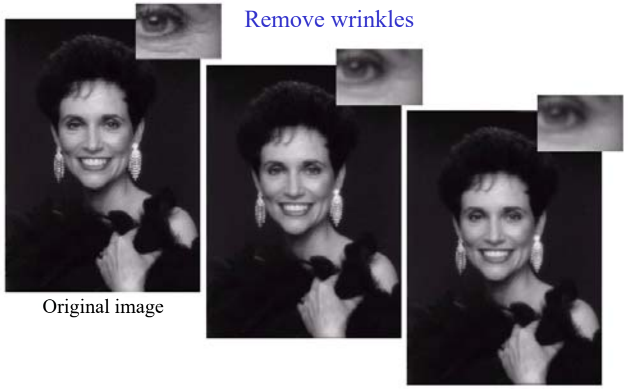
Original image

Better Looking

The GLPF can be used to remove jagged edges and “repair” broken characters.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Application of Gaussian Lowpass Filters (cont.)



Remove wrinkles

Original image

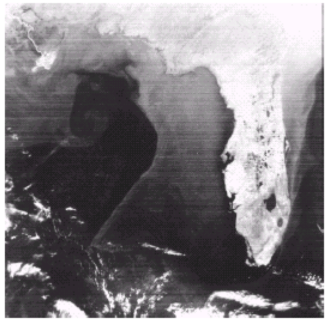
Softer-Looking

a b c

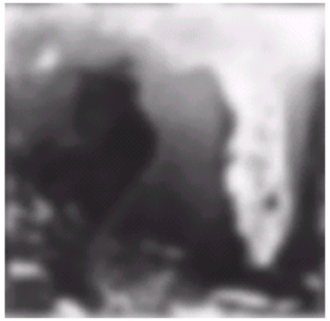
FIGURE 4.20 (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Application of Gaussian Lowpass Filters (cont.)



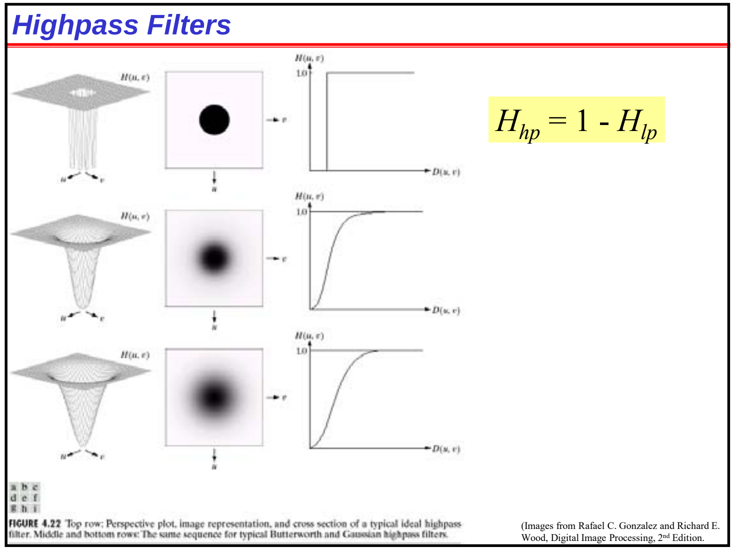
Original image : The gulf of Mexico and Florida from NOAA satellite.



Filtered image

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Remove artifact lines: this is a simple but crude way to do it!

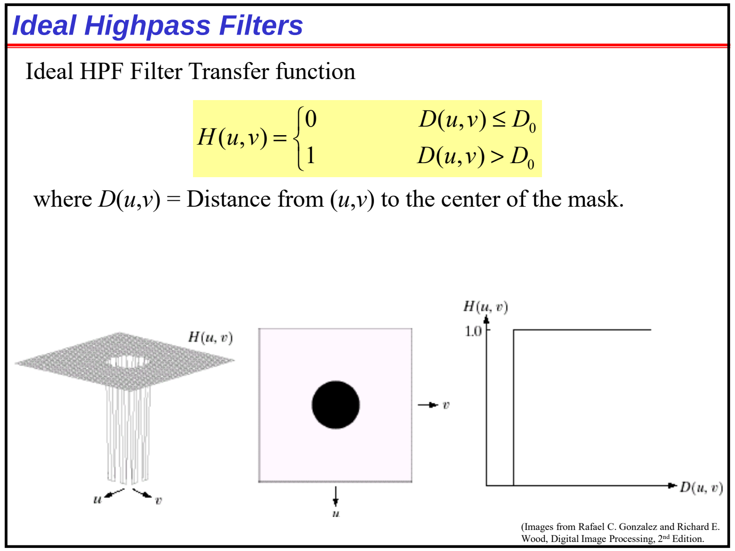


Highpass Filters

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

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.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



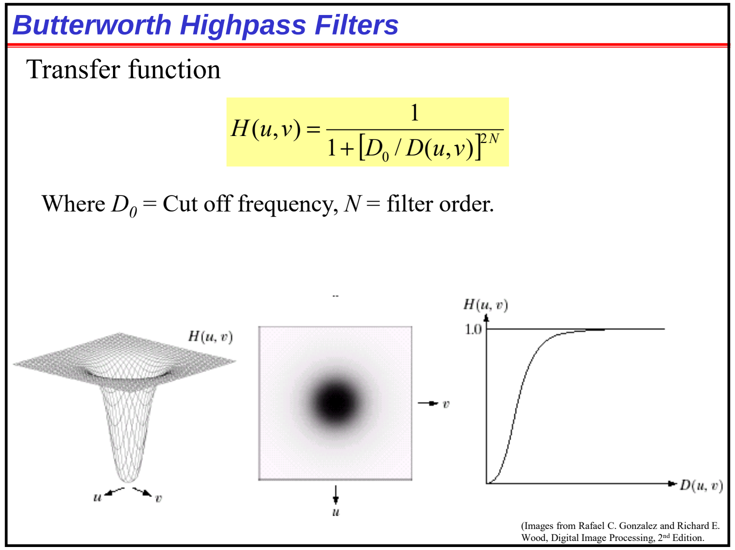
Ideal Highpass Filters

Ideal HPF Filter Transfer function

$$H(u,v) = \begin{cases} 0 & D(u,v) \le D_0 \\ 1 & D(u,v) > D_0 \end{cases}$$

where $D(u,v)$ = Distance from (u,v) to the center of the mask.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



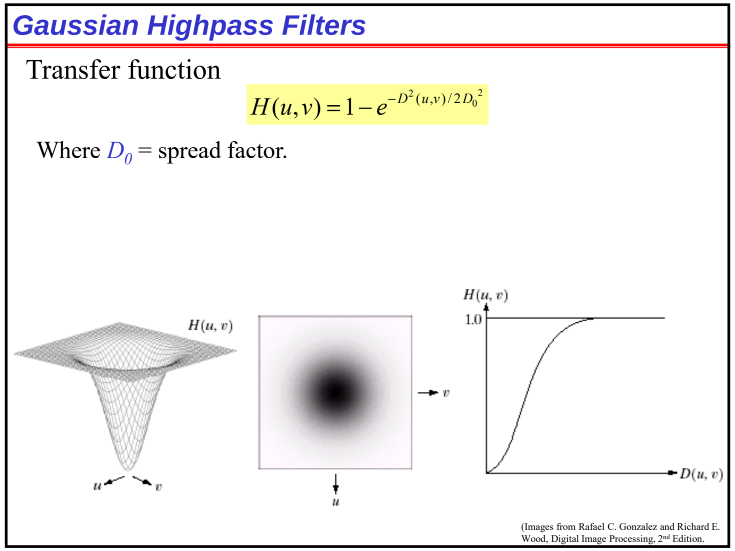
Butterworth Highpass Filters

Transfer function

$$H(u,v) = \frac{1}{1 + [D_0 / D(u,v)]^{2N}}$$

Where D_0 = Cut off frequency, N = filter order.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



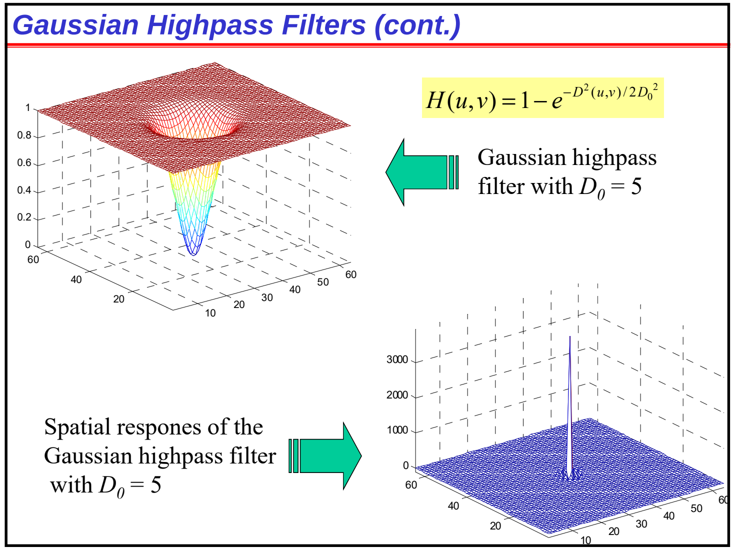
Gaussian Highpass Filters

Transfer function

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Where D_0 = spread factor.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

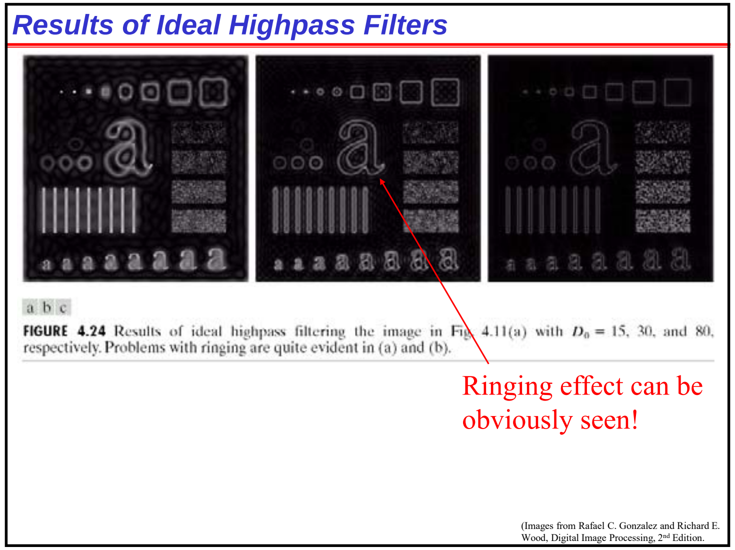


Gaussian Highpass Filters (cont.)

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Gaussian highpass filter with $D_0 = 5$

Spatial respones of the Gaussian highpass filter with $D_0 = 5$



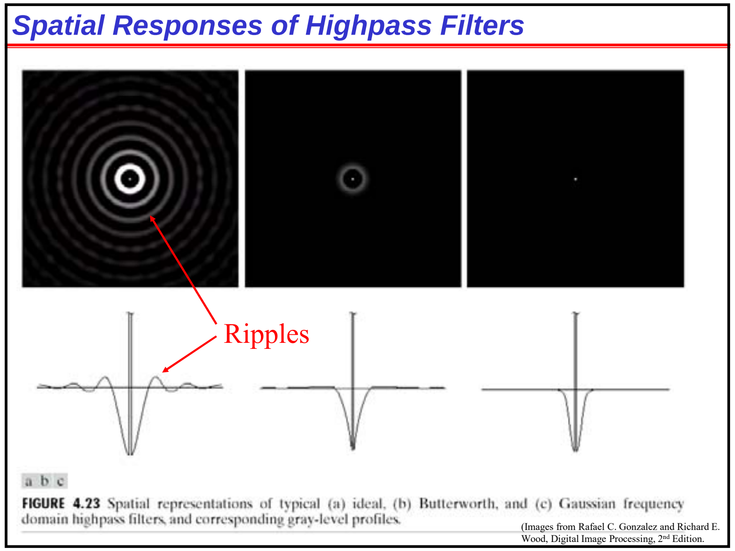
Results of Ideal Highpass Filters

a b c

FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

Ringing effect can be obviously seen!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



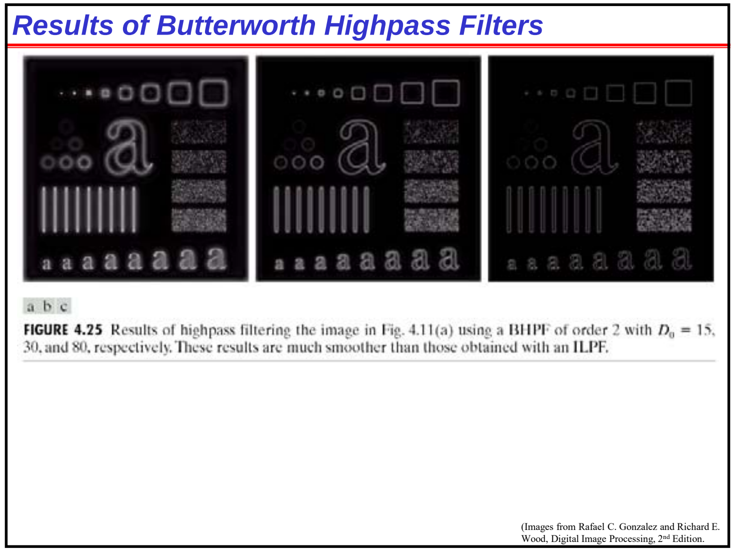
Spatial Responses of Highpass Filters

Ripples

a b c

FIGURE 4.23 Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



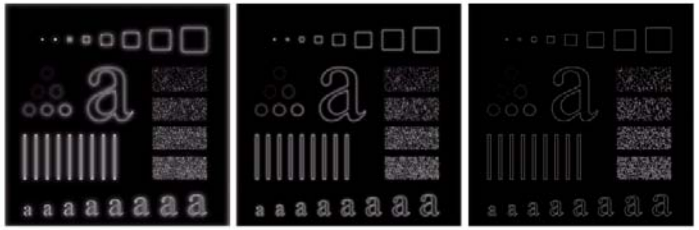
Results of Butterworth Highpass Filters

a b c

FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Results of Gaussian Highpass Filters



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.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Laplacian Filter in the Frequency Domain

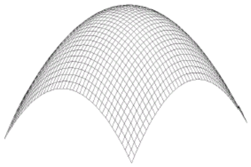
From Fourier Tr. Property: $\dfrac{d^n f(x)}{dx^n} \Leftrightarrow (ju)^n F(u)$

Then for Laplacian operator

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \Leftrightarrow -\left(u^2 + v^2\right) F(u,v)$$

We get

$$\nabla^2 \Leftrightarrow -\left(u^2 + v^2\right)$$



Surface plot

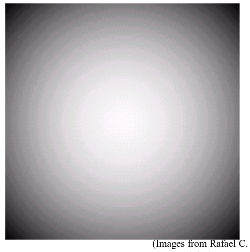


Image of $-(u^2+v^2)$

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Laplacian Filter in the Frequency Domain

Laplacian

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

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

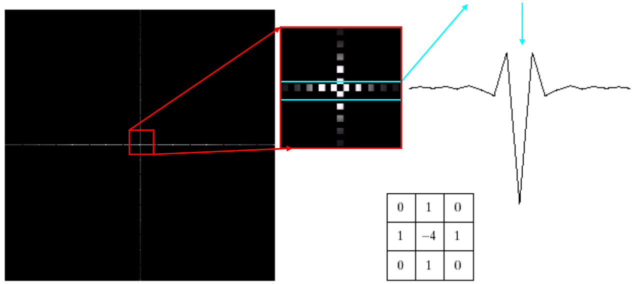
✓**Laplacian mask**

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

0	1	0
1	-4	1
0	1	0

Laplacian Filter in the Frequency Domain (cont.)

Spatial response of $-(u^2+v^2)$



0	1	0
1	–4	1
0	1	0

Laplacian mask in Chapter 3

Laplacian Filter in the Frequency Domain

The Laplacian in the Frequency Domain

$$\Im\left[\frac{\partial f(x,y)}{\partial x^2}+\frac{\partial f(x,y)}{\partial y^2}\right]=(ju)^2F(u)+(ju)^2F(v) \qquad \textbf{(4.4-5)}$$

$$\Im\left[\frac{d^n f(x)}{dx^n}\right]=(ju)^n F(u)$$

$$=-(u^2+v^2)F(u,v) \qquad \textbf{(4.4-6)}$$

Defined in Eq. (3.7-1)

$$\nabla^2 f=\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2} \qquad \textbf{(3.7-1)}$$

Laplacian Filter in the Frequency Domain

The enhancement : $g(x,y)=f(x,y)-\nabla^2 f(x,y)$

In freq. domain : $H(u,v)=1+[(u-M/2)^2+(v-N/2)^2]$

By inverse transform

$$g(x,y)=\Im^{-1}\{1+[(u-M/2)^2+(v-N/2)^2]F(u,v)\}$$

Laplacian Filter in the Frequency Domain

The *Laplacian* can be implemented in the *frequency domain* by using the filter.

$$H(u,v)=-(u^2+v^2) \qquad \textbf{(4.4-8)}$$

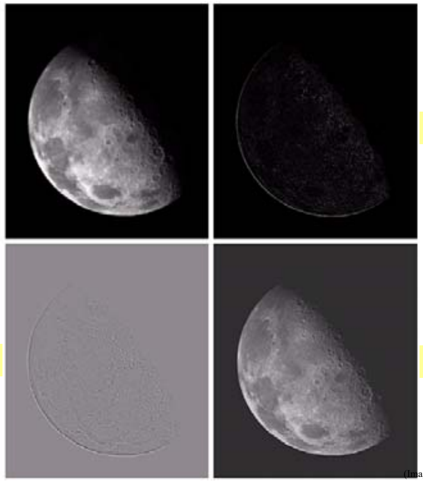
The *center* of the filter function also needs to be *shifted.*

$$H(u,v)=-[(u-M/2)^2+(v-N/2)^2] \qquad \textbf{(4.4-9)}$$

The ***Laplacian-filtered image*** in the spatial domain is obtained by computing the *inverse* Fourier transform of $H(u,v)F(u,v)$.

$$\nabla^2 f(x,y)=\Im^{-1}\{-[(u-M/2)^2+(v-N/2)^2]F(u,v)\} \qquad \textbf{(4.4-10)}$$

Sharpening Filtering in the Frequency Domain (cont.)



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Unsharp Masking, High-Boost Filtering

Unsharp Masking, High-Boost Filtering

High-boost filtering

- ✓ A *generalization* of unsharp masking
- ✓ To *increase the contribution* made by the original image to the overall filtered result.

Unsharp masking

- ✓ Consists *simply of generating a sharp image* by subtracting from an image a blurred version of itself.

Sharpening Filtering in the Frequency Domain

Spatial Domain

$f_{hp}(x,y) = f(x,y) - f_{lp}(x,y)$ Unsharp masking

$f_{hb}(x,y) = Af(x,y) - f_{lp}(x,y)$ High boost filtering

$f_{hb}(x,y) = (A-1)f(x,y) + f(x,y) - f_{lp}(x,y)$

$f_{hb}(x,y) = (A-1)f(x,y) + f_{hp}(x,y)$

Frequency Domain Filter

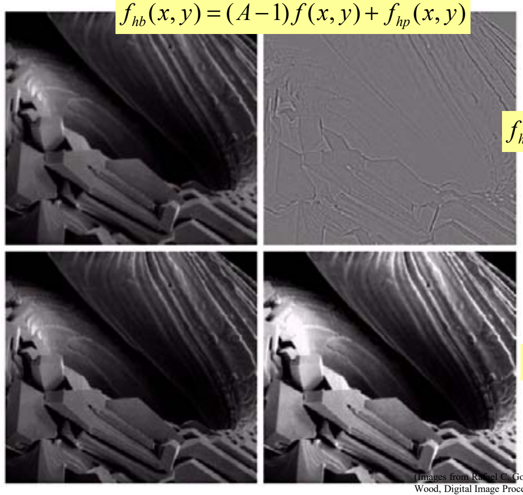
$H_{hp}(u,v) = 1 - H_{lp}(u,v)$

$H_{hb}(u,v) = (A-1) + H_{hp}(u,v)$

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Sharpening Filtering in the Frequency Domain (cont.)

$f_{hb}(x,y) = (A-1)f(x,y) + f_{hp}(x,y)$



f

$f_{hp} = \nabla^2 P$

$A = 2$

$A = 2.7$

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

High-Frequency Emphasis Filtering

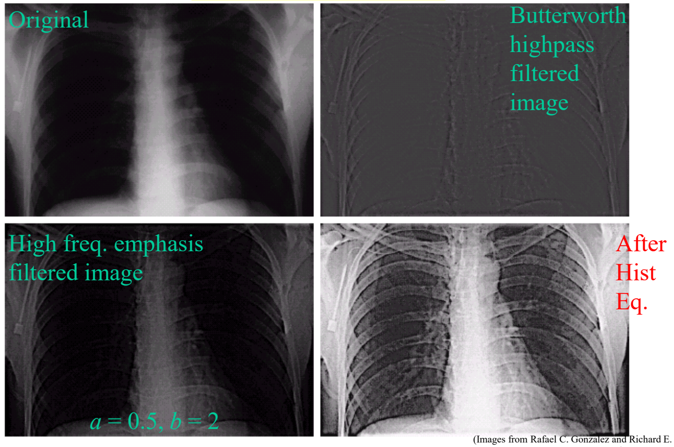
- To *accentuate the contribution* to enhancement made by the *high-frequency components* of an image
- *Multiplying* a highpass filter function by a constant and *adding* an offset so that the zero frequency term is not eliminated by the filter
- **Filter transfer function of *High frequency emphasis***

$H_{hfe}(u,v) = a + bH_{hp}(u,v)$ **($a \ge 0$ and $b>a$)** **(4.4-20)**

✓ Typical values of ***a*** are in the range 0.25 to 0.5 and typical values of ***b*** are in the range 1.5 to 2.0.

High Frequency Emphasis Filtering

$$H_{hfe}(u,v) = a + bH_{hp}(u,v)$$


Original
Butterworth highpass filtered image
High freq. emphasis filtered image
$a = 0.5, b = 2$
After Hist Eq.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Homomorphic Filtering

$$\begin{aligned}
f(x,y) &= i(x,y)r(x,y) \\
\Im\{f(x,y)\} &\neq \Im\{i(x,y)\}\Im\{r(x,y)\} \\
z(x,y) &= \ln f(x,y) \\
&= \ln i(x,y) + \ln r(x,y) \\
\Im\{z(x,y)\} &= \Im\{\ln f(x,y)\} \\
&= \Im\{\ln i(x,y)\} + \Im\{\ln r(x,y)\} \\
Z(u,v) &= F_i(u,v) + F_r(u,v) \\
S(u,v) &= H(u,v)Z(u,v) \\
&= H(u,v)F_i(u,v) + H(u,v)F_r(u,v) \\
s(x,y) &= \Im^{-1}\{S(u,v)\} \\
&= \Im^{-1}\{H(u,v)F_i(u,v)\} + \Im^{-1}\{H(u,v)F_r(u,v)\} \\
&= i'(x,y) + r'(x,y) \\
g(x,y) &= e^{s(x,y)} = e^{i'(x,y)}e^{r'(x,y)} \\
&= i_0(x,y)r_0(x,y)
\end{aligned}$$

Homomorphic Filtering

An image can be expressed as

$$f(x,y) = i(x,y)r(x,y)$$

$i(x,y)$ = illumination component
$r(x,y)$ = reflectance component

simultaneously normalizes the brightness across an image and increases contrast.
Remove multiplicative noise

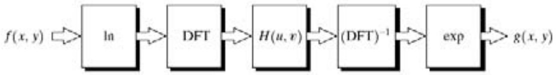


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

Homomorphic Filtering

- Illumination and reflectance are not separable, but their approximate locations in the frequency domain may be located.
- Since illumination and reflectance combine multiplicatively, the components are made additive by taking the logarithm of the image intensity, so that these multiplicative components of the image can be separated linearly in the frequency domain.
- Illumination variations can be thought of as a multiplicative noise, and can be reduced by filtering in the log domain.
- To make the illumination of an image more even, the high-frequency components are increased and low-frequency components are decreased, because the high-frequency components are assumed to represent mostly the reflectance in the scene, whereas the low-frequency components are assumed to represent mostly the illumination in the scene.
- That is, high-pass filtering is used to suppress low frequencies and amplify high frequencies, in the log-intensity domain.

Homomorphic Filter Function H(u,v)

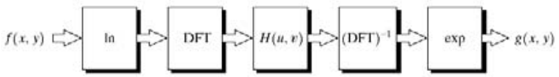


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

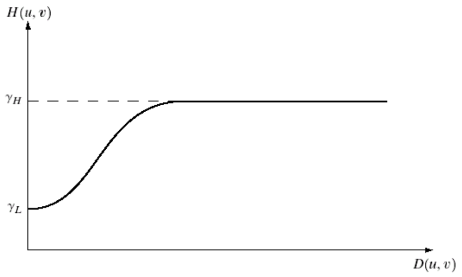
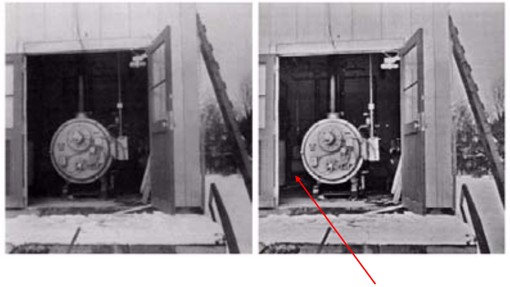


FIGURE 4.32 Cross section of a circularly symmetric filter function. $D(u, v)$ is the distance from the origin of the centered transform.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Homomorphic Filtering



a b

FIGURE 4.33 (a) Original image. (b) Image processed by homomorphic filtering (note details inside shelter). (Stockham.)

$\gamma_L = 0.5, \gamma_H = 2$

More details in the room can be seen!

A *reduction of dynamic range in the brightness*, together with an *increase in contrast*,
brought out the details of objects inside the shelter and balanced the gray levels of the outside walls.
Enhanced image is *sharper.*

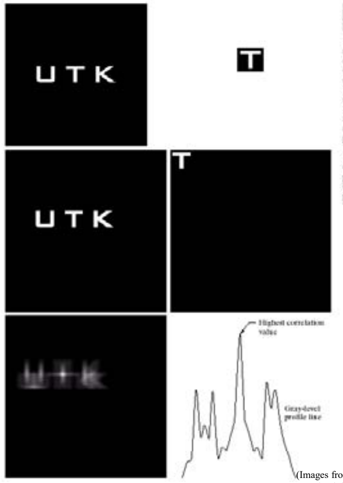
Homomorphic Filtering

Slightly modified from of the Gaussian highpass filter

$$H(u,v) = (\gamma_H - \gamma_L)[1 - e^{-c(D^2(u,v)/D_0^2)}] + \gamma_L$$

- ✓ Constant *c*
  To control the *sharpness of the slope* of the filter function as it transitions between γ_L and γ_H.
- ✓ This type of filter is *similar* to the *high-frequency emphasis filter*.

Correlation Application: Object Detection



a b
c d
e f

FIGURE 4.41 (a) Image. (b) Template. (c) and (d) Padded images. (e) Correlation function displayed as an image. (f) Horizontal profile line through the highest value in (e), showing the point at which the best match took place.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Some Properties of the 2-D Fourier Transform

Translation

Distributivity and Scaling

Rotation

Periodicity and Conjugate Symmetry

Separability

Convolution and Correlation

Translation

$$f(x,y)\exp[j2\pi(u_0x/M+v_0y/N)] \Leftrightarrow F(u-u_0,v-v_0)$$

and

$$f(x-x_0,y-y_0) \Leftrightarrow F(u,v)\exp[-j2\pi(ux_0/M+vy_0/N)]$$

Translation

- The previous equations mean:
  - Multiplying f(x,y) by the indicated exponential term and taking the transform of the product results in a shift of the origin of the frequency plane to the point (u_0,v_0).
  - Multiplying F(u,v) by the exponential term shown and taking the inverse transform moves the origin of the spatial plane to (x_0,y_0).
  - A shift in f(x,y) doesn’t affect the magnitude of its Fourier transform

Distributivity and Scaling

$$\Im\{f_1(x,y)+f_2(x,y)\}=\Im\{f_1(x,y)\}+\Im\{f_2(x,y)\}$$

$$\Im\{f_1(x,y)\cdot f_2(x,y)\}\neq\Im\{f_1(x,y)\}\cdot\Im\{f_2(x,y)\}$$

- Distributive over addition but not over multiplication.

Distributivity and Scaling

- For two scalars a and b,

$$af(x,y) \Leftrightarrow aF(u,v)$$

$$f(ax,by) \Leftrightarrow \frac{1}{|ab|}F(u/a,v/b)$$

Rotation

$$f(r,\theta+\theta_0) \Leftrightarrow F(\omega,\varphi+\theta_0)$$

- Which means that rotating f(x,y) by an angle $_0$ rotates F(u,v) by the same angle (and vice versa).

Rotation

- Polar coordinates:

$$x = r\cos\theta, \quad y = r\sin\theta, \quad u = \omega\cos\varphi, \quad v = \omega\cos\varphi$$

Which means that:

$$f(x,y), F(u,v) \text{ become } f(r,\theta), F(\omega,\varphi)$$

Periodicity & Conjugate Symmetry

- The discrete FT and its inverse are periodic with period N:

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

Periodicity & Conjugate Symmetry

- Although F(u,v) repeats itself for infinitely many values of u and v, only the M,N values of each variable in any one period are required to obtain f(x,y) from F(u,v).
- This means that only one period of the transform is necessary to specify F(u,v) completely in the frequency domain (and similarly f(x,y) in the spatial domain).

Periodicity & Conjugate Symmetry

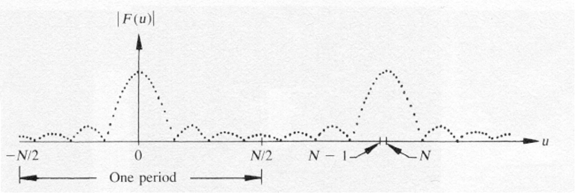
- For real f(x,y), FT also exhibits conjugate symmetry:

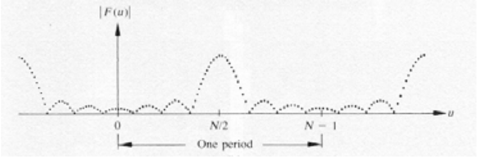
$$F(u,v) = F^*(-u,-v)$$

or

$$|F(u,v)| = |F(-u,-v)|$$

Periodicity & Conjugate Symmetry





(shifted spectrum)
move the origin of the transform to u=N/2.

Periodicity & Conjugate Symmetry

- In essence:

$$F(u) = F(u+N)$$
$$|F(u)| = |F(-u)|$$

- i.e. F(u) has a period of length N and the magnitude of the transform is centered on the origin.

Separability

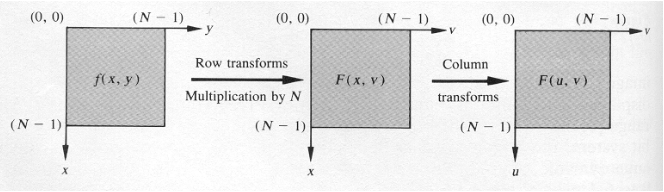
- The discrete FT pair can be expressed in separable forms which (after some manipulations) can be expressed as:

$$F(u,v) = \frac{1}{M}\sum_{x=0}^{M-1} F(x,v)\exp[-j2\pi ux/M]$$

Where: $$F(x,v) = \left[\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)\exp[-j2\pi vy/N]\right]$$

Separability

- The desired result F(u,v) is then obtained by making a transform along each column of F(x,v).



Separability

- For each value of x, the expression inside the brackets is a 1-D transform, with frequency values v=0,1,…,N-1.
- Thus, the 2-D function F(x,v) is obtained by taking a transform along each row of f(x,y) and multiplying the result by N.

Convolution

- Convolution theorem with FT pair:

$$f(x) * g(x) \Leftrightarrow F(u)G(u)$$

$$f(x)g(x) \Leftrightarrow F(u) * G(u)$$

Convolution

- Discrete equivalent:

$$f_e(x) * g_e(x) = \frac{1}{M}\sum_{m=0}^{M-1} f_e(m) g_e(x-m)$$

- Discrete, periodic array of length M.
- x=0,1,2,…,M-1 describes a full period of $f_e(x)*g_e(x)$.
- Summation replaces integration.

Correlation

- Correlation of two functions: f(x) $\circ$ g(x)

$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f^*(\alpha) g(x+\alpha) d\alpha$$

- Types: autocorrelation, cross-correlation
- Used in template matching

Correlation

- Correlation theorem with FT pair:

$$f(x,y) \circ g(x,y) \Leftrightarrow F^*(u,v) G(u,v)$$

$$f^*(x,y) g(x,y) \Leftrightarrow F(u,v) \circ G(u,v)$$

Correlation

- Discrete equivalent:

$$f_e(x) \circ g_e(x) = \frac{1}{M}\sum_{m=0}^{M-1} f_e^*(m) g_e(x+m)$$

For x=0,1,2,…,M-1

Fast Fourier Transform

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)\exp[-j2\pi ux/M]$$

- Number of complex multiplications and additions to implement Fourier Transform: M^2 (M complex multiplications and N-1 additions for each of the N values of u).

Fast Fourier Transform

- The decomposition of FT makes the number of multiplications and additions proportional to M $\log_2$M:
  - Fast Fourier Transform or FFT algorithm.
- E.g. if M=1021 the usual method will require 1000000 operations, while FFT will require 10000.

Homework

1. Implement homomorphic filtering in matlab. Test the code with more than three samples. Discuss the differences.
2. Implement unsharp masking in matlab. Test the code with more than three samples. Discuss the differences.
3. Implement high boost filtering in matlab. Test the code with more than three samples. Discuss the differences.
4. The above should be done with usual matlab code test.