Chapter 4: Image Enhancement in the Frequency Domain

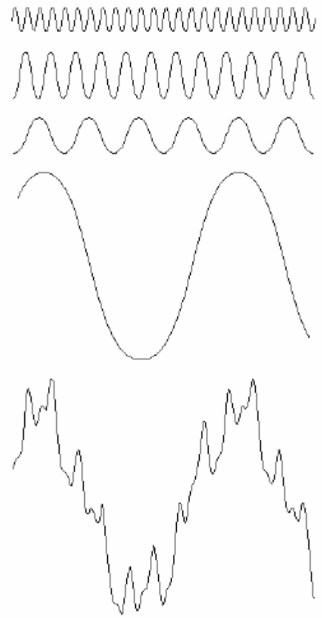


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Slides Credit

Dr. Syed Abd. Rahman Al-Attas

Vibot scholar 2012

Introduction

- Normally Fourier transform will be complex numbers i.e. $F = \alpha + j\beta$

- Can be represented using polar coordinates (magnitude and angle/phase) i.e. $F = |F|e^{j\theta}$ where

$$|F| = \sqrt{\alpha^2 + \beta^2} \quad \text{and } \theta = \tan^{-1}\left(\frac{\beta}{\alpha}\right)$$

Introduction

- Analog signals

Fourier transform

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

Here $\omega = 2\pi f$ and

$e^{-j\omega} = cos\omega - jsin\omega$

Inverse Fourier transform

$$f(t) = \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega$$

Introduction

- Discrete signals

  Discrete Fourier transform (DFT)

$$F(u) = \sum_{n=0}^{N-1} f(n) e^{-\frac{j2\pi nu}{N}} \qquad u = 0,1,\ldots,N-1$$

  Here k is the index for the frequency and N is the length of the discrete signal $f(n)$.

  Inverse Discrete Fourier transform

$$f(n) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) e^{\frac{j2\pi nu}{N}} \qquad n = 0,1,\ldots,N-1$$

2D Discrete Fourier Transform

- Direct extension of the 1D counterpart

$$F(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)e^{-j\frac{2\pi}{N}ux}e^{-j\frac{2\pi}{M}vy} \qquad \begin{aligned} u &= 1,2,\dots,N-1 \\ v &= 1,2,\dots,M-1 \end{aligned}$$

$$f(x,y)=\frac{1}{NM}\sum_{u=0}^{N-1}\sum_{v=0}^{M-1} F(u,v)e^{j\frac{2\pi}{N}ux}e^{j\frac{2\pi}{M}vy} \qquad \begin{aligned} x &= 1,2,\dots,N-1 \\ y &= 1,2,\dots,M-1 \end{aligned}$$

2D Discrete Fourier Transform

- Since 2D DFT in general is complex, we can express in polar form

$$F(u,v) = |F(u,v)| e^{j\phi(u,v)}$$

where

$$|F(u,v)| = \left(R^2(u,v) + I^2(u,v)\right)^{1/2}$$

Magnitude spectrum

and

$$\phi(u,v) = \arctan\left(\frac{I(u,v)}{R(u,v)}\right)$$

Phase spectrum

2D Discrete Fourier Transform

- Another important term is the Power Spectrum

$$P(u,v) = |F(u,v)|^2$$

From the 2D DFT we have

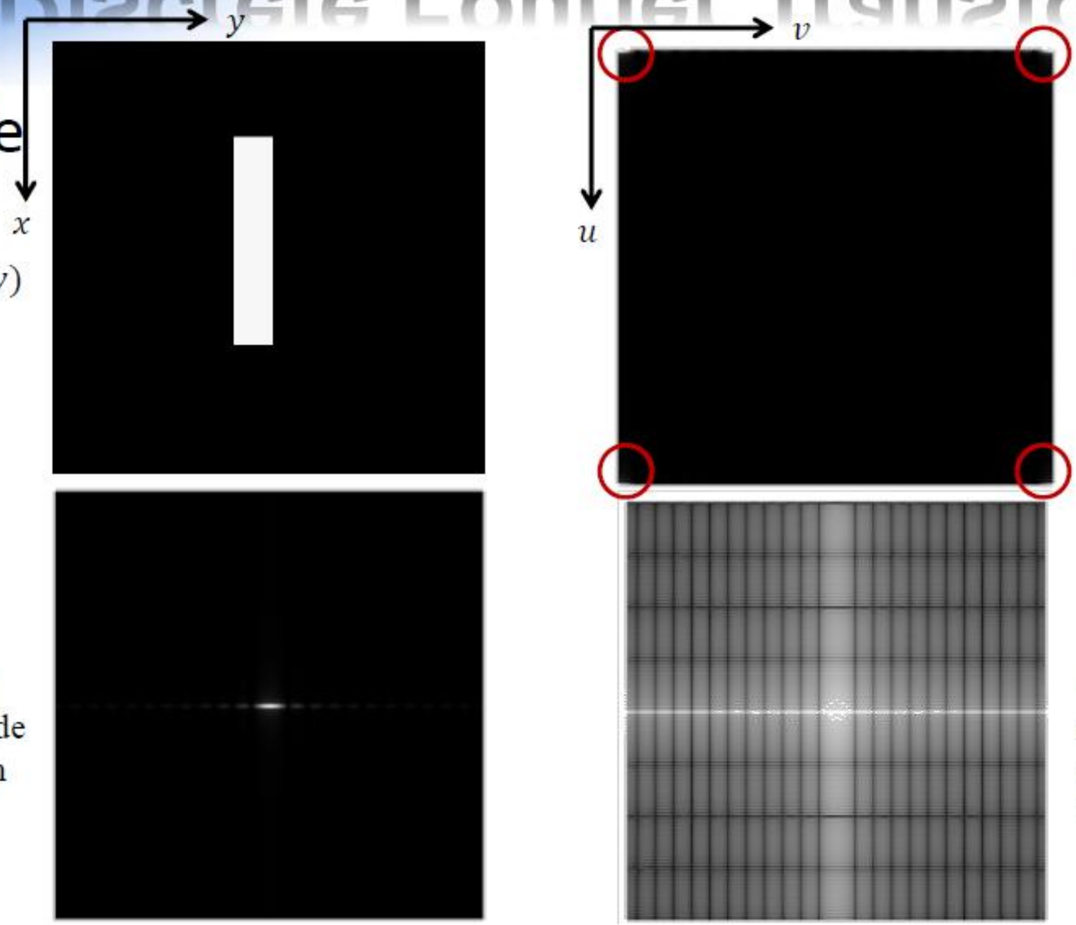
$$F(0,0) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y) e^{-j\frac{2\pi}{N}0x} e^{-j\frac{2\pi}{M}0y}$$

$$F(0,0) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y) \qquad \text{Maximum value}$$

$$\frac{1}{MN} F(0,0) = \frac{1}{MN}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y) = \overline{f}(x,y) \qquad \text{Average value}$$

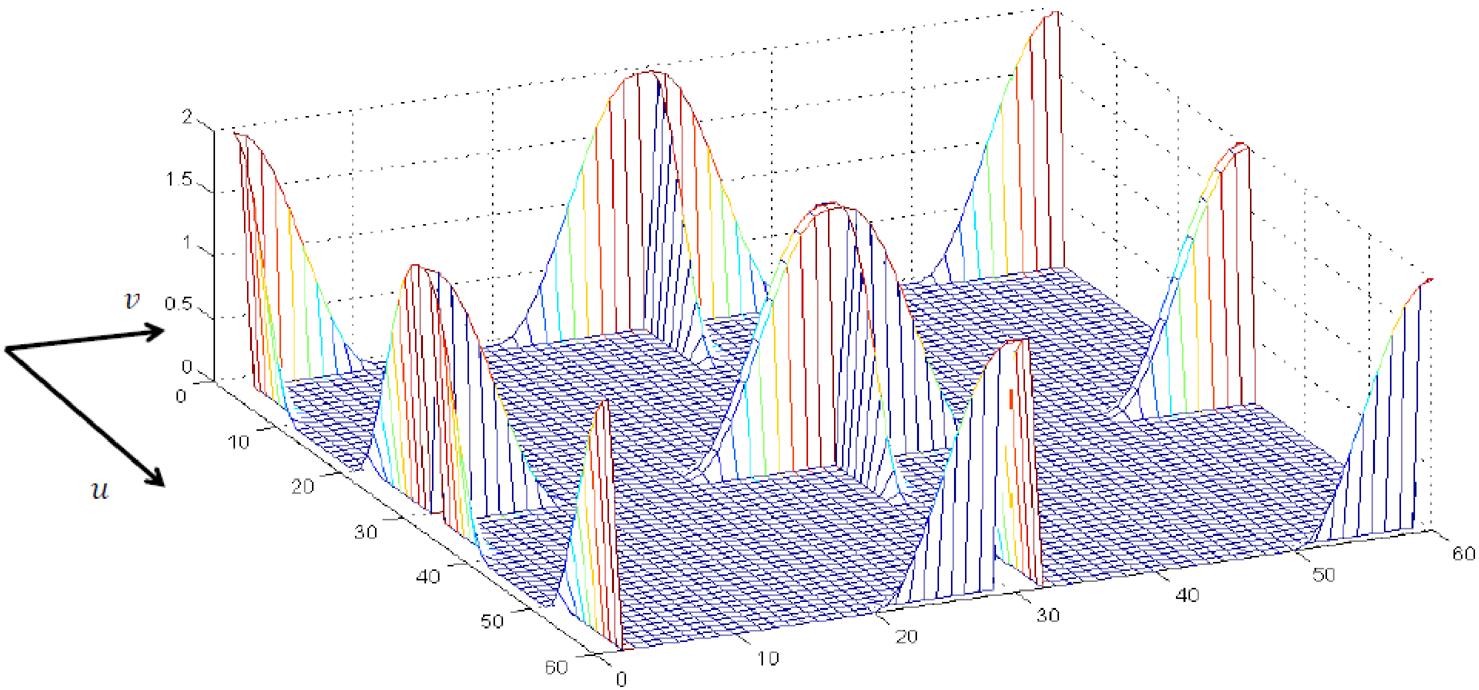
2D Discrete Fourier Transform

Example

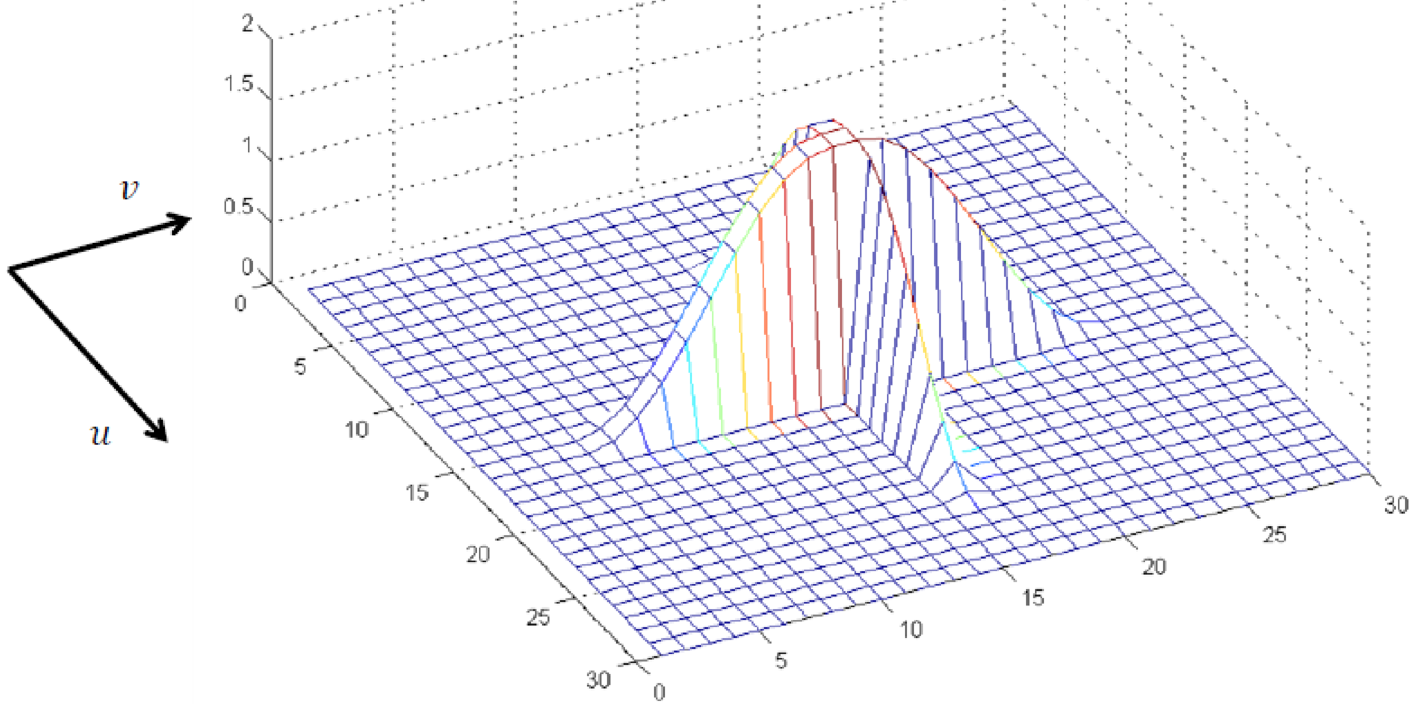


2D Discrete Fourier Transform

- Result of FT is periodic



- Much easier to look at the spectrum if we shift the centre.



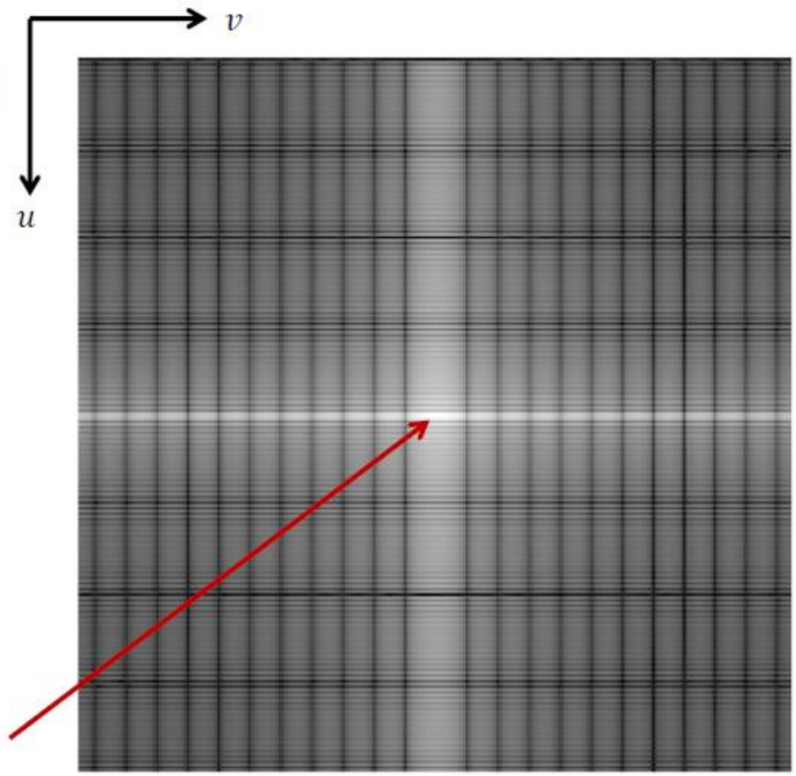
- How to do this?

$$F(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)(-1)^{x+y} e^{-j\frac{2\pi}{N}ux} e^{-j\frac{2\pi}{M}vy}$$

Shifting by N/2 and M/2 in frequency domain

- Fortunately, Matlab has a simple command for achieving this.

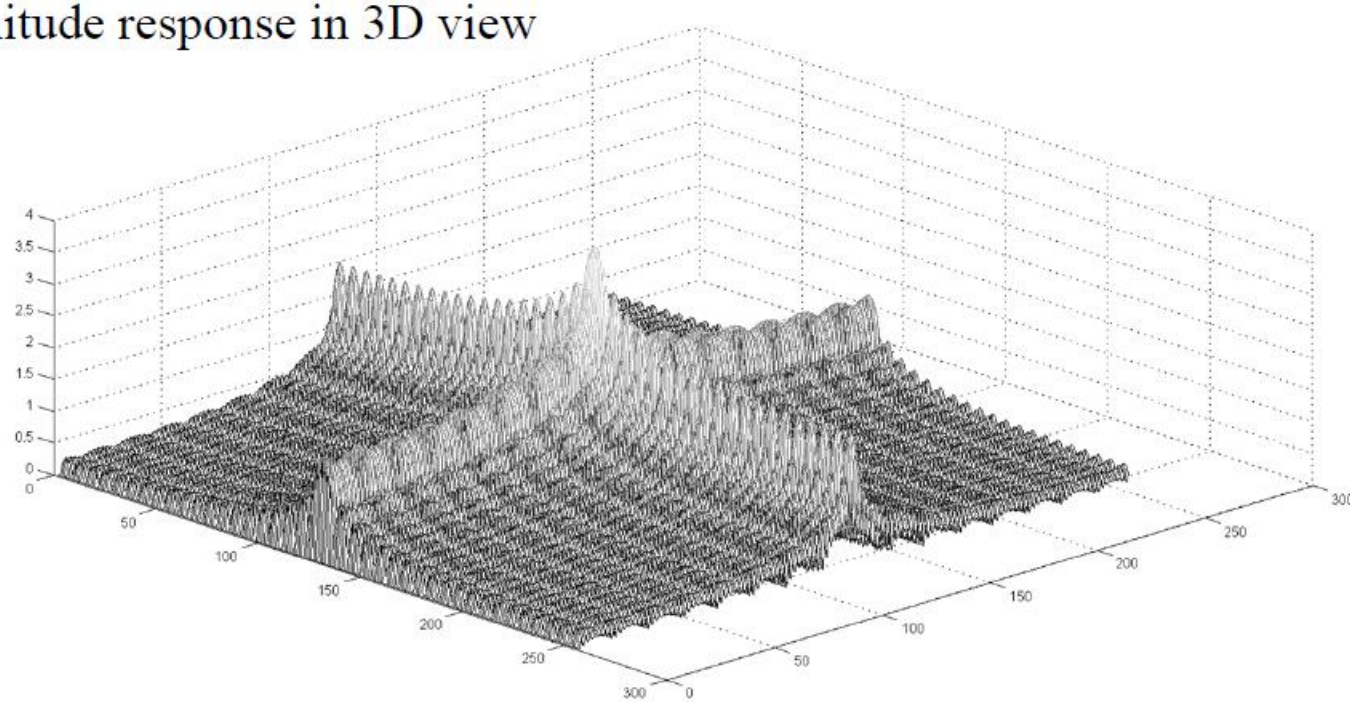
- Magnitude spectrum tells the amplitude of the sinusoids that form the image
- For any given frequency large amplitude (bright intensity) indicates high influence of that frequency while small amplitude indicates the opposite.



$$\left(\frac{M}{2},\frac{N}{2}\right)$$

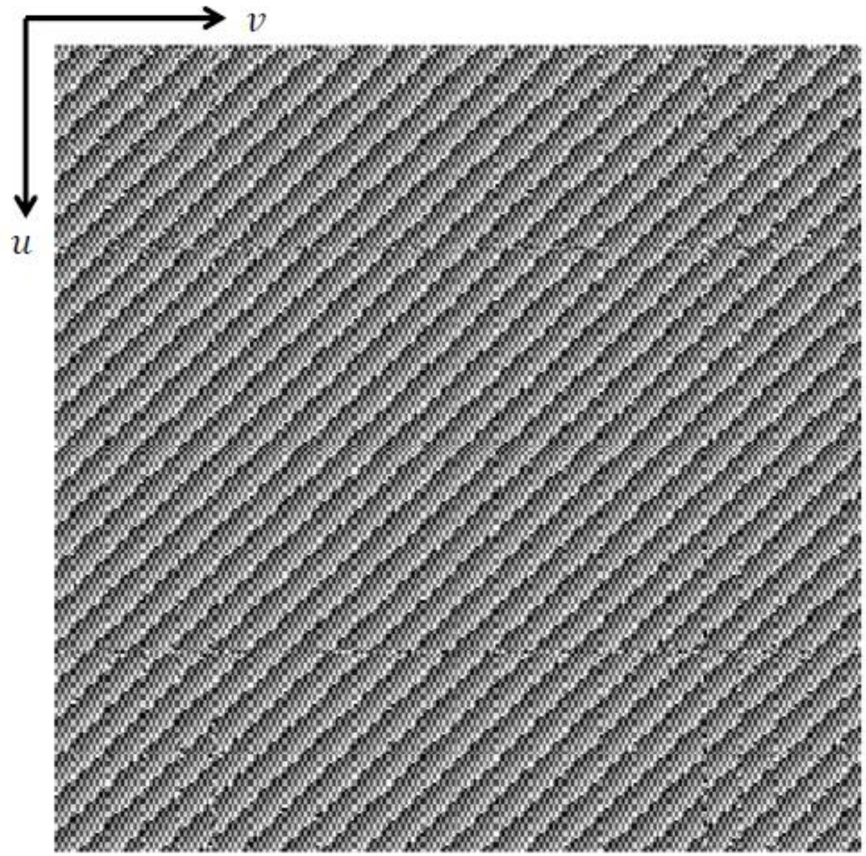
Magnitude spectrum

Magnitude response in 3D view



Phase tells the displacement of the various sinusoids with respect to their origin.

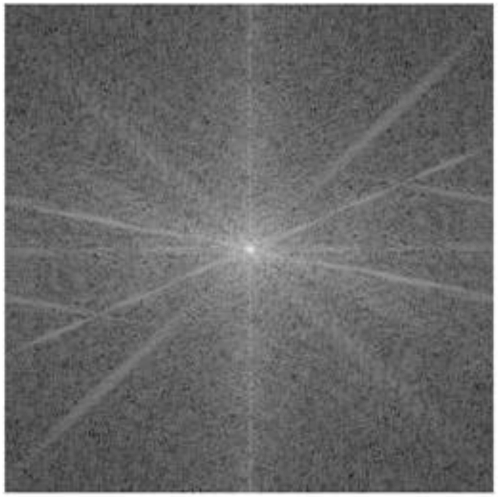
Not very intuitive but important

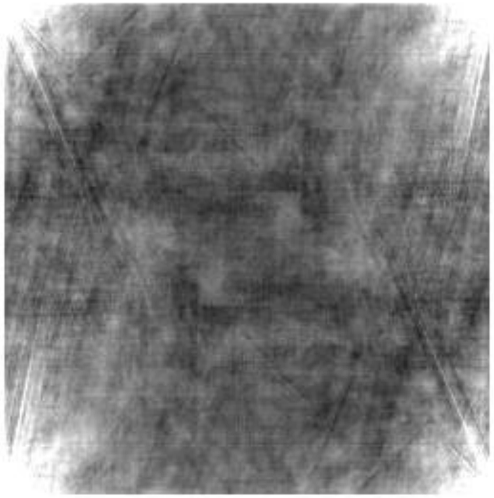


Phase spectrum

- Effect of magnitude only (setting phase = 0).

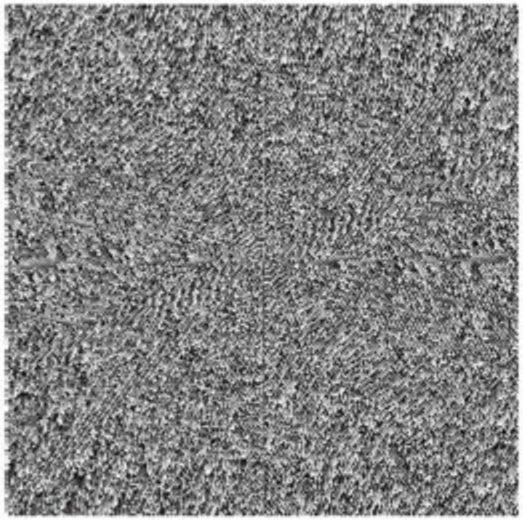

Original Image


Magnitude response

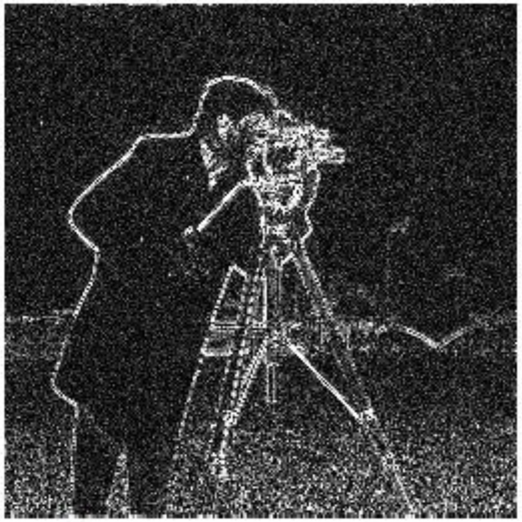

Reconstruction

- Effect of phase only - set amplitude = constant (80)

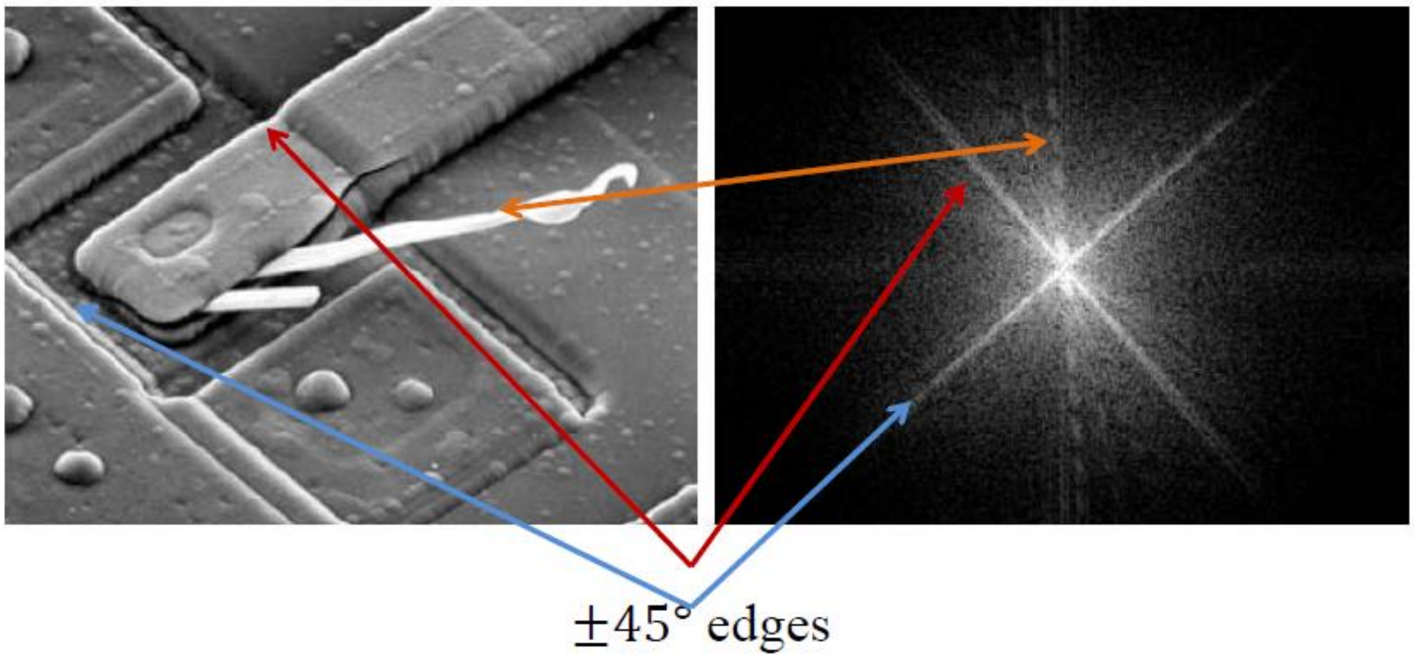
Original Image



Phase response



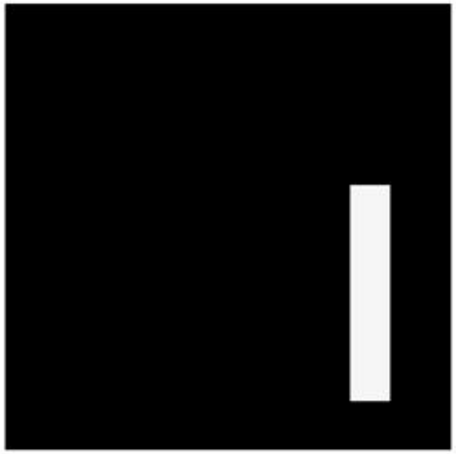
Reconstruction



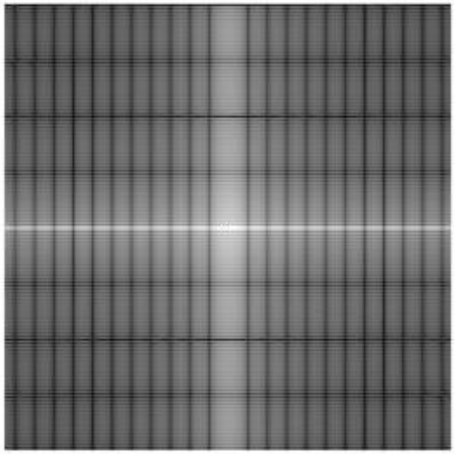
Spatial domain – with respect to horizontal line
Frequency domain – with respect to vertical line

- Translation

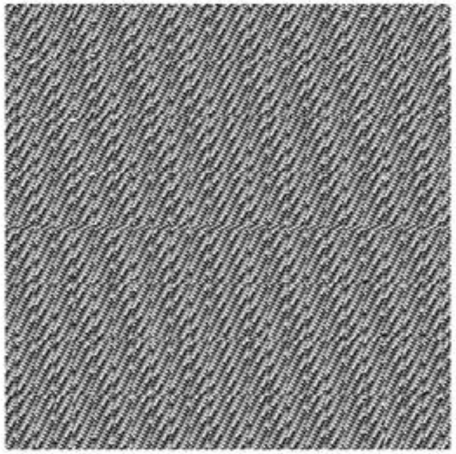
$$f(x-x_0,y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi\left(\frac{x_0u}{M}+\frac{y_0v}{N}\right)}$$



Translated bar



Magnitude spectrum



Phase spectrum

- Rotation
- If we write $f(x,y)$ in polar form i.e.

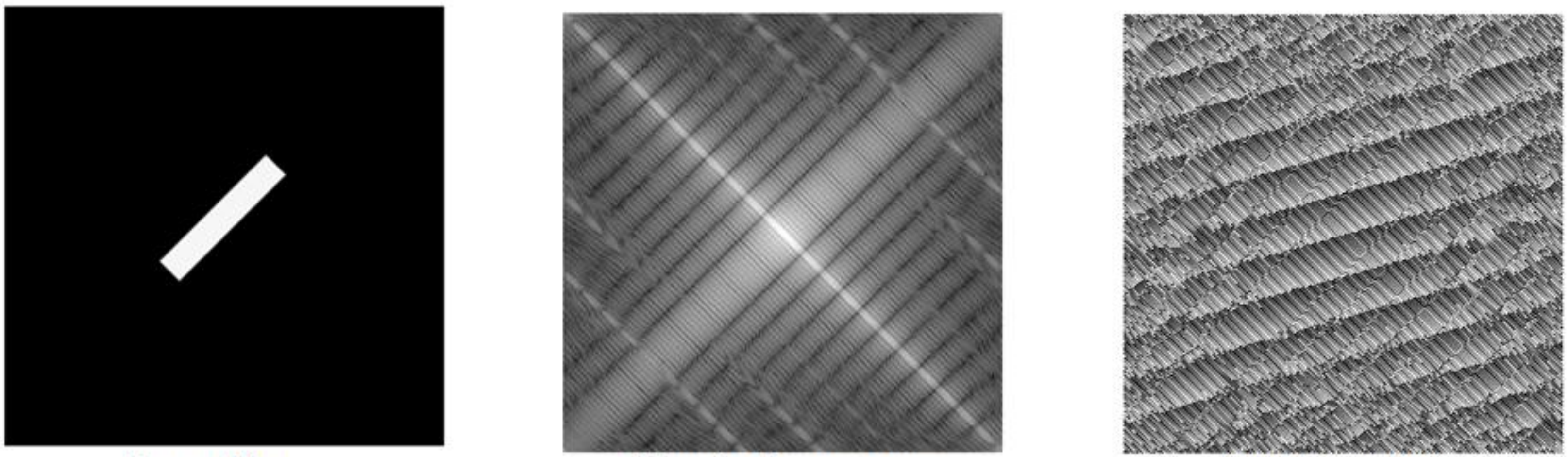
$$x = r\cos\theta \qquad u = w\cos\varphi$$

$$y = r\sin\theta \qquad v = w\sin\varphi$$

then $f(x,y)$ and $F(u,v)$ become $f(r,\theta)$ and $F(w,\varphi)$

thus

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



Rotated bar Magnitude spectrum Phase spectrum

- Amplitude spectrum - <u>How Much</u> of each sinusoid component is present.
- Phase spectrum - <u>Where</u> each of the sinusoidal components resides within the image.

- Addition

$$f(x, y) + g(x, y) \Leftrightarrow F(u, v) + G(u, v)$$

- Linear

$$\alpha f(x, y) + \beta g(x, y) \Leftrightarrow \alpha F(u, v) + \beta G(u, v)$$

- Convolution

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

maps to multiplication in the frequency domain

Recall

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

Filtering in Frequency Domain

- Hence, in the case $h(x, y)$ is given in spatial domain we can perform filtering operation by first transform both $f(x, y)$ and $h(x, y)$ into $F(u, v)$ and $H(u, v)$ respectively. Multiply the two to obtain $G(u, v)$.
$$G(u, v) = F(u, v)H(u, v)$$
- Finally take the inverse DFT of $G(u, v)$ to obtain $g(x, y)$.
- However, because of periodicity when taking DFT we got to avoid <u>wraparound error</u> or <u>aliasing.</u>

- zeropad both $f(x,y)$ and $h(x,y)$ so that their size is now $P \times Q$ where P and Q must satisfy the following

$$(P,Q) \geq \underbrace{(M,N)}_{size\ of\ f} + \underbrace{(C,D)}_{size\ of\ h} - (1,1)$$

thus choose

$$P = M + C - 1 \text{ and } Q = N + D - 1$$

- Note: to center the DFT, both zeropaded $f(x,y)$ and $h(x,y)$ must be multiplied by $(-1)^{x+y}$. Likewise $g(x,y)$ must also be multiplied by $(-1)^{x+y}$.

- Zeropadding $f(x, y)$ is quite straight forward

$a_{0,0}$	$a_{0,1}$	$\cdots$	$a_{0,N-1}$
$a_{1,0}$	$a_{1,1}$	$\cdots$	$a_{1,N-1}$
$a_{2,0}$	$a_{2,1}$	$\cdots$	$a_{2,N-1}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$a_{M-1,0}$	$a_{M-1,1}$	$\cdots$	$a_{M-1,N-1}$

							$Q-1$
	$a_{0,0}$	$a_{0,1}$	$\cdots$	$a_{0,N-1}$	0	$\cdots$	0
	$a_{1,0}$	$a_{1,1}$	$\cdots$	$a_{1,N-1}$	0	$\cdots$	0
	$a_{2,0}$	$a_{2,1}$	$\cdots$	$a_{2,N-1}$	0	$\cdots$	0
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$a_{M-1,0}$	$a_{M-1,1}$	$\cdots$	$a_{M-1,N-1}$	0	$\cdots$	0
	0	0	$\cdots$	0	0	$\cdots$	0
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$P-1$	0	0	$\cdots$	0	0	$\cdots$	0

- How to zeropad $h(x, y)$.
- Consider only filters of type

$$\begin{cases} symmetric: & h(x,y) = h(C-x, D-y) \\ antisymmetric: & h(x,y) = -h(C-x, D-y) \end{cases}$$

1	2	1
2	4	2
1	2	1

Symmetric

center

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

Anti-symmetric

e.g for 6x6

0	0	0	0	0	0
0	0	0	0	0	0
0	0	1	2	1	0
0	0	2	4	2	0
0	0	1	2	1	0
0	0	0	0	0	0

Symmetric

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

Anti-symmetric

- So for $P \times Q$ symmetric/anti-symmetric filter the centre of the filter will now be at location $\left(\frac{P}{2}, \frac{Q}{2}\right)$.
- So choose even integer values for P and Q.
- Note: from the obtained $g(x, y)$ simply choose the first $M \times N$ block.

Basic steps for filtering in the frequency domain

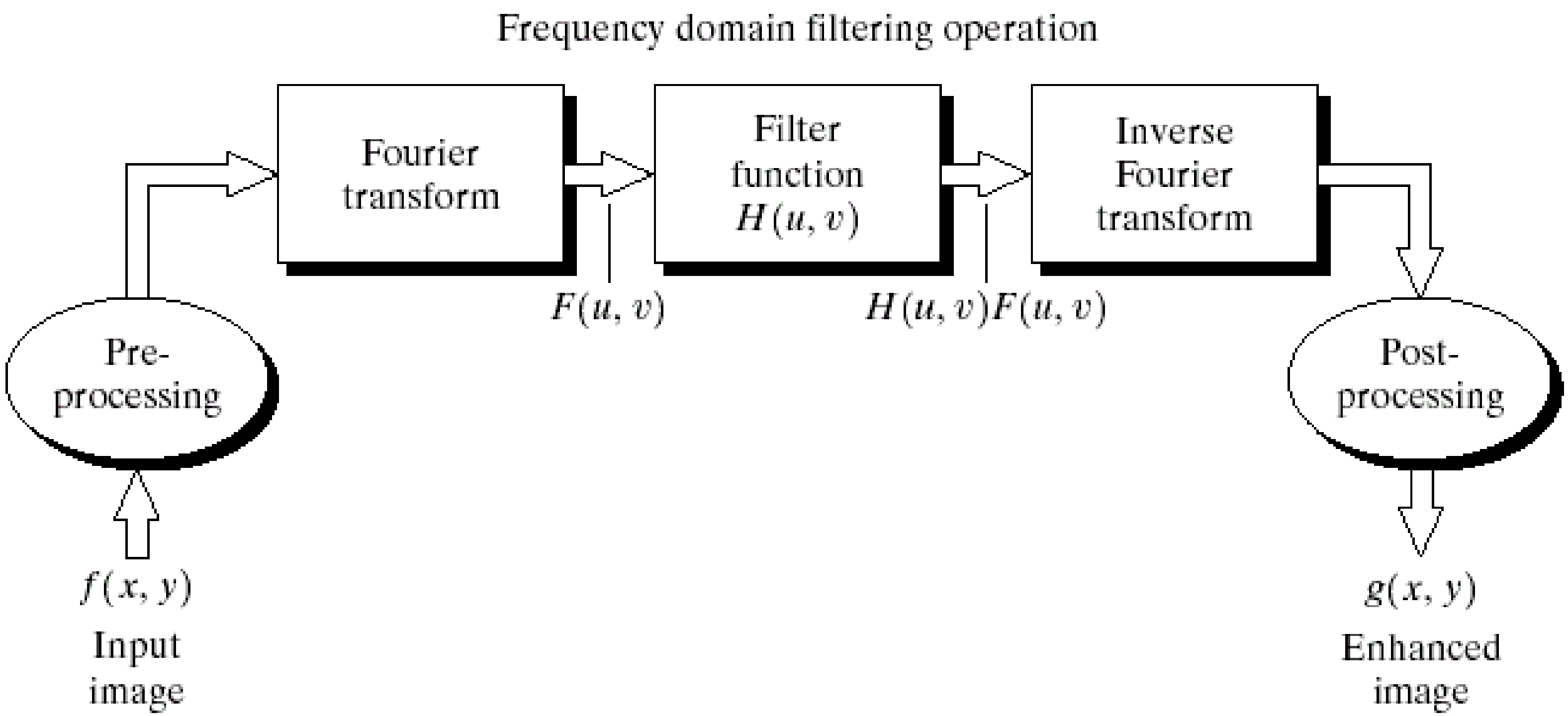


FIGURE 4.5 Basic steps for filtering in the frequency domain.

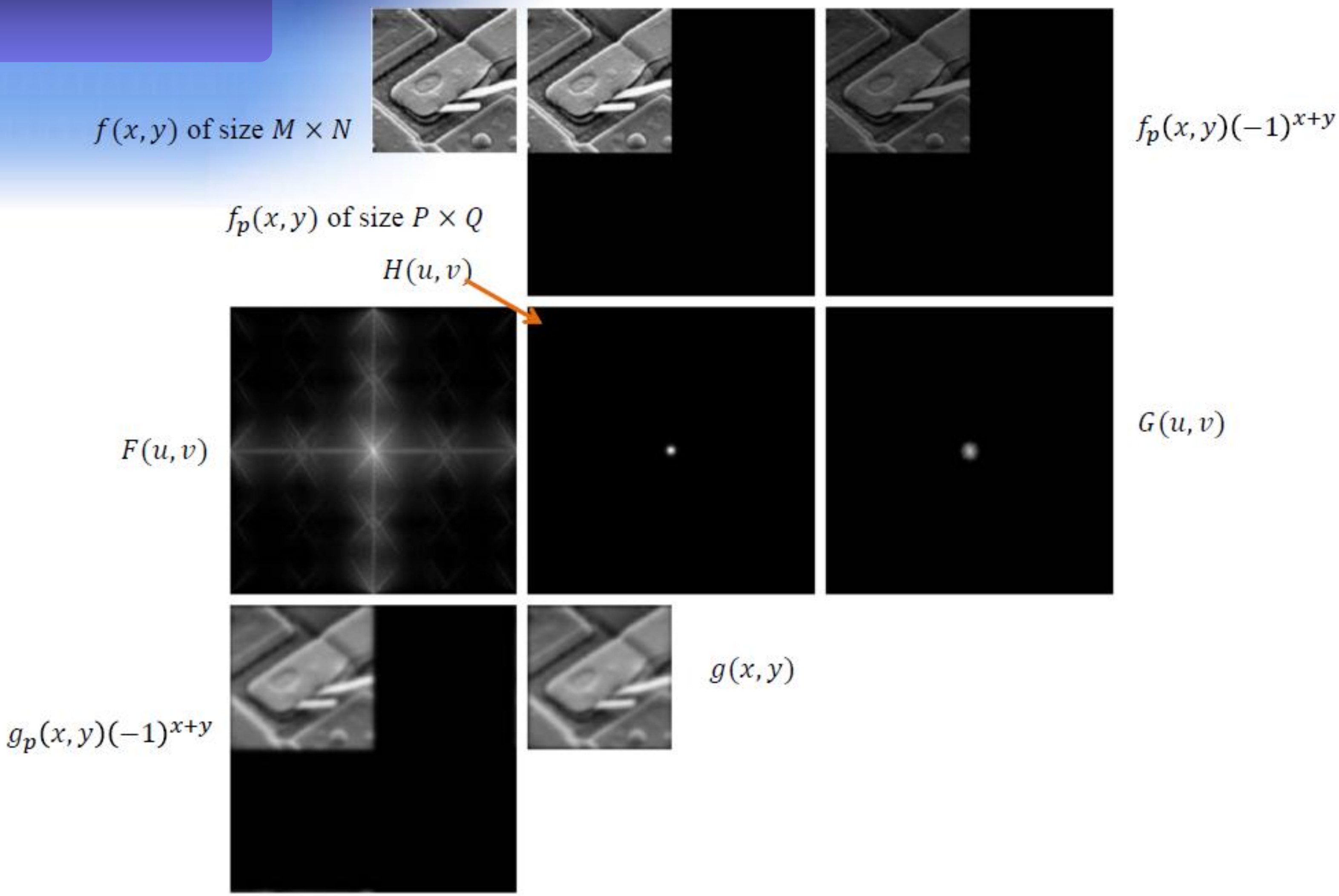
Frequency Domain Filtering

- If the filter is directly design in the frequency domain i.e. $H(u, v)$ then the following steps can be used

1. Given $f(x, y)$ of size $M \times N$, pad it with zeros to obtain $f_p(x, y)$ whose size is $P \times Q$. In this case set $P = 2M$ and $Q = 2N$.
2. To centre the transform at $\left(\frac{P}{2}, \frac{Q}{2}\right)$ perform $f_p(x, y)(-1)^{x+y}$.
3. Compute the DFT $F(u, v)$ of the image.
4. Generate the filter transfer function $H(u, v)$ of the size $P \times Q$ and centred at $\left(\frac{P}{2}, \frac{Q}{2}\right)$.

Frequency Domain Filtering

5. Perform the array multiplication to get the output i.e. $G(u, v) = F(u, v)H(u, v)$.
6. Perform the IDFT to get $g_p(x, y)$ and reverse the shift by multiplying it with $(-1)^{x+y}$. Note: $g_p(x, y)$ is of the size $P \times Q$.
7. Finally extract $M \times N$ array from the top left of $g_p(x, y)$ in order to get $g(x, y)$ – the desired output. Note: take only the real part.


$f(x,y)$ of size $M \times N$
$f_p(x,y)$ of size $P \times Q$
$f_p(x,y)(-1)^{x+y}$
$H(u,v)$
$F(u,v)$
$G(u,v)$
$g_p(x,y)(-1)^{x+y}$
$g(x,y)$

Smoothing (Lowpass Filter)

- Retain low frequency component of the image – uniform or "constant" pixel areas
- Removes or attenuate high frequency component – details of an object such as edges and boundaries
- In the frequency domain

  $$G(u,v) = F(u,v)H(u,v)$$ - array multiplication
- $F(u,v)$ is the 2-D F.T. of the image
- $H(u,v)$ is the filter transfer function

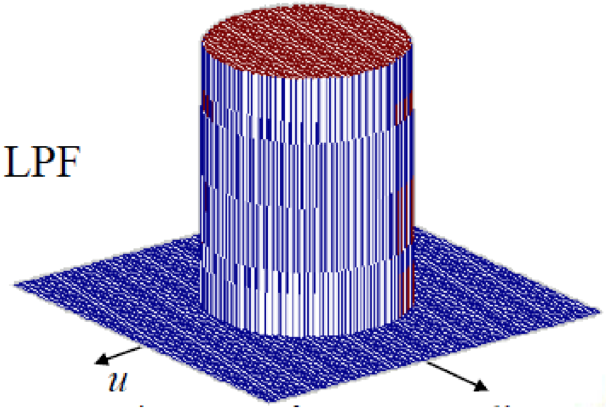
Smoothing (Lowpass Filter)

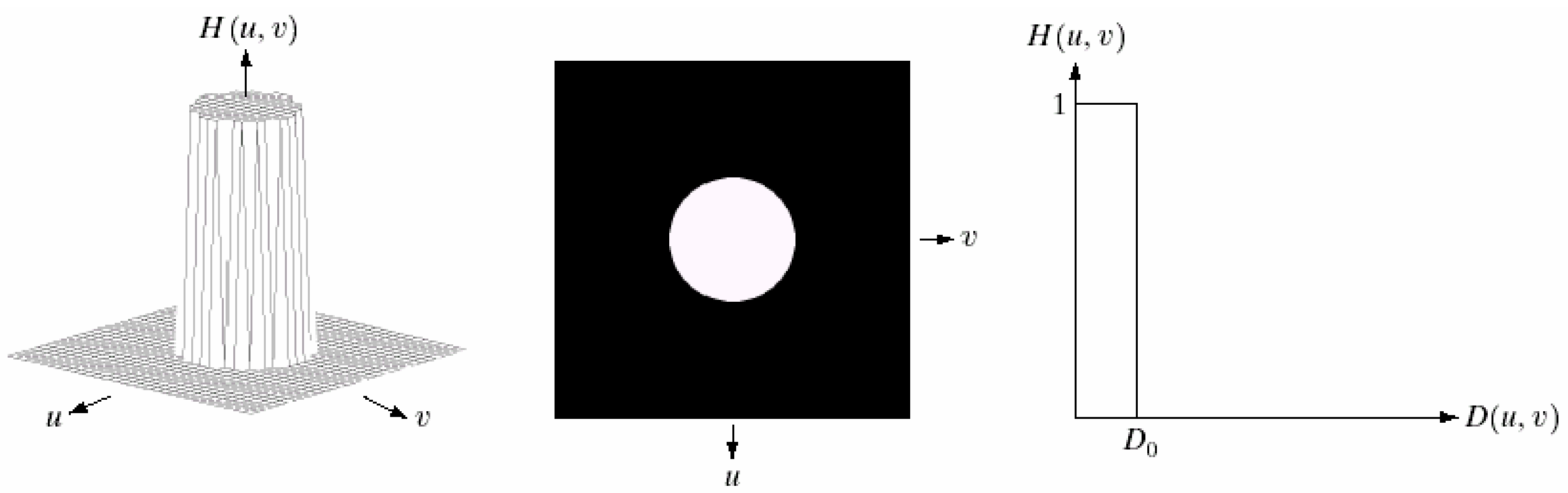
Ideal Lowpass Filter

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

- $D(u,v)$ is the filter characteristic describing the shape of the filter: circular, ellipse etc.

Circular-shape LPF





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.

Smoothing (Lowpass Filter)

- For circular-shaped and centred at $\left(\frac{P}{2}, \frac{Q}{2}\right)$:

$$D(u,v) = \sqrt{\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2}$$

- D_0 is the filter radius (cutoff point) from the centre of the filter.
- Hence, $G(u,v) = \begin{cases} F(u,v), & H(u,v) = 1 \\ 0, & H(u,v) = 0 \end{cases}$
- Inverse transform will yield $g(x,y)$ – the desired image.

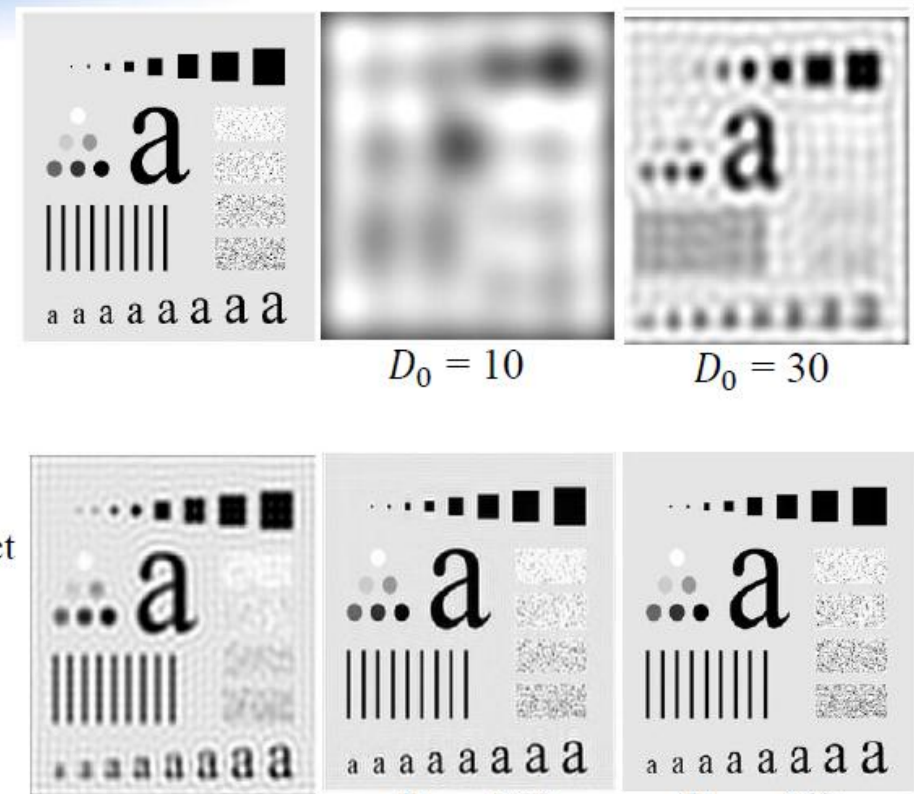
Smoothing (Lowpass Filter)

- Ideal filters – sharp cutoff (transition) from passband to stopband.
- Cannot be implemented with hardware components.
- Analysis – effect on performing the filter for various D_0 values.

Smoothing (Lowpass Filter)

Original
688 x 688

Ringing artifact
due to ideal
characteristic
of the filter



$D_0 = 10$ $D_0 = 30$

$D_0 = 60$ $D_0 = 160$ $D_0 = 460$

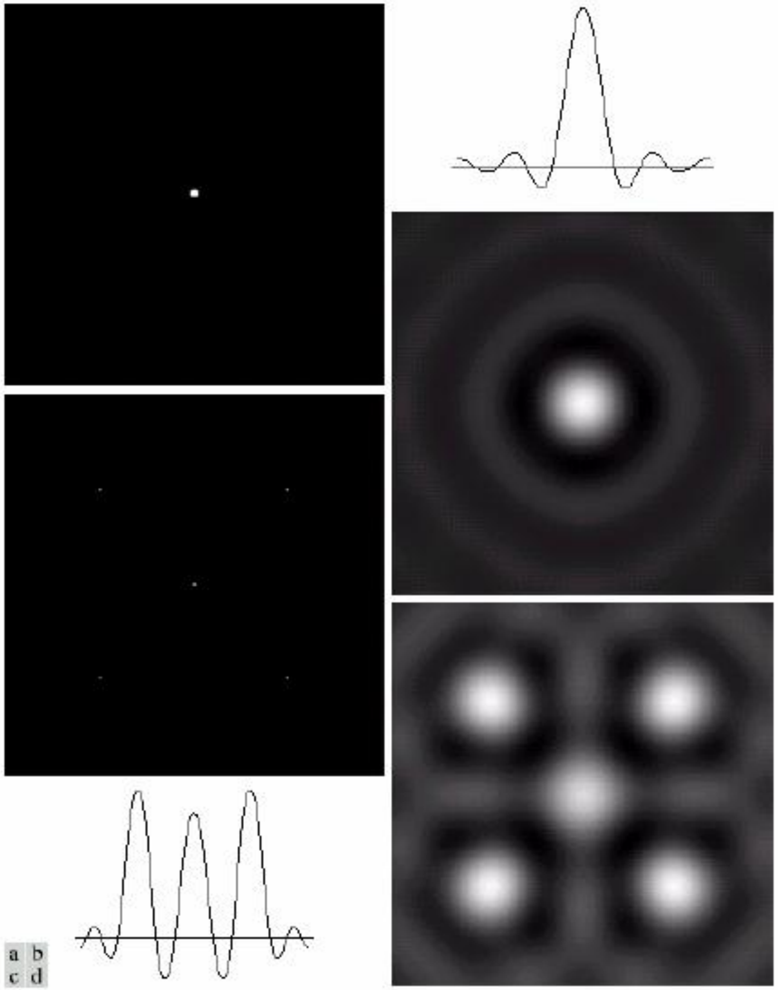


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

Smoothing (Lowpass Filter)

Butterworth Lowpass Filter

- General equation

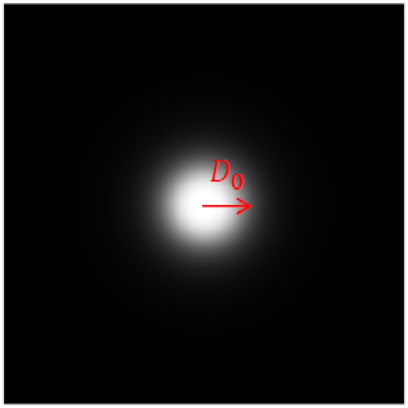
$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}}$$

- n – order of the filter.
- D_o – cutoff frequency locus (distance from the origin)
- $D(u,v)$ – filter characteristic as before

Smoothing (Lowpass Filter)

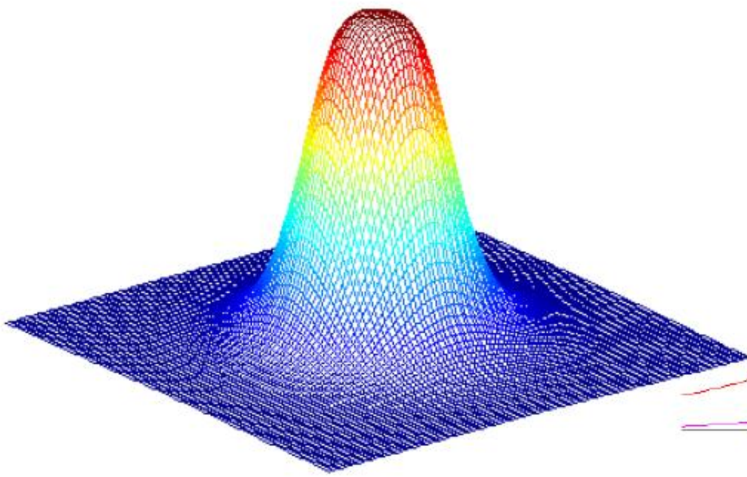
- $D(u, v) = (u^2 + v^2)^{\frac{1}{2}} \leftarrow$ circular shape
- In our case for shifted (centred) filter

$$D(u, v) = \left(\left[u - \frac{P}{2} \right]^2 + \left[v - \frac{Q}{2} \right]^2 \right)^{\frac{1}{2}}$$

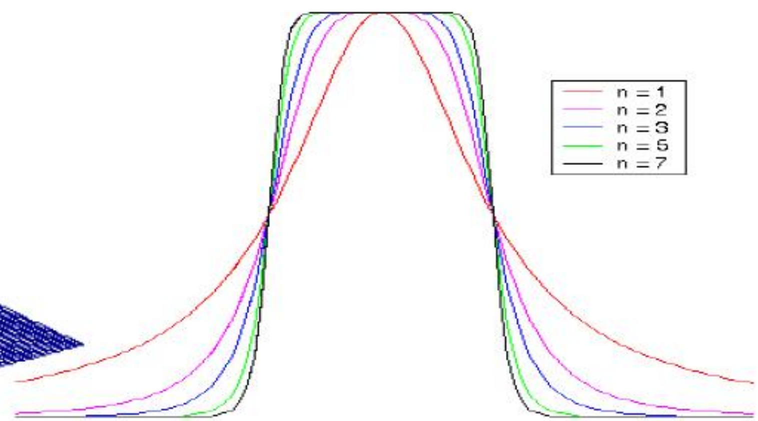


Smoothing (Lowpass Filter)

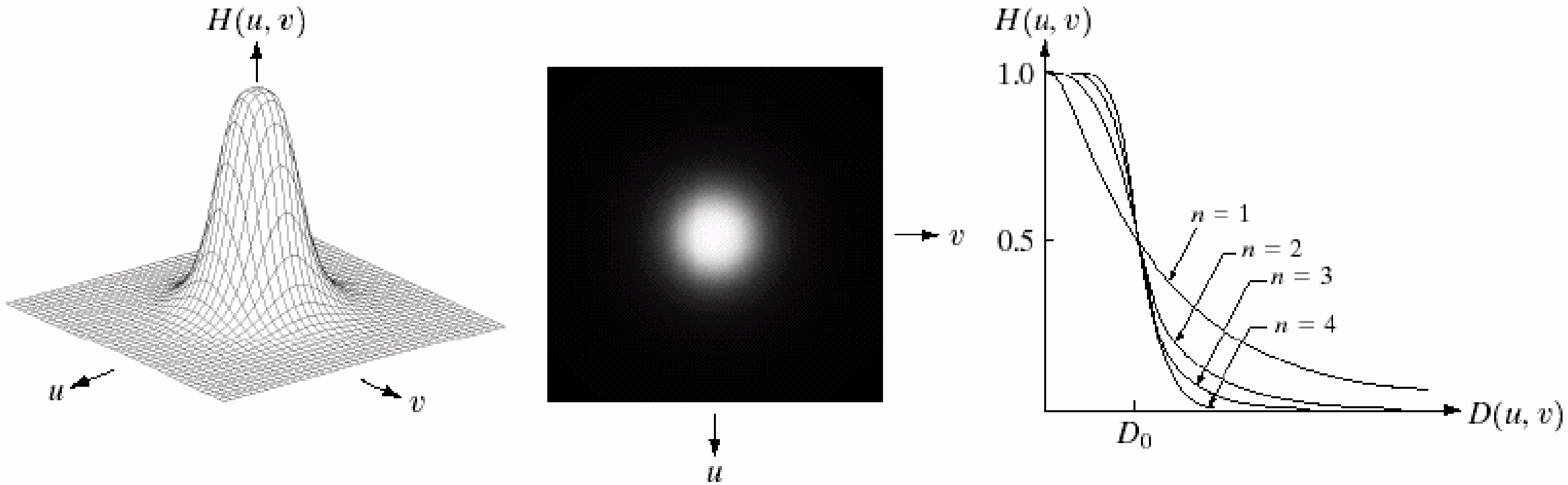
Perspective and cross section profile of BLPF



Perspective Plot $n = 2$



Radial cross section



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.

Smoothing (Lowpass Filter)

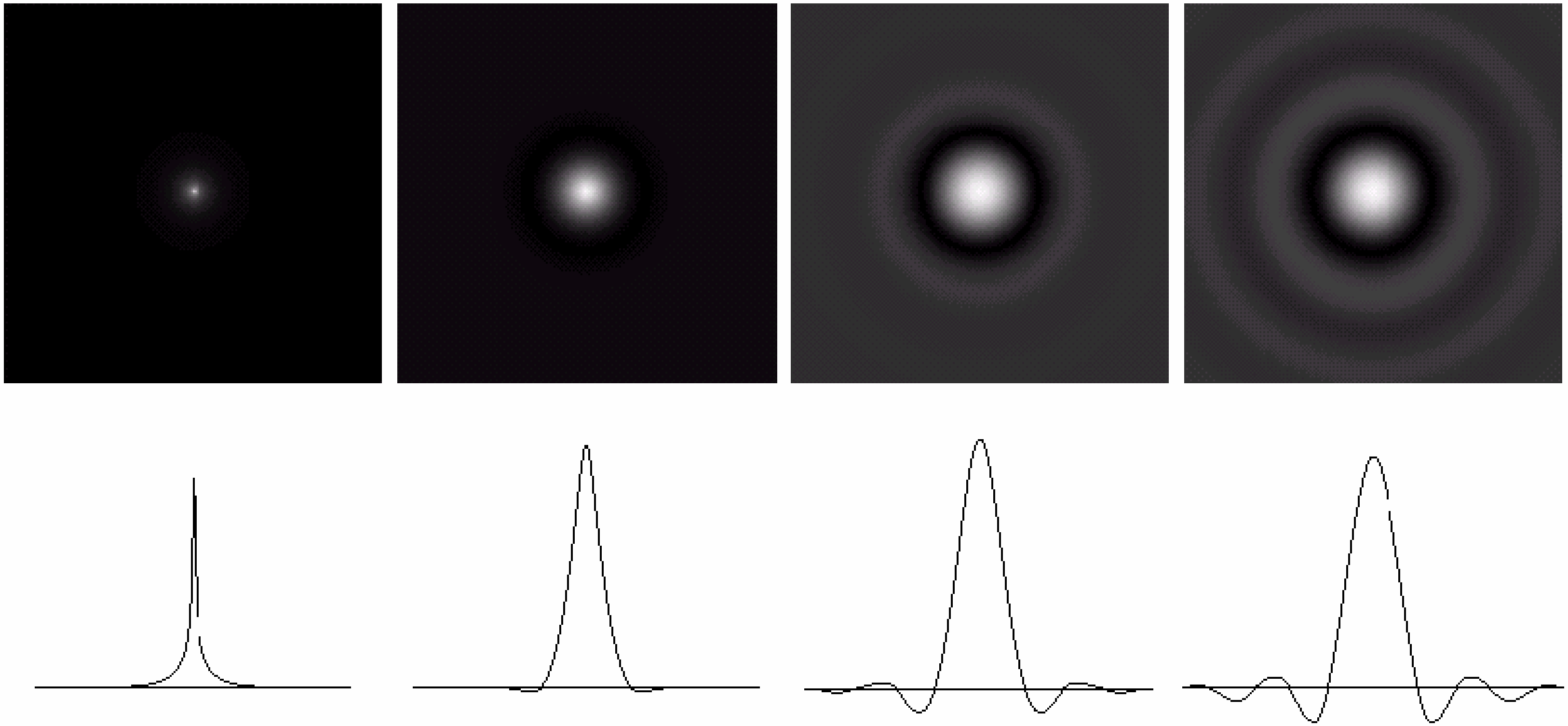
- Effect of applying different values of D_0 (5 values) with $n = 2$.

$D_0 = 10$ $D_0 = 30$

Smoothing (Lowpass Filter)

$D_0 = 60$ $D_0 = 160$ $D_0 = 460$

Ringing artifact is not visible



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.

Smoothing (Lowpass Filter)

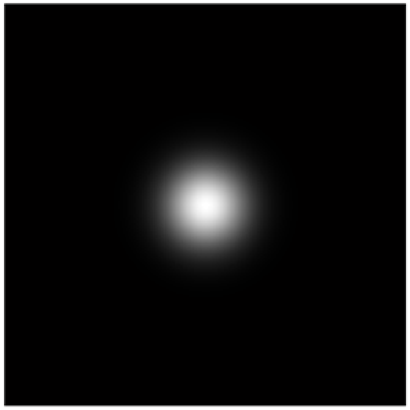
Gaussian Lowpass Filter

- General equation

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

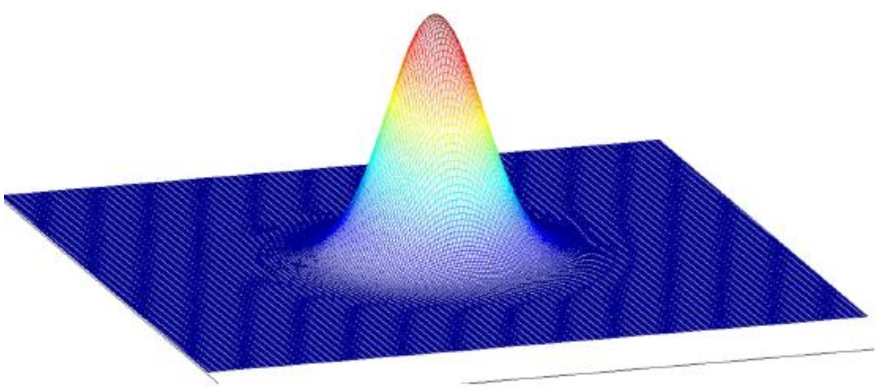
- Less control parameter – no filter order
- Less smoothing effect
- Centred version

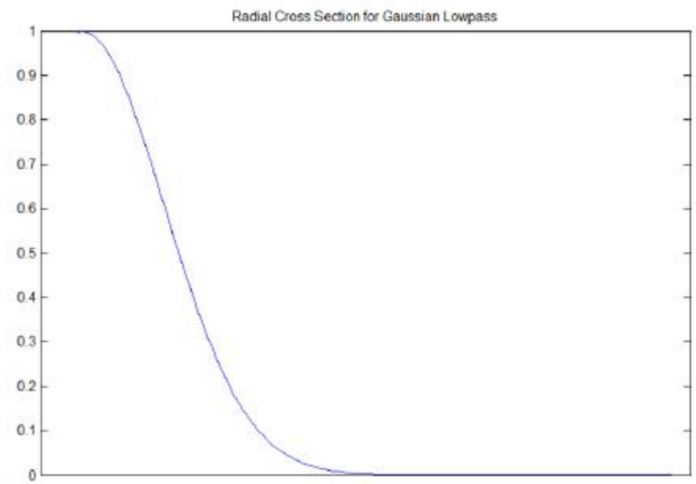
$$H(u,v) = e^{-\frac{\left(u-\frac{P}{2}\right)^2+\left(v-\frac{Q}{2}\right)^2}{2D_0^2}}$$

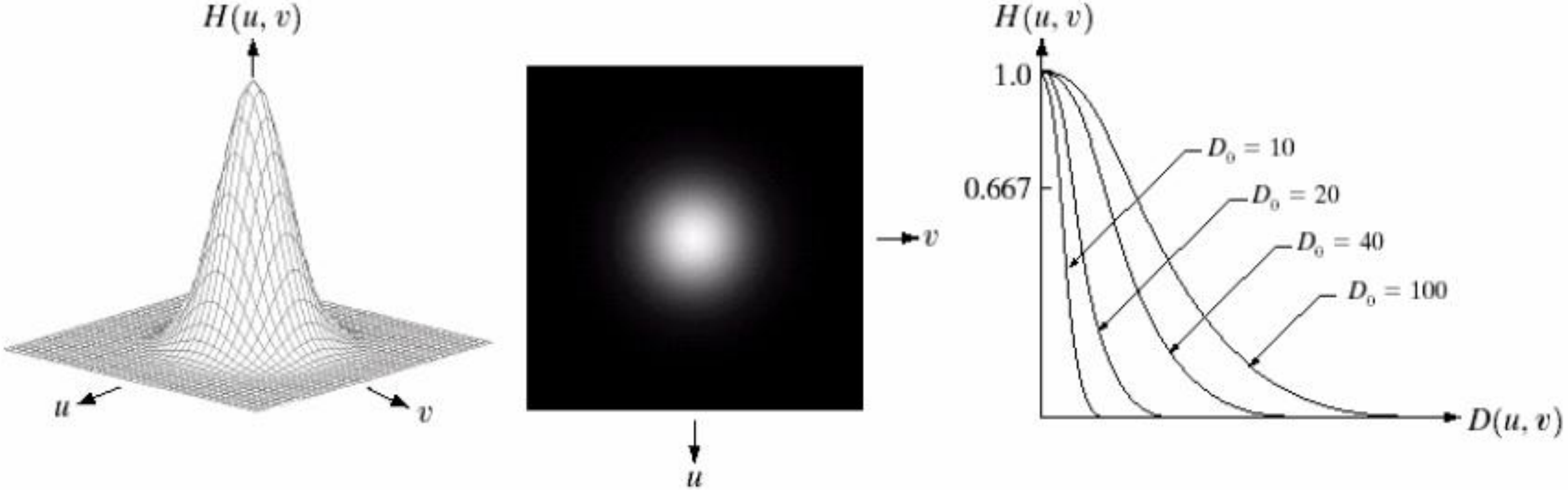


Smoothing (Lowpass Filter)

- Perspective and cross section profile of GLPF







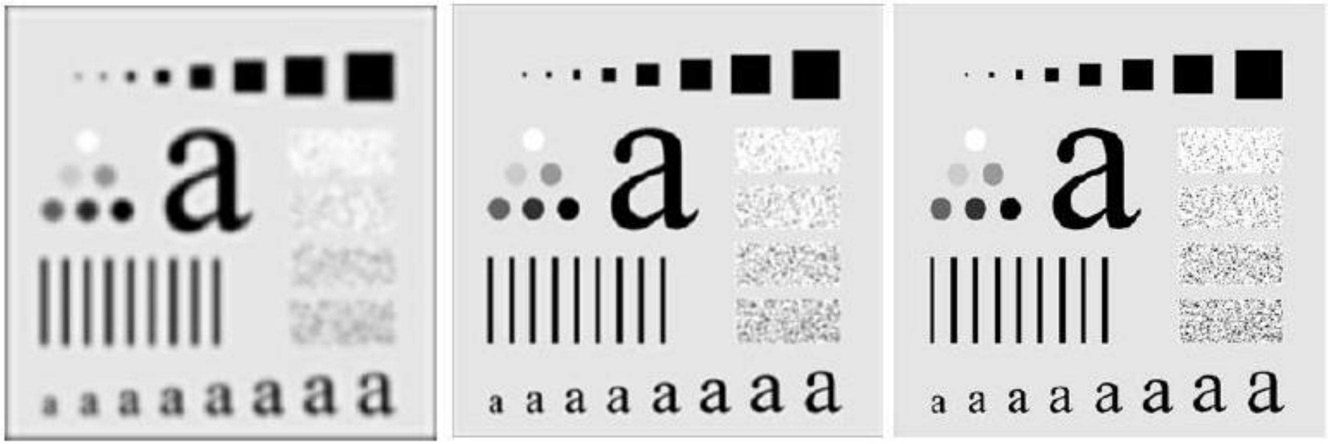
a b c

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.

Smoothing (Lowpass Filter)

$D_0 = 10$ $D_0 = 30$

Smoothing (Lowpass Filter)



$D_0 = 60$ $D_0 = 160$ $D_0 = 460$

Ringing artifact is not visible

Image is somewhat less blurred compared to that of BLPF

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).

Sharpening (Highpass) Filtering

- Image sharpening can be achieved by a highpass filtering process, which attenuates the low-frequency components without disturbing high-frequency information.

- Zero-phase-shift filters: radially symmetric and completely specified by a cross section.

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

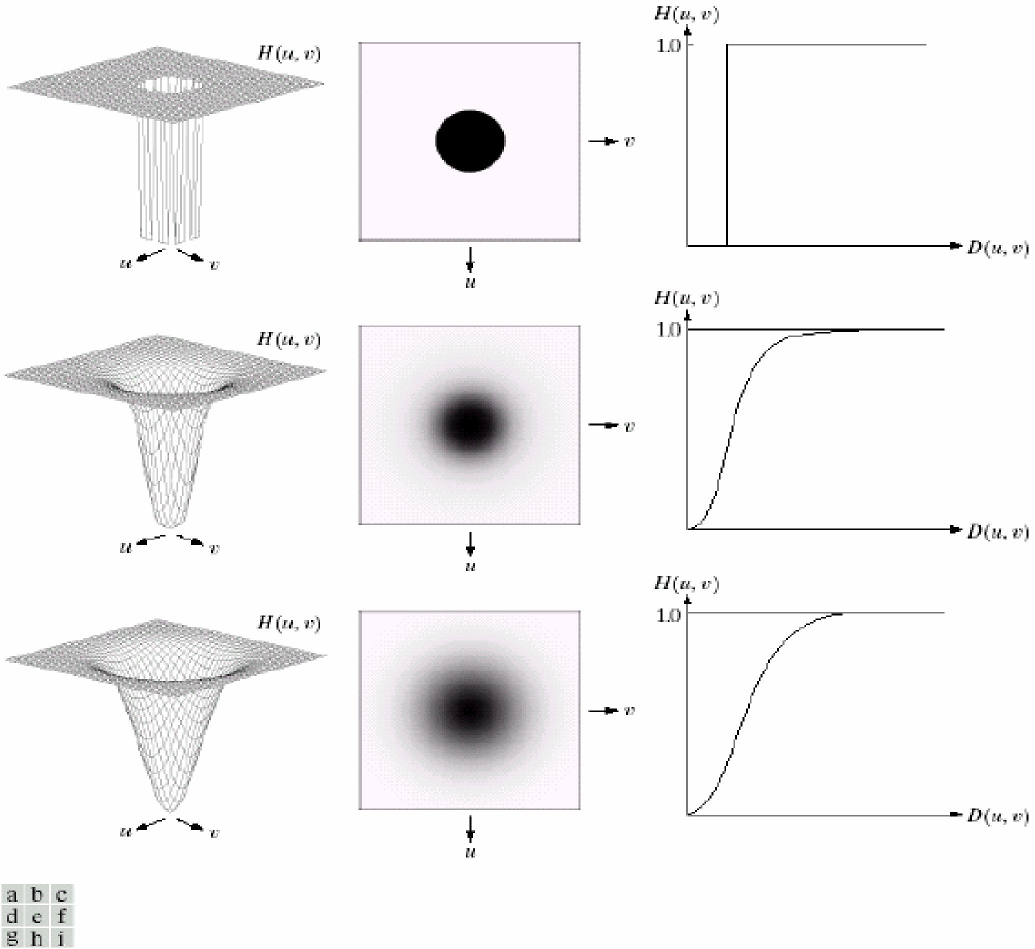


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Sharpening (Highpass Filter)

- Attenuate low-frequency components without disturbing high-frequency components.
- Retain the detail of the image information such as edges, fine texture etc.
- Highpass filter can be simply generated from a lowpass filter i.e.

$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

Sharpening (Highpass Filter)

Ideal Highpass Filter

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

- $D(u,v)$ – filter characteristic
- D_0 - cutoff point

Sharpening (Highpass Filter)

Gaussian Highpass

- Mathematical expression

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

For circular-shaped filter.

$$D(u,v) = \left(\left[u - \frac{P}{2}\right]^2 + \left[v - \frac{Q}{2}\right]^2\right)^{\frac{1}{2}}$$

Sharpening (Highpass Filter)

- Effects of using different D_0 for different filters

Ideal HPF

$D_0 = 30$ $D_0 = 60$ $D_0 = 160$

Sharpening (Highpass Filter)

Butterworth HPF $(n = 2)$

$D_0 = 30$

$D_0 = 60$

$D_0 = 160$

Sharpening (Highpass Filter)

Gaussian HPF

$D_0 = 30$ $D_0 = 60$ $D_0 = 160$

Laplacian (recap)

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

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

Laplacian in the FD

- It can be shown that:

$$\Im\left[\nabla^2 f(x,y)\right] = -(u^2 + v^2)F(u,v)$$

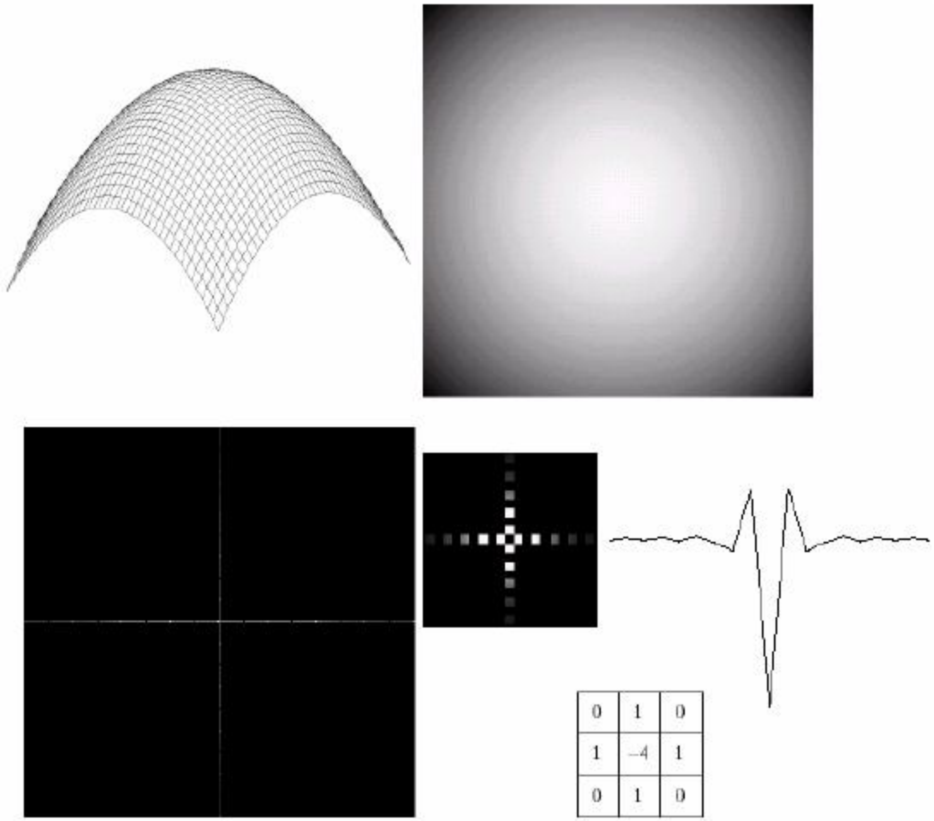
- The Laplacian can be implemented in the FD by using the filter

$$H(u,v) = -(u^2 + v^2)$$

- FT pair:

$$\nabla^2 f(x,y) \Leftrightarrow -[(u - M/2)^2 + (v - N/2)^2]F(u,v)$$

Laplacian in the Frequency Domain



a b
c d e
f

FIGURE 4.27 (a) 3-D plot of Laplacian in the frequency domain. (b) Image representation of (a). (c) Laplacian in the spatial domain obtained from the inverse DFT of (b). (d) Zoomed section of the origin of (c). (e) Gray-level profile through the center of (d). (f) Laplacian mask used in Section 3.7.

Homomorphic Filter

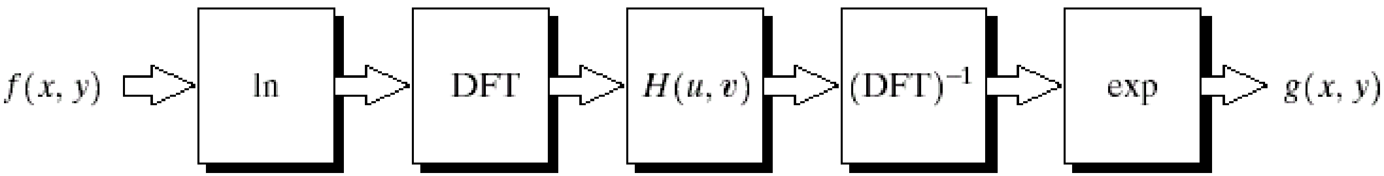


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

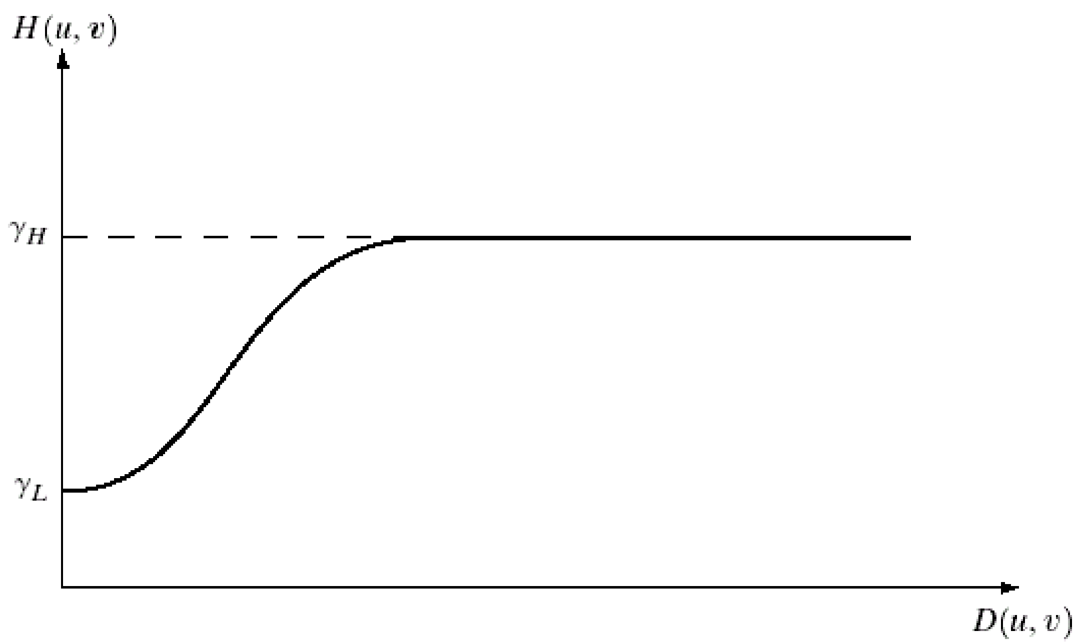
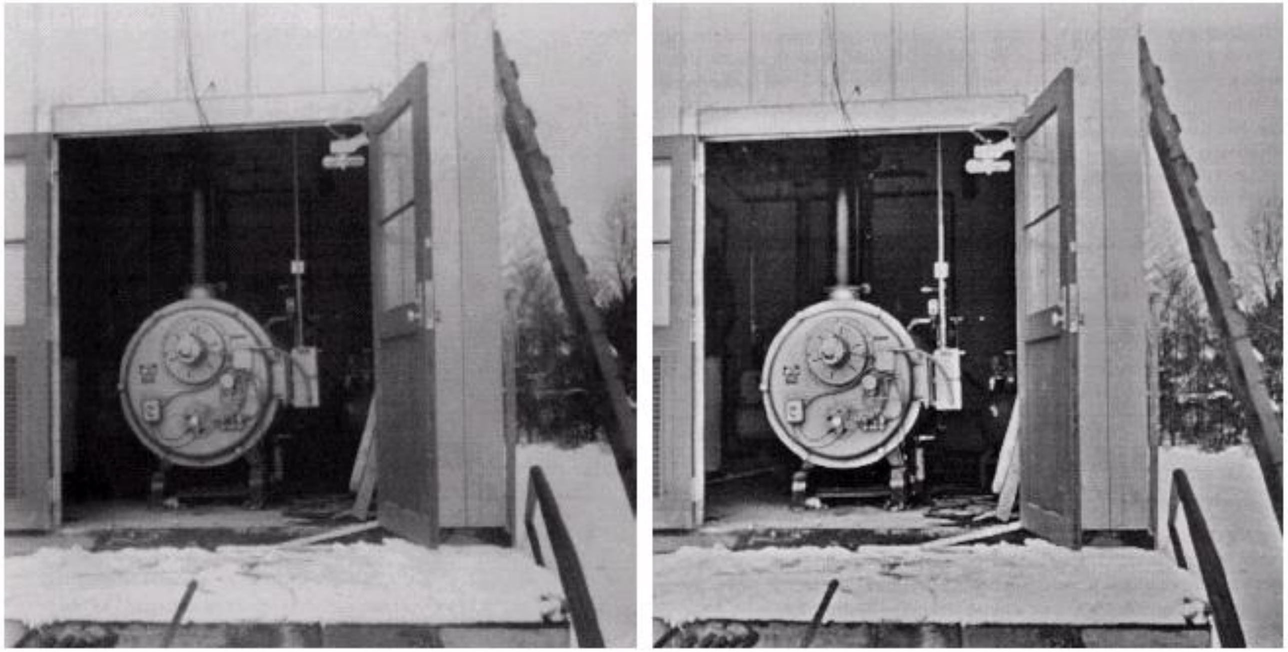


FIGURE 4.32 Cross section of a circularly symmetric filter function. $D(u, v)$ is the distance from the origin of the centered transform.

Result of Homomorphic filter



a b

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

a b
c d

FIGURE 4.34
(a) Fourier spectrum showing back-to-back half periods in the interval $[0, M-1]$.
(b) Shifted spectrum showing a full period in the same interval.
(c) Fourier spectrum of an image, showing the same back-to-back properties as (a), but in two dimensions.
(d) Centered Fourier spectrum.

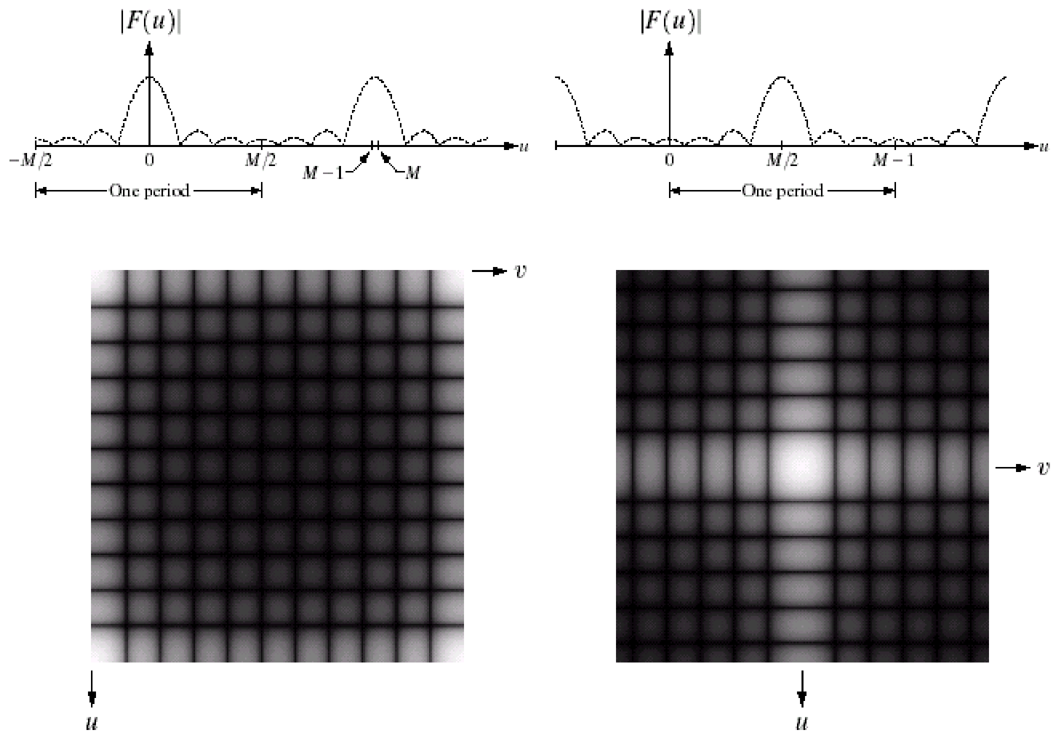


TABLE 4.1
Summary of some important properties of the 2-D Fourier transform.

Property	Expression(s)
Fourier transform	$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$
Inverse Fourier transform	$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$
Polar representation	$F(u,v) = \lvert F(u,v)\rvert e^{-j\phi(u,v)}$
Spectrum	$\lvert F(u,v)\rvert = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$, $R = \text{Real}(F)$ and $I = \text{Imag}(F)$
Phase angle	$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$
Power spectrum	$P(u,v) = \lvert F(u,v)\rvert^2$
Average value	$\bar{f}(x,y) = F(0,0) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$
Translation	$f(x,y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$ $f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M+vy_0/N)}$ When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$, then $f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$ $f(x-M/2, y-N/2) \Leftrightarrow F(u,v)(-1)^{u+v}$

TABLE 4.1 *(continued)*

Conjugate symmetry	$F(u, v) = F^*(-u, -v)$ $\lvert F(u, v)\rvert = \lvert F(-u, -v)\rvert$
Differentiation	$\dfrac{\partial^n f(x, y)}{\partial x^n} \Leftrightarrow (ju)^n F(u, v)$ $(-jx)^n f(x, y) \Leftrightarrow \dfrac{\partial^n F(u, v)}{\partial u^n}$
Laplacian	$\nabla^2 f(x, y) \Leftrightarrow -(u^2 + v^2)F(u, v)$
Distributivity	$\Im[f_1(x, y) + f_2(x, y)] = \Im[f_1(x, y)] + \Im[f_2(x, y)]$ $\Im[f_1(x, y) \cdot f_2(x, y)] \neq \Im[f_1(x, y)] \cdot \Im[f_2(x, y)]$
Scaling	$af(x, y) \Leftrightarrow aF(u, v), f(ax, by) \Leftrightarrow \dfrac{1}{\lvert ab\rvert} F(u/a, v/b)$
Rotation	$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$ $f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$
Periodicity	$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ $f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$
Separability	See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.

TABLE 4.1 *(continued)*

Property	Expression(s)
Computation of the inverse Fourier transform using a forward transform algorithm	$\frac{1}{MN} f^*(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi(ux/M + vy/N)}$ This equation indicates that inputting the function $F^*(u, v)$ into an algorithm designed to compute the forward transform (right side of the preceding equation) yields $f^*(x, y)/MN$. Taking the complex conjugate and multiplying this result by MN gives the desired inverse.
Convolution†	$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$
Correlation†	$f(x, y) \circ h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n)$
Convolution theorem†	$f(x, y) * h(x, y) \Leftrightarrow F(u, v) H(u, v)$; $f(x, y) h(x, y) \Leftrightarrow F(u, v) * H(u, v)$
Correlation theorem†	$f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v) H(u, v)$; $f^*(x, y) h(x, y) \Leftrightarrow F(u, v) \circ H(u, v)$

TABLE 4.1 *(continued)*

Some useful FT pairs:

Impulse	$\delta(x, y) \Leftrightarrow 1$
Gaussian	$A\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2(x^2+y^2)} \Leftrightarrow Ae^{-(u^2+v^2)/2\sigma^2}$
Rectangle	$\text{rect}[a, b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow \frac{1}{2}\left[\delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0)\right]$
Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow j\frac{1}{2}\left[\delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0)\right]$

† Assumes that functions have been extended by zero padding.

Master in Computer Vision