Computer Vision

Course 5

Filtering in the Frequency Domain

Filter: a device or material for suppressing or minimizing waves or oscillations of certain frequencies

Frequency: the number of times that a periodic function repeats the same sequence of values during a unit variation of the independent variable

Fourier series and Transform

Fourier in a memoir in 1807 and published in 1822 in his book *La Théorie Analitique de la Chaleur* states that any periodic function can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient (called now a *Fourier series*). Even function that are not periodic (but whose area under the curve is finite) can be expressed as the integral of sines and/or cosines multiplied by a weighing function – the *Fourier* *transform*. Both representations share the characteristic that a function expressed in either a Fourier series or transform, can be reconstructed (recovered) completely via an inverse process, with no loss of information. It allows us to work in the "Fourier domain" and then return to the original domain of the function without losing any information.

Complex Numbers

$$C = R + i\,I \;,\quad R, I \in \mathbb{R},\; i = \sqrt{-1}\;,\quad R \text{ - real part },\; C - \text{imaginary part}$$

$$C^* = R - i\,I - \;\text{ the conjugate of the complex number } C$$

$$C = |C|\,(\cos\theta + i\sin\theta)\;,\;\; |C| = \sqrt{R^2 + I^2} - \text{complex number in polar coordinates}$$

$$e^{i\theta} = \cos\theta + i\sin\theta \;-\; \text{Euler's formula}$$

$$C = |C|\,e^{i\theta}$$

Course 5

Fourier Transforms

- analyzing an image as a set of spatial sinusoids in various directions, each sinusoid having a precise frequency

One-Dimensional Fourier Transform

Continuous Fourier transform (CFT) of a continuous function ***f***:

$$F(z) = \int_{-\infty}^{+\infty} f(x) e^{-i 2\pi z x} dx = R(z) + i\, I(z)$$

The corresponding inverse Fourier transform is:

$$f(x) = \int_{-\infty}^{+\infty} F(z) e^{-i 2\pi z x} dz$$

The magnitude function **|*F(z)*|** is called the *Fourier spectrum* of the function ***f(x)***:

$$|\boldsymbol{F}(\boldsymbol{z})| = \sqrt{[\boldsymbol{R}(\boldsymbol{z})]^2 + [\boldsymbol{I}(\boldsymbol{z})]^2}$$

The phase angle ***Φ(z)*** of the function ***f*** is denoted by:

$$\boldsymbol{\Phi}(\boldsymbol{z}) = \mathbf{atan}\left[\frac{\boldsymbol{I}(\boldsymbol{z})}{\boldsymbol{R}(\boldsymbol{z})}\right].$$

Example:

$$f(x) = \Pi(\frac{x}{a}),\ \Pi(x) = \begin{cases} 0 & |x| > \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 1 & |x| < \frac{1}{2} \end{cases}$$

$$F(z) = \int_{-\infty}^{+\infty} \Pi(\frac{x}{a}) e^{-i2\pi z x} dx = a\frac{\sin(za)}{za} = a\,\mathrm{sinc}(az)$$

Course 5

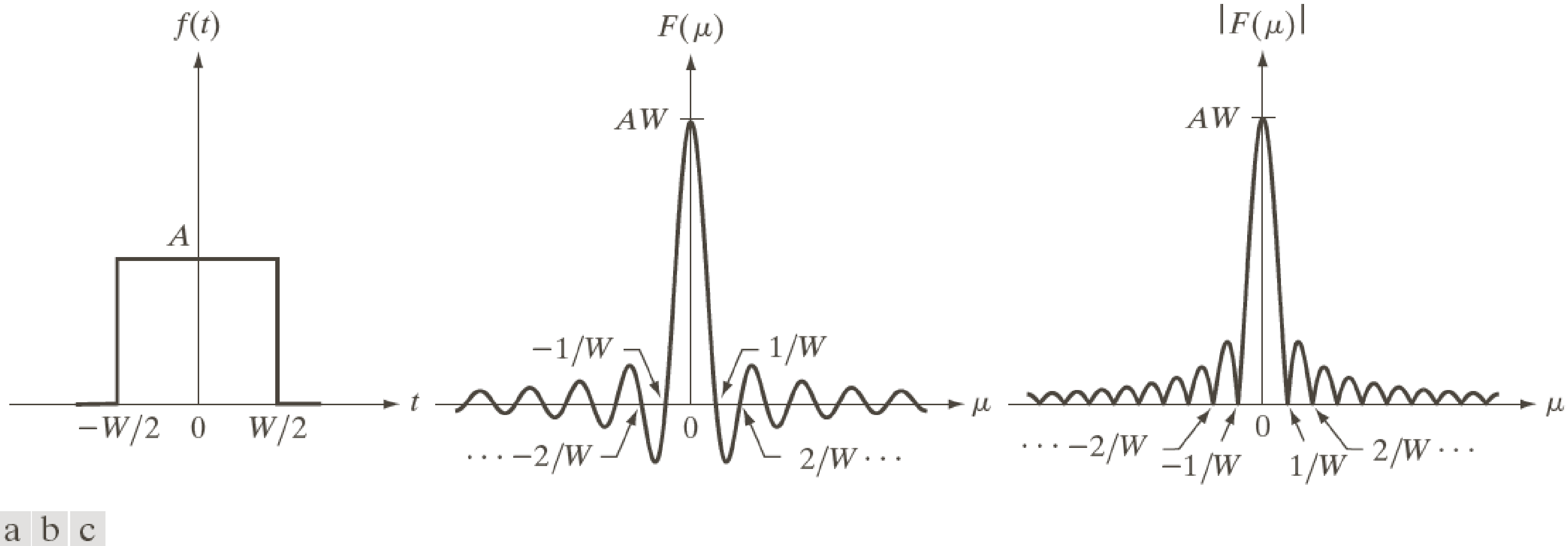


FIGURE 4.4 (a) A simple function; (b) its Fourier transform; and (c) the spectrum. All functions extend to infinity in both directions.

Course 5

Two-Dimensional Fourier Transform

Continuous Fourier transform (CFT) of a continuous function ***f(x,y)***:

$$F(z,w)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-i2\pi(zx+wy)}dy\,dx = R(z,w)+i\,I(z,w)$$

The magnitude function |***F(z,w)***| and the phase angle ***Φ(z,w)*** of the function ***f*** are defined by:

$$|F(z,w)|=\sqrt{\left[R(z,w)\right]^2+\left[I(z,w)\right]^2}$$

$$\Phi(z,w)=\text{atan}\left[\frac{I(z,w)}{R(z,w)}\right].$$

The *power spectrum* of ***f(x,y)*** is given by the relation:

$$P(z,w) = |F(z,w)|^2 = \left[R(z,w)\right]^2 + \left[I(z,w)\right]^2$$

The corresponding inverse Fourier transform is:

$$f(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(z,w)e^{-i2\pi(zx+wy)}dw\,dz.$$

Course 5

Discrete Fourier Transform (DFT)

f(x)={f(0), f(1), …, f(N-1)}

One-dimensional DFT

$$F(u)=\frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-i\frac{2\pi ux}{N}},\ u=0,1,\ldots,N-1$$

The inverse transformation

$$f(x)=\sum_{u=0}^{N-1} F(u)e^{i\frac{2\pi ux}{N}},\ x=0,1,\ldots,N-1$$

$f = \{f(x,y);\ x=0,1,\ldots,M-1\ ;\ y=0,1,\ldots,N-1\}$

Two-dimensional DFT

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i\,2\pi\left(\frac{u\,x}{M}+\frac{v\,y}{N}\right)},\ \ u = \overline{0, M-1}, v = \overline{0, N-1}$$

The inverse transformation

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{i\,2\pi\left(\frac{u\,x}{M}+\frac{v\,y}{N}\right)},\ \ x = \overline{0, M-1}, y = \overline{0, N-1}$$

Transformation Kernels

$$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)g(x,y,u,v),$$

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)h(x,y,u,v),$$

g(x,y,u,v) – forward transformation kernel

h(x,y,u,v) – inverse transformation kernel

$$g(x,y,u,v)=g_1(x,u)g_2(y,v) \rightarrow \text{the kernel is separable}$$

$$g_1 \equiv g_2 \rightarrow \text{the kernel is symmetric}$$

DFT – the kernel is separable and symmetric

Properties

- **Translation** – the translation of a Fourier transform pair is:

$$f(x,y)e^{i2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)} \leftrightarrow F(u-u_0, v-v_0)$$

$$f(x-x_0, y-y_0) \leftrightarrow F(u,v)e^{-i2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)}$$

- **Rotation** – assume that ***f(x,y)*** undergoes a rotation of angle θ_0. Using polar coordinates:

$$x = r\cos\theta\ ,\ \ y = r\sin\theta\ ,\ \ f(x,y) = \bar{f}(r,\theta)$$

$$F(u,v) = \bar{F}(s,\omega)\ ,\ \ u = s\cos\omega,\ \ v = s\sin\omega$$

$$f \leftrightarrow F\ ,\ \ \bar{f}(r,\theta+\theta_0) \leftrightarrow \bar{F}(s,\omega+\theta_0)$$

If ***f(x,y)*** is rotated by θ_0 then the Fourier transform ***F*** will be rotated by the same angle.

- **Separability**: this property ensures that the computations can be performed by decomposing the two-dimensional transform into two one-dimensional transforms

$$F(u,v)=\left[\frac{1}{M}\sum_{x=0}^{M-1} f(x,y)e^{-i\frac{2\pi u x}{M}}\right]\left[\frac{1}{N}\sum_{y=0}^{N-1} f(x,y)e^{-i\frac{2\pi v y}{N}}\right]$$

$$f(x,y)=\left[\sum_{u=0}^{M-1} F(u,v)e^{i\frac{2\pi u x}{M}}\right]\left[\sum_{v=0}^{N-1} F(u,v)e^{i\frac{2\pi v y}{N}}\right]$$

Hence the two-dimensional DFT (and the inverse) can be computed by taking the one-dimensional DFT row-wise in the two-dimensional image and the result is again transformed column-wise by the same one-dimensional DFT.

- **Distributivity:** the DFT of the sum of two images f_1 and f_2 is identical with the sum of the DFT of these two function

$$F\left\{ f_1 + f_2 \right\} = F\{ f_1 \} + F\{ f_2 \}.$$

The property does not hold for the product of two functions:

$$F\left\{ f_1 \cdot f_2 \right\} \neq F\{ f_1 \} \cdot F\{ f_2 \}$$

- **Scaling property**

$$F\left\{k\ f\right\} = kF\{f\}\ ,\ k \in \mathbb{R}$$

- **Convolution**

$$F\left\{f_1 \otimes f_2\right\} = F\{f_1\} \cdot F\{f_2\}$$

$$\left(f_1 \otimes f_2\right)(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_1(u,v)\, f_2(x-u, v-u)\, du\, dv$$

$$\left(f_1 \otimes f_2\right)(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} f_1(u,v)\, f_2(x-u, y-v)$$

- **Periodicity**

$$f(x,y) = f(x+k_1M, y) = f(x, y+k_2N) = f(x+k_1M, y+k_2N) \ ,$$

$$k_1, k_2 - \text{integers}$$

$$F(u,v) = F(u+k_1M, v) = F(u, v+k_2N) = F(u+k_1M, v+k_2N)$$

$$f(x,y)(-1)^{x+y} \quad \leftrightarrow \quad F(u-\frac{M}{2}, v-\frac{N}{2})$$

This last relation shifts the data so that ***F(0,0)*** is at the center of the frequency rectangle defined by the intervals ***[0,M-1]*** and ***[0,N-1].***

Symmetry Properties

Odd and *even* part of a function:

$$w(x,y) = w_e(x,y) + w_o(x,y)$$

$$w_e(x,y) = \frac{w(x,y) + w(-x,-y)}{2}$$

$$w_o(x,y) = \frac{w(x,y) - w(-x,-y)}{2}$$

$$w_e(x,y) = w_e(-x,-y) \; - \; \textit{symmetric}$$

$$w_o(x,y) = -w_o(-x,-y) \; - \; \textit{antisymmetric}$$

For digital images, evenness and oddness become:

$$w_e(x,y) = w_e(M-x, N-y)$$

$$w_o(x,y) = -w_o(M-x, N-y)$$

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} w_e(x,y)\,w_o(x,y) = 0$$

Course 5

	Spatial Domain†		Frequency Domain†
1)	$f(x, y)$ real	$\Leftrightarrow$	$F^*(u, v) = F(-u, -v)$
2)	$f(x, y)$ imaginary	$\Leftrightarrow$	$F^*(-u, -v) = -F(u, v)$
3)	$f(x, y)$ real	$\Leftrightarrow$	$R(u, v)$ even; $I(u, v)$ odd
4)	$f(x, y)$ imaginary	$\Leftrightarrow$	$R(u, v)$ odd; $I(u, v)$ even
5)	$f(-x, -y)$ real	$\Leftrightarrow$	$F^*(u, v)$ complex
6)	$f(-x, -y)$ complex	$\Leftrightarrow$	$F(-u, -v)$ complex
7)	$f^*(x, y)$ complex	$\Leftrightarrow$	$F^*(-u - v)$ complex
8)	$f(x, y)$ real and even	$\Leftrightarrow$	$F(u, v)$ real and even
9)	$f(x, y)$ real and odd	$\Leftrightarrow$	$F(u, v)$ imaginary and odd
10)	$f(x, y)$ imaginary and even	$\Leftrightarrow$	$F(u, v)$ imaginary and even
11)	$f(x, y)$ imaginary and odd	$\Leftrightarrow$	$F(u, v)$ real and odd
12)	$f(x, y)$ complex and even	$\Leftrightarrow$	$F(u, v)$ complex and even
13)	$f(x, y)$ complex and odd	$\Leftrightarrow$	$F(u, v)$ complex and odd

†Recall that x, y, u, and v are *discrete* (integer) variables, with x and u in the range $[0, M - 1]$, and y, and v in the range $[0, N - 1]$. To say that a complex function is *even* means that its real *and* imaginary parts are even, and similarly for an odd complex function.

TABLE 4.1 Some symmetry properties of the 2-D DFT and its inverse. $R(u, v)$ and $I(u, v)$ are the real and imaginary parts of $F(u, v)$, respectively. The term *complex* indicates that a function has nonzero real and imaginary parts.

Fourier Spectrum and Phase Angle

Express the Fourier transform in polar coordinates:

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

$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$ is called *Fourier* or *frequency spectrum*

$$\phi(u,v) = \arctan\left[\frac{I(u,v)}{R(u,v)}\right]$$ is the *phase angle*

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$ - *the power spectrum*

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

$$\phi(u,v) = -\phi(-u,-v)$$

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

$$F(0,0)=\bar{f} \quad , \quad \bar{f}=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)$$

$\bar{f}$ – the average value of the image f

$$|F(0,0)|=MN\,|\bar{f}|$$

Because ***MN*** usually is large, **|*F(0,0)*|** is the largest component of the spectrum by a factor that can be several orders of magnitude larger than other terms.

F(0,0) sometimes is called the ***dc*** *component* of the transform. (dc='direct current' – current of zero frequency)

The 2-D Convolution Theorem

2-D circular convolution:

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

The 2-D convolution theorem

$$f(x,y)\otimes h(x,y) \quad \leftrightarrow \quad F(u,v)H(u,v)$$

$$f(x,y)h(x,y) \quad \leftrightarrow \quad F(u,v)\otimes H(u,v)$$

Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x, y)$	$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$
2) Inverse discrete Fourier transform (IDFT) of $F(u, v)$	$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$
3) Polar representation	$F(u, v) = \lvert F(u, v) \rvert e^{j\phi(u,v)}$
4) Spectrum	$\lvert F(u, v) \rvert = \left[R^2(u, v) + I^2(u, v) \right]^{1/2}$ $R = \text{Real}(F); \quad I = \text{Imag}(F)$
5) Phase angle	$\phi(u, v) = \tan^{-1}\left[\frac{I(u, v)}{R(u, v)} \right]$
6) Power spectrum	$P(u, v) = \lvert F(u, v) \rvert^2$
7) Average value	$\bar{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} F(0, 0)$

(Continued)

TABLE 4.2
Summary of DFT definitions and corresponding expressions.

Course 5

Name	Expression(s)
8) Periodicity (k_1 and k_2 are integers)	$F(u, v) = F(u + k_1 M, v) = F(u, v + k_2 N) = F(u + k_1 M, v + k_2 N)$ $f(x, y) = f(x + k_1 M, y) = f(x, y + k_2 N) = f(x + k_1 M, y + k_2 N)$
9) Convolution	$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$
10) Correlation	$f(x, y) \,☆\, h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n)$
11) Separability	The 2-D DFT can be computed by computing 1-D DFT transforms along the rows (columns) of the image, followed by 1-D transforms along the columns (rows) of the result. See Section 4.11.1.
12) Obtaining the inverse Fourier transform using a forward transform algorithm.	$MNf^*(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi(ux/M + vy/N)}$ This equation indicates that inputting $F^*(u, v)$ into an algorithm that computes the forward transform (right side of above equation) yields $MNf^*(x, y)$. Taking the complex conjugate and dividing by MN gives the desired inverse. See Section 4.11.2.

Course 5

Name	DFT Pairs
1) Symmetry properties	See Table 4.1
2) Linearity	$af_1(x, y) + bf_2(x, y) \Leftrightarrow aF_1(u, v) + bF_2(u, v)$
3) Translation (general)	$f(x, y)e^{j2\pi(u_0x/M + v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0/M + vy_0/N)}$
4) Translation to center of the frequency rectangle, $(M/2, N/2)$	$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$
5) Rotation	$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$ $x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$
6) Convolution theorem[†]	$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$ $f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$

(*Continued*)

TABLE 4.3
Summary of DFT pairs. The closed-form expressions in 12 and 13 are valid only for continuous variables. They can be used with discrete variables by sampling the closed-form, continuous expressions.

Course 5

Name	DFT Pairs
7) Correlation theorem†	$f(x, y) \star h(x, y) \Leftrightarrow F^*(u, v) H(u, v)$ $f^*(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$
8) Discrete unit impulse	$\delta(x, y) \Leftrightarrow 1$
9) Rectangle	$\text{rect}[a, b] \Leftrightarrow ab \frac{\sin(\pi ua)}{(\pi ua)} \frac{\sin(\pi vb)}{(\pi vb)} e^{-j\pi(ua+vb)}$
10) Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $j\frac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) - \delta(u - Mu_0, v - Nv_0)\right]$
11) Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $\frac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) + \delta(u - Mu_0, v - Nv_0)\right]$
The following Fourier transform pairs are derivable only for continuous variables, denoted as before by t and z for spatial variables and by μ and ν for frequency variables. These results can be used for DFT work by sampling the continuous forms.	
12) *Differentiation* (The expressions on the right assume that $f(\pm\infty, \pm\infty) = 0$.)	$\left(\frac{\partial}{\partial t}\right)^m \left(\frac{\partial}{\partial z}\right)^n f(t, z) \Leftrightarrow (j2\pi\mu)^m (j2\pi\nu)^n F(\mu, \nu)$ $\frac{\partial^m f(t, z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu, \nu); \frac{\partial^n f(t, z)}{\partial z^n} \Leftrightarrow (j2\pi\nu)^n F(\mu, \nu)$
13) *Gaussian*	$A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+\nu^2)/2\sigma^2}$ (A is a constant)

†Assumes that the functions have been extended by zero padding. Convolution and correlation are associative, commutative, and distributive.

Course 5

The 2-D Discrete Fourier Transform and Its Inverse

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

$$=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\left(\cos\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)-i\sin\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)\right)$$

$$=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)-i\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\sin\left(2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right)$$

$$=A(u,v)+iB(u,v)=S(u,v)e^{i\Phi(u,v)}$$

f(x, y) is a digital image of size $M\times N$.

Given the transform ***F(u, v)*** we can obtain ***f(x, y)*** by using the *inverse discrete Fourier transform (IDFT)*:

Course 5

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{i2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \quad , \; x=0,1,\dots,M-1\;, y=0,1,\dots,N-1$$

$$f(x,y)=\begin{pmatrix} 0.5000 & 0.5000 & 0.7500 & 0.7500 & 0.7500 \\ 0.5000 & 0.5000 & 0.5000 & 0.7500 & 0.7500 \\ 0.2500 & 0.5000 & 0.5000 & 0.5000 & 0.7500 \\ 0.2500 & 0.2500 & 0.5000 & 0.5000 & 0.5000 \\ 0.2500 & 0.2500 & 0.2500 & 0.5000 & 0.5000 \end{pmatrix}$$

Course 5

$$F(u,v)=\begin{pmatrix} 12.50+i\,0.00 & -1.08+i\,1.48 & -0.79+i\,0.26 & -0.79-i\,0.26 & -1.08-i\,1.48 \\ 1.08-i\,1.48 & 0.00+i\,0.00 & 0.28+i\,0.09 & 0.28+i\,0.38 & 0.00-i\,0.18 \\ 0.79-i\,0.26 & -0.28-i\,0.09 & 0.00+i\,0.00 & 0.00-i\,0.77 & -0.28+i\,0.38 \\ 0.79-i\,0.26 & -0.28-i\,0.38 & 0.00+i\,0.77 & 0.00+i\,0.00 & -0.28+i\,0.09 \\ 1.08+i\,1.48 & 0.00+i\,0.18 & 0.28-i\,0.38 & 0.28-i\,0.09 & 0.00-i\,0.00 \end{pmatrix}=$$

$$\begin{pmatrix} 12.50 & -1.08 & -0.79 & -0.79 & -1.078 \\ 1.08 & 0 & 0.28 & 0.28 & 0 \\ 0.79 & -0.28 & 0 & 0 & -0.28 \\ 0.79 & -0.28 & 0 & 0 & -0.28 \\ 1.08 & 0 & 0.28 & 0.28 & 0 \end{pmatrix}+i\begin{pmatrix} 0 & 1.48 & 0.26 & -0.26 & -1.48 \\ -1.48 & 0 & 0.09 & 0.38 & -0.18 \\ -0.26 & -0.09 & 0 & -0.77 & 0.38 \\ 0.26 & -0.38 & 0.77 & 0 & 0.09 \\ 1.48 & 0.18 & -0.38 & -0.09 & 0 \end{pmatrix}$$

$$|F(u,v)|=\begin{pmatrix} 12.50 & 1.83 & 0.84 & 0.84 & 1.83 \\ 1.83 & 0 & 0.29 & 0.48 & 0.18 \\ 0.84 & 0.29 & 0 & 0.77 & 0.48 \\ 0.84 & 0.48 & 0.77 & 0 & 0.29 \\ 1.83 & 0.18 & 0.48 & 0.29 & 0 \end{pmatrix} \quad \Phi(u,v)=\begin{pmatrix} 0 & 7 & 9 & -9 & -7 \\ -3 & 5 & 1 & 3 & -5 \\ -1 & -9 & 0 & -5 & 7 \\ 1 & -7 & 5 & 0 & 9 \\ 3 & 5 & -3 & -1 & -5 \end{pmatrix}$$

Course 5

1NN, cityblock distance, “leave-one-out” cross validation

	Image features	Fourier features
Face recognition		
Faces95	95.49%	91.53%
AT&T	98.75%	97.50%
Gender		
Feret+AR (446f+672m)	83.27%	68.52%
CBIR		
Corel 1000	53.20%	63.8%
Texture		
T	32.20%	57.60%
Brodatz	72.582%	75.79%

Discrete Fourier Transform (DFT)

f(x)={f(0), f(1), …, f(N-1)}

One-dimensional DFT

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-i\frac{2\pi u x}{N}},\ u = 0,1,\ldots,N-1$$

The inverse transformation

$$f(x) = \sum_{u=0}^{N-1} F(u)e^{i\frac{2\pi u x}{N}},\ x = 0,1,\ldots,N-1$$

***f* = {*f*(*x*,*y*); *x*=0,1,…,*M*-1 ; *y*=0,1,…,*N*-1}**

Two-dimensional DFT

$$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-i\,2\pi\left(\frac{u\,x}{M}+\frac{v\,y}{N}\right)},\ \ u=\overline{0,M-1}, v=\overline{0,N-1}$$

The inverse transformation

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{i\,2\pi\left(\frac{u\,x}{M}+\frac{v\,y}{N}\right)},\ \ x=\overline{0,M-1}, y=\overline{0,N-1}$$

Course 5

Fast Fourier Transform

Using the above definition, the computational complexity of the DFT is $O(N^2)$. Using a divide-and-conquer approach one can reduce the computational complexity to $O(N\log_2 N)$. This algorithm is known as the *Fast Fourier Transform* (FFT).

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\left(k_N\right)^{ux}, \quad k_N = e^{-i\frac{2\pi}{N}}.$$

We assume that $N=2^P=2M$.

$$F(u)=\frac{1}{2M}\sum_{x=0}^{2M-1} f(x)\left(k_{2M}\right)^{ux} =$$

$$=\frac{1}{2}\left\{\frac{1}{M}\sum_{x=0}^{M-1} f(2x)\left(k_{2M}\right)^{u(2x)}+\frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)\left(k_{2M}\right)^{u(2x+1)}\right\}$$

$$=\frac{1}{2}\left\{\frac{1}{M}\sum_{x=0}^{M-1} f(2x)\left(k_{2M}\right)^{u(2x)}+\frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)\left(k_{2M}\right)^{2ux} k_{2M}^{u}\right\}$$

$$F(u)=\frac{1}{2}\left\{\frac{1}{M}\sum_{x=0}^{M-1} f(2x)\left(k_{M}\right)^{ux}+\frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)\left(k_{M}\right)^{ux} k_{2M}^{u}\right\}=$$

$$=\frac{1}{2}\left\{F_{\text{even}}(u)+k_{2M}^{u}F_{\text{odd}}(u)\right\} \quad (\text{ we used } k_{2M}^{2ux}=k_{M}^{ux})$$

$F_{\text{even}}(u)$ is the DFT of the sequence composed of the even samples ***f(2x) (f(0), f(2), …, f(2M-2))*** of the original discrete signal and $F_{\text{odd}}(u)$ is the DFT is the DFT of the odd samples ***f(2x+1) (f(1), f(3), …, f(2M-1))***. Both even and odd sequences have ***N/2*** length, thus the ***N***-point DFT ***F(u)*** can be computed as the sum of two $\frac{N}{2}$-point DFT.

Course 5

Filtering in the Frequency Domain

Let ***f(x,y)*** be a digital image and ***F(u,v)*** its (discrete) Fourier transform. Usually it is not possible to make direct associations between specific components of an image and its transform. We know that ***F(0,0)*** is proportional to the average intensity of the image. Low frequencies correspond to the slowly varying intensity components of an image, the higher frequencies correspond to faster intensity change in an image (edges, for ex.).

$$F(0,0) = MN\,\bar{f} \quad , \quad \bar{f} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$$

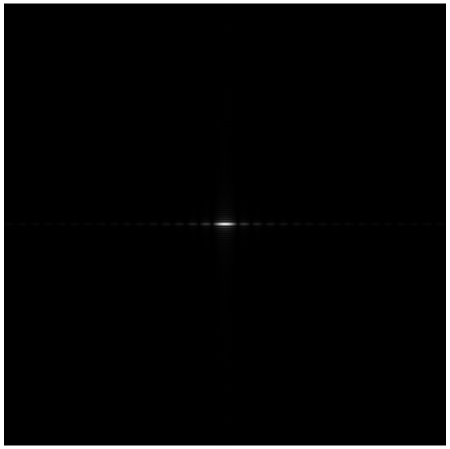
$\bar{f}$ – the average value of the image ***f***

$$f(x - x_0, y - y_0) \quad \leftrightarrow \quad F(u,v)e^{-i2\pi\left(\frac{x_0 y}{M} + \frac{y_0 v}{N}\right)}$$

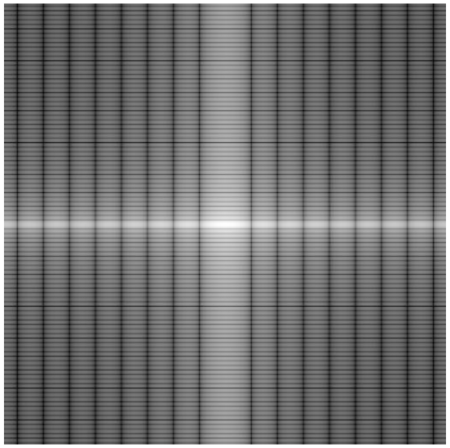
$$f(r, \theta + \theta_0) \quad \leftrightarrow \quad F(\omega, \varphi + \theta_0)$$

The spectrum is insensitive to image translation, and it rotates by the same angle as the image rotates.

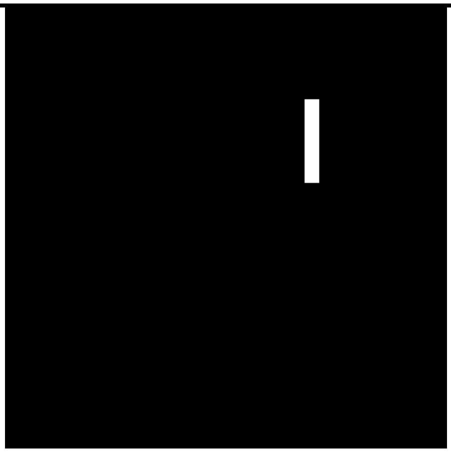
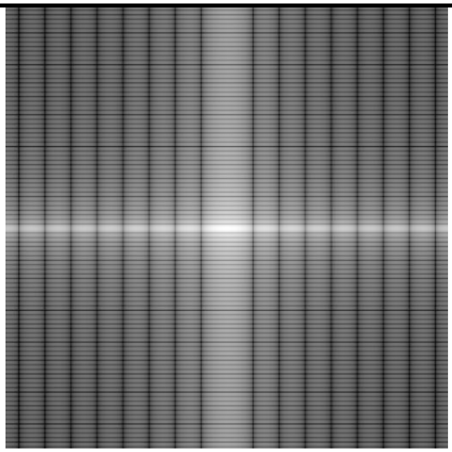
image | Fourier spectrum



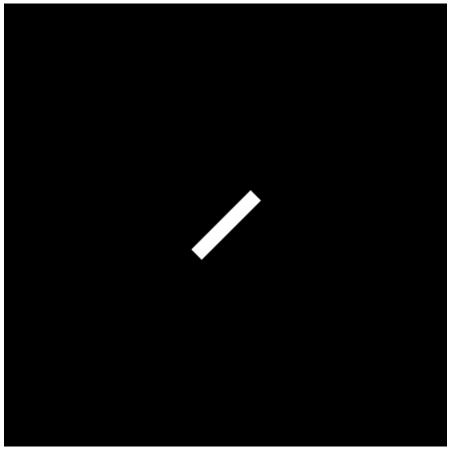
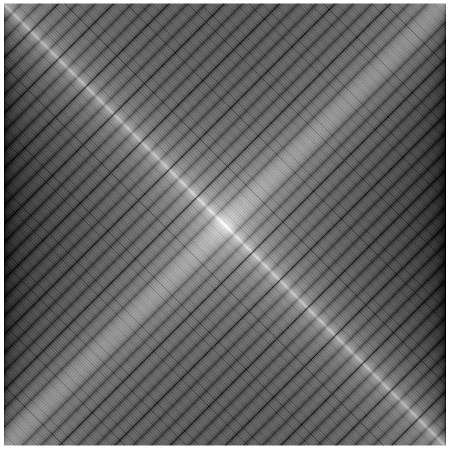
centered Fourier spectrum



log transformed centered Fourier spectrum

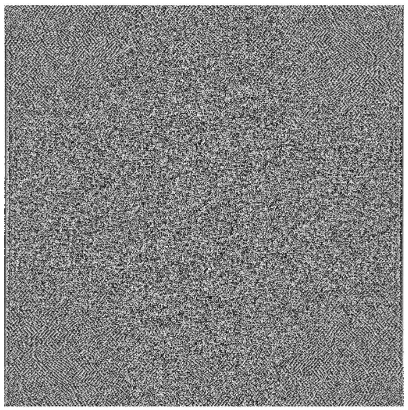
translated image Fourier spectrum

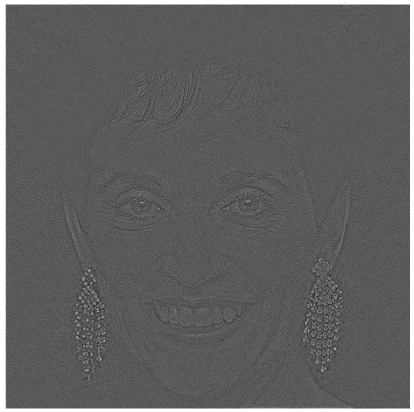
45° rotated image Fourier spectrum

The magnitude of the 2-D DFT is an array whose components determine the intensities in the image, the corresponding phase is an array of angles that carry the information about where discernible objects are located in the image.

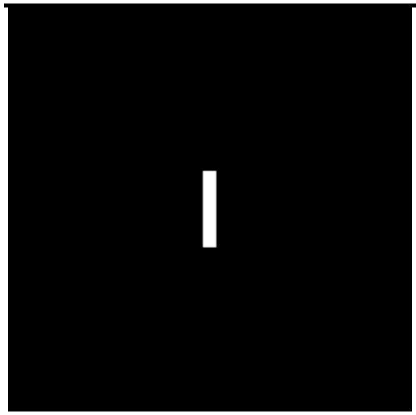
Woman



phase angle

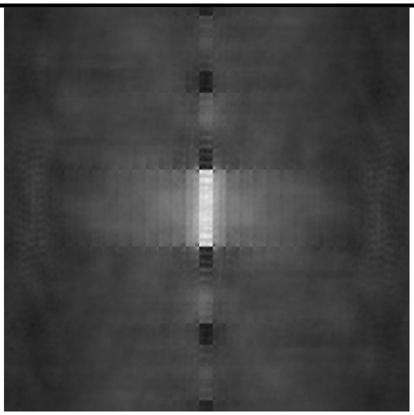


reconstruction only with phase angle



Rectangle

rectangle spectrum+phase angle woman



rectangle phase angle + spectrum woman

Course 5

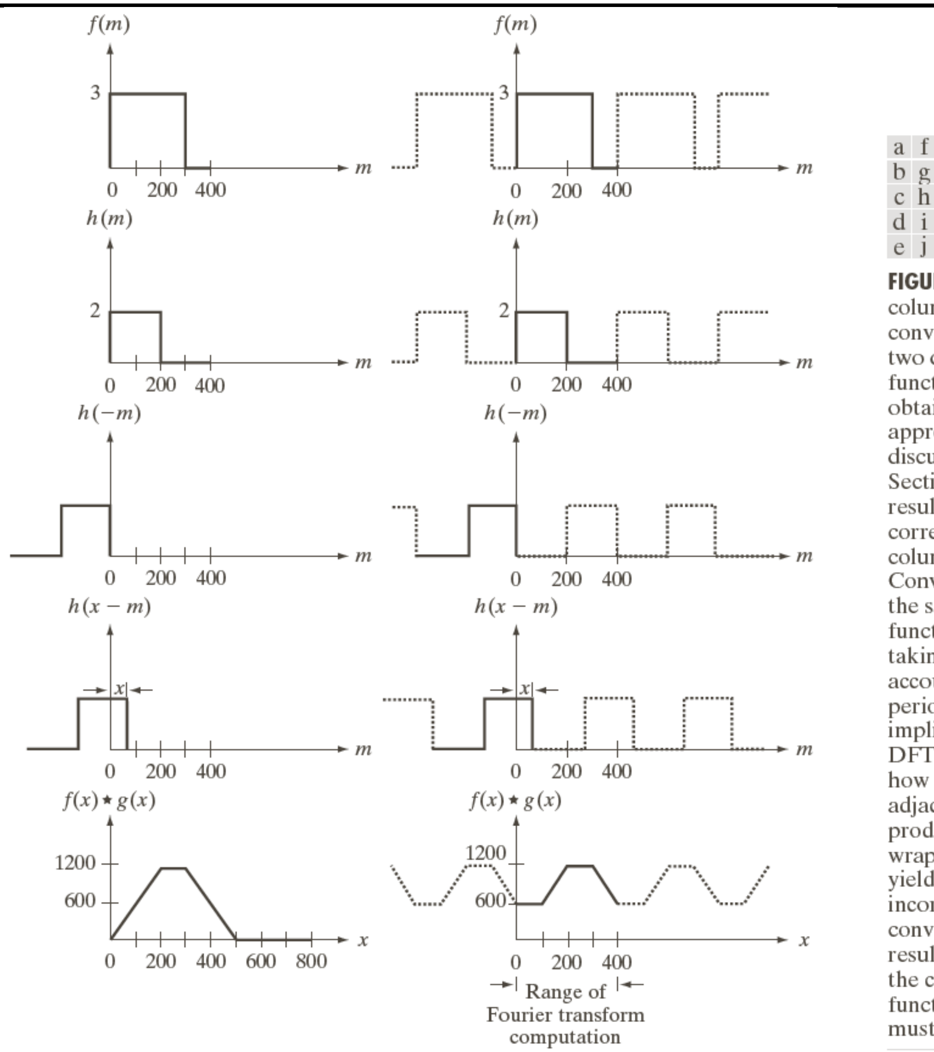


FIGURE 4.28 Left column: convolution of two discrete functions obtained using the approach discussed in Section 3.4.2. The result in (e) is correct. Right column: Convolution of the same functions, but taking into account the periodicity implied by the DFT. Note in (j) how data from adjacent periods produce wraparound error, yielding an incorrect convolution result. To obtain the correct result, function padding must be used.

If we use the DFT and the convolution theorem to obtain the same result in the left column of Figure 4.28, we must take into account the periodicity inherent in the expression for the DFT. The problem which appears in Figure 4.28 is commonly referred to as *wraparound error*. The solution to this problem is simple. Consider two functions ***f*** and ***h*** composed of ***A*** and ***B*** samples. It can be shown that if we append zeros to both functions so that they have the same length, denoted by ***P***, then wraparound is avoided by choosing:

$$P \geq A + B - 1$$

This process is called *zero padding*.

Let ***f(x,y)*** and ***h(x,y)*** be two image arrays of size ***A×B*** and ***C×D*** pixels, respectively. Wraparound error in their circular convolution can be avoided by padding these functions with zeros:

$$f_p(x,y)=\begin{cases} f(x,y) & 0\le x\le A-1 \text{ and } 0\le y\le B-1 \\ 0 & A\le x\le P \text{ and } B\le y\le Q \end{cases}$$

$$h_p(x,y)=\begin{cases} h(x,y) & 0\le x\le C-1 \text{ and } 0\le y\le D-1 \\ 0 & C\le x\le P \text{ and } D\le y\le Q \end{cases}$$

$$P\ge A+C-1 \quad (P\ge 2M-1) \quad , \quad Q\ge B+D-1 \quad (Q\ge 2N-1)$$

Frequency Domain Filtering Fundamentals

Given a digital image ***f(x,y)*** of size ***M×N***, the basic filtering equation has the form:

$$g(x,y) = \mathcal{F}^{-1}\left[H(u,v)F(u,v)\right] \qquad (1)$$

Where $\mathcal{F}^{-1}$ is the inverse discrete Fourier transform (IDFT), ***F(u,v)*** is the discrete Fourier transform (DFT) of the input image, ***H(u,v)*** is a *filter function* (also called *filter* or the *filter transfer function*) and ***g(x,y)*** is the filtered (output) image.

F*, *H*,** and ***g are arrays of the same size as ***f***, ***M×N***.

H(u,v) – symmetric about its center simplifies the computations and also requires that ***F(u,v)*** to be centered.

In order to obtain a centered ***F(u,v)*** the image ***f(x,y)*** is multiplied by $(-1)^{x+y}$ before computing its transform.

$$H(u,v)=\begin{cases} 0 & u=M/2, v=N/2\ (u=v=0) \\ 1 & \text{elsewhere} \end{cases}$$

This filter rejects the *dc* term (responsible for the average intensity of an image) and passes all other terms of ***F(u,v)***. This filter will reduce the average intensity of the output image to zero.

Low frequencies in the transform are related to slowly varying intensity components in an image (such as walls in a room, or a cloudless sky) and high frequencies are caused by sharp transitions in intensity, such as edges or noise. A filter ***H(u,v)*** that attenuates high frequencies while passing low frequencies (i.e. a *lowpass filter*) would blur an image while a filter with the opposite property (*highpass filter*) would enhance sharp detail, but cause a reduction of contrast in the image.

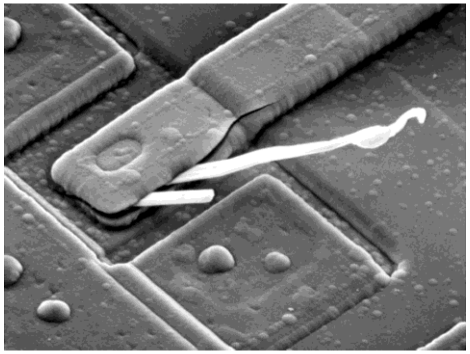
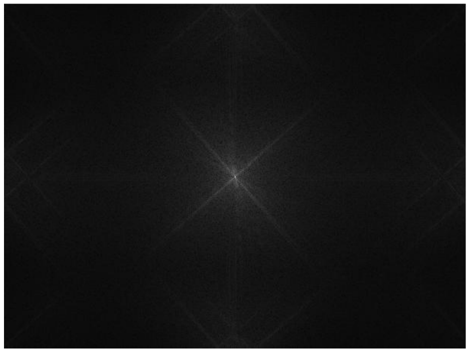
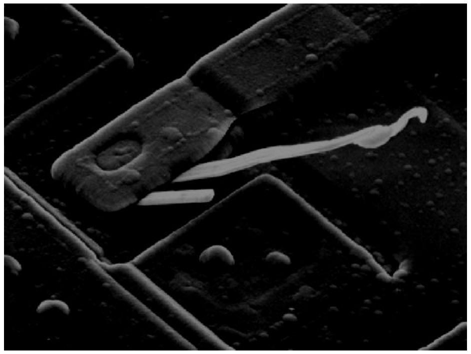


Image of damaged integrated circuit



Fourier spectrum



F(0,0)=0

Course 5

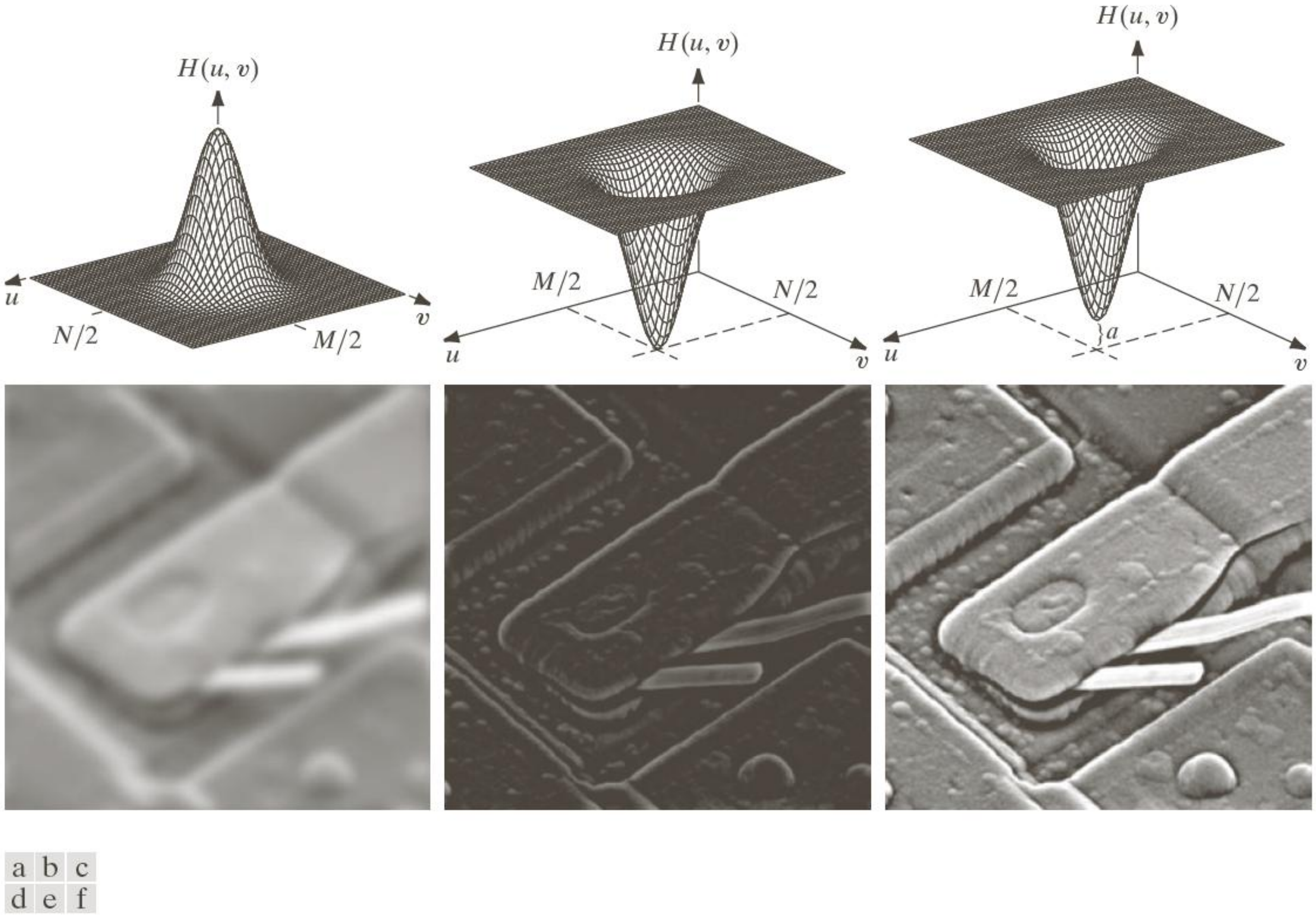


FIGURE 4.31 Top row: frequency domain filters. Bottom row: corresponding filtered images obtained using Eq. (4.7-1). We used $a = 0.85$ in (c) to obtain (f) (the height of the filter itself is 1). Compare (f) with Fig. 4.29(a).

The DFT is a complex array of the form:

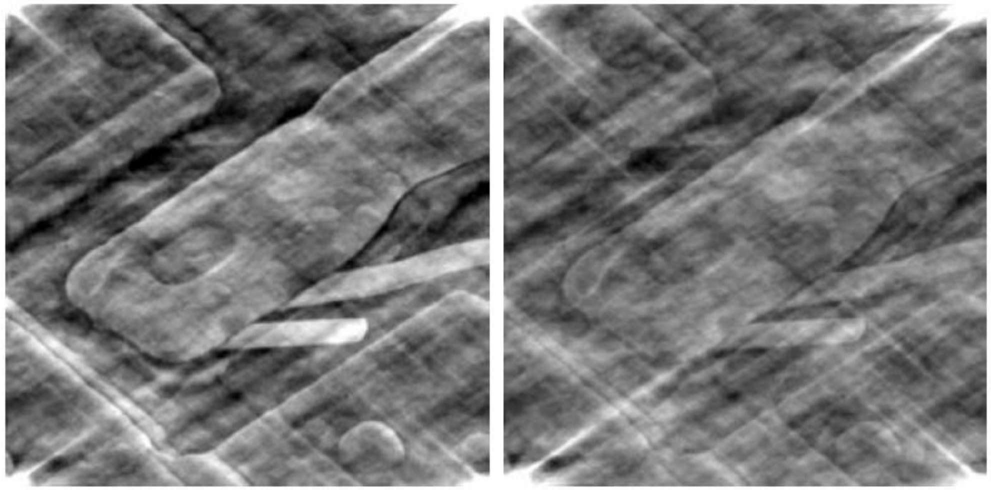
$$F(u,v) = R(u,v) + i\,I(u,v)$$

$$g(x,y) = \mathcal{F}^{-1}\left[H(u,v)R(u,v) + i\,H(u,v)I(u,v)\right]$$

The phase angle is not altered by filtering in this way. Filters that affect the real and the imaginary parts equally, and thus have no effect on the phase are called *zero-phase-shift* filters.

Even small changes in the phase angle can have undesirable effects on the filtered output.

Course 5



a b

FIGURE 4.35 (a) Image resulting from multiplying by 0.5 the phase angle in Eq. (4.6-15) and then computing the IDFT. (b) The result of multiplying the phase by 0.25. The spectrum was not changed in either of the two cases.

Main Steps for Filtering in the Frequency Domain

1. Given an input image $f(x,y)$ of size $M \times N$, obtain the padding parameters P and Q (usually $P=2M$, $Q=2N$)
2. Form a padded image $f_p(x,y)$, of size $P \times Q$ by appending the necessary numbers of zeros to $f(x,y)$ (f is in the upper left corner of f_p)
3. $f_p(x,y) = (-1)^{x+y} f_p(x,y)$ - to center its transform
4. Compute the DFT, $F(u,v)$, of the image obtain from 3.
5. Generate a real, symmetric filter function $H(u,v)$ of size $P \times Q$ with center at coordinates $\left(\frac{P}{2}, \frac{Q}{2}\right)$. Compute the array product

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

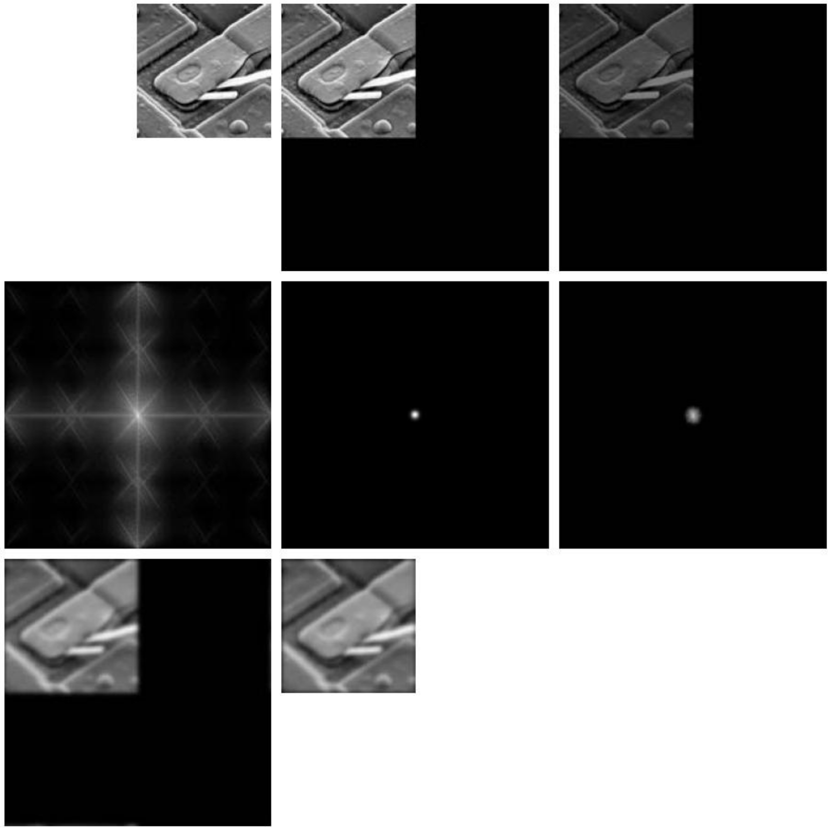
6. Obtain the processed image:

$$g_p(x,y)=\left\{\textbf{real}\left[\mathcal{F}^{-1}\left(G(u,v)\right)\right]\right\}(-1)^{x+y}$$

   The real part is selected in order to ignore parasitic complex components resulting from computational inaccuracies.

7. Obtain the output, filtered image, $g(x,y)$ by extracting the $M \times N$ region from the top, left corner of $g_p(x,y)$.

Course 5



a b c
d e f
g h

FIGURE 4.36 (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_p. (e) Centered Gaussian lowpass filter, H, of size $P \times Q$. (f) Spectrum of the product HF_p. (g) g_p, the product of $(-1)^{x+y}$ and the real part of the IDFT of HF_p. (h) Final result, g, obtained by cropping the first M rows and N columns of g_p.

Correspondence between Filtering in the Spatial and Frequency Domains

The link between filtering in the spatial domain and frequency domain is the convolution theorem.

Given a filter ***H(u,v)***, suppose that we want to find its equivalent representation in the spatial domain.

$$f(x,y) = \delta(x,y) \Rightarrow F(u,v) = 1$$

$$g(x,y) = \mathcal{F}^{-1}\left[H(u,v)F(u,v)\right] \Rightarrow h(x,y) = \mathcal{F}^{-1}\left[H(u,v)\right]$$

The inverse transform of the frequency domain filter, ***h(x,y)*** is the corresponding filter in the spatial domain.

Conversely, given a spatial filter, ***h(x,y)*** we obtain its frequency domain representation by taking the Fourier transform of the spatial filter:

$$h(x,y) \leftrightarrow H(u,v)$$

h(x,y) is sometimes called as the *(finite) impulse response* (FIR) of ***H(u,v)***.

One way to take advantage of the properties of both domains is to specify a filter in the frequency domain, compute its IDFT, and then use the resulting, full-size spatial filter as a guide for constructing smaller spatial filter masks.

Let ***H(u)*** denote the 1-D frequency domain Gaussian filter:

$$H(u) = Ae^{-\frac{u^2}{2\sigma^2}} \ , \ \sigma - \text{ the standard deviation}$$

The corresponding filter in the spatial domain is obtained by taking the inverse Fourier transform of ***H(u)***:

$$h(x) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2 x^2}$$

which is also a Gaussian filter. When ***H(u)*** has a broad profile (large value of σ), ***h(x)*** has a narrow profile and vice versa. As σ approaches infinity, ***H(u)*** tends toward a constant function and ***h(x)*** tends towards an impulse, which implies no filtering in the frequency and spatial domains.

Image Smoothing Using Frequency Domain Filters

Smoothing (blurring) is achieved in the frequency domain by high-frequency attenuation that is by *lowpass* filtering. We consider three types of lowpass filters:

ideal, *Butterworth*, *Gaussian*

The Butterworth filter has a parameter called the *filter order*. For high order values, the Butterworth filter approaches the ideal filter and for low values is more like a Gaussian filter.

All filters and images in these sections are consider padded with zeros, thus they have size $P \times Q$. The Butterworth filter may be viewed as providing a transition between the other two filters.

Ideal Lowpass Filters (ILPF)

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

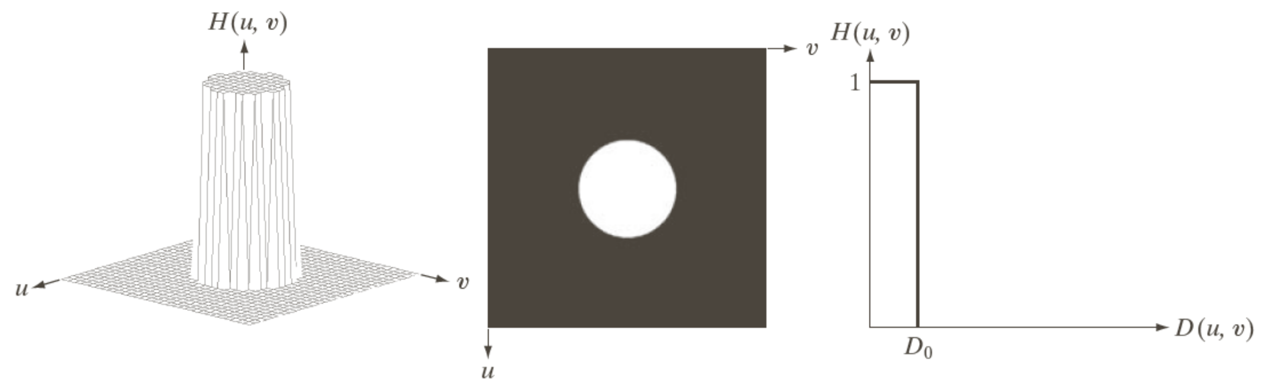
Where $D_0 \geq 0$ is a positive constant and $D(u,v)$ is the distance between (u,v) and the center of the frequency rectangle:

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

The name *ideal* indicates that all frequencies on or inside the circle of radius D_0 are passed without attenuation, whereas all frequencies outside the circle are completely eliminated (filtered out).

For an ILPF cross section, the point of transition between ***H(u,v)=1*** and ***H(u,v)=0*** is called the *cutoff frequency*. The sharp cutoff frequencies of an ILPF cannot be realized with electronic components, but they can be simulated in a computer.

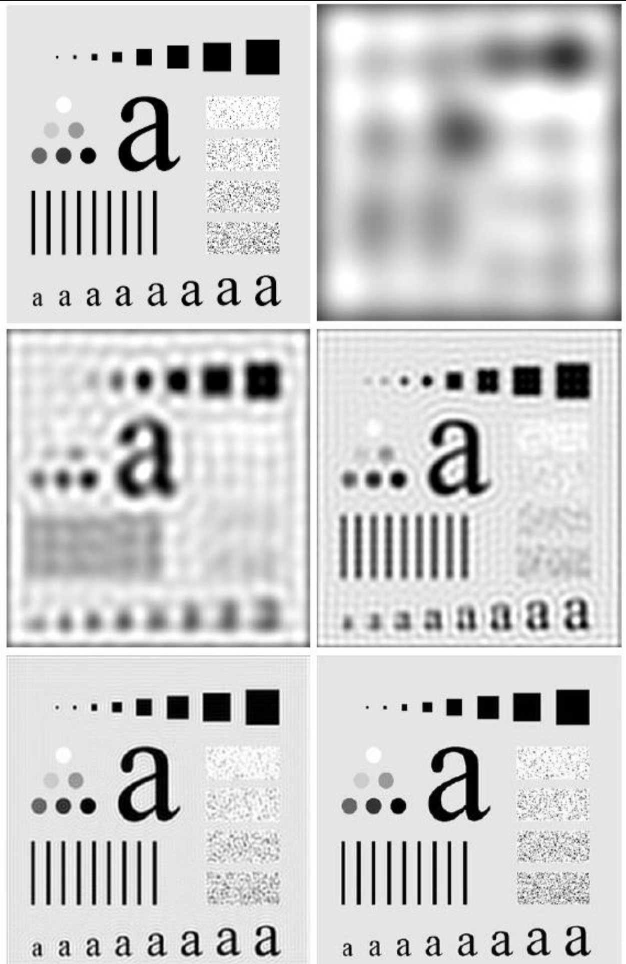
We can compare the ILPF by studying their behavior with respect to the cutoff frequencies.



a b c

FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Course 5



a b
c d
e f

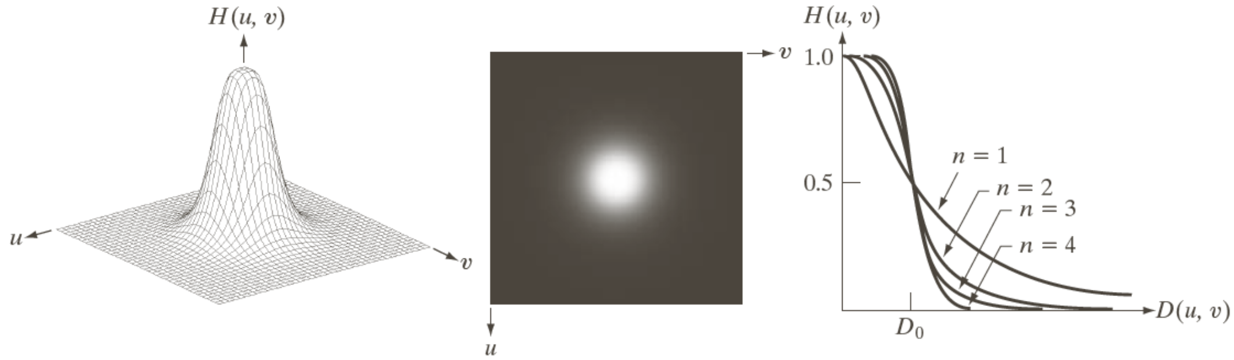
FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.

Butterworth Lowpass Filter (BLPF)

The transfer function of a Butterworth lowpass filter of order $\boldsymbol{n}$ and with cutoff frequency at distance $\boldsymbol{D_0}$ from the origin is:

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

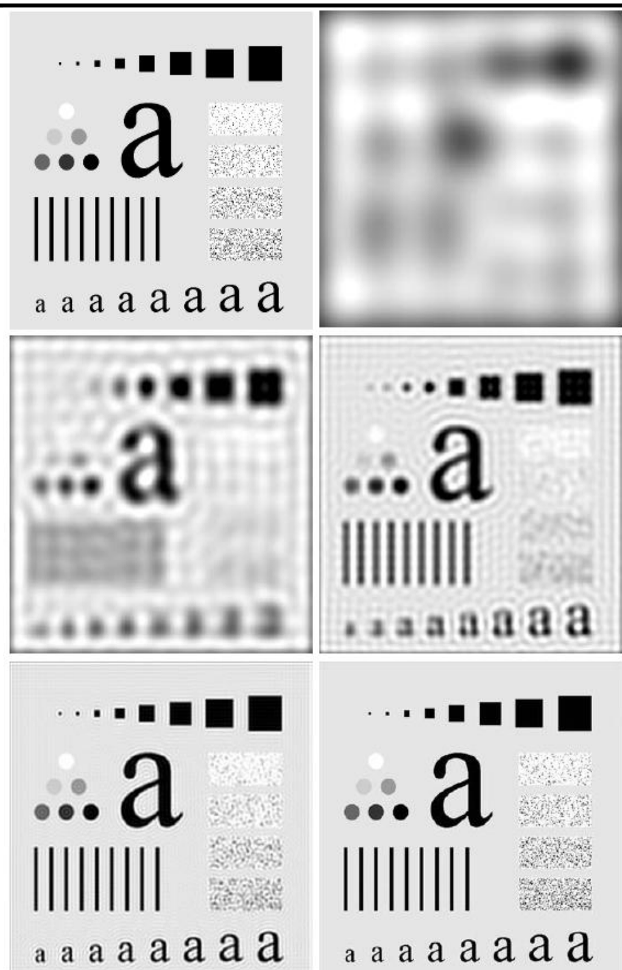
where $\boldsymbol{D(u,v)}$ is given by the relation (DUV).



a b c

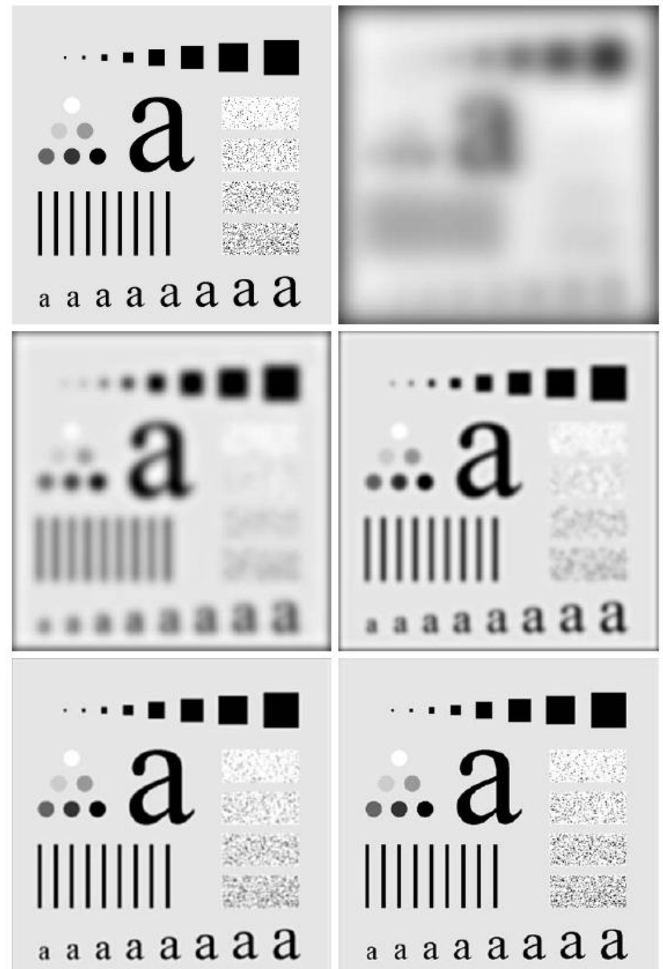
FIGURE 4.44 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

The BLPF transfer function does not have a sharp discontinuity that gives a clear cutoff between passed and filtered frequencies. For filters with smooth transfer functions, defining a cutoff frequency locus is made at points for which ***H(u,v)*** is down to a certain fraction of its maximum value.



a b
c d
e f

FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.



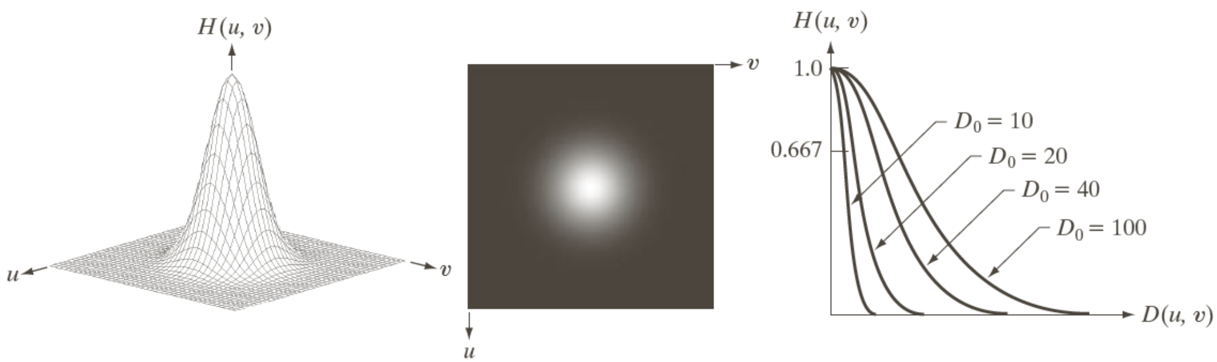
a b
c d
e f

FIGURE 4.45 (a) Original image. (b)–(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Fig. 4.42.

Gaussian Lowpass Filter (GLPF)

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

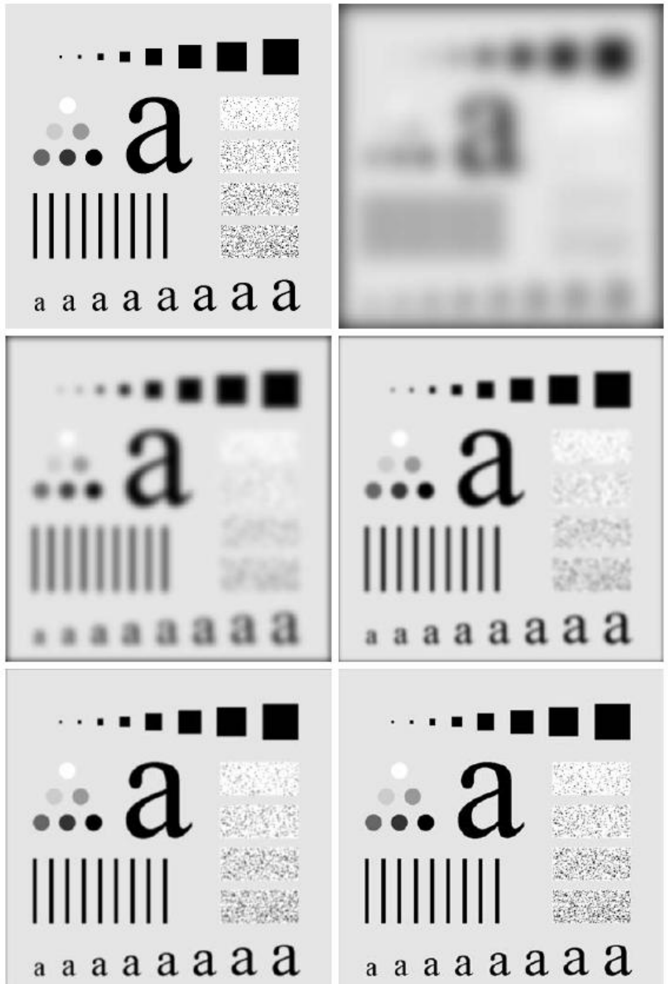
D_0 is the cutoff frequency. When $D(u,v) = D_0$ the GLPF is down to ***0.607*** of its maximum value.



a b c

FIGURE 4.47 (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.

Course 5



a b
c d
e f

FIGURE 4.48 (a) Original image. (b)–(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Figs. 4.42 and 4.45.

TABLE 4.4
Lowpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D(u,v)/D_0]^{2n}}$	$H(u,v) = e^{-D^2(u,v)/2D_0^2}$

Image Sharpening Using Frequency Domain Filters

Edges and other abrupt changes in intensities are associated with high-frequency components, image sharpening can be achieved in the frequency domain by highpass filters, which attenuates the low-frequency components without changing the high-frequency information in the Fourier transform.

A highpass filter is obtained from a given lowpass filter using the equation:

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

where $H_{LP}(u,v)$ is the transfer function of a lowpass filter.

Course 5

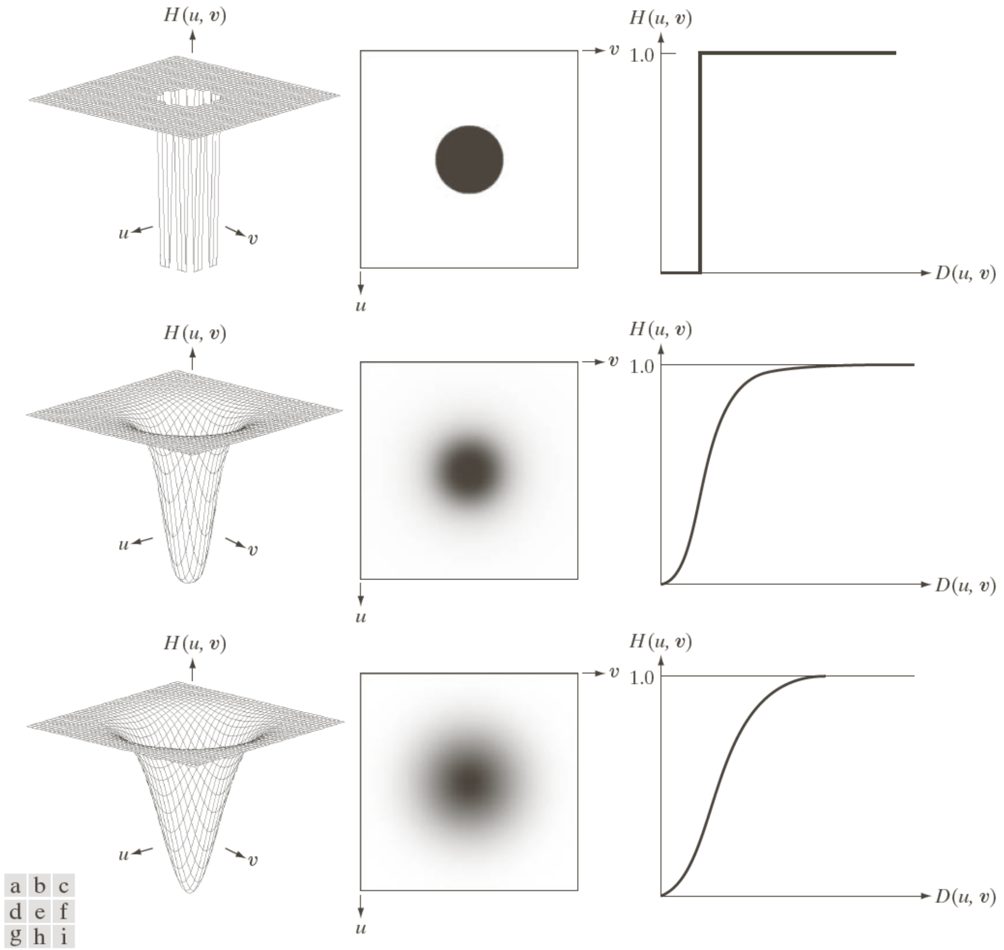


FIGURE 4.52 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Ideal Highpass Filter

A 2-D *ideal highpass filter* (IHPF) is defined as:

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

where D_0 is the cutoff frequency and $D(u,v)$ is given by equation (DUV). As for ILPF, the IHPF is not physically realizable.

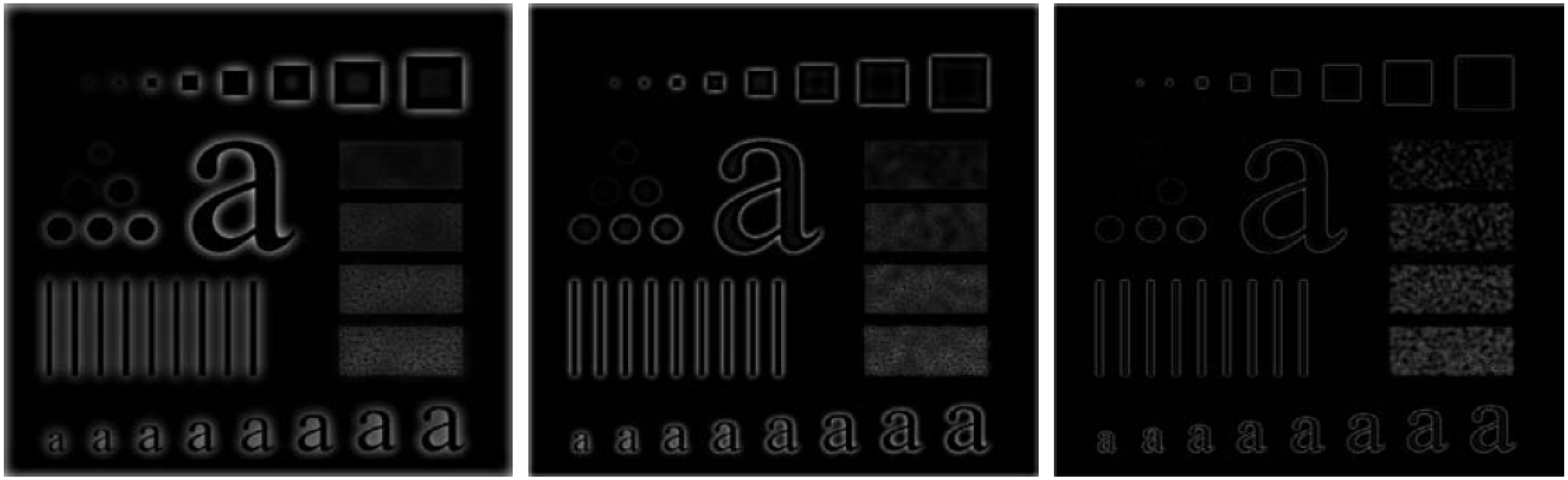
a b c

FIGURE 4.54 Results of highpass filtering the image in Fig. 4.41(a) using an IHPF with $D_0 = 30, 60,$ and 160.

Butterworth Highpass Filter (BHPF)

The transfer function of a Butterworth highpass filter of order $\boldsymbol{n}$ and with cutoff frequency at distance $\boldsymbol{D_0}$ from the origin is:

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



a b c

FIGURE 4.55 Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with $D_0 = 30, 60$, and 160, corresponding to the circles in Fig. 4.41(b). These results are much smoother than those obtained with an IHPF.

Gaussian Highpass Filter (GLPF)

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

a b c

FIGURE 4.56 Results of highpass filtering the image in Fig. 4.41(a) using a GHPF with $D_0 = 30, 60$, and 160, corresponding to the circles in Fig. 4.41(b). Compare with Figs. 4.54 and 4.55.

TABLE 4.5
Highpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D_0/D(u,v)]^{2n}}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$

a b c

FIGURE 4.57 (a) Thumb print. (b) Result of highpass filtering (a). (c) Result of thresholding (b). (Original image courtesy of the U.S. National Institute of Standards and Technology.)

Figure 4.57(a) is a ***1026×962*** image of a thumb print in which smudges are present. A keystep in automated figerprint recognition is enhancement of print ridges and the reduction of smudges. In this example a highpass filter was used to enhance ridges and reduce the effects of smudging. Enhancement of the ridges is accomplished by the fact that they contain high frequencies, which are unchanged by a highpass filter. This filter reduces low frequency components which correspond to slowly varying intensities in the image, such as background and smudges.

Figure 4.57(b) is the result of using a BHPF of order $n=4$, with a cutoff frequency $D_0=50$.

Figure 4.57(c) is the result of setting to black all negative values and to white all positive values in Figure 4.57(b) (a threshold intensity transformation)

The Laplacian in the Frequency Domain

The Laplacian can be implemented in the frequency domain using the filter:

$$H(u,v) = -4\pi^2\left(u^2 + v^2\right)$$

The centered Laplacian is:

$$H(u,v) = -4\pi^2\left[\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2\right] = -4\pi^2 D^2(u,v)$$

The Laplacian image is obtained as:

$$\nabla^2 f(x,y) = \mathcal{F}^{-1}\left\{H(u,v)F(u,v)\right\}$$

Enhancement is obtained with the equation:

$$g(x,y) = f(x,y) - \nabla^2 f(x,y) \qquad (1)$$

Computing $\nabla^2 f(x,y)$ with the above relation introduces DFT scaling factors that can be several orders of magnitude larger than the maximum value of ***f***. To fix this problem, we normalize the values of ***f(x,y)*** to the range ***[0,1]*** (before computing its DFT) and divide $\nabla^2 f(x,y)$ by its maximum value which will bring it to ***[-1,1]***.

Course 5

Spatial Correlation and Convolution

Correlation is the process of moving a filter mask over the image and computing the sum of products at each location. *Convolution* is similar with correlation, except that the filter is first rotated by ***180º***.

Correlation

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

Convolution

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

A function that contains a single ***1*** and the rest being ***0***s is called a *discrete unit impulse*. Correlating a function with a discrete unit impulse produces a rotated version of the filter at the location of the impulse.