

Digital Image Processing

Chap 4: Filtering in the Frequency Domain

清華大學電機系林嘉文

cwlin@ee.nthu.edu.tw

Tel: 03-5731152

Content

- Fourier Series
- Fourier Transform
- Discrete Fourier Transform (DFT)
- Fourier Transform for Image Enhancement
- Implementation

Background

- **Complex Number**
- **Definition:** A complex number C is defined as

$$C = R + jI$$
- R is the **real part**, I is the **imaginary part**
- The **complex conjugate** of a complex number C is

$$C^* = R - jI$$
- C can also be written as

$$C = |C|(\cos\theta + j\sin\theta) = |C|e^{j\theta}$$
- Magnitude** : $|C| = (R^2 + I^2)^{1/2}$ **Phase** : $\theta = \arctan(I/R)$
- **Euler's Formula:** $e^{j\theta} = \cos\theta + j\sin\theta$

2018/10/4

Digital Image Processing

3

Jean Baptiste Joseph Fourier (1768-1830)



Joseph Fourier

lived from 1768 to 1830

Fourier studied the mathematical theory of heat conduction. He established the partial differential equation governing heat diffusion and solved it by using infinite series of trigonometric functions.

*Find out more at:
<http://www-history.mcs.st-andrews.ac.uk/history/Mathematicians/Fourier.html>*

2018/10/4

Digital Image Processing

4

Jean Baptiste Joseph Fourier (1768-1830)

had crazy idea (1807):

Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.

- Don't believe it?
 - Neither did Lagrange, Laplace, Poisson and other big wigs
 - Not translated into English until 1878!
- But it's (mostly) true!
 - called Fourier Series
 - there are some subtle restrictions

2018/10/4

Digital



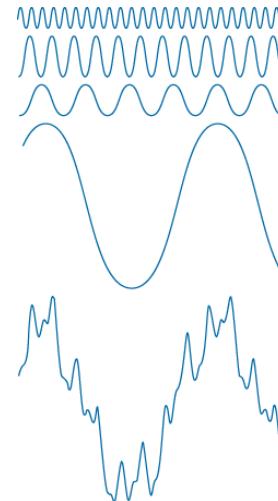
Fourier Series

• Fourier Series

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j \frac{2\pi n}{T} t}$$

where

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j \frac{2\pi n}{T} t} dt$$



2018/10/4

Digital Image Processing

6

Fourier Series

- A unit Impulse $\delta(t)$ is defined as

$$\delta(t) = \infty \text{ if } t = 0,$$

$$\delta(t) = 0 \text{ if } t \neq 0$$

- It satisfies the identity as

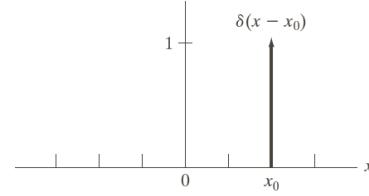
$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

- Shifting property

$$\int_{-\infty}^{\infty} f(t)\delta(t) dt = f(0)$$

- For an arbitrary point t_0 :

$$\int_{-\infty}^{\infty} f(t)\delta(t-t_0) dt = f(t_0)$$

**FIGURE 4.2**

A unit discrete impulse located at $x = x_0$. Variable x is discrete, and δ is 0 everywhere except at $x = x_0$.

2018/10/4

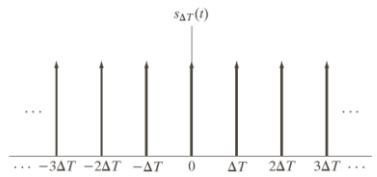
Digital Image Processing

7

Fourier Series

- Impulse train**

$$s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T)$$

**FIGURE 4.3** An impulse train.

2018/10/4

Digital Image Processing

8

Fourier Transform

- **Fourier Transform**

$$F(\mu) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t} dt$$

or $F(\mu) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi\mu x} dx$

- **Inverse Fourier Transform**

$$f(t) = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu t} du$$

or $f(x) = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu x} du$

2018/10/4

Digital Image Processing

9

Fourier Transform

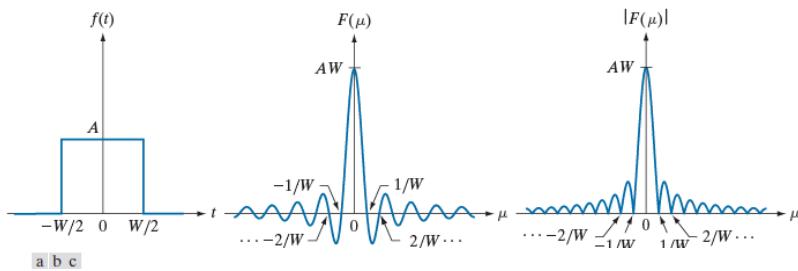


FIGURE 4.4 (a) A box function, (b) its Fourier transform, and (c) its spectrum. All functions extend to infinity in both directions. Note the inverse relationship between the width, W , of the function and the zeros of the transform.

$$F(\mu) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t} dt = \int_{-W/2}^{W/2} Ae^{-j2\pi\mu t} dt = AW \frac{\sin(\pi\mu W)}{\pi\mu W}$$

$$\text{sinc}(m) = \frac{\sin(\pi m)}{\pi m} \quad |F(\mu)| = AT \left| \frac{\sin(\pi\mu W)}{\pi\mu W} \right|$$

2018/10/4

Digital Image Processing

10

Fourier Transform of Sampled Functions

- **Sampling:**

Convert the continuous function $f(t)$ to a sequence of discrete values $f(k\Delta T)$, $k = 1, 2, \dots$ at uniform interval ΔT .

$$f_k = \int_{-\infty}^{\infty} f(t) \delta(t - k\Delta T) dt = f(k\Delta T)$$

- The sampled function is defined as

$$\tilde{f}(t) = f(t)s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} f(t)\delta(t - n\Delta T)$$

- An arbitrary sample k in the sequence is

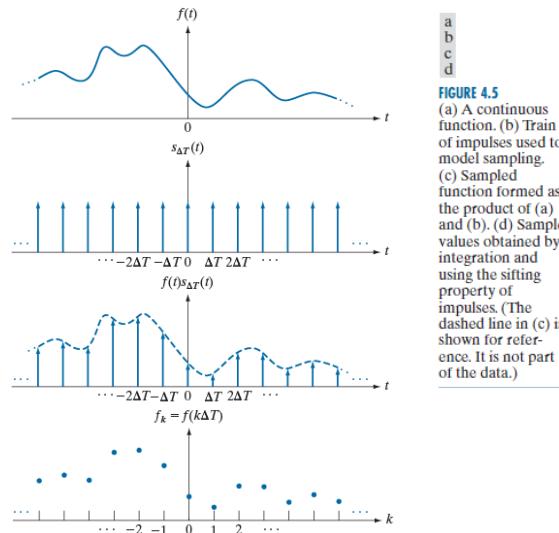
$$f_k = \int_{-\infty}^{\infty} f(t) \delta(t - k\Delta T) dt = f(k\Delta T)$$

2018/10/4

Digital Image Processing

11

Fourier Transform of Sampled Functions



2018/10/4

Digital Image Processing

12

Fourier Transform of Sampled Functions

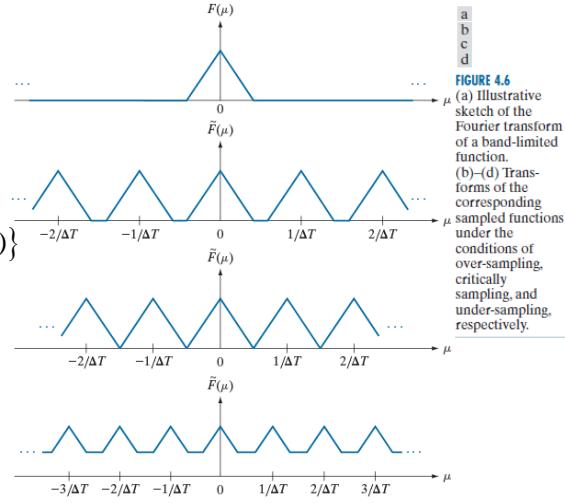
$F(\mu)$ is the Fourier transform of $f(t)$

$\tilde{F}(\mu)$ is the Fourier transform of $\tilde{f}(t)$

$$\tilde{F}(\mu) = \mathcal{F}\{\tilde{f}(t)\} = \mathcal{F}\{f(t)s_{\Delta T}(t)\} = F(\mu) * S(\mu)$$

$$S(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta(\mu - \frac{n}{\Delta T})$$

$$\tilde{F}(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F(\mu - \frac{n}{\Delta T})$$



a
b
c
d

FIGURE 4.6
(a) Illustrative sketch of the Fourier transform of a band-limited function.
(b)-(d) Transforms of the corresponding sampled functions under the conditions of over-sampling, critically sampling, and under-sampling, respectively.

2018/10/4

Digital Image Processing

13

Fourier Transform of Sampled Functions

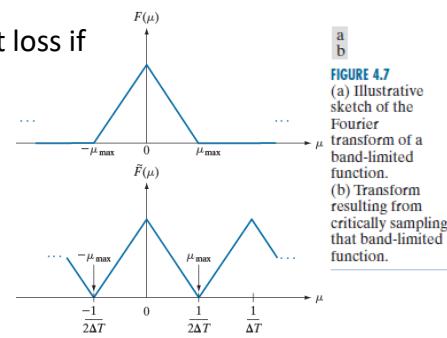
Extracting $F(\mu)$ from $\tilde{F}(\mu)$ without loss if the separation between copies is sufficient

$$1/2\Delta T > \mu_{\max}$$

$$\text{or } 1/\Delta T > 2\mu_{\max}$$

Sampling Theorem:

No information is lost if a **continuous band-limited** function is represented by samples acquired at a rate greater than twice the highest frequency content of the function



a
b

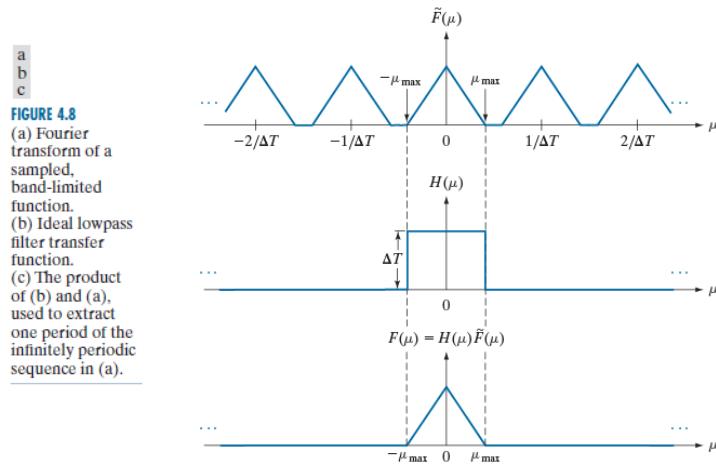
FIGURE 4.7
(a) Illustrative sketch of the Fourier transform of a band-limited function.
(b) Transform resulting from critically sampling that band-limited function.

2018/10/4

Digital Image Processing

14

Fourier Transform of Sampled Functions



2018/10/4

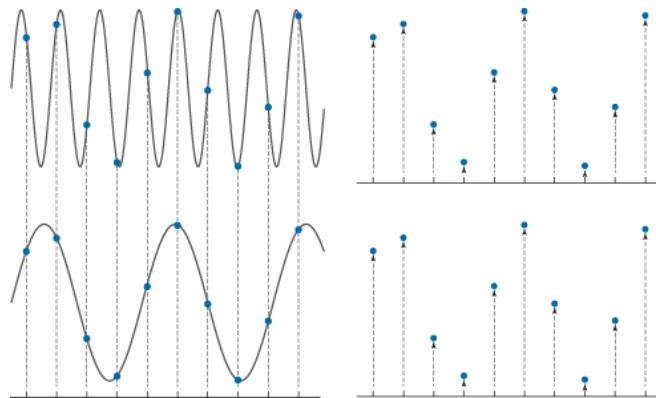
Digital Image Processing

15

Aliasing

a
b
c
d**FIGURE 4.9**

The functions in (a) and (c) are totally different, but their digitized versions in (b) and (d) are identical. Aliasing occurs when the samples of two or more functions coincide, but the functions are different elsewhere.



2018/10/4

Digital Image Processing

16

Aliasing

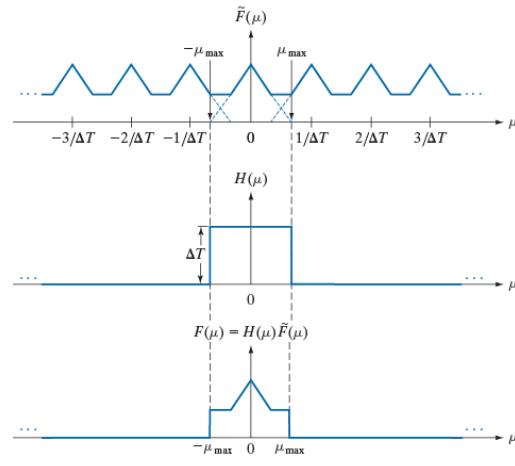


FIGURE 4.10 (a) Fourier transform of an under-sampled, band-limited function. (Interference between adjacent periods is shown dashed). (b) The same ideal lowpass filter used in Fig. 4.8. (c) The product of (a) and (b). The interference from adjacent periods results in aliasing that prevents perfect recovery of $F(\mu)$ and, consequently, of $f(t)$.

2018/10/4

Digital Image Processing

17

Aliasing

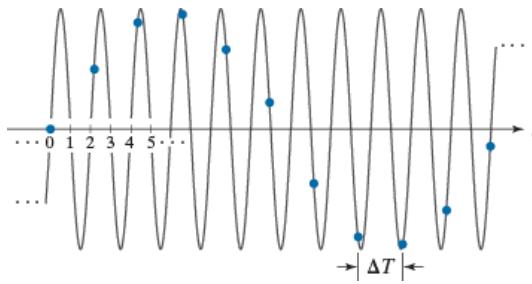


FIGURE 4.11 Illustration of aliasing. The under-sampled function (dots) looks like a sine wave having a frequency much lower than the frequency of the continuous signal. The period of the sine wave is 2 s, so the zero crossings of the horizontal axis occur every second. ΔT is the separation between samples.

2018/10/4

Digital Image Processing

18

Aliasing



No Aliasing

Aliased Image

2018/10/4

Digital Image Processing

19

Temporal Aliasing in Video



https://www.youtube.com/watch?v=QOwzkND_oU

2018/10/4

Digital Image Processing

20

Discrete Fourier Transform (DFT)

- Fourier transform of a sampled band-limited function $\tilde{f}(t)$ is a continuous periodic function $\tilde{F}(\mu)$

$$\begin{aligned}\tilde{F}(\mu) &= \int_{-\infty}^{\infty} \tilde{f}(t) e^{-j2\pi\mu t} dt = \int_{-\infty}^{\infty} f(t) s_{\Delta T}(t) e^{-j2\pi\mu t} dt \\ &= F(\mu) * S(\mu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F(\mu) * \delta(\mu - \frac{n}{\Delta T}) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F(\mu - \frac{n}{\Delta T})\end{aligned}$$

- $\tilde{F}(\mu)$ is continuous and infinite periodic with period $1/\Delta T$
- We only need one period of $\tilde{F}(\mu)$
- We obtain equally spaced sample of $\tilde{F}(\mu)$ from $\mu = 0$ to $\mu = 1/\Delta T$:

$$\tilde{F}(\mu) \rightarrow F_m \text{ with } \mu = m/M\Delta T, \quad m = 0, 1, \dots, M-1$$

2018/10/4

Digital Image Processing

21

Discrete Fourier Transform (DFT)

- Fourier transform $F(\mu)$ of $f(t)$ is defined as

$$F(\mu) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi\mu t} dt$$

- The inverse Fourier Transform is

$$f(t) = \int_{-\infty}^{\infty} F(u) e^{j2\pi\mu t} du$$

- $F(\mu)$ and $f(t)$ are **continuous** functions.

- $f(t)$ sampled $\rightarrow \tilde{f}(t)$ (discrete) $\xrightarrow{\text{FT}}$ $\tilde{F}(\mu)$ (continuous periodic)

- $\tilde{F}(\mu)$ sampled $\rightarrow F_m$ (discrete periodic) $\xrightarrow{\text{DFT}^{-1}}$ f_n (discrete periodic)

- DFT for f_n is

$$F_m = \sum_{n=0}^{M-1} f_n e^{-j2\pi mn/M} \quad m = 0, 1, \dots, M-1$$

- Inverse DFT for F_m is

$$f_n = \frac{1}{M} \sum_{m=0}^{M-1} F_m e^{j2\pi mn/M} \quad n = 0, 1, \dots, M-1$$

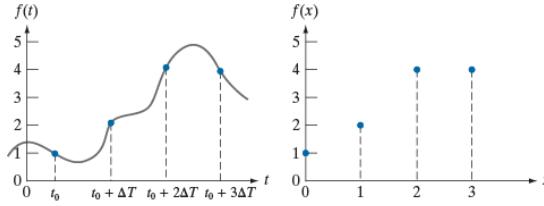
2018/10/4

Digital Image Processing

22

Discrete Fourier Transform (DFT)

- Example** $F(0) = \sum_{n=0}^3 f(n)e^{-j2\pi(0)/4} = \sum_{n=0}^3 f(n) = 1 + 2 + 4 + 4 = 11$
- $F(1) = \sum_{n=0}^3 f(n)e^{-j2\pi(1)n/4} = 1e^0 + 2e^{-j\pi/2} + 4e^{-j\pi} + 4e^{-j3\pi/2} = -3 + 2j$
- $F(2) = \sum_{n=0}^3 f(n)e^{-j2\pi(2)n/4} = -(1 + 0j), \quad F(3) = \sum_{n=0}^3 f(n)e^{-j2\pi(3)n/4} = -(3 + 2j)$
- $f(0) = \frac{1}{4} \sum_{m=0}^3 F(m)e^{j2\pi m(0)/4} = \frac{1}{4} \sum_{m=0}^3 F(m) = \frac{1}{4} [11 - 3 + 2j - 1 - 3 - 2j] = 1$



a b
FIGURE 4.12
(a) A continuous function sampled ΔT units apart.
(b) Samples in the x -domain.
Variable t is continuous, while x is discrete.

2018/10/4

Digital Image Processing

23

Discrete Fourier Transform (DFT)

- Example:** $f(x)$ sampled K points or $f(x)$ sampled $2K$ points
- More samples in **time domain (higher resolution)** \rightarrow Lower resolution (higher freq. interval) in **frequency domain**.
- Scaling property of DFT**
$$f(ax) \Leftrightarrow F(u/a)/|a|$$
- Sampling: $f(x) = f(x_0 + x\Delta x)$, Δx is the **time resolution**
- $f(x) \rightarrow F(u) = F(u_0 + u\Delta u)$, Δu is the **freq. resolution**.
- $\Delta u = 1/(M\Delta x)$

2018/10/4

Digital Image Processing

24

Discrete Fourier Transform (DFT)

- For discrete case (**DFT: Discrete Fourier Transform**)
- $f(x), x = 0, \dots, M - 1$
- $f(x_0), f(x_0 + \Delta x), \dots, f(x_0 + (M - 1)\Delta x)$
 $f(x) = f(x_0 + x\Delta x)$
- $F(u), u = 0, \dots, M - 1$
- $F(u_0), F(u_0 + \Delta u), \dots, F(u_0 + (M - 1)\Delta u)$
 $F(u) = F(u_0 + u\Delta u)$
 $\Delta u = 1/M\Delta x$

2018/10/4

Digital Image Processing

25

Discrete Fourier Transform (DFT)

2-D Impulse

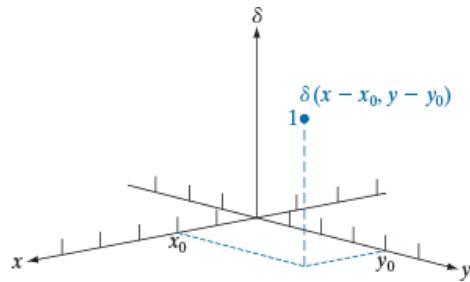


FIGURE 4.13
 2-D unit discrete impulse. Variables x and y are discrete, and δ is zero everywhere except at coordinates (x_0, y_0) , where its value is 1.

2018/10/4

Digital Image Processing

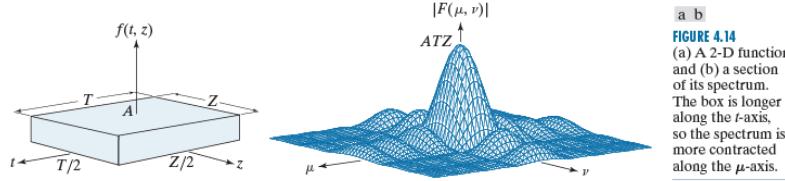
26

Discrete Fourier Transform (DFT)

2-D function and its Fourier Transform

$$\begin{aligned} F(\mu, \nu) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) e^{-j2\pi(\mu t + \nu z)} dt dz \\ &= \int_{-T/2}^{T/2} \int_{-Z/2}^{Z/2} A e^{-j2\pi(\mu t + \nu z)} dt dz = ATZ \left[\frac{\sin(\pi\mu T)}{\pi\mu T} \right] \left[\frac{\sin(\pi\nu Z)}{\pi\nu Z} \right] \end{aligned}$$

$$|F(\mu, \nu)| = ATZ \left| \frac{\sin(\pi\mu T)}{\pi\mu T} \right| \left| \frac{\sin(\pi\nu Z)}{\pi\nu Z} \right|$$



2018/10/4

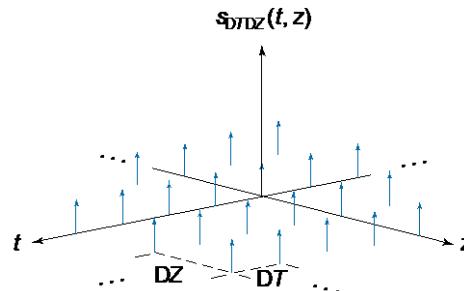
Digital Image Processing

27

Discrete Fourier Transform (DFT)

2-D impulse train

$$s_{\Delta T \Delta Z}(t, z) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \delta(t - m\Delta T, z - n\Delta Z)$$



2018/10/4

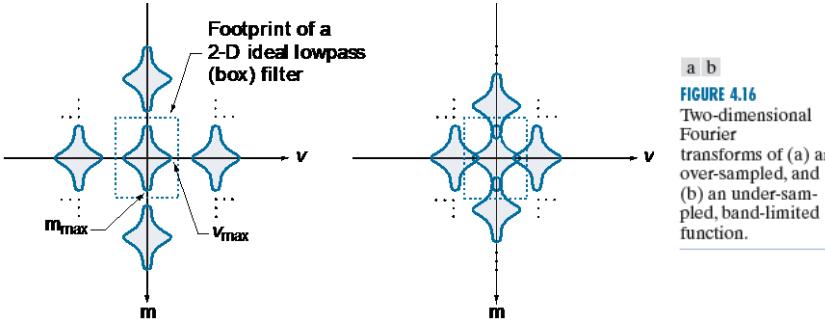
Digital Image Processing

28

Discrete Fourier Transform (DFT)

Extracting $F(\mu, \nu)$ from $\tilde{F}(\mu, \nu)$ without loss if the separation between copies is sufficient

$$\begin{aligned} 1/\Delta T &> 2\mu_{\max} \text{ and } 1/\Delta Z > 2\nu_{\max} \\ \text{or } \Delta T &< \frac{1}{2}\mu_{\max} \text{ and } \Delta Z < \frac{1}{2}\nu_{\max} \end{aligned}$$



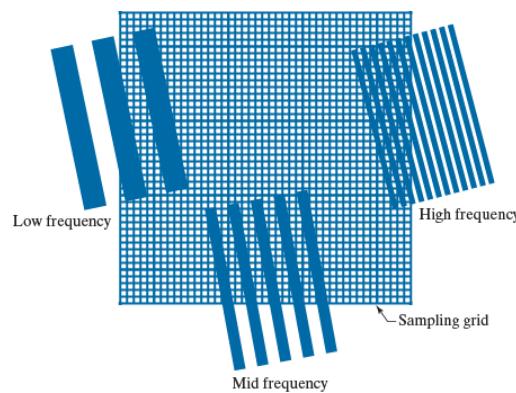
2018/10/4

Digital Image Processing

29

Aliasing in Images

FIGURE 4.17
Various aliasing effects resulting from the interaction between the frequency of 2-D signals and the sampling rate used to digitize them. The regions outside the sampling grid are continuous and free of aliasing.



2018/10/4

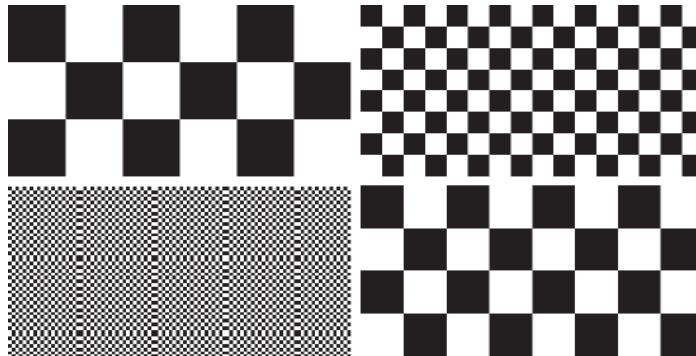
Digital Image Processing

30

Aliasing in Images

a b
c d

FIGURE 4.18
Aliasing. In (a) and (b) the squares are of sizes 16 and 6 pixels on the side. In (c) and (d) the squares are of sizes 0.95 and 0.48 pixels, respectively. Each small square in (c) is one pixel. Both (c) and (d) are aliased. Note how (d) masquerades as a “normal” image.



$0.95 < 1$

$0.48 < 0.5$

2018/10/4

Digital Image Processing

31

Aliasing in Images



a b c

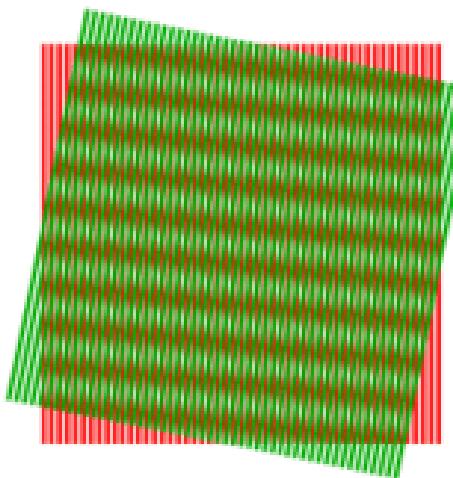
FIGURE 4.19 Illustration of aliasing on resampled natural images. (a) A digital image of size 772×548 pixels with visually negligible aliasing. (b) Result of resizing the image to 33% of its original size by pixel deletion and then restoring it to its original size by pixel replication. Aliasing is clearly visible. (c) Result of blurring the image in (a) with an averaging filter prior to resizing. The image is slightly more blurred than (b), but aliasing is not longer objectionable. (Original image courtesy of the Signal Compression Laboratory, University of California, Santa Barbara.)

2018/10/4

Digital Image Processing

32

Moiré Patterns



2018/10/4

Digital Image Processing

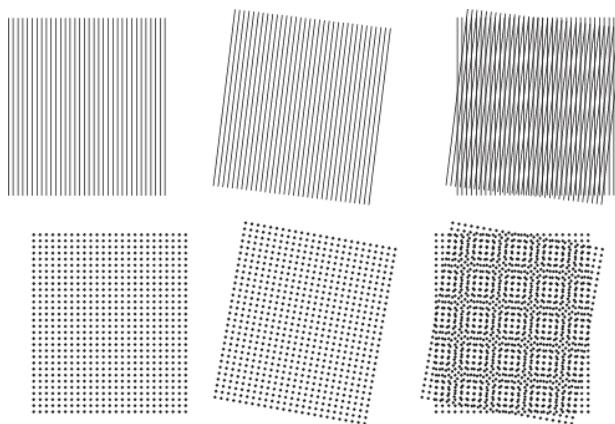
33

Moiré Patterns

Moiré Pattern

a	b	c
d	e	f

FIGURE 4.20
Examples of the moiré effect.
These are vector drawings, not
digitized patterns.
Superimposing
one pattern on the other is analogous
to multiplying the
patterns.



2018/10/4

Digital Image Processing

34

Moiré Patterns in a Halftone Image



FIGURE 4.21

A newspaper image digitized at 75 dpi. Note the moiré-like pattern resulting from the interaction between the $\pm 45^\circ$ orientation of the half-tone dots and the north-south orientation of the sampling elements used to digitized the image.

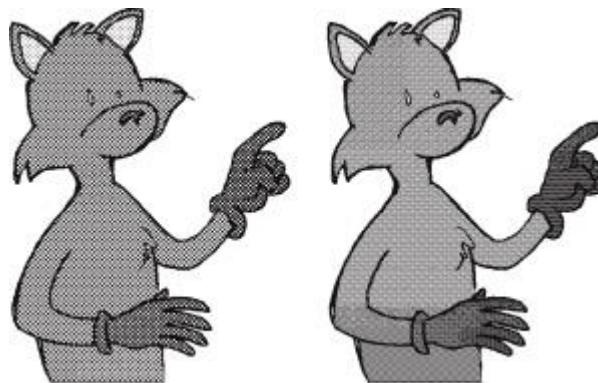
Newspaper of 75 dpi

2018/10/4

Digital Image Processing

35

Moiré Patterns



2018/10/4

Digital Image Processing

36

Moiré Patterns Caused by Aliasing



2018/10/4

Digital Image Processing

37

Properties of 2-D DFT

- 2D-DFT of $f(x, y)$ of size $M \times N$

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

- Inverse 2-D DFT

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

- **Magnitude** and **Phase** of $F(u, v) = R(u, v) + jI(u, v)$

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

$$\phi(u, v) = \tan^{-1}(I(u, v)/R(u, v))$$

- **Power Spectrum**

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

2018/10/4

Digital Image Processing

38

Properties of 2-D DFT

- For real $f(x, y)$
 - $F(u, v)$ is symmetric about the origin
 - $F(u, v) = F^*(-u, -v)$
 - $|F(u, v)| = |F(-u, -v)|$
- Samples in the space domain and frequency domain
 - $f(x_0, y_0), f(x_0 + \Delta x, y_0 + \Delta y), \dots, f(x_0 + (M-1)\Delta x, y_0 + (N-1)\Delta y)$
 - $F(u_0), F(u_0 + \Delta u, v_0 + \Delta v), \dots, F(u_0 + (M-1)\Delta u, v_0 + (N-1)\Delta v)$
 - with the sample intervals as
$$\Delta u = 1/M\Delta x$$

$$\Delta v = 1/N\Delta y$$

2018/10/4

Digital Image Processing

39

Properties of 2-D DFT

- Modulation in the spatial domain

$$f(x, y)(-1)^{x+y}$$
- $F(u, v)$ will be shifted to $(M/2, N/2)$

$$F(u - M/2, v - N/2)$$
- The center of $(u, v), u = 1, \dots, M, v = 1, \dots, N$

$$u_0 = M/2, v_0 = N/2$$
- Average of $f(x, y)$

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

2018/10/4

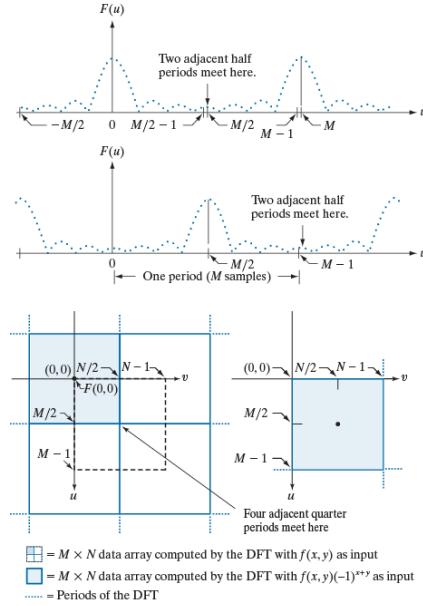
Digital Image Processing

40

Properties of 2-D DFT

FIGURE 4.22
 Centering the Fourier transform.
 (a) A 1-D DFT showing an infinite number of periods.
 (b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^{x+y}$ before computing $F(u)$.
 (c) A 2-D DFT showing an infinite number of periods. The area within the dashed rectangle is the data array, $F(u, v)$, obtained with Eq. (4-67) with an image $f(x, y)$ as the input. This array consists of four quarter periods.
 (d) Shifted array obtained by multiplying $f(x, y)$ by $(-1)^{x+y}$ before computing $F(u, v)$. The data now contains one complete, centered period, as in (b).

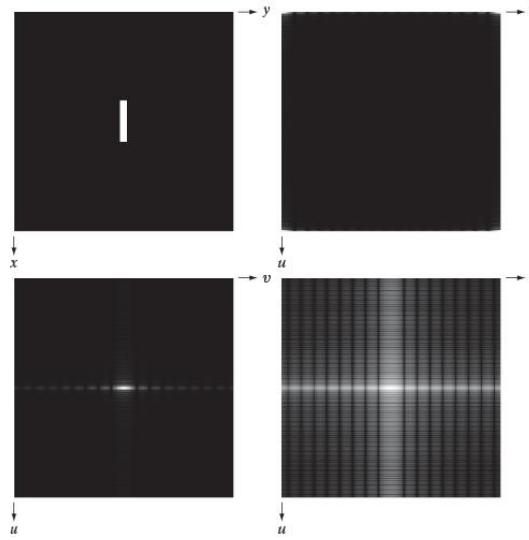
2018/10/4



41

Properties of 2-D DFT

FIGURE 4.23
 (a) Image.
 (b) Spectrum, showing small, bright areas in the four corners (you have to look carefully to see them).
 (c) Centered spectrum.
 (d) Result after a log transformation. The zero crossings of the spectrum are closer in the vertical direction because the rectangle in (a) is longer in that direction. The right-handed coordinate convention used in the book places the origin of the spatial and frequency domains at the top left (see Fig. 2.19).



2018/10/4

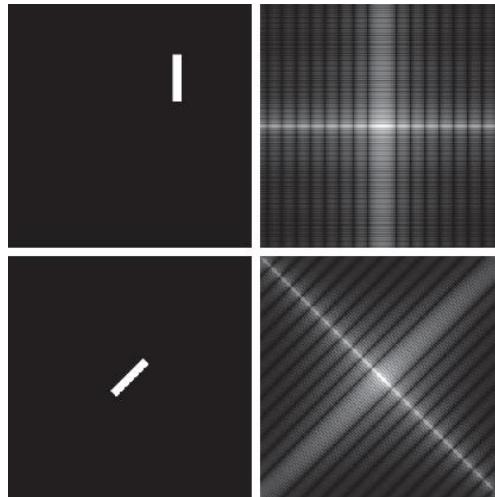
Digital Image Processing

42

Properties of 2-D DFT

a b
c d

FIGURE 4.24
 (a) The rectangle in Fig. 4.23(a) translated.
 (b) Corresponding spectrum.
 (c) Rotated rectangle.
 (d) Corresponding spectrum.
 The spectrum of the translated rectangle is identical to the spectrum of the original image in Fig. 4.23(a).



2018/10/4

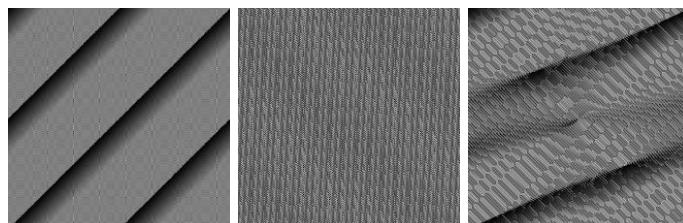
Digital Image Processing

43

Properties of 2-D DFT

a b c

FIGURE 4.25
 Phase angle images of (a) centered, (b) translated, and (c) rotated rectangles.



2018/10/4

Digital Image Processing

44

Properties of 2-D DFT

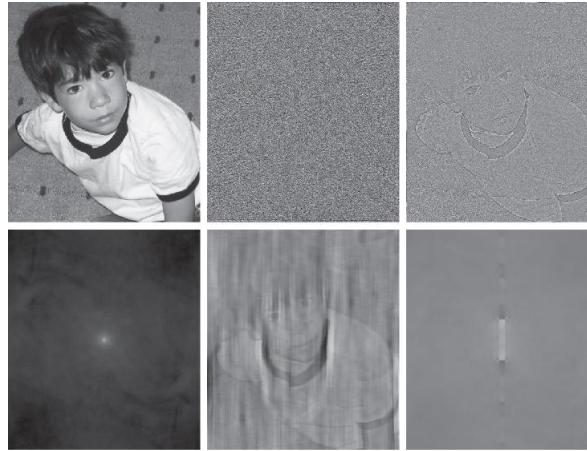


FIGURE 4.26 (a) Boy image. (b) Phase angle. (c) Boy image reconstructed using only its phase angle (all shape features are there, but the intensity information is missing because the spectrum was not used in the reconstruction). (d) Boy image reconstructed using only its spectrum. (e) Boy image reconstructed using its phase angle and the spectrum of the rectangle in Fig. 4.23(a). (f) Rectangle image reconstructed using its phase and the spectrum of the boy's image.

2018/10/4

Digital Image Processing

45

Properties of 2-D DFT

Periodic Convolution

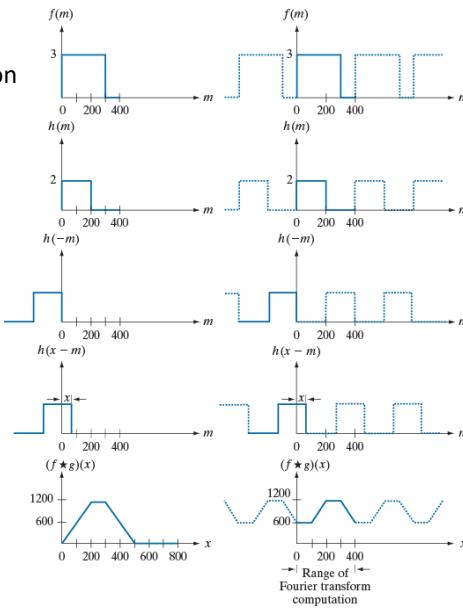


FIGURE 4.27
Left column: Spatial convolution computed with Eq. (3-44), using the approach discussed in Section 3.4. Right column: Circular convolution. The solid line in (j) is the result we would obtain using the DFT, or, equivalently, Eq. (4-48). This erroneous result can be remedied by using zero padding.

2018/10/4

46

Properties of 2-D DFT

TABLE 4.3
Summary of DFT definitions and corresponding expressions.

Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x,y)$	$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(uM+vy)/N}$
2) Inverse discrete Fourier transform (IDFT) of $F(u,v)$	$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(uM+vy)/N}$
3) Spectrum	$ F(u,v) = \left[R^2(u,v) + I^2(u,v) \right]^{1/2} \quad R = \text{Real}(F); I = \text{Imag}(F)$
4) Phase angle	$\phi(u,v) = \tan^{-1} \left[\frac{I(u,v)}{R(u,v)} \right]$
5) Polar representation	$F(u,v) = F(u,v) e^{j\theta(u,v)}$
6) Power spectrum	$P(u,v) = F(u,v) ^2$
7) Average value	$\bar{f} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) = \frac{1}{MN} F(0,0)$
8) Periodicity (k_1 and k_2 are integers)	$\begin{aligned} F(u,v) &= F(u+k_1 M, v) = F(u, v+k_2 N) \\ &= F(u+k_1, v+k_2 N) \\ f(x,y) &= f(x+k_1 M, y) = f(x, y+k_2 N) \\ &= f(x+k_1, y+k_2 N) \end{aligned}$
9) Convolution	$(f * h)(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x-m, y-n)$
10) Correlation	$(f \diamond h)(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m,n) h(x+m, y+n)$
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 4.1.1.
12) Obtaining the IDFT using a DFT algorithm	$MN ^* (x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u,v) e^{-j2\pi(uM+vy)/N}$ This equation indicates that inputting $F^*(u,v)$ into an algorithm that computes the forward transform (right side of above equation) yields $MN ^*(x,y)$. Taking the complex conjugate and dividing by MN gives the desired inverse. See Section 4.11.

2018/10/4

47

Properties of 2-D DFT

TABLE 4.4
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 continuous expressions.

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)	$\begin{aligned} f(x,y)e^{j2\pi(u_0M+vy_0)/N} &\Leftrightarrow F(u-u_0, v-v_0) \\ f(x-x_0, y-y_0) &\Leftrightarrow F(u,v)e^{j2\pi(uu_0/M+vy_0/N)} \end{aligned}$
4) Translation to center of the frequency rectangle, $(M/2, N/2)$	$\begin{aligned} f(x,y)(-1)^{xy} &\Leftrightarrow F(u-M/2, v-N/2) \\ f(x-M/2, y-N/2) &\Leftrightarrow F(u,v)(-1)^{xy} \end{aligned}$
5) Rotation	$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$ $r = \sqrt{x^2 + y^2} \quad \theta = \tan^{-1}(y/x) \quad \omega = \sqrt{u^2 + v^2} \quad \varphi = \tan^{-1}(v/u)$
6) Convolution theorem [†]	$\begin{aligned} f * h(x,y) &\Leftrightarrow (F * H)(u,v) \\ (f * h)(x,y) &\Leftrightarrow (1/MN)[(F * H)(u,v)] \end{aligned}$
7) Correlation theorem [†]	$\begin{aligned} (f \diamond h)(x,y) &\Leftrightarrow (F^* * H)(u,v) \\ (f^* * h)(x,y) &\Leftrightarrow (1/MN)[(F \diamond H)(u,v)] \end{aligned}$
8) Discrete unit impulse	$\delta(x,y) \Leftrightarrow 1 \quad 1 \Leftrightarrow MN\delta(u,v)$
9) Rectangle	$\text{rect}[a,b] \Leftrightarrow ab \frac{\sin(\pi ua)}{(\pi ua)} \frac{\sin(\pi vb)}{(\pi vb)} e^{-j\pi(uv+ab)}$
10) Sine	$\sin(2\pi u_0 x/M + 2\pi v_0 y/N) \Leftrightarrow \frac{1}{2} [\delta(u+u_0, v+v_0) - \delta(u-u_0, v-v_0)]$
11) Cosine	$\cos(2\pi u_0 x/M + 2\pi v_0 y/N) \Leftrightarrow \frac{1}{2} [\delta(u+u_0, v+v_0) + \delta(u-u_0, v-v_0)]$
12) Differentiation (the expressions on the right assume that $f(\pm\infty, \pm\infty) = 0$.)	$\begin{aligned} \left(\frac{\partial}{\partial t} \right)^m \left(\frac{\partial}{\partial z} \right)^n f(t,z) &\Leftrightarrow (j2\pi\mu)^m (j2\pi\nu)^n F(\mu,\nu) \\ \frac{\partial^2 f(t,z)}{\partial t^2} &\Leftrightarrow (j2\pi\mu)^2 F(\mu,\nu); \quad \frac{\partial^2 f(t,z)}{\partial z^2} \Leftrightarrow (j2\pi\nu)^2 F(\mu,\nu) \end{aligned}$
13) Gaussian	$A2\pi\sigma^2 e^{-j2\pi(u^2\sigma^2 + v^2\sigma^2)} \Leftrightarrow Ae^{-j(u^2 + v^2)/2\sigma^2} \quad (A \text{ is a constant})$

[†] Assumes that $f(x,y)$ and $h(x,y)$ have been properly padded. Convolution is associative, commutative, and distributive. Correlation is distributive (see Table 2.5). The products are elementwise products (see Section 2.6).

2018/10/4

48

Filtering in the Frequency Domain

Filtering in Frequency domain steps:

- 1) Zero-padding: $M \times N \rightarrow P \times Q$
- 2) Multiply the input image $f(x, y)$ by $(-1)^{x+y}$. Compute DFT of the input image and get

$$F(u, v) = \mathcal{F}\{f(x, y)(-1)^{x+y}\}$$
- 3) Multiply $F(u, v)$ by a filter function $H(u, v)$

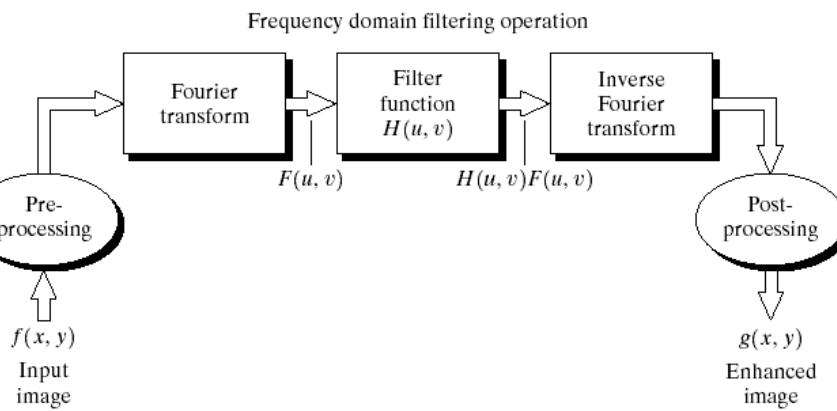
$$G(u, v) = F(u, v)H(u, v)$$
- 4) Compute the inverse DFT of $G(u, v)$, i.e., $\mathcal{F}^{-1}\{G(u, v)\}$
- 5) Obtain the real part of the $g(x, y)$
- 6) Multiply $g(x, y)$ with $(-1)^{x+y}$

2018/10/4

Digital Image Processing

49

Filtering in the Frequency Domain

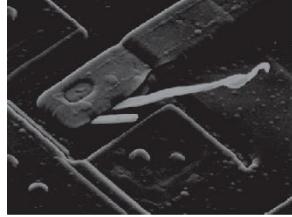
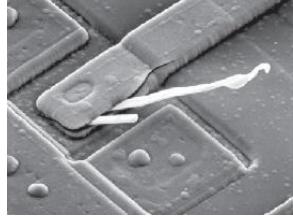


2018/10/4

Digital Image Processing

50

Filtering in the Frequency Domain



a b

FIGURE 4.28 (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.)

FIGURE 4.29

Result of filtering the image in Fig. 4.28(a) with a filter transfer function that sets to 0 the dc term, $F(P/2, Q/2)$, in the centered Fourier transform, while leaving all other transform terms unchanged.

2018/10/4

Digital Image Processing

51

Filtering in the Frequency Domain

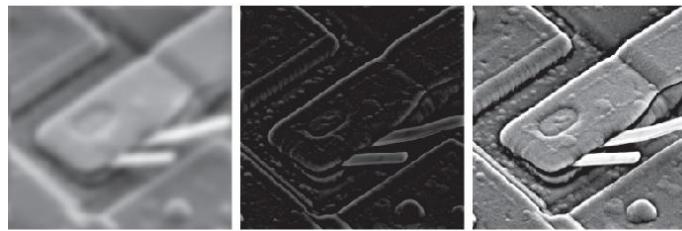
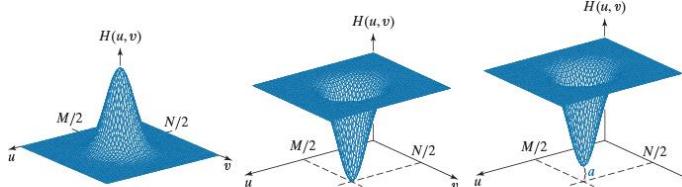
a b c
d e f

FIGURE 4.30 Top row: Frequency domain filter transfer functions of (a) a lowpass filter, (b) a highpass filter, and (c) an offset highpass filter. Bottom row: Corresponding filtered images obtained using Eq. (4-104). The offset in (c) is $a = 0.85$, and the height of $H(u, v)$ is 1. Compare (f) with Fig. 4.28(a).

2018/10/4

Digital Image Processing

52

Filtering in the Frequency Domain



a b c

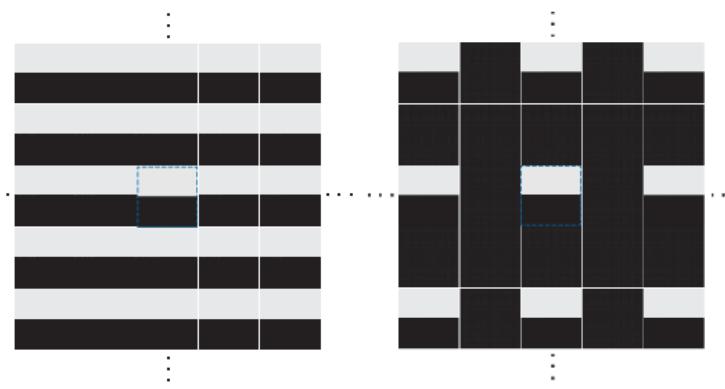
FIGURE 4.31 (a) A simple image. (b) Result of blurring with a Gaussian lowpass filter without padding. (c) Result of lowpass filtering with zero padding. Compare the vertical edges in (b) and (c).

2018/10/4

Digital Image Processing

53

Filtering in the Frequency Domain



a b

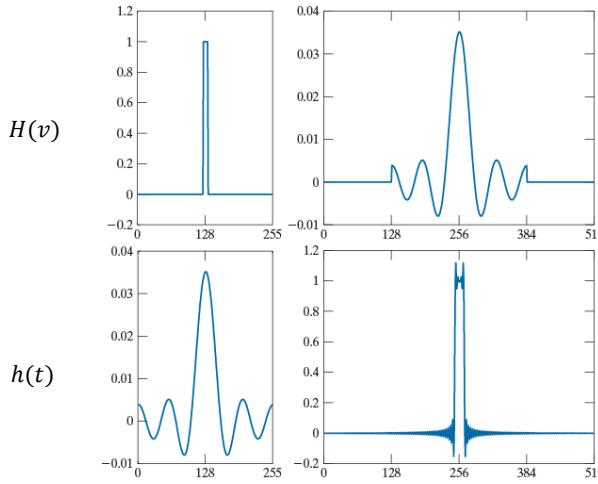
FIGURE 4.32 (a) Image periodicity without image padding. (b) Periodicity after padding with 0's (black). The dashed areas in the center correspond to the image in Fig. 4.31(a). Periodicity is inherent when using the DFT. (The thin white lines in both images are superimposed for clarity; they are not part of the data.)

2018/10/4

Digital Image Processing

54

Filtering in the Frequency Domain



a c

b d

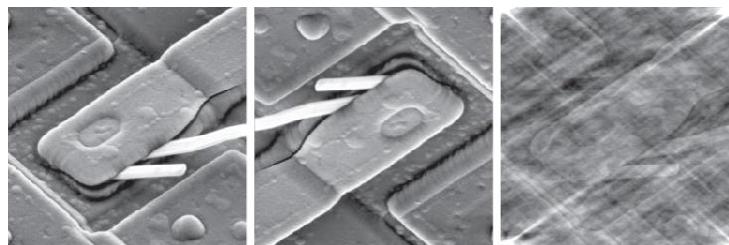
FIGURE 4.33
(a) Filter transfer function specified in the (centered) frequency domain.
(b) Spatial representation (filter kernel) obtained by computing the IDFT of (a).
(c) Result of padding (b) to twice its length (note the discontinuities).
(d) Corresponding filter in the frequency domain obtained by computing the DFT of (c). Note the ringing caused by the discontinuities in (c). Part (b) of the figure is below (a), and (d) is below (c).

2018/10/4

Digital Image Processing

55

Filtering in the Frequency Domain



a b c

FIGURE 4.34 (a) Original image. (b) Image obtained by multiplying the phase angle array by -1 in Eq. (4-86) and computing the IDFT. (c) Result of multiplying the phase angle by 0.25 and computing the IDFT. The magnitude of the transform, $|F(u,v)|$, used in (b) and (c) was the same.

$$f(x, y) \rightarrow \text{DFT} \rightarrow F(u, v)$$

$$F(u, v) = R(u, v) + jI(u, v)$$

$$F'(u, v) = |F(u, v)| e^{j0.25\phi(u, v)}$$

$$F'(u, v) \rightarrow \text{IDFT} \rightarrow f'(x, y)$$

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

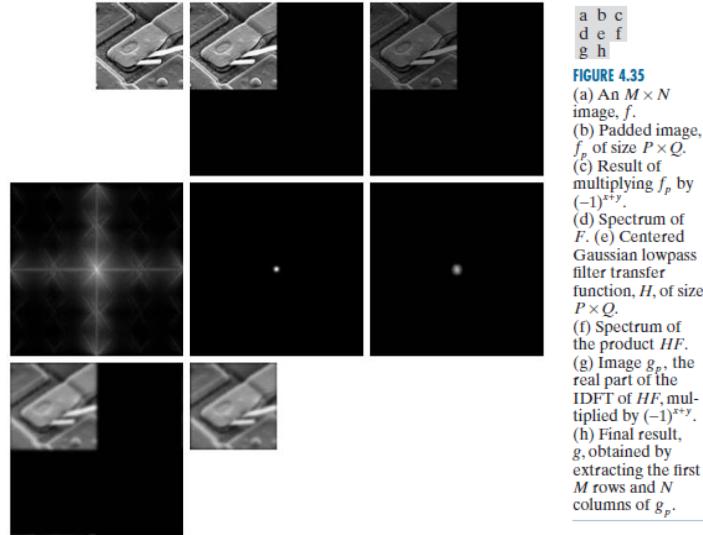
$$\phi(u, v) = \tan^{-1}(I(u, v)/R(u, v))$$

2018/10/4

Digital Image Processing

56

Filtering in the Frequency Domain



2018/10/4

Digital Image Processing

.7

Filtering in the Frequency Domain

- Convolution in the **time domain**

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

- $f(x, y) * h(x, y) \Leftrightarrow F(u, v)H(u, v)$
i.e., $f(x, y) * h(x, y) = \mathcal{F}^{-1}\{F(u, v)H(u, v)\}$



2018/10/4

Digital Image Processing

58

Filtering in the Frequency Domain

- An impulse of strength A , located at coordinates (x_0, y_0) , is denoted as $A\delta(x - x_0, y - y_0)$
- A **unit impulse** is defined as $\delta(x, y)$. i.e., $\delta(x, y) = 1$ only when $x = 0, y = 0$, $\delta(x, y) = 0$ otherwise.
- For function $s(x, y)$
 - (a) sample at $(0,0)$ is denoted as $\sum_x \sum_y s(x, y)\delta(x, y) = s(0,0)$
 - (b) sample at (x_0, y_0) , is $\sum_x \sum_y s(x, y)\delta(x - x_0, y - y_0) = s(x_0, y_0)$,
- The Fourier transform of $\delta(x, y)$ is

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

2018/10/4

Digital Image Processing

59

Filtering in the Frequency Domain

- If we let $f(x, y) = \delta(x, y)$ then the **impulse response** is

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

$$\delta(x, y) * h(x, y) \Leftrightarrow \mathcal{F}\{\delta(x, y)\} H(u, v) \text{ or } h(x, y) \Leftrightarrow H(u, v)$$

$h(x, y)$: impulse response of spatial-domain filter

$H(u, v)$: impulse response of frequency-domain filter

- Gaussian filters – **lowpass filtering**

$$H(u) = Ae^{-u^2/2\sigma^2} \quad h(x) = \sqrt{2\pi}\sigma Ae^{-2\pi u^2\sigma^2 x^2}$$

- Difference of two Gaussian filters – **highpass filtering**

$$H(u) = Ae^{-u^2/2\sigma_1^2} - Be^{-u^2/2\sigma_2^2} \text{ where } A \geq B, \sigma_1 > \sigma_2$$

- **The spatial filter**

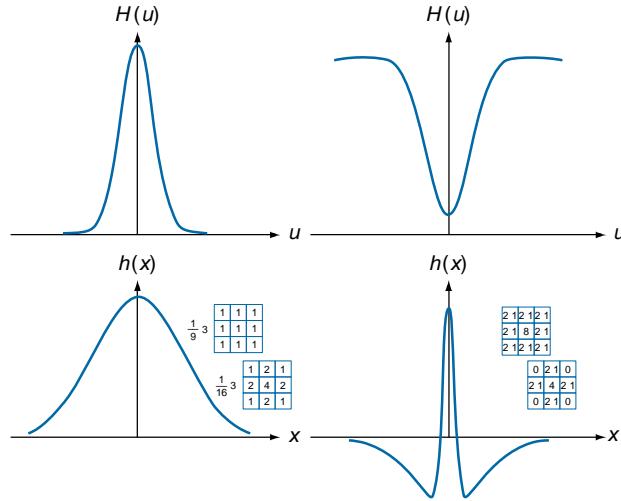
$$h(x) = \sqrt{2\pi}\sigma_1 Ae^{-2\pi u^2\sigma_1^2 x^2} - \sqrt{2\pi}\sigma_2 Be^{-2\pi u^2\sigma_2^2 x^2}$$

2018/10/4

Digital Image Processing

60

Filtering in the Frequency Domain



a c

b d

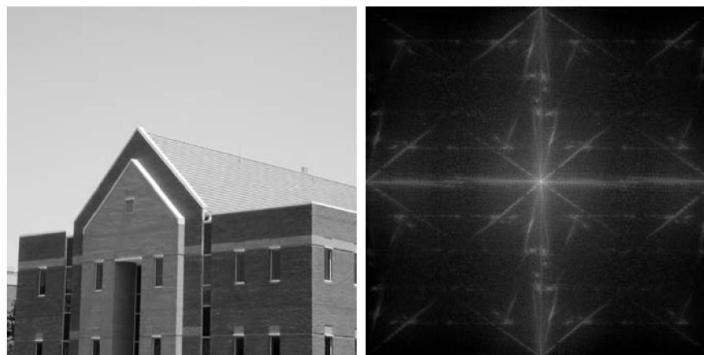
FIGURE 4.36
 (a) A 1-D Gaussian lowpass transfer function in the frequency domain.
 (b) Corresponding kernel in the spatial domain.
 (c) Gaussian highpass transfer function in the frequency domain.
 (d) Corresponding kernel. The small 2-D kernels shown are kernels we used in Chapter 3.

2018/10/4

Digital Image Processing

61

Filtering in the Frequency Domain



a b

FIGURE 4.37
 (a) Image of a building, and
 (b) its Fourier spectrum.

2018/10/4

Digital Image Processing

62

Filtering in the Frequency Domain

a b
c d

FIGURE 4.38
 (a) A spatial kernel and perspective plot of its corresponding frequency domain filter transfer function.
 (b) Transfer function shown as an image.
 (c) Result of filtering Fig. 4.37(a) in the frequency domain with the transfer function in (b).
 (d) Result of filtering the same image in the spatial domain with the kernel in (a). The results are identical.

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



2018/10/4

Digital Image Processing

63

Smoothing Using Freq-Domain Filters

- Frequency-Domain Filtering:

$$G(u, v) = H(u, v)F(u, v)$$
- Filter $H(u, v)$
 - Ideal filter
 - Butterworth filter
 - Gaussian Filter

2018/10/4

Digital Image Processing

64

Ideal Lowpass Filters

- $H(u, v)$ with **sharp cut-off** at **cut-off frequency** D_0 , i.e.,

$$\begin{aligned} H(u, v) &= 1 \text{ if } D(u, v) \leq D_0 \\ &= 0 \text{ if } D(u, v) > D_0 \end{aligned}$$
- For image of size $P \times Q$, the center is at $(u, v) = (P/2, Q/2)$.
The distance from any point to the center is

$$D(u, v) = [(u - P/2)^2 + (v - Q/2)^2]^{1/2}$$
- **Cut off frequency** is D_0
- Total power:

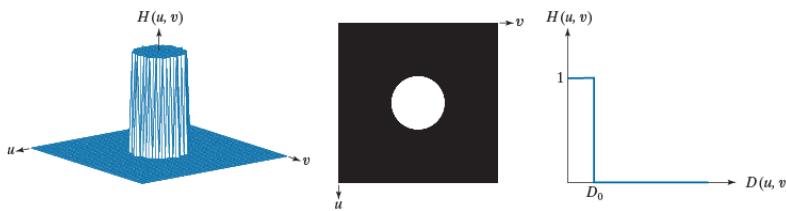
$$P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u, v)$$

2018/10/4

Digital Image Processing

65

Ideal Lowpass Filters



a b c

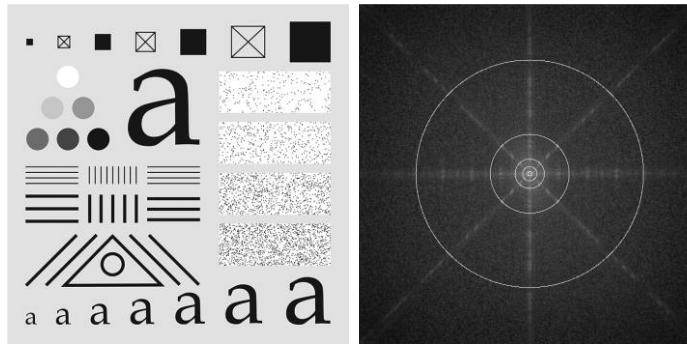
FIGURE 4.39 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross section.

2018/10/4

Digital Image Processing

66

Ideal Lowpass Filters



a b

FIGURE 4.40 (a) Test pattern of size 688×688 pixels, and (b) its spectrum. The spectrum is double the image size as a result of padding, but is shown half size to fit. The circles have radii of 10, 30, 60, 160, and 460 pixels with respect to the full-size spectrum. The radii enclose 86.9, 92.8, 95.1, 97.6, and 99.4% of the padded image power, respectively.

2018/10/4

Digital Image Processing

67

Ideal Lowpass Filters

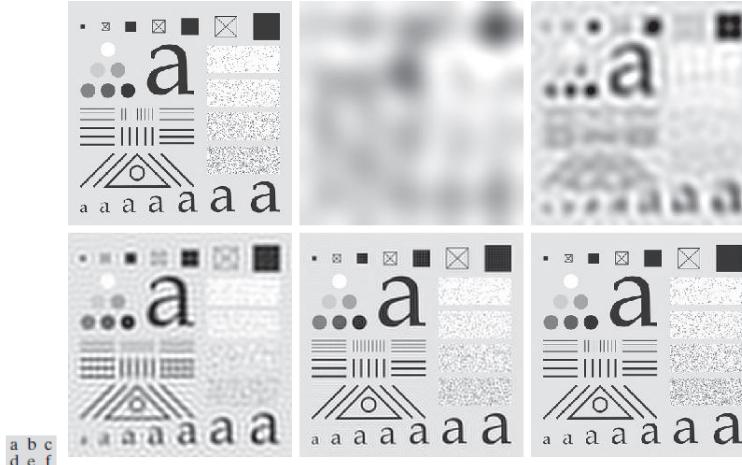


FIGURE 4.41 (a) Original image of size 688×688 pixels. (b)–(f) Results of filtering using ILPPs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.40(b). The power removed by these filters was 13.1, 7.2, 4.9, 2.4, and 0.6% of the total, respectively. We used mirror padding to avoid the black borders characteristic of zero padding, as illustrated in Fig. 4.31(c).

2018/10/4

Digital Image Processing

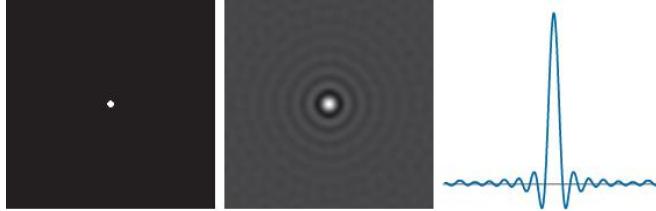
68

Ideal Lowpass Filters

a b c

FIGURE 4.42

(a) Frequency domain ILPF transfer function.
 (b) Corresponding spatial domain kernel function.
 (c) Intensity profile of a horizontal line through the center of (b).



2018/10/4

Digital Image Processing

69

Butterworth Lowpass Filter (BLPF)

- The magnitude-square response of an n -th order analog lowpass **Butterworth filter** is given by

$$|H_c(j\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2n}}$$

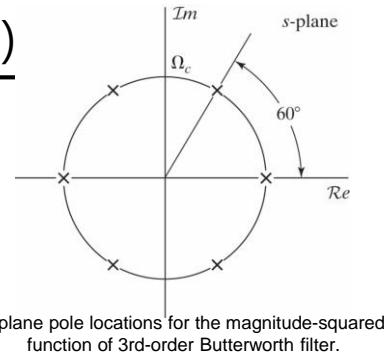
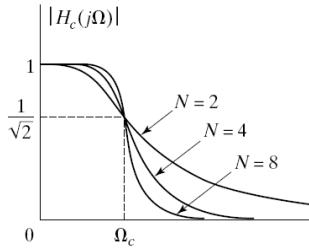
- First $2n - 1$ derivatives of $|H_c(j\Omega)|^2$ at $\Omega = 0$ are equal to zero
- The Butterworth lowpass filter thus is said to have a **maximally-flat magnitude** at $\Omega = 0$
- Gain in dB is $G(\Omega) = 10\log_{10}|H_c(j\Omega)|^2$
- As $G(0) = 0$ and $G(\Omega_c) = 10\log_{10}(0.5) = -3.0103 \cong -3dB$
- Ω_c is called the **3-dB cutoff frequency**

2018/10/4

Digital Image Processing

70

Butterworth LPF (BLPF)



$$H_c(s)H_c(-s) = |H_c(j\Omega)|^2 \Big|_{\Omega=s/j} = \frac{1}{1+(s/j\Omega_c)^{2n}} = \frac{(j\Omega_c)^{2n}}{s^{2n} + (j\Omega_c)^{2n}}$$

$$|H_c(j\Omega)|^2 = \frac{1}{1+(j\Omega/j\Omega_c)^{2n}}, \quad \Omega_c = 3 \text{ dB frequency for } |H_c(j\Omega_c)|^2 = \frac{1}{2}$$

$$p_k = (-1)^{1/2n} (j\Omega_c) = (e^{j(2k-1)\pi})^{1/2n} (j\Omega_c) = \Omega_c e^{(j\pi/2n)(2k+n-1)}, \quad k = 0, 1, \dots, 2n-1$$

$$\therefore H_c(s) = \frac{\Omega_c^n}{\prod_{\text{LHP poles}} (s - p_k)}$$

2018/10/4

Digital Image Processing

71

Butterworth LPF (BLPF)

- Butterworth filter has no sharp cutoff

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

- At cutoff frequency D_0 : $H(u, v) = 0.5$

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

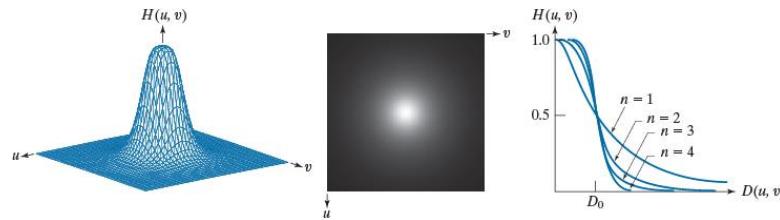
Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = \frac{1}{1+[D(u, v)/D_0]^{2n}}$	$H(u, v) = e^{-D^2(u,v)/2D_0^2}$

2018/10/4

Digital Image Processing

72

Butterworth LPF (BLPF)



a b c

FIGURE 4.45 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross sections of BLPFs of orders 1 through 4.

2018/10/4

Digital Image Processing

73

Butterworth LPF (BLPF)

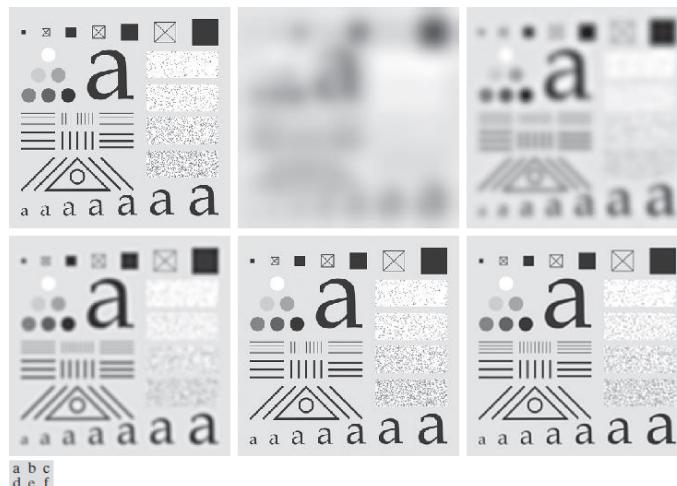


FIGURE 4.46 (a) Original image of size 688×688 pixels. (b)–(f) Results of filtering using BLPFs with cutoff frequencies at the radii shown in Fig. 4.40 and $n = 2.25$. Compare with Figs. 4.41 and 4.44. We used mirror padding to avoid the black borders characteristic of zero padding.

2018/10/4

74

Butterworth LPF (BLPF)

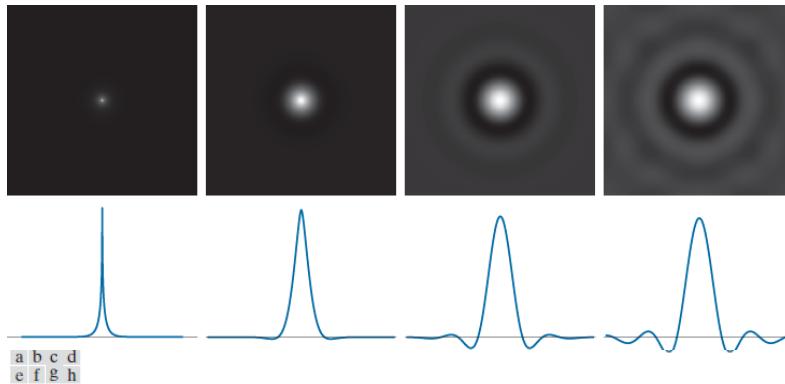


FIGURE 4.47 (a)–(d) Spatial representations (i.e., spatial kernels) corresponding to BLPF transfer functions of 1000×1000 pixels, cut-off frequency of 5, and order 1, 2, 5, and 20, respectively. (e)–(h) Corresponding intensity profiles through the center of the filter functions.

2018/10/4

Digital Image Processing

75

Gaussian Lowpass Filter (GLPF)

- Gaussian low-pass filter (GLPF)

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

- Let $\sigma = D_0$

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

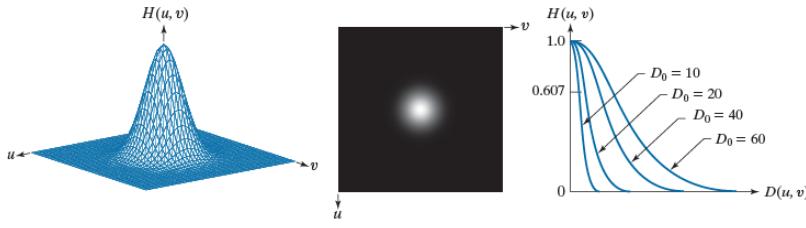
- When $D(u, v) = D_0$, $H(u, v) = 0.607$

2018/10/4

Digital Image Processing

76

Gaussian Lowpass Filter (GLPF)



a b c

FIGURE 4.43 (a) Perspective plot of a GLPF transfer function. (b) Function displayed as an image. (c) Radial cross sections for various values of D_0 .

2018/10/4

Digital Image Processing

77

Gaussian Lowpass Filter (GLPF)

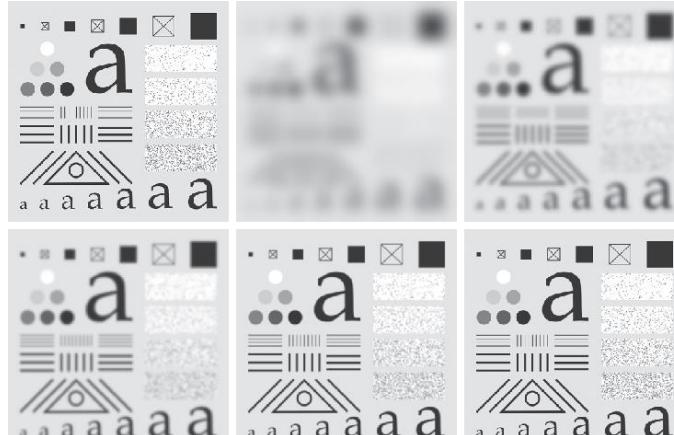


FIGURE 4.44 (a) Original image of size 688 × 688 pixels. (b)–(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in Fig. 4.40. Compare with Fig. 4.41. We used mirror padding to avoid the black borders characteristic of zero padding.

2018/10/4

78

Gaussian Lowpass Filter (GLPF)

a b

FIGURE 4.48

(a) Sample text of low resolution (note the broken characters in the magnified view). (b) Result of filtering with a GLPF, showing that gaps in the broken characters 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.



2018/10/4

Digital Image Processing

79

Gaussian Lowpass Filter (GLPF)



a b c

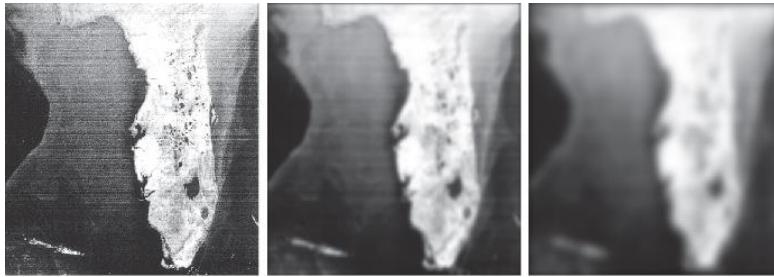
FIGURE 4.49 (a) Original 785 × 732 image. (b) Result of filtering using a GLPF with $D_0 = 150$. (c) Result of filtering using a GLPF with $D_0 = 130$. Note the reduction in fine skin lines in the magnified sections in (b) and (c).

2018/10/4

Digital Image Processing

80

Gaussian Lowpass Filter (GLPF)



a b c

FIGURE 4.50 (a) 808×754 satellite image showing prominent horizontal scan lines. (b) Result of filtering using a GLPF with $D_0 = 50$. (c) Result of using a GLPF with $D_0 = 20$. (Original image courtesy of NOAA.)

2018/10/4

Digital Image Processing

81

Sharpening Using Frequency-Domain Filter

- Highpass filtering:

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

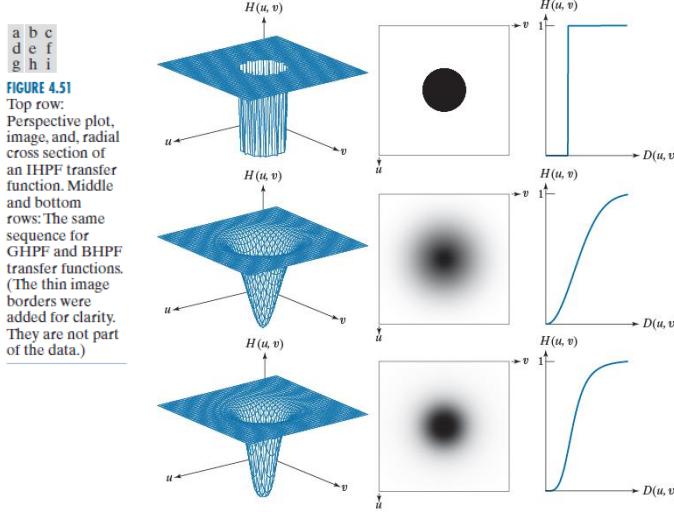
- Given a lowpass filter $H_{LP}(u, v)$, find the **spatial representation** of the highpass filter
 - (1) Compute the inverse DFT of $H_{LP}(u, v)$
 - (2) Multiply the real part of the result with $(-1)^{x+y}$

2018/10/4

Digital Image Processing

82

Sharpening Using Frequency-Domain Filter

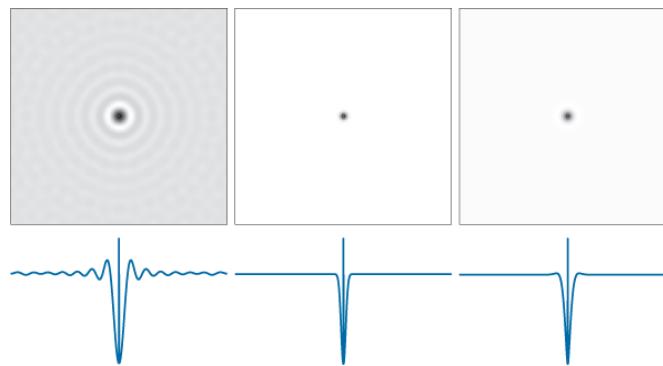


2018/10/4

Digital Image Processing

83

Sharpening Using Frequency-Domain Filter



a b c
d e f

FIGURE 4.52 (a)–(c): Ideal, Gaussian, and Butterworth highpass spatial kernels obtained from IHPF, GHPF, and BHPF frequency-domain transfer functions. (The thin image borders are not part of the data.) (d)–(f): Horizontal intensity profiles through the centers of the kernels.

2018/10/4

Digital Image Processing

84

Ideal Highpass Filter (IHPF)

- $H(u, v) = 0$ if $D(u, v) \leq D_0$
 $= 1$ if $D(u, v) > D_0$
- The center is at $(u, v) = (P/2, Q/2)$

$$D(u, v) = [(u-P/2)^2 + (v-Q/2)^2]^{1/2}$$
- Cut off frequency is D_0

2018/10/4

Digital Image Processing

85

Gaussian Highpass Filter (GHPF)

- Gaussian Highpass filter (GHPF)

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

$$H(u, v) = 1 - e^{-D^2(u, v)/2D_0^2}$$
- When $D(u, v) = D_0$, $H(u, v) = 0.607$

2018/10/4

Digital Image Processing

86

Butterworth Highpass Filter (BHPF)

- Butterworth filter has no sharp cutoff

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

- At cutoff frequency D_0 : $H(u, v) = 0.5$

TABLE 4.6

Highpass filter transfer functions. D_0 is the cutoff frequency and n is the order of the Butterworth transfer function.

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

Comparison of Highpass Filters

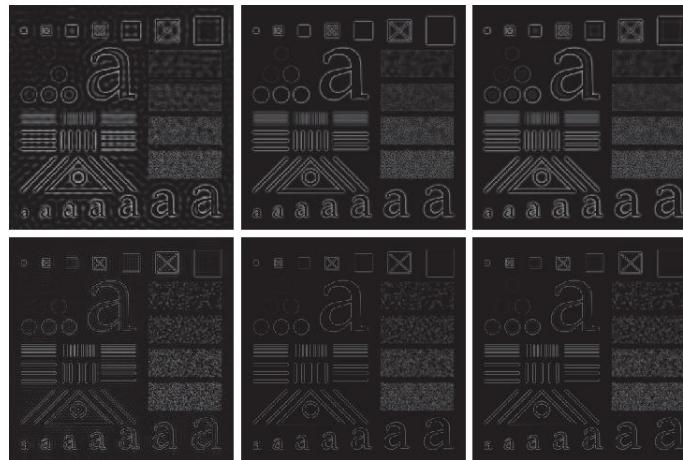
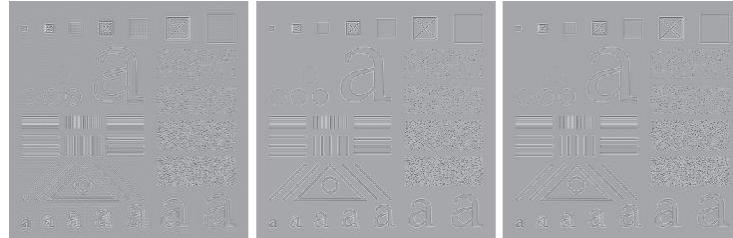


FIGURE 4.53 Top row: The image from Fig. 4.40(a) filtered with IHPF, GHPF, and BHPF transfer functions using $D_0 = 60$ in all cases ($n = 2$ for the BHPF). Second row: Same sequence, but using $D_0 = 160$.

Comparison of Highpass Filters



a b c

FIGURE 4.54 The images from the second row of Fig. 4.53 scaled using Eqs. (2-31) and (2-32) to show both positive and negative values.

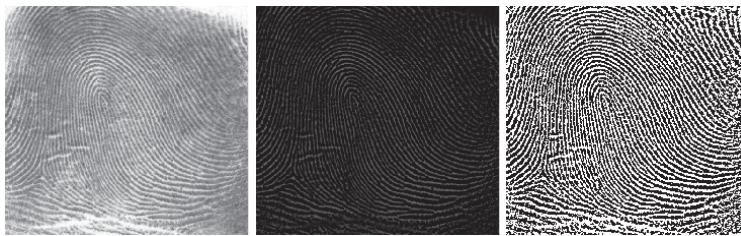
2018/10/4

Digital Image Processing

89

Comparison of Highpass Filters

Butterworth highpass filter of order 4 with cut-off frequency of 50.



a b c

FIGURE 4.55 (a) Smudged thumbprint. (b) Result of highpass filtering (a). (c) Result of thresholding (b). (Original image courtesy of the U.S. National Institute of Standards and Technology.)

2018/10/4

Digital Image Processing

90

Laplacian in the Frequency Domain

- Fourier property:

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

$$\begin{aligned} \mathcal{F}\left[\frac{d^n f(x, y)}{dx^n} + \frac{d^n f(x, y)}{dy^n}\right] &= (ju)^n F(u, v) + (jv)^n F(u, v) \\ &= -(u^2 + v^2) F(u, v), \quad \text{if } n = 2 \end{aligned}$$

$$\mathcal{F}[\nabla^2 f(x, y)] = -(u^2 + v^2) F(u, v) = H(u, v) F(u, v)$$

- So, $H(u, v) = -(u^2 + v^2)$
- If $F(u, v)$ has been centered by $f(x, y)(-1)^{x+y}$ then
 $H(u, v) = -[(u - P/2)^2 + (v - Q/2)^2]$

2018/10/4

Digital Image Processing

91

Laplacian in the Frequency Domain

- Laplacian filtering

$$\nabla^2 f(x, y) = \mathfrak{F}^{-1}\{-[(u - P/2)^2 + (v - Q/2)^2] \cdot F(u, v)\}$$

- Enhanced image: $g(x, y) = f(x, y) - \nabla^2 f(x, y)$

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

$$H(u, v) = 1 - [(u - P/2)^2 + (v - Q/2)^2]$$

- $g(x, y) = \mathfrak{F}^{-1}\{[1 - ((u - P/2)^2 + (v - Q/2)^2)] \cdot F(u, v)\}$

2018/10/4

Digital Image Processing

92

Laplacian in the Frequency Domain

a b

FIGURE 4.56
 (a) Original,
 blurry image.
 (b) Image
 enhanced using
 the Laplacian in
 the frequency
 domain.
 Compare with
 Fig. 3.52(d).
 (Original image
 courtesy of
 NASA.)



2018/10/4

Digital Image Processing

93

Unsharp Masking High-Boost Filtering & High Frequency Emphasis Filtering

- Highpass filtered image: $f_{HP}(x, y) = f(x, y) - f_{LP}(x, y)$
- High-boost image: $f_{HB}(x, y) = Af(x, y) - f_{LP}(x, y)$
 or $f_{HB}(x, y) = (A - 1)f(x, y) + f_{HP}(x, y)$
- In Freq. Domain: $F_{HP}(u, v) = F(u, v) - F_{LP}(u, v)$
 The highpass filter: $H_{HP}(u, v) = 1 - H_{LP}(u, v)$
 The high-boost filter: $H_{HB}(u, v) = (A - 1) + H_{HP}(u, v), A \geq 1$
- High frequency emphasis filter:

$$H_{HFE}(u, v) = k_1 + k_2 H_{HP}(u, v) \text{ where } k_1 \geq 0, k_2 > k_1$$

$$k_1 = 0.25 \sim 0.5, k_2 = 1.5 \sim 2.0$$

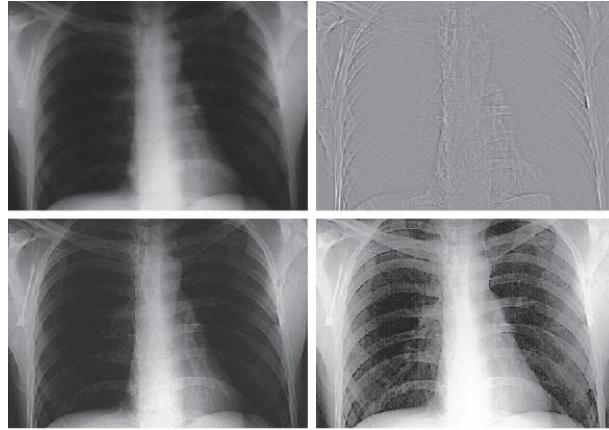
2018/10/4

Digital Image Processing

94

Unsharp Masking High-Boost Filtering & High Frequency Emphasis Filtering

FIGURE 4.57
 (a) A chest X-ray.
 (b) Result of filtering with a GHPF function.
 (c) Result of high-frequency-emphasis filtering using the same GHPF.
 (d) Result of performing histogram equalization on (c).
 (Original image courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)



2018/10/4

Digital Image Processing

95

Homomorphic Filtering

- Improve image by simultaneous **gray-level range compression** and **contrast enhancement**

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

where $i(x, y)$ = **illumination**, $r(x, y)$ = **reflectance**

$i(x, y)$ and $r(x, y)$ are not separable

- Define $z(x, y) = \ln\{f(x, y)\} = \ln\{i(x, y)\} + \ln\{r(x, y)\}$
 $\Im\{z(x, y)\} = \Im\{\ln f(x, y)\} = \Im\{\ln i(x, y)\} + \Im\{\ln r(x, y)\}$
 or $Z(u, v) = F_i(u, v) + F_r(u, v)$

- Applying filter $H(u, v)$:

$$\begin{aligned} S(u, v) &= H(u, v)Z(u, v) \\ &= H(u, v)F_i(u, v) + H(u, v)F_r(u, v) \end{aligned}$$

2018/10/4

Digital Image Processing

96

Homomorphic Filtering

- $$\begin{aligned} s(x, y) &= \mathfrak{J}^{-1}\{S(u, v)\} \\ &= \mathfrak{J}^{-1}\{H(u, v)Fi(u, v)\} + \mathfrak{J}^{-1}\{H(u, v)Fr(u, v)\} \\ &= i'(x, y) + r'(x, y) \end{aligned}$$
- $$g(x, y) = e^{s(x, y)} = e^{i'(x, y)} \cdot e^{r'(x, y)} = i_0(x, y)r_0(x, y)$$
- The **illumination** component is characterized by slow spatial variation. (**low frequency**)
- The **reflectance** component tends to vary abruptly especially at the junction of dissimilar objects. (**high frequency**)
- Homomorphic filter** $H(u, v)$ affects the low and high frequency components in different ways.

2018/10/4

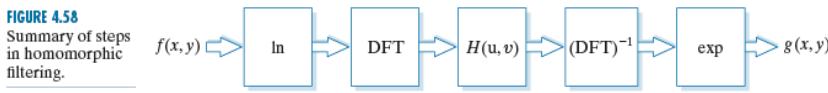
Digital Image Processing

97

Homomorphic Filtering

FIGURE 4.58

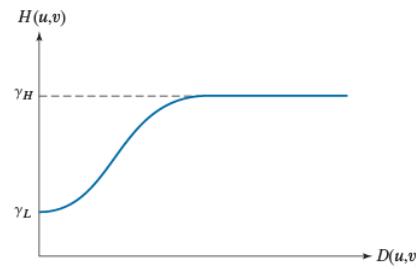
Summary of steps in homomorphic filtering.



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

$$\gamma_H > 1, \gamma_L < 1$$

Attenuate the low-frequency (illumination) and **amplify** the high-frequency (reflectance)

**FIGURE 4.59**

Radial cross section of a homomorphic filter transfer function.

2018/10/4

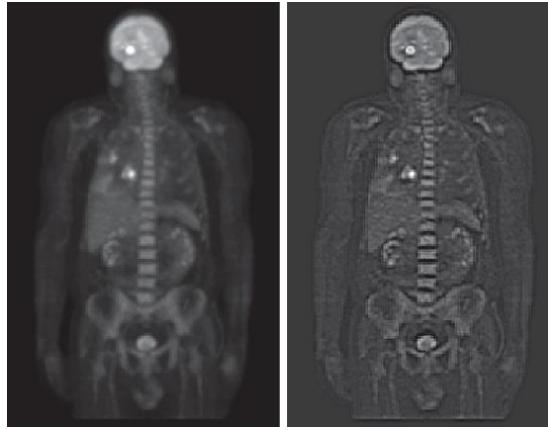
Digital Image Processing

98

Homomorphic Filtering

a b

FIGURE 4.60
 (a) Full body PET scan. (b) Image enhanced using homomorphic filtering. (Original image courtesy of Dr. Michael E. Casey, CTI Pet Systems.)



2018/10/4

Digital Image Processing

99

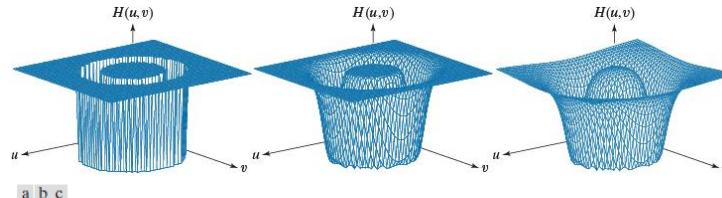
Band-Reject and Band-Pass Filters

- $H_{BR}(u, v) = 1 - H_{BP}(u, v)$

TABLE 4.7

Bandreject filter transfer functions. C_0 is the center of the band, W is the width of the band, and $D(u, v)$ is the distance from the center of the transfer function to a point (u, v) in the frequency rectangle.

Ideal (IBRF)	Gaussian (GBRF)	Butterworth (BBRF)
$H(u, v) = \begin{cases} 0 & \text{if } C_0 - \frac{W}{2} \leq D(u, v) \leq C_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u, v) = 1 - e^{-\left[\frac{D^2(u, v) - C_0^2}{D(u, v)W}\right]^2}$	$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)W}{D^2(u, v) - C_0^2}\right]^{2n}}$



a b c

FIGURE 4.62 Perspective plots of (a) ideal, (b) modified Gaussian, and (c) modified Butterworth (of order 1) bandreject filter transfer functions from Table 4.7. All transfer functions are of size 512×512 elements, with $C_0 = 128$ and $W = 60$.

2018/10/4

Digital Image Processing

100

Notch Filter

- A notch filter rejects (or passes) frequencies in a predefined neighborhood about the center of the frequency rectangle
- Notch filter:

$$H(u, v) = 0 \quad \text{if } (u, v) = (P/2, Q/2)$$

$$H(u, v) = 1 \quad \text{otherwise}$$

2018/10/4

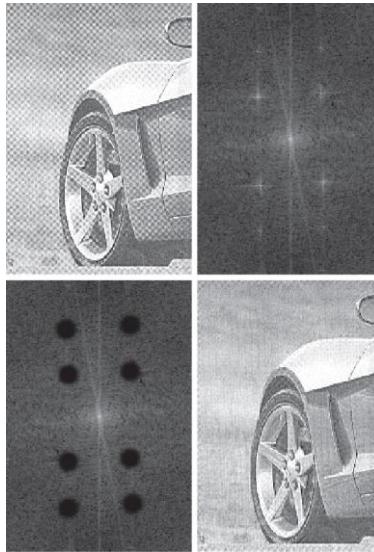
Digital Image Processing

101

Notch Filter

a
b
c
d

FIGURE 4.64
 (a) Sampled newspaper image showing a moiré pattern.
 (b) Spectrum.
 (c) Fourier transform multiplied by a Butterworth notch reject filter transfer function.
 (d) Filtered image.



2018/10/4

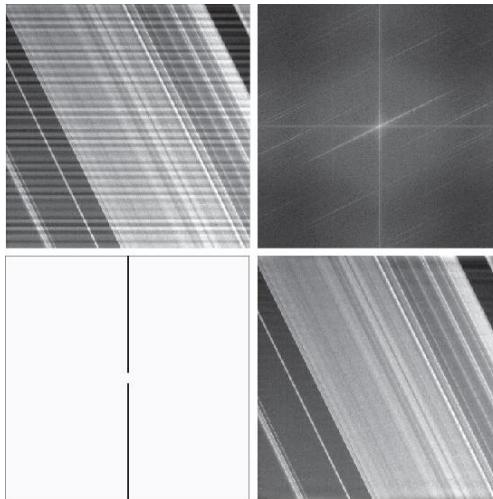
Digital Image Processing

102

Notch Filter

a
b
c
d

FIGURE 4.65
 (a) Image of Saturn rings showing nearly periodic interference.
 (b) Spectrum. (The bursts of energy in the vertical axis near the origin correspond to the interference pattern).
 (c) A vertical notch reject filter transfer function.
 (d) Result of filtering.
 (The thin black border in (c) is not part of the data.) Original image courtesy of Dr. Robert A. West, NASA/JPL.)



2018/10/4

Digital Image Processing

103

Notch Filter

a
b

FIGURE 4.66
 (a) Notch pass filter function used to isolate the vertical axis of the DFT of Fig. 4.65(a).
 (b) Spatial pattern obtained by computing the IDFT of (a).



2018/10/4

Digital Image Processing

104