

변환 영역 처리

Image Processing at Frequency Space

담당교수: 김민기

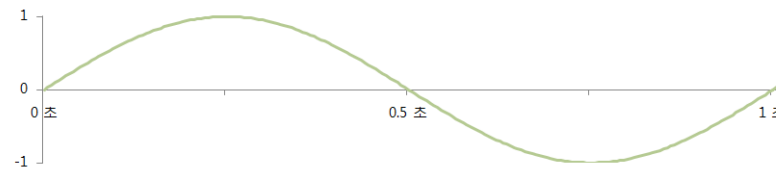
Contents

- ❖ 공간 주파수의 이해
- ❖ 이산 푸리에 변환 (Discrete Fourier Transform)
- ❖ 고속 푸리에 변환 (Fast Fourier Transform)
- ❖ 주파수 영역 필터링
- ❖ DFT를 이용한 회선과 상관

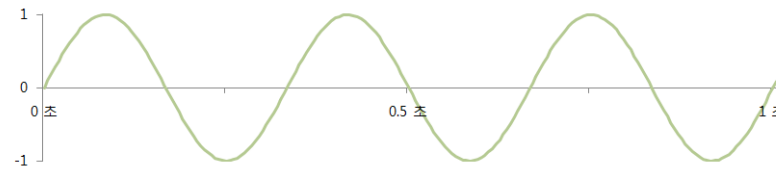
공간 주파수의 이해

❖ 주파수 (frequency)

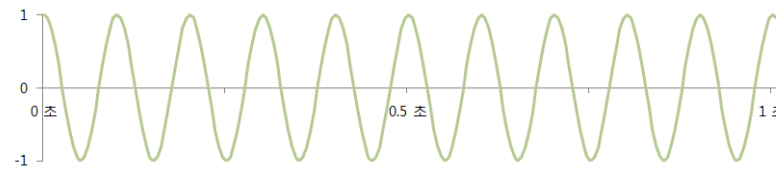
- 주파수의 단위: 헤르츠(Hz) 1초 동안에 진동하는 횟수



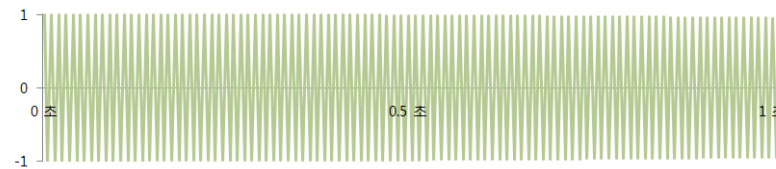
1초에 한번 진동 = 1Hz



1초에 세번 진동 = 3Hz



1초에 열번 진동 = 10Hz



1초에 백번 진동 = 100Hz

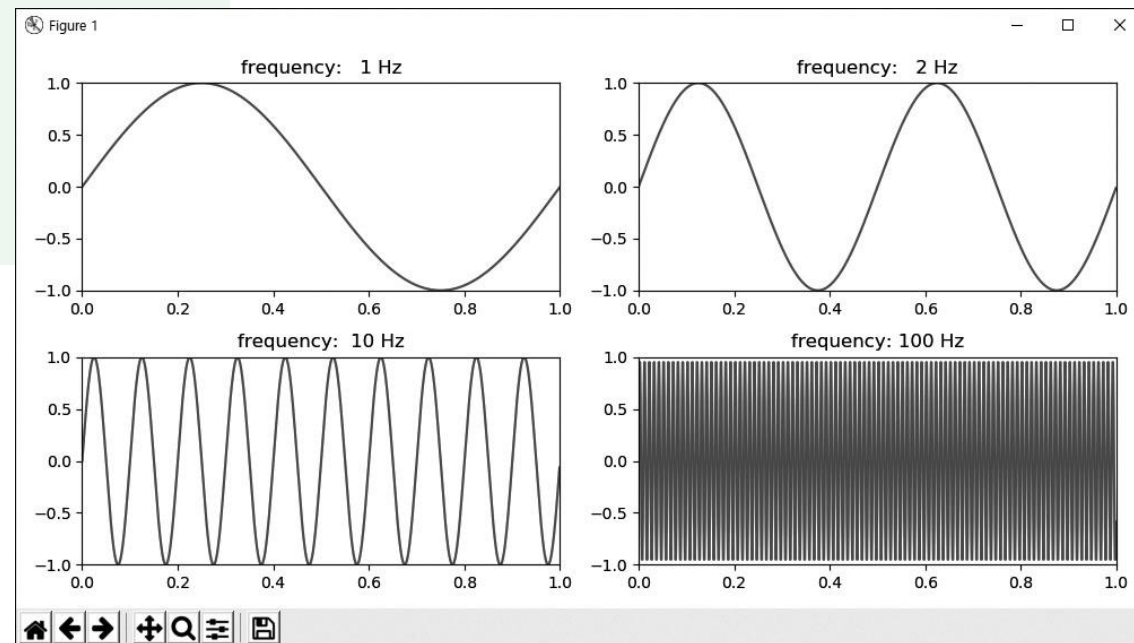
예제 9.1.1

주파수 그리기 | - 01.frequency.py

```

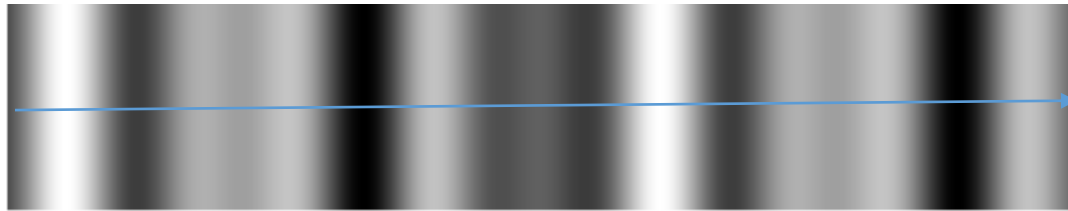
01 import matplotlib.pyplot as plt          # 그래프그리기 라이브러리 임포트
02 import numpy as np
03
04 t = np.arange(0, 1, 0.001)              # 샘플링 범위 및 개수
05 Hz = [1, 2, 10, 100]                   # 주파수 예시
06 gs = [np.sin(2 * np.pi * t * h) for h in Hz]  # sin 함수 계산
07
08 plt.figure(figsize=(10, 5))
09 for i, g in enumerate(gs):
10     plt.subplot(2, 2, i+1), plt.plot(t, g)    # 그래프 그리기
11     plt.xlim(0, 1), plt.ylim(-1, 1)          # x, y축 범위 지정
12     plt.title("frequency: %3d Hz" % Hz[i])
13 plt.tight_layout()
14 plt.show()

```



영상에서의 주파수

- 공간상에서 화소 밝기의 변화율
- 이런 의미에서 공간 주파수라는 표현을 사용



영상에서의 주파수

❖ 저주파 공간 영역

- 화소 밝기가 거의 변화가 없거나 점진적으로 변화하는 것
- 영상에서 배경 부분이나 객체의 내부에 많이 있음

❖ 고주파 공간 영역

- 화소 밝기가 급변하는 것
- 경계부분이나 객체의 모서리 부분



저주파 영역

고주파 영역

영상에서의 주파수

❖ 영상을 주파수 영역별로 분리할 수 있다면?



Fourier Transform

❖ 푸리에 변환

- 임의의 신호는 다양한 주파수를 갖는 주기 함수의 합으로 표현될 수 있다.
- 주기 함수: 정현파(sin), 여현파(cos)

$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot g_f(t) df$$

$g_f(t)$: 주파수가 f 인 기저함수

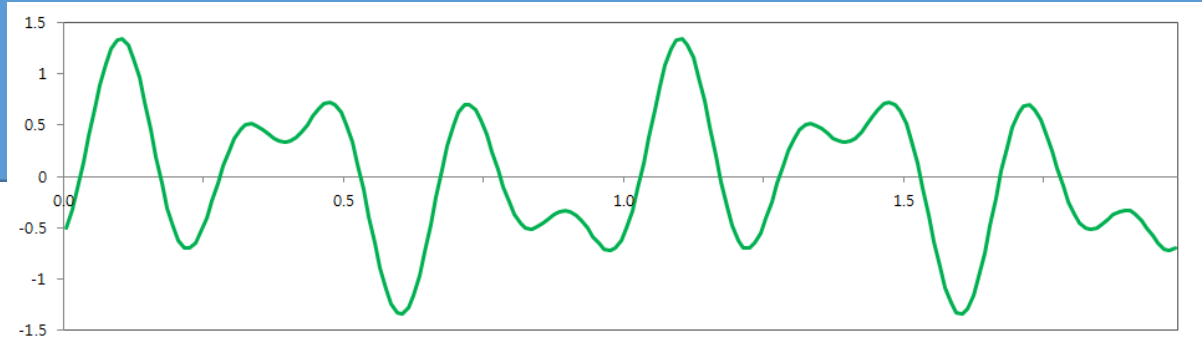
$G(f)$: 기저함수의 계수

푸리에 변환

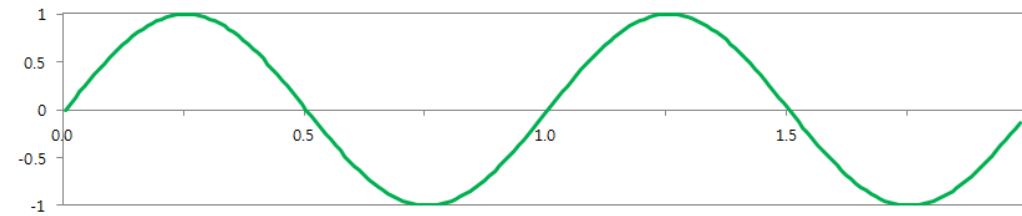
$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot g_f(t) df$$

$g_f(t)$: 주파수가 f 인 기저함수

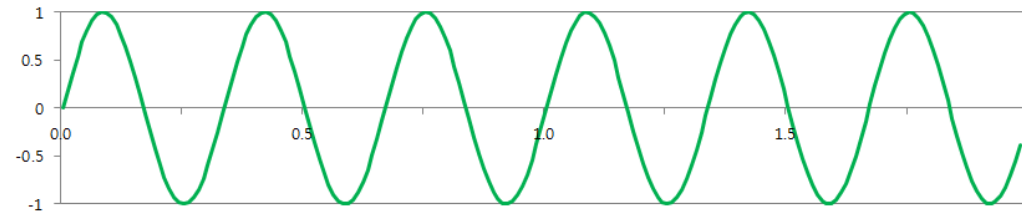
$G(f)$: 기저함수의 계수



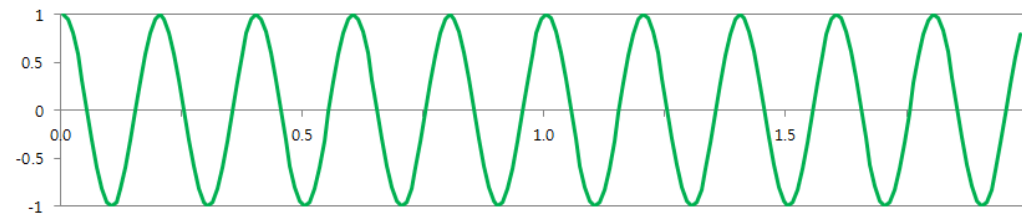
$$g(t) = 0.3 * g_1(t) + 0.7 * g_2(t) - 0.5 * g_3(t)$$



$$g_1(t) = \sin(2\pi 1t)$$



$$g_2(t) = \sin(2\pi 3t)$$



$$g_3(t) = \cos(2\pi 5t)$$

Sine & Cosine 함수

- Periodic functions
- General form of sine and cosine functions:

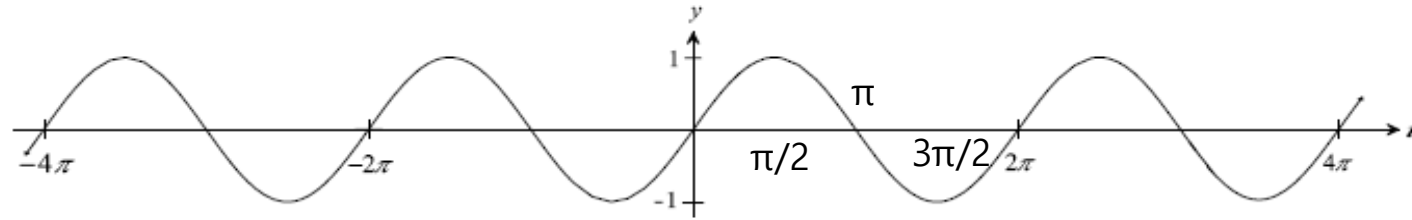
$$y(t) = A \sin[a(t + b)] \quad y(t) = A \cos[a(t + b)]$$

$ A $	amplitude
$\frac{2\pi}{ a }$	period
b	phase shift

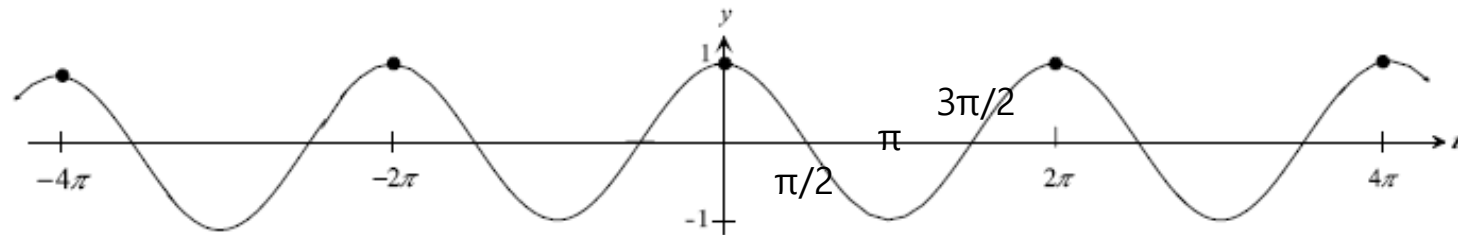
Sine & Cosine 함수

- case: $A=1, b=0, a=1$

$$y(t) = A \sin[a(t + b)]$$



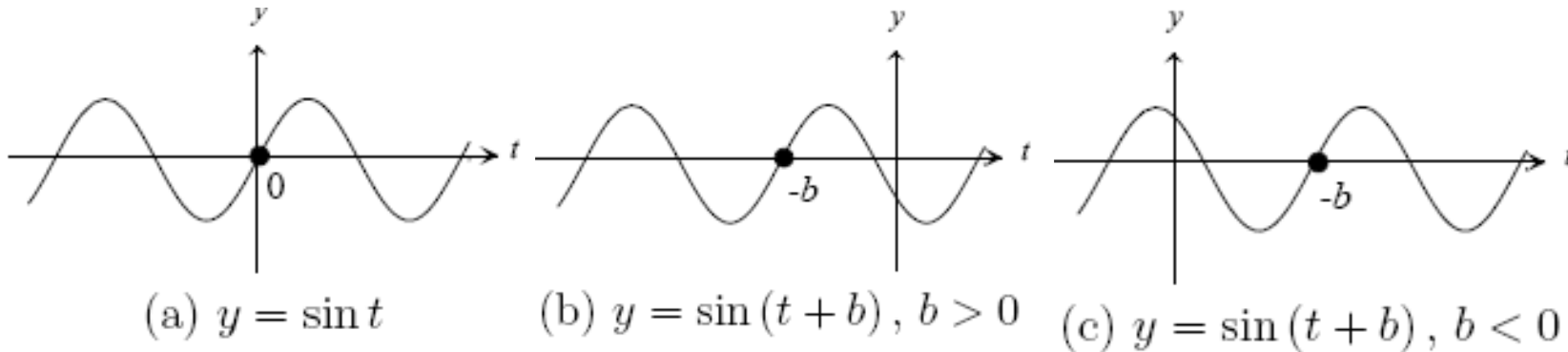
$$y(t) = A \cos[a(t + b)]$$



Sine & Cosine 함수

- Shifting or translating the sine function by a const b

$$y(t) = A \sin[a(t + b)]$$



Note: cosine is a shifted sine function:

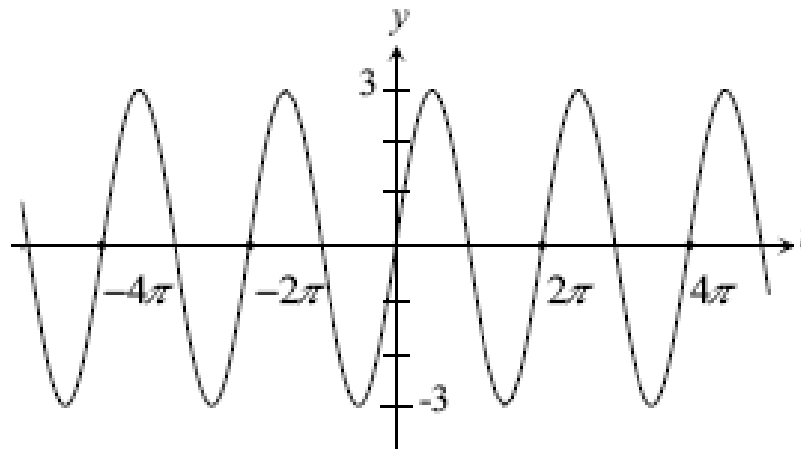
$$\cos(t) = \sin\left(t + \frac{\pi}{2}\right)$$

Sine & Cosine 함수

- Changing the amplitude A

$$y(t) = A \sin[a(t + b)]$$

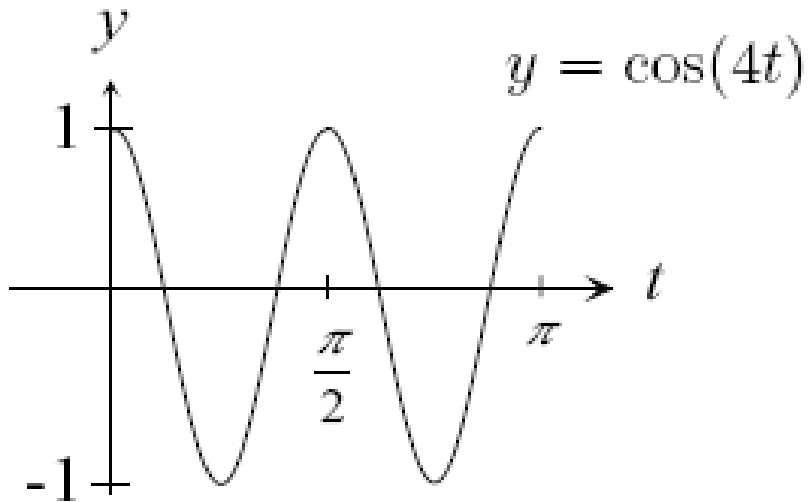
$$y = 3 \sin t.$$



Sine & Cosine 함수

- Changing the **period** $T=2\pi/|a|$
consider $A=1, b=0$: $y=\cos(at)$

$$y(t) = A \cos[a(t + b)]$$



$$a=4 \rightarrow \text{period } 2\pi/4 = \pi/2$$

shorter period
higher frequency
(i.e., oscillates faster)

Frequency is defined as $f=1/T$

Alternative notation: $\sin(at)=\sin(2\pi t/T)=\sin(2\pi ft)$
(if $a > 0, T=2\pi/a \rightarrow a=2\pi/T$)

Complex Number

❖ A complex number x is of the form:

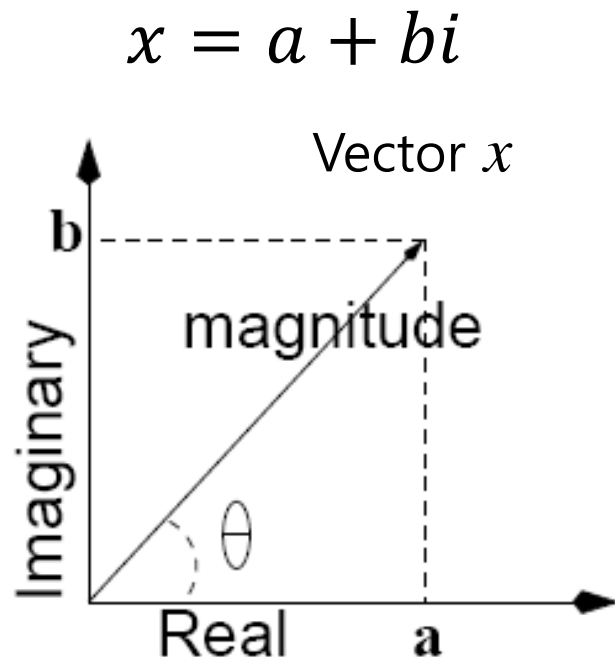
$$x = a + bi, \quad \text{where } i = \sqrt{-1}$$

❖ Addition: $(a + bi) + (c + di) = (a + c) + (b + d)i$

❖ Multiplication: $(a + bi)(c + di) = (ac - bd) + (ad + bc)i$

Complex Number

❖ Magnitude-Phase (i.e., vector) representation



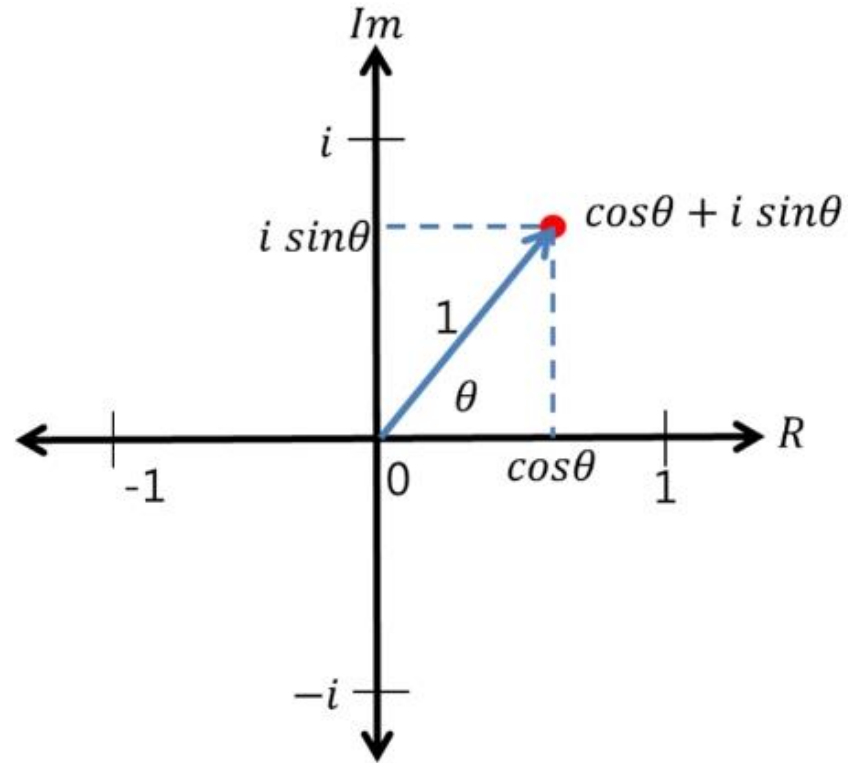
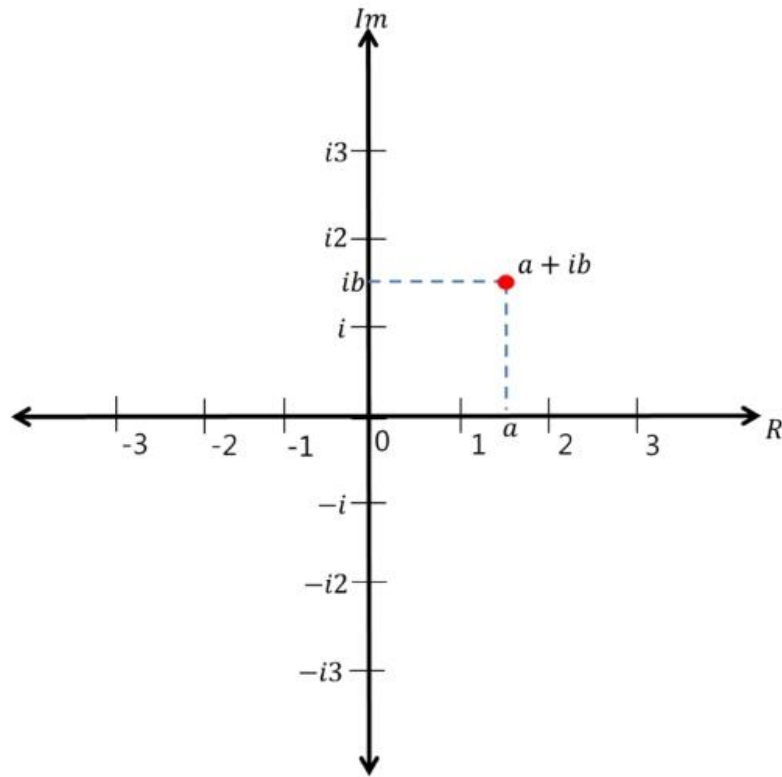
$$|x| = \sqrt{a^2 + b^2}$$

$$\theta(x) = \tan^{-1} \left(\frac{b}{a} \right)$$

Magnitude-Phase notation:

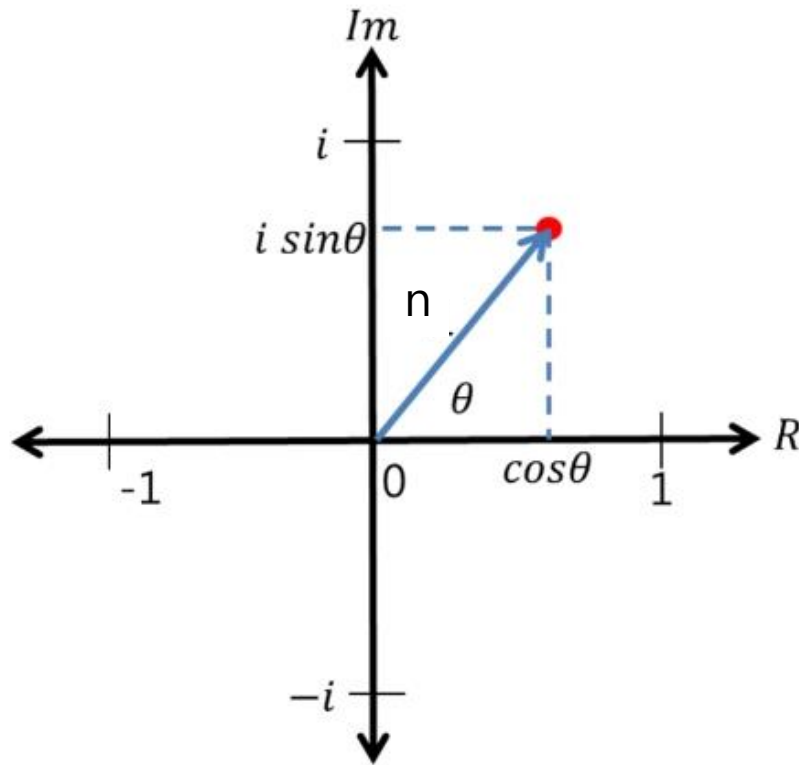
$$x = |x|(\cos\theta + i\sin\theta)$$

Complex Number \rightarrow Complex Plane

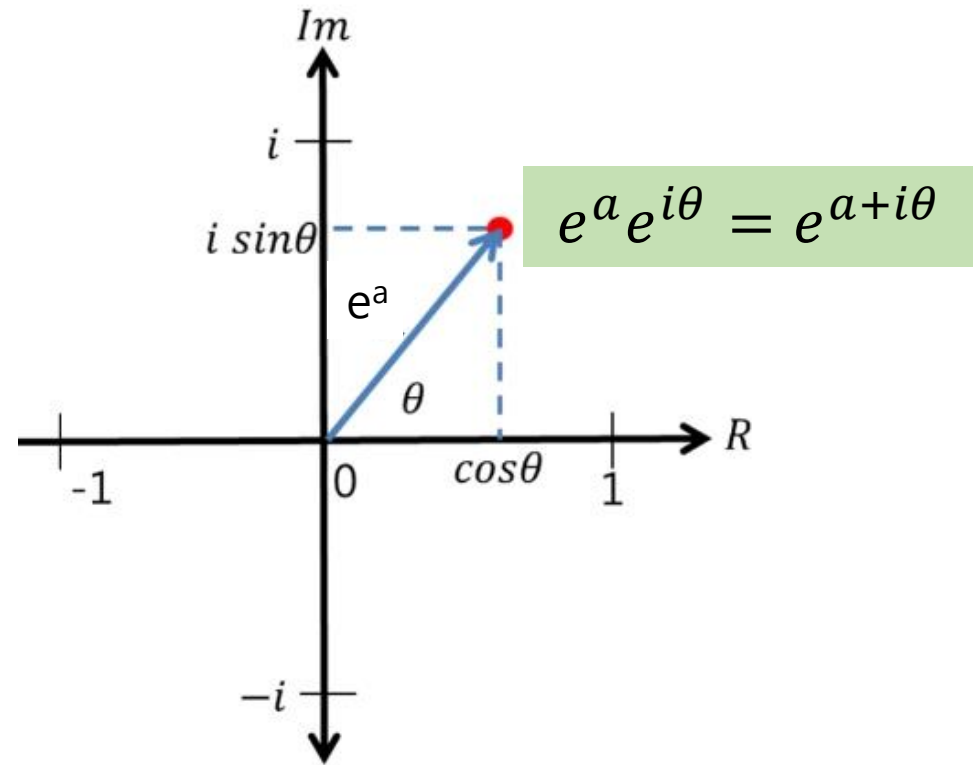


$$e^{i\theta} = \cos\theta + i\sin\theta$$

Complex Number \rightarrow Complex Plane



$$n(\cos\theta + i\sin\theta) = ne^{i\theta}$$



Fourier Transform

■ The **one-dimensional** Fourier transform and its inverse

● Fourier transform (**continuous case**)

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx \quad \text{where } j = \sqrt{-1}$$

● Inverse Fourier transform:

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

$$e^{i\theta} = \cos\theta + i\sin\theta$$

❖ u : the frequency variables

❖ x : the spatial variables

■ The **two-dimensional** Fourier transform and its inverse

● Fourier transform (**continuous case**)

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

● Inverse Fourier transform:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv$$

1차원 Discrete Fourier Transform

- The **one-dimensional** Fourier transform and its inverse
 - Fourier transform (**discrete case**)

$$F(u) = \sum_{x=0}^{N-1} f(x) e^{-\frac{j2\pi ux}{N}}, \text{ where } u = 0, 1, 2, \dots, N-1$$

DFT의 계산량은 입력신호의 길이(N)의 제곱에 비례

- Inverse Fourier transform:

$$f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) e^{\frac{j2\pi ux}{N}}, \text{ where } x = 0, 1, 2, \dots, N-1$$

Alternative notation: $\sin(at) = \sin(2\pi t/T) = \sin(2\pi ft)$
(if $a > 0$, $T = 2\pi/a \rightarrow a = 2\pi/T$)

2차원 Discrete Fourier Transform

- The **two-dimensional** Fourier transform and its inverse
 - Fourier transform (**discrete case**)

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

, where $u = 0, 1, 2, \dots, M-1, v = 0, 1, 2, \dots, N-1$

- Inverse Fourier transform:

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

, where $x = 0, 1, 2, \dots, M-1, y = 0, 1, 2, \dots, N-1$

Discrete Fourier Transform

$$F(u) = \sum_{x=0}^{N-1} f(x) e^{-\frac{j2\pi ux}{N}}, \text{ where } u = 0, 1, 2, \dots, N-1$$

$$\operatorname{Re}\{F(u)\} = \sum_{x=0}^{N-1} \left(\operatorname{Re}\{f(x)\} \cos\left(-\frac{2\pi}{N}ux\right) - \operatorname{Im}\{f(x)\} \sin\left(-\frac{2\pi}{N}ux\right) \right)$$

$$\operatorname{Im}\{F(u)\} = \sum_{x=0}^{N-1} \left(\operatorname{Im}\{f(x)\} \cos\left(-\frac{2\pi}{N}ux\right) + \operatorname{Re}\{f(x)\} \sin\left(-\frac{2\pi}{N}ux\right) \right)$$

$$f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) e^{\frac{j2\pi ux}{N}}, \text{ where } x = 0, 1, 2, \dots, N-1$$

$$\operatorname{Re}\{f(x)\} = \frac{1}{N} \sum_{u=0}^{N-1} \left(\operatorname{Re}\{F(u)\} \cos\left(\frac{2\pi}{N}ux\right) - \operatorname{Im}\{F(u)\} \sin\left(\frac{2\pi}{N}ux\right) \right)$$

$$\operatorname{Im}\{f(x)\} = \frac{1}{N} \sum_{u=0}^{N-1} \left(\operatorname{Im}\{F(u)\} \cos\left(\frac{2\pi}{N}ux\right) + \operatorname{Re}\{F(u)\} \sin\left(\frac{2\pi}{N}ux\right) \right)$$

푸리에 변환 Summary

❖ 복소수로 표현되는 기저 함수를 사용

$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot g_f(t) df$$

$g_f(t) : f$ 주파수에 대한 기저함수
 $G(f) : \text{기저함수의 계수}$

$$g_f(t) = \cos(2\pi ft) + j \cdot \sin(2\pi ft) = e^{j2\pi ft}$$

$$G(f) = \int_{-\infty}^{\infty} g(t) \cdot e^{-j2\pi ft} dt$$

원본 신호로부터 기저함수의 계수를 얻는 것
→ 푸리에 변환

$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot e^{j2\pi ft} df$$

기저함수의 계수로부터 원본 신호를 얻는 것
→ 푸리에 역변환

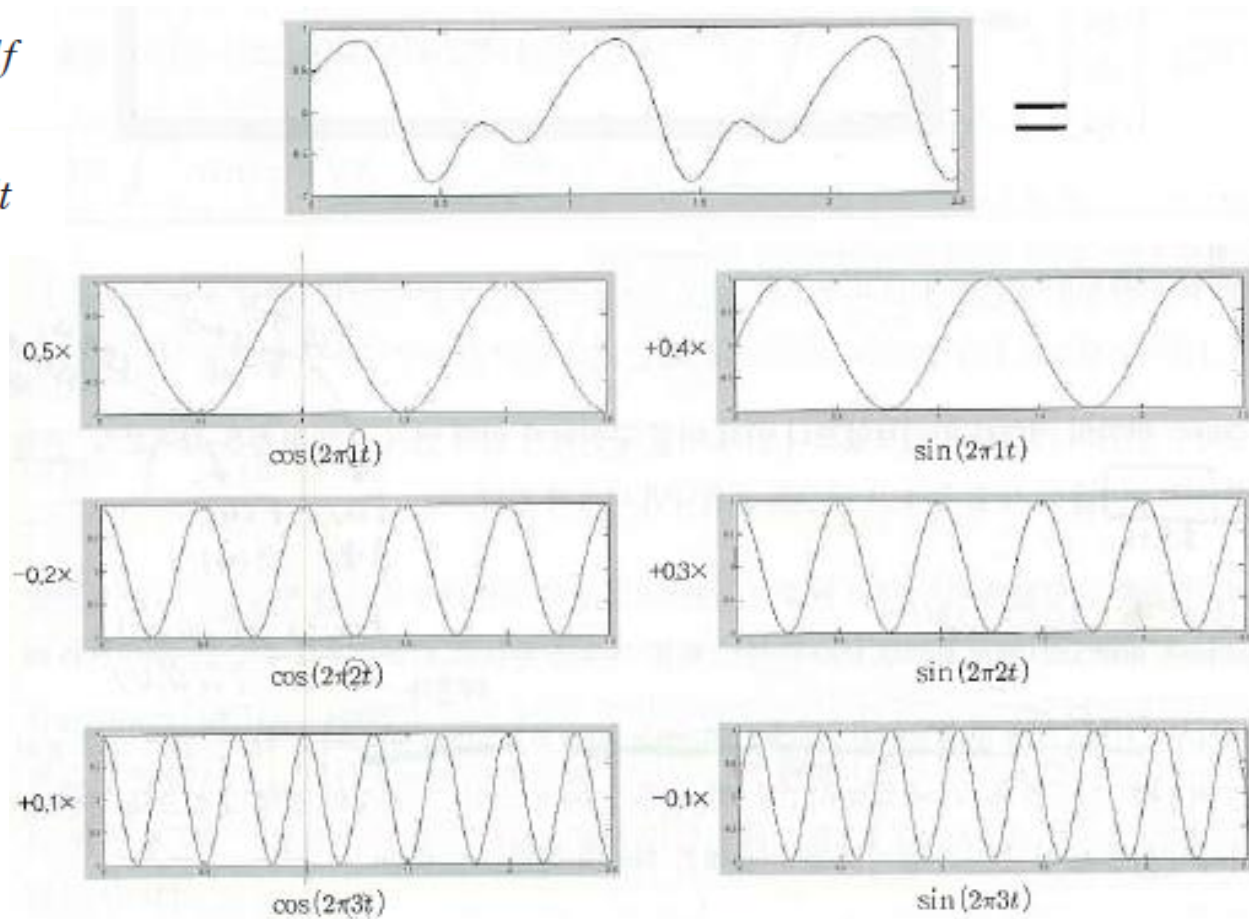
푸리에 변환

$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot g_f(t) df$$

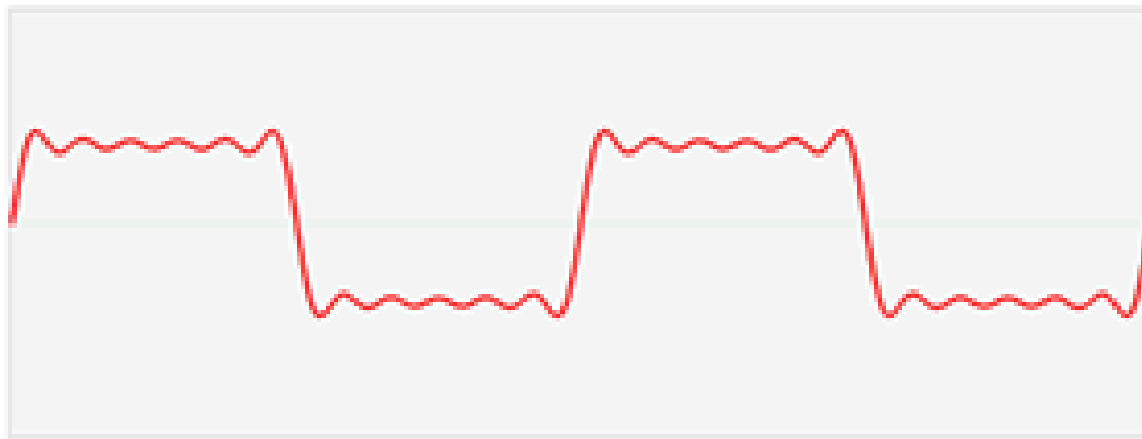
$$g_f(t) = \cos(2\pi ft) + j \cdot \sin(2\pi ft) = e^{j2\pi ft}$$

$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot e^{j2\pi ft} df$$

$$G(f) = \int_{-\infty}^{\infty} g(t) \cdot e^{-j2\pi ft} dt$$



푸리에 변환



Fast Fourier Transform

$$F(u) = \sum_{x=0}^{N-1} f(x) e^{-\frac{j2\pi}{N}ux} \quad , \text{where } N = 2^d (d \text{는 정수}), \text{ 즉 } N \text{은 짝수이므로 } N = 2L$$

$$F(u) = \sum_{x=0}^{2L-1} f(x) e^{-\frac{j2\pi}{2L}ux} \quad F(u) = F_{\text{even}}(u) + F_{\text{odd}}(u) e^{-\frac{j2\pi}{2L}u}$$

$$= \left\{ \sum_{x=0}^{L-1} f(2x) e^{-\frac{j2\pi}{2L}u2x} + \sum_{x=0}^{L-1} f(2x+1) e^{-\frac{j2\pi}{2L}u(2x+1)} \right\}$$

$$= \left\{ \sum_{x=0}^{L-1} f(2x) e^{-\frac{j2\pi}{2L}u2x} + \sum_{x=0}^{L-1} f(2x+1) e^{-\frac{j2\pi}{2L}u2x} e^{-\frac{j2\pi}{2L}u} \right\}$$

$$= \left\{ \sum_{x=0}^{L-1} f(2x) e^{-\frac{j2\pi}{L}ux} + \sum_{x=0}^{L-1} f(2x+1) e^{-\frac{j2\pi}{L}ux} e^{-\frac{j2\pi}{2L}u} \right\}$$

$$= \left\{ \sum_{x=0}^{L-1} f(2x) e^{-\frac{j2\pi}{L}ux} \right\} + \left\{ \sum_{x=0}^{L-1} f(2x+1) e^{-\frac{j2\pi}{L}ux} \right\} e^{-\frac{j2\pi}{2L}u}$$

Fast Fourier Transform

FFT는 삼각함수의 주기성을 이용
→ 중복되는 계산량을 대폭 감소

주기가 $2L$ 인 함수 $F(u)$

$$F(u) = F_{\text{even}}(u) + F_{\text{odd}}(u)e^{-\frac{j2\pi}{2L}u}$$

주기가 L 인 함수 $F(u)$

$$F_{\text{even}}(u + L) = F_{\text{even}}(u)$$

$$F_{\text{odd}}(u + L) = F_{\text{odd}}(u)$$

FFT는 삼각함수의 주기성을 이용 (주기= L)

$$\begin{aligned} e^{-\frac{j2\pi}{L}(u+L)x} &= e^{-\frac{j2\pi}{L}ux} e^{-\frac{j2\pi}{L}Lx} \\ &= e^{-\frac{j2\pi}{L}ux} e^{-j2\pi x} \\ &= e^{-\frac{j2\pi}{L}ux} \end{aligned}$$

FFT는 삼각함수의 주기성을 이용 (주기= $2L$)

$$\begin{aligned} e^{-\frac{j2\pi}{2L}(u+L)x} &= e^{-\frac{j2\pi}{2L}ux} e^{-\frac{j2\pi}{2L}Lx} \\ &= e^{-\frac{j2\pi}{2L}ux} e^{-j\pi x} \\ &= -e^{-\frac{j2\pi}{2L}ux} \end{aligned}$$

$$\begin{aligned} F(u + L) &= F_{\text{even}}(u + L) + F_{\text{odd}}(u + L)e^{-\frac{j2\pi}{2L}(u+L)} \\ &= F_{\text{even}}(u) + F_{\text{odd}}(u)e^{-\frac{j2\pi}{2L}(u+L)} \\ &= F_{\text{even}}(u) - F_{\text{odd}}(u)e^{-\frac{j2\pi}{2L}u} \end{aligned}$$

주기가 $2L$ 인 함수 $F(u)$ 를
주기가 L 인 함수 $F_{\text{even}}(u)$ 와 $F_{\text{odd}}(u)$ 로
변환하여 빠르게 계산

푸리에 변환의 표현

- $F(u)$ can be expressed in polar coordinates:

$$F(u) = |F(u)|e^{j\theta}$$

$$\text{where } |F(u)| = \sqrt{R^2(u) + I^2(u)} \quad (\text{magnitude or spectrum})$$

$$\theta(u) = \tan^{-1} \left(\frac{I(u)}{R(u)} \right) \quad (\text{phase angle or phase spectrum})$$

- Power spectrum:

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

Fourier Transform

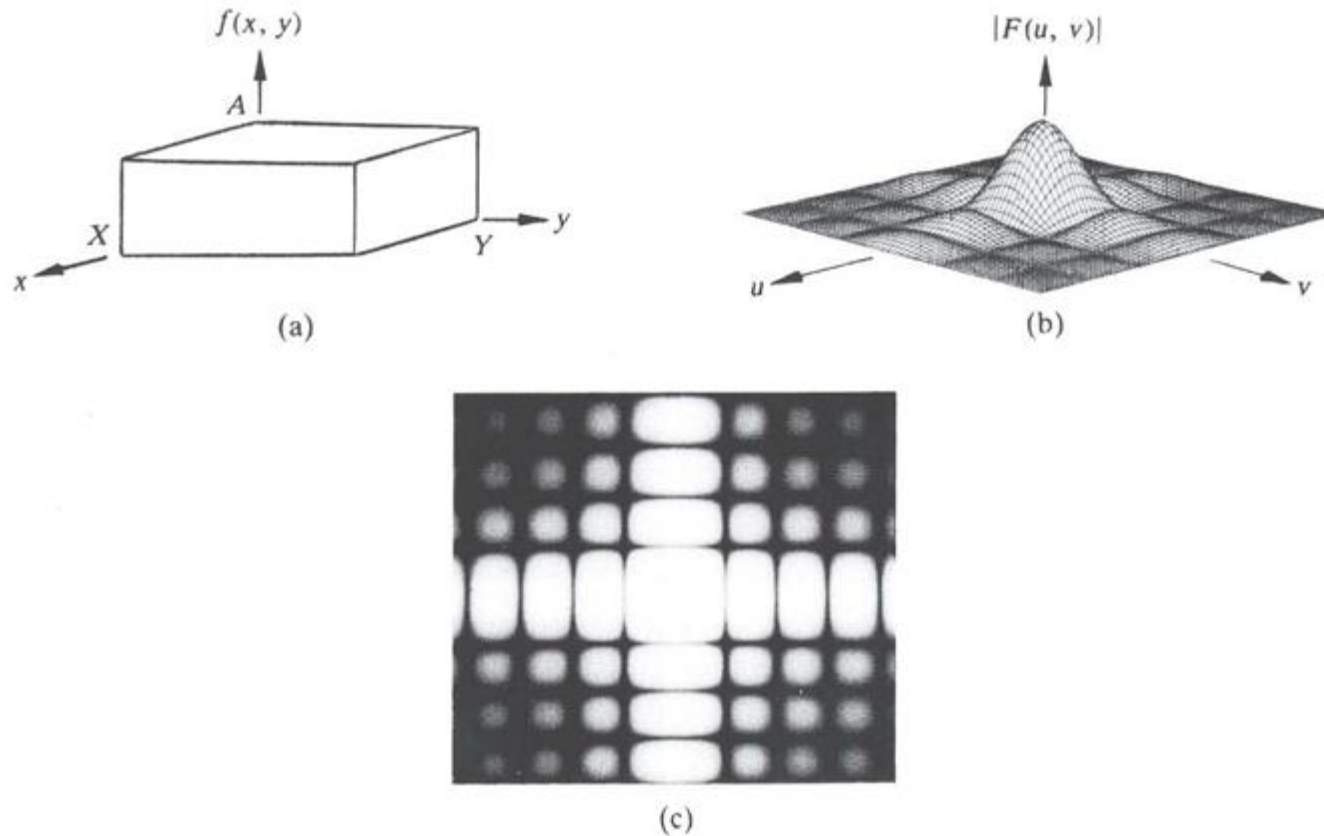
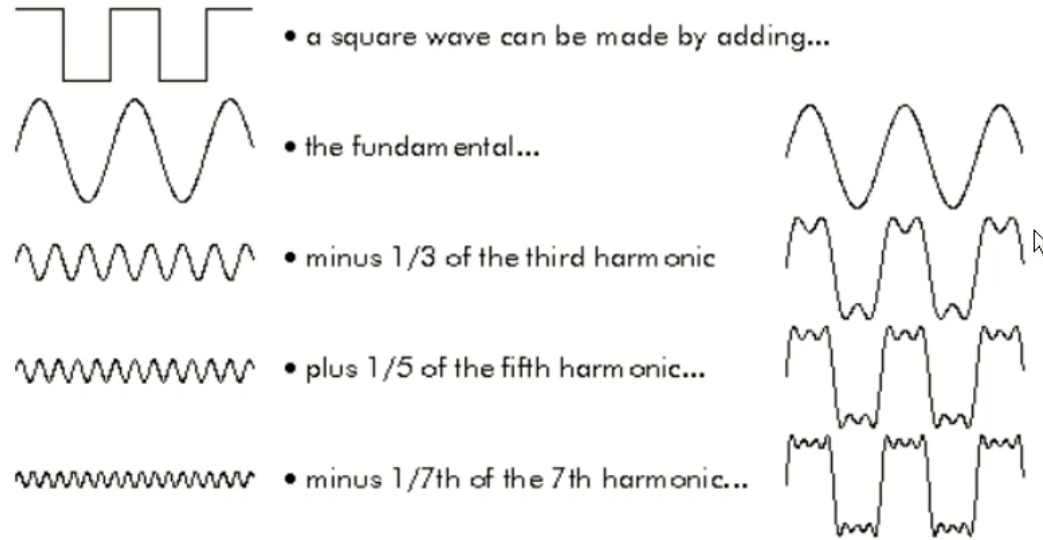
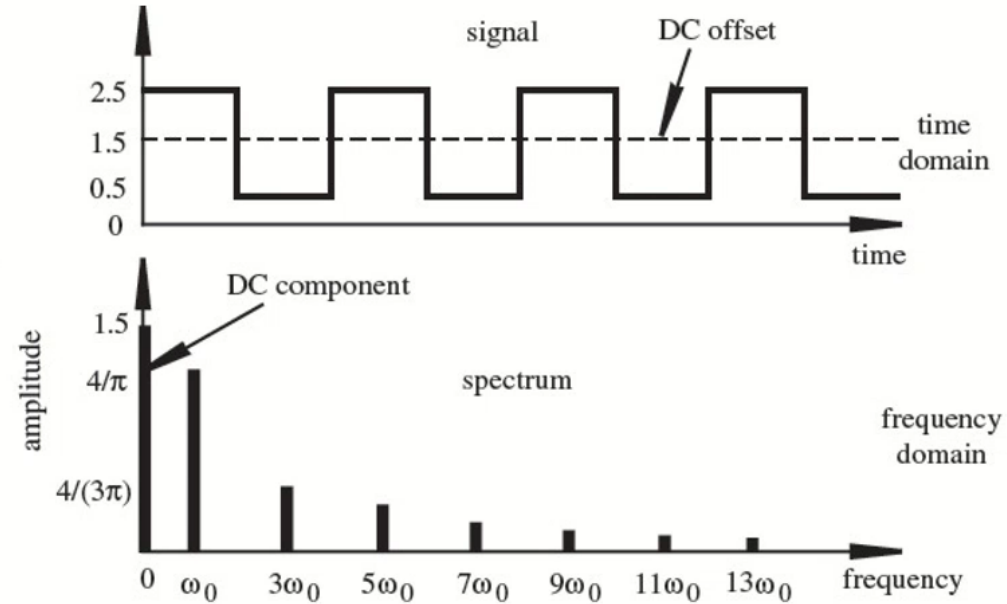


Figure 3.2 (a) A 2-D function; (b) its Fourier spectrum; and (c) the spectrum displayed as an intensity function.

Fourier Series



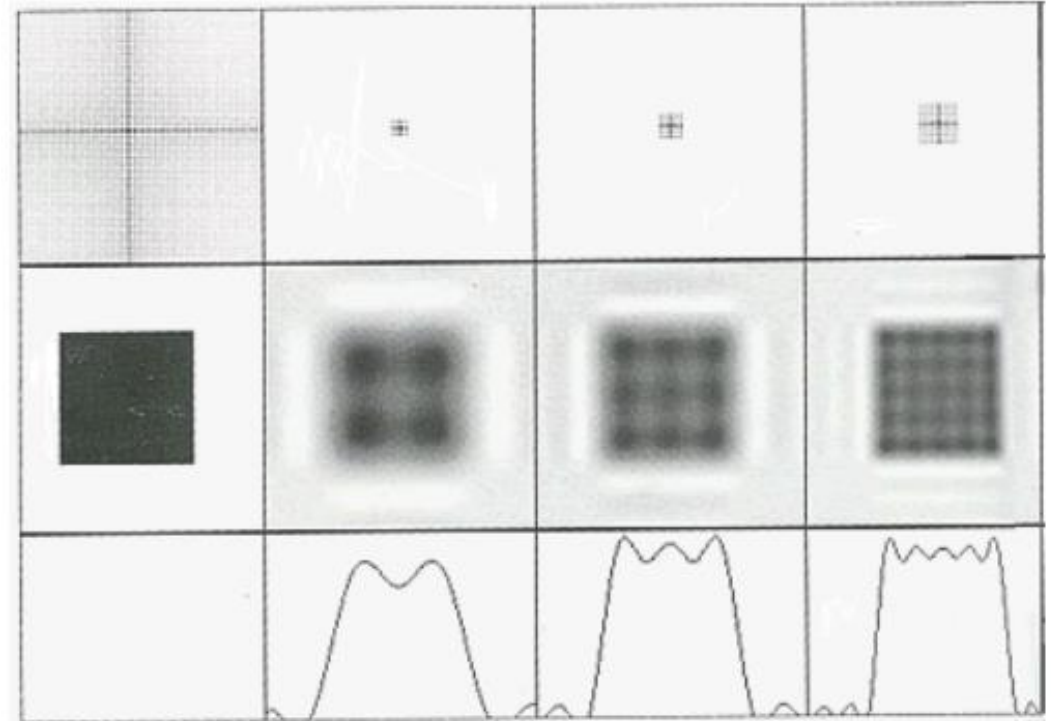
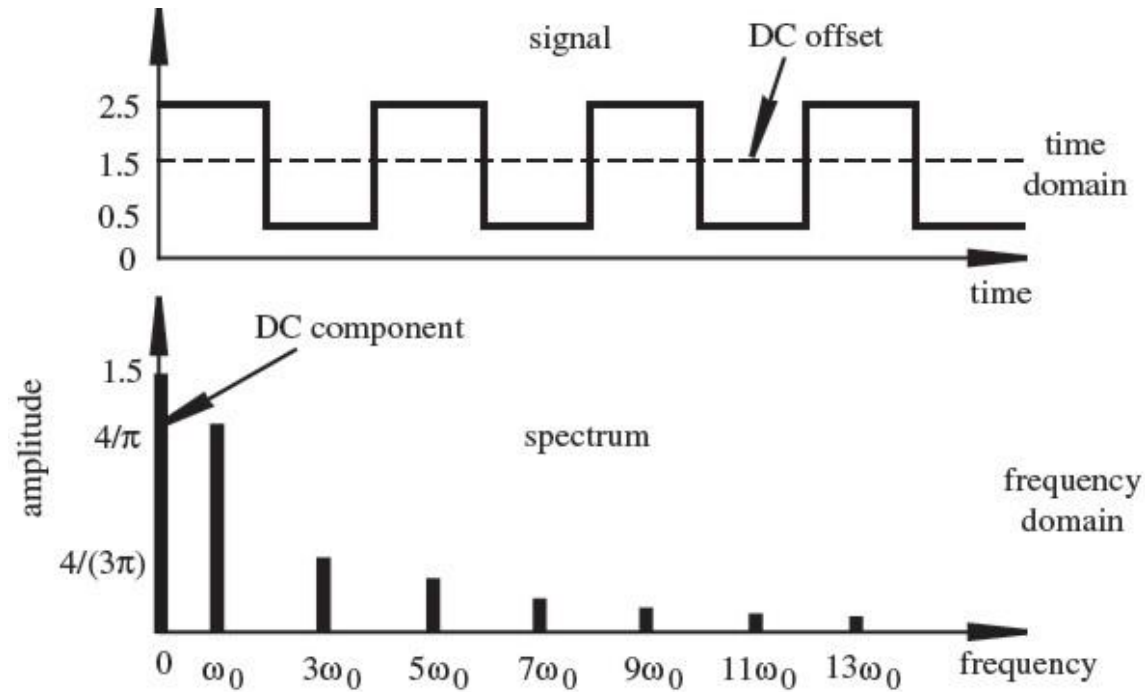
© **BORES** Signal Processing



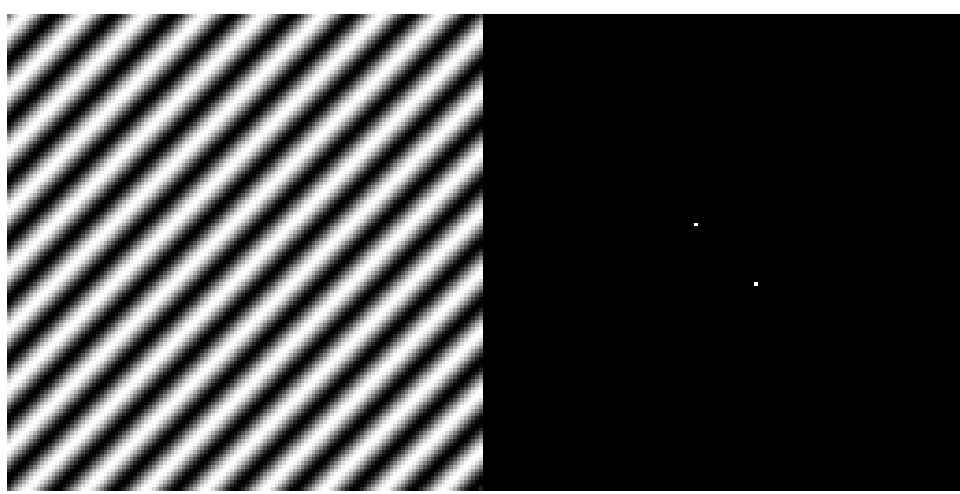
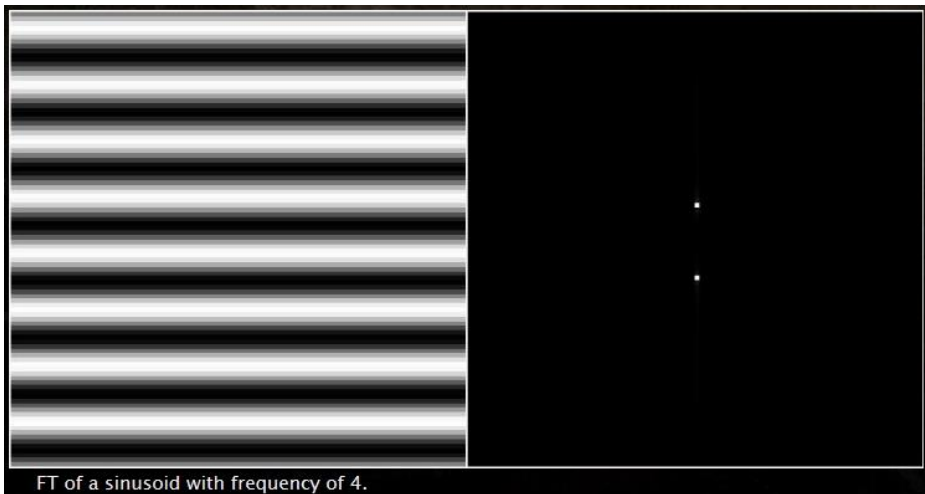
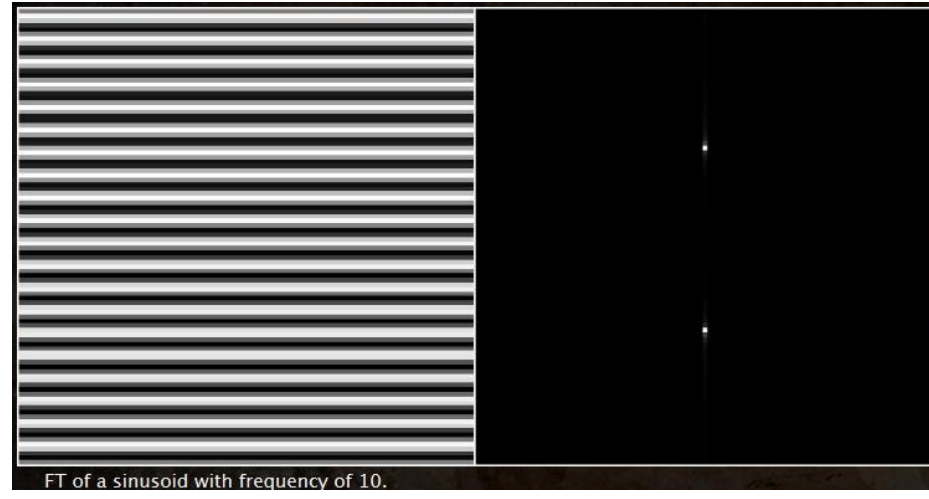
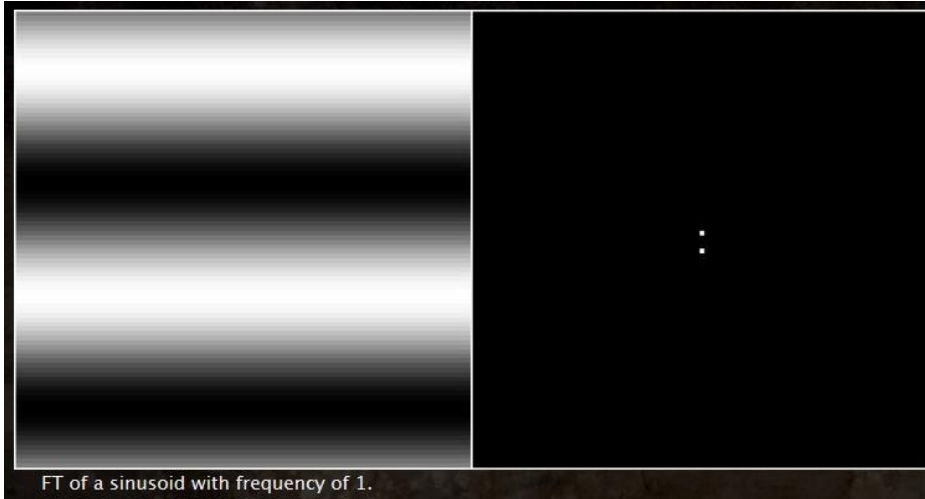
Square Wave > <http://www.falstad.com/fourier/e-square.html>

Magnitude & Phase View > <http://www.falstad.com/fourier/e-phase.html>

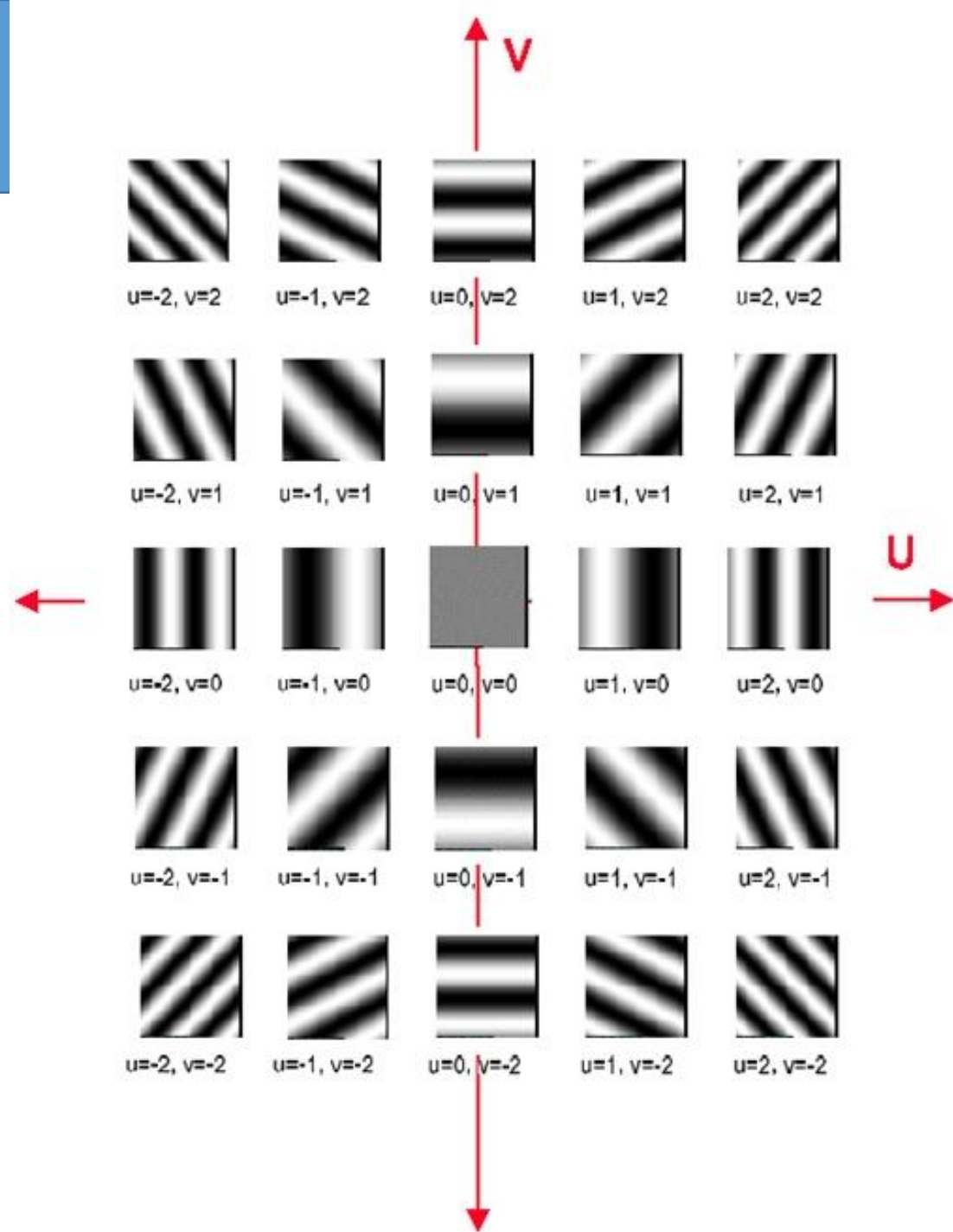
Fourier Transform



Fourier Transform



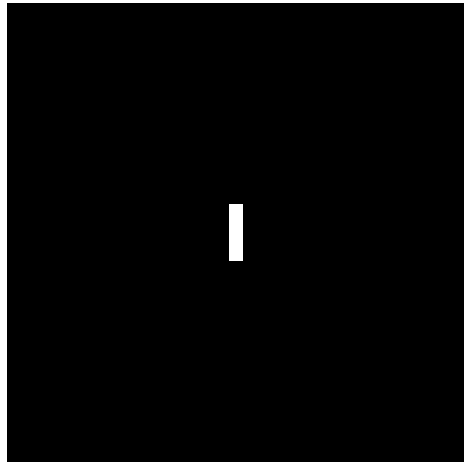
Fourier Transform



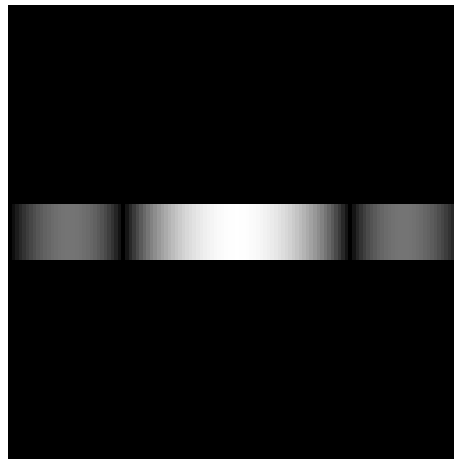
Fourier Transform

The 2D DFT $\mathbf{F(u,v)}$ can be obtained by

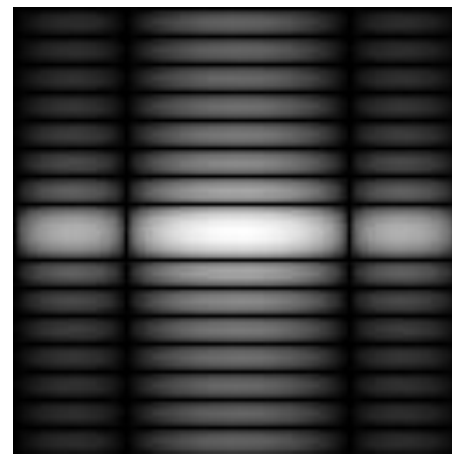
1. taking the 1D DFT of every row of image $\mathbf{f(x,y)} \rightarrow F(u, y)$
2. taking the 1D DFT of every column of $\mathbf{F(u,y)} \rightarrow F(u, v)$



(a) $f(x,y)$

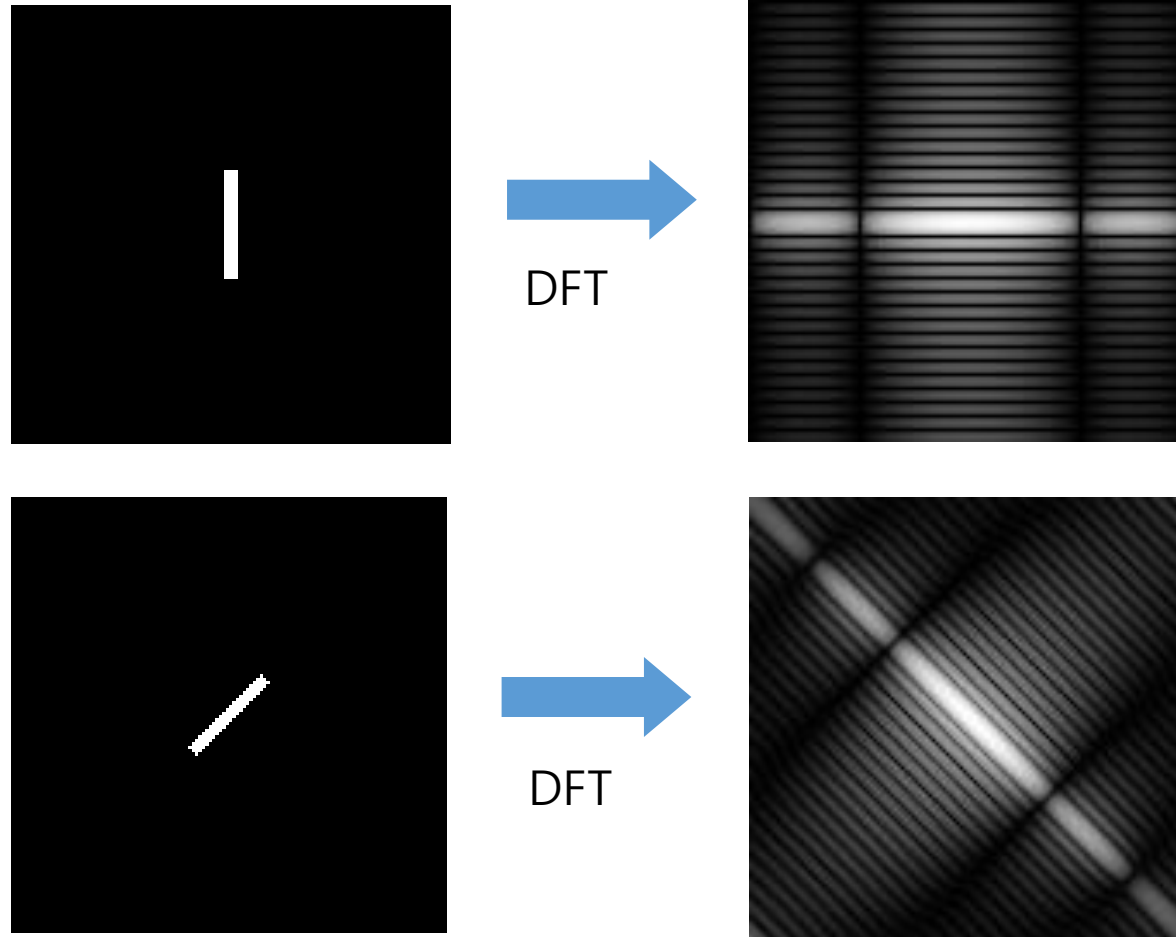


(b) $F(u,y)$



(c) $F(u,v)$

Fourier Transform

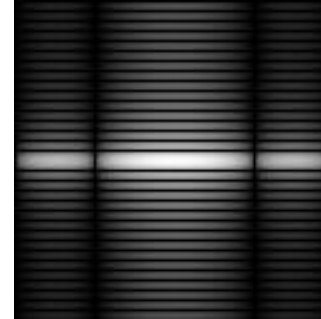


Fourier Transform

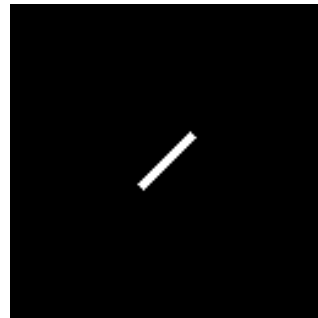
A



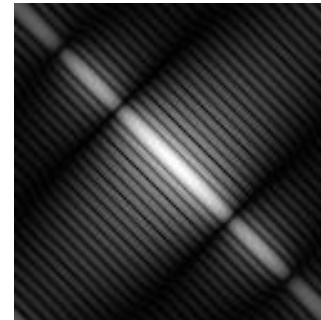
DFT



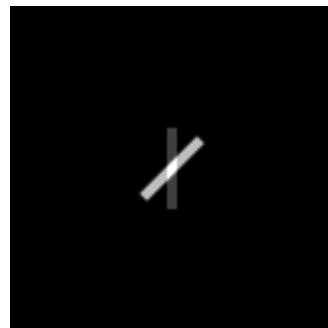
B



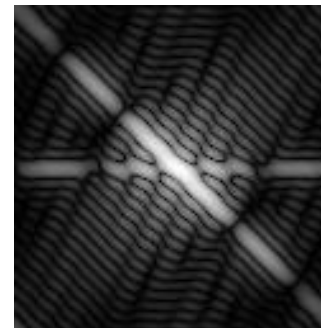
DFT



$0.25 * A$
 $+ 0.75 * B$

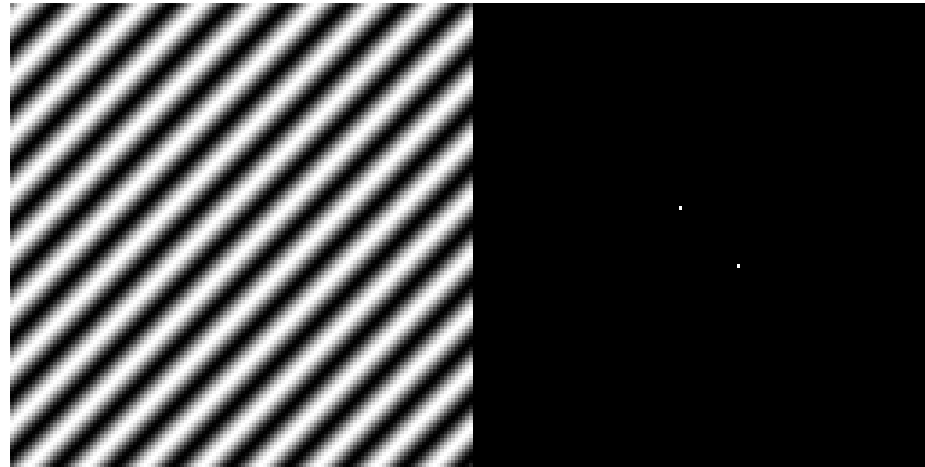


DFT



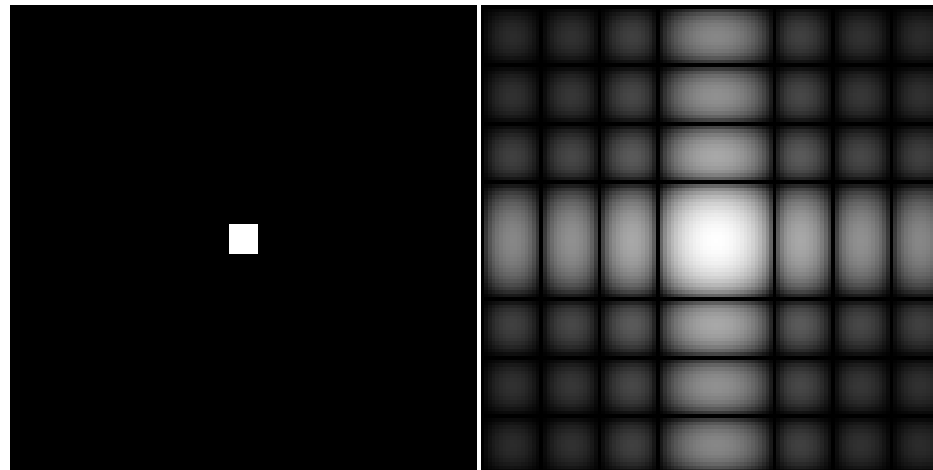
Fourier Transform

Sine wave



Its DFT

Rectangle



Its DFT

OpenCV DFT

❖ 이산 푸리에 변환 (DFT)

```
void dft (InputArray src, OutputArray dst, int flags1)=0, int nonzeroRows2)=0)  
void idft(InputArray src, OutputArray dst, int flags=0, int nonzeroRows=0)
```

1) 디폴트 0 : 정방향 푸리에 변환 수행

DFT_INVERSE : 역방향 푸리에 변환 수행

DFT_SCALE : 결과를 1/MN로 스케일

→ 일반적인 역방향 푸리에 변환을 하려면 DFT_INVERSE | DFT_SCALE

DFT_ROWS : 행 단위로 푸리에 변환 수행

DFT_COMPLEX_OUTPUT : 출력 dst가 2채널 복소수 행렬 (채널1: 실수, 채널2: 허수)

DFT_REAL_OUTPUT : 출력 dst가 1채널 실수 행렬 (실수부와 허수부 패킹)

CCS(Complex Conjugate Symmetry) 형태

2) DFT_INVERSE 가 설정되어 있지 않으면, src의 처음행에서부터 0이 아닌 행의 개수

DFT_INVERSE 가 설정되어 있으면, dst의 처음행에서부터 0이 아닌 행의 개수

→ 원본보다 큰 행렬 또는 영상으로 복사되어 0이 패딩된 경우, 계산이 필요한 0이 아닌 행의 개수(즉, 계산할 행의 크기)를 알려주어 계산 속도를 향상시키기 위한 것

OpenCV DFT

❖ DFT를 위한 최적의 크기 구하기

```
int getOptimalDFTSize( int vecsize)
```

* vecsize는 행렬 또는 영상의 행(또는 열)의 크기

- ✓ 대부분의 FFT 알고리즘에서 최적의 크기 $N = 2^n$ 을 만족시키는 최소의 N
- ✓ OpenCV에서 FFT 는 $N = 2^p 3^q 5^r$ 을 만족시키는 최소의 N
- ✓ OpenCV에서 DCT에서의 최적의 크기는
→ $\text{getOptimalDFTSize}((\text{vecsize}+1)/2) * 2$ 로 계산함

주파수 영역 필터링

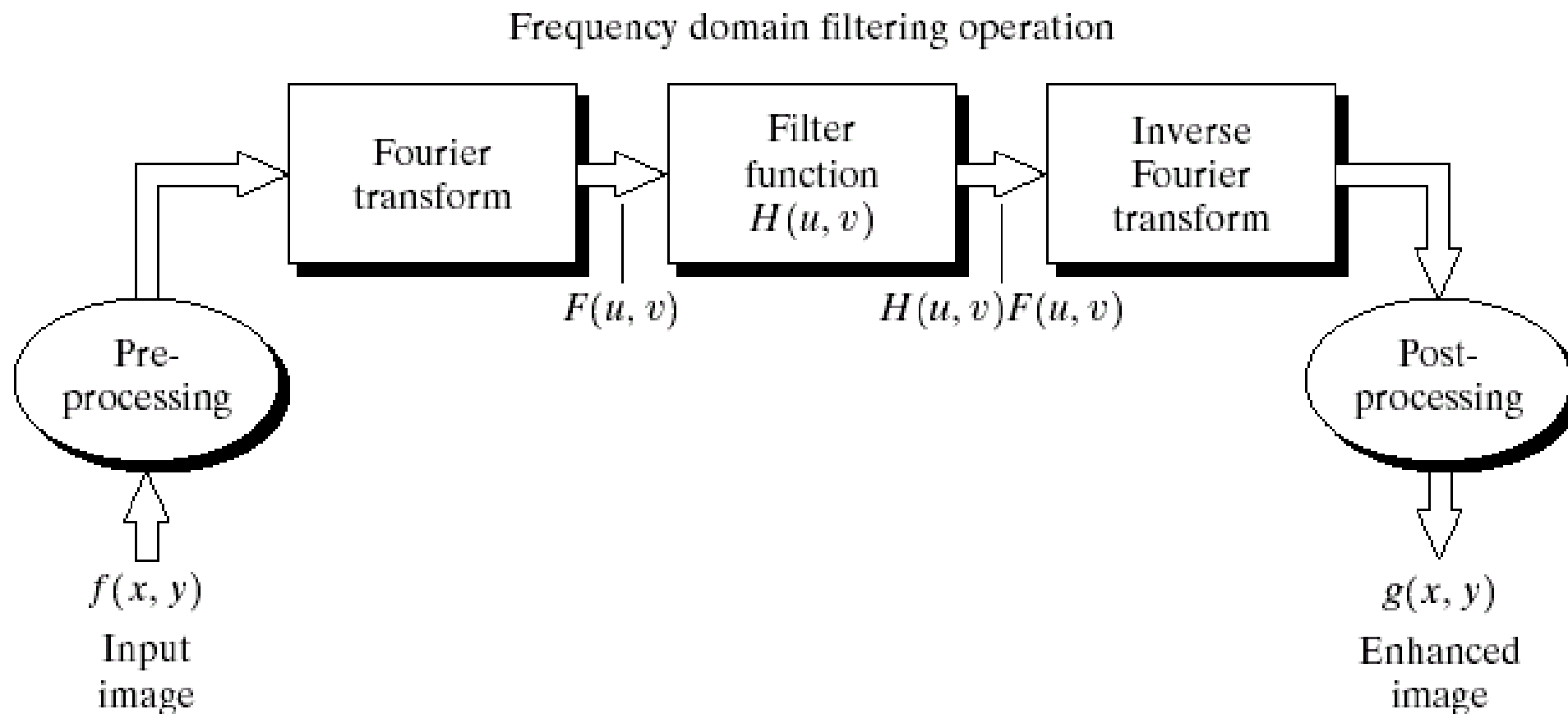


FIGURE 4.5 Basic steps for filtering in the frequency domain.

저주파 통과 필터 (Low Pass Filter)

❖ 저주파 영역은 통과시키고, 고주파 영역은 제거

❖ 저주파 통과 필터의 종류

- Ideal low pass filter
- Butterworth low pass filter
- Gaussian low pass filter

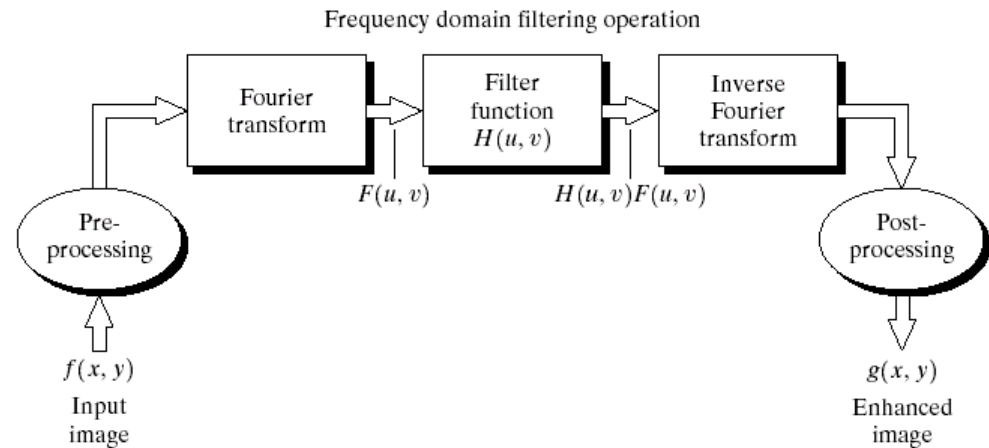


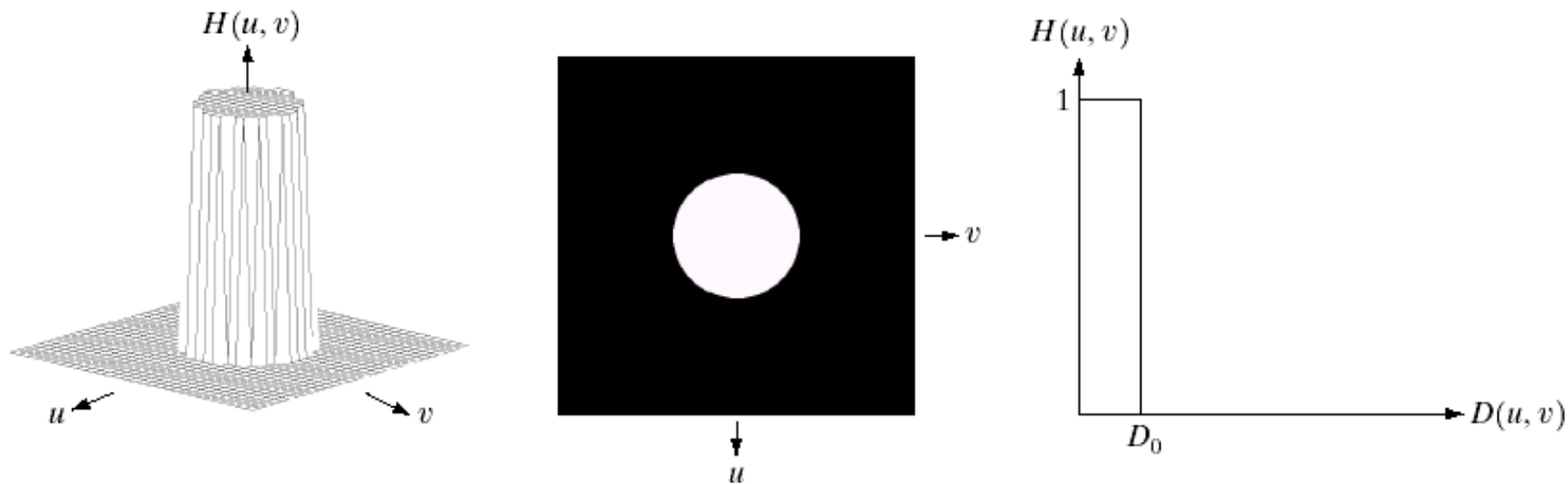
FIGURE 4.5 Basic steps for filtering in the frequency domain.

Low Pass Filter

Ideal low pass filter

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

D_0 : 절단주파수



a b c

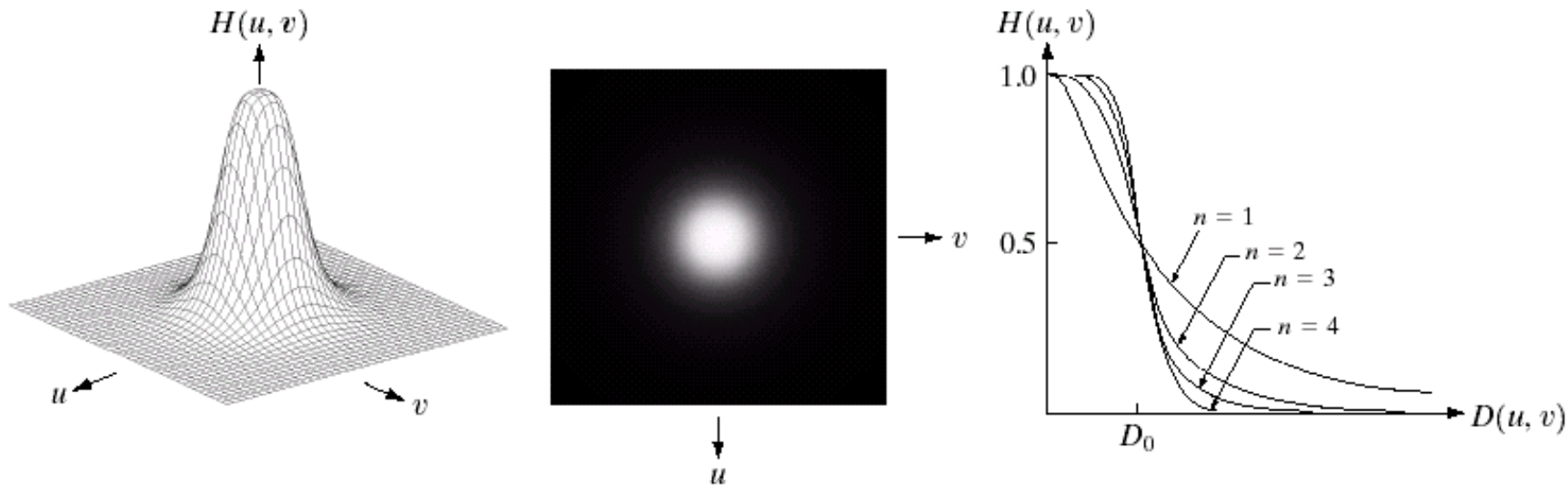
FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Low Pass Filter

Butterworth low pass filter

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

D_0 : 절단주파수



a b c

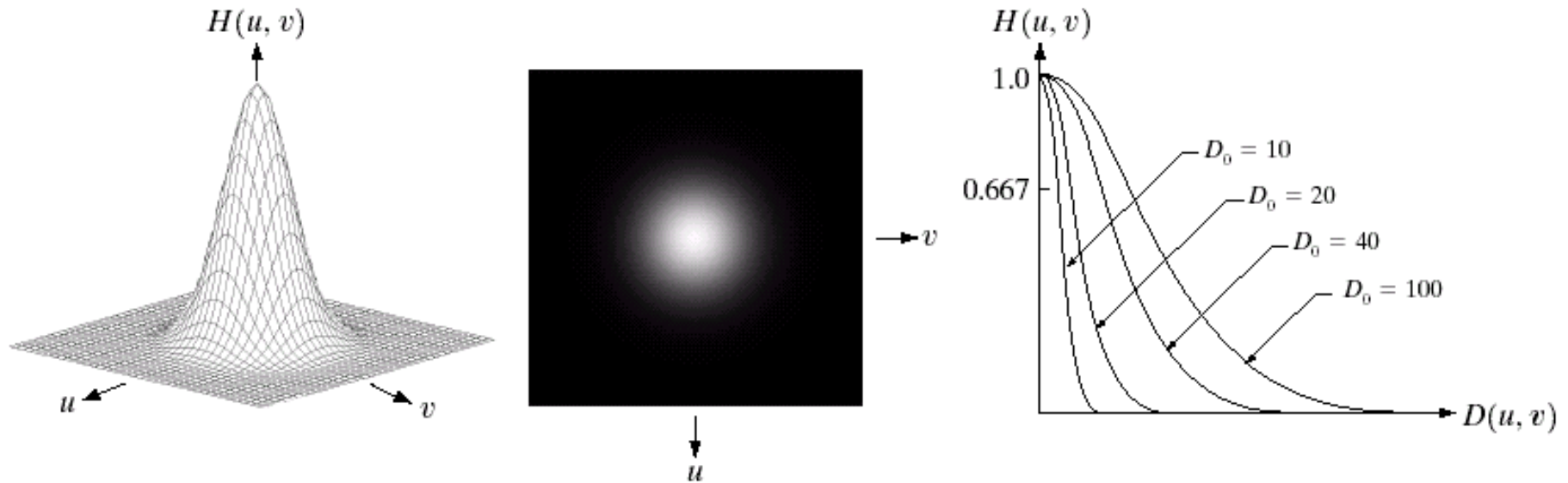
FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Low Pass Filter

Gaussian low pass filter

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

D_0 : 표준편차



a b c

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

Low Pass Filter

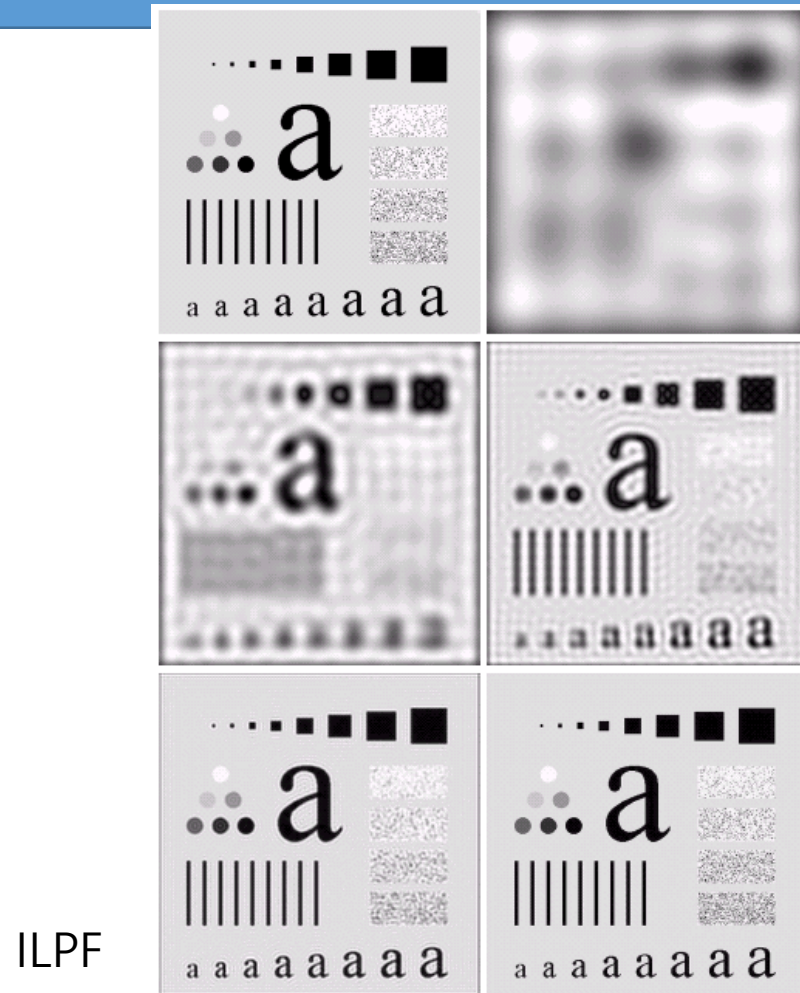
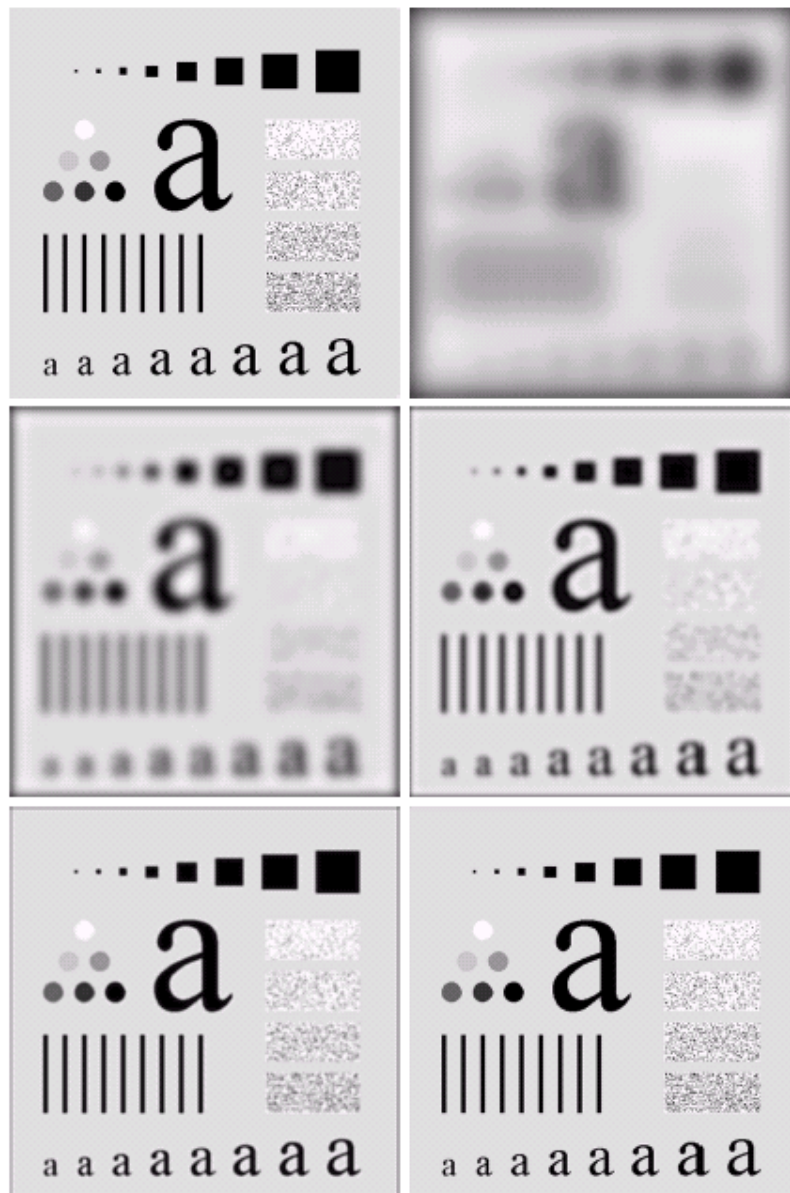


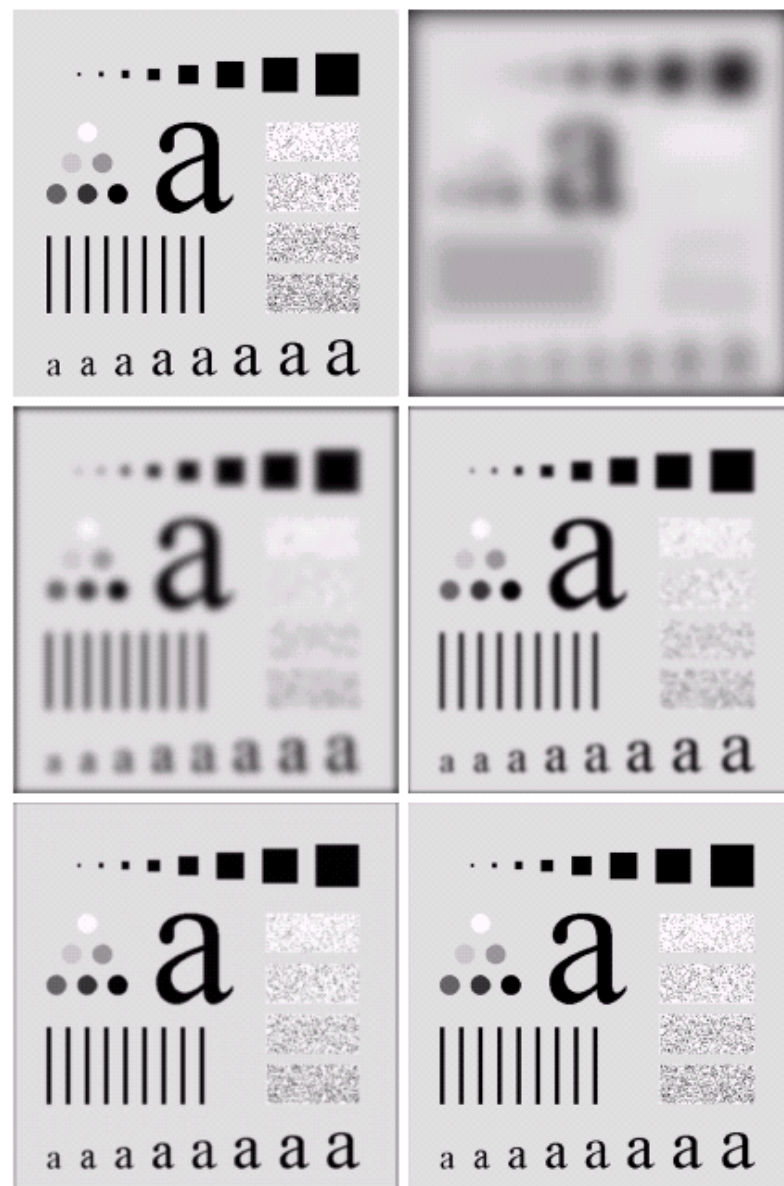
FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

BLPF



a b
c d
e f

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



GLPF

a b
c d
e f

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

고주파 통과 필터 (High Pass Filter)

❖ 고주파 영역은 통과시키고, 저주파 영역은 제거

$$H_{highpass}(u, v) = 1 - H_{Lowpass}(u, v)$$

❖ 고주파 통과 필터의 종류

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

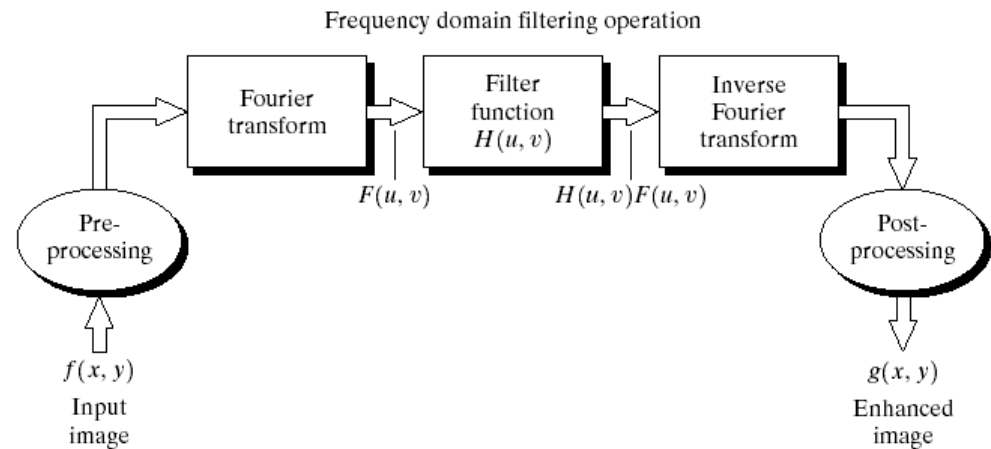
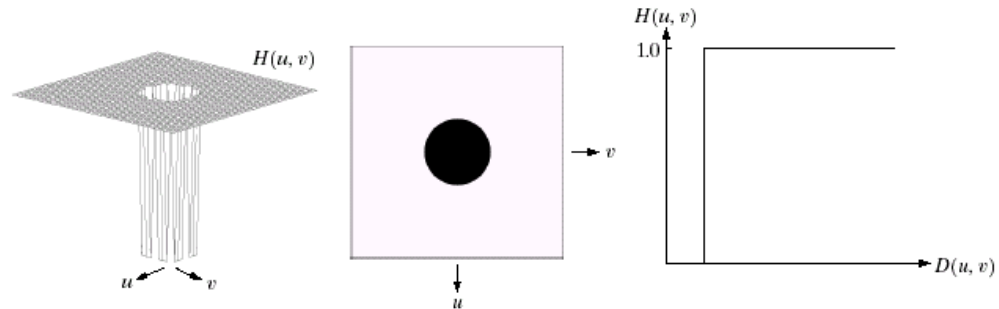


FIGURE 4.5 Basic steps for filtering in the frequency domain.

High Pass Filter

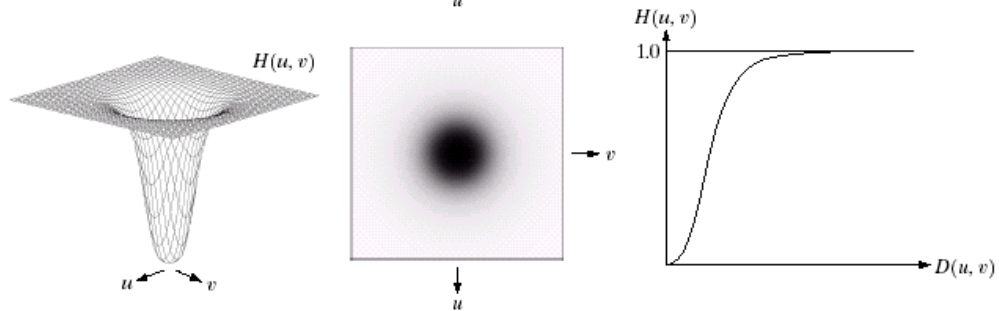
Ideal high pass filter

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



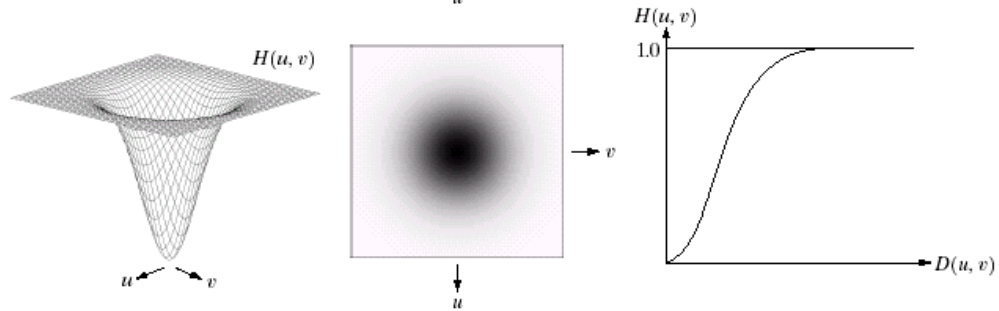
Butterworth high pass filter

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



Gaussian high pass filter

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



a b c
d e f
g h i

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

High Pass Filter

Ideal high pass filter

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

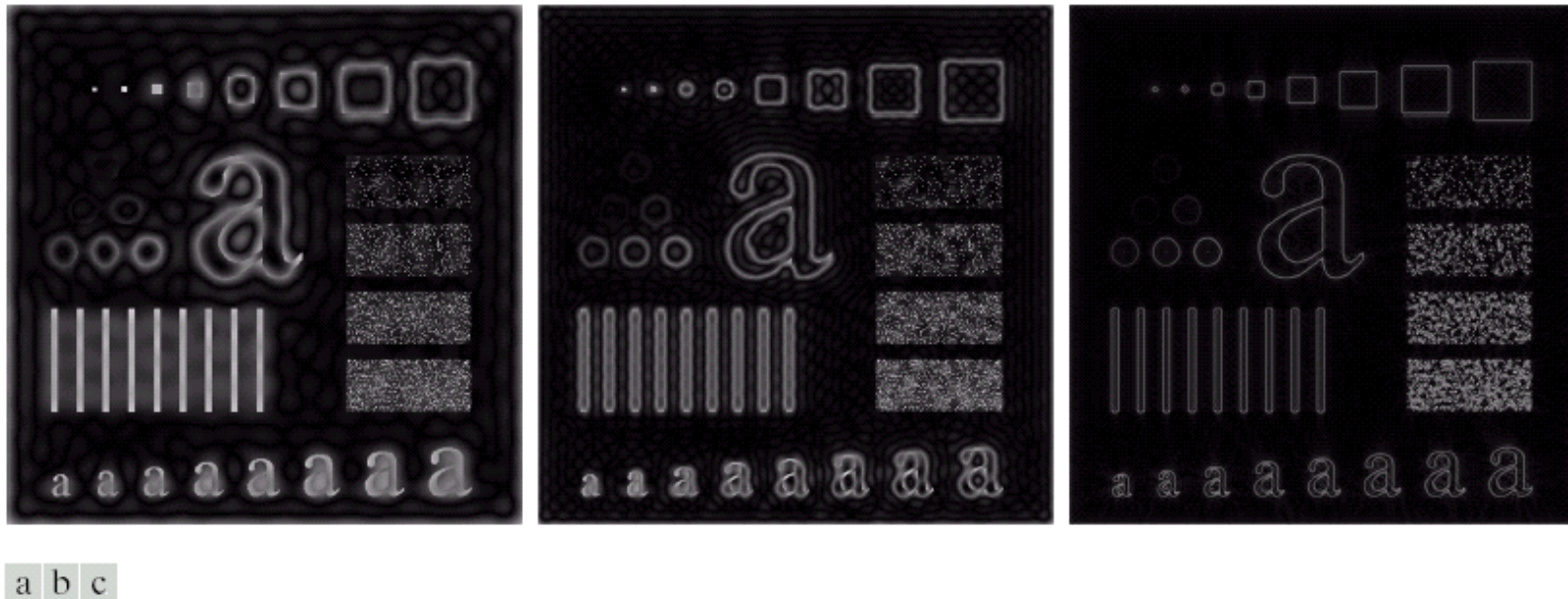


FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15, 30$, and 80 , respectively. Problems with ringing are quite evident in (a) and (b).

High Pass Filter

Butterworth high pass filter

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

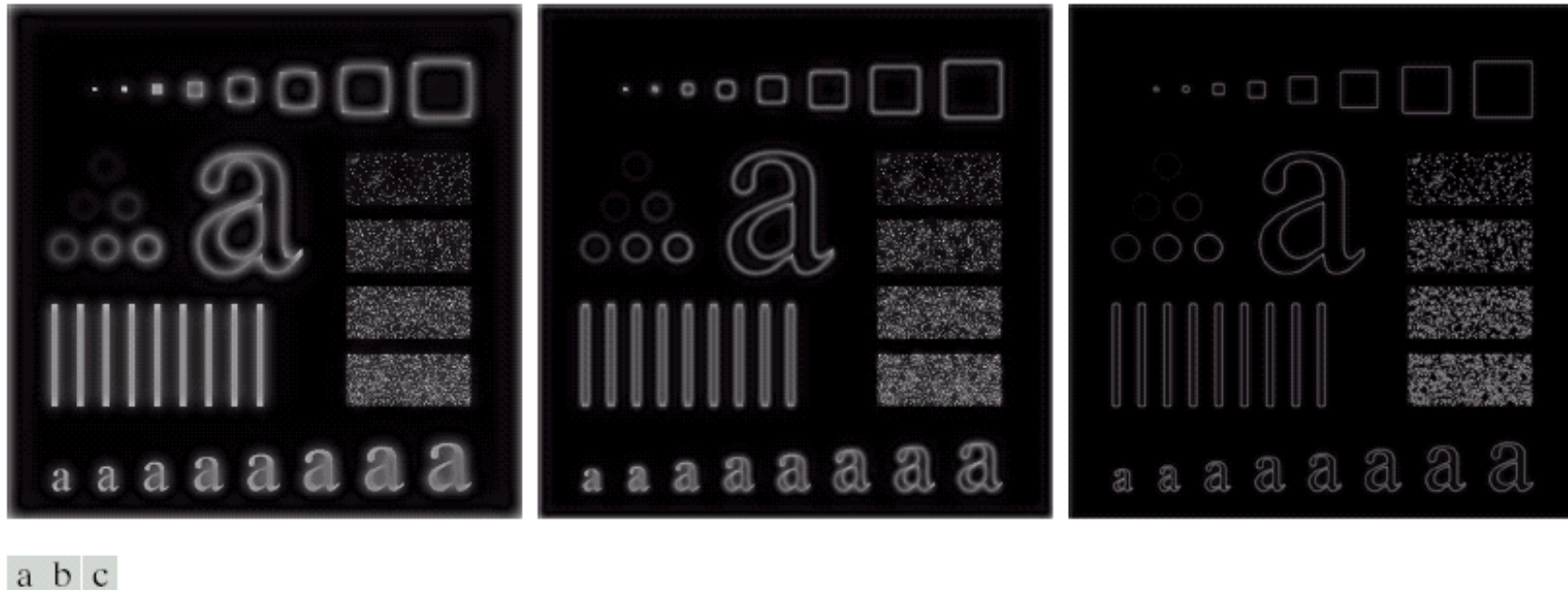


FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

High Pass Filter

Gaussian high pass filter

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

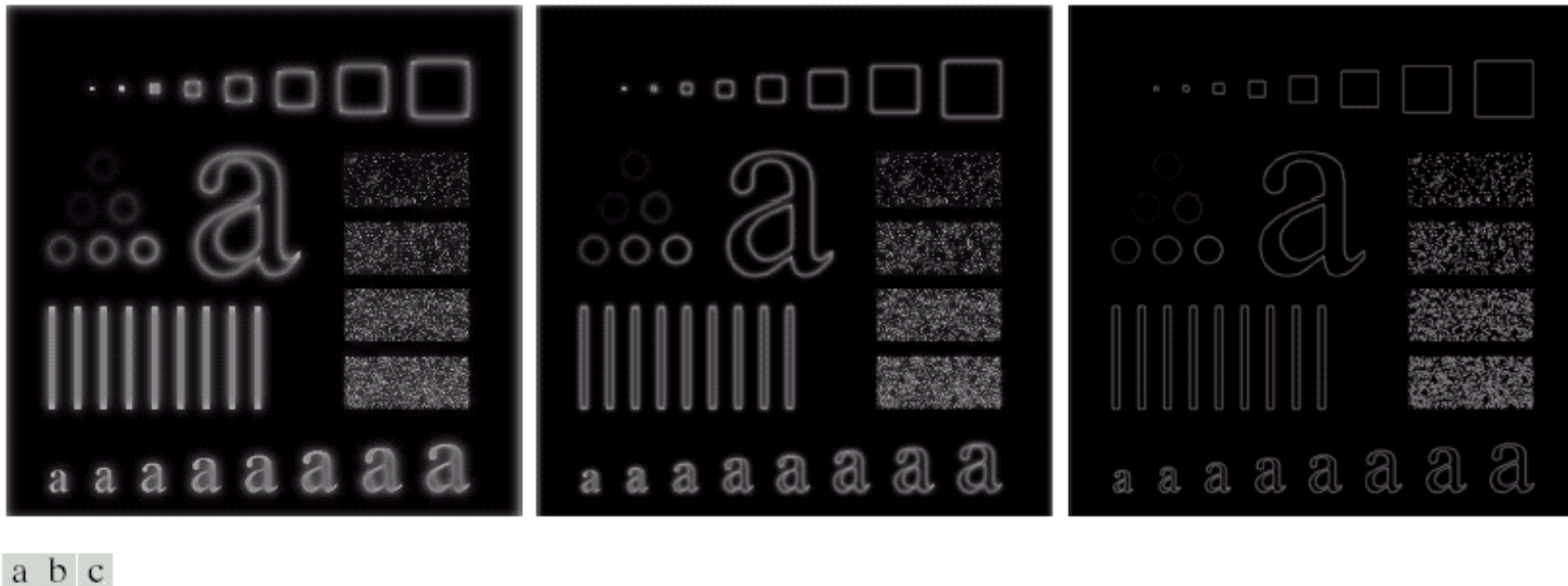


FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

DFT를 이용한 회선과 상관

회선 $f(x, y) * g(x, y) = IDFT(F(u, v)G(u, v))$

$F(u, v)$ 와 $G(u, v)$ 의 곱을 푸리에 역변환하면, 두 영상의 회선과 같다.

곱은 행렬곱셈이 아니라 대응되는 요소끼리의 곱셈임

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

$F(u, v)$ 와 $G^*(u, v)$ 의 곱을 푸리에 역변환하면, 두 영상의 상관과 같다.

위상 상관 (Phase Correlation)

→ 스펙트럼 $|F(u, v)G^*(u, v)|$ 에 의해 정규화하여 계산

$$PhaseCorr(f(x, y), g(x, y)) = IDFT\left(\frac{F(u, v)G^*(u, v)}{|F(u, v)G^*(u, v)|}\right)$$

OpenCV DFT

❖ DFT를 이용하여 회선, 상관 연산하기

```
void mulSpectrums(  
    InputArray a, InputArray b, // dft 함수에 의해 전방 푸리에 변환된 주파수 영상  
    OutputArray c,  
    int flags,                // DFT_ROWS 이면 a, b의 각 행이 독립적으로 푸리에 변환  
    bool conjB1)=false // 켈레 복소수 여부  
)
```

- 1) conjB=false 이면, $c = a * b$ (회선 연산)
conjB=true 이면, $c = a \bullet b$ (상관 연산)

회선 $f(x, y) * g(x, y) = IDFT(F(u, v)G(u, v))$

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

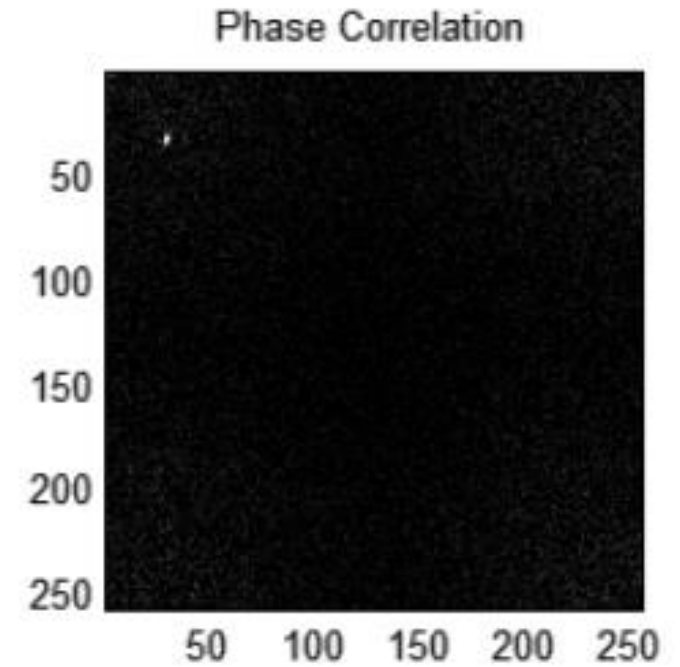
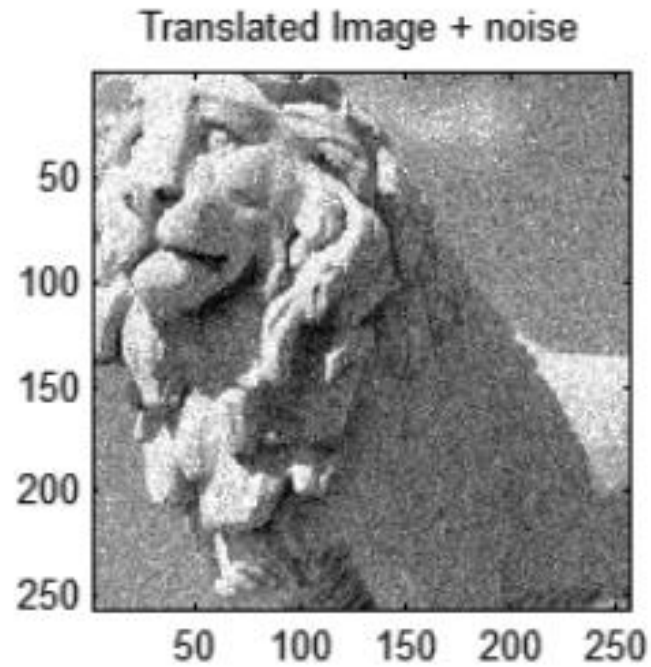
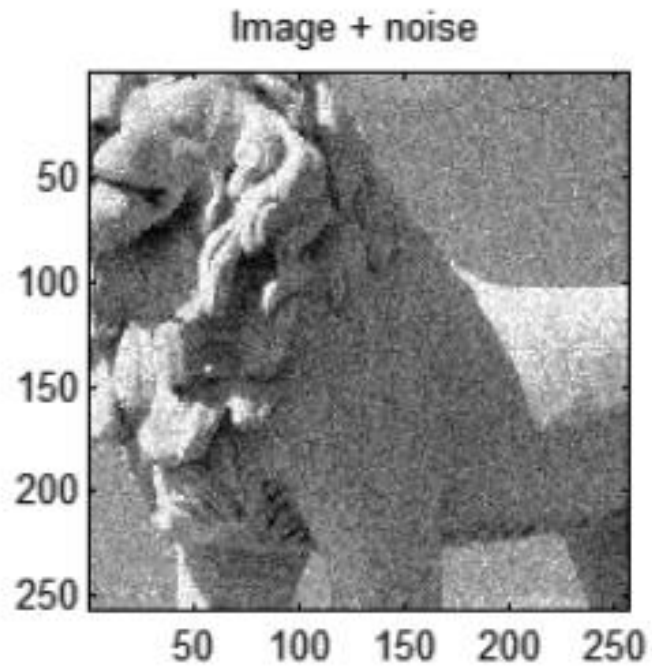
OpenCV DFT

❖ DFT를 이용하여 위상상관 구하기

```
Point2D1) phaseCorrelate (  
    InputArray src1, InputArray src2,  
    InputArray window = noArray() // 경계부분에서 에지 약화를 위한 윈도우  
)
```

- 1) 반환 값 : 위상상관 행렬의 4개 부분영역을 대각선 방향으로 교환하여 저장한 후, minMaxLoc 함수로 찾은 최대값 위치 peakLoc을 weightedCentroid 함수로 부화소 수준의 위치 t로 계산하고, 영상의 중심에 대한 상대위치인 (center-t)를 반환함

위상 상관의 응용




Summary

❖ 공간 주파수

- 저주파 영역, 고주파 영역

❖ 푸리에 변환 vs. 역변환

- 모든 신호는 다양한 주파수를 갖는 주기함수의 합으로 표현 가능
- 공간 도메인 영상  주파수 도메인 영상

❖ 주파수 필터링

- 주파수 필터링 후 역변환에 의해 영상 변환

❖ DFT를 이용한 회선과 상관

- 회선: $f(x, y) * g(x, y) = IDFT(F(u, v)G(u, v))$
- 상관: $f(x, y) \circ g(x, y) = IDFT(F(u, v)G^*(u, v))$