

제8장 주파수 영역에서의 처리

영상 처리

청주대학교 대학원 전자공학과
한철수

목차

- 푸리에 변환
- 고속 푸리에 변환
- 고속 푸리에 역변환
- 주파수 영역에서의 처리
- 이산 코사인 변환

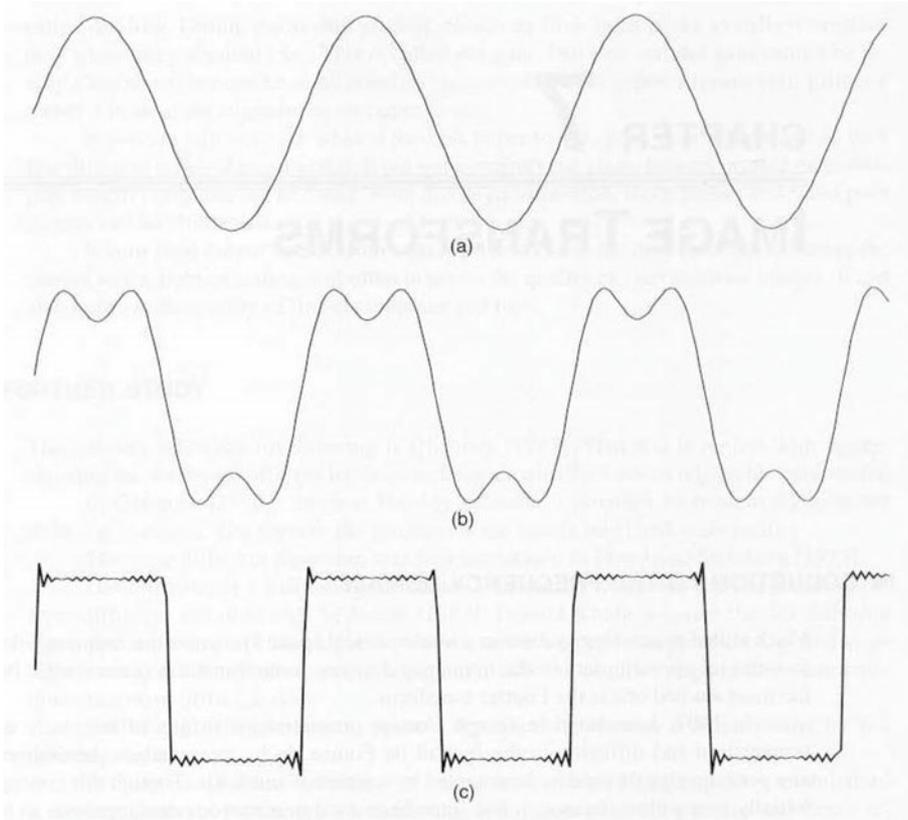
주파수 영역

- 지금까지는 공간 영역에서의 영상 처리를 다루었음
- 공간 영역의 문제를 주파수 영역으로 변환하여 다룰 수 있음
 - 주파수는 공간 영역에서 밝기나 색상의 변화 정도를 나타냄
 - 영상으로부터 고주파 성분을 제거
 - 영상이 부드러워짐
 - 영상으로부터 저주파 성분을 제거
 - 영상의 경계선이 강조됨

푸리에 변환

- 신호를 주파수 영역으로 변환하는 대표적인 방법임
- 임의의 주기적인 신호는 연속된 사인곡선의 조합으로 표현될 수 있다는 이론에 근거함

신호의 주파수 성분의 예



(a) $\sin(x)$

(b) $\sin(x) + \sin(3x)/3$

(c) $\sin(x) + \sin(3x)/3 + \sin(5x)/5$

2차원 푸리에 변환 (공간 영역 \rightarrow 주파수 영역)

- 2차원 연속 함수 $f(x, y)$ 에 대한 푸리에 변환
- 연속 함수를 다루는 것이므로 연속 푸리에 변환이라 함

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

$$j = \sqrt{-1}, \quad e^{\pm jx} = \cos(x) \pm j \sin(x)$$

2차원 푸리에 역변환 (주파수 영역 \rightarrow 공간 영역)

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

$$j = \sqrt{-1}, \quad e^{\pm jx} = \cos(x) \pm j \sin(x)$$

이산 푸리에 변환(DFT)과 역변환(IDFT)

- 디지털 영상 처리에서는 표본화된 이산 신호를 다루므로 이산 푸리에 변환을 사용함

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

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

M, N : 영상의 넓이와 높이

고속 푸리에 변환 (1/2)

- N개의 데이터에 대한 1차원 DFT
 - N^2 의 시간 복잡도
- $M \times N$ 크기의 영상에 대한 2차원 DFT
 - $M^2 \times N^2$ 의 시간 복잡도
 - 많은 시간이 소요되어 실시간으로 적용하기 어려움

고속 푸리에 변환 (2/2)

- 1차원 FFT
 - N 길이의 DFT는 N/2 길이의 두 개의 DFT의 합과 같음
 - 시간 복잡도가 $N \log_2 N$ 로 작아져서 매우 빠름
- 2차원 FFT
 - 2차원 푸리에 변환을 1차원 푸리에 변환으로 분할 적용할 수 있음
 - 영상의 각 행에 1차원 FFT를 적용한 다음에 계산 결과의 각 열에 1차원 FFT를 적용함
 - 시간 복잡도

$$\frac{MN}{2} \log_2 MN$$

- FFT의 제한점
 - 영상의 크기를 나타내는 M과 N의 값이 2의 지수승이어야 함

연산 예

- 영상의 크기
– 1024x1024

- DFT

$$1024^2 \times 1024^2 = (2^{10})^2 \times (2^{10})^2 = 2^{40}$$

- FFT

$$\frac{1024 \times 1024}{2} \log_2(1024 \times 1024) = \frac{2^{20}}{2} \log_2(2^{20}) = 2^{20} \times 10$$

1차원 FFT 알고리즘 (1/12)

- 아래와 같이 가정
 - 입력 신호 값: $x(0), x(1), \dots, x(N-1)$
 - 푸리에 변환 결과값: $X(0), X(1), \dots, X(N-1)$
- 1차원 DFT 변환식

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{nk}{N}} = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

$$\text{여기에서 } W_N^{nk} = e^{-j2\pi \frac{nk}{N}}$$

$$j = \sqrt{-1}, \quad e^{\pm jx} = \cos(x) \pm j \sin(x)$$

1차원 FFT 알고리즘 (2/12)

- 짝수항과 홀수항을 분리

$$X(k) = \sum_{n=0,2,\dots}^{N-2} x(n) W_N^{nk} + \sum_{n=1,3,\dots}^{N-1} x(n) W_N^{nk}$$

- n 이 짝수일 때 $n=2r$ 로 나타내고, n 이 홀수일 때 $n=2r+1$ 로 나타내면

$$\begin{aligned} X(k) &= \sum_{r=0}^{(N/2)-1} x(2r) W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x(2r+1) W_N^{(2r+1)k} \\ &= \sum_{r=0}^{(N/2)-1} x(2r) (W_N^2)^{rk} + \sum_{r=0}^{(N/2)-1} x(2r+1) W_N^{2rk} W_N^k \\ &= \sum_{r=0}^{(N/2)-1} x(2r) (W_N^2)^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1) (W_N^2)^{rk} \end{aligned}$$

1차원 FFT 알고리즘 (3/12)

- 이때

$$W_N^2 = e^{-j2\pi\frac{2}{N}} = e^{-j2\pi\frac{1}{(N/2)}} = W_{N/2}^1$$

- 따라서

$$\begin{aligned} X(k) &= \sum_{r=0}^{(N/2)-1} x(2r) W_{N/2}^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1) W_{N/2}^{rk} \\ &= G(k) + W_N^k H(k) \end{aligned}$$

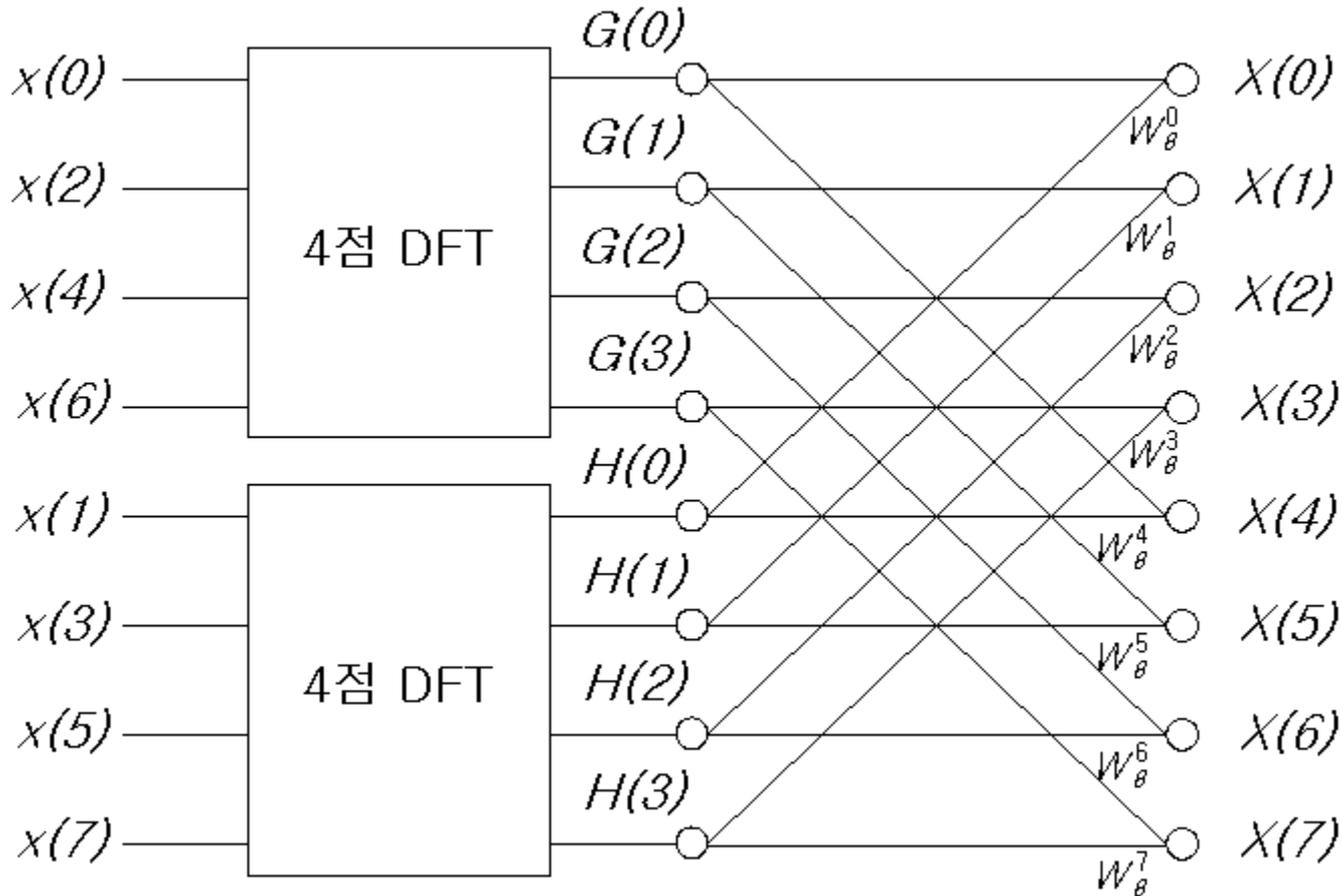
$X(k)$: N개의 데이터에 대한 1차원 DFT

$G(k)$: N/2개의 데이터에 대한 1차원 DFT

$H(k)$: N/2개의 데이터에 대한 1차원 DFT

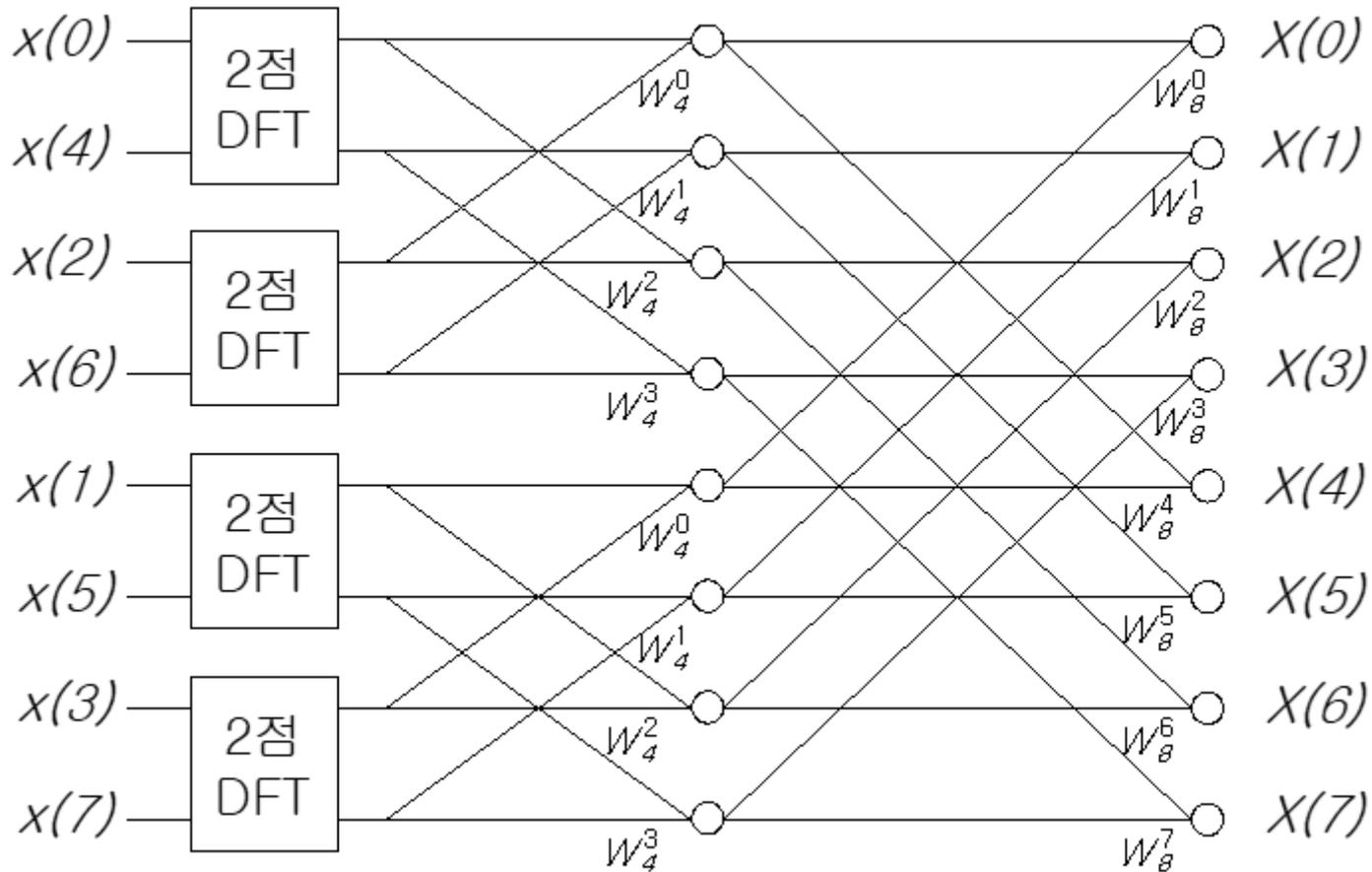
1차원 FFT 알고리즘 (4/12)

- N=8인 경우, 1차 분할 후의 $X(k)$ 연산에 대한 흐름도



1차원 FFT 알고리즘 (5/12)

- N=8인 경우, 2차 분할 후의 $X(k)$ 연산에 대한 흐름도



1차원 FFT 알고리즘 (6/12)

- N=2인 경우의 DFT

$$X(0) = x(0)W_2^0 + x(1)W_2^0 = x(0) + x(1)$$

$$X(1) = x(0)W_2^0 + x(1)W_2^1 = x(0) + x(1)W_2^1$$

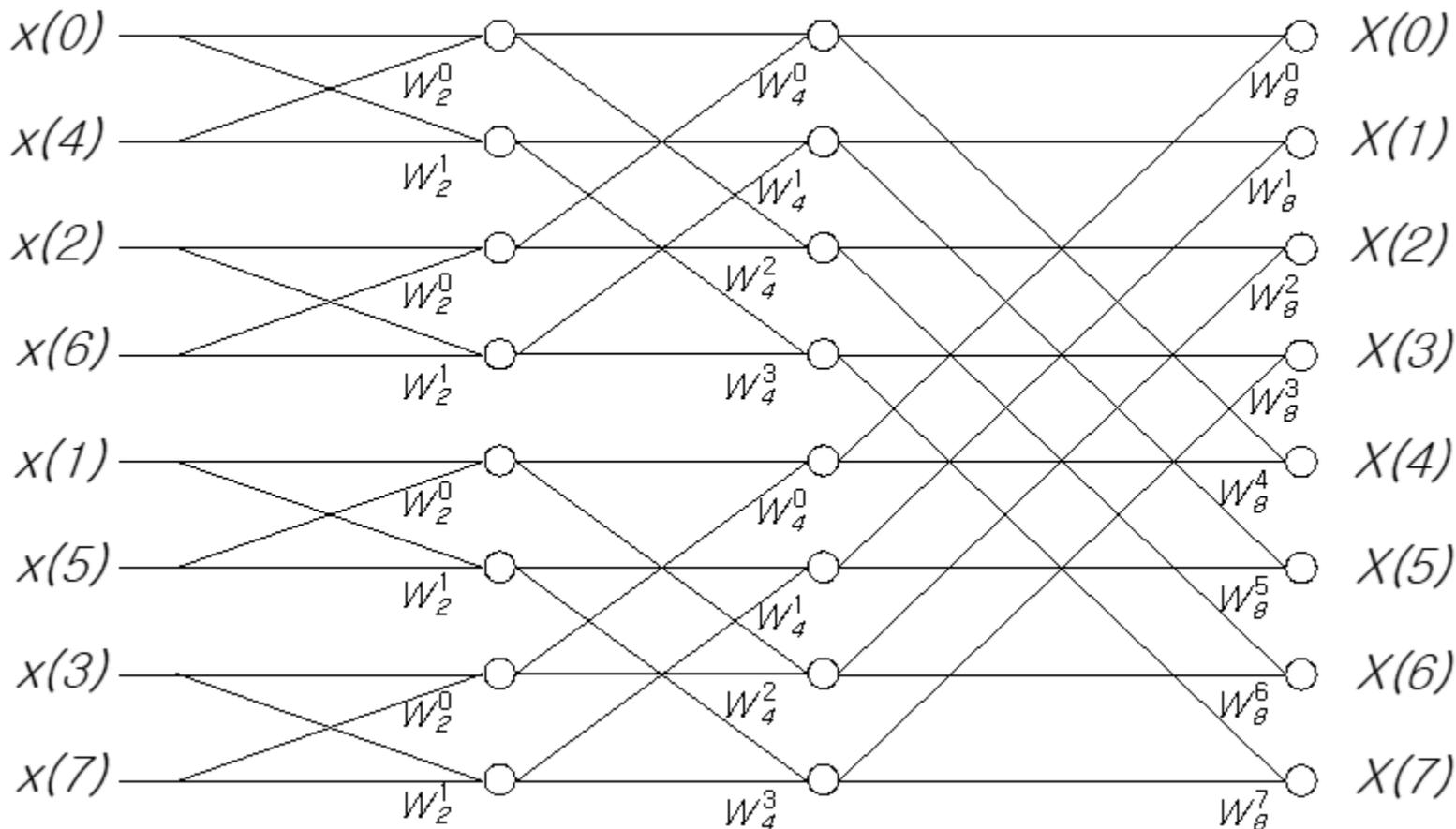
$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad W_N^{nk} = e^{-j2\pi\frac{nk}{N}}$$

$$W_2^0 = e^0 = \cos(0) - j \sin(0) = 1$$

1차원 FFT 알고리즘 (7/12)

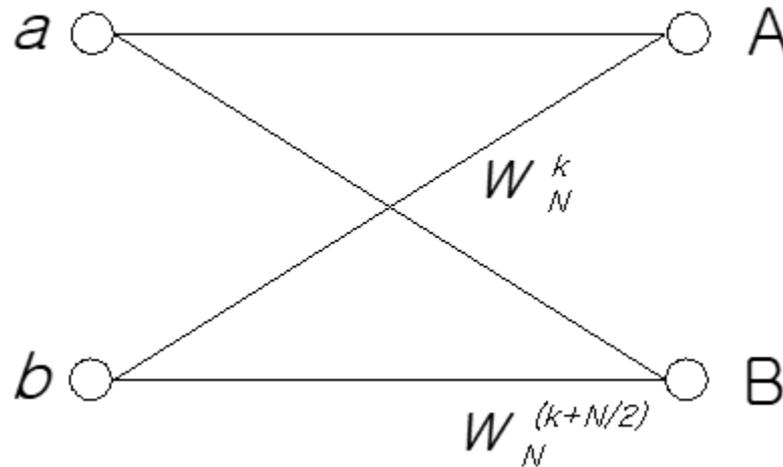
- N=8인 경우, 최종 FFT 흐름도

복잡도가 $N \log_2 N$



1차원 FFT 알고리즘 (8/12)

- 흐름도에서 각 노드는 아래 그림과 같이 나비 형태임
 - 모든 노드에서 W_N 값은 그림에 나타나 있는 형태를 가짐



기본적인 나비흐름도

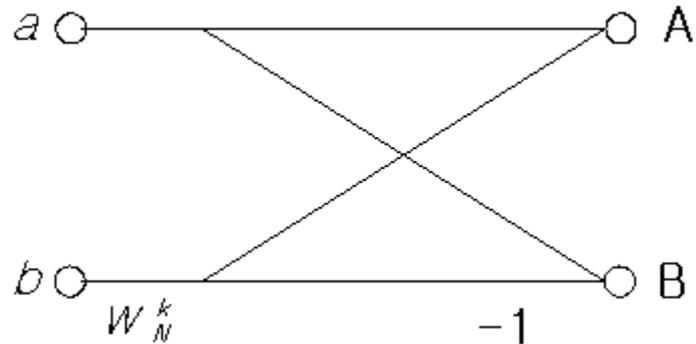
1차원 FFT 알고리즘 (9/12)

- $W_N^{(k+N/2)}$ 의 값은 아래와 같이 계산됨

$$\begin{aligned}W_N^{(k+N/2)} &= W_N^k W_N^{N/2} = W_N^k e^{-j2\pi(N/2)/N} \\ &= W_N^k e^{-j\pi} = W_N^k (\cos(\pi) - j\sin(\pi)) \\ &= -W_N^k\end{aligned}$$

1차원 FFT 알고리즘 (10/12)

- 나비 흐름도의 계산 식은 아래와 같이 간략화 됨



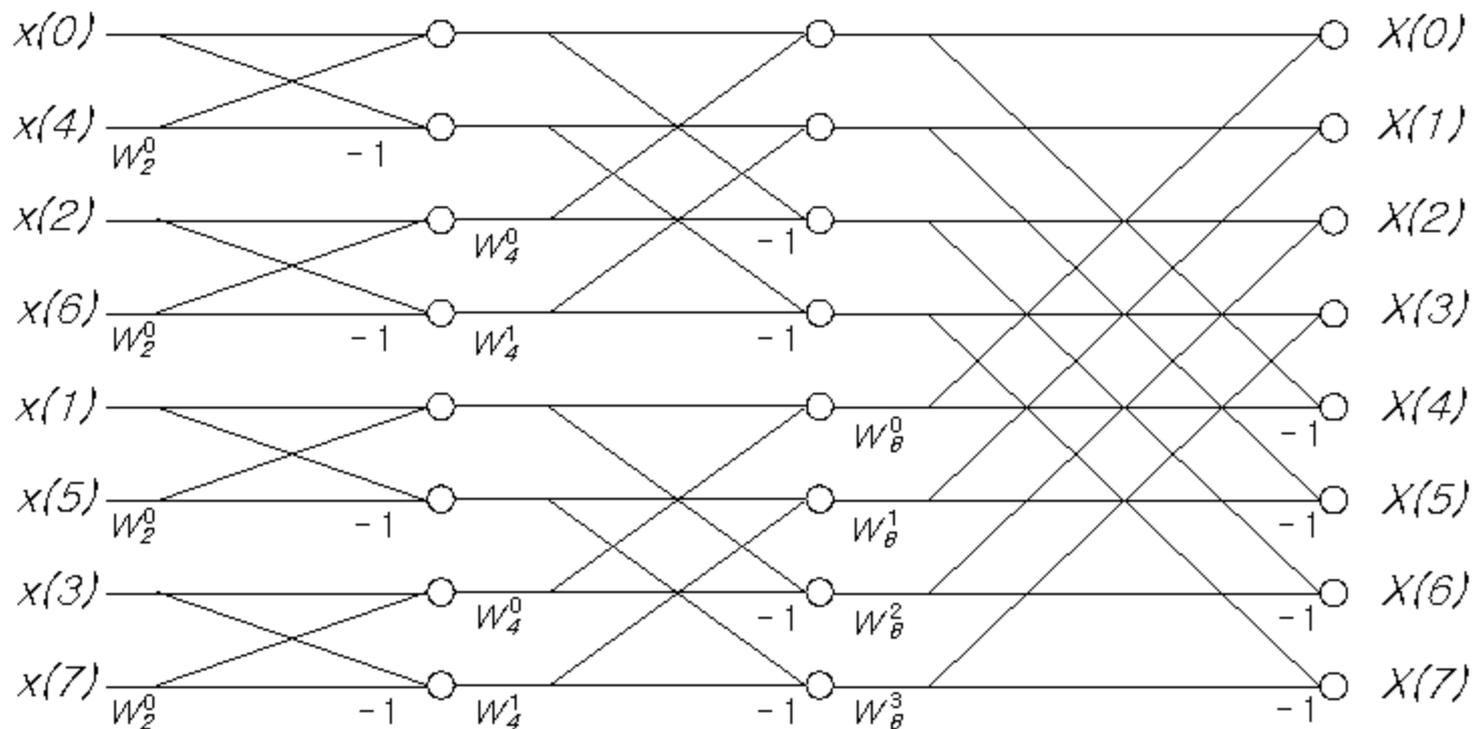
$$A = a + b W_N^k$$

$$B = a + b W_N^{(k+N/2)} = a - b W_N^k$$

1차원 FFT 알고리즘 (11/12)

- 간략화된 FFT 흐름도

복잡도가 $(N/2)\log_2 N$



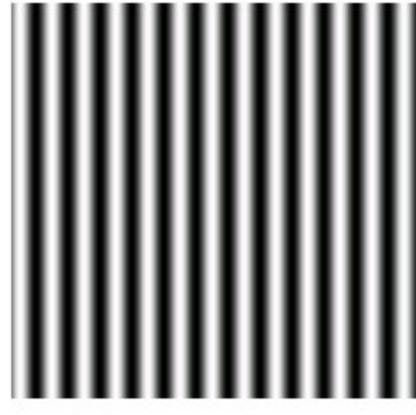
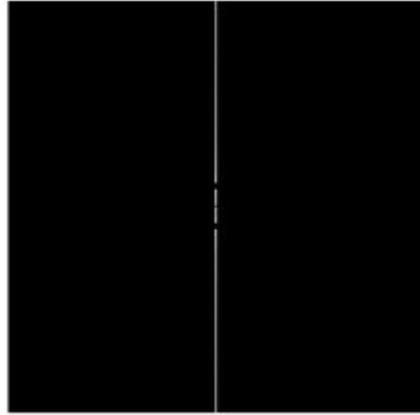
1차원 FFT 알고리즘 (12/12)

- 입력 데이터 재배열

- $N=8$ 인 경우의 FFT 흐름도에서 입력 데이터의 순서는 $x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)$ 과 같이 일정하지 않음
- 각 인덱스의 이진수 비트들을 역순으로 표현하면 이러한 순서를 간단하게 생성할 수 있음

인덱스	이진수 표현	역순의 이진수	변환된 인덱스
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

푸리에 변환 예 (1/3)



입력 영상

horizontal.raw

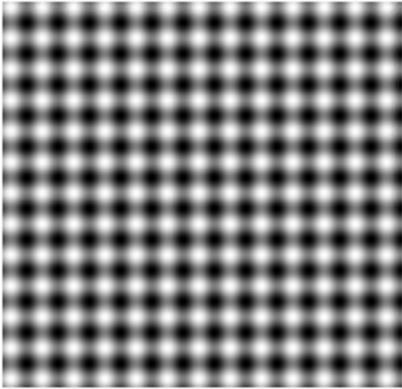
변환 결과

입력 영상

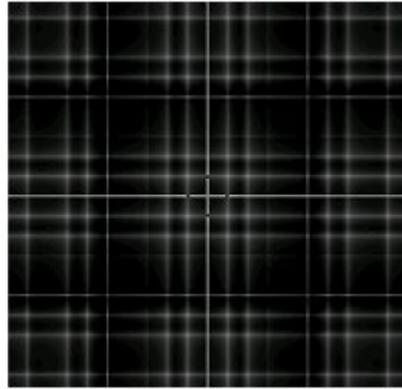
vertical.raw

변환 결과

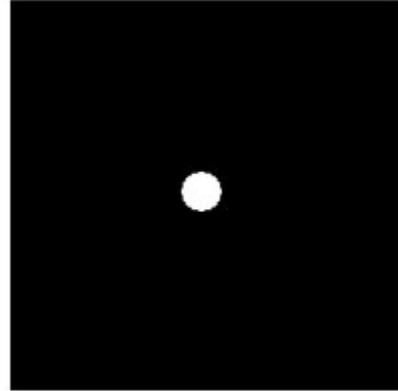
푸리에 변환 예 (2/3)



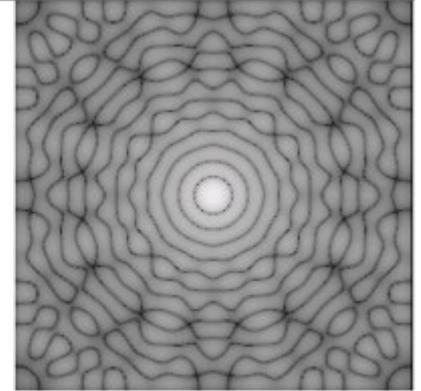
입력 영상
cross.raw



변환 결과

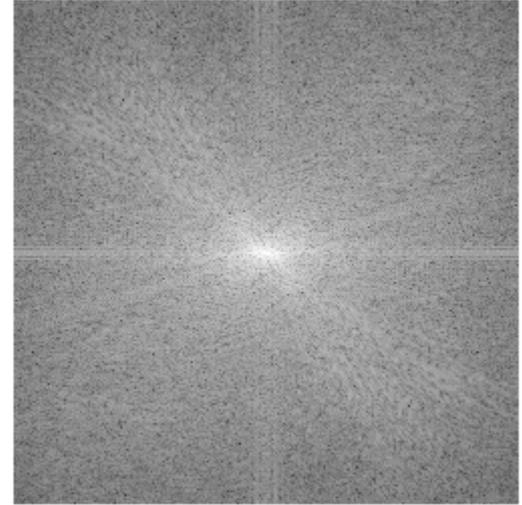


입력 영상
circle.raw



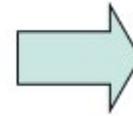
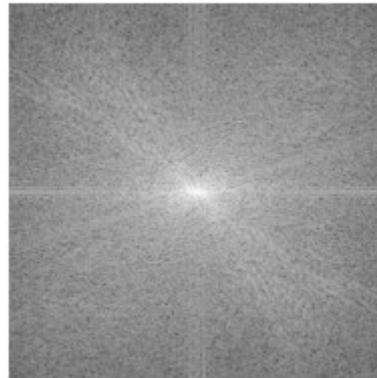
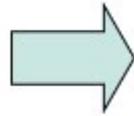
변환 결과

푸리에 변환 예 (3/3)



푸리에 역변환(IFFT)

- 주파수 영역에서 공간 영역으로 변환



푸리에 변환

푸리에 역변환

저주파 통과 필터

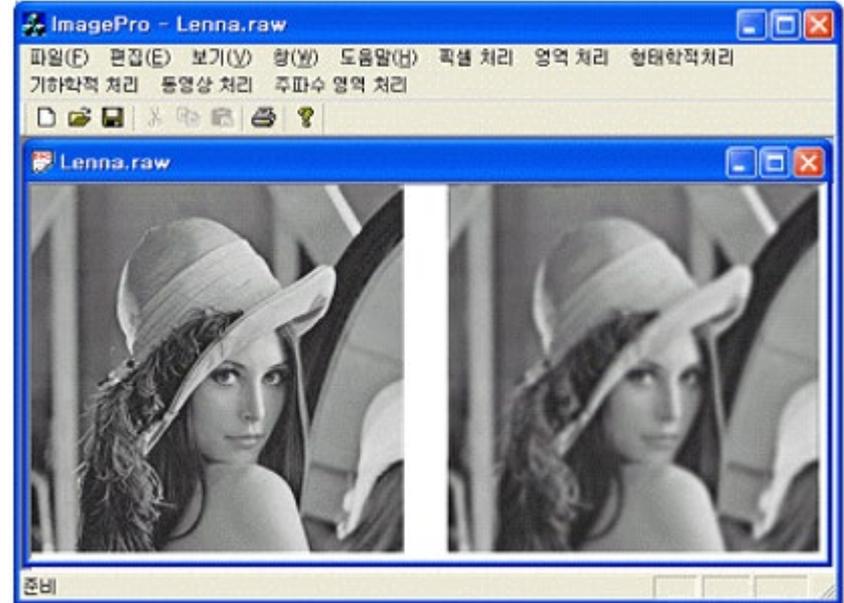
- 고주파 성분을 제거
- 버터워스 저주파 통과 필터

$$B(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2N}}$$

$$D(u, v) = \sqrt{(u^2 + v^2)} \quad D_0, N: \text{상수}$$

- 필터 적용 방법
 - FFT 변환 결과에 필터의 값을 곱한 다음에 역변환함

$$R(u, v) = F(u, v) \times B(u, v)$$



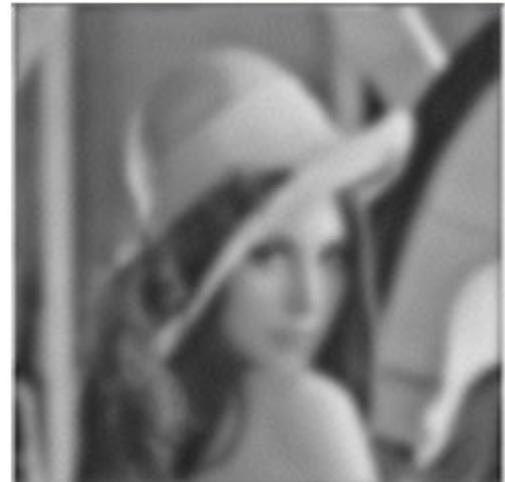
D_0 값의 변화에 따른 결과 영상



$D_0=64$



$D_0=32$



$D_0=16$

N 값의 변화에 따른 결과 영상



N=4



N=8

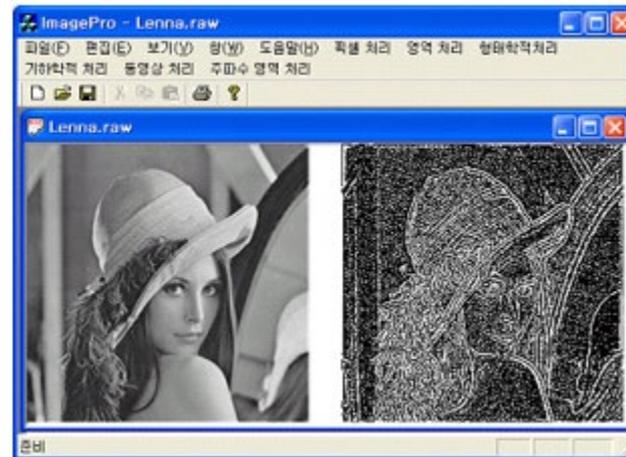


N=16

고주파 통과 필터

- 버터워스 고주파 통과 필터

$$B(u,v) = \frac{1}{1 + \left(\frac{D_0}{D(u,v)}\right)^{2N}}$$



D_0 값의 변화에 따른 결과 영상



$D_0=96$



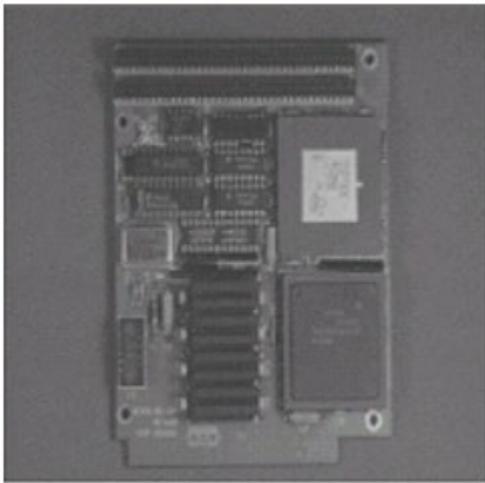
$D_0=128$



$D_0=160$

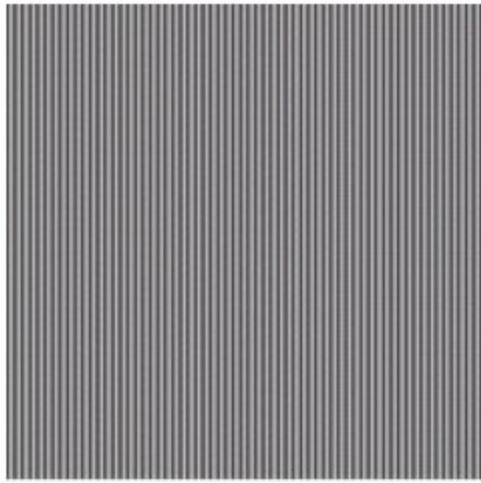
잡음 제거

- 아래 그림과 같이 일정한 형태의 잡음이 들어간 영상에서 잡음을 제거하는데 주파수 영역에서의 처리가 효율적임

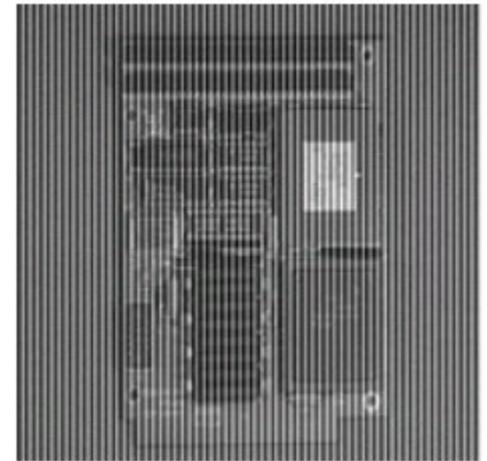
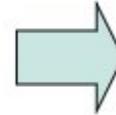


원영상

+



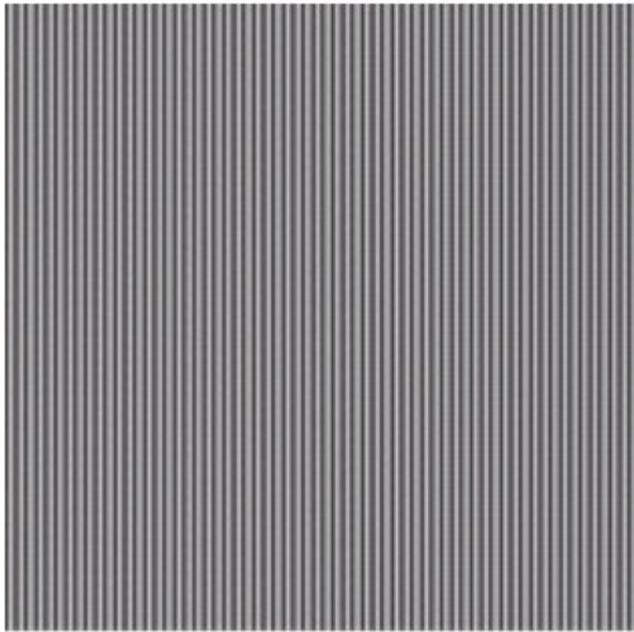
잡음영상



잡음이 들어간 영상

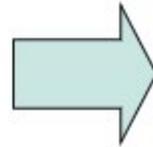
noise.pgm

잡음 영상에 대한 스펙트럼

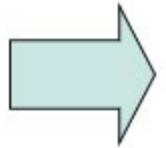
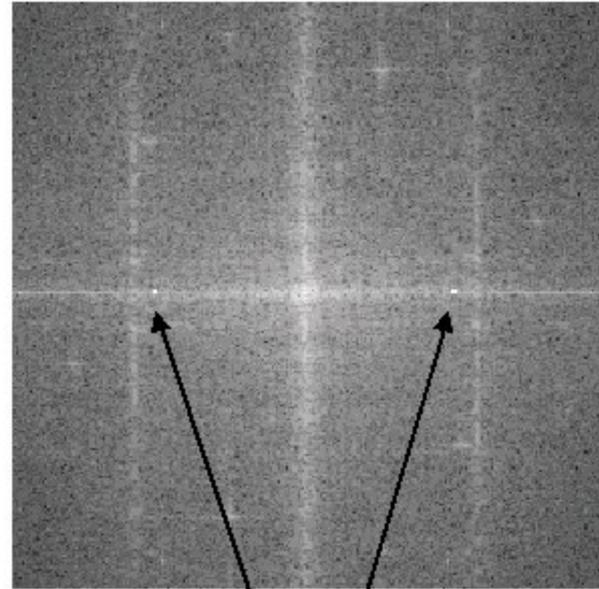
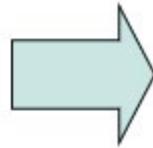
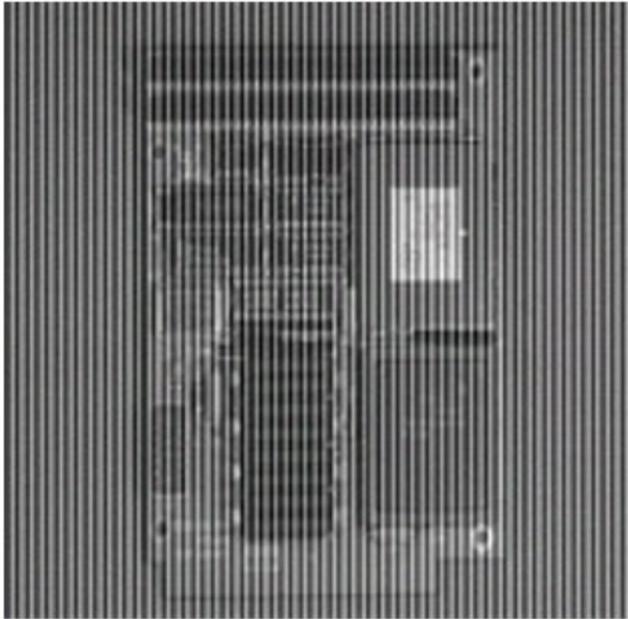


세로줄.pgm

푸리에 변환

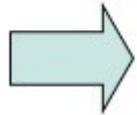


입력 영상에 대한 스펙트럼

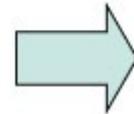
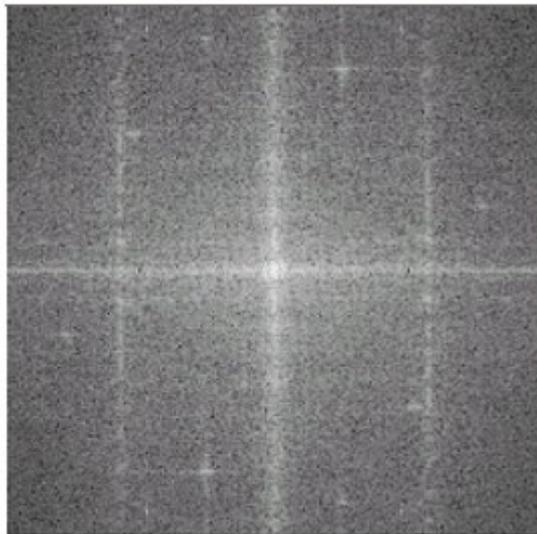


잡음 성분

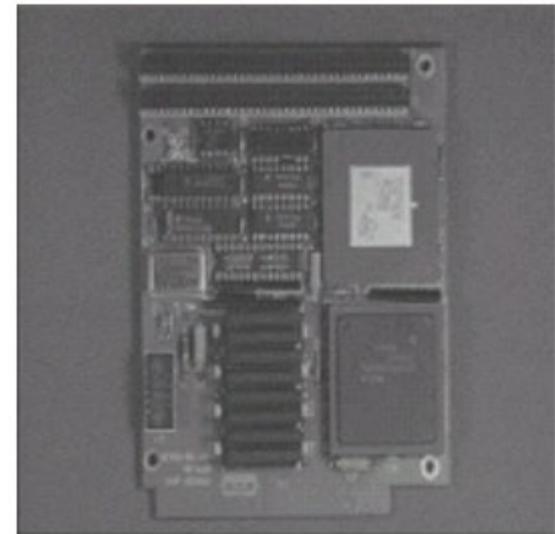
잡음 성분 제거



잡음 성분
제거



푸리에
역변환



잡음이 제거된 영상

이산 코사인 변환(DCT)와 역변환(IDCT)

- 많은 영상 압축 알고리즘의 기초가 됨
- 푸리에 변환에 비해 DCT의 장점은 복소수 계산을 안함

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

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

여기에서 $C(z) = \begin{cases} 1/\sqrt{2} & z = 0 \\ 1 & z > 0 \end{cases}$

2차원 DCT의 직접 계산

- 앞의 식을 이용하여 2차원 DCT를 직접 계산할 수 있음
- 많은 시간이 소요됨
- 1차원 DCT를 이용한 구현
 - 각 행에 1차원 DCT를 적용한 다음에 그 결과의 각 열에 1차원 DCT를 적용하면 보다 빠르게 구현 가능

1차원 DCT식과 IDCT식

$$F(u) = \frac{2}{\sqrt{N}} C(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

$$f(x) = \frac{2}{\sqrt{N}} \sum_{u=0}^{N-1} C(u) F(u) \cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

여기에서 $C(u) = \begin{cases} 1/\sqrt{2} & u = 0 \\ 1 & u > 0 \end{cases}$

질문

Q & A