Lecture 8: FFT and DCT

The numerical computation of spectrum estimation is commonly conducted with the use of the *Discrete Fourier Transform* (DFT). The DFT is a linear operator, which converts an N-point sequence $\{x(n)\}$ into the N-point spectral sequence $\{X(k)\}$,

$$X(k) = \sum_{n=0}^{N-1} x(n)\, exp(-j2\pi nk/N)$$

$$= \sum_{n=0}^{N-1} x(n)\, w(n,k)$$

where

$$w(n,k) = exp(-j2\pi nk/N)$$

In terms of computation format, it can be described in the form of an NxN matrix operation. Thus, it involves N^2 multiplications. When the length of the sequence is large, the dimension of the matrix increases, and the required computation becomes more time consuming. To be computationally efficient, it is desirable to reduce the computation complexity, in terms of the total number of multiplications.

1. Time-decimation technique

Time decimation is a technique for the simplification of computation complexity of the *DFT*. To start, we first partition the sequence into the even and odd components

$$X(k) = \sum_{n=0}^{N-1} x(n)\, exp(-j2\pi nk/N)$$

$$= \sum_{n::even} x(n)\, exp(-j2\pi nk/N) + \sum_{n:odd} x(n)\, exp(-j2\pi nk/N)$$

After rearranging the index system, the equation becomes,

$$X(k) = \sum_{m:=0}^{N/2-1} x(2m)\, exp(-j2\pi(2m)k/N) \;+\; \sum_{m:=0}^{N/2-1} x(2m+1)\, exp(-j2\pi(2m+1)k/N)$$

$$= \sum_{m:=0}^{N/2-1} x(2m)\, exp(-j2\pi(2m)k/N)$$

$$+\; exp(j2\pi k/N) \sum_{m:=0}^{N/2-1} x(2m+1)\, exp(-j2\pi(2m)k/N)$$

$$= \sum_{m:=0}^{N/2-1} x(2m) exp(-j2\pi mk/(N/2))$$

$$+\; exp(-j2\pi k/N) \sum_{m:=0}^{N/2-1} x(2m+1) exp(-j2\pi mk/(N/2))$$

The first term is in the form of an *N/2*-point *DFT* of the even component of the sequence, and the second term is an *N/2*-point *DFT* of the odd component, multiplied by the term $exp(-j2\pi k/N)$.

$$X(k) \;=\; DFT\{\, x_{even}(n)\} + exp(-j2\pi k/N)[DFT\{\, x_{odd}(n)\}]$$

$$=\; X_1(k) + exp(-j2\pi k/N)\, X_2(k)$$

where $x_{even}(n)$ and $x_{odd}(n)$ are the *N/2*-point sequences formed by the even and off terms of the sequence $x(n)$ respectively, and $X_1(k)$ and $X_2(k)$ are the corresponding spectral sequences of the even and odd components. The term $exp(-j2\pi k/N)$ is a function of the spectral index *k*. Hence, the term $X_2(k)$ requires *N* multiplications by $exp(-j2\pi k/N)$, one for each *k*.

This implies each time-decimation step converts an *N*-point *DFT* operation into two *N/2*-point *DFT* operations plus *N* multiplications. In terms of computation complexity, the conversion changes it from N^2 multiplications to $2(N/2)^2+N$ multiplications.

2. Frequency-decimation technique

Frequency-decimation is a different approach to the regrouping of the sequence. It partitions the sequence to two components, the first half and the second half

$$X(k) = \sum_{n=0}^{N-1} x(n)\, exp(-j2\pi nk/N)$$

$$= \sum_{n=0}^{N/2-1} x(n)\, exp(-j2\pi nk/N) \;+\; \sum_{n:=N/2}^{N-1} x(n)\, exp(-j2\pi nk/N)$$

$$= \sum_{n=0}^{N/2-1} x(n)\, exp(-j2\pi nk/N) \;+\; \sum_{n=0}^{N/2-1} x(n+N/2)\, exp(-j2\pi(n+N/2)k/N)$$

It can be simplified further in the form

$$X(k) \;=\; \sum_{n=0}^{N/2-1} x(n)\, exp(-j2\pi nk/N) \;+\; \sum_{n=0}^{N/2-1} x(n+N/2)\, exp(-j2\pi nk/N)\, exp(-j\pi k)$$

$$= \sum_{n=0}^{N/2-1} x(n)\, exp(j2\pi nk/N) \;+\; (-1)^k \sum_{n=0}^{N/2-1} x(n+N/2)\, exp(j2\pi nk/N)$$

The result $X(k)$ is an N-point spectral sequence, for $0 \le k \le N\text{-}1$. When k is even, $k = 2\hat{k}$, the spectral sequence is the $N/2$-point *DFT* of the sum of the first half and second half of the sequence $x(n)$.

$$X(2\hat{k}) \;=\; \sum_{n=0}^{N/2-1} x(n)\, exp(j2\pi n2\hat{k}/N) \;+\; \sum_{n=0}^{N/2-1} x(n+N/2)\, exp(j2\pi n2\hat{k}/N)$$

$$= \sum_{n=0}^{N/2-1} [\, x(n) + x(n+N/2)]\, exp(j2\pi n\hat{k}/(N/2))$$

$$= DFT_{(N/2)}\; \{\, x(n) + x(n+N/2)\}$$

When k is even, $k = 2\hat{k} + 1$, the spectral sequence is the $N/2$-point *DFT* of the difference of the first half and second half of the sequence $x(n)$, multiplied by a phase term $exp(j2\pi n/N)$.

$$
\begin{aligned}
X(2\hat{k}+1) &= \sum_{n=0}^{N/2-1} x(n)\, exp(j2\pi n(2\hat{k}+1)/N) \;-\; \sum_{n=0}^{N/2-1} x(n+N/2)\, exp(j2\pi n(2\hat{k}+1)/N) \\
&= \sum_{n=0}^{N/2-1} [\, x(n) - x(n+N/2) \,]\, exp(j2\pi n(2\hat{k}+1)/N) \\
&= \sum_{n=0}^{N/2-1} [\, x(n) - x(n+N/2)]\;\; exp(j2\pi n/N)\;\; exp(j2\pi n\hat{k}/(N/2)) \\
&= DFT_{(N/2)}\{\, [\, x(n) - x(n+N/2)]\, exp(j2\pi n/N) \,\}
\end{aligned}
$$

This decimation technique also converts an *N*-point *DFT* operation into two *N/2*-point DFT operations. In return, it adds *N* multiplications by the term *exp(j2πn/N)*.

3. Computation complexity

For simplicity, we utilize an 8-point *DFT* to illustrate the decimation procedures for the implementation of *FFT.* Figure (1) shows the time-decimation version, converting it to two 4-point *DFT* operations, with 8 additional multiplications.

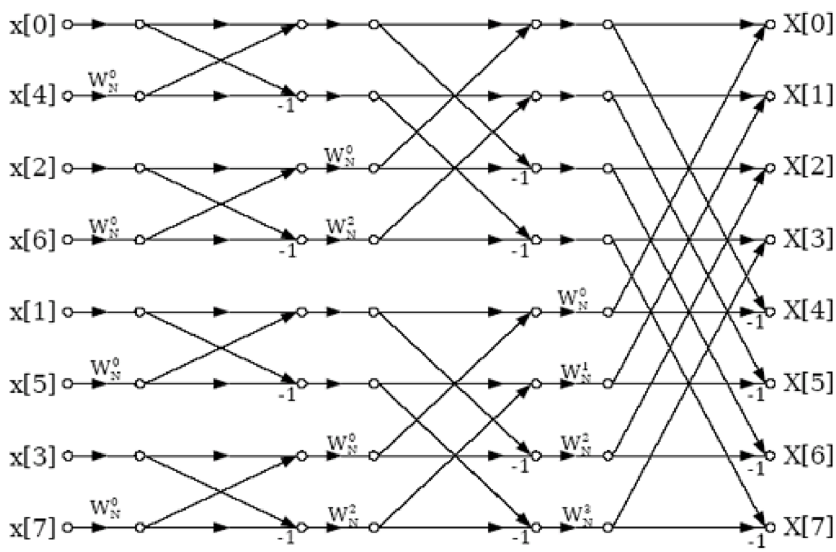

Figure (1): the time-decimation version

As it can be seen, the weighting elements of a 2-point *DFT* are either *(+1)* or *(-1).* This means the 2-point *DFT* does not require complex multiplications any longer. Thus, if *N* is a power of 2 and we continue the decimation procedure all the way to the level of 2-point DFT's, the complex multiplications associated with the DFT procedures go away. The remaining component is the *N* multiplications involved in each decimation step.

Figure (2) shows the frequency-decimation version, which is the reversed configuration of the time-decimation format, mathematically as well as structurally, with the same level of computational simplification.

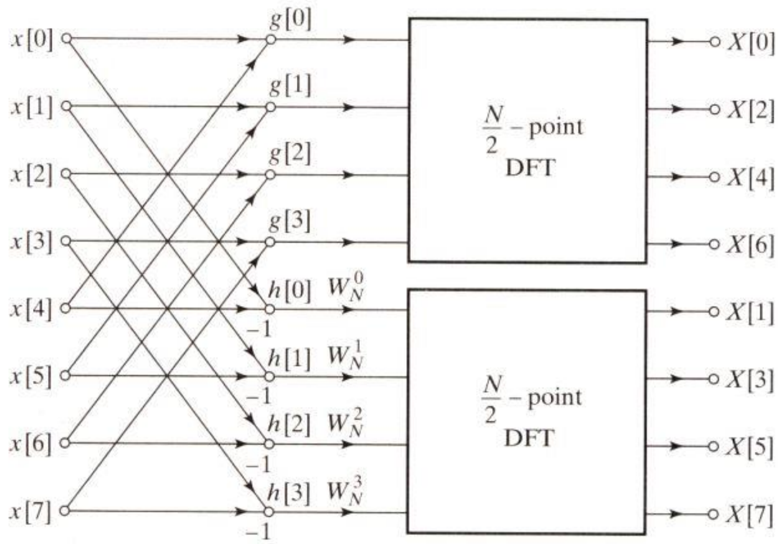

Figure (2) shows the frequency-decimation version

decimations	*DFT size*	***Multiplications for DFT's***	***Total multiplication***
0	*N*	N^2	N^2
1	*N/2*	$2(N/2)^2 = N^2/2$	$N + N^2/2$
2	*N/4*	$4(N/4)^2 = N^2/4$	$2N + N^2/4$
3	*N/8*	$8(N/8)^2 = N^2/8$	$3N + N^2/8$
4	*N/16*	$16(N/16)^2 = N^2/16$	$4N + N^2/16$
logN	*2*	*0*	*(logN) N*

4. Discrete Cosine Transform (DCT)

The DCT is a similar spectrum estimation procedure, designed for real sequences. The input, output, as well as the weighting coefficients are real. It is widely used in image processing, especially for image compression.

The definition of *DCT* is in the form of

$$DCT\{\, x(n) \,\} = \sum_{n=0}^{N-1} x(n) \cos(\pi(2n+1)k/2N) \qquad \text{for} \quad 0 \le k \le N\text{-}1$$

where $x(n)$ is a real N-point input sequence,

$$x(n) = \{\, x(0), x(1), x(2), \ldots.\ x(N\text{-}1) \,\}$$

As it can be seen, the result of *DCT* is also a real *N*-point sequence.

DCT is related to *DFT* in an interesting manner. To illustrate the relationship, we first extend it into a *2N*-point sequence,

$$\hat{x}(n) = \{\, x(0), x(1), x(2), \ldots.\ x(N\text{-}1), x(N\text{-}1), \ldots.\ x(2), x(1), x(0) \,\}$$

The second half of the sequence is symmetric to the first half. Then we take a *2N*-point *DFT* of the extended sequence,

$$\begin{aligned}\hat{X}(k) &= DFT_{(2N)}\{\, \hat{x}(n) \,\} = \sum_{n=0}^{2N-1} \hat{x}(n) \exp(-j2\pi nk/2N) \\ &= \sum_{n=0}^{N-1} \hat{x}(n) \exp(-j2\pi nk/2N) + \sum_{n=N}^{2N-1} \hat{x}(n) \exp(-j2\pi nk/2N) \\ &= \sum_{n=0}^{N-1} x(n) \exp(-j2\pi nk/2N) + \sum_{n=N}^{2N-1} x(2N\text{-}1\text{-}n) \exp(-j2\pi nk/2N)\end{aligned}$$

For the second term, we rearrange the index system in the form of

$$m = 2N - 1 - n$$

such that the two terms can be combined,

$$\hat{X}(k) = \sum_{n=0}^{N-1} x(n)\, exp(-j2\pi nk/2N) + \sum_{m=0}^{N-1} x(m)\, exp(-j2\pi(2N-1-m)k/2N)$$

$$= \sum_{n=0}^{N-1} x(n)\, [\, exp(-j2\pi nk/2N) + exp(-j2\pi(2N-1-n)k/2N)]$$

The kernel of the summation can be simplified further,

$$exp(-j2\pi nk/2N) + exp(-j2\pi(2N-1-n)k/2N)$$

$$= exp(-j2\pi nk/2N) + exp(j2\pi nk/2N)\, exp(-j2\pi(2N-1)k/2N)$$

$$= exp(-j2\pi nk/2N) + exp(j2\pi nk/2N)\, exp(j\pi k/N)$$

$$= exp(j\pi k/2N)\, [\, exp(j\pi k/2N)\, exp(j2\pi nk/2N) + exp(-j\pi k/2N)\, exp(-j2\pi nk/2N)\,]$$

$$= exp(j\pi k/2N)\, [\, exp(j\pi(2n+1)k/2N) + exp(-j\pi(2n+1)k/2N)\,]$$

$$= exp(j\pi k/2N)\; 2\cos(\pi(2n+1)k/2N)$$

Then the *DFT* is thus in the form

$$\hat{X}(k) = \sum_{n=0}^{N-1} x(n)\, exp(j\pi k/2N) \cdot 2\cos(\pi(2n+1)k/2N)$$

$$= 2\, exp(j\pi k/2N) \sum_{n=0}^{N-1} x(n)\cos(\pi(2n+1)k/2N)$$

Hence, the relationship between *DCT* and *DFT* can be written as

$$DCT\{\, x(n) \,\} = \sum_{n=0}^{N-1} x(n) \cos(\pi(2n+1)k/2N)$$

$$= \tfrac{1}{2} \exp(-j\pi k/2N)\, \hat{X}(k)$$

This relationship also suggests that *DCT* can be implemented through the use of *FFT* for computation efficiency.