Procesamiento de Señales e Imágenes Digitales: Cálculo Eficiente de la Transformada Discreta de Fourier

MSc. Renán Rojas G.

Pontificia Universidad Católica del Perú

Motivación

- Herramienta esencial para el análisis y diseño de sistemas en el dominio de la frecuencia.
- La capacidad de calcular la **DFT** con el menor costo computacional posible es crítica en muchas disciplinas.
- De manera colectiva, los algoritmos eficientes para el cálculo de la DFT se denominan **Transformada Rápida de Fourier (FFT)** .
- El costo computacional se mide según el número de multiplicaciones y sumas aritméticas requeridas.

Evaluación del cálculo directo de la DFT

1. DFT directa:

$$X(k) = \sum_{n=0}^{N-1} x[n] W_N^{kn};$$

$$W_N = e^{-j(\frac{2\pi}{N})};$$

2. DFT inversa:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn};$$

El cálculo directo de la **DFT** directa requiere de **N** multiplicaciones complejas y **(N-1)** sumas complejas para cada valor. Para todos los valores se requiere un total de N^2 multiplicaciones y $N(N-1)$ sumas complejas.

*Cada multiplicación compleja requiere 4 multiplicaciones reales y cada suma compleja requiere 2 sumas reales. Entonces, hallar $X(k)$ requiere $4N^2$ multiplicaciones reales y $4(4N-2)$ sumas reales (aproximadamente N^2 para N grande.)

Cálculo eficiente: Método divide y vencerás

- Descomposición del cálculo de una DFT de **N** puntos en múltiples **DFT** de menor tamaño
- Considerando una DFT de **N** puntos y descomponiendo en enteros: $N = L \cdot M$

  En caso $x[n]$ no sea de longitud compuesta (i.e. N es primo), es necesario concatenar ceros a la secuencia (zero padding).
- Es posible ordenar los **N** puntos de la secuencia $x[n]$ en una matriz bidimensional de L filas y M columnas. Lo mismo se puede hacer con los valores calculados de la **DFT**.

  El ordenamiento de los puntos tanto de $x[n]$ como de $X(k)$ puede ser **row-major** (ordenado por filas) o **col-major** (ordenado por columnas).

Cálculo eficiente: Método divide y vencerás

Column-wise $n = l + mL$

$l \backslash m$	0	1	2		$M-1$
0	$x(0)$	$x(L)$	$x(2L)$	$\cdots$	$x((M-1)L)$
1	$x(1)$	$x(L+1)$	$x(2L+1)$	$\cdots$	$x((M-1)L+1)$
2	$x(2)$	$x(L+2)$	$x(2L+2)$	$\cdots$	$x((M-1)L+2)$
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$L-1$	$x(L-1)$	$x(2L-1)$	$x(3L-1)$	$\cdots$	$x(LM-1)$

(a) col-major

Row-wise $n = Ml + m$

$l \backslash m$	0	1	2		$M-1$
0	$x(0)$	$x(1)$	$x(2)$	$\cdots$	$x(M-1)$
1	$x(M)$	$x(M+1)$	$x(M+2)$	$\cdots$	$x(2M-1)$
2	$x(2M)$	$x(2M+1)$	$x(2M+2)$	$\cdots$	$x(3M-1)$
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$L-1$	$x((L-1)M)$	$x((L-1)M+1)$	$x((L-1)M+2)$	$\cdots$	$x(LM-1)$

(b) row-major

Figura : Ordenamiento bidimensional

Cálculo eficiente: Método divide y vencerás

- Asumiendo ordenamiento col-major para $x[n]$ en la matriz $x(l,m)$: $n = l + m \cdot L$
- Asumiendo ordenamiento row-major para $X(k)$ en la matriz $X(p,q)$: $k = M \cdot p + q$

  Entonces, la **DFT** directa ahora está dada por:

$$X(p,q) = \sum_{m=0}^{M-1} \sum_{l=0}^{N-1} x(l,m) W_N^{(Mp+q)(mL+l)}$$

- Analizando W_N: $W_N^{(Mp+q)(Lm+l)} = W_N^{(Mp)(Lm)} \cdot W_N^{Mpl} \cdot W_N^{qLm} \cdot W_N^{lq}$

  $W_N^{(Mp)(Lm)} = W_N^{Npm} = 1;$

  $W_N^{Mpl} = W_{N/M}^{pl} = W_L^{pl};$

  $W_N^{qLm} = W_{N/L}^{mq} = W_M^{mq}.$

Cálculo eficiente: Método divide y vencerás

- Simplificando:

$$X(p,q) = \sum_{l=0}^{L-1} \left\{ W_N^{lq} \left[\sum_{m=0}^{M-1} x(l,m) W_M^{mq} \right] \right\} W_L^{pl}.$$

- Entonces, es posible subdividir el problema en tres pasos:
  1. Cálculo de la **DFT** de M puntos:

$$F(l,q) \triangleq \sum_{m=0}^{M-1} x(l,m) W_M^{mq}; \quad q \in [0; M-1]; \quad l \in [0; L-1].$$

  2. Cálculo del argumento de la sumatoria exterior:

$$G(l,q) = W_N^{lq} F(l,q); \quad q \in [0; M-1]; \quad l \in [0; L-1]$$

  3. Cálculo de la **DFT** de L puntos:

$$X(p,q) = \sum_{l=0}^{L-1} G(l,q) W_L^{pl}$$

Cálculo eficiente: Método divide y vencerás

- El procedimiento hace que el costo computacional sea menor respecto al cálculo directo!
  1. **L** transformadas **DFT** de **M** puntos: LM^2 multiplicaciones compuestas y $LM(M-1)$ sumas complejas.
  2. LM multiplicaciones complejas
  3. ML^2 multiplicaciones complejas y $ML(L-1)$ sumas complejas.

  Total: $N(M+L+1)$ multiplicaciones complejas y $N(M+L-2)$ sumas complejas

Ej: $N = 1000$ separado en $L = 2; M = 500$:

DFT directa: 10^6 multiplicaciones complejas.

Divide y vencerás: 503000 multiplicaciones complejas.

- Ej: Cálculo de la DFT de una secuencia de 15 puntos $(L = 5;\ M = 3)$

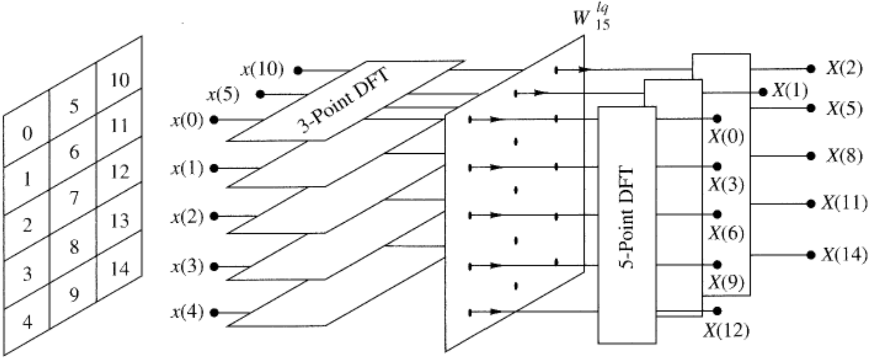

Figura : Algoritmo Divide y Vencerás

Algoritmos FFT base 2 (Radix-2)

- Considerando una señal $x[n]$ de $N = 2^v$ elementos.

  En caso no sea múltiplo de 2, siempre es posible incluir ceros hasta cumplir con la condición.

- Siguiendo la estrategia de **divide y vencerás**: $M = \frac{N}{2}; \quad L = 2$. Es decir, se divide $x[n]$ en dos secuencias:

$$f_1[n] = x[2n]; \quad f_2[n] = x[2n+1];$$

$$n \in [0; \frac{N}{2} - 1]$$

  Dado que $f_1[n]$ y $f_2[n]$ son determinadas a partir de diezmado, el método se conoce como **algoritmo de diezmado temporal (decimation in time)**.

Algoritmos FFT base 2 (Radix-2)

Procedimiento:

1.
$$X(k) = \sum_{\text{n par}} x[n]W_N^{kn} + \sum_{\text{n impar}} x[n]W_N^{kn}$$

$$X(k) = \sum_{m=0}^{\frac{N}{2}-1} x[2m]W_N^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x[2m+1]W_N^{k(2m+1)}$$

2. $W_N^2 = W_{\frac{N}{2}}$

$$\therefore \quad X(k) = \sum_{m=0}^{\frac{N}{2}-1} f_1[m]W_{\frac{N}{2}}^{km} + W_N^k \sum_{m=0}^{\frac{N}{2}-1} f_2[m]W_{\frac{N}{2}}^{km}$$

$$X(k) = F_1(k) + W_N^k F_2(k); \quad k \in [0; N-1].$$

Donde $F_1(k)$ y $F_2(k)$ son las **DFT** de $f_1[m]$ y $f_2[m]$, respectivamente.

Algoritmos FFT base 2 (Radix-2)

3 Dado que $F_1(k), F_2(k)$ son periódicas de periodo $\frac{N}{2}$ y $W_N^{k+\frac{N}{2}} = -W_N^k$:

$$F_1(k + \frac{N}{2}) = F_1(k); \quad F_2(k + \frac{N}{2}) = F_2(k);$$

Entonces podemos determinar $X(k)$ para dos rangos de k:

$$X(k) = F_1(k) + W_N^k F_2(k) \tag{1}$$

$$\begin{aligned} X(k + \frac{N}{2}) &= F_1(k + \frac{N}{2}) + W_N^{(k+\frac{N}{2})} F_2(k + \frac{N}{2}) \\ &= F_1(k) + (-W_N^k) F_2(k). \end{aligned} \tag{2}$$

(1) cubre k en $[0; \frac{N}{2} - 1]$. (2) cubre k en $[\frac{N}{2}; N - 1]$.

Algoritmos FFT base 2 (Radix-2)

- Cambio de notación:

$$X(k) = G_1(k) + G_2(k);$$

$$X(k + \frac{N}{2}) = G_1(k) - G_2(k); \quad k \in [0; \frac{N}{2} - 1];$$

$$G_1(k) = F_1(k);$$

$$G_2(k) = W_N^k F_2(k).$$

- Este cálculo requiere de $\frac{N^2}{2} + \frac{N}{2}$ multiplicaciones complejas. Entonces, para valores altos de N requiere aproximadamente la mitad de multiplicaciones complejas respecto al cálculo directo de la **DFT**!

Algoritmos FFT base 2 (Radix-2)

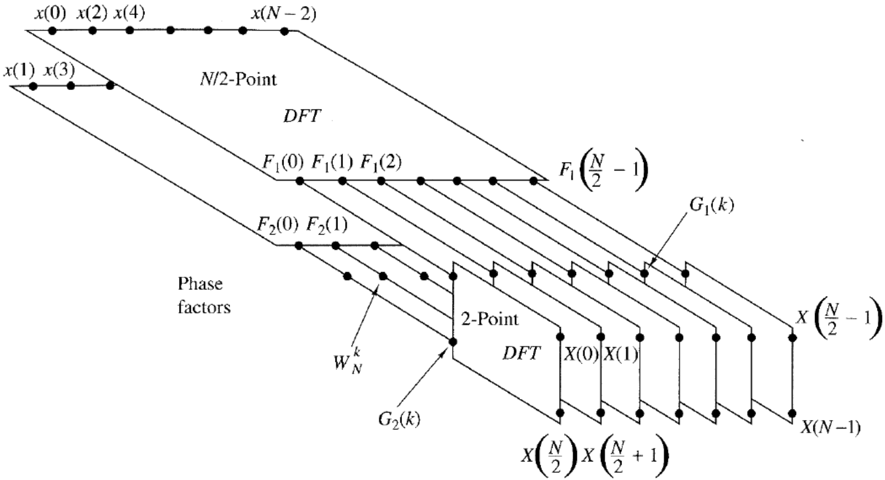

Figura : Cálculo de la DFT a partir de algoritmos base 2

Algoritmos FFT base 2 (Radix-2)

- A partir de diezmado, tal como se realizó en el procedimiento anterior, es posible generar el cálculo de la DFT a partir de múltiples subproblemas de menor tamaño. Entonces, de manera general, es posible diezmar iterativamente hasta obtener **secuencias de un solo elemento**.

  1. Ej: del procedimiento anterior, diezmar cada secuencia y obtener cuatro nuevos subproblemas:

$$v_{11}[n] = f_1[2n]; \quad v_{12}[n] = f_1[2n+1]; \quad n \in [0; \frac{N}{4} - 1];$$

$$v_{21}[n] = f_2[2n]; \quad v_{22}[n] = f_2[2n+1]; \quad n \in [0; \frac{N}{4} - 1];$$

2. Calcular **DFT** de $\frac{N}{4}$ puntos:

$$F_1(k) = V_{11}(k) + W_{\frac{N}{2}}^{k} V_{12}(k);$$

$$F_1(k+\frac{N}{4}) = V_{11}(k) - W_{\frac{N}{2}}^{k} V_{12}(k);$$

$$F_2(k) = V_{21}(k) + W_{\frac{N}{2}}^{k} V_{22}(k);$$

$$F_2(k+\frac{N}{4}) = V_{21}(k) - W_{\frac{N}{2}}^{k} V_{22}(k);$$

* $V_{ij}(k)$: DFT de la cada secuencia $v_{ij}[n]$.
* De la misma forma, el número de multiplicaciones complejas se reduce de $\frac{N}{2}$ a $\frac{N}{4}$.
*** Es posible seguir diezmando y el costo computacional se seguira reduciendo hasta el límite de 1 elemento por secuencia.**

Algoritmos FFT base 2 (Radix-2)

- Ej: Cálculo de la **DFT** de $N = 8$ puntos:

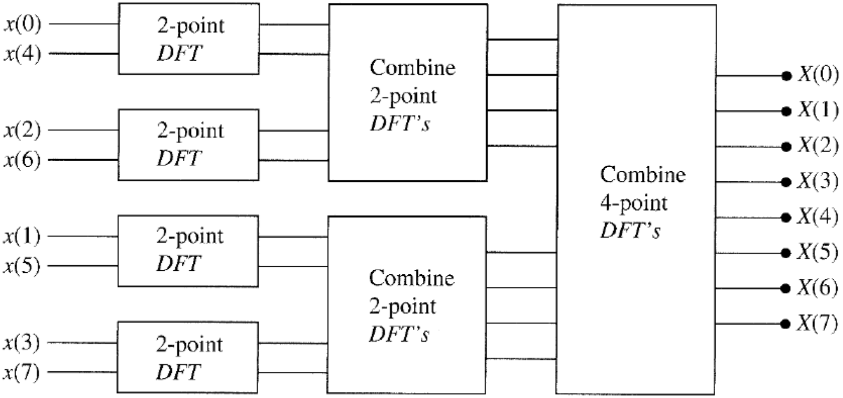

Figura : Cálculo de la DFT de 8 elementos a partir de algoritmos base 2

Algoritmos FFT base 2 (Radix-2)

- El bloque básico en cada etapa del algoritmo es tomar dos números complejos (a, b), multiplicar b por W_N^r, y luego sumarlos y restarlos de a. Esto da como resultado dos números complejos A, B. El bloque básico se denomina **Mariposa (twiddle)**.

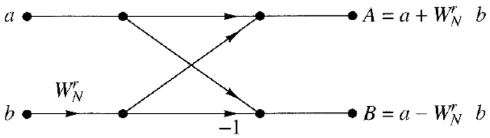

Figura : Bloque básico en el diezmado temporal: Mariposa (Twiddle)

Algoritmos FFT base 2 (Radix-2)

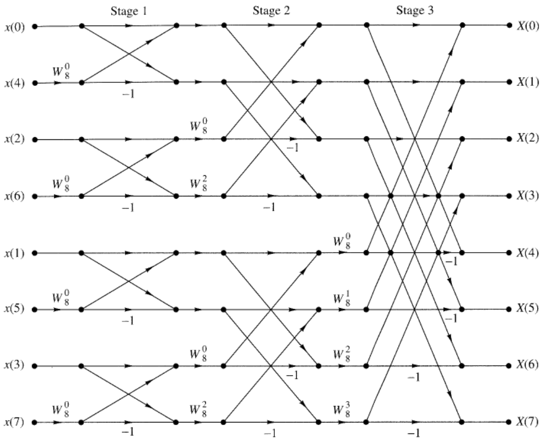

Figura : Diezmado temporal descrito a partir de bloques elementales (twiddles)

Algoritmos FFT base 2 (Radix-2)

- Dado $N = 2^v$, el diezmado puede realizarse $v = \log_2 N$ veces. En este caso, las multiplicaciones complejas se reducen a $\frac{N}{2}\log_2 N$ y las sumas complejas a $N \cdot \log_2 N$

TABLE 8.1 Comparison of Computational Complexity for the Direct Computation of the DFT Versus the FFT Algorithm

Number of Points, N	Complex Multiplications in Direct Computation, N^2	Complex Multiplications in FFT Algorithm, $(N/2)\log_2 N$	Speed Improvement Factor
4	16	4	4.0
8	64	12	5.3
16	256	32	8.0
32	1,024	80	12.8
64	4,096	192	21.3
128	16,384	448	36.6
256	65,536	1,024	64.0
512	262,144	2,304	113.8
1,024	1,048,576	5,120	204.8

Figura : Costo computacional de la DFT vs. la FFT a partir de algoritmos base 2

Referencias

(1) Proakis, J. G. & Manolakis, D. K. (2006), Digital Signal Processing (4th Edition), Prentice Hall.

(2) Oppenheim, A.V., Schafer, R. W., & Buck, J.R. (2009), Discrete-Time Signal Processing (3rd Edition), Prentice-Hall.