Capítulo 6

TRANSFORMADA RAPIDA DE FOURIER (FFT)

Los temas a tratar en el presente capítulo son:

La implementación de la ec. (4.15) involucra un número de sumas y multiplicaciones complejas que es proporcional a N^2.

Lo anterior se puede apreciar fácilmente ya que para cada uno de los N valores de μ, la expansión de la sumatoria requiere N multiplicaciones complejas de f(x) por $e^{-j2\pi\mu x/N}$ y (N-1) sumas de resultados.

El término $e^{-j2\pi\mu x/N}$ puede ser calculado de una vez y almacenado en una tabla para las aplicaciones subsecuentes; por tal razón, la multiplicación de μ por x en este término no se contabiliza normalmente como parte de la implementación.

Se demuestra en lo que sigue que la descomposición de la ec. (4.15) (se reitera para mayor comodidad):

$$F(\mu)=\frac{1}{N}\sum_{x=0}^{N-1} f(x)\cdot e^{-j2\pi\mu x/N} \qquad ((4.15))$$

permite reducir el número de sumas y multiplicaciones a un valor proporcional a $N\log_2 N$.

El procedimiento de descomposición se denomina Algoritmo de Transformada Rápida de Fourier (FFT).

El ahorro o reducción en el número de operaciones es significativo para valores de N como los que es doble esperar en imágenes prácticas, por ejemplo, para una imagen de 1024 x 1024 pixels, N = 1024, se tendría:

N^2 = 1.048.576 operaciones complejas,

Con FFT

$N^2\log_2 N$ = 10.240 operaciones complejas

Con una reducción de 102.4:1, el tiempo de cómputo, empleando máquinas equivalentes, se reduce a menos del 1%.

La descripción que sigue se refiere al desarrollo de un algoritmo de FFT para una variable. Tal como se señala en la ecuación (4.5.1) (“Separabilidad”), una T.F. de dos variables puede ser calculada por aplicación sucesiva de la T.F. de una variable.

6.1 Algoritmo para FFT.

El algoritmo que se plantea está basado en el método denominado “doblamiento sucesivo”. Para simplificar las expresiones, la ec. (4.15) se reescribe:

$$F(\mu)=\frac{1}{N}\sum_{x=0}^{N-1} f(x) W_N^{\mu x} \qquad (6.1)$$

donde

$$W_n = e^{-j2\pi/N} \qquad (6.2)$$

y N se supone de la forma :

$$N = 2^n \qquad (6.3)$$

con N entero positivo. Entonces puede expresarse:

$$N = 2M \tag{6.4}$$

con M también entero positivo. Sustituyendo (6.4) en (6.1) se tiene:

$$\begin{aligned} F(\mu) &= \frac{1}{2M}\sum_{x=0}^{2M-1} f(x)W_{2M}^{\mu x} \\ &= \frac{1}{2}\left\{\frac{1}{M}\sum_{x=0}^{M-1} f(x)W_{2M}^{\mu(2x)} + \frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)W_{2M}^{\mu(2x+1)}\right\} \end{aligned} \tag{6.5}$$

Puesto que, de ec(6.2), $W_{2M}^{2\mu x} = W_{M}^{\mu x}$, la ec. (6.5) puede expresarse:

$$F(\mu) = \frac{1}{2}\left\{\frac{1}{M}\sum_{x=0}^{M-1} f(2x)W_{M}^{\mu x} + \frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)W_{M}^{\mu x}W_{2M}^{\mu}\right\} \tag{6.6}$$

Si se define:

$$F_{par}(\mu) = \frac{1}{M}\sum_{x=0}^{M-1} f(2x)W_{M}^{\mu x} \tag{6.7}$$

para μ = 0, 1, …, M-1 y:

$$F_{impar}(\mu) = \frac{1}{M}\sum_{x=0}^{M-1} f(2x+1)W_{M}^{\mu x} \tag{6.8}$$

para μ = 0, 1, …, M-1, entonces la ec. (6.6) se hace:

$$F(\mu) = \frac{1}{2}\left\{F_{par}(\mu) + F_{impar}(\mu)W_{2M}^{\mu}\right\} \tag{6.9}$$

También, dado que $W_{M}^{\mu+M} = W_{M}^{\mu}$ y $W_{2M}^{\mu+M} = -W_{2M}^{\mu}$,

$$F(\mu + M) = \frac{1}{2}\left\{F_{par}(\mu) - F_{impar}(\mu)W_{2M}^{\mu}\right\} \tag{6.10}$$

Un análisis cuidadoso de las ecuaciones (6.7) a (6.10) muestra algunas propiedades interesantes de dichas expresiones. Nótese que una transformada de N-puntos puede ser calculada dividiendo la expresión original en dos partes, como se indica en las ecs (6.9) y (6.10).

El cálculo de la primera mitad de $F(\mu)$ requiere de la evaluación de las dos transformadas de N/2 puntos según las ecs. (6.7) y (6.8). Los valores resultantes de $F_{impar}(\mu)$ y $F_{par}(\mu)$ se sustituyen en la ecuación (6.9) para obtener $F(\mu)$ para μ = 0, 1, 2, …, (N/2-1). La otra mitad se obtiene mediante la ecuación (6.10) sin requerir evaluaciones adicionales de la transformada.

Considerando un número de muestras igual a 2^n, con n entero positivo, se puede demostrar que el número de operaciones complejas (multiplicaciones y sumas) está dado por:

$$m(n) = 2m(n.-1) + 2^{n-1} \quad n \geq 1 \tag{6.11}$$

$$a(n) = 2a(n.-1) + 2^n \quad n \geq 1 \tag{6.12}$$

expresiones recursivas que indican el número de multiplicaciones (ec. 6.11) y de sumas (ec. 6.12) . para las que m(0) y a(0) son iguales a cero, puesto que la transformada de un punto no requiere operación alguna.

<u>Número de operaciones.</u>

Es posible concluir, por inducción, que el número de operaciones, sumas y multiplicaciones complejas, que se requiere para implementar un algoritmo para FFT como el recién descrito está dado por:

$$\begin{aligned} m(n) &= \frac{1}{2} 2^n \log_2 2^n \\ &= \frac{1}{2} N \log_2 N \\ &= \frac{1}{2} Nn \qquad n \geq 1 \end{aligned} \tag{6.12}$$

y

$$\begin{aligned} a(n) &= 2^n \log_2 2^n \\ &= N \log_2 N \\ &= Nn \qquad n \geq 1 \end{aligned} \tag{6.13}$$

para multiplicaciones (6.12) y sumas (6.13).

6.2 <u>La FFT inversa.</u>

Resulta que todo algoritmo que se implemente para calcular la FFT discreta con modificaciones simples en sus entradas, puede ser utilizado para el cálculo de la inversa.

La ecuación de la directa:

$$F(\mu) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cdot e^{-j2\pi\mu x/N} \tag{6.14}$$

y

$$f(x) = \sum_{\mu=0}^{N-1} F(\mu) \cdot e^{j2\pi\mu x/N} \tag{6.15}$$

para la inversa, permiten el siguiente procedimiento:

-Tomando la ecuación (6.15) en su conjugada y dividiendo ambos lados por N, resulta:

$$\frac{1}{N} f^*(x) = \frac{1}{N} \sum_{x=0}^{N-1} F^*(\mu) \cdot e^{-j2\pi\mu x/N} \tag{6.17}$$

al comparar se aprecia que el lado derecho tiene la misma forma que la ecuación (6.14). Entonces, usando $F^*(\mu)$ como entrada para el algoritmo empleado en el cálculo de la FFT directa, el resultado que se obtiene es $f^*(x)/N$.

Al resultado obtenido se le conjuga (se obtiene su complejo conjugado) y se multiplica por N, resultando la inversa deseada $f(x)$.

Para el caso bidimensional corresponde obtener el complejo conjugado de la ec. (4.23), resultando:

$$f^*(x,y) = \frac{1}{N} \sum_{\mu=0}^{N-1} \sum_{v=0}^{N-1} F^*(\mu, v) \cdot e^{-j2\pi(\mu x + vy)/N} \qquad (6.18)$$

que tiene la misma forma que la FFT directa para dos dimensiones de la ec. (4.22).

Entonces, aplicando $F^*(\mu, v)$ a un algoritmo desarrollado para el cálculo de la transformada directa, el resultado obtenido será $f^*(x,y)$; tomando el complejo conjugado de este resultado se obtendrá $f(x,y)$.

Naturalmente, si $f(x)$ o $f(x,y)$ son reales, la operación de complejo conjugado es innecesaria.

6.3 Implementación.

El algoritmo planteado en 6.1 es directo; la cuestión relevante es que los datos de entrada deben ser reordenados para la aplicación sucesiva de las ecs. (6.7) y (6.8).

Un ejemplo simple: cálculo de la FFT, por el algoritmo de doblamiento sucesivo, de una función de 8 puntos $\{f(0), f(1), ..., f(7)\}$.

La ecuación (6.7) usa los argumentos de tipo par: $\{f(0), f(2), f(4), f(6)\}$. La ec. (6.8) los de tipo impar: $\{f(1), f(3), f(5), f(7)\}$.

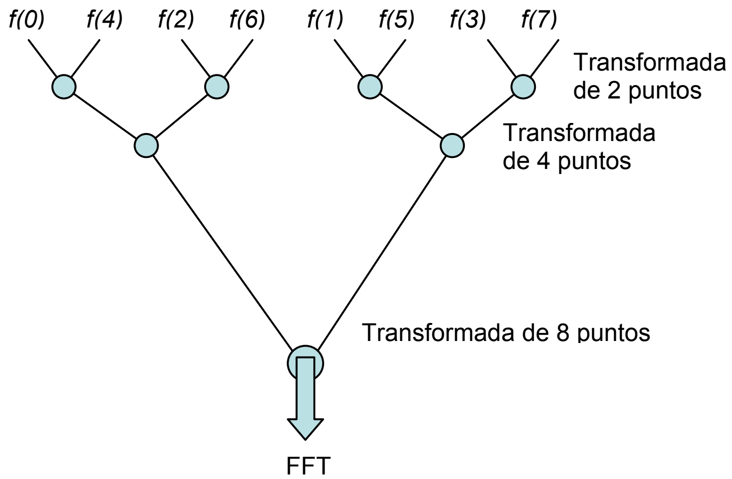
Cada transformada de 4 puntos se calcula como 2 transformadas de 2 puntos; esto utiliza también las ecs. (6.7) y (6.8), con su nuevo carácter de par/impar para los argumentos de cada grupo.

Así, para el primer conjunto la subdivisión genera $\{f(0), f(4)\}$ como parte par y $\{f(2), f(6)\}$ como parte impar. Igualmente $\{f(1), f(5)\}$ para la parte par y $\{f(3), f(7)\}$ para parte impar.

De acuerdo a lo anterior, el ordenamiento requerido para aplicar directamente el algoritmo es:

$$\{f(0), f(4), f(2), f(6), f(1), f(5), f(3), f(7)\}$$

El dibujo siguiente ilustra la forma en que opera el algoritmo:



El primer nivel de cálculo incluye 4 transformadas de 2 puntos. Estos 4 resultados se utilizan para el segundo nivel formando dos transformadas de 4 puntos cuyos resultados llegan al último nivel donde el cálculo produce la transformada deseada.

Al observar el reordenamiento generado para el cálculo, se aprecia que (felizmente) éste sigue una regla simple: reversión de los bits. Para el ejemplo de 8 puntos, son tres los bits que identifican cada elemento: 000, 001,..., 111.

Orden Inicial	Argumento	Orden Modificado	Argumento
f(0)	000	f(0)	000
f(1)	001	f(4)	100
f(2)	010	f(2)	010
f(3)	011	f(6)	110
f(4)	100	f(1)	001
f(5)	101	f(5)	101
f(6)	110	f(3)	011
f(7)	111	f(7)	111