



UNIVERSITÀ DEGLI STUDI DI CAGLIARI
FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Elettrica ed Elettronica

**ELABORAZIONE DI SEGNALI
MONODIMENSIONALI E BIDIMENSIONALI
TRAMITE TRASFORMATA WAVELET
DISCRETA**

Relatore:
Prof. Giuseppe Rodriguez

Candidato:
Francesca Marcello

Anno Accademico 2014/2015

Indice

Introduzione	1
1 L'analisi di Fourier	6
1.0.1 Serie di Fourier	6
1.0.2 Trasformata di Fourier	8
1.1 Trasformata di Fourier discreta	9
1.2 Short-time Fourier Transform	12
2 L'analisi wavelet	14
2.1 Trasformata wavelet continua	14
2.2 Trasformata wavelet discreta	17
2.3 Analisi Multirisoluzione	18
2.3.1 Scaling function	18
2.3.2 Wavelet	20
2.4 Trasformata wavelet veloce	24
2.4.1 Implementazione degli algoritmi	26
2.5 Test e confronti	30
3 Elaborazione di immagini	39
3.1 Wavelet in due dimensioni	40
3.1.1 Decomposizione e ricostruzione di immagini	41
3.2 Esperimento	45
4 Conclusioni	50
Bibliografia	51

Introduzione

I segnali sono quantità misurabili che trasportano informazioni riguardo un certo fenomeno fisico che può essere di natura diversa: acustica, ottica, elettrica. Matematicamente possono essere descritti tramite una relazione che lega una variabile indipendente, per esempio il tempo o lo spazio, ad una variabile dipendente, che è il valore misurato della grandezza fisica. Quando la funzione dipende da una sola variabile il segnale viene detto *monodimensionale*; se dipende da due o più variabili viene detto *multidimensionale*. Le immagini possono essere definite come segnali bidimensionali $f(x, y)$ dove x e y sono coordinate spaziali e il valore rappresentato dalla funzione per ciascuna coppia di coordinate (x, y) è l'intensità dell'immagine in quel punto. L'elaborazione di un segnale è l'insieme delle operazioni e degli algoritmi che servono per rappresentare, analizzare e in generale agire in qualche modo su di esso per ottenere informazioni utili.

Generalmente lo strumento utilizzato per l'analisi di segnali, soprattutto di quelli periodici, è la trasformata di Fourier, con cui è possibile scomporre il segnale in funzioni di base sinusoidali a frequenze diverse. Se consideriamo un segnale $f(x)$ essa è definita come:

$$\mathcal{F}\{f\} = F(k) = \hat{f}(k) = \int_{-\infty}^{\infty} e^{-ikx} f(x) dx.$$

Il problema principale della trasformata di Fourier è dato dal fatto che seni e coseni sono perfettamente localizzati in frequenza, ma non nel tempo. Si ottengono quindi informazioni accurate sul contenuto in frequenza del segnale ma si perdono le informazioni sulla localizzazione temporale di una data frequenza.

Un modo per risolvere questo problema consiste nel ricorrere ad una trasformata di Fourier localizzata, ovvero la short-time Fourier transformer (STFT). Utilizzando una funzione finestra che venga traslata lungo l'asse temporale, si può analizzare il segnale piccole sezioni per volta in modo da mantenere alcune informazioni in entrambi i domini. Ovviamente però ci sono dei limiti di precisione sulle informazioni così ottenute e tale precisione dipende dall'ampiezza della finestra considerata. Infatti una volta scelte le sue dimensioni

queste rimangono fissate sia per studiare fenomeni alle basse frequenze che per studiare fenomeni alle alte frequenze. Per molti segnali, in particolare quelli non stazionari, è richiesta una soluzione diversa, tale per esempio da permettere di variare le dimensioni della finestra e determinare una migliore analisi sia in tempo che in frequenza.

Con il termine *wavelets* si indicano degli strumenti matematici adatti per la rappresentazione e segmentazione di segnali, per l'analisi tempo-frequenza e per l'implementazione di algoritmi veloci. Il loro nome deriva dal fatto che si tratta di funzioni oscillanti con valore medio nullo e di durata limitata. Al contrario delle sinusoidi, che si estendono da meno a più infinito, le wavelets hanno un inizio e una fine precisi e al di fuori di tale intervallo sono nulle o decadono rapidamente.

Le principali operazioni che si effettuano su una wavelet $w(t)$ sono due: una traslazione e un ridimensionamento. La traslazione è ottenuta sottraendo dall'argomento della wavelet un certo valore k , passando da $w(t - 1)$ a $w(t - 2)$ e $w(t - k)$; per una funzione o segnale monodimensionale questa traslazione consiste semplicemente in uno spostamento della wavelet lungo l'asse temporale. L'operazione di ridimensionamento è ottenuta dividendo l'argomento per un fattore di scala a . La wavelet $w(t/a)$ è la stessa wavelet iniziale che viene compressa o dilatata a seconda che il valore di a sia minore o maggiore di uno rispettivamente. In molti casi è utile che $a = 1/2^j$ in modo da ottenere diverse wavelets $w(2^j t)$ ridimensionate secondo potenze di 2. La combinazioni di traslazione e dilatazioni produce un'intera famiglia di wavelets $w(2^j t - k)$ che derivano tutte dalla stessa $w(t)$.

La costruzione di tali funzioni avviene tramite un filtro passa-alto H_1 , un filtro passa-basso H_0 e la scelta di un insieme, finito e piccolo, di coefficienti $h_0(k)$ e $h_1(k)$. L'iterazione del filtro passa-basso permette di definire la funzione di scaling $\varphi(t)$:

$$\varphi(t) = \sum h_0(k)\varphi(2t - k). \quad (0.1)$$

Le wavelets derivano da $\varphi(t)$ tramite il filtro passa-alto:

$$w(t) = \sum h_1(k)\varphi(2t - k). \quad (0.2)$$

L'approssimazione del segnale A_j per ogni livello j è data dalla funzione di scaling, mentre i dettagli D_j derivano dalle wavelets. Questo permette di capire come funziona quella che viene chiamata *analisi multirisoluzione*. Il segnale viene infatti analizzato secondo scale differenti e la sua ricostruzione è data dalla somma $D_1 + D_2 + D_3 + D_4 + A_4$.

Possiamo quindi dire che le wavelets sono delle funzioni con 3 importanti proprietà:

- il parametro di scaling permette di cambiare sia la larghezza della banda di frequenze sia la frequenza centrale di tale banda;
- il parametro di traslazione permette di stabilire la posizione nel tempo o nella spazio di tale banda, dato che il filtro ha la capacità di restringersi automaticamente per analizzare i cambiamenti del segnale alle alte frequenze e di allargarsi automaticamente per stabilirne il comportamento alle basse frequenze;
- insieme alla funzione di scaling permette di realizzare algoritmi più efficienti.

Capitolo 1

L'analisi di Fourier

Quando si parla di analisi di Fourier generalmente ci si riferisce alla *trasformata di Fourier* o alla *serie di Fourier*. L'idea fondamentale è quella di rappresentare una funzione complessa tramite delle funzioni più semplici, quali sono seni e coseni. In questo modo si possono studiare diversi casi più difficili basandosi sulle caratteristiche delle funzioni di base e ottenere informazioni non ovvie sulle proprietà della funzione originale.

Se consideriamo un segnale analogico f il cui dominio sia quello temporale, la sua trasformata di Fourier \mathcal{F} ne descrive il comportamento spettrale, e poiché le informazioni dello spettro sono date in termini di frequenze, il dominio di definizione della trasformata è quello della frequenza. Quando il segnale analizzato è periodico, la sua trasformata è un insieme di valori discreti, ovvero si ottiene uno spettro discreto; la frequenza più bassa della sinusoidale che rappresenta il segnale è detta *armonica fondamentale* ed è quella di importanza maggiore per la sua ricostruzione, le altre frequenze sono multiple di quella fondamentale e sono spesso chiamate *armoniche secondarie*. Nel caso dei segnali periodici si parla di serie di Fourier.

1.0.1 Serie di Fourier

Se consideriamo una funzione periodica con periodo $2L$ e definita nell'intervallo $[-L, L]$, la sua serie di Fourier trigonometrica è del tipo

$$a_0 + \sum_{k=1}^{\infty} [a_k \cos k\omega x + b_k \sin k\omega x], \quad \omega = \frac{\pi}{L} \quad (1.1)$$

dove

$$a_0 = \frac{1}{2L} \int_{-L}^L f(x) dx,$$

$$\begin{aligned}
a_k &= \frac{1}{L} \int_{-L}^L f(x) \cos k\omega x \, dx, \\
b_k &= \frac{1}{L} \int_{-L}^L f(x) \sin k\omega x \, dx
\end{aligned} \tag{1.2}$$

sono definiti i coefficienti di Fourier della funzione.

Esiste un altro modo per rappresentare la serie che risulta più adatto in certi casi, soprattutto nell'analisi di segnali digitali, definito forma complessa. Per introdurla bisogna prima ricordare la formula di Eulero

$$e^{j\varphi} = \cos \varphi + j \sin \varphi$$

dalla quale si ricavano

$$\cos \varphi = \frac{e^{j\varphi} + e^{-j\varphi}}{2} \quad \sin \varphi = \frac{e^{j\varphi} - e^{-j\varphi}}{2j}$$

che sostituiti nella serie conducono alla forma

$$\begin{aligned}
& a_0 + \sum_{k=1}^{\infty} \left(a_k \frac{e^{j\varphi} + e^{-j\varphi}}{2} + b_k \frac{e^{j\varphi} - e^{-j\varphi}}{2j} \right) \\
&= a_0 + \sum_{k=1}^{\infty} \left[\frac{1}{2} (a_k - jb_k) e^{jk\omega x} + \frac{1}{2} (a_k + jb_k) e^{-jk\omega x} \right].
\end{aligned} \tag{1.3}$$

Ponendo

$$c_0 = a_0, \quad c_k = \frac{1}{2}(a_k - jb_k), \quad c_{-k} = \frac{1}{2}(a_k + jb_k), \quad k = 1, 2, \dots, \tag{1.4}$$

si possono calcolare i coefficienti e si ottiene la serie

$$\mathcal{S}_f(x) = c_0 + \sum_{k=1}^{\infty} [c_k e^{jk\omega x} + c_{-k} e^{-jk\omega x}] \tag{1.5}$$

che equivale a scrivere, in forma più compatta,

$$\mathcal{S}_f(x) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega x}. \tag{1.6}$$

Se si conosce la serie in forma complessa si può immediatamente passare alla forma trigonometrica dell'equazione 1.3 sempre ricordando la formula di Eulero per cui

$$e^{jk\omega x} = \cos k\omega x + j \sin k\omega x, \quad \text{per } k = \pm 1, \pm 2, \dots,$$

$$e^{jk\omega x} = 1 \quad \text{per } k = 0,$$

mentre per i coefficienti otteniamo dalla 1.4 le relazioni

$$a_0 = c_0, \quad a_k = c_k + c_{-k}, \quad b_k = j(c_k - c_{-k}), \quad k = 1, 2, \dots$$

Infine anche i coefficienti c_k possono essere espressi facilmente a partire dalla funzione f in questo modo

$$c_0 = \frac{1}{2L} \int_{-L}^L f(x) dx \quad \text{e} \quad c_k = \frac{1}{2L} \int_{-L}^L f(x) e^{-jk\omega x} dx. \quad (1.7)$$

Possiamo poi definire lo spettro delle ampiezze di una funzione periodica di periodo $T = 2L$, rappresentata da una serie di Fourier in forma complessa, come il grafico dei moduli $|c_k|$ in funzione delle frequenze $k\omega$, per cui il grafico visualizza i contributi al segnale delle sue armoniche.

1.0.2 Trasformata di Fourier

Per le funzioni periodiche di periodo T lo sviluppo mediante serie di Fourier utilizza solo le frequenze multiple di $\frac{2\pi}{T}$. Per le funzioni non periodiche invece sono ammesse tutte le frequenze, per cui la sommatoria diventa un integrale. I coefficienti a_k e b_k , o nel caso della forma complessa i coefficienti c_k , diventano i valori della trasformata $\mathcal{F}\{f\}$ calcolata in k , la quale è definita sull'intera retta reale.

Per capire come passare da una funzione periodica a una definita su tutto \mathbb{R} dobbiamo considerare la formula per i coefficienti e far tendere all'infinito l'ampiezza del periodo. La funzione deve essere sviluppabile in serie di Fourier su ogni intervallo $[-L, L]$ e f_T è la sua estensione periodica di periodo T su $[-\frac{T}{2}, \frac{T}{2}]$ di cui studiamo i coefficienti per T che tende all'infinito:

$$c_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(x) e^{-jk\omega x} dx \quad \text{con } \omega = \frac{2\pi}{T}.$$

Si può quindi scrivere

$$f_T(x) = \sum_{-\infty}^{\infty} c_k e^{-jk\omega x} = \sum_{-\infty}^{\infty} \left[\frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(x) e^{-jk\frac{2\pi}{T}x} dx \right] e^{jk\frac{2\pi}{T}x} \Delta k$$

dove $\Delta k = 1$ è una variabile che rappresenta la variazione dell'indice k . Per far tendere T all'infinito si deve effettuare un cambio di variabile $s = \frac{2\pi}{T}k$ da

cui si ottiene che $\Delta k = \frac{T}{2\pi} \Delta s$. Questo cambio di variabile conduce a

$$f_T(x) = \frac{1}{2\pi} \sum_{-\infty}^{\infty} \left[\int_{-\frac{T}{2}}^{\frac{T}{2}} f_T(x) e^{-jsx} dx \right] e^{jsx} \Delta s$$

e passando al limite per $T \rightarrow \infty$, il fattore $\Delta s \rightarrow 0$, $f_T(x) \rightarrow f(x)$ e la sommatoria diventa un'integrale rispetto a s

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f_T(x) e^{-jsx} dx \right] e^{jsx} ds.$$

Le variabili di integrazione possono essere indicate con altre lettere, generalmente la variabile per l'integrazione interna è indicata con x o t mentre quella esterna con k o ω . Indicando l'integrale interno alle parentesi quadre con $F(k)$ otteniamo infine

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jkx} F(k) dk.$$

Quindi possiamo definire la trasformata di Fourier come la funzione

$$\mathcal{F}\{f\} = F(k) = \int_{-\infty}^{\infty} e^{-jkx} f(x) dx. \quad (1.8)$$

L'ampiezza dello spettro della funzione f è data dal grafico di $|F(k)|$ che è il modulo della trasformata di Fourier della funzione stessa.

1.1 Trasformata di Fourier discreta

La trasformata discreta di Fourier (DFT) permette l'analisi in frequenza dei segnali a tempo discreto. In generale appare diversa rispetto all'analisi di Fourier continua ma in realtà risultano estremamente legate, ed è essenziale capire quale sia la loro relazione. La DFT è definita come

$$F_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n \omega_N^{-nk} \quad \text{con } k = 0, 1, \dots, N-1 \quad (1.9)$$

dove $\omega_N = e^{\frac{2\pi j}{N}}$ viene chiamata radice N-esima dell'unità in quanto $(\omega_N)^N = e^{2\pi j} = 1$. Partendo dalla serie di Fourier possiamo ricavare l'espressione della DFT come un'approssimazione dell'integrale che permette di calcolare i coefficienti c_k della serie di Fourier; per approssimare l'integrale utilizziamo la *regola del trapezio composta*.

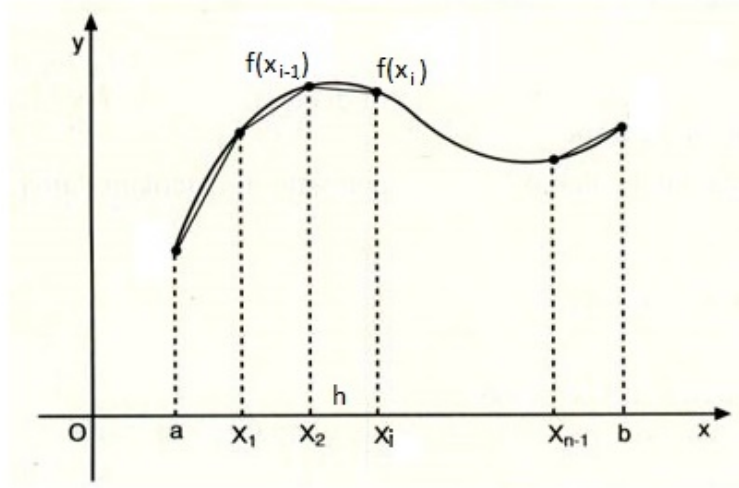


Figura 1.1: Regola del trapezio composta

Consideriamo una funzione nell'intervallo di integrazione $[a, b]$ e dividiamo tale intervallo in $N + 1$ punti che delimitano gli N sottointervalli, tutti di eguale ampiezza.

La distanza tra due punti consecutivi la indichiamo come $h = (x_i - x_{i-1})$ e conosciamo il valore della funzione $f(x_i)$ in ogni punto. L'area dei singoli trapezi sarà allora calcolabile come

$$[f(x_i) + f(x_{i-1})] \frac{h}{2}$$

e l'area totale si ottiene sommando quella di ciascuno trapezio. Il termine $\frac{h}{2}$ è comune a tutti, mentre ciascuno dei valori della funzione, esclusi il primo $f(a)$ e l'ultimo $f(b)$, viene sommato due volte.

I coefficienti della serie di Fourier li calcoliamo dalla formula

$$c_k = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) e^{-j2\pi kx/T} dx$$

equivalente alla 1.7 con $T = 2L$ e $\omega = \frac{2\pi}{T}$. L'integrale è definito nell'intervallo di ampiezza T , che suddividiamo in N sottointervalli ciascuno di ampiezza $h = \Delta x = \frac{T}{N}$. Gli $N + 1$ punti compresi tra $[-\frac{T}{2}, \frac{T}{2}]$ sono definiti da $x_n = n\Delta x$ per $n = [-\frac{N}{2} : \frac{N}{2}]$. Se poniamo $g(x) = f(x)e^{-j2\pi kx/T}$ approssimiamo

l'integrale come

$$\int_{-\frac{T}{2}}^{\frac{T}{2}} g(x) = \frac{\Delta x}{2} \left[g\left(-\frac{T}{2}\right) + 2 \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}-1} g(x_n) + g\left(\frac{T}{2}\right) \right].$$

Poichè la funzione è periodica

$$g\left(-\frac{T}{2}\right) = g\left(\frac{T}{2}\right)$$

e quindi approssimiamo i coefficienti in questo modo

$$\begin{aligned} c_k &= \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} g(x) = \frac{1}{T} \frac{\Delta x}{2} \left[2 \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} g(x_n) \right] \\ &= \frac{1}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{-j2\pi nk/N}. \end{aligned} \quad (1.10)$$

L'ultima espressione coincide con la definizione di trasformata discreta che abbiamo dato all'inizio se sostituiamo $(e^{\frac{2\pi j}{N}})^{-nk} = \omega_N^{-nk}$.

Anche nel caso della trasformata di Fourier continua la DFT sarà una sua approssimazione, ottenuta con un procedimento esattamente uguale e utilizzando sempre la regola del trapezio composta

$$F(\omega) \approx \frac{T}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{-j2\pi nk/N} \quad (1.11)$$

ottenendo che essa è T volte la DFT del campionamento della funzione su punti equispaziati nell'intervallo $[-\frac{T}{2}, \frac{T}{2}]$.

Consideriamo anche il dominio della frequenza $[-\frac{\Omega}{2}, \frac{\Omega}{2}]$, esso sarà diviso in N sottointervalli ognuno dei quali avrà ampiezza $\Delta\omega$. Gli $N + 1$ punti sono definiti come $\omega_k = k\Delta\omega$, dove $k = [-\frac{N}{2} + 1 : \frac{N}{2}]$. Serve un modo per mettere in relazione i parametri $T, \Omega, \Delta x, \Delta\omega$ e a questo scopo servono le *relazioni di reciprocità*.

T corrisponde al massimo periodo possibile per un'onda nell'intervallo $[-\frac{T}{2}, \frac{T}{2}]$, tale onda avrà la più bassa frequenza associata allo stesso intervallo e la indichiamo come l'unità di frequenza fondamentale per cui

$$\Delta\omega = \frac{1}{T},$$

tutte le altre frequenze della trasformata discreta saranno multiple di $\Delta\omega$. L'intervallo nel dominio della frequenza è suddiviso in N parti separate da $\Delta\omega$ per cui segue che $\Omega = N\Delta\omega$. Combinando queste due espressioni otteniamo la prima relazione di reciprocità

$$\Omega = N\Delta\omega = \frac{N}{T} \quad T\Omega = N. \quad (1.12)$$

Si vede che il dominio spaziale, o temporale, e quello della frequenza variano in modo inverso l'uno dall'altro. Fissato N , un incremento della lunghezza del dominio dello spazio comporta una riduzione della lunghezza del dominio della frequenza.

La seconda relazione di reciprocità si ricava considerando $N\Delta x = T$ e $\Delta\omega = \frac{1}{T}$, che combinate insieme danno

$$\frac{1}{\Delta\omega} = T = N\Delta x \quad \Delta x\Delta\omega = \frac{1}{N} \quad (1.13)$$

da cui ugualmente si nota che la variazione dell'ampiezza dei sottointervalli nel dominio dello spazio è inversamente proporzionale a quella del dominio della frequenza. Per cui, sempre per N fissato, dimezzare l'ampiezza Δx porta a dimezzare anche la lunghezza dell'intero dominio spaziale. Se prima la frequenza fondamentale era $\frac{1}{T}$ adesso è diventata $\frac{1}{T/2} = \frac{2}{T}$, ossia $\Delta\omega$ è raddoppiato nel processo

L'importanza della DFT è dovuta non solo al fatto che è necessario, anche per l'elaborazione dei segnali continui, che questi vengano prima campionati e discretizzati, ma anche perchè per calcolarla è possibile implementare un algoritmo ottimizzato, noto come *trasformata di Fourier veloce* (FFT). Grazie alla FFT si abbassa drasticamente la complessità computazionale; infatti per la trasformata discreta sono richieste $O(N^2)$ operazioni aritmetiche, mentre con la trasformata veloce otteniamo lo stesso risultato in $O(N \log(N))$ operazioni.

1.2 Short-time Fourier Transform

La maggior parte dei segnali reali è di tipo non stazionario per i quali abbiamo detto che la trasformata di Fourier non è molto efficiente nell'analisi. Proprio per come è definita si vede chiaramente che essa è perfettamente localizzata in frequenza ma non nel tempo, infatti dalla trasformata possiamo ricavare informazioni su quali siano le frequenze contenute nel segnale ma non su quando e come esse siano effettivamente presenti. Per risolvere il problema si deve inserire nella trasformata una dipendenza dal tempo e a questo scopo si

usa una funzione finestra $w(t-b)$ in cui il parametro b è usato per farla traslare in modo da ricoprire tutto il dominio temporale e ricavare informazioni sul segnale una porzione per volta in base a dove è posizionata la finestra.

Il centro e il raggio della finestra sono dati da

$$x^* = \frac{1}{\|w\|_2^2} \int_{-\infty}^{\infty} t |w(t)|^2 dt \quad (1.14)$$

$$\Delta_w = \frac{1}{\|w\|_2} \left\{ \int_{-\infty}^{\infty} (t - x^*)^2 |w(t)|^2 dt \right\}^{\frac{1}{2}} \quad (1.15)$$

e la sua ampiezza è pari a $2\Delta_w$. La trasformata di Fourier a tempo breve (STFT) è dunque definita come

$$STFT_f = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} \overline{w(t-b)} dt, \quad (1.16)$$

serve per evidenziare lo spettro del segnale $f(t)$ nella finestra

$$[x^* + b - \Delta_w, x^* + b + \Delta_w]$$

e permette di ottenere le informazioni sul suo contenuto armonico intorno all'istante di tempo b .

Il problema principale di questa soluzione è dato dal fatto che l'ampiezza della finestra, una volta stabilita, non può essere modificata e rimane sempre la stessa sia per analizzare fenomeni alle basse frequenze che per analizzare fenomeni alle alte frequenze, i cui comportamenti sono invece molto diversi. Inoltre la moltiplicazione nel dominio del tempo tra la funzione e la finestra equivale a fare un convoluzione tra i loro spettri nel dominio della frequenza, quindi le informazioni spettrali del segnale vengono alterate. Se si volesse ridurre questo effetto si dovrebbe ridurre l'ampiezza della finestra in frequenza il che comporterebbe l'aumento dell'ampiezza della finestra temporale e quindi minore capacità di localizzazione nel tempo, perchè due eventi separati tra loro meno di Δ_w non sono ovviamente distinguibili. Invece ridurre Δ_w aumenta la capacità di localizzazione nel tempo ma porta all'aumento anche dell'ampiezza della finestra in frequenza e dunque peggiora la localizzazione nel dominio della frequenza. Esiste una sorta di principio di indeterminazione per cui con una sola finestra è impossibile avere una buona risoluzione in entrambi i domini nello stesso momento: se $w(t)$ ha un supporto largo la sua trasformata $W(\omega)$ ha supporto stretto e viceversa.

Capitolo 2

L'analisi wavelet

2.1 Trasformata wavelet continua

La trasformata wavelet continua (CWT) permette di ottenere una finestra variabile, più ampia per l'analisi alle basse frequenze e che si restringa per riconoscere i dettagli alle alte frequenze, perchè mappando il parametro di scala con le frequenze si può agire sulla larghezza della finestra. Il secondo parametro è poi usato per far muovere la finestra lungo l'asse delle frequenze. Consideriamo la trasformata integrale

$$\int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt \quad (2.1)$$

dove a è il parametro di scala che determina l'espansione o compressione della wavelet come si vede in figura:

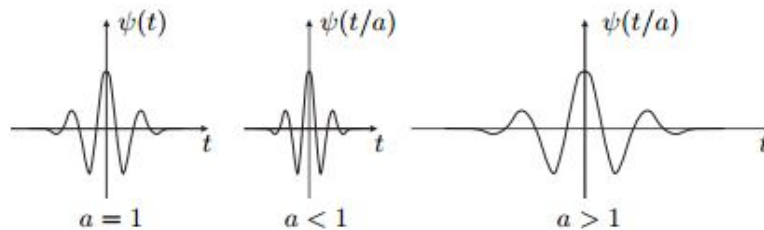


Figura 2.1: Wavelet per diversi valori del parametro a

Perchè il processo sia efficace la funzione $\psi(t)$ deve essere reale ed avere un comportamento passabanda, ovvero tale che la sua trasformata $\Psi(0) = 0$.

Questo equivale a dire che il valore medio della wavelet deve essere nullo dato che

$$\Psi(0) = \int_{-\infty}^{\infty} \psi(t) dt. \quad (2.2)$$

Queste due caratteristiche sono necessarie per la perfetta ricostruzione del segnale a partire dalla sua trasformata. Essendo infatti una funzione reale la sua trasformata di Fourier ha la proprietà che $\Psi(-\omega) = \overline{\Psi(\omega)}$ e quindi $|\Psi(\omega)|$ è una funzione pari. Dato che ci interessano solo i valori positivi delle frequenze ω e che $\psi(t)$ è un filtro passabanda possiamo considerare l'asse delle frequenze nel dominio $[0, \infty)$.

Consideriamo quindi una wavelet $\psi(t)$ tale che sia la funzione stessa ψ che la sua trasformata Ψ siano due finestre con centri t^* e ω_+^* dati rispettivamente da

$$t^* = \frac{\int_{-\infty}^{\infty} t |\psi(t)|^2 dt}{\int_{-\infty}^{\infty} |\psi(t)|^2 dt} \quad (2.3)$$

$$\omega_+^* = \frac{\int_0^{\infty} \omega |\Psi(\omega)|^2 d\omega}{\int_0^{\infty} |\Psi(\omega)|^2 d\omega} \quad (2.4)$$

e raggi Δ_ψ e Δ_Ψ^+ dati da

$$\Delta_\psi = \left\{ \frac{\int_{-\infty}^{\infty} (t - t^*)^2 |\psi(t)|^2 dt}{\int_{-\infty}^{\infty} |\psi(t)|^2 dt} \right\}^{\frac{1}{2}} \quad (2.5)$$

$$\Delta_\Psi^+ = \left\{ \frac{\int_0^{\infty} (\omega - \omega_+^*)^2 |\Psi(\omega)|^2 d\omega}{\int_0^{\infty} |\Psi(\omega)|^2 d\omega} \right\}^{\frac{1}{2}}. \quad (2.6)$$

A questo punto, introducendo un fattore di normalizzazione $a^{-\frac{1}{2}}$, si definisce la trasformata wavelet continua del segnale $f(t)$ come:

$$(W_\psi f)(b, a) := \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt. \quad (2.7)$$

Quindi la trasformata da informazioni sul segnale $f(t)$ con una finestra temporale

$$[b + at^* - a\Delta_\psi, b + at^* + a\Delta_\psi]$$

centrata in $b + at^*$ e con raggio $a\Delta_\psi$. La finestra è più stretta per valori piccoli di a e si allarga all'aumentare del valore del parametro.

Considerando che sia il segnale da analizzare che la funzione $\psi(t)$ sono funzioni reali, abbiamo

$$F(-\omega)e^{jb\omega}\overline{\Psi(-a\omega)} = \overline{F(\omega)e^{-jb\omega}\Psi(a\omega)}.$$

Ricordando poi un'identità della teoria della trasformata di Fourier, chiamata *identità di Parseval*, per cui

$$\int_{-\infty}^{\infty} f(t)\overline{g(t)}dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\overline{G(\omega)} d\omega$$

e applicandola alla trasformata otteniamo:

$$\begin{aligned} (W_{\psi}f)(b, a) &= \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t)\overline{\psi\left(\frac{t-b}{a}\right)} dt \\ &= \frac{\sqrt{a}}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{-jb\omega}\overline{\Psi(a\omega)} d\omega \\ &= \frac{\sqrt{a}}{\pi} \operatorname{Re} \int_0^{\infty} F(\omega)e^{-jb\omega}\overline{\Psi(a\omega)} d\omega \\ &= \frac{\sqrt{a}}{\pi} \operatorname{Re} \int_0^{\infty} F(\omega)e^{-jb\omega}\overline{\eta\left(a\left(\omega - \frac{\omega_+^*}{a}\right)\right)} d\omega \end{aligned} \tag{2.8}$$

dove $\eta(\omega) := \Psi(\omega + \omega_+^*)$ è la finestra in frequenza posta nell'origine. Quindi la trasformata localizza anche nel dominio della frequenza lo spettro della funzione $f(t)$ con la finestra

$$\left[\frac{\omega_+^*}{a} - \frac{1}{a}\Delta_{\Psi}, \frac{\omega_+^*}{a} + \frac{1}{a}\Delta_{\Psi}\right]$$

e si può notare che scegliendo la wavelet $\psi(t)$ in modo tale che $\omega_+^* > \Delta_{\Psi}$ la finestra si muove lungo l'asse delle frequenze per valori decrescenti di a e che si restringe per valori crescenti del parametro. In realtà si vede che definita in questo modo la CWT fornisce un'analisi tempo-scala e non tempo-frequenza, ma si può mappare il parametro a sull'asse delle frequenze e determinare per ogni wavelet una relazione tra scala e frequenza per cui ad ogni valore di a corrisponderà un certo intervallo di frequenze.

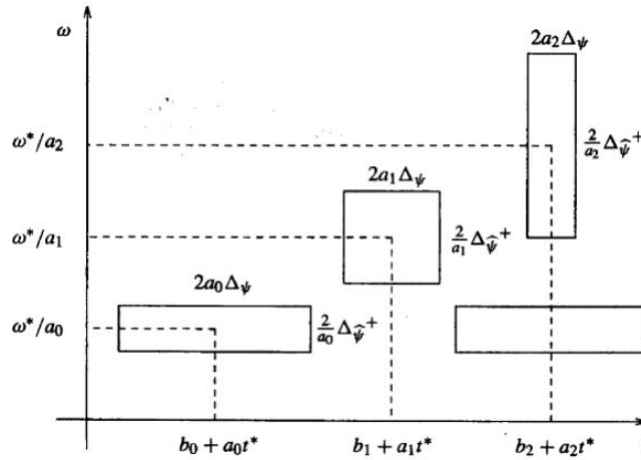


Figura 2.2: Finestre tempo-frequenza della CWT

2.2 Trasformata wavelet discreta

Con la CWT si analizza un certo segnale in modo continuo per mezzo di funzioni di base che sono delle versioni continuamente traslate e scalate di loro stesse. Questo comporta che tale trasformata abbia un certo grado di ridondanza, ovvero che molti dei valori prodotti non sono poi realmente utili per la ricostruzione del segnale. Inoltre per essere implementata richiede spesso un costo troppo oneroso, dovuto al calcolo dell'integrale di analisi, che non è accettabile nell'ambito del signal processing dove si richiedono efficienza e rapidità. Per questi motivi la CWT subisce un processo di discretizzazione che porta ad ottenere la *trasformata wavelet discreta* o *DWT*.

Quello che consente di ottenere una trasformata meno ridondante, ma con un numero di informazioni tali da consentire ancora la perfetta ricostruzione del segnale, è la scelta accurata dei valori dei parametri a e b . La trasformata discreta consiste essenzialmente nel prendere i parametri in questo modo:

$$a = 2^{-j} \quad (2.9)$$

$$b = k2^{-j} \quad (2.10)$$

Riconsiderando la trasformata continua con questi valori dei parametri di traslazione e scalatura, essa diventa

$$(W_{\psi}f)(k2^{-j}, 2^{-j}) = 2^{\frac{j}{2}} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t - k2^{-j}}{2^{-j}}\right)} dt = 2^{\frac{j}{2}} \int_{-\infty}^{\infty} f(t) \overline{\psi(2^j t - k)} dt \quad (2.11)$$

e per j e k fissati rappresenta un valore nel punto $(\frac{k}{2^j}, \frac{1}{2^j})$ del piano tempo-frequenza. Tali termini corrispondono ai coefficienti della trasformata discreta e dunque la DWT genera un insieme sparso di valori che rappresentano la correlazione tra il segnale analizzato e la wavelet.

Se in ingresso si considera una sequenza numerica, ottenuta per esempio del campionamento del segnale $f(t)$, l'integrale nella trasformata diventa una sommatoria

$$(W_\psi f)(k2^{-j}, 2^{-j}) \approx 2^{\frac{j}{2}} \sum_n f(n)\psi(2^j n - k). \quad (2.12)$$

2.3 Analisi Multirisoluzione

L'idea dell'analisi multirisoluzione (MRA) è quella di rappresentare e analizzare un segnale a risoluzioni diverse, in modo da osservare le caratteristiche di interesse alla risoluzione in cui siano più facilmente individuabili. Nella MRA una *funzione di scaling* viene usata per creare una serie di approssimazioni della funzione $f(t)$ ognuna delle quali differisce dalle altre per un fattore 2. Si usano poi ulteriori funzioni, che sono le *wavelets*, per aggiungere i dettagli che servono all'approssimazione per rappresentare effettivamente il segnale originale.

Per capire la teoria della analisi multirisoluzione consideriamo una funzione $f(t)$ che può essere rappresentata come una combinazione lineare di funzioni di espansione:

$$f(t) = \sum_k a_k \varphi_k(t) \quad (2.13)$$

dove a_k sono i coefficienti di espansione a valori reali, dati dalla relazione

$$a_k = \langle \varphi_k(t), f(t) \rangle = \int \varphi_k(t) f(t) dt. \quad (2.14)$$

L'insieme delle funzioni di base $\{\varphi_k(t)\}$ genera uno spazio di funzioni che viene definito come *span chiuso* dell'insieme di espansione ed è indicato come:

$$V = \overline{\text{Span}_k \{\varphi_k(t)\}} \quad (2.15)$$

per cui, dire che la funzione $f(t)$ appartiene allo spazio V vuol dire che appartiene allo span chiuso di $\{\varphi_k(t)\}$ e che può essere scritta nella forma dell'equazione 2.13.

2.3.1 Scaling function

Consideriamo come insieme di funzioni di espansione quello formato dalla funzione $\varphi(t)$, e dalle sue traslazioni intere e ridimensionamenti in scala

binaria, otteniamo quindi l'insieme $\{\varphi_{j,k}(t)\}$ dove

$$\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k) \quad (2.16)$$

per tutti $j, k \in \mathbb{Z}$, con $\mathbb{Z} = \{\dots, -1, 0, 1, \dots\}$ insieme dei numeri interi. Il termine k determina la posizione di $\varphi_{j,k}(t)$ lungo l'asse dei tempi, j determina l'ampiezza di $\varphi_{j,k}(t)$, e quindi come si restringe o allarga lungo l'asse temporale, il termine $2^{j/2}$ è una costante di normalizzazione. Questa funzione $\varphi_{j,k}(t)$ viene chiamata *funzione di scaling* per il fatto che la sua forma varia al variare di j . Per ogni valore di j fissato si determina uno spazio V_j e se la funzione $f(t) \in V_j$ può essere scritta come

$$f(t) = \sum_k a_k \varphi_{j,k}(t). \quad (2.17)$$

Aumentando il valore di j aumenta anche la misura dello spazio V_j corrispondente, perchè le funzioni $\varphi_{j,k}(t)$ vengono compresse e sono in grado di individuare maggiori variazioni e di analizzare più dettagli. Si può quindi rappresentare un maggior numero di funzioni, che verranno incluse nel sottospazio in questione. Questo vuol dire che ciascuno spazio V_j determinato dalle funzioni di scaling a scala minore è incluso nello spazio V_{j+1} formato dalle funzioni di scaling a scala maggiore. Gli spazi che contengono funzioni ad alta risoluzione devono necessariamente contenere anche tutte quelle a risoluzione più bassa, ovvero

$$V_{-\infty} \subset \dots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots \subset V_{\infty} \quad (2.18)$$

e se la funzione $f(t) \in V_j$ avremo che la funzione $f(2t) \in V_{j+1}$.

Al limite per $j \rightarrow \infty$ lo spazio V_j dovrebbe includere tutte le funzioni del dominio di interesse; poichè stiamo considerando funzioni reali, misurabili e integrabili al quadrato, questo vuol dire che per noi

$$V_{\infty} = \{L^2(\mathbb{R})\} \quad (2.19)$$

e che le serie di funzioni di espansione del sottospazio V_j può essere determinata tramite somme pesate delle funzioni di espansione a più alta risoluzione del sottospazio V_{j+1} , in questo modo:

$$\varphi_{j,l}(t) = \sum_k p(k) 2^{(j+1)/2} \varphi(2^{j+1} t - k). \quad (2.20)$$

La scelta del sottospazio di riferimento è arbitraria, per cui riferendoci allo spazio V_0 possiamo considerare $\varphi(t) = \varphi_{0,0}(t)$, con entrambi i parametri j e l nulli, e ottenere un'espressione più semplice

$$\varphi(t) = \sum_k p(k) \sqrt{2} \varphi(2t - k) \quad (2.21)$$

dove $p(k)$ sono chiamati *coefficienti della funzione di scaling*.

L'equazione 2.21 è fondamentale per l'analisi multirisoluzione e viene chiamata *equazione di raffinamento* o *equazione di dilatazione*.

La funzione di scaling deve essere tale da creare degli spazi annidati l'uno nell'altro come abbiamo detto e per cui al limite per $j \rightarrow \infty$ lo spazio V_∞ include tutte le funzioni presenti in $L^2(\mathbb{R})$; ci sono poi altre due condizioni che la funzione $\varphi(t)$ deve soddisfare per garantire l'analisi multirisoluzione:

- la funzione di scaling deve essere ortogonale rispetto alle sue traslazioni intere;
- l'unica funzione comune a tutti gli spazi $V(j)$ è la funzione $f(t) = 0$, il che equivale anche a dire che nello spazio a risoluzione più bassa, corrispondente a quello per $j = -\infty$, l'unica funzione rappresentabile è la funzione con informazione nulla

$$V_{-\infty} = \{0\}. \quad (2.22)$$

2.3.2 Wavelet

Uno spazio a risoluzione $j + 1$ è formato da tutte le funzioni a risoluzione inferiore a cui vengono aggiunti dei dettagli e questi dettagli sono dati da una certa funzione *wavelet* $\psi(t)$. Tale funzione, con le sue traslazioni intere e i ridimensionamenti in scala binaria, da origine ad uno span chiuso che è l'insieme delle differenze esistenti tra i due sottospazi adiacenti V_j e V_{j+1} . Questa situazione è rappresentabile graficamente come in Fig.2.3.

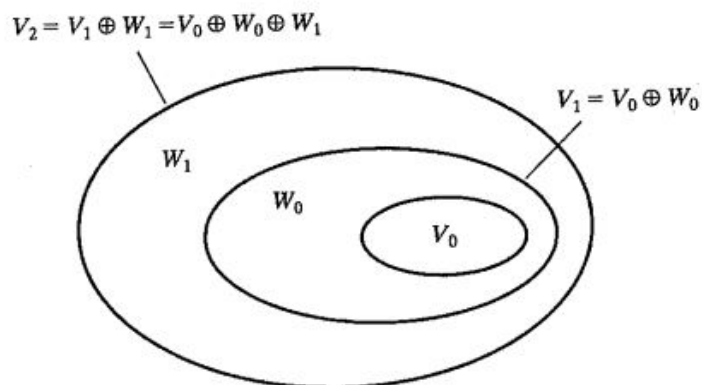


Figura 2.3: Relazione tra gli spazi delle funzioni wavelet e di scaling

Come già visto per la funzione di scaling, abbiamo l'insieme $\{\psi_{j,k}(t)\}$ formato dalle wavelets

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \quad (2.23)$$

che determinano lo spazio

$$W_j = \overline{\text{Span}_k\{\psi_{j,k}(t)\}}. \quad (2.24)$$

Se la funzione $f(t)$ appartiene a tale spazio allora potrà essere rappresentata tramite le wavelets di base $\psi_{j,k}(t)$

$$f(t) = \sum_k a_k \psi_{j,k}(t). \quad (2.25)$$

La relazione tra i sottospazi delle scaling function e i sottospazi delle wavelets è la seguente

$$V_{j+1} = V_j \oplus W_j \quad (2.26)$$

dove \oplus è il simbolo che determina l'unione tra spazi o insiemi, e tutti i termini di V_j sono ortogonali rispetto a quelli di W_j , ovvero

$$\langle \varphi_{j,k}(t), \psi_{j,\ell}(t) \rangle = 0 \quad (2.27)$$

per valori di $j, k, \ell \in \mathbb{Z}$. Si può quindi esprimere lo spazio di tutte le funzioni misurabili e integrabili al quadrato come

$$L^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus \dots \quad (2.28)$$

e se $f(t)$ è un elemento dello spazio V_1 , ma non di V_0 , si può rappresentare la funzione con un'approssimazione data dalle funzioni di scaling dello spazio V_0 mentre le wavelets dello spazio W_0 aggiungono le differenze tra la funzione originale e l'approssimazione ottenuta.

Uno spazio di wavelets ad una certa risoluzione è incluso nello spazio delle funzioni di scaling a risoluzione più alta per cui qualunque funzione wavelet può essere espressa, allo stesso modo della funzione $\varphi(t)$ nell'Eq.2.21, tramite una somma pesata di funzioni di scaling traslate e a risoluzione doppia

$$\psi(t) = \sum_k q(k)\sqrt{2}\varphi(2x - k) \quad (2.29)$$

dove $q(k)$ sono i coefficienti della funzione wavelet e sono legati ai coefficienti della funzione di scaling $p(k)$ dalla relazione

$$q(k) = (-1)^k p(1 - k). \quad (2.30)$$

Esistono numerose funzioni che rientrano nella definizione di wavelet ed è possibile individuare *famiglie* intere in cui le funzioni hanno caratteristiche tra loro comuni. Nel lavoro di questa tesi quelle utilizzate sono state le seguenti:

- HAAR. È stata la prima wavelet ad essere proposta ed è quella più semplice. Ha lo svantaggio di non essere una funzione continua, dunque non è derivabile. La wavelet è la funzione

$$\psi(t) = \begin{cases} 1 & 0 \leq t \leq \frac{1}{2}, \\ -1 & \frac{1}{2} \leq t \leq 1, \\ 0 & \text{altrove,} \end{cases} \quad (2.31)$$

e la funzione di scaling è

$$\varphi(t) = \begin{cases} 1 & 0 \leq t \leq 1, \\ 0 & \text{altrove.} \end{cases} \quad (2.32)$$

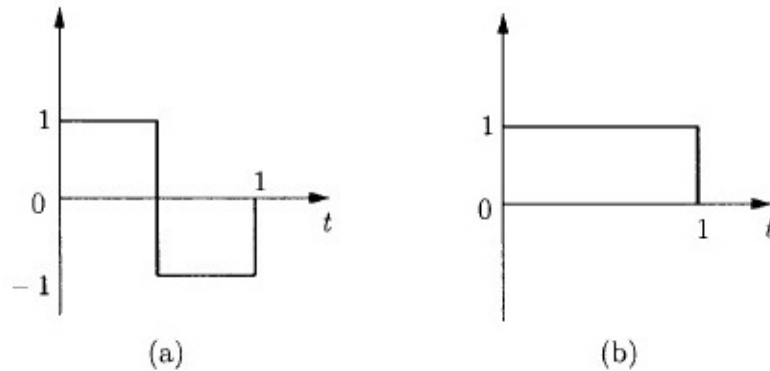
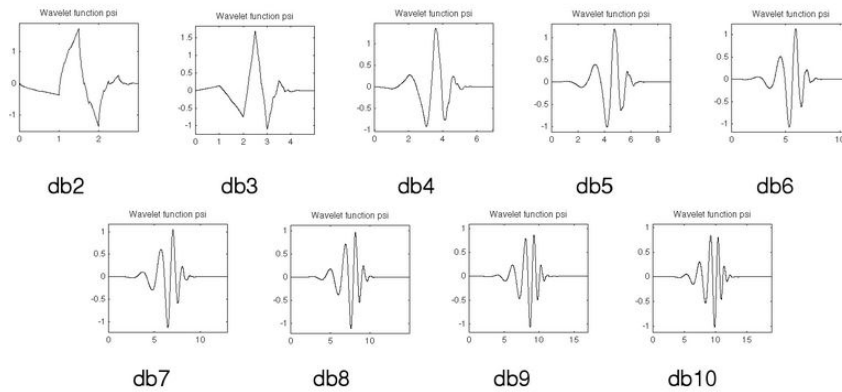
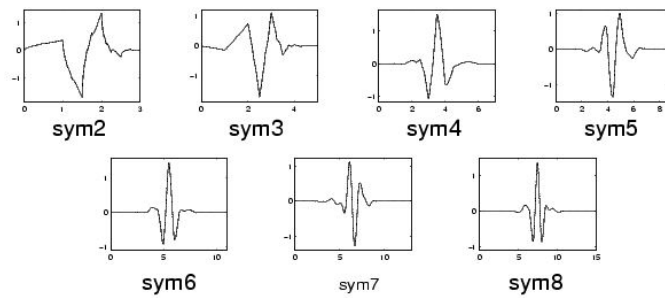


Figura 2.4: Wavelet di Haar (a) e funzione di scaling (b)

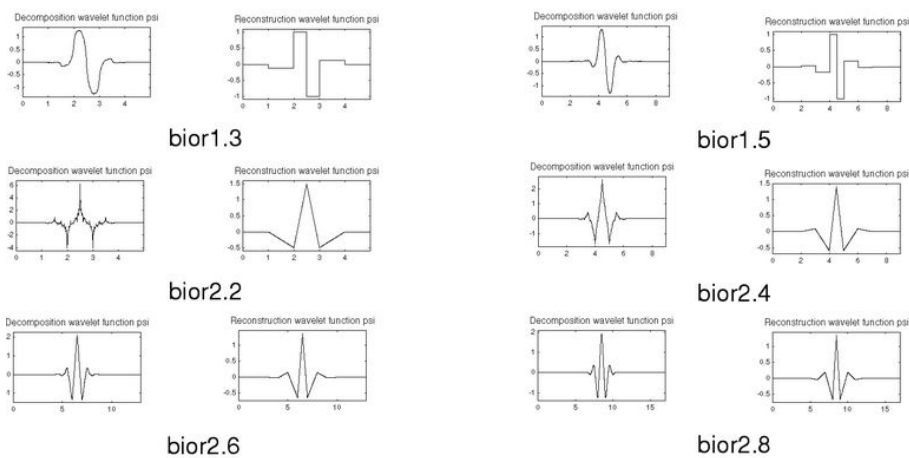
- DAUBECHIES. Si tratta di una famiglia di wavelets ortonormali molto usata per l'analisi discreta. Sono indicate come *dbN* dove N è l'ordine della wavelet. Sono generalmente asimmetriche e diventano sempre più regolari all'aumentare di N, ossia all'aumentare dei primi N momenti nulli.

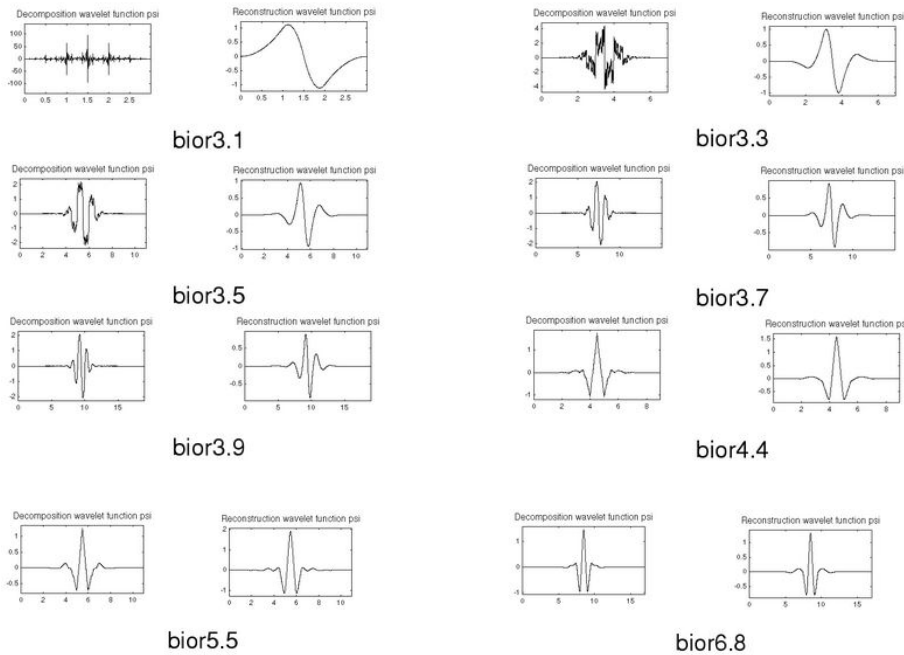


- SYMLETS. È una famiglia di wavelets che possono essere considerate delle varianti rispetto alle daubechies con maggiori proprietà di simmetria. Si indicano come *symN*.

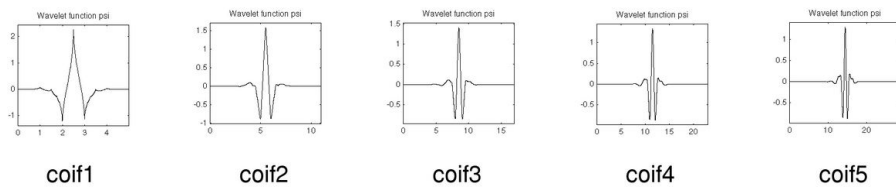


- BIORTHOGONAL. Sono molto usate per la ricostruzione di immagini e sono indicate come *biorNr.Nd*. Sono utilizzate in coppia, una serve per la ricostruzione (*Nr*) e l'altra per la decomposizione (*Nd*).





- COIFLETS. Vengono indicate con $coifN$ e hanno i primi $2N$ momenti nulli.



2.4 Trasformata wavelet veloce

La trasformata wavelet discreta riveste una particolare importanza nell'ambito dell'elaborazione dei segnali perchè può essere implementata in modo efficiente. Si determina in questo modo la trasformata wavelet veloce, algoritmo che sfrutta la relazione tra i coefficienti della DWT di due scale adiacenti. É il corrispondente della FFT per l'analisi wavelet.

Il problema principale consiste nello stabilire come decomporre un segnale nei sui coefficienti wavelet e come ricostruire il segnale originale a partire da tali coefficienti. Innanzitutto il segnale di ingresso $f(t)$ dovrà essere rappresentato tramite una funzione $f_n(t) \in V_n$ e poichè sarà quest'ultima ad essere

analizzata e poi ricostruita è evidente che $f_n(t)$ dovrà essere scelta accuratamente. Si intuisce quindi che la funzione di scaling $\varphi(t)$ e il sottospazio V_n che le corrisponde dipenderanno dalla forma del segnale originale $f(t)$. Una volta stabilito che $\varphi(t)$ è la funzione adatta, si rappresenta il segnale come

$$f_n(t) = \sqrt{2} \sum_k c_{n,k} \varphi(2^n t - k) \quad (2.33)$$

e l'algoritmo veloce può essere applicato alla serie di coefficienti

$$\mathbf{c}_n = \{c_{n,k}\} \quad (2.34)$$

invece che ai dati in ingresso originali.

Abbiamo già visto quale sia la struttura generale dell'analisi multirisoluzione e come siamo organizzati tra loro gli spazi delle funzioni di scaling e delle wavelets, lo spazio V_n che stiamo considerando è quindi dato dall'unione degli spazi a risoluzione inferiore

$$V_n = V_{n-1} \oplus W_{n-1}. \quad (2.35)$$

La funzione $f_n(t)$ è allora scomponibile in

$$f_n(t) = f_{n-1} + g_{n-1} \quad (2.36)$$

dove $f_{n-1}(t) \in V_{n-1}$ e $g_{n-1} \in W_{n-1}$ e possono essere rappresentate, tramite le funzioni di scaling e le wavelets che determinano i sottospazi considerati, nel modo seguente:

$$f_{n-1}(t) = \sqrt{2} \sum_k c_{n-1,k} \varphi(2^{n-1} t - k) \quad (2.37)$$

$$g_{n-1}(t) = \sqrt{2} \sum_k d_{n-1,k} \psi(2^{n-1} t - k). \quad (2.38)$$

A partire quindi dalla serie di coefficienti in ingresso $\mathbf{c}_n = \{c_{n,k}\}$, che rappresenta i dati originale, si devono ottenere delle nuove serie di coefficienti $\mathbf{c}_{n-1} = \{c_{n-1,k}\}$ e $\mathbf{d}_{n-1} = \{d_{n-1,k}\}$ tramite un'operazione di filtraggio con banco di filtri. Il banco di filtri può essere realizzato in modi diversi, qui consideriamo il caso in cui lo spettro dell'ingresso venga suddiviso in due parti uguali, la prima ottenuta con un filtro passa-basso la seconda con un filtro passa-alto. La parte del filtraggio passa-basso è quella che contiene l'approssimazione del segnale e la maggior parte della sua energia mentre la parte passa-alto contiene i dettagli. Il processo può essere ripetuto fino ad un certo livello m con cui il segnale sarà rappresentato a risoluzione inferiore e a cui

saranno aggiunti tutti i dettagli necessari per riportarlo al livello originale, ovvero

$$f_n = g_{n-1} + g_{n-2} + \dots + g_{n-m} + f_{n-m}. \quad (2.39)$$

Il numero di campioni totali deve rimanere lo stesso di quello iniziale, quindi ogni passaggio richiede anche un'operazione di sottocampionamento, ossia nel nostro caso vengono mantenuti solo i coefficienti con indice pari. Tutto questo può essere schematizzato come in Fig.2.5, dove le serie $\{a_\ell\}$ e $\{b_\ell\}$ sono chiamate *sequenze di decomposizione* e sono tali per cui

$$\begin{cases} c_{n-1,k} = \sqrt{2} \sum_{\ell} a_{\ell-2k} c_{n,\ell} ; \\ d_{n-1,k} = \sqrt{2} \sum_{\ell} b_{\ell-2k} c_{n,\ell} . \end{cases} \quad (2.40)$$

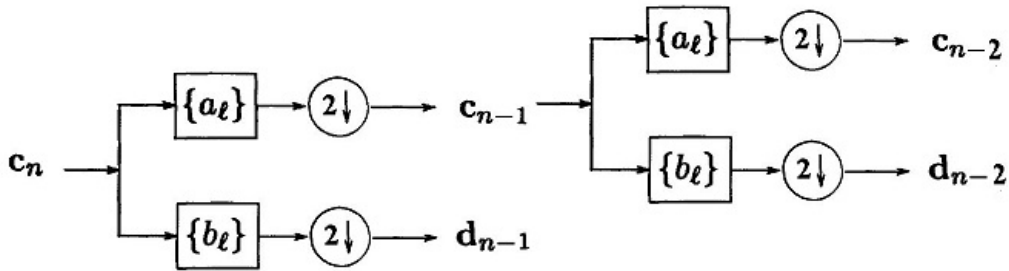


Figura 2.5: Scomposizione wavelet

Per la ricostruzione del segnale originale invece a partire dai coefficienti a risoluzione minore si calcolano quelli di livello superiore attraverso le serie $\{p_k\}$ e $\{q_k\}$ che sono chiamate *sequenze di ricostruzione*. In questo caso si deve fare un'operazione di sovracampionamento per riportare il numero di coefficienti al valore iniziale, per cui i campioni $c_{n-1,\ell}$ e $d_{n-1,\ell}$ sono usati per gli indici pari $m = 2\ell$ mentre per gli indici dispari $m = 2\ell + 1$ si inseriscono degli zeri.

$$c_{n,k} = \frac{1}{\sqrt{2}} \sum_{\ell} [p_{k-2\ell} c_{n-1,\ell} + q_{k-2\ell} d_{n-1,\ell}]. \quad (2.41)$$

2.4.1 Implementazione degli algoritmi

Per implementare in MATLAB gli algoritmi di scomposizione e ricostruzione visti precedentemente consideriamo un caso semplice utilizzando la wavelet di Haar. Innanzitutto avremo un segnale di ingresso discreto, quindi un vettore

di $N = 2^j$ elementi, e il massimo livello di scomposizione possibile sarà $n = \log_2 N$. Dal sistema scritto in 2.40 possiamo passare ad una notazione matriciale per cui

$$\begin{bmatrix} \mathbf{c}_{n-1} \\ \mathbf{d}_{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \begin{bmatrix} c_n \end{bmatrix}.$$

I vettori in uscita c_{n-1} e d_{n-1} , che sono rispettivamente i coefficienti dell'approssimazione e i coefficienti del livello di dettaglio, hanno un numero di elementi pari a $\frac{N}{2}$, ossia la metà di quello del vettore iniziale. Le due matrici \mathbf{A} e \mathbf{B} contengono gli elementi $\{a_\ell\}$ e $\{b_\ell\}$ delle sequenze di decomposizione. Entrambe le matrici avranno k righe, con $k = \frac{N}{2}$, e $\ell = N$ colonne. Gli elementi presenti nelle matrici, nel caso di Haar, hanno i seguenti valori

$$\begin{cases} \mathbf{A}_{k,\ell} = a_{2k-\ell}, & \text{con } a_0 = a_1 = \frac{1}{2} \text{ e } a_i = 0 \text{ per } i = 2k - \ell \neq 0, 1 \\ \mathbf{B}_{k,\ell} = (-1)^{\ell+1} b_{\ell+1-2k} & \text{con } b_0 = b_1 = \frac{1}{2} \text{ e } b_i = 0 \text{ per } i = \ell + 1 - 2k \neq 0, 1. \end{cases} \quad (2.42)$$

Per un vettore in ingresso, per esempio di 4 elementi, le due matrici avranno le seguenti strutture:

$$\mathbf{A} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}.$$

Creiamo una funzione in MATLAB che stabilisca come siano le due matrici quando riceve in ingresso un certo vettore \mathbf{v} .

```

1 function [A,B] = decmat(v)
2 N=length(v);
3 for k=1:(N/2)
4     for l=1:N
5         if ((2*k)-1)==1 | ((2*k)-1)==0
6             a=(1/2);
7         else a=0;
8         end
9         A(k,l)=a;
10    end
11 end
12
13 for k=1:(N/2)
14     for l=1:N
15         if (1+1-(2*k))==1 | (1+1-(2*k))==0
16             b=1/2;
17         else b=0;
18         end
19         B(k,l)=((-1)^(1+1))*b;

```

```

    end
21 end

```

La decomposizione può essere fatto fino al livello $n = 0$, ossia il ciclo può essere iterato n volte, ogni volta il nuovo vettore \mathbf{v} in ingresso da considerare sarà quello contenente i coefficienti di approssimazione c_{n-1} ottenuti dalla scomposizione precedente e le dimensioni delle matrici e dei vettori saranno dimezzate di volta in volta. La decomposizione totale si può determinare con un'altra funzione

```

1 function [C]= decomposition (x)
  cn=x(:);
3 N=length(x);
  n=log2(N);
5 C = zeros(N,1);
  for i=1:n
7     [A,B]=decmat(cn);
      L=length(cn);
9     c=(sqrt(2))*(A*cn);
      d=(sqrt(2))*(B*cn);
11    C(1:(L/2))=c;
      C(((L/2)+1):L)=d;
13    cn=c;
  end
15 cn=x(:);
  N=length(x);
17 n=log2(N);
  C = zeros(N,1);
19 for i=1:n
      [A,B]=decmat(cn);
21    L=length(cn);
      c=(sqrt(2))*(A*cn);
23    d=(sqrt(2))*(B*cn);
      C(1:(L/2))=c;
25    C(((L/2)+1):L)=d;
      cn=c;
27 end

```

da cui si ottiene come uscita un unico vettore \mathbf{C} , in cui il primo elemento è il coefficiente di approssimazione c_0 e gli altri sono i coefficienti dei dettagli. Se il segnale in ingresso avesse un numero di elementi $N = 8$ i coefficienti all'interno di \mathbf{C} sarebbero suddivisi in questo modo

$$\mathbf{C} = \left[\underbrace{C(1)}_{c_0} \underbrace{C(2)}_{d_0} \underbrace{C(3) \ C(4)}_{d_1} \underbrace{C(5) \ C(6) \ C(7) \ C(8)}_{d_2} \right].$$

Per l'algoritmo di ricostruzione possiamo ugualmente creare delle funzioni che determinino due matrici, \mathbf{P} e \mathbf{Q} , che contengono i termini delle sequenze di ricostruzione $\{p_k\}$ e $\{q_k\}$. Entrambe le matrici avranno k righe, con $k = N$ e $\ell = \frac{N}{2}$ colonne. Gli elementi presenti nelle matrici, nel caso di Haar, hanno i seguenti valori

$$\begin{cases} \mathbf{P}_{k,\ell} = p_{2\ell-k}, & \text{con } p_0 = p_1 = 1 \text{ e } p_i = 0 \text{ per } i = 2\ell - k\ell \neq 0, 1 \\ \mathbf{Q}_{k,\ell} = (-1)^{k\ell+1} q_{k+1-2\ell} & \text{con } q_0 = q_1 = 1 \text{ e } q_i = 0 \text{ per } i = k+1 - 2\ell \neq 0, 1. \end{cases} \quad (2.43)$$

Considerando la formula 2.41, in forma matriciale possiamo scrivere

$$\begin{bmatrix} \mathbf{c}_n \end{bmatrix} = \begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{c}_{n-1} \\ \mathbf{d}_{n-1} \end{bmatrix}.$$

Con un procedimento che è praticamente uguale ed opposto rispetto a quello trattato precedentemente possiamo costruire altre due funzioni, una per creare le matrici \mathbf{P} e \mathbf{Q} e una per riottenere il segnale originale a partire dal vettore \mathbf{C} che era l'uscita dell'algoritmo di decomposizione.

```

1 function [P,Q] = recmat(v)
N=length(v);
3 for l=1:(N/2)
    for k=1:N
5         if ((2*l)-k)==0 | ((2*l)-k)==1
                p=1;
7         else p=0;
            end
9         P(k,l)=p;
    end
11 end
for l=1:(N/2)
13     for k=1:N
15         if (k+1-(2*l))==1 | (k+1-(2*l))==0
                q=1;
17         else q=0;
            end
                Q(k,l)=((-1)^(k+1))*q;
19     end
end
21 end
end
23 end

```

```

function [X] = reconstruction(C)
2 C=C(:);
  a=C(1);
4 j=2;
  N=length(C);
6 n=log2(N);
  for i=1:n
8     b=C(j:(2^i));
      cn=[a;b];
10    [P,Q]= recmat(cn);
      R=[P Q];
12    s=(1/(sqrt(2))*(R*cn));
      a=s;
14    j=(2^i)+1;
  end
16 X=a;

```

2.5 Test e confronti

A questo punto possiamo fare alcuni confronti tra l'analisi wavelet e l'analisi di Fourier nel caso di segnali monodimensionali, per capire in quali situazioni e in quale modo possa essere più utile la trasformata wavelet rispetto a quella di Fourier.

Vediamo alcuni segnali semplici, quali informazioni siamo in grado di ricavare dai coefficienti e in che modo venga ricostruito il segnale dopo delle operazioni di filtraggio sui coefficienti stessi.

La decomposizione multilivello di un segnale, nel Wavelet Toolbox di MATLAB viene fatta tramite una funzione chiamata **wavedec**

$$[C, L] = \text{wavedec}(X, N, 'wname')$$

in cui X contiene i dati del segnale in ingresso, N è il massimo livello della decomposizione a cui si vuole arrivare, *'wname'* è il nome della wavelet. In uscita si ottengono due vettori, C contiene i coefficienti fino all'ordine N fissato mentre L contiene tutte le dimensioni dei relativi vettori. Per la ricostruzione invece si usa un'altra funzione chiamata **waverec**

$$X_{ric} = \text{waverec}(C, L, 'wname')$$

da cui si riottiene il segnale dato in ingresso.

Anche la trasformata di Fourier veloce è implementata in MATLAB tramite

la funzione **fft** da cui, se $y = \text{fft}(x)$, si ha che

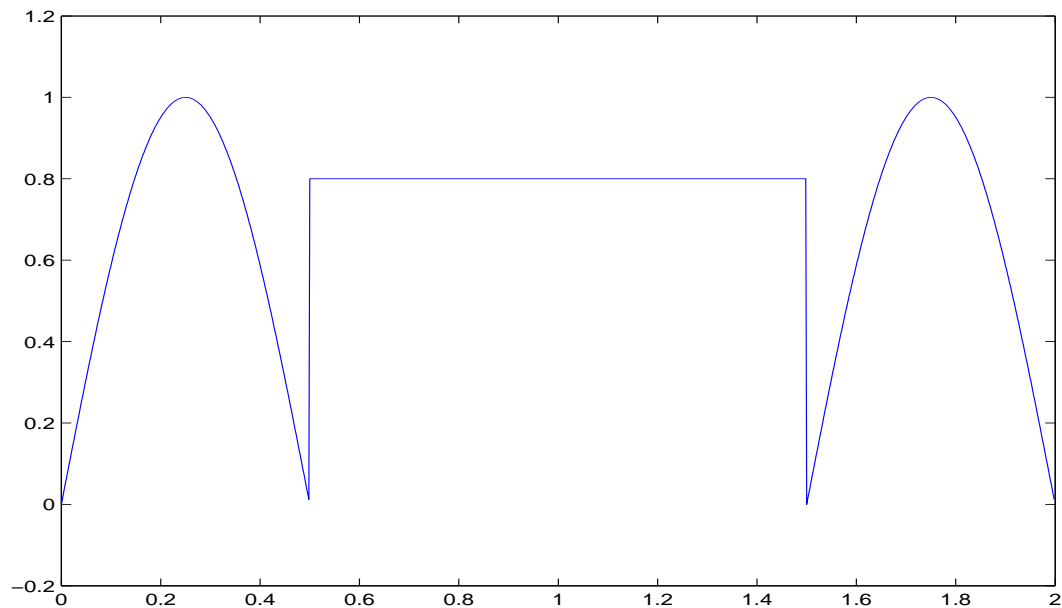
$$y_k = \sum_{j=1}^N x_j \omega_N^{-(j-1)(k-1)}.$$

Questa espressione è leggermente diversa rispetto a come abbiamo definito la trasformata discreta precedentemente, per cui per ottenere $y = \text{DFT}(x)$, si deve calcolare

$$y = \frac{1}{N} \text{fft}(x).$$

Inoltre bisogna considerare un'operazione di shift da effettuare sui valori trasformati, ottenibile con la funzione **fftshift**. Questo perchè vogliamo una rappresentazione degli elementi simmetrica rispetto a $x = 0$ e consideriamo la sommatoria nella DFT per $j = [-\frac{N}{2} + 1 : \frac{N}{2}]$ e non per $j = [1 : N]$ come fa la funzione di MATLAB.

Consideriamo un segnale di questo tipo



in cui ci siano dei netti cambiamenti di andamento, e inizialmente lo analizziamo tramite trasformata wavelet continua. Utilizziamo la wavelet di Haar e decomponiamo il segnale fino al livello 10, che in questo caso corrisponde al massimo livello di approssimazione possibile poichè abbiamo $N = 1024$ dati in ingresso. Questi coefficienti possono essere plottati in un grafico in cui l'asse x rappresenta la posizione del segnale nel tempo o nello spazio mentre

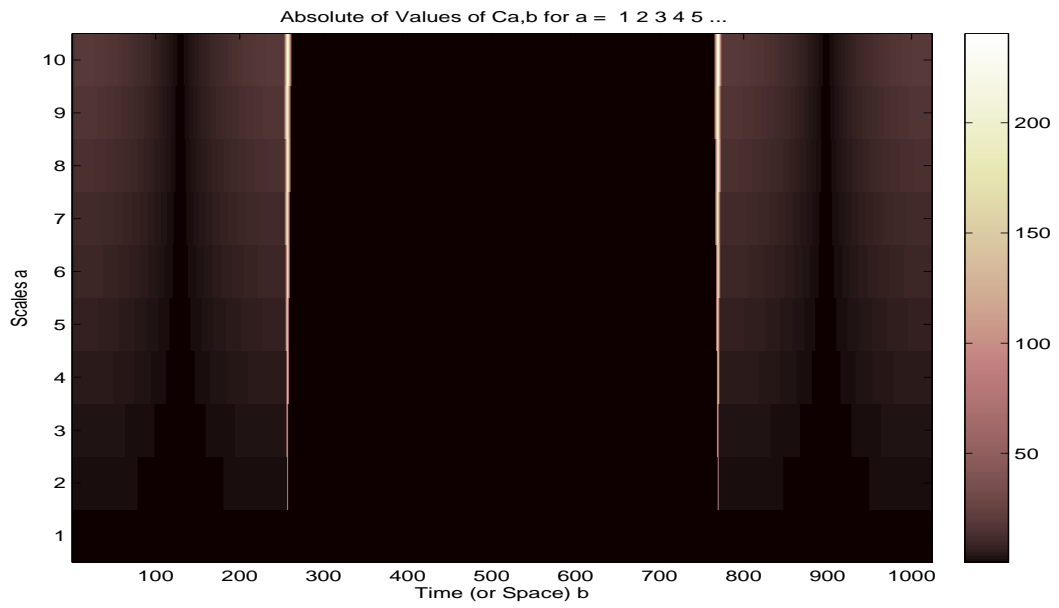


Figura 2.6: Coefficienti della CWT

l'asse y rappresenta la scala. Il colore di ciascun punto di coordinate x, y rappresenta la magnitudo dei coefficienti C della wavelet.

Si vede subito come dal grafico sia possibile individuare l'istante esatto in cui avvengono i cambiamenti del segnale, aspetto che non è possibile notare dal grafico dei coefficienti della trasformata di Fourier, in cui vengono riportate tutte le frequenze presenti nel segnale ma non dove cambino da un istante a quello successivo. Il segnale è costruito con N punti, per capire la corrispondenza tra gli indici dell'asse x del grafico dei coefficienti e l'istante temporale nel grafico del segnale, bisogna considerare $x_1 = a$, dove a è il momento in cui comincia il segnale, $x_N = b$, con b l'istante finale e h è la distanza tra due punti consecutivi

$$h = \frac{b - a}{N - 1}.$$

Se dal grafico dei coefficienti ricaviamo l'indice i in cui si ha la variazione del segnale il momento corrispondente x_i lo troviamo come

$$x_i = a + (i - 1)h = a + \frac{i - 1}{N - 1}(b - a).$$

Applichiamo anche la trasformata discreta, in questo caso il grafico nell'asse y riporterà i diversi livelli della decomposizione (Fig.2.7).

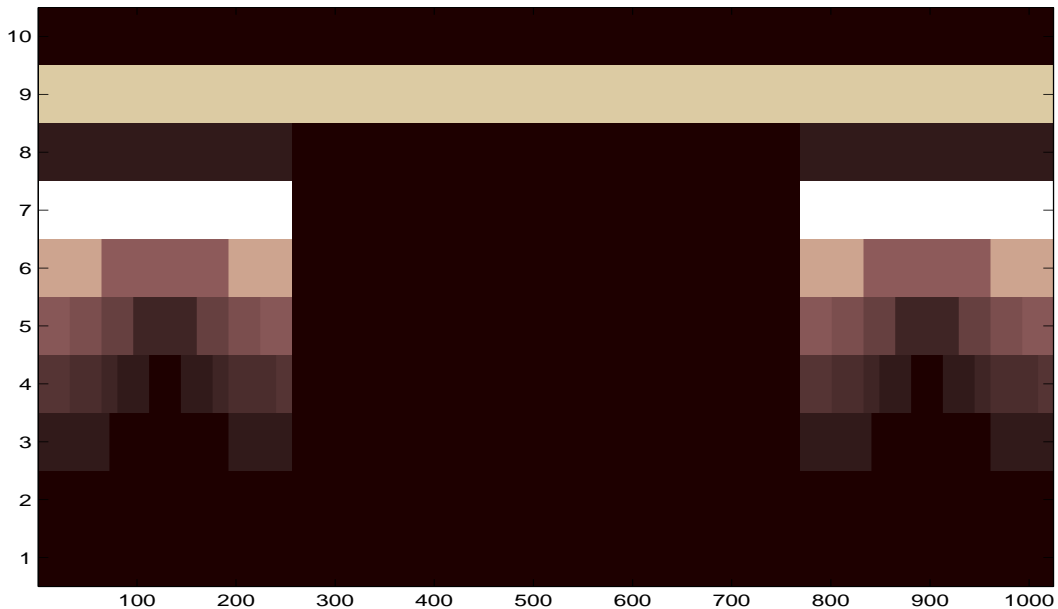


Figura 2.7: Coefficienti della DWT

La maggior parte di questi coefficienti è nulla o di valore piccolo; quindi molti di questi valori non sono necessari e potranno essere tralasciati quando

si vuole riottenere il segnale originale. Nel caso in cui il segnale dovesse essere trasferito o memorizzato il numero di informazione utile è dunque nettamente inferiore rispetto alla trasformata di Fourier. Possiamo fare dei confronti fra alcuni tipi diversi di wavelet e la FFT. Dopo aver trasformato il segnale stabiliamo una certa soglia. Il valore della soglia è calcolato rispetto al coefficiente, della FFT e di ciascuna trasformata wavelet, il cui valore assoluto è massimo, in questo caso ha valore dato da

$$10^{-3} \times \max_{i=1, \dots, N} \|C_i\|$$

dove C_i sono i coefficienti a seconda della trasformata considerata. I coefficienti con valore assoluto minore di tale soglia li poniamo nulli e ricostruiamo il segnale con i soli coefficienti rimasti. Nella tabella seguente vediamo il numero di coefficienti mantenuti e l'errore relativo per ciascuno dei casi presi in esame

wavelets	errore relativo	coefficienti	compressione
FFT	1.51e-02	533	0.520508
haar	1.72e-02	139	0.135742
db2	7.93e-03	85	0.0830078
sym3	6.13e-03	90	0.0878906
sym7	4.12e-03	148	0.144531
coif3	3.69e-03	153	0.149414
coif5	4.92e-03	214	0.208984

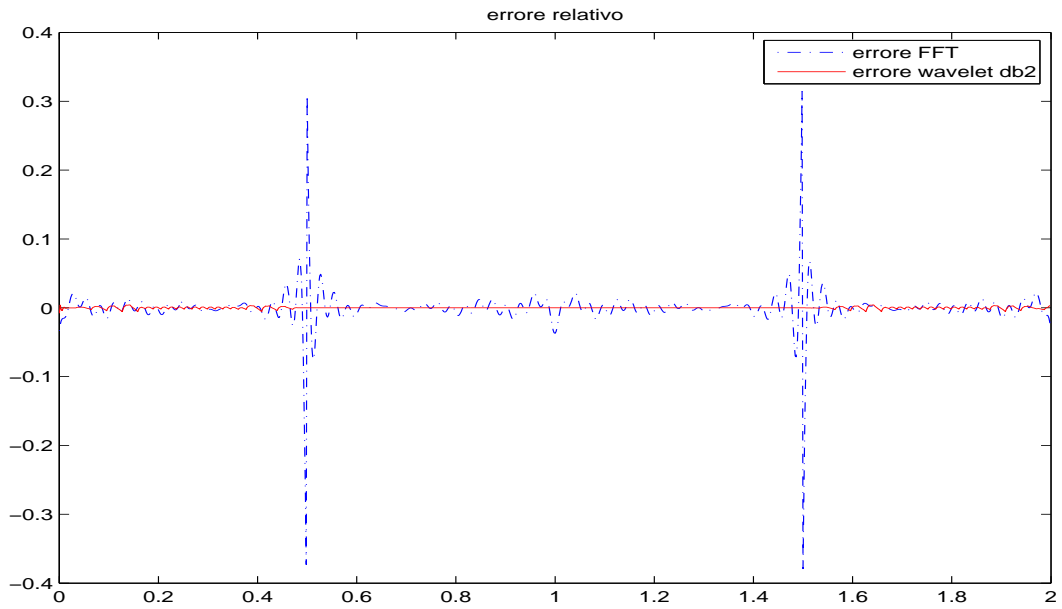
L'errore relativo è calcolato in questo modo

$$\frac{\|f - f_{ric}\|_{\infty}}{\|f\|_{\infty}}$$

dove f è il segnale originale, f_{ric} è il segnale ricostruito dopo essere stato trasformato e filtrato, $\|\cdot\|_{\infty}$ è la norma infinito che ci da una valutazione dell'errore punto per punto. Dato un vettore \mathbf{X} con x_n elementi, la norma infinito è definita come $\|x\|_{\infty} = \max_{i=1, \dots, n} \|x_i\|$.

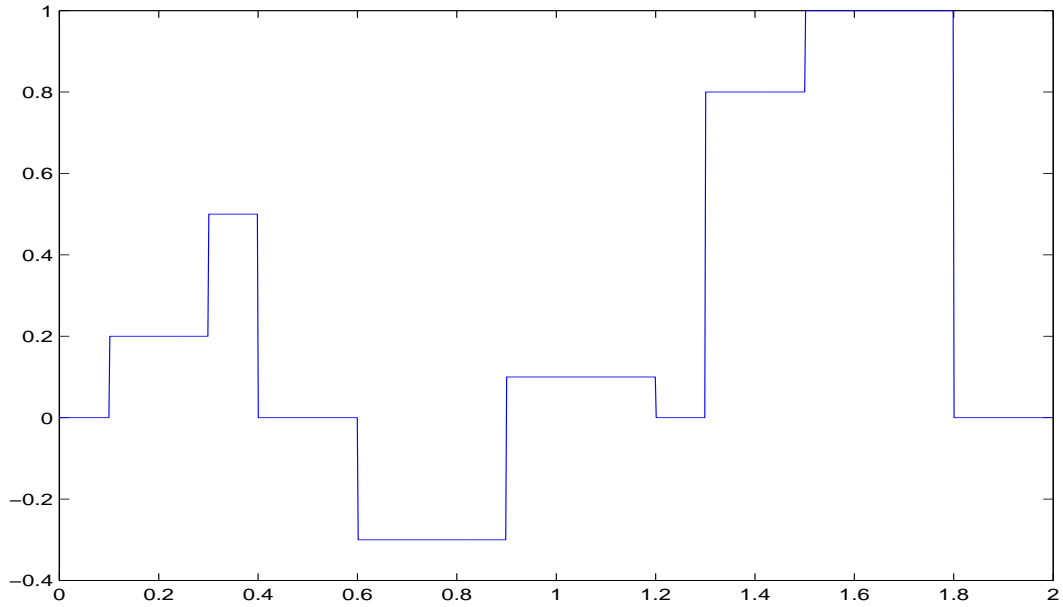
Dai dati in tabella si vede chiaramente come, in generale, con le wavelet siano necessari molti meno coefficienti per ricostruire il segnale mantenendo un errore sempre minore rispetto alla trasformata di Fourier. Usando la wavelet *db2* con solo l'8% dei coefficienti si ha una buona ricostruzione del segnale, mentre con lo stesso numero di coefficienti l'errore della FFT è maggiore (Fig2.8).

Il grafico dell'errore della FFT confrontato con quello della *db2* è il seguente



in cui si vede come in generale, e soprattutto in corrispondenza delle due variazioni nell'andamento del segnale, l'errore della trasformata di Fourier sia più grande.

La scelta della wavelet di analisi può portare a migliorare ulteriormente l'errore ottenuto e a ricostruire in modo più accurato il segnale. Se consideriamo un segnale costante a tratti come il seguente si vede come la wavelet di Haar sia la migliore per analizzare e ricostruire il segnale.



Come vediamo nella tabella 2.1, con la wavelet di Haar ricostruiamo il segnale con soli 63 coefficienti, rispetto ai 1024 totali, e otteniamo un errore dell'ordine di 10^{-15} , ossia si può dire che il segnale originale e quello ricostruito sono praticamente uguali.

Tabella 2.1:

wavelets	errore relativo	coefficienti	compressione
FFT	1.40e-02	853	0.833008
haar	1.33e-15	63	0.0615234
db2	5.18e-03	130	0.126953
sym3	5.82e-03	166	0.162109
sym7	4.79e-03	227	0.22168
coif3	3.75e-03	221	0.21582
coif5	4.72e-03	259	0.25293

Quindi si vede che ci sono diversi casi in cui l'analisi tramite wavelet risulta migliore; date le loro caratteristiche sono in grado di individuare senza problemi le variazioni di andamento o di frequenza di un segnale. Inoltre sono notevolmente più utili per la compressione dei dati, in tutti i casi visti il numero di coefficienti necessari per ricostruire il segnale originale è decisamente minore, così come il rispettivo errore, rispetto a quello della FFT.

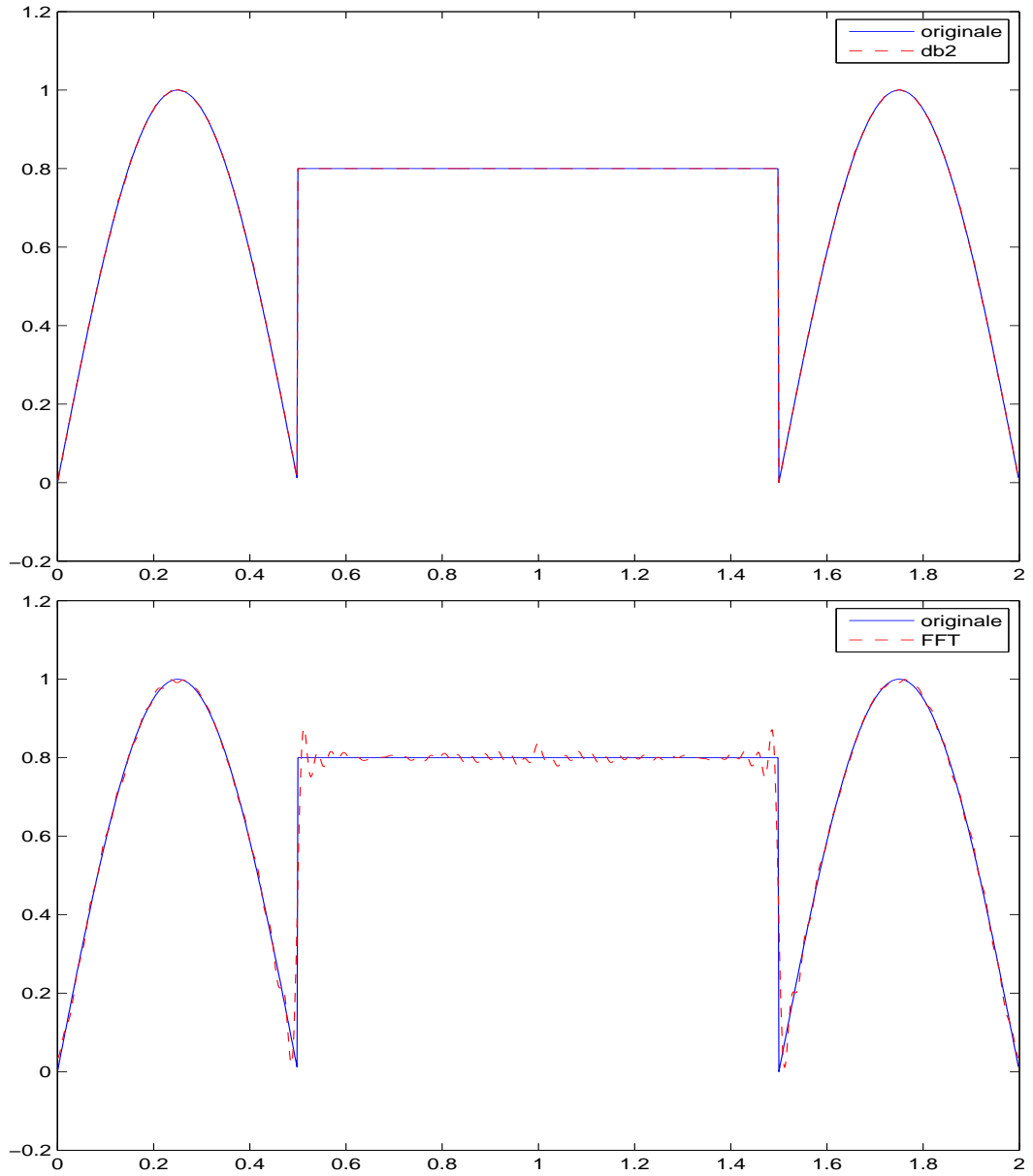


Figura 2.8: Ricostruzione del segnale

Capitolo 3

Elaborazione di immagini

L'elaborazione delle immagini digitali è l'insieme dei metodi utilizzati principalmente in due aree di applicazione: migliorare le informazioni contenute nell'immagine e manipolare i dati in essa contenuti, che potrebbero dover essere rappresentati, trasferiti o memorizzati. L'analisi di un'immagine riguarda l'estrazione delle informazioni che essa contiene; per questo è necessario rappresentare matematicamente la distribuzione di luminosità dell'immagine. Come abbiamo già detto precedentemente un'immagine può essere vista come una funzione bidimensionale $f(x, y)$ e ad ogni coppia di coordinate corrisponde un certo valore della funzione che rappresenta l'intensità dell'immagine in quel punto. Quando x, y e i valori di f sono tutti finiti l'immagine è un'immagine digitale. Essa è formata da un numero finito di elementi, ognuno dei quali ha una posizione e un valore preciso, che vengono chiamati *pixel*.

In MATLAB la rappresentazione delle immagini viene fatta per mezzo di matrici, in cui ogni elemento è un pixel. Se consideriamo un'immagine in bianco e nero, questa viene rappresentata con un'unica matrice di valori interi o reali e il valore di questi elementi varia in una range di rappresentazione di grigi. Quando l'immagine è a colori si crea invece una variabile che è una matrice di dimensioni $M \times N \times 3$, le cui componenti coincidono con le componenti di rosso, verde e blu che costituiscono i 3 piani di colore **RGB**. Gli elementi delle 3 matrici prese singolarmente danno informazioni relative alla quantità di ciascuno dei 3 colori che deve essere mescolata per ottenere il corrispondente pixel a colori dell'immagine. Nel nostro caso l'elaborazione tramite trasformata wavelet è stata fatta sull'unica matrice che rappresenta l'immagine in scala di grigi; per elaborare un'immagine a colori si dovrebbe trasformare singolarmente ciascuna matrice e i risultati ottenuti dovrebbero essere riuniti insieme per ottenere il risultato finale.

3.1 Wavelet in due dimensioni

Il discorso sulla trasformata wavelet in una sola dimensione che abbiamo trattato nel capitolo precedente può essere esteso alle funzioni in due dimensioni come sono le immagini. In questo caso si devono considerare una funzione di scaling bidimensionale $\varphi(x, y)$ e 3 funzioni wavelets bidimensionali $\psi^H(x, y)$, $\psi^V(x, y)$ e $\psi^D(x, y)$, tutte prodotte da un'unica funzione di scaling φ e dalla corrispondente wavelet ψ :

$$\varphi(x, y) = \varphi(x)\varphi(y) \quad (3.1a)$$

$$\psi^H(x, y) = \psi(x)\varphi(y) \quad (3.1b)$$

$$\psi^V(x, y) = \varphi(x)\psi(y) \quad (3.1c)$$

$$\psi^D(x, y) = \psi(x)\psi(y). \quad (3.1d)$$

Le 3 wavelets misurano le variazioni della funzione che avvengono in 3 diverse direzioni: ψ^H riconosce i dettagli lungo la direzione orizzontale, ψ^V riconosce i dettagli verticali e ψ^D riconosce i dettagli diagonali. Le funzioni di base scalate e traslate sono

$$\varphi_{j,k_1,k_2}(x, y) = 2^{j/2}\varphi(2^jx - k_1, 2^jy - k_2), \quad (3.2a)$$

$$\psi_{j,k_1,k_2}^i(x, y) = 2^{j/2}\psi^i(2^jx - k_1, 2^jy - k_2), \quad (3.2b)$$

dove $i = \{H, V, D\}$ indica la wavelet che si sta considerando.

La trasformata wavelet discreta per una funzione $f(x, y)$ di dimensioni $M \times N$ è data da

$$W_\varphi(j_0, k_1, k_2) = c_{j_0,k_1,k_2} = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\varphi_{j_0,k_1,k_2}(x, y), \quad (3.3a)$$

$$W_\psi^i(j_0, k_1, k_2) = d_{j_0,k_1,k_2} = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\psi_{j_0,k_1,k_2}^i(x, y). \quad (3.3b)$$

Come nel caso monodimensionale, j_0 è un livello di risoluzione iniziale, scelto arbitrariamente. I coefficienti c_{j_0,k_1,k_2} definiscono l'approssimazione della funzione in ingresso al livello considerato mentre d_{j_0,k_1,k_2} aggiungono i dettagli orizzontali, verticali e diagonali dello stesso livello. Le serie dei coefficienti sono sempre ottenute tramite filtraggio e sottocampionamento. L'implementazione viene fatta tramite la trasformata veloce monodimensionale delle righe di $f(x, y)$, seguita dalla trasformata delle colonne.

Consideriamo $W_\varphi(j, k_1, k_2)$ come rappresentazione iniziale dell'immagine, filtriamo le sue righe con i due filtri, passa-basso H_0 e passa-alto H_1 , e sottocampioniamo le colonne. Otteniamo due sottoimmagini in cui la risoluzione

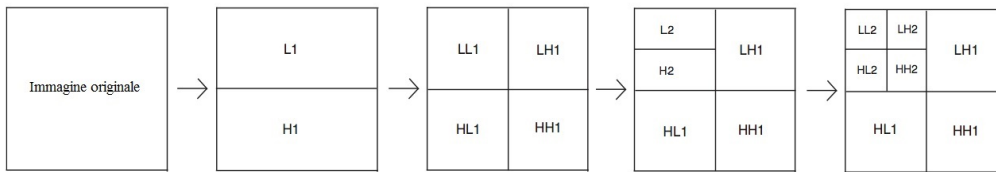


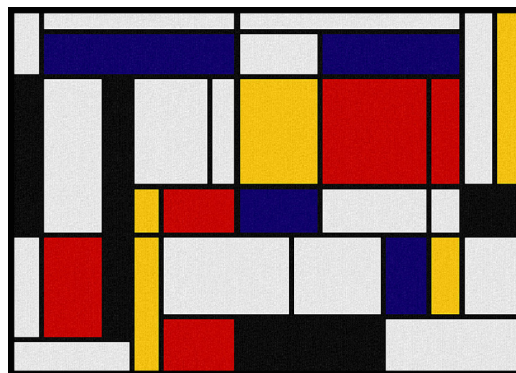
Figura 3.1: Decomposizione dell'immagine

orizzontale è dimezzata; la componente a bassa frequenza è l'approssimazione dell'originale e contiene le informazioni verticali a bassa frequenza, la componente ad alta frequenza contiene invece i dettagli verticali ad alta frequenza. Entrambe le sottoimmagini vengono poi filtrate lungo le colonne e sottocampionate e si ottengono quattro immagini le cui dimensioni sono un quarto dell'originale. Il processo può essere iterato diverse volte a partire dall'approssimazione dell'immagine ottenuta al livello precedente (Fig.3.1).

La trasformata wavelet è molto utilizzata nel campo di elaborazione delle immagini, perchè agendo solo sui coefficienti di particolari dettagli e particolare frequenze si possono eliminare rumore, individuare i bordi e comprimere efficacemente le immagini.

3.1.1 Decomposizione e ricostruzione di immagini

Consideriamo, come esempio su quanto appena detto riguardo la decomposizione, la seguente immagine



in cui è raffigurata una delle opere del pittore olandese Piet Mondrian, famoso per i suoi quadri che consistono in forme rettangolari di colori rosso, giallo, blu e nero. Possiamo vedere come la decomposizione livello per livello, effettuata utilizzando la wavelet di Haar, metta in evidenza i dettagli esattamente lungo

le direzioni delle linee del quadro in esame. Il primo livello di decomposizione è il seguente

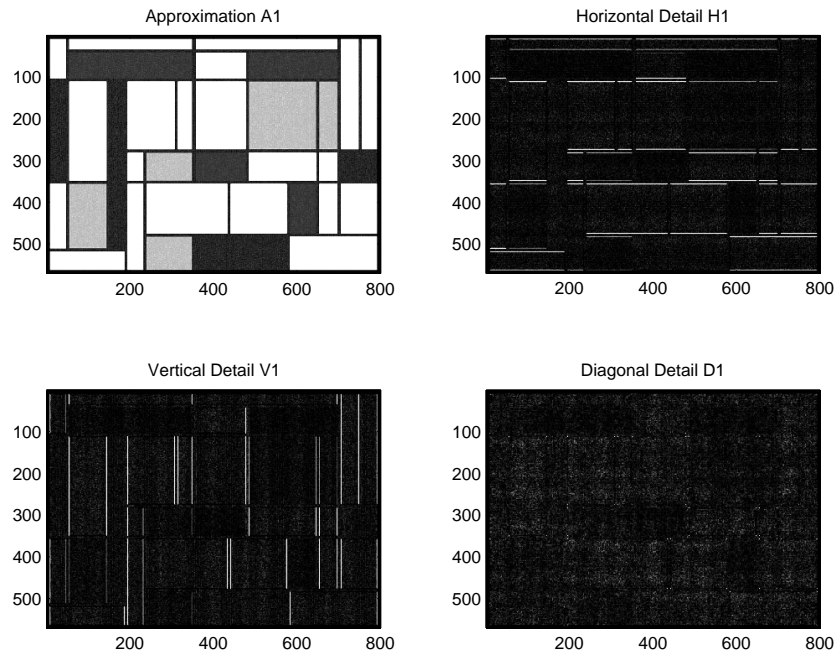


Figura 3.2: Decomposizione primo livello

Prima di essere analizzata l'immagine è stata modificata in modo da essere rappresentata secondo una scala di grigi. Dalla figura 3.2 vediamo chiaramente le linee orizzontali e verticali individuate dalle wavelet nelle rispettive direzioni, mentre non sono praticamente presenti dettagli diagonali. Se continuiamo la decomposizione, per esempio fino al livello 3, vediamo che sempre più dettagli vengono estratti dall'immagine approssimata, le cui dimensioni di volta in volta vengono ridotte di un quarto (Fig.3.3).

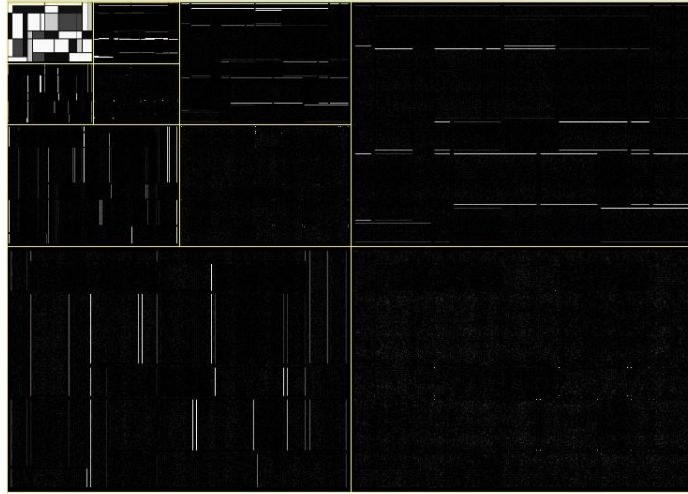


Figura 3.3: Decomposizione livello 3

Come per i segnali monodimensionali, anche per l'immagine è possibile stabilire il valore di una soglia per cui tutti i coefficienti il cui modulo è inferiore a tale soglia vengono annullati, e l'immagine viene ricostruita soltanto con i coefficienti mantenuti. Consideriamo diversi valori di soglia per vedere con quanti coefficienti si può ottenere un'immagine che sia abbastanza fedele all'originale. Le soglie sono calcolate rispetto al valore assoluto del massimo coefficiente, quindi nella tabella sottostante i valori inseriti nella colonna delle soglie devono in realtà essere moltiplicati per il valore del massimo coefficiente in modo da ottenere il vero valore della soglia.

soglia	coefficienti iniziali	coefficienti mantenuti	compressione	errore
0.1	456658	23	5.03659e-05	5.43e-01
0.01	456658	1016	0.00222486	2.05e-01
0.001	456658	7413	0.0162332	4.66e-02
0.0002	456658	34522	0.0755971	2.93e-02
0.0001	456658	116026	0.254076	1.64e-02
1e-05	456658	381493	0.835402	8.65e-04
1e-06	456658	422153	0.92444	1.56e-05
1e-07	456658	423446	0.927272	7.30e-07
1e-08	456658	423661	0.927742	1.85e-09

L'immagine ottenuta, per esempio, dopo aver eliminato i coefficienti minori della seconda soglia in tabella, è quella raffigurata in Fig.3.4 in cui la ricostruzione non è evidentemente molto buona.

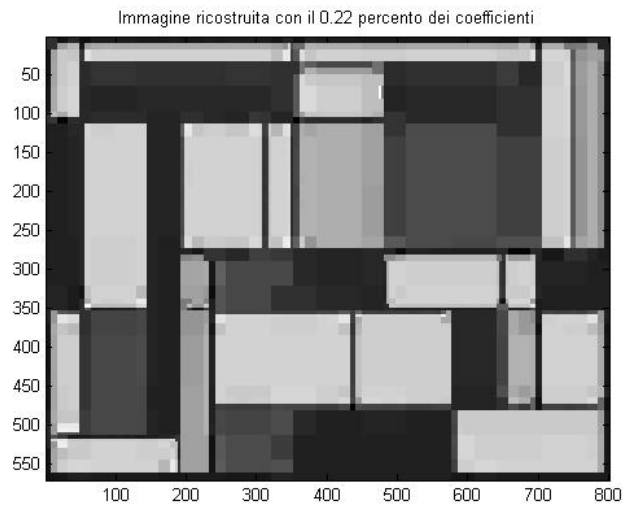


Figura 3.4: Ricostruzione immagine con 0,22% dei coefficienti

Se invece consideriamo l'immagine ottenuta dopo l'eliminazione dei coefficienti inferiori al valore di soglia inserito nella quarta riga della tabella, quindi un'immagine ricostruita con solo poco più del 7% dei coefficienti, le differenze con l'immagine originale sono praticamente indistinguibili per l'occhio umano (Fig.3.5). Da questa soglia in poi le ricostruzioni sono praticamente tutte uguali e indistinguibili dall'originale.

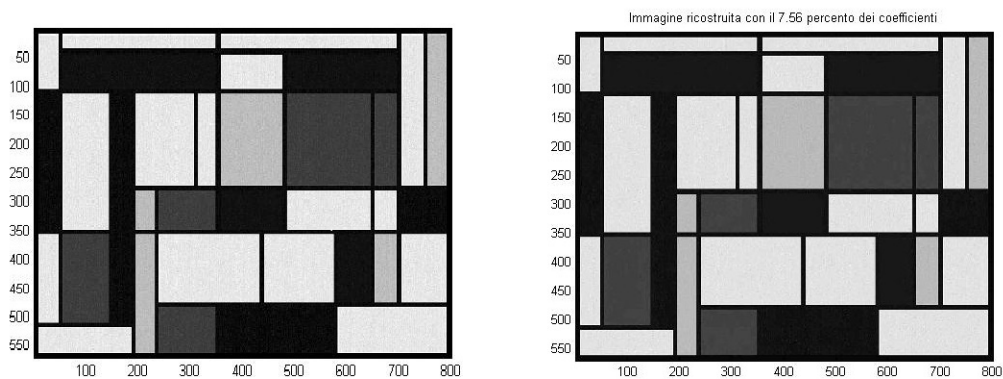


Figura 3.5: Immagine originale (sinistra) e immagine ricostruita (destra)

L'errore è stato calcolato nel modo seguente

$$\frac{\|F - X\|_F}{\|F\|_F}$$

dove F è l'immagine originale, X quella ricostruita e $\| \cdot \|_F$ indica la norma di *Frobenius*. Considerando una generica matrice \mathbf{A} , la norma di Frobenius è una norma matriciale definita come

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}.$$

3.2 Esperimento

Ci interessa adesso fare alcune prove di elaborazione sull'immagine in Fig.3.6 in cui è rappresentata una figura antropomorfa, chiamata "l'orante", incisa in una roccia ritrovata nella necropoli di *Sos Furrighesos*, ad Anela. Lo scopo dell'elaborazione è quello di ottenere un'immagine in cui la figura incisa sia l'unica visibile o in cui sia almeno più evidente rispetto all'originale.



Figura 3.6: Figura antropomorfa della necropoli di Anela

Vediamo nuovamente la decomposizione dell'immagine, questa volta tramite wavelet bior3.7. Il livello massimo di approssimazione a cui si può arrivare in questo caso, considerato la grandezza dell'immagine e la wavelet utilizzata, è l'ottavo; nell'immagine in Fig.3.7 mettiamo in evidenza soltanto i primi 4 livelli con le relative approssimazioni e i diversi dettagli orizzontali, verticali e diagonali.

Per rilevare i dettagli principali poniamo innanzitutto a zero tutti i coefficienti dell'approssimazione del livello massimo, inoltre dalle immagine si nota che il primo livello dei dettagli è formato da numerosi coefficienti di valore piuttosto piccolo, quindi anche questi possiamo azzerarli.

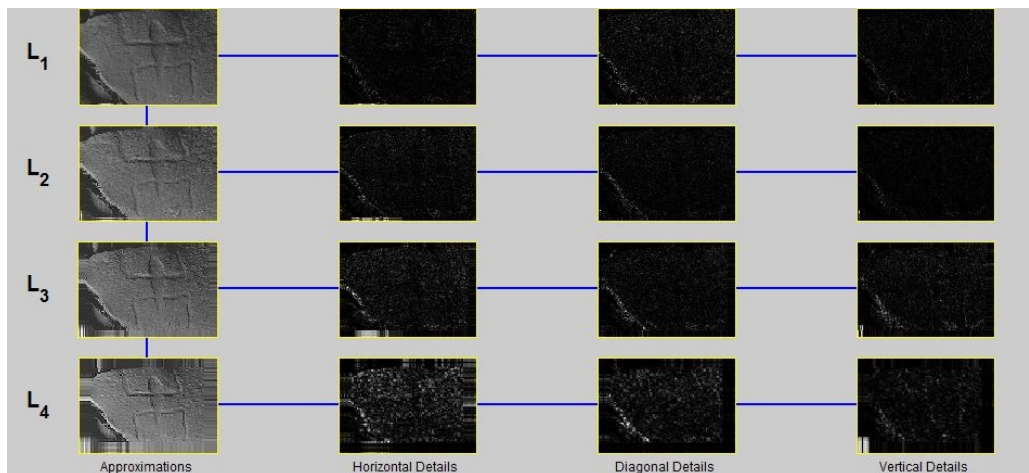


Figura 3.7: Primi 4 livelli di decomposizione

Il motivo per cui tenere solo i coefficienti alle frequenze più alte dovrebbe portare a un miglioramento nella visualizzazione della figura è dato dal fatto che il contorno della figura è dovuto allo spigolo dell'incisione, è quindi un'irregolarità della soluzione e una discontinuità della derivata prima. Le alte frequenze sono quelle che permettono di avere informazioni sulle irregolarità. Nella figura che vediamo di seguito l'immagine a sinistra è quella originale mentre quella di destra è ricostruita mantenendo pochissimi coefficienti dei dettagli ai livelli più bassi della decomposizione, sono presenti invece molti coefficienti dei livelli superiori e sono stati azzerati i coefficienti dell'approssimazione e del primo livello dei dettagli.

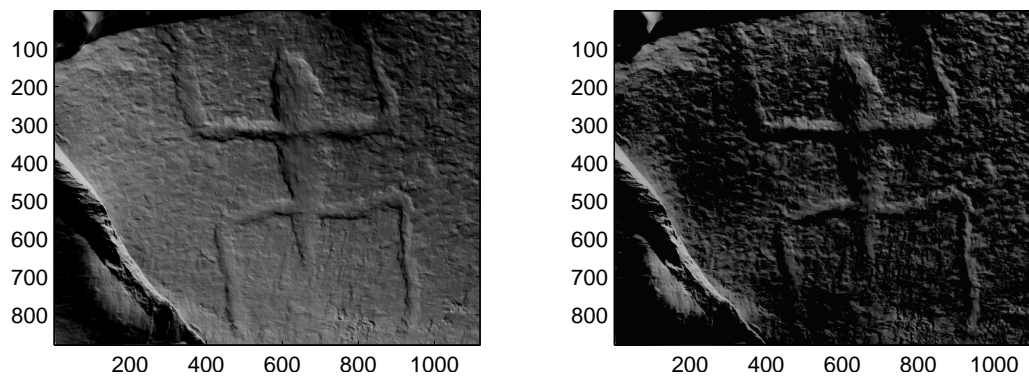


Figura 3.8:

Nell'immagine a destra la figura risulta leggermente più evidente rispetto a quella originale, ma non ci sono grandi differenze che permettano di dire che

l'immagine elaborata sia migliore della prima.

Si può provare ad effettuare la stessa operazione di decomposizione e selezione dei coefficienti sull'immagine che rappresenta il profilo 3D della figura. L'immagine è stata ottenuta tramite un metodo di ricostruzione di superfici 3D chiamato *photometric stereo*. In questo caso i pixel dell'immagine rappresentano non l'intensità della luce riflessa ma la quota del punto rispetto al piano di osservazione; quindi i punti più vicini vengono rappresentati con valore "1" e sono quelli più luminosi, mentre quelli più lontani vengono rappresentati con valore "0" e sono i punti più scuri presenti nell'immagine, otteniamo dunque informazioni spaziali sull'oggetto.

L'immagine di partenza è quella in Fig.3.9 che, anche in questo caso, è stata decomposta tramite wavelet bior3.7 ma il livello di approssimazione massimo è il sesto. L'immagine in Fig.3.10 è stata ricostruita dopo che sono stati azzerati i coefficienti dell'approssimazione e utilizzando solo i dettagli alle alte frequenze.

L'immagine ottenuta è migliore rispetto a quella originale, ed è migliore anche rispetto all'immagine analizzata precedentemente in Fig.3.8. Vediamo infatti che dal profilo 3D si ottiene un buon risultato perchè la figura incisa è più evidente e le funzioni wavelet sono state in grado di rilevare in modo abbastanza soddisfacente il contorno della figura rappresentata.

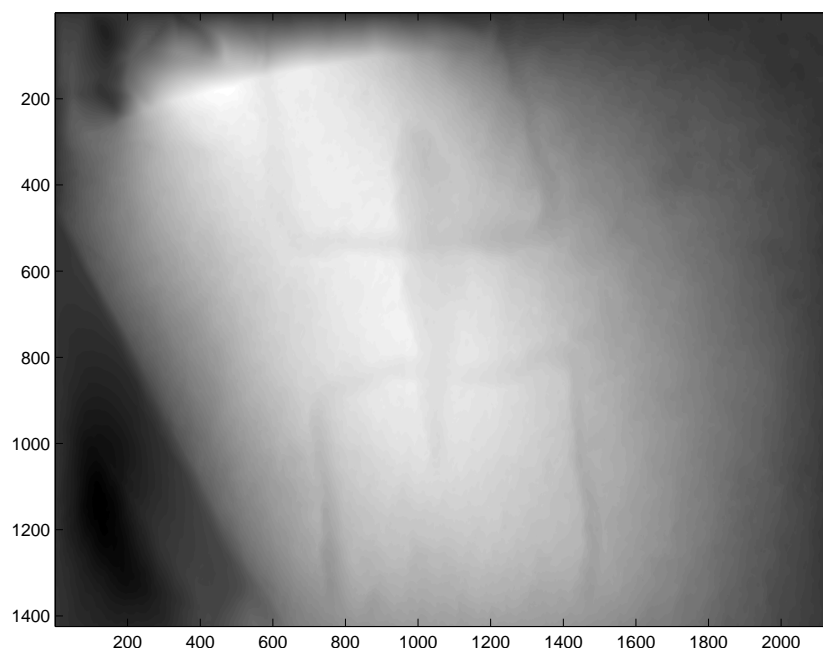


Figura 3.9: Profilo 3D della figura dell'orante

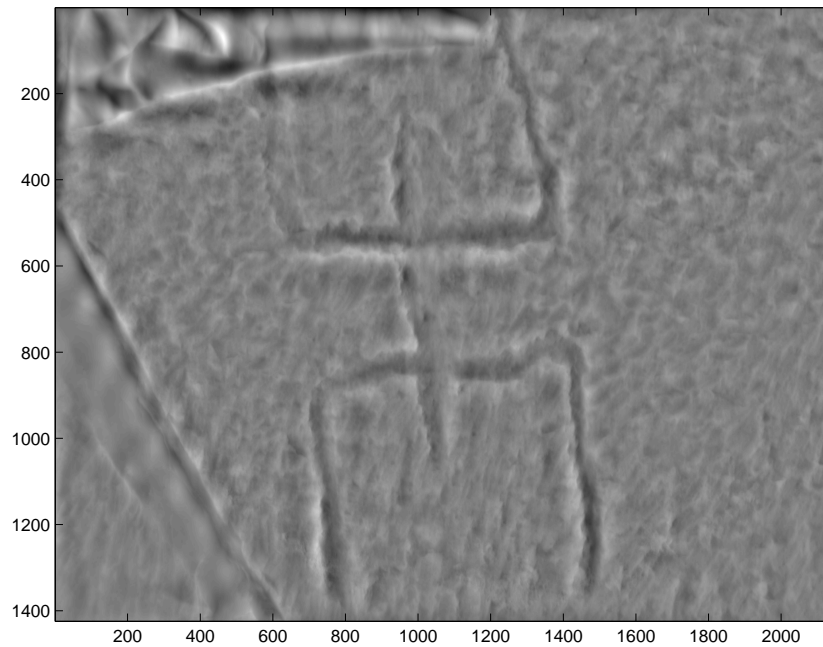


Figura 3.10: Immagine ricostruita

Nel Wavelet Toolbox di MATLAB esiste uno strumento grafico che permette la selezione dei coefficienti della trasformata che si vogliono annullare o mantenere. Dopo aver caricato l'immagine, si possono scegliere la wavelet che si vuole utilizzare e il livello di decomposizione a cui si vuole arrivare, e si ottengono le varie immagini relative alle approssimazioni e ai dettagli dei diversi livelli. A questo punto è possibile scegliere con quanti e quali coefficienti si vuole ricostruire l'immagine. La scelta dei coefficienti è fatta sempre rispetto ad un certo valore di soglia, quindi i coefficienti minori di tale valore vengono annullati, così come abbiamo già fatto in precedenza. Decomponendo l'immagine di partenza fino al quinto livello sempre per mezzo della wavelet bior3.7 e utilizzando i soli coefficienti dei dettagli si ottiene la seguente figura:



Anche in questo caso la ricostruzione con i soli coefficienti alle alte frequenze mette bene in evidenza i dettagli del bordo della figura. È dunque possibile che opportune tecniche di image processing consentano di isolare il profilo della figura in modo accurato.

Capitolo 4

Conclusioni

In questo lavoro di tesi si sono analizzati segnali e immagini per mezzo della trasformata wavelet discreta.

Nella prima parte ci siamo concentrati sull'analisi dei segnali monodimensionali, e in particolare ci interessava far vedere come in molti casi l'analisi wavelet risulti essere migliore e più efficace rispetto alla più utilizzata analisi di Fourier, soprattutto nel caso di segnali non stazionari e variabili come sono la maggior parte dei segnali reali.

Nella seconda parte abbiamo invece analizzato alcune immagini. Anche qui la trasformata wavelet si rivela uno strumento molto utile, in grado di garantire per esempio una buona ricostruzione di immagini utilizzando un ridotto numero di coefficienti, e in grado di rilevare importanti dettagli contenuti nell'immagine stessa. Si potrebbe continuare lo studio delle immagini, soprattutto archeologiche, per trovare il modo di identificare chiaramente soltanto le figure e i dettagli di interesse, come l'incisione che abbiamo analizzato nell'ultimo paragrafo di questa tesi.

Bibliografia

- [1] Chui,C.K. *An Introduction to Wavelets* (1992).
- [2] Chui,C.K. *Wavelets: A mathematical tool for signal analysis* (1997).
- [3] Rodriguez,G. and Seatzu,S. *Introduzione alla Matematica Applicata e Computazionale* (2010).
- [4] Briggs,W.L. and Henson,V.E. *The DFT: An owner's manual for the discrete Fourier transform* (1995).
- [5] Strang G. and Nguyen,T. *Wavelets and Filter Banks* (1996).
- [6] Gonzalez,R.C. and Woods,R.E. *Digital Image Processing* (2002).
- [7] Dessì,R., Mannu,C., Rodriguez,G., Tanda,G., Vanzi,M. *Recent improvements in photometric stereo for rock art 3D imaging. Digital Applications in Archaeology and Cultural Heritage* (2015).