



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA



DIPARTIMENTO  
DI INGEGNERIA  
DELL'INFORMAZIONE

TESI IN INGEGNERIA ELETTRONICA

# Compressione tramite trasformata wavelet e applicazioni nella segmentazione

CANDIDATO

**Filippo Baldini**

Student ID 2042756

RELATORE

**Prof. Marco Cagnazzo**

Università di Padova

ANNO ACCADEMICO  
2023/2024



*Ai miei genitori  
e agli amici*





## Sommario

La trasformata wavelet ha dimostrato di essere un potente strumento matematico versatile per l'analisi e la manipolazione delle immagini. Questo studio si concentra sull'applicazione della trasformata wavelet e sulla compressione d'immagini, con particolare attenzione al confronto tra i metodi di compressione JPEG e JPEG2000. Inizialmente, viene fornita una panoramica della teoria di base della trasformata wavelet, evidenziando le sue caratteristiche principali. Successivamente, si esamina la compressione a blocchi e come questa possa essere implementata con la trasformata wavelet, mettendo in evidenza le differenze di prestazioni tra JPEG e JPEG2000. Infine, l'attenzione si sposta alla segmentazione d'immagini. In particolare, si descrive come l'identificazione di un oggetto ben definito in un'immagine compressa dipenda dal livello di distorsione introdotta dalla compressione. Si osserva che, per il medesimo livello di distorsione, i formati JPEG e JPEG2000 offrono capacità di identificazione dell'oggetto comparabili.



# Indice

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Trasformata wavelet</b>                                      | <b>1</b>  |
| 1.1      | Trasformata Wavelet discreta . . . . .                          | 1         |
| 1.1.1    | Identificazione dei bordi . . . . .                             | 3         |
| 1.2      | Rappresentazione a multirisoluzione . . . . .                   | 8         |
| 1.2.1    | Fast wavelet transform . . . . .                                | 10        |
| 1.2.2    | Wavelet biortogonale . . . . .                                  | 13        |
| 1.3      | Trasformata wavelet a due dimensioni . . . . .                  | 15        |
| <b>2</b> | <b>JPEG 2000</b>  | <b>19</b> |
| 2.1      | Filtri utilizzati per la trasformata wavelet discreta . . . . . | 19        |
| 2.2      | Codifica a sotto banda . . . . .                                | 22        |
| 2.2.1    | Quantizzazione a blocchi . . . . .                              | 23        |
| 2.2.2    | Codifica con trasformata . . . . .                              | 24        |
| 2.3      | EZW . . . . .   | 26        |
| 2.4      | JPEG 2000 . . . . .   | 30        |
| 2.5      | Differenza di prestazioni tra JPEG e JPEG2000 . . . . .         | 32        |
| <b>3</b> | <b>Segmentazione</b>  | <b>37</b> |
| 3.1      | Segmentazione d'immagini compresse . . . . .                    | 38        |
| 3.2      | Segmentazione e PSNR . . . . .                                  | 40        |
|          | <b>Bibliografia</b>   | <b>45</b> |





# Trasformata wavelet

La trasformata wavelet è uno strumento matematico per l'analisi e l'elaborazione di segnali variabili nel tempo, risultando efficace in diversi campi di applicazione come in fisica (stati coerenti per gruppi  $ax + b$  in meccanica quantistica) e in ingegneria (compressione e denoising).

## 1.1 TRASFORMATA WAVELET DISCRETA

Consideriamo un segnale  $x : \mathbb{R} \rightarrow \mathbb{C}$  appartenente allo spazio  $L^2$  ovvero a energia finita. In generale tale segnale può essere rappresentato tramite una combinazione lineare di una famiglia di segnali scelta opportunamente [1]:

$$x(t) = \sum_n \sum_m T_{m,n} \psi_{m,n}(t)$$

I segnali  $\psi_{m,n}(t)$  sono definiti tramite un segnale originario  $\psi(t)$  chiamato wavelet madre.

La wavelet madre deve soddisfare le seguenti condizioni:

1.  $\psi$  deve essere ad energia finita:

$$E[\psi] = \int_{-\infty}^{\infty} |\psi(t)|^2 dt < +\infty$$

- 2.

$$\int_0^{\infty} \frac{|\Psi(f)|^2}{f} df < +\infty$$

## 1.1. TRASFORMATA WAVELET DISCRETA

Dove  $\Psi(f)$  è la trasformata di Fourier della wavelet madre, ovvero:

$$\Psi(f) = \int_{-\infty}^{\infty} \psi(t) e^{-j2\pi ft} dt$$

Ciò implica che  $\psi(t)$  sia a media nulla, affinché l'integrale converga,  $\Psi(0)$  deve essere uguale a zero.

3. Un'altra condizione per le wavelet complesse è che la trasformata di Fourier debba essere reale e avere supporto nel dominio della frequenza maggiore di zero. Tuttavia nel seguito non si vedranno wavelet complesse.

La famiglia di funzioni  $\psi_{a,b}$  con  $a \in \mathbb{R}^+$  e con  $b \in \mathbb{R}$  è definita partendo dalla wavelet madre nel seguente modo:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$$

Quindi  $b$  corrisponde ad una traslazione temporale e  $a$  corrisponde ad un cambiamento del fattore di scala. La scelta di  $b$  equivale a svolgere un'analisi in una determinata finestra temporale. La scelta di  $a$  cambia il fattore di scala e di conseguenza anche lo spettro del segnale  $\psi_{a,b}$ , in quanto:

$$\mathcal{F}[x(ct)] = \frac{1}{|c|} X\left(\frac{f}{c}\right)$$

La famiglia  $\psi_{a,b}$  deve essere una base discreta per come abbiamo scelto di rappresentare il segnale  $x$ . Bisogna scegliere dei valori discreti di  $a$  e  $b$ . In genere si opta per un cambiamento di scala logaritmico e una traslazione unitaria per rendere più semplice ed efficiente la discretizzazione. Quindi si definisce  $\psi_{n,m}$  nel seguente modo[1]:

$$\psi_{m,n} = \frac{1}{\sqrt{2^m}} \psi(2^{-m}t - n) = 2^{-m/2} \psi(2^{-m}t - n)$$

Con  $m, n \in \mathbb{Z}$ .

La scelta più semplice è d'imporre l'ortonormalità dei segnali  $\psi_{n,m}$ :

$$\langle \psi_{n,m}, \psi_{n',m'} \rangle = \int_{-\infty}^{\infty} \psi_{n,m} \overline{\psi_{n',m'}} dt = \begin{cases} 1 & m = m' \text{ e } n = n' \\ 0 & \text{altrimenti} \end{cases}$$

Adesso possiamo definire i coefficienti  $T_{m,n}$ , detti coefficienti della trasformata wavelet, come la proiezione del segnale sulla nuova base:

$$T_{m,n} = \langle x(t), \psi_{n,m} \rangle = \int_{-\infty}^{\infty} x(t) \overline{\psi_{n,m}} dt$$

In questo modo il segnale può essere espresso come combinazione lineare delle nuove basi.

$$x(t) = \sum_n \sum_m \langle x, \psi_{n,m} \rangle \psi_{m,n}(t) = \sum_n \sum_m T_{m,n} \psi_{m,n}(t)$$

La trasformata wavelet così definita consiste nella rappresentazione di un elemento dello spazio euclideo  $L^2(\mathbb{R})$  tramite una base ortogonale. Vale allora l'identità di Parseval, similmente a quanto si osserva per la trasformata di Fourier. Più precisamente abbiamo la conservazione dell'energia (norma quadratica):

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \sum_n \sum_m |T_{m,n}|^2$$

### 1.1.1 IDENTIFICAZIONE DEI BORDI

Un importante proprietà della trasformata wavelet è l'abilità d' identificare le discontinuità di un segnale [1]. Tale proprietà verrà illustrata tramite un esempio.

Supponiamo di avere un segnale gradino definito nel seguente modo:

$$1(t) = \begin{cases} 1 & \forall t \geq 0 \\ 0 & \forall t < 0 \end{cases}$$

Consideriamo come wavelet madre la funzione cappello messicano:

$$\psi_{0,0} = \frac{d^2}{dt^2} (-e^{-t^2}) = (2 - 4t^2) e^{-t^2}$$

In figura 1.1 sono illustrati i segnali.

## 1.1. TRASFORMATA WAVELET DISCRETA

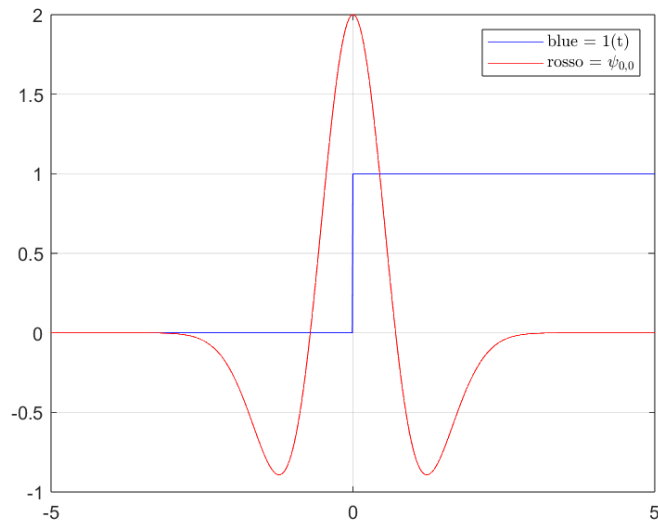


Figura 1.1: Funzione gradino e funzione madre

Si considera un arbitrario fattore di scala e analizziamo il comportamento  $T_{m,n}$  alla traslazione temporale:

1. Per valori di  $n$  molto minori di zero la wavelet si trova "lontano" della discontinuità.  $T_{m,n}$  è praticamente zero in quanto la wavelet è localizzata nel centro ed è praticamente nulla ad una breve distanza. In figura 1.2 è illustrato questo caso:

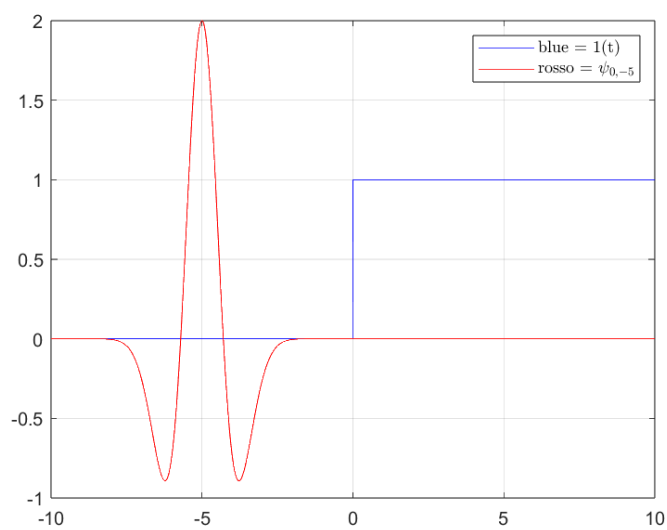


Figura 1.2:  $\psi_{n,m}$  con  $n = -5$  e  $m = 0$



2. In seguito la wavelet "incontra" la discontinuità. In questo caso il coefficiente è minimizzato, in quanto è presente solo il contributo negativo della wavelet. In figura 1.3 è illustrato questo caso:

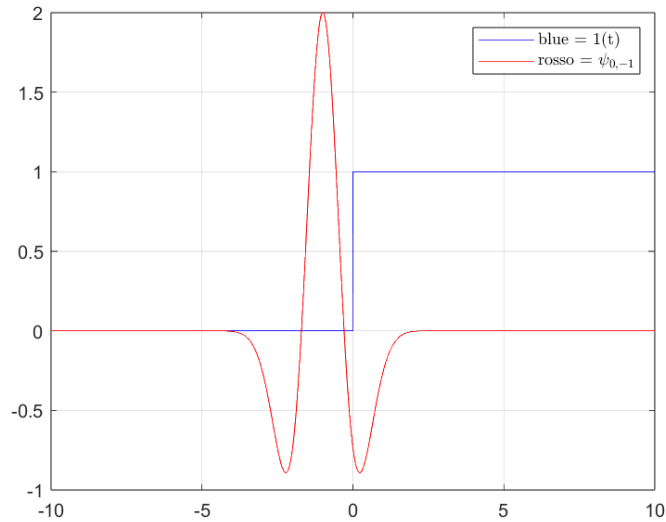


Figura 1.3:  $\psi_{n,m}$  con  $n = -1$  e  $m = 0$

3. Nel caso la wavelet sia centrata nella discontinuità,  $T_{m,0}$  è praticamente zero in quanto la parte negativa della wavelet viene compensata con la parte positiva. In figura 1.4 è illustrato questo caso:

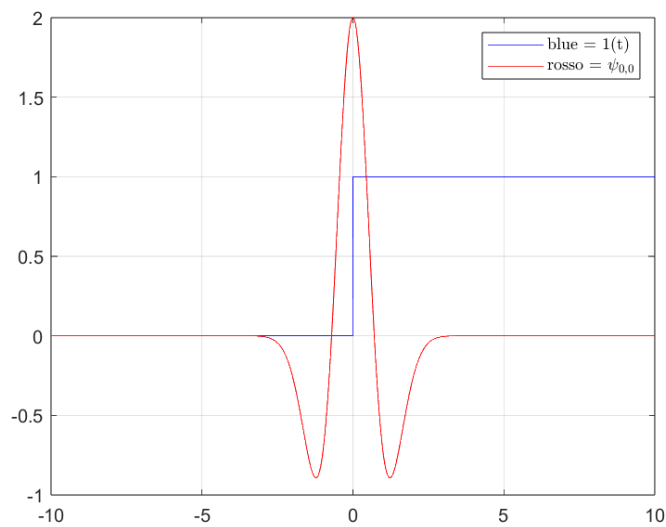


Figura 1.4:  $\psi_{n,m}$  con  $n = 0$  e  $m = 0$

## 1.1. TRASFORMATA WAVELET DISCRETA

4. In seguito la wavelet si "allontana" dalla discontinuità. In questo caso il coefficiente è massimizzato, in quanto una parte negativa è cancellata dal gradino. In figura 1.5 è illustrato questo caso:

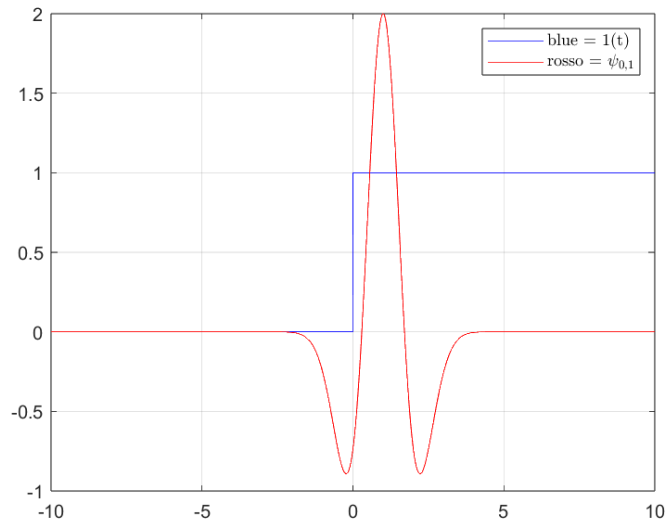


Figura 1.5:  $\psi_{n,m}$  con  $n = 1$  e  $m = 0$

5. In questo caso il coefficiente è praticamente zero, analogamente al primo. In figura 1.5 è illustrato questo caso:

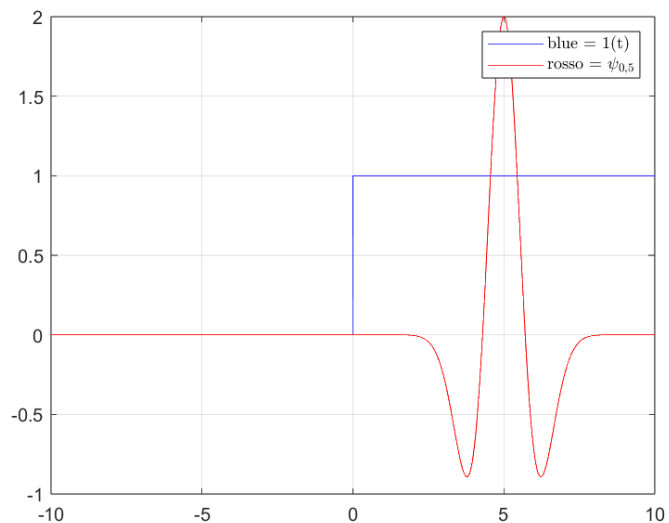


Figura 1.6:  $\psi_{n,m}$  con  $n = 5$  e  $m = 0$

In figura 1.7 sono illustrati i coefficienti della trasformata al variare di  $n$  ad un fattore di scala fissato.

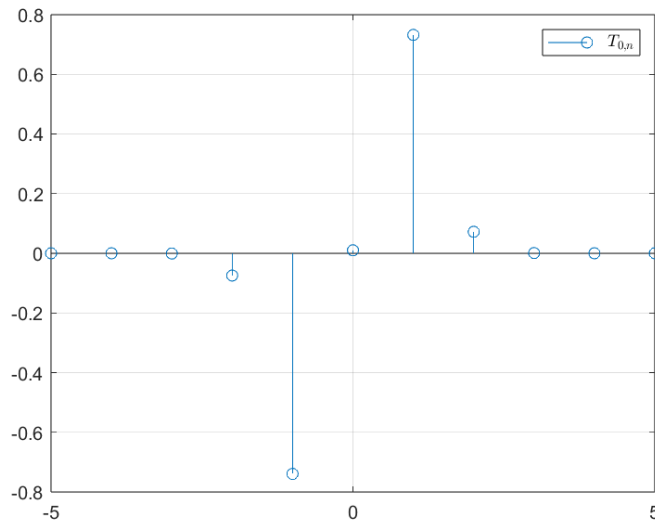


Figura 1.7: Coefficienti della trasformata

Come è possibile vedere dal grafico, in presenza di una discontinuità, i coefficienti  $T_{m,n}$  prima sono negativi e poi diventano positivi. I coefficienti  $T_{m,n}$  con ampiezza maggiore sono localizzati in prossimità del bordo, contenendo la maggior parte dell'informazione della discontinuità del segnale  $1(t)$ . Al contrario nella trasformata di Fourier, l'informazione associata alla discontinuità è contenuta in tutto lo spettro alle alte frequenze del segnale. Infatti la trasformata di Fourier del segnale gradino è uguale a:

$$\mathcal{F}[1(t)](f) = \frac{1}{2}\delta(f) + \frac{1}{j2\pi f}$$

Consideriamo la traslazione del gradino nel dominio della frequenza:

$$\mathcal{F}[1(t - t_0)](f) = e^{-j2\pi f t_0} \left( \frac{1}{2}\delta(f) + \frac{1}{j2\pi f} \right)$$

In questo caso l'informazione sulla localizzazione della discontinuità, ovvero  $t_0$ , è difficile da individuare. Infatti la traslazione temporale del segnale, nel dominio della frequenza è cambiamento di fase rispetto al segnale originale.

In conclusione nella trasformata wavelet è più facile localizzare la discontinuità e i coefficienti che contribuiscono maggiormente rispetto alla trasformata di Fourier.

## 1.2 RAPPRESENTAZIONE A MULTIRISOLUZIONE

In precedenza abbiamo visto come un segnale può essere rappresentato come combinazione lineare delle wavelet  $\psi_{n,m}$ , ma non è l'unica rappresentazione. Infatti un segnale può essere rappresentato tramite una sua versione a bassa risoluzione e un segnale che contiene il dettaglio del segnale. Ciò prende il nome di rappresentazione a multirisoluzione, ed è possibile implementarla tramite wavelet.

L'analisi a multirisoluzione consiste nell'utilizzare una successione di spazi vettoriali  $V_m$ , con  $m \in \mathbb{Z}$ , per approssimare  $L^2(\mathbb{R})$ . La successione di spazi deve essere tale da soddisfare le seguenti condizioni [4]:

$$\dots V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \dots$$

$$\bigcup_{j \in \mathbb{Z}} V_j = L^2 \quad \bigcap_{j \in \mathbb{Z}} V_j = \{0\}$$

Consideriamo  $x_m$  la proiezione del segnale  $x$  sullo spazio  $V_m$ , allora si ha:

$$\lim_{m \rightarrow -\infty} x_m = x$$

Al crescere di  $m$  si restringe l'insieme  $V_m$  e quindi la proiezione di  $x$  sullo spazio  $V_m$  si discosta maggiormente rispetto al segnale originario, risultando  $x_m$  una versione approssimata del segnale  $x$ .

Adesso introduciamo delle nuove condizioni per implementare gli aspetti di multirisoluzione:

$$x(t) \in V_m \Leftrightarrow x(2^m t) \in V_0$$

In questo modo tutti gli spazi  $V_m$  sono versioni scalate dello spazio  $V_0$ .

Introduciamo l'invarianza rispetto alla traslazione temporale:

$$x(t) \in V_0 \Leftrightarrow x(t - n) \in V_0 \quad \forall n \in \mathbb{Z}$$

Definiamo i segnali  $\phi$  appartenenti allo spazio  $V_0$  come:

$$\{\phi_{0,n}; \quad n \in \mathbb{Z}\} \quad \text{base ortonormale di } V_0$$

Deduciamo dalle condizioni iniziali:

$$\phi_{m,n} = 2^{-m/2} \phi(2^{-m}t - n) \quad \text{base ortonormale di } V_m$$

Le funzioni  $\phi_{m,n}$  prendono il nome di funzioni di scala e la seguente proprietà:

$$\int_{-\infty}^{\infty} \phi_{0,0}(t) dt = 1$$

$\phi_{0,0}$  è chiamata wavelet padre. Le funzioni di scala sono ortonormali alla traslazione per costruzione ma non al cambiamento di scala. Consideriamo  $x_m$ , la proiezione di un generico segnale  $x$  di  $L^2$  nello spazio  $V_m$  generato da  $\phi_{m,n}$ . Siccome al variare di  $n$   $\phi_{n,m}$  è una base ortonormale di  $V_m$ , abbiamo:

$$x_m(t) = \sum_n S_{m,n} \phi_{m,n}(t)$$

Dove:

$$S_{m,n} = \int_{-\infty}^{\infty} x(t) \phi_{m,n} dt$$

I segnali  $x_m$  sono fortemente ridondanti perché gli spazi  $V_m$ , per costruzione, hanno una relazione d' inclusione tra loro.

Supponiamo che un segnale appartenga a  $V_{m_0-1}$ :

$$x(t) \in V_{m_0-1} \implies x_{m_0-1} = x(t)$$

Quindi:

$$x(t) = x_{m_0} + (x_{m_0-1} - x_{m_0}) = x_{m_0} + D_{m_0}$$

Dove  $D_{m_0}$ :

$$\begin{aligned} D_{m_0} &= x_{m_0-1} - x_{m_0} = \sum_n S_{m_0-1,n} \phi_{m_0-1,n}(t) - \sum_n S_{m_0,n} \phi_{m_0,n}(t) = \\ &= \sum_n S_{m_0-1,n} 2^{-(m_0-1)/2} \phi(2^{-m_0+1}t - n) - \sum_n S_{m_0,n} 2^{-m_0/2} \phi(2^{-m_0}t - n) = \\ &= \sum_n T_{m_0,n} \psi_{m_0,n}(t) \end{aligned}$$

## 1.2. RAPPRESENTAZIONE A MULTIRISOLUZIONE

Tralasciando la dimostrazione, in questo passaggio si afferma che la differenza tra le rappresentazioni di un segnale con due gradi di risoluzione differenti,  $m_0$  e  $m_0 - 1$  può essere rappresentato attraverso una famiglia di wavelet  $\psi_{m_0,n}$  tutte con lo stesso fattore di scala del segnale con risoluzione minore.

Un segnale appartenente a uno spazio  $V_{m_0-1}$  può essere rappresentato a una risoluzione minore.

$$x_{m_0-1} = x_{m_0} + D_{m_0} = \sum_n S_{m_0,n} \phi_{m_0,n}(t) + \sum_n T_{m_0,n} \psi_{m_0,n}(t)$$

In genere un segnale appartiene a  $L^2$  quindi a  $V_{-\infty}$ , possiamo fare come segue:

$$\begin{aligned} x_{m_0-j} &= x_{m_0-j+1} + D_{m_0-j+1} = x_{m_0-j+2} + D_{m_0-j+2} + D_{m_0-j+1} = x_{m_0} + \sum_{i=1}^j D_{m_0-j+i} = \\ &= x_{m_0} + \sum_{m=m_0-j+1}^{m_0} D_m = x_{m_0} + \sum_{m=m_0-j+1}^{m_0} \sum_n T_{m,n} \psi_{m,n}(t) \end{aligned}$$

Facendo tendere  $j$  a meno infinito otteniamo:

$$x(t) = \sum_{n=-\infty}^{\infty} S_{m_0,n} \phi_{m_0,n}(t) + \sum_{m=-\infty}^{m_0} \sum_{n=-\infty}^{\infty} T_{m,n} \psi_{m,n}(t) = x_{m_0} + d_{m_0}$$

Possiamo vedere che il segnale originario può essere visto come il segnale approssimato  $x_{m_0}$  e una parte che rappresenta il segnale nel dettaglio ovvero  $d_{m_0}$ .

### 1.2.1 FAST WAVELET TRANSFORM

Come nel caso della trasformata di Fourier, esiste un algoritmo per il calcolo dei coefficienti che evita l'uso diretto della definizione. Tale algoritmo prende il nome di fast wavelet trasform [1][3].

In genere la funzione  $\phi$  può essere espressa come combinazione della funzione  $\phi$  ad una risoluzione maggiore:

$$\phi(t) = \sum_k c_k \phi(2t - k)$$

Dove i valori  $c_k$  devono soddisfare la seguente condizione:

$$\sum_k c_k = 2$$

In aggiunta per creare un sistema ortogonale:

$$\sum_k c_k c_{k+2k'} = \begin{cases} 2 & \text{se } k' = 0 \\ 0 & \text{altrimenti} \end{cases}$$

Con gli stessi coefficienti si possono rappresentare le wavelet:

$$\psi(t) = \sum_k (-1)^k c_{1-k} \phi(2t - k) = \sum_k b_k \phi(2t - k)$$

Esplicitiamo la rappresentazione di  $\phi_{m+1,n}$  attraverso la combinazione lineare delle funzioni a risoluzione maggiore  $\phi_{m,n}$ :

$$\phi_{m+1,n}(t) = 2^{-(m+1)/2} \phi\left(\frac{t}{2^{m+1}} - n\right) = 2^{-m/2} 2^{-1/2} \sum_k c_k \phi\left(\frac{t}{2^m} - 2n - k\right)$$

Quindi:

$$\phi_{m+1,n}(t) = \frac{1}{\sqrt{2}} \sum_k c_k \phi_{m,2n+k}(t)$$

Detta equazione di dilatazione.

Analogamente

$$\psi_{m+1,n}(t) = \frac{1}{\sqrt{2}} \sum_k b_k \phi_{m,2n+k}(t)$$

Detta equazione wavelet.

Adesso vediamo come ottenere i coefficienti della rappresentazione multidi-

## 1.2. RAPPRESENTAZIONE A MULTIRISOLUZIONE

mensionale  $m + 1$  partendo dai coefficienti  $m$ :

$$\begin{aligned} S_{m+1,n} &= \int_{-\infty}^{\infty} x(t) \phi_{m+1,n}(t) dt = \int_{-\infty}^{\infty} x(t) \left[ \frac{1}{\sqrt{2}} \sum_k c_k \phi_{m,2n+k}(t) \right] (t) dt = \\ &= \frac{1}{\sqrt{2}} \sum_k c_k \left[ \int_{-\infty}^{\infty} x(t) \phi_{m,2n+k}(t) dt \right] = \frac{1}{\sqrt{2}} \sum_k c_k S_{m,2n+k} = \\ &= \frac{1}{\sqrt{2}} \sum_k c_{k-2n} S_{m,k} \end{aligned}$$

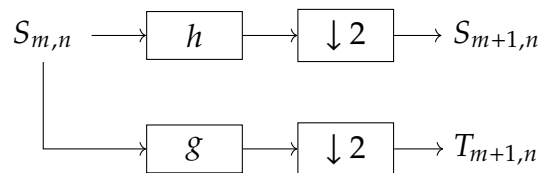
Il calcolo del coefficiente  $S_{m+1,n}$  equivale a fare una convoluzione con filtro aventi valori  $c_k$  e campionarlo con un fattore due. Notare che esso è un filtro passa-basso in quanto la sommatoria dei  $c_k$  è diverso da zero e quindi è un filtro a media non nulla.

Analogamente per  $T_{m+1,n}$  si ottiene:

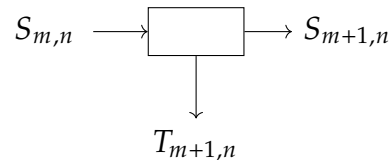
$$T_{m+1,n} = \frac{1}{\sqrt{2}} \sum_k b_{k-2n} S_{m,k}$$

Come prima il calcolo del coefficiente  $S_{m+1,n}$  equivale a fare una convoluzione con filtro aventi valori  $b_k$  e campionarlo con un fattore due. Notare che esso è un filtro passa-alto per costruzione dei coefficienti  $b_k$ .

Quindi il calcolo dei coefficienti della rappresentazione a multirisoluzione di grado  $m + 1$  partendo dai coefficienti con grado di risoluzione  $m$  equivale al seguente diagramma a blocchi.



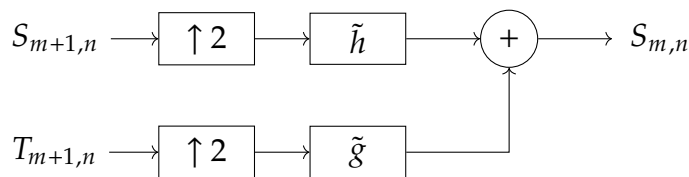
Questo è equivalente al seguente sistema:



Quindi mettendo in cascata tale sistema  $k$  volte si ottengono i coefficienti  $S_{m+k,n}$  e i coefficienti  $T_{m+i,n}$  con  $i \in \{1, \dots, k\}$ .



In modo analogo si dimostra che si possono ottenere i coefficienti ad una risoluzione minore partendo dai coefficienti a risoluzione maggiore. In seguito il diagramma a blocchi:



Notare che nel caso di applicazioni digitali i segnali non sono continui ma discreti. In questo caso il segnale a massima risoluzione può essere ottenuto come combinazione lineare di uno spazio  $V_0$ , in quanto un segnale discreto può essere visto come approssimazione di un arbitrario segnale continuo.

$$S_{0,n} = x(nT)$$

## 1.2.2 WAVELET BIORTOGONALE

Nella compressione delle immagini è necessario l'utilizzo di filtri simmetrici per evitare le discontinuità spurie introdotte dai bordi dell'immagine. Le uniche wavelet ortogonali simmetriche sono le wavelet di Haar. In figura 1.8 sono illustrate le wavelet padre e madre delle Haar.

## 1.2. RAPPRESENTAZIONE A MULTIRISOLUZIONE

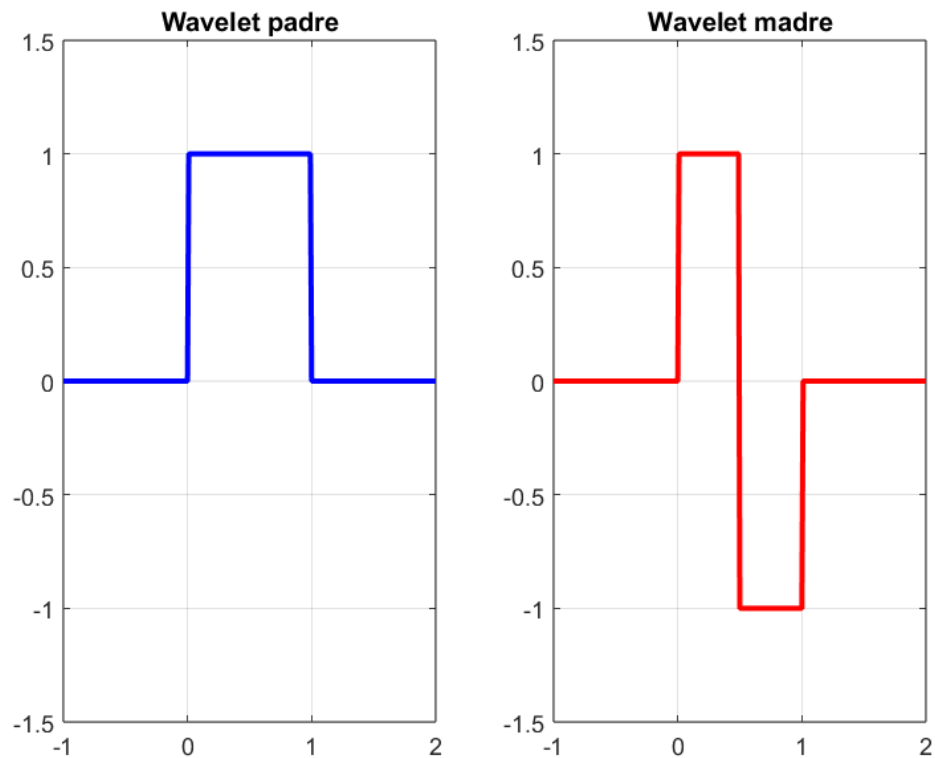


Figura 1.8: wavelet Haar

Tuttavia i filtri di Haar hanno solo un momento evanescente. Con momento evanescente si intende la capacità di un filtro di rappresentare un polinomio di grado inferiore al numero di momenti evanescenti. Di conseguenza i filtri Haar possono rappresentare segnali costanti mantenendo l'informazione del segnale originale.

Per ottenere delle wavelet simmetriche e con un numero arbitrariamente alto di momenti evanescenti bisogna lasciar cadere il vincolo di base ortogonale. Si possono invece considerare delle wavelet che soddisfino un vincolo più debole, e cioè [1]:

$$\int_{-\infty}^{\infty} \psi_{m,n}(t) \tilde{\psi}_{m',n'}(t) dt = \begin{cases} 1 & m = m' \text{ e } n = n' \\ 0 & \text{altrimenti} \end{cases}$$

Quindi possiamo rappresentare un segnale  $x$  nel seguente modo:

$$x(t) = \sum_n \sum_m T_{m,n} \tilde{\psi}_{m,n}(t)$$

$$T_{m,n} = \int_{-\infty}^{\infty} x(t) \psi_{m,n}(t) dt$$

Notare che in questo caso le funzioni  $\psi_{m,n}$  non sono ortonormali, ma sono semplicemente indipendenti. Questo equivalente alla proiezione su una base non ortogonale e quindi l'energia non si conserva.

Un esempio di wavelet biortogonali simmetriche con quattro momenti evanescenti sono le wavelet di Daubechies 9/7. Il caso di quattro momenti evanescenti è d'interesse in quanto si vede sperimentalmente che ben rappresentano le immagini.

### 1.3 TRASFORMATA WAVELET A DUE DIMENSIONI

La trasformata wavelet può essere generalizzata a due dimensioni. Si possono definire che la funzione padre e la funzione madre siano separabili lungo le colonne e le righe del segnale[1]:

|                         |   |
|-------------------------|---|
| Funzione di scala 2-D   | $\phi(t_1, t_2) = \phi(t_1)\phi(t_2)$   |
| wavelet orizzontale 2-D | $\psi^h(t_1, t_2) = \phi(t_1)\psi(t_2)$ |
| wavelet verticale 2-D   | $\psi^v(t_1, t_2) = \psi(t_1)\phi(t_2)$ |
| wavelet diagonale 2-D   | $\psi^d(t_1, t_2) = \psi(t_1)\psi(t_2)$ |

In figura 1.9 è illustrato lo schema a blocchi per la decomposizione wavelet bidimensionale:

### 1.3. TRASFORMATA WAVELET A DUE DIMENSIONI

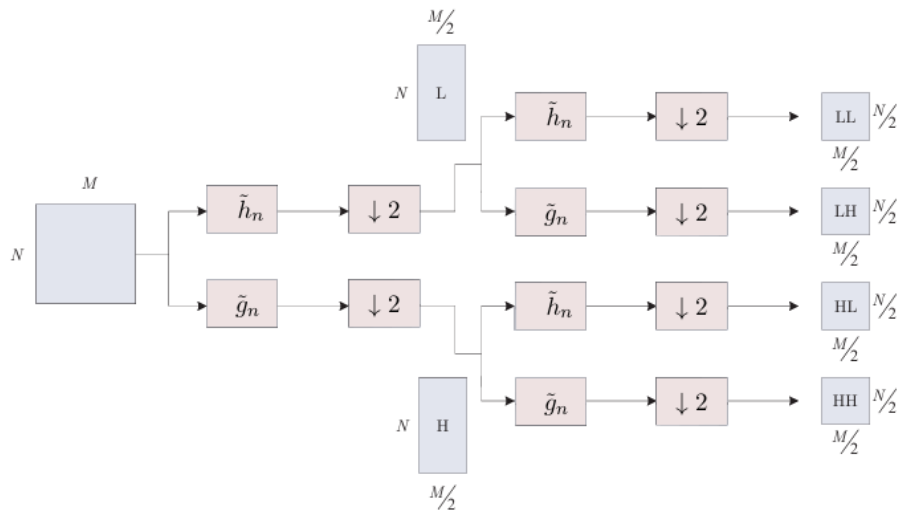


Figura 1.9: Schema a blocchi wavelet 2D

La trasformata viene applicata prima sulle righe determinando nel ramo superiore i coefficienti ottenuti dal filtraggio passa-basso L e in quello inferiore i coefficienti ottenuti dal filtraggio passa-alto H. Dopo si applica la stessa procedura per le righe ottenendo una matrice composta in quattro bande:

- LL: dove è contenuta l'immagine a bassa risoluzione;
- HL: dove sono evidenziate le discontinuità lungo le righe, ovvero i contorni verticali degli oggetti;
- LH: dove sono evidenziate le discontinuità lungo le colonne, ovvero i contorni orizzontali degli oggetti;
- HH: dove sono evidenziate le discontinuità lungo le diagonali;

Nella figura 1.12 è illustrata la decomposizione wavelet bidimensionale attraverso lo schema della figura 1.11 dell'immagine 1.10.



Figura 1.10: Immagine di input

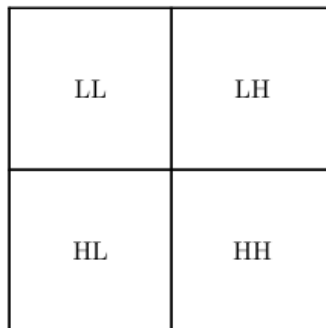


Figura 1.11: Matrice a quattro bande

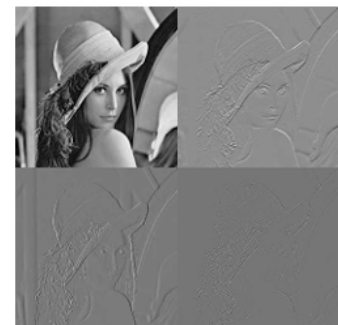


Figura 1.12: Decomposizione immagine

È possibile ripetere più volte l'analisi a multirisoluzione di figura 1.9 per un'immagine. Inizialmente si applica il sistema di figura 1.9 all'immagine originale ottenendo l'immagine a bassa risoluzione e le immagini che rappresentano il dettaglio. Poi si applica nuovamente l'analisi a multirisoluzione all'immagine a bassa risoluzione ottenuta precedentemente, ottenendo una versione a bassa risoluzione dell'immagine a bassa risoluzione e le immagini associate al dettaglio dell'immagine a bassa risoluzione. Adesso è possibile ripetere la procedura alla nuova immagine a bassa risoluzione ottenuta. Questo caso prende il nome di trasformata wavelet a  $k$  livelli, dove  $k$  è il numero di schemi a blocchi messi in cascata [13]. In figura 1.15 è illustrato un esempio di trasformata wavelet a due livelli.



Figura 1.13: Immagine di input

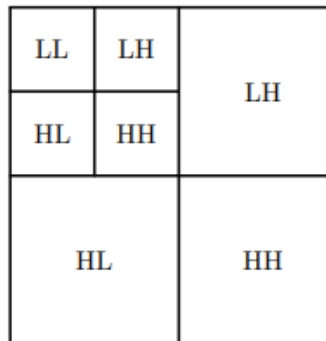


Figura 1.14: Matrice a otto bande

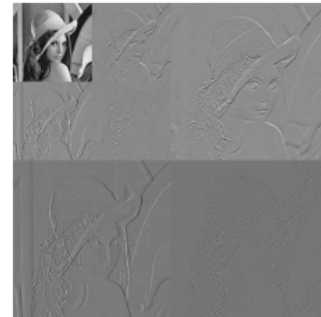


Figura 1.15: Decomposizione immagine

In questo modo la maggior parte dell'informazione dell'immagine di partenza è contenuta nella banda LL. Il segnale di partenza (l'immagine) è non sparso, nel senso che tutti i campioni (valori di luminosità) sono non trascurabili: infatti non si può dire a priori che in una certa zona dell'immagine non ci sono valori "importanti" mentre altrove sì. Invece, dopo la trasformata, è proprio così: nella banda di approssimazione LL si trovano i valori più grandi (in valore assoluto), mentre la maggior parte degli altri coefficienti wavelet è prossima a zero (colore grigio nella figura), e quindi, anche se viene trascurata (tramite compressione), la degradazione dell'informazione è piccola.





## JPEG 2000

Il comitato JPEG nel 2001 ha rilasciato un suo standard di codifica delle immagini che funge da successore allo standard originale JPEG introdotto nel 1992. Questo standard implementa un metodo di compressione delle immagini basato sulla trasformata wavelet invece di basarsi sulla trasformata discreta del coseno DCT.

Nel 1992 la trasformata wavelet era una nuova tecnologia, quindi si è preferito basarsi sulle tecniche basate sulla DCT che erano ben consolidate.

Inizialmente mancava un metodo efficace di compressione basato sulla trasformata wavelet discreta. La prima tecnica di codifica che ha mostrato prestazioni superiori rispetto a JPEG è stata l'Embedded Zero-Tree Wavelet Coding (EZW). Successivamente, è stato introdotto JPEG2000 che ha ulteriormente migliorato la qualità delle immagini compresse.

### **2.1** FILTRI UTILIZZATI PER LA TRASFORMATA WAVELET DISCRETA

Consideriamo per semplicità il caso monodimensionale[13]. Infatti la trasformata wavelet 1-D può essere facilmente estesa a una trasformata wavelet bidimensionale (2-D) utilizzando filtri wavelet separabili come visto 1.3.

Inizialmente le trasformate wavelet erano implementate utilizzando wavelet ortogonali . In questo caso l'energia si conserva come visto in precedenza dove

## 2.1. FILTRI UTILIZZATI PER LA TRASFORMATA WAVELET DISCRETA

$w(l)$  sono tutti i coefficienti della trasformata wavelet:

$$\sum_{n=0}^{N-1} x^2(n) = \sum_{l=0}^{L-1} w^2(l)$$

Questa proprietà è conveniente in quanto la distorsione quadratica media introdotta dalla quantizzazione dei coefficienti trasformati è uguale alla distorsione media quadratica introdotta nel segnale ricostruito, semplificando la progettazione del codificatore.

La trasformata wavelet ortogonale presenta alcune limitazioni che la rendono meno ideale per l'uso in un sistema di codifica.

In primo luogo il numero di coefficienti di input non è uguale al numero totale di coefficienti wavelet. Un modo semplice per eliminare l'espansione dei coefficienti è utilizzare la convoluzione circolare, anziché la convoluzione lineare, sull'input di lunghezza finita  $x(n)$ . In questo modo l'uscita del filtro sarà uguale al supporto massimo tra  $x(n)$  e  $h(n)$ .

Tuttavia considerare il segnale periodico può introdurre una grande componente in alta frequenza dovuto alla discontinuità tra  $x(0)$  e  $x(N - 1)$ , ovvero la discontinuità tra i lati opposti dell'immagine. Questi grandi coefficienti wavelet dovuti alle discontinuità ai bordi sono indesiderati perché richiedono più bit per la codifica. Gli artefatti ai bordi possono essere eliminati eseguendo un'estensione periodica simmetrica dell'input, invece di un'estensione periodica. Vale a dire che al posto di rendere periodico il segnale in sé, si rende periodico il segnale con in coda il segnale simmetrico. In questo modo si eliminano gli effetti di bordo indesiderati. Questo raddoppia le componenti in input ma non è problema in quanto i punti sono ridondanti per simmetria.

Il problema di questa scelta è che esistono un solo insieme di filtri a fase lineare (simmetrici o antisimmetrici) wavelet reali a supporto compatto ovvero i filtri Haar triviali.

L'assenza di filtri a fase lineare nelle wavelet ortogonali ha portato all'utilizzo delle wavelet biortogonali. In questo caso l'utilizzo delle wavelet biortogonali si discosta di poco dall'ortogonalità e si vede sperimentalmente che hanno prestazioni migliori. Notare che in questo caso la distorsione quadratica media introdotta dalla quantizzazione dei coefficienti trasformati non è uguale alla distorsione quadratica media introdotta nel segnale ricostruito. Due esempi comuni sono di filtri biortogonali simmetrici sono i filtri di Daubechies 9/7 e 5/3



i cui coefficienti sono riportati nella tabella 2.1.

| filtri 9/7 |           | filtri 5/3 |          | indice<br>filtri |
|------------|-----------|------------|----------|------------------|
| $h$        | $g$       | $h$        | $g$      |                  |
| 0.852699   | 0.788486  | 1.060660   | 0.707107 | 0                |
| 0.377402   | 0.418092  | 0.353553   | 0.353553 | -1, 1            |
| -0.110624  | -0.040689 | -0.176777  |          | -2, 2            |
| -0.023849  | -0.064539 |            |          | -3, 3            |
| 0.037828   |           |            |          | -4, 4            |

Tabella 2.1: Coefficienti filtri 9/7 e 5/3

Il nome deriva dalla matematica belga Ingrid Daubechies che per prima ha risolto il problema d'individuare i filtri simmetrici che, per un dato numero di momenti evanescenti, avessero il supporto minimo. I filtri 9/7 (con riferimento al supporto di  $h$  e  $g$ ) hanno 4 momenti nulli e sono, tra tutti quelle simmetrici e con 4 momenti nulli, quelli di supporto minimo. Analogamente per i 5/3 che hanno 2 momenti nulli.

Un metodo per verificare che questi filtri sono quasi ortogonali è applicare la condizione di ortogonalità dei filtri. Un altro modo è quello di utilizzare la seguente relazione che lega la varianza dell'intera immagine decomposta con le varianze delle sotto bande[13]:

$$\sigma^2 = \frac{1}{2^K} \sigma_{L0}^2 + \sum_{i=1}^K \frac{1}{2^i} w_{ij} \sigma_{i1}^2$$

Dove  $K$  sono il numero di livelli dei filtri e  $\sigma_{ij}^2$  sono le varianze associate alle varie sotto bande. Nel caso di filtri ortogonali i coefficienti  $w_{ij}$  sono tutti uguali a uno indicando che l'energia è conservata tra i coefficienti e l'immagine ricostruita. Nella tabella sono riportati i coefficienti  $w_{ij}$  in base alla scelta dei filtri.

## 2.2. CODIFICA A SOTTO BANDA

| <b>coefficienti</b> | <b>ortogonali</b> | 9/7     | 5/3     |
|---------------------|-------------------|---------|---------|
| $w_{10}$            | 1.00000           | 0.98295 | 0.75000 |
| $w_{11}$            | 1.00000           | 1.04043 | 1.43750 |
| $w_{20}$            | 1.00000           | 1.03060 | 0.68750 |
| $w_{21}$            | 1.00000           | 0.96721 | 0.92187 |
| $w_{30}$            | 1.00000           | 1.05209 | 0.67187 |
| $w_{31}$            | 1.00000           | 1.03963 | 0.79297 |
| $w_{40}$            | 1.00000           | 1.03963 | 0.66797 |
| $w_{41}$            | 1.00000           | 1.07512 | 0.76074 |

Tabella 2.2: Coefficienti  $w_{ij}$  in base ai filtri scelti

Si nota come i filtri 9/7 si avvicinano molto a dei filtri ortogonali. Di conseguenza la distorsione quadratica media introdotta dalla quantizzazione dei coefficienti trasformati è molto simile alla distorsione quadratica media introdotta nel segnale ricostruito.

Invece i filtri 5/3 si discostano maggiormente a filtri ortogonali, ma hanno il vantaggio che sono delle frazioni con denominatore  $2^k$ , a meno di un fattore di normalizzazione  $\sqrt{2}$ . Questo significa che i coefficienti della trasformata wavelet discreta possono essere calcolati senza errori di precisione quando l'ingresso è un segnale a valori interi.

## **2.2** CODIFICA A SOTTO BANDA

Utilizzando la trasformata wavelet si associa un segnale  $x$ , ovvero un'immagine, ai propri coefficienti. I coefficienti sono suddivisi nel segnale a bassa risoluzione e nella rappresentazione del dettaglio. Rappresentando i coefficienti in una matrice si vede come l'immagine sia suddivisa a blocchi. Nelle immagini 2.1 è illustrato un esempio nel caso la trasformata sia fatta con un solo livello (2.1b) o che sia fatta con tre livelli (2.1c):

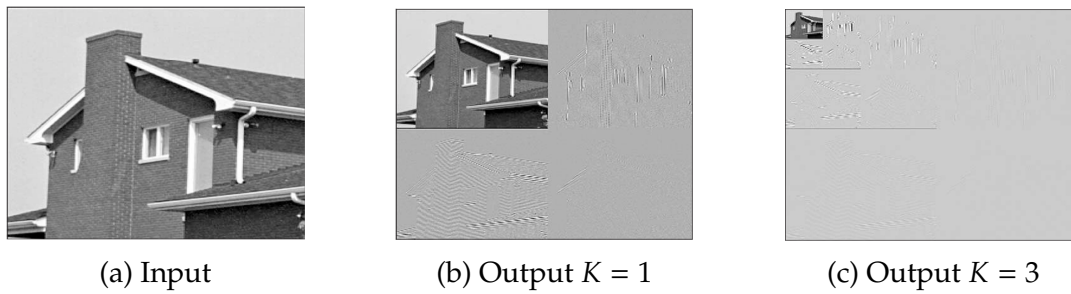


Figura 2.1: Esempio input e output della trasformata wavelet

Si nota come i vari blocchi che compongono l'immagine sono in genere incorrelati. Ogni blocco avrà una propria distribuzione a cui è associata una varianza. Quindi è possibile quantizzare ogni blocco in modo indipendente rispetto agli altri blocchi.

### 2.2.1 QUANTIZZAZIONE A BLOCCHI

Un'immagine può essere rappresentata come un processo aleatorio continuo  $X$ , allora quantizzandola uniformemente, la distorsione sarà proporzionale alla varianza dell'immagine e decrescerà esponenzialmente con il tasso (numero di bit per pixel):

$$D = h_x \sigma_X^2 2^{-2R}$$

Dove la distorsione è l'errore quadratico medio associato alla quantizzazione e  $h_x$  è dovuto alla forma della densità di probabilità di  $X$ .

Supponiamo ora che l'immagine sia suddivisa in blocchi, allora possiamo rappresentare l'immagine:

$$X = [X_1, X_2, \dots, X_N]$$

## 2.2. CODIFICA A SOTTO BANDA

La distorsione dell'intera immagine è:

$$\begin{aligned} D &= \frac{1}{N} \|X - Q(X)\|^2 = \frac{1}{N} E \left[ (X - Q(X))(X - Q(X))^T \right] = \\ &= \frac{1}{N} E \left[ \sum_{i=1}^N (X_i - Q(X_i))^2 \right] = \frac{1}{N} \sum_{i=1}^N E \left[ (X_i - Q(X_i))^2 \right] = \\ &= \frac{1}{N} \sum_{i=1}^N h_i \sigma_i^2 2^{-2R_i} = D(R) \end{aligned}$$

Si vuole scegliere  $R$  in modo tale da minimizzare la distorsione dell'intera immagine considerando un vincolo sul tasso, ovvero:

$$R_T = \sum_{i=1}^N R_i$$

Si dimostra che:

$$R_i = \bar{R} + \frac{1}{2} \log_2 \frac{h_i \sigma_i^2}{h_{GM} \sigma_{GM}^2}$$

Dove:

$$\bar{R} = \frac{R_T}{N} \quad \sigma_{GM}^2 = \left( \prod_{i=1}^N \sigma_i^2 \right)^{\frac{1}{N}} \quad h_{GM} = \left( \prod_{i=1}^N h_i \right)^{\frac{1}{N}}$$

Sostituendo nella distorsione dell'intera immagine:

$$D^* = h_{GM} \sigma_{GM}^2 2^{-2\bar{R}}$$

### 2.2.2 CODIFICA CON TRASFORMATA

Consideriamo una trasformazione lineare, quindi possiamo rappresentarla attraverso una matrice, dove  $X$  è il segnale in ingresso e  $Y$  i coefficienti ottenuti dalla trasformazione:

$$Y = AX$$

Siccome consideriamo una trasformazione ortogonale si ha che  $A^{-1} = A^T$ . Adesso si quantizza il segnale trasformato e si ricostruisce il segnale:

$$X_q = A^{-1}Q(y)$$

Quindi la distorsione:

$$\begin{aligned} D_X &= \frac{1}{N} \|X - X_q\|^2 = \frac{1}{N} \|A^{-1}Y - A^{-1}Q(Y)\|^2 = \\ &= \frac{1}{N} (A^{-1}Y - A^{-1}Q(Y))^T (A^{-1}Y - A^{-1}Q(Y)) = \\ &= \frac{1}{N} (Y - Q(Y))^T AA^T (Y - Q(Y)) = D_Y \end{aligned}$$

Nel caso che  $Y$  sia divisa in blocchi:

$$D_Y^* = h_{GM,Y} \sigma_{GM,Y}^2 2^{-2\bar{R}}$$

Ciò significa che la distorsione ottimale basata sulla trasformazione è ottenuta utilizzando l'allocazione ottimale  $X$ , e dobbiamo utilizzare la media geometrica delle varianze del vettore trasformato e delle forme  $Y$ . Pertanto, l'approccio di codifica tramite trasformazione potrebbe ridurre la distorsione per lo stesso rate se il vettore trasformato ha una media geometrica delle varianze minore rispetto a quelle del vettore di input.

Definiamo il guadagno di codifica come:

$$G = \frac{D_X^*}{D_Y^*} = \frac{h_{GM,X} \sigma_{GM,X}^2}{h_{GM,Y} \sigma_{GM,Y}^2}$$

Nel caso  $X$  sia una gaussiana,  $h_{GM,X} = h_{GM,Y}$  in quanto una trasformazione lineare ortogonale mappa una gaussiana in una gaussiana. Siccome  $X$  è un vettore di variabili identicamente distribuite, tutte le componenti del vettore di input hanno la stessa varianza:

$$\sigma_{GM,X}^2 = \sigma_X^2 = \sigma_{AM,X}^2$$

### 2.3. EZW

In particolare:

$$\begin{aligned}\sigma_{AM,X}^2 &= E(X_i^2) = \frac{1}{N} \sum_{i=1}^N E(X_i^2) = \frac{1}{N} E\left(\sum_{i=1}^N X_i^2\right) = \frac{1}{N} E(\|X\|^2) = \\ &= \frac{1}{N} E(\|Y\|^2) = \sigma_{AM,Y}^2\end{aligned}$$

Quindi si ottiene che il guadagno di codifica:

$$G = \frac{\sigma_{AM,Y}^2}{\sigma_{GM,Y}^2}$$

Il risultato ottenuto significa che l'utilizzo di diverse trasformato implica un diverso guadagno di codifica. Di conseguenza ci si aspetta che con l'utilizzo di JPEG o di JPEG2000 che utilizzano trasformate diverse si abbiano delle prestazioni differenti in termini di compressione.

A partire da quest'analisi teorica si sono sviluppati diversi metodi pratici come EZW e JPEG2000, che sono descritti nelle prossime sezioni.

## 2.3 EZW

Inizialmente i codificatori wavelet erano basati sulla codifica a sotto bande fin quando Jerry Shapiro nel 1993 ha introdotto la codifica EZW (Embedded Zero-Tree Wavelet). Questa codifica sfrutta le proprietà della rappresentazione a multirisoluzione. In genere nella rappresentazione a multirisoluzione la maggior parte dell'energia del segnale è contenuta nei componenti del segnale a bassa risoluzione. Nel caso delle immagini gran parte dell'energia è contenuta della banda LL e in aggiunta l'energia nelle altre sotto bande è anch'essa concentrata in un numero relativamente piccolo di coefficienti. Si può scegliere come modello per la densità di probabilità dell'energia dei coefficienti nelle sotto bande ad alta frequenza una gaussiana generalizzata, che ha la seguente espressione[13]:

$$f(x) = A e^{-\left(\frac{|x|}{\sigma^2}\right)^\beta}$$

Dove  $\sigma^2$  è la varianza e  $\beta$  è il rolloff. Si noti che la variabile aleatoria Gaussiana e la variabile aleatoria Laplaciana sono casi particolari corrispondenti rispet-

tivamente a  $\beta = 2$  e  $\beta = 1$ . In figura 2.2 è rappresentato un esempio della distribuzione.

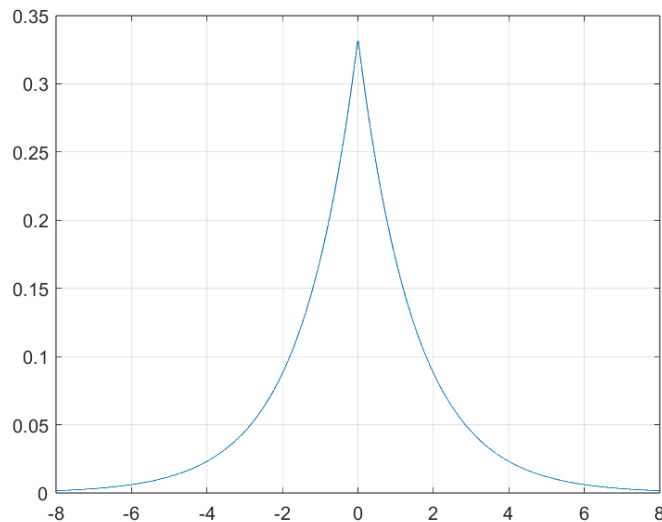


Figura 2.2: Distribuzione con parametri:  $\sigma = 1.5$ ,  $A = 1/3$ ,  $\beta = 1$

Un'altra proprietà da far presente è la forte correlazione tra i coefficienti tra le varie bande ad alta frequenza. Infatti in queste bande viene contenuta l'informazione dei dettagli dell'immagine rispetto a una direzione (orizzontale, verticale, diagonale) con fattori di scala differenti.

Consideriamo la decomposizione in figura 2.3 il coefficiente  $a$  della banda  $LL$  rappresenta la stessa zona spaziale dei coefficienti  $a_i$  con  $i \in \{1, 2, 3, 4\}$  nelle altre sotto bande[11]. A sua volta il coefficiente  $a_i$  rappresenta la stessa locazione dei coefficienti  $a_{ij}$  con  $j \in \{1, 2, 3, 4\}$  nelle altre sotto bande. In questo modo ad ogni coefficiente gli si può attribuire quattro discendenti, ad ogni discendenti si possono attribuire quattro nuovi discendenti e così via. Procedendo in questo modo per ogni coefficiente dell'immagine a bassa risoluzione si costruisce una struttura ad albero.

### 2.3. EZW

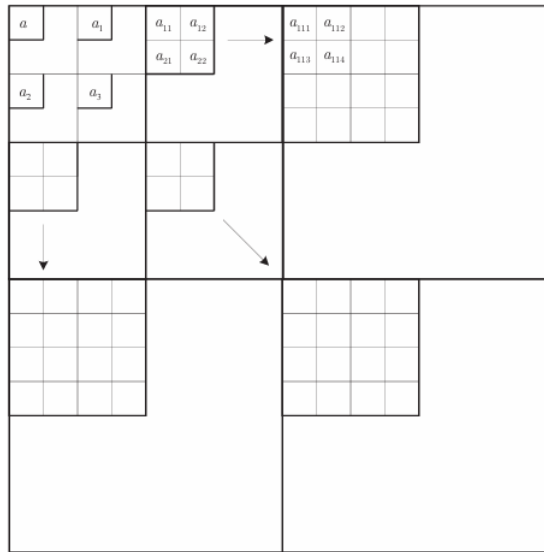


Figura 2.3: Struttura ad albero di EZW

Siccome la decomposizione wavelet tende a compattare l'energia dei segnali a maggiore risoluzione, i coefficienti vicini alla radice dell'albero tendono ad assumere valori maggiori rispetto ai discendenti. Ciò equivale ad assumere che se un coefficiente ha un valore assoluto minore di una data soglia  $T$ , si può affermare che con un'elevata probabilità tutti i discendenti hanno valore minore di quella soglia.

Procediamo introducendo la terminologia introdotta da Shapiro:

- Se un coefficiente è maggiore della soglia  $T$  è detto significativo;
- Se un coefficiente è minore della soglia e se tutti i suoi discendenti sono minori della soglia è detto radice zerotree;
- Se un coefficiente è minore della soglia ma se esiste un suo discendente maggiore della soglia è detto isolated zero;

Inizialmente la soglia assume un valore uguale:

$$T_0 = 2^{\lfloor \log_2 |c_{max}| \rfloor}$$

dove  $c_{max}$  è il più grande tra i valori assoluti dei coefficienti. L'algoritmo si suddivide in due passi:

1. Il passo dominante, in cui si definisce la mappa dei coefficienti significativi rispetto alla soglia  $T_i$ . Tali valori assoluti si trovano nell'intervallo  $[T_i, 2T_i)$  e vengono rappresentati tramite un valore di ricostruzione  $3T_i/2$ . In seguito al primo passo dominante i coefficienti sono divisi in due liste, nelle quali



si suddividono tutti i coefficienti non significativi nel passo precedente (lista dominante) e tutti i coefficienti significativi nel passo precedente (lista subordinata);

2. Il passo subordinato, in cui si attribuisce il valore dei nuovi bit più significativi, nella rappresentazione binaria dei coefficienti della lista tutti i coefficienti significativi;

Ad ogni passo si dimezza la soglia e ad ogni coefficiente gli si attribuisce una delle seguenti etichette:

- positivo significativo;
- negativo significativo;
- radice zerotree;
- zero isolato;

L'algoritmo procede nel seguente modo:

1. **Inizializzazione:** si inseriscono tutti i coefficienti nella lista dominante mediante l'ordine di scansione in figura 2.4 e si inizializza la soglia;

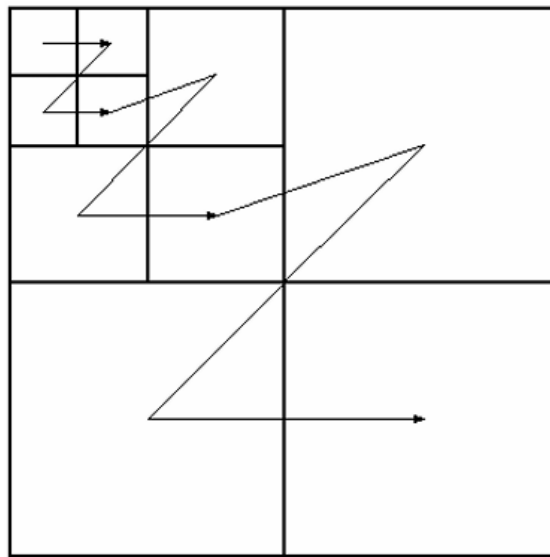


Figura 2.4: Scansione EZW

2. **Passo Dominante:** Si effettua la scansione della lista dominante e si associa, ad ogni coefficiente, la relativa etichetta ed il relativo valore di ricostruzione. I coefficienti che discendono da uno zerotree non vengono codificati. I coefficienti significativi vengono spostati nella lista subordinata ed i relativi valori nella lista dominante vengono posti a zero. Nel file compresso vengono codificati gli aggiornamenti alla lista dominante e le informazioni sullo spostamento dei coefficienti significativi nella lista subordinata.

## 2.4. JPEG 2000

3. **Passo Subordinato:** Si effettua la scansione della lista subordinata e al coefficiente gli si attribuisce il valore 1 se è maggiore del valore di ricostruzione o il valore 0 se è minore del valore di ricostruzione. Nel file compresso vengono codificati i valori binari che rappresentano la presenza o l'assenza di coefficienti significativi nella lista subordinata.
4. **Calcolo della soglia** Si calcola la nuova soglia dimezzando la soglia precedente. Nel caso il bit-rate non sia sufficiente si ripete il procedimento dal passo due. Nel file compresso vengono codificati i parametri aggiornati relativi alla soglia.

Gli esperimenti di Shapiro dell'articolo [10] mostrano che, utilizzando il codificatore, EZW i risultati mostrano PSNR più alto rispetto ad altri metodi riportati in precedenti studi a bit rate tra 0,25 e 1. Inoltre il confronto con JPEG mostra che EZW offre una qualità visiva superiore per lo stesso tasso di bit. Anche a parità di PSNR, EZW richiede meno spazio di archiviazione. Gli artefatti a bassi bit rate sono meno fastidiosi rispetto agli artefatti di blocco della codifica a trasformata a blocchi.

## 2.4 JPEG 2000

JPEG2000 non utilizza l'algoritmo EZW, ma segue un'altra struttura. La codifica può essere suddivisa in due fasi distinte.

Nella prima fase, viene eseguita la trasformata wavelet discreta seguita dalla quantizzazione. Successivamente, si effettua una codifica senza perdita dei codeblock. Dove i codeblock sono blocchi rettangolari di coefficienti wavelet che vengono processati separatamente durante la compressione.

Nella seconda fase, viene implementato EBCOT (embedded block coding with optimized truncation) insieme alla gestione della scalabilità (qualità, risoluzione) e del ROI. Dove la ROI è una funzionalità che consente di dare priorità a determinate aree di un'immagine durante la compressione.

Nella prima fase, detta first tier, coefficienti della trasformata wavelet sono quantizzati molto finemente. Per il caso di codifica senza perdita, i coefficienti sono interi e non vengono quantizzati. Di conseguenza non è nella quantizzazione in cui si realizza la compressione e la perdita d'informazione associata. La perdita viene effettuata nella seconda fase tramite il troncamento dei bitstream che rappresentano i codeblock.

Nella seconda fase ogni sottobanda viene divisa in blocchi di coefficienti di dimensioni uguali, chiamati codeblock. I codeblock vengono codificati

senza perdita e indipendentemente con un codificatore aritmetico, generando tanti flussi di bit quanti sono i codeblock nell'immagine. Nell'immagine 2.5 è illustrato un esempio di flusso di bit per ogni codeblock.

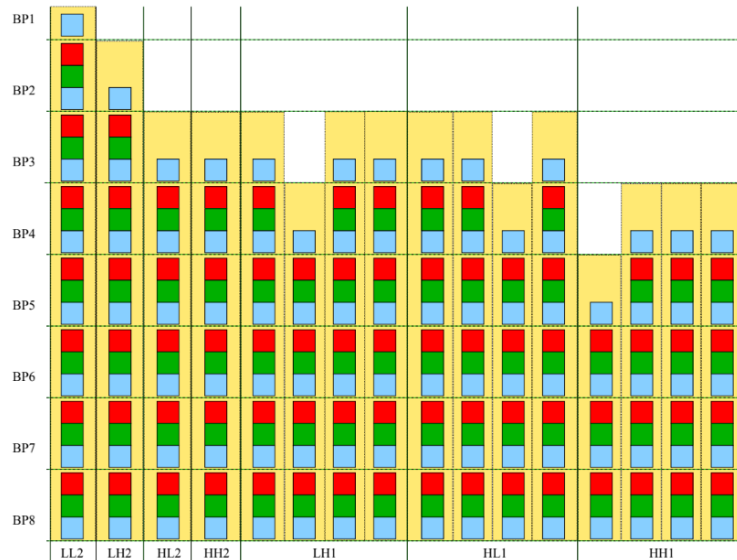


Figura 2.5: Flusso di bit per ogni codeblock

In particolare ogni flusso di bit è suddiviso in piani di bit. Supponiamo di considerare un piano di bit  $i$ -esimo corrispondente ad un codeblock. Allora in quel piano di bit sono contenute tutte le cifre  $i$ -esime di tutti i coefficienti della trasformata del corrispondente codeblock.

Mantenendo tutti i flussi di bit di tutti i codeblock, otteniamo un bit rate molto elevato quindi si troncano i flussi di bit per raggiungere il bit rate desiderato. Bisogna ottimizzare il troncamento per minimizzare la distorsione. Utilizzando i moltiplicatori di Lagrange, si dimostra che il punto di troncamento ottimo è caratterizzato da avere la stessa pendenza della curva tasso distorsione, indipendentemente dal codeblock  $i$  che stiamo considerando. In formule:

$$\frac{\partial D_i}{\partial R_i} = -\lambda$$

Dove  $D_i$  e  $R_i$  sono la distorsione e il bit rate associato all'  $i$ -esimo codeblock e  $\lambda$  è coefficiente ottenibile attraverso un algoritmo interattivo.

In figura sono illustrate le curve rate-distorsione per ogni codeblock. 2.6.

## 2.5. DIFFERENZA DI PRESTAZIONI TRA JPEG E JPEG2000

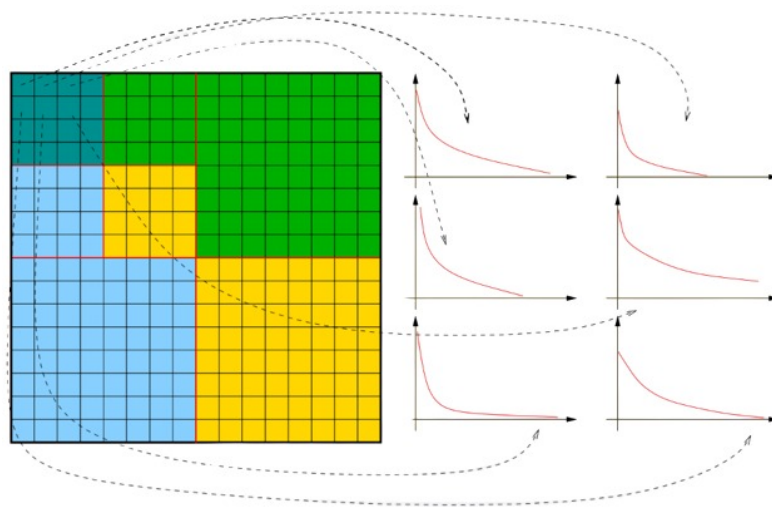


Figura 2.6: rate-distorsione per ciascun codeblock

Troncando i flussi di bit di un codeblock si troncano tutti i coefficienti associati al corrispettivo codeblock. Il file JPEG2000 è costituito dai bitstream troncati dei vari codeblock. L'inizio e la fine di ogni stream è individuato da opportuni marker.

## **2.5** DIFFERENZA DI PRESTAZIONI TRA JPEG E JPEG2000

Passiamo ora ad analizzare la differenza di prestazioni tra JPEG e JPEG2000. Si consideri un insieme d'immagini, illustrate in 2.7, che saranno utilizzate come esempio.



(a) Prima immagine



(b) Seconda immagine



(c) Terza immagine



(d) Quarta immagine



(e) Quinta immagine



(f) Sesta immagine

Figura 2.7: Immagini di partenza

Un'immagine a colori è composta da tre canali: rosso, verde e blu. Ogni pixel è composto da una combinazione di questi canali. Di conseguenza un'immagine a colori è uguale a considerare tre immagini ognuna delle quali assume dei valori del corrispondente canale. Per semplicità consideriamo delle immagini composte da un unico canale. Quindi prendiamo le versioni in bianco e nero delle immagini a colori dell'insieme di partenza, illustrate in 2.8.

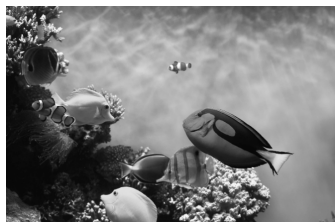
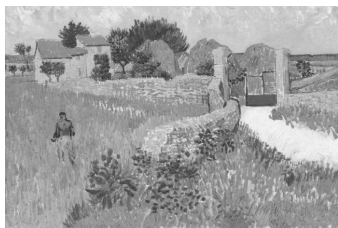


Figura 2.8: Immagini di partenza in bianco e nero

Per analizzare la differenza tra JPEG e JPEG2000 operiamo una compressione delle immagini con perdita utilizzando le due tecniche. Le immagini, in seguito

## 2.5. DIFFERENZA DI PRESTAZIONI TRA JPEG E JPEG2000

alla compressione, avranno un bit rate minore rispetto all'immagine di partenza. Si vuole analizzare la differenza tra le immagini di partenza e l'immagine compressa. Questa caratteristica è rappresentata dal PSNR definito nel seguente modo.

Consideriamo un'immagine di dimensione  $N \times M$  da comprimere scelta dell'insieme 2.8 che chiamiamo  $I$  e chiamiamo  $K$  la corrispondente immagine compressa. Definiamo il PSNR come:

$$PSNR = 20 \log_{10} \left( \frac{\max\{I\}}{\sqrt{MSE}} \right)$$

Dove  $\max\{I\}$  (in genere 255) è il massimo valore assumibile da un pixel dell'immagine di partenza e  $MSE$  è l'errore quadratico definito come:

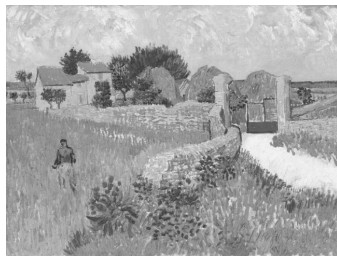
$$MSE = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M |I(i, j) - K(i, j)|^2$$

In figura 2.10 sono illustrati i grafici del PSNR, utilizzando JPEG o JPEG2000, al variare del bit rate delle immagini 2.8. Come è visibile dai grafici il PSNR associato a JPEG2000 è sempre maggiore del PSNR associato a JPEG a parità di bit rate. Ciò implica che l'errore quadratico medio dovuto alla compressione è sempre minore nel caso di JPEG2000 a parità di bit rate. Considerando i grafici possiamo anche dire che un'immagine compressa con JPEG con una certa distorsione  $D$  avrà un bit rate maggiore rispetto alla stessa immagine compressa con JPEG2000 che ha la stessa distorsione  $D$ .

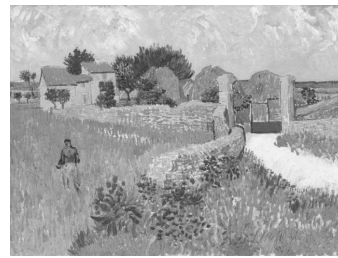
In Figura 2.9 sono illustrate delle immagini di esempio compresse con i formati JPEG e JPEG2000, con bit rate simile.



JPEG bit rate = 0.12



JPEG bit rate = 1.48



JPEG bit rate = 4.94



JPEG2000 bit rate = 0.12



JPEG2000 bit rate = 1.58



JPEG2000 bit rate = 4.90

Figura 2.9: Immagini di esempio

## 2.5. DIFFERENZA DI PRESTAZIONI TRA JPEG E JPEG2000

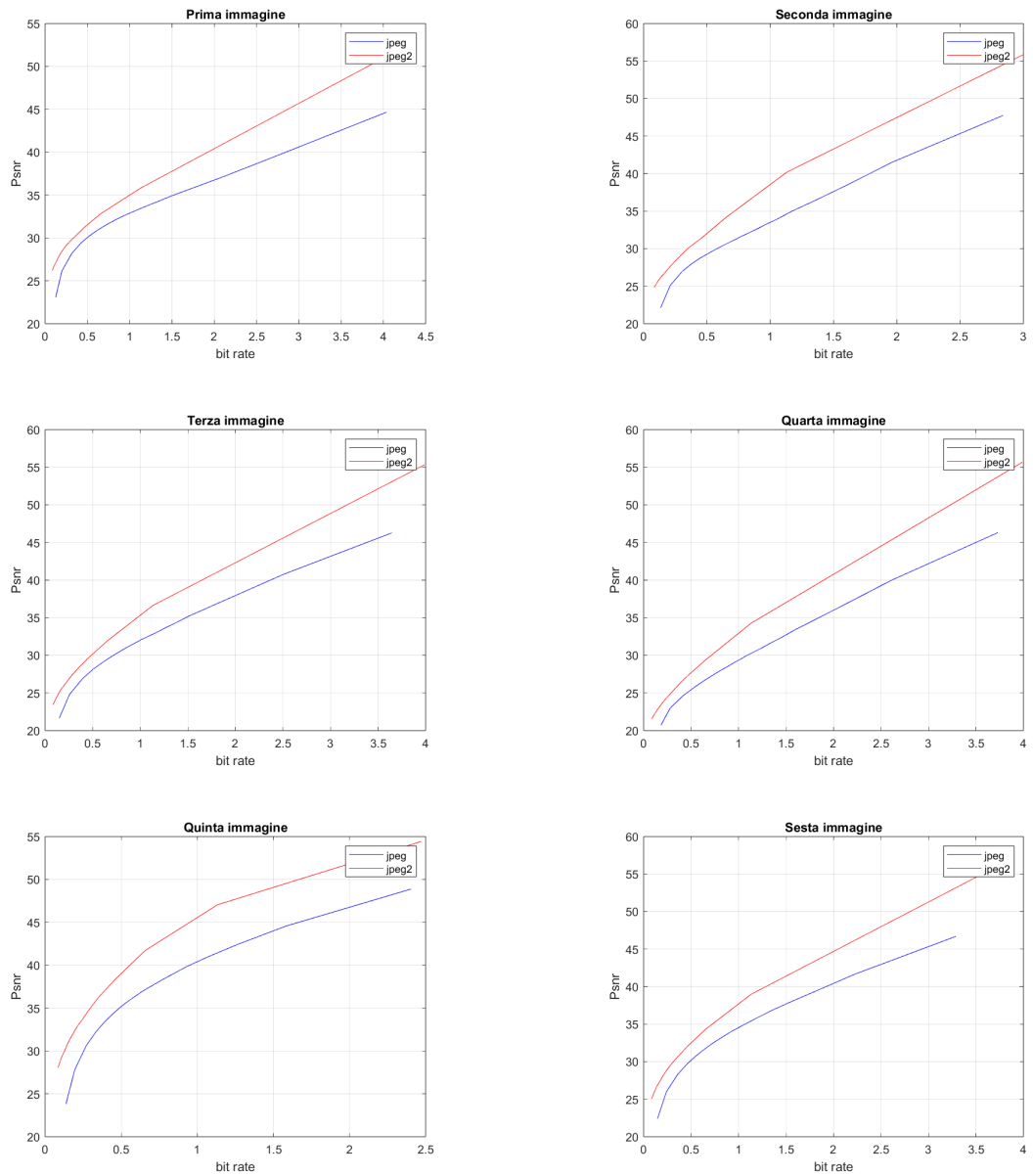


Figura 2.10: PSNR JPEG e JPEG2000





## Segmentazione

La segmentazione è un'operazione dove si suddivide un'immagine in regioni in cui sono contenuti degli oggetti appartenenti all'immagine. Il fine è quello di semplificare un'immagine considerando solo le parti più significative. Inoltre ci sono molte applicazioni alla segmentazione, sia di basso livello come il miglioramento dell'immagine, denoising e sharpening, sia di alto livello come la segmentazione semantica, decisioni di agenti, come per la guida autonoma.

La segmentazione ha vari campi di applicazione come:

- Imaging medico;
- Rilevamento di oggetti;
- Compiti di riconoscimento;
- Sistemi di controllo del traffico;

In alcune applicazioni si vuole operare sulle immagini compresse invece delle immagini originali, per diminuire lo spazio occupato dalle immagini. La segmentazione è strettamente legata all'individuazione dei bordi degli oggetti. Per come è stato descritto in 1.1.1 l'informazione dei bordi è meglio contenuta nella trasformata wavelet rispetto alla trasformata di Fourier. Di conseguenza ci aspettiamo che l'operazione di segmentazione su un'immagine compressa con JPEG2000 si avvicini di più alla segmentazione dell'immagine non compressa rispetto a JPEG.

Procediamo analizzando qualitativamente il comportamento di una segmentazione di una immagine compressa con JPEG o con JPEG2000.

### 3.1. SEGMENTAZIONE D' IMMAGINI COMPRESSE

## 3.1 SEGMENTAZIONE D' IMMAGINI COMPRESSE

Per la segmentazione useremo SAM (Segment Anything Model) un modello AI di Meta[6]. Consideriamo due immagini di partenza da comprimere:



(a) Prima immagine



(b) seconda immagine

Figura 3.1: Immagini da comprimere

Iniziamo considerando una compressione con una grande perdita:



(a) JPEG



(b) JPEG2000



(c) JPEG



(d) JPEG2000

Figura 3.2: Immagini compresse con bit rate di 0.2

Le maschere generate dalla segmentazione sono le parti evidenziate in azzurro. Nelle immagini 3.2a e 3.2b si è svolta la segmentazione sul cane e nelle

immagini 3.2c e 3.2d si è svolta la segmentazione sul cavallo di destra. In questo caso 3.2 in cui il bit rate è di circa 0.2 e il rapporto di compressione (dimensione tra immagini originali e compresse) è uguale a  $24 [bpp] / 0.2 [bpp] = 120$  (24 bit per pixel è dato da tre canali di colore, ognuno rappresentato su 8 bit). La differenza tra la segmentazione delle due immagini codificate in modo diverso è molto evidente. È possibile vedere che nel caso del JPEG la segmentazione non è in grado d'identificare anche gli oggetti più grandi, come nel caso del cane per la 3.2a o come il cavallo per la 3.2c. Invece utilizzando JPEG2000, le maschere ben identificano gli oggetti più grandi, anche se i bordi in alcuni casi si discostano molto dall'immagine originale, come l'orecchio del cane della figura 3.2b, o il muso del cavallo dell'immagine 3.2d.

Ora consideriamo un bit rate più elevato:



(a) JPEG



(b) JPEG2000



(c) JPEG



(d) JPEG2000

Figura 3.3: Immagini compresse con bit rate di 0.7

Nelle immagini 3.3a e 3.3b si è svolta la segmentazione sul cuscino di destra e nelle immagini 3.3c e 3.3d si è svolta la segmentazione sul cavallo di sinistra. In questo caso in cui il bit rate è di 0.7 e il rapporto di compressione è uguale a  $24 [bpp] / 0.7 [bpp] \approx 34.3$  le segmentazioni differiscono di poco. Infatti per la maggior parte degli oggetti le maschere sono identiche all'immagine originale. Nelle immagini 3.3a e 3.3b sono illustrati alcuni casi critici per JPEG. Si vede in

### 3.2. SEGMENTAZIONE E PSNR

3.3c e 3.3d che nel caso di JPEG2000 questi casi critici non sono presenti. Infatti si vede che nel caso di un bit rate di 0.7 le maschere delle immagini JPEG2000 in genere non differiscono dalle maschere dell'immagini originali. In genere si vede che anche per JPEG nel caso di un bit rate maggiore di uno la differenza delle maschere è minima rispetto all'immagine originale.

## 3.2 SEGMENTAZIONE E PSNR

Consideriamo le due immagini in figura 3.1. Nella tabella 3.1 si analizzano i bit rate e il PSNR necessari per ottenere una segmentazione degli oggetti che non si "discosta" troppo dalla segmentazione dell'immagine originale.

|          | Immagine | Oggetto  | $b^*$  | $PSNR^*$ |
|----------|----------|----------|--------|----------|
| JPEG     | Prima    | Cane     | 0.3916 | 33.8587  |
| JPEG2000 |          |          | 0.2244 | 33.5292  |
| JPEG     | Prima    | Divano   | 1.004  | 40.1687  |
| JPEG2000 |          |          | 0.5293 | 40.2504  |
| JPEG     | Prima    | Tappeto  | 0.2957 | 32.1961  |
| JPEG2000 |          |          | 0.1738 | 31.9728  |
| JPEG     | Prima    | Lampada  | 0.4288 | 34.4692  |
| JPEG2000 |          |          | 0.2711 | 34.7580  |
| JPEG     | Seconda  | Cavallo  | 0.3458 | 33.2641  |
| JPEG2000 |          |          | 0.1959 | 33.3513  |
| JPEG     | Seconda  | Nuvola   | 0.3133 | 32.6380  |
| JPEG2000 |          |          | 0.1737 | 32.5701  |
| JPEG     | Seconda  | Montagna | 0.2866 | 32.1476  |
| JPEG2000 |          |          | 0.2618 | 35.1643  |

Tabella 3.1

Come si può vedere nella tabella 3.1 nella maggior parte degli oggetti il PSNR minimo è molto simile tra JPEG e JPEG2000. L'unico caso in cui i PSNR si discostano è la montagna. Questo risultato è dovuto alla "polvere" che sfuma i contorni della montagna nella parte di sinistra. Perciò possiamo concludere che in genere la qualità della segmentazione di un oggetto ben definito all'interno di un immagine è dovuta dal PSNR.

Nelle figure 3.4 e di 3.5 le immagini con le maschere dove nella prima colonna è presente l'immagine originale, nella seconda colonna è presente l'immagine in formato JPEG e terza colonna è presente l'immagine in formato JPEG2000.

Dai risultati ottenuti, possiamo dedurre che esiste una relazione tra la qualità della compressione dell'immagine misurata con il PSNR e la qualità della segmentazione ottenibile. In particolare, questa relazione è sempre valida quando l'oggetto è ben definito all'interno dell'immagine. Al contrario, se l'oggetto ha bordi sfumati, la relazione tra la qualità della compressione e la qualità della segmentazione tende a indebolirsi. Quindi nel caso di oggetti ben definiti il PSNR può essere usato come indicatore della qualità che si può ottenere con la segmentazione.

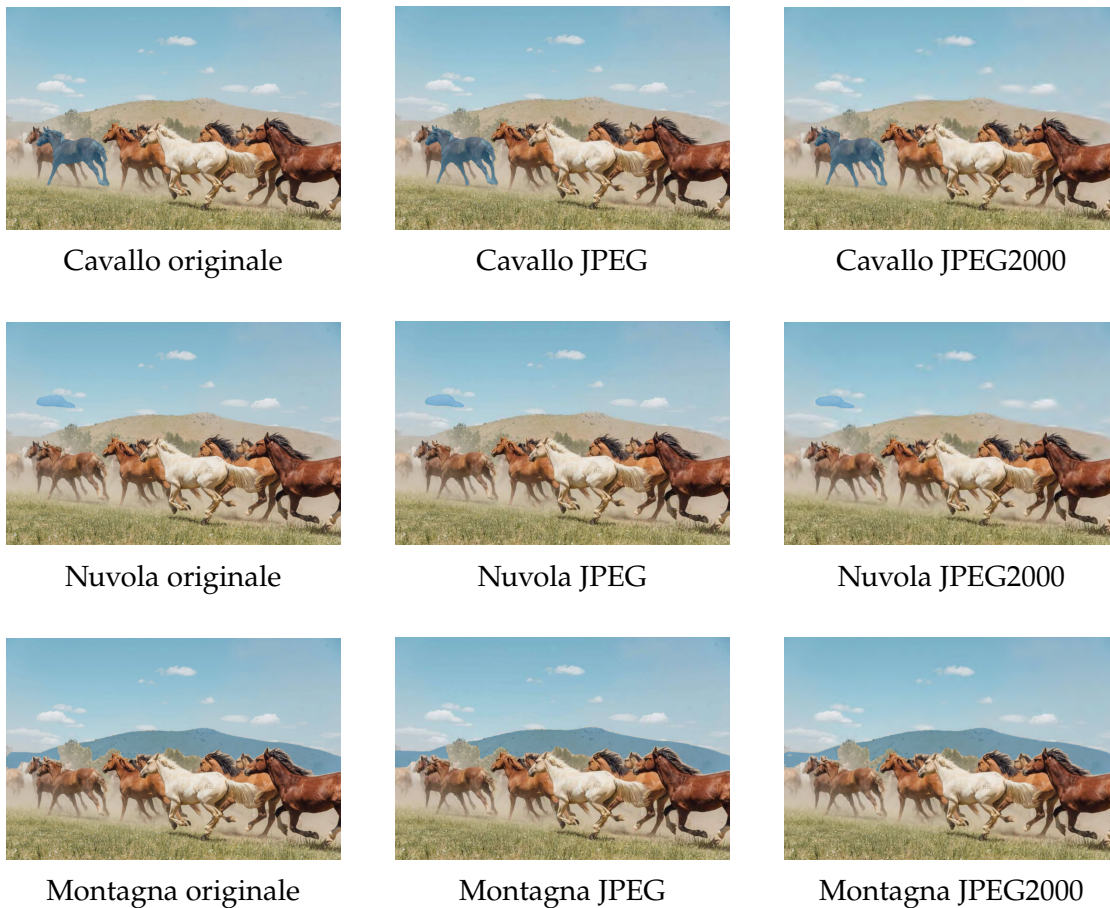


Figura 3.5: Maschere riferite alla seconda immagine di tabella 3.1



### 3.2. SEGMENTAZIONE E PSNR



Cane originale



Cane JPEG



Cane JPEG2000



Divano originale



Divano JPEG



Divano JPEG2000



Tappeto originale



Tappeto JPEG



Tappeto JPEG2000



Lampada originale



Lampada JPEG



Lampada JPEG2000

Figura 3.4: Maschere riferite alla prima immagine di tabella 3.1

# Conclusioni

Questa tesi affronta l'uso della trasformata wavelet come strumento matematico per la compressione delle immagini e l'applicazione nella segmentazione. Il lavoro inizia con una panoramica della teoria di base della trasformata wavelet, mettendo in luce le sue caratteristiche principali. Si prosegue esaminando la compressione a blocchi e la sua implementazione tramite la trasformata wavelet, con un confronto delle prestazioni tra i metodi di compressione JPEG e JPEG2000.

Si mostra che JPEG2000 offre vantaggi significativi in termini di qualità dell'immagine ed efficienza della compressione rispetto a JPEG.

In seguito si analizza la relazione tra la qualità di compressione delle immagini e l'accuratezza della segmentazione. I risultati mostrano che esiste una relazione tra la qualità della compressione e l'accuratezza della segmentazione per immagini con oggetti ben definiti. Tuttavia, questa relazione si indebolisce per immagini con bordi sfumati, suggerendo che la qualità della segmentazione è meno dipendente dalla compressione in certi casi.

In sviluppi futuri, sarà interessante approfondire il motivo per cui in certi casi il PSNR non è un buon indicatore della qualità della segmentazione.





# Bibliografia

- [1] Paul S Addison. *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance*. CRC press, 2017.
- [2] Alan C Bovik. *The essential guide to image processing*. Academic Press, 2009.
- [3] Liu Chun-Lin. «A tutorial of the wavelet transform». In: *NTUEE, Taiwan* 21.22 (2010), p. 2.
- [4] Ingrid Daubechies. *Ten lectures on wavelets*. SIAM, 1992.
- [5] Lorenzo Finesso, M Pavon e S Pinzoni. *Segnali e sistemi*. Libreria Progetto, 2021.
- [6] Alexander Kirillov et al. «Segment Anything». In: *arXiv:2304.02643* (2023).
- [7] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.
- [8] Amir Said e William A Pearlman. «A new, fast, and efficient image codec based on set partitioning in hierarchical trees». In: *IEEE Transactions on circuits and systems for video technology* 6.3 (1996), pp. 243–250.
- [9] Khalid Sayood. *Introduction to data compression*. Morgan Kaufmann, 2017.
- [10] J.M. Shapiro. «Embedded image coding using zerotrees of wavelet coefficients». In: *IEEE Transactions on Signal Processing* 41.12 (1993), pp. 3445–3462. DOI: 10.1109/78.258085.
- [11] Jerome M Shapiro. «Embedded image coding using zerotrees of wavelet coefficients». In: *IEEE Transactions on signal processing* 41.12 (1993), pp. 3445–3462.
- [12] Gilbert Strang e Truong Nguyen. *Wavelets and filter banks*. SIAM, 1996.
- [13] Bryan E Usevitch. «A tutorial on modern lossy wavelet image compression: foundations of JPEG 2000». In: *IEEE signal processing magazine* 18.5 (2001), pp. 22–35.

## BIBLIOGRAFIA

- [14] Martin Vetterli e Jelena Kovacevic. *Wavelets and subband coding*. Prentice-hall, 1995.