JPEG 2000

JPEG 2000

- E’ un nuovo algoritmo per la compressione delle immagini
- Il JPEG comittee ha definito lo standard ISO/IEC 15444-1 | ITU-T Rec. T.800: JPEG2000 Image Coding System

JPEG 2000 - Caratteristiche

- Compressione a basso bit rate
- Compressione lossless e lossy
- Trasmissione progressiva in funzione dell'accuratezza dei pixel e della risoluzione
- Codifica per regioni di interesse

JPEG 2000 – Caratteristiche (2)

- Robustezza ai bit-errors
- Codestream a elaborazione e accesso casuale
- Architettura aperta
- Content based description

JPEG 2000 - Applicazioni

- Internet
- Mobile
- Stampa
- Fotografia digitale
- Medicina
- Librerie digitali
- E-commerce

JPEG 2000 - Partecipanti

- 21 paesi / 80-100 incontri
- Europa:
  Ericsson, Nokia, Philips, Canon, Motorola, etc.
- USA/Canada:
  Kodak, HP, Ricoh, Sharp, Adobe, etc.
- Asia/Australia:
  Samsung, Sony, Panasonic, etc

JPEG 2000 - Standard

- Parte 1: Core image coding system
- Parte 2: extensions
- Parte 3: motion JPEG 2000
- Parte 4: conformance
- Parte 5: reference software
- Parte 6: compound file format
- Parte 7: techincal report

JPEG 2000 – caratteristiche parte 1

- Efficienza ad alta compressione
- Trasformazione di colore senza perdita
- Codifica lossless e lossy in un solo algoritmo
- Trasmissione progressiva in risoluzione, qualità, etc
- Codifica-decodifica statica-dinamica di ROI

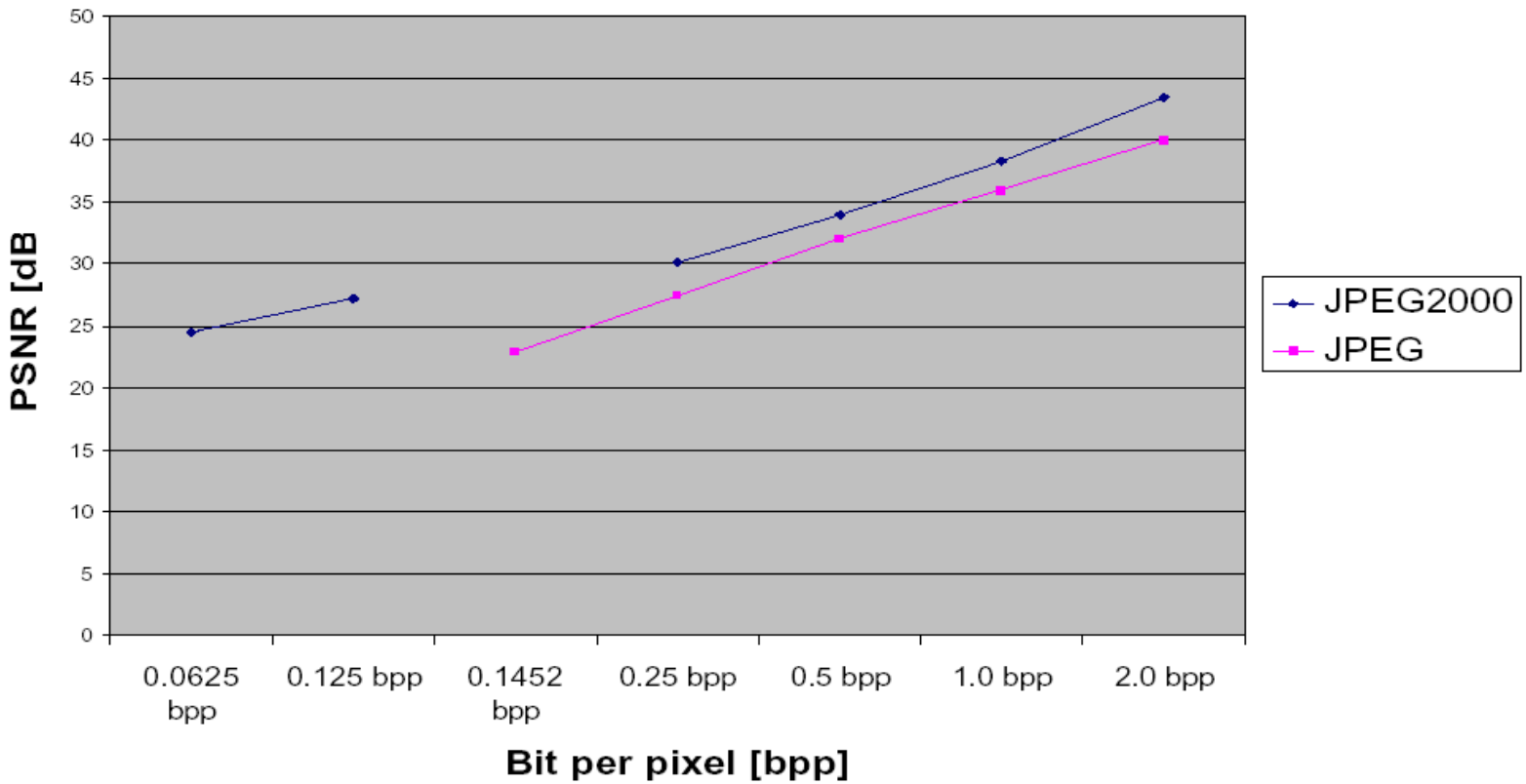
JPEG 2000 – caratteristiche parte 1 (2)

- Robustezza agli errori
- Codifica a qualità percettiva
- Codifica d'immagine a componente multiplo
- Formato di file poco pesante (opzionale)

JPEG2000 – esempi

JPEG 2000
contro
JPEG baseline

JPEG2000 - esempi (2)



JPEG2000 - esempi (3)

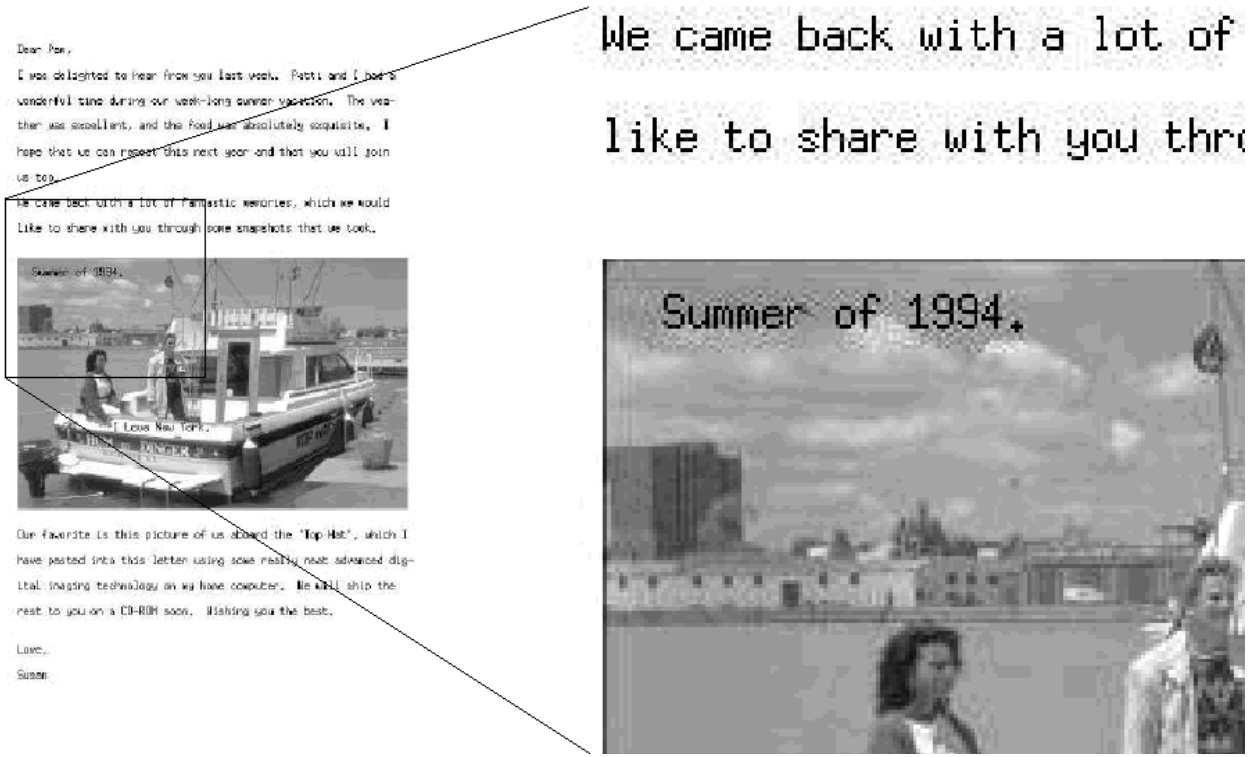
JPEG a 0.125 bpp

JPEG2000 - esempi (4)

JPEG 2000 a 0.125 bpp

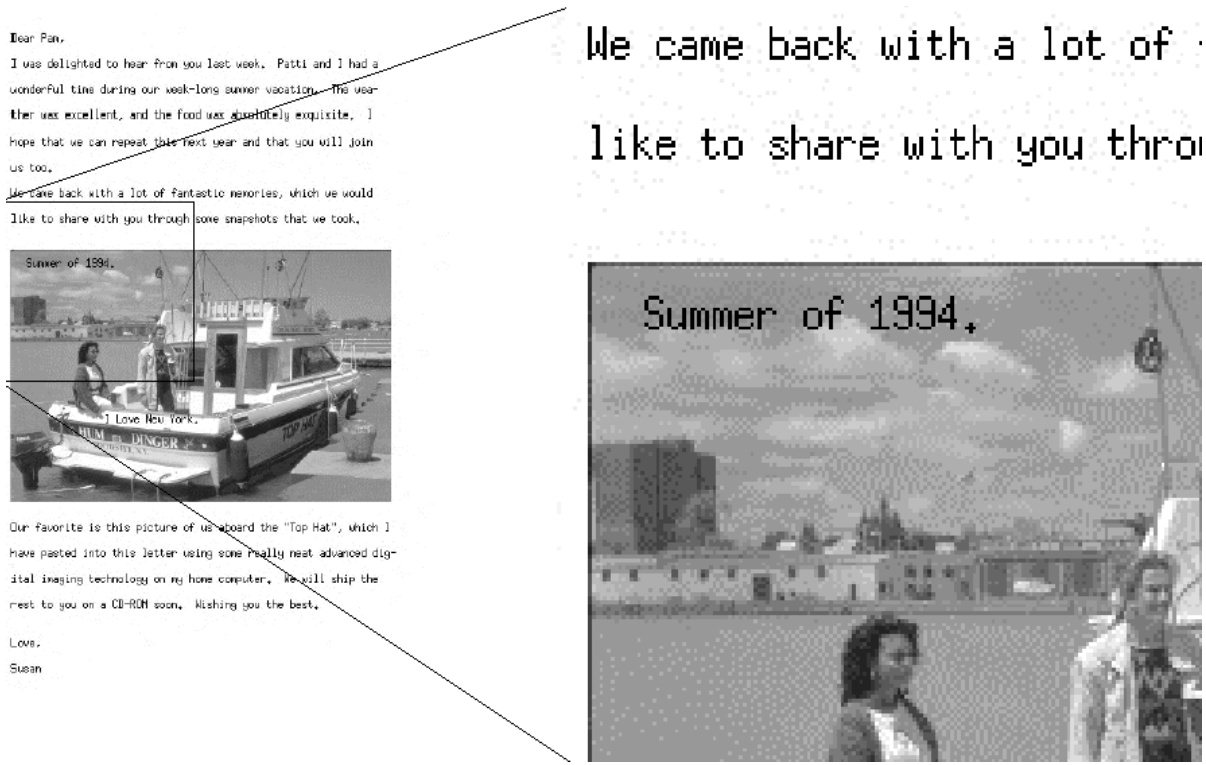
JPEG2000 - esempi (5)

JPEG immagine composta 1.0 bpp



JPEG2000 - esempi (5)

JPEG 2000 immagine composta 1.0 bpp



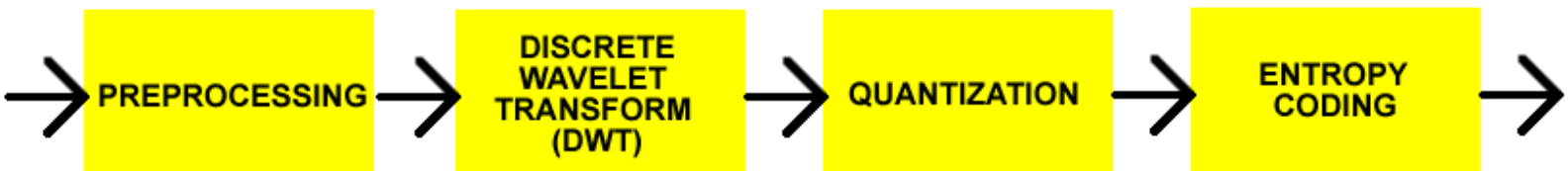
Descrizione dell'algoritmo

JPEG2000

- JPEG usa la trasformata chiamata DCT (Discrete Cosine Transform), JPEG 2000 usa la cosiddetta **DWT**, Discrete Wavelet Transform.

JPEG2000 - architettura

Schema di codifica di base



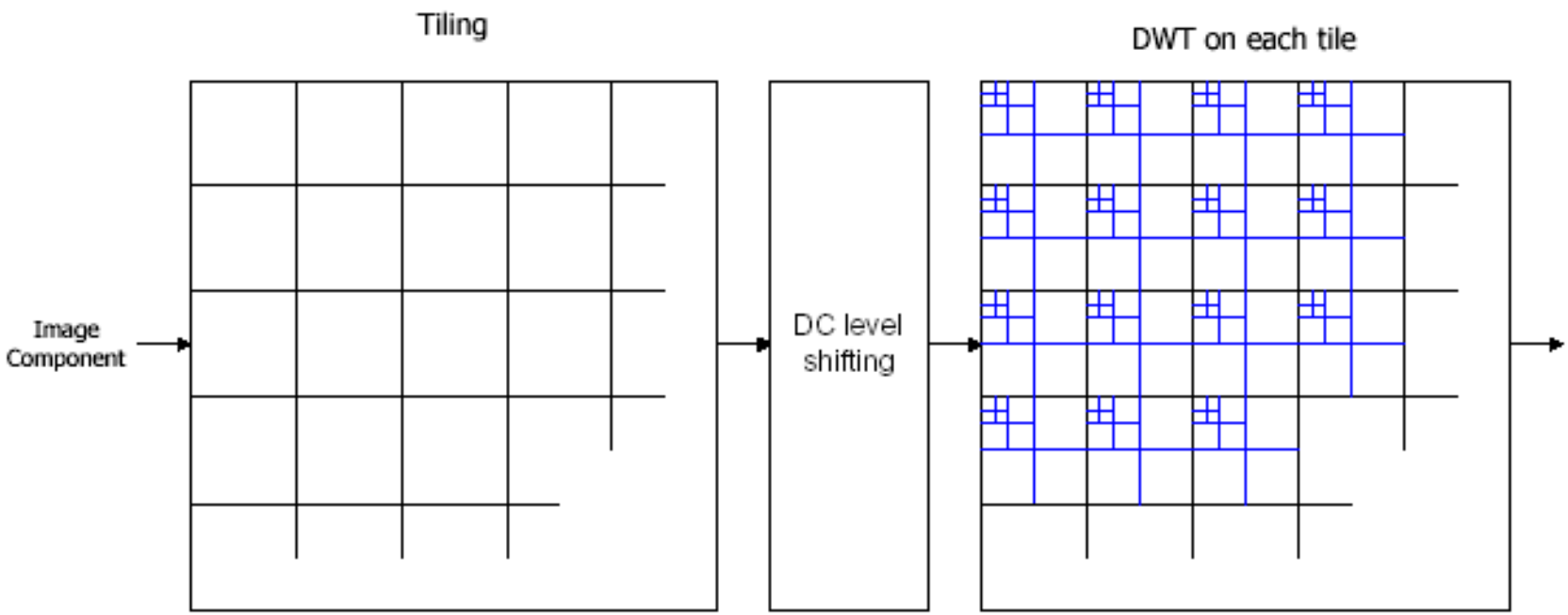
JPEG2000 – architettura (2)

- Preprocessing

- l'immagine viene suddivisa in componenti
- Ogni componente viene divisa in blocchi uguali rettangolari (*tiling*) e non sovrapposti che vengono codificati separatamente
- I campioni *unsigned* in ogni componente vengono sottratti ad una costante in modo da renderne i valori simmetrici rispetto allo zero (*DC offset*)

JPEG2000 – architettura (3)

- Preprocessing (2)



JPEG2000 – architettura (4)

- Preprocessing (3)

- La suddivisione in blocchi consente di ridurre la complessità di calcolo

- Ogni blocco viene gestito separatamente consentendo la decodifica di parti specifiche dell'immagine

JPEG2000 – architettura (5)

- **Discrete wavelet transform**

JPEG2000 – architettura (6)

- Discrete wavelet transform (2)

- La trasformazione wavelet suddivide i blocchi in diversi livelli di decomposizione

- Ogni livello di decomposizione consiste in un numero di sottobande, ognuna delle quali contiene i coefficienti che descrivono le frequenze spaziali del blocco originale

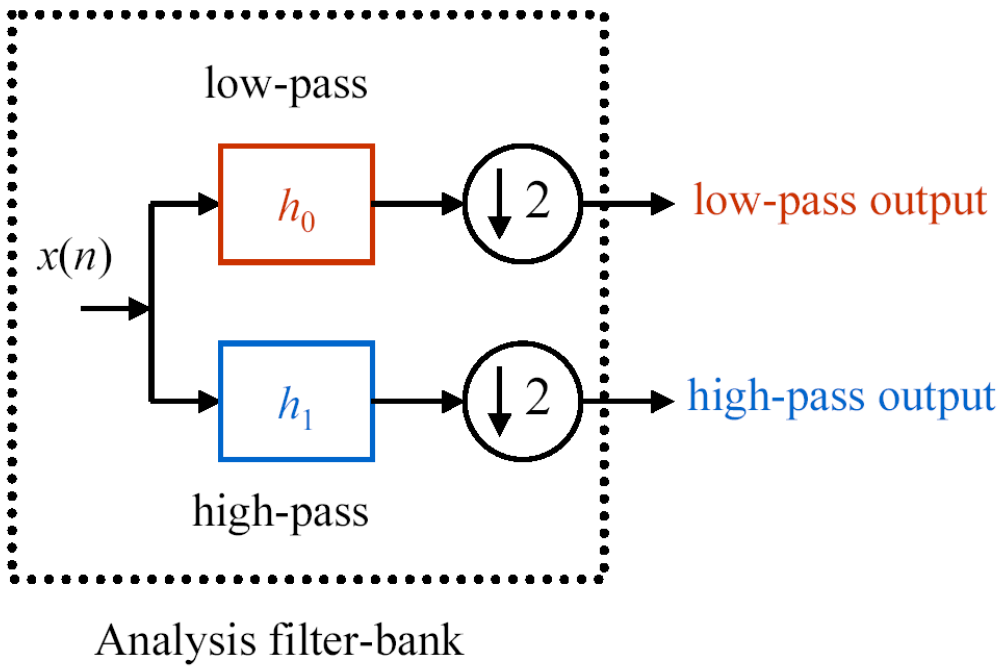
JPEG2000 – architettura (7)

- Discrete wavelet transform (3): caso 1D

- La DWT decompone una sequenza 1D (una linea di un'immagine) in due sequenze (le sottobande), ognuna delle quali ha un numero di campioni pari alla metà di quella originale
- La sequenza viene separata tramite un filtro passa-basso e uno passa-alto (analisys filter-bank)
- Le due sequenze vengono poi sottocampionate
- Ci sono due tipi di filtraggio: convolution-based e lifting-based

JPEG2000 – architettura (8)

- Discrete wavelet transform (4): caso 1D



JPEG2000 – architettura (9)

- Discrete wavelet transform (5): caso 1D

- 1-D signal:

  …100 100 100 100 200 200 200 200...

- Low-pass filter h_0: (-1 2 6 2 -1)/8
- High-pass filter h_1: (-1 2 -1)/2
- Before downsampling:

  … 100 100 87.5 112.5 187.5 212.5 200 200...

  … 0 0 0 -50 50 0 0 0…

- After downsampling:

  … 100 112.5 212.5 200...

  … 0 0 50 0 …

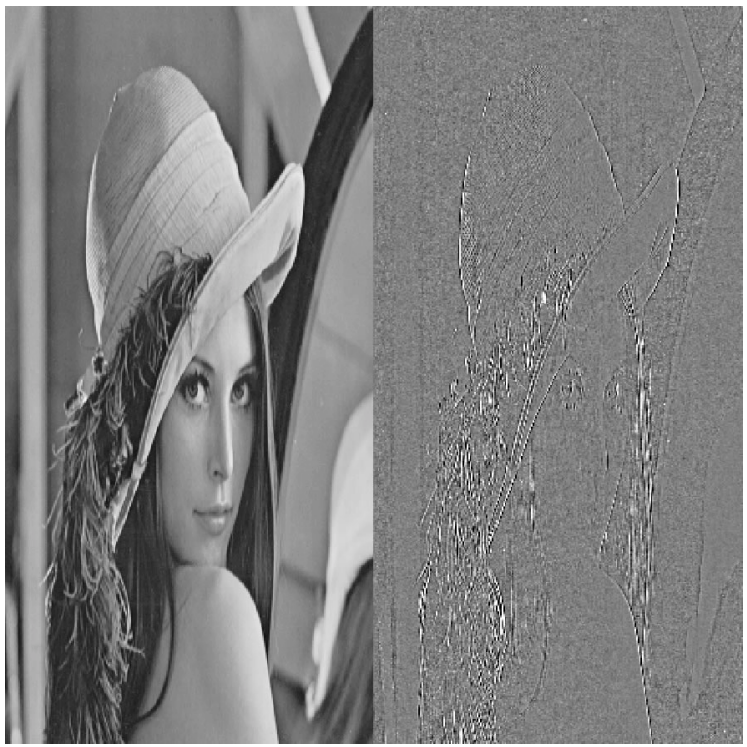
JPEG2000 – architettura (10)

- Discrete wavelet transform (6): caso 1D

Immagine originale Immagine passa-basso Immagine passa-alto

JPEG2000 – architettura (11)

- Discrete wavelet transform (7): caso 1D



JPEG2000 – architettura (12)

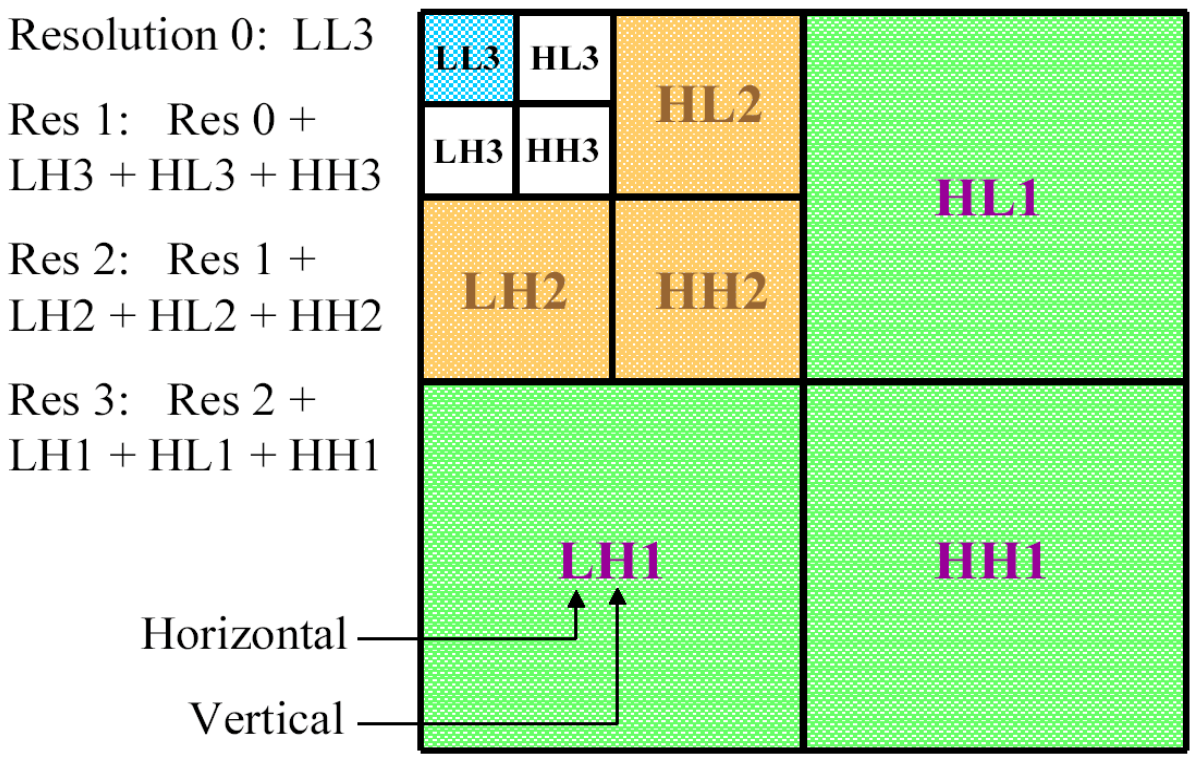
- Discrete wavelet transform (8): caso 2D

- L'immagine viene filtrata e poi scalata, in modo da ottenere quattro immagini alte e larghe ciascuna esattamente la metà dell'originale

- Il risultato di quest'operazione è la decorrelazione tra le informazioni di bassa e alta frequenza contenute nell'immagine

JPEG2000 – architettura (13)

- Discrete wavelet transform (9): caso 2D



Dyadic decomposition

JPEG2000 – architettura (14)

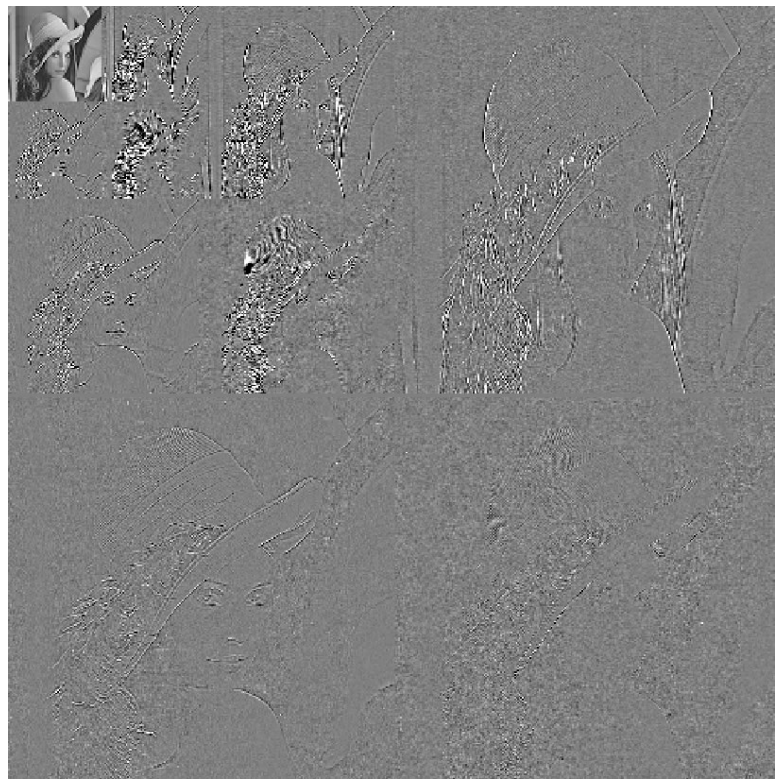
- Discrete wavelet transform (10): caso 2D

- Nel quadrante superiore sinistro sono salvate le basse frequenze (filtro passa-basso).
- Negli altri tre quadranti sono salvate le alte frequenze (filtro passa-basso)

JPEG2000 – architettura (15)

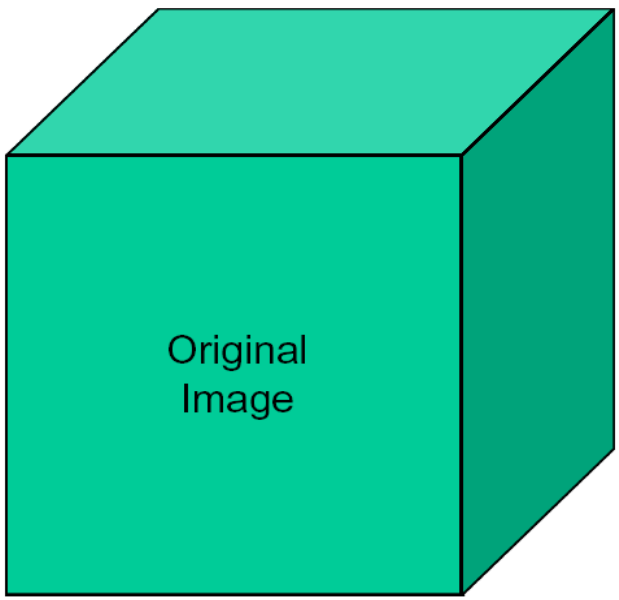
- Discrete wavelet transform (11): caso 2D

- Il passaggio successivo consiste nella ripetizione del medesimo procedimento, applicato stavolta solo all'immagine del quadrante superiore sinistro, contenente le basse frequenze



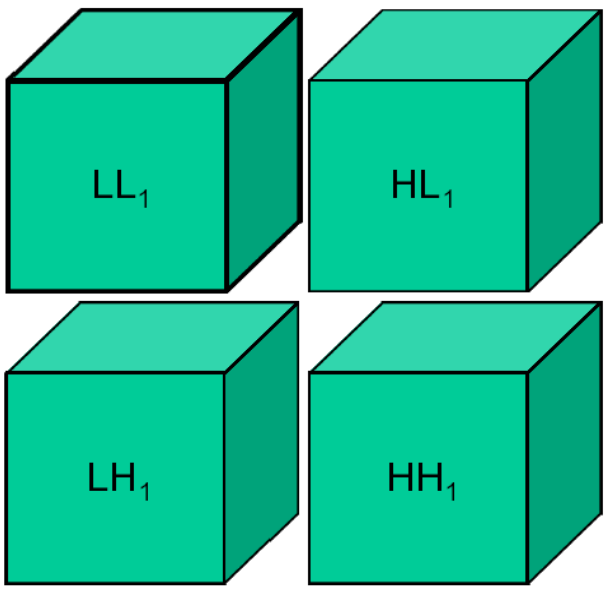
JPEG2000 – architettura (16)

- Discrete wavelet transform (12): caso 2D



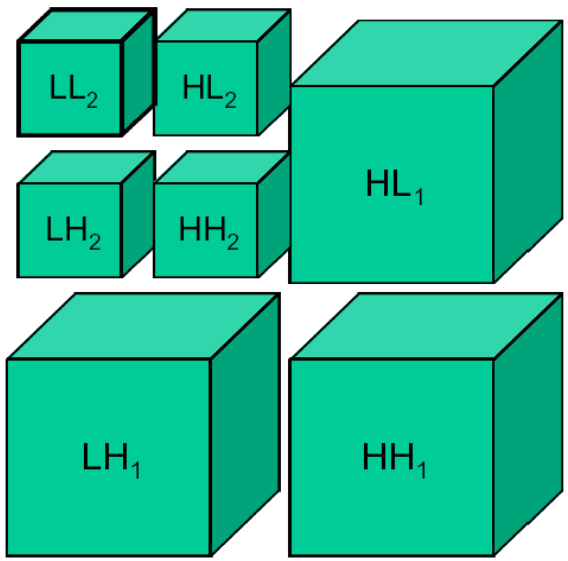
JPEG2000 – architettura (17)

- Discrete wavelet transform (13): caso 2D



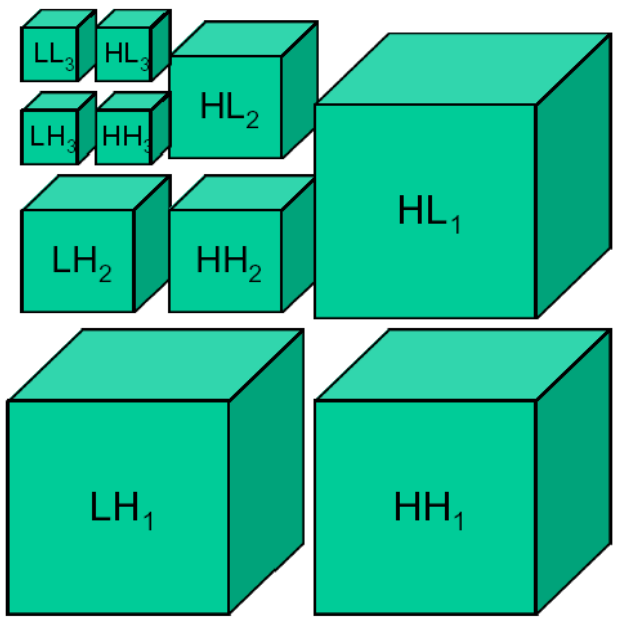
JPEG2000 – architettura (18)

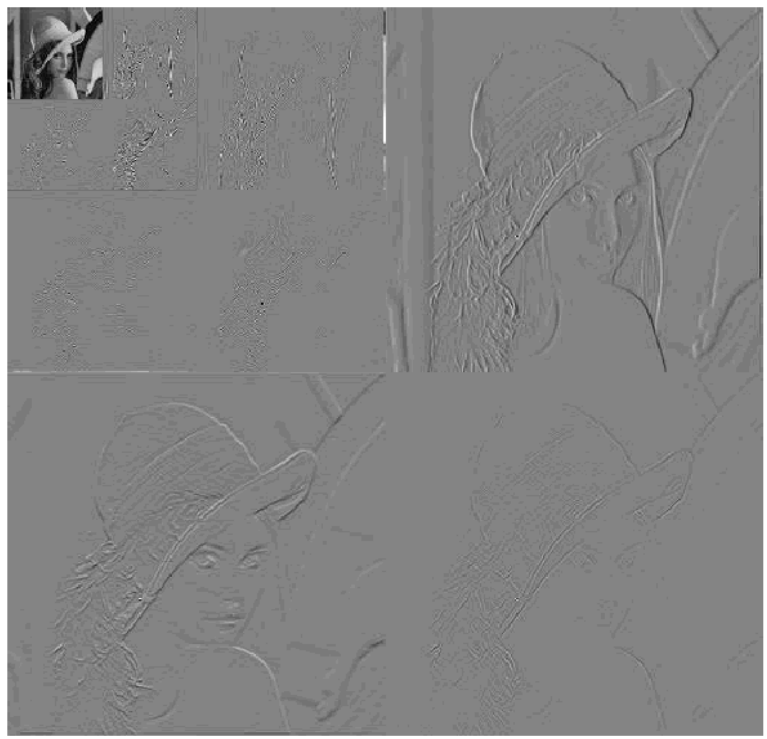
- **Discrete wavelet transform (14): caso 2D**



JPEG2000 – architettura (19)

- **Discrete wavelet transform (15): caso 2D**





JPEG2000 – architettura (20)

- Discrete wavelet transform (16): caso 2D

- Il risultato finale di tutta l'operazione è che l'intero contenuto informativo dell'immagine originale è stato segmentato in una serie di trasformazioni successive, che potranno poi essere compresse in un minimo spazio ed utilizzate in modo reversibile in fase di decompressione, per generare un'immagine il più possibile simile all'originale non compresso.

JPEG2000 – architettura (21)

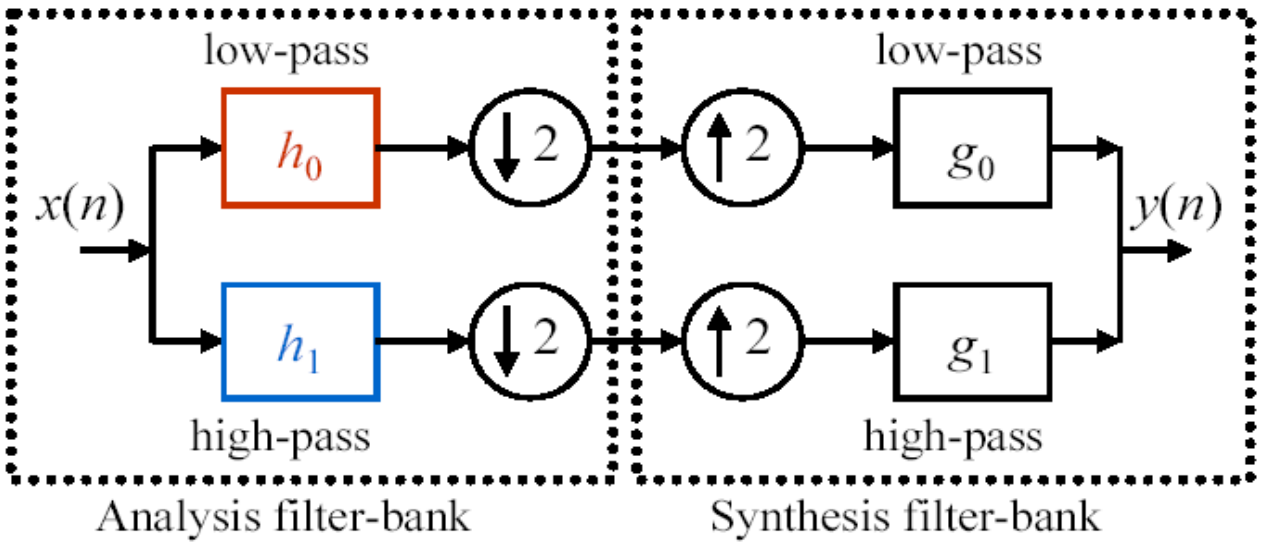
- Discrete wavelet transform (17): filtri

- Alcuni filtri supportati:
  1) W9x7 (floating point, lossy);
  2) W5x3 (integer, lossless);

JPEG2000 – architettura (22)

- Discrete wavelet transform (18): filtri

- La maggiorparte dei sistemi di compressione basati sulla wavelet utilizzano filtri bi-ortogonali;



h0 ortogonale a g1 h1 ortogonale a g0

JPEG2000 – architettura (23)

- Discrete wavelet transform (19): filtri

- Ci sono due casi:

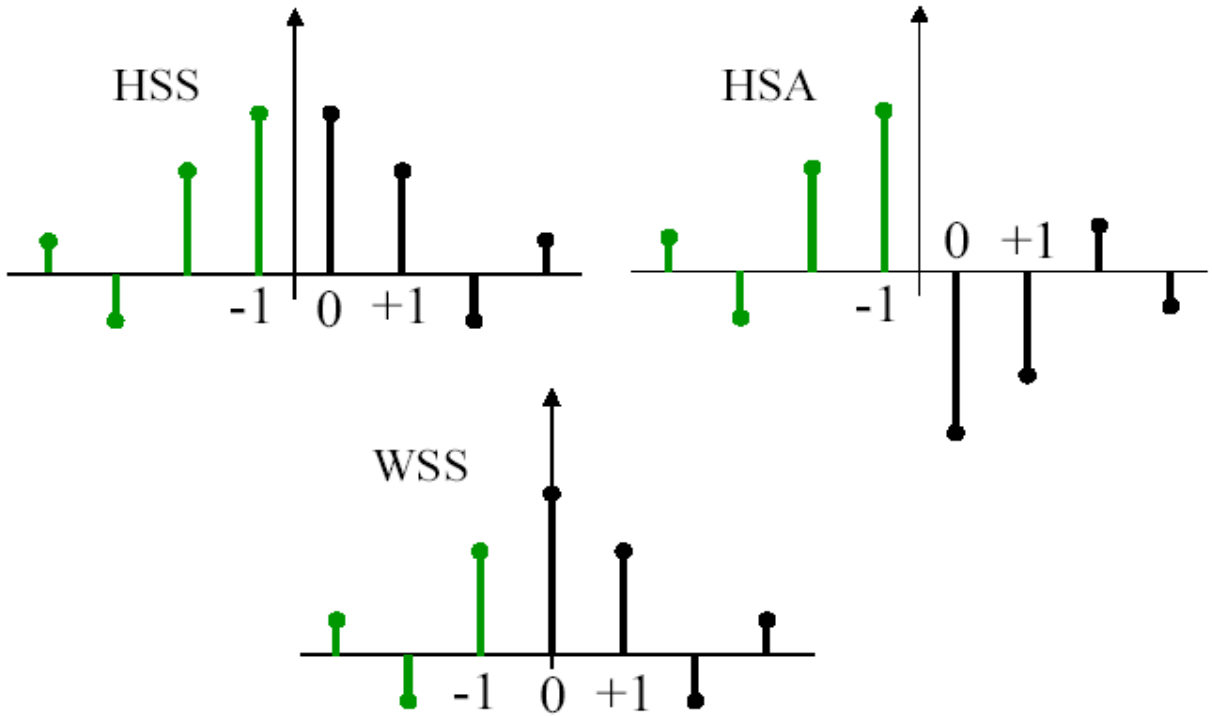
  1) ho e h1 sono *odd-length,* quindi simmetrici e vengono chiamati filtri **Whole-sample symmetric** (**WSS**);

  2) ho e h1 sono even-*length,* quindi h0 è simmetrico e h1 antisimmetrico e vengono chiamati rispettivamente filtri **Half-sample symmetric** (**HSS**) e **Half-sample anti symmetric** (**HSA**);

- Le condizioni di simmetria fanno si che solo la metà dei coefficienti dei fltri debba essere trasmessa

JPEG2000 – architettura (24)

- Discrete wavelet transform (20): filtri



JPEG2000 – architettura (25)

- Discrete wavelet transform (21): normalizzazione filtri

- Affinchè l'immagine venga decodificata in maniera corretta i filtri di sintesi e quelli di analisi devono rispettare la relazione:

$$g_0(n) = \alpha\,(-1)^n h_1(-n)$$
$$g_1(n) = \alpha\,(-1)^n h_0(-n)$$

JPEG2000 – architettura (26)

- Discrete wavelet transform (22): normalizzazione filtri

⌘ Coefficiente di normalizzazione:

$$\alpha = \frac{2}{\left(\sum_n h_0(n)\right)\left(\sum_n (-1)^n h_1(n)\right) + \left(\sum_n h_1(n)\right)\left(\sum_n (-1)^n h_0(n)\right)}$$

Dove:

$\left|\sum_n h_0(n)\right|$ DC gain del filtro passa-basso

$\left|\sum_n (-1)^n h_1(n)\right|$ Nyquist gain del filtro passa-alto

JPEG2000 – architettura (27)

Discrete wavelet transform (23): esempi di filtri

Filtro Daubechies (9,7) normalizzato con un DC gain di uno e un Nyquist gain di due (JPEG 2000)

n	$h_0(n)$	n	$h_1(n)$
0	+0.602949018236	-1	+1.115087052456
±1	+0.266864118442	-2, 0	-0.591271763114
±2	-0.078223266528	-3, 1	-0.057543526228
±3	-0.016864118442	-4, 2	+0.091271763114
±4	+0.026748757410		

JPEG2000 – architettura (28)

Discrete wavelet transform (24): esempi di filtri

Filtro Daubechies (10,18)

n	$h_0(n)$	$h_1(n)$
-1, 0	+0.536628801791	+0.440781829293
-2, 1	+0.054299075394	-0.115519002860
-3, 2	-0.111388018824	-0.060571607153
-4, 3	+0.000058297264	+0.009733420188
-5, 4	+0.020401844374	+0.021802742673
-6, 5		+0.001787592313
-7, 6		-0.006683900685
-8, 7		+0.000001928418
-9, 8		+0.000674873932

JPEG2000 – architettura (29)

Discrete wavelet transform (25): esempi di filtri

Filtro Daubechies (6,10)

n	$h_0(n)$	$h_1(n)$
-1, 0	+0.557543526228	+0.869813136679
-2, 1	+0.033728236885	-0.188640851913
-3, 2	-0.091271763114	-0.095087384971
-4, 3		-0.009884638967
-5, 4		+0.026748757410

JPEG2000 – architettura (30)

Discrete wavelet transform (26): esempi di filtri

Filtro Integer (2,10)

n	$h_0(n)$	$h_1(n)$
-1, 0	+1/2	+1
-2, 1		-22/128
-3, 2		-22/128
-4, 3		+3/128
-5, 4		+3/128

JPEG2000 – architettura (31)

Discrete wavelet transform (27): esempi di filtri

Filtri integer CRF (13,7) e Swelden (13,7)

n	$h_0(n)$ (SWE)	$h_0(n)$ (CRF)	n	$h_1(n)$
0	+348/512	+164/256	-1	+1
±1	+144/512	+80/256	-2, 0	-9/16
±2	-63/512	-31/256	-3, 1	0
±3	-6/512	-16/256	-4, 2	+1/16
±4	+18/512	+14/256		
±5	0	0		
±6	-1/12	-1/256		

JPEG2000 – architettura (32)

Discrete wavelet transform (28): esempi di filtri

Filtri integer (5,3) e (2,6)

n	$h_0(n)$	n	$h_1(n)$
0	+6/8	-1	+1
±1	+2/8	-2, 0	-1/2
±2	-1/8		

n	$h_0(n)$	$h_1(n)$
-1, 0	+1/2	+1
-2, 1		-1/128
-3, 2		-1/128

JPEG2000 – architettura (33)

Discrete wavelet transform (29): L2-norms

- Nella compressione Jpeg lo scaling della DCT è tale da creare una trasformazione ortonormale. In questo modo l'MSE (Mean-squared error) nel dominio dello spazio uguale a quello nel dominio trasformato. IN questo modo è possibile quantificare l'impatto della quantizzazione dei coefficienti sull'immagin ricostruita

- Questo non è valido invece per la DWT

JPEG2000 – architettura (34)

Discrete wavelet transform (30): L2-norms

- Per i coefficienti della DWT, sotto certe condizioni, l'MSE dell'immagine ricostruita può essere espresso come somma pesata degli MSE dei coefficienti della DWT, dove il peso per ogni sottobanda *b* è:

Dove:

$$\alpha^2{}_b = \sum_n \left|\psi^b(n)\right|^2$$

α_b Coefficiente L2-norm per la sottobanda *b*

ψ^b Funzioni base per i coefficienti wavelet per la sottobanda *b*

JPEG2000 – architettura (35)

Discrete wavelet transform (31): L2-norms

- La conoscenza della L2-norm è fondamentale per l'encoder perché rappresenta il contributo del rumore di quantizzazione di ogni sottobanda
- In funzione della L2-norm è possibile progettare quantizzatori ad hoc

JPEG2000 – architettura (36)

Discrete wavelet transform (32): L2-norms

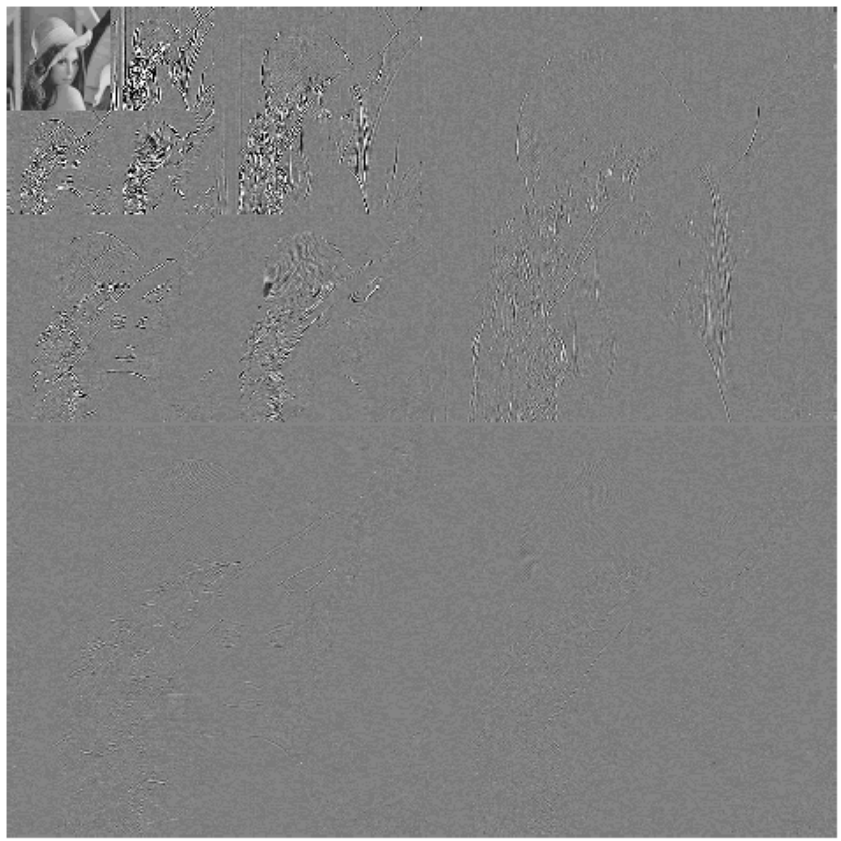
Subband L2-norms dopo 2D, 3 Level DWT

	$(\sqrt{2},\sqrt{2})$ Normalization		(1,2) Normalization	
Subband	(5,3)	(9,7)	(5,3)	(9,7)
3LL	0.67188	1.05209	5.37500	8.41675
3HL	0.72992	1.04584	2.91966	4.18337
3LH	0.72992	1.04584	2.91966	4.18337
3HH	0.79297	1.03963	1.58594	2.07926
2HL	0.79611	0.99841	1.59222	1.99681
2LH	0.79611	0.99841	1.59222	1.99681
2HH	0.92188	0.96722	0.92188	0.96722
1HL	1.03833	1.01129	1.03833	1.01129
1LH	1.03833	1.01129	1.03833	1.01129
1HH	1.43750	1.04044	0.71875	0.52022

JPEG2000 – architettura (37)

Discrete wavelet transform (33): L2-norms

3 Level DWT con Filtro (9,7) scalato con L2-norm



JPEG2000 – architettura (38)

Discrete wavelet transform (34): complessità

- La complessità della DWT dipende da:
  - dimensioni dei filtri;
  - tipo di filtro (floating vs integer)

- Tranne in casi particolari (filtri integer (5,3)) la DWT è computazionalmente più complessa (2-3x) della block-based DCT

- Essendo Full-frame, la DWT richiede una memoria maggiore rispetto alla DCT.

- Implementazioni line-based e lifting-based permettono una riduzione della memoria

JPEG2000 – architettura (39)

Discrete wavelet transform (36): lifting scheme

- E’ un metodo alternativo per il calcolo dei coefficienti wavelet
- Costo computazionale minore
- Richiede meno memoria
- Può essere facilmente adattato per produrre trasformazioni lossless
- La trasformazione diretta e quella inversa hanno la stessa complessità

JPEG2000 – architettura (40)

Discrete wavelet transform (37): algoritmo lifting

- Tre step:

  1) split step: il segnale originale X_k viene diviso in due sottosequenze (pari e dispari). *Lazy wavelet transform:*

$$s_i^0 \rightarrow x_{2i}, \qquad d_i^0 \rightarrow x_{2i+1},$$

JPEG2000 – architettura (41)

Discrete wavelet transform (38): algoritmo lifting

2) lifting step: in realtà sono *N sub-step*, dove le sequenze vengono trasformate tramite i coefficienti *prediction* **P** e *update* **U**, i cui valori dipendono dai filtri wavelet usati

$$d_i^{\,n} = d_i^{\,n-1} + \sum P_n(k) \times s_k^{\,n-1}, \qquad n \in [1,2,\cdots,N]$$

$$s_i^{\,n} = s_i^{\,n-1} + \sum U_n(k) \times d_k^{\,n}, \qquad n \in [1,2,\cdots,N]$$

JPEG2000 – architettura (42)

Discrete wavelet transform (39): algoritmo lifting

3) normalization step: vengono applicati fattori di normalizzazione per ottenere i coefficienti wavelet

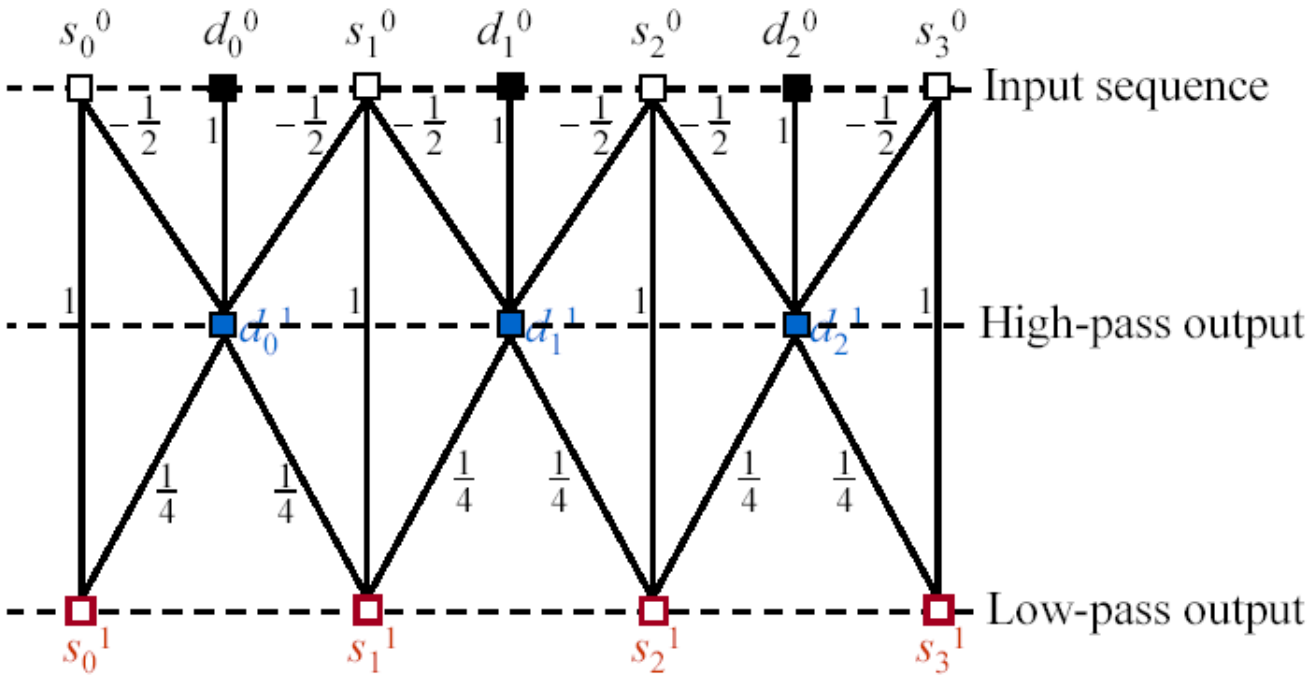
$$s_i^N \rightarrow K_0 \, s_i^N, \qquad d_i^N \rightarrow K_1 \, d_i^N.$$

JPEG2000 – architettura (43)

Discrete wavelet transform (40): esempio lifting

filtro (5,3)

N=1

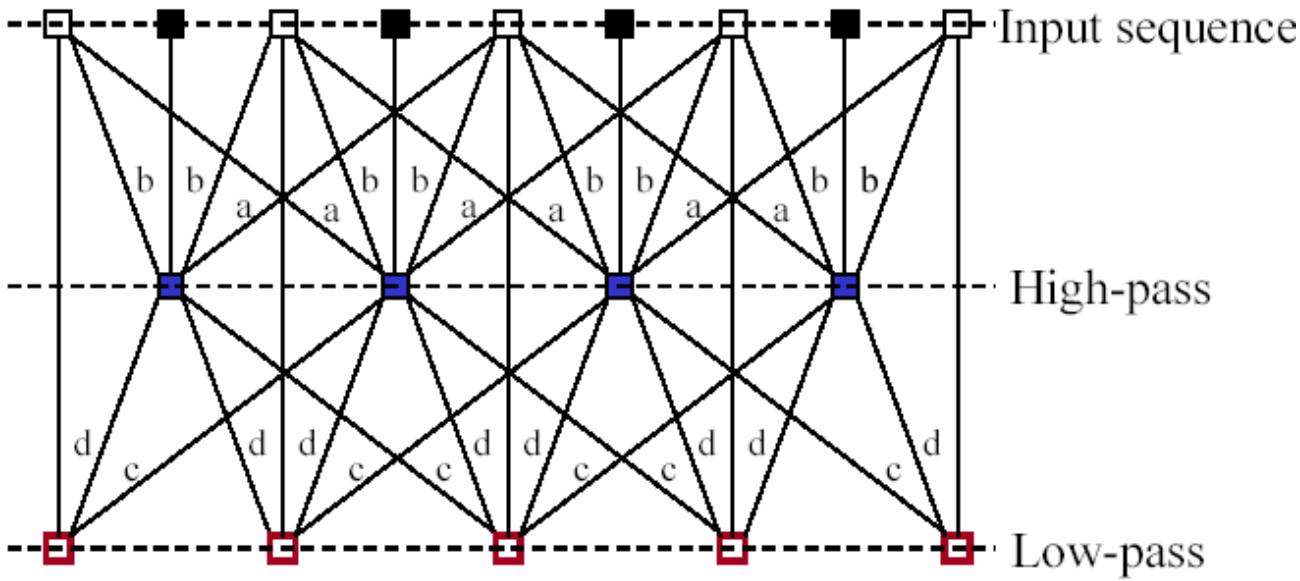


$$d_i^1 = d_i^0 - \tfrac{1}{2}(s_i^0 + s_{i+1}^0), \quad s_i^1 = s_i^0 + \tfrac{1}{4}(d_{i-1}^1 + d_i^1)$$

JPEG2000 – architettura (44)

Discrete wavelet transform (41): esempio lifting

filtro (13,7)
integer
struttura
lattice

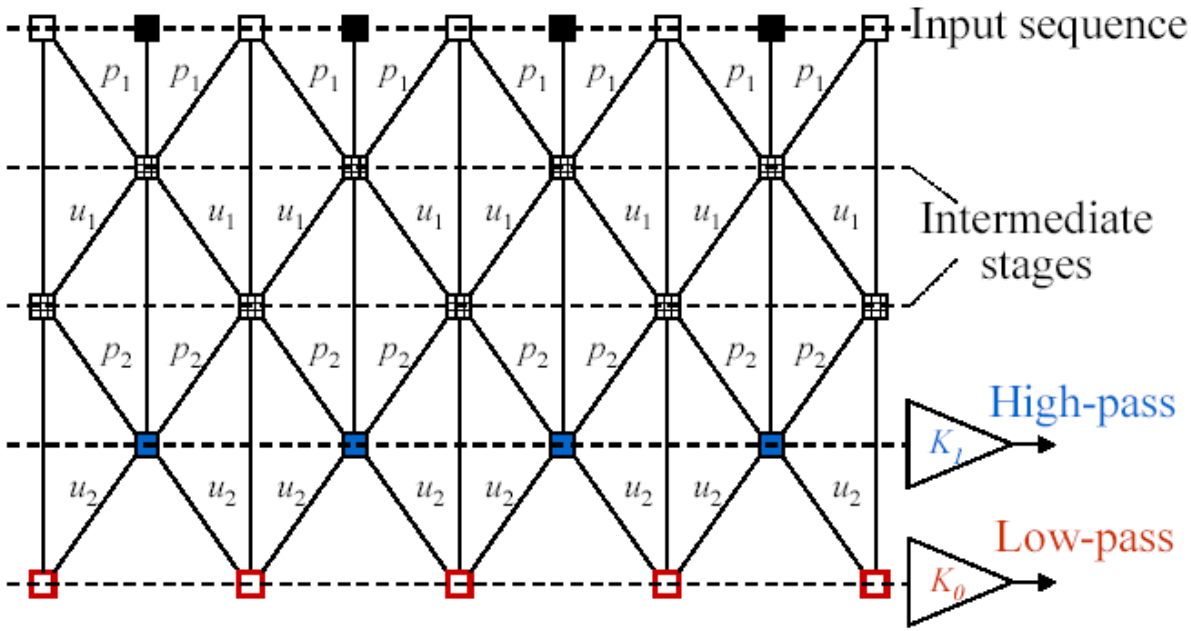


- CRF (13 x 7): a = 1/16, b = -9/16, c = -1/16, d = 5/16
- SWE (13 x 7): a = 1/16, b = -9/16, c = -1/32, d = 9/32

JPEG2000 – architettura (46)

Discrete wavelet transform (42): esempio lifting

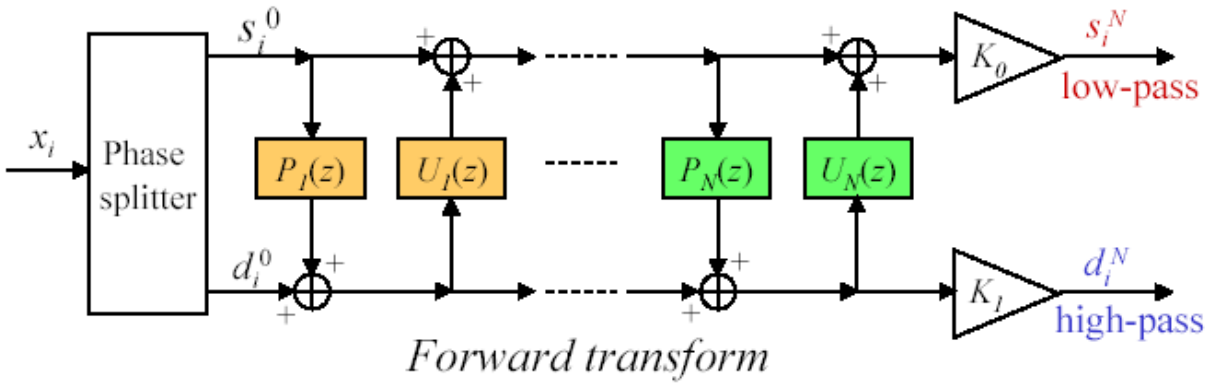
filtro (9,7)



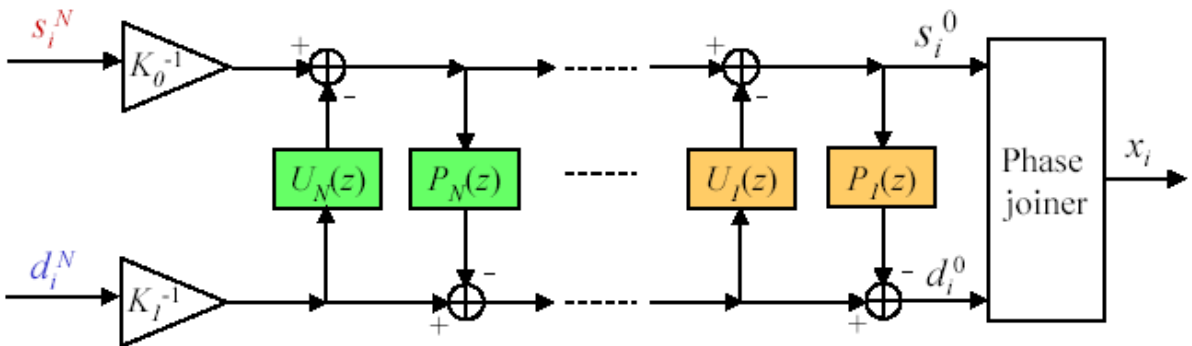
$p_1 = -1.586134342,$	$u_1 = -0.052980118$
$p_2 = +0.882911075,$	$u_2 = +0.443506852$
$K_0 = 1/K_1,$	$K_1 = 1.230174104$

JPEG2000 – architettura (47)

Discrete wavelet transform (43): lifting diagram



Forward transform



Inverse transform

JPEG2000 – architettura (48)

Discrete wavelet transform (44): JPEG 2000 parte 1

- La parte 1 dello standard JPEG 2000 ha scelto solo due tipi di filtri per la DWT:

  1) Daubechies (9,7) floating point, performante per la compressione lossy;

  2) Lifted integer (5,3), lossless, bassa complessità

JPEG2000 – architettura (49)

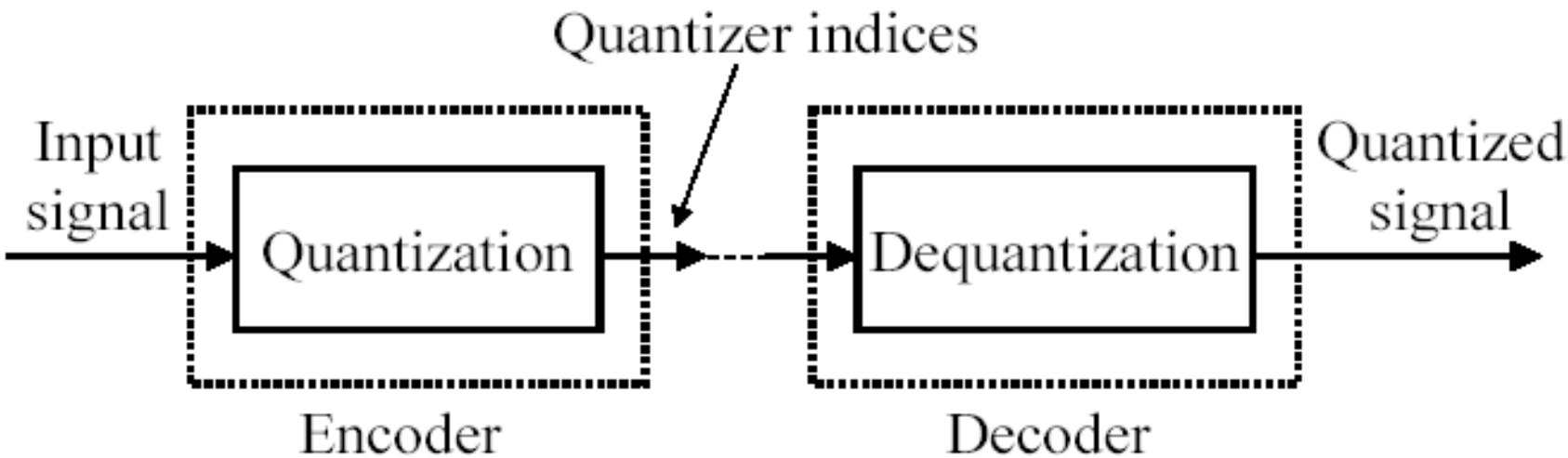
Quantizzazione:

- La quantizzazione è un'operazione in genere "distruttiva" (lossy), che divide per specifici coefficienti i valori trovati dai coefficienti della DWT e arrotonda poi i risultati all'intero più vicino

- La quantizzazione nel JPEG 2000 è simile a quella del JPEG standard

JPEG2000 – architettura (50)

Quantizzazione (2): schema



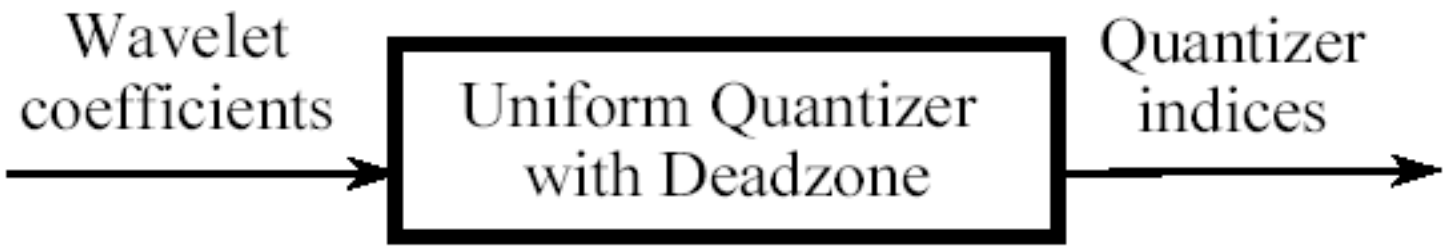
JPEG2000 – architettura (51)

Quantizzazione (3): quantizzazione e dequantizzazione

- In codifica: l'operazione di quantizzazione mappa un segnale su un indice del quantizzatore, che viene poi codificato come parte del bit stream
- In decodifica: l'indice del quantizzatore viene decodificato e convertiro nel corrispondente valore quantizzato (operazione di quantizzazione inversa);

JPEG2000 – architettura (52)

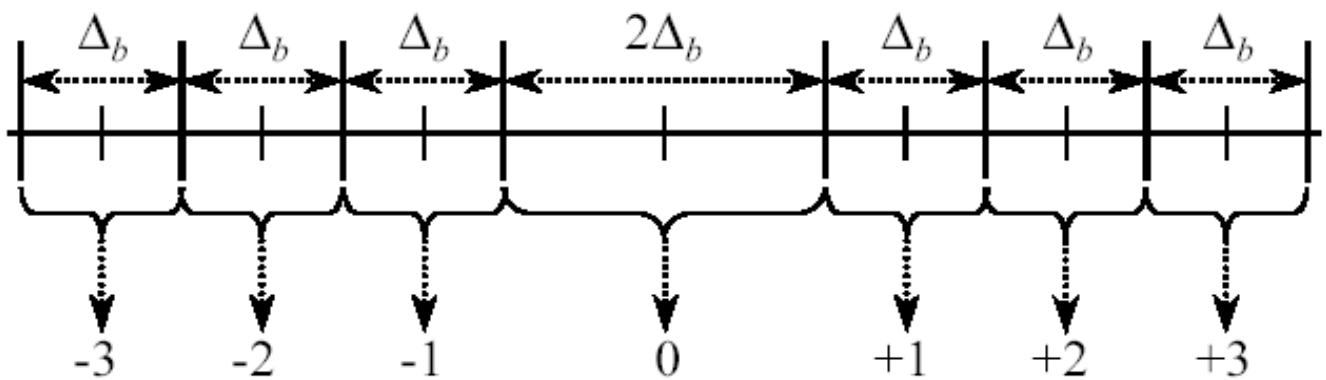
Quantizzazione (4): schema di quantizzazione



JPEG2000 – architettura (53)

Quantizzazione (5): tipologia

- I coefficienti DWT vengono elaborati tramite una quantizzazione uniforme con *deadzone* centrale



JPEG2000 – architettura (54)

Quantizzazione (6):

- Per ogni sottobanda *b* viene scelto un passo di quantizzazione Δ_b usato per quantizzare tutti i coefficienti in quella sottobanda

- La scelta del passo dipende dall'importanza percettiva di ogni sottobanda e può essere basata su modelli visuali conosciuti

JPEG2000 – architettura (55)

- Quantizzazione (7): indice del quantizzatore

- Ogni coefficiente ***a*** della sottobanda ***b*** viene quantizzato al valore ***q*** secondo la formula:

$$q_b(u,v) = \mathrm{sign}(a_b(u,v)) \left\lfloor \frac{|a_b(u,v)|}{\Delta_b} \right\rfloor$$

q è quindi l'indice del quantizzatore

JPEG2000 – architettura (56)

- Quantizzazione (8): passo di quantizzazione

- Il passo di quantizzazione Δ_b viene rappresentato con un totale di 2 byte, una mantissa μ_b di 11 bit e un esponente ε_b di 5 bit, secondo la relazione:

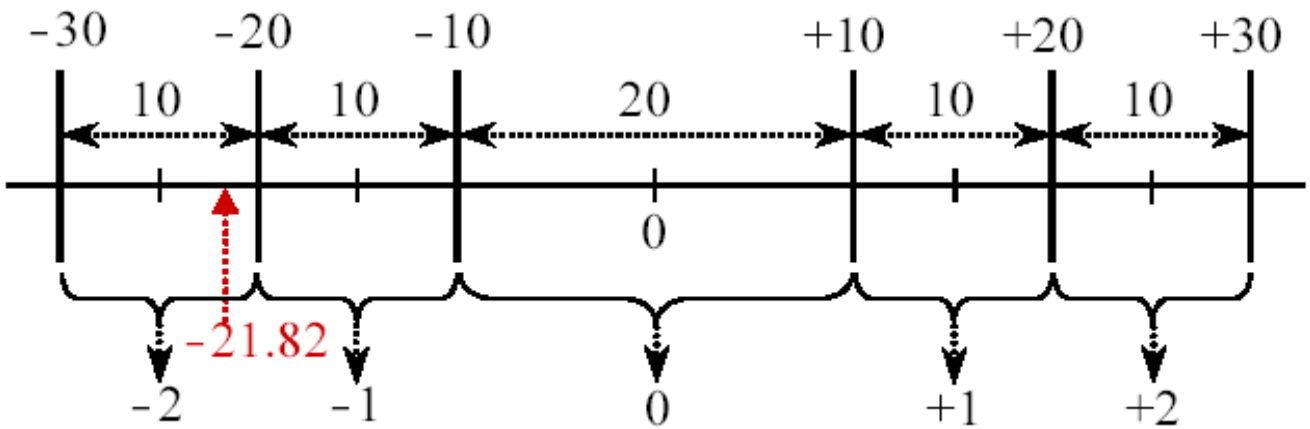
$$\Delta_b = 2^{R_b - \varepsilon_b}\left(1 + \frac{\mu_b}{2^{11}}\right)$$

Dove R_b è il numero di bit che rappresenta il range dinamico nominale della sottobanda *b*

JPEG2000 – architettura (57)

- Quantizzazione (9): esempio

- Encoder input value = -21.82
- Quantizer index = - ⌊21.82/10⌋ = -2



JPEG2000 – architettura (58)

- Quantizzazione (10): quantizzazione inversa

⌘ Regola di ricostruzione:

$$z = [q + r\ \mathrm{sign}(q)]\Delta_b\ , \quad \text{for } q \neq 0$$
$$z = 0, \quad \text{otherwise}$$

Dove: *q* è l'indice del quantizzatore, Δ_b è il passo di quantizzazione
z è il segnale ricostruito ed *r* un parametro di ricostruzione

JPEG2000 – architettura (59)

- Quantizzazione (11): quantizzazione inversa

⌘ Il parametro di ricostruzione assume i valori:

1) 0.5 per la ricostruzione midpoint, come nel JPEG standard

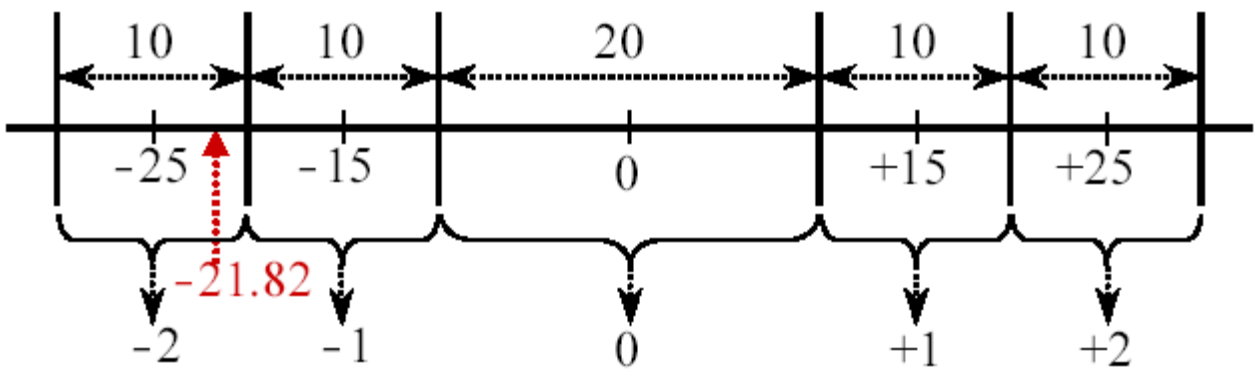
2) <0.5 per una ricostruzione verso lo zero (miglior PSNR)

Nel JPEG 2000 viene usato un valore pari a 0.375

JPEG2000 – architettura (60)

- Quantizzazione (12): esempio quantizzazione inversa

- Quantizer index = -2
- Reconstructed value $r = 0.5$ (midpoint): $(-2-0.5)\times 10 = -25$
- Reconstructed value $r = 0.375$: $(-2-0.375)\times 10 = -23.75$



JPEG2000 – architettura (61)

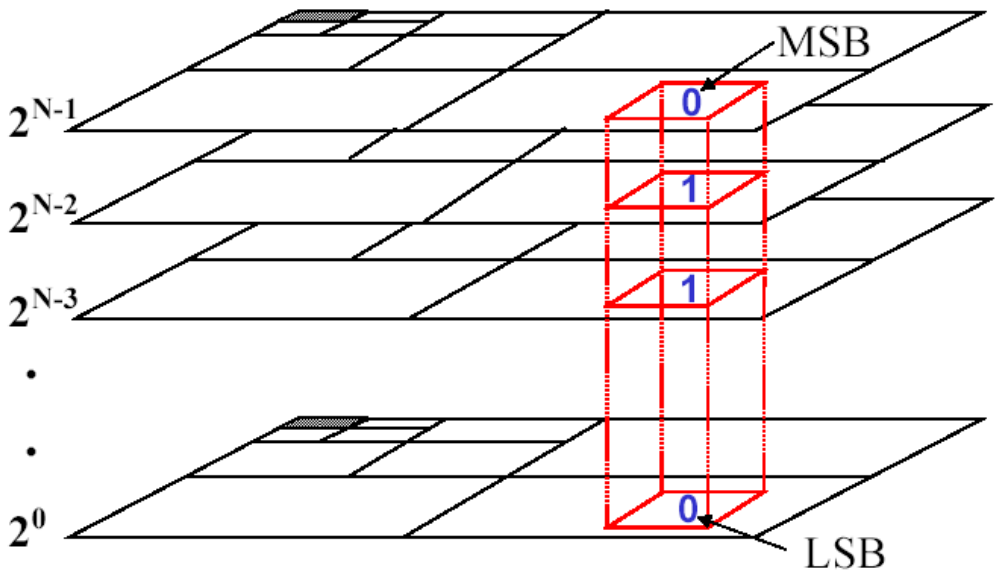
Entropy coding:

- Gli indici di quantizzazione, ovvero i coefficienti DWT quantizzati in ogni sottobanda, vengono codificati in modo da creare il flusso dei dati compressi

JPEG2000 – architettura (62)

Entropy coding (2): bit-plane coding

- Gli indici di quantizzazione vengono elaborati secondo una codifica di tipo bit-plane



JPEG2000 – architettura (63)

Entropy coding (3): bit-plane coding

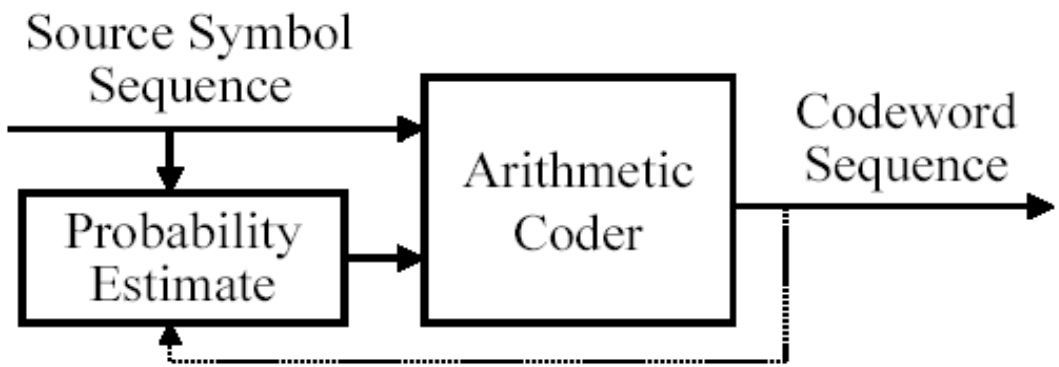
- I simboli che rappresentano i coefficienti quantizzati vengono codificati 1 bit per volta iniziando da MSB fino a LSB

- Lo standard JPEG 2000 usa un metodo efficiente per la codifica della ridondanza dei bit-plane, il cosiddetto context-based adaptive binary arithmetic coding

JPEG2000 – architettura (64)

Entropy coding (4): codifica aritmetica

- La codifica aritmetica può essere vista come un dispositivo che accetta in ingresso una sequenza di simboli e le corrispondenti stime statistiche producendo in uscita un code stream con lunghezza uguale alle lunghezze combinate ideali dei simboli in ingresso



JPEG2000 – architettura (65)

Entropy coding (5): codifica aritmetica

- In generale la distribuzione di probabilità di ogni simbolo binario in un coefficiente wavelet quantizzato dipende da tutti i precedenti bit compressi corrispondenti al coefficiente e a quelli vicini

JPEG2000 – architettura (64)

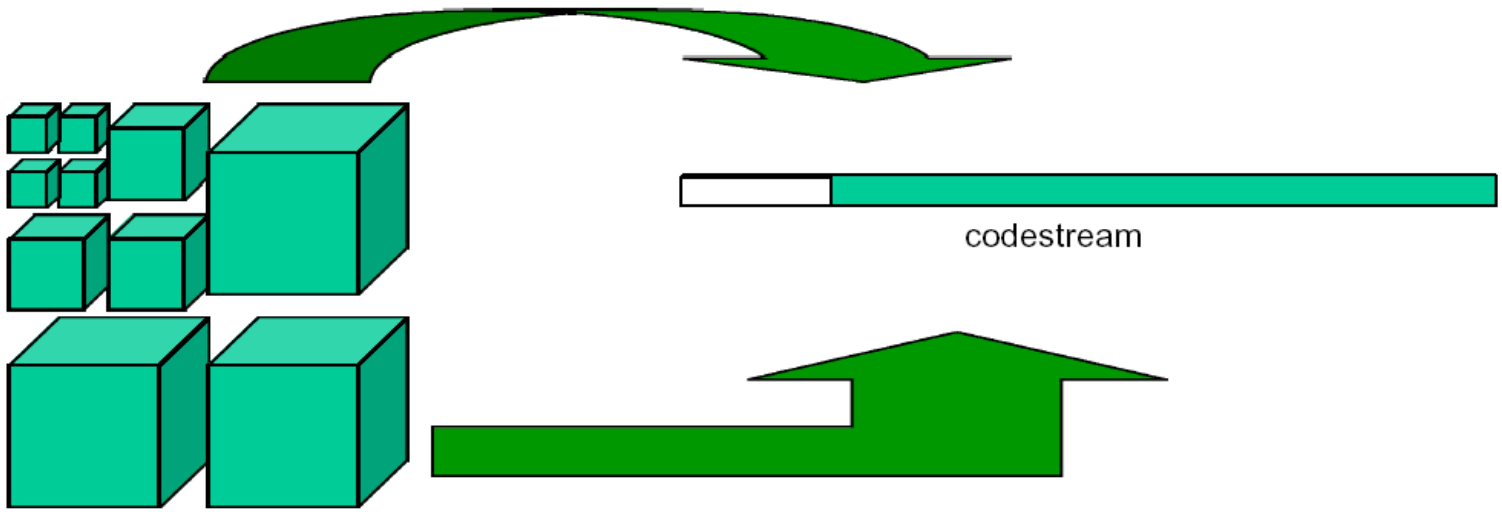
Entropy coding (4): codifica aritmetica

- La codifica aritmetica permette di ottenere un efficienza pari alla codifica di Huffmann per blocchi grandi, ma codificando un solo simbolo per volta
- Implementazioni pratiche usate nel JPEG 2000: Q-coder

JPEG2000 – caratteristiche

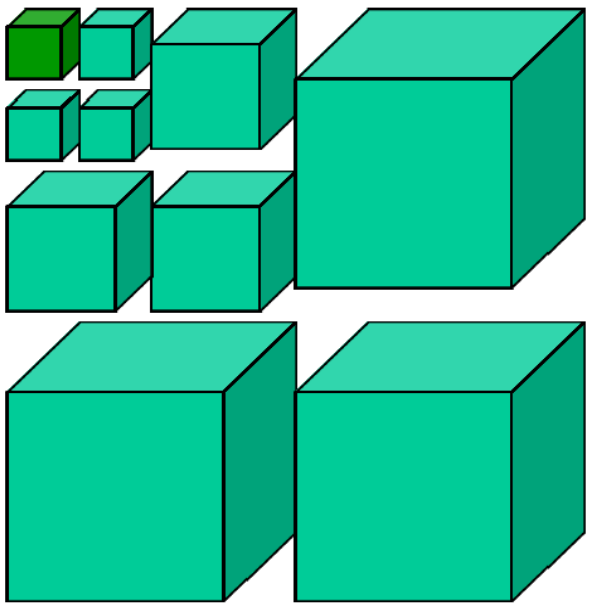
Scalabilità

- **Modi diversi possono essere implementati a seconda della costruzione del codestream**



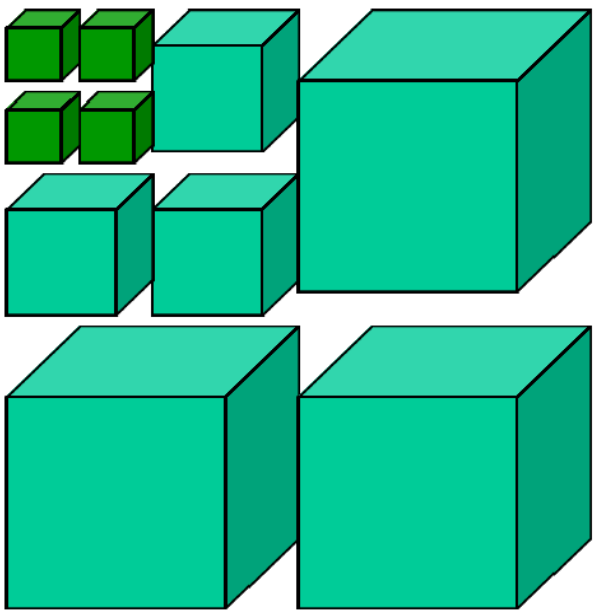
JPEG2000 – caratteristiche (2)

Scalabilità (2): progressiva per risoluzione



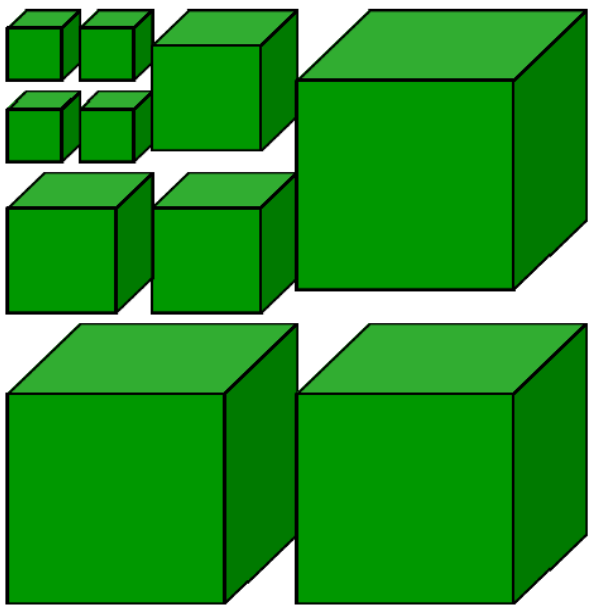
JPEG2000 – caratteristiche (3)

Scalabilità (3): progressiva per risoluzione



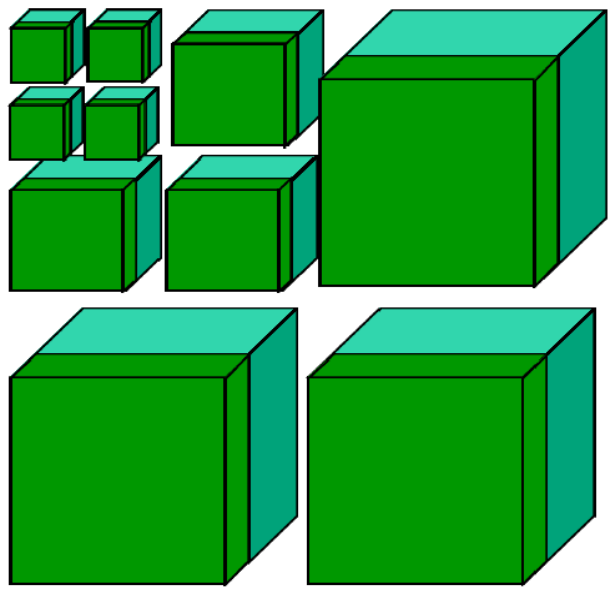
JPEG2000 – caratteristiche (4)

Scalabilità (4): progressiva per risoluzione



JPEG2000 – caratteristiche (5)

Scalabilità (5): progressiva per accuratezza



JPEG2000 – caratteristiche (6)

Scalabilità (6): progressiva per risoluzione

- Image: Woman
- Resolution levels: 5
- Decoded sizes: 1/16
  1/8
  1/4
  1/2
  1

JPEG2000 – caratteristiche (7)

Scalabilità (7): progressiva per risoluzione

JPEG2000 – caratteristiche (8)

Scalabilità (8): progressiva per risoluzione

JPEG2000 – caratteristiche (9)

Scalabilità (9): progressiva per risoluzione

JPEG2000 – caratteristiche (10)

Scalabilità (10): progressiva per risoluzione

JPEG2000 – caratteristiche (11)

Scalabilità (11): progressiva per risoluzione

JPEG2000 – caratteristiche (12)

- Una simile caratteristica è importantissima per i futuri usi di questo formato su Internet. I file compressi con JPEG 2000 racchiudono infatti al loro interno **più risoluzioni differenti** della stessa immagine.

- I produttori di software potranno implementare perciò dei comandi in grado di consentire all'utente collegato via Internet di decidere, in base al tempo stimato per il download, quale risoluzione dell'immagine visualizzare nel proprio browser.

- I file **JP2** (è questa l'estensione proposta per identificare lo standard JPEG 2000) consentiranno insomma di racchiudere in un unico documento l'anteprima – il cosiddetto thumbnail – la bassa, la media e l'alta risoluzione di una stessa immagine, senza però moltiplicare proporzionalmente il peso del file.

JPEG2000 – caratteristiche (13)

ROI: region of interest coding

- Questa tecnica permette di codificare certe parti dell'immagine in funzione della qualità

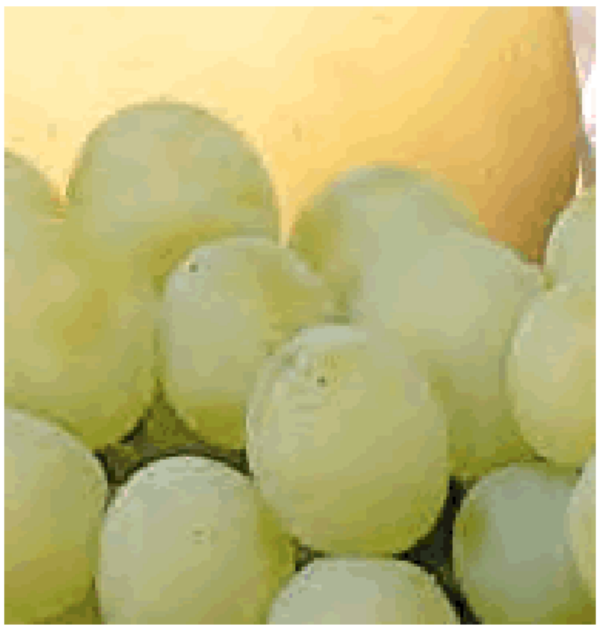
- Due tipi:

  1) Statica: la ROI viene decisa una volta per tutte dal lato della codifica

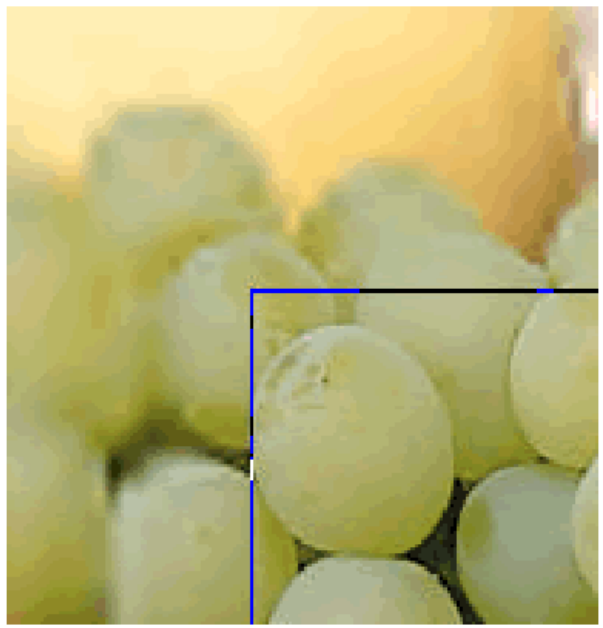
  2) Dinamica: la ROI può essere decisa e decodificata al volo da uno stesso bitstream

JPEG2000 – caratteristiche (14)

ROI (2): esempio



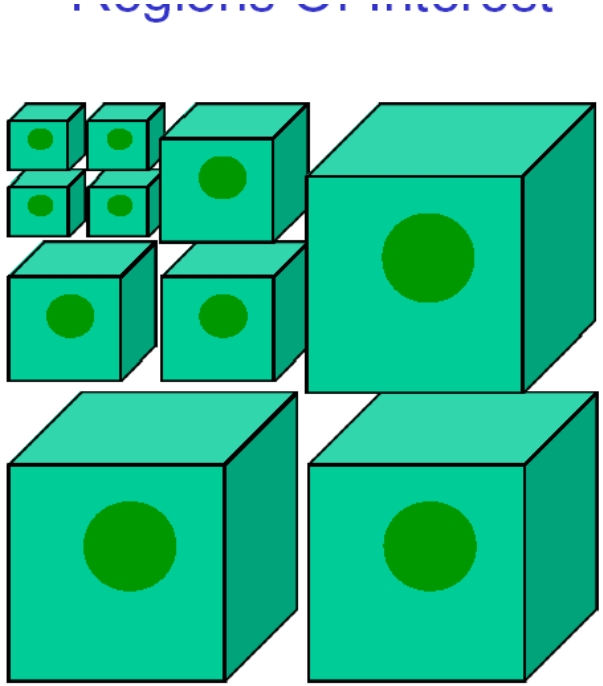
No ROI

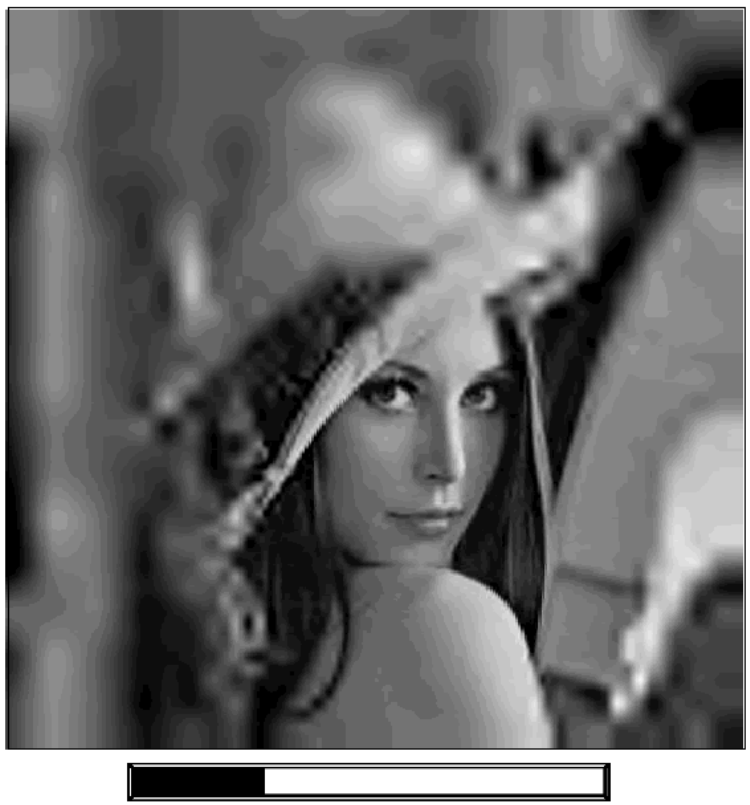


Rectangular ROI

JPEG2000 – caratteristiche (15)

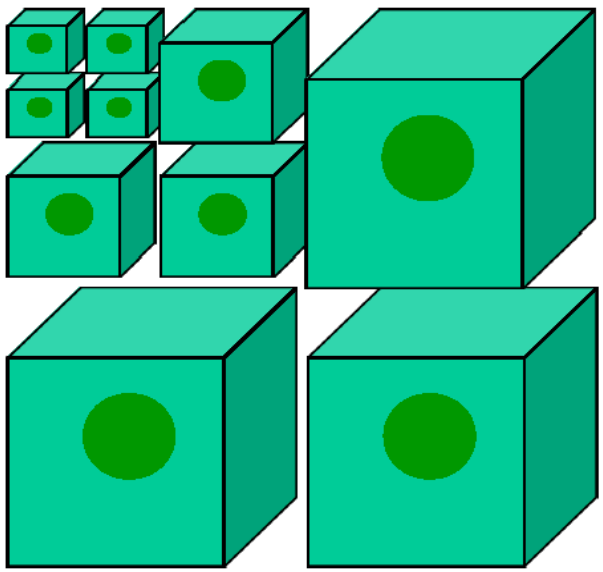
ROI (3): esempio





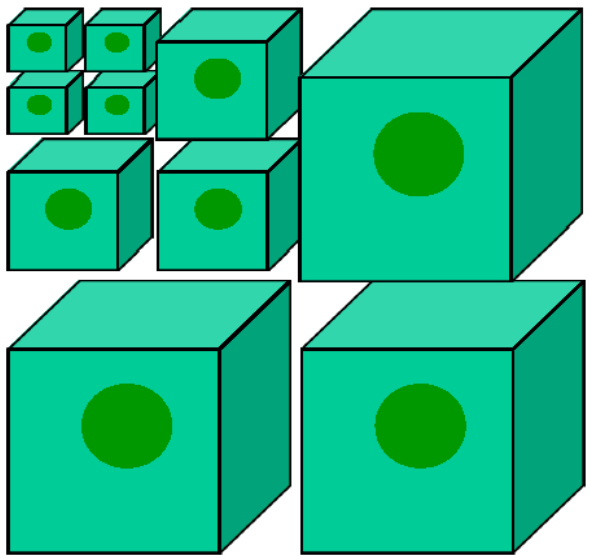
JPEG2000 – caratteristiche (16)

ROI (4): esempio



JPEG2000 – caratteristiche (17)

ROI (5): esempio



JPEG2000 – Specifiche previste

- supporto per differenti modalità e spazi-colore (immagini a due toni, in scala di grigi, a 256 colori, a milioni di colori, in standard RGB, PhotoYCC, CIELAB, CMYK);
- supporto per differenti schemi di compressione, adattabili in base alle esigenze;
- standard aperto a successive implementazioni legate al sorgere di nuove necessità;
- supporto per l'inclusione di un'illimitata quantità di metadati nello spazio di intestazione del file, utilizzabili per fornire informazioni private o per interagire con applicazioni software (guidare, ad esempio, il browser allo scaricamento di appositi plug-in da Internet);

JPEG2000 – Specifiche previste

- stato dell'arte per la compressione distruttiva e non distruttiva delle immagini, con un risparmio di spazio a parità di qualità, rispetto allo standard JPEG, dell'ordine del 20-30 %;
- supporto per immagini più grandi di 64k x 64k pixel, ovvero maggiori di 4 Gb;
- singola architettura per la decompressione dei file, in luogo dei 44 modi codificati per il vecchio JPEG, molti dei quali legati a specifiche applicazioni e non utilizzati dalla maggior parte dei decompressori;

JPEG2000 - Esempio

- Per vedere in anteprima i risultati della compressione wavelet scaricare **LuraWave**, un programma sviluppato da LuraTech (http://www.luratech.com)

- Nella **Figura** qui sotto riportata, si può vedere la differenza tra due immagini fortemente compresse, l'una con LuraWave l'altra con il normale JPEG, differenza che appare a tutto vantaggio della compressione di tipo wavelet.

PNG

PNG

- PNG (pronunciato ping) è l'acronimo di Portable Network Graphics.
- La questione nasce con l'annuncio fatto nel 1995 da CompuServe e Unisys: da quel momento in poi, l'implementazione del formato GIF in prodotti software avrebbe comportato il pagamento di una quota-diritti ad Unisys, quale legale detentore del brevetto sull'algoritmo di compressione LZW, usato all'interno del formato GIF.
- PNG nasce quindi in contrapposizione a GIF, come un formato grafico compresso e gratuito.

PNG - Caratteristiche

- Compressione
- Controllo di errore
- Supporto per milioni di colori
- Canali alfa
- Interlacciamento
- Correzione di gamma

PNG - Compressione

- PNG punta tutto sulla compressione non distruttiva.
- I risultati che si ottengono sono in genere del 20% migliori di quelli ottenibili con la compressione GIF.
- Lo strumento di compressione, gratuitamente utilizzabile da chiunque, è **zlib** - una variante dell'algoritmo LZ77 -, sviluppato per la parte di compressione da Jean-loup **Gailly** (http://gailly.net/) e per quella di decompressione da Mark **Adler** (http://www.alumni.caltech.edu/~madler/), ed attualmente giunto alla versione 1.1.3.

PNG - Compressione

- La capacità di compressione del PNG può in certi casi essere aumentata grazie all'adozione di particolari **filtri**: si tratta di sistemi di trasformazione dell'ordine dei dati che costituiscono l'immagine, studiati per esaltare il coefficiente di compressione raggiungibile.

- Vi è il caso limite di un'immagine che, non compressa, occupa 48 Mb, compressa con PNG ma non filtrata occupa 36 Mb, filtrata, infine, si riduce a soli 115.989 byte.

PNG – Controllo di errore

- PNG dispone di un sistema chiamato CRC-32, ovvero *cyclic redundancy check* ("controllo di ridondanza ciclico") a 32 bit, che associa valori di controllo ad ogni blocco di dati ed è in grado di rilevare immediatamente qualsiasi corruzione delle informazioni salvate o trasmesse via Internet.

PNG – Supporto per milioni di colori

- Le immagini PNG non sono limitate ad un massimo di 256 colori come accade per i file GIF, ma supportano anche la modalità RGB. Non supportano per ora la modalità CMYK, anche se si prevede la possibilità di una futura estensione che ne consenta il trattamento.

PNG – Canali alfa

- Mentre GIF supporta una trasparenza del tipo “tutto o niente” (pixel completamente trasparente o del tutto opaco), PNG permette, grazie all'uso dei cosiddetti canali alfa, la possibilità di una trasparenza variabile su 254 livelli di opacità. Questa caratteristica favorisce, ad esempio, la creazione del classico effetto di ombre cadute, ombre che si conservano indipendentemente dal colore dello sfondo su cui vengono visualizzate.

PNG – Interlacciamento

- In caso di connessioni via modem particolarmente lente, può essere molto utile cominciare a vedere fin da subito un'anteprima, sia pure poco definita, dell'immagine che si sta scaricando.
- Con la funzione di interlacciamento di cui è dotato il formato PNG ciò è possibile molto più velocemente che con l'analoga funzione incorporata nel formato GIF.
- L'interlacciamento - cioè la visualizzazione parziale e progressiva dell'immagine – in GIF è monodimensionale e la prima apparizione di contenuti visibili richiede l'invio di un ottavo dei dati-immagine complessivi. L'interlacciamento di PNG è invece bidimensionale e basta l'invio di solo un sessantaquattresimo dei dati per avere la prima visualizzazione grossolana dell'immagine

PNG – Correzione di gamma

- Il formato PNG integra un sistema di correzione di gamma, che permette di compensare, sia pure solo approssimativamente, le differenze di visualizzazione di un'immagine nel passaggio da una piattaforma hardware ad un'altra.

PNG – Esempio

- Il retro di una banconota da 10 euro: L'immagine, salvata come PNG, ottiene un coefficiente di compressione migliore di circa il 22% rispetto alla compressione LZW di TIFF (in questo esempio l'immagine è una normale JPG)