



COMPRESSIONE DI IMMAGINI ASPETTI TEORICI ED APPLICATIVI

Ennio Ottaviani

On AIR srl

Piazza Campetto 2/10, 16123 Genova

Email

ennio.ottaviani@ge.onairweb.com

INDICE DEGLI ARGOMENTI

Introduzione

Cenni di teoria delle comunicazioni

Algoritmi di codifica senza perdite

Immagini digitali (campionamento e quantizzazione)

Algoritmi di codifica con perdite

Trasformata di Karhunen-Loewe

La DCT ed il suo ruolo nella compressione

Lo standard JPEG

Algoritmi di codifica di sottobanda

Alcuni esempi di applicazioni

COMPRESSIONE DI IMMAGINI

Cosa è:

Un insieme di algoritmi matematici volti a ridurre il più possibile il numero di bits (o quanti di informazione) utilizzati per rappresentare una immagine digitale (ma anche altri tipi di dati discreti).

Motivazioni:

- Riduzione dello spazio (memoria) necessario per l'archiviazione
- Riduzione della banda (o del tempo) necessario per la trasmissione

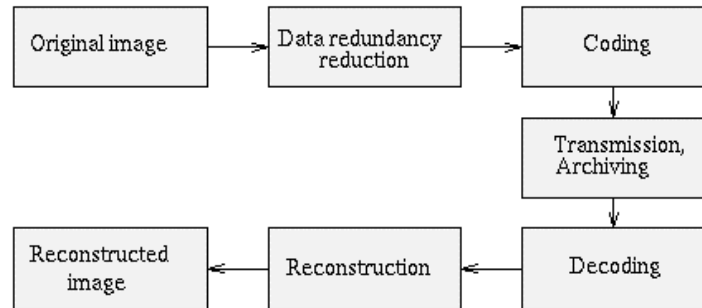
Applicazioni

- Biomedicale
- Documentale
- Telerilevamento
- Multimedia
- Telefonia
- Internet

Categorie

- Compressione senza perdite (*lossless*)
Il dato non viene alterato ma solo codificato in modo più efficiente
- Compressione con perdite (*lossy*)
Il dato viene anche modificato in modo impercettibile, allo scopo di ottenere efficienza ancora maggiore

ARCHITETTURA DI BASE



Esempi

- Immagine di qualità fotografica
 - Dimensione: 4000x4000
 - Numero di bande: 3
 - Totale: 48 Mbyte
- Immagine satellitare (es. Landsat)
 - Dimensione: 6000x6000
 - Numero di bande: 7
 - Totale: 260 Mbytes
- Videoconferenza
 - Dimensione: 320x240
 - Numero di bande: 1
 - Immagini/secondo: 10
 - Totale: 768 Kbyte/secondo
- TV digitale
 - Dimensione: 640x480
 - Numero di bande: 3
 - Immagini/secondo: 25
 - Totale: 23 Mbyte/secondo

TEORIA DELLE COMUNICAZIONI (1)

Assunzione: La sorgente è descritta da un processo casuale $\{U_n\}$ discreto, stazionario ed ergodico, dove:

$$U_n \in A = \{a_1, a_2, \dots, a_N\}$$

Definizione: L'entropia (del 1° ordine) di una sorgente è definita da:

$$H_1(U) = - \sum P(u) \log_2 P(u)$$

dove $P(u) = P(U_n = u)$, $u \in A$. L'entropia si misura in bit/campione

Teorema: Considerata una sorgente U discreta ad entropia finita e senza memoria, allora non è possibile rappresentarla con un numero medio di bit/campione inferiore ad $H_1(U)$

Esempio: Se $A = \{a_1, a_2, a_3, a_4, a_5\}$ con $P(a_1) = 0,30$ $P(a_2) = 0,25$ $P(a_3) = 0,25$ $P(a_4) = 0,10$ $P(a_5) = 0,10$, allora $H_1(U) = 2,185$ bit/campione. Il codice qui sotto porta a una lunghezza media di 2,2 bit/campione.

a_1	0,30	00
a_2	0,25	01
a_3	0,25	10
a_4	0,10	110
a_5	0,10	111

TEORIA DELLE COMUNICAZIONI (2)

Il concetto di entropia è generalizzabile a gruppi di simboli

$$H_L(U) = H(U_1 U_2 \dots U_L) / L \quad \text{Entropia di ordine } L$$

Definizione

$$\text{Entropia limite} \quad H_\infty(U) = \lim_{L \rightarrow \infty} H_L(U)$$

$$\text{In generale:} \quad H_\infty(U) \leq H_1(U) \leq \log_2 N$$

Teorema

Per qualunque sorgente stazionaria ed ergodica, l'entropia limite rappresenta il minimo numero di bit necessari per rappresentare la sorgente (in assenza di rumore)

Per sorgenti senza memoria: $H_\infty(U) = H_1(U)$.

Se i simboli sono equiprobabili: $H_\infty(U) = \log_2 N$

Esistono quindi due meccanismi base per aumentare l'efficienza di una codifica:

- Sfruttare la disuniformità della distribuzione di probabilità dei simboli
- Sfruttare la correlazione esistente tra simboli successivi di una sequenza di simboli

Ammettendo la modifica della sequenza nasce una terza possibilità:

- Sfruttare il fatto che l'utente non percepisce il dato nella sua globalità ma solo le caratteristiche salienti

CODIFICA HUFFMANN

Obiettivo

Produrre per una data sorgente una codifica che approssimi l'entropia

Nota:

I codici devono essere decodificabili in modo univoco!

Esempio:

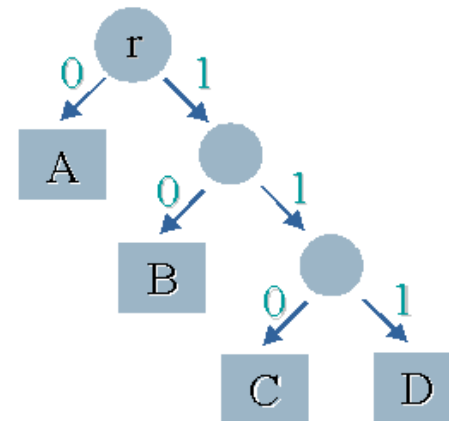
Simbolo	A	B	C	D
Codice	0	1	10	11

0110 \Rightarrow ABBA, ADA, ABC ?

La codifica Huffman costruisce un albero di simboli i passi seguenti:

1. valutazione della frequenza dei simboli
2. ordinamento decrescente
3. scelta della coppia a frequenza minore
4. creazione di un nodo
5. torna al passo 2 fino al completamento dell'albero
6. assegna i codici percorrendo l'albero

Simbolo	A	B	C	D
Frequenza	7	5	2	2
Codice	0	10	110	111



CODIFICA RUN-LENGTH (1)

E' un modo semplice di sfruttare la correlazione tra simboli successivi di una sequenza

Definizione

Run: sequenza di simboli identici $r = \{x_n, x_{n+k-1}\} : x_i = Q$

Approccio

Si trasforma la sequenza di simboli in sequenza di run (cioè di coppie simbolo-lunghezza)

Sorgenti binarie

Sono sorgenti di simboli a due valori, codificati con 0 ed 1. Ne segue un numero di bit/campione $R = 1$ (mentre l'entropia può essere < 1)

La trasformazione RL consente di ottenere $R < 1$ purchè la distribuzione delle lunghezze dei runs sia disuniforme.

CODIFICA RUN-LENGTH (2)



Esempio: CCITT gruppo 3 (fax)

Run Length	White	Black
1	00011	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
8	10011	000101
...

CODIFICA GIF

L'approccio RL è generalizzabile a sequenza non binarie, ma perde efficienza (più difficile trovare run con lunghezza > 1)

Variante: (Lempel, Ziv, Welch)

- Codifica basata su dizionario: si aggiungono ai simboli base i gruppi di simboli più comuni (generalizzando il concetto di run)
- Orientata alla compressione di dati di tipo letterale o comunque con elevata correlazione tra simboli adiacenti (es. immagini sintetiche)
- Utilizzata nel formato GIF (brevettato)

Le codifiche senza perdite raggiungono di norma compressioni modeste (20-40%) e comunque molto dipendenti dai dati utilizzati. Le compressioni maggiori si ottengono su immagini binarie ed in particolare su testi.

Per predire un nuovo nodo del codice sulla base di altri già posizionati, l'approccio impiegato è quello della definizione di una trasformazione prospettica (omografia) che mappa un insieme di nodi del reticolo nelle rispettive coordinate sul piano immagine. La conoscenza di tale omografia consente di costruire nuovi punti, supponendo che sia valida anche fuori dalla porzione di immagine utilizzata per valutarla. A partire da un reticolo di base, ad esempio 8x8, è quindi possibile calcolare gli 8 parametri dell'omografia (ai minimi quadrati), e tramite questi prevedere le posizioni dei nodi della nona riga (o colonna).

Dimensione 400x400 pixel = 160Kbit = 20KB

Compressione RL: 7876 B

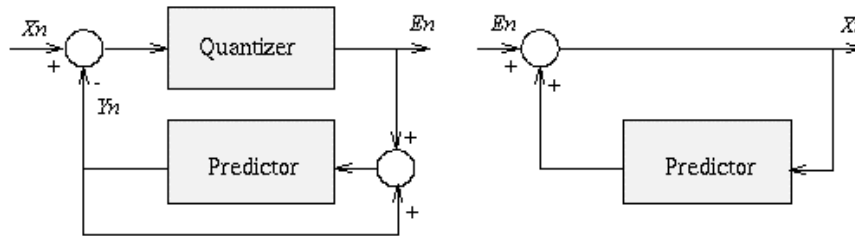
Compressione GIF: 7112 B

CODIFICA PREDITTIVA

E' storicamente la prima tecnica di compressione. Si basa sulla definizione di un operatore (filtro predittivo) che approssima il valore x_n

$$x_n = y_n + e_n \quad \text{con} \quad y_n = \sum a_i x_{n-i}$$

La componente y_n rappresenta la parte predicibile (e che non serve trasmettere), mentre e_n è la parte NON predicibile (e che va trasmessa). Può essere senza o con perdite (incluso il quantizzatore)



Il coefficienti del filtro predittivo si ottengono minimizzando l'errore (es. ai minimi quadrati). Si ottiene un sistema lineare basato sui coefficienti di aurocorrelazione di X , cioè $R_k = E [X_n X_{n+k}]$

La tecnica è generalizzabile ad immagini 2D con predittori bidimensionali

$$X_{ij} = a_1 X_{i-1j} + a_2 X_{ij-1} + a_3 X_{i-1j-1}$$

Di norma pochi coefficienti (es. 3) bastano per eliminare la ridondanza della sequenza trasmessa (purchè stazionaria). Con dati non stazionari, il filtro predittivo non può essere ottimale

CODIFICHE CON PERDITE

Le immagini vengono di norma codificate con perdite perché i dettagli contenuti sono spesso inessenziali

Codifiche con trasformate

I dati vengono trasformati in un dominio in cui la correlazione tra di loro viene ridotta

- Nel dominio trasformato deve essere semplice scegliere quali dati sono importanti e quali no
- La trasformazione deve essere invertibile

A questo punto si opera come con le codifiche senza perdite.

Obiettivo:

- minimizzare il numero di bit utilizzati
- con un vincolo sulla misura di distorsione

$$\text{MSE} = \sum (x_{ij} - y_{ij})^2 \quad \text{PSNR} = 10 \log_{10} = (255^2 / \text{MSE})$$

Altri fattori importanti:

- complessità computazionale
- requisiti di memoria
- robustezza a condizioni variabili
- possibile codifica progressiva

TRASFORMATA DI KARHUNEN-LOEWE

Dato un processo stazionario X a media nulla consideriamo il vettore

$$X_i = (x_{iL}, x_{iL+1}, \dots, x_{iL+L-1})$$

Sia $\Phi = E[X^T X]$ la matrice di covarianza di X . Questa si decompone in

$$\Phi = \Gamma \Lambda \Gamma^T$$

dove $\Lambda = \text{diag}(\lambda_1 \dots \lambda_L)$ e Γ indicano la matrice diagonale degli autovalori e gli autovettori corrispondenti. Sia allora: $Y = \Gamma X$ e quindi $E[Y^T Y] = \Lambda$

Quindi le componenti di Y sono scorrelate. Analizzando gli autovalori λ_i è possibile decidere quanti sono rilevanti per la trasmissione (in generale tutti quelli > 0). L'autovalore λ_i misura il contenuto informativo della i -esima componente. Le componenti di Y sono anche dette componenti principali di X

Problema

La trasformazione è ottima ma dipende dai dati (secondo Φ), per cui non è generalizzabile

Soluzione

Cercare trasformazioni sub-ottime ma indipendenti dai dati

TRASFORMATA COSENO

La trasformata coseno è l'approccio più usato per ottenere dati fra loro scorrelati.

$$X(u, v) = \frac{2}{N} C(u) C(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i, j) \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N}$$

$$i, j, u, v = 0, 1, 2, \dots, N-1$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases}$$

$$x(i, j) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) X(u, v) \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N}$$

Proprietà

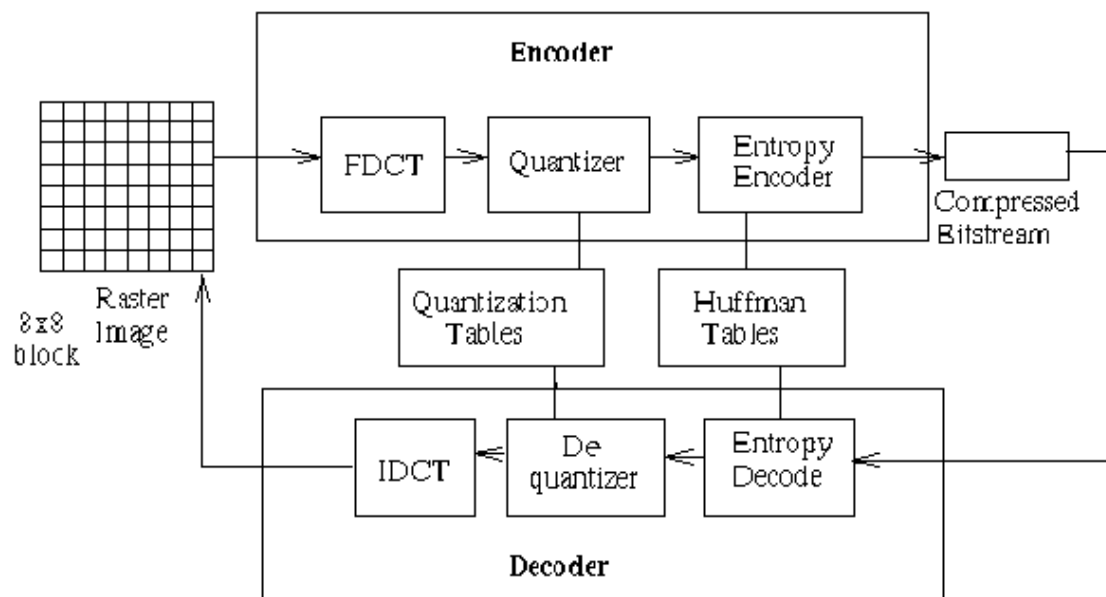
- non dipende dai dati
- non richiede aritmetica complessa
- condizioni al contorno tali da ridurre le discontinuità
- inversa con espressione analoga (a meno della normalizzazione)
- possiede implementazioni efficienti (FFT)
- facilmente estendibile a N dimensioni (nucleo separabile)
- prestazioni vicine a KLT

I coefficienti della DCT sono interpretabili come ampiezze delle componenti alle varie frequenze, per cui

$$Y_{00} = \text{componente continua (DC)} \quad Y_{mn} = \text{componenti "alternate" (AC)}$$

ALGORITMO JPEG (1)

Standard per compressione di immagini sia a livelli di grigio che a colori



ALGORITMO JPEG (2)

Componenti

- DCT a blocchi 8x8

L'immagine viene divisa in blocchi quadrati e per ogni blocco vengono calcolati i corrispondenti coefficienti della DCT (in aritmetica intera)

- Quantizzazione

I coefficienti della DCT sono quantizzati secondo una matrice Q_{mn} selezionata in base alla qualità desiderata del risultato (e molti quindi possono annullarsi)

- Codifica RL

I coefficienti non nulli sono ordinati secondo una scansione che massimizza la probabilità di trovare dei coefficienti nulli consecutivi (run)

- Codifica Huffman

Le lunghezze dei run nulli ed i coefficienti non nulli vengono infine codificati con simboli di lunghezza variabile

ALGORITMO JPEG (3)

La compressione si ottiene tramite una bit allocation in cui ad ogni posizione si assegna un certo numero di bit.

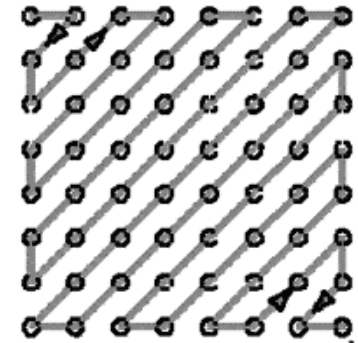
Esempio

8 7 6 4 4 3 3 3
7 5 3 3 2 2 1 1
6 3 2 2 1 0 0 0
4 3 2 1 0 0 0 0
4 2 1 0 0 0 0 0
3 2 0 0 0 0 0 0
3 1 0 0 0 0 0 0
3 1 0 0 0 0 0 0

Blocco immagine originale: 512 bit
Blocco immagine codificato: 108 bit
Compressione 5:1

La tabella di bit allocation viene scelta una volta sola per ogni immagine e può quindi risultare non ottima per alcune zone

La trasformazione in run length viene fatta secondo un percorso a zig-zag a partire dalla DC e toccando le AC di frequenza via via crescente



ALGORITMO JPEG (4)

Prestazioni

Bit rate [bit/pixel]	MSE [livelli]	PSNR [dB]
1,5	2,5	40,0
1,0	3,2	38,0
0,75	3,8	36,5
0,50	4,7	34,5
0,25	7,3	31,0

Vantaggi

- Buone prestazioni su immagini di tipo fotografico
- Semplicità di uso e configurazione
- Velocità
- Simmetria tra codifica e decodifica
- Miglior compromesso bit rate/qualità rispetto ad altre trasformazioni

Svantaggi

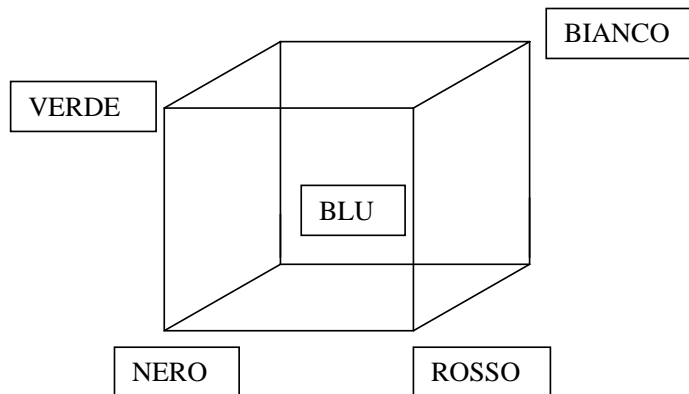
- Effetto di blocchettatura dovuti alla scelta fissa 8x8
- Perdita di dettagli sulle zone con tessitura
- Aumento del rumore in prossimità dei contorni

IMMAGINI A COLORI

Una immagine a colori viene rappresentata come la sovrapposizione di 3 immagini a livelli di grigio

RGB

- i 3 canali sono gli assi di uno spazio 3D (color space)
- rappresentazione comoda per elaborazioni numeriche
- metrica non uniforme (non percettiva)



Esistono rappresentazioni del colore piu` legate alla percezione umana, e quindi piu` adatte per le applicazioni legate alla visualizzazione (es. multimedia)

Sono tutte espresse in termini di trasformazioni lineari o non lineari della terna (R,G,B)

RAPPRESENTAZIONI DEL COLORE

Sono riconducibili a concetti percettivi come luminosità, tinta, brillantezza, etc.

YIQ (o YUV o YCbCr) Nasce all'interno dello standard televisivo NTSC

$$Y = 0.299 R + 0.587 G + 0.114 B$$

$$I = 0.596 R - 0.274 G - 0.322 B$$

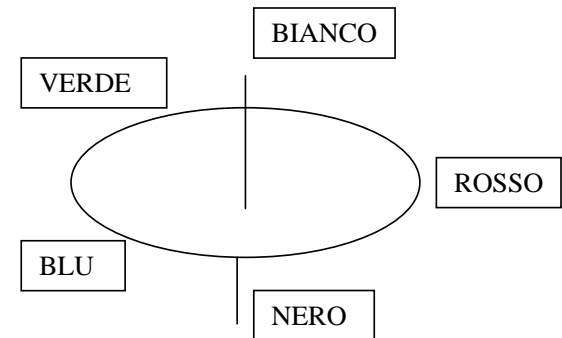
$$Q = 0.211 R - 0.523 G + 0.312 B$$

- Trasformazione lineare (invertibile) che tende a decorrelare (in media) i tre canali RGB
- Y in relazione con la luminosità (secondo l'occhio)
- Differenti scelte possibili per la matrice dei coefficienti
- I canali cromatici possono essere sottocampionati

HSI (o HLS)

Hue, Saturation, Intensity (o Hue, Lightness, Saturation)

- Legame non lineare con RGB
- La hue codifica il di colore dominante (in forma polare)
- La saturazione codifica il concetto di brillantezza



Lo standard JPEG non definisce a priori alcuna codifica del colore, assumendo che ciascuna componente venga compressa in modo indipendente.

CODIFICHE DI SOTTOBANDA (1)

Approccio:

Decomporre il segnale in bande frequenziali non sovrapposte e codificarle in modo separato. Di norma si usa la decomposizione binaria

Metodologia

La tecnica standard utilizza due filtri: un passa-basso $H_1(\omega)$ ed un passa-alto $H_2(\omega)$ tali che:

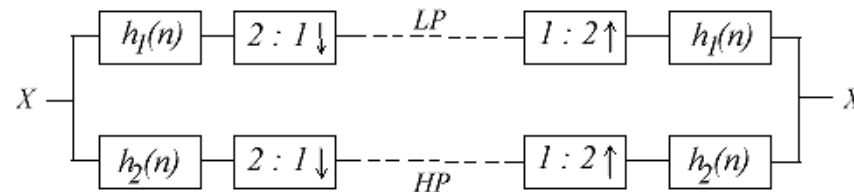
$$|H_1(\omega)|^2 + |H_2(\omega)|^2 = 1$$

I filtri vanno realizzati tramite convoluzioni con nuclei $h_1(n)$ (simmetrico) ed $h_2(n)$ (antisimmetrico) a supporto compatto.

Osservazione

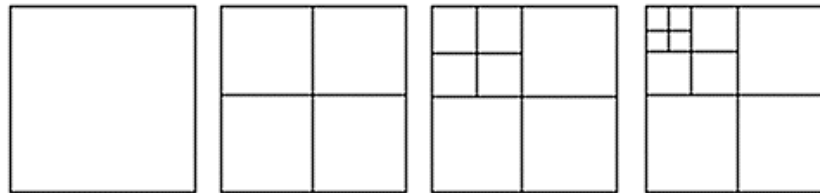
La decomposizione binaria ideale non è ottenibile in modo esatto, ma solo in modo approssimato.

Le due bande ottenute possono essere decimate (cioè sottocampionate) di un fattore 2, in modo che partendo da N dati si ottengono due sottobande da $N/2$ dati ciascuna.



CODIFICHE DI SOTTOBANDA (2)

Lo schema di codifica può essere facilmente generalizzato a dati a più dimensioni (es. immagini) per separabilità. Può essere replicato per la sottobanda a bassa frequenza, ottenendo quindi una piramide di sottobande.



La formalizzazione matematica di questa metodologia porta alla Discrete Wavelet Transform (DWT)

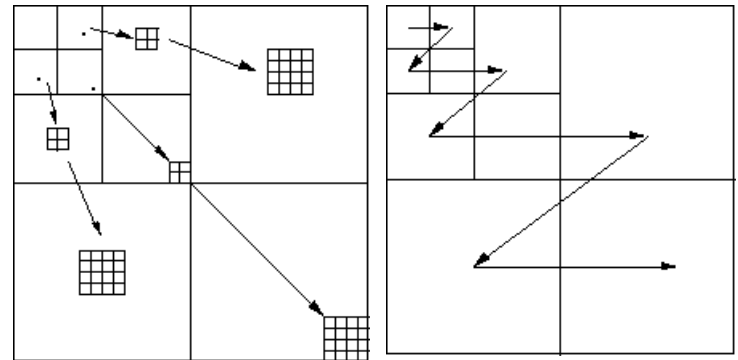
La DWT è utilizzata per un nuovo schema di codifica (JPEG-2000) concettualmente simile al JPEG ma con un vantaggio rilevante: l'assenza di effetti di blocchettatura dovuti alla definizione rigida di una scala di analisi

Problema:

Come codificare in modo efficiente i coefficienti della DWT?

Soluzione:

Codifica ad albero (zero-tree encoding)



LIFTING

Approccio per generare una rappresentazione multirisoluzione computazionalmente efficiente. Sia λ_0 un dato digitale. Si vuole una trasformazione in due componenti decimate λ_1 (bassa risoluzione) e γ_1 (alta risoluzione)

- **Suddivisione** (*lazy transform*) Partizione arbitraria di λ_0 in λ_1 e γ_1
- **Predizione** (*dual lifting*) Stima dei dati in γ_1 noti quelli in λ_1 e calcolo errore $\gamma_1 \leftarrow \gamma_1 - P(\lambda_1)$
- **Aggiornamento** (*primal lifting*) Ricalcolo di λ_1 per conservare alcune proprietà (es. media) $\lambda_1 \leftarrow \lambda_1 + U(\gamma_1)$

Note:

- I passi possono essere ripetuti a partire da λ_1
- Il processo di inversione è banale
- Possibile aritmetica intera (purchè con regola deterministica)

Esempio: Dato il processo X generiamo

$$S_n \leftarrow X_{2n} \quad \text{e} \quad D_n \leftarrow X_{2n+1}$$

Da cui	;	Oppure
$D_n \leftarrow D_n - S_n$;	$D_n \leftarrow D_n - (S_n + S_{n+1}) / 2$
$S_n \leftarrow S_n + D_n / 2$;	$S_n \leftarrow S_n + (D_{n-1} - D_n) / 4$

COMPRESSIONE VIDEO

Per comprimere una immagine si usa la ridondanza spaziale (i pixel vicini tendono ad avere valori simili).
Per una sequenza video risulta naturale utilizzare anche la ridondanza temporale (lo stesso pixel nel tempo tende a mantenere lo stesso colore), che nel video è di norma molto elevata.

- Standard ITU: H.261e H.263
trasmissione su canale telefonico a banda stretta, videoconferenza (64-1920 Kb/s)
- Standard ISO: MPEG-1 e MPEG-2
applicazioni multimediali, CD-ROM, TV digitale, DVD (2-200 Mbit/s)

Approccio base:

- codifica predittiva (frame differencing) con feedback dell'errore – si analizza la differenza tra l'immagine corrente e la precedente (ricostruita in modo da evitare accumulo dell'errore)
- compensazione del movimento (motion compensation) – non si fa la differenza pixel per pixel ma stimando un vettore di moto tale da minimizzare le differenze

Codifiche proprietarie

- Quicktime (Apple)
- Video For Windows (Microsoft)
- RealVideo (Real Networks)