Capitolo 10

Codifica di sorgente multimediale

Affrontiamo gli aspetti applicativi della teoria esposta al capitolo 9 e relativi alla codifica di sorgente continua per *segnali audio-visivi*, che dunque si particolarizza in virtù delle caratteristiche peculiari di tali segnali, associati a quelle degli organi sensoriali umani che ne sono i destinatari naturali. Per quanto riguarda *l'audio* sono illustrate le tecniche orientate a riprodurre il segnale nel tempo come ad es. nel PCM e quindi quelle più orientate al segnale vocale, basate su di un modello del sistema fonatorio di produzione; infine, sono accennate le tecniche basate su di un modello del sistema uditivo, come per l'MP3. Si passa quindi alla codifica di *immagini* fisse, alle quali si possono applicare le tecniche nate per la codifica di sorgente discreta come ad es. per le GIF, oppure altre più ispirate alla fisiologia del sistema visivo come per il JPEG. La trattazione della codifica *video* si articola attraverso l'evoluzione storica dei vari standard che se ne sono occupati, arricchendo le tecniche di codifica di immagine con la rappresentazione del movimento e di come questo possa essere predetto, fino a generare il multiplex numerico con i programmi televisivi che pervadono la nostra esistenza.

10.1 Codifica audio

Al § 4.3.1.1 abbiamo svolto una valutazione approssimata della distorsione introdotta dal processo di quantizzazione di segnale audio, ricavando che l'utilizzo di M bit/campione si traduce in $SNR_q\,(M)\big|_{dB} \simeq 6 \cdot M$ dB. Quindi, al § 4.3.2 si è mostrato come adottando una caratteristica di quantizzazione logaritmica anziché lineare ci si possa adattare meglio alla effettiva densità di probabilità del segnale vocale, rendendo inoltre SNR_q relativamente poco sensibile alla sua effettiva dinamica, dando luogo alla cosiddetta codifica PCM con *legge A* o *legge* μ, standardizzata nel 1988 da ITU-T come G.711,[1]. Mentre questa costituisce un formato universale di scambio permettendo la compatibilità tra dispositivi e tecnologie, nel seguito sono state sviluppate diverse tecniche alternative[2], in grado di offrire la stessa (o migliore) qualità di ascolto con velocità di trasmissione

[1] `http://www.itu.int/rec/T-REC-G.711/e`

[2] Una raccolta di riferimenti a risorse relative a codec audio orientati alle applicazioni multimediali può essere trovata presso `https://teoriadeisegnali.it/story/labtel/`

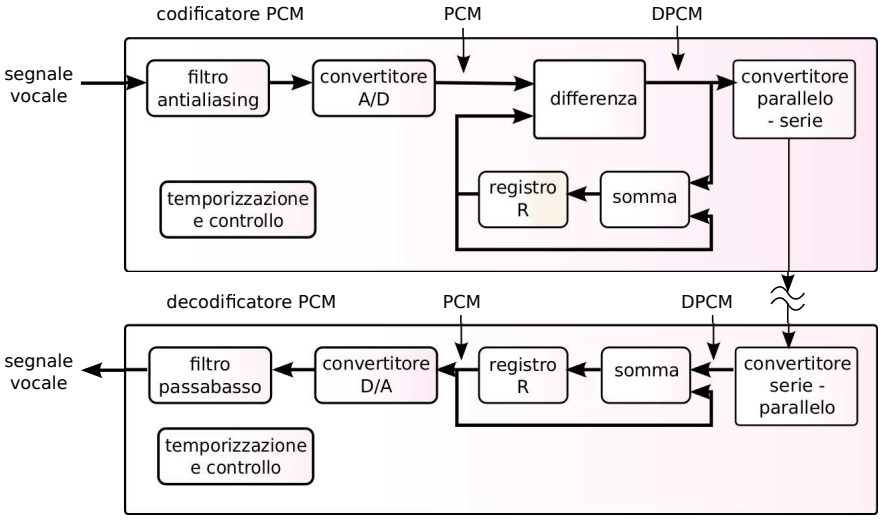


Figura 10.1: Codec audio Differential PCM o DPCM

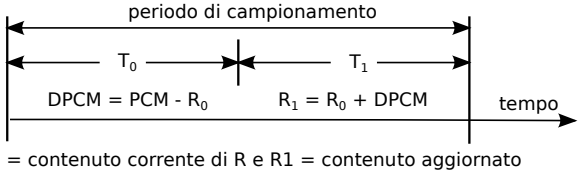
contenute, non solo per segnali vocali in banda telefonica, ma anche per segnali a banda larga, musicali, e multicanale, di cui tentiamo ora una sommaria rassegna.

10.1.1 Codifica di forma d'onda

Questa classe di codificatori opera esclusivamente nel dominio del tempo, agendo campione per campione, e ottiene una qualità comparabile o superiore a quella del PCM sfruttando le caratteristiche *di memoria* presenti nel segnale, e/o adattando alcuni parametri di funzionamento alle caratteristiche tempo varianti del segnale.

10.1.1.1 DPCM o PCM Differenziale

La prima variazione rispetto al PCM è stata quella di applicare il principio della codifica predittiva (pag. 263), semplicemente adottando il precedente campione di ingresso come *predizione* di quello successivo. Il corrispondente schema di elaborazione è mostrato in fig. 10.1, ed il suo funzionamento è suddiviso in due fasi come rappresentato a lato: nella prima (T_0) il codificatore sottrae il campione precedente (all'inizio nullo) all'attuale, e nella seconda (T_1) questa differenza è risommata al valore di differenza precedente (all'inizio nullo) in modo da ri-calcolare il valore attuale, e salvarlo nel registro di memoria R. Il segnale differenza è caratterizzato da valori di ampiezza ridotti rispetto all'originale, e può essere codificato con 7 bit/campione, producendo ora una velocità di 56 kbps per ottenere un segnale di qualità telefonica. Il decodificatore si limita quindi a sommare alla differenza ricevuta il valore ricostruito del campione precedente, ed effettuare l'operazione di restituzione analogica. Osserviamo che il codificatore calcola



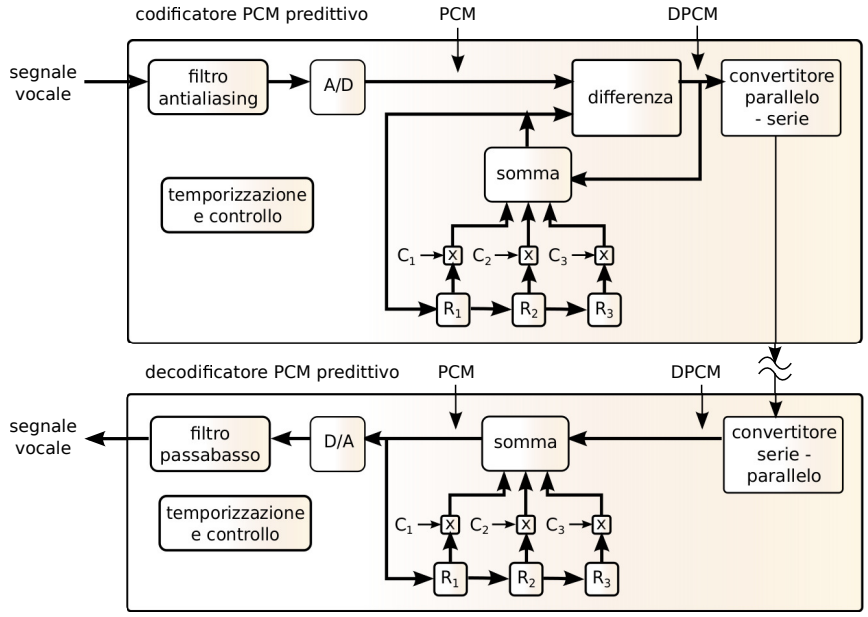


Figura 10.2: Codec DPCM con predittore a tre coefficienti costanti

il valore precedente mediante un circuito identico a quello presente al decodificatore, e per questo l'operazione è perfettamente invertibile.

10.1.1.2 ADPCM o DPCM Adattivo

Questo metodo differisce dal precedente per due aspetti: da un lato il processo di predizione tiene conto di più di un campione passato e non di uno solo come nel DPCM, come descritto in fig. 10.2 in cui è mostrato un predittore del terzo ordine che in pratica consiste in un filtro trasversale i cui coefficienti sono fissati in base alle caratteristiche statistiche medie del segnale vocale. Il secondo aspetto è che ora il quantizzatore *modifica nel tempo* la propria dinamica di azione (da cui il termine *adattativo*, o *adattivo*) in base ad una stima della dinamica del segnale.

Nel lato sinistro della fig. 10.3 è mostrata una caratteristica di quantizzazione uniforme operante su di una dinamica di ingresso $\phi_x \hat{\sigma}_x$, con $\phi_x > 1$ scelto in modo da rendere trascurabile la probabilità che un valore di ingresso troppo elevato determini la saturazione del quantizzatore. Utilizzando una stima a breve termine della varianza $\hat{\sigma}_x^2$ calcolata sugli ultimi campioni di segnale (a media nulla), ossia ad es. calcolando $\hat{\sigma}_x^2(n) = \frac{1}{N}\sum_{i=1}^{N} x^2(n-i)$, si possono rendere gli intervalli di decisione Δ piccoli nelle fasi di segnale piccolo, in modo da mantenere l'SNR costante anche per segnali con ampiezze molto variabili. Inoltre, è possibile *omettere* la trasmissione della stima di varianza se quest'ultima è calcolata in modalità *backward*, ossia a partire dai valori $y(n) = Q[x(n)]$, dato che la stessa operazione è eseguibile in modo indipendente anche dal lato del decodificatore. La stima della varianza è ulteriormente semplificata

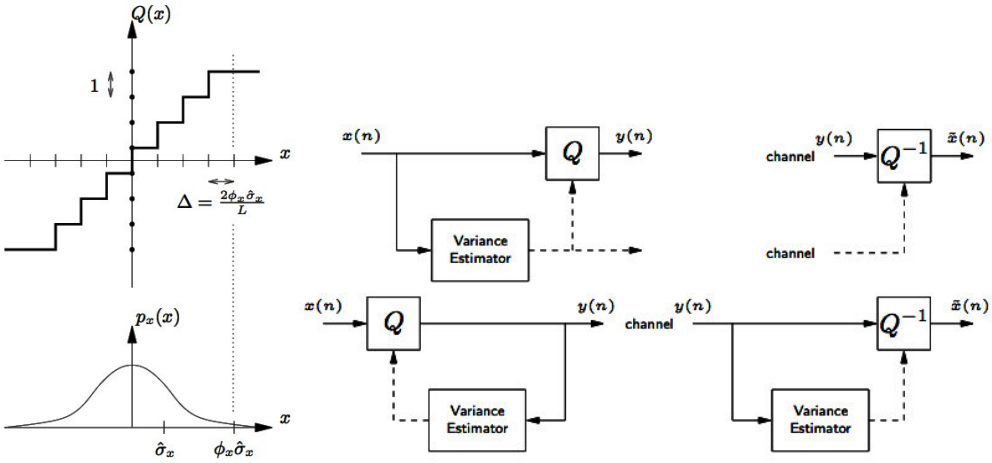


Figura 10.3: Quantizzatore a dinamica variabile (a sinistra) e stima della dinamica $\hat{\sigma}_x^2$ per via *diretta* (in altro a ds.) o *backward* (in basso) - tratto da `http://cnx.org/content/m32074/latest/`

se realizzata mediante una formula ricursiva, ossia

$$\hat{\sigma}_x^2(n) = \alpha\hat{\sigma}_x^2(n-1) + (1-\alpha)\, y^2(n)$$

il cui risultato è mostrato in fig. 10.4, dove la linea tratteggiata rappresenta il valore istantaneo di $y^2(n)$, mentre quella continua mostra i valori di $\hat{\sigma}_x^2(n)$ ottenuti in modo ricursivo.

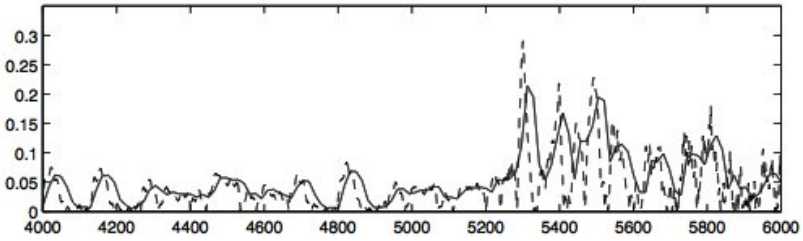


Figura 10.4: Stima ricursiva backward della varianza $\hat{\sigma}_x^2(n)$ confrontata con $y^2(n)$, per $\alpha = 0.9$

La fig. 10.5 mostra infine i due estremi del codec ADPCM, che rimangono sincronizzati anche nel caso di saturazione del quantizzatore adattativo.

Il miglioramento della qualità ottenibile ha determinato la possibilità di ridurre il numero di bit (e di conseguenza di livelli) del quantizzatore a 5, 4, 3, 2 bit/campione, a cui corrispondono rispettivamente velocità di codifica di 40, 32, 24, 16 kbps. Questi sono i valori a cui si riferisce lo standard ITU-T G.721, successivamente confluito nel G.726.

10.1.1.3 Codifica per sottobande

Anche la raccomandazione G.722 è basata sulla codifica ADPCM, ma applicata ad un segnale audio con banda più larga, riproducendo correttamente frequenze fino a 7 KHz. Ciò avviene dopo aver suddiviso le componenti frequenziali del segnale in due

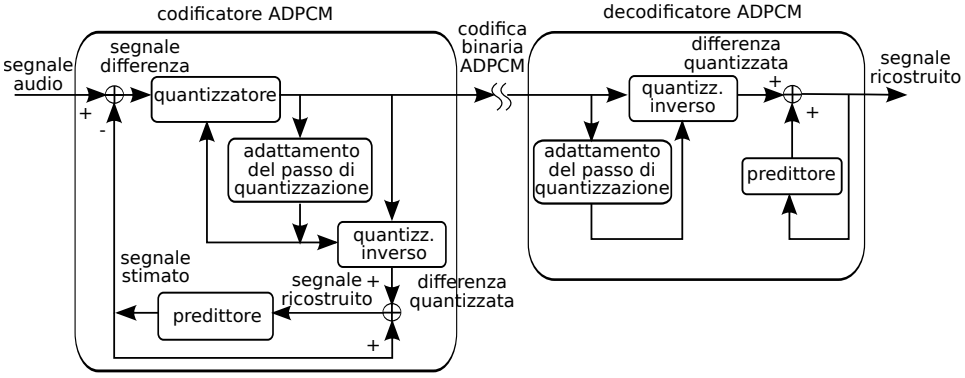


Figura 10.5: Architettura di un codec ADPCM

sottobande come mostrato in fig. 10.6, mediante una coppia di filtri passa-basso e passa-alto con frequenza di taglio comune a 3.5 KHz.

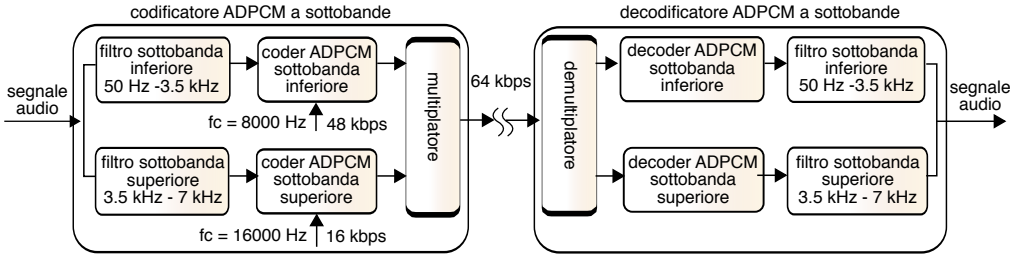


Figura 10.6: Architettura di un codec ADPCM a sottobande

Il canale relativo alla semi banda superiore è campionato a frequenza di 16 kHz, mentre l'altro è praticamente equivalente al segnale in banda telefonica preso in esame fino ad ora. Per entrambi i canali è applicata la codifica ADPCM, ma le rispettive velocità sono impostate in modo differente, dando più importanza alla componente di bassa frequenza, percettivamente più rilevante: ad esempio, si può scegliere di assegnare 16 kbps alle alte frequenze e 48 alle basse, ottenendo un totale di 64 kbps per una qualità risultante migliore del G.711, in quanto ora si opera su di un segnale a larga banda, con risultati idonei ad applicazioni come la videoconferenza.

Lo stesso schema di codifica per sottobande più ADPCM è proposto anche dallo standard G.726, ma applicato ad un segnale a qualità telefonica, offrendo le velocità di 40, 32, 24 e 16 kbps.

10.1.2 Codifica basata su modello

I metodi fin qui discussi non tengono particolarmente conto della natura del segnale da codificare. Restringendo viceversa il campo al solo caso di segnale vocale, le conoscenze relative alla sua particolare modalità di produzione possono essere usate per ridurre le informazioni da trasmettere, costituite ora dai parametri che caratterizzano un suo modello di generazione. Essendo questo il dominio storico delle scienze linguistiche e fonetiche, svolgiamo una piccola digressione in tal senso.

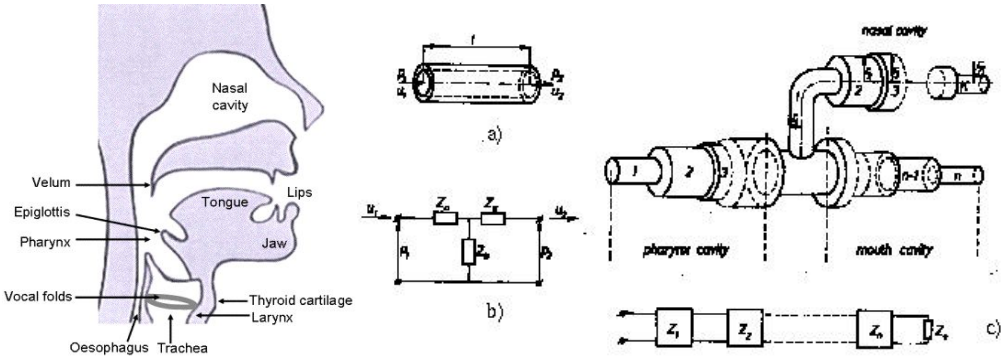


Figura 10.7: Rappresentazione schematica del tratto vocale e relativo modello a tubi

10.1.2.1 Produzione del segnale vocale

L'apparato fonatorio viene idealizzato per mezzo del cosiddetto *modello a tubi* (vedi fig. 10.7), in cui sia il *tratto vocale* (compreso tra le corde vocali e le labbra) che il *tratto nasale* (dal velo alle narici) sono pensati come una concatenazione di tubi di diversa sezione. Nei suoni vocalici la muscolatura della *laringe* determina la chiusura periodica delle *corde vocali*, interrompendo il flusso d'aria che le attraversa, e dando origine ad un *segnale di eccitazione* anch'esso periodico detto *onda glottale*, la cui frequenza è detta *pitch*[3]; la differenza di area delle diverse sezioni del tratto vocale provoca un *disadattamento di impedenza acustica*[4] e la conseguente formazione di onde riflesse (vedi fig. 10.8) , che per lunghezze d'onda in relazione intera con la lunghezza del tratto vocale, determinano fenomeni di *onde stazionarie*, ovvero di *risonanze*[5], le

[3] `http://en.wikipedia.org/wiki/Pitch_accent_(intonation)`

[4] Si applica in pratica la stessa teoria valida per le linee elettriche, in cui al posto di tensione e corrente, ora si considerano rispettivamente pressione p e velocità u

[5] Si tratta di un fenomeno in qualche modo simile a quello che si verifica soffiando in una bottiglia, e producendo un suono che dipende dalla dimensione della stessa.

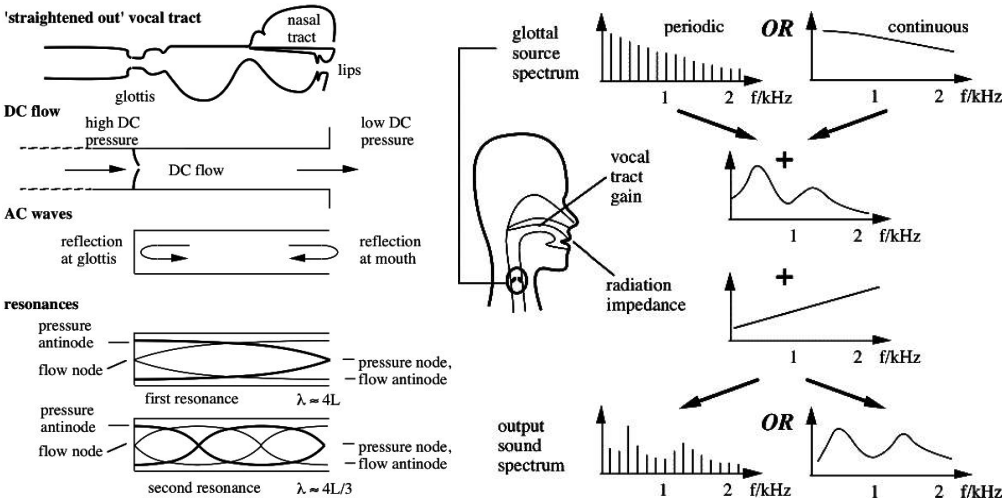


Figura 10.8: Natura delle risonanze del tratto vocale e loro effetto filtrante sull'onda glottale

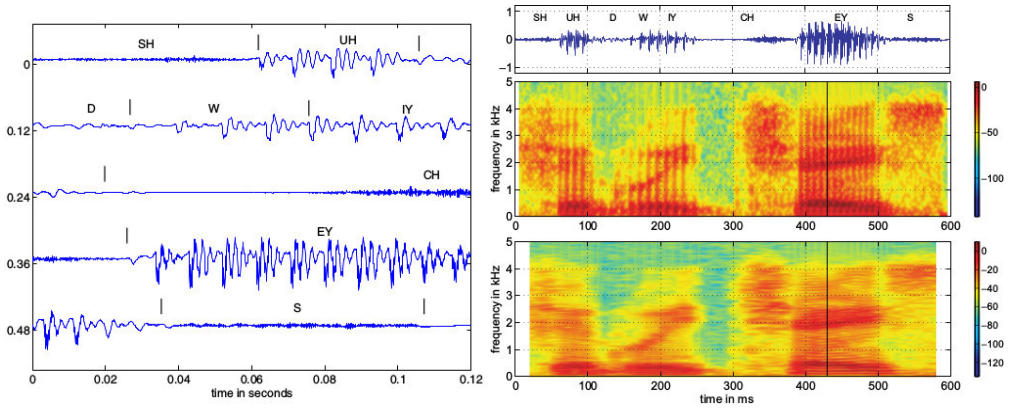


Figura 10.9: Forma d'onda e spettrogramma per la frase *should we chase?*

cui frequenze sono indicate in fonetica come *formanti*. Il verificarsi di tali risonanze genera un *effetto filtrante* che modifica lo spettro dell'onda glottale, producendo così *il timbro* corrispondente ai diversi suoni della lingua[6]. Il tratto vocale termina quindi con l'apertura delle labbra, che nel modello a tubi corrisponde ad una *impedenza di radiazione* che produce un effetto di derivata, e dunque *un'enfasi* delle alte frequenze per lo spettro complessivo del parlato. Infine, il modello si assume valido anche per i suoni *fricativi*, prodotti anziché mediante le corde vocali, mediante una occlusione che causa *turbolenza* nel flusso d'aria.

Caratteristiche tempo-frequenza del segnale vocale La parte sinistra di fig. 10.9 mostra la forma d'onda relativa alla frase inglese "should we chase?" (*dovremmo inseguire?*) assieme alla relativa trascrizione fonetica[7], mettendone in luce il carattere quasi periodico in corrispondenza delle vocali e quello tipo rumore per le consonanti, nonché la diversa durata dei vari suoni, l'assenza di confini temporali precisi tra gli stessi, e la diminuzione del periodo di pitch a fine frase, corrispondente all'intonazione crescente tipica di una frase interrogativa. In particolare, notiamo come per i suoni vocalici i singoli periodi di pitch siano caratterizzati da una brusca discontinuità prodotta dall'onda glottale, seguita da oscillazioni smorzate legate alle risonanze del tratto vocale.

Il segnale viene quindi campionato a 10 KHz e suddiviso in *finestre di analisi*, per le quali vengono calcolate delle DFT, la cui densità di energia in dB è riprodotta *in verticale* mediante una scala cromatica come mostrato negli *spettrogrammi*[8] presenti al lato destro di fig. 10.9, che permettono di valutarne la variabilità temporale delle

[6] I diversi suoni vocalici e/o consonantici (detti *fonemi*) sono prodotti mediante diverse posture articolatorie (la posizione di lingua, mascella e labbra), ovvero diversi profili d'area del tratto vocale, nonché l'attivazione o meno del tratto nasale. Presso `https://www.youtube.com/watch?v=6dAEE7FYQfc` è mostrato il video di una *risonanza magnetica* effettuata durante l'eloquio. In definitiva ai diversi fonemi corrispondono differenti frequenze formanti, e dunque una diversa risposta in frequenza.

[7] I simboli usati sono noti come *arphabet*, vedi `http://en.wikipedia.org/wiki/Arpabet`, e la pronuncia dovrebbe essere qualcosa del tipo *sciuduiceis*.

[8] Tratti da `https://books.google.it/books?id=Z6Otr8Hj1WsC`

caratteristiche spettrali. Il differente aspetto dei due diagrammi è dovuto alla diversa lunghezza di finestra, pari rispettivamente a 10 e 40 msec per il grafico superiore ed inferiore[9]. In entrambe le rappresentazioni sono ben evidenti le *traiettorie delle formanti*, che evolvono in modo *continuo*, coerentemente con la velocità di articolazione del parlante.

10.1.2.2 Codifica a predizione lineare - LPC

Il modello di produzione e le caratteristiche illustrate portano a formulare un processo di codifica basato sulla suddivisione del segnale vocale in intervalli (o finestre di analisi) di estensione tra i 10 ed i 30 msec, durante i quali il segnale può essere considerato praticamente stazionario[10], e su tali finestre condurre una *analisi* (o stima) dei parametri del modello, che sono

- il tipo di eccitazione (periodica o caotica), la sua frequenza fondamentale (o *pitch*) se periodica, e la sua intensità;
- i parametri che caratterizzano l'effetto filtrante del tratto vocale.

e quindi trasmettere questi valori, in modo che in ricezione sia possibile riprodurre un segnale simile all'originale mediante un decodificatore del tipo illustrato in fig. 10.10.

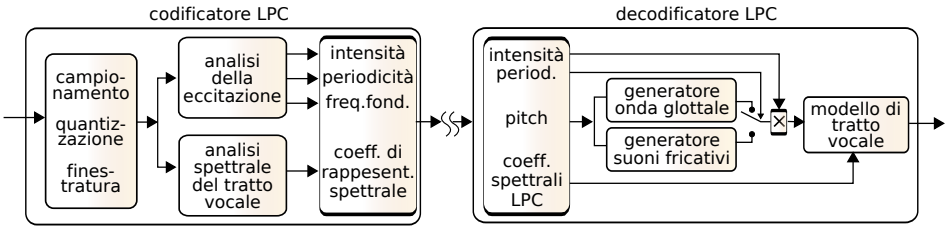


Figura 10.10: Schema di codificatore e decodificatore LPC

Il modello *a tubi* (e quindi basato sulle risonanze) del tratto vocale illustrato in fig. 10.8 si presta a considerare un filtro *di sintesi* di tipo numerico e *ricursivo* o IIR (§ 5.3.2.2) di ordine p, che calcola il valore dei campioni di uscita y_n a partire da una combinazione lineare di p campioni passati $\hat{y}_n = \sum_{i=1}^{p} a_i y_{n-i}$, a cui sommare un *errore di predizione* e_n che rappresenta il processo di eccitazione, ovvero

$$y_n = \hat{y}_n + e_n = \sum_{i=1}^{p} a_i y_{n-i} + e_n \tag{10.1}$$

Per ogni finestra di analisi i coefficienti a_i (o coefficienti LPC del predittore di ordine p) si ottengono come quelli che rendono *minimo* il valore atteso dell'errore quadratico

$$E\left\{e_n^2\right\} = E\left\{\left(y_n - \sum_{i=1}^{p} a_i y_{n-i}\right)^2\right\}$$

[9] Una finestra di 10 msec ha durata comparabile con il periodo di pitch, e ciò produce l'effetto a striature *verticali* del primo diagramma, meno pronunciato verso la fine, dove il pitch è più elevato. Una finestra di 40 msec si estende su più periodi di pitch, e determina una migliore risoluzione in frequenza, cosicché nel diagramma inferiore si possono notare delle striature *orizzontali* che corrispondono alle *armoniche* della frequenza di pitch.

[10] Una sillaba può estendere la sua durata tra 10-15 msec per le vocali *ridotte*, fino a più di 100 msec per quelle *accentate*.

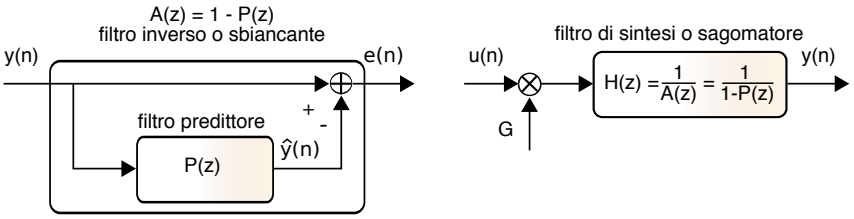


Figura 10.11: Filtro predittore, filtro inverso associato, e filtro di sintesi LPC

(ovvero, l'energia dell'errore), e sono individuati eguagliando a zero l'espressione delle derivate parziali di $E\left\{e_n^2\right\}$ rispetto ai coefficienti a_j. Scriviamo dunque

$$\frac{\partial}{\partial a_j} E\left\{\left(y_n - \sum_{i=1}^{p} a_i y_{n-i}\right)^2\right\} = 2E\left\{\left(y_n - \sum_{i=1}^{p} a_i y_{n-i}\right) y_{n-j}\right\} = 0$$

da cui si ottiene

$$E\left\{y_n y_{n-j}\right\} = \sum_{i=1}^{p} a_i E\left\{y_{n-i} y_{n-j}\right\} \tag{10.2}$$

Il valore atteso $E\left\{y_{n-i} y_{n-j}\right\}$ viene *stimato*[11] come quello della autocorrelazione discreta calcolata sui campioni di segnale delimitati dalla finestra di analisi corrente, ovvero

$$\begin{aligned} E\left\{y_{n-i} y_{n-j}\right\} &\doteq \mathcal{R}_{yy}(|i-j|) \qquad \text{e ponendo } k = |i-j| \\ &= \mathcal{R}_{yy}(k) = \sum_{n=1}^{N-k} y_n y_{n+k} \end{aligned} \tag{10.3}$$

dove l'estremo superiore della sommatoria varia in modo da includere solo i campioni effettivamente presenti nella finestra[12]. La (10.3) permette di riscrivere (10.2) come

$$\mathcal{R}_{yy}(j) = \sum_{i=1}^{p} a_i \mathcal{R}_{yy}(|i-j|)$$

che valutata per $j = 1, \cdots, p$ individua un sistema di p equazioni[13] in p incognite

$$\begin{bmatrix} \mathcal{R}(1) \\ \mathcal{R}(2) \\ \vdots \\ \mathcal{R}(p) \end{bmatrix} = \begin{bmatrix} \mathcal{R}(0) & \mathcal{R}(1) & \cdots & \mathcal{R}(p-2) & \mathcal{R}(p-1) \\ \mathcal{R}(1) & \mathcal{R}(0) & \cdots & \mathcal{R}(p-3) & \mathcal{R}(p-2) \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ \mathcal{R}(p-1) & \mathcal{R}(p-2) & \cdots & \cdots & \mathcal{R}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \tag{10.4}$$

che può essere risolto nei termini dei coefficienti a_i mediante metodi particolarmente efficienti[14]; ed i coefficienti utilizzati dal decodificatore per applicare la (10.1).

[11] Sottintendendo una ipotesi di stazionarietà ed ergodicità non vera, ma molto comoda per arrivare ad un risultato.

[12] La (10.3) è effettivamente una stima della *autocorrelazione* del segnale a durata limitata che ricade nella finestra di analisi, mentre l'inclusione nella sommatoria di un numero di termini pari al numero di campioni disponibili porta ad un diverso tipo di risultato, detto *metodo della covarianza*, ed un diverso modo di risolvere il sistema (10.4).

[13] dette di *Yule-Walker*, vedi ad es. `https://it.wikipedia.org/wiki/Equazioni_di_Yule-Walker`

[14] In base alle assunzioni adottate, $\mathcal{R}_{yy}(j)$ risulta una funzione pari dell'indice j, e la corrispondente matrice dei coefficienti viene detta di *Toeplix*, consentendone l'inversione mediante il metodo di *Levinson-Durbin* (vedi `https://en.wikipedia.org/wiki/Levinson_recursion`), che presenta una complessità $O\left(n^2\right)$ anziché $O\left(n^3\right)$, come sarebbe necessario per invertire la matrice dei coefficienti.

Il filtro autoregressivo che esegue il calcolo $\hat{y}_n = \sum_{i=1}^{p} a_i y_{n-i}$ è indicato come *predittore*, ed è associato ad un polinomio[15] $P(z) = \sum_{i=1}^{p} a_i z^{-i}$; viceversa il filtro FIR che valuta l'errore di predizione (o *residuo*) $e_n = y_n - \sum_{i=1}^{p} a_i y_{n-i}$ è indicato come *filtro inverso* o *sbiancante*, viene associato al polinomio $A(z) = 1 - P(z)$, ed è mostrato nel lato sinistro della fig. 10.11. Indicando ora con $G \cdot u_n$ una *codifica* del residuo e_n, il segnale di partenza può essere (quasi) ri-ottenuto come mostrato nella parte destra della fig. 10.11, ossia facendo passare e_n attraverso il filtro IIR $H(z) = \frac{1}{A(z)} = \frac{1}{1-P(z)}$.

Dato che, in base a considerazioni che non svolgiamo, e_n è caratterizzato da una densità spettrale *bianca*, $|H(z)|^2$ (calcolato per $z = e^{i\omega}$) rappresenta una vera e propria *stima spettrale* del segnale di partenza, come mostrato in fig. 10.12 per diversi valori di p, verificando che per suoni vocalici si ottengono risultati accettabili già per valori di p tra 8 e 14, mentre per le fricative l'ordine può essere ancora inferiore.

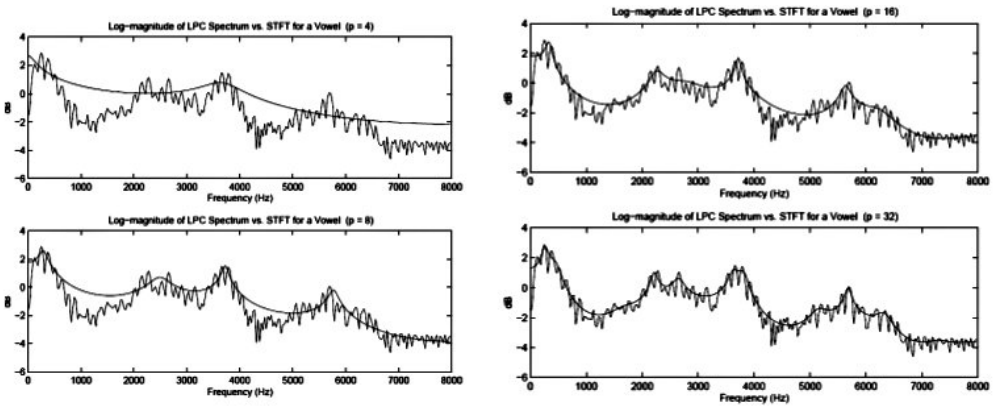


Figura 10.12: Approssimazione spettrale LPC per ordine di predizione p pari a 4, 8, 16 e 32

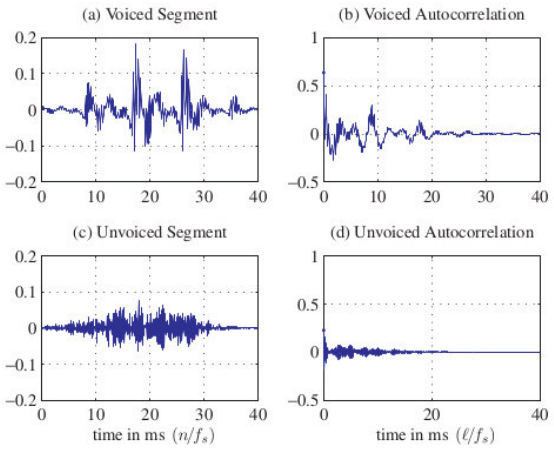
Stima del periodo di pitch Resta ora da illustrare il modo di decidere se la finestra di analisi contenga un suono sordo o sonoro, e nel secondo caso, il suo periodo. Osserviamo che *in media* la frequenza di pitch risulta pari a circa 120 e 210 Hz nel caso rispettivamente di voci maschili e femminili, con una estensione che varia approssimativamente da metà al doppio del pitch medio[16]. La stima del periodo di pitch può essere realizzata a partire dalla funzione di autocorrelazione a breve termine (10.3), mostrata nella colonna di destra della figura che segue, a fianco delle finestre di segnale su cui è stata calcolata[17], per un suono vocalico (sopra) e fricativo (sotto). Come evidente, nel caso del suono vocalico l'autocorrelazione presenta un primo picco a 9 msec ed un

[15] Una breve analisi della relazione tra DFT e trasformata *zeta* è svolta al § 4.5.1, ma vedi anche § 5.3.2.2.

[16] Il pitch varia durante la pronuncia di una frase in accordo alla sua semantica, alla lingua, ed all'enfasi emotiva impressa dal parlatore. Da un punto di vista musicale, la dinamica dei valori (da metà al doppio) si estende quindi su di un intervallo di due ottave. L'intera gamma dei registri dell'opera si differenzia per 22 semitoni, dal Mi2 del basso al Do4 del soprano, ovvero un rapporto di frequenze pari a 3,6.

[17] In realtà prima del calcolo della autocorrelazione il segmento di segnale è stato moltiplicato per una finestra di Hamming, che provoca lo smussamento visibile ai bordi.

secondo a 18 msec, corrispondenti al periodo di pitch ed al suo doppio; viceversa nel caso del suono simile al rumore, non sono visibili picchi, come da aspettarsi nel caso di una segnale incorrelato. Pertanto, l'autocorrelazione può essere usata per indicare la presenza o meno di un suono vocalico, e nel caso affermativo, stimare il suo pitch.



Nella pratica per i suoni sordi si ottengono buoni risultati di sintesi usando come eccitazione un vero e proprio rumore bianco; d'altra parte, per i suoni sonori l'uso di forme d'onda impulsive con periodo pari al pitch stimato, sebbene capaci di produrre un bit rate riducibile fino a 2.4 kbps, non fornisce risultati particolarmente utilizzabili, producendo un voce piuttosto robotica. Per questo motivo, si sono sviluppate le tecniche seguenti.

10.1.2.3 Predizione lineare ad eccitazione residuale - RELP

Per ovviare alla sovra-semplificazione dello schema di sintesi riportato in fig. 10.10, dopo aver svolto l'analisi spettrale LPC il residuo di predizione relativo alla finestra di analisi viene effettivamente calcolato, applicando poi allo stesso una tecnica di codifica di forma d'onda[18]: questo modo di operare è indicato come codifica RELP (*Residual Excited LP*).

Analysis by synthesis - ABS Anziché *calcolare* il residuo di predizione, codificarlo, e trasmetterlo in tale forma, la tecnica di *analisi via sintesi* adotta una tecnica *ad anello chiuso*, cercando di trovare quale segnale di eccitazione[19] fornire al filtro di sintesi in modo che il risultato sia quanto più possibile simile al segnale originale (vedi fig. 10.13); i parametri del filtro di sintesi e della eccitazione sono quindi trasmessi al decoder. La funzione di minimizzazione opera dunque una vera e propria *ricerca tra i possibili segnali* di eccitazione.

[18] In questo modo si evita anche di dover operare una esplicita decisione *sonoro/sordo*, visto che in realtà le due fonti di eccitazione posso essere presenti contemporaneamente, come per i cosiddetti suoni *affricati*.

[19] Generato per tentativi, oppure da scegliere in un *dizionario* di sequenze di eccitazione già codificate.

Figura 10.13: Schema di codifica vocale ABS - *Analysis by Synthesis*

Filtraggio percettivo Sempre in fig. 10.13 si mostra come il processo di minimizzazione prenda in considerazione un segnale di errore ottenuto filtrando l'errore effettivo mediante un filtro di *pesatura percettiva*, il cui andamento frequenziale è sostanzialmente *reciproco* rispetto a quello stimato del segnale[20] (vedi fig. 10.14), in modo da attenuare la rilevanza dell'errore di predizione nelle regioni dove c'è più segnale[21] ed esaltarla invece nelle regioni con meno segnale, sfruttando così il fenomeno percettivo noto come *mascheramento uditivo* (vedi pag. 300). Anche se per questa via l'energia totale del rumore è maggiore, l'effetto soggettivo è migliore.

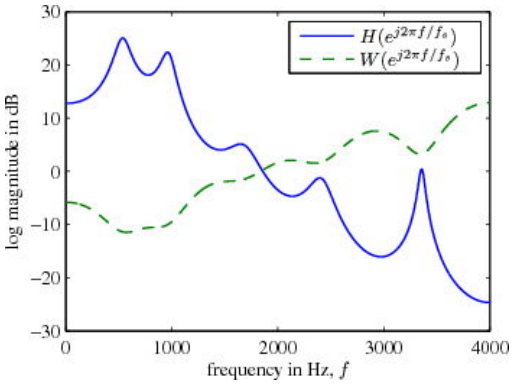


Figura 10.14: Spettro LPC vocalico e relativo filtro di pesatura percettiva dell'errore di predizione

Multi pulse linear prediction - MPLP

Lo schema operativo suggerito dalla tecnica ABS è stato inizialmente realizzato cercando di *costruire* la sequenza di eccitazione ottima (ossia in grado di minimizzare l'errore pesato percettivamente) come una sequenza di pochi impulsi sparsi, decidendone uno alla volta. Tale approccio prevede dunque di trovare l'ampiezza e posizione *ottime* per un unico primo impulso, quindi per un secondo (con il primo fisso), e così via, fino al numero di impulsi desiderati, tipicamente 4-5 ogni 5 msec, ottenuti suddividendo una finestra di 20 msec in quattro sotto-trame, ognuna con 40 campioni, se $f_c = 8000$ Hz.

Regular pulse excitation with long-term prediction - RPE-LTP o GSM 6.10 Il metodo MPLP presentava una complessità proibitiva, ma ha dato luogo alla versione semplificata RPE-LTP usata inizialmente nella telefonia GSM per fornire una velocità di 13 kbps. In questo caso, dopo aver determinato la posizione del primo impulso nella sottofinestra ne sono piazzati altri 9, ad intervalli regolari (un campione si e tre no), in modo che l'ottimizzazione riguardi solo i valori delle ampiezze.

Rispetto allo schema di fig. 10.13 viene inoltre aggiunto un *predittore a lungo termine* o LTP, utilizzato per rimuovere dal segnale di eccitazione l'eventuale periodicità caratteristica dei suoni vocalici, e stimato a partire da sotto-finestre consecutive (vedi fig. 10.15). Il filtro LTP in essenza consiste in un semplice ritardo pari al periodo di pitch

[20] Il filtro di pesatura percettiva si ottiene a partire dagli stessi coefficienti di predizione a_i che descrivono l'andamento spettrale della finestra di segnale, definendo la sua trasformata zeta come $W(z) = \frac{A(z/\alpha_1)}{A(z/\alpha_2)} = \frac{H(z/\alpha_2)}{H(z/\alpha_1)}$ in cui, se $\alpha_{1,2}$ sono numeri reali, i poli di $W(z)$ si trovano alle stesse frequenze di quelli di $H(z)$ ma con raggio α_2 volte maggiore, così come gli zeri di $W(z)$ hanno modulo α_1 volte maggiore. Scegliendo $0 < \alpha_{1,2} < 1$ e $\alpha_1 > \alpha_2$ per la $W(z)$ si ottiene l'effetto desiderato, e mostrato in fig. 10.14

[21] La procedura di minimizzazione determina una eccitazione tale da rendere bianco il residuo al suo ingresso; dato però che questo ha subito il filtraggio da parte di $W(z)$, significa che le frequenze attenuate da $W(z)$ sono in realtà *enfatizzate* nel segnale di errore reale.

(e dunque $\gg p$), ed il predittore LTP relativo (vedi lo schema di decodifica) *ripropone* in uscita una copia ritardata ed attenuata dell'uscita stessa. Il codificatore GSM pertanto determina ritardo e attenuazione dell'LTP in base all'analisi del residuo di predizione LPC[22], e lo usa per reintrodurre la componente periodica nella sequenza RPE di cui si sta valutando l'idoneità. Una volta che al residuo LPC viene sottratta la componente predicibile per tramite del LTP, ciò che rimane risulta effettivamente assimilabile ad un rumore, ed è indicato anche come *processo di innovazione*.

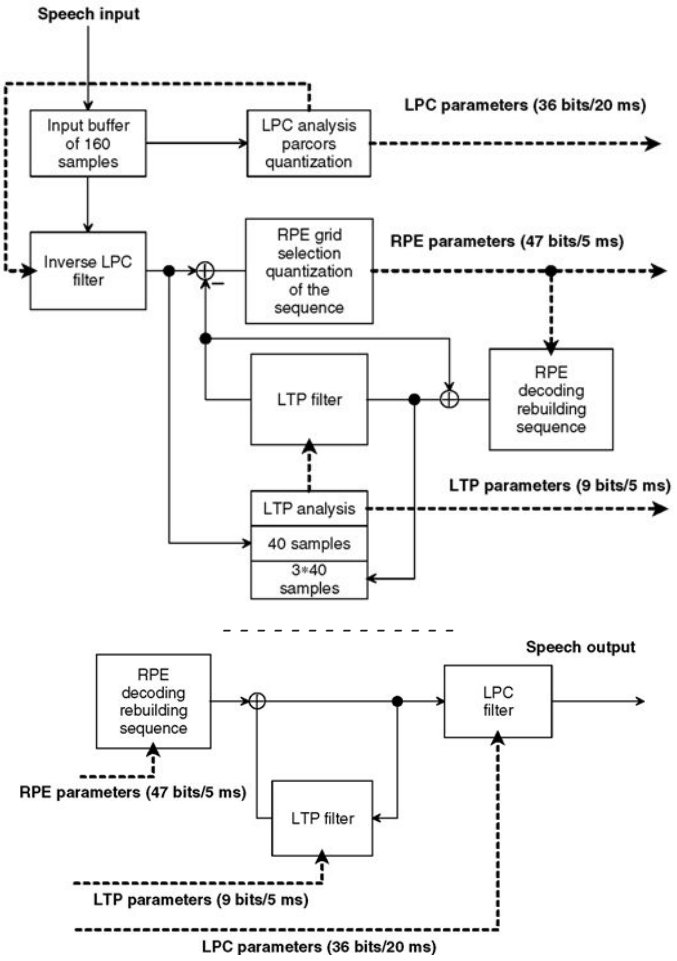


Figura 10.15: Codifica e decodifica GSM 6.10 *full rate* o RPE-LTP

10.1.2.4 Quantizzazione vettoriale dell'eccitazione

Dato che la codifica della sequenza di eccitazione impegna la maggior parte dei bit da trasmettere, si è fatta strada l'idea di... non codificarla affatto! Invece, viene realizzato un dizionario o *codebook* di *possibili* sequenze di eccitazione, e per ciascuna delle quali viene misurata *la distanza* tra essa e la sequenza *vera*. Ciò che viene trasmesso è quindi *l'indice* della codeword di minima distanza rispetto alla sequenza di eccitazione, e l'intero procedimento prende il nome di *quantizzazione vettoriale*[23].

[22] In effetti, mentre i coefficienti spettrali (denominati *parcor* in questo caso) sono determinati a partire dall'analisi dell'intera finestra di 20 msec, l'eccitazione RPE ed i parametri LTP sono ottenuti a partire da *sottofinestre* di 40 campioni, pari a 5 msec.

[23] Vedi ad es. `https://en.wikipedia.org/wiki/Vector_quantization`

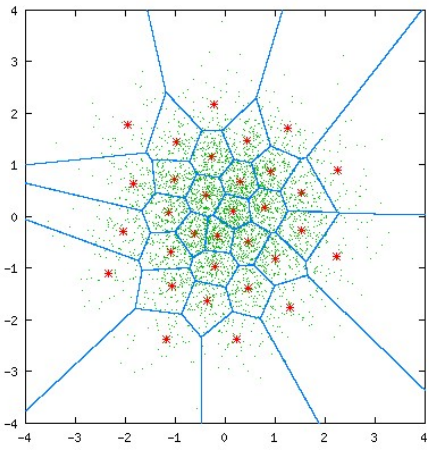


Figura 10.16: Regioni di decisione e centroidi per un quantizzatore vettoriale bidimensionale

La costruzione del codebook è ottenuta *partizionando* la distribuzione campionaria dei vettori in più regioni di decisione come quelle mostrate nell'esempio di fig. 10.16, in modo che ciascun vettore possa essere classificato[24] come interno ad una di esse, e venire quindi rappresentato dal *centroide* (i punti rossi) associato alla regione. I centroidi ed i confini di decisione sono determinati mediante un procedimento iterativo tale da minimizzare l'errore quadratico medio di rappresentazione[25].

I valori che descrivono le sequenze di eccitazione (vettori) associate ai centroidi del *codebook* sono noti anche al lato di ricezione, in modo che ogni particolare sequenza possa essere rappresentata, anziché da tutti i suoi campioni, dal solo indice della *codeword* del centroide più vicino: al solito, utilizzando M bit per rappresentare l'indice, il codebook sarà formato da 2^M diverse codeword. Oltre al codebook utilizzato per rappresentare le sequenze di innovazione, la codifica del segnale vocale si può avvantaggiare anche di un secondo codebook, usato per approssimare il vettore dei possibili coefficienti spettrali.

10.1.2.5 Predizione lineare ad eccitazione codificata - CELP

La fig. 10.17 mostra lo schema realizzativo di un codificatore CELP, in cui sono evidenziati il filtro di predizione a lungo termine ed il filtro LPC, stimati in modalità *ad anello aperto*, ed il *filtro percettivo* che fa in modo che la densità spettrale dell'errore di predizione sia concentrata nelle regioni dove è presente segnale.

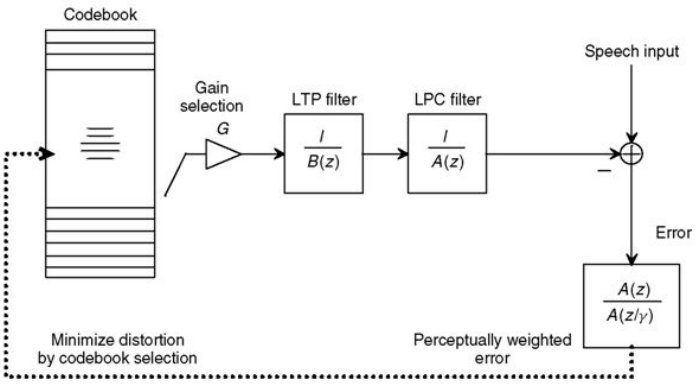


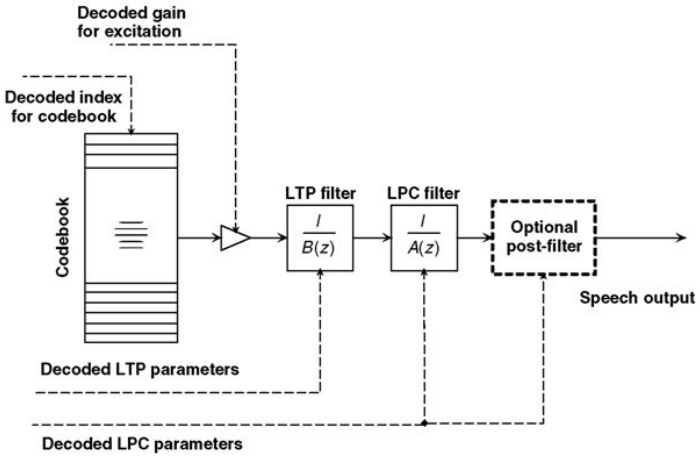
Figura 10.17: Codificatore e decodificatore CELP

Per ogni codeword di eccitazione selezionata dal codebook, e per il guadagno associato, viene calcolata l'energia dell'errore ottenuto, ed il risultato confrontato con

[24] Per questa classificazione, così come per poter definire l'insieme dei centroidi, occorre che sia definita una funzione di *distanza* tra vettori.

[25] Vedi `http://www.data-compression.com/vq.html` *(al 11/2021 non sembra rispondere)*, ma anche la nota 26 a pag. 100, così come `https://en.wikipedia.org/wiki/K-means_clustering`

quello ottenibile mediante le altre codeword, finché non si trova la codeword che minimizza l'errore. Ovviamente questo modo di procedere è estremamente oneroso, ma si sono trovati metodi di ricerca più efficienti adottando tecniche di costruzione del codebook come combinazione di sequenze elementari, dando luogo alla famiglia dei codificatori *algebrici* o ACELP[26].



D'altra parte, anche l'identificazione del LTP può essere ricondotta ad una ricerca ad anello chiuso, stavolta nell'ambito di un *codebook adattivo*, costruito a partire dalla precedente sequenza di eccitazione ottima, replicata in forma traslata di un campione alla volta, come illustrato in fig. 10.18, che mostra appunto l'uso della eccitazione per la trama precedente per popolare il codebook adattativo: da questo viene quindi individuata la codeword I_a ed il guadagno G_a ottimi, e quindi individuata la codeword di innovazione I_s e G_s ottimi, riferiti ad un codebook detto *stocastico* perché costituito da sequenze pseudo casuali. Infine, in fig. 10.18 viene mostrato come anche i coefficienti spettrali LPC sono trasmessi mediante una codeword (LSP o *line spectrum pair*) derivata da un processo di quantizzazione vettoriale. Possiamo elencare i seguenti standard che adottano una tecnica di questo tipo:

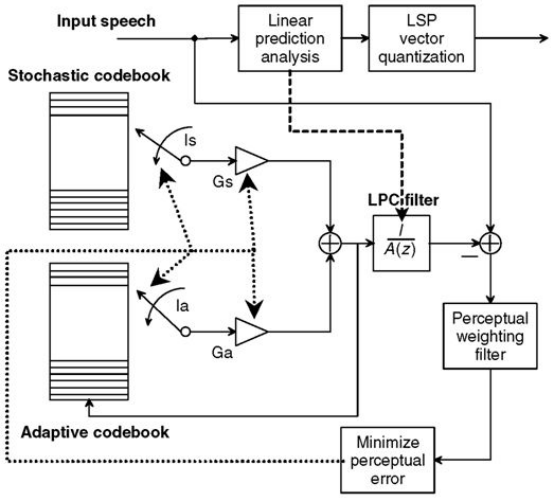


Figura 10.18: Codificatore CELP con codebook adattivo per la predizione a lungo termine

- Federal Standard 1016 (4800-16000 bit/s) CELP
- ITU-T 8-kbit/s G.729 CS-ACELP (*conjugate-structure algebraic* CELP);
- dual-rate multimedia ITU-T G.723.1 a 5.3 kbit/s con ACELP e 6.3 kbit/s con MP-MLQ (*multi-pulse maximum likelihood quantization*);
- ITU-T low-delay CELP 16-kbit/s G.728 - usa finestre di analisi multo brevi e una predizione lineare all'indietro per conseguire un ritardo di 2 msec;

[26] Vedi ad es. `https://en.wikipedia.org/wiki/Algebraic_code-excited_linear_prediction`

- ETSI enhanced full-rate EFR-GSM e half-rate HR-GSM, con velocità di 12.2 e 5.6 kbps, così come i codec AMR (*adaptive multirate*) e WB-AMR, con velocità da 7.95 a 4.75 kbps;
- Speex[27] - un insieme di codecs open source esenti da brevetti e liberamente utilizzabili, con velocità (a banda stretta) da 5,95 a 24,6 kbps, e da 5.75 a 42,4 kbps per segnali con banda di 16 kHz

10.1.3 Codifica psicoacustica

Mentre la codifica di forma d'onda (§ 10.1.1) non fa assunzioni a riguardo della natura del segnale, i metodi esposti al § 10.1.2 sono tutti fortemente orientati a rappresentare segnali vocali. Viceversa, il gruppo di lavoro MPEG di ISO si è dedicato ad individuare metodi di codifica idonei alla trasmissione di segnali multimediali di natura qualsiasi, come ad esempio brani musicali. Inoltre, i vincoli relativi al basso ritardo necessario ad assicurare un buon grado di interattività vengono meno, e si possono dunque intraprendere elaborazioni più complesse, e che richiedono un tempo maggiore. Infine, vengono trascurati rigidi vincoli sulla velocità risultante, accettando invece che questa *vari* nel tempo in funzione del tipo di segnale da rappresentare.

Come vedremo tra breve, per queste tecniche si fa di nuovo uso di una codifica per sottobande, introdotta nella discussione dell'ADPCM, tenendo però anche conto di caratteristiche molto importanti della percezione sonora, il cui sfruttamento è già stato illustrato nella discussione del filtro di pesatura percettiva, ma che ora hanno un impatto ancora maggiore sulla realizzazione del codificatore. I codificatori che fanno uso di queste caratteristiche sono l'MPEG *layer* 3 o MP3, il *Dolby* AC, e l'*advanced audio coding* o AAC.

Sensibilità uditiva e mascheramento in frequenza La fig. 10.19-a mostra la curva di sensibilità del sistema uditivo, ovvero il livello di intensità minimo perché possa essere percepito un suono: come si vede, questo è molto variabile con la frequenza, per cui anche se il suono B (sinusoide o tono puro) ha la stessa intensità di A non può essere udito, mentre invece A si. Ma ad una analisi più approfondita, si scopre che la presenza di un suono in una determinata regione di frequenza ha l'effetto di

[27] vedi `http://en.wikipedia.org/wiki/Speex`

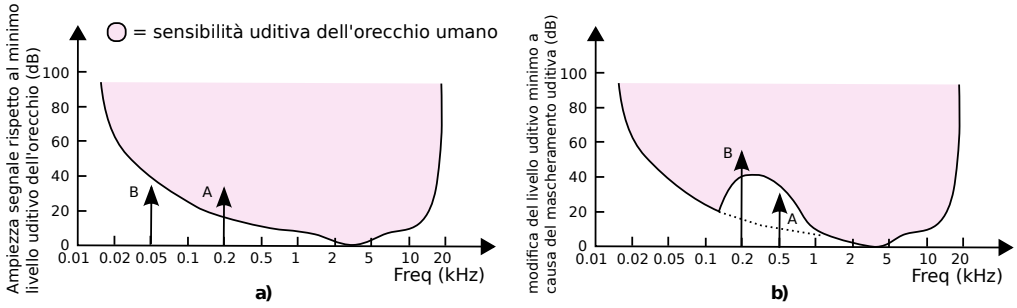


Figura 10.19: a) sensibilità uditiva alle diverse frequenze; b) mascheramento uditivo

modificare la curva di sensibilità per le frequenze vicine, di fatto *mascherando* suoni a frequenze vicini che altrimenti avrebbero superato la soglia di sensibilità, come mostrato in fig. 10.19-b: la presenza del suono B rende A non più udibile.

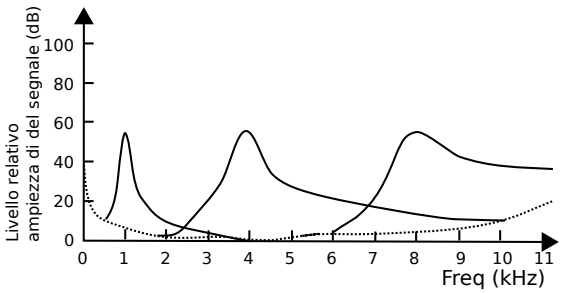


Figura 10.20: Variazione della banda critica in funzione della frequenza

In realtà, l'estensione in frequenza per cui si verifica l'effetto di mascheramento dipende sia dalla frequenza del tono mascherante (come mostrato dalle curve in fig. 10.20 ottenute con toni a 1, 4 ed 8 kHz) che dalla sua intensità. In particolare, la banda delle frequenze mascherate viene detta *banda critica* ed ha una estensione differente alle diverse frequenze: si trova che sotto i 500 Hz la banda critica ha una estensione di circa 100 Hz, mentre a frequenze superiori aumenta (circa) linearmente per multipli di 100 Hz. Ad esempio, un segnale ad 1 KHz (2x500) produce una banda critica di 200 Hz (2x100), mentre a 5 kHz (10x500) questa vale circa 1 kHz (10x100).

Mascheramento temporale Il secondo effetto percettivo riguarda ancora una modifica alle curve di sensibilità, stavolta in modo *non selettivo* in frequenza, ma che coinvolge tutte le frequenze: si verifica infatti che dopo aver udito un suono forte, per il tempo necessario all'estinzione del suono e che tipicamente dura qualche decina di millisecondi (vedi fig. 10.21), l'orecchio non è più in grado di percepire suoni con intensità minore a quello che si sta estinguendo.

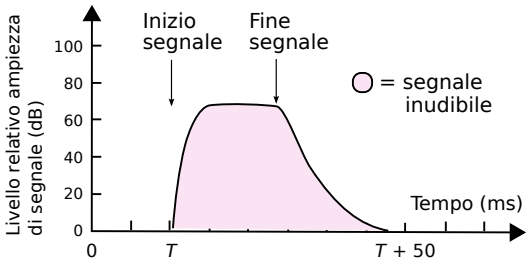


Figura 10.21: Mascheramento temporale

MPEG layer 3 Il gruppo di lavoro MPEG di ISO ha definito uno standard di codifica audio basato su tre livelli di complessità (e potere di compressione) crescente, ed il terzo (o MP3) è quello di gran lunga più popolare, anche grazie alla diffusione che ha avuto via Internet. Lo schema di funzionamento di principio è mostrato in fig. 10.22: il segnale campionato in ingresso (a 32, 44.1 o 48 kHz) transita attraverso un *PCM encoder* che esegue un filtraggio[28] in 32 sottobande di eguale ampiezza, le cui uscite

[28] Eseguito mediante un banco di *filtri polifase*, vedi ad es. `http://en.wikipedia.org/wiki/Polyphase_quadrature_filter` o `http://cnx.org/content/m32148/latest/`. Le uscite dei filtri polifase, anche se campionate a frequenza inferiore della velocità di Nyquist, sono esenti da aliasing, che viene cancellato dall'effetto delle altre sottobande.

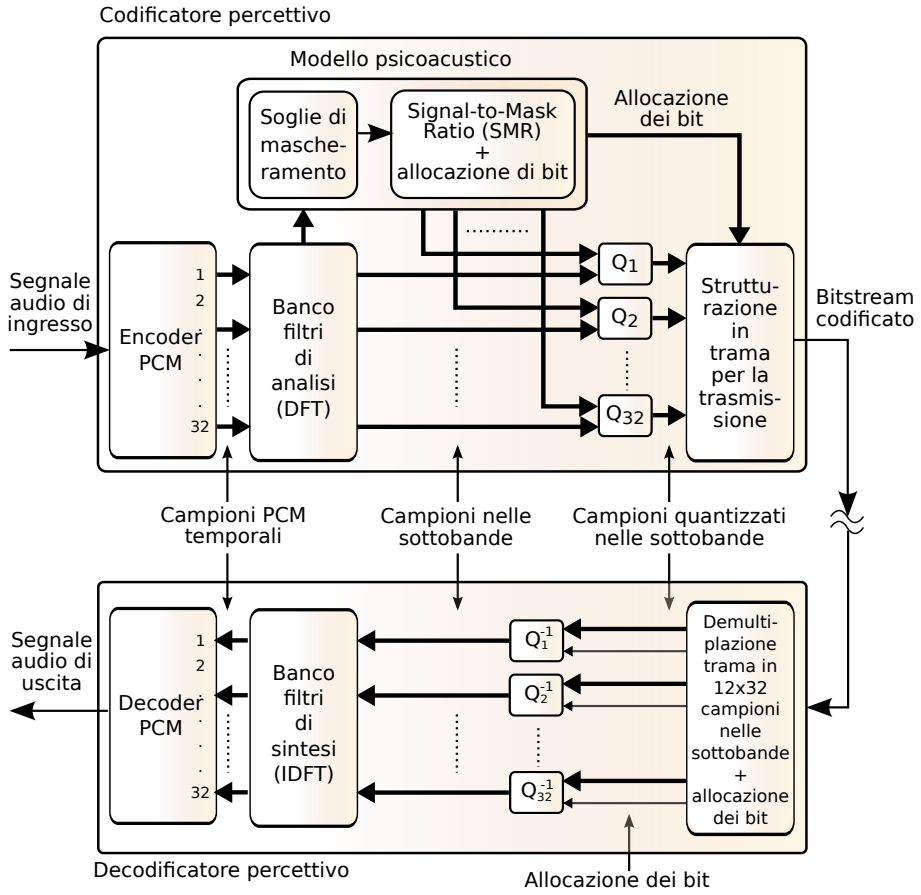


Figura 10.22: Codec percettivo MPEG

sono campionate a frequenza $1/32$ di quella di ingresso. Ogni 384 campioni di ingresso (pari a 12 msec se $f_c = 32$ kHz) sono quindi prodotti $384/32 = 12$ campioni per ogni sotto-banda, e per ognuna di esse è individuato il valore del campione più grande, che contribuisce sia ad impostare la dinamica del quantizzatore per quella banda, sia come parametro per il modello psicoacustico.

Il modello psicoacustico riceve le informazioni prodotte da un banco di filtri di analisi realizzati mediante una MDCT,[29] che produce una stima spettrale con risoluzione maggiore di quella del primo banco di filtri, su cui basare le valutazioni di mascheramento uditivo, che a loro volta determinano per ogni sottobanda l'indicazione di un *signal to mask ratio* (*SMR*), che a sua volta determina *quanti bit utilizzare* (e quindi quanti livelli) per la quantizzazione Q dei campioni relativi alle singole sottobande. Quelle contraddistinte da una maggiore sensibilità (ovvero nelle quali si percepiscono anche suoni deboli) saranno quantizzate con più accuratezza, e quindi con più bit e meno rumore; mentre le sottobande caratterizzate da una sensibilità inferiore possono essere quantizzate con meno bit, almeno finché l'*SNR* di quantizzazione si mantiene

[29] Vedi `http://en.wikipedia.org/wiki/Modified_discrete_cosine_transform`

superiore all'*SMR*, dato che in tal caso il rumore è mascherato, e dunque non viene udito. Quindi, i 12 campioni delle 32 sottobande sono quantizzati tenendo conto sia della dinamica effettiva, che del numero di livelli in cui suddividere la dinamica. Infine, viene prodotta una struttura di trama che contiene, oltre ai campioni, anche le informazioni sulla effettiva allocazione dei bit.

Ad una futura edizione, una trattazione più approfondita.

Riferimenti Si citano dei riferimenti essenziali sulla codifica audio, da cui sono anche tratte alcune illustrazioni

- Introduction to Digital Speech Processing, L. R. Rabiner and R. W. Schafer, `https://books.google.it/books?id=Z6Otr8Hj1WsC`
- Beyond VoIP Protocols: Understanding Voice Technology And Networking, O. Hersent, J.P. Petit, D. Gurle `http://what-when-how.com/category/voip-protocols/`
- `http://www.data-compression.com/index.shtml`
- Voice Acoustics: an introduction - University of New South Wales, J. Wolfe, M. Garnier, J. Smith `http://www.phys.unsw.edu.au/jw/voice.html`
- Let's build an MP3-decoder! - Björn Edström `http://blog.bjrn.se/2008/10/lets-build-mp3-decoder.html`

10.2 Codifica di immagine

Un segnale di immagine può essere di natura *vettoriale*[30], come nel caso di un disegno prodotto da un *plotter*, e rappresentato mediante un linguaggio descrittivo che codifica le operazioni grafiche necessarie alla sua realizzazione; al contrario, un segnale di immagine è detto di tipo *bitmap*, o *raster* (griglia, reticolo), quando è il risultato di un campionamento spaziale, come nel caso di una foto digitale, di un fax, o del risultato di un processo di scansionamento ottico. Mentre le immagini vettoriali sono pienamente *scalabili* e ridimensionabili senza perdita di definizione, quelle bitmap sono ottimizzate per essere riprodotte nelle loro dimensioni originali, avendo già operato un processo di distorsione tale da sfruttare al più possibile le caratteristiche di predicibilità e di sensibilità percettiva.

10.2.1 Dimensioni

Per quanto riguarda le immagini bitmap, queste sono definite nei termini di una matrice di elementi di immagine o PIXEL (*picture elements*)[31], che sono l'equivalente bidimensionale dei campioni estratti da un segnale unidimensionale. Per ogni pixel è definito un valore associato alla intensità con la quale deve essere riprodotto: nel caso di immagini a colori, sono necessari tre valori di intensità, per cui una immagine è in realtà descritta da tre matrici, come approfondiamo di seguito.

Sebbene le dimensioni della matrice di pixel possano essere qualunque, nel corso del tempo si sono affermati una serie di valori di riferimento, associati ad altrettante

[30] Esempi di formati per la grafica vettoriale sono PDF, EPS, PDF, e VRML.

[31] Per alcuni anni, si è usato come sinonimo anche il termine PEL, vedi ad es. `http://www.foveon.com/files/ABriefHistoryofPixel2.pdf`.

	banda	*linee*	*fps*	*aspetto*	*colonne*	*righe*	*colore*
PAL	6 MHz	625	25 int	4:3			
NTSC	5 MHz	525	30 int	4:3			
HDTV		1080		4:3	1440	1152	
				16:9	1920	1152	
PDFA				4:3	1024	768	
4:2:2		625/525	50/60	4:3	720	576/480	360 x
			non int				576/480
4:2:0		625/525	25/30	4:3	720	576/480	360 x
			int				288/240
VGA				4:3	640	480	
SIF		625/525	25/30	4:3	360	288/240	180 x
			non int				144/120
CIF			30	4:3	360	288	180 x 144
			non int				
QCIF			15:7.5	4:3	180	144	90 x 72
			non int				

Tabella 10.1: Griglia dei parametri corrispondenti ai formati video

sigle, legate al tipo di dispositivo che deve poi riprodurre l'immagine, ma anche a quello da cui l'immagine viene acquisita; la tabella 10.1 riassume tali corrispondenze.

Ad esempio, la risoluzione VGA (640 x 480) trae origine dai parametri dello standard NTSC della televisione analogica (§ 25.1), i cui quadri sono composti da una serie di 525 linee, di cui solo 480 visibili: volendo mantenere una risoluzione orizzontale pari a quella verticale, con un rapporto d'aspetto di 4:3, ogni linea deve essere campionata su 480/3x4 = 640 punti. Già prima dell'uso broadcast della TV digitale, la raccomandazione BT 601[32] ha stabilito le regole per la conversione tra standard video differenti, mediante l'uso di una comune frequenza di campionamento del segnale video a 13.5 MHz, individuando così nei 52 μsec ([33]) di una linea, un numero di 52 x 10^{-6}x 13.5 x 10^{6} = 702 campioni per linea, a cui si aggiungono 9 campioni neri in testa ed in coda per ottenere 720 campioni per linea; per un segnale a 525 linee si ottiene quindi la matrice 720 x 480 del formato 4:2:2, che approfondiremo tra breve.

Le matrici più grandi di 1024 x 768 sono spesso descritte in termini di *Megapixel* (es 1600 x 1200 = 1,9 Mpixel), spesso usati per confrontare la risoluzione (ma non necessariamente la qualità) dei mezzi di fotografia digitale; inoltre, i *grandi formati* traggono origine anche dalla tecnologia delle schede video per computer da un lato, e da quella della televisione ad alta definizione da un altro, come riassunto nella figura 10.23[34].

[32] Il sito di ITU-R `http://www.itu.int/ITU-R/index.asp?category=information&link=rec-601&lang=en` non consente l'accesso pubblico alla raccomandazione. Un approfondimento può essere svolto presso Wikipedia `http://it.wikipedia.org/wiki/BT.601`.

[33] Vedi fig. 25.2 a pag. 856.

[34] La figura è tratta da Wikipedia, dove possono essere approfonditi gli altri aspetti legati a queste

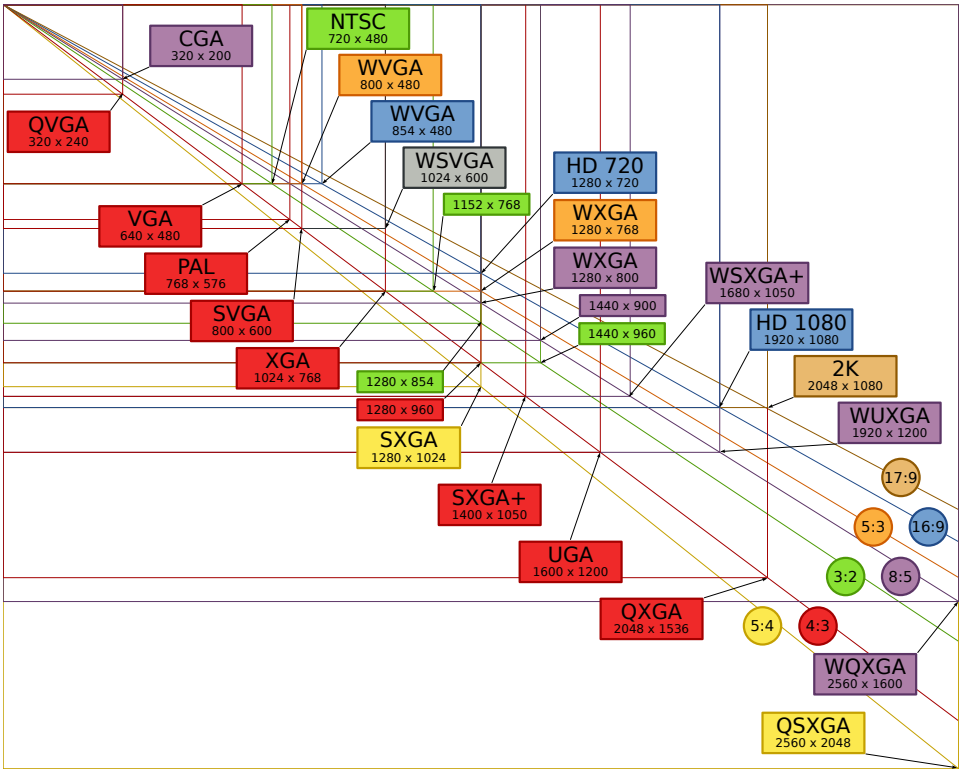


Figura 10.23: Risoluzioni standard o modalità video digitale

Il formato SIF (*source intermediate format*) è ottenuto a partire dal 4:2:2, conservando la metà dei pixel sia in verticale che in orizzontale, e trascurando la metà dei quadri di immagine; il suo uso è orientato alla memorizzazione, e quindi usa una scansione non interallacciata. Il formato CIF (*common intermediate format*) è simile al SIF, tranne per aver perso il riferimento al numero di linee analogiche da cui deriva; il suo uso è orientato ai sistemi di videoconferenza, e da questo sono definiti formati a maggior risoluzione, come il 4CIF ed il 16CIF, equivalenti al 4:2:2 ed all'HDTV. Il formato QCIF (*quarter CIF*) è orientato alla videotelefonia, dimezzando ancora sia la risoluzione spaziale che quella temporale. Da questo è a sua volta derivato il formato SUB-QCIF (o S-QCIF) di 128 x 96 pixel, orientato a collegamenti lenti come quelli via modem.

10.2.2 Spazio dei colori

I dispositivi di acquisizione e riproduzione di immagini a colori operano su tre diverse matrici di pixel, che rappresentano i tre colori di base della *sintesi additiva*, ossia *rosso*, *verde*, e *blu*, o RGB (dalle iniziali inglesi *Red*, *Green* e *Blue*). In figura 10.24 viene mostrato il principio di funzionamento di un *prisma dicroico*, che devia le tre componenti di colore verso tre diversi dispositivi di acquisizione. Variando quindi la proporzione con cui si sommano gli stimoli dei tre colori, si ottiene, oltre al bianco,

risoluzioni video `https://it.wikipedia.org/wiki/Risoluzione_dello_schermo`.

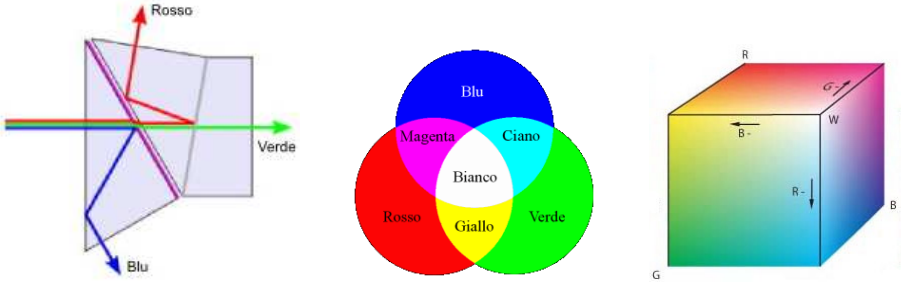


Figura 10.24: Prisma dicroico, sintesi cromatica additiva, cubo dei colori

anche qualunque altro colore. Sebbene dalle figure riportate sembra che il bianco risulti dal contributo in parti uguali delle tre componenti RGB, in realtà la scala di grigi della immagine *monocromatica* corrispondente si ottiene calcolando un segnale Y di *luminanza* secondo la formula

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{10.5}$$

che è quella usata per modulare il segnale video analogico[35]. Come già discusso, in tale ambito la componente di colore viene trasmessa utilizzando due altri segnali, C_b o *crominanza blu* e C_r o *crominanza rossa*, secondo la formula

$$C_b = B - Y \quad \text{e} \quad C_r = R - Y \tag{10.6}$$

Disponendo dei segnali Y, C_b e C_r, si possono riottenere i valori RGB inserendo la (10.5) nelle (10.6), e risolvendo il sistema di tre equazioni in tre incognite risultante.

Segnale video composito Al § 25.1 abbiamo descritto come nel segnale televisivo analogico la componente di colore sia trasmessa assieme alla luminanza, su di una diversa portante, con modulazione di ampiezza in fase e quadratura. In realtà, per diversi motivi le componenti trasmesse non sono direttamente quelle individuate dalle (10.6), ma piuttosto componenti denominate U, V oppure I, Q, e così definite:

$$\begin{array}{lrcl@{\qquad}lrcl} PAL: & U & = & 0.493 \cdot C_b & NTSC: & I & = & 0.74 \cdot C_r - 0.27 \cdot C_b \\ & V & = & 0.877 \cdot C_r & & Q & = & 0.48 \cdot C_r + 0.41 \cdot C_b \end{array}$$

Pertanto, in funzione delle diverse modalità di rappresentazione, un segnale video a colori può essere descritto indifferentemente da una delle seguenti quattro terne di segnali: RGB, YC_rC_b, YUV, YIQ.

Una descrizione alternativa dello spazio di colore è fornita dai parametri di *tinta*, *saturazione* e *luminosità*, ovvero HUE, SATURATION e LIGHTNESS, o HSL: si tratta di attributi legati più alla descrizione percettiva che non alle tecnologie della riproduzione dell'immagine. Mentre la tinta descrive una famiglia di colori (es tutti i rossi), la saturazione ne indica il grado di purezza, ossia la presenza congiunta di altre tonalità; la chiarezza, infine, denota la luminosità del colore, rispetto ad un punto bianco. La terna HSL viene a volte usata per descrivere un colore nell'ambito di programmi di *computer graphic*, mediante i quali è fornito anche l'equivalente RGB.

[35] Vedi nota 8 a pag. 858.

Profondità di colore Dato che l'occhio umano non distingue più di 250 tinte diverse, e di 100 livelli di saturazione, si ritiene che utilizzare 8 bit per ogni componente dello spazio di colore RGB sia più che sufficiente. Con 8x3=24 bit per pixel (bpp) si possono infatti rappresentare $2^{24} - 1$ diversi colori, ovvero più di 16 milioni, molti dei quali indistinguibili ad occhio nudo. Modalità più spinte di quella a 24 bpp (detta *truecolor*) adottano 10, 12, 16 bit/componente, o rappresentazioni in virgola mobile, e sebbene non migliorino la qualità visiva, possono comunque essere usate in contesti professionali, per non perdere precisione nelle operazioni di editing ripetuto. Al contrario, profondità inferiori sono comunemente usate per risparmiare memoria, come nel caso di 15 bpp, che usa 5 bit per componente, o 16 bpp, che usa 6 bit per il verde, offrendo 65.536 colori diversi.

Palette Nel caso si decida di adottare profondità molto ridotte, come 8 bpp, si preferisce ricorrere ad una modalità detta a *colore indicizzato*: l'insieme dei colori presenti nell'immagine viene *quantizzato*[36] in un insieme ridotto, i cui valori a 24 bpp sono memorizzati in una tavolozza (la *palette* detta anche *colour look-up table* o CLUT), che viene quindi utilizzata come un dizionario. La figura a lato mostra una immagine di esempio, assieme alla palette dei colori che usa. In questo modo, per ogni pixel dell'immagine è ora sufficiente specificare l'indice della palette dove è memorizzata la rappresentazione a 24 bpp del colore più prossimo.

Esempio Consideriamo una immagine in formato VGA rappresentata mediante una palette di 256 elementi da 24 bit: ognuno dei 640x480 = 307.200 pixel può quindi assumere uno tra 256 diversi colori, scelti tra 2^{24} = 16 milioni. La dimensione di memoria occupata si ottiene considerando che per ogni pixel occorrono 8 bit per l'indice nella palette, e che la palette stessa ha dimensioni 256x24 = 6144 bit = 768 byte, e quindi in totale 307.968 byte.

Sottocampionamento del colore Nella tabella riportata a pag. 304 è presente la colonna *colore*, che mostra come la dimensione riservata alle matrici di pixel che codificano le informazioni di crominanza sia ridotta di metà rispetto a quella della luminanza. Questo fatto trae origine da due buoni motivi: il primo è che l'acutezza visiva dell'occhio umano per ciò che riguarda le variazioni cromatiche è ridotta rispetto a quella relativa alle variazioni di luminosità; il secondo è che il segnale di crominanza presente nel segnale video composito occupa una banda circa metà di quella del segnale di luminanza. Pertanto, le componenti di luminanza sono generalmente campionate con una risoluzione spaziale inferiore a quella del segnale di luminanza. Il tipo di sotto-campionamento spaziale adottato per le componenti di crominanza è generalmente caratterizzato da quattro numeri, in accordo allo schema di fig. 10.25:

[36] Per una breve introduzione alla *quantizzazione cromatica*, può essere consultata Wikipedia `http://en.wikipedia.org/wiki/Color_quantization`

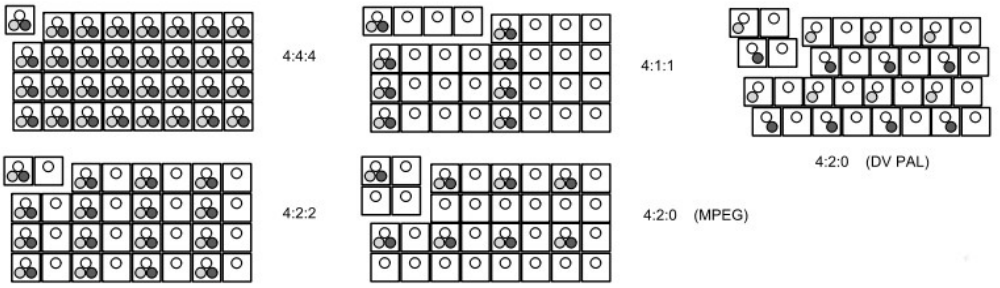


Figura 10.25: Sottocampionamento delle componenti di colore

- **4:4:4** - Non si effettua sottocampionamento, e le tre componenti hanno lo stesso numero di campioni. Applicato principalmente a segnali RGB trattati in studio di produzione.
- **4:2.2** - Questo schema si applica tipicamente alle rappresentazioni YC_bC_r, memorizzando per ogni 4 campioni di luminanza, 2 campioni della componente C_b e 2 della componente C_r, ed è utilizzato in ambito professionale e broadcast.
- **4:1:1** - In questo caso ogni quattro campioni di luminanza su una riga, ne viene preso uno per C_b ed uno per C_r. E' lo schema usato nello standard DV NTSC.
- **4:2:0** - Ogni 4 campioni di luminanza, ne vengono salvati uno per C_b ed uno per C_r come per il caso 4:1:1, ma ora la crominanza è campionata su righe alterne. In particolare, la versione utilizzata per l'MPEG-1 campiona assieme entrambi i segnali di crominanza, una riga si ed una no, mentre quella usata con il DV PAL li campiona a righe alternate, e prevede una riproduzione in modalità interallacciata.

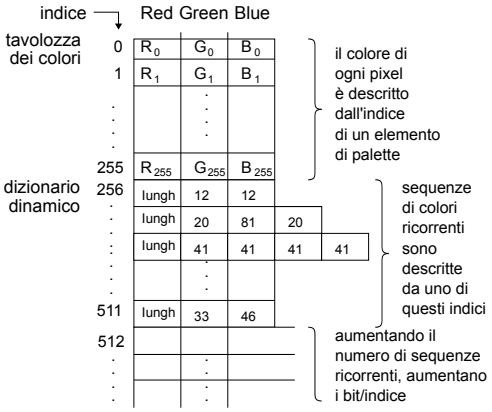
10.2.3 Formato GIF

Il *Graphics Interchange Format* è un formato ad 8 bpp definito da *CompuServe* nel 1987[37] e da allora ha continuato ad essere molto popolare. Usa una *palette* con cui rappresentare 256 colori scelti tra 16 milioni, e quindi comprime l'immagine mediante l'algoritmo LZW (§ 9.2.3.1), individuando sequenze ricorrenti dei valori di colore. Un singolo file può contenere più immagini (ognuna con la sua palette) in modo da realizzare brevi animazioni. Il numero ridotto di colori rende il formato poco idoneo alla riproduzione di fotografie, ma più che adatto ad immagini più semplici, come ad es. un logo di pagina web. Per rappresentare i colori assenti dalla palette, il codificatore può ricorrere ad una operazione di *dithering*, alternando colori che, osservati da lontano, ricreano l'effetto della tonalità mancante.

Il metodo di compressione è illustrato con l'ausilio della figura che segue, e adotta come anticipato l'algoritmo LZW, il cui dizionario è inizialmente composto dalla palette, o meglio dai 256 valori ad 8 bit che indicizzano la terna RGB a 24 bit nella palette. Quando si incontra una sequenza di codici di colore già osservata, viene aggiunta una

[37] Il documento di specifica può essere trovato presso W3C: `http://www.w3.org/Graphics/GIF/spec-gif89a.txt`

riga al dizionario, ed il valore dell'indice corrispondente viene usato per rappresentare tutta la sotto-sequenza; eventualmente, il numero di bit usati per indicare le righe del dizionario viene aumentato di uno. Per disegnare le sequenze di pixel rappresentate da indici inclusi nella sezione dinamica della tabella, occorre dunque individuare prima le rispettive terne RGB nella tavolozza.



PNG Dato che la compressione LZW era stata brevettata, venne sviluppata una codifica alternativa, denominata *Portable Network Graphics*. Al giorno d'oggi i brevetti relativi al formato GIF sono tutti scaduti, ed il formato PNG è stato standardizzato nella RFC 2083[38]. Come per GIF, anche PNG è di tipo *lossless* (senza perdite), ossia individua una compressione invertibile, capace di replicare in modo identico l'immagine di partenza, ovviamente senza considerare il processo di quantizzazione che porta alla generazione della palette. Oltre alla modalità di colore indicizzato, PNG offre anche una modalità *truecolor* a 24 o 32 bpp, e per questo può correttamente rappresentare anche materiale fotografico, al punto da consigliare l'uso di PNG (anziché JPEG) nel caso si prevedano successive operazioni di editing dell'immagine.

Per quanto riguarda la compressione, PNG fa uso dell'algoritmo *deflate*, preceduto da un passaggio di compressione differenziale, in cui al valore che rappresenta il colore di un pixel viene sottratto il valore predetto a partire dai pixel adiacenti: in tal modo l'algoritmo *deflate* riesce a conseguire rapporti di compressione più elevati, riuscendo quasi sempre a battere le prestazioni di GIF.

10.2.4 Codifica JPEG

Il *Joint Photographic Experts Group* è un comitato congiunto ISO/ITU che ha definito lo standard internazionale per la compressione di immagini ISO 10918-1[39], particolarmente adatto alla codifica di immagini fotografiche. Descriviamo di seguito il funzionamento della modalità operativa detta *baseline*, o *lossy sequential mode*, che è quella che offre il migliore grado di compressione, e che prevede cinque stadi di elaborazione, mostrati alla fig. 10.26: preparazione dei blocchi, Discrete Cosine Transform (DCT), quantizzazione, codifica entropica, e formattazione.

Preparazione dell'immagine e dei blocchi L'immagine *raster* di partenza è formata da una o più matrici bidimensionali di valori (scala di grigi, oppure a colori indicizzati, o RGB, YC_rC_b, YUV, ...), eventualmente di dimensioni differenti (come nel caso YC_rC_b). Sebbene sia possibile elaborare direttamente una rappresentazione RGB, le

[38] Reperibile presso il sito di IETF: `http://tools.ietf.org/html/rfc2083`
[39] Scaricabile presso il W3C: `http://www.w3.org/Graphics/JPEG/itu-t81.pdf`

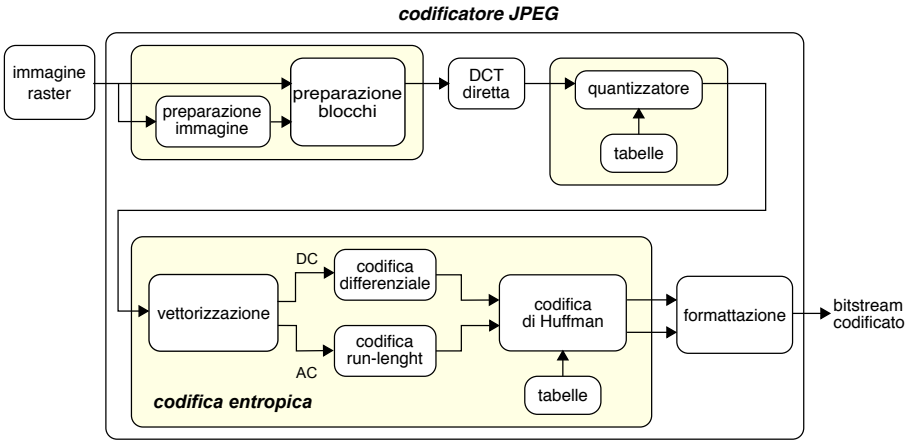


Figura 10.26: Stadi di elaborazione nella compressione jpeg

migliori prestazioni si ottengono nello spazio YC_rC_b con sotto-campionamento spaziale 4:2:2 o (meglio) 4:2:0, e dunque il primo passo è quello di convertire l'immagine in questa modalità di rappresentazione.

Ogni matrice viene quindi suddivisa in *blocchi* della dimensione di 8x8 pixel[40], ognuno dei quali è elaborato in sequenza in modo indipendente dagli altri.

DCT diretta Prima di procedere, la matrice Y (oppure le tre matrici R, G e B) che contiene valori ad 8 bit tutti positivi, viene normalizzata sottraendo ad ogni pixel il valore 128, in modo da ottenere valori tra -128 e 127. Quindi, per ogni blocco di 8x8 pixel, i cui valori indichiamo con $p(x, y)$, viene calcolata una nuova matrice di 8x8 valori $D(i, j)$ ottenuti come coefficienti di una *trasformata coseno discreta* (DCT) bidimensionale (vedi § 4.5.3):

$$D(i,j) = \frac{1}{4} c_i c_j \sum_{x=0}^{7} \sum_{y=0}^{7} p(x,y) \cos \frac{(2x+1)\, i\pi}{16} \cos \frac{(2y+1)\, j\pi}{16}$$

in cui c_i e c_j sono ognuno pari a $1/\sqrt{2}$ con indice i o j pari a zero, oppure $c_i = c_j = 1$ negli altri casi, mentre gli indici i e j variano tra zero e sette. Tralasciando di approfondire le relazioni esistenti tra DCT e DFT[41], consideriamo invece come i coefficienti $D(i, j)$ così ottenuti permettano la ricostruzione della matrice originaria nei termini di una somma pesata delle superfici rappresentate (per mezzo di una scala di grigi) nel diagramma riportato alla figura 10.27, mediante l'applicazione della DCT *inversa*

$$p(x,y) = \frac{1}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} c_i c_j D(i,j) \cos \frac{(2x+1)\, i\pi}{16} \cos \frac{(2y+1)\, j\pi}{16}$$

[40] Notiamo incidentalmente come le dimensioni definite nella tabella di pag 304 siano multipli interi di 8. Se questo non è il caso, i blocchi ai bordi destro ed inferiore vengono riempiti con pixel scelti in modo da minimizzare le distorsioni risultanti.

[41] Potremmo tentare comunque di estendere le considerazioni svolte al § 4.5.3 al caso bidimensionale...

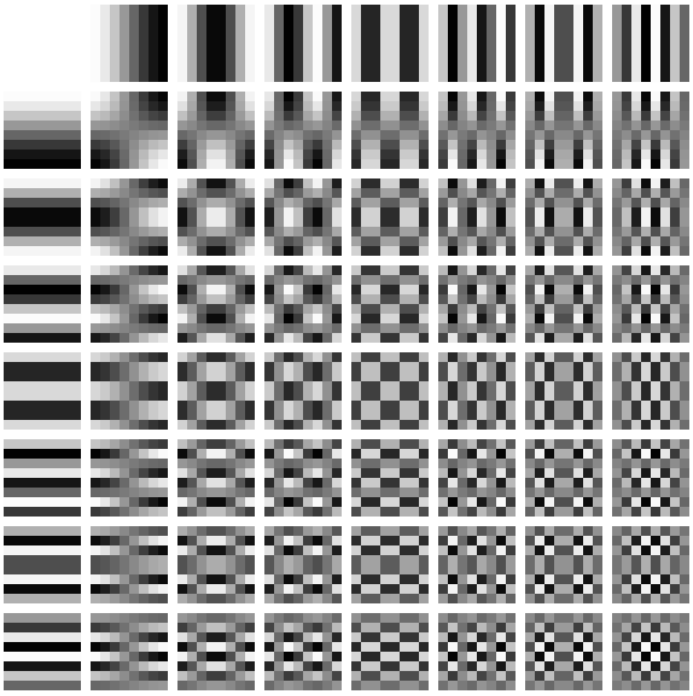


Figura 10.27: Grafico delle superfici 8×8 che costituiscono la base di rappresentazione DCT

Ma se fosse tutto qui, non avremmo realizzato la funzione di compressione! Questa è infatti realizzata dalle elaborazioni successive, a partire dalla rappresentazione in termini di blocchi DCT, di cui ora approfondiamo il significato. Osserviamo quindi che ognuna delle superfici elementari rappresentate in fig. 10.27 è legata ad una coppia i, j associata ad un coefficiente $D(i, j)$ della DCT calcolata, in modo che tale coefficiente esprime il contenuto di frequenze *spaziali* descritto da quella particolare funzione della base. Per questo l'elemento $(i, j) = (0, 0)$ in alto a sinistra, ad andamento costante, è indicato come *coefficiente DC*, o componente continua, dato che essendo calcolato come somma di tutti i pixel, riflette un valore che è legato alla intensità media dell'intero blocco. I coefficienti legati alle funzioni della prima riga rappresentano contenuti di frequenza spaziale orizzontale, con un periodo via via minore spostandosi verso il margine destro, mentre quelli della prima colonna, frequenze verticali. I coefficienti localizzati all'interno della matrice esprimono contenuti di frequenze spaziali in entrambe le direzioni, con valori di frequenza tanto più elevati, quanto più ci si sposta verso l'angolo in basso a destra. Pertanto, i coefficienti descritti da indici diversi da $(0, 0)$ sono indicati come *coefficienti AC*.

L'esperienza pratica mostra come quasi sempre i coefficienti $D(i, j)$ presentino nella regione in alto a sinistra valori ben più elevati di quelli riscontrabili in basso a destra, come conseguenza della predominanza dei blocchi posti in corrispondenza ad aree dell'immagine quasi costanti, rispetto a quelli associati alla presenza di contorni netti e particolari dettagliati.

Quantizzazione Questo passo della elaborazione JPEG mira a sfruttare il fenomeno percettivo della ridotta sensibilità dell'occhio umano alle frequenze spaziali più elevate, ovvero la capacità di *filtrare percettivamente* le componenti di errore corrispondenti ai dettagli più minuti. Per questo, il processo di quantizzazione è orientato a ridurre, ed eventualmente sopprimere, le componenti di immagine legate alle frequenze spaziali più elevate, introducendo di fatto *una soglia* sotto la quale si stabilisce di non trasmettere quelle informazioni che tanto non sarebbero percepibili. A questo scopo, ogni coefficiente $D(i,j)$ viene diviso per un coefficiente $Q(i,j)$ dipendente da (i,j), ed il risultato viene arrotondato:

$$B(i,j) = \text{round}\left(\frac{D(i,j)}{Q(i,j)}\right)$$

Il risultato corrisponde ad un processo di quantizzazione, perché quando in ricezione il processo viene invertito (ri-moltiplicando il coefficiente per la stessa quantità), viene persa la precisione legata all'arrotondamento, e pari alla metà del coefficiente di divisione. La scelta dei $Q(i,j)$ è fatta in modo tale da utilizzare valori più elevati per gli indici (i,j) più elevati, in modo da ottenere due risultati: ridurre le componenti ad alta variabilità *spaziale* dell'immagine, e poter usare meno bit per codificare questi valori (più piccoli). Inoltre, molti dei coefficienti con (i,j) elevato, già piccoli di per se, quando divisi per un coefficiente di quantizzazione più elevato, non *sopravvivono* all'operazione di arrotondamento, in modo che tipicamente la parte in basso a destra della matrice $B(i,j)$ sarà tutta pari a zero, facilitando il compito della codifica run-length dello stadio successivo.

Esempio La figura 10.28 mostra un esempio di matrice di coefficienti DCT, assieme alla tabella di quantizzazione, ed al risultato dell'operazione. Notiamo come il valore dei coefficienti di quantizzazione aumenti allontanandosi dal coefficiente DC, e come nella matrice dei coefficienti quantizzati siano *sopravvissuti* solo i coefficienti relativi alle frequenze spaziali più basse.

Sebbene esistano delle tabelle di quantizzazione predefinite, i valori effettivi possono essere variati in base ad un compromesso tra qualità che si intende conseguire e fattore di compressione; tali valori vengono poi acclusi assieme al bitstream codificato durante la fase di formattazione, in modo che il processo di quantizzazione possa essere invertito nella fase di riproduzione dell'immagine.

Codifica entropica Questo passo è un processo senza perdita, nel senso che non aggiunge altre distorsioni oltre a quelle introdotte dal passo di quantizzazione, ma è essenziale ai fini della compressione, e sfrutta le caratteristiche statistiche del risultato delle elaborazioni precedenti. Come posto in evidenza nello schema di fig. 10.26, la codifica entropica adotta due diverse procedure per i coefficienti DC e AC, che in entrambi i casi culminano con uno stadio di codifica a lunghezza variabile mediante codici di Huffman.

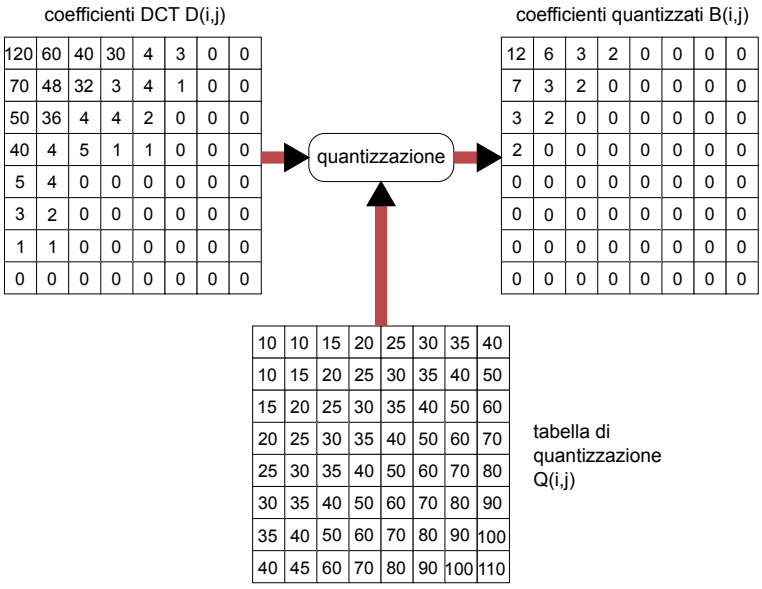
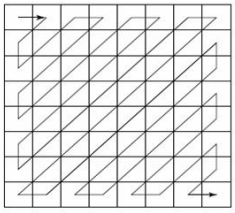


Figura 10.28: Processo di quantizzazione dei coefficienti DCT

Vettorizzazione Le matrici 8x8 relative ai blocchi di elaborazione visti fin qui vengono ora trasformate in sequenze *ad una dimensione* mediante un processo di scansione a *zig zag* dei blocchi, il cui percorso è illustrato alla figura seguente.



La sequenza così ottenuta presenta il coefficiente DC in testa, a cui fanno seguito i rimanenti 63 coefficienti AC, ordinati in base al massimo valore di frequenza spaziale che rappresentano. Se applichiamo la scansione zig-zag ai valori riportati nell'esempio di fig. 10.28 otteniamo come risultato la sequenza

12 6 7 3 3 3 2 2 2 2 0 0 0 ..... 0 0

Codifica differenziale Blocchi adiacenti generalmente possiedono coefficienti DC molto simili tra loro, in virtù dell'omogeneità di ampie zone dell'immagine (pensiamo ad un porzione di cielo). Per questo motivo, anziché codificarli in modo indipendente, i singoli coefficienti DC di blocchi consecutivi vengono sottratti l'uno all'altro, e viene codificata solo la loro differenza. Ad esempio, se una sequenza di coefficienti DC risultasse pari a 12 13 11 11 10 ..., il risultato di questo processo di codifica differenziale darebbe luogo alla sequenza 12 1 −2 0 −1 ... (di fatto, il valore *precedente* al primo coefficiente si assume pari a zero). Dato che differenze in valore assoluto piccole sono relativamente più frequenti di differenze grandi, si è scelto di adottare per queste una codifica a lunghezza di parola variabile, realizzata

- descrivendo innanzitutto ogni valore di differenza mediante una coppia (sss, valore), in cui sss rappresenta il numero di bit necessario per rappresentare il

valore, e quindi

- per ogni coppia (sss, valore) il termine sss è rappresentato mediante una codeword di Huffman, ed il valore con numero variabile di bit.

Esempio Per chiarire le idee, mostriamo le corrispondenze citate mediante due tabelle, che poi applichiamo al caso dell'esempio precedente.

differenza	*N. di bit sss*	*valore codificato*	
0	0		
-1, 1	1	1=1	-1=0
-3, -2, 2, 3	2	2=10	-2=01
		3=11	-3=00
-7...-4, 4...7	3	4=100	-4=011
		5=101	-5=010
		6=110	-6=001
		7=111	-7=000
-15...-8, 8...15	4	8=1000,	-8=0111
⋮		⋮	

sss	codeword di Huffman
0	010
1	011
2	100
3	00
4	101
5	110
6	1110
7	11110
⋮	⋮
11	111111110

Tornando dunque al nostro esempio della sequenza differenziale 12 1 −2 0 −1 ..., in termini di coppie (sss, valore) questa diviene (4, 12), (1, 1), (2, -2), (0, 0), (1, -1),... e quindi, sostituendo ad sss il relativo codice di Huffman preso dalla seconda colonna della seconda tabella, ed ai valori la loro rappresentazione indicata dalla terza colonna della prima tabella, otteniamo la sequenza di bit 101 1100, 011 1, 100 01, 010, 011 0,... in cui si sono mantenute le virgole per chiarezza. In definitiva, abbiamo usato un totale di 23 bit per rappresentare 5 differenze, che ne avrebbero richiesti 45 se codificate con 9 bit.

Codifica run-length Viene applicata alla sequenza di coefficienti AC che è il risultato dello *zig-zag scan*. In base all'effetto congiunto delle caratteristiche dei coefficienti della DCT, e del processo di quantizzazione, la sequenza degli AC in uscita dal vettorizzatore presenta lunghe sequenze di zeri, consentendo di conseguire buoni rapporti di compressione mediante l'uso di una codifica *run-length*, realizzata scrivendo gli AC come una sequenza di coppie *(skip, ACN)*, in cui *skip* rappresenta il numero di zeri nel run, e *ACN* è il coefficiente AC non nullo che viene dopo la sequenza di zeri. Quindi, il campo *ACN* viene espresso a sua volta nella forma *sss, valore*, come indicato dalla prima tabella riportata nell'ultimo esempio. Infine, la coppia *skip, sss* viene rappresentata con una codeword di Huffman individuata in un nuovo codebook appositamente definito.

Esempio Applicando la codifica run-length alla sequenza dei coefficienti AC individuati nell'esempio di vettorizzazione, ossia alla sequenza 6 7 3 3 3 2 2 2 2 0 0 0 ... 0 0, si ottiene una sequenza di coppie *(skip, ACN)*, pari a *(0,6), (0,7), (0,3), (0,3), (0,3), (0,2), (0,2), (0,2), (0,2) (0,0)* in cui l'ultima coppia *(0,0)* indica la fine del blocco, che in fase di decodifica viene quindi ricostruito riempiendolo di zeri. Anziché usare questa, proseguiamo adottando una diversa sequenza di coppie *(skip, ACN)*, pari a *(0,6), (0,7), (3,3), (0,-1),*

(0,0)[42]: sostituendo ai termini ACN di questa, la coppia *sss, valore*, e codificando quindi il termine *valore* come indicato nella prima tabella dell'esempio precedente, si ottiene *(0, 3, 110), (0, 3, 111), (3, 2, 11), (0, 1, 0), (0,0).* Il *bitstream* finale viene quindi realizzato sostituendo alle attuali coppie *skip, sss*, le rispettive codeword individuate alla colonna *Run/Size* della tabella a pagina 150 e segg. delle specifiche ITU-T T.81 `http://www.digicamsoft.com/itu/itu-t81-154.html`, ottenendo *(100, 110), (100, 111), (111110111, 11), (00, 0), (1010)*, e producendo così un totale di 30 bit per rappresentare i 63 coefficienti AC.

Formattazione Lo standard JPEG definisce, oltre alla sequenza di operazioni indicata, anche il formato di trama con il quale deve essere memorizzato il bitstream finale. La struttura risultante è gerarchica, e mostrata alla figura 10.29. Al livello superiore troviamo un *frame header* che contiene le dimensioni complessive dell'immagine, il numero ed il tipo di componenti usate (CLUT, RGB, YC_bC_r, etc), ed il formato di campionamento (4:2:2, 4:2:0, etc.). Al secondo livello, troviamo uno o più *Scan*, ognuno preceduto da una intestazione in cui viene riportata l'identità del componente (R, G, B, o Y, C_b, C_r), il numero di bit usato per rappresentare ogni coefficiente di DCT, e la tabella di quantizzazione usata per quella componente. Ogni Scan è composto da uno o più *segmenti*, preceduti da un'ulteriore intestazione, che contiene il codebook di Huffman usato per rappresentare i valori dei blocchi del segmento, nel caso non siano stati usati quelli standard. Infine, nel segmento trovano posto le sequenze di blocchi dell'immagine, così come risultano dopo lo stadio di codifica entropica.

10.3 Codifica video

In accordo al metodo di realizzazione dei segnali video analogici, in cui i singoli quadri sono codificati indipendentemente gli uni dagli altri, la codifica video digitale può essere

[42] La nuova sequenza di coppie corrisponde ad una sequenza di coefficienti AC pari a 6 7 0 0 0 3 −1 0 0 0

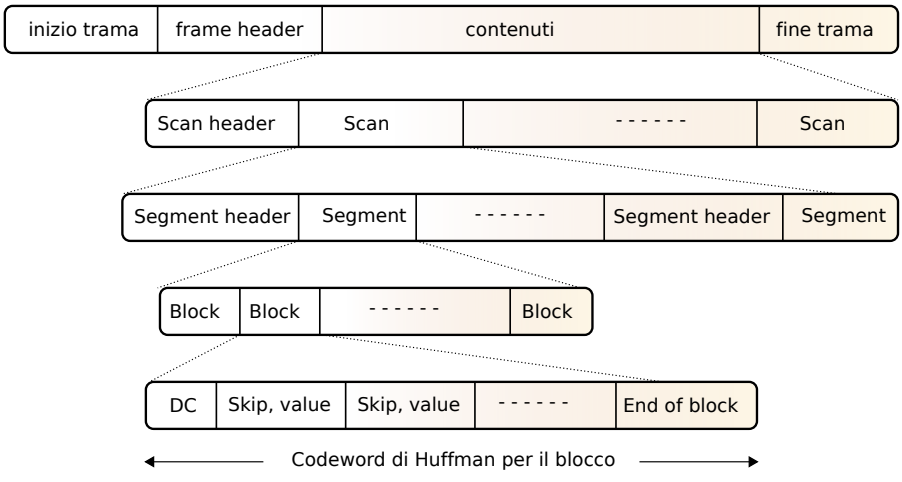


Figura 10.29: Formato del bitstream per la codifica JPEG

realizzata semplicemente applicando tecniche di codifica di immagine (come JPEG) ad ognuno dei quadri che costituiscono la sequenza video: questo tipo di approccio prende il nome di *moving* JPEG o MJPEG.

D'altra parte, i quadri relativi ad istanti temporali vicini sono spesso molto simili tra loro, anche se quanto siano simili, e per quanto tempo, dipende dal tipo di filmato. La presenza di *memoria* nella sorgente determina quindi la possibilità di ridurre il tasso informativo prodotto dalla codifica ricorrendo a tecniche predittive, tentando quindi di *stimare il movimento* presente in quadri contigui, e trasmettere solo l'informazione necessaria a *compensare* l'errore di predizione. Nella parte destra di fig. 10.30 è mostrata l'immagine differenza ΔY tra la componente di luminanza di due quadri consecutivi, consentendo di apprezzarne la relativa semplicità.

Figura 10.30: Due fotogrammi consecutivi, e la differenza tra i rispettivi valori di luminanza

Considerando poi che alcune regioni si sono mosse più di altre, il quadro da codificare è scomposto in sottoimmagini, per ognuna delle quali si ha un diverso spostamento, e viene calcolata una specifica differenza rispetto alla sotto-immagine precedente (e spostata); questa tecnica prende il nome di *compensazione del movimento*.

Tipo di quadro Come abbiamo fatto notare al § 9.2.2, le tecniche di codifica predittiva sono particolarmente sensibili agli errori di trasmissione, che possono causare una perdita di sincronismo tra i predittori di trasmissione e ricezione, e quindi l'impossibilità di ricostruire la restante parte di segnale. Per questo nella codifica video sono presenti dei quadri *di riferimento* in corrispondenza biunivoca con un unico quadro di partenza, detti *intracoded frames* o **I**-frames, che permettono al ricevitore di ri-partire da una condizione nota. Tra due quadri **I** sono poi presenti un certo numero di quadri **P** (*predicted*) come in fig. 10.31-a, oltre che quadri **B** (*bidirectional*) come in fig. 10.31-b, e che corrispondono rispettivamente alla codifica della compensazione del movimento calcolato a partire da un unico quadro precedente, o da una coppia di quadri passato e futuro.

I quadri **I** sono codificati mediante l'algoritmo JPEG, usando lo stesso coefficiente di quantizzazione per tutti i pixel delle DCT, conseguendo un rapporto di compressione relativamente basso, e sono inseriti a cadenza fissa con un periodo N tipicamente compreso tra 3 e 12: la sequenza di quadri compresi tra due quadri **I** è detta *group of pictures* o GOP. Come mostrato in figura 10.31, la codifica di quadri **P** può dipendere dal quadro **I** immediatamente precedente, o dalla ricostruzione di un precedente quadro **P**, ottenendo un fattore di compressione maggiore che per i quadri **I**; la distanza temporale tra **P** e l'originale **I** è detta *intervallo di predizione*, indicato con M.

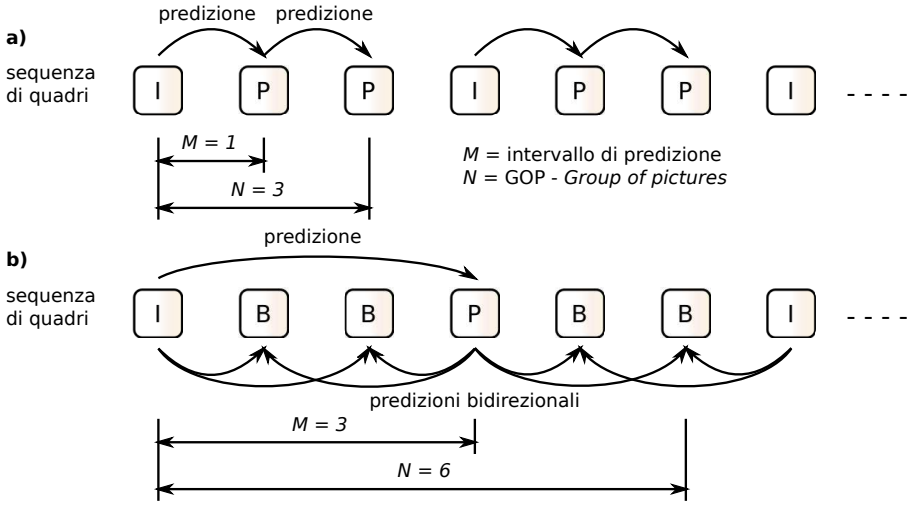


Figura 10.31: Esempi di sequenze di quadri con **a)** solo quadri di tipo I e P; **b)** quadri di tipo I, P e B

Per realizzare la compensazione del movimento, ogni regione del nuovo quadro è confrontata con regioni *limitrofe* del quadro precedente, riducendo così la complessità di ricerca. Nel caso dei quadri **B** la ricerca delle regioni simili è invece svolta rispetto ai quadri **I** (o **P)** situati sia nel passato che al futuro, migliorando la precisione della stima di movimento, e conseguendo rapporti di compressione ancora maggiori, a patto di subire un aumento del ritardo di codifica, legato al dover attendere un quadro futuro.

Allo scopo di ridurre il ritardo di decodifica la sequenza di quadri viene trasmessa con un ordine diverso da quello dei quadri originali, consentendo ai quadri **B** di essere riprodotti non appena ricevuti, e non dopo la ricezione del quadro *futuro* da cui dipendono. Pertanto, se la sequenza originale è ad esempio

IBBPBBPBBIBBP...

questa verrà trasmessa nell'ordine

IPBBPBBIBBPBB...

Stima di movimento e compensazione Specifichiamo innanzitutto cosa intendere con il termine *regione* prima usato per definire il dominio dell'operazione di confronto necessaria alla stima di movimento. Come mostrato in fig. 10.32a, considerando una suddivisione in componenti $Y\ C_b\ C_r$ ed un sottocampionamento 4:1:1, il quadro originale è suddiviso in N righe e M colonne di *macroblocchi* di 16x16 pixel, ed ogni macroblocco è rappresentato da sei blocchi 8x8 pixel, di cui quattro blocchi per la luminanza, più due blocchi per le componenti di crominanza; ogni blocco 8x8 corrisponde quindi ad un equivalente numero di coefficienti DCT, ed è individuato all'interno del quadro, in base al suo indirizzo di riga e colonna.

Nella codifica dei quadri **P**, ogni macroblocco M_T del quadro corrente (*target*) è confrontato pixel per pixel con il corrispondente macroblocco M_R del quadro *di*

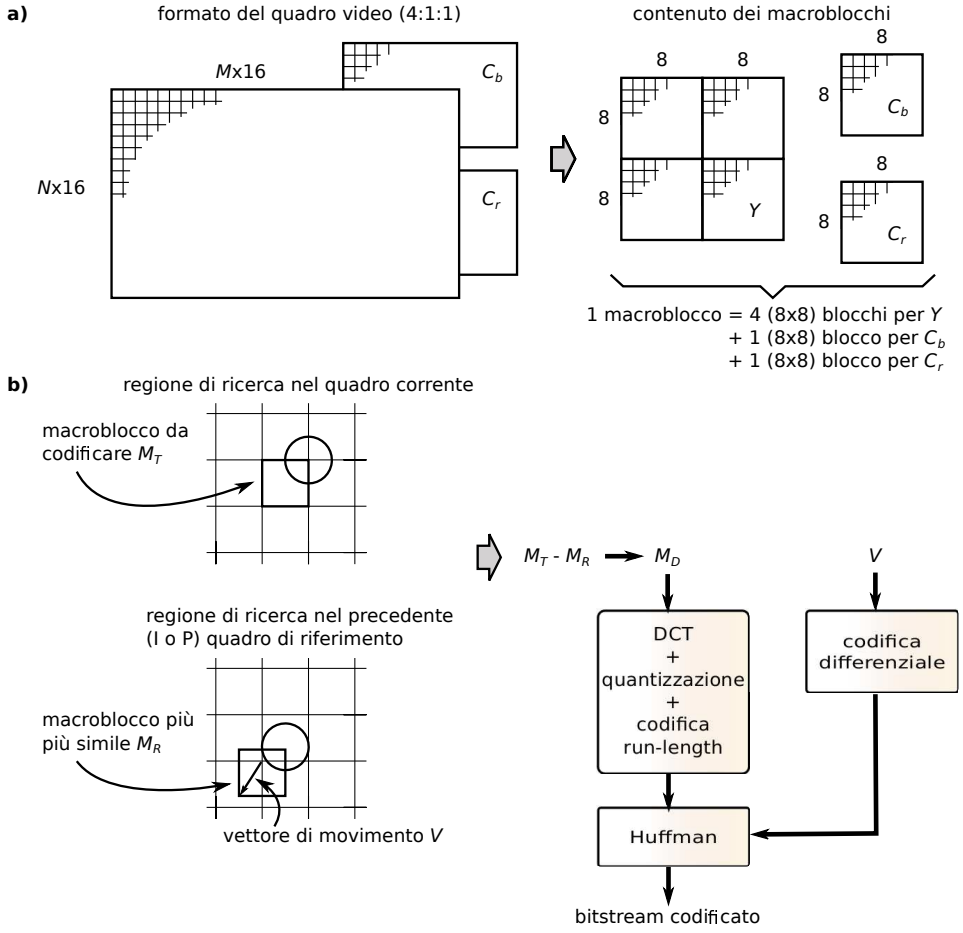


Figura 10.32: Codifica di un quadro P: a) struttura del macroblocco; b) procedura di codifica

riferimento, e nel caso sia riscontrata una sufficiente similitudine[43] complessiva, viene trasmesso solo l'indirizzo del blocco. Altrimenti, il confronto viene ripetuto per tutti i possibili spostamenti del macroblocco target nell'ambito dei macroblocchi contigui[44], e qualora sia individuata una buona corrispondenza, il macroblocco viene codificato dal *vettore di movimento* V e dall'*errore di predizione* M_D. Con riferimento alla fig. 10.32-b in cui l'immagine è simboleggiata da un cerchio, V rappresenta lo spostamento da applicare a M_T per portarlo a coincidere al meglio con il quadro precedente, ed è codificato come una coppia (x, y) corrispondente ad una *risoluzione di un pixel*. Al contrario M_D è composto dalle tre matrici $(Y\ C_b\ C_r)$ dei valori differenza tra quelli di M_T spostato di V, ed M_R. I valori di V e di M_D relativi ai diversi macroblocchi di un quadro seguono poi due diversi percorsi di codifica, come specificato appresso.

[43] Il confronto è svolto considerando i soli valori di luminanza, e la similitudine valutata come media tra i valori assoluti delle differenze di luminanza.

[44] l'effettiva estensione dell'area di ricerca non è oggetto di standardizzazione, mentre lo è la rappresentazione del risultato della ricerca.

Nel caso in cui la regione di ricerca sia estesa, i valori V possono risultare relativamente grandi; d'altra parte è probabile che macroblocchi vicini esibiscano vettori di spostamento molto simili tra loro. Per questi motivi, la sequenza dei V calcolati per macroblocchi contigui viene prima sottoposta ad un processo di codifica differenziale, e quindi i valori di differenza sono rappresentati da codeword a lunghezza variabile di Huffman. D'altra parte, le tre matrici differenza sono invece sottoposte alla stessa sequenza di operazioni dei quadri **I** (DCT, quantizzazione, codifica entropica), conseguendo però un fattore di compressione più elevato, essendo il macroblocco differenza con valori quasi tutti molto piccoli.

Nel caso in cui la stima di movimento fallisca[45] (o a causa di una estensione di ricerca insufficiente, oppure per un reale cambio di scena), il macroblocco è codificato in modo indipendente come avviene per i quadri **I**.

I macroblocchi dei quadri B (vedi fig. 10.33) sono invece confrontati sia con il *precedente* quadro M_P che con il *successivo* M_S, ottenendo due possibili insiemi di

[45] Viene decretato il fallimento quando anche la migliore compensazione di movimento possibile non determina una riduzione della quantità di bit, rispetto ad una codifica JPEG.

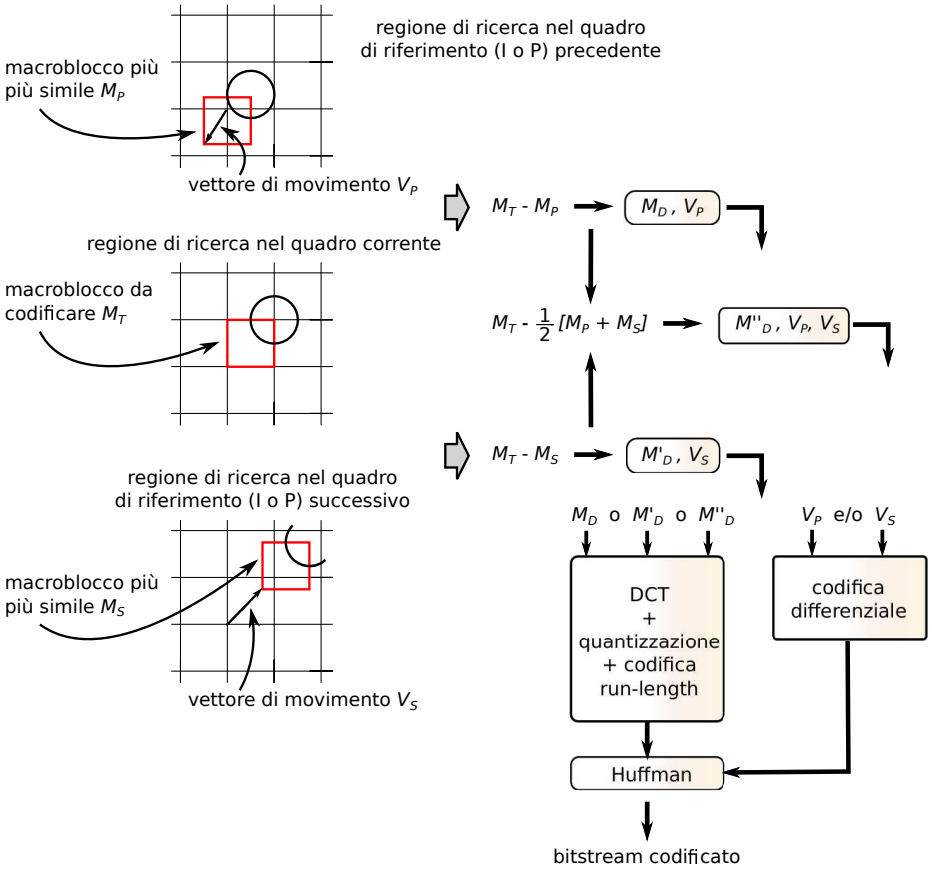


Figura 10.33: Procedura di codifica dei quadri B

matrici differenza M_D e $M_D^{'}$ ed associati vettori V_P e V_S; viene inoltre calcolato un ulteriore insieme $M_D^{''}$ come differenza tra M_R e la media dei macroblocchi (spostati) di riferimento, e determinato infine quale delle tre possibilità fornisca il minimo errore di predizione. In base a questa scelta, si individua quale macroblocco differenza codificare, assieme ai rispettivi vettori di movimento. Nel caso prevalga la predizione basata sulla media tra macroblocchi di riferimento, il vettore di movimento complessivo può determinare un potere di risoluzione a livello di *sub-pixel*.

Questioni realizzative La fig. 10.34 riassume la sequenza di operazioni applicate alle tre tipologie di quadro **I**, **P** e **B**. Mentre nel primo caso queste seguono lo schema previsto dalla codifica JPEG, i quadri **P** meritano qualche commento: allo scopo di alimentare correttamente il componente di stima di movimento, il codificatore mantiene memoria del quadro di riferimento, all'inizio posto pari ad un quadro **I**, e quindi sostituito da una copia dell'ultimo quadro **P**, ottenuto risommando il quadro differenza al precedente quadro di riferimento. Lo stesso schema di calcolo è svolto nel caso di quadri **B**, tenendo ora conto anche del quadro successivo.

Rimarchiamo ora il fatto che, in funzione dell'esito del processo di stima di movimento, esistono tre diverse possibilità di rappresentazione per ogni macroblocco dei

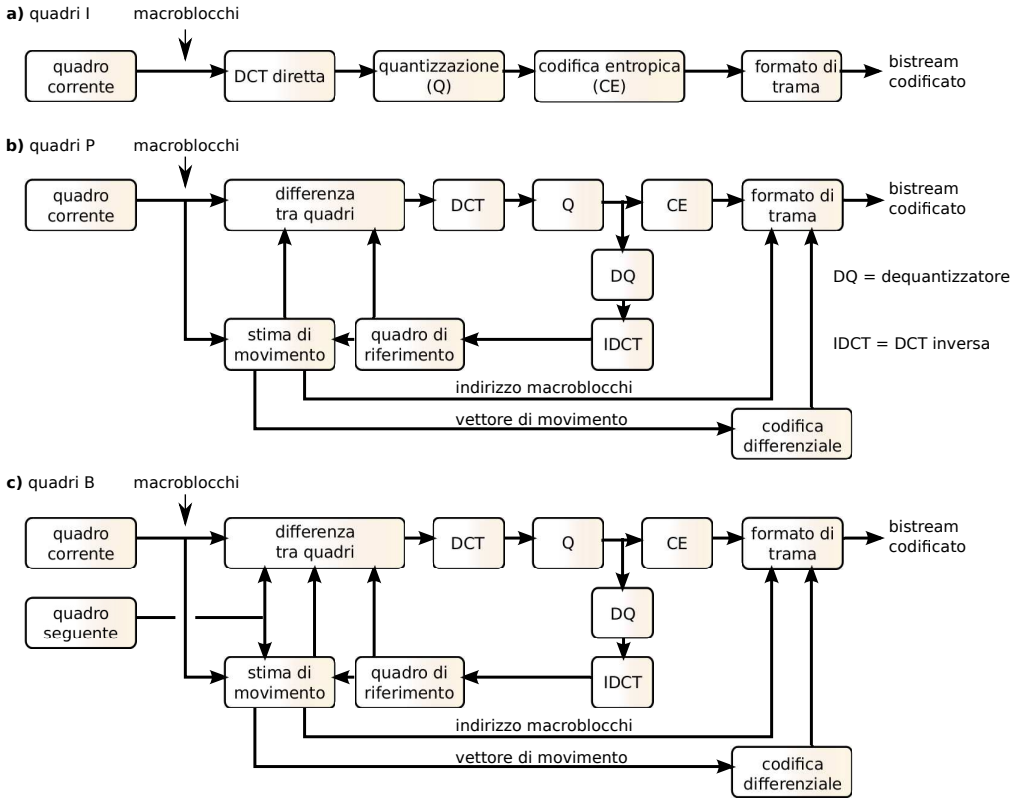


Figura 10.34: Stadi di elaborazione nella codifica di: a) quadri I; b) quadri P; c) quadri B

quadri **P** e **B**:

- se non vi è movimento, viene trasmessa solo la sua posizione;
- se vi è movimento e si trova un riferimento abbastanza simile, sono trasmessi il vettore di movimento e le matrici differenza;
- se non si è trovato un riferimento abbastanza simile, viene effettuata una codifica *inter* come per il caso dei quadri **I**.

Ciò determina l'esigenza di disporre di un formato di trama di dimensione (e velocità) variabile, come realizzato nell'esempio mostrato in fig. 10.35, in cui ad ogni macroblocco è associato un tipo (**I**, **P** o **B**), il suo indirizzo nell'ambito del quadro, il coefficiente di quantizzazione relativo ai termini della DCT, ed il vettore di movimento (se presente). Quindi si dichiara l'identità dei blocchi presenti (che potrebbero essere assenti in caso di immagini statiche), e per questi viene infine prodotta la sequenza di informazioni previste dalla codifica JPEG.

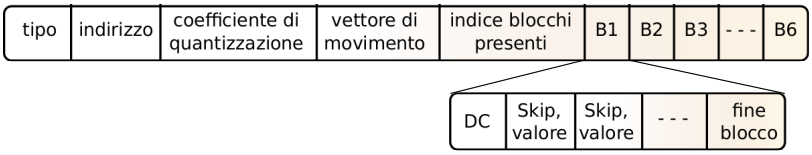


Figura 10.35: Esempio di formato trama per i macroblocchi di un bitstream video

10.3.1 Standard video

Come per l'audio, anche il numero dei *codec video* è elevato, ed è istruttivo prenderli in esame secondo l'ordine cronologico con cui si sono sviluppati, dato che in pratica ognuno prende il precedente come base di partenza.

10.3.1.1 H.261

E' lo standard di codifica video definito da ITU-T a fine anni '80 per le applicazioni di videotelefonia su ISDN, ed anche se oggi tecnicamente superato, resta comunque un valido sistema di riferimento che consente la retro-compatibilità tra apparati[46]. Il suo principale limite è il vincolo di dover produrre una velocità ridotta e comunque *quantizzata* a multipli di 64 kbps.

La scelta del formato di immagine è limitata a quanto mostrato in tabella[47], mentre la scansione è non interallacciata e la velocità di rinfresco di 30 quadri/secondo per CIF oppure 15 o 7.5 per QCIF. Sono usati solo quadri di tipo **I** e **P**, con un GOP di 4 (ossia 3 **P**

Formato	Y	C_b, C_r
CIF	352 x 288	176 x 144
QCIF	176 x 144	88 x 72

[46] Vedi ad es. `http://www0.cs.ucl.ac.uk/teaching/GZ05/08-h261.pdf` (una presentazione di *Mark Handley*), o la trattazione su `http://en.wikipedia.org/wiki/H.261`.

[47] Il *Common Intermediate Format* (CIF) è stato pensato per facilitare la compatibilità con PAL e NTSC; il *Quarter*-CIF ha una superficie di 1/4. Sono poi stati anche definiti il 4CIF e 16CIF, oltre che il SIF (352 x 240) che interopera con flussi MPEG.

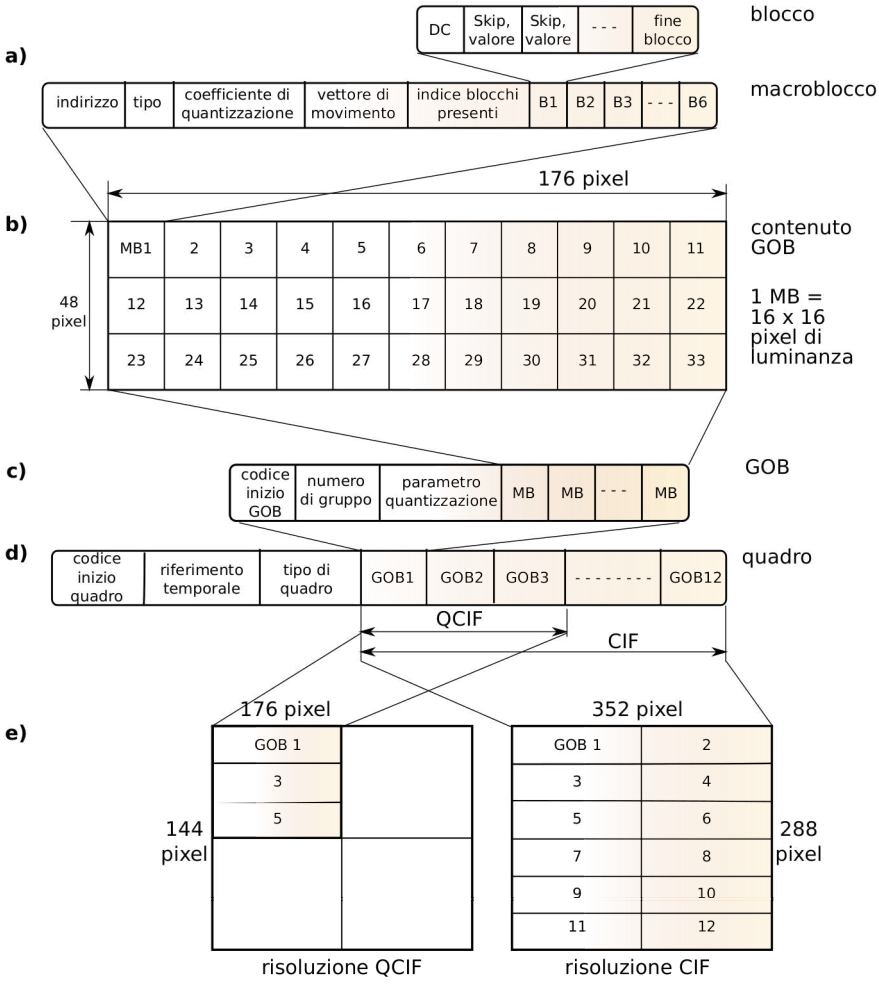


Figura 10.36: Formato codifica H.261: a) macroblocco; b) costruzione e c) formato di un GOB; d) trama di quadro; e) interoperabilità tra formati

ogni **I**), e sono usate le procedure descritte alla sezione precedente per rappresentare ogni quadro nei termini di macroblocchi composti da 16x16 pixel (4 blocchi di 8x8) di luminanza e 2 blocchi 8x8 per ogni componente di colore C_b, C_r.

Ogni macroblocco segue la tipica formattazione mostrata in fig. 10.36-a; tre file di 11 macroblocchi sono poi raggruppati in una nuova struttura sintattica detta GOB (*Group of (macro)Blocks*), che si articola in un contenuto (fig. 10.36-b) ed una intestazione (fig. 10.36-c), in cui troviamo un *codice di inizio* scelto in modo da non poter essere presente nella sequenza di codici di Huffman che seguono, e che permette la risincronizzazione nel caso di GOB *mancanti* (vedi appresso), in modo da poter tornare a riprodurre un quadro in corrispondenza del primo GOB disponibile. L'intero quadro è quindi realizzato con il formato di fig. 10.36-d), in cui compare un *codice di inizio quadro*, un *riferimento temporale* necessario alla sincronizzazione con la traccia audio,

e l'indicazione del tipo di quadro (**I** o **P**); a cui segue la sequenza dei GOB, in numero di 3 oppure 12 a seconda se il quadro rappresenti una immagine QCIF o CIF, in modo da permettere l'interoperabilità tra formati come mostrato in fig. 10.36-e.

Controllo di velocità Dato che la codifica video produce una velocità di trasmissione variabile, questa può eccedere la capacità del canale a disposizione, ed un modo *drastico* per risolvere il problema è di scartare alcuni GOB. Il campo *Group number* dell'intestazione dei GOB permette quindi di collocare il nuovo GOB anche in mancanza dei suoi predecessori.

Un approccio più articolato è quello mostrato dalla figura 10.37-a, che ripercorre le tappe già discusse e relative al calcolo del vettore di movimento ed alla codifica degli errori di predizione, ma pone in evidenza il campo di intervento di un componente di *controllo quantizzazione*, che variando l'entità dei coefficienti di quantizzazione della DCT, permette di ridurre e/o aumentare la velocità di codifica complessiva. In particolare, il controllo di quantizzazione opera in base allo stato di riempimento del *buffer* FIFO[48] mostrato in fig. 10.37-b, alimentato dal risultato del processo di codifica e formattazione video, e da cui sono prelevati i dati da inviare a velocità costante. Nel caso in cui la velocità media di codifica ecceda quella disponibile, l'aumento della occupazione del buffer determina l'aumento del coefficiente di quantizzazione, e quindi una riduzione della qualità ma anche della velocità media di codifica; ovviamente, anche l'inverso è possibile, ossia un miglioramento di qualità mediante riduzione del coefficiente di quantizzazione, nel caso in cui la scena sia statica, e la codifica produca un basso bit rate che consente alla FIFO di svuotarsi.

[48] *First in First out,* è la disciplina di coda del primo arrivato primo servito, opposta a LIFO *Last In First Out*, realizzata come uno *stack*.

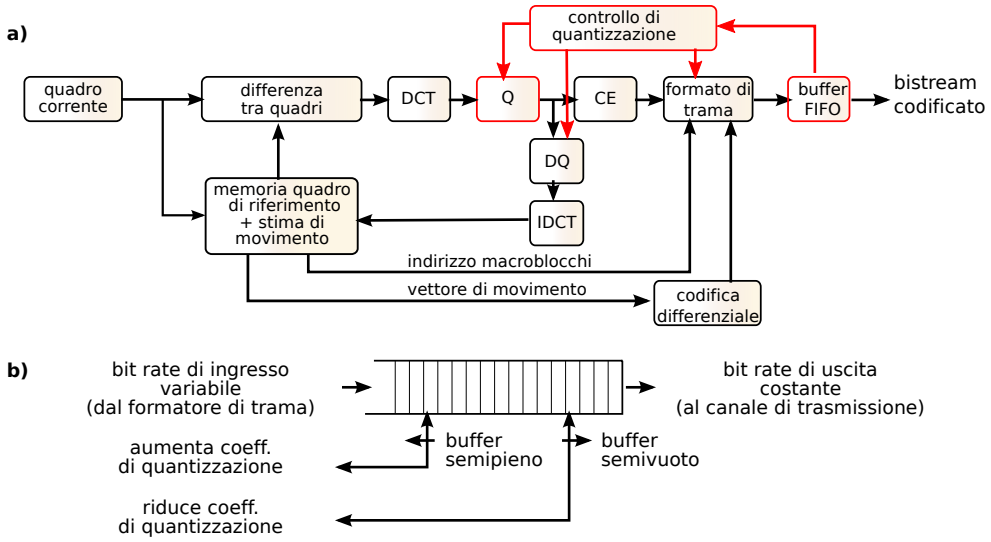


Figura 10.37: Principi della codifica H.261: a) schema del codificatore b) funzionamento del buffer FIFO

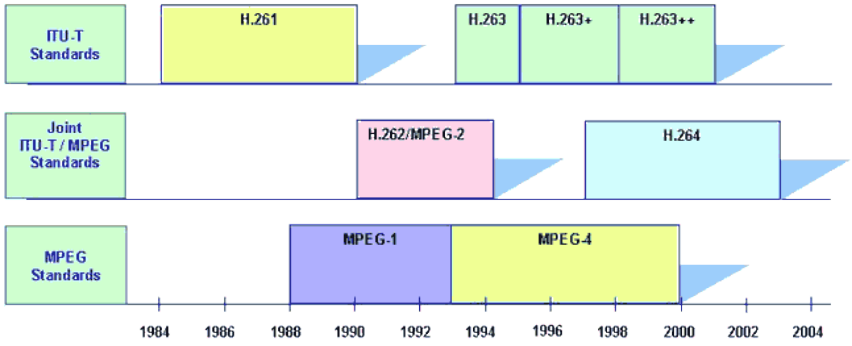


Figura 10.38: Evoluzione temporale dei formati di codifica video

Nel caso di un aumento improvviso di velocità, come anticipato si possono addirittura *scartare* alcuni GOB, mentre per i successivi si adottano coefficienti di quantizzazione ridotti, comunicati anche al lato ricevente per mezzo dell'apposito campo della intestazione GOB, come mostrato in fig. 10.36-c.

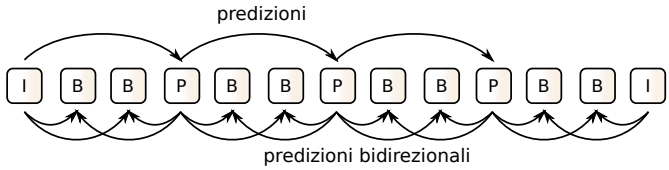
10.3.1.2 H.263

Anche questo definito da ITU-T a partire dal 1995, nasce per risolvere i problemi di bassa qualità dell'H.261 a velocità molto ridotte, come quelle offerte dai collegamenti modem *dial-up* precedenti all'introduzione dell'ADSL, ovvero per migliorare la gestione delle possibili condizioni di errore sia sul canale *dial-up* che *wireless*. Le specifiche originarie si sono in seguito arricchite[49] di estensioni, favorendo l'adozione del codec da parte di altre applicazioni (inclusi i filmati di *youtube*), ed aggiungendo il supporto oltre che ai formati nativi CIF e QCIF, anche a S-QCIF, 4CIF, 16CIF, SIF e 4SIF. A partire dal 2003 si è formato un gruppo di lavoro congiunto tra ITU-T VCEG (*Video Coding Expert Group*) e ISO/IEC MPEG (*Moving Pictures Experts Group*), che segue la definizione del suo successore, l'H.264 detto anche AVC (Advanced Video Coding) o MPEG-4 *part 10*, determinando l'arresto dello sviluppo di H.263, che resta comunque (assieme all'H.261) supportato da un gran numero di applicazioni multimediali. Sebbene la struttura generale del codificatore e del bitstream ricalchi quella vista per l'H.261, sono state introdotte alcune novità significative, che tentiamo di elencare appresso.

Tipi di quadro In H.263 sono usati, oltre ai quadri di tipo **I** e **P**, anche quelli bidirezionali **B**, consentendo di ottenere fattori di compressione maggiori, a parità di qualità percepita.

Slice I GOB sono ridefiniti come singole *strisce* di macroblocchi, quindi ad esempio per i formati CIF e QCIF un GOB è ora formato da 11 macroblocchi in fila, anziché 33 come avveniva per l'H.261.

[49] Nel 1998 viene rilasciato l'H.263v2, noto anche come H.263+ o H.263 1998, e nel 2000 è emesso l'H.263v3 noto anche come H.263++ o H.263 2000; inoltre l'MPEG-4 Part 2 è compatibile con l'H.263, in quanto un bitstream H.263 di base viene correttamente riprodotto da un decodificatore MPEG-4.



Esempio di sequenza di quadri MPEG-1

Vettori di movimento estesi La stima di movimento dell'H.261 si arresta in corrispondenza dei bordi del quadro, per cui anche se un oggetto è solo parzialmente uscito di scena, il macroblocco corrispondente viene codificato in modalità *intra*. Al contrario H.263 permette di estendere la ricerca anche a vettori di spostamento che cadono al di fuori del quadro, alla ricerca di una corrispondenza parziale, consentendo al contempo maggiore efficienza e minor distorsione.

Predizione avanzata Anziché determinare il vettore di movimento in base al confronto di un intero macroblocco, i 4 blocchi 8x8 che lo costituiscono sono confrontati in modo indipendente con il quadro di riferimento, permettendo una migliore compensazione del movimento anche per l'immagine di oggetti che non solo traslano, ma si deformano. In definitiva, sono prodotti 4 diversi vettori di movimento per ogni macroblocco.

Resistenza agli errori La presenza di un errore nella ricezione[50] di un GOB, oltre ad impedire la corretta riproduzione dello stesso, ostacola la riproduzione anche dei quadri successivi che dipendono dai pixel presenti nel GOB, e peggio ancora l'errore finisce per estendersi anche ad altri GOB, in virtù degli effetti dell'errore sulla ricostruzione dei macroblocchi *predetti* in presenza di movimento.

Per ridurre l'estensione temporale dell'effetto dell'errore, e non dover attendere fino alla ricezione del successivo quadro **I**, si può usare il canale di ritorno presente nei collegamenti punto-punto, consentendo al decodificatore di inviare dei NACK che notificano al mittente la coppia (quadro, GOB) per la quale si è rilevato un errore. Il codificatore è quindi in grado di valutare esso stesso le conseguenze sui quadri successivi, e può provvedere a fornire una codifica *intra* per tutti i blocchi che necessitano di essere rapidamente risincronizzati.

10.3.1.3 MPEG-1

Il *Moving Pictures Expert Group* di ISO emette una serie di standard ognuno orientato ad un particolare dominio applicativo di segnali multimediali, come

[50] Qualcuno potrebbe aver notato che nella definizione degli standard fin qui discussi, non sono previsti controlli di tipo *checksum* nel bitstream prodotto. D'altra parte essendo le informazioni codificate di natura auto-sincronizzante, la presenza di errori determina presto presso il ricevitore una condizione di disallineamento, e la decodifica di valori non previsti, come ad esempio la ricezione di vettori di movimento o coefficienti DCT fuori dinamica, o codeword di Huffman non valide, od un numero eccessivo di coefficienti. Per tale via, il ricevitore diviene in grado di accorgersi che si è verificato un errore.

- MPEG-1 adotta un formato SIF di 352x288 pixel inteso per la *memorizzazione* audio-video a qualità VHS su CDROM, a velocità fino a 1.5 Mbps;
- MPEG-2 è orientato alla *memorizzazione e trasmissione* audio-video secondo quattro livelli di risoluzione, per ognuno dei quali diversi profili individuano tecniche alternative di codifica;
- MPEG-4 è stato inizialmente concepito per applicazioni simili a quelle dell'H.263, ma il suo uso si è successivamente esteso ad un'ampia gamma di applicazioni Internet.

Anche MPEG-1 adotta tecniche del tutto simili a quelle dell'H.261, con una scansione dell'immagine progressiva ed un sottocampionamento delle componenti di colore 4:1:1, una frequenza di quadro di 25 Hz, l'adozione di quadri di tipo **I**, **P** e **B**, la rappresentazione dei quadri in termini di macroblocchi composti da 16x16 pixel di luminanza, più due blocchi 8x8 per ciascuna componente di colore. Le principali differenze sono che

- possono essere inseriti riferimenti temporali *all'interno* di un quadro, permettendo al decodificatore di sincronizzarsi più rapidamente. L'intervallo tra due marche temporali è chiamato *slice* e comprende una sequenza orizzontale di macroblocchi, tipicamente che copre una intera riga[51], o meno, ma non di più;
- l'uso dei quadri di tipo B aumenta la distanza temporale tra i quadri di tipo P ed il loro riferimento, e quindi determina una maggiore distanza coperta dalle porzioni di immagine in movimento, cosicché l'ampiezza della finestra di ricerca adottata dal componente di detezione di movimento è stata estesa.

La figura 10.39-a) illustra la *struttura gerarchica* del bitstream risultante, secondo il quale l'intero filmato (*sequenza*) è costituito da una successione di GOP, ed ogni GOP da una sequenza di quadri, ognuno costituito da una successione di *slice* che comprendono ognuno 22 macroblocchi, ognuno con 6 blocchi. La sezione -b) di fig. 10.39 entra più nel dettaglio del formato del bitstream.

10.3.1.4 MPEG-2

Allo scopo di poter usare questo stesso standard per diversi contesti applicativi, sono stati definiti i quattro *livelli* qualitativi mostrati alla tabella seguente, e per ogni livello sono quindi definiti cinque profili (*simple, main, spatial resolution, quantization accuracy, high*) in modo da permettere lo sviluppo di nuove tecnologie. Il livello *Low* è compatibile con MPEG-1. Affrontiamo ora la descrizione di ciò che è offerto dal profilo *Main* al livello *Main* (MP@ML).

Livello MPEG-2	formato	bit rate (Mbps)	applicazione
Low	SIF	< 1.5	registrazione qualità VHS
Main	4:2:0	< 15	DVB - MP@ML
	4:2:2	< 20	
High 1440	4:2:0	< 60	HDTV 4/3
	4:2:2	< 80	
High	4:2:0	< 80	HDTV 16/9
	4:2:2	< 100	

[51] ad es., 22 macroblocchi in risoluzione CIF

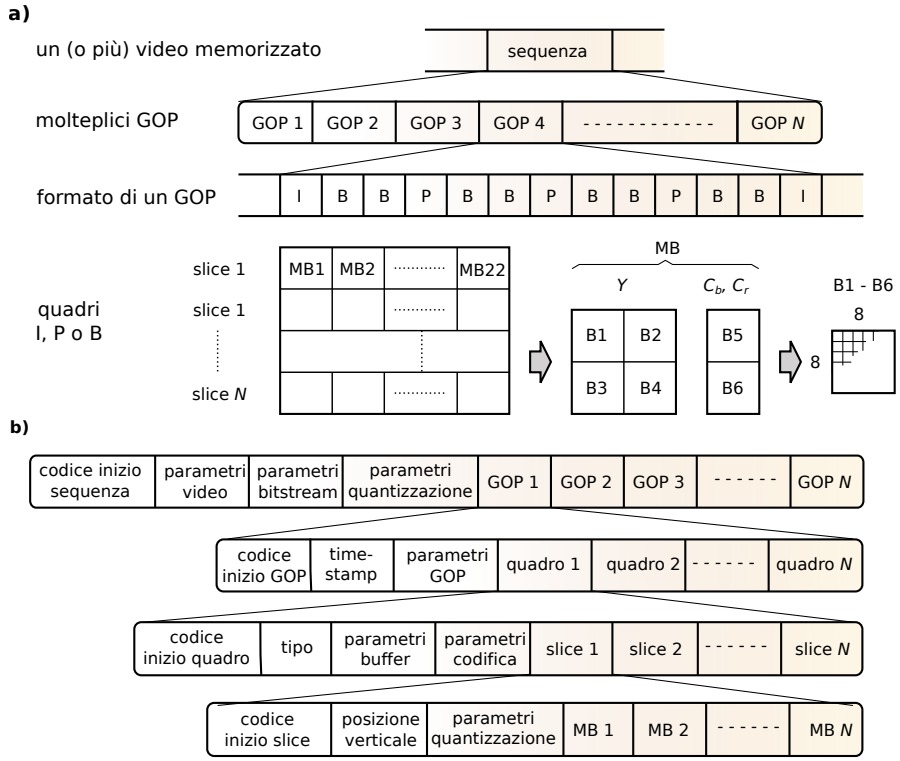


Figura 10.39: Struttura del bitstream MPEG-1: a) composizione; b) formato gerarchico

MP@ML L'obiettivo è la diffusione televisiva DVB, con scansione interallacciata, risoluzione 720x576 a 25 quadri/secondo (PAL), sottocampionamento 4:2:0 per una velocità risultante tra i 4 ed i 15 Mbps. La principale differenza rispetto all'MPEG-1 è legata alla modalità di scansione *interallacciata*, in modo che come mostrato in fig. 10.40, ogni quadro è costituito da due sottoquadri (o *campi*) con le righe rispettivamente dispari e pari, ponendo la questione: come comporre i blocchi da 8x8 pixel su cui eseguire la DCT? Sono possibili due alternative:

- la *modalità campo* (fig. 10.41-a) in cui i 16x16 pixel di un macroblocco sono ripartiti tenendo assieme prima le sole righe dispari del primo campo, e quindi le sole righe pari del secondo campo, oppure
- la *modalità quadro* (fig. 10.41-b) in cui si usa la stessa suddivisione già vista per il caso non interallacciato, mescolando i due campi in ognuno dei blocchi.

La scelta migliore su quale tra le due modalità adottare dipende dal tipo di scena che si sta rappresentando. Se è presente molto movimento, è meglio adottare la modalità campo: essendo infatti i pixel di uno stesso blocco collezionati in un tempo pari a metà dell'intervallo di quadro (mentre nella seconda metà si collezionano i pixel della seconda serie di blocchi), si ottiene un *fotogramma meno mosso*; viceversa in presenza di una scena con poco movimento, può essere adottata la modalità quadro.

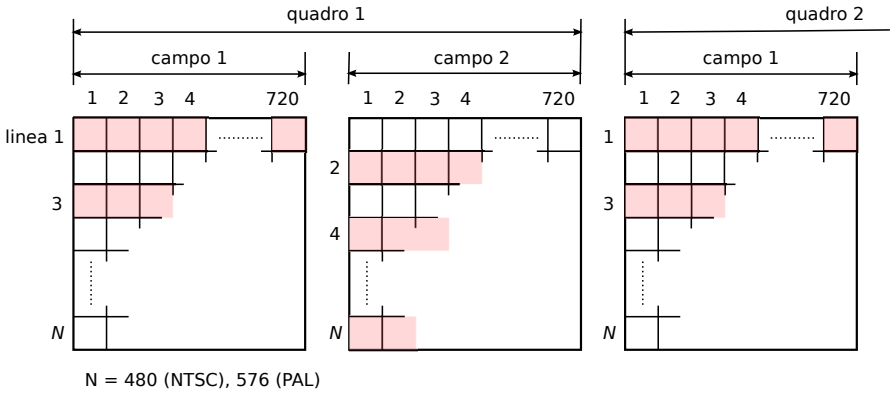


Figura 10.40: Effetto della scansione interallacciata in MPEG-2

Per quanto riguarda la stima di movimento, sono ora previste tre possibilità: la *modalità campo* prevede che i campi dispari usino come riferimento i campi pari del quadro precedente, ed i campi pari quelli dispari dello stesso quadro: in tal modo l'intervallo temporale su cui è valutato il movimento è metà del periodo di quadro. Nella *modalità quadro* invece, i campi pari e dispari usano come riferimento i rispettivi campi pari e dispari del quadro precedente, ed è più idoneo nel caso di movimenti lenti. Il meglio di entrambi i modi si ottiene con la *modalità mista*, in cui sono attuati entrambi gli approcci, e viene scelto per la trasmissione quello in grado di dar luogo alla distorsione minore.

HDTV Sono definiti tre standard, ATV, DVB e MUSE, rispettivamente per il Nord America, l'Europa ed il Giappone, a cui si aggiunge la specifica HDTV di ITU-R relativa a studi di produzione e scambio internazionale, e che definisce un rapporto di aspetto 16/9 con 1920 colonne per 1152 righe (di cui solo 1080 visibili), con scansione interallacciata e sottocampionamento 4:2:2. ATV include le specifiche di ITU-R, oltre che un formato ridotto, sempre con aspetto 16/9 ma risoluzione 1280x720, e che adotta la codifica video MPEG-2 MP@ML e quella audio AC-3. Il DVB è basato su di un rapporto di aspetto 4/3 con 1440x1152 pixel (di cui 1080 visibili), pari cioè al doppio[52] della risoluzione PAL di 720x576. La codifica video è MPEG-2 SSP@H1440 (*Spatially Scaleable Profile at High 1440*), simile all'MP@ML, mentre la codifica audio è MPEG audio layer 2.

10.3.2 Contenitori

I flussi binari prodotti dai diversi codificatori (audio, video) prendono il nome di *Elementary Stream* (ES), e possono essere multiplati assieme per produrre un nuovo formato idoneo alla registrazione di un contenuto multimediale completo, eventualmente arricchito da un flusso di *dati privati*, come mostrato in fig. 10.42a per il caso di MPEG-1.

Prima di effettuare la multiplazione, gli ES sono suddivisi in *pacchetti* di dimensione

[52] Si intende una risoluzione verticale ed orizzontale, mentre il rapporto tra le superfici è pari a 4 volte.

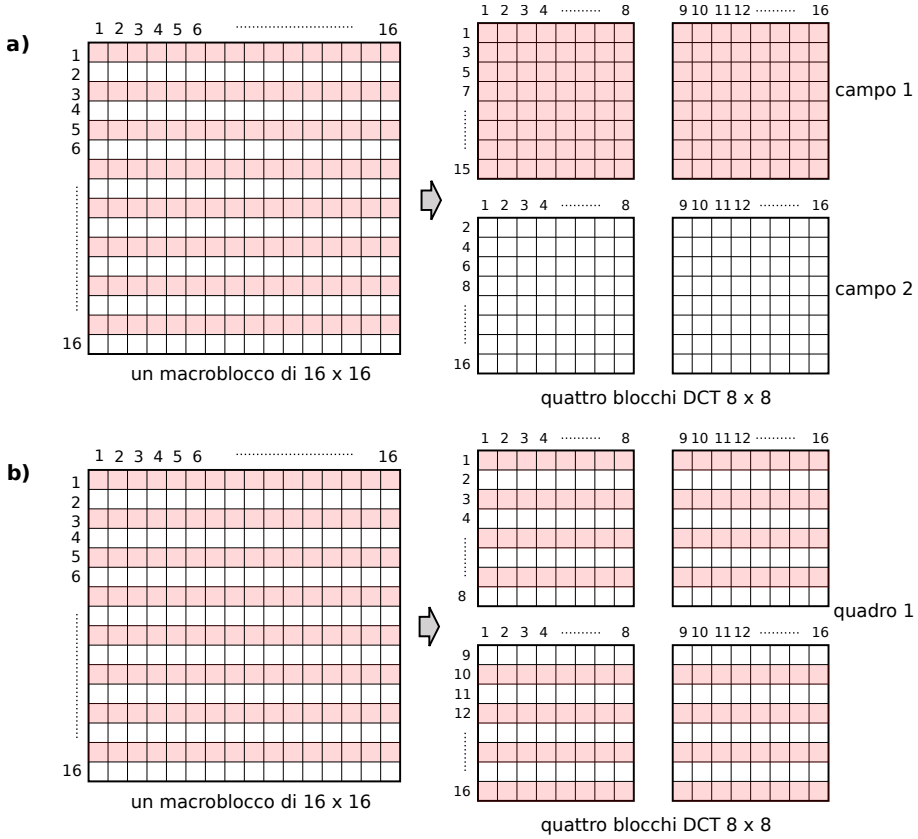
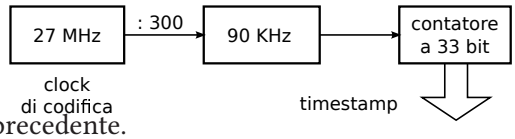


Figura 10.41: Composizione dei blocchi DCT per i quadri I di MPEG-2: a) modalità campo; b) modalità quadro

variabile denominati *Packetized ES* (PES) in cui trova posto un *payload* contenente[53] il risultato della codifica (ad es. un intero quadro), preceduto da una *intestazione* contenente un codice univoco di inizio PES ed un codice che individua il tipo di payload del pacchetto (audio, video o dati). L'intestazione PES può inoltre contenere un riferimento temporale necessario alla sincronizzazione audio-video, con risoluzione 33 bit, prodotto dal clock a 90 kHz descritto in fig. 10.42 come *orologio di sistema*, e ottenuto a partire da un oscillatore a 27 MHz come mostrato alla figura a pagina precedente.



Nel momento in cui un ES è pacchettizzato, viene inserito un *presentation timestamp* (PTS) che ne individua l'istante di riproduzione; per i flussi video è inserito anche[54] un *decode timestamp* (DTS) perché come anticipato a pag. 317 l'ordine di trasmissione (e quindi di decodifica) può differire dall'ordinamento naturale.

[53] Per un approfondimento, vedi `http://en.wikipedia.org/wiki/Packetized_elementary_stream`

[54] In realtà PTS e DTS non sono inseriti in tutti i pacchetti, ma una volta ogni tanto (con intervalli fino a 700 msec per i PS e 100 msec per i TS): il decoder rigenera infatti localmente il clock, ed i timestamp ricevuti servono a mantenerlo al passo con quello trasmesso.

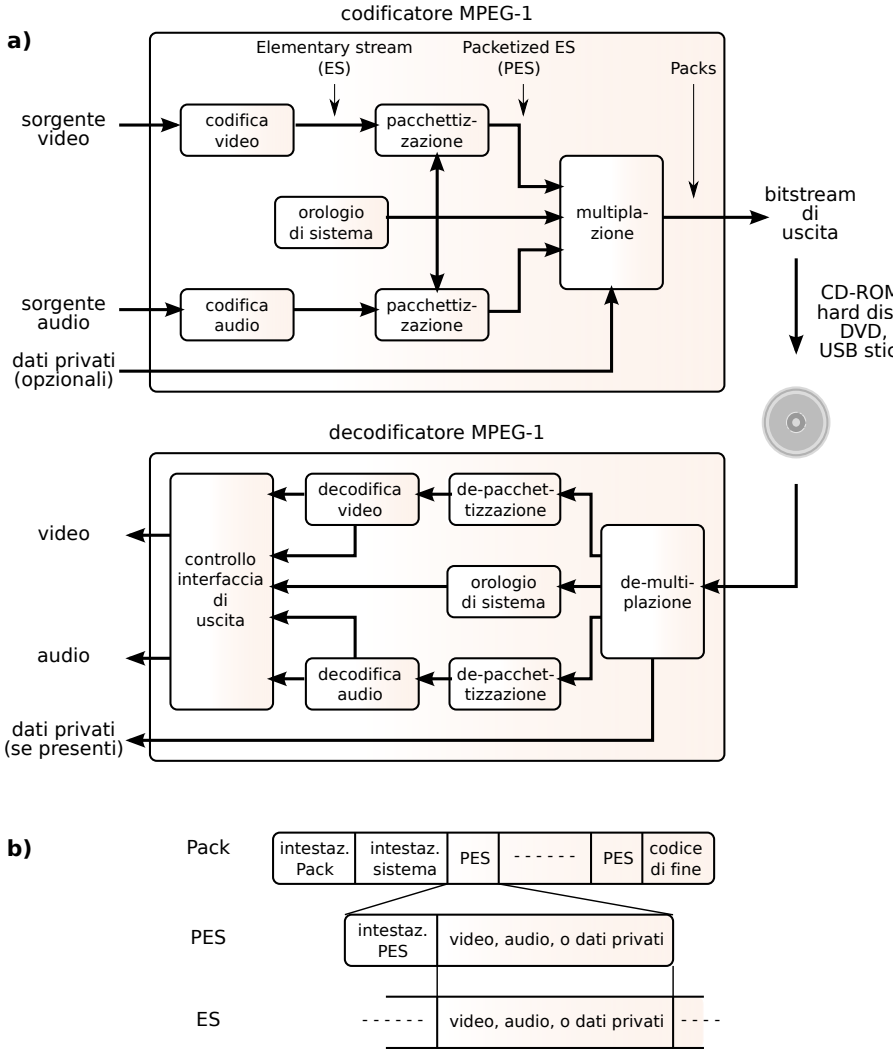


Figura 10.42: Generazione di un contenuto multimediale MPEG-1: a) co-decodificatore; b) formato del bistream di uscita

I PES derivanti da diversi ES possono essere quindi inseriti in una struttura di trama detta *Pack*, che può infine essere memorizzata ai fini di una successiva riproduzione.

10.3.2.1 Transport Stream

Nel caso di trasmissione dei contenuti mediante un sistema broadcast, tipicamente il singolo *Program Stream* PS (equivalente a quello prima indicato come *Pack*) viene ulteriormente multiplato assieme ad altri, in modo da realizzare un *Transport Stream* (TS), come mostrato in fig. 10.43. In particolare, la parte b) della figura mostra come i PES siano ora suddivisi in segmenti di lunghezza fissa e pari a 184 byte, intestati

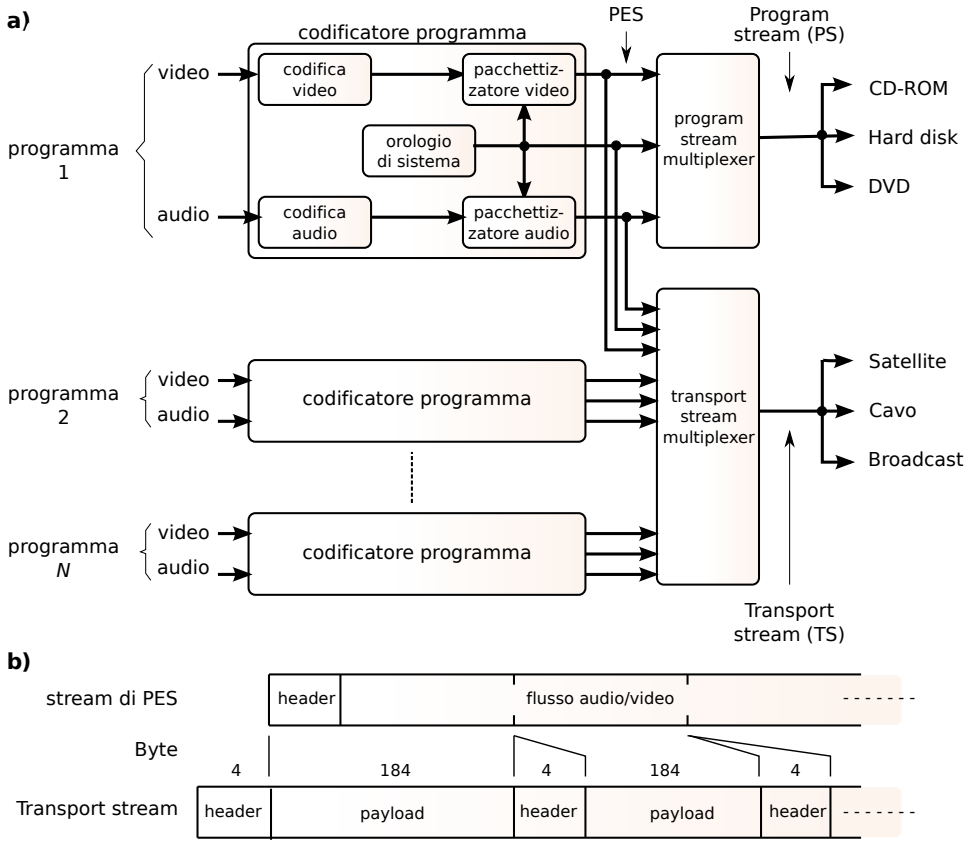


Figura 10.43: Multiplazione di programmi: a) generazione dei PS e TS; b) formato del Transport Stream

con 4 byte, producendo una struttura di trama con pacchetti di 188 byte[55]; l'ultimo pacchetto del TS che origina da uno stesso PES è riempito con uni fino a raggiungere i 188 byte. L'intestazione contiene, oltre ad un byte di inizio con pattern unico, un *Packet IDentification code* (PID) di 13 bit, che identifica il PES a cui appartiene il pacchetto, permettendo al decoder di recuperare il programma a cui è interessato.

Alcuni PID sono riservati, come il PID 8191 che indica assenza di payload[56], ed il PID 0, che annuncia l'inserimento nel payload della *Program Association Table* (PAT), la cui ricezione permette al decoder di conoscere quali PID sono utilizzati per individuare i diversi PES (audio, video) di uno stesso programma[57]. Questo avviene per mezzo delle

[55] in realtà le intestazioni dei pacchetti del TS possono essere estese e contenere più di 4 byte: in questo caso, la dimensione del payload si riduce, in modo che il totale sia ancora 188.

[56] Un payload vuoto è in realtà comunque riempito di 184 bytes inutili, e viene inserito da parte del multiplatore che realizza il TS per mantenere una riserva di banda che consenta di assecondare le fluttuazioni di velocità dei tributari.

[57] Altri PID riservati sono l'uno, che annuncia la presenza di una *Conditional Access Table* (CAT) contenente i parametri crittografici per visualizzare contenuti a pagamento, ed il PID 18, che annuncia la presenza della *Network Information Table* (NIT), che descrive altri TS disponibili. Per approfondimenti, vedi `http://en.wikipedia.org/wiki/Program-specific_information`.

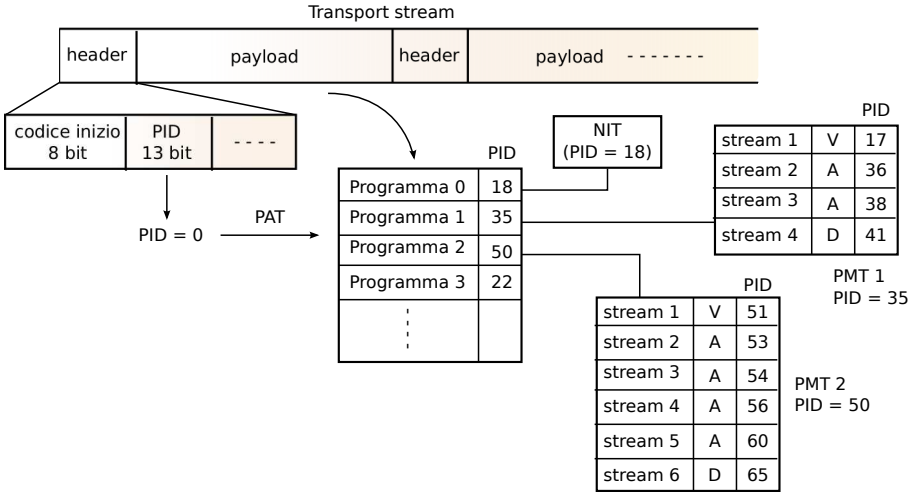


Figura 10.44: Estrazione della PAT e delle PMT dal Transport Stream

Program Map Table (PMT) rappresentate in fig. 10.44: ogni riga della PAT individua infatti un nuovo PID, alla ricezione del quale viene estratta dal payload la PMT che descrive i PID dei flussi che compongono il programma. Pertanto, quando uno spettatore seleziona un programma, il decodificatore cerca nella PAT il PID della PMT associata, e quindi inizia a prelevare i pacchetti intestati con ognuno dei PID trovati nella PMT, per ricostruire i relativi PES, sincronizzarli, ed iniziare la riproduzione.

L'opera

Trasmissione dei Segnali e Sistemi di Telecomunicazione

è il risultato di un progetto ventennale di cultura libera, aggiornato di continuo ed evolutosi fino alla forma attuale. La sua disponibilità pubblica è regolata dalle norme di licenza Creative Commons

Attribuzione - Non commerciale - Condividi allo stesso modo

`https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it`

e tutte le risorse relative al testo sono accessibili presso

`https://teoriadeisegnali.it/libro/`

Puoi contribuire al suo successo promuovendone la diffusione e supportarne lo sviluppo attraverso una donazione, in buona parte devoluta ai progetti *open source*[1] che ne hanno resa possibile realizzazione e divulgazione. Ai donatori viene accordato un accesso *vitalizio* al formato PDF *navigabile* di tutte le edizioni presenti *e future*.

[1]
- Lyx - `http://www.lyx.org/`
- LaTeX- `https://www.latex-project.org/`
- TeX Users Group - `https://tug.org/`
- Inkscape - `http://www.inkscape.org/`
- Gnuplot - `http://www.gnuplot.info/`
- Octave - `http://www.gnu.org/software/octave/`
- Geany - `https://www.geany.org/`
- Linux - `https://www.linux.it/`
- Free Software Foundation - `https://shop.fsf.org/`
- GNOME Foundation - `https://www.gnome.org/`
- Mozilla Foundation - `https://www.mozilla.org/it/`
- Wikipedia - `https://it.wikipedia.org`
- Internet Archive - `https://archive.org/about/`
- Creative Commons - `https://creativecommons.it/chapterIT/`
- WordPress - `https://it.wordpress.org/`
- Phplist - `https://www.phplist.org/`