Appunti sulla codifica video

Marco Cagnazzo

a.a. 2011–2012

Questi appunti costituiscono del materiale di supporto per il corso di Elaborazione dei Segnali Multimediali.Essi coprono alcuni aspetti relativi alla codifica video, ed in particolare quelli relativamente ai quali i libri di testo sono meno soddisfacenti.

Si prega di segnalare ogni tipo di errore scrivendo all'indirizzo cagnazzo@unina.it

Indice

1 La codifica video

Il segnale video digitale è costituito da una sequenza di fotogrammi o *frame*, ognuno dei quali può essere un'immagine in scala di grigi o a colori. In quest'ultimo caso, lo spazio dei colori più frequentemente utilizzato è quello YUV 4:2:0. Tipici valori della risoluzione del segnale video sono riportati in Tab. 1.

È abbastanza evidente che il segnale video non compresso (detto anche segnale video *grezzo*) richiede imponenti risorse per la memorizzazione e la trasmissione. Ad esempio un video in formato SD (simile per risoluzione a quello della televisione analogica tradizionale) ad una frequenza di 25 fotogrammi al secondo (fps) richiederebbe circa 250 Mbps per la trasmissione, o equivalentemente, circa 100 GB per la memorizzazione di una sola ora di video. Ovviamente questi requisiti sono difficili da soddisfare e fino a pochi anni fa praticamente impossibili. È possibile però ridurre drasticamente queste cifre grazie alla compressione o codifica video.

	Righe	Colonne
QCIF	144	176
CIF	288	352
SD	576	704
HD720	720	1280
HD1080	1080	1920

Tabella 1: Risoluzioni del segnale video

1.1 Classificazione delle tecniche di codifica

Per comprimere il segnale video si sfrutta la notevole ridondanza presente in esso. Si usa distinguere da un lato, la ridondanza *spaziale* (cioè la presenza nei singoli fotogrammi di aree omogenee) da quella temporale (cioè la somiglianza tra fotogrammi consecutivi) e, dall'altro, la ridondanza statistica (la dipendenza statistica dei pixel) da quella psicofisica (legata al fatto che non tutta l'informazione presente nel video viene effettivamente percepita da un osservatore umano). Le tecniche di codifica video sfruttano questi tipi di ridondanza. In particolare, praticamente tutti gli schemi di codifica video presentano uno stadio di compressione temporale ed uno di compressione spaziale, in cui si cercano di eliminare i rispettivi tipi di ridondanza. Un codificatore video può essere classificato sulla base di come sono realizzati i diversi stadi che lo compongono, come illustrato in Tab. 2.

Le tecniche di codifica senza compressione temporale sono le più semplici: ogni fotogramma viene compresso come se fosse una singola immagine, ad esempio con JPEG o con JPEG2000. Questo dà luogo alla versione "motion" dei vari standard: Motion-JPEG, Motion-JPEG2000. Queste tecniche hanno una grande semplicità concettuale ed implementativa, e sono utilizzate per applicazioni come il cinema digitale (i film proiettati in molte sale ormai sono in questo formato) o i programmi di video-chiamate su computer. Ovviamente tali tecniche hanno un rapporto di compressione ridotto, perché non sfruttano la ridondanza temporale.

Le tecniche di codifica ibride, così dette perché utilizzano due approcci diversi per la compressione spaziale (trasformata) e quella temporale, sono le più diffuse, e sono alla base di tutti i principali standard di codifica video (serie H.26x dell'ITU e serie MPEG-n dell'ISO [1, 2, 3, 4, 5, 6]). Esse hanno i migliori rapporti di compressione e coprono ogni intervallo di qualità finale, da livelli ottimi (utilizzati ad esempio per i DVD) a livelli bassi (utilizzati per ottenere la massima compressione come nel formato DiVX). Essi però hanno un limitato supporto alla scalabilità che è invece un requisito importante per la diffusione del video su rete (si veda la sezione 6).

Per contro, le tecniche basate su trasformata 3D hanno rapporti di compressione leggermente inferiori a quelli che si possono ottenere con codificatori ibridi a parità di qualità, ma forniscono un supporto naturale alla scalabilità. Queste tecniche sono allo studio per lo sviluppo di un nuovo standard di codifica video, indicato con SVC (scalable video coding).

	Spazio	Tempo	Vantaggi e svantaggi	Applicazioni	Usata in
Motion Picture	Trasformata	-	Semplicità computazionale e di implementazione, basso rapporto di compressione	Cinema digitale, video chat	M-JPEG, M-JPEG2000
Ibrido	Trasformata	Predizione	Ottimo rapporto di compressione ma scarsa scalabilità	Memorizzazione su supporto di massa	MPEG, H.264
Trasformata 3D	Trasformata	Trasformata	Ottima scalabilità, rapporti di compressione leggermente inferiori alle tecniche ibride	Video su rete	Nessuno standard
Altro	Varie tecniche tra cui VQ	Principalmente predizione	-	Poco rilevanti	-

Tabella 2: Tecniche di codifica video

Esistono poi approcci basati su tecniche diverse (ad es. quantizzazione vettoriale, VQ) ma al momento sembrano non competitivi con le tecniche ibride o basate su trasformata 3D, per cui sono raramente presi in considerazione.

1.2 Criteri di valutazione di una tecnica di compressione video

Un primo criterio di valutazione per una tecnica di compressione video è il tasso di codifica R, misurato in *bit per secondo* (bps), o equivalentemente il rapporto di compressione C. Detto B_{in} in numero di bit su cui è rappresentato il video all'ingresso e B_{out} il numero di bit su cui è rappresentato il video compresso, ed infine T la durata del video, si hanno le seguenti definizioni:

$$R = \frac{B_{\text{out}}}{T}\text{bps} \qquad\qquad C = \frac{B_{\text{out}}}{B_{\text{in}}}$$

Più raramente, il tasso di codifica può essere definito come $R = \frac{B_{\text{out}}}{PK}$ dove P è il numero di pixel per immagine ed K è il numero di immagini che compongono il video. In questo caso R è misurato in *bit per pixel* (bpp).

Esclusi i rari esempi di tecniche di codifica video senza perdite (*lossless*), è necessario anche definire la *qualità*, o equivalentemente la *distorsione* del video decodificato. I criteri di valutazione della qualità (detti spesso *metriche*) si possono classificare in oggettivi e soggettivi.

I criteri soggettivi si basano sulla raccolta e sull'analisi statistica di giudizi di qualità pronunciati da un opportuno insieme di osservatori. La valutazione soggettiva della qualità di un immagine o di un video è la più affidabile, tuttavia si tratta di un metodo lento e costoso: tipicamente le immagini sono sottoposte al giudizio di decine di persone, secondo una configurazione ben precisa (tutti i parametri sono predefiniti, dalla distanza tra lo schermo e l'osservatore, alla tonalità di grigio delle pareti della stanza).

I criteri oggettivi sono invece definiti come una funzione matematica del video originale $I_k(\mathbf{p})$ e di quello decodificato $\widehat{I}_k(\mathbf{p})$. Una metrica oggettiva può essere calcolata da un computer, in modo estremamente più veloce ed economico rispetto ad una metrica soggettiva. Idealmente, un criterio oggettivo dovrebbe discriminare la qualità di due immagini (o video) allo stesso modo di una misura soggettiva, in particolare mantenendo lo stesso ordinamento (cioè per ogni coppia di immagini A e B, se soggettivamente A è migliore di B, la metrica oggettiva deve mantenere lo stesso ordine di preferenza). Purtroppo non è stata ancora trovata una metrica oggettiva che soddisfi pienamente questo requisito.

In particolare i criteri oggettivi si dividono in percettivi e non percettivi, a seconda che essi prendano o non prendano in conto le caratteristiche specifiche della visione umana. Le più comuni misure oggettive non

percettive sono l'errore quadratico medio (o MSE) e il rapporto segnale-rumore di picco (o PSNR), definiti come segue:

$$\mathrm{MSE} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{P}\sum_{\mathbf{p}}\left[I_k(\mathbf{p}) - \widehat{I}_k(\mathbf{p})\right]^2$$

$$\mathrm{PSNR} = 10\log_{10}\frac{(2^b-1)^2}{\mathrm{MSE}}$$

dove b è la dinamica (numero di bit) su cui è rappresentato il segnale (tipicamente $b = 8$) e quindi $(2^b - 1)^2$ è il massimo valore possibile (valore di picco) della potenza del segnale utile. Come si evince dalle definizioni, una tecnica che massimizza l'una delle due misure, massimizza anche l'altra. Il PSNR e l'MSE sono spesso usate come metriche di qualità perché:

- hanno una correlazione abbastanza buona con la qualità percepita da un osservatore umano (qualità soggettiva);
- sono relativamente semplici da calcolare;
- l'MSE è facile da interpretare perché corrisponde (a meno di una normalizzazione e di una radice quadratica) alla metrica euclidea nello spazio $\mathbb{R}^{PK}$;
- è relativamente facile ottimizzare una tecnica di codifica rispetto all'MSE (e quindi al PSNR).

Nonostante queste caratteristiche, a volte è preferibile usare metriche percettive, in quanto l'MSE può non tenere in conto opportunamente alcuni caratteristiche del sistema percettivo umano. Alcune di queste sono:

- maggiore sensibilità alle basse frequenza spaziali;
- maggiore sensibilità alle basse frequenze temporali;
- effetti di mascheramento spaziale (piccoli errori vicino a zone di brusca variazione non vengono percepiti);
- effetti di mascheramento temporale (una brusca variazione temporale del segnale comporta una ridotta sensibilità per un breve periodo);
- fenomeni di completamento (il cervello tende a vedere ciò che ci si aspetta di vedere).

L'insieme di questi fenomeni rende poco affidabile l'MSE per una valutazione accurata della qualità percepita. Diverse metriche percettivi sono state suggerite, tra cui le più popolari sono il PSNR pesato (Weighted PSNR, o WPSNR) [7] e l'indice di somiglianza strutturale (Structure similarity index, SSIM) [8]. Anche se le metriche percettive sono meglio correlate alla qualità soggettiva delle immagini rispetto al PSNR ed all'MSE, esse non sono universalmente utilizzate per i seguenti motivi:

- sono più complesse da calcolare;
- non permettono una ottimizzazzione analitica degli algoritmi di compressione;
- la correlazione con la qualità soggettiva, per quanto buona, non è perfetta.

In ogni caso, la qualità delle immagini (o del video) a valle della decodifica, cresce al crescere del tasso di codifica. Per questo motivo, per confrontare due tecniche di compressione, è necessario tracciare le rispettive curve tasso-distorsione. È evidente che i requisiti di un basso tasso di codifica e di un'alta qualità sono contraddittori.

Un altro criterio di valutazione di una tecnica di compressione è la complessità. In molti casi una tecnica di codifica più complessa permette di migliorare la qualità a parità di tasso, o di ridurre il tasso a parità di qualità. D'altra parte, non sempre è possibile ricorrere a tecniche molto complesse, per motivi come:

- limiti intrinseci alla complessità hardware e software del sistema;
- esigenza di codifica o decodifica in tempo reale;
- limiti legati alla batteria di sistemi portatili.

Si vorrebbe dunque una tecnica con la minima complessità possibile. Questo requisito è contraddittorio con quelli di basso tasso ed alta qualità.

In alcuni contesti, alla tecnica di compressione è richiesto un certo grado di robustezza. Ciò significa che la qualità del segnale decodificato deve essere quanto meno influenzata possibile dalla perdita di una parte dei dati. Ovviamente questo requisito è importante quando il segnale è trasmesso su reti poco affidabili (p. e. wireless). Ci sono vari modi per rendere più robusto un segnale codificato: si possono per esempio usare tecniche di codifica di canale, oppure tecniche cosiddette *a descrizioni multiple*, che associate ad opportune strategie di trasmissione possono garantire una qualità minima sufficiente anche in ambienti molto rumorosi. Tuttavia il requisito di robustezza richiede un aumento di complessità ed un aumento del tasso di codifica (o della distorsione a parità di tasso).

Un'ultima funzionalità che spesso viene considerata al momento di scegliere la tecnica di codifica è la scalabilità, di cui si parlerà nella sezione 6. Come accennato in precedenza, le tecniche basate su trasformata 3D consentono di ottenere la scalabilità praticamente senza influenzare né la complessità, e con soltanto una piccola riduzione delle prestazioni tasso-distorsione. Le tecniche ibride invece possono essere rese scalabili tipicamente con un forte aumento della complessità e con una degradazione non trascurabile del compromesso tasso-distorsione.

Come si deduce da questa breve introduzione, sono molti i criteri sulla base dei quali si può scegliere una tecnica di codifica video, e la soluzione migliore dipende da tutti i parametri del problema. Ciò spiega l'esistenza del grande numero di tecniche di codifica video che si osservano attualmente.

2 La compressione spaziale

La compressione spaziale nell'ambito della codifica video è tipicamente implementata con tecniche direttamente mutuate dalla codifica di immagini fisse. Ad esempio negli standard H.261, H.263, MPEG-1 e MPEG2 la compressione spaziale è concettualmente identica alla codifica con JPEG. In H.264 (il più recente e complesso standard di codifica video) la compressione spaziale è più sofisticata, e prevede l'uso di tecniche di predizione spaziale seguite dalla trasformata dell'errore di predizione risultante.

Tra le tecniche di compressione spaziale che si è provato ad utilizzare per la codifica video va citata la quantizzazione vettoriale. Tuttavia, data l'intrinseca complessità di tale tecnica, essa non ha riscontrato successo, e non si conoscono tecniche di compressione spaziale basata su VQ con prestazioni confrontabili con quelle che si possono ottenere con le tecniche basate su trasformata.

3 La compressione temporale

Le tecnica di compressione temporale è quella che più caratterizza un codificatore video. In effetti nella classificazione riportata in Tab. 2, la differenza principale tra le varie tecniche risiede proprio nella parte temporale. I due approcci principali alla compressione temporale sono quello predittivo e quello con trasformata. Per quanto riguarda le tecniche predittive, considereremo tre schemi di complessità via via crescente:

- codifica differenziale (DPCM)
- conditional replenishment (CR)
- compensazione del movimento (motion compensation, MC)

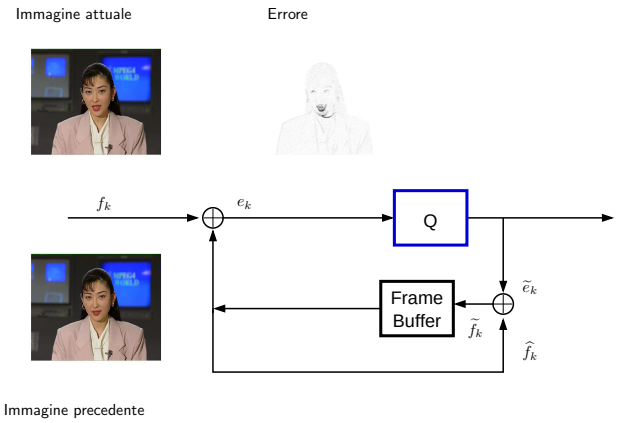


Figura 1: Schema di principio di un codificatore DPCM

3.1 Codifica differenziale

Nel caso di **codifica differenziale** si sfrutta il fatto che fotogrammi consecutivi di una sequenza video sono molto simili tra loro. Invece di trasmettere il fotogramma originale si può allora trasmettere la differenza tra quest'ultimo ed il precedente. In altri termini stiamo applicando la tecnica DPCM, ma al livello di fotogramma invece che campione per campione come illustrato in Fig. 3.1. Si noti la presenza di una memoria nell'anello di retroazione. Tale memoria consente di effettuare la predizione *esattamente allo stesso modo di come la può effettuare il decodificatore*. Ciò è importante perché assicura la corretta decodifica.

La tecnica DPCM è estremamente semplice, è può risultare conveniente quando la scena è molto statica. Tuttavia bisogna tenere presente che le tecniche differenziali potenzialmente aumentano la dinamica del segnale (a causa dell'operazione di sottrazione), per cui nelle aree del fotogramma dove c'è movimento, l'errore di predizione può diventare notevole, e rendere non conveniente questo approccio.

3.2 Conditional replenishment

Si può pensare di migliorare la codifica differenziale tramite il **conditional replenishment** (CR). L'immagine è divisa in blocchi (la terminologia corrente è *macroblocchi* – MB – perché le dimensioni sono maggiori rispetto ai blocchi utilizzati nella parte di compressione spaziale), e per ognuno di essi si verifica se ci sono stati cambiamenti significativi rispetto al fotogramma precedente. Se è così, il MB va ritrasmesso (modo *new*), altrimenti si dice che il CR ha avuto successo, e si può decidere di non trasmettere nulla (modo *skip*) o di trasmettere l'errore di predizione (modo *refine*). Ovviamente bisogna indicare al decodificatore qual è la modalità scelta, attraverso ad esempio un codice a due bit (00 per new, 10 per skip, 11 per refine). Questi bit, che non rappresentano direttamente valori dei campioni del segnale, sono detti *side information*.

In formule, sia $B_k^{(\mathbf{p})}$ il MB centrato nel pixel $\mathbf{p} = (p_r, p_c)$ nella frame k. Si consideri una funzione che fornisca una misura di similitudine tra due macroblocchi, e la si indichi con $d(\cdot,\cdot)$. Tipiche funzioni di similitudine (a volte con leggero abuso chiamate anch'esse metriche) sono la SAD (somma delle differenze assolute) e la SSD (somma delle differenze al quadrato). Il CR prevede quindi di calcolare

$$d = d\left(B_k^{(\mathbf{p})}, B_{k-1}^{(\mathbf{p})}\right)$$

cioè di confrontare il MB corrente con quello che si trova nella stessa posizione ma nella frame precedente. Se d è inferiore ad un'opportuna soglia γ i blocchi sono ritenuti simili e quindi si può scegliere tra i modi skip

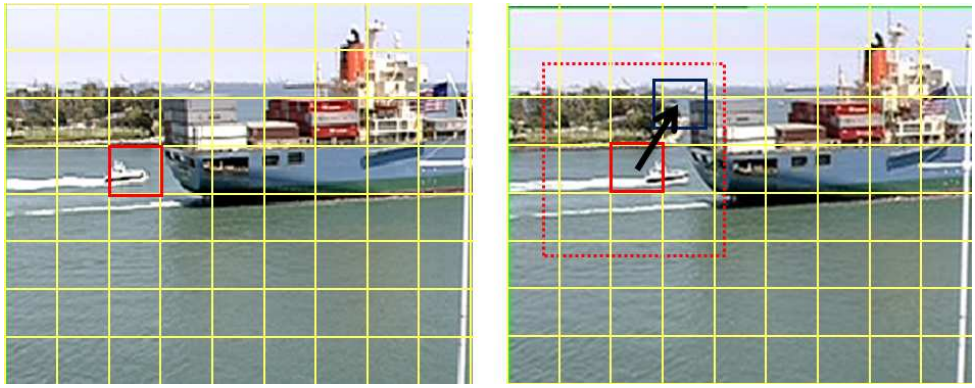


Figura 2: Stima del movimento

e refine; se invece i blocchi sono diversi, si usa il modo new. Il CR realizza una sorta di DPCM adattativo, stimando quando può essere conveniente usare la codifica differenziale e quando invece è meglio mandare i valori originali.

I parametri che influenzano le prestazioni del CR sono le dimensioni dei blocchi e il valore della soglia. Avere blocchi grandi riduce le probabilità di successo del CR, ma quando ciò accade si codificano con pochi bit ampie regioni, ed inoltre si riduce la side information. Quindi è opportuno usare blocchi grandi per sequenze molto statiche. Valori piccoli della soglia riducono la probabilità di successo del CR, e permettono la sostituzione di un MB solo quando esso è molto simile a quello di riferimento. Quindi il valore della soglia permette di realizzare un trade-off tra tasso e distorsione.

3.3 Stima e compensazione del movimento

Il modello su cui si basa il CR fallisce in alcune condizioni molto comuni. Si consideri per esempio una scena fissa ripresa da una telecamera in movimento. Nonostante fotogrammi consecutivi siano molto ridondanti, il CR fallirà quasi sempre, perché i MB simili tra loro *non si trovano nella stessa posizione* quando si passa da un'immagine alla successiva. Bisogna pensare allora ad integrare nel modello del segnale dei parametri che tengano conto del movimento della scena. Questo è fatto con le tecniche di **stima e compensazione del movimento** (ME, motion estimation e MC, motion compensation). La stima del movimento consiste nel calcolare un insieme di parametri che descrivono il movimento relativo tra due fotogrammi. La compensazione consiste nel produrre una stima accurata del fotogramma corrente utilizzando tali parametri insieme con il fotogramma di riferimento. In questi appunti faremo riferimento alle tecniche di ME/MC basate sui MB, ma si tenga conto che altri modelli sono possibili.

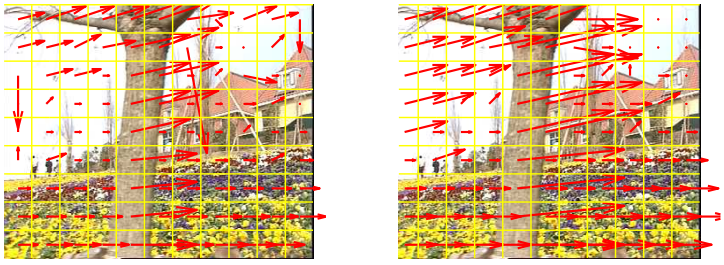
L'idea della MC basata su MB è quella di predire il MB corrente $B_p^{(\mathbf{k})}$ utilizzando un MB che non si trova necessariamente nella stessa posizione nella frame di riferimento. Si definisce allora un vettore di movimento (VM), indicato con $\mathbf{v}$, che permette di passare dal pixel $\mathbf{p}$ al pixel $\mathbf{p}+\mathbf{v}$. Si può definire la funzione:

$$d(\mathbf{v}) = d\left(B_k^{(\mathbf{p})}, B_{k-1}^{(\mathbf{p}+\mathbf{v})}\right) \tag{1}$$

che valuta la somiglianza del MB corrente con quelli della frame precedente che si trovano intorno alla posizione $\mathbf{p}$. La ME si realizza cercando il VM che minimizza la funzione d:

$$\mathbf{v}^* = \arg\min_{\mathbf{v}\in W} d(\mathbf{v})$$

dove W è un opportuno insieme di vettori di test, detto finestra. Tipicamente la finestra è fatta da vettori le cui componenti hanno valori compresi tra $-r$ ed r, ed in tal caso r è detto raggio della finestra. Il vettore $\mathbf{v}^*$ è



(a) (b)

Figura 3: Esempi di stima del movimento: (a) campo di vettori non regolarizzato, (b) campo regolarizzato

quello che permette di selezionare il MB più simile tra quelli *intorno* alla posizione $\mathbf{p}$. Se tra i due fotogrammi c'è stato un movimento puramente traslatorio, il vettore stimato rappresenta il moto del MB. Il procedimento è illustrato in Fig. 2.

La compensazione del movimento generalizza l'approccio del CR, che in effetti si può vedere come una ME/MC con finestra che contiene il solo vettore nullo, $W = \{(0,0)\}$. Una volta calcolato $d(\mathbf{v}^*)$ si confronta tale valore con la soglia, e si può quindi decidere tra i modi new e refine. In effetti la MC consiste nel costruire una predizione della frame k usando $B_{k-1}^{(\mathbf{p}+\mathbf{v}^*)}$ al posto di $B_k^{(\mathbf{p})}$ quando $d(\mathbf{v}^*) < \gamma$. L'errore di predizione è poi mandato in ingresso allo stadio di compressione spaziale. L'effetto dei parametri (soglia e dimensione dei blocchi) è simile a quello che si ha nel caso del CR.

3.4 Regolarizzazione della stima del movimento

Nella figura 3.3(a) è illustrato il campo di VM che si ottengono minimizzando blocco per blocco il criterio definito nell'equazione (1). Si può notare che, soprattutto in corrispondenza delle aree omogenee, il VM stimato è a volte scorrelato dai suoi vicini e dal movimento degli oggetti. Nonostante il blocco puntato dal vettore si quello che minimizza la misura di dissimilarità, è possibile effettuare delle scelte migliori. Per esempio si può scegliere di minimizzare un criterio più complesso di quello nell'eq. (1), penalizzando i vettori meno regolari. Introducendo allora una funzione di costo $r(\mathbf{v})$, la ME può essere definita come:

$$\mathbf{v}^* = \arg\min_{\mathbf{v}\in W} \left[d(\mathbf{v}) + \lambda r(\mathbf{v}) \right]. \tag{2}$$

La funzione di costo può essere legata ad esempio al gradiente spaziale del campo di VM, oppure può essere il costo di codifica del vettore corrente dati i suoi vicini già codificati. In figura 3.3(b) è mostrato il campo di VM ottenuto con un vincolo di regolarità (r è la norma della differenza tra $\mathbf{v}$ ed il valore mediano dei suoi vicini). Spesso dei vettori leggermente meno precisi (in termini di energia dell'errore di compensazione risultante) possono essere molto meno costosi da codificare, e quindi i bit "risparmiati" sui vettori possono essere utilizzati per codificare con maggiore precisione il residuo, permettendo quindi di raggiungere una maggiore qualità a parità di tasso. Il parametro λ gestisce il compromesso tra costo di codifica (o regolarità) dei VM e la loro precisione. Quando λ è piccolo, si privilegiano vettori precisi. Quando è grande, si scelgono piuttosto vettori poco costosi. Per ogni valore del bit-rate totale disponibile, in principio esiste un valore ottimale di λ che permette di scegliere il miglior compromesso tra precisione e costo dei vettori (cioè quello che porta alle migliori prestazioni tasso-distorsione globali). Tale valore potrebbe essere determinato ripetendo molte volte la ME e poi la codifica con valori diversi e scegliendo infine migliore. Si comprende tuttavia che tale approccio è estremamente complesso. Esistono allora delle relazioni trovate sperimentalmente che legano il passo di quantizzazione dei coefficienti della trasformata spaziale (e quindi il tasso totale) con il valore di λ da utilizzare. Questa relazione dipende fortemente dal tipo di codificatore utilizzato.

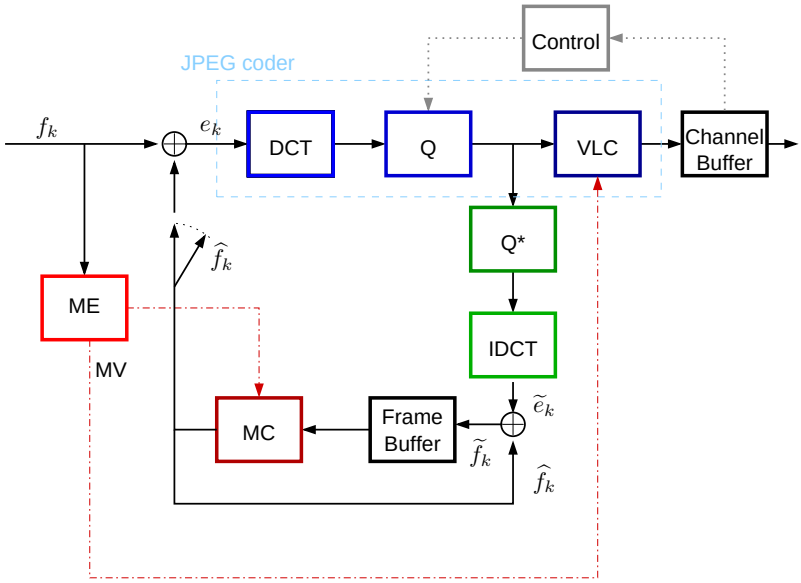


Figura 4: Schema di principio del codificatore ibrido

Generalizzando, le tecniche di stima del movimento si possono distinguere sulla base del *modello di moto* utilizzato, della *funzione di costo* $d(\cdot,\cdot)$, e della *strategia di ricerca*. Oltre al modello di moto basato su MB visto in questi appunti, è possibile considerare modelli di moto che prevedono la traslazione dell'intero fotogramma, che prendono in conto traslazioni e rotazioni, oppure ancora che considerano possibili deformazioni dell'immagine. Il modello basata su MB è tuttavia un'ottimo compromesso tra complessità e prestazioni, per cui è di gran lunga il più comune. Per quanto riguarda le funzioni di costo, SSD e SAD sono le più comuni, ma esistono anche metriche che cercano di tener conto di altri aspetti (ad esempio variazioni di luminosità, sensibilità dell'osservatore umano). Infine, le strategie di ricerca riguardano il modo in cui si cerca il VM in W. Il modo più semplice è quello di considerare tutti i vettori nella finestra e valutare quello ottimale. Tuttavia la complessità di questo approccio, chiamato *full search* è notevole. Esistono tecniche subottime, per le quali non è garantito che si trovi il vettore ottimo $\mathbf{v}^*$, ma che consentono di trovare dei vettori con valori di d vicini all'ottimo con una complessità molto ridotta. Tali tecniche sono basate sulla scansione della finestra lungo particolari cammini, che non obbligano a calcolare $d(\mathbf{v})$ su tutti i vettori. Tra queste tecniche citiamo la *cross-search*, che stima prima la componente orizzontale e poi quella verticale del movimento, seguendo un cammino a forma di croce, e la *log-search*, che determina in passi successivi una tra otto direzioni possibili, e dimezza ad ogni passo la distanza alla quale effettuare la ricerca.

4 Il codificatore ibrido

Il codificatore ibrido è il più diffuso tipo di codificatore video. Tutti gli standard attuali usano uno schema di questo tipo, mostrato in Fig. 4.

Esso si basa su stima e compensazione del movimento per produrre l'errore di predizione, da codificare poi con una tecnica spaziale che il più delle volte è concettualmente coincidente con JPEG.

4.1 Scelta dei modi in un codificatore ibrido

In un codificatore ibrido, ogni macroblocco può essere codificato utilizzando un insieme di tecniche diverse, dette *modi* di codifica. Ad esempio, un MB può anche codificato utilizzando unicamente lo stadio spaziale, senza effettuare predizione da altre immagini. Questo modo è detto *intra*. Quando invece si usa la predizione temporale, si parla di modi *inter*. Nei codificatori ibridi, la predizione temporale è ottenuta con stima e compensazione del movimento. È possibile utilizzare diversi modi inter, per esempio cambiando la dimensione dei blocchi per la ME/MC, aumentando il numero di immagini che possono essere usate come riferimento, o cambiandole. Si pone allora il problema di come scegliere il miglior modo di codifica.

Questo problema può essere posto in termini di un'ottimizzazzione vincolata. Sia Q_P il passo di quantizzazione dei coefficienti della trasformata. Il tasso di codifica del k-esimo MB dell'immagine corrente dipende dal modo scelto, indichiamolo con i_k, e da Q_P. Altrettanto si può dire per la distorsione. Possiamo allora scrivere, per l'intera immagine (o se si vuole per l'intera sequenza):

$$D = \sum_k D_k(Q_P, i_k) \qquad\qquad R = \sum_k R_k(Q_P, i_k)$$

Supponendo di avere un vincolo sul tasso totale $R \leq R_{\text{tot}}$, possiamo passare al problema (non vincolato) della minimizzazione di J:

$$\begin{aligned} J &= \sum_k D_k(Q_P, i_k) + \lambda_{\text{mode}} \sum_k R_k(Q_P, i_k) \\ &= \sum_k \left[D_k(Q_P, i_k) + \lambda_{\text{mode}} R_k(Q_P, i_k) \right] \end{aligned}$$

In genere si rinuncia ad un'ottimizzazione globale, e si sceglie il modo di codifica tale che, macroblocco per macroblocco sia minimizzata la quantità:

$$J(Q_P, i_k, \lambda_{\text{mode}}) = \left[D_k(Q_P, i_k) + \lambda_{\text{mode}} R_k(Q_P, i_k) \right] \tag{3}$$

In effetti il modo scelto dipende quindi da Q_P e da λ_{mode}. Il passo di quantizzazione è spesso considerato come un dato di ingresso del problema. Il valore di λ_{mode} è invece determinato sperimentalmente in funzione di Q_P in modo da ottenere la minima distorsione.

In conclusione dato il passo Q_P, viene scelto λ_{mode} secondo una relazione sperimentale. A questo punto per ogni macroblocco k si sceglie il modo di codifica i_k che minimizza il criterio definito nell'equazione (3).

4.2 Il ruolo del buffer di canale

Il flusso di dati prodotto dal codificatore ha un bit-rate variabile, che dipende molto dall'efficacia della ME/MC. Per ridurre la variabilità del bit-rate in uscita, si usano tecniche basate su un buffer di canale. Quando il grado di riempimento del buffer supera un certo limite, la quantizzazione viene resa più grossolana, in modo da ridurre il tasso, e viceversa più fine quando il buffer si sta svuotando. Si verifica quindi che nelle scene con poco movimento il passo di quantizzazione tende a scendere rapidamente, consentendo ad un osservatore umano di apprezzare i dettagli. Quando c'è molto moto i dettagli vengono quantizzati, ma d'altra parte in tali condizione un osservatore non sarebbe comunque in grado di apprezzarli.

4.3 Il decodificatore ibrido

In Fig. 5 è rappresentato il decodificatore. Si noti che, come in generale accade negli schemi di codifica predittiva, esso è un sottoinsieme dei moduli presenti nel codificatore.

Lo schema ibrido è quindi *asimmetrico* nel senso che il codificatore è molto più complesso del decodificatore. Ciò è adeguato nel caso in cui la codifica venga effettuata una sola volta e la decodifica molte (esempio:

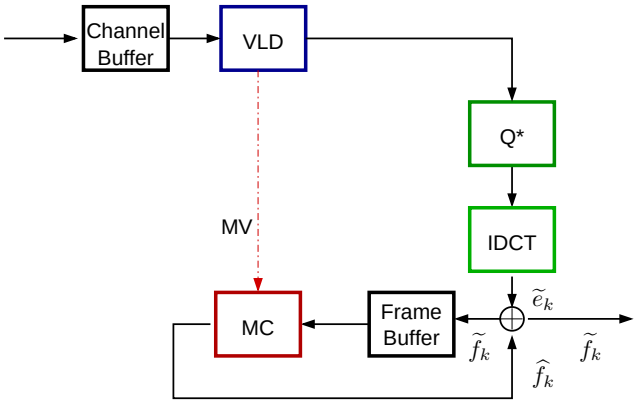


Figura 5: Decodificatore ibrido

distribuzione di video su DVD o su rete). Tuttavia esistono applicazioni dove si richiede uno schema simmetrico (ad esempio, videocomunicazione) per le quali spesso si ricorre a tecniche di codifica senza predizione temporale, o anche casi in cui si preferisce avere molti codificatori semplici ed un sono decodificatore complesso (ad esempio reti di sensori video che inquadrano la stessa scena da punti diversi), per le quali si usano tecniche di codifica completamente diverse (si parla di codifica video distribuita) e che integrano concetti legati alla codifica di canale. Tuttavia il codificatore video ibrido rimane quello più largamente usato per la maggior parte delle applicazione, ed è senz'altro il riferimento a cui confrontarsi anche quando si considerano problemi come la codifica simmetrica o quella distribuita.

5 Lo standard di codifica MPEG-1

Lo standard di codifica video MPEG-1 è stato sviluppato dall'ISO all'inizio degli anni novanta. Pur essendo attualmente obsoleto, esso ha dato luogo in seguito a MPEG-2, che è largamente usato nella distribuzione video su supporti di massa (DVD) e via etere (digitale satellitare e terrestre). In effetti MPEG-2 non differisce tecnicamente da MPEG-1 se non per due aspetti: la capacità di supportare tassi di codifica più elevati (quindi miglior qualità) ed il supporto alla scalabilità.

MPEG-1 definisce un modo standardizzato di rappresentare audio e video digitali. Lo strato audio di MPEG-1 non sarà trattato in queste dispense, ma vale la pena notare che esso ha avuto un successo enorme: il formato MP3 non è altro che la versione più completa (il cosiddetto strato 3) del formato audio di MPEG-1.

Il codificatore video di MPEG-1 è semplicemente un codificatore video ibrido con *stima del movimento frazionaria* e con il supporto ai tipi di immagine.

5.1 La compensazione del movimento a precisione frazionaria

La stima del movimento nei primissimi codificatori video (come H.261) permetteva di descrivere soltanto traslazioni di ampiezza esprimibili su di un numero intero di pixel (stima a precisione intera). Tuttavia, quando un oggetto si muove, la sua posizione non corrisponde necessariamente alla griglia di pixel sui quali è rappresentata la scena. MPEG-1 ed MPEG-2 consentono VM con precisione al mezzo pixel. Stimare il movimento su una griglia con risoluzione più fine può migliorare notevolmente l'affidabilità della predizione. Come mostrato in figura 6, poiché non c'è nessun pixel nelle locazioni a coordinate frazionarie, è necessario interpolare le frame per ottenere i valori di luminanza in tali posizioni. MPEG-1 e 2 usano l'interpolazione

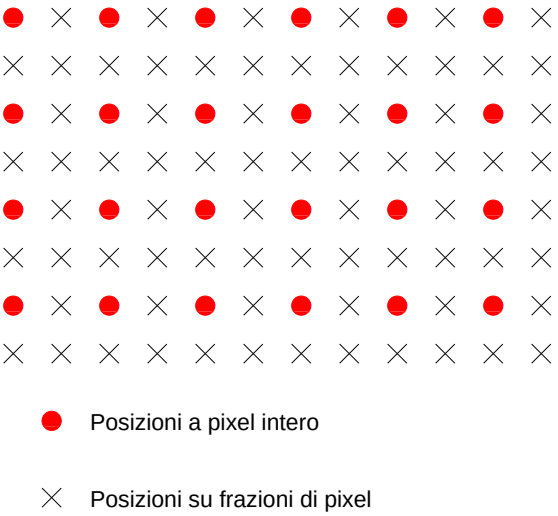


Figura 6: Griglie di campionamento

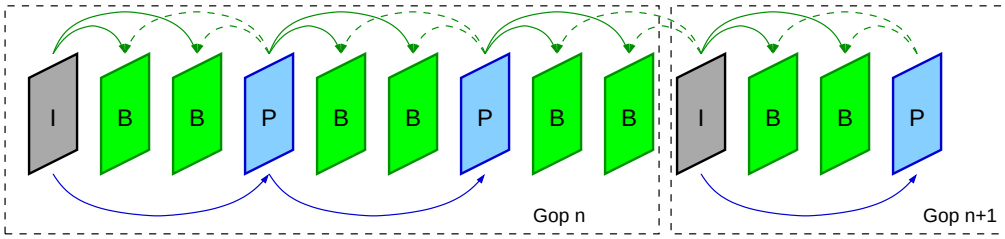


Figura 7: Tipi di immagini e struttura del GOP in MPEG-1

bilineare, per la sua semplicità. Si noti che gli stessi VM vengono usati per compensare sia la componente di luminanza che le componenti di crominanza: se quest'ultime sono sottocampionate (formato 4:2:0), i vettori utilizzati per compensarle sono ottenuti dimezzando quelli relativi alla luminanza.

5.2 Tipi di immagine in MPEG-1

Le immagini di un video codificato con MPEG-1 possono essere di quattro tipi:

I Immagini Intra

P Immagini Predictive

B Immagini Bidirectional

D Immagini DC-coded [1]

Le immagini sono organizzate in una struttura chiamata GOP (group of pictures), illustrata in Fig. 7. Un GOP comincia sempre con una immagine I. Non ci sono altre I in un GOP per cui la distanza N tra due I è uguale al numero di immagini del GOP. Tale parametro è fisso in un singolo video MPEG, ma può variare

[1] Le immagini D sono raramente usate e sono state soppresse dalle versioni successive di MPEG, per cui esse non saranno trattate in queste dispense.

tra un video e l'altro. Le immagini I e le immagini P sono dette *anchor frame*. Solo le anchor frame possono essere usate come riferimento per la ME/MC. Tra due anchor frame c'è sempre lo stesso numero M di frame B. Quindi la struttura di tutti i GOP di un video MPEG è univocamente definita dai parametri M ed N.

Le immagini Intra sono codificate senza predizione temporale: tutti i MB sono codificati in modo Intra. In pratica sono codificate con una tecnica concettualmente identica a JPEG. Non richiedendo stima e compensazione del movimento, la codifica Intra è computazionalmente meno onerosa delle altre. Tuttavia esse saranno anche meno compresse della immagini con predizione temporale. La presenza di frame I rende molto facile l'implementazione delle seguenti funzionalità:

Accesso casuale. È possibile accedere ad un qualsiasi GOP senza dover decodificare tutti i GOP precedenti, perché le I sono indipendenti dalle immagini procedenti

Avanti veloce. È possibile decodificare solo le immagini I senza richiedere una complessità computazionale elevata. Se si volesse implementare l'avanti veloce con immagini predette, bisognerebbe semplicemente decodificare più rapidamente il video, il ché significa installare una potenza di calcolo notevolmente superiore al quello che serve per la riproduzione normale.

Recupero dagli errori/Robustezza . Se in un GOP ci sono errori di qualsiasi natura, la codifica predittiva ne propaga gli effetti. Tuttavia la presenza di immagini I ne confina la propagazione: il nuovo GOP non ne sarà influenzato.

Le immagini P sono codificate con ME/MC. Per ogni macroblocco si effettua la stima del movimento e si decide la modalità di codifica, che può essere Intra o Inter. Tutta via ogni MB di un'immagine P può essere predetta unicamente dall'anchor frame precedente, come illustrato in Fig. 7 (frecce blu). Le immagini P hanno una complessità di codifica maggiore rispetto alle I, ma anche una maggiore compressione.

Infine le immagini B sono codificate con ME/MC, ma ogni macroblocco può essere predetto sia dalla anchor frame precedente che da quella successiva, sia come media delle due (modo *bidirezionale*). Le immagini B richiedono quindi due stime del movimento, ed inoltre l'invio dell'informazione supplementare relativa al riferimento usato. In conclusione le immagini B sono quelle caratterizzate dalla maggiore complessità e dal maggiore rapporto di compressione.

È opportuno sottolineare che lo standard non specifica né quale deve essere la tecnica usata per la stima del movimento, né i valori di N e M. Questo permette alle implementazioni del codificatore di gestire liberamente il compromesso tra i vari parametri considerati. Ad esempio tecniche di stima del movimento del tipo *log-search* permettono di ridurre enormemente la complessità della ME con piccole perdite nelle prestazioni RD; cambiando i valori dei parametri M ed N si può modificare il compromesso tra complessità, compressione, qualità e robustezza. Tutte queste modifiche non influenzano la conformità con lo standard.

6 MPEG-2 e la scalabilità

L'obiettivo della codifica video è quello di ottimizzare la qualità della sequenza decodificata per un assegnato tasso. In un sistema di comunicazione tradizionale, il codificatore comprime il segnale di ingresso fino ad ottenere un tasso che è inferiore ma prossimo alla capacità del canale che verrà usato per la trasmissione. Il decodificatore ricostruisce la sequenza video utilizzando tutta l'informazione prodotta dal codificatore. In un tale modello, sono sottintese due ipotesi: il codificatore conosce la capacità del canale; ed il decodificatore è abbastanza veloce per decodificare tutti i bit man mano che li riceve dal canale. Tuttavia non in tutti gli scenari applicativi queste due ipotesi sono verosimili.

6.1 La scalabilità

Quando si considerano applicazioni come la distribuzione di video su rete, l'obiettivo del codificatore cambia leggermente, e le due ipotesi fatte in precedenza possono non essere soddisfatte. Infatti in questo caso

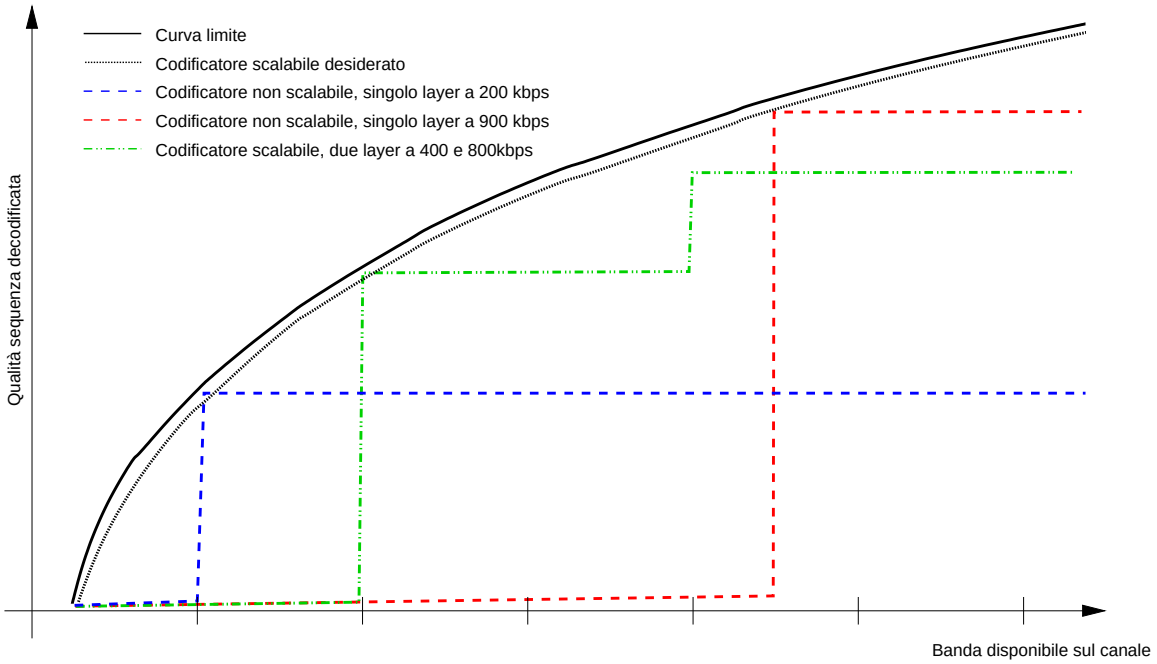


Figura 8: Diagramma della scalabilità

la capacità del canale di comunicazione varia nel tempo ed inoltre i destinatari della trasmissione possono essere molto diversi tra loro sia in termini di velocità di trasmissione disponibile, sia in termini di potenza computazionale. Di conseguenza, il codificatore non sa qual è tasso a cui ottimizzare il video; in secondo luogo, il decodificatore deve spesso condividere le risorse computazionali con altre applicazioni, e potrebbe non essere in grado di lavorare in tempo reale. Quindi, l'obiettivo della codifica video per la distribuzione su rete diventa: ottimizzare la qualità della sequenza decodificata per *un intervallo di tassi* $[R_1, R_2]$, anziché per un singolo tasso. Il bitstream viene prodotto al bit-rate massimo R_2, ma dovrebbe essere decodificabile (almeno in parte) per ogni tasso $R \in [R_1, R_2]$, con la qualità ottimale per quel tasso. Questa proprietà è detta *scalabilità* del flusso codificato. La figura 8 illustra alcuni aspetti di questo concetto.

L'asse orizzontale indica il tasso disponibile sul canale, e quello verticale la qualità del video ricostruito. La curva a tratto continuo indica il limite superiore della qualità, $Q^*(R)$ per un dato tasso R. In altre parole, $Q^*(R)$ è la qualità che può essere ottenuta usando il miglior codificatore possibile, nell'ipotesi che si conosca perfettamente il tasso R con cui è possibile trasmettere sul canale, ed che il decodificatore sia in grado di decodificare la sequenza ricevuta a tale tasso. Le due curve a gradino tratteggiate, rappresentano il caso di un codificatore che ottimizza la qualità per un assegnato bit-rate, ad esempio 200 o 900 kbps. In questo caso si parla di codifica *non scalabile*. Una volta fissato l'algoritmo di codifica (per esempio, un codificatore ibrido come MPEG-1), e fissato anche il tasso-obiettivo, un codificatore non scalabile cerca di ottimizzare la qualità per quel singolo tasso, portando il vertice superiore della curva a gradino quanto più vicino possibile alla curva limite. Se il tasso disponibile sul canale coincide con il tasso di codifica, le prestazioni sono le migliori. Se invece il tasso del canale è inferiore a quello di codifica, si verifica il cosiddetto "digital cutoff", e la qualità del video decodificato è estremamente insoddisfacente. D'altra parte, se il tasso disponibile sul canale è superiore al tasso di codifica, le prestazioni non hanno alcun beneficio. Quello che si vorrebbe invece, è che il flusso fosse codificato in modo da sfruttare al meglio le risorse disponibili sul canale. Questa situazione è rappresentata idealmente dalla curva punteggiata in figura 8: il flusso è decodificabile a tutti i valori di tasso disponibile, ed al crescere di tale parametro, cresce anche la qualità della sequenza decodificata. Vediamo come i codificatori

reali cercano di approssimare questo comportamento ideale.

6.2 La codifica scalabile a strati

Un flusso codificato in modo scalabile (sia esso rappresentativo di video, immagini, audio o quant'altro) è costituito da un insieme di N strati (il termine inglese corrispondente, più comune, è *layer*). Questi strati devono essere tali che anche avendone a disposizione solo un sottoinsieme, sia ancora possibile decodificare l'informazione originale, seppure con parametri di qualità ridotti (ad es., per il video, SNR ridotto, risoluzione ridotta, frame-rate ridotto). Inoltre, un tale sottoinsieme deve costituire in sé una rappresentazione *efficiente*[2] dell'informazione originale. Nonostante la codifica scalabile preveda sempre l'esistenza di strati, spesso si parla di codifica scalabile a strati quando il numero di strati è ridotto (fino a 3), mentre si parla di scalabilità a grana fine (Fine Grain Scalability) quando il numero di strati è elevato. In generale, esiste uno strato che può essere decodificato indipendentemente dagli altri, fornendo una prima approssimazione del segnale originale. Tale strato è detto *base*. Gli strati successivi permettono invece di arricchire la descrizione, fornendo i dettagli mancanti allo strato base. Tali strati sono detti quindi di *enhancement* (cioè arricchimento, miglioramento).

La codifica scalabile costituisce una possibile soluzione del problema della trasmissione su di un canale di cui non si conosce esattamente il tasso. Bisognerà in tal caso effettuare una codifica scalabile dell'informazione, utilizzando N layer. Il decodificatore ne riceverà soltanto una parte, a seconda dello stato corrente del canale, realizzando quindi un adattamento della qualità della sequenza decodificata alle caratteristiche attuali del canale.

Le prestazioni di un codificatore scalabile con due strati sono rappresentate dalla curva tratto-punto in figura 8. Tale curva si riferisce al caso in cui lo strato base corrisponda ad un bit-rate di 400 kbps, e quello di enhancement ad altri 400 kbps. Quando il tasso disponibile sul canale è compreso tra 400 e 800 kbps, il livello base è completamente ricevuto e decodificato. Quando invece il tasso disponibile è superiore a 800 kbps, anche il secondo strato viene ricevuto e decodificato.

Come si vede, la scalabilità a strati trasforma la curva a singolo gradino del caso non scalabile, in una a due gradini, che significa una maggiore capacità di adattamento alle caratteristiche del canale. Si potrebbe pensare allora di arrivare ad una piena scalabilità semplicemente aumentando il numero N di strati. Tuttavia, negli schemi ibridi, questo comporta un progressivo peggioramento delle prestazioni: al crescere di N, la curva corrispondente sale sì in modo più graduale, ma, per contro, essa si allontana sempre di più dalla curva limite. Per ottenere una piena scalabilità, dal punto di vista teorico la strada è quella della codifica video basata su trasformata 3D ma è possibile ottenere un buon livello di scalabilità anche con schemi ibridi. In particolare, si sta diffondendo sempre più lo standard H.264/SVC (scalable video coding), che è una variante scalabile di H.264 (sezione 8).

6.3 La scalabilità in MPEG-2

MPEG-2 fornisce tre tipi di scalabilità (vedere la sezione 6.3): in SNR, in risoluzione ed in frame-rate. In ogni caso, la scalabilità è ottenuta con uno strato base ed un secondo detto di miglioramento, o di *enhancement*. È prevista anche la possibilità di un ulteriore livello di miglioramento, per un totale di tre strati.

MPEG-2 può produrre un flusso codificato in modo scalabile, ma ciò ha un costo, in termini di aumento della complessità del codificatore ed in termini di riduzione delle prestazioni: tipicamente, il flusso a livello enhancement non arriva alla qualità massima che si potrebbe ottenere con codifica a singolo layer e con lo stesso bit-rate. Ciò è mostrato nella figura 8: il vertice del secondo "gradino" della curva verde non arriva alla stessa altezza della curva limite a parità di bit-rate. Ciò perché la scalabilità impone un vincolo abbastanza pesante sulla struttura del codificatore che influisce sulle prestazioni. Questo è vero in generale per tutti i codificatori video di tipo ibrido.

Vediamo in particolare come è ottenuta la scalabilità in MPEG-2.

[2] In termini di prestazioni tasso-distorsione

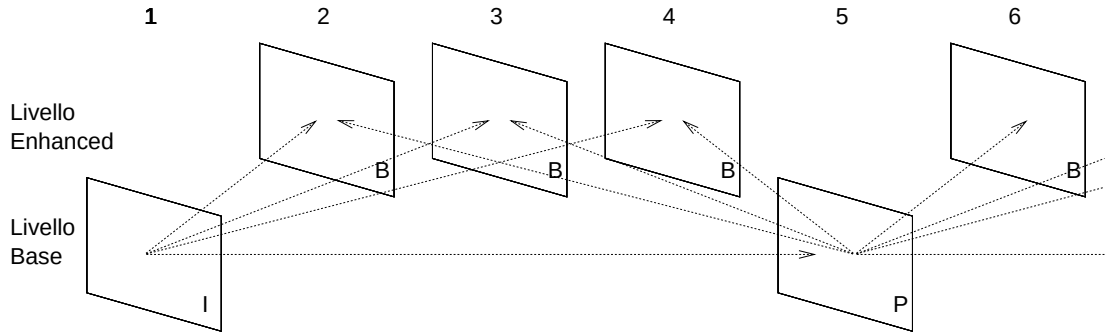


Figura 9: Scalabilità temporale. Le frecce indicano la dipendenza della compensazione del movimento. Il livello base (frame I e P) può essere decodificato indipendentemente dal livello enhanced.

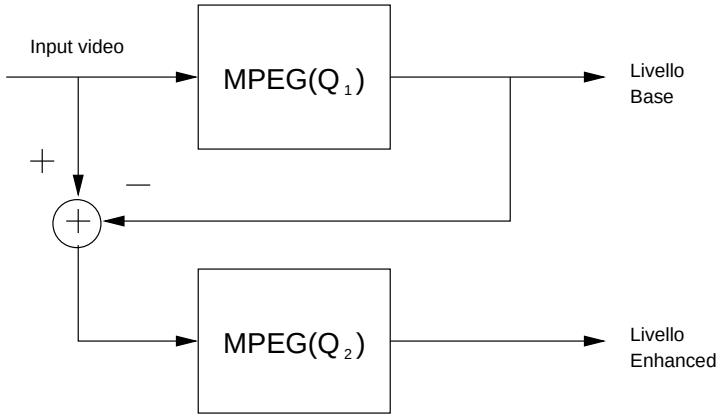


Figura 10: Scalabilità in qualità

La scalabilità nel tempo (cioè in frame-rate) è ottenuta inserendo le frame I e P nello strato base e le B nello strato enhancement. In questo modo il livello base può essere decodificato autonomamente, e fornisce una versione della sequenza video a frame-rate ridotto. Infatti la compensazione del movimento delle frame P non utilizza frame B. Il livello di enhanced da solo è inutile (come accade sempre nella codifica scalabile), mentre, insieme al livello base, permette di arrivare al pieno frame-rate.

Con una struttura del GOP come quella di figura 9, è possibile ottenere un flusso base a frame-rate pari a $1/4$ dell'originale, ed un livello di enhancement che aggiunge tutte le frame mancanti. A seconda del frame-rate desiderato, l'utilizzatore può quindi scegliere di ricevere solo il flusso base o il flusso base più quello di enhancement.

Per quanto riguarda la scalabilità in qualità (o in SNR), il livello base corrisponde concettualmente ad un codificatore MPEG con quantizzazione piuttosto grossolana, evidenziata dal passo di quantizzazione Q_1 in figura 10. Tale livello base è utilizzato come predizione della sequenza a qualità migliore: nel livello enhancement quindi si codifica l'errore di predizione, ossia la differenza tra il livello base e la sequenza originale, ancora con uno schema concettualmente equivalente a MPEG. Tuttavia questa volta la quantizzazione è realizzata con un passo di quantizzazione più piccolo, indicato con Q_2. Si noti che una tale struttura può essere iterata per un numero arbitrario di livelli di qualità. È opportuno comunque evidenziare che ogni nuovo livello di scalabilità richiede in pratica un nuovo coder MPEG, e che al crescere del numero di livelli, le prestazioni globali - in termini di curva tasso-qualità - si allontanano sempre di più da quelle ideali, rappresentate dalla curva a tratto continuo in figura 8.

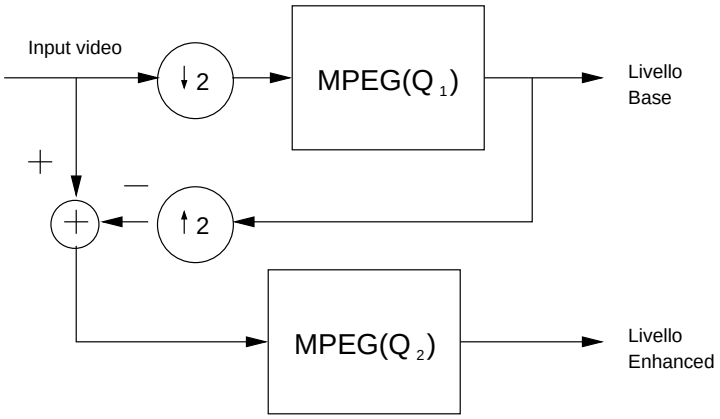


Figura 11: Scalabilità spaziale

La scalabilità in risoluzione (detta anche scalabilità spaziale) si ottiene in modo simile. Prima si codifica una versione della sequenza a risoluzione ridotta (ottenuta sottocampionando ed eventualmente filtrando la sequenza originale) con uno schema tipo MPEG. Il livello base, decodificato ed opportunamente interpolato, costituisce la predizione della versione a piena risoluzione: nel livello enhanced si codifica l'errore di predizione come illustrato in figura 11.

Si noti come anche la scalabilità spaziale fornita da MPEG-2 ha un notevole costo in termini di complessità (bisogna duplicare la struttura del codificatore per ogni livello di scalabilità). Inoltre il codificatore ibrido è penalizzato dagli anelli di predizione, il che è la causa della riduzione di prestazioni. Infine, MPEG-2 non fornisce la scalabilità a grana fine.

7 Note su MPEG-4

Lo standard MPEG-4 fornisce un insieme di tecnologie orientate a soddisfare le esigenze di chi produce (autori) il segnale video, chi lo distribuisce (provider di servizi di rete) e chi ne usufruisce (utenti).

- Per gli autori, MPEG-4 rende possibile la produzione di contenuti con una elevata riusabilità, maggiore flessibilità, grafici animati, interazione con pagine WWW. Inoltre esso permette una migliore gestione dei diritti d'autore.
- Per chi fornisce servizi di rete, MPEG-4 offre un incapsulamento tale che le informazioni di trasporto possano essere interpretate e tradotte in appropriati messaggi di segnalazione su ogni diversa rete. Sono forniti anche dei descrittori di qualità di servizio (QoS), senza però la pretesa di adattare tali descrittori ad ogni possibile definizione di QoS su reti diverse.
- Per gli utenti, MPEG-4 porta un elevato livello d'interazione con il contenuto. Inoltre consente di fruire di contenuti multimediali anche su reti a basso bit-rate, grazie all'utilizzo di tecniche di codifica ad elevata efficienza.

Questi obiettivi vengono raggiunti grazie a:

1. un modo standardizzato di descrivere i "media objects" ossia unità di contenuto audio e/o video. Questi media objects possono avere origine naturale (registrati con telecamera o microfono) o sintetica (generati al calcolatore);
2. un modo standardizzato di descrivere come i media objects compongono una scena;

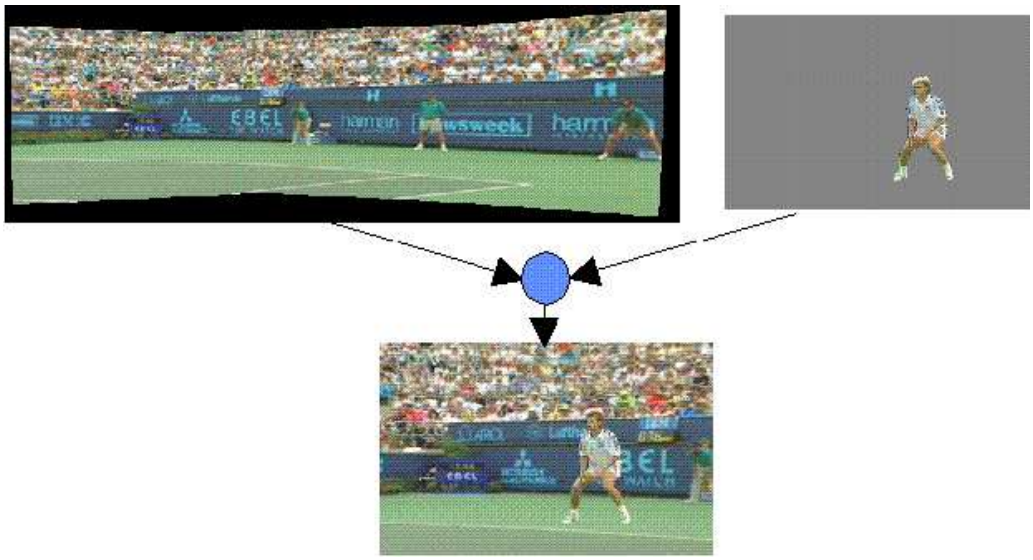


Figura 12: La codifica a oggetti in MPEG-4, con l'uso di sprite per lo sfondo

3. un modo standardizzato di comporre e sincronizzare i dati (multiplexing) associati ai media objects, in modo che possano essere trasmessi su canali di rete;

4. un modo standardizzato di consentire l'interazione dell'utente con la scena audio/video generata.

Le scene audio/video di MPEG-4 sono composte da diversi media objects, organizzati in modo gerarchico. Al livello più basso della gerarchia ci sono i media objects primitivi:

- immagini fisse (ad esempio, sfondi);
- oggetti video (persone o oggetti che si muovono, senza lo sfondo);
- oggetti audio (dialoghi, musica);

Inoltre MPEG-4 definisce la rappresentazione di testo, grafici, "talking heads" sintetiche, suoni sintetici.

L'approccio a oggetti consente di codificare nel modo più efficiente ognuno di essi, utilizzando le tecniche di volta in volta più adeguate. Lo schema di base del codificatore MPEG-4 è ancora quello ibrido, ma ogni oggetto viene codificato indipendentemente. Tra le tecniche utilizzate a questo scopo, ricordiamo la trasformata wavelet per oggetti di forma arbitraria (completata con un opportuna versione dell'algoritmo di codifica basato sugli Zero-Tree); segnaliamo anche modelli di movimento basati su mesh deformabili, capaci di descrivere efficacemente movimenti complessi di oggetti di forma arbitraria. Queste tecniche sono integrate da un approccio che consente di codificare indipendentemente lo sfondo (sprite) rispetto agli oggetti (vedere figura 12), e formare poi una scena utilizzando i vari media objects. Questa flessibilità da luogo anche ad una cosiddetta scalabilità a oggetti, che consiste nella possibilità, da parte dell'utente, di scegliere quali oggetti vadano effettivamente a costituire la scena riprodotta.

8 Note su H.264

Lo standard video MPEG-2 è stata la tecnologia che ha reso possibile i sistemi di televisione digitale, dalla televisione via cavo, alle trasmissioni satellitari, fino al recente digitale terrestre. Tuttavia, altri sistemi di

comunicazione, come l'ADSL o l'UMTS hanno tassi di trasmissione disponibili molto più bassi, il che costituisce un limite per il numero di canali oppure per la qualità ricevuta. Il nuovo standard video H.264 è stato sviluppato con l'obiettivo di migliorare il più possibile l'efficienza di codifica degli standard esistenti. Il progetto H.264 è stato portato avanti congiuntamente dall'ISO, che ha creato la famiglia MPEG e dall'ITU, che ha creato gli standard H.26x. Questo si riflette nella varietà di nomi con cui è noto: oltre ad H.264, si parla di "H.26L", "codec JVT" (Joint Video Team), "ISO/IEC 14496-10", "MPEG-4 Parte 10", e "MPEG-4 AVC" (Advanced Video Coding).

Questo standard si compone di due parti: lo strato di codifica in senso stretto (Video Coding Layer – VCL), e lo strato di astrazione di rete (Network Abstraction Layer – NAL). Quest'ultimo si occupa di dare un formato opportuno alle informazioni di codifica prodotte dal VCL, in modo da adattarsi al trasporto su rete o alla memorizzazione su dispositivi ottici e magnetici. In questi appunti ci occuperemo solo di alcuni aspetti del VCL.

Lo schema di codifica di H.264 è quello ibrido, comune a tutti gli standard video: le frame "intra" sono codificate indipendentemente dalle altre, con un opportuna tecnica che utilizza, oltre alla trasformata presente anche negli standard precedenti, anche la predizione *spaziale*; le frame "inter" (o predittive) sono invece predette da un insieme di frame già codificate, e l'errore di predizione è codificato con un'opportuna tecnica basata su trasformata. Ci sono poi delle frame B generalizzate. Ogni elemento di questo schema è stato ottimizzato in modo da raggiungere le migliori prestazioni possibile, e non è possibile individuarne uno che contribuisce in maniera determinante all'incremento complessivo di efficienza, che per altro è notevole: si arriva fino ad oltre il 70% di riduzione del bit-rate a parità di qualità rispetto a MPEG-2. Si può dire piuttosto che tale miglioramento è possibile proprio grazie alla somma di tanti piccoli contributi. Vedremo ora gli aspetti più significativi dell'algoritmo di codifica H.264.

Le slice. Il concetto di frame è generalizzato in quello di "slice". Una slice è un insieme di macroblocchi in una frame, e quindi può coincidere con la frame stessa. Ciò rende notevolmente più flessibile la struttura del flusso codificato. H.264 parla di *slice* di tipo I, P e B, piuttosto che di frame. Tuttavia, per semplicità, in questi appunti si farà riferimento solo al caso (di gran lunga più comune) in cui una slice coincide con una frame.

Predizione spaziale. Negli standard di codifica precedenti, non è prevista la predizione spaziale, cioè la predizione del blocco corrente a partire da quelli adiacenti, perché si lascia che sia la trasformata DCT a eliminare la ridondanza spaziale. In H.264 si cerca invece di sfruttare più direttamente tale ridondanza. Esistono due modalità di predizione spaziale: la prima utilizza blocchi piccoli (4×4 pixel) ed è adatta per zone con dettagli; la seconda usa blocchi più grandi (16×16 pixel) e va bene per regioni omogenee.

Compensazione del movimento. La compensazione del movimento è ottimizzata sotto diversi punti di vista. Per prima cosa si usano blocchi di dimensione variabile: possiamo scegliere blocchi grandi quando ci sono oggetti grandi che si muovono uniformemente, o blocchi per oggetti piccoli o per modellare movimenti più complessi. In secondo luogo la predizione del blocco corrente può essere realizzata usando fotogrammi anche molto lontani temporalmente dalla frame corrente. Si hanno poi le frame B generalizzate, in cui la compensazione bidirezionale viene effettuata con una combinazione lineare a coefficienti arbitrari di blocchi provenienti dalle due frame di riferimento (invece in MPEG si usa solo la media tra i due blocchi). Infine si usa una stima del movimento con precisione fino al quarto di pixel; sono realizzate due interpolazioni per ottenere un fotogramma a risoluzione 4 volte maggiore dell'originale; per la prima interpolazione si usa un filtro di lunghezza 6 (quindi più complesso dell'interpolazione bilineare di MPEG); per la seconda un filtro bilineare.

Trasformata spaziale. Si usa una trasformata a coefficienti interi e con blocchi 4×4 invece che 8×8. Si può usare un supporto piccolo (rispetto a MPEG) perché la maggiore efficienza della predizione, tanto spaziale quanto temporale, rende meno correlato l'errore di predizione. Inoltre in questo modo si riduce la complessità della trasformata. La particolare trasformata a coefficienti interi approssima abbastanza

bene la DCT, e non ne ha il principale difetto, cioè il disallineamento tra DCT e IDCT dovuto alla precisione finita; in altre parole, i coefficienti di DCT e IDCT sono numeri reali: rappresentandoli - come inevitabile - con una precisione finita, si perde la perfetta invertibilità della trasformata. Questo problema è ovviamente superato se i coefficienti della matrice di trasformazione sono interi.

Deblocking filter. Gli algoritmi ibridi operano indipendentemente sui blocchi in cui è suddivisa l'immagine. Ciò significa che niente assicura la continuità tra blocchi adiacenti. Ciò è particolarmente visibile nella codifica a basso tasso con MPEG (si parla di "blocking artefact"). Per ridurre questo effetto, in H.264 si inserisce un filtro di deblocking nell'anello di predizione, in modo da migliorare la continuità tra blocchi adiacenti. Il filtro di deblocking consente, a parità di qualità, una riduzione del bit-rate del 10% circa.

Codifica entropica. In H.264 sono previste due tecniche di codifica entropica dei simboli di codifica. La tecnica CAVLC (Context Adaptive Variable Length Coding) è una variante delle tecniche a lunghezza variabile usate in MPEG, ed è in grado di adattare la scelta delle codeword alle caratteristiche del video che si sta considerando. La seconda tecnica, più avanzata, è chiamata CABAC (Context Adaptive Binary Adaptive Coding) e si basa sulla codifica aritmetica; anche in questo caso tale codifica può essere tarata sulle caratteristiche del segnale da codificare. CABAC consente una riduzione del bit-rate del 10%–15% rispetto a CAVLC.

9 Per approfondire...

Per approfondimenti sulla stima del movimento a precisione frazionaria (sezione 5.1), riferirsi a [9]. Per la scalabilità (sezione 6), riferirsi a [10] ed a [11]. Per approfondimenti su MPEG-4 (sezione 7), vedere [12, 13]. Inoltre esiste un buon riferimento *on-line* su:
`http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm`

Per maggiori informazioni su H.264, (sezione 8), riferirsi a [14]. Molti sono i possibili riferimenti sulla codifica video con trasformata tridimensionale, tra i quali citiamo [15, 16, 17, 18, 19].

Riferimenti bibliografici

[1] *Video codec for audiovisual services at $p \times 64$ kbits/s*, International Telecommunication Union - Telecommunication Standardization Sector, Aug. 1990, ITU-T Recommendation H.261.

[2] *Video coding for low bit rate communication*, International Telecommunication Union - Telecommunication Standardization Sector, Mar. 1996, ITU-T Recommendation H.263.

[3] *Advanced video coding for generic audiovisual services*, iTU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 1: May 2003, Version 8: Consented in July 2007.

[4] *Coding of Moving Pictures and Associated Audio for Digital Sotrage Media at up to about 1.5 Mbit/s*, ISO/IEC JTC1, 1993, ISO/IEC 11172-2.

[5] *Generic Coding of Moving Pictures*, ISO/IEC JTC1, 1995, ISO/IEC 13818-2.

[6] *Coding of audio-visual objects—Part 2: Visual*, iSO/IEC 14496-2 (MPEG-4 Visual), ISO/IEC JTC 1, Version 1: Apr. 1999, Version 3: May 2004.

[7] M. Miyahara, K. Kotani, and V. Algazi, "Objective picture quality scale (PQS) for image coding," *IEEE Transactions on Communications*, vol. 46, no. 9, pp. 1215–1226, Sep 1998.

[8] Z. Wand, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 3, pp. 600–612, Apr. 2004.

[9] A. Bovik, Ed., *Handbook of image and video compression*. Academic Press, 2000.

[10] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317, Mar. 2001.

[11] J.-R. Ohm, "Advances in scalable video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 42–56, Jan. 2005.

[12] T. Sikora, "MPEG digital video-coding standard," *IEEE Signal Processing Magazine*, Sep. 1997.

[13] ——, "The MPEG-4 video standard verification model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 19–31, Feb. 1997.

[14] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[15] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1530–1542, Dec. 2003.

[16] ——, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in *Proceedings of IEEE Internantional Conference on Image Processing*, Thessaloniki, Greece, Oct. 2001, pp. 1029–1032.

[17] J.-R. Ohm, "Three dimensional subband coding with motion compensation," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 559–571, Sep. 1994.

[18] S. J. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 155–167, Feb. 1999.

[19] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001, pp. 1793–1796.