

Capitolo 5

Il controllo del bit rate

5.1 Introduzione

Nel capitolo 3 dello standard MPEG-2 (*ISO/IEC 13818-2*) il bit rate viene definito come la velocità con la quale i bit che formano la sequenza codificata arrivano al decoder. L'esigenza di considerare la questione del bit rate nasce dai limiti del mezzo trasmissivo e dalla necessità di assicurare la decodifica e la visualizzazione delle immagini con la cadenza originale con cui sono state acquisite (frame rate).

Il problema del bit rate presenta due aspetti principali: da un lato esistono le regole sintattiche stabilite nello standard per descrivere il bit rate all'interno del bitstream affinché quest'ultimo possa essere correttamente decodificato; dall'altro ci sono le tecniche per assicurare che durante la codifica di una sequenza video si rispetti il vincolo indicato dal bit rate, mantenendo la massima qualità possibile. Bisogna osservare come questo secondo aspetto non è trattato dallo standard MPEG che, come già detto precedentemente, si limita a definire la sintassi e la decodifica di un bitstream conforme allo standard.

Il controllo del bit rate è decisivo per assicurare un'alta qualità della codifica in quanto permette di mettere a disposizione più risorse per le parti di una sequenza più critiche da comprimere. Un controllo che non reagisce tempestivamente alle variazioni di complessità delle immagini tende a riportare il bit rate nella norma comprimendo eccessivamente e facendo così scendere bruscamente la qualità.

5.2 Constant Bit Rate e Variable Bit Rate

Il controllo del bit rate può avvenire secondo due modalità diverse: *Constant Bit Rate* e *Variable Bit Rate*. Con la codifica Constant Bit Rate (CBR) il numero di bit trasmessi per unità di tempo su un canale è costante. Poiché il contenuto di informazione varia da un'immagine all'altra è necessario cambiare continuamente il grado di compressione. Di conseguenza la qualità della sequenza codificata non rimane costante nel tempo e questo può essere un fattore di disturbo per l'osservatore.

Al contrario, con la codifica Variable Bit Rate (VBR) il numero di bit trasmessi sul canale varia nel tempo in funzione della complessità delle immagini garantendo così che la qualità

percepita sia pressoché costante. Per consentire la decodifica lo standard MPEG impone comunque determinati vincoli a questa variabilità del bit rate.

La codifica CBR è necessaria quando si trasmette su un canale in cui la banda allocata è fissa (ad esempio TV via satellite e via cavo, canali CDN, collegamenti ATM in modalità CBR) oppure quando si legge il bitstream da una memoria di massa lenta (ad esempio il CD-ROM). La codifica VBR si usa quando si legge da una memoria di massa veloce (ad esempio hard disk), con cui bisogna rispettare un bit rate di picco corrispondente al massimo transfer rate del dispositivo e un bit rate medio determinato dalla capacità di memorizzazione: in quest'ultimo caso la sequenza codificata ha una qualità media maggiore. Questa tecnica è adottata ad esempio nei supporti DVD che contengono dei filmati in formato MPEG-2.

In Figura 5.1 si mostra un esempio di codifica VBR ottenuta cercando di mantenere costante la qualità intorno ai 29,2dB (misurata attraverso il Peak SNR calcolato sulle matrici di luminanza dopo la decodifica), mentre non si sono posti vincoli sul bit rate che, come si può vedere, varia molto da una picture all'altra.

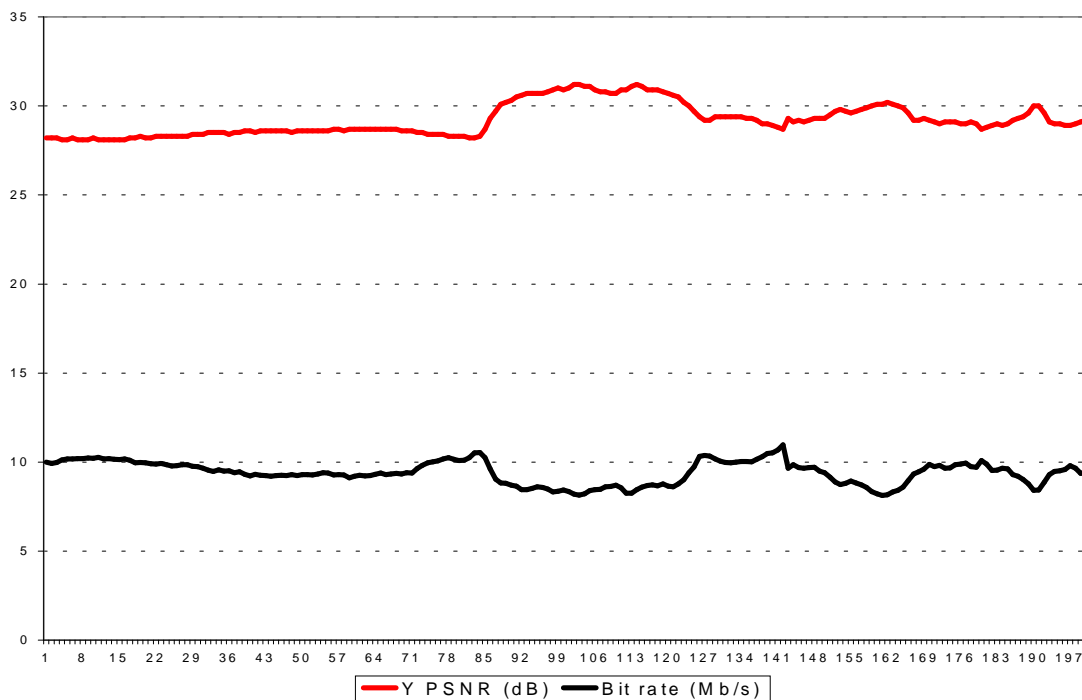


Figura 5.1 Esempio di codifica VBR

Questo esperimento permette di fare un'altra importante osservazione: i massimi locali della curva del PSNR sono in corrispondenza dei minimi sulla curva del bit rate. Ciò permette di verificare a posteriori che MPEG effettua una compressione *lossy* e che la qualità della codifica dipende dalla quantità di bit utilizzati.

Se si codifica la stessa sequenza utilizzando un algoritmo di controllo del bit rate con modalità CBR, impostando come bit rate quello medio di Figura 5.1, che è 9.417 Mbps, si ottiene l'andamento della qualità mostrato in Figura 5.2. Si può notare che il PSNR varia molto di più rispetto a prima e che il suo valore medio è leggermente più basso (28.9 dB invece di 29.2 dB): questo dimostra come la modalità VBR permette di ottenere una qualità di codifica maggiore.

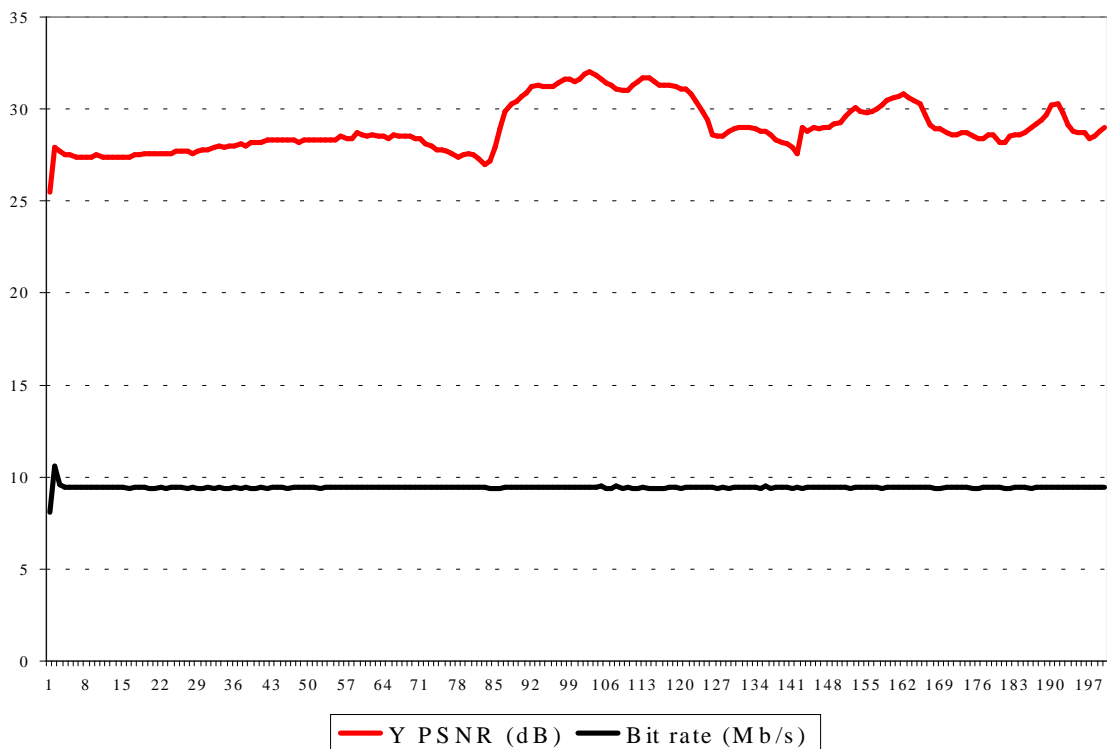


Figura 5.2 Esempio di codifica CBR

5.3 Il Video Buffer Verifier e la sintassi per il bit rate

Secondo lo standard MPEG-2 (annex C) un bitstream valido deve poter essere decodificato da un ipotetico sistema composto da un buffer e da un decoder chiamato Video Buffer Verifier (VBV) come mostrato in Figura 5.3.

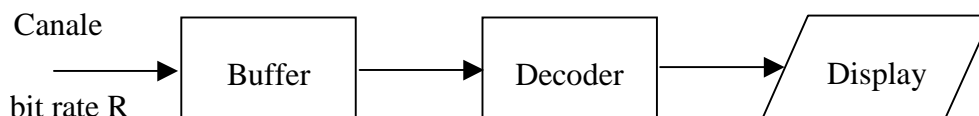


Figura 5.3 Diagramma a blocchi del Video Buffer Verifier

La dimensione del buffer (B_{VBV}) è specificata dall'encoder nel bitstream nel *sequence header* e nel *sequence extension header*: il decoder usa questa informazione per verificare se il proprio buffer ha dimensione sufficiente per contenere i dati in arrivo (in realtà tutto ciò ha senso unicamente nel caso di «streaming» da un canale trasmissivo mentre non si applica nel caso di lettura da una memoria di massa).

Nel *sequence header* viene anche specificato su 18 bit un valore di bit rate (in unità di 400 bit/s), che nel caso di MPEG-2 VBR indica il bit rate di picco, mentre in MPEG-1 VBR deve essere una sequenza di 1 (cioè 3FFFF in esadecimale).

Un altro parametro caratterizzante la gestione del bit rate è *vbv_delay* codificato nel *picture header* come intero senza segno su 16 bit; esso indica al VBV il numero di periodi di un clock a 90 kHz che bisogna attendere dopo aver ricevuto l'ultimo byte del *picture_start_code* prima di decodificare la picture. In modalità CBR *vbv_delay* per la picture n-esima si può calcolare con la formula seguente:

$$vbv_delay(n) = \frac{90000 * S(n)}{R}$$

dove $S(n)$ è l'occupazione in bit del VBV buffer immediatamente prima di rimuovere la picture n-esima ma dopo aver rimosso tutti gli header, gli *user data* e byte di riempimento che la precedono; mentre R è il bit rate in bps.

La Figura 5.4 mostra graficamente il legame tra *vbv_delay* e l'occupazione del VBV buffer. Se il bitstream è VBR, *vbv_delay* assume il valore esadecimale FFFF per tutte le picture.

Infine esiste un parametro chiamato *low_delay* che in questa implementazione vale sempre 0 indicando che il VBV buffer non deve mai subire underflow.

I dati possono essere trasferiti al VBV buffer, inizialmente vuoto, con velocità costante oppure variabile (CBR o VBR); in entrambe le modalità il numero di bit prodotti per ciascuna picture deve soddisfare alcuni vincoli imposti dallo standard. Nel momento in cui il VBV buffer viene esaminato dal decoder e prima di rimuovere i dati relativi alla picture, il numero di bit nel buffer deve essere compreso tra zero e la dimensione B_{VBV} impostata nel bitstream. Va notato che per “dati relativi alla picture” si intendono tutti i bit della picture codificata, tutti gli header e “user data” immediatamente precedenti (compresi eventuali byte di riempimento) e tutti i byte di riempimento (*stuffing bytes*) che la seguono fino al successivo start code escluso (ad eccezione del caso in cui tale codice è quello di fine sequenza). I dati vengono rimossi dal VBV buffer in maniera FIFO e la picture n-esima

da decodificare deve essere presente completamente al tempo $t(n)$ altrimenti il VBV buffer subirebbe underflow e ciò non è permesso perché `low_delay` è messo a zero.

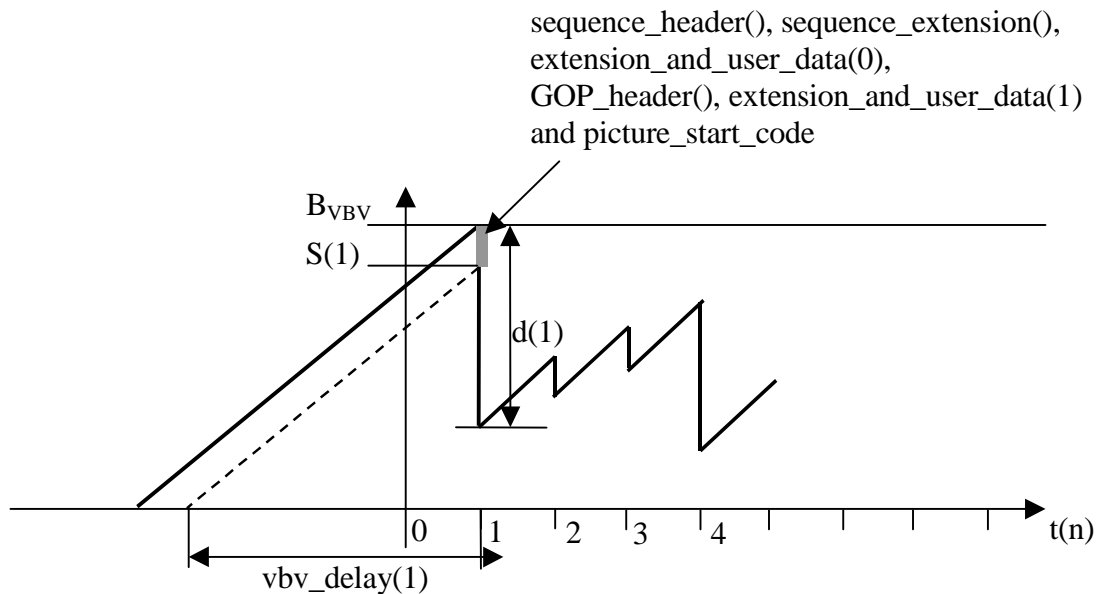


Figura 5.4 Occupazione del VBV buffer in modalità CBR

La definizione del *decoding time* $t(n)$ dipende dal valore dei parametri `progressive_sequence`, `repeat_first_field` e `top_field_first`. Se `progressive_sequence=1`, l'intervallo $[t(n+1);t(n)]$ è multiplo di T , dove T è l'inverso del frame rate, secondo la Tabella 5.1.

<code>repeat_first_field</code>	<code>top_field_first</code>	$[t(n+1);t(n)]$
0	X	T
1	0	$2 * T$
1	1	$3 * T$

Tabella 5.1 Valore dell'intervallo $[t(n+1);t(n)]$ con `progressive_sequence=1`

All'inizio della sequenza $[t(n+1);t(n)]$ è arbitrario e lo stato del VBV buffer deve essere tale da evitare situazioni di overflow o underflow.

Se `progressive_sequence=0` l'intervallo $[t(n+1);t(n)]$ è multiplo di T , dove T è l'inverso di 2 volte il frame rate, secondo la Tabella 5.2.

picture structure	repeat_first_field	Top_field_first	[t(n+1);t(n)]
frame	0	X	2*T
frame	1	X	3*T
Field	X	X	T

Tabella 5.2 Valore dell'intervallo [t(n+1);t(n)] con progressive _sequence=0

Se la picture n-esima è il *first field* di una I picture o P picture con *field structure* allora [t(n+1);t(n)] vale T.

Se la picture n-esima è il *second field* di una I picture o P picture con *field structure* e se *repeat_first_field*=0 allora [t(n+1);t(n)] vale [2*T;T].

Se la picture n-esima è il *second field* di una I picture o P picture con *field structure* e se la precedente I picture o P picture codificata usa una *frame structure* con *repeat_first_field*=1 allora [t(n+1);t(n)] vale [3*T - T].

All'inizio della sequenza $t(n+1) - t(n)$ è arbitrario e lo stato del VBV buffer deve essere tale da evitare situazioni di overflow o underflow.

Nel caso di bitstream CBR il bit rate R deve soddisfare la relazione espressa dalla formula seguente:

$$R(n) = \frac{d(n)}{\tau(n) - \tau(n+1) + t(n+1) - t(n)}$$

dove:

- ♦ $R(n)$ è la velocità in bps con cui i dati della n-esima picture entrano nel VBV
- ♦ $d(n)$ è il numero di bit dopo l'n-esimo *picture_start_code* fino alla fine del *picture_start_code* della picture n+1
- ♦ $\tau(n)$ è il ritardo codificato in *vbv_delay* per la picture n-esima misurato in secondi
- ♦ $t(n)$ è l'istante, misurato in secondi, in cui la picture n-esima è rimossa dal VBV buffer

In Figura 5.4 è mostrato graficamente il legame tra $d(n)$, *vbv_delay* e l'occupazione del VBV buffer.

Per i bit che precedono il primo *picture_start_code* e per quelli che seguono l'ultimo, $R(n)$ è uguale al valore impostato nel bitstream.

Dopo aver riempito il VBV buffer con tutti i dati che precedono il primo *picture_start_code* della sequenza (esso compreso), il buffer riceve dati dal bitstream per un tempo specificato da *vbv_delay* nel *picture header*: in quell'istante inizia la decodifica. Occorre notare che nella modalità CBR $R(n)$ deve rimanere costante nella sequenza con l'accuratezza permessa dalla quantizzazione di *vbv_delay*.

Nel caso di bitstream VBR i dati entrano nel VBV buffer con il bit rate specificato nel *sequence header* fino a riempirlo; quando esso è pieno non riceve più dati fino a quando

non viene esaminato per rimuovere e decodificare una picture. Quindi nella modalità VBR non ha senso il caso di overflow del buffer.

Poiché il buffer è riferito al decoder, una situazione di underflow corrisponde ad un bit rate troppo elevato mentre l'overflow si ha nel caso di bit rate troppo basso: per questo motivo il caso di overflow non ha senso nel VBR: infatti in questa modalità non ci sono limiti inferiori al valore istantaneo del bit rate.

5.4 Tecniche per il controllo del bit rate

Per controllare il bit rate è necessario poter regolare il numero di bit per picture prodotti in quanto bit rate e bit per picture sono legati dalla relazione seguente:

$$bps = \frac{bit}{picture} * framerate$$

Il modo più efficace per regolare la produzione dei bit è agire sul grado di quantizzazione dei coefficienti ottenuti dopo la DCT: infatti, maggiore è la quantizzazione, maggiore sarà il numero di coefficienti nulli e maggiore sarà l'efficienza della codifica run-length. Per contro aumentare la quantizzazione equivale a diminuire la precisione con cui l'informazione è trasmessa al decoder e quindi abbassare la qualità della sequenza codificata.

Modificare le matrici di quantizzazione comporta la necessità di doverle specificare nel bitstream e questo può compromettere l'eventuale guadagno in termini di compressione: quindi lo standard MPEG ha introdotto un unico fattore moltiplicativo chiamato `quantiser_scale` da indicare nell'header della slice e che può essere cambiato all'inizio di ciascun macroblocco. In realtà nel bitstream viene specificato un codice su 5 bit che può assumere i valori da 1 a 31 chiamato `quantiser_scale_code`. Questo codice viene usato per calcolare il valore di `quantiser_scale` in funzione di un altro parametro logico chiamato `q_scale_type` specificato nella picture extension: `q_scale_type=0` significa `quantiser_scale=2*quantiser_scale_code` mentre con `q_scale_type=1` la relazione tra `quantiser_scale_code` e `quantiser_scale` è quasi esponenziale. In questa implementazione dell'encoder si è deciso di tenere `q_scale_type=0`.

Il fattore `quantiser_scale` regola il grado di quantizzazione tramite le seguenti due formule, valide rispettivamente per la codifica intra e per quella non intra:

$$Q_{DCT}(i, j) = \frac{16 * DCT(i, j) + sign(DCT(i, j)) * quantizer_scale * IQ(i, j)}{2 * quantizer_scale * IQ(i, j)}$$

$$Q_{DCT}(i, j) = \frac{16 * DCT(i, j)}{2 * quantizer_scale * NQ(i, j)}$$

dove Q_{DCT} è la matrice dei coefficienti DCT dopo la quantizzazione, $DCT(i,j)$ i coefficienti ottenuti dalla trasformata DCT, $IQ(i,j)$ e $NQ(i,j)$ i coefficienti riportati nelle matrici di quantizzazione per le picture codificate con tecniche intra e non intra rispettivamente.

La quasi totalità delle tecniche di controllo del bit rate utilizza la variabile `quantiser_scale` per controllare i bit prodotti per ogni picture, cambiandola opportunamente per rispettare i limiti imposti al bit rate.

5.5 Il controllo del bit rate nel Test Model 5

L'encoder MPEG TM-5 controlla il bit rate della sequenza codificata variando la quantizzazione dei coefficienti DCT dei macroblocchi tramite il parametro `quantizer_scale`.

L'algoritmo lavora in 3 passi:

1. *valutazione della quantità di bit necessari*: in questa fase si stima il numero di bit disponibili per codificare la prossima picture;
2. *controllo del bit rate*: viene calcolato il valore del parametro di quantizzazione per ogni macroblocco, in funzione del grado di riempimento di un buffer virtuale e ipotizzando una distribuzione uniforme dei bit allocati sulla picture;
3. *quantizzazione adattativa*: il valore calcolato al punto precedente viene modulato in base all'attività spaziale del macroblocco ottenendo il parametro `quantizer_scale` definitivo.

5.5.1 Valutazione della quantità di bit necessari

Stima della complessità di una picture

Per ogni tipo di picture (I, P, B) il TM-5 stima di volta in volta un grado di complessità globale il cui valore iniziale è calcolato secondo le seguenti formule per i tre tipi di picture:

$$X_i = \frac{160 * bitrate}{115}$$

$$X_p = \frac{60 * bitrate}{115}$$

$$X_b = \frac{42 * bitrate}{115}$$

dove *bitrate* è misurato in bit al secondo.

Dopo la codifica di ciascuna picture i valori delle complessità sono aggiornati secondo le formule:

$$X_i = S_i Q_i$$

$$X_p = S_p Q_p$$

$$X_b = S_b Q_b$$

dove S_i, S_p, S_b contengono il numero di bit generati dalla codifica e Q_i, Q_p, Q_b il valore del parametro `quantizer_scale` mediato su tutti i macroblocchi della picture compresi quelli skipped.

Calcolo dei bit disponibili

Il numero dei bit disponibili per la codifica di una picture (*target bit*) viene calcolato prima della sua codifica, secondo le formule:

$$T_i = \max \left\{ \frac{R}{1 + \frac{N_p X_p}{X_i K_p} + \frac{N_b X_b}{X_i K_b}}, \frac{bitrate}{8 * framerate} \right\} \quad T_p = \max \left\{ \frac{R}{N_p + \frac{N_b X_b K_p}{X_p K_b}}, \frac{bitrate}{8 * framerate} \right\}$$

$$T_i = \max \left\{ \frac{R}{N_b + \frac{N_p X_p K_b}{X_b K_p}}, \frac{bitrate}{8 * framerate} \right\}$$

dove:

- ♦ K_p e K_b sono costanti dipendenti dalle matrici di quantizzazione; valgono rispettivamente 1 e 1.4 se si usano le matrici di quantizzazione standard;
- ♦ N_p e N_b sono il numero delle restanti picture di tipo P e B del GOP corrente;
- ♦ R (*remaining*) è indica il rimanente numero di bit che si possono usare per la codifica del GOP corrente; R viene aggiornato nel seguente modo:
 - $R=0$ all'inizio della sequenza
 - prima di codificare la prima picture di ciascun GOP R viene aggiornato secondo la formula

$$R = R_{prec} + \frac{bitrate * N}{framerate}$$

dove N è il numero totale di picture nel nuovo GOP.

- Dopo la codifica di ciascuna picture all'interno dello stesso GOP, R viene aggiornato con la formula

$$R = R - S$$

dove S può essere S_i , S_p , S_b a seconda del tipo di picture e rappresenta il numero di bit prodotti per la codifica della picture che si è appena codificata.

5.5.2 Controllo del bit rate

L'algoritmo di controllo del bit rate usato dal TM-5 è molto articolato. Esso si fonda su alcune ipotesi di base e su una serie di formule che lo implementano.

Il controllo del bit rate agisce prima della codifica vera e propria di un macroblocco generico all'interno di una picture. Esistono tre buffer virtuali, uno per ogni tipo di picture, d_i , d_p , d_b . L'algoritmo prevede un'allocazione uniforme dei bit all'interno di una picture, cioè ogni macroblocco dovrebbe essere codificato con un numero uguale di bit. Se, arrivati alla codifica di un determinato macroblocco, l'andamento reale del numero di bit spesi si discosta dall'andamento ideale (che dovrebbe essere lineare nell'ipotesi di allocazione uniforme), l'algoritmo decide se alzare o abbassare il `quantizer_scale` a seconda di tale scostamento. La Figura 5.5 descrive il comportamento dell'algoritmo: si nota come per uno scostamento positivo superiore ad una certa soglia, il `quantizer_scale` viene innalzato, sino a riportare nei limiti indicati il bit rate per quella specifica picture.

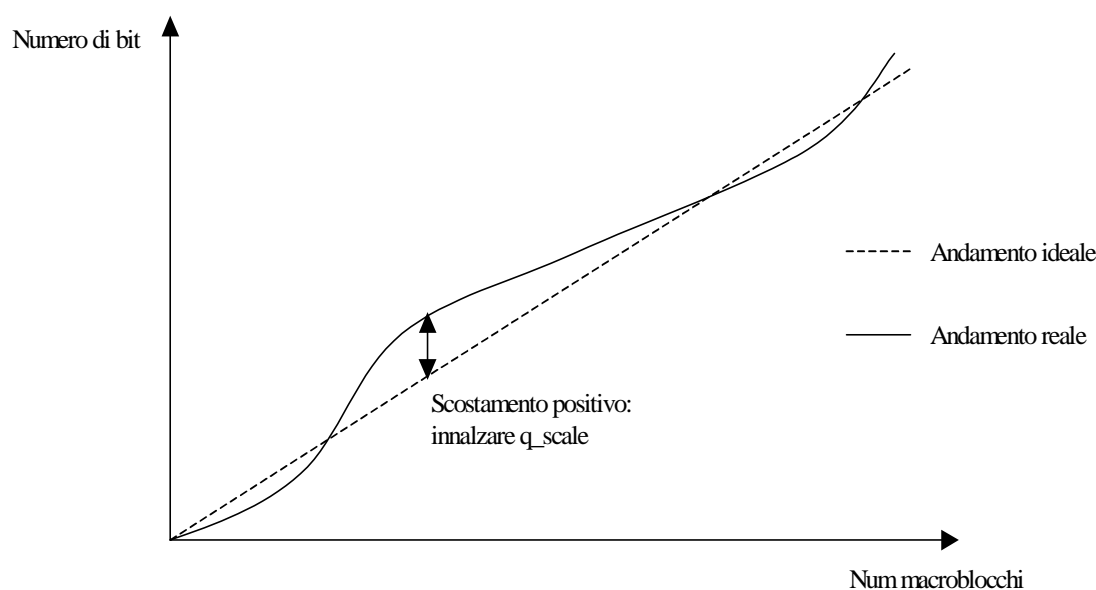


Figura 5.5 Distribuzione uniforme dei bit in una picture

Prima di codificare il macroblocco j -esimo l'algoritmo valuta il riempimento del buffer virtuale relativo al tipo di picture in questione, secondo la formula seguente (o le analoghe formule valide per picture di tipo B o P):

$$d'_j = d'_o + B_{j-1} - \left(\frac{T_l * (j-1)}{MB_count} \right)$$

dove:

- ♦ d'_o è il riempimento all'inizio della picture del buffer virtuale; tiene conto dei bit risparmiati o usati in più nella codifica della picture precedente dello stesso tipo
- ♦ B_j è il numero di bit prodotti nella codifica dei macroblocchi sino al j-esimo incluso
- ♦ MB_count è il numero di macroblocchi totali all'interno della picture
- ♦ d'_j misura il riempimento del buffer virtuale al macroblocco j-esimo

Alla fine della codifica di una picture, il riempimento del buffer virtuale (d'_j per $j=MB_count$) diventa il valore da usare come d'_o all'inizio della codifica della prossima picture dello stesso tipo.

Dopo aver valutato d'_j , l'algoritmo calcola il valore di `quantizer_scale` per il macroblocco j-esimo come secondo la relazione:

$$Q_j = \frac{d'_j * 31}{r}$$

dove r viene chiamato dal TM-5 parametro di reazione (*reaction parameter*) e viene calcolato secondo la formula:

$$r = 2 * \frac{bitrate}{framerate}$$

5.5.3 Quantizzazione adattativa

La fase di quantizzazione adattativa del TM-5 ha come obiettivo quello di legare il coefficiente di quantizzazione ad una misura della complessità, della criticità del macroblocco. Il TM-5 ha scelto come parametro di complessità un coefficiente chiamato attività spaziale (*spatial activity*). Dopo avere calcolato questo coefficiente lo si normalizza su tutta la picture e lo si usa per modulare il valore di `quantizer_scale` Q_j calcolato al passo precedente.

Il coefficiente di attività spaziale di macroblocco viene calcolato secondo la formula:

$$act_j = 1 + \min(vblk_1, vblk_2, vblk_3, vblk_4)$$

dove $vblk_n$ è relativo a ciascuno dei quattro blocchi di luminanza ed è calcolato secondo la formula:

$$vblk_n = \frac{1}{64} * \sum_{k=1}^{64} (P_k^n - P_mean_n)^2$$

$$P_mean_n = \frac{1}{64} * \sum_{k=1}^{64} P_k^n$$

dove P_k è il valore di luminanza originario del pel k-esimo nel blocco 8x8. Successivamente il valore normalizzato dell'attività viene calcolato, secondo la formula:

$$Nact_j = \frac{(2 * act_j) + avg_act}{act_j + (2 * avg_act)}$$

Infine l'algoritmo usa $Nact$ e Q_j per determinare il `quantizer_scale` secondo la relazione:

$$quantizer_scale = Q_j * Nact_j$$

5.6 Il nuovo algoritmo di controllo del bit rate

Il controllo del bit rate che è stato realizzato è ispirato a principi di semplicità e di robustezza. E esso rende la retroazione sul sistema molto stretta e realizza diversi tipi di retroazione, su più livelli.

L'algoritmo mantiene la notazione usata dal TM-5 e alcuni dei concetti già esposti.

Resta fermo il concetto di complessità della picture (X_i , X_p e X_b), così come rimangono valide le formule usate per calcolarne il valore, ma assume un'importanza maggiore, in quanto è tramite la X che si calcola il `quantizer_scale` per il macroblocco j-esimo. Rimane inalterato il concetto espresso dalla variabile R , che indica ancora il restante numero di bit utilizzabili per la codifica del GOP corrente. La variabile T (target bit) viene mantenuta con lo stesso significato e viene calcolata nello stesso modo. In aggiunta vengono calcolate la dimensione massima e minima che una picture codificata può assumere (*Upper Bound U* e *Lower Bound L*) per non causare situazioni di underflow o overflow nel video buffer verifier. La variabile T viene poi aggiornata per tenere conto di U secondo la formula:

$$T = \min(U, T)$$

Scompaiono invece i buffer virtuali, con tutto il meccanismo di calcolo dello scostamento per stabilire il coefficiente di quantizzazione.

Si fa invece uso del concetto di *target bit per il macroblocco* (t_j): similmente a quanto si fa con le picture si calcola un numero di bit da usare per codifica di un singolo macroblocco.

L'algoritmo di controllo del bit rate funziona secondo i seguenti passi:

- All'inizio di ogni GOP viene calcolato il valore di R, tenendo conto, sia in positivo che in negativo, di eventuali resti dalla codifica dei GOP precedenti, secondo una formula analoga a quella usata dal TM-5. Questo realizza un primo livello di retroazione.
- All'inizio di ogni picture viene calcolato il target bit per quella picture, tenendo conto anche delle limitazioni imposte dal VBV, e il `quantizer_scale` iniziale da usare con la formula

$$Q_scale_{iniz.} = \frac{X}{T}$$

Il calcolo dei target bit per una picture realizza un secondo livello di retroazione, in quanto tiene conto mediante X dei bit effettivamente usati per la codifica della picture precedente.

- Per ogni macroblocco vengono calcolati i target bit secondo la formula:

$$t_j = \frac{T-S}{MB_count-j+1}$$

Questo realizza un terzo livello di retroazione. Infatti se la codifica di un macroblocco occupa meno bit di quelli previsti dal suo t_j , questi bit vengono ridistribuiti su tutti i successivi macroblocchi della picture corrente.

Quindi viene calcolato il coefficiente di quantizzazione `q_scale` per quel macroblocco, secondo la relazione:

$$q_scale = \frac{X}{MB_count * t_j}$$

Dopo avere calcolato `quantizer_scale`, questo valore può venire innalzato in base a considerazioni percettive, legate al coefficiente di *smoothness* di quel macroblocco (si veda il paragrafo 5.8).

- Alla fine della codifica di una picture, il valore del coefficienti di complessità per quel tipo di picture viene aggiornato.

5.7 Prove sul nuovo algoritmo di controllo del bit rate

Per provare il nuovo algoritmo sono stati fatti due tipi di test. La prima prova è stata eseguita codificando una sequenza a differenti bitrate e controllando poi che la dimensione del bitstream prodotto fosse compatibile con i limiti posti. La seconda prova mette in evidenza l'andamento istantaneo del bit rate secondo per secondo all'interno di un bitstream codificato.

La Tabella 5.3 riporta i dati raccolti eseguendo la codifica di una sequenza con parametri di GOP $n=12$ e $M=13$. La prima colonna riporta i valori della dimensione teorica che il bitstream prodotto avrebbe dovuto avere, mentre le altre due le contengono le dimensioni ottenute usando come algoritmi di controllo del bit rate il TM-5 e il nuovo, implementato durante questo progetto. La Tabella 5.4 riporta i risultati ottenuti codificando la stessa sequenza con tutte picture di tipo Intra (GOP con $N=1$ e $M=1$)

Bit rate (kbps)	Dimensione teorica del bitstream (byte)	Dimensione del bitstream con TM-5	Dimensione del bitstream col nuovo algoritmo
480	1658880	1674187	1962173
1000	3456000	3486127	4085625
1500	5184000	5229044	6128120
1900	6566400	6623697	7762119
2500	8640000	8715411	10213120
3000	10368000	10458820	12255612
4000	13824000	17973389	16340611
5000	17280000	26669452	17318046
6000	20736000	26671139	17318061
7000	24192000	26671281	17318068
8000	27648000	21260749	17318077
9000	31104000	21709604	17318092
11000	38016000	19612344	17318172
14000	48384000	19360837	17318276

Tabella 5.3 Dimensioni del bitstream con GOP $N=12$, $M=3$

Bit rate (kbps)	Dimensione teorica del bitstream (byte)	Dimensione del bitstream con TM-5	Dimensione del bitstream col nuovo algoritmo
480	1658880	2301894	1962173
1000	3456000	3492230	4091291
1500	5184000	5235437	6133783
1900	6566400	6630139	7767733
2500	8640000	8722119	10218770
3000	10368000	10465485	12261248
4000	13824000	13952170	16345940
5000	17280000	31480004	20431082
6000	20736000	31480260	24515446
7000	24192000	18595149	26564243
8000	27648000	17932789	26564255
9000	31104000	18068119	26564213
11000	38016000	17876670	26564279
14000	48384000	18025665	26564320

Tabella 5.4 Dimensioni del bitstream con tutte I-picture

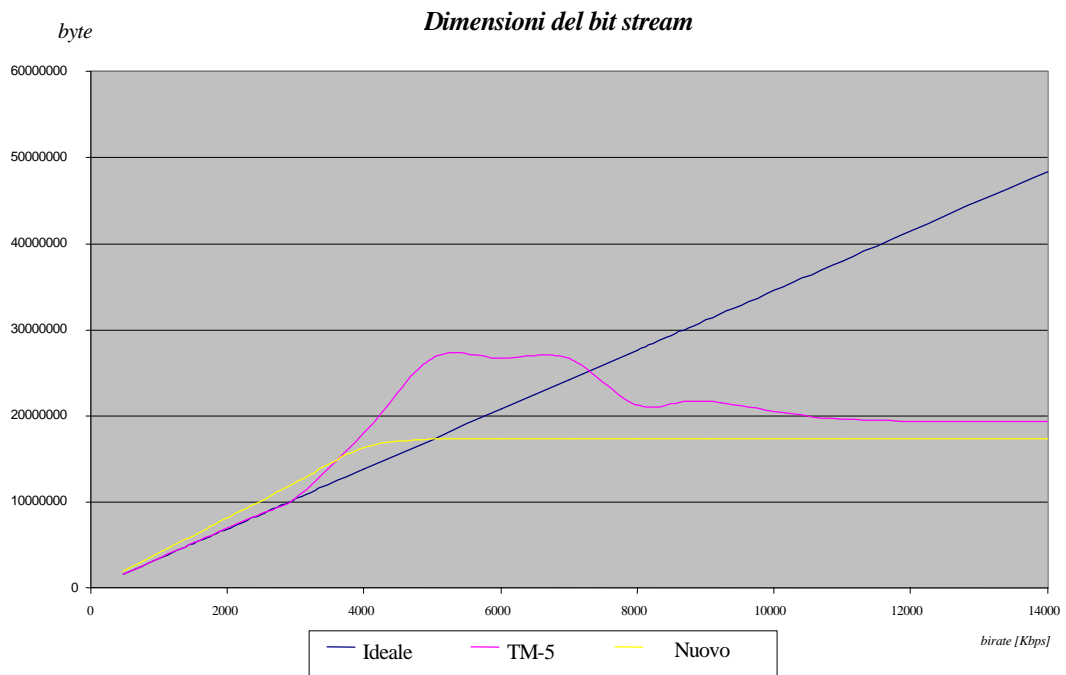


Figura 5.6 Grafico per la codifica con GOP N=12, M=3

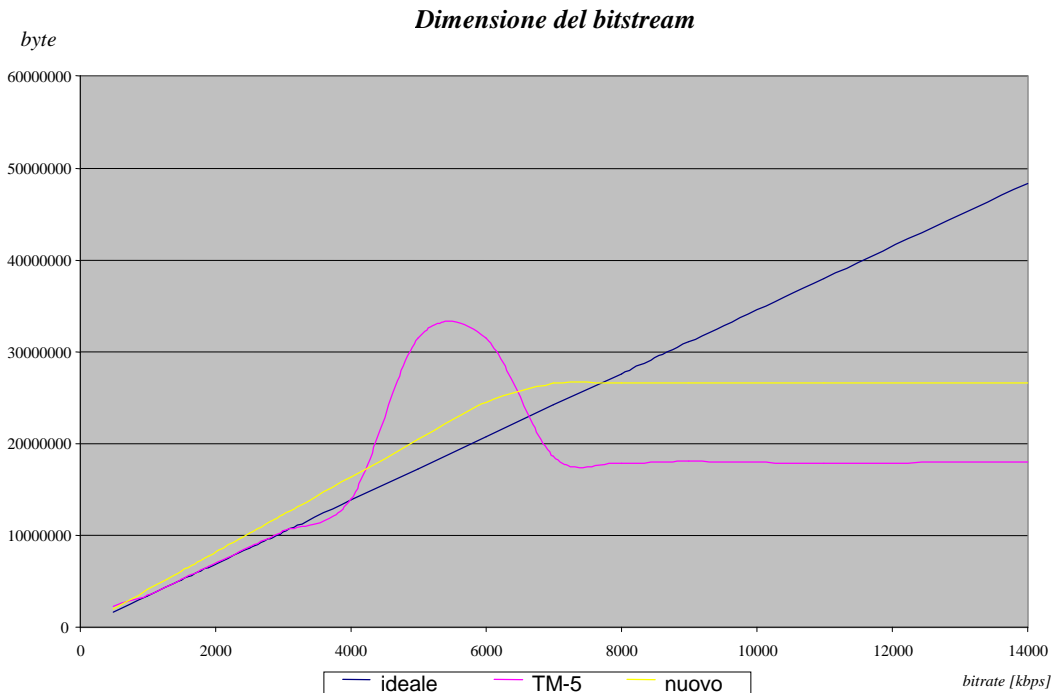


Figura 5.7 Grafico per la codifica con sole I-picture

La Figura 5.6 e la Figura 5.7 riportano i grafici dei risultati raccolti nelle precedenti tabelle. Si nota come il controllo del bit rate del TM-5 va fuori linearità in una zona di valori di bitrate compresa tra i 3 Mbps e i 7 Mbps. In quella zona il TM-5 non rispetta i vincoli. Invece il nuovo algoritmo realizzato mantiene sempre la linearità, pur con qualche scostamento dal comportamento ideale. A bit rate molto alti, la dimensione del bitstream si assesta su un valore asintotico, relativo al contenuto massimo di informazione delle immagini di input (entropia della sorgente).

La seconda prova che è stata effettuata ha come obiettivo misurare per ogni secondo di sequenza i bit prodotti con i due algoritmi. Infatti anche se globalmente un bitstream può rispettare i vincoli di bitrate, i bit prodotti secondo per secondo danno una misura più stringente sulla qualità dell'algoritmo. Le prove sono state effettuate codificando una sequenza ad un bit rate desiderato di 1,8 Mbps.

La Tabella 5.5 e il relativo grafico di Figura 5.8 mostrano i risultati ottenuti. Anche in questo caso si nota che il TM-5 si scosta in maniera molto elevata dal comportamento ideale e produce un bitstream che ha un picco di 9Mbps al quinto secondo. Nei secondi di sequenza successivi, per rispettare globalmente il bit rate, deve tenere un bit rate molto basso, degradando notevolmente la qualità della sequenza. Il nuovo algoritmo invece si mantiene sempre molto vicino al limite di bit per secondo desiderato, mantenendo la qualità della codifica costante su tutta la sequenza.

Istante	Bit rate istantaneo TM5 (bps)	Bit rate istantaneo con il nuovo algoritmo (bps)
1	1414404	1148861
2	1831525	1604810
3	1888695	1648825
4	4447922	2630786
5	7930758	1937873
6	2064646	1820160
7	467900	1881954
8	428116	1696476
9	554045	1878304
10	601812	1827419

Tabella 5.5 Bit prodotti per ogni secondo di sequenza con una codifica a 1,8 Mbps

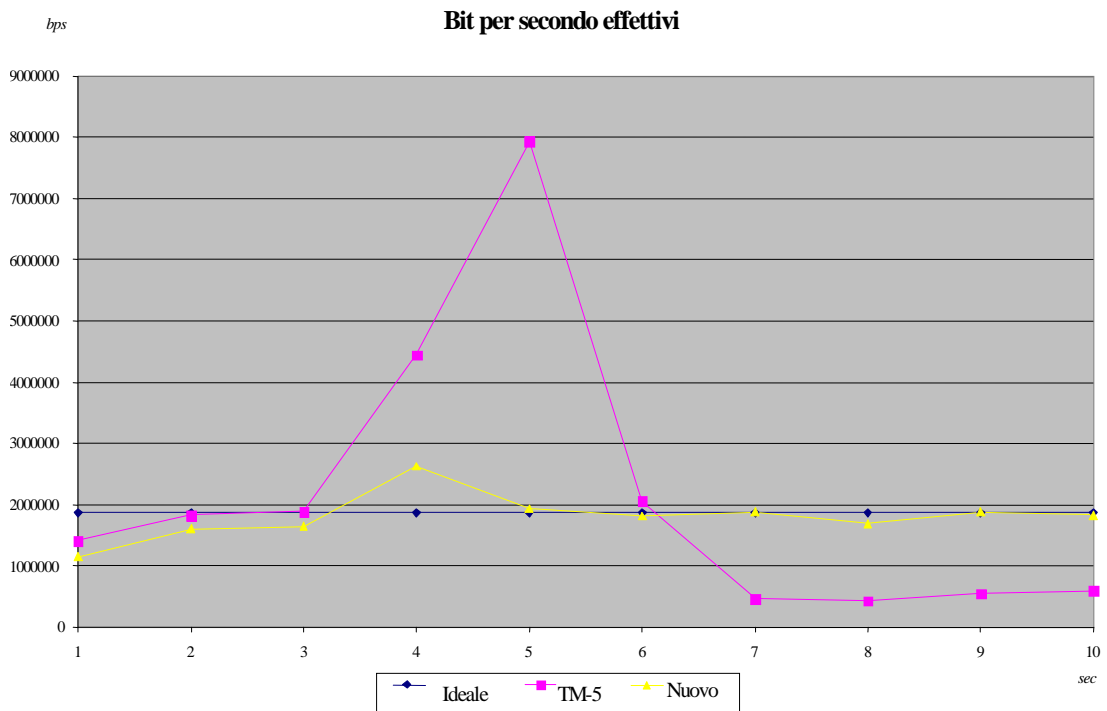


Figura 5.8 Variazione istantanea del bit rate con i vari algoritmi di controllo

5.8 Quantizzazione delle aree molto uniformi

Durante questo lavoro si è notato, basandosi sull'osservazione delle sequenze codificate, che l'occhio umano è poco sensibile alla distorsione introdotta in aree con poco dettaglio e che siano molto luminose o (in maniera minore) molto scure. L'occhio viene come abbagliato (o accecato nel caso di zone scure) e non percepisce la distorsione introdotta durante la codifica. Per sfruttare questo effetto percettivo si è stabilito che è possibile quantizzare maggiormente le zone uniformi molto chiare o molto scure all'interno di una picture. L'encoder MPEG sviluppato esegue questa azione valutando su ogni macroblocco una funzione di *smoothness*. Se il macroblocco viene classificato *smooth*, il coefficiente di quantizzazione calcolato dal processo di controllo del bit rate viene moltiplicato per 2,5, in modo da quantizzare tale macroblocco in maniera molto più accentuata. L'obiettivo è ottenere una distorsione concentrata solo in alcune zone, dove essa non è percepibile, evitando così di avere una distorsione uniforme ma avvertibile su tutta la picture.

5.8.1 La funzione di smoothness

La funzione di smoothness viene valutata solo sui macroblocchi codificati con tecniche di tipo Intra. Nella realizzazione fatta si è scelto di valutare la luminosità di un macroblocco sulla sola componente di luminanza. In MPEG la DCT viene fatta sui blocchi (non sui macroblocchi) e poiché un macroblocco è composto da quattro blocchi di luminanza, un macroblocco viene classificato smooth se ognuno dei quattro blocchi di luminanza è anche smooth.

La funzione di smoothness prende in esame il rapporto tra l'energia totale del blocco e l'energia concentrata nel coefficiente DC ottenuto con la DCT: se tale rapporto supera una certa soglia il blocco è smooth. Si è stabilito di avere 0.8 come soglia per le zone luminose e 0.9 per le zone scure.

5.8.2 Prove sull' algoritmo di smoothness

Per provare l'efficacia di questo algoritmo sono stati codificati frammenti di sequenze tratti da riprese naturali (sequenza "Biciclette") e da animazioni di sintesi (sequenza "IT").

A rigore l'unico parametro oggettivo che può essere valutato è il numero di macroblocchi sui quali l'algoritmo va ad innalzare il coefficiente di quantizzazione. Se infatti tale numero rimane basso significa che pochi bit risparmiati qua e là non rappresentano un reale beneficio per la qualità globale della codifica.

La Tabella 5.6 mostra i risultati sperimentali ottenuti: la percentuale rappresenta la frazione di macroblocchi di tipo Intra che sono stati classificati smooth e sui quali l'algoritmo ha quindi innalzato il coefficiente di quantizzazione.

Nome della sequenza	% macroblocchi
Biciclette	5,22
IT	21,61

Tabella 5.6 Percentuale di macroblocchi considerati “smooth”

Si può notare che la sequenza IT presenta un numero maggiore di macroblocchi che soddisfano le condizioni stabilite dalla funzione di smoothness: questo è dovuto al fatto che le immagini di sintesi presentano più zone di colore uniforme che, se sono anche molto luminose o molte scure, vengono selezionate dall’algoritmo.

Per quanto riguarda il numero di bit risparmiati innalzando il coefficiente di quantizzazione va notato che esso non può essere valutato dalle dimensioni del bitstream prodotto in quanto lo scopo dell’algoritmo è proprio distribuire diversamente i bit pur mantenendo uguale il bit rate globale. Per superare questo inconveniente è stato temporaneamente modificato l’algoritmo di controllo del bit rate in modo che i bit risparmiati in un GOP non andassero a beneficio del GOP successivo. Inoltre le codifiche sono state fatte usando solo picture di tipo I in modo che il GOP coincidesse con la singola picture e che i bit risparmiati in un macroblocco andassero solo a favore dei macroblocchi successivi all’interno della stessa picture. Per tenere conto anche di quest’ultimo effetto sono state fatte prove impostando diversi valori del bit rate. A questo punto è stato possibile valutare l’efficacia dell’algoritmo guardando direttamente le dimensioni in byte dei bitstream prodotti che sono riportate nella Tabella 5.7 per la sequenza “Biciclette” e nella Tabella 5.8 per la sequenza “IT”.

Bit rate nominale (bps)	Dimensioni del bitstream senza algoritmo di smoothing (byte)	Dimensioni del bitstream con algoritmo di smoothing (byte)	Guadagno %
5.000.000	5.129.343	5.100.157	0,57
10.000.000	9.978.036	9.943.643	0,34
15.000.000	14.927.043	14.865.157	0,41

Tabella 5.7 Effetto dell’algoritmo di smoothing sulla sequenza “Biciclette”

Bit rate nominale (bps)	Dimensioni del bitstream senza algoritmo di smoothing (byte)	Dimensioni del bitstream con algoritmo di smoothing (byte)	Guadagno %
1.872.000	6.132.054	5.855.677	4,51
15.000.000	26.611.306	26.594.115	0,06

Tabella 5.8 Effetto dell’algoritmo di smoothing sulla sequenza “IT”

Infine nella Tabella 5.9 sono riportate le misure di PSNR per “Biciclette” e “IT” al variare del bit rate. Bisogna precisare che tale misura non è molto significativa per valutare direttamente l’efficacia di questo algoritmo, in quanto il suo scopo è proprio quello di risparmiare bit perdendo informazioni soggettivamente non percepibili mentre il PSNR è una misura oggettiva di qualità. Il valore di PSNR riportato in tabella è in decibel ed è la media su tutta la sequenza. Inoltre, per ottenere questi risultati è stato ripristinato l’algoritmo di controllo del bit rate in modo che i bit risparmiati fossero allocati in zone più critiche della sequenza, contribuendo ad aumentarne la qualità.

Tipo di sequenza	PSNR senza algoritmo di smoothing	PSNR con algoritmo di smoothing
IT 1.872.000 bps	36,38	36,35
IT 15.000.000 bps	51,18	51,18
Biciclette 5.000.000 bps	25,62	25,6
Biciclette 10.000.000 bps	29,38	29,32
Biciclette 15.000.000 bps	31,8	31,75

Tabella 5.9 Effetti dell’algoritmo di smoothing sul PSNR delle sequenze

Le prove effettuate dimostrano che in complesso l’algoritmo di quantizzazione legato alla smoothness permette un certo risparmio di bit, maggiore nelle animazioni sintetiche che in quelle naturali. Tali bit possono così essere allocati in zone più critiche della sequenza contribuendo a mantenere alta la qualità della codifica.