

# Capitolo 4

## Compressione sorgenti dinamiche

- **Audio**

- DPCM
- APCM e ADPCM
- LPC
- CELP
- MPEG-1
- MPEG-2
- Dolby

- Video

- MJPEG
- Frame I,P,B
- Stima moto e compensazione
- H.261
- H.263
- MPEG
- Prestazioni
- Codificatori avanzati
  - H.264
  - FGS

# Motion JPEG (MJPEG)

- **Utilizzo di un compressore JPEG per ogni frame**
- Semplicità
- Non sfrutta la correlazione presente tra frame successivi
- Rapporto di compressione: tra 10 e 20 volte
- Utilizzo: sistemi a bassa complessità (es: telecamere IP)

# Ridondanza

- **Singola immagine**
- **Immagini successive**
  - Compressione prende come riferimento un'immagine e valuta cosa risulta cambiato nel passaggio alla seconda immagine
  - Differenze: parti della prima immagine che si sono spostate, indicando anche la variazione per poterla correggere
  - La seconda immagine viene creata a partire dalla prima sull'indicazione delle parti che sono cambiate e delle variazioni
  - **Meccanismo PREDITTIVO**: un'immagine viene formata a partire da un frame di riferimento, dalla **STIMA DEL MOTO** e dalla sua **COMPENSAZIONE**
  - Compressione se la rappresentazione di stima moto e compensazione richiedono meno bit rispetto al frame originale. Anche stima moto e compensazione possono essere compresse.
  - Predizione in avanti e all'indietro
  - Errori del meccanismo predittivo

# Predizione

## In avanti

$$\hat{Z}_k = \begin{cases} Z_k & \text{se } k \bmod N = 0 \\ f^{k \bmod N} \left( Z_{N \lfloor \frac{k}{N} \rfloor} \right) & \text{altrimenti} \end{cases}$$

$Z_0$  riferimento

**Esempio,  $N=3$**

$$Z_0^\wedge = Z_0$$

$$Z_1^\wedge = f(Z_0)$$

$$Z_2^\wedge = f^2(Z_0)$$

$$Z_3^\wedge = Z_3$$

$$Z_4^\wedge = f(Z_3)$$

$$Z_5^\wedge = f^2(Z_3)$$

$$Z_6^\wedge = Z_6$$

## All'indietro

$$\hat{Z}_k = \begin{cases} Z_k & \text{se } k \bmod N = 0 \\ f^{k \bmod N} \left( Z_{N \lfloor \frac{k}{N} \rfloor} \right) & \text{se } k \bmod N = 1, \dots, \lfloor \frac{N}{2} \rfloor \\ f^{-(N-k \bmod N)} \left( Z_{N \lfloor \frac{k}{N} \rfloor + 1} \right) & \text{altrimenti} \end{cases}$$

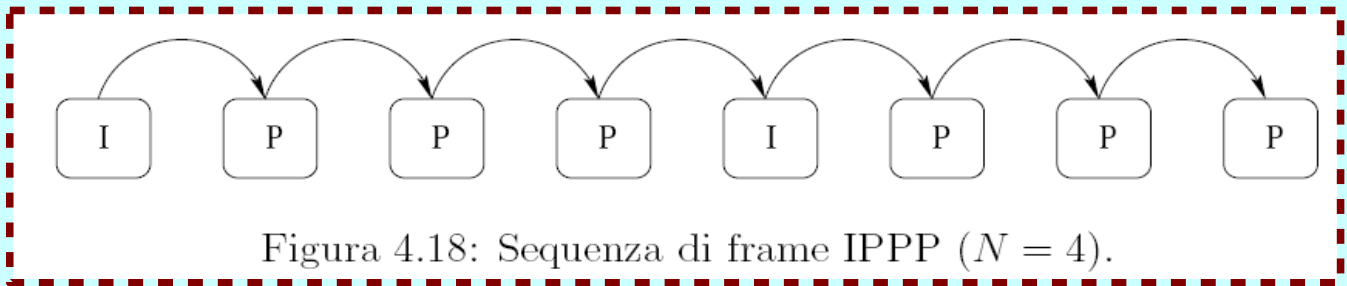
# Frame I, P, B

- **I intracoded**
  - Compresse indipendentemente da altri frame, con JPEG
  - Rapporto compressione ( $\rho_I$ ) tra 10 e 20 volte
- **P Predictive**
  - Predizione in avanti
  - $\rho_P$  tra 20 e 30 volte
  - $Z_k^{\wedge} = f(Z_{k-b})$ ,  $b \geq 1$
- **B Bidirectional**
  - Utilizza anche predizione all'indietro
  - $\rho_B$  tra 30 e 50 volte
  - $Z_k^{\wedge} = f(Z_{k-b}, Z_{k+w})$
- **GOP (Group Of Picture)**: distanza N tra due frame di tipo I. (N tra 3 e 12)

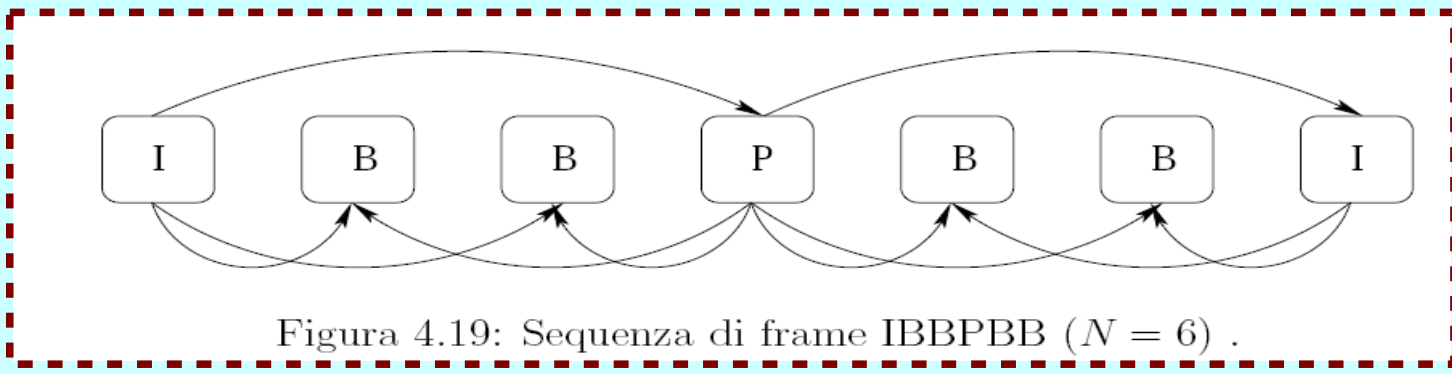
# Frame I, P, B

$$N = 1 + n_P + n_B$$

$$\rho = \frac{\rho_I + n_P \rho_P + n_B \rho_B}{N}$$



$$\rho = 23.3$$

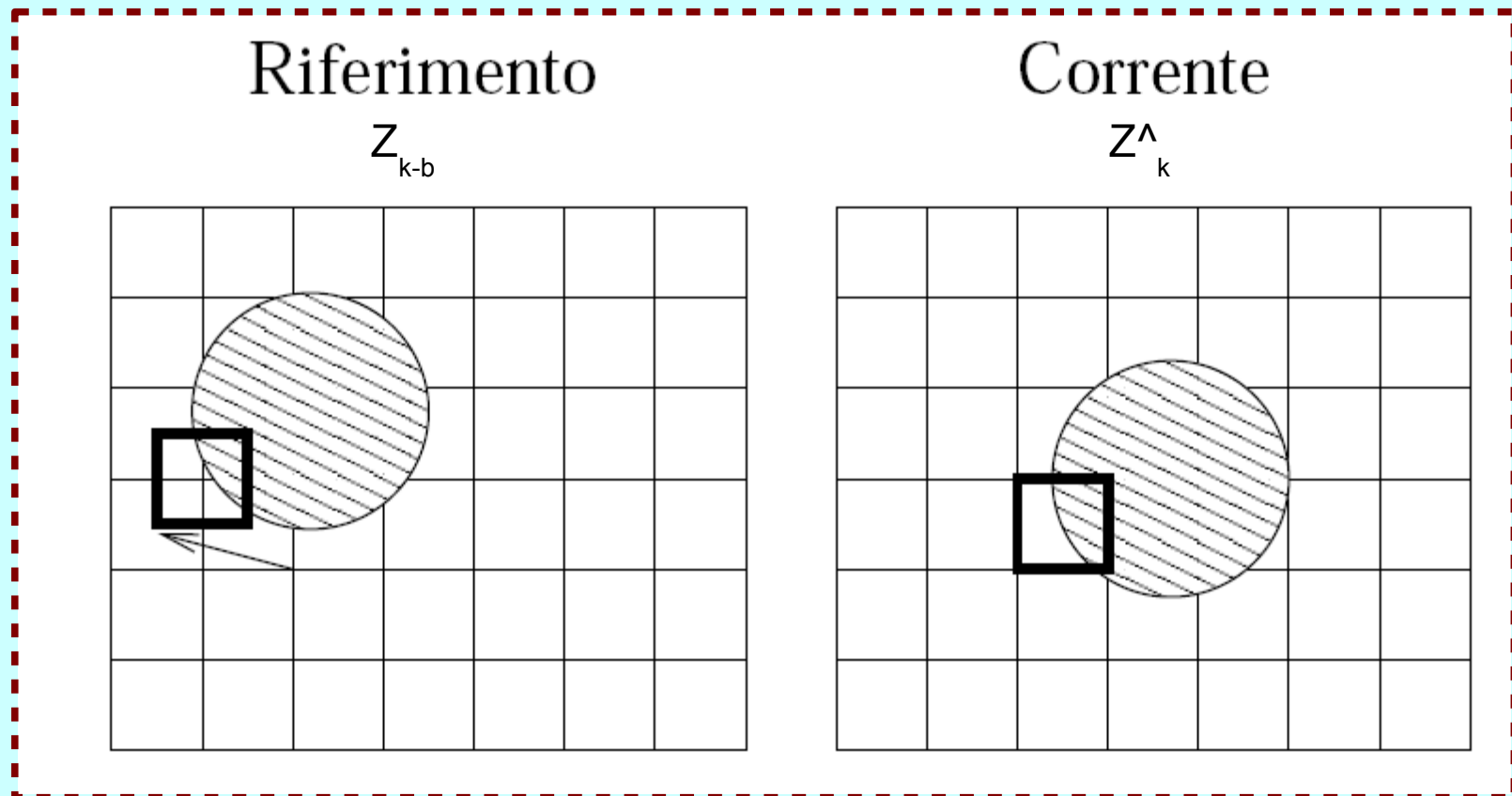


$$\rho = 33.3$$

- Riordino di sequenze → frame PB

# Stima del moto

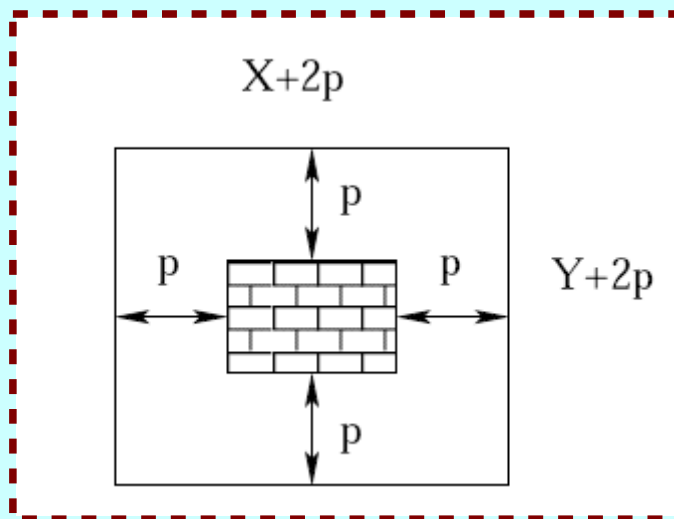
- Frame corrente suddiviso in **MACROBLOCCHI** rettangolari **XxY**
- Ricerca basata su differenza tra macroblocchi (info di *luminanza*)
- **VETTORE DI MOTO** = differenza tra posizione di un qualsiasi angolo del macroblocco corrente e la rispettiva posizione del macroblocco sul frame di riferimento
- Movimenti → traslazioni



# Stima del moto

- Metrica di confronto: **Mean Absolute Difference (MAD)**
- **Area di ricerca:**  $(X+2p) \times (Y+2p)$  → dove variano gli indici  $i$  e  $j$

$$MAD(i, j, x, y) = \frac{1}{XY} \sum_{m=1}^X \sum_{n=1}^Y |Z_{k-b}(x + m + i, y + n + j) - Z_k(x + m, y + n)|$$



**x** e **y**: angolo di partenza del macroblocco  
**m** e **n**: per ogni pixel del macroblocco  
**i** e **j**: per esplorare immagine di riferimento

- **Principio di Località** → un buon match si trova in vicinanza di un altro buon match → **ALGORITMI DI RICERCA DEL VETTORE DI MOTO**
- **FULL SEARCH** →  $(2p+1)^2$  valutazioni di MAD
- **Soglia di Accettabilità** Cap.3 - Compressione Sorgenti Statiche



# Q-Step Search

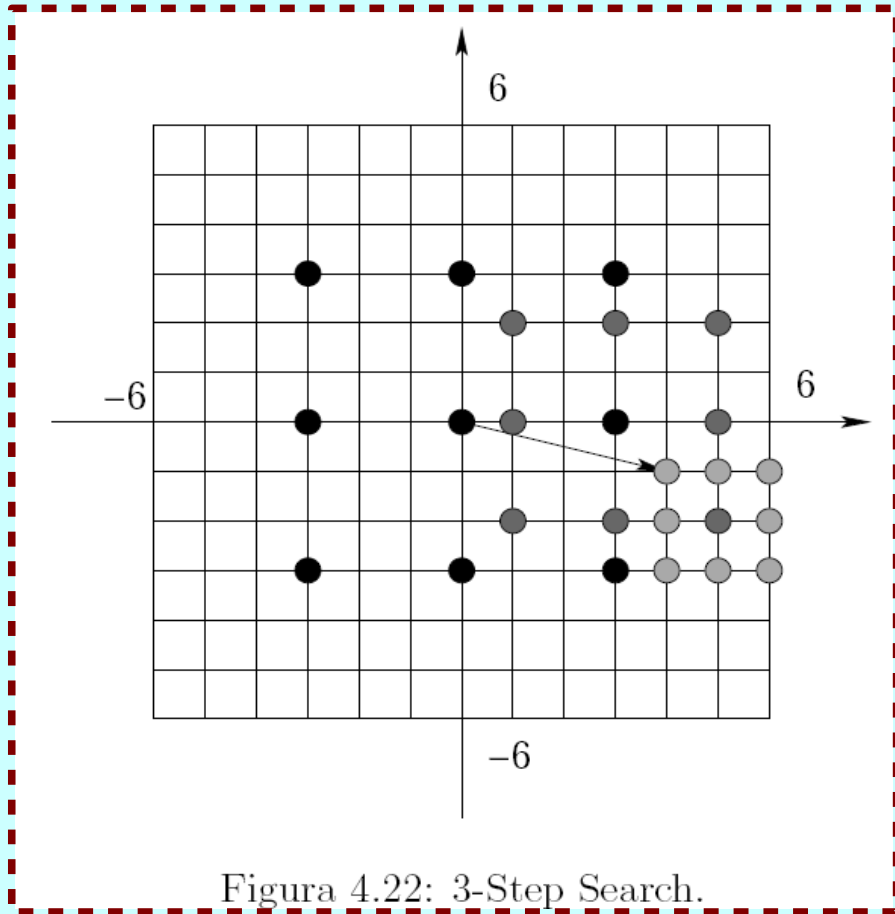
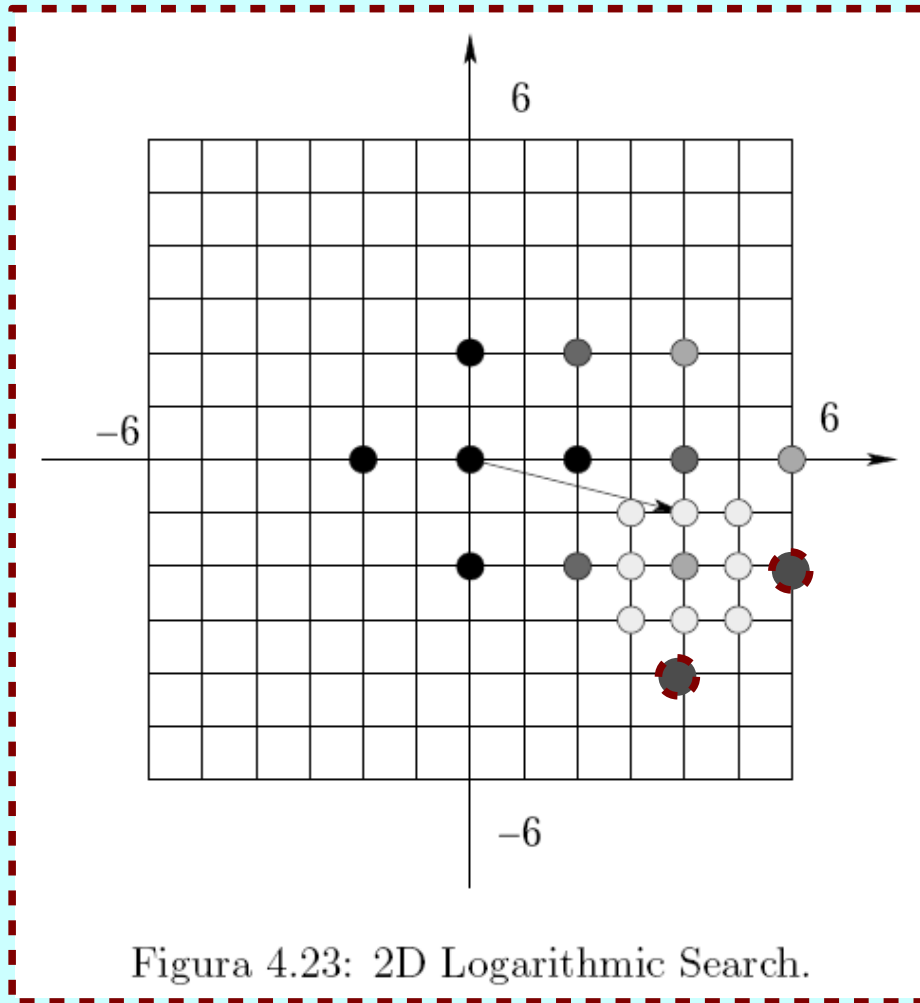


Figura 4.22: 3-Step Search.

- 9 confronti di MAD ad ogni passo
- $(i,j) \in \{x_s, x_s+d_s, x_s-d_s\} \times \{y_s, y_s+d_s, y_s-d_s\}$
- $(x_{sm}, y_{sm})$  MAD inferiore
- Start:
  - $s=1, d_1=p/2, x_s=y_s=0$
- Ad ogni passo ( $s=s+1$ ):
  - $d_s = d_{s-1} - 1$
  - $x_s = x_{s-1 m}, y_s = y_{s-1 m}$
- Stop:
  - $d_s = 1$
- # valutazioni MAD =  **$9p/2$**

# 2D Logarithmic Search



- Al max **5** confronti di **MAD** per passo, tranne ultimo passo (**9** confronti)
- $(i,j) \in \{(x_s, y_s), (x_s - d_s, y_s), (x_s + d_s, y_s), (x_s, y_s - d_s), (x_s, y_s + d_s)\}$
- $(x_{sm}, y_{sm})$  MAD inferiore
- Start:
  - $s=1, d_1 = 2^{\text{ceil}(\log_2 p) - 1}, x_s = y_s = 0$
- Ad ogni passo ( $s=s+1$ ):
  - $d_s = (d_{s-1} - 1) / 2$  se il punto con MAD minimo è il centrale, altrimenti rimane invariata
- Stop:
  - $d_s = 1$

# Orthogonal Search

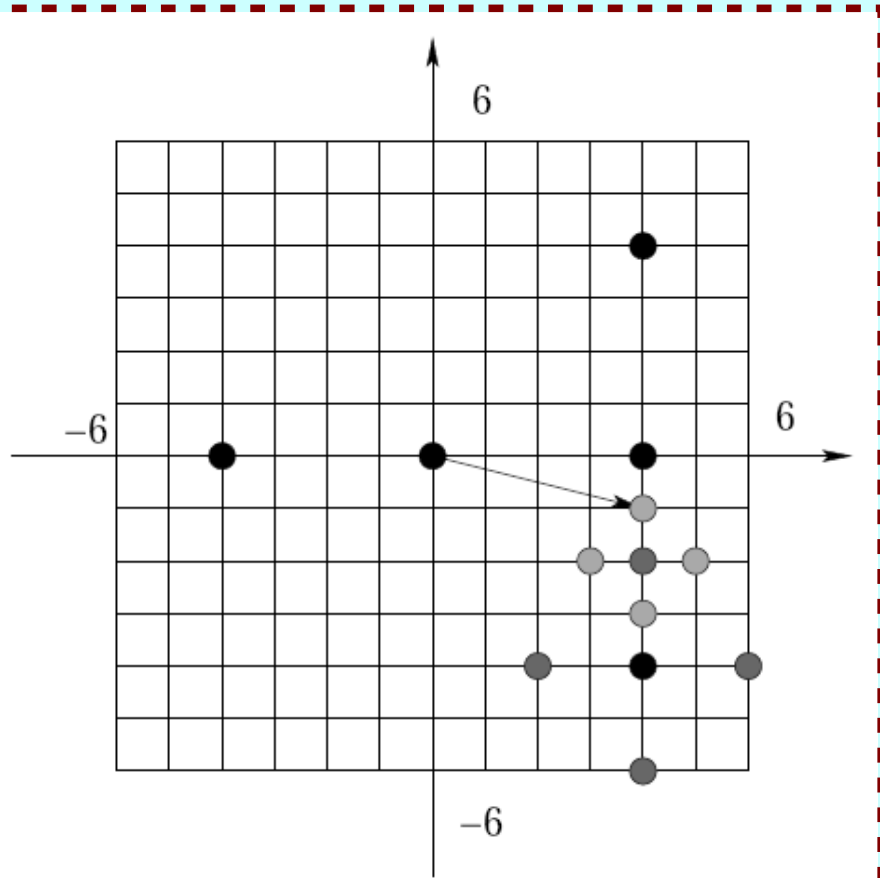


Figura 4.24: Orthogonal Search.

- 5 confronti di MAD ad ogni passo
- Ricerca orizzontale e SUCCESSIVA da ricerca verticale
- Start:
  - $s=1, d_1=p/2 + 1, x_s=y_s=0$
- Ad ogni passo ( $s=s+1$ ):
  - $d_s = \text{ceil} [(d_{s-1} - 1)/2]$
- Stop:
  - $d_s = 1$
- # passi =  $\text{Ceil}[\log_2(p/2+1)]+1$
- # valutazioni MAD =  $5 * (\text{Ceil}[\log_2(p/2+1)]+1)$

# Stima del moto

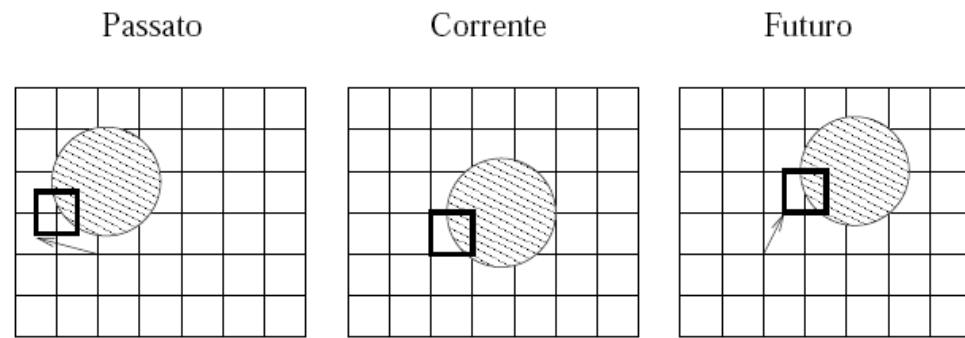


Figura 4.25: Vettori di moto per un macroblocco in frame B.

- **FRAME B**

- **Coppia**

$$MAD_b(i, j, x, y) = \frac{1}{XY} \sum_{m=1}^X \sum_{n=1}^Y |Z_{k-b}(x+m+i, y+n+j) - Z_k(x+m, y+n)|$$
$$MAD_w(i, j, x, y) = \frac{1}{XY} \sum_{m=1}^X \sum_{n=1}^Y |Z_{k-w}(x+m+i, y+n+j) - Z_k(x+m, y+n)|$$

- **di vettori di predizione di moto**

- Algoritmi di ricerca lavorano indipendentemente sui due vettori di MAD
    - I vettori di moto vengono codificati con **codifica differenziale**, relativa a vettori di moto di macroblocchi adiacenti nella stessa immagine, seguita da codificatore **Huffman** a lunghezza variabile
  - **Video a colori**
    - immagine originale scomposta nel piano colore **YCbCr**
    - Vettore di moto calcolato solo sul piano **Y** e si assume valido anche per Cb e Cr

# Compensazione del moto

- Applicata su Y e su Cb e Cr
- Le compensazioni vengono codificate con la codifica usata per frame I
- Frame P

$$U(x, y) = Z_k(x, y) - Z_{k-b}(x + x^m, y + y^m)$$

- Frame B
  - Si sceglie l'immagine differenza con il minor numero di differenze

$$U_b(x, y) = Z_k(x, y) - Z_{k-b}(x + x_b^m, y + y_b^m)$$

$$U_w(x, y) = Z_k(x, y) - Z_{k+w}(x + x_w^m, y + y_w^m)$$

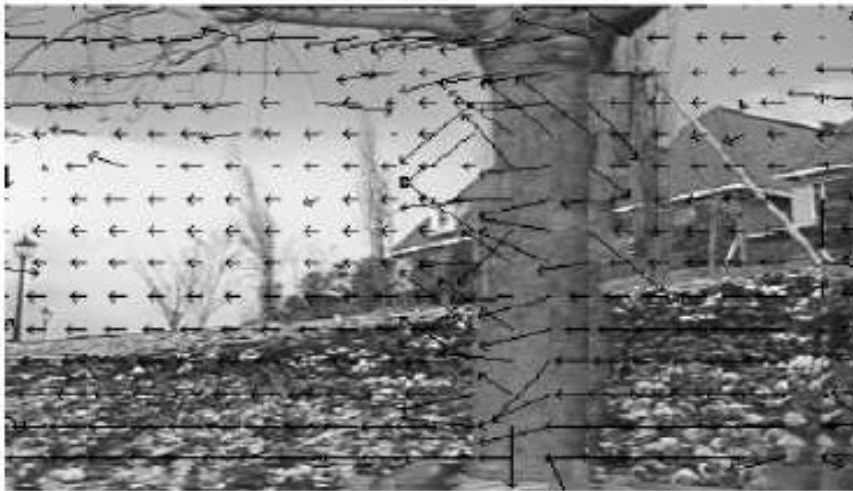
$$U_m(x, y) = Z_k(x, y) - \frac{1}{2} [Z_{k-b}(x + x_b^m, y + y_b^m) + Z_{k+w}(x + x_w^m, y + y_w^m)]$$

# *Esempio di vettori di moto e di compensazione del moto (frame P)*

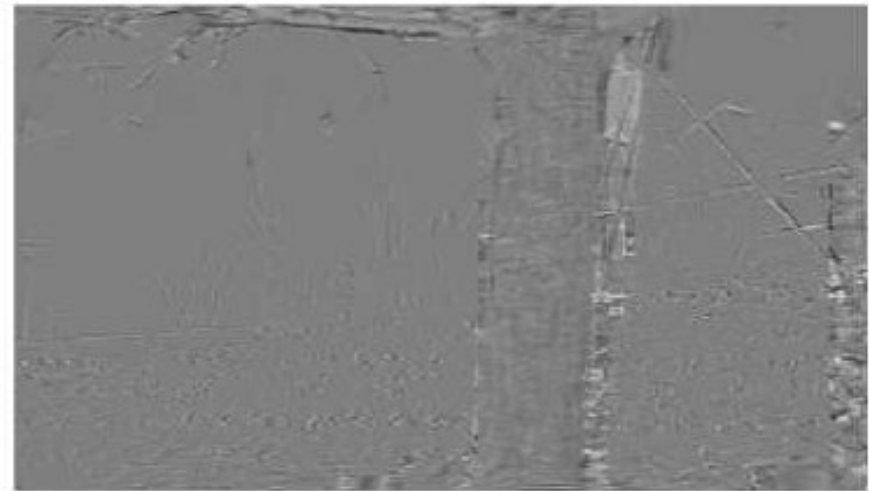
Frame passato



Frame corrente



Vettori di moto



Compensazione

# H.261

- Bit rate **64f Kbps**,  $f=1, \dots, 30$

- Formati: **CIF, QCIF**

- Frame: **I, P**

- Formato

- Inizio immagine
- Timestamp

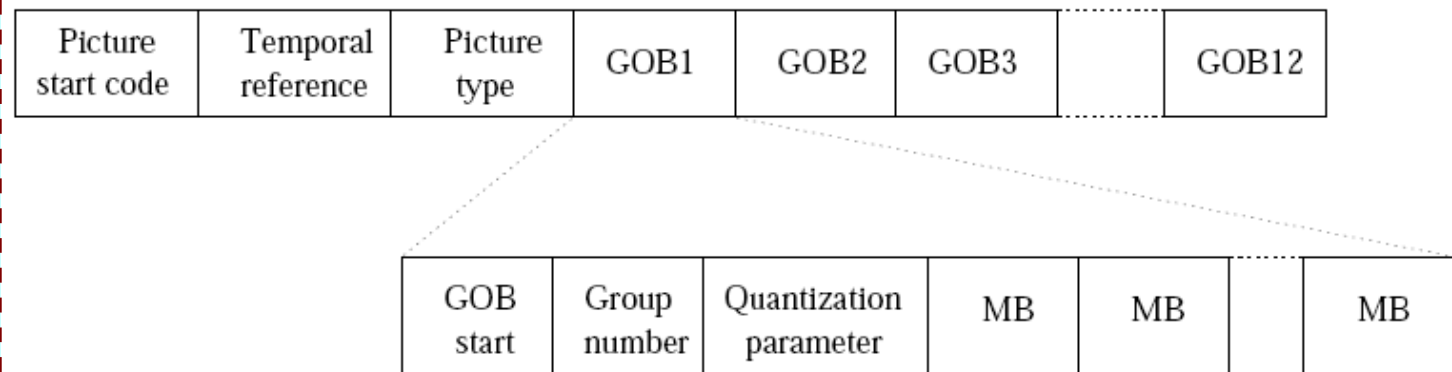


Figura 4.28: Formato H.261.

(sincronizzazione di riproduzione di diverse sorgenti al decodificatore, es: audio con video)

- Picture Type: I o P

- **GOB (3 per QCIF, 12 per CIF)**

- Ogni GOB è composto da **33 MacroBlocchi (MB)**
- GOB start utile per sincronizzazioni in caso di problemi sul canale: in questo modo si può recuperare sincronismo sulla stessa immagine, senza attendere immagine successiva e il suo picture start code.

# H.261

- **GOB (3 per QCIF, 12 per CIF)**
  - Ogni GOB è composto da **33 MacroBlocchi (MB)**
  - GOB start utile per sincronizzazioni in caso di problemi sul canale: in questo modo si può recuperare sincronismo sulla stessa immagine, senza attendere immagine successiva e il suo picture start code.

**CIF**

**QCIF**

GOB	

GOB

**GOB**

1	2			11
				33



# H.261

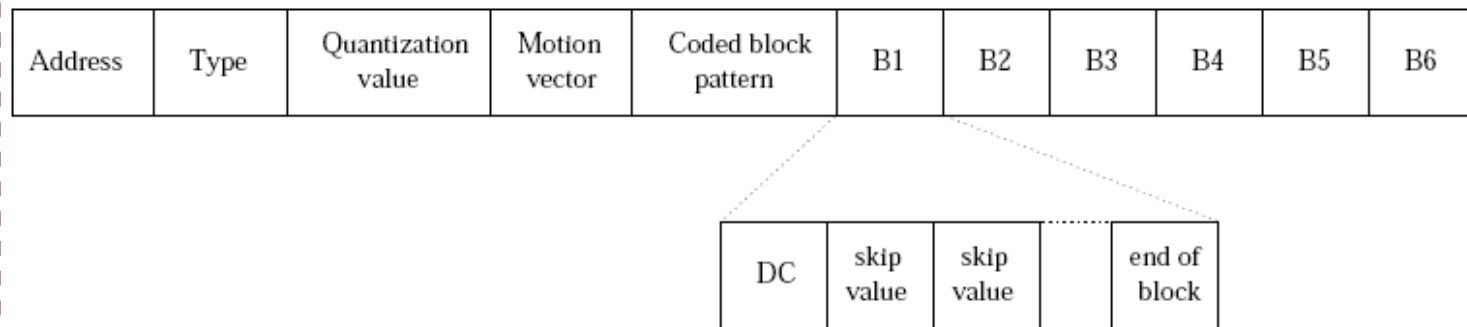


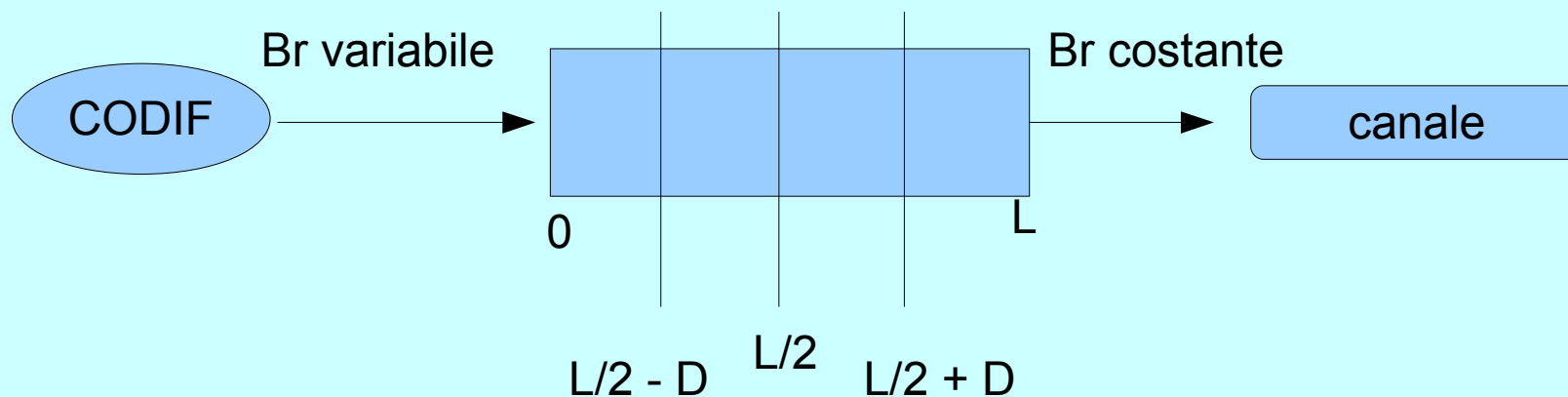
Figura 4.27: Macroblocco H.261.

## • **Macroblocco**

- 16x16 pixel per Y
- 8x8 per Cb e Cr (CIF e QCIF hanno formato 4:2:0)
- Address: identificativo
- Type: codificato indipendente o in riferimento ad altri macroblocchi (stima moto)
- Valore di quantizzazione (il sistema usa una matrice di quantizzazione uniforme, tutti i valori uguali)
- Vetore di moto, nel caso di frame P
- Coded block pattern: specifica quali blocchi sono successivamente presenti, fino a un massimo di 6
- **Blocco** → valori quantizzati e trasformati (DCT) rappresentati con meccanismo RLE (DC, skip, value). No Huffman. Anche vettore moto e compensazione sono codificate.

# H.261

- Risoluzioni
  - Multipli di 16 pixel
  - CIF: 352x288 (Y), 176x144 (Cb,Cr) → 12 GOB (a matrice 2x6)
  - QCIF: 176x144 (Y), 88x72 (Cb,Cr) → 3 GOB (a matrice 1x3)
- Buffer FIFO per uniformare la bit rate del flusso verso il canale di comunicazione



$$\alpha_k = \begin{cases} \alpha_{k-1} - C & \text{se } O_k > \frac{L}{2} - \delta \text{ e } \alpha_{k-1} > T_m \\ \alpha_{k-1} + C & \text{se } O_k < \frac{L}{2} + \delta \text{ e } \alpha_{k-1} < T_M \\ \alpha_{k-1} & \text{altrimenti} \end{cases}$$

# H.261 - prestazioni

- PSNR calcolato come media del PSNR di ogni singolo frame
- Due spezzoni di 60s presi da due film diversi
- Formato CIF

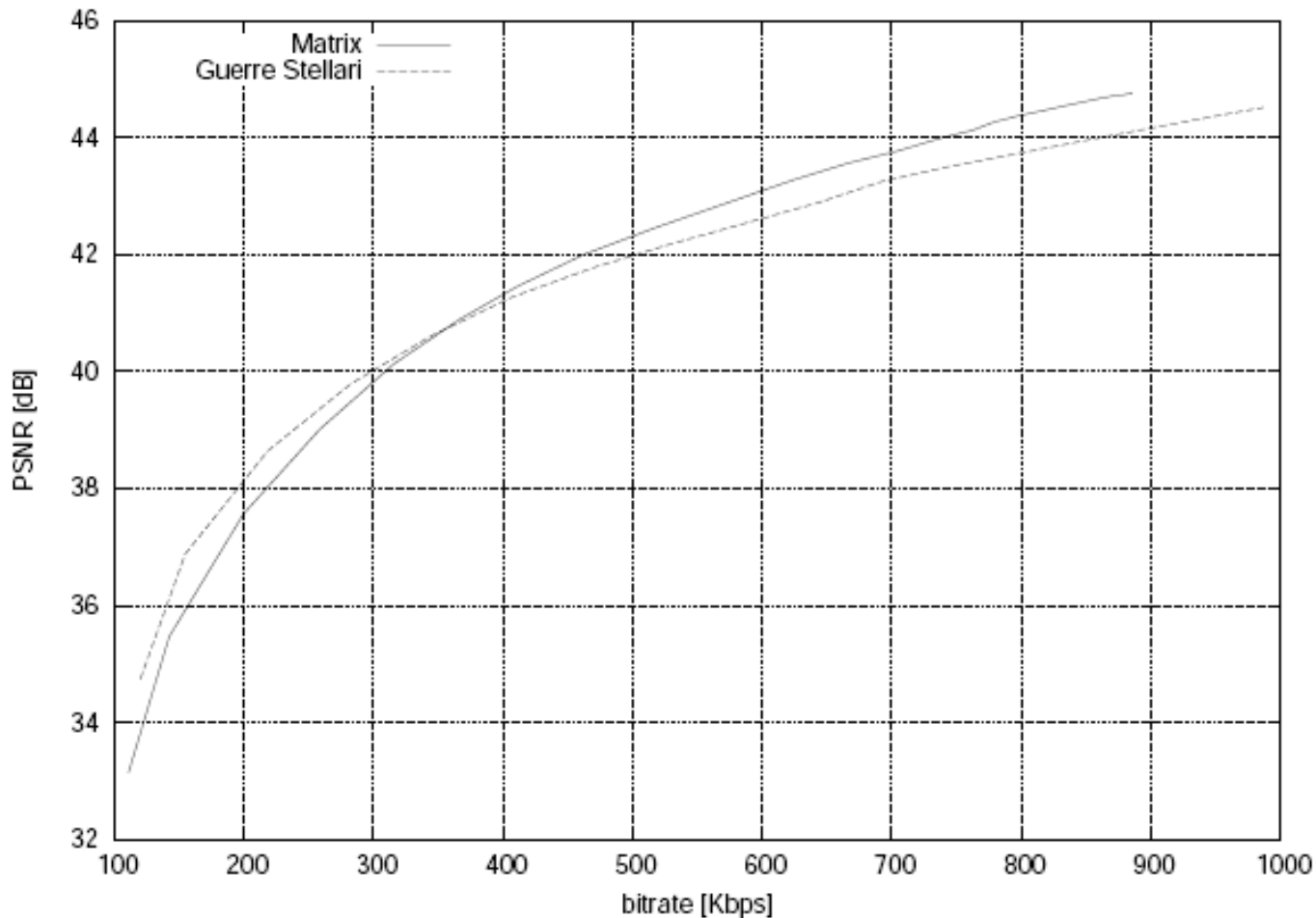


Figura 4.29: Prestazioni H.261.

# H.263

- Canali a bassa bit rate
- Codificatore più avanzato
- Formati: **CIF, QCIF, SQCIF, 16CIF, 4CIF**
- Frame: **I, P, B**
- Utilizza codifica Huffman
- Formato: simile a H.261
- Stima del moto: possibile anche al di fuori dell'area classica di ricerca
- Resistenza agli errori causati dal canale
  - Errore su un singolo GOB: decoder salta al GOB successivo e sostituisce macroblocchi con macroblocchi dell'immagine precedente
  - Assumendo la presenza di un canale di ritorno per scambio segnalazioni
    - ERROR TRACKING
    - INDEPENDENT SEGMENT DECODING
    - REFERENCE PICTURE SELECTION

# H.263 – Resistenza agli errori

- **ERROR TRACKING**

- Decompressore se rileva errori invia NACK alla sorgente, specificando immagine e posizione nel GOB in cui è presente errore
- Compressore valuta quali GOB successivi risulterebbero molto probabilmente errati a causa dei meccanismi di predizione e dell'errore su quel GOB e invia tali GOB codificati come I invece che come predetti

- **INDEPENDENT SEGMENT DECODING**

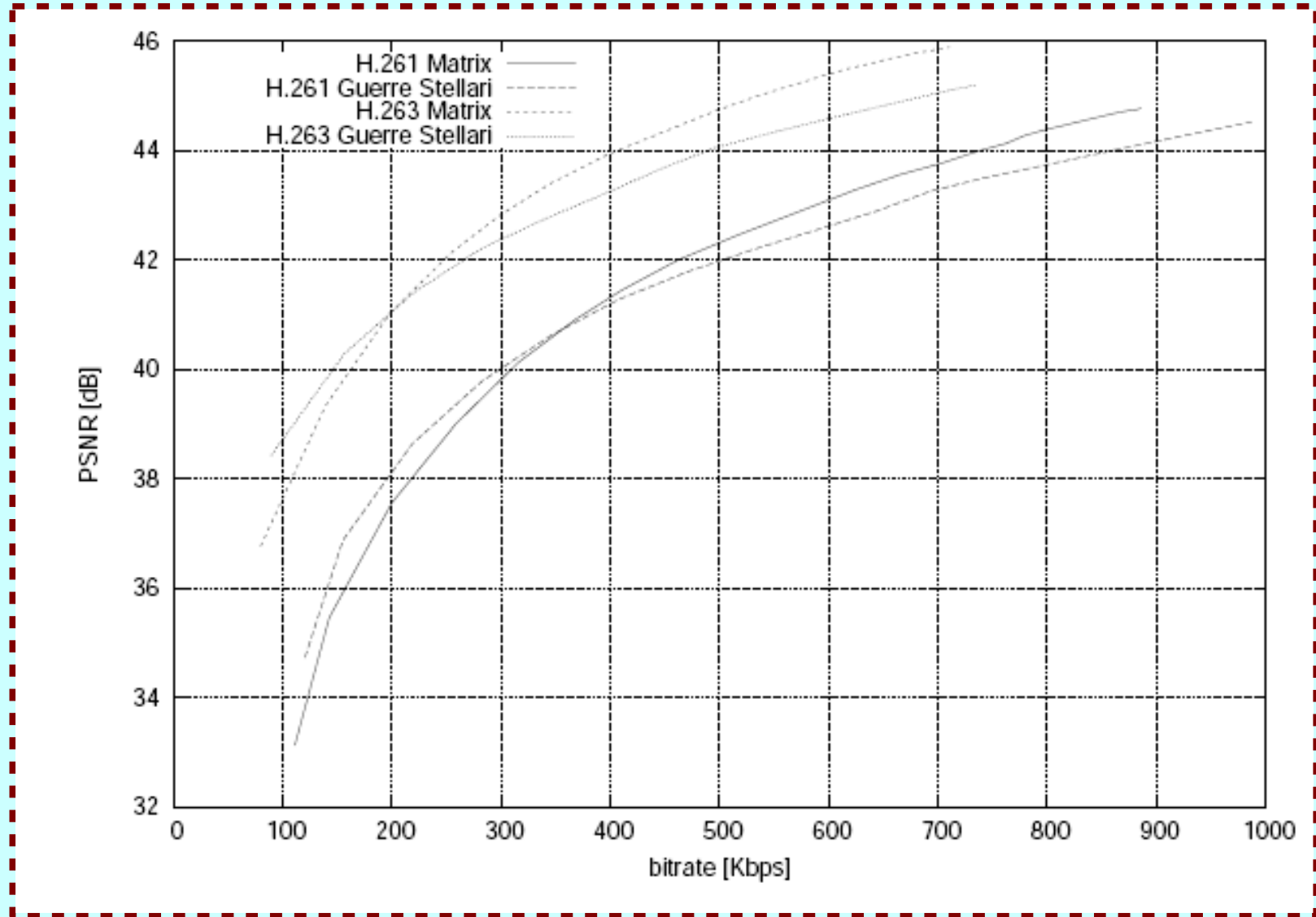
- Compressore tratta indipendentemente ogni GOB, limitando area di ricerca solo per macroblocchi contenuti all'interno di ogni GOB. Questo evita propagazione di errori in GOB adiacenti ma riduce efficacia di compressione

- **REFERENCE PICTURE SELECTION**

- Analogo a error tracking ma per la codifica predittiva vengono considerati solo GOB per i quali si è certi che siano corretti
- Decoder invia alla sorgente indicazione dei soli blocchi errati (NACK) e tutti gli altri possono essere usati come riferimento o indicazione dei soli blocchi corretti (ACK) e solo questi possono essere usati come riferimento

# H.263 – prestazioni

## Confronto H.261 e H.263



# MPEG

- Esistono vari codificatori MPEG (stream video associati ai corrispondenti stream audio)
- Formato: struttura gerarchica a più livelli
  - Alcuni parametri:
    - Buffer parameter (quantità di riempimento buffer da assumere prima di iniziare la decodifica)
    - Tipologia vettori di moto (risoluzione di vettore di moto, per specificarne accuratezza)
    - Slice: sequenza di un numero variabile di macroblocchi (gestione degli errori più dinamica)

# MPEG - formato

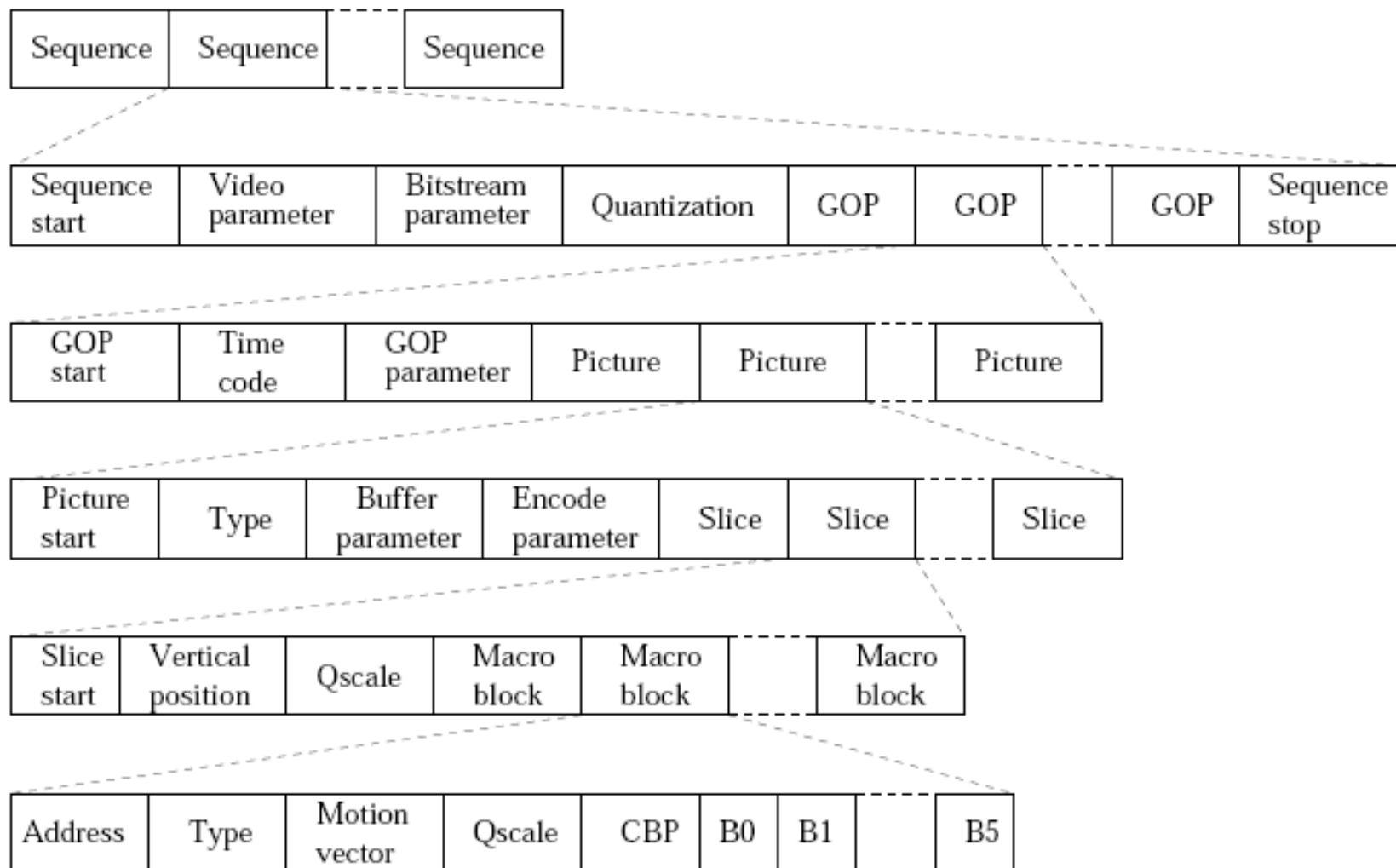


Figura 4.31: Formato file MPEG video.



# MPEG

- **MPEG-1**
  - Per memorizzazione di video su supporto digitale.
  - Formato SIF e Br=1.5Mbps
- **MPEG-2**
  - Sistema suddiviso in profili e livelli
  - **Profili** specificano **tipo di frame, formato di decimazione di crominanze, # stream paralleli supportati**
  - **Livelli** specificano **risoluzioni e bit rate**
- **MPEG-4**
  - Ogni scena definita come insieme di oggetti, composti in modo gerarchico, su uno sfondo
  - Ogni **oggetto** è codificato e inviato al dec con le info di dove e come posizionarlo
  - Esempi di codificatori: ***DIVX, XVID, H.264, FGS***

# MPEG-2

Profilo	Nome	Frame	Formato	Stream
SP	Simple Profile	I P	4:2:0	1
MP	Main Profile	I P B	4:2:0	1
422P	4:2:2 Profile	I P B	4:2:2	1
SNR	SNR Profile	I P B	4:2:0	1-2
SP	Spatial Profile	I P B	4:2:0	1-3
HP	High Profile	I P B	4:2:2	1-3

Livello	Nome	Pixel per linea	Linee	Bitrate (Mbps)
LL	Low Level	352	288	4
ML	Main Level	720	576	15
H-14	High 1440	1440	1152	60
HL	High Level	1920	1152	80

Nome	Risoluzione	Framerate (Hz)	Formato	Bitrate (Mbps)
SP@LL	176x144	15	4:2:0	0.096
SP@ML	352x288	15	4:2:0	0.384
SP@ML	320x240	24	4:2:0	0.384
MP@LL	352x288	30	4:2:0	4
MP@ML	720x480	30	4:2:0	15
MP@ML	720x576	25	4:2:0	15
MP@H-14	1440x1080	30	4:2:0	60
MP@H-14	1280x720	30	4:2:0	60
MP@HL	1920x1080	30	4:2:0	80
MP@HL	1280x720	60	4:2:0	80
422P@ML	720x480	30	4:2:2	50
422P@ML	720x576	25	4:2:2	50
422P@H-14	1440x1080	30	4:2:2	80
422P@H-14	1280x720	60	4:2:2	80
422P@HL	1920x1080	30	4:2:2	300
422P@HL	1280x720	60	4:2:2	300

# MPEG-4 – resistenza agli errori

- Codifica di **Huffman** a dimensione variabile e **REVERSIBILI** → letti sia da sinistra a destra che da destra a sinistra
  - La presenza di un errore, invece che far saltare al prox blocco di sincronizzazione presente nello stream (con perdita di tutte queste info) implica l'identificazione di tale punto e la decodifica dello stream all'indietro fino al punto in cui è avvenuto l'errore
- Costruzione di un codice di Huffman reversibile
  - Selezionare un codice a lunghezza variabile con PESO DI HANNING COSTANTE (stesso numero di “1” presenti in ogni parola)
  - Aggiungere un bit a uno in testa e un bit a 1 in coda, a ciascuna parola

Codificatore	A	B	C	D	A	A	A
	111	1011	10011	100011	111	111	111
			errore				
			↓				
Decodificatore	111	1011	10111	100011	111	111	111
	A,1	B,2		D,6	A,5	A,4	A,3

# H.264 – prestazioni

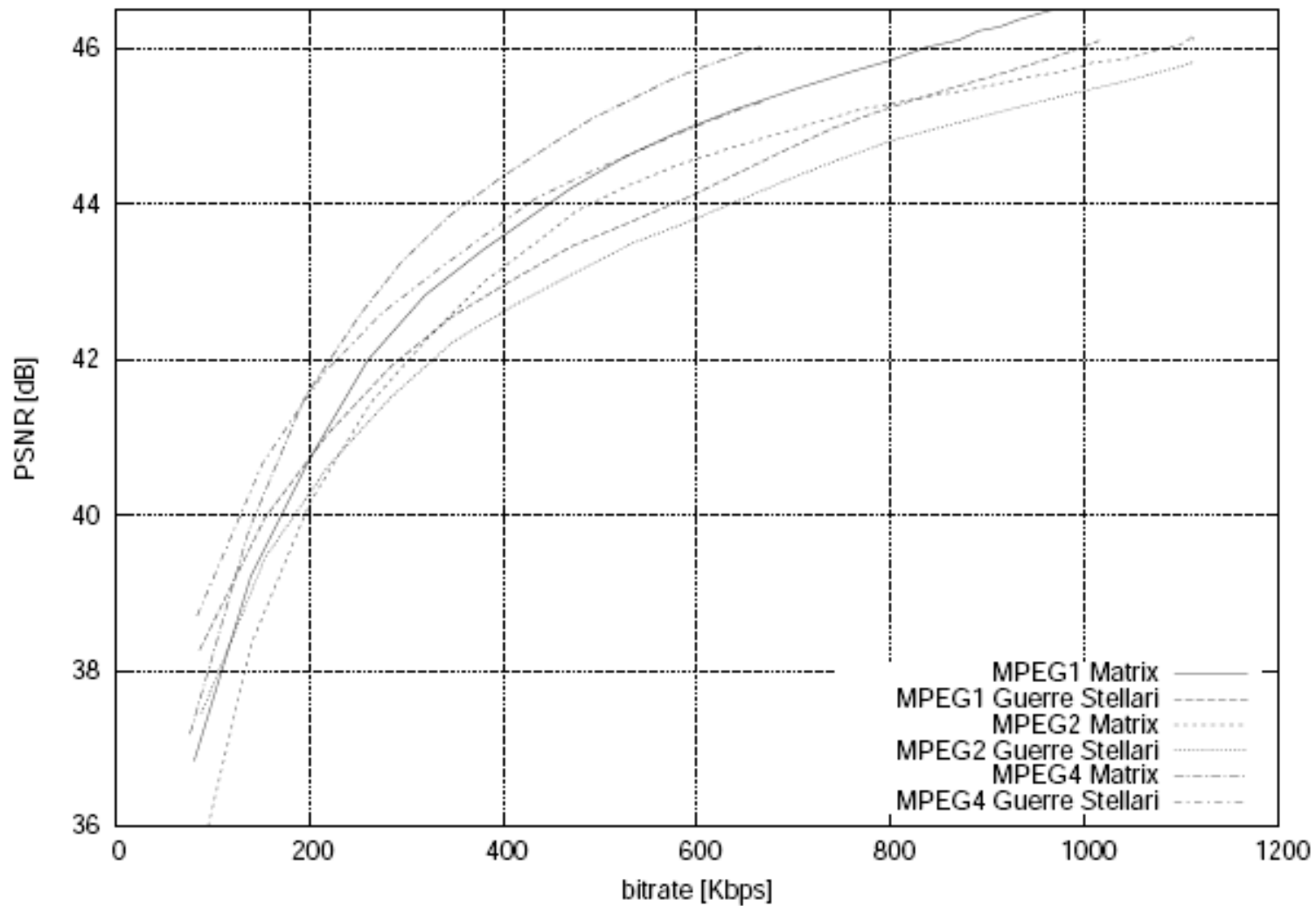
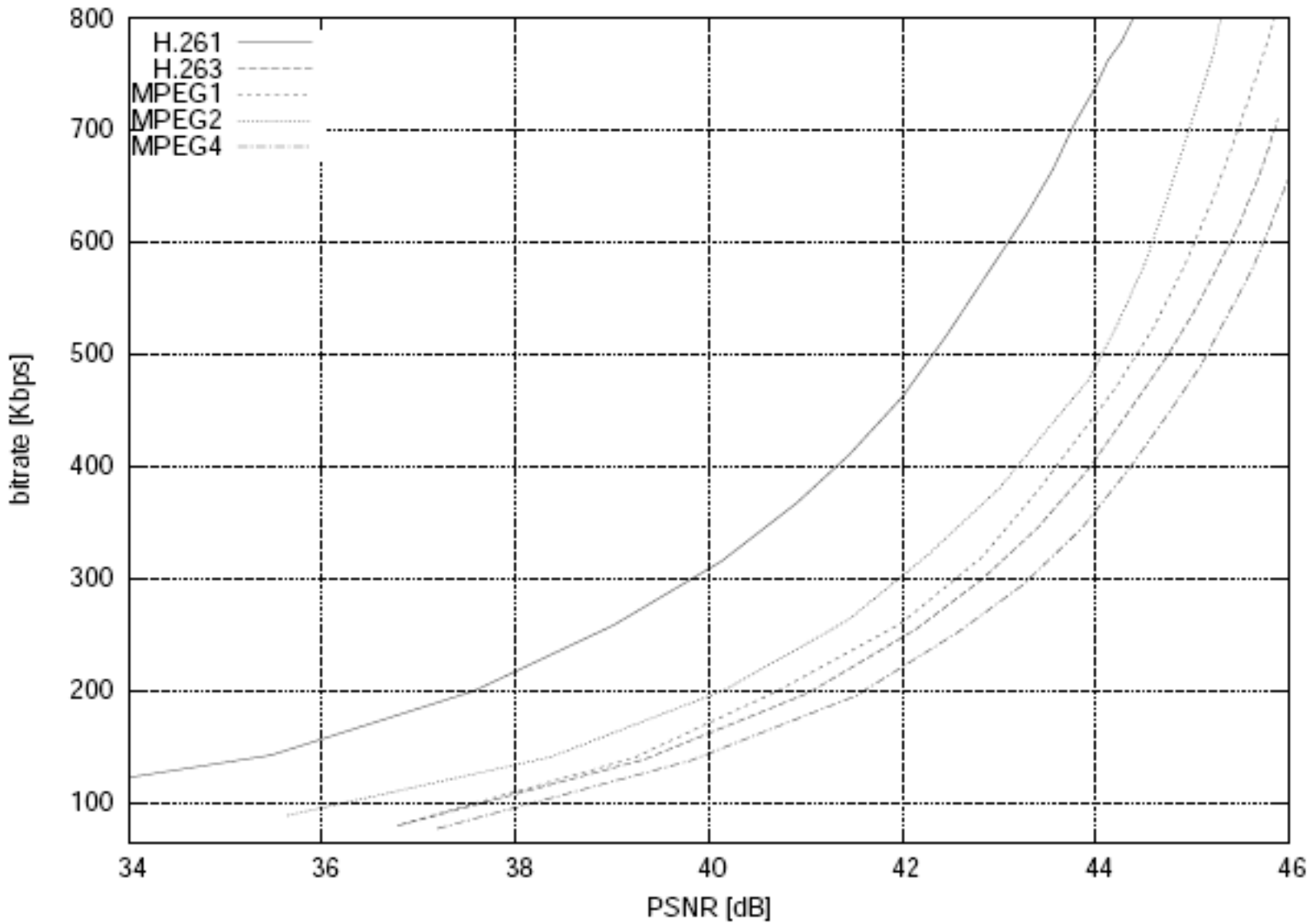


Figura 4.33: Prestazioni MPEG-1, MPEG-2, MPEG-4.

# Confronto prestazioni



# H.264

- **H.264 Advance Video Coding (AVC)** chiamato anche **MPEG-4 Parte 10**
- Elevata qualità a bit rate inferiori
- Meccanismi **predizione moto** migliorati
  - Scelta flessibile delle dimensioni del blocco di compensazione moto
  - Accuratezza rappresentazione vettore di moto
  - Più ampio numero di immagini di riferimento
- **Codifica aritmetica**
- Sistema di codifica a due livelli: **Video Coding Layer (VCL)** per contenuto video e **Network Abstraction Layer (NAL)** header per livello di trasporto o dispositivo di memorizzazione

# FGS

- **Reti radiomobili e Internet**
  - Richiede bit rate scalabili per adattarsi in tempo reale alle condizioni variabili del canale
- Codifica del bitstream è divisa in due layer:
  - **BASE LAYER** non scalabile con qualità minima
  - **ENHANCEMENT LAYER** scalabile usato per migliorare la qualità in caso le condizini del canale lo permettano