



# Video – Parte 4b

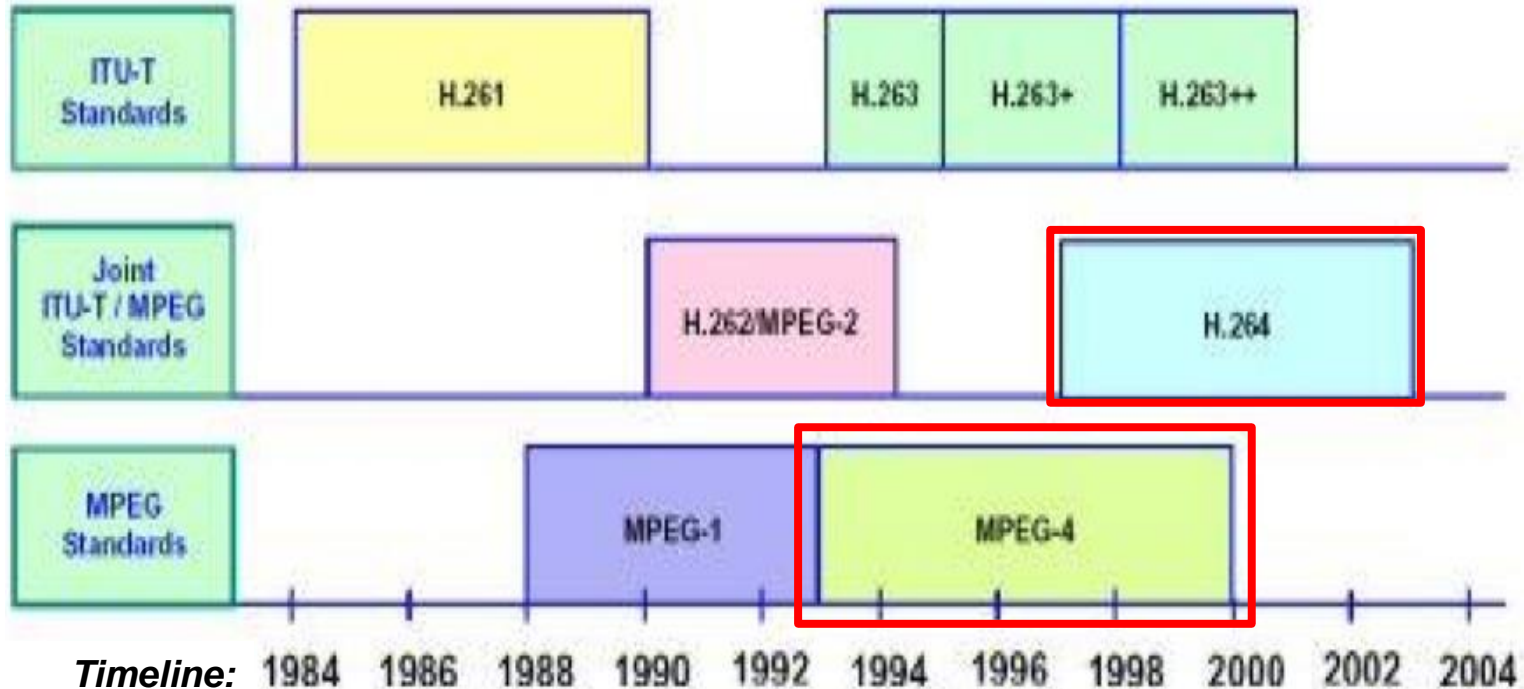
---

Codifiche e Compressione  
MPEG1, MPEG2, MPEG4  
H.264



# Formati di Compressione

## Associazioni

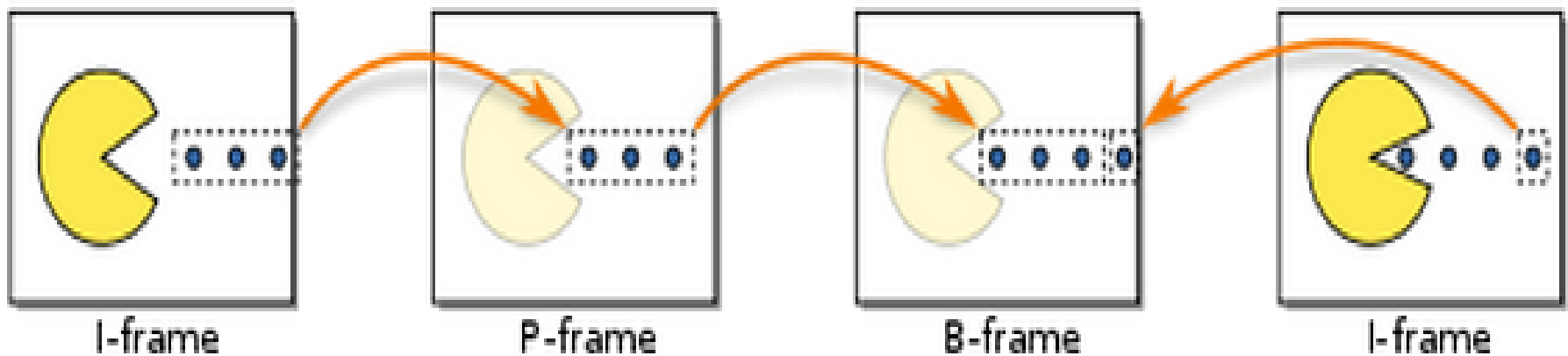


- L'immagine sovrastante mostra l'evoluzione nel tempo dei vari formati di compressione video realizzati dalle associazioni sopra indicate



# Classificare i frame

- *I-frame*: Intra/Independent (Key)
- *P-frame*: Predicted/Previous-Dependent
- *B-frame*: Bidirectional Predicted/Dependent



*Una sequenza di fotogrammi video, costituito da due fotogrammi chiave (I), un fotogramma predetto in avanti (P) e un fotogramma predetto bidirezionalmente (B)*



# Classificare i frame:

## *I-frames*

- **Intra, Independent (o Key) frames**
- I fotogrammi sono **codificati senza riferimento ad altri fotogrammi**
- Può essere generato da un encoder per creare un **punto di accesso casuale**
  - Per consentire a un decodificatore di avviare la decodifica in maniera corretta, partendo da zero in quella posizione del video
- In genere richiedono più bit per essere codificati rispetto ad altri tipi di frame
- Sono usati come riferimento per la decodifica di altre immagini



# Classificare i frame:

## *P-frames*

- **Predicted, Previous-dependent frames**
- Inizialmente **richiede la decodifica di qualche altro fotogramma precedente** per essere decodificato
- Può contenere
  - i dati del fotogramma
  - gli spostamenti (vettore di movimento) rispetto al fotogramma da cui dipende
  - una combinazione dei due
- In genere richiedono meno bit per la codifica rispetto agli I-Frames.



# Classificare i frame:

## *B-frames* (1)

- **Bidirectional Predicted, Bidirectional-dependent frames**
- **Richiede la precedente decodifica di altri frames** prima di essere decodificato
  - Sostanzialmente i fotogrammi B sono di tipo "Bidirezionale", nel senso che fanno riferimento sia a ciò che li precede, sia a quello che segue
- **Può contenere**
  - i dati del fotogramma
  - gli spostamenti (vettore di movimento) rispetto al fotogramma da cui dipende
  - una combinazione dei due



## Classificare i frame: *B-frames* (2)

- Inserire in un fotogramma informazioni che si riferiscono ad un fotogramma successivo è possibile solo **alterando l'ordine in cui i fotogrammi vengono archiviati** all'interno del file video compresso
- In genere richiedono meno bit per la codifica rispetto agli I-frame o ai P-frame.



Esempio:

## Memorizzazione di I/P/B-frames (1)

- Consideriamo la seguente sequenza di frame con i relativi macroblocchi:

f = frame, ma = macroblocco

f 1	f 2	f 3	f 4	f 5	f 6	f 7	f 8	f 9	f 10	f 11	f 12	f 13
I	B	B	P	B	B	P	B	B	P	B	B	I

i ma del f 1 sono tutti codificati come Intra f,  
i ma del f 2 cercheranno i blocchi simili nel f 1 (I) e nel f 4 (P)  
i ma del f 3 cercheranno i blocchi simili nel f 1 (I) e nel f 4 (P)  
i ma del f 4 cercheranno i blocchi simili nel f 1 (I)  
i ma del f 5 cercheranno i blocchi simili nel f 4 (P) e nel f 7 (P)  
i ma del f 6 cercheranno i blocchi simili nel f 4 (P) e nel f 7 (P)  
i ma del f 7 cercheranno i blocchi simili nel f 4 (P)  
i ma del f 8 cercheranno i blocchi simili nel f 7 (P) e nel f 10 (P)  
i ma del f 9 cercheranno i blocchi simili nel f 7 (P) e nel f 10 (P)  
i ma del f 10 cercheranno i blocchi simili nel f 7 (P).....

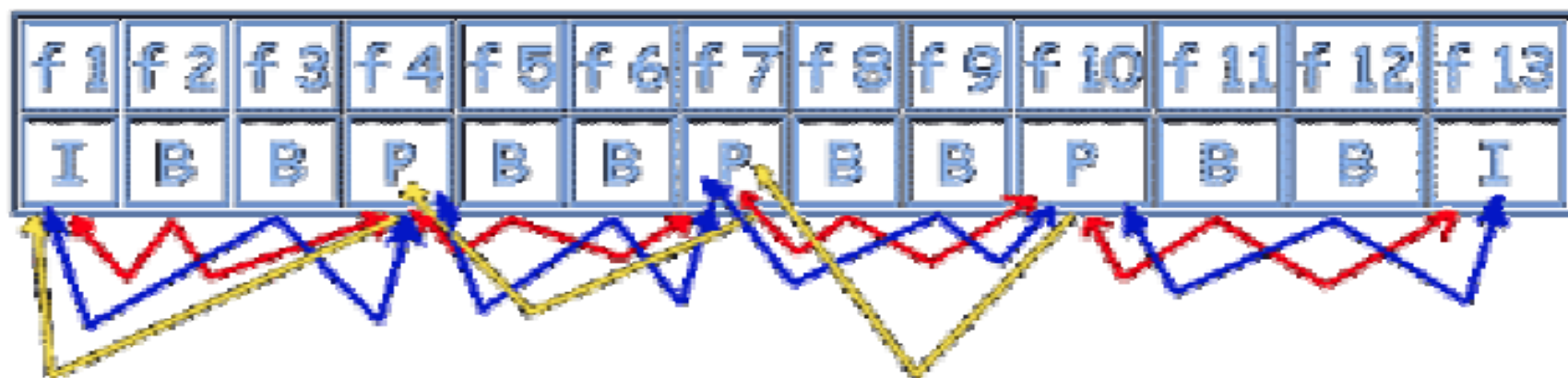




Esempio:

## Memorizzazione di I/P/B-frames (2)

- Ecco uno schema grafico:



- L'encoder e il decoder processano i frame in un ordine diverso da quello iniziale, calcolato in maniera automatica dai sw, in questo caso:

1	4	2	3	7	5	6	10	8	9	13	11	12
I	P	B	B	P	B	B	P	B	B	I	B	B



# Vecchi formati: **MPEG-1** (1)

- L'obiettivo di questo formato era il raggiungimento della **qualità VHS**
- Per ottenerla, il video è codificato
  - per quanto riguarda la **luminosità** a **352x288** pixel
  - mentre per quanto riguarda il **colore** l'immagine è ulteriormente divisa per due ed è pertanto codificata a **176x288**



## Vecchi formati: **MPEG-1** (2)

- Il **Codec MPEG-1** effettua una serie di operazioni di compressione delle immagini che sfruttano non solo la **trasformata DCT**, ma anche le differenze tra un fotogramma e l'altro
- Anzichè memorizzare tutti i fotogrammi per intero, se ne memorizzano soltanto alcuni come tali ad intervalli prefissati e regolari (*I-frames*)
- Tra di essi ci si limita a memorizzare una serie di frames incompleti nei quali vengono "scritte" solo le informazioni che subiscono delle variazioni rispetto alle immagini precedenti (*P/B-frames*)



# Vecchi formati: **MPEG-2** (1)

- L'obiettivo di questo formato era quello di essere **flessibile ed adatto a varie applicazioni**
  - In grado anche di **codificare in digitale le immagini con una qualità equivalente a quella analogica** (corrispondente alla qualità delle trasmissioni televisive)
  - e **l'audio con quella cinematografica**, utilizzando flussi di dati fino a 60 Mbit/sec



## Vecchi formati: MPEG-2 (2)

- La caratteristica principale dell'MPEG-2 è la sua **scalabilità**
  - Ossia la possibilità di creare soluzioni di codifica e decodifica più o meno complesse **in base al tipo di prodotto da realizzare**, aggiungendo poi altre caratteristiche
    - quali la possibilità di trasmettere il flusso multimediale su reti a banda larga
    - assicurare una buona robustezza nei confronti degli errori della rete
    - il trasporto parallelo di molteplici canali audio
    - le funzioni di protezione e di controllo di accesso al flusso
    - ...



## Vecchi formati: MPEG-2 (3)

- Per consentire all'industria di procedere gradualmente con l'implementazione dello standard, il comitato di lavoro dell'MPEG ha definito una **serie di livelli e di profili** in base ai quali ogni soluzione tecnica può essere sviluppata e verificata
- Non tutte le combinazioni portano ad un sottoinsieme di specifiche valide



# Vecchi formati: MPEG-2 (4)

- Cinque profili e quattro livelli sono combinati in 11 soluzioni ufficiali
- I **profili** sono:
  - ❑ Simple (SP)
  - ❑ Main (MP),
  - ❑ SNR Scalable
  - ❑ Spatial Scalable
  - ❑ High
- I **livelli** sono:
  - ❑ Low (LL)
  - ❑ Main (ML)
  - ❑ High1440 (H-14)
  - ❑ High (HL)

Profilo	Livello	Pixel Orizzontali	Pixel Verticali	Frame Rate Max	Bitrate Max (Mbit/sec)
Simple	main	720	576	30	15
Main	low	352	288	30	4
Main	main	720	576	30	15
Main	high 1440	1440	1152	60	60
Main	high	1920	1152	60	80
SNR Scalable	low	352	288	30	3(4)
SNR Scalable	main	720	576	30	10(15)
Spatially Scal.	high 1440	720 (1440)	576 (1152)	30 (60)	15(40 o 60)
High	main	352 (720)	288 (576)	30 (30)	4 (15 o 20)
High	high 1440	720 (1440)	576 (1152)	30 (60)	20 (60 o 80)
High	high	960 (1920)	576 (1152)	30 (60)	25 (80 o 100)



# Uno sguardo a MPEG-4 (1)

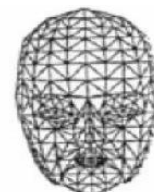
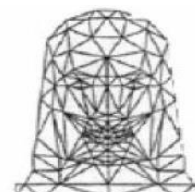
- L' MPEG-4 usa fondamentalmente lo stesso algoritmo di compressione di MPEG-1 e MPEG-2, ma in modo molto più efficiente
  - I/P/B-frames
  - Profili e Livelli
  - 32 “Parti”
    - Ogni parte descrive il funzionamento specifico di un sottostandard (ad es.: multiplexing audio-video, procedure di testing, descrizione della scena, compressione dei font nei fotogrammi, ...)





## Uno sguardo a **MPEG-4** (2)

- Una funzionalità aggiuntiva è che **il sistema riesce a distinguere i vari livelli di un'immagine**: lo sfondo e i primi piani
  - Se lo sfondo rimane uguale allora nei fotogrammi successivi non verrà memorizzato, risparmiando così prezioso spazio
- Inoltre è possibile elaborare queste immagini più semplicemente, estrapolando gli attori o gli oggetti dallo sfondo con grande facilità
  - Il *Virtual Reality Modeling Language* (**VRML**) permette di rappresentare in formato testuale mesh 3D





## Uno sguardo a **MPEG-4** (3)

- Dal 2002, MPEG-4 sfrutta al suo interno **QuickTime** (v. 6 o superiore) di Apple per aumentare le potenzialità di trasmissione dei dati via web
  - QuickTime è un formato file “contenitore”
    - Può raccogliere audio, immagini, video e altri dati multimediali (ad esempio flash), organizzandoli in una struttura ad albero
  - Dal 1994 QuickTime (v2) è compatibile anche con la piattaforma Microsoft



## H.264: Premessa

- H.264 è un formato di compressione video rilasciato nel 2003
- H.264 è noto anche come **(MPEG-4) AVC** (*Advanced Video Coding*), parte 10 dei sottostandard di MPEG-4
- H.264 applica principalmente un metodo di **compressione lossy** (con perdita), anche se è possibile rendere talmente impercettibile la perdita da considerarlo **lossless**



# H.264: Miglioramenti

- Riassunto dei miglioramenti apportati rispetto ad MPEG-2/4:
  - Miglioramento della codifica entropica, tramite una codifica a lunghezza variabile
  - Applicazione della trasformata DCT su **blocchi più piccoli**
  - Miglioramenti relativi alla valutazione e alla compensazione del movimento
  - Filtro di ricostruzione nella fase di decodifica per ridurre l'effetto di blocchettizzazione



# H.264: Codifica VLC

- I simboli che rappresentano i parametri relativi ai modi di codifica e predizione, i vettori movimento e i coefficienti della trasformata vengono codificati con **codici a lunghezza variabile** (Codifica *Variable Length Coding* – **VLC**)
  - VLC è basata su tabelle di assegnazione da trasmettere insieme ai frame
  - VLC sfrutta le **sequenze di zero, +1 e -1**, e la correlazione fra il numero di coefficienti non nulli di un blocco e quello nei blocchi adiacenti



# H.264: Blocchi e Macroblocchi

- I **blocchi** sono costituiti da 4x4 pixel
  - La dimensione dei blocchi è ridotta a  $1/4$  rispetto a MPEG-2
- I **macroblocchi**
  - Hanno una dimensione di 16x16 campioni per la luminanza e di
  - 8x8 campioni, rispettivamente per ciascuna delle due componenti della cromaticanza



## H.264: Slice (1)

- I macroblocchi sono organizzati in **slice**
- Una slice è un sottoinsieme di immagine decodificabile indipendentemente dalle altre
- L'ordine di trasmissione delle slice (e quindi dei macroblocchi) non è necessariamente quello originario nell'immagine, ma
  - ❑ è indicato dal codificatore in una apposita mappa (**Macroblock Allocation Map**).



## H.264: Slice (2)

- Sono definiti **5 differenti tipi di slice**:
  - I primi tre, analogamente a quanto visto per MPEG-2, sono I (intra), P (predictive) e B (bi-predictive) e le predizioni sono ottenute a partire dalle slice precedentemente codificate
    - Nota: ciascuna frame precedentemente poteva fare riferimento al più ad altri 2 frame
    - In H.264 più slice possono essere utilizzate per le predizioni e pertanto codificatore e decodificatore memorizzano le slice utilizzate per le predizioni in una apposita memoria (**multipicture buffer**) e il controllo per la gestione del buffer è specificato nel flusso dati





## H.264: Slice (3)

- In H.264 sono inoltre stati definiti due ulteriori tipi di slice, denominati **SI (Switching I)** e **SP (Switching P)** che consentono un'**efficiente commutazione fra flussi di dati a bit-rate differente**, senza rinunciare al massimo sfruttamento della ridondanza temporale
  - Nelle applicazioni di streaming via internet spesso lo stesso video è codificato a differenti bitrate ed il decoder tenta di accedere al flusso a bit-rate più elevato, che fornisce una più elevata qualità, ma
  - se le condizioni del canale non lo permettono, commuta al flusso a bit-rate più basso.



## H.264: Predizione Codifica Intra

- ❑ Nella codifica Intra è sfruttata la sola **correlazione spaziale all'interno dello stesso macroblocco**
- ❑ Per aumentare l'efficienza **vengono codificate le differenze fra i pixel del macroblocco e i pixel precedentemente codificati**, tipicamente quelli posizionati sopra e a sinistra



## H.264: Predizione Codifica Inter (1)

- Nella codifica Inter si sfrutta la **correlazione temporale fra uno o due quadri precedentemente codificati**
- La predizione può essere ottenuta mediante una stima ed una compensazione del movimento (*motion compensated prediction*)

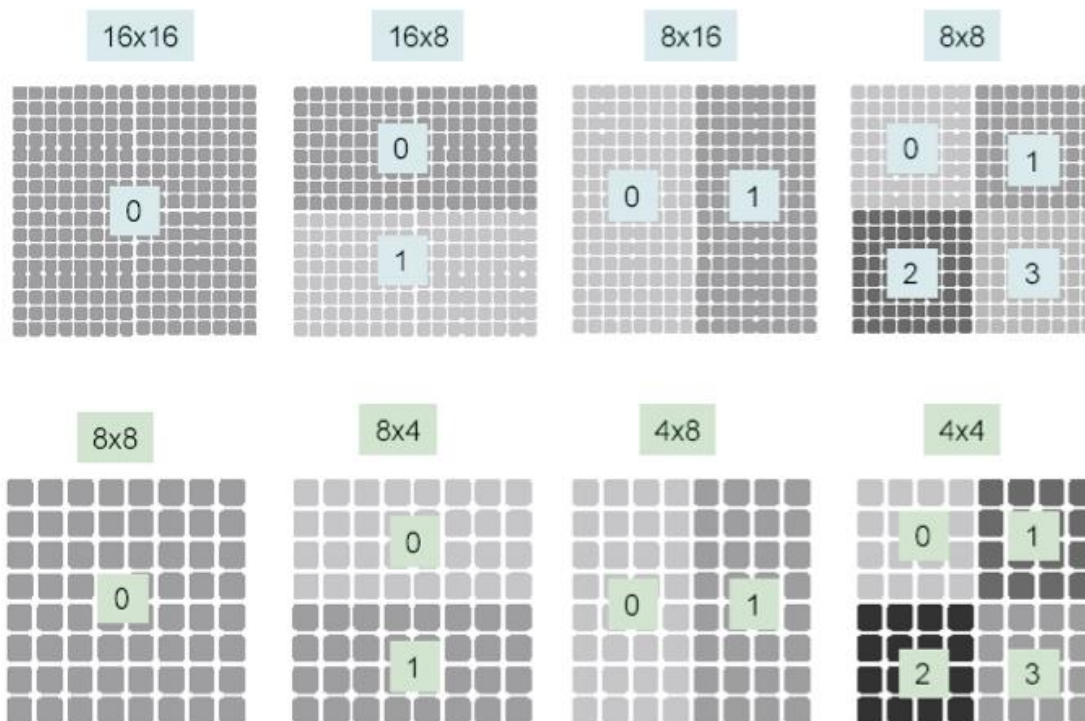


# H.264: Predizione Codifica Inter (2)

- A differenza degli standard precedenti, la dimensione del blocco su cui si effettua la predizione può variare da 16x16 fino a 4x4

**Tree Structured  
Motion Compensation**  
*Segmentazione dei macroblocchi  
(luminanza) al fine della  
compensazione del movimento  
prevede i 4 modi indicati nella parte  
alta della figura.*

*Se è scelto il modo 8x8 ciascuna  
delle 4 partizioni possono essere  
ulteriormente suddivise in  
sottopartizioni, come indicato in  
basso nella figura*





## H.264: Predizione Codifica Inter (3)

- Questo metodo di partizionare i macroblocchi in sottoblocchi è denominato **Tree Structured Motion Compensation**
- In fase di codifica sono possibili molteplici scelte che hanno implicazioni differenti sul numero di bit necessario a codificare i vettori movimento e le differenze residue
  - In genere dimensioni elevate del blocco sono convenienti in aree piatte, mentre in aree ricche di dettagli si può trarre vantaggio dall'uso di aree ridotte



## H.264: Trasformata DCT

- Il tipo di trasformata adottato è basato sulla **DCT** (Discrete Cosine Transform), ma
- sono state apportate delle modifiche affinché le operazioni richiedano somme e scalamenti effettuabili con **numeri interi a 16 bit in modo da non avere perdita di precisione** effettuando la trasformazione diretta seguita da quella inversa



# H.264: Quantizzazione

- Esistono **52 livelli di quantizzazione** e questa ampia gamma di valori permette al codificatore di raggiungere il miglior compromesso fra qualità e bitrate
- Pertanto, la gamma della scala di quantizzazione è **0-51**:
  - dove 0 è lossless, 23 è di default, e 51 è il valore peggiore possibile totalmente lossy
- Un valore più basso corrisponde a una qualità superiore e un valore accettabile si aggira intorno al range **18-28**
- Consideriamo il valore 18 come un valore visivamente senza perdita o quasi:
  - dovrebbe apparire lo stesso input ma effettivamente non è tecnicamente senza perdita



# H.264: Filtro antibloccettizzazione

- **L'effetto di bloccettizzazione** è uno dei degradamenti caratteristici delle tecniche di compressione che operano su blocchi: è particolarmente visibile e fastidioso
  - Lo avevamo già incontrato nella codifica JPEG
- H.264 introduce un filtro apposito che è applicato prima della trasformata inversa sia nel codificatore, sia nel decodificatore
- Si ottengono due principali vantaggi:
  - una minore visibilità dei bordi dei blocchi
  - una migliore predizione inter con compensazione del movimento





# H.264: Applicazioni

- H.264 trova spazio in differenti scenari, soprattutto grazie alla possibilità di aumentare il bit-rate (ad es.: a parità di qualità, il bit-rate rispetto a MPEG-2 è quasi il doppio!)
  - HDTV
    - nelle trasmissioni televisive digitali in HD
    - nelle trasmissioni che prevedono risoluzioni elevate (come 4K o UHD TV)
  - Web Streaming in HD
  - Blu-ray Disc