

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Analisi sperimentale degli standard di codifica video H.265/266 in modalità Intra

Relatore:

PROF. MARCO CAGNAZZO

Laureando:

MATTEO ZANELLA

2000139

Anno Accademico 2022/2023

Data di Laurea 27/09

Abstract

Nel corso degli anni, è aumentato esponenzialmente l'utilizzo dei contenuti multimediali in rete, sia per applicazioni in streaming che on-demand.

I contenuti video rappresentano una grande fetta del traffico online, come mostrato dal CISCO Annual Internet Report; gli standard di codifica video H.265 e H.266 hanno portato alla luce nuove tecniche per la codifica video.

Sebbene le nuove tecnologie introdotte siano molte, nel corso di questa tesi ci si soffermerà nello studiare le tecniche di codifica Intra introdotte, per poi analizzare le loro prestazioni sperimentalmente, in quanto queste tecniche sono utili non solo per la codifica video, ma anche per la codifica di immagini.

Verranno messi a confronto i due standard utilizzando la suite FFmpeg, analizzando vari parametri, tra cui bit-rate e PSNR dei video.

Indice

Introduzione	1
1 Analisi metodologica dei modi Intra negli standard H.265 e H.266	5
1.1 Standard H.265	5
1.1.1 Predizione angolare	5
1.1.2 Predizione planare	6
1.1.3 Filtro di smoothing	6
1.2 Standard H.266	7
1.2.1 Predizione angolare di granularità fine	7
1.2.2 Combinazione di predizione intra dipendente dalla posizione (PDPC)	7
1.2.3 Linee di riferimento estese	8
2 Test di codifica	9
2.1 Suite FFMPEG	9
2.1.1 Installazione	9
2.1.2 Integrazione del codec vvc	9
2.1.3 Utilizzo	10
2.2 Dataset video	10
2.3 Test di codifica effettuati	11
2.3.1 Parametri utilizzati	11
2.4 Valutazione qualità dei video	13
2.5 Metrica di Bjøntegaard	14
3 Analisi risultati	15
3.1 Test su video a bassa risoluzione	15
3.1.1 Prestazioni per bit-rate richiesti	15
3.1.2 Metrica di Bjøntegaard	19
3.2 Test su video HD	20
3.2.1 Bit-rate raggiunti	20
3.2.2 Qualità visiva	21
3.2.3 Metrica di Bjøntegaard	23
3.3 Test su video 4k	24
3.3.1 Bit-rate raggiunti	24
3.3.2 Prestazioni per bit-rate effettivi	24

3.3.3	Metrica di Bjøntegaard	25
4	Considerazioni finali	26
5	Bibliografia	27
5.1	Articoli	27
5.2	Collegamenti ipertestuali	28

Introduzione

Negli ultimi decenni, i contenuti video hanno guadagnato sempre più popolarità in rete, creando quindi la necessità di nuovi standard di codifica video; c'era, e c'è tutt'ora, la necessità di processare e trasmettere video in tempo reale, aumentando quindi il bisogno di standard efficienti per trasmettere grandi quantità di dati tramite mezzi non ottimali. Come ci viene esposto dal CISCO Annual Internet Report [5], il numero di persone che utilizzano un servizio cellulare passerà da 5.1 miliardi nel 2018 a 5.7 miliardi nel 2023 e il numero di tv con qualità superiore (UHD, 4K), passerà da 268 a 891 milioni. Dispositivi con una qualità così alta hanno bisogno di un bit-rate maggiore, passando quindi da una media di 2Mbps ad una di 15/18Mbps, bit-rate raggiunti grazie all'introduzione di nuovi standard di codifica video, come i due soggetti di questa tesi. Per mostrare un video HD (1920×1080) con un frame rate di 30 fps e 8 bit per canale, senza usare tecniche di compressione, sarebbe necessario un bit-rate di 1.5Gbps, contro i 5Mbps di connessione suggerita da Netflix [L5].

Nel 2003 è stata completata la stesura dello standard di codifica video H.264, il quale fungerà da base per i successivi standard H.265 (2013) e H.266 (2017).

Quando si parla di vari standard di codifica video, sebbene ognuno abbia le proprie tecniche, i concetti base sono gli stessi:

- **Spazi di colore:** uno spazio di colore è uno spazio tridimensionale in cui, tramite vettori tridimensionali, è possibile rappresentare ogni colore.

Lo spazio di colore più conosciuto è lo spazio RGB, che utilizza come coordinate la quantità di rosso, verde e blu per comporre il colore desiderato.

Per quanto riguarda la codifica di immagini, è comunemente utilizzato lo spazio YCbCr, ovvero lo spazio composto dalla componente luma, indicante la luminosità, la componente crominanza blu e la componente crominanza rossa, che indicano l'intensità dei rispettivi colori.

Questo spazio di colore è utilizzato spesso, nella codifica delle immagini, in quanto l'occhio umano è poco sensibile ai cambiamenti nelle due componenti della crominanza, di conseguenza è possibile diminuire la precisione di queste due componenti (sottocampionamento), diminuendo il numero di bit per pixel necessari per rappresentare l'immagine.

Per rappresentare un pixel, quindi, utilizzando uno spazio RGB, vengono utilizzati 8 bit per ogni componente, arrivando ad un totale di 24 bit; utilizzando lo spazio YCbCr, con sottocampionamento 4:2:0, riusciamo a rappresentare ogni pixel, in media, con solo 12 bit: il sottocampionamento effettuato è spaziale, di conseguenza i valori di cromianza, rappresentati sempre da 8 bit, sono condivisi da 4 pixel.

- **Suddivisione in blocchi:** ogni immagine viene suddivisa nelle sue unità di codifica, dette blocchi.

I blocchi hanno dimensione variabile, in base al contenuto del blocco, dalla qualità desiderata e dalla tecnica di ottimizzazione RD utilizzata.

I blocchi, ad esempio, hanno dimensioni piccole quando una o più di queste condizioni sono verificate (dettagli fini, contorni irregolari o qualità richiesta alta).

- **Predizione Spaziale:** all'interno delle immagini che compongono un video, spesso si possono riconoscere pattern o zone contigue simili: questo fatto è indicato come ridondanza spaziale.

Le tecniche di predizione spaziale hanno lo scopo di sfruttare questa ridondanza spaziale, riuscendo a codificare con meno bit, le informazioni contenute in un gran numero di blocchi. raggiunto

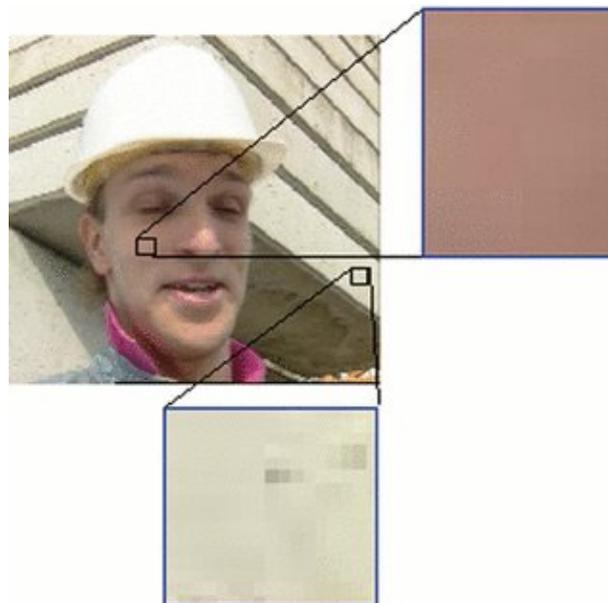


Figura 1: Ridondanza spaziale in un immagine

- **Predizione Temporale:** oltre alla ridondanza spaziale, esiste anche la ridondanza temporale, ovvero la propagazione nel tempo di elementi delle immagini all'interno del video.

Una tecnica che permette di sfruttare questo tipo di ridondanza è la stima del movimento: riuscire a rappresentare il movimento di un elemento nel video, tramite dei vettori, in modo da poter derivare il contenuto di un'immagine da altre immagini, adiacenti nel tempo.

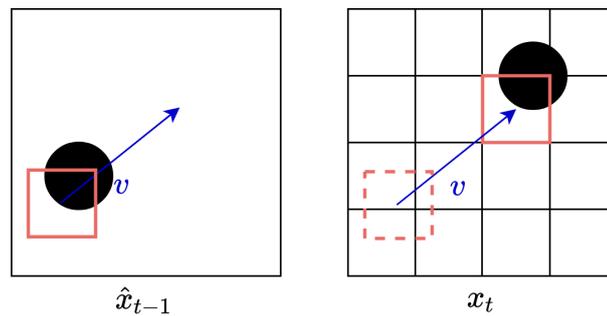


Figura 2: Rappresentazione di un MV [L1]

- **Frame Inter e Intra:** i frame di un video vengono suddivisi in due tipi di frame: frame Inter e frame Intra.

I frame di tipo Intra, utilizzano tecniche di predizione spaziale, o tecniche Intra per codificare il proprio contenuto, mentre i frame di tipo Inter, utilizzano tecniche di predizione temporale, o tecniche Inter.

I frame di tipo Intra riescono mediamente a raggiungere una qualità superiore, in quanto sono codificati come immagini singole, mentre frame di tipo Inter, in quanto dipendenti da frame adiacenti, sono più sensibili agli errori che si possono propagare nel tempo, da un frame Inter all'altro.

Tuttavia, frame di tipo Intra, raggiungono dimensioni maggiori rispetto ai frame di tipo Inter, in quanto non possono sfruttare la ridondanza temporale presente nel video.

Ottimizzare la codifica dei frame Intra permette quindi di aumentare l'efficienza di codifica globale di uno standard.

- **GOP:** Un GOP, o Group Of Picture, è un gruppo di frame sequenziali. Il primo frame di un GOP è un frame Intra, utilizzato come ancora per gli altri frame del GOP, che possono essere frame di tipo Inter o Intra (che formeranno nuove ancore all'interno del GOP).

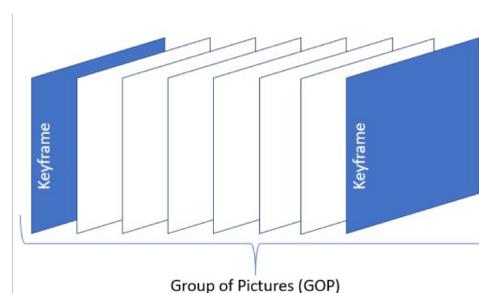


Figura 3: Rappresentazione di un GOP

- **Trasformata:** in un'immagine non compressa, ogni valore è importante in quanto ogni pixel ha la stessa importanza.

La trasformata è un processo che permette di rendere sparsa l'informazione di un'immagine, o di un blocco, permettendo di raggruppare le informazioni importanti in pochi valori.

Grazie alla trasformata, è possibile rappresentare l'informazione di un'immagine utilizzando pochi valori importanti e tanti valori poco importanti i quali potranno essere scartati senza diminuire visibilmente la qualità visiva.

- **Quantizzazione:** la quantizzazione a cui si fa riferimento consiste nel diminuire il numero di bit su cui sono rappresentati i valori diminuendo la precisione dell'informazione e risparmiando bit.

Capitolo 1

Analisi metodologica dei modi Intra negli standard H.265 e H.266

1.1 Standard H.265

Lo standard di codifica video H.265, o più comunemente chiamato HEVC (High Efficiency Video Coding), è stato finalizzato nel 2013 con lo scopo di migliorare l'efficienza della codifica sui contenuti ad alta risoluzione, tenendo bassa la complessità, in modo da poter operare su un'ampia gamma di applicazioni.

La codifica Intra in questo standard utilizza varie tecniche, tra cui:

- **Predizione angolare:** permette di modellare efficientemente varie strutture direzionali, spesso presenti nelle immagini;
- **Predizione planare:** permette di modellare zone omogenee nello spazio;
- **Filtro di smoothing:** riduce gli artefatti generati ai bordi dei blocchi

È possibile trovare un'analisi più dettagliata delle tecniche Intra dello standard HEVC all'interno dell'articolo [1].

1.1.1 Predizione angolare

Questa tecnica consiste nel copiare i pixel della riga sovrastante o della colonna alla sinistra, lungo una data direzione in modo da rappresentare efficientemente contorni o linee presenti nell'immagine.

La predizione angolare era già presente nello standard precedente, ma con molte limitazioni: lo standard HEVC riutilizza alcuni concetti dello standard H.264 applicando qualche miglioria.

Il numero di direzioni con cui si può effettuare la predizione passa da 8 a 33, e la dimensione dei blocchi su cui viene effettuata la predizione può variare da 4×4 a 32×32 .

Nella maggior parte delle immagini, patter verticali e orizzontali sono più frequenti di

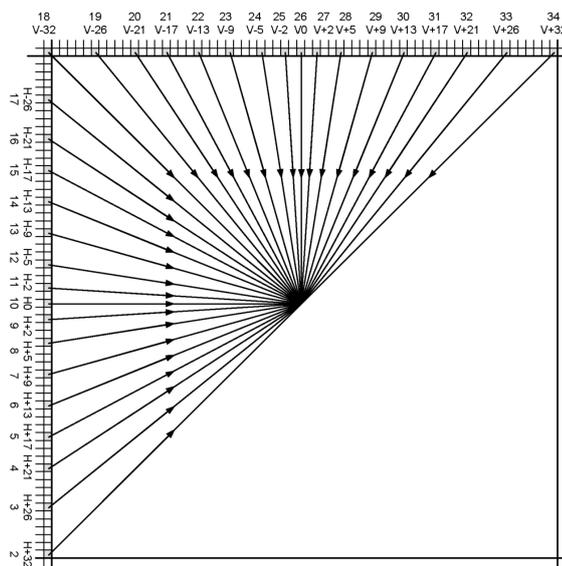


Figura 1.1: Direzioni disponibili nella predizione angolare

altri pattern: viene utilizzato un set di 33 predizioni angolari ottimizzato tenendo conto di questo fatto.

Vengono utilizzati dei piccoli cambiamenti di pendenza in prossimità di angolazioni verticali e orizzontali in modo da poter rappresentare con più accuratezza questi pattern.

1.1.2 Predizione planare

La predizione angolare fornisce una buona predizione in presenza di bordi, tuttavia è necessario sfruttare un altro tipo di predizione per quanto riguarda zone uniformi.

In queste aree dell'immagine, viene utilizzata la predizione planare, che consiste nell'approssimare i valori di luminosità di un blocco tramite una funzione $f(x) = ax + by + c$: i coefficienti sono determinati con un'approssimazione ai minimi quadrati, rispetto ai pixel dei blocchi adiacenti (riga sovrastante e colonna a sinistra).

1.1.3 Filtro di smoothing

Quando si fanno operazioni su blocchi contigui, si rischia di generare dei bordi evidenti tra due blocchi: il filtro di smoothing si occupa di rendere liscio il bordo tra i blocchi.

Vengono confrontati i valori dei pixel in prossimità dei bordi, e in base ai risultati si decide se applicare o no il filtro.

Il filtro di smoothing viene applicato anche nelle zone in cui è stata effettuata la predizione planare, in modo da rendere lisce le zone omogenee.

Siccome si è notato che le predizioni nelle componenti Cb e Cr non presentano tanti artefatti, il filtro non viene applicato a queste due componenti in modo da ridurre la complessità.

Ci sono inoltre altre tecniche, mirate all'ottimizzazione della codifica, tra cui:

- Utilizzare una lista di predizioni probabili (MPM) costruita analizzando i blocchi adiacenti, in modo da poter codificare solo l'indice della lista, invece che l'intera predizione
- Le predizioni effettuate nelle componenti croma, spesso seguono le predizioni della componente luma; la predizione di un blocco di componenti croma, quindi, viene effettuata sulla base della predizione del blocco rispettivo nella componente luma

1.2 Standard H.266

Lo standard di codifica H.266, o VVC (Versatile Video Coding), si evolve dallo standard HEVC con lo scopo di ottimizzare la codifica per una più ampia gamma di contenuti video, tra cui video immersivi (360°), a qualità superiore (HDR, 4K), etc.

In questo standard vengono implementate molte migliorie, soprattutto riguardanti il partizionamento delle immagini.

Un'altra importante innovazione è quella dell'accesso casuale nel video: questa innovazione richiede di ottimizzare la codifica per le immagini Intra, che fungono da ancora per le immagini Inter, in quanto l'accesso casuale comporta l'accesso ad immagini Intra, se non si vuole perdere qualità all'inizio del video.

È possibile trovare un'analisi più dettagliata delle tecniche Intra dello standard VVC all'interno dell'articolo [2].

1.2.1 Predizione angolare di granularità fine

Per ogni dimensione di blocco, ci sono 65 diverse direzioni per la predizione angolare, definiti da un angolo di 45° fino ad un angolo di 225°

Questo standard gestisce anche blocchi che non sono di forma quadrata, aumentando il numero di direzioni per la predizione angolare da 65 a 93, utilizzando un range di predizione più ampio per il lato corto.

1.2.2 Combinazione di predizione intra dipendente dalla posizione (PDPC)

I modi intra dello standard HEVC utilizzano la combinazione convenzionale di codifica predittiva e trasformata, in cui la predizione si basa su singole linee di pixel di riferimento, a partire da blocchi precedentemente ricostruiti.

Questo approccio si è dimostrato abbastanza efficace, ma uno dei suoi limiti è che la stessa forma di predizione viene utilizzata per blocchi di dimensioni che possono variare da 4x4 a 64x64, e mentre la predizione direzionale esatta può funzionare bene nei blocchi piccoli, è molto meno efficace nei blocchi più grandi.

Una possibile soluzione è l'applicazione di un insieme di filtri ricorsivi in quanto può fornire il controllo desiderato: risultati sperimentali hanno dimostrato che in questo modo è

possibile migliorare significativamente l'efficienza della compressione.

Tuttavia, al giorno d'oggi, con l'hardware personalizzato e processori generici, le applicazioni video utilizzano il calcolo parallelo, pertanto è vantaggioso utilizzare tecniche che consentano di prevedere simultaneamente un gran numero di pixel di blocco: sotto questa considerazione questo filtraggio ricorsivo perde di utilità.

La PDPC ha lo scopo di risolvere questo problema mantenendo una bassa complessità e un'efficiente implementazione parallela, utilizzando una semplice combinazione ponderata di riferimenti, dove i pesi dipendono dalla posizione dei pixel, dalla modalità di predizione e dalla dimensione del blocco.

Per maggiori dettagli riguardo alla PDPC, si faccia riferimento all'articolo [3]

1.2.3 Linee di riferimento estese

Per effettuare predizioni di tipo Intra, possono essere utilizzati dei blocchi non adiacenti, fino a 2 o 3 linee di blocchi di distanza.

Dal momento che l'uso delle linee di riferimento estese è vantaggioso solo per i contenuti nitidi, combinare tali linee con la modalità planare, la riduzione del campionamento delle reference e PDPC non è consentito.

Inoltre, vengono applicate le seguenti migliorie:

- La predizione intra viene migliorata ulteriormente utilizzando una **variante a bassa complessità di una rete neurale**
- **Modalità probabile estesa (MPM)**: la tecnica Most Probable Mode viene ereditata dallo standard HEVC, applicando migliorie per tenere conto del maggior numero di tipi di predizione

Capitolo 2

Test di codifica

2.1 Suite FFMPEG

FFmpeg è una suite software completa per registrare, convertire e riprodurre audio e video che si basa su libavcodec, libreria per la codifica audio/video [L2].

2.1.1 Installazione

L'installazione di ffmpeg è semplice, è sufficiente clonare la repository e compilare la libreria utilizzando cmake:

```
git clone https://git.ffmpeg.org/ffmpeg.git ffmpeg
```

o più semplicemente, installando direttamente l'eseguibile ffmpeg:

Su linux:

```
sudo apt install ffmpeg
```

Per semplicità, d'ora in avanti, quando saranno illustrati comandi da terminale, sarà sottinteso che si fa riferimento a comandi utilizzabili su un sistema operativo Linux.

2.1.2 Integrazione del codec vvc

La suite ffmpeg, al momento di scrittura di questo elaborato, ancora non supporta l'utilizzo di un codec vvc; esiste tuttavia un'integrazione creata dall'istituto delle telecomunicazioni Fraunhofer che permette di utilizzare il codec vvc all'interno di ffmpeg.

L'integrazione utilizzata è una patch, non un'implementazione ufficiale, di conseguenza è possibile che alcuni risultati siano contaminati da questo fatto.

Nonostante ciò, è stata scelta questa patch in modo da poter ridurre i tempi di esecuzione e rendere più semplice l'esecuzione e preparazione dei test.

Per installare l'integrazione del codec vvc, il lettore è invitato a seguire la procedura guidata indicata dalla wiki della pagina github, dedicata all'integrazione di vvc nella suite ffmpeg, dell'istituto delle telecomunicazioni Fraunhofer [L3].

Per lo scopo di questo elaborato, è necessario poter usare sia lo standard hevc che lo standard vvc, quindi durante la procedura di integrazione, invece di

```
./configure --enable-pthreads --enable-pic --enable-shared
--enable-rpath --arch=amd64 --enable-demuxer=dash
--enable-libxml2 --enable-libvavdec --enable-libvavenc
```

sarà necessario scrivere

```
./configure --enable-pthreads --enable-pic --enable-shared
--enable-rpath --arch=amd64 --enable-demuxer=dash
--enable-libxml2 --enable-libvavdec --enable-libvavenc
--enable-gpl --enable-libx265
```

per poter configurare ffmpeg all'utilizzo non solo del codec vvc, ma anche del codec hevc (libx265).

2.1.3 Utilizzo

Tra le funzionalità che la suite offre, saranno utilizzate le seguenti:

- **ffmpeg**: Comando utilizzato per convertire video tra formati diversi
- **ffprobe**: Comando per ottenere informazioni riguardo a un video

2.2 Dataset video

Per effettuare i test, è stato utilizzato il dataset fornito dal sito Xiph.org [L4]. Questo dataset contiene video nel formato y4m: i video in questo formato sono video non compressi che utilizzano lo spazio di colore YCbCr e presentano degli header contenenti informazioni come bit-rate, durata del video, ecc.

VIDEO NAME	RESOLUTION	FRAME-RATE	COLOR SPACE
akiyo	352x288	30fps	yuv420p
claire	176x144	30fps	yuv420p
deadline	352x288	30fps	yuv420p
ice	704x576	60fps	yuv420p
controlled_burn	1920x1080	30fps	yuv422p
CSGO	1920x1080	60fps	yuv420p
ducks_take.off	1920x1080	50fps	yuv420p
Four_people	1280x720	60fps	yuv420p
Netflix_BarScene	4096x2160	60fps	yuv420p10le
Netflix_RollerCoaster	4096x2160	60fps	yuv420p10le
Netflix_FoodMarket	4096x2160	60fps	yuv420p10le

Tabella 2.1: Video presenti nel dataset utilizzato

Nella tabella 2.1 sono indicati alcuni dei video presenti nel dataset utilizzato: sono stati utilizzati video che partono da una bassa risoluzione fino ad arrivare a video 4K a 60fps.

2.3 Test di codifica effettuati

Per realizzare un confronto equo, tra codec diversi, sono stati utilizzati gli stessi parametri per entrambi i codec: stessi parametri di quantizzazione e stessi bit-rate richiesti. Sono stati utilizzati i seguenti comandi ffmpeg per confrontare i due codec:

```
ffmpeg -i <input_file> -an -vcodec <codec>
      -b:v <bit-rate> -g 1 <output_file>
```

```
ffmpeg -i <input_file> -an -vcodec <codec>
      -qp <passo_di_quantizzazione> -g 1 <output_file>
```

Il primo comando è stato utilizzato per misurare le prestazioni dei due codec nelle situazioni in cui è richiesto un bit-rate specifico, mentre il secondo comando è stato utilizzato per confrontare le qualità raggiunte a parità di passo di quantizzazione.

- -i <input_file>: file di input da utilizzare
- -an -vcodec <codec>: codec da utilizzare
- -b:v <bit-rate>: bit-rate richiesto
- -qp <passo_di_quantizzazione>: passo di quantizzazione richiesto
- -g 1: dimensione della GOP (con dimensione 1, tutti i frame saranno codificati come Intra)
- <output_file>: file in cui salvare il video generato

2.3.1 Parametri utilizzati

Per i video a bassa risoluzione, sono stati utilizzati bit-rate tra i 100kbps e i 6Mbps: è stato osservato che per bit-rate superiori ai 6Mbps, le qualità raggiunte erano troppo alte in quanto, superato un PSNR di 40, la qualità visiva non subisce modifiche percettibili. Per video ad alta risoluzione non è stato possibile utilizzare valori di bit-rate come parametri in quanto, lavorando solo con tecniche intra, i bit-rate richiesti per avere performance significative erano oltre la soglia di bit-rate che la suite FFmpeg poteva offrire. Di conseguenza sono stati ricavati i bit-rate dei video generati dai test sui passi di quantizzazione.

Per quanto riguarda il parametro dedicato alla quantizzazione, sono stati scelti i valori utilizzati dal comitato di standardizzazione per lo sviluppo di HEVC e VVC [4]: 22, 27, 32, 37, 42.

Valutare le prestazioni di qp troppo bassi (ovvero che si avvicinano a compressioni di tipo loss-less) non è di nostro interesse, in quanto non rispecchiano le richieste comuni presenti in rete: situazioni in cui sono necessarie compressioni loss-less possono essere nel campo medico o forense, e in questa tesi si andrà a trattare principalmente di applicazioni come contenuti in streaming o media di uso comune.

Su ffmpeg, specificare il preset da utilizzare serve a dire al codec scelto la velocità con cui deve operare, a discapito della qualità del video codificato.

Effettuando vari test con preset diversi, è stato notato che la qualità dei video codificati non cambiava: la scelta del preset influenza il funzionamento del crf (Constant Rate Factor), parametro che non è stato utilizzato per l'esecuzione di questi test in quanto non supportato dalla patch utilizzata per integrare il codec vvc.

Facendo dei test a riguardo è stato notato come i risultati non sembrano variare:

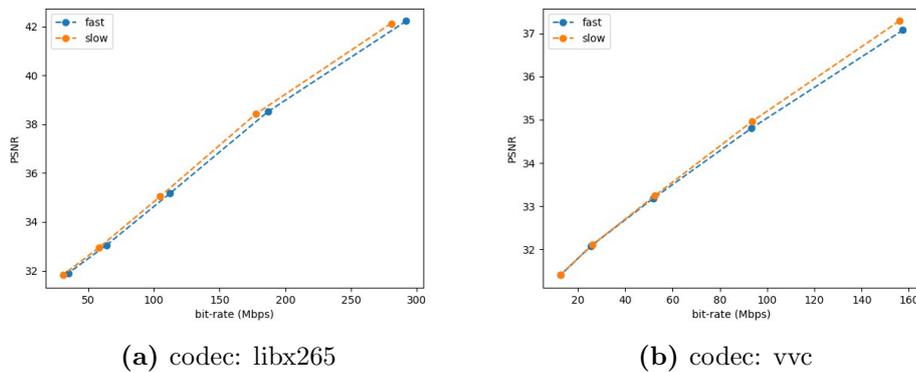


Figura 2.1: Test effettuati sul video HD park_joy

Inoltre, andando ad analizzare i tempi d'esecuzione medi di codifica di un video di 2 secondi con un frame-rate di 50fps, si ottiene una disparità di tempo necessario che aumenterebbe esponenzialmente il tempo di computazione.

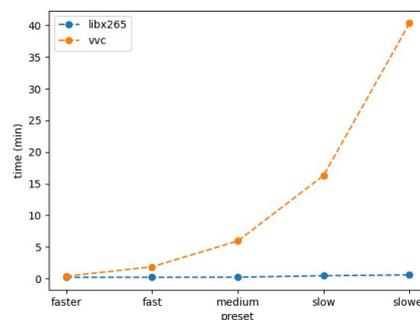


Figura 2.2: Test effettuati sul video HD park_joy

La differenza nei tempi di esecuzione è da attribuire principalmente alla patch utilizzata per effettuare i test, piuttosto che allo standard.

Alla luce di questi risultati, effettuare test con una maggior mole di dati permette di produrre risultati più generici, in modo da non cadere in singoli casi isolati; è stato quindi scelto un preset intermedio in modo da non danneggiare le prestazioni dei due codec, ma allo stesso tempo produrre risultati in un tempo ragionevole.

2.4 Valutazione qualità dei video

Per misurare la qualità effettiva di un video, è stato utilizzato il PSNR (Peak Signal-to-Noise Ratio).

Il PSNR di un video è stato calcolato nel modo seguente:

- Per ogni canale di un frame, è stato calcolato il MSE (Mean Squared Error) tra l'immagine originale e la versione convertita:

$$\text{MSE}_x = \frac{1}{M \cdot N} \sum_{i=1, j=1}^{N, M} (f_{1,x}[i][j] - f_{2,x}[i][j])^2$$

$f_{1,x}$ indica il frame x del video 1 (stessa logica per $f_{2,x}$)

M e N indicano la larghezza e l'altezza di un frame.

- Per immagini a colori, saranno presenti tre diversi MSE, indicati come

$$\text{MSE}_Y, \quad \text{MSE}_{Cb} \quad \text{MSE}_{Cr}$$

- Per ogni frame viene calcolato il MSE pesato secondo la seguente formula:

$$\text{MSE}_x = \text{MSE}_Y \cdot w_Y + \text{MSE}_{Cb} \cdot w_{Cb} + \text{MSE}_{Cr} \cdot w_{Cr}$$

w_Y , w_{Cb} e w_{Cr} non hanno un valore fisso: per questa tesi sono stati scelti dei valori di $[\cdot75, \cdot125, \cdot125]$, come mostrato nel [6].

- Si calcola il PSNR di ogni frame:

$$\text{PSNR}_x = 10 \cdot \log_{10}\left(\frac{255^2}{\text{MSE}_x}\right)$$

- Infine viene fatta la media aritmetica tra i PSNR di ogni frame

$$\text{PSNR} = \frac{1}{L} \sum_{x=1}^L \text{PSNR}_x$$

L è il numero di frame presenti nel video.

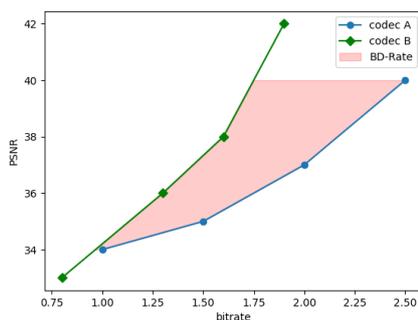
2.5 Metrica di Bjøntegaard

La metrica di Bjøntegaard offre un metodo di calcolare le differenze tra due curve tasso-distorsione. Il processo con cui si calcola tale metrica è composto dai seguenti 3 elementi principali:

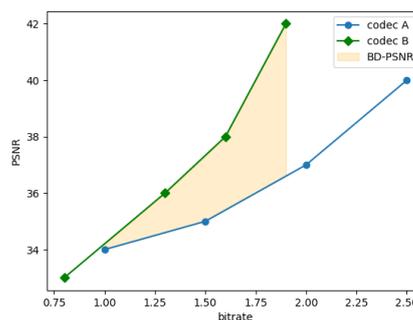
1. Effettuare un'interpolazione della curva tasso-distorsione
2. Trovare l'espressione che descrive l'integrale della curva
3. La differenza media tra le due curve è alla fine calcolata facendo la differenza tra gli integrali divisa per l'intervallo di integrazione.

Con questi passaggi è possibile calcolare due metriche:

1. **BD-PSNR**: La differenza media in dB dei PSNR sull'intero intervallo di bit-rate
2. **BD-Rate**: La differenza media in percentuale dei bit-rate sull'intero intervallo di PSNR



(a) BD-Rate



(b) BD-PSNR

Per maggiori informazioni riguardo la metrica di Bjøntegaard, si faccia riferimento all'articolo [7].

Capitolo 3

Analisi risultati

3.1 Test su video a bassa risoluzione

3.1.1 Prestazioni per bit-rate richiesti

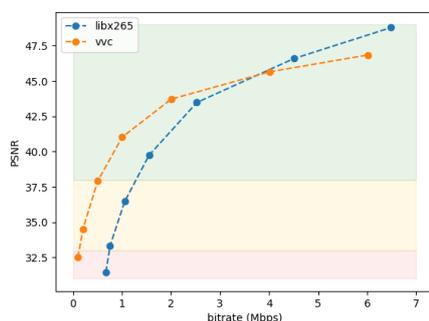
Un primo round di test che è stato effettuato comprendeva la specifica del bit-rate richiesto, per poi studiare la qualità, il bit-rate effettivo e il peso.

Per ogni video su cui sono stati effettuati test, è stato prodotto un grafico che mette in relazione il bit-rate effettivo e la qualità del video generato, misurata tramite il PSNR.

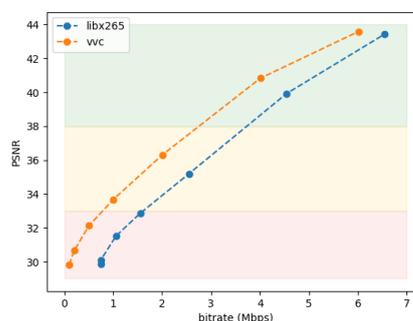
Effettuando i test su vari video, è possibile notare che tutti i video producono risultati coerenti tra loro, a meno di qualche piccola variazione.

Prestazioni a bit-rate richiesti bassi

Il codec vvc dimostra prestazioni superiori per bit-rate richiesti bassi (tra i 100kbps e i 2Mbps), raggiungendo sempre una qualità maggiore rispetto alla controparte:



(a) Video: akiyo



(b) Video: deadline

Si nota dalle immagini mostrate, che il video generato dal codec hevc presenta più artefatti rispetto a quello generato dal codec vvc, visibili soprattutto nella parte del volto a sinistra, sull'occhio guancia e labbra.

Un altro fattore da analizzare è il peso dei video: il video generato utilizzando il codec vvc occupa circa la metà dello spazio usato da quello generato dal codec hevc:



(a) akiyo, original



(b) akiyo, 1059kbps, hevc



(c) akiyo, 508kbps, vvc



(a) deadline, original



(b) deadline, 1,5Mbps, hevc



(c) deadline, 1Mbps, vvc

Peso del video originale (akiyo.y4m): 45,6MB

Peso del video utilizzando lo standard hevc con bit-rate effettivo 1059kbps: 1,3MB

Peso del video utilizzando lo standard vvc con bit-rate effettivo 508kbps: 636kB

Peso del video originale (deadline.y4m): 208,9MB

Peso del video utilizzando lo standard hevc con bit-rate effettivo 1,5Mbps: 8,9MB

Peso del video utilizzando lo standard vvc con bit-rate effettivo 1Mbps: 5,8MB

Il codec vvc riesce a raggiungere in entrambi i casi pesi inferiori, ottenendo comunque PSNR superiori.

Prestazioni per bit-rate richiesti alti

A bit-rate elevati, il codec vvc riesce a raggiungere PSNR superiori o bit-rate effettivi inferiori rispetto al codec hevc.

Si possono notare situazioni in cui il codec hevc riesce a raggiungere PSNR superiori: osservando i bit-rate a cui si raggiungono questi risultati, si può notare come siano bit-rate estremamente elevati, di conseguenza è possibile ignorare questi risultati in quanto si cerca sempre di utilizzare bit-rate bassi per la trasmissione di media in rete. Inoltre si può notare che il PSNR raggiunto da entrambi i codec, è superiore ai 40, ovvero qualità visive ottime:



(a) akiyo, original

(b) hevc, 2,5Mbps, PSNR=43

(c) hevc, 6,5Mbps, PSNR=49



(d) vvc, 2Mbps, PSNR=43

(e) vvc, 6Mbps, PSNR=46

Come si può notare, non ci sono differenze nella qualità visiva dei vari frame.

Analisi dei bit-rate effettivi

Si può notare dai grafici come i bit-rate richiesti non rispecchiano a pieno i bit-rate che i codec raggiungono effettivamente:

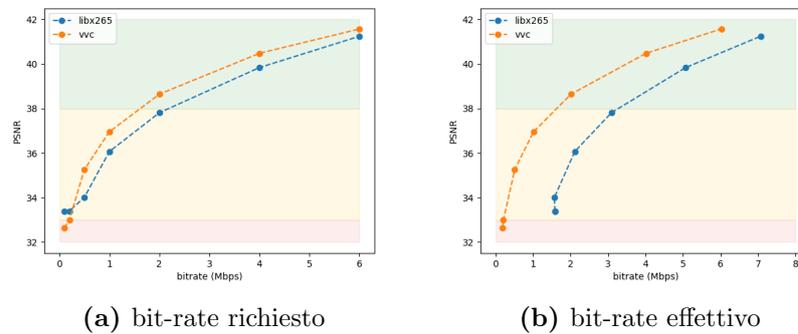


Figura 3.5: Video: ice

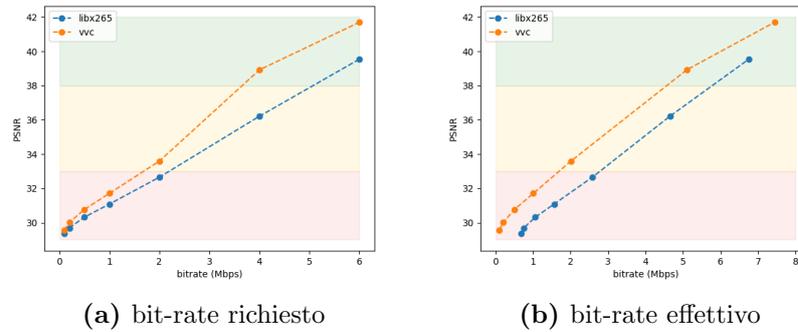


Figura 3.6: Video: waterfall

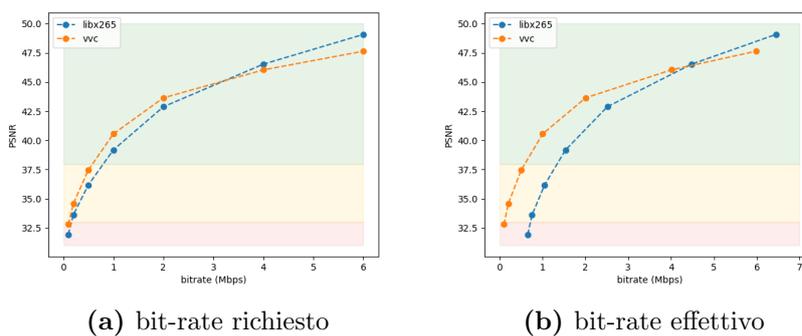


Figura 3.7: Video: mother_daughter

Come si può vedere, analizzando il grafico ottenuto utilizzando i bit-rate richiesti, le prestazioni dei due codec sembrano pari, ma andando ad analizzare il bit-rate effettivo raggiunto dai due codec, e utilizzando questi valori per l'asse delle ascisse, le due curve iniziano a distaccarsi.

Questa differenza tra bit-rate richiesto e bit-rate effettivo è dovuta alla tecnica di controllo di tasso che viene utilizzata dai codec: questa tecnica non è standardizzata e può dipendere

dalla risoluzione del video, di conseguenza potrebbe non essere ottimizzata per video a bassa risoluzione.

3.1.2 Metrica di Bjøntegaard

La metrica di Bjøntegaard è stata calcolata sia su tutta la curva bit-rate/PSNR, che sui 4 valori con un bit-rate inferiore.

Come discusso in precedenza, i risultati ottenuti per bit-rate elevati sono meno rilevanti, di conseguenza andando a calcolare la metrica di Bjøntegaard per bit-rate inferiori, si otterranno risultati il più possibile indipendenti da questa possibile inaccuratezza.

Di seguito sono mostrati valori di BD-RATE e BD-PSNR per alcuni dei video testati.

VIDEO NAME	BD-RATE	BD-RATE (primi 4 punti)	BD-PSNR
mother_daughter	-50.09%	-69.45%	2.50dB
waterfall	-41.63%	-38.34%	1.81dB
container	-39.62%	-70.51%	2.01dB
football	-38.89%	-70.49%	1.96dB
akiyo	-52.66%	-69.78%	2.24dB
claire	-59.67%	-68.10%	2.6dB
hall_monitor	-45.06%	-69.61%	2.03dB
grandma	-45.94%	-62.61%	2.37dB
coastguard	-35.17%	-67.55%	1.70dB
bowing	-48.44%	-68.02%	2.31dB

I risultati sono tutti a favore del codec vvc: riesce a raggiungere sempre bit-rate inferiori rispetto al codec hevc e al contempo raggiunge PSNR superiori.

Calcolando le medie, si ottengono i seguenti risultati:

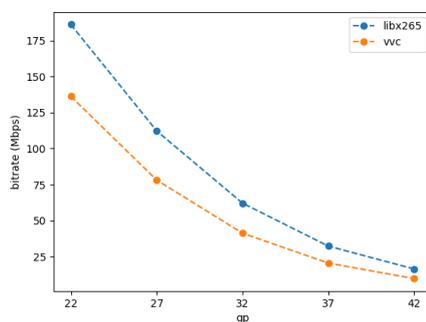
BD-RATE	BD-RATE (primi 4 punti)	BD-PSNR	BD-PSNR (primi 4 punti)
-45.71%	-68.84%	2.15dB	4.22dB

3.2 Test su video HD

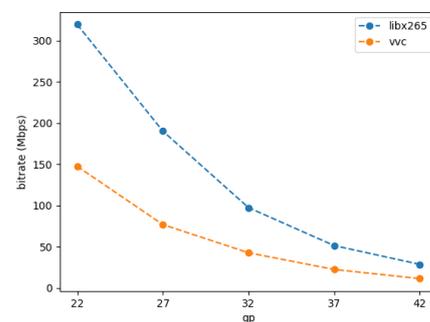
I risultati ottenuti dai test effettuati sui video ad alta qualità rimarcano i risultati già visti, andando a evidenziare la capacità dello standard vvc di raggiungere spesso bit-rate inferiori rispetto allo standard hevc.

3.2.1 Bit-rate raggiunti

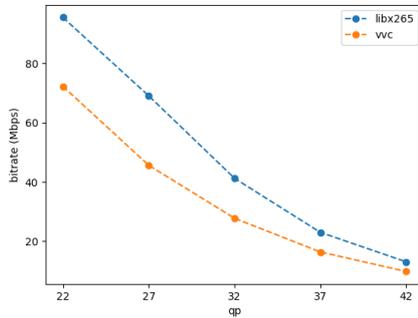
Andando a calcolare il bit-rate raggiunto in funzione del passo di quantizzazione richiesto, si può notare la tendenza del codec hevc a raggiungere bit-rate più elevati rispetto al codec vvc: Andando ad analizzare le curve bit-rate/PSNR si ottengono spesso qualità



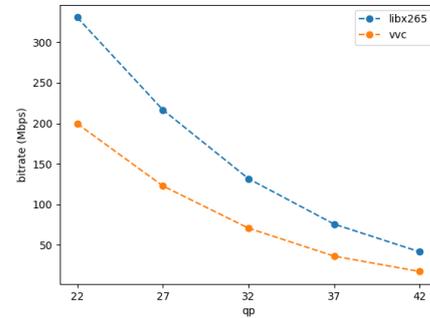
(a) Video: CSGO



(b) Video: ducks take off

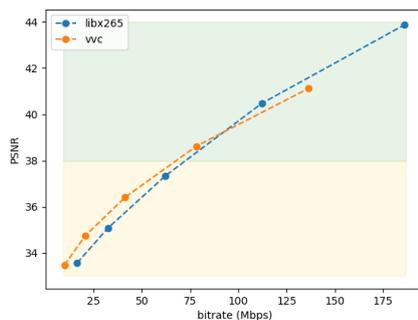


(c) Video: rush field cuts

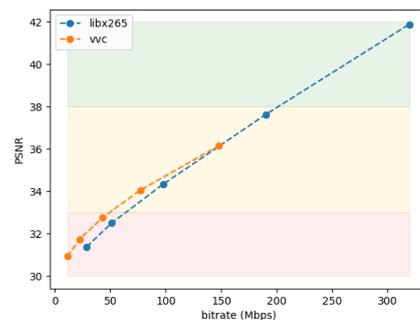


(d) Video: park joy

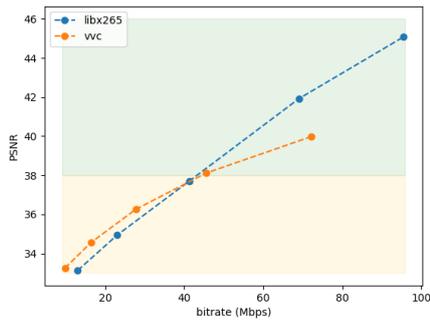
superiori con il codec vvc, a parità di bit-rate.



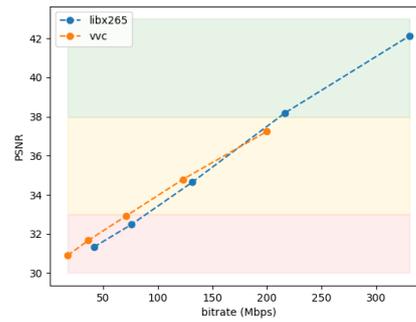
(a) Video: CSGO



(b) Video: ducks take off



(a) Video: rush field cuts



(b) Video: park joy

Come è stato osservato in alcuni video a bassa risoluzione, anche nei video HD vengono raggiunte qualità massime inferiori dal codec vvc: andando ad analizzare la qualità visiva delle immagini si osservano qualità simili.

Tuttavia è possibile osservare come i bit-rate effettivi raggiunti dal codec vvc siano spesso inferiori rispetto a quelli raggiunti dal codec hevc: questo fatto sarà analizzato più nel dettaglio in seguito, nel capitolo dedicato alla metrica di Bjøntegaard.

3.2.2 Qualità visiva

Andando a confrontare la qualità visiva ottenuta dai codec, è possibile notare come la qualità visiva è la stessa: questo perchè, come era già stato precisato, per PSNR elevati, i cambiamenti nella qualità sono quasi impercettibili.

Lo standard vvc riesce a raggiungere qualità superiori per bit-rate inferiori: andando a guardare la qualità visiva, si possono notare nel video generato dal codec hevc degli artefatti che creano un effetto di flickering che sono assenti nel video generato dal codec vvc.

Lo standard vvc, infatti, ha tecniche di smoothing che permettono di rendere più liscia l'immagine.



(a) rush_field_cuts, original

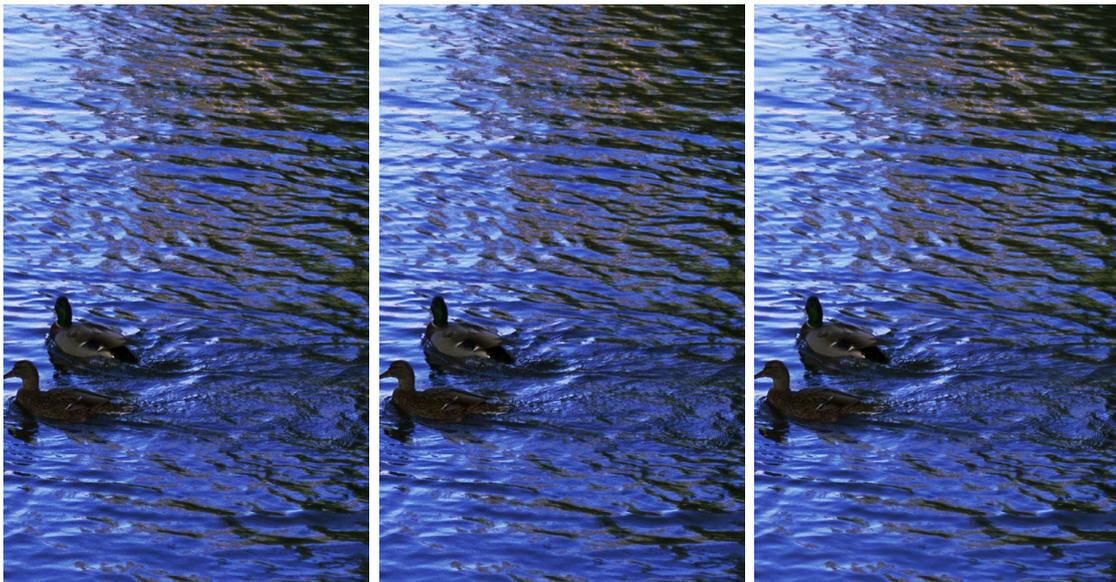
(b) hevc, 13Mbps, PSNR=33

(c) vvc, 9Mbps, PSNR=33

Si può notare in questo esempio, il prato nel frame del video generato con il codec vvc è più liscio rispetto a quello generato dal codec hevc.



(a) hevc, 95Mbps, PSNR=45 (b) vvc, 72Mbps, PSNR=40



(a) ducks_take_off, original (b) hevc, 300Mbps, PSNR=42 (c) vvc, 147Mbps, PSNR=36

Differenze di qualità visiva così piccole, non giustificano la differenza di bit-rate tra i due video. Il bit-rate che il codec vvc riesce a raggiungere è addirittura la metà, in questo esempio, rispetto al bit-rate raggiunto dal codec hevc.

Questo bit-rate inferiore porta un enorme vantaggio in quanto sarà possibile trasmettere con un bit-rate molto più basso video con buone qualità visive.

3.2.3 Metrica di Bjøntegaard

Di seguito sono mostrati valori di BD-RATE e BD-PSNR per alcuni dei video testati.

VIDEO NAME	BD-RATE	BD-RATE (primi 4 punti)	BD-PSNR
park_joy	-12.65%	-19.09%	0.46dB
ducks_take_off	-18.46%	-26.45%	0.49dB
FourPeople	-15.91%	-24.39%	0.86dB
CSGO	-12.55%	-19.42%	0.39dB
DOTA2	-11.52%	-17.08%	0.53dB
life	-8.56%	-15.43%	0.25dB
controlled_burn	-9.58%	-14.08%	0.48dB
RUST	-12.69%	-19.79%	0.36dB
rush_field_cuts	-7.08%	-14.60%	0.06dB

In media:

BD-RATE	BD-RATE (primi 4 punti)	BD-PSNR	BD-PSNR (primi 4 punti)
-12.12%	-18.93%	0.44dB	0.70dB

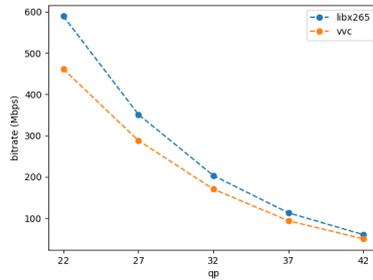
Nonostante sembri che per la classe di video HD il codec vvc abbia guadagni più bassi rispetto ai guadagni ottenuti nei video a bassa risoluzione, bisogna tener conto dei maggiori bit-rate raggiunti dai video HD.

È possibile notare, come per i video a bassa risoluzione, che il BD-RATE è più grande per qualità richieste inferiori (ovvero calcolato solo per i primi 4 punti dei grafici); come conseguenza di questo fatto, per applicazioni di uso comune, il codec vvc riesce a raggiungere prestazioni ulteriormente migliorate rispetto al codec hevc.

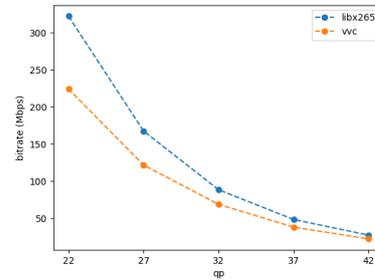
3.3 Test su video 4k

3.3.1 Bit-rate raggiunti

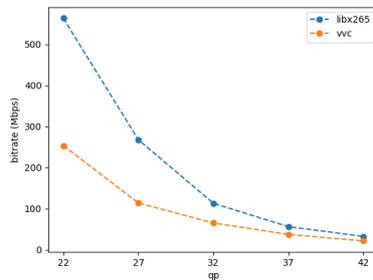
Il codec vvc riesce, come per i video HD, a raggiungere bit-rate inferiori rispetto al codec hev.



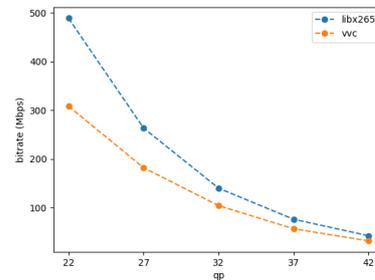
(a) Video: Netflix_Aerial



(b) Video: Netflix_PierSeaside



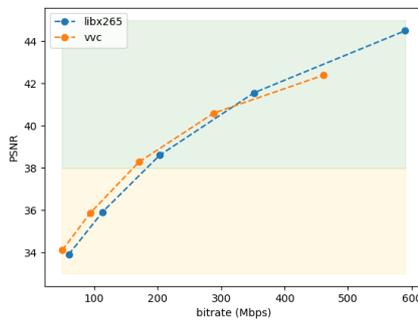
(c) Video: Netflix_FoodMarket



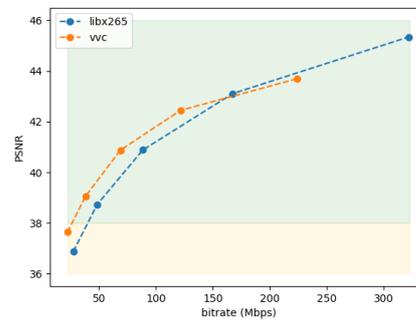
(d) Video: Netflix_DrivingPOV

3.3.2 Prestazioni per bit-rate effettivi

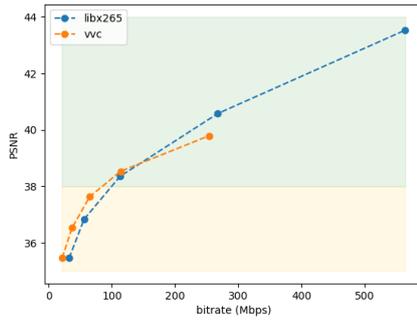
Nonostante i bit-rate inferiori raggiunti dal codec vvc, la qualità non ne risente in quanto vengono raggiunti PSNR superiori ai 40 con enormi guadagni di bit-rate:



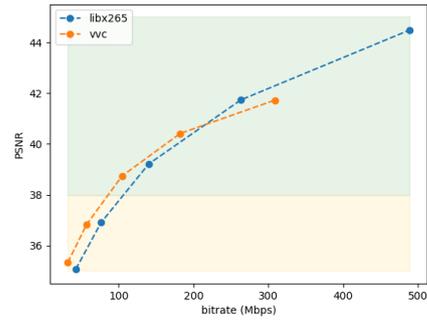
(a) Video: Netflix_Aerial



(b) Video: Netflix_PierSeaside



(a) Video: Netflix_FoodMarket



(b) Video: Netflix_DrivingPOV

3.3.3 Metrica di Bjøntegaard

Di seguito sono mostrati valori di BD-RATE e BD-PSNR per alcuni dei video testati.

VIDEO NAME	BD-RATE	BD-RATE (primi 4 punti)	BD-PSNR
Netflix_BarScene	-7.23%	-31.13%	0.11dB
Netflix_RollerCoaster	-18.49%	-22.79%	0.58dB
Netflix_FoodMarket	-14.75%	-22.52%	0.22dB
Netflix_Aerial	-9.36%	-12.74%	0.36dB
Netflix_PierSeaside	-20.69%	-24.92%	0.67dB
Netflix_ToddlerFountain	-5.15%	-10.19%	0.12dB
Netflix_Tango	-24.49%	-29.68%	0.35dB
Netflix_DrivingPOV	-14.88%	-19.36%	0.45dB

In media:

BD-RATE	BD-RATE (primi 4 punti)	BD-PSNR	BD-PSNR (primi 4 punti)
-9.44%	-29.42%	0.29dB	0.52dB

Per video 4K, considerando bit-rate utili per applicazioni di uso comune, si ottengono valori di BD-RATE medi di -30%, quasi raddoppiando il guadagno osservato per i video HD.

Capitolo 4

Considerazioni finali

Lo standard vvc, come visto in questo elaborato, introduce varie tecniche di codifica Intra, le quali sono mirate non solo ad aumentare la qualità visiva dei video codificati, ma soprattutto a raggiungere bit-rate inferiori in modo da poter permettere una trasmissione dei video più fluida e adatta al web.

Per video a bassa risoluzione, lo standard vvc riesce ad aumentare la qualità dei video mantenendo comunque un bit-rate contenuto.

Sono stati notati alcuni punti in cui lo standard hevc riesce a codificare video con PSNR superiori, ma ad un bit-rate più elevato rispetto allo standard vvc: bit-rate elevati non sono utili per contenuti a risoluzione SD o inferiore, di conseguenza le prestazioni raggiunte a bit-rate bassi sono più rilevanti.

Inoltre, PSNR tanto alti, come già dimostrato, non portano vantaggi visivi.

Effettuando analisi più in dettaglio, tramite la metrica di Bjøntegaard, si può quantificare questo guadagno di bit-rate ottenuto dallo standard vvc: per video HD si raggiunge un BD-Rate medio del -12% (-19% considerando bit-rate utili per applicazioni di uso comune) mentre per i video 4K si raggiunge un BD-Rate medio del -9% (-29% considerando bit-rate utili per applicazioni di uso comune).

Per le applicazioni di uso comune, quindi, le tecniche intra introdotte dallo standard vvc portano enormi vantaggi, che permettono di raggiungere qualità superiori per bit-rate inferiori, rendendo quindi la trasmissione dei contenuti video più fluida e resistente a disturbi o problemi di connessione.

Capitolo 5

Bibliografia

5.1 Articoli

1. **Intra Coding of the HEVC Standard**

Jani Lainema, Frank Bossen, Member, IEEE, Woo-Jin Han, Member, IEEE, Junghye Min, and Kemal Ugur

2. **Intra Prediction and Mode Coding in VVC**

Jonathan Pfaff, Alexey Filippov, Shan Liu, Senior Member, IEEE, Xin Zhao, Member, IEEE, Jianle Chen, Senior Member, IEEE, Santiago De-Luxán-Hernández, Thomas Wiegand, Fellow, IEEE, Vasily Ruffitskiy, Adarsh Krishnan Ramasubramonian, and Geert Van der Auwera, Senior Member, IEEE

3. Said, A.; Zhao, X.; Karczewicz, M.; Chen, J.; Zou, F. **Position Dependent Prediction Combination for Intra-Frame Video Coding**. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016

4. Boyce, Jill & Suehring, Karsten & Li, Xiang & Seregin, Vadim. (2018). **JVET-J1010: JVET common test conditions and software reference configurations**

5. **Cisco white paper (2018-2023)**

6. X. Shang et al. **Color-Sensitivity-Based Combined PSNR for Objective Video Quality Assessment**

7. G. Bjontegaard, **Calculation of average PSNR differences between RD-curves (VCEG-M33)** S. Pateux, J. Jung, An excel add-in for computing Bjontegaard metric and its evolution

5.2 Collegamenti ipertestuali

[L1] <https://deeprender.ai/blog/motion-compensation-and-prediction-h265>

[L2] <https://ffmpeg.org>

[L3] **Patch vvc**: <https://github.com/fraunhoferhhi/vvenc/wiki/FFmpeg-Integration>

[L4] **Dataset video**: <https://media.xiph.org/video/derf>

[L5] **Velocità internet suggerita (Netflix)**: <https://help.netflix.com/en/node/306>

[L6] **Metrica di Bjøntegaard su Python**: <https://pypi.org/project/bjontegaard/>