

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA



Corso di Laurea Magistrale in Informatica

Compressione Dati

Light Field Image Compression

Relatore:

Prof.

Bruno CARPENTIERI

Team:

Matteo Della Rocca

Mat. 0522501500

Luca Boffa

Mat. 0522501521

Vincenzo Di Leo

Mat. 0522501408

ANNO ACCADEMICO 2023/2024

INDICE

1	Introduzione	4
1.1	Gli ologrammi	4
1.2	Un pò di storia	4
1.3	Obiettivi dello studio	5
1.4	Organizzazione del paper	5
2	LIGHT FIELD IMAGE	7
3	Lavori correlati	9
4	Fase di ricerca	10
4.1	Tabella riassuntiva	11
4.1.1	Cirrus Logic AccuPak	11
4.1.2	FLV	11
4.1.3	Motion JPEG	11
4.1.4	MPEG4	12
4.1.5	ProRes	12
4.1.6	FFVHUFF	13
4.1.7	LCL-ZLIB	13
4.1.8	MagicYuv	13
4.2	Metriche selezionate per il confronto	13
4.2.1	SSIM	13
4.2.2	PSNR	14
4.2.3	Test indice SSIM con riferimento	15
4.2.4	Test indice PSNR con riferimento	15
5	Sistema proposto	17
6	Implementazione	18
6.1	Linguaggio, librerie principali e tool usati	18
6.2	Organizzazione del progetto e codice	19
6.2.1	Moduli python principali	19

6.2.2	Directory principali	19
6.3	Metodologie esperimento	19
6.4	Fase di compressione	19
6.5	Fase di decompressione	20
6.6	Studio codec Lossy a parità di SSIM	21
7	Esperimenti svolti	22
7.1	Applicazione del sistema proposto sui dataset utilizzati in precedenza	22
7.2	Studio sulla correlazione tra light field contigui	22
7.3	Aggiunta di nuovi codec e nuovi dataset	22
7.3.1	Dataset utilizzati	23
8	Analisi dei risultati	25
8.1	Configurazione Hardware	25
8.2	Nomenclatura esperimenti svolti	25
8.3	Risultati ottenuti	26
8.4	Codec 2022 sui dataset 2022	26
8.5	Codec 2022 sui dataset 2022 (random)	27
8.6	Codec 2022 sui dataset 2023	28
8.7	Codec 2022 sui dataset 2023 (random)	30
8.8	Codec 2023 sui dataset 2022	32
8.8.1	Codec lossless 2023 sui dataset 2022	32
8.8.2	Codec lossy	33
8.9	Codec 2023 sui dataset 2022 (random)	34
8.9.1	Codec lossless	34
8.9.2	Codec lossy 2023 sui dataset 2022 random	35
8.10	Codec 2023 sui dataset 2023	36
8.10.1	Codec lossless	36
8.10.2	Codec lossy	38
8.11	Codec 2023 sui dataset 2023 (random)	40
8.11.1	Codec lossless	40
8.11.2	Codec lossy	42
8.12	Decompressione codec 2023 su Dataset 2022	44
8.13	Decompressione codec 2023 su Dataset 2022 (random)	45
8.14	Decompressione codec 2023 su Dataset 2023	46
8.15	Decompressione codec 2023 su Dataset 2023 (random)	48
8.16	Confronto su diversi livelli di SSIM	50
8.16.1	Cobblestone	50
8.16.2	Cobblestone Random	51
8.16.3	Messerschmitt	52
8.16.4	Messerschmitt Random	53
8.16.5	Dragons	54
8.16.6	Dragons_random	55
8.16.7	Mannequin	56
8.16.8	Mannequin_random	57

INDICE

8.16.9 ArtGallery2	58
8.16.10 ArtGallery2_random	58
8.16.11 Blob	59
8.16.12 Blob_random	59
8.16.13 Car	60
8.16.14 Car_random	60
8.16.15 Dice	61
8.16.16 Dice_random	61
8.16.17 Fish	62
8.16.18 Fish_random	62
8.17 Analisi riassuntiva dei risultati	63
8.17.1 Analisi sui dataset random	64
9 Conclusione e Sviluppi futuri	65
Riferimenti	66
Elenco delle figure	68
Elenco delle tabelle	72

CAPITOLO 1

INTRODUZIONE

1.1 Gli ologrammi

Gli ologrammi sono definiti come figure o pattern d'onda interferenti, ottenute tramite l'uso di un laser, aventi la specificità di creare un effetto fotografico tridimensionale: essi, a differenza delle normali fotografie, ci mostrano una rappresentazione tridimensionale dell'oggetto proiettato. Poiché le statistiche e le proprietà dei segnali olografici differiscono notevolmente dalle immagini naturali come le fotografie, le soluzioni di codifica convenzionali non sono ottime. Ad oggi si hanno problemi nel salvataggio e trasmissione degli ologrammi dovuti alla loro dimensione. Pertanto, sono necessarie nuove soluzioni di codifica e trasformazioni per comprimere gli ologrammi in modo più efficace.

1.2 Un pò di storia

Gli ologrammi nascono intorno alla prima metà degli anni '40. Dennis Gabor, famoso scienziato ungherese, sviluppò la teoria dell'olografia mentre lavorava per migliorare la risoluzione di un microscopio elettronico. Fu egli stesso a coniare il termine "ologramma", derivante dall'unione delle due parole greche **holos** (intero) e **gramma** (messaggio). Purtroppo, negli anni a seguire gli ologrammi non suscitarono grande successo, date le sorgenti luminose ancora poco sviluppate ai tempi.

Finalmente, negli anni '60, fu inventato il laser che, grazie al suo potente lampo di luce, dalla durata di pochi nanosecondi, si rivelò perfetto per creare ologrammi. Il laser, infatti, riesce a bloccare il movimento efficacemente, creando in tal modo ologrammi di eventi o persone ad alta velocità. In particolare, il primo ologramma di una persona è stato creato nel 1967. Da qui nasce la ritrattistica olografica pulsata.

Nel 1962 due studiosi americani decisero di superare la teoria di Gabor e utilizzare, oltre al laser, anche una tecnica messa in atto già nel loro lavoro (studiavano un modo per realizzare un radar a lettura laterale). Il risultato di tutto ciò fu la creazione dei primi ologrammi 3D in movimento: un trenino e un uccello.

Un ulteriore sviluppo nel mondo degli ologrammi è stato raggiunto nel '68 da Stephen A. Benton che ha inventato l'olografia a trasmissione di luce bianca, la quale permette di creare un'immagine “arcobaleno” dai sette colori che compongono la luce bianca. L'invenzione di Benton è importante in quanto ha reso possibile la produzione in serie di ologrammi attraverso una tecnica di goffratura.

Oggi gli strumenti necessari a creare ologrammi 3D (un laser a onda continua, dispositivi ottici come lenti o specchi per dirigere la luce laser, un supporto per pellicola e un tavolo isolante su cui vengono effettuate le esposizioni) sono posseduti da moltissimi laboratori e studi.[1]

1.3 Obiettivi dello studio

Questo studio rappresenta un'estensione di una ricerca precedente nel campo della compressione delle immagini Light Field, mantenendo lo stesso approccio metodologico. L'obiettivo principale è di confermare o smentire le conclusioni raggiunte nel precedente studio e di arricchire la ricerca mediante l'introduzione di nuovi codec, dataset e criteri di valutazione.

Le immagini Light Field sono particolari tipologie di immagini utilizzate per rappresentare gli ologrammi attraverso dispositivi di visualizzazione appositi. La loro compressione richiede particolare attenzione per mantenere la qualità dell'immagine e ridurre lo spazio di archiviazione necessario.

Per questo studio, sono stati selezionati algoritmi di compressione lossy e lossless applicati ai dataset delle immagini Light Field. La valutazione delle prestazioni di tali algoritmi si basa su criteri oggettivi come il rapporto di compressione e il tempo impiegato durante la compressione. Durante la decompressione, l'attenzione è focalizzata sull'accuratezza della ricostruzione dell'immagine, valutata tramite l'indice SSIM (Structural Similarity Index) e un nuovo indice, il PSNR (Peak Signal-to-Noise Ratio), specifico per i codec lossy.

L'inclusione di nuovi codec e dataset amplia la portata dello studio, consentendo di esplorare una gamma più ampia di opzioni di compressione e di valutare il loro impatto sulle prestazioni complessive degli algoritmi. Inoltre, è stato condotto un esperimento sulle relazioni inter-frame nelle immagini light field per comprendere meglio il loro effetto sull'efficacia dei codec durante le fasi di compressione e decompressione.

1.4 Organizzazione del paper

La struttura del paper rispecchia gli argomenti affrontati per lo studio, la progettazione e l'implementazione della soluzione proposta:

- **LIGHT FIELD IMAGE:** raccolta dei dati relativi ai light field image e l'importanza della compressione di questi ultimi.
- **LAVORI CORRELATI:** studio dei paper precedentemente realizzati per la compressione dei light field.

1. INTRODUZIONE

- FASE DI RICERCA: approfondimento delle tecnologie e dei linguaggi necessari al progetto, nonché la ricerca degli algoritmi di compressione disponibili sul mercato allo stato dell'arte.
- SISTEMA PROPOSTO: nella quale viene spiegata la nostra idea di soluzione.
- IMPLEMENTAZIONE: nella quale si è messo in atto quanto definito in fase di progettazione.
- ESPERIMENTI SVOLTI: esperimenti condotti al fine di valutare l'efficacia del sistema proposto nella compressione video applicata alle light field images.
- ANALISI DEI RISULTATI: esposizione degli esempi di compressione con gli algoritmi implementati e confronto tra i risultati ottenuti.
- CONCLUSIONE E SVILUPPI FUTURI: analisi del lavoro svolto nel complesso e i possibili sviluppi futuri.

CAPITOLO 2

LIGHT FIELD IMAGE

Light Field (campo luminoso) si riferisce alla quantità di luce in funzione della posizione e della direzione, ovvero una funzione vettoriale 4D nello spazio libero. Il campo luminoso è stato introdotto nella computer graphics nel 1996 da Levoy e Hanrahan, i quali hanno proposto un sistema di rendering basato su immagini (IBR) e sul campo luminoso registrato di una scena. Tale sistema IBR è potente nel rendering di nuove view, non solo con il cambiamento del punto di vista, ma anche per il punto focale cambiato, noto come ri-messa a fuoco. Tuttavia, la registrazione del campo luminoso 4D non è banale poiché i sensori di immagini esistenti sono progettati per l'imaging 2D. Una fotocamera tradizionale cattura solo una rappresentazione piatta e bidimensionale dei raggi di luce che raggiungono l'obiettivo in una data posizione. Il sensore di immagine 2D registra la somma della luminosità e del colore di tutti i raggi luminosi che arrivano a ogni singolo pixel. Al contrario, una light field camera può registrare non solo i valori di luminosità e colore nel sensore di immagine 2D, ma anche la direzione/l'angolo di tutti i raggi luminosi che arrivano al sensore. Queste informazioni aggiuntive ci permettono di ricostruire da dove proveniva esattamente ogni raggio di luce prima di raggiungere la telecamera, rendendo possibile il calcolo di un modello tridimensionale della scena catturata.

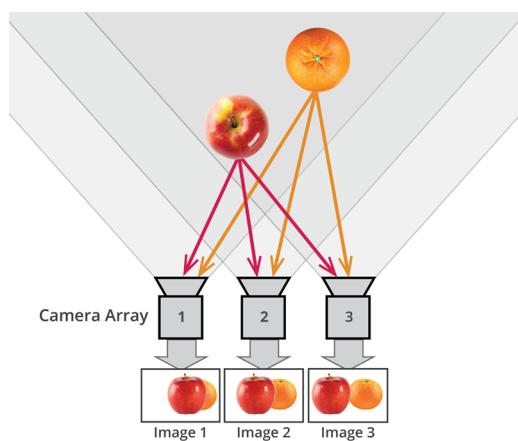


Figura 2.1: Cattura di un ligh field

I light field possono essere catturati utilizzando diverse tecniche, tra cui:

- una singola telecamera controllata roboticamente,
- un arco rotante di telecamere,
- una serie di telecamere o moduli telecamere,
- una singola fotocamera dotata di un array di microlenti.

A seconda della tecnica utilizzata, i dati dell'immagine acquisita sono tipicamente costituiti da un'immagine contenente molte immagini secondarie (ottenute da un singolo sensore e una matrice di microlenti) o molte immagini (da più fotocamere o esposizioni). Entrambi i set di dati mostrano immagini con leggere variazioni, poiché catturano raggi di luce da diverse angolazioni nello spazio. Queste variazioni permettono di determinare la posizione degli oggetti e di creare un volume di campo luminoso 3D. Tuttavia, le fotocamere plenottiche come Lytro e Raytrix, sebbene offrano nuove opportunità, richiedono una compressione efficace per gestire le dimensioni considerevoli delle immagini raw e la loro rappresentazione non convenzionale, mantenendo al contempo la qualità dell'immagine.

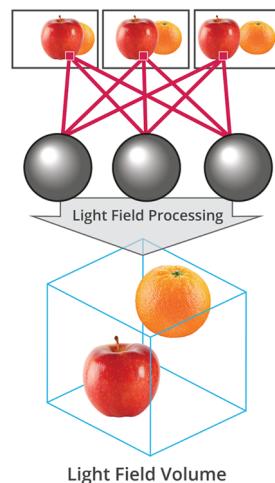


Figura 2.2: Informazioni racchiuse in un ligh field

CAPITOLO 3

LAVORI CORRELATI

Per la compressione dei light field ci siamo rifatti allo studio di un progetto passato svolto da nostri colleghi nell'università di Salerno[2].

I loro risultati si basano sulla compressione dei light field (che sono rappresentati tramite una serie immagini) in file di tipo video, così da sfruttare vari codec che si usano per comprimere le immagini in video.

Precisamente i nostri colleghi hanno utilizzato i seguenti codec:

- HVEC (lossy e lossless)
- VP9 (lossy e lossless)
- AV1 (lossy e lossless)
- HuffYUV (lossless)
- UT Video (lossless)
- FFV1 (lossless)

La nostra idea è quella di proseguire la strategia che i nostri colleghi hanno proposto andando ad espandere il loro studio già presente.

CAPITOLO 4

FASE DI RICERCA

Parte della ricerca si è basata sullo studio degli algoritmi utilizzati, e precisamente:

- perché e quando sono stati creati,
- da chi sono stati prodotti,
- a quali bisogni rispondono,
- da chi sono stati utilizzati nel tempo.

Inoltre, è stato condotto un esperimento per determinare se vi fosse una dipendenza effettiva tra i vari frame dei light field e se tale dipendenza potesse influenzare la qualità complessiva della compressione/decompressione.

Infine, è stata preparata una tabella riassuntiva per evidenziare le peculiarità fondamentali per la comprensione dei risultati del progetto. Ulteriormente, sono stati esaminati il Structural Similarity Index (SSIM) e il Peak Signal-to-Noise Ratio (PSNR), due metodi utilizzati per valutare la qualità percepita delle immagini in scenari di compressione lossy.

4.1 Tabella riassuntiva

Codec	Licenza	Perdita
Cirrus Logic FLV	Commerciale	Lossy
MJPEG	GNU	Lossy
MPEG4	GNU	Lossy
ProRes	Commerciale	Lossy
FFVHUFF	GNU	Lossless
LCL-ZLIB	GNU	Lossless
MAGICYUV	Commerciale	Lossless

4.1.1 Cirrus Logic AccuPak

Cirrus Logic AccuPak è un codec YUV a precisione ridotta.

Il codec AccuPak racchiude 4 campioni Y e 2 campioni C in 32 bit rappresentando ogni campione Y con 5 bit e ogni campione C con 6 bit. È essenzialmente un metodo ridimensionato di codifica YUV 4:1:1, in cui ogni gruppo di 4 pixel su una linea è rappresentato da un campione di luminanza ciascuno ma condivide campioni C[3].

4.1.2 FLV

Flash Video è un formato di file utilizzato per fornire contenuti video digitali (ad esempio, programmi TV, film, ecc.) su Internet utilizzando Adobe Flash Player versione 6 e versioni più recenti. Il contenuto Flash Video può anche essere incorporato nei file SWF. Ci sono due diversi formati di file Flash Video: FLV e F4V. I dati audio e video all'interno dei file FLV sono codificati allo stesso modo dei file SWF. FLV è stato originariamente sviluppato da Macromedia. Nei primi anni 2000, Flash Video era lo standard de facto per lo streaming video basato sul web (su RTMP).

I file Flash Video FLV di solito contengono materiale codificato con codec che seguono i formati di compressione video Sorenson Spark o VP6. A partire dal 2010 le versioni pubbliche di Flash Player (collaborazione tra Adobe Systems e MainConcept) supportano anche il video H.264 e l'audio HE-AAC. Tutti questi formati di compressione sono limitati dai brevetti. Flash Video è visualizzabile sulla maggior parte dei sistemi operativi tramite Adobe Flash Player e il plugin del browser web o uno dei numerosi programmi di terze parti. I dispositivi iOS di Apple, insieme a quasi tutti gli altri dispositivi mobili, non supportano il plugin Flash Player e quindi richiedono altri metodi di consegna come quelli forniti da Adobe Flash Media Server[4].

4.1.3 Motion JPEG

Motion JPEG (M-JPEG) è un codec video nel quale ogni singolo frame del video viene compresso in un'immagine JPEG.

Non offre nessuna compressione interframe, questo fa sì che la qualità della compressione sia

indipendente dal movimento presente nell'immagine, a differenza della compressione MPEG dove ci possono essere problemi di qualità quando il video contiene movimenti veloci o cambi scena. Questo codec facilita il montaggio video, in quanto permette tagli su ogni singolo frame, e non solo all'inizio di un gruppo di frame.

Si tratta di un formato che ha avuto una certa diffusione su fotocamere digitali e camere IP in quanto permetteva di usare la tecnologia della compressione JPEG anche per i filmati. Pur richiedendo un bitrate superiore al formato MPEG-1 permette risoluzioni superiori. È stato gradualmente soppiantato dai formati MOV(Apple), MPEG-2 e MPEG-4[5].

4.1.4 MPEG4

In elettronica e telecomunicazioni MPEG-4, nato nel 1996 e finalizzato nel 1998 (fu presentato pubblicamente a settembre di quell'anno), è il nome dato a un insieme di standard per la codifica dell'audio e del video digitale sviluppati dall'ISO/IEC Moving Picture Experts Group (MPEG). L'MPEG-4 è uno standard utilizzato principalmente per applicazioni come la videotelefonia e la televisione digitale, per la trasmissione di filmati via Web, e per la memorizzazione su supporti CD-ROM.

MPEG-4 è suddiviso in vari sotto standard chiamati part (termine inglese che in italiano significa "parte") e noi abbiamo usato la parte 2: un codec di compressione per i dati visivi (video, still textures...). Uno dei tanti "profili" della parte 2 è il Advanced Simple Profile (ASP).

Il concetto alla base del codec (CODificatore-DECodificatore) MPEG-4 è la quantizzazione. Senza scendere nello specifico, si può riassumere come quel processo che permette di trasmettere solamente la variazione dell'immagine mediante un apposito algoritmo di compressione[6].

4.1.5 ProRes

Apple ProRes è un formato di compressione video lossy di alta qualità, "visually lossless" sviluppato da Apple Inc. per l'uso in post-produzione che supporta una risoluzione video fino a 8K. È il successore dell'Apple Intermediate Codec ed è stato introdotto nel 2007 con Final Cut Studio 2. Proprio come gli standard H.26x e MPEG, la famiglia di codec ProRes utilizza algoritmi di compressione basati sulla trasformata del coseno discreto (DCT). ProRes è ampiamente utilizzato come metodo di consegna del formato finale per i file di trasmissione HD in pubblicità, funzionalità, Blu-ray e streaming.

ProRes è una linea di codec intermedi, il che significa che sono destinati all'uso durante l'editing video e non alla visualizzazione pratica da parte dell'utente finale. Ciò si ottiene utilizzando solo la compressione intra-frame, dove ogni fotogramma viene memorizzato in modo indipendente e può essere decodificato senza dipendenze da altri fotogrammi. Il vantaggio di un codec intermedio è che offre eccellenti prestazioni di accesso casuale nelle applicazioni di post-produzione e mantiene una qualità superiore rispetto ai codec dell'utente finale, pur richiedendo sistemi disco molto meno costosi rispetto ai video non compressi. È paragonabile al codec DNxHD o CineForm di Avid che offrono bitrates simili e sono anche destinati ad essere utilizzati come codec intermedi. ProRes è un codec solo intra-frame basato su scalare DCT ed è quindi più semplice da decodificare rispetto ai formati orientati alla distribuzione come H.264. Nel 2018 Apple ha aggiunto un nuovo "ProRes RAW" (filtro Bayer compresso) a Final Cut Pro

X, dopo che Blackmagic Design ha implementato Bayer compresso come "CinemaDNG 3:1" e "CinemaDNG 4:1" nelle loro fotocamere e DaVinci Resolve[7].

4.1.6 FFVHUFF

FFVHUFF è un codec video senza perdita di dati ed è una versione migliorata, e più veloce, del codec huffyuv. Può gestire più formati pixel[8].

4.1.7 LCL-ZLIB

Il codec converte i dati dell'immagine RGB24 originali in uno spazio colore di destinazione e lo comprime con un algoritmo selezionato. Il codec può anche rimuovere i fotogrammi invariati e sostituirli con fotogrammi nulli e può filtrare i dati dell'immagine prima della compressione. L'unica differenza tra avimsh e avizlib è nel compressore di flusso. Il filtraggio PNG è disponibile solo in avizlib. Fatta eccezione per i frame nulli, non c'è compressione temporale, e tutti i frame possono essere decodificati indipendentemente dagli altri. Ogni blocco AVI contiene un fotogramma. In caso di modalità multithread, le due sezioni sono memorizzate nello stesso blocco.

Come suggerisce il nome del codec, tutti i compressori sono lossless.

Questa modalità utilizza il metodo standard di sgonfiamento zlib. Per la descrizione dell'algoritmo fare riferimento ai documenti zlib. Lo stato del compressore viene reimpostato ogni fotogramma (decodifica ogni fotogramma in modo indipendente). I codici di compressione (1, 9, -1) hanno lo stesso significato dei flag di compressione zlib. Zlib non richiede un livello di compressione al decompressore, quindi il valore è lì solo a scopo informativo[9].

4.1.8 MagicYuv

Un codec video ad alte prestazioni, ultraveloce e matematicamente senza perdite per la registrazione, l'archiviazione, la post-produzione e l'editing ad alte risoluzioni.

MagicYUV è stato progettato per la velocità e per supportare pienamente la codifica e la decodifica multi-thread.

Questo codec è utilizzato per:

- Gaming
- Registrazioni e catture
- Editing e post-produzione
- Ricerca

Il Codec MagicYUV consente ai ricercatori di catturare ed elaborare video alle più alte risoluzioni e framerate possibili, in profondità di bit regolari e a colori profondi, mantenendo intatta ogni bit delle informazioni.

Questo rende MagicYUV uno dei codec video lossless più veloci del suo genere[10].

4.2 Metriche selezionate per il confronto

4.2.1 SSIM

L'indice SSIM (Structural Similarity Index Measure) è un modello basato sulla percezione che valuta la degradazione dell'immagine considerando i cambiamenti nella percezione delle

informazioni strutturali. Questo metodo tiene conto di importanti fenomeni basati sulla percezione, come il mascheramento della luminanza e il mascheramento del contrasto. Il termine "informazioni strutturali" si riferisce ai pixel fortemente interdipendenti o spazialmente vicini, enfatizzando la correlazione tra di essi. SSIM stima la qualità percepita di immagini e video. Misura la somiglianza tra due immagini: l'originale e la recuperata[11].

In particolare, date due immagini o segnali N -dimensionali (o porzioni corrispondenti), $x = (x_1, \dots, x_N)$ e $y = (y_1, \dots, y_N)$, l'indice SSIM esamina le similarità tra luminanza, contrasto e struttura.

1. Per la luminanza, $l(x, y)$, si utilizzano i valori medi, ad esempio,

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i.$$

2. Per il contrasto, $c(x, y)$, si utilizzano le varianze, ad esempio,

$$s_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2.$$

3. Per la struttura, $s(x, y)$, si utilizzano segnali normalizzati (deviazione standard unitaria), ad esempio,

$$x' = \frac{x - \bar{x}}{s_x}.$$

Successivamente, si combinano questi componenti (in qualche modo!) per ottenere una misura complessiva di similarità, cioè,

$$S(x, y) = F(l(x, y), c(x, y), s(x, y)).$$

4.2.2 PSNR

Il PSNR (Peak Signal-to-Noise Ratio) è una metrica utilizzata per calcolare il rapporto tra la potenza massima del segnale possibile e la potenza del rumore di distorsione che influisce sulla qualità della sua rappresentazione. Questo rapporto tra due immagini viene misurato in decibel per tener conto della vasta gamma dinamica dei segnali, che variano dai valori più grandi a quelli più piccoli. Il PSNR viene comunemente calcolato come il logaritmo in scala decibel, poiché i segnali hanno una gamma dinamica estesa. Questa gamma dinamica riflette la variazione tra i valori più alti e più bassi possibili, che sono critici per valutare la qualità dell'immagine in termini di fedeltà alla sua rappresentazione originale. Utilizzato ampiamente come tecnica di valutazione della qualità, il PSNR misura la qualità della ricostruzione nei codec di compressione delle immagini lossy. In questo contesto, il segnale rappresenta i dati originali, mentre il rumore è l'errore introdotto dalla compressione o dalla distorsione.

Nel degrado della qualità della compressione dell'immagine e del video, il valore PSNR varia da 30 a 50 dB per la rappresentazione dei dati a 8 bit e da 60 a 80 dB per i dati a 16 bit.

Il PSNR è espresso come:

$$PSNR = 10 \log_{10}(peakval^2) / MSE$$

Qui, *peakval* (Peak Value) è il massimo nei dati dell'immagine. Se si tratta di un tipo

Video Quality (MOS)	PSNR Range
Excellent	> 37
Good	31-37
Fair	25-31
Poor	20-25
Bad	< 20

Figura 4.1: Rapporto tra PSNR e qualità del video

di dati intero senza segno a 8 bit, il peakval è 255. Dall’equazione, possiamo vedere che è una rappresentazione dell’errore assoluto in dB[11].

Nel paper ”Social-Aware Video Multicast Based on Device-to-Device Communications” [12], viene condotto uno studio che esamina il rapporto tra la qualità del video e il valore dell’indice PSNR. La Tabella 4.1 fornisce una guida per interpretare i valori di PSNR e associarli a diverse categorie di qualità video. In particolare, si osserva che una qualità del video superiore è indicata da valori di PSNR pari o superiori a 37, mentre una qualità inferiore è associata a valori di PSNR inferiori a 25.

Nella fase di analisi dei risultati, la Tabella 4.1 sarà utilizzata come riferimento per valutare la qualità dei video decompressi. In altre parole, i valori ottenuti durante l’esperimento saranno confrontati con i range indicati nella tabella per determinare se la qualità del video è considerata ottima, buona, accettabile o pessima, in base ai criteri definiti dagli intervalli di PSNR.

4.2.3 Test indice SSIM con riferimento

Per valutare l’accuratezza del calcolo dell’indice di similarità strutturale (SSIM) nel nostro lavoro, è stato condotto un confronto con i valori riportati nel paper di riferimento ”*The SSIM Index for Image Quality Assessment*” [13]. A tale scopo, le stesse immagini utilizzate nel paper di riferimento sono state scaricate e utilizzate per il calcolo dell’indice SSIM mediante uno script personalizzato basato sulla libreria *skimage.metrics*. I valori ottenuti da questo script sono stati quindi confrontati con quelli riportati nel paper di riferimento.

Risultati

Il confronto tra i valori SSIM ottenuti dal nostro script e quelli riportati nel paper di riferimento ha mostrato una buona corrispondenza tra i due. Tuttavia, è stato osservato un errore medio relativo, calcolato come la media delle differenze percentuali tra i valori calcolati e quelli di riferimento, pari a **0.030**. Questo risultato suggerisce che il nostro script per il calcolo dell’indice SSIM ha una precisione accettabile e può essere considerato affidabile per valutazioni quantitative della qualità delle immagini.

4.2.4 Test indice PSNR con riferimento

Per valutare l’accuratezza del calcolo del PSNR, sono state selezionate due immagini - l’originale e quella con rumore - estratte dalla pagina web specificata [14], la quale fa riferimento a due paper accademici. Successivamente, utilizzando le informazioni fornite in questi paper, il PSNR è stato calcolato per le due immagini mediante l’utilizzo di uno script apposito che testa la funzione per il calcolo del PSNR fornita dalla libreria *skimage.metrics* e usata nell’esperimento.

Risultati

I valori del PSNR ottenuti sono stati confrontati con quelli riportati sulla pagina web di riferimento, che fornisce i valori di PSNR associati alle immagini specificate. Questo confronto ha permesso di determinare l'errore relativo del PSNR, risultato essere dello 0.026%. Questo risultato indica che vi è una buona corrispondenza tra i valori calcolati utilizzando la funzione di calcolo del PSNR fornita dalla libreria skimage.metrics e quelli riportati sulla pagina web di riferimento, confermando l'accuratezza del calcolo del PSNR.

CAPITOLO 5

SISTEMA PROPOSTO

Il nostro sistema ha come scopo principale quello di fornire un confronto tra vari algoritmi di compressione video e analizzare le loro prestazioni nella compressione di light field. Sostanzialmente, il nostro sistema è progettato per migliorare ed espandere uno studio precedente.

Come detto nel capitolo 3, le light field image hanno una forte componente di similarità tra frame della stessa scena. Si è deciso di sfruttare l'alta inter-dipendenza tra le varie immagini proprio come si fa per i video.

A tal fine abbiamo ri-implementato due script Python diversi, uno per la compressione dei light field in video e l'altro per la decompressione del video nelle sue componenti. Per la compressione usiamo diversi codec video, tutti di tipo lossless e lossy.

Per la decompressione invece usiamo il video creato nelle fasi precedenti e ne estraiamo i singoli frame. I frame estratti vengono poi salvati nello stesso formato delle immagini originali per monitorare così eventuali perdite di dati. Per ogni algoritmo di compressione effettuiamo un'analisi dello spazio risparmiato tramite il confronto della dimensione totale del dataset non compresso e del video compresso. Confrontiamo ogni frame decompresso con il suo corrispettivo originale, usando la structural similarity index measure (SSIM), per verificare che tra le due versioni non ci sia stata una perdita di informazioni nel caso di algoritmi lossless e che la perdita sia "accettabile" per gli algoritmi lossy.

I vari processi verranno spiegati nel dettaglio al capitolo 6.

CAPITOLO 6

IMPLEMENTAZIONE

6.1 Linguaggio, librerie principali e tool usati

Per la realizzazione di questo progetto, è stato impiegato il linguaggio di programmazione Python (versione 3.11.6), facendo uso delle seguenti librerie:

1. **Subprocess (versione 3.8):** La libreria subprocess è stata utilizzata per generare nuovi processi, connettersi alle loro pipe di input/output/errore e ottenere i relativi codici di ritorno;
2. **FFmpeg (versione 6.1):** FFmpeg è una completa suite software nata nel dicembre 2000, specializzata in registrazione, conversione e riproduzione di audio e video. Basata sulla libreria libavcodec per la codifica audio/video, FFmpeg è sviluppata principalmente su Linux, ma può essere compilata ed eseguita su vari sistemi operativi, compreso Microsoft Windows. In particolare, l'uso di FFmpeg in questo contesto è stato orientato alla conversione video da un formato all'altro attraverso uno strumento da riga di comando.
3. **OpenCV (opencv-python==4.8.1.78):** OpenCV è una libreria open-source che fornisce un'ampia varietà di strumenti per la visione artificiale e il machine learning. Nell'implementazione, OpenCV (cv2) è utilizzata per le operazioni di elaborazione delle immagini e manipolazione dei frame video.
4. **CSV (csv):** La libreria csv inclusa in Python che offre funzionalità per la lettura e la scrittura di file CSV (Comma-Separated Values). Nell'implementazione è stata impiegata per la gestione per registrare risultati e statistiche del processo.
5. **Scikit-image (scikit-image==0.22.0):** Scikit-image è una raccolta di algoritmi per il processamento delle immagini basata su Scikit-Learn. Le funzioni **structural_similarity (ssim)** e **peak_signal_noise_ratio (psnr)** fornite da skimage.metrics sono utilizzate per calcolare le metriche di similarità strutturale e rapporto segnale-rumore, rispettivamente.

6.2 Organizzazione del progetto e codice

6.2.1 Moduli python principali

- *test_compression.py*: Contiene funzioni e logica per la compressione video sui dataset presenti nel file *utils.py*.
- *test_decompression.py*: Gestisce la decompressione dei video compressi.
- *utils.py*: Contiene funzioni di utilità condivise, nel nostro caso sono presenti i dataset testati, le estensioni corrette dei codec video e le cartelle di output per la compressione e la decompressione.
- *random_dataset.py*: lo scopo di questo script è randomizzare l'ordine dei file corrispondenti al pattern nella cartella di origine e copiarli in modo ordinato nella cartella di destinazione.

6.2.2 Directory principali

- */datasets*: Contiene i dataset di input, in particolare ogni sottocartella fa riferimento ad uno specifico dataset.
- */compression_test*: Salva i risultati della compressione in una cartella contenente una sottocartella per ogni dataset. Ognuna di queste sottocartelle contiene a seconda dei codec usati il video compresso corrispondente. La seguente cartella in fase di compressione è creata in automatico.
- */decompression_test*: Salva i risultati della decompressione in una cartella contenente una sottocartella per ogni dataset: Ognuna di queste contiene a seconda dei codec usati un'altra sottocartella all'interno della quale sono salvate le immagini decomprese. La cartella in fase di decompressione è creata in automatico.
- */ffmpeg*: Nel caso di un sistema Windows, questa cartella è di fondamentale importanza in quanto contiene l'eseguibile di *ffmpeg*.

6.3 Metodologie esperimento

L'esperimento è stato automatizzato per semplificare il processo, trasformandolo in un processo di *benchmarking automatico* sia durante la fase di compressione che durante quella di decompressione. Di conseguenza, il programma eseguirà automaticamente i test con vari codec e sui vari dataset presenti nel file python *utils.py*, senza richiedere un'interazione continua da parte dell'utente. Questo approccio è particolarmente utile per condurre test di benchmark o valutare le prestazioni di diversi algoritmi in modo efficiente. Per lanciare il test di compressione è possibile usare il comando *python test_compression.py* all'interno della cartella, mentre per testare la decompressione *python test_decompression.py*.

6.4 Fase di compressione

In fase di compressione sono state utilizzate diverse funzioni le quali firme seguono il pattern `comp_{codec_scelto}` e implementano il processo di compressione video mediante

l'utilizzo di un codec specificato tramite FFmpeg. Il procedimento può essere riassunto in termini generici:

1. Parametri di Input:

- input_path: Percorso del file video originale;
- output_path: Percorso in cui verrà salvato il file video compresso.

2. Calcolo delle dimensioni: Viene calcolata la dimensione totale del file originale attraverso la somma delle dimensioni di tutti i file presenti nella stessa cartella del file originale.

3. Utilizzo di FFmpeg per la compressione: Utilizzando la libreria subprocess, viene eseguito FFmpeg dalla riga di comando. Il comando FFmpeg include specifiche come il tasso di frame di input, il codec video da utilizzare, insieme ai percorsi del file di input e di output;

4. Calcolo del tempo di compressione: Il tempo di inizio viene registrato prima della chiamata FFmpeg, e il tempo di fine viene registrato dopo il completamento della compressione. La durata totale del processo di compressione viene calcolata sottraendo il tempo di inizio da quello di fine.

5. Calcolo del rapporto di compressione: Utilizzando una funzione ausiliaria, vengono ottenute la dimensione del file compresso e il rapporto di compressione $\frac{\text{dimensione originale}}{\text{dimensione compressa}}$;

6. Output: Ogni funzione di questo tipo restituisce la dimensione iniziale, la dimensione finale, il rapporto di compressione e il tempo impiegato per la compressione. Inoltre nella cartella di output sarà possibile visualizzare il video ottenuto dalla compressione.

Questo approccio è flessibile e può essere adattato per eseguire test con diversi codec, contribuendo così alla valutazione delle prestazioni dei codec nel contesto dell'esperimento.

6.5 Fase di decompressione

In fase di decompressione di un particolare video, il procedimento generale è questo:

1. Ciclo di decompressione: Questa è la fase più importante e principale della decompressione. Il codice itera attraverso ogni frame del video compresso usando la libreria av. Durante l'iterazione, i frame vengono decodificati e salvati come immagini in una nuova directory, corrispondente al nome del dataset;

2. Creazione della struttura di output: Viene creata una nuova directory per l'output della decompressione per ogni dataset. All'interno di ciascuna directory, vengono create sotto-directory per ogni algoritmo di compressione utilizzato.

3. Applicazione di metriche di qualità: Per ogni coppia di immagini decompresse (originale e compressa), vengono calcolate metriche di qualità quali SSIM e PSNR. Le metriche vengono calcolate tra ogni frame dell'immagine originale e del risultato della decompressione;

4. **Raccolta e salvataggio dei risultati:** I risultati delle metriche vengono raccolti e salvati in una struttura dati, che include informazioni come il nome del dataset, l'algoritmo di compressione utilizzato, e le medie delle metriche SSIM e PSNR (poiché vi è un confronto 1:1 tra immagine originale e immagine decompressa). I risultati vengono inoltre salvati in un file CSV per un'ulteriore analisi o documentazione.

6.6 Studio codec Lossy a parità di SSIM

Il file `dataset_options.py` contiene un dizionario chiamato `DATASET_OPTIONS`, che definisce le impostazioni di compressione per ciascun dataset considerato durante gli esperimenti condotti. Ogni chiave del dizionario rappresenta un dataset specifico, mentre i valori associati a ciascuna chiave sono sotto-dizionari che specificano le opzioni di compressione per i diversi codec lossy utilizzati.

Per ogni dataset, sono elencati i codec FLV1, MJPEG, ProRes e MPEG4, insieme ai rispettivi valori del parametro `-q:v[15]` di `ffmpeg` utilizzati per regolare la qualità della compressione. Questi valori di qualità sono stati scelti in base ai risultati degli esperimenti condotti, tenendo conto dell'effetto sul rapporto di compressione e sulla qualità dell'immagine, valutata principalmente attraverso l'indice SSIM.

È importante notare che l'aumento del valore del parametro `-q:v` tende a diminuire l'indice SSIM, il che indica una riduzione della qualità dell'immagine a fronte di una maggiore compressione. Questo aspetto è stato preso in considerazione durante la selezione delle opzioni di compressione al fine di mantenere un indice SSIM stabile per ciascun dataset, consentendo così un confronto accurato delle prestazioni dei codec lossy.

CAPITOLO 7

ESPERIMENTI SVOLTI

Una serie di esperimenti è stata condotta al fine di valutare l'efficacia del sistema proposto nella compressione video applicata alle light field images.

7.1 Applicazione del sistema proposto sui dataset utilizzati in precedenza

Inizialmente sono stati eseguiti una serie di test utilizzando il nostro sistema per valutare le prestazioni dei dataset precedenti. Ciò ha consentito di ottenere una comprensione più approfondita delle dinamiche operative e dei risultati ottenuti dai nostri predecessori, permettendoci, inoltre, di identificare fino a quale punto il sistema precedente ha risposto alle esigenze previste. L'obiettivo principale è stato quello di valutare con maggiore precisione l'efficacia delle metodologie utilizzate precedentemente e di identificare eventuali punti deboli o aree di miglioramento.

7.2 Studio sulla correlazione tra light field contigui

Al fine di valutare l'impatto dell'ordine dei frames nei dataset originali, è stato condotto uno studio supplementare. Ogni dataset originale è stato sottoposto a un processo di trasformazione, generando così un nuovo dataset in cui i frames sono disposti in maniera casuale. L'obiettivo di questo studio è comprendere come la disposizione casuale dei frames influenzi le prestazioni e i risultati dei codec di compressione. Tale approccio è fondamentale per valutare la resilienza dei codec di compressione rispetto a variazioni nell'ordine temporale dei frames.

7.3 Aggiunta di nuovi codec e nuovi dataset

La scelta di aggiungere nuovi codec e di sottoporre tutti i dataset, vecchi e nuovi, a un riesame completo è stata fondamentale per comprendere in maniera esaustiva come le nuove implementazioni influiscano sulle prestazioni complessive del sistema. Tale approccio ha consentito di identificare correlazioni significative tra le caratteristiche dei codec, la selezione del

dataset e le performance globali, contribuendo a delineare un quadro completo delle dinamiche del sistema in esame.

7.3.1 Dataset utilizzati

Per il miglioramento dello studio già effettuato dai nostri colleghi, abbiamo deciso di utilizzare i 3 dataset implementati nel loro progetto, in modo da poterli testare su nuovi codec, in aggiunta a 7 ulteriori dataset. Lo scopo è di avere un confronto tra più dati, in modo da raggiungere risultati più accurati.

I dadaset aggiunti sono i seguenti:

- Il dataset "Messerschmitt" [16], realizzato dal MIT Media Lab, propone 25 immagini in formato .png rappresentanti scene renderizzate, Figura 7.1. Esse mostrano modelli 3D di una vettura del marchio Messerschmitt realizzati dallo Stanford University Computer Graphics Laboratory [17]. Tutti i light fields hanno una risoluzione di 840 x 593 pixel.
- Il dataset "Dice" [18], realizzato dal MIT Media Lab, propone 25 immagini in formato .png rappresentanti scene renderizzate, Figura 7.2. Esse mostrano modelli 3D di dadi da gioco realizzati dallo Stanford University Computer Graphics Laboratory [17]. Tutti i light fields hanno una risoluzione di 840 x 593 pixel.
- Il dataset "Fish" [19], realizzato dal MIT Media Lab, propone 25 immagini in formato .png rappresentanti scene renderizzate, Figura 7.3. Esse mostrano modelli 3D pesci realizzati dallo Stanford University Computer Graphics Laboratory [17]. Tutti i light fields hanno una risoluzione di 840 x 593 pixel.
- Il dataset "Car" [20] realizzato dal Max Planck Institut Informatik. Esso consiste in 101 immagini in formato .png, rappresentanti scene renderizzate, Figura 7.4. Esse presentano scene 3D di una autovettura. Tutti i light fields hanno una risoluzione spaziale di 960 x 720 pixel
- Il dataset "Cobblestone" [21] realizzato dal Max Planck Institut Informatik. Esso consiste in 101 immagini in formato .png, rappresentanti scene renderizzate, Figura 7.5. Esse presentano scene 3D di un sentiero di pietra. Tutti i light fields hanno una risoluzione spaziale di 960 x 720 pixel
- Il dataset "Mannequin" [22] realizzato dal Max Planck Institut Informatik. Esso consiste in 101 immagini in formato .png, rappresentanti scene renderizzate, Figura 7.6. Esse presentano scene 3D di un manichino in una stanza. Tutti i light fields hanno una risoluzione spaziale di 960 x 720 pixel
- Il dataset "Blob" [23] realizzato dal Max Planck Institut Informatik. Esso consiste in 101 immagini in formato .png, rappresentanti scene renderizzate, Figura 7.7. Esse presentano scene 3D di un "blob". Tutti i light fields hanno una risoluzione spaziale di 960 x 720 pixel



Figura 7.1: Immagine estratta dal dataset "Messerschmitt"



Figura 7.2: Immagine estratta dal dataset "Dice"

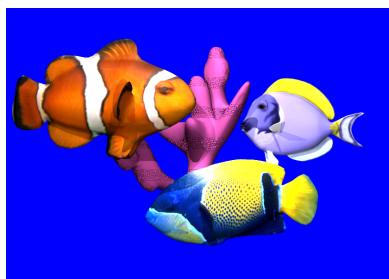


Figura 7.3: Immagine estratta dal dataset "Fish"



Figura 7.4: Immagine estratta dal dataset "Car"

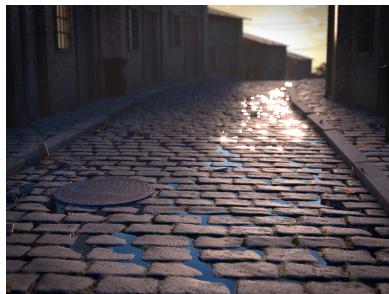


Figura 7.5: Immagine estratta dal dataset "Cobblestone"



Figura 7.6: Immagine estratta dal dataset "Mannequin"



Figura 7.7: Immagine estratta dal dataset "Blob"

CAPITOLO 8

ANALISI DEI RISULTATI

8.1 Configurazione Hardware

Gli esperimenti sono stati condotti utilizzando una configurazione hardware di questo tipo:

- processore Ryzen 5 2600;
- scheda grafica GTX 1660;
- 16 GB di RAM

Questa combinazione di componenti è stata scelta per garantire prestazioni affidabili e rappresentative nell'ambito delle attività di compressione.

8.2 Nomenclatura esperimenti svolti

Nel corso delle nostre ricerche e dei vari esperimenti condotti, abbiamo adottato diversi nomi per distinguere e identificare in modo chiaro i diversi aspetti delle nostre indagini. Per riferirci ai codec video e ai dataset forniti dai nostri stimati colleghi, abbiamo deciso di utilizzare l'anno 2022 come riferimento nei nomi degli esperimenti.

Per quanto riguarda i dataset e i codec video scelti e utilizzati nei nostri studi, abbiamo adottato l'anno 2023 come punto di riferimento nei nomi assegnati a tali risorse.

8.3 Risultati ottenuti

Nelle prossime pagine elenchiamo gli esperimenti proposti, con dimensione originale del dataset inclusa in ogni caption.

8.4 Codec 2022 sui dataset 2022

Lo studio riproduce i test eseguiti in precedenza dai nostri predecessori, ma con l'uso di un codice sorgente modificato. Essendo i codec lossless, non sono stati considerati gli indici SSIM e PSNR.

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
ArtGallery2	HEVC	1.94	20.48s	45.92 MB
ArtGallery2	VP9	1.64	33.54s	54.23 MB
ArtGallery2	AV1	1.77	1791.34s	50.35 MB
ArtGallery2	FFV1	1.36	1.96s	65.35 MB
ArtGallery2	HUFFYUV	1.05	0.28s	84.40 MB
ArtGallery2	UTVIDEO	1.08	0.36s	81.90 MB

Tabella 8.1: Studio codec 2022 sul dataset "ArtGallery" (89.26 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dragons	HEVC	1.80	3.55s	6.53 MB
Dragons	VP9	1.45	10.27s	8.13 MB
Dragons	AV1	1.51	251.60s	7.78 MB
Dragons	FFV1	1.32	0.44s	8.87 MB
Dragons	HUFFYUV	0.74	0.12s	15.77 MB
Dragons	UTVIDEO	0.88	0.14s	13.36 MB

Tabella 8.2: Studio codec 2022 su "Dragons and bunnies" (11.80 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
OpEx	HEVC	1.13	47.13s	65.27 MB
OpEx	VP9	1.09	56.67s	67.41 MB
OpEx	AV1	1.24	2519.93s	59.50 MB
OpEx	FFV1	1.59	7.92s	46.21 MB
OpEx	HUFFYUV	0.42	5.35s	173.65 MB
OpEx	UTVIDEO	0.67	5.19s	109.07 MB

Tabella 8.3: Studio codec 2022 sul dataset "OpEx Room" (73.8 MB)

8.5 Codec 2022 sui dataset 2022 (random)

Questo studio è stato condotto per testare i codec del 2022 sui dataset del 2022, tuttavia con i frame disposti in maniera casuale (random). Essendo i codec lossless, non sono stati considerati gli indici SSIM e PSNR.

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
ArtGallery2_random	HEVC	1.38	24.83s	64.40 MB
ArtGallery2_random	VP9	1.14	37.77s	78.14 MB
ArtGallery2_random	AV1	1.45	1477.19s	61.22 MB
ArtGallery2_random	FFV1	1.36	1.93s	65.37 MB
ArtGallery2_random	HUFFYUV	1.05	0.27s	84.40 MB
ArtGallery2_random	UTVIDEO	1.08	0.35s	81.90 MB

Tabella 8.4: Studio codec 2022 sul dataset **random** "ArtGallery" (89.26 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dragons_random	HEVC	1.55	3.51s	7.59 MB
Dragons_random	VP9	1.31	11.40s	9.00 MB
Dragons_random	AV1	1.43	244.98s	8.23 MB
Dragons_random	FFV1	1.32	0.44s	8.87 MB
Dragons_random	HUFFYUV	0.74	0.11s	15.77 MB
Dragons_random	UTVIDEO	0.88	0.13s	13.36 MB

Tabella 8.5: Studio codec 2022 sul dataset dataset **random** "Dragons and bunnies" (11.80 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
OpEx_random	HEVC	1.00	54.78s	73.70 MB
OpEx_random	VP9	1.03	69.50s	71.11 MB
OpEx_random	AV1	1.16	2797.33s	63.52 MB
OpEx_random	FFV1	1.50	7.95s	48.96 MB
OpEx_random	HUFFYUV	0.39	5.64s	187.65 MB
OpEx_random	UTVIDEO	0.62	4.99s	117.38 MB

Tabella 8.6: Studio codec 2022 sul dataset **random** "OpEx Room" (73.8 MB)

8.6 Codec 2022 sui dataset 2023

Questo studio è stato svolto per testare i codec del 2022 sui nuovi dataset del 2023, implementati da noi. L'obiettivo di questo esperimento è di ottenere una quantità maggiore di dati al fine di ottenere un'idea più completa sulle prestazioni dei codec del 2022. Essendo i codec considerati tutti lossless, non sono stati considerati gli indici SSIM e PSNR.

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Blob	HEVC	2.64	16.23s	20.28 MB
Blob	VP9	1.12	46.23s	47.74 MB
Blob	AV1	2.31	1425.33s	23.19 MB
Blob	FFV1	1.64	1.46s	32.66 MB
Blob	HUFFYUV	0.82	0.28s	65.16 MB
Blob	UTVIDEO	1.00	0.36s	53.50 MB

Tabella 8.7: Studio codec 2022 sul dataset "Blob" (53.65 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Car	HEVC	2.14	15.36s	29.54 MB
Car	VP9	1.40	32.35s	45.08 MB
Car	AV1	1.75	2647.55s	36.20 MB
Car	FFV1	1.75	2.33s	36.14 MB
Car	HUFFYUV	0.97	0.33s	65.43 MB
Car	UTVIDEO	1.19	0.51s	53.14 MB

Tabella 8.8: Studio codec 2022 sul dataset "Car" (63.50 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Cobblestone	HEVC	2.04	21.64s	50.28 MB
Cobblestone	VP9	1.23	39.56s	83.43 MB
Cobblestone	AV1	2.00	1597.50s	51.23 MB
Cobblestone	FFV1	1.36	2.54s	75.26 MB
Cobblestone	HUFFYUV	1.02	0.26s	99.91 MB
Cobblestone	UTVIDEO	1.05	0.34s	97.68 MB

Tabella 8.9: Studio codec 2022 sul dataset "Car" (102.80 MB)

8. ANALISI DEI RISULTATI

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dice	HEVC	1.00	2.93s	3.13 MB
Dice	VP9	0.92	8.07s	3.37 MB
Dice	AV1	0.99	95.82s	3.16 MB
Dice	FFV1	1.44	0.22s	2.16 MB
Dice	HUFFYUV	0.28	0.09s	10.82 MB
Dice	UTVIDEO	0.46	0.11s	66.63 MB

Tabella 8.10: Studio codec 2022 sul dataset "Dice" (3.13 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Fish	HEVC	2.10	2.46s	3.48 MB
Fish	VP9	1.72	5.93s	4.26 MB
Fish	AV1	1.94	125.60s	3.78 MB
Fish	FFV1	1.22	0.36s	5.99 MB
Fish	HUFFYUV	0.51	0.12s	14.19 MB
Fish	UTVIDEO	0.65	0.13s	11.16 MB

Tabella 8.11: Studio codec 2022 sul dataset "Fish" (7.35 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Mannequin	HEVC	1.09	25.17s	72.75 MB
Mannequin	VP9	1.08	40.57s	73.66 MB
Mannequin	AV1	1.26	1962.75s	63.30 MB
Mannequin	FFV1	1.56	1.70s	51.22 MB
Mannequin	HUFFYUV	0.92	0.28s	86.78 MB
Mannequin	UTVIDEO	0.99	0.41s	80.21 MB

Tabella 8.12: Studio codec 2022 sul dataset "Mannequin" (79.96 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Messerschmitt	HEVC	1.37	4.02s	6.46 MB
Messerschmitt	VP9	0.97	11.24s	9.10 MB
Messerschmitt	AV1	1.08	201.96s	8.12 MB
Messerschmitt	FFV1	1.72	0.37s	5.14 MB
Messerschmitt	HUFFYUV	0.75	0.11s	11.75 MB
Messerschmitt	UTVIDEO	1.02	0.13s	8.61 MB

Tabella 8.13: Studio codec 2022 sul dataset "Messerschmitt" (8.85 MB)

8.7 Codec 2022 sui dataset 2023 (random)

Questo studio ha testato i codec del 2022 sui nuovi dataset del 2023, da noi implementati, con i frames randomizzati. Essendo i codec lossless, non sono stati considerati gli indici SSIM e PSNR.

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Blob_random	HEVC	1.68	19.63	31.91 MB
Blob_random	VP9	1.03	42.06s	51.99 MB
Blob_random	AV1	1.96	934.95s	27.32 MB
Blob_random	FFV1	1.64	1.71s	32.68 MB
Blob_random	HUFFYUV	0.82	0.28s	65.16 MB
Blob_random	UTVIDEO	1.00	0.36s	53.50 MB

Tabella 8.14: Studio codec 2022 sul dataset **random** "Blob" (53.65 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Car_random	HEVC	1.40	22.08s	45.30 MB
Car_random	VP9	1.01	41.80s	62.81 MB
Car_random	AV1	1.31	2314.73s	48.13 MB
Car_random	FFV1	1.75	1.51s	36.17 MB
Car_random	HUFFYUV	0.97	0.28s	65.43 MB
Car_random	UTVIDEO	1.19	0.37s	53.14 MB

Tabella 8.15: Studio codec 2022 sul dataset **random** "Car" 63.50 MB

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Cobblestone_random	HEVC	1.38	20.03s	74.08 MB
Cobblestone_random	VP9	0.94	40.91s	108.91 MB
Cobblestone_random	AV1	1.60	1118.97s	64.17 MB
Cobblestone_random	FFV1	1.36	2.59s	75.27 MB
Cobblestone_random	HUFFYUV	1.02	0.29s	99.91 MB
Cobblestone_random	UTVIDEO	1.05	0.36s	97.68 MB

Tabella 8.16: Studio codec 2022 sul dataset **random** "Cobblestone" (102.80 MB)

8. ANALISI DEI RISULTATI

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dice_random	HEVC	0.94	2.51s	3.31 MB
Dice_random	VP9	0.88	8.05s	3.52 MB
Dice_random	AV1	0.97	97.59s	3.20 MB
Dice_random	FFV1	1.44	0.26s	2.16 MB
Dice_random	HUFFYUV	0.28	0.13s	10.82 MB
Dice_random	UTVIDEO	0.46	0.12s	6.66 MB

Tabella 8.17: Studio codec 2022 sul dataset **random** "Dice" (3.13 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Fish_random	HEVC	1.93	2.54s	3.80 MB
Fish_random	VP9	1.61	6.52s	4.56 MB
Fish_random	AV1	1.85	127.43s	3.97 MB
Fish_random	FFV1	1.22	0.32s	5.99 MB
Fish_random	HUFFYUV	0.51	0.11s	14.19 MB
Fish_random	UTVIDEO	0.65	0.13s	11.16 MB

Tabella 8.18: Studio codec 2022 sul dataset **random** "Fish" (7.35 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Mannequin_random	HEVC	0.97	31.77s	81.60 MB
Mannequin_random	VP9	0.91	44.22s	87.02 MB
Mannequin_random	AV1	1.16	1462.18s	68.68 MB
Mannequin_random	FFV1	1.56	1.73s	51.22 MB
Mannequin_random	HUFFYUV	0.92	0.29s	86.78 MB
Mannequin_random	UTVIDEO	0.99	0.34s	80.21 MB

Tabella 8.19: Studio codec 2022 sul dataset **random** "Mannequin" (79.96 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Messerschmitt_random	HEVC	1.06	4.16s	8.35 MB
Messerschmitt_random	VP9	0.84	15.90s	10.46 MB
Messerschmitt_random	AV1	1.03	195.90s	8.53 MB
Messerschmitt_random	FFV1	1.71	0.30s	5.15 MB
Messerschmitt_random	HUFFYUV	0.75	0.11s	11.75 MB
Messerschmitt_random	UTVIDEO	1.02	0.13s	8.61 MB

Tabella 8.20: Studio codec 2022 sul dataset **random** "Massershmitt" (8.85 MB)

8.8 Codec 2023 sui dataset 2022

Questo studio è stato svolto per testare i codec scelti da noi sui dataset del 2022, al fine di valutare l'andamento dei codec sui dataset selezionati dai nostri predecessori.

8.8.1 Codec lossless 2023 sui dataset 2022

Di seguito riportiamo le tabelle (una per dataset) su questo studio utilizzando **codec lossless**. Essendo i codec lossless, non sono stati considerati gli indici SSIM e PSNR.

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
ArtGallery2	MagicYUV	1.03	0.37s	86.17 MB
ArtGallery2	FFVHUFF	1.05	0.31s	84.40 MB
ArtGallery2	LCL	0.75	1.42s	118.07 MB

Tabella 8.21: Studio codec lossless 2023 sul dataset "ArtGallery" (89.26 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dragons	MagicYUV	0.87	0.15s	13.53 MB
Dragons	FFVHUFF	0.74	0.11s	15.77 MB
Dragons	LCL	0.86	0.25s	13.68 MB

Tabella 8.22: Studio codec lossless 2023 sul dataset "Dragons and bunnies" (11.80 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
OpEx	MagicYUV	0.66	5.00s	110.55 MB
OpEx	FFVHUFF	0.42	5.82s	173.65 MB
OpEx	LCL	0.68	5.89s	107.32 MB

Tabella 8.23: Studio codec lossless 2023 sul dataset "OpEx" (73.8 MB)

8.8.2 Codec lossy

Di seguito sono riportate le tabelle relative a questo studio, una per dataset, utilizzando codec lossy. Durante l'esperimento, l'indice SSIM è stato monitorato costantemente per valutare la qualità delle immagini decomprese e confrontare accuratamente le prestazioni dei codec su tali dataset. I risultati della fase di decompressione sono separati dalla fase di compressione, come descritto nella sezione 8.12 del paper.

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
ArtGallery2	FLV1	216.52	0.42s	0.41 MB
ArtGallery2	MJPEG	28.76	0.39s	3.10 MB
ArtGallery2	ProRes	33.26	0.41s	2.68 MB
ArtGallery2	MPEG4	260.42	0.31s	0.34 MB

Tabella 8.24: Studio codec lossy 2023 sul dataset "ArtGallery" (89.26 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dragons	FLV1	89.17	0.16s	0.13 MB
Dragons	MJPEG	14.56	0.15s	0.81 MB
Dragons	ProRes	13.10	0.15s	0.90 MB
Dragons	MPEG4	93.62	0.13s	0.12 MB

Tabella 8.25: Studio codec lossy 2023 sul dataset "Dragons and bunnies" (11.80 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
OpEx	FLV1	40.47	4.82s	1.82 MB
OpEx	MJPEG	15.63	5.46s	4.72 MB
OpEx	ProRes	20.43	4.98s	3.61 MB
OpEx	MPEG4	50.21	4.31s	1.46 MB

Tabella 8.26: Studio codec lossy 2023 sul dataset "OpEx" (73.8 MB)

8.9 Codec 2023 sui dataset 2022 (random)

Questo studio è stato svolto per testare i codec scelti da noi sui dataset del 2022, con i frames disposti in modo randomico, al fine di valutare l'andamento delle prestazioni rispetto al precedente esperimento. Durante l'esperimento, è stato monitorato l'indice SSIM per garantire una valutazione accurata della qualità dell'immagine decompressa e consentire un confronto attendibile delle prestazioni dei codec.

8.9.1 Codec lossless

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
ArtGallery2_random	MagicYUV	1.03	0.36s	86.17 MB
ArtGallery2_random	FFVHUFF	1.05	0.28s	84.40 MB
ArtGallery2_random	LCL	0.75	1.44s	118.07 MB

Tabella 8.27: Studio codec lossless 2023 sul dataset **random** "ArtGallery" (89.26 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dragons_random	MagicYUV	0.87	0.15s	13.53 MB
Dragons_random	FFVHUFF	0.74	0.11s	15.77 MB
Dragons_random	LCL	0.86	0.28s	13.68 MB

Tabella 8.28: Studio codec lossless 2023 sul dataset **random** "Dragons and bunnies" (11.80 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
OpEx_random	MagicYUV	0.62	4.58s	118.86 MB
OpEx_random	FFVHUFF	0.39	5.48s	187.65 MB
OpEx_random	LCL	0.65	5.76s	112.81 MB

Tabella 8.29: Studio codec lossless 2023 sul dataset **random** "OpEx" (73.8 MB)

8.9.2 Codec lossy 2023 sui dataset 2022 random

Di seguito sono riportate le tabelle relative a questo studio, una per dataset, riferite ai risultati della decompressione presentati nella sezione 8.13.

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
ArtGallery2_random	FLV1	92.76	0.50s	0.96 MB
ArtGallery2_random	MJPEG	28.76	0.35s	3.12 MB
ArtGallery2_random	ProRes	33.26	0.43s	2.68 MB
ArtGallery2_random	MPEG4	119.66	0.31s	0.74 MB

Tabella 8.30: Studio codec lossy 2023 sul dataset **random** "ArtGallery" (89.26 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dragons_random	FLV1	72.88	0.17s	0.16 MB
Dragons_random	MJPEG	14.56	0.15s	0.81 MB
Dragons_random	ProRes	13.10	0.15s	0.90 MB
Dragons_random	MPEG4	78.33	0.14s	0.15 MB

Tabella 8.31: Studio codec lossy 2023 sul dataset dataset **random** "Dragons and bunnies" (11.80 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
OpEx_random	FLV1	13.81	5.41s	5.34 MB
OpEx_random	MJPEG	14.66	5.32s	5.03 MB
OpEx_random	ProRes	19.29	4.67s	3.82 MB
OpEx_random	MPEG4	27.66	4.21s	2.66 MB

Tabella 8.32: Studio codec lossy 2023 sul dataset **random** "OpEx" (73.8 MB)

8.10 Codec 2023 sui dataset 2023

Questo studio è stato condotto per testare i codec scelti da noi sui dataset aggiunti da noi (2023). Per quanto riguarda i codec lossy, è stato monitorato l'indice SSIM durante la fase di decompressione al fine di valutare accuratamente la qualità delle immagini risultanti.

8.10.1 Codec lossless

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Blob	MagicYUV	0.96	0.36s	55.59 MB
Blob	FFVHUFF	0.82	0.25s	65.16 MB
Blob	LCL	0.68	1.15s	78.69 MB

Tabella 8.33: Studio codec lossless 2023 sul dataset "Blob" (53.65 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Car	MagicYUV	1.12	0.38s	56.36 MB
Car	FFVHUFF	0.97	0.33s	65.43 MB
Car	LCL	0.89	1.93s	70.78 MB

Tabella 8.34: Studio codec lossless 2023 sul dataset "Car" (63.50 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Cobblestone	MagicYUV	0.99	0.37s	103.38 MB
Cobblestone	FFVHUFF	1.02	0.27s	99.92 MB
Cobblestone	LCL	0.72	1.06s	114.17 MB

Tabella 8.35: Studio codec lossless 2023 sul dataset "Cobblestone" (102.80 MB)

8. ANALISI DEI RISULTATI

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dice	MagicYUV	0.43	0.14s	7.16 MB
Dice	FFVHUFF	0.28	0.10s	10.83 MB
Dice	LCL	0.97	0.15s	31.99MB

Tabella 8.36: Studio codec lossless 2023 sul dataset "Dice" (3.13 MB)"

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Fish	MagicYUV	0.50	0.18s	14.54 MB
Fish	FFVHUFF	0.51	0.13s	14.19 MB
Fish	LCL	0.72	0.19s	10.20 MB

Tabella 8.37: Studio codec lossless 2023 sul dataset "Fish" (7.35 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Mannequin	MagicYUV	0.97	0.37s	82.24 MB
Mannequin	FFVHUFF	0.92	0.29s	86.78 MB
Mannequin	LCL	0.71	1.02s	111.51 MB

Tabella 8.38: Studio codec lossless 2023 sul dataset "Mannequin" (79.96 MB)"

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Messerschmitt	MagicYUV	0.97	0.19s	9.09 MB
Messerschmitt	FFVHUFF	0.75	0.14s	11.7 MB
Messerschmitt	LCL	1.00	0.26s	8.82 MB

Tabella 8.39: Studio codec lossless 2023 sul dataset "Messerschmitt" (8.85 MB)

8.10.2 Codec lossy

Di seguito sono riportate le tabelle relative a questo studio, una per dataset, riferite alla sezione 8.14.

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Blob	FLV1	140.71	0.37s	0.38 MB
Blob	MJPEG	23.09	0.33s	2.32 MB
Blob	ProRes	24.75	0.42s	2.16 MB
Blob	MPEG4	154.85	0.29s	0.34 MB

Tabella 8.40: Studio codec lossy 2023 sul dataset "Blob" (53.65 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Car	FLV1	186.62	0.38s	0.34
Car	MJPEG	23.73	0.44s	2.67 MB
Car	ProRes	25.77	0.48s	2.46 MB
Car	MPEG4	185.23	0.33s	0.34 MB

Tabella 8.41: Studio codec lossy 2023 sul dataset "Car" (63.50 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Cobblestone	FLV1	113.05	0.42s	0.90 MB
Cobblestone	MJPEG	16.40	0.36s	6.26 MB
Cobblestone	ProRes	14.78	0.42s	6.95 MB
Cobblestone	MPEG4	124.57	0.31s	0.82 MB

Tabella 8.42: Studio codec lossy 2023 sul dataset "Cobblestone" (102.80 MB)

8. ANALISI DEI RISULTATI

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dice	FLV1	21.09	0.17s	0.14 MB
Dice	MJPEG	5.76	0.13s	0.54 MB
Dice	ProRes	5.60	0.15s	0.56 MB
Dice	MPEG4	23.30	0.12s	0.13 MB

Tabella 8.43: Studio codec lossy 2023 sul dataset "Dice" (3.13 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Fish	FLV1	67.31	0.15s	0.11 MB
Fish	MJPEG	11.36	0.14s	0.64 MB
Fish	ProRes	9.66	0.15s	0.76 MB
Fish	MPEG4	38.84	0.13s	0.20 MB

Tabella 8.44: Studio codec lossy 2023 sul dataset "Fish" (7.35 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Mannequin	FLV1	174.08	0.38s	0.46 MB
Mannequin	MJPEG	22.22	0.35s	3.60 MB
Mannequin	ProRes	22.36	0.44s	3.57 MB
Mannequin	MPEG4	201.65	0.31s	0.39 MB

Tabella 8.45: Studio codec lossy 2023 sul dataset "Mannequin" (79.96 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Messerschmitt	FLV1	26.55	0.20s	0.33 MB
Messerschmitt	MJPEG	11.73	0.15s	0.75 MB
Messerschmitt	ProRes	10.70	0.15s	0.82 MB
Messerschmitt	MPEG4	34.70	0.13s	0.25 MB

Tabella 8.46: Studio codec lossy 2023 sul dataset "Messerschmitt" (8.85 MB)

8.11 Codec 2023 sui dataset 2023 (random)

Questo studio è stato svolto per testare i codec scelti da noi sui dataset aggiunti da noi nel 2023, con i frames disposti in modo randomico, al fine di valutare l'andamento delle prestazioni rispetto agli esperimenti precedenti.

8.11.1 Codec lossless

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Blob_random	MagicYUV	0.96	0.38s	53.01 MB
Blob_random	FFVHUFF	0.82	0.27s	65.16 MB
Blob_random	LCL	0.68	1.15s	78.69 MB

Tabella 8.47: Studio codec lossless 2023 sul dataset **random** "Blob" (53.65 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Car_random	MagicYUV	1.13	0.38s	56.36 MB
Car_random	FFVHUFF	0.97	0.29s	65.43 MB
Car_random	LCL	0.90	1.83s	70.78 MB

Tabella 8.48: Studio codec lossless 2023 sul dataset **random** "Car" (63.50 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Cobblestone_random	MagicYUV	0.99	0.36s	103.38 MB
Cobblestone_random	FFVHUFF	1.03	0.28s	99.91 MB
Cobblestone_random	LCL	0.72	1.09s	141.76 MB

Tabella 8.49: Studio codec lossless 2023 sul dataset **random** "Cobblestone" (102.80 MB)

8. ANALISI DEI RISULTATI

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dice_random	MagicYUV	0.44	0.13s	7.14 MB
Dice_random	FFVHUFF	0.29	0.10s	10.82 MB
Dice_random	LCL	0.98	0.15s	3.20 MB

Tabella 8.50: Studio codec lossless 2023 sul dataset **random** "Dice" (3.13 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Fish_random	MagicYUV	0.50	0.15s	14.54 MB
Fish_random	FFVHUFF	0.52	0.13s	14.20 MB
Fish_random	LCL	0.72	0.21s	10.19 MB

Tabella 8.51: Studio codec lossless 2023 sul dataset **random** "Fish" (7.35 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Mannequin_random	MagicYUV	0.97	0.38s	82.24 MB
Mannequin_random	FFVHUFF	0.92	0.30s	86.78 MB
Mannequin_random	LCL	0.72	0.99s	111.51 MB

Tabella 8.52: Studio codec lossless 2023 sul dataset **random** "Mannequin" (79.96 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Messerschmitt_random	MagicYUV	0.97	0.15s	90.96 MB
Messerschmitt_random	FFVHUFF	0.75	0.13s	117.56 MB
Messerschmitt_random	LCL	1.00	0.26s	8.82 MB

Tabella 8.53: Studio codec lossless 2023 sul dataset **random** "Messerschmitt" (8.85 MB)

8.11.2 Codec lossy

Durante l'esperimento, abbiamo tenuto sotto stretto controllo la fase di decompressione per mantenere costanti gli indici SSIM per ciascun dataset. Questo approccio ci ha permesso di assicurare un confronto accurato delle prestazioni dei codec, garantendo che la qualità dell'immagine decompressa rimanesse uniforme tra i diversi dataset. I risultati dettagliati di questa fase sono riportati nella sezione 8.15 del nostro studio.

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Blob_random	FLV1	27.95	0.51s	19.19 MB
Blob_random	MJPEG	23.10	0.33s	23.24 MB
Blob_random	ProRes	24.75	0.40s	21.67 MB
Blob_random	MPEG4	48.96	0.32s	10.95 MB

Tabella 8.54: Studio codec lossy 2023 sul dataset **random** "Blob" (53.65 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Car_random	FLV1	53.64	0.54s	11.83 MB
Car_random	MJPEG	23.73	0.35s	26.76 MB
Car_random	ProRes	25.77	0.42s	24.64 MB
Car_random	MPEG4	72.11	0.32s	8.80 MB

Tabella 8.55: Studio codec lossy 2023 sul dataset **random** "Car" (63.50 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Cobblestone_random	FLV1	32.26	0.52s	3.18 MB
Cobblestone_random	MJPEG	16.40	0.35s	6.27 MB
Cobblestone_random	ProRes	14.78	0.44s	6.95 MB
Cobblestone_random	MPEG4	49.67	0.32s	2.06 MB

Tabella 8.56: Studio codec lossy 2023 sul dataset **random** "Cobblestone" (102.80 MB)

8. ANALISI DEI RISULTATI

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Dice_random	FLV1	19.01	0.17s	1.64 MB
Dice_random	MJPEG	5.76	0.13s	5.44 MB
Dice_random	ProRes	5.60	0.15s	5.59 MB
Dice_random	MPEG4	20.77	0.12s	1.50 MB

Tabella 8.57: Studio codec lossy 2023 sul dataset **random** "Dice" (3.13 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Fish_random	FLV1	60.93	0.15s	0.12 MB
Fish_random	MJPEG	11.36	0.14s	0.64 MB
Fish_random	ProRes	9.66	0.15s	0.76 MB
Fish_random	MPEG4	36.35	0.13s	0.20 MB

Tabella 8.58: Studio codec lossy 2023 sul dataset **random** "Fish" (7.35 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Mannequin_random	FLV1	50.63	0.48s	15.79 MB
Mannequin_random	MJPEG	22.22	0.35s	36.00 MB
Mannequin_random	ProRes	22.36	0.42s	35.76 MB
Mannequin_random	MPEG4	59.25	0.32s	13.50 MB

Tabella 8.59: Studio codec lossy 2023 sul dataset **random** "Mannequin" (79.96 MB)

Dataset	Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale
Messerschmitt_random	FLV1	19.39	0.18s	0.45 MB
Messerschmitt_random	MJPEG	11.73	0.14s	0.75 MB
Messerschmitt_random	ProRes	10.70	0.15s	0.83 MB
Messerschmitt_random	MPEG4	23.84	0.14s	0.37 MB

Tabella 8.60: Studio codec lossy 2023 sul dataset **random** "Messerschmitt" (8.85 MB)

8.12 Decompressione codec 2023 su Dataset 2022

Per ciascun dataset del 2022, è stato eseguito uno studio per valutare l'efficacia della decompressione utilizzando i codec selezionati. Si è lavorato per mantenere costante l'indice SSIM per ogni dataset, al fine di consentire un confronto delle prestazioni dei codec in condizioni di parità di qualità dell'immagine decompressa. È stata presa in considerazione la configurazione di un indice SSIM di riferimento diverso per ciascun dataset, poiché in alcuni casi al di sotto di una certa qualità non è stato possibile configurare il codec.

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Average SSIM	Average PSNR
ArtGallery2	FLV1	0.93	35.96
ArtGallery2	MJPEG	0.93	36.49
ArtGallery2	ProRes	0.93	34.80
ArtGallery2	MPEG4	0.93	35.99

Tabella 8.61: Studio decompressione codec 2023 sul dataset "Art Gallery" (89.26 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Dragons	FLV1	0.90	32.63
Dragons	MJPEG	0.90	32.43
Dragons	ProRes	0.90	32.34
Dragons	MPEG4	0.90	32.53

Tabella 8.62: Studio decompressione codec 2023 sul dataset "Dragons and bunnies" (11.80 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
OpEx	FLV1	0.98	42.83
OpEx	MJPEG	0.98	48.55
OpEx	ProRes	0.98	43.33
OpEx	MPEG4	0.98	42.67

Tabella 8.63: Studio decompressione codec 2023 sul dataset "OpEx" (73.8 MB)

8.13 Decompressione codec 2023 su Dataset 2022 (random)

Questo studio è stato condotto per testare la decompressione dei codec scelti da noi sui dataset del 2022, in cui i frame sono stati disposti in modo randomico. Si è cercato di mantenere costante l'indice SSIM per ogni dataset al fine di consentire un confronto delle prestazioni dei codec in condizioni di parità di qualità dell'immagine decompressa.

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Average SSIM	Average PSNR
ArtGallery2_random	FLV1	0.93	35.58
ArtGallery2_random	MJPEG	0.93	36.49
ArtGallery2_random	ProRes	0.93	34.80
ArtGallery2_random	MPEG4	0.93	35.57

Tabella 8.64: Studio decompressione codec 2023 sul dataset **random** "Art Gallery" (89.26 MB)

Dataset	Algoritmo	Average SSIM	Average PSN
Dragons_random	FLV1	0.90	32.47
Dragons_random	MJPEG	0.90	32.43
Dragons_random	ProRes	0.90	32.34
Dragons_random	MPEG4	0.90	32.38

Tabella 8.65: Studio decompressione codec 2023 dataset **random** "Dragons & bunnies" (11.80 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
OpEx_random	FLV1	0.98	43.41
OpEx_random	MJPEG	0.98	48.70
OpEx_random	ProRes	0.98	43.41
OpEx_random	MPEG4	0.98	42.82

Tabella 8.66: Studio decompressione codec 2023 sul dataset **random** "OpEx" (73.8 MB)

8.14 Decompressione codec 2023 su Dataset 2023

Questo studio è stato svolto per testare la decompressione dei codec scelti da noi sui dataset aggiunti da noi. Si è cercato di mantenere costante l'indice SSIM per ogni dataset al fine di consentire un confronto delle prestazioni dei codec in condizioni di parità di qualità dell'immagine decompressa.

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Average SSIM	Average PSNR
Blob	FLV1	0.93	37.27
Blob	MJPEG	0.93	37.76
Blob	ProRes	0.93	37.15
Blob	MPEG4	0.93	37.37

Tabella 8.67: Studio decompressione codec 2023 sul dataset "Blob" (53.65 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Car	FLV1	0.94	35.34
Car	MJPEG	0.94	35.52
Car	ProRes	0.94	34.84
Car	MPEG4	0.94	35.77

Tabella 8.68: Studio decompressione codec 2023 sul dataset "Car" (53.65 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Cobblestone	FLV1	0.90	31.73
Cobblestone	MJPEG	0.90	31.21
Cobblestone	ProRes	0.90	31.21
Cobblestone	MPEG4	0.90	31.63

Tabella 8.69: Studio decompressione codec 2023 sul dataset "Cobblestone" (102.80 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Dice	FLV1	0.95	35.24
Dice	MJPEG	0.95	36.30
Dice	ProRes	0.95	34.69
Dice	MPEG4	0.95	35.84

Tabella 8.70: Studio decompressione codec 2023 sul dataset "Dice" (3.13 MB)

8. ANALISI DEI RISULTATI

Dataset	Algoritmo	Average SSIM	Average PSNR
Fish	FLV1	0.95	33.96
Fish	MJPEG	0.95	33.59
Fish	ProRes	0.95	33.43
Fish	MPEG4	0.95	33.81

Tabella 8.71: Studio decompressione codec 2023 sul dataset "Fish" (7.35 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Mannequin	FLV1	0.90	32.89
Mannequin	MJPEG	0.90	32.97
Mannequin	ProRes	0.90	32.67
Mannequin	MPEG4	0.90	32.79

Tabella 8.72: Studio decompressione codec 2023 sul dataset "Mannequin" (79.96 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Messerschmitt	FLV1	0.90	35.20
Messerschmitt	MJPEG	0.90	35.32
Messerschmitt	ProRes	0.90	34.97
Messerschmitt	MPEG4	0.90	35.11

Tabella 8.73: Studio decompressione codec 2023 sul dataset "Messerschmitt" (8.85 MB)

8.15 Decompressione codec 2023 su Dataset 2023 (random)

Questo studio è stato svolto per testare la decompressione dei codec scelti da noi sui dataset aggiunti da noi, disponendo i frame in modo randomico. Si è cercato di mantenere costante l'indice SSIM per ogni dataset al fine di consentire un confronto delle prestazioni dei codec in condizioni di parità di qualità dell'immagine decompressa.

Di seguito riportiamo le tabelle (una per dataset) su questo studio:

Dataset	Algoritmo	Average SSIM	Average PSNR
Blob_random	FLV1	0.93	37.62
Blob_random	MJPEG	0.93	37.76
Blob_random	ProRes	0.93	37.15
Blob_random	MPEG4	0.93	37.28

Tabella 8.74: Studio decompressione codec 2023 sul dataset **random** "Blob" (53.65 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Car_random	FLV1	0.94	34.79
Car_random	MJPEG	0.94	35.52
Car_random	ProRes	0.94	34.84
Car_random	MPEG4	0.94	34.92

Tabella 8.75: Studio decompressione codec 2023 sul dataset **random** "Car" (63.50 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Cobblestone_random	FLV1	0.90	31.38
Cobblestone_random	MJPEG	0.90	31.21
Cobblestone_random	ProRes	0.90	31.21
Cobblestone_random	MPEG4	0.90	31.05

Tabella 8.76: Studio decompressione codec 2023 dataset **random** "Cobblestone" (102.80 MB)

8. ANALISI DEI RISULTATI

Dataset	Algoritmo	Average SSIM	Average PSNR
Dice_random	FLV1	0.95	35.11
Dice_random	MJPEG	0.95	36.30
Dice_random	ProRes	0.95	34.69
Dice_random	MPEG4	0.95	35.72

Tabella 8.77: Studio decompressione codec 2023 sul dataset **random** "Dice" (3.13 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Fish_random	FLV1	0.95	33.80
Fish_random	MJPEG	0.95	33.59
Fish_random	ProRes	0.95	33.43
Fish_random	MPEG4	0.95	33.62

Tabella 8.78: Studio decompressione codec 2023 sul dataset **random** "Fish" (7.35 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Mannequin_random	FLV1	0.90	32.75
Mannequin_random	MJPEG	0.90	32.97
Mannequin_random	ProRes	0.90	32.67
Mannequin_random	MPEG4	0.90	33.03

Tabella 8.79: Studio decompressione codec 2023 sul dataset **random** "Mannequin" (79.96 MB)

Dataset	Algoritmo	Average SSIM	Average PSNR
Messerschmitt_random	FLV1	0.90	35.01
Messerschmitt_random	MJPEG	0.90	35.32
Messerschmitt_random	ProRes	0.90	34.97
Messerschmitt_random	MPEG4	0.90	34.90

Tabella 8.80: Studio decompressione codec 2023 dataset **random** "Messerschmitt" (8.85 MB)

8.16 Confronto su diversi livelli di SSIM

Lo studio ha condotto un confronto delle prestazioni dei codec lossy su diversi livelli di qualità dell'immagine, fissando l'indice SSIM su tre livelli (0.85, 0.90, 0.95) per i dataset considerati e relativi dataset randomizzati. Questo approccio ha consentito di valutare le prestazioni dei codec mantenendo costante la qualità dell'immagine. Tuttavia, a causa del parametro -q:v di ffmpeg impostato al massimo, in alcuni casi non è stato possibile ottenere un indice SSIM inferiore a 0.90, il che ha influenzato la valutazione della qualità dell'immagine a questo livello.

I codec lossy che non sono presenti nei risultati dello studio potrebbero non esserlo perché non è stato possibile configurarli correttamente per garantire una qualità di compressione adeguata. Questo potrebbe essere dovuto a limitazioni tecniche specifiche dei codec stessi o a difficoltà nell'impostare i parametri di compressione in modo ottimale per i test condotti nello studio. Inoltre, il dataset Opex non è stato incluso nell'analisi a causa della sua qualità di base inferiore rispetto agli altri dataset, presumibilmente a causa di immagini di base corrotte o altri problemi tecnici che hanno reso difficile l'analisi della compressione su questo set di dati.

8.16.1 Cobblestone

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	199.14	0.40s	0.51 MB	0.85	29.52
MJPEG	23.12	0.37s	4.44 MB	0.85	29.30
ProRes	24.36	0.45s	4.42 MB	0.85	29.20
MPEG4	230.48	0.32s	0.44 MB	0.85	29.45

Tabella 8.81: Codec lossy 2023 sul dataset "Cobblestone" (102.80 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	113.05	0.42s	0.90 MB	0.90	31.73
MJPEG	16.40	0.36s	6.26 MB	0.90	31.20
ProRes	14.78	0.42s	6.95 MB	0.90	31.20
MPEG4	124.57	0.31s	0.82 MB	0.90	31.63

Tabella 8.82: Codec lossy 2023 sul dataset "Cobblestone" (102.80 MB)

8. ANALISI DEI RISULTATI

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	55.44	0.51s	1.85 MB	0.95	34.50
MJPEG	10.12	0.38s	10.15 MB	0.95	34.40
ProRes	7.86	0.44s	13.06 MB	0.95	34.60
MPEG4	60.29	0.32s	1.70 MB	0.95	34.40

Tabella 8.83: Codec lossy 2023 sul dataset "Cobblestone" (102.80 MB)

8.16.2 Cobblestone Random

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	48.04	0.51s	2.13 MB	0.85	29.40
CMJPEG	23.12	0.34s	4.44 MB	0.85	29.30
ProRes	24.36	0.54s	4.42 MB	0.85	29.20
MPEG4	76.41	0.36s	1.34 MB	0.85	29.45

Tabella 8.84: Codec lossy 2023 sul dataset "Cobblestone" **random** (102.80 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	32.26	0.52s	3.18 MB	0.90	31.40
MJPEG	16.40	0.35s	6.27 MB	0.90	31.21
ProRes	14.78	0.44s	6.95 MB	0.90	31.21
MPEG4	49.67	0.32s	2.06 MB	0.90	31.05

Tabella 8.85: Codec lossy 2023 sul dataset "Cobblestone" **random** (102.80 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	15.33	0.74s	6.70 MB	0.95	34.42
MJPEG	10.12	0.44s	10.15 MB	0.95	34.40
ProRes	7.86	0.45s	13.06 MB	0.95	34.60
MPEG4	20.46	0.34s	5.02 MB	0.95	34.05

Tabella 8.86: Codec lossy 2023 sul dataset "Cobblestone" **random** (102.80 MB)

8.16.3 Messerschmitt

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	49.44	0.14s	0.17 MB	0.85	32.52
MJPEG	16.86	0.13s	0.52 MB	0.85	32.42
ProRes	16.74	0.15s	0.52 MB	0.85	32.54
MPEG4	60.84	0.16s	0.14 MB	0.85	32.74

Tabella 8.87: Codec lossy 2023 sul dataset "Messerschmitt" (8.85 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	26.55	0.20s	0.33 MB	0.90	35.20
MJPEG	11.73	0.15s	0.75 MB	0.90	35.32
ProRes	10.70	0.15s	0.82 MB	0.90	34.97
MPEG4	34.70	0.13s	0.25 MB	0.90	35.11

Tabella 8.88: Codec lossy 2023 sul dataset "Messerschmitt" (8.85 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	10.81	0.20s	0.81 MB	0.95	38.51
MJPEG	6.91	0.15s	1.28 MB	0.95	38.69
ProRes	5.56	0.23s	1.59 MB	0.95	38.50
MPEG4	13.35	0.14s	0.66 MB	0.95	38.35

Tabella 8.89: Codec lossy 2023 sul dataset "Messerschmitt" (8.85 MB)

8.16.4 Messerschmitt Random

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	32.35	0.16s	0.27 MB	0.85	32.57
MJPEG	16.86	0.13s	0.52 MB	0.85	32.42
ProRes	16.74	0.14s	0.52 MB	0.85	32.54
MPEG4	40.41	0.13s	0.21 MB	0.85	32.75

Tabella 8.90: Codec lossy 2023 sul dataset "Messerschmitt" **random** (8.85 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	19.39	0.18s	0.45 MB	0.90	35.01
MJPEG	11.73	0.14s	0.75 MB	0.90	35.32
ProRes	10.70	0.15s	0.83 MB	0.90	34.97
MPEG4	23.84	0.14s	0.37 MB	0.90	34.90

Tabella 8.91: Codec lossy 2023 sul dataset "Messerschmitt" **random** (8.85 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	8.05	0.19s	1.10 MB	0.95	38.41
MJPEG	6.91	0.16s	1.28 MB	0.95	38.69
ProRes	5.56	0.17s	1.59 MB	0.95	38.50
MPEG4	9.11	0.15s	0.97 MB	0.95	38.28

Tabella 8.92: Codec lossy 2023 sul dataset "Messerschmitt" **random** (8.85 MB)

8.16.5 Dragons

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	143.28	0.12s	0,08 MB	0.85	29.98
MJPEG	18.51	0.15s	6,37 MB	0.85	30.58
MPEG4	157.72	0.11s	0.07 MB	0.85	30.05

Tabella 8.93: Codec lossy 2023 sul dataset "Dragons and bunnies" (11.80 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	89.17	0.16s	0.13 MB	0.90	32.63
MJPEG	14.55	0.15s	0.81 MB	0.90	32.42
MPEG4	93.62	0.13s	0.12 MB	0.90	32.53

Tabella 8.94: Codec lossy 2023 sul dataset "Dragons and bunnies" (11.80 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	31.20	0.14s	0.37 MB	0.95	36.77
MJPEG	7.49	0.12s	1.57 MB	0.95	36.81
MPEG4	33.28	0.11s	0.35 MB	0.95	36.64

Tabella 8.95: Codec lossy 2023 sul dataset "Dragons and bunnies" (11.80 MB)

8.16.6 Dragons_random

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	119.39	0.13s	0.09 MB	0.85	29.87
MJPEG	18.51	0.12s	0.63 MB	0.85	30.58
MPEG4	133.39	0.11s	0.08 MB	0.85	29.93

Tabella 8.96: Codec lossy 2023 sul dataset **random** "Dragons and bunnies" (11.80 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	72.87	0.16s	0.16 MB	0.90	32.47
MJPEG	14.55	0.15s	0.81 MB	0.90	32.42
MPEG4	78.33	0.14s	0.15 MB	0.90	32.38

Tabella 8.97: Codec lossy 2023 sul dataset **random** "Dragons and bunnies" (11.80 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	25.57	0.15s	0,46 MB	0.95	36.62
MJPEG	7.49	0.14s	1.57 MB	0.95	36.81
MPEG4	27.43	0.11s	0.43 MB	0.95	36.52

Tabella 8.98: Codec lossy 2023 sul dataset **random** "Dragons and bunnies" (11.80 MB)

8.16.7 Mannequin

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	258.09	0.54s	0.30 MB	0.85	30.58
MJPEG	28.63	0.75s	2.79 MB	0.85	30.68
MPEG4	317.49	0.48s	0.25 MB	0.85	30.66

Tabella 8.99: Codec lossy 2023 sul dataset "Mannequin" (79.96 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	174.07	0.38s	0.45 MB	0.90	32.89
MJPEG	22.21	0.35s	3.59 MB	0.90	32.97
MPEG4	201.64	0.30s	0.39 MB	0.90	32.79

Tabella 8.100: Codec lossy 2023 sul dataset "Mannequin" (79.96 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	77.85	0.54s	1.02 MB	0.95	36.50
MJPEG	13.56	0.58s	5.89 MB	0.95	36.81
MPEG4	85.70	0.51s	0.93 MB	0.95	36.34

Tabella 8.101: Codec lossy 2023 sul dataset "Mannequin" (79.96 MB)

8.16.8 Mannequin_random

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	79.55	0.60s	1.00 MB	0.85	30.40
MJPEG	28.63	0.59s	2.79 MB	0.85	30.68
MPEG4	106.39	0.46s	0.75 MB	0.85	30.51

Tabella 8.102: Codec lossy 2023 sul dataset **random** "Mannequin" (79.96 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	50.63	0.48s	1.57 MB	0.90	32.75
MJPEG	22.21	0.34s	3.59 MB	0.90	32.97
MPEG4	59.24	0.31s	1.34 MB	0.90	33.03

Tabella 8.103: Codec lossy 2023 sul dataset **random** "Mannequin" (79.96 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	21.11	0.73s	3.78 MB	0.95	37.16
MJPEG	13.56	0.58s	5.89 MB	0.95	36.81
MPEG4	25.22	0.53s	3.16 MB	0.95	37.01

Tabella 8.104: Codec lossy 2023 sul dataset **random** "Mannequin" (79.96 MB)

8.16.9 ArtGallery2

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	342.90	0.52s	0.26 MB	0.90	32.37
MJPEG	36.22	0.48s	2.46 MB	0.90	32.65
MPEG4	441.03	0.42s	0.20 MB	0.90	32.60

Tabella 8.105: Codec lossy 2023 sul dataset "Art Gallery" (89.26 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	100.52	0.52s	0.89 MB	0.95	39.39
MJPEG	19.16	0.57s	4.65 MB	0.95	40.05
MPEG4	123.99	0.52s	0.71 MB	0.95	39.27

Tabella 8.106: Codec lossy 2023 sul dataset "Art Gallery" (89.26 MB)

8.16.10 ArtGallery2_random

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	154.24	0.55s	0.57 MB	0.90	31.92
MJPEG	36.22	0.64s	2.46 MB	0.90	32.65
MPEG4	209.45	0.50s	0.42 MB	0.90	32.14

Tabella 8.107: Codec lossy 2023 sul dataset **random** "Art Gallery" (89.26 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	44.61	0.60s	2.00 MB	0.95	39.18
MJPEG	19.16	0.55s	4.65 MB	0.95	40.05
MPEG4	56.31	0.53s	1.58 MB	0.95	39.06

Tabella 8.108: Codec lossy 2023 sul dataset **random** "Art Gallery" (89.26 MB)

8.16.11 Blob

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	167.06	0.65s	0.32 MB	0.90	35.09
MJPEG	25.30	0.59s	2.12 MB	0.90	35.03
MPEG4	209.39	0.56s	0.25 MB	0.90	34.93

Tabella 8.109: Codec lossy 2023 sul dataset "Blob" (53.65 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	117.24	0.48s	0.45 MB	0.95	39.12
MJPEG	19.59	0.46s	2.73 MB	0.95	39.99
MPEG4	127.63	0.44s	0.42 MB	0.95	39.27

Tabella 8.110: Codec lossy 2023 sul dataset dataset "Blob" (53.65 MB)

8.16.12 Blob_random

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	35.66	2.02s	1.50 MB	0.90	35.20
MJPEG	25.30	0.88s	2.12 MB	0.90	35.03
MPEG4	62.63	0.80s	0.85 MB	0.90	35.62

Tabella 8.111: Codec lossy 2023 sul dataset **random** "Blob" (53.65 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	22.31	0.73s	2.40 MB	0.95	39.58
MJPEG	19.00	0.52s	2.82 MB	0.95	39.35
MPEG4	35.32	0.46s	1.51 MB	0.95	39.44

Tabella 8.112: Codec lossy 2023 sul dataset **random** "Blob" (53.65 MB)

8.16.13 Car

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	252.45	0.38s	0.25 MB	0.90	31.73
MPEG4	258.16	0.29s	0.24 MB	0.90	32.05

Tabella 8.113: Codec lossy 2023 sul dataset "Car" (63.50 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	172.45	0.36s	0.36 MB	0.95	35.00
MPEG4	168.72	0.46s	0.37 MB	0.95	39.44

Tabella 8.114: Codec lossy 2023 sul dataset "Car" (63.50 MB)

8.16.14 Car_random

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	88.47	0.41s	0.71 MB	0.90	31.39
MPEG4	112.77	0.80s	0.56 MB	0.90	31.68

Tabella 8.115: Codec lossy 2023 sul dataset **random** "Car" (63.50 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	46.81	0.48s	1.35 MB	0.95	35.87
MPEG4	60.74	0.39s	1.04 MB	0.95	35.82

Tabella 8.116: Codec lossy 2023 sul dataset **random** "Car" (63.50 MB)

8.16.15 Dice

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	32.25	0.13s	0.09 MB	0.90	30.88
MPEG4	43.62	0.12s	0.07 MB	0.90	31.09

Tabella 8.117: Codec lossy 2023 sul dataset "Dice" (3.13 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	21.08	0.17s	0.15 MB	0.95	35.23
MPEG4	23.29	0.12s	0.13 MB	0.95	36.30

Tabella 8.118: Codec lossy 2023 sul dataset "Dice" (3.13 MB)

8.16.16 Dice_random

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	29.94	0.16s	0.10 MB	0.90	30.75
MPEG4	38.58	0.13s	0.08 MB	0.90	31.02

Tabella 8.119: Codec lossy 2023 sul dataset **random** "Dice" (3.13 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	19.00	0.16s	0.16 MB	0.95	35.11
MPEG4	20.76	0.12s	0.15 MB	0.95	35.72

Tabella 8.120: Codec lossy 2023 sul dataset random "Dice" (3.13 MB)

8.16.17 Fish

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	92.70	0.14s	0.08 MB	0.90	29.98
MPEG4	101.43	0.14s	0.07 MB	0.90	29.97

Tabella 8.121: Codec lossy 2023 sul dataset "Fish" (7.35 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	67.30	0.15s	0.10 MB	0.95	33.96
MPEG4	38.84	0.13s	0.18 MB	0.95	33.81

Tabella 8.122: Codec lossy 2023 sul dataset "Fish" (7.35 MB)

8.16.18 Fish_random

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	73.41	0.16s	0.10 MB	0.90	29.89
MPEG4	87.48	0.13s	0.08 MB	0.90	29.86

Tabella 8.123: Codec lossy 2023 sul dataset **random** "Fish" (7.35 MB)

Algoritmo	Rapporto compressione	Tempo compressione	Dimensione finale	Average SSIM	Average PSNR
FLV1	60.93	0.15s	0.12 MB	0.95	33.79
MPEG4	36.35	0.13s	0.20 MB	0.95	33.62

Tabella 8.124: Codec lossy 2023 sul dataset **random** "Fish" (7.35 MB)

8.17 Analisi riassuntiva dei risultati

Nel primo studio, abbiamo adottato un approccio con SSIM fisso per ciascun dataset, il che significa che abbiamo mantenuto costante il valore SSIM durante le compressioni eseguite con i codec selezionati. Abbiamo regolato manualmente il valore SSIM per ogni dataset considerato, assicurandoci che rimanesse invariato durante i test. Inoltre, abbiamo utilizzato il parametro ”-q:v” per regolare la qualità delle compressioni al fine di valutare le prestazioni dei codec. Nel corso di questo studio, abbiamo osservato che i codec MPEG4 e FLV1 si sono distinti tra i vari codec in termini di rapporto di compressione, tempo e qualità delle immagini compresse.

Durante il secondo studio, i codec MPEG4 e FLV1 sono emersi in modo costante nei risultati delle tabelle, suggerendo la loro affidabilità e flessibilità in svariati contesti di compressione. Tuttavia, è emersa una sfida nel gestire il dataset Opex. Questo dataset potrebbe presentare una qualità di base inferiore rispetto agli altri dataset considerati, il che ha complicato il processo di gestione e potrebbe aver influenzato i risultati dell’analisi. Nel corso dello studio, abbiamo fissato il valore SSIM su tre livelli: 0.85, 0.90 e 0.95, consentendo una panoramica completa delle prestazioni dei codec in condizioni diverse. Tale impostazione ci ha permesso di confrontare le prestazioni dei codec mantenendo costante il livello di qualità dell’immagine. Tuttavia, è emerso che in alcuni casi non è stato possibile ottenere un indice SSIM inferiore a 0.90 a causa di parametri impostati al massimo. Nonostante ciò, lo studio ha fornito comunque una visione approfondita delle prestazioni dei codec lossy, consentendo di analizzare il rapporto tra qualità dell’immagine e rapporto di compressione.

8.17.1 Analisi sui dataset random

Di seguito analizziamo le prestazioni dei vari codec di compressione sui rispettivi dataset randomizzati. In generale abbiamo notato che per ogni esperimento c'è un peggioramento importante su alcuni codec mentre altri sembrano soffrire di meno la disposizione dei frames in ordine randomico. Nello specifico i codec che peggiorano la loro performance, sia in termini di velocità di esecuzione e sia in termini di compression rate:

- HEVC
- VP9
- FLV1
- MPEG4

Per i codec non citati sopra, le prestazioni risultano essere molto vicine a quelle sui dataset originali, anche se, in alcuni casi, di poco peggiori.

Si può quindi affermare che **c'è effettivamente dipendenza** tra i frame dei vari Light Field. Quindi l'approccio di compressione/decompressione degli stessi, deve tener conto di questa dipendenza tra i frame per ottenere una compressione efficiente senza compromettere la qualità dell'immagine riprodotta. Gli esperimenti condotti sui dataset con frame disposti casualmente hanno evidenziato dei peggioramenti sia in fase di compressione, in termini di tempi e rapporto di compressione, sia in fase di decompressione, in termini di similarità e rapporto segnale/rumore.

CAPITOLO 9

CONCLUSIONE E SVILUPPI FUTURI

La conclusione dello studio enfatizza l'importanza fondamentale della scelta accurata del dataset e del codec nell'ambito della compressione delle immagini light field, poiché tali decisioni influenzano in modo significativo le prestazioni complessive. L'analisi condotta ha evidenziato variazioni sostanziali nell'efficacia della compressione in relazione ai diversi dataset impiegati, con anche l'ordine dei frame che può influire sulla qualità e sulle prestazioni generali. Inoltre, si sottolinea come la scelta tra codec lossy e lossless giochi un ruolo cruciale.

Da un altro punto di vista, si è riscontrato che codec come MPEG4 e FLV1 sono stati capaci di bilanciare in maniera ottimale tutte le variabili coinvolte nella compressione, offrendo eccellenti rapporti di compressione unitamente a una qualità visiva e una fedeltà strutturale notevoli. Questi codec si distinguono per la loro efficienza nello spazio di archiviazione, mantenendo allo stesso tempo una qualità dell'immagine elevata.

Questi risultati evidenziano la complessità delle decisioni nel campo della compressione video, poiché diversi codec possono eccellere in modi diversi.

Nel corso degli esperimenti condotti, è fondamentale notare che è stato stabilito un valore di similarità utilizzando l'indice SSIM per ciascun dataset. Questo indice valuta le differenze nell'intensità, nella struttura e nella luminanza tra due immagini, offrendo una misurazione affidabile della loro somiglianza. Attraverso questo approccio, è stato possibile determinare che, in termini di rapporto di compressione e qualità dell'immagine, i codec migliori sono risultati essere MPEG4, seguito da FLV1. L'inclusione del PSNR ha ulteriormente confermato l'affidabilità dell'indice SSIM, consentendo un confronto coerente dei risultati in termini di qualità.

Per futuri sviluppi, potrebbe essere interessante esplorare ottimizzazioni specifiche per determinati dataset, tenendo conto della disposizione dei frame, poiché questo può avere un impatto cruciale sulle prestazioni complessive della compressione. Inoltre, valutare l'evoluzione di nuovi approcci o codec potrebbe offrire prestazioni superiori, considerando l'adattamento continuo delle tecniche di compressione alle immagini light field come un ambito di ricerca promettente.

RIFERIMENTI

- [1] PMF RESEARCH. *PMF RESEARCH COSA SONO GLI OLOGRAMMI? DEFINIZIONE E PROSPETTIVE.* 2022. URL: [https://pmf-research.eu/realta-aumentata-nuova-frontiera%20ologrammi/#:~:text=L%27ologrammi%3A%20storia%20ed%20evoluzione,-Gli%20ologrammi%20nascono&text=Dennis%20Gabor%2C%20famoso%20scienziato%20ungherese,\)%20e%20gramma%20\(messaggio\)..](https://pmf-research.eu/realta-aumentata-nuova-frontiera%20ologrammi/#:~:text=L%27ologrammi%3A%20storia%20ed%20evoluzione,-Gli%20ologrammi%20nascono&text=Dennis%20Gabor%2C%20famoso%20scienziato%20ungherese,)%20e%20gramma%20(messaggio)..)
- [2] Antonio Giammetta Raffaele Squillante. «Ligh Field Hologram Compression». In: (2023), p. 34.
- [3] wikipedia. *CirrusLogicAccuPak.* URL: https://wiki.multimedia.cx/index.php/Cirrus_Logic_AccuPak.
- [4] wikipedia. *FLV.* URL: https://en.wikipedia.org/wiki/Flash_Video.
- [5] wikipedia. *M-JPEG.* URL: https://it.wikipedia.org/wiki/Motion_JPEG.
- [6] wikipedia. *MPEG-4.* URL: <https://it.wikipedia.org/wiki/MPEG-4>.
- [7] wikipedia. *ProRes.* URL: https://en.wikipedia.org/wiki/Apple_ProRes.
- [8] Christoph Gerstbauer. *FFVHUFF.* URL: <https://ffmpeg-user.ffmpeg.narkive.com/kNF1DpI2/ffvhuff-codec-improvement>.
- [9] wikipedia. *zlib.* URL: https://wiki.multimedia.cx/index.php/Lossless_Codec_Libraries.
- [10] MagicYuv. *magicyuv.* URL: <https://www.magicyuv.com>.
- [11] Umme Sara, Morium Akter e Mohammad Shorif Uddin. «Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study». In: *Journal of Computer and Communications* 7.3 (2019), pp. 8–18.
- [12] Yang Cao et al. «Social-Aware Video Multicast Based on Device-to-Device Communications». In: *IEEE Transactions on Mobile Computing* 15 (giu. 2016), pp. 1528–1539. DOI: 10.1109/TMC.2015.2461214.
- [13] Hamid R. Sheikh Zhou Wang Alan C. Bovik e Eero P. Simoncelli. *The SSIM Index for Image Quality Assessment.* 2011. URL: <https://www.cns.nyu.edu/~lcv/ssim/#test>.
- [14] PMF RESEARCH. *PSNR-HVS-M.* 2009. URL: <https://www.ponomarenko.info/psnrhvsms.htm>.

- [15] FFmpeg Developers. *FFmpeg Documentation*. 2024.
- [16] MIT Media Lab Light Fields. *Messerschmitt*. URL: <https://web.media.mit.edu/~gordonw/SyntheticLightFields/>.
- [17] The Stanford 3D Scanning Repository. *3Dscanrep*. URL: <http://graphics.stanford.edu/data/3Dscanrep/>.
- [18] MIT Media Lab Light Fields. *Dice*. URL: <https://web.media.mit.edu/~gordonw/SyntheticLightFields/>.
- [19] MIT Media Lab Light Fields. *Fish*. URL: <https://web.media.mit.edu/~gordonw/SyntheticLightFields/>.
- [20] Max Planck Institut Informatik. *Car*. URL: <https://lightfields.mpi-inf.mpg.de/Dataset.html>.
- [21] Max Planck Institut Informatik. *Cobblestone*. URL: <https://lightfields.mpi-inf.mpg.de/Dataset.html>.
- [22] Max Planck Institut Informatik. *Mannequin*. URL: <https://lightfields.mpi-inf.mpg.de/Dataset.html>.
- [23] Max Planck Institut Informatik. *Blob*. URL: <https://lightfields.mpi-inf.mpg.de/Dataset.html>.

ELENCO DELLE FIGURE

2.1	Cattura di un ligh field	7
2.2	Informazioni racchiuse in un ligh field	8
4.1	Rapporto tra PSNR e qualità del video	15
7.1	Immagine estratta dal dataset "Messerschmitt"	24
7.2	Immagine estratta dal dataset "Dice"	24
7.3	Immagine estratta dal dataset "Fish"	24
7.4	Immagine estratta dal dataset "Car"	24
7.5	Immagine estratta dal dataset "Cobblestone"	24
7.6	Immagine estratta dal dataset "Mannequin"	24
7.7	Immagine estratta dal dataset "Blob"	24

ELENCO DELLE TABELLE

8.1	Studio codec 2022 sul dataset "ArtGallery" (89.26 MB)	26
8.2	Studio codec 2022 su "Dragons and bunnies" (11.80 MB)	26
8.3	Studio codec 2022 sul dataset "OpEx Room" (73.8 MB)	26
8.4	Studio codec 2022 sul dataset random "ArtGallery" (89.26 MB)	27
8.5	Studio codec 2022 sul dataset dataset random "Dragons and bunnies" (11.80 MB)	27
8.6	Studio codec 2022 sul dataset random "OpEx Room" (73.8 MB)	27
8.7	Studio codec 2022 sul dataset "Blob" (53.65 MB)	28
8.8	Studio codec 2022 sul dataset "Car" (63.50 MB)	28
8.9	Studio codec 2022 sul dataset "Car" (102.80 MB)	28
8.10	Studio codec 2022 sul dataset "Dice" (3.13 MB)	29
8.11	Studio codec 2022 sul dataset "Fish" (7.35 MB)	29
8.12	Studio codec 2022 sul dataset "Mannequin" (79.96 MB)	29
8.13	Studio codec 2022 sul dataset "Messerschmitt" (8.85 MB)	29
8.14	Studio codec 2022 sul dataset random "Blob" (53.65 MB)	30
8.15	Studio codec 2022 sul dataset random "Car" 63.50 MB	30
8.16	Studio codec 2022 sul dataset random "Cobblestone" (102.80 MB)	30
8.17	Studio codec 2022 sul dataset random "Dice" (3.13 MB)	31
8.18	Studio codec 2022 sul dataset random "Fish" (7.35 MB)	31
8.19	Studio codec 2022 sul dataset random "Mannequin" (79.96 MB)	31
8.20	Studio codec 2022 sul dataset random "Massershmitt" (8.85 MB)	31
8.21	Studio codec lossless 2023 sul dataset "ArtGallery" (89.26 MB)	32
8.22	Studio codec lossless 2023 sul dataset "Dragons and bunnies" (11.80 MB)	32
8.23	Studio codec lossless 2023 sul dataset "OpEx" (73.8 MB)	32
8.24	Studio codec lossy 2023 sul dataset "ArtGallery" (89.26 MB)	33
8.25	Studio codec lossy 2023 sul dataset "Dragons and bunnies" (11.80 MB)	33
8.26	Studio codec lossy 2023 sul dataset "OpEx" (73.8 MB)	33
8.27	Studio codec lossless 2023 sul dataset random "ArtGallery" (89.26 MB)	34
8.28	Studio codec lossless 2023 sul dataset random "Dragons and bunnies" (11.80 MB)	34
8.29	Studio codec lossless 2023 sul dataset random "OpEx" (73.8 MB)	34
8.30	Studio codec lossy 2023 sul dataset random "ArtGallery" (89.26 MB)	35

8.31 Studio codec lossy 2023 sul dataset dataset random "Dragons and bunnies" (11.80 MB)	35
8.32 Studio codec lossy 2023 sul dataset random "OpEx" (73.8 MB)	35
8.33 Studio codec lossless 2023 sul dataset "Blob" (53.65 MB)	36
8.34 Studio codec lossless 2023 sul dataset "Car" (63.50 MB)	36
8.35 Studio codec lossless 2023 sul dataset "Cobblestone" (102.80 MB)	36
8.36 Studio codec lossless 2023 sul dataset "Dice" (3.13 MB)"	37
8.37 Studio codec lossless 2023 sul dataset "Fish" (7.35 MB)	37
8.38 Studio codec lossless 2023 sul dataset "Mannequin" (79.96 MB)"	37
8.39 Studio codec lossless 2023 sul dataset "Messerschmitt" (8.85 MB)	37
8.40 Studio codec lossy 2023 sul dataset "Blob" (53.65 MB)	38
8.41 Studio codec lossy 2023 sul dataset "Car" (63.50 MB)	38
8.42 Studio codec lossy 2023 sul dataset "Cobblestone" (102.80 MB)	38
8.43 Studio codec lossy 2023 sul dataset "Dice" (3.13 MB)	39
8.44 Studio codec lossy 2023 sul dataset "Fish" (7.35 MB)	39
8.45 Studio codec lossy 2023 sul dataset "Mannequin" (79.96 MB)	39
8.46 Studio codec lossy 2023 sul dataset "Messerschmitt" (8.85 MB)	39
8.47 Studio codec lossless 2023 sul dataset random "Blob" (53.65 MB)	40
8.48 Studio codec lossless 2023 sul dataset random "Car" (63.50 MB)	40
8.49 Studio codec lossless 2023 sul dataset random "Cobblestone" (102.80 MB)	40
8.50 Studio codec lossless 2023 sul dataset random "Dice" (3.13 MB)	41
8.51 Studio codec lossless 2023 sul dataset random "Fish" (7.35 MB)	41
8.52 Studio codec lossless 2023 sul dataset random "Mannequin" (79.96 MB)	41
8.53 Studio codec lossless 2023 sul dataset random "Messerschmitt" (8.85 MB)	41
8.54 Studio codec lossy 2023 sul dataset random "Blob" (53.65 MB)	42
8.55 Studio codec lossy 2023 sul dataset random "Car" (63.50 MB)	42
8.56 Studio codec lossy 2023 sul dataset random "Cobblestone" (102.80 MB)	42
8.57 Studio codec lossy 2023 sul dataset random "Dice" (3.13 MB)	43
8.58 Studio codec lossy 2023 sul dataset random "Fish" (7.35 MB)	43
8.59 Studio codec lossy 2023 sul dataset random "Mannequin" (79.96 MB)	43
8.60 Studio codec lossy 2023 sul dataset random "Messerschmitt" (8.85 MB)	43
8.61 Studio decompressione codec 2023 sul dataset "Art Gallery" (89.26 MB)	44
8.62 Studio decompressione codec 2023 sul dataset "Dragons and bunnies" (11.80 MB)	44
8.63 Studio decompressione codec 2023 sul dataset "OpEx" (73.8 MB)	44
8.64 Studio decompressione codec 2023 sul dataset random "Art Gallery" (89.26 MB)	45
8.65 Studio decompressione codec 2023 dataset random "Dragons & bunnies" (11.80 MB)	45
8.66 Studio decompressione codec 2023 sul dataset random "OpEx" (73.8 MB)	45
8.67 Studio decompressione codec 2023 sul dataset "Blob" (53.65 MB)	46
8.68 Studio decompressione codec 2023 sul dataset "Car" (53.65 MB)	46
8.69 Studio decompressione codec 2023 sul dataset "Cobblestone" (102.80 MB)	46
8.70 Studio decompressione codec 2023 sul dataset "Dice" (3.13 MB)	46
8.71 Studio decompressione codec 2023 sul dataset "Fish" (7.35 MB)	47
8.72 Studio decompressione codec 2023 sul dataset "Mannequin" (79.96 MB)	47
8.73 Studio decompressione codec 2023 sul dataset "Messerschmitt" (8.85 MB)	47

8.74 Studio decompressione codec 2023 sul dataset random "Blob" (53.65 MB)	48
8.75 Studio decompressione codec 2023 sul dataset random "Car" (63.50 MB)	48
8.76 Studio decompressione codec 2023 dataset random "Cobblestone" (102.80 MB)	48
8.77 Studio decompressione codec 2023 sul dataset random "Dice" (3.13 MB)	49
8.78 Studio decompressione codec 2023 sul dataset random "Fish" (7.35 MB)	49
8.79 Studio decompressione codec 2023 sul dataset random "Mannequin" (79.96 MB)	49
8.80 Studio decompressione codec 2023 dataset random "Messerschmitt" (8.85 MB)	49
8.81 Codec lossy 2023 sul dataset "Cobblestone" (102.80 MB)	50
8.82 Codec lossy 2023 sul dataset "Cobblestone" (102.80 MB)	50
8.83 Codec lossy 2023 sul dataset "Cobblestone" (102.80 MB)	51
8.84 Codec lossy 2023 sul dataset "Cobblestone" random (102.80 MB)	51
8.85 Codec lossy 2023 sul dataset "Cobblestone" random (102.80 MB)	51
8.86 Codec lossy 2023 sul dataset "Cobblestone" random (102.80 MB)	51
8.87 Codec lossy 2023 sul dataset "Messerschmitt" (8.85 MB)	52
8.88 Codec lossy 2023 sul dataset "Messerschmitt" (8.85 MB)	52
8.89 Codec lossy 2023 sul dataset "Messerschmitt" (8.85 MB)	52
8.90 Codec lossy 2023 sul dataset "Messerschmitt" random (8.85 MB)	53
8.91 Codec lossy 2023 sul dataset "Messerschmitt" random (8.85 MB)	53
8.92 Codec lossy 2023 sul dataset "Messerschmitt" random (8.85 MB)	53
8.93 Codec lossy 2023 sul dataset "Dragons and bunnies" (11.80 MB)	54
8.94 Codec lossy 2023 sul dataset "Dragons and bunnies" (11.80 MB)	54
8.95 Codec lossy 2023 sul dataset "Dragons and bunnies" (11.80 MB)	54
8.96 Codec lossy 2023 sul dataset random "Dragons and bunnies" (11.80 MB)	55
8.97 Codec lossy 2023 sul dataset random "Dragons and bunnies" (11.80 MB)	55
8.98 Codec lossy 2023 sul dataset random "Dragons and bunnies" (11.80 MB)	55
8.99 Codec lossy 2023 sul dataset "Mannequin" (79.96 MB)	56
8.100 Codec lossy 2023 sul dataset "Mannequin" (79.96 MB)	56
8.101 Codec lossy 2023 sul dataset "Mannequin" (79.96 MB)	56
8.102 Codec lossy 2023 sul dataset random "Mannequin" (79.96 MB)	57
8.103 Codec lossy 2023 sul dataset random "Mannequin" (79.96 MB)	57
8.104 Codec lossy 2023 sul dataset random "Mannequin" (79.96 MB)	57
8.105 Codec lossy 2023 sul dataset "Art Gallery" (89.26 MB)	58
8.106 Codec lossy 2023 sul dataset "Art Gallery" (89.26 MB)	58
8.107 Codec lossy 2023 sul dataset random "Art Gallery" (89.26 MB)	58
8.108 Codec lossy 2023 sul dataset random "Art Gallery" (89.26 MB)	58
8.109 Codec lossy 2023 sul dataset "Blob" (53.65 MB)	59
8.110 Codec lossy 2023 sul dataset dataset "Blob" (53.65 MB)	59
8.111 Codec lossy 2023 sul dataset random "Blob" (53.65 MB)	59
8.112 Codec lossy 2023 sul dataset random "Blob" (53.65 MB)	59
8.113 Codec lossy 2023 sul dataset "Car" (63.50 MB)	60
8.114 Codec lossy 2023 sul dataset "Car" (63.50 MB)	60
8.115 Codec lossy 2023 sul dataset random "Car" (63.50 MB)	60
8.116 Codec lossy 2023 sul dataset random "Car" (63.50 MB)	60
8.117 Codec lossy 2023 sul dataset "Dice" (3.13 MB)	61
8.118 Codec lossy 2023 sul dataset "Dice" (3.13 MB)	61

ELENCO DELLE TABELLE

8.119Codec lossy 2023 sul dataset random "Dice" (3.13 MB)	61
8.120Codec lossy 2023 sul dataset random "Dice" (3.13 MB)	61
8.121Codec lossy 2023 sul dataset "Fish" (7.35 MB)	62
8.122Codec lossy 2023 sul dataset "Fish" (7.35 MB)	62
8.123Codec lossy 2023 sul dataset random "Fish" (7.35 MB)	62
8.124Codec lossy 2023 sul dataset random "Fish" (7.35 MB)	62