



Università degli Studi di Salerno

Dipartimento di Informatica

Progetto Esame
Compressione Dati

Ligh-Field Image Compression

Docente

Prof. Bruno Carpentieri

Studenti

Francesco Fasolino 0522501693

Lorenzo Sorrentino 0522501849

Alessio Maddaluno 0522501729

Anno Accademico 2025-2026

Indice

1	Introduzione	1
2	Stato dell'Arte	2
2.1	Introduzione alle Light Field Images	2
2.1.1	Come vengono acquisite	3
2.1.2	Come vengono rappresentate	4
2.1.3	Perché sono utili	4
2.2	Codec Video per compressione lossy	4
2.2.1	HEVC	4
2.2.2	AV1	5
2.2.3	VP9	5
2.3	Eripolar Plane Images (EPI)	5
2.3.1	EPI e compressione basata su codec video	6
2.4	JPEG Pleno	7
2.4.1	JPEG Pleno Parte 2: Light Field Coding	8
3	Metodologia	12
3.1	Metriche per la valutazione della fidelity	12
3.1.1	PSNR	13
3.1.2	SSIM	13
3.2	Implementazione Codec Video	14
3.2.1	Organizzazione dell'input	14
3.2.2	Padding on-the-fly	15
3.2.3	Parametri dei codec	15
3.2.4	Decodifica e Ricostruzione	15
3.3	Implementazione Eripolar Plane Images (EPI)	16
3.3.1	Caricamento del Light Field	16
3.3.2	Dal Light Field al volume EPI (tiled)	17
3.3.3	Compressione con codec video	17

3.3.4	Decodifica e Ricostruzione del Volume EPI	17
3.4	Implementazione JPEG Pleno	19
3.4.1	Workflow di elaborazione	19
3.4.2	Preprocessing e Conversione (Step 1-2)	19
3.4.3	Encoding (Step 3)	20
3.4.4	Decoding e Post-processing (Step 4-6)	20
4	Risultati	21
4.1	Analisi dei Risultati: Dataset <i>Bikes</i>	22
4.1.1	Qualità Visiva	22
4.1.2	Efficienza di Compressione	23
4.2	Analisi dei Risultati: Dataset <i>Danger De Mort</i>	25
4.2.1	Qualità Visiva	25
4.2.2	Efficienza di Compressione	26
4.3	Analisi dei Risultati: Dataset <i>Greek</i>	28
4.3.1	Qualità Visiva	28
4.3.2	Efficienza di Compressione	28
4.4	Analisi dei Risultati: Dataset <i>Sideboard</i>	31
4.4.1	Qualità Visiva	31
4.4.2	Efficienza di Compressione	31
4.5	Analisi dei Risultati: Dataset <i>Tarot</i>	34
4.5.1	Qualità Visiva	34
4.5.2	Efficienza di Compressione	34
5	Conclusioni	38
5.1	Risultati Ottenuti	38
5.2	Sviluppi Futuri	39
Bibliografia		41

Capitolo 1

Introduzione

Negli ultimi anni le tecnologie di acquisizione e rappresentazione delle immagini hanno compiuto passi significativi, andando oltre la semplice registrazione bidimensionale della realtà. Tra queste innovazioni, le *light field images* occupano un ruolo di crescente interesse grazie alla loro capacità di catturare non solo l'aspetto visibile di una scena, ma anche la direzione dei raggi di luce che la compongono. Questo permette di ottenere una descrizione più ricca e flessibile dell'informazione visiva, aprendo la strada a nuove possibilità nel rendering, nella realtà virtuale, nella ricostruzione tridimensionale e nella fotografia computazionale.

Tuttavia, la maggiore quantità di dati generata da questo tipo di acquisizione rende fondamentale lo sviluppo di tecniche di compressione efficienti, in grado di ridurre lo spazio necessario senza compromettere la qualità percepita. In questo contesto, la valutazione della *fidelity* delle immagini compresse assume un ruolo centrale: è necessario disporre di metriche affidabili che permettano di misurare quanto la ricostruzione sia fedele all'originale.

Questa relazione si propone di introdurre il concetto di light field, descrivere le principali modalità di acquisizione, presentare e valutare le tecniche di compressione lossy più diffuse, con particolare attenzione alle metriche utilizzate per valutarne la qualità. L'obiettivo è fornire una panoramica chiara e accessibile, utile per comprendere le potenzialità e le sfide legate a questo tipo di rappresentazione visiva.

Capitolo 2

Stato dell'Arte

Il presente capitolo fornisce una panoramica completa del contesto tecnologico e teorico in cui si inserisce il progetto. La trattazione inizia con la definizione formale delle immagini Light Field, analizzandone le modalità di acquisizione e le proprietà uniche rispetto alla fotografia tradizionale.

Successivamente, l'attenzione si sposta sulle strategie di compressione attualmente adottate in letteratura e nell'industria. Vengono esaminati i codec video standard (HEVC, AV1, VP9) come soluzioni mature per la codifica delle singole viste e introdotto il concetto di *Epipolar Plane Images* (EPI) come metodo per sfruttare la coerenza geometrica della scena. Infine, viene descritto il framework JPEG Pleno, lo standard internazionale di riferimento progettato specificamente per superare i limiti degli approcci tradizionali nella gestione dei dati plenottici.

2.1 Introduzione alle Light Field Images

Le *light field images* sono un tipo particolare di immagini che non catturano solo ciò che si vede da un singolo punto di vista, ma registrano anche da quale direzione arriva la luce. In pratica, invece di avere una sola immagine piatta, si ottiene un insieme di immagini che rappresentano la scena da angolazioni leggermente diverse.

Questa informazione aggiuntiva permette di fare operazioni che con una normale fotografia non sarebbero possibili, come:

- cambiare il punto di vista dopo lo scatto,

- mettere a fuoco oggetti diversi in un secondo momento,
- stimare meglio la profondità della scena.

2.1.1 Come vengono acquisite

Esistono diversi modi per acquisire light field images, tutti basati sull'idea di catturare più punti di vista della stessa scena:

- **Telecamere con microlenti:** sono fotocamere speciali che hanno una serie di piccole lenti davanti al sensore. Ogni microlente cattura una parte leggermente diversa della scena, permettendo di registrare sia la posizione sia la direzione della luce.
- **Array di telecamere:** si utilizzano molte telecamere disposte in una griglia. Ognuna scatta la scena da una posizione diversa, e tutte insieme formano il light field.
- **Una telecamera in movimento:** si può anche muovere una singola telecamera lungo una traiettoria (ad esempio su uno slider) e scattare molte immagini. Se la scena è statica, queste immagini possono essere combinate per ottenere un light field.

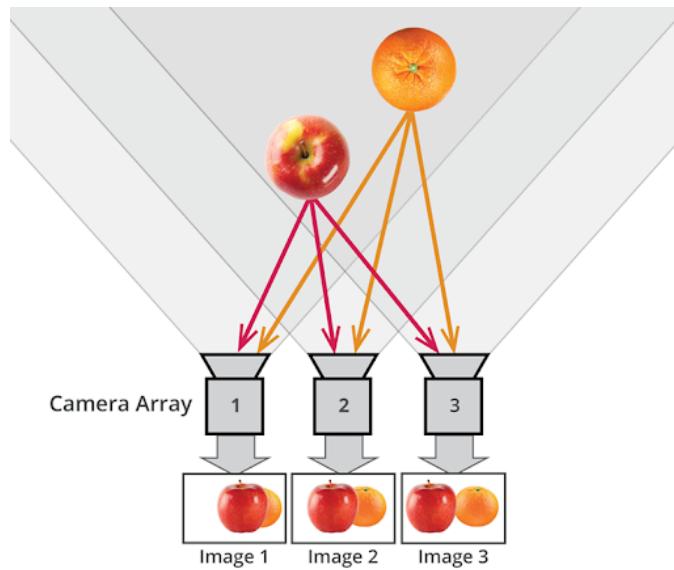


Figura 2.1: Esempio di array di telecamere

2.1.2 Come vengono rappresentate

Una volta acquisite, le light field images possono essere viste in modi diversi:

- come una collezione di immagini 2D, ognuna con un punto di vista leggermente diverso;
- come strutture che mettono in evidenza come gli oggetti si spostano tra una vista e l'altra, utili per analizzare la profondità;
- come dati che permettono di ricostruire nuove immagini con effetti particolari, come il refocusing.

2.1.3 Perché sono utili

Le light field images sono interessanti perché permettono di fare cose che una normale fotografia non può fare. Sono utilizzate in:

- realtà virtuale e aumentata,
- cinematografia e effetti speciali,
- ricostruzione 3D,
- applicazioni scientifiche e industriali dove serve conoscere la profondità.

2.2 Codec Video per compressione lossy

Per la compressione delle immagini Light Field, è possibile sfruttare codec video standard che implementano tecniche avanzate di predizione e trasformazione. I tre codec considerati sono HEVC, AV1 e VP9, tutti in grado di gestire profondità di colore a 10 bit.

2.2.1 HEVC

HEVC, noto anche come H.265, è uno standard di compressione video progettato per migliorare l'efficienza rispetto al predecessore H.264. Le principali caratteristiche:

- **Predizione intra/inter-frame:** sfrutta la correlazione spaziale e temporale tra blocchi.
- **CTU (Coding Tree Units):** blocchi fino a 64×64 pixel, con suddivisione adattiva.

- **Supporto 10-bit:** ideale per contenuti ad alta gamma dinamica.

HEVC offre un eccellente rapporto qualità/bitrate, ma richiede più potenza di calcolo in codifica.

2.2.2 AV1

AV1 è un codec open-source sviluppato dall’Alliance for Open Media, pensato per streaming su larga scala e alta efficienza:

- **Compressione superiore a HEVC:** riduzione del bitrate a parità di qualità.
- **Struttura flessibile dei blocchi:** fino a 128×128 pixel.
- **Ottimizzato per web e HDR:** supporta 10-bit e formati YUV.

Il principale svantaggio è la complessità computazionale: la codifica AV1 è più lenta rispetto a HEVC.

2.2.3 VP9

VP9 è il codec open-source di Google, successore di VP8, ampiamente usato su piattaforme web:

- **Efficienza simile a HEVC:** bitrate ridotto rispetto a H.264.
- **Compatibilità web:** integrato in browser e servizi di streaming.
- **Supporto 10-bit:** utile per contenuti Light Field.

VP9 è meno complesso di AV1, ma generalmente meno efficiente in compressione rispetto ad AV1.

2.3 Epipolar Plane Images (EPI)

Il *Light Field* è una rappresentazione multidimensionale della radianza, comunemente parametrizzata come $L(u, v, x, y)$, dove (u, v) identificano la posizione della vista nella griglia angolare e (x, y) le coordinate spaziali del pixel. Questa struttura consente di catturare informazioni sulla geometria della scena e sulla direzione dei raggi luminosi.

Un **Epipolar Plane Image (EPI)** è una sezione bidimensionale del Light

Field ottenuta fissando una delle coordinate spaziali e variando le coordinate angolari. Ad esempio, fissando $y = y_0$ e $v = v_0$, si ottiene:

$$\text{EPI}(u, x) = L(u, v_0, x, y_0)$$

In questa rappresentazione, ogni punto 3D della scena genera una linea nell'EPI, la cui inclinazione è proporzionale alla profondità del punto stesso. Questa proprietà è fondamentale per due motivi:

- **Analisi della profondità:** la disparità tra viste è codificata nella pendenza delle linee epipolari.
- **Compressione:** gli EPI presentano forti correlazioni lungo le linee, riducendo la ridondanza direzionale.

Un esempio di EPI può essere osservato nella Figura 2.2, nella Sezione 2.4.1.

2.3.1 EPI e compressione basata su codec video

Gli EPI possono essere organizzati in sequenze di immagini che mostrano variazioni regolari tra viste. Questa struttura è analoga a quella di un flusso video, dove i frame consecutivi presentano alta correlazione temporale. Per questo motivo, è possibile sfruttare codec video standard (HEVC, AV1, VP9) per la compressione degli EPI:

- **HEVC (H.265):** ottimizzato per alta efficienza, supporta profondità a 10-bit e predizione inter-frame.
- **AV1:** codec open-source con ottime prestazioni su contenuti ad alta risoluzione.
- **VP9:** alternativa open-source con buona compatibilità web.

L'idea è trasformare il Light Field in un volume di EPI, suddiviso in *tiles* e trattato come sequenza video. In questo modo:

1. Si sfruttano le tecniche di predizione spaziale e temporale dei codec.
2. Si ottiene una compressione lossy controllata tramite parametri come il *CRF* (Constant Rate Factor).
3. Si mantiene compatibilità con strumenti consolidati (FFmpeg, librerie standard).

2.4 JPEG Pleno

Con l'avvento di nuove modalità per le immagini, JPEG ha ritenuto necessario la ricerca e lo sviluppo di un nuovo standard per la compressione di queste nuove modalità. Per questo motivo è nato JPEG Pleno [3] [1], un framework standardizzato per rappresentare e codificare texture-plus-depth, light field, point cloud e immagini olografiche. Tali modalità devono essere intese come rappresentazioni della luce ispirate alla funzione plenottica, indipendentemente dal modello che ha catturato o creato tutto o parte del contenuto. In questa relazione ci concentreremo sulla sezione che riguarda le Light Field.

Le diverse parti dello standard JPEG Pleno sono progettate in modo coordinato considerando sinergie e dipendenze, in modo tale che il risultato complessivo sia migliorativo rispetto alla somma delle singoli parti. JPEG Pleno, infatti, è concepito non come un insieme di singoli strumenti di codifica, ma un framework di rappresentazione completamente integrato per fornire supporto avanzato per la manipolazione dei dati plenottici, i metadati, l'accesso casuale, l'interazione e vari formati di file.

Lo standard JPEG Pleno (ISO/IEC 21794) definisce un ecosistema completo per la rappresentazione, la compressione e lo scambio di dati plenottici. Attualmente, lo standard è composto da sei parti:

- **Parte 1: Framework (ISO/IEC 21794-1).** Definisce l'architettura di sistema e il formato file contenitore (.jp1). Utilizza una struttura a "box" (superbox e sottobox) per incapsulare simultaneamente diverse modalità di imaging e metadati eterogenei, garantendo l'interoperabilità e l'estensibilità futura [1].
- **Parte 2: Light Field Coding (ISO/IEC 21794-2).** Specifica gli algoritmi di compressione per i dati Light Field, definendo due modalità principali: il *4D Transform Mode* (4DTM) per light field densi e il *4D Prediction Mode* (4DPM) per light field sparsi, oltre ai relativi metadati geometrici [1] [5].
- **Parte 3: Conformance Testing (ISO/IEC 21794-3).** Stabilisce le procedure di test e fornisce bitstream di riferimento per verificare la conformità delle implementazioni encoder/decoder rispetto alle specifiche normative delle altre parti [1].

- **Parte 4: Reference Software (ISO/IEC 21794-4).** Fornisce il software di riferimento ufficiale, il *JPEG Pleno Model* (JPLM), che implementa le funzionalità del framework (Parte 1) e degli strumenti di codifica (Parte 2), fungendo da base per lo sviluppo e la verifica sperimentale [1] [4].
- **Parte 5: Holography (ISO/IEC 21794-5).** Definisce il formato del codestream per la memorizzazione e la trasmissione di modalità olografiche e dei relativi metadati specifici. Questa parte supporta diverse rappresentazioni del segnale olografico (come ologrammi di ampiezza, di fase o complessi) gestendo sia ologrammi acquisiti otticamente che generati al computer (CGH) [2].
- **Parte 6: Point Cloud Coding (ISO/IEC 21794-6).** Specifica una tecnologia di codifica basata su apprendimento automatico (*learning-based*) per le nuvole di punti 3D. L'obiettivo di questa parte è fornire algoritmi efficienti per la compressione di geometria e attributi, ottimizzati non solo per la visualizzazione umana ma anche per compiti di visione artificiale e analisi automatica [6].

2.4.1 JPEG Pleno Parte 2: Light Field Coding

Di seguito si approfondisce la Parte 2 dello standard (ISO/IEC 21794-2) [5], di nostro interesse ai fini del progetto, la quale è specificamente dedicata alla codifica dei dati Light Field. L'obiettivo di questa parte è fornire algoritmi di compressione efficienti che tengano conto della struttura quadridimensionale dei dati (u, v, x, y) , superando i limiti dei codec video tradizionali adattati.

Lo standard riconosce che non esiste un unico approccio ottimale per tutte le tipologie di Light Field. In particolare, la densità di campionamento angolare influisce drasticamente sulle performance di codifica. Per questo motivo, JPEG Pleno definisce due modalità di codifica indipendenti:

- **4D Prediction Mode (4D-PM):** più adatto per Light Field sparsi (es. array di telecamere con ampia baseline).
- **4D Transform Mode (4D-TM):** più adatto per Light Field densi (es. immagini plenottiche lenslet), dove la correlazione tra viste adiacenti è molto alta.

4D Prediction Mode (4D-PM)

Il *4D Prediction Mode* basa la sua efficienza sulla sintesi di nuove viste a partire da un sottoinsieme di viste di riferimento. L'approccio sfrutta algoritmi di warping basati sulla profondità (depth-based warping).

Il processo segue un ordine gerarchico:

1. Le viste vengono suddivise in "viste di riferimento" e "viste intermedie".
2. Le viste di riferimento, che contengono l'informazione base di texture e profondità, vengono codificate utilizzando codec 2D standard già esistenti (come JPEG 2000).
3. Le viste intermedie vengono predette sintetizzandole a partire dalle viste di riferimento decodificate. Eventuali errori di predizione (residui) vengono poi codificati e trasmessi per raffinare la qualità finale.

Questa modalità è particolarmente efficace quando la distanza tra le viste è elevata, poiché la geometria della scena (profondità) è il preditore più robusto per collegare prospettive diverse.

La modalità 4D-PM non risulta attualmente implementata nel software di riferimento JPLM [4], quindi non è stato possibile testarla nella fase sperimentale.

4D Transform Mode (4D-TM)

Il *4D Transform Mode* è l'approccio "nativo" per i Light Field e rappresenta la modalità utilizzata nella parte sperimentale di questo lavoro. A differenza del 4D-PM, questo metodo tratta il Light Field come un unico volume di dati 4D, sfruttando congiuntamente la ridondanza spaziale (intra-view) e quella angolare (inter-view).

Il funzionamento ricalca l'architettura classica a blocchi del JPEG, ma estesa a quattro dimensioni:

1. **Partizionamento:** Il volume 4D viene suddiviso in super-blocchi 4D di dimensione variabile.
2. **Trasformata 4D-DCT:** Viene applicata una Trasformata del Coseno Discreta (DCT) separabile su tutte e quattro le dimensioni. Questo permette di compattare l'energia non solo spazialmente, ma anche attraverso le viste vicine.

3. **Quantizzazione e Codifica:** I coefficienti trasformati vengono quantizzati. Successivamente, i coefficienti non nulli vengono raggruppati utilizzando una struttura dati chiamata *4D Hexadecatree* e codificati tramite un codificatore aritmetico binario adattivo.

Il 4D-TM eccelle nei dataset di tipo lenslet (micro-lenti), dove le viste sono estremamente vicine e la ridondanza è massima.

Sviluppi futuri: Slanted 4D Transform Mode

Recentemente è stata proposta una seconda edizione dello standard che introduce il *Slanted 4D Transform Mode*. Le motivazioni che hanno spinto a ricercare una modifica per la 4D-TM, sono la possibilità di fallimento da parte della DCT quando la baseline aumenta e gli oggetti hanno uno spostamento notevole tra una vista e l'altra, creando un light field poco "denso".

Mentre il 4D-TM classico applica la trasformata direttamente sui blocchi 4D, la nuova modalità Slanted 4D-TM introduce un passaggio preliminare di allineamento geometrico. Questo approccio riconosce che il volume 4D è intrinsecamente composto da Epipolar Plane Images (EPI), descritte nella Sezione 2.3, dove la disparità appare come linee inclinate. La S4D-TM applica quindi uno scorrimento ('slanting') adattivo per verticalizzare queste strutture, rendendo il segnale molto più facile da comprimere per la successiva 4D-DCT.

Questo processo viene suddiviso in tre parti:

1. **Stima della pendenza:** Il codec analizza il blocco 4D per individuare la pendenza dominante (che corrisponde alla disparità/profondità principale degli oggetti in quel blocco).
2. **Allineamento (Slanting):** Il blocco viene "inclinato" o deformato geometricamente in modo che le linee diagonali dell'EPI diventino orizzontali o verticali.
3. **Compattazione energetica:** Una volta che i dati sono allineati, la 4D-DCT torna a essere estremamente efficiente, perché vede segnali molto più regolari e facili da comprimere.

I vantaggi di questo approccio sono l'indipendenza dalla baseline, surclassando entrambe le 4DTM e 4DPM in quasi ogni scenario; l'indipendenza dalla

presenza di una mappa di profondità esplicita, deducendo l'inclinazione geometrica direttamente dai dati, alleggerendo e irrobustendo il decoder; e infine aumenta l'efficienza riducendo il bitrate a parità di qualità visiva rispetto la prima edizione, soprattutto per contenuti VR/AR con viste distanziate.

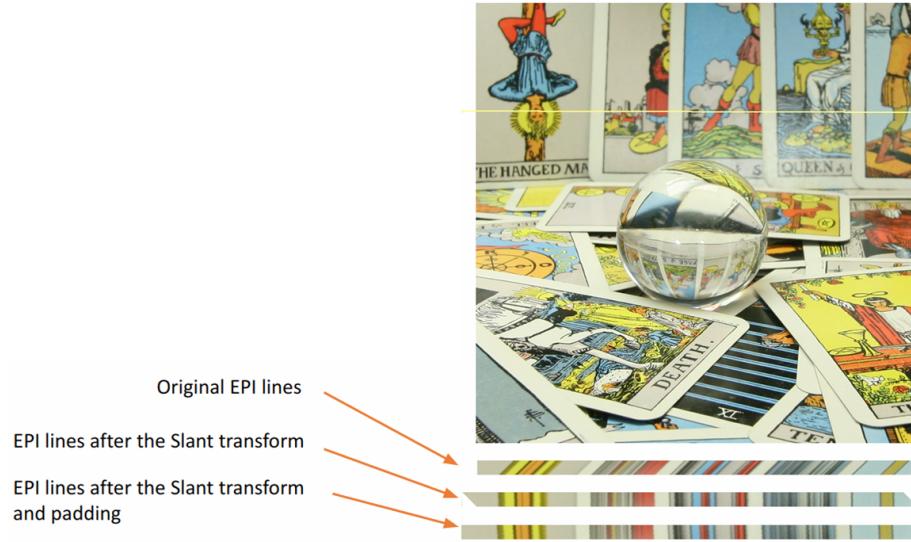


Figura 2.2: Steps della trasformazione geometrica (slanting) [5]

Capitolo 3

Metodologia

In questo capitolo viene descritta la metodologia operativa adottata per l’implementazione e la valutazione delle diverse strategie di compressione Light Field.

La trattazione si apre con la definizione delle metriche di qualità oggettiva e percettiva (PSNR e SSIM) utilizzate per validare la fedeltà delle ricostruzioni. Successivamente, vengono dettagliate le tre pipeline di elaborazione software sviluppate per il progetto: l’adattamento dei codec video standard (HEVC, AV1, VP9) per la codifica sequenziale delle viste, l’approccio basato sulla trasformazione in Epipolar Plane Images (EPI) per sfruttare la coerenza volumetrica, e infine l’utilizzo del framework di riferimento standardizzato JPEG Pleno. Per ciascun approccio verranno analizzati il flusso di lavoro, le scelte implementative e i parametri di configurazione adottati.

3.1 Metriche per la valutazione della fidelity

La valutazione della *fidelity* nelle immagini light field compresse è fondamentale per quantificare quanto la ricostruzione sia fedele ai dati originali. Tra le metriche più utilizzate in letteratura vi sono il *Peak Signal-to-Noise Ratio* (PSNR) e lo *Structural Similarity Index* (SSIM). Queste metriche permettono di analizzare la qualità sia dal punto di vista numerico sia da quello percettivo, fornendo una misura oggettiva dell’impatto della compressione.

3.1.1 PSNR

Il *Peak Signal-to-Noise Ratio* (PSNR) è una metrica classica per valutare la qualità di immagini compresse. Si basa sulla differenza punto per punto tra l'immagine originale e quella ricostruita, misurata tramite l'errore quadratico medio (MSE). Il PSNR è definito come:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right)$$

dove:

- MAX_I è il valore massimo possibile per un pixel (ad esempio 255 per immagini a 8 bit),
- MSE è l'errore quadratico medio tra immagine originale I e ricostruita \hat{I} :

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (I_i - \hat{I}_i)^2$$

Un valore di PSNR più alto indica una ricostruzione più fedele. Tuttavia, il PSNR non tiene conto della percezione visiva umana: due immagini con lo stesso PSNR possono apparire qualitativamente molto diverse. Nel contesto dei light field, il PSNR può essere calcolato sia sulle singole viste sia sull'intero set di viste, ma non cattura direttamente la coerenza angolare tra le immagini.

3.1.2 SSIM

Lo *Structural Similarity Index* (SSIM) è una metrica percettiva progettata per modellare la sensibilità del sistema visivo umano. A differenza del PSNR, l'SSIM non confronta semplicemente i valori dei pixel, ma valuta la similarità strutturale tra due immagini considerando tre componenti fondamentali:

1. **Luminanza** $l(x, y)$
2. **Contrasto** $c(x, y)$
3. **Struttura** $s(x, y)$

La formula generale dell'SSIM tra due patch x e y è:

$$\text{SSIM}(x, y) = [l(x, y)]^\alpha [c(x, y)]^\beta [s(x, y)]^\gamma$$

dove tipicamente $\alpha = \beta = \gamma = 1$. Le tre componenti sono definite come:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

dove:

- μ_x, μ_y sono le medie locali,
- σ_x, σ_y sono le deviazioni standard locali,
- σ_{xy} è la covarianza tra le due patch,
- C_1, C_2, C_3 sono costanti di stabilizzazione.

L'SSIM assume valori tra 0 e 1, dove 1 indica identità perfetta. Nel caso dei light field, l'SSIM è particolarmente utile perché è più sensibile alla preservazione delle strutture e delle texture, che sono fondamentali per mantenere la coerenza visiva tra le viste angolari.

3.2 Implementazione Codec Video

Questa sezione descrive come lo script Python `codec_video.py` gestisce la compressione delle viste Light Field usando FFmpeg: lettura dei PPM in sequenza, padding on-the-fly e configurazione dei codec e ricostruzione delle immagini.

3.2.1 Organizzazione dell'input

Le viste originali sono file PPM (P6) a 16 bit. Lo script:

- legge le *dimensioni* dal primo PPM;
- usa FFmpeg in modalità *pattern glob* per costruire una sequenza: `-pattern_type glob -i "<folder>/*.ppm"`, con un framerate definito (es. `-framerate 120`) per comporre il video.

3.2.2 Padding on-the-fly

Gli encoder sono ottimizzati per dimensioni multiple di 8, per *HEVC* è obbligatorio. Per questo motivo, lo script calcola una dimensione target (multipli di 8) e applica il padding *on-the-fly*, per tutti i codec, con il filtro:

```
-vf "pad=WIDTH:HEIGHT:0:0:black"
```

Il padding è *top-left anchored* ($x=0, y=0$) così che il crop successivo sia deterministico. Le informazioni di padding (dimensioni originali vs target) sono salvate per la fase di ricostruzione.

3.2.3 Parametri dei codec

La compressione avviene con:

- **HEVC** (`libx265`) in `.mp4`;
- **AV1** (`libaom-av1`) in `.mkv` con `-cpu-used 5` per velocizzare;
- **VP9** (`libvpx-vp9`) in `.webm` con `-b:v 0` per CRF puro e `-cpu-used 3` per velocizzare.

Il parametro `-cpu-used`, utilizzato con i codec *AV1* e *VP9*, è stato opportunamente tarato per ottenere un considerevole miglioramento dei tempi di compressione, senza impattare significativamente le dimensioni dei file ottenuti. Per tutti, lo script imposta `-crf <valore>` e `-pix_fmt yuv420p10le` per ottenere una codifica a **10 bit** con campionamento cromatico 4:2:0 (bilanciando fedeltà e compatibilità). Opzionalmente si può usare `yuv444p10le` per massima fedeltà cromatica (a costo di bitrate superiore).

3.2.4 Decodifica e Ricostruzione

La fase di decodifica e ricostruzione delle viste originali richiede una gestione attenta per invertire le trasformazioni applicate in fase di input, in particolare il padding e le conversioni di spazio colore. Lo script implementa una pipeline di decompressione tramite FFmpeg progettata per risolvere le problematiche legate al sottocampionamento cromatico su dimensioni non standard (dispari). La catena di elaborazione per ogni frame video segue questi passaggi:

1. **Conversione RGB anticipata:** Viene applicato immediatamente il filtro `lutrgb` per convertire lo spazio colore da YUV 4:2:0 a RGB a 48 bit (16 bit per canale). Questo passaggio è fondamentale: convertire in

RGB *prima* di effettuare il ritaglio (crop) rende ogni pixel indipendente, svincolando la geometria dai blocchi di sottocampionamento 2×2 tipici del formato YUV. Ciò permette di gestire correttamente anche dimensioni dispari che altrimenti verrebbero arrotondate erroneamente da FFmpeg.

2. **Rimozione del Padding (Crop):** Una volta nello spazio RGB, viene applicato il filtro `crop` utilizzando le dimensioni originali salvate durante la fase di compressione. Il ritaglio rimuove le bande nere aggiunte precedentemente, ripristinando la risoluzione esatta dell'immagine sorgente (es. 625×434).
3. **Esportazione:** I frame vengono esportati in formato `rgb48be` (Big Endian) per generare file PPM grezzi.
4. **Header Patching:** FFmpeg, esportando in `rgb48be`, imposta di default il valore massimo nell'header PPM a $65535 (2^{16} - 1)$. Tuttavia, dato che la compressione è avvenuta a 10 bit, lo script esegue una correzione binaria post-processo su ogni file decompresso, modificando il valore massimo dell'header da 65535 a $1023 (2^{10} - 1)$. Questo garantisce che la profondità di colore effettiva sia interpretata correttamente durante il calcolo delle metriche.

3.3 Implementazione Epipolar Plane Images (EPI)

Questa sezione descrive come lo script Python `epi_codec_video.py` gestisce la compressione e decompressione delle viste Light Field usando FFmpeg, preprocessando queste ultime come un volume EPI. L'obiettivo è sfruttare la forte correlazione lungo le linee epipolari per una compressione lossy efficiente.

3.3.1 Caricamento del Light Field

Il dataset è una griglia di viste (u, v) salvate come immagini a 16 bit (`.ppm`). Il codice:

- ordina i file e verifica che il numero totale sia $U \cdot V$;
- legge la prima immagine per determinare $H \times W \times C$ (`uint16`);
- alloca un tensore `lf_data` di shape (U, V, H, W, C) e lo popola con la mappatura: $u = idx \bmod U$, $v = \lfloor idx/U \rfloor$;

- se necessario, applica un resize per uniformare le dimensioni.

3.3.2 Dal Light Field al volume EPI (tiled)

Gli EPI si ottengono fissando una coordinata spaziale e variando l'angolo. Operativamente:

1. trasposizione $(U, V, H, W, C) \rightarrow (V, H, U, W, C)$ per ottenere sezioni indizzate per (v, y) ;
2. `reshape` in `epi_raw` di shape $((V \cdot H), U, W, C)$, dove ogni EPI ha altezza U e larghezza W ;
3. padding sul numero di EPI (replicando l'ultima EPI) per ottenere un multiplo di `TILES_PER_FRAME` (nel codice: 64);
4. impilamento verticale di $T = \text{TILES_PER_FRAME}$ EPI per formare un frame video: altezza frame = $T \cdot U$, larghezza = W .

Il risultato è `tiled_volume` di shape $(F, T \cdot U, W, C)$, dove F è il numero di frame.

3.3.3 Compressione con codec video

La sequenza di frame EPI è compressa con FFmpeg utilizzando gli stessi parametri scelti per la compressione standard con codec video, descritti nella sezione 3.2.3. L'unica differenza risiede nella gestione dell'input dei frame a FFmpeg, dato che questi vengono passati uno per volta tramite pipe dallo script Python. Per questo motivo, l'input raw deve essere gestito tramite canali a 16 bit per pixel, quindi con `-pix_fmt bgr48le`, e i dati a 10 bit devono essere ampliati per ricoprire l'intero spazio a 16 bit, tramite il filtro:

```
-vf "lutrgb=r=val*64:g=val*64:b=val*64"
```

In caso fosse necessario, anche qui viene aggiunto del padding ai frame per ottenere dimensioni multiple di 8. Questo, però, viene fatto direttamente sul volume EPI prima che venga dato in input a FFmpeg.

3.3.4 Decodifica e Ricostruzione del Volume EPI

La procedura di ricostruzione del Light Field a partire dal flusso video EPI compresso è speculare alla fase di codifica, ma richiede una manipolazione rigorosa delle strutture dati multidimensionali per ripristinare la coerenza

angolare e spaziale.

Il processo è gestito dallo script Python e si articola nelle seguenti fasi sequenziali:

1. **Decodifica del flusso video:** Il file video compresso viene processato da FFmpeg. Viene applicato il filtro `lutrgb` inverso (divisione per 64) per riportare i valori dei pixel nel range dinamico corretto. L'output video decompresso viene trasferito direttamente nella memoria dello script Python tramite pipe, evitando la scrittura di file temporanei su disco.
2. **Rimozione del Padding Spaziale:** Il buffer grezzo ricevuto viene interpretato da NumPy come un tensore 4D. A differenza della compressione video standard dove il ritaglio è affidato a FFmpeg, qui lo script agisce direttamente sulla matrice numerica: se durante la codifica erano stati aggiunti pixel neri per raggiungere dimensioni multiple di 8, questi vengono rimossi tramite slicing dell'array, ripristinando le dimensioni esatte del volume tiled ($W \times H_{tiled}$).
3. **Ricostruzione del Volume EPI:** L'array viene ristrutturato (*reshape*) per separare le tile impilate verticalmente nei frame video. In questa fase vengono scartati anche eventuali "frame di padding" temporali, ovvero EPI vuote che erano state aggiunte in coda per completare l'ultimo frame video (multiplo di 64 tile).
4. **Trasposizione Inversa (5D):** Il passaggio cruciale consiste nel riorganizzare i dati dalla rappresentazione EPI alla rappresentazione Light Field classica. Attraverso operazioni di trasposizione sugli assi del tensore, si inverte la trasformazione geometrica iniziale, mappando le coordinate delle EPI indietro alle coordinate delle viste originali:

$$\text{Volume EPI} \xrightarrow{\text{Reshape \& Transpose}} (U, V, H, W, C)$$

5. **Salvataggio e Patching:** Infine, le viste ricostruite vengono estratte dal tensore 5D e salvate singolarmente. Analogamente alla sezione 3.2.4, viene applicato il patch binario all'header dei file PPM risultanti per correggere il valore massimo di profondità colore da 65535 a 1023.

3.4 Implementazione JPEG Pleno

Questa sezione descrive l'utilizzo del framework di codifica JPEG Pleno per la compressione delle immagini Light Field. L'implementazione si basa sul software di riferimento standard, orchestrato tramite lo script Python `jpl-processor.py` sviluppato per automatizzare l'intero workflow di codifica e decodifica, dalla gestione dei dati grezzi alla ricostruzione finale.

3.4.1 Workflow di elaborazione

Lo script gestisce un flusso di lavoro sequenziale composto da sei fasi distinte, necessarie per adattare i diversi tipi di dataset (*lenslet*, *synthetic* e *HDCA*) alle specifiche richieste dall'encoder JPEG Pleno e viceversa. Il processo è suddiviso in due macro-fasi:

- **Fase di Codifica (Step 1-3):** Preprocessing delle immagini raw, conversione di formato e codifica JPL.
- **Fase di Decodifica (Step 4-6):** Decodifica del bitstream, riconversione di formato e post-processing per il ripristino della geometria originale.

3.4.2 Preprocessing e Conversione (Step 1-2)

Alcuni dei dataset utilizzati (*Bikes* e *Danger de Mort*) sono acquisiti tramite sensori plenottici (*lenslet*). Per questi ultimi, prima della codifica, è necessario riallineare le viste per gestire la disposizione esagonale delle microlenti (Step 1).

1. **Lenslet Shifting:** Viene utilizzato il tool `lenslet_13x13_shifter` in modalità *encode*. Questo strumento elabora i file PPM originali e produce una versione "shifted", riorganizzando i pixel per renderli compatibili con la trasformata 4D utilizzata dallo standard.
2. **Conversione in PGX:** Il software di riferimento JPEG Pleno opera nativamente sul formato PGX (JPEG 2000). Pertanto, le immagini PPM shiftate vengono convertite individualmente in file PGX non compressi tramite l'utilità `convert_ppm_to_pgx`.

3.4.3 Encoding (Step 3)

La fase di codifica vera e propria è eseguita dal binario `jpl-encoder-bin`. Lo script configura l'encoder per utilizzare la modalità 4D Transform, ideale per sfruttare la ridondanza sia spaziale (intra-view) che angolare (inter-view). I principali parametri di configurazione passati all'encoder sono:

- **Modalità di Trasformata:** Viene impostato `--type 0` e `--part 2`, attivando la codifica basata su trasformata 4D DCT.
- **Spazio Colore:** Si utilizza lo spazio YCbCr_2 per separare la luminanza dalla crominanza, migliorando l'efficienza di compressione.
- **Dimensioni della Trasformata:** Lo script calcola dinamicamente le dimensioni dei blocchi. Per le dimensioni spaziali (intra-view), viene impostata una dimensione massima di 32×32 pixel, mentre per le dimensioni angolari (inter-view) si utilizzano le dimensioni della griglia del light field (ad es. 13×13).
- **Parametro Lambda (λ):** Questo parametro regola il trade-off tra bitrate e distorsione (Rate-Distortion Optimization). Lo script permette di specificare un valore manuale o di utilizzarne uno di default (700), che funge da fattore di qualità lagrangiano: valori più alti di λ comportano una maggiore compressione e minore qualità.

Il risultato di questa fase è un singolo file con estensione `.jpl` contenente l'intero light field compresso.

3.4.4 Decoding e Post-processing (Step 4-6)

Per valutare la qualità della compressione, il file `.jpl` viene decodificato e riportato al formato originale per il confronto con il riferimento.

1. **Decoding:** Il binario `jpl-decoder-bin` decomponе il file `.jpl` ritornando all'insieme di viste nel formato intermedio PGX (Step 4).
2. **Conversione in PPM:** I file PGX decodificati vengono riconvertiti nel formato PPM (Step 5).
3. **Lenslet Post-processing:** Infine, sui dataset *lenslet*, viene applicato nuovamente il tool `lenslet_13x13_shifter`, questa volta in modalità *decode*. Questa operazione inverte lo shift geometrico applicato nella fase 1, restituendo le viste ricostruite (Step 6).

Capitolo 4

Risultati

In questo capitolo vengono presentati e discussi i risultati sperimentali ottenuti applicando le diverse strategie di compressione analizzate ai dataset Light Field di riferimento. L’obiettivo è valutare le performance relative di JPEG Pleno, dei codec video standard (HEVC, AV1, VP9) e dell’approccio basato su EPI, mettendo a confronto l’efficienza di compressione dei diversi approcci.

Per garantire la coerenza delle misurazioni, in particolare per quanto riguarda i tempi di codifica e decodifica, tutti i test sono stati eseguiti su un’unica piattaforma hardware dedicata, configurata come segue:

- **CPU:** AMD Ryzen 7 6800H (8 cores, SMT, 3.2 GHz base clock, fino a 4.7 GHz boost);
- **GPU:** NVIDIA GeForce RTX 3060 Mobile (3840 CUDA cores, 900 MHz base clock, fino a 1425 MHz boost);
- **Sistema Operativo:** Windows 11 con ambiente di esecuzione Windows Subsystem for Linux (WSL).

Nonostante ciò, nell’analisi dei risultati non ci si è soffermati sulle tempistiche di compressione e decompressione, dato che queste ultime possono essere fortemente influenzate dall’hardware e dal software utilizzato. In particolare, il software di riferimento JPLM non è stato sviluppato per ottimizzare le velocità di encoding e decoding [1].

Per quanto concerne i dataset di tipo *lenslet* (*Bikes* e *Danger de Mort*), è importante sottolineare che le analisi sono state condotte utilizzando le viste in formato *shifted* (sia per l’input raw che per l’output decompresso). Poiché

questi rappresentano i dati effettivamente sottoposti al processo di codifica, l'operazione di shifting geometrico (pre- e post-processing) è stata considerata computazionalmente trascurabile e non influente ai fini del confronto diretto tra i codec.

Al fine di comparare adeguatamente architetture di compressione così eterogenee, è stato necessario adottare una metodologia rigorosa basata sull'allineamento della qualità visiva. I diversi encoder sono controllati da parametri non omogenei (il parametro λ per JPEG Pleno e il fattore CRF per i codec video); pertanto, è stata effettuata una calibrazione sperimentale di tali parametri per ciascun dataset. L'obiettivo è stato ottenere immagini ricostruite con un indice di similarità strutturale (SSIM) pressoché identico tra le varie tecniche, minimizzando lo scarto statistico. In questo modo, le differenze osservate in termini di bitrate e tempi di elaborazione possono essere considerate indicative della reale efficienza dei codec a parità di qualità percettiva.

4.1 Analisi dei Risultati: Dataset *Bikes*

Il primo dataset analizzato è *Bikes*, un esempio rappresentativo di acquisizione plenottica tramite array di microlenti (lenslet). I dati di partenza sono caratterizzati da una griglia angolare originale di 15×15 viste. A seguito della fase di preprocessing (*lenslet shifting*), necessaria per gestire la disposizione esagonale delle microlenti e rendere i dati compatibili con le pipeline di compressione, la griglia effettiva processata risulta essere di 13×13 viste, per un totale di 169 immagini. Ogni vista possiede una risoluzione spaziale di 625×434 pixel. La dimensione complessiva del dataset RAW in formato *shifted* (uncompressed) è pari a 262.31 MB.

4.1.1 Qualità Visiva

La Figura 4.1 mostra un confronto visivo tra un dettaglio della vista centrale originale e la corrispettiva ricostruzione ottenuta con JPEG Pleno. Come premesso nell'introduzione al capitolo, i parametri di quantizzazione (λ per JPEG Pleno e CRF per i codec video) sono stati finemente tarati per garantire un indice SSIM medio molto elevato (≈ 0.954). Grazie a questa scelta metodologica, è possibile osservare che non vi è alcun degrado visivamente percepibile: le immagini compresse risultano indistinguibili dall'originale ad occhio nudo, e non presentano artefatti di blocco o ringing

evidenti, garantendo così una valutazione focalizzata puramente sull'efficienza di compressione a parità di fedeltà.



Figura 4.1: Immagini di esempio del Dataset Bikes (006_006 shifted). A sinistra l'immagine RAW e a destra l'immagine in seguito alla compressione con JPEG Pleno. Con gli altri codec sono stati ottenuti risultati qualitativamente equivalenti.

4.1.2 Efficienza di Compressione

I risultati quantitativi relativi alla dimensione dei file compressi sono riportati nelle Tabelle 4.1 e 4.2, inoltre sono visualizzati graficamente nell'istogramma di Figura 4.2.

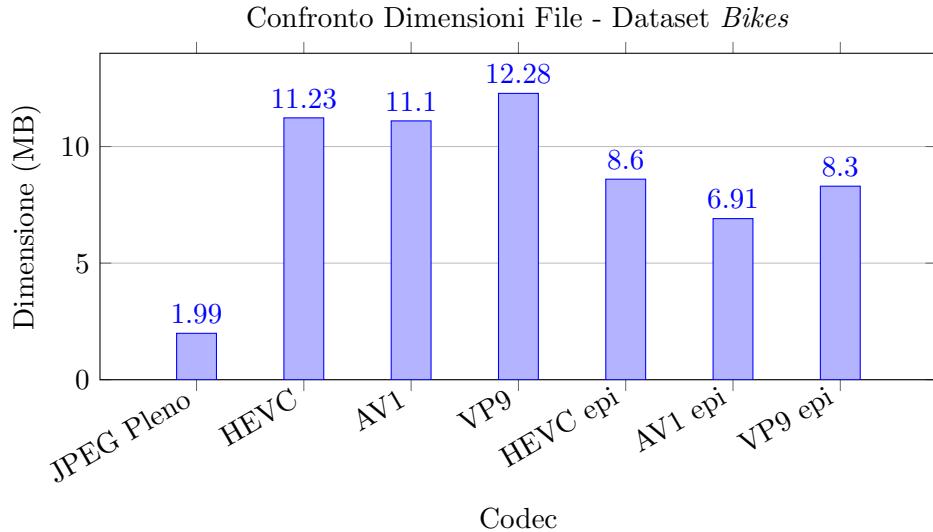


Figura 4.2: Dimensioni finali in MB per il Dataset Bikes. I risultati sono ottenuti a parità di qualità visiva approssimata, con un SSIM medio di 0.9544 ± 0.0008 .

Dall’analisi dei dati emergono diverse considerazioni significative:

- **Supremazia di JPEG Pleno:** Il framework standardizzato JPEG Pleno (4D Transform Mode) dimostra prestazioni nettamente superiori rispetto a tutte le altre soluzioni, comprimendo il dataset a soli 1.99 MB. Questo risultato conferma l’efficacia della trasformata 4D nativa nello sfruttare l’elevata correlazione presente nei light field densi di tipo lenslet.
- **Vantaggio dell’approccio EPI:** Confrontando le due strategie basate su codec video, l’approccio EPI (*Epipolar Plane Images*) risulta sistematicamente più efficiente rispetto alla codifica delle viste come sequenza pseudo-temporale classica. Ad esempio, il codec AV1 riduce la dimensione da 11.10 MB (modalità Video) a 6.91 MB (modalità EPI), evidenziando come la coerenza geometrica delle EPI sia più facile da predire per gli encoder moderni rispetto alla semplice successione delle viste.
- **Confronto tra Codec:** Tra i codec video testati in modalità EPI, AV1 si conferma il più performante (6.91 MB), seguito da VP9 (8.30 MB) e HEVC (8.60 MB). Sebbene nessuno di questi riesca a raggiungere i tassi di compressione di JPEG Pleno, l’uso di AV1 con le EPI rappresenta la migliore alternativa ”general-purpose” tra quelle analizzate.

Tabella 4.1: Parametri e metriche di compressione - Dataset Bikes

Codec	Size (MB)	Ratio	Saved (%)	Bitrate (bpp)	λ/CRF
JPEG Pleno	1.99	131.89:1	99.24	0.3639	700
HEVC	11.23	23.35:1	95.72	2.0559	2
AV1	11.10	23.64:1	95.77	2.0307	5
VP9	12.28	21.37:1	95.32	2.2465	7
HEVC epi	8.60	30.50:1	96.72	1.5740	14
AV1 epi	6.91	37.95:1	97.37	1.2647	11
VP9 epi	8.30	31.61:1	96.84	1.5185	14

Tabella 4.2: Metriche di qualità dell'immagine e prestazioni temporali - Dataset Bikes

Codec	PSNR	PSNR σ	SSIM	SSIM σ	Comp. (s)	Dec. (s)
JPEG Pleno	35.47	0.74	0.954 236	0.011 129	51.94	7.9
HEVC	33.40	0.88	0.954 242	0.007 965	5.10	4.1
AV1	33.54	0.94	0.955 763	0.007 286	34.00	4.4
VP9	33.44	0.88	0.954 319	0.007 025	19.50	4.4
HEVC epi	34.72	0.68	0.954 203	0.010 644	4.70	2.01
AV1 epi	34.94	0.64	0.952 950	0.010 851	26.19	1.74
VP9 epi	34.99	0.62	0.954 915	0.010 185	19.92	1.69

4.2 Analisi dei Risultati: Dataset *Danger De Mort*

Il secondo dataset di tipo *lenslet* preso in esame è *Danger de Mort*. Analogamente al caso precedente, i dati sono stati acquisiti tramite un array di microlenti con una griglia nativa di 15×15 elementi. Il processo di preprocessing e shifting ha prodotto un output effettivo di 13×13 viste (169 immagini totali), ciascuna con una risoluzione di 625×434 pixel. Essendo le caratteristiche geometriche e la profondità di colore (16 bit utilizzati, 10 bit effettivi) identiche a quelle del dataset *Bikes*, la dimensione totale del volume RAW shiftato rimane invariata a 262.31 MB.

4.2.1 Qualità Visiva

Anche per questo dataset, la calibrazione dei parametri di compressione è stata guidata dall'obiettivo di mantenere un'elevata fedeltà visiva. Come evidenziato dai dettagli in Figura 4.3, la qualità percettiva delle immagini ricostruite è eccellente e priva di difetti manifesti. L'indice SSIM medio ottenuto è pari a 0.9369 ± 0.0008 ; sebbene questo valore sia numericamente inferiore a quello di *Bikes*, rappresenta comunque una soglia di "trasparenza" visiva dove le differenze rispetto all'originale sono trascurabili per l'osservatore umano.



Figura 4.3: Immagini di esempio del Dataset Danger De Mort (006_006 shifted). A sinistra l'immagine RAW e a destra l'immagine in seguito alla compressione con JPEG Pleno. Con gli altri codec sono stati ottenuti risultati qualitativamente equivalenti.

4.2.2 Efficienza di Compressione

I risultati quantitativi sono riassunti nelle Tabelle 4.3 e 4.4, inoltre sono illustrati nel diagramma a barre di Figura 4.4.

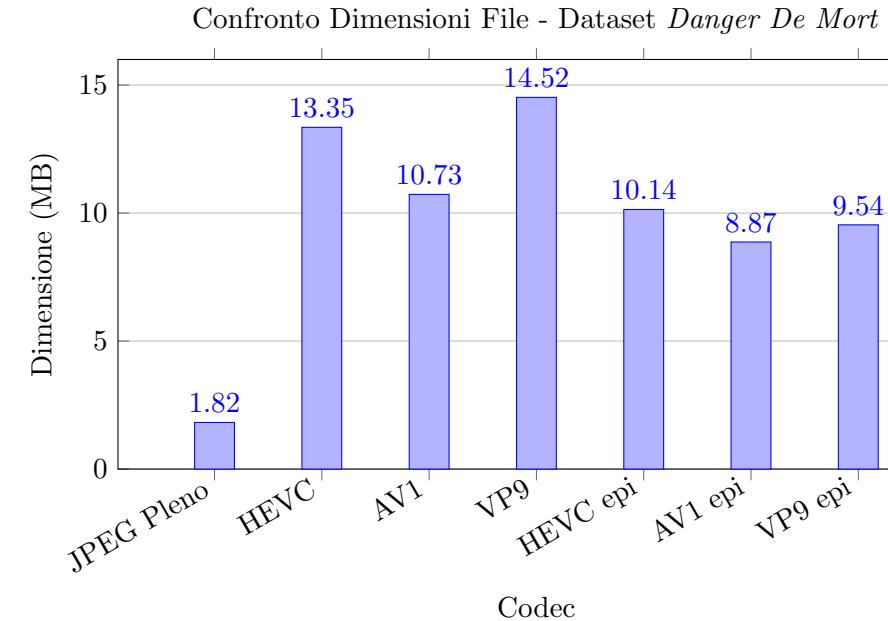


Figura 4.4: Dimensioni finali in MB per il Dataset Danger De Mort. I risultati sono ottenuti a parità di qualità visiva approssimata, con un SSIM medio di 0.9369 ± 0.0008 .

L'analisi dei dati conferma i trend generali osservati in precedenza, con alcune specificità:

- **Dominio di JPEG Pleno:** Con una dimensione finale di soli 1.82 MB, JPEG Pleno si conferma la soluzione di gran lunga più efficiente, offrendo un rapporto di compressione superiore a 140:1 rispetto al raw. Questo ribadisce come l'architettura 4D-DCT sia estremamente robusta indipendentemente dal contenuto specifico della scena *lenslet*.
- **Conferma dell'approccio EPI:** Anche in questo scenario, la riorganizzazione dei dati in *Epipolar Plane Images* porta benefici tangibili rispetto alla compressione video sequenziale classica. Ad esempio, il codec AV1 passa da 10.73 MB (Video) a 8.87 MB (EPI), mentre VP9 scende da 14.52 MB a 9.54 MB. Il miglioramento è consistente, seppur leggermente meno marcato in percentuale rispetto al dataset *Bikes*, suggerendo che la struttura delle EPI in questa scena potrebbe essere più complessa da predire.
- **Gerarchia dei Codec:** AV1 (in modalità EPI) si mantiene saldo come la migliore alternativa ai codec plenottici dedicati, distaccando significativamente sia HEVC (10.14 MB) che VP9 (9.54 MB). È interessante notare come, nella modalità video classica, HEVC (13.35 MB) risulti meno efficiente di AV1, confermando la superiorità delle tecnologie di codifica royalty-free più moderne in questo specifico contesto applicativo.

Tabella 4.3: Parametri e metriche di compressione - Dataset Danger De Mort

Codec	Size (MB)	Ratio	Saved (%)	Bitrate (bpp)	λ/CRF
JPEG Pleno	1.82	144.14:1	99.31	0.3330	1100
HEVC	13.35	19.65:1	94.91	2.4427	2
AV1	10.73	24.45:1	95.91	1.9631	6
VP9	14.52	18.07:1	94.47	2.6566	7
HEVC epi	10.14	25.86:1	96.13	1.8561	14
AV1 epi	8.87	29.57:1	96.62	1.6235	11
VP9 epi	9.54	27.50:1	96.36	1.7457	15

Tabella 4.4: Metriche di qualità dell'immagine e prestazioni temporali - Dataset Danger De Mort

Codec	PSNR	PSNR σ	SSIM	SSIM σ	Comp. (s)	Dec. (s)
JPEG Pleno	33.41	0.57	0.937 141	0.013 586	61.10	8.5
HEVC	31.76	0.75	0.936 914	0.006 600	4.00	4.4
AV1	31.77	0.79	0.935 396	0.007 670	31.40	4.5
VP9	31.82	0.76	0.937 671	0.005 998	19.80	4.7
HEVC epi	32.46	0.65	0.936 591	0.011 002	5.63	2.01
AV1 epi	32.75	0.62	0.938 253	0.010 100	30.04	1.8
VP9 epi	32.56	0.61	0.936 624	0.009 973	20.01	1.78

4.3 Analisi dei Risultati: Dataset *Greek*

Il terzo dataset analizzato è *Greek*, appartenente alla categoria dei dataset sintetici (HCI). A differenza dei casi precedenti acquisiti tramite microlenti, questo dataset presenta una struttura a griglia più sparsa di 9×9 viste (81 immagini totali). Le singole immagini hanno una risoluzione quadrata di 512×512 pixel. La dimensione originale del volume non compresso è di 121.50 MB. Essendo un dataset sintetico generato al computer, la geometria è perfetta e non richiede operazioni di rettifica o shifting preliminare.

4.3.1 Qualità Visiva

Anche in questo caso, la priorità è stata mantenere una qualità visiva eccellente per garantire un confronto equo. L'indice SSIM medio ottenuto è estremamente elevato, pari a 0.9763 ± 0.0005 . Come visibile in Figura 4.5, le immagini ricostruite non presentano artefatti; la natura sintetica e pulita del dataset originale facilita il mantenimento di bordi nitidi e texture definite anche a tassi di compressione elevati.

4.3.2 Efficienza di Compressione

I risultati quantitativi (Tabelle 4.5 e 4.6, Figura 4.6) mostrano un'inversione di tendenza significativa rispetto ai dataset *lenslet*, evidenziando come la struttura del Light Field influenzi drasticamente le performance degli algoritmi.



Figura 4.5: Immagini di esempio del Dataset Greek (004_004). A sinistra l'immagine RAW e a destra l'immagine in seguito alla compressione con JPEG Pleno. Con gli altri codec sono stati ottenuti risultati qualitativamente equivalenti.

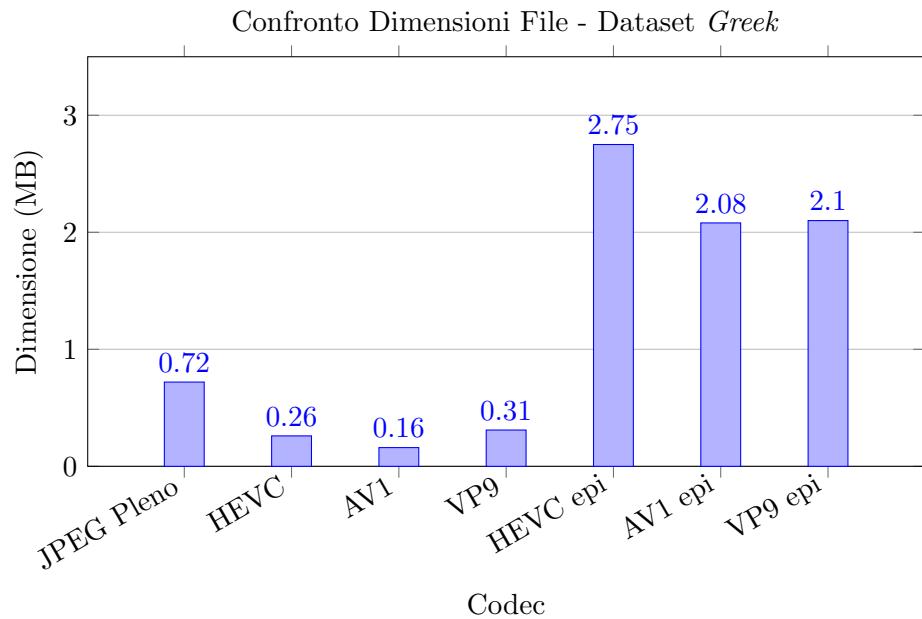


Figura 4.6: Dimensioni finali in MB per il Dataset Greek. I risultati sono ottenuti a parità di qualità visiva approssimata, con un SSIM medio di 0.9763 ± 0.0005 .

In particolare, notiamo:

- **Primato dei Codec Video Standard:** In questo scenario, la codifica video sequenziale classica (HEVC, AV1, VP9) si rivela la strategia vincente. In particolare, **AV1** raggiunge una dimensione finale incredibilmente ridotta di soli **0.16 MB**, seguito da HEVC (0.26 MB). Questo suggerisce che, in presenza di scene sintetiche con disparità ampie ma regolari, gli algoritmi di stima del movimento (Inter-frame prediction) dei codec video sono estremamente efficaci nel predire le viste successive.
- **Limitazioni di JPEG Pleno (4D-TM):** Il framework JPEG Pleno, pur ottenendo un buon risultato (0.72 MB), viene superato dai codec video. Questo comportamento conferma quanto descritto nella letteratura tecnica (si veda Sezione 2.4): la modalità *4D Transform Mode* è ottimizzata per Light Field densi. Quando la distanza tra le viste aumenta (wide baseline), la correlazione 4D diminuisce, riducendo l'efficienza della compattazione energetica rispetto alla predizione temporale dei codec video.
- **Inefficienza dell'approccio EPI:** L'approccio basato su EPI risulta il meno performante in assoluto (es. AV1 EPI a 2.08 MB). La causa è da ricercarsi nella geometria della scena: con una baseline più ampia, le linee nelle immagini EPI diventano molto inclinate e potenzialmente discontinue (aliasing angolare). Queste caratteristiche rendono le "texture" EPI complesse e ricche di alte frequenze, difficili da comprimere per i codec video rispetto alle immagini naturali delle viste originali.

Tabella 4.5: Parametri e metriche di compressione - Dataset Greek

Codec	Size (MB)	Ratio	Saved (%)	Bitrate (bpp)	λ/CRF
JPEG Pleno	0.72	169.23:1	99.41	0.2836	200
HEVC	0.26	458.67:1	99.78	0.1047	14
AV1	0.16	782.92:1	99.87	0.0613	22
VP9	0.31	392.61:1	99.75	0.1223	22
HEVC epi	2.75	44.19:1	97.74	1.0862	14
AV1 epi	2.08	58.43:1	98.29	0.8216	11
VP9 epi	2.10	57.76:1	98.27	0.8310	15

Tabella 4.6: Metriche di qualità dell'immagine e prestazioni temporali - Dataset Greek

Codec	PSNR	PSNR σ	SSIM	SSIM σ	Comp. (s)	Dec. (s)
JPEG Pleno	41.76	0.50	0.976 326	0.001 801	20.90	3.7
HEVC	41.42	0.74	0.976 383	0.002 746	2.70	2.5
AV1	41.29	0.33	0.976 103	0.001 059	6.60	1.8
VP9	41.47	0.76	0.976 546	0.002 436	4.60	2.4
HEVC epi	41.47	0.67	0.977 158	0.002 154	11.77	1.25
AV1 epi	41.65	0.57	0.975 221	0.002 113	13.23	1.24
VP9 epi	41.70	0.58	0.976 280	0.002 124	7.04	1.18

4.4 Analisi dei Risultati: Dataset *Sideboard*

Il quarto dataset preso in esame è *Sideboard*, anch'esso di natura sintetica e proveniente dalla suite HCI. Le caratteristiche strutturali sono identiche al dataset precedente: una griglia angolare di 9×9 viste (81 immagini totali) con risoluzione spaziale di 512×512 pixel. La dimensione del volume originale non compresso è pari a 121.50 MB. Essendo dati generati artificialmente, la geometria è nativamente rettificata e non ha richiesto le operazioni di pre-processing necessarie per i dataset lenslet.

4.4.1 Qualità Visiva

La validazione qualitativa ha prodotto risultati eccellenti, con un indice SSIM medio di 0.9742 ± 0.0002 . Come illustrato in Figura 4.7, le immagini decompresse mantengono una fedeltà visiva elevatissima rispetto all'originale. La natura sintetica della scena, caratterizzata da texture dettagliate e illuminazione controllata, viene preservata efficacemente da tutti i metodi di compressione testati alla qualità target stabilita.

4.4.2 Efficienza di Compressione

L'analisi dei dati quantitativi (Tabelle 4.7 e 4.8, Figura 4.8) conferma la predisposizione dei dataset sintetici *wide-baseline* verso la compressione video, ma con sfumature diverse rispetto al caso *Greek*.



Figura 4.7: Immagini di esempio del Dataset Sideboard (004_004). A sinistra l'immagine RAW e a destra l'immagine in seguito alla compressione con JPEG Pleno. Con gli altri codec sono stati ottenuti risultati qualitativamente equivalenti.

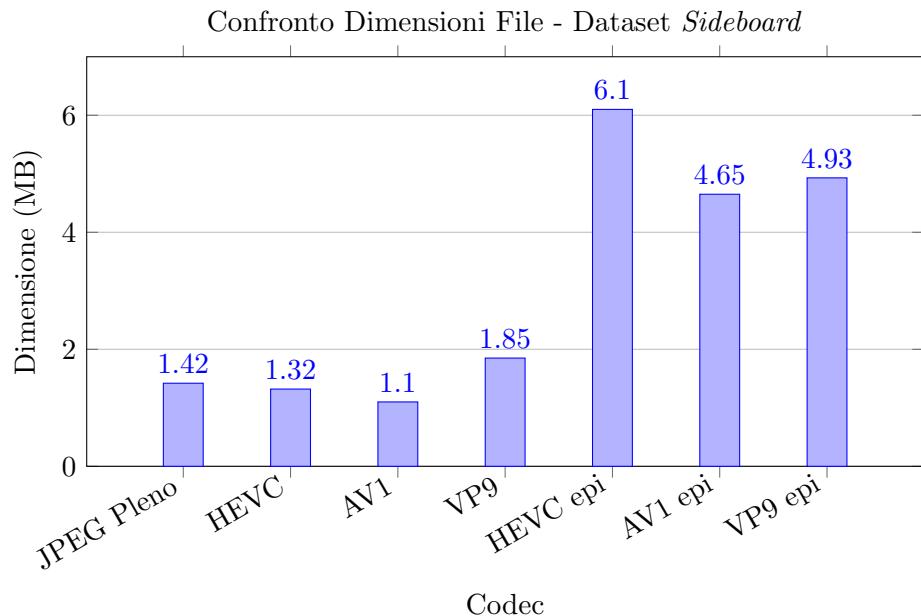


Figura 4.8: Dimensioni finali in MB per il Dataset Sideboard. I risultati sono ottenuti a parità di qualità visiva approssimata, con un SSIM medio di 0.9742 ± 0.0002 .

Dai risultati possiamo notare:

- **Conferma di AV1 e HEVC:** La codifica video standard (modalità Pseudo-Video) si conferma la strategia più efficace per questa tipologia di dati. **AV1** ottiene la dimensione minore in assoluto (1.10 MB), seguito a breve distanza da **HEVC** (1.32 MB). L'elevata coerenza temporale simulata dalla scansione delle viste permette a questi codec di sfruttare al meglio la predizione inter-frame.
- **Competitività di JPEG Pleno:** A differenza del dataset *Greek*, dove il distacco era netto, in questo scenario **JPEG Pleno** (1.42 MB) mostra prestazioni molto competitive. Esso riesce a superare il codec VP9 (1.85 MB) e si posiziona molto vicino a HEVC. Questo risultato indica che, all'aumentare della complessità testuale della scena (il dataset *Sideboard* è più ricco di dettagli ad alta frequenza rispetto a *Greek*), l'efficienza della trasformata 4D-DCT nel compattare l'energia diventa comparabile a quella della predizione di movimento dei codec video di generazione precedente.
- **Difficoltà dell'approccio EPI:** Si conferma invece la scarsa idoneità dell'approccio EPI per i dataset a baseline ampia. Con dimensioni che oscillano tra 4.65 MB (AV1) e 6.10 MB (HEVC), questa modalità produce file significativamente più grandi rispetto alle alternative. Le forti discontinuità angolari presenti nelle immagini EPI di scene sintetiche continuano a rappresentare un ostacolo difficile da comprimere efficientemente rispetto alle viste prospettiche naturali.

Tabella 4.7: Parametri e metriche di compressione - Dataset Sideboard

Codec	Size (MB)	Ratio	Saved (%)	Bitrate (bpp)	λ/CRF
JPEG Pleno	1.42	85.66:1	98.83	0.5603	700
HEVC	1.32	91.87:1	98.91	0.5225	11
AV1	1.10	110.09:1	99.09	0.4360	13
VP9	1.85	65.56:1	98.47	0.7321	17
HEVC epi	6.10	19.93:1	94.98	2.4081	13
AV1 epi	4.65	26.12:1	96.17	1.8379	11
VP9 epi	4.93	24.65:1	95.94	1.9476	14

Tabella 4.8: Metriche di qualità dell’immagine e prestazioni temporali - Dataset Sideboard

Codec	PSNR	PSNR σ	SSIM	SSIM σ	Comp. (s)	Dec. (s)
JPEG Pleno	34.79	0.62	0.974 281	0.003 123	28.00	3.7
HEVC	34.36	0.56	0.974 023	0.002 848	2.70	2.5
AV1	34.55	0.13	0.974 422	0.000 823	11.10	2.4
VP9	34.53	0.54	0.974 113	0.002 742	7.10	2.5
HEVC epi	34.14	1.02	0.974 180	0.005 267	3.90	1.29
AV1 epi	34.62	1.00	0.974 473	0.004 978	17.32	1.19
VP9 epi	34.48	1.00	0.973 978	0.005 018	8.21	1.12

4.5 Analisi dei Risultati: Dataset *Tarot*

L’ultimo dataset oggetto di analisi è *Tarot*, il più oneroso dell’intero set di prova in termini di risorse computazionali e spazio di archiviazione. Si tratta di un Light Field ad alta densità *High Density Camera Array (HDCA)* caratterizzato da una griglia angolare molto estesa di 17×17 viste (ben 289 immagini totali). Oltre alla densità angolare, anche la risoluzione spaziale è la più alta tra i casi studio, pari a 1024×1024 pixel per vista. La combinazione di queste caratteristiche porta la dimensione del volume RAW originale alla cifra considerevole di **1734 MB** (circa 1.7 GB).

4.5.1 Qualità Visiva

Nonostante la mole di dati, la metodologia di test a isocostanza di qualità è stata mantenuta rigorosa. L’indice SSIM medio registrato è di 0.9738 ± 0.0003 , un valore che denota un’elevatissima fedeltà. Come mostrato in Figura 4.9, i dettagli fini delle carte e le texture complesse della sfera di cristallo sono preservati senza artefatti visibili, confermando che i risultati di compressione discussi di seguito non sono frutto di un degrado dell’immagine.

4.5.2 Efficienza di Compressione

I risultati quantitativi (Tabella 4.9 e 4.10, Figura 4.10) delineano uno scenario di compressione estrema, dove le differenze architettonali tra i codec emergono con forza.



Figura 4.9: Immagini di esempio del Dataset Tarot (008_008). A sinistra l'immagine RAW e a destra l'immagine in seguito alla compressione con JPEG Pleno. Con gli altri codec sono stati ottenuti risultati qualitativamente equivalenti.

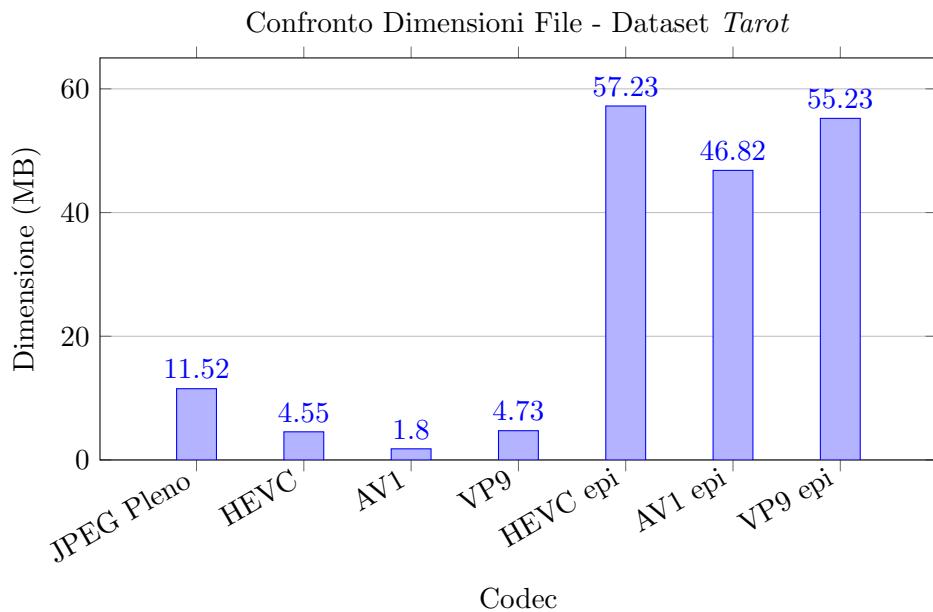


Figura 4.10: Dimensioni finali in MB per il Dataset Tarot. I risultati sono ottenuti a parità di qualità visiva approssimata, con un SSIM medio di 0.9738 ± 0.0003 .

Dai risultati ottenuti emergono le seguenti considerazioni:

- **Prestazioni Eccezionali di AV1:** Il codec **AV1** (modalità Video) ottiene un risultato straordinario, comprimendo i 1.7 GB originali a soli **1.80 MB**. Questo corrisponde a un rapporto di compressione prossimo a 1000:1. Un tale risultato dimostra come, su sequenze lunghe (289 frame) e ad alta risoluzione con variazioni prospettiche regolari, la predizione temporale e il partizionamento a blocchi di AV1 siano incredibilmente efficienti nel rimuovere la ridondanza.
- **Buona tenuta di JPEG Pleno:** Il framework JPEG Pleno si attesta a **11.52 MB**. Sebbene in valore assoluto sia un ottimo risultato (rapporto $\approx 150:1$), risulta significativamente meno efficiente rispetto ai codec video standard in questo specifico contesto. La trasformata 4D-DCT, pur efficace, fatica a competere con la sofisticata compensazione del movimento dei codec video su risoluzioni così elevate.
- **Criticità dell'approccio EPI:** L'approccio EPI mostra qui i suoi limiti più evidenti, producendo file tra i 46 MB (AV1) e i 57 MB (HEVC). La causa principale risiede nell'alta risoluzione spaziale (1024×1024): le immagini EPI risultanti contengono linee con pendenze molto elevate e dettagli ad altissima frequenza. Mentre i codec video "spalmano" questi dettagli nel tempo, trattare le EPI come fotogrammi video costringe l'encoder a preservare texture geometriche complesse e spesso discontinue, richiedendo un bitrate molto più alto per mantenere il target SSIM.

Tabella 4.9: Parametri e metriche di compressione - Dataset Tarot

Codec	Size (MB)	Ratio	Saved (%)	Bitrate (bpp)	λ/CRF
JPEG Pleno	11.52	150.56:1	99.34	0.3188	700
HEVC	4.55	380.96:1	99.74	0.1260	17
AV1	1.80	964.52:1	99.90	0.0498	31
VP9	4.73	366.82:1	99.73	0.1309	32
HEVC epi	57.23	30.30:1	96.70	1.5841	13
AV1 epi	46.82	37.03:1	97.30	1.2962	12
VP9 epi	55.23	31.39:1	96.81	1.5290	15

Tabella 4.10: Metriche di qualità dell'immagine e prestazioni temporali - Dataset Tarot

Codec	PSNR	PSNR σ	SSIM	SSIM σ	Comp. (s)	Dec. (s)
JPEG Pleno	36.99	0.50	0.973 710	0.002 051	316.5	48.9
HEVC	36.32	0.84	0.974 317	0.002 200	16.9	19.2
AV1	35.97	0.35	0.973 676	0.001 049	54.4	19.1
VP9	36.19	0.85	0.973 340	0.002 471	35.3	19.2
HEVC epi	36.57	2.32	0.974 020	0.007 871	22.95	9.58
AV1 epi	36.86	2.36	0.973 792	0.007 444	130.0	8.52
VP9 epi	36.96	2.41	0.973 736	0.007 627	65.0	9.56

Capitolo 5

Conclusioni

Il presente lavoro ha esplorato il complesso panorama della compressione di immagini Light Field, mettendo a confronto l'efficienza delle tecnologie di codifica video standardizzate (HEVC, AV1, VP9) con il framework specifico JPEG Pleno e con approcci ibridi basati sulla trasformazione in *Epipolar Plane Images* (EPI). L'analisi sperimentale, condotta su una piattaforma hardware unificata e metodologicamente rigorosa nel garantire la parità di qualità visiva (isocostanza SSIM), ha permesso di delineare un quadro prestazionale eterogeneo, dove non esiste un'unica soluzione dominante per ogni scenario applicativo.

5.1 Risultati Ottenuti

Dalle evidenze sperimentali emerge una netta distinzione nelle performance in funzione della tipologia di acquisizione e della densità angolare del Light Field:

- **Dominio di JPEG Pleno su Light Field Densi:** Nel caso di dataset acquisiti tramite microlenti (*lenslet*), come *Bikes* e *Danger de Mort*, l'architettura nativa 4D di JPEG Pleno (4D Transform Mode) si è dimostrata imbattibile. La capacità della trasformata 4D-DCT di sfruttare la micro-correlazione tra viste adiacenti ha prodotto file di dimensioni estremamente ridotte (fino a 1.82 MB), distaccando nettamente qualsiasi approccio basato su video codec.
- **Superiorità dei Codec Video su Light Field Sparsi:** Al contrario, nei dataset sintetici e a baseline ampia (*Greek*, *Sideboard*, *Tarot*), si è assistito a un ribaltamento delle gerarchie. La minore correlazione 4D

ha penalizzato l'efficienza di JPEG Pleno, mentre gli algoritmi di stima del movimento dei codec video (in particolare **AV1**) hanno eccelso nel gestire le ampie disparità, raggiungendo rapporti di compressione straordinari (come nel caso limite di *Tarot*, compresso a 1.80 MB contro gli 11.52 MB di JPEG Pleno).

- **Il ruolo dell'approccio EPI:** L'organizzazione dei dati in EPI si è rivelata una strategia valida ma condizionata dalla geometria della scena. Ha offerto vantaggi significativi rispetto alla codifica video sequenziale nei dataset *lenslet*, dove le EPI presentano strutture lineari continue. Tuttavia, si è dimostrata inefficace nei dataset sparsi, dove l'aliasing angolare e le forti pendenze delle linee epipolari introducono alte frequenze difficili da comprimere per gli encoder video attuali.
- **Confronto tra Codec Video:** Tra le tecnologie video analizzate, **AV1** è emerso sistematicamente come il codec più performante, superando sia HEVC che VP9 in quasi tutti gli scenari di test, confermandosi come la soluzione *royalty-free* più promettente per l'adattamento a dati volumetrici.

In conclusione, mentre JPEG Pleno rappresenta lo stato dell'arte per l'acquisizione plenottica densa (micro-lens array), i codec video moderni rimangono la scelta preferibile per array di telecamere o render sintetici con ampie distanze tra le viste, a meno di non adottare trasformazioni geometriche specifiche.

5.2 Sviluppi Futuri

Il campo della compressione Light Field è in rapida evoluzione. Alla luce dei risultati ottenuti e delle limitazioni osservate, si individuano diverse direzioni promettenti per la ricerca futura:

1. **Adozione dello Slanted 4D Transform Mode:** Come evidenziato dai risultati sui dataset sparsi, il 4D-TM standard soffre quando la disparità è elevata. L'integrazione della nuova modalità *Slanted* (attualmente in fase di standardizzazione come estensione di JPEG Pleno Parte 2) permetterebbe di colmare questo divario. Applicando una trasformazione geometrica adattiva per allineare le EPI prima della codifica, si potrebbe estendere l'efficienza di JPEG Pleno anche ai dataset *wide-baseline*, rendendolo competitivo anche laddove oggi prevale AV1.

2. **Ottimizzazione del Pre-processing EPI:** Per migliorare l'approccio EPI sui dataset sparsi, si potrebbero investigare tecniche di *shearlet transform* o filtri direzionali prima della compressione video, al fine di ridurre le discontinuità e l'energia delle alte frequenze che attualmente penalizzano il bitrate.
3. **Approcci Learning-Based:** Infine, seguendo la direzione intrapresa dalla Parte 6 dello standard (dedicata ai Point Cloud), un futuro sviluppo naturale riguarda l'applicazione di reti neurali profonde per la compressione Light Field. Tecniche come i *Neural Radiance Fields* (NeRF) o autoencoder variazionali 4D potrebbero offrire nuovi paradigmi per superare i limiti delle trasformate lineari classiche, specialmente nella ricostruzione di viste mancanti e nella gestione di scene con geometria complessa; a patto di avere a disposizione grandi moli di dati con cui effettuare l'addestramento.

Bibliografia

- [1] Pekka Astola et al. «JPEG Pleno: Standardizing a Coding Framework and Tools for Plenoptic Imaging Modalities». In: *ITU Journal: ICT Discoveries* 3.1 (giu. 2020).
- [2] ISO/IEC JTC 1/SC 29/WG 1.
JPEG White Paper: JPEG Pleno Holography.
White Paper ISO/IEC JTC 1/SC 29/WG1 N101066.
ISO/IEC JTC 1/SC 29/WG 1, 2025. URL: <https://jpeg.org/>.
- [3] *JPEG - JPEG Pleno.*
URL: <https://jpeg.org/jpegpleno/index.html>.
- [4] *JPEG Pleno Reference Software (JPLM).*
URL: <https://gitlab.com/wg1/jpeg-pleno-refsw>.
- [5] Carla Pagliari, Eduardo A. Barros da Silva e Ismael Seidel.
JPEG White Paper: JPEG Pleno Light Field Coding.
White Paper ISO/IEC JTC 1/SC 29/WG1 N101065. Version 2.0.
ISO/IEC JTC 1/SC 29/WG 1, gen. 2025. URL: <https://jpeg.org/>.
- [6] Stuart Perry e Luis Cruz.
JPEG White Paper: JPEG Pleno Point Cloud Coding.
White Paper ISO/IEC JTC 1/SC 29/WG1 N100984.
ISO/IEC JTC 1/SC 29/WG 1, 2024. URL: <https://jpeg.org/>.