

# Riassunto P.P.M. // Corso di Progettazione e Produzione Multimediale

Francesco Gori                      Elisa Farri

22 giugno 2024

## Indice

<b>1</b>	<b>Immagini Digitali</b>	<b>3</b>
1.1	Profondità di Bit (bit depth)	3
1.2	Formazione dell'Immagine a Colori	3
1.3	Fotocamere Digitali RGB	3
1.4	Algoritmi di Demosaicizzazione	3
1.5	Risoluzione di un'Immagine e Dimensioni dei Pixel	4
1.6	Dimensioni del Sensore nelle Fotocamere	4
1.7	Telecamere a Infrarossi e Termiche	4
1.8	Telecamere Neuromorfiche per Eventi	4
1.9	Risoluzione dello Schermo	5
1.10	Raccomandazioni per Immagini Web	5
<b>2</b>	<b>Rappresentazione di Immagini a Colori</b>	<b>6</b>
2.1	Spazi Colore	6
2.1.1	Spazio Colore CIE RGB	6
2.1.2	Spazio Colore CIE XYZ	6
2.1.3	Spazio Colore CIE xyY	6
2.2	Spazio Colore RGB e Derivati	7
2.3	Rappresentazione Truecolor	7
2.4	Dimensione di File Immagine e Compressione	7
<b>3</b>	<b>Compressione Lossless - Standard di Compressione di Immagini Digitali</b>	<b>8</b>
3.1	Algoritmi di Compressione Lossless	8
3.1.1	Run-Length Coding	8
3.1.2	Prefix-free Coding	8
3.1.3	Huffman Coding	8
3.1.4	LZW Coding	9
3.2	GIF - Graphic Interchange Format	10
3.2.1	Animazione GIF	10
3.2.2	Trasparenza GIF	10
3.3	PNG - Portable Network Graphics	10
3.3.1	Compressione DEFLATE	11
3.3.2	Compressione LZ77	11

<b>4</b>	<b>Compressione Lossy - Standard di Compressione di Immagini Digitali</b>	<b>12</b>
4.1	JPEG - Joint Photographic Experts Group . . . . .	12
4.1.1	Spazi Colore YCbCr e Y'CbCr . . . . .	12
4.1.2	Chroma Subsampling . . . . .	12
4.1.3	Fasi Principali dell'Algoritmo . . . . .	13
4.1.4	DCT - Trasformata Discreta del Coseno . . . . .	13
4.1.5	Quantizzazione . . . . .	14
4.1.6	Scansione a Zig-Zag . . . . .	14
4.1.7	Codifica dei Coefficienti . . . . .	14
4.1.8	Codifica Huffman . . . . .	15
4.2	Standard a Confronto - GIF, PNG, JPEG . . . . .	16
<b>5</b>	<b>Video Digitali</b>	<b>17</b>
5.1	Formati di Cattura . . . . .	17
5.1.1	Video Interlacciati . . . . .	17
5.2	Spazi Colore . . . . .	17
5.3	Codifica . . . . .	18
5.4	Risoluzione . . . . .	18
5.5	Aspect Ratio - Proporzioni . . . . .	18
5.6	Bitrate . . . . .	18
<b>6</b>	<b>Compressione di Video Digitali</b>	<b>19</b>
6.1	Sampling . . . . .	19
6.2	Standard di Compressione . . . . .	20
6.2.1	Compressione MPEG . . . . .	20
6.2.2	Container Video . . . . .	20
6.3	MPEG-1 H.261 . . . . .	20
6.3.1	Struttura del Bit-Stream . . . . .	21
6.3.2	GOP - Group of Pictures . . . . .	21
6.3.3	Macroblocks . . . . .	22
6.3.4	Macroblock Encoding . . . . .	22
6.3.5	Motion Estimation . . . . .	22
6.3.6	Search Matching Techniques . . . . .	23
6.3.7	MPEG1 Encoder . . . . .	23
6.4	MPEG-2, MPEG-4 H.264, MPEG-4 H.265 . . . . .	23

# 1 Immagini Digitali

Le immagini digitali sono istantanee elettroniche di una scena o scansionate da documenti, come fotografie, manoscritti, testi stampati e opere d'arte.

Esse vengono campionate e mappate come griglie di pixel, a ciascuno dei quali viene associato un valore tonale in codice binario (nero, bianco, scala di grigi o di colore).

I bit di ogni pixel vengono memorizzati da un computer e sono spesso compressi con rappresentazioni matematiche.

## 1.1 Profondità di Bit (bit depth)

La profondità di bit è determinata dal numero di bit utilizzati per definire ciascun pixel, ad esempio:

- Un'immagine bitonale è composta da pixel costituiti da 1 bit ciascuno (tipicamente bianco o nero)
- Un'immagine in scala di grigi è composta da pixel rappresentati ciascuno da più bit di informazione (in genere da 2 a 8, o anche più)
- Un'immagine a colori è tipicamente rappresentata con una bit depth tra 8 e 24 o più

## 1.2 Formazione dell'Immagine a Colori

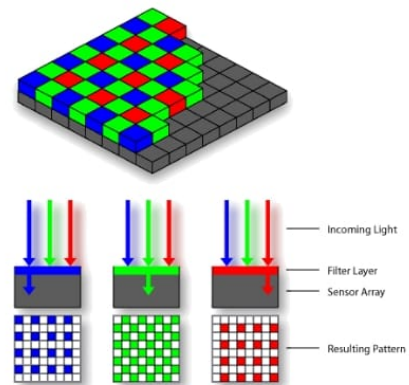
La formazione dell'immagine a colori è determinata dalla distribuzione della potenza radiante della luce incidente, dalla riflessione dei materiali e dalle caratteristiche della telecamera.

## 1.3 Fotocamere Digitali RGB

Le immagini digitali sono oggi ottenute da fotocamere digitali che usano sensori CMOS o CCD per acquisire tre segnali nelle lunghezze d'onda del rosso (R), del verde (G) e del blu (B).

Il motivo per questa scelta viene dalla volontà di imitare il comportamento dei recettori presenti negli occhi umani. Lo spettro della luce riflessa può essere quindi rappresentato con un vettore di 3 elementi, che rappresentano le proporzioni dei colori primari rosso, verde e blu utilizzati per produrlo. Questi valori vengono detti "Tristimolo", quindi:  $RGB_{values} = Colour * Tristimulus$

Nelle fotocamere digitali i recettori RGB vengono spesso disposti in un Filtro Bayer: al verde vengono assegnati il doppio di recettori rispetto al rosso e al blu, in modo da favorire la luminosità dell'immagine.



## 1.4 Algoritmi di Demosaicizzazione

Dato che su ogni singolo pixel viene originariamente salvato solo un colore su tre, non si può determinare a priori il colore della luce riflessa. Per questo motivo, sono stati introdotti degli algoritmi che stimano per ciascun pixel i livelli dei colori mancanti. Questi algoritmi di Demosaicizzazione vanno infatti a calcolare un valore verosimile per gli altri colori in base a quelli presenti nei pixel vicini, attraverso un processo detto di "interpolazione".

## 1.5 Risoluzione di un'Immagine e Dimensioni dei Pixel

La risoluzione è definita come la densità di pixel in una data area e viene comunemente misurata in PPI (pixel per pollice). Le dimensioni dei pixel sono le misure orizzontali e verticali di una data area misurate in pixel, e possono essere determinate moltiplicando le dimensioni dell'immagine per la risoluzione. Ad esempio:

$$8'' * 300ppi = 2400pixels, 10'' * 300ppi = 3000pixels$$

In una fotocamera digitale la risoluzione è definita come il numero di pixel diviso per l'area del sensore. Le dimensioni dei pixel vengono quindi espresse come il numero di pixel orizzontali e verticali che quindi definiscono la risoluzione dell'immagine. Alcuni preset tipici sono:

640 × 480	low end
1216 × 912	1 Megapixel
1600 × 1200	2 Megapixels
2240 × 1680	4 Megapixels
4064 × 2704	11 Megapixels

## 1.6 Dimensioni del Sensore nelle Fotocamere

Le fotocamere hanno sensori e pixel di dimensione variabile in base all'uso. In generale, in fotocamere consumer, le dimensioni dei sensori vanno dai 4 ai 16mm di diagonale, mentre in fotocamere industriali le dimensioni sono normalmente di 6mm per ottenere una risoluzione di 640 \* 480 pixel.

Anche i pixel variano in dimensione: in fotocamere consumer vanno da un minimo di 1.1micron fino a 8.4micron, mentre in fotocamere industriali vanno da 4.6 a 7µm.

Un alto numero di megapixel non rappresenta in modo preciso la qualità di una fotocamera: altri fattori rivelanti includono infatti la quantità di luce disponibile, la tolleranza al rumore e la precisione delle misurazioni. Anche se pixel più piccoli permettono di catturare dettagli più fini, necessitano di più luce e più tempo per catturare un'immagine. Questo, in condizioni di scarsa luminosità, si traduce in immagini più rumorose.

## 1.7 Telecamere a Infrarossi e Termiche

Le telecamere a infrarossi e termiche usano un approccio diverso:

- Quelle a infrarossi utilizzano luce infrarossa con onde brevi per illuminare un'area, catturando parte dell'energia riflessa dagli oggetti per generare un'immagine: l'energia riflessa varia in base alla temperatura del corpo colpito.
- Le telecamere termiche invece usano energia infrarossa di lunghezza media/grande e rilevano le differenze di calore degli oggetti da esse colpiti, che vengono quindi rappresentate nel termogramma (ovvero l'immagine generata).

## 1.8 Telecamere Neuromorfiche per Eventi

Una telecamera per eventi è un sensore che funziona in modo radicalmente diverso rispetto alle telecamere standard: misura esclusivamente il movimento in una scena. Invece di catturare un flusso sincrono di fotogrammi, una telecamera per eventi misura i cambiamenti di luminosità (chiamati eventi) per ogni pixel. Non producono fotogrammi, ma eventi asincroni che rappresentano i cambiamenti nell'intensità dei pixel e vengono misurati molto frequentemente (fino a 100 000fps). Questo tipo di telecamere offre un grande potenziale nell'ambito della robotica

e della visione artificiale in scenari difficili, come ad alte velocità o con alto dynamic range (rapporto tra parte più e meno luminosa di un'immagine). I vantaggi sono:

- alta risoluzione temporale (nell'ordine dei microsecondi)
- altissimo dynamic range ( $140dB$  vs  $60dB$ )
- bassi consumi energetici (media:  $1\text{ mW}$  invece di  $1\text{ W}$ )
- bassa latenza (nell'ordine dei microsecondi)
- assenza di motion blur (sfocatura dovuta ai movimenti rapidi)

### 1.9 Risoluzione dello Schermo

La risoluzione di uno schermo digitale è il numero di pixel in ciascuna dimensione. La dimensione con cui un'immagine appare in uno schermo dipende da due cose, ovvero la dimensione dei pixel dell'immagine e la risoluzione dello schermo. La risoluzione nativa di un display è quella rappresentata dallo schermo, anche se la maggior parte dei PC permette di passare a risoluzioni più basse. Se immagine e schermo avessero la stessa dimensione in pixel, allora ogni pixel dell'immagine occuperebbe esattamente un pixel nello schermo. Risoluzioni tipiche sono:

SD	$640 \times 480, 480p$
HD	$1280 \times 720, 720p$
Full HD	$1920 \times 1080, 1080p$
2K	$2048 \times 1080$
QHD, WQHD	$2560 \times 1440, 1440p$
UHD	$3840 \times 2160, 2160p$
4K	$4096 \times 2160, 2160p$
8K	$7680 \times 4320, 4320p$

### 1.10 Raccomandazioni per Immagini Web

Per determinare la corretta dimensione di un'immagine destinata al Web, si devono considerare esclusivamente le dimensioni in pixel, cioè l'altezza e la larghezza espresse in pixel. La risoluzione perde significato sul Web, poiché la visualizzazione dipende dal dispositivo utilizzato. L'immagine viene quindi ridimensionata in pixel alla dimensione desiderata, tenendo conto che i dispositivi potrebbero avere proporzioni diverse rispetto all'immagine originale e verranno ritagliate durante la visualizzazione sul Web. Inoltre, vi sono diverse situazioni in cui è opportuno utilizzare immagini di dimensioni diverse per varie sezioni del sito web.

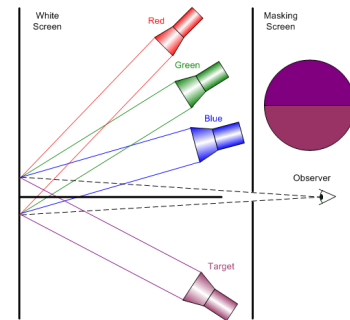
## 2 Rappresentazione di Immagini a Colori

### 2.1 Spazi Colore

Uno spazio colore è una definizione tridimensionale di un sistema di colori. Gli attributi del sistema sono mappati sugli assi delle coordinate dello spazio colore.

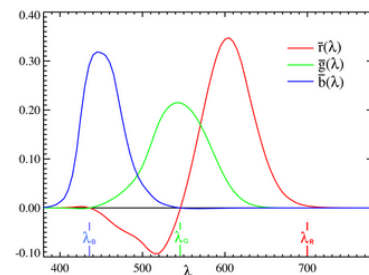
I primi esperimenti, ideati alla fine degli anni '20, miravano a caratterizzare la relazione tra gli spettri fisici e il colore percepito dagli occhi umani.

Data la necessità di avere una rappresentazione del colore indipendente dal sistema di visualizzazione, è stato definito lo spazio colore di riferimento CIE, mentre tutti gli altri spazi colore sono sottoinsiemi di quest'ultimo.



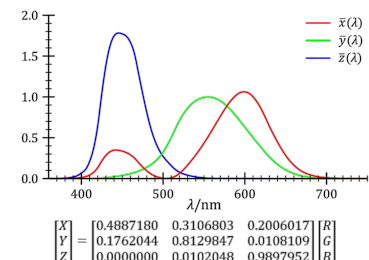
#### 2.1.1 Spazio Colore CIE RGB

Nel 1931, la CIE standardizzò le funzioni di corrispondenza dei colori RGB ottenute utilizzando i tre primari monocromatici che corrispondono ai colori rosso, verde e blu. Queste funzioni mostrano la quantità di primario necessaria per corrispondere al colore di prova a una determinata lunghezza d'onda. Lo spazio colore CIE RGB è uno tra i tanti spazi colore RGB, distinti da un particolare set di colori primari monocromatici.



#### 2.1.2 Spazio Colore CIE XYZ

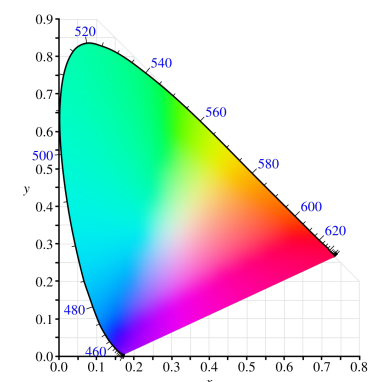
Per evitare la presenza di valori negativi nelle funzioni di corrispondenza dei colori CIE RGB, fu sviluppato lo spazio colore CIE XYZ. Lo spazio colore CIE XYZ è collegato allo spazio colore CIE RGB da una trasformazione lineare, progettata in modo che: Y fosse una misura della luminosità o luminanza di un colore; Z fosse quasi uguale alla stimolazione blu; X sia una combinazione lineare delle tre curve di risposta dei coni, scelte per essere non negative.



#### 2.1.3 Spazio Colore CIE xyY

Definendo Y come luminanza, si ottiene il risultato utile che per un determinato valore di Y, il piano XZ conterrà tutte le cromaticità possibili a quella luminanza. Normalizzando XYZ (cioè dividendo per X+Y+Z), si ottengono valori derivati denominati x, y, z. Essendo x+y+z = 1, la cromaticità di un colore può essere specificata da due parametri x e y, dei tre valori normalizzati, e può essere mappata nel diagramma cromatico CIE.

Il diagramma cromatico CIE rappresenta tutti i colori visibili dall'occhio umano con intensità costante pari a 1 (il bianco più luminoso supportato da un display a colori). Il grado di luminanza è espresso dalla coordinata Y di XYZ.



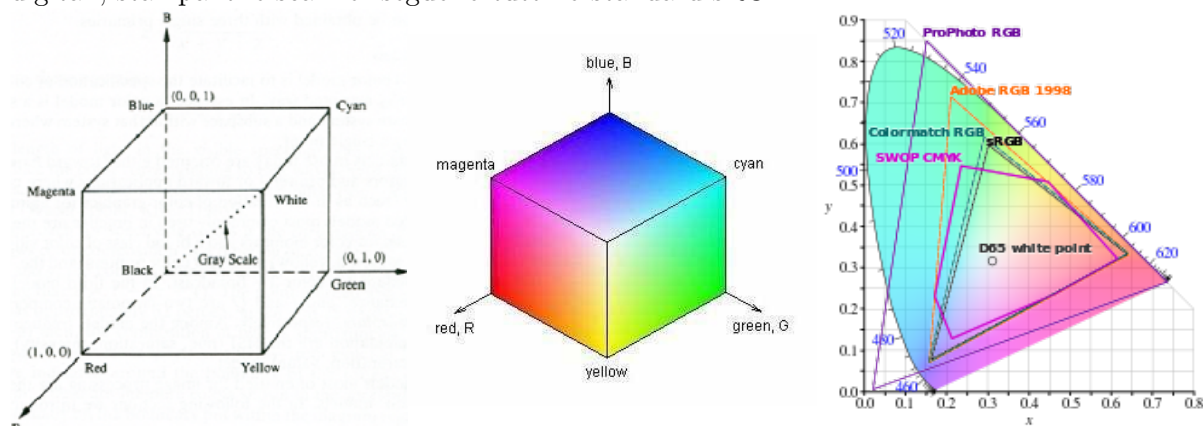
Esistono anche ulteriori spazi colori, tutti derivati matematicamente dallo spazio colore di riferimento CIE. Ognuno di essi presenta vantaggi e svantaggi particolari, che li rendono utili in applicazioni diverse.

## 2.2 Spazio Colore RGB e Derivati

Lo spazio colore CIE RGB è uno spazio colore additivo, cioè ogni colore è ottenuto dalla somma dei tre primari e può essere rappresentato come un cubo. Lo spazio colore CIE RGB è correlato a XYZ da una trasformazione lineare.

Lo spazio colore CIE RGB è uno tra i tanti spazi colore RGB, ognuno dei quali è caratterizzato da un particolare set di colori primari monocromatici (a singola lunghezza d'onda) generati dall'hardware, di fatto dipendendone.

Lo spazio colore sRGB (creato in collaborazione da HP e Microsoft nel 1996) è lo spazio colore più usato nella pratica. Utilizza gli stessi primari dello standard ITU-R BT.709, ed è lo standard di riferimento utilizzato per monitor, stampanti e su Internet. Anche schermi LCD, fotocamere digitali, stampanti e scanner seguono tutti lo standard sRGB.



## 2.3 Rappresentazione Truecolor

Il numero di uno e zero (bit) utilizzati per creare ogni pixel indica la profondità del colore che è possibile inserire nelle immagini. Il "Truecolor" è un metodo per rappresentare e memorizzare informazioni grafiche dell'immagine, che definisce 256 ( $2^8$ ) tonalità di rosso, verde e blu per ogni pixel dell'immagine digitale, risultando in  $256^3$  o 16 777 216 (circa 16,7 milioni) variazioni di colore per ogni pixel.

Vengono anche usate profondità di colore di 30, 36 o 48 bit per pixel, anche indicate come 10, 12 o 16 bit per canale RGB, risultando in un miliardo o più di colori.

Sample Length:	8								8								8							
Channel Membership:	Red								Green								Blue							
Bit Number:	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RGBAX									R. G. B. A. X															
Sample Length Notation:									8.8.8.0.0															

## 2.4 Dimensione di File Immagine e Compressione

La dimensione di un file immagine si calcola moltiplicando l'area dell'immagine per la profondità di bit:  $fileSize = (pixelDimensions * bitDepth) / 8$  (risultato in byte).

La compressione si impiega per ridurre le dimensioni dei file delle immagini al fine di facilitare lo stoccaggio, l'elaborazione e la trasmissione. Tutte le tecniche di compressione abbreviano la sequenza di codice binario di un'immagine mediante trasformazioni matematiche basate su algoritmi complessi.

### 3 Compressione Lossless - Standard di Compressione di Immagini Digitali

La compressione è il metodo per ottenere una rappresentazione codificata dell'immagine che offre un aspetto simile con un minor numero di bit. La compressione può essere senza perdita (lossless) o con perdita (lossy), che consente un più alto tasso di compressione. La scelta della tecnica di compressione dipende dall'applicazione.

#### 3.1 Algoritmi di Compressione Lossless

##### 3.1.1 Run-Length Coding

Il run-length coding è uno schema di codifica a lunghezza fissa, in cui una sequenza di simboli uguali viene codificata con un solo simbolo, preceduto da un simbolo speciale (ad esempio \$) seguito dal numero che specifica quante volte il simbolo appare consecutivamente. Ad esempio:  
Input string: *aaaaahhbbbfcdddddeegggggggggg*

Encoded string: *\$5ahh\$4bf\$5c\$6deee\$11g*

In questo caso, se il carattere si ripete meno di 4 volte, è più conveniente scriverlo normalmente invece di codificarlo. Un tempo questo tipo di codifica era usata per le immagini PCX, adesso invece viene usato solo come parti di algoritmi di compressione più complessi (es: JPEG).

##### 3.1.2 Prefix-free Coding

Il prefix-free coding è uno schema di codifica in cui ogni simbolo è rappresentato da una codeword, che non è mai il prefisso di qualsiasi altro simbolo. Ogni messaggio codificato in questo modo è quindi univocamente decifrabile, cercandolo in un codebook.

Esempio:

A: 00

B: 010

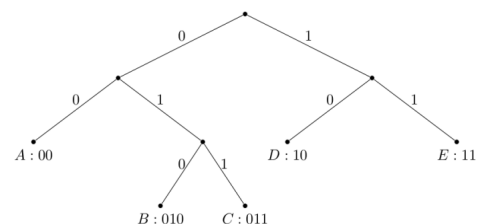
C: 011

D: 10

E: 11

Messaggio codificato: 1011011000100010

Messaggio decodificato: DECABAD



##### 3.1.3 Huffman Coding

I prefissi usati per i simboli nel prefix-free coding vengono generati tramite l'Huffman coding. Per ogni simbolo viene utilizzato un prefisso che esprime la frequenza con cui esso viene utilizzato, in modo che non sia mai un prefisso di qualsiasi altro simbolo.

Può essere generato costruendo un albero binario con nodi contenenti i simboli da codificare e la loro frequenza di occorrenza. L'albero può essere costruito come segue:

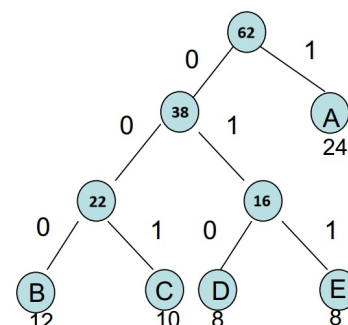
1. selezioniamo i due nodi senza genitori con le frequenze più basse
2. creiamo un nuovo nodo che sia il genitore dei due nodi a frequenza più bassa
3. assegniamo al nuovo nodo una frequenza pari alla somma delle frequenze dei suoi figli
4. ripetiamo dal passo 1 finché non rimane un solo nodo senza genitore

La codifica di ogni simbolo può essere quindi ottenuta tracciando un percorso verso il simbolo a partire dalla radice: 1 viene assegnato per ogni ramo in una direzione e 0 nell'altra. L'esecuzione è abbastanza efficiente, necessitando di  $O(n \log n)$  operazioni per costruirlo. Esempio:

Consideriamo una sequenza di 5 con date frequenze e applichiamo i passi



- A: 24  
 B: 12  
 C: 10  
 D: 8  
 E: 8
- 1: Combiniamo D e E in DE con frequenza 16
  - 2: Combiniamo B e C in BC con frequenza 22
  - 3: Combiniamo BC e DE in BCDE con frequenza 38
  - 4: Combiniamo A e BCDE in ABCDE con frequenza 62



I simboli allora saranno codificati come:

A=1 B=000 C=001 D=010 E=011

Per codificare il messaggio serviranno quindi 138 bit invece dei 186 originali.

Per decodificare i file con codifica Huffman, l'algoritmo di decodifica deve sapere quale codice sia stato usato per codificare i dati: questo può essere fatto tramite una tabella con i simboli e i loro relativi codici, oppure con l'albero. Il metodo più veloce per decodificare consiste nel leggere un bit alla volta e attraversare l'albero fino a raggiungere la foglia contenente il simbolo corrispondente, ricominciando poi dalla radice.

Nell'esempio precedente, se la sequenza in ingresso fosse "1011000010011", la sequenza decodificata sarebbe: A E B D E.

L'algoritmo di Huffman è ottimale quando i simboli dei messaggi non sono correlati fra loro e quando la distribuzione di probabilità in ingresso è nota, ma se così non fosse esistono algoritmi di compressione migliori.

Ad esempio la codifica LZW usata nella compressione delle immagini GIF, che si basa sulla frequenza di ripetizione di combinazioni di simboli, può essere più efficiente.

### 3.1.4 LZW Coding

La compressione LZW, tipicamente utilizzata nei file GIF TIFF e PDF, sostituisce le sequenze formate da 2 o più simboli con un singolo codice, aggiungendo via via le nuove stringhe di simboli ad una tabella.

Il codice LZW può essere di lunghezza arbitraria, ma deve necessariamente avere più bit di un singolo simbolo: ad esempio, per i simboli codificati con 8bit, è possibile usare codice LZW a 12bit; i codici 0/255 vengono quindi assegnati all'insieme standard dei simboli, mentre i codici 256/4095 si occuperanno delle stringhe trovate via via che l'algoritmo procede.

Esempio - codifica e decodifica LZW: stringa in ingresso "/WED/WE/WEE/WEB/WET"  
 Supponiamo di assegnare ai singoli simboli i valori 0/255.

A partire dalla prima coppia di caratteri, viene controllato se la stringa "/W" sia presente nella tabella e, dato che non è presente, le viene assegnato il codice 256.

Dopo la lettura del simbolo "E", viene controllata la sequenza "WE" e viene aggiunta alla tabella con il codice 257, stampando poi il codice per "W".

Dopo la lettura del simbolo "D", viene controllata la sequenza "ED" e viene aggiunta alla tabella con il codice 258, stampando poi il codice per "E".

Dopo la lettura del simbolo "/", viene controllata la sequenza "D/" e viene aggiunta alla tabella con il codice 259, stampando poi il codice per "D". Viene quindi letto il simbolo "W" e viene

INPUT Character	existing?	string checked	existing?	OUTPUT Code	Table	
					New code	New String
/	Y	-	N	-	-	-
W	Y	/W	N	/	256	/W
E	Y	WE	N	W	257	WE
D	Y	ED	N	E	258	ED
/	Y	D/	N	D	259	D/
W	Y	/W	Y			
E	Y	/WE	N	256	260	/WE
/	Y	E/	N	E	261	E/
W	Y	/W	Y			
E	Y	/WE	Y			
E	Y	/WEE	N	260	262	/WEE
/	Y	E/	Y			
W	Y	E/W	N	261	263	E/W
E	Y	WE	Y			
B	Y	WEB	N	257	264	WEB
/	Y	B/	N	B	265	B/
W	Y	/W	Y			
E	Y	/WE	Y			
T	Y	/WET	N	260	266	/WET
EOF				T		

controllata la stringa `"/W"`. Dato che questa corrisponde al valore 256 della tabella, viene quindi letto il nuovo carattere `"E"` e si procede a controllare la presenza della stringa `"/WE"`. Dato che questa non è presente, viene aggiunta alla tabella con il codice 260, stampando poi il codice 256 corrispondente alla sequenza `"/W"`.

Il processo continua così fino all'esaurimento della stringa e all'emissione di tutti i codici, ottenendo la stringa compressa `"/W E D 256 E 260 261 257 B 260 T"`.

INPUT		Output Character	String checked	New table entry
/	Y	/	/	
W	Y	W	/W N	256 = /W
E	Y	E	WE N	257 = WE
D	Y	D	ED N	258 = ED
256	Y	/W	D/ N	259 = D/
E	Y	E	/WE N	260 = /WE
260	Y	/WE	E/ N	261 = E/
261	Y	E/	/WEE N	262 = /WEE
257	Y	WE	E/W N	263 = E/W
B	Y	B	WEB N	264 = WEB
260	Y	/WE	B/ N	265 = B/
T	Y	T	/WET N	266 = /WET

### 3.2 GIF - Graphic Interchange Format

Il GIF è un formato a *8bit* per pixel che può usare una palette di massimo 256 colori scelti nello spazio RGB a *24bit*. Esso utilizza la compressione lossless LZW, presentando quindi immagini con bordi molto nitidi, ed è supportato da tutti i browser.

La limitazione a 256 colori lo rende inadatto alle fotografie o alle immagini con colori continui, mentre è adatto per immagini più semplici come grafici, line-art o loghi.

Un tempo molto usato sul web, è stato via via rimpiazzato dal formato PNG, anche se adesso viene utilizzato per GIFs animate, stickers...

#### 3.2.1 Animazione GIF

Il formato GIF supporta l'animazione: dopo l'immagine iniziale vengono rappresentate le parti dell'immagine che cambiano dopo ogni frame. Un timer interno conta i frame dell'immagine GIF e la fa ripartire alla fine. È anche possibile avere diverse palette di colori per ogni frame.

#### 3.2.2 Trasparenza GIF

I pixel di un'immagine GIF sono normalmente opachi, a meno che uno specifico colore sia designato come trasparente (solo quel colore). Viene usato un processo simile al *chroma-key*, dove il computer va a cancellare e sostituire i pixel di un colore, tipicamente il verde.

### 3.3 PNG - Portable Network Graphics

Il PNG è un formato lossless creato per migliorare il formato GIF, superando la sua limitazione a 256 colori, ed è supportato da tutti i browser.

Usando i colori dello spazio RGB a *24bit* è adatto al trasferimento di immagini su internet.

Non supporta però gli *image data EXIF* (informazioni varie sulle impostazioni della fotocamera, come velocità dell'otturatore, lunghezza focale, esposizione etc...), ed è per questo inadatto per scopi fotografici professionali.

### 3.3.1 Compressione DEFLATE

Il formato PNG impiega l'algoritmo di compressione lossless DEFLATE, diviso in due blocchi che utilizzano ognuno una singola modalità di compressione:

- Compressione LZ77: cerca le stringhe duplicate e le sostituisce con puntatori alla posizione della stessa sequenza trovata prima
- Huffman coding: sostituisce i simboli con nuovi simboli ponderati in base alla loro frequenza d'uso

Il formato PNG supporta sia immagini in RGB che in scala di grigi. Il numero di canali dipende anche dalla presenza di un canale "alpha", che va a specificare il grado di opacità dei pixel. Le combinazioni di canali possibili sono quindi:

- scala di grigi
- scala di grigi + alpha (0/1 indica il livello di trasparenza di ciascun pixel)
- rosso, verde e blu (RGB / truecolor)
- rosso, verde, blu e alpha

### 3.3.2 Compressione LZ77

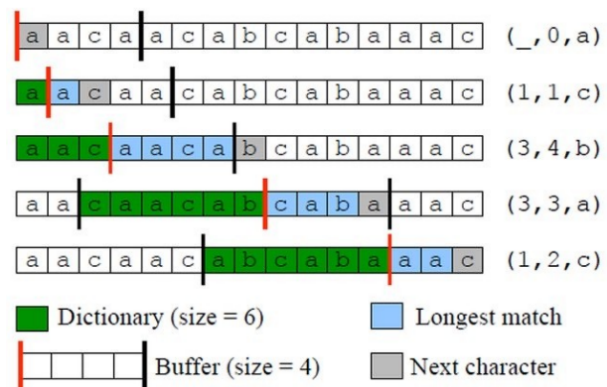
individua le sequenze di caratteri che si ripetono, utilizzando una finestra scorrevole di  $32K$ , ovvero gli ultimi 32768 caratteri.

Quando incontra la stessa sequenza di caratteri già trovata in precedenza le sostituisce con la distanza (ovvero quanto indietro nella finestra si trova la sequenza precedente) e la lunghezza (ovvero il numero di caratteri da controllare se sono identici).

Esempio:

Le finestre del dizionario e del buffer scorrono con il cursore. L'output è una tripletta (p,l,c) di valori dove:

- p = posizione rispetto al cursore della corrispondenza più lunga che inizia nel dizionario
- l = lunghezza della corrispondenza più lunga
- c = il prossimo carattere dopo la corrispondenza più lunga



## 4 Compressione Lossy - Standard di Compressione di Immagini Digitali

### 4.1 JPEG - Joint Photographic Experts Group

La compressione lossy JPEG usa il "transform coding", che sacrifica i dettagli dell'immagine per avere una dimensione minore. È il formato più usato per archiviare e trasmettere immagini sul web, perché le immagini JPEG hanno più colori del GIF e hanno dimensione minore rispetto al PNG. Non è però adatto per immagini fatte con linee, testo o icone.

Lo standard JPEG specifica sia il codec, che definisce come un'immagine debba essere trasformata in un flusso di bytes, sia il formato JFIF (JPEG File Interchange Format) utilizzato per contenere tale flusso. La compressione JPEG si basa su due aspetti, ovvero che:

- la maggior parte dei contenuti utili varia lentamente nelle immagini
- l'occhio umano è più sensibile alla perdita di dettagli in alta frequenza spaziale rispetto a quelli in bassa frequenza

Come risultato quindi si ha che un'immagine, di norma, cambia poco in un'area ridotta e che le componenti a bassa frequenza contengono più informazioni di quelle ad alta frequenza.

#### 4.1.1 Spazi Colore YCbCr e Y'CbCr

Per utilizzare lo standard JPEG si passa dallo spazio colore RGB agli spazi YCbCr o Y'CbCr, che consentono una maggiore qualità percettiva dato che assegnano ad un intero canale le informazioni sulla luminosità (canale Y o Y'). La compressione verrà poi applicata singolarmente a ciascun canale.

- Y (o Y') è la luminanza (o luma), ottenuta come combinazione di R G e B
- Cb e Cr sono la cromaticanza del blu e del rosso, ottenuti sottraendo Y (o Y') da B e R

Il luma Y' differisce dalla luminanza Y in quanto è codificato in modo non lineare attraverso una correzione di gamma. La correzione di gamma aiuta a far sì che le immagini digitali appaiano più simili a come le percepiamo con i nostri occhi, ripristinando una relazione lineare tra la luce che colpisce il sensore e la luminosità percepita.

Nello standard JFIF è specificata una particolare conversione da RGB a Y'CbCr, da eseguire per ottenere una immagine JPEG con massima compatibilità, definita da:

$$\begin{aligned} Y' &= 0.299R + 0.587G + 0.114B \\ Cb &= -0.1687R - 0.3313G + 0.5B + 128 \\ Cr &= 0.5R - 0.4287G - 0.0813B + 128 \end{aligned}$$

#### 4.1.2 Chroma Subsampling

Possono essere usati 4 diversi livelli di sottocampionamento:

- 4:4:4 = la risoluzione della cromaticanza è preservata allo stesso modo della luminanza
- 4:2:2 = nella cromaticanza si perde metà della risoluzione orizzontale, lasciando quella verticale uguale alla luminanza
- 4:1:1 = viene preservato solo  $\frac{1}{4}$  della cromaticanza orizzontale rispetto alla luminanza
- 4:2:0 = si perde metà dell'informazione sia orizzontale che verticale della cromaticanza

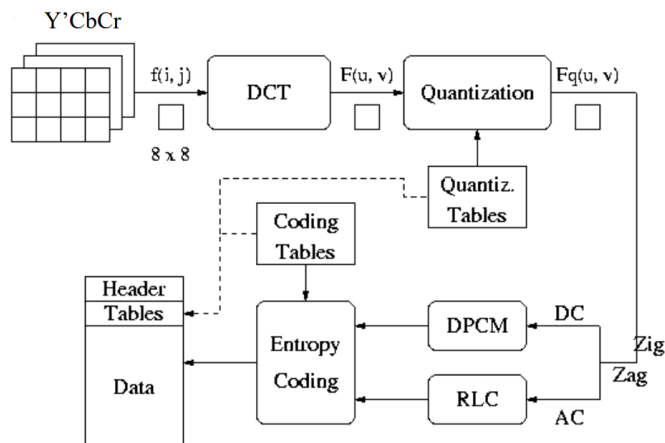
### 4.1.3 Fasi Principali dell'Algoritmo

Inizialmente l'immagine viene divisa in blocchi piccoli da 8\*8 pixel, aggiungendo righe e colonne clonate dell'ultima in caso le dimensioni non siano multiple di 8, per rimuovere le componenti ad alta frequenza. L'algoritmo viene eseguito su ogni sottoimmagine 8 \* 8, i passi sono:

1. DCT - trasformata discreta del coseno
2. Quantizzazione
3. Scansione a zig-zag
4. Codifica dell'entropia

Quest'ultima è a sua volta divisa in:

1. Codifica dei coefficienti (DPCM su componenti DC, RLE su componenti AC)
2. Codifica Huffman



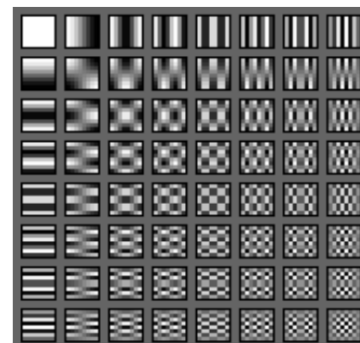
### 4.1.4 DCT - Trasformata Discreta del Coseno

Tramite la DCT passiamo dal dominio spaziale a quello della frequenza, secondo la formula:

$$F(u, v) = \frac{\Delta(u)\Delta(v)}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} f(i, j), \quad \Delta(\epsilon) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{se } \epsilon = 0 \\ 1, & \text{altrimenti} \end{cases}$$

Nella formula  $f(i, j)$  rappresenta i valori presenti nelle posizioni  $(i, j)$  del blocco 8 \* 8 dell'immagine originale, mentre  $F(u, v)$  sono i valori dei coefficienti DCT nelle posizioni  $(u, v)$  della matrice 8 \* 8 dei coefficienti trasformati.

Così facendo andiamo a mettere le componenti in più bassa frequenza (valori molto alti) nelle celle in alto a sinistra, mentre quelle in alta frequenza (valori molto piccoli) in basso a destra. Spostandosi lungo le righe aumentiamo la frequenza verticale, spostandosi lungo le colonne quella orizzontale.



Esempio:

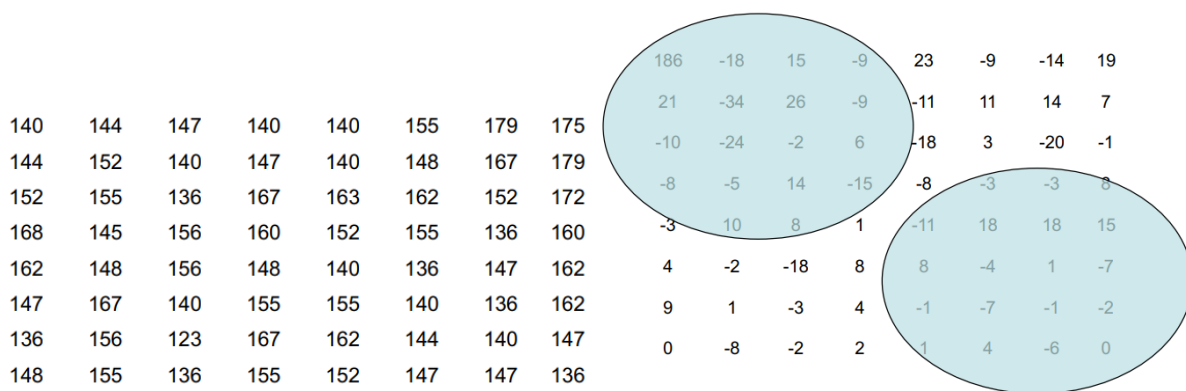


Figura 1: Valori in ingresso

Figura 2: Valori DCT di uscita

### 4.1.5 Quantizzazione

La quantizzazione viene usata per ridurre notevolmente il numero di bit per campione, ed è anche la fonte principale di perdita di informazione. Per farlo va a dividere ed arrotondare all'intero più vicino i valori della matrice ottenuta con la DCT, secondo la formula:

$$F'[u, v] = \text{round}(F[u, v]/q[u, v])$$

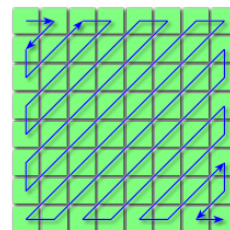
dove  $q[u, v]$  è detta matrice di quantizzazione, della quale esistono vari tipi:

- Quantizzazione uniforme: ogni coefficiente  $F[u, v]$  viene diviso per la stessa costante  $N$
- Quantizzazione non uniforme: tiene conto che l'occhio umano è più sensibile alle basse frequenze rispetto a quelle alte, ed è più sensibile alla luminanza rispetto che al colore

<i>Luminance Quantization Table q(u, v)</i>	<i>Chrominance Quantization Table q(u, v)</i>
16 11 10 16 24 40 51 61	17 18 24 47 99 99 99 99
12 12 14 19 26 58 60 55	18 21 26 66 99 99 99 99
14 13 16 24 40 57 69 56	24 26 56 99 99 99 99 99
14 17 22 29 51 87 80 62	47 66 99 99 99 99 99 99
18 22 37 56 68 109 103 77	99 99 99 99 99 99 99 99
24 35 55 64 81 104 113 92	99 99 99 99 99 99 99 99
49 64 78 87 103 121 120 101	99 99 99 99 99 99 99 99
72 92 95 98 112 100 103 99	99 99 99 99 99 99 99 99

### 4.1.6 Scansione a Zig-Zag

Come risultato della quantizzazione avremo una matrice  $8 \times 8$  con molti valori uguali a 0, e i valori diversi da 0 saranno concentrati in alto a sinistra. Trasformiamo quindi la matrice in un vettore di 64 elementi andando "a zig-zag", in modo da raggruppare i coefficienti delle basse frequenze (diversi da 0) in cima al vettore.



### 4.1.7 Codifica dei Coefficienti

La codifica dei coefficienti si divide in:

- DPCM (differential pulse code modulation) sulla componente DC: la componente DC è ampia e variegata, ma comunque con valore simile ai blocchi vicini; l'algoritmo JPEG quindi codifica la differenza tra il blocco  $8 \times 8$  precedente e quello corrente
- RLE (run length encoding) sulle componenti AC: molti dei coefficienti AC sono pari a 0, per questo è conveniente usare il RLE che conta il numero di 0 consecutivi, terminando poi il blocco con (0, 0)

Per il componente DC avremo quindi la coppia  $(SIZE)(AMPLITUDE)$  dove  $SIZE$  è il numero di bit necessari a rappresentare il valore DC e  $AMPLITUDE$  è il valore della differenza DC. Per le componenti AC invece avremo la rappresentazione  $(RUNLENGTH, SIZE)(AMPLITUDE)$  dove  $RUNLENGTH$  è il numero di 0 consecutivi,  $SIZE$  è il numero di bit necessari a rappresentare il valore e  $AMPLITUDE$  è il valore dei coefficienti diversi da 0.



## 4.2 Standard a Confronto - GIF, PNG, JPEG

I formati GIF e PNG usano entrambi una compressione senza perdita, ottenendo livelli medi di compressione:

- GIF funziona meglio su immagini con pochi colori o immagini dove un colore è dominante; non ha perdita purché l'immagine originale abbia meno di 256 colori; hanno inoltre il vantaggio di poter usare la trasparenza e l'animazione.
- PNG è l'alternativa senza perdita per quando si hanno più colori

Il formato JPEG è con perdita, ma ottiene alti livelli di compressione. La compressione funziona bene per immagini a tonalità continua, ovvero dove la variazione tra pixel adiacenti è piccola. A causa della notevole perdita di qualità, è bene usare JPEG quando la dimensione dei file è importante, ad esempio nelle pagine web.

JPEG è quindi lo standard per la fotografia digitale, dove si riescono ad ottenere file molto più piccoli dei corrispettivi PNG perdendo relativamente pochi dettagli. PNG è invece la scelta migliore se abbiamo immagini contenenti testo, line art o transizioni nette.

Alcune regole generali sono:

- GIF: immagini in bianco e nero, testo su sfondo a tinta unita, immagini con trasparenza o animate, line-art o cartoon-art fatta a computer, icone, pulsanti, immagini con un colore prevalente (> 80%)
- JPEG: fotografie scannerizzate a 16/24 bit, immagini fatte a computer con toni continui, immagini e fotografie scannerizzate, immagini grandi con molti dettagli



## 5 Video Digitali

Un video è una sequenza di frame (immagini), di norma a 24/30 fps, in modo da apparire all'occhio umano come un flusso continuo di informazione. Il video analogico è un segnale analogico che contiene luminanza e cromaticanza, ma non è più in uso a favore del video digitale, ovvero una sequenza rapida di immagini in successione. Come per le immagini, ne esistono formati compressi come il H.264 e il H.265.

I vantaggi del video digitale sono:

- può essere copiato senza perdita di qualità
- può essere manipolato o editato su computer
- registrare video digitale è molto economico

Per questi motivi si è via via affermato in ambito televisivo, nell'industria cinematografica e sul web, via via aumentando di qualità e introducendo compressioni più efficienti. Aspetti fondamentali del video digitale sono: formati di cattura, spazi colore, codifica, risoluzione, aspect ratio (proporzioni), bitrate, sampling (campionamento) e gli standard di compressione.

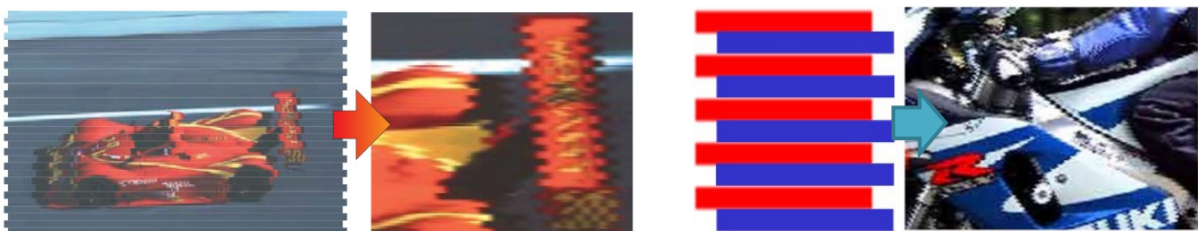
### 5.1 Formati di Cattura

Esistono due diversi formati di cattura per i video digitali: interlacciato e progressivo

- Interlacciato: le videocamere interlacciate catturano le immagini a set alternati di righe, prima vengono scansionate le righe dispari e successivamente quelle pari. Un set di righe dispari o pari viene chiamato field e una coppia di field consecutivi viene chiamata frame.
- Progressivo: le videocamere progressive scannerizzano ogni singolo frame in modo distinto, catturando tutte le linee dell'immagine nello stesso momento.

#### 5.1.1 Video Interlacciati

I video interlacciati erano stati introdotti inizialmente quando non era possibile trasmettere una grande quantità di dati, in modo che il video risultasse più fluido all'occhio umano. Questo però ha introdotto alcune inefficienze, soprattutto in video contenenti oggetti che si muovono velocemente o contenenti dettagli orizzontali, portando ad avere degli interlacing effects.



Oggi il video interlacciato è sempre meno in uso: rimane supportato da alcune videocamere digitali e, per essere riprodotto, ha bisogno di alcuni processi di deinterlacciamento.

Nel caso di un video interlacciato, la sua risoluzione può essere identificata da una  $i$  posta accanto al numero, come ad esempio: 1080i.

### 5.2 Spazi Colore

I video vengono riprodotti in RGB sui monitor, ma vengono trasmessi e conservati usando gli spazi colore YCbCr o Y'CbCr in modo da distinguere le informazioni di luminanza e cromaticanza.

### 5.3 Codifica

La luminanza e la crominanza delle immagini che compongono un video possono essere trasportate combinate in un unico canale (composite encoding) oppure tramite canali separati (component encoding).

- I video analogici possono usare sia il composite che il component encoding
- I video digitali usano esclusivamente il component color encoding

### 5.4 Risoluzione

Esistono varie risoluzioni per i video digitali, le più note sono:

- SD (Standard Definition) 675p: 720 \* 576 pixel se in 4 : 3, 1024 : 576 pixel se in 16 : 9
- HD (High Definition) 720p: 1280 \* 720 pixel
- HD 1080i/1080p (Full HD): 1920 \* 1080 pixel in modo interlacciato o progressivo
- ULTRA HD 4K: 3840 \* 2160 pixel per la tv, 4096 \* 2160 per il cinema
- ULTRA HD 8K: 7680 \* 4320 pixel per la tv, 8192 \* 4320 per il cinema

### 5.5 Aspect Ratio - Proporzioni

L'aspect ratio di un'immagine descrive le proporzioni tra la sua larghezza ed altezza. Gli standard principali sono 2:

- 4:3 (1.33:1): usato dalla creazione dei video, è usato nella standard television
- 16:9: più moderno, è lo standard per la HDTV

Ci sono anche altri formati, come quello quadrato o verticale, via via più usati perché più adatti per i dispositivi mobili.

### 5.6 Bitrate

Il bitrate è l'equivalente nei video della larghezza di banda: misura la quantità di bit inviati per unità di tempo (bits per second), più è alto e più informazioni vengono inviate:

- 16 Kbit/s per i videotelefon
- 128-364 Kbit/s per le videoconferenze con compressione video
- 1.25 Mbit/s per i CD con compressione MPEG1
- 5 Mbit/s per i DVD con compressione MPEG2
- 8-16 Mbit/s per la HDTV con compressione MPEG4
- 29.4 Mbit/s per i video in FULL HD
- 50 Mbit/s per i video in 4K ULTRA HD

Il bitrate può essere costante oppure variabile:

- Constant bitrate (BCR): mantiene lo stesso bitrate per tutto il video, ma può produrre video di qualità non uniforme dovendo comprimere di più le parti di video più complesse
- Variable bitrate (VBR): per massimizzare la qualità video e minimizzare il bitrate, vengono usati più bit nelle scene più complesse e meno in quelle meno complesse, in modo da ottenere una qualità visiva uniforme

## 6 Compressione di Video Digitali

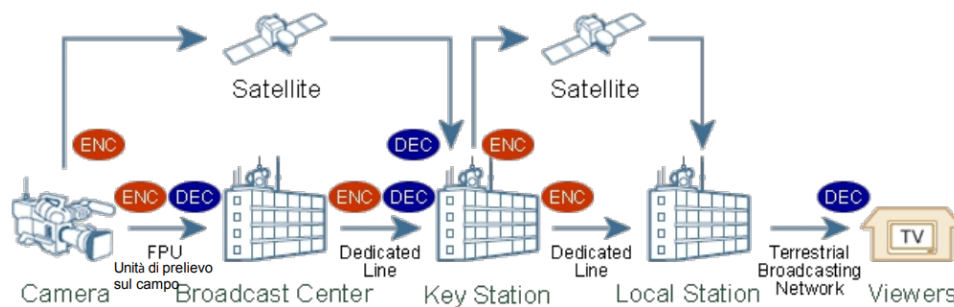
È sempre necessaria una certa quantità di compressione nei file: più alta è la risoluzione e più basso è il bitrate, più compressione sarà necessaria e più dati verranno persi.

Gli algoritmi di compressione tentano quindi di ridurre la quantità di informazioni mantenendo la qualità video, e sono in generale di tipo lossy.

Un codec (o coder/decoder) è uno strumento di codifica che processa il video e lo memorizza in un flusso di byte. I codec usano algoritmi per diminuire la dimensione dei file, decomprimendoli poi quando necessario:

- Gli algoritmi possono essere sia simmetrici che asimmetrici in termini di tempo e complessità di (de)compressione; tipicamente sono molto asimmetrici.
- La compressione è sia spaziale che temporale, cioè rimuove dati ridondanti sia spazialmente (tipo JPEG) che temporalmente.

Per la trasmissione di video in diretta i sistemi di codec video sono generalmente connessi in cascata, cercando di limitare il più possibile la degradazione della qualità.

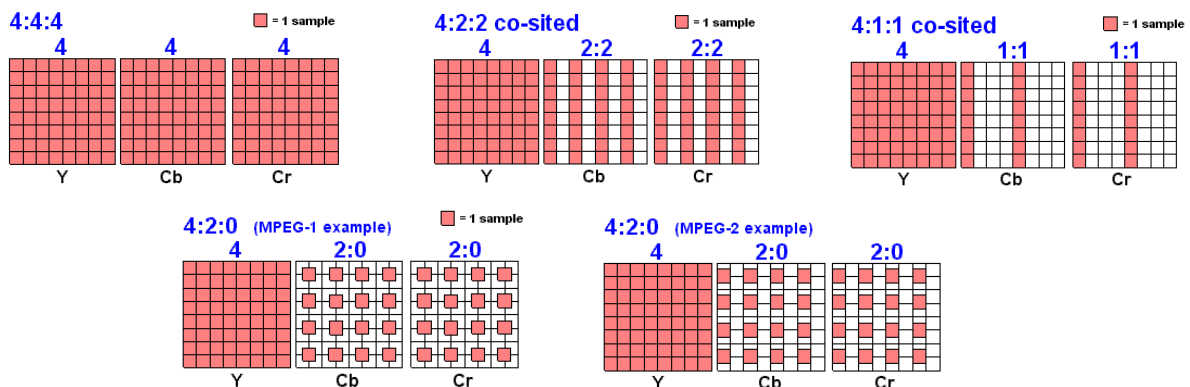


### 6.1 Sampling

Il sampling (campionamento) è il meccanismo più semplice per la compressione video; viene applicato alla luminanza e alla cromaticanza di ogni frame video.

Viene espresso con tre valori:

- $x$  = numero relativo di campioni della luminanza (Y), normalmente 4
- $y$  = numero di campioni della cromaticanza (CbCr) per le linee dispari
- $z$  = numero di campioni della cromaticanza (CbCr) per le linee pari



## 6.2 Standard di Compressione

Considerare un video come una semplice sequenza di frame, applicando le compressioni studiate per le immagini, è molto riduttivo: in questo modo non andremmo a considerare il fatto che, nel tempo, alcune parti di un video rimangono uguali o simili. Per questo motivo sono stati via via introdotti nuovi standard di compressione appositi per i video.

### 6.2.1 Compressione MPEG

I metodi di compressione MPEG mirano a ridurre la ridondanza nelle sequenze video. MPEG definisce il protocollo di flusso dei bit tra l'encoder e il decoder; l'encoder stesso e il decoder non sono specificati, ma sono lasciati da fare agli sviluppatori.

Nel corso degli anni sono stati create diverse versioni del protocollo MPEG:

- MPEG-1 H.261: usato per VHS, video-CD e audio MP3
- MPEG-2 H.262: usato per broadcasting, DTTV, TV satellitare e DVD
- MPEG-4 H.264: il più usato oggi per TV, mobile, internet, blu-ray etc...
- MPEG-H H.265: ad alta efficienza, usato attualmente per videosorveglianza

### 6.2.2 Container Video

I video sono accessibili agli utenti tramite dei container (contenitori), che hanno lo scopo di contenere tutti i file audio, video e codec compressi in un unico pacchetto. I container spesso contengono anche informazioni aggiuntive come informazioni sui capitoli dei film, metadati, sottotitoli o diverse tracce audio. I più popolari sono:

- MKV (open source)
- MP4
- AVI, WMV, ASF (Microsoft)
- MOV (Apple)
- XAVC, XAVC-S (Sony)
- FLV, SWF (Flash Video)

## 6.3 MPEG-1 H.261

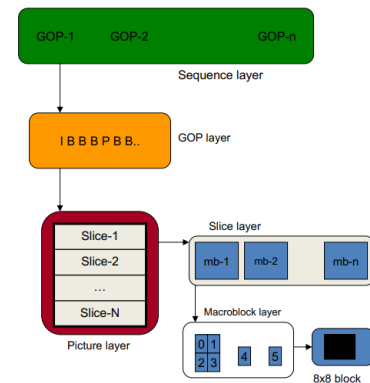
Originalmente sviluppato per supportare la qualità video VHS (videocassette) a circa 1.5Mbps, supportando solo video progressivi, si basa su due principi fondamentali:

- un video è una semplice successione di immagini ferme
- la differenza tra due immagini adiacenti è generalmente piccola

Basandosi su questi concetti fondamentali, le caratteristiche principali dell'MPEG1 sono:

- intra-frame coding: compressione sulla trasformazione del dominio, simile alla DCT, quantizzazione e RLE del JPEG,
- inter-frame coding: blocchi simili di pixel in frame adiacenti vengono rimpiazzati da dei puntatori (motion vector) riferiti a quelli di un frame ancora

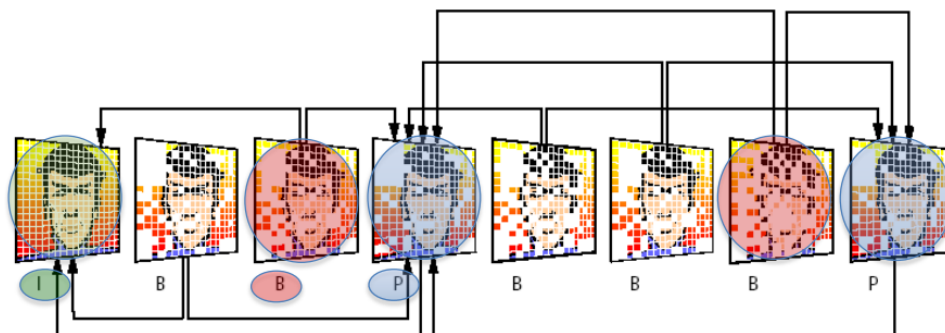
- Sequence layer: dimensioni dell'immagine, aspect-ratio, dimensione minima del buffer, matrici DCT
- GOP layer: unità per accesso random
- Picture layer: unità di codifica primaria
- Slice layer: unità di sincronizzazione
- Macroblock layer: unità di compensazione del moto
- Block layer: unità per i processi DCT



### 6.3.2 GOP - Group of Pictures

Una sequenza video viene scomposta in gruppi di immagini (GOPs), dove GOP è la distanza tra due frame chiave. I frame possono essere di diversi tipi:

- I (intra-coded): sono fotogrammi completi, identici ad un'immagine JPEG, e sono anche detti keyframes; si chiamano così perché vengono decodificati in modo indipendente dagli altri frame, minimizzando quindi gli errori e permettendo l'accesso random. La loro compressione è veloce ma produce file grandi.
- P (predictive): sono codificati con previsioni di movimento rispetto ai frame I e P precedenti, migliorando quindi la compressione sfruttando la ridondanza temporale delle informazioni; memorizzano queste differenze usando dei motion vectors.
- B (bi-directional): sono codificati con previsioni di movimento rispetto ai frame I e P precedenti e futuri; il movimento è dedotto mediante la media delle previsioni passate e future, il che li rende computazionalmente complessi e lenti da codificare.
- D: contengono i coefficienti DC, usati solo nelle anteprime



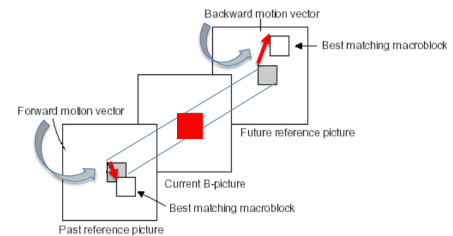
Un GOP si dice chiuso se è possibile decodificarlo completamente senza bisogno di informazioni da altri GOP precedenti/successivi.

A parità di bitrate desiderato, un GOP più grande consente di comprimere più efficientemente e mantenere più dettagli nei frame. Questo però potrebbe far propagare per troppo tempo gli errori, che potrebbero diventare troppo evidenti: aggiungere quindi dei frame I può migliorare la resilienza agli errori, pur diminuendo la compressione totale.

### 6.3.3 Macroblocks

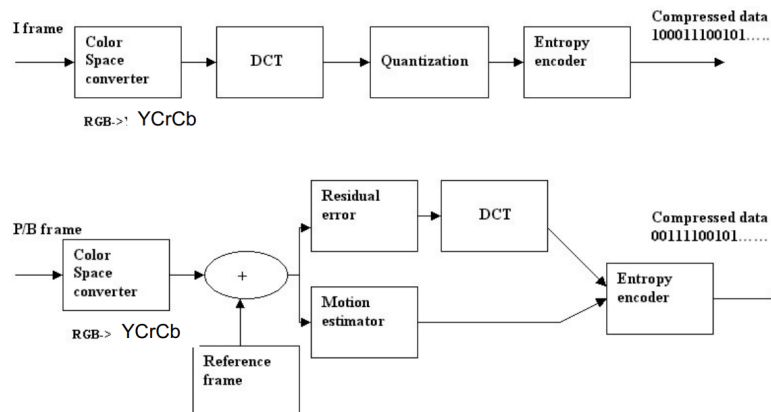
Ogni frame video è diviso in macroblocchi (la più piccola unità indipendente considerata dall'MPEG), ovvero dei set di  $16 \times 16$  pixel, che vengono usati per utilizzare i motion vector e gli error block. Ne esistono di 3 tipi:

- Macroblock I: codificati indipendentemente dagli altri macroblock
- Macroblock P: codificano il motion vector e l'error block del frame precedente (forward predicted macroblock)
- Macroblock B: codificano il motion vector e l'error block del frame precedente e successivo (forward/backward predicted macroblock)



I frame I e D, contenendo l'intera immagine, sono fatti di soli macroblock I; i frame P sono fatti di macroblock I o P; i frame B sono fatti di macroblock I, P o B.

### 6.3.4 Macroblock Encoding



- Block motion compensation: processo di sostituzione dei blocchi con un motion vector e un error block.
- Motion vector: descrive la differenza tra blocchi simili in frame adiacenti; vengono infatti specificate 2 componenti (offset orizzontale e verticale). I motion vector vengono resettati quando si trova un nuovo macroblock I.
- Error block: è un blocco ottenuto come differenza tra due motion compensated block

### 6.3.5 Motion Estimation

La motion estimation (stima del moto) è fatta attraverso algoritmi di block matching, che utilizzano diversi criteri tra loro:

- Mean Squared Error (MSE) =  $\frac{1}{N^2} \sum_{ij} (C_{ij} - R_{ij})^2$
- Sum of Squared Differences (SSD) =  $\sum_{ij} (C_{ij} - R_{ij})^2$
- Mean Absolute Error (MAE) =  $\frac{1}{N^2} \sum_{ij} |C_{ij} - R_{ij}|$
- Sum of Absolute Differences (SAD) =  $\sum_{ij} |C_{ij} - R_{ij}|$
- Matching Pel Count (MPC)

La SSD è più sensibile agli outliers (valori molto diversi) rispetto alla SAD.

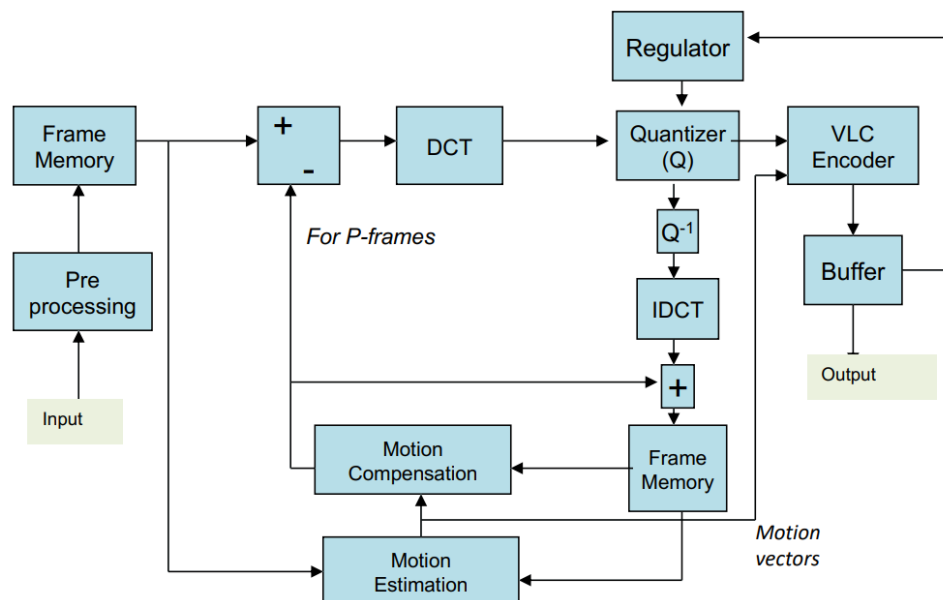
### 6.3.6 Search Matching Techniques

Vengono utilizzate per trovare la miglior corrispondenza tra un blocco di riferimento e gli altri blocchi, utilizzando le formule scritte sopra.

- Full search technique: trova sempre il minimo globale, ma controllando tutte le posizioni (computazionalmente complesso)
- Three step search (TSS, tipo ricerca binaria): complessità e performance basse
- Logarithmic search: complessità e performance basse
- One-ad-a-Time search: complessità e performance basse
- Nearest Neighbours search: complessità moderata, ma buone performance

In alcuni casi la corrispondenza è migliorata se la ricerca viene eseguita su regioni generate artificialmente interpolando i pixel della regione originaria (sub-pixel)

### 6.3.7 MPEG1 Encoder



### 6.4 MPEG-2, MPEG-4 H.264, MPEG-4 H.265

Le fondamenta su cui si basano MPEG-1 e gli MPEG successivi sono più o meno simili, basandosi sui principi fondamentali di compressione video visti in precedenza. Le principali differenze fra MPEG-1 e gli MPEG successivi includono l'implementazione di tecniche di compressione più avanzate, che supportano una maggiore qualità video, risoluzione più alta, miglior gestione del movimento e l'introduzione di nuove funzionalità come la codifica a livelli di oggetti e la trasmissione su reti a larga banda.