

03/12/2022

Quando si codifica un'immagine vengono estremamente cancellate parti di informazioni contenute in essa.

## Codifica J-Peg

Si studierà come la rappresentazione digitale di una immagine in bianco e nero, parte in cui risiede circa il 90% del valore informativo dell'immagine a colori.

B/N

Ogni immagine si rappresenta tramite una matrice di  $n$  righe e  $n$  colonne. Ogni elemento dell'immagine, che occupa un posto nella matrice nelle righe  $i$  e nelle colonne  $j$ , si chiama Picture element  
PIXEL

Quindi il numero totale di pixels (#Pixel) è il n. righe. n colonne

$$\text{DIM: } 640 \times 480 \Rightarrow \# \text{Pixels: } 640 \cdot 480$$

1080x720  $\Rightarrow$  formato standard 2019-2020

320x240  $\Rightarrow$  formato blackberry  
# Pixels: bassa risoluzione

Ogni pixel ha una risoluzione ossia gli vengono associati un numero di bit etti o dorsi maggiore riducendo e diversificando ai colori.  $\Rightarrow +$  bit per pixel + risoluzione

$N_{[\text{bit}]} \geq 16$  minimo per immagine a colore

$N_{[\text{bit}]} \geq 8$  minimo per immagine in bianco e nero (8 sfumature nero)

## Esempio

$$Q[\text{bit}] = \underbrace{320 \times 240}_{\text{* Pixels}} \times 8\text{bit} = 614,400 \text{ bit}$$

Servono 6400 bit per rappresentare in digitale un'immagine a basso risoluzione.

30 foto in bianco e nero in formato 352x288

per memorizzazione 1 foto

$$Q_{[bit]} = -352.288 \times 8 \text{ bit} = \underbrace{-811.008}_{\text{bit}}$$

$$Q[\text{bit}] = 811 \cdot 008 \cdot 30 = \underbrace{24^{\circ}330 \cdot 240}_{\text{per 30 foto b/n}} \text{ bit}$$

$$26 \cdot 330 \cdot 240 \text{ bit} = 3 \text{ Mbyte}$$

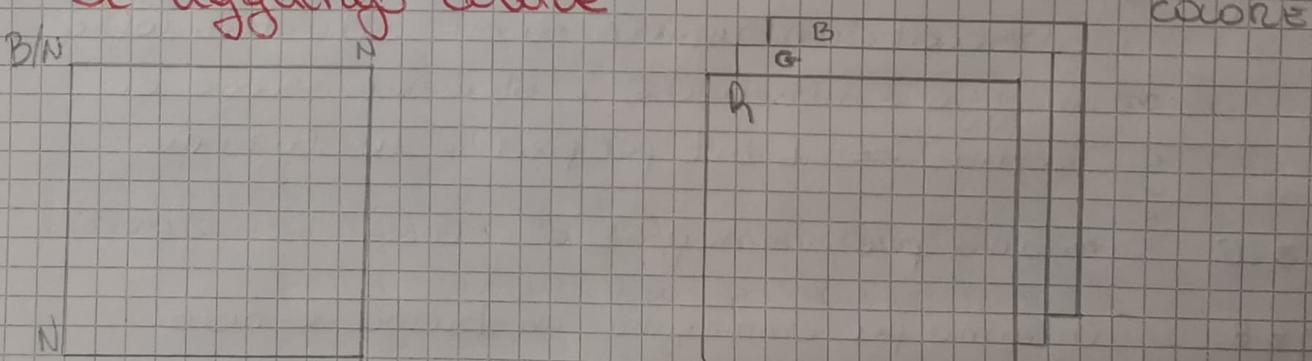
Se la volessi ~~puoi~~, trasmettere le 30 foto su Consul GSM con 9,6 kbit/s di trasferimento impiegherei:

$$T_{transm} = \frac{26^{\circ}330.240 \text{ [bit]}}{9.6 \text{ kbit/s}} \approx 42.2 \text{ min}$$

Con un cono de 64 kbit/s

$$T_{\text{transm}} = \frac{24^{\circ}330.240}{64 \text{ Kbit}} = 6.3 \text{ min}$$

Se aggiungono colori



Una foto a colori viene rappresentata con 3 matrici:  
queste 3 matrici rappresentano la quantità  
di colore presente sulla foto. Dunque si può  
pensare la foto a colori come una combinazione  
lineare di 3 colori (Verde rosso e blu):  
 $\alpha R + \beta G + \gamma B$ .

Ogni foto è dunque rappresentabile tramite uno  
spazio di colore. Questi spazi sono infiniti, ma si  
scieglie di usare lo spazio RGB, e dunque di raffigurare  
ogni foto e, in particolare, ogni sfumatura come combinazione  
di 3 colori fondamentali (rosso, verde e blu), poiché  
risulta essere la rappresentazione più vicina alle  
percezioni dei colori da parte dell'occhio umano.

Sebbene sia la migliore raffigurazione, in termini  
di trasmissione è lo schema peggiore in quanto  
le 3 matrici sono tutte equiprobabili e dunque devono  
essere trasmesse tutte con la stessa precisione ciò  
implica che, qualora una matrice non venisse trasmessa,  
perderei il 33% di informazione.

Per rendere sicure le ricezione dell'immagine  
e per rendere le matrici non equiprobabili si deve  
applicare un combo di coordinate:

V	colore 2
U	colore 1
B/N	Y

l'effetto di questo combo coordinate  
è quello di creare una una matrice  
luminanza (indice quantità bianco e nero) e  
2 matrici crominante che riportano  
i colori delle foto.

Standard  
YUV o YCbCr

Ora le matrici non sono più equiprobabili  
ma la luminante contiene il 90% di info  
quindi quelle che le mancano puente in  
trasmissione queste perdono, una matrice  
colore one perde solo il 5% di info

$$[YUV] = [A] \cdot [RGB]$$

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

**Standard JPEG (1991)** Standardizzazione contenuti multimediali creata da: Leonardo Chiarigatti

Standard di compressione definito dal gruppo Joint Photographic Expert Group.

Esistono 2 tipi di Standard:

- **Chiuso:** chi ha creato lo standard ha fissato delle regole a cui non si può deviare. Ossia se si cambia anche una piccola parte del metodo descritto allora è come se stessi cambiando Standard.

- **Aperto:** Si dice cosa bisogna fare ma il codice per eseguirlo non è specificato. In questo caso si nota che le cose che si occupano di contenuti multimediali hanno un loro codice proprietario => Questo è il caso di Jpeg

Il JPEG permetteva 2 tipi di compressioni:

- **Lossy:** compressione che scarta informazioni poco visibili all'occhio umano ossia come risultato della compressione ha una buona approssimazione dell'immagine originale.
- **Loss-Less:** compatta le foto in modo tale che l'immagine prima compressa poi decompressa sia identica all'originale.

Poiché la compressione loss-less per le foto risulta poco efficace allora lo Standard J-Peg è di tipo lossy.

L'occhio umano non percepisce allo stesso modo le informazioni su tutte le frequenze, cioè si comporta come un filtro passo basso. Ha ottima risoluzione per oggetti e bassa frequenza, perde di risoluzione alle alte frequenze, fino a non percepirla.

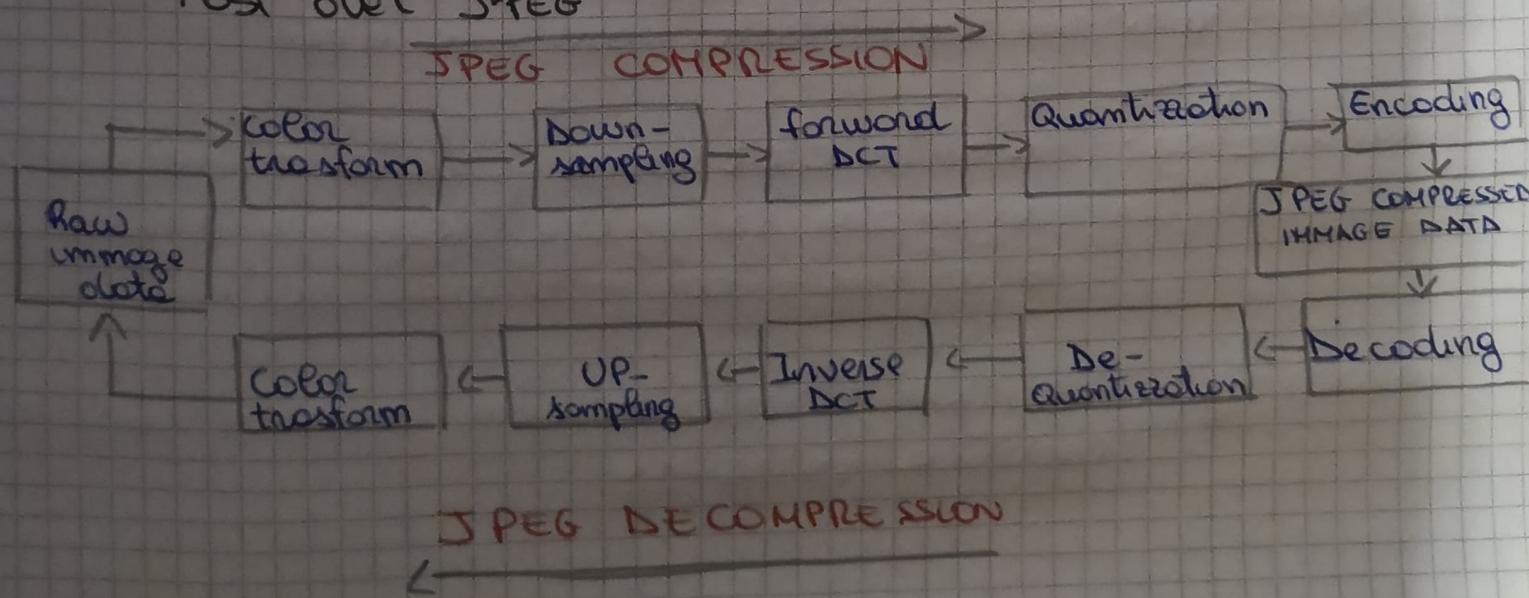
Gli algoritmi di compressione sfruttano queste caratteristiche, ossia vanno a togliere le alte frequenze

Più informazioni scarto più migliora il tasso di compressione  
JPEG: diminuisco la quantità di bit ma diminuisco anche i dettagli della foto. (fili d'erba di un campo d'erba)

La codifica JPEG DEVE:

- Essere in grado di ottenere elevati rapporti di compressione
- Poder ricostruire l'immagine con elevate fedeltà
- Poder essere parametrizzato in modo tale che bascure all'applicazione la possibilità di scegliere la fedeltà dell'immagine ricostruita e il rapporto di compressione.
- L'algoritmo non deve dipendere né dalla complessità né dal quantitativo di pixels dell'immagine
- Le sue complessità deve essere tale da poterne permettere una facile implementazione.

Fasi del JPEG



**Color transform:** trasformazione spazio di colore  
reversibile  
Da RGB a YUV

**Down-Sampling:** "sotto-compONENTE" ossia  
reversibile  
perdita di info minima  
dovuta a errori  
di approssimazione  
dunque uscita + bit per le luminanze  
ph tali matrice ha il 90% di info.  
Le oltre la sotto-compONENTE, ossia usc  
meno bit per rappresentare U e V

Introduco operazioni per togliere info

**Forward DCT:** ricercare le informazioni che l'occhio umano  
non può percepire, ossia i dettagli col  
alte frequenze, per fare ciò uso una  
trasformata di Fourier che però non  
ha una moltiplicazione per  $e^{-j2\pi ft}$ , ma  
devo usare una trasformata adatto  
alle immagini ossia la trasformata **DCT**

DCT: trasformata discreta coseno, è discreta  
perchè sto lavorando con segni (i.e. discreti) non  
analitici (cioè mi permette di non fare l'integrale  
ma di fare una sommatoria di doppie ossia  
 $\sum_{k=0}^K \sum_{l=0}^{L-1}$  su righe e colonne). È coseno perchè non  
moltiplico il valore del pixel per  $e^{-j2\pi ft}$  ma  
lo moltiplico per  $\cos(2\pi ft)$  perchè è reale e  
può con trasformate reali più e più

**Quantization:** tramite dct sono passato in frequenze  
(a brevità del programma) quindi one less in che frequenze uscendo  
le informazioni e li posso scartare.

**Encoding:** codifico le informazioni rimaste con una  
codifica entropica.

## Lettura

L'algoritmo non lavora su tutta l'immagine, altrimenti sarebbe troppo complicato e si rischierebbe di togliere troppa informazione. Quindi questo lavora successivamente su una sottomatrice  $8 \times 8$  (64 pixel). Questo permette una migliore scelta nello scarto di informazioni.

! Se non arrivo a  $8 \times 8$  aggiungo le righe/colonne per arrivare a 8, tali aggiunte (composte da 0) saranno in seguito rimosse in de-codifica.

## TRASFORMAZIONE SPAZIO-COLORE DA RGB a YUV

### SOTTO CAMPIONAMENTO

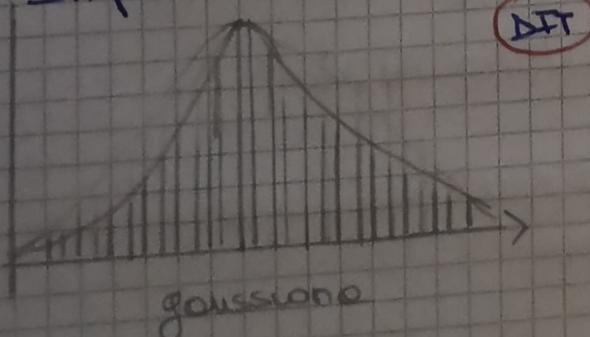
Proteggo maggiormente l'immagine Y (90% info)

### TRASFORMATA DCT

$$\sum_{l=0}^{N-1} \sum_{m=0}^{M-1} X_{lm} \cos(2\pi ft)$$

Si applica sui blocchi  $8 \times 8$ . Mi permette di passare dalla rappresentazione nel dominio spaziale a quella nel dominio delle frequenze. (sempre  $8 \times 8$ )

La DCT, a differenza delle altre trasformate, ha la caratteristica di posse nell'angolo in alto a sinistra tutte le informazioni relative alle basse frequenze e nell'angolo in basso a destra quelle sulle alte frequenze. In pratica separa la frequente.



I valori energetici sono quelli tutti concentrati nel centro, ossia tutti i pixel sono egualmente importanti, hanno lo stesso livello energetico.  
 $\Rightarrow$  trasformato discerto DFT

Dct

compatto olli bosse per queste  
i valori energeticamente + alti  
e olli medie alte es alte freq.  
massimo contenuto energetico su pochi  
valori

exp mom

## Quantizzazione

abbasse le valori dei bit nei blocchi 8x8  
→ divide i valori nelle frequenze per una matrice  
chiamata di Quantizzazione (codifica) e prendere  
per il numero intero più vicino.

La matrice di Quantizzazione ha come caratteristiche:

- dividere per valori piccoli i valori molto a sx (bosse freq)
- Il valore per cui divide aumenta all'aumentare  
delle frequenze.
- A fine quantizzazione noto che circa i 3/4 dei valori  
nel blocco 8x8 sono pari a 0 e quindi non li trasmetto  
→ valori scartati

In decodifica lo moltiplico per la matrice di  
quantizzazione ma moltiplicare  $0 \times \forall k \in R = 0$   
per cui da quantizza e smettere l'eliore  $\Rightarrow$  Inversibile

In codifica:  $X_q(K_1, K_2) = \text{NearestInteger} \left( \frac{x(K_1, K_2)}{a_{\min}(K_1, K_2)} \right)$

In decodifica:  $X(K_1, K_2) \approx Q(K_1, K_2) \cdot X_q(K_1, K_2)$

La scelta della matrice di quantizzazione è  
uno dei punti aperti dello JPEG

La matrice  $8 \times 8$  risultato della quantizzazione ha la particolarità di avere:

- Pochi valori molto a si diversi da zero
- Procedendo verso l'angolo in basso a dx tutti i valori uguali a zero.

L'informazione sopravvissuta alla quantizzazione è l'informazione minima che deve essere trasmessa per ricostruire l'immagine.

### Riordino coefficienti

Per trasmettere in modo efficiente, devo riordinare i valori partendo da quelli con valori energeticamente + alti, ossia le basse frequenze fino ad arrivare alle alte frequenze.

I valori non nulli sono inseriti in un vettore uguale con 64 colonne, questi elementi saranno per i primi posti diversi da 0 per i restanti pari a zero. In trasmissione non trasmetterò tutti gli zero, ma solo i segnali dell'info sul numero in cui quello zero si ripete.  $\Rightarrow$  CODIFICA RLE  
Run length encode

Il primo elemento di ogni blocco  $8 \times 8$  cioè quello a minor frequenza (frequenze continue f=0) ed è l'elemento con valore energetico + alto. Si codifica diversamente e non con RLE. Questo codice si chiama DPCM ossia codice differenziale, ossia poiché in genere, gli elmi a basse frequenze sono strettamente correlati dopo aver codificato il primo, il secondo lo troverà per differenza del primo.

Infine per alleggerire le mole di dati trasmessi dopo la codifica RLE e DPCM viene eseguito la codifica di HUFFMAN