

Compressione JPG di Immagini Raster

prof. Elio Toppano

Campus CompreSsione e CompreNsione (15 giugno 2021)

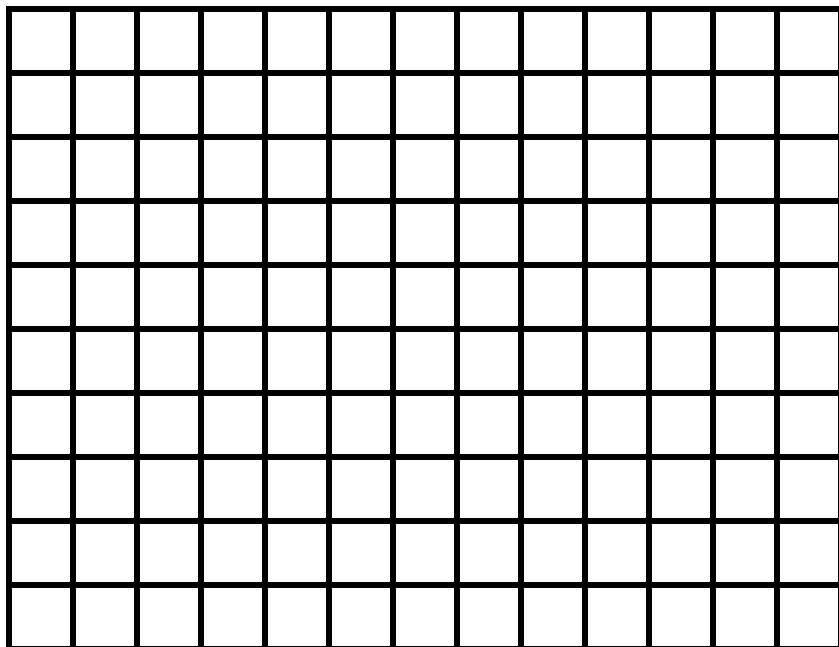
Immagini

- Una immagine è un *oggetto concreto*, un insieme di *elementi visivi* fissati su un *supporto* mediante apposite *tecniche* e pertanto disponibile ad una attività di *esplorazione visiva* e di *interpretazione* da parte di un soggetto
- **Cosa si può rappresentare in una immagine:**
 - fenomeni/entità visibili direttamente (a occhio nudo) nel mondo reale
 - fenomeni del mondo reale non visibili - perchè troppo piccoli, troppo grandi, nascosti (intestini), troppo veloci... - se non con strumenti opportuni (canocchiali, microscopi, sonde, ecc.)
 - fenomeni non visibili (es. suono, calore, campi magnetici...)
 - collezioni di dati di natura non visuale (es. torte, diagrammi radar, istogrammi)
 - costruzioni concettuali: concetti astratti, relazioni, idee
 - ecc.

Immagini raster (o bit mapped)

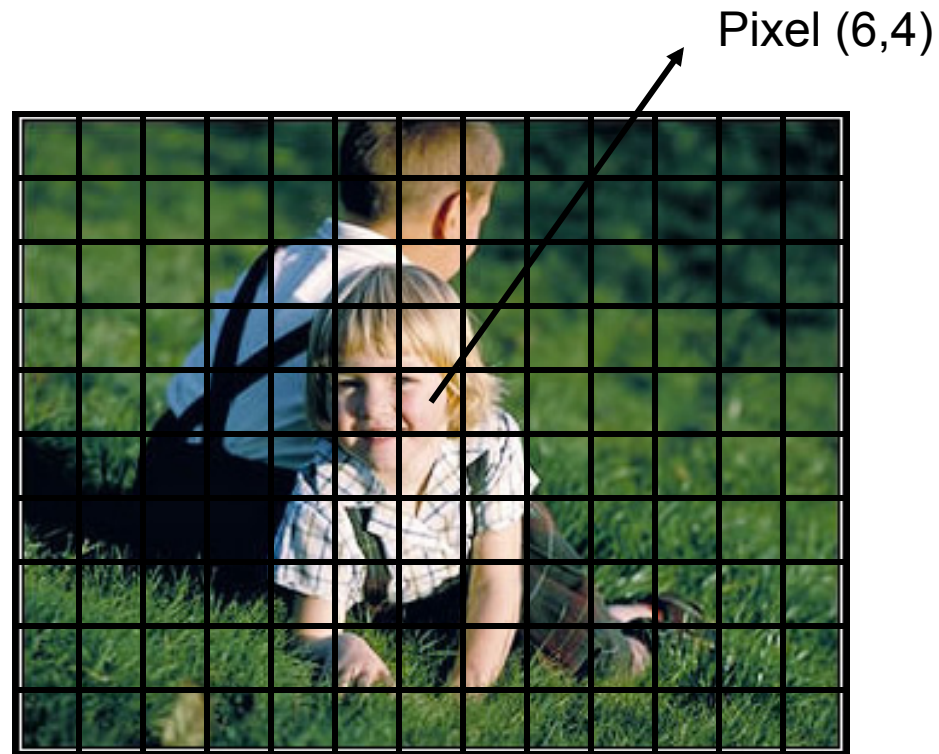
- Una immagine (analogica) viene *digitalizzata* attraverso due processi principali: il campionamento e la quantizzazione
- **Campionamento**: l'immagine viene suddivisa in una griglia di celle quadrate dette *pixel logici*. Tanto più fitta è la griglia (più numerose sono le celle), tanto migliore è la *risoluzione* spaziale della immagine (aumenta però la *dimensione* logica della immagine!)
- **Quantizzazione**: ogni pixel assume come valore il tono di grigio (o di colore) medio all'interno della cella che rappresenta; la gamma tonale della immagine viene quindi discretizzata.

Griglia [13 x 10]



Immagine





Campionamento della immagine

Spazio colore:

R G B

Pixel (6, 4) → (255,193,193)

8b 8b 8b

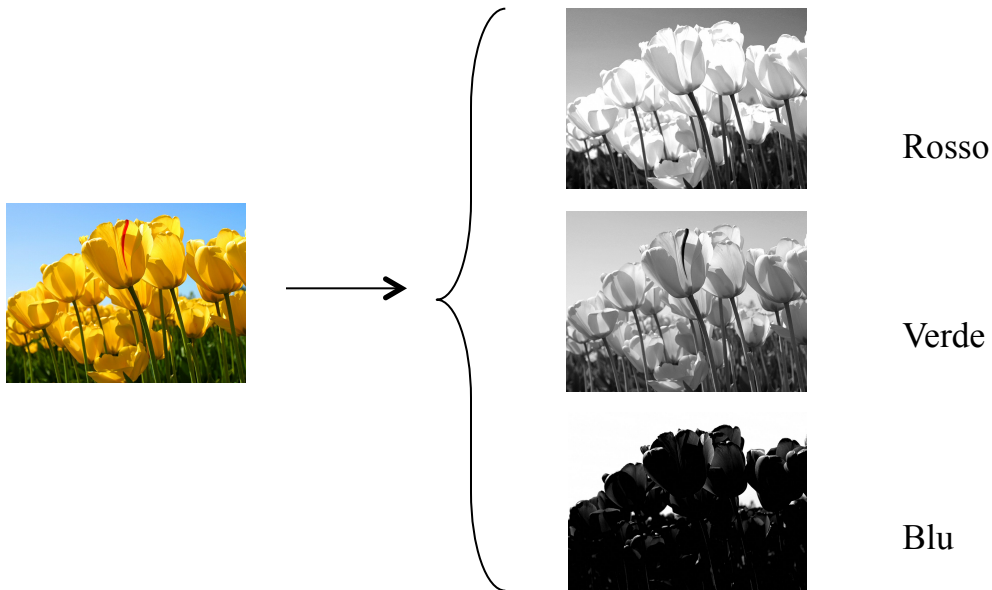
Profondità del colore: 24 bit



Quantizzazione della immagine

I canali R G B

- L'immagine digitalizzata può essere vista come la sovrapposizione di tre immagini (o canali) a tono di grigio. Ciascun canale codifica una componente specifica di colore (R, G, B)



Spazio di memoria occupato da una immagine raster (non compressa!)

- Si calcola nel seguente modo:

Spazio (bit) = *dimensione logica* (n di pixel) x *profondità di colore* (bit/pixel)

- Si ricorda che:
 - 1 byte (B) = 8 bit (b)
 - 1 Kb = 1024 bit
 - 1 Mb = 1024 x 1024 bit
 - 1 Gb = 1024 x 1024 x 1024 bit

Tempo di trasmissione di una immagine raster (non compressa!)

- Si calcola nel seguente modo:

$$\text{Tempo di trasmissione } (T_t) \text{ (s)} = \text{Spazio (b)} / Br \text{ (b/s)}$$

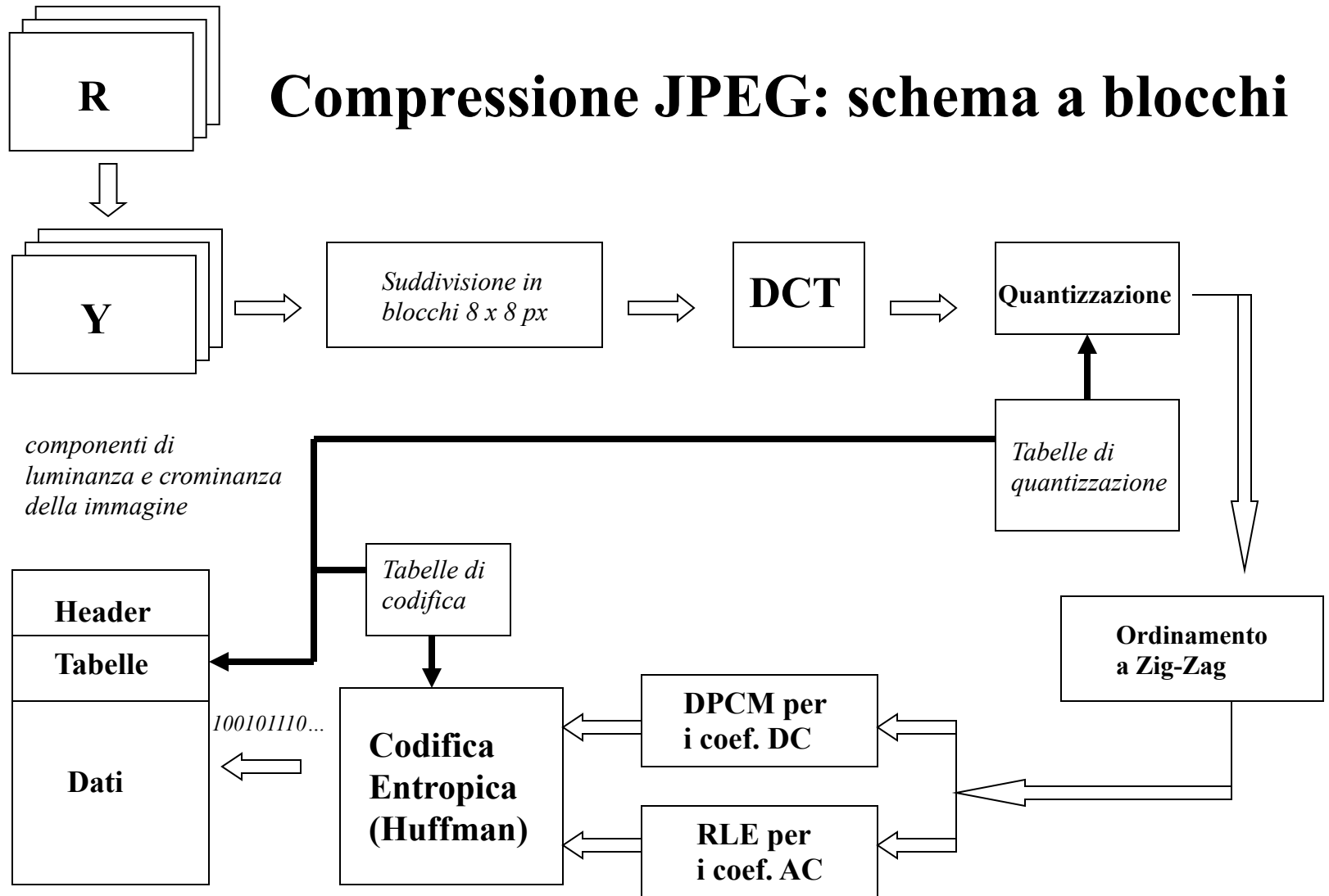
- Br : velocità di trasmissione (*bit rate*) del canale trasmissivo

Compressione JPG

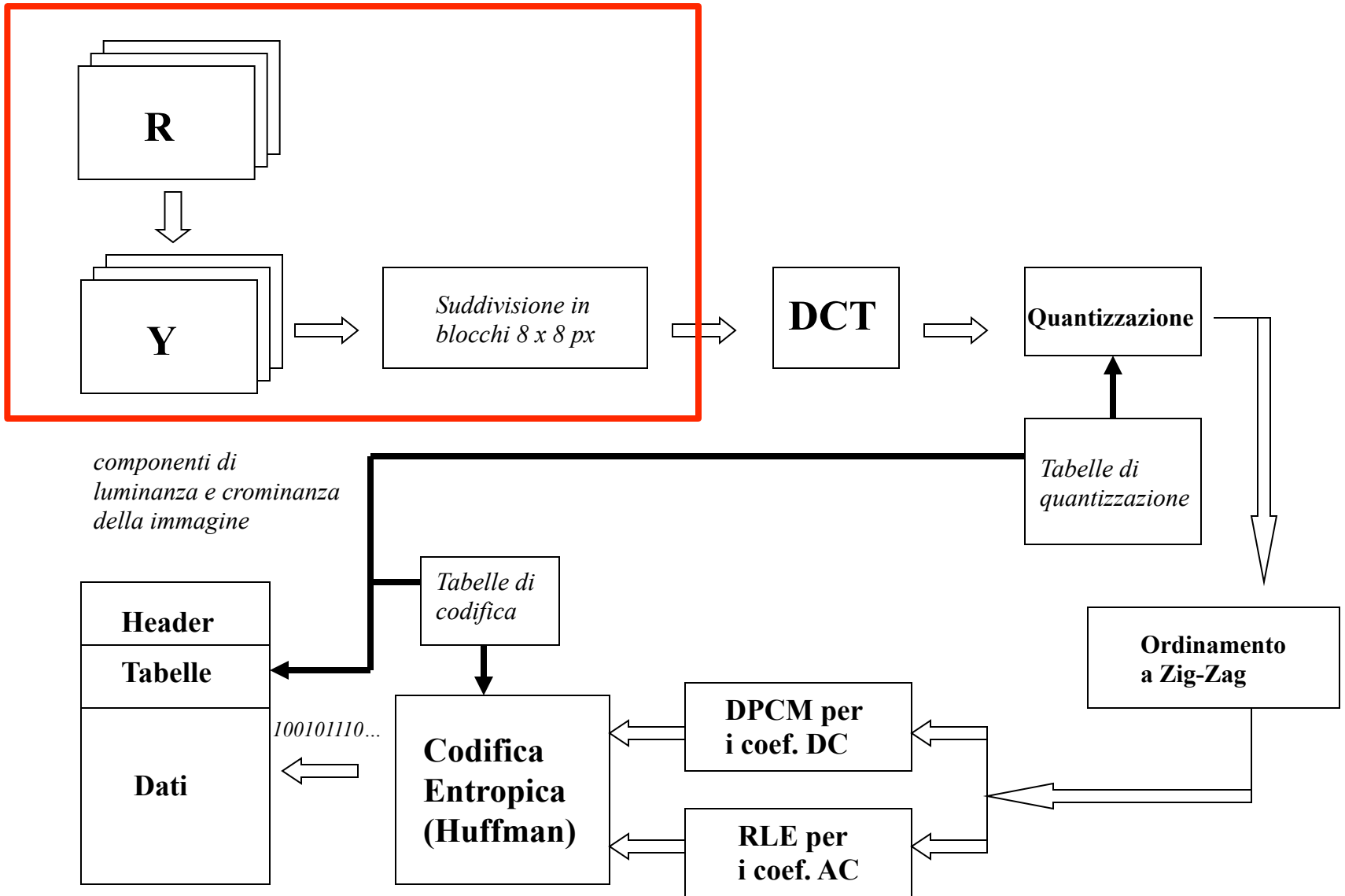
Introduzione

- JPEG è l'acronimo di *Joint Photographic Experts Group*
- Tecnica di compressione creata nel 1990 con l'intenzione di ridurre lo spazio di memoria e i tempi di trasmissione di immagini fotografiche mantenendone comunque una qualità elevata.
- La compressione (e la decompressione) vengono eseguite da un *codec* (codificatore-decodificatore)
- Esistono diverse versioni di JPEG sia lossy (con perdite) che lossless (senza perdite). Noi vedremo la versione standard che è lossy
- I file (contenitori) che contengono immagini compresse JPEG hanno tipicamente la estensione .jpg o jpeg.
- JPG utilizza la trasformata del coseno (DCT) e le codifiche di Huffman e di RLE (senza perdite) come componenti di base

Compressione JPEG: schema a blocchi



Organizzazione file .jpeg



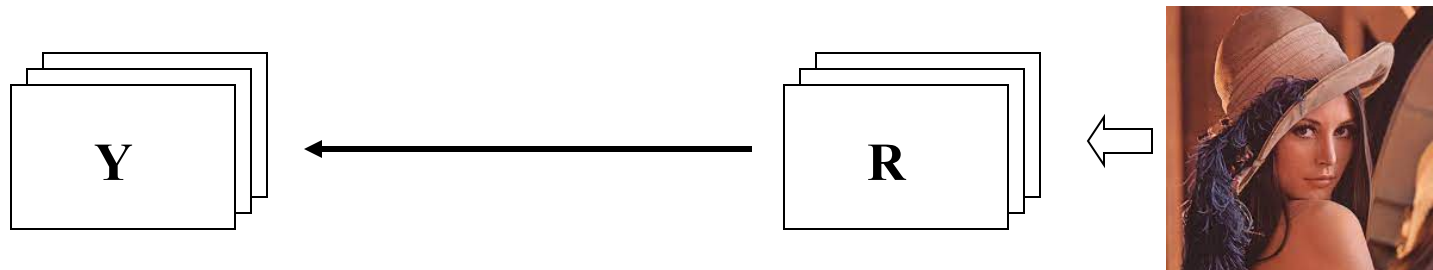
*componenti di
luminanza e crominanza
della immagine*

Organizzazione file .jpeg

Conversione da RGB a YIQ

- L'immagine RGB viene trasformata in YIQ (o YUV o YCbCr)

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.532 & 0.311 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$



Lena Sjöblom (playmate 1972)

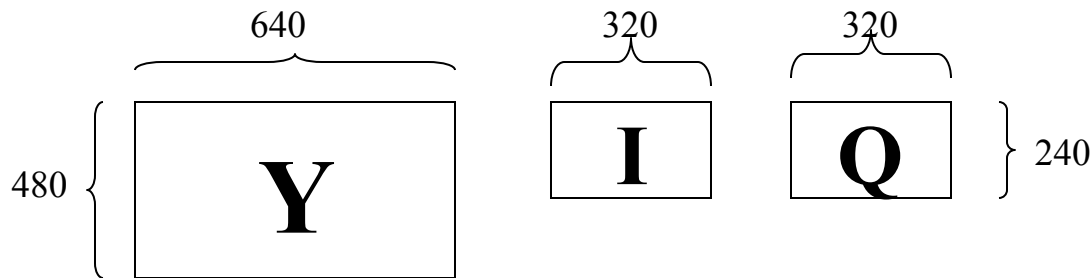
Sottocampionamento delle componenti I e Q

Esempio:

l'immagine RGB è composta da 640 x 480 px

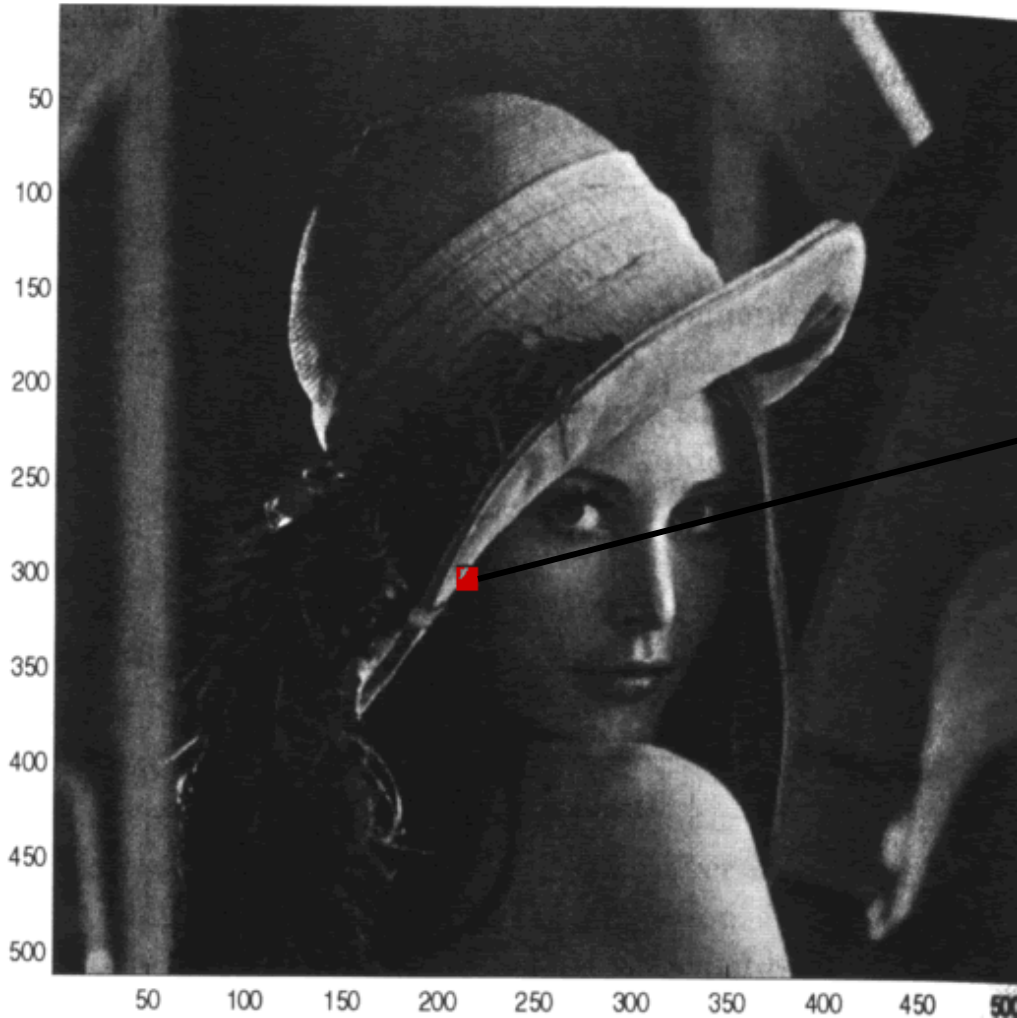
viene trasformata in

- 1) immagine Y (luminanza) di 640 x 480 px (completa)
- 2) immagine I (crominanza) di 320 x 240 px (sottocampionata)
- 3) immagine Q (crominanza) di 320 x 240 px (sottocampionata)



NOTA: l'occhio umano è più sensibile alle differenze di luminosità che di colore!

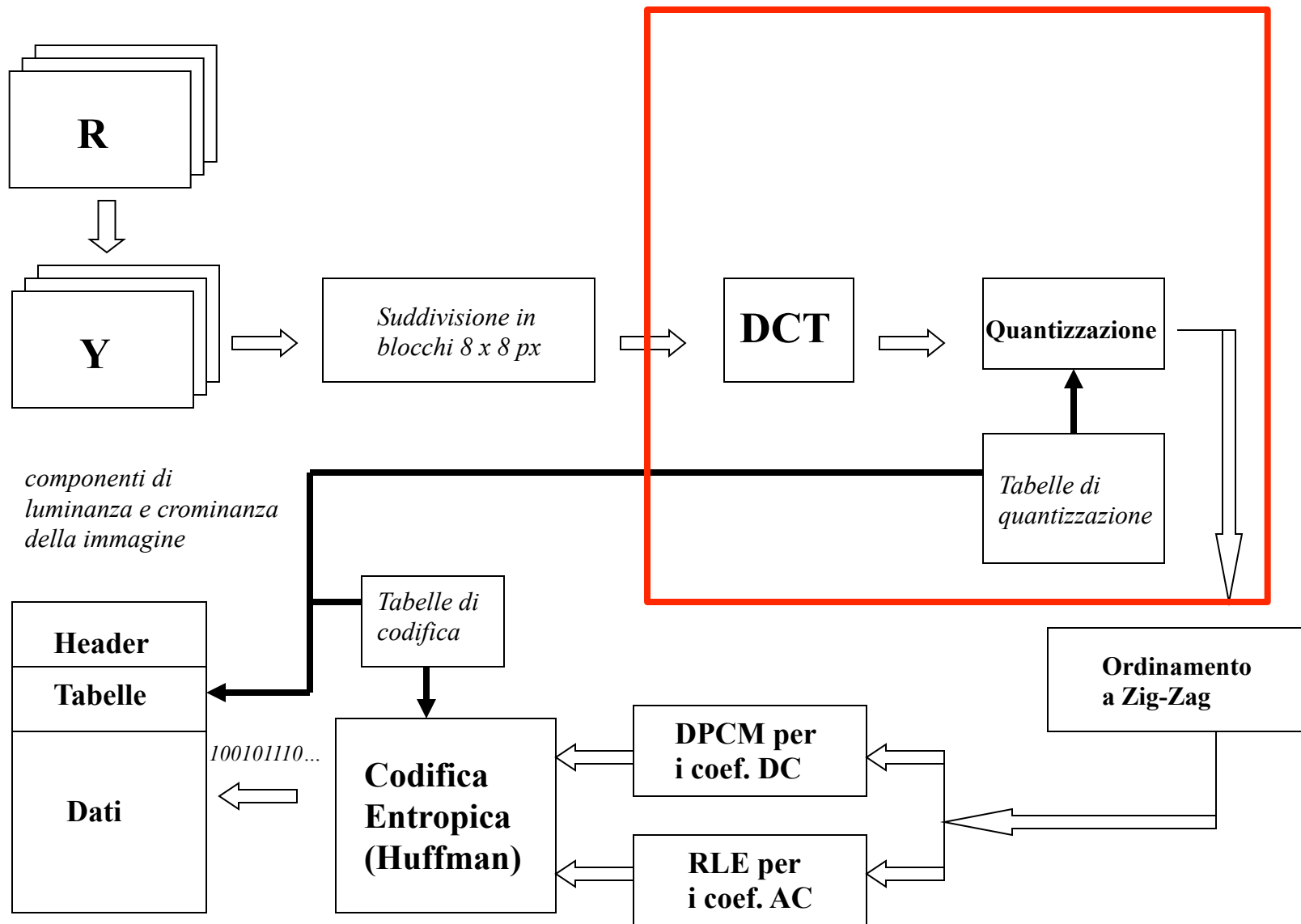
Suddivisione di Y,I,Q in blocchi



Blocco $x[m,n]$

187	188	189	202	209	175	66	41
191	186	193	209	193	98	40	39
188	187	202	202	144	53	35	37
189	195	206	172	58	47	43	45
197	204	194	106	50	48	42	45
208	204	151	50	41	41	41	53
209	179	68	42	35	36	40	47
200	117	53	41	34	38	39	63

Ciascuna immagine (Y,I,Q) viene suddivisa in **blocchi** di 8 x 8 px. Si prende questa dimensione perché in una area piccola i valori cambiano poco, sono fortemente correlati, caratteristica che rende efficiente la trasformata DCT



Organizzazione file .jpeg

Trasformata discreta del coseno (DCT)

- **Trasformata (DCT)**

$$\mathbf{X}[u,v] = C[u]C[v] / 4 \sum_m \sum_n \mathbf{x}[m,n] \cos [(2m+1)u\pi / 16] * \cos[(2n+1)v\pi/16] \quad \mathbf{0} \leq u,v \leq 7$$

dove

$$C(u) = 1/\sqrt{2} \quad \text{se } u=0$$

$$C(u) = 1 \quad \text{se } 1 \leq u \leq 7$$

- **Antitrasformata (DCT⁻¹)**

$$\mathbf{x}[m,n] = \sum_u \sum_v C[u]C[v] / 4 \mathbf{X}[u,v] \cos [(2m+1)u\pi / 16] * \cos[(2n+1)v\pi/16] \quad \mathbf{0} \leq m,n \leq 7$$

Blocco

$x[m,n]$

187	188	189	202	209	175	66	41
191	186	193	209	193	98	40	39
188	187	202	202	144	53	35	37
189	195	206	172	58	47	43	45
197	204	194	106	50	48	42	45
208	204	151	50	41	41	41	53
209	179	68	42	35	36	40	47
200	117	53	41	34	38	39	63

$$0 \leq m \leq 7$$

$$0 \leq n \leq 7$$



DCT

$X[u,v]$

915.6	451.3	25.6	-12.6	16.1	-12.3	7.9	-7.3
216.8	19.8	-228.2	-25.7	23.0	-0.1	6.4	2.0
-2.0	-77.4	-23.8	102.9	45.2	-23.7	-4.4	-5.1
30.1	2.4	19.5	28.6	-51.1	-32.5	12.3	4.5
5.1	-22.1	-2.2	-1.9	-17.4	20.8	23.2	-14.5
-0.4	-0.8	7.5	6.2	-9.6	5.7	-9.5	-19.9
5.3	-5.3	-2.4	-2.4	-3.5	-2.1	10.0	11.0
0.9	0.7	-7.7	9.3	2.7	-5.4	-6.7	2.5

$$0 \leq u \leq 7$$

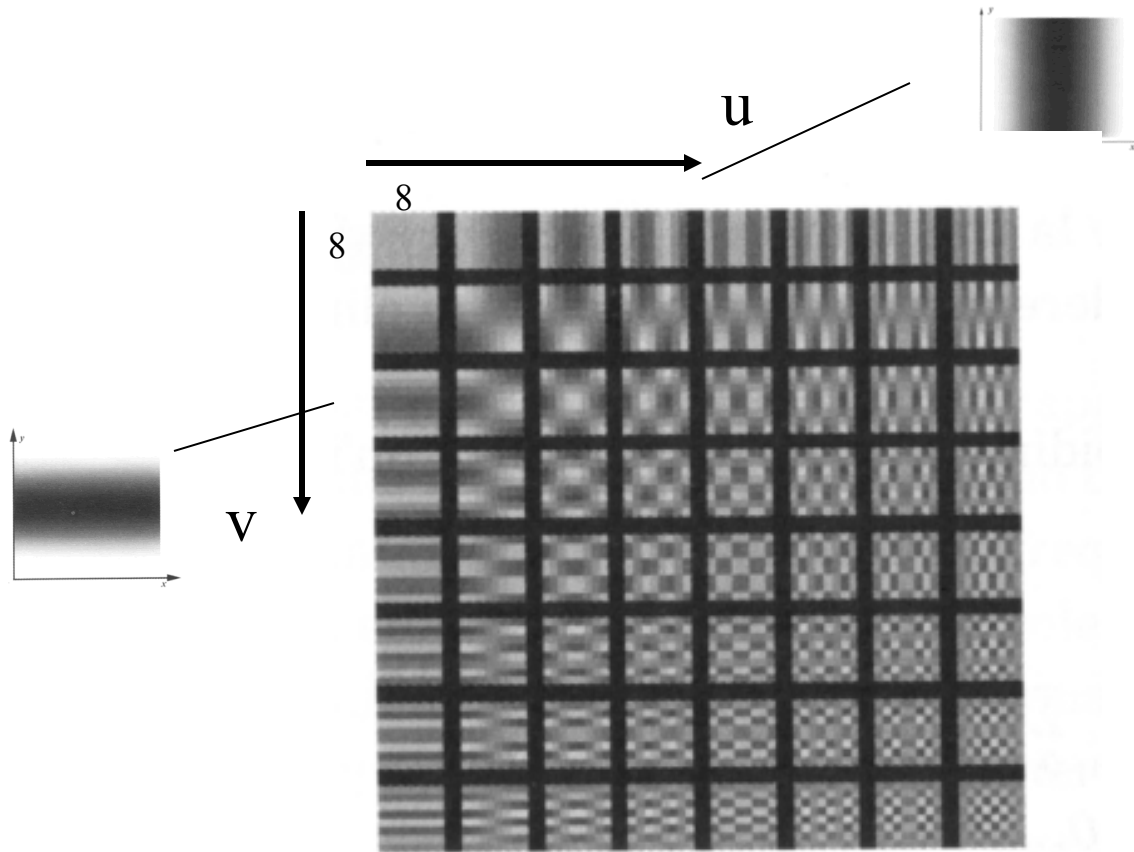
$$0 \leq v \leq 7$$

Osservazione

- Il blocco originale viene ricostruito sommando 64 funzioni base (vedi slide successiva) opportunamente pesate con i valori della trasformata discreta del coseno

$$x[m,n] = \sum_u \sum_v F_{u,v}(m,n) C[u]C[v] / 4 X[u,v] \cos [(2m+1)u\pi / 16] \cos [(2n+1)v\pi / 16]$$

The diagram illustrates the reconstruction of the original block $x[m,n]$ as a sum of 64 basis functions. The equation is: $x[m,n] = \sum_u \sum_v F_{u,v}(m,n) C[u]C[v] / 4 X[u,v] \cos [(2m+1)u\pi / 16] \cos [(2n+1)v\pi / 16]$. The terms $C[u]C[v] / 4$, $X[u,v]$, and the two cosine terms are circled in blue, while $X[u,v]$ is also circled in pink. The term $F_{u,v}(m,n)$ is positioned above the equation, with lines connecting it to the summation indices u and v .



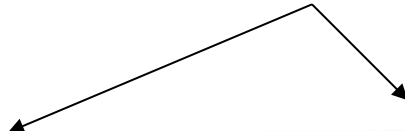
Le 64 **funzioni base** $F_{u,v}(m,n)$ della DCT:

$$C[u]C[v]/4 * \cos[(2m+1)u\pi/16] * \cos[(2n+1)v\pi/16]$$

Quantizzazione

- Si dividono i coefficienti della DCT per i corrispondenti valori specificati in una *tabella di quantizzazione* $\{q[m,n]\}$. Nota: l'occhio umano è meno sensibile alle componenti ad alta frequenza!

$$qX[m,n] = \text{round} \{X[m,n] / q[m,n]\}$$



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

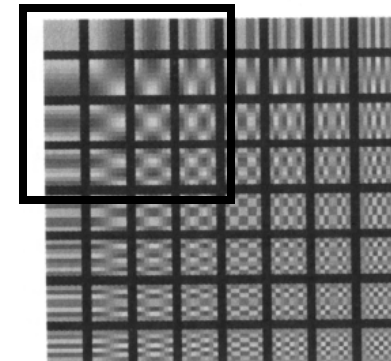
FIGURE 5 Example quantization tables for luminance (left) and chrominance (right) components provided in the informative sections of the standard.

DCT

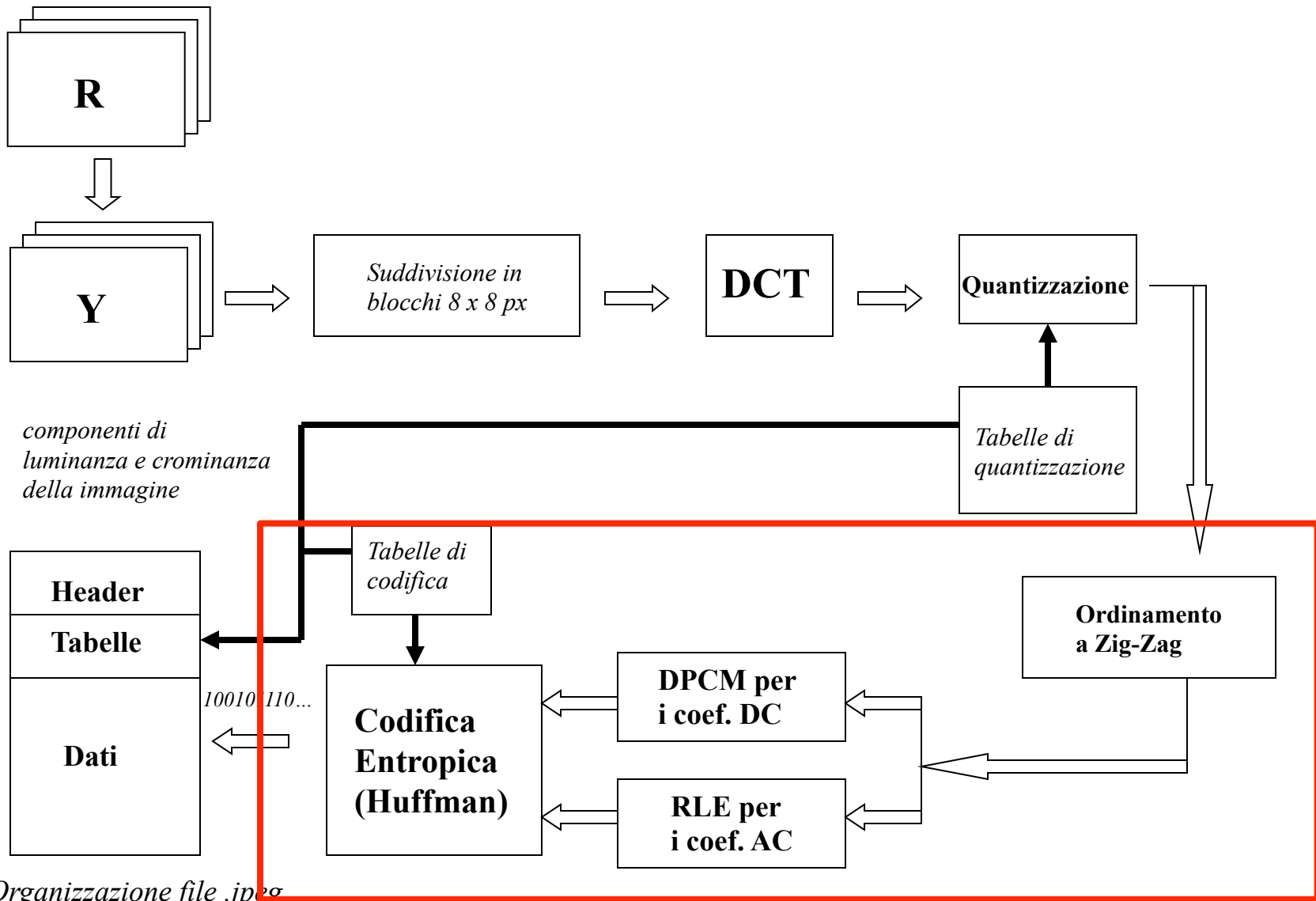
915.6	451.3	25.6	-12.6	16.1	-12.3	7.9	-7.3
216.8	19.8	-228.2	-25.7	23.0	-0.1	6.4	2.0
-2.0	-77.4	-23.8	102.9	45.2	-23.7	-4.4	-5.1
30.1	2.4	19.5	28.6	-51.1	-32.5	12.3	4.5
5.1	-22.1	-2.2	-1.9	-17.4	20.8	23.2	-14.5
-0.4	-0.8	7.5	6.2	-9.6	5.7	-9.5	-19.9
5.3	-5.3	-2.4	-2.4	-3.5	-2.1	10.0	11.0
0.9	0.7	-7.7	9.3	2.7	-5.4	-6.7	2.5



57	41	2	0	0	0	0	0
18	1	-16	-1	0	0	0	0
0	-5	-1	4	1	0	0	0
2	0	0	0	-1	0	0	0
0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

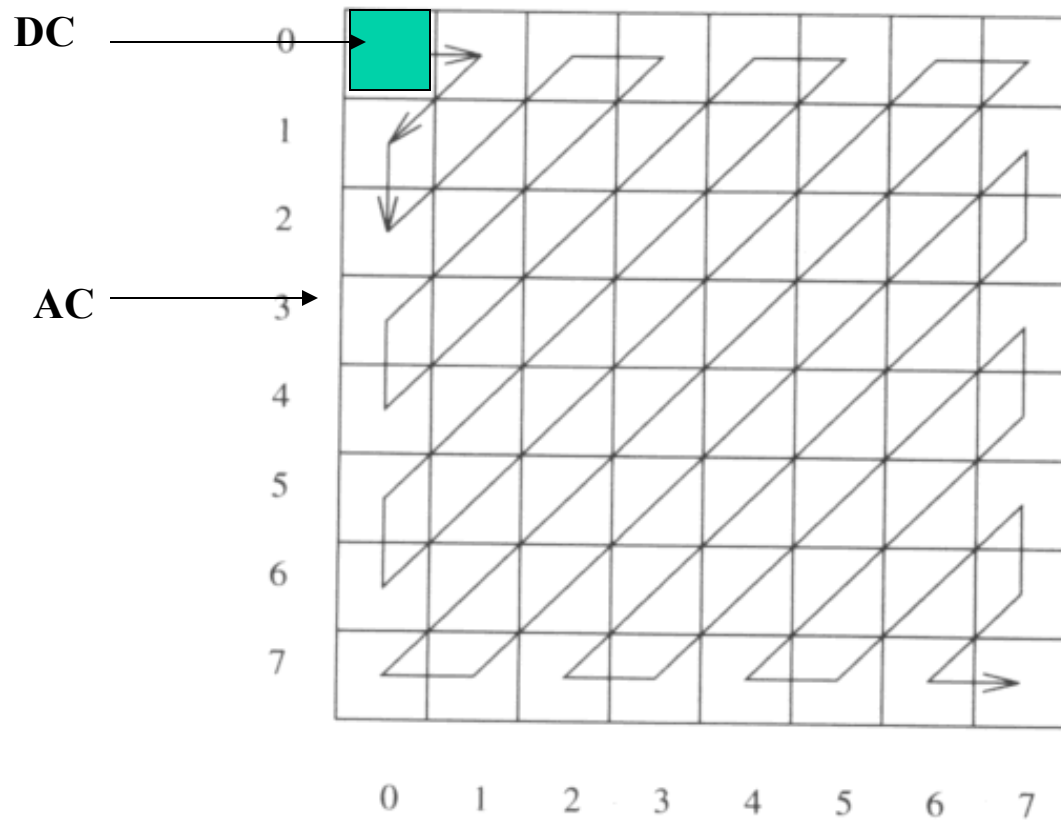


NOTA: l'occhio umano è meno sensibile alle variazioni rapide di colore/luminosità!



Organizzazione file .jpeg

Rappresentazione a Zig-Zag



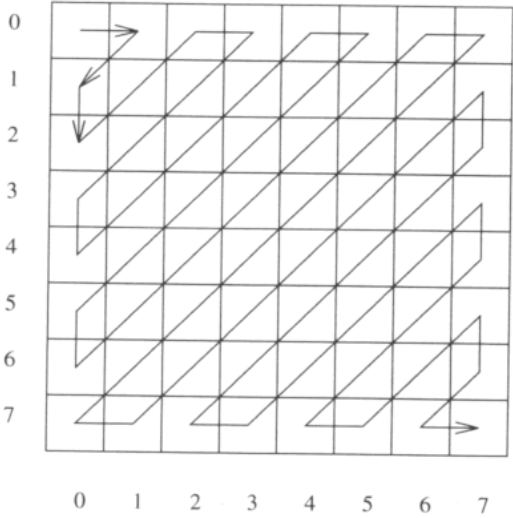
57	41	2	0	0	0	0	0
18	1	-16	-1	0	0	0	0
0	-5	-1	4	1	0	0	0
2	0	0	0	-1	0	0	0
0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Ordinamento
a Zig-Zag

57 41 18 0 1 2 0 -16 -5 ...

↑ DC

↑ AC

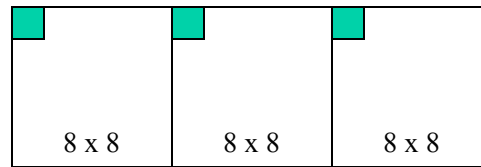


Codifica dei coefficienti DC

- Si calcola la differenza tra DCi

DC i-1 ↔ DC i ↔ DC i+1

- $\delta = qXi[0,0] - qXi-1[0,0]$



blocchi adiacenti

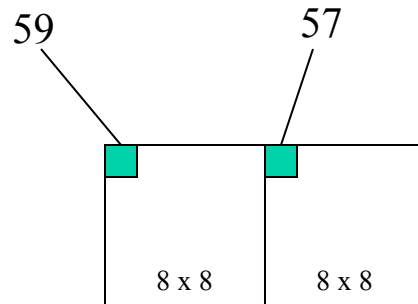
- Ci sono 12 categorie di ampiezza per δ . La categoria k-esima specifica 2^K possibili “ampiezze”

K=0	0
K=1	-1, 1
K=2	-3, -2, 2, 3
K=3	-7..-4, 4..7
....	
K= i	$-2^i + 1 .. -2^{i-1}, 2^{i-1} .. 2^i - 1$
....	

Nota. K indica il numero di bit che servono per codificare i valori della categoria

- La differenza δ viene associata ad un simbolo (*categoria, ampiezza*) e **codificata mediante Huffman**

Esempio



$$\delta = 57 - 59 = -2$$

appartiene alla categoria $K=2$: -3 -2 2 3

0 1 2 3

δ viene codificato con il simbolo [2, 1]

categoria

valore (posizione interna a cat)

in binario: 011

01

Tabella di Huffman per
i coefficienti DC
(standard)

Categoria	Lunghezza Codice	Parola di codice
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

Codifica dei coefficienti AC

- Ogni coefficiente (diverso da zero) viene rappresentato da un simbolo composto da tre informazioni

1) numero di 0 consecutivi che precedono il coefficiente

2) categoria di appartenenza (simile a DC)

3) ampiezza del coefficiente all' interno della categoria

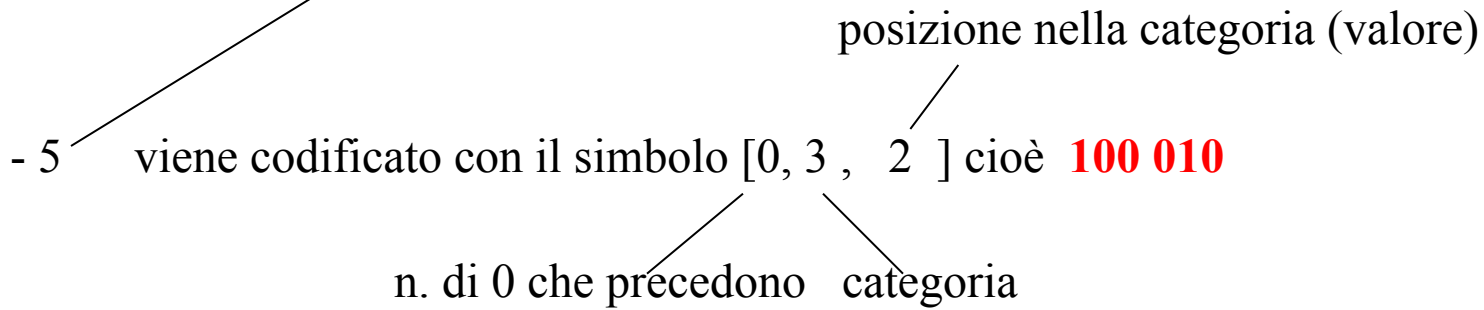
- I simboli vengono **codificati mediante Huffman**

Nota: se dopo un coefficiente AC ci sono solo zeri si usa il simbolo (0/0) che denota il fine blocco (EOB EndOfBlock)

Esempio

Sviluppo a zig zag dei coefficienti AC:

41 18 0 1 2 0 -16 **-5** 2 0 0 -1 -1 0 0 0 4 0 -1 0 0 0 0 0 1

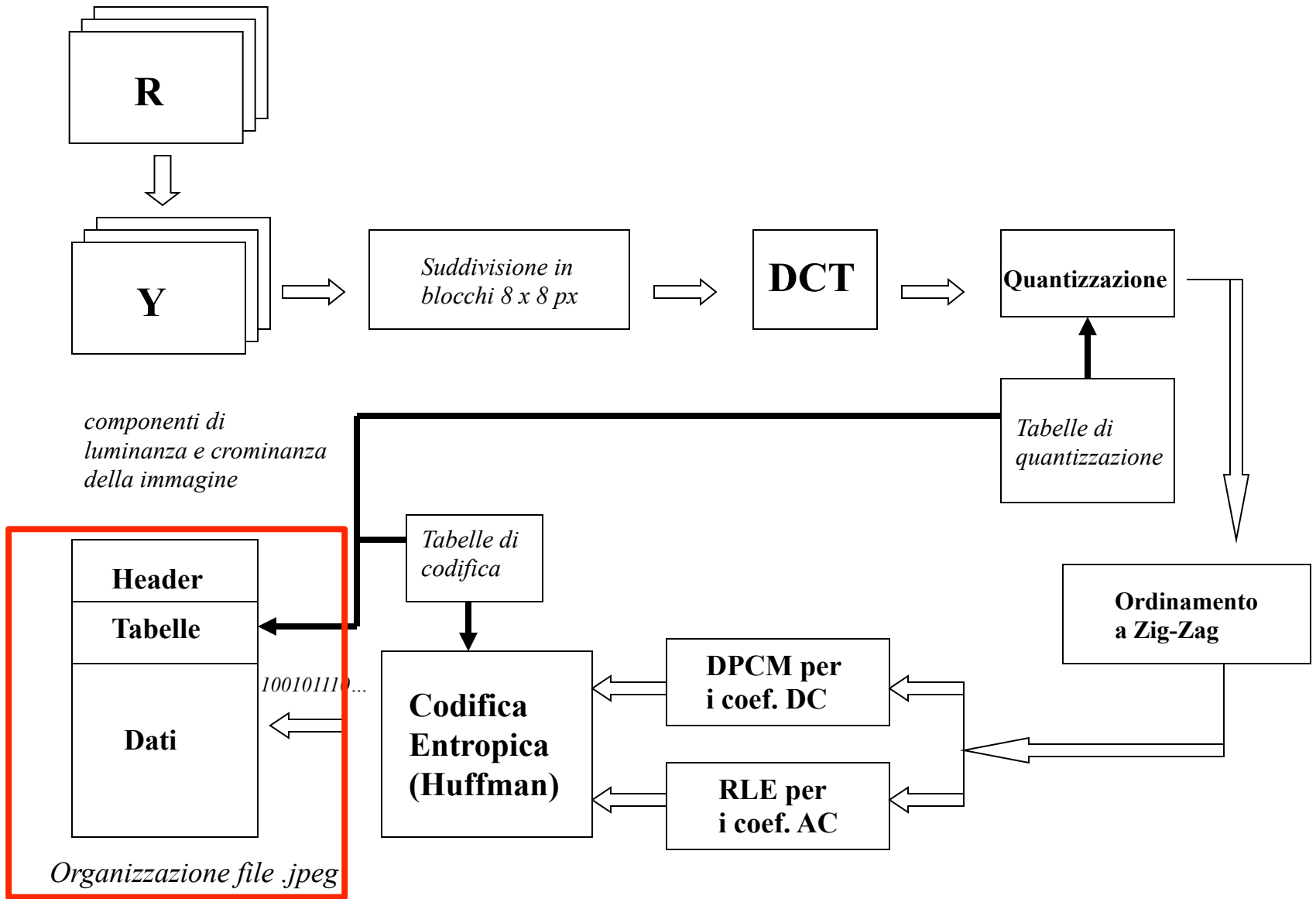


categoria 3 : -7 -6 -5 -4, +4 +5 +6 +7
posizione : 0 1 2 3 4 5 6 7

Categoria	Valori
0	0
1	-1,+1
2	-3 -2, +2 +3
3	-7 -6 -5 -4, +4 +5 +6 +7
4	-15 .. -8, +8 .. +15
5	-31 ... -16, +16 .. +31
6	...
7	...
8	...
9	...
10	-1023 ... -512, +512 ..+1023

Corsa zeri/Categoria	Parola di codice
0/0	1010
0/1	00
0/2	01
0/3	100
0/4	1011
...	...
...	...
F/A	1111111111111110

Tabella di Huffman per
i coefficienti AC (standard)



Decompressione

- Si eseguono le operazioni viste all'incontrario:
 - Si estraggono i coefficienti AC e DC e si ricostruiscono le sequenze a zig-zag e i blocchi
 - Si ricostruiscono i coefficienti della DCT (approssimati)
 - Si applica la trasformazione inversa DCT^{-1} e si ottengono i blocchi corrispondenti a Y^* , I^* , Q^*
 - Si ricostruisce l'immagine $R^*G^*B^*$ (compressa)...

Ricostruzione della immagine JPEG

- *Baseline*

l'immagine viene ricostruita sullo schermo una riga dopo l'altra

- *Progressiva*

l'immagine viene ricostruita a strati. Viene dapprima visualizzata l'intera immagine ma con una risoluzione bassa. Nella successiva/e passata/e viene aggiunto alla immagine un altro "layer" di dati in modo da migliorarne la qualità. Richiede che i valori compressi siano riordinati in modo opportuno.

Va bene per web!

Conclusione (1)

- JPEG sfrutta le caratteristiche della percezione umana per eliminare informazioni non importanti:
 - l'occhio umano è più sensibile alle differenze di luminosità che di colore (allora si usa lo spazio colore YIQ e si sottocampiona le componenti cromatiche I e Q)
 - l'occhio umano è meno sensibile alle variazioni rapide di luminosità o di colore (allora si eliminano le componenti ad alta frequenza nella trasformata del coseno con la quantizzazione)
- Si utilizzano le tecniche RLE e Huffman per comprimere (senza perdite) l'informazione rimasta. In particolare per i coefficienti DC si usa una *codifica differenziale* (si codificano le differenze tra i valori non i valori originali)
- Le perdite di informazione si hanno nel sottocampionamento delle componenti I e Q e durante la quantizzazione (eliminazione delle alte frequenze)

Conclusione (2)

- L'uso di una tecnica di compressione piuttosto che un'altra dipende da diversi fattori:
 - il grado di compressione voluto (rapporto di compressione: es. 46:1)
 - i tempi necessari per la compressione e la decompressione (ritardo di codifica)
 - la complessità del metodo usato (es. numero di operazioni, spazio di memoria utilizzato, ecc.)
 - il tipo di immagine e l'uso previsto.



2.6:1



15:1



23:1



46:1



Qualità 10

1.443 byte

Qualità 30

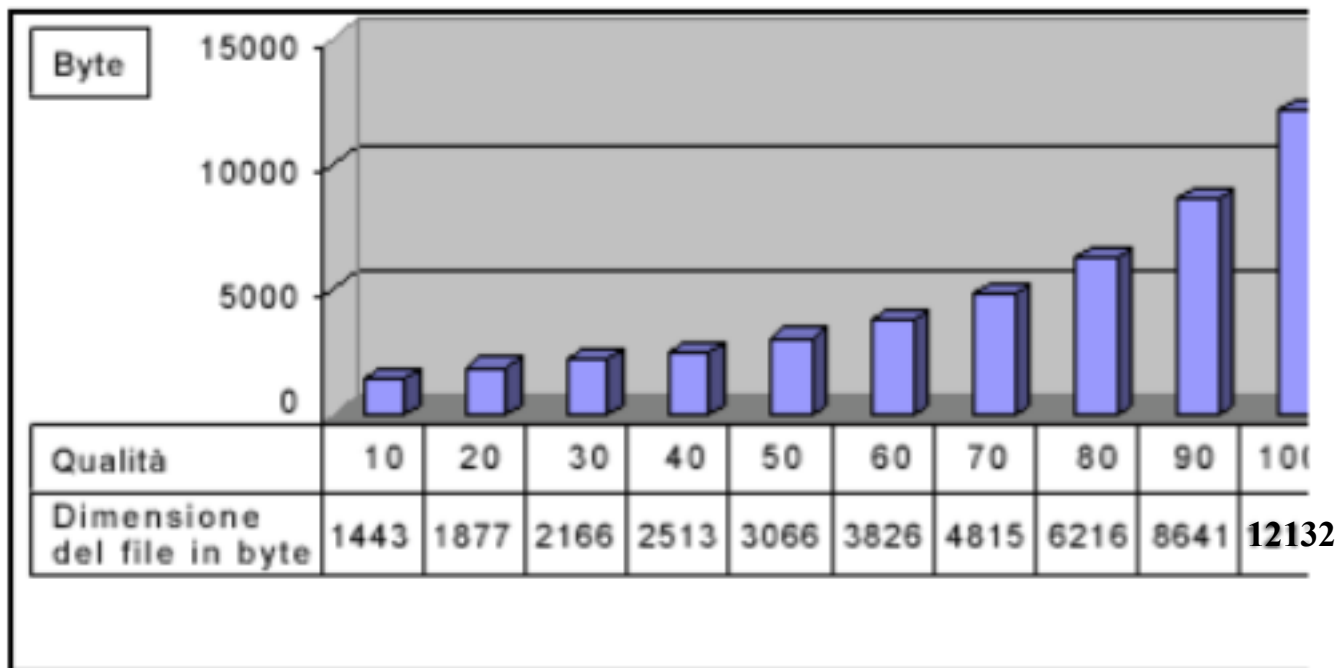
2.166 byte

Qualità 60

3.826 byte

Qualità 80

6.216 byte



Conclusione (3)

- La compressione JPG è adatta a immagini con sfumature di colore. Se applicata a immagini con molti dettagli e variazioni rapide del colore può produrre artefatti che ne diminuiscono la qualità. Bisogna chiedersi se l'eliminazione della informazione sul colore e sulle variazioni rapide può avere effetti negativi sull'uso/pratica prevista (es. immagini destinate a diagnosi medica). Non va sottovalutata l'esperienza dell'utente (vedi immagini di opere d'arte in un museo virtuale).
- La compressione JPG viene utilizzata anche per la compressione video (MPEG1 e MPEG2). In questo caso si frutta la ridondanza temporale (i frame consecutivi sono molto simili tra di loro). Alcuni frame del video (I, chiave) sono compressi alla jpeg (*compressione intraframe*). Altri frame (P, B) sono compressi utilizzando una codifica differenziale (*compressione interframe*): si comprime alla JPG la differenza tra il frame da comprimere e il frame precedente o successivo (di tipo I, P).
- L'Alliance for Open Media (include Netflix, Google, Microsoft e altri) sta pensando di sostituire JPG e HEIF (High Efficiency Image Format introdotto da Apple con iOS11) con uno standard più performante (vedremo nei prossimi anni!)

Grazie per l'attenzione!