unito.it
L'UNIVERSITÀ DI TORINO ON LINE

Tecnologie digitali per il suono e l'immagine 2020/21

Vincenzo Lombardo
Corso di Laurea in DAMS
Università di Torino

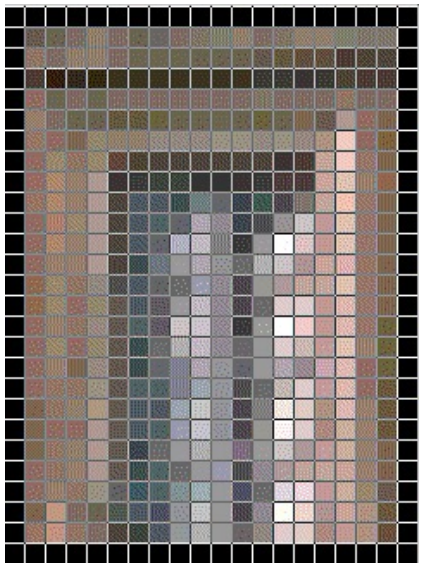
Mutuato in parte da Elaborazione audio e musica
(Laurea Magistrale di Informatica)

Memorizzazione delle immagini

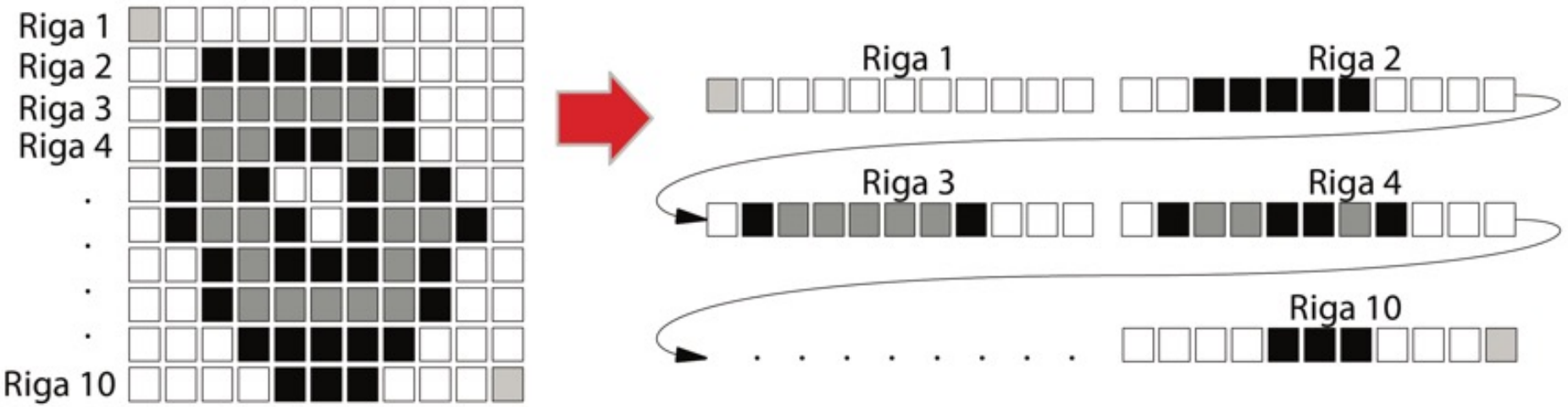
Metodi e Formati

Dati bitmap

Insieme di valori numerici che specificano i colori dei singoli pixel



Linearizzazione per righe



Formati e file grafici

- Come si registrano i dati ? Quale formato utilizzare ? GIF, JPEG, ecc.

- Ciascun formato si preoccupa di definire
  - quali sono le informazioni necessarie
  - in che modo vengono memorizzate

Il formato raw

- elenco dei valori numerici di ciascun pixel
- file che permette di ricostruire l'immagine
- sembra un modo semplice ed universale per descrivere un'immagine

Problemi con il raw format

- Chi legge il file non conosce …
  - le dimensioni in pixel dell'immagine
  - quanti bit per pixel
  - cosa rappresenta il valore numerico (intensità luminosa, valore di colore, ecc.)

- quando inizia o termina un pixel?
- quando inizia o termina una linea?

File grafici: esigenza di un formato

Immagine

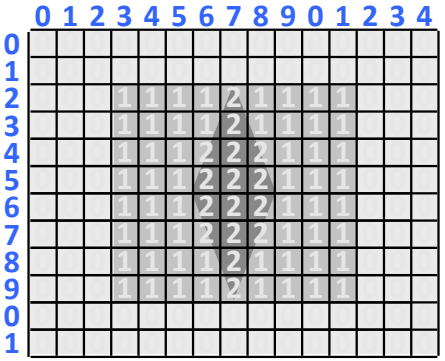
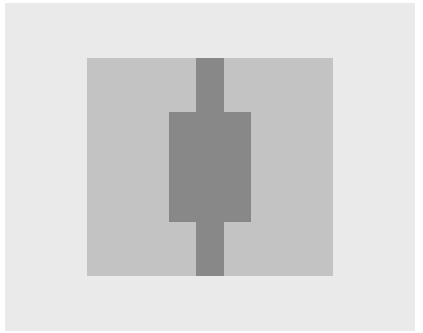


Immagine Quantizzata

15 x 12

Bitmap

Rendering (20 x 9)



Rendering (15 x 12)

File grafici con formato esplicito

- Esistono più di 200 formati di file grafici

- Parametri
  - profondità di pixel
  - palette (esistenza e tipo)
  - compressione dati (esistenza e tipo)
  - formati utilizzabili su una o più piattaforme
  - formati per applicazioni specifiche

File bitmap

- I formati bitmap variano molto nei dettagli ma condividono la stessa struttura generale

- Un file bitmap è organizzato
  - componenti base
  - componenti specifiche

Componenti di base

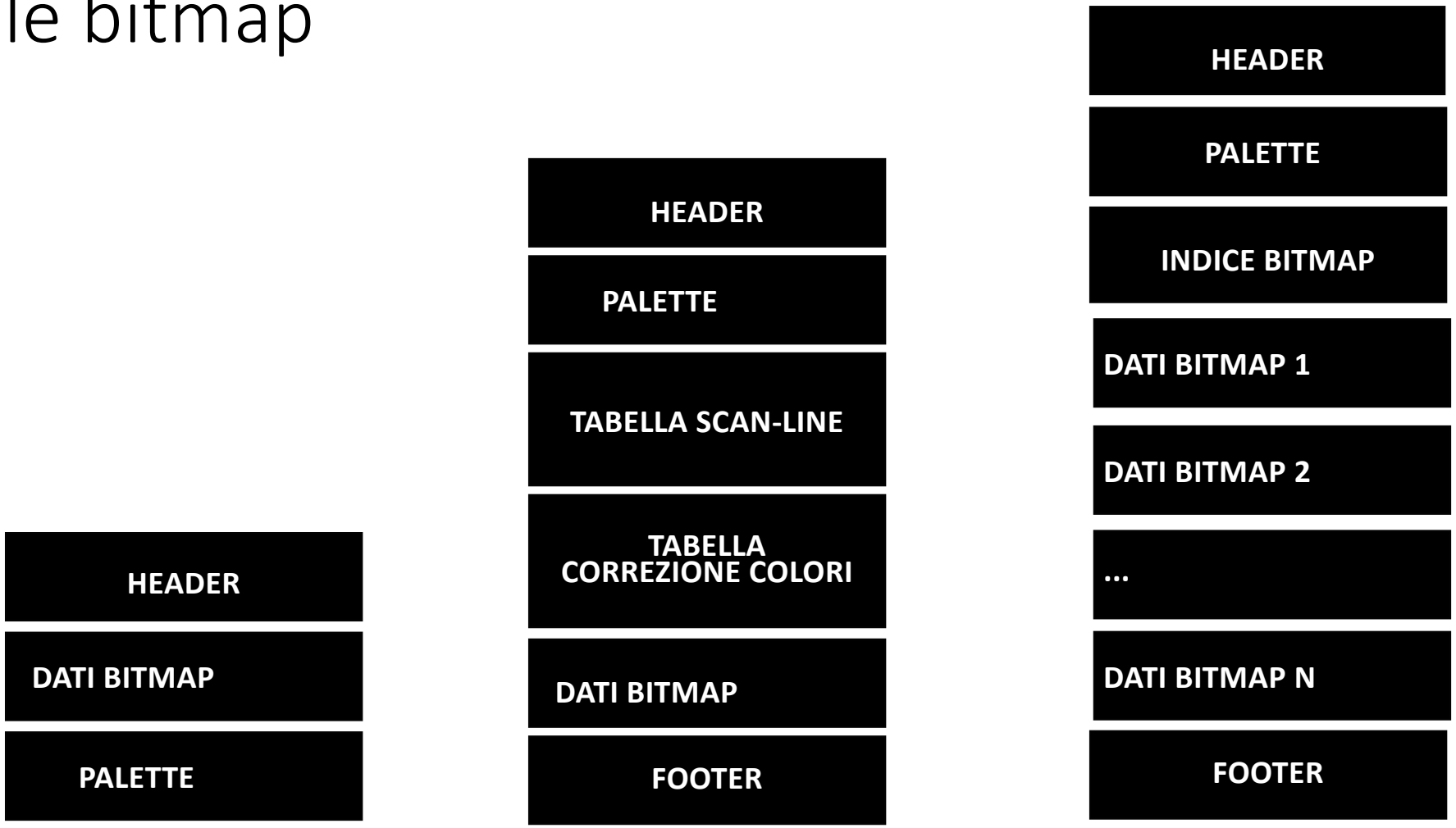
- intestazione (header)
- dati bitmap
- coda (footer) (non sempre presente)

Componenti specifiche

- la palette dei colori
- la tabella delle scan-line
- la tabella di correzione dei colori
- l'indice delle bitmap (immagini multiple)

$$V_{out} = AV_{in}^{\gamma}$$

File bitmap


HEADER
DATI BITMAP
PALETTE
HEADER
PALETTE
TABELLA SCAN-LINE
TABELLA CORREZIONE COLORI
DATI BITMAP
FOOTER
HEADER
PALETTE
INDICE BITMAP
DATI BITMAP 1
DATI BITMAP 2
...
DATI BITMAP N
FOOTER

Header (Intestazione)

- Informazioni sui dati bitmap all'inizio del file

- Campi fissi comuni:
  - identificatore del formato
  - linee per immagine, pixel per linea
  - bit per pixel, canali di colore
  - tipo di compressione
  - origine X e Y dell'immagine
  - spazio utilizzato

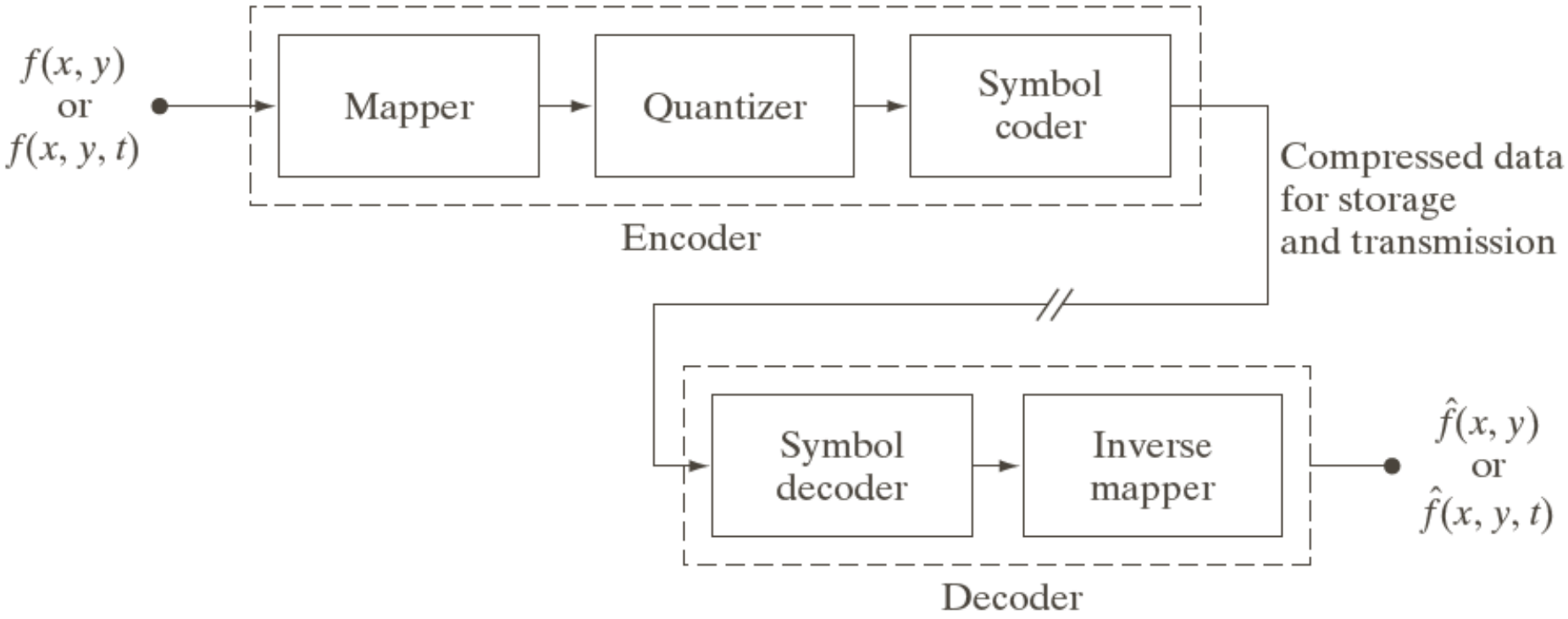
Compressione delle immagini digitali

Esempi di ridondanza dei dati

a b c

FIGURE 8.1 Computer generated 256 × 256 × 8 bit images with (a) coding redundancy, (b) spatial redundancy, and (c) irrelevant information. (Each was designed to demonstrate one principal redundancy but may exhibit others as well.)

Diagrammi a blocchi della compressione



Meccanismi di compressione dati

- simmetrici/asimmetrici
- lossless/lossy
- adattativi/non-adattativi

Compressione dei dati bitmap

- RLE (Run Length Encoding)
- LZW (Lempel-Ziv-Welch)
- CCITT (variante del metodo di Huffman)

- Uso di trasformata DCT (Discrete Cosine Transform, usato in JPEG)

RLE

Metodo Run Length Encoding

Si adatta a qualsiasi tipo di dato

- in genere non raggiunge rapporti di compressione molto buoni
- ma è veloce da eseguire

Metodo RLE

- Sequenza di elementi (codifiche di pixel) uguali detta *run*
- run "codificato" con due byte: numero + codice
  - Sequenza AAAAAAAAAAAAAAA (15 volte la lettera A) …
  - … codificata come <15, A>

- Codifica RLE detta *packet RLE*
- Si genera un *packet* ad ogni cambiamento di carattere
  - Sequenza AAAAAAbbbXXXXXt …
  - … codificata come <6,A><3,b><5,X><1,t>

Codifica/Decodifica RLE

Codifica

- Si genera un *packet* ad ogni cambiamento di carattere
  - Sequenza AAAAAAbbbXXXXXt …
  - … codificata come <6,A><3,b><5,X><1,t>

Decodifica

- Per ogni coppia <#c,p>, si genera una sequenza di p lunga #c
  - Sequenza <6,A><3,b><5,X><1,t> …
  - … decodificata come AAAAAAbbbXXXXXt

Applicazione di RLE (BMP)

- Tipico di immagini binarie, con lunghi run
- Varianti del metodo prendono in considerazioni pixel adiacenti su linee diverse

LZW

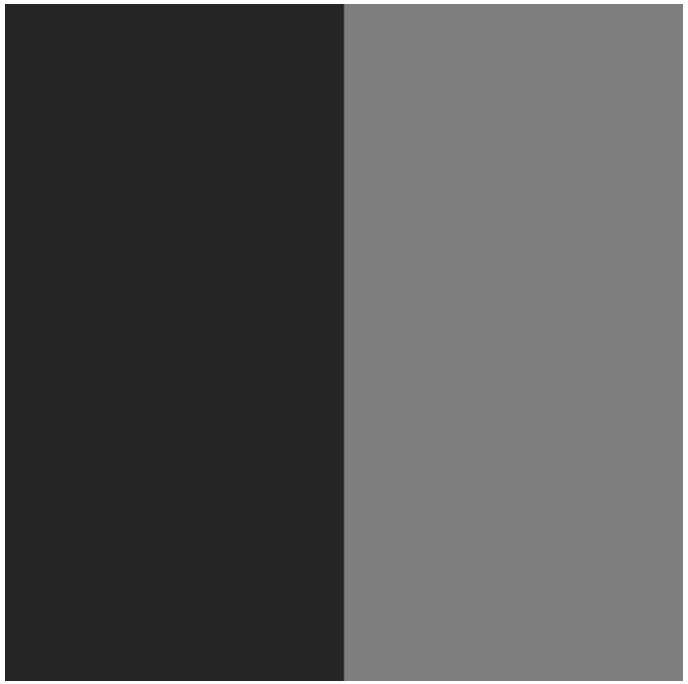
Metodo Lempel-Ziv-Welch

- metodo molto diffuso (compress, pkzip, gzip, ecc.)
- lavora su tutti i tipi di dati
- veloce in compressione e decompressione
- metodo a sostituzione o a dizionario

Metodo LZW

- Immagine come sequenza di dati
- Si identificano pattern (sottostringhe o sottosequenze) e ricercati nel dizionario
- Se non presenti, si costruisce un codice per il pattern e si aggiunge al dizionario
- Se presente, il codice viene scritto nell'output del file compresso

Esempio di codifica LZW



……
… 39 39 126 126 …
… 39 39 126 126 …
… 39 39 126 126 …
… 39 39 126 126 …
……

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

Esempio di codifica LZW

.....
... 39 39 126 126 ...
... 39 39 126 126 ...
... 39 39 126 126 ...
... 39 39 126 126 ...
.....

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

Metodo di Differencing

- Si usa per rendere i dati “meglio comprimibili”

- Soprattutto per immagini con variazioni continue di colore …
  - pixel adiacenti variano molto poco
  - si memorizza la differenza tra valori adiacenti
  - riduzione della quantità di info

Metodo di Huffman

- Info di natura statistica (frequenza con cui si presentano determinate sequenze)

- Codificare sequenze
  - più frequenti con “parole corte”
  - meno frequenti con “parole lunghe”

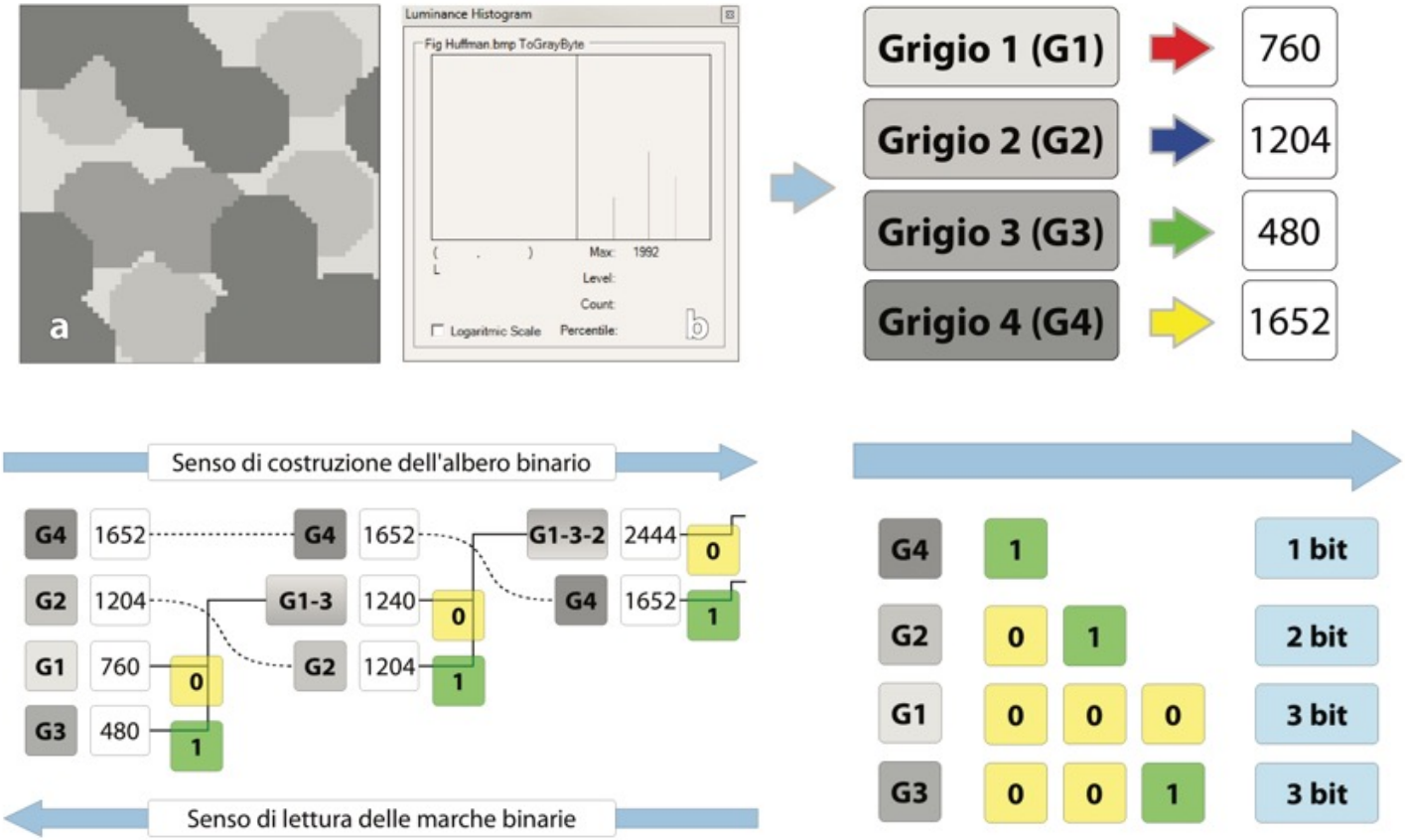
Esempio: Huffman

- Input 11100011 01101110 10011000 01101110 01101110 11001011

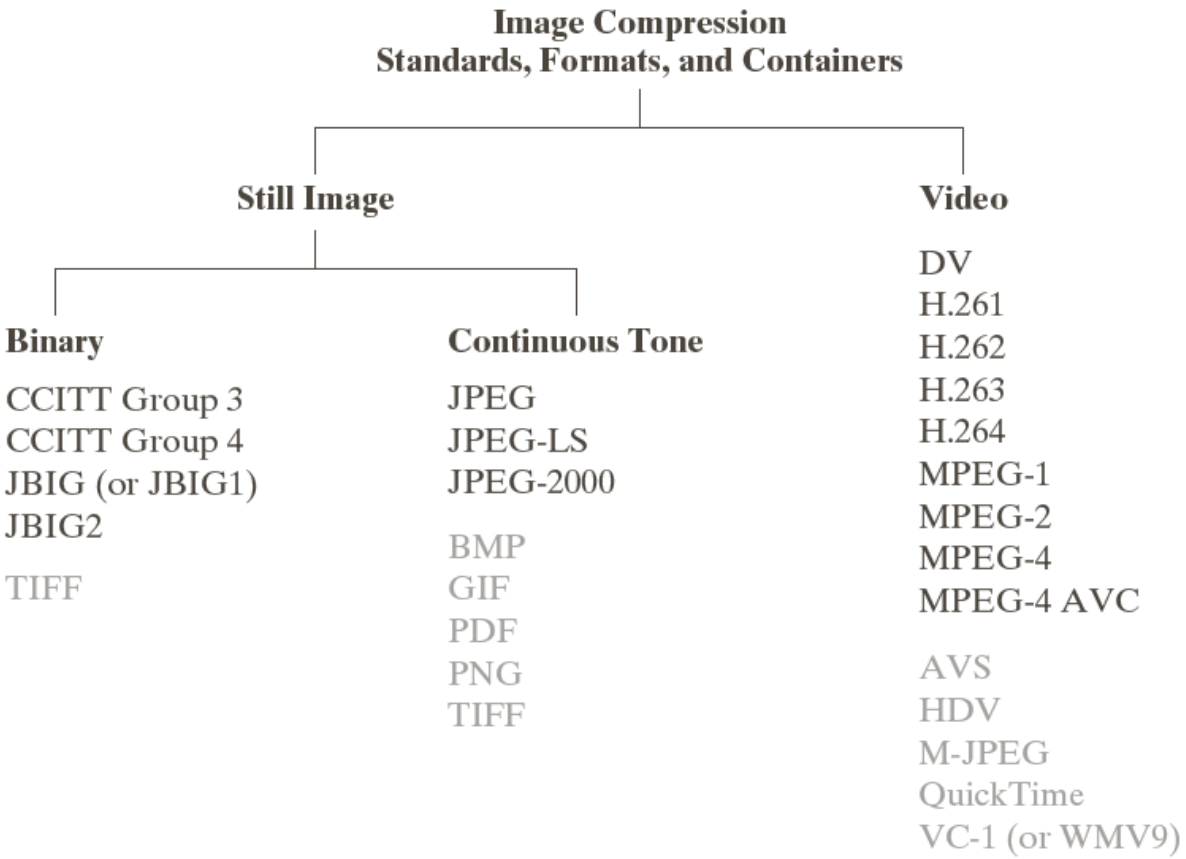
  - Sottostringa Frequenza Codifica
  - 01101110 35 % 11
  - 10011000 25 % 10
  - 01010100 20 % 01
  - 11100011 12 % 001
  - 00001000 5 % 0001
  - 11001011 3 % 0000

  - Output compresso: 001 11 10 11 10 0000

Applicazione di Huffman



Compressione e formati



Alcuni formati di file grafici

- Adobe Photoshop
- Microsoft BMP
- CGM
- EPS
- GIF
- JPEG – JFIF
- Macintosh Pict
- Microsoft RIFF (.AVI, .WAV)
- MPEG
- PDF
- PNG
- QuickTime
- TGA
- TIFF

TIFF (Tagged Image File Format)

- Tipo: Bitmap
- Colori: da 1 a 24-bit
- Compressione: RLE, LZW, nessuna, ma anche JPEG e JPEG 2000
- Piattaforma: Macintosh, Windows, Unix
- Formato bitmap molto versatile
- Sviluppato da Aldus
- Possibilità di memorizzare più immagini nello stesso file

PNG (Portable Network Graphics)

- Formato standard potente e versatile (W3C)
- Motivazione: royalty su LZW, uso di LZ77 (compressione efficiente, differenza e predizione)
- completa portabilità su tutti i sistemi
- Full color + trasparenza (alpha-channel)
- metodo di interlacciamento molto efficace
- codici di autocontrollo per verifica di dati trasmessi

Adobe Photoshop (.psd)

- Formato utilizzato dal programma Photoshop
- Tipo bitmap
- Più modelli di colore (RGB, CMY, ecc.)
- Compressione: Nessuna, RLE
- Sistemi: Macintosh, Windows
- Utilizzo: Applicazione Adobe Photoshop

PDF (Portable Document Format)

- Tipo: Metafile
  - rappresenta documenti 2-D in modalità indipendente da risoluzione e dispositivo
  - Contenitore per immagini compresse (es. JPEG)
- Piattaforme: quasi tutte
- Creato da Adobe.
- Può essere considerato una sorta di evoluzione del formato Postscript (alcune versioni standard)
- È più efficiente nella memorizzazione

TGA

- Formato diffuso (spesso chiamato Targa)
- Utilizzato originalmente su stazioni grafiche dedicate alla gestione di immagini digitalizzate direttamente da segnali video
- Colori: 8-bit, 16-bit, 24-bit, 32-bit
- livelli di grigio e colori
- Compressione: RLE, nessuna

GIF (Graphics Interchange Format)

- Tipo: Bitmap
- Colori: da 1 a 8 bit (da 2 a 256 colori)
- Compressione: LZW
- Piattaforme: Macintosh, Windows, Unix
- Formato di uso generale: ben definito, ben documentato, molto diffuso, supportato da numerose applicazioni
- Sviluppato dalla Compuserve Inc.
  - ne ha definito le specifiche nel 1987
  - ha aggiunto nel 1989 nuove funzionalità
  - esistono due versioni del formato (GIF87a e GIF89a)

GIF Interlacciate

- Interlacciamento = suddivisione del fotogramma in linee numerate che vengono poi lette in un ordine stabilito, diverso da quello sequenziale

- Bozza dell’immagine, che si affina sempre più

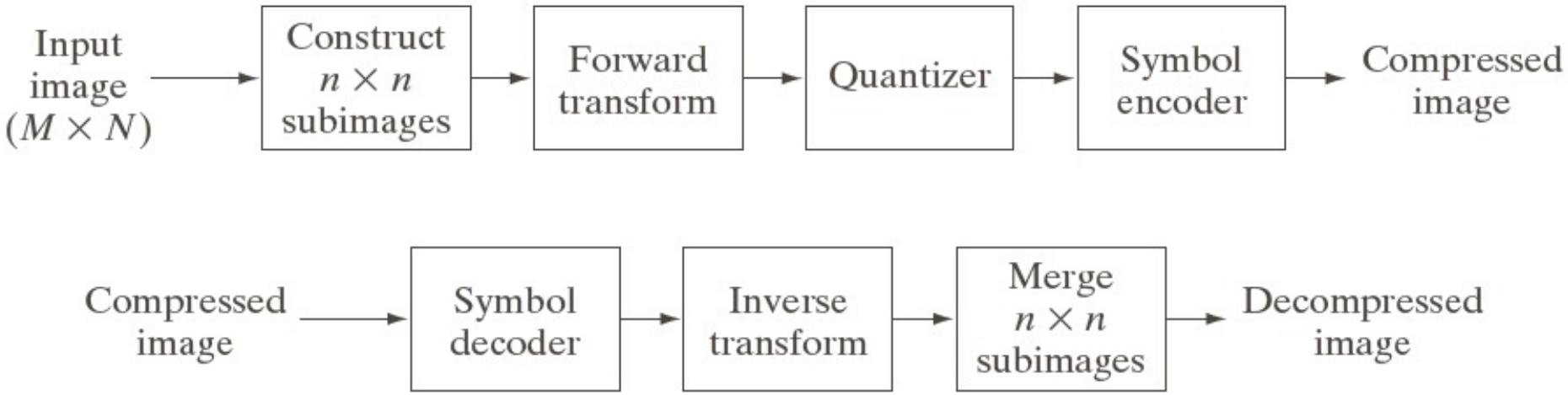
GIF animate

- È possibile costruire delle animazioni «componendo» una sequenza di immagini codificate come GIF

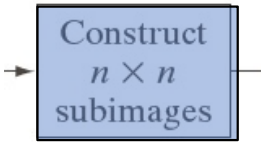
- Esistono diversi programmi che permettono di costruire queste animazioni
  - Es: https://gifmaker.me
  - Livelli di Photoshop o GIMP

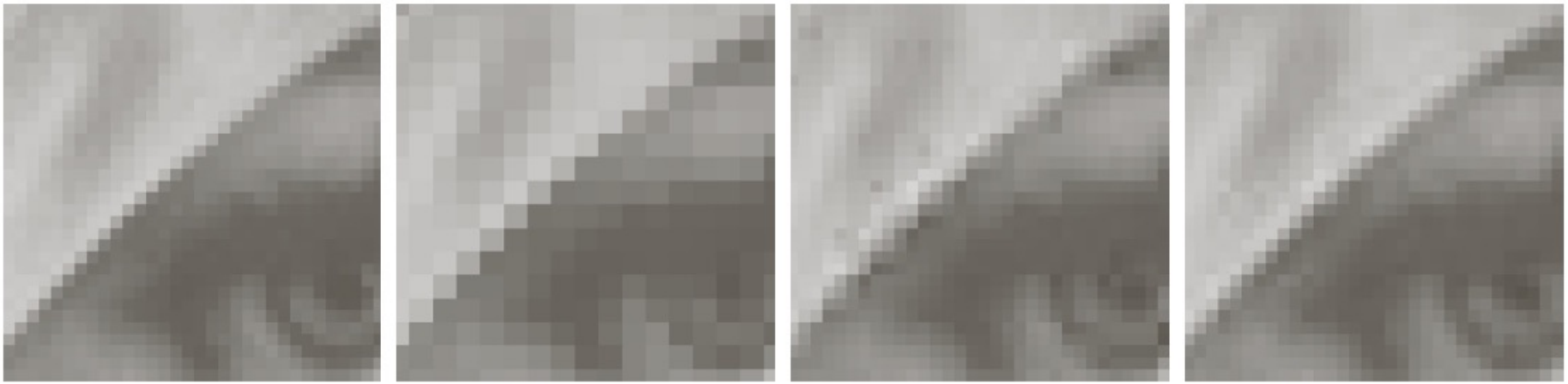
Compressione basata su trasformata

Block coding con trasformata



Dimensione sotto-immagini

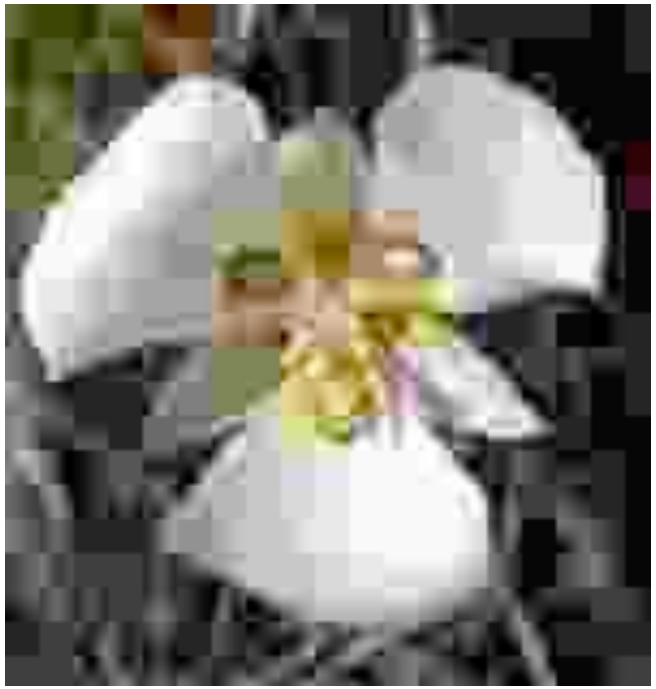




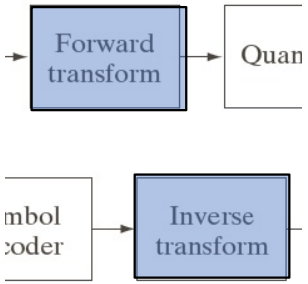
a b c d

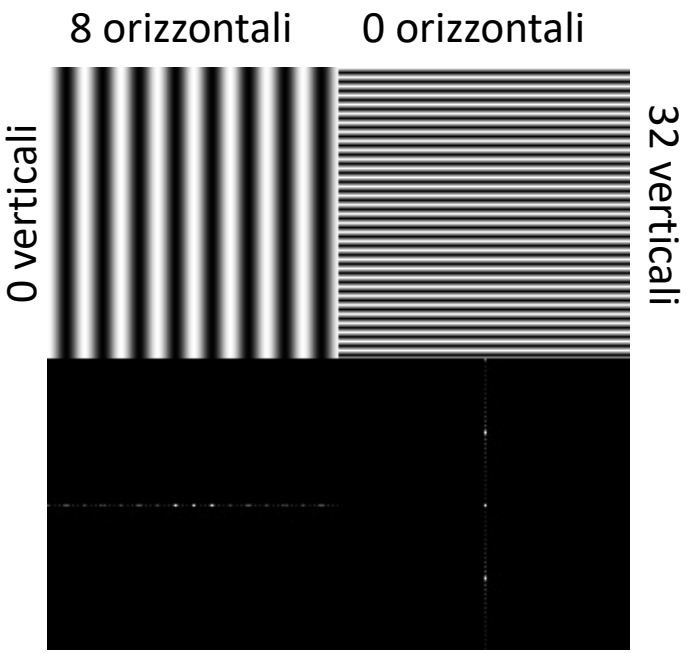
FIGURE 8.27 Approximations of Fig. 8.27(a) using 25% of the DCT coefficients and (b) 2×2 subimages, (c) 4×4 subimages, and (d) 8×8 subimages. The original image in (a) is a zoomed section of Fig. 8.9(a).

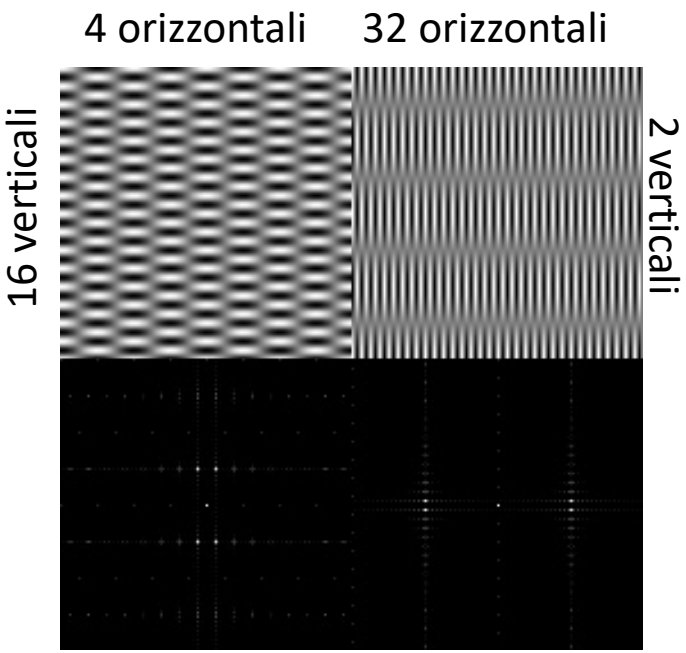
Blocking artefact



Funzioni base Fourier

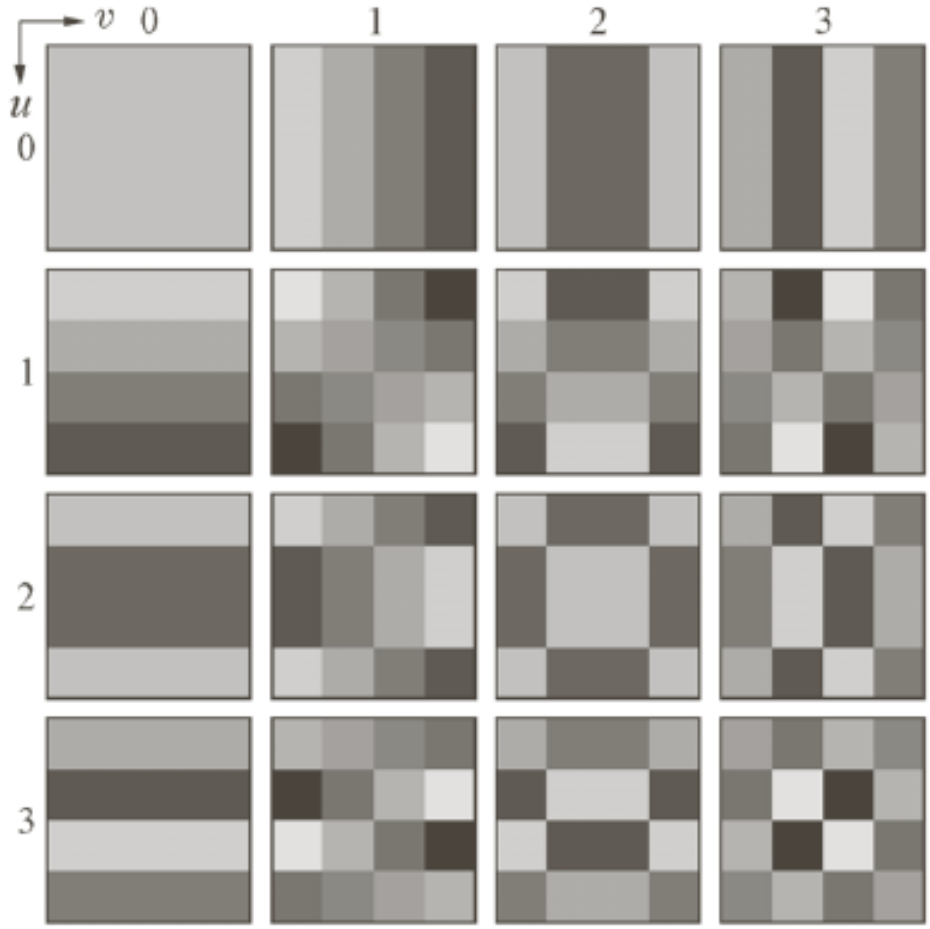






http://www.cs.unm.edu/~brayer/vision/fourier.html

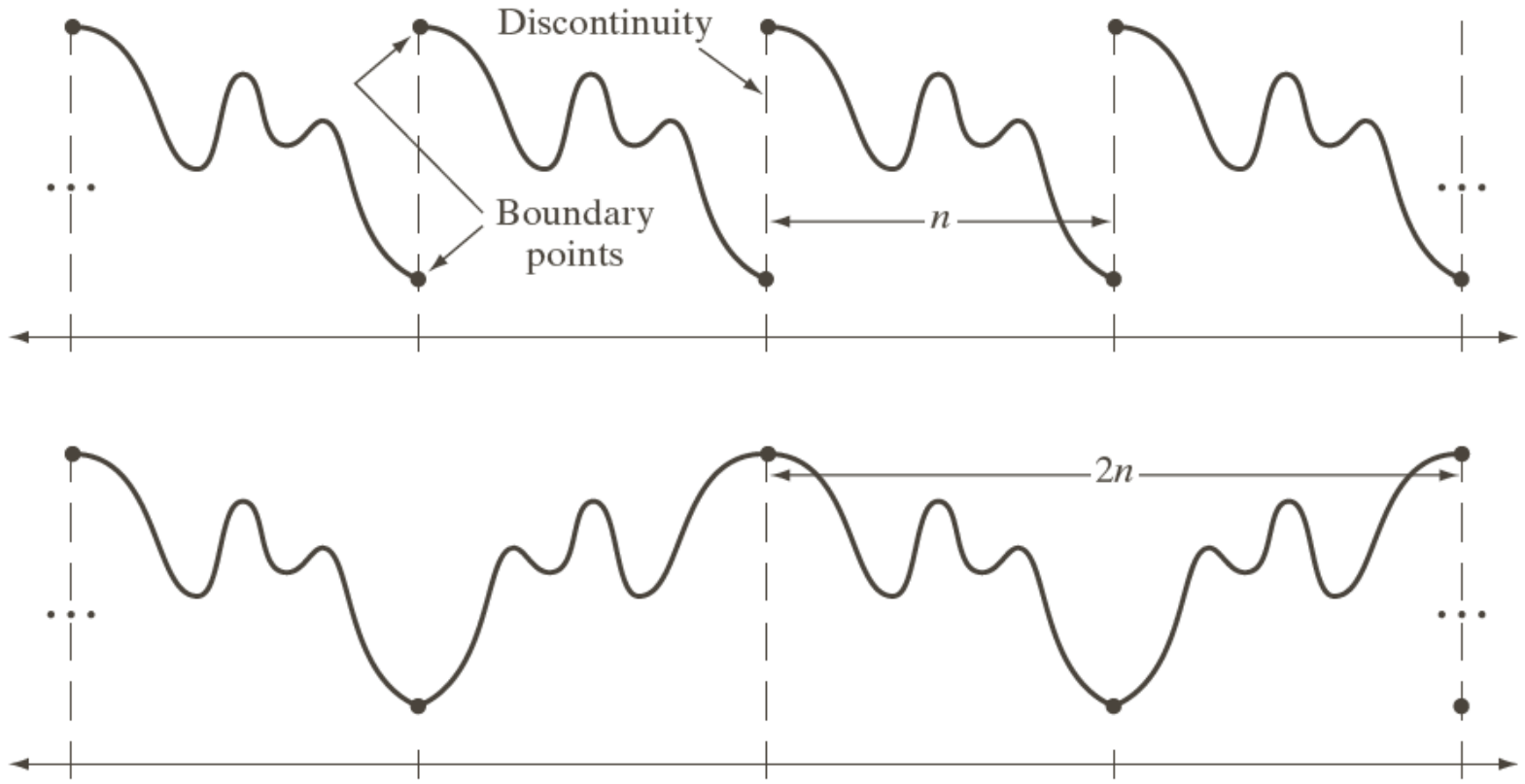
Funzioni base DCT (n=4)



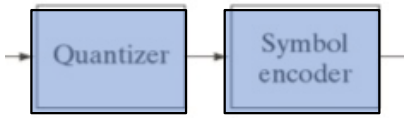
Periodicità implicita

DFT

DCT



Allocazione dei bit



a b
c d

FIGURE 8.28 Approximations of Fig. 8.9(a) using 12.5% of the 8×8 DCT coefficients: (a)—(b) threshold coding results; (c)—(d) zonal coding results. The difference images are scaled by 4.

Codifica zonale/soglia

1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	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	0

8	7	6	4	3	2	1	0
7	6	5	4	3	2	1	0
6	5	4	3	3	1	1	0
4	4	3	3	2	1	0	0
3	3	3	2	1	1	0	0
2	2	1	1	1	0	0	0
1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0

1	1	0	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	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	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

a b
c d

FIGURE 8.29
A typical (a) zonal mask, (b) zonal bit allocation, (c) threshold mask, and (d) thresholded coefficient ordering sequence. Shading highlights the coefficients that are retained.

JPEG

JPEG (Joint Photographic Expert Group)

- Uno standard
- JFIF = JPEG File Interchange Format
- Più formati grafici utilizzano la tecnica JPEG
  - JFIF
  - TIFF, versione 6

Compressione JPEG

- Si applica sia a immagini a colori a 24 bit che a immagini a livelli di grigio
- Particolarmente indicato per immagini di tipo fotografico
- Non adatto a variazioni brusche di intensità e colore delle immagini artificiali (contorni netti)

Vantaggi JPEG

- elevati fattori di compressione senza perdita di qualità
- diversi livelli di qualità

JPEG

- Rapporto di compressione C dipende dal contenuto dei dati (tipicamente da 20:1 a 25:1)
- Controllo qualità variabile mediante parametro Q
- JPEG non ideale con immagini con grandi zone dello stesso colore

Passi compressione JPEG

- Trasforma l'immagine in uno spazio di colori ottimale
- Downsampling sulle componenti di crominanza
- Applica la DCT (Discrete Cosine Transform) ai blocchi di pixel per rimuovere i dati "ridondanti"
- Quantizza ogni blocco di coefficienti DCT usando una funzione ottimizzata per l'occhio umano
- Codifica i coefficienti risultanti con uno dei metodi di Huffman

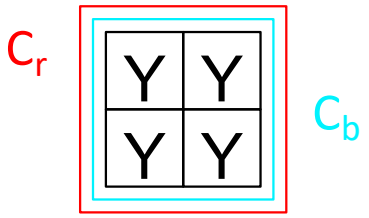
1. Trasformare lo spazio dei colori

- Si parte da qualsiasi modello di colore (RGB, CMY, ecc.).
- Si arriva al modello luminanza/crominanza (YC_bC_r)
- Occhio più sensibile alla luminanza (Y) rispetto alla crominanza (C_bC_r)

2. Downsampling sulle componenti di crominanza

- Più pixel per il canale Y, meno pixel per C_b e C_r

- Esempio: immagine di 1000x1000 pixel
  - 1000x1000 pixel per (Y)
  - solo 500x500 pixel per C_b e C_r
  - Totale: 6 valori, 4 di luminanza e 1 per ogni canale di crominanza (invece di 12)

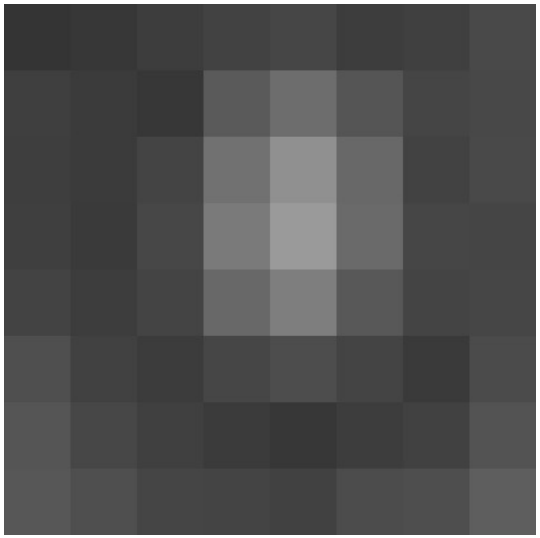


3-4-5 DCT, quantizzazione e codifica

- Immagine è divisa in blocchi di 8x8 pixel
- Ad ogni blocco si applica la DCT: si rappresentano le variazioni di colore nei blocchi
- Si rimuovono le variazioni piccole (alte frequenze)

Esempio: immagine 8x8

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94



[Wikipedia]

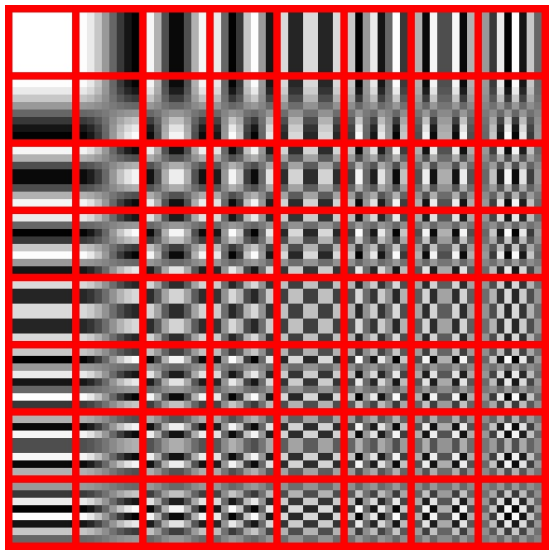
Traslazione di livello: -2^{k-1}, k numero di bit

52	55	61	66					
63	59	66	90					
62	59	68	113	-128				
		−76	−73	−67	−62	−58	−67	−64	−55
		−65	−69	−62	−38	−19	−43	−59	−56
		−66	−69	−60	−15	16	−24	−62	−55
		−65	−70	−57	−6	26	−22	−58	−59
		−61	−67	−60	−24	−2	−40	−60	−58
		−49	−63	−68	−58	−51	−65	−70	−53
		−43	−57	−64	−69	−73	−67	−63	−45
		−41	−49	−59	−60	−63	−52	−50	−34

Dati centrati su 0 [−128,127]; si riduce la gamma dinamica per la trasformata DCT

Trasformata DCT, n=8

−415	−29	−62	25	55	−20	−1	3
7	−21	−62	9	11	−7	−6	6
−46	8	77	−25	−30	10	7	−5
−50	13	35	−15	−9	6	0	3
11	−8	−13	−2	−1	1	−4	1
−10	1	3	−3	−1	0	2	−1
−4	−1	2	−1	2	−3	1	−2
−1	−1	−1	−2	−1	−1	0	−1



Matrice di quantizzazione

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

Diverse normalizzazioni

FIGURE 8.31 Approximations of Fig. 8.9(a) using the DCT and normalization array of Fig. 8.30(b): (a) **Z**, (b) 2**Z**, (c) 4**Z**, (d) 8**Z**, (e) 16**Z**, and (f) 32**Z**.

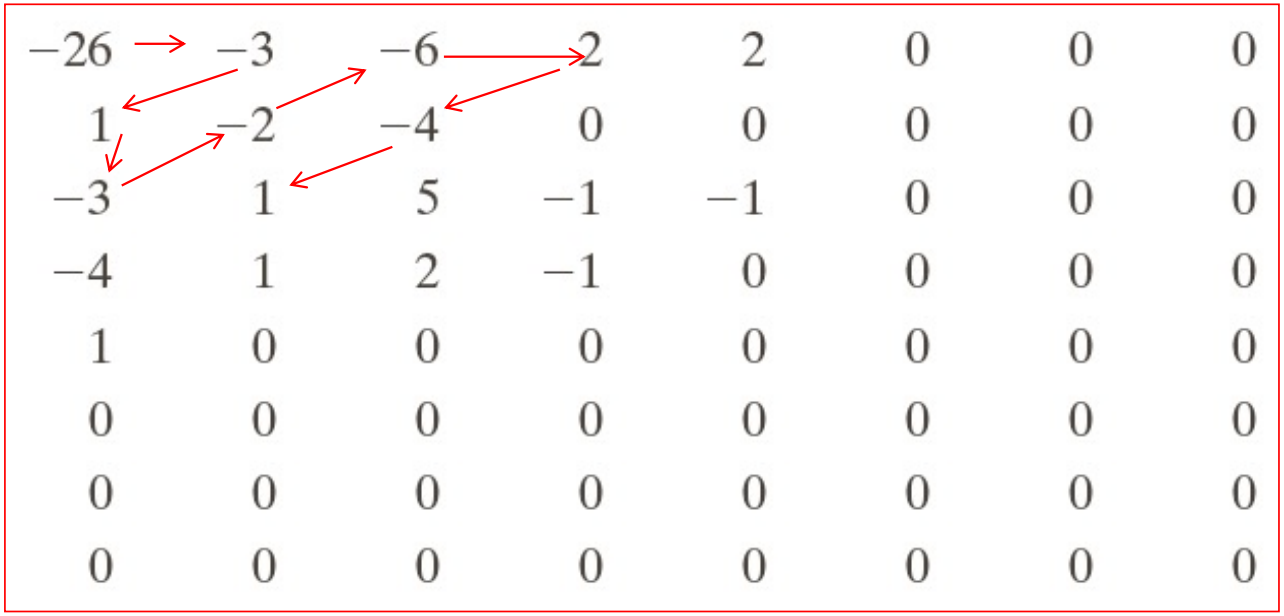
Coefficienti scalati e troncati

−415	−29	−62	25	55	−20	−1	3
7	−21	−62	9	11	−7	−6	6
−46	8	77	−25	−30	10	7	−5
−50	13	35	−15	−9	6	0	3
11	−8	−13	−2	−1	1	−4	1
−10	1	3	−3	−1	0	2	−1
−4	−1	2	−1	2	−3	1	−2
−1	−1	−1	−2	−1	−1	0	−1

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

−26	−3	−6	2	2	0	0	0
1	−2	−4	0	0	0	0	0
−3	1	5	−1	−1	0	0	0
−4	1	2	−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	0

Sequenza a zig-zag

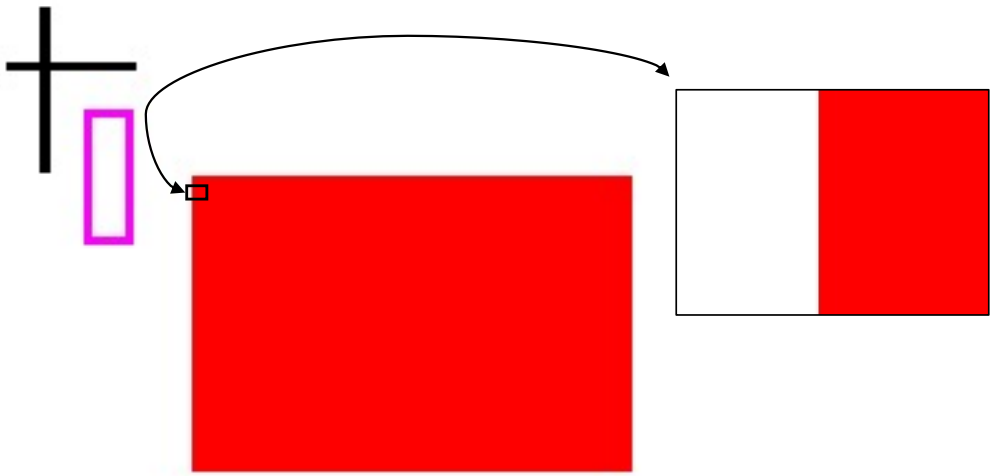


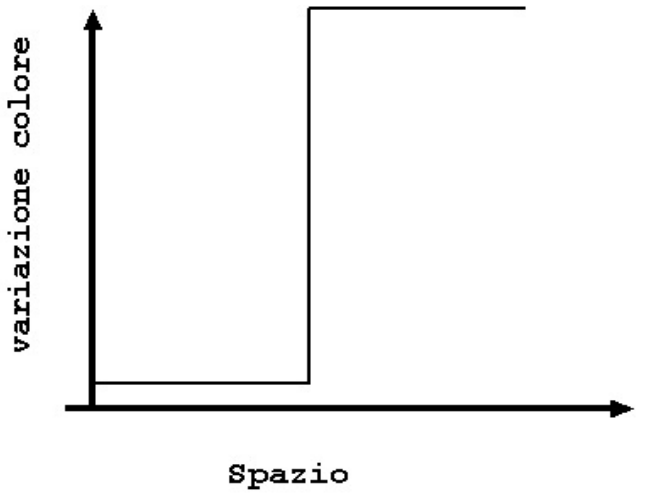
−26	−3	−6	2	2	0	0	0
1	−2	−4	0	0	0	0	0
−3	1	5	−1	−1	0	0	0
−4	1	2	−1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

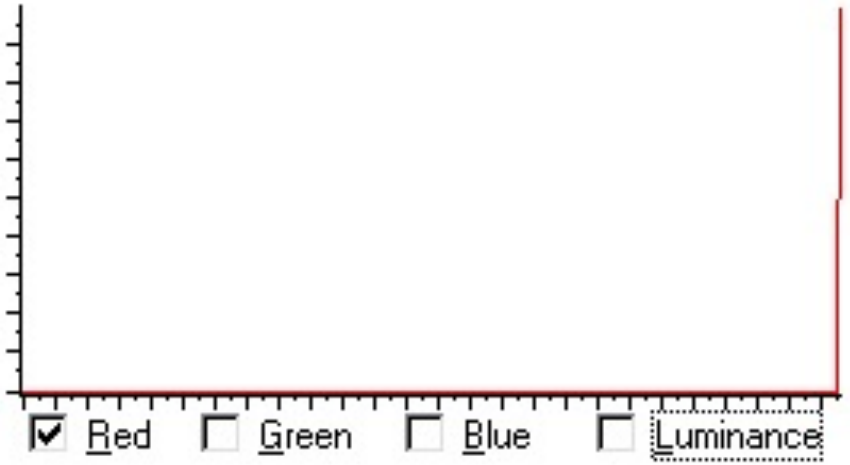
[−26 −3 1 −3 −2 −6 2 −4 1 −4 1 1 5 0 2 0 0 −1 2 0 0 0 0 0 −1 −1 EOB]

Animazione DCT

By Hanakus - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=12624397



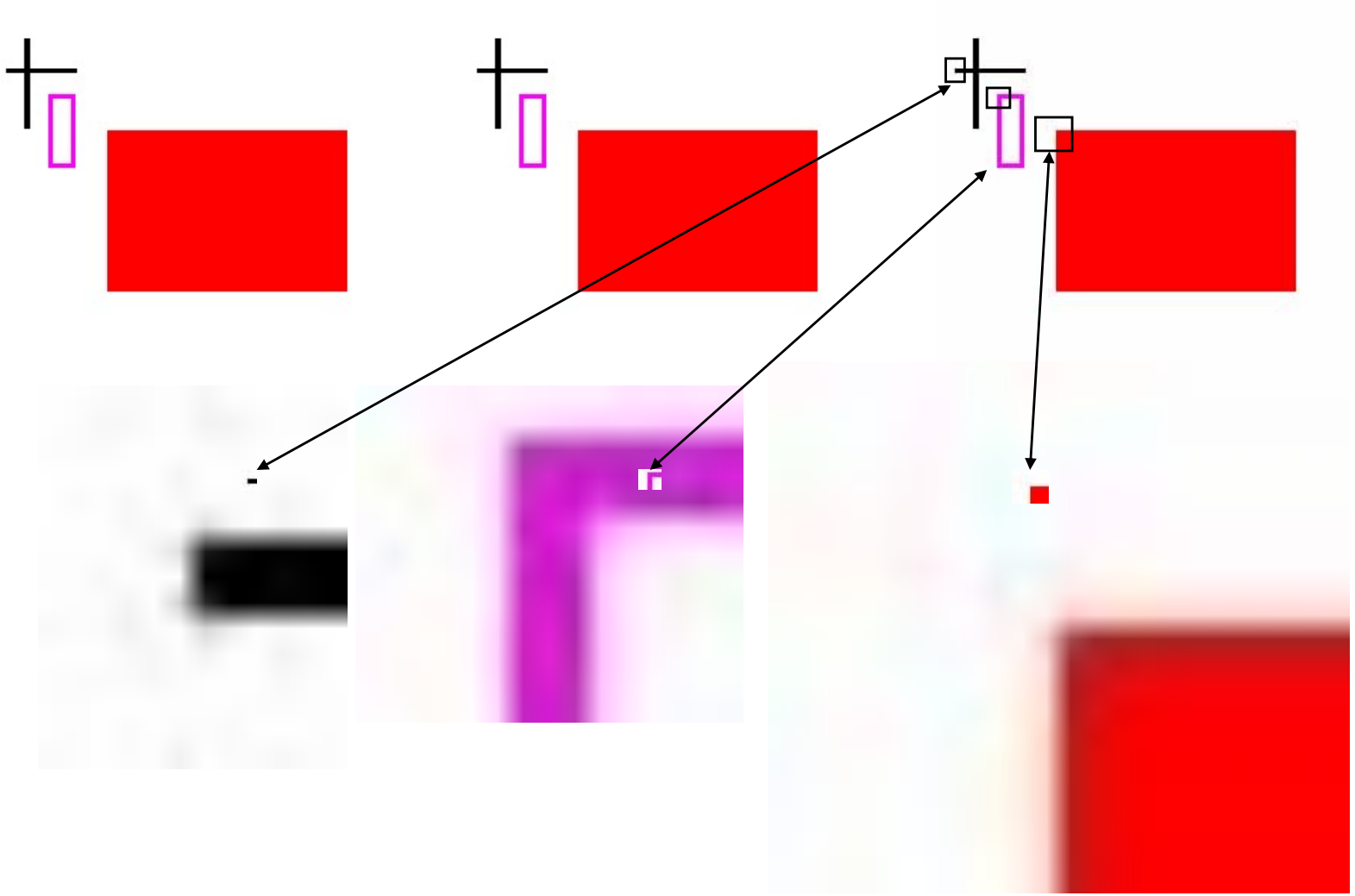

variazione colore
Spazio


Red
Green
Blue
Luminance

Originale (TIFF o GIF)

JPEG buona qualità

JPEG bassa qualità



Grazie dell’attenzione