

MAR/Gio

8:00 - 11:00

# INTERAZIONE e MULTIMEDIA

prof. DARIO  
ALLEGRA

prof. FRANCESCO  
GUARNERA

## LEZ. INTRODUTTIVA

Propedeuticità: Prog I

- scaricare processing x phass. LEZIONE

- ESAME LABOR. : 2 esercizi da 15 PUNTI (EXAM BOX) NO ITINERE

- ESAME SCRITTO : Itinere, tempo SCRITTO FINALE

1 metà CORSO  
1 FINE CORSO

2 FASI (UNICA SESSIONE)

SE SUP  
Si VA ↗ 1) RISPOSTA MULTIPURA  
2) SCRITTO CON 2 ESERCIZI  
( $3 \times 2 + b - 4$ )  
PUNTEGGIO

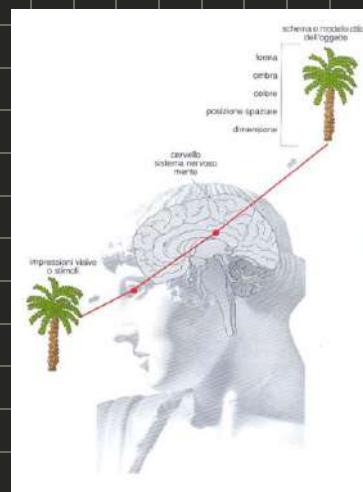
## PPT. 2 - i FONDAMENTI

- Leggi della PERCEZIONE VISIVA

Le nostre modalità di REAZIONE

a gli STIMOLI VISIVI e FORMARE

IMMAGINI è SOGGETTUO



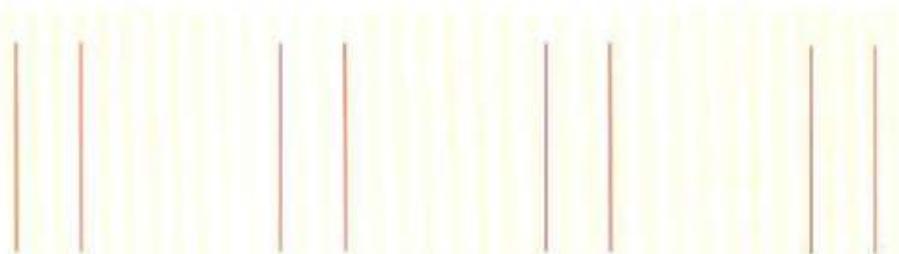
Le nostre occhi sono corpi di CARNE quindi è un BIANCO  
NELLA REALTÀ, come il BIANCHIMENTO del BIANCO nella  
FOTOCAMERÀ

→ TUTTI GLI ALTRI COLORI LI CALIBRAVANO  
di CONSEGUENZA. GAY UN BASE  
PER LA PERCEZIONE

### 1) VICINANZA

#### 1. LEGGE DELLA VICINANZA

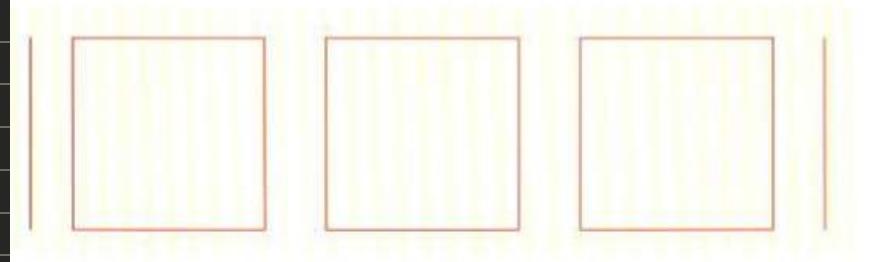
Le parti più vicine di un insieme entrato nel nostro campo visivo tendono a essere percepiti come raggruppate in unità. Nell'esempio in figura i segmenti verticali non sono percepiti uno a uno ma accoppiati, due a due.



### 2) CHIUSURA

#### 2. LEGGE DELLA CHIUSURA

Le figure chiuse si vedono più facilmente di quelle aperte e tendono a prevalere su di esse. Nell'esempio in figura i segmenti sono stati uniti orizzontalmente e li percepiamo come quadrati. Triangoli, quadrati, cerchi, sono forme chiuse semplici e unitarie che si impongono facilmente alla nostra visione.

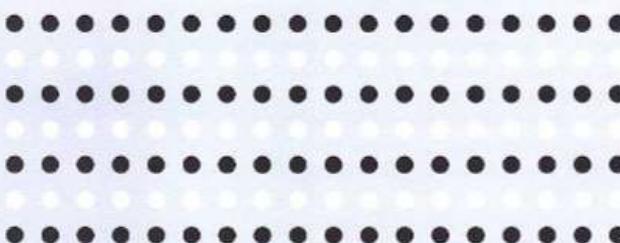


### 3) UGUALIANZA

#### 3. LEGGE DELL'UGUALIANZA (O DELLA SOMIGLIANZA)

Elementi uguali, o che si assomigliano, vengono percepiti insieme.

I punti neri e i punti bianchi della figura a fianco si uniscono in strutture e tendiamo a vederli come linee orizzontali di punti neri e di punti bianchi.



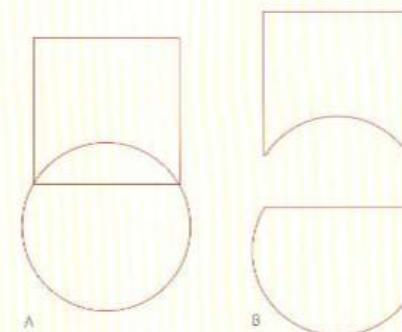
### a) CONTINUITÀ

#### 4. LEGGE DELLA CONTINUITÀ

Un insieme di punti posti uno di seguito all'altro in una determinata direzione vengono percepiti come una struttura unitaria.

Per lo stesso motivo, quando si uniscono o sovrappongono forme diverse continuiamo a vederle distinte.

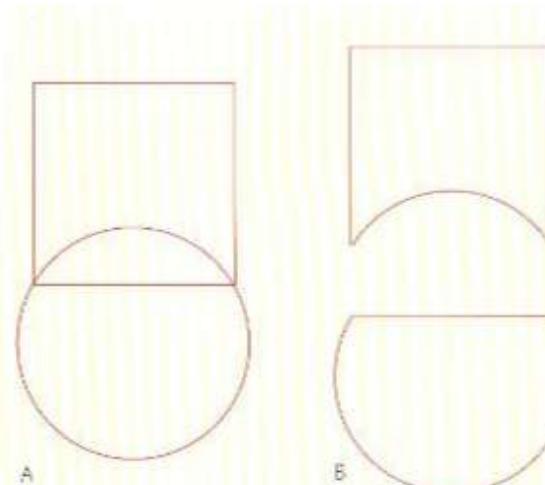
Nella figura il cerchio e il quadrato vengono percepiti come figure complete e non come mancanti di qualche parte.



### 5) BUONA FORMA

#### 5. LEGGE DELLA BUONA FORMA

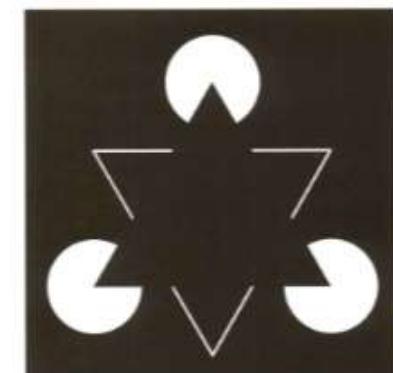
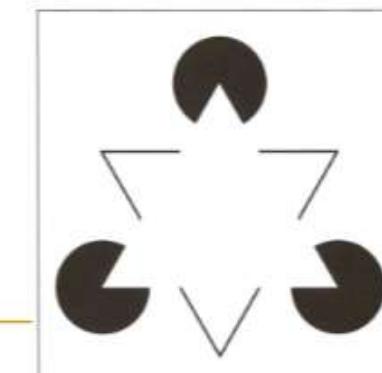
Figure diverse, quando si uniscono, finiscono per avere ciascuna la propria forma, anche se in realtà questa non compare. Quando le figure di B vengono unite come in A, vengono percepite come un cerchio e un quadrato sovrapposti.



### 6) NIENTE NOME... SPORRY

7. La mente tende naturalmente a vedere figure chiuse, a completare i contorni interrotti, a percepire spontaneamente forme semplici, regolari, strutture unitarie e compatte.

Nelle due figure in basso il triangolo che si forma, in bianco o in nero, è una figura nascosta, un fantasma, cioè un'immagine non reale materialmente, ma illusoria.



## INTENSITÀ PERCEPITA

l'intensità percepita è funzione logaritmica dell'intensità incidente nell'occhio.



## RANGE DINAMICO

Il sistema visivo umano riesce a operare solo su una **porzione** di "Range delle intensità percepite".



Inoltre riesce a distinguere in maniera differente se si trova in una **zona chiara o in una zona scura**.

## RAPPRESENT. di UN' IMMAGINE

Si può rappresent. da una **FUNZIONE BIIDIMENSIONE  $f(x,y)$**

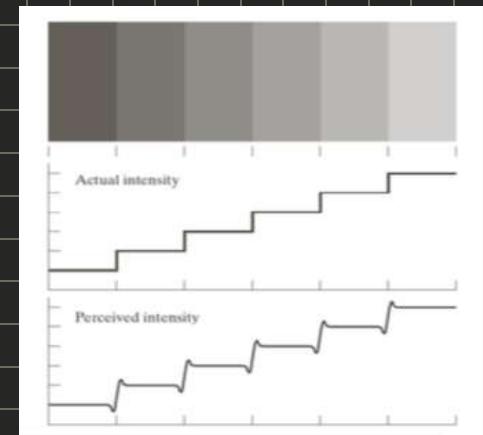
## ILLUSIONI OTTICHE

Esistono dei fenomeni di illusioni ottiche che sono tipiche del nostro sistema visivo umano e che non sono ancora state spiegate.

**luminosità percepita non è semplicemente in funzione dell'intensità emessa.**

## BANDE DI MACH

Anche se le bande hanno una **intensità costante**, esse vengono **percepite in maniera non uniforme all'approssimarsi dei bordi**.



**$f(x,y)$**

$$\underline{u(x,y)} \cdot \underline{r(x,y)}$$

*lumen su metro quadrato*  
 $\text{lm} / \text{m}^2$

luce incidente  
 $0 < u(x,y) < \infty$

luce riflessa  
 $0 < r(x,y) < 1$

IL PIANO IN CUI STANNO LE COORDINATE dell' IMMAGINE  
é detto **DOMINIO SPAZIALE** e le  $(x,y)$  variano **COORDINATE SPAZIALI**

# IMMAGINI VETTORIALI e RASTER

DISEGNO  
A MANO

↳ NUM. LIMITATO di PUNTI, rappresentato con una MATRICE

↳ IN BASE AL VALORE INSERITO si REGOLA l'INTENSITÀ LUMINOSA

Pro

- FOTOREALISMO
- STANDARD SU WEB

Contro

- NESSUNA DESCR. SEMANTICA
- GRANDI DIMENSIONI

(non so cosa ho messo)  
NELL'IMMAGINE come quadrati, cerchi

DISEGNO  
TECNICO

DESCRIVO il mio GRADÙ oli ELEMENTI GRAFICI (PRIMITIVE e GEOMETRICHE)

CON CUI XÌ POSS. RAPPRESENTAZIONI IMMAG. COMPLESSE ma molto più COMPLICATO

Si "DISEGNA DA 0", quindi per OPERAZ. oli SCALING o ZOOMING è più "SEMPLICE"  
(in maniera libera)

STAMPATE CON MACHINE APROPRIATE (PLOTTER)

PICTURE  
ELEMENTS

Pro

- (SEMPL.)

Contro

- NO FOTOREALISMO
- FORMATI PROPRIETARI

# PSNR

PÚ É AUTO, PIÙ 2 IMMAGINI SI SOMIGLIANO.

( QUANDO SUPERÀ IL 35,000 LA DIFFERENZA DI IMMAGINI È IMPERCETTABILE )

## FORMATI GRAFICI VETTORIALI

- **PS** (Postscript): Formato sviluppato da AdobeSystems originariamente per la stampa di documenti su stampanti laser, è utilizzato anche per la memorizzazione di immagini vettoriali.
- **EPS** (Encapsulated Postscript): Estensione del formato PostScript che consente di incapsulare immagini bitmap (raster).
- **DCS** (Desktop Color Separation): Un caso speciale di EPS sviluppato originariamente da Quark per tenere separati i dati al alta risoluzione dall'anteprima a bassa risoluzione.
- **PDF** (Portable Data Format): Sviluppato da Adobe, è il formato più diffuso per condividere, indipendentemente dalla piattaforma, documenti di testi e immagini.
- **PICT**: Formato grafico sviluppato da Apple Computer per la piattaforma Macintosh in grado di memorizzare sia immagini vettoriali che raster.

## PIXEL

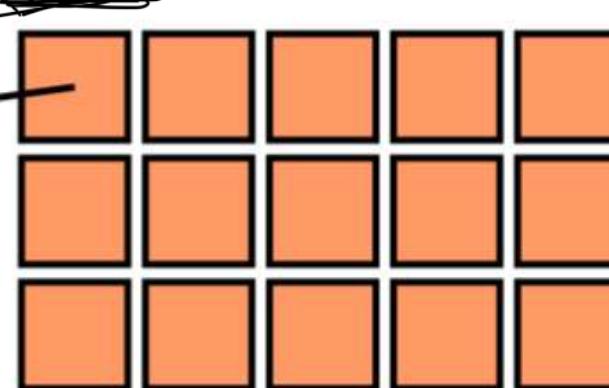
LE VALORIE QUANTIZZATO MISURATO DA CIASCON GENERE DIVENTANO

CHE UN PICTURE ELEMENT = PIXEL DELL'IMMAGINE

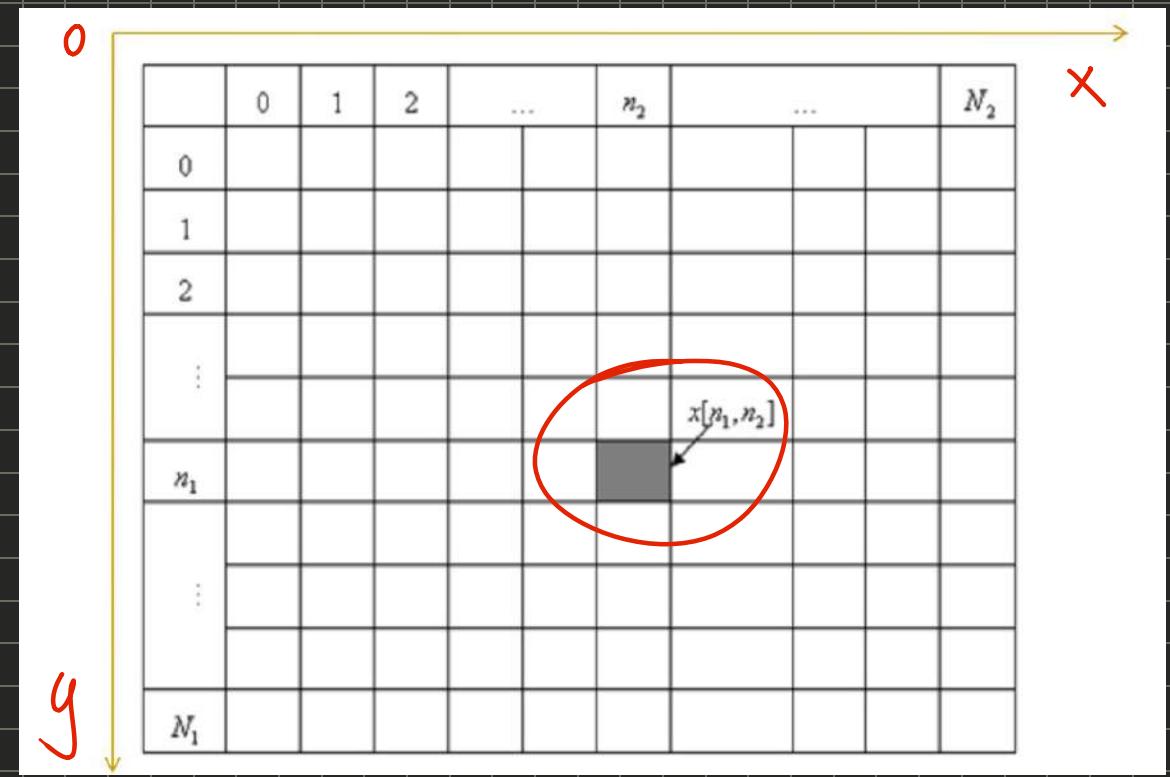
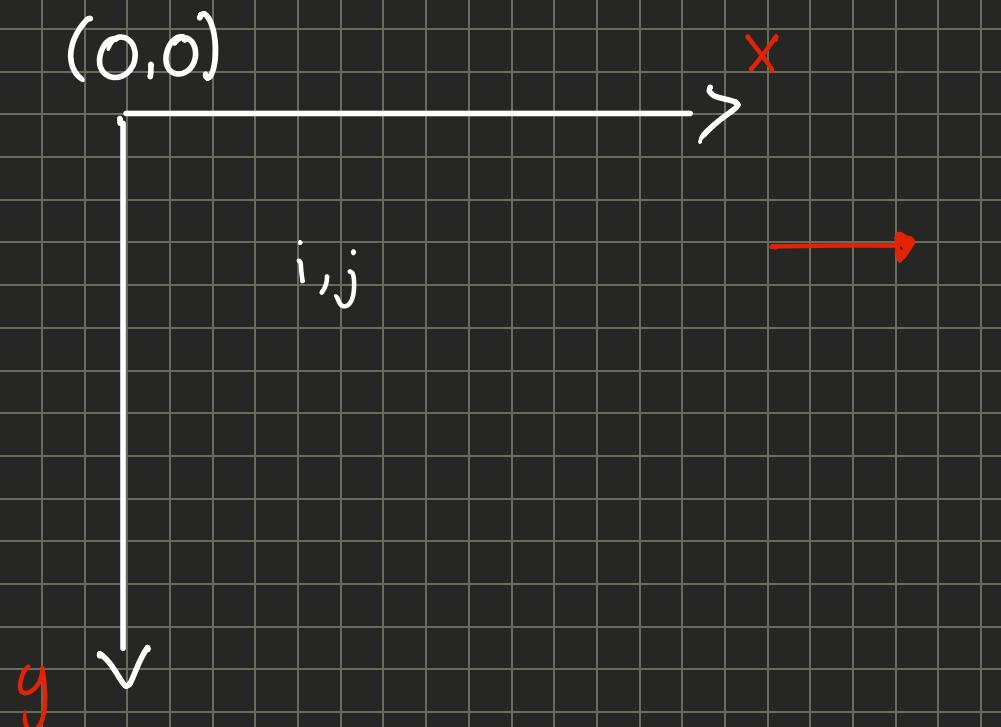
• 0 = NERO

• 0 o 1 = GRIGIO MEDIO

• 1 = BIANCO



## RAPPRESENTAZIONE:



BIANCO / NERO

1 bit per pixel e NEGLIA  
POSIZIONE  $i, j$  CI SARÀ O

0 o 1

SCALA DI GRIGIO

8 bit per pixel e  
NEGLIA POSIZIONE  $i, j$   
CI SARÀ UN VAL COMPU.  
 $[0, 255]$

A COLORI

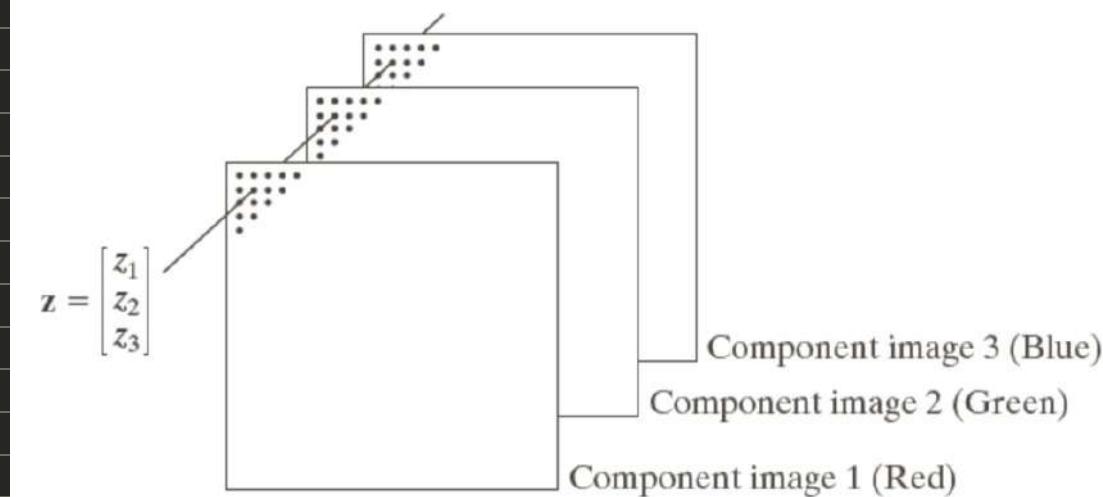
8 bit per CANALE (3)  
quindi 24 bit

IN  $i, j$  ci sarà UNA Terna  $(x, y, z)$   
CON VALORI COMPRESI  $[0, 255]$

ES:

LA TERNA RGB (RED GREEN  
BLUE)

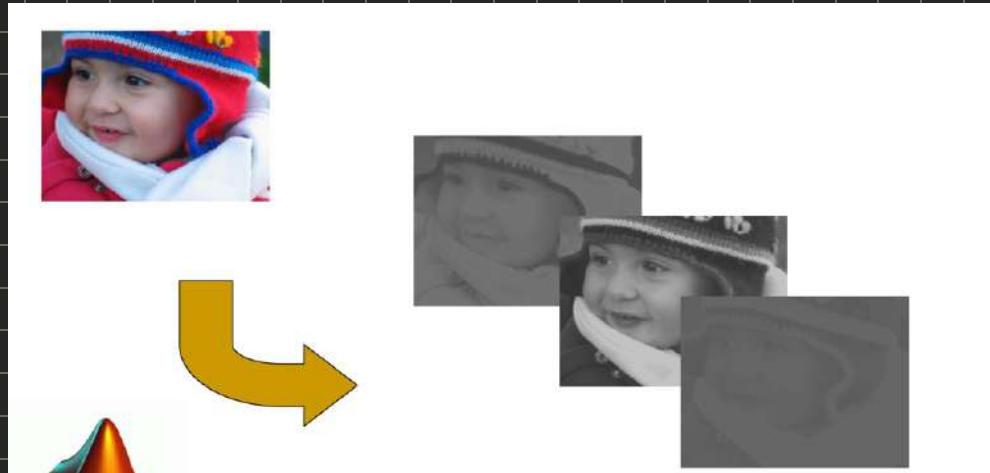
RGB ( RED, GREEN, BLUE )



3 MATRICI UNITE  
da cui da OGNIUNA si  
PRENDE UNA TONALITÀ

MA NON (come si pensa)  
ROSSO - VERDE - BLU

3 TONALITÀ che  
GRIGIO che  
COMPONGONO  
L'IMMAGINE o  
colori



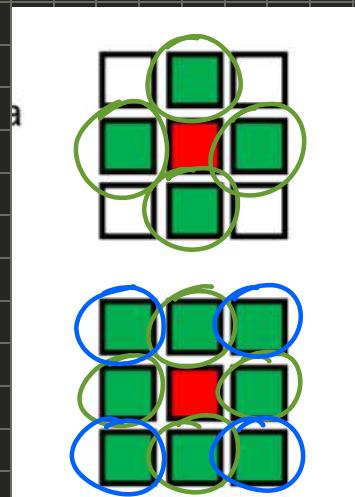
## PPT. 3 - OPERAZ. SU IMMAGINI e MATRICI

Per codificare un'immagine COME UNA MATRICE, deve operare un'immagine con OPERAZIONI TRA MATRICI

NON tutte le operazioni sono SENSESE, es. il PRODOTTO RIGA × COLONNA che  
PORTA UN RISULTATO IRRICONOSCIBILE

### Neighborhood ( $N_p$ )

- I vicini 9 connessi sono quelli alla DESTRA e SINISTRA e quelli SOPRA e SOTTO di un DATO PIXEL
- I vicini 8 connessi sono quelli 4 connessi + 4 in diagonale di un DATO PIXEL



### TRASFORMAZIONI AFFINI COS'È?

MODIFICA della POSIZIONE  
dei PIXEL di un IMMAGINE

Una trasformazione di RIMAPPAGGIO prende IN INPUT delle COORDINATE, tornando IN OUTPUT alle NUOVE COORDINATE. Si dice AFFINE se PRESEVA tutti i PARALLELSMI dell'IMMAGINE

## Tipi di OPERAZIONI AFFINI

### IDENTITÀ

~~non~~ modifica l' IMMAGINE (matrice)

$$T = \underline{\text{MATRICE IDENTITÀ}}$$

### TRASLAZIONE

SPOSTA i PIXEL sui DUE ASSI

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \underline{tx} & \underline{ty} & 1 \end{bmatrix}$$

### SHEAR VERTICALE

SPOSTAMENTO o DEFORMAZIONE che MODIFICA  
la POSIZIONE VERTICALE di CIASCUN PIXEL risp.  
alla SUA POSIZIONE ORIZZONTALE

$$T = \begin{bmatrix} 1 & 0 & 0 \\ \underline{sv} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

### SHEAR ORIZZONTALE

SPOSTAMENTO di PIXEL lungo l' ASSE  
ORIZZONTALE in base alla loro POSIZIONE  
sull' ASSE VERTICALE

$$T = \begin{bmatrix} 1 & \underline{sh} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# ROTAZIONE

$$[x \ y \ 1] \cdot [v \ w \ 1] = T$$

(0,0)

1	2	3
4	5	6
7	8	9

V

			(0,0) <sup>-1</sup>	
1	2	3	1	2
4	5	6	2	1
7	8	9	3	2

y

$$(0) \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

NOTA: OSNI VALORE X COLONNA

$$\begin{aligned}
 & \text{1^a colonna: } (0 \cdot 0) + (0 \cdot -1) + (1 \cdot 0) = 0 \\
 & \text{2^a colonna: } (0 \cdot 1) + (0 \cdot 0) + (1 \cdot 0) = 0 \\
 & \text{3^a colonna: } (0 \cdot 0) + (0 \cdot 0) + (1 \cdot 1) = 1
 \end{aligned}
 \rightarrow \boxed{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}}$$

$$(1) \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \cdot T$$

$$= \begin{array}{cccc} \backslash & \backslash & \backslash & \backslash \end{array} = \boxed{\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}}$$

$$(2) \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 1 \end{bmatrix} \cdot T$$

$$= \begin{array}{cccc} \backslash & \backslash & \backslash & \backslash \end{array} = \boxed{\begin{bmatrix} 0 & 2 & 1 \end{bmatrix}}$$

$$(-1) \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \cdot T$$

$$= \begin{array}{cccc} / & / & / & / \end{array} = \boxed{\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}}$$

$$T = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\theta = \frac{\pi}{2}$$

$$T = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

## FORWARD e BACKWARD MAPPING

IN AMBITO di IMAGE PROCESSING, esistono due diversi approcci alla TRASFORMAZIONE delle IMMAGINI (funzione di RIMAPPAGGIO)

- FORWARD MAPPING

Si scorre ogni pixel in INPUT, tramite la matrice  $T$  per calcolare la nuova posizione dei pixel in OUTPUT

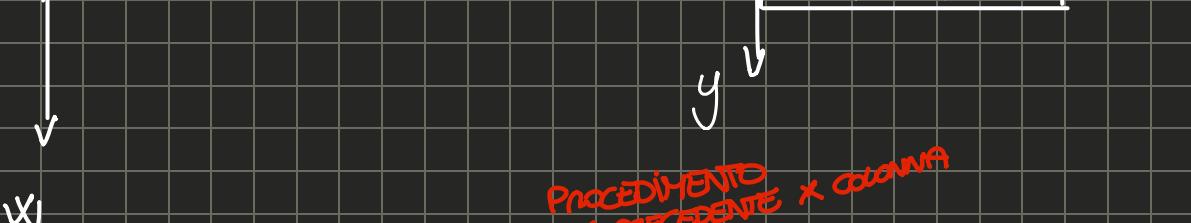
- BACKWARD (INVERSE) MAPPING

Si scorre ogni pixel in OUTPUT, cambiando il pixel corretto dall'immagine di INPUT, basandosi su  $T^{-1}$

# INVERSE MAPPING (SCALING)

VISITA I PIXEL DELL'OUTPUT  $(x, y)$  E CALCOLA LE COORDINATE NELL'IMMAGINE DI INPUT

1	2	3
4	5	6
7	8	9



PROCEDIMENTO  
PPG. PRECEDENTE \* COLONNA

$$\text{ES: } [v \ w \ 1] = [0 \ 0 \ 1] \cdot \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \boxed{[0 \ 0 \ 1]} (1)$$

$$\text{ES: } [v \ w \ 1] = [1 \ 0 \ 1] \cdot \text{INV}(T)^* = \boxed{[0 \ -1 \ 1]} (2)$$

(x) -1 quindi DIVIDI FARE STIMA (METT' UNA X)

$$T = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(3) [v \ w \ 1] = [2 \ 0 \ 1] \cdot \text{INV}(T) \ " " = \boxed{[0 \ -1 \ 1]}$$

PROCEDIMENTO RIGA

$$(4) [v \ w \ 1] = [0 \ 1 \ 1] \cdot \text{INV}(T) \ " " = \boxed{[1 \ 0 \ 1]}$$

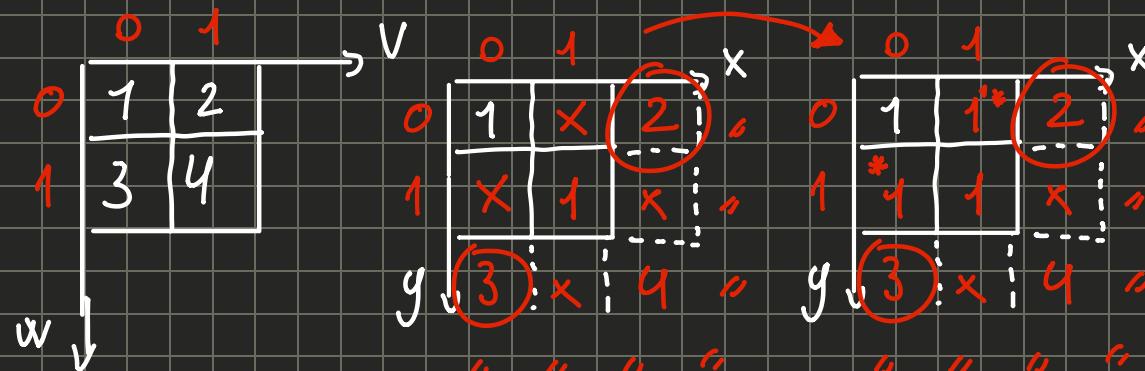
\*2

## SCALING

$$T = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{INV}(T) = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

INVERSE  
MAPANG



$$[\begin{matrix} x & y & 1 \end{matrix}] = [\begin{matrix} 0 & 0 & 1 \end{matrix}] \cdot \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = [\begin{matrix} 0 & 0 & 1 \end{matrix}]$$

$$[\begin{matrix} x & y & 1 \end{matrix}] = [\begin{matrix} 1 & 0 & 1 \end{matrix}] \cdot \dots = [\begin{matrix} 2 & 0 & 1 \end{matrix}] (2) \quad \text{FORWARD MAPPING}$$

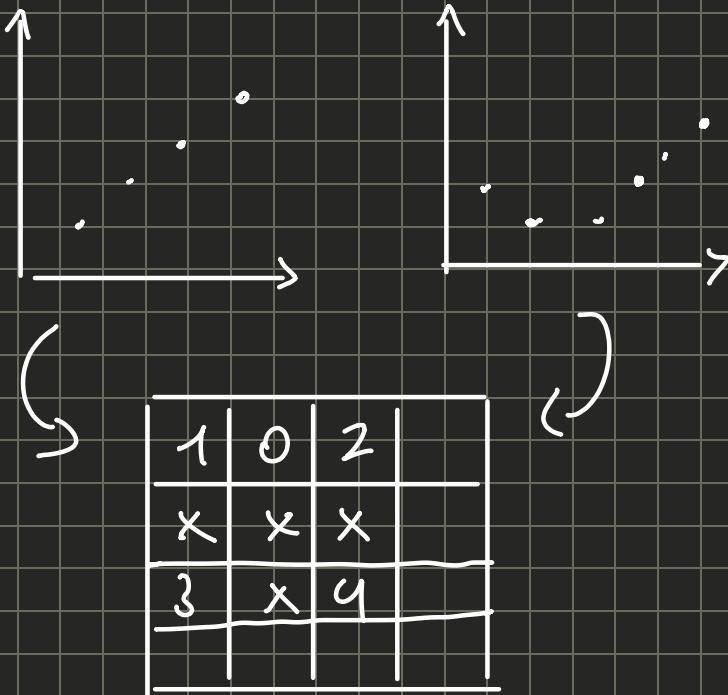
$$[\begin{matrix} x & y & 1 \end{matrix}] = [\begin{matrix} 0 & 1 & 1 \end{matrix}] \cdot \dots = [\begin{matrix} 0 & 2 & 1 \end{matrix}] (3) \quad \text{FORWARD MAPPING}$$

$$[\begin{matrix} v & w & 1 \end{matrix}] = [\begin{matrix} 1 & 0 & 1 \end{matrix}] \cdot \text{INV}(T) = [\begin{matrix} 0,5 & 0 & 1 \end{matrix}] *^1 \quad \text{INVERSE MAPPING}$$

$$[\begin{matrix} v & w & 1 \end{matrix}] = [\begin{matrix} 0 & 1 & 1 \end{matrix}] \cdot \text{INV}(T) = [\begin{matrix} 0 & 0,5 & 1 \end{matrix}] *^1 \quad \text{POUR INVERSE MAPPING}$$

(deuxième ligne) \* 0,0 C'E  
1 au pos 0,1

## INTERPOLAZIONE



TIROVARE VALORI MANDANTI  
BASANDOCI SU QUELLI PRESENTI



l' IMMAGINE **NON** viene MIGLIORATA  
ma **RACQUISICE** i valori MANDANTI  
FACENDO LA **STIMA** dei VAL. IGNOTTI

## ZOOMING IN

L' INTERPOLAZ. è effettuata nei processi ZOOMING

TIPI di INTERPOLAZIONE :

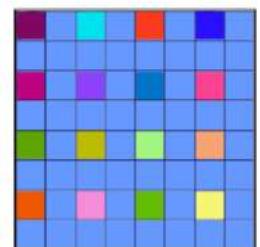
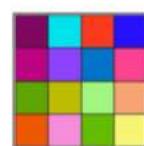
- NEAREST NEIGHBOR (REDUCTION)
- BILINEAR • BICUBIC • ALTRI...



## ZOOMING 2x



Zooming in (2x)



Dopo aver posizionato i valori già noti, occorre stimare i valori nelle zone vuote.

## REPLICACIÓN

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

$2 \times$   
—  
7

1	1	2	2	3	3	4	4
1	1	2	2	3	3	9	9
5	5	6	6	7	7	8	8
5	5	6	6	7	7	8	8
8	8	10	10	14	14	12	12
8	8	10	10	14	14	12	12
13	13	14	14	15	15	16	16
13	13	14	14	15	15	16	16

Questo metodo assegna a ogni nuova posizione l'intensità del pixel più prossimo nell'immagine originale.

Questo approccio è molto semplice ma introduce artefatti come distorsioni lungo i lati degli oggetti rappresentati nell'immagine.

## BILINEAR

	0	1
0	10	20
1	30	40

	0	1	2	3
0	10	15	20	25
1	20	25	30	35
2	30	35	40	
3				

STIMIAMO IL VALORE PER  
CONTINUITÀ (da 5 a 10)  
MA NON SEMPRE UTILIZZATO

$$V(x,y) = ax + by + cxy + d$$

$$V(x,y) = 5x + 10y + 10$$

TEST (SOSTITUZIONE)

$$V(0,0) = 10$$

$$V(2,0) = 20$$

$$V(0,2) = 30$$

$$V(2,2) = 40$$

STIMIAMO \*

$$V(1,1) = 5 + 10 + 10 = 25$$

$$V(1,0) = 15 / V(0,1) = 20$$

$$V(2,1) = 30 / V(1,2) = 35$$

1

$$\left\{ \begin{array}{l} V(0,0) = \\ V(2,0) = \\ V(0,2) = \\ V(2,2) = \end{array} \right.$$

## BICUBIC

L'interpolazione bicubica utilizza **sedici pixel** più vicini al punto. Il valore di intensità assegnato al punto  $(x, y)$  si ottiene attraverso l'equazione

$$V(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

Le operazioni dunque sono  
attraverso POTENZA

Cosa fanno ai bordi?

- 2 tipi :

1) NON FADE NIENTE



Considerare solo le zone centrali dell'immagine.

- In questo caso non si calcolano i valori ai bordi

input			output		
1	2	3			
4	5	6			
7	8	9			

1	1.5	2	2.5	3	
2.5	3	3.5	4	4.5	
4	4.5	5	5.5	6	
5.5	6	6.5	7	7.5	
7	7.5	8	8.5	9	

2) INTERPOLAZIONE CON I VALORI PRESENTI



Dopo aver fatto i calcoli replicare le ultime righe e colonne.

- In questo caso per fare i calcoli si replicano i valori nelle righe e nelle colonne «isolate»

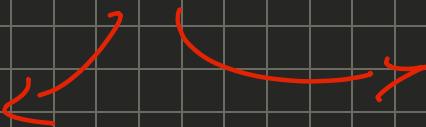
input			output		
1	2	3	1	1.5	2
4	5	6	2.5	3	3.5
7	8	9	4	4.5	5

1	1.5	2	2.5	3	3
2.5	3	3.5	4	4.5	4.5
4	4.5	5	5.5	6	6
5.5	6	6.5	7	7.5	7.5
7	7.5	8	8.5	9	9

## ZOOMING OUT

- NUMERO INFERIORE DI 1, SI OTT. UNA IMMAGINE PIÙ PICCOLA tramite IL PROCESSO di DECIMAZIONE

METODO 1 =



### Decimazione: metodo 1

- Ogni quattro pixel se ne sceglie uno.

input		output	
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	3
9	11



### Decimazione: metodo 2

- Di quattro pixel se ne calcola il valore medio.

input		output	
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

3	5
11	13

## QUALITÀ di UN ALGORITMO

MSE (Mean Square Error) : Serve per stimare l'ERRORE QUADRATICO medio tra due IMMAGINI; Più BASSO, MINORE DIFFERENZA di IMMAGINI

PSNR (Peak Signal to Noise Ratio) : MISURA la qualità di un'IMMAGINE COMPRESA rispetto all'ORIGINALE; Più ALTO, MAGGIORE SIMIGLIANZA di IMMAGINI

- Full REFERENCE
- CALCOLO TRA  $I^*(M \times N)$  e  $I(M \times N)$  -  
IMMAGINE DA VALUTARE      IMMAGINE ORIGINALE
- NON è il parametro migliore per valutare un algoritmo

Come calcolarlo?

Vi sono 3 FORMULE : \*

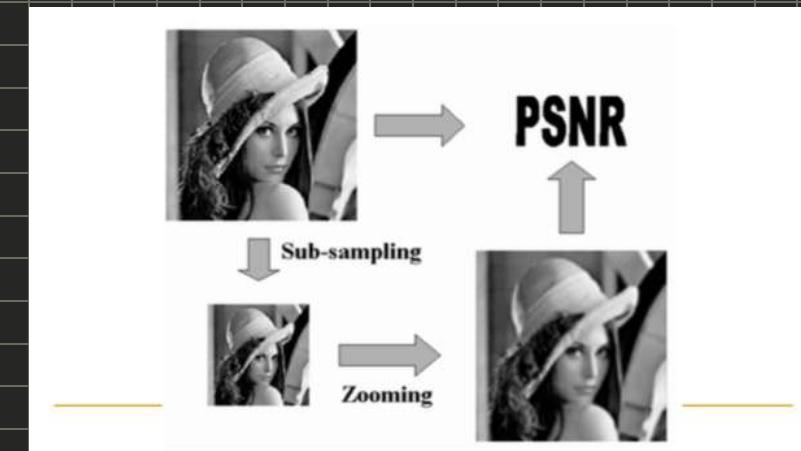
$$PNSR = 10 \log_{10} \frac{MSE}{S^2}$$

$$PNSR = 20 \log_{10} \left( \frac{S}{\sqrt{MSE}} \right)$$

$$PNSR = 10 \log_{10} \left( \frac{S^2}{MSE} \right)$$

da cui come  
RISULTATO un  
DECIBEL  
→

S è il MAX VALORE  
dei PIXEL = 255



## CASO PEGGIORIS PSNR

$$MSE(G, G') = \frac{1}{q} \sum \sum (G' - G)^2 = \frac{1}{q} (255^2) = \boxed{255^2} / 255^2, \text{ quindi } log_10 = 0$$

G	G'
0   0	255   255
0   0	255   255

calcolo PSNR (FOORMA  $\frac{1}{\sqrt{MSE}}$ )

$$PSNR(I, I') = 10 \log_{10} \frac{255^2}{MSE} = 95,7 \underline{\text{duabel}}$$

I	I'
100   100	101   201
10   30	12   28

NON AMMESSO! Quando MSE tende a 0, PSNR tende a +∞ poiché le immagini sono uguali

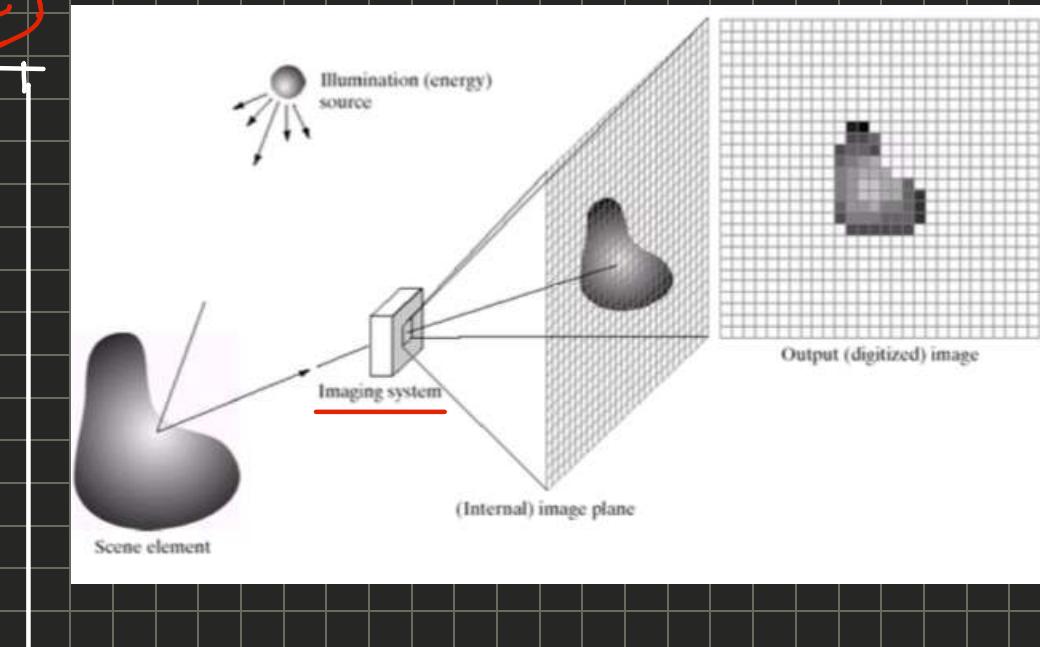
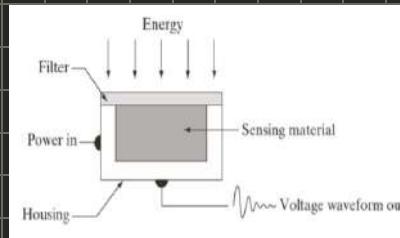
$$\begin{aligned} MSE(I, I') &= \frac{1}{q} \sum_{x=1}^2 \sum_{y=1}^2 [I'(x, y) - I(x, y)]^2 \\ &= \frac{1}{4} [(101-100)^2 + (201-200)^2 + (12-10)^2 + (28-30)^2] \\ &= \frac{1}{4} (1+1+4+4) = \boxed{\frac{7}{4}} \end{aligned}$$

## PDF 4 - ACQUISIZIONE delle immagini digitali

Quando la luce colpisce un oggetto, una parte viene ASSORBITA, l'altra RIFLESSA che dà origine al Colore

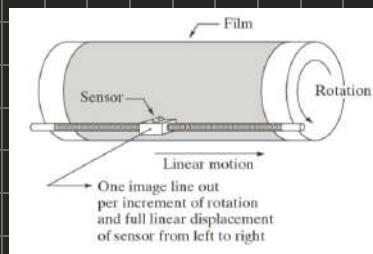
### - SENSORE

L'energia che colpisce il sensore è trasformata in impulso elettrico dal sensore che è fatto di un materiale particolarmente sensibile alla luce. Tale impulso elettrico è successivamente digitalizzato.



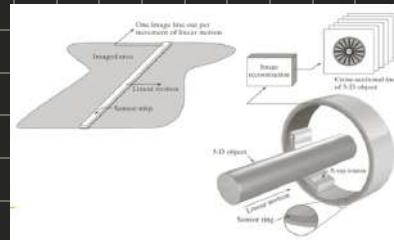
### - SINGOLO SENSORE

Gli scanner usano un singolo sensore che viene spostato lungo la sorgente da digitalizzare.



### - SENSORI IN LINEA

Le macchine per fare le TAC usano dei sensori che sono disposti in un'unica linea.



### - SENSORI IN 2D array

Nelle macchine fotografiche digitali, i sensori sono disposti su una matrice.

- Non è necessario spostare il sensore, come nei casi precedenti, per effettuare una scansione.
- I più diffusi sensori di questo tipo sono i CCD

#### CCD : charged coupled Device

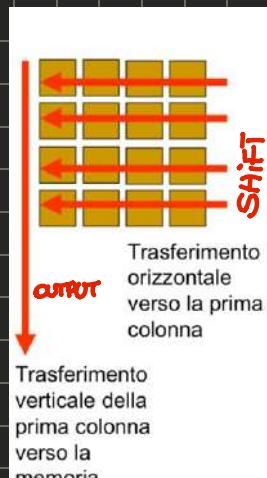
Dispositivi elettronici che capti i光子 -> CARICA POSITIVA

- NON POSSONO CARICARSI oltre UN LIMITE (si creerebbe una SOPRA-SATURAZIONE)

IL NUMERO DI CELLE È MISURATO in MEGAPIXEL

Dopo che le cariche sono state acquisite debbono essere trasferite in una memoria digitale tramite 3 fasi:

- Viene trasferita in memoria la prima colonna della matrice, nello stesso tempo tutti gli elementi (dalla seconda colonna in poi) vengono trasferiti dalla propria colonna a quella precedente



## CFA: COLOR FILTER ARRAY

Ogni cella memorizza solo 1 colore per volta e non una Terna, quindi occorre scegliere il Modello di memorizzazione ottimale (CFA).

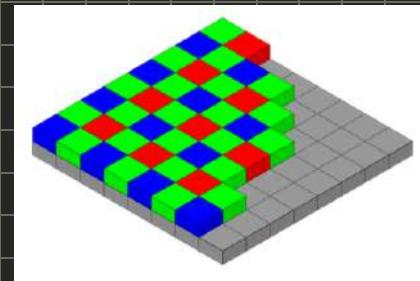
I 2 colori mancanti saranno ottenuti con INTERPOLAZIONE dai pixel vicini (color interpolation)

Il grado di accuratezza del risultato dipende da quanto è sofisticato il metodo di interpolazione.

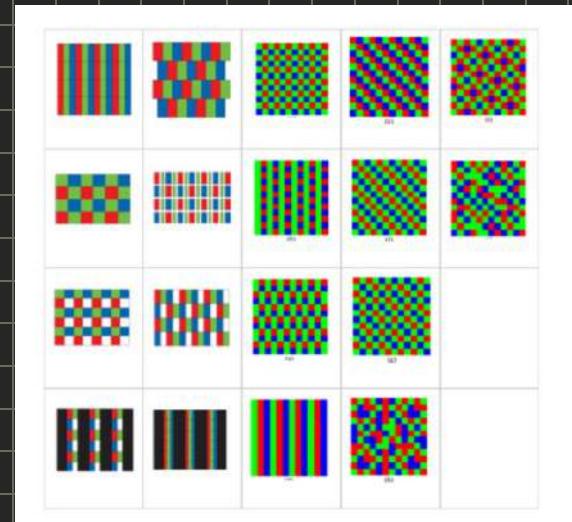
### - BAYER PATTERN

Presento un rapporto  $1:2:1$  fra R:G:B, dove i pixel verdi sono disposti sulla diagonale.

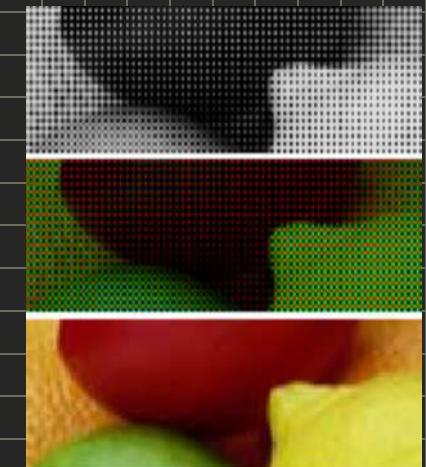
Il risultato di una cattura è un FORMATO RAW, deve essere poi INTERPOLATO per ottenere il colore effettivo.



### Possibili modelli di CFA



### Color Interpolation



FALSE  
color

TRUE  
color

Esistono numerosi algoritmi usati nell'ambito della color interpolation.

NEAREST-NEIGHBOUR  
INTERPOLATION

INTERP.  
BILINEARE

INTERPOLAZ.  
BICUBICA

Dato un qualsiasi pixel, sarà solitamente noto uno dei suoi 3 valori RGB. Gli altri possono essere ottenuti tramite interpolazione. In questo modo, otterremo una terna di valori che indica il true-color.

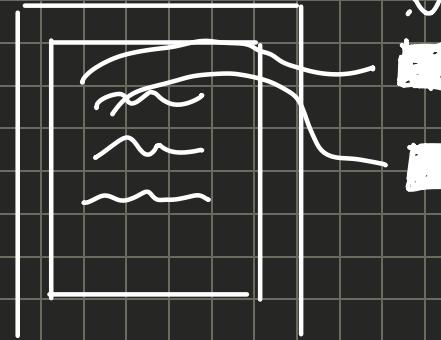
## PDF 5 - La RISOLUZIONE

Il numero di pixel per u. di MISURA →  
e indica il GRADO di QUALITÀ di un'IMAG.

Si può misurare in  
CENTIMETRI, DOTS, POLlicI  
...

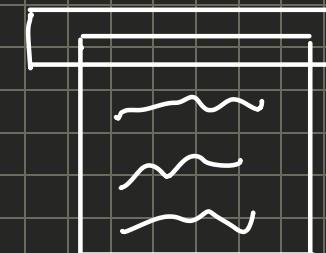
### RISOLUZIONE dell'apparecchiatura di RIPRESA

- SCANNER fino a 6000 dpi e OLTRE
- FOTOCAMERE che si misurano in **MEGAPIXEL**



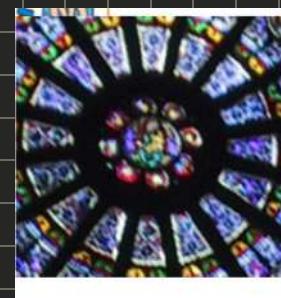
### RISOLUZIONE dell'apparecchiatura di RIPRESA

- STAMPANTE fino a 3000 dpi e OLTRE
- SCHERMI: m. di elementi sullo SCHERMO (f2 dpi)  
per u. di MISURA



## MISURAZ. PIXEL

1 dot × 1 cm



200pixel (1cm)

1 dot × 2 cm

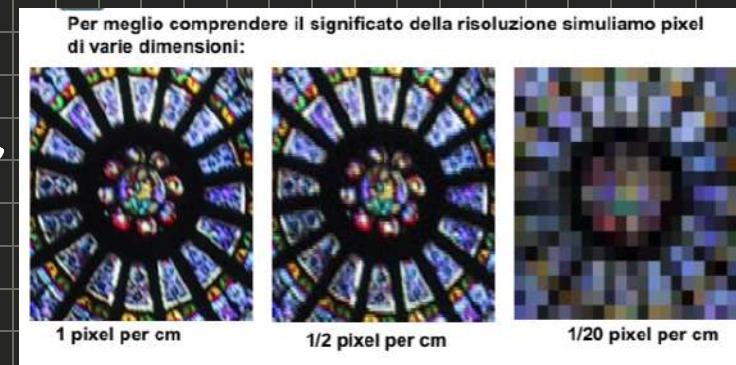


100pixel (2cm)

1 dot × 20 cm



10pixel (20cm)



# IMPORTANTE !!

IMMAGINI NATE CON UNA RISOLUZIONE, NON AVERE LA MAXIMA RISOLUZIONE dentro perché VISUALIZZ. CON LA RISOLUZIONE ORIGINALE



Sigla	Definizione	Risoluzione	Rapporto
Quarter QVGA	QQVGA	160 x 120	4:3
CGA o Quarter VGA	CGA / QVGA	320 x 240	4:3
Half VGA	HVGA	640 x 240	8:3
Monochrome Display Adapter	MDA	720 x 350	
Enhanced Graphics Array	EGA	640 x 350	
Video Graphics Array	VGA	640 x 480	4:3
Super VGA	SVGA	800 x 600	4:3
Quad VGA	QVGA	1280 x 960	4:3
Extended Graphics Array	XGA	1024 x 768	4:3
Super XGA	SXGA	1280 x 1024	5:4
Super XGA Plus	SXGA+	1400 x 1050	4:3
Ultra XGA	UXGA	1600 x 1200	4:3
Quad XGA	QXGA	2048 x 1536	4:3
Quad Ultra XGA	QUXGA	3200 x 2400	4:3
(*) Wide XGA	WXGA	1366 x 768	16:9
(*) Wide XGA	WXGA	1280 x 800	16:10
Wide XGA Plus	WXGA+	1440 x 900	16:10
Wide Super XGA Plus	WSXGA+	1680 x 1050	16:10
Wide Ultra XGA	WUXGA	1920 x 1200	16:10

SI È PASSATO A  
16:9 PER OTTIMIZZARE  
LA VISTA UMANA  
(PIÙ GRADELLI)

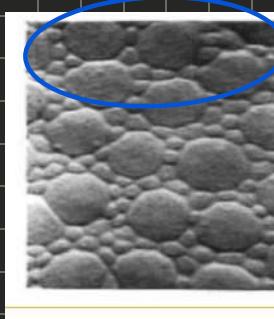
## PDF 6 - L'occhio

IN PARTICOLARE la RETINA, è una MEMBRANA che RICOPRE la PARTE POSTERIORE dell'occhio

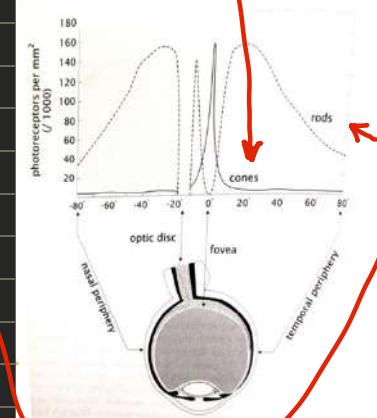
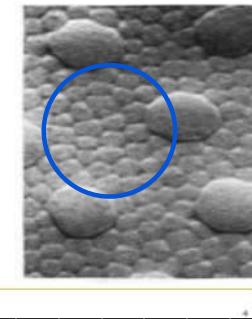
- FORMATA da

- CONI → SONO 6/7 MILIONI e sono in una ZONA CENTRALE, la FOVEA. Sono SENSIBILI e COLLEGATI ai NERVI OTTICI, rispondono alla VISTA FOTOTICA

- BASTONCELLI → SONO 75/150 MILIONI e sono OVUNQUE. Sono POCO SENSIBILI e COLLEGATI ai GRUPPI di NERVI OTTICI, rispondono alla VISTA SCOTOPICA



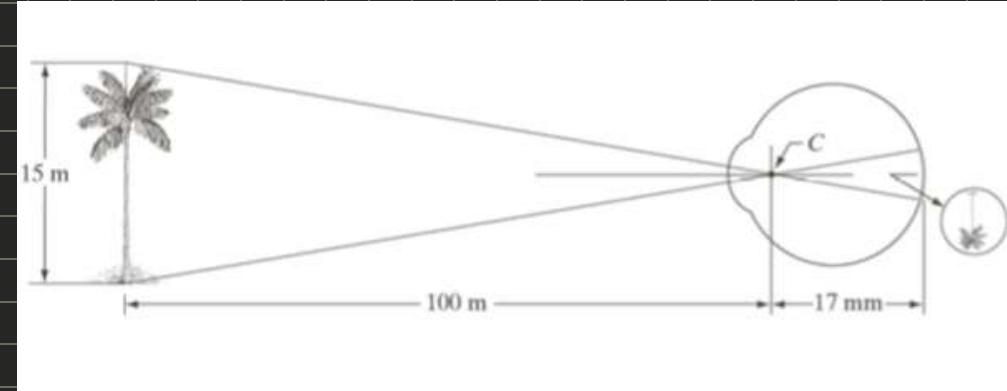
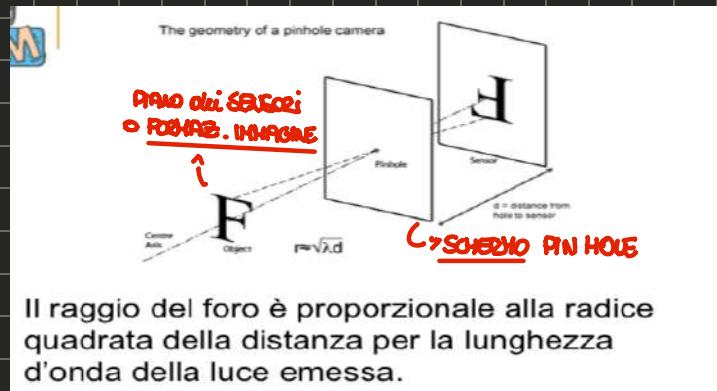
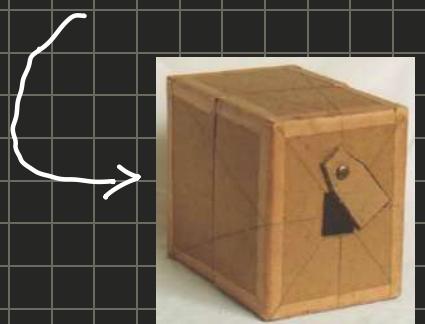
DISPOSIZIONI:



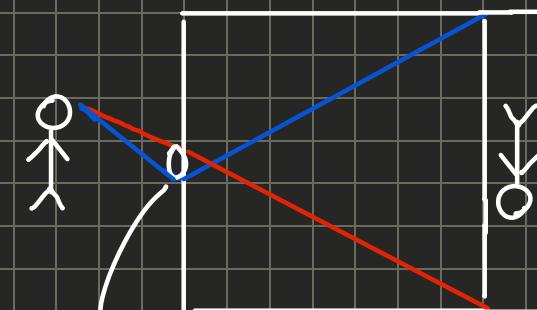
# COME SI FORMA L'IMMAGINE?

## - MODELLO DEL PINHOLE

IN CUI SI APPOGGIA L'OCCHIO CON UNA SCATOLA, AL CUI INTERNO VIENE POSIZ.  
UNA PELLICOLA SENSIBILE ALLA LUCE E  
ALL'OPPOSTO VI È UN FORO (PINHOLE)



RAPPRESENT.  
CON PINHOLE



CRISTALLO che  
DEVIA LA LUCE o  
cazione che è  
**PENSABILE**, quindi può  
essere **DEFORMATO**

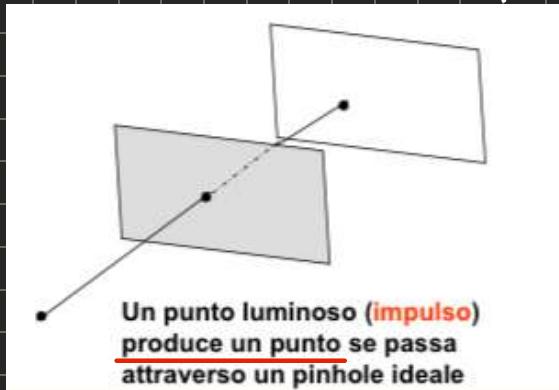
- INVECE una **FOTOCAMERA**  
lavora con dei **SENSORI** che  
**SI MOVONO**, perché **NON FESS**

## MESSA A FUOCO

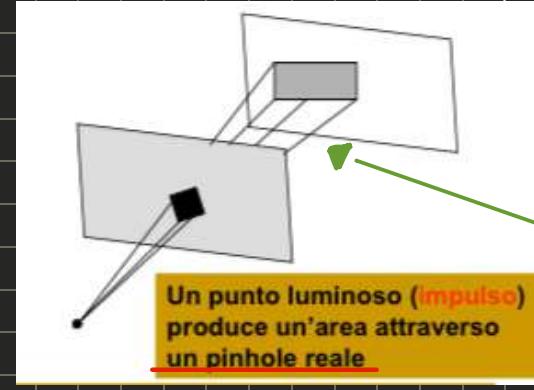
Def INS. di PUNTI che sto ATTIVANDO  
VADO A FINIRE SUL OGGETTO che  
STO PUNTANDO, quindi ci deve essere.  
UNA CONVERGENZA di LUCE

## - Tipi di PINHOLE

### PINHOLE IDEALE (PUNTO)



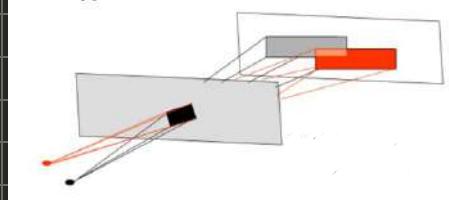
### PINHOLE REALE (AREA)



POINT SPREAD FUNCTION  
(spargesi dei punti luminosi)

### Princípio di SOVRAPPOSIZIONE

L'effetto di più sorgenti puntiformi vicine si "sovrappongono":

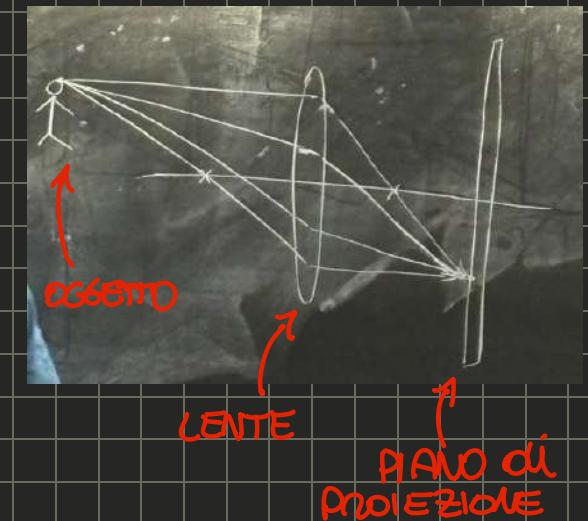
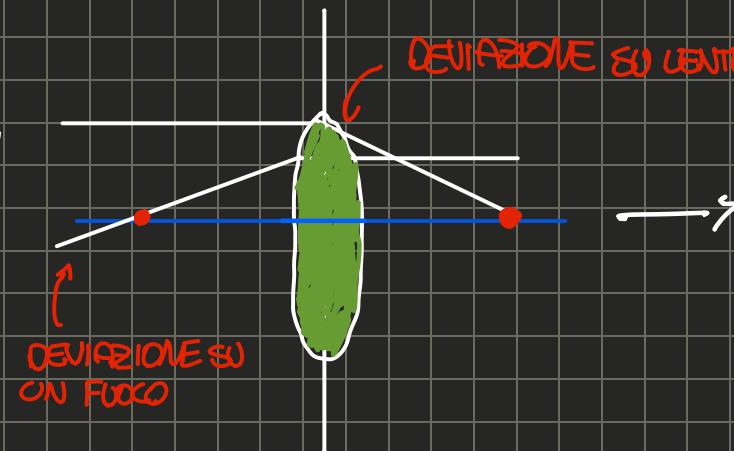


## LENTI SOTTILI

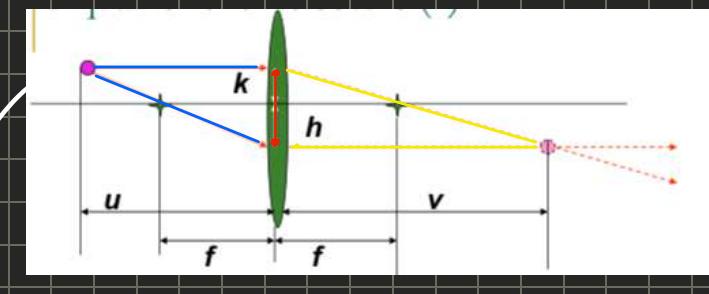
Chiamate "SOTTILI" perché il DIAMETRO  
è MOLTO PIÙ GRANDE risp alla SPESSEZZE della  
LENTE.

### - CARATTERISTICHE:

2 FUOCHI opposti sull' ASSE OTTICO e  
solo SIMMETRI su fuoco, ad eccez.



# Equazione lente sottile



nel triangolo di BASE  $uh$  e altezza  $h+k$  e quello di BASE  $f$  e ALTEZZA  $h$  (sulla SINISTRA della LENTE), da cui

$$\frac{uh}{f} : (h+k) = f : h \text{ da cui si ricava } (h+k) = uh/f$$

nel triangolo di BASE  $vK$  e altezza  $h+k$  e quello di BASE  $f$  e ALTEZZA  $K$  (sulla DESTRA della LENTE), da cui

$$\frac{vK}{f} : (h+k) = f : K \text{ da cui si ricava } (h+k) = vK/f$$

Eguagliando le due relazioni ed eliminando  $f$  si ottiene che  $h/v = k/u$ . Ora da  $(h+k) = uh/f$  possiamo dividere ambo i membri per  $u$  ed otteniamo  $h/u + k/u = h/f$  cioè ancora sostituendo  $h/u + h/v = h/f$  da cui eliminando il fattore comune  $h$  si giunge a  $1/u + 1/v = 1/f$ . equazione della lente sottile.

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f}$$

a)

In una lente "fissa" la quantità  $f$  è costante. Se la distanza dell'oggetto dalla lente, cioè  $u$ , cresce, per la relazione di cui sopra  $v$  non può che diminuire:

la

messaggio a fuoco richiede che il piano dei sensori possa essere avvicinato o allontanato dalla lente.

b)

Se il piano dei sensori non può essere mosso (retina umana!) l'unica cosa da fare è aggiustare la lunghezza focale (cioè fanno i muscoli che mantengono in tensione il cristallino).

La variabilità della lunghezza focale della lente si misura in dioptri

c)

Se due oggetti sono a distanza  $u_1$  e  $u_2$  e entrambe queste quantità sono molto maggiori di  $f$ : essi formano le loro immagini approssimativamente su un unico piano (i due valori corrispondenti  $v_1$  e  $v_2$  sono vicinissimi).

Se  $u_1$  e  $u_2$  sono però differenti e comparabili, allora essi NON possono essere focalizzati contemporaneamente: si manifesta il fenomeno della "profondità di campo" che risulta più accentuato se  $f$  è grande

# Esempio di MAGNIFICAZIONE

Def Riguarda l' **AMPLIFICAZIONE** o **POTENZIAMENTO** di un elemento o dell' **IMMAGINE** per migliorare l'esperienza visiva o COMUNICAZIONE di un **CONCETTO**, data da:

$$f = u/m / (m+1)$$

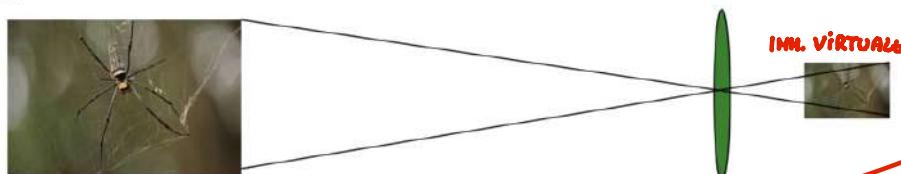
\*

con

$$m = v/u$$

Es:

Esempio: Un oggetto (il ragno) distante  $u=1\text{cm}$  è fotografato da una macchina digitale con "chip ccd" distante  $v=0.5\text{cm}$ . A quanto si deve fissare il fuoco per potere avere lo stesso effetto su una farfalla che si trova a  $u'=500\text{cm}$  dalla lente?



Il fattore di magnificazione è  $m=0.5/1=0.5$

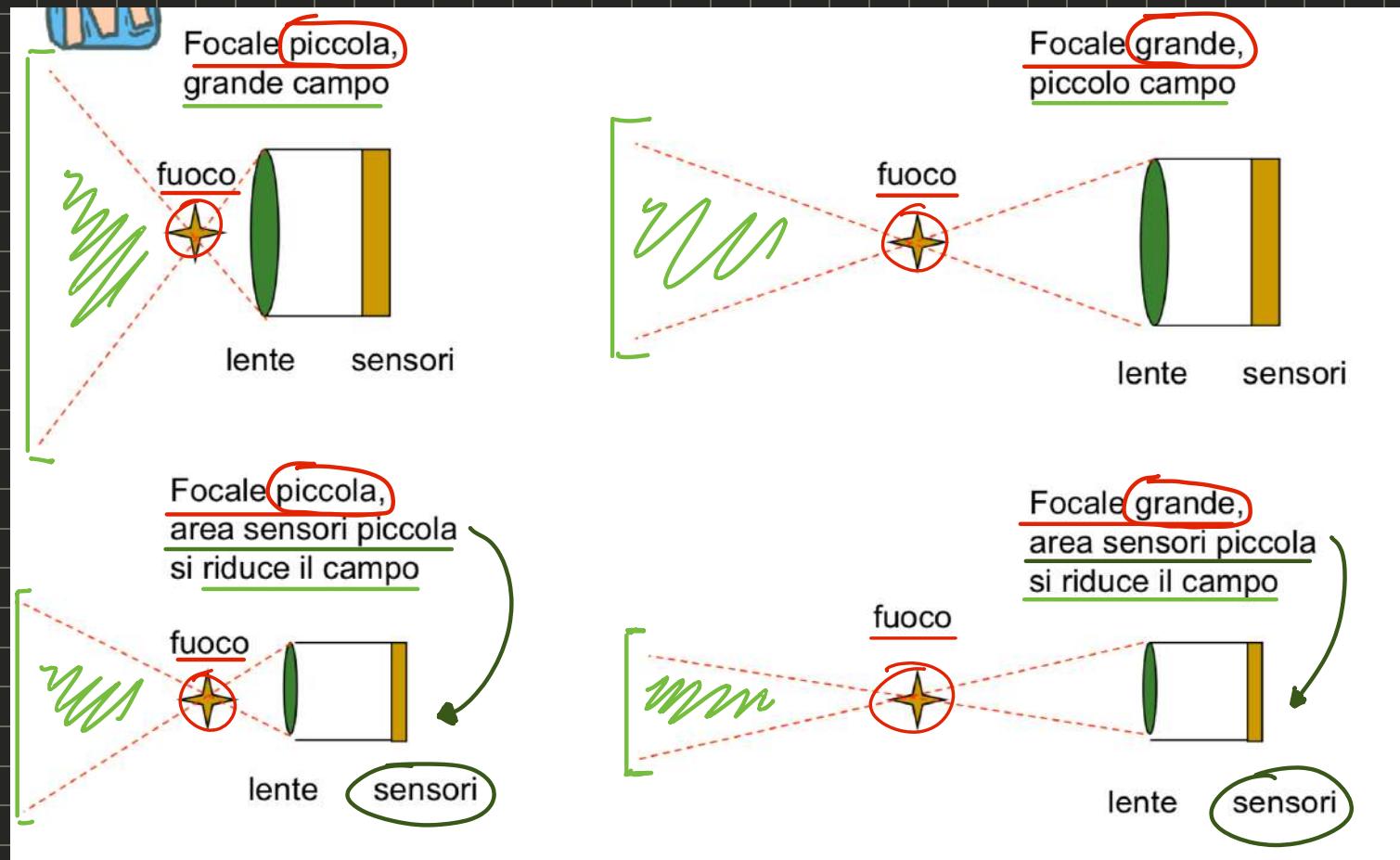
\*  $f = 500 * 0.5 / 1.5 = 166.66 \text{ cm}$  (un teleobiettivo con una focale di più di un metro mezzo non è esattamente un oggetto comodo da portare nelle passeggiate nel bosco).

)

ALTRI ESEMPI:

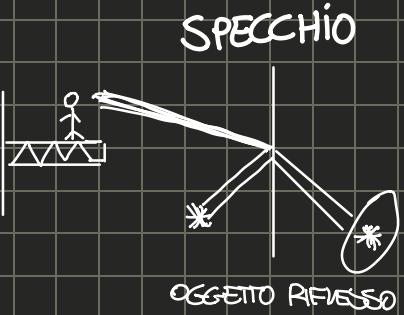
- **Altro scenario:** l'oggetto è sempre ad una distanza  $u=1\text{cm}$  e si usa una macchina analogica con  $v=2\text{cm}$ . Se la farfalla è a distanza  $u'=5\text{ cm}$  si ha che  $m=2/1=2$  e  $f=5*2/3=3.3 \text{ cm}$
- **Ultimo scenario:** l'oggetto è sempre ad una distanza  $u=1\text{cm}$  e si usa una macchina digitale con CCD con  $v=1\text{cm}$ . Se la farfalla è a distanza  $u'=5\text{ cm}$  risulta  $m=1$  e  $f=5*1/2 = 2.5 \text{ cm}$ .

# AMPIEZZA di CAMPO e FOCALE



Distanza FOCALE  
Distanza tra la  
LENTE e il FUORI

ES

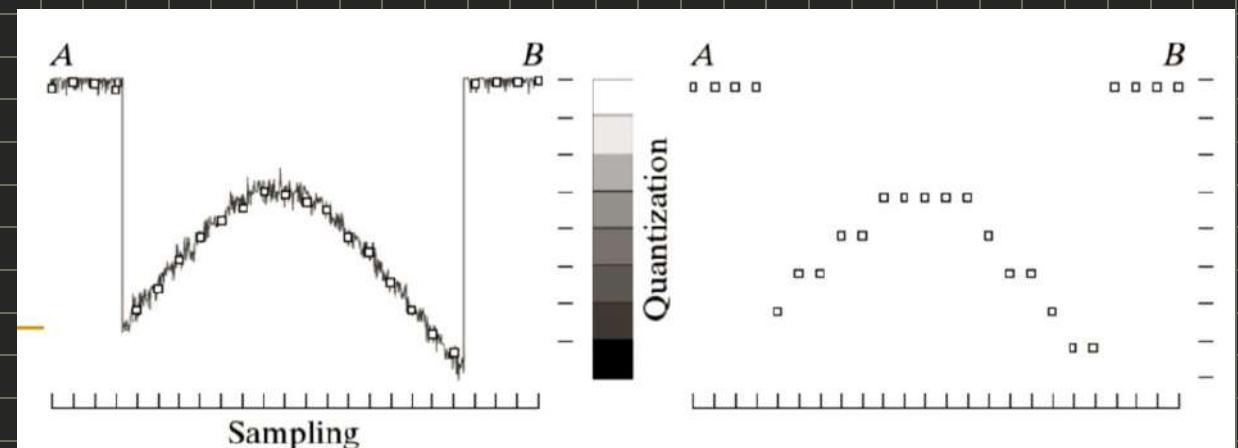
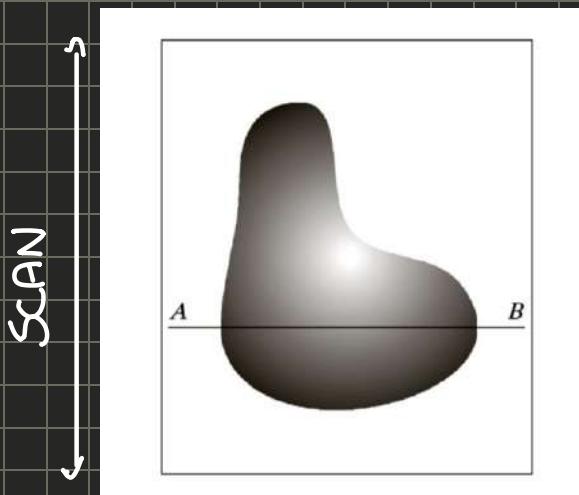


SENSORI: CATTURA la  
LUCE e TRASFORMA un  
IMMAGINE

## PDF 7 - CAMPIONAMENTO e QUANTIZZAZIONE

DA VALORE CONTINUO O DISCRETO, quindi DIGITALIZZIAMO l'IMMAGINE e rappresentiamola correttamente sul SEGNALE inviando i VARI CANALI

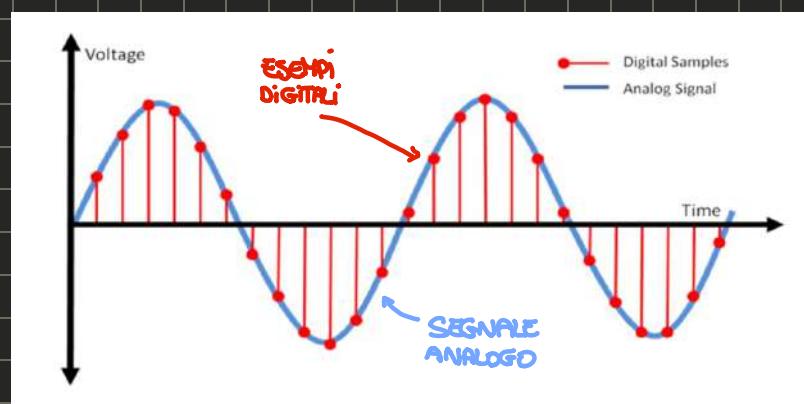
### IMMAGINE ESEMPIO



CAMPIONAMENTO  
(SEGMENTO del TAB.)

QUANTIZZAZIONE  
(PUNTINI)

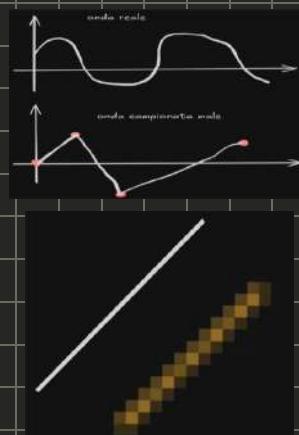
### CAMPIONAMENTO - ESEMPIO



- ERRORE CAMPIONAMENTO !!

a) UN CAMPIONAMENTO BASSO fa PERDERE DETTAGLI ed INFORMAZIONI

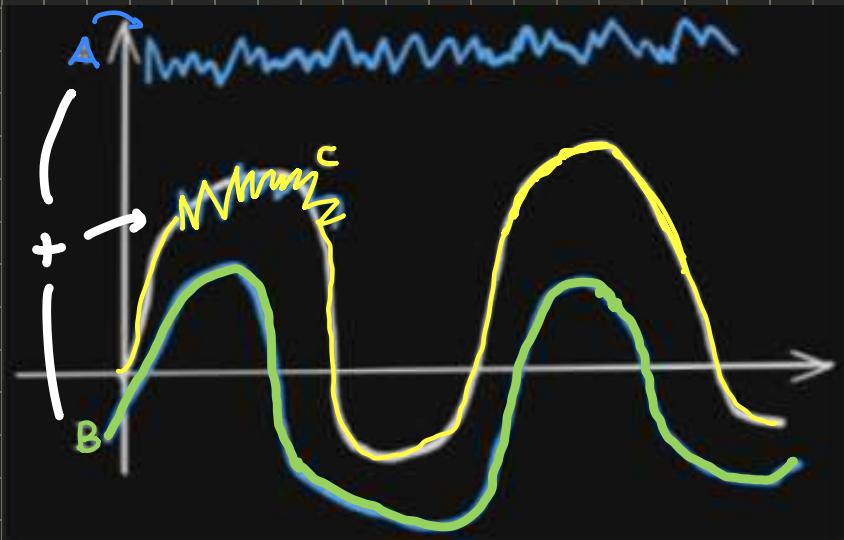
b) UN CAMPIONAMENTO BASSO può far ANNULARE dettagli non presenti PRECEDENTEMENTE. Per questo si parla di "aliasing"



## NYQUIST RATE (HARRY NYQUIST, 1928)

Def

Si definisce **Nyquist rate** il **doppio della più alta frequenza** in un segnale continuo e limitato.

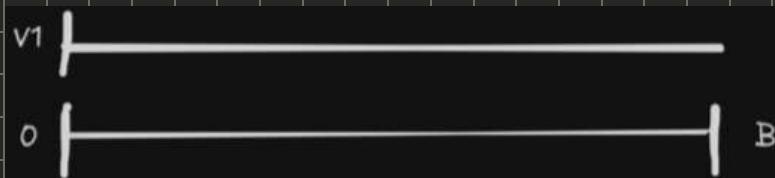


Esempio di N.R

In particolare la somma della FREQUENZA più ALTA (A) e la CURVA (B) dà come risultato la curva + FREQUENZA + ALTA (C)

Altri esempi di N.R

- In quanti PARTI OVR suddividere per rappresentare il VALORE da 0 a  $B^?$  = ①



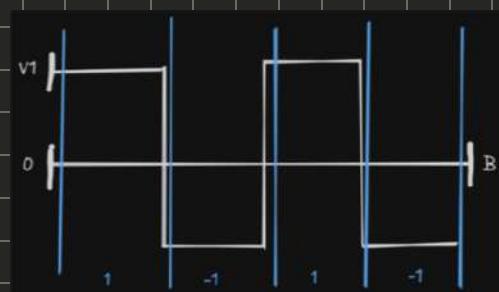
1 SEGMENTO

- Se il VALORE è su 2 PIANI (SAE e SCENDE) DEVO suddivid. in ②



2 SEGMENTI

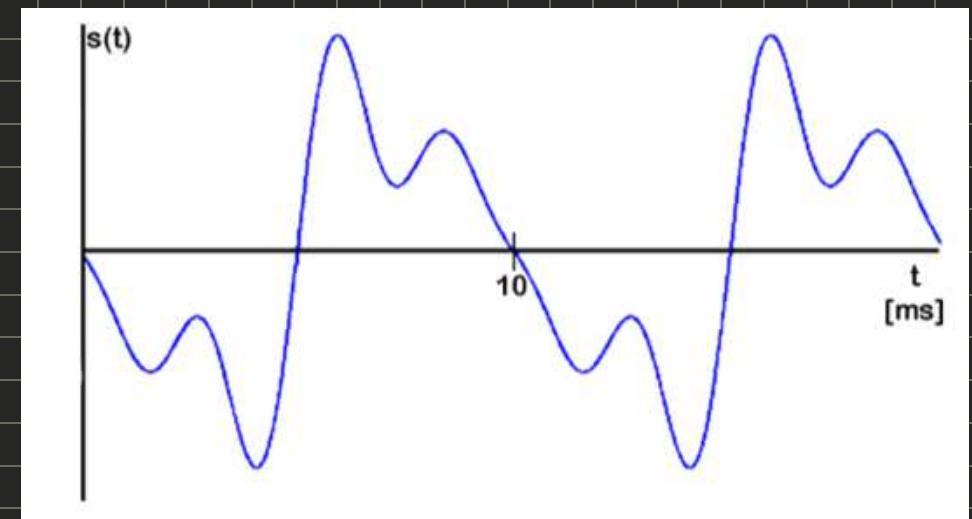
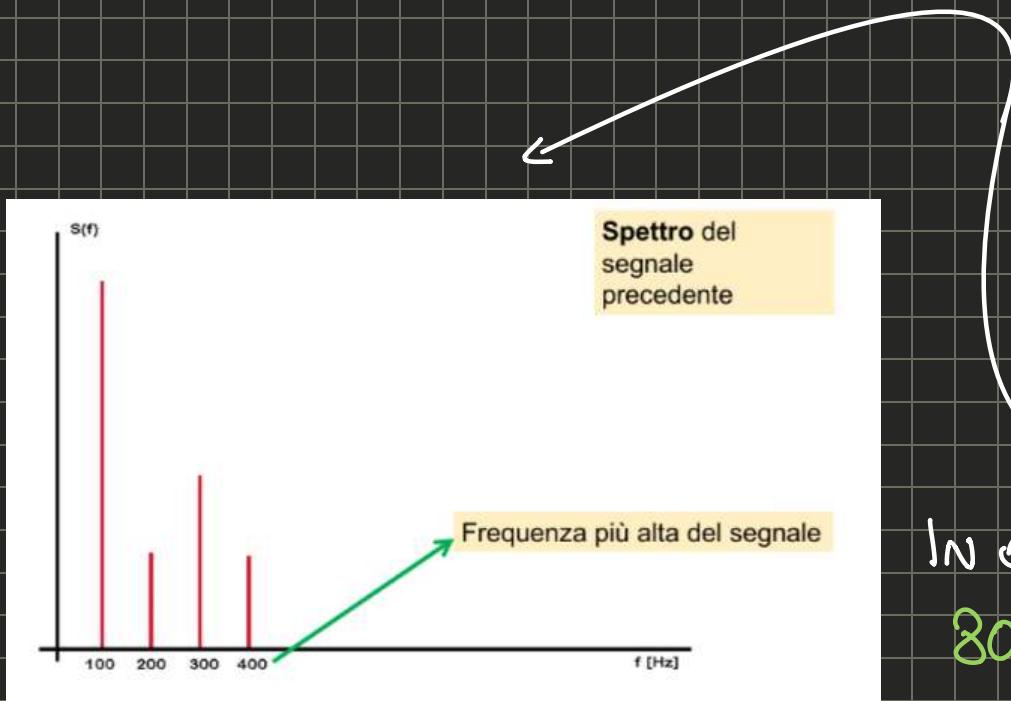
- Se il VALORE è su 4 PIANI (SAE e SCENDE) DEVO suddivid. in ④



4 SEGMENTI

→

Se invece vi è una fotografia, un quarto  
PARTI dentro **SUDDIVIDERE** la SCENA per rappres.  
l' IMMAGINE?



SOMMANDO 4 SEGNALI (100, 200, 300, 400) Hz

In questo caso il Nyquist Rate è  
 $800 \text{ Hz} = (400 \text{ Hz} (\text{Freq} + \text{Alta}) \cdot 2)$

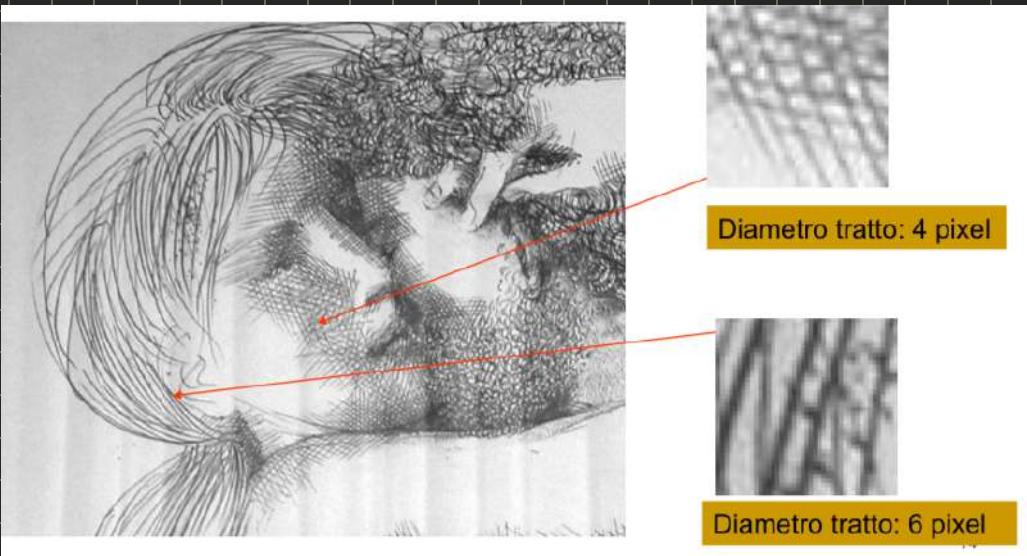
# TEOREMA DEL CAMPIONAMENTO

## di SHANNON (1948)

TEO.

Se si raccolgono campioni con frequenza più alta del Nyquist rate il segnale può essere ricostruito FEDELMENTE in ogni suo punto!

### APPLICAZIONE ALLE IMMAGINI



2  
CAMPIONAM. SBAGLIATO

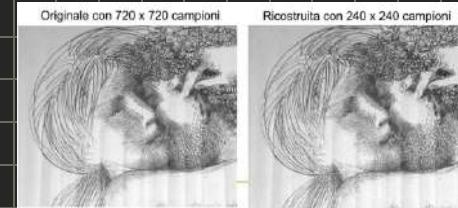
Decidiamo di volere trascurare i tratti fini. La nostra ipotetica «frequenza più alta» è allora:

- dimensione quadro 720 pixel, dettaglio massimo 6 pixel, possiamo dividere l'intervallo in  $720/6=120$  tratti.
- Il Nyquist rate è quindi  $2 \cdot 120 = 240$ . Prenderemo allora solo 240 campioni e ricostruiremo con l'interpolazione binomiale l'immagine.

Originale con 720 x 720 campioni



Campionata con 240 x 240 campioni



17

1  
→

CAMPIONAMENTO CORRETTO

Usiamo i tratti fini. Se preserviamo questi, allora abbiamo preservato anche gli altri. La nostra «frequenza più alta» è allora:

- dimensione quadro 720 pixel, dettaglio massimo 4 pixel, possiamo dividere l'intervallo in  $720/4=180$  tratti.
- Il doppio di tale frequenza è il Nyquist rate: 360. Prenderemo allora solo 360 campioni e ricostruiremo con l'interpolazione bilineare l'immagine.

Originale con 720 x 720 campioni

Originale con 720 x 720 campioni

Campionata con 360 x 360 campioni

Campionata con 360 x 360 campioni



Ricostruita

3  
→

CAMPIONAM. SBAGLIATISSIMO

Campioniamo a frequenza inferiore al Nyquist rate.

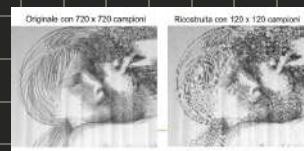
Originale con 720 x 720 campioni

Originale con 720 x 720 campioni

Campionata con 120 x 120 campioni

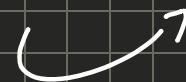
Campionata con 120 x 120 campioni

Ricostruita



# ALIASING

Con esso le **ALTE FREQUENZE SONO "MASCHERATE"**  
da BASSE FREQUENZE e TRATTATE come  
falsi nella fase di CAMPIONAMENTO (alias = falsa identità)



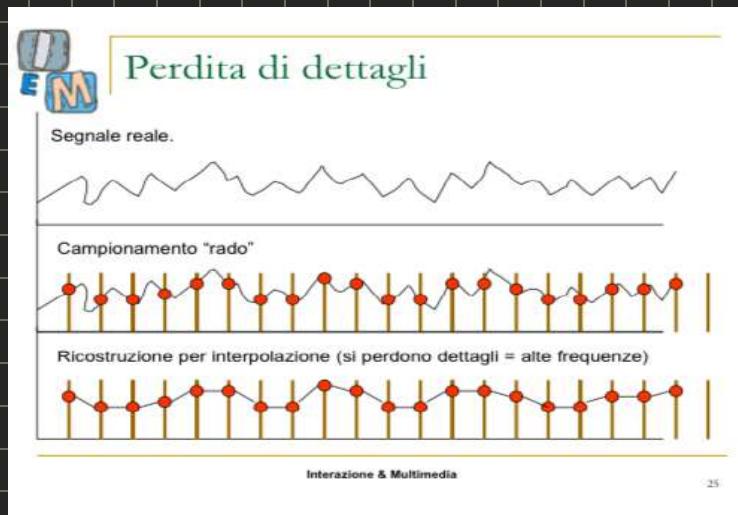
**Aliasing**

Campionamento Un pixel ogni 256

Ricostruzione con interpolazione bicubica

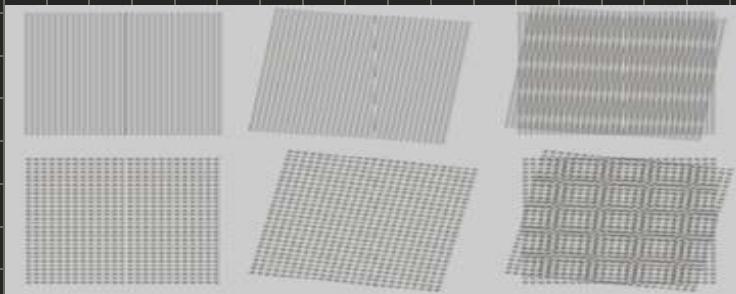
Si perdono dettagli, graffi e disegni sulle rocce sono divenuti indistinguibili e sono apparsi NUOVI dettagli!

- a) Ovvie scalettature sui bordi dei sassi.
- b) Fori che non erano presenti nell'originale!



Si perdono dettagli significativi  
e spesso si introd.  
**NUOVI DETTAGLI** che  
**NON erano presenti**

# EFFETTO MOIRÉ



Es:



# QUANTIZZAZIONE

I SENSORI forniscono MISURE REALI di LUMINOSITÀ come NUMERI REALI.

Per questo è utile ARROTONDARE tali valori e MANTENERLI in un RANGE. ( )

PROBLEMA!

ESISTONO CORRIENTI PARASSITE CHE INDUC.  
RUMORE, "DARK CURRENT"

## - PROCEDURA GENERALE

Se i valori da quantizzare sono numeri reali nel range [a, b] e si vuole quantizzare su n livelli:

1) - Si fissano n+1 numeri ( $t_0, t_1, \dots, t_n$ ) in  $[a, b]$  tali che:

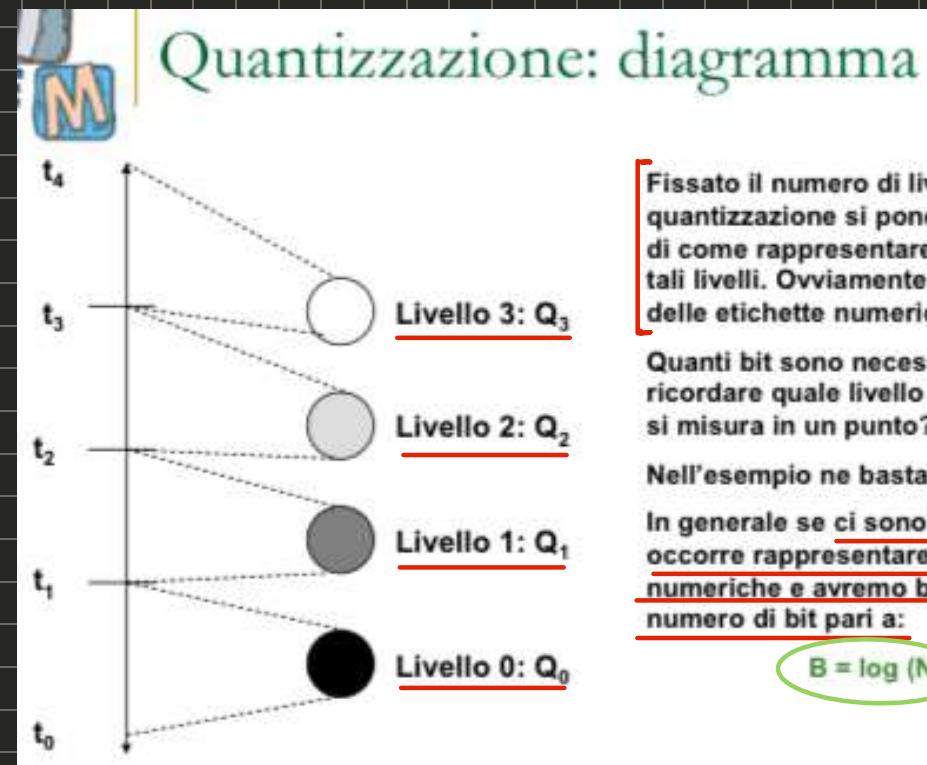
$$t_0 = a < t_1 < t_2 < \dots < t_n = b$$

2) - Il numero  $x$  in  $[a, b]$  verrà assegnato al livello di quantizzazione  $k$  se risulta:

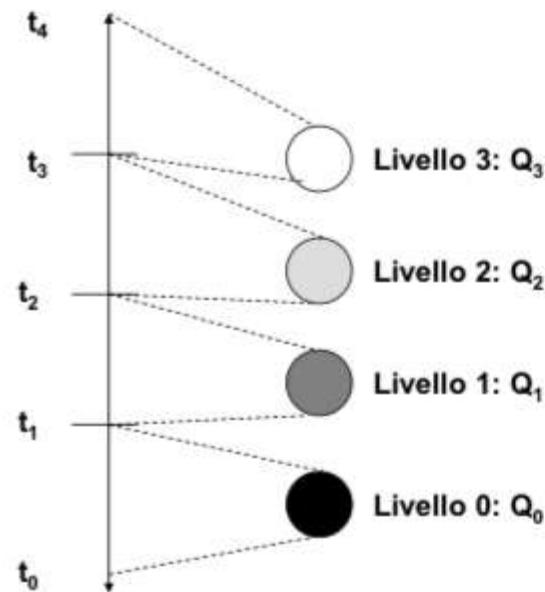
$$t_k \leq x < t_{k+1}$$

( $b$  viene assegnato al livello  $k$ .)

3) A ciascun livello si assegna un valore rappresentativo  $Q_i$



# Diagramma



Fissato il numero di livelli di quantizzazione si pone il problema di come rappresentare in memoria tali livelli. Ovviamente utilizzeremo delle etichette numeriche.

Quanti bit sono necessari per ricordare quale livello di luminosità si misura in un punto?

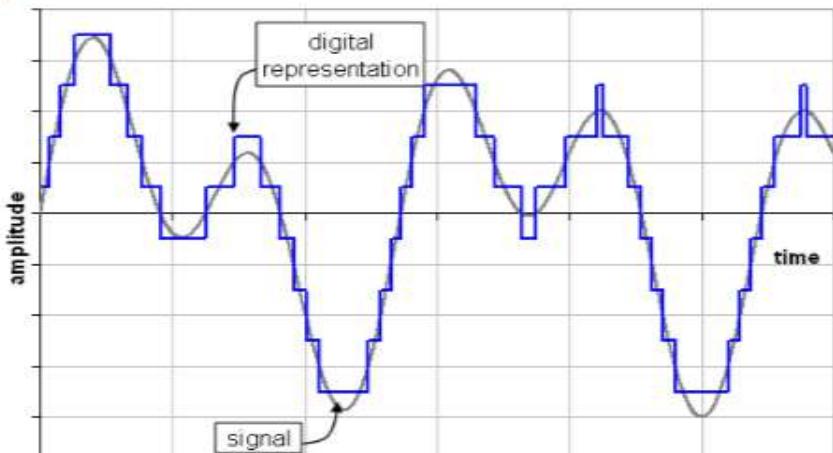
Nell'esempio ne bastano  $2 = \log(4)$

In generale se ci sono  $N$  livelli occorre rappresentare  $N$  etichette numeriche e avremo bisogno di un numero di bit pari a:

$$B = \log(N)$$

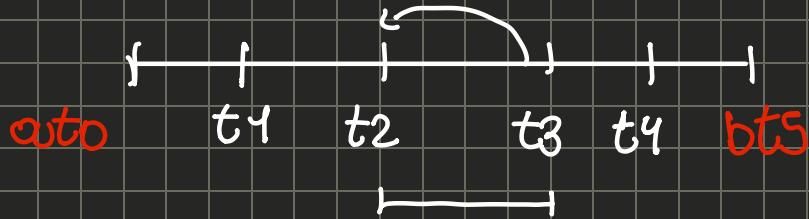
## Esempio di QUANTIZZAZIONE

### Quantizzazione - Esempio



## TIPI di QUANTIZZAZIONE





ARROTONDA (DF.)

17 ←

17,5	*	*	*

CAMPIONAMENTO

17	17	17	17	17
*	*	*	*	*
17	17	17	17	17
*	*	*	*	*
17	17	17	17	17

Se FACCIAMO UN ERRORE e QUANTIZZIAMO  
un questo SEGMENTO non si NOTA DIFFERENZA

## USI QUANTIZZAZIONE

La quantizzazione effettuata dagli scanner commerciali e dalla fotocamere digitali è non uniforme e logaritmica:

- permette di assegnare più livelli nella area dei toni scuri e meno livelli nella area dei toni chiari.

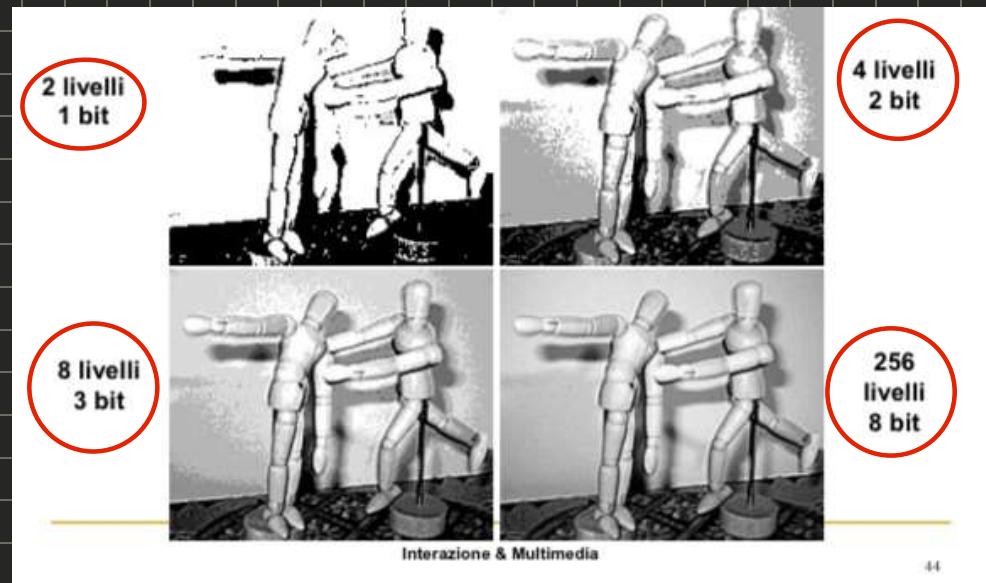
Questo è particolarmente importante quando si elaborano dati medici (es.radiografie)



Quantizzazione  
uniforme



Quantizzazione  
logaritmica



## QUANTIZZAZIONE UNIFORME

- INGRESSO  $0 \dots \overset{N=256}{\overbrace{N-1}}$
  - USCITA  $0 \dots K-1$  con  $K \leq N$
- ES:

IMM. 8 bit

0	50
100	200

IMM. 4 bit

0	3,125
1,56	6,25

$$N = 256$$

$\Rightarrow$

$$L' = \frac{L \cdot K}{N} \Rightarrow$$

$$\frac{0 \cdot 8}{256} = 0$$

$$\frac{100 \cdot 8}{256} = 3,125$$

$$\frac{50 \cdot 8}{256} = 1,56$$

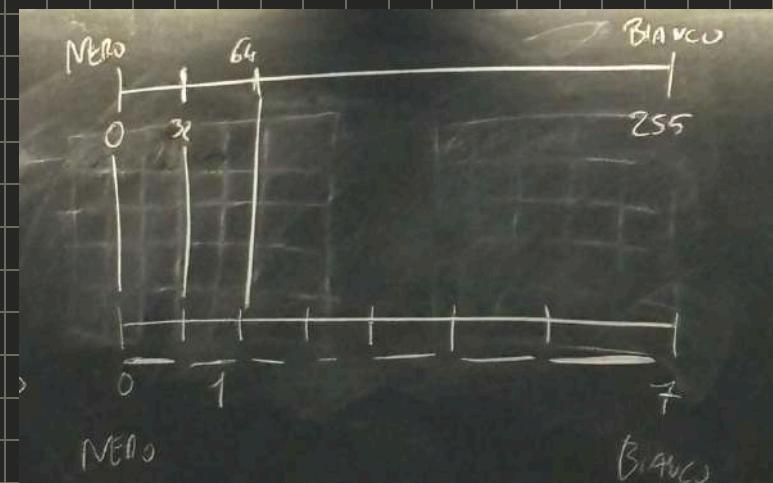
$$\frac{200 \cdot 8}{256} = 6,25$$

0	3
1	6

FORMULA

$$L' = \log_2 L * K$$

$$\log_2 N$$



## QUANTIZZAZIONE NON UNIFORME

0	6,64
5,64	7,64

→

0	6
5	7

$$\frac{\log_1 0 \cdot 8}{\log_2 256} = 0$$

$$\frac{\log_2 100}{\log_1 256} \cdot 8 \rightarrow \frac{\log_2 100 \cdot 8}{8}$$

Se  $L$  è il livello di ingresso rappresentato da un intero il livello  $L'$  corrispondente dopo la riquantizzazione è:

$$L' = f(L, N, K)$$



## QUANTIZZAZIONE LOGARITMICA

Tra le più comuni è la

Quantizzazione logaritmica:  $f(L, N, K) = (\log_2 (L) * K) / \log_2(N)$

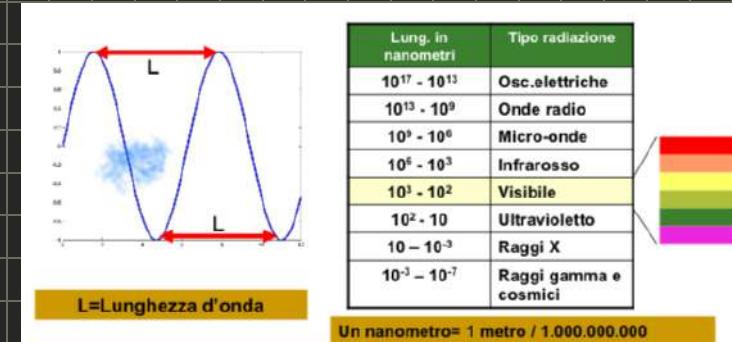
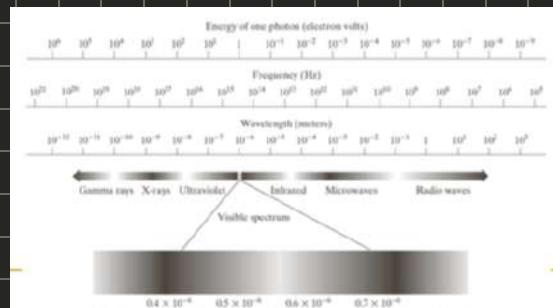
Nel caso più comune  $N=256$ ,  $\log_2(N)=8$  e  $K=8$   $\log_2(N)=K$ , per cui  $f(L, N, K)=\log_2 (L)$ .



## PDF 8 - COLORI

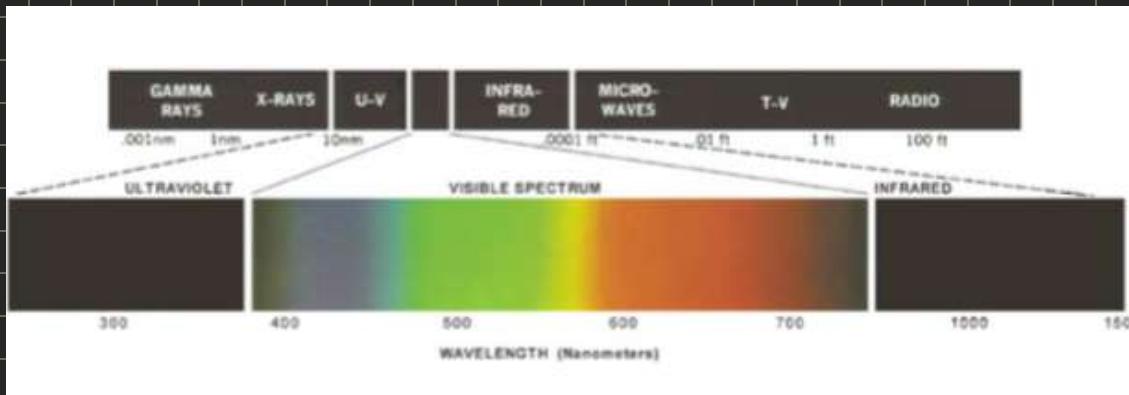
"SE UN RAGGIO LUMINOSO BIANCO attraversa UN PRISMA DI VETRO, ciò che si OTTIENE è UNO SPECTRUM di colori dal VIALE AL ROSSO". Quindi la LUCE può essere DECOMPOSTA in ONDE LUMINOSE DIFFERENTI

GAMMA  
BIANCO e  
NERO



GAMMA  
Colori

La luce è FORMATA da LUNGHEZZE diverse in REGIONI che vanno dal VIOLETTO al ROSSO



L'occhio umano percepisce come colore di un oggetto quella luce che l'oggetto stesso riflette.

Se un oggetto riflette tutte le lunghezze d'onda luminosa, allora l'oggetto sarà percepito come bianco.

Un oggetto che riflette le lunghezze d'onda da 500 a 570 nm ed assorbe tutto il resto, sarà percepito come di colore verde.

Radianza: cioè la quantità di luce emessa dalla sorgente luminosa;

Luminanza: cioè la misura dell'energia percepita dall'utente;

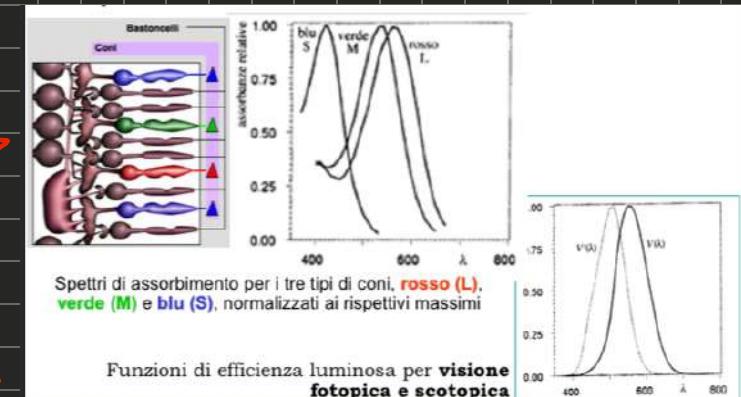
Brillantezza: è un valore soggettivo che indica la sensazione di colore.

# I CONI

**TIPO S** : (Shorts) Sensibili alle lunghezze d'onda corte

**TIPO M** : (Medium) Sensibili alle lunghezze d'onda medie →

**TIPO L** : (Large) Sensibili alle lunghezze d'onda lunghe



# DIFFERENZE di COLORI

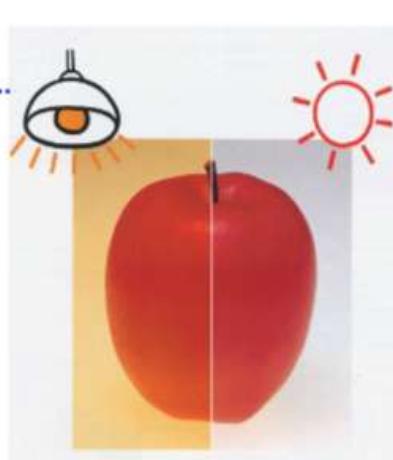
Differenze di sorgenti luminose  
luce solare, lampade ad incandescenza, neon,...

Differenze di osservatore  
sensibilità, vista,...

Differenze di dimensioni  
Effetto area = colori che ricoprono vaste superficie appaiono più luminosi e vivaci

Differenze di sfondo  
Effetto contrasto = colori appaiono più cupi se lo è sfondo scuro

Differenze di direzione  
Caratteristiche direzionali di alcune vernici  
costanza degli angoli di osservazione e di illuminazione



# COMPOSIZIONE COLORI

## Composizione dei colori

Illuminando una superficie bianca (coefficiente di riflessione del 100%) con una o più luci monocromatiche si ottiene (per riflessione) una radiazione colorata per

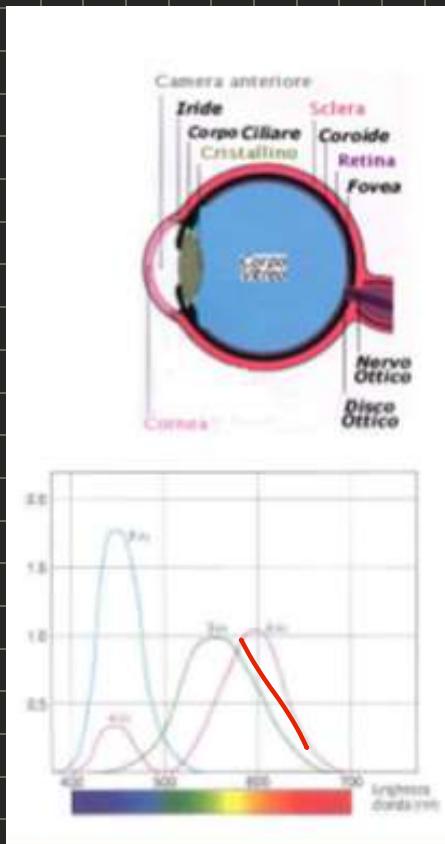
### Sintesi Additiva

**Def**  
Qualsiasi luce di spettro complesso può essere ottenuta come "somma" di tre luci monocromatiche agenti simultaneamente.

La CIE, partendo dall'ipotesi che tutti i colori possono essere valutati in termini di tre componenti pure, ha scelto le tre radiazioni primarie in modo che le lunghezze d'onda siano il più possibile distanziate nello spettro.

Rosso	700.0 nm
Verde	546.1 nm
Blu	435.8 nm

# COLOMETRIA



l'occhio serve per MISURARE il COLORE

HA 3 tipi di CONI

↗

- Per calcolare il colore serve una Terna

valori del tristimolo X, Y, Z

$$X = \int_{380}^{780} L_{e,\lambda} \bar{x}(\lambda) d\lambda$$

$$Y = \int_{380}^{780} L_{e,\lambda} \bar{y}(\lambda) d\lambda = \int_{380}^{780} L_{e,\lambda} V(\lambda) d\lambda$$

$$Z = \int_{380}^{780} L_{e,\lambda} \bar{z}(\lambda) d\lambda$$

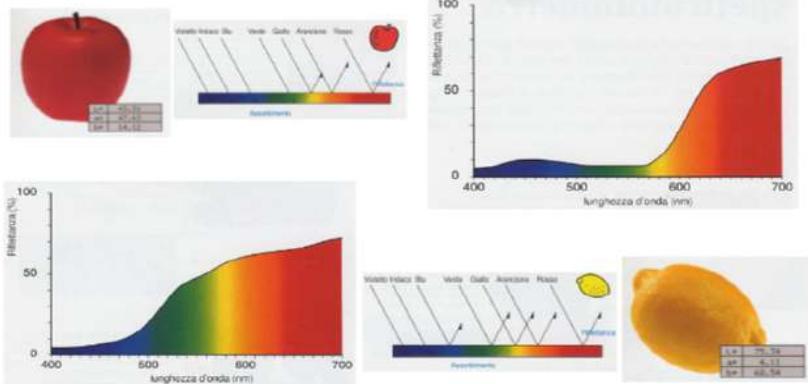
①

②

③

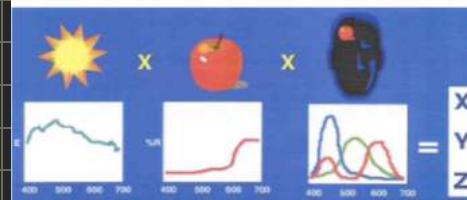
## COLORE e RIFLETTANZA SPEGTRALE

### Colore e riflettanza spettrale



## DATI COLORIMETRICI

### Dati colorimetrici



$(L^*, a^*, b^*)$

1. Distribuzione energetica della luce che lo illuminata
2. Curva spettrale dell'oggetto colorato,
3. Sensibilità al colore dell'occhio umano.

Distribuzione spettrale degli illuminanti più diffusi:



FORMULA  
(RGB)

# VISIONE e MISURA

## VISIONE UMANA

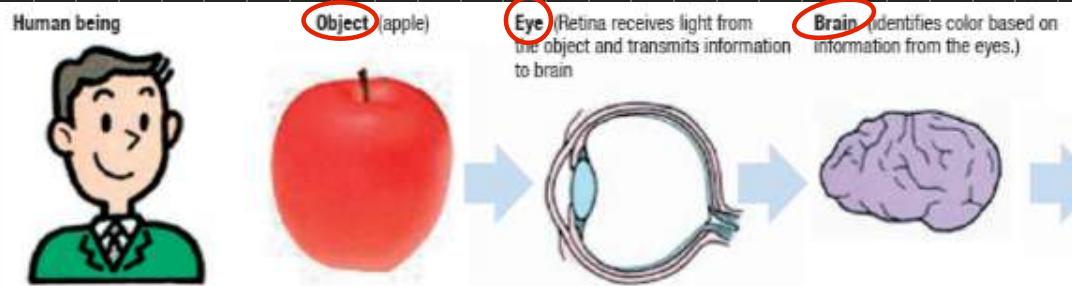
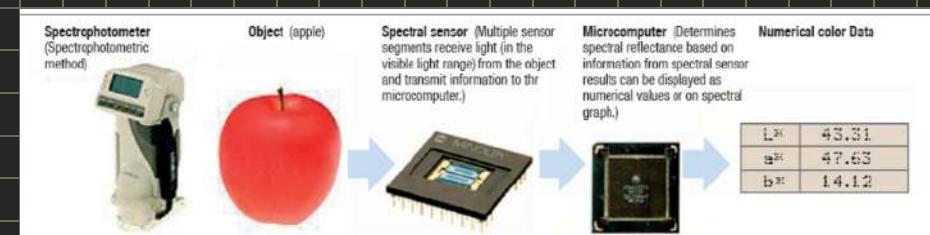
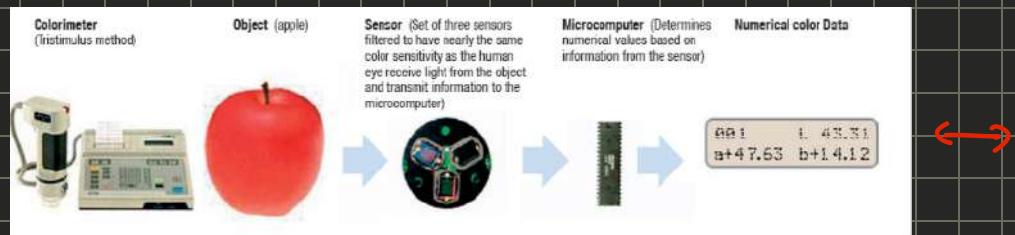


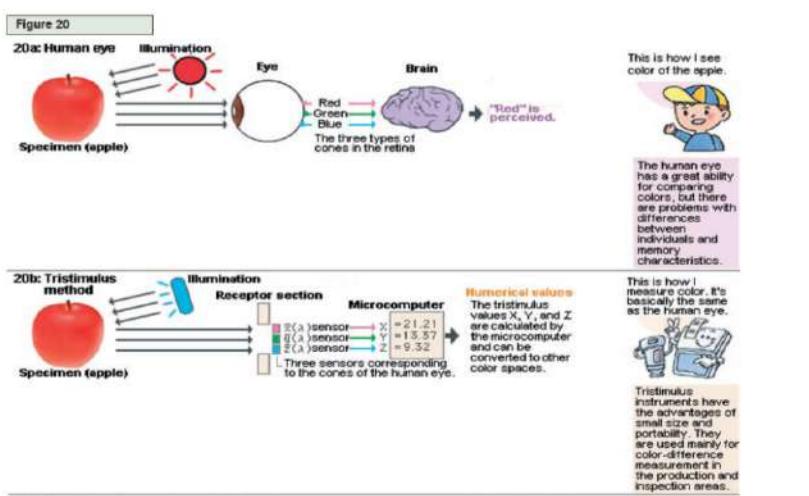
Figure 16

Colore  
INDIVIDUATO

## VISIONE MACCHINA



ES:



colori RGB tramite **coni**

colori tramite **SENSORI**

L'occhio umano ha una grande abilità, ma vi sono molte differenze tra individui e caratteristiche della memoria

Piccole e portabili. Vengono usati per le differenze di colore nelle aree di produzione →

# METODO SPECTROFOTOMETRICO



- Una **luce** allo Xenon illumina il campione in maniera uniforme;
- La luce riflessa dalla superficie del **campione** viene raccolta da un **sistema di lenti** ed entra nel sensore spettrale principale
- La luce che entra in ciascun **sensore** viene divisa in lunghezze d'onda da un reticolo di diffrazione e convertita in segnale elettrico da una array di fotodiodi

## I LUMINANTI

### Illuminante

Energia radiante con distribuzione spettrale di energia relativa definita nel campo di lunghezza d'onda capace di influenzare la visione del colore degli oggetti.

### Illuminante CIE :

Gli illuminanti colorimetrici A, B, C e D65 sono quelli le cui distribuzioni spettrali di energia relativa sono definite dalla CIE.



Esempi :

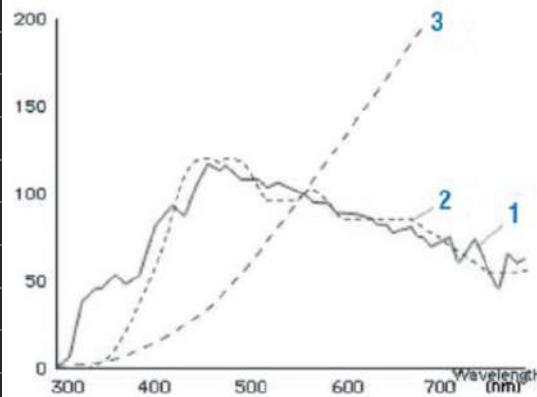


Figure 22a: Standard Illuminants

**1** Standard Illuminant Average daylight (including ultraviolet wavelength region) with a correlated color temperature of 6504K; should be used for measuring specimens which will be illuminated by daylight including ultraviolet radiation. **2** Standard Illuminant C Average daylight (not including ultraviolet wavelength region) with a correlated color temperature of 6774K; should be used for measuring specimens which will be illuminated by daylight in the visible wavelength range but not including ultraviolet radiation. **3** Standard Illuminant A Incandescent light with a correlated color temperature of 2856K; should be used for measuring specimens which will be illuminated by incandescent lamps.

Nome	CIE 1931 2°		CIE 1964 10°		CCT (K)	Hue	Note
	x <sub>2</sub>	y <sub>2</sub>	x <sub>10</sub>	y <sub>10</sub>			
A	0.44757	0.40745	0.45117	0.40554	2856	Incandescente / Tungsteno	
B	0.34842	0.35161	0.34980	0.35270	4874	(obsoleta) luce diretta del sole a mezzogiorno	
C	0.31006	0.31616	0.31039	0.31905	6774	(obsoleta) Average / North sky Daylight	
D50	0.34567	0.35850	0.34773	0.35952	5003	Horizon Light ICC profile PCS	
D55	0.33242	0.34743	0.33411	0.34877	5503	Mid-morning / Mid-afternoon Daylight	
D65	0.31271	0.32902	0.31382	0.33100	6504	Luce diurna a mezzogiorno: Televisione, sRGB	
D75	0.29902	0.31485	0.29968	0.31740	7504	North sky Daylight	
E	1/3	1/3	1/3	1/3	5454	Equal energy	
F1	0.31310	0.33727	0.31611	0.33559	6430	Luce diurna fluorescente	
F2	0.37208	0.37526	0.37925	0.36733	4230	Luce Fluorescente bianca fredda	
F3	0.40910	0.39430	0.41761	0.38324	3450	Luce fluorescente bianca	
F4	0.44018	0.40329	0.44520	0.39074	2940	Warm White Fluorescent	
F5	0.31379	0.34531	0.31975	0.34246	6350	Luce diurna fluorescente	
F6	0.37790	0.38635	0.38660	0.37847	4150	Fluorescente bianco chiaro	
F7	0.31292	0.32933	0.31569	0.32960	6500	D65 simulatore, simulatore di luce diurna	
F8	0.34588	0.35875	0.34902	0.35939	5000	D50 simulator, Sylvania F40 Design 50	
F9	0.37417	0.37281	0.37829	0.37045	4150	Cool White Deluxe Fluorescent	
F10	0.34609	0.35986	0.35090	0.35444	5000	Philips TL85, Ultralume 50	
F11	0.38052	0.37713	0.38541	0.37123	4000	Philips TL84, Ultralume 40	
F12	0.43695	0.40441	0.44256	0.39717	3000	Philips TL83, Ultralume 30	

# CIE 1931 e 1964

## CIE 1931 e 1964

### XYZ Tristimulus Values (CIE 1931)

Tristimulus values determined based on the color matching functions  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$  and  $\bar{z}(\lambda)$  defined in 1931 by CIE; also referred to as  $2^\circ$  XYZ tristimulus values. They are suitable for a viewing angle of  $4^\circ$  or less and are defined for reflecting objects by the following formulas:

where

$S(\lambda)$ : Relative spectral power distribution of the illuminant

$\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$ ,  $\bar{z}(\lambda)$ : Color-matching functions for CIE  $2^\circ$  Standard Observer (1931)

$R(\lambda)$ : Spectral reflectance of specimen

### X<sub>10</sub> Y<sub>10</sub> Z<sub>10</sub> Tristimulus Values (CIE 1964)

Tristimulus values determined based on the color matching functions  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$  and  $\bar{z}(\lambda)$  defined in 1964 by CIE; also referred to as  $10^\circ$  XYZ tristimulus values. They are suitable for a viewing angle of more than  $4^\circ$  and are defined for reflecting objects by the following formulas:

where

$S(\lambda)$ : Relative spectral power distribution of the illuminant

$\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$ ,  $\bar{z}(\lambda)$ : Color-matching functions for CIE  $10^\circ$  Supplementary Standard Observer (1964)

$R(\lambda)$ : Spectral reflectance of specimen

## STANDARD CIE

$$\begin{array}{c} \frac{x}{x+y+z} \\ \downarrow \\ \frac{y}{x+y+z} \\ \downarrow \\ \frac{z}{x+y+z} \end{array}$$

$$x = \frac{\textcircled{x}}{x+y+z}$$

$$y = \frac{\textcircled{y}}{x+y+z}$$

$$z = \frac{\textcircled{z}}{x+y+z}$$

VERSIONE  
NORMALIZZATA

Nel 1931 il **CIE** (Commission Internationale de l'Eclairage) ha fissato le **lunghezze d'onda standard** per i tre colori primari:  
**Blue** = 435,8 nm  
**Verde** = 546,1 nm  
**Rosso** = 700 nm

- Anche se nel 1964 i dati sperimentali hanno dimostrato che in realtà il **valore reale** è lievemente differente.

**Blue** = 445 nm  
**Verde** = 535 nm  
**Rosso** = 575 nm

## DIAGRAMMA CHROMATICO CIE

$$\begin{array}{l} x = 1 \\ y = 1 \\ z = 1 \end{array}$$

**COLORE 1**

$$x = \frac{1}{3} \downarrow$$

$$y = \frac{1}{3}$$

$$z = \frac{1}{3}$$

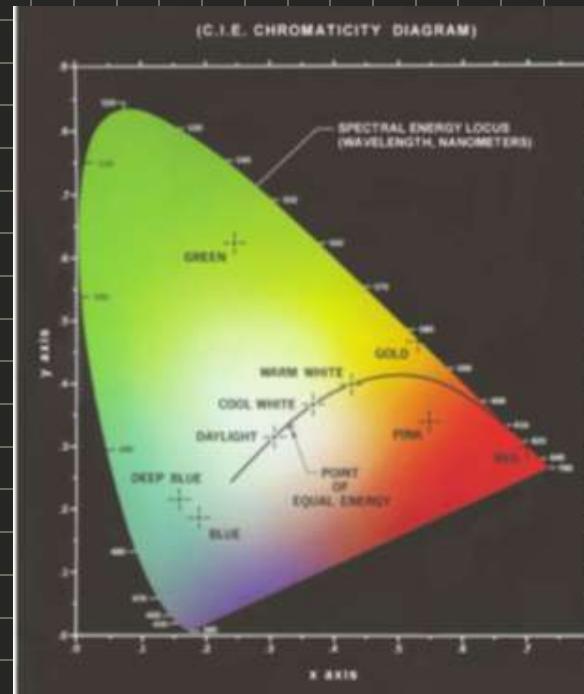
$$\begin{array}{l} x = 2 \\ y = 2 \\ z = 2 \end{array}$$

**COLORE 2**

$$\downarrow$$

$$x = \frac{2}{2+2+2} = \frac{1}{3}$$

$$y = \frac{1}{3}, z = \frac{1}{3}$$



Tutti i colori delle lunghezze d'onda visibili sono disposti lungo i bordi.

- Il punto di uguale energia è il bianco.

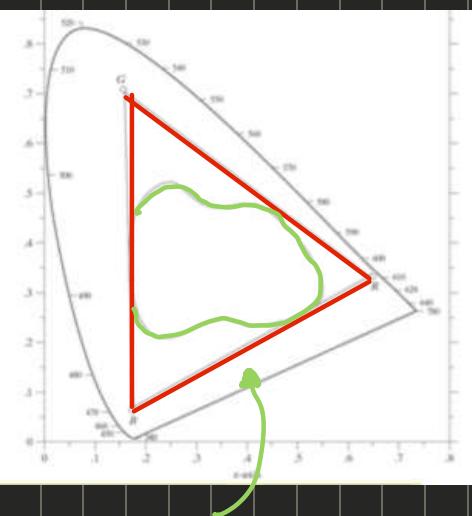
$$0 < b < \frac{1}{3}$$

- Qualsiasi colore lungo il bordo non ha bianco, quindi è puro.

**colori  
NON  
DEFINITI**

MODELLI DEL COLORE  
 Lo scopo di un MODELLO DI COLORE è  
quello di consentire la specificazione  
dei colori con modalitá 3D, cioè un  
 un SOTTOSPAZIO dove ogni colore è  
RAPPRESENTATO, ola UN PUNTO.

- Unendo R G e B si ottiene un triangolo che contiene tutti i colori che si possono produrre.
- Da notare che il triangolo non copre tutta l'area, quindi non tutti i colori si ottengono unendo R G e B.



L'area irregolare dentro il triangolo rappresenta tutti i colori che una stampante può ottenere.

- Questi sono in numero minore rispetto a quelli dei monitor perché è differente il modo di mischiare i colori (additivo vs sottrattivo).

## Tipi del MODELLO

I modelli del colore più utilizzati in image processing sono orientati o all'hardware di acquisizione (RGB) e restituzione (RGB, CMY) delle immagini o alla loro trasmissione (YUV, YIQ), compressione (YCbCr), elaborazione (od analisi) mediante trattamento del colore (RGB, HSI, HSV, LUV, ...).

## DIFETTO DEL SISTEMA CIE !!!

### PROBLEMA DELLA DISTANZA

In altre parole, dati due colori  $C_1$  e  $C_2$ , consideriamone le distanze  $\Delta C$ , rispettivamente, dal colore  $C_3 = C_1 + \Delta C$  e dal colore  $C_4 = C_2 + \Delta C$ . Supponendo che le due distanze siano quantitativamente uguali, sarebbe desiderabile che i due colori  $C_3$  e  $C_4$  fossero percepiti come ugualmente distanti da  $C_1$  e  $C_2$ . Dato che il sistema non è percettivamente uniforme, in generale le due distanze, benché uguali, saranno percepite come differenti.



Per questo vi è CREATO 2 SPAZI DI COLORE

PERCETTIVAMENTE UNIFORMI

↳ CIE  $L^* * a^* * b^*$       CIE  $L^* * a^* * b^*$   
 (\* = STAR)

# FORMULE

Si può passare dallo SPAZIO di Colore XYZ a CIELAB con:

$$L^* = 116 \left( \frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 \quad a^* = 500 \left[ \left( \frac{X}{X_n} \right)^{\frac{1}{3}} - \left( \frac{Y}{Y_n} \right)^{\frac{1}{3}} \right] \quad b^* = 200 \left[ \left( \frac{Y}{Y_n} \right)^{\frac{1}{3}} - \left( \frac{Z}{Z_n} \right)^{\frac{1}{3}} \right]$$

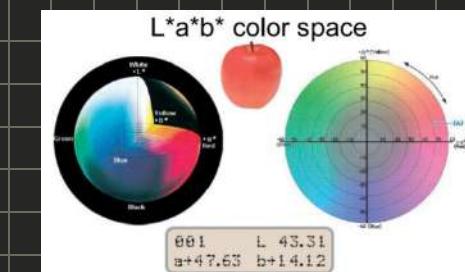
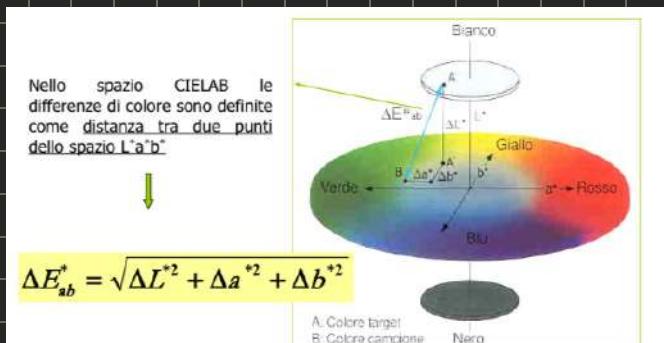
- Dove  $X/X_n$ ,  $Y/Y_n$ , e  $Z/Z_n$ , sono maggiori di 0.01 e  $X_n$ ,  $Y_n$  e  $Z_n$  definiscono il punto bianco. Lo spazio colore CIELAB in confronto al CIE XYZ rispetta maggiormente la risposta non lineare dell'occhio umano alle radiazioni elettromagnetiche.
- La metrica CIELAB si ottiene dalla seguente formula:

$$\Delta E_{ab}^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$$

!!!

$L^*$  rappresenta la **luminanza** mentre  $a^*$  e  $b^*$  rappresentano la **crominanza**.  
 $L^*$ ,  $a^*$  e  $b^*$  sono le trasformazioni dei tre valori di tristimolo X, Y e Z dello spazio colore CIE XYZ.

## DIFFERENZE di Colore



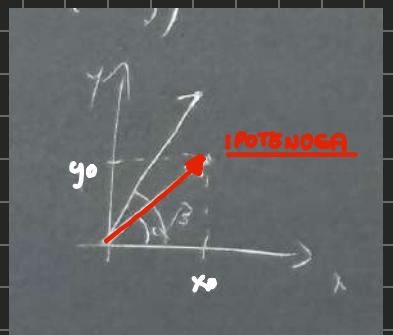
$$\Delta E_{ab}^* = \sqrt{(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2}$$

$C_1 = (11, 3, 4)$   
 $C_2 = (10, 3, 5)$

$$\Delta E(C_1, C_2) = \sqrt{(11-10)^2 + (3-3)^2 + (4-5)^2} = \sqrt{1+0+1} = \sqrt{2}$$

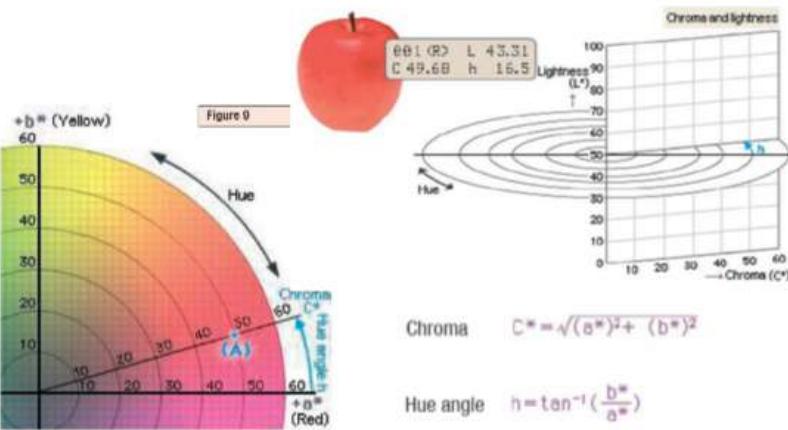
ES: NELL' INDUSTRIA dei cosmetici il **ΔE** sopra UNA SOGLIA oppure IL **ROSSO FERRARI** lo HA MOLTO BASSO

## COORDINATE POLARI ( $L^*C^*h$ )

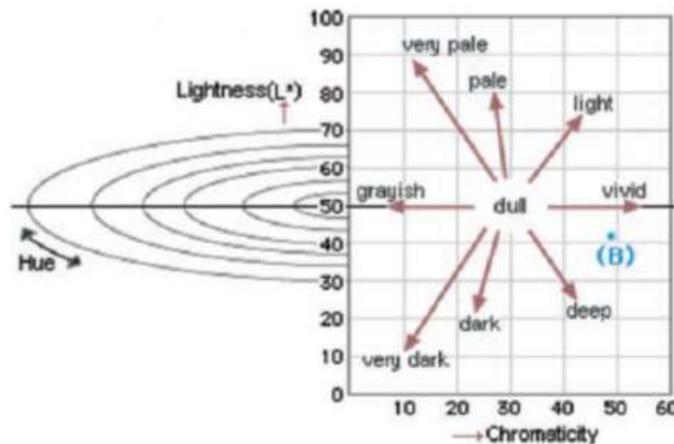


# $L^*C^*h$ color space

## $L^*C^*h$ color space



## $L^*C^*h$ color space



$L$ : LUMINOSITÀ

$0 = \text{NERO} / 100 = \text{BIANCO}$

$C$ : DISTANZA dal CENTRO

$h$ : AMPIEZZA del RAGGIO

BIANCO e NERO

NERO

IL NERO MATEMATICO è rappresentato da  $X = 0$ ,  $Y = 0$ ,  $Z = 0$  e  
mentre un altro corpo possiede queste SPECIFICHE

BIANCO

PER IL BIANCO la distribuzione SPECTRALE di POTENZA è 1 CON  $X = 1$ ,  $Y = 1$ ,  $Z = 1$   
e diversamente il BIANCO è quello associato all'ILLUMINANTE SCELTO

## Il bianco e il nero

L'osservatore e l'illuminante devono essere scelti, perché diversamente non c'è misurazione del colore

Quale osservatore scegliere?

- Per oggetti da vedersi in modo analitico, cioè quasi tutti gli oggetti che rientrano tra i beni culturali, l'osservatore è il CIE 1931.
- Per oggetti di grandi superfici, quali oggetti architettonici e forse anche affreschi destinati ad essere visti da lontano, l'osservatore è il CIE 64.

L'illuminante classico per tutti gli usi è il D65.

L'atlante di Munsell è definito per l'osservatore CIE 64 e l'illuminante D65. Le sue proprietà dal punto di vista percettivo non valgono più se si opera in condizioni differenti

Le comparazioni visive vanno fatte in situazione visiva controllata

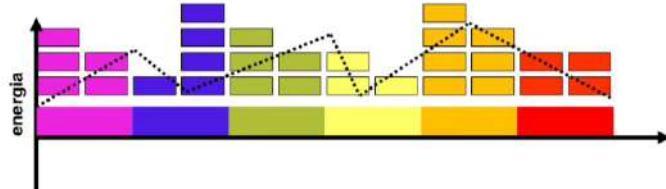
## Spazi di colore:

I colori dipendono dallo spettro ma non riusciamo a caratterizzarlo a livello visivo; infatti spettri diversi possono percepirci uguali.



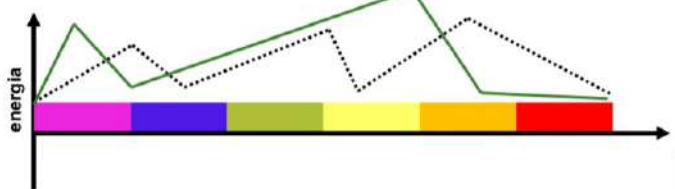
### Spettro

- Il disegno mostra uno spettro "discretizzato" con una convenzione grafica del tipo visto in una applicazione di "equalizzazione" nello stereo.



### Metameri

Spettri diversi possono produrre colori uguali: coppie di spettri con questa reciproca proprietà si chiamano **metameri**.



*Lo spettro tratteggiato e quello continuo producono (nel cervello) il medesimo colore!*  
Tra i vari metameri di un dato spettro se ne può sempre individuare uno assai importante che è alla base del modello dei colori detto "del pittore".

Per ogni colore abbiamo un metamero



### Modello del pittore: basi fisiche

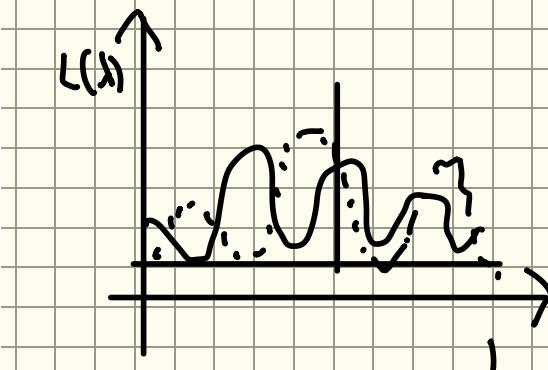
Ogni spettro ha un metamero della seguente forma:



La lunghezza d'onda in cui si ha il picco è responsabile del "colore percepito" (detto anche in inglese HUE).

Il rapporto  $(e_{picco} - e_{media}) / (e_{picco} + e_{media})$  è la **SATURAZIONE**, cioè quanto il colore è puro. Meno luce bianca equivale ad un maggiore valore del rapporto.

**e media** è proporzionale al contenuto energetico della radiazione: essa può essere considerata una misura della "luminosità" di una radiazione (detto anche **VALORE**). Esso da un contributo "bianco" al colore percepito.

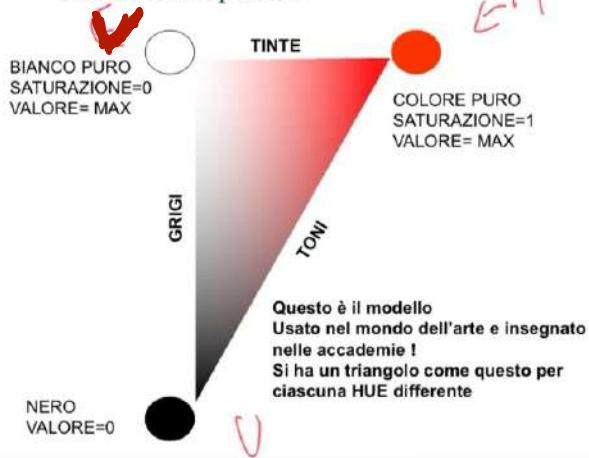


Saturazione è 0 quando  $e_{picco}$  ed  $e_{media}$  sono uguali

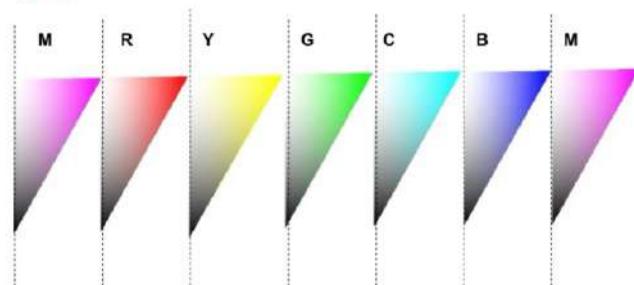
$$\text{Saturazione} = \frac{e_{picco}}{e_{media}}$$

Formula Saturazione =  $\frac{e_{picco}}{e_{media}}$

## Modello del pittore



## Mettere assieme i triangoli...



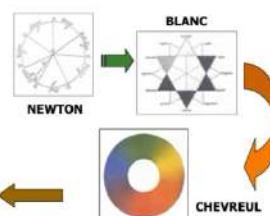
Tutte le linee verticali tratteggiate sono la rappresentazione della medesima "linea dei grigi".  
IDEA! Attachiamo tutti i triangoli in una "girandola" facendo coincidere le - linee dei grigi.

## Strumenti di rappresentazione del colore:



### Rappresentazione del colore

• Ruota dei colori organizza lo spazio colorato in uno schema simmetrico.



• Munsell collocazione dei colori in uno spazio tridimensionale.

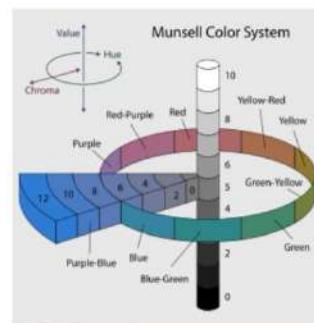


• Nascita di colorimetria e sistemi CIE per una misura fisica del colore.

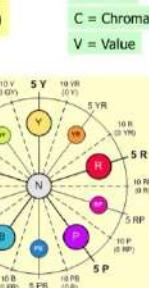
Interazione & Multimedia



### Munsell System



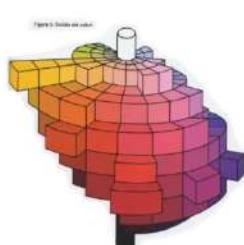
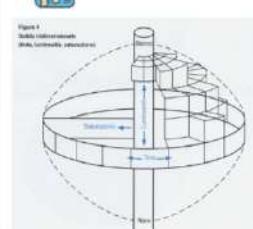
(H,C,V)



Interazione & Multimedia



### Il solido dei colori



**Tinta** = identifica il rosso, il blu, il giallo,... e forma la ruota dei colori  
**Luminosità** = colori chiari e colori scuri e cambia verticalmente  
**Satuzazione** = colori brillanti e colori opachi; cambia a partire dal centro

Interazione & Multimedia



### Discussione del modello del pittore

#### ■ PRO:

- Intuitivo;
- Percettivamente significativo: i parametri HSV hanno una perfetta interpretazione nelle nostre percezioni.

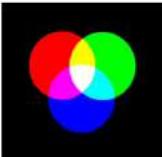
#### ■ CONTRO:

- Modello non lineare;
- Perché una piramide esagonale?
- Quanti sono i "colori base"?

# Sintesi Additiva

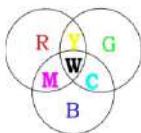
Parto da contributi di base  
e vado ad aggiungerlo in  
un supporto attraverso la sovrapposizione.

## Sintesi Additiva



RGB

Composizione additiva dei colori partendo dai tre colori primari RGB.  
Dalla loro sovrapposizione si ottiene il **bianco** (W), dalla sovrapposizione di due luci si ottengono **giallo** (Y), **magenta** (M) e **ciano** (C).



Su questo principio si basa il funzionamento di monitor e schermi televisivi.



Laboratorio PHIDRA - Physics for Dating Diagnostic and Dosimetry Research and Applications

74

La specificazione del colore

Gueli Anna

## I colori complementari

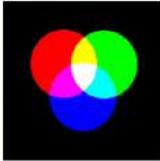
Colore Complementare = quando la somma di un dato colore ad un altro dà il Bianco (W)

$$R + G + B = W$$

$$R + G = W - B = Y$$

$$R + B = W - G = M$$

$$B + G = W - R = C$$



Il **Giallo** è complementare del **Blu**  
Il **Magenta** è complementare del **Verde**  
Il **Ciano** è complementare del **Rosso**

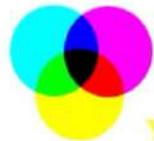
Il colore può essere ottenuto anche come mescolanza di **sostanze colorate** (pigmenti, coloranti, ecc.) il cui comportamento può essere simulato considerando dei **filtri colorati**.

Il complementare del nero è il bianco

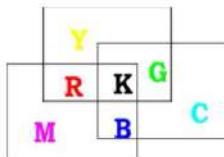
# Sintesi sottrattiva

agisce su tutte le lunghezze d'onda e definiamo ciò che dobbiamo rimuovere da certi contributi.

## Sintesi sottrattiva



YMC



Composizione sottrattiva dei colori partendo dai tre colori primari YMC ottenuti ponendo i tre filtri sul cammino di un fascio di luce bianca.

Dalla loro sovrapposizione si ottiene il nero (K), dalla sovrapposizione di due filtri si ottengono rosso (R), verde (G) e blu (B).

Sintesi Sottrattiva = aggiunge un filtro viene "sottratta" una componente che modifica il colore della luce.



Laboratorio PHODRA - Physics for Dating Diagnostic and Dosimetry Research and Applications

77

La specificazione del colore

Gueli Anna

## Immagini in sintesi sottrattiva

Nella sintesi sottrattiva si ottengono le seguenti combinazioni cromatiche (+ = sovrapposizione di filtri) :

$$Y + M = R$$

$$Y + C = G$$

$$M + C = B$$

$$Y + M + C = K$$



I pigmenti colorati si basano sulla sintesi sottrattiva.



Su questo principio si basa la riproduzione delle immagini colorate nella fotografia a colori e nella stampa con inchiostri.

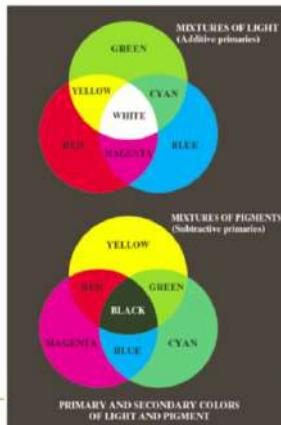


## Modello sottrattivo

- Quando non c'è inchiostro nella carta la luce riflessa è bianca; quando sono presenti tutti e tre i colori, la luce viene (in linea di principio) assorbita e la carta appare nera.
- In pratica, l'assorbimento completo è difficile da ottenere quindi si utilizza un quarto inchiostro, il nero (CMYK, dove K sta per black).

Interazione & Multimedia

80



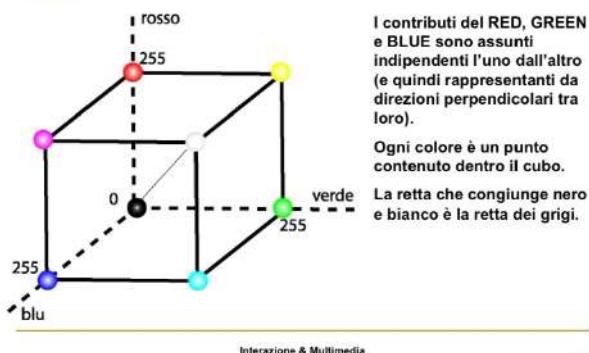
81

# RGB

Con coordinate cartesiane, il modello RGB è descritta da un cubo.



## Descrizione geometrica



## Discussione

- **PRO:**  
semplice da usare e implementare in software e hardware.  
De facto è uno STANDARD.
- **CONTRO:**  
percettivamente poco comodo: difficile capire guardando un colore in natura in quale proporzione vi contribuiscono l'R, il G e il B.

CUBO RGB e piramide HSV si trasformano l'un l'altro mediante semplici algoritmi (non lineari).

Esempio conversione:

RGB (100, 20, 35)

↓

CY (155, 225, 220)

In questo caso bisogna trovare i valori che sommano a quelli RGB tronciamo il 255 (franco)

- Le rappresentazioni dei colori nello spazio RGB (o CMY il duale) non sempre sono le più convenienti.
- Sono disponibili altre rappresentazioni che usano componenti che sono specificatamente relazionate al criterio usato per descrivere la luminanza, la tinta e la saturazione (HSV).
- La tinta descrive che colore è presente (rosso, verde, giallo, ecc.) e può essere correlato alla lunghezza d'onda dominante della sorgente di luce.
- La saturazione, invece, esprime quanto è vivo il colore (molto forte, pastello, vicino al bianco) e può essere correlato alla purezza o alla distribuzione dello spettro della sorgente.
- La luminanza è la grandezza che tende a valutare la sensazione luminosa ricevuta dall'occhio, è legata quindi all'intensità della luce (quanto il colore è bianco, grigio o nero) e può essere correlata alla luminosità della sorgente.

Il suo Complementare  
è il CMY

La differenza sta  
nella gamma di  
colori

# Colori sicuri per il Web:

Entrono dei sottosistemi che sono costituiti per avere la stessa mappa per quelle coordinate

#10 AB2F

Colore non sicuro per il web

# 33 00 CC

Colore sicuro per il web



## Colori sicuri per il web

- Si sa che 40 dei 256 colori usati tipicamente per il web, vengono processati in modo diverso da vari tipi di sistemi operativi, mentre 216 colori sono comuni alla maggior parte dei sistemi. Questi 216 colori sono diventati gli standard di fatto dei colori sicuri, specialmente nelle applicazioni Internet. Essi vengono utilizzati quando si vuole che i colori visti dalla maggior parte delle persone siano gli stessi.
- Ognuno dei 216 colori sicuri è formato come sempre da tre valori RGB, ma ogni valore può essere solo i numeri in esadecimale 00, 33, 66, 99, CC, FF.
- Quindi sono colori sicuri tutti quelli che in esadecimale sono scritti usando terne con questi 6 possibili lavori: ad esempio #33CCFF è sicuro mentre non lo è il colore #12A3FE.

Interazione & Multimedia

96



## Colori sicuri per il web



N.B (x ESAME)

Colori sicuri per il web:

Coppia di:

00 33 66 99 FF

dovranno esserci queste  
coppie



## Rappresentazioni luminanza-crominanza

- Gli spazi colore, nei quali una componente è la luminosità e le altre due componenti sono legate alla crominanza, vengono chiamate rappresentazioni luminanza-crominanza.
- La luminanza fornisce una versione a scala di grigi dell'immagine mentre la crominanza fornisce le informazioni "extra" che trasformano l'immagine in scala di grigi in un'immagine a colori.
- Le rappresentazioni luminanza-crominanza sono particolarmente importanti nella compressione delle immagini.
- L'occhio umano è più sensibile alla luminanza che ai colori. Posso dunque "spendere" molti bit per registrare la luminanza e risparmiarne un po' sulle crominanze.

Interazione & Multimedia

98



## Lo spazio YUV

- Lo spazio YUV viene spesso utilizzato per la codifica di immagini o video analogici, tenendo separate la luminanza dalla crominanza. ATTENZIONE! Con YUV ci si riferisce NON ad uno specifico spazio, ma ad una famiglia di spazi con le caratteristiche sopracitate. Come esempio riportiamo lo spazio YUV definito nello standard ITU-R BT.601-4.
- Tra le possibili controparti digitali degli spazi YUV troviamo gli spazi  $YC_bC_r$ .

**Es: Conversione  
DA RGB a YUV**

RGB (100, 20, 35)



$$Y = 0.299 \cdot R + 0.581 G + 0.114 B$$



## Da RGB a YUV

- La luminanza può essere ottenuta mediante una combinazione lineare delle intensità luminose dei canali rosso, verde e blu di RGB. Un'approssimazione abbastanza fedele della luminanza  $Y$  si ottiene attraverso la somma pesata:

$$Y = 0.299R + 0.581G + 0.114B$$

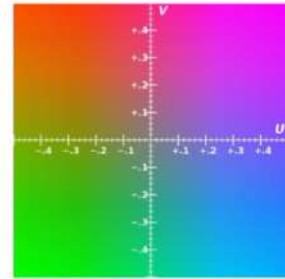
- Il termine **crominanza** è definito come la differenza tra il colore e un bianco di riferimento alla stessa luminanza opportunamente pesato:

$$U = 0.564(B - Y) \rightarrow U = -0.169R - 0.331G + 0.5B$$

$$V = 0.713(R - Y) \rightarrow V = +0.5R - 0.419G - 0.081B$$

Interazione & Multimedia

Nel caso in cui  $R$ ,  $G$  e  $B$  siano compresi tra 0 e 1, si ottengono valori di  $Y$  compresi tra 0 e 1 e valori di  $U$  e  $V$  compresi tra -0.5 e 0.5.



Nella figura sono rappresentati i colori al variare di  $U$  e  $V$  con  $Y$  fissato a 0.5 (luminanza media).

Quando  $R=G=B$ ,  $U$  e  $V$  valgono 0 e si ottengono solo grigi (nessuna crominanza)



## La luminanza



MEDIA PESATA

$$Y = aR + bG + cB$$

$$a = .3$$

$$b = .6$$

$$c = .1$$



Immagine RGB

I tre canali RGB non danno eguale contributo alla luminanza.  
Il valore di "luminanza" è mantenuto massimamente nel canale G.





## Da YUV a $YC_bC_r$

- Gli spazi  $YC_bC_r$  si possono ottenere facilmente normalizzando ed eventualmente quantizzando i canali di YUV.
- Nel caso in cui R, G e B siano interi compresi tra **0 e 255**, si possono ottenere  $C_b$  e  $C_r$  shiftando U e V. La Y è la stessa di YUV. In questo modo tutti e 3 i canali Y,  $C_b$  e  $C_r$  saranno compresi tra 0 e 255.

$$Y = 0.299R + 0.587G + 0.114B$$

$$C_b = U + 128$$

$$C_r = V + 128$$

- Lo spazio  $YC_bC_r$  è largamente utilizzato nella compressione.  
Noi lo ritroveremo nel formato JPEG.

Si cercano U e V

$$U = 0.564(B - Y) \rightarrow U = -0.169R - 0.331G + 0.5B$$

$$V = 0.713(R - Y) \rightarrow V = +0.5R - 0.419G - 0.081B$$



## Lo spazio $YC_bC_r$

### $YC_bC_r$ :

Y rappresenta la *luminanza*  
mentre  $C_b$  e  $C_r$  rappresentano la  
*crominanza del blu e del rosso*.



# Colori e Memoria

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

0	(0,0,100)
1	(255,255,255)

TIPO di IMMAGINE



## Più colori che pixel!

Una immagine "grande" è di  $2400 \times 1800 = 4.320.000$  pixel.  
Una immagine "media" è di  $1600 \times 1200 = 1.920.000$  pixel.  
Una immagine "piccola" è di  $800 \times 600 = 480.000$  pixel.

In ogni caso ho più colori che pixel!

Inoltre le immagini "naturali" hanno una proprietà di coerenza interna per cui raramente si ha un colore differente per ogni differente pixel.

Questo porta ad adottare la modalità a COLORI INDICIZZATI (indexed color) o a PALETTE o a LOOK-UP-TABLE (LUT).

TAVOLOZZA o

TABELLA di INDICIZZAZIONE → sconsigliato dal formato

VIENE UTILIZZATO DAL ~~FOTOATO~~

~~GIF~~ che USIAMO PER LE ~~GIF~~  
(SEQUENZE di IMM.)

ESEMPIO :



Dovrei ricordare:

255, 0, 0	255, 0, 0	0, 255, 0
255, 0, 0	255, 0, 0	0, 255, 0
0, 0, 255	255, 255, 255	255, 0, 0

Totale (9 pixel x 3 byte) = 27 byte = 216 bit

00	00	01
00	00	01
11	10	00

00 = (255, 0, 0)
01 = (0, 255, 0)
10 = (255, 255, 255)
11 = (0, 0, 255)

Ricordo queste "etichette" e questa tabella

Totale  
18 bit (9 pixel x 2 bit) per l'immagine  
+ 12 byte per la palette  
= 114 bit

Per rappresentare  
un'immagine dettagliata  
a servizio di 256 colori



Si usa una PALETTA CUSTOM  
(ADATTA ALL'IMMAGINE) in due modi

- se nell'immagine true color (16 milioni di colori) ci sono meno di 256 colori, essi vengono replicati.
- se nell'immagine true color ci sono più di 256 colori, essi vengono "ridotti" scegliendo 256 rappresentanti che garantiscono una buona qualità visiva

PALETTA OTTIMIZZATA

Interazione & Multimedia 124



SI PASSA Poi A 16  
colori, MENO DETTAGLIATI.

ESEMPIO - DIFFERENZA



A 8 colori e così  
via...

## IL RE-INDEXING

Cambiare la posizione della palette e quindi anche l'indice corrispondente in modo da creare una matrice di indici che abbia l'entropia minima.

### SPIEGAZIONE

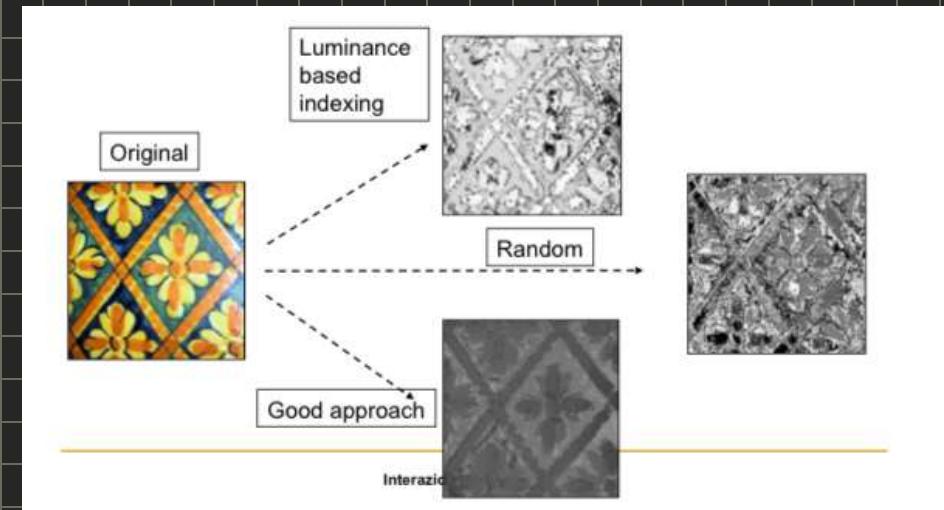
L'idea generale è che se ai colori di pixel adiacenti associo indici numericamente vicini, la differenza tra tali indici diventa più piccola.

- Questo permette di **diminuire l'entropia** dell'immagine, garantendo un miglior rapporto di compressione quando si utilizzano formati che usano la codifica differenziale (che vedremo più avanti nel corso)

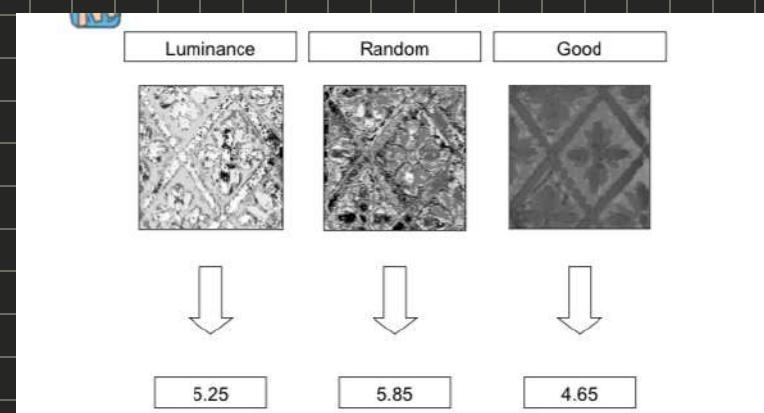
- Purtroppo, **trovare la disposizione ottimale (quella ad entropia minima)** è un problema NP-hard. Per una palette con **M colori**, bisogna cercare tra  **$M!$  possibili ordinamenti.**

SERVONO \*

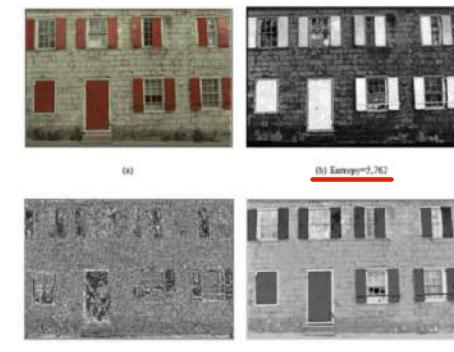
- Pertanto, gli algoritmi di reindexing si basano sulla ricerca di soluzioni che si avvicinano a quella ottima.



MINIMIZZARE  
L'ENTROPIA



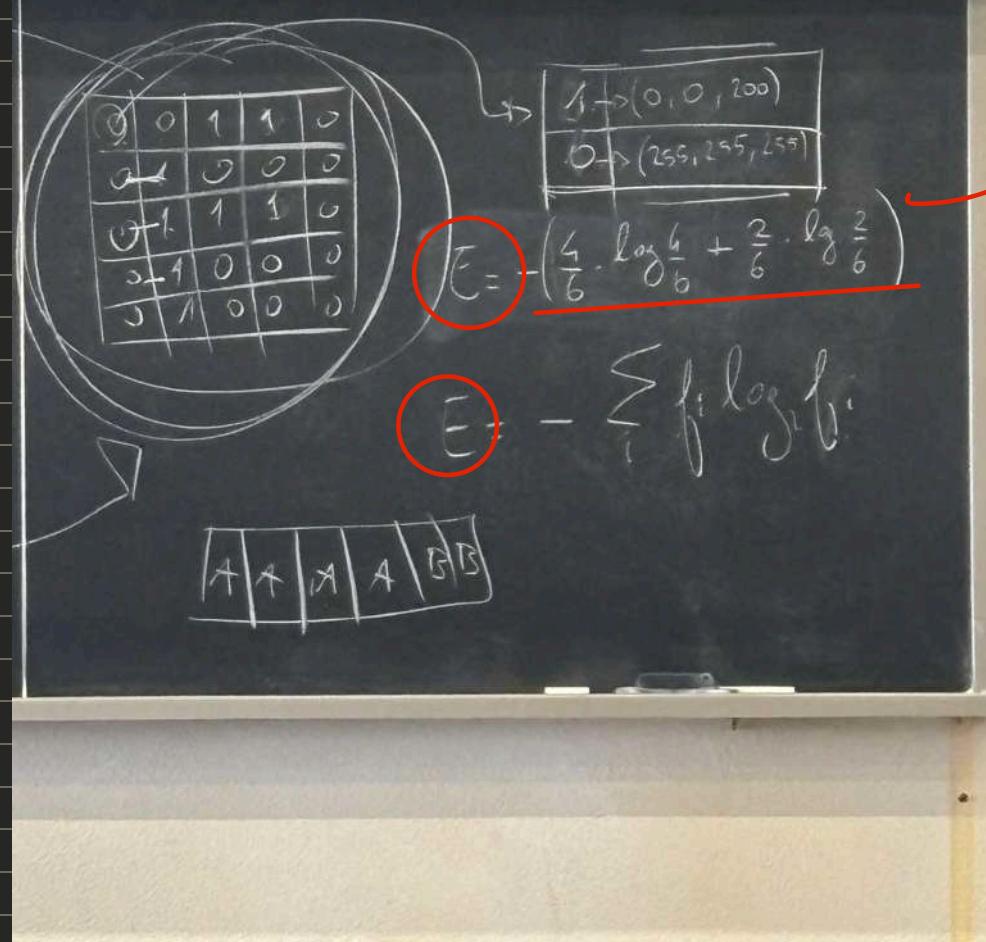
ES:



MANCA COHERENZA  
NON SI CAPISCE

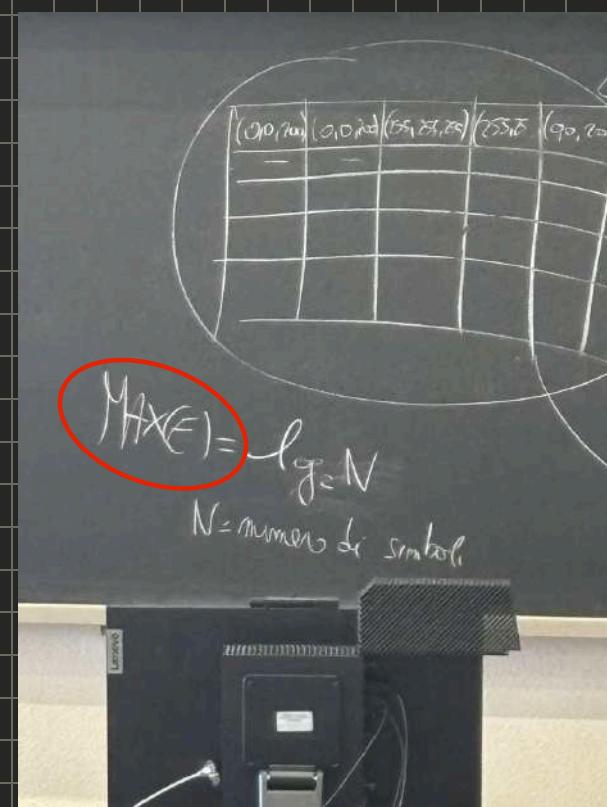
RIUSC. A CAPIRE L'IMM.  
SCELTA OTTIMA

# CALCOLO ENTROPIA di UNA SEQUENZA

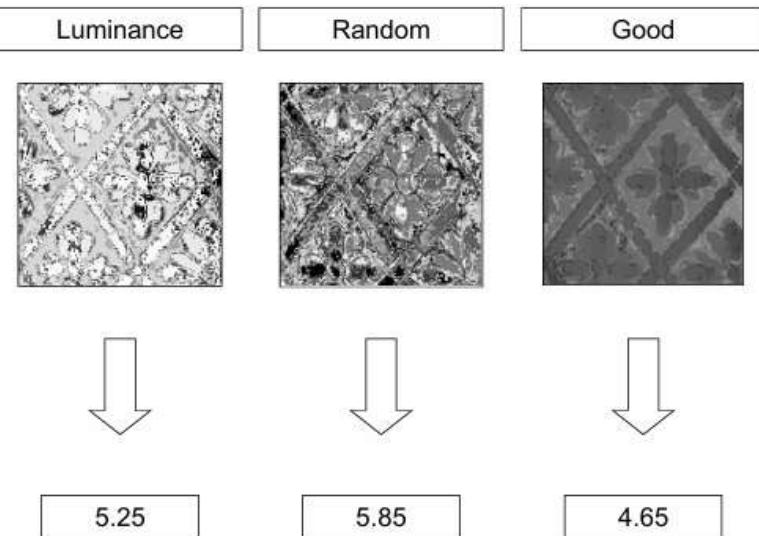


$< 1$  (MINIMA ENTROPIA)

MASSIMA ENTROPIA



### Esempio



- se uso l'ordinamento a sinistra otterò di media 5.25 bit
- se uso l'ordinamento al centro otterò di media 5.85 bit
- se uso l'ordinamento a destra otterò di media 4.65 bit  
se i valori sono simili gli indici che gli assegno devono essere simili

Formula reindexing

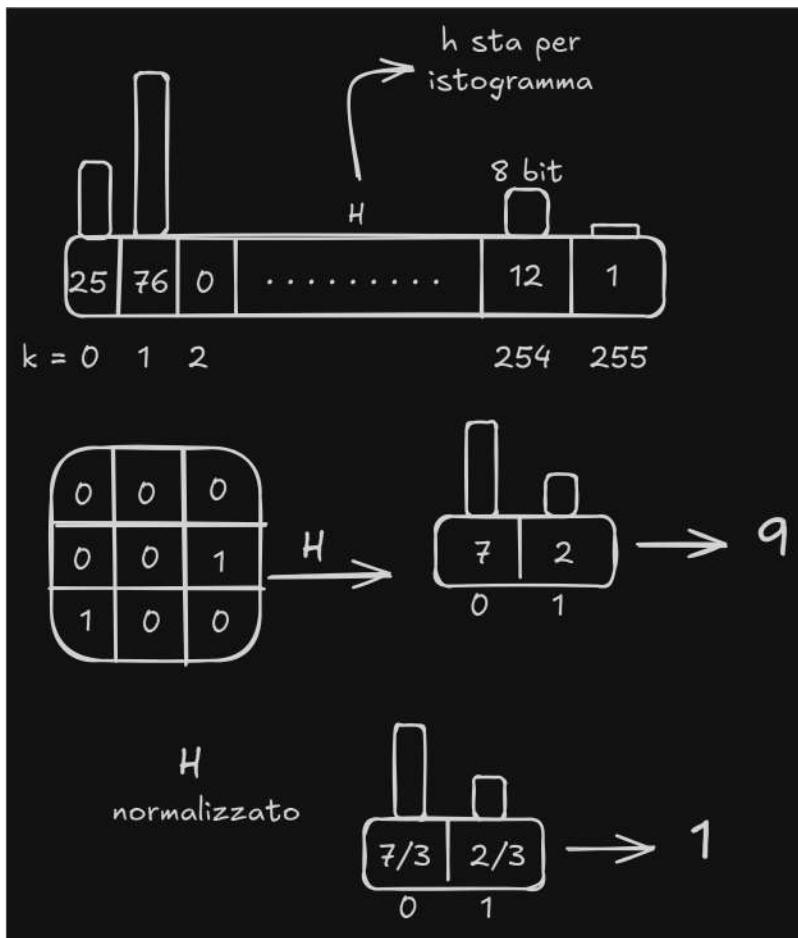
(foto massima e minima entropia)

## PDF. 9 - OPERAZIONI PUNTUALI

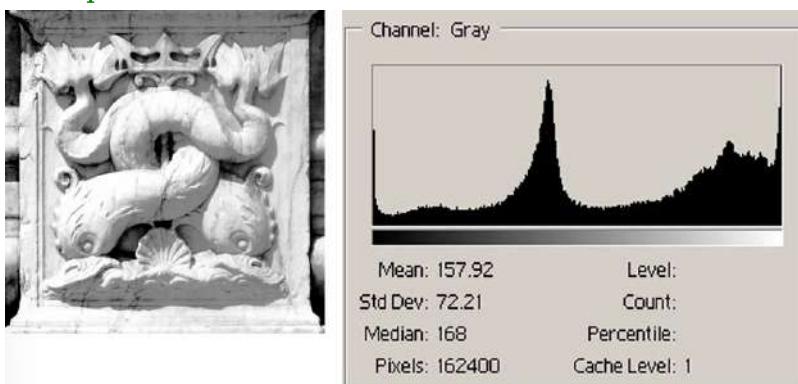
### Iistogramma

I'istogramma sono dei grafici che descrivono una quantità, il numero di occorrenze in un determinato tempo, quindi ho una serie di eventi e posso rappresentare questa statistica con l'istogramma.

I pixel di una immagine sono una "popolazione" sulla quale possiamo calcolare tutte le quantità descrittive che si usano normalmente: Media, mediana, varianza, ecc... Particolarmente importante è la **conoscenza della distribuzione delle frequenze dei toni di grigio**.



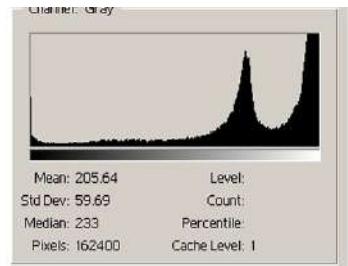
### Esempio



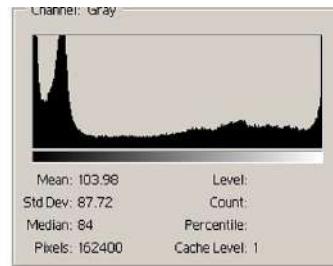
Immagini diverse potrebbero avere istogrammi simili in quanto L'istogramma non tiene conto della distribuzione spaziale dei pixel!

infatti se abbiamo 2 istogrammi diverse abbiamo 2 immagini certamente diverse se invece abbiamo 2 istogrammi uguali non possiamo dire che anche le immagini sono uguali

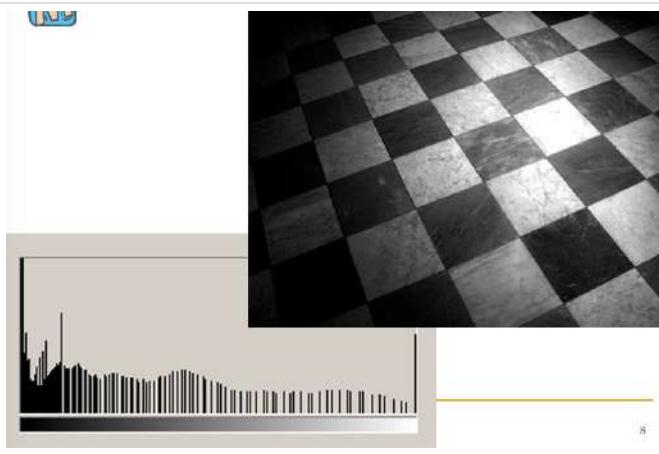
**immagine chiara, abbiamo un istogramma più denso a destra**



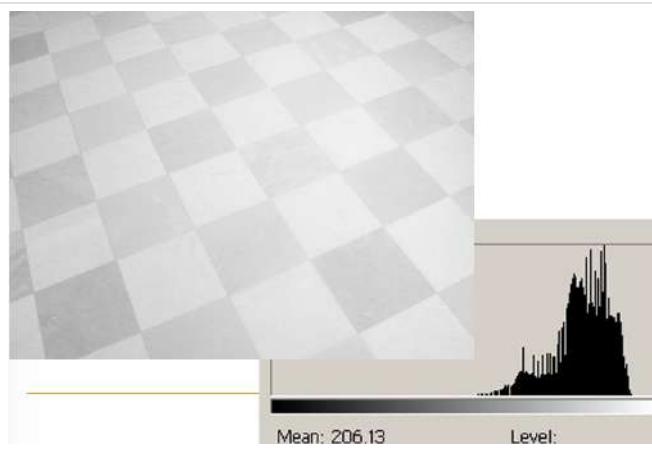
**immagine scura, abbiamo un istogramma più denso a sinistra**



**immagine sottoesposta**



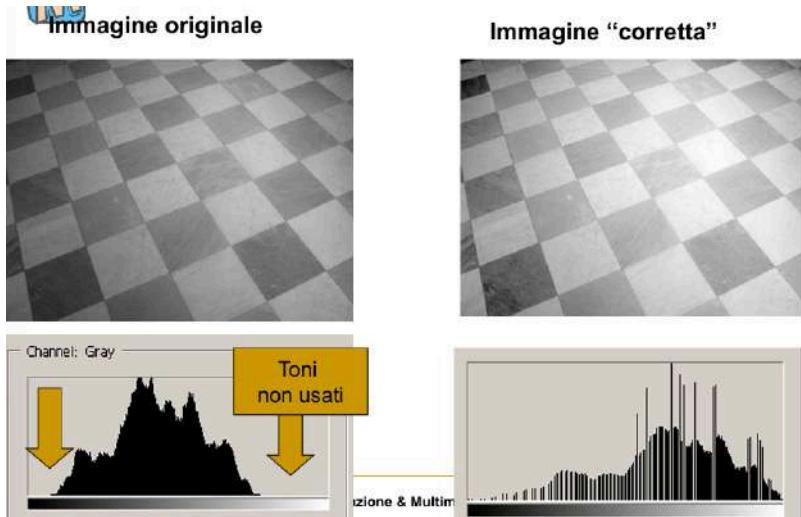
**immagine sovraesposta**



### Espansione del contrasto (contrast stretching)

Serve per aumentare la dinamica di un'immagine il cui istogramma è concentrato su un intervallo limitato dei valori possibili. Si ottiene spostando (con appositi algoritmi) i valori di un bin dell'istogramma verso un altro bin non utilizzato. L'istogramma apparirà in maniera differente, tipo pettine. Ciò è fatto per mettere in risalto che i bin mancanti sono stati distribuiti lungo altri livelli

## Contrast stretching, Esempio



nell'immagine corretta l'istogramma è stato tirato

## Aritmetica sulle immagini

Operando aritmeticamente può accadere che un pixel abbia: a) Un valore negativo; b) Un valore maggiore del massimo (tipicamente 255); c) Un valore non intero (facilmente risolubile con una approssimazione o un troncamento)

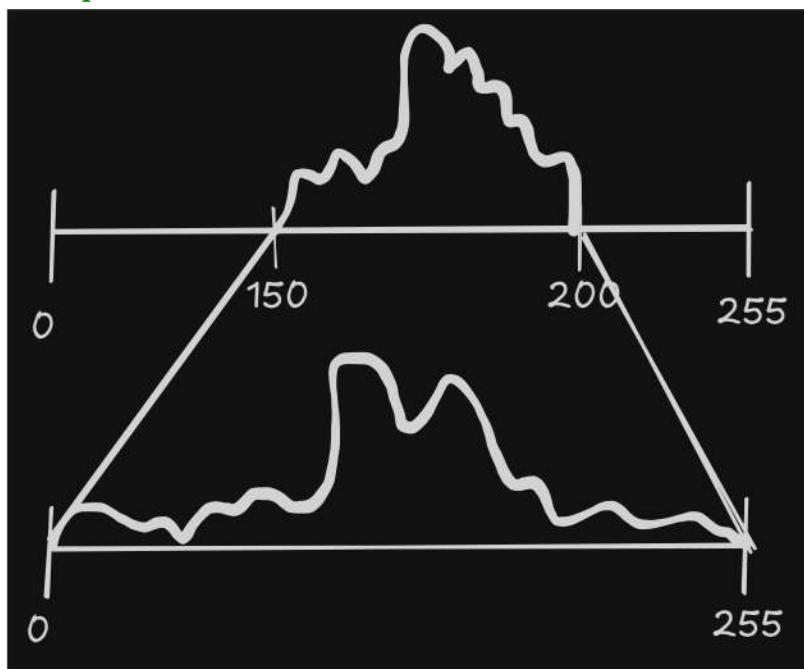
I problemi a) e b) della precedente slide si chiamano problemi di range. Due le soluzioni più comuni:

- Settare a 0 (nero) i valori negativi e a 255 (bianco) i valori maggiori di 255.
- Ri-normalizzare il range trasformando ciascun valore secondo la equazione

## Formula

$$y = \frac{x - min}{max - min} \cdot 255$$

## Esempio Grafico



## Esempio Pratico

150	150	200
175	155	190
190	200	180

contrast  
stretched  
lineare

0	0	255
128	26	204
204	255	153

applichiamo questa formula

$$v_{\text{nuovo}} = 255 * \frac{(v_{\text{vecchio}} - \min_{\text{osservato}})}{(\max_{\text{osservato}} - \min_{\text{osservato}})}$$

$$\frac{150-150}{50} * 255 = 0$$

$$\frac{200-150}{50} * 255 = 255$$

$$\frac{175-150}{50} * 255 = 128$$

$$\frac{155-150}{50} * 255 = 26$$

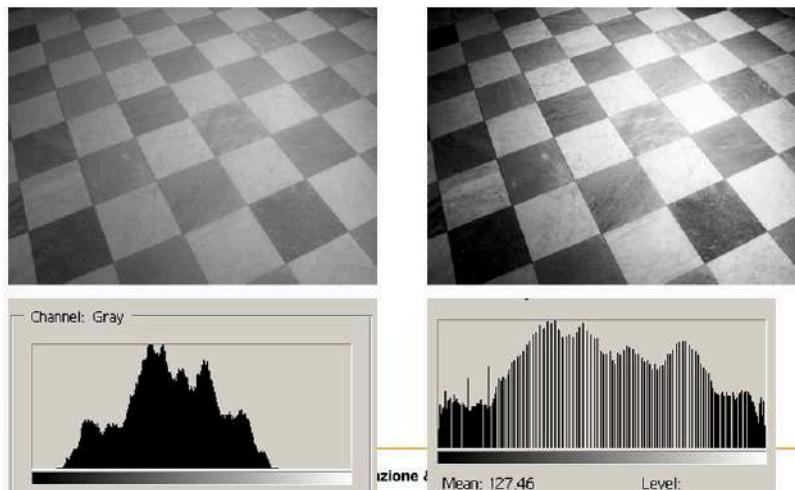
$$\frac{190-150}{50} * 255 = 204$$

$$\frac{180-150}{50} * 255 = 153$$

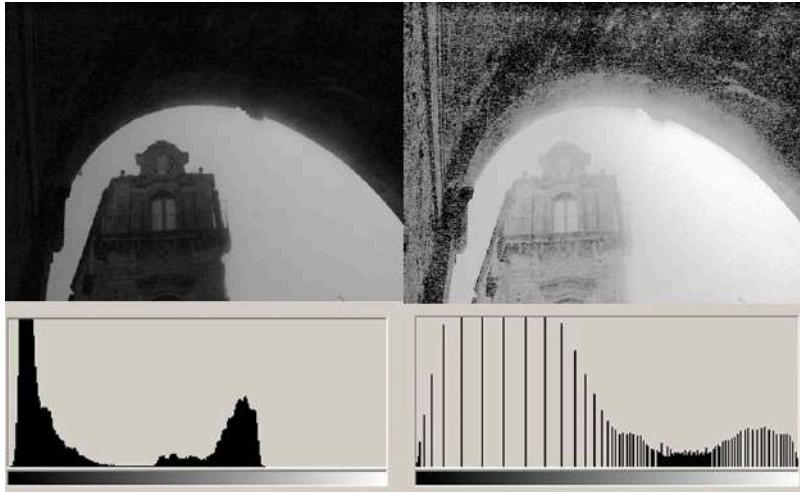
## Equalizzazione

Si parla di immagine equalizzata quando il contributo di ogni differente tonalità di grigio è pressappoco eguale. È detto anche "istogramma" uniforme o appiattito. L'equalizzazione si ottiene usando appositi algoritmi ma, **Attenzione**, non sempre la equalizzazione migliora l'immagine!

Esempio imm. migliorata (a destra l'immagine equalizzata)



Esempio imm. peggiorata (a destra l'immagine equalizzata)



## Algoritmo di Equalizzazione

Se  $r_k$  è un livello di grigio e  $n_k$  il numero di pixel nell'immagine  $M \times N$  di quel livello di grigio, si può definire

$$p_r(r_k) = \frac{n_k}{MN} \quad k = 0, 1, 2, \dots, L - 1$$

Se facciamo il plot di  $r_k$  versus  $p_r(r_k)$  quello che si ottiene è l'istogramma dell'immagine, i nuovi valori di grigio dell'istogramma sono così definiti:

$$\begin{aligned} s_k &= T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) \\ &= \frac{L(-1)}{MN} \sum_{j=0}^k n_j \quad k = 0, 1, 2, \dots, L - 1 \end{aligned}$$

### Esempio

Sia data una immagine a 3 bit ( $L=8$ ) con  $64 \times 64$  pixel ( $MN=4096$ ) con la seguente distribuzione di intensità:

$r_k$	$n_k$	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Applicando la formula si ha:

$$s_0 = T(r_0) = 7 \sum_{j=0}^0 p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7 \sum_{j=0}^1 p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

$$s_2 = 4.55, s_3 = 5.67, s_4 = 6.23, s_5 = 6.65, s_6 = 6.86, s_7 = 7.00$$

Arrotondando:

$$s_0 = 1.33 \rightarrow 1 \quad s_4 = 6.23 \rightarrow 1$$

$$s_1 = 3.08 \rightarrow 1 \quad s_5 = 6.65 \rightarrow 1$$

$$s_2 = 4.55 \rightarrow 1 \quad s_6 = 6.86 \rightarrow 1$$

$$s_3 = 5.67 \rightarrow 1 \quad s_7 = 7.00 \rightarrow 1$$

## Operazioni Sulle Immagini

Per semplificare la trattazione del problema lavoreremo solo su immagini a toni di grigio.

*Formula*

$$g(x, y) = T[f(x, y)]$$

f l'immagine di ingresso alla elaborazione, g quella di uscita e T un operatore su f definito in un intorno di (x,y).

quando g(x,y) dipende SOLO da f(x,y) BHOOOO

abbiamo **3 Tipi di operazioni** dove la dimensione dell'intorno di (x,y) definisce il carattere della elaborazione:

- *puntuale* (l'intorno coincide con il pixel stesso);
- *locale* (per esempio una piccola regione quadrata centrata sul pixel);
- *globale* (l'intorno coincide con l'intera f).

## OPERAZIONI PUNTUALI

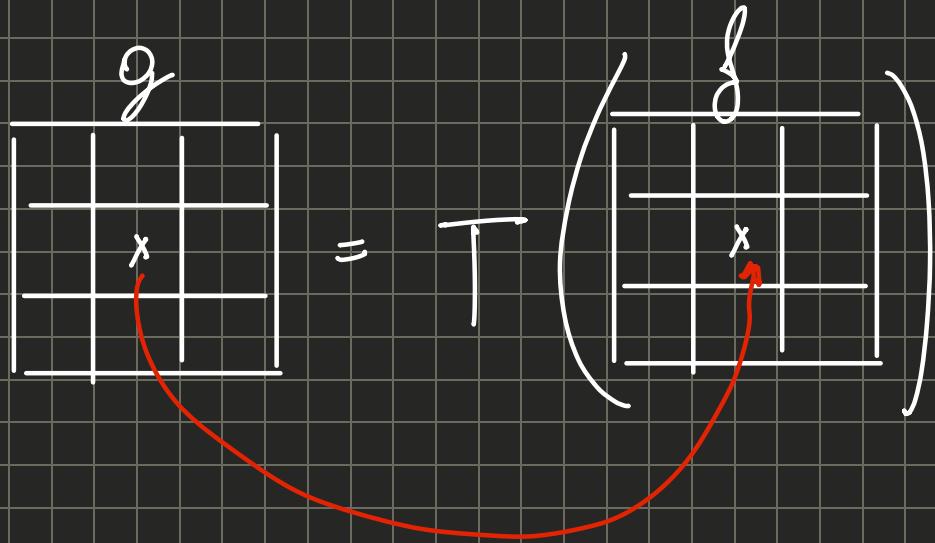
$$g(x,y) = T[f(x,y)] \Rightarrow f(x+1) + 10$$

AGGIUNTA  
(o SOTTRAZ.)

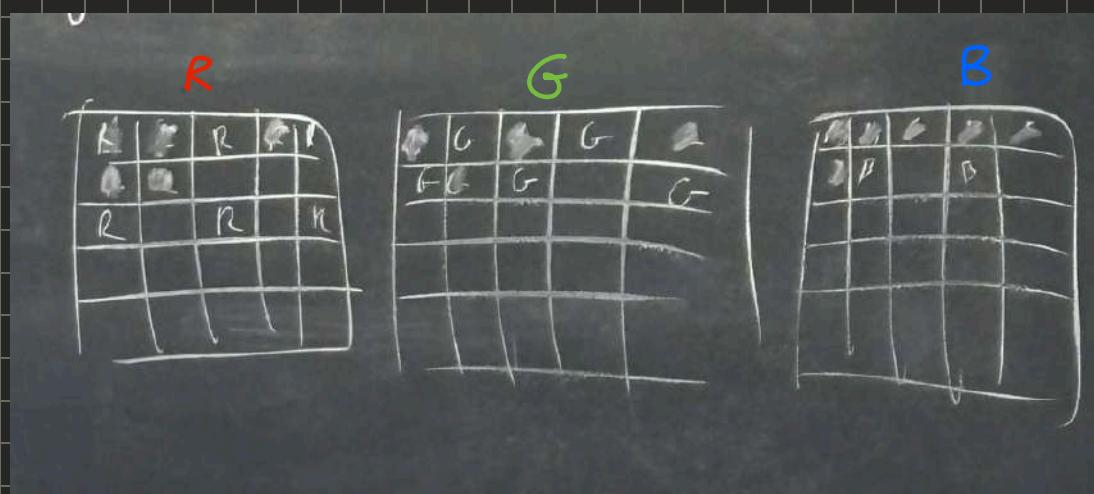
di UNA COSTANTE

NEGATIVO (8 bit)

$$g(x,y) = 255 - f(x,y)$$



ES:

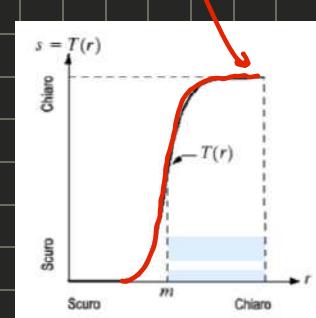


NEUTRINO

$f$	$g$
0	255
1	254
2	253
...	...
255	0

CUT  
Look-OP-TABLE

CURVA AUM. CONTRASTO



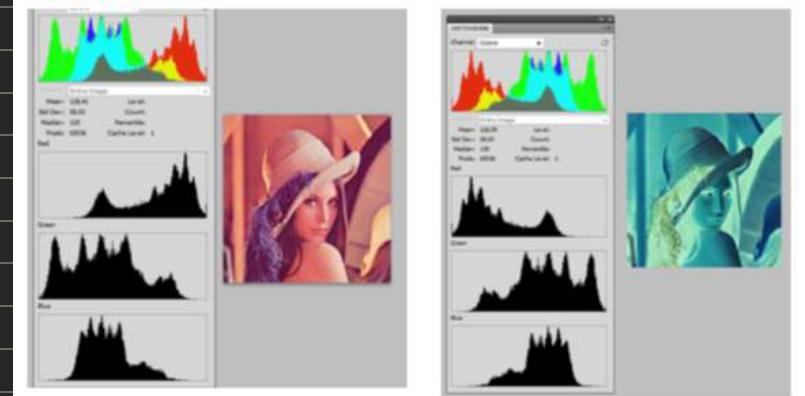
Ho AUMENT.  
I TONI CHIARI

## ESEMPIO di NEGATIVO (INVERT)

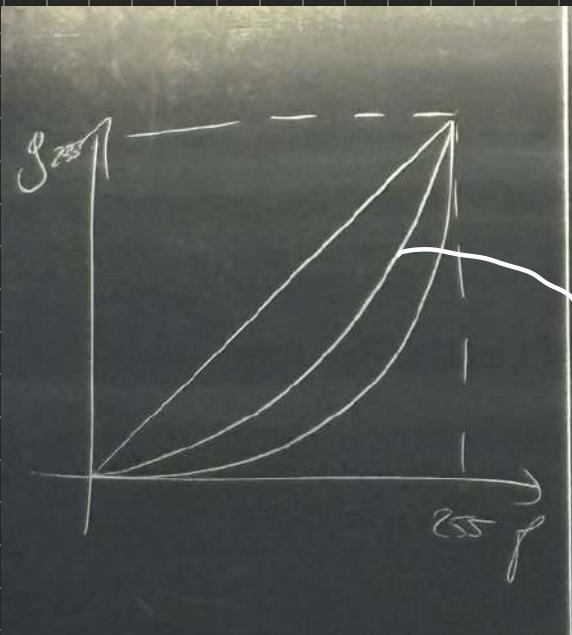
- Consiste nell'associare al valore  $f(x,y)$  del pixel il valore  **$255-f(x,y)$**



ES:

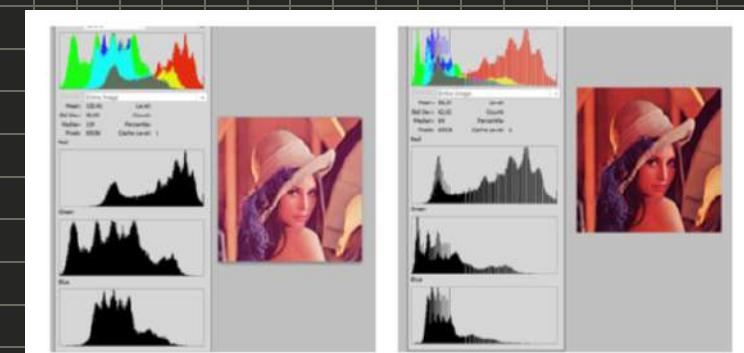


## INCUPIMENTO IMMAGINE



Piú va a DESTRA,  
piú l'incupimento  
aumenta

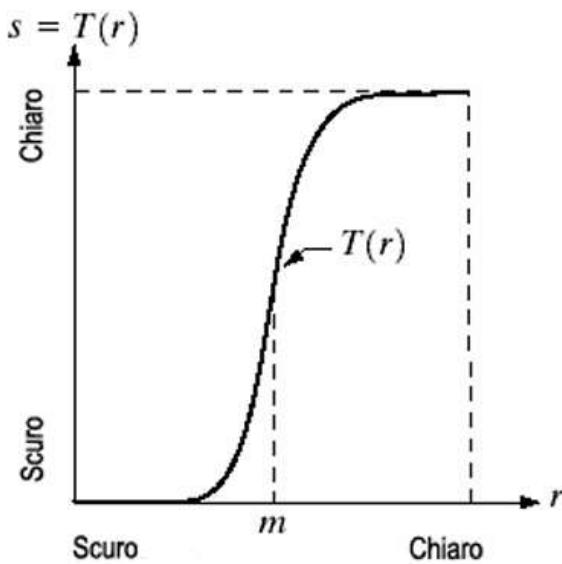
$$g = f - K \quad \left\{ \begin{array}{l} g > K \\ \text{o altrimenti} \end{array} \right.$$



1	2	3
4	5	6
7	8	9
		1 2 3
		4 5 6
		7 8 9

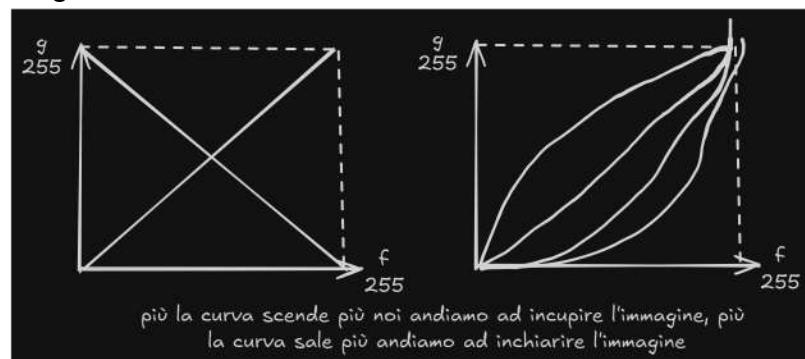
## LUT (SCHIARIMENTO)

Questo tipo di grafico si chiama look-up tables (LUT).

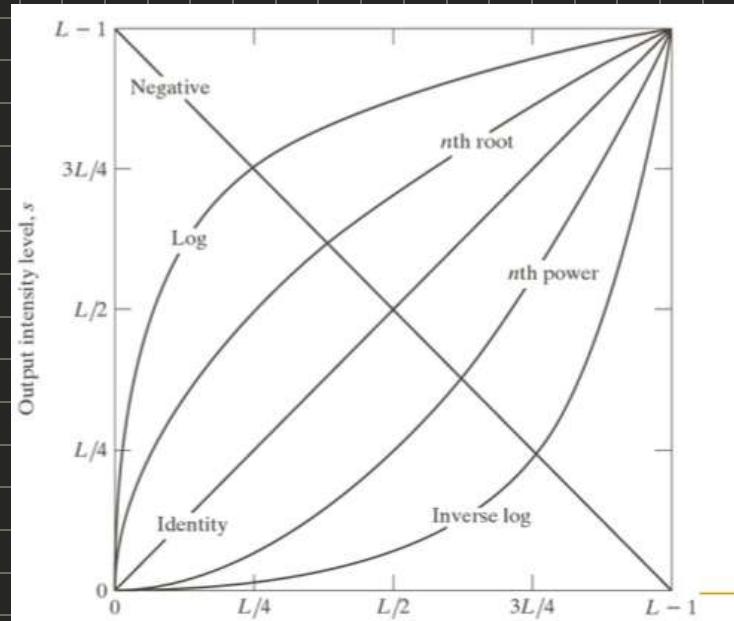


questa operazione fa sì che i colori scuri diventano più scuri e i colori chiari diventano più chiari e infine i colori medi sono stati diminuiti, in altre parole è una curva che incrementa il contrasto.

Negativo



# TRANSFORMAZIONE LOGARITMICA

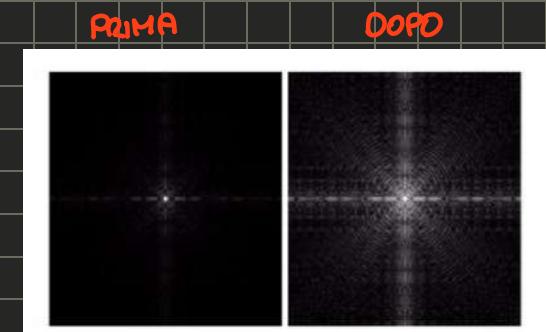


consente di **comprimere la gamma dinamica**, permettendo la **memorizzazione** o la **visualizzazione**, con una scala dei grigi usuale, di immagini caratterizzate da escursioni di intensità molto ampie. Può essere espressa come:

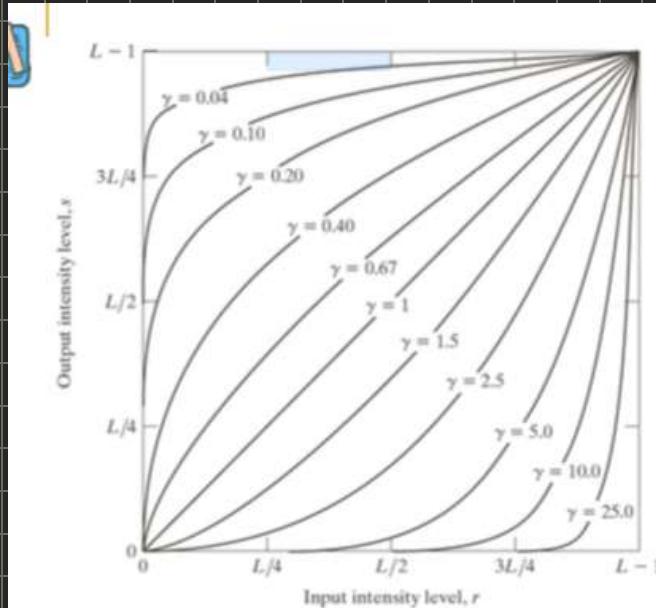
$$g(x,y) = c \log(1 + f(x,y))$$

- $f \in [0, 255]$
- $1+f \in [1, 256]$
- $\log_2(1+f) \in [0, 8]$
- $c \log_2(1+f) \in [0, 255]$

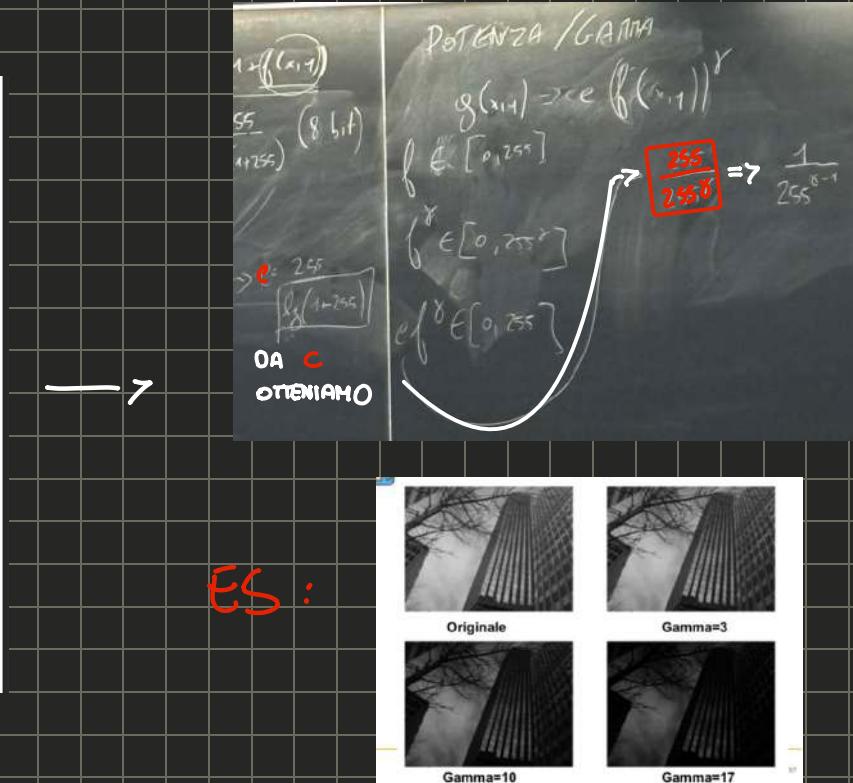
$$c = \frac{255}{\log(1+255)}$$



# TRASFORMAZIONE di POTENZA

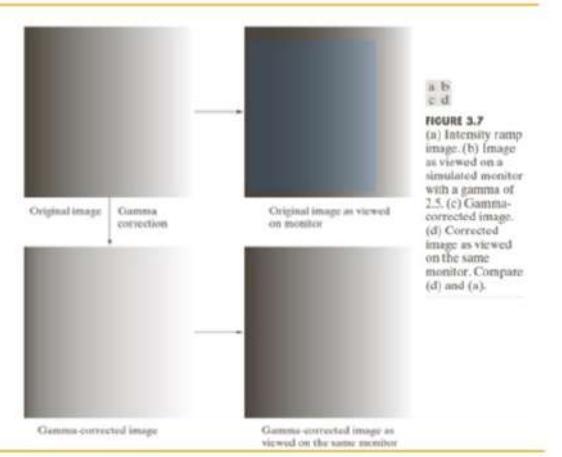


**FIGURE 3.6** Plots of the equation  $s = cr^\gamma$  for various values of  $\gamma$  ( $c = 1$  in all cases). All curves were scaled to fit in the range shown.

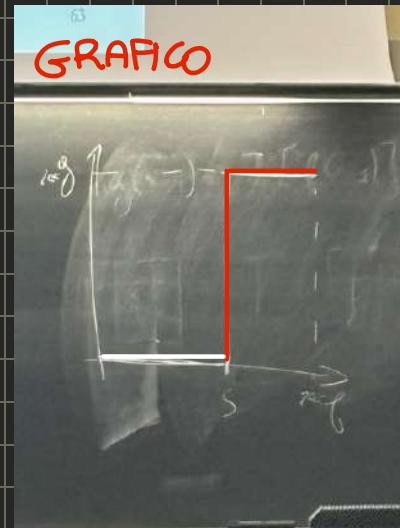


ES:

## APPLICAZIONE di GAMMA NELLA RESA dei MONITOR



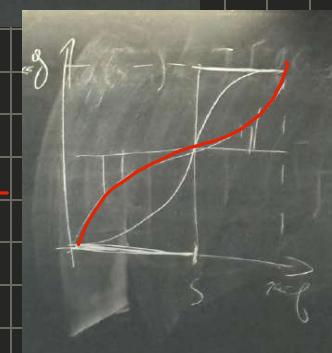
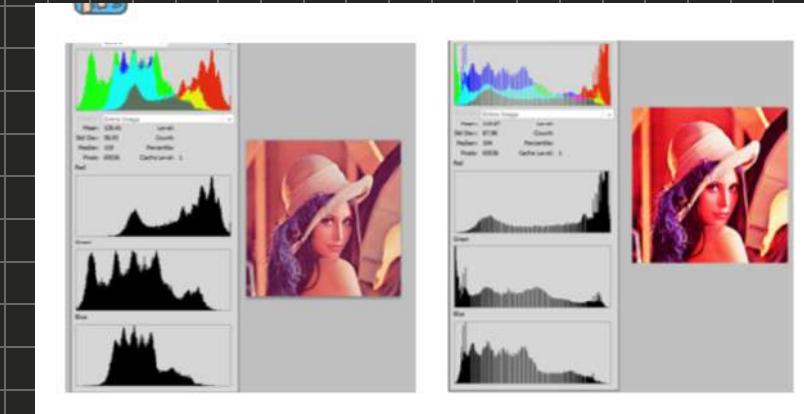
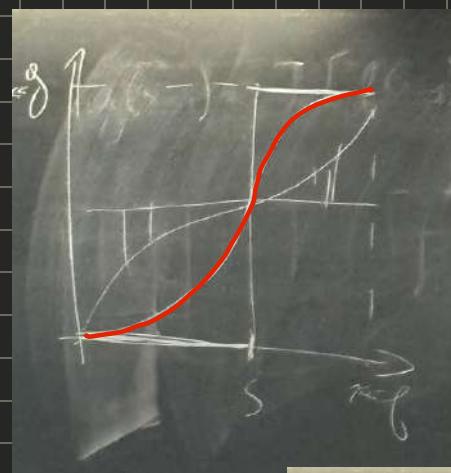
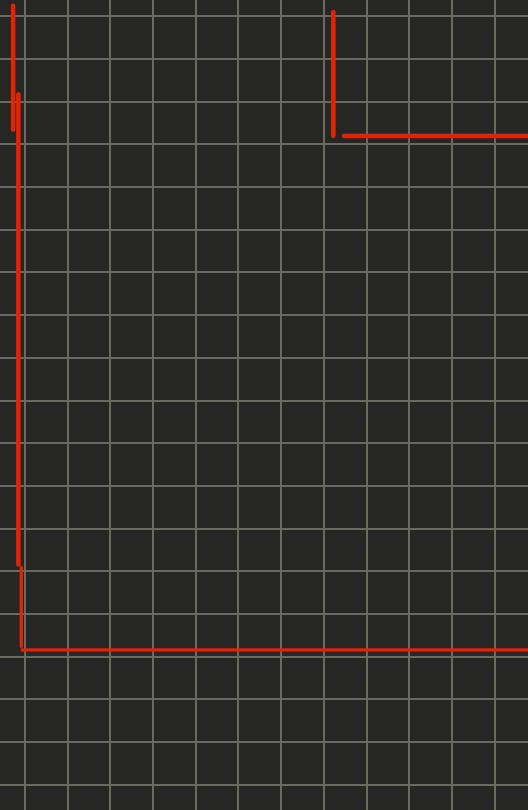
Su un monitor CRT (con  $\gamma = 2.5$ ) si può applicare una correzione pre-processando l'input con la corrispondente funzione inversa:  $g(x,y) = f(x,y)1/2.5 = f(x,y)0.4$



Produce una immagine che ha solo due livelli: nero e bianco.

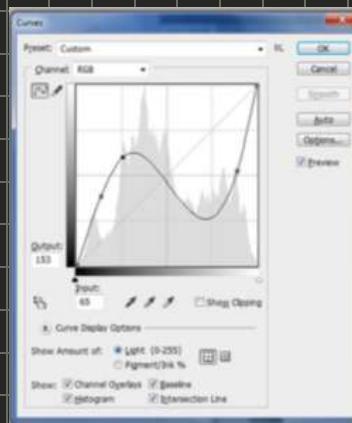
- Si ottiene scegliendo una soglia  $T$  e mettendo a nero tutti i pixel il cui valore è minore a  $T$  e a bianco tutti gli altri.

## AUMENTO / DIMINUZIONE CONTRASTO

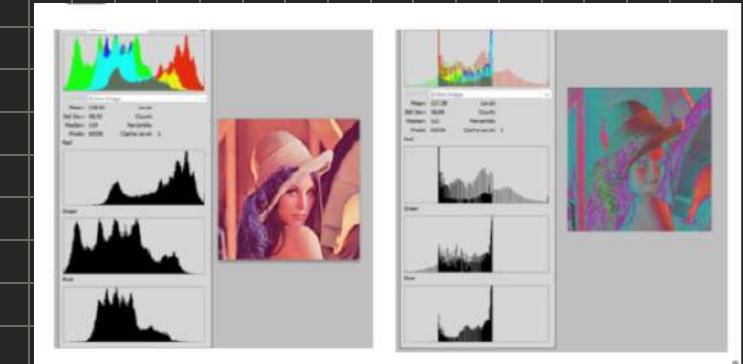


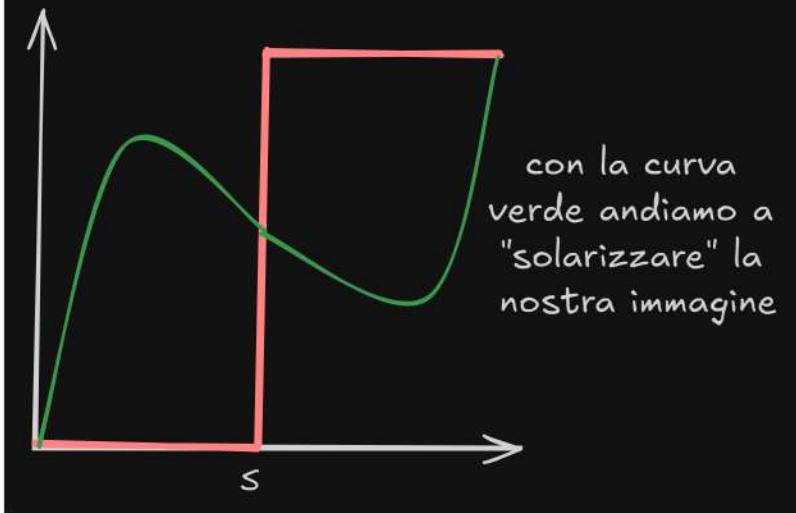
## CURVE NON MONOTONE

*È possibile fare delle variazioni alle curve in modo che questa diventi non monotona.*



Un esempio è la **solarizzazione**





## Base canonica

PDF 10. OPERAZIONI COGLI  
E CONVOLUZIONE

Dato un vettore di lunghezza N, questo può essere pensato come un elemento di uno spazio N dimensionale.

234	204	34	16	44	134	12	11	56
-----	-----	----	----	----	-----	----	----	----

Quindi possiamo scomporlo usando la base canonica di tale spazio. Qual è la base canonica?



<b>M</b>	<table border="1"><tr><td>234</td><td>204</td><td>34</td><td>16</td><td>44</td><td>134</td><td>12</td><td>11</td><td>56</td></tr></table>	234	204	34	16	44	134	12	11	56	=	<b>E</b>	<table border="1"><tr><td>234</td><td>204</td><td>34</td></tr><tr><td>16</td><td>44</td><td>134</td></tr><tr><td>12</td><td>11</td><td>56</td></tr></table>	234	204	34	16	44	134	12	11	56	=																					
234	204	34	16	44	134	12	11	56																																				
234	204	34																																										
16	44	134																																										
12	11	56																																										
234 *	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	1	0	0	0	0	0	0	0	0	+	234 *	<table border="1"><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	1	0	0	0	0	0	0	0	0	+ 204 *	<table border="1"><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	1	0	0	0	0	0	0	0	+ 34 *	<table border="1"><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0																																				
1	0	0																																										
0	0	0																																										
0	0	0																																										
0	1	0																																										
0	0	0																																										
0	0	0																																										
0	0	1																																										
0	0	0																																										
0	0	0																																										
204 *	<table border="1"><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	1	0	0	0	0	0	0	0	+	16 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	1	0	0	0	0	0	+ 44 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	1	0	0	0	0	+ 134 *	<table border="1"><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0																																				
0	0	0																																										
1	0	0																																										
0	0	0																																										
0	0	0																																										
0	1	0																																										
0	0	0																																										
0	0	1																																										
0	0	0																																										
0	0	0																																										
34 *	<table border="1"><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	1	0	0	0	0	0	0	+	12 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	0	+ 11 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	1	0	0	+ 56 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table>	0	0	0	0	0	0	0	1	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																																										
1	0	0																																										
0	0	0																																										
0	0	0																																										
0	1	0																																										
16 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	1	0	0	0	0	0	+																																	
0	0	0	1	0	0	0	0	0																																				
44 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	1	0	0	0	0	+																																	
0	0	0	0	1	0	0	0	0																																				
134 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	1	0	0	0	+																																	
0	0	0	0	0	1	0	0	0																																				
12 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	1	0	0	+																																	
0	0	0	0	0	0	1	0	0																																				
11 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	0	0	0	0	0	1	0	+																																	
0	0	0	0	0	0	0	1	0																																				
56 *	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	0	0	0	0	0	0	0	0	1	+																																	
0	0	0	0	0	0	0	0	1																																				

## Operatori locali

Il valore d'uscita di ogni pixel dipende da un limitato intorno del corrispondente punto in input. Sono usati per migliorare la qualità delle immagini o per estrarre delle informazioni dall'immagine. Si possono pensare come filtraggi dell'immagine.

## Operatori Lineari

Un operatore  $F : V \rightarrow W$  si dice LINEARE se per ogni coppia di vettori  $v_1$  e  $v_2$  in  $V$  e per ogni coppia di scalari  $a, b$  si ha che:

$$F(a v_1 + b v_2) = aF(v_1) + bF(v_2)$$

Conseguenza : se conosco una base di  $V$  ed il comportamento dell'operatore  $F$  su ogni elemento di tale base, posso calcolare il comportamento di  $F$  su ogni elemento di  $V$ .

questa funzione è lineare?  $f(x, y) = (\frac{x}{2}, \frac{y}{3})$  SI

$$\begin{aligned} & \textcircled{2} \quad 1^{\circ} \text{ MEMBRO : } a * f(x_1, y_1) + b * f(x_2, y_2) = \underline{f(ax_1 + bx_2, ay_1 + by_2)} \\ & \textcircled{3} \quad 2^{\circ} \text{ MEMBRO : } a * (\frac{x_1}{2}, \frac{y_1}{3}) + b * (\frac{x_2}{2}, \frac{y_2}{3}) = \underline{(ax_1/2 + bx_2/2, ay_1/3 + by_2/3)} \\ & \quad \quad \quad \textcircled{4} \quad \underline{(\frac{a(x_1 + bx_2)}{2}, \frac{(ay_1 + by_2)}{3})} \end{aligned}$$

Si deve  
verificare questa  
uguaglianza

1 2 membri sono uguali e quindi

la funzione è lineare

## Invariante per traslazione?

- Operatori non invarianti per traslazione:

Il comportamento varia a seconda della posizione nell'immagine. Per descriverli, bisogna studiare ogni impulso in ogni posizione. Non sono "cattivi", ma complessi da analizzare.

1	2	3						
4	5	6						
7	8	9						
			1	2	3			
			4	5	6			
			7	8	9			

il risultato qui è uguale anche qui

- **Operatori invarianti per traslazione (shift invariant):**

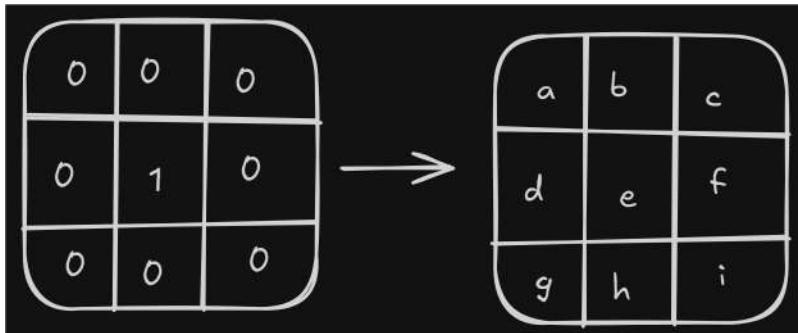
Il comportamento sugli impulsi è **sempre lo stesso**, indipendentemente dalla posizione.

tutti gli **operatori puntuali** sono sempre invarianti per traslazione, anche se non lineari.

Riassumendo

- Se  $F$  è lineare per descriverlo basta conoscere il comportamento su tutte le immagini impulsive
- Se  $F$  è shift invariant si comporta allo stesso modo su tutti gli impulsi, indipendentemente dalla loro posizione
- Se  $F$  è sia lineare che shift invariant per descriverlo basta conoscere come si comporta su un solo impulso.

La "risposta all'impulso" o "point spread function" di  $F$  è la carta d'identità di tale operatore



### SPIEGAZIONE :



Ad un operatore lineare e shift invariante corrisponde una maschera ma vale anche il viceversa: ad una maschera corrisponde un simile operatore

#### ESEMPIO

Si consideri l'operazione che preso un impulso:

	0 0 0 0 1 0 0 0 0	lo trasforma in:	0 0 0 0.5 0 0.5 0 0 0	
--	-------------------------	------------------	-----------------------------	--

Tale "risposta all'impulso" o PSF definisce completamente un operatore lineare e invariante per traslazione  $F$ . Spesso un operatore su una immagine prende il nome di "filtro".

La matrice che descrive la risposta all'impulso si chiama anche **kernel** o maschera dell'operatore.

Essa è detta anche **maschera di convoluzione** di  $F$  per ragioni che vedremo tra breve.

## Kernel finiti o infiniti e complessità

La grandezza del kernel può variare fino ad essere infinita ovvero che la risposta all'impulso è grande infinite volte, ad esempio:





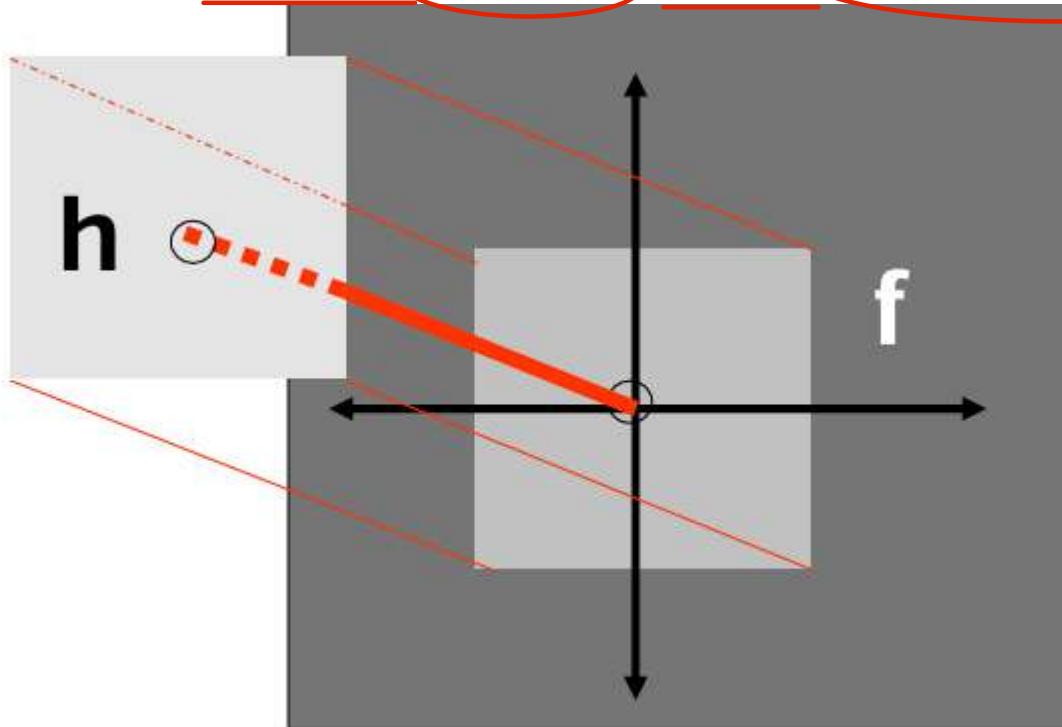
## - Convoluzione

I filtri lineari e invarianti per traslazione vengono chiamati anche filtri convolutivi. Dobbiamo studiare la operazione di convoluzione per capire meglio come un filtro può essere calcolato

*proprietà*

Per indicare l'operazione di convoluzione si usa la notazione  $g = f \oplus h$ .

La convoluzione è commutativa  $f \oplus h = h \oplus f$  e associativa  $(f \oplus h) \oplus h1 = f \oplus (h \oplus h1)$



### Nel caso finito (1)

Se il kernel  $h$  ha dimensioni  $s \times t$  la formula va riscritta nella seguente maniera

$$g_{m,n} = \sum_{i=-\lfloor s/2 \rfloor}^{\lfloor s/2 \rfloor-1} \sum_{j=-\lfloor t/2 \rfloor}^{\lfloor t/2 \rfloor-1} (h_{i,j} * f_{m+i,n+j})$$

	-1	0	1
-1	a	b	c
0	d	e	f
1	g	h	i

Se gli indici del kernel sono disposti da avere il punto di coordinate  $(0,0)$  nella posizione centrale.

1	2	3	4	5
2	30			
3				
4				
5				

## Nel caso finito (2)

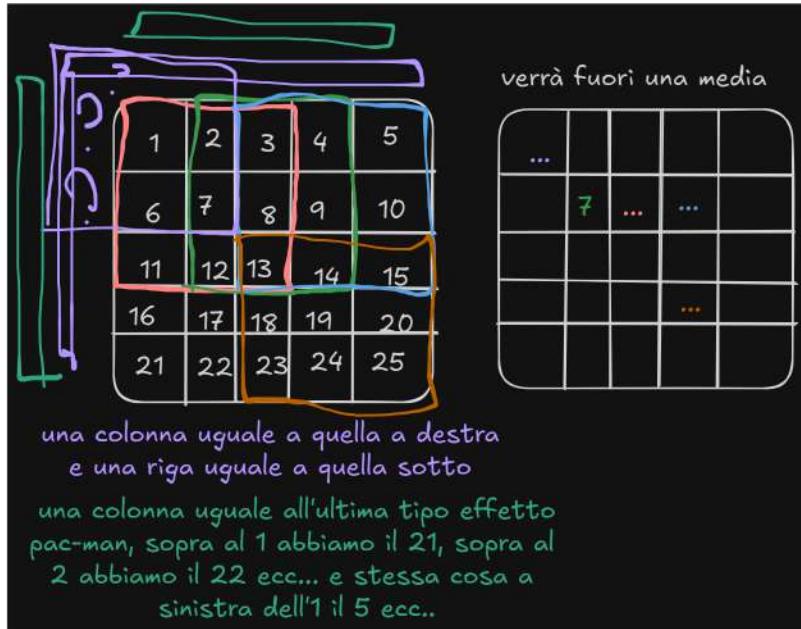
Se il kernel  $h$  ha dimensioni  $s \times t$  la formula va riscritta nella seguente maniera.

$$g_{m,n} = \sum_{i=1,j=1}^{s,t} h_{i,j} * f_{m+(i-s+\lfloor s/2 \rfloor), n+(j-t+\lfloor t/2 \rfloor)}$$

1	2	3
a	b	c
d	e	f
g	h	i

Se gli indici del kernel sono disposti partendo da 1 fino ad arrivare a  $s$  e  $t$

### ESEMPIO PROF



praticamente la media è:

$$(1 + \frac{1}{9}) + (2 + \frac{1}{9}) + (3 + \frac{1}{9}) + (6 + \frac{1}{9}) + (7 + \frac{1}{9}) + (8 + \frac{1}{9}) + (11 + \frac{1}{9}) + (12 + \frac{1}{9}) + (13 + \frac{1}{9})$$

se prendiamo il punto marrone è:

$$\frac{13}{9} + \dots$$

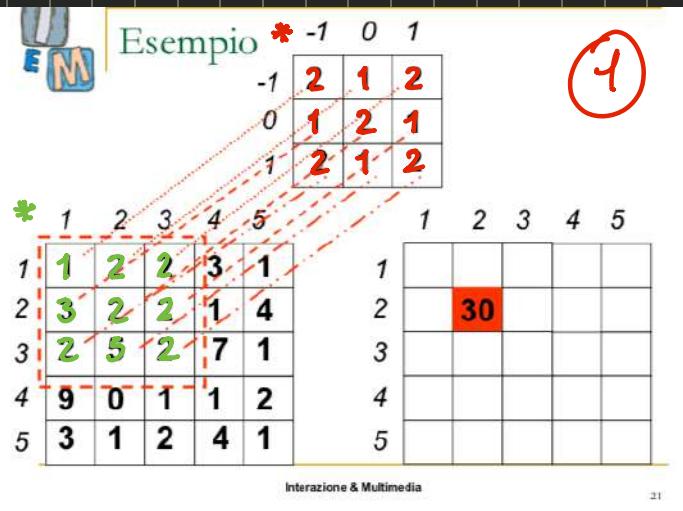
come fare per i bordi?

POSSIBILI SOLUZIONI:

- Filtrare solo le zone centrali dell'immagine
- Supporre che tutto intorno all'immagine ci sia 0 (alcune volte a senso altre no)
- Assumere una topologia "toroidale": quando si "sfiora a destra" si rientra a sinistra, quando si "sfiora" in basso si rientra in alto e viceversa;
- Aggiungere una riga all'inizio uguale alle righe precedenti, una riga alla fine uguale all'ultima riga, una colonna all'inizio uguale alla colonna iniziale, e una colonna alla fine



## ESEMPPIO CONVOLUZIONE



Moltiplico ogni indice della MATRICE di INPUT con la "MATRICE" KERNEL, quindi

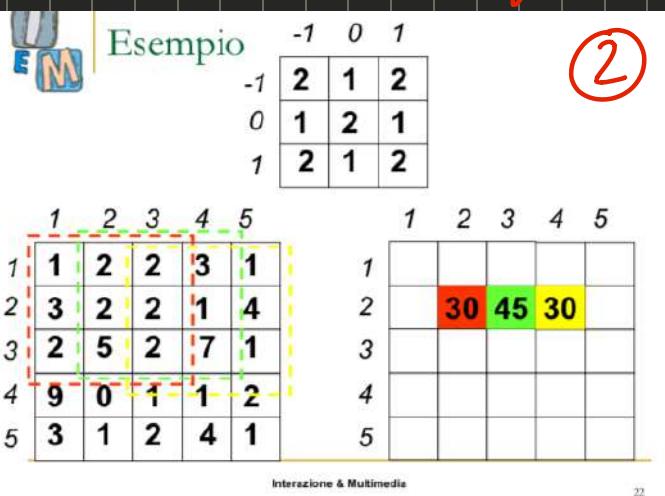
$$(1 \cdot 2) + (2 \cdot 1) + (2 \cdot 2)$$

$$(3 \cdot 1) + (2 \cdot 2) + (2 \cdot 1)$$

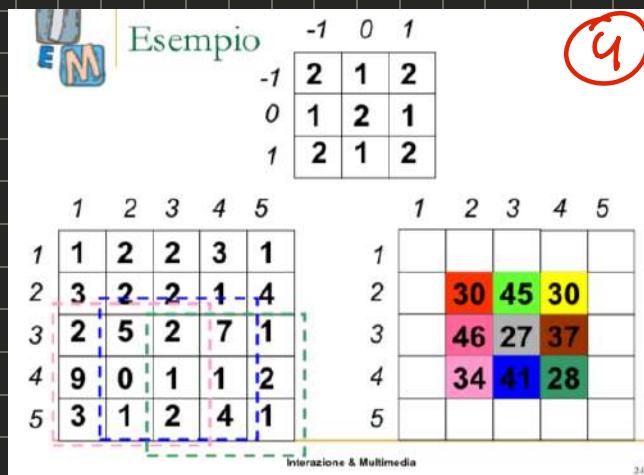
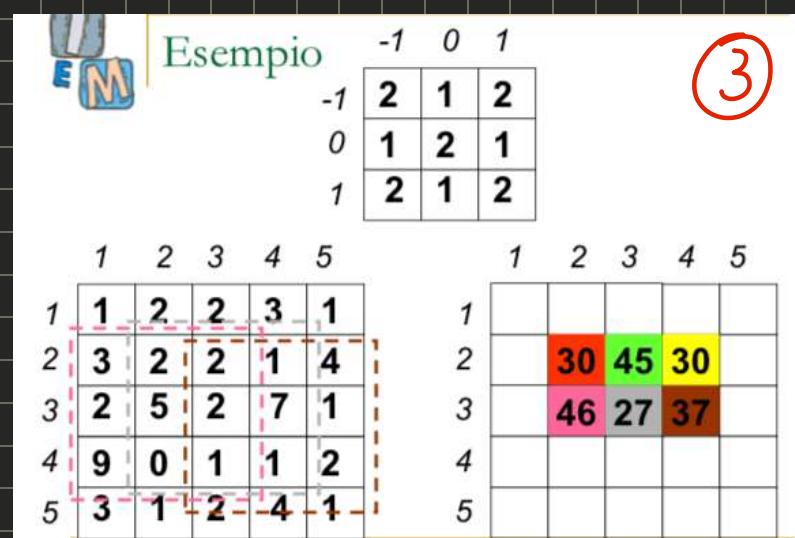
$$(2 \cdot 2) + (5 \cdot 1) + (2 \cdot 2)$$

= (30) che è il  
PIXEL CENTRALE

Facciamolo anche con gli altri quadranti...



continua...



Fino a comp.  
un'altra  
MATRICE 3x3

# FILTRAGGIO

Def

Applicare un **filtro lineare** e **shift invariante** ad una immagine è equivalente a calcolare la convoluzione del kernel del filtro con l'immagine.

→ PROBLEMA!!!

Cediamoci...



a) Filtrare solo le zone centrali dell'immagine

(a)

Le aree in grigio non verranno calcolate

input

1	2	2	3	1
3	2	2	1	4
2	5	2	7	1
9	0	1	1	2
3	1	2	4	1

output

30	45	30
46	27	37
34	41	28



b) Supporre che tutto intorno all'immagine di input ci sia lo "0"

(b)

input

0	0	0	0	0	0	0
0	1	2	2	3	1	0
0	3	2	2	1	4	0
0	2	5	2	7	1	0
0	9	0	1	1	2	0
0	3	1	2	4	1	0
0	0	0	0	0	0	0

output

11	19	17	22	11
25	30	45	30	31
25	46	27	37	19
35	34	41	28	29
16	27	12	18	10

come fare per i bordi?

POSSIBILI SOLUZIONI:

- a) Filtrare solo le zone centrali dell'immagine
- b) Supporre che tutto intorno all'immagine ci sia 0 (alcune volte a senso altre no)
- c) - Assumere una topologia "toroidale": quando si "sfiora a destra" si rientra a sinistra, quando si "sfiora" in basso di rientra in alto e viceversa;
- d) - Aggiungere una riga all'inizio uguale alle riga precedente, una riga alla fine uguale all'ultima riga, una colonna all'inizio uguale alla colonna iniziale, e una colonna alla fin



c) Riempire le righe e le colonne aggiunte in maniera "toroidale"

(c)

input

1	3	1	2	4	1	3
1	1	2	2	3	1	1
4	3	2	2	1	4	3
1	2	5	2	7	1	2
2	9	0	1	1	2	9
1	3	1	2	4	1	3
1	1	2	2	3	1	1

output

27	30	29	32	33
33	30	45	30	40
38	46	27	37	45
41	34	41	28	48
28	35	24	27	40



d) Riempire le righe e le colonne aggiunte con i valori più vicini

(d)

input

1	1	2	2	3	1	1
1	1	2	2	3	1	1
3	3	2	2	1	4	4
2	2	5	2	7	1	1
9	9	0	1	1	2	2
3	3	1	2	4	1	1
3	3	1	2	4	1	1

output

25	27	29	31	33
34	30	45	30	39
51	46	27	37	32
54	34	41	28	35
48	32	24	34	26

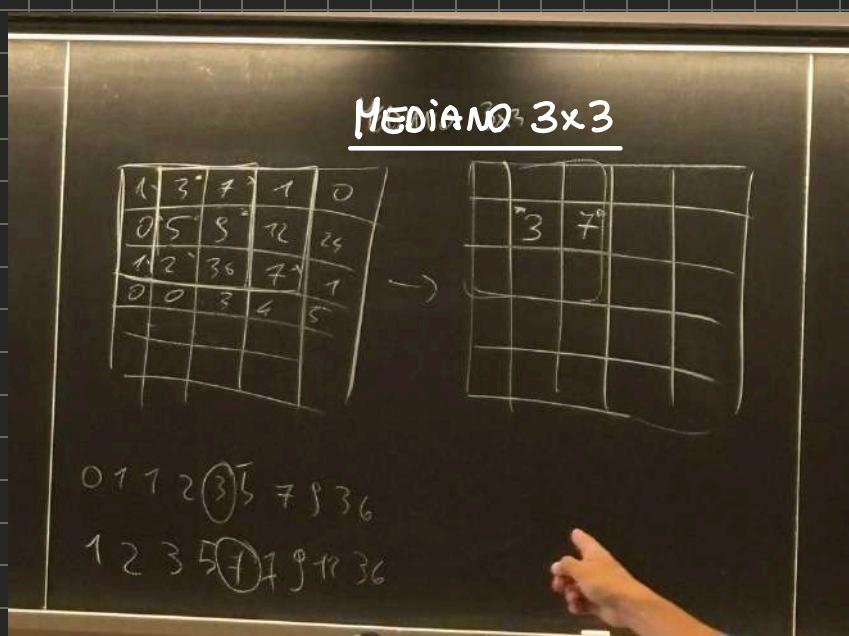
# Esempi di operatori locali

## - MEDIANO

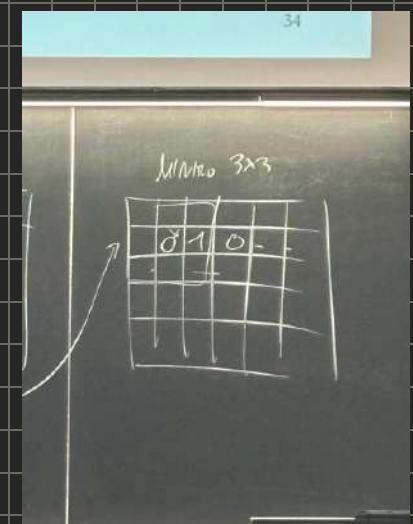
IL VALORE che STA AL CENTRO dopo l'ORDINE

7	10	12
6	38	11
9	11	6

6 6 7 9 10 11 11 12 38  
↑  
valore mediano



34



## PROPRIETÀ

- RIDOTTA QUAL. DETTAGLI  
(ALTE FREQUENZE ATTENUATE)
- LOCALMENTE ATTENUA LE DIFFERENZE (MENO NITIDEZZA)
- NON può ESSERE APPLICATO PER CONVOLUZIONE
- MINIMO e MASSIMO

Il **filtro di minimo** preso un intorno  $m \times m$  di un pixel (con  $m$  generalmente dispari), sostituiscono il valore del pixel con il **valore minimo di tutti i valori osservati in tale intorno**.

Se si sostituisce con il minimo si ottiene un **incupimento dell'immagine** (si eliminano per esempio macchie chiare);



Il **filtro di massimo** preso un intorno  $m \times m$  di un pixel (con  $m$  generalmente dispari), sostituiscono il valore del pixel con il **valore massimo di tutti i valori osservati in tale intorno**.

Se si sostituisce con il massimo si ottiene uno **schiarendo dell'immagine** (si eliminano per esempio punti neri).



N.B.: Non sono lineari, nessun kernel

# N- Box ( MEDIA )

Sono definiti da kernel  $N \times N$  con ogni elemento pari a  $1/N^2$

Si sceglie generalmente un valore  $N$  **dispari**.

Hanno l'effetto di **sfocare le immagini**: la sfocatura è **moltò forte in orizzontale e verticale** ma meno in diagonale.

ESEMPI:

3-box

1	1	1
1	1	1
1	1	1

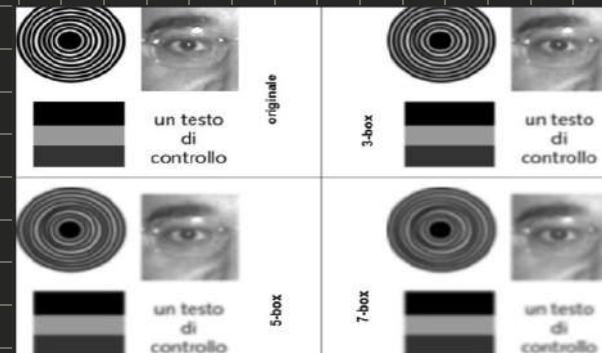
$1/9 *$

5-box

$1/25 *$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Lo uso per ATTENUARE IL RUMORE e si può APPLICARE CON UNA CONVOLUZIONE



# N- BINOMIALE

USATO per SMOOTHARE il KERNEL DERIVATI ed è IN POCHE PAROLE UNA

## MEDIA PONDERATA

= Hanno il pregio di **smussare egualmente in tutte le direzioni**.  
Smussano meno vigorosamente degli n-box.



I VALORI CENTRALI S' OTTENGONO TRAMITE  
IL PRODOTTO degli ESTERNI



3-binomiale

1	2	1
2	4	2
1	2	1

$1/16 *$

5-binomiale

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

$1/256 *$

## CONSERVAZIONE dell'ENERGIA

un filtro “conserva l'energia” se la somma dei suoi pesi fa 1.

Tutti i filtri di smoothing visti prima sono “**energy preserving**”

## NOISE CLEANING e SMOOTHING

Servono anche a ridurre il rumore in una immagine. In questo caso, più è grande il kernel e migliore sarà il risultato anche se si rischia di aumentare la sfocatura.

I filtri N-box e N-binomiali sono anche usati per **sfocare l'immagine (smoothing)**. In questo caso, più è grande il kernel e maggiore sarà la sfocatura ma si riduce meglio il rumore.

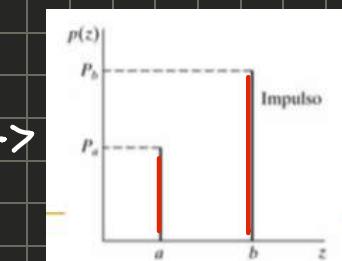
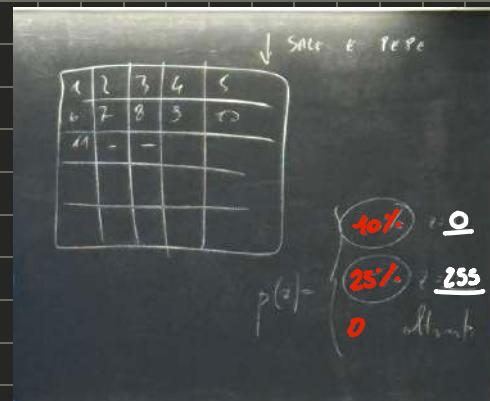
## IL RUMORE

- **RUMORE IMPULSIVO :**

detto anche “**sale e pepe**”. (valori alti o bassi)  
Viene caratterizzato dalla frazione dell'immagine modificata (in %);

$$p(z) = \begin{cases} p_a & \text{per } z=a \\ p_b & \text{per } z=b \\ 0 & \text{altrimenti} \end{cases}$$

Secondo questa legge vi è



Come RIMUOVERLO?

Tramite il **FILTRIO MEDIANO o di MEDIA**

Sia il filtro di media che il mediano hanno dimensioni 3x3

Per questo tipo di rumore il filtro mediano funziona meglio.

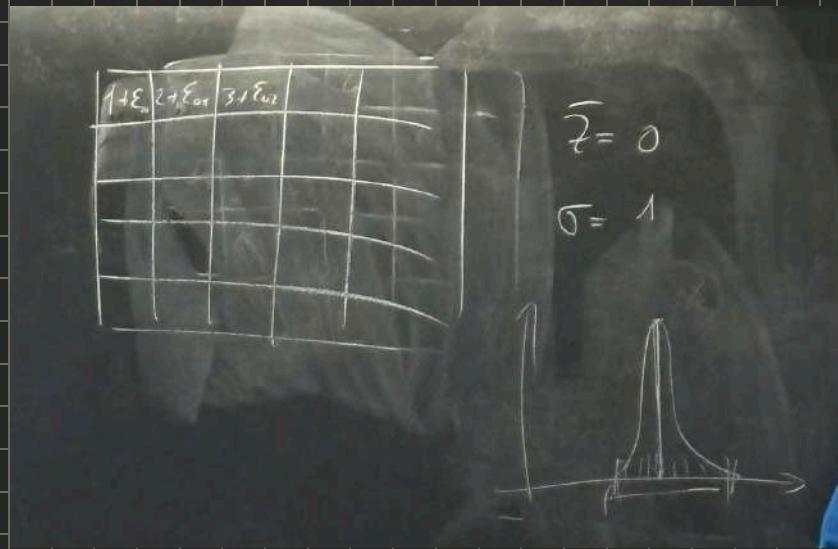


## - RUMORE GAUSSIANO

Rumore **gaussiano bianco**.

Viene caratterizzato dalla media e dalla varianza.

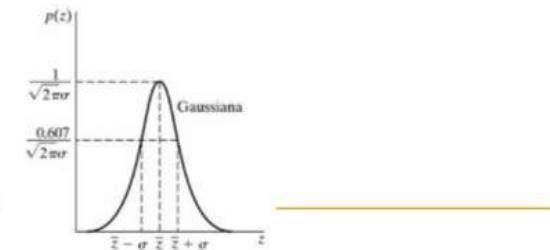
**es.**



$$P(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

(

dove  $z$  rappresenta l'intensità,  $\bar{z}$  è il valore medio<sup>1</sup> (media) di  $z$  e  $\sigma$  è la sua deviazione standard. La deviazione standard al quadrato  $\sigma^2$  è detta *varianza* di  $z$ . Il grafico di questa funzione è mostrato nella Figura 5.2a. Quando  $z$  viene descritta dall'Equazione (5.2-1), circa il 70% dei suoi valori ricade nell'intervallo  $[(\bar{z}-\sigma), (\bar{z}+\sigma)]$  e circa il 95% ricade nell'intervallo  $[(\bar{z}-2\sigma), (\bar{z}+2\sigma)]$ .



**Come rimuoverlo?**

Esattam. come nel prec. mo  
funzionava meglio il FILTRONE di MEDIA



**DIMENSIONE del Kernel**



= In entrambi i casi si elimina il rumore, ma il primo approccio sfoca di meno l'immagine finale.

**ALTRI FILTRI:**

**Outlier:** il valore del pixel centrale viene confrontato con il valore della media dei suoi 8 vicini. Se il valore assoluto della differenza è maggiore di una certa soglia, allora il punto viene sostituito dal valore medio, altrimenti non viene modificato.

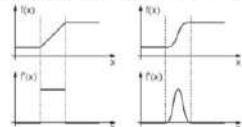
**Olimpico:** da un dato intorno si scartano i valori massimo e minimo e sul resto si fa la media.

# ESTRAZIONE dei CONTORNI

I contorni sono definiti come delle discontinuità locali della luminanza.

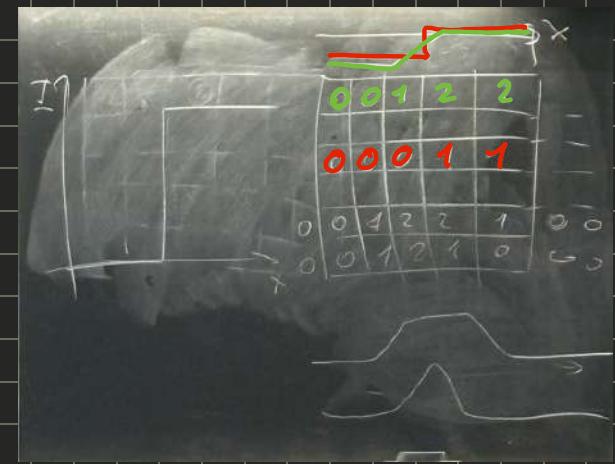
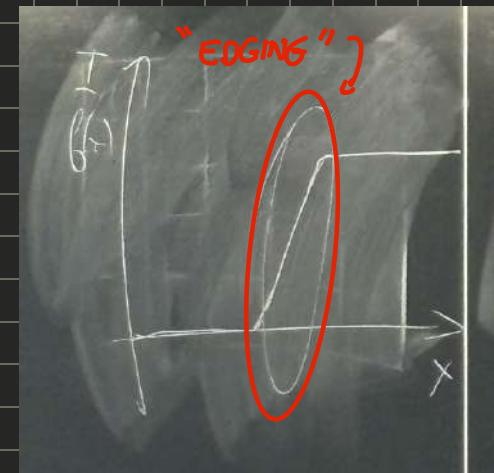
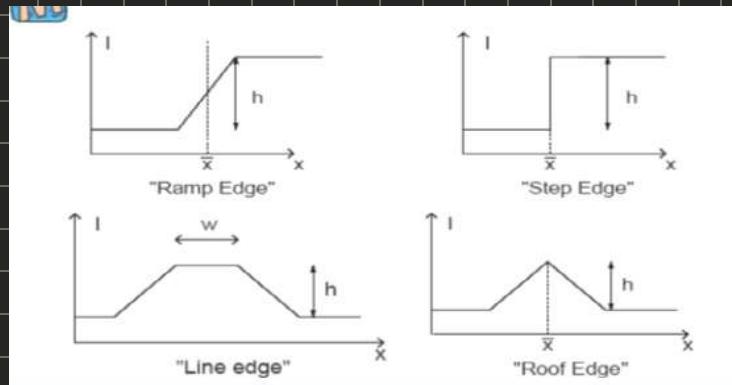
Gli edge detector forniscono immagini in cui sono preservate le variazioni di luminanza ed eliminate tutte le altre informazioni.

- Se ho un segnale monodimensionale e calcolo la derivata prima, scopro che i lati sono i corrispondenti dei massimi della derivata.



- Quindi i filtri devono calcolare la derivata in direzione x quella in direzione y e poi combinarle insieme.

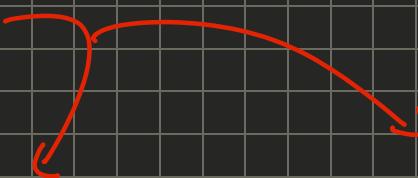
## ES. 1 DIMENSIONE



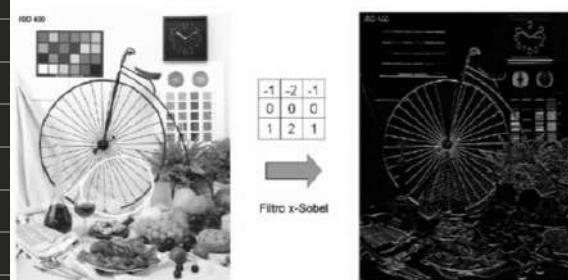
## KERNEL NOTEVOLI

$$Sobel_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad Prewitt_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

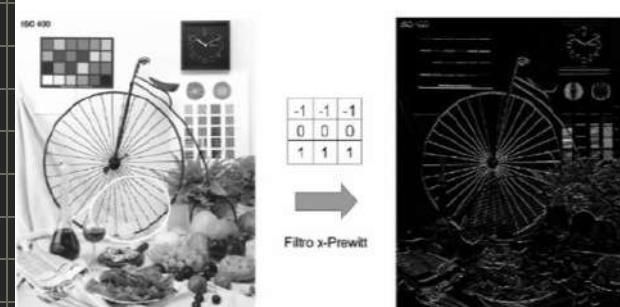
(ORIZZONTALE x)



ES SOBELx



ES Prewittx



CON SOBELx, IL LATO obliquo  
emerge PARTIALMENTE

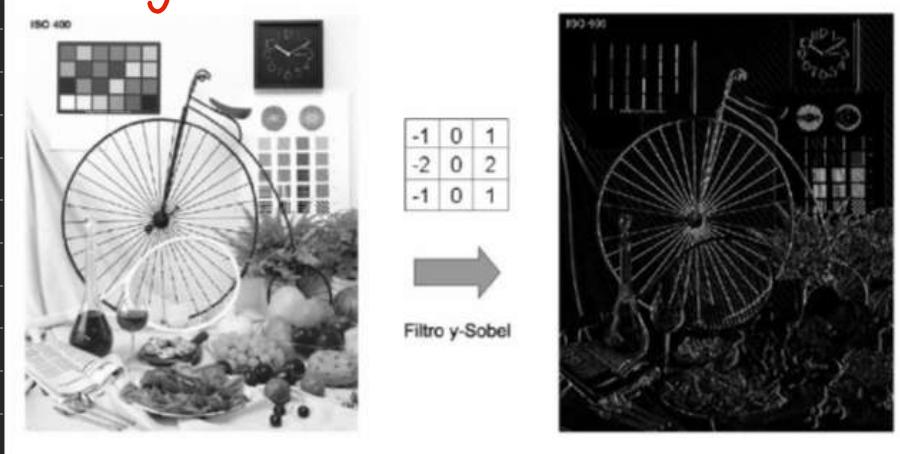


$$Sobel_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

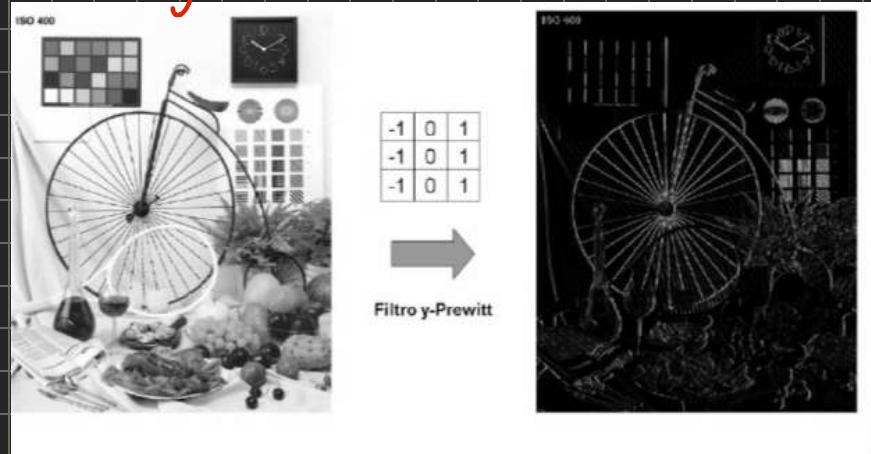
$$Prewitt_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

( VERTICALE y )

Sobel y



Prewitt y

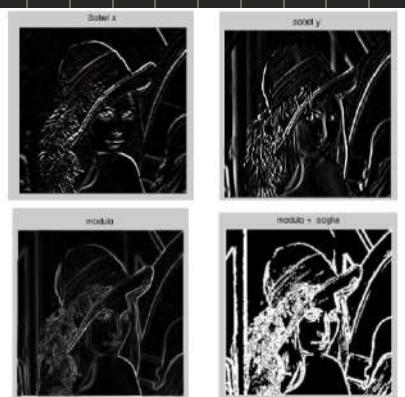


## MAGNITUDO

Sobel x fornisce una matrice con i lati orizzontali (e le componenti orizzontali dei lati obliqui) che hanno valori non nulli.

Sobel y fornisce una matrice con i lati verticali (e le componenti verticali dei lati obliqui) che hanno valori non nulli.

→ magnitudo =  $\sqrt{Sobel_x^2 + Sobel_y^2}$



la matrice ottenuta ha valori non nulli per i pixel «di lato».

Se si fissa una soglia adeguata, si può ottenere una matrice binaria che per ogni pixel ci dica se è o non è di lato:

Migliori risultati...

Si ottengono con algoritmi più sofisticati (non linear), per il calcolo della grandezza del gradiente (somma del quadrato della risposta di un edge finder orizzontale e del quadrato della risposta di un edge finder verticale).

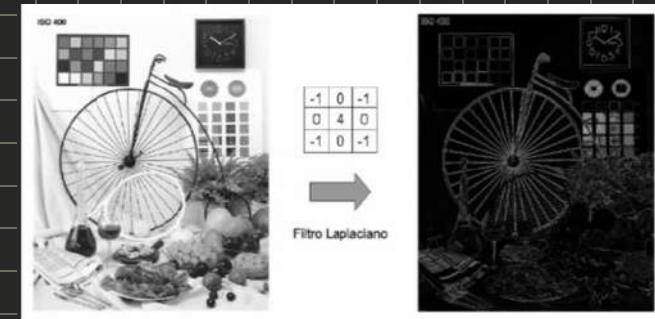
Si ottengono con strategie più "intelligenti" (algoritmo di Canny, algoritmi fuzzy, tecniche di backtracking eccetera)

## Kernel notevoli : Laplaciano

Il filtro più diffuso per calcolare la derivata seconda è detto Laplaciano, ed è definito dalla maschera:

$$\text{Laplaciano} = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

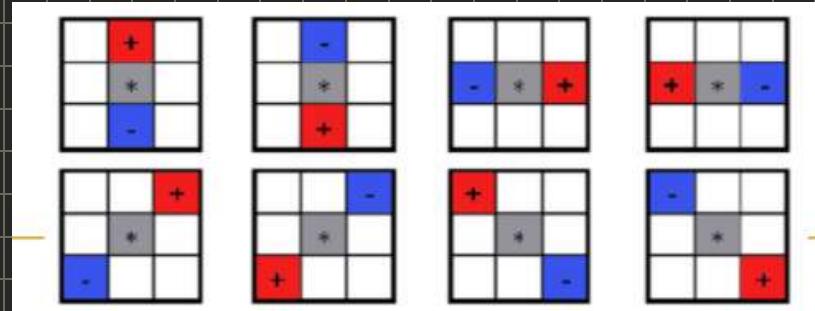
( APPROXIMAZIONE della DERIVATA SECONDA tramite FILTRO )



SENZA controllo pass. per 0  
( zero-crossing )

## ZERO - CROSSING

Oggi Deve sempre accadere, quando si applica Laplaciano, che rispetto al punto in questione ci sia nel suo intorno un valore positivo e un valore negativo.



CON controllo pass. per lo 0  
( zero-crossing )



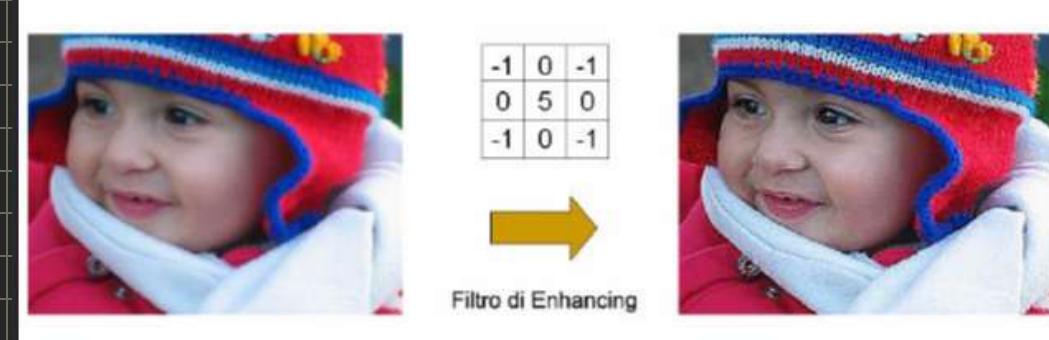
## FILTRI di SHARPENING

Def  
Sono filtri il cui scopo è quello di incrementare la nitidezza di una immagine — 7 aumentando il contrasto locale.

( STIAMO prend. un BLURRING e  
INCREMENTARLO ULTERIORM. )

Per ottenere tale effetto si può adottare una maschera che, derivata dal Laplaciano, "rinforza" i lati presenti nell'immagine.

Purtroppo essa rinforza anche il rumore !!!  
presente nella immagine!

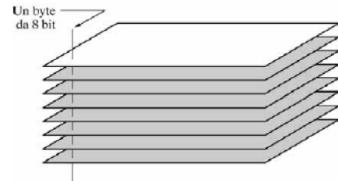


FINO A QUI X MINEDE 1  
 $77 + 44 + 32 + 13 + 23 + 44 + 147 + 71 + 89 = 507$  2 RIASSUNTE IN

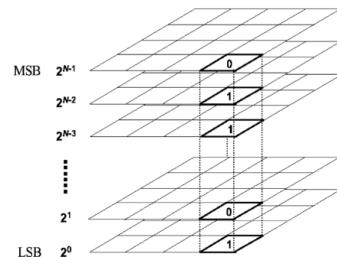
# Bit-plane (lez. 11)

Definizione:

- Un'immagine digitale a N bit può essere suddivisa in N piani di bit (*bit-planes*), ciascuno dei quali rappresenta un'immagine binaria.



- I piani vanno dal *Most Significant Bit* (MSB) al *Least Significant Bit* (LSB).

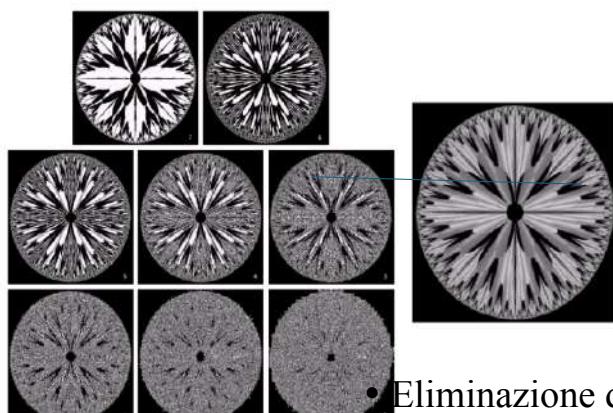


## Codifica Binaria Pura:

- I piani di bit più significativi contengono la struttura dell'immagine.
- I piani meno significativi includono dettagli e rumore.



- Piani significativi dal punto di vista visivo: dal 7 al 3.



## Usi dei Bit-Planes:

- Eliminazione di intervalli di grigi specifici.

- Modifica di singoli piani per manipolare l'immagine.



(a) Immagine ricostruita ci i piani di bit da 7 e 6. L'immagine assume solamente i valori 0, 64, 128, 192; (b) usando i piani di bit 7, 6, 5. L'immagine assume solamente, oltre lo zero, alcuni valori nel range 32 - 224; (c) i piani di bit 7, 6, 5, 4. L'immagine assume, oltre lo zero, alcuni valori nel range 16 – 240.

### Problemi del Binario Puro:

- Piccole variazioni di intensità possono causare cambiamenti in tutti i piani. (SVANTAGGIO)
- Es.: Se un pixel ha ad esempio intensità 127 (01111111) e il suo adiacente ha intensità 128 (10000000) allora la transizione tra 0 e 1 si ripercuote su tutti i piani di bit (in binario puro).

### Soluzione: Codice Gray

- Codifica alternativa che minimizza le variazioni tra pixel consecutivi.

Utilizziamo la seguente formula:

$$\begin{aligned} g_i &= a_i \oplus a_{i+1} \quad 0 \leq i \leq m-2 \\ g_{m-1} &= a_{m-1} \end{aligned}$$

dove  $\oplus$  denota l'operatore XOR (OR esclusivo).

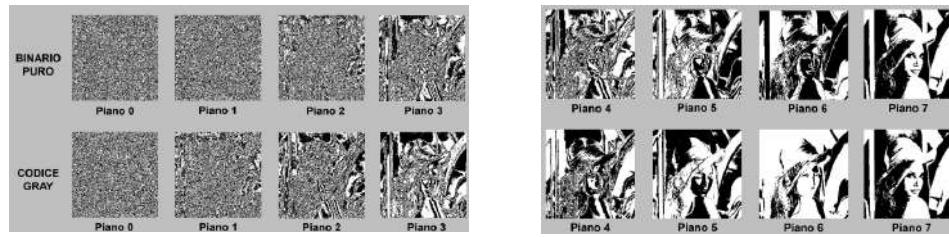
$m$  = numero di bit del codice Gray

- Proprietà: ogni *codeword* differisce dalla precedente per un solo bit (distanza di Hamming = 1).
- Vantaggi:
  - Maggiore coerenza tra i piani.
  - Minor entropia (più ridondanza), utile per la compressione.
  - Riduzione delle transizioni bianco-nero nei piani singoli.

### Confronto Binario Puro vs. Codice Gray

- **Binario Puro:**

- Dettagli e rumore concentrati nei piani bassi
- Cambiamenti drastici tra pixel adiacenti.



### • Gray Code:

- Cambiamenti graduali (un solo bit varia).
- Maggior semplicità nella compressione, ma proprietà diverse rispetto al binario puro.

#### Esempio con i numeri a 3 bit

Numero in decimale	Numero in binario puro	Numero in gray code
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Si osservi il caso con maggiore variazione in binario, quello relativo ai numeri decimali 3 e 4.

In questo caso a fronte di una grande variazione di bit nel binario puro si osserva la variazione di un solo bit nel gray code.

Inoltre, nel gray code ogni numero con il suo successivo differisce solo per la variazione di un bit

### ATTENZIONE!!!

#### • Significato dei bit diverso tra codifiche:

- Il significato associato ai bit nei *bit-planes* varia tra binario puro e codice Gray.
- Le proprietà di una codifica non sempre valgono per l'altra.

#### • Azzeramento di piani di bit:

- In codice Gray, l'eliminazione di un piano di bit rimuove intervalli di valori diversi rispetto al binario puro.
- I valori eliminati tendono a essere meno significativi in Gray.

#### • Concentrazione di dettagli e rumore:

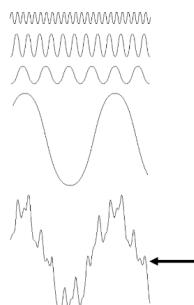
- Anche nel codice Gray, i dettagli e il rumore si concentrano nei piani bassi.
- Tuttavia, eliminare direttamente questi piani può causare artefatti indesiderati.

## LEZ 12 – Trasformata di Fourier

**Image Enhancement nel Dominio delle Frequenze** → miglioramento delle immagini nel dominio delle frequenze.

### Introduzione

- La **Serie di Fourier** esprime funzioni periodiche come somme di seni e coseni.
- La **Trasformata di Fourier** si applica a funzioni non periodiche.
- Cenno storico: Jean Baptiste Joseph Fourier.

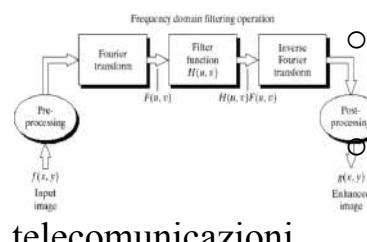


### Un primo esempio

Rappresentazione grafica: una funzione data come somma di 4 funzioni periodiche.

### Introduzione

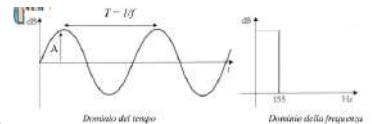
- Proprietà:
  - Sia la Serie di Fourier che la Trasformata permettono una ricostruzione senza perdita di informazione.
- Applicazioni iniziali nella diffusione del calore.
- Contributi della **Fast Fourier Transform (FFT)**:



- Rivoluzione nell'elaborazione digitale del segnale (DSP).
- Applicazioni industriali, mediche, e nelle telecomunicazioni.

## Immagini e segnali

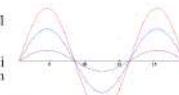
- Le immagini possono essere viste come funzioni discrete bidimensionali.
- Concetto di segnale: funzioni variabili con frequenza associata.



## Caratteristiche delle frequenze



- Ampiezza (A) espressa in decibel dB;
- Periodo (T) espresso in secondi;
- Frequenza ( $f$ ) numero di cicli (onde) al secondo; si misura in Hertz Hz



Rappresentare un'immagine su uno spazio di funzioni sinusoidali:



### Discrete Fourier Transform

Nel caso 2-D la coppia trasformata antitrasformata della sequenza bidimensionale  $f(x,y)$  assume la seguente forma:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad \text{per } u=0,1,\dots,M-1 \quad v=0,1,\dots,N-1$$

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{i2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad \text{per } x=0,1,\dots,M-1 \quad y=0,1,\dots,N-1$$

$u$  e  $v$  sono gli indici relativi agli assi frequenze discretizzati, mentre  $M$  e  $N$  sono le dimensioni (in pixel) dell'immagine.

$$\begin{bmatrix} 10 & 90 \\ 30 & 40 \end{bmatrix} = \underbrace{\begin{bmatrix} 10 \\ 30 \end{bmatrix}}_{\downarrow} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \underbrace{90}_{\downarrow} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \dots$$
  

$$\begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \underbrace{\begin{bmatrix} c_0 \\ c_1 \end{bmatrix}}_{\circlearrowleft} \begin{bmatrix} a & b \\ c & d \end{bmatrix} + c_1' \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Serve per elaborare oggetti  $c_1 \neq c_1'$

ES:  
TRASF FOURIER

2+3i	-1
i 9	3



## Trasformata di Fourier

Dato che la trasformata  $F$  ha valori complessi, può essere espressa in termini della sua parte reale e della sua parte immaginaria.

$$\text{Spettro della Trasformata} \quad |F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

$$\text{Angolo di Fase} \quad \phi(u,v) = \tan^{-1} \left[ \frac{I(u,v)}{R(u,v)} \right]$$

$$\text{Potenza Spettrale} \quad P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

**Spettro della Trasformata:**

modulo del numero complesso

$$\begin{array}{|c|c|} \hline 2+i3 & -1 \\ \hline i4 & 3 \\ \hline \end{array} \xrightarrow{|F(u,v)|} \begin{array}{|c|c|} \hline \sqrt{13} & -1 \\ \hline i4 & 3 \\ \hline \end{array}$$

$$\sqrt{2^2+3^2} = \sqrt{13}$$

**Angolo di Fase**

$$\begin{array}{|c|c|} \hline 2+i3 & -1 \\ \hline i4 & 3 \\ \hline \end{array} \xrightarrow{\phi(u,v)} \begin{array}{|c|c|} \hline \tan^{-1} \frac{3}{2} & \tan^{-1} \frac{0}{-1} \\ \hline \frac{\pi}{2} & \\ \hline \end{array}$$

$\lim_{x \rightarrow \infty} \tan^{-1} x = \frac{\pi}{2}$

**Potenza spettrale:**

Parte reale al quadrato + Parte immaginaria al quadrato

**Esempio 1D (importante dettaglio)**

- Ogni valore della funzione contribuisce a ogni campione della trasformata e viceversa.



### Esempio 1-D

$$F(0) = \sum_{x=0}^3 f(x) = [f(0) + f(1) + f(2) + f(3)] \\ = 1 + 2 + 4 + 4 = 11$$

$$F(1) = \sum_{x=0}^3 f(x) e^{-j2\pi(1)x/4} \\ = 1e^0 + 2e^{-j\pi/2} + 4e^{-j\pi} + 4e^{-j3\pi/2} = -3 + 2j$$

$$F(2) = -(1 + 0j) \text{ ed } F(3) = -(3 + 2j)$$

Se invece prendiamo  $F(u)$  e vogliamo calcolarne la sua inversa, dobbiamo procedere alla stessa maniera, stavolta utilizzando l'antitrasformata. Per esempio,

$$f(0) = \frac{1}{4} \sum_{u=0}^3 F(u) e^{j2\pi u(0)} \\ = \frac{1}{4} \sum_{u=0}^3 F(u) \\ = \frac{1}{4} [11 - 3 + 2j - 1 - 3 - 2j] = \frac{1}{4} [4] = 1$$

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-j\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)}$$

$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

$$\stackrel{-j2\pi(0+0)}{\rightarrow} \stackrel{0}{\rightarrow} \stackrel{1}{\rightarrow}$$

quindi la trans di  $F(0,0)$  è la media

### Range dinamico

- Problemi di visualizzazione dello spettro di Fourier:
  - Grande dinamica dei valori
  - Soluzione: compressione logaritmica per rappresentazioni visibili.

**Range dinamico**

Poiché  $0 < |F(u,v)| < R = 6.47 \times 10^6$ , si ha  $0 < D(u,v) < \log(1+R)$ . Dato che  $R \gg 1$ , come peraltro avviene normalmente per lo spettro di Fourier di una immagine, si può porre  $c \log R = L-1$ , da cui  $c = (L-1)/\log R = 255/\log(6.47 \times 10^6) = 16.26$ . Pertanto  $D(u,v)$  ha tutti i valori nell'intervallo [0, 255], e ciò consente la visualizzazione di molti più dettagli.

$f(x,y)$        $|F(u,v)|$        $D(u,v)$

parti + chiare formano punti di luce bianchi

## Visualizzazione dell'immagine "Lena"

- **Modulo**
- **logaritmo del modulo**
- **fase rappresentati graficamente**

## Esempi di DFT

- Esempi di rappresentazioni:
  - Trasformata discreta 1D e 2D.

**Esempio**

Un esempio di trasformata discreta nel caso 2D: un **impulso** sponzorizzato da un piccolo cerchio bianco su fondo nero, in un'immagine di  $256 \times 256$  pixel. I differenti toni di grigio nell'immagine di input della spettro evidenziano le ampiezze decrescenti dei diversi lobi.

$f(x,y)$        $F(u,v)$

**Esempio**

$f(x,y)$        $|F(u,v)|$        $\Phi(u,v)$

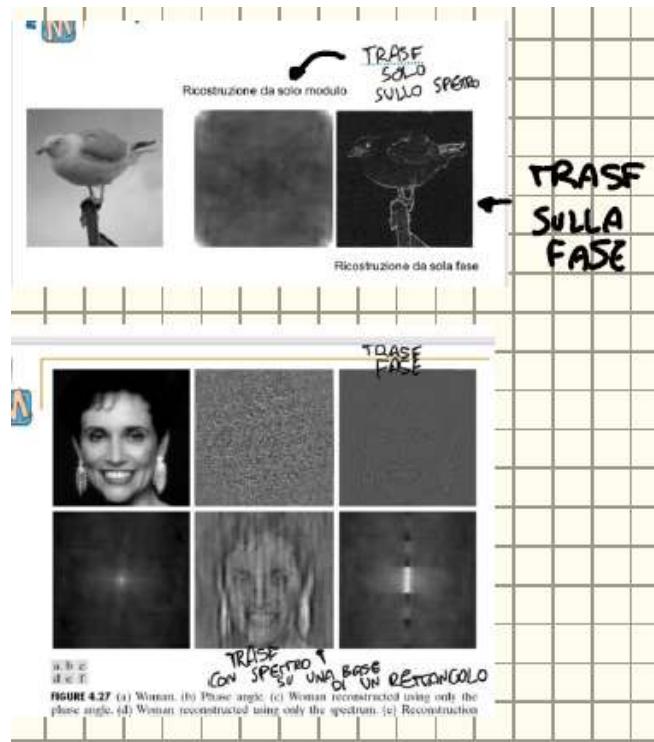
La visualizzazione dello spettro riguarda in realtà non  $|F(u,v)|$  ma una sua versione compresa logaritmicamente. Altrimenti si vedrebbe solo un punto al centro.

L'ampiezza contiene l'informazione relativa al fatto che una certa struttura periodica è presente nell'immagine.

La fase contiene l'informazione relativa ai dove le strutture periodiche esistenti nella DFT sono collocate. Quindi è molto più significativa di quella che possa sembrare nell'immagine.

Lo spettro vede le strutture geometriche presenti  $|F(u,v)|$   
La fase dove le strutture vengono collocate  $\Phi(u,v)$

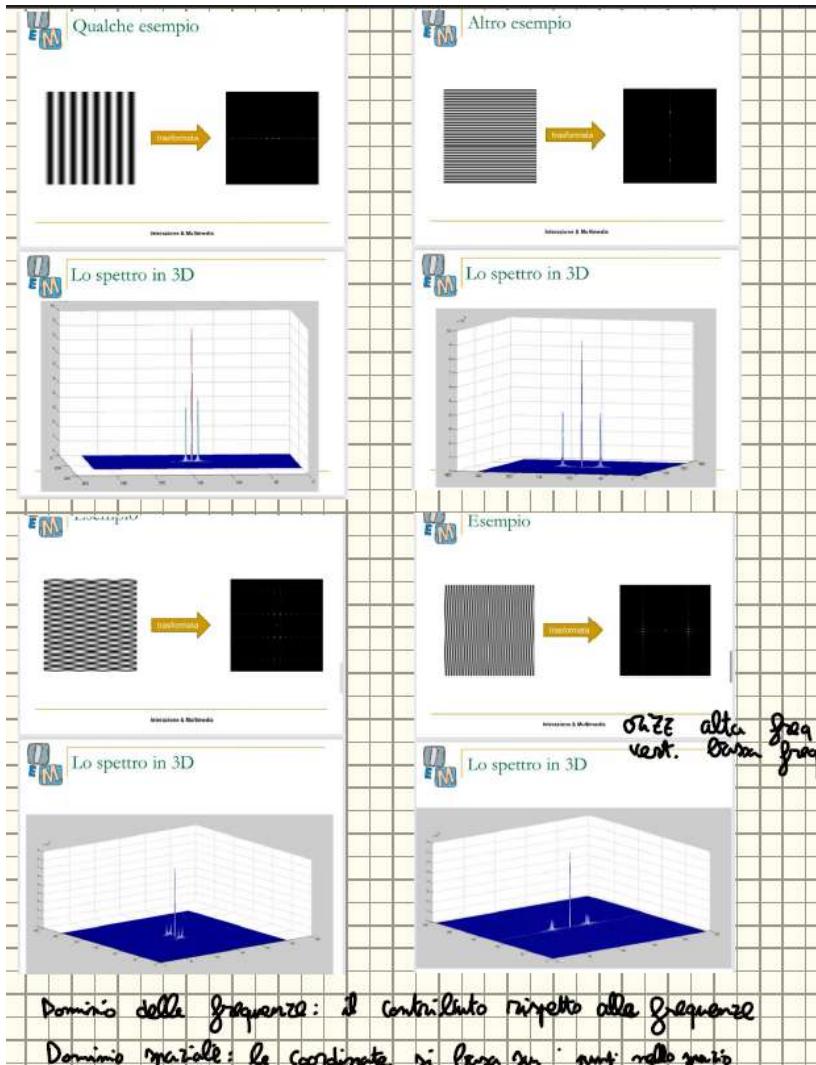
- Importanza della fase nella ricostruzione.



## Modulo in 3D

- Visualizzazione tridimensionale dello spettro

## Esempi grafici



*Domino delle frequenze: il contributo rispetto alle frequenze*

*Domino spaziale: le coordinate si basano sui punti nello spazio*

## Vantaggi della Trasformata di Fourier

- sopprimere frequenze indesiderate
- ridurre lo spazio occupato dai dati pur limitando la degenerazione del segnale (JPEG, MPEG, DivX, MP3)
- rigenerare segnali degradati

## Tipi di trasformata

- La trasformata discreta di Walsh (DWT)
- La trasformata discreta di Hadamard (DHT)
- La trasformata discreta del Coseno (DCT)
- La trasformata discreta di Karhunen Loeve (KLT)

## Proprietà della DFT 2D

- Separabilità
- Traslazione
- Valore Medio

**Separabilità:** se devo trasformare in Fourier in 2 dimensioni posso separarli in 2 operazioni in 1 dimensione

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} k(x, v) e^{-j \frac{2\pi u x}{N}}$$

$$k(x, v) = \left| \sum_{y=0}^{N-1} f(x, y) e^{-j \frac{2\pi v y}{N}} \right|$$

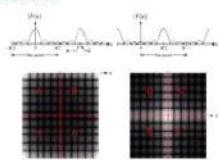
Il principale vantaggio delle proprietà di separabilità è che la  $F(u, v)$  può essere ottenuta applicando in due passi successivi la trasformata 1-D.

**Traslazione:** Separare in quadranti.

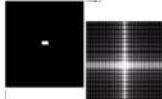
Ha l'obiettivo di portare

la trasformata  $(0,0)$  al centro

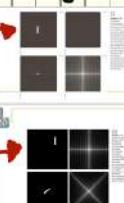
### Traslazione



La trasformata di un piccolo rettangolo bianco su uno sfondo nero sarà dunque (dopo lo shift)



Se le immagini hanno solo una traslazione allora la trasformata è uguale



sono uguali anche se i rettangoli sono disposti diversamente

### Valore medio:

Il valore della trasformata nell'origine, cioè nel punto  $(u, v) = (0, 0)$  è dato da:

$$F(0,0) = \frac{1}{N \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \quad \tilde{f}(x, y) = \frac{1}{N \times N} F(0,0)$$

Come si può vedere non è altro che la media di  $f(x, y)$ . Il valore della trasformata di Fourier di un'immagine  $f(x)$  nell'origine è uguale alla media dei valori di grigio contenuti nell'immagine.

$F(0,0)$  prende anche il nome di **componente continua** o componente DC.

### FFT:

#### Fast Fourier Transform

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-j 2\pi u x / N]$$

- Nella sua forma classica implementare la trasformata di Fourier richiederebbe un numero di operazioni proporzionale a  $N^2$  ( $N$  moltiplicazioni complesse e  $N-1$  addizioni per ciascuno degli  $N$  valori di  $u$ ).
- Utilizzando opportune tecniche di decomposizione è possibile abbassare la complessità a  $N \log N$ , implementando la cosiddetta Fast Fourier Transform (FFT).

Ottieniamo tutte le trasformate in Complessità  $N \log N$  ottenendo la matrice dei coefficienti

**Convoluzione:** Scorrimento di matrici con somma di prodotti.

### Frequenze: Low and High

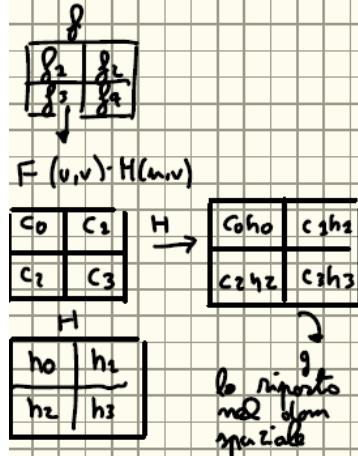
- Basse frequenze: zone uniformi.
- Alte frequenze: bordi e rumore.

### Filtraggio nel dominio della frequenza

#### Tipologie:

- **Low-pass** (ideale, Butterworth, Gaussiano): rimuove alte frequenze.

- **High-pass:** elimina le basse frequenze.
- **Band-reject:** elimina intervalli specifici.



### Teorema della convoluzione

La trasformata della convoluzione di due segnali nel dominio spaziale equivale al prodotto delle trasformate dei due segnali.

Che significa anche:

La convoluzione di due segnali nel dominio spaziale equivale all'antitrasformata del prodotto delle trasformate dei due segnali.

Il fondamento teorico delle tecniche di elaborazione nel dominio della Frequenza, basate sulla manipolazione della DFT dell'immagine, è rappresentato dal **teorema della convoluzione** che fa corrispondere, alla operazione così definita nel dominio spaziale:

$$g(x,y) = f(x,y) \otimes h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m, y-n)$$

L'operazione, nel dominio delle frequenze:

$$\begin{aligned} G(u,v) &= F(u,v)H(u,v) \\ \text{e quindi} \quad g(x,y) &= F^{-1}[F(u,v)H(u,v)] \end{aligned}$$

■ Complessità per un segnale 1D:

Nel dominio delle frequenze  $O(n \log n)$

Nel dominio spaziale  $O(n^2)$

Effettivamente vale la pena passare al dominio delle frequenze!

$f$  immagine

$\mathcal{F}$

$h$  Kernel

$$\mathcal{F}[f \otimes h] = \mathcal{F}[f] \cdot \mathcal{F}[h]$$

$g$  output (imm. filtrata)

$\mathcal{F}$

$G$  transf. di  $g$ .

$$\mathcal{F}[f \otimes h] = F \cdot H \Rightarrow g \otimes h = \mathcal{F}^{-1}(F \cdot H)$$

$F$  transf. di  $f$

$$G = F \cdot H$$

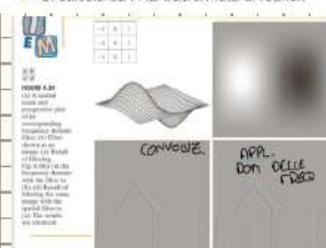
Se il filtro è più piccolo : dominio spaziale

Se il filtro è più grande : prodotto punto a punto



Come ottenere un filtro a partire da una maschera spaziale

1. Il filtro  $H$  ha la stessa dimensione dell'immagine  $I$ :
2.  $H$  deve avere in alto a sinistra i valori della maschera spaziale, nel resto sempre il valore 0;
3. Si fa lo shift di  $H$
4. Si calcola da  $H$  la trasformata di Fourier.

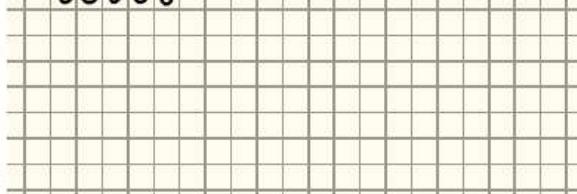


Aggiungendo degli

zeri sui bordi non

c'sono ulteriori contributi.

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$



## Filtri low pass nel dominio della frequenza

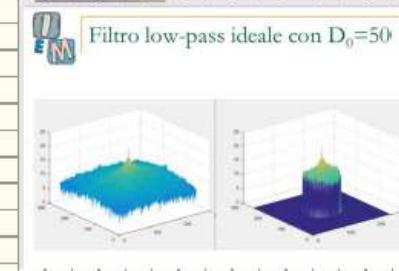
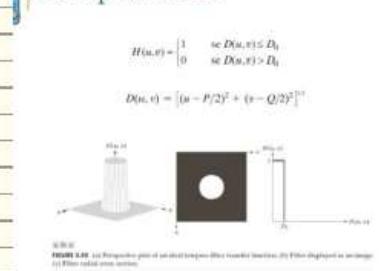
Lowpass filters.  $D_0$  is the cutoff frequency and  $\alpha$  is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = \frac{1}{1 + (D(u, v)/D_0)^\alpha}$	$H(u, v) = e^{-\pi D(u, v)/D_0}$

Ha valori: 0 o 1

metto 1 se una certa quantità di mv (Se mi trovo vicino al centro)  
 $D(u, v)$ : distanza dal centro al filtro

Low pass ideale



**Filtro di Butterworth**

**Filtro low-pass di Butterworth**

**Filtri low pass di Butterworth**

La funzione di trasferimento del filtro **passa-basso di Butterworth** di ordine  $n$  e frequenza di taglio  $D_0$  è:

$$H(u, v) = \frac{1}{1 + \left[ \frac{D(u, v)}{D_0} \right]^{2n}}$$

**Filtro gaussiano (DEVE AVERE O  $\epsilon^2$ )**

- I filtri Gaussiani sono definiti da:

$$H(u, v) = e^{-\frac{\pi^2 D^2(u, v)}{2n^2}}$$

I filtri gaussiani hanno il grande vantaggio di avere come trasformata di Fourier ancora una gaussiana.

**Filtro gaussiano**

**Filtro low-pass gaussiano**

**Vantaggio:** la transf. di Fourier è ancora una f. gaussiana con qualche piccola varianza  
Sia Kernel che filtro hanno la stessa definizione



## Filtri high pass nel dominio della frequenza

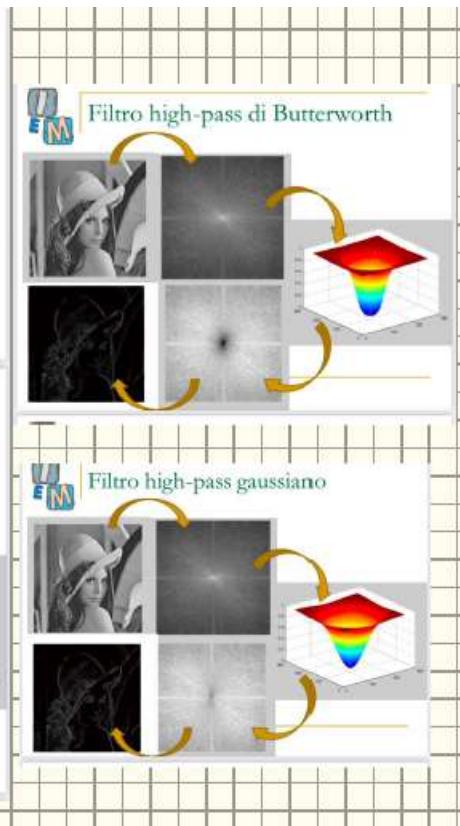
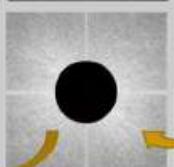
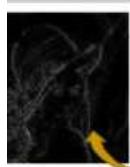
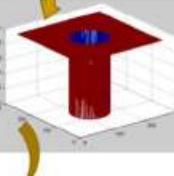
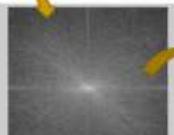
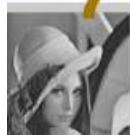
Highpass filters.  $D_0$  is the cutoff frequency and  $n$  is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$	$H(u, v) = 1 - e^{-D(u, v)/2\sigma}$

SE SIANO  
LONTANI DAL CENTRO  
SE SIANO VICINI AL CENTRO



### Filtro high-pass ideale

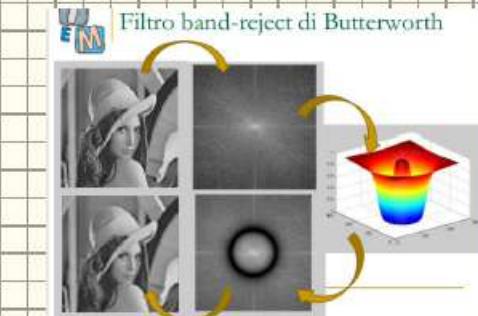
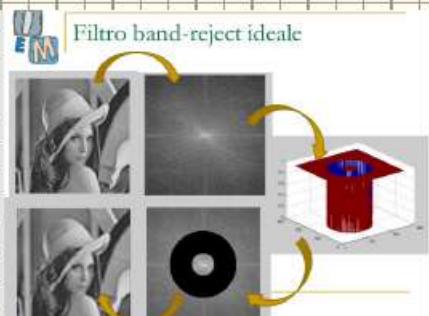




## Filtri band reject nel dominio della frequenza

Bandreject filters.  $W$  is the width of the band,  $D$  is the distance  $D(u, v)$  from the center of the filter,  $D_0$  is the cut-off frequency, and  $n$  is the order of the Butterworth filter. We show  $D$  instead of  $D(u, v)$  to simplify the notation in the table.

Ideal	LARGEZZA BANDA	Butterworth	Gaussian
$H(u, v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u, v) = \frac{1}{1 + \left[ \frac{DW}{D^2 - D_0^2} \right]^{2n}}$	$H(u, v) = 1 - e^{\frac{(D^2 - D_0^2)}{2\sigma^2}}$	



# Riassunto formule

**TABLE 4.4**  
Lowpass filters.  $D_0$  is the cutoff frequency and  $n$  is the order of the Butterworth filter.

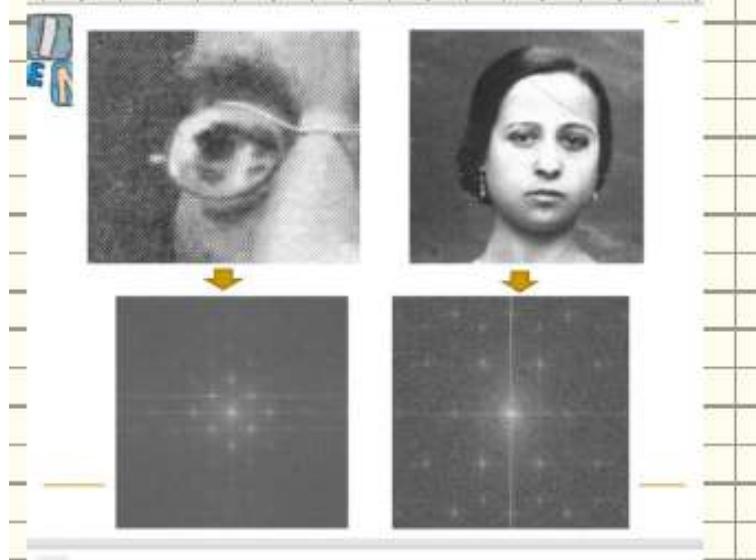
Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^n}$	$H(u, v) = e^{-D(u, v)/2D_0}$

**TABLE 4.5**  
Highpass filters.  $D_0$  is the cutoff frequency and  $n$  is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^n}$	$H(u, v) = 1 - e^{-D(u, v)/2D_0}$

**TABLE 4.6**  
Bandreject filters.  $W$  is the width of the band.  $D$  is the distance  $D(u, v)$  from the center of the filter.  $D_0$  is the cutoff frequency, and  $n$  is the order of the Butterworth filter. We show  $D$  instead of  $D(u, v)$  to simplify the notation in the table.

Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u, v) = \frac{1}{1 + \left[ \frac{D}{D_0 - D} \right]^{2n}}$	$H(u, v) = 1 - e^{-\frac{ D }{DW}}$



## LEZ 13 – Compressione

### Compressione

- Introduzione al concetto di compressione: riduzione della quantità di bit per rappresentare un'informazione digitale.

- Obiettivi della compressione:

- Ridurre le dimensioni dei file.
  - Risparmiare spazio di memorizzazione.
  - Ottimizzare l'occupazione di banda per la trasmissione.

### Ridondanza della codifica

- **Codice**: insieme di simboli (lettere, numeri, bit) per rappresentare informazioni.
  - Lunghezza = numero di simboli che costituiscono il codice
- **Codeword**: sequenza di simboli assegnata a ciascun elemento.
- **Ridondanza**: dati ripetuti o irrilevanti.

### Ridondanza spaziale e temporale

- Ridondanza spaziale: correlazione tra pixel vicini in un'immagine → replicazione inutile dell'informazione.
- Ridondanza temporale: correlazione tra frame consecutivi nei video → anch'essi rappresentano un'informazione duplicata.

### Informazione percettivamente irrilevante

- Dati ignorati dal sistema sensoriale umano.
- Considerati ridondanti poiché inutilizzati.

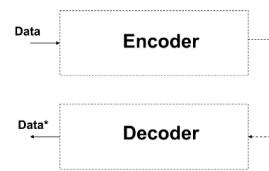
### Algoritmo di compressione

Tecnica che:

- Elimina ridondanza dai dati.
- Risparmio di memoria.

### Processo di compressione

- Compressione: dati → forma compressa.
- Decompressione: forma compressa → dati originali.



## Classificazione dei metodi di compressione

- Per tipo di dati: generici, audio, immagini, video.
- Per tipo di compressione:
  - **Lossless o Reversibile**: senza perdita di informazione.
  - **Lossy o Irreversibile**: con perdita di informazione.

## Compressione LOSSLESS

- Trasformazione reversibile: dati ricostruiti senza errore.
- Essenziale per documenti e software.

## Criterio per una buona compressione LOSSLESS

- Raggiungere il limite teorico stabilito dal **primo teorema di Shannon**.

## Frequenza

- **Frequenza di un simbolo**: numero di occorrenze in rapporto al totale.

Dati:

- S = sequenza di caratteri
- N = caratteri dell'alfabeto
- M = possibili caratteri →  $a_1, \dots, a_M$

$$f_i = (\# \text{occorrenze } a_i) / N$$

## Entropia

- **Definizione**: quantità media di informazione associata alla singola generazione di un simbolo nella sequenza.
- **Formula**:

$$E = -\sum f_i \log_2(f_i) \quad i \in S$$

- Maggiore incertezza della sequenza → maggiore entropia.

## Teorema di Shannon (1948)

### Per una sorgente discreta a memoria zero:

- Bit-rate minimo = entropia della sorgente.
- **Numero minimo di bit necessari per la rappresentazione di dati lossless**:

$$N * E$$

N = numero di caratteri

E = entropia

- **Fissa il numero minimo di bit, ma non ci dice come trovarli**
- **Soluzione** → **Algoritmo di Huffman** = Algoritmo greedy che ci permette di codificare i nostri caratteri usando il

numero di bit ricavati con il teorema di Shannon.

### Codifica di Huffman (1953)

- Algoritmo per una compressione quasi ottimale (eccede di poco il limite di Shannon).
- Codifica a lunghezza variabile:
  - Simboli frequenti → codici brevi.
  - Simboli rari → codici lunghi.
- Applicazione: esempi con frequenze e alberi binari.

### Proprietà della codifica di Huffman

- **Principio chiave:** assegnare codici binari più brevi ai simboli più frequenti e codici più lunghi a quelli meno frequenti.
  - ↓
- **Lunghezza variabile:** i simboli più frequenti ricevono codici più corti.
- **Non-prefissata:** nessun codice è prefisso di un altro, garantendo una decodifica univoca.
- **Ottimalità:** la codifica raggiunge quasi il limite di compressione teorico imposto dal **teorema di Shannon**, con un eccesso di al massimo pochi bit.

### Procedura dell'algoritmo

#### 1. Calcolo delle frequenze:

- Dati i simboli e le loro frequenze, si costruisce un albero



#### Codifica di Huffman (1)

Illustro l'algoritmo con un esempio.

Dati: AABABCAACAAADDDD.

A : frequenza pari a  $8/16 = 1/2$

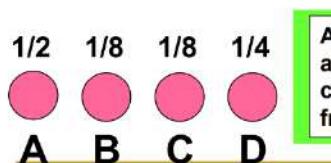
B : frequenza pari a  $2/16 = 1/8$

C: frequenza pari a  $2/16 = 1/8$

D: frequenza pari a  $4/16 = 1/4$

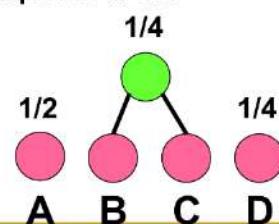
L'algoritmo procede costruendo un albero binario le cui foglie sono i caratteri da codificare come segue:

INIZIO: una foresta con 4 alberi ciascuno composto di un singolo nodo.



Aggrego i due alberi della foresta con minore frequenza

Interazione & Multimedia



binario.

## 2. Costruzione dell'albero binario:

- Ogni simbolo è rappresentato come una foglia con peso pari alla sua frequenza.
- Si aggregano iterativamente i due nodi con peso minimo, creando un nuovo nodo la cui frequenza è la somma delle frequenze dei nodi aggregati.
- Ripetere fino a ottenere un unico albero.



### Codifica di Huffman(2)

Procedo in tal modo fino ad avere un solo albero che aggreghi tutte le foglie



## 3. Assegnazione dei codici:

- Si etichettano i rami dell'albero: "1" per il ramo sinistro e "0" per quello destro.
- Ogni simbolo riceve un codice binario corrispondente al



### Codifica di Huffman(3)

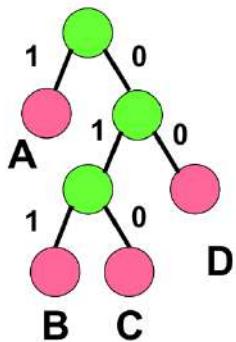
In questo caso ecco il risultato finale:



percorso dalla radice alla foglia.



## Codifica di Huffman(4)



Etichetto con 1 i rami sinistri e 0 i destri. Il cammino dalla radice al simbolo fornisce la parola del dizionario che “codifica” in maniera ottimale il simbolo.

Si osservi che ho parole di codice di varia lunghezza, ma nessuna è prefissa delle altre. Inoltre i simboli più frequenti richiedono meno bit, i meno frequenti più bit.

A : 1  
B : 011  
C : 010  
D : 00

### 4. Esempio pratico

### 5. Verifica della codifica ottimale

Limite teorico di Shannon:



## Codifica di Huffman(5)

A : 1  
B : 011  
C : 010  
D : 00

- Codice per la sequenza AABABCACAAADDDDD.
- A : frequenza pari a  $8/16 = 1/2$
- B : frequenza pari a  $2/16 = 1/8$
- C : frequenza pari a  $2/16 = 1/8$
- D : frequenza pari a  $4/16 = 1/4$

Codifica:

1-1-011-1-011-010-1-1-010-1-1-1-00-00-00-00      pari a **28 bit**

(si osservi che i trattini sono del tutto superflui perché nessun codice per i caratteri è prefisso degli altri, cioè posso decodificare senza fare errori se ho solo:

1101110110101101011100000000

Quale è il limite previsto da Shannon?

$$16 * (-1/2 * \log_2(1/2) - 1/8 * \log_2(1/8) - 1/8 * \log_2(1/8) - 1/4 * \log_2(1/4)) =$$

$$16 * (1/2 + 3/8 + 3/8 + 2/4) = 8 + 6 + 6 + 8 = **28 bit** - CODIFICA OTTIMALE!$$

Appli  
cazi  
on  
i

Uti  
lizz  
ata  
in  
sta  
nda

rd di compressione come:

- JPEG: nella fase finale per la compressione entropica.
- MPEG, JBIG2 e formati come ZIP.

### Limiti

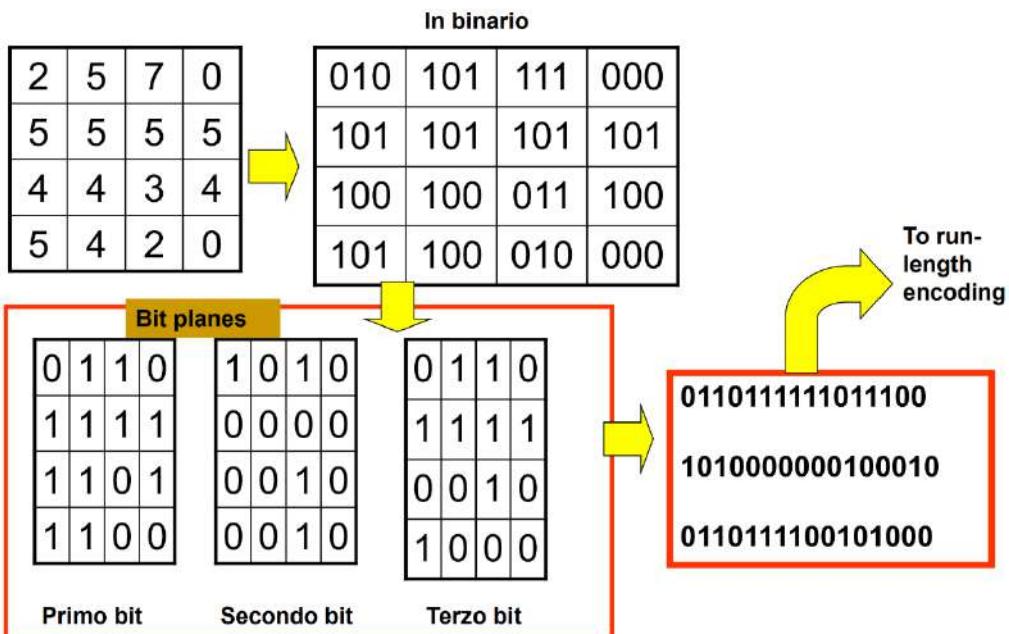
- **Costo aggiuntivo:** si deve memorizzare la tabella caratteri-codici → se i caratteri sono tanti questo può essere costoso.
- **Huffman viene usato per comprimere alcune informazioni nella fase finale della codifica JPEG.**

## Altri algoritmi LOSSLESS

- Run-Length Encoding (RLE): rappresentazione di sequenze ripetute.
  - Se lungo le righe (o colonne) sono presenti ripetizioni di intensità → possono essere compresse rappresentando tali sequenze (run) sottoforma di coppie run-length.
  - L'inizio di una nuova intensità è dato da tali coppie e il numero dei pixel consecutivi ne condividono il valore in questione.
- **Concetti principali:**
- **Esempio:**
  - Sequenza originale: 000001111001011101111111
- **Compressione:**
  - Si rappresentano i segmenti (run) come "numero di occorrenze" di ogni simbolo consecutivo.
  - Esempio: 5 volte 0, 3 volte 1, 2 volte 0, ....
- **Rappresentazione compressa:**
  - La sequenza è espressa solo con le lunghezze dei segmenti: 5, 3, 2, 1, 1, 3, 1, 1, 1, 7.
- **Scrittura in binario:**
  - Le lunghezze dei run vengono memorizzate in formato binario.
  - Non sempre questa codifica è più efficiente, soprattutto se i run sono brevi.
- **Efficienza della compressione:**
  - Funziona bene quando ci sono molte sequenze lunghe di simboli uguali.
  - Se i run sono brevi o frequenti, il risparmio potrebbe essere minimo o inesistente.
  - In sintesi, la **RLE** è efficace su dati con ripetizioni lunghe e uniformi, ma il vantaggio dipende dalla struttura dei dati.

## Run Length + bit planes

Debo memorizzare la seguente immagine a 8 toni di grigio (3 bit depth):



- **Codifica differenziale:** se la sequenza dei valori varia lentamente → memorizzazione del valore iniziale e delle differenze tra valori consecutivi.

- **Esempio:**

- 134, 137, 135, 128, 130, 134, 112, ...
- Ricorderò il valore iniziale 134 e poi la sequenza delle differenze successive: -3, +2, 7, -2, -4, 22, ...
- La sequenza delle differenze ha una entropia minore di quella dei valori originali e quindi richiede meno bit per essere memorizzata.

## Compressione LOSSY

- Introduce perdita di informazione.
- Obiettivo: maggiore compressione rispetto a lossless.
- Applicazioni:
  - MP3 (audio).
  - JPEG (immagini).
  - MPEG (video).

## Requantization

- Riduzione dei livelli disponibili per canale.
- Perdita di qualità visiva proporzionale alla compressione.

## Standard JPEG

- Processo di compressione:

1. **Preprocessing:**

- Trasformazione colori (RGB → YC<sub>b</sub>C<sub>r</sub>).

- Sottocampionamento della crominanza.
- Suddivisione della immagine in sottoimmagini → Divisione in blocchi  $8 \times 8$  pixel.

## 2. Trasformazione:

- Discrete Cosine Transform (DCT).
- Quantizzazione.

## 3. Codifica:

Preprocessing (i): da RGB a  $Y C_b C_r$ . DC Coefficient

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Encoding.  
• Zig-zag ordering of AC Coefficients.

La  $Y$  è la luminanza, gli altri due canali codificano i colori.

Si tratta di una trasformazione reversibile.

Il modello  $Y C_b$  e  $C_r$  è un modello di colori che permette di avvantaggiarsi della "debolezza" del sistema visivo umano.

• Entropy Coding (Huffman).

Sottocampioname

nto della crominanza

- L'occhio umano è più sensibili alla luminanza che alla crominanza
- JPEG prende tutte le informazioni sulla luminanza ma ne prende solo un campione dagli altri canali
  - Si sceglie 1 valore su 4 per  $C_b$  e  $C_r$

## Prima della DCT

- Viene sottratta ai 64 pixel di ciascun blocco una quantità pari a  $2^{n-1}$
- $2^n$  rappresenta il numero massimo di livelli di grigio dell'immagine
- Shift dei livelli di grigio = il grigio medio (128) diventa 0.

## Trasformazione (i): la DCT

- JPEG trasforma i blocchi  $8 \times 8$  di pixel secondo l'algoritmo della Trasformata Discreta del Coseno (DCT)
- Si tratta di un algoritmo della famiglia delle trasformate di Fourier.
- "Decorrela" al massimo i dati permettendo maggiori rapporti di compressione nella fase successiva di codifica
- Un'immagine di  $8 \times 8$  pixel si può pensare come un vettore nello spazio a 64 dimensioni.
- Immagine = somma pesata di 64 immagini impulsive → che andranno a costituire la "base impulsiva" → La trasformata del coseno esprime l'immagine in un'altra base.



## La formula della DCT per un blocco di dimensioni NxN (N=8 nel jpeg)

$$F(u,v) = \frac{2}{N} \left[ \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(u)C(v)f(x,y) \cos \frac{(2x+1)u\pi}{2*N} \cos \frac{(2y+1)v\pi}{2*N} \right]$$

$$f(x,y) = \frac{2}{N} \left[ \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cos \frac{(2x+1)u\pi}{2*N} \cos \frac{(2y+1)v\pi}{2*N} \right]$$

where:

$$C(u) = \frac{1}{\sqrt{2}} \text{ for } u = 0; C(u) = 1 \text{ otherwise}$$

$$C(v) = \frac{1}{\sqrt{2}} \text{ for } v = 0; C(v) = 1 \text{ otherwise}$$

**Una implementazione diretta delle formule sopra richiede O(N<sup>2</sup>) Esistono algoritmi "fast" per calcolare i coefficienti in O(N log(N)) derivati dalla Fast Fourier Transform.**



## Esempio

- E sono i coefficienti che devono essere moltiplicati alle basi della DCT (NON a quella canonica!) per ottenere il blocco precedente.

$$\begin{matrix} -415 & -29 & -62 & 25 & 55 & -20 & -1 & 3 \\ 7 & -21 & -62 & 9 & 11 & -7 & -6 & 6 \\ -46 & 8 & 77 & -25 & -30 & 10 & 7 & -5 \\ -50 & 13 & 35 & -15 & -9 & 6 & 0 & 3 \\ 11 & -8 & -13 & -2 & -1 & 1 & -4 & 1 \\ -10 & 1 & 3 & -3 & -1 & 0 & 2 & -1 \\ -4 & -1 & 2 & -1 & 2 & -3 & 1 & -2 \\ -1 & -1 & -1 & -2 & -1 & -1 & 0 & -1 \end{matrix}$$



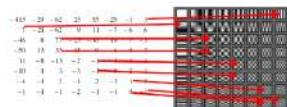
## Esempio

- Dopo l'applicazione della DCT, i nostri coefficienti diventano:

$$\begin{matrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -62 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -63 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{matrix} \quad \text{DCT} \quad \begin{matrix} -415 & -29 & -62 & 25 & 55 & -20 & -1 & 3 \\ 7 & -21 & -62 & 9 & 11 & -7 & -6 & 6 \\ -46 & 8 & 77 & -25 & -30 & 10 & 7 & -5 \\ -50 & 13 & 35 & -15 & -9 & 6 & 0 & 3 \\ 11 & -8 & -13 & -2 & -1 & 1 & -4 & 1 \\ -10 & 1 & 3 & -3 & -1 & 0 & 2 & -1 \\ -4 & -1 & 2 & -1 & 2 & -3 & 1 & -2 \\ -1 & -1 & -1 & -2 & -1 & -1 & 0 & -1 \end{matrix}$$



Quindi, se volessimo tornare indietro:



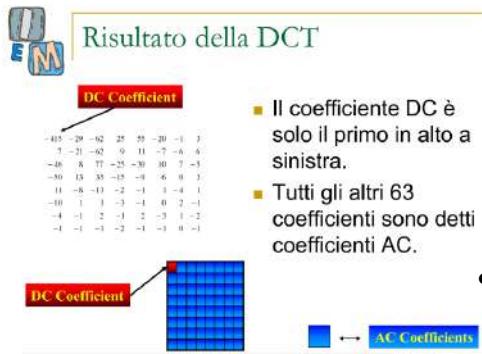
Somma di prodotti tra 64 coefficienti e 64 basi della DCT di 8x8 elementi ciascuno!

=



Lo stesso calcolo va ripetuto per tutti gli elementi della matrice.

$$\begin{matrix} -415 & -29 & -62 & 25 & 55 & -20 & -1 & 3 \\ 7 & -21 & -62 & 9 & 11 & -7 & -6 & 6 \\ -46 & 8 & 77 & -25 & -30 & 10 & 7 & -5 \\ -50 & 13 & 35 & -15 & -9 & 6 & 0 & 3 \\ 11 & -8 & -13 & -2 & -1 & 1 & -4 & 1 \\ -10 & 1 & 3 & -3 & -1 & 0 & 2 & -1 \\ -4 & -1 & 2 & -1 & 2 & -3 & 1 & -2 \\ -1 & -1 & -1 & -2 & -1 & -1 & 0 & -1 \end{matrix}$$



quantizzazione e  $F =$   
numero:

- Il coefficiente DC è solo il primo in alto a sinistra.
- Tutti gli altri 63 coefficienti sono detti coefficienti AC.

## Traformazione(ii): Quantizzazione

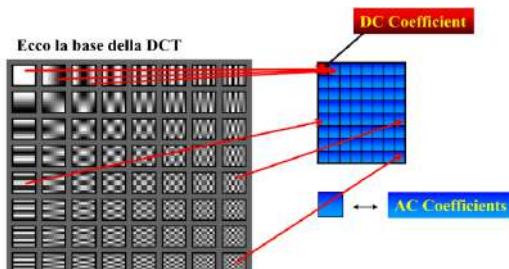
- Dato  $Q =$  fattore di

Per la prima riga si ottiene:

$$F_{\text{quantizzato}} = \text{round}(F / Q)$$

- 415 \* - 29 \* - 62 \* + 25 \*   
 + 55 \* - 20 \* - 1 \* + 3 \* + ...

- Ognuna delle basi (cioè delle piccole immagini riportate sopra) è grande 8x8 pixel



Ogni immagine 8 x 8 si ottiene moltiplicando ciascuna delle immagini a sinistra per un coefficiente e sommando tutte le immagini.

I coefficienti di tale somma sono i coefficienti della DCT.

Il coefficiente in alto a sinistra è un valore proporzionale al valor medio della luminanza dell'immagine. E' detto anche coefficiente DC

- il valore ricostruito si ottiene con  $\rightarrow F_{\text{quantizzato}} * Q$
- E' un processo irreversibile**
- Non è conveniente usare un unico fattore di quantizzazione per 64 coefficienti della DCT della luminanza, o per quantizzare i valori provenienti dalla DCT delle crominanze**

- Si preferisce adottare per il coefficiente  $F(i,j)$  un fattore di

quantizzazione  $Q(i,j)$  scelto a priori o dall'utente → i fattori  $Q(i,j)$  costituiscono la “tabella di quantizzazione”



Nel nostro esempio

-40	-29	-62	23	35	-20	-1	3
-31	-19	-29	1	3	12	12	11
-46	8	-25	-30	10	7	-5	14
-56	13	35	-15	-9	6	0	13
11	-8	-13	-2	-1	-4	-1	18
-10	1	7	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-3	-2	-1	-1	0	-1



-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

Tabella di quantizzazione luminanza

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

Tabella di quantizzazione crominanza

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

Si osservi che un fattore di compressione maggiore comporta una maggiore perdita di informazione.

L'utente del JPEG può scegliere il “grado” di quantizzazione da adottare fornendo un “quality factor” QF che va da 1 a 100. La tabella di quantizzazione adottata sarà una copia delle tabelle sopra i cui elementi sono divisi per QF.

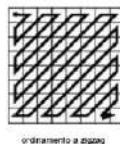
Maggiore il QF, minore i fattori di quantizzazione e minore la perdita di informazioni.



Codifica



Codifica zig-zag



ordinamento a zigzag

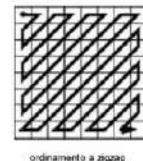
0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Gli indici che determinano l'ordinamento a zig-zag dei coefficienti quantizzati



Nel nostro esempio

-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



ordinamento a zigzag

-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

## Differenti codifiche

- 2 differenti codifiche:

- I coefficienti DC → stanno nella posizione (1,1) del blocco 8 x 8, sono codificati usando una **codifica differenziale**.

SSSS	$\Delta$
0	0
1	-1, 1
2	-3, -2, 2, 3
3	-7 ... -4, 4 ... 7
4	-15 ... -8, 8 ... 15
5	-31 ... -16, 16 ... 31
6	-63 ... -32, 32 ... 63
7	-127 ... -64, 64 ... 127
8	-255 ... -128, 128 ... 255
9	-511 ... -256, 256 ... 511
10	-1023 ... -512, 512 ... 1023
11	-2047 ... -1024, 1024 ... 2047

SSSS = categoria  
 $\Delta$  = differenza tra due coefficienti DC (eventuo)

- Il valore n=SSSS=4 ha come codice base 101. Ed n è anche il numero di bit mancanti che occorre aggiungere.
- La corrispondenza tra il valore e il codice è fissata dalla tabella dei codici di Huffman che varia in base al fatto che stiamo trattando la luminanza o la crominanza.

SSSS	Codice base
0	010
1	10
2	100
3	00
4	101
5	110
6	1110
7	11110
8	111110
9	1111110
10	11111110
11	11111110

n.b.: le tabelle complete sono disponibili su team

o:

- Per completare il codice → si usa la seguente regola:

- **Se  $\Delta > 0$ :** i bit da aggiungere sono gli n bit meno significativi del valore  $\Delta$  in binario.
- **Se  $\Delta < 0$ :** i bit da aggiungere sono gli n bit meno significativi del valore in binario di  $\Delta$  (con complemento a due) → si deve sottrarre il valore 1
- **Se  $\Delta = 0$ :** anche SSSS = 0 → non viene aggiunto alcun bit.

- **I coefficienti AC** → tutti gli altri del blocco, sono codificati usando una **codifica run-length**.
- Si elimina dalla sequenza il primo coefficiente.
- Dato che si hanno per lo più coefficienti nulli → **trasformazione in skip-value** = memorizzazione in una sequenza di valori del numero degli zeri seguito dal primo valore diverso da 0 che si incontra → coppia (0, v) o coppia (run, categoria)

(run, category)	Codice base	Lunghezza codice completo
(0, 0)	1010 (= EOB)	4
(0, 1)	00	3
(0, 2)	01	4
(0, 3)	100	6
...	...	...
(F, 0)	111111110111	12
...	...	...
(F, A)	1111111111111110	26

SSSS	Coefficiente AC
1	-1, 1
2	-3 -2, 2 3
3	-7 ... -4, 4 ... 7
4	-15 ... -8, 8 ... 15
5	-31 ... -16, 16 ... 31
6	-63 ... -32, 32 ... 63
7	-127 ... -64, 64 ... 127
8	-255 ... -128, 128 ... 255
9	-511 ... -256, 256 ... 511
A	-1023 ... -512, 512 ... 1023

- Per completare il codice:
  - Se  $v > 0$ : stesso discorso dei coefficienti DC
  - Se  $v < 0$ : //
  - Se  $v = 0$ : //

### Nella ricostruzione...

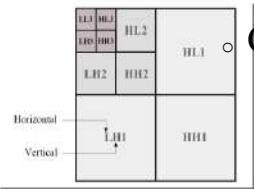
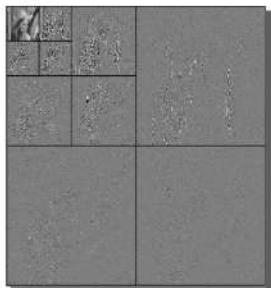
- “**ricostruzione**” dei dati originali o delle approssimazioni per i passi irreversibili.
- **Strategie di ricostruzione** → abilità la ricostruzione progressiva, gerarchica, lossless

### Effetti della compressione JPEG

- Divisione dell’immagine in “quadrotti” 8 x 8 di 64 pixel non sovrapposti.

- Effetti visivi della "quadrettatura".
- Impatto del fattore di qualità (QF) sulla compressione e qualità.

### 2-D Wavelet decomposition



### Nuovi standard

#### • JPEG2000:

- Usa le wavelets invece della DCT.

◦ Compressione più efficiente ma scarso successo commerciale.