

Analisi di Immagini e Video (Computer Vision)

Giuseppe Manco

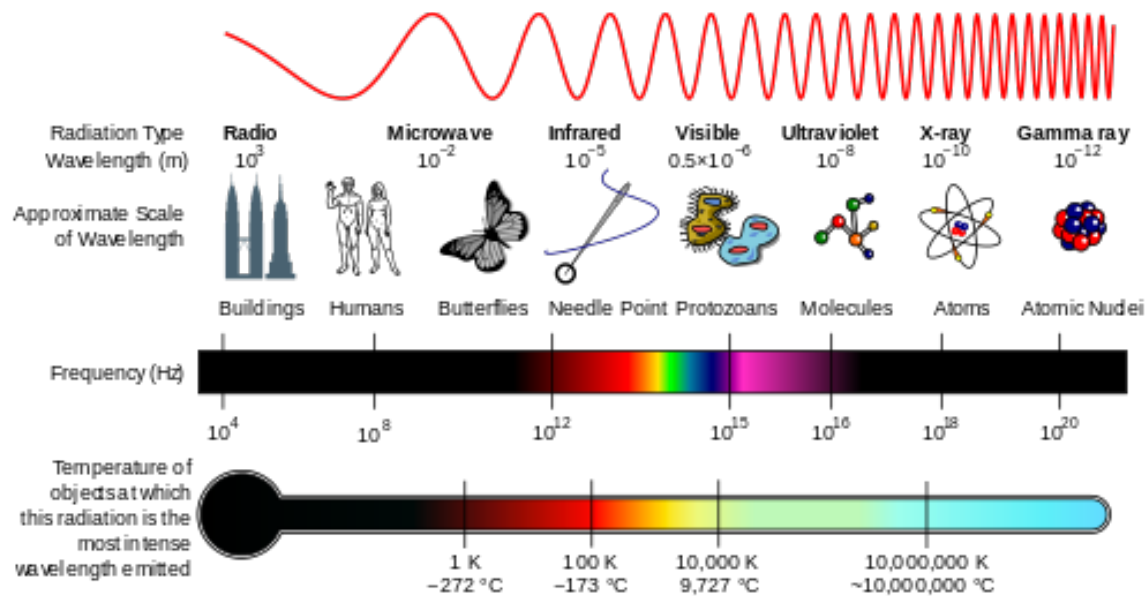
Outline

- Concetti introduttivi
 - Immagini
 - Operazioni di base

Concetti Introduttivi

Immagini

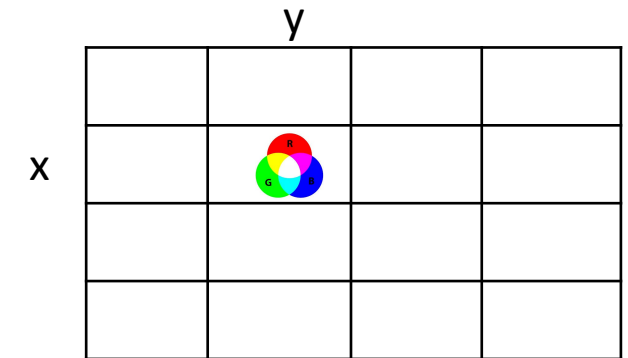
- Rappresentazione bidimensionale dello spettro di luce (visibile)



https://en.wikipedia.org/wiki/Electromagnetic_spectrum

$$\lambda = \frac{c}{\nu} \quad E = h\nu$$

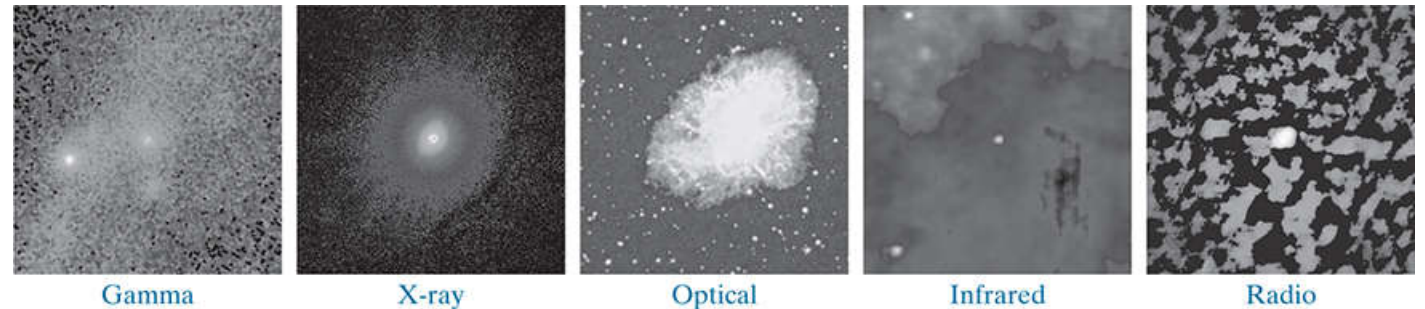
Color	Wavelength	Frequency	Photon energy
Violet	380–450 nm	680–790 THz	2.95–3.10 eV
Blue	450–485 nm	620–680 THz	2.64–2.75 eV
Cyan	485–500 nm	600–620 THz	2.48–2.52 eV
Green	500–565 nm	530–600 THz	2.25–2.34 eV
Yellow	565–590 nm	510–530 THz	2.10–2.17 eV
Orange	590–625 nm	480–510 THz	2.00–2.10 eV
Red	625–740 nm	405–480 THz	1.65–2.00 eV



$C(x, y, \lambda)$

Immagini e spettro elettromagnetico

- Gamma rays
 - PET, medicina nucleare
- X-Rays
 - Radiografie, angiografie, TAC
- Ultravioletti
 - Fluorescenza microscopica
- Spettro visibile
 - Immagini standard, immagini satellitari
- Microonde
 - Scansioni radar
- Radio
 - MRI



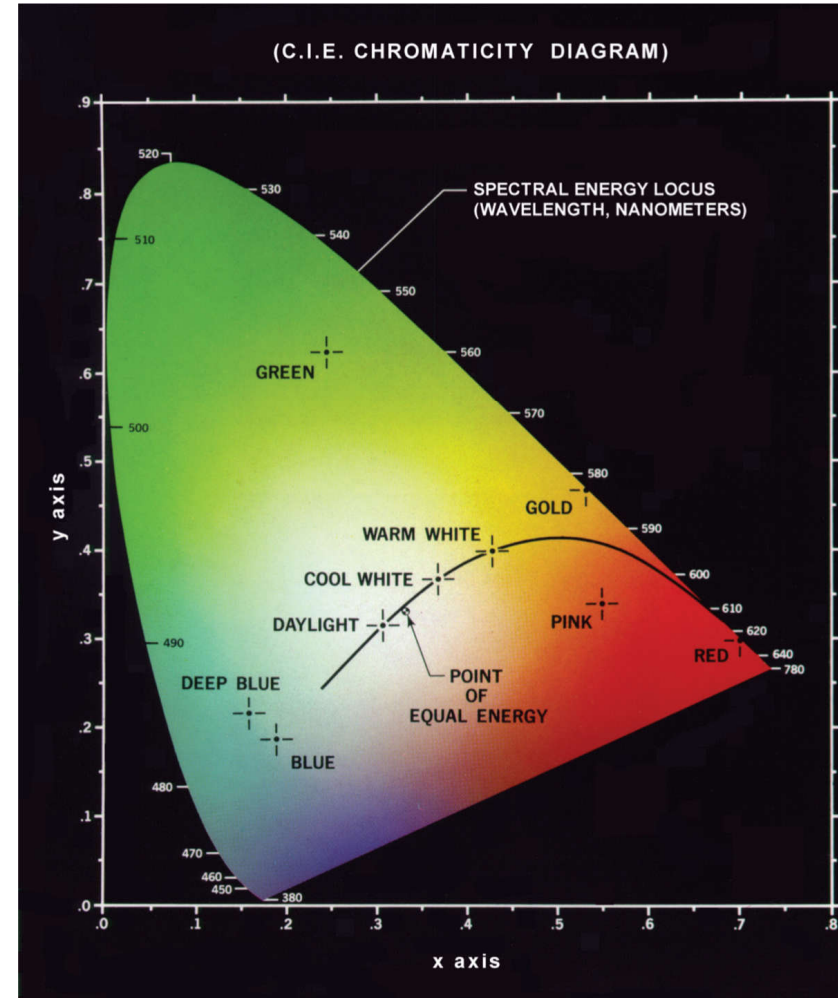
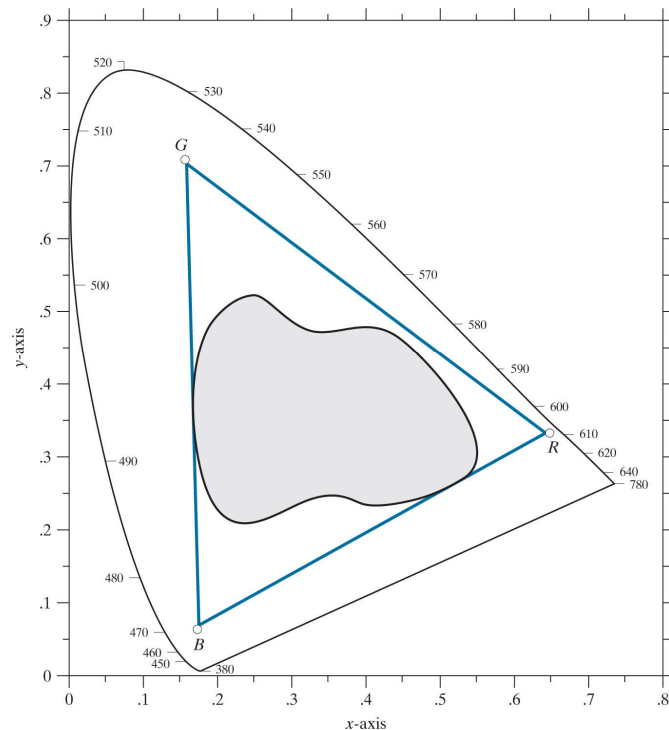
Crab pulsar (NASA. Source: [Gon18])

Luce cromatica

- Spazia nello spettro tra 400 e 700 nm
 - Tre misure
 - Radiance
 - Ammontare di energia che parte dalla sorgente, misurata in Watts
 - Luminance
 - Ammontare di energia percipita da un osservatore. Misurata in lumen
 - Ad esempio, la luce emessa da una sorgente che opera nella regione infrarossa dello spettro
 - Alta radianza, zero luminanza
 - Brightness
 - Misura soggettiva che dipende dalla percezione visiva. Ha senso quando comparata ad un riferimento

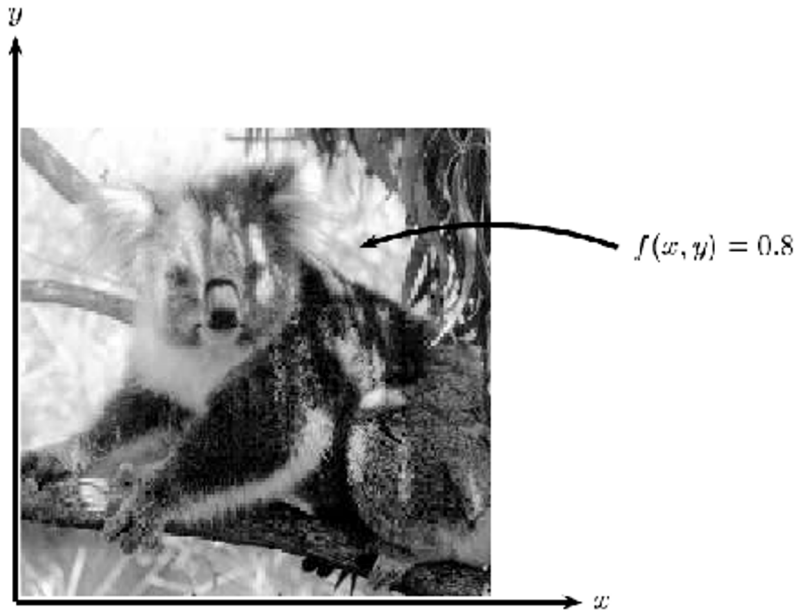
Luce cromatica e colori percepiti

- Quantità di fotoni con una determinata frequenza



Source: [Gon18]

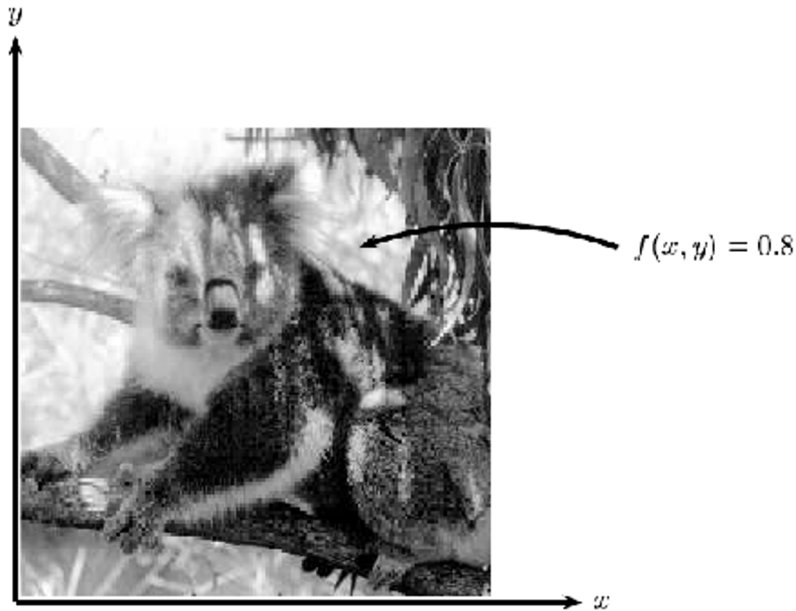
Immagini come funzioni



- Funzione bidimensionale
 $f: [0, X_{max}] \times [0, Y_{max}] \mapsto [0, L_{max}]$
- $f(x, y) = \textit{intensità}$ (o *luminosità*; *brightness*) della posizione (x, y)
- 0 (nero), L_{max} (bianco)

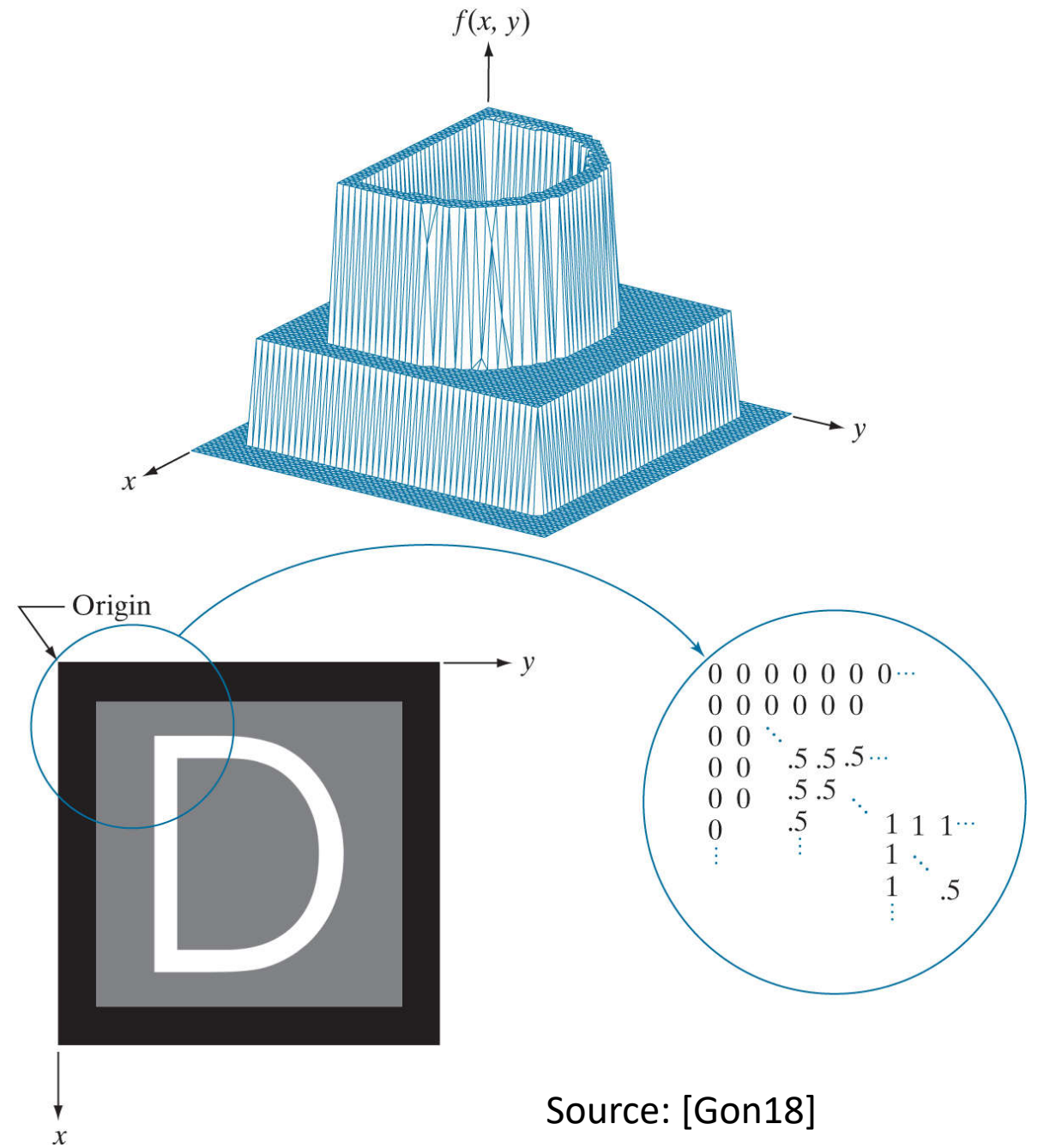
Intensità: misura dell'energia emessa da una sorgente di luce in una direzione particolare.

Immagini digitali

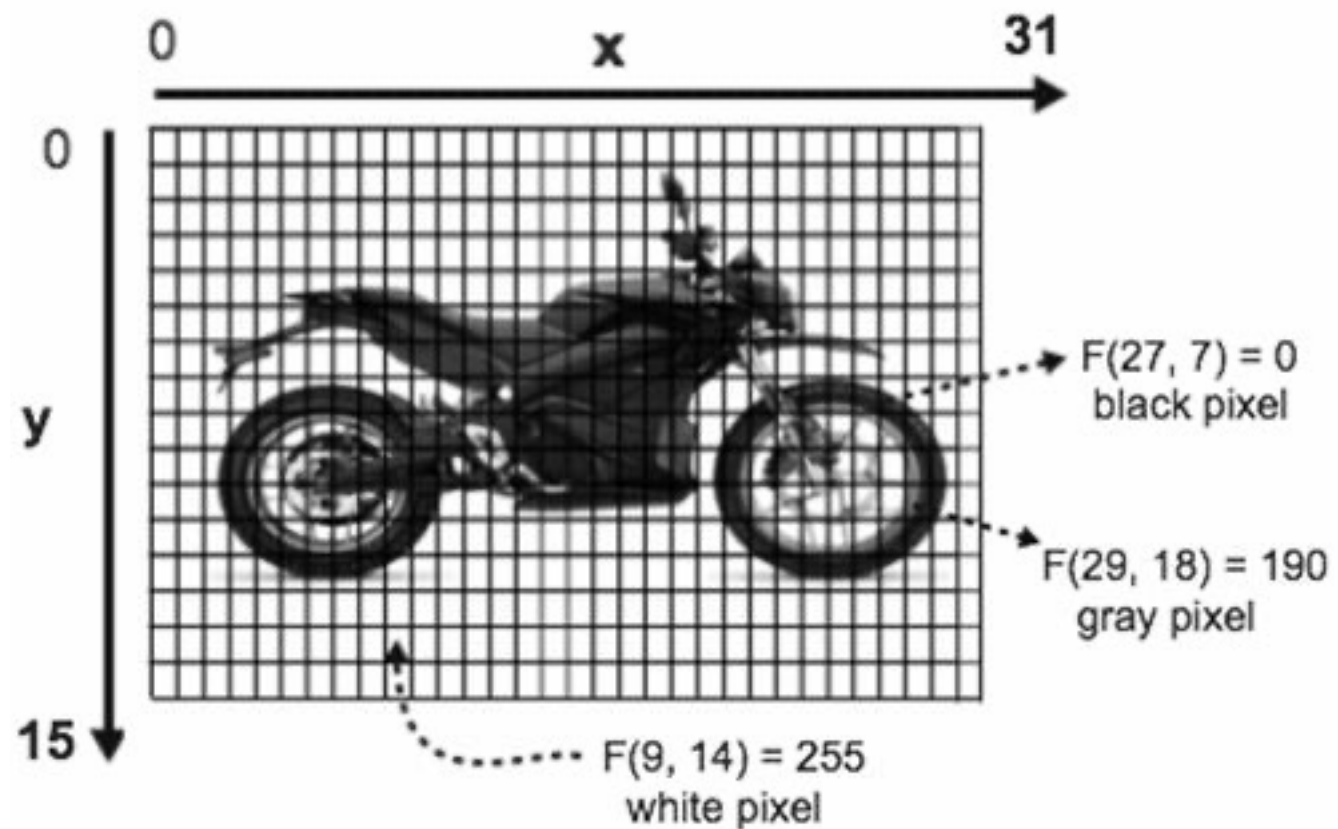


- Immagine **bitmap/raster**
- x, y ed $f(x, y)$ discreti
- Pixel (**p**icture **e**lement)
- Funzione bidimensionale,
 $f: [0, N - 1] \times [0, M - 1] \mapsto [0, L - 1]$
- $f(x, y) = \textit{intensità}$ (o *luminosità*; *brightness*) del *pixel* di posizione (x, y)
- 0 (nero), $L-1$ (bianco)

Immagini digitali



Grayscale image (32 x 16)



Immagini digitali

Rappresentazione matriciale: $f(y,x)$

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix}$$

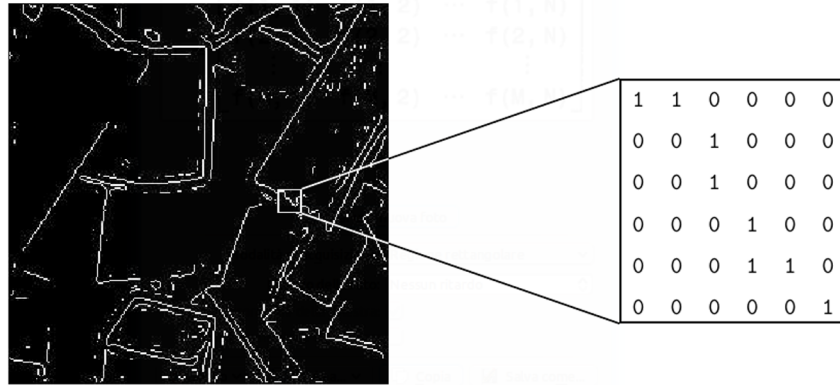
Sistema di coordinate:

- $(0,0)$ è l'angolo superiore sinistro dell'immagine
- $(M-1,N-1)$ è l'angolo inferiore destro dell'immagine

Rappresentazione delle immagini digitali

- Tipi di immagini digitali
 - Binarie
 - A scala di grigi
 - A colori
 - Indicizzate

Immagini binarie



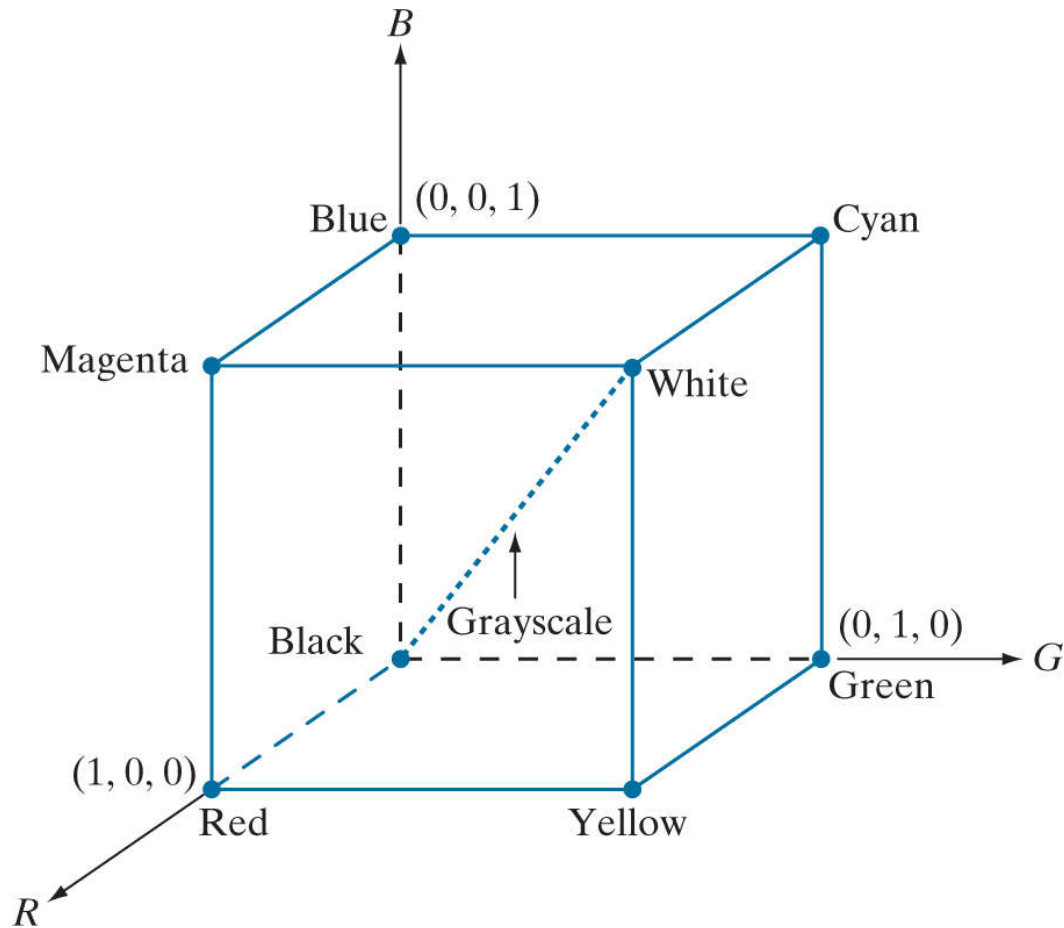
- Ogni pixel è bianco (1) oppure nero (0)
- E' sufficiente 1 bit per pixel (anche se non sempre è così)
- Efficienti in termini di spazio di memoria richiesto
- Adatte per testi, impronte digitali, schemi/piante, ...

Immagini a scala di grigi (grayscale)



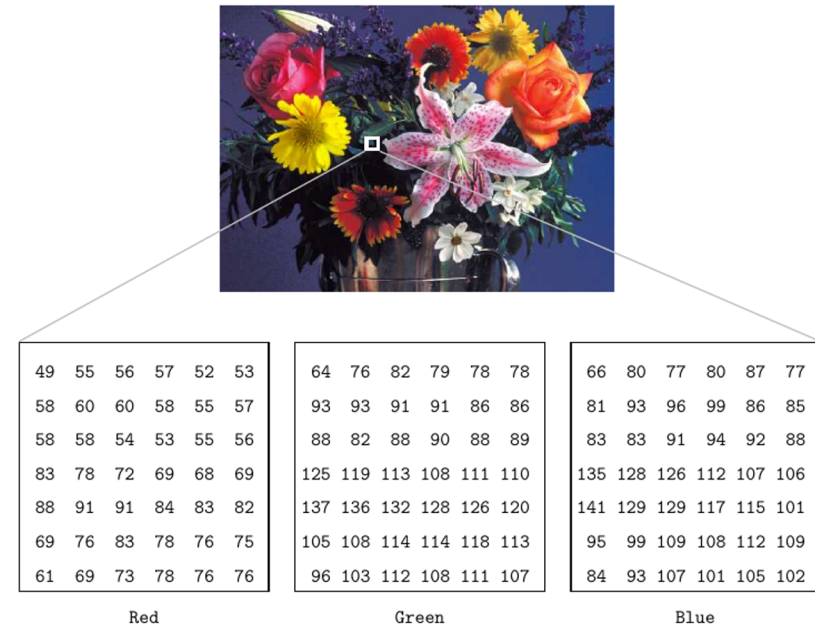
- Ogni pixel è una intensità di grigio
- Ad esempio, ogni pixel può essere rappresentato con 1 byte
- In questo caso valori variano da 0 (nero) a 255 (bianco)
- 256 livelli bastano per riconoscere molti oggetti naturali

Immagini a colori (true color, RGB)



- Color model
- RGB standard per monitor e TV
- Ogni pixel ha una intensità di Red, Green e Blue
- Le intensità di grigio si trovano sulla diagonale $(0,0,0)$ - $(1,1,1)$

Immagini a colori (true color, RGB)



- Ogni pixel ha una intensità di red, green e blue (0-255)
- 24 bit per pixel (3 byte), dette 24-bit color images
- Numero di colori, $255^3 = 16,777,216$ (~16,8M)
- “Stack” di 3 matrici

Immagini indicizzate



- Molte immagini usano solo alcuni colori
- Mappa o tavolozza dei colori (*color map/palette*): lista dei colori disponibili (es. GIF ammette 256 colori)
- Valore di ogni pixel: indice del colore nella mappa

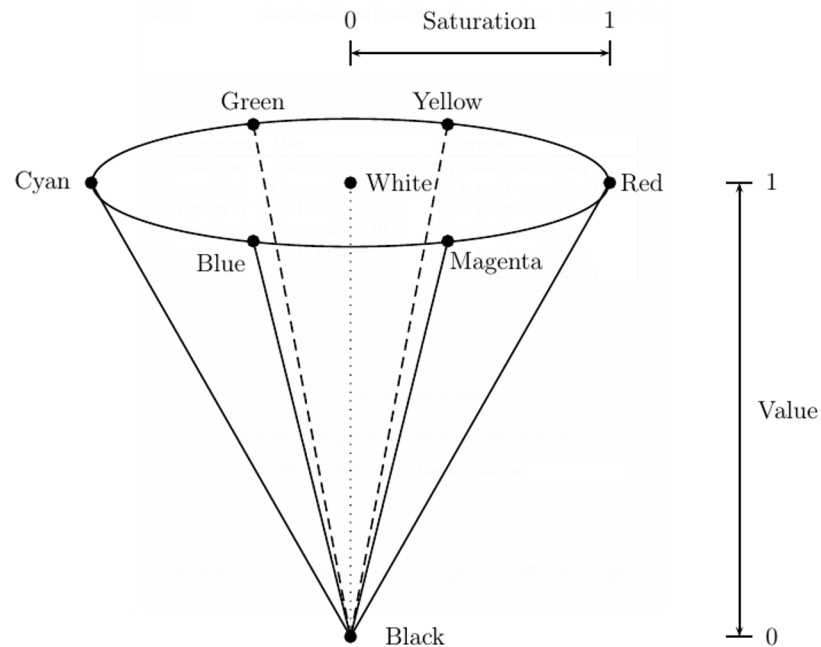
Manipolazione di immagini in Python

- Breve panoramica sulle librerie principali
 - Focus su NumPy e Matplotlib
- Apertura di un'immagine e analisi della sua rappresentazione
 - Scala di grigi
 - Binaria
 - Colore
- Conversione di formati
- Riferimenti: notebook disponibile al seguente indirizzo
 - https://github.com/gmanco/cv_notebooks/blob/master/lezione1.ipynb

Altri formati

- HSV, HSL, LUV, XYZ, Lab, CMYK, etc

HSV color model



- **Hue** (colore, tinta): il “vero colore”
- **Saturation** (saturazione, sfumatura): quanto è diluito col bianco
- **Value**: il grado di luminosità o intensità
- Conversione da RGB a grayscale:
- $RGB \rightarrow HSV \rightarrow V$

Operazioni di base

Che tipi di trasformazioni si possono fare?



Filtraggio



Cambio dei valori dei pixel

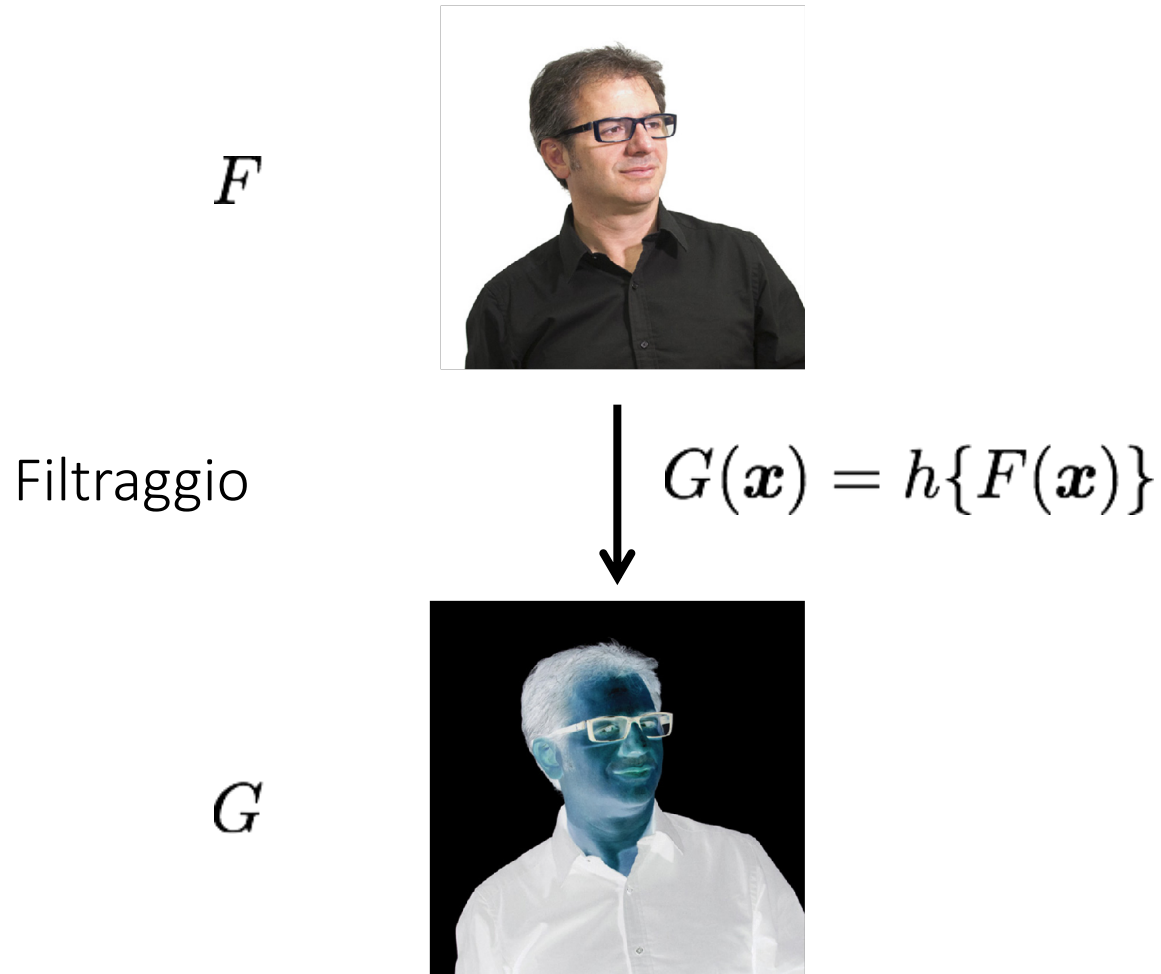


Warping

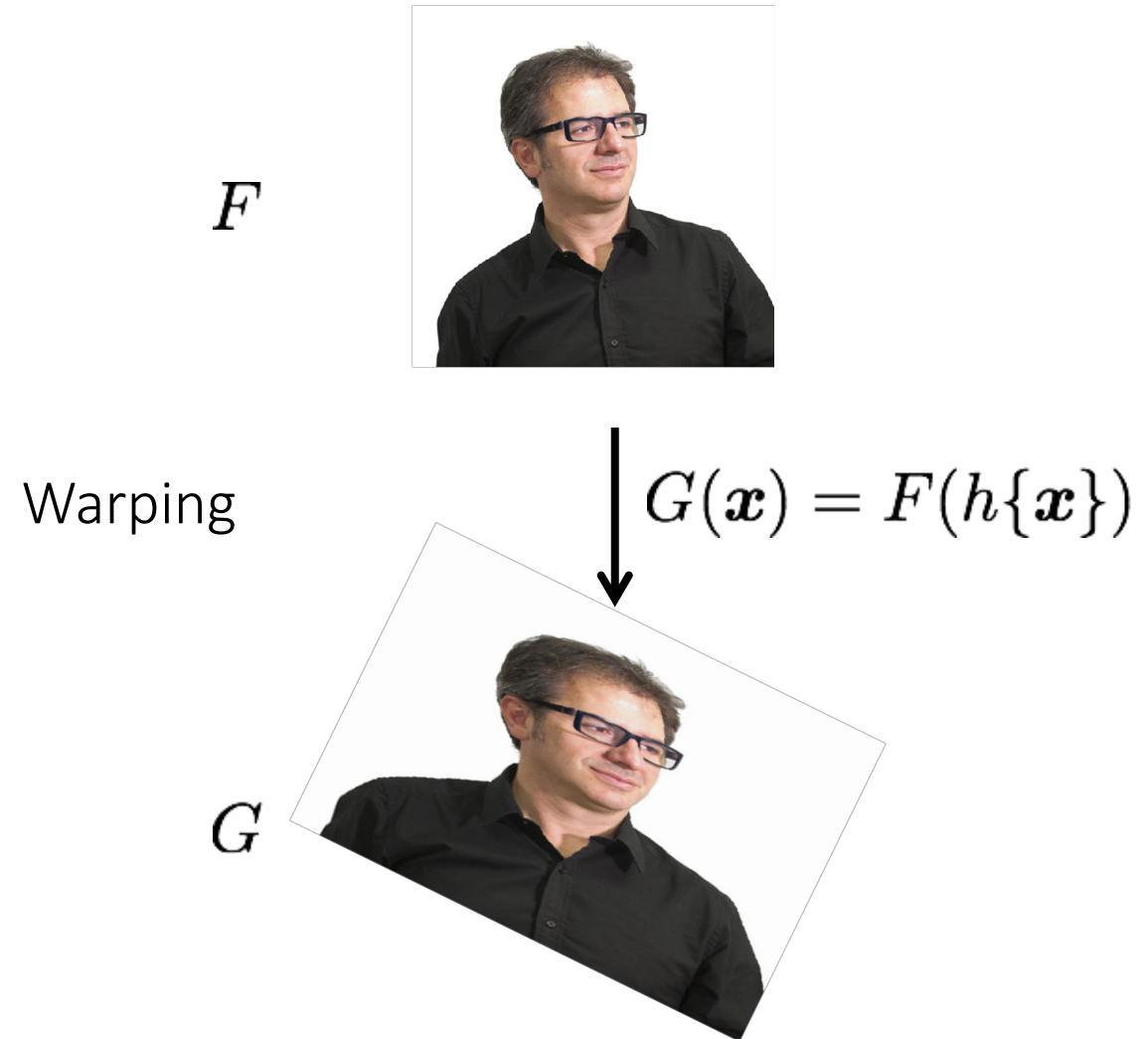


Cambio delle posizioni dei pixel

Che tipi di trasformazioni si possono fare?



Cambio dei valori dei pixel



Cambio delle posizioni dei pixel

Trasformazioni geometriche

- Trasformazioni di coordinate
- Interpolazione di intensità

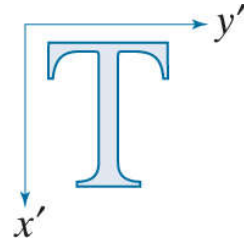
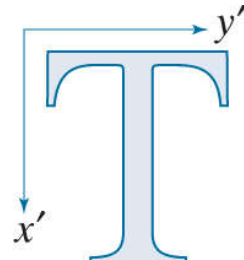
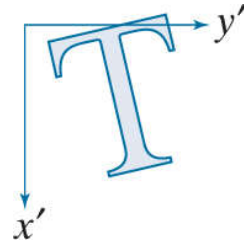
Trasformazioni di coordinate

- Trasformazioni affini

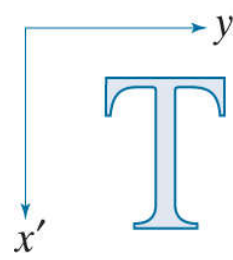
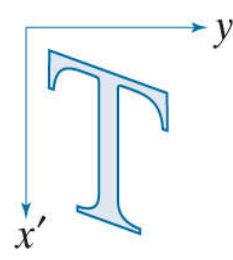
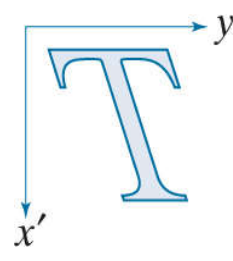
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Preserva punti, linee, piani
- Trasformazioni in scala, rotazioni, traslazioni, curvature

Trasformazioni di coordinate

Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x' &= x \\ y' &= y \end{aligned}$	
Scaling/Reflection (For reflection, set one scaling factor to -1 and the other to 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x' &= c_x x \\ y' &= c_y y \end{aligned}$	
Rotation (about the origin)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x' &= x \cos\theta - y \sin\theta \\ y' &= x \sin\theta + y \cos\theta \end{aligned}$	

Trasformazioni di coordinate

Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x' &= x + t_x \\ y' &= y + t_y \end{aligned}$	
Shear (vertical)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x' &= x + s_v y \\ y' &= y \end{aligned}$	
Shear (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x' &= x \\ y' &= s_h x + y \end{aligned}$	

Trasformazioni di coordinate

- Forward mapping

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Inverse mapping

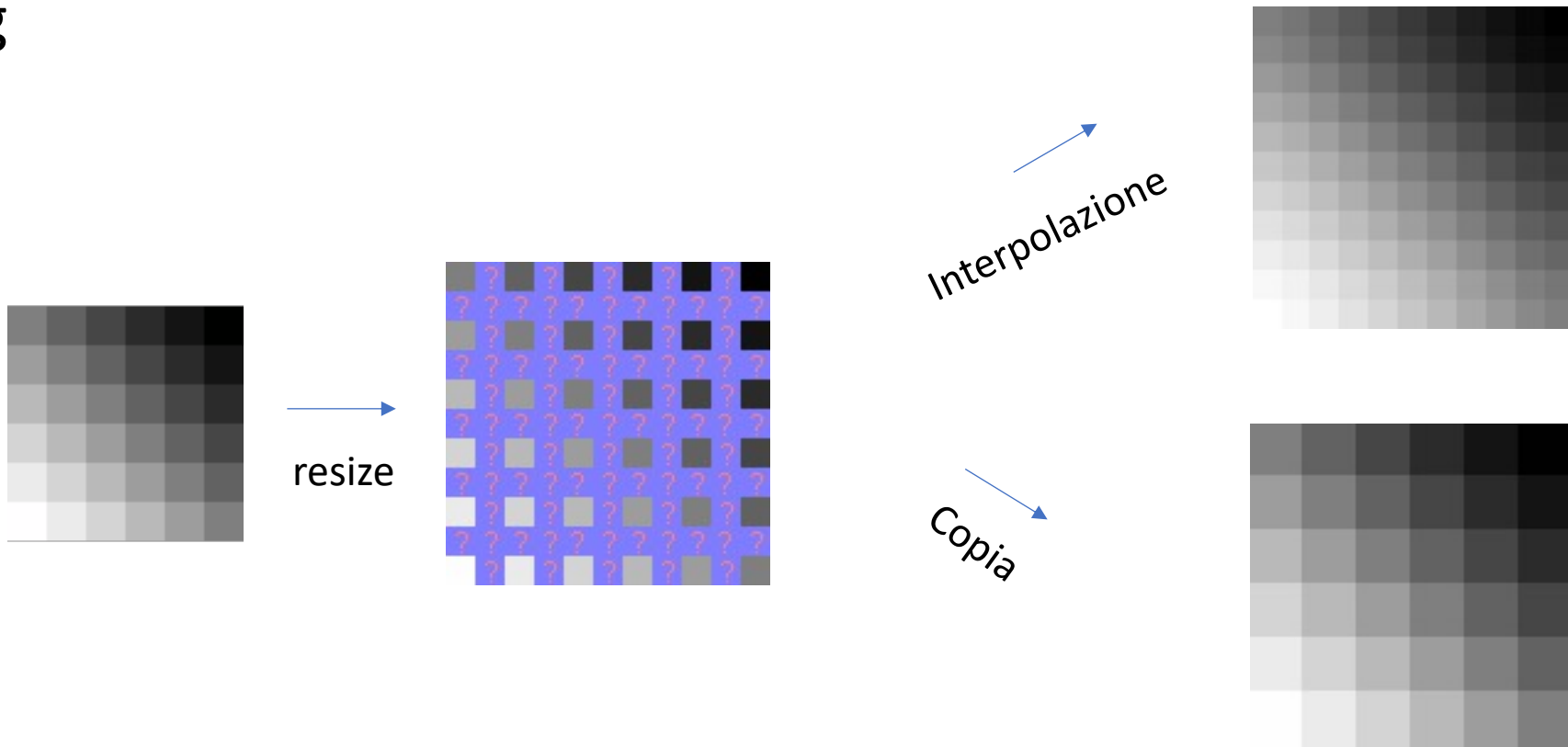
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

- Le operazioni si possono cumulare

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = (\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Interpolazione

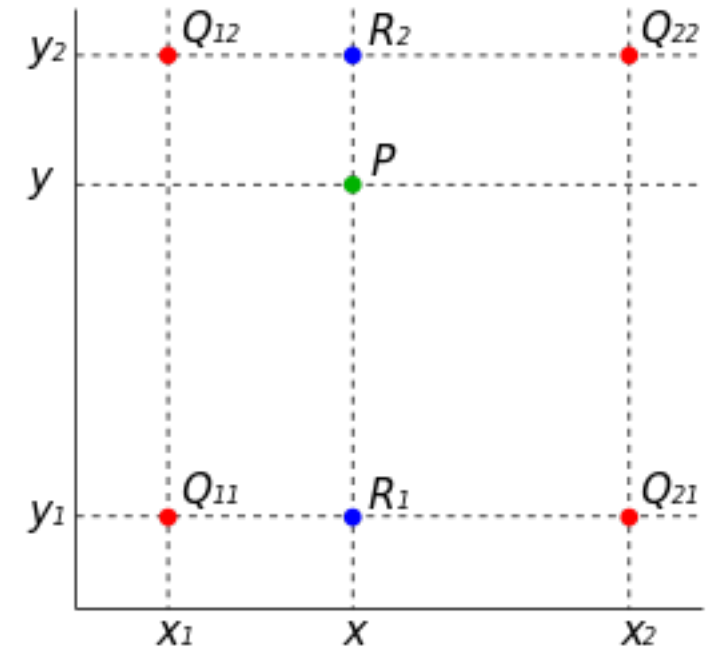
- Resizing



Interpolazione

- Bilineare

$$\begin{aligned} f(x, y) = & \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{11}) \\ & + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{21}) \\ & + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{12}) \\ & + \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} f(Q_{22}) \end{aligned}$$

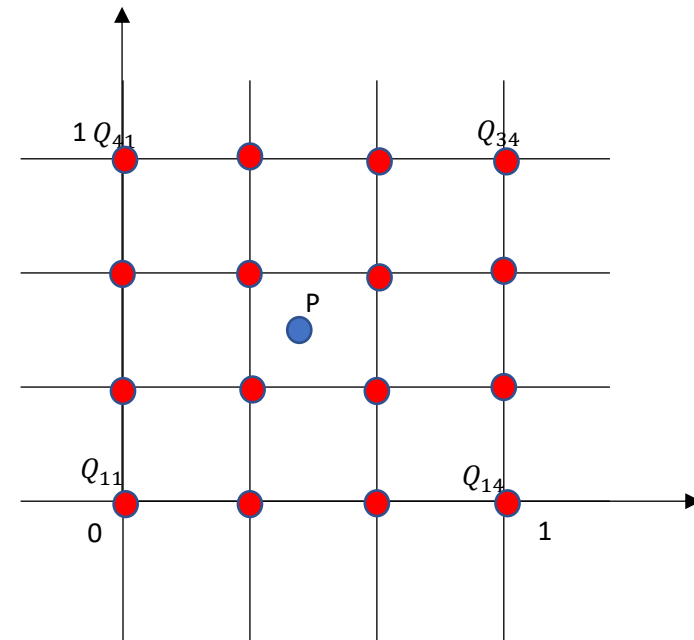


Interpolazione

- Bicubica

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

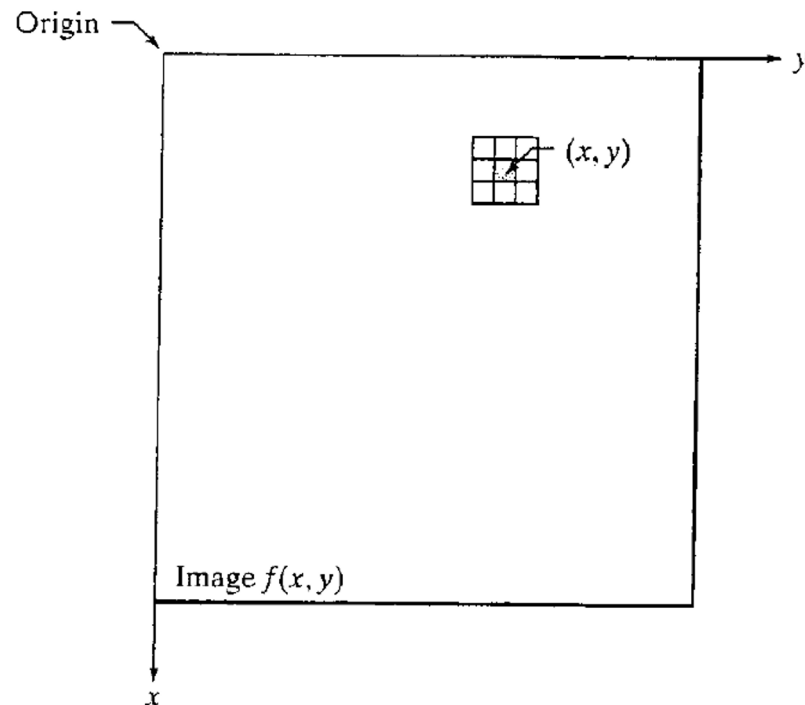
- \tilde{x} e \tilde{y} rappresentano la posizione di P nel quadrato unitario formato dai 16 vicini più vicini
- I coefficienti $a_{i,j}$ sono ottenuti utilizzando il sistema di equazioni sui 16 punti $Q_{i,j}$



Elaborazione nel dominio spaziale

- Il termine **dominio spaziale** fa riferimento al piano contenente i pixel dell'immagine
- Elaborazioni nel dominio spaziale:
 - **Trasformazioni d'intensità** (point operations)
Lavorano sul singolo pixel
 - **Filtraggio spaziale** (neighborhood processing)
Lavorano sul vicinato spaziale del pixel
- Da ora consideriamo immagini in scala di grigi

Elaborazione nel dominio spaziale



- Forma generale:
 - $g(x, y) = T[N[f(x, y)]]$
- $N[f(x, y)]$ è un **vicinato (neighborhood)** del pixel di coordinate (x, y)
- T è la **trasformazione spaziale (operatore)** definito sul vicinato
- T lavora unicamente sui punti del vicinato spaziale di (x, y) (“finestra”)

Trasformazioni d'intensità (point operations)

- Se il vicinato si riduce al pixel stesso (1x1) allora $g(x,y)$ dipende solo da $f(x,y)$
- Parliamo di trasformazione d'intensità o point operation
 - $s = T(r)$
- dove r è l'intensità di un generico pixel $f(x,y)$ ed s è la corrispondente intensità del pixel $g(x,y)$
 - $T : [0, L-1] \rightarrow [0, L-1]$

Trasformazioni d'intensità: operazioni aritmetiche

- **Operazioni aritmetiche:**

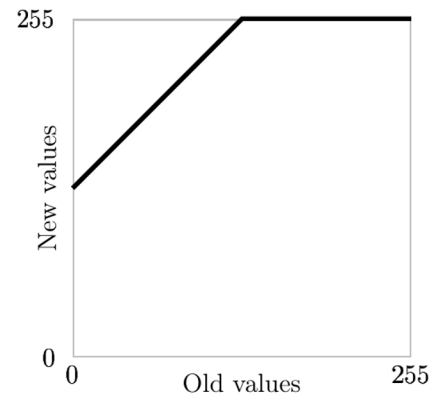
- $s = T(r)$

- con $T(r)=r+C$, $T(r)=r-C$, $T(r)=C*r$ o $T(r)=r/C$

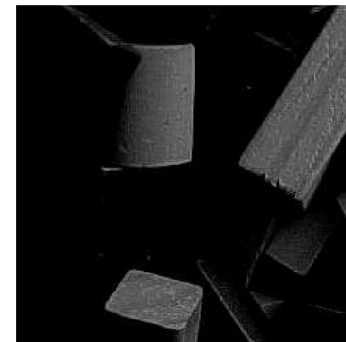
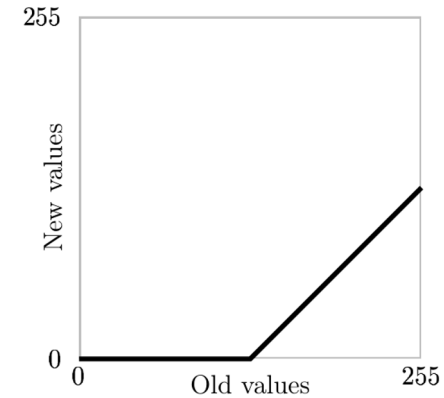
- **Clipping:**

$$s = \begin{cases} 255 & \text{se } r > 255 \\ 0 & r < 0 \end{cases}$$

Trasformazioni d'intensità: addizione e sottrazione

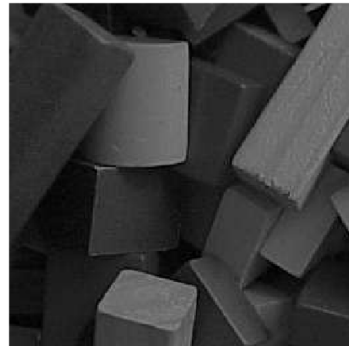
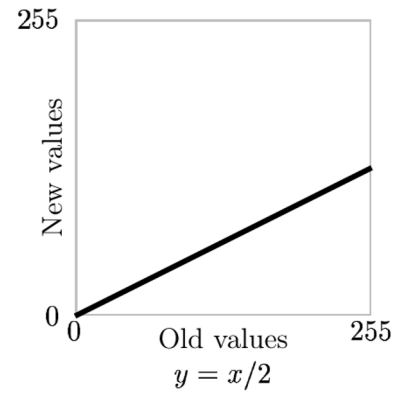


b1: Adding 128

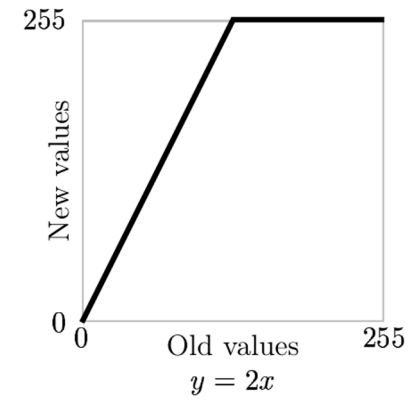


b2: Subtracting 128

Trasformazioni d'intensità: moltiplicazione e divisione



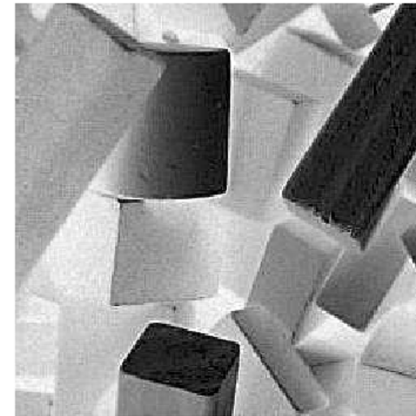
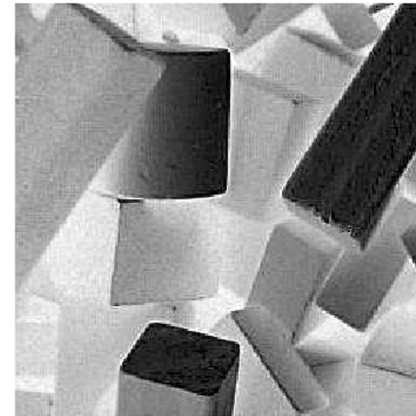
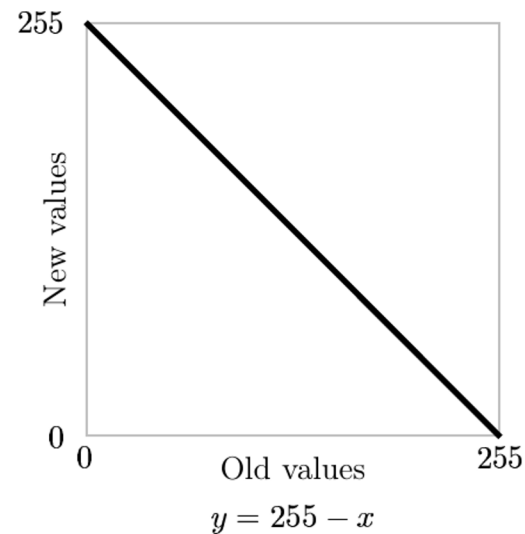
b3: $u = x/2$



b4: $y = 2x$

Trasformazioni d'intensità: immagine negativa o complemento

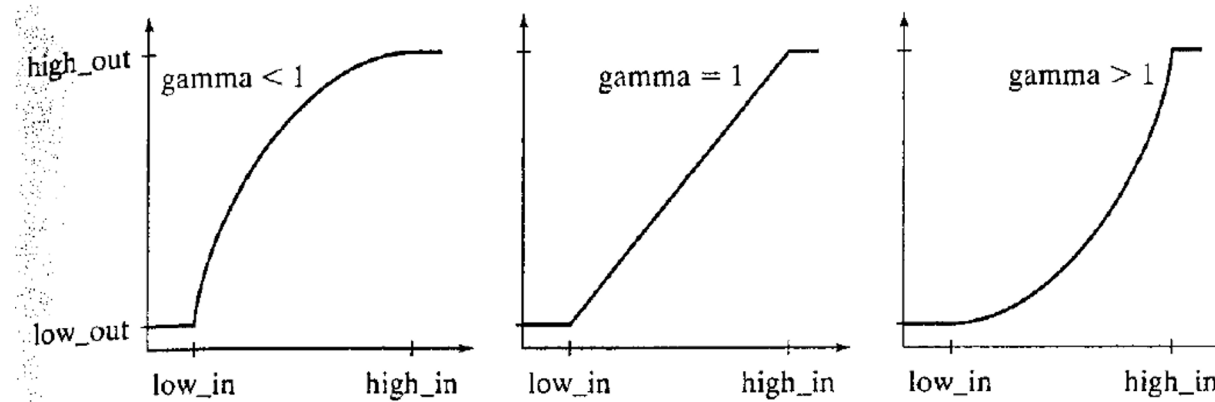
- $T(r) = 1-r$
- Utile per evidenziare zone bianche o grigie all'interno di regioni scure, specialmente quando queste sono dominanti



Dynamic range e contrasto

- **Dynamic range:** intervallo tra la minima e la massima intensità presenti in un'immagine
- **Contrasto:** differenza tra l'intensità massima e minima presenti in un'immagine (nitidezza)
- High dynamic range → alto contrasto
- Low dynamic range → sbiadita (washed-out)

Trasformazioni lineari ed esponenziali



- i valori d'intensità sono compresi tra 0 ed 1
- *gamma* determina la forma della curva : $T(r) = r^\gamma$

Riassumendo

original



Riassumendo

original



x

Riassumendo

original



x

darken



$x - 128$

Riassumendo

original



$$x$$

darken



$$x - 128$$

lower contrast



$$\frac{x}{2}$$

Riassumendo

original



$$x$$

darken



$$x - 128$$

lower contrast



$$\frac{x}{2}$$

non-linear lower contrast



$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

Riassumendo

original



$$x$$

darken



$$x - 128$$

lower contrast



$$\frac{x}{2}$$

non-linear lower contrast



$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

invert



$$255 - x$$

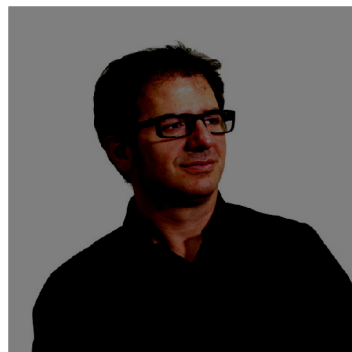
Riassumendo

original



$$x$$

darken



$$x - 128$$

lower contrast



$$\frac{x}{2}$$

non-linear lower contrast



$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

invert



$$255 - x$$

lighten



$$x + 128$$

Riassumendo

original



$$x$$

darken



$$x - 128$$

lower contrast



$$\frac{x}{2}$$

non-linear lower contrast



$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

invert



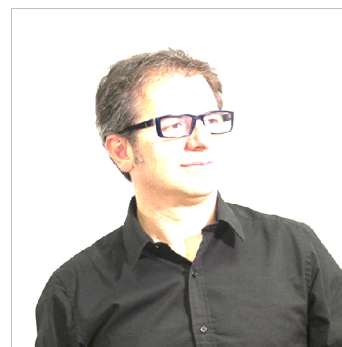
$$255 - x$$

lighten



$$x + 128$$

raise contrast



$$x \times 2$$

Riassumendo

original



$$x$$

darken



$$x - 128$$

lower contrast



$$\frac{x}{2}$$

non-linear lower contrast



$$\left(\frac{x}{255}\right)^{1/3} \times 255$$

invert



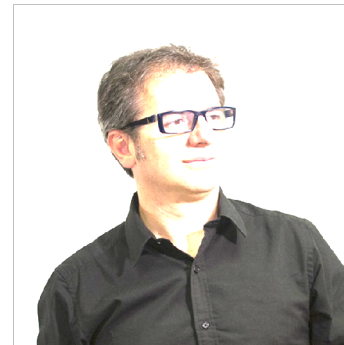
$$255 - x$$

lighten



$$x + 128$$

raise contrast



$$x \times 2$$

non-linear raise contrast



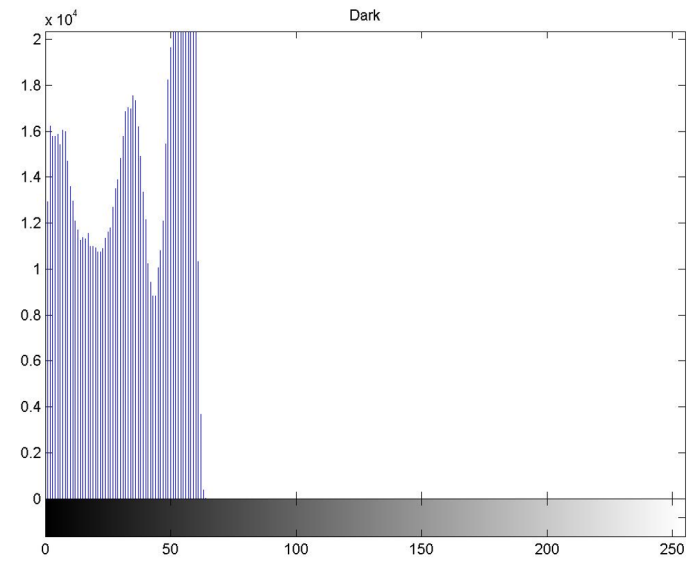
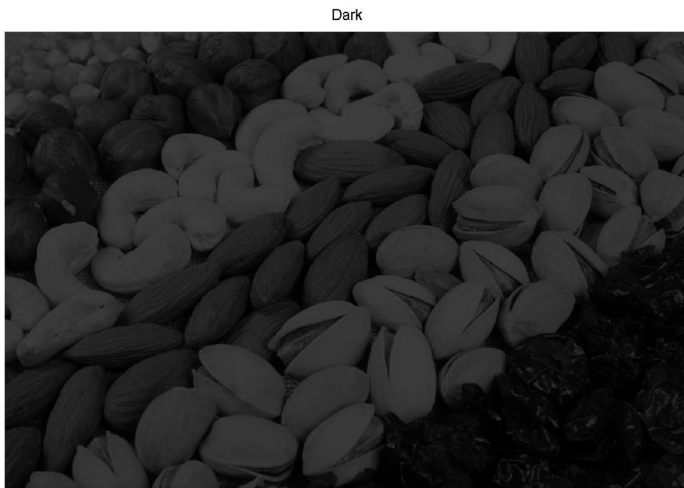
$$\left(\frac{x}{255}\right)^2 \times 255$$

Istogrammi

- L'**istogramma** di un'immagine digitale con livelli d'intensità $[0, L-1]$ è una funzione discreta
 - $h(r_k) = n_k$
- dove r_k è il k -esimo livello d'intensità ed n_k è il numero di pixel aventi tale intensità ($0 \leq k \leq L-1$)
- L'**istogramma normalizzato** è dato da
 - $h(r_k) = n_k / (MN)$
- la cui somma delle componenti vale 1
- $h(r_k)$ rappresenta la probabilità di occorrenza di r_k
- Poco costosi da calcolare, sono utilizzati da diverse tecniche di elaborazione delle immagini

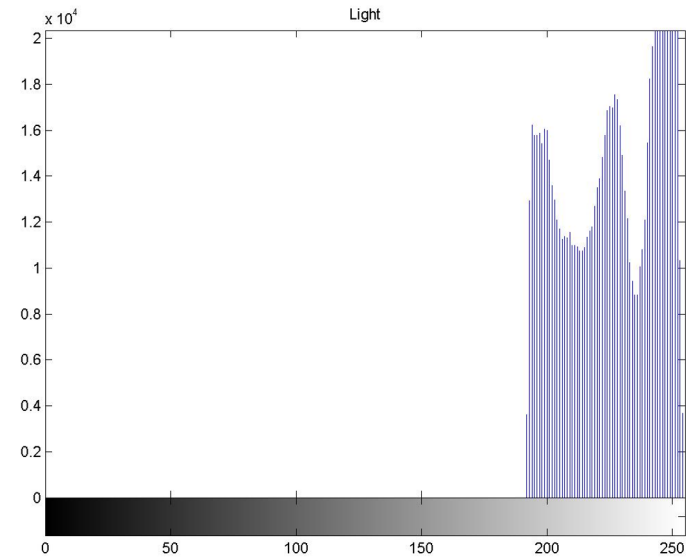
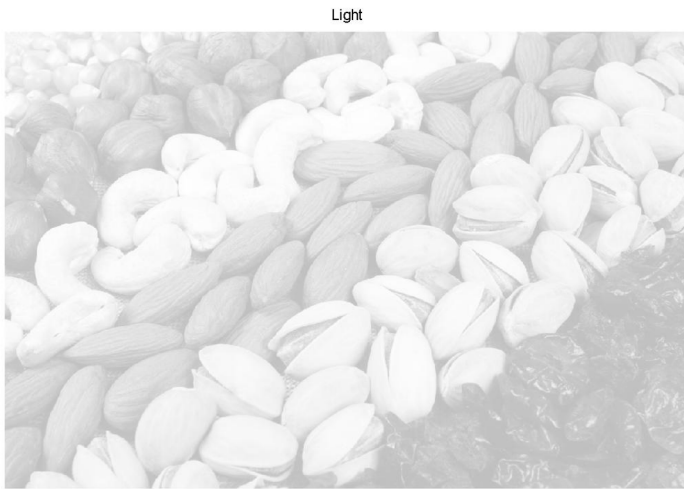
Istogrammi

Immagine scura



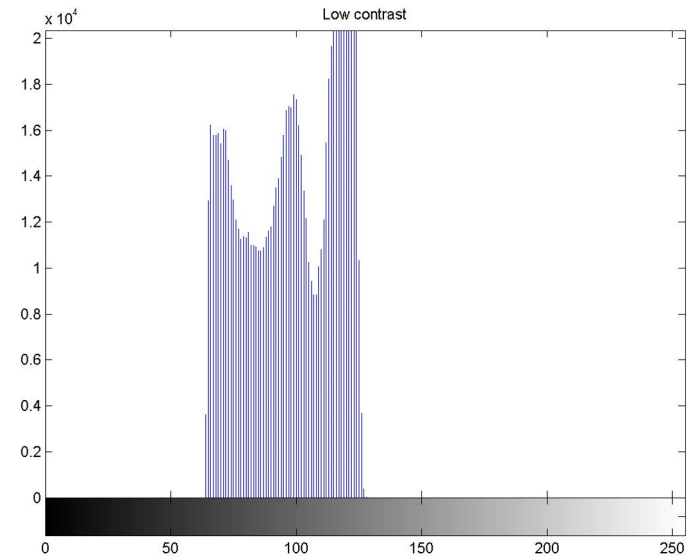
Istogrammi

Immagine chiara



Istogrammi

Basso contrasto: immagine sbiadita



Istogrammi

Alto contrasto (high dynamic range)

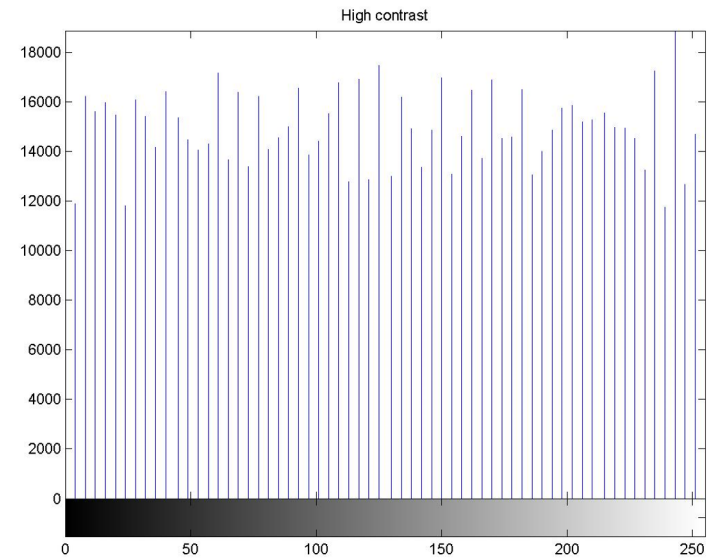


Image Enhancement: Equalizzazione dell'istogramma

- Funzione di trasformazione che intende produrre immagine di output con istogramma uniforme
- Si consideri la seguente trasformazione, con $p_r(r)$ la PDF (density funct.) dell'immagine input

$$s = T(r) = \int_0^r p_r(w) dw \quad \begin{array}{l} \text{CDF (distribution funct.),} \\ \text{funzione di ripartizione} \end{array}$$

- nel caso continuo vale che, la PDF dell'immagine output è uniforme

$$p_s(s) = \begin{cases} 1 & \text{for } 0 \leq s \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$s = T(r) = \int_0^r p_r(w) dw \longrightarrow$$

$$\frac{ds}{dr} = \frac{d}{dr} T(r) = p_r(r)$$



$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| \longrightarrow$$

$$p_s(s) = (r) \left| \frac{1}{p_r(r)} \right| = 1$$

Equalizzazione dell'istogramma

- Equalizzazione dell'istogramma:

$$\begin{aligned}s_k &= T(r_k) \\ &= \sum_{j=1}^k p_r(r_j) \\ &= \sum_{j=1}^k \frac{n_j}{n}\end{aligned}$$

- A causa della natura discreta delle variabili, l'istogramma di output non sarà uniforme
 - se si vuole approssimare la distribuzione uniforme occorre ridurre il numero di livelli in output

Adaptive Histogram Equalization

- L'equalizzazione si comporta bene quando la distribuzione delle intensità è simile su tutta l'immagine
- Adaptive Histogram Equalization (AHE)
 - Equalizzazione sul neighborhood di ogni pixel
- Contrast Limited AHE (CLAHE)
 - Evitare di amplificare troppo zone a intensità quasi costante
- Calcolo efficiente mediante interpolazione
 - Divide l'immagine in tiles il cui centro si ottiene per equalizzazione
 - Il valore dei restanti pixel si ottiene per interpolazione

Image adjustment in Python

- Demo Python

Esercizio

- Scegliete una vostra immagine
- Create un notebook e caricate l'immagine
- Eseguite tutte le operazioni della slide 40