

# Operazioni di Elaborazione delle Immagini Digitali

Corso di «Strumentazione Diagnostica per Immagini»

a.a. 2021 – 2022

Prof. Roberto Pirrone

# Sommario

- Criteri di valutazione della qualità di un'immagine
- Iistogramma di un'immagine digitale
- Operazioni puntuali ed elaborazione dell'istogramma
- Operazioni multi-immagine
- Operazioni geometriche
- Filtri

# Criteri di valutazione della qualità di un'immagine

- Rumore
  - Un'immagine digitale  $I$ , acquisita tramite un qualunque dispositivo di ripresa con funzione di trasferimento  $H$ , subisce delle alterazioni del segnale legate al processo stesso di acquisizione piuttosto che a problemi/guasti nel processo di digitalizzazione e storage.
  - Il *rumore* o *disturbo* rappresenta un segnale stocastico che si somma all'immagine acquisita in forma ideale e rappresenta l'insieme di tutte le alterazioni

$$I(i, j) = h * f(i, j) + \eta(i, j)$$

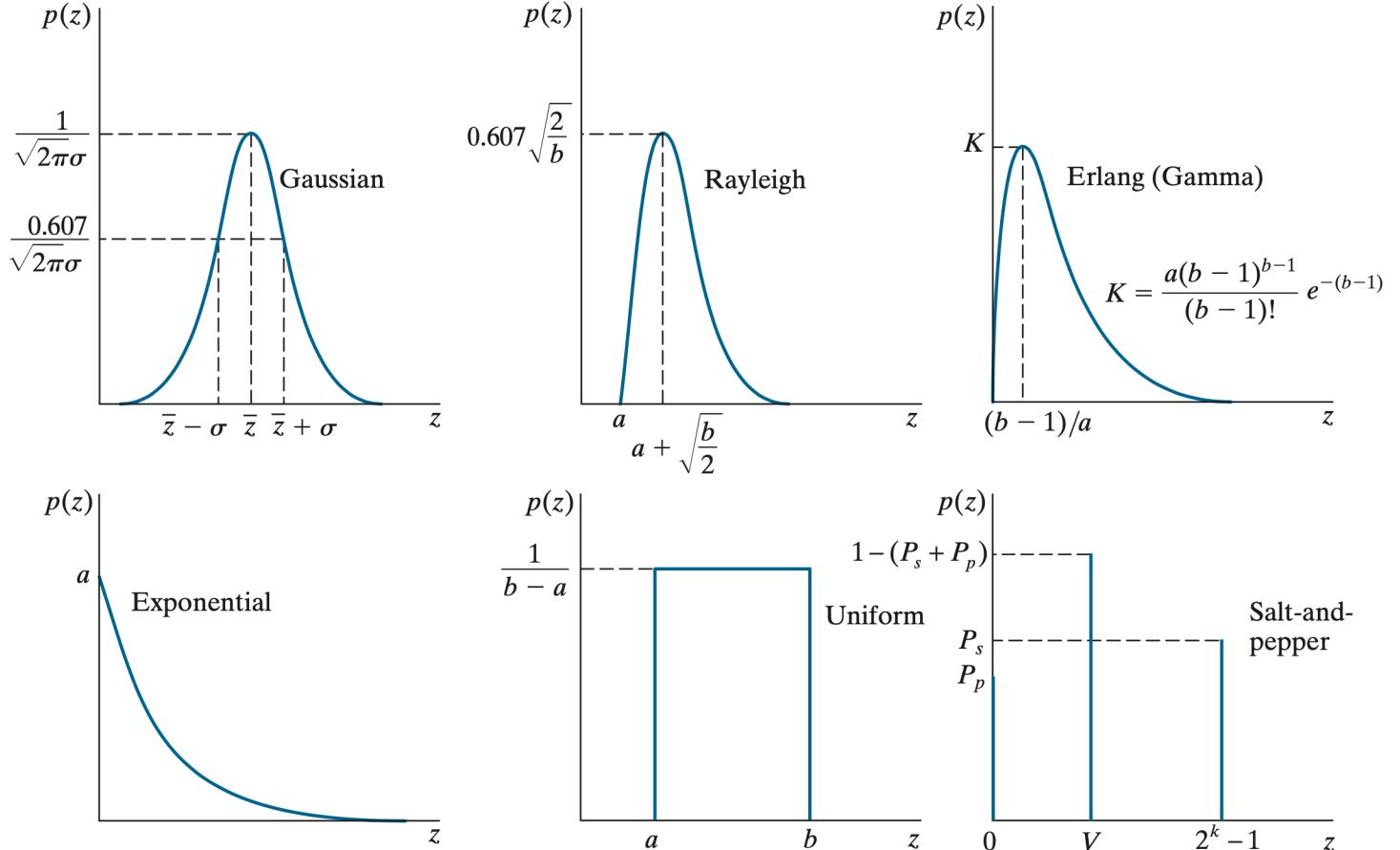
$$\mathcal{I} = H \cdot F + N$$

# Criteri di valutazione della qualità di un'immagine

- L'escursione dei livelli di intensità in un'immagine digitale è detta *gamma dinamica* o semplicemente *dinamica*
  - Ovviamente la dinamica dipende direttamente dal numero  $k$  di bit usati per la quantizzazione
- Il contrasto di un'immagine è definito come la differenza tra il massimo livello riproducibile e il minimo livello di segnale rilevabile non confuso con il rumore
- Il *Contrast-to-Noise Ratio* (CNR) è il rapporto tra queste due quantità ed è in genere la misura preferita per indicare globalmente la qualità di un'immagine digitale.
  - E' l'equivalente del *SNR* (Signal-to-Noise Ratio) per i segnali tempovarianti

# Criteri di valutazione della qualità di un'immagine

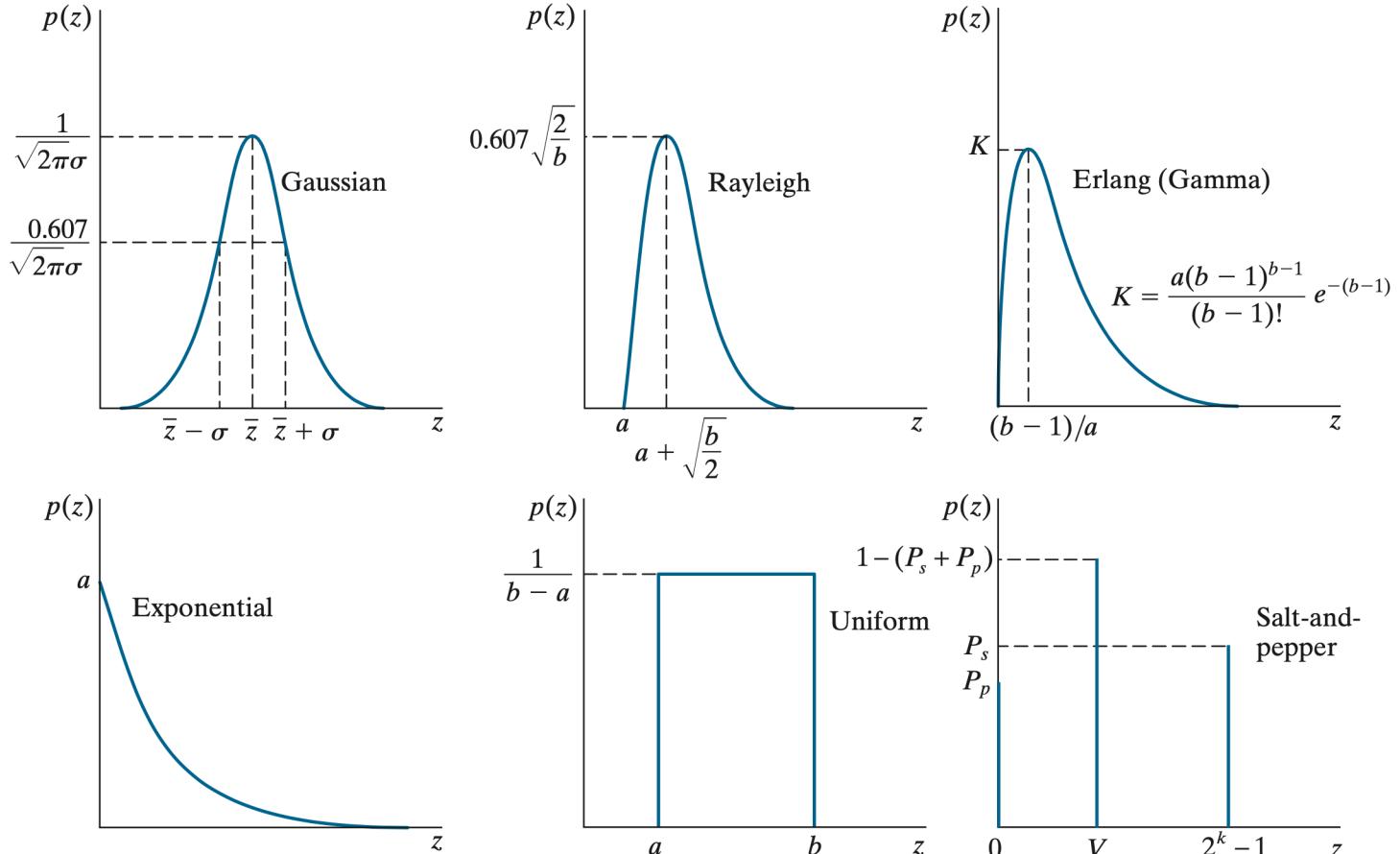
- Il rumore è una grandezza dipendente dalla frequenza e, in quanto segnale stocastico, può essere descritto da una distribuzione di probabilità
- *Rumore bianco*:  $N(k_x, k_y) = \text{costante}$



Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Criteri di valutazione della qualità di un'immagine

- Il *rumore sale e pepe* è un caso particolare: esso ha probabilità  $P_s$  di assumere il livello massimo della dinamica  $2^k - 1$ , ovvero probabilità  $P_p$  di valere 0.
- Qualunque altro valore  $V$  in  $[0, 2^k - 1]$  ha probabilità  $p = 1 - (P_s + P_p)$



Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Criteri di valutazione della qualità di un'immagine

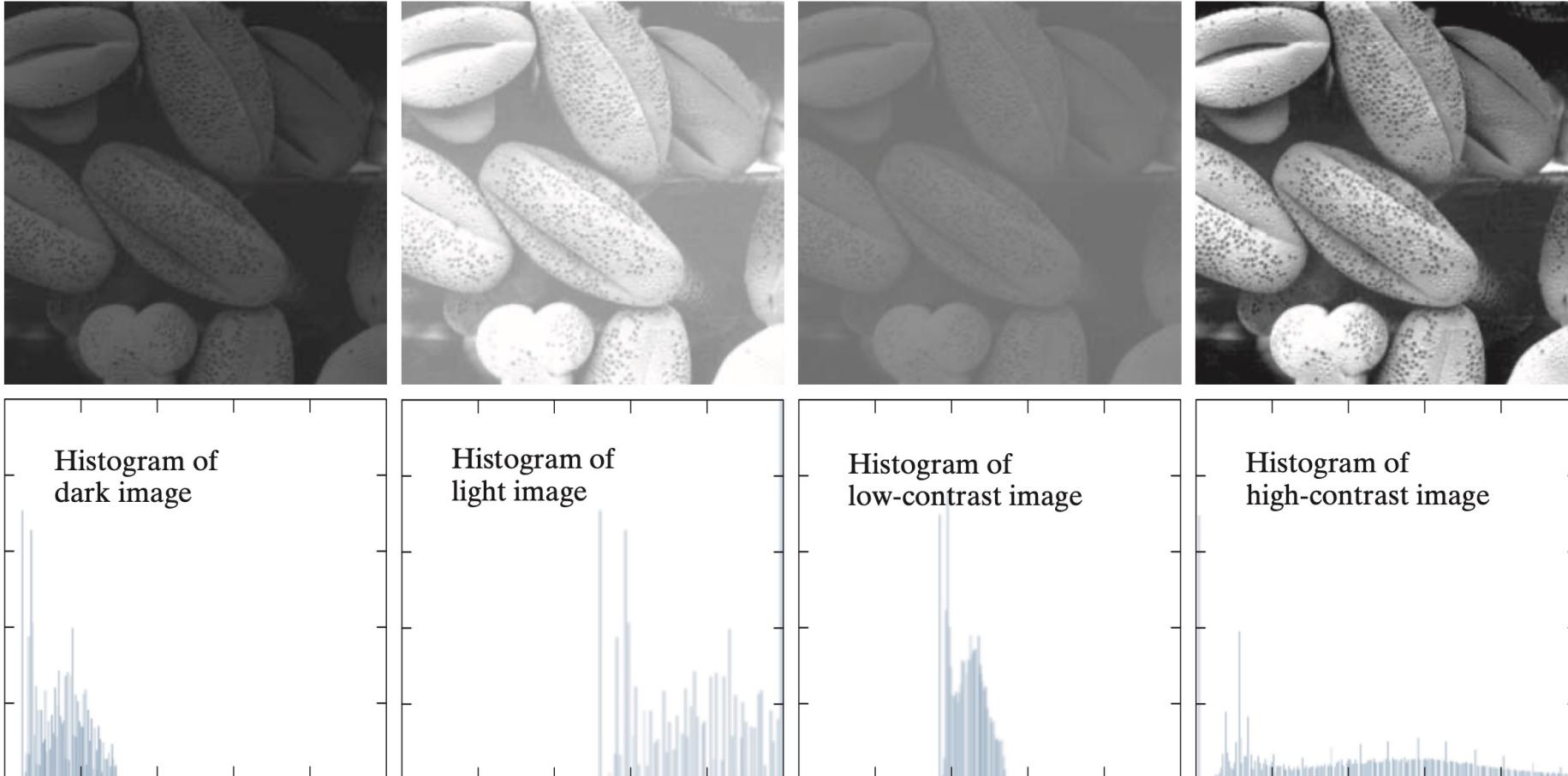
- Nella pratica una percentuale di pixel pari a  $P_s + P_p$  è affetta da rumore che si ripartirà in due aliquote di pixel bianchi (sale) o neri (pepe)
- Di solito i due valori sono uguali e quindi metà dei pixel alterati sarà bianca e metà nera.

# Istogramma di un'immagine digitale

- L'istogramma di un'immagine digitale di dimensioni  $N \times M$  è la distribuzione di probabilità dei livelli di intensità presenti al suo interno calcolata con approccio frequentista
- Detto  $n_k$  il numero di pixel che hanno il livello  $r_k$ , tale probabilità è espressa semplicemente come:

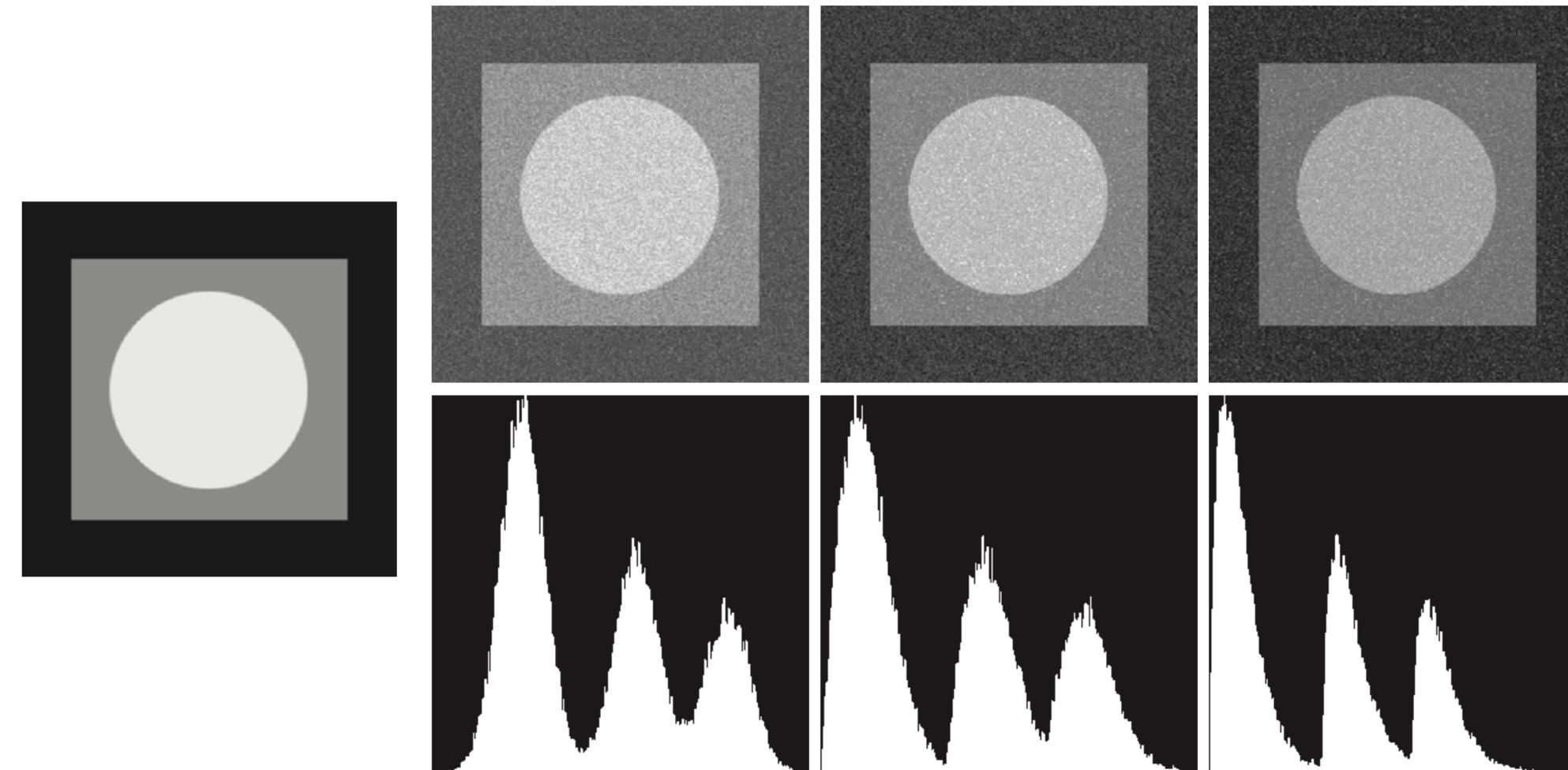
$$p(r_k) = n_k / NM$$

# Istogramma di un'immagine digitale



Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Istogramma di un'immagine digitale

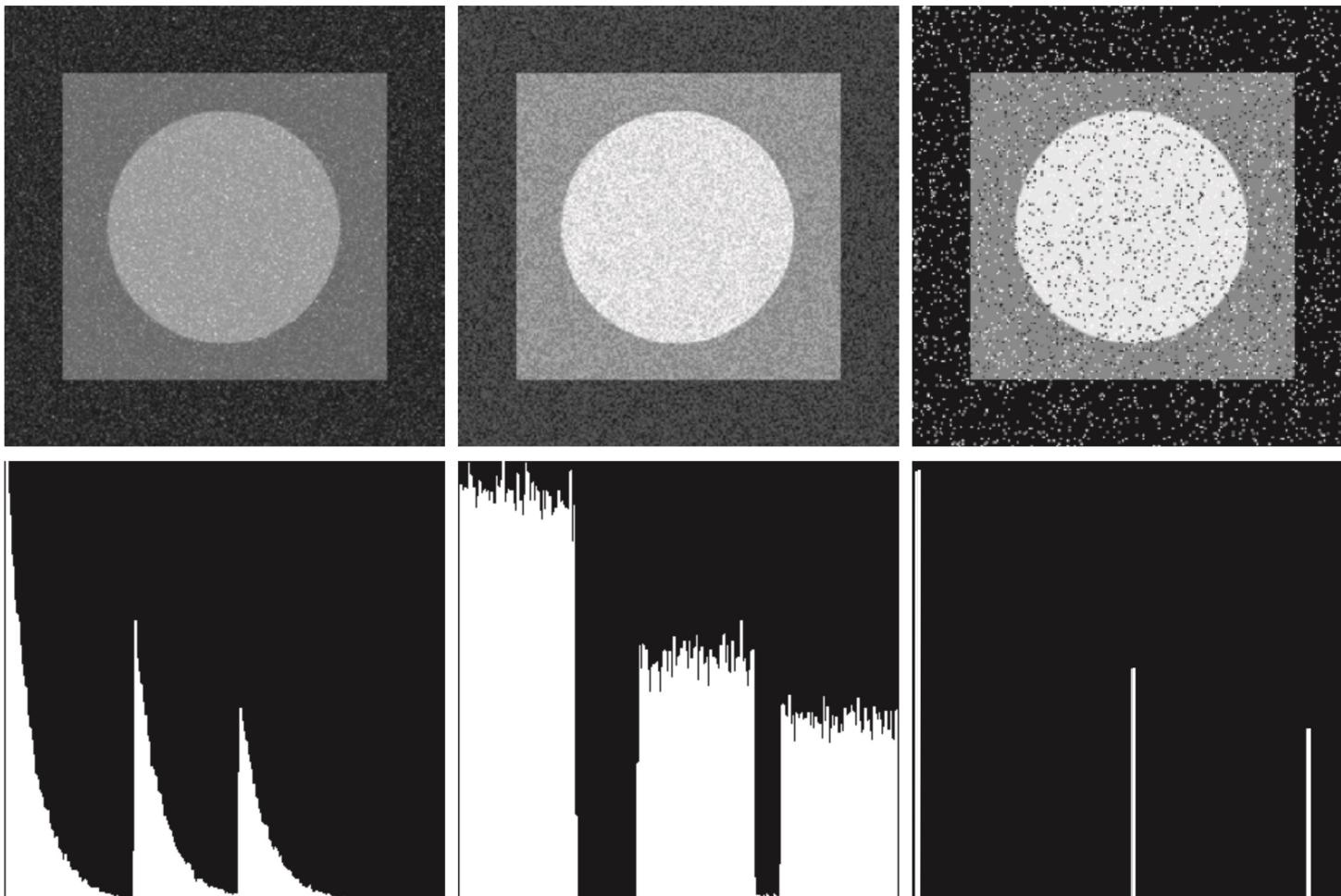
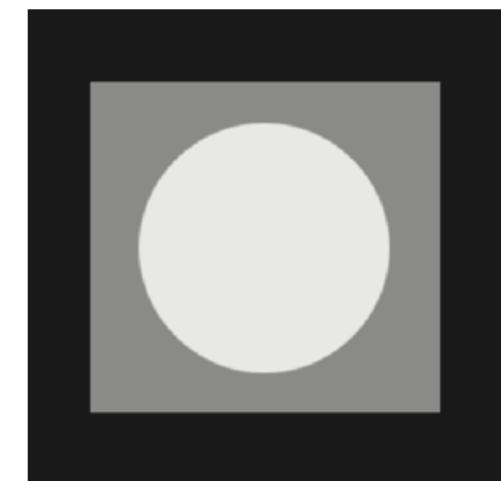


Rumore:

- Gaussiano
- Rayleigh
- Gamma

Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Istogramma di un'immagine digitale



Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Operazioni puntuale ed elaborazione dell'istogramma

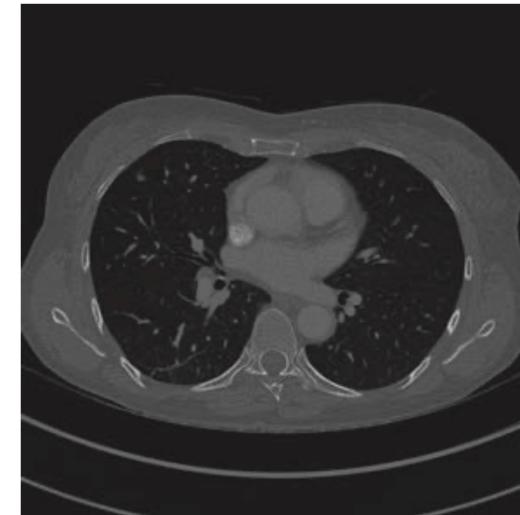
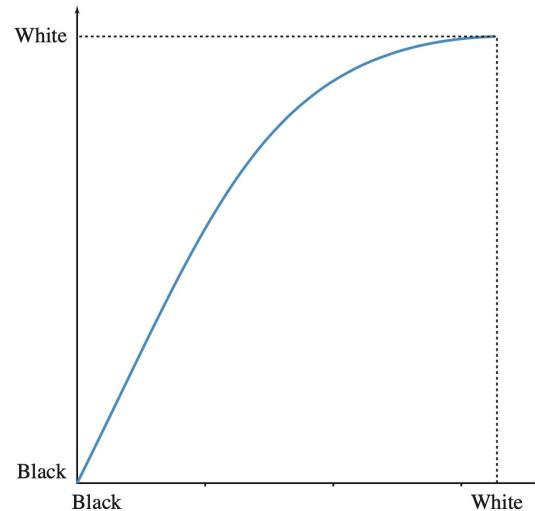
- L'ispezione diretta dell'istogramma ci fornisce spesso indicazioni per applicare delle trasformazioni ai livelli di grigio in modo da ottenere un'immagine di contrasto più elevato
- Questo tipo di trasformazioni sono dette *operazioni puntuali*

# Operazioni puntuali ed elaborazione dell'istogramma

- Una trasformazione puntuale è espressa semplicemente da una legge che mappa il valore di ogni pixel in un altro sulla base di una legge di corrispondenza  $T$  tra le gamme dinamiche di ingresso e di uscita

$$I'(i, j) = T[I(i, j)]$$

$$s = T(r), s, r \in [0, L-1]$$



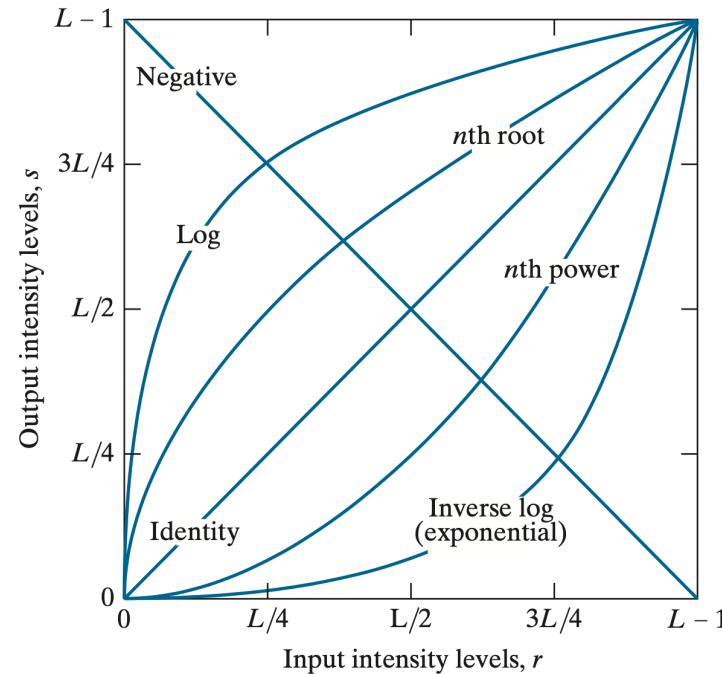
Fonte: Suetens «Fundamentals of Medical Imaging», 3rd ed.

# Operazioni puntuali ed elaborazione dell'istogramma

- Una trasformazione puntuale è espressa semplicemente da una legge che mappa il valore di ogni pixel in un altro sulla base di una legge di corrispondenza  $T$  tra le gamme dinamiche di ingresso e di uscita

$$I'(i, j) = T[I(i, j)]$$

$$s = T(r), s, r \in [0, L-1]$$

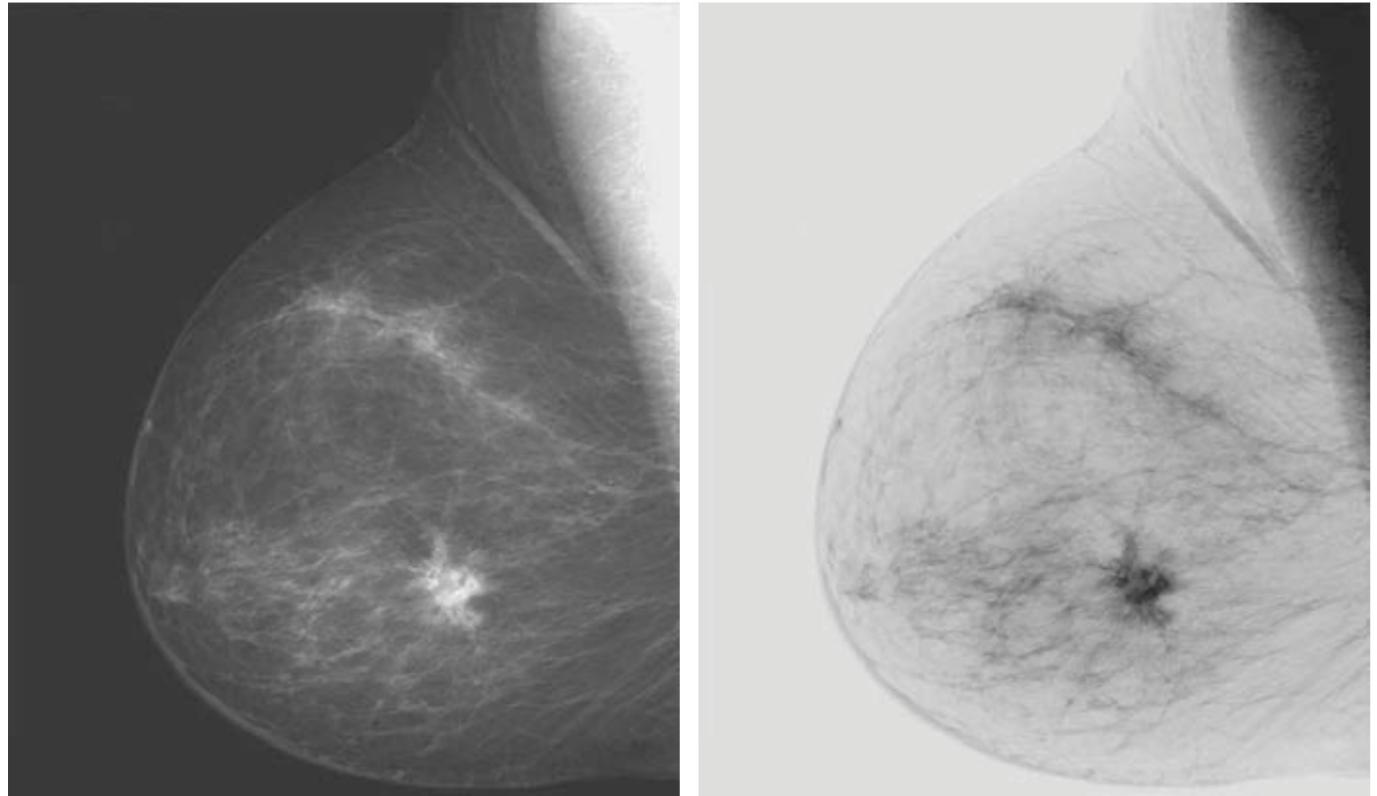


Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Operazioni puntuali ed elaborazione dell'istogramma

- Negativo

$$s = L - 1 - r, s, r \in [0, L-1]$$

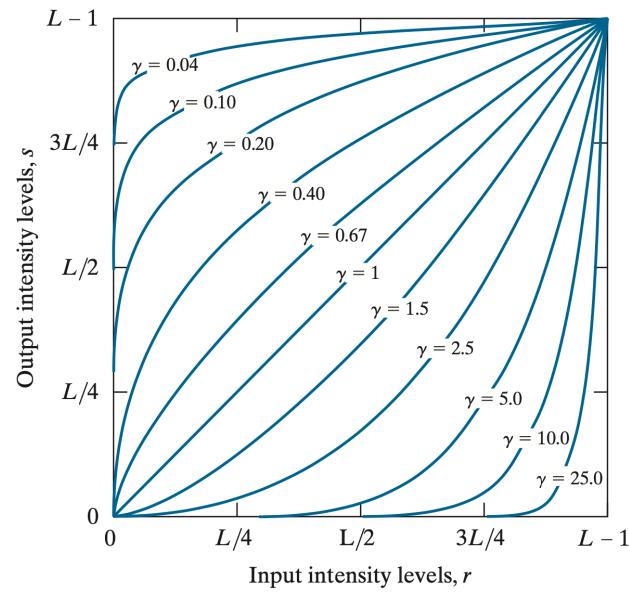


Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

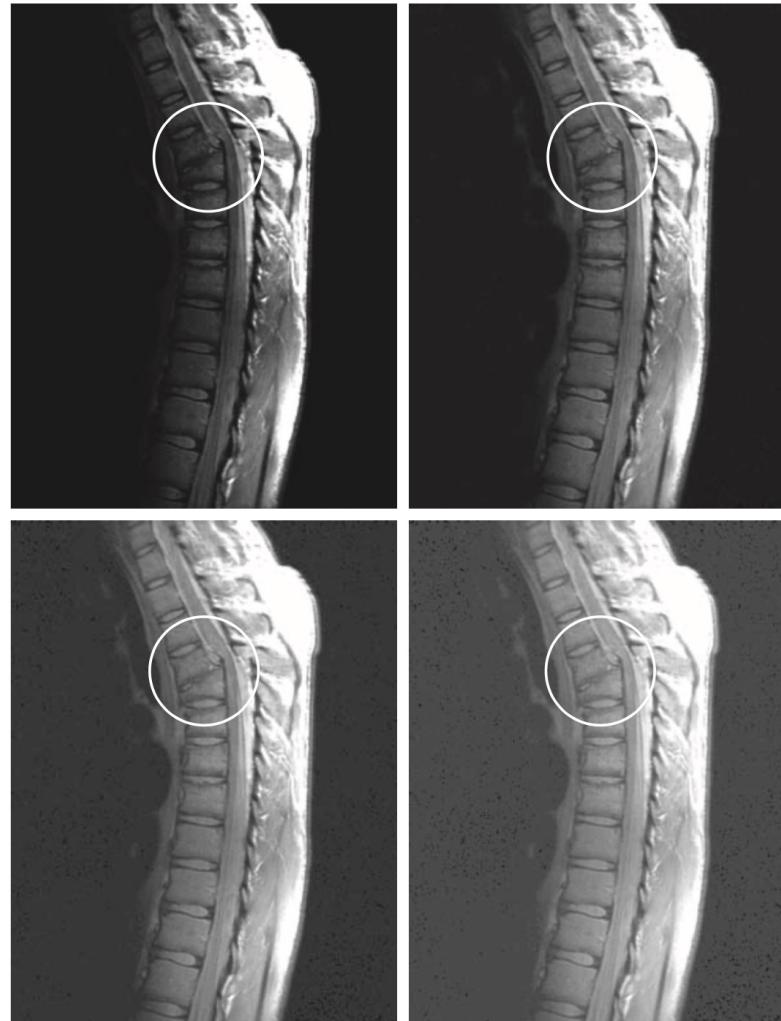
# Operazioni puntuali ed elaborazione dell'istogramma

- Correzione gamma

$$s = c \cdot r^\gamma, s, r \in [0, L-1], c = 1 \text{ (di norma)}$$



$$\gamma = 0.4$$



$$\gamma = 0.6$$

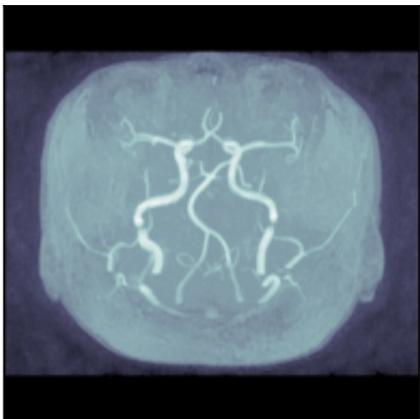
$$\gamma = 0.3$$

Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

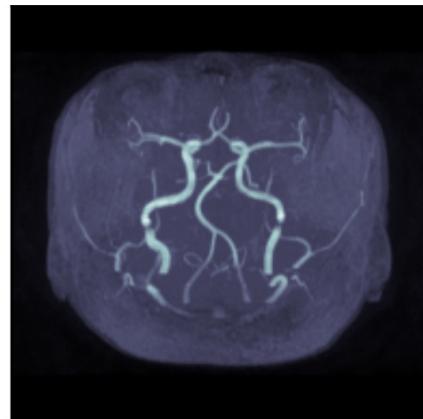
# Operazioni puntuali ed elaborazione dell'istogramma

- Correzione gamma

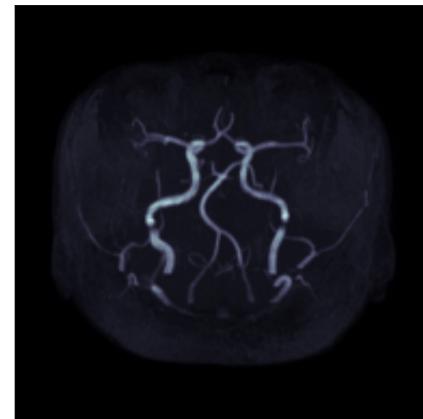
$$s = c \cdot r^\gamma, s, r \in [0, L-1], c = 1 \text{ (di norma)}$$



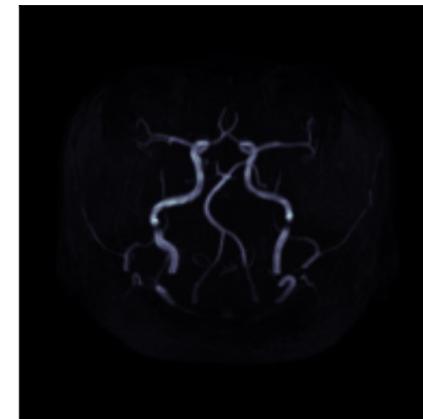
gamma = 0.2



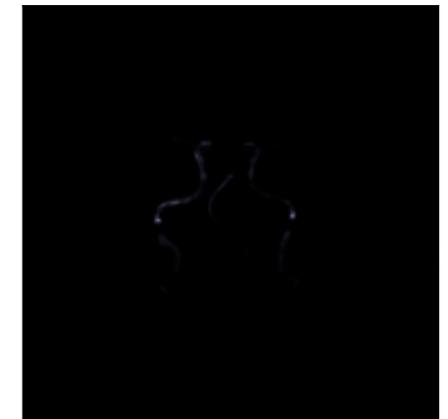
gamma = 0.5



gamma = 1.0



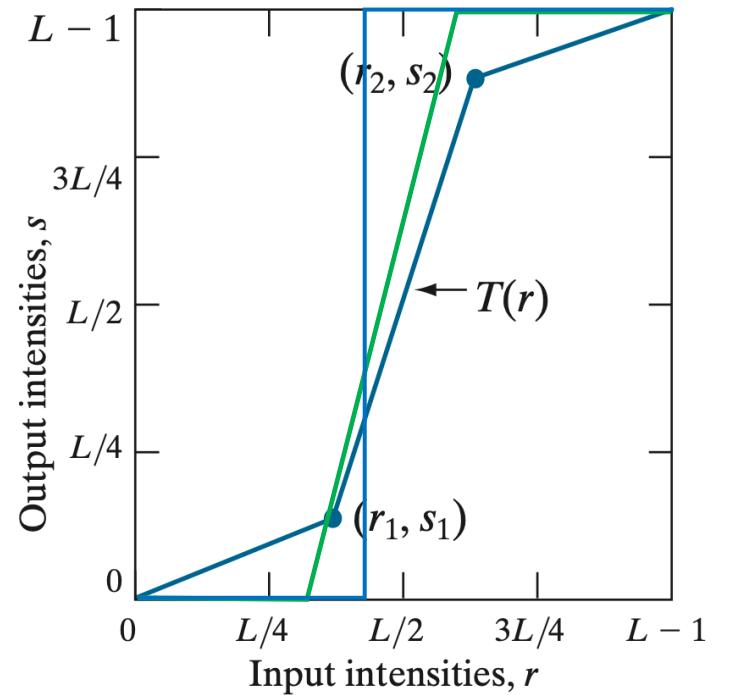
gamma = 1.5



gamma = 5.0

# Operazioni puntuali ed elaborazione dell'istogramma

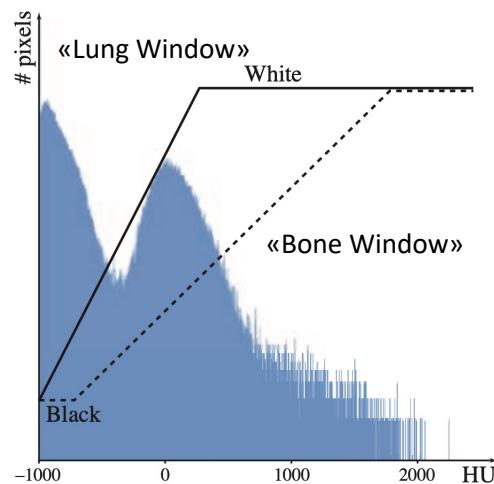
- Contrast stretching
  - Trasformazione lineare a tratti mirata a espandere la dinamica nella regione tra i punti  $(r_1, s_1)$  ed  $(r_2, s_2)$
  - Non è necessario che gli estremi siano in  $(0, 0)$  e  $(L-1, L-1)$
  - $s_1 = 0, s_2 = L-1 \rightarrow$  selezione di una «finestra» nella dinamica di ingresso
  - $s_1 = 0, s_2 = L-1, r_1 = r_2 \rightarrow$  sogliatura



Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Operazioni puntuali ed elaborazione dell'istogramma

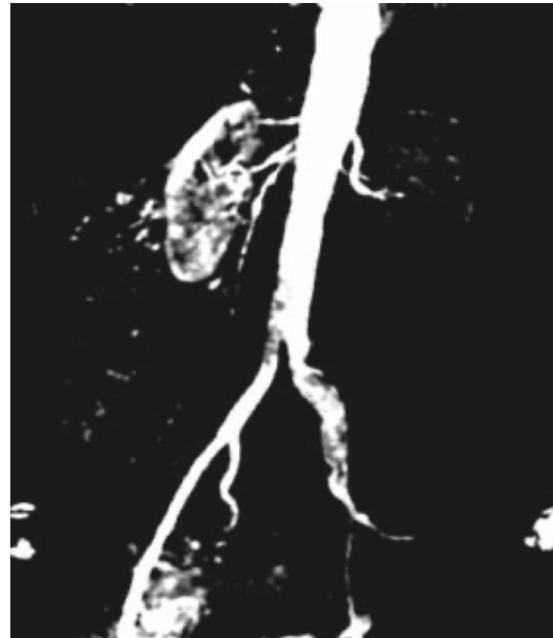
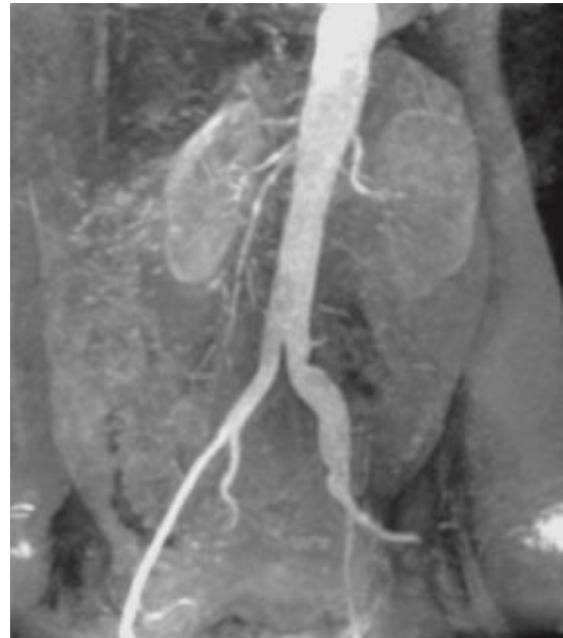
- Contrast stretching
  - Nella visualizzazione di immagini mediche a 12 bpp (*bit per pixel*) la «finestratura» è molto usata per selezionare le regioni della dinamica che corrispondono a particolari regioni anatomiche.



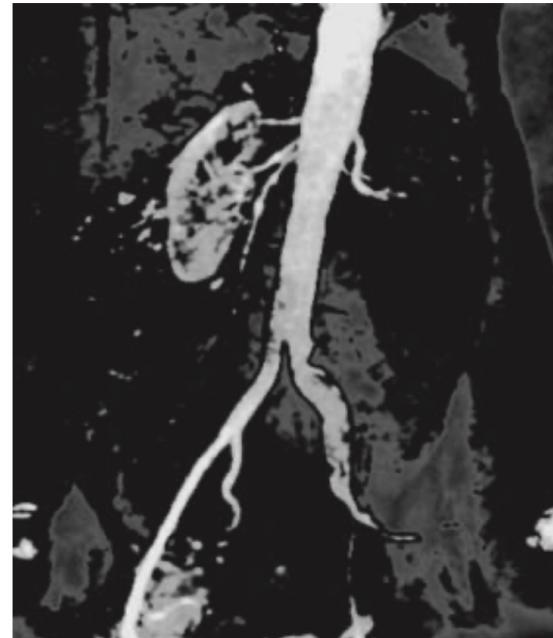
Fonte: Suetens «Fundamentals of Medical Imaging», 3rd ed.

# Operazioni puntuali ed elaborazione dell'istogramma

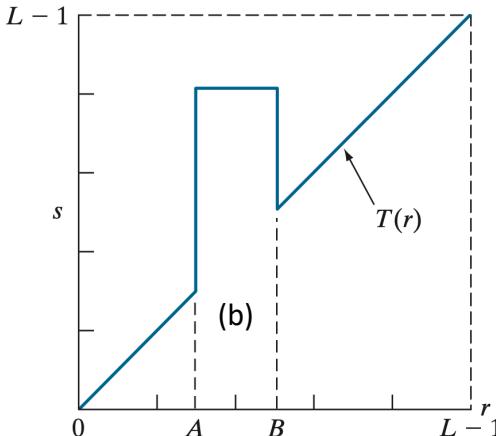
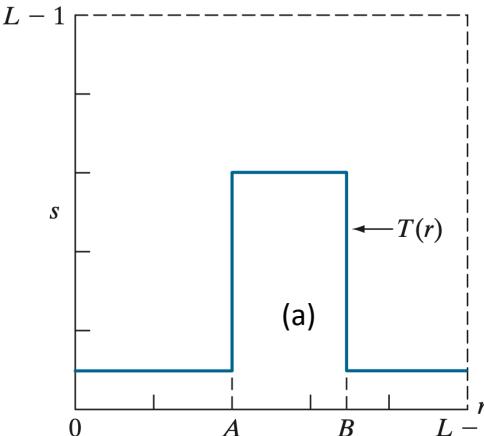
- Slicing dei livelli di grigio



Slicing (a)



Slicing (b)



# Equalizzazione dell'istogramma

- L'equalizzazione dell'istogramma tende determinare una trasformazione  $T$  tale che l'istogramma dell'immagine trasformata  $I'(i,j)$  sia costante su tutta la dinamica e pari a  $1/(L-1)$
- L'area sottesa dall'istogramma deve essere unitaria poiché lo abbiamo definito come una *funzione densità di probabilità* (PDF – Probability Distribution Function)  $p_r(r_k)$  e quindi tale area costituisce la *funzione di distribuzione cumulativa di probabilità* (CDF – Cumulative Distribution Function)

# Equalizzazione dell'istogramma

- Assumiamo una trasformazione

$s=T(r)$  *monotona crescente* di modo che si possa definire una inversa  $r=T^{-1}(s)$  che è una corrispondenza biunivoca

$$\begin{aligned} P_s(s_k) &\triangleq P(s \leq s_k) = P(T(r) \leq s_k) = \\ &= P(r \leq T^{-1}(s_k)) = P_r(r_k) \end{aligned}$$

$$p_s(s_k) \triangleq \frac{dP_r(r_k)}{ds} = p_r(r_k) \left| \frac{dr}{ds} \right|$$

- Vogliamo trovare il nuovo istogramma come una PDF  $p_s(s_k)$

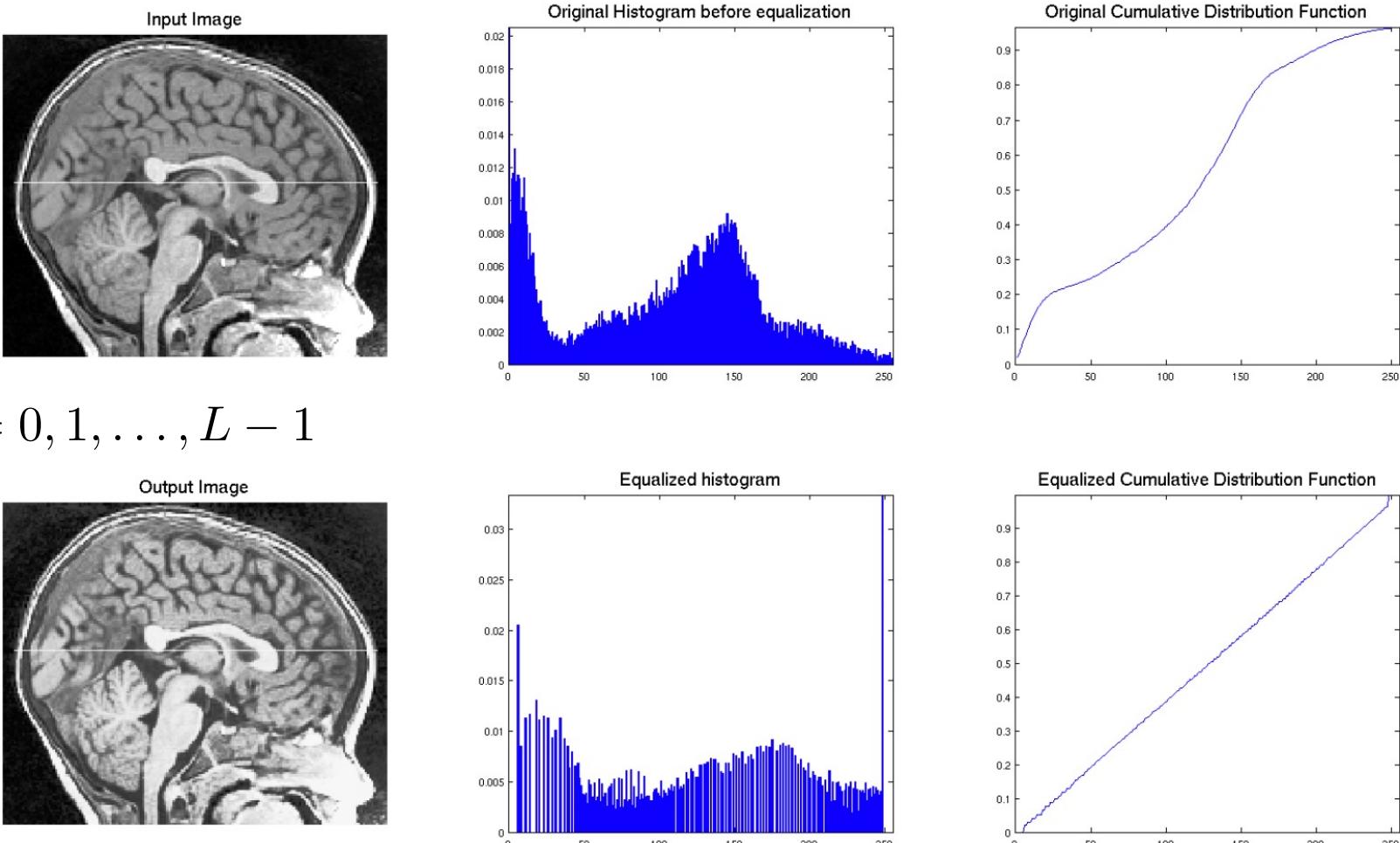
# Equalizzazione dell'istogramma

- La trasformazione di equalizzazione dell'istogramma si ottiene come riportato
- L'istogramma risultante è quello richiesto

$$\begin{aligned}s &= T(r) = (L - 1) \int_0^r p_r(w) dw \\p_s(s_k) &= p_r(r_k) \left| \frac{dr}{ds} \right| = p_r(r_k) \cdot 1 \left| \frac{dT(r)}{dr} \right|_{r_k} = \\&= p_r(r_k) \cdot 1 \left/ (L - 1) \frac{d}{dr} \left[ \int_0^{r_k} p_r(w) dw \right] \right. = \\&= p_r(r_k) \frac{1}{(L - 1)p_r(r_k)} \equiv \frac{1}{L - 1}\end{aligned}$$

# Equalizzazione dell'istogramma

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j), \quad k = 0, 1, \dots, L - 1$$



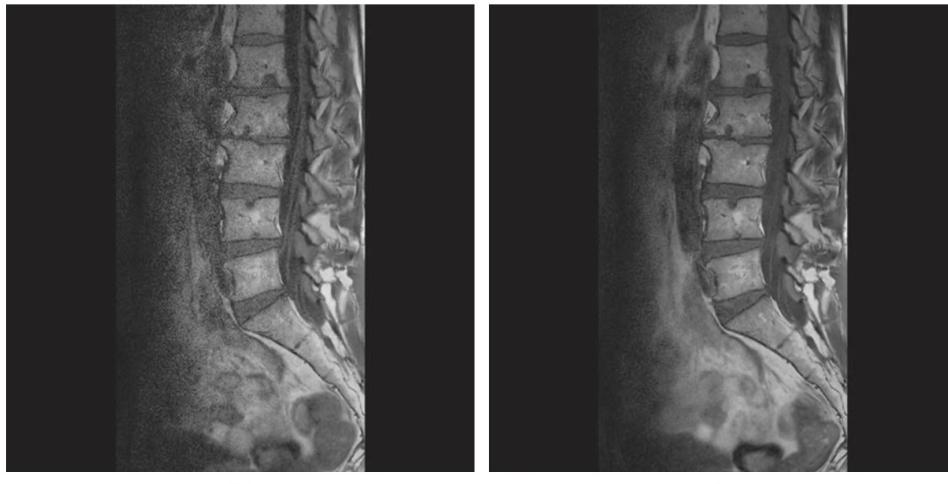
# Operazioni multi-immagine

- Si tratta di operazioni applicate pixel a pixel su coppie di immagini
  - $I_+(i,j) = I_1(i,j) + I_2(i,j)$
  - $I_-(i,j) = I_1(i,j) - I_2(i,j)$
  - $I_{avg}(i,j) = 1/n [I_1(i,j) + I_2(i,j) + \dots + I_n(i,j)]$

# Operazioni multi-immagine

- Si tratta di operazioni applicate pixel a pixel su coppie di immagini

- $I_+(i,j) = I_1(i,j) + I_2(i,j)$
- $I_-(i,j) = I_1(i,j) - I_2(i,j)$
- $I_{avg}(i,j) = 1/n [I_1(i,j) + I_2(i,j) + \dots + I_n(i,j)]$



(b) (d): media di 5 sequenze pesate in  $T_1$  (a) e  $T_2$  (c), entrambe TurboSE



Fonte: Suetens «Fundamentals of Medical Imaging», 3rd ed.

# Operazioni multi-immagine

- Si tratta di operazioni applicate pixel a pixel su coppie di immagini

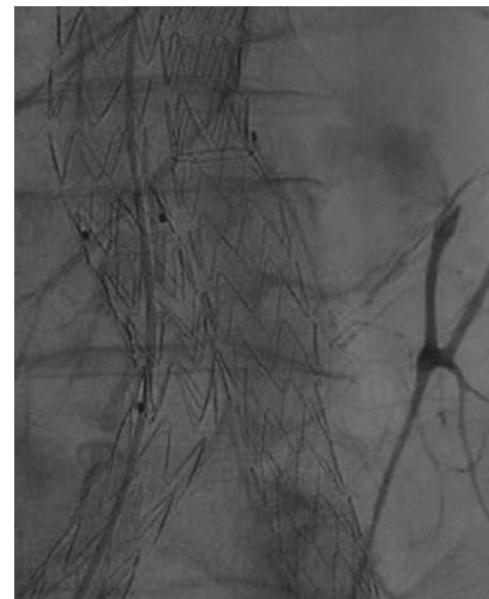
$$\bullet \quad I_+(i,j) = I_1(i,j) + I_2(i,j)$$

- $I_{\text{-}}(i,j) = I_1(i,j) - I_2(i,j)$

- $I_{avg}(i,j) = 1/n [I_1(i,j) + I_2(i,j) + \dots + I_n(i,j)]$



(a)



(b)



(c)

- (b): contrast stretching di (a)
- (c): contrast stretching di (b) – (a)

Fonte: Suetens «Fundamentals of Medical Imaging», 3rd ed.

# Operazioni geometriche

- Le operazioni geometriche fanno riferimento alle cosiddette trasformazioni *affini* ovvero a quelle *proiettive*
  - Trasformazione affine  $A(p) \rightarrow (p')$ : trasforma un punto  $p$  nello spazio  $\mathbb{R}^2$  o  $\mathbb{R}^3$  in un punto  $p'$  nello stesso spazio, mantenendo le relazioni di parallelismo tra le rette
  - Trasformazione proiettiva  $P(p) \rightarrow (p')$ : trasforma un punto  $p \in \mathbb{R}^3$  in un punto  $p' \in \mathbb{R}^2$  che è l'intersezione tra un *raggio proiettore* che passa per  $p$  e per un punto fisso detto *centro di proiezione* o *fuoco*  $F$ , con un *piano di proiezione*  $\pi$ .
    - In generale, non c'è un vincolo nella mutua posizione di  $F$  e  $\pi$  nello spazio

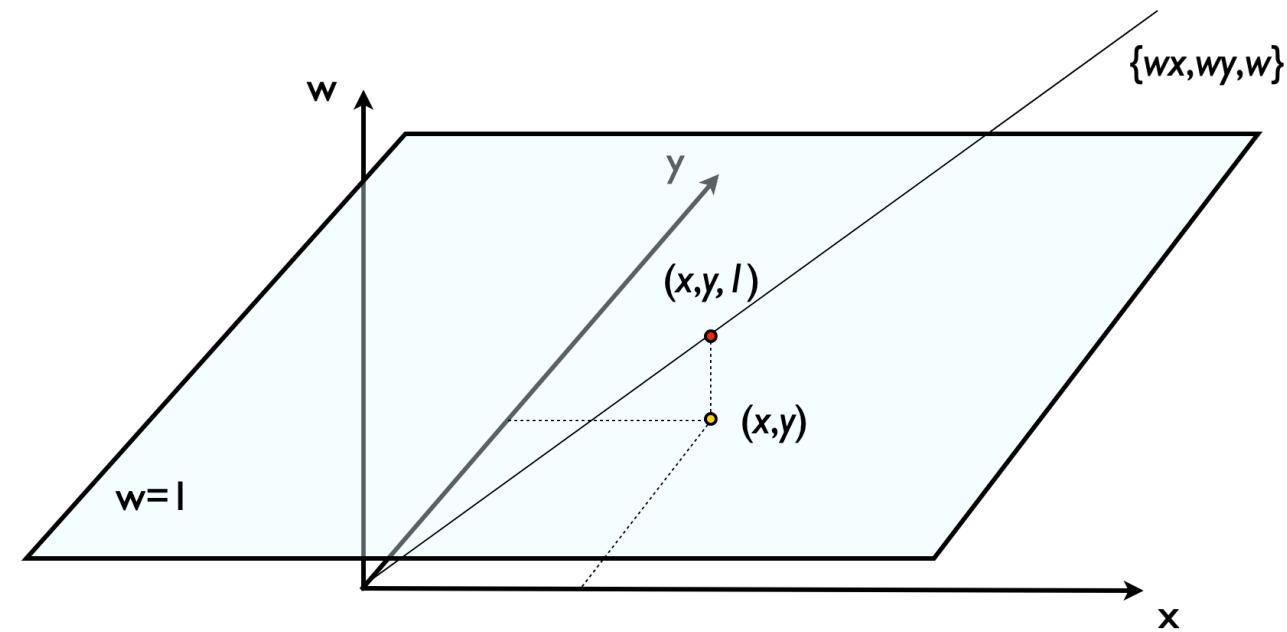
# Operazioni geometriche

- Entrambe le tipologie di trasformazioni vengono calcolate attraverso *operazioni di moltiplicazione matrice-vettore* anche se non sono definite nativamente come tali
- Di conseguenza a ogni trasformazione corrisponde una matrice i cui elementi hanno una disposizione particolare.
- Da un punto di vista del calcolo questo risultato si ottiene tramite la rappresentazione dei punti in *coordinate omogenee*

# Operazioni geometriche

- Coordinate omogenee

- Aggiungiamo una terza (quarta) coordinata  $w$  a  $\mathbb{R}^2$  ( $\mathbb{R}^3$ )
- *Proiettiamo* dall'origine degli assi un raggio di equazione  $\{wx, wy, w\}$
- Usiamo il piano  $w = 1$  come piano di proiezione e otteniamo il punto di coordinate  $(x, y, 1)$  che è il *punto omogeneo a  $(x, y)$*



# Operazioni geometriche

- Trasformazioni affini
  - Una trasformazione affine è definita come
    - Traslazione
    - Rotazione
    - Trasformazione di scala
    - Shearing
    - Riflessione
    - ...

$$\left\{ \begin{array}{l} x^* = ax + by + l \\ y^* = cx + dy + m \end{array} \right.$$
$$\mathbf{T} \equiv \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$
$$\mathbf{L} \equiv \begin{bmatrix} m \\ n \end{bmatrix}$$
$$\vec{p}^* = \mathbf{T} \cdot \vec{p} + \mathbf{L}$$

# Operazioni geometriche

- Trasformazioni affini

- Passando alle coordinate omogenee si ottiene
- Una trasformazione affine 2D generica richiede la conoscenza di *sei parametri* e quindi la *corrispondenza nota tra tre punti*

$$\left\{ \begin{array}{l} x^* = ax + by + l \\ y^* = cx + dy + m \\ 1 = 1 \end{array} \right.$$
$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & l \\ c & d & m \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Componente di rotazione e scaling

Componente di traslazione

# Operazioni geometriche

- Trasformazioni affini

- In  $\mathbb{R}^3$  si ha

- Trasformazione generica:

*12 parametri  
ovvero 4 punti*

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

# Operazioni geometriche

- Trasformazioni affini
  - Traslazione

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$\left\{ \begin{array}{l} x^* = x + t_x \\ y^* = y + t_y \\ 1 = 1 \end{array} \right.$$

# Operazioni geometriche

- Trasformazioni affini

- Rotazione positiva di  $\theta$  in senso antiorario rispetto all'asse x

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\left\{ \begin{array}{l} x^* = x \cos(\theta) - y \sin(\theta) \\ y^* = y \sin(\theta) + y \cos(\theta) \\ 1 = 1 \end{array} \right.$$

# Operazioni geometriche

- Trasformazioni affini
  - Scalatura in  $x$  e  $y$

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$\left\{ \begin{array}{l} x^* = s_x x \\ y^* = s_y y \\ 1 = 1 \end{array} \right.$$

# Operazioni geometriche

- Trasformazioni affini
  - Shearing (scorrimento)

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \alpha & 0 \\ \beta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$\left\{ \begin{array}{l} x^* = x + \alpha y \\ y^* = \beta x + y \\ 1 = 1 \end{array} \right.$$

# Operazioni geometriche

- Trasformazioni affini
  - Proprietà delle matrici di trasformazione:

- La trasformazione di segno opposto a  $M$  si ottiene come  $M^{-1}$
- Le trasformazioni si compongono come prodotto di matrici e possono essere calcolate *a blocchi*

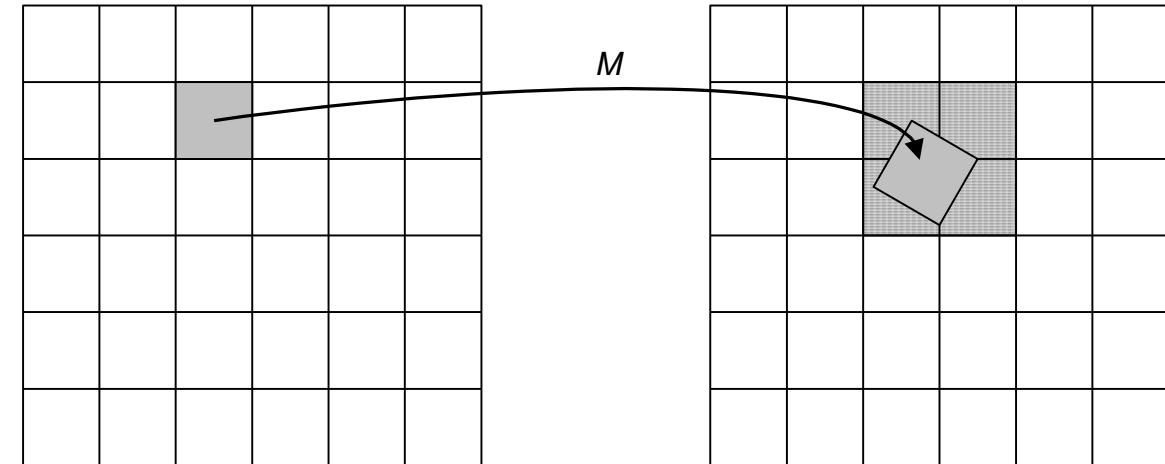
$$\mathbf{T} \cdot \mathbf{R} \cdot \mathbf{S} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x \cos(\theta) & -s_y \sin(\theta) & t_x \\ s_x \sin(\theta) & s_y \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

# Operazioni geometriche

- Trasformazioni affini
  - Proprietà delle matrici di trasformazione:
    - La *componente rotazionale* 2x2 di una trasformazione affine è una matrice ortogonale e la sua trasposta esprime la rotazione inversa (vale anche in 3D)
    - Data una trasformazione  $M$ , la sua inversa  $M^{-1}$  esprime *la trasformazione di cambio di coordinate con lo stesso segno di  $M$* 
      - Es. il cambio di coordinate  $(t_x, t_y)$  cioè il nuovo sistema con origine traslata di  $(t_x, t_y)$  esprime le coordinate dei punti come  $(x - t_x, y - t_y)$

# Operazioni geometriche

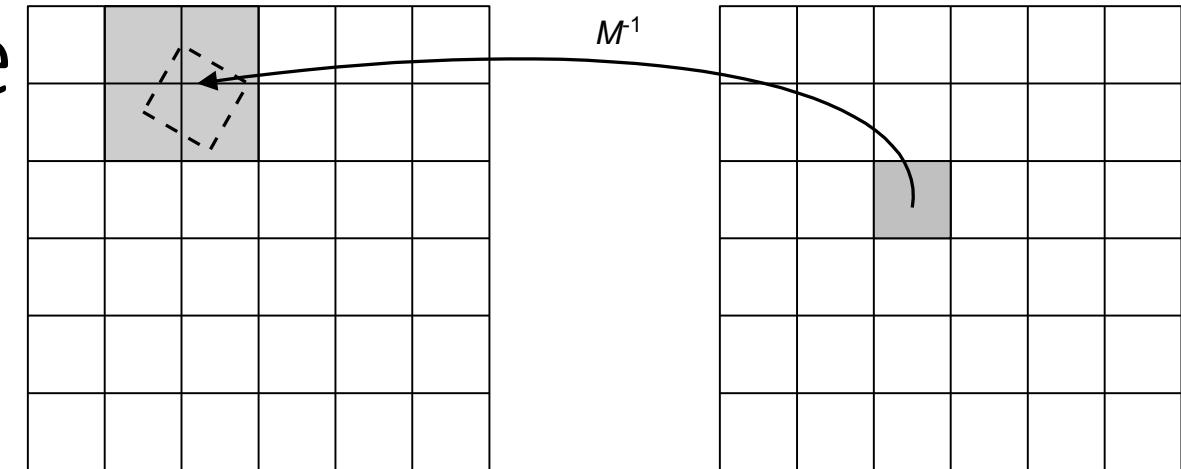
- Trasformazioni affini
  - Mapping inverso e interpolazione



- Applicare una trasformazione  $M$  a un'immagine  $I(i,j)$  *discreta* genera un'immagine  $I'(i',j')$  tale per cui ogni pixel si mappa in una posizione  $(i',j', 1)^T = M (i,j, 1)^T$
- Questo *non garantisce* che le posizioni  $(i',j')$  *coincidano con i centri dei pixel dell'immagine* ovvero che  $(i',j')$  *siano interi e, soprattutto, disposti sempre su centri di pixel adiacenti*

# Operazioni geometriche

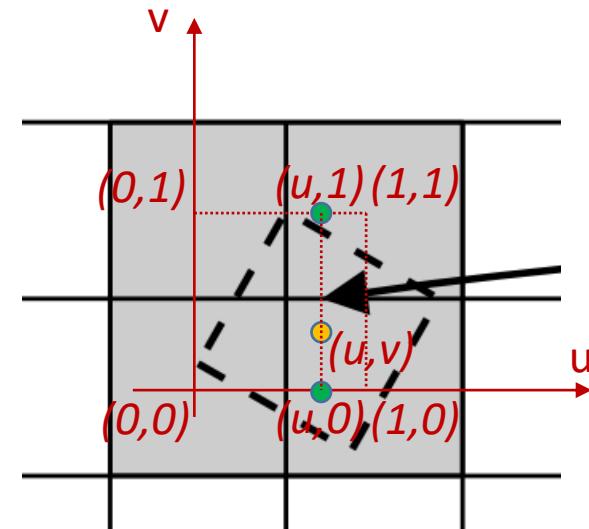
- Trasformazioni affini
  - Mapping inverso e interpolazione



- Questo problema si affronta utilizzando il cosiddetto *mapping inverso* ovvero  $M^{-1}$ , per stabilire quali sono i pixel in  $I$  più vicini a  $M^{-1}(i',j',1)^\top$
- Il valore  $I'(i',j')$  viene ottenuto per *interpolazione* tipicamente con *legge bilineare o con legge cubica*

# Operazioni geometriche

- Trasformazioni affini
  - Interpolazione bilineare
    - Il valore di  $I'(i', j')$  viene calcolato a partire dai quattro pixel più vicini in  $I$ , considerati in un sistema di coordinate locali  $(u, v) \in [0, 1]$ .
    - Il generico valore  $I(u, v)$  si ottiene interpolando linearmente dapprima  $I(u, 0)$  tra  $I(0,0)$  e  $I(1,0)$  nonché  $I(u, 1)$  tra  $I(0,1)$  e  $I(1,1)$
    - Successivamente si interpola linearmente  $I(u, v)$  tra  $I(u, 0)$  e  $I(u, 1)$



# Operazioni geometriche

- Trasformazioni affini

- Interpolazione bilineare

- Il valore interpolato di  $I(u, v)$  segue l'equazione di un **paraboloide iperbolico**

$$I(u, 0) = I(0, 0) + u[I(1, 0) - I(0, 0)]$$

$$I(u, 1) = I(0, 1) + u[I(1, 1) - I(0, 1)]$$

$$I(u, v) = I(u, 0) + v[I(u, 1) - I(u, 0)] =$$

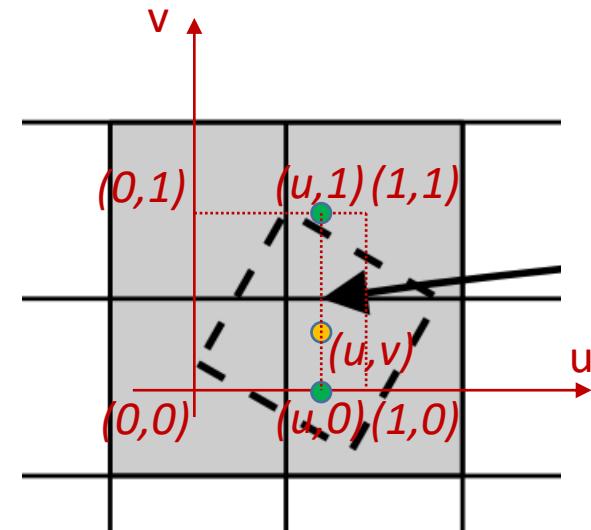
$$= I(0, 0) + u[I(1, 0) - I(0, 0)] +$$

$$+ v\{I(0, 1) + u[I(1, 1) - I(0, 1)] -$$

$$- I(0, 0) - u[I(1, 0) - I(0, 0)]\} =$$

$$= I(0, 0) + u[I(1, 0) - I(0, 0)] + v[I(0, 1) - I(0, 0)] +$$

$$+ uv[I(0, 0) + I(1, 1) - I(0, 1) - I(1, 0)]$$



# Operazioni geometriche

- Trasformazioni affini

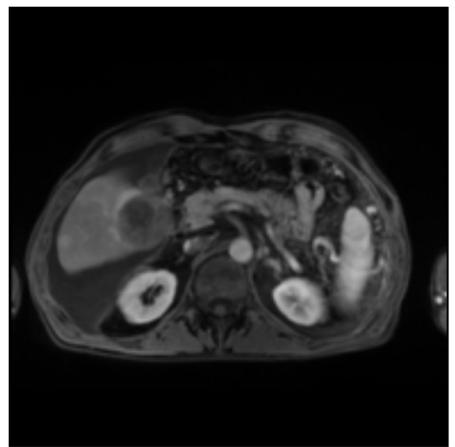
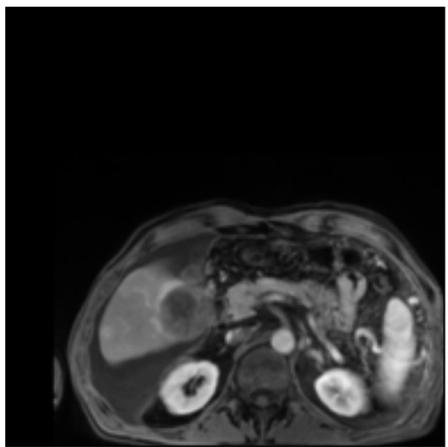
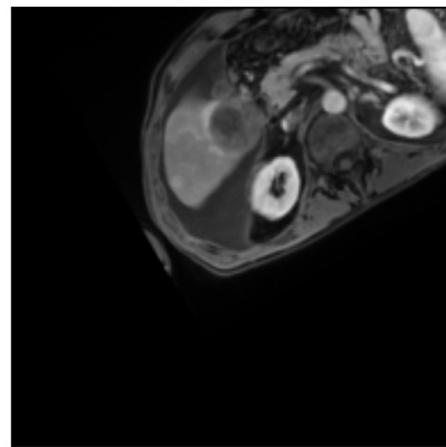


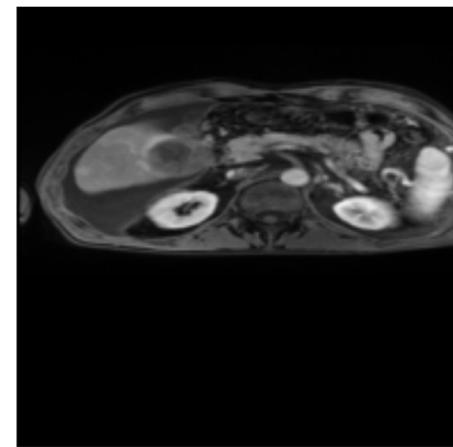
immagine originale



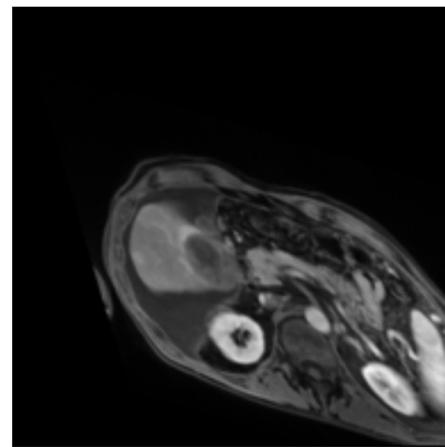
traslazione (100,50)



rotazione 30°



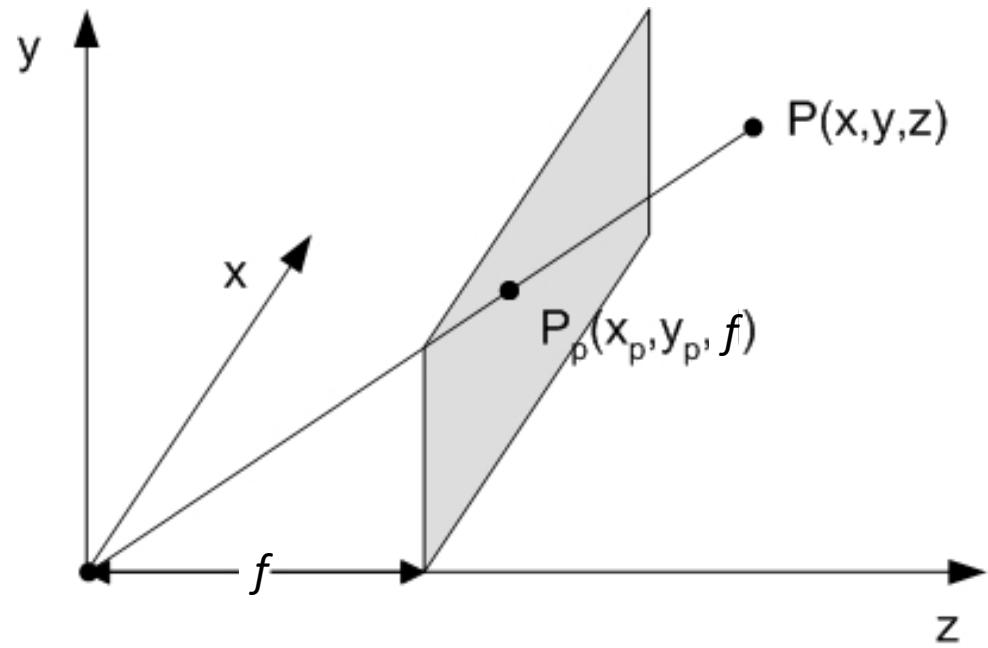
scalatura (0.7, 1.2)



shear (0.3, 0.3)

# Operazioni geometriche

- Trasformazioni proiettive
  - Le trasformazioni proiettive sono usate per proiettare un volume di dati 3D su un piano ed ottenere quindi un'unica immagine
  - La proiezione, in questo caso, ha una geometria semplice: il fuoco  $F$  si trova nell'origine e il piano di proiezione è parallelo al piano  $(x, y)$  a distanza  $f$  per cui ha equazione  $z = f$

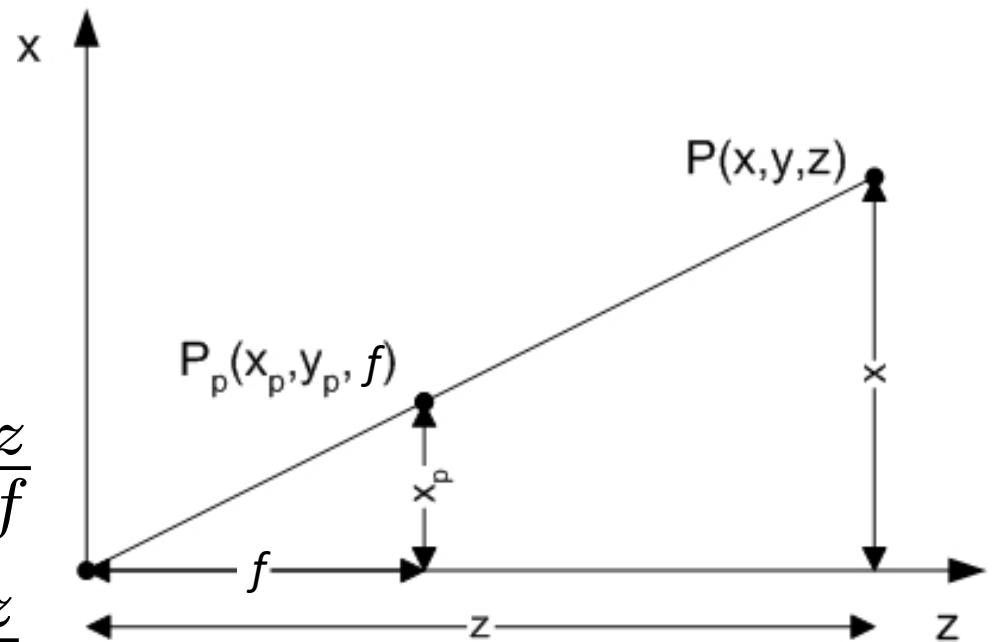


# Operazioni geometriche

- Trasformazioni proiettive

- Anche la proiezione prospettica può essere rappresentata in coordinate omogenee a *meno della divisione esplicita per la coordinata omogenea*

$$\left\{ \begin{array}{l} x_p/x = f/z \\ y_p/y = f/z \end{array} \right. ; \quad \left\{ \begin{array}{l} x_p = x \Big/ \frac{z}{f} \\ y_p = y \Big/ \frac{z}{f} \end{array} \right.$$

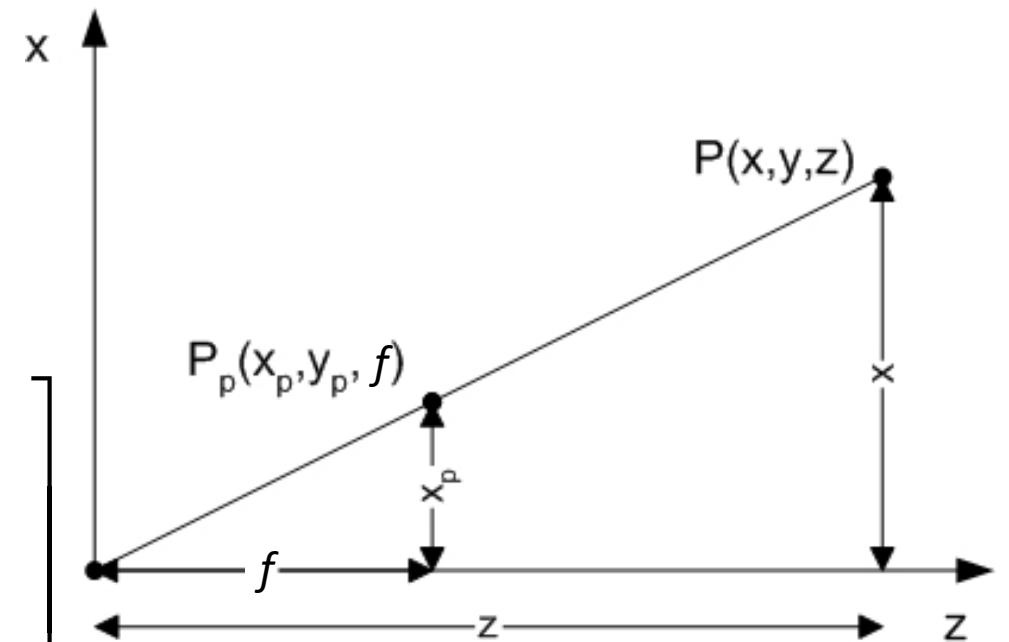


# Operazioni geometriche

- Trasformazioni proiettive

- Anche la proiezione prospettica può essere rappresentata in coordinate omogenee a *meno della divisione esplicita per la coordinata omogenea*

$$\begin{bmatrix} x \\ y \\ z \\ (z/f) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & (1/f) & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} x / \frac{z}{f} \\ y / \frac{z}{f} \\ f \\ 1 \end{bmatrix}$$



# Filtr

- L'applicazione di un filtro ad un'immagine digitale è orientata al cosiddetto *miglioramento della qualità* (image enhancement)
- Il miglioramento della qualità può essere inteso in senso generale, per esempio per ridurre il rumore o aumentare la nitidezza dell'immagine, ma può essere anche mirato a mettere in evidenza delle caratteristiche non ben visibili nell'immagine originale

# Filtr

- I filtri possono essere applicati sia direttamente nel dominio spaziale sia nel dominio della frequenza
  - Filtri spaziali: sono kernel  $h(x, y)$  a supporto finito e, in genere, molto piccolo rispetto alle dimensioni dell'immagine che vengono applicati su ogni pixel usando l'operatore di convoluzione (cross-correlazione)
  - Filtri frequenziali: sono funzioni  $H(u, v)$  della frequenza definite su tutto lo spettro che si applicano tramite moltiplicazione punto a punto con la DFT dell'immagine
  - Ovviamente sussiste la relazione per cui  $H(u, v) = \mathfrak{J}[h(x, y)]$

# Filtrari

- Filtri nel dominio spaziale
  - Dalla teoria dei sistemi lineari sappiamo che per un'immagine  $I(i, j)$  di dimensione  $M \times N$

$$I(i, j) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} I(k, l) \delta(i - k, j - l)$$

# Filtrari

- Filtri nel dominio spaziale
  - Se applichiamo una trasformazione LSI

$$\begin{aligned}\mathcal{L}\{I(i, j)\} &= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} I(k, l) \mathcal{L}\{\delta(i - k, j - l)\} = \\ &= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} I(k, l) h(i - k, j - l) = \\ &= \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h(k, l) I(i - k, j - l) \equiv \boxed{h} * I(i, j)\end{aligned}$$

Filtro

# Filtr

- Filtri nel dominio spaziale
  - Il supporto all'operatore di convoluzione è l'intera immagine, ma normalmente i kernel  $h$  (detti anche *maschere*) sono definiti a supporto finito (tipicamente 3x3 o 5x5) e spesso sono *simmetrici rispetto al loro punto di origine*  $h(0,0)$  per cui è possibile riscrivere il tutto usando la *cross correlazione*
$$\mathcal{L}\{I(i,j)\} = \sum_{k=-a}^a \sum_{l=-a}^a h(-k, -l) I(i+k, j+l) \equiv h \bullet I(i,j)$$
  - La riflessione del kernel rispetto a  $h(0,0)$  non ha alcun effetto pratico se i kernel sono simmetrici

# Filtrari

- Filtri nel dominio spaziale
  - L'applicazione di un kernel ad un pixel sui bordi dell'immagine necessita di aggiungere dei valori attorno all'immagine stessa (*padding*) per rispettare la formulazione dell'operazione di convoluzione o cross-correlazione
    - Zero padding: si aggiungono  $w/2$  colonne e  $w/2$  righe di valori nulli attorno all'immagine i quali non daranno contributo all'applicazione della maschera sui punti di bordo
    - Constant padding: l'ultima riga/colonna viene ripetuta per riempire il supporto del filtro
    - Wrap o mirror padding: si «avvolge» l'immagine su sé stessa quindi la prima riga/colonna fornisce contributo alla prima e viceversa

# Filtrari

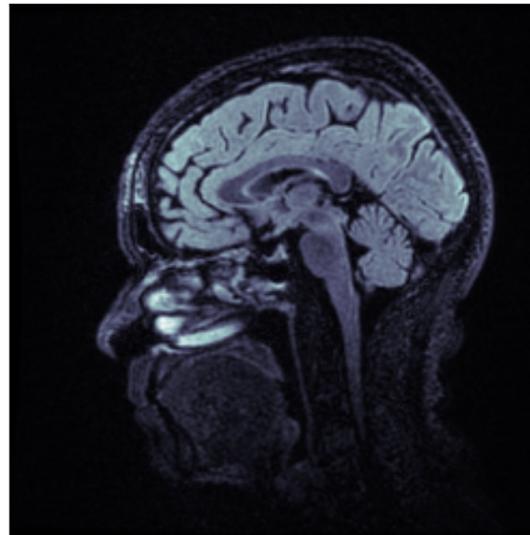
- Filtri nel dominio spaziale – passa basso
  - Sono i filtri che tendono a rimuovere le alte frequenza spaziali e lasciar passare quelle basse. Si usano per la riduzione del rumore («smussamento» o *smoothing*)
  - Box filter o filtro di media: ha una maschera in cui  $h(k_x, l_y) = 1/w^2$ , dove  $w$  è la dimensione della maschera

$$\begin{array}{|c|c|c|} \hline & 1/9 & 1/9 & 1/9 \\ \hline 1/9 & & & \\ \hline & 1/9 & 1/9 & 1/9 \\ \hline 1/9 & & & \\ \hline \end{array}$$

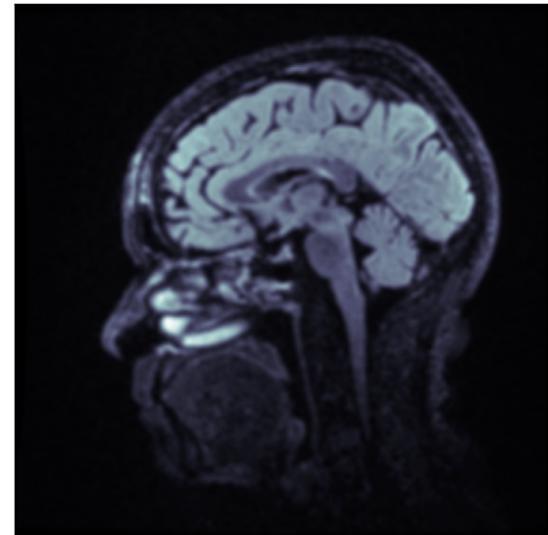
Fonte: Suetens «Fundamentals of Medical Imaging», 3rd ed.

# Filtri

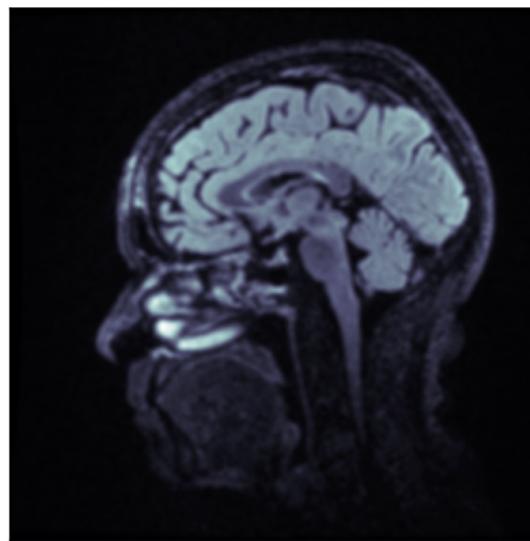
- Filtri nel dominio spaziale – passa basso
  - Sono i filtri che tendono a rimuovere le alte frequenza spaziali e lasciar passare quelle basse. Si usano per la riduzione del rumore («smussamento» o *smoothing*)
    - Box filter o filtro di media: ha una maschera in cui  $h(k, l) = 1/w^2$ , dove  $w$  è la dimensione della maschera



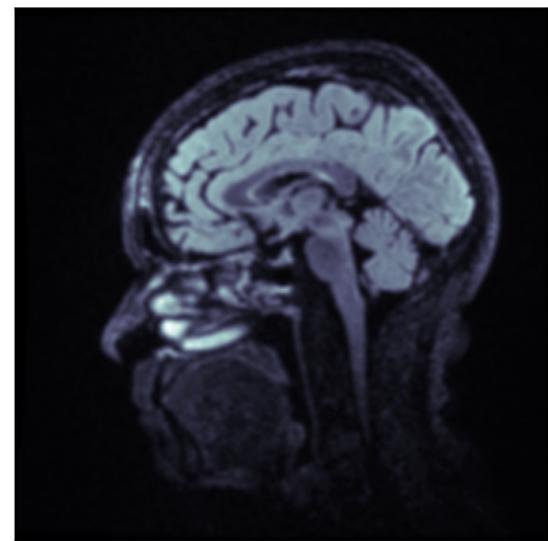
Originale



Box filter 3x3



Box filter 5x5



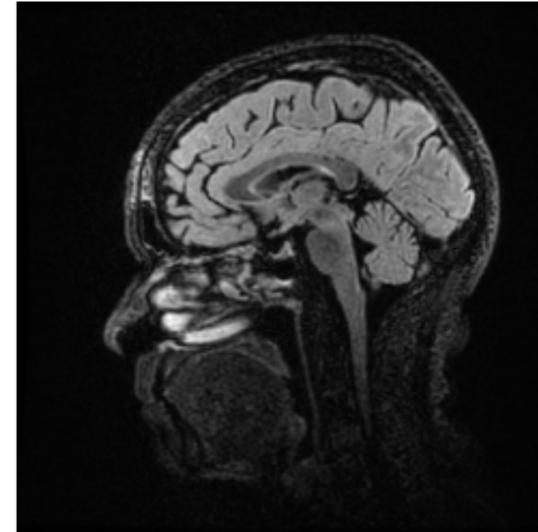
Box filter 7x7  
LABORATORIO DI INTERAZIONE UOMO-MACCHINA  
CHILAB

# Filtr

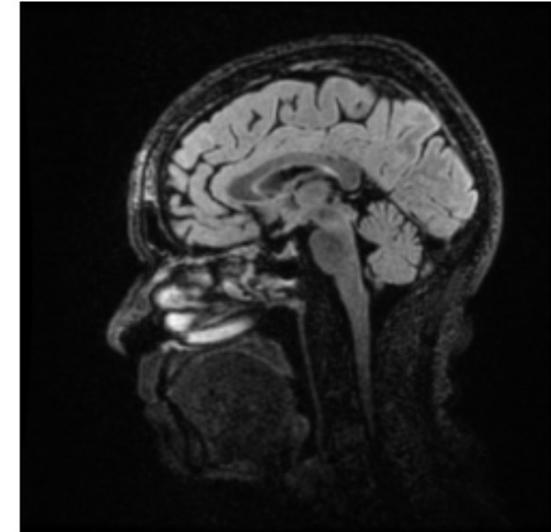
- Filtri nel dominio spaziale – gaussiano
  - E' un filtro passa-basso *a simmetria circolare* (isotropico: la risposta non dipende dall'orientazione). Basta definirne il profilo 1D e ruotarlo sul suo asse
- L'azione passa-basso aumenta all'aumentare di  $\sigma$ , ma questo influenza la dimensione  $w \geq \lceil 6\sigma \rceil$  in modo tale da consentire l'azzeramento dei valori della gaussiana

# Filtri

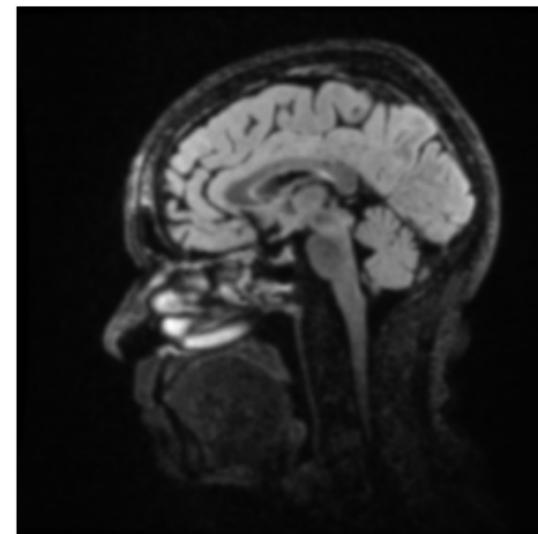
- Filtri nel dominio spaziale – gaussiano



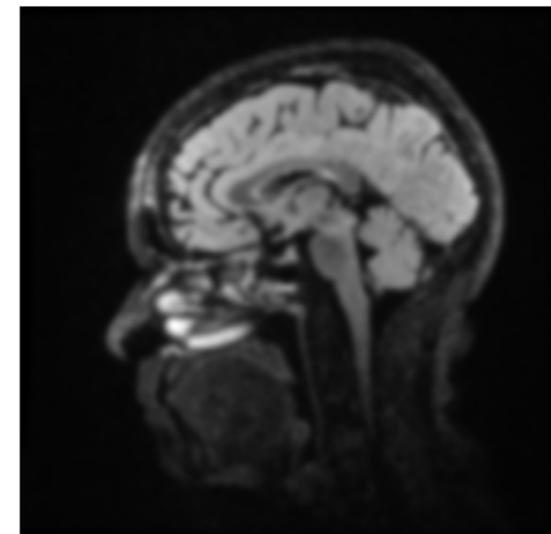
Originale



Gaussiano  $\sigma = 0.5$  (3x3)



Gaussiano  $\sigma = 1.0$  (6x6)

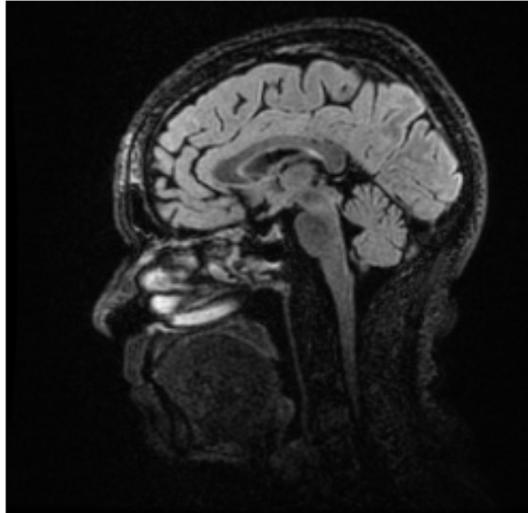


Gaussiano  $\sigma = 1.5$  (9x9)

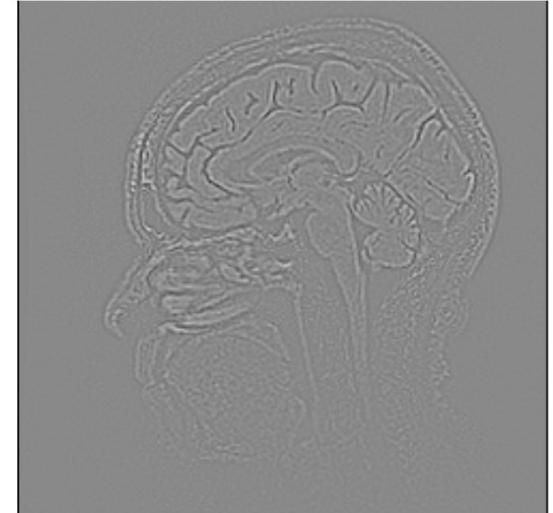
# Filtri

- Filtri nel dominio spaziale – passa alto
  - Il filtro passa alto mira a esaltare le alte frequenza spaziali (contorni dettagli)
  - Si può ottenere sottraendo da  $I$  il passa basso complementare

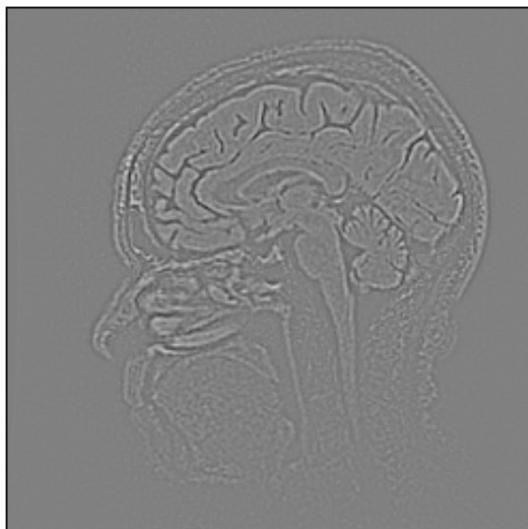
$$hp * I(i, j) = I(i, j) - lp * I(i, j) = [\delta - lp] * I(i, j)$$



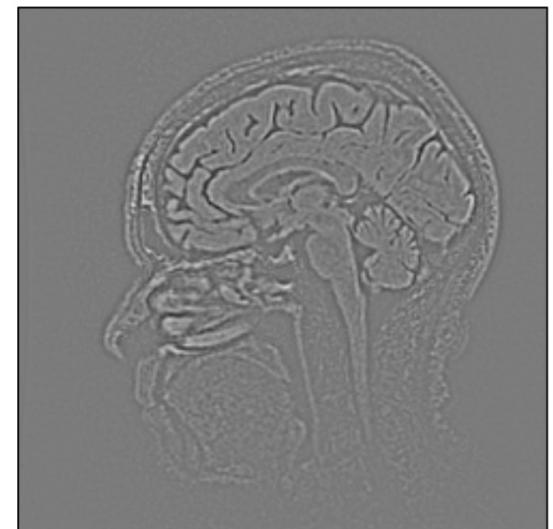
Originale



Passa alto da Gaussiano (3x3)



Passa alto da Gaussiano (6x6)



Passa alto da Gaussiano (9x9)

# Filtr

- Filtri nel dominio spaziale – passa banda e rigetta banda
  - Questi filtri mirano a estrarre o rimuovere un certo intervallo di frequenze spaziali
  - Possono essere ottenuti algebricamente dal passa basso

Filter type	Spatial kernel in terms of lowpass kernel, $lp$
Lowpass	$lp(x, y)$
Highpass	$hp(x, y) = \delta(x, y) - lp(x, y)$
Bandreject	$br(x, y) = lp_1(x, y) + hp_2(x, y)$ $= lp_1(x, y) + [\delta(x, y) - lp_2(x, y)]$
Bandpass	$bp(x, y) = \delta(x, y) - br(x, y)$ $= \delta(x, y) - [lp_1(x, y) + [\delta(x, y) - lp_2(x, y)]]$

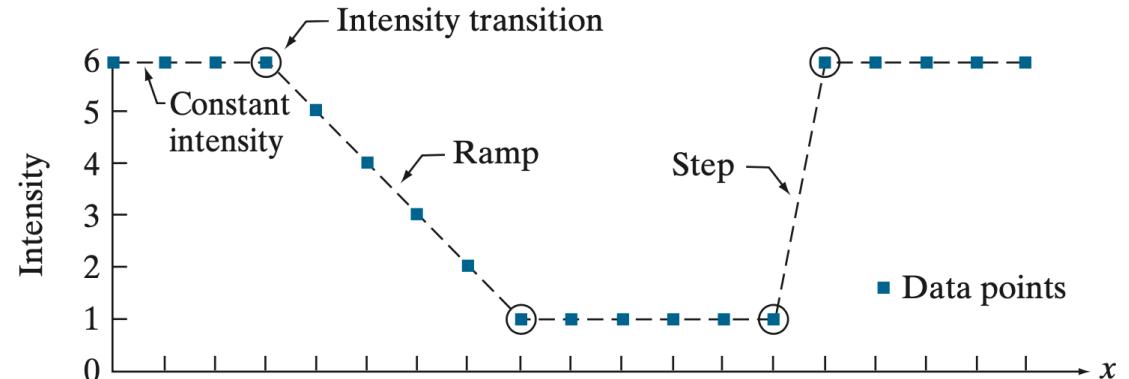
Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Filtri

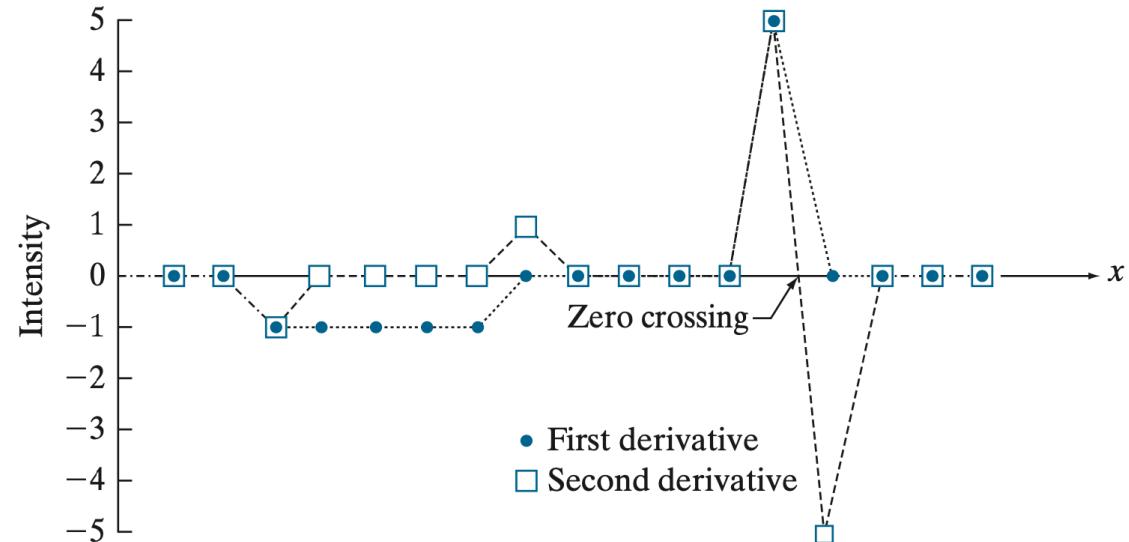
- Filtri nel dominio spaziale – filtri derivativi
  - Sono dei filtri orientati al miglioramento della qualità dell'immagine in termini di esaltazione dei contorni
  - Sono da considerarsi operatori lineari perché le derivate prime e seconde discrete sono espresse con sommatorie

# Filtri

- Filtri nel dominio spaziale – filtri derivativi
  - Le derivate prime hanno dei massimi/minimi in corrispondenza delle variazioni
  - Le derivate seconde hanno un andamento particolare in corrispondenza dei bordi, detto *zero-crossing*



Values of scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	1	6	6	6	6	6
1st derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	0	5	0	0	0	0
2nd derivative	0	0	-1	0	0	0	0	0	1	0	0	0	0	0	0	5	-5	0	0	0



Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Filtr

- Filtri nel dominio spaziale – filtri derivativi

$$\frac{\partial I}{\partial x} \approx I(x+1, y) - I(x, y), \quad \frac{\partial I}{\partial y} \approx I(x, y+1) - I(x, y)$$

$$\frac{\partial^2 I}{\partial x^2} \approx \left. \frac{\partial I}{\partial x} \right|_{x+1} - \left. \frac{\partial I}{\partial x} \right|_x = I(x+1, y) + I(x-1, y) - 2I(x, y)$$

$$\frac{\partial^2 I}{\partial y^2} \approx \left. \frac{\partial I}{\partial y} \right|_{y+1} - \left. \frac{\partial I}{\partial y} \right|_y = I(x, y+1) + I(x, y-1) - 2I(x, y)$$

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} = I(x+1, y) + I(x-1, y) + I(x, y+1) + I(x, y-1) - 4I(x, y)$$

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Filtr

- Filtri nel dominio spaziale – filtri derivativi
  - I *filtri di Sobel* approssimano il gradiente lungo x ed y
  - Il coefficiente 2 al centro vuole esaltare un po' la zona centrale e fornire smoothing nella direzione perpendicolare al gradiente

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Filtr

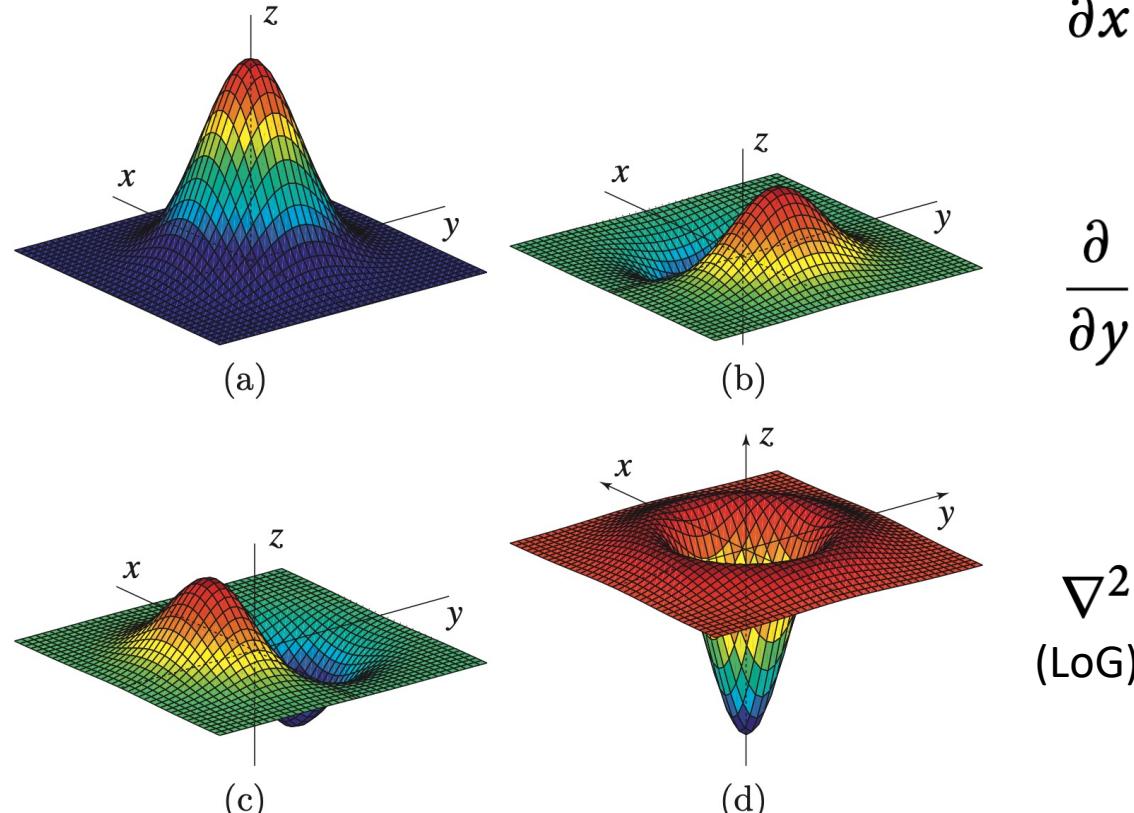
- Filtri nel dominio spaziale – filtri derivativi
  - Un altro approccio alla discretizzazione della derivata è la sua approssimazione con la derivata della convoluzione di  $I$  con una funzione continua e derivabile come, per esempio la gaussiana:  $g^*I$
  - L'operatore di convoluzione gode di commutatività rispetto agli operatori di gradiente e laplaciano che sono lineari:

$$\Im[\nabla(g(x) * I(x))] = [j2\pi k \cdot G(k)] \cdot I(k) = \Im[\nabla g(x) * I(x)]$$

$$\Im[\nabla^2(g(x) * I(x))] = [(j2\pi k)^2 \cdot G(k)] \cdot I(k) = \Im[\nabla^2 g(x) * I(x)]$$

# Filtri

- Filtri nel dominio spaziale – filtri derivativi



Le maschere sono ottenute  
campionando le derivate  
della gaussiana

0.01	0.08	0.01
0.08	0.64	0.08
0.01	0.08	0.01

0.05	0	-0.05
0.34	0	-0.34
0.05	0	-0.05

0.05	0.34	0.05
0	0	0
-0.05	-0.34	-0.05

0.3	0.7	0.3
0.7	-4	0.7
0.3	0.7	0.3

# Filtrati

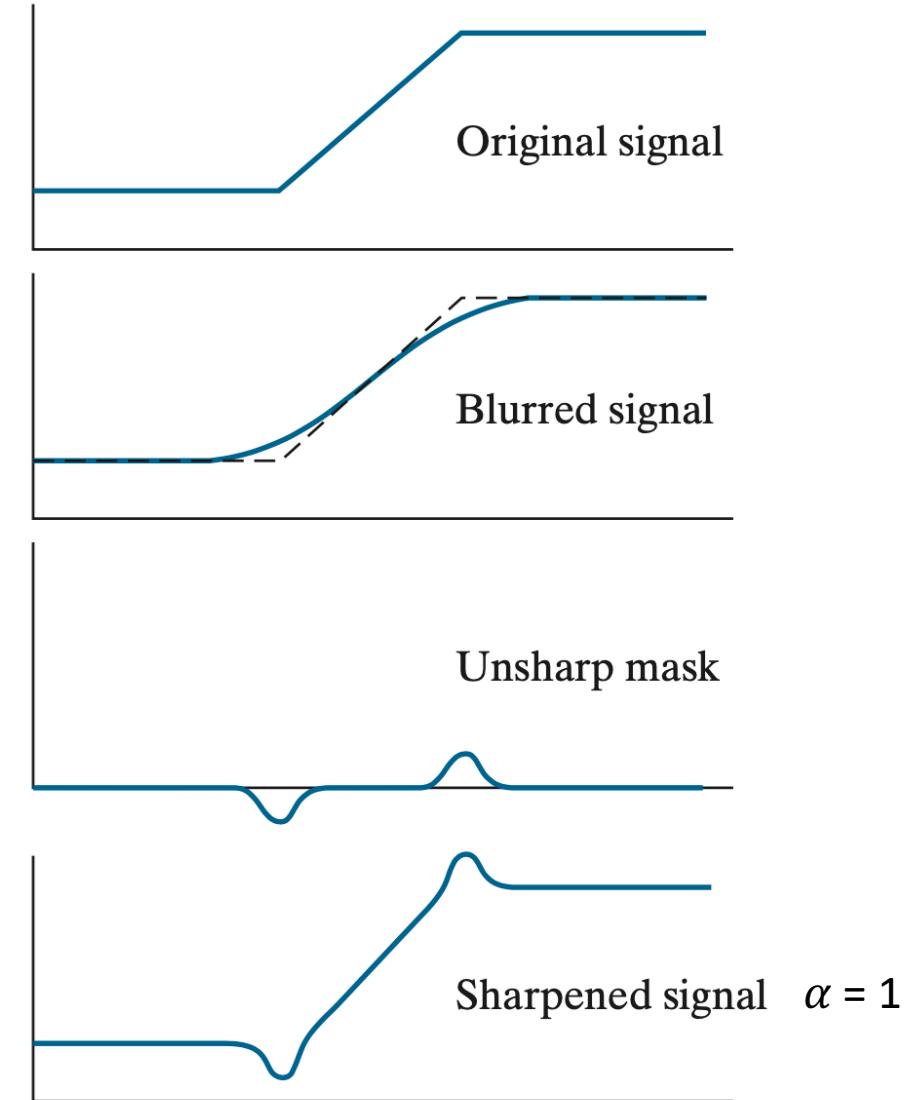
- Filtri nel dominio spaziale – filtri derivativi
  - Il filtro *LoG – Laplacian of the Gaussian* – può essere approssimato con la *Difference of Gaussians (DoG)*
  - Per cui  $DoG * I = (G_{\sigma_1} * I - G_{\sigma_2} * I) = (G_{\sigma_1} - G_{\sigma_2}) * I$
  - Le due deviazioni standard sono molto vicine tra loro

$$\nabla^2 g(\vec{r}) = \frac{1}{\sigma^4} (\vec{r}^2 - 2\sigma^2) g(\vec{r})$$
$$\frac{\partial g(\vec{r})}{\partial \sigma} = \sigma \nabla^2 g(\vec{r}) \approx \frac{g(\vec{r}, k\sigma) - g(\vec{r}, \sigma)}{k\sigma - \sigma}$$
$$g(\vec{r}, k\sigma) - g(\vec{r}, \sigma) \approx (k - 1)\sigma^2 \nabla^2 g(\vec{r})$$

Approssimazione valida per  $k$  piccolo

# Filtri

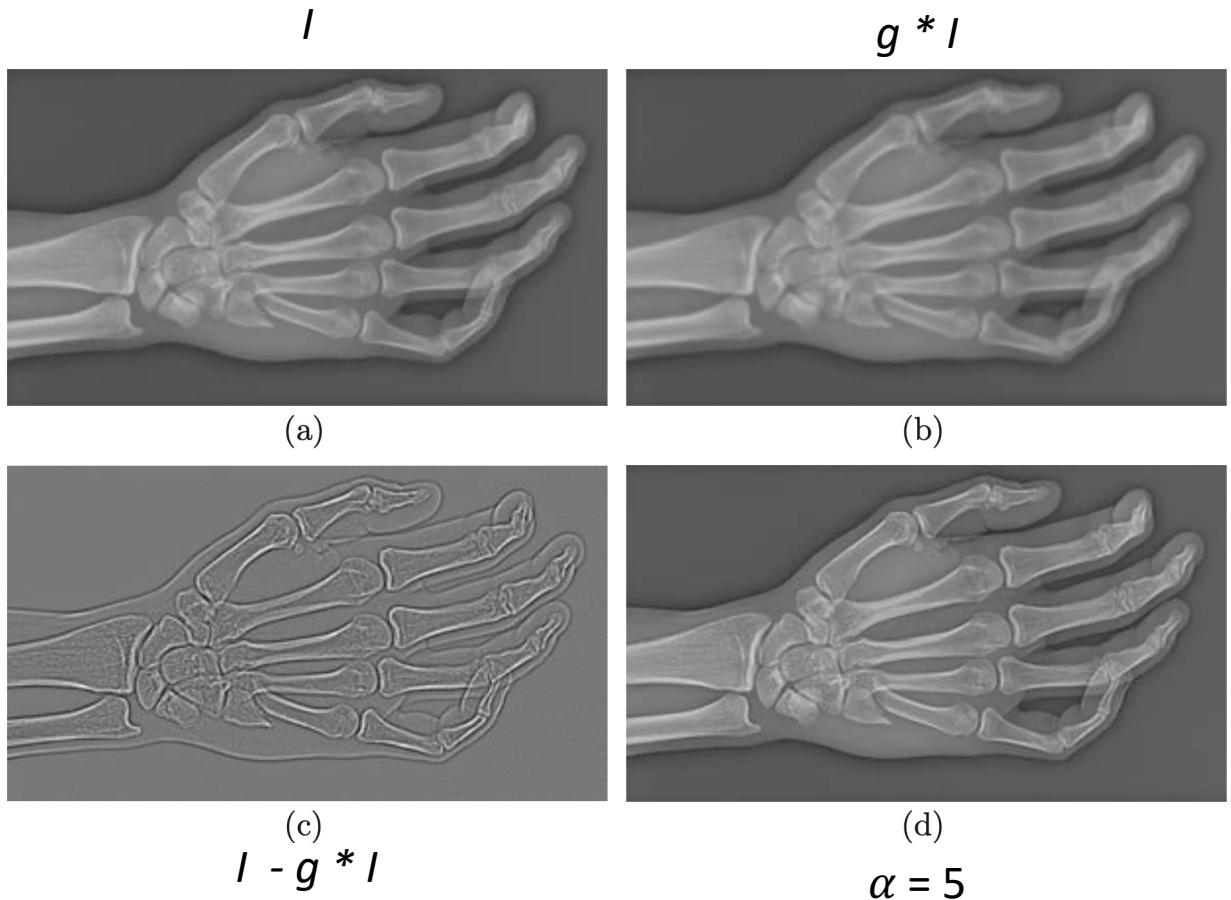
- Filtri nel dominio spaziale – unsharp masking
  - Il processo di unsharp masking esalta i contorni dell'immagine e preserva le zone con variazione lenta
  - $I' = I + \alpha(I - g * I)$



Fonte: Gonzalez, Woods «Digital Image Processing», 4th ed.

# Filtrari

- Filtri nel dominio spaziale – unsharp masking
  - Il processo di unsharp masking esalta i contorni dell'immagine e preserva le zone con variazione lenta
  - $I' = I + \alpha(I - g * I)$



Fonte: Suetens «Fundamentals of Medical Imaging», 3rd ed.

# Filtrri

- Filtri nel dominio della frequenza
  - Un filtro frequenziale trasforma l'immagine originale  $I(x, y)$  in una nuova immagine  $I'(x, y) = \text{Re}\{\mathcal{F}^{-1}[H(u, v) I(u, v)]\}$   
 $= \text{Re}\{\mathcal{F}^{-1}[H(u, v)\text{Re}\{I(u, v)\} + j H(u, v)\text{Im}\{I(u, v)\}]\}$
  - Questo tipo di filtri viene detto *zero-phase-shift* poiché viene applicato egualmente sulla parte reale e su quella immaginaria della DFT di  $I$
  - Nel dominio della frequenza i filtri visti sinora sono di semplice implementazione perché la funzione di trasferimento va semplicemente moltiplicata per la trasformata dell'immagine

# Filtrari

- Filtri nel dominio della frequenza – algoritmo di filtraggio
  1. L'immagine viene trasformata con la *Discrete Fourier Transform* (DFT) che viene implementata tramite l'algoritmo *Fast Fourier Transform* (FFT)
  2. Si crea una pseudo-immagine nel dominio della frequenza, con la stessa dimensionalità di  $I(u, v)$ , che descrive punto a punto  $H(u, v)$
  3. Si moltiplicano  $H$  e  $I$  per ogni  $(u, v)$
  4. Si calcola la IDFT del prodotto

# Filtrari

- Filtri nel dominio della frequenza – caratteristiche della DFT
  - Da un punto di vista puramente computazionale, la matrice dei coefficienti complessi della DFT viene gestita con gli indici che variano come le frequenze per cui il punto  $(u, v) = (0, 0)$  *si trova in alto a sinistra e non al centro*
  - La soluzione sarà quella di effettuare lo swap dei quadranti: 1° e 4°, 2° e 3°
    - Corrisponde a fare lo swap delle righe che distano metà della dimensione di riga, seguito dallo swap delle colonne che distano metà della dimensione di colonna

# Filtrari

- Filtri nel dominio della frequenza – caratteristiche della DFT
  - La DFT e la IDFT sono funzioni periodiche per cui la IDFT del prodotto  $H \cdot I$ , che è equivalente alla convoluzione delle IDFT separate porta a un *errore dovuto all'interferenza tra queste due funzioni periodiche con periodo simile*
  - Si può dimostrare che, se  $I$  ha dimensione  $M \times N$ , la soluzione il padding di  $I$  nel dominio spaziale fino a  $2M \times 2N$  e il prodotto nel dominio della frequenza con un filtro  $H$  della medesima dimensione

# Filtri

- Filtri nel dominio della frequenza – algoritmo di filtraggio (finale)
  1. Si crea una nuova immagine  $I_p$  di dimensioni  $2M \times 2N$  con padding e se ne calcola la DFT
  2. Si effettua lo swap dei quadranti di  $I_p(u, v)$ , la DFT di  $I_p$
  3. Si crea il filtro  $H(u, v)$  di dimensioni  $2M \times 2N$  che sia una funzione simmetrica rispetto al centro di coordinate  $(M, N)$
  4. Si moltiplicano  $H$  e  $I_p$  per ogni  $(u, v)$  – parte reale e immaginaria ovvero zero-phase-shift
  5. Si fa di nuovo lo swap dei quadranti del prodotto, si calcola la sua IDFT e se ne considera la parte reale
  6. Si considera la regione centrale del risultato, avente dimensione  $M \times N$ , come immagine filtrata