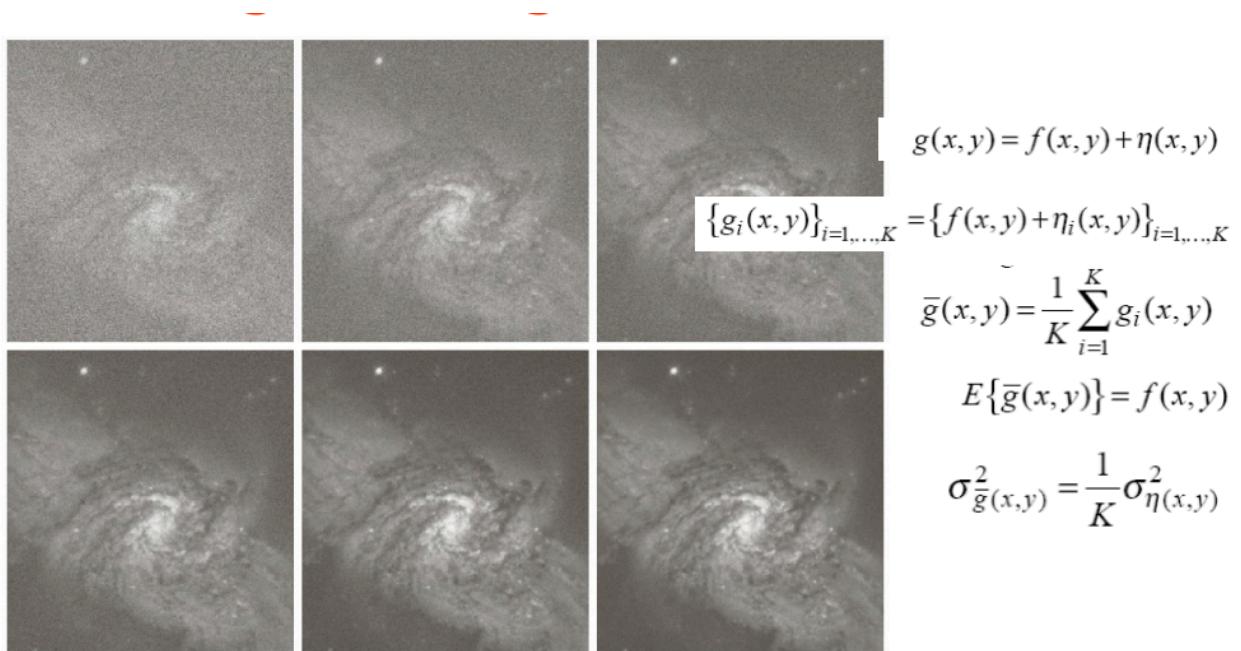


Lezione 4 - Fondamenti di immagini digitali - 21/10/2024

Un caso comune, è dove abbiamo dati molto sporchi (**noisy**), per esempio se stiamo prendendo immagini astronomiche, il noise è random (di solito a distribuzione gaussiana).

In alcuni casi possiamo prendere **immagini multiple**, perché non c'è nulla che si muove. Possiamo quindi fare una **media** del contenuto di ciascuna immagine. Quindi più immagini abbiamo per fare la media, e più noise possiamo rimuovere.



Quindi non stiamo usando nessun image processing, ma usiamo il fatto che sappiamo la distribuzione della noise (gaussiana). Questo approccio non funziona in tutti i domini, per esempio per un'x-ray non possiamo fare 100 foto, oppure non possiamo tenere fermo un cane per 100 foto.

Nell'immagine, la galassia è pian piano creata dalla media di più immagini.

A sinistra vediamo la foto prima l'iniezione del liquido di contrasto, a destra dopo. Nella prima riga non possiamo vedere molte differenze.

Sapendo che i blood vessels non cambiano posizione, possiamo prendere la prima foto senza liquido di contrasto, e la seconda con il liquido, e fare una sottrazione.

Quindi **sottraiamo** l'immagine originale a quella con il contrasto. Il risultato è in basso a sinistra.

A questo punto possiamo fare dell'Image processing per rendere il liquido più visibile. In basso a destra c'è il risultato. (questo liquido può essere radioattivo quindi non possiamo usarne troppo)

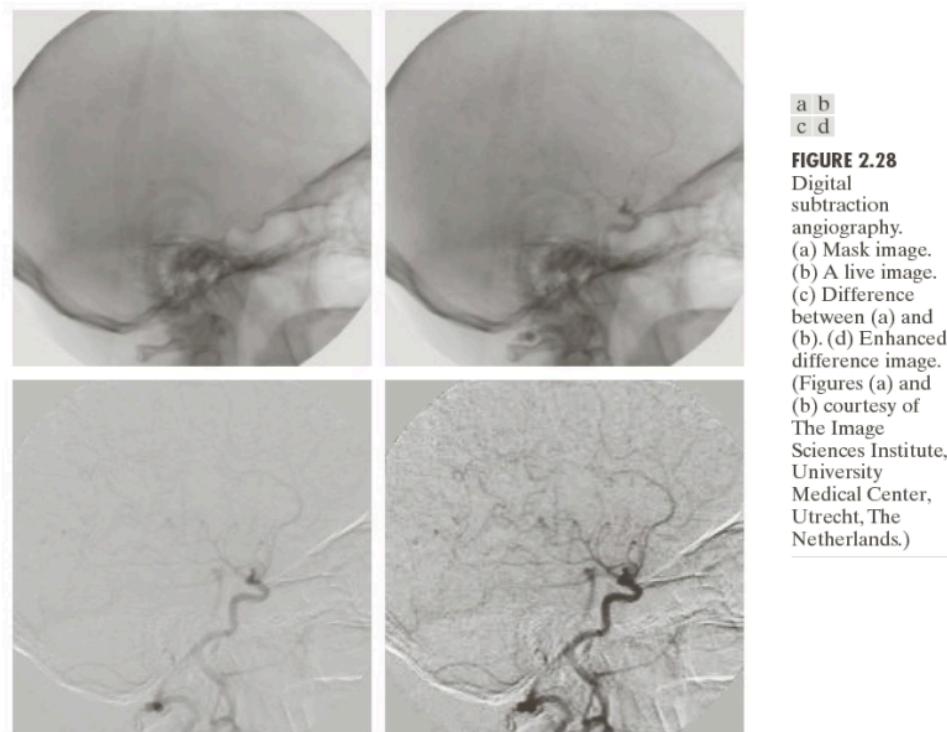


FIGURE 2.28
Digital subtraction angiography.
(a) Mask image.
(b) A live image.
(c) Difference between (a) and (b). (d) Enhanced difference image.
(Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

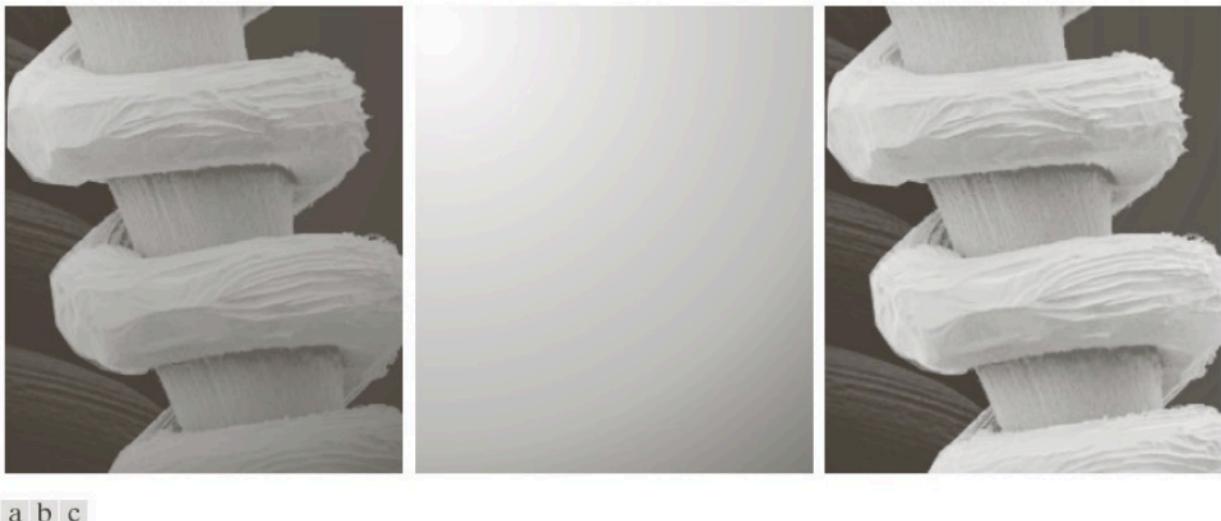
Questo è il filamento di una lampadina in un microscopio elettronico. Nella parte sopra dell'immagine c'è più luce che sotto, è un limite del microscopio.

A sinistra possiamo infatti vedere come sopra ci sia più luce che sotto.

Quindi facciamo un'immagine di un oggetto piatto, per capire la distribuzione della luce.

A questo punto possiamo **dividere** l'immagine originale per ciascun valore della distribuzione della luce. Quindi sopra divideremo per valori grandi, e sotto per

valori minori di 1, incrementando il valore iniziale. In questo modo raggiungiamo il risultato a destra, dove l'oggetto ha la stessa luminosità in tutte le aree.

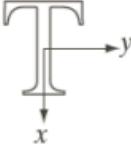
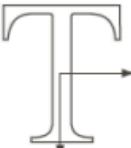
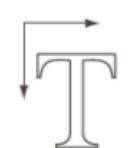


a b c

FIGURE 2.29 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Il tipo più semplice di trasformazione che potremmo applicare ai nostri dati, sono le **trasformazioni geometriche**.

Pensiamo per esempio di avere un'immagine con la lettera T, possiamo ruotarla, cambiare la scala...

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v \cos \theta - w \sin \theta$ $y = v \cos \theta + w \sin \theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Per esempio, nello scaling, assumiamo che nel dominio originale le coordinate erano v e w , ora le nuove coordinate saranno uno scaling di queste coordinate, un coefficiente per la x e uno per la y

La rotazione, ruota l'immagine di una quantità θ .

Traslazione, sposta l'immagine.

Nello shear (taglio?) stiamo aggiungendo un coefficiente diagonale, che deforma l'oggetto.

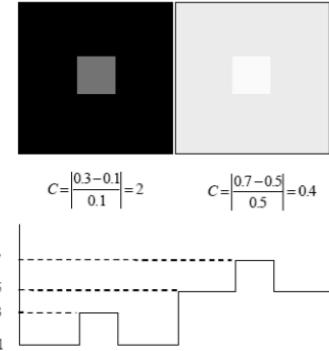
Queste trasformazioni geometriche possono essere usate per **aumentare il training set**.

Contrasto

Il **contrasto in un punto (pixel)** dell'immagine denota la differenza (relativa) tra l'intensità del punto, rispetto all'intensità dei punti vicini.

$$C_p = |(I_p - I_n)/I_n|$$

$$C = \left| \frac{I_p - I_n}{I_n} \right|$$



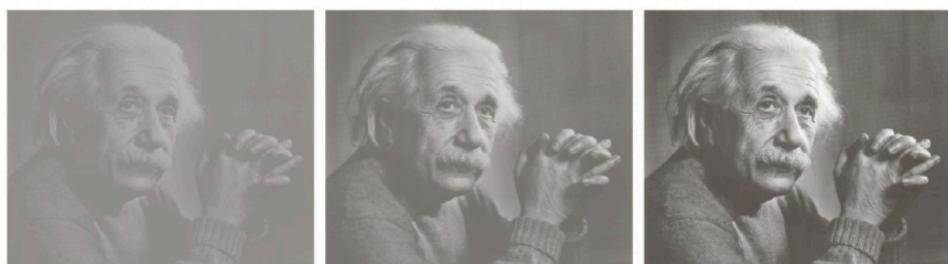
Nell'immagine a destra, possiamo vedere come quella a sinistra ha un contrasto più alto. Il grafico sotto indica l'intensità, il nero è 0.1, il centro dell'immagine a sinistra è 0.3, a destra invece 0.9 fuori e 0.7 il centro.

Se prendiamo solo la differenza tra i due colori, abbiamo che le differenze sono identiche tra le due immagini, però non abbiamo questa percezione. Dobbiamo quindi considerare la misura in modo relativo, raggiungiamo quindi il risultato che ci dice che l'immagine a sinistra ha un contrasto molto più alto che a destra.

Ora spostiamoci da un singolo pixel all'intera immagine.

a b c

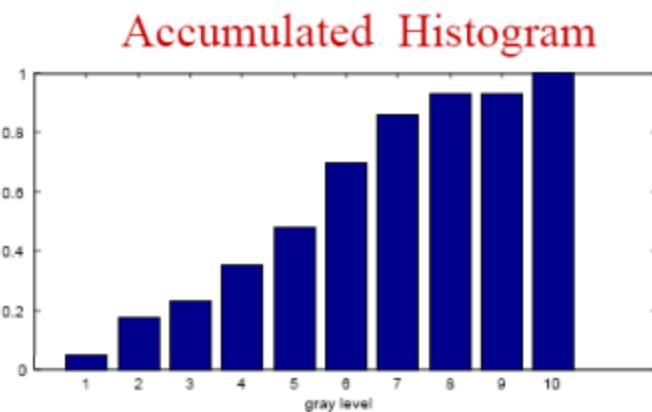
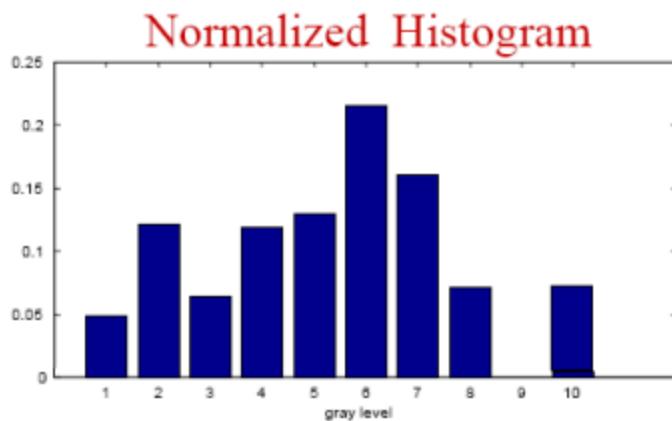
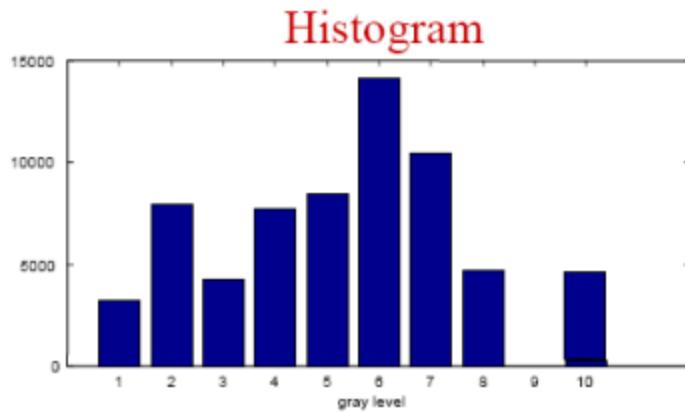
FIGURE 2.41
Images exhibiting
(a) low contrast,
(b) medium
contrast, and
(c) high contrast.



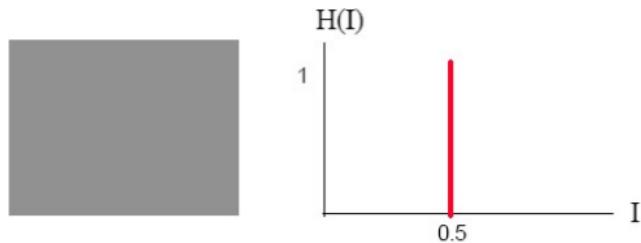
Nelle immagini possiamo essere d'accordo a dire che l'immagine a destra ha più contrasto delle altre.

La definizione del contrasto per l'intera immagine è ambigua. In generale, il contrasto nell'immagine è più alto se i livelli di grigio dell'immagine riempiono l'intero range.

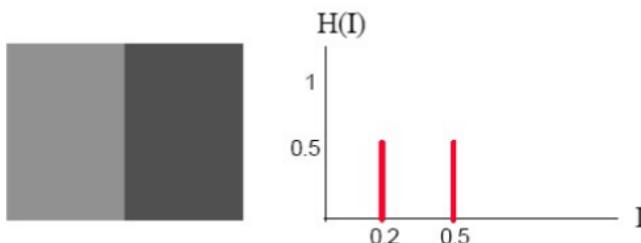
Quindi se abbiamo un istogramma che ci dice per ciascun valore di grigio, quanti pixel hanno quel valore, diciamo che un'immagine ha un contrasto più alto se ha pixel per ciascun valore di grigio.



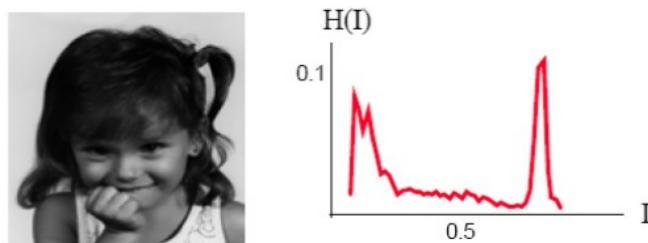
Se abbiamo solamente un colore, nell'istogramma la distribuzione è interamente su un valore, e vale 1.



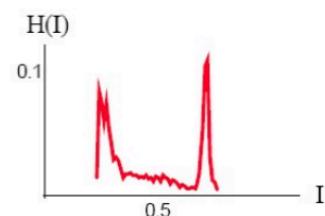
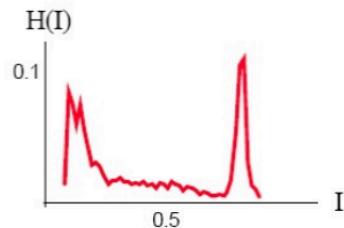
Se abbiamo metà di un colore e metà di un altro, l'istogramma avrà 2 barre con stessa altezza a 0.5. L'istogramma sarebbe lo stesso con una divisione di colore orizzontale, o se fosse una scacchiera. Non stiamo guardando la **geometria**.



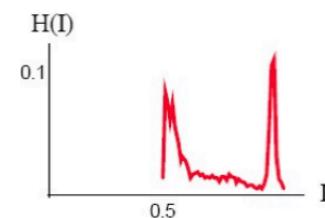
Questo è l'istogramma di un'immagine con un contenuto naturale. Il picco a sinistra indica i capelli, il picco a destra indica il background.



Se vogliamo **cambiare il contrasto** dell'immagine, possiamo agire sull'istogramma, facendo **stretching** e **shifting**.



Decreasing the image contrast



Increasing the new image average

Facendo stretching, riduciamo la distribuzione, avendo zone a destra e a sinistra senza valori.

Se spostiamo l'istogramma a destra, abbiamo la stessa quantità di valori, l'immagine rimane a basso contrasto, però tutti i pixel vengono resi più luminosi.

L'istogramma è molto utile, perché guardando solo all'istogramma, senza vedere l'immagine, sappiamo già quali problemi potrebbero esserci nell'immagine.

Ci sono diversi tipi di operazioni che possiamo fare sulle immagini.

Pixel Operations

- Operation depends on Pixel's value.
- Context free.
- Operation can be performed on the Histogram.

$$I'(x, y) = a \cdot I(x, y) + b$$

• Geometric Operations

- Operation depend on Pixel's coordinates.
- Context free.
- Independent of pixels value.

$$I(x', y') = I(x + a, y + b)$$

Spatial Operations

- Operation depends on Pixel's value and coordinates.
- Context dependent.
- Binary v.s. Gray-Scale images.
- Spatial v.s. Frequency domain.

$$I(x, y) = \sum_{i, j \in \text{Neighbor}} I(i, j) / n$$

Le **operazioni su pixel** prendono in considerazione solamente del valore del singolo pixel. Quindi sono operazioni context-free, non stiamo guardando nessun altro pixel. Questa operazione può essere applicata direttamente all'istogramma.

Le **operazioni geometriche** dipendono solamente dalle coordinate dei pixels.

Le **operazioni spaziali** combinano i primi 2 tipi di operazioni. Sono context-dependent, perchè un pixel anche se ha lo stesso valore di un altro, per il fatto che hanno diverse posizioni, possono essere trasformati in modo diverso. L'esempio è quello che prende la media dei pixels intorno.

Image enhancement

Vogliamo processare un'immagine di modo che l'output sia visualmente migliore rispetto all'input.

Per esempio se vogliamo leggere le targhe di delle macchine, vogliamo togliere la nebbia, mentre se è un'immagine invernale, vogliamo mantenere la nebbia per mantenere il mood dell'immagine. Quindi il miglioramento dipende molto dal contesto.

L'enhancement può essere fatto nel dominio spaziale, operando nell'immagine originale, oppure nel dominio delle frequenze, usando la fourier transform e anti fourier transform.

- To process an image so that output is “visually better” than the input, for a specific application.
- Enhancement is therefore, very much dependent on the particular problem/image at hand.
- Enhancement can be done in either:

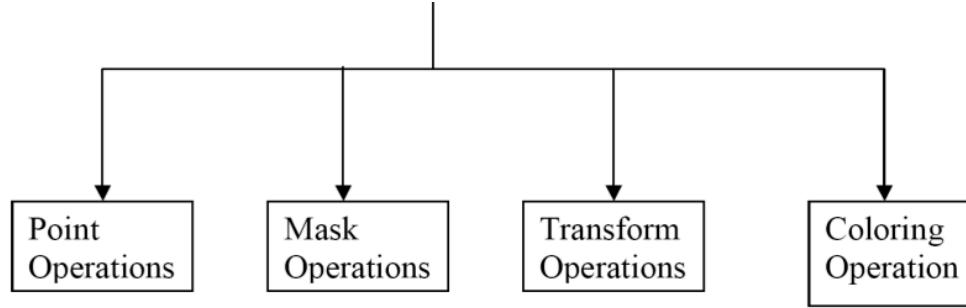
Spatial domain: operate on the original image

$$g(m,n) = T [f(m,n)]$$

Frequency domain: operate on the DFT of the original image

$$G(u,v) = T [F(u,v)], \quad \text{where} \\ F(u,v) = F [f(m,n)], \text{ and } G(u,v) = F [g(m,n)]$$

Abbiamo diverse opzioni per le operazioni che possiamo applicare:



- Image Negative
- Contrast Stretching
- Compression of dynamic range
- Graylevel slicing
- Image Subtraction
- Image Averaging
- Histogram operations
- Smoothing operations
- Median Filtering
- Sharpening operations
- Derivative operations
- Histogram operations
- Low pass filtering
- High pass Filtering
- Band pass filtering
- Homomorphic filtering
- Histogram operations
- False coloring
- Full color processing

La nuova immagine sarà l'immagine originale a cui abbiamo applicato un operatore T

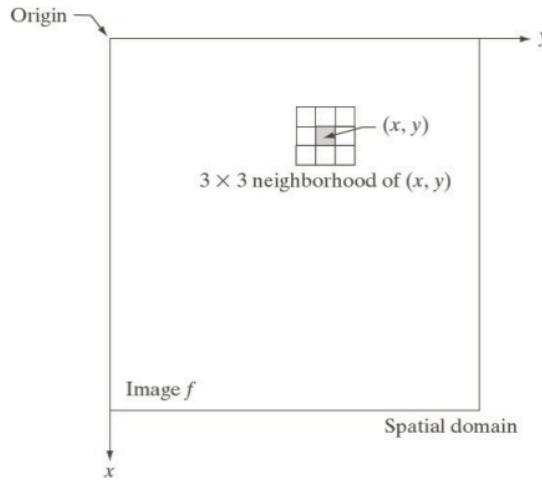


FIGURE 3.1
A 3×3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

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

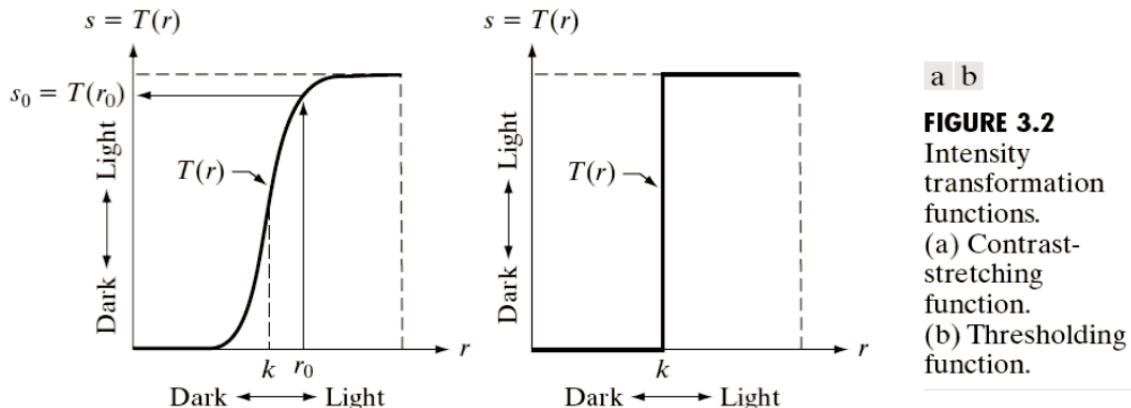
↓ ↓ ↓
 Processed image Input image Operator on f defined over
 some neighborhood of (x, y)

Nelle point operations il nuovo valore dipende solamente dal valore iniziale del pixel. quindi r è il valore in input, a cui applichiamo la transform T, e ricaviamo s.

$$s = T(r)$$

Quindi tutti i pixel sono trattati egualmente.

T è una funzione che mappa $[0,1]$ in $[0,1]$.



a b

FIGURE 3.2
Intensity transformation functions.
(a) Contrast-stretching function.
(b) Thresholding function.

Sono due operazioni diverse, la seconda (thresholding function) è una versione estrema di quella a destra.

Ci sono alcune curve molto usate.

La negative è la retta opposta rispetto all'identità, inverte i livelli di grigio.

La power-law applica $y = x^n$, ci interessa rimanere nel range 0-1 perché qualsiasi sia l'esponente n , rimarremo sempre nel range 0-1. Tutte le curve sono limitate nel range 0-1.

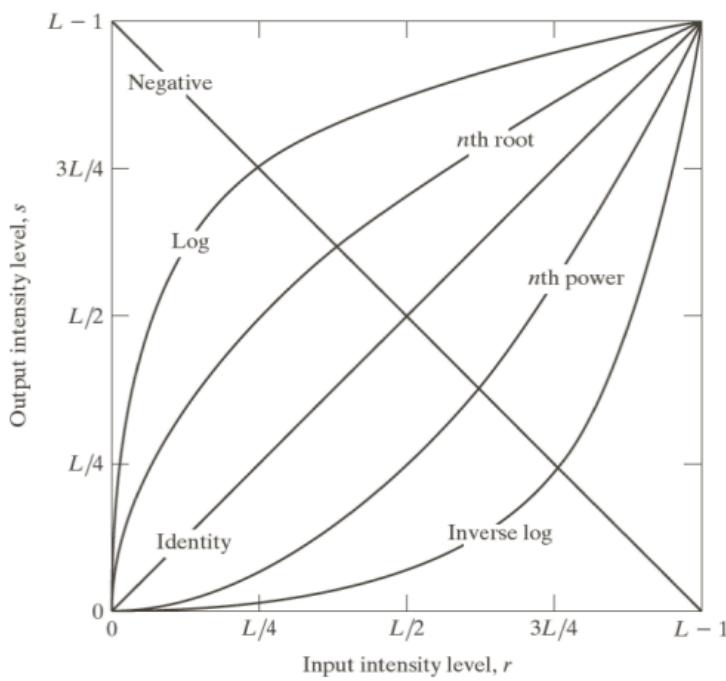
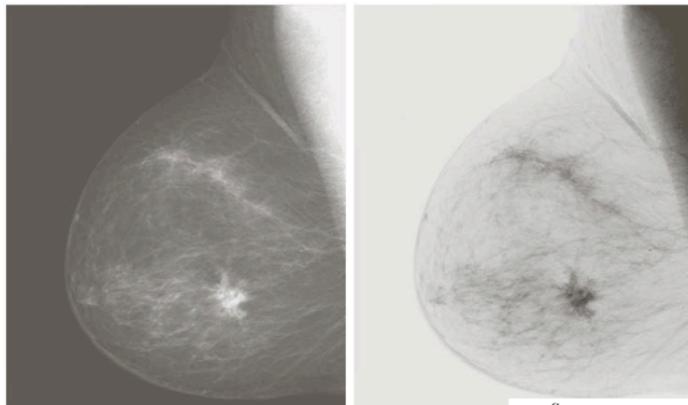


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

- Negative
- Logarithmic
- Power-law

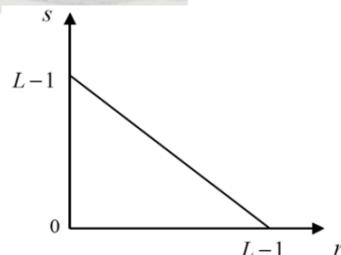
Questa è una mammografia, è un x-ray. In alcuni casi questa rappresentazione può non essere facilmente leggibile. Applicando la negative transform, inverto i colori.



a | b

FIGURE 3.4

(a) Original digital mammogram.
 (b) Negative image obtained using the negative transformation in Eq. (3.2-1).
 (Courtesy of G.E. Medical Systems.)



If the gray levels of the image are
 in the range L-1

$$T(r) = s = L - 1 - r, \quad L : \text{max. grayvalue}$$

Una famiglia importante di trasformazioni è la **gamma correction**.

il "c" è un re-scale nel range desiderato dell'output.

In un lato del grafico abbiamo le curve con gamma >1, e nell'altro le curve con gamma tra 0 e 1.

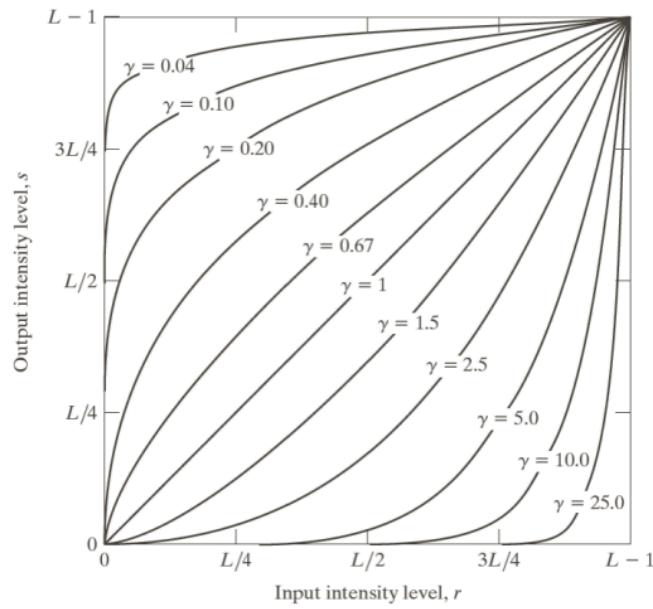


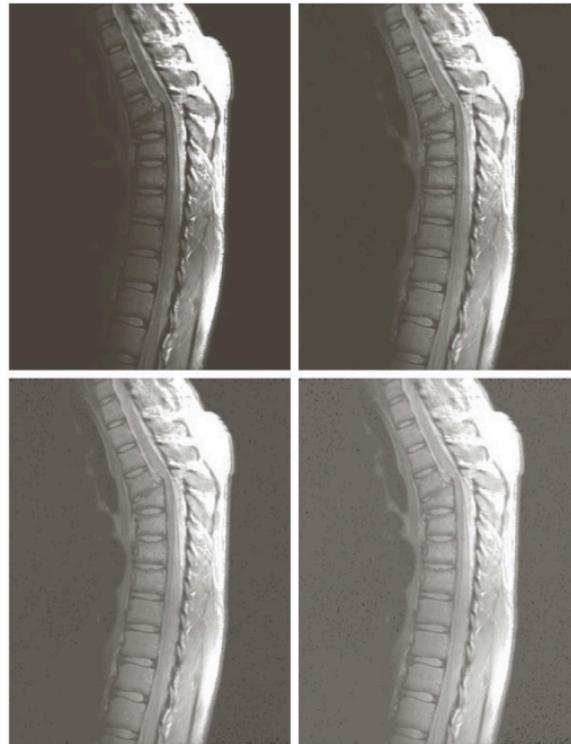
FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

$$s = c r^\gamma$$

where c and γ are positive constants

Per esempio con gamma = 10 stiamo cambiando il contrasto dell'immagine. In questo caso l'immagine viene resa più scura. Siamo ri-scalando i pixel più chiari in un range più grande.

(a) Magnetic resonance image (MRI) of a fractured human spine.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3 , respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)



$$\gamma = 0.6$$

$$\gamma = 0.4$$

$$\gamma = 0.3$$

La risonanza magnetica può essere troppo scura, riducendo il gamma vado a rendere l'immagine più chiara, dove posso vedere meglio le aree dell'immagine erano troppo scure. Possiamo vedere come un disco sia collassato, dove c'è la curva della spina dorsale.

FIGURE 3.9
 (a) Aerial image.
 (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0 , respectively.
 (Original image for this example courtesy of NASA.)

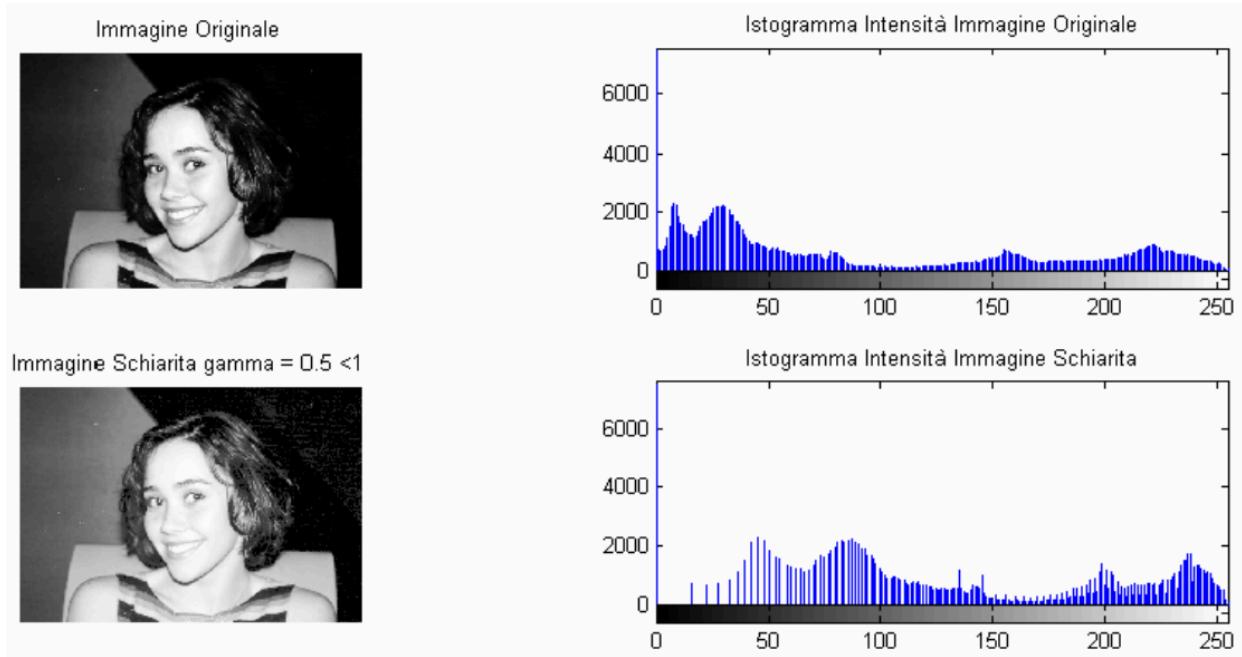


L'inverso può essere applicato a queste immagini che deve usare il pilota automatico. L'informazione è contenuta nelle aree più chiare dell'immagine. Aumentiamo quindi il range di valori per i colori più chiari, scurendo l'immagine.

L'immagine originale ha un buono contrasto, abbiamo valori per praticamente tutti i valori possibili. Se invece voglio rendere l'immagine più chiara, applico una gamma correction, l'istogramma si muove verso a destra, avremo più valori sulla destra dell'istogramma e meno sulla sinistra.

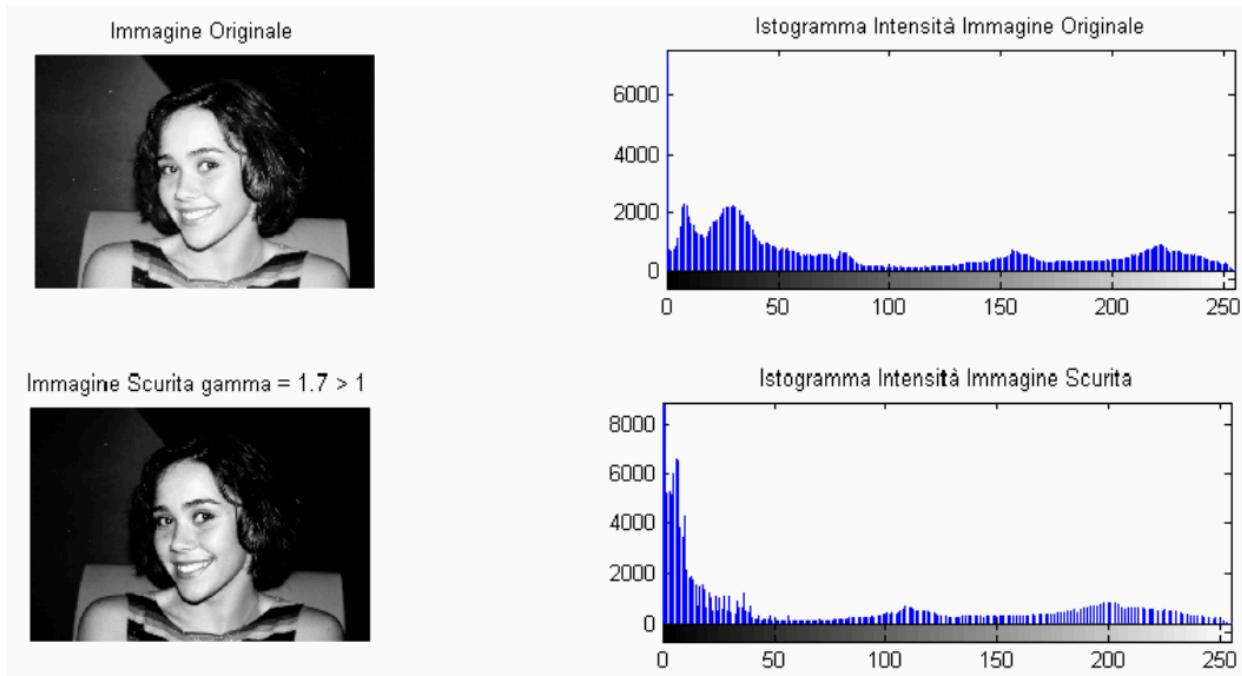
Stiamo quindi comprimendo l'istogramma verso destra.

Gamma Correction (immagine $_{[0,1]}^{\text{gamma}}$)



In questo caso, faccio l'opposto, comprimo verso sinistra, preservando bene o male la forma dell'istogramma.

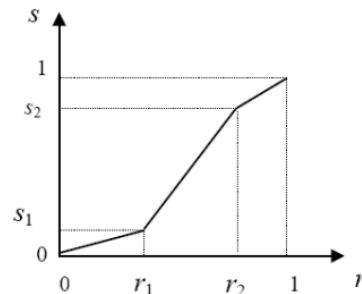
Gamma Correction (immagine $_{[0,1]}^{\text{gamma}}$)



Contrast stretching

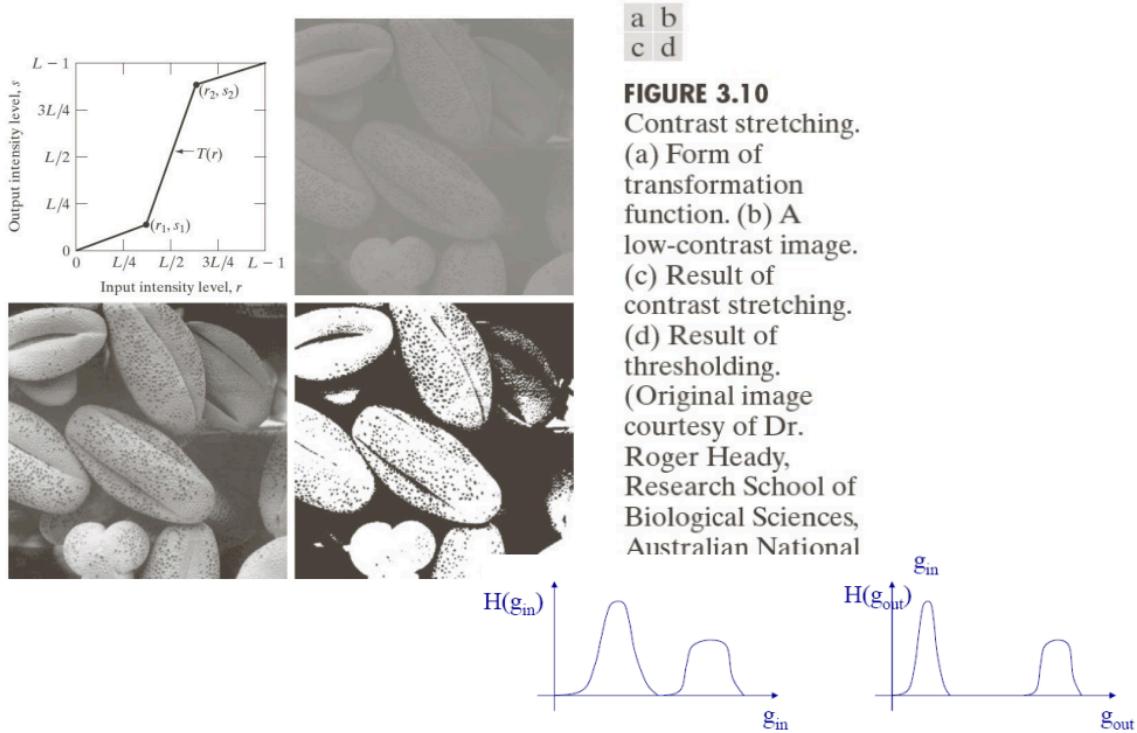
Abbiamo 3 intervalli, in ciascuno abbiamo una linea di trasformazione diversa.

- Increase the dynamic range of gray values in the input image.
- Suppose you are interested in stretching the input intensity values in the interval $[r_1, r_2]$:



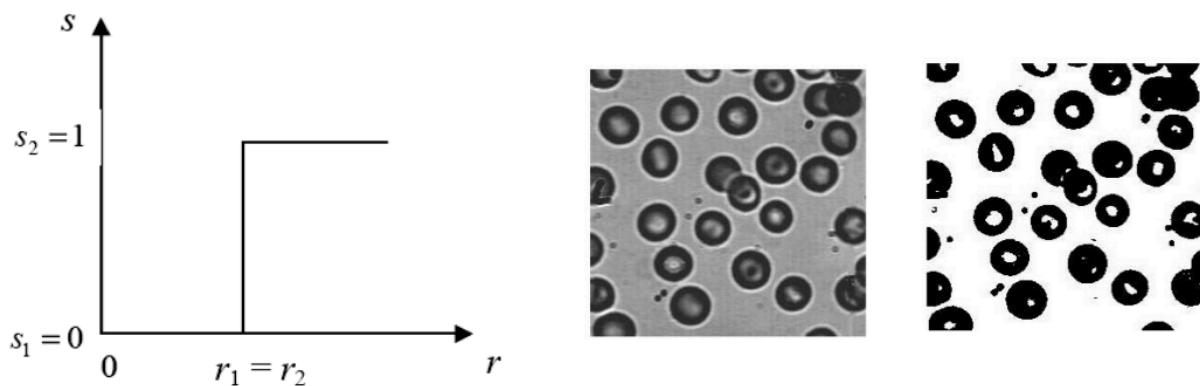
- Note that $(r_1 - r_2) < (s_1 - s_2)$. The gray values in the range $[r_1, r_2]$ is stretched into the range $[s_1, s_2]$.

Questo è un esempio di applicazione di questa trasformazione:



Abbiamo perso molta informazione, ma in alcuni casi per esempio vogliamo solo contare il numero di cellule nell'immagine. è più facile contare tutti i blob neri, rispetto a quelli nell'immagine originale.

- **Binarization: useful when we are only interested in the shape of the objects.**
- $r_1=r_2$, $s_1=0$ and $s_2=1$



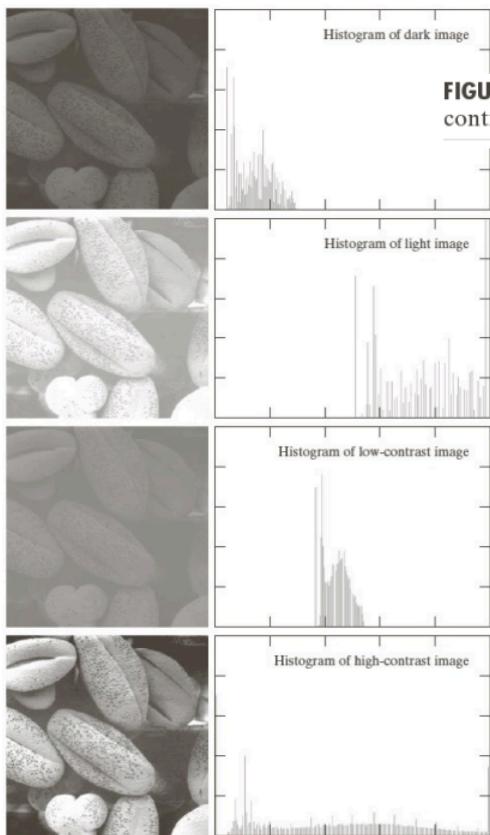


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

La prima immagine è un'immagine scura a basso contrasto. La seconda è un'immagine chiara a basso contrasto. La terza immagine è a basso contrasto. La quarta immagine ha valori su tutto l'istogramma e quindi è un'immagine ad alto contrasto.

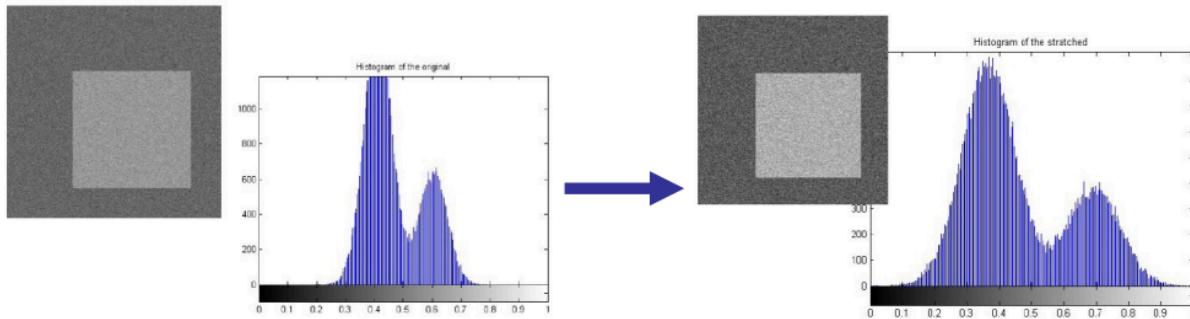
Ci sono operazioni che usano solamente l'istogramma.

L'**istogram sliding**, sposta l'istogramma, rendendo l'immagine più chiara o più scura.

L'**istogram stretching/shrinking** è una compressione o un'espansione del dynamic range.

L'**istogram equalization**, ci arriviamo....

Iistogram stretching:



$$stretch[I] = \left[\frac{I - g_{\min}^0}{g_{\max}^0 - g_{\min}^0} \right] [g_{\max}^1 - g_{\min}^1] + g_{\min}^1$$

$g_{\max}^0 = \max_g I[i, j]$ = maximum gray value in the original image

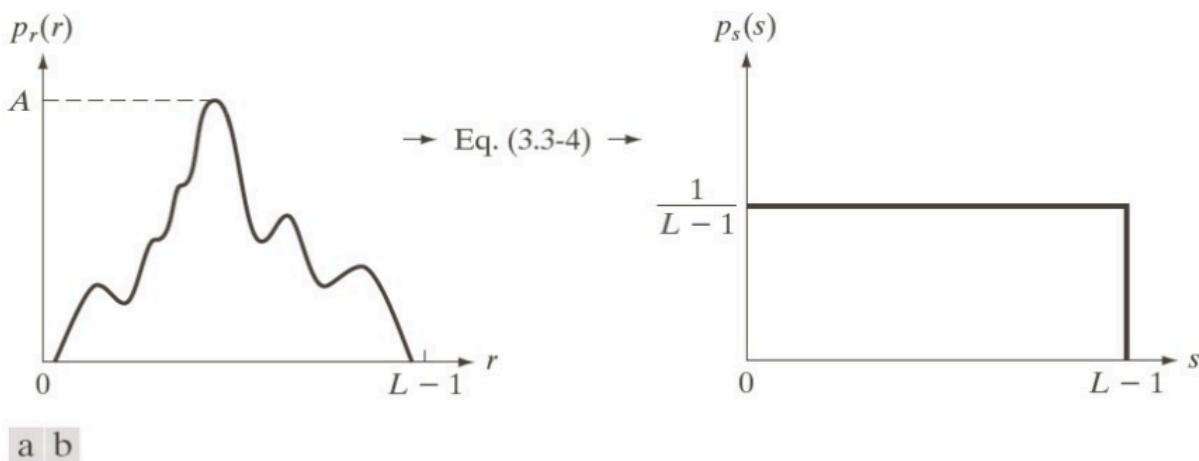
$g_{\min}^0 = \min_g I[i, j]$ = minimum gray value in the original image

g_{\max}^1, g_{\min}^1 = maximum and minimum gray values in the processed image

L'istogramma non ha valori sui lati, quindi vogliamo allargarlo da entrambe le parti, mantenendo la forma dell'istogramma.

Il primo termine dell'equazione è quello che fa lo stretching, il secondo e terzo servono a cambiare il nuovo centro dell'istogramma.

L'histogram equalization è usato per aumentare il contrasto modellando l'istogramma dell'immagine in una distribuzione uniforme su un determinato numero di livelli di grigio. I valori di grigio vengono ridistribuiti sull'intervallo dinamico per avere un numero costante di campioni in ciascun intervallo (cioè in ciascun "bin" dell'istogramma).

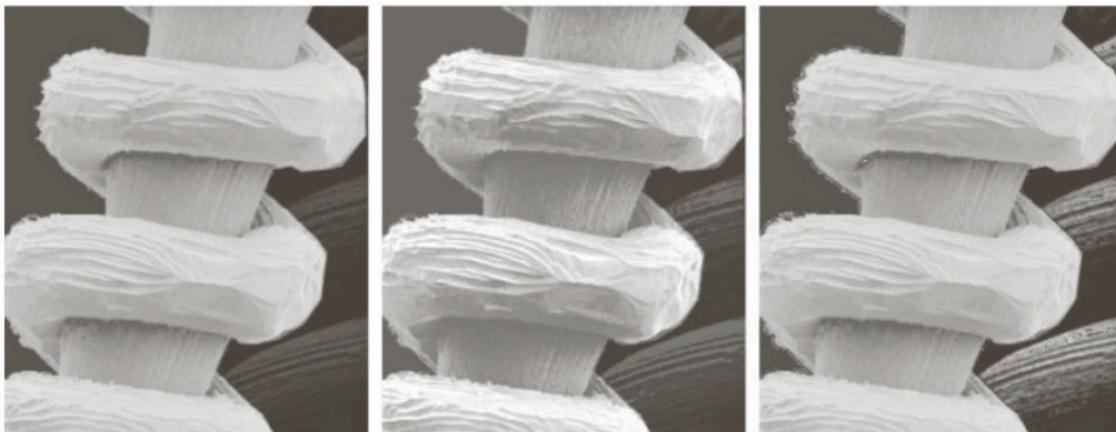


a b

FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r . The resulting intensities, s , have a uniform PDF, independently of the form of the PDF of the r 's.

Questo non è fatto nello stretching, dove viene mantenuta la forma dell'istogramma.

Gli istogrammi ci permettono di applicare **operazioni LOCALI**, piuttosto che su tutta l'immagine.



a b c

FIGURE 3.27 (a) SEM image of a tungsten filament magnified approximately 130 \times . (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Spatial, Smoothing Filters

L'idea è di estendere in 2 dimensioni quello che abbiamo visto con la convoluzione in 1 dimensione. Abbiamo un filter kernel che ha 2 dimensioni.

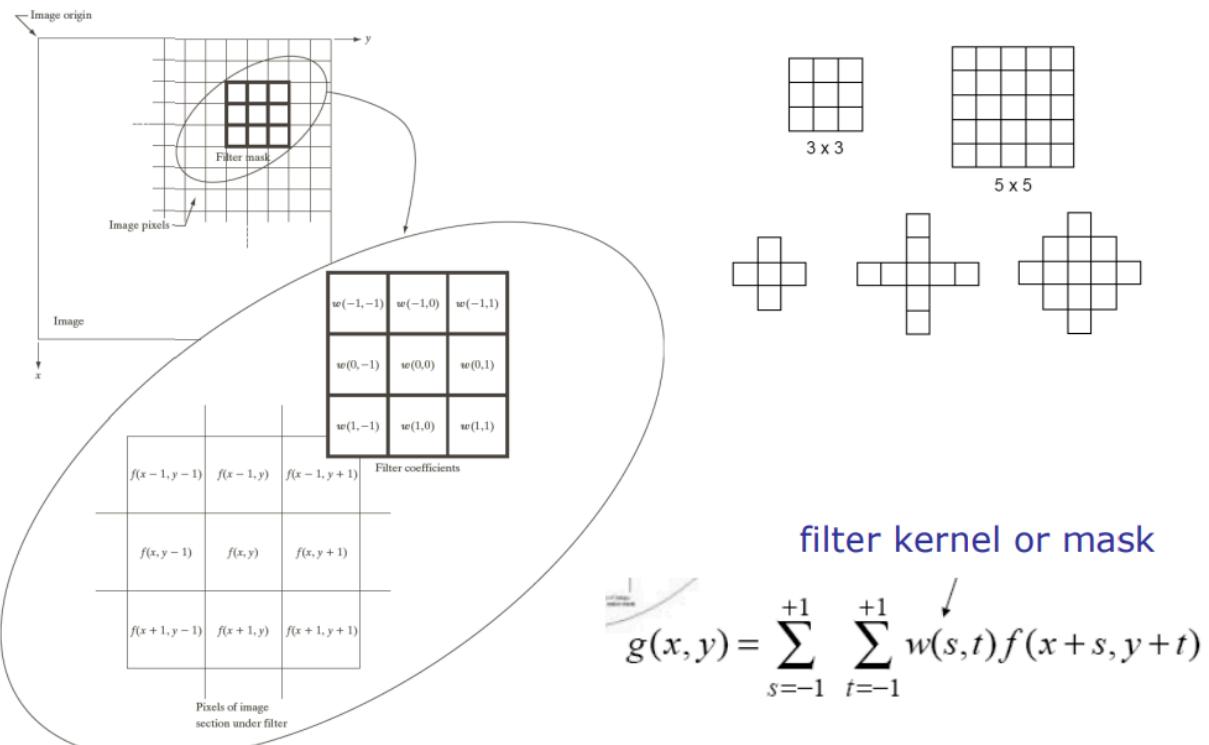
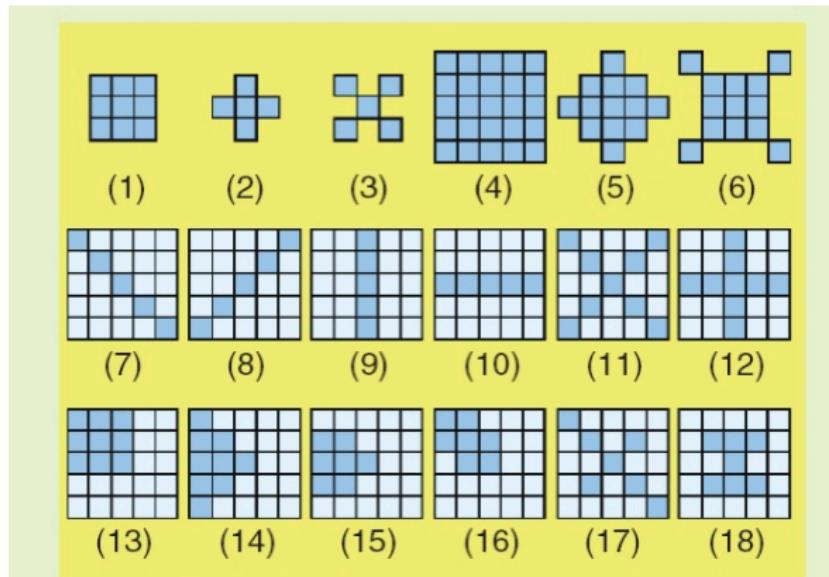


FIGURE 3.28 The mechanics of linear spatial filtering using a 3×3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

Per ogni pixel, facciamo un'operazione con il filtro (3×3 per esempio), e il risultato è quello di quel singolo pixel, poi passiamo al prossimo...

Di solito questi filtri hanno un numero dispari di righe/colonne e di solito sono quadrati.

Ci sono altre forme ma di solito sono embedded in un filtro quadrato.



[FIGB] Most popular window shapes: (1)–(6) conventional spatial windows, (7)–(10) unidirectional windows, (11) and (12) bidirectional windows, (13)–(18) special window structures. Other structures can be obtained by rotating those shown in (13)–(18).

Gli smoothing filters sono anche chiamati low-pass filters perché rimuovono le frequenze alte, che sono quelle che causano un cambiamento rapido di colore tra 2 pixel.

Questi filtri sono usate per blurring (di solito è uno step di preprocessing per rimuovere piccoli dettagli non voluti, prima di estrarre gli oggetti più grandi, o chiudere gaps/curve) e noise reduction (mitigate the effect of noise by linear or nonlinear operations).

Lo smoothing è fatto applicando un'averaging mask.

Aumentando la forza del filtro, questi sono i risultati:

Smoothing

Used to reduce noise through blurring (sharp details are lost). The output is a weighted average of the pixels contained in the neighborhood of the filter mask.

Examples of smoothing masks

$$\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{32} \begin{bmatrix} 1 & 3 & 1 \\ 3 & 16 & 3 \\ 1 & 3 & 1 \end{bmatrix} \quad \frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

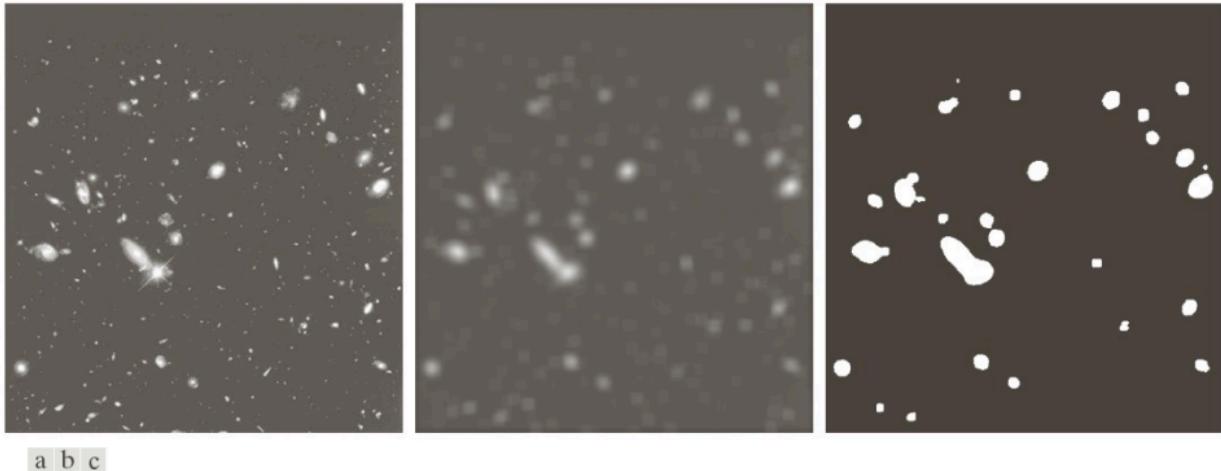


a b
c d
e f

FIGURE 3.33 (a) Original image, of size 500×500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15, 25, 35, 45$, and 55 , respectively. The black squares at the top are of sizes $3, 5, 9, 15, 25, 35, 45$, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50×120 pixels.

Un'applicazione per questo, oltre a ridurre il noise, se abbiamo un'immagine dello spazio, vogliamo mantenere le galassie più grandi e rimuovere quelle piccole.

Applichiamo uno smoothing filter, e poi applichiamo una funzione di thresholding.



a b c

FIGURE 3.34 (a) Image of size 528×485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15×15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Idea:

- Ordinare i pixel in un intorno secondo i loro valori
- Sostituire il pixel centrale con un certo valore trovato dalla lista ordinata

Esempi di filtri statistici d'ordine:

- Minimo
- Massimo
- Mediana: "valore nel mezzo della lista ordinata", 50° percentile, il 50% dei valori è inferiore, il 50% è superiore

I filtri sono non lineari!

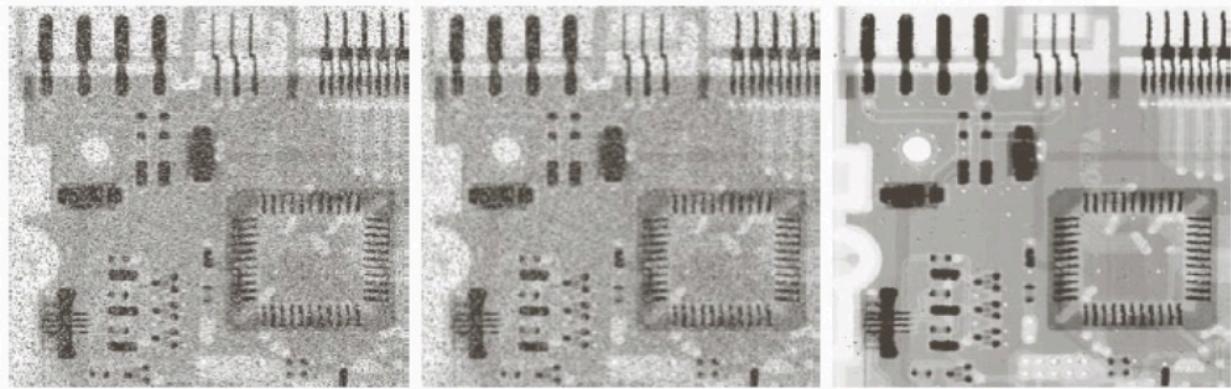
Il filtro mediano per esempio prende la finestra di 9 valori che stiamo valutando, e sovrascrive quello centrale con il valore mediano.



C'è un tipo di noise chiamato "salt and pepper noise" perchè non ha una distribuzione gaussiana, ma ha alcuni pixel completamente bianchi o completamente neri.

Quindi questo tipo di noise apparirà agli estremi della distribuzione della finestra.

Questo è un circuito con salt and pepper noise.



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Color spaces

Ci sono molti spazi di colore, non c'è uno spazio di colore migliore dell'altro, dipende dal tipo di task.

Un modello di colore è un modello matematico che descrive come il colore è rappresentato da un set di numeri. Il numero di componenti può essere più di 3, per esempio in una stampante abbiamo 4 colori. Quando questo modello è associato a una descrizione precisa di come devono essere interpretati i componenti (condizioni di visualizzazione, ecc.), l'insieme di colori risultante può essere effettivamente chiamato spazio colore.

- Device-derived (device-dependent)
 - convenient for describing display device levels
 - RGB, CMY,...
- Intuitive (device-dependent)
 - based in familiar color description terms
 - HSV, HSB, HLS, HIS,...
- Device independent
 - CIE XYZ, Yxy,...
- Perceptually uniform AND device independent
 - CIELUV, CIELAB,...
- Perceptually uniform AND device independent
AND intuitive
 - Munsell,...
- ...

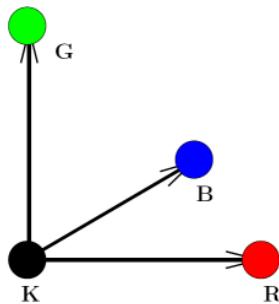
Quelli che **derivano dal device**, significa che ho per esempio una stampante, che per stampare un colore, sa che ho bisogno di una certa quantità di primari, mentre in una stampante diversa ho delle quantità diverse, perchè ho inchiostro diverso. Quindi questi spazi dipendono dal device. Anche nei monitor.

Gli **spazi di colore indipendenti** dal device sono utili quando voglio trasmettere qualcosa, e voglio che l'altra persona veda i colori come li vedo io.

Gli **spazi di colore intuitivi** sono basati su descrizioni del colore come quelle che vengono usate nel parlato. Avremo anche la saturazione, che ci indica quanto è vivido il colore...

Gli spazi di colore **perceptually uniform e device independent**, vuol dire che se ho una differenza di 10, posso sapere quanto sia. Mentre invece in altri spazi di colore la distanza cambia in base al colore.

Il colore finale è la combinazione dei primari pesati.



$$\text{Color} = r\mathbf{R} + g\mathbf{G} + b\mathbf{B}$$

$$= [\mathbf{R}, \mathbf{G}, \mathbf{B}] \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

- R,G,B color primaries are basis vectors
- (r, g, b) tristimulus values specify 3-D coordinates

Each color can be specified by its (r, g, b) coordinates

Red = R \leftrightarrow (r, g, b) = (1, 0, 0)

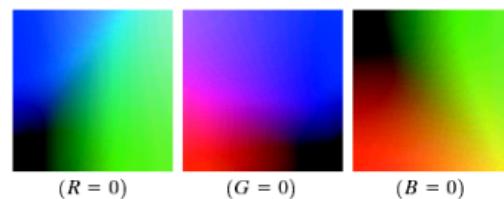
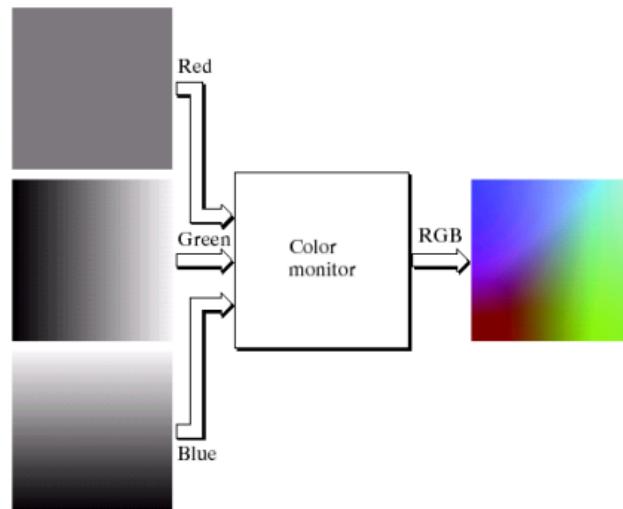
Green = G \leftrightarrow (r, g, b) = (0, 1, 0)

Blue = B \leftrightarrow (r, g, b) = (0, 0, 1)

Il canale rosso non è rosso, è un'informazione grayscale che ci dice quanto rosso c'è. È quel coefficiente applicato al primario.

a
b

FIGURE 6.9
(a) Generating
the RGB image of
the cross-sectional
color plane
(127, G, B).
(b) The three
hidden surface
planes in the color
cube of Fig. 6.8.



Il color space RGB è in realtà un cubo, dove la profondità è quanto chiaro/scuro è.

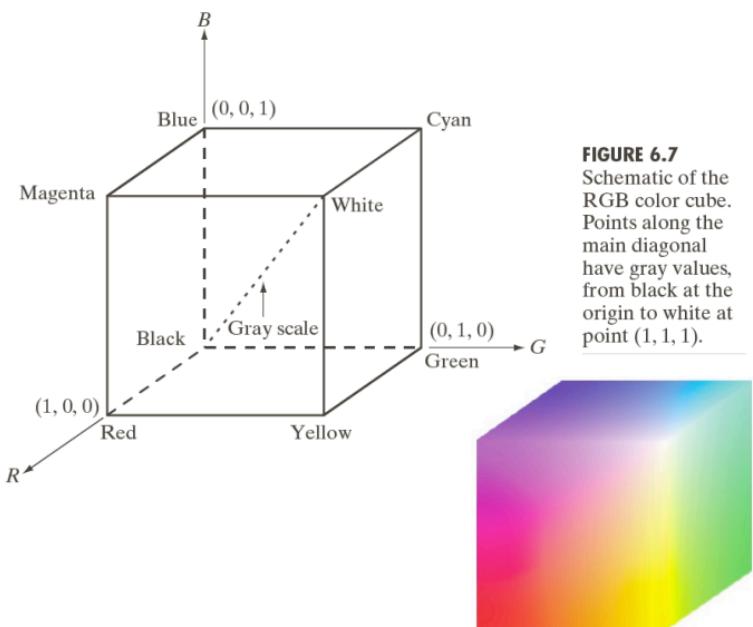
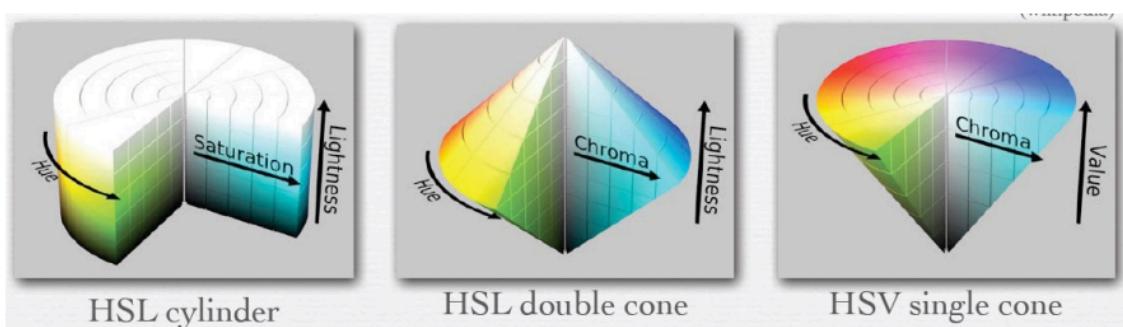


FIGURE 6.7
Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).

FIGURE 6.8 RGB 24-bit color cube.

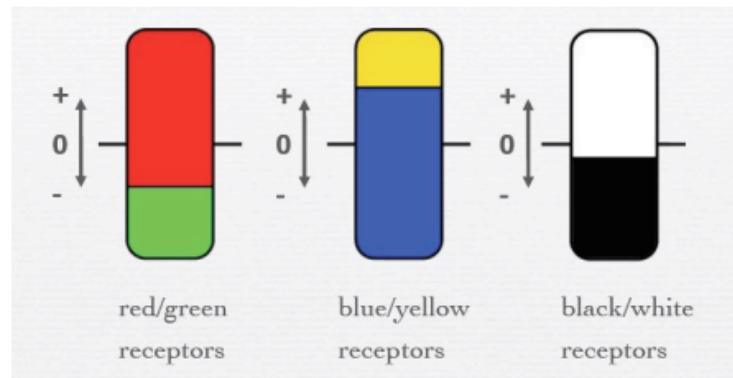
Di solito per ogni spazio di colore abbiamo 3 informazioni:

- Hue: qual è il colore dominante (es rosso o giallo), è di solito rappresentato da un'informazione a cerchio nei grafici.
- Saturazione: quanto siamo lontani da un grigio neutrale. è la coordinata radiale sulla color wheel. Colori al centro sono non saturati, colori ai bordi sono saturati.
- Intensity: quanta luce c'è, quanto è chiaro o scuro il colore.



La hue è l'informazione a cerchio di ciascuna immagine, la saturazione è quanto sei lontano dal centro e l'intensità è su/giù. Questo è giusto per vedere come abbiamo rappresentazioni diverse per diversi spazi di colore.

Opponent colors



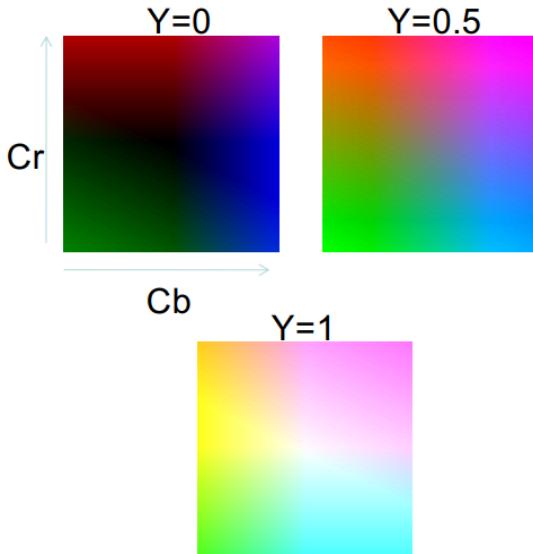
Hering ha osservato che gli esseri umani non vedono colori rosso-verdi o blu-gialli, e ha ipotizzato tre recettori, come mostrato sopra.

Questo ci serve per la compressione delle immagini. Siamo più sensibili al cambio della luce rispetto al cambiamento in questi opponent channels.

Lo spazio di colore opponent più famoso è YCbCr. Abbiamo un canale con la luminosità e altri 2 canali con le informazioni in termini di opponent channels.

Color spaces: YCbCr

Fast to compute, good for compression, used by TV



Quindi sfruttiamo il fatto che siamo meno sensibili sugli altri 2 canali opponents, andando a comprimerli di più, facendo downsampling. Manteniamo invece l'informazione nel primo canale perché siamo più sensibili.

Se questo è fatto bene, non ci sarà una differenza percepita.

The "Y" Color space

- YIQ, YUV,...
- Used in television sets and videos;
 - Y is the color luminance;
 - I and Q represent the color chromaticity;
- YUV corresponds to PAL, YIQ corresponds to NTSC;
- YC_bC_r makes sense in image compression.



FIGURE 6.51
Color image compression.
(a) Original RGB image. (b) Result of compressing and decompressing the image in (a).

Questo è lo spazio colore YIQ, anche questo è un opponent color space.

Nella seconda immagine, l'informazione nel canale grayscale è stata sfocata. Vediamo che qui c'è una grossa perdita di qualità percepita. Se facciamo la stessa cosa negli altri due canali invece, non abbiamo la stessa percezione, al massimo i colori sono leggermente diversi in alcune aree.

NTSC color television
color space is YIQ

Y = luminance
I = orange-blue axis
Q = purple-green axis

