

Elaborazione delle Immagini

Sara Angeretti

@Sara1798

Fabio Ferrario

@fefabo

2023/2024

Indice

1	Percezione, acquisizione, digitalizzazione	5
2	Operatori Puntuali	6
2.1	Gli Operatori sulle Immagini	6
2.1.1	Introduzione agli Operatori Locali	7
2.1.2	Gli operatori Puntuali	7
2.2	Alcuni Operatori Puntuali di base	9
2.2.1	Contrast Stretching	9
2.2.2	Thresholding o Binarizzazione	10
2.2.3	I Negativi	10
2.2.4	Gamma Correction	10
2.2.5	Look-Up Table	11
2.3	L'Istogramma	12
2.3.1	Equalizzazione dell'istogramma	14
3	Filtraggio Spaziale	20
3.1	Meccaniche del Filtraggio Spaziale Lineare	20
3.2	Correlazione e Convoluzione Spaziale	22
3.3	Filtri di Smoothing	22
3.3.1	Box Filter Kernels	23
3.3.2	Filtro Gaussiano	23
4	Immagini a Colori	26
4.1	Spazi Colore	26
4.1.1	YCbCr	26
4.1.2	HSV	27
5	Analisi e Classificazione delle Immagini	28
5.0.1	Segmentazione	29

Il corso

Il corso di Elaborazione delle Immagini 2023-2024 è erogato dai professori **Schettini** Raimondo e **Ciocca** Gianluigi.

Programma del corso

Durante il corso lo studente acquisirà competenze specifiche che lo porranno in grado di comprendere la catena di elaborazione, analisi e classificazione di immagini e video. Lo studente acquisirà inoltre le competenze necessarie per progettare, sviluppare ed integrare specifici moduli in sistemi applicativi complessi.

Competenze per cui questo corso prepara: esempi

Image Recognition in the Consumer Goods and Services Industry (supported by Accenture). Per esempio, una certa marca di prodotto paga un costo aggiuntivo al proprio valore per essere venduto ad altezza occhi per 10-12 metri di scaffale, mentre il prodotto sull'ultimo scaffale ha meno costi aggiuntivi. Parole sue: ***meglio essere quello che automatizza di quello che viene automatizzato.***

Programma Esteso

1. Cenni sulla percezione visiva, la visione umana e artificiale, il colore. Acquisizione e digitalizzazione di immagini.
2. Miglioramento delle immagini con operatori puntuali.
3. Filtraggio spaziale lineare e non-lineare.
4. Spazi colore. Elaborazione delle immagini a colori.
5. Segmentazione di immagini per regioni e per contorni.

6. Analisi tessiturale; Morfologia Matematica.
7. Descrizione e rappresentazione di immagini (regioni, contorni, approssimazione poligonale).
8. Riconoscimento, classificazione supervisionata e non supervisionata.
9. Introduzione alle reti neurali convoluzionali profonde.

Libri di testo

- R. Gonzalez, R. Woods, Digital Image Processing, Second Edition, 2002. Prentice Hall.
- R. Gonzalez, R. Woods, Digital Image Processing, Third edition, 2008 Prentice Hall. <http://www.imageprocessingplace.com/>
- R. Gonzalez, R. Woods, Elaborazione Digitale delle Immagini, terza edizione (utile per chi ha difficoltà con l'inglese) non ha tutti i capitoli.
- R. Gonzalez, R. Woods, Digital Image Processing, forth edition, 2018 Prentice Hall.

Si sconsiglia il libro tradotto in italiano perchè mancano alcuni capitoli.

Capitolo 1

Percezione, acquisizione, digitalizzazione

Capitolo 2

Operatori Puntuali

Tutte le tecniche di Image Processing che discuteremo in questo capitolo sono implementate nel **Dominio Spaziale**, ovvero il piano contenente i **pixel** di un'immagine. Le tecniche che introdurremo operano direttamente su i pixel, a differenza dell'elaborazione nel dominio delle Frequenze, nel quale le operazioni sono effettuate sulla trasformata di Fourier di un'immagine invece che sull'immagine stessa.

Enhancement Anche se i metodi di Filtraggio Spaziale¹ e Intensity Transformation² hanno moltissime applicazioni, noi ci concentreremo sull'**Enhancement** delle immagini.

L'Enhancement è un operazione di Image Processing a basso livello possibile sia nel dominio spaziale che nel dominio delle Frequenze ed è il processo di manipolazione di un'immagine per far sì che il risultato sia migliore dell'originale per una **specifica applicazione**. Diamo enfasi alla parola *specifica*, dato che le tecniche di enhancement sono **problem oriented**.

2.1 Gli Operatori sulle Immagini

Nel miglioramento (enhancement) delle immagini si possono applicare diversi operatori:

- **Operatori Puntuali**, il valore di ogni pixel dell'immagine di uscita è in funzione solo del corrispondente pixel dell'immagine di ingresso.

¹Spacial Filtering o Operatori Locali

²Operatori Puntuali

- **Operatori Locali**, il valore di ogni pixel dell'immagine di uscita è in funzione dei valori del pixel di ingresso corrispondente e *di un suo intorno locale di pixel*.
- **Operatori Globali**, i pixel dell'immagine di uscita dipendono dal valore di tutti i pixel dell'immagine di ingresso.

2.1.1 Introduzione agli Operatori Locali

I processi nel dominio spaziale che affrontiamo si basano sull'espressione:

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

Dove $f(x, y)$ è l'immagine di input, $g(x, y)$ è l'immagine di output e T è un operatore di f definito su un *intorno* del punto (x, y) .

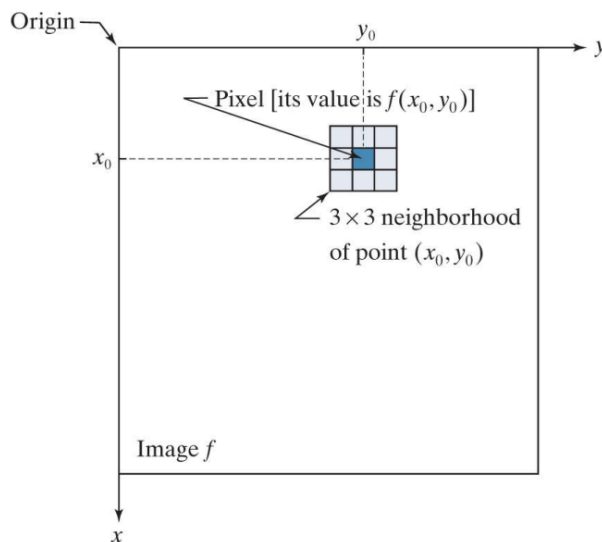


FIGURE 3.1
A 3×3 neighborhood about a point (x_0, y_0) in an image. The neighborhood is moved from pixel to pixel in the image to generate an output image. Recall from [Chapter 2](#) that the value of a pixel at location (x_0, y_0) is $f(x_0, y_0)$, the value of the image at that location.

Tipicamente l'intorno (neighborhood) è rettangolare e centrato sul pixel (x_0, y_0) ed è molto più piccolo dell'immagine, però la sua forma e dimensione *dipendono dall'applicazione*. Nota che T può anche essere applicato su una sequenza di immagini, per esempio per effettuare una somma.

2.1.2 Gli operatori Puntuali

L'intorno più piccolo possibile è di dimensione 1×1 , in questo caso T considera solo il valore di (x, y) , diventando di fatto un **Operatore Puntuale** o

Intensity Transformation Function (Manipolazioni della scala di grigio) di forma:

$$s = T(r)$$

in cui, per semplicità di notazione, usiamo s e r per denotare rispettivamente l'intensità di g (output) e f (input) di un qualunque punto (x, y) .

Elaborazioni Puntuali Omogenee Il risultato di una elaborazione puntuale omogenea dipende solo dal valore del pixel cui è applicata, per cui tali elaborazioni vengono anche dette manipolazioni della scala di grigio (o dei colori).

Se invece il risultato dell'elaborazione dipende anche dalla posizione del pixel dell'immagine si parla di elaborazioni puntuali **non omogenee**.

Alcune tipiche elaborazioni puntuali omogenee sono:

- Aggiunta o sottrazione di una costante a tutti i pixel.
- Inversione della scala di grigi (negativo).
- Clipping.
- Modifiche del contrasto (trasformazioni lineari e non della scala dei grigi).
- Equalizzazione dell'istogramma.
- Presentazione falso in colore.

Operatori Puntuali Lineari

Come abbiamo già detto un operatore puntuale calcola il valore di un pixel dell'immagine di uscita in funzione esclusivamente del valore del corrispondente dell'immagine di ingresso.

Un operatore puntuale h applicato all'immagine $f(n)$ è una funzione che è applicata identicamente ad ogni pixel creando una nuova immagine $g(n)$. Se l'operatore puntuale in questione è della forma:

$$g(n) = Pf(n) + L$$

Dove P è un fattore di scala moltiplicativo e L un fattore di offset, allora è chiamato **Operatore Puntuale Lineare**.

2.2 Alcuni Operatori Puntuali di base

Le funzioni di Intensity Transformation sono tra le più semplici tecniche di IP. In queste funzioni i valori di un pixel viene prima e dopo la trasformazione, che chiamiamo rispettivamente r ed s , sono messi in relazione tramite la funzione T che mappa i valori r su quelli s .

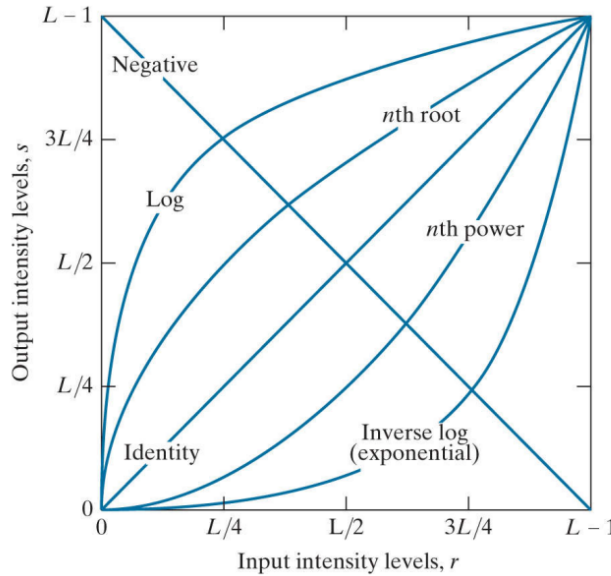


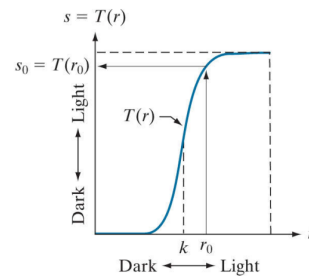
FIGURE 3.3
Some basic intensity transformation functions. Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.

Look Up Tables Siccome stiamo parlando di grandezze digitali, i valori di una Intensity Transformation Function sono generalmente mappati in una Look Up Table, che permette di associare con migliore efficienza i valori di input e output.

2.2.1 Contrast Stretching

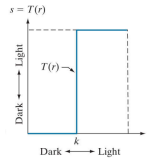
Un esempio di elaborazione puntuale omogenea è quello del *contrast stretching*. In questo caso $T(r)$ prende la forma come nell'immagine.

Il risultato dell'applicazione della trasformazione ad ogni pixel in f per generare il corrispondente pixel in g sarebbe quello di produrre una **immagine di contrasto maggiore** dell'originale, scurendo i livelli di intensità sotto a k e schiarendo le intensità sopra a k .



2.2.2 Thresholding o Binarizzazione

In quest'altro esempio $T(r)$ produce un'immagine binaria, ponendo a 0 (nero) tutti i livelli sotto k e a 1 (bianco) tutti quelli sopra a k .



2.2.3 I Negativi

Il negativo di un'immagine con livelli di intensità nel range $[0, L - 1]$, con L numero di livelli (256 per una immagine in 8bit), è ottenuto usando la *Funzione di trasformazione Negativa*, che ha la forma:

$$s = L - 1 - r$$

Invertendo in valori di grigio di un'immagine digitale in questa maniera produce l'equivalente di un negativo fotografico.

2.2.4 Gamma Correction

La trasformazione potenza, anche detta **Gamma Correction**, è uno degli operatori puntuali più utilizzati. Questa trasformazione è nella forma:

$$s = cr^\gamma$$

Dove c e γ sono costanti positive. Le curve power-law con valori frazionari di γ mappano un range ristretto di valori scuri in ingresso in una gamma più ampia di valori in uscita, facendo l'opposto per i valori più chiari.

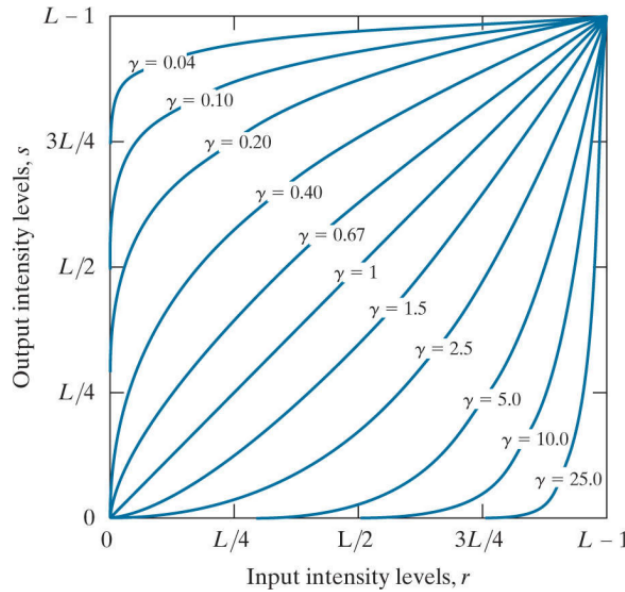


FIGURE 3.6
Plots of the gamma equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases). Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.

In parole povere Quando facciamo una Gamma Correction con un valore di $\gamma < 1$, quindi frazionario, stiamo "stretchando" gli scuri restringendo i chiari, viceversa se $\gamma > 1$. Se $\gamma = 1$ invece, la funzione risultante sarà quella identità.

2.2.5 Look-Up Table

Diverse trasformazioni con diverse espressioni analitiche hanno dei costi computazionali variabili.

Per una generica immagine 1024×1024 , un operatore puntuale applicato alla sola intensità richiederebbe più di un milione di volte la sequenza:

1. Conversione intero-float.
2. Calcolo dell'operatore.
3. Conversione float-intero

Questa implementazione è tutt'altro che efficiente, quindi si ricorre spesso alle **Look Up Table (LUT)**. Una LUT è una tabella con tante entry quante sono i possibili valori di ingresso all'operazione. In questa tabella viene associato ad ogni valore possibile di ingresso ($[0,255]$) un valore a cui è stato già applicato l'operatore in questione che verrà messo nell'immagine di Output.

2.3 L'Istogramma

L'istogramma di un'immagine è un tipo di istogramma che rappresenta in modo grafico la **distribuzione tonale** (livelli di grigio) **di un'immagine digitale** ovvero traccia il numero di pixel per ogni valore tonale.

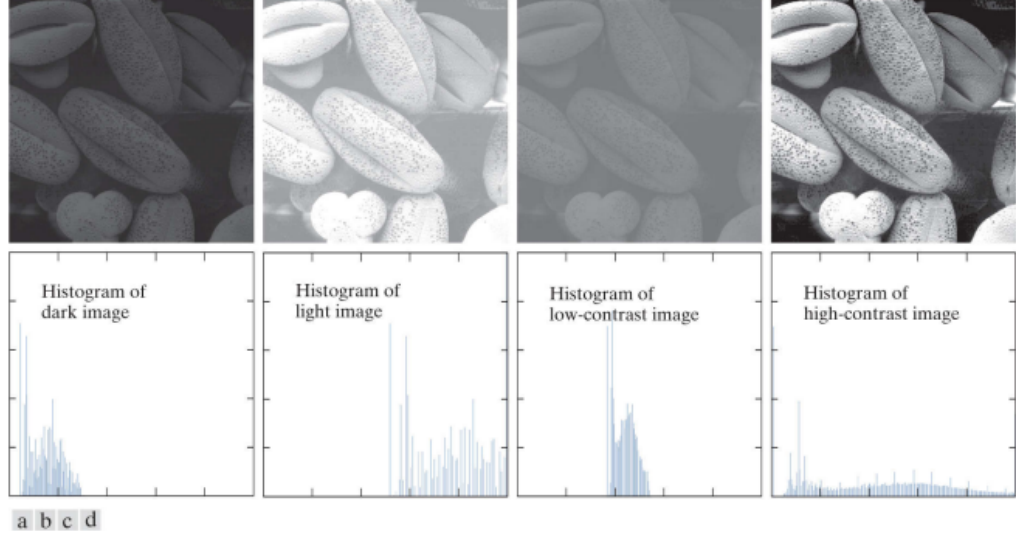


FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

L'analisi dell'istogramma fornisce generalmente utili informazioni sulle proprietà dell'immagine legate alle frequenze dei livelli di grigio. L'informazione spaziale è persa, quindi da un istogramma non possiamo risalire alla posizione dei pixel nell'immagine.

Definizione Matematica Abbiamo r_k , per $k = 0, 1, \dots, L - 1$ che denota le intensità di un'immagine $f(x, y)$ a L -livelli. L'istogramma **Non Normalizzato** di f è definito come:

$$h(r_k) = n_k \text{ for } k = 0, 1, \dots, L - 1$$

Dove n_k è il numero di pixel in f con intensità r_k , e la suddivisione delle scale di intensità sono chiamate Histogram Bins.

Istogramma Normalizzato L'istogramma normalizzato di f è definito come:

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN}$$

Dove M e N sono rispettivamente il numero di righe e colonne di un'immagine. L'istogramma normalizzato è come l'istogramma, ma elimina la considerazione della dimensione dell'immagine poichè la somma dei valori $p(r_k)$ per ogni valore di k è sempre 1. In genere lavoreremo con gli istogrammi normalizzati, che chiameremo semplicemente istogrammi di immagine.

Cosa mi dice l'istogramma normalizzato? L'istogramma normalizzato rappresenta la *frequenza di occupazione* dei livelli ed è interpretata come *stima della probabilità di avere il valore k nell'immagine*. Gli istogrammi sono facili da computare e sono utilizzabili per rapide implementazioni hardware, rendendo le tecniche basate sugli istogrammi uno strumento molto utilizzato per l'Image Processing in Real Time.

2.3.1 Equalizzazione dell'istogramma

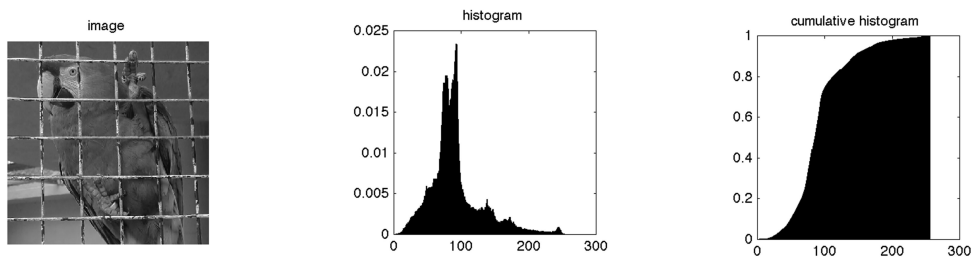
L'equalizzazione dell'istogramma è una tecnica di image enhancement che tende ad **aumentare il contrasto dell'immagine** mutando la forma dell'istogramma per bilanciare meglio i livelli, mantenendone però l'ordinamento. Questa tecnica sfrutta il concetto alla base del contrast stretching, ovvero utilizzare più livelli possibili per accentuare il contrasto. La sua comodità è il fatto che **non occorre specificare alcun parametro**, motivo che la rende una tecnica molto popolare, ma spesso abusata. Per poter parlare di equalizzazione dell'istogramma dobbiamo prima introdurre l'istogramma Cumulativo.

Istogramma Cumulativo Abbiamo già introdotto l'istogramma normalizzato $P(k) = \frac{H(k)}{N}$ che rappresenta la frequenza di occupazione dei livelli, ovvero la stima della probabilità di avere il valore k , Introduciamo ora l'istogramma cumulativo $A(k)$ così definito:

$$A(k) = \sum_{j \leq k} P(j)$$

In cui la somma $\sum_L P(j) = 1$ con N numero di livelli.

Questo istogramma indica la percentuale di pixel con livello di grigio $\leq k$, ovvero la stima della probabilità di avere un valore minore o uguale a k .



Contrast stretching VS Equalizzazione Il contrast stretching tende a stirare l'istogramma senza mutarne la forma, mentre l'equalizzazione tende a mutare la forma dell'istogramma in modo da ottenere una distribuzione il più possibile **a densità costante**. Cerchiamo quindi di definire un'istogramma che mantenga l'ordine dei livelli di grigio, utilizzando una trasformazione strettamente monotona, che sia *il più piatto possibile* e sia definito sullo stesso range di valori, oltre che essere completamente automatica.

Il funzionamento dell'equalizzazione

Supponendo valori di intensità continui, denotiamo con la variabile r le intensità di un'immagine che deve essere processata, e assumiamo che r sia nel range $[0, L - 1]^3$ con $r = 0$ che rappresenta il colore nero e $r = L - 1$ che rappresenta il colore bianco.

Per r che soddisfa queste condizioni, poniamo l'attenzione su trasformazioni (intensity mappings) della forma:

$$s = T(r) \quad 0 \leq r \leq L - 1$$

Che produce un valore di intensità di output s per ogni valore r nell'immagine di input.

Assumiamo che:

1. $T(r)$ sia una funzione **Monotona Strettamente Crescente** nell'intervallo $[0, L - 1]$
2. $0 \leq T(r) \leq L - 1$ per $0 \leq r \leq L - 1$, ovvero $T(r)$ deve essere sempre dentro l'intervallo.

In alcune formulazioni che verranno discusse tra poco, useremo la trasformazione inversa:

$$r = T^{-1}(s) \quad 0 \leq s \leq L - 1$$

Che esattamente il motivo per cui è richiesto che $T(r)$ sia Strettamente Monotona, come dimostrato nella seguente immagine.

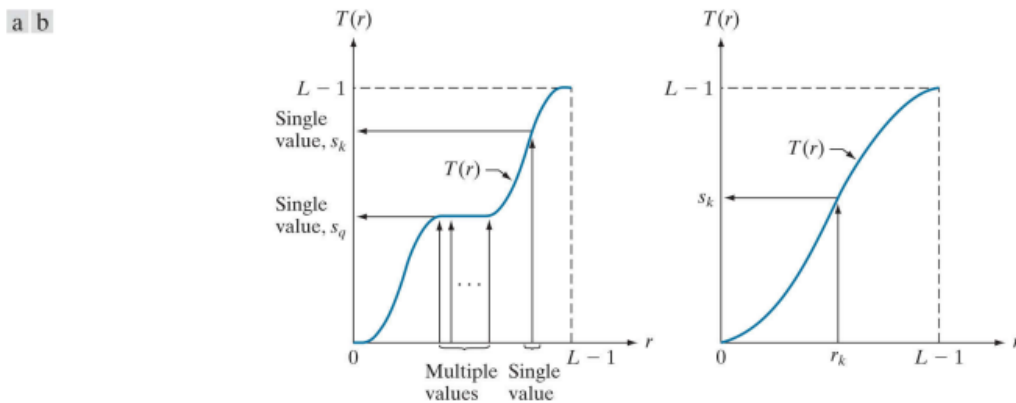


FIGURE 3.17

(a) Monotonic increasing function, showing how multiple values can map to a single value. (b) Strictly monotonic increasing function. This is a one-to-one mapping, both ways.

³ r può anche essere normalizzato $r \in [0, 1]$

Una funzione non strettamente monotona andrebbe bene quando stiamo mappando da r a s , ma diventa problematica quando vogliamo recuperare i valori di r unicamente dai valori mappati.

L'intensità di un'immagine (quindi r e s) possono essere viste come variabili random nell'intervallo $[0, L-1]$, ed è possibile caratterizzarle mediante i rispettivi PDF⁴:

$$p_r(r) \text{ e } p_s(s)$$

Dalla teoria della probabilità sappiamo che se $p_r(r)$ e $T(r)$ sono **note** e T^{-1} è continua e differenziabile nel range, allora il PDF della variabile mappata s può essere ottenuto tramite:

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| \quad (3.10)$$

Difatti, dati gli intervalli $[r, r+dr]$ e $[s, s+ds]$, tutti i punti $[r, r+dr]$ devono essere trasformati in punti di $[s, s+ds]$. Ne segue che le aree, ovvero le probabilità, devono essere uguali.

Una funzione trasformazione particolarmente importante in IP è:

$$s = T(r) = (L-1) \int_0^r p_r(w) dw \quad (3.11)$$

Dove w è una *dummy variable* di integrazione. L'integrale sulla destra è la **Funzione di Distribuzione Cumulativa** (CDF) di r . Siccome i PDF sono sempre positivi, e l'integrale di una funzione è l'area sottesa, ne segue che questa funzione soddisfa la condizione (1), questo è perchè l'area sotto la funzione non può diminuire all'incrementare di r . Quando il limite superiore in questa equazione è $r = (L-1)$, l'integrale diventa 1, come deve per un PDF, rendendo il valore massimo di s $(L-1)$, rispettando la seconda condizione.

Derivando s rispetto a r si ottiene, per la regola di Leibniz⁵:

$$\begin{aligned} \frac{ds}{dr} &= \frac{dT(r)}{dr} \\ \rightarrow \frac{ds}{dr} &= (L-1) \cdot \frac{d}{dr} \left[\int_0^r p_r(w) dw \right] \\ \rightarrow \frac{ds}{dr} &= (L-1) \cdot p_r(r) \end{aligned}$$

⁴Probability Density Function, ovvero l'istogramma

⁵The derivative of a definite integral with respect to its upper limit is the integrand evaluated at the limit

Sostituendo questo risultato per $\frac{dr}{ds}$ ⁶ nell'equazione (3.10), e tenendo conto del fatto che tutti i valori di probabilità sono positivi⁷, da il risultato:

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

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

$$\rightarrow p_s(s) = \frac{1}{L-1} \quad 0 \leq s \leq L-1$$

La densità di probabilità (PDF) della variabile trasformata è uniforme nel suo intervallo di definizione, *indipendentemente* dalla forma di $p_r(r)$. Da questo ne deriva un incremento nella gamma dinamica dei pixel che può avere notevoli effetti sulla "apparenza" dei pixel.

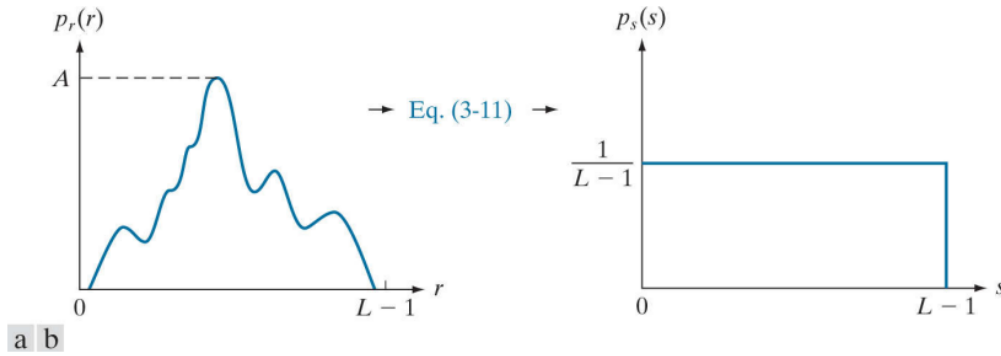


FIGURE 3.18

(a) An arbitrary PDF. (b) Result of applying Eq. (3-11) to the input PDF. The resulting PDF is always uniform, independently of the shape of the input.

Equalizzazione nel caso Discreto

Consideriamo ora il caso in cui stiamo lavorando con valori discreti, ovvero in \mathbb{N} , che è esattamente quello che succede in una immagine digitale.

Interpretiamo i livelli di grigio come una variabile aleatoria discreta. Avremo che la probabilità di occorrenza dei livelli di grigio in una immagine digitale è approssimata dalla seguente formula:

$$p_r(r_k) = \frac{n_k}{MN}$$

⁶ $\frac{ds}{dr} = (L-1) \cdot p_r(r) \implies \frac{dr}{ds} = \frac{1}{(L-1) \cdot p_r(r)}$

⁷ Permettendoci di rimuovere il valore assoluto

Dove MN è il numero totale di pixel nell'immagine, e n_k denota il numero di pixel che hanno intensità r_k . Come sappiamo $p_r(r_k)$ è l'istogramma normalizzato. La forma discreta della trasformazione in Eq 3.11, ovvero la funzione per l'equalizzazione dell'istogramma, diventa:

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) \quad k = 0, 1, 2, \dots, L - 1$$

Dove, come prima, L è il numero di livelli di grigio possibili nell'immagine. Quindi la trasformazione consiste nel trasformare ogni pixel di livello r_k in un pixel di livello s_k , attraverso la $T(r_k)$ che si calcola direttamente dall'istogramma dell'immagine di input.

Esempio: Illustrazione delle meccaniche dell'equalizzazione dell'istogramma.

Per capire come funziona l'equalizzazione dell'istogramma facciamo un esempio. Immagina di avere un'immagine a 3bit, quindi $L = 8$, di dimensione 64×64 ($MN = 4096$). Questa immagine ha una Tabella di distribuzione delle intensità come qui sotto:

TABLE 3.1
Intensity distribution and histogram values for a 3-bit, 64×64 digital image.

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

Notiamo che i valori di intensità sono nel range $[0, L - 1] = [0, 7]$. Consideriamo il primo livello r_0 , e il relativo valore di output s_k :

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

Si noti che l'istogramma p_r è quello normalizzato, e che $\sum_{j=0}^k p_r(r_j)$ equivale al valore k -esimo dell'istogramma cumulativo.

Capitolo 3

Filtraggio Spaziale

In questo capitolo discuteremo l'uso dei **Filtri Spaziali** nell'Image Processing. Il nome "filtro" è preso in prestito dal processing nel dominio delle frequenze, dove il filtraggio è il passaggio, modifica o bloccaggio di specifiche componenti di frequenze delle immagini, per esempio un filtro "passa basso" fa passare le basse frequenze risultando in uno smoothing delle immagini.

Differenze con operatori locali A differenza degli operatori locali, in cui il valore del pixel di uscita era calcolato solamente partendo dal valore del pixel di entrata, i filtri spaziali (anche chiamati operatori locali) modificano una immagine rimpiazzando il valore di ogni pixel in funzione del suo valore e di **quello dei suoi vicini**.

Se l'operazione effettuata sui pixel è lineare, allora il filtro è chiamato *filtro spaziale lineare*.

3.1 Meccaniche del Filtraggio Spaziale Lineare

Un filtro spaziale lineare effettua una **somma di prodotti** tra un'immagine f e un *kernel*¹ w . Il **Kernel** è un array la cui dimensione definisce la zona² dell'operazione, e il cui coefficienti determinano la natura del filtro.

L'immagine 3.34 illustra le meccaniche del filtraggio spaziale utilizzando una maschera 3×3 . In ogni punto (x, y) nell'immagine il pixel risultante $g(x, y)$ dal filtro è la somma dei prodotti dei coefficienti del kernel e i pixel

¹Filter kernel, anche detto Maschera, Template o Finestra.

²Neighborhood

compresi nel kernel:

$$g(x, y) = w(-1, -1)f(x-1, y-1) + w(-1, 0)f(x-1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 1)f(x+1, y+1)$$

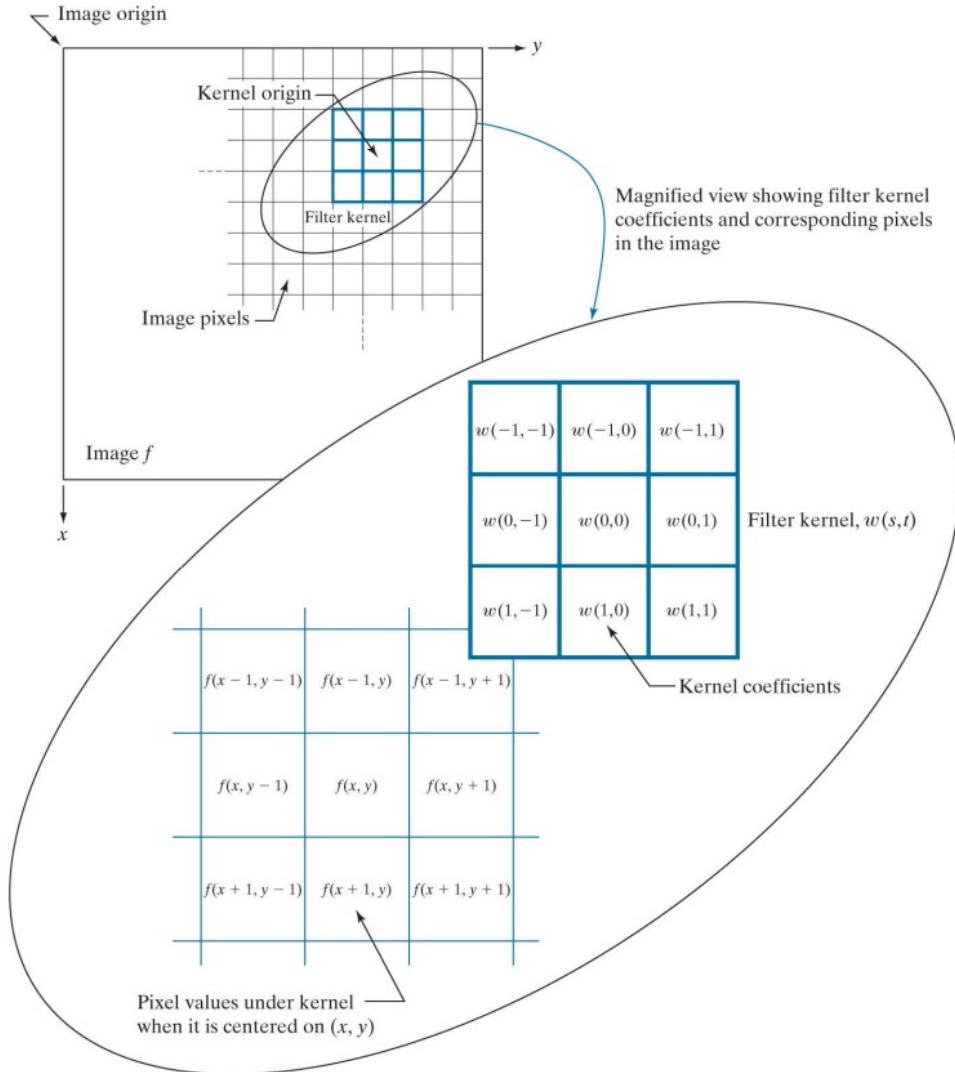


FIGURE 3.34

The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.

Al variare delle coordinate x e y , il centro del kernel si muove da pixel a pixel, generando l'immagine filtrata g .

La forma del Filtro Si osserva che il coefficiente centrale del kernel, $w(0, 0)$, è allineato al pixel di coordinata (x, y) . Per un kernel di dimen-

sione $m \times n$, assumiamo che $m = 2a + 1$ e $n = 2b + 1$, dove a e b sono interi non negativi. Questo significa che i Kernel sono di dimensione dispari in entrambe le direzioni.

In generale, il filtraggio spaziale di un'immagine di dimensione $M \times N$ e un kernel di dimensione $m \times n$ è dato dall'espressione:

$$g(x, y) = \sum_{s=-A}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (3.10)$$

Dove x e Y sono variati in modo che il centro (origine) del kernel visita ogni pixel in f una volta.

Osservazione: Un Kernel può essere pari? Sì, è possibile lavorare con kernel di dimensioni pari, o dimensioni miste. Però è molto più semplice lavorare con kernel dispari perchè l'indicizzazione è molto più facile e intuitiva.

3.2 Correlazione e Convoluzione Spaziale

La correlazione spaziale è illustrata graficamente nell'immagine in fig 3.34, ed è descritta matematicamente dall'equazione 3.31. La correlazione consiste in muovere il centro del kernel su una immagine e computando la somma dei prodotti ad ogni punto. Le meccaniche della Convoluzione sono le stesse, ma il kernel è ruotato di 180° .

Osservazione: Poichè la convoluzione ha lo stesso kernel della correlazione ma ruotato di 180° , se i valori del kernel sono simmetrici sul centro allora convoluzione e correlazione daranno lo stesso risultato.

3.3 Filtri di Smoothing

I filtri di Smoothing, anche detti filtri di **Media**, sono filtri spaziali utilizzati per ridurre transizioni nette di intensità. Siccome il rumore *random* generalmente consiste di transizione nette in intensità, un ovvio utilizzo dello smoothing è la **riduzione del rumore**.

Un'altra applicazione dello smoothing è la riduzione dei falsi contorni che appaiono quando campioniamo un'immagine con un livello insufficiente di intensità.

Filtraggio spaziale Lineare Il filtraggio spaziale lineare consiste nella convoluzione di un'immagine con un kernel. Fare la convoluzione di un'immagine con un kernel di un filtro di smoothing risulta nel *Blur* dell'immagine, con il livello di sfocatura determinato dalla dimensione del kernel e dai valori dei suoi coefficienti.

3.3.1 Box Filter Kernels

Il filtro spaziale lineare più semplice è il **box kernel**, in cui i coefficienti hanno sempre lo stesso valore (tipicamente 1). Il nome "box kernel" deriva dal fatto che il kernel sembra una scatola quando visualizzato in 3D.

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

Un box-filter di dimensione $m \times n$ è un array $m \times n$ con tutti 1 come coefficienti, con una costante di normalizzazione davanti, il cui valore è 1 diviso dalla somma dei valori dei coefficienti.

Normalizzazione Questa normalizzazione, che viene applicata a tutti i filtri di smoothing, ha due funzioni:

1. Il valore medio di un'area di intensità costante è di uguale intensità nell'immagine filtrata, come dovrebbe.
2. Normalizzare il kernel in questo modo previene l'introduzione di un Bias durante il filtraggio. ovvero, la somma dei pixel nell'immagine originale e in quella filtrata è lo stesso.

3.3.2 Filtro Gaussiano

Essendo estremamente semplici, i box-filters sono ottimi per rapide sperimentazioni e spesso danno risultati visivamente accettabili. Hanno però le loro

limitazioni, per esempio quando si vuole simulare la sfocatura di una lente il box filter è una approssimazione abbastanza povera delle sue caratteristiche. Inoltre, il box filtering favorisce il blurring lungo direzioni **perpendicolari**, quindi dove ci sono immagini con tanti dettagli o forti componenti geometriche, la direzionalità dei box filters spesso produce risultati indesiderabili. In quest'ultimo tipo di applicazioni i kernel scelti sono quelli **Circolarmente Simmetrici**.

Il Kernel Gaussiano Il kernel del filtro gaussiano è calcolato usando la seguente equazione:

$$G(r) = K e^{-\frac{r^2}{2\sigma^2}}$$

in cui $G(r)$ è il coefficiente del elemento a distanza r dal centro all'interno del Kernel, e K e σ sono delle costanti che vengono scelte in base all'intensità del filtraggio di cui abbiamo bisogno.

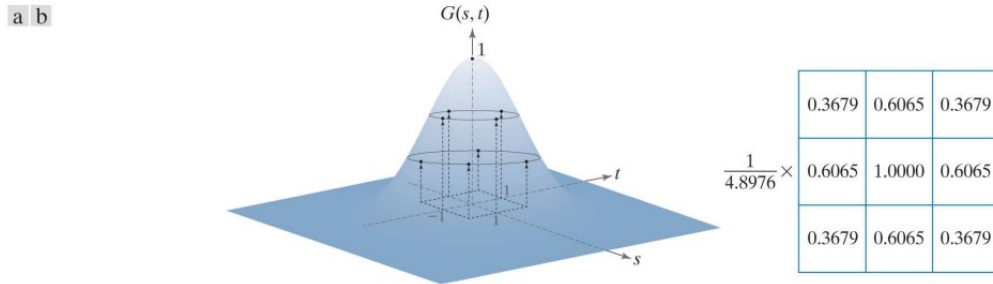


FIGURE 3.41

(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for $K = 1$ and $\sigma = 1$. (b) Resulting 3×3 kernel [this is the same as Fig. 3.37(b)].

Distanza dal centro Giusto per curiosità, questa è la tabella in cui sono riportate le varie distanze dal centro per i filtri quadrati. Nel calcolo dei coefficienti del filtro gaussiano usiamo, per ogni coefficiente, questa distanza per calcolare il valore del coefficiente.

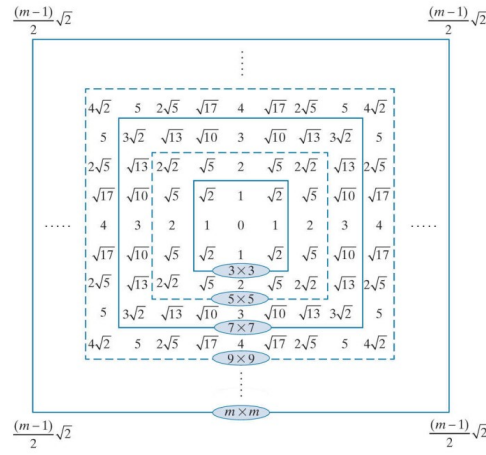


FIGURE 3.40
Distances from the center for various sizes of square kernels.

Dimensione del Kernel La dimensione del kernel gaussiano non deve essere arbitraria, perchè se abbiamo un kernel troppo grosso per il sigma che gli abbiamo assegnato sprechiamo energia computazionale. Un filtro molto grande e uno più piccolo con lo stesso sigma daranno lo stesso risultato.

La dimensione del filtro N e la σ sono quindi legati da questa regola empirica:

$$N = 1 + 2 \cdot \lfloor 2.5 \cdot \sigma \rfloor$$

Capitolo 4

Immagini a Colori

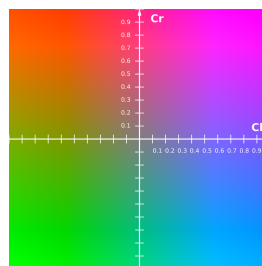
4.1 Spazi Colore

4.1.1 YCbCr

Lo spazio color YCbCr è uno spazio orientato alla **compressione e trasmissione**. Questo è uno spazio colore **Opponente**, ovvero invece che rappresentare "l'intensità" di un colore (come percepito dai filtri) rappresenta in un asse il confronto tra due colori opposti. Lo spazio YCbCr ha le seguenti componenti:

- Y = Intensità luminosa pura
- Cb = Asse Blu - Giallo
- Cr = Asse Rosso - Verde

Ignorando la Luma (Y), la rappresentazione dei colori è così composta:



YCbCr è utile perchè possiamo processare la pura informazione colore indipendentemente dall'intensità (Luma).

N.B. In MATLAB la funzione `imshow` visualizza solo immagini in scala di grigio o `rgb`, se gli diamo una immagine con uno spazio colore diverso cercherà di interpretare in 3 canali come R, G, e B.

4.1.2 HSV

N.B. Al progetto non è il caso di usare HSV o YCbCr, ma sarebbe bene trovarne un altro

Capitolo 5

Analisi e Classificazione delle Immagini

Il pattern recognition che affronteremo noi assume che le condizioni siano abbastanza controllate.

Classificazione: ho una base di dati (per esempio carte di uno) e devo dire quale carta è. Riconoscimento e classificazione sono processi diversi: riconoscimento riconosce un oggetto specifico (tipo una persona specifica), nella classificazione riconosco una classe di oggetti.

Il riconoscimento di forme e oggetti nelle immagini non si può basare su elementi puntuali, locali o globali, ma deve basarsi su come questi attributi locali si combinano tra loro e vengono interpretati dall'essere umano. L'analisi delle immagini può essere fatta a diversi livelli: Basso, medio e alto.

Analisi di Basso Livello Nell'analisi di Basso livello abbiamo in ingresso le immagini, e in uscita avremo un insieme di matrici, ognuna delle quali descrive un particolare attributo rilevato sulla scena.

Dal punto di vista computazionale:

- è spesso indipendente dal tipo di applicazione finale.
- Si basa su modelli statistici legati a proprietà locali.
- è spesso legata al processo di riduzione dei disturbi e le distorsioni del processo di acquisizione e trasmissione.
- In genere prepara per un'elaborazione successiva

Questo tipo di analisi è una trasformazione del tipo:

Immagine \rightarrow Immagine

Analisi di Livello Intermedio La descrizione per pixel della scena è troppo complessa per essere utile in applicazioni pratiche. Per ottenere una descrizione significativa il primo passo è la **Segmentazione**, in cui associamo dei descrittori ad ogni segmento.

Questo tipo di analisi è una trasformazione del tipo:

$$\text{Immagine} \rightarrow \text{Parametri}$$

Otteniamo quindi delle informazioni quantificabili dall'immagine attraverso la segmentazione.

Analisi di Alto Livello Questo tipo di analisi viene definita di alto livello in analogia alla visione umana e spesso impiega tecniche di intelligenza artificiale.

La trasformazione in questo caso è di tipo:

$$\text{Parametri} \rightarrow \text{Simboli}$$

Ha quindi in ingresso dei descrittori e in uscita una **Classificazione** o interpretazione.

5.0.1 Segmentazione

La segmentazione è quindi l'individuazione di regioni omogenee. Ci sono moltissimi approcci per segmentare un'immagine.