

Appunti di Elaborazione di Segnali Multimediali

a.a. 2017/2018

Elaborazione di immagini nel dominio spaziale

L. Verdoliva

In questa prima parte del corso studieremo le principali elaborazioni che possono essere effettuate sulle immagini digitali, focalizzando l'attenzione su quelle che operano direttamente sui campioni dell'immagine (*dominio spaziale*). Molte di queste elaborazioni saranno presentate attraverso diversi esempi in cui si mostra come migliorare un'immagine affinché risulti più adeguata per una specifica applicazione (*Enhancement*). L'Enhancement ha come obiettivo quello di produrre un'immagine in modo da migliorare la visualizzazione oppure consentire un'analisi più corretta da parte dell'utilizzatore, di solito accentuando o riducendo caratteristiche dell'immagine, come i bordi e i contrasti. È importante sottolineare che la soluzione è fortemente dipendente dal problema considerato: un metodo utile per un certo tipo di immagini potrebbe non essere valido per altre. La difficoltà maggiore risulta quindi proprio quella di “quantificare” un giusto criterio per l'enhancement, e in questo risiede il motivo per cui la maggior parte di queste tecniche sono empiriche e richiedono procedure interattive per ottenere risultati soddisfacenti. Se, per esempio, l'obiettivo finale è l'interpretazione visiva, è l'osservatore a dover stabilire la qualità dell'immagine, che risulta pertanto fortemente soggettiva.

1 Le immagini digitali

Tutte le elaborazioni che studieremo in questo capitolo possono essere definite su immagini multidimensionali, come il segnale video. Tuttavia per semplicità considereremo solo immagini bidimensionali, restringendo l'attenzione a quelle monocromatiche, costituite quindi dalla sola informazione di *luminanza*, che fornisce indicazioni sul livello di luminosità dell'immagine.

L'immagine digitale risulta dalla discretizzazione di un'immagine spazio-continua, che viene campionata nello spazio e i cui valori sono quantizzati su un numero finito K di livelli. La frequenza di campionamento usata determina la *risoluzione spaziale*, cioè il più piccolo dettaglio che è possibile discriminare, mentre la qualità della quantizzazione determina la *risoluzione dei livelli di grigio*, che rappresenta invece il più piccolo cambiamento di luminosità presente nell'immagine. In figura 1 si mostrano due esempi di immagini a diversa risoluzione spaziale, l'immagine a sinistra, a causa dell'elevata risoluzione, dà la sensazione di un'immagine “analogica”, mentre quella a destra è campionata con frequenza otto volte più piccola per cui si notano facilmente i “blocchetti” che rappresentano l'informazione di livello di grigio associata ad ogni pixel. In figura 2 invece si mostrano due esempi di immagini quantizzate con un numero diverso di livelli, in particolare l'immagine a sinistra è stata quantizzata con $K = 32$ livelli, mentre quella a destra con $K = 4$ livelli.

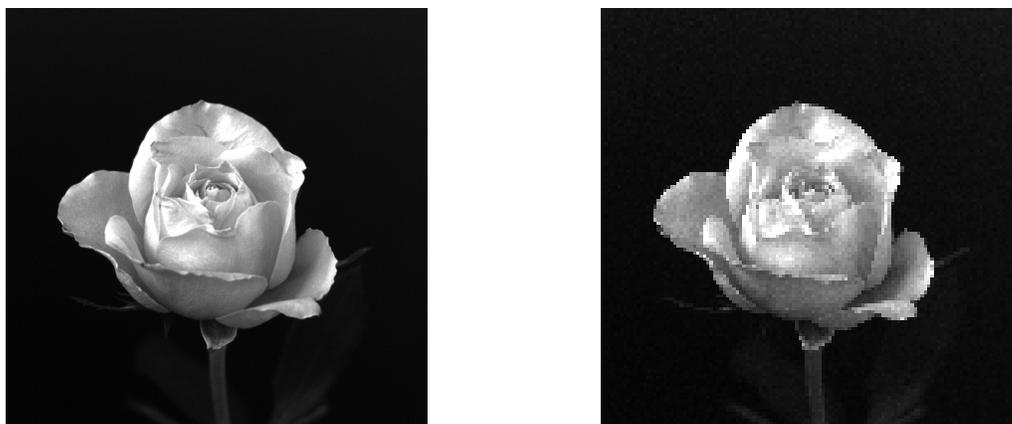


Figura 1: Esempi di immagini a diversa risoluzione spaziale.

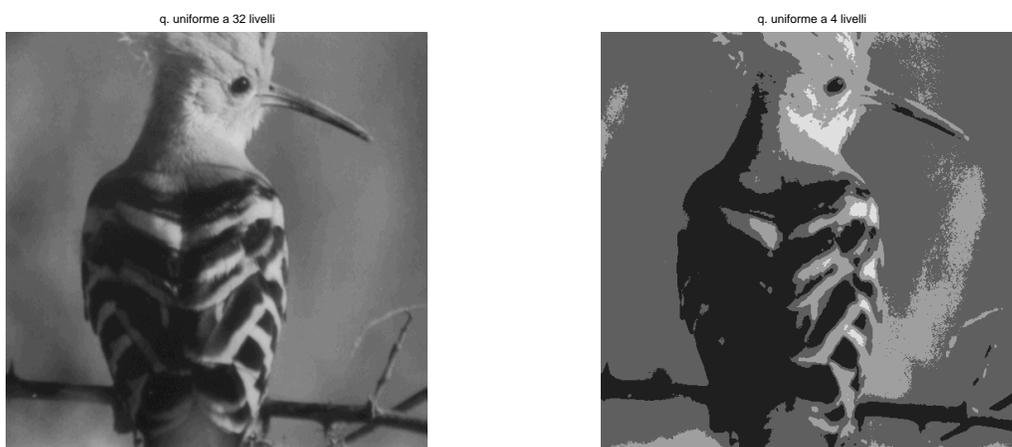


Figura 2: Esempi di immagini a diversa risoluzione dei livelli di grigio.

Il risultato del campionamento e della quantizzazione è una matrice di numeri, $x(m, n)$, definita su un supporto finito $[0, M - 1] \times [0, N - 1]$, e i cui elementi assumono valori nel range $[0, 1, \dots, K - 1]$ (*dinamica* dell'immagine). Il generico campione è anche chiamato *pixel* (picture element), e i valori assunti, spesso indicati con *livelli di grigio*, saranno rappresentati con $\log_2 K$ bit. La notazione utilizzata ci permette di scrivere l'immagine digitale $M \times N$ nella seguente forma matriciale:

$$x(m, n) = \begin{bmatrix} x(0, 0) & x(0, 1) & \dots & x(0, N - 1) \\ x(1, 0) & x(1, 1) & \dots & x(1, N - 1) \\ \dots & \dots & \dots & \dots \\ x(M - 1, 0) & x(M - 1, 1) & \dots & x(M - 1, N - 1) \end{bmatrix}$$

Spesso si considerano immagini con $K = 256$, cioè ogni campione è rappresentato con 8 bit/pixel. Si noti che, per semplicità, si è assunto che sia i valori delle coordinate che i valori quantizzati siano interi. In figura 3 si mostra il sistema di coordinate che useremo nel seguito per la rappresentazione di un'immagine digitale.

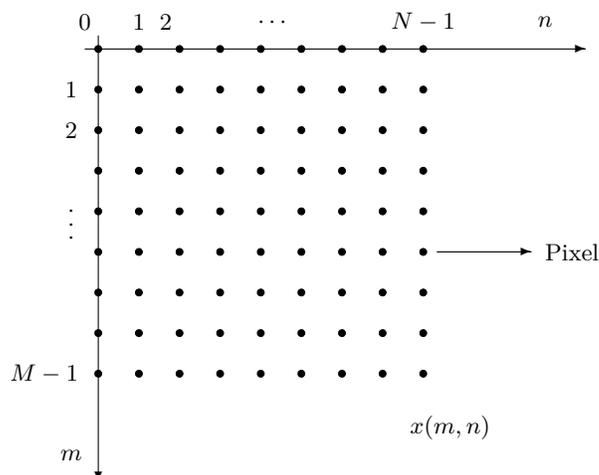


Figura 3: Sistema di coordinate per la rappresentazione di un'immagine.

Un modo per caratterizzare un'immagine digitale è attraverso il suo *istogramma*. L'istogramma $h_x(k)$ di un'immagine $x(m, n)$ è un grafico in cui è rappresentata la frequenza di occorrenza di ogni livello di grigio. E' quindi una funzione monodimensionale definita in $[0, \dots, K - 1]$, i cui valori si estendono da zero al numero di pixel che costituisce un'immagine, $M \times N$. Possiamo allora definire l'istogramma come:

$$h_x(k) = n_k$$

se $x(m, n)$ contiene esattamente n_k valori del k -esimo livello di grigio. Il calcolo dell'istogramma è estremamente semplice: bisogna contare il numero di volte che un certo valore si ripete nell'immagine.

E' chiaro che a partire dall'istogramma non è possibile dire qual è l'immagine che l'ha generato, esistono infatti infinite immagini che condividono lo stesso istogramma. Tuttavia esaminandolo si può stabilire se sono presenti livelli di grigio più chiari o più scuri, e sebbene ciò si possa spesso percepire anche visivamente, l'occhio umano ha l'abilità di adattarsi ai cambiamenti globali di luminanza, che possono oscurare traslazioni nella distribuzione dei livelli di grigio. In effetti, l'istogramma ci permette proprio di determinare tale distribuzione.

Una caratterizzazione più sintetica dell'immagine è data dal valore assunto mediamente dai pixel, che rappresenta una misura della luminanza media globale, detta anche *densità ottica media* (AOD) o luminosità media:

$$AOD_x = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \quad (1)$$

Evidentemente questa quantità fornisce una stima del centro della distribuzione dei livelli di grigio data dall'istogramma, e può essere anche calcolata come:

$$AOD_x = \frac{1}{MN} \sum_{k=0}^{K-1} k h_x(k) \quad (2)$$

In figura 4 si mostra un'immagine (quantizzata su 8 bit) con il relativo istogramma, in cui si può notare che solo i valori compresi nell'intervallo $[90, 138]$ sono contenuti nell'immagine; poichè non sono stati utilizzati tutti i valori compresi nell'intervallo $[0, 255]$, la figura presenta un contrasto molto basso. Le immagini di figura 5 e 6 sono invece caratterizzate rispettivamente da valori molto scuri (prossimi allo zero) e valori molto chiari (prossimi a 255). L'istogramma ci rivela, in effetti, che la prima immagine probabilmente è stata sottoesposta, mentre la seconda sovraesposta, oppure che il processo di conversione analogico-digitale non è stato eseguito correttamente. Nel seguito di questo capitolo vedremo come alcune delle trasformazioni nel dominio spaziale permettano di modificare opportunamente l'istogramma dell'immagine per migliorarne la visualizzazione.

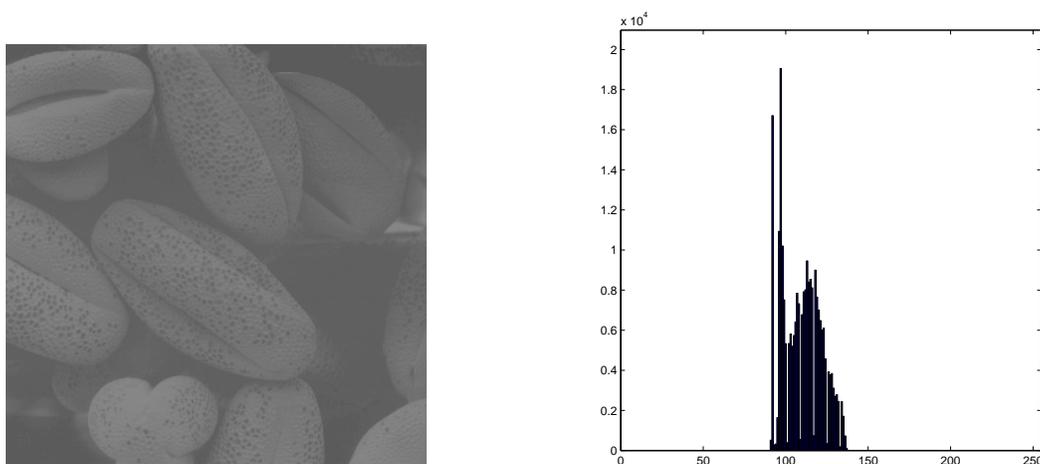


Figura 4: Immagine e relativo istogramma.

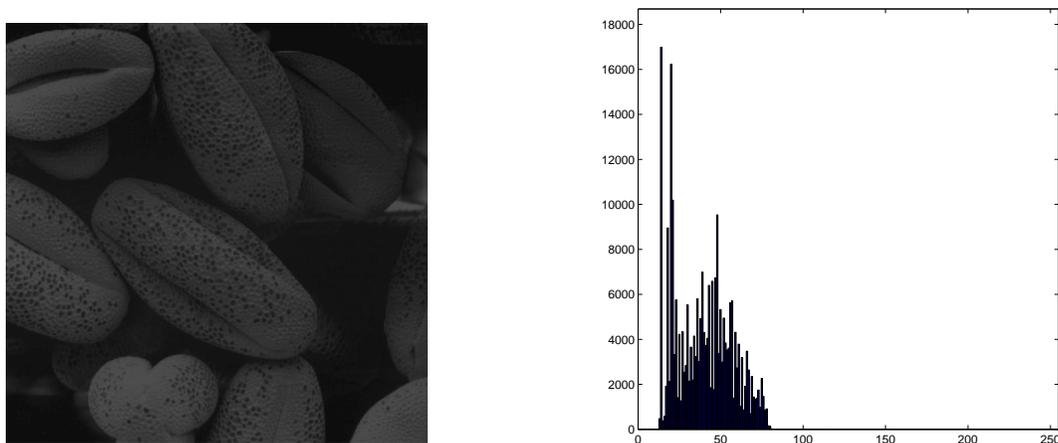


Figura 5: Immagine e relativo istogramma.

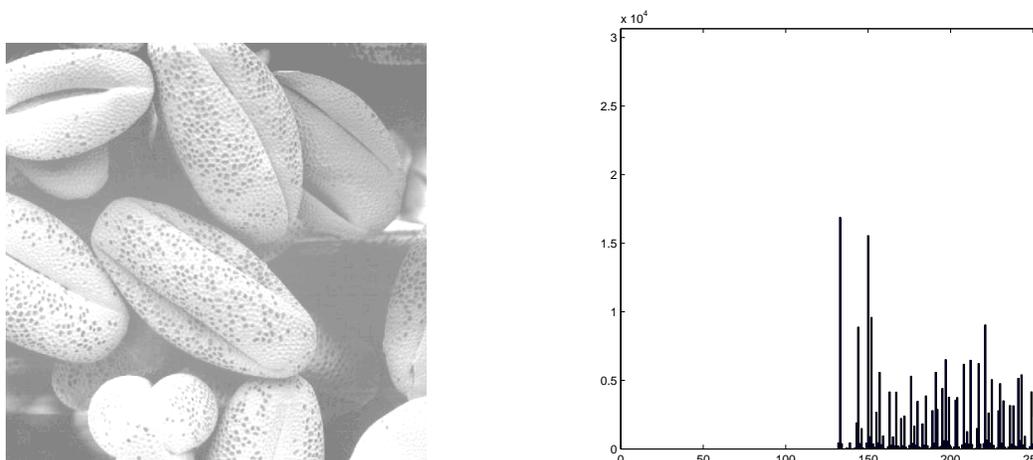


Figura 6: Immagine e relativo istogramma.

2 Trasformazioni senza memoria

Queste trasformazioni si applicano individualmente a ciascun pixel dell'immagine, e possono essere definite in funzione:

1. dell'intensità (o livello di grigio) del pixel (*operazioni puntuali*) e possono anche coinvolgere più immagini in ingresso (*operazioni aritmetiche*);
2. della posizione spaziale (*operazioni geometriche*).

Nel primo caso poichè la trasformazione opera sui livelli di grigio e non sulle posizioni spaziali, scriveremo per semplicità: $y = \mathcal{T}[x]$. Evidentemente ci sono infiniti modi di scegliere il tipo di trasformazione, per comodità rappresenteremo tali elaborazioni attraverso la rappresentazione mostrata in figura 7, in cui come esempio si mostra la trasformazione identica. Si noti come sull'asse delle ascisse ci siano i valori che possono assumere i livelli di grigio $[0, 1, \dots, K - 1]$, si noti anche come si sia posto il vincolo che anche le uscite appartengano allo stesso range di valori. Poiché queste operazioni non hanno alcun effetto sulle posizioni spaziali degli oggetti presenti in un'immagine, tantomeno sulla loro forma, esse modificano la distribuzione dei livelli di grigio e quindi l'istogramma dell'immagine. Per questo motivo, l'istogramma rappresenta lo strumento base che permette di progettare le operazioni puntuali.

2.1 Operazioni lineari

Una trasformazione molto semplice è quella di tipo lineare:

$$y = ax + b$$

dove a è il fattore di scala, mentre b l'offset additivo. E' importante sottolineare che questo tipo di trasformazione ha come effetto che i livelli di grigio in uscita non appartengono più al range $[0, \dots, K - 1]$, per evitare ciò la trasformazione è modificata in modo che se $y < 0$ i valori in uscita vengono posti pari a zero, se invece $y > K - 1$ vengono posti pari al valore $K - 1$. Essendo

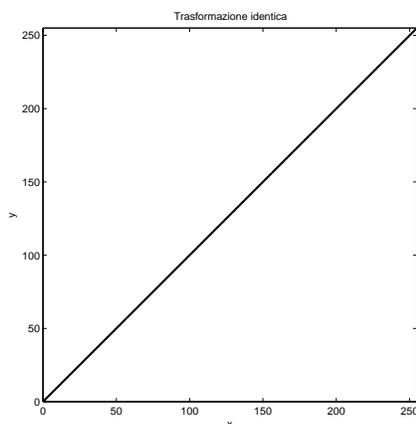


Figura 7: Caratteristica della trasformazione identica.

presente un fenomeno di saturazione, l'operazione non risulta più essere lineare, e bisogna porre molta attenzione, perché si può causare una perdita di informazione tale da introdurre artefatti nell'immagine. In figura 8 si mostra la caratteristica di questa trasformazione in due casi diversi.

Per comprendere quali siano gli effetti di questa operazione sull'istogramma dell'immagine, separiamo le due operazioni di somma di un offset e di moltiplicazione per un fattore di scala.

1. $y = x + b$. L'effetto del solo offset additivo tende a migliorare la visibilità, ma non il contrasto. In particolare, se $b > 0$ l'immagine risulta più brillante, il contrario accade invece se $b < 0$. L'effetto che si ha sull'istogramma è che anch'esso viene traslato; per esempio, data l'immagine di figura 6, ponendo $b = -100$, si ottiene l'immagine mostrata a sinistra in figura 9.

Questa operazione può essere usata per calibrare le immagini ad un dato livello di luminosità media. Supponiamo per esempio di dover confrontare immagini della stessa scena, acquisite in diversi istanti temporali (per esempio, immagini per la sorveglianza di un'area

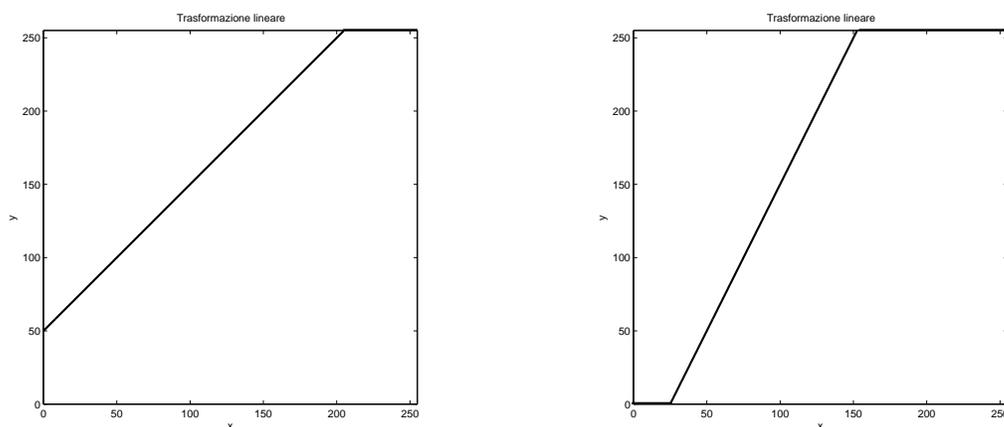


Figura 8: Esempi di trasformazione lineare.

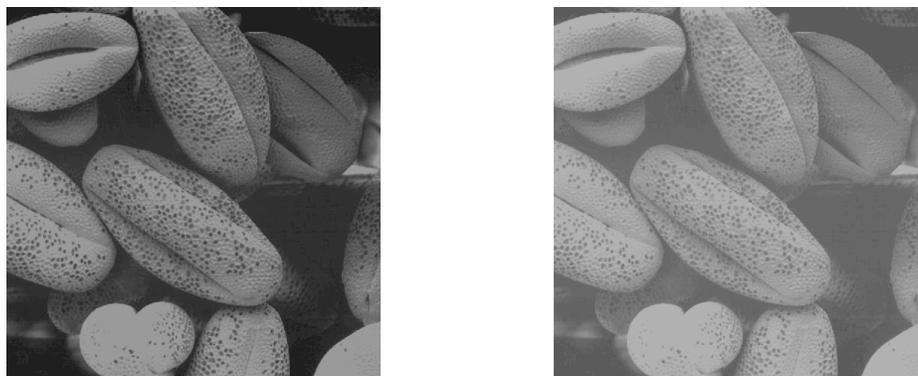


Figura 9: Esempi di trasformazione lineare.

sicura dove ci sono cambiamenti nell'illuminazione globale dell'ambiente). Un modo semplice per confrontarle è fare in modo che la luminosità media (AOD) sia la stessa per tutte, allora se $L_i = AOD(x_i)$ per $i = 1, \dots, N$, bisogna realizzare la seguente trasformazione:

$$y_i = x_i - L_i + K/2$$

In questo modo tutte le immagini prodotte presenteranno un valor medio pari a $K/2$.

2. $y = ax$. Supponiamo $a > 0$, allora se $a > 1$ si ha un'espansione, per cui aumenta la dinamica, se invece $a < 1$ diminuisce (compressione), pertanto la moltiplicazione per un fattore di scala crea una compressione o un'espansione dell'istogramma. E' importante notare che nel caso in cui l'istogramma venga espanso può verificarsi il fenomeno della saturazione, mentre quando l'istogramma viene compresso c'è una corrispondenza molti a uno e si può avere perdita di informazione. Una regola empirica è che l'espansione tende a generare un'immagine più distinta, sempre che gli effetti di saturazione non siano visibili, mentre la compressione riduce la visibilità dei dettagli; questa non è però una regola generale e si possono avere eccezioni. In figura 9 a destra si mostra l'elaborazione dell'immagine di figura 6 in cui si è posto $a = 0.7$.

Esempio: *Negativo di un'immagine*

Un esempio di operazione lineare è il negativo di un'immagine, in cui si pone

$$\begin{cases} a = -1 \\ b = K - 1 \end{cases}$$

La moltiplicazione per $a = -1$ ribalta l'istogramma, mentre l'offset è necessario per riportare i valori ottenuti nel range positivo $[0, 1, \dots, K - 1]$. Questa operazione è comoda quando l'immagine è molto scura ed ha pochi oggetti luminosi, come accade per le immagini delle galassie e delle stelle ottenute attraverso un telescopio, oppure in applicazioni biomedicali (fig.10).

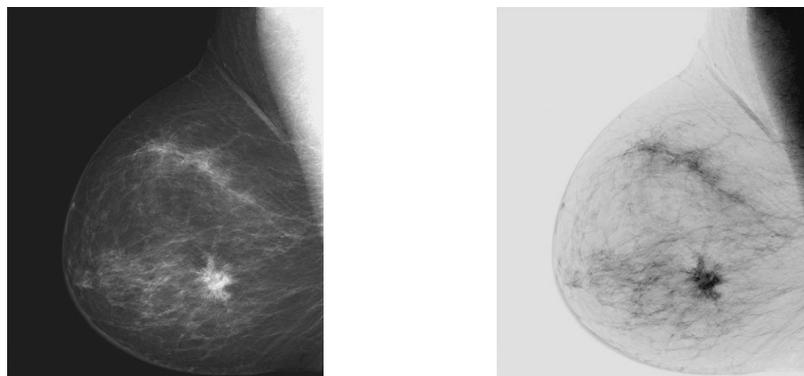


Figura 10: Negativo di una mammografia.

Esempio: *Full Scale Histogram Stretch*

Abbiamo già notato come le immagini delle figure 4, 5 e 6 rivelino un cattivo uso del range dei livelli di grigio. Infatti, se si utilizzassero tutti i valori disponibili, l'immagine tenderebbe ad avere un contrasto più alto e quindi un'apparenza migliore. L'operazione che raggiunge questo obiettivo prende il nome di *Full Scale Histogram Stretch* o *Contrast Stretch* ed è spesso realizzata automaticamente da molti sistemi come le videocamere digitali. Per espandere l'istogramma bisogna fare in modo che il valore minimo che può assumere l'immagine sia zero, e il valore massimo sia $K - 1$, cioè:

$$\begin{cases} a x_{min} + b = 0 \\ a x_{max} + b = K - 1 \end{cases}$$

Risolvendo il sistema si ottiene:

$$\begin{cases} a = \frac{K-1}{x_{max}-x_{min}} \\ b = -x_{min} \left[\frac{K-1}{x_{max}-x_{min}} \right] \end{cases}$$

Per cui complessivamente si ha:

$$y = \text{FSHS}(x) = \left[\frac{K-1}{x_{max}-x_{min}} \right] [x - x_{min}]$$

In figura 11 si mostra l'immagine di figura 5 elaborata con l'operazione appena descritta e il relativo istogramma. Il contrasto e la visibilità dell'immagine sono notevolmente migliorati.

2.2 Operazioni non lineari

Ci sono diversi tipi di operazioni non lineari che permettono di migliorare l'aspetto di un'immagine. Di seguito se ne elencano alcune:

1. *Operazione logaritmo:* $y = c \log(x + 1)$.

Questa operazione comprime in maniera non lineare il range dei livelli di grigio nell'intervallo $[0, \log K]$: i livelli più luminosi vengono compressi di più di quelli scuri (si noti che l'aggiunta di 1 all'immagine evita la possibilità di considerare il logaritmo di zero).

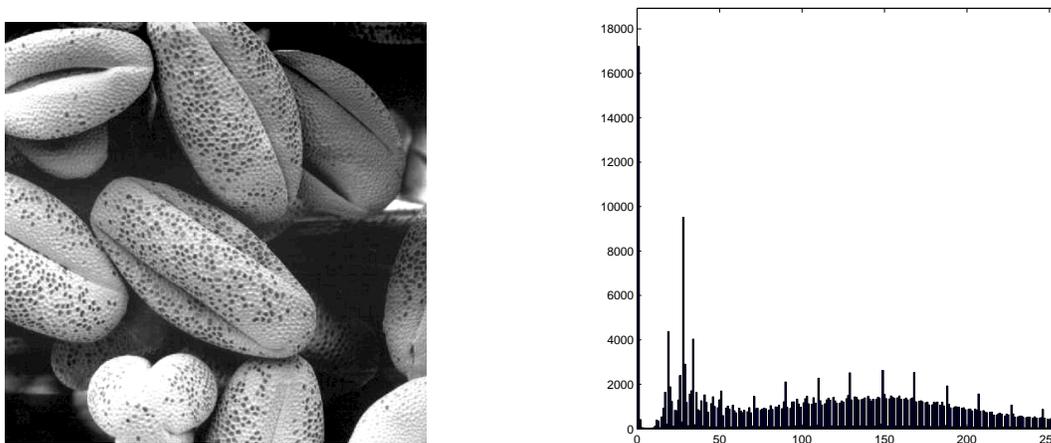


Figura 11: Full Scale Histogram Stretch.

Questa elaborazione è particolarmente utile quando l'immagine è costituita da una quantità molto grande di valori prossimi allo zero e pochi valori estremamente elevati come accade per la trasformata di Fourier di un'immagine, in cui la dinamica varia generalmente in un intervallo molto ampio, che i sistemi di visualizzazione difficilmente riescono a riprodurre. In tal caso i coefficienti più grandi sono quelli relativi alle basse frequenze, mentre i coefficienti alle alte frequenze, essendo molto piccoli, difficilmente vengono visualizzati (figura 12, a sinistra). Dopo l'operazione logaritmo e dopo aver applicato il contrast stretch si ottiene l'immagine di figura 12 a destra.

2. *Operazione potenza: $y = cx^\gamma$.*

La caratteristica di questa trasformazione dipende da γ : se $\gamma > 1$ l'effetto è quello di migliorare il contrasto per i valori chiari e peggiorarlo per quelli scuri (fig.13), il contrario accade per $\gamma < 1$ (fig.14). E' bene sottolineare che questa operazione modifica notevolmente la dinamica dei dati aumentandola ($\gamma > 1$) o riducendola ($\gamma < 1$), per questo motivo per una corretta visualizzazione è necessario realizzare il contrast stretch.

3. *Trasformazioni lineari a tratti.* Un altro approccio per realizzare l'enhancement è attraverso funzioni lineari a tratti; in tal caso uno dei principali vantaggi è che l'andamento della funzione può essere scelto con maggiore flessibilità e può essere arbitrariamente complesso, tuttavia per specificarlo sono necessari un numero maggiore di parametri. In figura 15 si mostrano due esempi. La trasformazione a sinistra può essere utilizzata qualora si voglia realizzare il contrast stretch di un'immagine, modificando opportunamente la posizione dei punti (x_1, y_1) e (x_2, y_2) (es., per $x_1 = x_{min}, x_2 = x_{max}$ e $y_1 = 0, y_2 = K - 1$ si ottiene il Full Scale Histogram Stretch). La trasformazione a destra, invece, può essere usata se si vuole enfatizzare solo un range di valori, (x_1, x_2) , lasciando inalterati gli altri.

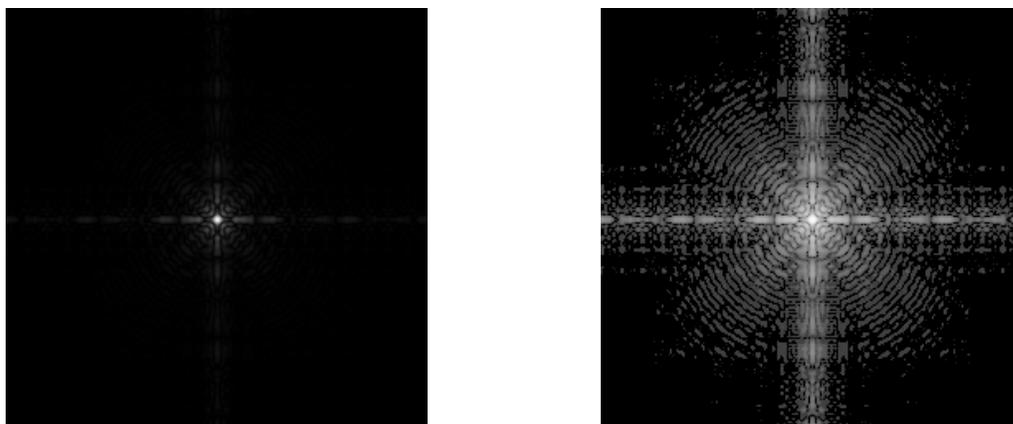


Figura 12: Operazione logaritmo sulla trasformata di Fourier di un'immagine.



Figura 13: Operazione potenza con $\gamma = 5$ (fotografia aerea).

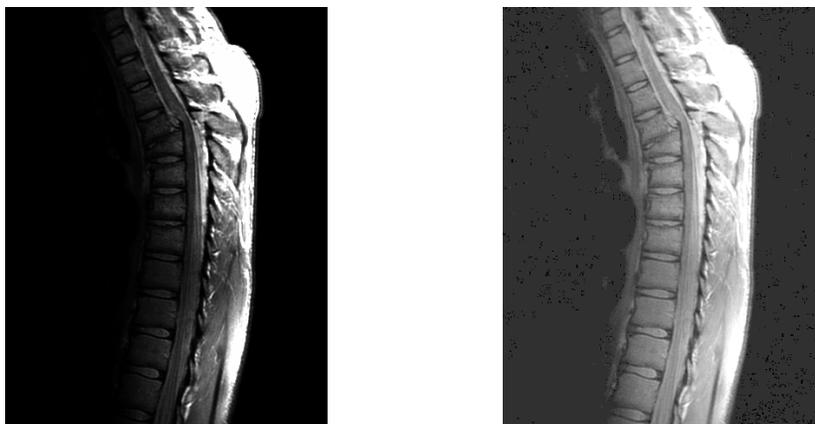


Figura 14: Operazione potenza con $\gamma = 0.3$ (spina dorsale fratturata).

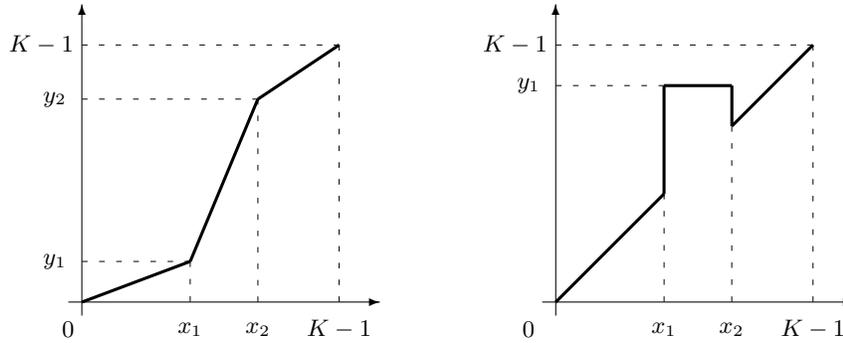


Figura 15: Esempi di trasformazioni lineari a tratti.

2.3 Equalizzazione dell'istogramma

Se si considera l'immagine di figura 16 e si osserva il suo istogramma, si può facilmente notare che vengono usati tutti i livelli di grigio appartenenti alla dinamica, tuttavia l'immagine appare molto scura, a causa del fatto che la distribuzione dei livelli di grigio è fortemente sbilanciata. Un modo per risolvere questo problema è quello di fare in modo che l'istogramma risulti piatto (*equalizzazione dell'istogramma*), cioè i valori che assume l'immagine devono essere uniformemente distribuiti sul range, in questo modo si ottiene una maggiore varietà dei livelli di grigio e quindi un maggiore contrasto. Per comprendere l'elaborazione che bisogna effettuare, è necessario definire l'istogramma normalizzato, dato da

$$p_x(k) = \frac{1}{MN} h_x(k) \tag{3}$$

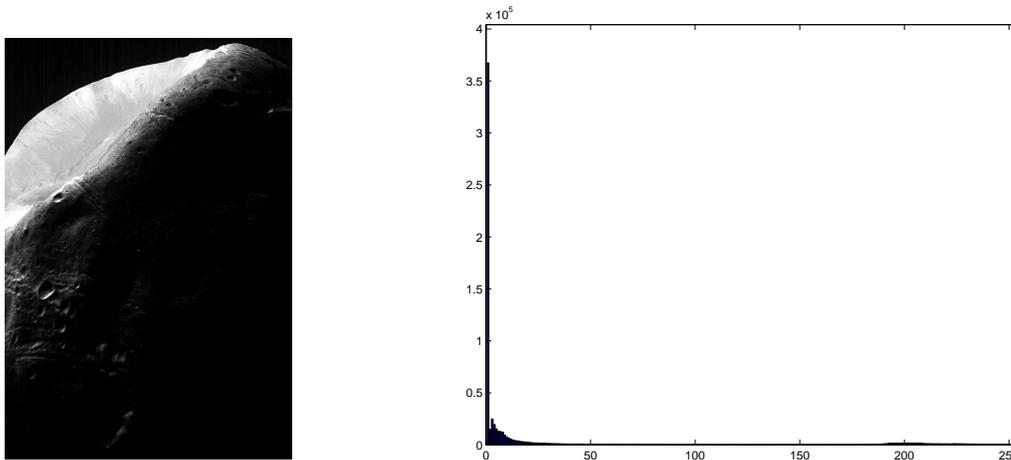


Figura 16: Immagine della superficie di Marte e relativo istogramma.

per $k = 0, 1, \dots, K - 1$. Tale funzione ha la proprietà per cui $\sum_{k=0}^{K-1} p_x(k) = 1$. L'istogramma normalizzato può essere interpretato come una stima della distribuzione di probabilità (pmf o DF) dei valori di livello di grigio dell'immagine¹. In altri termini, se un pixel dell'immagine viene scelto casualmente la probabilità che sia pari proprio a k è

$$p_x(k) = \frac{n_k}{n} = \Pr\{x = k\} \quad (4)$$

dove abbiamo ricordato che $h_x(k) = n_k$ e abbiamo posto $MN = n$, numero di pixel complessivi dell'immagine. Interpretando l'istogramma come una stima della distribuzione di probabilità, l'equalizzazione ha come obiettivo quello di ottenere una distribuzione uniforme, in cui tutti i livelli si verificano con la stessa probabilità e quindi trasportano la massima informazione possibile (massima entropia).

A questo punto è anche possibile definire l'istogramma normalizzato cumulativo

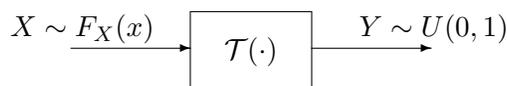
$$P_x(r) = \sum_{k=0}^r p_x(k) \quad r = 0, 1, \dots, K - 1 \quad (5)$$

che invece fornisce una stima della funzione distribuzione cumulativa (CDF), funzione non decrescente e tale che $P_x(K - 1) = 1$. Risulta

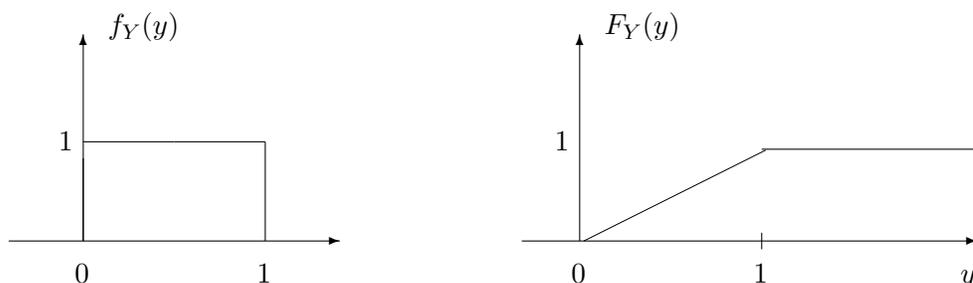
$$p_x(r) = P_x(r) - P_x(r - 1) \quad r = 0, 1, \dots, K - 1$$

evidentemente le due funzioni sono legate l'un l'altra ed entrambe forniscono una descrizione completa della distribuzione dei livelli di grigio dell'immagine.

Per semplicità affrontiamo il problema dell'equalizzazione nell'ipotesi in cui le variabili coinvolte sono continue, successivamente formuleremo il caso discreto come approssimazione di quello continuo. Supponiamo allora di considerare una variabile aleatoria X con CDF $F_X(x)$ o equivalentemente con pdf $f_X(x) = \frac{dF_X(x)}{dx}$. Vogliamo cercare di capire qual è la trasformazione \mathcal{T} che a partire da X ci permette di ottenere una variabile aleatoria uniforme $Y \sim U(0, 1)$.



Cominciamo col ricordare che una variabile aleatoria uniforme ha le seguenti pdf e CDF:



¹Si tenga presente che l'immagine è un segnale aleatorio, che può essere caratterizzato da un punto di vista statistico mediante la distribuzione di probabilità, tuttavia è molto difficile avere tale caratterizzazione e un modo per stimarla dalla singola immagine è proprio attraverso l'istogramma. Si noti a questo punto che l'equazione (1) non è altro che la stima della media statistica dell'immagine.

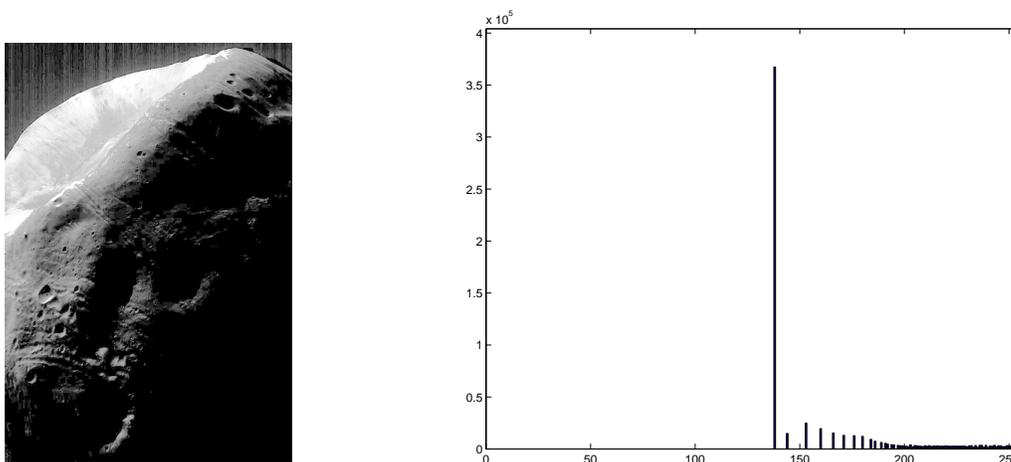


Figura 17: Equalizzazione dell'istogramma.

Si ha

$$F_Y(y) = \Pr\{Y \leq y\} = \Pr\{\mathcal{T}(X) \leq y\} = \Pr\{X \leq \mathcal{T}^{-1}(y)\} = F_X[\mathcal{T}^{-1}(y)]$$

Ora se scegliamo $\mathcal{T} \equiv F_X$ avremo:

$$F_Y(y) = F_X[F_X^{-1}(y)] = y \quad 0 \leq y \leq 1$$

Di conseguenza la pdf risulta essere costante e pari a 1 per $0 \leq y \leq 1$. Allora per ottenere una variabile aleatoria uniforme in $(0, 1)$ bisogna usare come trasformazione proprio la CDF dei dati da elaborare. Purtroppo non è possibile fare una dimostrazione analoga anche nel caso discreto, però possiamo affermare che se utilizziamo come trasformazione proprio la stima della CDF, l'istogramma del segnale in uscita è approssimativamente piatto, o almeno più di quanto non lo sia quello del segnale in ingresso. La trasformazione da adottare è la seguente:

$$y_k = \mathcal{T}[x = k] = \sum_{j=0}^k \frac{n_j}{n} \quad k = 0, 1, \dots, K-1$$

In figura 17 si mostra l'immagine di Marte di figura 16 su cui è stata effettuata l'equalizzazione dell'istogramma. Si può notare come adesso siano visibili regioni che prima erano completamente oscurate, sebbene l'istogramma non sia realmente piatto.

2.4 Esempio di enhancement mediante l'uso delle statistiche locali

Invece di usare l'istogramma dell'immagine per realizzare enhancement, è anche possibile far riferimento a parametri statistici che possono essere derivati direttamente dall'istogramma. Nel paragrafo precedente abbiamo introdotto l'istogramma normalizzato (4), che può essere interpretato come stima della distribuzione di probabilità dei livelli di grigio. Definiamo allora la *media* di tali livelli come:

$$\mu_x = \sum_{k=0}^{K-1} k p_x(k) \quad (6)$$

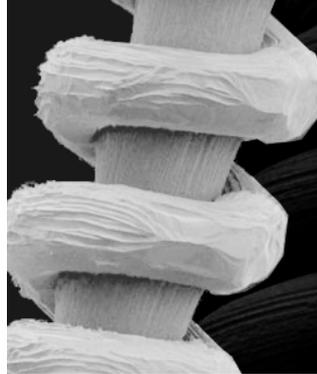


Figura 18: Immagine ottenuta mediante microscopio elettronico.

che coincide di fatto con la (1) e rappresenta il valore medio che possono assumere i livelli di grigio. Possiamo poi definire la *varianza* come:

$$\sigma_x^2 = \sum_{k=0}^{K-1} (k - \mu_x)^2 p_x(k) \quad (7)$$

che rappresenta una misura del contrasto presente in un'immagine. Entrambi questi parametri sono definiti su tutta l'immagine per cui danno indicazioni sulle caratteristiche globali. Queste misure possono essere molto utili per realizzare l'enhancement *locale*, cioè per modificare le caratteristiche di un'immagine in una regione predefinita. In tal caso sarà necessario determinare media e varianza locali, calcolate cioè in una zona dell'immagine di opportune dimensioni.

Facciamo un esempio, per comprendere meglio come si può realizzare tale elaborazione. Consideriamo l'immagine di figura 18 che mostra l'ingrandimento (ottenuto mediante microscopio elettronico) di un filamento di tungsteno avvolto attorno a un supporto. Il filamento al centro dell'immagine e il suo supporto sono chiari e semplici da studiare, c'è però un altro filamento sulla destra dell'immagine che è molto più scuro e poco visibile. In questo caso risulta perciò interessante effettuare un enhancement solo nella regione che presenta una struttura nascosta.

L'approccio che può essere usato per enfatizzare le zone scure e lasciare inalterata l'area chiara sfrutta proprio il calcolo della media e varianza locali nell'immagine. Per stabilire se un pixel dell'immagine va elaborato o meno (quindi se appartiene alla regione chiara o scura) è possibile calcolare la media intorno al pixel considerato (μ_l , media locale) e confrontarla con quella dell'intera immagine (μ_x , media globale). Se

$$\mu_l \leq k_0 \mu_x$$

dove k_0 è una costante positiva con valore minore di 1, il pixel va elaborato. Poiché siamo interessati ad enfatizzare le aree con basso contrasto, è necessario effettuare un ulteriore test per stabilire se il pixel deve essere enfatizzato. Essendo la varianza o la deviazione standard una misura del contrasto presente in un'immagine, allora quella calcolata intorno al pixel considerato (σ_l , deviazione standard locale) deve essere più bassa di quella globale (σ_x , deviazione standard globale), ma non troppo piccola per non individuare regioni omogenee, deve quindi accadere:

$$k_1 \sigma_x \leq \sigma_l \leq k_2 \sigma_x$$



Figura 19: Immagine delle medie e delle deviazioni standard.

con $k_1 < k_2$ e k_2 costante positiva. Se il pixel soddisfa le condizioni specificate sarà soggetto ad enhancement (moltiplicandolo per una costante E), altrimenti verrà lasciato inalterato:

$$y = \begin{cases} E \cdot x & \mu_l \leq k_0 \mu_x \text{ AND } k_1 \sigma_x \leq \sigma_l \leq k_2 \sigma_x \\ x & \text{altrimenti} \end{cases}$$

I valori da assegnare ai parametri in gioco (E, k_0, k_1 e k_2) dipendono dall'applicazione in esame, in particolare è richiesta una fase sperimentale per acquisire sufficienti conoscenze delle immagini da elaborare. I valori selezionati sono stati $E = 4, k_0 = 0.4, k_1 = 0.02, k_2 = 0.4$ e sono fortemente legati alle statistiche globali dell'immagine e a quelle della regione da enfatizzare. La dimensione dell'area locale su cui valutare le statistiche è stata scelta pari a 3×3 .

In figura 19 si mostrano l'immagine delle medie locali e quella delle deviazioni standard, si noti come l'immagine delle medie è simile a quella originale, semplicemente più sfocata, mentre quella delle deviazioni standard è molto luminosa nelle aree ad elevato contrasto. Infine, in figura 20 si mostra il risultato del test binario (che indica se il pixel dell'immagine va elaborato (1, bianco) oppure no (0, nero)) e l'immagine finale.



Figura 20: Immagine che indica i pixel da enfatizzare e risultato dell'elaborazione.

2.5 Bit-plane slicing

Invece di enfatizzare i livelli di grigio, può essere desiderabile mettere in luce il contributo dato dai bit che costituiscono la codifica (binaria) dei valori di un'immagine. E' necessario allora effettuare il *bit-plane slicing*, che consiste nel suddividere l'immagine in piani, in cui si rappresentano ognuno dei bit da quello meno significativo a quello più significativo.

In figura 22, 23, 24 e 25 sono rappresentati i bit-plane dell'immagine frattale di figura 21, codificata su 8 bit. In particolare in figura 22 ci sono i due bit-plane più significativi, mentre in figura 25 quelli meno significativi. Si può notare come i bit-plane di ordine più elevato contengono i dati più significativi da un punto di vista visivo, i successivi bit-plane contribuiscono a formare i dettagli presenti nell'immagine, dal momento che migliorano ulteriormente la precisione con cui i livelli di grigio vengono rappresentati. In effetti, la separazione di un'immagine in bit-plane è molto utile per analizzare l'importanza relativa che gioca ogni bit e quindi l'adeguatezza del numero di bit usato per quantizzare ogni pixel.

Supponiamo di annullare gli ultimi 4 bit-plane dell'immagine e poi di ricostruirla nuovamente.

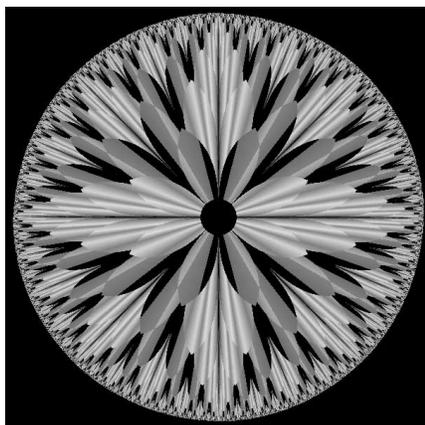


Figura 21: Immagine originale su 8 bit.

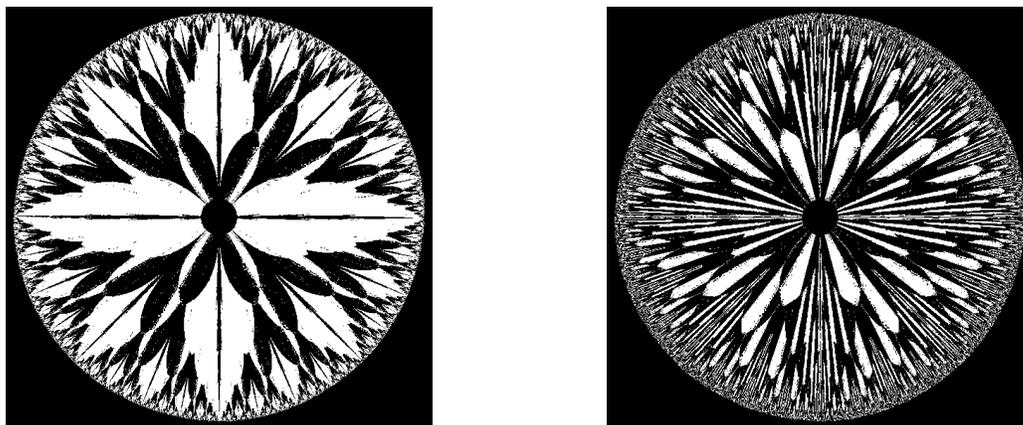


Figura 22: Bit-plane 7 e 6.

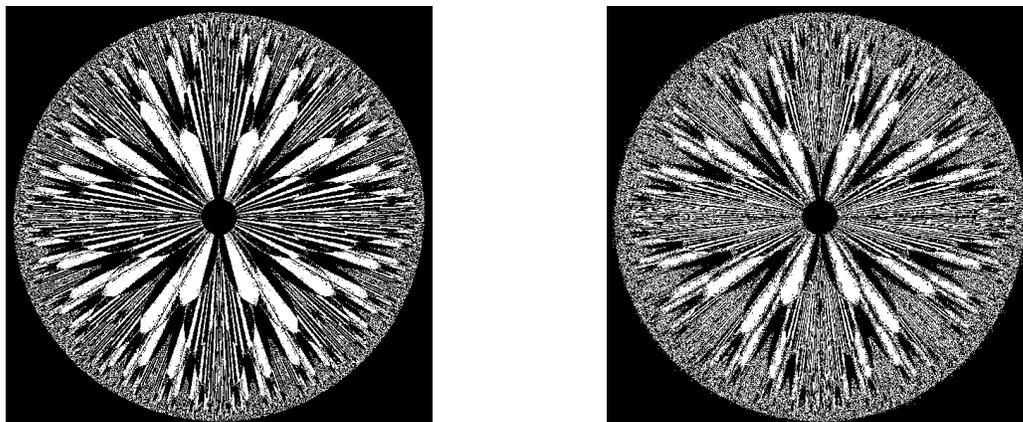


Figura 23: Bit-plane 5 e 4.

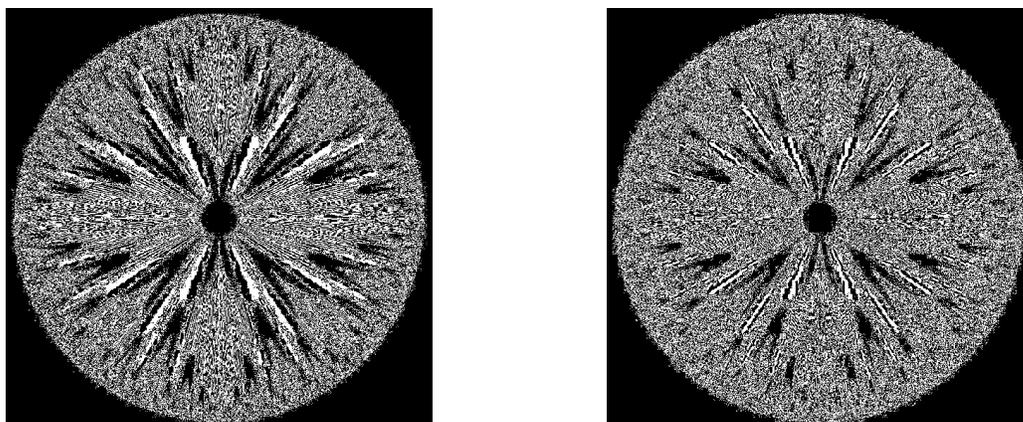


Figura 24: Bit-plane 3 e 2.

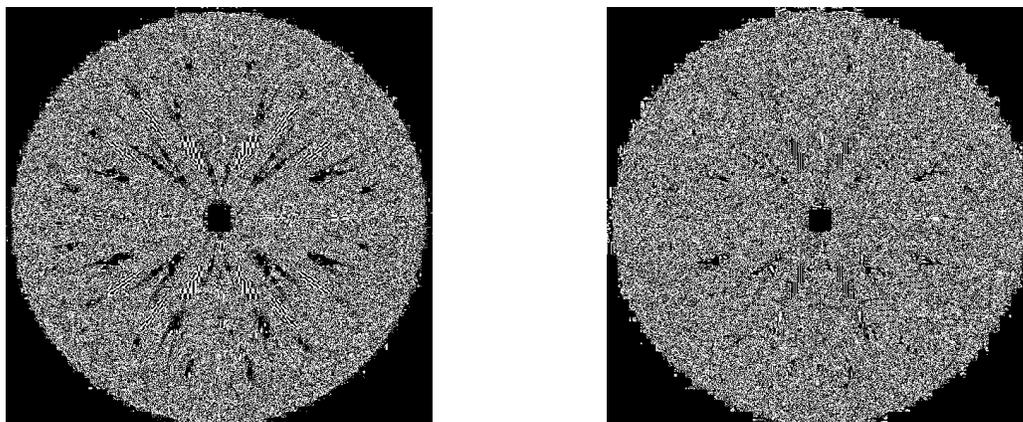


Figura 25: Bit-plane 1 e 0.

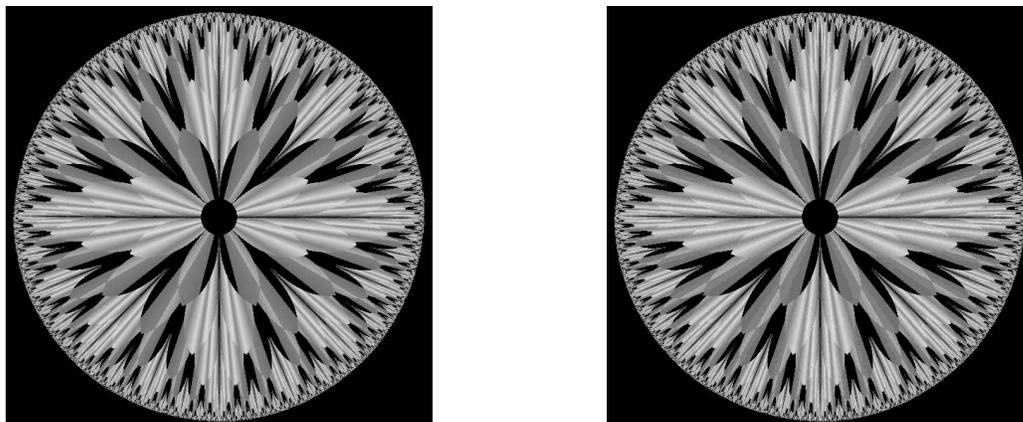


Figura 26: Immagine originale e immagine ricostruita con i primi 4 bit-plane.

Il risultato di questa elaborazione è mostrato in figura 26, in cui l'immagine è confrontata con quella originale, e da cui difficilmente si riesce a distinguere. Questa osservazione ci conferma il fatto che i bit-plane meno significativi sono in genere meno importanti nella ricostruzione dell'immagine, almeno da un punto di vista visivo.

2.6 Operazioni aritmetiche

Le operazioni aritmetiche (addizione/sottrazione, prodotto/divisione) coinvolgono due o più immagini, sono realizzate pixel per pixel e risultano estremamente utili nel processo di enhancement. Per esempio sottrarre due immagini permette di enfatizzare le differenze che esistono tra le due. In figura 27 è mostrata l'immagine $z(m, n) = x(m, n) - y(m, n)$ con $x(m, n)$ e $y(m, n)$ immagini rappresentate in figura 26; si noti come la differenza $z(m, n)$ consenta di stabilire con precisione quali sono le regioni in cui le due immagini differiscono sensibilmente.

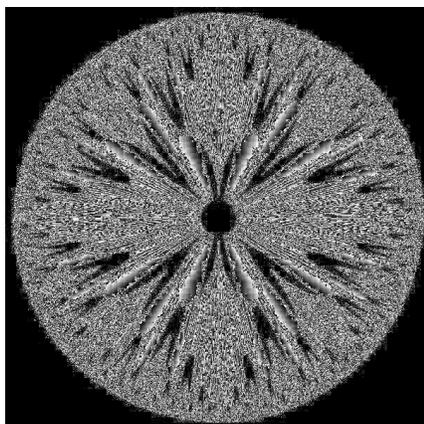


Figura 27: Immagine differenza delle immagini di figura 26.

2.7 Operazioni geometriche

Le trasformazioni geometriche modificano l'aspetto di un'immagine, cambiando le coordinate dei valori di luminosità. Si ottiene così una nuova immagine nella quale i livelli di grigio non sono stati modificati, ma sono prelevati in posizioni diverse da quelle originali. Si realizza quindi una trasformazione del tipo:

$$y(m, n) = x(m', n')$$

con

$$\begin{cases} m' = \mathcal{T}_1(m, n) \\ n' = \mathcal{T}_2(m, n) \end{cases} \quad (8)$$

Per effettuare un'operazione geometrica è necessario risolvere due problemi: individuare le funzioni \mathcal{T}_1 e \mathcal{T}_2 e assegnare correttamente i valori di intensità ai pixel trasformati. Quest'ultimo punto è fondamentale perché può capitare di dover valutare $y(m, n)$ quando m' e/o n' non sono definite nell'immagine originale (assumono cioè valori non interi). Una tra le più comuni trasformazioni di coordinate spaziali è la *trasformazione affine* che ha la forma generica:

$$[m' \ n' \ 1] = [m \ n \ 1] \mathbf{T} = [m \ n \ 1] \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

Specificando i valori della matrice \mathbf{T} è possibile ridimensionare, ruotare, traslare o inclinare l'immagine. Di seguito sono elencate sia le equazioni delle coordinate, specificate da \mathcal{T}_1 e \mathcal{T}_2 , che i valori da assegnare alla matrice per le varie trasformazioni geometriche:

1. Ridimensionamento.

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{cases} m' = a m \\ n' = b n \end{cases}$$

a e b sono costanti positive. In particolare, se $a < 1$ si effettua un ingrandimento dell'immagine (*zoom out*), mentre per $a > 1$ l'immagine viene rimpicciolita (*zoom in*).

2. Traslazione.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m_0 & n_0 & 1 \end{bmatrix} \quad \begin{cases} m' = m + m_0 \\ n' = n + n_0 \end{cases}$$

L'immagine risultante sarà traslata di un fattore m_0 nella direzione orizzontale e n_0 in quella verticale.

3. *Rotazione.*

$$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{cases} m' = m \cos \theta - n \sin \theta \\ n' = m \sin \theta + n \cos \theta \end{cases}$$

Si noti che per $\theta = \pi$ l'immagine viene ribaltata $y(m, n) = x(-m, -n)$.

4. *Distorsione.*

Se si vuole deformare l'immagine lungo la direzione verticale la trasformazione è:

$$\begin{bmatrix} 1 & 0 & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{cases} m' = m + cn \\ n' = n \end{cases}$$

Invece lungo la direzione orizzontale:

$$\begin{bmatrix} 1 & d & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{cases} m' = m \\ n' = dm + n \end{cases}$$

Il vantaggio della rappresentazione matriciale è che consente di concatenare una sequenza di operazioni. Per esempio, se si vuole ruotare e poi traslare un'immagine, è possibile generare una matrice 3×3 ottenuta dal prodotto delle relative matrici di rotazione e traslazione.

Nelle trasformazioni considerate c'è la possibilità che i valori assunti da m' e n' non siano interi, per cui è necessario interpolare i dati allo scopo di definire correttamente i valori di intensità nelle nuove posizioni spaziali. Si utilizza il seguente approccio:

- si considera $x(m, n)$ come la versione campionata di un'opportuna immagine analogica $x_a(u, v)$;
- si sceglie un modello per $x_a(u, v)$, in modo che tale funzione possa essere calcolata per qualsiasi punto $(u, v) \in \mathbb{R}^2$;
- si ricampiona $x_a(u, v)$.

Il passaggio più delicato e più oneroso computazionalmente è l'*interpolazione* cioè il passaggio da $x(m, n)$ a $x_a(u, v)$. Considereremo due approcci, e cioè l'interpolazione *nearest neighbour* e quella *bilineare*. Supponiamo per esempio di voler ingrandire un'immagine di un fattore 2, la nuova griglia di punti avrà dimensioni maggiori della prima, per cui sarà necessario assegnare dei nuovi livelli di grigio a posizioni spaziali che prima non esistevano. In particolare bisogna porre $a = b = 1/2$:

$$y(m, n) = x\left(\frac{m}{2}, \frac{n}{2}\right)$$

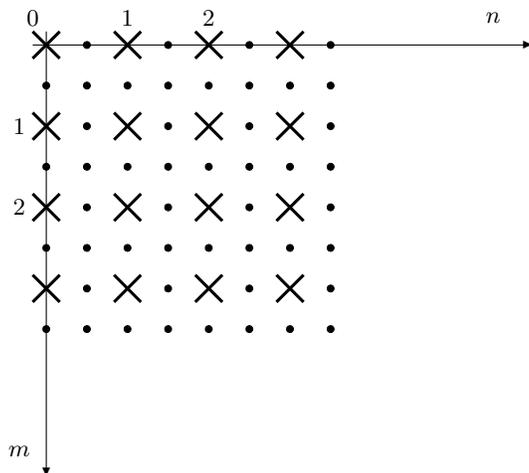


Figura 28: Griglia per l'espansione di un fattore 2 di un'immagine.

In tal caso risultano definiti solo i valori corrispondenti a posizioni spaziali multiple di 2 (fig.28). Per assegnare anche gli altri valori si può considerare $x(m, n)$ come la versione campionata di un'opportuna immagine analogica $x_a(u, v)$ e poi ricampionare tale immagine nei punti in cui il segnale non era definito. In questo processo risulta di fondamentale importanza il modello utilizzato per l'immagine analogica, in base al quale sono possibili diversi tipi di interpolazioni:

1. *Nearest Neighbor*. I livelli di grigio nelle nuove posizioni spaziali vengono valutati come:

$$x(m', n') = x(\text{round}(m'), \text{round}(n'))$$

dove $\text{round}(R)$ denota l'operazione di arrotondamento all'intero più vicino di R , in questo modo m' e n' assumeranno sempre valori interi. Questa trasformazione corrisponde a considerare un modello di immagine costante in ogni regione in cui bisogna interpolare

$$x_a(u, v) = a_0$$

in modo analogo all'interpolazione costante a tratti (o di ordine 0) del caso monodimensionale. La costante a_0 è il valore del pixel $x(m', n')$, dove (m', n') è il nearest neighbor di (u, v) . Poiché questo tipo di soluzione semplicemente assegna il valore del pixel più “vicino” alle nuove posizioni spaziali, diverse coordinate spaziali assumeranno lo stesso livello di grigio. Per questo motivo nell'immagine sarà visibile un effetto di “blocchettizzazione”, molto fastidioso soprattutto quando sono presenti repentini cambiamenti di intensità.

2. *Bilineare*. Un modello più complesso per l'immagine analogica è il seguente:

$$x_a(u, v) = a_0 + a_1u + a_2v + a_3uv \quad (9)$$

In tal caso il valore assegnato alle nuove posizioni spaziali è dato da una combinazione lineare dei livelli di grigio dei quattro pixel più vicini. Per esempio supponiamo di dover determinare il valore del pixel $x(1/2, 1/2)$, allora bisognerà interpolare i livelli assegnati ai pixel $x(0, 0)$, $x(0, 1)$, $x(1, 0)$ e $x(1, 1)$, risolvendo il seguente sistema di equazioni:

$$\begin{cases} x(0, 0) = a_0 \\ x(0, 1) = a_0 + a_2 \\ x(1, 0) = a_0 + a_1 \\ x(1, 1) = a_0 + a_1 + a_2 + a_3 \end{cases}$$

da cui si ottengono i coefficienti a_0 , a_1 , a_2 e a_3 :

$$\begin{cases} a_0 = x(0, 0) \\ a_1 = x(1, 0) - x(0, 0) \\ a_2 = x(0, 1) - x(0, 0) \\ a_3 = x(0, 0) - x(0, 1) - x(1, 0) + x(1, 1) \end{cases}$$

In conclusione, per la (9) risulta:

$$\begin{aligned} x\left(\frac{1}{2}, \frac{1}{2}\right) &= a_0 + \frac{a_1}{2} + \frac{a_2}{2} + \frac{a_3}{4} \\ &= x(0, 0) + \frac{x(1, 0) - x(0, 0)}{2} + \frac{x(0, 1) - x(0, 0)}{2} + \frac{x(0, 0) - x(0, 1) - x(1, 0) + x(1, 1)}{4} \\ &= \frac{x(0, 0) + x(0, 1) + x(1, 0) + x(1, 1)}{4} \end{aligned}$$

Si noti come il valore centrale si ottenga proprio come media pesata dei quattro pixel vicini, così come accade nell'interpolazione lineare (interpolazione di ordine 1) nel caso monodimensionale.

Ovviamente è possibile utilizzare modelli più complessi in cui vengono coinvolti più pixel nell'interpolazione (interpolazioni di ordine maggiore di 1). Teoricamente l'interpolazione ideale è quella mediante le funzioni sinc, tuttavia in tal caso sarebbero richiesti infiniti pixel nel processo di interpolazione, per cui bisognerebbe comunque effettuare un'approssimazione.

In figura 30 sono mostrati due zoom dell'immagine di figura 29, il primo con interpolazione nearest neighbor e il secondo con quella lineare. Si noti l'effetto a blocchetti causato dall'interpolazione nearest neighbor. Infine, in figura 31 si mostrano due esempi di traslazione e rotazione della stessa immagine.



Figura 29: Immagine originale di dimensioni 256×256 .

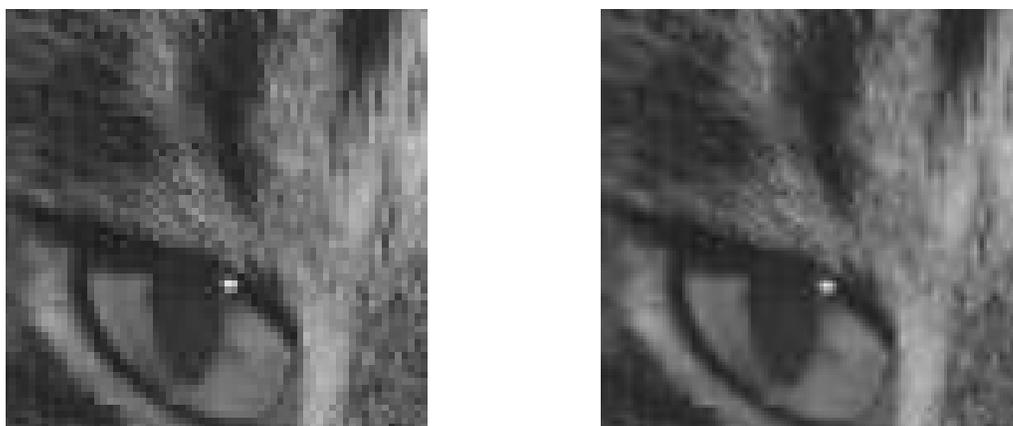


Figura 30: Zoom ($a = b = 1/4$) con interpolazione costante e bilineare.

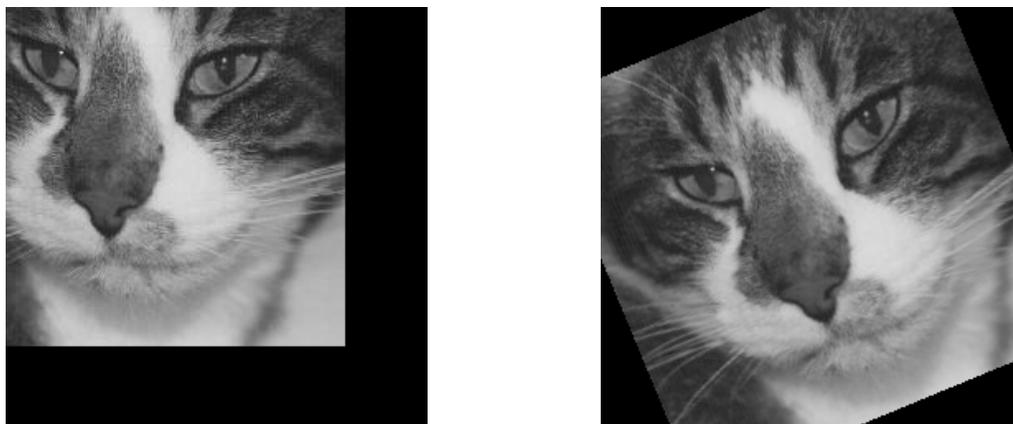


Figura 31: Immagine traslata con $m_0 = n_0 = 50$ e ruotata con $\theta = \pi/8$.

3 Trasformazioni con memoria

Il filtraggio spaziale opera direttamente sui pixel dell'immagine per ottenere l'uscita. Nell'ipotesi di considerare trasformazioni lineari e spazio invarianti, il valore del pixel dell'immagine risultante, $y(m, n)$, è dato da una combinazione lineare dei livelli di grigio di $x(m, n)$, pesati secondo opportuni coefficienti $a(\cdot, \cdot)$, che solitamente appartengono ad una finestra di dimensioni fissate, detta anche *maschera* del filtro ($W_{KL} = (k, l) \in \{-K, \dots, K\} \times \{-L, \dots, L\}$), pertanto:

$$y(m, n) = \sum_{k, l \in W_{KL}} a(k, l) x(m + k, n + l) \quad (10)$$

Supponiamo, per esempio, di considerare una maschera di dimensioni 3×3 il risultato dell'elaborazione è dato da:

$$y(m, n) = a(-1, -1) x(m - 1, n - 1) + \dots + a(0, 0) x(m, n) + \dots + a(1, 1) x(m + 1, n + 1)$$

Quindi il valore del pixel $x(m, n)$ viene sostituito con $y(m, n)$ che rappresenta una media pesata dei valori che appartengono alla maschera. Di seguito si mostra la matrice con i coefficienti della maschera 3×3

$$\begin{bmatrix} a(-1, -1) & a(-1, 0) & a(-1, 1) \\ a(0, -1) & a(0, 0) & a(0, 1) \\ a(1, -1) & a(1, 0) & a(1, 1) \end{bmatrix}$$

e la corrispondente sezione dell'immagine che viene elaborata:

$$\begin{bmatrix} x(m - 1, n - 1) & x(m - 1, n) & x(m - 1, n + 1) \\ x(m, n - 1) & x(m, n) & x(m, n + 1) \\ x(m + 1, n - 1) & x(m + 1, n) & x(m + 1, n + 1) \end{bmatrix}$$

Si noti come la relazione (10) sia molto simile a quella della convoluzione bidimensionale:

$$y(m, n) = \sum_{k, l} h(k, l) x(m - k, n - l) \quad (11)$$

In effetti, pur di scegliere la risposta impulsiva come:

$$h(k, l) = \begin{cases} a(-k, -l) & k, l \in W_{KL} \\ 0 & \text{altrove} \end{cases} \quad (12)$$

La (10) e la (11) coincidono, per cui l'operazione che stiamo realizzando è a tutti gli effetti un filtraggio. In realtà, la (10) calcola la correlazione tra l'immagine e la maschera del filtro, ma dal momento che spesso si considerano maschere a simmetria circolare, in cui nessuna direzione dell'immagine viene privilegiata, allora possiamo confondere l'operazione di correlazione e di convoluzione. La scelta dei coefficienti della maschera influenza ovviamente il tipo di elaborazione. In particolare, quando nell'immagine prodotta le brusche variazioni vengono ridotte, e quindi l'immagine è una versione "sfocata" di quella originale, si parla di filtri di *smoothing*, se invece si enfatizzano i dettagli contenuti in un'immagine i filtri si dicono di *sharpening*.

E' chiaramente anche possibile considerare trasformazioni con memoria in cui il legame non sia di tipo lineare, così come accade per il filtro *mediano*, in cui il valore del pixel $x(m, n)$ viene sostituito con il valore mediano dei pixel appartenenti alla maschera.

Indipendentemente dal tipo di filtraggio, poiché queste trasformazioni elaborano un insieme di pixel che circondano quello attuale, c'è il problema di gestire i pixel che si trovano ai bordi dell'immagine, infatti in tal caso la finestra si trova solo parzialmente sovrapposta all'immagine. Supponiamo di considerare una maschera di dimensioni $n \times n$, ci sono tre possibili approcci al problema:

1. limitare l'escursione del centro della maschera ad una distanza non inferiore a $(n - 1)/2$ dal bordo. L'immagine risultante sarà più piccola di quella originale, tanto più piccola quanto più è grande la dimensione del filtro;
2. filtrare i pixel che si trovano ai bordi solo con la sezione della maschera che è contenuta nell'immagine, questo significa che i pixel vicino ai bordi saranno elaborati in modo diverso dagli altri, dato che si considera una maschera parziale, tuttavia si garantisce che le dimensioni dell'immagine siano uguali all'originale;
3. estendere l'immagine ai bordi aggiungendo righe e colonne di zeri (*zero padding*). Questa soluzione di solito crea una distorsione ai bordi dell'immagine, che diventa rilevante al crescere della dimensione del filtro, per questo motivo si effettua di solito un'estensione periodica o simmetrica.

3.1 Filtri di smoothing

I filtri di smoothing sono utilizzati per esempio per rimuovere piccoli dettagli in un'immagine per poi poter estrarre oggetti più grandi o per riempire piccoli vuoti su linee o curve; o ancora sono molto utili per ridurre rumore in un'immagine. In ogni caso, l'effetto complessivo è quello di rendere l'immagine meno nitida di quella originale (*blurring*).

Questi filtri non fanno altro che mediare i valori dei pixel del vicinato secondo opportuni pesi in modo da ridurre le brusche variazioni dei livelli di grigio. Dal momento che il rumore aleatorio è tipicamente costituito da forti cambiamenti nei livelli di luminosità, da ciò si spiega l'utilizzo di tali filtri nel limitare il rumore presente in un'immagine (*denoising*). D'altra parte le stesse immagini sono caratterizzate da bordi (che solitamente trasportano una parte rilevante dell'informazione), per cui l'immagine elaborata risulta soggetta ad una distorsione non desiderabile, che ha il risultato visivo di sfocare i contorni. Bisogna allora trovare il giusto compromesso tra riduzione del rumore e distorsione dimensionando opportunamente la finestra.

Esempi di maschere 3×3 utilizzate per effettuare lo smoothing sono presentate di seguito:

$$a_1(m, n) = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} /9 \qquad a_2(m, n) = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} /16$$

La prima maschera non fa altro che realizzare la media aritmetica dei livelli di grigio del vicinato e quindi pesa tutti i pixel allo stesso modo, la seconda maschera invece pesa i pixel in modo proporzionale dalla loro distanza dal centro della maschera; quest'ultima soluzione tende

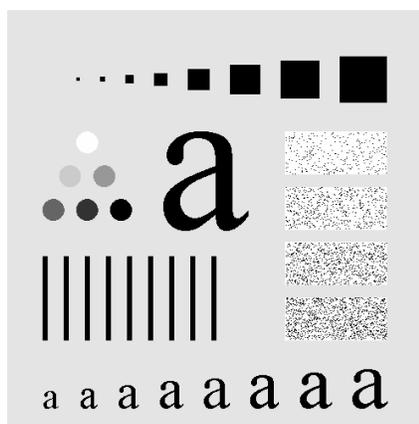
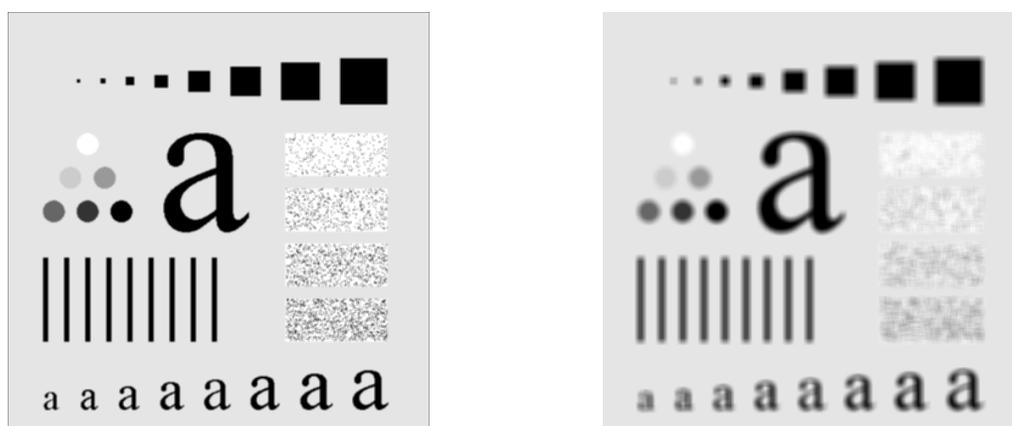


Figura 32: Immagine originale.

Figura 33: Immagine elaborata con un filtro media aritmetica con $n = 3$ e $n = 9$.

a ridurre gli effetti di blurring, seppure risulti abbastanza difficile notare le differenze nelle immagini prodotte in questo caso, essendo la dimensione della finestra molto piccola. Notate come la costante di normalizzazione sia data proprio dalla somma di tutti i coefficienti presenti nella maschera, in modo tale da non modificare il livello medio di intensità dell'immagine.

Esempio: *Effetto di blurring al variare della dimensione della finestra.* In figura 33 e 34 si mostrano gli effetti dello smoothing sull'immagine mostrata in figura 32 utilizzando il filtro media aritmetica con dimensioni della finestra, supposta quadrata $n \times n$, via via crescente per $n = 3, 9, 15, 35$. Si noti l'effetto di perdita di nitidezza che si ha in tutti i casi. In particolare, per $n = 3$ e 9 l'effetto di blurring è particolarmente evidente per gli oggetti che hanno dimensioni confrontabili con la maschera del filtro. Al crescere della dimensione della finestra ($n = 15$ e 35) l'effetto diventa rilevante, inoltre si comincia a notare una cornice nera intorno all'immagine, a causa dell'estensione con zeri ai bordi (zero-padding). Tale effetto può essere ridotto o addirittura eliminato estendendo opportunamente l'immagine ai bordi.

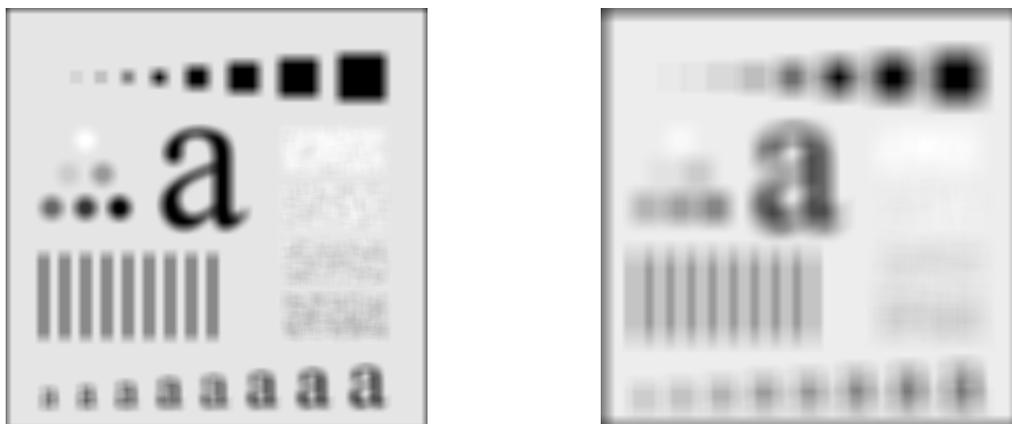


Figura 34: Immagine elaborata con un filtro media aritmetica con $n = 15$ e $n = 35$.

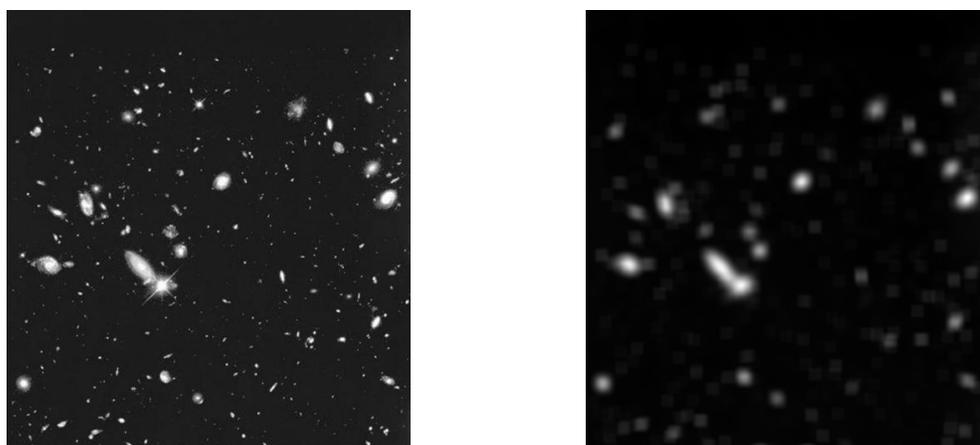


Figura 35: Immagine originale e elaborata con un filtro media aritmetica con $n = 3$.

Esempio: *Riduzione di oggetti piccoli in un'immagine.* Come già accennato un'importante applicazione di un filtro di smoothing è quella di eliminare oggetti molto piccoli per mettere alla luce quelli di interesse, che risulteranno più facili da rilevare. La dimensione della maschera è scelta in rapporto alla grandezza degli oggetti che si vogliono oscurare.

Consideriamo allora l'immagine di figura 35 (a sinistra), proveniente dal telescopio Hubble in orbita intorno alla terra, ed elaboriamola con un filtro che realizza la media aritmetica con una finestra 15×15 (fig.35 a destra). Si può notare come un certo numero di oggetti abbiano diminuito fortemente la loro intensità confondendosi con lo sfondo. A questo punto risulta utile realizzare un'operazione di thresholding, proprio per eliminare gli oggetti cui compete una ridotta luminosità. Usando una soglia pari al 25% della massima intensità presente nell'immagine si ottiene la maschera binaria mostrata in figura 36 (a sinistra) e il risultato complessivo dell'elaborazione (a destra). Confrontando quest'ultima immagine con quella originale si può ritenere che effettivamente siano stati eliminati gli oggetti molto piccoli dall'immagine per poter esaminare meglio quelli più grandi.

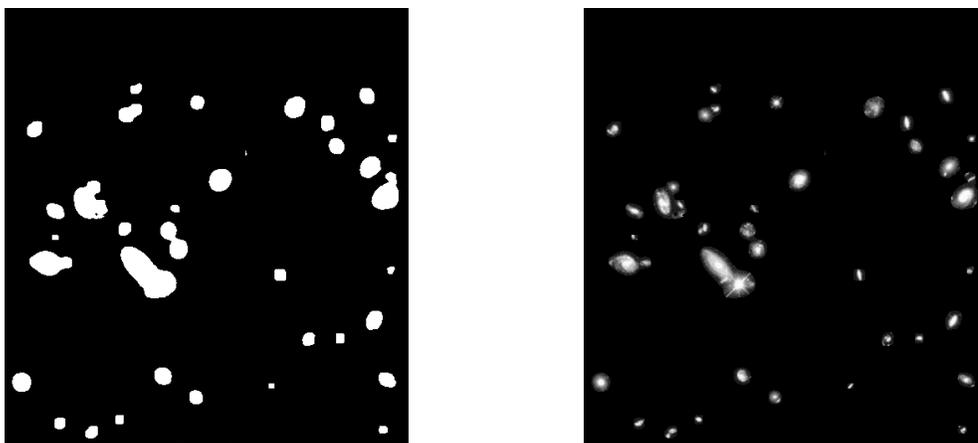


Figura 36: Immagine binaria risultato del test e risultato finale dell'elaborazione.

3.2 Filtri di sharpening

Obiettivo principale dei filtri di sharpening è quello di portare alla luce i dettagli di un'immagine, vengono quindi enfatizzati i bordi e le discontinuità. Questa operazione è estremamente utile quando si vuole fare *edge detection*, ma anche per realizzare l'enhancement di un'immagine, come vedremo di seguito.

Focalizzeremo la nostra attenzione sui filtri che si basano sulla derivata prima e seconda, sarà quindi necessario definire questa operazione per segnali tempo discreto monodimensionali per poi estenderla al caso bidimensionale. La derivata per segnali numerici viene definita in termini di differenze; dato che la minima distanza su cui può verificarsi una variazione è il pixel, possiamo definire la derivata prima di $x(n)$ come:

$$\frac{dx}{dn} = x(n+1) - x(n) \quad (13)$$

Evidentemente questo non è l'unico modo per definire la derivata, infatti in maniera equivalente si può anche definire come:

$$\frac{dx}{dn} = x(n) - x(n-1) \quad (14)$$

A questo punto possiamo definire la derivata seconda applicando due volte la differenza prima così come definita nella (13):

$$\frac{d^2x}{dn^2} = x(n+2) - 2x(n+1) + x(n) \quad (15)$$

Si noti, tuttavia, che in questo modo si accumula un ritardo rispetto alla posizione su cui è stata valutata la derivata. Per questo motivo è più conveniente applicare prima l'operatore (13) e poi il (14) in modo da ottenere la seguente definizione di derivata seconda:

$$\frac{d^2x}{dn^2} = x(n+1) - 2x(n) + x(n-1) \quad (16)$$

Nel caso bidimensionale la derivata può essere definita nelle 8 direzioni spaziali dell'immagine in particolare se restringiamo l'analisi alla direzione orizzontale e verticale, avremo per la derivata prima:

$$\frac{\partial x}{\partial m} = x(m+1, n) - x(m, n) \quad (17)$$

$$\frac{\partial x}{\partial n} = x(m, n+1) - x(m, n) \quad (18)$$

le operazioni così definite corrispondono alle seguenti maschere:

$$a_1(m, n) = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \quad a_2(m, n) = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}$$

mentre per la derivata seconda si ha:

$$\frac{\partial^2 x}{\partial m^2} = x(m-1, n) - 2x(m, n) + x(m+1, n) \quad (19)$$

$$\frac{\partial^2 x}{\partial n^2} = x(m, n-1) - 2x(m, n) + x(m, n+1) \quad (20)$$

con maschere corrispondenti pari a:

$$a_1(m, n) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad a_2(m, n) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Esempio: *Filtro Gradiente.* L'operatore gradiente per un'immagine si definisce come:

$$\nabla \mathbf{x} = \frac{\partial x}{\partial m} \mathbf{i}_m + \frac{\partial x}{\partial n} \mathbf{i}_n \quad (21)$$

Poché spesso si è interessati a usare filtri isotropici, la cui risposta risulta essere indipendente dalla direzione, si preferisce usare nell'elaborazione di un'immagine il filtro derivante dal modulo del gradiente:

$$|\nabla \mathbf{x}| = \sqrt{\left(\frac{\partial x}{\partial m}\right)^2 + \left(\frac{\partial x}{\partial n}\right)^2} \quad (22)$$

Spesso si fa riferimento alla definizione (22) come al *gradiente* dell'immagine $|\nabla \mathbf{x}| \equiv \nabla x$ (sebbene sia il modulo), e generalmente viene approssimato nel seguente modo:

$$\nabla x \simeq \left| \frac{\partial x}{\partial m} \right| + \left| \frac{\partial x}{\partial n} \right| \quad (23)$$

Questa equazione, più semplice da calcolare, si comporta ancora come una derivata: è nulla nelle aree di intensità costante e i valori sono proporzionali al livello di variazione di intensità nelle aree non omogenee.

Esempio: *Filtro Laplaciano.* Il più semplice filtro isotropico è il Laplaciano:

$$\nabla^2 x = \frac{\partial^2 x}{\partial m^2} + \frac{\partial^2 x}{\partial n^2} \quad (24)$$

Questo operatore ha l'interessante proprietà di produrre bordi in cui il passaggio per lo zero definisce esattamente la posizione della transizione. Sfruttando le definizioni (19) e (20) si ottiene allora:

$$\nabla^2 x = x(m+1, n) + x(m-1, n) - 4x(m, n) + x(m, n-1) + x(m, n+1)$$

Quindi la corrispondente maschera del filtro è:

$$a_1(m, n) = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Se si vogliono portare in conto anche le variazioni lungo le direzioni oblique, la maschera si può modificare nel seguente modo:

$$a_2(m, n) = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & -8 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Si noti come la somma dei coefficienti della maschera sia sempre uguale a zero, ciò coerentemente col fatto che la derivata deve essere nulla su regioni omogenee. In figura 37 si mostra il laplaciano dell'immagine superficie lunare, che può essere utilizzato per realizzare enhancement, semplicemente sommando l'immagine di partenza al laplaciano, cioè:

$$y(m, n) = x(m, n) - \nabla^2[x(m, n)]$$

In figura 38 si mostrano i risultati ottenuti dall'operazione appena descritta con le maschere $a_1(m, n)$ e $a_2(m, n)$, rispettivamente.

Esempio: *Definizione alternativa di derivata.* E' bene sottolineare che le (17) e (18) non sono le uniche possibili definizioni di derivata nel caso bidimensionale, ovviamente se modifichiamo la definizione cambia anche la maschera corrispondente. Per esempio, se si vuole considerare la variazione lungo la direzione diagonale, si possono considerare i filtri di *Roberts*, che presentano le seguenti maschere:

$$a_1(m, n) = \begin{array}{|c|c|} \hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array} \quad a_2(m, n) = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

3.3 Filtri non lineari

Un esempio di filtro spaziale non lineare è il filtro mediano, che non fa altro che sostituire al valore del pixel il valore mediano dei livelli di grigio nell'intorno del pixel stesso (definito dalla

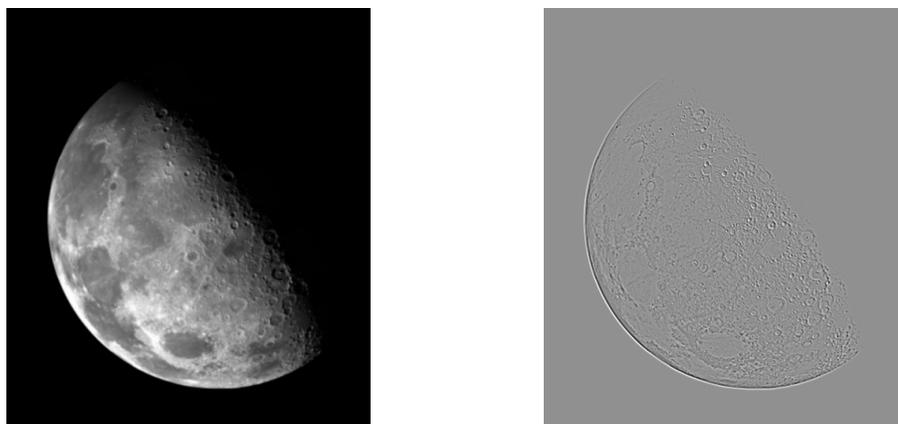


Figura 37: Immagine originale e suo laplaciano.



Figura 38: Enhancement dell'immagine superficie lunare.

maschera del filtro). I filtri mediani sono molto noti soprattutto perché riescono a ridurre (quasi perfettamente) alcuni tipi di rumore, evitando l'effetto di "blurring" tipico dei filtri di smoothing. I filtri mediani sono particolarmente efficaci in presenza di rumore impulsivo, noto come rumore sale e pepe, che ha l'aspetto di puntini bianchi e neri che si sovrappongono all'immagine.

Vediamo con un esempio come opera il filtro mediano. Supponiamo di considerare una finestra di dimensioni 3×3 , e supponiamo che i valori dei livelli di grigio siano: (3 255 0 25 36 24 39 250 29). Questi valori vengono ordinati: (0 3 24 25 29 36 39 250 255) e viene prelevato il valore centrale (valore mediano), in questo caso pari a 29. In questo modo i valori in uscita prodotti dal filtro tendono ad assumere valori più simili ai propri vicini, isolando cluster di pixel che sono troppo chiari (250, 255) o troppo scuri (0, 3) e che caratterizzano il rumore di tipo impulsivo. E' bene notare che un filtro che realizza la media aritmetica avrebbe sostituito il valore 73.4, tendendo così a creare l'effetto tipico del blurring. In figura 39 si mostra un esempio di elaborazione con un filtro mediano con dimensioni della finestra 3×3 .

Oltre al filtro mediano, in letteratura sono noti altri tipi di filtri come quelli che prelevano il valore massimo (max filter) o minimo (min filter) nella finestra considerata.

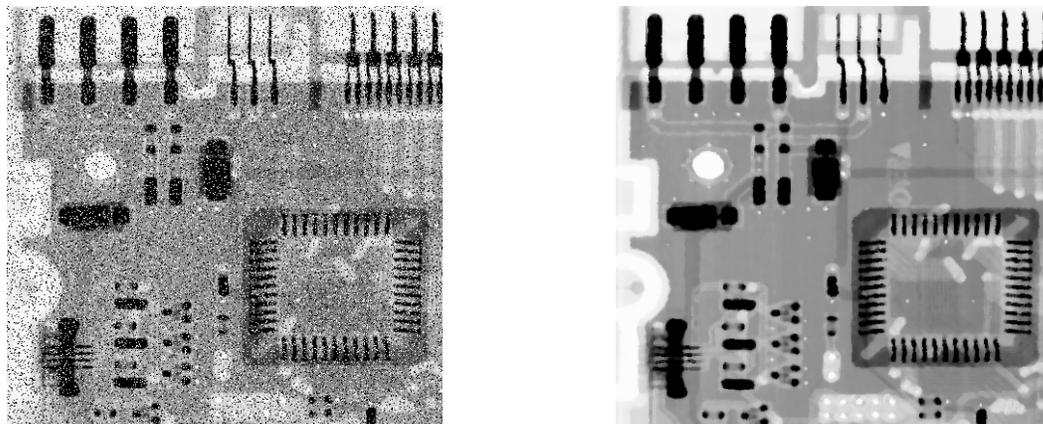


Figura 39: Immagine originale e immagine elaborata con un filtro mediano.

Riferimenti bibliografici

- [1] A.Bovik, *The essential guide to Image Processing*, Academic Press, 2009.
- [2] R.C.Gonzalez and R.E.Woods: *Digital Image Processing, 3rd Ed.*, Prentice Hall, 2008.
- [3] J.G.Proakis and D.G.Manolakis: *Digital Signal Processing, 3rd Ed.*, Prentice Hall.