

Laboratorio di multimedialità

A.A. 2024-25
Andrea Macale

Indice

I Immagini	1
1 Elaborazione delle immagini	2
1.1 Sistema visivo umano	2
1.1.1 Cornea e coroide: due membrane come protezione	2
1.1.2 Retina: la membrana per la vista	3
1.1.3 Formazione dell'immagine	3
1.2 Rappresentazione delle immagini	4
1.2.1 Intensità di un'immagine	4
1.2.2 Digitalizzazione delle immagini	4
1.3 Manipolazione delle immagini	5
1.3.1 Manipolazioni principali	6
1.3.2 Funzione di trasformazione lineare a tratti	9
1.3.3 Equalizzazione dell'istogramma	10
2 Filtri nel dominio spaziale	13
2.1 Filtri di smussamento	14
2.1.1 Filtri di smussamento lineari	14
2.1.2 Filtri di smussamento non lineari	15
2.2 Filtri di nitidezza	16
2.2.1 Il Laplaciano	16
2.2.2 Filtri di contrasto	17
2.2.3 Filtri di rilevamento dei bordi	18
3 Filtri nel dominio della frequenza	21
3.1 Dominio della frequenza	21
3.1.1 Trasformata di Fourier in due dimensioni	22
3.1.2 Proprietà della trasformata di Fourier	22
3.1.3 Teorema della convoluzione e applicazione al filtraggio	23
3.2 Filtro di Notch	23
3.3 Filtri passa-basso	24
3.3.1 Filtro passa-basso ideale	24
3.3.2 Filtro passa-basso di Butterworth	25
3.3.3 Filtro passa-basso Gaussiano	25
3.4 Filtri passa-alto	26
3.4.1 Filtro passa-alto ideale, di Butterworth e Gaussiano	26
3.4.2 Il Laplaciano, i filtri di contrasto e l'high-boost filtering	27
4 Trasformata Wavelet	28
4.1 Limiti della trasformata di Fourier e possibile soluzione	28
4.1.1 Caratteristiche di un'immagine	28
4.1.2 Problemi della trasformata di Fourier	29
4.1.3 Una soluzione temporanea: la trasformata di Fourier a breve termine	29
4.2 Introduzione alla trasformata Wavelet	30

4.2.1	Determinazione della trasformata Wavelet	31
4.2.2	Proprietà	31
4.3	Applicazione della trasformata Wavelet: l'analisi multirisoluzione	32
4.3.1	Creazione delle piramide dell'immagine	32

Elenco delle figure

1	Struttura dell'occhio umano	2
2	Coni e bastoncelli	3
3	Illusione del mais dolce (a sinistra) e la griglia di Hermann (a destra)	4
4	Esempi di possibili manipolazioni di immagini	5
5	Possibili manipolazioni delle immagini	6
6	Immagine originale (a sinistra) e immagine in negativo (a destra)	7
7	Esempio di due immagini a cui sono state applicate con r differenti	7
8	Realizzazione della correzione gamma sui monitor	8
9	Calibrazione del gamma (azzurro la gamma dell'immagine, viola la gamma del display e la rossa la gamma complessiva)	8
10	Definizione analitica e grafico del contrast stretching	9
11	Definizione analitica e grafico del thresholding	9
12	Gray-level slicing a sfondo costante (a sinistra) e a sfondo invariato (a destra)	10
13	Esempio di istogramma di un'immagine 4x4 con livello di grigio [0,9]	11
14	Calcolo dell'istogramma normalizzato	12
15	Esempio di istogramma di un'immagine equalizzata	12
16	Filtro media e gaussiano	14
17	Rimozione del rumore sale e pepe con un filtro mediano (sinistra originale e destra modificata)	15
18	Applicazione del Laplaciano (sinistra originale e destra modificata)	17
19	Originale, Laplaciano filtro di contrasto e high-boost filtering da in alto a sinistra in senso orario	18
20	Immagine originale (a sinistra) e immagine con i bordi rilevati con il rilevatore di Canny (a destra)	20
21	Rappresentazione della trasformata di Fourier	21
22	Immagine nel dominio spaziale (a sinistra) ed in frequenza (a destra)	22
23	Filtraggio nel dominio della frequenza	23
24	Immagine originale (a sinistra) e immagine filtrata con un filtro Notch (a destra)	23
25	Funzionamento di un filtro passa-basso ideale	24
26	Da sinistra: immagine originale, LPF ideale $D_0 = 8$ e LPF ideale $D_0 = 16$	24
27	Vari ordini del filtro passa-basso di Butterworth	25
28	Filtri passa-basso Gaussiani	25
29	Da alto a sinistra senso orario: originale, HPF ideale, HPF di Butterworth e HPF Gaussiano	26
30	Immagine originale (a sinistra) e maschera del filtro Laplaciano (a destra)	27
31	Variazione delle frequenze nell'asse temporale da $t_0 = 0\text{ s}$ a $t_3 = 1\text{ s}$	28
32	Rappresentazione dei problemi della trasformata di Fourier	29
33	Esecuzione della STFT	29
34	Tipologie wavelet fondamentali	30
35	Rappresentazione di una trasformata Wavelet	31
36	Analisi dei dettagli delle serie	32

PARTE I:

IMMAGINI

Capitolo 1:

Elaborazione delle immagini

In questo capitolo viene spiegato come avviene l'elaborazione delle immagini. Per prima cosa, è necessario capire come l'occhio umano cattura e percepisce l'immagine, soprattutto per comprendere quali sono i suoi grandi limiti. Una volta capito ciò, si può procedere all'elaborazione delle immagini.

1.1: Sistema visivo umano

L'occhio umano è racchiuso da tre membrane, dove ognuna ha una funzione rilevante per l'elaborazione delle immagini.

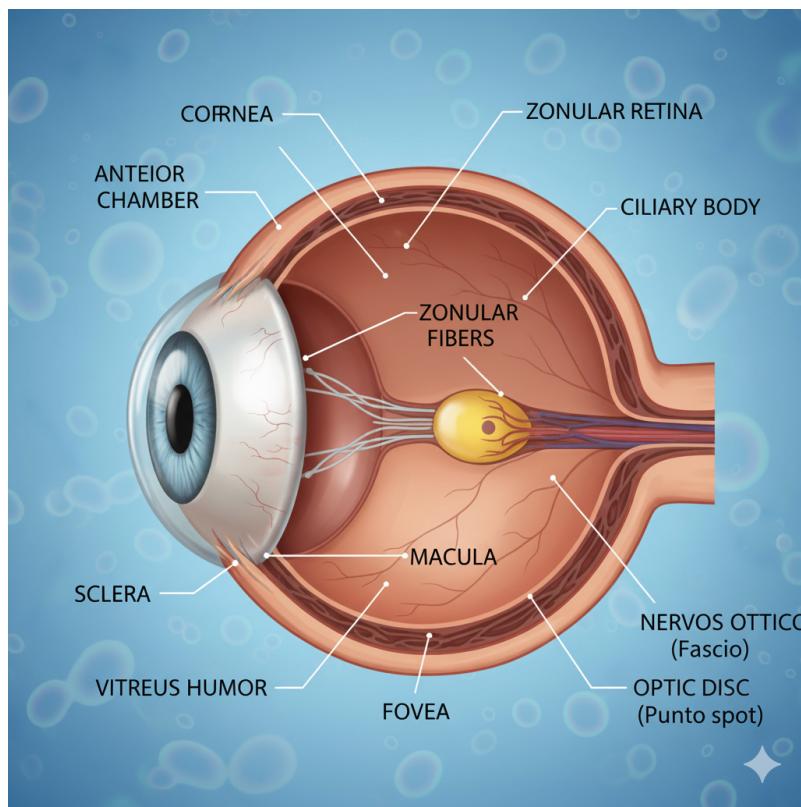


Figura 1: Struttura dell'occhio umano

1.1.1: Cornea e coroide: due membrane come protezione

La cornea è la membrana più esterna dell'occhio umano: infatti, essendo composta da un tessuto resistente e trasparente, è perfetta per racchiudere la superficie anteriore dell'occhio. Inoltre, per ridurre la quantità di luce estranea che entra nell'occhio, è presente la coroide.

1.1.2: Retina: la membrana per la vista

La retina è quella membrana che fornisce il senso della vista all'essere umano. In particolare, permette di mettere a fuoco gli oggetti, grazie alla luce dell'oggetto stesso che entra nella retina.

Inoltre, sono presenti i cosiddetti percettori luminosi, che rendono possibile la visione a pattern.

Il primo percettore sono i coni, che permettono la visione cromatica: infatti possono essere a lunghezza d'onda corta per il blu, a lunghezza d'onda media per il verde ed a lunghezza d'onda lunga per il rosso.

Per la visione acromatica, invece, sono presenti i bastoncelli, per esempio la visione scotopica e la penombra.

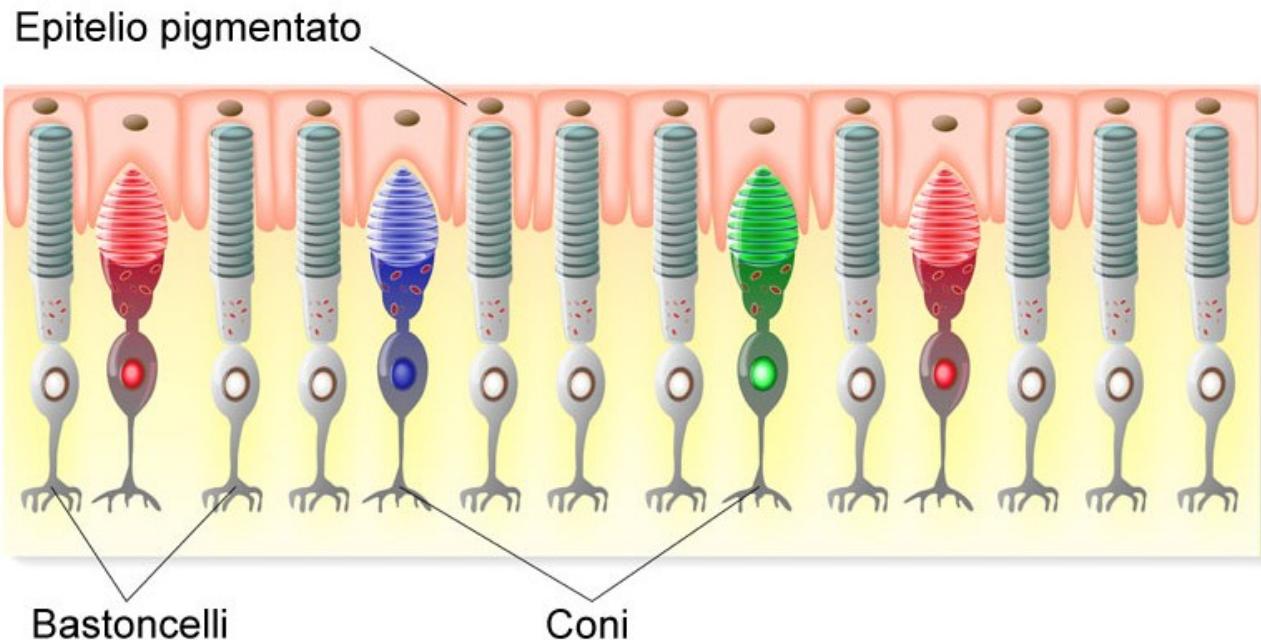


Figura 2: Coni e bastoncelli

1.1.3: Formazione dell'immagine

A questo punto, la formazione dell'immagine avviene nel modo seguente. Le lenti dell'occhio umano sono flessibili: la sua forma è controllata delle fibre del corpo ciliare.

Inoltre, l'abilità dell'occhio di discriminare i cambiamenti delle intensità di luce a qualsiasi livello di adattibilità specifico, è descritto dalla legge di Weber.

$$k = \frac{\Delta I_C}{I}$$

In particolare:

- k è la costante di Weber, che è un valore costante e caratteristico per ogni specifica modalità sensoriale;
- ΔI_C è la soglia differenziale, ossia la quantità minima di cambiamento affinché il soggetto percepisca una differenza;
- I è l'intensità di riferimento dello stimolo.

Ciò spiega che il sistema visivo tende a sottostimare od a sovrastimare i bordi delle regioni a diverse intensità. Ciò genera le illusioni: esempi sono l'illusione di dolcezza di mais e la griglia di Hermann.

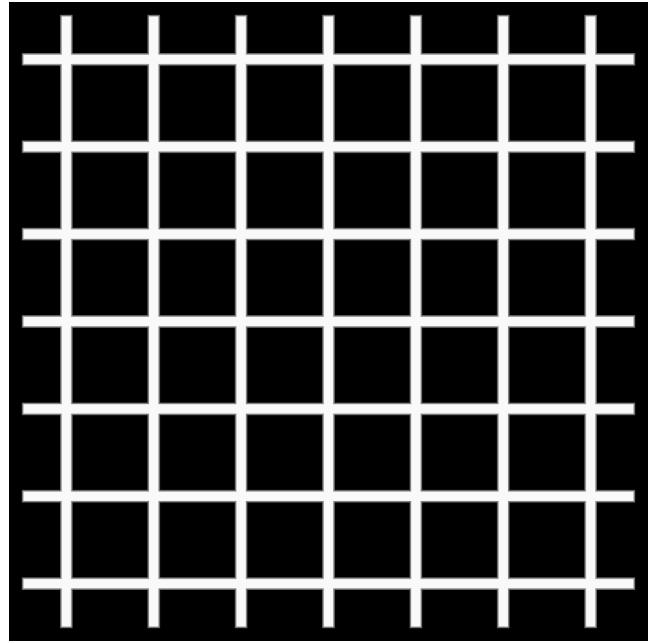


Figura 3: Illusione del mais dolce (a sinistra) e la griglia di Hermann (a destra)

1.2: Rappresentazione delle immagini

Un'immagine è una funzione bidimensionale ($f(x, y)$), definita come il prodotto tra altrettante due funzioni bidimensionali:

- la luminanza ($i(x, y)$), che è la quantità di luce della sorgente incidente sulla scena osservata;
- la riflettanza ($r(x, y)$), che è la quantità di luce riflessa dall'oggetto nella scena.

$$\begin{aligned} f(x, y) &= i(x, y)r(x, y) \\ 0 < i(x, y) &< \infty \\ 0 < r(x, y) &< 1 \\ 0 < f(x, y) &< \infty \end{aligned}$$

1.2.1: Intensità di un'immagine

L'intensità di un'immagine monocromatica a (x_0, y_0) è il livello di grigio L in quel punto.

$$L = f(x_0, y_0)$$

Il livello di grigio trovato appartiene ad un intervallo dove il minimo corrisponde al nero ed il massimo corrisponde al massimo.

$$L \in [L_{\min}, L_{\max}]$$

1.2.2: Digitalizzazione delle immagini

Per effettuare la digitalizzazione di un'immagine, sono necessari i seguenti parametri:

- M , che è la larghezza dell'immagine;
- N , che è l'altezza dell'immagine;
- L , che è il numero dei livelli di grigi, che è un multiplo di 2^n .

A questo punto, avviene la digitalizzazione, che si compone in due fasi.

La prima fase è il campionamento, che consiste nel suddividere l'immagine in una griglia regolare di punti o celle. Ogni cella produce un pixel, che è l'unità minima di un'immagine digitale. A questo punto viene misurata la risoluzione in dpi, che non è altro che la densità della griglia: più è fitta più l'immagine sarà fedele all'originale.

La seconda ed ultima fase è la quantizzazione, che è il processo di discretizzazione dell'intensità. In particolare, ad ogni pixel gli viene assegnato un valore numerico discreto che ne codifica il livello di grigio. Infine, il valore numerico viene convertito in una stringa binaria.

Il risultato della digitalizzazione, produce una matrice di numeri reali di dimensioni $M \times N$.

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

1.3: Manipolazione delle immagini

La manipolazione delle immagini si pone come obiettivo principale quello di modificare l'immagine originale che si adatta di più al contesto richiesto: come sfocare lo sfondo in un ritratto oppure rendere più luminose le vene in un'immagine catturata da un'esame medico.

In questo paragrafo, vengono elencati una serie di possibili manipolazioni, andando a modificare la $f(x, y)$ con un operatore T definito con i suoi vicini (x, y) .

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

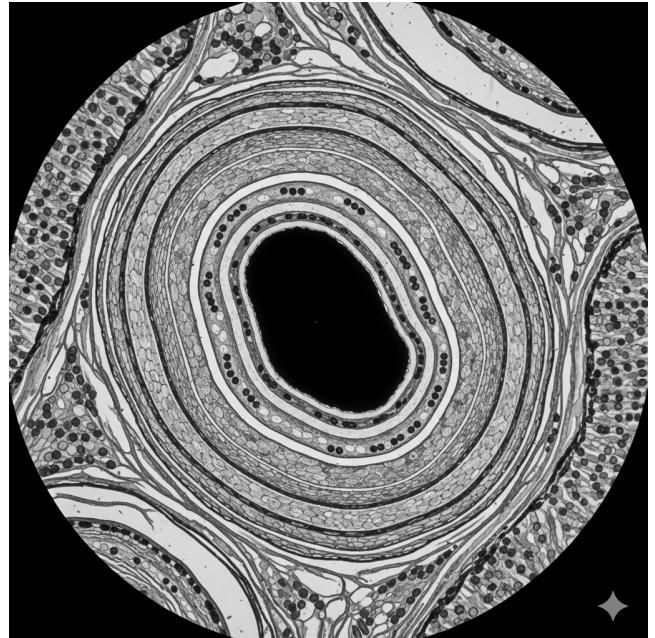


Figura 4: Esempi di possibili manipolazioni di immagini

In particolare, si considera:

- r è il livello di grigio di input;
- s è il livello di griglio di output.

1.3.1: Manipolazioni principali

Di seguito ne è riportato un grafico che ne riporta le principali tipologie di manipolazione delle immagini.

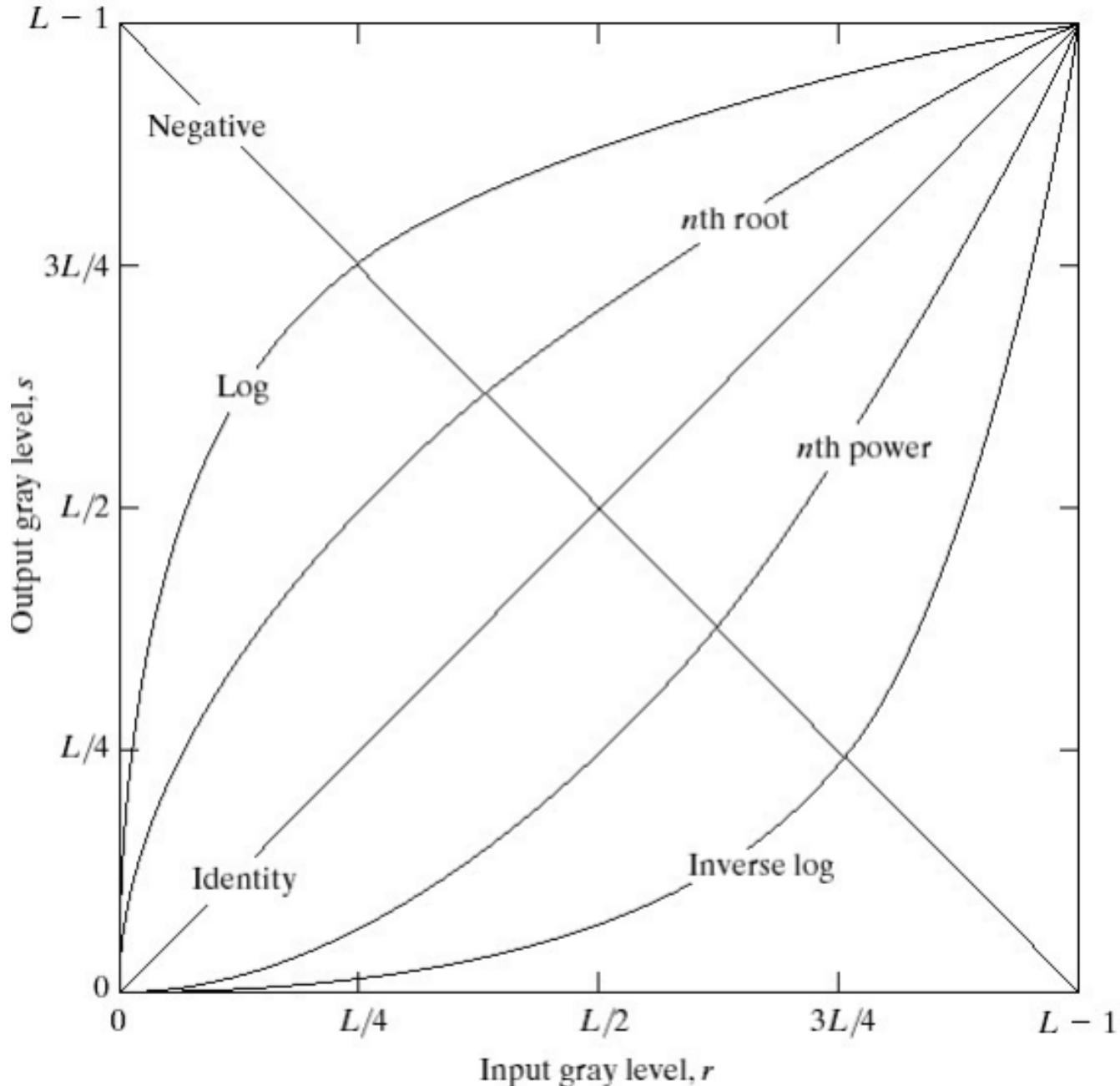


Figura 5: Possibili manipolazioni delle immagini

La prima ed anche la più semplice è la funzione identità, in cui il livello di grigio in entrata corrisponde a quello d'uscita, perciò l'immagine non riporta alcuna alterazione. Tale funzione è una retta che va da $(0, 0)$ a $(L - 1, L - 1)$.

$$s = r$$

La seconda è la funzione negativa, in cui inverte i livelli di grigio, creando la cosiddetta immagine negativa. Questa funzione non è altro che una retta che va da $(0, L - 1)$ a $(L - 1, 0)$.

$$s = L - 1 - r$$

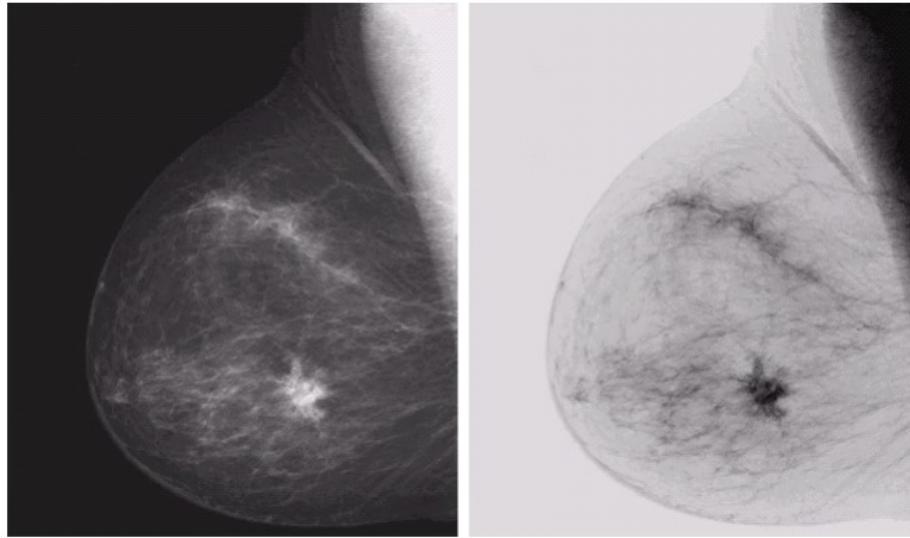


Figura 6: Immagine originale (a sinistra) e immagine in negativo (a destra)

Tale funzione risulta molto utile per risaltare regioni di grigio incorporate da regioni scure.

La terza funzione è la funzione logaritmica, che tende ad espandere i valori dei pixel scuri ed ad comprimere i pixel chiari. Se invece si vuole ottenere l'opposto, allora si usa la funzione logaritmica inversa. La prima equazione è la funzione logaritmica, mentre la seconda è la funzione logaritmica inversa. Entrambi sfruttano la costante di scala $r \in [0, L - 1]$.

$$s = c \ln(1 + r)$$

$$s = \exp\left(\frac{r}{c}\right) - 1$$

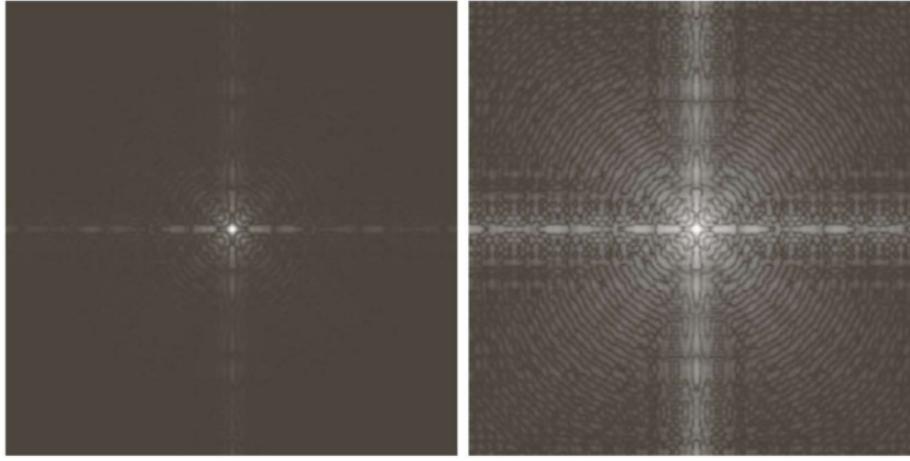


Figura 7: Esempio di due immagini a cui sono state applicate con r differenti

L'ultimo gruppo di funzioni riguarda la correzione gamma, che è un'operazione non lineare usata per codificare e decodificare i valori di luminanza in un sistema di visualizzazione di immagini. Esse sfruttano le costanti c e γ , entrambe positive.

$$s = cr^\gamma$$

Ciò avviene per due motivi principali:

- l'occhio umano non percepisce la luminosità in modo lineare, perciò le immagini risulterebbero scure e con pochi dettagli nelle ombre;
- gli schermi più datati, come quelli a tubo catodico, hanno una risposta non lineare alla tensione in ingresso, seguendo una legge di potenza, e si è scelto di mantenere ciò anche per gli schermi più moderni per mantenere compatibilità e coerenza visiva per la percezione umana.

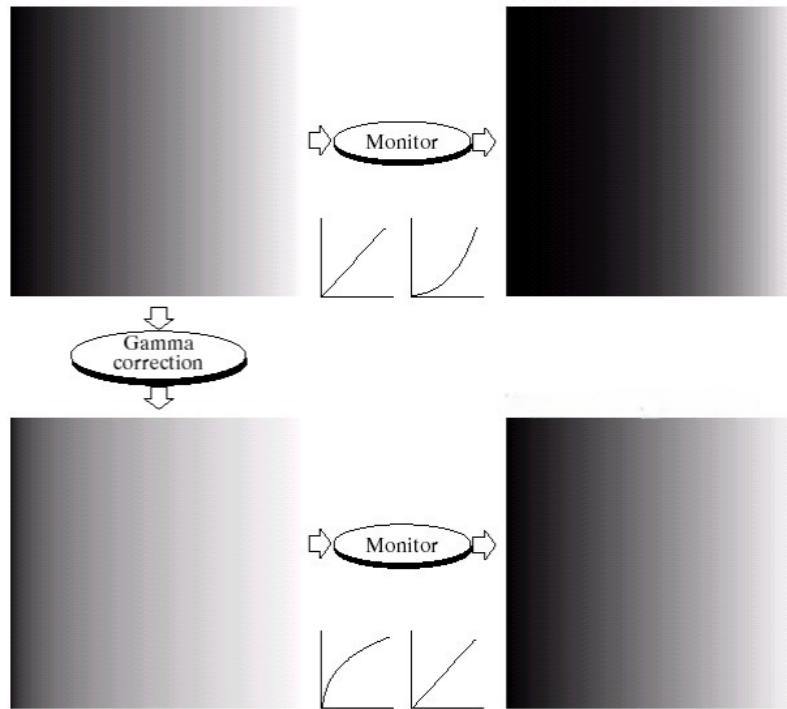


Figura 8: Realizzazione della correzione gamma sui monitor

Di seguito, viene riportata una figura che dimostra quando sia importante la calibrazione corretta della correzione gamma sui monitor.

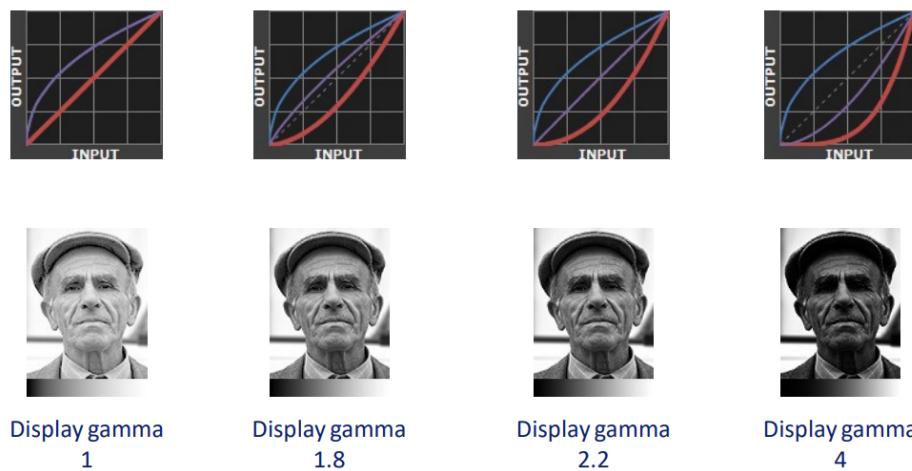


Figura 9: Calibrazione del gamma (azzurro la gamma dell'immagine, viola la gamma del display e la rossa la gamma complessiva)

Nella prima colonna non è stata applicata alcuna correzione; nella seconda l'applicazione è insufficiente, generando un'immagine troppo chiara; nella terza è l'applicazione ideale; infine nell'ultima l'applicazione è eccessiva generando un'immagine troppo scura.

1.3.2: Funzione di trasformazione lineare a tratti

La funzione di trasformazione lineare a tratti ha la particolarità di non essere descritta da una singola equazione per tutto l'intervallo dei livelli di grigio, ma da più segmenti lineari, ognuno applicabile ad un intervallo di livello di grigio ben specifico. Il vantaggio sta nell'avere una forma con livello di complessità a scelta, a discapito, però, nell'avere più input dall'utente.

La prima funzione di trasformazione lineare a tratti è il contrast stretching, che si pone come obiettivo quello di aumentare il contrasto di un'immagine che appare sbiadita o con scarso intervallo dinamico. A questo punto, si scelgono due punti (r_1, s_1) e (r_2, s_2) in cui i livelli sotto r_1 (molto scuri) oppure sopra r_2 (molto chiari) vengono compressi, in altre parole vengono mappati rispettivamente quasi a 0 e a $L - 1$. Mentre se è compreso tra r_1 ed r_2 viene aumentato il contrasto, espandendo i livelli di grigio sull'intero intervallo di uscita.

$$s = \begin{cases} \frac{s_1}{r_1} r & \text{se } 0 \leq r < r_1 \\ \frac{s_2 - s_1}{r_2 - r_1}(r - r_1) + s_1 & \text{se } r_1 \leq r \leq r_2 \\ \frac{L - 1 - s_2}{L - 1 - r_2}(r - r_1) + s_2 & \text{se } r_2 < r \leq L - 1 \end{cases}$$

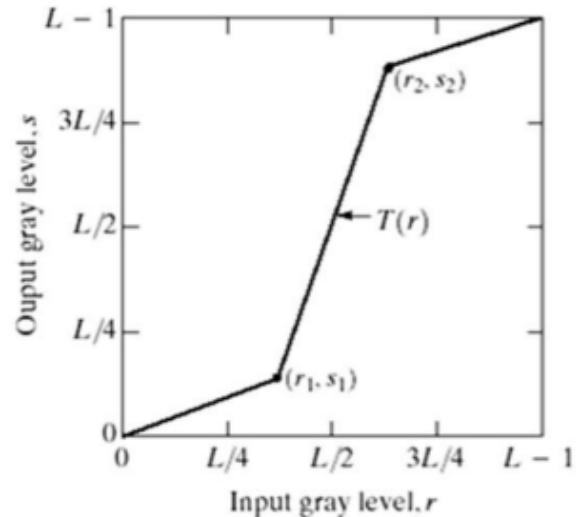


Figura 10: Definizione analitica e grafico del contrast stretching

Un caso limite si ha se avviene un'immagine binaria (bianco o nero), avendo il cosiddetto thresholding.

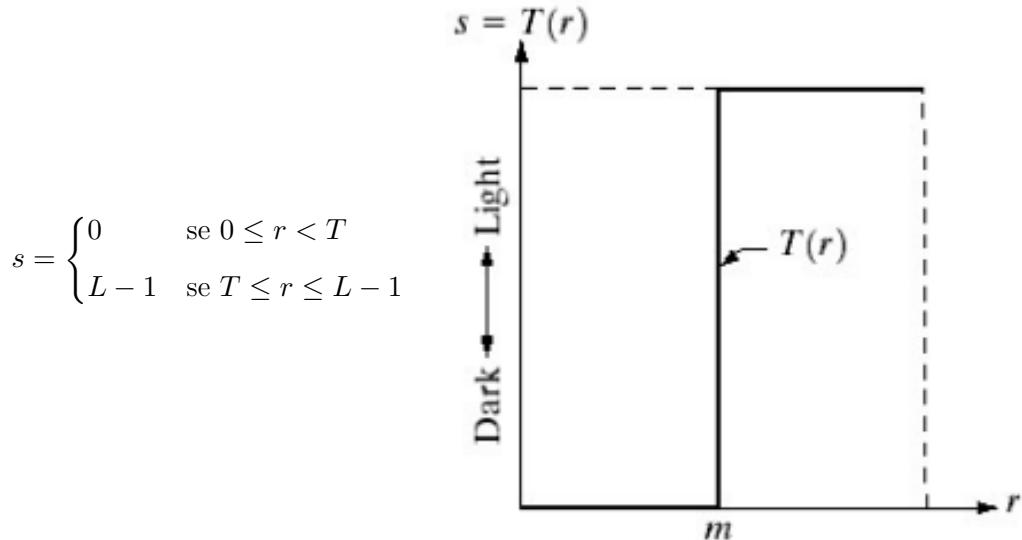


Figura 11: Definizione analitica e grafico del thresholding

La seconda funzione di trasformazione lineare a tratti è il gray-level slicing, che si pone come obiettivo quello di evidenziare un intervallo di livello di grigio ben specifico e sopprimere tutti gli altri. In particolare si fa riferimento a due casi ben specifici:

- sfondo costante, in cui l'intervallo viene mappato ad un livello alto ($L - 1$) ed il resto al livello 0;

$$s = s_{high} \text{rect}_{B-A} \left(r - \frac{A+B}{2} \right)$$

- sfondo invariato, in cui l'intervallo viene mappato ad un livello alto ($L - 1$) ed il resto viene lasciato invariato.

$$s = r + (s_{high} - r) \text{rect}_{B-A} \left(r - \frac{A+B}{2} \right)$$

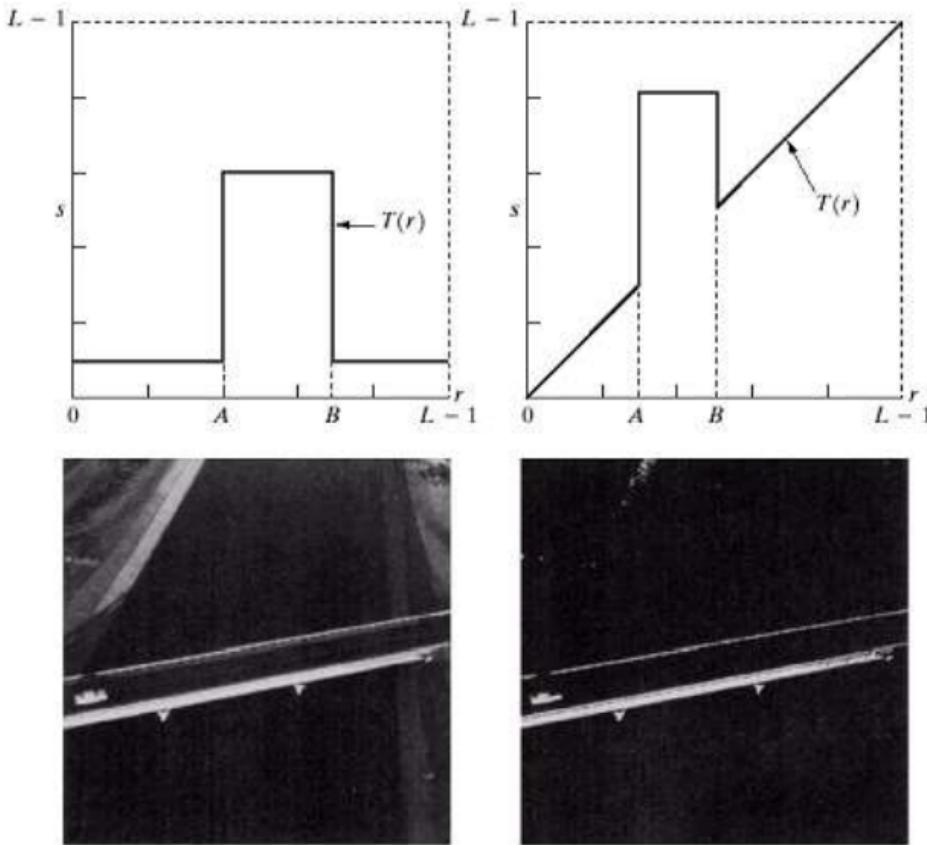


Figura 12: Gray-level slicing a sfondo costante (a sinistra) e a sfondo invariato (a destra)

La terza ed ultima funzione di trasformazione lineare a tratti è il bit-plain slicing, che consiste nel mettere evidenza solamente i bit più significativi (che sono i primi bit).

1.3.3: Equalizzazione dell'istogramma

Quando si parla di un'immagine, è possibile costruire l'istogramma dell'immagine stessa, che non è altro che il numero di pixel che contengono un determinato valore di livello di grigio per ogni livello n_k .

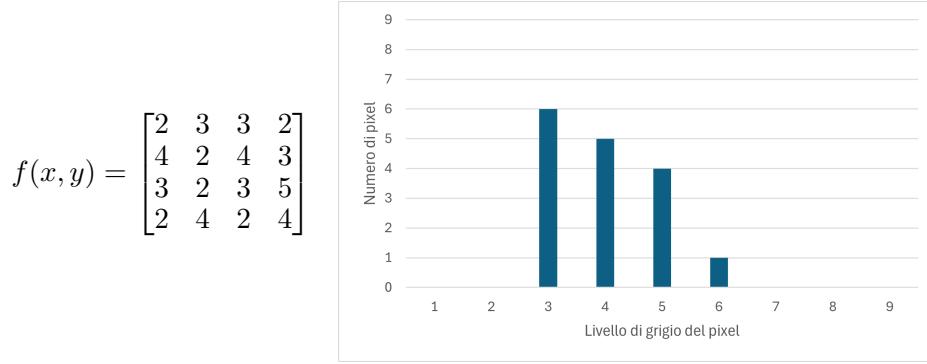


Figura 13: Esempio di istogramma di un’immagine 4x4 con livello di grigio [0, 9]

In termini statistici, è preferibile rappresentare l’istogramma normalizzato, ossia dividere n_k con il numero totale di pixel n .

$$p(n_k) = \frac{n_k}{n}$$

A questo punto, la tecnica dell’equalizzazione dell’istogramma consiste nel cambiare l’istogramma dell’immagine in un’istogramma uniforme, in cui la percentuale di ogni livello di grigio rimane sempre la stessa. Per effettuare ciò, occorre dei fondamenti di probabilità e statistica e di calcolo integrale.

Per prima cosa, questa operazione effettua una trasformata, perciò si può scrivere ciò come $s = T(r)$ e siccome la trasformata è reversibile, esiste anche la sua inversa $r = T^{-1}(s)$. Per semplicità, risulta molto conveniente studiare ciò nel continuo. Dati rispettivamente, $p_{in}(r)$ e $p_{out}(s)$, come la probabilità di livello di grigi di input e di output, dalla teoria di probabilità, si ha la formula seguente (per $0 \leq r \leq L - 1$ e $0 \leq s \leq L - 1$).

$$p_{out}(s) = \left[p_{in}(r) \frac{ds}{dr} \right]_{r=T^{-1}(s)}$$

A questo punto, la trasformata di $p_{in}(r)$ è pari alla formula seguente:

$$s = T(r) = \int_0^r p_{in}(r') dr', 0 \leq r \leq 1$$

che è la funzione di distribuzione cumulativa (CDF). Perciò, dal teorema fondamentale del calcolo, si sfrutta la seguente formula.

$$p_{in}(r) = \frac{dr}{ds}$$

Infine, facendo dei semplici conti e sfruttando le proprietà delle derivate delle funzioni inverse, si ricava che:

$$p_{out}(s) = \left[p_{in}(r) \frac{ds}{dr} \right]_{r=T^{-1}(s)} = \left[p_{in}(r) \frac{1}{\frac{dr}{ds}} \right]_{r=T^{-1}(s)} = \left[p_{in}(r) \frac{1}{p_{in}(r)} \right]_{r=T^{-1}(s)} = [1]_{r=T^{-1}(s)} = 1, 0 \leq s \leq 1$$

la densità di probabilità in uscita risulta uniforme.

Quindi per effettuare l’equalizzazione dell’istogramma:

1. per ogni pixel si calcola il $p_{in}(r_k)$;

$$p_{in}(r_k) = \frac{n_k}{n}, 0 \leq r_k \leq 1, 0 \leq k \leq L - 1$$

2. basandosi sulla CDF, si esegue la trasformata discreta.

$$s_k = T_{r_k} = \sum_{j=0}^k p_{in}(r_j), 0 \leq k \leq L - 1$$

Tornando all'esempio precedente, si effettua la CDF nel modo seguente.

$$f(x, y) = \begin{bmatrix} 2 & 3 & 3 & 2 \\ 4 & 2 & 4 & 3 \\ 3 & 2 & 3 & 5 \\ 2 & 4 & 2 & 4 \end{bmatrix}$$

k	r_k	n_k	$p_{in}(r_k)$
2	0	6	3/8
3	1/3	5	5/16
4	2/3	4	1/4
5	1	1	1/16

Figura 14: Calcolo dell'istogramma normalizzato

$$\begin{aligned} s_2 &= p_{in}(r_2) = \frac{3}{8} \rightarrow 0 \\ s_3 &= p_{in}(r_3) = \frac{3}{8} + \frac{5}{16} = \frac{11}{16} \rightarrow \frac{2}{3} \\ s_4 &= p_{in}(r_4) = \frac{3}{8} + \frac{5}{16} + \frac{1}{4} = \frac{15}{16} \rightarrow 1 \\ s_5 &= p_{in}(r_5) = \frac{3}{8} + \frac{5}{16} + \frac{1}{4} + \frac{1}{16} = 1 \rightarrow 1 \end{aligned}$$

Perciò:

- al livello 2 si associa il livello 2;
- al livello 3 si associa il livello 4;
- al livello 4 si associa il livello 5;
- al livello 5 si associa il livello 5.

A questo punto, si ottiene l'immagine equalizzata, come mostrato di seguito.

$$g(x, y) = \begin{bmatrix} 2 & 4 & 4 & 2 \\ 5 & 2 & 5 & 4 \\ 4 & 2 & 4 & 5 \\ 2 & 5 & 2 & 5 \end{bmatrix}$$

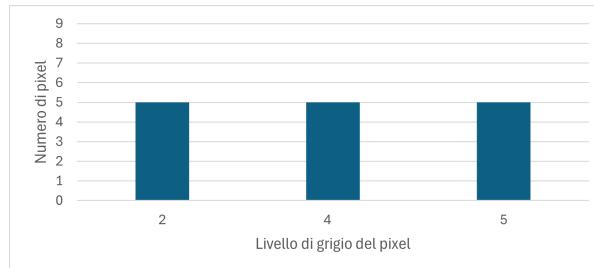


Figura 15: Esempio di istogramma di un'immagine equalizzata

Capitolo 2:

Filtri nel dominio spaziale

Nelle telecomunicazioni, per filtrare un segnale nel dominio del tempo, si l'operazione di convoluzione.

$$x(t) * h(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau) d\tau \quad x[n] * h[n] = \sum_{i=-\infty}^{\infty} h[i]x[n-i]$$

Per quanto riguarda le immagini, nel dominio spaziale, la convoluzione avviene in due dimensioni (somma del prodotto elemento per elemento delle due matrici), dove:

- l'immagine è il segnale d'ingresso;
- il filtro è il nucleo della convoluzione, detto maschera.

$$g(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(i, j)f(x - i, y - j)$$

Per esempio, data l'immagine e il filtro seguente:

$$f(x, y) = \begin{bmatrix} 5 & 8 & 3 & 4 \\ 3 & 2 & 1 & 1 \\ 0 & 9 & 5 & 3 \\ 4 & 2 & 7 & 2 \end{bmatrix} \quad h(x, y) = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{bmatrix} \quad h_{flip}(x, y) = - \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{bmatrix} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

si mette all'inizio in posizione $(0, 0)$ dell'immagine e quello è il punto centrale dell'immagine. Come si può notare, sono presenti dei punti dell'immagine che vanno fuori dall'immagine. Per evitare ciò, è possibile:

- ignorare i bordi, partendo dai punti in cui si ha una piena sovrapposizione dell'immagine;
- si assumono i valori fuori dai bordi pari a 0.

$$g(0, 0) = \begin{bmatrix} -2 & -1 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 2 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 \\ 0 & 5 & 8 \\ 0 & 3 & 5 \end{bmatrix} = -1 \times 5 + 1 \times 8 + 1 \times 3 + 5 \times 5 = 20$$

L'immagine finale, ponendo i punti fuori dall'immagine pari a 0, è dunque la seguente.

$$g(x, y) = \begin{bmatrix} 20 & 10 & 2 & 2 \\ 18 & 1 & -8 & -7 \\ 14 & 22 & 5 & -3 \\ 6 & -4 & -16 & -18 \end{bmatrix}$$

Nel caso in cui si ignorassero i bordi, si applicherebbe il filtro solamente nei punti $(1, 1)$, $(1, 2)$, $(2, 1)$ e $(2, 2)$. Come si può intuire, l'immagine di output viene tagliata.

$$g(x, y) = \begin{bmatrix} 3 & 10 \\ -4 & 5 \end{bmatrix}$$

Infine, eseguire convoluzioni con maschere risulta essere molto versatile, poiché a seconda dei coefficienti della maschera, si possono ottenere risultati differenti: per esempio la sfocatura, il contrasto od il rilevamento dei bordi. In questo capitolo, vengono spiegati i filtri di smussamento e di nitidezza.

2.1: Filtri di smussamento

I filtri di smussamento permettono di gestire la sfocatura dell'immagine e della riduzione del rumore. In particolare, la sfocatura è un processo che serve per rimuovere piccoli dettagli e colmare piccoli scalini in linee e curve. La sfocatura accompagna la riduzione del rumore.

2.1.1: Filtri di smussamento lineari

I filtri di smussamento lineari applicano la media dei pixel del vicinato. Essi ripiazzano il valore di ogni pixel con la media dei livello di grigio definiti dalla maschera. In particolare, a seconda dei valori della maschera si hanno tre tipologie di filtro.

La prima tipologia è il filtro media aritmetica, in cui la maschera contiene il prodotto scalare tra l'inverso del numero di elementi della maschera e la maschera stessa di tutti 1.

$$h(x, y) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h(x, y) = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Tale filtro applica una sfocatura uniforme all'immagine: più la maschera contiene elementi, maggiore sarà la sfocatura. La seconda tipologia è il filtro media ponderata, in cui si vuole dare più importanza a dei pixel rispetto ad altri, ad esempio il pixel centrale.

$$h(x, y) = \begin{bmatrix} 3/40 & 1/8 & 3/40 \\ 1/8 & 1/5 & 1/8 \\ 3/40 & 1/8 & 3/40 \end{bmatrix}$$

La terza ed ultima tipologia è il filtro gaussiano, in cui la maschera è una gaussiana a due dimensioni, in cui la deviazione standard (σ) ne determina la larghezza della campana.

$$h(x, y) = \exp \left[\frac{-(x^2 + y^2)}{2\sigma^2} \right]$$

La deviazione standard controlla l'intensità della sfocatura: più è alto più sfoca. Comunque, essa sfoca molto meno brutalmente rispetto al filtro media.

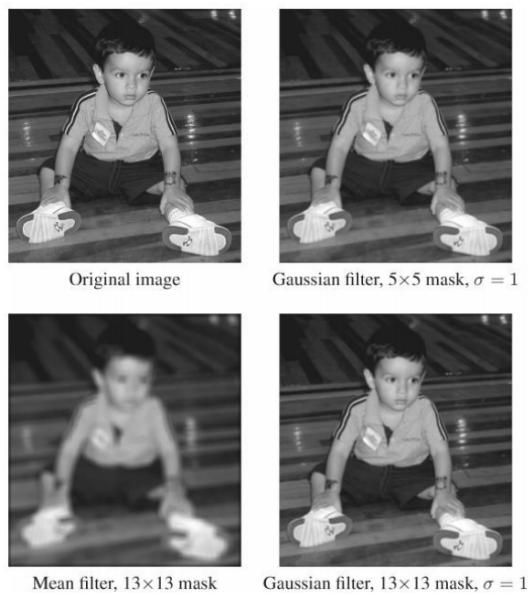


Figura 16: Filtro media e gaussiano

2.1.2: *Filtri di smussamento non lineari*

I filtri di smussamento non lineari non presentano un'operazione di convoluzione vera e propria, ma si basano su ordinamento statistico: in particolare, in base al ranking dei pixel. I pixel che sono considerati non rappresentativi vengono eliminati. Inoltre, vengono sostituiti i valori dei pixel con il valore della classifica. Questa tipologia viene usata per la riduzione del rumore, sfocando però l'immagine.

Il filtro di smussamento non lineare più comune è il filtro mediano, che consiste nel:

1. prendere una porzione di un'immagine (solitamente 3x3 o 5x5);

$$\begin{bmatrix} 9 & 12 & 0 \\ 5 & 5 & 9 \\ 8 & 10 & 7 \end{bmatrix}$$

2. convertire tale porzione in un array ordinato;

$$\{0, 5, 5, 7, 8, 9, 9, 10, 12\} \rightarrow 8$$

3. sostituire la mediana dell'array nella porzione presa, sfocando l'immagine;

$$\begin{bmatrix} 8 & 8 & 8 \\ 8 & 8 & 8 \\ 8 & 8 & 8 \end{bmatrix}$$

4. iterare i passaggi precedenti per tutta l'immagine.

Questo filtro permette di rimuovere il rumore sale e pepe, molto comune nelle foto più datate, in cui l'immagine presenta dei pixel completamente bianchi e neri: infatti, tali pixel non saranno mai dei mediani: per questo motivo vengono eliminati. Inoltre, la sfocatura non è così evidente, un altro lato positivo da non sottovalutare.

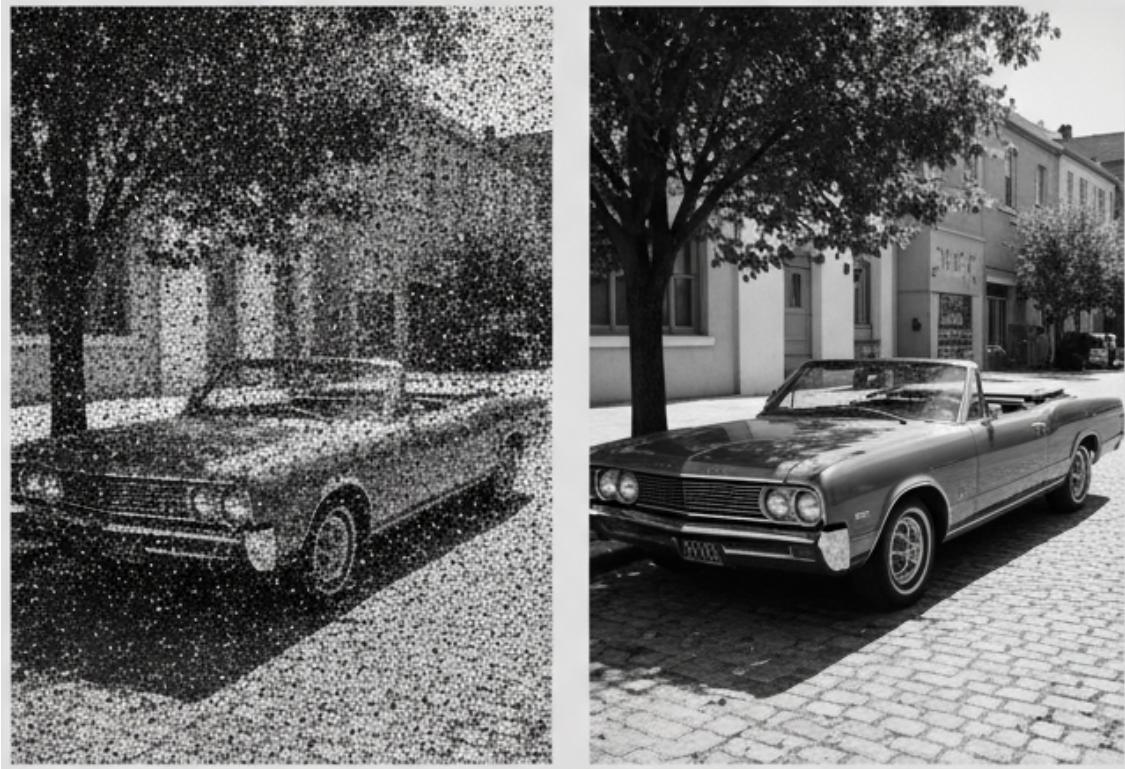


Figura 17: Rimozione del rumore sale e pepe con un filtro mediano (sinistra originale e destra modificata)

2.2: Filtri di nitidezza

I filtri di shapening preservano i dettagli, andando ad evidenziare i bordi. Per fare ciò, è necessario trovare quella operazione che permetta di distinguere quali sono i pixel dell'immagine che sono uguali oppure molto simili. Tale operazioni sono:

- la derivata prima, che esegue la differenza tra il pixel successivo ed il pixel corrente;

$$\frac{\delta f}{\delta x} = f(x+1) - f(x)$$

- la derivata seconda, che esegue la somma tra il pixel successivo ed il pixel precedente e infine sottrae due volte il pixel corrente.

$$\frac{\delta^2 f}{\delta x^2} = f(x+1) + f(x-1) - 2f(x)$$

Per tali scopi, risulta molto conveniente usare la derivata seconda, poiché presenta una risposta più forte per i dettagli (facilmente intuibile) ed ha un'implementazione decisamente più semplice (poco intuibile).

2.2.1: Il Laplaciano

Il filtro di nitidezza più usato è il Laplaciano, che sfrutta il gradiente secondo dell'immagine: ciò è decisamente logico, dato che in più di una dimensione e con le derivate parziale, si fa riferimento ad ogni derivata parziale, perciò è necessario il gradiente.

$$\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

Con dei calcoli matematici si ricava la formula seguente:

$$\frac{\delta^2 f}{\delta x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\delta^2 f}{\delta y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

da cui si ricava la maschera, ricavando i coefficienti del gradiente.

$$\nabla^2 f = \begin{bmatrix} f(x-1, y-1) & f(x-1, y) & f(x-1, y+1) \\ f(x, y-1) & f(x, y) & f(x, y+1) \\ f(x+1, y-1) & f(x+1, y) & f(x+1, y+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Tuttavia, tale filtro trova solamente i bordi dell'immagine. Per questo motivo, l'immagine di output è solamente uno dei fattori della somma algebrica con l'immagine originale.

$$g(x, y) = f(x, y) + c\nabla^2 [(x, y)]$$

In particolare, $c = 1$ se il coefficiente centrale è positivo; $c = -1$ altrimenti. A questo punto, la nuova immagine è la seguente.

$$g(x, y) = f(x, y) - f(x+1, y) - f(x-1, y) - f(x, y+1) - f(x, y-1) + 4f(x, y)$$

$$g(x, y) = 5f(x, y) - f(x+1, y) - f(x-1, y) - f(x, y+1) - f(x, y-1)$$

Perciò, la maschera è la seguente.

$$h(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Infatti, ora si comprende il motivo per cui la sua semplicità di implementazione. Inoltre, il Laplaciano è presenta i seguenti vantaggi: isotopico: perciò è indipendente dal punto in cui viene applicato, poiché applica gli array circolare, quindi la rotazione resta invariata.

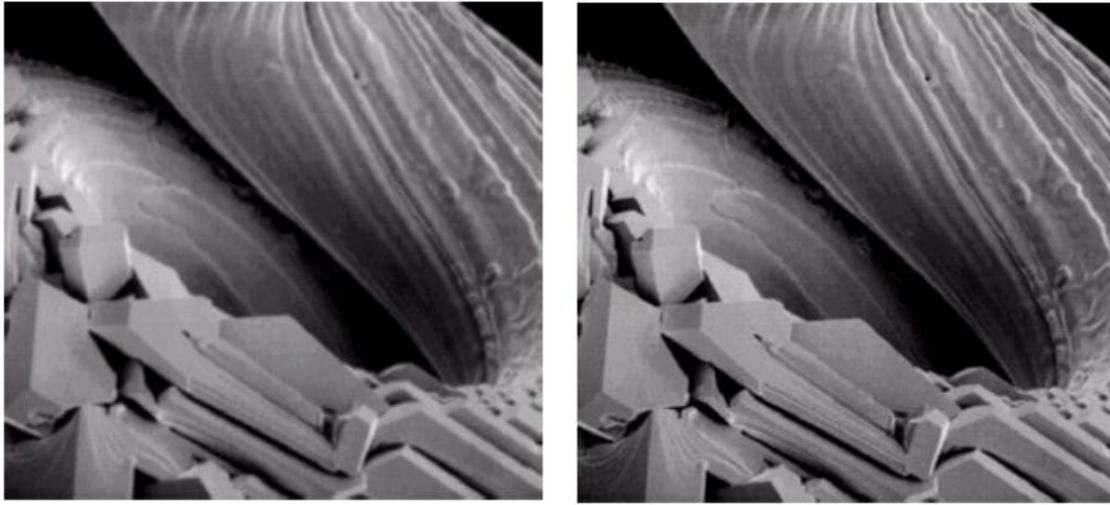


Figura 18: Applicazione del Laplaciano (sinistra originale e destra modificata)

2.2.2: Filtri di contrasto

Nel caso serve capire una scala di dettagli da evidenziare all'immagine, piuttosto che sfruttare un'operazione fissa, conviene usare i filtri di contrasto (in inglese unsharp mask). Ciò consiste in:

1. si applica un filtro Gaussiano;

$$\bar{f}(x, y) = f(x, y) * \exp \left[\frac{-(x^2 + y^2)}{2\sigma^2} \right]$$

2. si sottrae l'immagine originale con l'immagine filtrata con il filtro gaussiano, ottenendo la maschera dell'immagine;

$$g_{mask}(x, y) = f(x, y) - \bar{f}(x, y)$$

3. si aggiunge all'immagine originale la maschera dell'immagine moltiplicata per un fattore di scala di dettagli k .

$$g(x, y) = f(x, y) + kg_{mask}(x, y)$$

Ciò si dimostra nel modo seguente.

$$g(x, y) = f(x, y) + kg_{mask}(x, y) = f(x, y) + k[f(x, y) - \bar{f}(x, y)] = (1 + k)f(x, y) - \bar{f}(x, y)$$

Siccome l'immagine mascherata è un'ottima approssimazione del Laplaciano ed il filtro guassiano si può approssimare con un filtro medio 3x3, si approssima $c \approx 1+k$, che prende il nome di fattore di amplificazione.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & c & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Ciò prende il nome di high-boost filtering.

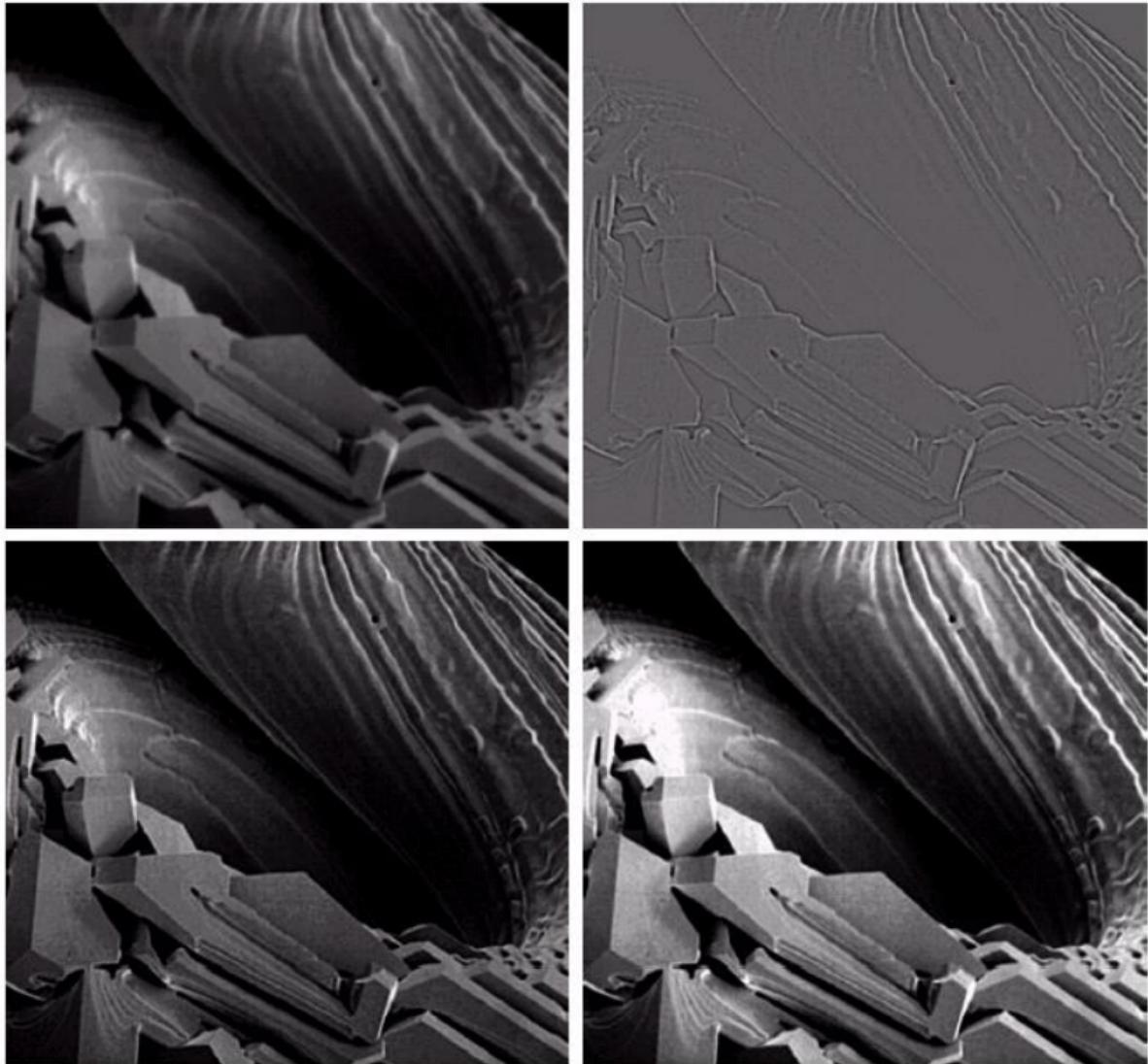


Figura 19: Originale, Laplaciano filtro di contrasto e high-boost filtering da in alto a sinistra in senso orario

2.2.3: Filtri di rilevamento dei bordi

Per rilevare i bordi, sfruttare la derivata seconda è molto semplice da implementare. Tuttavia, presenta un problema da non sottovalutare: è molto sensibile al rumore. Per questo motivo, se l'obiettivo è quello di rilevare i bordi, conviene invece usare la derivata prima. Inoltre, in matematica, la derivata prima rileva i picchi: per questo motivo, è l'operazione che permette di rilevare i bordi. Tuttavia, va eseguito sia per le righe che per le colonne.

Il primo nucleo prende il nome di operatori di Roberts, in cui si approssima attraverso l'equivalenza digitale della derivata del primo ordine, prendendo in considerazione la differenza tra il pixel successivo e il pixel precedente lungo la riga e lungo la colonna.

$$h_x(x, y) \approx f(x + 1, y) - f(x - 1, y)$$

$$h_y(x, y) \approx f(x, y + 1) - f(x, y - 1)$$

Ciò si può rappresentare anche con due matrici 2x2.

$$h_x = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$h_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Di seguito, è riportato un esempio con l'immagine 3x3 seguente.

$$f(x, y) = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

Adesso, si applicano le due convoluzioni: viene mostrato solo un prodotto per riga e per colonna.

$$h_x(1, 1) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 10 & 20 \\ 40 & 50 \end{bmatrix} = 0 \times 10 + (-1) \times 20 + 1 \times 40 + 0 \times 50 = 20$$

$$h_y(1, 1) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 10 & 20 \\ 40 & 50 \end{bmatrix} = -1 \times 10 + 0 \times 20 + 0 \times 40 + 1 \times 50 = 40$$

Tuttavia, si preferisce usare la coppia di kernel di Prewitt e di Sobel, riportati di seguito rispettivamente.

Prewitt

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Sobel

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Tali nuclei condividono le seguenti due proprietà:

- hanno i coefficienti opposti per avere un'alta risposta nella regione dell'immagine con molta variazione d'intensità, in cui la probabilità che ci sia un bordo è molto alta;
- la somma dei coefficienti è pari a 0: ciò significa che quando è applicato ad un'immagine con una regione completamente omogenea, il risultato è 0.

Una volta fatto ciò, si calcolano:

- la magnitudine, che misura quanto è forte il bordo;

$$h = \sqrt{h_x^2 + h_y^2}$$

- la direzione, che misura l'orientamento del bordo.

$$\theta = \text{atan2}(h_y, h_x) = \begin{cases} \arctan\left(\frac{h_y}{h_x}\right) & \text{se } h_x > 0 \\ \arctan\left(\frac{h_y}{h_x}\right) + \pi & \text{se } h_x < 0 \wedge h_y \geq 0 \\ \arctan\left(\frac{h_y}{h_x}\right) - \pi & \text{se } h_x < 0 \wedge h_y < 0 \\ \frac{\pi}{2} & \text{se } h_x = 0 \wedge h_y > 0 \\ -\frac{\pi}{2} & \text{se } h_x = 0 \wedge h_y < 0 \\ 0 & \text{se } h_x = 0 \wedge h_y = 0 \end{cases}$$

Combinando questi due risultati, è possibile comprendere bene la forza effettiva del bordo.

A seconda del kernel utilizzato, è possibile applicare dei filtri di rilevamento dei bordi. In questo corso ne vengono analizzati di due tipologie.

Il primo, che è anche quello più potente disponibile ad oggi, è il rilevatore di bordi di Canny. L'algoritmo consiste nei seguenti passaggi:

1. si applica un filtro gaussiano per ridurre il rumore;

$$\bar{f}(x, y) = f(x, y) * \exp \left[\frac{-(x^2 + y^2)}{2\sigma^2} \right]$$

2. si calcolano i gradienti locali con la coppia di kernel di Sobel;

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$h = \sqrt{h_x^2 + h_y^2}$$

$$\theta = \text{atan2}(h_y, h_x)$$

3. avviene la cosiddetta soppressione dei non-massimi, in cui si guarda la direzione e si confronta la magnitudine calcolata con i suoi vicini, che viene mantenuta solamente se è un massimo locale;
4. si effettua un'ulteriore processo decisionale a soglia, per classificare un pixel come bordo forte ($h(p) > T_{alta}$), bordo debole ($h(p) < T_{bassa}$) oppure un non bordo ($T_{bassa} \leq h(p) \leq T_{alta}$);
5. si prendono solo i bordi forti oppure i bordi deboli adiacenti (anche in diagonale) con un bordo forte, il resto dei bordi deboli vengono scartati.

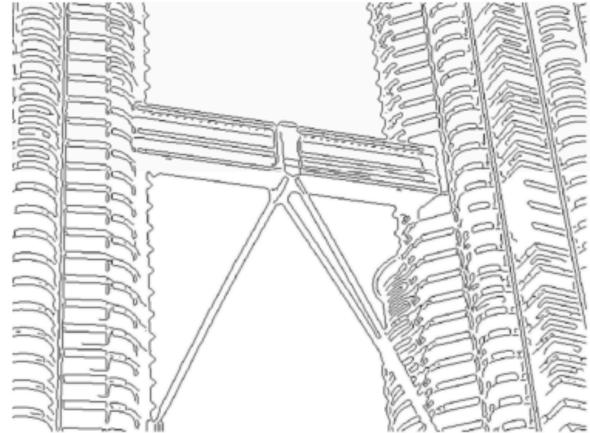


Figura 20: Immagine originale (a sinistra) e immagine con i bordi rilevati con il rilevatore di Canny (a destra)

Infine, la seconda ed ultima tipologia di rilevamento dei bordi è la trasformata di Hough, che sfrutta un metodo matematico per rilevare linee in un'immagine. Ciò, può essere usata per rilevare bordi potenzialmente sparsi, rotti od isolati; in linee utili, corrispondenti ai bordi dell'immagine. Tale processo consiste nel prendere un punto di coordinata (x, y) e memorizzare tutti i punti appartenenti alla retta $y = ax + b$. Ad esempio data il punto $(x, y) = (1, 3)$ si ottengono i punti alle coordinate $3 = a + b$, perciò $a = 3 - b$, come $(x, y) = (0, 3)$. Ripetendo ciò per gli altri punti, si ottengono una serie di punti che intersecano varie linee: i punti che hanno più intersezioni corrispondono alle linee più lunghe dell'immagine.

Tuttavia, le linee verticali hanno gradiente infinito: per risolvere ciò si usano le coordinate normali.

$$\rho = x \cos \theta + y \sin \theta$$

A questo punto, le linee più rilevanti dell'immagine, corrispondono ai valori più elevati di (ρ, θ) .

Capitolo 3:

Filtri nel dominio della frequenza

Fino ad ora, sono state trattate le immagini nel dominio spaziale, considerandole come funzioni a due variabili $f(x, y)$. Questa rappresentazione è sicuramente molto comoda ed intuitiva, poiché nelle immagini in bianco e nero, non è altro che una matrice $M \times N$ dove ogni elemento corrisponde al livello di grigio di un pixel. Tuttavia, tale rappresentazione diventa decisamente meno comoda nel momento in cui si eseguono delle operazioni di filtraggio, perché l'operazione di convoluzione risulta decisamente costosa in termini computazionali.

3.1: Dominio della frequenza

Per risolvere tale problema, si passa nel dominio della frequenza. L'idea di base, è:

- una funzione periodica si può riscrivere in serie di Fourier come somma di seni e coseni moltiplicati per coefficienti distinti;

$$f(t) = \sum_{n=-\infty}^{\infty} c_n \exp\left(j2\pi \frac{nt}{T}\right) : f(t) = f(t - T)$$

- una funzione non periodica si può rappresentare in frequenza come l'integrale di seni e coseni moltiplicati per una funzione pesata.

$$F(\nu) = \int_{-\infty}^{\infty} f(t) \exp(-j2\pi t\nu) dt$$

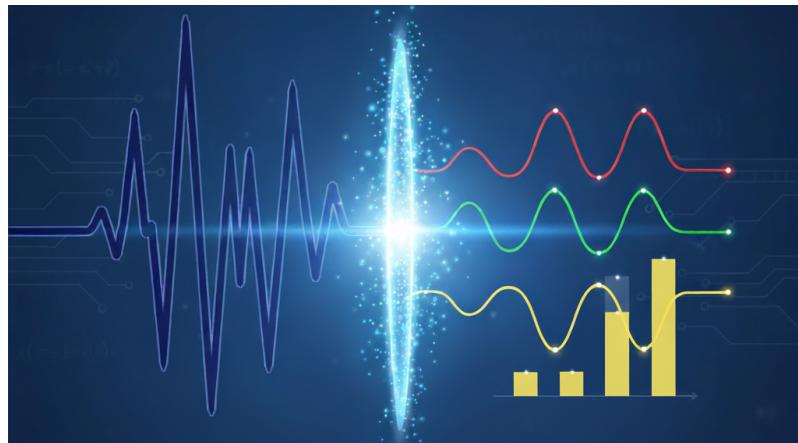


Figura 21: Rappresentazione della trasformata di Fourier

3.1.1: Trasformata di Fourier in due dimensioni

Passando in un due dimensioni, le frequenze sono due:

- frequenza spaziale lungo l'asse x , definita come u ;
- frequenza spaziale lungo l'asse y , definita come v .

Come si può intuire facilmente, se la funzione pesata in una dimensione ha due variabili, in due dimensione ne servono ben quattro: infatti x e y saranno gli indici di sommatoria, stessa cosa per l'antitrasformata che saranno u e v .

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

Il valore della trasformata di Fourier alla frequenza di origine prende il nome di componente DC: esso si calcola facendo la media tra il prodotto dell'intensità di $f(x, y)$ e un fattore MN . Infatti, risulta molto comune eseguire una traslazione della componente DC sul punto $(M/2, N/2)$.

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) + (1)^{x+y}] \exp \left[-j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

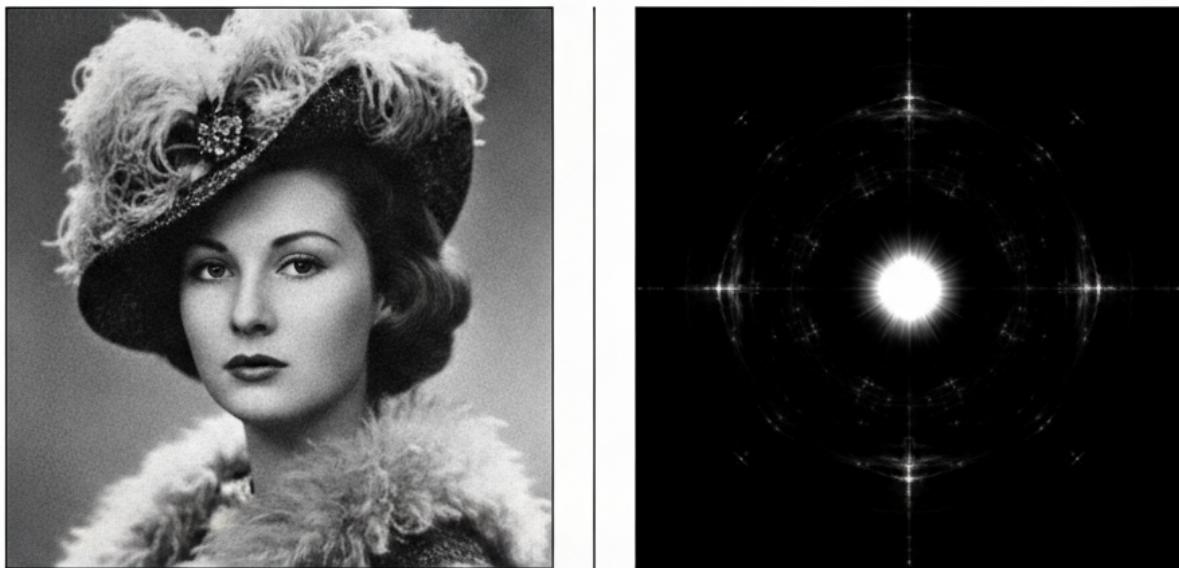


Figura 22: Immagine nel dominio spaziale (a sinistra) ed in frequenza (a destra)

3.1.2: Proprietà della trasformata di Fourier

La trasformata di Fourier gode delle seguenti proprietà:

- proprietà di traslazione, che afferma che se l'immagine è traslata nello spazio, in frequenza sarà traslata solo nella fase ma non in ampiezza;

$$F[f(x - x_0, y - y_0)] = F(u, v) \exp \left[-j2\pi \left(\frac{ux_0}{M} + \frac{vy_0}{N} \right) \right]$$

- linearità;

$$F[af_1(x, y) + bf_2(x, y)] = aF_1(u, v) + bF_2(u, v)$$

- proprietà di rotazione, che afferma che se l'immagine è ruotata di un angolo θ , anche in frequenza sarà ruotata dello stesso angolo;

- separabilità, ossia si può eseguire prima la trasformata sulle righe e poi sulle colonne (o viceversa).

3.1.3: Teorema della convoluzione e applicazione al filtraggio

La potenzialità del passaggio dal dominio spaziale al dominio della frequenza sta nel teorema della convoluzione. Tale teorema afferma che eseguire una convoluzione nel dominio spaziale corrisponde ad eseguire un prodotto nel dominio della frequenza, con un costo computazione molto meno elevato.

$$f(x, y) * h(x, y) \longleftrightarrow F(u, v)H(u, v)$$

Ciò consiste nel trasformare l'immagine originale in frequenza; moltiplicarla per il filtro in frequenza, denominato anche funzione di trasferimento; e trasformare l'immagine filtrata nel dominio spaziale. In particolare, si fa riferimento a due tipologie di frequenza:

- le basse frequenze, che rappresentano le aree omogenee;
- le alte frequenze, che rappresentano i dettagli.

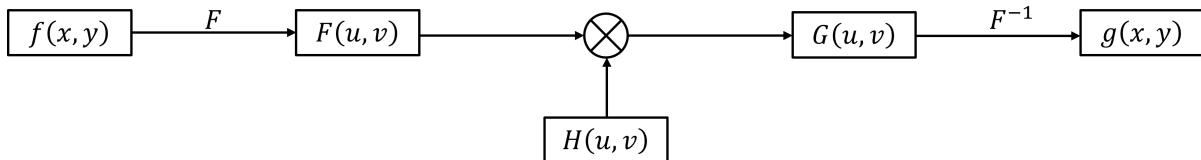


Figura 23: Filtraggio nel dominio della frequenza

3.2: Filtro di Notch

Il filtro più semplice da implementare è il filtro di Notch, chiamato anche filtro elimina-banda, che consiste nel far passare tutte le frequenze, eccetto per una tacca (notch in inglese), che corrisponde alla componente DC.

$$H(u, v) = \begin{cases} 0 & \text{se } (u, v) = \left(\frac{M}{2}, \frac{N}{2}\right) \\ 1 & \text{se } (u, v) \neq \left(\frac{M}{2}, \frac{N}{2}\right) \end{cases}$$

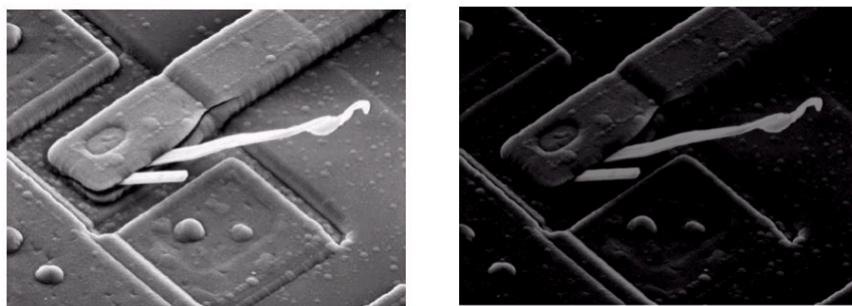


Figura 24: Immagine originale (a sinistra) e immagine filtrata con un filtro Notch (a destra)

3.3: Filtri passa-basso

I filtri passa-basso (LPF, dall'inglese low-pass filter) attenuano le alte frequenze, applicando quindi l'effetto di sfocatura e riduzione del rumore. Infatti, corrispondono ai filtri di smussamento nel dominio spaziale. Innanzitutto, si considera la distanza euclidea in frequenza $D(u, v)$, calcolata come segue.

$$D(u, v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$$

Inoltre, si considera la cosiddetta frequenza di cutoff (D_0), che cambia il comportamento del filtro in base a tale valore.

3.3.1: Filtro passa-basso ideale

Il filtro passa-basso ideale fa passare solamente le frequenze la cui distanza euclidea è minore od al più uguale alla frequenza di cutoff.

$$H(u, v) = \begin{cases} 1 & \text{se } D(u, v) \leq D_0 \\ 0 & \text{se } D(u, v) > D_0 \end{cases}$$

In particolare, maggiore sarà il cutoff e minore sarà la sfocatura, poiché passeranno più frequenze.

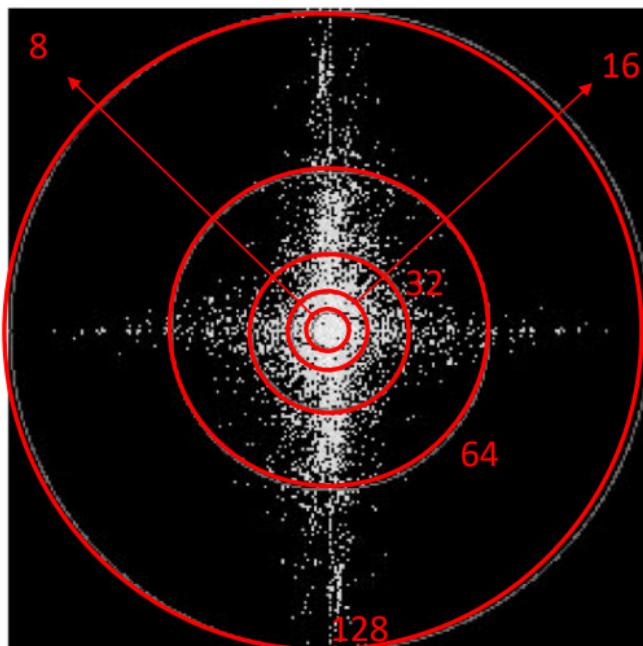


Figura 25: Funzionamento di un filtro passo-basso ideale

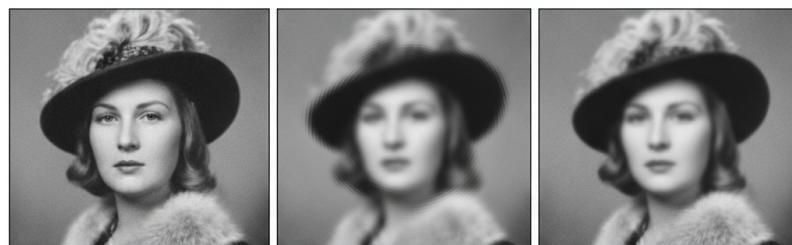


Figura 26: Da sinistra: immagine originale, LPF ideale $D_0 = 8$ e LPF ideale $D_0 = 16$

3.3.2: Filtro passa-basso di Butterworth

Il filtro passa-basso di Butterworth permette di eseguire una sfocatura molto meno intensa rispetto a quella ideale, ma neanche ai livello del filtro Gaussiano. Oltre alla distanza euclidea ed alla frequenza di cutoff, si aggiunge un parametro n , che indica l'ordine del filtro, ovvero la sua ripidità: infatti, maggiore è l'ordine, più si avvicina ad un filtro passa basso ideale.

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}}$$

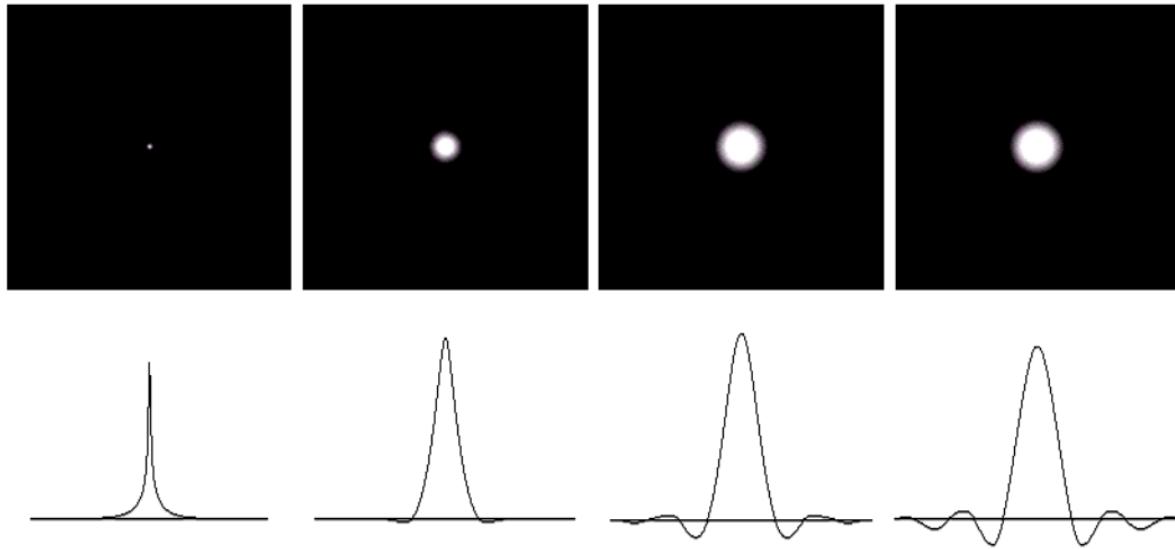


Figura 27: Vari ordini del filtro passa-basso di Butterworth

3.3.3: Filtro passa-basso Gaussiano

Il filtro passa-basso Guassiano è in grado di attenuare le alte frequenze con poca intensità. In particolare, il cutoff corrisponde alla deviazione standard della gaussiana.

$$H(u, v) = \exp \left[-\frac{D^2(u, v)}{2D_0^2} \right]$$

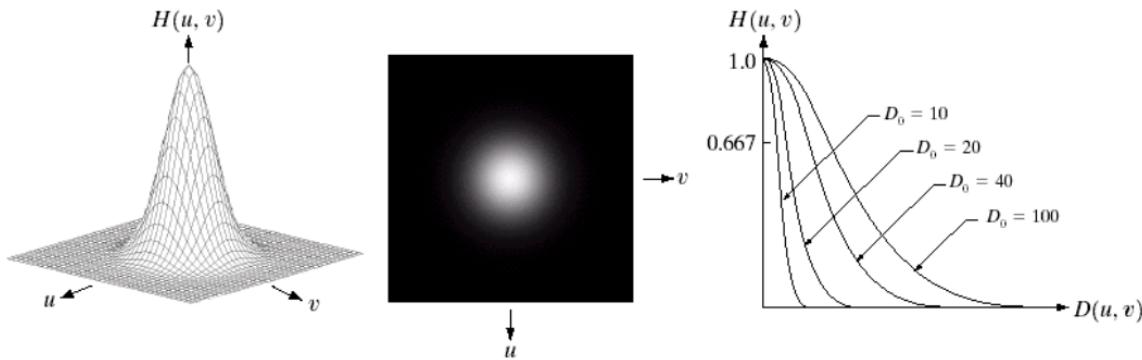


Figura 28: Filtri passa-basso Gaussiani

3.4: Filtri passa-alto

I filtri passa-alto (HPF, dall'inglese high-pass filter) attenuano le basse frequenze e risaltano le alte frequenze, applicando quindi l'effetto di sfocatura e riduzione del rumore. Infatti, corrispondono ai filtri di nitidezza nel dominio spaziale. Anche in questo caso, si ha la distanza euclidea e la frequenza di cutoff. Inoltre, la funzione di trasferimento di un filtro passo-alto è l'inverso di quello del filtro passa-basso.

$$H_{HPF}(u, v) = 1 - H_{LPF}(u, v)$$

3.4.1: Filtro passa-alto ideale, di Butterworth e Gaussiano

Di seguito, sono riportate le formule dei filtri passa-alto ideale, di Butterworth e Gaussiano, che presentano le proprietà inverse di quelle del passo-basso.

$$H(u, v) = \begin{cases} 0 & \text{se } D(u, v) \leq D_0 \\ 1 & \text{se } D(u, v) > D_0 \end{cases} \quad H(u, v) = \frac{1}{1 + \left[\frac{D_0}{D(u, v)} \right]^{2n}} \quad H(u, v) = 1 - \exp \left[-\frac{D^2(u, v)}{2D_0^2} \right]$$



Figura 29: Da alto a sinistra senso orario: originale, HPF ideale, HPF di Butterworth e HPF Gaussiano

3.4.2: Il Laplaciano, i filtri di contrasto e l'high-boost filtering

Nel dominio spaziale, il Laplaciano presenta la formula seguente:

$$g(x, y) = f(x, y) - \nabla^2 f(x, y)$$

dove:

$$\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

In frequenza, la trasformata di una Fourier della derivata è la seguente:

$$F \left[\frac{d^2 f}{dt^2} \right] = -\nu^2 F(\nu)$$

che in due dimensioni e per le immagini, corrisponde:

$$F [\nabla^2 f] = -(u^2 + v^2) F(u, v)$$

Perciò, l'immagine trasformata in frequenza si ottiene nella maniera che segue:

$$G(u, v) = F(u, v) + (u^2 + v^2) F(u, v) = [1 + (u^2 + v^2)] F(u, v)$$

da cui si ricava la funzione di trasferimento.

$$H(u, v) = 1 + (u^2 + v^2)$$

Infine, per quanto riguarda i filtri di contrasto e l'high-boost filtering, le formule sono rispettivamente le seguenti.

$$H(u, v) = 1 - H_{lp}(u, v)$$

$$H(u, v) = (A - 1) + H_{lp}(u, v)$$

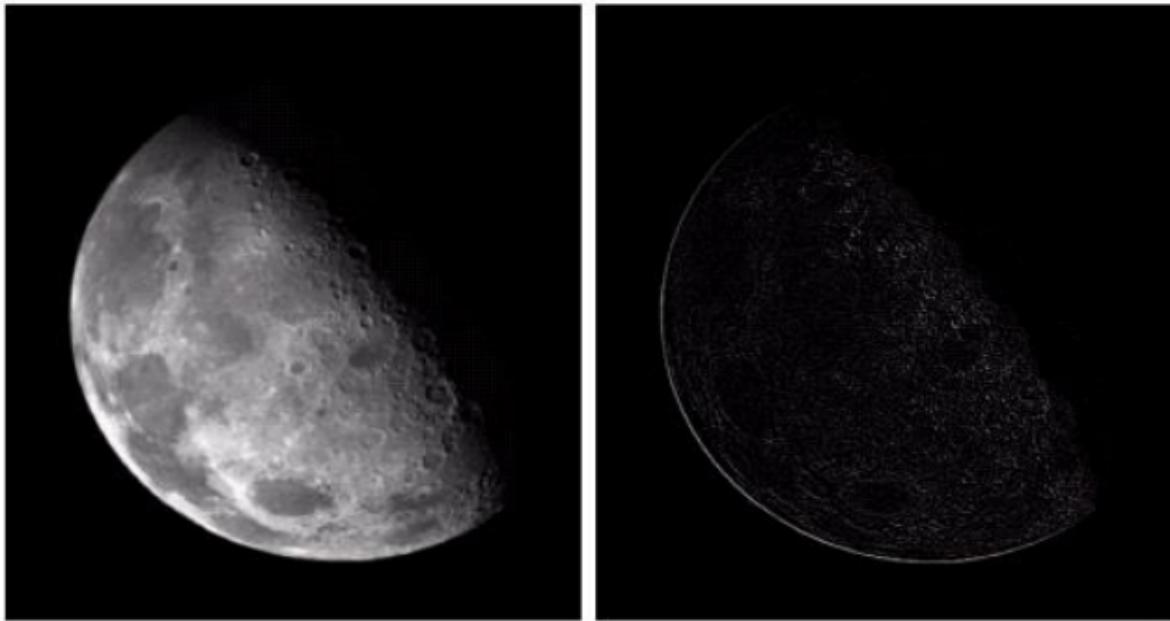


Figura 30: Immagine originale (a sinistra) e maschera del filtro Laplaciano (a destra)

Capitolo 4:

Trasformata Wavelet

In questo capitolo verranno mostrati i limiti della trasformata di Fourier, per poi introdurre la trasformata Wavelet, le sue potenzialità e le possibili applicazioni.

4.1: Limiti della trasformata di Fourier e possibile soluzione

Per comprendere bene i limiti della trasformata di Fourier, è necessario analizzare bene le caratteristiche di un'immagine.

4.1.1: Caratteristiche di un'immagine

Un'immagine si può vedere come un'insieme di regioni di texture e livelli di grigio simili, che combinati tra loro formano oggetti. In particolare, si considera:

- l'analisi a bassa risoluzione, che analizza gli oggetti ad alto contrasto, quindi le basse frequenze;
- l'analisi ad alta risoluzione, che analizza gli oggetti a basso contrasto, perciò le alte frequenze;
- l'analisi multirisoluzione, che analizza gli oggetti a contrasto variabile.

Da qui, si nota che l'immagine è un segnale non stazionario: per prima conviene analizzare la figura seguente. Come si può osservare:

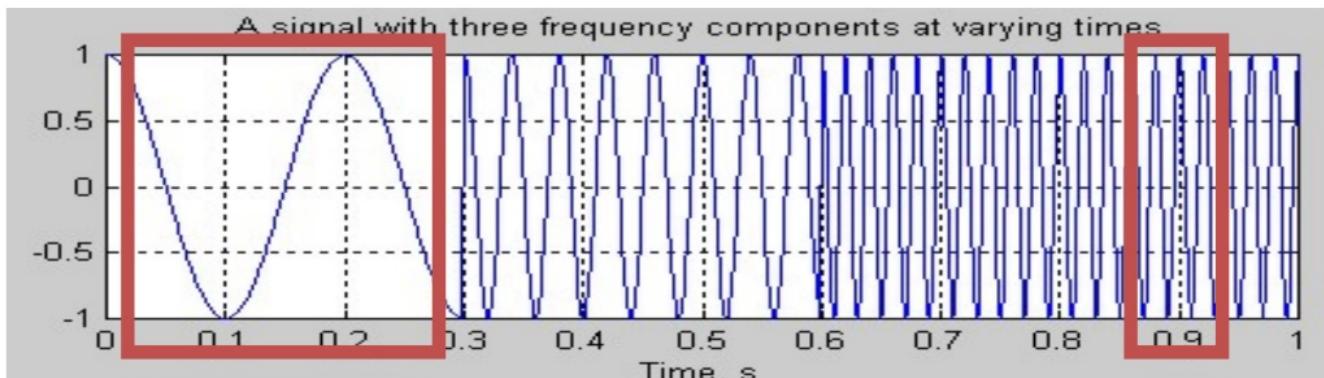


Figura 31: Variazione delle frequenze nell'asse temporale da $t_0 = 0\text{ s}$ a $t_3 = 1\text{ s}$

- da $t_0 = 0\text{ s}$ a $t_1 = 0,3\text{ s}$ si hanno le basse frequenze, in cui il segnale oscilla molto lentamente;
- da $t_1 = 0,3\text{ s}$ a $t_2 = 0,85\text{ s}$ si hanno le medie frequenze, dove le oscillazioni diventano sempre più ravvicinate e veloci;
- da $t_2 = 0,85\text{ s}$ a $t_3 = 1\text{ s}$ si hanno le alte frequenze, dove il segnale oscilla molto velocemente.

Perciò, le immagini hanno le statistiche che variano localmente: per questo motivo costruire un modello statistico che copre l'intera immagine, risulta decisamente molto complesso.

4.1.2: Problemi della trasformata di Fourier

Tornando alla figura precedente, se si eseguisse la trasformata di Fourier di quel segnale, essa darebbe come unica informazione la presenza di una bassa, media ed alta frequenza, senza fornire la loro posizione nel tempo. Generalizzando, eseguendo la trasformata di Fourier su un'immagine, si perdono completamente le informazioni temporali e spaziali.

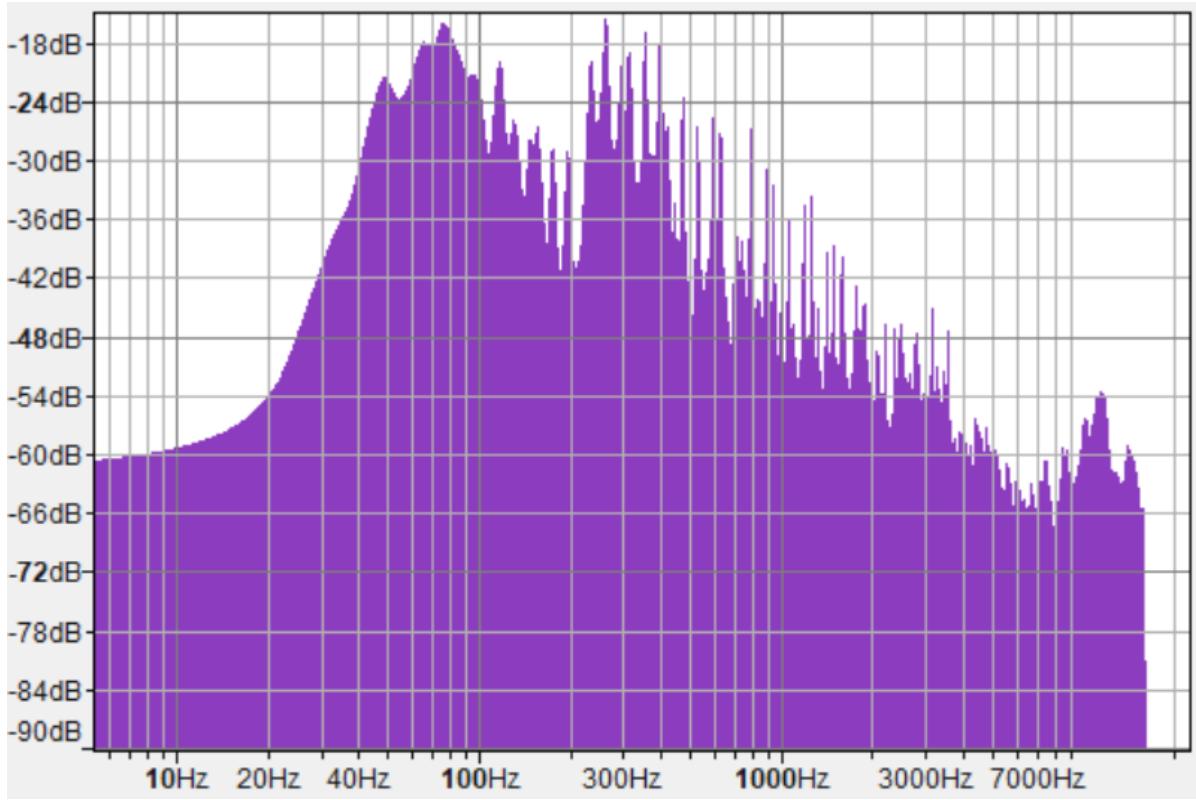


Figura 32: Rappresentazione dei problemi della trasformata di Fourier

4.1.3: Una soluzione temporanea: la trasformata di Fourier a breve termine

La trasformata di Fourier a breve termine (STFT, dall'inglese Short Time Fourier Transform) è una tecnica fondamentale per analizzare come il contenuto in frequenza di un segnale cambia nel tempo. Ciò consiste in:

1. si definisce una finestra di tempo di una dimensione fissa;
2. si applica la trasformata di Fourier al segnale finestrato;
3. si memorizzano le frequenze trovate in quel intervallo di tempo;
4. si scorre la finestra leggermente avanti nel tempo e si ripete le analisi, finché non finisce il segnale, costruendo una mappa.

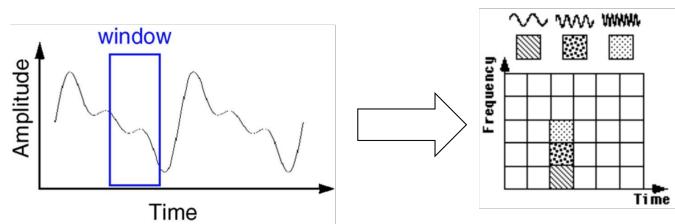


Figura 33: Esecuzione della STFT

Come si può facilmente intuire, la scelta della finestra risulta cruciale, poiché più la finestra è piccola, maggiore sarà la risoluzione temporale. Tuttavia, se si prende una finestra troppo piccola, si rischia che le frequenze rappresentabili siano insufficienti, tanto sarà debole il potenziale di discriminazione delle frequenze.

4.2: Introduzione alla trasformata Wavelet

La trasformata Wavelet si basa su piccole onde, dette appunto wavelet, che godono delle seguenti tre proprietà:

- frequenza ed ampiezza variabile;
- durata limitata, perciò non devono essere sinusoidi;
- valor medio nullo.

Di seguito, è riportata una tabella che mostra le wavelet più utilizzate.

Nome	Formula analitica	Grafico
Shannon	$\psi(t) = 2 \operatorname{sinc}(2t) - \operatorname{sinc}(t)$	
Morlet	$\psi(t) = \exp\left(-\frac{t^2}{2}\right) \cos(5t)$	
Haar	$\psi(t) = \operatorname{rect}_{1/2}\left(t - \frac{1}{4}\right) - \operatorname{rect}_{1/2}\left(t - \frac{3}{4}\right)$	

Figura 34: Tipologie wavelet fondamentali

4.2.1: Determinazione della trasformata Wavelet

A questo punto, ogni segnale (non necessariamente periodico), può essere scritto in serie di wavelet.

$$f(t) = \sum_i a_i \psi_i(t)$$

$\psi_{s,\tau}$ è così definito:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right)$$

dove:

- s è la scalatura;
- τ è la traslazione nel tempo;
- $\frac{1}{\sqrt{s}}$ è la normalizzazione;
- ψ è la wavelet madre;
- $\psi_{s,\tau}$ è la wavelet scalata e traslata nel tempo.

Adesso, si possono ricavare i coefficienti della trasformata Wavelet.

$$\gamma(s, \tau) = \int_{-\infty}^{\infty} f(t) \psi_{s,\tau}^*(t) dt$$

Una volta eseguito ciò, è possibile ricostruire il segnale con la formula seguente.

$$f(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \gamma(s, \tau) \psi_{s,\tau}(t) d\tau ds$$

Nella figura che segue, viene mostrata una rappresentazione di una possibile trasformata Wavelet, partendo da un segnale $f(t)$.

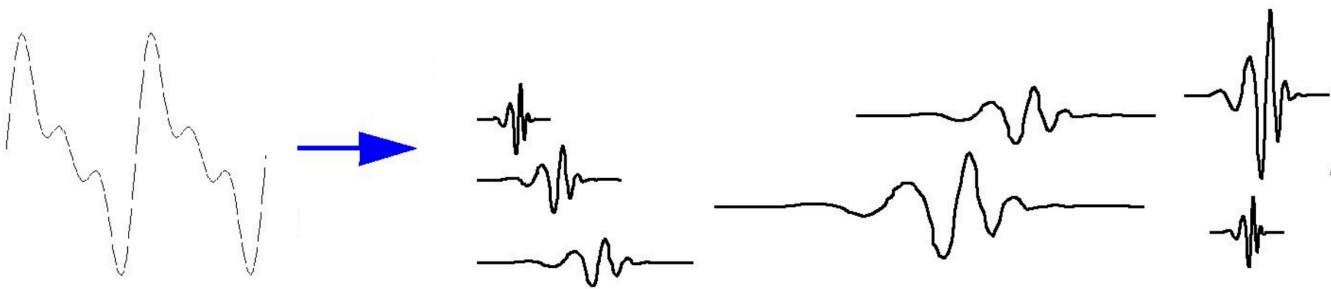


Figura 35: Rappresentazione di una trasformata Wavelet

4.2.2: Proprietà

La trasformata Wavelet gode delle seguenti proprietà:

- localizzazione simultanea spaziale e temporale, in cui la localizzazione della wavelet permette esplicitamente di rappresentare gli eventi nel tempo e la forma delle wavelet permettono di rappresentare i dettagli differenti e risoluzione differenti;
- sparsità, in cui le funzioni usate nella pratica presentano coefficienti pari a zero o molto piccoli;
- adattabilità, poiché si possono rappresentare funzioni discontinue e con angoli in modo molto efficiente;
- complessità temporale lineare, cioè diverse trasformazioni si possono compiere in tempo $O(N)$.

4.3: Applicazione della trasformata Wavelet: l'analisi multirisoluzione

Per effettuare l'analisi multirisoluzione di un'immagine, è necessario definire in maniera compatta la formula dell'espansione di un segnale $f(t)$ in serie.

$$f(t) = \sum_j \sum_i a_{ij} \psi_{ij}(t) = \sum_j \sum_i a_{ij} 2^{\frac{j}{2}} \psi(2^j t - j)$$

In particolare:

- a_{ij} sono coefficienti reali di espansione;
- $\psi_{ij}(t)$ sono le funzioni di espansione.

In particolare, si crea un mapping dove le j sono le ascisse e i le ordinate: le $f_i(t)$ più in basso rappresentano i dettagli grossolani; fino ad arrivare più vicino a $f(t)$, che rappresentano i dettagli fini, come mostrato nell'immagine di seguito.

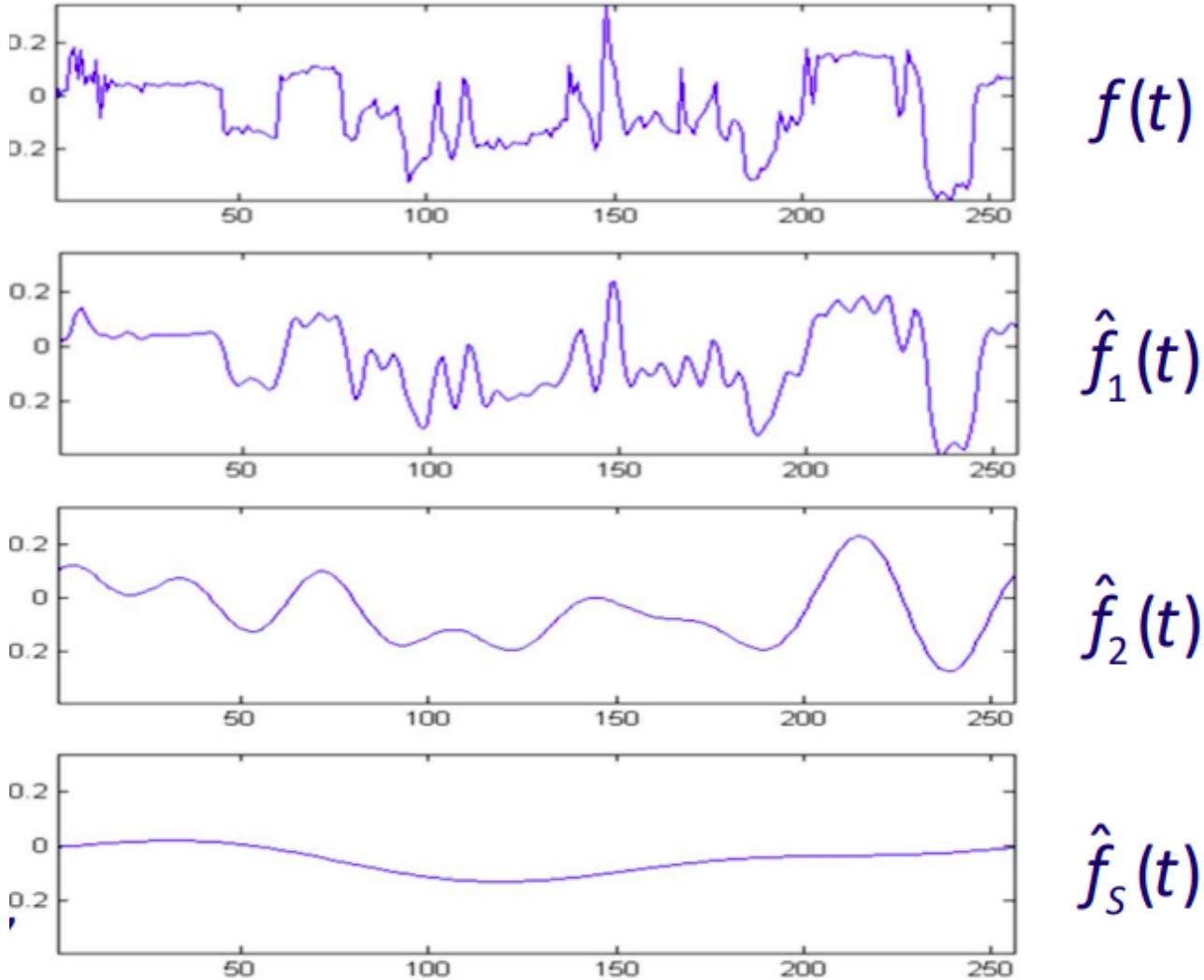


Figura 36: Analisi dei dettagli delle serie

4.3.1: Creazione delle piramide dell'immagine