

---

# APPUNTI DI ELABORAZIONE DELLE IMMAGINI DIGITALI

---

Supporto allo studio



# Sommario

1. Elaborazione delle immagini .....	5
1.1. Cos'è un'immagine .....	5
1.2. Acquisizione delle immagini .....	5
1.3. Campionamento di Immagini .....	6
1.4 Rappresentazione delle immagini digitali.....	7
1.5 Risoluzione di un'immagine.....	8
1.5.1 Risoluzione spaziale (numero di campioni per unità di aria) .....	8
1.5.2 Risoluzione radiometrica (numeri di livelli di quantizzazione).....	8
1.5.3 Risoluzione spettrale (larghezza delle bande sulle quali è acquisito il segnale).....	9
1.5.4 Risoluzione temporale .....	9
1.6. Relazione di base tra i pixel .....	9
1.6.1 Adiacenza .....	9
1.6.2 Connattività .....	10
1.7. Misure di distanza .....	10
1.8. Operazioni tra i pixel.....	11
1.8.1 Operazioni aritmetiche .....	11
1.8.2 Operazioni matriciali e puntuali.....	11
1.8.3 Operazioni logiche .....	12
1.9 Roi (region of interest) .....	12
1.8.4 Operazioni spaziali.....	13
2. Tecniche puntuali per il miglioramento della qualità .....	14
2.1 Funzioni per la trasformazione di intensità .....	14
2.2 Iistogrammi .....	16
2.2.1 Calcolo di un Iistogramma .....	16
2.2.2 Equalizzazione di istogrammi.....	18
2.2.3 Specifica di un istogramma .....	20
3. Tecniche di analisi nel dominio spaziale .....	21
3.1 Filtri di smoothing.....	21
3.1.1 Filtri di media.....	21
3.1.2 Filtri di smoothing non lineari basati sulle statistiche d'ordine .....	22
3.2 Filtri di sharpening.....	24
3.3 Filtro Gradiente .....	25
3.3.1 Operatori di Roberts .....	26
3.3.2 Operatore Isotropico .....	27
3.3.3 Operatore Di Prewitt e Sobel.....	27

3.4 Filtro Laplaciano .....	28
3.5 Filtro high-boost [Enfasi di alta Frequenza] .....	29
4. Convoluzione, Correlazione e Campionamento .....	31
4.1 Teorema della Convoluzione .....	31
4.1.1 Teorema della Convoluzione 2-D.....	32
4.2 Teorema della Correlazione.....	33
4.3.Teorema del Campionamento .....	35
5.Le basi per il filtraggio nel dominio della frequenza .....	38
5.1 Filtraggio nel dominio della frequenza .....	38
5.2 Corrispondenza tra il filtraggio nel dominio spaziale e quello nel dominio della frequenza.....	39
5.3 Smoothing utilizzando i filtri nel dominio della frequenza .....	40
5.3.2 Filtri passa-basso di Butterworth.....	41
5.3.3 Filtri passa-basso Gaussiani .....	42
5.4 Sharpening utilizzando i filtri nel dominio della frequenza .....	43
5.4.1 Filtri passa-alto ideali.....	43
5.4.2 Filtri passa-alto di Butterworth.....	44
5.4.3 Filtri passa-alto Gaussiani .....	45
5.5 Il laplaciano nel dominio della frequenza .....	45
5.7 Filtraggio omomorfico .....	46
6.Filtraggio nel dominio della frequenza .....	47
6.1 Serie di Fourier .....	47
6.2 Trasformata e Antitrasformata di Fourier continua.....	47
6.3 Trasformata e Antitrasformata di Fourier (DFT) Discreta .....	49
6.4 La trasformata e Antitrasformata di Fourier 2D (DFT) Discreta .....	50
6.4.1 Proprietà di Separabilità: .....	51
6.4.2 Traslazione.....	51
6.4.3 Rotazione.....	52
6.4.4 Valore Medio .....	52
Periodicità .....	52
6.4.5 Simmetria .....	53
7.Segmentazione di immagini .....	54
Segmentazione orientata alla Regione .....	54
Region Growing (accrescimento di regioni).....	54
7.3.2 Split & Merge.....	56
7.2. Segmentazione orientata sulla forma.....	57
7.2.1 La trasformata di Hough .....	57

7.1 Segmentazione orientata sul contorno .....	61
7.1.1 Modelli di edge .....	61
7.1.4 Canny edge detector .....	61
7.1.5 Harris corner detection.....	63
8. Morfologia matematica .....	66
8.1 Erosione e Dilatazione .....	66
8.2 Apertura e Chiusura.....	67
8.3 Trasformata hit-or-miss.....	69
8.3.1 Umbra.....	72
8.5 Morfologia in scala di grigio.....	73
8.5.1 Dilatazione.....	73
8.5.2 Erosione.....	74
8.5.3 Apertura e Chiusura.....	75
8.6 Trasformata distanza .....	76
9. Stima del movimento .....	81
9.1 Sottrazione dello sfondo.....	81
9.2 Il flusso ottico .....	82
9.2.1 Ipotesi fondamentale .....	83
9.2.2 Calcolo del flusso ottico .....	83
9.2.3 Calcolo completo del flusso ottico:.....	84
9.3 Stime basate su corrispondenze discrete .....	85
9.4 Algoritmi.....	86
9.4.1 Algoritmo di scelta.....	87
9.4.2 Algoritmo FULL SEARCH.....	87
9.4.3 Three step search (algoritmo di ricerca approssimato) .....	88
10 Il colore.....	89
10.1 Modelli di Colore .....	90
10.1.2 Modello RGB.....	91
10.1.3 Il modello CMY .....	91
10.1.4 Il modello HSV .....	92
10.5 Segmentazione di immagini a colori .....	92
11. Clustering .....	94
11.1 Algoritmo K-means .....	96
11.2 Variante di Meng-Hee- Heng .....	97
11.2 Isodata .....	97
11.3 Il metodo di Ohlander.....	98

11.4 Tagli minimali (jinbo shi).....	99
11.4.1 Taglio normalizzato.....	99
Possibili Domande d'Esame.....	100

# 1. Elaborazione delle immagini

## 1.1. Cos'è un'immagine

Un'immagine è una funzione bidimensionale  $f(x,y)$  dove  $x,y$  sono le **coordinate spaziali** sul piano e l'ampiezza di  $f$  è l'**intensità o livello di grigio** dell'immagine in quel determinato punto. Siccome parliamo di immagini digitali avremo a che fare con valori di  $x,y$  e  $f$  **finiti**. Gli elementi che compongono un'immagine si chiamano **pixel** o picture element; sono in numero finito e ciascuno di essi ha un particolare valore e una particolare posizione.

## 1.2. Acquisizione delle immagini

Abbiamo detto che un'immagine viene definita come una funzione bidimensionale  $f(x,y)$  con valori discreti.  $f(x,y)$  deve essere finita e diversa da zero; solitamente ciò si indica con l'intervallo  $[0, L-1]$  dove 0 indica il **nero**, e  $L-1$  indica il **bianco** (ovviamente si parla di un'immagine monocromatica).

La convenzione vuole che l'**origine** di un'immagine digitale si trovi nella parte **in alto a sinistra**, con l'asse positivo delle  $x$  che si estende verso il basso, l'asse positivo delle  $y$  verso destra.

Inoltre è caratterizzata da due componenti:

- **illuminazione**: quantità di luce diretta incidente sulla scena. La sua natura è determinata dalla fonte di illuminazione.
- **riflettanza**: quantità di luce riflessa dagli oggetti presenti nella scena. La sua natura è determinata dalle caratteristiche degli oggetti.

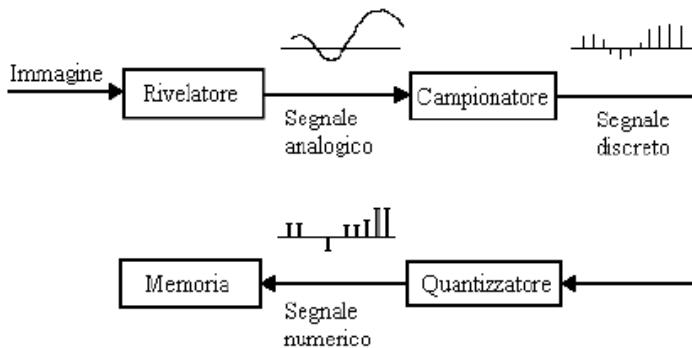
Queste due componenti danno origine alla  $f(x,y)$ :

$$\underline{f(x,y) = i(x,y) r(x,y)}$$

$$\begin{array}{ll} \text{con} & 0 < i(x,y) < \infty \\ \text{e} & 0 < r(x,y) < 1 \end{array}$$

Un'immagine viene rilevata da un **Rilevatore** che in ingegneria viene detto **Rivelatore** che rileva il valore numerico associato ad un fenomeno, in questo caso il fenomeno è l'energia irradiata da un corpo in virtù di una fonte luminosa. Il segnale Rivelato è un segnale continuo es. (radiografia) che veniva prodotto in analogico e per poterla tradurre in digitale necessitava di uno scanner particolare poiché soffriva di un problema di riflessione essendo una pellicola trasparente. Questo scanner effettuava proprio la discretizzazione di un segnale, cioè un processo propriamente detto **Campionamento** che si suddivide in due fasi: il **Campionamento** e la **Quantizzazione**.

## 1.3. Campionamento di Immagini



- **Campionamento:** operazione di **digitalizzazione delle coordinate**. Dato un segnale continuo occorre scegliere un numero finito di “campioni” rappresentativi del segnale. Il valore numerico in ogni singolo punto del segnale è un **numero reale** quindi ancora non rappresentabile dal sistema di calcolo e per questo motivo necessitiamo di un passaggio da **un numero reale** ad un **numero reale macchina**, (floating point , cioè quel numero espresso dalla mantissa e dall'esponente che può essere con segno o senza segno il quale poi limita il range di valori rappresentabili) e questa processo è detto Quantizzazione.
- **Quantizzazione:** operazione di **digitalizzazione dell'ampiezza**.  
Una volta che il segnale è stato campionato, deve essere quantizzata l'ampiezza dei valori campionati. Un quantizzatore in sostanza “mappa” un range di valori di ingresso in un unico valore di uscita. Questo deve avvenire in quanto, in un segnale analogico, il range di valori tra due estremi è infinito mentre nel mondo digitale solo un certo numero di valori potrà essere rappresentato. Esempio di tecnica di **Quantizzazione Lineare**, cioè l’obiettivo è quello di portare in un range espresso da un valore minimo e un valore massimo espresso tramite esponente e mantissa.

$$L_{\text{Min}} \leq f(x, y) \leq L_{\text{Max}}$$

$L_{\text{min}}$  è ottenuto dal prodotto dell’illuminazione minima per la riflettanza minima mentre  $L_{\text{max}}$  è il viceversa. Il modo più semplice per portare un numero reale in questo range definito **[0, L - 1]**, dove  $L$  è detto **Livello di Grigio**, è quello di inizialmente prendere il livello di grigio di partenza ( $f(x, y)$ ) sottrargli il valore minimo ( $L_{\text{min}}$ ), dividere per il **Delta** (la **dinamica**, cioè  $L_{\text{max}} - L_{\text{min}}$ ) questo fa sì che il valore risulti poi compreso tra 0 e 1, questo procedimento è detto **Standardizzazione**. A questo punto si moltiplica il risultato per  $L - 1$  (cioè il valore massimo) in modo tale da ottenere un valore compreso tra 0 e  $L - 1$ .

Il nostro sistema visivo non percepisce più di 300 livelli di grigio, il valore più prossimo in binario è **256** cioè **L = 256** che i valori vanno da 0 a 255 e per rappresentarli servono **8 bit**.

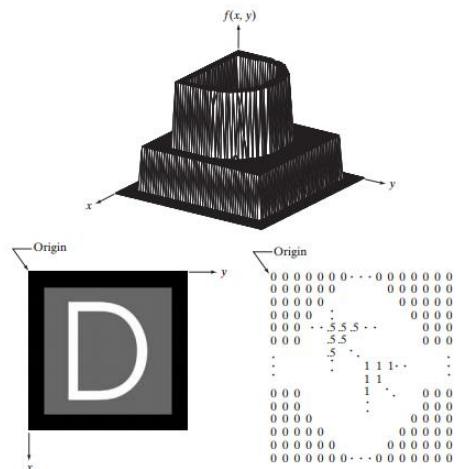
$$gl = \frac{l - L_{\text{min}}}{L_{\text{max}} - L_{\text{min}}} L$$

## 1.4 Rappresentazione delle immagini digitali

Esistono tre modi di base per rappresentare  $f(x, y)$ :

- La figura in alto è un diagramma della funzione, in cui i due assi ne determinano la posizione spaziale e un terzo asse ne indica i valori assunti da  $f$ (le intensità) come funzione di due variabili  $x$  e  $y$ .
  - La figura a sinistra è la più comune e mostra la  $f(x, y)$  come potrebbe apparire su di un monitor o su una fotografia.
  - La terza è semplicemente la rappresentazione dei valori numerici della funzione  $f(x, y)$  sotto forma matriciale.

Quest'ultima è la più usata per descrivere un'immagine digitale e i suoi elementi, esempio:



$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \dots & a_{M-1,N-1} \end{bmatrix}$$

Ogni elemento della matrice è chiamato **image element** o **pixel**. Tipicamente il numero dei valori dei livelli di grigio è dato da una potenza del 2:

$$L = 2^k$$

La matrice è costituita da  $M \times N$  pixel, ciascuno dei quali ha un valore appartenente all'intervallo [0, L-1] essendo L i livelli possibili di intensità (o di grigio). Si ha tipicamente  $L = 2^k$ , dove k è il numero di bit usato per codificare ciascun pixel (**profondità del pixel**). Per esempio, con 8 bit si ha la possibilità di rappresentare **256 livelli**. Una immagine monocromatica tipica (512 \* 512 \* 8) occupa pertanto  $\approx 2\text{Mbit} \approx 256\text{ kbyte}$  di memoria. Il numero, b, di bit necessari per memorizzare un'immagine è

$$b = M \times N \times k$$

In realtà, si è fatta l'ipotesi implicita di una **quantizzazione lineare** per la produzione degli  $L$  livelli discreti di grigio a partire dalla intensità luminosa. Sappiamo invece che la risposta del sistema visivo umano non è lineare, bensì **logaritmica**: questo mette l'occhio in condizione di discriminare un numero enorme di livelli di luminosità, ben maggiore dei 256 precedentemente ipotizzati. Pertanto una immagine digitalizzata ha normalmente una gamma dinamica molto meno ampia di quella del sistema visivo umano.

## 1.5 Risoluzione di un'immagine

I tipi di risoluzione di un'immagine sono:

- spaziale
- radiometrica
- spettrale
- temporale

### 1.5.1 Risoluzione spaziale (numero di campioni per unità di area)

La risoluzione spaziale è il più piccolo dettaglio distinguibile in una immagine, determina il numero di pixel che devono essere utilizzati per creare l'immagine e viene espressa tramite un valore, dpi, cioè punti per pollice. Se i pixel sono di grande dimensione la risoluzione spaziale è scadente, in quanto appaiono visibili le discontinuità di grigio al confine tra i pixel. Man mano che la dimensione dei pixel si riduce l'effetto diventa meno visibile, fino al punto che si ha l'impressione di un'immagine continua.

**Def.**

Per **risoluzione spaziale** s'intende in genere lo spazio minimo tra due oggetti per cui all'interno di un'immagine essi sono ancora visibili come due entità separate.

**Risoluzione Spaziale:** numero di campioni per unità di area (espressa in termini di pixel nello spazio immagine)

Variazione della risoluzione spaziale



### 1.5.2 Risoluzione radiometrica (numeri di livelli di quantizzazione)

Si riferisce alla capacità di discriminare diversi livelli d'intensità che è legata al numero di bit con i quali è rappresentato il dato. La riduzione del numero di livelli provoca il peggioramento della qualità dell'immagine: appaiono falsi contorni e non è possibile distinguere oggetti che differiscono per variazioni di grigio lente.

La **risoluzione radiometrica** può anche essere definita come **risoluzione di densità (di toni)**, o anche **risoluzione di contrasto**, essa sta ad indicare quanti toni si possono associare ad un pixel, ovvero è legata al  $k$ , si specifica l'immagine con  $M \times N \times k$ . Se tale risoluzione è alta l'immagine appare migliore. La **risoluzione di contrasto** si riferisce al più piccolo cambiamento distinguibile del livello di intensità. Con la **risoluzione di contrasto** ci si riferisce al numero di bit usati per quantizzare l'intensità.

Per esempio si dice che un'immagine quantizzata in 256 livelli ha 8 bit di risoluzione di contrasto.

### 1.5.3 Risoluzione spettrale (larghezza delle bande sulle quali è acquisito il segnale)

Si riferisce alla capacità di discriminare le diverse lunghezze d'onda delle onde elettromagnetiche ricevute.

In base al numero di bande che il sensore può acquisire, si distinguono i sistemi:

- MONOSPETTRALI;
- MULTISPETTRALI (da 2 a poche decine di bande)
- IPERSPETTRALI (molte decine di bande fino a migliaia di bande)

### 1.5.4 Risoluzione temporale

Indica il *periodo di tempo che intercorre tra due riprese successive di una stessa area*.

## 1.6. Relazione di base tra i pixel

I "parametri" che definiscono le relazioni spaziali che intercorrono tra i vari oggetti all'interno di una scena sono:

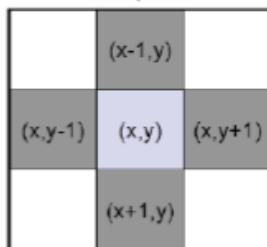
1. Adiacenza
2. Connettività

### 1.6.1 Adiacenza

Prima di definire l'adiacenza, definiamo il concetto di intorno di un pixel. L'intorno di un pixel può essere in due formati:

- **4-intorno o N4(p):** quando consideriamo i quattro pixel orizzontali e verticali, vicini al pixel p di coordinate (x,y), le cui coordinate sono:

Un punto  $P=(x,y)$  appartenente ad un'immagine digitale binaria ha 4 punti vicini in orizzontale e in verticale



Tali punti si definiscono 4-vicini (4-neighbors) di P e si dicono essere 4-adiacenti a P.

Ciascun pixel si trova ad una distanza unitaria da (x,y). Nel caso in cui il pixel p si trovi vicino ai bordi dell'immagine, l'intorno dei pixel può non essere definito.

Oppure quando consideriamo i vicini diagonali, che hanno coordinate:

Lo stesso punto  $P=(x,y)$  ha altri 4 punti vicini disposti in diagonale



Gli 8 punti evidenziati si definiscono 8-vicini (8-neighbors) di P e si dicono essere 8-adiacenti a P.

Un punto 4-adiacente ad un punto P è anche 8-adiacente a P.

- **8-intorno o N8(p):** quando consideriamo sia i vicini orizzontale e verticali che quelli diagonali.

Definiamo tre tipi di adiacenza:

- **4-adiacenza:** due pixel p e q con valori in V sono 4-adiacenti se q è nell'insieme N4(p)

- **8-adiacenza**: due pixel  $p$  e  $q$  con valori in  $V$  sono 8-adiacenti se  $q$  è nell'insieme  $N8(p)$
- **m-adiacenza** (adiacenza mista): due pixel  $p$  e  $q$  con valori in  $V$  sono m-adiacenti se
  - $q$  è in  $N4(p)$ , oppure
  - $q$  è in  $N8(p)$  e l'insieme  $N4(p)$  intersecato  $N4(q)$  non ha pixel i cui valori sono in  $V$ .

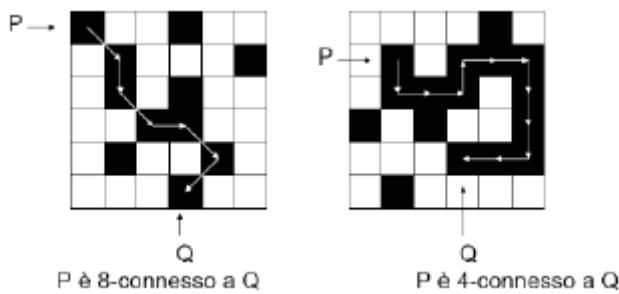
dove  $V$  è l'insieme dei valori di intensità utilizzati per definire l'adiacenza. L'adiacenza mista è stata introdotta per eliminare le ambiguità che sorgono quando si utilizza l'8-adiacenza

## 1.6.2 Connattività

Oltre alle relazioni di adiacenza spaziale, sono importanti nell'analisi di immagini quelle di connessione (connectivity) tra i pixel che stabiliscono i contorni di un'immagine e le componenti di una regione.

Due pixel sono connessi se, oltre ad esserci tra loro una certa relazione di adiacenza spaziale, i loro livelli di grigio soddisfano uno specifico criterio di similarità, per esempio sono uguali. Più in generale, il criterio di similarità può consistere nell'appartenenza di entrambi i valori dei pixel ad uno specifico range di valori di grigio.

Due pixel 4-, 8- o m-connessi stanno tra loro in una relazione di 4-, 8- o m-adiacenza



- **def. Pixel connessi**: Definito  $S$  come un sottoinsieme di pixel in un'immagine, due pixel  $p$  e  $q$  si dicono *connessi* in  $S$  se esiste un path tra di essi formato interamente di pixel appartenenti a  $S$ .
- **Def. Regione**: Definito  $R$  come un sottoinsieme di pixel in un'immagine. Chiamiamo  $R$  una **regione** dell'immagine se  $R$  è un insieme connesso. L'insieme di tutte le regioni presenti in un'immagine forma il **foreground** (primo piano), mentre il suo complemento forma il **background** (sfondo).
- **Def. Bordo**: Il *bordo* di una regione  $R$  è l'insieme dei punti che sono adiacenti a punti dell'insieme complementare di  $R$ , cioè l'insieme dei pixel che hanno almeno un vicino nel background.

## 1.7. Misure di distanza

Dati i pixel  $p$ ,  $q$ ,  $z$  rispettivamente di coordinate  $(x, y)$ ,  $(s, t)$ ,  $(v, w)$ ,  $D$  è una funzione di distanza se:

- $D(p, q) \geq 0$  ( $D(p, q) = 0$  se e solo se  $p = q$ )
- $D(p, q) = D(q, p)$
- $D(p, q) \leq D(p, q) + D(q, z)$

### Def. Distanza euclidea

La distanza euclidea tra **p** e **q** si definisce come:

$$D_e(p, q) = \sqrt{(x - s)^2 + (y - t)^2}$$

Per questa misura di distanza, i pixel con una distanza minore o uguale a un valore  $r$  da  $(x, y)$  sono i punti contenuti in un disco di raggio  $r$  centrati in  $(x, y)$ .

### Def. Distanza city-block

La distanza city-block tra  $p$  e  $q$  è denotata con il simbolo  $D_4$  ed è definita come:

$$D_4(p, q) = |x - s| + |y - t|$$

In questo caso i pixel aventi distanza D4 da (x, y) minore o uguale a un valore r formano un rombo centrato in (x, y).

$$\begin{matrix} & & 2 \\ & 2 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 \\ & & 2 \end{matrix}$$

I pixel con distanza D4 = 1 sono i 4-vicini di (x, y).

#### Def Distanza a scacchiera

La distanza a scacchiera tra p e q è denotata con il simbolo D8 ed è definita come:

$$D_8(p, q) = \max(|x - s|, |y - t|)$$

In questo caso i pixel con distanza D8 minore o uguale a un valore r formano un quadrato centrato in (x, y).

$$\begin{matrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{matrix}$$

## 1.8. Operazioni tra i pixel

### 1.8.1 Operazioni aritmetiche

Le operazioni aritmetiche tra due pixel, p e q, sono quelle ordinarie di

- addizione:  $p + q$
- sottrazione:  $p - q$
- moltiplicazione:  $p \times q$
- divisione:  $p / q$

Il risultato dell'operazione in ogni posizione (x,y) dipende solo dai valori dei pixel nella stessa posizione nelle immagini di partenza, le operazioni aritmetiche trovano largo impiego nelle elaborazioni appartenenti alla classe delle **operazioni puntuali** (*pixel-by-pixel operations*)

Appartengono invece alla classe delle **operazioni locali** (*neighborhood operations*) le elaborazioni che coinvolgono i valori dei pixel in un intorno, generalmente piccolo, della posizione (x,y) di interesse.

### 1.8.2 Operazioni matriciali e puntuali

Spesse volte le immagini possono essere trattate come matrici, per questo motivo è utile effettuare una distinzione tra operazioni matriciali e puntuali (pixel per pixel).

Prendiamo in considerazione queste due immagini:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

Il loro prodotto Puntuale corrisponde a:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Mentre il loro prodotto matriciale (riga x colonna) a:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

### 1.8.3 Operazioni logiche

Le principali operazioni logiche usate nella elaborazione delle immagini sono quelle solite di:

- AND :  $p \cdot q$  oppure  $p$  AND  $q$
- OR :  $p + q$  oppure  $p$  OR  $q$
- NOT : "q" oppure NOT  $q$

Le operazioni logiche si applicano di norma ad immagini binarie. L'uso più interessante si ha nelle operazioni di estrazione di caratteristiche, analisi della forma, applicazione di maschere, ecc. L'applicazione ad intere immagini si effettua tra pixel omologhi delle due immagini, per tutti i pixel.

Tavole di verità degli operatori logici:

<b>AND</b>	<b>0</b>	<b>1</b>
<b>0</b>	0	0
<b>1</b>	0	1

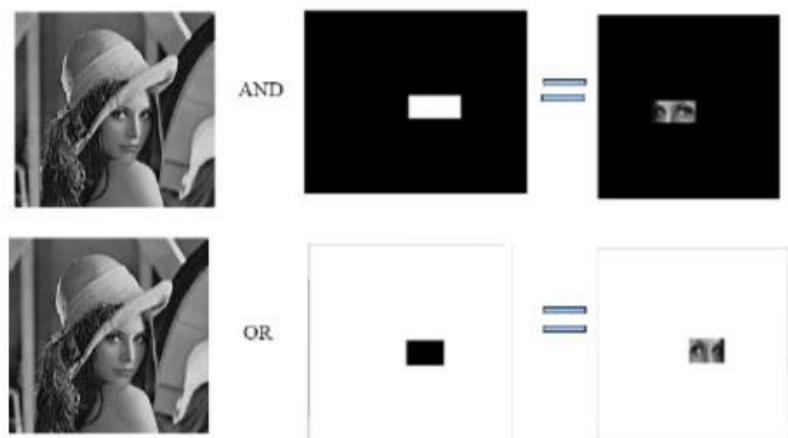
<b>OR</b>	<b>0</b>	<b>1</b>
<b>0</b>	0	1
<b>1</b>	1	1

<b>XOR</b>	<b>0</b>	<b>1</b>
<b>0</b>	0	1
<b>1</b>	1	0

### 1.9 Roi (region of interest)

Le operazioni logiche di AND e di OR possono essere utilizzate, in particolare per l'estrazione dalla immagine di una parte significativa (**ROI – Region of Interest**) che debba essere successivamente elaborata. Questa operazione, detta anche di masking, è effettuata bit per bit, tra pixel nella stessa posizione nelle due immagini. Per esempio, se un pixel vale 10010011 si ha:

10010011 AND 11111111 = 10010011  
 10010011 AND 00000000 = 00000000  
 10010011 OR 11111111 = 11111111  
 10010011 OR 00000000 = 10010011



## 1.8.4 Operazioni spaziali

### 2.6.4.1 Operazioni a singolo pixel

L'operazione più semplice che possiamo eseguire su un'immagine è quella di alterare il valore dei pixel basandoci sulla loro intensità.

Questo processo può essere espresso come una funzione di trasformazione,  $T$ , della forma:

$$s = T(z)$$

### 2.6.4.2 Operazioni basate sull'intorno

Denotiamo con  $S_{x,y}$  l'insieme delle coordinate di un intorno centrato su un punto arbitrario  $(x, y)$  in un'immagine  $f$ . Le operazioni basate sull'intorno generano un pixel alle stesse coordinate nell'immagine di input. Per esempio, supponiamo che l'operazione consista nel calcolare la media dei valori dei pixel in un intorno rettangolare di dimensioni  $m \times n$  centrato in  $(x, y)$ . L'immagine  $g$  è ottenuta variando le coordinate  $(x, y)$  in modo che il centro dell'intorno si sposti di pixel in pixel nell'immagine  $f$ , ripetendo l'operazione di media per ciascuna nuova posizione.

### 2.6.4.3 Trasformazioni spaziali geometriche e registrazione di immagini

Le **trasformazioni geometriche** modificano le relazioni spaziali tra i pixel di un'immagine. Un esempio è lo zoom. Lo zooming richiede due fasi:

1. creazione di nuove locazioni di pixel
2. assegnazione dei valori di grigio ai nuovi pixel.

Se per esempio si vuole effettuare uno zoom di 1.5 volte di una immagine  $500 \times 500$ , evidentemente la nuova griglia di campionamento sarà  $750 \times 750$ . Infatti, ogni volta che su un'immagine viene eseguita una trasformazione geometrica o un cambiamento di scala, il risultato è che i pixels si trovano in una nuova posizione più accurata ma la cui radiometria non rappresenta i valori reali misurati sulla scena. Una stima dei nuovi indici dei pixels viene fatta mediante procedure di interpolazione che sono dette tecniche di ricampionamento. Vediamone tre tipi:

- **interpolazione nearest neighbor:** assegna a ogni nuova posizione l'intensità del pixel più prossimo nell'immagine originale. Questo metodo è poco utilizzato perché introduce **artefatti**.
- **interpolazione bilineare:** l'intensità da assegnare a ciascuna nuova posizione è stimata tramite i quattro pixel più vicini. L'equazione che regola questo metodo è

$$v(x,y) = ax + by + cxy + d$$

dove  $v$  è il valore dell'intensità; mentre i quattro coefficienti sono determinati a partire dalle quattro equazioni nelle quattro incognite ottenibili utilizzando i quattro pixel più vicini al punto  $(x,y)$ . Ovviamente questo tipo di interpolazione offre dei risultati migliori rispetto alla precedente.

- **interpolazione bicubica:** è come la bilineare con la differenza che questa volta consideriamo i 16 pixel più vicini. Quindi avremo 16 coefficienti determinati con 16 equazioni in 16 incognite.

## 2.Tecniche puntuali per il miglioramento della qualità

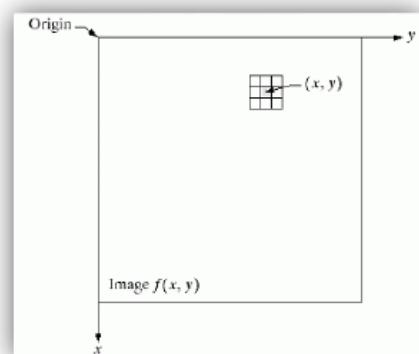
### 2.1 Funzioni per la trasformazione di intensità

Le trasformazioni di intensità e i filtri trattati in questo capitolo lavorano nel **dominio spaziale**, cioè nel piano che contiene i pixel dell'immagine, andando ad applicare una manipolazione diretta dei pixel. L'espressione che riassume i processi nel dominio spaziale, quindi il **filtraggio spaziale**, è la seguente:

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

dove  $f(x, y)$  è l'immagine di input contenente valori di intensità luminosa compresi tra  $[0, L-1]$ ,  $g(x, y)$  è l'immagine di output e  $T$  è l'operatore definito su  $f$  nell'intorno di un punto  $(x, y)$ . L'intorno solitamente è rettangolare, centrato su  $(x, y)$  e piccolo. La dimensione dell'intorno di  $(x, y)$  definisce il carattere della elaborazione:

- **Puntuale** (l'intorno coincide con il pixel stesso);
- **Locale** (nei casi più comuni l'intorno è una piccola regione quadrata centrata sul pixel);
- **Globale** (l'intorno coincide con l'intera  $f$ ).
- **Sequenziali**
- **Parallele**



L'operatore  $T$  applicato all'immagine può essere:

- **Omogeneo**, cioè si comporta allo stesso modo con i pixel indipendentemente dalla loro posizione ma **dipendentemente** soltanto dal valore di intensità luminosa
- **Non omogeneo**, in questo caso il *comportamento dipende sia* dal valore di intensità luminosa che *dalla posizione assunta dal pixel* stesso. Quindi non solo dal valore di  $f$  ma anche dagli argomenti della  $f$ .

#### Elaborazioni puntuali omogenee: Operatore Omogeneo

Il risultato di una elaborazione puntuale omogenea dipende solo dal valore del pixel cui è applicata, per cui tali elaborazioni vengono anche dette **manipolazioni della scala dei grigi** o dei colori. Ponendo  $s = g(x, y)$  e  $r$  un qualsiasi valore di intensità luminosa della  $f$  possiamo scrivere:

$$s = T(r)$$

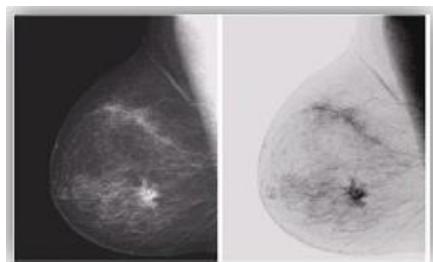
indicando la stessa trasformazione.

I filtri solitamente riguardano il miglioramento dell'immagine o **enhancement**, cioè quel processo di manipolazione dell'immagine tale che il risultato sia più adatto dell'originale a una specifica applicazione. Vediamo ora tre tipologie di funzioni per le trasformazioni di intensità:

1. **lineari (negative e identità)**

**Negativo**: si ottiene tramite l'espressione

$$s = (L - 1) - r$$



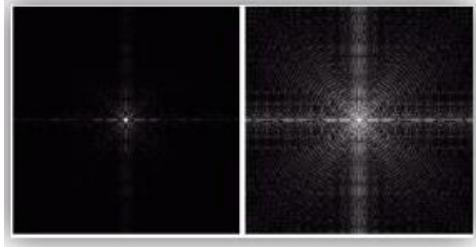
con un'immagine nella gamma  $[0, L-1]$ . Il risultato ottenuto è l'equivalente di un negativo fotografico, infatti vengono invertiti i livelli di intensità dell'immagine. Questa elaborazione è utile per migliorare i dettagli bianchi o grigi inseriti nelle regioni scure di un'immagine.

## 2. logaritmica (log e log inverso)

L'espressione che regola la trasformazione logaritmica è la seguente:

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

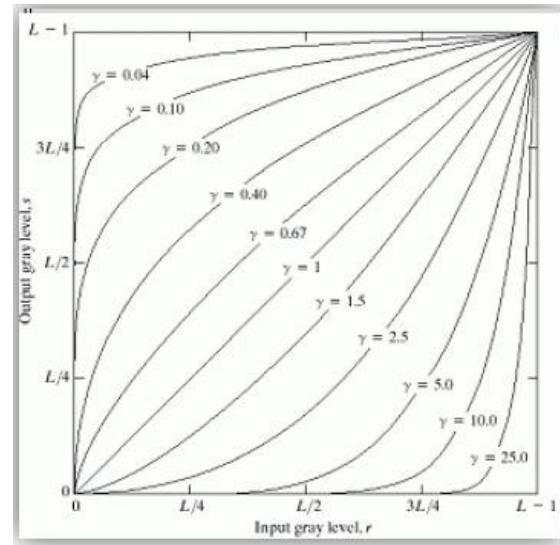
dove  $c$  (solitamente vale  $L-1$ ) è una costante che va scelta opportunamente per far rientrare i valori trasformati nel range corretto  $[0, L-1]$  ed  $r \geq 0$ , cioè si riscalano i valori. Si aggiunge 1 per evitare il caso di  $\log(0)$ .



In questo caso stiamo comprimendo la prima parte della dinamica (da 0 ad  $L-1/2$ ) e espandendo la seconda parte (da  $L-1/2$  ad  $L-1$ ). La caratteristica della funzione logaritmica è quella di **comprimere** la gamma dinamica delle immagini che presentano ampie variazioni dei valori dei pixel, permettendo di avere maggiore dettaglio.

## 3. di potenza [gamma] (potenza n-esima e radice n-esima)

Questa trasformazione è anche detta **Gamma Correction**, l'esponente appunto gamma può assumere valori compresi tra 0 e 1 (rappresenta una Radice poiché un numero elevato a 0.5 non è altro che la radice quadrata del numero) oppure valori maggiori di 1 (rappresenta una potenza). Nel primo caso ovvero  $0 <= \text{gamma} <= 1$  stiamo comprimendo la seconda parte della dinamica (da  $L-1/2$  ad  $L-1$ ) e espandendo la prima parte (da 0 ad  $L-1/2$ ). Espandere la prima parte equivale a rendere più chiara l'immagine. È una trasformazione inversa alla precedente, infatti al posto del logaritmo usiamo un esponenziale. Le trasformazioni di potenza chiamate anche **trasformazioni gamma** hanno la forma:



$$s = c r^\gamma$$

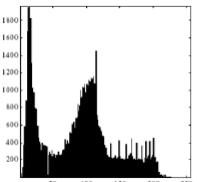
dove  $c$  ( $L-1$ ) e  $\gamma$  sono costanti positive e l'esponente è detto **gamma**. Variando  $\gamma$  è possibile ottenere varie curve di trasformazione che permettono di migliorare la qualità dell'immagine, migliorare il contrasto e permettere una corretta visualizzazione dell'immagine sullo schermo. Immagini che non hanno un corretto valore di  $\gamma$  possono apparire sbiadite o troppo scure. Questo concetto vale non solo per le immagini in toni di grigio, ma anche per le immagini a colori, infatti la correzione gamma non modifica soltanto l'intensità, ma anche le percentuali di rosso, verde e blu, quindi un corretto valore di gamma permette di visualizzare i colori in modo fedele.

## 2.2 Iistogrammi

L'istogramma è un modo per descrivere un'immagine digitale misurandola statisticamente. Viene calcolato in modo molto semplice, preso un sistema di assi cartesiani:

- sulle ascisse indichiamo tutti i valori di intensità luminosa che possiamo ottenere (0 a L-1)
- sull'asse delle ordinate le occorrenze di intensità luminosa, cioè quanti ne sono.

Quello che si fa è contare le occorrenza di ogni intensità luminosa su tutta l'immagine. Il caso peggiore che possiamo avere è che l'immagine sia piatta, cioè l'immagine è formata dalla stessa intensità luminosa quindi tutte le occorrenze sono 0, per esempio, tranne una che assume valore  $M \times N$  (dove MN sono le dimensioni dell'immagine) che indica che il valore 0 è ripetuto MN volte su tutta l'immagine. Vi è una particolarità, se dividiamo tutti i valori delle ordinate per MN otteniamo che i valori delle ordinate saranno compresi tra 0 e 1 e il valore rappresenta la **Probabilità di occorrenza**, di un certo valore di intensità luminosa.



L'**istogramma** di un'immagine digitale con livelli di intensità nella gamma  $[0, L-1]$  è una funzione discreta

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

dove  $r_k$  è il valore d'intensità k-esimo ( $k=0 \dots L-1$ ) ed  $n_k$  è il numero di pixel dell'immagine con intensità  $r_k$  ed  $n$  è il numero totale di pixel.

Se l'immagine risultante è finalizzata alla visualizzazione, i valori dei pixel devono restare entro la gamma dei livelli di grigio rappresentabili (es. 0 - 255). A questo scopo, i coefficienti del filtro devono soddisfare la condizione:

$$R = \frac{1}{mn} \sum_{i=1}^{mn} \omega_i = 1$$

La somma dei coefficienti del filtro deve essere uguale a 1. In tal modo una zona a valore di grigio costante entro la maschera del filtro resta immutata dopo il filtraggio. Questo processo è detto **Normalizzazione**.

La *normalizzazione* di un istogramma avviene dividendo ogni componente per il numero totale dei pixel dell'immagine, quindi tramite la seguente espressione:

$$p(r_k) = \frac{n_k}{MN} \quad \text{per } k = 0, 1, \dots, L-1$$

$p(r_k)$  può essere visto come una stima della probabilità dell'occorrenza del livello di intensità  $r_k$  in un'immagine. La somma delle componenti di un istogramma normalizzato deve dare 1.

### 2.2.1 Calcolo di un Iistogramma

L'operazione di calcolo di un istogramma è il **conteggio del numero di volte che ogni valore è presente nell'immagine**. Il risultato è quindi un vettore di interi che indicherà il numero di occorrenze. Le istruzioni che permettono facilmente di ottenere un istogramma di un'immagine:

```
for (i=0; i < M-1; i++)  
{  
    for (j=0; j < N-1; j++)  
    {  
        istogramma[immagine(i, j)]++;  
    }  
}
```

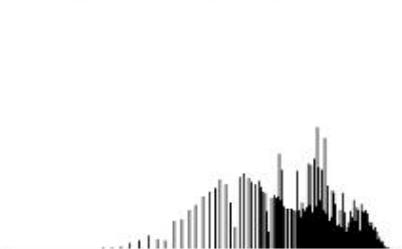
M e N sono il size dell'immagine, istogramma è un array che conterrà i valori di intensità luminosa determinati da  $\text{immagine}(i,j)$  che sarà utilizzato come indice . Ogni volta che si incontra un valore di intensità luminosa si incrementa il valore dell'istogramma. La sua complessità computazionale è  $M \times N$  ovviamente perché bisogna scansionare tutta l'immagine.

Consideriamo, per semplicità, immagini sempre in **scala di grigio**.

Se ,una volta costruito l'istogramma di un immagine, otteniamo un istogramma che tende verso destra cioè abbiamo un numero elevato di valori di intensità luminosa nella parte alta del range della dinamica cioè da  $L-1/2$  e  $L-1$ . Questo significa che l'immagine è **Sovraesposta** (molto luminosa e quindi poco contrastata). Viceversa, se otteniamo un istogramma che tende verso sinistra significa che abbiamo un numero elevato di valori di intensità luminosa nella prima parte del range della dinamica cioè tra 0 e  $L-1/2$ , ottenendo così un immagine **Sottoesposta** (scura).

Esempio:

*Immagine sovraesposta*



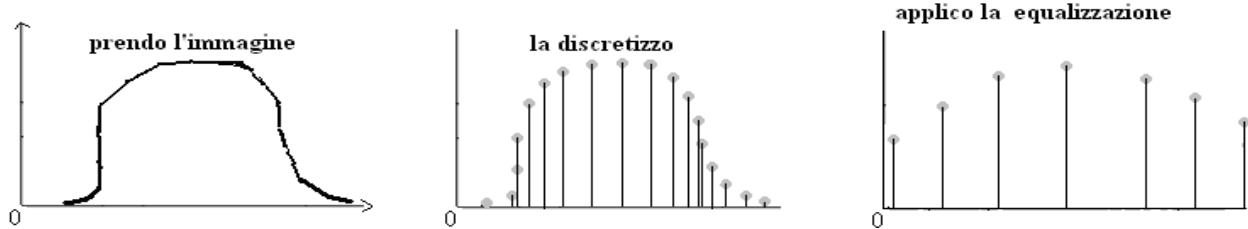
*Immagine sottoesposta*



Si noti che alcuni autori considerano non puntuali ma **globali** le elaborazioni di questo tipo, in quanto l'istogramma rappresenta una descrizione in un certo senso globale delle caratteristiche visuali dell'immagine. In senso stretto, però, la **manipolazione dell'istogramma** è una operazione puntuale. Il risultato ottimo sarebbe ottenere un **equilibrio** di risultati, cioè senza squilibri né nella parte destra né nella parte sinistra. Se riuscissimo a distribuire in modo equo i valori di intensità su tutto il range avremmo ottenuto un **equalizzazione dell'istogramma** ovvero un istogramma senza squilibri né verso destra (sottoesposta) né verso sinistra(sovraesposta).

## 2.2.2 Equalizzazione di istogrammi

L'**equalizzazione** serve per **migliorare il contrasto** e produce un'immagine con istogramma “**piatto**”, cioè con livelli di grigio uniformemente distribuiti sui pixel (ogni livello di grigio caratterizza lo stesso numero di pixel dell'immagine). Si ottiene pertanto un'immagine con densità uniforme dei livelli di grigio. I livelli di grigio dell'immagine equalizzata spaziano su tutta la dinamica e raggiungono sempre il bianco



Consideriamo  $r$ , una variabile continua che indica il valore di intensità luminosa in ingresso, normalizzata in modo che  $0 \leq r \leq 1$  ( $r=0$  nero,  $r=1$  bianco, se non normalizziamo, allora  $r=0$  nero, mentre  $r=L-1$  bianco), l'obiettivo è trovare una funzione  $T$  che applicata ad  $r$  quindi  $T(r)$ , produca un valore di intensità luminosa in uscita cioè  $s$ :

$$s = T(r)$$

calcoliamo la **Pr( $r$ )** cioè la **densità di probabilità di  $r$** , facendo l'istogramma dell'immagine originale, inserendo  $r$  sulle ascisse e il numero di occorrenze di  $r$  (0-255) sulle ordinate, dividiamo per il numero complessivo di valori dell'immagine (MN) e otteniamo valori compresi tra 0 e 1.

$$s_k = T(r_k) = (L - 1) \sum_{k=0}^k p_r(r_k) = \frac{(L - 1)}{MN} \sum_{k=0}^k n_k$$

Dove  $p(r_k) = n_k/MN$  è la probabilità dei livelli di intensità. Quindi un'immagine equalizzata si ottiene trasformando ogni pixel dell'immagine di input di intensità  $rK$  in un pixel con intensità  $sK$

Quindi a questo punto abbiamo una **densità di probabilità**. Quello che desideriamo è che  $P(s)$  sia **equa**. Cioè che tutti i valori abbiano la stessa occorrenza e che le mutue realazioni tra i due livelli di intensità luminosa  $r$  e  $s$  rimangano inalterati, in altre parole se  $r$  si trova prima della trasformazione nella parte bassa del range della dinamica dopo la trasformazione tramite  $T$  questo aspetto deve rimanere inalterato. Un altro aspetto da rispettare è che dati  $r_1$  e  $r_2$  due valori di  $r$  se

$$r_1 < r_2 \text{ implica che } s_1 < s_2 \text{ in } s,$$

questo ci dice che la funzione  $T$  è una funzione **Monotona Crescente**.

Inostre se :

$$r_1 \neq r_2 \text{ implica che } T(r_1) \neq T(r_2)$$

cioè che  $T$  sia **univoca**.

$T$  deve anche essere **invertibile**, cioè possiamo ritornare indietro dopo la trasformazione. Se  $T$  è invertibile esiste  $T^{-1}$ , e possiamo ricavare  $r$ :

$$r = T^{-1}(s)$$

Anche qui la funzione  $T^{-1}$  deve essere una funzione monotona crescente ( $s_1 < s_2$  implica che  $r_1 < r_2$ ).

Un'altra condizione da rispettare è che dati  $r_1, r_2$  appartenenti a  $r$  e  $s_1, s_2$  appartenenti a  $s$  definiamo  $dr = r_2 - r_1$  differenziale di  $r$  e  $ds = s_2 - s_1$  differenziale di  $s$ , possiamo dire che:

$$\Pr(r) dr = Ps(s)ds$$

Cioè che la densità di probabilità dopo la trasformazione rimane **invariata**.

Definite queste condizioni possiamo scrivere che:

$$Ps(s) = \Pr(r) \frac{dr}{ds}$$

Dove

$$s = T(r)$$

L'obiettivo finale è quello di ottenere un istogramma "piatto" cioè che sia equo, ovvero che  $Ps(s) = k$  cioè sia una **costante**. Per soddisfare questa condizione bisogna scegliere opportunamente la funzione  $T$ , la soluzione è scegliere  $T$  come **Funzione Cumulativa**, cioè:

$$s = T(r) = \int_0^r \Pr(r) dr$$

cioè facciamo corrispondere al valore 0  $\rightarrow \Pr(0)$ , al valore 1  $\rightarrow \Pr(0)+\Pr(1)$ , al valore 2  $\rightarrow \Pr(0)+\Pr(1)+\Pr(2)\dots$  cioè cumuliamo i valori.

LA funzione cumulativa soddisfa le condizioni , cioè sia monotona crescente con valori tra 0 e 1. Ora derivando rispetto a  $r$  si ottiene:

$$\frac{ds}{dr} = \Pr(r)$$

Ora riconsiderando  $Ps(s) = \Pr(r) \frac{dr}{ds}$  e portando i differenziali al primo membro otteniamo:

$$Ps(s) \frac{ds}{dr} = \Pr(r)$$

Sostituendo e portando al secondo membro:

$$Ps(s) = \frac{\Pr(r)}{\Pr(r)} = 1$$

Quindi la funzione cumulativa, scelta come l'integrale precedente ci garantisce l'invertibilità e il rispetto delle mutue relazioni tra  $r$  e  $s$ .

Tornando a considerare discreta la variabile  $r$  si ha:

$$p(r_k) = \frac{n_k}{n} \quad \text{per } 0 \leq r_k \leq 1 \text{ e } k = 0, \dots, L-1$$

La funzione da utilizzare per l'equalizzazione dell'istogramma nel caso discreto è la seguente:

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j), \quad 0 \leq r_k \leq 1 \text{ e } k = 0, 1, \dots, L-1$$

La trasformazione inversa è  $r_k = T^{-1}(s_k)$ , con  $0 \leq s_k \leq 1$

Sia  $T(r_k)$  che  $T^{-1}(s_k)$  devono soddisfare le condizioni a) e b) precedentemente viste

La  $T(r_k)$  si calcola direttamente dall'immagine, mentre il calcolo della  $T^{-1}(s_k)$  non è necessario per l'equalizzazione.

**L'equalizzazione dell'istogramma** ha lo **svantaggio** che si ottengono effetti come lo sgranamento dell'immagine e la comparsa di false regioni ovvero di classi inesistenti nella realtà, è l'approssimazione nel calcolo dei nuovi livelli, dovuta alla natura discreta dei livelli stessi, a causare la non uniformità dell'istogramma, però gode di un ottimo **vantaggio**, quello di essere un metodo completamente

automatico, ma proprio per questo è limitato nell'applicazione: infatti non è possibile specificare forme particolari dell'istogramma, che possono invece essere utili quando si vogliono evidenziare determinati intervalli della gamma dei livelli di grigio. Per questo motivo una seconda tecnica utilizzata per ottenere un istogramma uniforme è la **Specificità dell'istogramma** (o matching tra istogrammi).

### 2.2.3 Specifica di un istogramma

Parliamo di *specificità di un istogramma* quando **indichiamo la forma dell'istogramma** che vogliamo ottenere dal processo di enhancement. Cioè quando vogliamo, per esempio, aumentare la distribuzione dei livelli di grigio del bianco rispetto ad altri ecc.

Il procedimento da seguire è il seguente:

- **calcolare l'istogramma** dell'immagine originale ed effettuare **l'equalizzazione** dell'istogramma applicando la distribuzione cumulativa sull'intero istogramma ottenendo un istogramma "idealmente piatto"
- **creare l'istogramma "desiderato"** come risultato finale ed **equalizzarlo** ottenendo ancora una volta un istogramma "idealmente piatto"
- **switchare** i due istogrammi equalizzati e applicare l'operazione inversa dell'equalizzazione cioè applicare la **densità di probabilità** invece che la funzione cumulativa
- a questo punto il risultato è quello che volevamo ottenere inizialmente, l'istogramma modificato come indicato.

# 3. Tecniche di analisi nel dominio spaziale

## Filtraggio spaziale

Il **filtraggio spaziale** consiste in una *regione di prossimità* (solitamente un rettangolo) e un'*operazione predefinita* che viene applicata ai pixel dell'immagine appartenenti alla regione **creando nuovi pixel**, il cui valore è il risultato dell'operazione di filtraggio. In generale il filtraggio spaziale di un'immagine di dimensioni  $M \times N$  con un filtro di dimensioni  $m \times n$  è dato dalla seguente espressione:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t), \text{ con } a = \frac{m-1}{2} \text{ e } b = \frac{n-1}{2}$$

dove  $x$  e  $y$  variano in modo tale che ogni pixel di  $w$  visiti ogni pixel di  $f$ .

I filtri spaziali si dividono in:

- **Filtri di Smoothing**
- **Filtri di Shaping**

## 3.1 Filtri di smoothing

I filtri spaziali di smoothing sono usati per il **blurring** dell'immagine e per la riduzione del rumore. L'operazione di **blurring** è normalmente utilizzata in fase di pre-elaborazione, allo scopo di eliminare piccoli dettagli inutili o addirittura dannosi per elaborazioni, ovvero si cerca di compensare piccole imperfezioni, quali le interruzioni che spesso si verificano nelle linee di contorno. I filtri di smoothing, servono ad eliminare le brusche transizioni ed il rumore.

### 3.1.1 Filtri di media

Questi filtri sostituiscono il valore di ogni pixel dell'immagine con la **media** dei livelli di intensità nella regione definita dalla maschera del filtro. Applicando questo tipo di filtro i risultati ottenuti sono:

- ridurre le transizioni di intensità brusche, spesso dovute alla presenza di rumore.
- sfocare l'immagine, per far emergere oggetti di interesse
- eliminare i falsi contorni dovuti all'utilizzo di un numero insufficiente di livelli d'intensità.
- ridurre i dettagli irrilevanti
- introduce dei bordi neri quando si utilizzano delle maschere di grandi dimensioni.

In generale ci sono due tipi di *filteri media*:

- **filtro box**: filtro di media spaziale in cui tutti i coefficienti sono uguali.  
$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$
- **filtro di media ponderata**: filtro che fa in modo che i pixel vengano moltiplicati per coefficienti diversi, dando maggiore peso ad alcuni pixel piuttosto che ad altri.  
$$\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

### Filtro Media Mobile

Mediante questa tecnica si sostituisce al valore di ogni campione la media di campioni vicini. E' una tecnica matematica utilizzata per smussare le fluttuazioni nel segnale. Si dice "mobile" perché il numero degli elementi considerati è fisso (finestra), ma l'intervallo di tempo avanza.

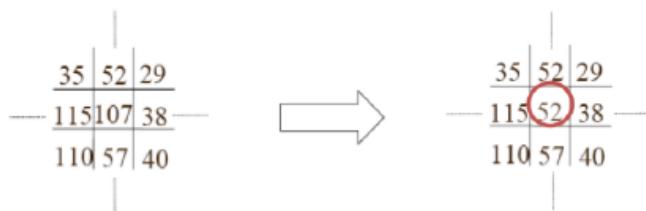
$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Il fattore moltiplicativo serve a normalizzare la somma dei pesi a 1. In tal modo il range dei valori di output risulta uguale a quello dei valori di input

### 3.1.2 Filtri di smoothing non lineari basati sulle statistiche d'ordine

#### Filtro mediano

Il **filtro mediano** (es. 3x3) è un filtro opera in un intorno 3x3 centrato in un pixel, l'idea è quella di eliminare il **rumore Impulsivo** dove per rumore impulsivo si intende rumore che altera il valore dell'intensità luminosa portandola a 0 oppure a 255 e di conseguenza quello che si deve fare è non distribuire il rumore su tutto l'intorno sul quale opera il filtro ma soltanto sul valore nel quale esso è centrato. Nell'esempio sottostante sostituisce il valore di un pixel con il valore mediano delle intensità della regione in cui si trova il pixel. Per esempio:



Il mediano  $M$  di un insieme di valori è tale che metà dei valori sono minori di  $M$  e metà dei valori sono maggiori di  $M$ . Pertanto, nell'intorno di ogni pixel si devono ordinare i valori dei pixel in esso contenuti, compreso quello centrale, ed il valore mediano di essi è assegnato al pixel centrale. Nell'esempio:

$$29, 35, 38, 40, \mathbf{52}, 57, 107, 110, 115$$

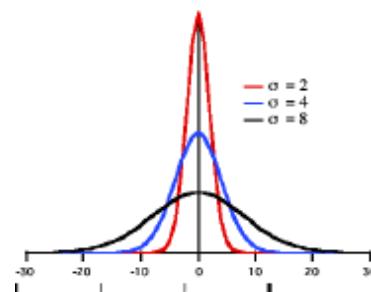
Quindi l'effetto del filtro mediano è di forzare i pixel ad assumere un valore uguale a quello di uno dei pixel circostanti che sono quelli con cui di solito si manifesta il cosiddetto rumore **impulsivo** (o salt-and-pepper). Non funziona bene con il rumore additivo (o gaussiano) dove dovrebbe essere usato il filtro di smoothing lineare (blurring). Questo modo non lineare di operare consente di non distribuire il rumore su tutta la maschera ma solo su quel pixel, da questo filtro nascono anche altri filtri spaziali : **K-Trimmmed**, cioè possiamo selezionare il massimo, il minimo, il secondo massimo e via dicendo.

#### Filtro Gaussiano

Il filtro Gaussiano è il filtro più efficiente per l'eliminazione del rumore dalle immagini, per questo risulta essere anche il filtro più diffuso ed usato. L'idea è quella di eliminare il rumore additivo che è stato aggiunto all'immagine originale. **Si utilizza quando si vuole che il pixel in esame abbia maggiore importanza nel calcolo del nuovo valore di grigio, rispetto a quelli del suo intorno: il peso di tale pixel sarà maggiore rispetto a quello dei pixel dell'intorno**

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1



Caso monodimensionale:

l'equazione di una gaussiana  $G=G(x)$  con media nulla è data dall'equazione:

$$G(x) = A e^{-\frac{x^2}{2\sigma^2}} \text{ dove } A = \frac{1}{\sigma\sqrt{2\pi}}$$

Dove A è un **fattore di normalizzazione**, cioè garantisce che la gaussiana sia una distribuzione di probabilità quindi in modo tale che l'integrale tra meno inf e più inf di questa gaussiana ( $G(x) dx$ ) sia = 1.

Nel caso bidimensionale l'equazione della gaussiana con mesia (0,0) è:

$$G(x, y) = A e^{-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}} \text{ dove } A = \frac{1}{2\pi\sigma_x\sigma_y}$$

è un filtro **separabile**, cioè l'esponenziale di una somma è separabile come il prodotto di due esponenziali calcolando quindi la gaussiana nella direzione delle x e nella direzione delle y. Questo è un grande vantaggio poiché se riusciamo a costruire un filtro basato sulla gaussiana bidimensionale, essendo il kernel separabile otteniamo un vantaggio computazionale cioè invece di ottenere **O(M^2 per k^2)** (Convoluzione bidimensionale) abbiamo **O(kM)** (Convoluzione monodimensionale).

Ora, per progettare un filtro nel dominio spaziale a partire da una gaussiana bisogna discretizzare la gaussiana cioè si effettua un campionamento prendendo dei punti applicando una griglia sulla gaussiana. Prendiamo dei punti in modo tale che  $\sigma_x = \sigma_y$ , cioè le varianze nella direzione x e y sono uguali in modo tale che una sezione trasversale della gaussiana sia una circonferenza utilizzando come relazione:

$$\frac{G(i, j)}{A} = e^{-(\frac{i^2}{2\sigma^2} + \frac{j^2}{2\sigma^2})}$$

La maschera corrispondente ad una tale funzione dipende dalle dimensioni della maschera stessa e dal valore di  $\sigma^2$ . Se ad esempio  $\sigma^2 = 3$  e la maschera è 5x5 i valori di i e j sono limitati ai valori  $-2 \leq i, j \leq 2$  ed il calcolo della maschera procede nel seguente modo:

Ora sostituiamo nell'equazione di sopra i valori per ogni punto della maschera ottenendo:

- $G(0,0) = 1$  poiché l'esponenziale elevato a 0 da 1.
- $G(1,0) = 0.85$
- $G(1,1) = 0.71$
- $G(2,0) = 0.51$
- $G(2,1) = 0.43$
- $G(2,2) = 0.26$

I valori che si ottengono sono compresi tra 0 e 1 poiché è normalizzata. Ma all'interno della maschera noi inseriamo valori interi quindi bisogna portare i valori ottenuti ad un valore intero relativo. Per fare ciò basta trovare una costante moltiplicativa B in modo tale che il valore più piccolo (0.26 in questo caso) moltiplicato per B ritorni 1. In altre parole  $1 = B * 0.26$ , da cui  $B = 1/0.26$ , cioè  $B = 3.85$ . Ora moltiplicando tutti i valori per la costante e arrotondando all'intero più vicino otteniamo:

- $G(0,0) = 4$
- $G(1,0) = 3$
- $G(1,1) = 3$
- $G(2,0) = 2$
- $G(2,1) = 2$
- $G(2,2) = 1$

Da notare che non abbiamo calcolato tutti i coefficienti del kernel, questo perché il kernel è separabile, l'altra parte si calcola di rimando.

### Filtro di Rango

I filtri di rango operano sui pixel contenuti nella finestra individuata dalla maschera ordinandoli in senso crescente secondo il loro valore.

Una volta terminato l'ordinamento si applica la scelta in base ad un opportuno criterio.

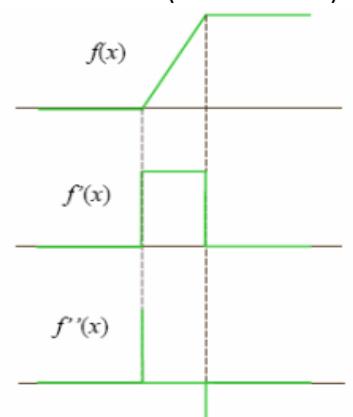
- **Operatore minimo:** viene scelto il primo valore della sequenza;
- **Operatore massimo:** viene scelto l'ultimo valore della sequenza;
- **Filtro Mediano:** viene scelto il valore mediano tra quelli presenti nella finestra
- **Operatore medio k-trimmed:** assegna al pixel il valore medio calcolato utilizzando una finestra di ampiezza  $k$  centrata sul pixel mediano  $M$ . Al variare di  $k$  tra 0 ed  $M$  si ottiene una riduzione del rumore presente nell'immagine: se  $k \rightarrow M$  si riduce il rumore gaussiano, se  $k \rightarrow 0$  si riduce il rumore impulsivo.

## 3.2 Filtri di sharpening

I filtri di **sharpening** sono utilizzati per evidenziare i contorni degli oggetti. Non è applicabile ad un'immagine corrotta da rumore, ma bisogna applicarlo ad un'immagine alla quale è stato già applicato precedentemente un filtro di smoothing. La maschera non gode della proprietà che la somma dei suoi valori sia uguale a 1 poiché nella zona costante non vogliamo che il valore rimanga inalterato ma che si **annulli** poiché vogliamo evidenziare le zone di contrasto, per questo motivo il nostro obiettivo è che i valori della maschera si annullino. Per ottenere questo possiamo far sì che il valore centrale della maschera venga **intensificato** mentre vengano ridotte le intensificazioni nell'intorno. Un modo semplice di implementare un filtro di sharpening è quello di appellarsi alle **derivate spaziali del I e II ordine**.

Considerando un'immagine a livelli di grigio costante dove sono presenti punti linee e bordi (discontinuità):

- **La derivata I:**
  - Nulla nelle zone piatte
  - Non nulla in presenza di una discontinuità
  - Non nulla lungo una rampa
- **La derivata II:**
  - Nulla nelle zone piatte
  - Non nulla all'inizio e alla fine di una discontinuità
  - Nulla lungo una rampa di pendenza costante



Ricordiamo Derivata del **primo ordine**:

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

La derivata è definita come il limite del rapporto incrementale in un intorno fratto la dimensione dell'intorno stesso quando la dimensione dell'intorno stesso tende a 0, questo se la funzione è continua. Nel caso discreto lo si può considerare come la differenza dei valori di intensità della funzione nell'intorno stesso del punto. Nel caso precedente i pixel vicini erano  $x$  e  $x+1$  poiché  $\frac{\partial f}{\partial x}$  viene definito usando il pixel successivo:

- non si può calcolare sull'ultimo pixel di una riga;

- è diverso da zero nel pixel che precede un gradino.

Derivata del **secondo ordine**:

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= f(x+1) - f(x) - (f(x) - f(x-1)) \\ &= f(x+1) - 2f(x) + f(x-1)\end{aligned}$$

La derivata del secondo ordine non è altro che la derivata della derivata. Da notare che i coefficienti delle  $f(x)$  sommandoli otteniamo 0 (1-2+1). Cioè i coefficienti dei valori della funzione rispetto ai quali calcolo le derivate si sommano a 0 questo indica che la derivata di una funzione costante è 0.

I filtri di sharpening sono utilizzati per aumentare il contrasto locale dell'immagine, in modo da arricchire i dettagli fini, o per evidenziare i contorni degli oggetti. A tal fine questi filtri lavorano aumentando le differenze tra pixel vicini. Sono complementari ai filtri di smoothing e possono provocare l'aumento del rumore presente nell'immagine, per cui la loro azione va spesso compensata con quella di un filtro per la riduzione del rumore.

### 3.3 Filtro Gradiente

Le caratteristiche delle derivate prime vengono sfruttate tramite l'utilizzo del gradiente, che permette di effettuare lo sharpening non lineare delle immagini. Il **gradiente** di  $f$  alle coordinate  $(x, y)$  è definito come un vettore colonna bidimensionale ed ha l'importante proprietà geometrica di puntare verso la direzione dove è concentrata la più alta percentuale di variazione di  $f$  rispetto al punto  $(x, y)$ .

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

La *magnitudo* (lunghezza o modulo) del vettore è data da:

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

I calcoli di  $g_x$  e  $g_y$  sono operazioni lineari perché implicano l'uso di derivate e possono essere implementati come somma di prodotti utilizzando delle particolari maschere spaziali; invece il calcolo di  $M(x, y)$  implica quadrati e radici quadrate o l'uso di valori assoluti quindi utilizza operazioni non lineari.

La maniera più semplice di generare una approssimazione discreta del gradiente prende in considerazione la differenza mobile (**running difference**) dei pixel nelle due direzioni.

Supporto 3x3:

$\vdots$	$\vdots$	
$z_1$	$z_2$	$z_3$
$\dots$		
$z_4$	$z_5$	$z_6$
$\dots$		
$z_7$	$z_8$	$z_9$
$\vdots$		

Le componenti del gradiente nel pixel  $z_5$  sono  $G_x = z_8 - z_5$  e  $G_y = z_6 - z_5$ , per cui:

$$\nabla f = \left[ (z_8 - z_5)^2 + (z_6 - z_5)^2 \right]^{1/2} \approx |z_8 - z_5| + |z_6 - z_5|$$

Questo ci dà il calcolo del gradiente su una parte del supporto 3x3 (in particolare una porzione 2x2).

In alternativa, si possono usare le differenze incrociate:

$$\nabla f = \left[ (z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{\frac{1}{2}} \approx |z_9 - z_5| + |z_8 - z_6|$$

Una volta calcolato, il gradiente permette di:

- effettuare dello sharpening
- facilitare l'individuazione dei difetti
- facilitare lo sforzo computazionale per eventuali ispezioni automatiche
- evidenziare piccole caratteristiche non facilmente visibili in un'immagine in scala di grigio
- evidenziare piccole discontinuità in aree pressoché piatte

Dunque **ripetiamo** che:

per rilevare i bordi in genere si fa uso del calcolo dei gradienti. La variazione rapida di intensità è un bordo. Calcoliamo ora il gradiente (la velocità di variazione) di questo bordo, cioè la sua derivata prima. Dobbiamo decidere se un bordo venga rilevato quando il gradiente supera un valore di soglia. E' spesso necessaria una leggera sfumatura prima di applicare i filtri di rilevamento dei bordi perché attenua le piccole oscillazioni di colore e previene il rilevamento di falsi bordi. Quindi in uno spazio bidimensionale, al posto della derivata, si ha il gradiente che è un vettore con 2 componenti. Il modulo del gradiente è alla base di parecchi metodi di derivazione di immagini digitali. Ricordiamo che il gradiente (derivata prima) è un vettore bidimensionale che ha una direzione, un verso ed una magnitudo. La direzione è sempre quella di massima variazione, quindi normale al contorno; Il verso è positivo andando dalla regione più scura a quella più chiara; La magnitudo è massima in corrispondenza della massima variazione.

### 3.3.1 Operatori di Roberts

In alternativa si possono usare le differenze incrociate . L'implementazione di queste equazioni può essere effettuata usando le seguenti maschere  $2 \times 2$ , quindi meno oneroso:

-1	0
0	1

0	-1
1	0

I valori assoluti delle risposte delle due maschere vengono sommati per determinare  **$\nabla f$  (gradiente nella posizione  $(x,y)$ )**

Le due maschere sono detti **operatori di Roberts**. Più comune è tuttavia l'impiego di maschere  $3 \times 3$ , che rendono più semplici le operazioni di filtraggio:

0	0	0
0	1	0
0	-1	0

0	0	0
0	1	-1
0	0	0

Running difference

0	0	0
0	1	0
0	0	-1

0	0	0
0	0	1
0	-1	0

Roberts

### 3.3.2 Operatore Isotropico

Si considera che le relazioni di ogni pixel rispetto al centro alla stessa distanza ma questo non è vero poiché i valori dei 4-vicini del pixel centrale hanno distanza 1 mentre quelli nelle posizioni diagonali hanno distanza maggiore di 1 (solitamente 1.41 cioè radical 2). Per cui l'isotropico, cioè quel filtro che vede tutti i vicini alla stessa distanza dal centro pesa il contributo di quelli che si trovano nelle posizioni non diagonali con Radical 2.

Si sfruttano 2 maschere  $M_x$  e  $M_y$

Componenti del gradiente:  $G_x = M_x * I$ ;  $G_y = M_y * I$ .

Il modulo del gradiente è :  $G_m = \sqrt{G_x^2 + G_y^2}$

La fase:  $G_\phi = \arctan(G_y/G_x)$

-1	0	1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

1	$\sqrt{2}$	1
0	0	0
-1	$-\sqrt{2}$	-1



Per evitare i problemi con i calcoli con numeri reali, l'operatore isotropico viene normalmente approssimato:

$$\nabla f \approx |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$

Per evitare i problemi di calcolo con i numeri reali, l'operatore isotropico viene normalmente approssimato con l'**'Operatore Prewitt,e l'Operatore Sobel'**.

### 3.3.3 Operatore Di Prewitt e Sobel

Gli operatori di gradiente sicuramente più diffusi, che utilizzano una maschera 3x3 e non più una 2x2 sono:

Operatori di Prewitt

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

Operatori di Sobel

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

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

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Il funzionamento prevede di sottrarre ai valori della maschera in basso quelli in alto calcolando il valore del gradiente nella direzione delle X. Se intensifichiamo il valore della posizione (x,y) che passa per il centro del kernel (il valore centrale) otteniamo un'altra variante chiamata **Operatore di Sobel**.

Sono filtri passa alto che estraono i contorni dell'oggetto. Esaltano le variazioni significanti dell'intensità lungo gli assi verticale ed orizzontale. Se estratto il bordo calcolassimo il gradiente di un punto su quel bordo questo avrà la direzione **ortagonale alla tangente** del punto. Questi metodi, essendo un'approssimazione del gradiente, sono sensibili alle alte frequenze (brusche variazioni di colore) e mettono in evidenza i punti di variazione mentre il resto tenderà al nero. I metodi di Sobel e di Prewitt forniscono risultati migliori in quanto, utilizzando una maschera di derivazione del terzo ordine anziché del secondo, esaltando i punti di discontinuità appartenenti a linee rispetto a quelli isolati.

## 3.4 Filtro Laplaciano

Il **laplaciano** è l'operatore derivativo di secondo ordine isotropico più semplice da utilizzare. **Isotropico** significa che è invariante per rotazione, cioè ruotare l'immagine e poi applicare il filtro da lo stesso risultato di applicare prima il filtro e poi ruotare il risultato. Il laplaciano aumenta il contrasto nei punti di discontinuità di un'immagine e lascia in secondo piano le zone con livelli di intensità costante o poco variabile. Ciò produce immagini con linee grigastre e altre discontinuità, su uno sfondo scuro. Lo sfondo può essere recuperato sommando (o sottraendo) l'immagine originale all'immagine laplaciana.

Il laplaciano è definito come:

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

Il modo più semplice di definire il laplaciano nel caso discreto consiste nel calcolare le derivate seconde rispetto alle due direzioni, e sommare i risultati.

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

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

Sommando le due componenti:

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

Il laplaciano si può quindi implementare con la seguente maschera, nota come laplaciano dei 4-vicini:

$\nabla^2 f$																													
$\frac{\partial^2 f}{\partial x^2}$	$\frac{\partial^2 f}{\partial y^2}$	$\rightarrow$																											
<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>+1</td><td>0</td></tr> <tr><td>0</td><td>-2</td><td>0</td></tr> <tr><td>0</td><td>+1</td><td>0</td></tr> </table>	0	+1	0	0	-2	0	0	+1	0	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>+1</td><td>-2</td><td>+1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	+1	-2	+1	0	0	0	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>+1</td><td>0</td></tr> <tr><td>+1</td><td>-4</td><td>+1</td></tr> <tr><td>0</td><td>+1</td><td>0</td></tr> </table>	0	+1	0	+1	-4	+1	0	+1	0
0	+1	0																											
0	-2	0																											
0	+1	0																											
0	0	0																											
+1	-2	+1																											
0	0	0																											
0	+1	0																											
+1	-4	+1																											
0	+1	0																											

le immagini filtrate con il laplaciano perdono completamente l'informazione di sfondo: questo risultato è ora giustificato pienamente dal comportamento della derivata seconda. Il fatto che l'usita del laplaciano presenti valori diversi da 0 solo in corrispondenza dei dettagli evidenziati suggerisce un metodo per il ripristino (parziale) dell'informazione di sfondo: sommare all'uscita del laplaciano l'immagine originale.

Si ha pertanto:

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{se il coefficiente centrale è } < 0 \\ f(x, y) + \nabla^2 f(x, y) & \text{se il coefficiente centrale è } > 0 \end{cases}$$

Sostituendo l'espressione precedentemente derivata per il laplaciano:

$$\begin{aligned} g(x, y) &= f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] + 4f(x, y) \\ &= 5f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] \end{aligned}$$

Questa equazione può essere implementata utilizzando la maschera:

$\begin{array}{ c c c } \hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$	Mentre tenendo conto dei 4-diagonali si otterrebbe: $\begin{array}{ c c c } \hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$
-------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------

In definitiva:

Il Laplaciano evidenzia le variazioni di intensità che circonda un pixel. Il filtro estrae il contorno degli oggetti ed esalta i dettagli. Questo è omnidirezionale.

Kernel

<b>a</b>	<b>d</b>	<b>c</b>
<b>b</b>	<b>x</b>	<b>b</b>
<b>c</b>	<b>d</b>	<b>a</b>

- Se  $x = 2(|a| + |b| + |c| + |d|)$  estrae i contorni
- Se  $x \geq 2(|a| + |b| + |c| + |d|)$  evidenzia i contorni

### 3.5 Filtro high-boost [Enfasi di alta Frequenza]

Il filtro high-boost ha l'obiettivo di intensificare le componenti a basse frequenze lasciate fuori dal filtraggio passa-alto. Se guardiamo il filtro passa alto come una semplice sottrazione di componenti a bassa frequenza dall'immagine originale ( $HP=O-LP$ ), questo solo nel caso IDEALE, possiamo ottenere un filtro High Boost intensificando il valore di intensità luminosa originale che abbiamo per un parametro A che non è altro che un fattore di amplificazione delle componenti sia in bassa che alta frequenza.

Una immagine filtrata in modo passa-alto ( $HP$ ) può essere pensata come differenza tra l'immagine originale ( $O$ ) e una versione passa-basso dell'immagine stessa ( $LP$ ) ;

$$HP = O - LP$$

Denotando con A un fattore di amplificazione, la definizione del filtro di high-boost è la seguente:

$$\text{High boost} = A \times O - LP$$

Ora sottraendo entrambi i membri per  $HP$ :

$$\text{High Boost} - HP = A \times O - LP - HP$$

Sostituendo  $HP$

$$\text{High Boost} - HP = A \times O - LP - (O - LP)$$

Cambiando i segni

$$\text{High Boost} - HP = A \times O - LP + O - LP$$

$LP$  si semplifica:

$$\text{High Boost} - HP = A \times O - O$$

Messa in evidenza di  $O$

$$\text{High Boost} - HP = (A - 1) \times O$$

Porto  $HP$  dall'altra parte

$$\text{High Boost} = (A - 1) \times O + HP$$

Questo significa che l'High Boost aggiunge al risultato del filtro  $HP$  (passa alto) un'altra componente che non è altro che la parte dell'immagine originale amplificata per  $A - 1$ . Per ottenere questo basta cambiare la maschera inserendo nella posizione centrale un valore W che diventa 8 quando  $A=1$  (cioè non vi è alcuna amplificazione) mentre quando assume il valore di  $9*A - 1$ , cioè abbiamo scelto quel valore di W tale per cui se  $A=1$  il risultato dell'operazione da 8.

Ricapitolando:

Se  $A=1$ , si ottiene il filtro passa-alto classico.(nessuna amplificazione)

Se  $A > 1$ , parte dell'immagine originale viene nuovamente aggiunta al risultato del filtraggio passa-alto, ripristinando (almeno parzialmente) le componenti di bassa frequenza eliminate dalla operazione di filtraggio. Ne risulta una immagine più simile a quella originale, con un grado di edge crispening che dipende dal valore del fattore di amplificazione  $A$ .

L'implementazione del filtro di high-boost mediante una maschera è la seguente:

-1	-1	-1
-1	w	-1
-1	-1	-1

dove  $w = 9A - 1$ , con  $A \geq 1$

Con valori di  $W > 8$  l'immagine porta con sé una certa percentuale di basse frequenze che permettono di distinguere meglio gli oggetti.

# 4. Convoluzione, Correlazione e Campionamento

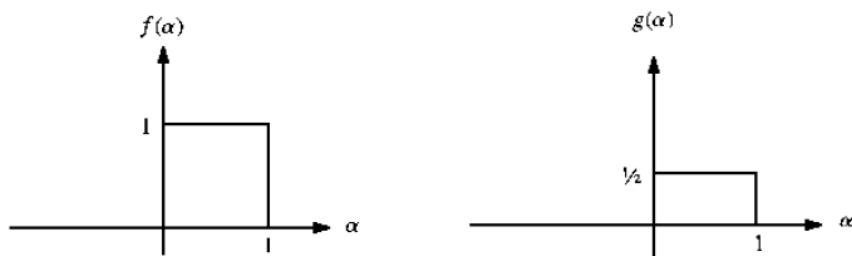
## 4.1 Teorema della Convoluzione

Date due funzioni continue  $f(x)$  e  $g(x)$  la loro **convoluzione** è definita da:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha)g(x-\alpha)d\alpha$$

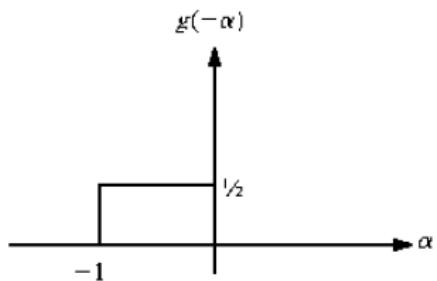
Dove alpha è una variabile di comodo per l'integrazione. In pratica teniamo ferma una funzione ( $f(x)$ ), trasliamo l'altra funzione di una certa quantità e ogni volta che effettuiamo questo procedimento effettuiamo il prodotto delle due funzioni e ne calcoliamo l'area. La funzione che trasla è detta **maschera**.

Esempio, due funzioni boxcar(a scatola):

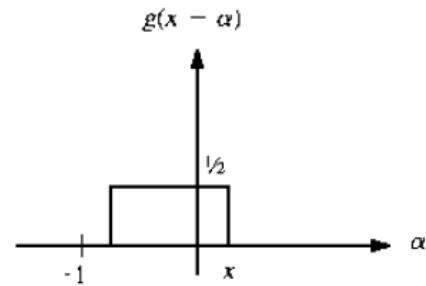


Quindi per trovare  $g(x-\alpha)$  per prima cosa bisogna ribaltare la  $g(\alpha)$  rispetto all'origine in modo da ottenere  $g(-\alpha)$  e quindi traslare la quantità ottenuta aggiungendo  $x$ :

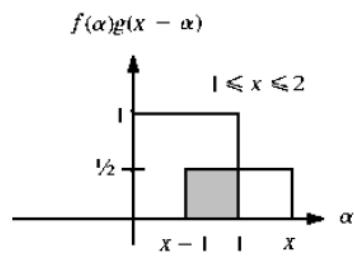
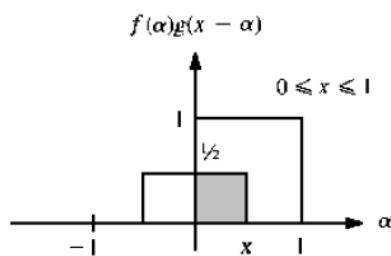
Ribalto  $g(\alpha)$ :



Aggiungo  $x$ , ottenendo una traslazione:



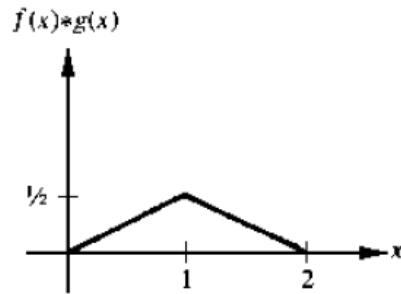
Dopo aver ottenuto  $g(x-\alpha)$  possiamo calcolare la convoluzione. Per calcolarla dobbiamo tenere ferma la  $f(\alpha)$  e far traslare la  $g(x-\alpha)$  di una quantità  $x$ :



Quindi la  $g(x-\alpha)$  sta traslando mentre la  $f(x)$  è ferma, il prodotto delle due funzioni è l'area in grigio (base per altezza in quanto sono rettangolini). Il valore massimo lo si ottiene quando la  $g(x-\alpha)$  sarà pienamente all'interno della  $f(x)$ , cioè quando  $x = 1$ , dopodiché inizia a decrescere perché lo shift va avanti e quindi la  $g(x-\alpha)$  esce dalla  $f(x)$ .

Riassumendo, si ha:

$$f(x) * g(x) = \begin{cases} x/2 & 0 \leq x \leq 1 \\ 1-x/2 & 1 \leq x \leq 2 \\ 0 & \text{altrimenti} \end{cases}$$



Questo ci dice che il valore massimo lo otteniamo quando la g è completamente nella f.

La caratteristica molto importante della convoluzione è che:

Fare la convoluzione nel dominio spaziale corrisponde ad un abbattimento della complessità nel dominio delle frequenze. Poiché nel dominio spaziale una convoluzione che ha nel caso discreto una complessità di O(MN) dobbiamo effettuare per ogni punto dobbiamo effettuare la sommatoria dei prodotti quindi MN volte, nel dominio delle frequenze la complessità diviene O(1) quindi costante perché basta farne il prodotto puntuale degli spetti di fourier.

Questo risultato costituisce il vantaggio nell'andare nel dominio delle frequenze. Ricordiamo che il costo per passare nel dominio delle frequenze con Fourier è O(nlogn) cioè la complessità della FFT.

Le due relazioni che costituiscono il teorema della convoluzione sono:

$$f(x) * g(x) \Leftrightarrow F(u)G(u)$$

$$f(x)g(x) \Leftrightarrow F(u)*G(u)$$

Cioè la convoluzione di due funzioni f(x) e g(x) ha come trasformata di fourier il prodotto delle trasformate delle due funzioni. E viceversa.

#### 4.1.1 Teorema della Convoluzione 2-D

Nel caso 2D la convoluzione di due funzioni continue f(x,y) e g(x,y) è definita come:

$$f(x, y) * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)g(x-\alpha, y-\beta)d\alpha d\beta$$

Il teorema della convoluzione 2D è espresso dalle seguenti due relazioni:

$$f(x, y) * g(x, y) \Leftrightarrow F(u, v)G(u, v)$$

$$f(x, y)g(x, y) \Leftrightarrow F(u, v)*G(u, v)$$

Che come nel caso 1D definiscono che la convoluzione di due funzioni continue f(x,y) e g(x,y) ha come trasformata di fourier il prodotto delle trasformate delle due funzioni di partenza e viceversa.

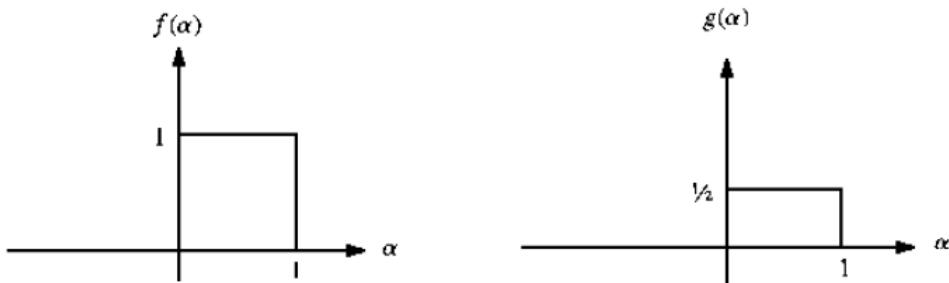
## 4.2 Teorema della Correlazione

Insieme con la convoluzione, costituisce il legame essenziale tra il dominio spaziale e il dominio della frequenza. Date due funzioni continue 1-D,  $f(x)$  e  $g(x)$ , la loro correlazione è definita da:

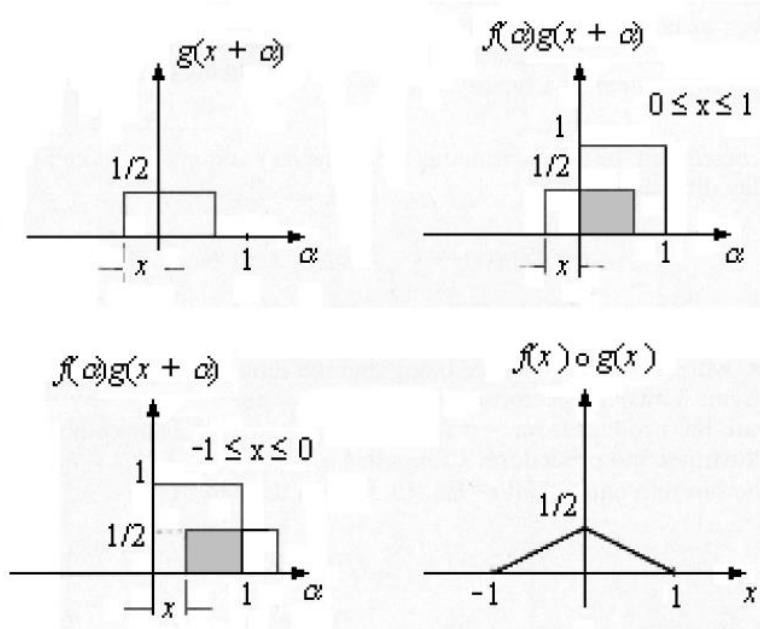
$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f^*(\alpha)g(x+\alpha)d\alpha$$

Dove  $\alpha$  è una variabile di comodo per l'integrazione e  $f^*$  è la complessa coniugata di  $f$ .

La differenza con la convoluzione è che la  $g(x)$  non è ribaltata attorno all'origine. Pertanto il calcolo della correlazione richiede, per ciascun valore di  $x$ , solo lo scorrimento di  $g(x)$  rispetto a  $f(x)$  e l'integrazione del prodotto così ottenuto (area grigia nelle figure seguenti):



Bisogna costruire la  $g(x+\alpha)$  ma dato che non vi è ribaltamento per ottenerla basta sommare all'argomento la  $x$ . Dopodiché ci accorgiamo che il valore massimo si ha quando la  $x$  assume valore 0 e di conseguenza la  $g(x)$  è interamente compresa nella  $f(x)$  quindi quando si ha un **matching** perfetto.



Di fatti la correlazione viene utilizzata quando si vuole effettuare **un template matching**, cioè ricercare una sottoimmagine in un'immagine di riferimento in modo esaustivo.

Il teorema della correlazione 2D è espresso dalle relazioni:

$$\begin{aligned} f(x, y) \circ g(x, y) &\Leftrightarrow F^*(u, v)G(u, v) \\ f^*(x, y)g(x, y) &\Leftrightarrow F(u, v) \circ G(u, v) \end{aligned}$$

Quindi fare la correlazione di due funzioni nel dominio spaziale corrisponde ad effettuare il prodotto dello spettro di una funzione per la trasformata della seconda e viceversa.

Questo ci dice che invece di effettuare il matching punto punto in modo esaustivo , possiamo effettuare le trasformate di entrambe le funzioni, effettuarne il prodotto punto punto e laddove otteniamo il valore massimo significa che quel punto è il punto di correlazione massima nel dominio delle frequenze. Effettuiamo, a questo punto, l'antitrasformata e otteniamo la posizione spaziale in cui abbiamo la correlazione massima.

### **Correlazione Continua 2-D**

$$f(x, y) \circ g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f^*(\alpha, \beta) g(x + \alpha, y + \beta) d\alpha d\beta$$

### **Correlazione Discreta 1-D**

La formulazione della correlazione discreta assume ancora che  $f(x)$  e  $g(x)$  siano estese e periodiche con un certo periodo  $M$ , di valore opportuno:

$$f_e(x) \circ g_e(x) = \sum_{m=0}^{M-1} f_e^*(m) g_e(x+m), \quad \text{per } x = 0, 1, \dots, M-1$$

La  $f_e^*$  equivale al complesso coniugato, questo perchp la correlazione può essere applicata a funzioni a variabili reali ma anche a funzioni complesse, nel nostro caso poiché non lo applichiamo mai a funzioni complesse questo \* non lo consideriamo.

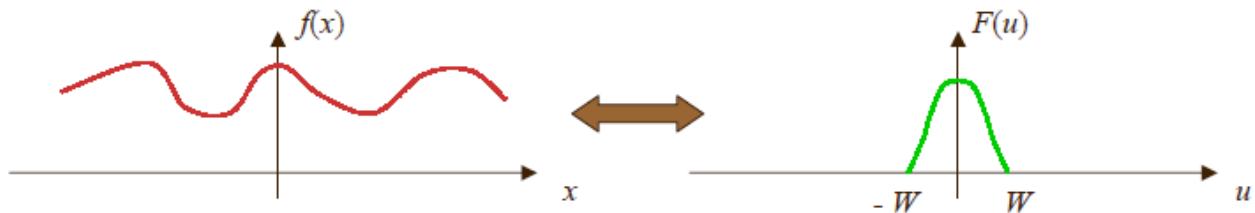
### **Correlazione Discreta 2-D**

$$f_e(x, y) \circ g_e(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e^*(m, n) g_e(x+m, y+n)$$

## 4.3. Teorema del Campionamento

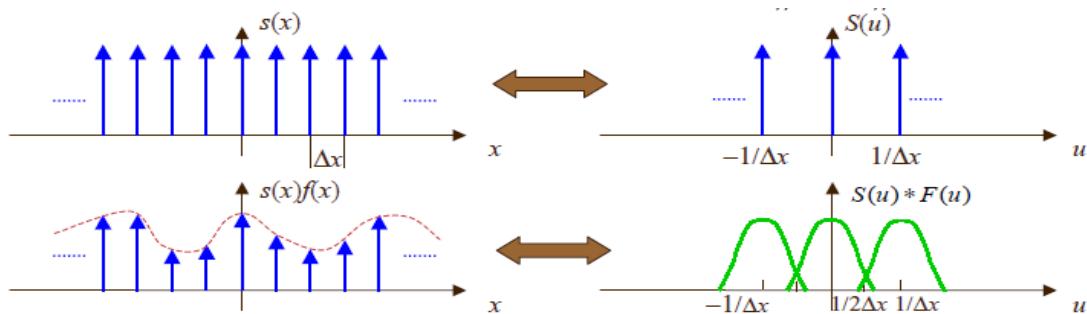
Il campionamento fa parte della fase di acquisizione dell'immagine, quando vogliamo digitalizzare il segnale analogico in ingresso. La digitalizzazione di una immagine continua comporta ovviamente una enorme perdita di informazione. E' quindi importante analizzare più a fondo il processo di campionamento, per determinare in quali condizioni una immagine continua possa essere interamente ricostruita (cioè senza perdita di informazione) a partire da un insieme di valori campionati.

Consideriamo una funzione  $f(x)$  continua:



La funzione presa in esame è una funzione continua che si estende da  $-\infty$  a  $+\infty$  e la corrispondente trasformata di Fourier è una funzione non nulla nell'intervallo  $-W$  a  $W$ .

Il campionamento può essere effettuato moltiplicando la  $f(x)$  per **una funzione di campionamento  $s(x)$** , cioè un treno di impulsi (Delta di dirac) distanziati di  $\Delta x$ , la cui trasformata è ancora un treno di impulsi, distanziati di  $1/\Delta x$ :



Dal teorema della convoluzione, il prodotto nel dominio spaziale corrisponde alla convoluzione nel dominio della frequenza.

### Delta di dirac

la delta di dirac è quella funzione  $\delta(x-x_0)$  che assume valore 1 per  $x = x_0$  e 0 per  $x \neq x_0$ .

Cioè:

$$\int_{-\infty}^{\infty} \delta(x - x_0) dx = \int_{x_0^-}^{x_0^+} \delta(x - x_0) dx = 1$$

Quindi :

$$\int_{-\infty}^{\infty} f(x) \delta(x - x_0) dx = f(x_0)$$

Cioè effettuando l'integrale del prodotto della funzione per la delta di dirac questo ci restituirà il valore della funzione nel punto dove la delta di dirac assume valore non nullo.

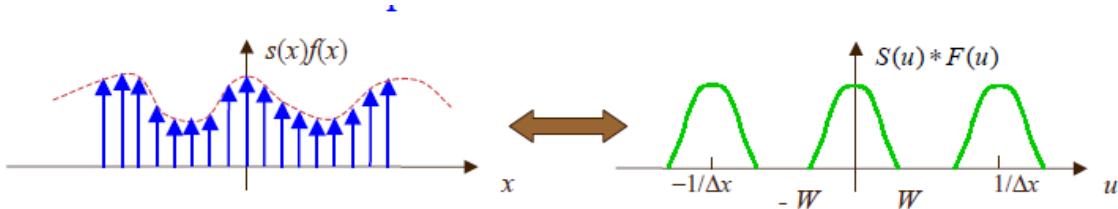
La convoluzione della  $F(u)$  con un treno di impulsi non è altro che la replica della  $F(u)$  centrata nel punto in cui il treno di impulsi è non nullo. Come si vede, essa risulta periodica, con periodo pari a  $1/\Delta x$ , e i singoli periodi possono parzialmente sovrapporsi, in **dipendenza del valore di  $\Delta x$** . Per evitare questo  $W$  deve essere  $< 1/\Delta x$ , in modo tale che l'antitrasformata della convoluzione  $S(u)F(u)$  restituirà fedelmente il prodotto  $s(x)f(x)$ , mentre invece se si sovrappongono abbiamo **una perdita di informazioni** sulla parte finale (si nota seguendo la curva verde, cioè la parte di curva che tocca l'asse  $x$  viene persa perché si incicia con l'inizio dell'altra curva). Questa perdita nel dominio spaziale si riscontra con la ricostruzione della  $f(x)$  **NON fedele**. Per evitare la sovrapposizione dei periodi della convoluzione, deve rispettare la condizione (teorema del campionamento di Whittaker-Shannon):

$$W \leq \frac{1}{2\Delta x}, \text{ cioè } \Delta x \leq \frac{1}{2W}$$

$W$  deve essere  $\leq$  a quella quantità perché  $1/\Delta x$  è la distanza che intercorre tra il punto centrale della curva verde (trasformata della delta di dirac in un punto  $x_0$  non nullo e il successivo), quindi per avere la separazione netta  $W$  deve rispettare quel vincolo. Diminuendo  $\Delta x$  si ottiene infatti una separazione dei periodi che impedisce la sovrapposizione. Se otteniamo l'uguaglianza, cioè  $W$  è proprio uguale a quella quantità abbiamo il numero di campioni necessari ad avere una ricostruzione fedele e quindi otteniamo la **frequenza di Nyquist**.

**Def.** Una funzione continua può essere completamente ricostruita a partire dall'insieme dei suoi campioni, se questi ultimi sono stati acquisiti con un valore pari o superiore al doppio della frequenza più alta della funzione; un tasso di campionamento *esattamente* uguale al doppio della frequenza più alta è chiamato **frequenza di Nyquist**.

Una volta determinata la frequenza giusta per la quale i periodi non risultano sovrapposti, bisogna isolare un singolo periodo.



Per fare ciò dobbiamo costruirci una funzione ad hoc  $H(u)$  definita come segue

$$H(u) = \begin{cases} 1 & -w \leq u \leq w \\ 0 & \text{altrimenti} \end{cases}$$

Quindi costruiamo la nostra funzione boxcar cercando di isolare un solo periodo della convoluzione di  $S(x)*F(u)$  effettuandone il prodotto. Successivamente applicando l'antitrasformata al periodo è possibile risalire alla funzione  $f(x)$  senza nessuna perdita di informazione.

Nel caso reale, però, la  $f(x)$  presa in esame risulta essere finita quindi la corrispettiva trasformata di Fourier si estende con un periodo infinito. Per questo motivo andando a cercare di isolare il periodo indipendentemente dal valore della frequenza scelta si produrrà un errore detto **Aliasing** inversamente proporzionale alla dimensione della finestra boxcar definita.

**DEF:** *L'aliasing è il processo in cui le componenti ad alta frequenza di una funzione continua si "mascherano" alle frequenze più basse nella funzione campionata. Alias=falsa identità.*

Nella pratica gli effetti di aliasing possono essere *ridotti* dallo smoothing delle funzioni di input, che ne attenua le frequenze più alte. Questo processo chiamato *anti-aliasing* deve essere effettuato prima che la funzione venga campionata, perché l'aliasing è una fase del campionamento che non può essere "eliminato dopo". Per ridurre il fenomeno dell'aliasing possiamo utilizzare due metodi:

- **sfocare** leggermente l'immagine prima di rimpicciolirla
- **sovracampionare** (utilizzare una risoluzione maggiore) la scena originale e ridurre (ricampionando) le sue dimensioni attraverso la cancellazione di righe e colonne. Ovviamente con questa tecnica è necessario avere accesso alla scena originale.

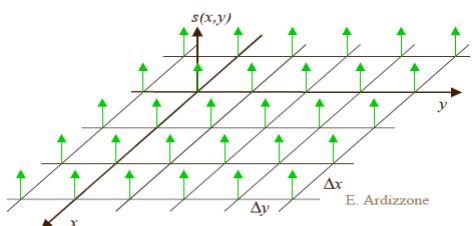
Di conseguenza, anche se il teorema del campionamento è soddisfatto, la ricostruzione esatta di una funzione campionata in un intervallo finito **non è in generale possibile**. La distorsione è tanto più severa quanto meno estesa è la finestra di campionamento. La trasformata di Fourier della funzione originale può essere completamente isolata solo se  $f(x)$  è a **banda limitata e periodica**, con un periodo uguale a  $X$ .

### Campionamento 2D

Gli stessi concetti utilizzati nel caso 1-D possono essere applicati a funzioni 2-D. Il processo di campionamento può essere formulato matematicamente usando la funzione impulsiva 2-D  $\delta(x,y)$ , definita come:

$$\int \int_{-\infty}^{\infty} f(x,y) \delta(x-x_0, y-y_0) dx dy = f(x_0, y_0)$$

Una funzione di campionamento 2-D consiste di un treno di impulsi, distanti  $\Delta x$  lungo  $x$  e  $\Delta y$  lungo  $y$ :



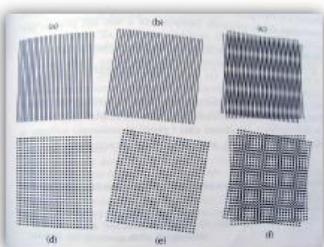
Una funzione continua  $f(x,y)$  può essere campionata formando il prodotto  $s(x,y)f(x,y)$ . Viene effettuata una grigliatura dell'immagine che è detta **Pixerizzazione**, cioè il passaggio da un'immagine analogica ad una digitale campionata, effettuando un'integrazione nell'area determinata dal Deltax e Deltay. Per determinare il Deltax e Deltay si ricorre nuovamente al teorema di Niyquist Shannon:

$$\Delta x \leq \frac{1}{2W_u} \quad \Delta y \leq \frac{1}{2W_v}$$

### Effetto Moirè

Questo tipo di artefatto è visibile nel campionamento di scene con componenti periodiche o quasi. Nell'elaborazione delle immagini questo problema compare nella scansione di stampe multimediali, come i giornali e le riviste, o nelle immagini con componenti periodiche la cui distanza è confrontabile con lo spazio di separazione tra i campioni.

Vediamo un esempio chiarificatore:



Individualmente le singole componenti sono chiare e prive di interferenze, ma la sovrapposizione di un disegno sull'altro crea una struttura periodica delle frequenze completamente assente nelle strutture originali

Le immagini (a) e (b) sono delle linee verticali, ma siccome l'immagine è stata importata dalla digitale si sono creati proprio gli artefatti che stiamo descrivendo.

I giornali e gli altri supporti stampati fanno uso dei cosiddetti *punti a mezzotono (halftone)*, che sono punti o ellissi neri le cui dimensioni variano a seconda degli schemi utilizzati per simulare i toni di grigio. Importante ricordare che la dimensione del punto è inversamente proporzionale all'intensità dell'immagine; nelle aree chiare infatti i punti sono piccoli o totalmente assenti, nelle aree grigie non troppo scure i punti sono larghi. Solitamente i giornali sono stampati utilizzando 75 *dpi* (*dots per inch*), le riviste utilizzano 133 dpi e le brochure ad alta qualità 175 dpi.

## 5. Le basi per il filtraggio nel dominio della frequenza

Le tecniche di filtraggio nel dominio della frequenza si basano sulla modifica della trasformata di Fourier per raggiungere gli obiettivi prefissati e poi, calcolando l'antitrasformata discreta, tornare al dominio dell'immagine.

### 5.1 Filtraggio nel dominio della frequenza

Data un'immagine digitale  $f(x, y)$ , di dimensioni  $M \times N$  l'equazione del filtraggio è la seguente:

$$g(x, y) = \mathfrak{J}^{-1}[H(u, v)F(u, v)]$$

Dove

- $\mathfrak{J}^{-1}$  è l'antitrasformata di Fourier;
- $F(u, v)$  è la trasformata dell'immagine di input  $f(x, y)$ ;
- $H(u, v)$  è la funzione filtro, che modifica la trasformata dell'immagine di input per ottenere un output processato;
- $g(x, y)$  è ovviamente l'immagine filtrata.
- Le funzioni  $F$ ,  $H$  e  $g$  sono matrici di dimensioni  $M \times N$ . Da ricordare che si utilizzano funzioni simmetriche rispetto al loro centro, che fanno in modo che  $F(u, v)$  sia centrata a sua volta.

data una immagine  $f(x,y)$  e calcolata la sua trasformata  $F(u,v)$ , l'**obiettivo di una elaborazione nel dominio della frequenza è scegliere una  $H(u,v)$  tale che l'immagine di uscita  $g(x,y)$  esibisca le proprietà desiderate.**

Solitamente conviene introdurre dello **zero-padding** (cioè aggiungere degli zeri allo spettro originale) a un'immagine per evitare di ottenere risultati erronei. Il metodo migliore per adoperare il padding di un filtro nel dominio della frequenza potrebbe essere quello di

1. **creare un filtro** (delle stesse dimensioni dell'immagine);
2. **calcolare la sua IDFT** per ottenere il filtro spaziale corrispondente;
3. **eseguire il padding** del filtro nel dominio spaziale
4. **calcolare la DFT** per tornare nel dominio della frequenza.

Questo metodo però non va bene, perché le discontinuità nel filtro spaziale creano un **ringing** (risonanza) nella sua controparte del dominio della frequenza.

Perciò il metodo da utilizzare è quello di applicare lo zero padding alle immagini e successivamente creare i filtri nel dominio della frequenza delle stesse dimensioni delle immagini padded (utilizzando la DFT filtri e immagini devono avere le stesse dimensioni).

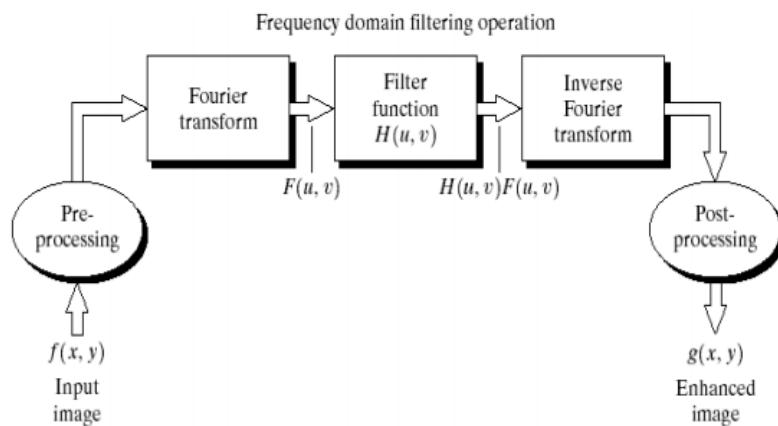
**Riassumiamo i passi da seguire per effettuare il filtraggio nel dominio della frequenza:**

1. Data un'immagine di input  $f(x, y)$  di dimensioni  $M \times N$ , si ricavano i parametri di padding  $P$  e  $Q$ , solitamente  $P = 2M$  e  $Q = 2N$ .

2. Si forma un'immagine padded  $f_p(x, y)$  di dimensioni  $P \times Q$ , estendendo  $f(x, y)$  con il necessario numero di zeri
3. Si moltiplica  $f_p(x, y)$  per  $(-1)^{x+y}$  per centrarne la trasformata
4. Si calcola la DFT  $F(u, v)$  dell'immagine del punto 3
5. Si genera una funzione filtro simmetrica e reale  $H(u, v)$ , di dimensioni  $P \times Q$  con il centro nelle coordinate  $(P/2, Q/2)$
6. Si ottiene l'immagine:

$$g_p(x, y) = \{\text{real}[\mathcal{F}^{-1}[G(u, v)]]\}(-1)^{x+y}$$

7. Il risultato finale  $g(x, y)$  è ottenuto attraverso l'estrazione della regione  $M \times N$  dal quadrante in alto a sinistra di  $g_p(x, y)$



## 5.2 Corrispondenza tra il filtraggio nel dominio spaziale e quello nel dominio della frequenza

Dato un filtro spaziale, otteniamo la sua rappresentazione nel dominio della frequenza prendendo la trasformata di Fourier del filtro spaziale. I due filtri infatti formano una coppia di Fourier:

$$h(x, y) \Leftrightarrow H(u, v)$$

dove  $h(x, y)$  è un filtro spaziale, che viene chiamato anche **risposta finita all'impulso di  $H(u, v)$** , in quanto questo filtro può essere ottenuto dalla risposta di un filtro nel dominio della frequenza ad un impulso. Un modo per trarre vantaggio dalla proprietà di entrambi i domini è quello di specificare un filtro nel dominio della frequenza, calcolare la sua IDFT, e poi utilizzarne il filtro spaziale risultante come una *guida* per la costruzione di maschere filtro spaziali più piccole. Il dominio della frequenza può essere considerato come un **laboratorio** nel quale approfittare dei vantaggi della corrispondenza tra il contenuto della frequenza e l'aspetto dell'immagine. In molti casi infatti gli obiettivi non sono raggiungibili nel dominio spaziale, mentre diventano banali nel dominio della frequenza. Una volta individuato un filtro specifico attraverso la sperimentazione nel dominio della frequenza, l'implementazione reale del metodo viene svolta nel dominio spaziale.

## 5.3 Smoothing utilizzando i filtri nel dominio della frequenza

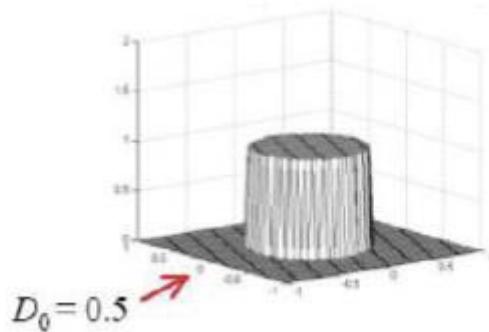
I filtri di smoothing, come già visto, servono ad eliminare le brusche transizioni ed il rumore; questo corrisponde nel dominio della frequenza ad attenuare le componenti ad alta frequenza. Esistono molti tipi di lowpass filter: i principali sono quello **ideale**, quello di **Butterworth** e quello **Gaussiano**.

### 5.3.1 Filtri passa-basso ideali

Il filtro passa-basso ideale è un filtro low pass 2-D che lascia passare, senza attenuarle, tutte le frequenze all'interno di un cerchio di raggio  $D_0$ , centrato sull'origine, e "taglia fuori" (annulla) tutte le frequenze al di fuori di questo cerchio.

La funzione che lo caratterizza è la seguente:

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



$D_0$  è la **frequenza di taglio** del filtro (una quantità non negativa)

$D(u, v)$  è la distanza del punto  $(u, v)$  dall'origine del piano delle frequenze:

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

I filtri ideali causano un forte fenomeno di sfocatura ad anello (**ringing**).

Ma perché "Ideale"?

Il termine ideale indica che tutte le frequenze all'interno della circonferenza di raggio  $D_0$  rimangono inalterate senza alcun tipo di attenuazione, mentre tutte le frequenze fuori dal cerchio vengono annullate.

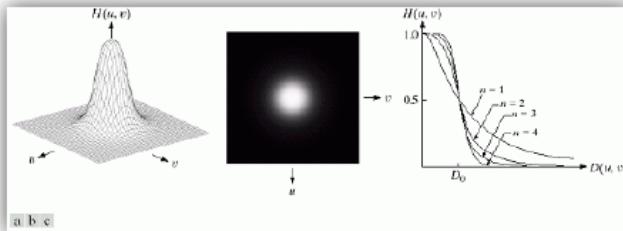
**Effetto ringing (sfocatura ad anello)**

L'origine dell'effetto di ringing è legato alla funzione  $h(x, y)$  antitrasformata del filtro lowpass ideale (ILPF). La componente centrale è responsabile del blurring quelle negative e poi positive concentriche generano brusche transizioni nella immagine. Il ringing è tanto più accentuato quanto minore è la frequenza di taglio, mentre al crescere della frequenza di taglio le transizioni diventano via via meno accentuate, cosicché il ringing diminuisce.

### 5.3.2 Filtri passa-basso di Butterworth

La funzione di trasferimento di un filtro low pass di Butterworth di ordine  $n$ , con frequenza di taglio posta a distanza  $D_0$  dall'origine, è definita:

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



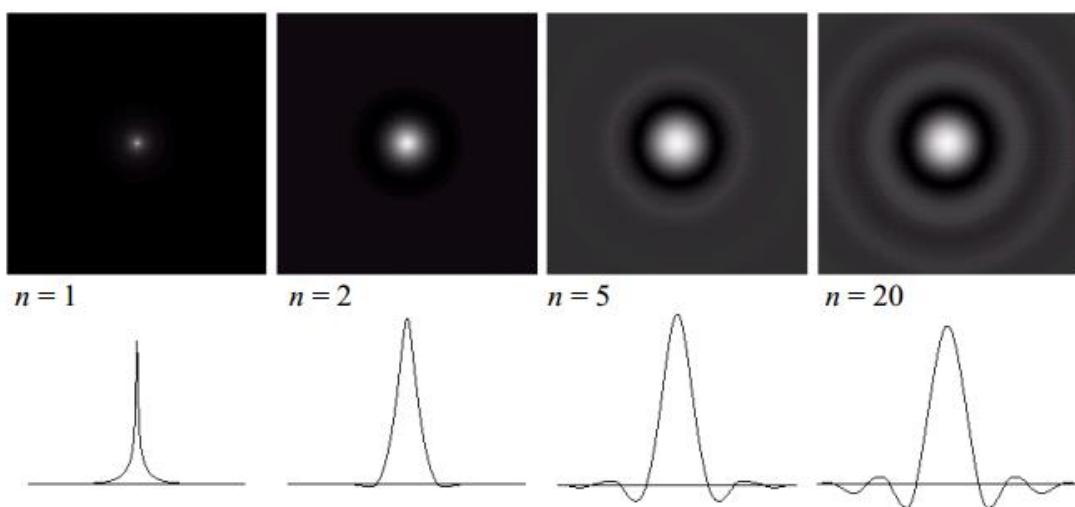
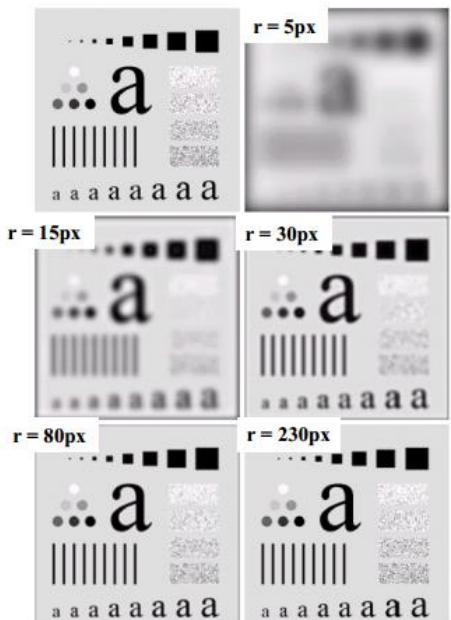
- (a) Diagramma prospettico della funzione di trasferimento di un filtro lowpass di Butterworth
- (b) Filtro visualizzato come immagine
- (c) Sezioni trasversali radiali del filtro di ordine 1-4

La funzione di trasferimento del filtro low pass di Butterworth è graduale, in quanto non ha una discontinuità marcata che indica un taglio netto tra le frequenze filtrate e quelle che vengono lasciate passare. In questo caso è possibile definire la posizione della frequenza di taglio come il luogo dei punti in cui  $H(u, v)$  è minore di una determinata frazione del suo valore massimo; il 50% del valore massimo si ha per  $D(u, v) = D_0$ .

Esempio:

Specifichiamo però che la presenza del ringing dipende dall'ordine utilizzato, infatti un filtro di ordine 1 non presenta ringing nel dominio spaziale; di ordine 2 è impercettibile, quindi risulta essere una soluzione ottimale per avere un buon compromesso tra l'effettivo filtraggio e la presenza di ringing; il fenomeno diventa significativo negli ordini superiori, infatti un filtro low pass di Butterworth di ordine 20 porta a dei risultati simili a quelli del filtro low pass ideale.

Le immagini illustrano quattro diversi ordini:



### 5.3.3 Filtri passa-basso Gaussiani

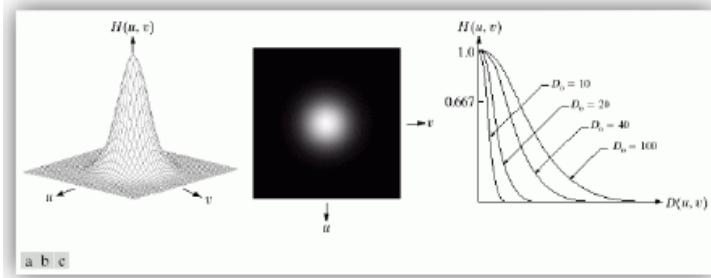
I filtri low pass gaussiani GLPF sono definiti dalla seguente espressione:

$$H(u, v) = e^{\frac{-D^2(u, v)}{2D_0^2}}$$

Dove  $D_0$  è la frequenza di taglio;

I filtri gaussiani hanno il grande vantaggio di avere come trasformata di Fourier ancora una gaussiana quindi il filtro spaziale gaussiano, ottenuto dal calcolo della IDFT, **non avrà ringing**.

Vediamo la solita rappresentazione grafica:



- (a) Diagramma prospettico della funzione di trasferimento del filtro low pass gaussiano
- (b) Filtro visualizzato come immagine
- (c) Sezioni trasversali radiali dei differenti valori di  $D_0$

Vediamo come si comporta il filtro gaussiano con l'esempio utilizzato per i precedenti due filtri:  
notiamo che la transizione graduale nella sfocatura è legata all'incremento della frequenza di taglio. Il filtro gaussiano ha come risultato un minore livello di smoothing rispetto al filtro low pass di Butterworth di ordine 2 sebbene la frequenza di taglio sia uguale. Siamo comunque certi dell'assenza di ringing.

(a) Immagine originale  
(b) – (f) risultati del filtraggio effettuato con il filtro low pass gaussiano, con le stesse frequenze di taglio dell'esempio del filtro ideale



## 5.4 Sharpening utilizzando i filtri nel dominio della frequenza

Lo sharpening di un'immagine può essere effettuato nel dominio della frequenza attraverso il filtraggio passa alto, che attenua le componenti a bassa frequenza e lascia passare le frequenze più alte nella trasformata di Fourier.

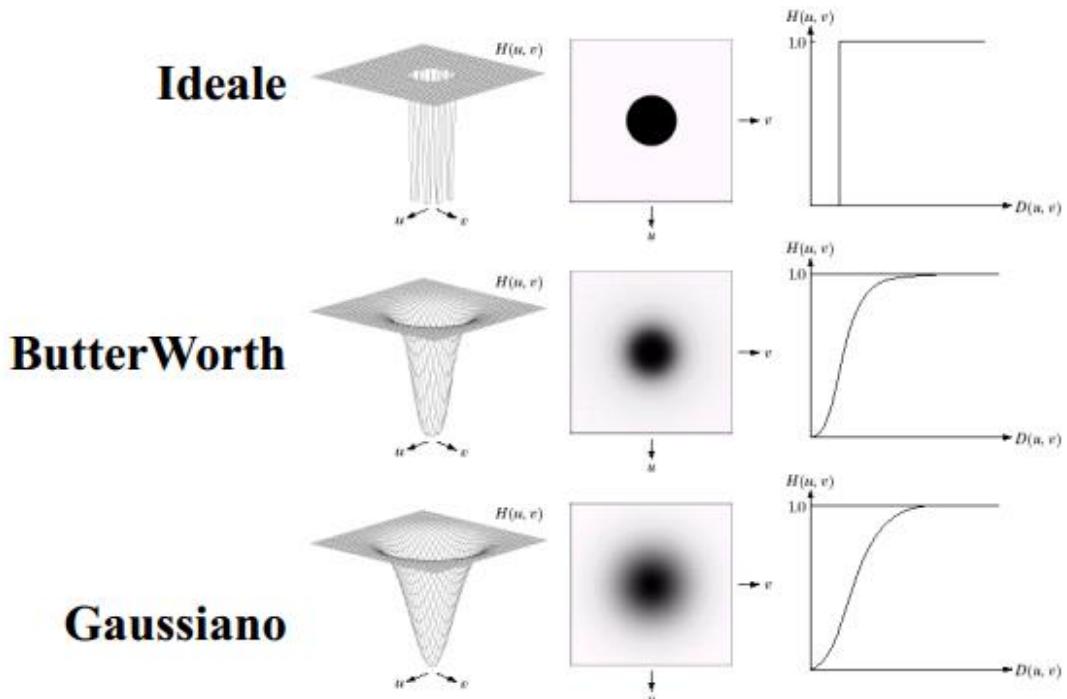
Un filtro high pass si ottiene a partire da un filtro low pass utilizzando l'equazione:

$$H_{HP}(u, v) = 1 - H_{LP}(u, v)$$

Dove  $H_{LP}(u, v)$  è la funzione di trasferimento del filtro low pass. Quando il filtro low pass attenua le frequenze, il filtro high pass le lascia passare e viceversa.

Anche nel caso dello sharpening il filtro di Butterworth può essere considerato una via di mezzo tra il filtro ideale e quello gaussiano.

Vediamo le caratteristiche generali dei tre filtri (ideale; Butterworth; gaussiano) con queste due immagini:



Ora vediamoli più in dettaglio.

### 5.4.1 Filtri passa-alto ideali

Un filtro high pass ideale 2-D IHPL è definito:

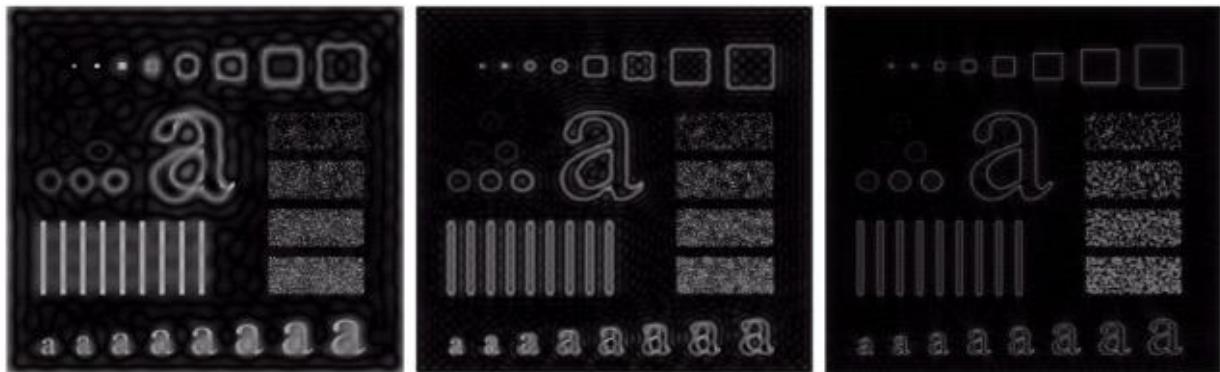
$$H(u, v) = \begin{cases} 0 & \text{se } D(u, v) \leq D_0 \\ 1 & \text{se } D(u, v) > D_0 \end{cases}$$

dove  $D_0$  è la **frequenza di taglio** e dove  $D(u, v)$  è la distanza dall'origine ed è data da :

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

Il filtro high pass porta a zero tutte le frequenze all'interno del cerchio di raggio  $D_0$  e lascia passare, senza attenuarle, tutte le frequenze esterne. Nella pratica non è fisicamente realizzabile.

Esempio:



$D_0 = 30,60,160$ . Il ringing nella prima immagine è così forte da ottenere delle linee di contorno distorte e spesse mentre nella terza immagine i contorni sono più puliti e meno distorti e gli oggetti più piccoli sono stati filtrati.

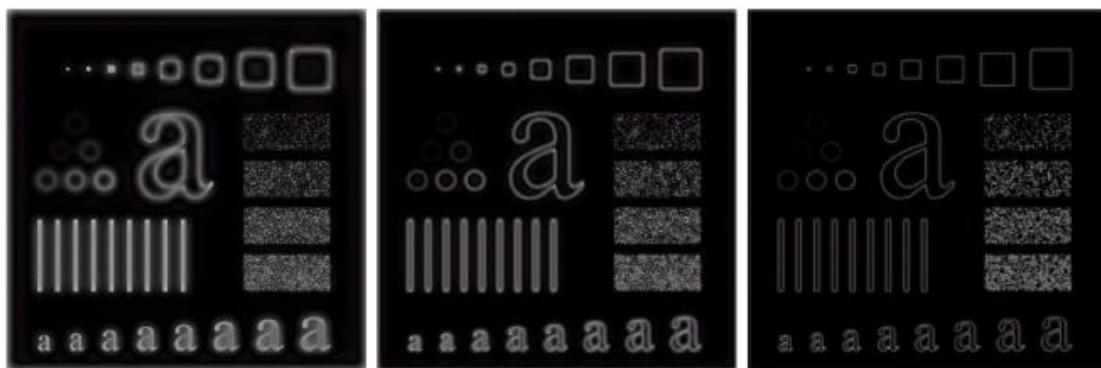
#### 5.4.2 Filtri passa-alto di Butterworth

Un filtro high pass di Butterworth 2-D BHPF di ordine  $n$  con frequenza di taglio uguale a  $D_0$  è definito:

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

Come nel caso dei filtri low pass, i filtri high pass di Butterworth hanno dei risultati più tenui rispetto ai filtri high pass ideali.

Esempio:



Filtro Butterworth di ordine 2 con  $D_0=30,60,160$ .

Come si nota i contorni sono molto meno distorti rispetto al filtro ideale anche per valori più piccoli della frequenza di taglio. I filtri di Butterworth non presentano effetto ringing, i bordi sono meno distorti il comportamento con i piccoli oggetti e l'effetto spot è comune ai filtri IHPF.

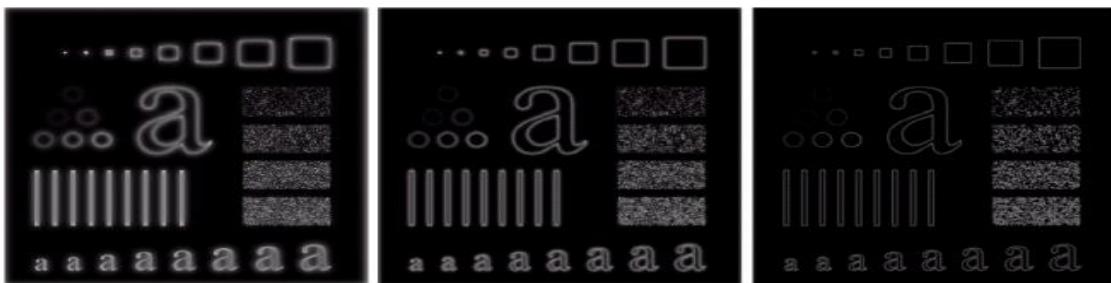
### 5.4.3 Filtri passa-alto Gaussiani

La funzione di trasferimento del filtro high pass gaussiano GHPF, la cui frequenza di taglio è posta a distanza  $D_0$  dal centro del rettangolo della frequenza è data da:

$$H(u,v) = 1 - e^{\frac{-D^2(u,v)}{2D_0^2}}$$

I risultati ottenuti sono più tenui di quelli dei precedenti due filtri.

Esempio:



Il filtraggio gaussiano dà risultati sensibilmente più smussati che negli altri due tipi e anche il filtraggio degli oggetti più piccoli e delle barre sottili è più pulito.

## 5.5 Il laplaciano nel dominio della frequenza

Un noto filtro per realizzare un enhancement controllato delle alte frequenze è il Filtro Laplaciano, già visto nel dominio spaziale. Si tratta di un filtro “isotropico”, la cui risposta cioè è indipendente dalla direzione della discontinuità ed è basato sull'utilizzo delle derivate seconde. In pratica enfatizza le eventuali discontinuità.

Si può dimostrare che:

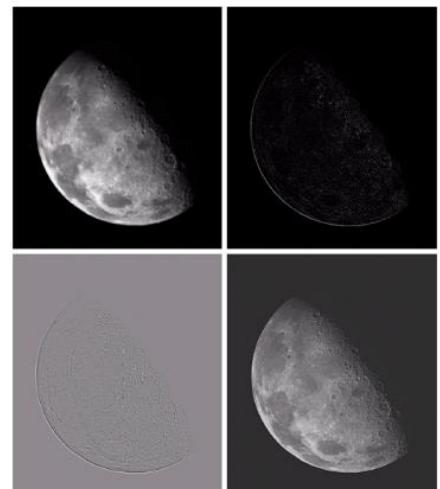
$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \Leftrightarrow - \left[ (u - \frac{M}{2})^2 + (v - \frac{N}{2})^2 \right] F(u,v)$$

Il filtraggio si realizza nella seguente maniera:

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

$$g(x,y) = \mathfrak{I}^{-1} \left\{ 1 + \left( (u - \frac{M}{2})^2 + (v - \frac{N}{2})^2 \right) \right\} F(u,v)$$

Tutti le immagini risultanti dai filtri passa alto fin'ora incontrate avevano in comune lo sfondo nero causato dall'eliminazione della componente continua. La soluzione a questo problema consiste nell'aggiungere una porzione dell'immagine di sfondo all'immagine filtrata. Il filtraggio con il



Laplaciano fa proprio questo aggiungendo l'intera immagine.

Esempio:

All'immagine in alto a sinistra viene applicato un filtro laplaciano. Non è possibile visualizzare pienamente l'output del filtro perché sono presenti valori sia positivi che negativi. La visualizzazione richiede quindi uno scaling opportuno dei valori. L'output del filtraggio evidenzia come i "dettagli" della scena siano maggiormente visibili.

## 5.7 Filtraggio omomorfico

Il filtraggio omomorfico si basa sui concetti dell'illuminazione e della riflettanza per andare a migliorare l'aspetto dell'immagine effettuando operazioni di compressione del range dell'intensità e miglioramento del contrasto.

Ciò è possibile perché un'immagine può essere espressa come prodotto delle componenti di illuminazione  $i(x,y)$  e riflettanza  $r(x,y)$ :

$$f(x,y) = i(x,y)r(x,y)$$

questa equazione non può essere utilizzata direttamente nel dominio della frequenza perché la trasformata di Fourier di un prodotto non è uguale al prodotto delle trasformate. Per poter operare nella frequenza è necessario separare le componenti di illuminazione e riflettanza, in modo tale che il filtro possa operare separatamente sulle due componenti (ad operare sulle due componenti separatamente è la funzione del filtro omomorfico  $H(u, v)$ , che tratta le componenti a bassa e alta frequenza in modo differente); il risultato finale di output sarà dato da:

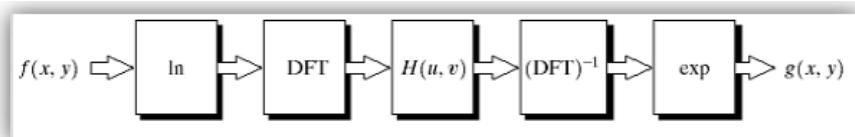
$$i_0(x,y) = e^{i'(x,y)}$$

$$r_0(x,y) = e^{r'(x,y)}$$

Passando ai logaritmi si ottiene:

$$z(x,y) = \ln f(x,y) = \ln i(x,y) + \ln r(x,y)$$

I passi fondamentali per arrivare a questo risultato sono i seguenti:



Il concetto fondamentale legato a questo tipo di filtraggio è che nel dominio spaziale la componente di illuminazione è caratterizzata da lente variazioni spaziali, mentre la componente di riflettanza tende a variare bruscamente soprattutto nei punti di separazione di oggetti differenti. Queste caratteristiche portano ad associare **le basse frequenze** della trasformata di Fourier con **l'illuminazione** e **le alte frequenze** con **la riflettanza**. Quindi per ottenere una compressione del range dinamico e un miglioramento del contrasto bisogna amplificare gli effetti dovuti alle alte frequenze(riflettanza) e attenuare gli effetti dovuti alle basse frequenze(illuminazione).

# 6. Filtraggio nel dominio della frequenza

Il filtraggio nel dominio della frequenza è possibile grazie alle caratteristiche della **serie di Fourier** e della **trasformata di Fourier**, che permette di lavorare su una funzione nel dominio di Fourier e, successivamente, ritornare al dominio originale della funzione senza perdere nessuna informazione. L'intuizione di Fourier attesta che ogni funzione periodica può essere espressa come la somma di seni e coseni moltiplicati per coefficienti appropriati (**serie di Fourier**). Anche se invece la funzione non è periodica (la cui area sotto la curva è finita) può essere espressa come somma di seni e coseni moltiplicati per una funzione peso (**trasformata di Fourier**). Visto che trattiamo funzioni (immagini) di durata finita il nostro interesse principale è nella trasformata di Fourier.

## 6.1 Serie di Fourier

La **serie di Fourier** è una somma di seni e coseni moltiplicati per coefficienti appropriati, che esprime una funzione  $f(t)$  di una variabile continua periodica  $t$ , con periodo  $T$ .

**DEF** Una **Serie Trigonometrica**

$$\frac{\alpha_0}{2} + \sum_{k=1}^{\infty} [\alpha_k \cos kx + \beta_k \sin kx] \quad (\text{forma reale})$$
$$\sum_{k=-\infty}^{+\infty} \gamma_k e^{ikx} \quad (\text{forma complessa})$$

è una **Serie di Fourier di  $f(x)$**  relativamente all'intervallo  $[-\pi, \pi]$   
se i coefficienti della serie  
**(coefficienti di Fourier)**  
sono dati da

coefficients of Fourier

$$\begin{cases} \alpha_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos kx \, dx & \text{forma reale} \\ \beta_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin kx \, dx & \text{forma complessa} \\ \gamma_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) e^{-ikx} \, dx & \end{cases}$$

## 6.2 Trasformata e Antitrasformata di Fourier continua

La **trasformata di Fourier** di una funzione continua  $f(t)$  di una variabile continua  $t$  è definita dalla seguente equazione:

$$F\{f(x)\} = F(u) = \int_{-\infty}^{+\infty} f(x) \exp[-j2\pi ux] dx$$

Quindi ad  $f(x)$  applico la Trasformata di Fourier ed ottengo  $F(u)$ . Per avere un informazione sul contenuto di un segnale possiamo considerare l'analisi in frequenza con Fourier. Cioè preso un segnale qualsiasi questo può essere esteso come una serie infinita di termini, questi termini sono detti **Funzioni base**. Nel caso di Fourier le Funzioni base sono :  $e^{-j2\pi ut}$  dove  $u$  è la **frequenza**,  $2\pi$  è la periodicità e  $t$  è la molteplicità della periodicità. L'idea è che la frequenza cioè l'andamento nell'unità di tempo di un segnale ci dà l'informazione sul contenuto informativo di un segnale. Ovviamente se dato un segnale qualsiasi riuscissimo a rappresentare tutti i termini (infiniti) avremmo una rappresentazione esatta del segnale originario ma se invece effettuiamo un troncamento sull'espansione effettuandola su una serie finita di termini anziche infinita abbiamo una rappresentazione del segnale che non è esattamente fedele

all'originale. La trasformata di fourier trasforma il segnale originario reale in un altro segnale che è rappresentato da numeri **complessi** poiché:

$$e^{-j2\pi\mu t} = \cos 2\pi\mu t - j \sin 2\pi\mu t$$

Attraverso la formula di Eulero, dove la prima parte è la parte reale mentre la seconda rappresenta la parte immaginaria ( $j$  è la radice di  $-1$ ). Queste due parti rappresentano il **numero complesso** che non lo si può rappresentare su un piano reale, allora il modo migliore per rappresentare un numero complesso su di un piano reale è quello di utilizzarne il **Modulo**. Per fare ciò si moltiplica il numero complesso per il suo **Coniugato**. Il complesso coniugato  $F(u)$  è ottenuto:

$$F(u) = \Re(u) + j\Im(u) = |F(u)| e^{j\phi(u)}$$

Considerando  $R$  come parte reale e  $I$  come parte immaginaria. Ed è anche uguale al **Modulo** per **La Fase**.

Il modulo si calcola effettuando il prodotto del numero complesso per il suo complesso coniugato che corrisponde a:

$$[R(u) - j I(u)] * [R(u) + j I(u)]$$

Ricordando che  $-j = \sqrt{-1}$ . Si effettua il semplice prodotto di un binomio tenendo solamente presente che il prodotto di radice di  $-1$  per radice di  $-1 = -1$  moltiplicato per  $-I(u) = +I(u)$ . e si ottiene alla fine  $R(u)^2 + I(u)^2$ . Effettuandone la radice quadrata si ottiene lo **Spettro di Fourier**:

$$|F(u)| = [\Re^2(u) + \Im^2(u)]^{1/2}$$

E si ottiene il **modulo**. Per rappresentare un numero complesso su di un piano immaginario abbiamo l'asse delle ascisse che rappresenta l'asse Reale e l'asse delle ordinate che rappresenta l'asse Immaginario. Quindi  $R(u)$  verrebbe rappresentato sull'asse delle ascisse mentre  $I(u)$  su quello delle ordinate identificando un punto quindi un vettore che è identificato da due quantità il **modulo** e la fase o angolo. La fase è calcolata mediante il teorema di pitagora applicando l'arcotangente:

$$\phi(u) = \tan^{-1} \left[ \frac{\Im(u)}{\Re(u)} \right]$$

A questo punto possiamo rappresentare

in modi o come **Modulo** o come **Fase**.

Ricapitolando se prendiamo un segnale 1D  $F(x)$ , ne facciamo la Trasformata di Fourier che a  $F(x)$  per ogni  $x$  associa un numero complesso, lo trasforma in numero reale effettuando il prodotto di numero complesso per complesso coniugato e otteniamo che per  $F(x)$  il suo Spettro di Fourier (numero reale).

Il vantaggio di passare dal dominio spaziale nel dominio delle frequenze sta nel fatto che nel dominio delle frequenze possiamo effettuare delle operazioni in modo migliore rispetto al dominio spaziale ma a discapito di un costo computazionale elevato poiché dobbiamo passare dal dominio spaziale al dominio delle frequenze e poi ritornare al dominio spaziale. **FFT(nlogn)** è la versione velocizzata di fourier. Quello che ci aiuta in questo caso è il Campionamento cioè passare da un segnale continuo ad un segnale discreto discretizzando il segnale a campioni regolari con un intervallo regolare determinato dal Delta. Il delta viene determinato grazie al contenuto informativo che ci viene fornito da Fourier grazie alla trasformata di fourier. La seconda cosa è il filtraggio cioè dopo aver campionato vogliamo effettuare un filtraggio cioè un miglioramento del segnale eliminando il rumore.

### Antitrasformata di Fourier continua

Ad  $F(u)$  applico l'antitrasformata di Fourier ed ottengo  $F(x)$ . In questo caso la  $f(x)$  deve essere continua e integrabile. L'obiettivo è trovare un  $F(u)$  tale che una volta integrata ci ridia  $f(x)$ .

$$F^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{+\infty} F(u) \exp[j2\pi ux] du$$

L' $F(u)$  scelta è:

$$F(u) = \int_{-\infty}^{+\infty} f(x) \exp[-j2\pi ux] dx$$

Sostituendo si ottiene:

$$f(x) = \iint_{-\infty}^{+\infty} f(x) \exp[-j2\pi ux] \exp[j2\pi ux] dx du$$

Il prodotto dei due esponenziali sommando gli apici si ha  $\exp[0] = 1$  e quindi:

$$f(x) = \iint_{-\infty}^{+\infty} f(x) dx du$$

Appare evidente che la funzione nel continuo **non è pienamente invertibile**, poiché essendo un doppio integrale possiamo portare fuori la  $f(x)$  dal secondo integrale poiché rispetto a  $du$  è costante ottenendo:

$$f(x) = \int_{-\infty}^{+\infty} f(x) dx \int_{-\infty}^{+\infty} du$$

Dove il secondo integrale =  $u$  quindi una **costante**, e il primo integrale non è  $f(x)$  ma bensì **l'Area di  $f(x)$** . Per ottenere solo  $f(x)$  e quindi garantire l'invertibilità della trasformata di fourier bisogna inserire dei vincoli sulla  $f(x)$ .

## 6.3 Trasformata e Antitrasformata di Fourier (DFT) Discreta

Considerando ora il caso discreto, supponiamo di voler applicare la trasformata di fourier dopo aver campionato e discretizzato. Per garantire l'invertibilità della trasformata di fourier bisogna applicare dei vincoli sui valori. A differenza del caso continuo invece di avere un integrale tra meno infinito e più infinito abbiamo una sommatoria di un numero **Finito** di valori, questo numero di valori è  $N$  perché vi è una periodicità. Sostituendo l'integrale di prima con la sommatoria si ottiene:

$$F(u) = \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux] \text{ per } u = 0, \dots, N-1$$

Ora, per poter garantire in assoluto la sua **invertibilità** dobbiamo tenere conto di due fattori, cioè di un fattore di **scala sull'esponenziale** considerato e l'altro è **un fattore di scala sul numero di valori considerati**. Il fattore di scala è il numero di campioni, cioè  $N$ .

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp\left[-j2\pi ux/N\right], \quad \text{per } u = 0, \dots, N-1$$

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp\left[j2\pi ux/N\right], \quad \text{per } x = 0, \dots, N-1$$

Possiamo osservare che per  $u = 0$ , nella trasformata di fourier, l'esponenziale diventa 1 e quindi restituisce il valore medio dei valori di  $f(x)$ . Per ogni valore di  $u$  otteniamo una sommatoria dei valori di  $f(x)$  modulati da una funzione base dove nel primo caso ( $u=0$ ) abbiamo il valor medio di tutta la serie di valori considerati.

Effettuando, come nel caso continuo, una sostituzione di  $f(x)$  nella trasformata otteniamo:

$$F(u) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{x=0}^{N-1} F(u) \exp[-j2\pi ux] \exp[j2\pi ux]$$

L'esponenziale anche in questo caso diviene 1:

$$F(u) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{x=0}^{N-1} F(u)$$

Ora il numero di valori di entrambe le sommatorie è  $N$  quindi si semplifica con la frazione ottenendo come risultato la  $F(u)$ . Questa trasformata così costruita è detta **Per Costruzione**, cioè la **frazione** è messa appositamente per garantire l'invertibilità, mentre l' $N$  dell'esponenziale è messa perché se scaliamo il valore finale dobbiamo necessariamente scalare anche l'esponenziale.

La DFT ha una complessità di  **$O(n^2)$** . Un algoritmo che apporta riduzioni significative in termini di calcoli è l'FFT (Fast Fourier Trasform) o Trasformata veloce di Fourier che riduce i calcoli a  $MN \log MN$ . Applicando l'FFT all'immagine di 1024x1024 pixel richiederebbe circa 20 milioni di moltiplicazioni e addizioni il che è una riduzione significativa rispetto ai tempi richiesti da un implementazione diretta.

## 6.4 La trasformata e Anttrasformata di Fourier 2D (DFT) Discreta

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

con  $u = 0, \dots, M-1$  e  $v = 0, \dots, N-1$ .

La funzione iniziale è  $f(x, y)$ , stiamo supponendo che sia già capionata quindi il numero di campioni saranno  $MN$  e la seconda parte scaliamo una volta sulla direzione delle  $x$  e l'altra sulla dimensione delle  $y$ . La particolarità della Trasformata2d di Fourier è che le funzioni base sono Separabili, cioè è possibile riscrivere la funzione base  $\exp$  di dimensione 2 è possibile riscriverla come prodotto di due funzioni base di dimensione 1. Questa proprietà è detta **Separabilità**:

### 6.4.1 Proprietà di Separabilità:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \exp[-j2\pi u x/N] \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi v y/N]$$

Cioè effettuiamo il prodotto della  $f(x,y)$  per l'esponenziale dopodiche lo sommo moltiplicandolo con l'altro esponenziale. Ma la prima parte non è altro che La Trasformata di Fourier bidimensionale nella direzione delle  $y$  a meno di un fattore di scala ( $N$ ) al cui risultato viene applicata la trasformata di fourier nell'altra direzione.

Cioè la funzione di prima è riscrivibile:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} F(x, v) \exp[-j2\pi u x/N]$$

dove:

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

*Da notare che  $F(x, v)$  è una notazione matematicamente sbagliata perché  $x$  indica lo spazio mentre  $v$  indica la frequenza.* Quindi, riassumendo, sfruttando questa proprietà è possibile abbassare il costo computazionale cioè si è passata da **una polilogaritma ( $m^*n \log m^*n$ )**, ad una complessità lineare :

$$(n \log n) + (m \log m)$$

### 6.4.2 Traslazione

La traslazione è molto utile per una migliorare visualizzazione dello spettro. Se calcolo la TF di un'immagine e successivamente questa immagine trasla, non c'è bisogno di calcolare la trasformata, infatti sapendo lo spostamento faccio una correzione alla trasformata di fourier già calcolata.

La proprietà di traslazione dice che:

$$f(x, y) \exp[j2\pi(u_0 x + v_0 y)/N] \Leftrightarrow F(u - u_0, v - v_0)$$

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v) \exp[-j2\pi(ux_0 + vy_0)/N]$$

Cioè moltiplicando  $f(x, y)$  per l'esponenziale mostrato si trasla l'origine della DFT in  $(u_0, v_0)$  e, viceversa, moltiplicando la DFT,  $F(u, v)$ , per il negativo dello stesso esponente si trasla l'origine della  $f(x, y)$  in  $(x_0, y_0)$ . Quindi non bisogna ricalcolare la trasformata di fourier , cioè traslando di un certo valore nel dominio spaziale sappiamo cosa cambiare nel dominio delle frequenze.

**La traslazione non ha effetti sulla spettro** di  $F(u, v)$  della trasformata dato che:

$$|F(u, v) \exp[-j2\pi(ux_0 + vy_0)/N]| = |F(u, v)|$$

Queste proprietà vengono utilizzate per una migliore visualizzazione dello spettro.

Nel caso bidimensionale è utile prima di effettuare la trasformata applicare uno shift (traslazione) dell'origine nel punto  $(M/2, N/2)$  cioè nel centro del rettangolo delle frequenze. Dalle relazioni di cui sopra ciò si ottiene ponendo  $u_0=M/2$ ,  $v_0=N/2$  da cui:

$$\Im[f(x,y)(-1)^{(x+y)}] = F(u-M/2, v-N/2)$$

In questo modo i dati vengono traslati in maniera tale che  $F(0,0)$  risulti il centro del rettangolo delle frequenze definito tra  $[0,M-1]$  e  $[0,N-1]$ .

### 6.4.3 Rotazione

Per la rotazione dobbiamo passare in coordinate polari quindi da un valore  $(x,y)$  passiamo ad un valore  $(\rho, \theta)$ . Dove  $\rho$  è la distanza del punto dall'origine e  $\theta$  è l'angolo. La proprietà ci dice che se effettuiamo una rotazione nel dominio spaziale (sommendo un angolo) lo stesso effetto di rotazione si avrà nel dominio delle frequenze e viceversa:

Rappresentando  $f(x,y)$  e  $F(u,v)$ , rispettivamente, nelle coordinate polari  $r$  e  $\theta$ ,  $\omega$  e  $\phi$ :

$$x=r \cos \theta, \quad y=r \sin \theta \quad u=\omega \cos \phi, \quad v=\omega \sin \phi$$

$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \phi + \theta_0)$$

### 6.4.4 Valore Medio

Definita prima

### Periodicità

La periodicità dice che se osserviamo la trasformata di fourier oltre i valori per la quale la calcoliamo ci ritroviamo ancora la trasformata di fourier poiché:

Nel caso **1D** la DFT e la sua inversa sono periodiche con periodo pari ad  $N$ . Infatti per il kernel diretto vale la seguente relazione:

$$\exp\left[\frac{-j2\pi ux}{N}\right] = \exp\left[\frac{-j2\pi(u+kN)x}{N}\right]$$

Per qualunque  $k$  intero. Ricordando che  $\exp[\dots] = \cos 2\pi ux - j \sin 2\pi ux$ , il cos e il sen sono periodici e quindi risulta anche che l'esponenziale è periodico di periodo  $N$ . quindi risulta evidente che:

$$F(u) = F(u + N).$$

Cioè che la trasformata di fourier di  $u$  risulta essere uguale alla trasformata di fourier di  $u+N$  poiché è periodica di passo  $N$ . Vale lo stesso anche per l'antitrasformata:

$$f(x) = f(x + N)$$

Nel caso 2D, si ottiene che:

$$F(u, v) = F(u + hM, v + kN)$$

Con  $h$  e  $k$  due interi qualsiasi, e  $M, N$  sono le dimensioni dell'immagine. In particolare se  $N = M$ , cioè per immagini che hanno la stessa dimensione di righe e colonne si ha:

$$F(u, v) = F(u + N, v) = F(u, v + N) = F(u + N, v + N)$$

Cioè, è periodica per righe, periodica per colonne e quindi è periodica sia per righe che per colonne.

#### La periodicità della DFT nel caso 1D da luogo ad un interessante interpretazione geometrica:

Dato che abbiamo valori distinti  $F(0), F(1), F(2) \dots F(N-1)$  con  $F(N) = F(0)$  cioè l'ultimo è uguale al primo , quindi è possibile vedere i valori come se fossero su un cerchio invece che su di una linea retta. Questa interpretazione è detta **Anello di Fourier**.

Nel caso 2D, poiche l'anello lo si ottiene non piu su una sola direzione ma su entrambe le direzioni otteniamo una forma toroidale(ciamella). Questo viene detto **Toro di Fourier**.

#### 6.4.5 Simmetria

Se spezzassimo lo spettro di fourier a metà noteremmo che vi è una riflessione. Se  $f(x)$  è reale la  $F(u)$  è dotata di quella che viene definita **Simmetria Coniugata**:

$$F(u) = F^*(-u)$$

Dove  $F^*(u)$  è il Complesso coniugato di  $F(u)$ . Per lo spettro si può scrivere che:

$$|F(u)| = |F(-u)|$$

Poiche :

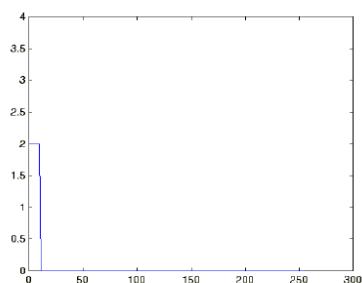
$$|F(u)| = F(u) * F^*(u)$$

$$|F(-u)| = F(-u) * F^*(-u)$$

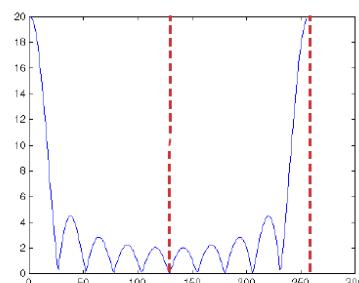
Dato che  $F(u) = F^*(-u)$  i due spettri sono uguali.

#### Esempio 1D:

Funzione boxcar:

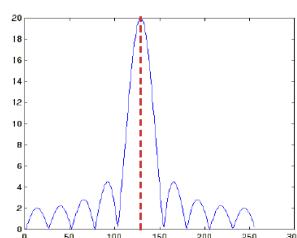


$f(x), x = 0, \dots, 255$



$|F(u)|, u = 0, \dots, 255$

Quella a sinistra è la funzione boxcar reale, mentre quello a destra è il risultato della trasformata di fourier sulla boxcar calcolata su 256 valori, ci si accorge subito che la simmetria la abbiamo nel valore 128 (a metà). Si spezza a 256 per replicare significa che dopo il 256 si ripete. Per visualizzare **un solo periodo**, si può sfruttare la proprietà di traslazione spostando l'origine dell'asse  $u$  nel punto  $u = N/2$  moltiplicando la  $f(x)$  per  $(-1)^x$  ottenendo:



## 7. Segmentazione di immagini

La **segmentazione** suddivide un'immagine nelle regioni o negli oggetti che la compongono questo per migliorare la comprensione della scena. Esistono vari metodi per effettuare la segmentazione di un'immagine:

- *Metodi basati sul Pixel (Pixel-based)*: si utilizzano i valori di luminosità dei singoli pixel;
- *Metodi basati sul Contorno (Edge-based)*: si estraggono i contorni per delimitare gli oggetti;
- *Metodi basati sulla forma (Model-based)*: si parte dalla conoscenza a priori del modello dell'oggetto;
- *Metodi basati sulla Regione (Region-based)*: si analizzano i valori di luminosità di aree dell'immagine.

### Segmentazione orientata alla Regione

La segmentazione orientata alla regione tenta di conservare le informazioni strutturali delle regioni interne alla regione stessa. L'immagine R viene partizionata in sottoregioni  $R_1, R_2, \dots, R_n$  tale che:

- $\bigcup_{i=1}^n R_i = R$ , l'unione di tutte le regioni forma l'immagine completa
- $R_i$  è un'area连通的, cioè *presi due punti all'interno di una regione esiste un cammino che li congiunge e appartiene ancora alla regione*
- $R_i \cap R_j = \emptyset$ , cioè prese a due a due le regioni per tutti i valori  $i$  e  $j$ ,  $i \neq j$ , non hanno nulla in comune
- $Q(R_i) = VERO$  per  $i = 1, 2, \dots, n$  indica le proprietà che devono essere soddisfatte dai pixel in una regione segmentata, questo predicato è detto **predicato di verità**. Si richiede che la regione sia uniforme (non necessariamente uniforme in colore, tessitura o forma) ma uniformità del livello di grigio all'interno della regione tale che sia valido un criterio di uniformità. Cioè l'esistenza di un predicato.
- $Q(R_i \cup R_j) = FALSO$ , sulla base dello stesso predicato, ogni regione sia diversa da tutte le altre.

Esistono due approcci:

- **Region Growing**, o accrescimento di regioni: è un approccio bottom-up cioè a partire dai singoli pixel e dai loro livelli di grigio si costruiscono le regioni nelle quali viene partizionata l'immagine.
- **Split & Merge**, è un approccio top-down a partire dall'immagine determina le regioni in cui suddividere l'immagine stessa. L'immagine viene spezzata in modo iterativo fino ad arrivare al punto in cui l'elemento costitutivo delle sottoregioni non ha un predicato vero e quindi non possa essere ulteriormente suddiviso.

Il problema della segmentazione delle regioni si riduce alla definizione ed implementazione del **predicato di verità**, cioè: qual'è il predicato di verità tale che applicato ad una regione sia vero e applicato all'unione di regioni sia falso?

#### Region Growing (accrescimento di regioni)

Il Region Growing è una tecnica che raggruppa i pixel o sottoregioni in regioni via via più grandi basandosi su criteri predefiniti di **similarità** o **similitudine**. I pixel da cui si parte vengono chiamati **seed** (semi), e man mano si aggiunge a questi semi quei pixel dell'intorno (4 o 8) che hanno delle proprietà predefinite simili a quelle del seme stesso. Vengono introdotti 3 concetti:

1. Il seme da cui partire
2. Concetto di adiacenza (4 adiacenza / 8 adiacenza)
3. Similarità valutata ad un predicato di verità

Il primo pixel selezionato può essere semplicemente **il primo pixel** non etichettato nell'immagine oppure un insieme di pixel. Come **predicato di verità** tipicamente viene adottato un **test statistico** per decidere quali pixel vanno aggiunti alla regione (pixel simili).

### In dettaglio:

Sia **R** la regione di **N** pixel finora creata e **P** un pixel vicino con livello di grigio **y**. Definiamo la media **X** e la varianza **S<sup>2</sup>**:

$$X = 1/N \sum_{(r,c) \in R} I(r, c)$$

$$S^2 = \sum_{(r,c) \in R} (I(r, c) - X)^2$$

Stiamo calcolando due valori statistici, uno è la media e l'altro è il valore indicante di quanto l'intensità luminosa di ogni pixel si discosta dal valor medio (**test statistico T**). Se il test statistico T risulta essere al di sotto di una certa soglia allora i pixel con quel livello di grigio y appartengono alla regione R altrimenti no.

**Il test statistico T** è definito come:

$$T = \left( \frac{(N-1)N}{(N+1)} (y - X)^2 / S^2 \right)^{1/2}$$

Y (valore di intensità del pixel nuovo) meno lo scarto quadratico (X) fratto la varianza. Tutto questo viene normalizzato e ne viene effettuata la radice quadrata.

*In conclusione il test statistico T possiede una distribuzione T n-1 se tutti i pixel in R e il pixel di test y sono normali, indipendenti e identicamente distribuiti.*

Cioè ogni livello di grigio dipende solo da se stesso e non dai livelli di grigio circostante. Inoltre ipotizziamo una distribuzione di tipo normale e quindi con una certa media e una certa varianza.

Esempio:

Si consideri il frammento di immagine a lato, in cui i valori mostrati rappresentano livelli di grigio e quelli segnati indicano i **punti-seme**. Utilizzando due semi si potranno ottenere al più due regioni, **R1** e **R2**. Il **predicato P** è il seguente:

un pixel appartiene ad una regione se la differenza, in valore assoluto, tra il suo livello di grigio e il livello di grigio del seme corrispondente è inferiore ad una soglia T.

Se un pixel soddisfa il criterio per entrambi i semi, lo si assegna a R1. In basso è mostrato il risultato della segmentazione per T = 3: si ottengono appunto due regioni, i cui pixel sono marcati rispettivamente con a e b.

Se invece T=8 si ottiene solo la regione mostrata:

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

1e	<i>R<sub>1</sub></i>	5	6	7	
0	0	5	6	7	
1	1	5	8	7	<i>R<sub>2</sub></i>
0	1	6	7	7	
2	0	7	6	6	
0	1	5	6	5	

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

Il problema è la determinazione della soglia di discriminazione in modo tale da riuscire ad accrescere le regioni a partire dai semi iniziali.

### 7.3.2 Split & Merge

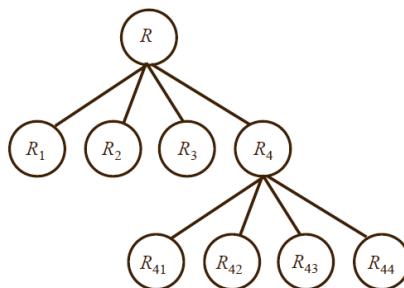
Questa tecnica inizia con l'intera immagine considerandola come un'unica regione  $R$ , viene poi separata in 4 sottoregioni se il predicato ad essa applicato è falso. La suddivisione continua fintanto che il predicato applicato ad ogni regione continua ad essere falso. Una volta terminata la suddivisione inizia l'unione dal basso verso l'alto unendo le regioni adiacenti per cui il predicato di verità risulta essere VERO. La struttura dati dietro quest'algoritmo è un albero quaternario (al max 4 figli).

L'algoritmo quindi è composto dei seguenti passi:

1. **scindere** in quattro quadranti disgiunti ogni regione  $R_i$  per la quale  $Q(R_i) = \text{FALSO}$
2. quando non è più possibile separarle, applicare il processo di merging alle regioni adiacenti  $R_j$  e  $R_k$  per le quali  $Q(R_j \cup R_k) = \text{VERO}$
3. fermarsi quando il processo di merging non è più realizzabile

Vediamo un esempio:

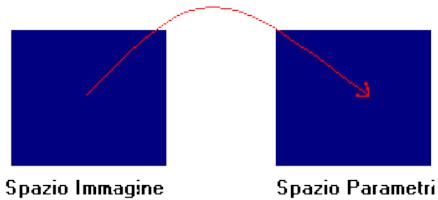
$R_1$	$R_2$	
$R_3$	$R_{41}$	$R_{42}$
	$R_{43}$	$R_{44}$



## 7.2. Segmentazione orientata sulla forma

### 7.2.1 La trasformata di Hough

La trasformata di Hough (HT) è una tecnica che consente di individuare rette, curve, forme note presenti nell'immagine piuttosto che nell'immagine, in uno spazio ad esso correlato, appunto definito **spazio di ricerca della trasformata di Hough**, cercando dei massimi di una funzione valutata in questo nuovo spazio.



Ricerca di rette in uno SP

Iniziamo introducendo due passaggi di base del lavoro HT utilizzando il caso di rette:

1. **Trasformare il rilevamento di una retta in un problema di intersezione di rette.** Ogni retta,  $y=mx + q$ , è identificata da un'unica coppia di parametri,  $(m, q)$ . Pertanto, la retta è rappresentata da un punto nel piano  $m, q$  (spazio dei parametri). Viceversa, ogni punto  $p=[x,y]$  nell'immagine, corrisponde ad una retta  $q=x(-m) + y$  nello spazio dei parametri, che al variare di  $m$  ed  $q$ , rappresenta tutte le possibili rette passanti per  $p$ . Pertanto, una retta definita da  $N$  punti  $p_1 \dots p_N$ , è identificata nello spazio dei parametri dall'intersezione delle rette associate con  $p_1 \dots p_N$ .
2. **Trasformare l'intersezione di retta in un semplice problema di rilevamento di massimi locali.** Immaginiamo di dividere il piano  $m, q$  in un numero di celle finito, la cui risoluzione dipende dall'accuratezza di cui abbiamo bisogno, e di associare un contatore,  $c(m,n)$ , inizialmente settato a 0, per ogni cella. Assumiamo per semplicità che l'immagine contiene soltanto una retta,  $(m', n')$ , formata dai punti  $p_1 \dots p_N$ . Per ogni punto  $P_i$ , si **incrementano** (voting) tutti i contatori corrispondenti a quelle rette (nello SP) passanti per il punto  $P_i$  ed infine si ricercano i massimi locali che rappresentano l'evidenza che quei punti appartengono ad una retta.

L'equazione classica della retta  $y=mx+q$  presenta dei problemi, in quanto  $m$  e  $q$  assumono valori reali:

$$-\infty < m, q < +\infty.$$

In particolare il problema sorge quando le rette considerate sono **perpendicolari all'asse delle ascisse**, o parallele all'asse delle ordinate, in quanto la tangente al punto risulta essere infinito, e quindi il coefficiente angolare è anch'esso infinito.

**Problema:** non è rappresentabile nello SP la retta perpendicolare all'asse delle ascisse.

Lo spazio immagine per immagini reali è ovviamente limitato (esempio 256x256) e quindi per ovviare al problema, vanno rapportati dei valori che abbiamo un intervallo limitato, si usa una diversa rappresentazione della retta, trasformare quindi tutti i punti in coordinate polari:

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

- $\rho$ : Distanza che intercorre tra la retta e l'origine del piano,
- $\theta$ : È l'angolo che forma la retta con l'asse delle ascisse.

In questo caso i parametri risultano limitati:  $0 < \rho < R\sqrt{2}$ ;  $-\pi < \theta < +\pi$

### Processo di voto:

Immaginiamo di discretizzare lo spazio SP come una **matrice multidimensionale** (denominata **array di accumulatori**) le cui **entry** sono dette **celle**, e corrispondono ad una edizione quantizzata della curva. Lo SP ha sulle ascisse l'angolo  $-\pi \leq \theta \leq +\pi$  e sulle ordinate  $\rho$ .

Per eseguire la trasformata si esegue una **scansione** dell'immagine nello SI:

Per ogni punto appartenente alla retta nello SI si votano tutte le celle rappresentate da  $\theta$  e  $\rho$  appartenenti alle rette passanti per quel punto, compreso la retta alla quale essa stessa appartiene. Quindi ogni pixel **vota per una o più celle** dello SP incrementandone la probabilità. Al termine della scansione le celle che hanno accumulato i **valori più elevati** indicano le **soluzioni più probabili** del problema. Nel caso della retta, ogni punto sull'immagine vota tutte le rette  $(\theta, \rho)$ , passanti per quel punto. Più voti ha per una retta più si ha la possibilità che nel SI quell'insieme di pixel rappresentano una retta, a questo punto si fa la ricerca dei **massimi locali** che non è agevole se si considera che ci può essere rumore, nello SI, pertanto va necessariamente introdotta una **soglia**, che deve rappresentare un compromesso tra il rischio di non rilevare oggetti, e quello di ottenere dei falsi positivi. In generale, al **termine del processo di voto** ogni cella misura il numero di contributi al riconoscimento della curva corrispondente. La **definizione dello spazio di parametri e dei meccanismi di voto** dipendono dal tipo di oggetto che si vuole identificare.

### - Uso del gradiente:

La dimensione delle celle dipende dalla precisione massima richiesta. Per ridurre la dimensione nello spazio dei parametri è opportuno far votare non tutte le coppie dei parametri dato un punto nello SI, ma soltanto quelli **rilevanti**, cioè soltanto quelle il cui valore del **gradiente** supera una certa soglia. Il valore del gradiente può essere calcolato con gli operatori differenziali (Sobel, oppure operatore isotropico).

Per esempio per l'equazione della retta:

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

Si vota solo il punto  $(\rho, \theta)$ , dove,  $\theta$  è la **direzione del gradiente**.

L'idea è: per un punto di coordinate  $(x, y)$ , conservo il valore del gradiente (vettore, derivata della funzione rispetto a  $x$ , e  $y$ , ne calcolo la direzione del gradiente con l'arcotangente), e voto soltanto tutti i valori  $(\rho, \theta)$ , dove,  $\theta$  è fissato.

Per la circonferenza è lo stesso:

piuttosto di votare tutte le circonferenze che distano di un certo raggio  $r$ , si vota soltanto le coordinate le centro della circonferenza per il punto  $(x_c, y_c)$  calcolati come:

$$x_c = x - \rho \sin(\theta); y_c = y - \rho \cos(\theta)$$

dove ancora  $\theta$  è calcolato come direzione del gradiente.

Sia  $pn$  che  $\theta$  sono array contenenti gli intervalli discretizzati degli spazi parametrici  $p$  e  $\theta$ .

### Algoritmo di Hough

1. Si azzeri la matrice  $H(.,.)$ ;
2. Per ogni punto  $P \in F$ ,  $P=(x, y)$ 
  1. per  $\theta_n$  che varia tra  $-\pi/2$  e  $\pi/2$  con passo  $d\theta$ 
    1. si valuti  $\rho(n) = x \cos(\theta_n) + y \sin(\theta_n)$
    2. si ricavi l'indice  $m$  corrispondente a  $\rho(n)$
    3. si incrementi  $H(m, n)$
  2. end
3. end
4. Si individuino i massimi locali su  $H(.,.)$  corrispondenti ai parametri dei segmenti individuati

Rilevamento circonferenza noto il raggio

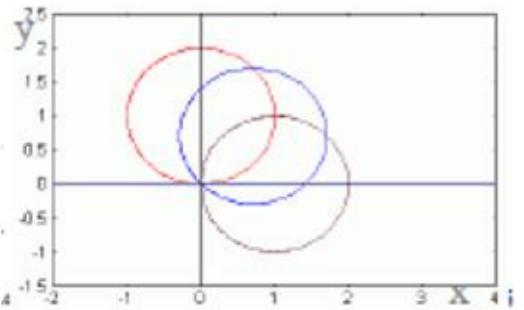
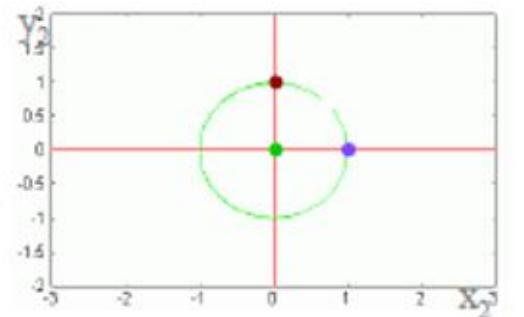
**Equazione:**

$$(y - yc)^2 + (x - xc)^2 = r^2$$

Noto il raggio, cerchiamo tutte le circonferenze aventi quel raggio facendo variare solo l'ascissa e l'ordinata del centro della circonferenza. A questo punto il problema di riduce a "votare" tutti i centri di circonferenze che si trovano a una distanza di Raggio pari a "r" prefissato, dal punto preso in esame.

Esempio:

consideriamo lo SI in alto a Dx, e prendiamo il punto rosso, esso vota il centro verde, pertanto vota una circonferenza che ha come centro il punto **verde**, ma vota anche tutte le altre circonferenze di cui esso può essere centro, pertanto vota tanti centri di circonferenze. Essendo nel discreto il punto rosso (o blu) non voterà infinite circonferenze che si trovano a distanza  $r$  da esso, ma infinite curve alle quali il punto in questione può appartenere. In questo caso lo SP è **bidimensionale** e la curva generata da ogni punto nello SI è essa stessa una circonferenza.



**N.B. è raro che a una curva nello SI corrisponda la stessa curva nello SP**

Lo spazio dei parametri è generato dalle coordinate del centro  $(xc, yc)$ . Nota la tangente nel punto  $(dy/dx)_i$  la compatibilità è ridotta ad un punto:

**invece di votare infinite circonferenze possiamo votare quelle circonferenze che distano  $r$  dal punto in esame:  $(xc, yc)$  riducendo così l'insieme dei punti che sono compatibili all'insieme originale.**

**Esempio circonferenza con raggio incognito**

In questo caso lo SP è **tridimensionale** e la curva assumerà la forma:

$$f((x, y), (xc, yc), r) = (y - yc)^2 + (x - xc)^2 - r^2 = 0$$

La curva generata in questo caso è un **cono**.

Nota la tangente nel punto  $(dy/dx)_i$  la compatibilità è ridotta ad una retta:

$$yc = -1/m_i \cdot xc + (y_i - m_i x_i).$$

Nota la curvatura nel punto ( $\Rightarrow r$ ) la compatibilità è ridotta ad un punto:  $xc, yc$ , e veniamo a conoscenza del raggio, quindi lo SP da tridimensionale diventa bidimensionale-

**Processo di voto:**

Ogni punto nello SI vota un insieme di coppie nello SP, coppie  $(m, q)$ , coppie (ascisse, ordinata) del centro o (ascisse, ordinate, raggio) del centro della circonferenza.

Voto: pesare la significatività di trovare proprio la circonferenza di quel centro con quel raggio. Trovare il massimo voto che un punto ha ottenuto, per stabilire la presenza della curva cercata.

Per esempio se voglio cercare le rette vado a cercare dopo aver effettuato tutta la scansione nello SI, aver votato tutte le rette di coefficienti angolari  $m_i$  e  $q_i$  nello SP, vado poi a trovare il massimo locale o i massimi

locali nello SP , questi corrispondono ancora alle coppie  $(m_i, q_i)$ , e quindi corrispondono a uno o più rette presenti nello SI.

Lo stesso si applica per le circonference.

Nella ricerca di curve espressa in forma analitica, l'algoritmo opera nel seguente modo:

- Ogni punto nello SI genera una superficie nello SP
- Quindi nello SP una intersezione di molte superfici è l'indizio della presenza di una particolare istanza della curva analitica cercata

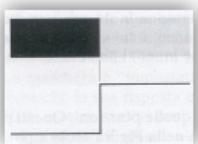
**Per avere un tale indizio è necessario avere un numero di punti almeno pari al numero dei parametri per individuare una curva (cioè le sue caratteristiche)**

# 7.1 Segmentazione orientata sul contorno

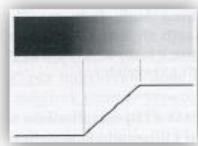
## 7.1.1 Modelli di edge

L'**edge detection**, cioè l'individuazione dei bordi, è il metodo più utilizzato per la segmentazione di immagini e si basa sui bruschi cambiamenti locali di intensità. Esistono tre tipi di edge, classificati secondo i loro profili di intensità:

1. **a gradino**:



2. **a rampa**:



3. **roof**:



Per estrarre gli edge possono essere utilizzati due algoritmi:

- **Canny edge detector**
- **Harris Corner Detector**

## 7.1.4 Canny edge detector

L'algoritmo di Canny è un operatore per il **riconoscimento dei contorni** (edge). Lo scopo di Canny era quello di trovare il miglior algoritmo possibile per riconoscere i contorni delle immagini poiché i filtri tradizionali (Prewitt,Sobel...) producono immagini dove gli edge sono spesso ben visibili ma non direttamente utilizzabili in quanto l'applicazione di una soglia di **binarizzazione** (cioè applicare il thresholding, scegliendo una soglia o manualmente oppure tramite un algoritmo “**Otsu**”, portando per esempio tutti i valori superiori ad una soglia a 255 in modo tale da determinare i bordi) ha l'effetto di spezzettare uno stesso edge in molti frammenti tra loro **non connessi**. Vengono quindi generati numerosi punti e piccoli segmenti “spuri” a causa di rumore o piccole variazioni di intensità locali. Il metodo di Canny produce, invece, edge **connessi** (cioè presi due pixel esiste un cammino che li unisce).

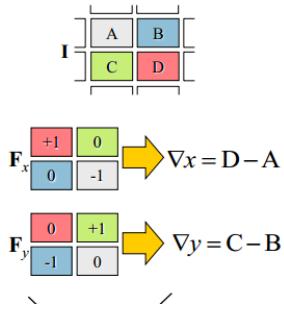
Fasi di elaborazione dell'algoritmo di Canny

### 1. Riduzione del rumore

La riduzione del rumore avviene applicando un filtro passa-basso, utilizzando uno **smoothing gaussiano**. Smussare equivale a rendere più piane le zone uniformi e a marcare i contorni. Molto importante per un buon risultato sono le scelte dei parametri (dimensione del supporto e sigma).

### 2. Ricerca del gradiente della luminosità di un'immagine

Dopo il passo precedente applica un filtro passa-alto utilizzando l'operatore di Roberts poiché roberts ha un supporto 2x2 ed è computazionalmente meno oneroso.



Attraverso il quale si stimano il **modulo** e l'**orientazione** del gradiente:

$$\|\nabla\| = \sqrt{\nabla x^2 + \nabla y^2} \cong |\nabla x| + |\nabla y|$$

$$\theta = \arctan_q(\nabla y, \nabla x) + \frac{\pi}{4}$$

### 3. Soppressione dei non-massimi

L'obiettivo è quello di eliminare dall'immagine modulo-gradiante i pixel che non sono massimi locali rispetto all'orientazione del gradient. Per fare questo vengono implementati due diversi approcci:

- Il primo è basato sul **tracking** delle creste. Cioè partendo da un pixel di bordo si inseguono i pixel vicini appartenenti a un 4-intorno o ad un 8-intorno con passo unitario. Gli edge ottenuti sono già di spessore unitario
- Il secondo (più semplice) analizza l'intorno 3x3 di ogni pixel ed elimina i pixel che non rispettano la condizione di Massimo locale (orientato). Viene eseguito un ulteriore passo di **thinning** (assottigliamento) per ridurre a spessore unitario gli edge ottenuti. Viene utilizzato il concetto di **isteresi**.

### 4. Individuazione dei contorni mediante sogliatura con **isteresi**

L'estrazione dei contorni dalla mappa generata dallo step precedente si esegue con un procedimento chiamato **sogliatura con isteresi**. Vengono definite due soglie **T1** e **T2**, con **T1 > T2**, che vengono confrontate con il gradiente in ciascun punto. Se il valore del gradiente è:

- **inferiore alla soglia bassa**, il punto è scartato;
- **superiore alla soglia alta**, il punto è accettato, cioè sicuramente è un punto di edge;
- **compreso fra le due soglie**, il punto è accettato solamente se solo se sono **Connnessi** in un 4-intorno o 8-intorno ad un pixel ritenuto valido nel passo precedente
- vengono considerati validi il problema che si presenta è la scelta corretta di T1 e T2

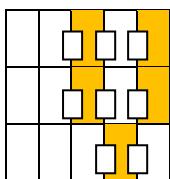
La presenza di due soglie (*da cui il riferimento all'isteresi*) è giustificato dal fatto che è praticamente impossibile trovare un unico valore del gradiente di luminosità per discriminare se un punto appartiene o no ad un contorno. Al termine di questo step si ottiene un'immagine binaria dove ciascun pixel è marcato come appartenente o no ad un contorno. La mappa ottenuta in questo modo può essere trattata come un insieme di curve di contorno che, previa ulteriore elaborazione, può essere approssimato da una poligonale.

### 7.1.5 Harris corner detection

I contorni di un'immagine sono anche i dettagli, le caratteristiche peculiari dell'immagine, in quanto se un'immagine è piatta non ci dà alcuna informazione. Quindi l'EDGE DETECTOR è visto anche come un modo per evidenziare le caratteristiche peculiari di un'immagine. Tutti i punti di contorno sono punti d'interesse, sono punti caratteristici di un'immagine.

Tali punti caratteristici sono detti: "Corner".

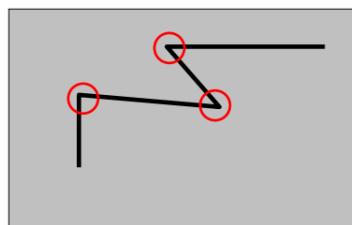
Prendiamo in esame una griglia:



I CORNER è un contorno comune a due contorni, cioè è un punto d'interesse, in comune tra due contorni.

Rilevare i corner su un'immagine bidimensionale:

vogliamo rilevare i corner in modo tale che tale rilevamento sia ponderato al rumore, in particolare al rumore gaussiano. D'altronde il rilevamento dei corner su un'immagine pulita non è preciso.



L'obiettivo è rilevare i corner in modo tollerante al rumore. Per calcolare i corner prendiamo un punto  $P(x,y)$ , consideriamo il suo intorno  $7 \times 7$ , e all'interno del suo intorno calcoliamo una matrice di **covarianza**.

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

Dove  $I$  è l'immagine,  $I_x$  e  $I_y$  sono le componenti del **gradiente** dell'immagine rispettivamente nelle direzioni x e y. Il corner viene determinato se il più piccolo autovalore di questa matrice  $\lambda_2$  è maggiore di una soglia prefissata.

#### Funzionamento:

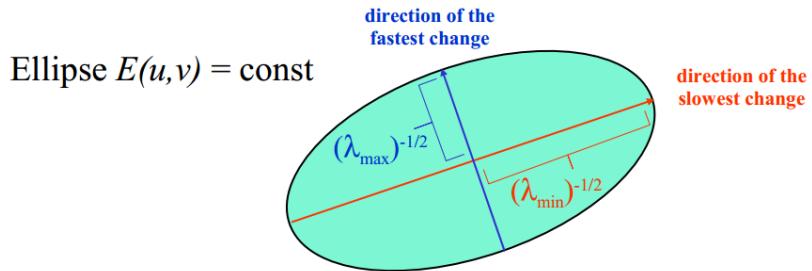
Calcoliamo il gradiente di tutti i pixel  $7 \times 7$  in entrambe le direzioni dopodiché sommiamo tutti i valori nella direzione delle x facendone il quadrato, sommiamo tutti i valori nella direzione delle y facendone il quadrato, poi sommiamo ogni valore della direzione x per la direzione y su tutti i pixel  $7 \times 7$ . Come si può osservare è una **matrice simmetrica  $2 \times 2$** . Il passo successivo è calcolare **gli autovalori** di un sistema  $7 \times 7$  localizzato intorno ad un pixel p danno l'informazione sulla struttura del sistema caratterizzando la **patch** (sottoimmagine in questo caso  $7 \times 7$ ).

Autovalori=

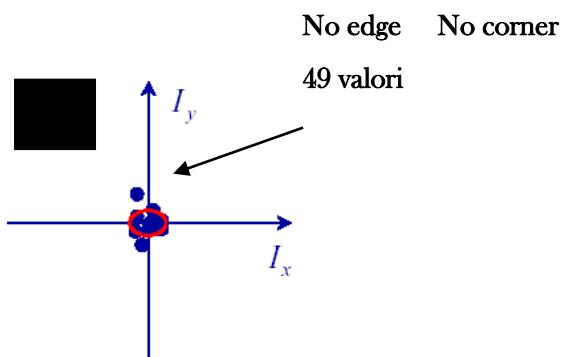
$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Esempio: Quando un tenore fa un acuto e rompe un bicchiere si dice che questo accade quando viene trovato l'autovalore del bicchiere o meglio il punto di rottura del bicchiere.

Gli autovalori in una matrice 2x2 sono **DUE**. Una volta trovati li **ordiniamo** e se il più piccolo è maggiore di una soglia prefissata, il punto p è considerato **un punto di corner**. Gli autovalori rappresentano la lunghezza dei semiassi dell'ellisse che racchiude tutti i punti per i quali stiamo calcolando la matrice di covarianza, in questo caso si tratta di una 7x7 quindi 49 valori.



Se prendiamo una patch 7x7 costante( es. nere), il gradiente della derivata parziale dovrà essere tutti quanti 0.



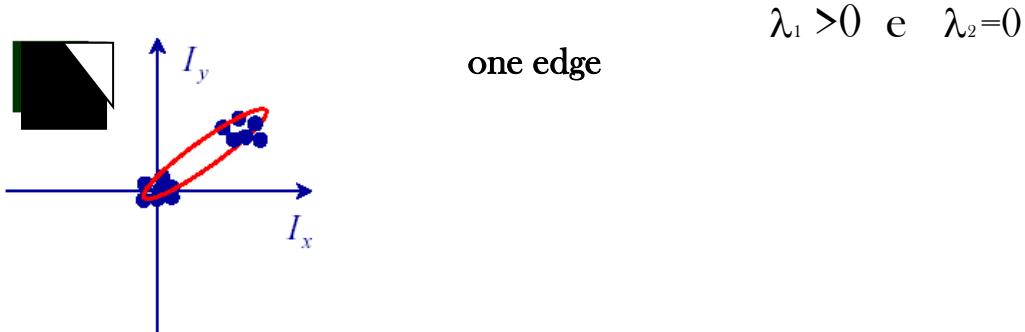
Nota: i 49 valori di pixel sono addensati tutti intorno allo 0, significa che l'ellisse è piccola cos' anche i semi assi. In quanto

$$\lambda_1 = \lambda_2 = 0$$

Se prendiamo una patch in cui c'è una zona nera e una bianca (quindi c'è un edge). Accade che una buona parte di derivate x,y, sono tutti o quasi =0 ( quelle della zona nera) ma ci sarà un altro insieme di pixel ( con gradiente in direzione ortogonale all'edge) che sono calcolate sul bordo.

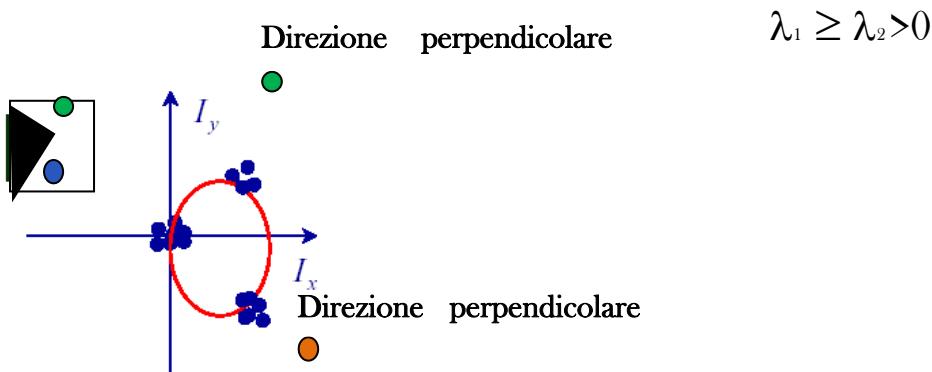
Pertanto  $\lambda_1$  sarà il semi-asse maggiore e  $\lambda_2$  il semiasse minore dell'ellisse dei punti , dunque  $\lambda_2$  sarà certamente < di una certa soglia, quindi non ha un corner ma un edge dato che  $\lambda_1 >$  della soglia.

Quindi i controlli vanno fatti su tutte e due valori di  $\lambda_1$  e  $\lambda_2$  , ovviamente però se  $\lambda_2 < \lambda_1$  e  $\lambda_2 >$  di una certa soglia quindi anche  $\lambda_1 >$  della soglia.



Nel caso di una patch sempre  $7 \times 7$ , ma con due edge=Corner. Per racchiudere tutti questi pixel ci vuole un ellisse più ampia, quindi con asse maggiore e asse minore superiore ad una certa soglia, quindi si ha un punto di corner in p.

Abbiamo un edge per il primo auto valore, un edge per il secondo auto valore, quindi un "corner".



Questo è l'algoritmo di Harris, questo lo si fa per ogni pixel.

Passi dell'algoritmo:

- Calcolare i gradienti dell'immagine
- Per ogni pixel p:
  - Formare la matrice C in un intorno  $(2N+1) \times (2N+1)$
  - Determinare il più piccolo autovalore di C

Se questo autovalore è  $> I$  (soglia), salvare le coordinate del punto p in una lista L

- Ordinare L in ordine decrescente dell'autovalore più piccolo
- Scendere la lista ordinata: per ogni punto attuale p cancellare tutti i punti che appaiono dopo nella lista appartenenti all'intorno di p
- L'output è la lista dei punti di corner che non si sovrappongono
- Nota: per rendere robusto il risultato, applicare un filtro Gaussiano a  $I_x$  e  $I_y$ ,  $I_x^2$ ,  $I_y^2$  prima di calcolare la matrice C

P.s. I corners possono essere rilevati anche se  $R = \det C - k$  (traccia C al quadrato) è maggiore di una certa soglia questo perché il determinante della matrice di covarianza è uguale al **prodotto degli autovalori** e la traccia della matrice di covarianza è uguale alla **somma degli autovalori**, per cui è la stessa cosa cioè stiamo sempre valutando se entrambi gli autovalori sono maggiori di una certa soglia.

# 8. Morfologia matematica

La morfologia ci è utile per estrarre delle componenti di interesse da delle immagini di input e poter quindi rappresentare e descrivere regioni, bordi e superfici. Inoltre si utilizzano delle tecniche morfologiche per realizzare pre o post elaborazioni, come ad esempio il **filtraggio morfologico** o l'assottigliamento.

Il linguaggio della morfologia matematica fa parte della **teoria degli insiemi** e nel nostro caso rappresentano degli oggetti all'interno di un'immagine. Nelle immagini binarie questi insiemi sono membri dello spazio degli interi 2D cioè  $Z_2$ , dove ogni elemento di un insieme è una tupla (vettore 2D) le cui coordinate sono le coordinate  $(x, y)$  di un pixel bianco (o nero) dell'immagine. Le immagini in scala di grigio possono essere rappresentate come insiemi le cui componenti si trovano in  $Z_3$ ; due componenti di ogni elemento dell'insieme fanno riferimento alle coordinate di un pixel mentre la terza corrisponde al valore discreto di intensità.

In morfologia due concetti fondamentali e molto usati sono:

- Riflessione:

indicata con  $\hat{B}$  è definita:

$$\hat{B} = \{w | w = -b, \text{ per } b \in B\}$$

in sostanza se  $B$  è l'insieme dei pixel che rappresentano un oggetto in un'immagine, allora  $\hat{B}$  è l'insieme dei punti in  $B$  le cui coordinate  $(x, y)$  sono state sostituite da  $(-x, -y)$ .

- Traslazione:

di un insieme  $B$  tramite un punto  $z = (z_1, z_2)$  e indicata con  $(B)_z$  è definita:

$$(B_z) = \{c | c = b + z, \text{ per } b \in B\}$$

in sostanza se  $B$  è l'insieme dei pixel che rappresentano un oggetto di un'immagine, allora  $(B)_z$  è l'insieme dei punti in  $B$  le cui coordinate  $(x, y)$  sono state sostituite da  $(x+z_1, y+z_2)$ .

## 8.1 Erosione e Dilatazione

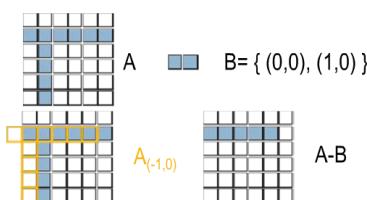
Le due operazioni primitive su cui si basano tutti gli algoritmi che vedremo sono:

1. **Erosione (o Differenza di Minkowski)** : questa operazione di filtraggio morfologico viene utilizzata per eliminare o assottigliare gli oggetti in un'immagine binaria. Matematicamente parlando l'erosione è definita:

$$A \ominus B = \{ c \in E^n \mid c+b \in A, \text{ per ogni } b \in B \}$$

$$A \ominus B = \cap_{b \in B} A_b$$

$A \ominus B$  è definita quindi come: l'insieme dei pixel  $c$  del piano immagine ( $E$ ) tale che  $c+b$  sia un elemento di  $A$  per ogni  $b$  appartenente a  $B$ , equivalentemente l'erosione può essere definita come l'intersezione delle traslazioni di  $A$  rispetto a  $-b$  per ogni posizione  $b$  dell'elemento strutturante  $B$ .  
Esempio:



Data l'immagine A, e elemento strutturante B. L'erosione di A rispetto a B equivale all'intersezione delle traslazioni di A rispetto a B. Cioè la traslazione di A rispetto a (0,0) consiste con l'immagine originale e la traslazione dell'immagine A rispetto a (-1,0) consiste nello spostare l'immagine A di una colonna a sinistra, quindi l'intersezione delle due immagini consiste nei pixel presenti in entrambe le traslazioni quindi A-b consiste nei soli pixel di cui sopra.

## 2. Dilatazione (o Somma di Minkowski):

questa operazione di filtraggio morfologico viene utilizzata per accrescere e ispessire gli oggetti in un'immagine binaria. Matematicamente parlando, la dilatazione è definita:

$$A \oplus B = \{ c \in E^n \mid c = a + b, a \in A, b \in B \}$$

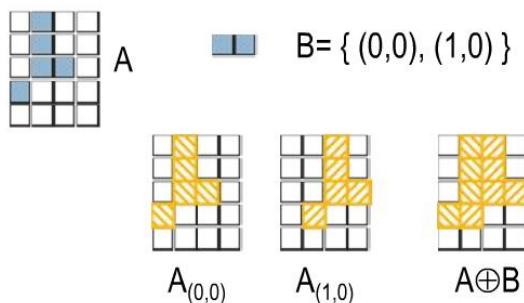
$$A \oplus B = \bigcup A_b, b \in B$$

Si dimostra che:

$$A \oplus B = B \oplus A$$

, cioè che vale la proprietà commutativa.

$A \oplus B$  è definita come i punti c del piano immagine E tale che  $c = a + b$  per ogni a appartenente ad un oggetto A ed ogni b appartenente ad un oggetto B. Equivalentemente la dilatazione può essere definita come l'unione delle traslazioni di A rispetto a B.



Consideriamo l'oggetto A e l'oggetto B. La traslazione di A nella posizione (0,0) consiste con l'immagine immutata mentre la traslazione dell'immagine A rispetto alla posizione (1,0) consiste nello spostare l'immagine A di una colonna in avanti quindi, l'unione delle due traslazioni consiste in tutti i pixel appartenenti sia alla prima traslazione che alla seconda.

## 8.2 Apertura e Chiusura

Altre due importanti operazioni morfologiche derivanti dalla dilatazione e dall'erosione sono:

### 1. Apertura:

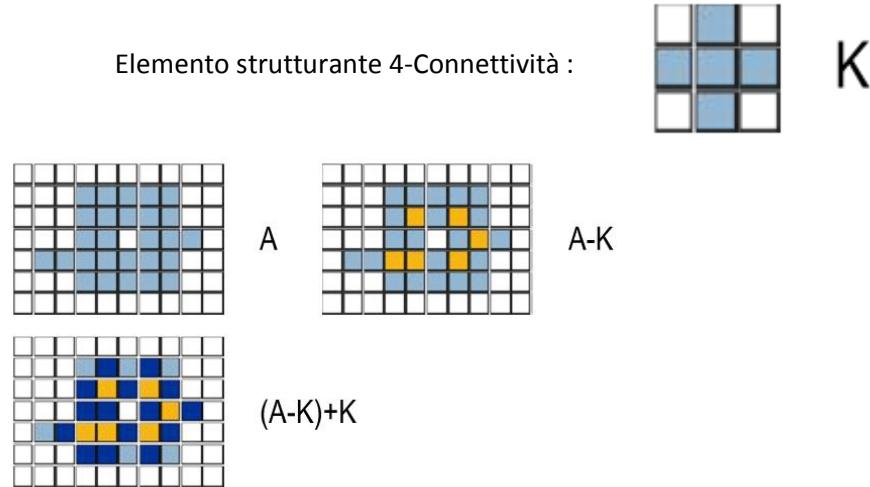
rende più omogenei i contorni di un oggetto, elimina le piccole interruzioni e le protuberanze sottili. L'apertura è definita come  $O(A,K)$  dove K è l'elemento strutturante e A è l'immagine originale. Viene indicata come Dilatazione rispetto a K dell'Erosione di A rispetto a K.

$$O(A, K) = (A - K) + K$$

$$\square O(O(A, K), K) = O(A, K) \subseteq A$$

Vale che l'Apertura rispetto a K di un Apertura di A rispetto a K non cambia il risultato cioè vale l'**Idempotenza** e l'Apertura non è altro che un sottinsieme di A.

Esempio:



- **A** è l'immagine originale rappresentata dai pixel celesti sul piano immagine bianco.
- **K** è l'elemento strutturante a croce.
- **A-K** è l'Erosione di A rispetto a K, quindi si applica l'elemento strutturante pixel per pixel all'immagine originale facendo cancellare solo il pixel che corrisponde al pixel centrale dell'elemento strutturante qualora i 4 pixel vicini siano compresi nell'immagine A.
- **(A-K)+K** indica la Dilatazione del risultato del passo precedente. Questo procedimento fa sì che vengano considerati validi solo i pixel adiacenti 4-connessi al pixel arancio (risultato dell'erosione di A rispetto a K) che vengono denotati con il colore blu scuro.

## 2. **Chiusura**:

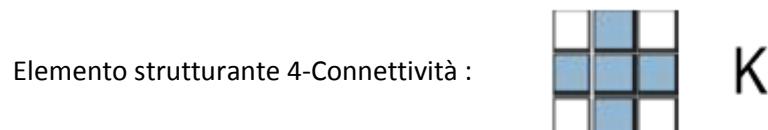
rende più omogenee le sezioni del contorno, fonde le interruzioni sottili, elimina piccoli vuoti, riempie vuoti nel contorno. La chiusura è definita come **C(A,K)** dove **K** è l'**elemento strutturante** e A è l'immagine originale. Viene indicata come l'**Erosione** rispetto a K della **Dilatazione** di A rispetto a K.

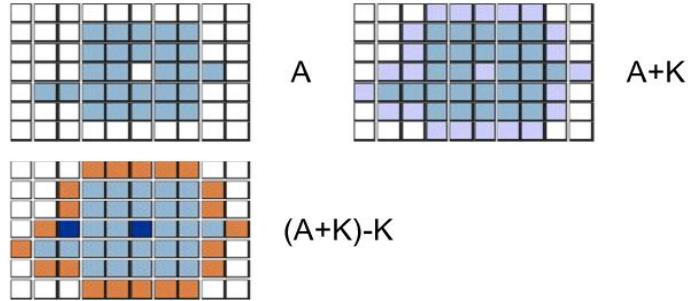
$$C(A, K) = (A+K)-K$$

$$\square A \subseteq C(A, K) = C(C(A, K), K)$$

Vale che A è un **sottinsieme** della **Chiusura** di A rispetto a K ed inoltre che applicare la Chiusura rispetto a K di una Chiusura di A rispetto a K non altera il risultato cioè vale l'**Idempotenza**.

Esempio:





- **A** è l'immagine originale rappresentata dai pixel in celeste sul piano immagine bianco.
- **K** è l'elemento strutturante rappresentato da una croce in celeste.
- **A+K** indica la **Dilatazione** di **A** rispetto a **K**. Cioè dilatare equivale ad aggiungere pixel secondo la struttura **K** laddove esiste una corrispondenza tra la posizione centrale di **K** e un pixel celeste di **A**. In altre parole si posiziona l'elemento strutturante **K** facendo corrispondere il pixel centrale di **K** con ogni pixel dell'immagine **A**. Il risultato di questa operazione sarà la Dilatazione di **A** rispetto a **K** cioè **A+K**. I pixel aggiunti sono denotati con il colore violaceo.
- **(A+K) – K**, si applica l'Erosione del risultato del passo precedente rispetto a **K**. Cioè sopravvivono solo quei pixel per i quali vi è una corrispondenza di tutti i pixel dell'elemento strutturante **K**. In altre parole se la croce di pixel che rappresenta, in questo caso, l'elemento strutturante è contenuta completamente nella Dilatazione del passo precedente allora il pixel centrale è un pixel valido. I pixel validi aggiunti all'immagine sono denotati con il colore blu scuro.

### 8.3 Trasformata hit-or-miss

Si definisce Trasformata hit-or-miss di un immagine originale **A** rispetto a due operatori **J** e **K** come l'intersezione dell'Erosione di **A** rispetto a **K** intersecato all'Erosione del complemento di **A** rispetto a **J** con il vincolo che i due operatori non abbiano nulla in comune, ovvero la loro intersezione dia un insieme vuoto.

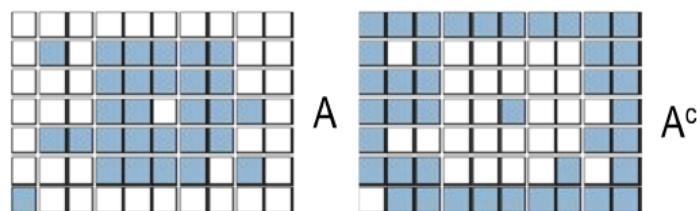
$$A \otimes (J, K) = (A - J) \cap (A^c - K)$$

■ con il vincolo  $J \cap K = \emptyset$

Esempio:

#### Ricerca di punti isolati (4-connessi)

Elementi strutturanti:



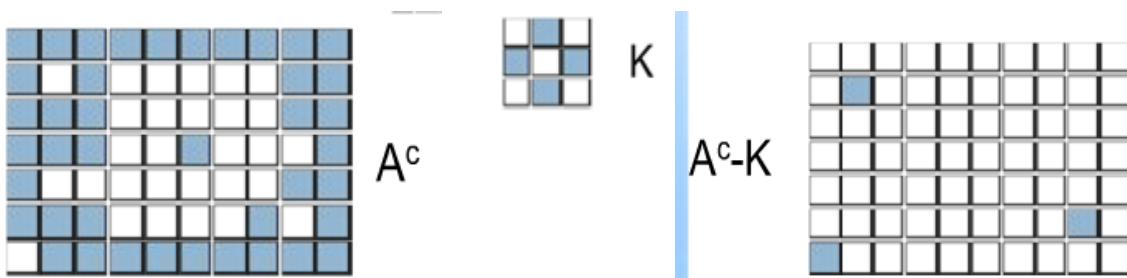
- A è l'immagine originale definita dai pixel celesti sul piano immagine bianco.
- $A^c$  è il complemento di A, quindi tutti i pixel che non appartengono ad A.
- J è l'elemento strutturante composto da un solo pixel celeste
- K è un altro elemento strutturante compreso da i pixel a croce celesti tranne il centrale

L'intersezione dei due elementi strutturanti da un insieme vuoto.

La prima operazione che si deve effettuare è l'Erosione di A rispetto a J. Dato che l'elemento strutturante J è composto da un solo pixel, questo matcherà perfettamente su tutti i pixel dell'immagine A, quindi l'erosione di A rispetto a J è uguale all'immagine A.

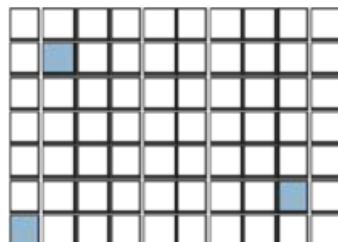
$$A-J=A$$

La seconda operazione da effettuare è l'Erosione del complemento di A rispetto a K.



Che da come risultato l'immagine di destra composta dai soli pixel che hanno matchato perfettamente con l'elemento strutturante.

L'ultimo passaggio è effettuare l'intersezione dei due risultati ovvero l'immagine A che è il risultato dell'erosione di A rispetto a J e l'ultima immagine ottenendo come risultato:



Poiche i pixel sono compresi in A.

Finora gli elementi strutturanti sono stati considerati in modo separato ma possono anche essere visti come un'unica maschera contenente tre tipi di valori:

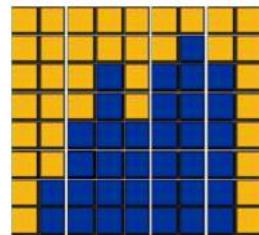
- Punti necessariamente di immagine
- Punti di background
- Punti non rilevanti



M è la maschera dell'elemento strutturante unico.

Analizziamo ora un esempio utilizzando come elemento strutturante la maschera unica.

Immagine originale:

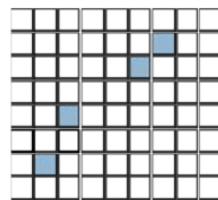


Elemento strutturante M:



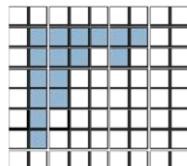
In giallo troviamo i pixel che indicano lo sfondo e in blu i pixel che indicano l'immagine.

Il primo passo è effettuare un match tra l'elemento strutturante e l'immagine originale, ottenendo come risultato:



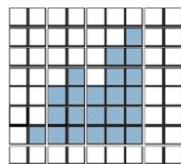
I pixel in azzurro sono i pixel centrali alla maschera dove il match è stato perfetto tra pixel di sfondo e pixel di immagine.

Possiamo trovare i pixel che hanno un matching con i pixel che indicano lo sfondo, quindi solo con i pixel gialli della maschera e otteniamo:



Pixel compatibili con la maschera di background

Possiamo effettuare lo stesso con i pixel dell'immagine quindi quelli blu scuro nell'elemento strutturante, ottenendo:



Pixel compatibili con la maschera per l'immagine

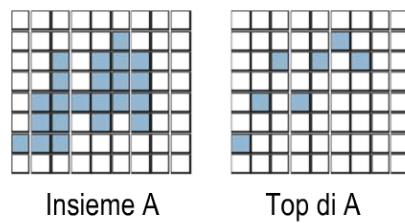
### 8.3.1 Umbra

Dati  $A \subseteq E^n$ ,  $F \subseteq E^{n-1}$ ,  $x \in F$ ,  $y \in E$  definiamo:

**Top di un insieme A ( $T[A]:F \rightarrow E$ ):**  
 $T[A](x) = \max \{ y \mid (x, y) \in A \}$

Cioè, Top di un insieme A una funzione che associa un elemento di F ad un elemento di E tale che il Top di A di un elemento X è uguale al massimo di Y tali che le coppie (x,y) cioè quel pixel appartengono ad A.

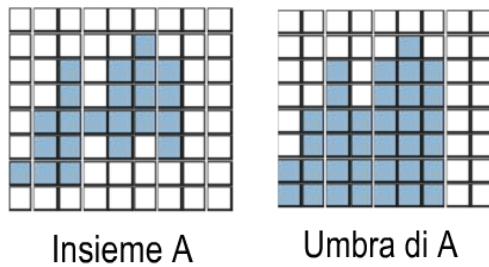
In altre parole il Top di un insieme A sono i pixel caratterizzati dal massimo valore di ordinata appartenenti all'immagine originale. Esempio:



Mentre, definiamo:

**Umbra di f ( $f:F \rightarrow E$ ):**  
 $U[f] = \{ (x, y) \in F \times E \mid y \leq f(x) \}$

Cioè, definiamo Umbra una funzione che associa ad ogni elemento di F un elemento di E tale che l'Umbra di f sono quei pixel appartenenti al prodotto cartesiano di F e E tale che la loro ordinata sia minore o uguale ad f(x). In altre parole calcolare l'umbra equivale a riempire le colonne al di sotto del Top di A. Esempio:



#### 8.3.1.1 Proprietà

$$T[A] \subseteq A \subseteq U[A] \subseteq E^n$$

$$U[U[A]] \equiv U[A]$$

Il Top di A è un sottinsieme di A che a sua volta è un sottinsieme dell'umbra di A che è un sottinsieme del piano immagine. Vale la proprietà di idempotenza, cioè calcolare l'umbra dell'umbra di A non altera il risultato.

## 8.5 Morfologia in scala di grigio

### 8.5.1 Dilatazione

Dati:  $F, K \subseteq E^{n-1}$ ,  $f: F \rightarrow E$ ,  $k: K \rightarrow E$

Si definisce dilation di  $f$  tramite  $k$

$$f \oplus k = T\{U[f] \oplus U[k]\}$$

Cioè si definisce Dilatazione il Top della dilatazione dell'umbra di  $f$  tramite l'umbra di  $k$ .

Si dimostra che una definizione equivalente è:

$$(f \oplus k)(x) = \max\{f(x-z) + k(z) \mid z \in K, x-z \in F\}$$

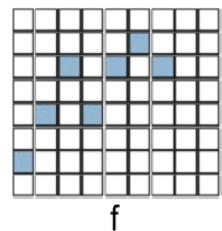
Cioè che la dilatazione di  $f$  rispetto a  $k$  in un punto  $x$  è uguale al massimo della somma della funzione  $f$  in  $x-z$  più  $k$  in  $z$ , dove  $k$  è l'elemento strutturante,  $z$  è un elemento appartenente a  $K$  e  $x-z$  appartiene a  $F$ .

#### Complessità computazionale

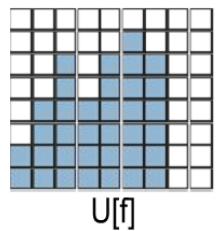
Dal punto di vista computazionale la complessità equivale ad una convoluzione quindi è direttamente proporzionale al numero di pixel dell'immagine dove il fattore di proporzionalità è l'elemento strutturante.

#### Esempio:

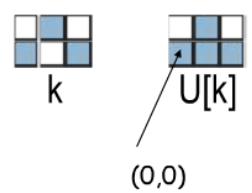
Immagine  $f$ :



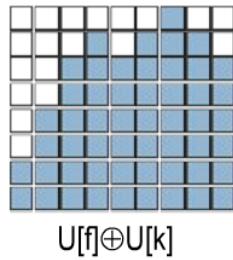
Calcoliamo l'Umbra di  $f$ :



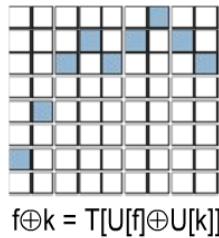
Elementi strutturanti: L'Umbra di  $k$  ha il pixel di riferimento  $(0,0)$  indicato dalla freccia.



Calcoliamo la dilatazione dell'umbra di  $f$  rispetto all'umbra di  $k$ :



E infine ne calcoliamo il Top:



Un modo alternativo per calcolare la Dilatazione è implementare direttamente la formula per esempio con  $x = 6$ :

$$\begin{aligned} (f \oplus k)(6) &= \\ \max\{f(6-0)+k(0), f(6-1)+k(1), f(6-2)+k(2)\} &= \\ \max\{f(6)+k(0), f(5)+k(1), f(4)+k(2)\} &= \\ \max\{5+0, 6+1, 5+0\} &= 7 \end{aligned}$$

$x$	$f$	$k$	$f \oplus k$
0	1	0	1
1	3	1	3
2	5	0	5
3	3		6
4	5		5
5	6		6
6	5		7
7			6
8			5

Il risultato (7) determinerà il valore di  $Y$  quindi verrà colorato in celeste il pixel caratterizzato dalle coordinate (6,7).

### 8.5.2 Erosione

Dati:  $F, K \subseteq E^{n-1}$ ,  $f: F \rightarrow E$ ,  $k: K \rightarrow E$

Si definisce erosione di  $f$  tramite  $k$   
 $f \ominus k = T\{U[f] \ominus U[k]\}$

Cioè si definisce Erosione di  $f$  rispetto a  $k$  il Top dell'Erosione dell'Umbra di  $f$  rispetto all'Umbra di  $k$ .

Si dimostra che una definizione equivalente è:

$$(f \ominus k)(x) = \min\{f(x+z)-k(z) \mid z \in K, x+z \in F\}$$

Cioè che l'Erosione di  $f$  rispetto a  $k$  in  $x$  è uguale al Minimo di  $f$  calcolato in  $x+z$  meno  $k(z)$  dove  $z$  è in  $K$  e  $x+z$  è in  $F$ .

Esempio applicando la formula con  $x=5$ :

$$\begin{aligned}(f \ominus k)(5) &= \\ \min\{f(5-(-1))-k(-1), f(5-0)-k(0), f(5-1)-k(1)\} &= \\ \min\{f(6)-k(-1), f(5)-k(0), f(4)-k(1)\} &= \\ \min\{5+1, 6-0, 5+1\} &= 6\end{aligned}$$

x	f	k	f-k
-1		-1	
0	1	0	
1	3	-1	2
2	5		4
3	3		3
4	5		4
5	6		6
6	5		
7			

Il risultato (6) determinerà il valore di  $Y$  quindi verrà colorato in celeste il pixel caratterizzato dalle coordinate (5,6).

### 8.5.3 Apertura e Chiusura

Effettuando prima l'operazione di Dilatazione e poi di erosione possiamo ottenere la Chiusura mentre viceversa otteniamo l'Apertura.

## 8.6 Trasformata distanza

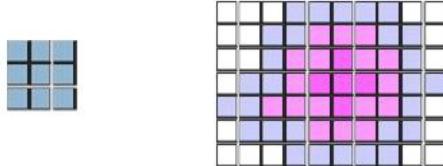
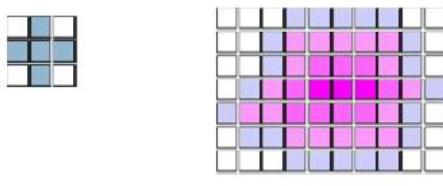
La trasformata distanza è una trasformata che a partire da un'immagine binaria produce in uscita un'immagine in scala di grigio poiché il valore di livello di grigio attribuito ad ogni pixel non è altro che la distanza che il pixel ha dallo sfondo pertanto la distanza può essere etichettata da un numero che indica le posizioni dallo sfondo e questo rappresenta le sfumature di grigio. La distanza dipende dal tipo di connessione che si assume per la figura cioè 4-connessione o 8-connessione. La trasformata di stanza può essere vista come un'applicazione delle operazioni di morfologia matematica e in particolare come successione di Erosione e somma dell'immagine ottenuta tenendo conto dell'elemento strutturante scelto inizialmente.

Esempio di funzionamento:

```
R = ∅  
while(A<>∅) do  
    R = R+A  
    A = R⊖K  
done
```

Se  $R$  è un insieme vuoto, finché  $A$  è non vuoto si somma ad  $R$  l'immagine  $A$  e ad  $A$  viene assegnato l'Erosione di  $R$  rispetto ad  $K$  (evento strutturante scelto)

Esempio figurato:



Quello che facciamo è applicare l'elemento strutturante sull'immagine ed effettuare di volta in volta un'erosione rispetto all'elemento strutturante ottenendo un insieme di distanze per ogni posizione dei pixel rappresentate da colori, cioè il rosa scuro indica la distanza maggiore dallo sfondo che si schiarisce via via che la distanza diminuisce. Quello che si può notare nelle due diverse applicazioni con elementi strutturanti diversi (a croce o 8-connesso) si ottiene una 8-connettività con l'elemento strutturante a croce mentre una 4-connettività con l'elemento strutturante 8-connesso.

Al fine di descrivere gli algoritmi per il calcolo della DT consideriamo la seguente notazione per la rappresentazione dei vicini di un pixel  $p$  preso in considerazione:

$$\begin{array}{ccc} n_2 & n_3 & n_4 \\ n_1 & p & n_5 \\ n_8 & n_7 & n_6 \end{array}$$

## Algoritmo sequenziale per il calcolo della DT

La DT può essere calcolata mediante due scansioni in cui vengono elaborati solo i pixel dell'oggetto . Nella scansione diretta si procede dall'alto verso il basso e da sinistra verso destra, nella scansione inversa dal basso verso l'alto e da destra verso sinistra. Assumendo come metrica D4:

1. *Scansione Diretta:*  $d(p) = \min(n_1, n_3) + 1$
2. *Scansione Inversa:*  $d(p) = \min(n_5 + 1, n_7 + 1, d(p))$

In altre parole

- La prima dall'alto al basso e da sinistra a destra
  - ogni pixel della figura è uguale al valore minore fra i **vicini** già considerati incrementato di 1
- La seconda dal basso all'alto, da destra a sinistra
  - ogni pixel della figura è uguale al minimo fra i **vicini** già considerati incrementato di 1 ed il pixel stesso

Nel caso della metrica D8 `e necessario considerare i pixel n1, n2, n3, n4 nella scansione diretta ed i pixel n5, n6, n7, n8 nella scansione inversa.

Nella scansione diretta ogni pixel viene etichettato con un valore via via sempre maggiore man mano che la scansione avanza assegnando distanze errate ai pixel in basso pertanto necessita di un **fattore di correzione** che risulta essere la **scansione inversa**.

Esempio:

Applicando la scansione diretta con un elemento strutturante a croce otteniamo:

		1	1	1	1	1		
	1	2	2	2	2	2	1	
	1	2	3	3	3	3	2	
	1	2	3	4	4	4	4	2
1	2	3	4	5	5	5	3	2
1	2	3	4	5	6	4	3	
	1	2	3	4	5	6	4	3
	1	2	3	4	5	6	4	3
		1	2	3	4	5		

Quello che si può notare è che le distanze aumentano man mano che si arriva in basso a destra della figura. Per correggere questo difetto si applica una scansione inversa al risultato ottenuto ottenendo:

Le distanze (numeri) che visualizziamo rappresentano le distanze di un pixel dallo sfondo secondo la 4-connettività.

Esempio 8-connettività:

		1	1	1	1	1		
	1	2	2	2	2	2	1	
	1	2	3	3	3	3	2	1
	1	2	3	4	4	4	4	3
1	2	3	4	3	3	3	2	1
1	1	2	2	2	2	2	2	1
1	1	1	2	2	2	1	1	
	1	1	1	1	1	1		
		1	1	1	1	1		



		1	1	1	1	1		
	1	1	2	2	2	1	1	
	1	2	2	3	2	2	1	
	1	1	2	3	3	2	1	
1	1	2	2	3	4	3	2	1
1	2	3	3	4	3	3	2	1
1	1	2	2	3	4	3	3	1
	1	2	3	4	2	1	1	
		1	2	3	4	2	1	

		1	1	1	1	1		
	1	1	2	2	2	1	1	
	1	2	2	3	2	2	1	
	1	1	2	3	3	2	1	
1	1	2	2	3	4	3	2	1
1	2	3	3	4	3	3	2	1
1	1	2	2	3	4	3	3	1
1	1	1	2	2	2	1	1	
	1	1	1	1	1	1		

## Algoritmo parallelo per il calcolo della DT

La DT puo' essere calcolata realizzando il processo di propagazione in parallelo. Anche in questo caso vengono elaborati solo i pixel dell'oggetto, che modificano la loro etichetta ad ogni iterazione fino a che questa non raggiunge il valore della distanza. Il processo termina quando nessun pixel modifica più la propria etichetta.

Assumendo come metrica D4:

$$d(p)^{i+1} = \min(n_1, n_3, n_5, n_7)^i + 1$$

Nel caso della metrica D8 `e necessario considerare il minimo fra tutti e 8 i vicini n1 . . . n8.

### Trasformata Distanza Inversa (RDT)

La trasformata distanza inversa consente di costruire a partire da un pixel etichettato con il valore di distanza d il "cerchio" di "diametro"  $2d-1$  contenente tutti i pixel aventi distanza da p minore di d.

Nel caso della metrica D4 l'algoritmo sequenziale per la RDT `e dato da

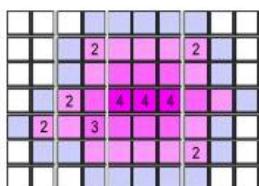
1. *Scansione Diretta:*  $d(p) = \max(n_1 - 1, n_3 - 1, d(p))$
2. *Scansione Inversa:*  $d(p) = \max(n_5 - 1, n_7 - 1, d(p))$

mentre quello parallelo è dato da:

$$d(p)^{i+1} = \max(n_1 - 1, n_3 - 1, n_5 - 1, n_7 - 1, d(p))^i$$

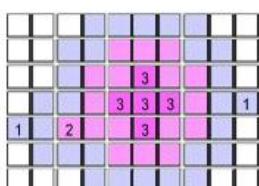
Gli algoritmi per la metrica D8 si ottengono in modo analogo a quanto visto per la DT. La figura seguente mostra la costruzione del cerchio relativo a  $D4 = 3$  mediante l'algoritmo parallelo:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	0	0
0	0	3	0	0	0	2	3	2	0
0	0	0	0	0	0	0	2	0	0
0	0	0	0	0	0	0	0	0	0



### Massimi locali

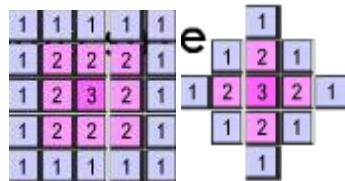
Dopo aver applicato la Trasformata Distanza è possibile far sopravvivere solo i massimi locali calcolati secondo un intorno 3x3 ottenendo:



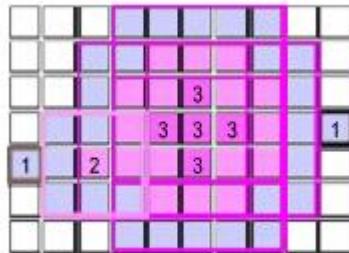
Nel primo caso sopravvivono solo i pixel etichettati con i valori 2-3-4 calcolati con un elemento strutturante a croce mentre nel secondo caso con un elemento strutturante 8-connesso. Questi pixel possono essere considerati come una versione "compatta" dell'immagine cioè è possibile a partire da questi pixel applicando i "dischi massimali" ricostruire l'immagine originale unendo i dischi aventi per dimensione il numero stesso.

## Dischi massimali

Prendendo in considerazione questi elementi strutturanti:



E' possibile effettuare una ricostruzione dell'immagine originale iterando un operazione di Dilatazione tramite l'elemento strutturante utilizzato ottenendo:



A partire dai pixel salvati calcolando i massimi locali.

## Calcolo distanza

Se  $A$  è l'insieme costituito da  $X$ , e  $D=A$ , la distanza utilizzando la trasformata distanza può essere calcolata iterando 2 operazioni finchè  $Y$  non appartiene ad  $A$ . Definiamo la distanza tra due punti:

Questa procedure fa determinare la distanza fra due pixel  $X, Y$  qualsiasi. Seguendo la direzione del massimo gradiente si ottiene anche il percorso minimo (non è detto sia unico).

Vale che se:

$$\text{Se } A \equiv (A \oplus K) \cap F$$

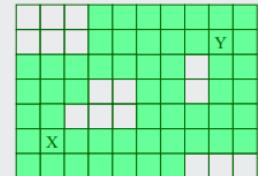
### Distanza fra due punti ( $X, Y$ )

```

 $A = \{ X \}; D = A$ 
while( $Y \notin A$ ) do
     $A = (A \oplus K) \cap F$ 
     $D = D + A$ 
done

```

Seguendo la direzione del massimo gradiente si ottiene anche il percorso più breve  
(non è detto sia unico)



- allora  $Y$  non è raggiungibile da  $X$

- $F$  non è connesso

Il metodo può essere pesante computazionalmente  
(si pensi ad una spirale)

### Trasformata distanza pesata

La trasformata distanza può essere estesa considerando la trasformata distanza pesata, cioè assumendo che la distanza di ogni vicino non è unitaria ma pesata da un peso  $w$ . L'algoritmo sequenziale può essere quindi modificato come segue:

#### Algoritmo sequenziale

- Scansione diretta
  - $val = \min_i \{p_i + w_i\}$  (i vicini **precedenti**)
- Scansione inversa
  - $new-val = \min_k \{p_k + w_k\}$  (k vicini **successivi**)

Il valore attribuito ad un pixel sarà il minimo su i valori attribuiti ai pixel già visitati più un peso  $w_i$ . Il peso può essere scelto come maschera ad esempio:

3	2	3
2		2
3	2	3

### Scheletro

Lo scheletro di  $F$  è un sottoinsieme di  $F$  con le seguenti proprietà

- è connesso se  $F$  è connesso ed ha lo stesso ordine di molteplicità di  $F$
- ha spessore unitario
- è centrato in  $F$
- include i massimi locali di  $F$

# 9. Stima del movimento

Uno degli argomenti più interessanti dell'analisi di immagini e di video in generale è quello di rilevare delle caratteristiche che sono comuni a più oggetti, pixel o oggetti di dimensione maggiore del pixel nelle immagini tenendo conto sia delle caratteristiche spaziali ma anche temporali. Pertanto trattiamo l'analisi di frequenze di immagini, in particolare di una delle caratteristiche peculiari che è data dal **movimento**. Per introdurre il concetto di **movimento** dobbiamo introdurre naturalmente il concetto del **tempo** che si aggiunge alle 2 informazioni spaziali date dall'ascissa e dell'ordinata del pixel all'interno dell'immagine stessa.

Per esempio facendo un calcolo in un'immagine a risoluzione **VGA** (640x480), considerare un tempo di acquisizione di **25 fotogrammi al sec.** Quello che si può fare come calcolo è che se abbiamo 3 bande di colore per ogni immagine (esempio RGB) e quindi 3 immagini per ognuna, abbiamo **3 x 640 x 480 x 25** il numero di pixel da elaborare **al secondo** il cui spazio di memorizzazione risulta essere uguale a circa **23MB**, considerando la memorizzazione di 30 secondi di sequenza video questo richiede circa una quantità di memoria pari a 700MB. L'esigenza della compressione video è evidente ma l'esigenza di rilevare quanto di più caratteristiche peculiari vi siano all'interno della sequenza video lo è altrettanto (**stima del movimento**). Tipicamente l'evoluzione temporale sulla scena non dipende semplicemente dal moto degli oggetti, poiché si hanno dei moti dovuti anche ai cambiamenti del moto relativo all'osservatore (**Egomotion**), ovvero la camera. Per cui cambiando la scena da rilevare cambia anche il movimento apparente dell'oggetto. E molto spesso i due effetti sono combinati. Naturalmente rilevare il movimento equivale anche poi a permettere di effettuare il **tracciamento** dell'oggetto, inseguendo il movimento dell'oggetto e quindi l'**evoluzione temporale** dello stesso.

E' opportuno però tenere in considerazione il fatto che le velocità rilevate sono delle **velocità apparenti**, cioè non sono le velocità effettive, ma una **stima** delle velocità effettive. Tipicamente quello che noi riprendiamo dell'oggetto nello spazio tridimensionale è soltanto la sua proiezione sul piano immagine, pertanto ad una velocità  $v_0$  relativa al punto  $p_0$  rileviamo la velocità  $v_i$  del punto  $p_i$  che non è altro che la proiezione del punto  $p_0$  nello spazio tridimensionale sul piano immagine bidimensionale.

## 9.1 Sottrazione dello sfondo

Un modo alternativo per stimare il movimento è non stimare lo spostamento. Mentre il flusso ottico che è un campo di insiemi di vettori spostamento cioè per ogni pixel io so di quanto si è spostato e in che direzione, nel caso del problema delle corrispondenze ho lo spostamento ma non tutti i punti ma solo di punti caratteristici.

Con la sottrazione dello sfondo ho una stima del movimento ma non degli spostamenti che hanno subito i pixel, cioè voglio sapere cosa si è spostato ma non di quanto. **A cosa serve?**

**Serve per stimare il modo globale del movimento.**

Un modo banale di verificare la stima è quella di effettuare la sequenza video e poi prendere un frame al tempo  $T$  e fare la differenza pixel x pixel con il frame al tempo  $T-1$ .

Se due frame consecutivi producono una differenza diversa da 0 si può concludere che qualche cosa si è spostato, ma non so in che direzione e di quanto, ma solo che si è spostato.

Risultato della cosa è che se io ho 2 frame adiacenti e faccio tale differenza quello che può capitare è che se io abbia un problema che si chiama ghost ( ovvero fa vedere parti di pixel che non sono comuni ne al tempo  $t$  ne al tempo  $t+1$ ).

Ora il problema è cosa è lo sfondo?

Considero lo sfondo come la media dell'immagine, considerando uno sfondo medio quindi mi aspetto di trovare le parti comuni.

Quindi l'algoritmo è formato così:

- fase di "boot strapping", prendere un insieme di immagini e fare la media e decidere che tale media è lo sfondo.
- Ogni frame successivo verrà sottratto dallo sfondo di conseguenza quello che è il movimento verrà etichettato come bianco e quello che non è in movimento come nero.

- Il frame viene inglobato nello sfondo , il frame successivo viene sottratto al nuovo sfondo poi viene inglobato nel nuovo sfondo e quindi si identificano 2 fasi DETECTION rilevamento del foreground facendo la sottrazione dello sfondo e il calcolo del nuovo sfondo , dunque calcolo lo sfondo ad ogni passo, ricalcolando così il nuovo sfondo come la media tra il frame attuale e lo sfondo calcolato precedentemente che è a sua volta media del precedente.

Calcolo:

alfa è il peso per il BG al tempo T

$$\alpha t+1 = \alpha u t + (1-\alpha)z t+1$$

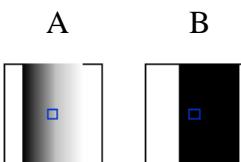
## 9.2 Il flusso ottico

**Quello che noi rileviamo in termine di velocità è diverso da quello che è effettivamente la velocità reale.** Il campo vettoriale della variazione di luminosità che rappresenta poi la velocità apparente, viene denotato come **flusso ottico**.

**Def:**

**Il flusso ottico è definito come il campo vettoriale della variazione di luminosità.**

Se ad esempio abbiamo un oggetto :



Se facessimo ruotare A esso diventa B cioè passa dal colore chiaro al colore scuro, variando così la luminosità, rilevando quindi un movimento.

Se invece lo facessimo traslare verticalmente non cambia colore rimane ancora chiaro e quindi non c'è stata variazione di luminosità, pertanto non avremmo rilevato alcun movimento

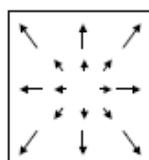
Concludiamo quindi, che la variazione di luminosità è indice di variazione di velocità.

**Pertanto la variazione puntuale del livello di grigio è legata sia alla velocità del moto che al gradiente**

Viene rilevato variazione di velocità solo se la direzione dell'oggetto è parallelo al gradiente, se invece è perpendicolare al gradiente allora non è rilevata alcuna variazione della velocità dell'oggetto.

**Idealmente:**

il flusso ottico dovrebbe essere quel campo vettoriale prodotto da un oggetto che si avvicina con velocità uniforme all'osservatore. Quindi se volessi rilevare il movimento di un oggetto che si avvicina all'osservatore basta guardare l'intensità del campo vettoriale che aumenta man mano che l'oggetto si avvicina.



**Fuoco di espansione:**

quando effettuiamo uno zoom out, cioè o l'oggetto si avvicina all'oggetto, o la camera si avvicina all'oggetto, lo zoom viene fatto lungo la perpendicolare al piano immagine.

**Fuoco di contrazione:**

procedimento inverso, ci allontaniamo dall'oggetto, cambia l'intensità del campo vettoriale, dove l'intensità è decrescente.

### 9.2.1 Ipotesi fondamentale

il problema è quindi determinare il flusso ottico , ovvero stimare correttamente le velocità reali.

Per l'applicazione del flusso ottico devono essere verificate due ipotesi fondamentali:

- 1) la luminosità della scena **dovrà rimanere costante** (poiché la variazione di luminosità potrebbe essere interpretato come uno spostamento)
- 2) il punto in esame rispetto al quale sto misurando l'eventuale spostamento, deve rimanere **resta in vista**, cioè tra un frame e un altro il punto deve rimanere all'interno nel piano immagine, per poter rilevare l'eventuale spostamento.

Per effettuare il **calcolo del flusso ottico** , v'è ipotizzato una variazione dell'intensità luminosa non solo nello spazio  $2d(x,y)$  ma anche nel tempo  $t$ , quindi l'immagine sarà rappresentata come:

$$f(x,y,t)$$

Verifica dell'ipotesi:

### 9.2.2 Calcolo del flusso ottico

Se la luminosità deve rimanere costante, cioè il punto deve spostarsi ma deve portare con sé la stessa luminosità (ipotesi abbastanza forte poiché non tiene conto di rumore, fluttuazione nella ripresa e quant'altro..). Il punto si sposta di una quantità  $x$  e  $y$ , rappresentato da  $\delta x$  e  $\delta y$  nel tempo quindi in un frame successivo o più frame successivi e che quindi si trovi ad un istante temporale pari a  $t+\delta t$  e si ipotizza che porti con sé la luminosità.

Questo si può rappresentare attraverso la relazione:

$$f(x+\delta x, y+\delta y, t+\delta t) = f(x,y,t)$$

Cioè la luminosità del punto nella posizione  $x,y,t$ , quindi nella posizione  $x,y$  del piano immagine al tempo  $t$  è la stessa luminosità del punto che si è spostato nel piano immagine relativo ad un frame all'istante  $t+\delta t$  di un  $\delta x$  e/o di un  $\delta y$ , da cui sviluppando in serie di Taylor rispetto a  $f(x,y,t)$  si ottiene:

$$f(x,y,t) + f_x \delta x + f_y \delta y + f_t \delta t + \epsilon = f(x,y,t)$$

con  $f_x, f_y, f_t$  derivate prime di  $f$  rispetto a  $x, y$  e  $t$  e con  $\epsilon$  derivate seconde, terze e via dicendo..

Sottraendo  $f(x,y,t)$  entrambi membri e poniamo  $\epsilon=0$  otteniamo un'equazione, per lo spostamento spaziale  $\delta x, \delta y, \delta t$ . L'equazione che si ottiene è  $f_x \delta x + f_y \delta y + f_t \delta t = 0$

se dividiamo ancora tutto per  $\delta t$ , dove per comodità indichiamo con  $u = (\delta x / \delta t)$  e con  $v = (\delta y / \delta t)$  otteniamo, dove  $u$  e  $v$  rappresentano le componenti del flusso ottico, nelle direzioni  $x$  e  $y$ :

$$f_x u + f_y v + f_t = 0$$

osservando che  $u$  e  $v$  non sono altro che la velocità (spostamento nel tempo) rispetto a  $x$  e  $y$ , si ottiene l'equazione del flusso ottico in forma vettoriale:

$$\mathbf{v} * \nabla f = - f_t$$

questa espressione è possibile , dato che il gradiente  $\nabla f$  non è altro che un vettore a componenti:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad \text{mentre il vettore velocità } \mathbf{v} = \begin{bmatrix} u \\ v \end{bmatrix}$$

- Calcolo del flusso ottico quindi equivale a determinare i vettori  $\mathbf{v}$  tale che il prodotto del vettore

$$\mathbf{v}^* \nabla f = -\mathbf{f}_t,$$

dove  $\mathbf{f}_t$  è la derivata della funzione  $f$  rispetto al tempo.

- Per calcolare il gradiente basta applicare uno dei filtri derivativi (es Sobel) nel punto considerato
- Il problema è calcolare proprio le componenti del vettore velocità  $u$ ,  $v$ .

Una considerazione da fare è che se il gradiente è perpendicolare al flusso, la variazione non è percepibile e risulta nulla,  **$\mathbf{v}$  perpendicolare a  $\nabla f$ ,  $\mathbf{v}^* \nabla f = 0$**

Pertanto riusciamo a rilevare soltanto velocità, quindi movimenti che sono avvenuti in modo parallelo alla direzione del gradiente, e questo si può determinare, utilizzando semplicemente il calcolo della componente parallela della velocità  $v$  pari a:

$$v = f_t / \sqrt(f_x^2 + f_y^2)$$

è esattamente questo il limite del flusso ottico, cioè se vogliamo rilevare le variazioni di velocità dell'oggetto, lo possiamo fare solo nella direzione del gradiente, come già detto qualsiasi spostamento perpendicolare al gradiente risulterà **NON RILEVABILE**.

Questo vale se la luminosità è sempre la stessa, quindi è costante.

Se invece la luminosità varia, viene rilevato un movimento apparente ma non è reale.

### 9.2.3 Calcolo completo del flusso ottico:

Per ricavare in modo completo il flusso ottico occorre integrare il modello con ipotesi aggiuntive (in generale si minimizza un funzionale)

$$\iint \{(f_x u + f_y v + f_t)^2 + \lambda [(u^2 x + u^2 y) + (v^2 x + v^2 y)]\} dx dy$$

Dove  $u$ ,  $v$  (componenti vettore velocità)

Ulteriore vincolo:

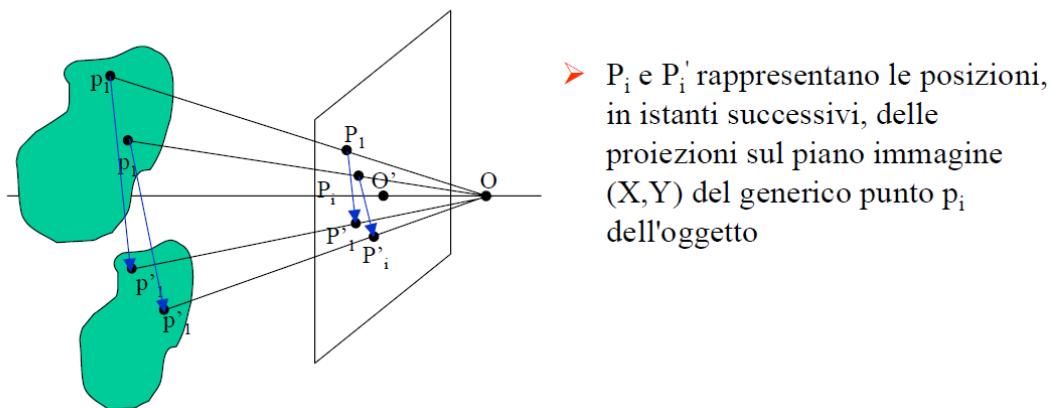
L'oggetto nel suo movimento deve essere regolare, cioè tutti i punti del corpo rigido si muovono di poco nella stessa direzione  $\lambda [(u^2 x + u^2 y) + (v^2 x + v^2 y)]$ . Questo vincolo viene rappresentato, dal fatto che si introduce una regolarizzazione sull'intensità luminosa dei pixel, in particolare la somma dei quadrati delle componenti velocità nella direzione  $x$  e  $y$  siano quante più piccole possibili, questo vincolo può rappresentare un termine da aggiungere ad un equazione da minimizzare,  $f_x u + f_y v + f_t < \epsilon$  al limite pari a 0. Questo rappresenta la **minimizzazione di un funzionale**, posto al quadrato per evitare fluttuazioni e misurare solo l'intensità della variazione.

Il tutto per tutti i punti del corpo rigido rappresentato nel continuo rappresentato con il doppio integrale, variando la  $x$  e  $y$ .

Quindi il problema si riduce a trovare il minimo di questo funzionale, in termine di  $u$  e  $v$ , ovvero delle componenti del vettore velocità, questa tecnica è nota come **l'algoritmo di Lukase Kanade**.

## 9.3 Stime basate su corrispondenze discrete

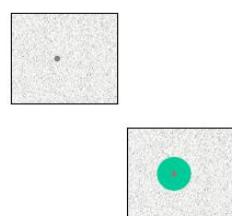
Uno dei metodi alternativi per stimare il movimento è quello basato sulle corrispondenze discrete, cioè sulla corrispondenza di un punto all'istante  $i$  rispetto ad un istante  $i+1$  di una proiezione di punto di un oggetto su di un piano  $(x,y)$ . Siano cioè  $P_i$  e  $P'_i$  le posizioni in istanti successivi delle proiezioni sul piano immagine del generico punto  $p_i$  dell'oggetto è necessario stimare le corrispondenze per rappresentare le discrepanze, di spostamento, che l'oggetto ha subito nello spostarsi da una posizione ad un'altra posizione in istanti differenti.



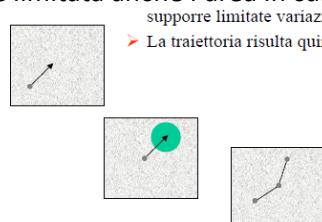
Il problema è reso abbastanza complicato dal fatto che i punti corrispondenti non sono naturalmente noti a priori e vi sono varie proiezioni dei punti dell'oggetto sul piano immagine è necessario pertanto utilizzare una serie di euristiche che guidano la ricerca di queste corrispondenze inoltre è necessario anche ridurre il problema ad un insieme di punti che tipicamente è in numero ridotto rispetto al flusso ottico. Cioè si dice che il **flusso ottico** è una **tecnica** di stima del movimento **denso** cioè tale che per ogni punto si possa sapere la discrepanza o lo spostamento che il punto ha subito nel passare da un istante  $i$  ed un istante  $i+1$ , mentre con il metodo delle **corrispondenze discrete** si ha una **idea di massima** cioè si ha quella che è la stima del movimento per alcuni punti caratteristici dell'oggetto stesso.

Per guidare la ricerca è necessario utilizzare delle conoscenze a priori (procedimenti euristici) che possono essere rappresentate da una serie di proprietà: massima velocità, massima accelerazione, moto comune.

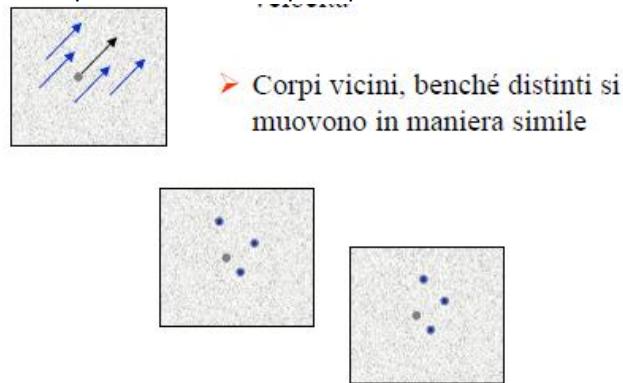
- **Massima velocità:** cioè si può fissare la massima velocità all'interno di un'area in cui si può trovare il punto omologo, tale area risulterà naturalmente limitata. Per esempio, se consideriamo un punto, l'area limitata sarà una circonferenza.



- **Massima accelerazione:** ad una massima velocità può far corrispondere anche una massima accelerazione cioè un'ipotesi che l'oggetto si muove di moto regolare pertanto sono limitate anche le variazioni di velocità. La traiettoria risulta smussata. Naturalmente le variazioni di velocità risultano essere limitate e pertanto è limitata anche l'area in cui viene effettuata la ricerca.



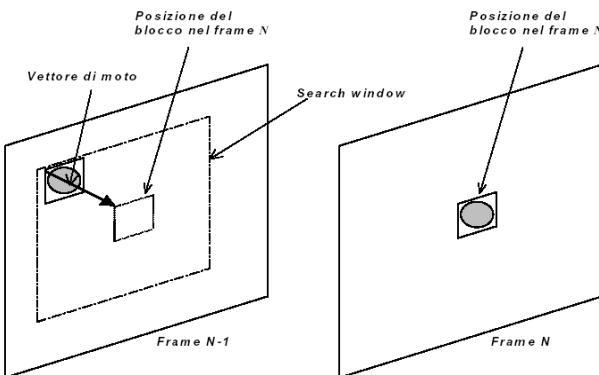
- **Moto comune:** In presenza di corpi rigidi i punti vicini si spostano alla medesima velocità, pertanto ci aspettiamo che i punti o corpi vicini si muovono in maniera simile e quindi se facciamo un'ipotesi per un punto la stessa ipotesi vale anche per i punti che sono nelle immediate vicinanze.



Quindi il moto comune può essere rappresentato dal moto di un singolo oggetto, dove tutti i punti appartenenti ad esso si sposteranno nella medesima maniera, oppure dal moto di un gruppo di corpi rigidi i quali si sposteranno più o meno alla medesima maniera pertanto ci aspettiamo che la direzione, almeno, sia la stessa o pressoché la stessa come lo spostamento.

## 9.4 Algoritmi

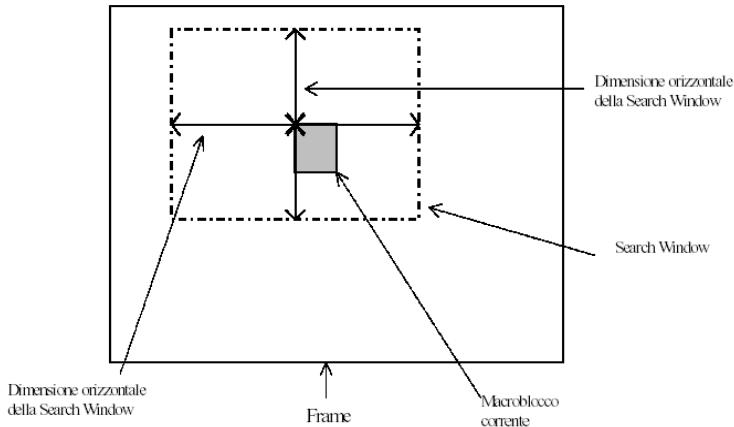
Un metodo per affrontare l'analisi delle corrispondenze si basa sullo spostamento (corrispondenza) di blocchi di pixel. Considerando un blocco che circoscrive un'area di pixel in virtù della massima velocità o massima accelerazione, applichiamo la ricerca su una finestra circoscritta di pixel.



Le proprietà che la finestra di ricerca deve possedere è quella innanzitutto di essere un macroblocco diciamo adiacente, cioè avente le stesse coordinate del blocco da codificare presente nel frame di riferimento ma le sue dimensioni possono variare purché non eccedono le dimensioni di tutta quanta l'immagine.

In generale cmq la finestra di ricerca individua l'area sull'immagine di riferimento all'interno della quale si deve cercare la migliore predizione per un macroblocco. Si parla di codificare perché questa tecnica delle analisi delle corrispondenze è alla base di tecniche abbastanza naif di codifica delle immagini in movimento, cioè di video. Le dimensioni della finestra condizionano le prestazioni dell'algoritmo di analisi delle corrispondenze e quindi anche di un codificatore che cerca di effettuare tale corrispondenza.

Maggiore è l'area più lunga è la ricerca per la migliore predizione. Con il termine "La migliore predizione" intendiamo determinare una corrispondenza tra un macroblocco al frame n-1 ed un macroblocco al frame n, tale corrispondenza viene misurata pixel per pixel, utilizzando delle misure di seguito illustrate.



#### 9.4.1 Algoritmo di scelta

Le più note misure di corrispondenza tra macroblocco sono la **MAD** e la **MSE**

$$\text{MAD} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n |Y(i, j) - Z(i, j)|$$

$$\text{MSE} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (Y(i, j) - Z(i, j))^2 \quad (\text{scarto quadratico medio})$$

Blocco da approssimare      Blocco di riferimento

- La **MAD** misura le distanze tra i pixel appartenenti ai 2 macroblocchi effettuando una **norma 1** tra i pixel, cioè il valore assoluto delle differenze tra i pixel della stessa posizione normalizzato sulla dimensione del blocco indicata nella formula da **mn**
- La **MSE** misura esattamente la stessa distanza soltanto che in **norma 2** ed è indicata anche come scarto quadratico medio Mean Square Error.

**Y** rappresenta il blocco da approssimare, cioè il blocco di cui si effettua la ricerca mentre **Z** è il blocco di riferimento da cui si parte, quindi **Z** il blocco all'istante del frame n-1 e **Y** è un blocco individuato dal frame n all'interno della finestra di ricerca che abbiamo circoscritto.

#### 9.4.2 Algoritmo FULL SEARCH

Esso consiste banalmente nel calcolare la distorsione, cioè la misura di distanza tra due macroblocchi in tutte le possibili posizioni che un macroblocco può assumere all'interno della finestra di ricerca. **Due macroblocchi che differiscono per un solo pixel sia in orizzontale che in verticale sono considerati diversi.** Esempio:

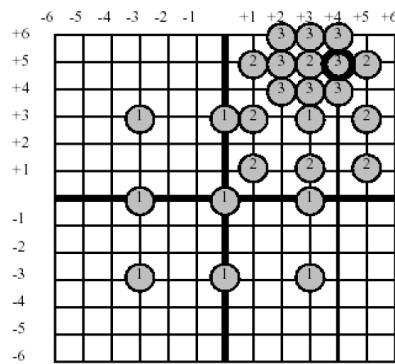
se in una finestra con parametri 20 e 15 (semiampiezze, e quindi le dimensioni della finestra sono 40x30), sono presenti 1200 diversi macroblocchi che possiamo distinguere all'interno di questa finestra, sono necessarie ben 256 sottrazioni + 256 addizioni tutto per  $1200 = 613200$  operazioni.

E' evidente che gli algoritmi di **full search** sono costosi computazionalmente. I nuovi algoritmi calcolano la stessa funzione di distorsione solo per alcune delle posizioni all'interno dell'area di ricerca e poi scelgono quel che è il match migliore, cioè non cercano la corrispondenza 1 a 1 ma la migliore corrispondenza in questo sottoinsieme. Non vengono esaminati tutti i possibili casi per cui la scelta fatta potrebbe non essere quella ottima ma l'algoritmo può fermarsi su un minimo locale di distorsione invece che trovare il minimo globale (o assoluto). Pertanto accettando una condizione sub-ottimale cioè quella che determina il macroblocco corrispondente ad un minimo locale piuttosto che ad un minimo globale otteniamo una soluzione non ottimale ma ad un tempo computazionale più ridotto. **Si assume cioè che la distorsione**

cresca con la distanza dal punto in cui la distorsione stessa è minima, più una posizione è vicina al minimo minore sarà la distorsione calcolata in quel punto.

### 9.4.3 Three step search (algoritmo di ricerca approssimato)

TSS esamina il centro e 8 posizioni attorno ad esso. Il punto di distorsione minimo diventa il **centro** per il successivo passo, cioè se considero la posizione 1 e gli 8 vicini tutti etichettati con l'1, supponendo che il minimo lo ottengo nella posizione 3,3, ricercherò al secondo passo gli 8 vicini che etichetto con il 2, poi supponendo di aver trovato il minimo nella posizione 5,3, cercherò a questo passo gli 8 vicini che etichetto con il 3, al termine non potendo ridurre la dimensione della finestra mi fermo alla posizione 5,5 come migliore approssimazione della posizione del macroblocco che stavo cercando. Quindi ad ogni passo dell'algoritmo lo step viene dimezzato e l'algoritmo termina quando lo step diventa uguale a 1.



La Figura fornisce un esempio visivo di applicazione della TSS che trova (+4;+5) come punto di minima distorsione.

# 10 Il colore

Il colore viene utilizzato nelle immagini digitali per due motivi principali:

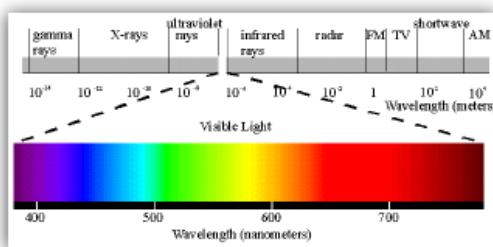
1. è un descrittore che semplifica l'identificazione di un oggetto e la sua estrazione da una scena
2. gli uomini riescono a distinguere migliaia di gradazioni di colore e di intensità, in confronto a solo due dozzine di tonalità di grigio

Nel caso **monocromatico**, l'unico attributo necessario a caratterizzare in ogni punto l'immagine è l'intensità, che si traduce nel valore di grigio del corrispondente pixel. Nel caso della radiazione **cromatica**, per ogni punto dell'immagine occorre tener conto dei tre valori dei *primari* che ne determinano il colore. A tal fine, una nota rappresentazione delle immagini a colori prevede l'impiego di *3 matrici di intensità*.

Le due principali *aree di interesse* sono:

5. **elaborazione full color**: quando le immagini vengono acquisite con un sensore full-color, come uno scanner o una macchina fotografica digitale
6. **elaborazione a falsi colori**: quando si assegna un colore a particolari valori di intensità monocromatici o appartenenti ad una gamma di intensità. Questo tipo di elaborazione viene fatto per semplificare l'analisi delle immagini

Lo **spettro del colore**, in full-color cioè a piena risoluzione cromatica, è formato da sei regioni che non terminano bruscamente, ma ogni colore sfuma gradualmente nel successivo.

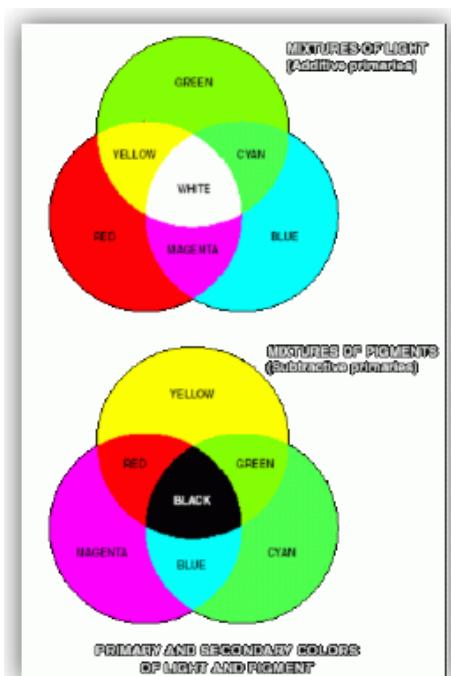


La qualità della luce cromatica viene descritta tramite tre quantità:

- **radianza**: quantità totale di energia che fuoriesce dalla fonte di luce
- **luminanza**: da una misura della quantità di energia percepita dall'osservatore, misurata in lumen
- **luminosità**: descrittore soggettivo impossibile da misurare

Nell'occhio umano tre tipi di foto-pigmenti caratterizzano i ricettori noti come **coni**, con tre curve di sensibilità spettrale che coprono tre differenti bande. In base a questa struttura dell'occhio umano, ogni colore percepito può essere visto come una combinazione di tre colori primari, **il rosso (R)**, **il verde (G)** ed **il blu (B)**. I colori primari possono essere sommati a due a due da produrre i cosiddetti colori secondari: **il magenta (M)**, miscela di rosso e blu, **il ciano (C)**, miscela di verde e blu, e il **giallo (Y)**, miscela di rosso e verde. La miscela dei tre primari, o di un secondario col primario opposto, produce luce bianca.

Mentre la combinazione dei secondari, o di un primario con il suo primario dei coloranti opposto, produce il nero. Quindi distinguiamo i



colori primari della luce (**RGB**), di natura additiva e i colori secondari dei primari RGB (CMY) o primari dei coloranti, di natura sottrattiva.

I colori percepiti sono descritti da 3 attributi:

- **Tinta (hue)**: rispetto alla tinta si distinguono colori *cromatici*, quelli che hanno una tinta, ed i colori *acromatici*, quelli che non la hanno.
- **Saturazione (saturation)**: attributo secondo il quale giudichiamo, nel colore, l'*ammontare* relativo della *tinta componente*. È un attributo che si riferisce alla *purezza del colore*, ovvero alla *quantità di luce bianca* mescolata con una tinta.
- **Brillanza (brightness)**: attributo dell'*illuminazione* in cui si è visto un colore isolato. La brillantezza è la percezione del livello generale di luminanza.

### Teoria del tristimolo

Ogni radiazione elettromagnetica che entra nell'occhio può essere classificata mediante una terna di numeri che rappresentano le attivazioni dei tre tipi di coni. Queste terne di numeri godono di proprietà additiva lineare. Essi sono efficacemente rappresentati come punti di uno spazio 3D detto **spazio tristimolo**. I valori **tristimolo** corrispondono alle quantità di *RGB rosso, verde, blu* necessarie per formare un dato colore; queste quantità sono indicate con *X, Y, Z*. Un colore viene specificato tramite i suoi coefficienti tricromatici definiti:

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

$$z = \frac{Z}{X + Y + Z}$$

quindi  $x + y + z = 1$ . Costituita da solo coefficienti positivi:

- $Y$  = percezione della luminanza sulla retina,
- $x,y$  = chromaticity.

Per ogni lunghezza d'onda di luce nello spettro visibile, i valori tristimolo necessari per produrre il colore corrispondente a quella lunghezza d'onda possono essere ottenuti dalle tabelle o curve che sono state compilate a seguito di sperimentazioni.

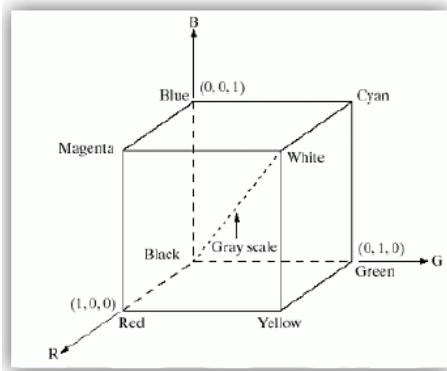
## 10.1 Modelli di Colore

Scopo di un **modello del colore** (o spazio dei colori) è di consentirne la specificazione dei colori con modalità standardizzate. I modelli del colore sono normalmente inventati per motivi pratici e quindi ne esistono molti. A partire dallo spazio della **tricotomia** e al **CIE 1931** i più noti sono quelli degli:

- **Spazi di colore derivati dai dispositivi:**
  - RGB (utilizzati per la rappresentazione a video delle immagini a colori)
  - CMYK (utilizzati per la stampa)
- **Spazi di colore derivati dallo studio della percezione:**
  - HSV (più simile al comportamento vero della percezione del sistema visivo umano)

### 10.1.2 Modello RGB

Nel modello RGB ogni colore è rappresentato dalle sue componenti primarie spettrali di *rosso*, *verde* e *blu*, infatti le immagini rappresentate nel modello RGB sono formate da tre immagini, una per ogni colore primario. Si basa su un sistema di coordinate cartesiane, mentre il suo sottospazio è il seguente cubo:



i *valori primari RGB* si trovano su tre spigoli; i *colori secondari* (ciano, magenta e giallo) sono in altri tre spigoli, Il **nero** è all'origine ed il **bianco** è al vertice più lontano, mentre la diagonale che li congiunge è la **scala dei grigi**.

È lo schema classico utilizzato sia nell'acquisizione che nella visualizzazione, dato che normalmente telecamere, scanner, monitor producono o utilizzano segnali **RGB**.

Se i colori primari sono sommati producono il **bianco**: per questo motivo è detto di tipo additivo, per produrre il bianco tutte le lunghezze d'onda visibili sono trasmesse all'occhio.

### 10.1.3 Il modello CMY

Questo modello è basato sui tre colori secondari *ciano*, *magenta* e *giallo*, che, sono i *primari dei coloranti*. Tenendo in considerazione **il cubo RGB**, e considerano i 3 vertici corrispondente a CMY, rappresenta lo stesso spazio di colore ma utilizza una base alternativa per la rappresentazione pertanto un qualsiasi colore può essere rappresentato tramite una combinazione lineare dei primari dei coloranti. Il modello è pertanto utilizzato nelle operazioni di **output** di immagini *su carta* (stampanti, fotocopiatrici, plotter). Quando la luce bianca colpisce gli inchiostri, alcune lunghezze d'onda visibili vengono assorbite, mentre altre vengono riflesse e quindi viste. Per questo motivo questo modello di formazione del colore si dice **sottrattivo**. Dalla luce, quindi dal bianco, vengono sottratte alcune componenti rimanendone altri e questo permette di generare un colore secondo il modello sottrattivo. Una variante di questo modello usa esplicitamente il **nero**, indicato con **K**, come quarto colore.

Il modello **CMYK** trova largo impiego nei processi di stampa a 4 colori.

Naturalmente il nero può essere ottenuto anche per sottrazione di tutti e 3 i primari dal bianco. I vantaggi principali del modello **CMYK** rispetto al modello **CMY** sono relativi proprio ai problemi posti dalla riproduzione del nero mediante combinazione dei tre primari.

Essendo che il modello **RGB** e **CMY** appartengono allo stesso modello di rappresentazione, cioè il **cubo**, naturalmente vi è un meccanismo abbastanza semplice per la conversione da un modello additivo RGB ad uno sottrattivo CMY. La conversione è rappresentata dalle seguenti equazioni:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \iff \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

Tutte le intensità si intendono normalizzate tra 0 e 1

#### 10.1.4 Il modello HSV

Gli spazi **RGB** e **CMYK** sono stati concepiti in funzione dei dispositivi di visualizzazione (monitor e stampanti), ma non rispecchiano la nostra percezione.

Un modello alternativo molto più vicino al nostro modo di “vedere” i colori è il **modello HSV**, (*Tinta - Saturazione - Valore*), nel quale le componenti di crominanza sono esplicitamente correlate alle proprietà dei colori cui il sistema visivo umano è sensibile. Si basa su uno spazio conoidale in cui i colori sono rappresentati attraverso questo cono. Il **value** prende valore 0 all’apice del cono fino ad arrivare al massimo valore man mano ci si avvicini al piano del cono, mentre la tinta (**hue**) varia sulla circonferenza del cono, mentre la saturazione (**saturation**) cambia partendo dal centro verso l’estremo del cono pertanto varia secondo il raggio della base del cono.

Più in dettaglio possiamo certamente dire che la **hue** è la tinta vera e proprio, cioè la tinta del colore grezzo. Nel contempo noi percepiamo anche una saturazione, quindi diciamo un intensità della tinta che rappresenta la distanza del colore dal grigio più vicino. Il colore poi è più o meno brillante a seconda dell’illuminazione o quantità di luce che viene assorbita dall’oggetto e riflessa dallo stesso.

Lo *svantaggio principale* del modello **HSV** deriva dalle complicate formule di conversione da e verso il modello **RGB**, che ne hanno tra l’altro impedito l’uso nei sistemi **TV**.

#### RGB -> HSV

Calcolare in termini numerici i valori di **H,S,V** vi sono delle conversioni da **RGB** e **HSV** che permettono di passare da un modello additivo ad un modello più percettivo simile a come opera il nostro occhio:

$$V = (R + G + B)/3 \text{ (media tra i valori R G B)}$$

$$S = 1 - \min(R, G, B)$$

$$H = \begin{cases} \theta & \text{se } g \geq b \\ 2\pi - \theta & \text{se } g < b \end{cases} \text{ con } \theta = \cos^{-1} \left( \frac{\frac{1}{2}(r-g) + (r-b)}{\sqrt{(r-g)^2 + (r-b)(g-b)}} \right)$$

(che tiene conto molto dell’angolo di incidenza rispetto alla forma dello spazio conoidale del modello **HSV**).

### 10.5 Segmentazione di immagini a colori

La segmentazione, il processo che divide l’immagine in regioni, può essere applicato in due spazi colore:

- **Spazio a colori HSI:**

si utilizza questo spazio quando si desidera segmentare un’immagine basandosi sul colore e si vuole portare avanti l’elaborazione su singoli piani.

In questo spazio:

- la *tonalità* rappresenta bene il colore, quindi i suoi valori possono essere utilizzati per segmentare l’immagine
- la *saturazione* è usata come immagine maschera per isolare ulteriori regioni di interesse rispetto alla tonalità
- l’*intensità* è utilizzata di meno per la segmentazione in quanto non contiene informazioni riguardanti i colori

- **Spazio colori RGB:**

spazio nel quale si ottengono i risultati migliori. Dato un insieme di punti campione rappresentativo dei colori di interesse è possibile ottenere una stima del colore medio che si vuole segmentare; quindi l’obiettivo della segmentazione è quello di classificare ogni pixel RGB in una data immagine come un colore appartenente alla gamma specificata o al di fuori di essa; per poter fare questo confronto è necessario avere una *misura di similarità*, in questo caso si utilizza la *distanza euclidea*.

## Individuazione di edge a colori

Uno degli strumenti principali che permettono di segmentare un'immagine è l'individuazione dei bordi (edge). Nelle immagini in toni di grigio per trovare i bordi viene utilizzato il gradiente, ma in questo caso non possiamo utilizzarlo perché non è definito per quantità vettoriali, quindi non possiamo calcolare il gradiente sulle componenti RGB singole e poi usare il risultato per formare un'immagine a colori. Al massimo possiamo utilizzarlo lavorando su una singola componente, sempre se ci interessa soltanto individuare i bordi, perché se l'obiettivo è la precisione è necessaria una nuova definizione del gradiente applicabile alle quantità vettoriali.

Siano  $r, g$  e  $b$  i vettori unitari lungo gli assi  $R, G$  e  $B$  dello spazio colore RGB e definiamo i vettori:

$$u = \frac{\partial R}{\partial x} \mathbf{r} + \frac{\partial G}{\partial x} \mathbf{g} + \frac{\partial B}{\partial x} \mathbf{b}$$

$$v = \frac{\partial R}{\partial y} \mathbf{r} + \frac{\partial G}{\partial y} \mathbf{g} + \frac{\partial B}{\partial y} \mathbf{b}$$

$u$  e  $v$  sono vettori unici ottenuti dalla somma dei tre gradienti.

La direzione della percentuale massima di variazione di  $c(x,y)$  (cioè la variazione del gradiente percepita da noi) è data dall'angolo:

$$\theta(x,y) = \frac{1}{2} \tan^{-1} \left[ \frac{2g_{xy}}{g_{xx} - g_{yy}} \right]$$

## Rumore nelle immagini a colori

Il rumore nelle immagini a colori può essere presente in ogni canale con le stesse caratteristiche, ma è possibile anche che i singoli canali vengano interessati in maniera diversa. La causa principale, di avere diversi livelli di rumore, è la **differenza di illuminazione** disponibile per ogni singolo canale. Per eliminare il rumore tramite filtraggio di immagini full-color è possibile lavorare sulle singole immagini componenti o direttamente nello spazio vettoriale a colori.

# 11. Clustering

Quando si parla di immagini a colori le tecniche di elaborazioni di immagini trattate per immagini a livello di grigio possono essere estese alle immagini a colori considerando le 3 componenti RGB, o le 3 bande HSV, in modo separati. Immaginando di prendere per ogni pixel le sue componenti, 3 componenti considerando il modello RGB, in uno spazio 3D i punti nello spazio saranno dei punti che possono essere trattate in modo equivalente a delle misure prese su tanti campioni, e quindi possono essere considerati come dei punti i cui valori possono essere più o meno simili o dissimili e quindi lo spazio colore relativo all'immagine può essere trattato con tecniche di data mining. Una classica tecnica di data mining adatta allo scopo, è il "clustering", essa serve per affrontare problemi nel caso in cui non abbiamo informazioni sul pixel, oltre i valori d'intensità del Rosso Verdi Blu.

Il punto di partenza di tutti gli algoritmi di clustering è un modello che prescinde completamente dalla natura dei dati impiegati e dalle specifiche problematiche disciplinari. Si fa riferimento in generale ad una matrice dei dati contenente informazioni su N oggetti (sono pixel presenti in un'immagine) specificate dai valori assegnati a V variabili (colonne della matrice). Nella matrice avremo tante righe quanti sono i pixel, e tante colonne quanti sono gli attributi da associare ai pixel, nel caso di RGB, avremo 3 colonne.

Scelta delle variabili:

- Indice di somiglianza
- Metodo di formazione dei gruppi
- Determinazione dei criteri di valutazione

Dalla matrice dei dati originaria (di dimensione NxV) si passa ad una matrice di distanze o di similarità fra pattern (di dimensione NxN)

Una volta stabiliti i criteri per la misura del grado di similarità/diversità, è possibile sviluppare molteplici algoritmi per la classificazione dei casi.

- Per variabili di tipo quantitativo si calcolano **misure di distanza**, distanziare quanto più possibile gruppi di pixel che non sono simili tra loro.
- Per variabili di tipo qualitativo si calcolano **misure di similarità**.

## Misure di distanza

Consideriamo dei dati di tipo quantitativo essendo naturalmente valori numerici quelli associati allo spazio colore determinato per cui le misure di distanza possono essere le classiche **misure euclidean** di **norma 2**, che non è altro che la radice quadrata della somma delle distanze al quadrato elemento per elemento delle osservazioni, oppure la **distanza di Manhattan** o anche detta **a blocchi** che non è altro che la sommatoria su tutte quante le componenti dei vettori osservazioni delle distanze valore per valore corrispondente in valore assoluto.

$$d(i, k) = \left[ \sum_{j=1}^d (x_{ij} - x_{kj})^2 \right]^{1/2}$$

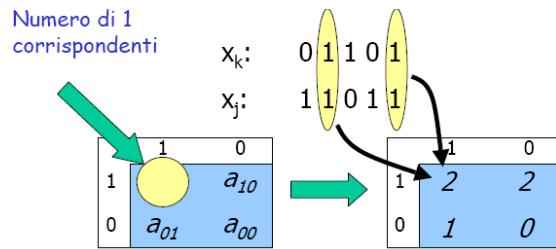
**Distanza euclidea (di norma 2)**

$$d(i, k) = \sum_{j=1}^d |x_{ij} - x_{kj}|$$

**Distanza di Manhattan (o a blocchi)**

La distanza euclidea non è altro che la lunghezza dell'ipotenusa del triangolo rettangolo, mentre la distanza di Manhattan è il modo con il quale si raggiunge y a partire da x attraversando prima le colonne e poi le righe.

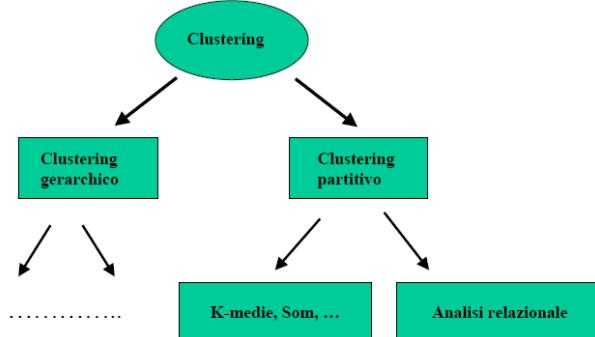
## Misure di similarità



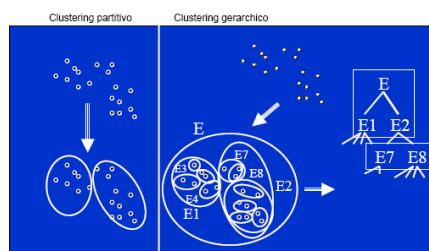
- ✓ Jaccard:  $d(i,k) = (a_{11}) / (a_{11} + a_{10} + a_{01})$
- ✓ Condorcet:  $d(i,k) = a_{11} / [a_{11} + 0.5(a_{10} + a_{01})]$
- ✓ Dice bis:  $d(i,k) = a_{11} / [a_{11} + 0.25(a_{10} + a_{01})]$

Questo è un esempio mostra una **misura di similarità** con le osservazioni rappresentate **in binario**, ed in questo caso si posso applicare misure di similarità di **Jaccard**, **Condorcet** e **Dice bis**. Sono misure molto usate per le tecniche di **datamining** soprattutto quando vogliamo applicare **information retrieval**, cioè quando vogliamo effettuare recupero di informazioni da basi si dati che naturalmente rappresentano informazioni per esempio in binario.

## Tecniche di Clustering



Qualsiasi sia la misura di similarità per determinare gruppi di osservazioni, più in generale chiamati **cluster di osservazioni**, le tecniche di clustering si suddividono in **clustering gerarchico** e **clustering partitivo**. Per **gerarchico** intendiamo dire che partiamo da i singoli pixel (singole osservazioni) a determinare dei gruppi sempre più numerosi di osservazioni, quindi partiamo da cluster che contengono un pixel ciascuno e poi fino a costituire cluster più numerosi, finché naturalmente un certo criterio di similarità non venga più soddisfatto. Viceversa con **partitivo** parte a determinare delle partizioni dello spazio tale che ogni sottopartizione soddisfi certi criteri e in questo calcolo (insieme di algoritmi di clustering partitivo) certamente possiamo considerare il **K-means**, gli algoritmi che derivano dalle reti neurali quali le **Som** (Self Organizing Maps),



Quindi ricapitolando il clustering partitivo cerca di suddividere un insieme di osservazioni rappresentati da quei cerchi in alto in sottogruppi all'interno dei quali naturalmente le osservazioni **soddisfano un criterio di similarità**. Viceversa il clustering gerarchico fa sì che ogni punto venga associato ad un cluster qui rappresentato in bianco i quali poi vengono agglomerati (o fusi) per determinare dei cluster di dimensione sempre maggiore in modo tale da terminare quello che è un albero del clustering, quindi un analisi di tipo gerarchica del clustering applicato ai dati.

### In generale

Possiamo formalizzare la tecnica del clustering considerando che i K cluster,  $C_1, \dots, C_k$  in cui vogliamo suddividere l'insieme delle osservazioni, abbia delle medie,  $m_1, \dots, m_k$ , che rappresentano appunto **la media del cluster**.

Quello che si vuole fare tipicamente tipicamente è minimizzare l'errore ai minimi quadrati, definito come:

$$D = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - m_k\|^2$$

- $\sum_{k=1}^K$  sommatoria di tutti i cluster
- $\sum_{x_i \in C_k}$  sommatoria di tutti i pattern (o osservazione all'interno di ogni cluster)
- $\|x_i - m_k\|^2$  differenza che intercorre tra l'osservazione e la sua media, misurata con una distanza euclidea

Tra tutte le possibili partizioni in **K cluster**, in cui è possibile suddividere lo spazio, quello che si vuole fare è scegliere quello che corrisponde alla partizione che minimizza l'errore **D** ai minimi quadrati. Vi sono vari algoritmi di clustering, di sicuro il più noto è il **K-means** e poi l'**Isodata**, altri invece più adatti, o meglio mirati all'operare sulle immagini a colori sono quelli basati sull'**istogramma** (Histogram-Based e Recursive Variant) e quelli basati su **grafi** (Graph-Theoretic).

## 11.1 Algoritmo K-means

L'algoritmo K-means opera come segue:

Determinate inizialmente delle medie casuali in numero di K, dove K è il numero di cluster in cui vogliamo dividere il nostro spazio, viene misurato la distanza che intercorre da ogni osservazione/pattern/oggetto, dalle k medie e viene scelta quella media più vicina al pattern in considerazione. Una volta effettuata questa operazione per ogni pattern, viene determinata una prima partizione, considerando come partizione l'insieme di tutte le osservazioni che sono più simili ad una media, e quindi sono più vicini ad una media, poiché abbiamo k medie e k cluster. A questo punto viene ricalcolata la media, effettuando la media dei punti/osservazioni che fanno parte del cluster, il valore del vettore medio si sposta al centro di ogni cluster, vengono ricalcolate le distanze e riposizionate le medie, questo procedimento iterativo procede fintanto che ad un passo t le medie non cambiano da quelle calcolate al passo t-1, quindi basta calcolare le distanze tra le medie, se esse sono al di sotto di una certa soglia gamma, che rappresenta la precisione, allora il procedimento termina.

Riepilogando:

1. Assegna ogni oggetto/osservazione ad uno dei cluster  $k = 1, \dots, K$ .
2. Calcola le medie di ogni cluster:

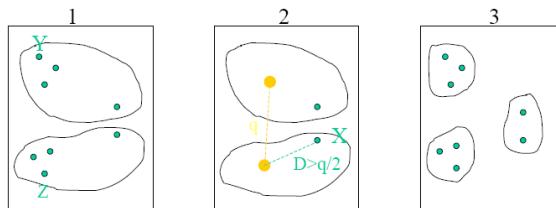
$$\mu_k = \frac{1}{N_k} \sum_{z_i \in C_k} z_i$$

3. Riassegna ogni oggetto/osservazione  $z_i$  al cluster con la media più vicina  $\mu_i$
4. Ritorna al passo 2 finché le medie dei cluster non si modificano sostanzialmente.

## 11.2 Variante di Meng-Hee- Heng

Una variante del K-means, il procedimento è il seguente:

1. Vengono selezionati 2 punti  $Y$  e  $Z$  che sono più lontani nello spazio delle misure e vengono considerate come media iniziale dei cluster
2. Vengono poi assegnati tutti i punti al cluster la cui media è più vicina e vengono ricalcolate le medie
3. Sia poi  $d$  la distanza massima da ogni punto alla propria media del cluster e sia  $X$  il punto con tale distanza.
4. Sia inoltre  $q$  la distanza media tra ogni coppia di medie.
5. Se  $d > q / 2$ ,  $X$  risulta essere la nuova media di cluster.
6. Se si forma un nuovo cluster, ripetere a partire dal passo 2.



## 11.2 Isodata

Un metodo alternativo di clustering è noto come **isodata**, l'idea è quella di selezionare molte medie di cluster all'inizio e di formare i cluster.

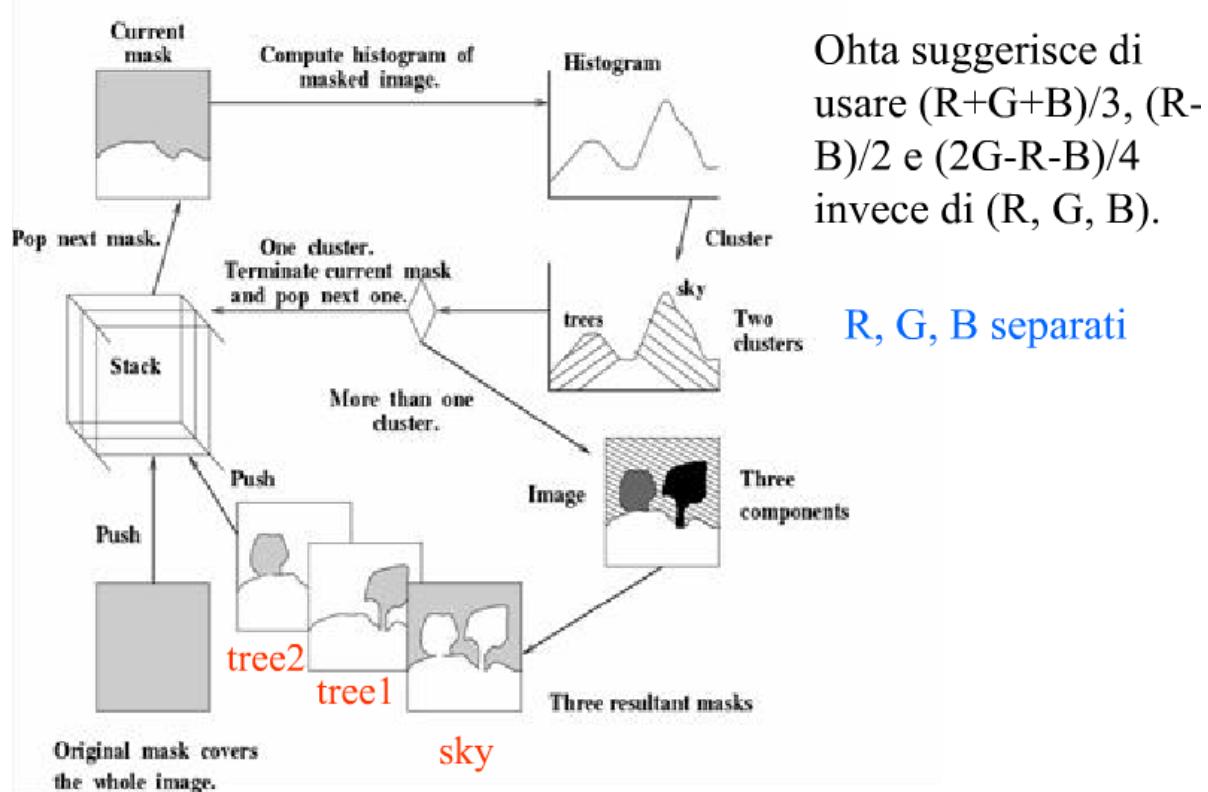
Passi dell'algoritmo:

1. **Selezionare molte medie di cluster e formare i cluster**, il cluster viene formato calcolando la distanza che intercorre fra ogni punto e tutte le medie scelte inizialmente e attribuire successivamente un punto alla media per la quale si ha una distanza minima. Il cluster ricordiamo che rappresenta un gruppo di pixel che hanno intensità di colore fra di essi abbastanza omogenea, pertanto rappresenterebbe una regione da discriminare dall'immagine di partenza.
2. **Suddividere qualsiasi cluster la cui varianza è troppo elevata**, (somma dei quadrati delle differenze tra l'intensità di ogni pixel sulle 3 componenti e la media del cluster stesso) è troppo elevata, cioè supera una certa soglia.
3. **Raggruppare i cluster** (numero di pixel appartenente al cluster stesso), se sono troppo piccoli di una certa soglia.
4. **Ricalcolare le medie**
5. **Ripetere fino a che i passi 2 e 3 non possono essere applicati**

## 11.3 Il metodo di Ohlander

Il **metodo di ohlander** si applica ad un'immagine a colori di 3 componenti rosso, verde e blu (RGB), considerando l'istogramma di ogni banda separatamente. A partire pertanto da un istogramma questo viene clusterizzato in più cluster e vengono determinati tre componenti chiamati **albero 1, albero 2 e cielo**. Le 3 immagini che vengono individuate a partire dai cluster in cui è stato suddiviso l'istogramma e sono rappresentate da maschere. L'immagine originale avrà una maschera per tutti quei pixel che appartengono ad un cluster settata ad uno, e per i pixel non appartenenti settata a 0. Le 3 maschere vengono poste al top di uno stack dopodiché l'algoritmo ricorsivamente preleva una maschera dallo stack e analizza l'istogramma dell'immagine di cui si è fatta la maschera analizzando l'istogramma dell'area mascherata.

L'istogramma viene di nuovo clusterizzato se vengono determinati 2 o più cluster. Il procedimento si ripete fin quando non viene determinato un solo cluster. L'idea è quindi quella di analizzare ricorsivamente gli istogrammi delle parti in cui può essere suddivisa un'immagine e quindi considerare segmenti più raffinati in cui può essere suddiviso un segmento principale tratto dall'immagine. Naturalmente la clusterizzazione, il clustering dell'istogramma, è fatto considerando la multi modalità dell'istogramma, pertanto il cluster viene individuato individuando una soglia a partire dalla quale si considera una suddivisione dell'istogramma e la soglia tipicamente è associata ad una valle, come rappresentato qui in alto nell'istogramma bimodale con una valle. Pertanto la valle rappresenterà una soglia di discriminazione tra un cluster rappresentante l'albero (alla sinistra) e un cluster (alla destra) rappresentante il cielo.

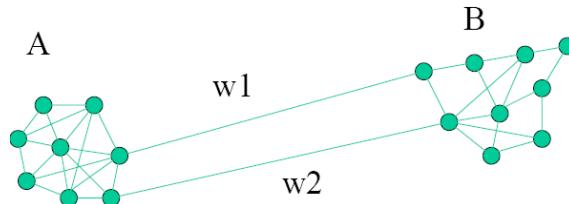


## 11.4 Tagli minimali (jinbo shi)

Tagliare le distanze fra punti appartenenti a cluster differenti equivale ad effettuare un taglio minima (minimal cut). Sia un grafo **G** rappresentato da un insieme di vertici **V** e da un insieme di archi **E**, tali che ogni arco **(u,v)** possiede un peso **w(u,v)** che rappresenta in questo caso la similarità tra il pixel rappresentato dal nodo **u** e il pixel rappresentato dal nodo **v**, il grafo può essere partizionato in 2 insieme disgiunti con insiemi di nodi **A** e **B** rimuovendo gli archi che connettono questi insiemi. Sia pertanto il taglio fra 2 sottografi indicati con **A** e **B** come la sommatoria su tutti i vertici appartenenti ad **A** e alla sommatoria su tutti i vertici appartenenti a **B** dei pesi intercorrenti tra il nodo **u** appartenenti al gruppo **A** e il nodo appartenente al gruppo **B**, pertanto i pesi degli archi **u** e **v**, un modo possibile è quello di dividere sementare **G** in modo tale da trovare il **taglio minima**, il taglio che corrisponde ad avere la somma dei pesi minima all'interno di ogni sottografo **A** oppure sottografo **B**.

$$\text{cut}(A,B) = \sum_{u \in A, v \in B} w(u,v)$$

Nel caso appena illustrato il tagliare i pesi **w<sub>1</sub>** con **w<sub>2</sub>** risulta essere il taglio minima che si può effettuare lasciando i 2 sottografi **A** e **B** essere quelli rappresentanti il taglio minima del grafo complessivo.



### 11.4.1 Taglio normalizzato

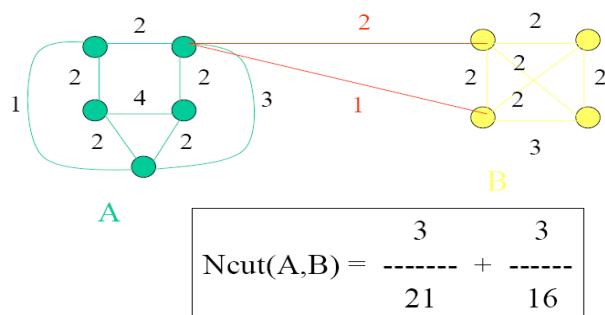
Poiché il taglio minima favorisce il taglio di piccoli gruppi di nodi, cioè è possibile che dei nodi o dei gruppi di nodi risultano essere distanti da gruppi più densi di nodi tale che questi nodi vengono visti come se fossero degli outlier rispetto a dei gruppi di nodi più densi, più corposi, questo taglio comporterebbe la determinazione di aree di regioni piccole e insignificanti per evitare ciò si utilizza un **taglio normalizzato**:

$$N\text{cut}(A,B) = \frac{\text{cut}(A,B)}{\text{asso}(A,V)} + \frac{\text{cut}(B,A)}{\text{asso}(B,V)}$$

$$\text{asso}(A,V) = \sum_{u \in A, t \in V} w(u,t)$$

qui indicato con **Ncut** tra **A** e **B**, non è altro che la somma tra il taglio di **A** e **B** diviso l'asso di **A** e **V** più il taglio fra **A** e **B** diviso l'asso di **B** e **V** dove l'asso di **(A,V)** non è altro che la sommatoria di tutti i pesi dell'arco **(u, t)** dove **u** appartiene al sottogrammo **A** e **t** è un vertice qualsiasi.

Quell'assi di **(A,V)** misura quindi il grado di connessione di **A** al grafo.



# Possibili Domande d'Esame

- Algoritmo di Olander. Bimodale o Multimodale?
- Grafo, il peso è associato a? Come si calcola il peso?
- Cos'è l'asso? Serve? Perche è normalizzato?
- Esempio: un segnale è corrotto da un rumore superiore a 100Hz, lo si vuole campionare come si fa?
- Filtro Passo a passo come si applica?
- Come faccio a applicare un filtro field a un segnale?
- Come si campiona un segnale?**(Teorema del Campionamento)**
- Come si ricava il background : **(Sottrazione dello sfondo)**
- Confronto tra più frame con un algoritmo, quale?
- Filtro Hi-boost
- relazione tra filtro hi-boost e filtro hi-pass qual'è?
- come si applicano i coefficienti di un filtro Low-pass? La mascherina come si calcola?
- Da dove esce 6 sigma? xke si usa un valore maggiore?
- Filtro gaussiano
- Come si trovano i valori discreti
- come si calcola il laplasiano
- Shimanic
- minimizzano cosa?
- cat?
- sottrazione dello sfondo
- le medie si aggiornano? e come?
- come facciamo a sapere se è una gaussiana forground o background?
- come si calcola la freq massima?
- Equalizzazione e specifica dell'istogramma
- Filtri domino dello spazio
- Fourier
- Filtri dominio frequenze
- Campionamento e quantizzazione
- Segmentazione immagini a colori
- region growing
- Stima del movimento
- Trasformata distanza
- Trasformata di Hough per rette e per cerchi
- hsv rgb il colore in generale