

# **Elaborazione delle immagini – Staiano**

Appunti a cura di Fiorentino Michele  
- Università degli studi di Napoli “Parthenope”

## **3. LEZ 1: Introduzione**

- 5. Discipline, categorie di elaborazione e tipi di immagini
- 6. Fasi chiave nell’elaborazione delle immagini

## **7. LEZ 2: Introduzione all’IA e al ML**

- 7. Intelligenza Artificiale
- 9. Machine Learning

## **10. LEZ 3: Apprendimento e Apprendimento supervisionato**

- 10. Apprendimento
- 12. Apprendimento supervisionato
- 14. Classificazione
- 15: Matrice di conclusione

## **16. LEZ 4: Apprendimento non supervisionato**

- 16. Clustering:
  - 17. Clustering k-means;
  - 18. Clustering gerarchico;

## **20. LEZ 5: Formazione delle immagini**

- 22. Geometria della formazione dell’immagine
- 24. Fotometria della formazione dell’immagine

## **25. LEZ 6: Fondamenti(1)**

- 25. Radiazione elettromagnetica
- 26. Acquisizione delle immagini
- 28. Modello di creazione delle immagini
- 29. Campionamento e quantizzazione
- 30. Rappresentazione delle immagini digitali
- 31. Risoluzione spaziale e dell’intensità
- 31. Interpolazione delle immagini
- 33. Relazioni tra pixel:
  - 33. Vicinanza;
  - 34. Adiacenza;
  - 35. Componenti connesse;
  - 36. Distanza.

## **37. LEZ 7: Fondamenti(2)**

- 37. Operazioni puntuali e matriciali
- 37. Operazioni lineari e non lineari
- 38. Operazioni aritmetiche (operazioni tra pixel)
- 38. Operazioni insiemistiche e logiche
- 39. Operazioni spaziali

- 42. LEZ 8: Trasformazioni di intensità**
- 43. Funzioni per la trasformazione di intensità
  - 46. Funzioni di trasformazione lineare a tratti
  - 48. Elaborazione di istogrammi:
    - 52. Specifica dell'istogramma;
    - 53. Elaborazione locale dell'istogramma;
    - 54. Statistiche da istogramma.
- 55. LEZ 9: Trasformata di Fourier (1D)**
- 55. Introduzione
  - 56. Serie di Fourier
  - 58. Trasformata di Fourier
  - 61. Convoluzione
  - 62. Campionamento
  - 63. Trasformata di Fourier di una funzione campionata
  - 64. Teorema di campionamento
  - 65. Aliasing
  - 66. Ricostruzione di una funzione campionata
  - 67. Trasformata di Fourier discreta
- 69. LEZ 10: Trasformata di Fourier per immagini (2D)**
- 69. Impulso e proprietà di Sifting
  - 70. Coppia di trasformate di Fourier continue 2D
  - 71. Teorema di campionamento 2D
  - 72. Aliasing nelle immagini digitali
  - 74. Coppia DFT 2D
  - 76. Spettro di Fourier e Angolo di fase
  - 78. Teorema di convoluzione 2D
- 80. LEZ 11: Filtraggio nel dominio delle frequenze (1)**
- 81. Tipi di filtri
  - 85. Progettazione di filtri spaziali
  - 86. Filtri passa basso
  - 88. Filtri passa alto
- 89. LEZ 12: Filtraggio nel dominio delle frequenze (2)**
- 89. Laplaciano
  - 90. Filtraggio omorfico
  - 92. Filtraggio selettivo
  - 93. Computazione della DFT
- 94. LEZ 13: Morfologia applicata alle immagini**
- 95. Morfologia per immagini binarie
  - 98. Trasformazioni hit-or-miss
  - 99. Morfologia per immagini in scala di grigio
  - 100. Trasformazioni top-hat e bottom-hat

## LEZ 1 – Introduzione

L'**elaborazione delle immagini** è una disciplina che si focalizza su due attività principali:

- il miglioramento dell'informazione grafica per l'interpretazione umana;
- l'elaborazione dei dati dell'immagine per scopi di memorizzazione, trasmissione e rappresentazione per la percezione autonoma delle macchine.

Essa considera, in particolare, l'informazione contenuta nelle immagini digitali.

Le immagini possono essere generate da segnali.

Nel mondo fisico, un **segnale** è potenzialmente una qualsiasi quantità che esibisca una variazione nel tempo e/o nello spazio. Infatti è possibile generare immagini da altre sorgenti che non siano la luce, come le onde acustiche/ultrasoniche, elettroniche e sintetiche.

Fondamentalmente, il segnale viene catturato da un dispositivo di acquisizione (microfono, fotocamera), il quale lo converte in un segnale elettrico. Il segnale elettrico deve essere poi discretizzato attraverso un convertitore A/D così che possa poi essere elaborato da un calcolatore.

Il processo inverso riguarda convertire il segnale digitale in analogico attraverso un convertitore D/A.

Un'**immagine** può essere definita come una funzione bidimensionale  $f(x,y)$ , dove  $x$  e  $y$  sono le coordinate spaziali (sul piano), e l'ampiezza di  $f$  in ogni coppia di coordinate  $(x,y)$  viene chiamata *intensità*, o livelli di grigio in quel punto.

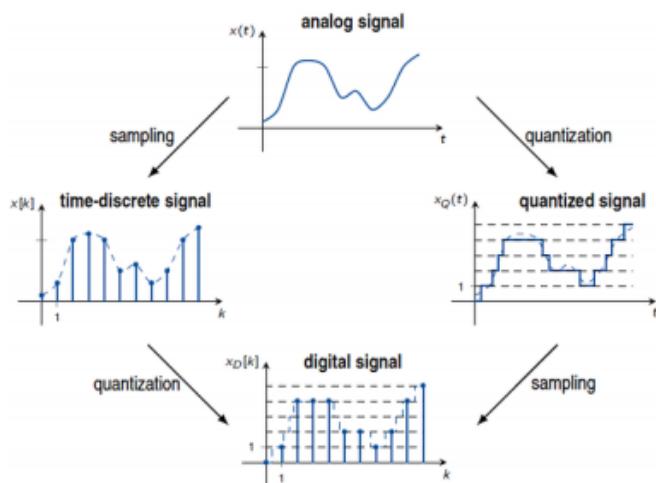
L'immagine è dunque una matrice, costituita da un numero finito di elementi, detti *pixel*, ciascuno dei quali ha una specifica posizione e un valore corrispondente.

Il livello di dettaglio dipende sia dalla dimensione della matrice che dal numero di livelli di intensità rappresentabili, che normalmente sono 256 valori, dunque 8 bit.

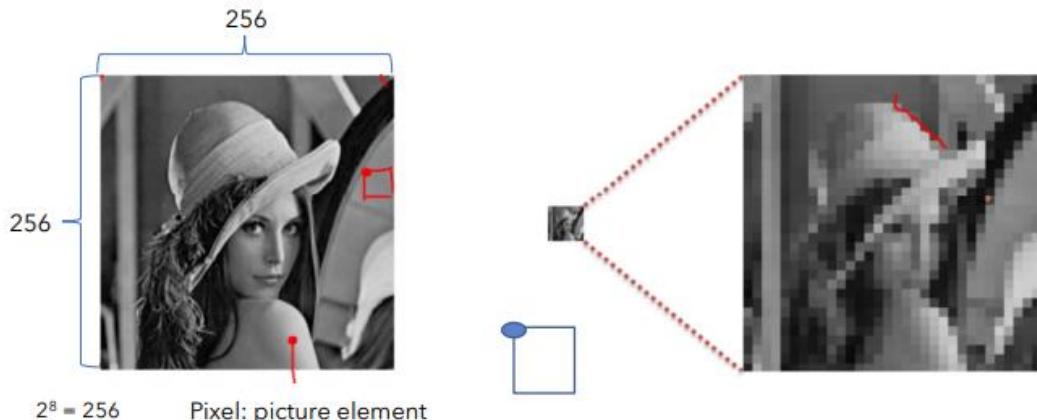
Un'immagine non è altro che un segnale analogico. Un'immagine digitale invece è un segnale discreto. La conversione da segnale analogico a segnale discreto è detta *discretizzazione*.

La **discretizzazione** di un segnale analogico si realizza per mezzo di un'operazione di *campionamento* ed una di *quantizzazione*:

- il **campionamento** discretizza l'asse del **tempo**, cioè suddivide l'asse a intervalli regolari e considera il segnale solo in quei punti;
- la **quantizzazione** discretizza l'asse dei **valori**, cioè approssima il valore dei campioni a un certo numero prefissato di valori.



## CAMPIONAMENTO (SAMPLING)



## QUANTIZZAZIONE



*Esempio di campionamento e quantizzazione.*

La prima immagine 256x256 viene campionata così che per ogni ogni 8 pixel ne corrisponda 1. Il ridimensionamento consiste nel prendere un pixel e ripeterlo per 8 volte. L'immagine 1 ha 256 livelli di grigio, la 2 ne ha 16, la 3 ha 4. Un altro obiettivo della quantizzazione è cercare di trovare il giusto compromesso affinché non si abbiano immagini troppo pesanti.

## Discipline riguardo l'elaborazione delle immagini

Ci sono almeno **tre discipline** che si sovrappongono: elaborazione delle immagini (*image processing*), comprensione delle immagini (*image analysis*), e computer vision.

Il confine fra queste tre discipline non è ben definito. Possiamo dire che l'elaborazione delle immagini è una disciplina dove sia l'input che l'output sono immagini, la computer vision riguarda emulare la visione umana (settore dell'IA), mentre la comprensione delle immagini è a cavallo fra queste due.

Il continuum che va dall'elaborazione delle immagini alla computer vision è, dunque, non ben definito. Possiamo comunque distinguere **tre categorie di elaborazione**, caratterizzate dal loro livello di **astrazione**:

- *basso livello*: riguarda il miglioramento delle immagini; in input viene passata un'immagine e si ottiene in output un'altra immagine.
- *medio livello*: riguarda l'estrazione di caratteri o feature; in input viene passata un'immagine e si ottiene in output degli attributi.
- *alto livello*: riguarda il riconoscimento di oggetti. in input vengono passati degli attributi e si ottiene in output una "comprensione".

Ad esempio, nell'OCR (optical character recognition) è un'operazione di basso livello l'eliminazione del rumore e la binarizzazione, di medio livello la segmentazione dei caratteri, e di alto livello il riconoscimento dei caratteri e della struttura del testo.

## Tipi di immagini

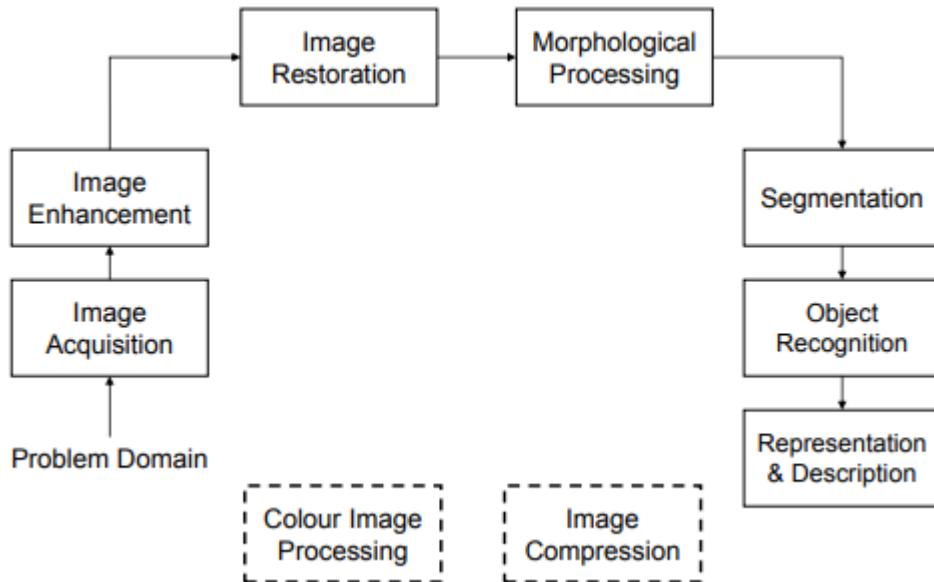
Le immagini possono essere classificate in **3 categorie**:

- **immagini di riflessione**: contengono informazioni sulla superficie dell'oggetto;
- **immagini di emissione**: contengono informazioni interne all'oggetto;
- **immagini di assorbimento**: contengono informazione sulla struttura interna all'oggetto.

Possiamo pensare alle immagini ai raggi gamma, ai raggi X o all'ultravioletto, fra le tante. Diversi tipi di immagini possono dare informazioni importanti per diversi campi applicativi, fra cui la medicina, l'astronomia o l'industria.

## Fasi chiave nell'elaborazione delle immagini

Possiamo distinguere diversi fasi (o processi) tipiche dell'elaborazione delle immagini digitali, dove ogni fase ha uno scopo e un obiettivo diverso.



1. **L'acquisizione delle immagini** è il primo processo, che riguarda generare immagini partendo da una sorgente, come può essere la luce.
2. **Il miglioramento delle immagini** è un processo di manipolazione di un'immagine che, in base alla specifica applicazione, ne migliora la qualità. Ovviamente l'applicazione è importante, ad esempio un metodo per migliorare un'immagine a raggi X potrebbe non essere ideale per le immagini satellitari.
3. **Il ripristino dell'immagine** è un processo che si basa su tecniche di ripristino basate su modelli matematici. Esso è "oggettivo" (al contrario del miglioramento che è "soggettivo").
4. **L'elaborazione morfologica** è un processo che si occupa degli strumenti per estrarre le componenti di un'immagine.
5. **La segmentazione** è un processo che divide l'immagine nelle sue parti costituenti.

Distinguiamo inoltre altre fasi, come: il *riconoscimento di oggetti*, la *rappresentazione e descrizione*, l'*elaborazione di un'immagine a colori*, e la *compressione*.

## LEZ 2 – Introduzione all'IA e al ML

L'IA e il Machine Learning sono **fondamentali** per il riconoscimento di oggetti e volti, per la segmentazione di scene e la classificazione di immagini.

La normale segmentazione distingue i contorni degli oggetti ma non è in grado di riconoscerli. La *segmentazione semantica* tiene conto anche del significato, e comprende:

- scene parsing: tutti le istanze della stessa classe hanno lo stesso colore;
- instance segmentation: istanze diverse della stessa classe hanno colori diversi;
- semantic boundary detection: effettua la segmentazione, ma tiene conto solo dei bordi e non della parte interna.

### Intelligenza Artificiale

L'IA è un'intelligenza contrapposta a quella naturale. Aristotele fu il primo a cercare di formulare un insieme preciso di leggi che governano la parte razionale della mente, da cui i *sillogismi*.

Un sillogismo è un tipo di ragionamento che permette di generare conclusioni meccanicamente, a partire da un insieme dato di premesse di partenza (es. tutti gli uomini sono mortali. Socrate è un uomo, dunque è mortale).

Possiamo dare diverse **definizioni di Intelligenza Artificiale**, che possiamo organizzare in quattro categorie in base a criteri diversi:

- sistemi che pensano come gli umani;
- sistemi che agiscono come gli umani;
- sistemi che pensano razionalmente;
- sistemi che agiscono razionalmente.

Il test di Turing permette di verificare se un sistema si comporta in modo intelligente. Tuttavia tale test rimane controverso come test di intelligenza in quanto, sebbene si concentri su conoscenza, comprensione linguistica, ragionamento e apprendimento, non considera molti aspetti percepiti come intelligenza come la percezione, le abilità motorie e la strategia. Inoltre, si tratta di un test difficile da superare.

L'IA ha fondamenta in molte discipline, come la filosofia, la matematica, la psicologia, la neuroscienza, e così via. Il termine fu coniato nel '56, nonostante abbia origini molto più antiche.

Nel 1943 fu costruito un modello di neuroni artificiali booleani per eseguire calcoli da McCulloch & Pitts. Successivamente, possiamo trovare altri nomi di spicco come Arthur Samuel, John McCarthy o Caianiello.

L'IA, dagli anni '50, ha avuto uno sviluppo graduale e fu limitata principalmente dall'hardware dell'epoca, motivo per cui si verificò anche l' "A.I. Winter" fra gli anni 70 e 90, con l'eccezione dei "sistemi esperti" negli anni '80.

I **sistemi esperti** sono programmi che cercano di riprodurre le prestazioni di una o più persone esperte in un determinato campo di attività. Si pensi a MYCIN, usato per la diagnostica per le infezioni del sangue (e introdusse il concetto di incertezza nel ragionamento).

Un sistema esperto è composto da **3 componenti principali**:

- **base di conoscenza**: contiene informazioni essenziali sul dominio del problema, ed è rappresentata con fatti e regole codificate in regole IF... THEN.
- **motore inferenziale**: meccanismo per derivare nuova conoscenza dalla base di conoscenza e dalle informazioni fornire dall'utente.
- **interfaccia utente**: permette l'interazione con gli utenti, oltre allo sviluppo e la gestione della base di conoscenza.

Nel 1987 l'IA diventa una scienza, e distinguiamo due tipologie di ricercatori: neats (perfettini) e scruffues (scapigliati).

Il termine **razionale** è usato in modo molto specifico e tecnico, cioè indica raggiungere nel modo migliore gli obiettivi prefissati, i quali sono espressi in termini di utilità dei risultati. Di conseguenza, essere razionale significa massimizzare la propria utilità attesa.

Un **agente** è un'entità che percepisce e agisce. Un **agente razionale** seleziona le azioni che massimizzano la sua utilità attesa, in particolar modo percependo l'ambiente attraverso l'ambiente e agendo in questo attraverso degli attuatori.

Col tempo si è arrivati all'IA Revolution. Basta pensare a tutti quei sistemi intelligenti che troviamo e con cui interagiamo oggigiorno (come Alexa). Ciò è stato possibile anche per la continua riduzione del costo della computazione, e dunque del prezzo dei dispositivi, e come questi dispositivi siano interconnessi.

## Machine Learning

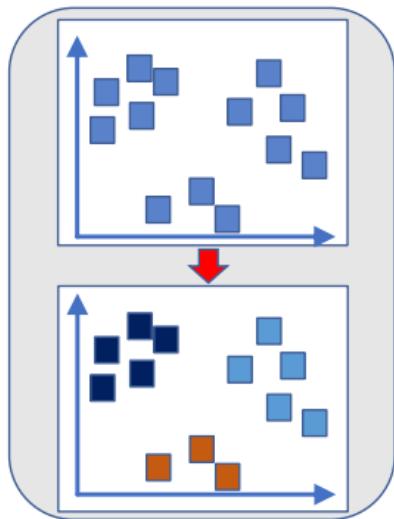
Il Machine Learning è un sottoinsieme dell'area dell'IA, ed ha lo scopo di dotare i computer della capacità di apprendere dall'esperienza. Più esempi sono disponibili per l'apprendimento, migliori saranno le prestazioni.

Il **reinforcement learning** è un paradigma di apprendimento finalizzato alla costruzione di agenti software che intrapendono azioni in un ambiente al fine di massimizzare un profitto espresso in forma di premio (se l'azione è giusta viene premiato, altrimenti viene punito).

Possiamo distinguere **due tipologie di apprendimento:**

- **apprendimento supervisionato:** sviluppa un modello predittivo sulla base sia dei dati in input che di output.  
Esso usa sia tecniche di *classificazione* che di *regressione*:
  - o **tecniche di classificazione:** prediscono risposte concrete (se un animale è un cane o un gatto, se un email è spam o meno);
  - o **tecniche di regressione:** prediscono risposte continue (cambiamenti di temperatura).
- **apprendimento non supervisionato:** trova pattern nascosti o strutture intrinsiche nei dati. Usa tecniche di *clustering*, cioè forma dei gruppi individuando le similitudine fra gli elementi.

es. *di clustering*



## LEZ 3 – Apprendimento e Apprendimento Supervisionato

### Apprendimento

Il Machine Learning è suddivisibile in due categorie: *apprendimento supervisionato* e *apprendimento non supervisionato*.

L'apprendimento supervisionato comprende tecniche di *regressione* e di *classificazione*, mentre l'apprendimento non supervisionato comprende tecniche di *clustering*.

Un programma **apprende** dall'esperienza **E**, rispetto ad una data classe di compiti **T** e ad una misura delle prestazioni **P**, se le sue prestazioni sui compiti in T, come misurato in P, migliorano con l'esperienza E.

Un **sistema di apprendimento** può essere trattato come una scatola nera (black-box). Tale sistema è costituito da 5 passi:

1. **raccogliere gli esempi di addestramento o di training** (esperienza **E**).  
Infatti senza alcun esempio il sistema non sarà in grado di apprendere.
2. **rappresentare l'esperienza**.  
Cioè si deve scegliere uno schema di rappresentazione per l'esperienza/esempi.
3. **scegliere una rappresentazione per la black-box**.  
È necessario scegliere una funzione F per approssimare la black-box. Per un dato input X, il valore di F fornirà la classificazione di X.  
F sarà funzione di alcuni parametri adattivi, o pesi,  $W = (w_1, w_2 \dots w_n)$  che l'algoritmo di apprendimento può modificare (imparare).
4. **apprendere/modificare i pesi**.  
È necessario un algoritmo di apprendimento per modificare i pesi in modo tale che possano essere appresi dal sistema dall'esperienza dei dati di training.
$$E = (X, D)$$
$$F(W, X) = D$$
dove D rappresenta la predizione corretta.  
L'algoritmo deve avere l'obiettivo di minimizzare quanto più possibile l'errore.
5. **Testare il sistema**.  
Viene fornito un nuovo input X, calcolato dal sistema in base a  $F(W, X)$ .

Lo **spazio delle feature** è una rappresentazione di oggetti del mondo reale con i vettori delle feature.

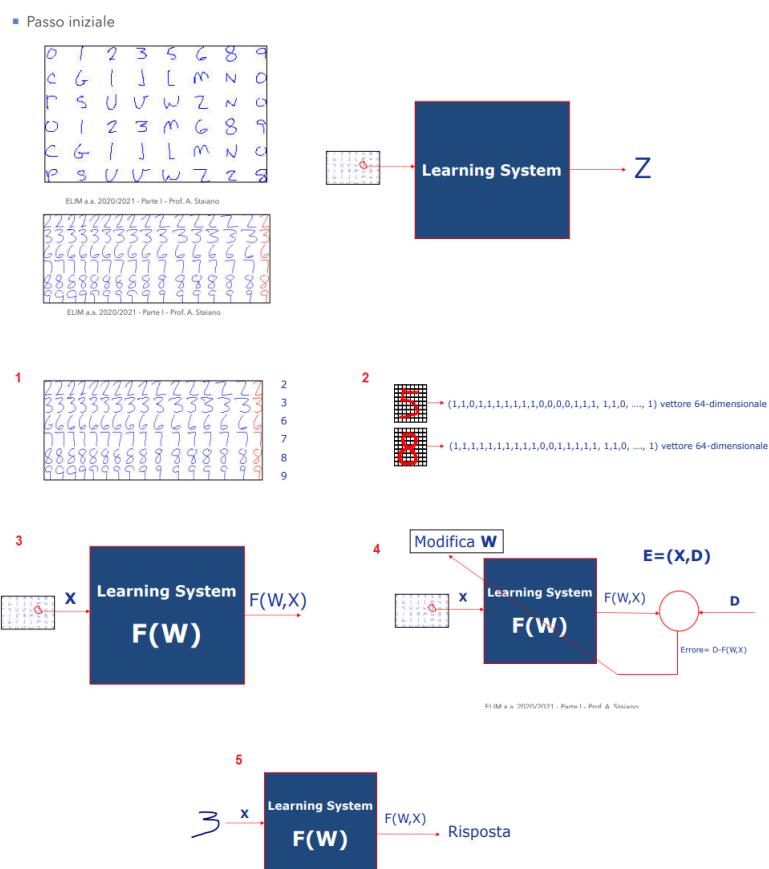
## Esempio, progettare un sistema di apprendimento

Consideriamo il riconoscimento dei caratteri:

- **Compito T:** riconoscere i caratteri scritti a mano;
- **Misura delle prestazioni P:** percentuale di caratteri classificati correttamente;
- **Esperienza E di addestramento:** una base di dati di caratteri scritti a mano con le relative classificazioni.

Ogni carattere passerà attraverso una black-box che restituirà il carattere riconosciuto.

1. raccogliamo gli esempi di addestramento.
2. rappresentiamo l'esperienza. In questo caso utilizziamo un'immagine 8x8 (diciamo in b/n), dove 1 rappresenta un pixel nero, 0 rappresenta un pixel bianco.  
L'immagine 8x8 è l'input X ed è rappresentabile come un vettore n-dimensionale, dove n=64.
3. scegliamo una rappresentazione per la black-box, cioè un modello di ML, ed otterrò dunque una funzione che dipende sia dall'input che dal valore dei parametri.
4. il sistema apprende/modifica i pesi. L'algoritmo scelto ha l'obiettivo di minimizzare l'errore. In fase di apprendimento devo dunque calcolare la funzione di errore, e ad ogni valutazione questo errore deve essere sempre più piccolo.
5. Testo il sistema.



## Apprendimento supervisionato

L'apprendimento supervisionato è un tipo di apprendimento automatico, così chiamato in quanto nell'insieme dei dati di addestramento vengono fornite anche le soluzioni. L'obiettivo è usare gli input per predire gli output.

Un insieme di dati è rappresentato con **X**, che altro non è che un vettore di **input** (o feature). Ogni input ha un insieme di parametri.

Ad ogni input è associato un output, indicato con **Y** (se quantitativo) o **C** (se qualitativo). Per costruire il modello predittivo abbiamo bisogno di un insieme di dati che rappresentano il punto di partenza, chiamati **data training**.

Le variabili possono essere di tipo quantitativo o qualitativo:

- **quantitativo**: dati basati su valori calcolabili;
- **qualitativo**: dati basati su proprietà e attributi. Queste sono rappresentate con codici numerici, dove ogni valore può essere 0 od 1. Tali valori sono detti **target**.

es. un terreno può essere di vari tipi, da cui possiamo individuare un insieme finito  $C = \{\text{terreno rosso, cotone, misto...}\}$ . Ogni input avrà quindi un codice del tipo '1010...'. Anche per questo motivo le variabili qualitative sono dette *categoriche* o *discrete*.

Lo specifico **compito di predizione** dipende dal tipo di output:

- **regressione**: per predire output quantitativi;
- **classificazione**: per predire output qualitativi.

Entrambi possono essere visti come metodi per l'approssimazione di funzioni. Anche la tipologia di input può determinare il metodo di predizione da usare.

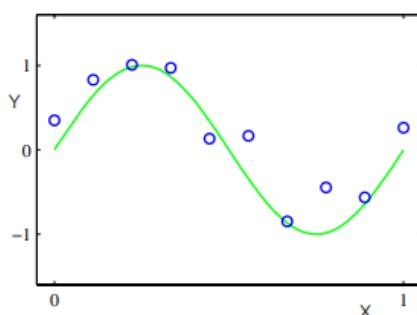
## Regressione

L'obiettivo della regressione è sfruttare il *training set*, costituito dalle coppie  $(x_i, y_i)$ , per predire i valori della variabile target Y, corrispondenti a nuovi valori di X.

In pratica, assumiamo che ci sia una relazione tra Y e X, esprimibile in forma generale come:

$$Y = f(X) + \epsilon$$

dove  $f$  è una funzione fissata ma non nota, ed  $\epsilon$  è un termine che rappresenta un errore casuale additivo.



In linea generale, supponiamo di avere una funzione polinomiale della forma:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j \text{ con } M, \text{ ordine del polinomio}$$

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - y_n\}^2$$

dove con  $\mathbf{w}$  denotiamo i coefficienti del polinomio, i cui valori saranno determinati dalla procedura di fitting.

Una semplice **funzione d'errore E** è la somma dei quadrati degli errori tra le predizioni  $y(x_n, \mathbf{w})$ , per ogni dato  $x_n$ , e il corrispondente target  $y_n$ .

Si vuole che tale funzione sia prossima allo zero, dunque possiamo risolvere il problema del fitting della curva scegliendo il valore  $\mathbf{w}$  per cui  $E(\mathbf{w})$  sia il più piccolo possibile.

Anche la scelta di **M** è fondamentale. In linea generale, maggiore è il grado del polinomio, più accurata sarà la predizione, tuttavia possiamo rischiare di creare overfitting.

(Ad esempio, se  $n=5$  ed  $M=5$ , significa che la funzione di predizione passerà per tutti i punti, tuttavia noi non vogliamo un'interpolazione e invece rischiamo solo di sballare tutto il modello).

Un polinomio di grado 0 non va bene, perché il modello non apprenderebbe nulla.

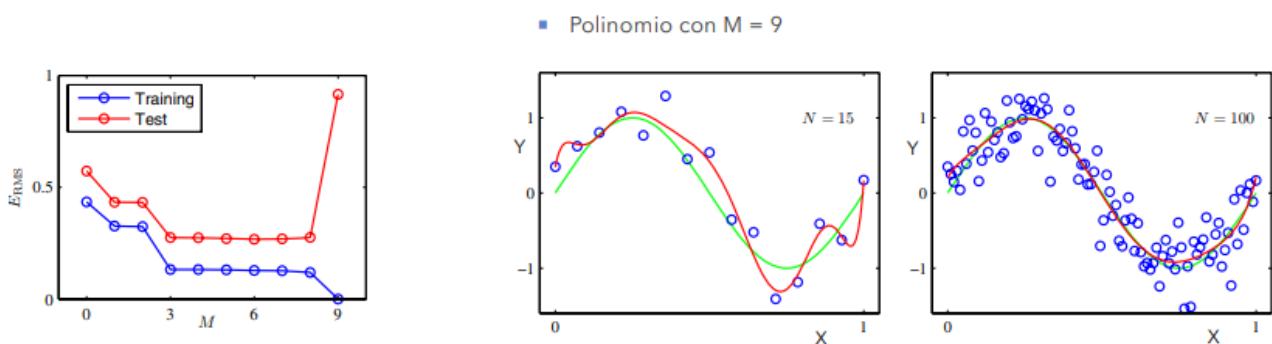
Un polinomio di grado 1 è già un miglioramento, ma non è comunque affidabile.

Lo scopo è raggiungere una buona **generalizzazione**.

Per capire se il modello generalizza bene si utilizza un nuovo insieme di dati, detto **test set**, che contiene solo variabili di input e non di output, e che dovrà ovviamente essere diverso dal training set.

Per ogni scelta di  $M$ , andrà a valutare l'errore. Possiamo notare che se  $M=n$ , nel caso del training set l'errore sarà nullo in quanto la funzione passa esattamente per quei punti, ma in generale per qualsiasi altro test set l'errore sarà enorme.

Per diminuire l'overfitting possiamo aumentare il numero di dati.



**nb:** un dataset non è altro che un file che contiene una matrice di numeri, compresi gli output. Inoltre, più variabili di input userò, più dimensioni avrò.

## Classificazione

L'obiettivo della classificazione è prendere un campione di input  $X$  e assegnarlo ad una di  $K$  classi discrete  $C_k$ .

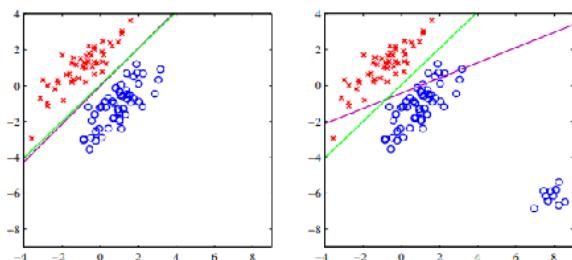
Geometricamente è come dividere lo spazio di input in regioni di decisione i cui confini sono chiamati **confini di decisione**. La classe di output da predire è modellata come una probabilità.

Se  $K=2$  si parla di classificazione binaria. Se  $K>2$  si parla di classificazione multiclasse.

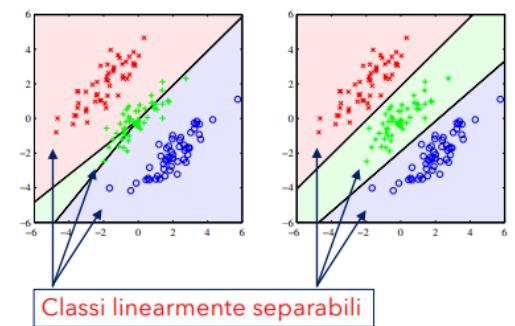
Le classi possono essere **linearmente separabili**, se possono essere divise da una retta, o **non linearmente separabili**, se ciò non avviene.

Ovviamente c'è sempre un margine di errore, che se mantenuto piccolo è considerato soddisfacente.

CLASSIFICAZIONE BINARIA: CONFINI DI DECISIONE

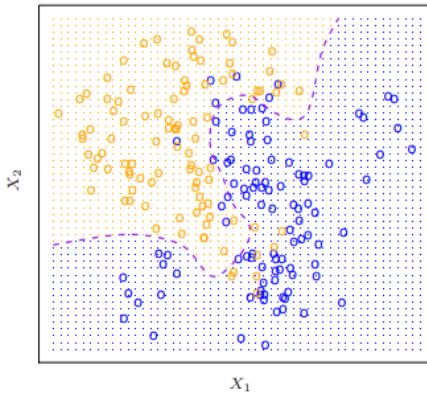


CLASSIFICAZIONE MULTI-CLASSE



CLASSIFICAZIONE

■ Classi **non linearmente**  
separabili



Esistono molti **approcci** alla classificazione, fra cui uno potente e diffuso consiste nel:

- modellare la distribuzione di probabilità condizionata  $p(C_k/x)$  in fase di inferenza;
- prendere le decisioni di classificazione sulla base di tale distribuzione.

### • Matrice di conclusione

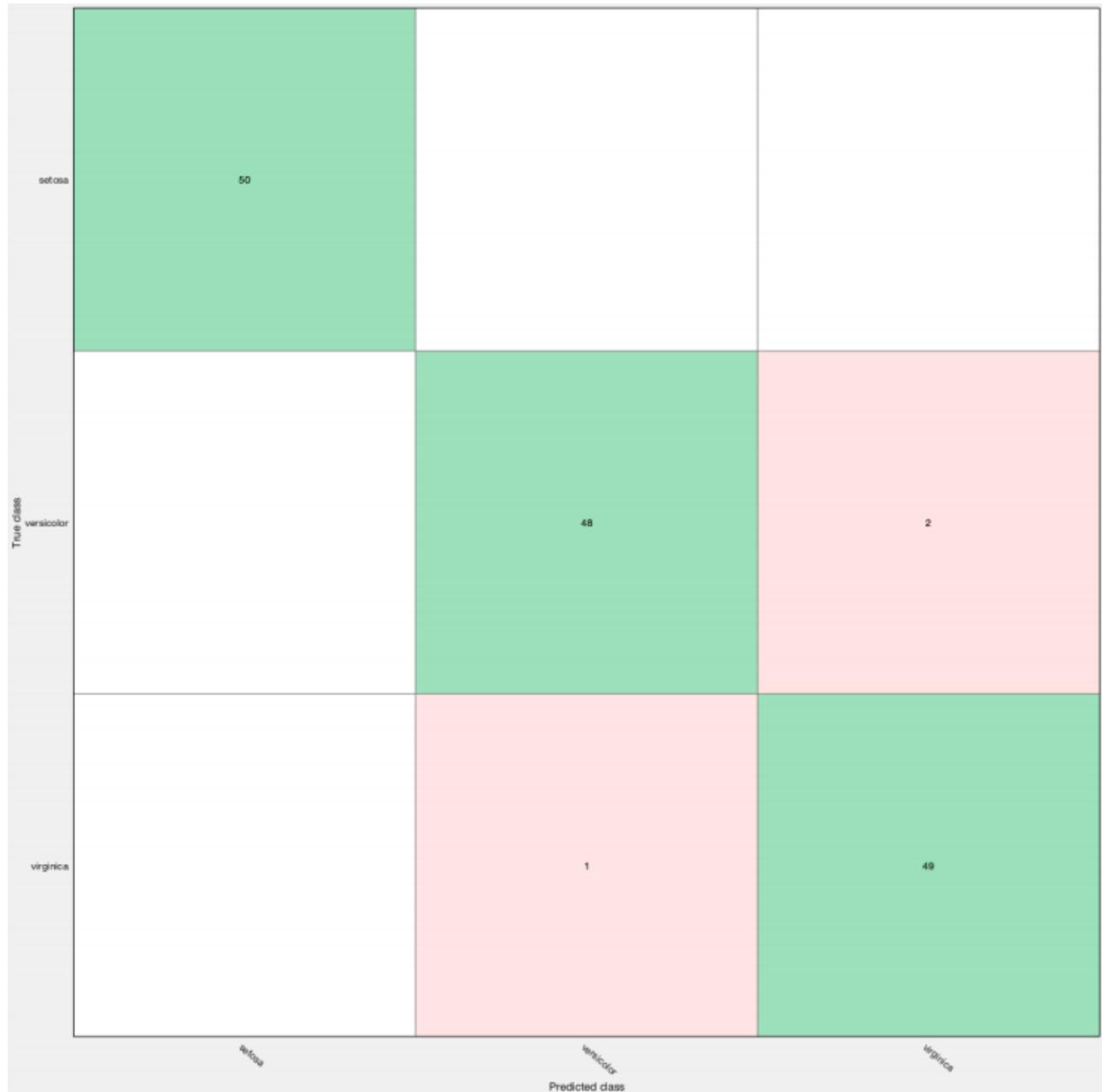
Una **matrice di conclusione** è una matrice che ha tante righe e colonne quante sono le classi da predire. Sulle righe rappresento i valori del dataset, sulle colonne rappresento il valore di risposta.

Le risposte corrette sono quelle sulla diagonale principale, tutte le altre sono sbagliate.

Per sapere la percentuale di correttezza faccio la somma dei risultati corretti fratto tutti i valori in input.

Un famoso esempio consiste nel classificare correttamente le tre specie di Iris, tenendo in considerazione le loro misure caratteristiche.

$X = (\text{L\_Petalo}, \text{A\_Petalo}, \text{L\_Sepalo}, \text{A\_Sepalo})$ ,  $Y = (\text{Setosa}, \text{Versicolor}, \text{Virginica})$ .



## LEZ 4 – Apprendimento non supervisionato

L'apprendimento non supervisionato è un tipo di apprendimento automatico, così chiamato in quanto nell'insieme dei dati di addestramento NON vengono fornite le soluzioni. Non siamo interessati alla predizione, poiché non abbiamo alcuna variabile di output.

L'obiettivo è cercare delle similitudini all'interno delle osservazioni (campioni), così da formare dei sottogruppi che prendono il nome di **cluster**.

Inoltre, è importante per capire la variabilità e la struttura di raggruppamento di un insieme di dati “senza etichetta” e può essere un'utile fase di pre-elaborazione per l'apprendimento supervisionato.

L'apprendimento non supervisionato è più complesso rispetto a quello supervisionato in quanto non abbiamo un riferimento. Ciò significa che ogni volta che applichiamo una tecnica di clustering ed otteniamo un risultato, non siamo certi che quel risultato sia corretto (la valutazione spetta agli esperti di quel dominio).

Tuttavia è utile quando si vuole valutare automaticamente qualcosa, come il sentimento generale di una recensione: è favorevole o no?

### Clustering

Il clustering riguarda un insieme di tecniche per individuare sottogruppi, o cluster, in un data set.

I gruppi saranno formati in modo tale che le osservazioni siano abbastanza simili tra di loro, tuttavia ciò vuol dire che dobbiamo definire cosa significa per un insieme di due o più osservazioni essere *simili* o *diverse*, e per far ciò possiamo usare diverse metriche.

Distinguiamo principalmente due tecniche di clustering: *k-means* e *gerarchico*

## • Clustering K-means

La tecnica di clustering K-means cerca di partizionare le osservazioni in un numero predefinito di Cluster.

Sia  $C_i$  un cluster, per  $i=1:k$ . Tutti gli insiemi soddisfano **due proprietà**:

- se faccio l'**unione** di tutti gli insiemi, ottengo tutte le osservazioni. Quindi, ogni osservazione appartiene almeno ad uno dei  $k$  cluster;
- se faccio l'**intersezione** di due cluster, per qualsiasi coppia di cluster, ottengo un'intersezione vuota. Quindi, i cluster non si sovrappongono e dunque non c'è alcuna osservazione appartenente a più cluster.

L'obiettivo che l'algoritmo si propone è **minimizzare** la varianza totale intra-gruppo (intra-cluster); ogni cluster viene identificato mediante un centroide.

L'algoritmo segue una procedura iterativa: inizialmente crea  $k$  partizioni e assegna i punti d'ingresso a ogni partizione casualmente; quindi calcola il centroide di ogni cluster; costruisce in seguito una nuova partizione associando ogni punto d'ingresso al cluster il cui centroide è più vicino ad esso; infine vengono ricalcolati i centroidi per i nuovi cluster e così via finché l'algoritmo non converge.

Ad ogni passo è garantito che il valore della **funzione obiettivo** decresce, non è tuttavia garantito che fornisca il minimo globale (funzione obiettivo *non convessa*).

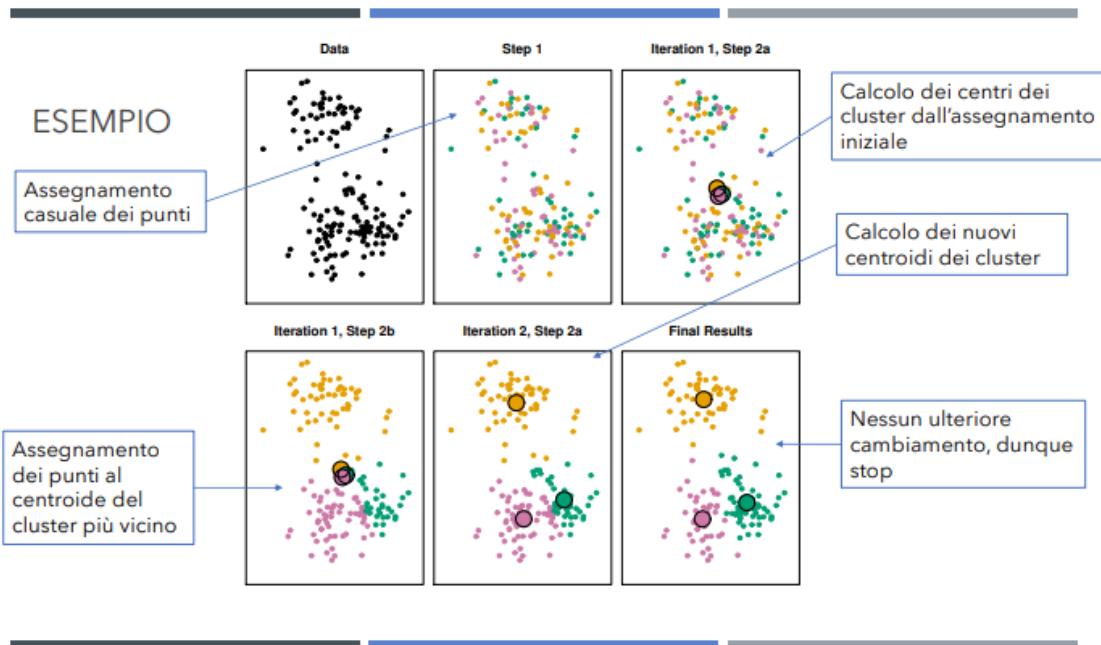
È importante usare valori di partenza diversi ogni volta in quanto il K-means può restare bloccato in *ottimi locali* e non trovare la migliore soluzione.

La **funzione obiettivo** è data combinando due formule:

1. vogliamo partizionare le osservazioni in  $K$  cluster in modo che la varietà intra-cluster, sommata su tutti i  $K$  cluster, sia quanto più piccola possibile.
2. distanza euclidea, dove  $|C_k|$  denota il numero di osservazioni nel  $k$ -esimo cluster.

$$\min_{C_1, \dots, C_K} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$$

**Funzione obiettivo**



## DETTAGLI FIGURA

- Il progredire dell'algoritmo K-means con  $K = 3$ 
  - Alto a sinistra:** osservazioni (data set)
  - Alto al centro:** al passo **1** dell'algoritmo, ogni osservazione è assegnata casualmente ad un cluster
  - Alto a destra:** al passo **2.1**, sono calcolati i centroidi dei cluster (dischi colorati). All'inizio i centroidi sono quasi completamente sovrapposti a causa dell'assegnamento iniziale casuale
  - Basso a sinistra:** Al passo **2.2**, ogni osservazione è assegnata al centroide più vicino
  - Basso al centro:** è eseguito nuovamente il passo 2.1 che determina i nuovi centroidi dei cluster
  - Basso a destra:** i risultati ottenuti dopo 10 iterazioni

## • Clustering Gerarchico

La tecnica di clustering Gerarchico fornisce una rappresentazione ad albero delle osservazioni, chiamata dendrogramma, che permette di vedere i raggruppamenti ottenuti per ogni possibile numero di cluster da 1 a n.

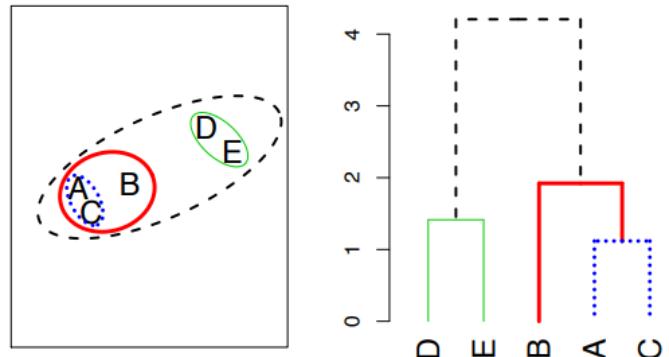
In questo caso, non conosciamo il numero di cluster che cerchiamo.

Il tipo più comune di clustering gerarchico è quello bottom-up (o agglomerativo), così detto in quanto il dendrogramma è costruito a partire dalle foglie e combinando i cluster fino alla radice.

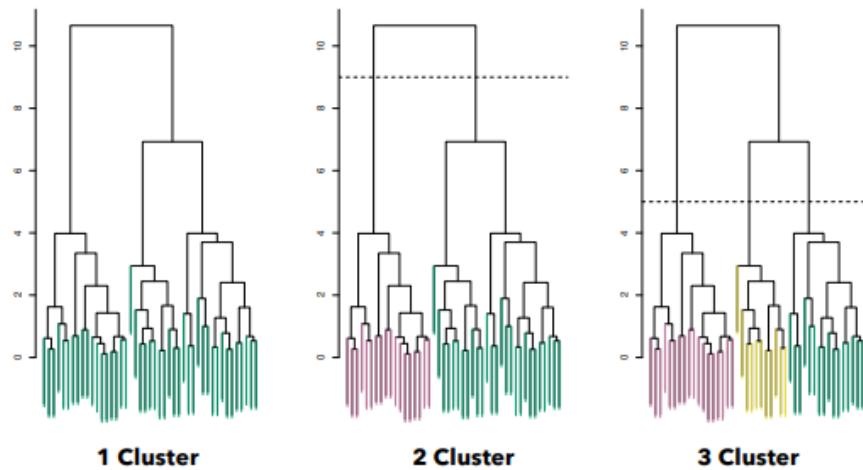
L'algoritmo comincia dall'inserimento di ciascun elemento in un cluster differente e si procede quindi all'accorpamento graduale di cluster a due a due.

Termina quando tutti gli elementi sono in un unico cluster.

esempio: A e C sono i punti più vicini, dunque li accorpo in un unico cluster. Il cluster [A,C] è vicino al punto B, e dunque li accorpo a loro volta. Dall'altra parte i punti D ed E sono vicini fra di loro, e li accorpo. Rimangono infine i cluster [[A,C],B] e [D,E], che accorpo in un unico cluster finale.



Per scegliere i cluster, tracciamo una linea che attraversa il dendrogramma. Possiamo formare un qualsiasi numero di cluster a seconda del punti in cui “tagliamo”.



Per definire la **dissimilarità**, o *collegamento* (linkage), abbiamo 4 opzioni:

- **complete linkage**: distanza più grande tra le osservazioni;
- **single linkage**: distanza più piccola tra le osservazioni;
- **average linkage**: distanza media tra le osservazioni;
- **centroide**: distanza tra i centrodi delle osservazioni.

In base al metodo di linkage possiamo ottenere risultati molto diversi: i complete e average linkage costruiscono cluster di dimensioni uguali, mentre il single linkage porta a cluster estesi.

Finora è stata usata la distanza Euclidea, che tuttavia non sempre restituisce il miglior risultato. Un'alternativa è la *distanza basata su correlazione*, che considera due osservazioni simili se le loro feature sono altamente correlate.

## LEZ 5 – Formazione delle immagini

L'occhio umano ha una forma pressoché sferica, con un diametro complessivo di circa 20mm. È circondato da tre membrane: la **cornea** e la **sclera**, che lo coprono dall'estero, la **coroide** e la **retina**.

La **sclera** è la continuazione della **cornea**. Si tratta di una membrana opaca che copre la parte rimanente. Questa membrana contiene una rete di vasi sanguigni che costituiscono la principale fonte di nutrimento dell'occhio.

La **coroide** si trova sotto la sclera, ed è di colore scuro. Ha lo scopo di ridurre la quantità di luce esterne che penetra nel bulbo oculare.

Fra l'iride e il corpo vitreo si trova il **cristallino**, il quale è una lente biconvessa che, grazie all'azione del muscolo ciliare, è capace di modificare la sua forma per regolare "automaticamente" la messa a fuoco dei raggi luminosi sulla retina, in base alla vicinanza dell'oggetto (distanza focale). Invece, la distanza tra il centro del cristallino e la retina è fissa.

La **retina** è la membrana più interna dell'occhio. Si tratta della superficie sulla quale viene impressa l'immagine, e sulla quale sono distribuiti i **coni** e i **bastoncelli**.

I coni e i bastoncelli sono i recettori di luce sulla retina:

- i **coni** sono molto sensibili ai colori, ed anche per questo sono principalmente attivi in condizioni di luce. Ciascun cono è connesso ad una cellula nervosa, e dunque hanno un'elevata capacità di discriminare i dettagli.

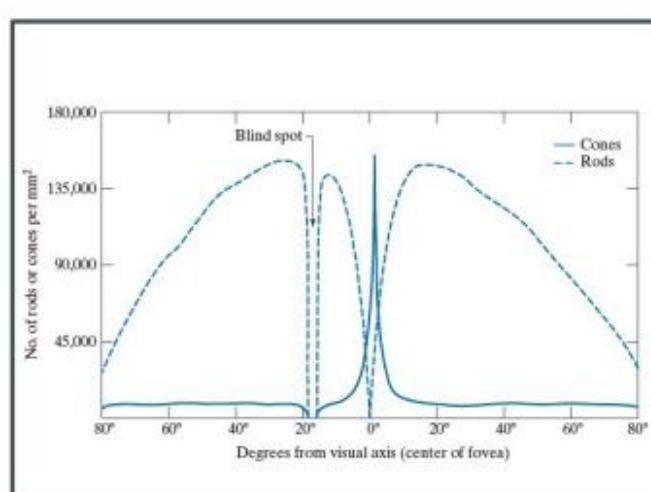
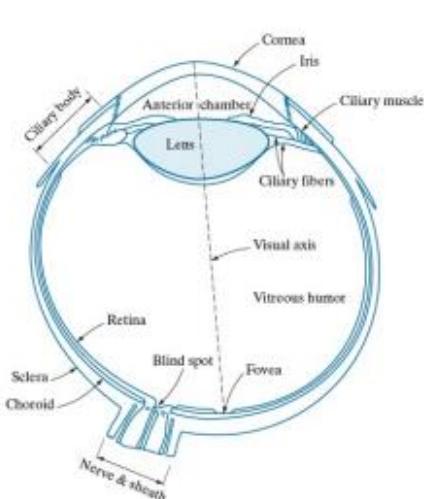
Sono circa 6-7 milioni posizionati intorno alla parte centrale della retina chiamata **fovea**.

La visione legata ai coni è chiamata *fotopica*.

- i **bastoncelli** sono poco sensibili ai colori, ed anche per questo sono principalmente attivi in condizioni di buio. I bastoncelli sono connessi alle cellule nervose con una relazione 1 a n, e dunque hanno una scarsa capacità di discriminare i dettagli. Sono circa 75-150 milioni distribuiti su tutta la retina, ma più densamente intorno al centro dell'occhio.

La visione legata ai bastoncelli è chiamata *scotopica*.

Vicino alla fovea è presente un punto nel quale sono assenti recettori, detto *punto cieco*.



## Luminosità e Intensità percepita

La luminosità percepita è una funzione logaritmica dell'intensità di luce.

L'occhio umano è in grado di percepire un intervallo molto grande di livelli di intensità di luce, pari a  $10^{10}$  valori discriminati. Tuttavia, tale intervallo non è ricoperto simultaneamente, ma solo un sottoinsieme di livelli vicini alla luminosità della scena può essere discriminata.

La luminosità percepita non è tuttavia solo in funzione dell'intensità. Si pensi alle bande di Mach, dove nel confine tra due bande sembra comparire una banda di "rinforzo" più scura, o si pensi alle illusioni ottiche, che altro non sono che percezioni che rappresentano in modo erroneo la realtà osservata.

## Immagini ad intensità

Il nostro obiettivo sono le immagini ad intensità, le quali misurano la quantità di luce che incide un dispositivo fotosensibile.

Consideriamo tre tipi di parametri: geometrici, ottici e fotometrici.

## Ottica di base

La messa a fuoco di tutti i raggi da un punto di una scena su un singolo punto dell'immagine può essere ottenuta in due modi:

1. riducendo l'apertura della camera ad un punto, chiamato pinhole.  
Solo un raggio proveniente da un qualsiasi punto della scena può entrare nella camera;
2. introducendo un sistema ottico composto da lenti.  
I raggi provenienti da uno stesso punto 3D convergono su un singolo punto dell'immagine.



## Geometria della formazione dell'immagine

La **telecamera pinhole** rappresenta il modello geometrico più semplice della formazione dell'immagine.

Sia  $P$  un punto della scena di coordinate  $(X, Y, Z)$  e sia  $P'$  la sua proiezione sul piano immagine di coordinate  $(X', Y', Z')$ .

Sia  $f$  la *distanza focale*, cioè la distanza del foro  $O$  (centro di proiezione) dal piano immagine.

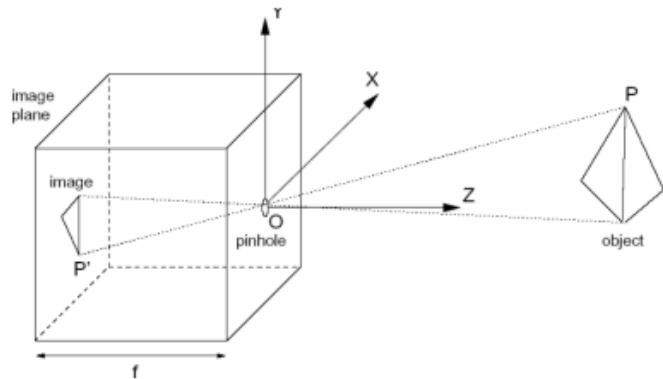
Allora, dalla similitudine dei triangoli si ottiene i punti  $P, O, P'$  sono collineari, quindi:

- $\overrightarrow{OP'} = \lambda \overrightarrow{OP}$  per qualche  $\lambda \in \mathbb{R}$ , da cui

- $$\begin{cases} X' = \lambda X \\ Y' = \lambda Y \\ Z' = -f \end{cases} \Rightarrow \lambda = \frac{X'}{X} = \frac{Y'}{Y} = \frac{-f}{Z}$$

da cui si ottiene

$$\begin{cases} X' = -f \frac{X}{Z} \\ Y' = -f \frac{Y}{Z} \end{cases}$$



Come si evince dal segno meno, l'immagine è invertita rispetto alla scena.

Tuttavia se modelliamo la proiezione prospettica ponendo il piano immagine davanti al centro di proiezione, possiamo eliminare tale segno.

La  $Z$  al denominatore determina l'*effetto di scorci*, per cui la dimensione dell'immagine di un oggetto varia in ragione della sua distanza dall'osservatore.

Nella **proiezione prospettica** le linee parallele convergono in un punto sull'orizzonte, detto *punto di fuga*.

Se l'oggetto inquadrato è relativamente sottile (rispetto alla sua distanza media dalla telecamera), la proiezione prospettica si può approssimare con la *proiezione ortografica*.

ESEMPIO: IMMAGINE PROSPETTICA ED ORTOGRAFICA



*Dimostrazione:*

Possiamo descrivere una linea che nella scena attraversa il punto  $(X_0, Y_0, Z_0)$  in direzione  $(U, V, W)$  come un insieme di punti:  $(X_0 + \lambda U, Y_0 + \lambda V, Z_0 + \lambda W)$  dove  $-\infty < \lambda < +\infty$

La proiezione di un qualsiasi punto  $P_\lambda$  da tale linea sul piano di immagine è data da:

$$P_\lambda = \left( f \frac{X_0 + \lambda U}{Z_0 + \lambda W}, f \frac{Y_0 + \lambda V}{Z_0 + \lambda W} \right) \quad \text{Per } \lambda \rightarrow \pm\infty \text{ otteniamo } P_\infty = \left( f \frac{U}{W}, f \frac{V}{W} \right), \text{ se } W \neq 0$$

dunque, le linee che hanno la stessa direzione condividono lo stesso punto di fuga.

## Lenti sottili

Una lente è in grado di raccogliere più luce rispetto ad uno spillo, tuttavia non tutta la scena può essere messa a fuoco allo stesso tempo.

Come ottica del sistema di acquisizione terremo conto delle *lenti sottili*, sebbene questa sia più complessa in quanto composta da più lenti.

Una lente sottile ha diverse **proprietà**:

- i raggi paralleli all'asse ottico incidenti sulla lente vengono rifratti in modo da passare per un punto dell'asse ottico chiamato *fuoco F*;
- i raggi che passano per il *centro C* della lente rimangono inalterati;
- la distanza del fuoco F dal centro C prende il nome di *distanza focale D*.

Dato un punto P della scena, è possibile costruirne graficamente l'immagine P' (o *punto coniugato*) servendosi di due raggi particolari che partono da P:

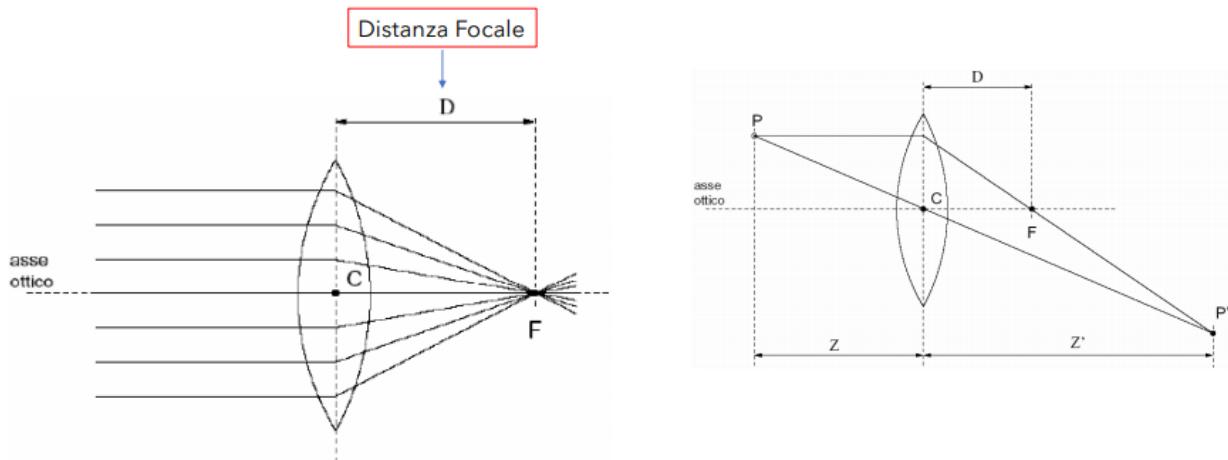
1. il raggio parallelo all'asse ottico, che dopo la rifrazione passa per F;
2. il raggio che passa inalterato per C.

Se teniamo in considerazione la *similarità dei triangoli*, otteniamo l'**equazione della lente sottile**:

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{D}$$

dove Z rappresenta la distanza fra P e C, e Z' rappresenta la distanza fra C e P'. Cambiando Z si ottiene l'immagine sfocata del punto (**cerchio di confusione**).

Più un soggetto è lontano dal punto di messa a fuoco, più grandi saranno i cerchi che il soggetto forma sul sensore. La stessa cosa avviene se il soggetto è troppo vicino.



**nb:** la lunghezza focale della lente è una cosa diversa dalla lunghezza focale della telecamera pinhole.

## Fotometria della formazione dell'immagine

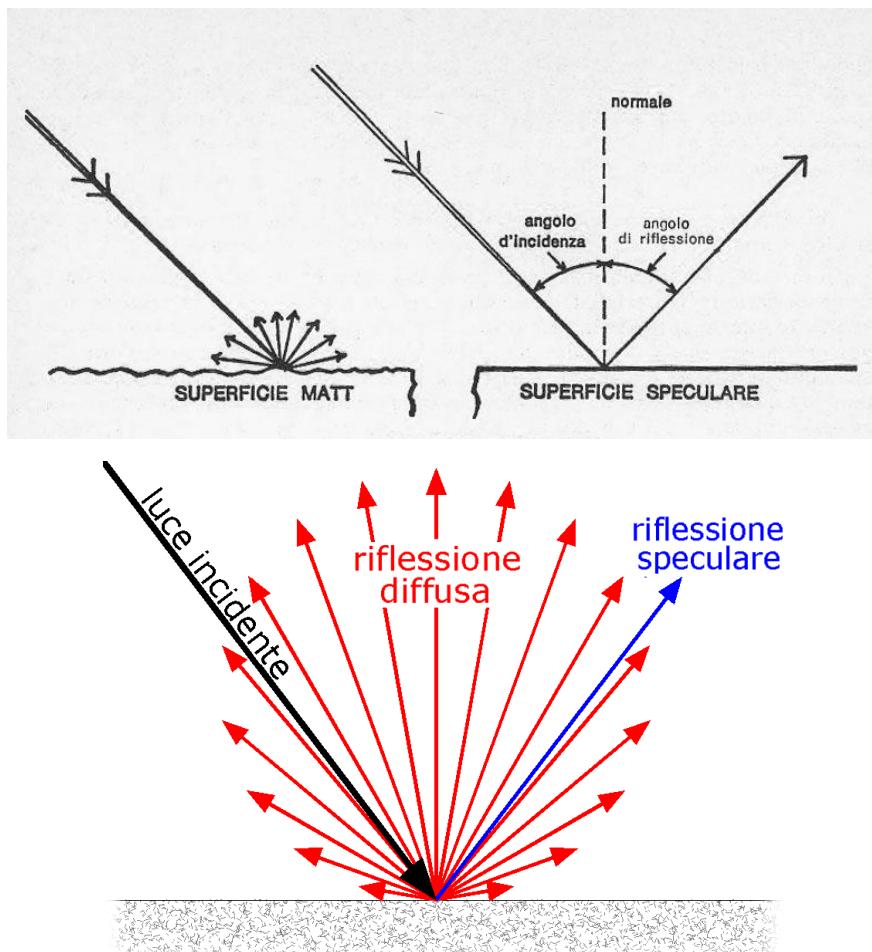
La **luminosità di un pixel** è proporzionale alla quantità di luce che la superficie riflette verso la telecamera. Questa a sua volta dipende dal modo in cui la superficie riflette la luce e dalla posizione, tipo e distribuzione delle sorgenti luminose.

La **riflessione** è un fenomeno complesso, di cui possiamo distinguere due casi estremi: *diffusione* e *riflessione speculare*:

- **diffusione:** la riflessione avviene in varie direzioni.  
Il raggio di luce incidente non viene rimandato indietro ad un angolo determinato, ma viene diffuso su molte direzioni.
- **speculare:** la riflessione avviene in un'unica (o quasi) direzione.  
Il raggio di luce incidente viene riflesso in una direzione con angolo di incidenza uguale a quello di riflessione.

In realtà le superfici presentano un comportamento che è una combinazione di diffusione e riflessione speculare.

La riflessione speculare, così come definita, è un comportamento che ci aspetteremmo solo in uno *specchio perfetto*.



**nb:** le ombre sono raramente completamente nere, questo perché la luce può provenire anche dalla riflessione della luce su altri oggetti.

## LEZ 6 – Fondamenti (1)

### Radiazione elettromagnetica

Lo **spettro elettromagnetico** può essere considerato come un'onda, o come un flusso di particelle prive di massa (fotoni).

Lo spettro può essere espresso in termini di *lunghezza d'onda* (metri), di *frequenza* (hz) o di *energia* (elettronvolt). Indicate rispettivamente con  $\lambda$ ,  $v$ ,  $E$ .

La **lunghezza d'onda** è data dal rapporto fra la frequenza e la velocità della luce:  $\lambda = \frac{c}{v}$

L'**energia** è data dal prodotto fra la costante di Planck e la frequenza:  $E = h\nu$

Notiamo come l'Energia sia direttamente proporzionale alla frequenza; quindi una frequenza più alta (che corrisponde ad una minore lunghezza d'onda) trasporta una maggiore energia per fotone.

La **luce visibile** è un particolare tipo di radiazione elettromagnetica che rappresenta solo una piccola frazione dell'intero spettro, che va dal violetto al rosso.

Nessun colore si interrompe bruscamente, ma ciascuno di essi si mescola gradualmente al successivo.

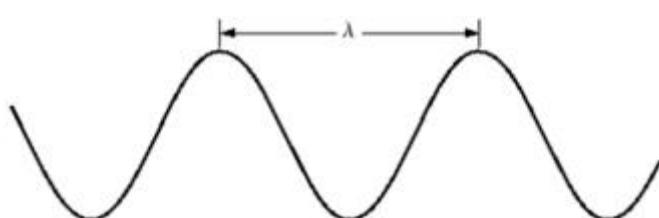
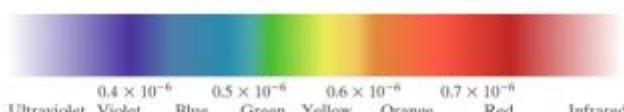
La luce composta di onde da una sotto banda sufficientemente piccola tale che non contiene più di un colore è chiamata monocromatica, ed è caratterizzata solo dalla sua *intensità* (o livelli di grigio).

Gli oggetti possono essere visti solo attraverso la luce riflesse, in quanto solo le frequenze riflesse possono essere individuate.

Infatti, non è possibile osservare un oggetto che non è investito da un fascio luminoso, e quelle che vediamo non sono altro che le frequenze riflesse (le altre sono assorbite).

Oltre alla sua **frequenza** (o colore), la luce può essere caratterizzata da:

- **radianza**: *energia totale* emessa dalla fonte di luce (W, watt);
- **luminanza**: *energia percepita* dall'osservatore dalla fonte di luce (lm, lumen);
- **luminosità**: *percezione soggettiva*, impossibile da misurare.



## Acquisizione delle immagini

\*processo di acquisizione visto nell'introduzione.

Un **sistema** di acquisizione di immagini consiste di **tre componenti fondamentali**:

- una *telecamera*: composta dall'ottica e da una matrice di sensori CCD/CMOS, che costituiscono l'immagine;
- un *frame grabber*: è la "scheda di acquisizione" che cattura singoli fotogrammi digitali da un segnale video analogico (o da un flusso video digitale);
- un *calcolatore host*: acquisisce l'immagine trasferendola dal frame buffer a una memoria centrale.

Il **CCD** (Charged-Coupled Device) è una matrice NxM di celle rettangolari fotosensibili, ciascuna delle quali converte l'intensità della radiazione luminosa incidente in un potenziale elettrico.

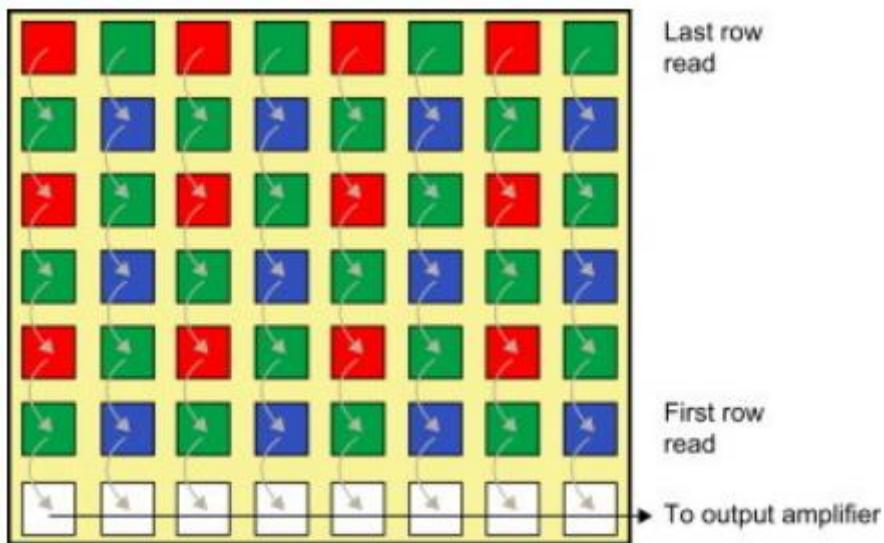
L'uscita della telecamera a CCD è un segnale elettrico analogico, ottenuto leggendo il potenziale degli elementi della matrice CCD per righe.

Ogni CCD misura circa 1cm<sup>2</sup> e contiene circa 500.000 elementi.

In **linea di principio**, dopo l'esposizione:

1. le cariche sulla prima riga vengono trasferite sul registro di lettura;
2. i segnali vengono inviati ad un amplificatore, e quindi ad un convertitore A/D.

Ad ogni lettura, le cariche sul registro di lettura sono eliminate ed entra la riga successiva; le cariche su ogni riga sono "accoppiate" a quelle sulla riga superiore, quindi quando la prima si sposta verso il basso, anche la successiva si sposta verso il basso.



Il segnale (analogico) viene letto dal frame grabber che lo digitalizza, convertendolo in una matrice NxM (es. 512x512) di valori interi (es. 0... 255), memorizzati in un'opportuna area di memoria (frame buffer), dove ogni elemento è detto **pixel**.

Non è detto che la dimensione della matrice CCD sia la stessa dell'immagine NxM, ma possiamo assumerlo per comodità.

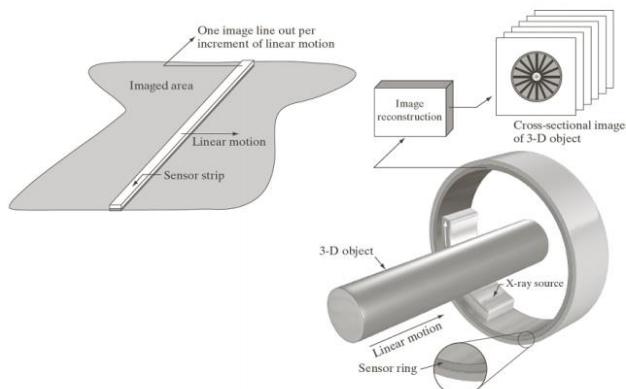
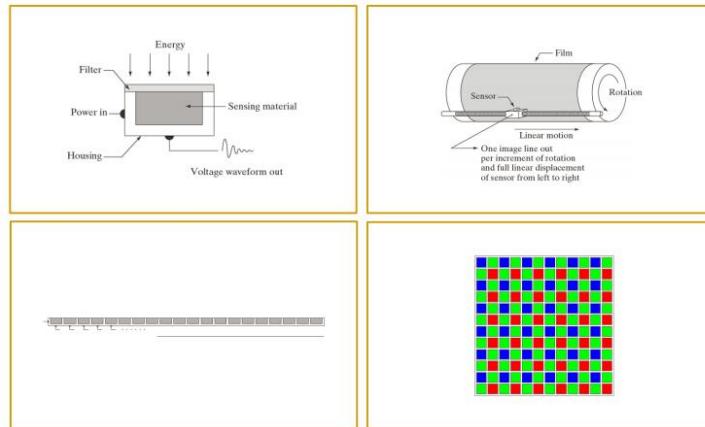
Di solito per creare un'immagine bidimensionale utilizzando un singolo sensore, si compiono delle traslazioni del sonore rispetto all'area che si vuole acquisire. Possiamo distinguere diverse **strategie**:

- **singoli sensori**, che fanno uso di congegni meccanici per spostare il sensore rispetto alla scena (si pensi a un **rullo**);  
L'immagine viene acquisita attraverso l'incremento della rotazione di una pellicola posta sul rullo, e un completo spostamento lineare del sensore da sx a dx. Lo spostamento combinato del sensore e della pellicola mi da una matrice.
- **sensori disposti in linea**, che fanno uso del movimento di un mezzo per ottenere la seconda dimensione (si pensi agli aerei);
- **array 2D di sensori**, che richiedono ulteriori elaborazione e che vengono usati sia nelle macchine digitali che negli scanner tomografici.

Esistono diverse tipologie di **dispositivi di acquisizione**, ed ognuno ha caratteristiche diverse in quanto possono avere sensori diversi per captare segnali diversi.

In generale, un sensore singolo è costituito da un materiale sensibile alle onde elettromagnetiche di una certa banda. Riceve l'energia proveniente da un'onda e risponde inviando un segnale elettrico. È anche possibile applicare dei filtri davanti al sensore per migliorarne la selettività.

Un altro dispositivo di imaging è ad. es quello usato per le tac, dove l'anello è la sorgente luminosa (raggi X). Con questa procedura si producono tante immagini 2D di sezioni trasversali dell'oggetto, da cui potrà ricostruire la forma 3D.



## Modello di creazione di un'immagine

Ricordiamo che:

Un'immagine può essere definita come una funzione bidimensionale  $f(x,y)$ , dove  $x$  e  $y$  sono le coordinate spaziali (sul piano), e l'ampiezza di  $f$  in ogni coppia di coordinate  $(x,y)$  viene chiamata *intensità*, o livelli di grigio in quel punto.

Di conseguenza  $f(x,y)$  deve essere finita e diversa da zero. cioè  $0 < f(x,y) < \infty$

La funzione  $f(x,y)$  può essere caratterizzata da due componenti:

1. **illuminazione**: quantità di luce incidente sulla scena;  
 $i(x,y)$ ; dove  $0 < i(x,y) < \infty$ ;
2. **riflettanza**: quantità di luce riflessa dagli oggetti presenti nella scena.  
 $r(x,y)$ ; dove  $0 < r(x,y) < 1$  (0 totale assorbimento, 1 totale riflessione).

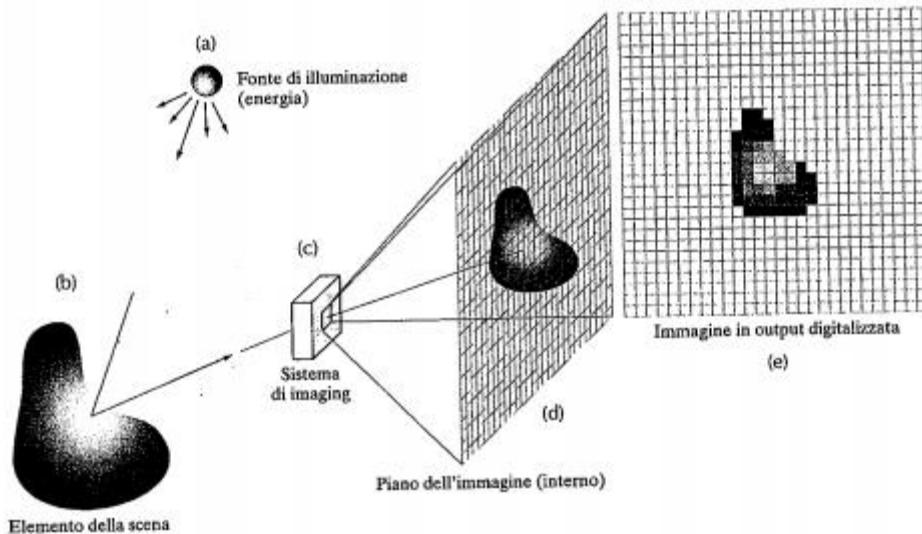
Dunque avremo che:  $f(x,y) = i(x,y) r(x,y)$

Nella pratica, l'intensità  $f(x,y)$  è compresa in una **scala di grigio** [ $L_{\min}, L_{\max}$ ], convenzionalmente scalati in  $[0, L-1]$  ([nero, bianco]);

dunque:  $L_{\min} < f(x,y) < L_{\max}$

dove:  $L_{\min} = i_{\min} r_{\min}$  e  $L_{\max} = i_{\max} r_{\max}$

Esempio di acquisizione delle immagini digitali:



**Figura 2.15** Esempio del processo di acquisizione delle immagini digitali: (a) fonte di energia ("illuminazione"); (b) un elemento della scena; (c) sistema di imaging; (d) proiezione della scena sul piano dell'immagine; (e) immagine digitalizzata.

## Campionamento e quantizzazione

Per creare un'immagine abbiamo bisogno di convertire i dati continui in una forma digitale.

Ricordo:

La **discretizzazione** di un segnale analogico si realizza per mezzo di un'operazione di **campionamento** ed una di **quantizzazione**:

- il **campionamento** discretizza l'asse del **tempo**, cioè suddivide l'asse a intervalli regolari e considera il segnale solo in quei punti;
- la **quantizzazione** discretizza l'asse dei **valori**, cioè approssima il valore dei campioni a un certo numero prefissato di valori.

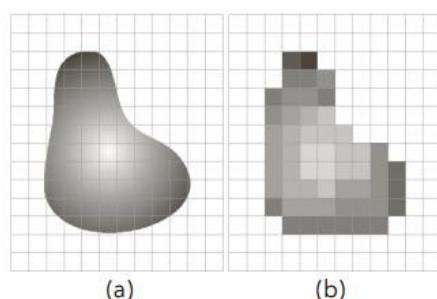
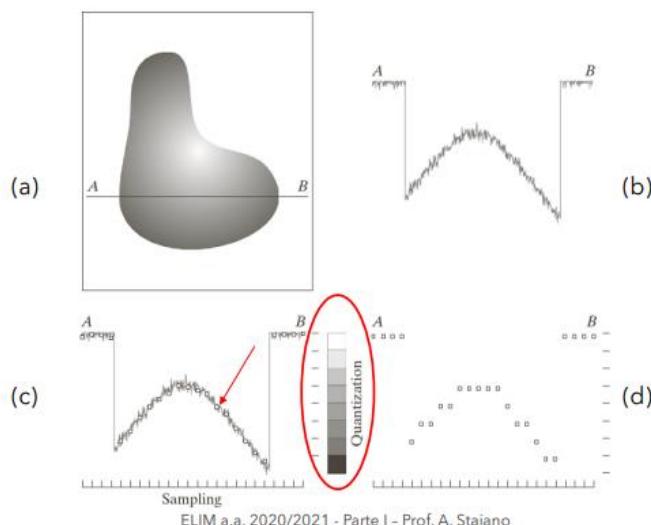
I livelli di intensità dell'immagine possono subire delle *variazioni casuali*, dovute al rumore

Esempio:

a/b) discretizzazione: prendiamo il segnale analogico e lo spaziamo sull'asse delle x (ogni quadratino rappresenta un campione). Notare che sono presenti delle piccole oscillazioni dovute al rumore;

c/d) quantizzazione: il range continuo è suddiviso in 8 campioni.

Il livello di grigio è ottenuto come valore medio della regione corrispondente nel campionamento, approssimato al livello di grigio più vicino tra quelli disponibili.



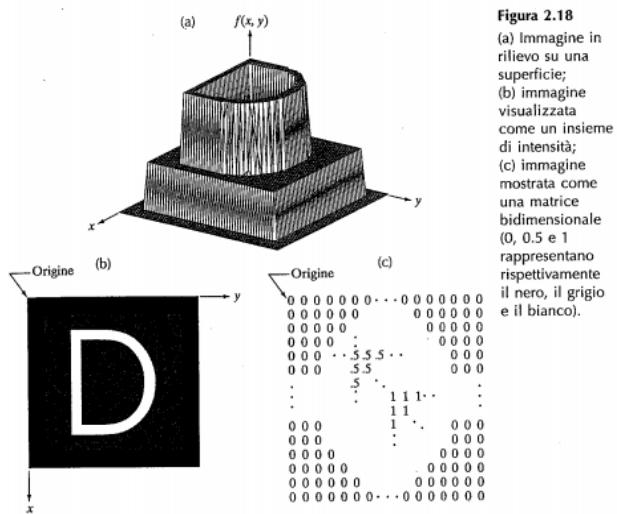
## Rappresentazione delle immagini digitali

Un'immagine discretizzata può essere trattata in più modi: può essere rappresentata nello spazio, ma di solito è più conveniente rappresentarla nel piano.

L'immagine  $f(x,y)$  è rappresentata come una matrice  $M \times N$  ad  $L$  valori discreti,  
 $x$  è convenzionalmente associata alle coordinate discrete  $\{0 \dots M-1\}$ ,  
 $y$  alle coordinate discrete  $\{0 \dots N-1\}$   
 $f(x,y)$  alle coordinate discrete  $\{0 \dots L-1\}$

Per ragioni pratiche,  $L$  è generalmente una potenza di 2:  $L = 2^k$ .  
In questo modo, ogni pixel è rappresentato usando  $k$  bit.

Di fatto la nostra immagine è una matrice. Può essere anche rappresentata in forma matriciale.



$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1, 0) & f(M-1, 1) & \cdots & f(M-1, N-1) \end{bmatrix} \quad (2.4-1)$$

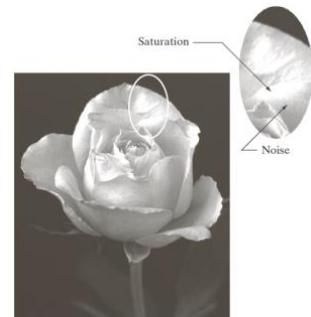
$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad (2.4-2)$$

Il sensore esegue una misurazione dell'intensità di luce, ma poiché è una misura strumentale, il sensore **non è privo di errori** (limitazione del campionamento).

Il **range dinamico** di un sistema di imaging è il rapporto tra l'intensità massima misurabile e il minimo livello di intensità rilevabile nel sistema. In quanto il campionamento è **limitato**, possono crearsi fenomeni come la saturazione.

- La **saturazione** è il fenomeno per cui tutte le intensità oltre una data soglia sono rappresentate come bianco;
- il **rumore** è l'errore di misurazione del sensore. Può essere individuato nelle regioni più scure, dove, invece di essere neri, alcuni pixel sono grigio scuro.

Il **contrasto** è definito come la differenza di intensità tra il più alto e il più basso livello di intensità di un'immagine.



Il **numero di bit di un'immagine** è calcolato come:  $b = MN \log_2 L$ .  
Per  $L = 2^k$ ,  $b = MNk$ .

Ad esempio, un'immagine  $1024 \times 1024$  per 256 livelli di grigio sono circa 8MB.

## Risoluzione spaziale e dell'intensità

La **risoluzione spaziale** è la dimensione del più piccolo dettaglio che può essere riconosciuto nell'immagine. Spesso, la risoluzione è misurata in dpi (dot per inch).

Infatti, la risoluzione "stessa" non è tutto, in quanto solo perché un'immagine ha grandi dimensioni non significa che sia anche ben definita.

La **risoluzione dell'intensità** (r. del contrasto) riguarda il numero di livelli di grigio. Un basso numero di livelli di grigio in una regione quasi uniforme può causare l'effetto del *falso contorno*.

## Interpolazione delle immagini

Fondamentalmente, l'**interpolazione** è il processo mediante il quale si utilizzano dei dati noti di un segnale per stimarne apparentemente i valori in "locazioni" sconosciute.

Si tratta dello strumento di base utilizzato in operazioni quali l'ingrandimento, il rimpicciolimento, la rotazione e le correzioni geometriche.

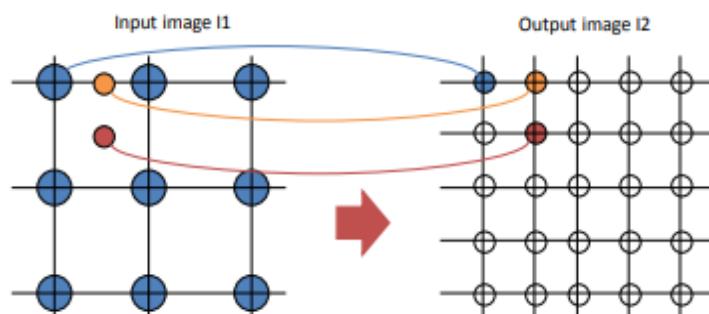
Distinguiamo 3 tipi di **interpolazione**:

- **nearest neighbour**: prendiamo il valore del dato più vicino;
- **bilineare**: prendiamo una combinazione dei 4 più vicini;
- **cubica**: prendiamo i valori dei 16 più vicini.

Ovviamente, la *cubica* è quella più costosa ma che offre i risultati migliori, seguita dalla *bilineare*, seguita dalla *nn* (la più grezza).

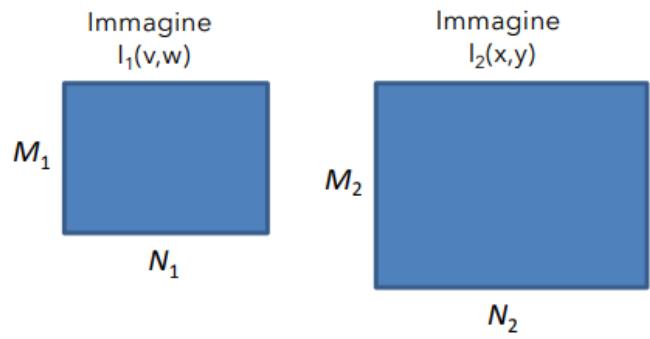
Fondamentalmente partiamo da una matrice dove in ogni posizione intera è presente un valore. Supponiamo di voler effettuare uno zoom, quindi passiamo come input l'immagine originale e otterremo in output quella scalata.

Scandiamo i pixel in output e per ognuno calcoliamo il valore dell'immagine di input dalla posizione corrispondente. Se la posizione nell'immagine di input non è intera, interpoliamo.



## Funzione di mapping

- Assumiamo
  - Immagine  $I_1$  di dimensioni  $M_1 \times N_1$
  - Immagine  $I_2$  di dimensioni  $M_2 \times N_2$
- Allora
  - $x = c_x \times v$
  - $y = c_y \times w$
  - Dove  $c_x, c_y$  sono i fattori di scala
    - $c_x = N_2 / N_1$
    - $c_y = M_2 / M_1$
  - In realtà vogliamo la funzione inversa
    - $v = x / c_x$
    - $w = y / c_y$



## Esempio interpolazione bilineare

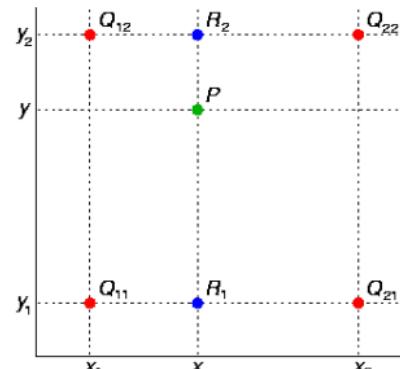
Vogliamo ottenere  $P$ . Interpoliamo prima rispetto all'asse  $x$  (ottengo  $R_1$  e  $R_2$ ) e poi rispetto all'asse  $y$  (ottenendo così  $P$ ).

- Il valore in  $(x,y)$  è  $f(x,y) = ax + by + cxy + d$ 
  - $a, b, c, d$  sono i coefficienti determinati dai 4 più vicini
- Equivalente a fare l'interpolazione in una dimensione e poi nell'altra
- Per trovare un valore in  $P$ , se conosciamo i valori nelle  $Q$ 
  - Prima interpoliamo orizzontalmente per trovare i valori in  $R_1$  e  $R_2$ 

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$
  - Poi interpoliamo verticalmente
 
$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)$$
- Negli angoli  $(0,0), (0,1), (1,0)$  e  $(1,1)$

$$f(x,y) \approx f(0,0)(1-x)(1-y) + f(1,0)x(1-y) + f(0,1)(1-x)y + f(1,1)xy$$



## Relazioni tra Pixel

La struttura di un'immagine digitale permette di definire alcune relazioni basilari tra pixel, che possono essere utili in vari casi pratici.

Alcune operazioni considerano gruppi di pixel che condividono le stesse caratteristiche, o che comunque sono legate da qualche caratteristica peculiare.  
(es. in un immagine sat. verificare quali pixel blu appartengono al fiume e quali no).

**Strumenti** che possono essere molto utili:

- una relazione di **vicinanza** tra pixel;
- una relazione di **adiacenza** tra pixel;
- una relazione di **connettività** tra pixel;
- una misura di **distanza sul dominio dell'immagine**.

(ciò vale anche per sottoinsiemi di pixel).

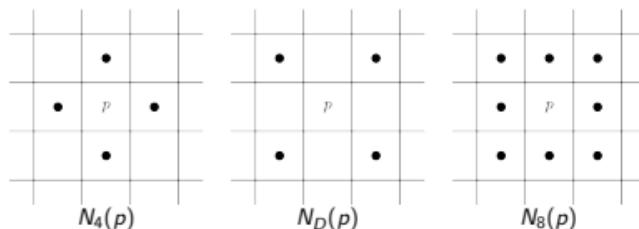
### • Vicinanza

La relazione di *vicinanza* è definita considerando i pixel “vicini”.

Distinguiamo 3 tipi di vicinanza:  $N_4$ ,  $N_D$  (diagonale),  $N_8$ .

In base al tipo di vicinanza che consideriamo, un pixel  $p$  posizionato in  $(x,y)$  ha vicini diversi:

- $N_4$ :  $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$
- $N_D$ :  $(x+1, y+1), (x-1, y+1), (x+1, y-1), (x-1, y-1)$
- $N_8$ :  $N_4 \cup N_D$



## • Adiacenza

La relazione di *adiacenza* è definita considerando i pixel “simili”.

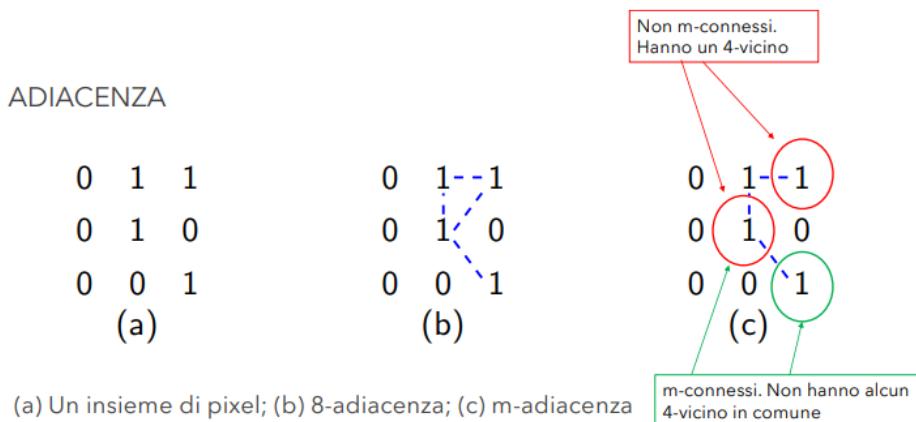
In questo contesto, l'intensità dei pixel è usata per definire la relazione di similarità: l'insieme  $V$  è composto dalle intensità che (arbitrariamente) devono essere considerate nella definizione di adiacenza.

Distinguiamo 3 tipi di adiacenza: 4-adiacenza, 8-adiacenza, m-adiacenza (mista).

I pixel  $p$  e  $q$ , che hanno valori di intensità in  $V$ , sono:

- 4-adiacenti se  $q \in N_4(p)$
- 8-adiacenti se  $q \in N_8(p)$
- M-adiacenti se:
  - $q \in N_4(p)$ , oppure
  - $q \in N_D(p)$  e nessun pixel in  $N_4(p) \cap N_4(q)$  ha valori in  $V$

L'm-adiacenza è usata per evitare l'ambiguità che deriva dall'uso della 8-adiacenza. Ad esempio, quando un pixel è legato a più pixel.



Il **percorso** (path) dal pixel  $p$  con coordinate  $(x_p, y_p)$  al pixel  $q$  con coordinate  $(x_q, y_q)$  è una sequenza di  $n+1$  pixel distinti che hanno coordinate:  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ . Dove:

- $(x_0, y_0) = (x_p, y_p)$  e  $(x_n, y_n) = (x_q, y_q)$
- $(x_{i-1}, y_{i-1})$  e  $(x_i, y_i)$  sono adiacenti ( $1 \leq i \leq n$ )

Se  $(x_0, y_0) = (x_n, y_n)$ , il percorso è chiuso.

la definizione del percorso dipende dalla definizione di adiacenza: 4-path, 8-path, o m-path.



Il concetto di percorso è importante per definire le *componenti connesse*.

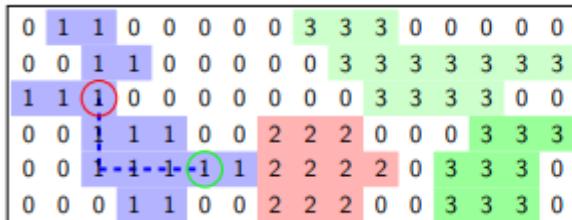
## • Componenti connesse

Sia  $S$  un sottoinsieme di pixel in un'immagine. Due pixel  $p$  e  $q$  sono detti *connessi in*  $S$  se esiste un percorso tra di essi formato interamente di pixel appartenenti ad  $S$ .

Per ciascun pixel  $p$  in  $S$ , l'insieme dei pixel che sono connessi ad esso è detto **componente connessa** di  $S$ .

Se si ha un'unica componente connessa, allora l'insieme  $S$  è detto **insieme connesso**.

Una **regione dell'immagine**  $R$  è un qualsiasi sottoinsieme di pixel connesso.



Usando la 4-adiacenza

Oss: ci sono due regioni per 3

La relazione di adiacenza può essere estesa alle **regioni** (*relazione tra regioni*).

- due regioni  $R_i$  e  $R_j$  sono **adiacenti** se  $R_i \cup R_j$  è un insieme connesso;
- le regioni non adiacenti sono dette **disgiunte**;
- le regioni adiacenti dipendono dalla relazione di vicinanza adottata.

L'unione di tutte le regioni di un'immagine è chiamata **foreground**.

Il complemento del foreground (cioè, l'insieme dei pixel che non appartengono ad alcuna regione) è chiamato **background**.

È pratico che il foreground sia circondato dal background (in evenienza, si può aggiungere ai quattro lati dell'immagine una riga).

Il **bordo** (o contorno) di una regione  $R$  è l'insieme di pixel di  $R$  adiacenti al complemento di  $R$ . Cioè, il bordo di una regione è composto di pixel che hanno almeno uno dei propri vicini nel background.

Il bordo di una regione è anche chiamato **bordo interno** per distinguerlo dal bordo esterno che è composto da pixel di background che sono adiacenti al bordo esterno.

In modo simile, il bordo interno può essere identificato come quei pixel di foreground che sono adiacenti al bordo esterno.

Il bordo esterno è sempre un percorso chiuso, mentre la regione e il bordo interno possono essere percorsi aperti.

La definizione di bordo dipende dall'adiacenza.

|   |   |   |
|---|---|---|
| $\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix} \left\{ R_i \right. \right)$ | $\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{matrix}$ | $\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{matrix}$ |
| $\left. \begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{matrix} \right\} R_j$         | $\begin{matrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$ | $\begin{matrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$ |

(d) due regioni (di 1) che sono adiacenti utilizzando l'8-adiacenza;  
(e) il punto cerchiato è parte del confine soltanto se si utilizza l'8-adiacenza tra la regione e il background; (f) il limite interno della regione con valore 1 non forma un path chiuso, mentre il limite esterno sì.

Il concetto di bordo non devo essere confuso con quello di *edge*, che invece è una sottosequenza del bordo.

## • Distanza

Dati i pixel  $p, q$  e  $z$  rispettivamente di coordinate  $(x, y)$ ,  $(s, t)$  e  $(v, w)$ ,  $D$  è una **funzione di distanza** se rispetta tali proprietà: *dist. positiva; commutativa; dist. triangolare.*

1.  $D(p, q) \geq 0$ ,  $(D(p, q) = 0 \text{ sse } p = q)$
2.  $D(p, q) = D(q, p)$
3.  $D(p, q) \leq D(p, z) + D(z, q)$

Possiamo considerare **3 tipi di distanze**:

- **distanza Euclidea  $D_e$** , norma ( $L_2$ ): identifichiamo un cerchio di raggio  $r$ .

$$D_e(p, q) = \left[ (x_p - x_q)^2 + (y_p - y_q)^2 \right]^{\frac{1}{2}}$$

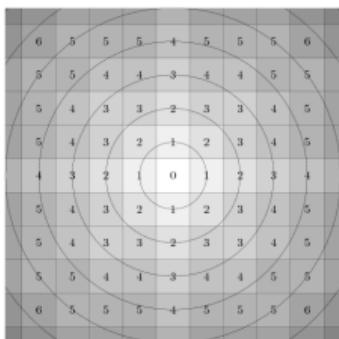
- **distanza  $D_4$** , detta anche Manhattan o city-block, norma ( $L_1$ ): identifichiamo un rombo.

$$D_4(p, q) = |x_p - x_q| + |y_p - y_q|$$

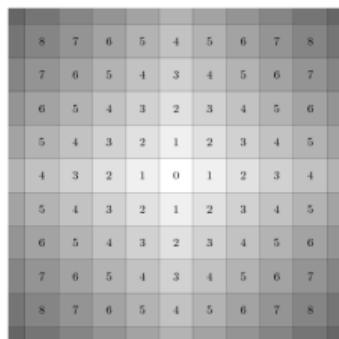
- **distanza  $D_8$** , detta anche scacchiera, norma ( $L_{\infty}$ ): identifichiamo un quadrato.

$$D_8(p, q) = \max(|x_p - x_q|, |y_p - y_q|)$$

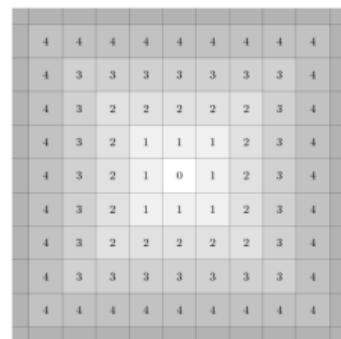
Esempi:



$D_e$



$D_4$



$D_8$

La distanza può essere valutata anche usando la **m-adiacenza**. In questo caso la distanza tra i due pixel dipenderà dal valore dei pixel lungo il path e dai valori dei vicini. Infatti, prima la distanza era definita solo in termine spaziali quando in realtà è importante anche il valore stesso del pixel.

La distanza  $D_m$  fra due punti è definita come l'*m-path* più breve tra i punti.

|         |         |         |
|---------|---------|---------|
| 0 1 1 1 | - 2 2 3 | - 3 4 5 |
| 1 1 0 0 | 1 1 - - | 1 2 - - |
| 1 0 0 0 | 0 - - - | 0 - - - |
| 0 1 0 0 | - 1 - - | - 1 - - |
| 0 0 1 1 | - - 2 3 | - - 2 3 |

(a)

(b)

(c)

■ (a) Distribuzione di pixel. (b) distanza  $D_8$  (solo per pixel a valore 1) (c) La distanza  $D_m$  corrispondente

## LEZ 7 – Fondamenti (2): Operazioni su immagini

Possiamo distinguere diverse operazioni sulle immagini: *matriciali e puntuali, lineari e non lineari, aritmetiche, sugli insiemi, logiche, spaziali.*

### Operazioni puntuali e matriciali

Le operazioni puntuali, su una o più immagini, vengono eseguite pixel per pixel. Le immagini possono essere trattate come matrici, tuttavia proprio per questo è importante specificare se stiamo effettuando delle operazioni puntuali o matriciali.

Si assume che le operazioni vengano applicate sempre in maniera puntuale.

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

Il prodotto puntuale di queste due immagini è

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

D'altra parte il prodotto matriciale è dato da

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

### Operazioni lineari e non lineari

Si consideri un operatore generale  $H$  che, data in input un'immagine  $f(x,y)$ , fornisca in output un'immagine  $g(x,y)$ :  $H[f(x,y)] = g(x,y)$

e si considerino due costanti arbitrarie  $a$  e  $b$ .  $H$  è detto *operatore lineare* se:

$$H[af_1(x,y) + bf_2(x,y)] = aH[f_1(x,y)] + bH[f_2(x,y)] = ag_1(x,y) + bg_2(x,y)$$

se un operatore non soddisfa tale equazione, allora è detto *non lineare*.

Ad esempio, la somma è lineare perché soddisfa l'equazione, invece l'operazione di massimo non è lineare ed è dimostrabile per assurdo.

## Operazioni aritmetiche (op. tra pixel)

Le operazioni aritmetiche fra immagini sono operazioni puntuale che indicano operazioni aritmetiche eseguite tra coppie di pixel corrispondenti.

Date due immagini  $f(x,y)$  e  $g(x,y)$ , di dimensioni  $M \times N$ , si ha:

$$s(x, y) = f(x, y) + g(x, y)$$

$$d(x, y) = f(x, y) - g(x, y)$$

$$p(x, y) = f(x, y) \times g(x, y)$$

$$v(x, y) = f(x, y) \div g(x, y)$$

Ad esempio,

la somma è usata in astronomia per ridurre il rumore delle immagini. In particolare, la media della somma delle immagini ci fornisce un'immagine con meno rumore.

Maggiore è il numero di immagini, meno sarà il rumore.

La sottrazione è usata per il confronto, per verificare di quanto siano variati i pixel fra due immagini. Maggiore è l'intensità di un pixel, più sarà variato; se il pixel è nero, non è variato per nulla.

## Operazioni insiemistiche e logiche

Per ambo i casi si presume che le immagini siano binarie (bianco e nero).

Per le **operazioni insiemistiche**, banalmente, verifichiamo quale pixel appartiene a quale insieme, e su queste possiamo effettuare operazioni quali: l'unione, l'intersezione, ecc...

Per le **operazioni logiche** vengono utilizzati gli operatori logici (NOT, AND, OR...). Indichiamo con il termine *foreground* i pixel con valore 1, e con *background* i pixel con valore 0.

## Operazioni spaziali

Le operazioni spaziali sono eseguite direttamente sui pixel di una data immagine. Le classifichiamo in tre categorie: (1) operazioni a singolo pixel; (2) operazioni di vicinanza; (3) trasformazioni spaziali geometriche.

### 1. Operazioni a singolo pixel

Con le operazioni a singolo pixel si altera l'intensità dei singoli pixel di un'immagine usando una funzione di trasformazione, della forma:

$$s = T(z)$$

dove  $z$  è l'intensità di un pixel nell'immagine originale ed  $s$  è l'intensità del corrispondente pixel nell'immagine elaborata.

Ad esempio, supponiamo di voler effettuare una trasformazione negativa.

Teniamo conto di avere  $2^k$  livelli, dunque abbiamo un intervallo  $[0, L-1]$ . Consiste nel prendere ogni pixel e trasformarlo come:  $s = (L-1) - z$  es.  $(2,4) = (L-1) - (2,4)$ .

### 2. Operazioni di vicinanza

Parliamo di *operazioni locali* quando le elaborazioni coinvolgono i valori dei pixel in un piccolo intorno di una posizione  $(x,y)$  di interesse. Tali operazioni fanno uso di *maschere* (filtri, finestre o template).

Il risultato di un'operazione locale in  $(x,y)$  è funzione del suo livello di grigio e del livello di grigio dei pixel del suo contorno.

Ad esempio, consideriamo una tabelle  $3 \times 3$  di valori  $z$ .

Il valore del pixel centrale si ottiene:

1. "centrando" la maschera su di esso,
2. moltiplicando ogni pixel dell'intorno per il corrispondente coefficiente (o peso  $w$ ),
3. sommando i risultati.

Dunque avremo:

$$z = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i$$

|       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|
| $z_1$ | $z_2$ | $z_3$ | $w_1$ | $w_2$ | $w_3$ |
| $z_4$ | $z_5$ | $z_6$ | $w_4$ | $w_5$ | $w_6$ |
| $z_7$ | $z_8$ | $z_9$ | $w_7$ | $w_8$ | $w_9$ |

L'effetto dell'elaborazione dipende dai coefficienti della maschera. Ad esempio se volessimo effettuare una sfocatura (utile per ridurre il rumore) dovremo sostituire al pixel originale il valore medio dei livelli di grigio nell'intorno.

Per una maschera  $3 \times 3$  si può porre  $w_i = 1/9$ , per  $i = 1:9$ . Ottenendo:  $\bar{z} = \frac{1}{9} \sum_{i=1}^9 z_i$

Le operazioni locali sono tra le più adottate, tuttavia applicare un filtro a tutti i pixel di un'immagine è un'operazione computazionalmente costosa, soprattutto se consideriamo maschere di maggiori dimensioni. Oltre un certo punto, si preferisce mettere questa tecnica da parte in favore di altre.

### 3. Trasformazioni spaziali geometriche

Le trasformazioni spaziali geometriche sono usate per modificare la disposizione dei pixel in un'immagine. Consistono in due operazioni base:

- trasformazione spaziale delle coordinate;
- interpolazione che assegna i valori di intensità ai pixel trasformati spazialmente.

Infatti, dopo una trasformazione potremmo non avere più una corrispondenza 1 a 1, dunque deve seguire un'interpolazione.

La trasformazione delle coordinate può essere espressa come:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

con  $(x,y)$  coordinate del pixel nell'immagine originale e  $(x',y')$  le coordinate dei pixel corrispondenti nell'immagine trasformata.

*Ad esempio,*

la trasformazione  $(x',y') = (x/2, y/2)$  riduce l'immagine originale della metà in ambo le direzioni.

Le **trasformazioni affini** (così dette in quanto preservano rette, punti e piano) sono:

- scaling;
- traslazione;
- rotazione.

Tutte si rappresentano con l'equazione (2), ovvero come una matrice, eccetto la traslazione che richiede l'aggiunta di un vettore 2D costante al membro destro dell'equazione.

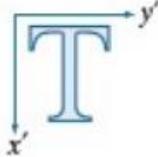
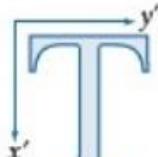
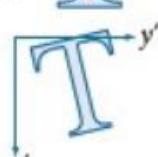
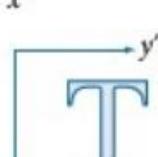
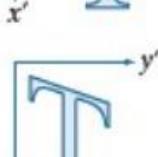
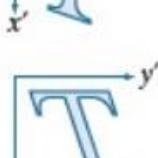
Tutte le trasformazioni affini possono essere espresse mediante una singola matrice generale 3x3, della forma:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3)$$

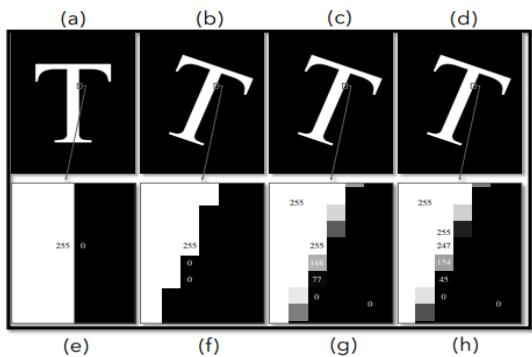
Si tratta una rappresentazione con *coordinate omogenee*. È utile in quanto ci permette di concatenare una serie di trasformazioni, o anche per rappresentare i punti all'infinito.

Quando ruotiamo un'immagine, può capitare che esca fuori dai contorni.  
Possiamo adottare due soluzioni: tagliare l'immagine; ingrandire l'immagine.  
I pixel in background saranno neri.

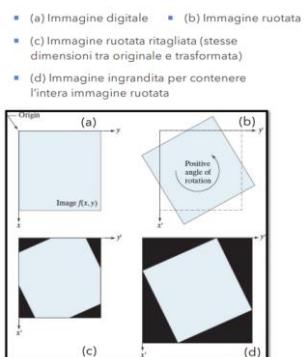
### Trasformazioni affini in coordinate omogenee

| Transformation Name   | Affine Matrix, A   | Coordinate Equations   | Example   |
|---|--|--|---|
| Identity  | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  | $x' = x$<br>$y' = y$   |    |
| Scaling/Reflection<br>(For reflection, set one scaling factor to -1 and the other to 0) | $\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$                                      | $x' = c_x x$<br>$y' = c_y y$   |    |
| Rotation (about the origin)   | $\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x' = x \cos \theta - y \sin \theta$<br>$y' = x \sin \theta + y \cos \theta$ |    |
| Translation   | $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$                                      | $x' = x + t_x$<br>$y' = y + t_y$   |   |
| Shear (vertical)  | $\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  | $x' = x + s_v y$<br>$y' = y$   |  |
| Shear (horizontal)  | $\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  | $x' = x$<br>$y' = s_h x + y$   |  |

Esempio rotazione e Interpolazione



Esempio dimensioni immagini trasformate



- (a) Immagine digitale
- (b) Immagine ruotata
- (c) Immagine ruotata ritagliata (stesse dimensioni tra originale e trasformata)
- (d) Immagine ingrandita per contenere l'intera immagine rotata

## LEZ 8 – Trasformazioni di intensità

Il **dominio spaziale** di un'immagine è il piano che contiene i pixel dell'immagine.

Le **tecniche del dominio spaziale** operano direttamente sui pixel di un'immagine, a differenza di quanto avviene, ad esempio, nel dominio della frequenza.

Formalmente, per queste tecniche possiamo scrivere che  $\mathbf{g(x,y) = T[f(x,y)]}$

Generalmente, le tecniche del dominio spaziale sono meno gravose da un punto di vista computazionale.

Potremmo fare queste operazioni, ad esempio, per poter rendere le immagini più nitide (ie aumentando il contrasto).

L'operatore  $T$  è di solito definito su un'opportuna area di vicinanza di  $(x,y)$ .

Siccome la vicinanza può fuoriuscire dall'immagine (si immagini di considerare un rettangolo 3x3 di coordinate 0,0) devono essere usati alcuni criteri di estensione, cioè si deve effettuare un padding.

Se il raggio di vicinanza è 0, la trasformazione coinvolge solo il pixel considerato e dipende (solo) dalla sua intensità.

In questo caso  $g$  dipende solo dal valore di  $f$  in un singolo punto  $(x,y)$  e  $T$  nell'equazione diventa una *funzione di trasformazione dell'intensità* (aka livello di grigio/mapping), la cui formula è:

$$s = T(r)$$

dove  $s$  ed  $r$  sono variabili che denotano rispettivamente l'*intensità di g* e *di f* in ogni punto  $(x,y)$ .

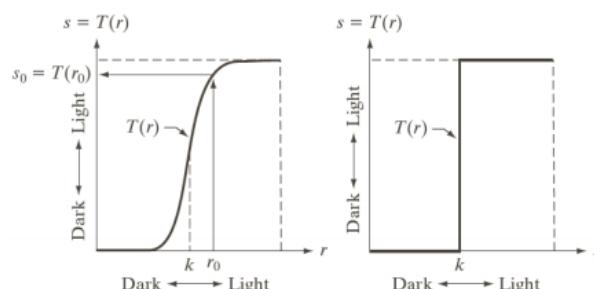
Le tecniche di trasformazione di intensità sono anche chiamate **elaborazioni puntuali**, contrapposte a quelle locali. Tali tecniche sono semplici da implementare.

I valori di  $T$  vengono immagazzinati in array monodimensionali e le mappataure da  $r$  ad  $s$  vengono implementate attraverso delle tabelle. Per un ambiente a 8 bit, la tabella contenente i valori di  $T$  avrà 256 inglesi.

Non c'è nessuna regola generale di ottimalità: la trasformazione da usare dipende dall'applicazione e dall'utente.

Es.

Binarizzazione di un'immagine: i pixel sotto una certa soglia sono tutti messi a nero, sopra una certa soglia sono tutti messi a bianco.

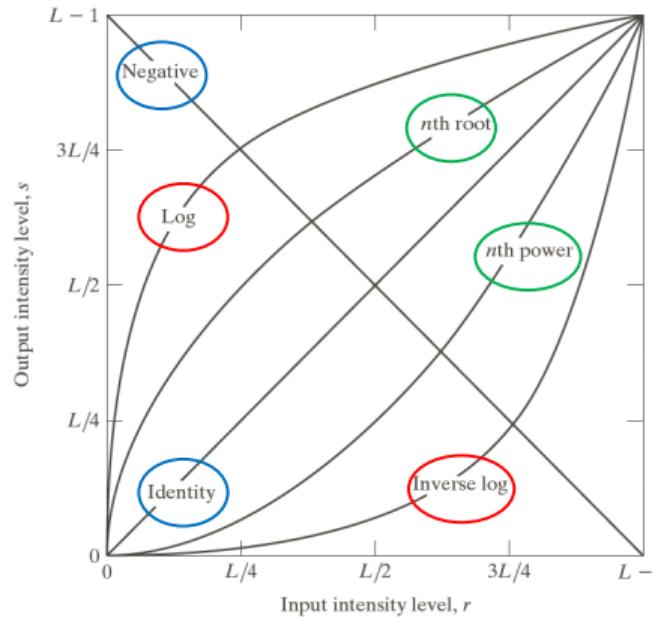


## Funzioni per la trasformazione di intensità

Possiamo distinguere **3 tipologie di base di funzioni** utilizzate frequentemente per il miglioramento dell'immagine:

- *lineare*: trasformazioni negative e identità (caso banale, ie i. input = i. output);
- *logaritmica*: trasformazioni log e log inverso;
- *potenza*: trasformazioni delle potenza n-esima e della radice n-esima.

$$s = T(r)$$



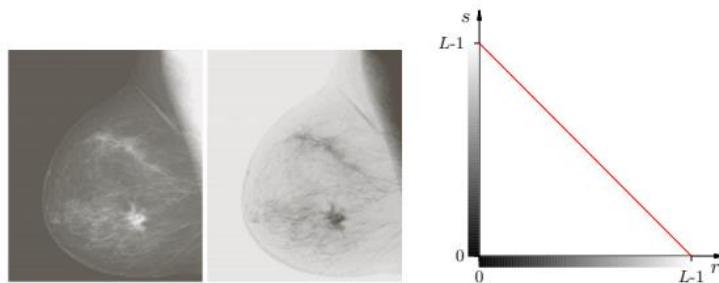
## Lineare (Negatività)

Talvolta, i dettagli sono più evidenziabili quando l'intensità dei pixel è *invertita*. Ad esempio, quando i dettagli sono bianco o grigio chiaro e il background è scuro e copre gran parte dell'immagine. È utile per scopi diagnostici.

Il negativo di un'immagine, con livelli di intensità nella gamma [0, L-1], si ottiene utilizzando la **trasformazione negativa** data dall'espressione:

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

es.



## Logaritmica

Le trasformazioni logaritmiche sono utili per rappresentare funzioni bidimensionali che sono definite su ampi intervalli ed hanno picchi grandi e piccoli.

Ad esempio, si immagini di avere un'immagine prevalentemente scura in cui però ho pochi pixel estremamente chiari. Attraverso una trasformazione logaritmica, posso rendere i pixel scuri più visibili.

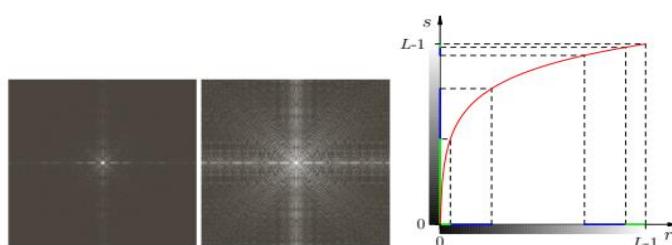
Infatti, queste trasformazioni permettono di associare ad una stretta gamma di valori a bassa intensità una gamma più ampia di valori di output.

La trasformazione logaritmica inversa procede invece al contrario.

La forma generale della **trasformazione logaritmica** è:

$$s = c \log(1 + r), c = \frac{L-1}{\log L}$$

es.



## Potenza (aka gamma)

Le trasformazioni potenza (aka gamma) sono usate per correggere l'output del dispositivo di visualizzazione.

Sono utili inoltre per la correzione del contrasto (o miglioramento).

È detta "gamma" in quanto deriva dalle procedure di correzione che si usavano per i monitor CRT (in quanto mostrano un'immagine più scura di quello che dovrebbe essere, e dunque cerco di schiarirla).

La forma generale della **trasformazione potenza** è:

$$s = cr^\gamma \quad \text{dove } c \text{ e } \gamma > 0.$$

(talvolta è riportata anche come  $s = c(r+\varepsilon)$ , per tener conto dello scarto (offset)).

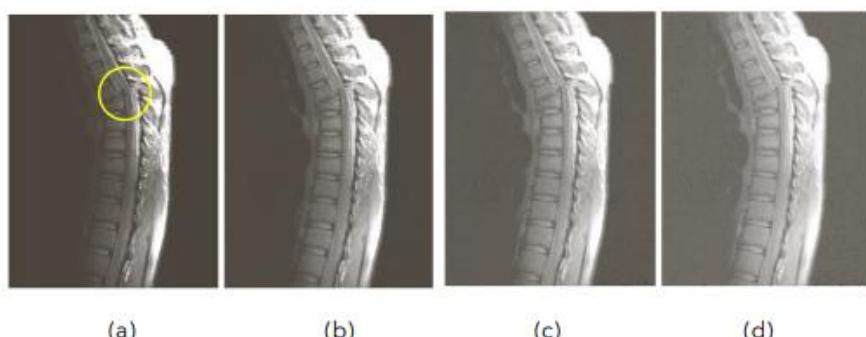
La costante  $\gamma$  è quella che ha maggior peso nell'equazione. In particolar modo:

- $\gamma = 1$ , identità;
- $\gamma < 1$ , schiarimento;
- $\gamma > 1$ , oscuramento.

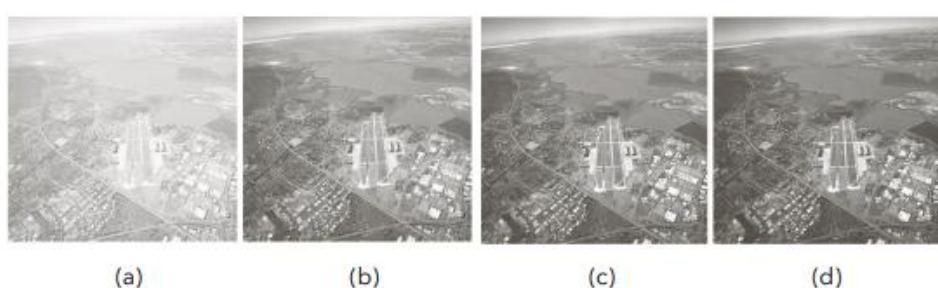
Anche  $c$  influenza, ma per comodità possiamo assumere che sia 1.

Ci permette di "giochicchiare" un po' di più, soprattutto quando vediamo che  $\gamma$  va bene e vogliamo lavorare sui dettagli.

es.



(a) Immagine originale. Immagine gamma trasformata con (b)  $c = 1, \gamma = 0.6$ , (c)  $c = 1, \gamma = 0.4$ , (d)  $c = 1, \gamma = 0.3$



(a) Immagine originale. Immagini gamma trasformate con (b)  $c = 1, \gamma = 3.0$ , (c)  $c = 1, \gamma = 4.0$ , (d)  $c = 1, \gamma = 5.0$

## Funzioni di trasformazione lineare a tratti

Le funzioni di trasformazione lineare a tratti rappresentano un approccio alternativo alle trasformazioni viste prime.

Possiamo distinguere le trasformazioni di: *stretching del contrasto*, *selezione del livello di intensità*, e di *selezione mediante piani di bit (bit-plane)*.

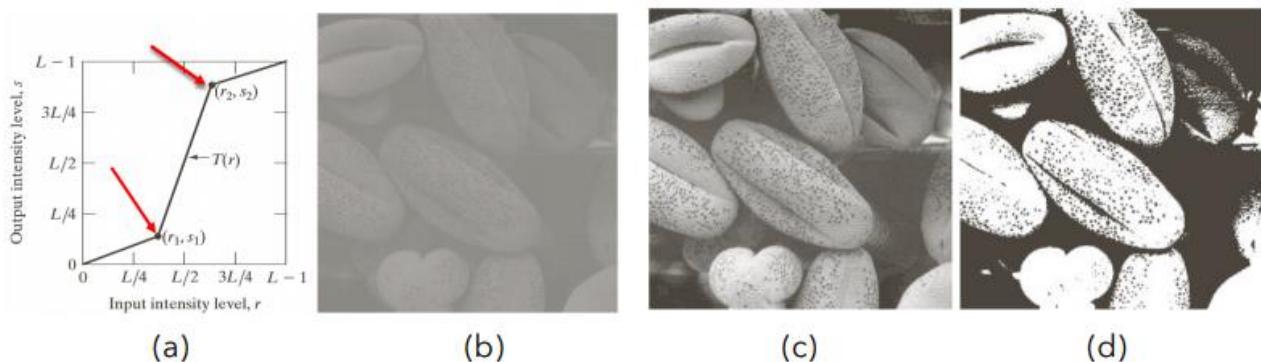
### Espansione del contrasto (stretching)

Lo stretching del contrasto è un processo che amplia la gamma dei livelli di intensità di un'immagine in modo tale che venga utilizzata l'intera gamma di valori nel mezzo di registrazione o nello strumento di visualizzazione.

Le posizioni dei punti  $(r_1, s_1)$  e  $(r_2, s_2)$  determinano la forma della *funzione di trasformazione*. L'idea è selezionare un certo intervallo da 0 a  $L-1$  su cui andremo ad ampliare la gamma dei livelli di intensità così da poter ottenere un miglioramento dell'immagine (es, aumentando il contrasto).

Ad esempio, supponiamo di avere un'immagine a basso contrasto (b). Di questa immagine selezioniamo un intervallo da evidenziare ponendo  $(r_1, s_1) = (r_{\min}, 0)$  e  $(r_2, s_2) = (r_{\max}, L-1)$ : potremo notare un miglioramento del contrasto (c).

Se  $r_1 = s_1 = 0$  ed  $s_2 = L-1$ . La trasformazione diventa una *funzione soglia* che crea un'immagine binaria (d).

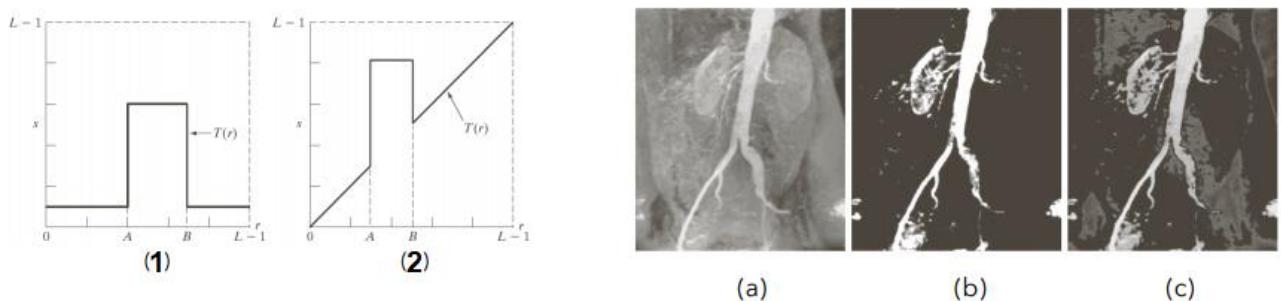


## Trasformazioni di selezione (slicing) del livello di intensità

Le trasformazioni di slicing del livello di intensità ha lo scopo di evidenziare alcune caratteristiche particolari di un'immagine, prendendo in considerazione una specifica gamma di intensità.

Possiamo distinguere due *strategie*:

- impostare tutte le intensità che non sono in  $[A, B]$  ad un valore basso. Così facendo, schiariremo le intensità in  $[A, B]$  e andremo a scurire tutto il resto (1,b).
- preservare le intensità che non sono in  $[A, B]$ . Così facendo, schiariremo le intensità in  $[A, B]$  ma lasciando inalterati tutti gli altri livelli di intensità (2,c).



## Trasformazione bit-plane

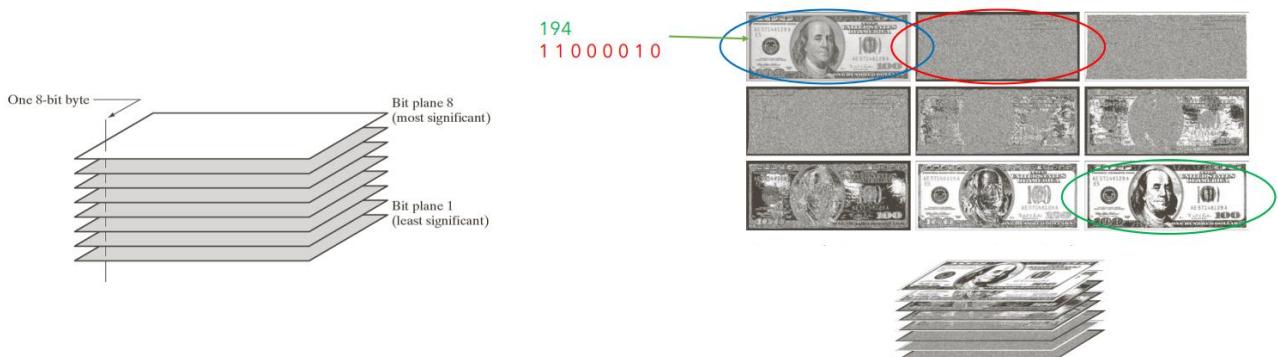
Nella trasformazione bit-plane l'immagine viene vista come composta da strati di bit (invece che da una matrice di interi).

Dunque, un'immagine a 8 bit avrà 8 strati: il piano bit 8 sarà il più significativo, mentre il piano bit 1 sarà il meno significativo.

Ogni strato contribuisce all'aspetto finale dell'immagine, ma il più dell'informazione è negli strati più alti. Quelli più bassi riguardano le piccole variazioni di intensità.

(I pixel del piano n-esimo vengono moltiplicati con la costante  $2^{n-1}$ ).

Ovviamente maggiore è il numero di strati, maggiore è il livello di dettaglio. Tuttavia, già con i soli 3/4 strati più significativi possiamo ottenere un'immagine molto buona, e dunque risulta essere un ottimo metodo per le compressioni.



## Elaborazione di istogrammi

L'istogramma fornisce uno strumento (visuale) intuitivo per valutare alcune proprietà statistiche dell'immagine, e sono utili nel caso volessimo rappresentare quanti pixel ci sono di uno specifico valore di intensità.

Nello specifico, l'istogramma di un'immagine a L-valori è una **funzione discreta**:

$$h(k) = n_k, \quad k \in [0, \dots, L-1]$$

dove  $n_k$  è il numero di pixel con intensità  $k$ .

Spesso è preferibile considerare l'istogramma normalizzato rispetto al numero di pixel,  $M \times N$ :

$$p(k) = \frac{n_k}{MN}$$

la funzione  $p(k)$  è una probabilità di occorrenza di  $k$ .

Ovviamente, la somma di ogni  $p(k)$  è uguale ad 1.

Le trasformazioni basate su istogramma sono numerose (miglioramento, compressione, segmentazione), e possono essere implementate facilmente.

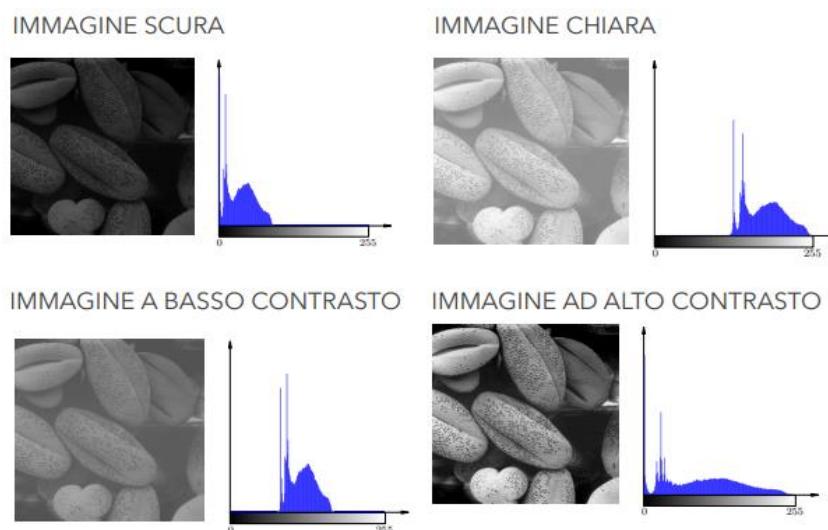
Più è uniforme la distribuzione dei valori sull'istogramma, migliore sarà l'immagine.

La situazione che vogliamo evitare è che i valori si concentrino su una zona specifica dell'istogramma (creando una "montagnola").

Infatti, considerando una range da 0 a 255:

- avremo un'immagine scura quando i valori sono localizzati sulla sinistra dell'istogramma, mentre chiara se sono localizzati sulla destra;
- avremo un'immagine ad alto contrasto se i valori sono distribuiti uniformemente, mentre a basso contrasto se sono concentrati al centro.

Il nostro obiettivo ovviamente è avere un istogramma con una distribuzione più uniforme possibile, con pochi picchi. Infatti l'immagine tenderà ad avere un elevato range dinamico e dunque i dettagli saranno percepiti più facilmente.



Per studiare le *trasformazioni di istogramma* dobbiamo introdurre le **trasformazioni monotone** (continue) su  $[0, L-1]$ .

(Ricordiamo che un f. monotona può essere crescente, non crescente, decrescente, e non decrescente. Ad es, una f. monotona crescente è una funzione che cresce sempre).

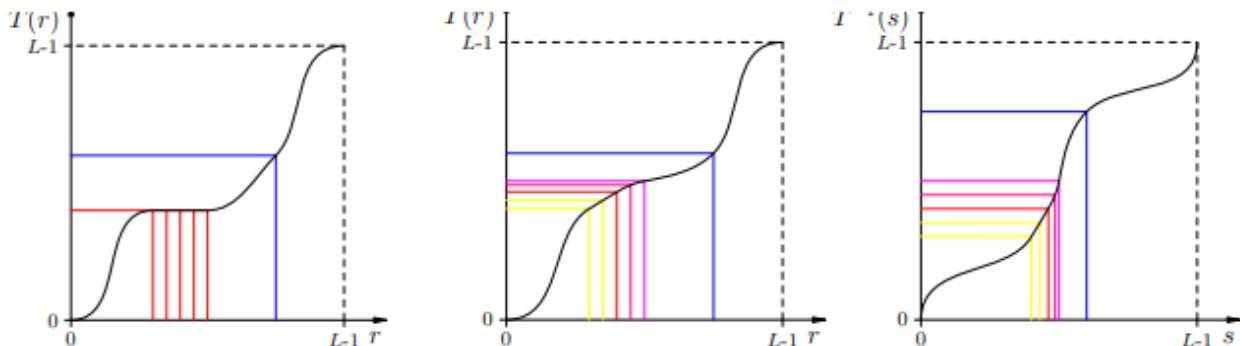
**Supponiamo che:**

1.  $s = T(r)$ ,  $0 \leq r \leq L-1$
2.  $T(r_2) \geq T(r_1)$ ,  $r_2 > r_1$
3.  $0 \leq T(r) \leq L-1$ ,  $0 \leq r \leq L-1$

(cioè che: (1)  $T(r)$  produce un output di intensità  $s$  per ogni pixel dell'immagine di input di intensità  $r$ , dove  $r$  è compreso fra 0 e  $L-1$  (nero e bianco). (2)  $T(r)$  è una funzione monotona non decrescente, infatti una f. è tale se  $T(r_2) \geq T(r_1)$  oer  $r_2 > r_1$ . (3)  $T(r)$  ed  $r$  sono compresi nell'intervallo  $[0, L-1]$  ).

**Diremo che:** se  $T$  è strettamente monotona crescente, esiste  $T^{-1}$ :

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



Le intensità (continue) possono essere considerate come variabili casuali in  $[0, L-1]$ .

- Se  $s = T(r)$  e  $T(r)$  è continua e differenziabile:

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

- In particolare, è interessante la seguente trasformazione:

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

- Allora:

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

- Quindi:

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| = p_r(r) \left| \frac{1}{(L-1)p_r(r)} \right| = \frac{1}{L-1}, \quad 0 \leq s \leq L-1$$

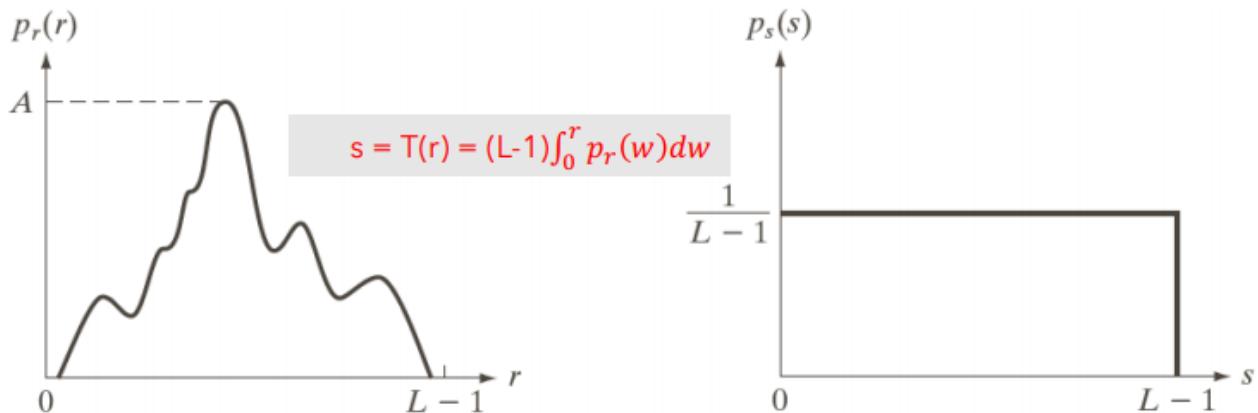
- **s è uniforme**, indipendentemente da  $p_r$

In pratica, se applico la **trasformazione**:  $s = T(r) = (L-1) \int_0^r p_r(w) dw$ , riuscirò ad ottenere un'immagine **equalizzata**.

Applicando la trasformazione di intensità si arriva a una variabile casuale,  $s$ , caratterizzata da una funzione uniforme.

$P_s(s)$  è *sempre* uniforme, *indipendentemente* dalla forma di  $p_r(r)$ .

(nb: “p” indica la probabilità. Dunque tutti i pixel hanno la stessa probabilità. Inoltre non ci interessa la dimostrazione, solo la formula).



Ricordiamo che gli integrali nel discreto diventano sommatorie.

Sapendo che  $r_k$  è il livello di intensità in  $[0, L-1]$ , dunque  $p_r(r_k) = \frac{n_k}{MN}$

un'immagine può essere equalizzata assegnando l'intensità  $s_k$  a quei pixel che hanno intensità  $r_k$  tramite:

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

Questa trasformazione è detta **equalizzazione di istogramma** e soddisfa le stesse condizioni del caso continuo.

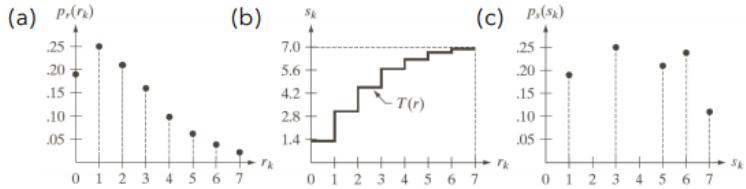
## Esempio

Supponiamo

- Immagine di 64x64 (4096) pixel con 3 bit per l'intensità, ovvero livelli da 0 a 7
- L'immagine abbia una distribuzione di intensità del tipo riportata in tabella

| $r_k$     | $n_k$ | $p_r(r_k)$ | $T(r_k)$ | $s_k$ | $p_s(s_k)$ |
|-----------|-------|------------|----------|-------|------------|
| $r_0 = 0$ | 790   | 0.19       | 1.33     | 1     | 0.19       |
| $r_1 = 1$ | 1023  | 0.25       | 3.08     | 3     | 0.25       |
| $r_2 = 2$ | 850   | 0.21       | 4.55     | 5     | 0.21       |
| $r_3 = 3$ | 656   | 0.16       | 5.67     | 6     | 0.24       |
| $r_4 = 4$ | 329   | 0.08       | 6.23     | 6     |            |
| $r_5 = 5$ | 245   | 0.06       | 6.65     | 7     |            |
| $r_6 = 6$ | 122   | 0.03       | 6.86     | 7     | 0.11       |
| $r_7 = 7$ | 81    | 0.02       | 7.00     | 7     |            |

$$s_k = \frac{L-1}{MN} \sum_{j=0}^k n_j, \quad k = 0, \dots, 7$$



DIST

I valori della funzione di trasformazione sono ottenuti dall'Equazione di  $s_k$ .

A dire il vero, il valore calcolato è in realtà  $T(r_k)$ , che però noi dobbiamo approssimare perché lavoriamo con valori discreti, e da qui esce  $s_k$ .

$n_k$  rappresenta il numero di pixel che hanno intensità  $r_k$ , mentre  $p(r_k)$  rappresenta, in modo simile, la "probabilità" che si presenti quel pixel.

Ovviamente la somma di tutti gli  $n_k$  è  $M \times N$ , mentre la somma di tutti i  $p(r_k)$  è 1.

Ad esempio,

$$s_0 = T(r_0) = 7 \sum_{j=0}^0 p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7 \sum_{j=0}^1 p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

...

Alla fine, noteremo che avremo 5 livelli distinti.

## Specificità dell'istogramma

Talvolta, è utile avere la possibilità di ottenere un'immagine con una *specifica distribuzione* (matching dell'istogramma).

Il problema può essere formalizzato come:

Data un'immagine di input, i cui pixel sono distribuiti con una densità di probabilità  $p_r$ ;  
Data la distribuzione di intensità desiderata,  $p_z$ ;  
Trovare la trasformazione  $F$ , tale che  $z = F(r)$ .

Sia  $s$  una variabile casuale tale che:

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

Definiamo una funzione  $G$  su  $z$  che soddisfa:

$$G(z) = (L-1) \int_0^z p_z(t) dt = s$$

Quindi,  $G(z) = s = T(r)$ .

Il mapping cercato  $F$ , tale che  $z = F(r)$  può essere ottenuto come:

$$z = G^{-1}(T(r))$$

Si applica quest'ultima trasformazione per ottenere i pixel dell'immagine di output.

Ovviamente, noi lavoriamo nel discreto, dunque gli integrali diventano sommatorie:

$$s_k = \frac{L-1}{MN} \sum_{j=0}^k n_j, \quad k = 0, \dots, L-1, \quad G(z_q) = (L-1) \sum_{i=0}^q p_z(z_i) = s_k$$

$$z_q = G^{-1}(s_k)$$

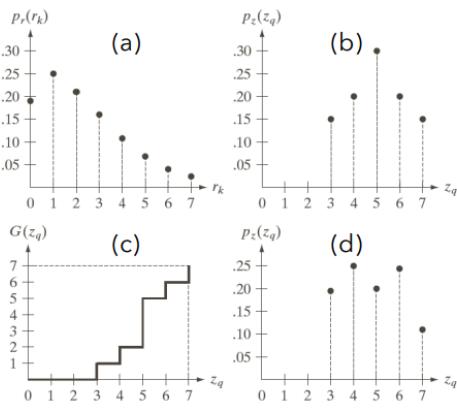
La **procedura di matching** dell'istogramma può essere realizzata:

1. Ottenere  $p_r$  dall'immagine di input
2. Ottenere il mapping  $T$  usando la relazione di equalizzazione  $s_k = \frac{L-1}{MN} \sum_{j=0}^k n_j, \quad k = 0, \dots, L-1$  (arrotondando  $s_k$  a numeri in  $[0, \dots, L-1]$ )
3. Ottenere il mapping  $G$  dalla  $p_z$  specificata  $G(z_q) = (L-1) \sum_{i=0}^q p_z(z_i)$  (arrotondando  $z_q$  a numeri in  $[0, \dots, L-1]$ )
4. Per ogni  $s_k, k=0,1, \dots, L-1$ , usare i valori di  $G$  in tabella per trovare i  $z_q$  corrispondenti, in modo che  $G(z_q)$  è più vicino a  $s_k$ 
  1. Se ci sono più valori  $z_q$  per cui  $G(z_q)$  è il più vicino a  $s_k$ , scegliere il più piccolo
5. Applicare la trasformazione  $F$  all'immagine originale tramite in mapping trovato al passo 4

Perché sia invertibile,  $G$  deve essere strettamente monotona. Tuttavia, nei casi pratici, tale proprietà è raramente soddisfatta (anche se a noi non interessa se  $G$  sia monotona, in quanto lavoriamo nel discreto e ci basta un'approssimazione).

Esempio:

- Supponiamo di avere un'immagine  $64 \times 64$  il cui istogramma è riportato in (a)
- Vogliamo trasformare l'istogramma in modo che abbia i valori specificati in (b)



$$G(z_q) = (L-1) \sum_{i=0}^q p_z(z_i)$$

| $r_k$     | $n_k$ | $p_r(r_k)$ | $T(r_k)$ | $s_k$ | $p_s(s_k)$ |
|-----------|-------|------------|----------|-------|------------|
| $r_0 = 0$ | 790   | 0.19       | 1.33     | 1     | 0.19       |
| $r_1 = 1$ | 1023  | 0.25       | 3.08     | 3     | 0.25       |
| $r_2 = 2$ | 850   | 0.21       | 4.55     | 5     | 0.21       |
| $r_3 = 3$ | 656   | 0.16       | 5.67     | 6     | 0.24       |
| $r_4 = 4$ | 329   | 0.08       | 6.23     | 6     |            |
| $r_5 = 5$ | 245   | 0.06       | 6.65     | 7     |            |
| $r_6 = 6$ | 122   | 0.03       | 6.86     | 7     | 0.11       |
| $r_7 = 7$ | 81    | 0.02       | 7.00     | 7     |            |

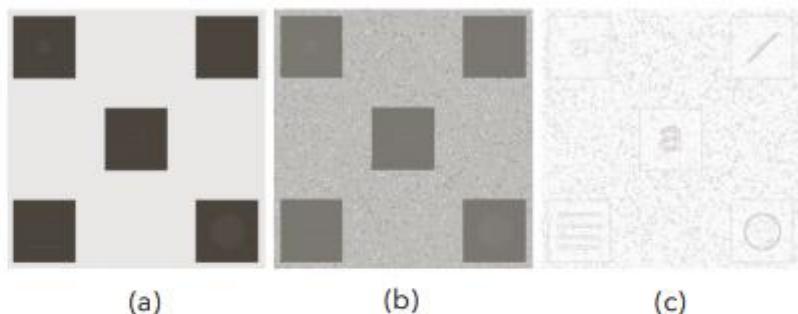
| $z_q$     | Specified<br>$p_z(z_q)$ | Actual<br>$p_z(z_k)$ | $z_q$     | $G(z_q)$ | $s_k \rightarrow z_q$ |
|-----------|-------------------------|----------------------|-----------|----------|-----------------------|
| $z_0 = 0$ | 0.00                    | 0.00                 | $z_0 = 0$ | 0        | 1 → 3                 |
| $z_1 = 1$ | 0.00                    | 0.00                 | $z_1 = 1$ | 0        | 3 → 4                 |
| $z_2 = 2$ | 0.00                    | 0.00                 | $z_2 = 2$ | 0        | 5 → 5                 |
| $z_3 = 3$ | 0.15                    | 0.19                 | $z_3 = 3$ | 1        | 6 → 6                 |
| $z_4 = 4$ | 0.20                    | 0.25                 | $z_4 = 4$ | 2        | 7 → 7                 |
| $z_5 = 5$ | 0.30                    | 0.21                 | $z_5 = 5$ | 5        |                       |
| $z_6 = 6$ | 0.20                    | 0.24                 | $z_6 = 6$ | 6        |                       |
| $z_7 = 7$ | 0.15                    | 0.11                 | $z_7 = 7$ | 7        |                       |

## Elaborazione locale dell'istogramma

L'equalizzazione dell'istogramma è un approccio globale.

L'equalizzazione locale dell'istogramma è realizzata selezionando, per ogni pixel, un'area di vicinanza adeguata su cui è calcolata l'equalizzazione dell'istogramma (o matching). Si tratta di una procedura più pesante computazionalmente.

Esempio:



(a) immagine originale

(b) immagine equalizzata

(c) immagine equalizzata localmente (vicinanza 3x3)

## Statistiche da istogramma

Possono essere facilmente elaborati alcuni indici statistici dall'istogramma:

- Media (average):  $m = \sum_{i=0}^{L-1} r_i p(r_i)$
- Varianza:  $\sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$
- Deviazione standard:  $\sigma = \sqrt{\sigma^2}$
- Momento n-esimo:  $\mu_n = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i)$

Possono essere calcolati indici statistici locali limitando l'istogramma ad una data area di vicinanza  $S_{xy}$ :

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i p_{S_{xy}}(r_i)$$

$$\sigma_{S_{xy}}^2 = \sum_{i=0}^{L-1} (r_i - m_{S_{xy}})^2 p_{S_{xy}}(r_i)$$

Esempio:

- Solo le regioni scure devono essere migliorate  
 $m_{S_{xy}} \leq k_0 m_G$
- Le regioni uniformi devono essere preservate  
 $\sigma_{S_{xy}} \geq k_1 \sigma_G$
- Le regioni a basso contrasto devono essere migliorate  
 $\sigma_{S_{xy}} \leq k_2 \sigma_G$

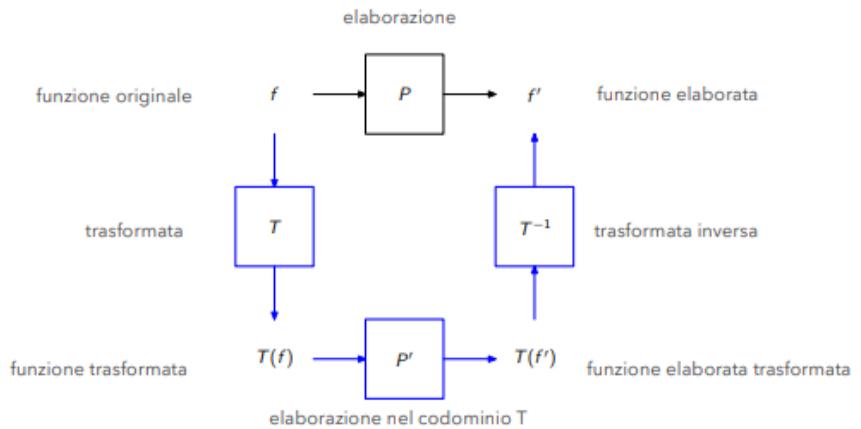
$$g(x, y) = \begin{cases} E \cdot f(x, y) & \text{if } m_{S_{xy}} \leq k_0 m_G \\ & \text{AND } k_1 \sigma_G \leq \sigma_{S_{xy}} \leq k_2 \sigma_G \\ f(x, y) & \text{otherwise} \end{cases}$$

$$E = 4, k_0 = 0.4, k_1 = 0.02, k_2 = 0.4.$$

## LEZ 9 – Trasformata di Fourier (1D)

Talvolta può risultare più semplice lavorare su una classe di funzioni se queste sono espresse in un altro dominio, come ad esempio il dominio delle *frequenze* invece del dominio del tempo (o spazio). Dunque queste funzioni verranno trasformate su un altro dominio e, una volta elaborate, verranno retro-trasformate al dominio originale.

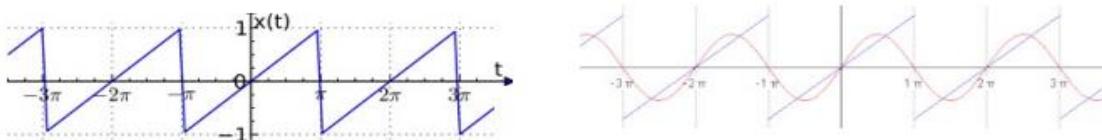
La funzione originale  $f$  può essere direttamente elaborata (ottenendo  $f'$ ), oppure possiamo trasformarla (ottenendo  $T(f)$ ), effettuare l'elaborazione sulla funzione trasformata (ottenendo  $T(f')$ ) e infine facendo la trasformata inversa per ottenere  $f'$ .



Di queste trasformate noi considereremo quella di **Fourier**.

L'**idea principale** di Fourier è che: qualsiasi funzione periodica può essere espressa come somma di funzioni seno e coseno di frequenze diverse, ciascuna moltiplicata per un apposito coefficiente. In particolar modo, più termini aggiungiamo alla serie, più questa approssimazione diventa precisa. Tale somma prende il nome di **serie di Fourier**.

Fourier dimostrò inoltre che anche le funzioni non periodiche possono essere esprimesse come una somma di funzioni seno e coseno, però in questo caso non parliamo più di **serie** ma di **trasformata di Fourier**.



## Numeri complessi

Nell'ambito dell'analisi di Fourier entrano in gioco i numeri complessi.

Sia  $j$  il numero immaginario  $j = \sqrt{-1}$ , un numero complesso  $C$  è definito come:  $C = R + jI$  , dove  $R$  denota la *parte reale* ed  $I$  denota la *parte immaginaria*.

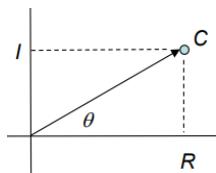
Il *coniugato* di un numero complesso  $C^*$  è definito come:  $C^* = R - jI$  .

I numeri complessi possono essere visualizzati geometricamente come punto in un piano (chiamato *piano complesso*) di cui l'asse delle ascisse è l'*asse reale* (valori di  $R$ ) mentre l'asse delle ordinate è l'*asse immaginario* (valori di  $I$ ).

Può essere utile rappresentare tali numeri in *coordinate polari*:  $C = |C|(\cos \theta + j \sin \theta)$  , o anche  $C = |C| e^{j\theta}$  , dove  $e^{j\theta} = \cos \theta + j \sin \theta$  .

Theta rappresenta l'angolo del vettore:  $\theta = \arctan(\frac{I}{R})$

Il modulo (magnitudo) del vettore può essere calcolato come:  $|C| = \sqrt{R^2 + I^2}$  .



## Serie di Fourier

Ogni funzione periodica  $f(t)$ , con periodo  $T$ , può essere rappresentata come combinazione lineare di seni e coseni moltiplicati per coefficienti appropriati. Ha la forma:

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i \frac{2\pi n}{T} t}$$

nb:  $i$  sarebbe  $j$

dove:

$$c_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-i \frac{2\pi n}{T} t} dt \quad n = 0, \pm 1, \pm 2, \dots,$$

sono i coefficienti.

Notiamo come la base sia una composta da un insieme infinito di seni e coseni, in quanto maggiore è il numero di coefficienti, migliore sarà l'approssimazione.

## Impulso

Un **impulso unitario**, denotato attraverso la **funzione delta di Dirac  $\delta$** , è definito come:

$$\delta(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0 \end{cases}$$

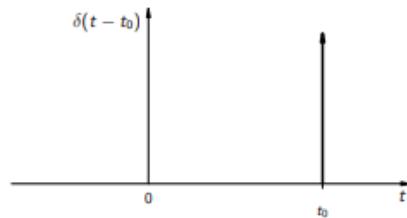
e deve inoltre soddisfare l'identità:

$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

Questo è valido se l'impulso è centrato nell'origine, ma possiamo anche posizionarlo in un punto  $t_0$  arbitrario. In tal caso.

$$\delta(t - t_0) = \begin{cases} \infty, & t = t_0 \\ 0, & t \neq t_0 \end{cases}$$

$$\int_{-\infty}^{+\infty} \delta(t - t_0) dt = 1$$



L'impulso ha una **proprietà** interessante rispetto all'integrazione, detta **sifting**. Tale proprietà fornisce semplicemente il *valore* della funzione  $f(t)$  nella *posizione* dell'impulso  $t_0$ .

$$\int_{-\infty}^{\infty} f(t) \delta(t) dt = f(0)$$

$$\int_{-\infty}^{\infty} f(t) \delta(t - t_0) dt = f(t_0)$$

- Con  $f(t)$  continua in 0 e  $t_0$ , rispettivamente

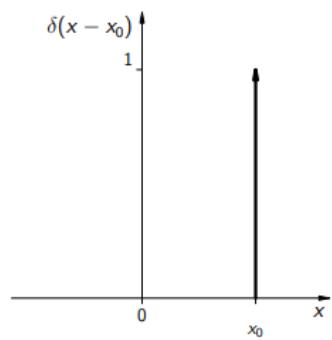
Ciò che abbiamo detto per l'impulso (continuo) vale anche per l'**impulso discreto**, solo che in questo caso l'impulso discreto sarà definito come:

$$\delta(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0 \end{cases}$$

$$\sum_{x=-\infty}^{\infty} \delta(x) = 1$$

La proprietà di sifting è soddisfatta anche in questo caso:

$$\sum_{x=-\infty}^{\infty} f(x) \delta(x) = f(0)$$

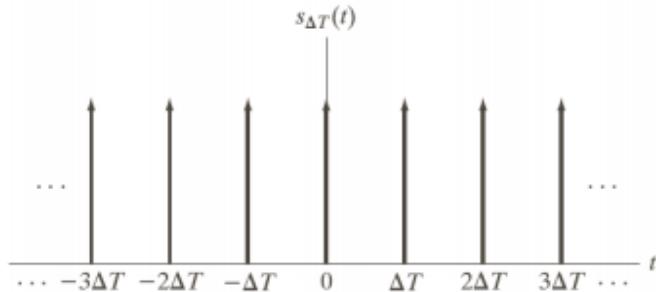


$$\sum_{x=-\infty}^{\infty} f(x) \delta(x - x_0) = f(x_0)$$

nb: ricordiamo che nel discreto gli integrali diventano sommatorie.

Possiamo introdurre anche un **treno di impulsi**  $S_{\Delta T}(t)$ , definito come la somma di infiniti impulsi *periodici*  $\Delta T$  unitari e separati. Dnq,  $S_{\Delta T}(t)$  è una funzione periodica con periodo  $\Delta T$ . Quando sono in  $\Delta T$  ho un impulso, altrimenti ho un valore continuo.

$$s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta T)$$



## Trasformata di Fourier $\mathcal{F}$ (f calligrafica)

Sotto certe condizioni, per ogni funzione  $f(t)$ , la **trasformata di Fourier** di  $f(t)$  è definita come:

$$\mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-\imath 2\pi\nu t} dt$$

poiché  $t$  è integrata, la trasformata di Fourier di  $f(t)$  è una funzione della variabile  $\nu$ . Quindi solitamente indicata con:

$$\mathcal{F}\{f(t)\} = F(\nu)$$

La trasformata di Fourier descrive  $f(t)$  come una combinazione lineare (complessa) di seni e coseni. Utilizzando la formula di Eulero, possiamo esprimere l'equazione come:

$$F(\nu) = \int_{-\infty}^{\infty} f(t) [\cos(2\pi\nu t) - \imath \sin(2\pi\nu t)] dt$$

poiché:

$$e^{\imath\theta} = \cos\theta + \imath \sin\theta$$

Data  $F(\nu)$ , è possibile risalire a  $f(t)$  attraverso la **trasformata di Fourier inversa** (o antitrasformata). In questo caso integriamo rispetto a  $\nu$ , e non a  $t$ .

$$f(t) = \mathcal{F}^{-1}\{F(\nu)\} = \int_{-\infty}^{\infty} F(\nu) e^{\imath 2\pi\nu t} d\nu$$

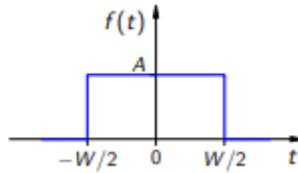
Notiamo come con la trasformata di Fourier passiamo dal dominio del tempo a quello delle frequenze (integriamo rispetto a  $t$ ), mentre con l'antitrasformata passiamo dal dominio delle frequenze a quello del tempo (integriamo rispetto a  $\nu$ ).

Le trasformate  $\mathcal{F}\{f(t)\}$  e  $\mathcal{F}^{-1}\{F(\nu)\}$  sono chiamate **coppia di trasformate di Fourier**.

### ESEMPIO 1: trasformata della funzione BOX

La funzione box (così definita perché ha la forma di uno scatolo) è definita come:

$$f(t) = \begin{cases} A, & -W/2 \leq t \leq W/2 \\ 0, & \text{otherwise} \end{cases}$$



Calcoliamo la trasformata di Fourier a partire da questa funzione:

ricordando che:  $F(\nu) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi\nu t} dt$

allora avremo che:  $F(\nu) = \int_{-W/2}^{W/2} A e^{-i2\pi\nu t} dt$  **nb:**  $A = f(t)$ .

andiamo ad **integrare**, ed otteniamo: (nb:  $\mu$  sarebbe  $\nu$ ).

$$\begin{aligned} &= \frac{-A}{j2\pi\mu} [e^{-j2\pi\mu t}]_{-W/2}^{W/2} = \frac{-A}{j2\pi\mu} [e^{-j\pi\mu W} - e^{j\pi\mu W}] \\ &= \frac{A}{j2\pi\mu} [e^{j\pi\mu W} - e^{-j\pi\mu W}] \\ &= AW \frac{\sin(\pi\mu W)}{(\pi\mu W)} \end{aligned}$$

dove si utilizza l'identità trigonometrica  $\theta = (e^{j\theta} - e^{-j\theta})/2j$ . Dunque il risultato ottenuto prende il nome di **funtione sinc**:

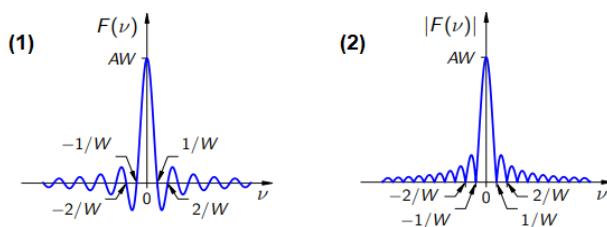
$$= AW \operatorname{sinc}(\nu W)$$

dove:

$$\operatorname{sinc}(t) = \begin{cases} 1, & t = 0 \\ \frac{\sin \pi t}{\pi t}, & \text{otherwise} \end{cases}$$

Tale funzione è importante, in quanto i salti della funzione sono in prossimità di questi zeri.

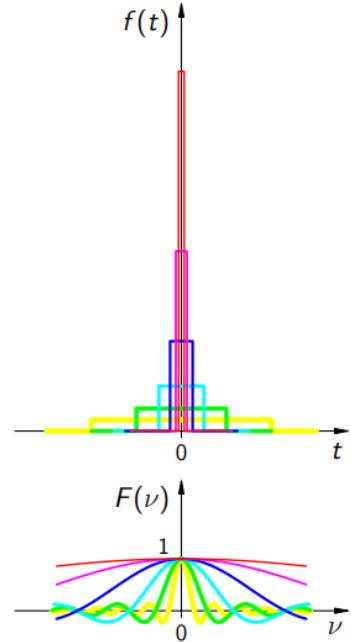
Il grafico di  $F(\nu)$  è (1), tuttavia per agevolare la visualizzazione, si preferisce lavorare con la magnitudo  $|F(\nu)|$  (2), detta anche *spettro di Fourier* o *spettro della Frequenza*:



Notiamo inoltre che c'è una **proporzionalità inversa** tra la funzione box e la sua trasformata. Infatti, più allarghiamo la "box" (ovvero maggiore è il valore di W) più si restringe la funzione *sinc* (e viceversa)

Per questo motivo, la funzione box può essere sfruttata per calcolare la trasformata di Fourier di  $\delta$ .

- Quando si considerano solo funzioni box dove  $AW=1$ , il limite per  $W \rightarrow 0$  è  $\delta$ 
  - $A \rightarrow \infty$
  - L'integrale è 1
  - Al di fuori  $[-W/2, W/2]$  la funzione è 0
- Poiché il lobo principale di  $F(\nu)$  si estende su  $[-1/W, 1/W]$ , per  $W \rightarrow 0$  esso tende ad occupare l'intera linea reale  $(-\infty, \infty)$
- Quindi, il limite della trasformata di Fourier di queste sequenze è la funzione costante 1



### ESEMPIO 2: trasformata di Fourier di un impulso e di un treno di impulsi

Usando la proprietà di sifting di  $\delta$ , la sua trasformata di Fourier può essere calcolata direttamente come:

$$\mathcal{F}(\delta(t)) = F(\nu) = \int_{-\infty}^{\infty} \delta(t) e^{-i2\pi\nu t} dt = e^{-i2\pi\nu 0} = 1$$

Il motivo per cui la "t" all'esponente diventa 0 dopo l'integrale è dovuta proprio alla proprietà di sifting. Infatti ricordo che tale proprietà afferma:  $\int_{-\infty}^{\infty} f(t) \delta(t) dt = f(0)$

Ciò ovviamente possiamo farlo anche se l'impulso non è centrato in 0 ma in un arbitrario punto  $t_0$ . Solo che in questo caso non avremo  $F(\nu) = 1$  ma:  $F(\mu) = \cos(2\pi\mu t_0) - j\sin(2\pi\mu t_0)$ .

Può essere anche mostrato che:

$$\mathcal{F}\{s_{\Delta T}(t)\} = S(\nu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} \delta\left(\nu - \frac{n}{\Delta T}\right)$$

Questo risultato ci dice che la trasformata di Fourier di un treno di impulsi con periodo  $\Delta T$  (ie  $S(\nu)$ ) è *anch'essa un treno di impulsi*, il cui periodo è  $1/\Delta T$ .

Questa proporzionalità inversa tra i periodi di  $s_{\Delta T}(t)$  e  $S(\nu)$  è analoga a quanto visto nei grafici, nel rapporto tra una funzione "box" e la sua trasformata.

## Convoluzione e trasformata di Fourier

La convoluzione tra due funzioni continue,  $f$  ed  $h$ , è definita come:

$$f(t) * h(t) = \int_{-\infty}^{\infty} f(\tau) h(t - \tau) d\tau$$

La convoluzione è proprio rappresentata dal simbolo “ $*$ ”.

Corrisponde a prendere la funzione  $f$ , e per ogni punto della seconda funzione vado a fargliela scivolare sopra.

La *trasformata* della convoluzione di due funzioni continue consiste nel prodotto delle trasformate delle funzioni di  $f$  ed  $h$ .

$$\mathcal{F}\{f(t) * h(t)\} = F(\nu) H(\nu)$$

Vale anche l'opposto, cioè la trasformata di Fourier del prodotto di due funzioni continue consiste nella convoluzione delle trasformate di  $f$  ed  $h$ .

$$\mathcal{F}\{f(t) h(t)\} = F(\nu) * H(\nu)$$

Questo risultato prende il nome di **teorema di convoluzione**.

## Campionamento

Il campionamento è un'operazione che permette di convertire una funzione continua in una sequenza di valori discrete, in modo che possa essere elaborata da un computer.

La **funzione campionata**  $\tilde{f}$  è definita come il prodotto fra la funzione considerata  $f$  e un treno di impulso di opportuno periodo  $\Delta T$ , chiamato passo di campionamento.

Ciascun componente di questa sommatoria è un impulso pesato al quale viene assegnato il valore assunto da  $f(t)$  nella posizione dell'impulso stesso:

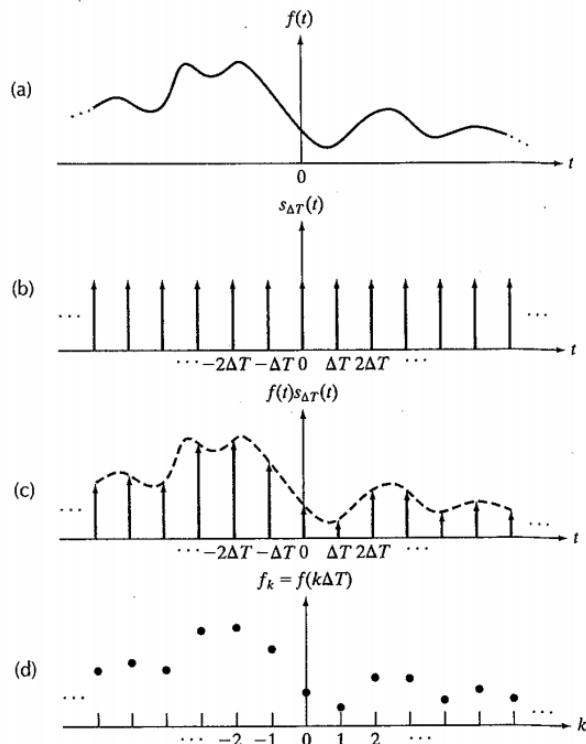
$$\tilde{f}(t) = f(t) s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} f(t) \delta(t - n\Delta T)$$

Il *valore* di ogni campione è dato dalla “forza” dell’impulso pesato ottenuto dall’integrazione. Cioè, il valore  $f_k$  di un campione arbitrario nella sequenza è dato da:

$$f_k = \int_{-\infty}^{\infty} f(t) \delta(t - k\Delta T) dt = f(k\Delta T), \quad k \in \mathbb{Z}$$

**graficamente...**

- (a) una funzione continua.
- (b) treno di impulsi utilizzato per modellare il processo di campionamento.
- (c) funzione campionata formata dal prodotto di (a) e (b).
- (d) valori campione ottenuti dall’integrazione e dall’utilizzo della proprietà di sifting dell’impulso.



nb: la linea tratteggiata in (c) è mostrata solo per convenienza; non fa parte dei dati.

## Trasformata di Fourier di una funzione campionata

La trasformata di Fourier di una funzione campionata  $\tilde{F}(\nu)$  può essere ottenuta usando il teorema di convoluzione, ed è definita come:

$$\begin{aligned}\tilde{F}(\nu) &= \mathcal{F}\{\tilde{f}(t)\} \\ &= \mathcal{F}\{f(t) s_{\Delta T}(t)\} \\ &= F(\nu) * S(\nu)\end{aligned}$$

dalla proprietà di sifting di  $\delta$ :

$$\tilde{F}(\nu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F\left(\nu - \frac{n}{\Delta T}\right)$$

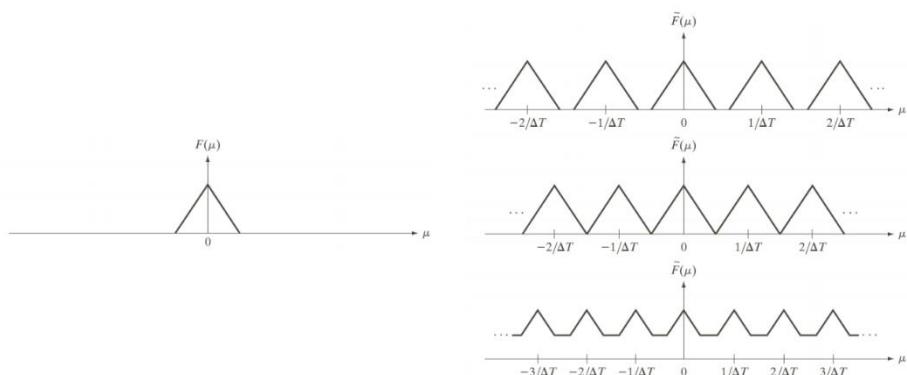
**Quindi**, la trasformata di Fourier della funzione campionata  $\tilde{F}(\nu)$  è una sequenza infinita di copie di  $F(\nu)$  (la trasformata di Fourier della funzione originale  $f(t)$ ). Il periodo, cioè la separazione tra le copie, è  $1/\Delta T$ .

Sebbene  $\tilde{f}(t)$  sia una funzione campionata, la sua trasformata  $\tilde{F}(\nu)$  è *continua* poiché è costituita da copie di  $F(\nu)$ , che a sua volta è una funzione continua.

La separazione tra le copie è regolata da  $1/\Delta T$ , infatti rappresenta la frequenza di campionamento usata per generare le funzioni campionate.

Possiamo distinguere **vari risultati** di campionamento.

- **sovracampionamento**: quando la frequenza di campionamento è abbastanza alta da permettere un sufficiente *distacco* tra i vari periodi, riuscendo a preservare l'integrità della  $F(\nu)$ .  
(Notiamo come le varie copie di  $F(\nu)$  presentino un po' di spazio fra di loro);
- **campionamento critico** (o *tasso di Nyquist*): quando la frequenza di campionamento è *alta appena* per permettere di preservare l'integrità di  $F(\nu)$ .  
(Notiamo come le varie copie di  $F(\nu)$  non abbiano alcun distacco fra di loro, ma non arrivano a sovrapporsi);
- **sottocampionamento**: quando la frequenza di campionamento è *più bassa del minimo richiesto*, non riuscendo quindi a mantenere distinte le copie di  $F(\nu)$  e di conseguenza non riuscendo a preservare l'integrità di  $F(\nu)$ .  
(Notiamo come le varie copie di  $F(\nu)$  si sovrappongono).



## Teorema di campionamento

Una funzione  $f(t)$  è detta **a banda limitata** se la sua trasformata è 0 per i valori di frequenza esterni ad un intervallo  $[-V_{\max}, V_{\max}]$ , detto *banda*.

Possiamo ricostruire  $f(t)$  dalla sua versione campionata se isoliamo una copia di  $F(\nu)$  dalla sequenza periodica di copie di questa funzione contenute in  $\tilde{F}(\nu)$ , cioè la trasformata della funzione campionata  $\tilde{f}(t)$ .

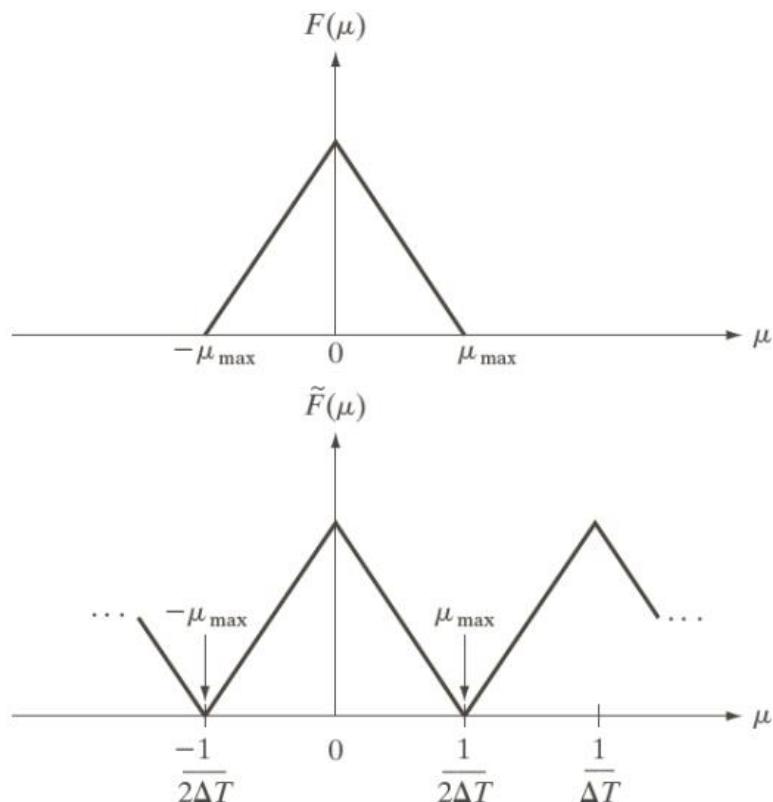
Ottenuta la copia,  $f(t)$  è data dall'antitrasformata di  $F(\nu)$ .

È possibile estrarre da  $\tilde{F}(\nu)$  un singolo periodo uguale a  $F(\nu)$ , se la separazione tra le varie copie è sufficiente. Ciò significa che il tasso di campionamento deve essere almeno il doppio della frequenza massima in  $f(t)$ , cioè se:

$$\frac{1}{\Delta T} > 2V_{\max}$$

Un tasso di campionamento *esattamente* uguale al doppio della frequenza più alta è (come già visto) il *tasso di Nyquist*.

Questo risultato è conosciuto come *teorema del campionamento*.



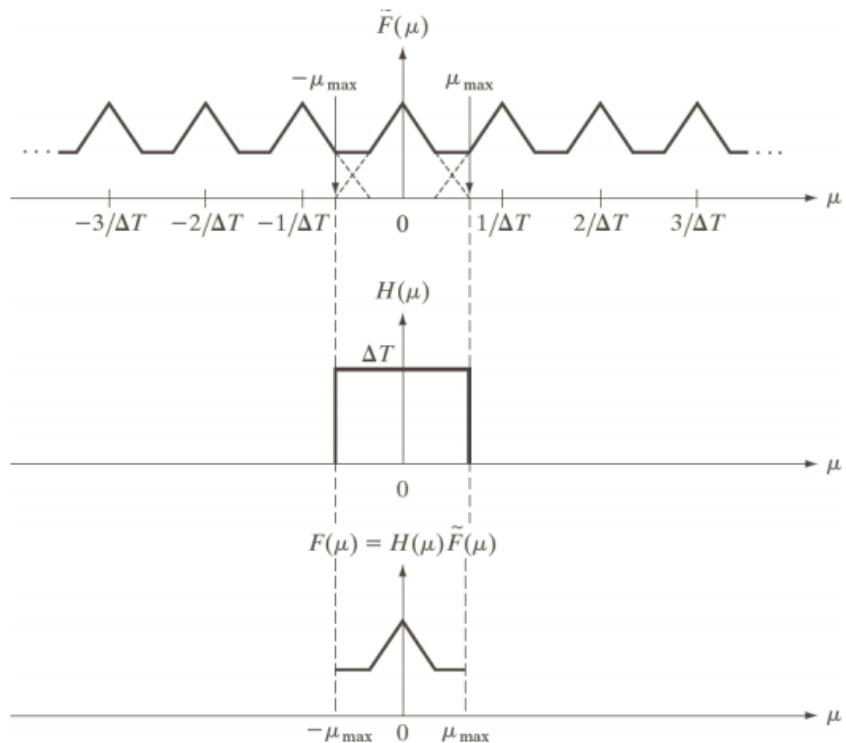
## Aliasing

L'aliasing è il fenomeno per il quale due segnali analogici diversi possono diventare indistinguibili una volta campionati.

In particolar modo, ciò si verifica quando non viene rispettato il *teorema del campionamento di Nyquist*. Cioè, si verifica un sottocampionamento del segnale analogico nel dominio del tempo, e di conseguenza nel dominio delle frequenze si ha la produzione di frequenze non proprie del segnale originario (*alias*), e viceversa dal dominio della frequenza al dominio del tempo si genera una distorsione del segnale originario, divenuto ora non più fedele.

Questo accade perché cerchiamo di campionare la funzione con il tasso di Nyquist quando tuttavia si è verificato un sottocampionamento, e dunque i periodi si sovrappongono, ovvero le componenti tra  $1/\Delta T$  e  $\nu_{\max}$  sono mescolate con le altre componenti.

Questo effetto è chiamato **aliasing**.



Per i casi pratici, l'aliasing è quasi inevitabile, sebbene gli strumenti che effettuano il campionamento adottano un filtro per ridurre le frequenze più elevate.

## Ricostruzione di una funzione campionata

La ricostruzione di una funzione campionata si riduce, nella pratica, all'**interpolazione** tra i campioni stessi.

Per far ciò, dobbiamo riprendere in mano il concetto di convoluzione.

Ricordiamo che:

$$\tilde{F}(\nu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F\left(\nu - \frac{n}{\Delta T}\right)$$

Moltiplichiamo  $\tilde{F}(\nu)$  per una funzione box  $H$ :

$$H(\nu) = \begin{cases} \Delta T, & -\nu_{\max} \leq \nu \leq \nu_{\max} \\ 0, & \text{otherwise} \end{cases}$$

...in modo da ottenere un periodo di  $F$ . Ovvero,  $F(t) = H(\nu) \tilde{F}(\nu)$ .

Allora, può essere applicata la trasformata di Fourier inversa per ottenere  $f$ :

$$f(t) = \mathcal{F}^{-1}\{F(\nu)\} = \mathcal{F}^{-1}\{H(\nu) \tilde{F}(\nu)\} = h(t) * \tilde{f}(t)$$

può essere mostrato che:

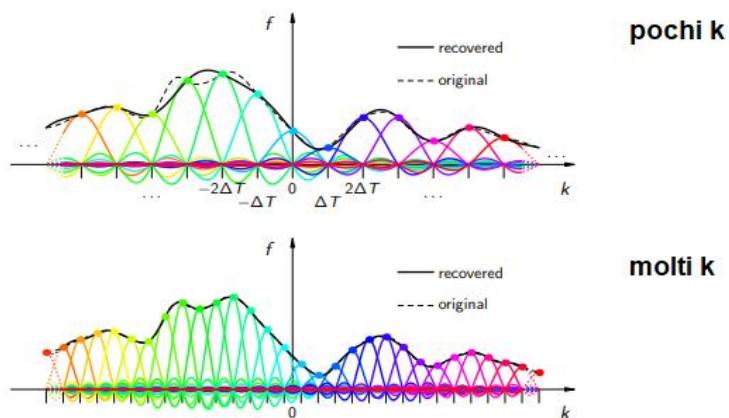
$$f(t) = \sum_{n=-\infty}^{\infty} f(n \Delta T) \operatorname{sinc}\left(\frac{t - n \Delta T}{\Delta T}\right)$$

$f(t)$  è uguale  $f_k$  in  $t = k \Delta T$ .

Altrove, è ottenuta interpolando le funzioni di base sinc shiftate.

Nell'equazione dovremmo interpolare infiniti campioni, ma nella pratica ne facciamo interpolare solo un numero finito e quindi otteniamo un'**approssimazione**.

Maggiore è il numero di campioni, più fedele sarà la ricostruzione.



## Trasformata di Fourier discreta (DFT)

La trasformata di Fourier di una funzione campionata a banda limitata è una funzione *continua e periodica*, che si estende da  $-\infty$  a  $+\infty$ . Tuttavia nella pratica si lavora con un numero finito di campioni ed è per questo che usiamo la trasformata di Fourier discreta.

Il nostro scopo è, data la trasformata di Fourier, **derivare** la sua DFT corrispondente.

*...ragionamento*

Ricodiamo la definizione della Trasformata di Fourier:

$$\tilde{F}(\nu) = \frac{1}{\Delta T} \sum_{n=-\infty}^{\infty} F\left(\nu - \frac{n}{\Delta T}\right)$$

- $\tilde{F}$  può essere espressa in termini di  $\tilde{f}$ :

$$\tilde{F}(\nu) = \int_{-\infty}^{\infty} \tilde{f}(t) e^{-i2\pi\nu t} dt$$

$$\tilde{f}(t) = f(t) s_{\Delta T}(t) = \sum_{n=-\infty}^{\infty} f(t) \delta(t - n\Delta T)$$

- Da cui può essere mostrato:

$$\tilde{F}(\nu) = \sum_{n=-\infty}^{\infty} f_n e^{-i2\pi\nu n \Delta T}$$

- Poiché  $\tilde{F}$  è periodica, tutta l'informazione portata da  $\tilde{F}$  è contenuta in un singolo periodo

- Se consideriamo  $M$  campioni di  $\tilde{F}$  in un periodo,  $\Delta T$ , sono ispezionate le seguenti frequenze:

$$\nu_m = \frac{m}{M \Delta T}, \quad m = 0, \dots, M-1$$

- E i campioni sono:

$$F_m = \sum_{n=0}^{M-1} f_n e^{-i2\pi m n / M}$$

- Gli  $M$  campioni  $\{F_m\}$  sono calcolati usando solo  $M$  campioni di  $f$

Questa trasformata è chiamata **Trasformata di Fourier discreta**.

Gli  $M$  campioni  $\{f_m\}$  possono essere ricostruiti da  $\{F_m\}$  usando la seguente trasformazione:

$$f_n = \frac{1}{M} \sum_{m=0}^{M-1} F_m e^{i2\pi m n / M}, \quad n = 0, \dots, M-1$$

Questa trasformata è chiamata **Trasformata di Fourier discreta inversa**.

La **coppia** di trasformate *diretta* e *inversa* di Fourier sono solitamente rappresentate come:

nb: non è presente alcun riferimento al tempo e alla frequenze. Sono solo sequenze.

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-i2\pi u x / M}, \quad u = 0, \dots, M-1$$

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{i2\pi u x / M}, \quad x = 0, \dots, M-1$$

## Convoluzione circolare

La convoluzione di una sequenza finita di M elementi può essere definita attraverso:

$$g(x) = f(x) * h(x) = \sum_{m=0}^{M-1} f(m) h(x - m)$$

È detta "convoluzione circolare" in quanto periodica, e la periodicità di  $g$  deriva dalla periodicità di  $f$  ed  $h$ .

Attraverso questa operazione, il teorema di convoluzione (per la Trasformata di Fourier continua) può essere esteso alla DFT.

## Relazioni fra il campionamento e gli intervalli in frequenza

Se  $f(x)$  consiste di M campioni di una funzione  $f(t)$ , rispetto a  $\Delta T$  (unitari e separati), l'intervallo temporale che contiene l'insieme  $\{f(x)\}$ , con  $x$  in  $0:M-1$ , è:  $T = M\Delta T$ .

La distanza corrispondente,  $\Delta u$ , nel dominio della frequenza discreta è data da:

$$\Delta u = \frac{1}{M \Delta T} = \frac{1}{T}$$

Quindi, l'intero range in frequenza attraversato da M componenti della DFT è:

$$\Omega = M \Delta u = \frac{1}{\Delta T}$$

Da queste due equazioni, possiamo inoltre vedere che la *risoluzione in frequenza*  $\Delta u$  della DFT dipende dalla durata di  $T$ , e il *range delle frequenze* coperte dalla DFT dipende dal passo di campionamento  $\Delta T$ .

## LEZ 10 – Trasformata di Fourier per immagini (2D)

Estendiamo i concetti visti al dominio bidimensionale.

### Impulso 2D e proprietà di sifting

Un **impulso 2D**, denotato attraverso la **funzione delta di Dirac  $\delta$** , è definito come:

$$\delta(t, z) = \begin{cases} \infty, & t = z = 0 \\ 0, & t \neq 0, z \neq 0 \end{cases}$$

e deve inoltre soddisfare l'identità:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(t, z) dt dz = 1$$

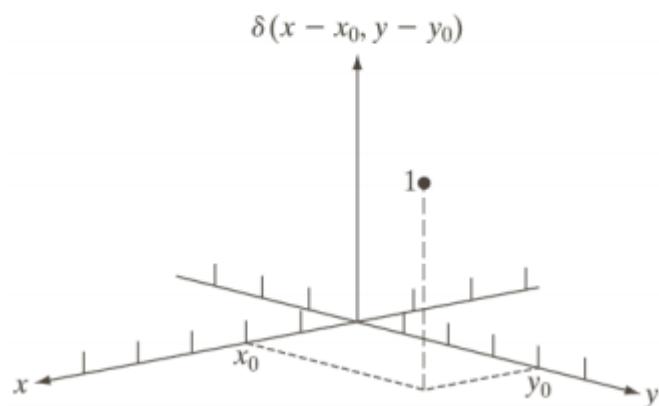
Anche in questo caso l'impulso possiede la proprietà di **sifting**.

Tale proprietà fornisce semplicemente il *valore* della funzione  $f(t, z)$  nella *posizione* dell'impulso  $(t_0, z_0)$ :

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) \delta(t - t_0, z - z_0) dt dz = f(t_0, z_0)$$

Ciò che abbiamo detto per l'impulso (continuo) vale anche per l'**impulso discreto**, solo che in questo caso l'impulso discreto sarà definito come:

$$\delta(x, y) = \begin{cases} 1, & x = y = 0 \\ 0, & \text{otherwise} \end{cases}$$



## Coppia di trasformate di Fourier continue 2D

In questo caso, le trasformate sono solitamente funzioni di due variabili ( $\nu, \mu$ ).

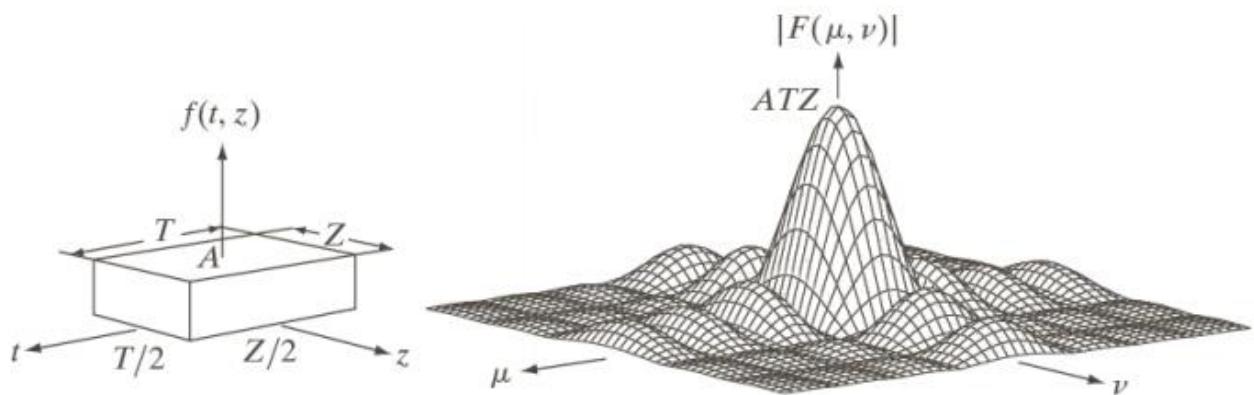
$$F(\nu, \mu) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) e^{-i2\pi(\nu t + \mu z)} dt dz$$

$$f(t, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\nu, \mu) e^{i2\pi(\nu t + \mu z)} d\nu d\mu$$

La **trasformata della funzione box 2D** è definita da due dimensioni: T e Z.

$$\begin{aligned} \text{Avremo che: } F(\nu, \mu) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(t, z) e^{-i2\pi(\nu t + \mu z)} dt dz \\ &= \int_{-\frac{T}{2}}^{+\frac{T}{2}} \int_{-\frac{Z}{2}}^{+\frac{Z}{2}} A e^{-i2\pi(\nu t + \mu z)} dt dz \\ &= ATZ \left[ \frac{\sin(\pi\nu T)}{\pi\nu T} \right] \left[ \frac{\sin(\pi\mu Z)}{\pi\mu Z} \right] \end{aligned}$$

Anche qui c'è **proporzionalità inversa** tra la funzione box e la sua trasformata. Infatti, più saranno grandi le dimensioni di T e Z, più lo spettro si contrae (e viceversa).



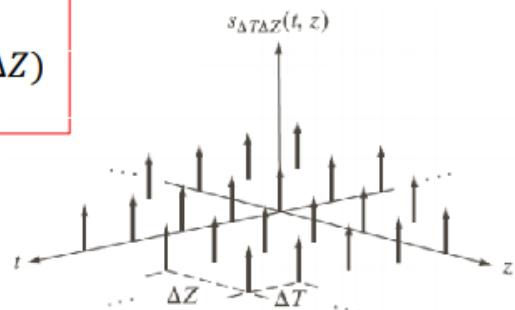
## Teorema di campionamento 2D

In modo analogo al modello monodimensionale, il campionamento bidimensionale può essere modellato utilizzando la funzione di campionamento (treno di impulsi 2D).

Il treno di impulsi ovviamente non terrà conto più solo di  $\Delta T$ , ma anche di  $\Delta Z$ . Ovvero, le distanze tra i campioni lungo l'asse delle  $t$  e delle  $z$  della funzione continua  $f(t, z)$ .

$$\tilde{f}(t, z) = f(t, z) s_{\Delta T \Delta Z}(t, z) = \sum_{m, n=-\infty}^{\infty} f(t) \delta(t - n\Delta T, z - m\Delta Z)$$

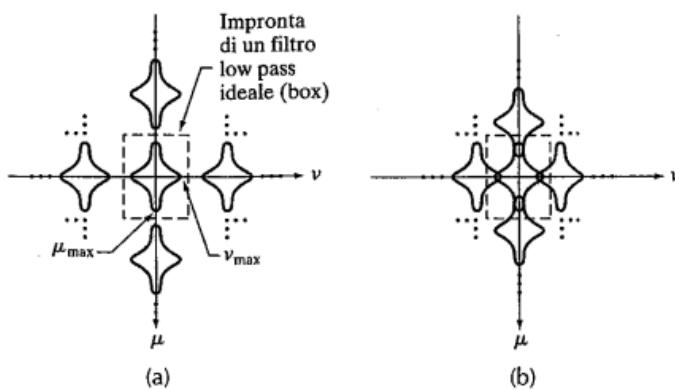
$$s_{\Delta T \Delta Z}(t, z) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \delta(t - m\Delta T, z - n\Delta Z)$$



Anche in questo caso vale il discorso dei vari tipi di campionamento, e in modo analogo il tasso di campionamento deve essere almeno il doppio della frequenza massima in  $f(t, z)$ , cioè se:

$$\frac{1}{\Delta T} > 2 \nu_{\max} \quad \text{and} \quad \frac{1}{\Delta Z} > 2 \mu_{\max}$$

Se non rispettiamo il tasso di Nyquist, si genera *aliasing*.

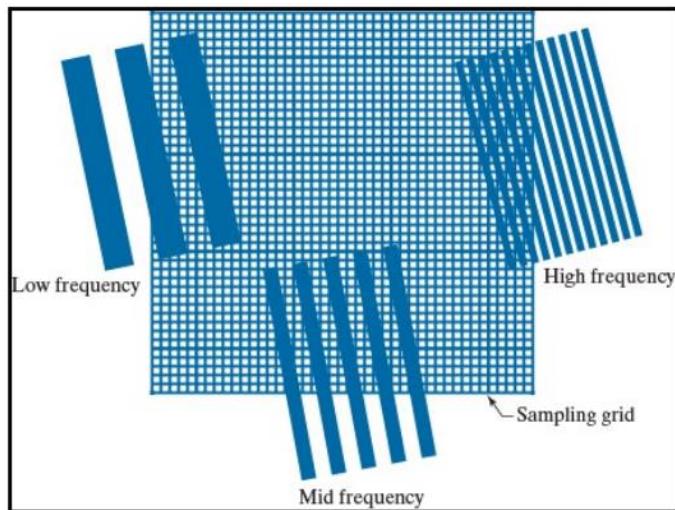


Trasformate di Fourier bidimensionali di due funzioni a banda limitata (a) una sovraccampionata e (b) l'altra sottocampionata.

## Aliasing nelle immagini digitali

L'aliasing è un aspetto sempre presente all'interno delle immagini digitali (così come avveniva nel caso 1D).

Possiamo immaginare di avere una griglia che rappresenta un treno di impulsi, e di sovrapporre un'immagine a tale griglia. In base alla frequenza possiamo ottenere effetti diversi: in particolare, fissato un  $\Delta T$ , maggiore è la frequenza più evidente diventerà l'aliasing. Di conseguenza possono presentarsi dei frastagliamenti di linee o dei propri e veri artefatti, cioè pattern che non appartengono all'immagine originale (es. strisce).

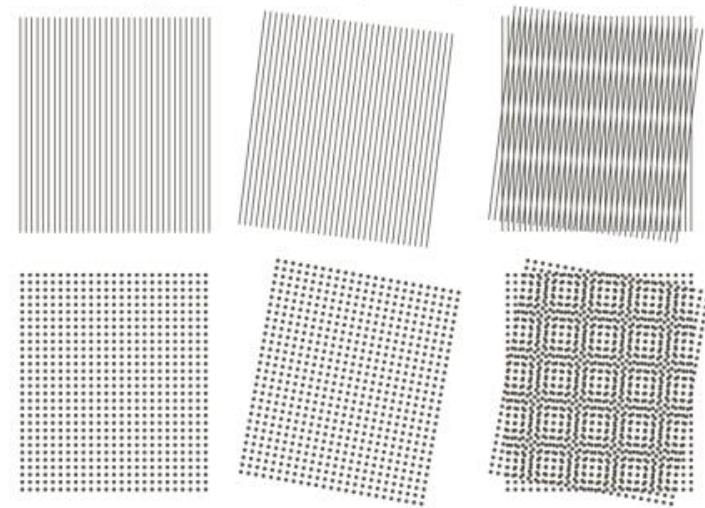


Gli effetti dell'*aliasing* possono essere *ridotti*, sfocando leggermente la scena da digitalizzare in modo da attenuare le frequenze più alte. Non deve essere confuso tuttavia con l'anti-aliasing utilizzato nelle immagini digitali DOPO il campionamento (infatti, non esistono software in grado di ridurre gli artefatti di aliasing indotti dalle violazioni del teorema del campionamento). L'anti-aliasing in questo caso è un vero e proprio filtro che viene applicato nelle lenti delle fotocamere.

- (a) Immagine originale
- (b) Immagine ricampionata
- (c) applicazione dello smoothing prima del ricampionamento
- Oss.: il ricampionamento è stato fatto attraverso la cancellazione di righe e colonne



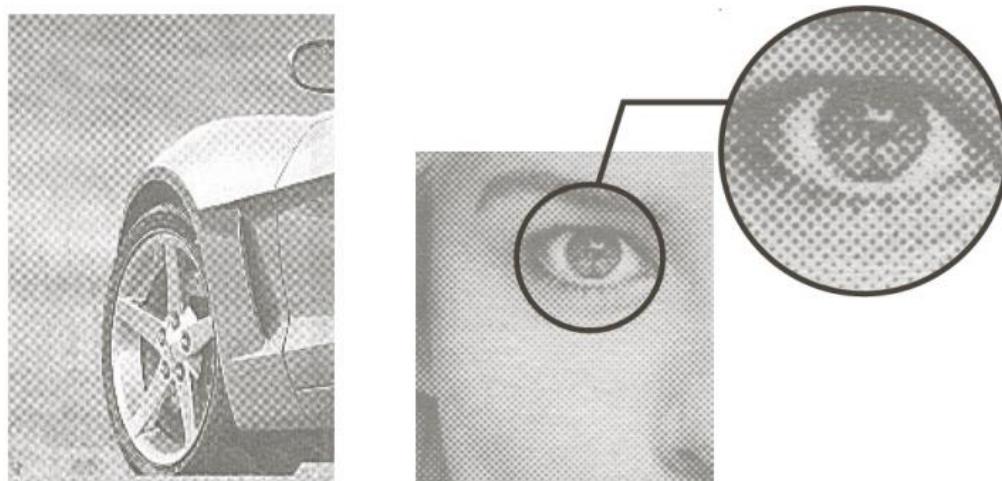
**L'effetto Moiré** è un artefatto causato dalla sovrapposizione di due pattern periodici (come due griglie). Si pensi quando si fa una foto al monitor del PC.



Tale effetto si genera anche nelle stampe di foto: questo perché nell'acquisizione delle immagini il campionamento è verticale e orizzontale; nella stampa abbiamo dei punti allineati di 45 gradi. Questa differenza si manifesta come effetto Moiré.

Inoltre notiamo un altro effetto interessante, ovvero i punti sono più piccoli nelle zone chiare e più grandi nelle zone scure.

Anche in questo caso è possibile applicare dei filtri.



## Coppia DFT 2D

Si tratta semplicemente di aggiungere la seconda dimensione:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(ux/M+vy/N)}$$

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi(ux/M+vy/N)}$$

dove

$$u = 0, \dots, M - 1 \quad v = 0, \dots, N - 1$$

$$x = 0, \dots, M - 1, \quad y = 0, \dots, N - 1$$

Le trasformate di Fourier discrete hanno però delle **proprietà**: *traslazione, rotazione, periodicità e simmetria*.

- **traslazione**: se moltiplichiamo  $f(x,y)$  per un certo esponenziale si trasla l'origine della DFT in  $(u_0, v_0)$  e, viceversa, moltiplicando  $F(u,v)$  per il negativo dello stesso esponente si trasla l'origine della  $f(x,y)$  in  $(x_0, y_0)$ .  
La traslazione non ha effetti sullo spettro di  $F(u,v)$ .

$$f(x, y) e^{i2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$$

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v) e^{-i2\pi(x_0u/M+y_0v/N)}$$

- **rotazione**: trasformata e antitrasformata possono essere rappresentate in coordinate polari. Se ruotiamo l'immagine di partenza di un certo angolo, anche la trasformata sarà ruotata di un certo angolo;
- **periodicità**: come nel caso monodimensionale, la trasformata di Fourier 2D e la sua inversa sono infinitamente periodiche nelle direzioni  $u$  e  $v$ . cioè:

$$F(u, v) = F(u + k_1 M, v + k_2 N)$$

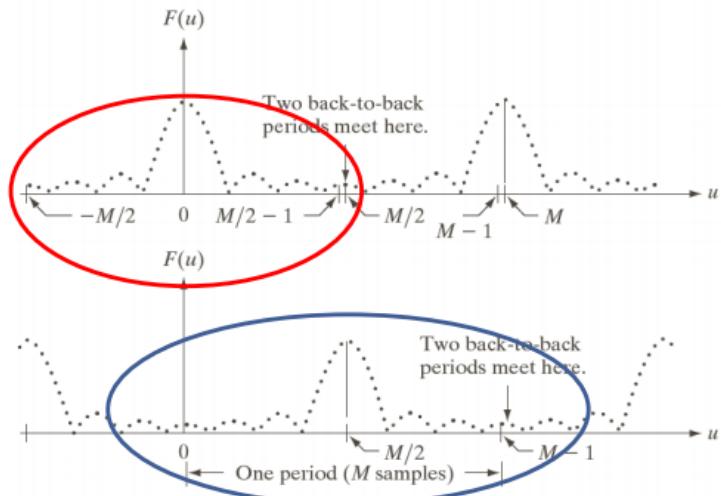
$$f(x, y) = f(x + k_1 M, y + k_2 N)$$

dove  $k_1$  e  $k_2$  sono numeri interi.

Da notare tuttavia che i dati della trasformata nell'intervalle tra 0 e  $M-1$  consistono in due mezzi periodi invertiti che si incontrano nel punto  $M/2$ .

per la visualizzazione e il filtraggio, è più conveniente avere in questo intervallo un periodo completo della trasformata. Dunque dobbiamo traslare la trasformata  $F(u,v)$  in modo che sia centrata in  $(M/2, N/2)$  invece che in  $(0,0)$ .

Per fare ciò, basta porre  $u_0 = M/2$  e  $v_0 = N/2$  nella formula per calcolarci la traslazione di  $f(x,y)$ , che diventerà:  $f(x,y)e^{i\pi(x+y)} = f(x,y)(-1)^{(x+y)}$ .



- **simmetria:** ogni funzione reale o complessa può essere espressa come la somma di una parte e di una parte dispari.

- Funzioni pari (simmetriche)
  - $f(x,y) = f(-x,-y)$  [  $f(x,y) = f(M-x,N-y)$  ]
- Funzioni dispari (antisimmetriche)
  - $f(x,y) = -f(-x,-y)$  [  $f(x,y) = -f(M-x,N-y)$  ]

|     | Spatial Domain <sup>†</sup> | Frequency Domain <sup>‡</sup>               |
|-----|-----------------------------|---|
| 1)  | $f(x,y)$ real               | $\Leftrightarrow F^*(u,v) = F(-u,-v)$       |
| 2)  | $f(x,y)$ imaginary          | $\Leftrightarrow F^*(-u,-v) = -F(u,v)$      |
| 3)  | $f(x,y)$ real               | $\Leftrightarrow R(u,v)$ even; $I(u,v)$ odd |
| 4)  | $f(x,y)$ imaginary          | $\Leftrightarrow R(u,v)$ odd; $I(u,v)$ even |
| 5)  | $f(-x,-y)$ real             | $\Leftrightarrow F^*(u,v)$ complex          |
| 6)  | $f(-x,-y)$ complex          | $\Leftrightarrow F(-u,-v)$ complex          |
| 7)  | $f^*(x,y)$ complex          | $\Leftrightarrow F^*(-u,-v)$ complex        |
| 8)  | $f(x,y)$ real and even      | $\Leftrightarrow F(u,v)$ real and even      |
| 9)  | $f(x,y)$ real and odd       | $\Leftrightarrow F(u,v)$ imaginary and odd  |
| 10) | $f(x,y)$ imaginary and even | $\Leftrightarrow F(u,v)$ imaginary and even |
| 11) | $f(x,y)$ imaginary and odd  | $\Leftrightarrow F(u,v)$ real and odd       |
| 12) | $f(x,y)$ complex and even   | $\Leftrightarrow F(u,v)$ complex and even   |
| 13) | $f(x,y)$ complex and odd    | $\Leftrightarrow F(u,v)$ complex and odd    |

Le proprietà di simmetria in  $f$  coinvolge le corrispondenti proprietà in  $F$  che sono utili nell'elaborazione. Ad esempio, se  $f$  è reale e pari, allora anche  $F$  è reale e pari.

## Spettro di Fourier e angolo di fase

Dal momento che la DFT 2D in generale è complessa, può essere espressa in **forma polare** come:

$$F(u, v) = |F(u, v)| e^{i\phi(u, v)}$$

dove la magnitudo è chiamata **spettro di Fourier**, calcolata come la radice quadrata della somma delle parti reali e immaginarie al quadrato:

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$$

e  $\Phi(u, v)$  è l'**angolo di fase**, calcolato come l'arcotangente del rapporto fra la parte immaginaria e la parte reale.

$$\phi(u, v) = \arctan \left( \frac{I(u, v)}{R(u, v)} \right)$$

Infine, lo **spettro di potenza**  $P(u, v)$  è definito come:

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$$

Poiché si ha che  $F(0,0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$ , allora si ha che:

$$F(0,0) = MN \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = MN |\bar{f}(x, y)| \text{ e quindi } |F(0,0)| = MN |\bar{f}|$$

dove  $\bar{f}$  è il valore medio di  $f$ .

Generalmente,  $F(0,0)$  è molto più grande degli altri termini di  $F$ , e per questo motivo è necessario utilizzare una trasformazione logartimica per visualizzarla.

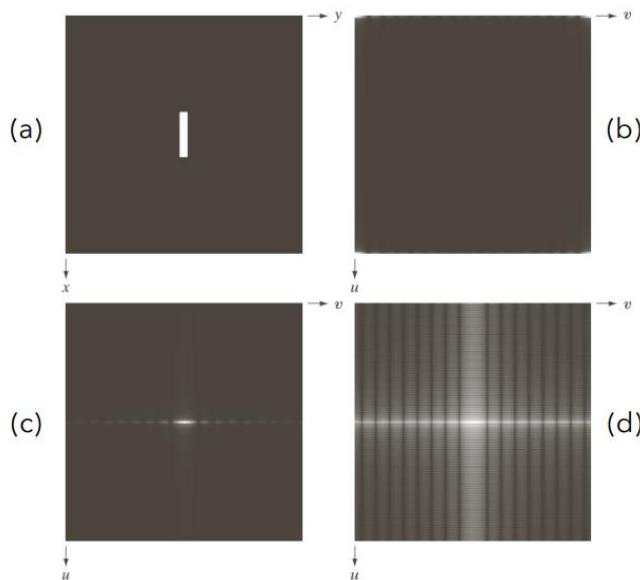
Lo **spettro** è una matrice le cui componenti determinano le intensità dell'immagine. L'**angolo di fase** è una matrice di angoli che ci fornisce numerose informazioni sulla posizione degli oggetti discernibili nell'immagine.

Lo spettro è *insensibile alla traslazione* dell'immagine (il valore assoluto del termine esponenziale è 1). Infatti, anche se traslassimo il "rettangolo" in un altro punto dell'immagine, il risultato sarebbe sempre lo stesso.

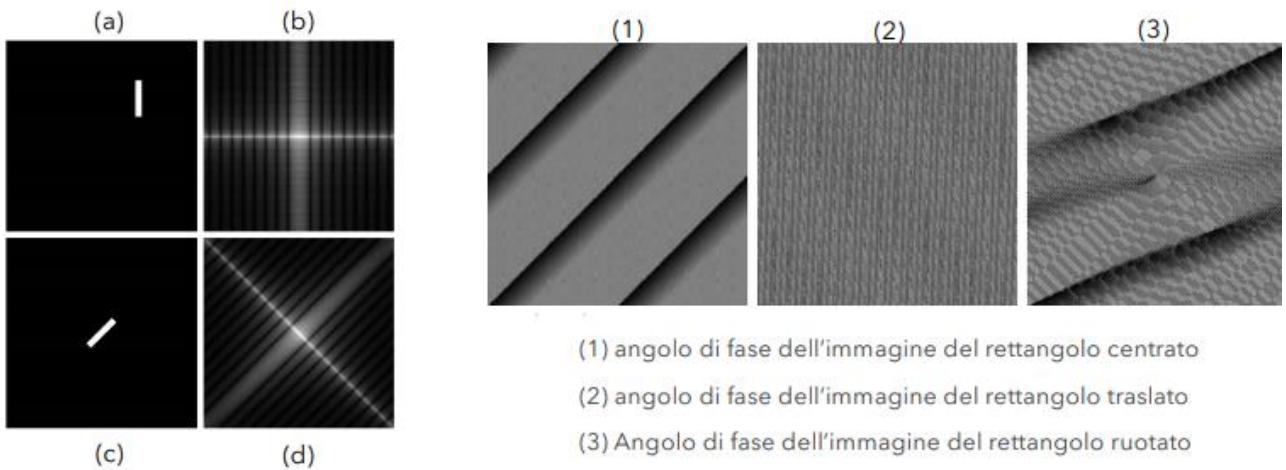
È invece *sensibile alla rotazione*.

Esempio: spettro di Fourier di una funzione semplice (box)

- prendiamo in considerazione una semplice immagine nera con un rettangolo bianco in mezzo;
- mi calcolo lo spettro di Fourier. Il risultato è un'immagine che può sembrare completamente nera ma invece il bianco (ie i valori più alti) sono concentrati agli angoli. Infatti, questo è il risultato della *periodicità*.
- per centrare lo spettro, ovvero per centrare la trasformata in  $(M/2, N/2)$ , moltiplichiamo semplicemente l'immagine in (a) per  $(-1)^{x+y}$ .
- infine, per visualizzare in maniera più evidente il risultato finale (siccome abbiamo molti pixel scuri e pochi luminosissimi), applichiamo una trasformazione logaritmica.

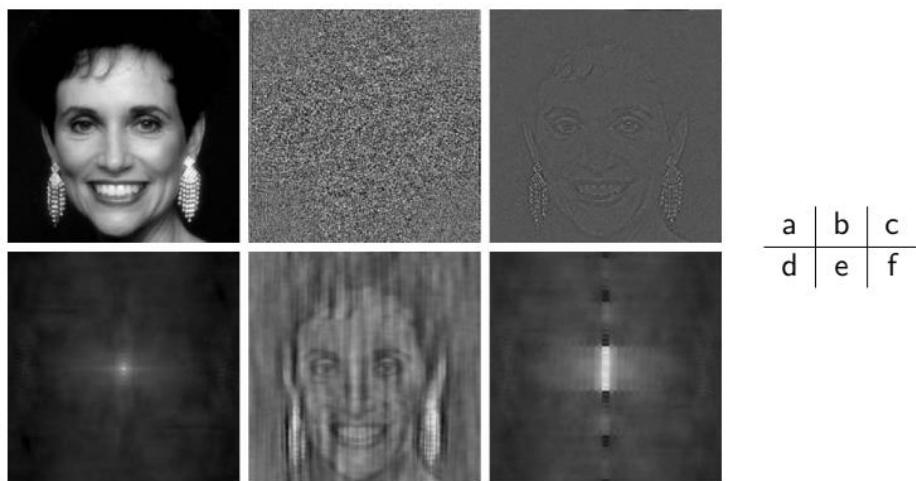


Esempio: traslazione e rotazione di un'immagine



Esempio: Immagine di una donna

- a) immagine originale;
- b) visualizzazione dell'angolo di fase dell'immagine originale;
- c) ricostruzione della donna facendo uso solo dell'angolo di fase. Notiamo come perdiamo le intensità ma il volto è distinguibile;
- d) spettro dell'immagine originale;
- e) risultato: angolo di fase della donna + spettro del rettangolo. Notiamo come il risultato sia sempre il volto di una donna in quanto le informazioni del volto sono contenute all'interno dell'angolo di fase stesso.
- f) risultato: angolo di fase del rettangolo + spettro della donna. Notiamo come il risultato sia il rettangolo (è la situazione opposta ad (e)).



## Teorema di convoluzione 2D

Il teorema di convoluzione 2D è analogo a quello 1D, solo che abbiamo due variabili.

La *trasformata* della convoluzione di due funzioni continue consiste nel prodotto delle trasformate delle funzioni di  $f$  ed  $h$ .

$$\mathcal{F}\{f(x, y) * h(x, y)\} = F(u, v) H(u, v)$$

Vale anche l'opposto, cioè la trasformata di Fourier del prodotto di due funzioni continue consiste nella convoluzione delle trasformate di  $f$  ed  $h$ .

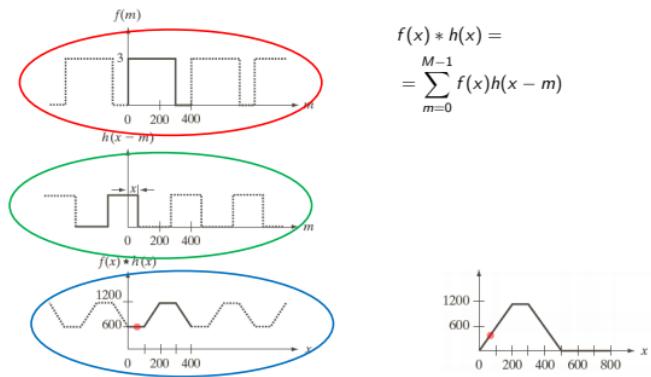
$$\mathcal{F}\{f(x, y) h(x, y)\} = F(u, v) * H(u, v)$$

Se si decide di calcolare la convoluzione spaziale facendo uso della IDFT (Inverse DFT) del prodotto di due trasformate, bisogna tener conto della periodicità.

$$f(x, y) * h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(x, y)h(x - m, y - n),$$

$$x = 0, \dots, M-1 \text{ e } y = 0, \dots, N-1$$

Per capire, immaginiamo un caso monodimensionale. Lo scostamento  $x$  deve essere sufficiente a far scorrere  $h$  su tutta  $f$ . Mentre questo va bene nel caso 1D, nel caso 2D la convoluzione (circolare) di due funzioni periodiche può causare il cosiddetto *errore di wraparound*, ovvero per via della vicinanza dei periodi questi possono interferire l'uno con l'altro.



$$f(x) * h(x) = \sum_{m=0}^{M-1} f(x)h(x - m)$$

Può essere risolto usando lo zero padding. Ovvero, date due sequenze di campioni A e B, si aggiungono zero ad esse in modo che abbiano P elementi (ovvero la stessa lunghezza).

$$P = A + B - 1.$$

In questo modo, la convoluzione spaziale e quella circolare forniscono lo stesso risultato.

*Ad esempio*, supponiamo di avere due immagini di dimensioni  $A \times B$  e  $C \times D$ . Allora eseguiamo lo zero padding lungo righe e colonne in modo che:

$$P = A+C-1$$

$$Q = B+D-1$$

e le immagini risultanti hanno dimensioni  $P \times Q$ .

(nb: se ambo le immagini hanno dimensione uguale, allora  $P = 2M-1$  e  $Q = 2N-1$ ).

Se una funzione non è zero alla fine dell'intervallo, lo zero padding introduce artefatti: le componenti ad altra frequenza nella trasformata.

Tuttavia, possiamo attenuare questo effetto utilizzando *tecniche di windowing* (ad esempio, moltiplicando per una Gaussiana).

## LEZ 11 – Filtraggio nel dominio delle frequenze (1)

La trasformata di Fourier di un'immagine non fornisce un'informazione esplicita, infatti basta pensare che l'equazione  $F(u,v)$  contiene *tutti* i valori di  $f(x,y)$ , modificati dai valori dei termini esponenziali. Ne consegue che di solito è impossibile fare delle associazioni dirette tra le componenti specifiche di un'immagine e la sua trasformata.

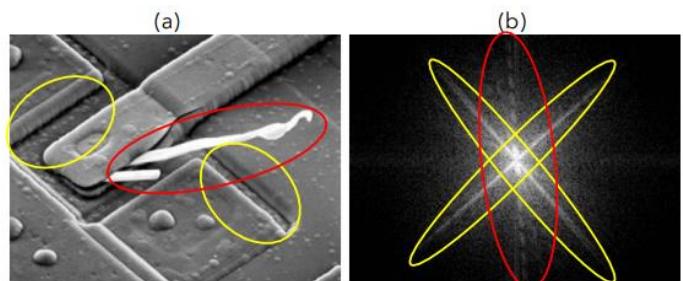
Tuttavia, si possono fare alcune considerazioni generali riguardo le relazioni tra le componenti in frequenza della trasformata di Fourier e le caratteristiche spaziali di un'immagine.

Ad esempio, le alte frequenze corrispondono a variazioni repentine, mentre le basse frequenze corrispondono a variazioni lente (a larga scala).

Alcune informazioni sulla struttura della scena (ad esempio, direzione dei contorni) possono essere dedotte dallo spettro.

Es. notiamo come le componenti più marcate dell'immagine (a), quali i solchi e il filamento al centro, siano evidenziati anche nello spettro (b).

Dunque lo spettro ci aiuta ad effettuare dei paralleli rispetto all'immagine.



In base al caso, potrebbe convenire lavorare nel dominio spaziale o in quello delle frequenze, anche se solitamente viene utilizzato il dominio delle frequenze per costruire i filtri che verranno poi usati nel dominio spaziale (è come se fosse un laboratorio).

Le tecniche di filtraggio nel dominio della frequenza si basano sulla modifica della trasformata di Fourier per raggiungere specifici obiettivi, per poi calcolare l'anttrasformata così da ottenere l'immagine elaborata.

Sostanzialmente, se lavoro nel dominio delle frequenze devo lavorare con questa formula (della quale ci interessa la parte reale, ed ignoriamo per il momento quella complessa).

$$g(x, y) = \mathcal{F}^{-1}\{H(u, v) F(u, v)\}$$

L'immagine filtrata  $g(x,y)$  è data dall'anttrasformata della funzione filtro  $H(u,v)$  per la trasformata di Fourier  $F(u,v)$ .

$F$ ,  $H$ , e  $g$  sono array della stessa dimensione  $M \times N$  dell'immagine originale.

La componente di  $F$  che corrisponde all'origine delle frequenze,  $\mathbf{F}(0,0)$ , chiamata **dc**, corrisponde all'intensità media di  $f$ .

Tenendo ciò in considerazione, possiamo costruire un semplice filtro  $H(u,v)$  il cui valore è: 0 al centro della trasformata; 1 altrove.

Questo filtro annulla il termine dc e lascia invariati gli altri termini. La conseguenza sarà un'immagine più scura (infatti compaiono intensità negative che però poniamo semplicemente a 0).

Questo filtro tuttavia non è molto utile (è solo un esempio).

## Filtri passa basso e Filtri passa alto

Un **filtro passa basso** attenua le alte frequenze e lascia inalterate le basse frequenze. Esso produrrà una copia sfocata dell'immagine.

Un **filtro passa alto** preserva le alte frequenze ed attenua le basse frequenze (la dc, in particolare, dovrebbe sparire).

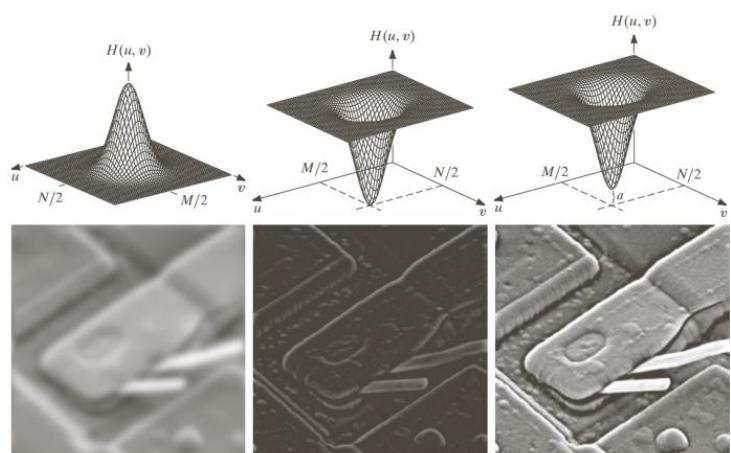
Notiamo un miglioramento dei dettagli ma anche un contrasto minore.

È anche possibile aggiungere una piccola costante al filtro così che le informazioni associate al termine dc siano parzialmente preservate.

L'ampiezza della campana rappresenta l' "aggressività" della funzione filtro.

Il primo è il filtro passa basso, avremo un'immagine sfocata.

Il secondo è il filtro passa alto (notiamo come sia praticamente il primo ma rovesciato); l'immagine risulta scura ma si vedono bene i dettagli. Per questo motivo, posso combinare l'immagine filtrata con quella originale così da ottenere un'immagine più dettagliata ma preservando il contrasto.



## Wraparound e Padding

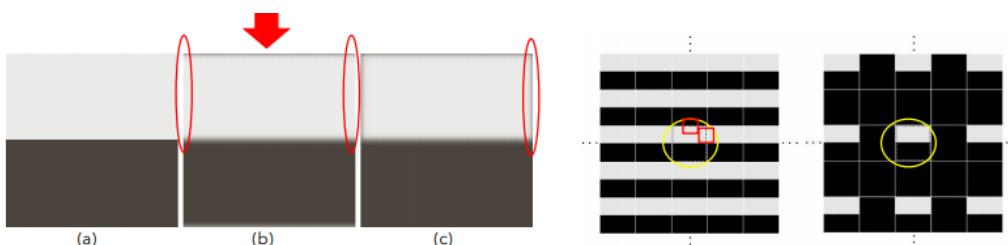
*Ricordiamo:* la convoluzione (circolare) di due funzioni periodiche può causare il cosiddetto errore di wraparound, ovvero per via della vicinanza dei periodi questi possono interferire l'uno con l'altro (ovviamente, se le due funzioni NON sono state sottoposte prima a zero padding).

Quando andiamo ad effettuare lo smoothing (filtro passa basso) ad un'immagine senza padding, la sfocatura non sarà uniforme. Caso contrario invece se aggiungiamo il padding.

es. (b) e (c) sono versioni smoothed di (a), rispettivamente senza e con padding.

Notiamo che (b) non sia sfocato ai lati, mentre (c) è sfocato in maniera uniforme.

Il motivo va trovato nella periodicità dell'uso della DFT: senza padding, è come se l'immagine (a) fosse ripetuta in ogni direzione uguale. Ne consegue che nei lati chiari andiamo a fare una sfocatura su pixel uguali (dunque, non otteniamo alcuna sfocatura); con padding, è come se l'immagine (a) avesse nel suo intorno tutti pixel scuri. Dunque la sfocatura sarà uniforme.



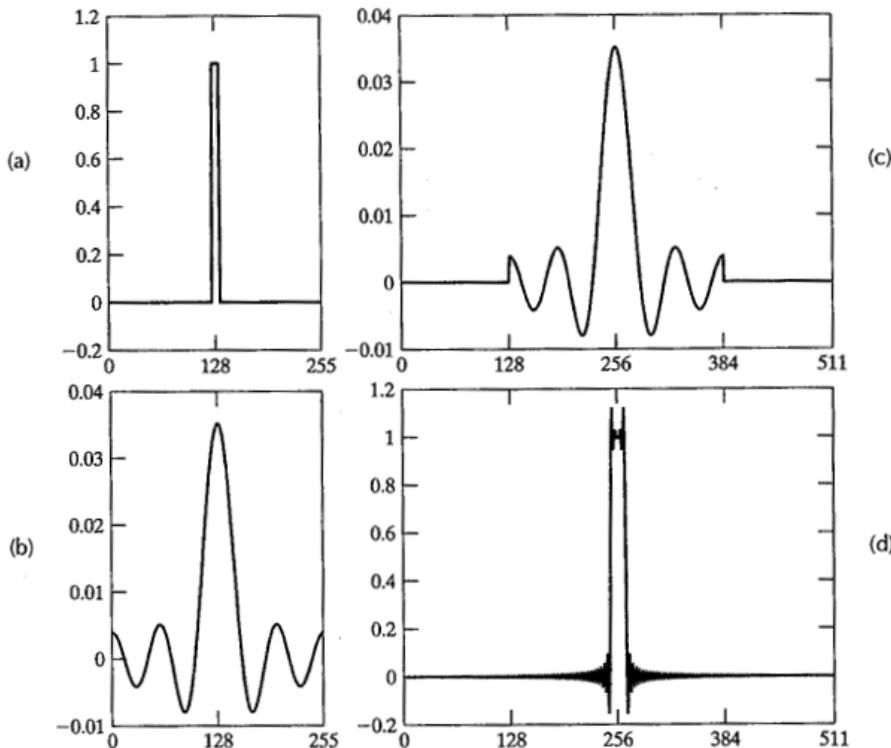
Dunque, il padding dell'immagine originale può evitare errori di wraparound. Ambo l'immagine ed il filtro dovrebbero essere sottoposti a padding, e questo dovrebbe essere applicato nel dominio spaziale.

Se il filtro è specificato nel dominio delle frequenze, esso può essere trasformato nel dominio spaziale, applicato il padding, e trasformato all'indietro nel dominio delle frequenze. Tuttavia questa operazione può comportare la comparsa di un fenomeno detto **ringing**, che causa errori considerevoli.

Possiamo giustificare il ringing facendo un'osservazione: si ricordi la TF di una funzione box, ovvero una funzione *sinc* con frequenze componenti che si estendono all'infinito. Allora anche la trasformata inversa della box deve avere componenti (spaziali) che si estendono all'infinito. Perciò, un qualsiasi troncamento spaziale del filtro per implementare lo zero padding introduce una discontinuità che comporta il fenomeno di ringing nel dominio della frequenza.

Si può quindi concludere che NON è possibile utilizzare un filtro ideale nel dominio della frequenza e contemporaneamente utilizzare lo zero padding per evitare l'errore di wraparound, in quanto non si può gestire un numero infinito di componenti.

Una **migliore procedura** consiste nel fare il padding dell'immagine e successivamente si creeranno i filtri nel dominio della frequenza delle stesse dimensioni delle immagini padded. Ciò provocherà comunque degli errori di wraparound dal momento che non è stato utilizzato un padding per il filtraggio ma questi sono comunque mitigati dal padding dell'immagine (e comunque, è una soluzione preferibile al ringing).



**Figura 4.34** (a) Filtro originale individuato (centrato) nel dominio della frequenza. (b) Rappresentazione spaziale attenuata dal calcolo della IDFT di (a). (c) Risultato del padding di (b) per il doppio della sua lunghezza (si notino le discontinuità). (d) Filtro corrispondente nel dominio della frequenza ottenuto dal calcolo del DFT di (c). Si noti il ringing causato dalla discontinuità in (c). (Le curve appaiono continue a causa dei punti collegati per semplificare l'ispezione visiva.)

## Filtri di shift zero phase (f. senza traslazione di fase)

I *filtris senza traslazione di fase* sono tutti quei filtri che interessano in maniera identica sia la parte reale che la parte immaginaria, e che non hanno effetti sull'angolo di fase (che infatti viene preservato).

Ricordiamo che l'angolo di fase contiene informazioni sul contenuto dell'immagine stessa, dunque una perdita di questa informazioni potrebbe avere conseguenze disastrose.

$$F(u, v) = R(u, v) + \iota I(u, v)$$

$$g(x, y) = \mathcal{F}^{-1}\{H(u, v) R(u, v) + \iota H(u, v) I(u, v)\}$$

## Riepilogo della procedura per il filtraggio delle sequenze

1. Data un'immagine  $f(x,y)$  di dimensioni  $M \times N$ , si ricavano i parametri di padding  $P$  e  $Q$ . Di solito si pone  $P=2M$  e  $Q=2N$ .
2. Si forma un'immagine padded  $f_p(x,y)$  di dimensioni  $P \times Q$ , estendendo la  $f(x,y)$  con il necessario numero di zeri.
3. Opzionalmente, possiamo moltiplicare  $f_p(x,y)$  per  $(-1)^{x+y}$  per centrarne la Trasf.
4. Si calcola la DFT di  $f_p(x,y)$ .
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 calcola  $G(u,v)$  (data come  $G(u,v) = H(u,v) F(u,v)$  ).
7. Si ottiene quindi l'immagine:

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

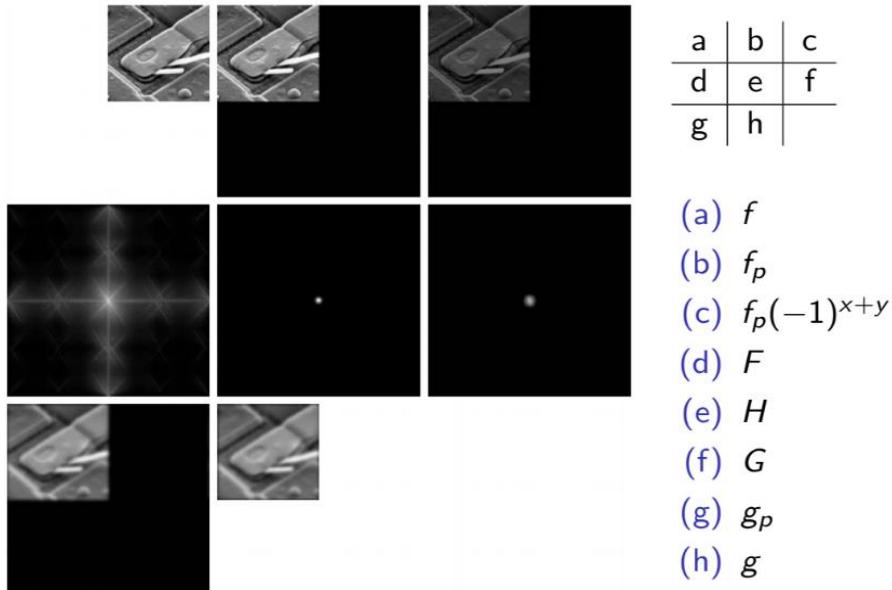
nb:  $(-1)^{x+y}$  è opzionale.

nb: teniamo in considerazione solo la parte reale, e il pedice  $p$  indica che si sta trattando array padded.

8. Il risultato finale  $g(x,y)$  è ottenuto effettuando il cropping della regione  $M \times N$  di  $g_p(x,y)$ .

## Esempio

- a) immagine di partenza  $f$ , di dimensioni  $M \times N$ ; .1
- b) estensione dell'immagine  $f$ ,  $f_p$ , di dimensioni  $P \times Q$ ; .2
- c) risultato della moltiplicazione di  $f$  per  $(-1)^{x+y}$ ; .3
- d) spettro di  $F_p$ ; .4 (?)
- e) filtro  $H$  di dimensioni  $P \times Q$ ; .5
- f)  $G(u,v)$ ; .6
- g)  $g_p$ ; .7
- h) risultato finale  $g$ , ottenuto selezionando le prime  $M$  righe e  $N$  colonne di  $g_p$ . .8



## Risposta di impulso

Sappiamo che il filtraggio nel dominio della frequenza  $G = H(u,v) F(u,v)$ . Dato un filtro  $H(u,v)$  possiamo trovare una rappresentazione equivalente nel dominio spaziale:

Se  $f(x,y) = \delta(x,y)$  allora  $F(u,v) = 1$ .

Quindi,  $G(u,v) = H(u,v) F(u,v) = H(u,v)*1 = H(u,v)$ , da cui  $g(x,y) = h(x,y)$

$h(x,y)$  e  $H(u,v)$  sono una coppia di Fourier, e dove  $h(x,y)$  è chiamata risposta di impulso di  $H(u,v)$ .

Questi filtri sono detti “a risposta di impulso finiti” poiché tutte le quantità nell’implementazione discreta sono finite.

## Progettazione di filtri spaziali

La conoscenza del dominio delle frequenze può essere usata per *guidare* la progettazione di filtri spaziali (è come se fosse un laboratorio appunto).

Ad esempio, prendiamo in considerazioni i filtri che utilizzano le funzioni gaussiane.

Sia  $H(u)$  il filtro gaussiano 1D nel dominio della frequenza:

$$H(u) = Ae^{-\frac{u^2}{2\sigma^2}}$$

il corrispondente filtro nel dominio spaziale è ottenuto prendendo l'antitrasformata di Fourier di  $H(u)$ :

$$h(x) = \sqrt{2\pi}\sigma Ae^{-2\pi^2\sigma^2x^2}$$

queste equazioni sono importanti per due motivi:

1. rappresentano una coppia di Fourier, le cui componenti sono entrambe gaussiane e reali (dunque non abbiamo a che fare con numeri complessi); inoltre, le curve gaussiane sono intuitive e facili da trattare;
2. entrambe le funzioni hanno un comportamento che si influenza reciprocamente: quando  $H(u)$  ha un profilo ampio,  $h(x)$  ha un profilo ristretto, e viceversa.

Il risultato sarà un *filtro passa basso*.

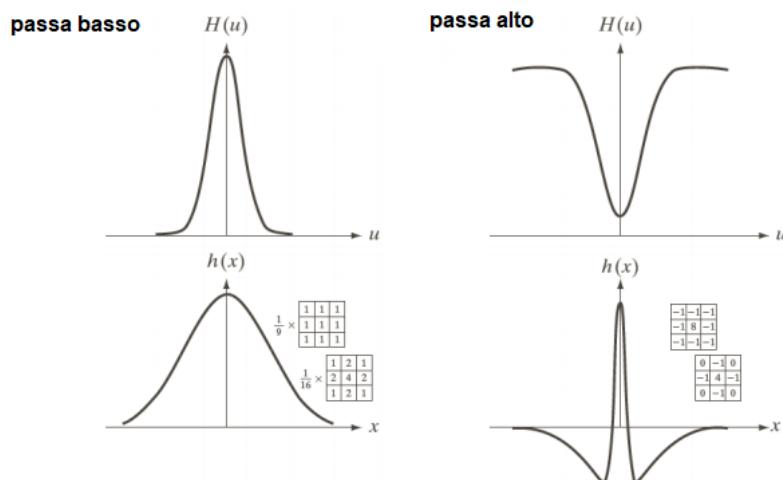
Possiamo anche utilizzare filtri più complessi. Ad esempio, un filtro definito come differenza di due Gaussiane che corrisponde ad un *filtro passa alto*.

Anche qui valgono i due punti espressi prima.

$$H(u) = Ae^{-\frac{u^2}{2\sigma_1^2}} - Be^{-\frac{u^2}{2\sigma_2^2}}$$

$$h(x) = \sqrt{2\pi}\sigma_1 Ae^{-2\pi^2\sigma_1^2x^2} - \sqrt{2\pi}\sigma_2 Be^{-2\pi^2\sigma_2^2x^2}$$

$$A \geq B \text{ e } \sigma_1 \geq \sigma_2$$



## Filtri passa basso

Possiamo distinguere diversi filtri passa basso: *ideale*, *butterworth* e *gaussiano*.

- Un **filtro passa basso ideale** è un filtro che lascia passare, senza attenuarle, tutte le frequenze all'interno di un cerchio di raggio  $D_0$  centrato sull'origine, e “taglia fuori” tutte le frequenze al di fuori di questo cerchio.

Al crescere del raggio, si rimuove una quantità di spettro via via minore, e anche la sfocatura diminuisce. La discontinuità è *marcata*.

È definito come:

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

dove  $H$  è lo spettro del filtro,  $D$  è la distanza tra i pixel e  $D_0$  è la “frequenza di cut-off”.

- Un **filtro passa basso Butterworth** è un filtro che ha una discontinuità *graduale* tra le frequenze che lascia passare (quelle all'interno di un cerchio di raggio  $D_0$ ) e quelle che “taglia fuori”.

È definito come:

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

È importante l'ordine  $n$ , in quanto maggiore è questo valore maggiore è l'effettivo filtraggio low pass ma anche maggiore è la presenza di ringing.

I filtri Butterworth di ordine 1 non presentano ringing, quelli di ordine 2 costituiscono un buon compromesso tra filtraggio e ringing (ringing che comunque è molto più accettabile rispetto al filtro ideale).

**nb:** il motivo per cui il ringing è maggiormente presente nel filtro ideale è dovuto alla marcata discontinuità delle frequenze, che invece nel filtro Butterworth è graduale.

- Un **filtro passa basso Gaussiano** è un filtro che ha una discontinuità *graduale* tra le frequenze che lascia passare e quelle che “taglia fuori”, ma a differenza del filtro Butterworth (in particolare  $n=2$ ) ha un minor effetto di smoothing ma con il vantaggio di non introdurre ringing.

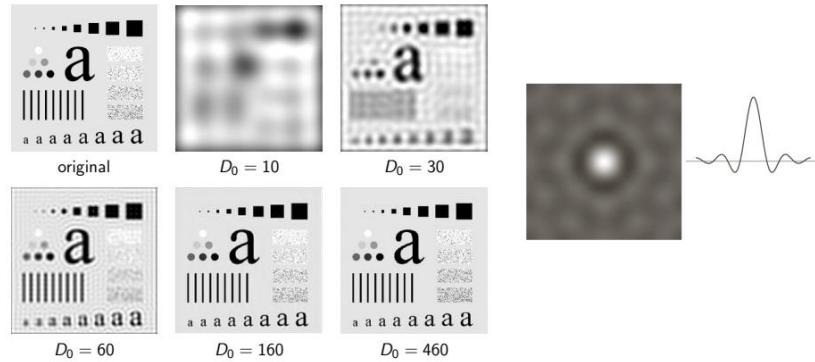
Si tratta di una caratteristica utile nella pratica per quelle situazioni in cui bisogna essere sicuri di non introdurre artefatti (es. imaging medico).

È definito come:

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

*A confronto:*

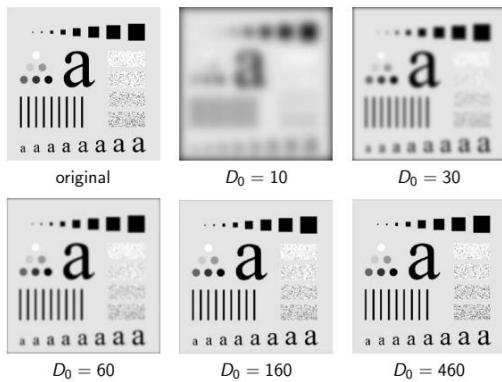
#### FILTRAGGIO FILTRO PASSA BASSO IDEALE



#### FILTRAGGIO PASSA BASSO DI BUTTERWORTH



#### FILTRAGGIO PASSA BASSO GAUSSIANO



*Esempi pratici:*

1. testo scansionato a bassa risoluzione corrotto dal rumore (i caratteri sono tipo "spezzettati"). Sebbene per noi possa ancora essere facilmente riconoscibile, di sicuro non lo è per i sistemi di riconoscimento dei caratteri.  
Il filtraggio passo basso unisce le parti dei caratteri spezzati (sfocandoli) così da questi sistemi possano anche loro avere vita facile.
2. rimozione delle rughe (che in effetti sono dettagli).
3. rimozione delle linee di scansione.

## Filtri passa alto

I filtri passa alto si ottengono dai corrispondenti filtri passa basso. Nello specifico, sottraiamo ad 1 il filtro passa basso corrispondente:

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

nb: HP = High Pass; LP = Low Pass.

Di conseguenza, possiamo distinguere 3 filtri passa alto: *ideale*, *butterworth* e *gaussiano*. Di base vale il discorso fatto per il passa basso, ma al contrario (ad esempio, il filtro ideale *taglia* tutte le frequenze di un cerchio di raggio  $D_0$  centrato sull'origine, e lascia passare tutte le frequenze al di fuori di questo cerchio).

- Un **filtro passo alto ideale** è definito come:

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

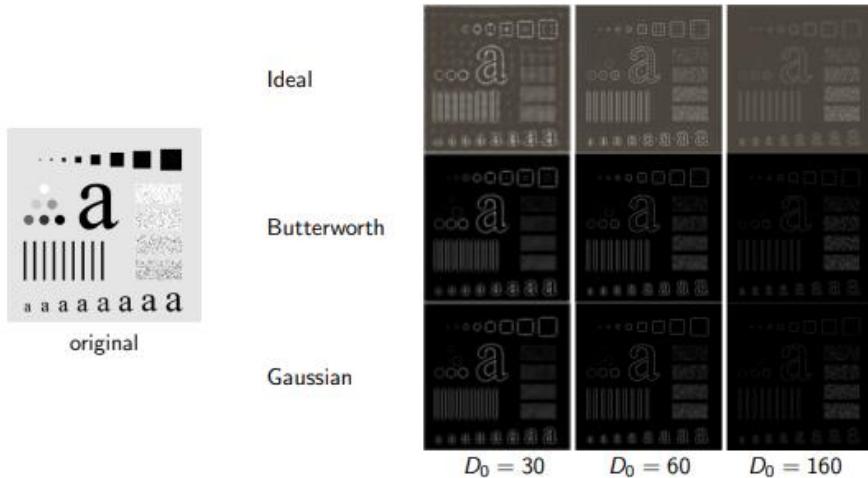
- Un **filtro passo alto Butterworth** è definito come:

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

- Un **filtro passo alto Gaussiano** è definito come:

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

A confronto:



Esempi pratici:

Accentuazione delle impronte digitali.

## LEZ 12 – Filtraggio nel dominio delle frequenze (2)

### Laplaciano

Abbiamo visto come applicare il Laplaciano nel dominio dello spazio, adesso faremo lo stesso ma nel dominio delle frequenze. In entrambi i casi, otterremo lo stesso risultato.

Il **filtro Laplaciano** nel dominio delle frequenze è:

$$H(u, v) = -4\pi^2(u^2 + v^2)$$

di conseguenza, l'**immagine Laplaciana** è ottenuta come:

$$\nabla^2 f(x, y) = \mathcal{F}^{-1}\{H(u, v) F(u, v)\}$$

Il miglioramento, ovvero l'**immagine filtrata g(x,y)**, sarà data da:

$$g(x, y) = f(x, y) + c \nabla^2 f(x, y) \quad c = -1$$

Utilizzando i metodi del dominio della frequenza possiamo usare la tecnica di *mascheramento di unsharp* applicando una **maschera di unsharp**  $g_{\text{mask}}(x, y)$ , definita come:

$$g_{\text{mask}}(x, y) = f(x, y) - f_{\text{LP}}(x, y)$$

dove  $f_{\text{LP}}(x, y)$  è un'immagine sfocata analoga a  $f(x, y)$ , calcolata a sua volta come:

$$f_{\text{LP}}(x, y) = \mathcal{F}^{-1}\{H_{\text{LP}}(u, v) F(u, v)\}$$

dove  $H_{\text{LP}}(u, v)$  è un filtro *low pass* e  $F(u, v)$  è la trasformata di Fourier.

**Nel dominio spaziale, l'immagine filtrata f(x,y)** sarà ottenuta come:

$$g(x, y) = f(x, y) + k g_{\text{mask}}(x, y)$$

dove: quando  $k = 1$  questa espressione definisce l'unsharp masking;  
quando  $k > 1$  definisce il filtraggio highboost.

Utilizzando i risultati precedenti, possimo esprimere l'espressione dell'**immagine filtrata nel dominio della frequenza** come:

$$g(x, y) = \mathcal{F}^{-1}\{(1 + k(1 - H_{\text{LP}}(u, v))) F(u, v)\}$$

La stessa espressione può anche essere espressa in termini di filtro *high pass*, basta solo tener conto che si tratta dell'opposto del filtro Low Pass, e dunque invece di avere  $(1 - H_{\text{LP}})$  avremo solo  $H_{\text{HP}}$ .

$$g(x, y) = \mathcal{F}^{-1}\{(1 + k H_{\text{HP}}(u, v)) F(u, v)\}$$

Possiamo anche *generalizzare*:

$$g(x, y) = \mathcal{F}^{-1}\{(k_1 + k_2 H_{HP}(u, v)) F(u, v)\}$$

dove  $k_1$  specifica la distanza dall'origine, e  $k_2$  determina il contributo delle alte frequenze. Entrambi devono essere  $\geq 0$ .

## Filtraggio omorfico

L'aspetto di un'immagine può anche essere migliorato attraverso operazioni simultanee di compressione dei range dell'intensità e il miglioramento del contrasto.

In questo caso, si parla di *filtraggio omorfico*.

Infatti, un'immagine spaziale può essere espressa come la combinazione di due componenti: *illuminazione  $i$*  e *riflettanza  $r$* .

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

Tuttavia, questa relazione NON può essere espressa nel dominio delle frequenze, e questo perché la trasformata di Fourier di un prodotto non risulta essere uguale al prodotto della trasformate (proprietà invece valida nel dominio spaziale), dunque:  $\mathcal{F}\{f(x, y)\} \neq \mathcal{F}\{i(x, y)\} \mathcal{F}\{r(x, y)\}$ .

Tuttavia, possiamo adottare un **piccolo trucco**: esprimo la *versione logaritmica* dell'immagine. Infatti, così nel dominio delle frequenze sarò in grado di separare le due componenti.

La **versione logaritmica  $z(x,y)$** , sarà:

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

di conseguenza, avremo che la **trasformata di Fourier discreta  $Z(x,y)$**  sarà:

$$Z(u, v) = \mathcal{F}\{z(x, y)\} = \mathcal{F}\{\log i(x, y)\} + \mathcal{F}\{\log r(x, y)\}$$

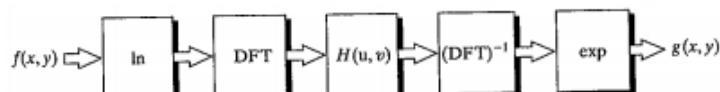
a questo punto, posso applicare il filtro  $H(u,v)$  nel dominio delle frequenze. Applicare il filtro a  $Z(u,v)$  significa applicare la trasformata a entrambe le componenti:

$$S(u, v) = H(u, v)Z(u, v) = H(u, v)F_i(x, y) + H(u, v)F_r(x, y)$$

Se voglio ritornare nel dominio spaziale, siccome  $z(x,y)$  è stata ottenuta considerando il logaritmo naturale dell'immagine di input, mi basterà invertire la funzione  $S(u,v)$  utilizzando gli esponenziali:

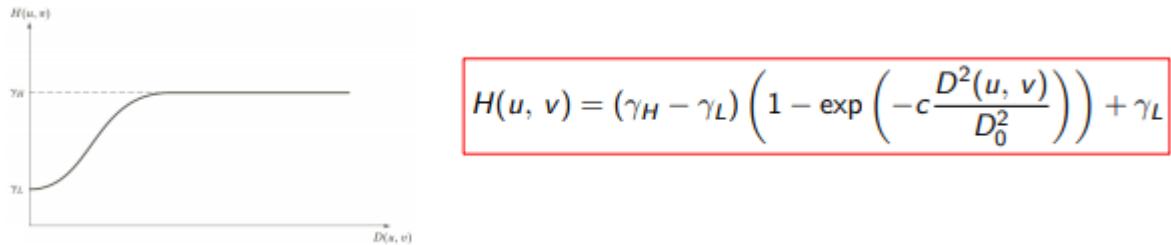
$$g(x, y) = e^{\mathcal{F}^{-1}\{S(u, v)\}}$$

**Ricapitolando** i passi fondamentali: nel dominio delle frequenze per poter applicare il filtraggio omorfico, che agisce sulle componenti dell'illuminazione e della riflettanza, devo esprimere l'immagine  $f(x,y)$  come una funzione logaritmica, ne calcolo la DFT, applico il filtro, e ritorno nel dominio spaziale invertendo la trasformazione logaritmica.



Il vantaggio del filtraggio omorfico è che mi permette di migliorare l'immagine ma mantendo il controllo sulle componenti di illuminazione e riflettanza.

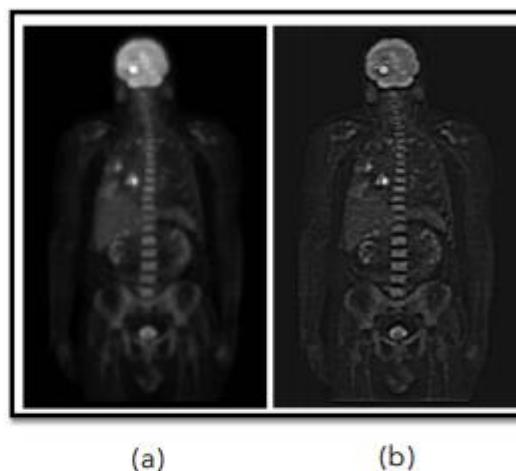
Il processo può essere realizzato usando un filtro come:



Se i parametri  $\gamma_L$  e  $\gamma_H$  sono scelti in modo che  $\gamma_L < 1$  e  $\gamma_H > 1$ , la funzione filtro tende ad attenuare gli effetti prodotti dalla basse frequenze (illuminazione) e ad amplificare gli effetti dovuti alle alte frequenze (riflettanza).

$D(u, v)$  rappresenta la distanza dal centro.

Può essere usata ad esempio in campo medico, per migliorare una PET.



(a) Immagine scansione PET dell'intero corpo  
 (b) miglioramento attraverso il filtraggio omomorfico con  $\gamma_L = 0.25$  e  $\gamma_H = 2$ ,  $D_0 = 80$

## Filtraggio selettivo

Il filtraggio selettivo è una tipologia di filtraggio che ha l'obiettivo di processare solo determinate bande di frequenza o piccole regioni del rettangolo della frequenza, e non l'intero rettangolo. In particolare:

- alla prima categoria appartengono i filtri **band pass** (passa banda) e i filtri **band reject** (elimina banda);
- alla seconda categoria appartengono i filtri **notch**.

I filtri **band pass** e **band reject** sono relativamente semplici da costruire. In particolar modo possiamo far riferimento ai filtri: ideale, di Butterworth, e Gaussiano.

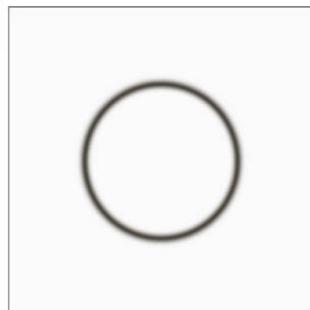
| Ideal (IBRF)   | Gaussian (GBRF)   | Butterworth (BBRF)  |
|--|---|---|
| $H(u,v) = \begin{cases} 0 & \text{if } C_0 - \frac{W}{2} \leq D(u,v) \leq C_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$ | $H(u,v) = 1 - e^{-\left[\frac{D^2(u,v) - C_0^2}{D(u,v)W}\right]^2}$ | $H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)W}{D^2(u,v) - C_0^2}\right]^{2n}}$ |

dove  $D(u,v)$  è la distanza dal centro del rettangolo della frequenza,  $C_0$  è il raggio del cerchio della banda, e  $W$  è uguale alla larghezza della banda.

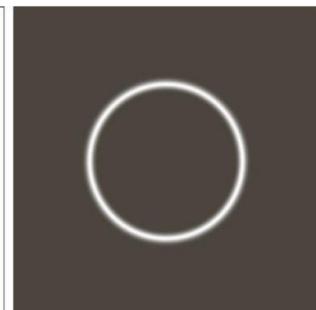
Il filtro band pass si ottiene dal filtro band reject nella stessa maniera in cui otteniamo un filtro high pass da un filtro low pass.

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

Filtro Gaussiano  
band-reject



Filtro Gaussiano  
pass-band



I filtri **notch** eliminano (o lasciano passare) le frequenze in un contorno predefinito del centro del rettangolo della frequenza.

Si può progettare un filtro con “zero” in corrispondenza di rumori periodici così da eliminarlo.

Poiché i filtri zero-phase shift sono simmetrici rispetto al centro del rettangolo della frequenza, un filtro notch con centro  $(u_0, v_0)$  deve avere un corrispondente punto notch in  $(-u_0, -v_0)$ .

I filtri notch band reject sono costruiti come prodotti di filtri high pass i cui centri sono stati traslati nei centri dei notch. La forma generale è:

$$H_{NR}(u, v) = \prod_{k=1}^Q H_k(u, v) H_{-k}(u, v)$$

dove  $H_k(u, v)$  e  $H_{-k}(u, v)$  sono filtri passa alto con centri  $(u_k, v_k)$  e  $(-u_k, -v_k)$ , rispettivamente.

I centri sono specificati rispetto al centro del rettangolo della frequenza  $(M/2, N/2)$  e la distanza per ogni filtro è calcolata come:

$$D_k(u, v) = \left[ \left( u - \frac{M}{2} - uk \right)^2 + \left( v - \frac{N}{2} - vk \right)^2 \right]^{\frac{1}{2}} \text{ e } D_{-k}(u, v) = \left[ \left( u - \frac{M}{2} + u_k \right)^2 + \left( v - \frac{N}{2} + v_k \right)^2 \right]^{\frac{1}{2}}$$

Il filtro notch passante si ottiene dal filtro notch di rigetto nella stessa maniera in cui otteniamo un filtro high pass da un filtro low pass.

$$H_{NP}(u, v) = 1 - H_{NR}(u, v)$$

## Computazione della DFT

Il Teorema della computazione lega il mondo del dominio delle frequenze con quello del dominio spaziale.

La DFT è separabile in trasformate monodimensionali, cioè la DFT 2D di  $f(x, y)$  si può ottenere calcolando la trasformata di ciascuna riga di  $f(x, y)$  e poi di ciascuna colonna del risultato. Ciò semplifica l'implementazione in quanto si considera una sola variabile per volta. Ciò vale anche per la IDTF 2D (trasformata di Fourier discreta inversa).

La complessità della DFT è  $O((MN)^2)$  dove  $MN$  è la dimensione dell'immagine. Tuttavia, la DFT può essere implementata con la FFT (Fast Fourier Transform) che ha una complessità  $O(MN \log_2(M))$ , che però richiederebbe che il numero di campioni sia uguali ad una potenza intera di 2, ma questo vincolo non è un requisito generale.

La FFT risulta dunque essere molto più veloce della DFT.

Se conviene lavorare nel dominio delle frequenze o nel dominio spaziale dipende dalla situazione: quando abbiamo kernel relativamente piccoli conviene sempre lavorare nel dominio spaziale, viceversa conviene lavorare nel dominio delle frequenze.

## LEZ 13 – Morfologia applicata alle immagini

La morfologia di un'immagine descrive le forme rappresentate nell'immagine. Gli oggetti sono visti come un *insieme* di punti del piano (i pixel), di conseguenza i *processi morfologici* sono definiti come operazioni su tali insiemi.

Per semplicità si considerano insiemi  $B$  che appartengono a  $Z^2$  (immagini binarie, dunque si considera il colore binario del pixel) ma queste operazioni possono essere generalizzate anche ad altri domini.

Consideriamo un'immagine binaria  $f$ :

- Se  $f(x,y)$  è bianco, allora  $(x,y)$  appartiene all'insieme  $B$ ;
- Se  $f(x,y)$  è nero, allora  $(x,y)$  NON appartiene all'insieme  $B$

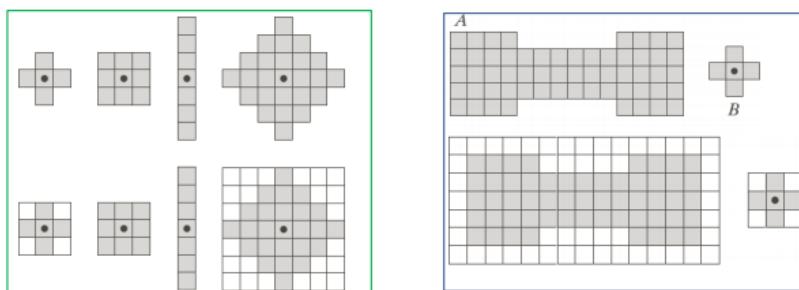
Di conseguenza,  $B$  sarà costituito da tutti i punti bianchi ( $f(x,y) = 1$ ).

Le operazioni morfologiche sono definite rispetto ad un *insieme particolare*, chiamato **elemento strutturante ES**: sono array di pixel definiti relativamente ad un'origine, tipicamente il centro di gravità.

Per convenzione: le celle riempite appartengono all'ES, le celle vuote NON appartengono all'ES, le croci rappresentano i pixel "don't care" (che siano bianche o nere è indifferente), un puntino nero rappresenta l'origine dell'ES.

Gli *ES* possono avere una forma arbitraria, dunque per poterli utilizzare devono essere prima trasformati in array di pixel (dovranno avere una forma quadrata/rettangolare).

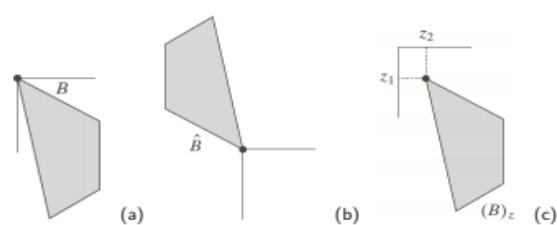
Le *immagini* invece hanno sempre una forma quadrata/rettangolare, tuttavia dobbiamo stare attenti al fatto che l'ES possa essere completamente contenuto nell'immagine senza che questo fuoriesca (cosa che può avvenire lungo i bordi), per questo aggiungiamo del *padding* all'immagine.



Nella morfologia matematica sono di fondamentale importanza i **concetti** di *riflessione* e *traslazione*:

- La **riflessione** consiste nella rotazione di 180° di un ES intorno alla sua origine, effetto che si ottiene banalmente *invertendo il segno* delle coordinate  $(x,y)$ ;
- La **traslazione** consiste nel *traslare* tutti i punti di  $B$  di un certo valore  $z$ .

- La riflessione,  $\hat{B}$ , è definito come:  $\hat{B} = \{-\mathbf{b} | \mathbf{b} \in B\}$
- La traslazione di  $B$  tramite  $z$ ,  $(B)_z$ , è definita come:  $(B)_z = \{\mathbf{b} + z | \mathbf{b} \in B\}$



## Operatori morfologici: erosione e dilatazione

Molti algoritmi morfologici si basano sue due operazioni "primitive": *erosione* e *dilatazione*. Data un'immagine A e un elemento strutturante B, diremo:

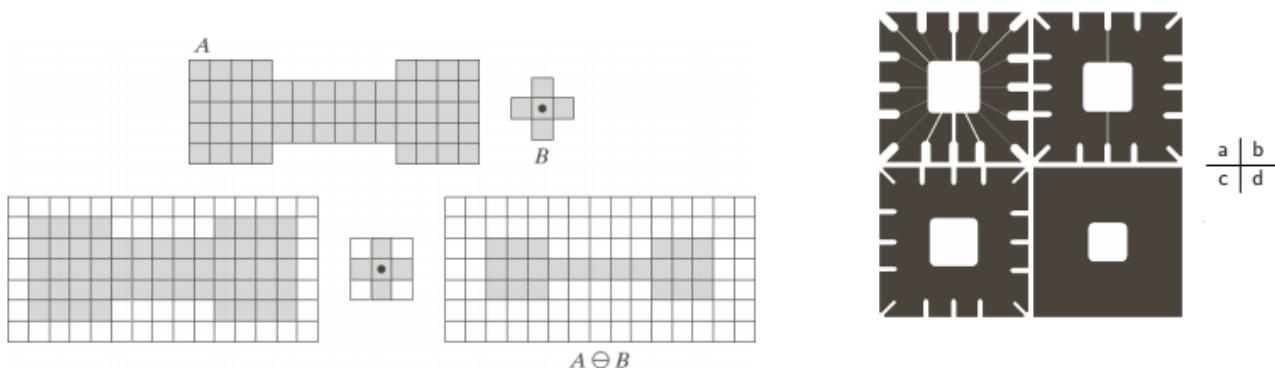
L'**erosione** di A attraverso B,  $A \ominus B$ , è definita come l'insieme di tutti i punti z tali che B traslato di z sia contenuto in A.

Dire che B deve essere contenuto in A equivale a dire che B non ha elementi in comune con lo sfondo, di conseguenza possiamo definire l'erosione come "tutti i punti di z tali che B intersecato con il *complemento* di A ( $A^c$ , ovvero lo sfondo) dia un insieme vuoto".

$$A \ominus B = \{z | (B)_z \cap A^c = \emptyset\}$$

L'erosione può essere utilizzata per realizzare un filtraggio morfologico, dove maggiore è la dimensione dell'ES maggiore sarà l'erosione.

L'operazione consiste fondamentalmente in un'operazione di convoluzione, dove si fa scorrere l'ES lungo l'immagine. L'effetto ottenuto è che l'oggetto viene **rimpicciolito**, dunque "eroso". Nello specifico, l'erosione cancella i dettagli più piccoli dell'ES.

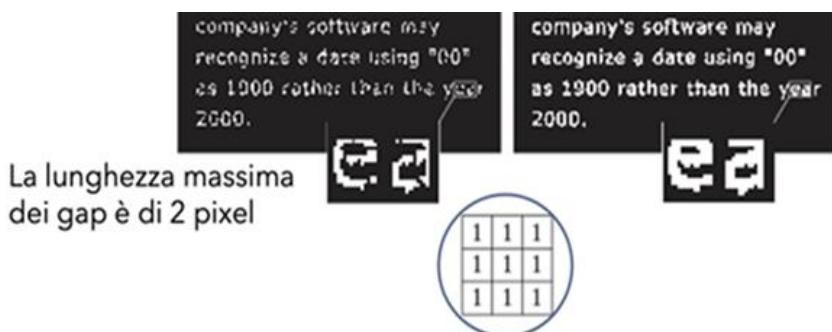


La **dilatazione** di A attraverso B,  $A \oplus B$ , è definita come l'insieme di tutti i punti z tali che B riflesso ed A si sovrappongano almeno per un elemento (la loro intersezione NON deve dare un insieme vuoto).

Dire che B riflesso intersecato A non deve dare un insieme vuoto equivale a dire che tutti gli elementi di B riflesso sono contenuti in A, di conseguenza possiamo definire la dilatazione come "tutti i punti di z tali che B riflesso intersecato A appartenga ad A".

$$A \oplus B = \{z | ((\hat{B})_z \cap A) \subseteq A\}$$

(Anche) la dilatazione può essere utilizzata per realizzare un filtraggio morfologico. L'operazione produce effetti simili a quelli del filtraggio passa-basso: i dettagli sono assorbiti. Utile nel caso si vogliano **riempire** le interruzioni.



## • Dualità

Erosione e dilatazione sono operazioni **duali** l'una dell'altra rispetto al complemento e alla riflessione. Cioè:

- L'erosione di A attraverso B è il complemento della dilatazione di  $A^C$  attraverso  $\hat{B}$ ;
- La dilatazione di A attraverso B è il complemento dell'erosione di  $A^C$  attraverso  $\hat{B}$ . (Viceversa, quindi).

$$(A \ominus B)^C = A^C \oplus \hat{B} \quad (A \oplus B)^C = A^C \ominus \hat{B}$$

La proprietà di dualità è particolarmente utile quando l'elemento strutturante è simmetrico rispetto alla sua origine (come si verifica spesso), così che  $\hat{B} = B$ . Quindi, possiamo ottenere l'erosione di un'immagine attraverso B semplicemente dilatando il suo sfondo (ovvero  $A^C$ ) con lo stesso ES e complementando il risultato (viceversa per la dilatazione).

## Operatori morfologici: apertura e chiusura

Gli operatori morfologici di apertura e chiusura sono definiti utilizzando gli operatori di erosione e dilatazione.

L'**apertura** di A attraverso B,  $A \circ B$ , è definita come l'esecuzione in sequenza delle operazioni di *erosione* e *dilatazione*.

$$A \circ B = (A \ominus B) \oplus B$$

L'apertura rende generalmente più omogenei i contorni di un oggetto, elimina le piccole interruzioni e le protuberanze sottili.

Geometricamente, l'apertura è rappresentata dai punti di A coperti dalla traslazione di B lungo il bordo interno di A.

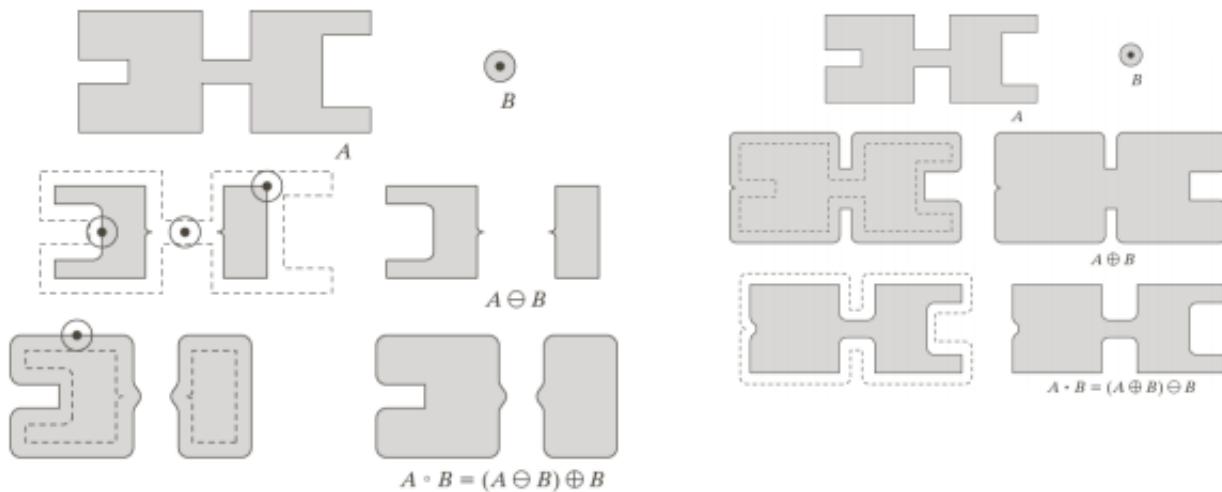
La **chiusura** di A attraverso B,  $A \bullet B$ , è definita come l'esecuzione in sequenza delle operazioni di *dilatazione* ed *erosione*.

$$A \bullet B = (A \oplus B) \ominus B$$

La chiusura rende anch'essa generalmente più omogenei i contorni di un oggetto ma, al contrario dell'apertura, riempie i buchi e i buchi troppo stretti.

Geometricamente, la chiusura aggiunge ad A quei punti del background che non sono coperti dalla traslazione di B lungo il bordo esterno di A.

## Esempi di Apertura e Chiusura



### • Dualità

Apertura e chiusura sono operazione **duali** l'una dell'altra rispetto al complemento e alla riflessione (come nel caso dell'erosione e della dilatazione).

Dunque l'apertura di A attraverso B è il complemento della chiusura di  $A^c$  attraverso  $\hat{B}$ , e viceversa.

$$(A \bullet B)^c = A^c \circ \hat{B}$$

$$(A \circ B)^c = A^c \bullet \hat{B}$$

### • esempio di applicazione

L'apertura e la chiusura possono essere usate per il filtraggio del rumore.

Supponiamo di avere un'immagine A affetta da rumore, che noi vogliamo rimuovere:

Applichiamo un'operazione di apertura andiamo ad eliminare il rumore esterno attraverso l'erosione, che tuttavia migliora il rumore interno, che a sua volta può essere eliminato attraverso la dilatazione.

Il rumore è stato rimosso dall'apertura, ma questa fase di elaborazione ha causato l'interruzione di alcune dorsali. Per questo utilizziamo la dilatazione per recuperare la dilatazione delle dorsali e successivamente l'erosione per ripristinare il loro spessore originale.



## Trasformazioni hit-or-miss

La trasformazione morfologica hit-or-miss è uno strumento che permette di individuare degli oggetti all'interno di un'immagine. Questi oggetti (aka forme) devono essere *disgiunti*, dunque devono essere separati da almeno un pixel di background.

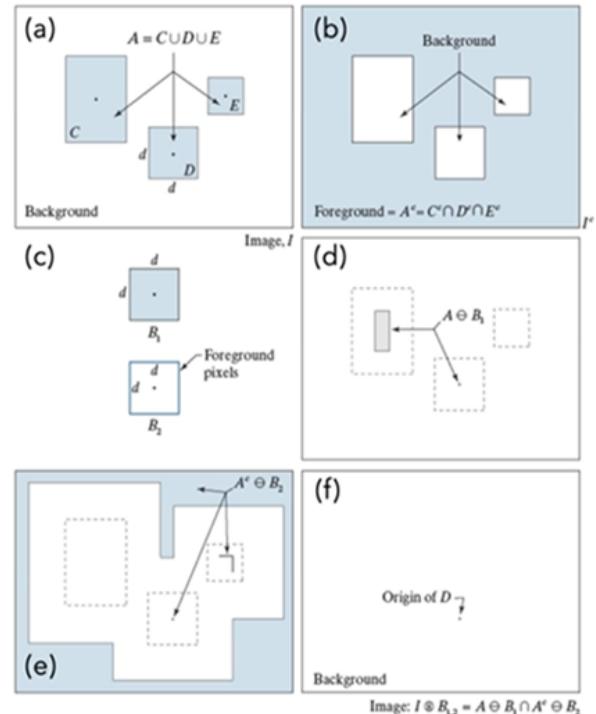
L'elaborazione è basata su due elementi strutturati:  $B_1$ , con la forma dell'oggetto da individuare, e  $B_2$ , per determinare le forme nel background.

La trasformazione di un'immagine  $I$  sarà definita come "tutti i punti di  $z$  tali che  $B_1$  appartenga ad  $A$  e  $B_2$  appartenga al *complemento di A*", che **in pratica** significa fare l'intersezione fra l'erosione di  $A$  attraverso  $B_1$  e l'erosione di  $A^c$  attraverso  $B_2$ .

$$I \odot B_{1,2} = \{z | (B_1)_z \subseteq A \text{ e } (B_2)_z \subseteq A^c\} = (A \ominus B_1) \cap (A^c \ominus B_2)$$

*Esempio:*

- Supponiamo di voler trovare la posizione dell'origine dell'oggetto D in I
  - (a) Immagine che consiste di un foreground che è l'unione,  $A$ , di un insieme di oggetti, ed un sfondo
  - (b) Immagine con il suo foreground definito come  $A^c$
  - (c) ES per determinare l'oggetto D
  - (d) erosione di  $A$  tramite  $B_1$
  - (e) erosione di  $A^c$  tramite  $B_2$
  - (d) Intersezione di (d) ed (e) che mostra la posizione dell'origine di D. I punti sono le origini dei rispettivi componenti. Ogni punto è un pixel



## Morfologia in scala di grigio

Finora abbiamo trattato la morfologia per le immagini binarie, ma le operazioni viste prima possono essere estese anche alle immagini in scala di grigi.

In questo caso le immagini devono essere considerate come funzioni (e non più come insiemi). Possono essere concepiti due tipi di elementi di strutturazione: *flat* e *non flat*.

### Erosione e dilatazione attraverso ES flat

L'**erosione** di  $f$  attraverso un elemento strutturante *flat*  $b$  è definita, per ogni posizione  $(x,y)$ , come il valore *minimo* dell'immagine nella regione coincidente con  $b$ , quando l'origine di  $b$  si trova in  $(x,y)$ .

$$[f \ominus b](x, y) = \min_{(s, t) \in b} \{f(x + s, y + t)\}$$

La **dilatazione** di  $f$  attraverso un elemento strutturante  $b$  è definita, per ogni posizione  $(x,y)$ , come il valore *massimo* dell'immagine nella finestra indicata da  $\hat{b}$  quando l'origine di  $\hat{b}$  si trova in  $(x,y)$ .

**nb:**  $\hat{b} = b(-x, -y)$ .

$$\begin{aligned}[f \oplus b](x, y) &= \max_{(s, t) \in \hat{b}} \{f(x + s, y + t)\} \\ &= \max_{(s, t) \in b} \{f(x - s, y - t)\}\end{aligned}$$

### Apertura e chiusura attraverso ES flat

L'apertura e chiusura sono analoghe al loro corrispettivo binario, dunque:

L'**apertura** di  $f$  attraverso  $b$ ,  $f \circ b$ , è definita come l'esecuzione in sequenza delle operazioni di *erosione* e *dilatazione*.

$$f \circ b = (f \ominus b) \oplus b$$

La **chiusura** di  $f$  attraverso  $b$ ,  $f \bullet b$ , è definita come l'esecuzione in sequenza delle operazioni di *dilatazione* ed *erosione*.

$$f \bullet b = (f \oplus b) \ominus b$$

Valgono anche le proprietà di **dualità**:

$$\begin{aligned}(f \bullet b)^c &= f^c \circ \hat{b} \\ (f \circ b)^c &= f^c \bullet \hat{b}\end{aligned}$$

Le operazioni di apertura limano i picchi, mentre quelle di chiusura riempiono le valli. In pratica tali operazioni corrispondono al filtraggio passa alto/passa basso.

Dalla combinazione delle operazioni di apertura e chiusura si ottiene lo **smoothing morfologico**.

### Trasformazioni top-hat e bottom-hat

Le *trasformazione top-hat* e *bottom-hat* sono la conseguenza della combinazione della sottrazione di un'immagine con la sua apertura o chiusura.

La trasformazione **top-hat** di un'immagine in scala di grigio  $f$  è definita come  $f$  meno la sua apertura:

$$T_{\text{hat}} = f - (f \circ b)$$

La trasformazione **bottom-hat** di un'immagine in scala di grigio  $f$  è definita come  $f$  meno la sua chiusura:

$$B_{\text{hat}} = (f \bullet b) - f$$

Queste trasformazioni preservano l'informazione rimossa delle operazioni di apertura e chiusura, rispettivamente. In particolar modo la top-hat viene usata quando si hanno oggetti chiari su sfondo scuro, la bottom-hat quando si hanno oggetti scuri su sfondo chiaro.

Tale operazioni sono utili per correggere l'illuminazione.

